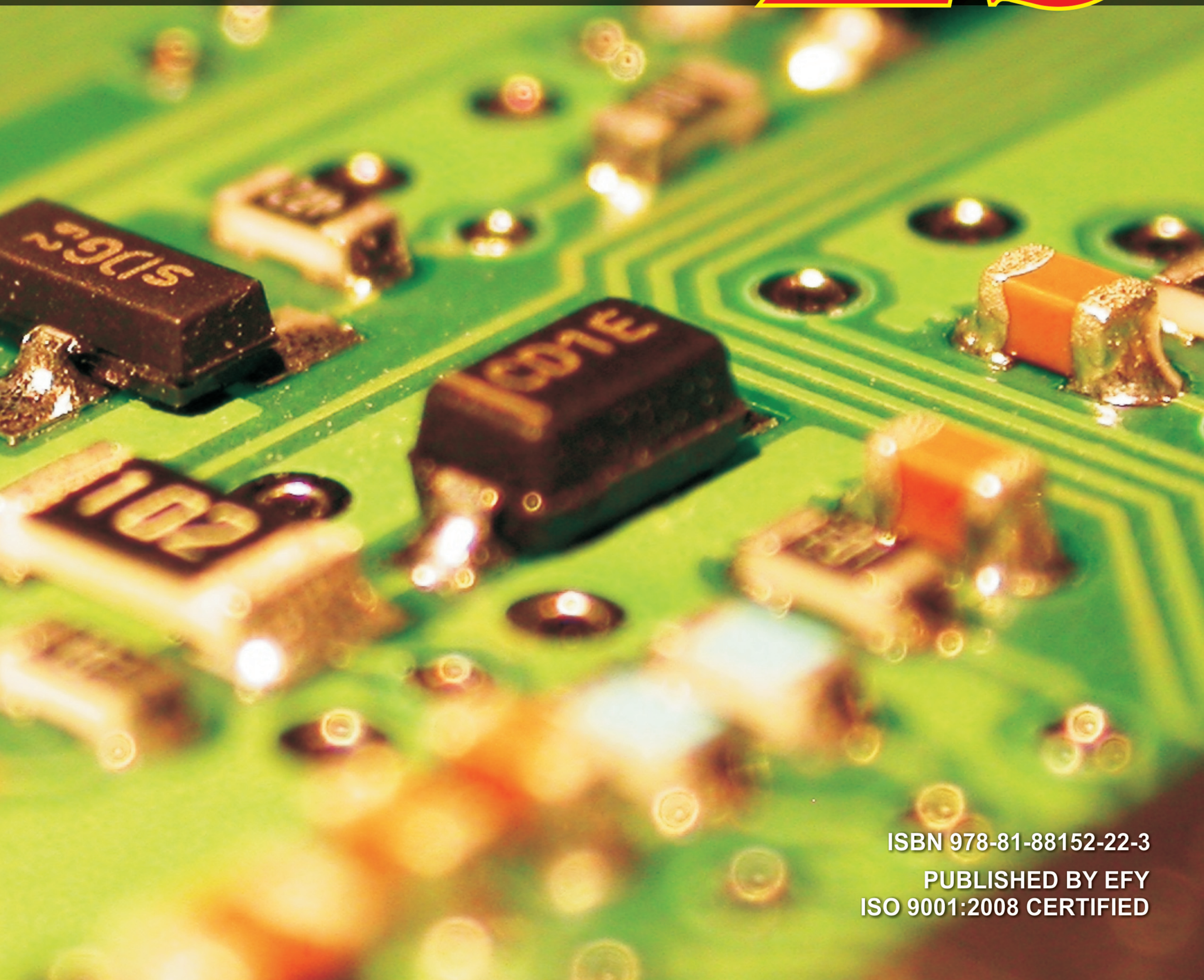
Rs 200/-
(with CD)

ELECTRONICS PROJECTS VOL. 25

A Compilation of 89 tested Electronic Construction Projects and Circuit Ideas for Professionals and Enthusiasts



ISBN 978-81-88152-22-3

PUBLISHED BY EFY
ISO 9001:2008 CERTIFIED

Electronics Projects
Vol. 25

First Published in this Edition, January 2010

ISBN 978-81-88152-22-3

Published by Ramesh Chopra for EFY Enterprises Pvt Ltd,
D-87/1, Okhla Industrial Area, Phase 1, New Delhi 110020.
Typeset at EFY Enterprises Pvt Ltd and
Printed at: Shree Gobind Printers
Y-56, Okhla Phase 2, New Delhi 110020.

ELECTRONICS PROJECTS VOL. 25

EFY Enterprises Pvt Ltd
D-87/1 Okhla Industrial Area, Phase 1
New Delhi 110020

FOREWORD

This volume of Electronics Projects is the twenty forth in the series published by EFY Enterprises Pvt Ltd. It is a compilation of 23 construction projects and 66 circuit ideas published in 'Electronics For You' magazine during 2004.

We are also including a CD with this volume, which not only contains the datasheets of major components used in construction projects but also the software source code and related files pertaining to various projects. This will enable the reader to copy these files directly on to his PC and compile/run the program as necessary, without having to prepare them again using the keyboard. In addition, the CD carries useful software, tutorials and other goodies (refer 'contents' in CD).

In keeping with the past trend, all modifications, corrections and additions sent by the readers and authors have been incorporated in the articles. Queries from readers along with the replies from authors/EFY have also been published towards the end of concerned articles. It is a sincere endeavour on our part to make each project as error-free and comprehensive as possible. However, EFY cannot take any responsibility if readers are unable to make a circuit successfully, for whatever reason.

This collection of tested circuit ideas and construction projects in a handy volume would provide all classes of electronics enthusiasts—be they students, teachers, hobbyists or professionals—with a valuable resource of electronic circuits, which can be fabricated using readily-available and reasonably-priced components. These circuits could either be used independently or in combination with other circuits, described in this and other volumes. We are confident that this volume, like its predecessors, will generate tremendous interest amongst the readers.

CONTENTS

Section A: Construction Projects

Section B: Circuit Ideas:

SECTION A :
CONSTRUCTION PROJECTS

MICROCONTROLLER-BASED CALL INDICATOR

UDAY B. MUJUMDAR

In large establishments, such as hotels and hospitals, intercoms and call bell systems are essential for communication between inmates and the assisting staff. Intercom being a costlier option, in many the relatively inexpensive call indication systems are preferred. The call indication system gives an audio-visual indication of the call point (room or cabin number).

In conventional call indication systems, different call points are connected to the indication system via separate cables. This makes the installation complicated and costly, especially when the number of calling points is quite large.

Here's a simple and economical system for call point identification and display. The system has the following features:

1. Uses only two wires for connecting different call points.
2. Up to 36 call points (in two circuits comprising 18 call points each) can be connected.
3. The control panel has a bright display for visual indication of call point with floor number and a buzzer for audible indication. The buzzer can be snoozed using the Call Acknowledge key.
4. The call point number can be changed without changing the wiring.
5. The system can be expanded to accomodate more call points in the future.

System overview

Fig. 1 shows block diagram of the call indication system. The system comprises different call points connected to a control panel through a two-core shielded cable. The call points are arranged in two circuits. The maximum number of call points connected per circuit is 18. Thus a total of 36 call points can be connected. The two-circuit system is useful when the call points are on different floors.

Fig. 2 shows connection of different call points arranged in two circuits. Table I shows connection details for different numbers of call points on the same and different floors.

The hardware

Fig. 3 shows the microcontroller-based call indication system built around Motorola's MC68HC705J1A microcon-troller. The system comprises four main sections, namely, call-point detection section, analogue-to-digital conversion (ADC) section, display section and microcont-roller section.

The call-point detection section detects the key pressed from a call point. A fixed DC voltage (decided by a resistor in series with the key) is transmitted to the ADC section through the cable. The ADC section converts the analogue signal into equivalent digital data. The microcontroller decodes the data and displays the call point number accordingly.

The call-point detection section. The two-core shielded cable connects the call points internally as well as to the control unit. A shielded cable is used because it reduces the noise. Rext-1 through Rext-18 are the resistors in series with keys 1 through 18 (refer Fig. 2). The values of resistors are the same for both call-point circuit-1 and call-point circuit-2.

Call-point circuits 1

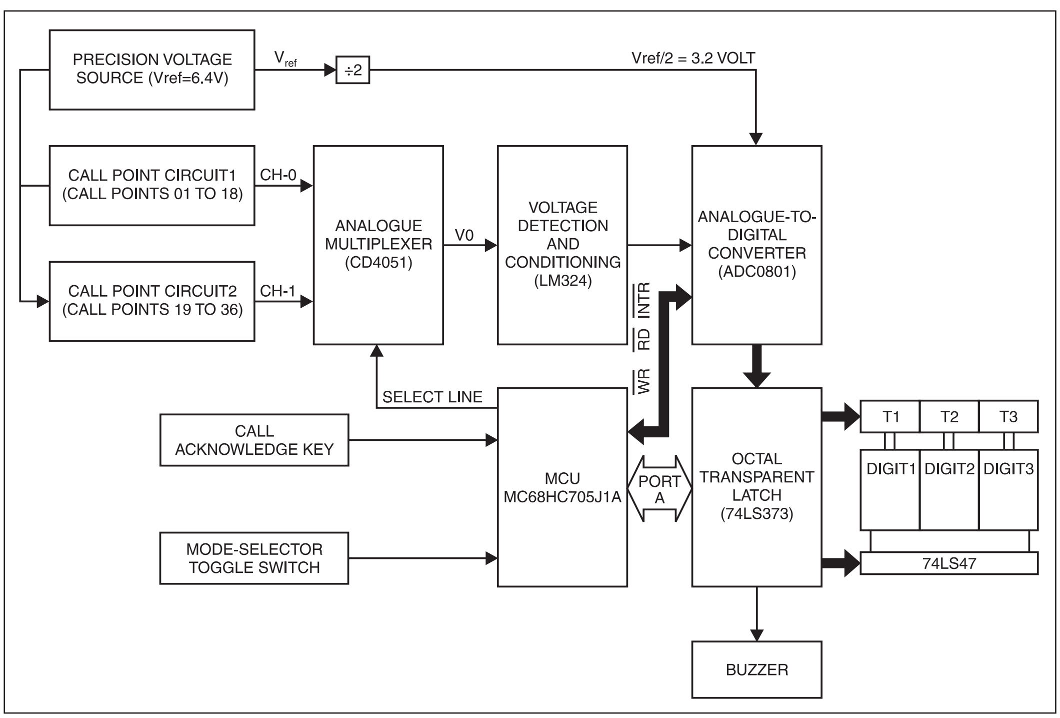


Fig. 1: Block diagram of the microcontroller-based call indicator system

and 2 are powered by a fixed, temperature-compensated 6.4V reference source. R24 and R25 (1.2k) limit the current drawn from the source. R22 and R23 are fixed resistors connected on the circuit board. Capacitors C11 and C12 (0.1 µF) bypass the noise signals. Voltage V0 is the voltage drop across internal resistors R22 and R23 when any key is pressed. The typical value of the external resistor, corresponding call point number and voltage drop (V0) across R22 or R23 for each key are shown in Table II. The voltage V0 is decided by the key pressed (precisely by the resistor in series with that key). This voltage is transmitted via the two-core cable to the main circuit.

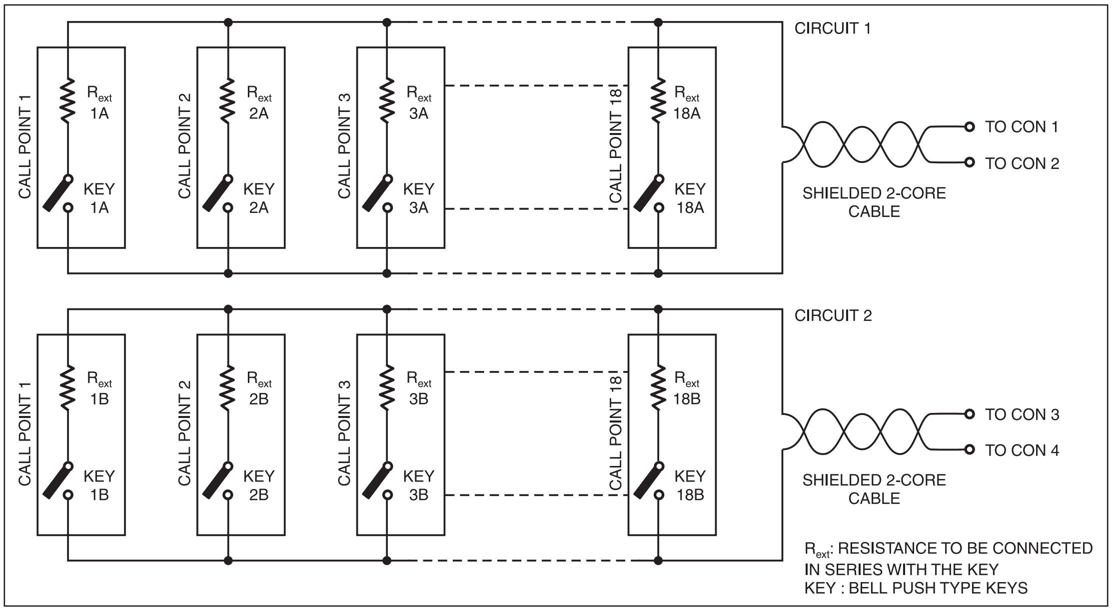


Fig. 2: Wiring of 36 call points arranged in two circuits

CMOS analogue multiplexer CD4051 (IC6) is a single 8-channel multiplexer having binary control inputs A, B and C. The three binary signals select one of the eight channels and connect it to the output. Fig. 3 shows the connection details of IC CD4051. Here only two channels of IC CD4051 have been used.

The outputs of the two circuits are continuously scanned using the multiplexer. The output of call-point circuit 1 is connected to channel-0 (CH-0) and the output of call-point circuit 2 is connected to channel-1(CH-1).

The fixed temperature-compensated voltage source Vref is derived from National Semiconductor's active temperature-compensated reference zener diode LM329 (IC5). It is very essential to have a temperature-compensated voltage source as a little change in V_{ref} changes the voltage drop (V0) across resistor R22 or R23, leading to wrong identification of the call point number. IC LM329 gives a fixed output voltage of 6.9 volts. It has a very low dynamic impedance of 0.8 ohm. The low impedance minimises the errors due to input voltage variations, load variations and feed resistor drift.

When all the keys are open (no key is pressed), voltage V_0 is zero. When any key is pressed, V_0 is given by:

where R24 is 1.2 kilo-ohms and R22 is 10 kilo-ohms.

$$V_0 = V_{ref} \times \frac{R22}{(R24+R_{ext}+R22)} \text{ volts}$$

IC LM324 comprising N1 through N4 (IC4) is used as a voltage follower to buffer the respective voltage signals.

The ADC section. The potential drop across resistor R22 on pressing a key varies from 0 volt to 5.25 volts (refer Table II). This analogue voltage is converted into digital equivalent by IC ADC0801 (IC2).

IC ADC0801 is an 8-bit, successive approximation-type, CMOS analogue-to-digital converter housed in a 20-pin dip package. The input voltage for the converter can range from 0 to 5 volts and it operates off a single power supply of 5 volts. It has two inputs, namely, $V_{in}(+)$ pin and $V_{in}(-)$ pin, for the differential analogue signal. When the analogue signal is single-ended positive, i.e. it varies from 0 volt to 5 volts, $V_{in}(+)$ pin is used as the input and V_{in} (–) pin is grounded.

The converter requires a clock at pin 4 (CLK-IN); the frequency can range from 100 kHz to 800 kHz. The user has two options: one is to connect an external clock at CLK-IN and the other is to use the built-in internal clock by connecting a resistor and a capacitor externally at pins 19 and 4, respectively. Here we've used the second option for giving clock pulses to the ADC. The frequency (f) is calculated from the following relationship:

f= 1/1.1RC Hz

The three control signals of ADC0801 ($\overline{CS}$, $\overline{WR}$ and $\overline{RD}$) are used for interfacing. It is enabled when chip-select $\overline{CS}$ goes low. When write line $\overline{WR}$ goes low, the

TABLE I

Distribution of Call Point	Suggested Mode	Call Point Connection Details	Numbering of Call Point
1. 18 or more call points on the same floor	0	1. Connect call point circuit-1 output to CON1 and CON2 2. Connect call point circuit-2 output to CON3 and CON4	001 to 036
2. 18 or less call points only 1st floor	1	1. Connect call point circuit-1 output to CON1 and CON2 2. CON3 and CON4 unused	001 to 018
3. 18 or less call points on two floors	1	1. Connect call point circuit-1 output to CON1 and CON2 one floor. 2. Connect call point circuit-1 output to CON1 and CON2 second floor.	001 to 018 101 to 118

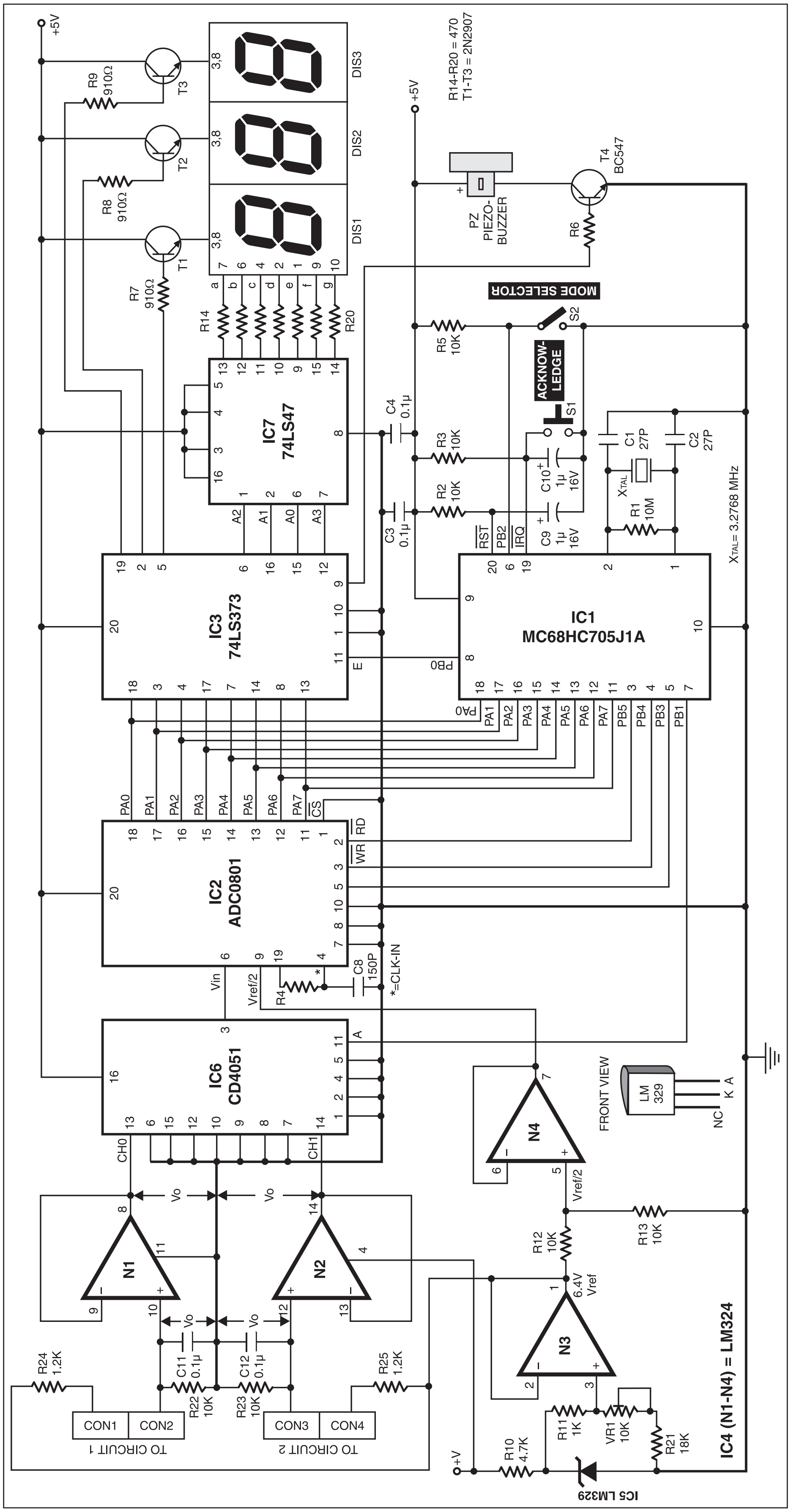


Fig. 3: Circuit diagram of microcontroller-based call indicator system

internal successive approximation register (SAR) is reset, and the output lines go to high-impedance state. When WR transits from low to high state, the conversion begins. When the conversion is completed, the interrupt request line $\overline{INTR}$ is asserted low and the data is placed on the output lines. The $\overline{INTR}$ signal can be used to know the completion of conversion. When the data is read by asserting read line $\overline{RD}$ low, the $\overline{INTR}$ is reset.

When Vcc is 5 volts, the input voltage (V_{in}(+)) can range from 0 to 5 volts and the corresponding output is 00H to FFH. However, the full-scale output can be restricted to a lower range of inputs by using pin $V_{ref}/2$. The voltage at pin $V_{ref}/2$ decides the conversion step size.

An optimal step size of ADC is 25 mV (6.4/256 = 25 mV). Thus an analogue voltage signal of 6.4 volts at pin V_{in}(+) gets converted into FFH (11111111b) at the output data pins. The ADC clock frequency is about 600 kHz. This gives a conversion period of approximately 100 microseconds.

The ADC continuously converts the analogue input into digital data. This minimises the chances of malfunctioning when keys from two or more call points are simultaneously pressed.

Table II gives the analogue voltage V0 and its digital equivalent for different call points. When no key is pressed, V0 is nearly zero and its digital equivalent produced by the ADC is 00D. When a key is pressed, the digital equivalent varies from 10D to 200 D (refer Table II). This digital data is further processed by the microcontroller into the equivalent call-point number.

The latch and display section. The display device is an interface between the user and the machine. The call-point location information is displayed on three 7-segment displays (DIS1 through DIS2) driven using the time-multiplexed technique. DIS1 displays the floor number, while DIS2 and DIS3 display the call point number.

PARTS LIST	
Semiconductors:	
IC1	- MC68HC705J1A microcontroller
IC2	- ADC0801 8-bit analogue-to-digital converter
IC3	- 74LS373 octal transparent latch
IC4	- LM324 quad operational amplifier
IC5	- LM329 temperature-compensated ref. diode
IC6	- CD4051 analogue multiplexer
IC7	- 74LS47 BCD-to-7-segment decoder/driver
IC8	- 7805 +5V regulator
T1-T3	- 2N2907 npn transistor
T4	- BC547 npn transistor
D1, D2	- IN4007 rectifier diode
DIS1-DIS3	- LTS542 7-segment common-anode display
Resistors (all ¼-watt, ±5% carbon, unless stated otherwise):	
R_{ext-1}-R_{ext-18} (values are stated in Table II)	
R1	- 10-mega-ohm
R2-R6, R12, R13	- 10-kilo-ohm
R7-R9	- 910-ohm
R14-R20	- 47-ohm
R10	- 4.7-kilo-ohm
R11	- 1-kilo-ohm
R21	- 18-kilo-ohm
R22, R23	- 10-kilo-ohm
R24, R25	- 1.2-kilo-ohm
Capacitors:	
C1-C4, C6, C7	- 27pF ceramic disk
C11, C12	- 0.1µF ceramic disk
C5	- 1000µF, 25V electrolytic
C8	- 150pF ceramic disk
C9, C10	- 1µF, 16V electrolytic
Miscellaneous:	
X1	- 230V primary to 12V-0-12V, 300mA secondary transformer
XTAL	- 3.2768MHz crystal
PZ	- Piezobuzzer
S1, S2	- Push-to-on switch

All the three 7-segment displays share common input lines. Data entered for the first digit enables only the first 7-segment display. After a few milliseconds, the data for the first digit is replaced by that of the next digit, but this time only the second display is enabled. After all the digits are displayed in this way, the cycle repeats. Because of this repetition at 100 times a second, there is an illusion that all the digits are being continuously displayed. BCD-to-7-segment decoder/driver 74LS47 (IC7) and 2N2907 (T1 through T3) are used for driving the common-anode displays.

Port A of the microcontroller (IC1) is used for reading the ADC output as well as the data display. Octal transparent latch 74LS373 (IC3) is used to avoid the bus contention. While refreshing the displays, the latch is made transparent and the data is displayed digitwise. During this period, the data lines of ADC0801 are in high-impedance state as $\overline{RD}$ and $\overline{WR}$ are high. Once all the digits are refreshed, the latch is made non-transparent. Now if there is any change in the data line of the ADC, it will not be reflected on the data displayed.

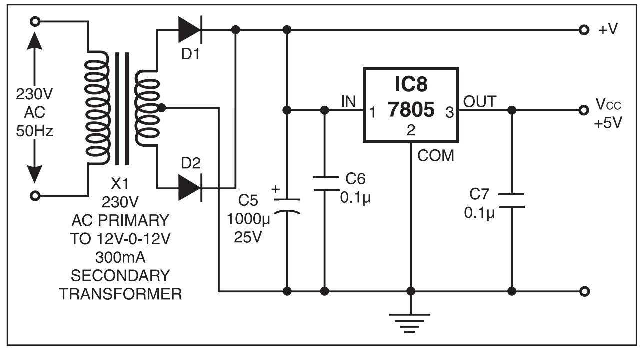


Fig. 4: Circuit diagram of power supply

Let's assume that the data to be displayed is 126. BCD equivalent of 1 (0001) is placed on the input lines of IC 74LS47 (IC7). IC 74LS47 gives the 7-segment equivalent data of 01. Now digit '1' is selected using transistor T1 and displayed on DIS1 for about 2 milliseconds. In a similar way, digits '2' and '3' are displayed on DIS2 and DIS3 for 2 ms each with the help of transistors T2 and T3, respectively. The digit is refreshed every 10 ms. The timer interrupt generates the interrupt every 10 ms. The displays are refreshed during the timer interrupt service routine.

The microcontroller section. Motorola's MC68HC705J1A microcontroller (IC1) is programmed to perform the following functions:

- Scan the keys to detect pressing of any key
- Read the data from ADC0801
- Identify the destination where key is pressed
- Display the call point number and also give audio indication
- Check for the pressing of Acknowledge key to snooze the buzzer

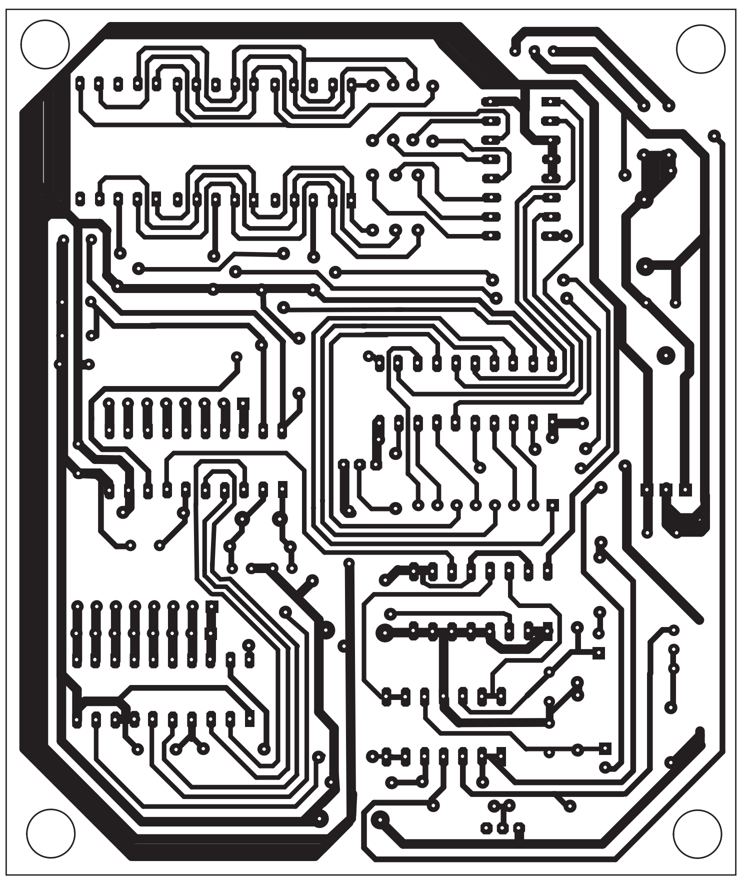


Fig. 5: Actual-size, single-side PCB layout for microcontroller-based call indicator

Fig. 3 shows how the different sections are connected to the microcontroller. Port A is used for reading the data from the ADC as well as the display. When the controller reads the ADC, port A is in input mode; while during data display, the same port is configured in output

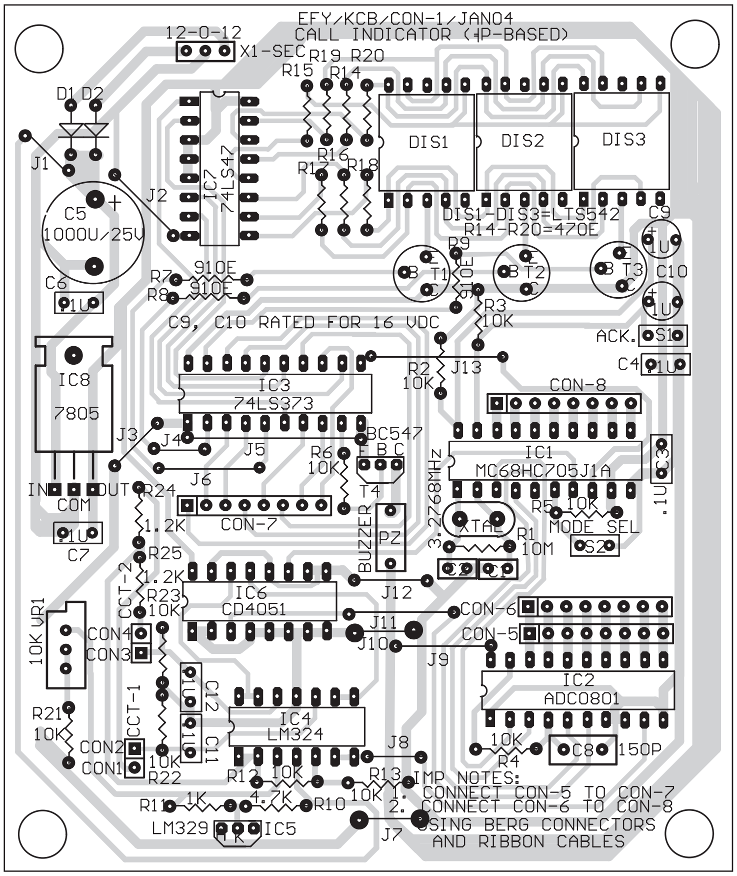


Fig. 6: Component layout for the PCB

mode. Port B is used for controlling the ADC and the latch.

Power supply. The power supply circuit is shown in Fig. 4. The AC mains supply is stepped down by transformer X1 to deliver a secondary output of 12V-0-12V AC, 300mA. The output of the transformer is rectified by a full-wave rectifier comprising diodes D1 and D2 and filtered by capacitor C5. The direct +V output is used for IC LM324 (IC4) and the reference circuit, while the regulated 5V from regulator IC 7805 (IC8) powers the entire circuit excluding IC4 and the reference circuit.

An actual-size, single-side PCB of Figs 3 and 4 is shown in Fig. 5, with its component layout in Fig. 6.

TABLE II

R_{ext}	$V_o = \frac{64 \times 10k}{(11.2k + R_{ext})}$	ADC Equi $D = \frac{V_o}{25 \times 10^{-3}}$	Range of identification	Call point number
220k (R_{ext-1})	64/231.2=0.2768	11	07-15	01
100k (R_{ext-2})	64/111.2=0.5755	23	16-26	02
68k (R_{ext-3})	64/79.2=0.808	33	27-38	03
47k (R_{ext-4})	64/58.2=1.0996	44	39-50	04
33k (R_{ext-5})	64/44.2=1.4479	58	51-62	05
27k (R_{ext-6})	64/38.2=1.6753	67	63-72	06
22k (R_{ext-7})	64/33.2=1.9277	77	73-82	07
18k (R_{ext-8})	64/29.2=2.1917	88	83-92	08
15k (R_{ext-9})	64/26.2=2.4427	98	93-103	09
12k (R_{ext-10})	64/23.2=2.7586	110	104-114	10
10k (R_{ext-11})	64/21.2=3.0188	120	115-125	11
8.2k (R_{ext-12})	64/19.4=3.2989	132	126-137	12
6.8k (R_{ext-13})	64/18=3.5555	142	138-147	13
5.6k (R_{ext-14})	64/16.8=3.8009	152	148-156	14
4.7k (R_{ext-15})	64/15.9=4.0251	161	157-170	15
3k (R_{ext-16})	64/14.2=4.5070	180	171-188	16
2k (R_{ext-17})	64/13.2=4.8484	194	189-200	17
1k (R_{ext-18})	64/12.2=5.2459	210	200 and above	18

The software

Motorola offers integrated development environment (IDE) software for programming the microcontroller and complete development of the system. The development tool comes with editor, assembler and programmer software to support Motorola's device programmer and a software simulator. The ICS05JW in-circuit simulator, along with the development board (pod), forms a complete simulator and non-real-time input/output emulator for simulating, programming and debugging the code of an MC68HC705J family device.

When you connect the pod to your host computer and target hardware, you can use the actual inputs and outputs of the target system during simulation of the code. You can also use the ISC05JW software to edit and assemble the code in standalone mode, without input/output to/from the pod. The pod (MC68HC705J1CS) can be interfaced to any IBM computer running on Windows 3.x/Windows 9X using the serial port.

The software routines for the call bell indicator, along with their Assembly language code, are given in Appendix 'A'. The following functions are performed by the software program:

1. Initialise ports A and B, timer and display
2. Monitor pressing of keys using the ADC
3. Display the data
4. Identify the call point number

For perfect functioning of any system, the associated software requires many data manipulation tricks and internal branching. Here the software is divided into Initialise, Identify, DispCon, Refresh, Read and Acknowledge modules. The sequence of operation and logic can be understood from the program listing. A brief description of each module is given below.

Init. Initially ports A and B are assigned as the output ports. The latch is made transparent and the display shows 000, indicating no key is pressed. The timer interrupt is initialised to give an interrupt every 10 ms.

Identify. In this part of the program, the ADC data is analysed and the call point destination is identified. If any key is found pressed, the particular call point number is stored in hex form in the display register.

DispCon. This part of the software is used for finding out the decimal equivalent of hex data. The microcontroller manipulates the data, which is essentially in hex, but for display purpose, data should be in BCD.

Refresh. The timer of the microcontroller is initialised to give an interrupt every 10 ms. For multiplexed display, it is mandatory to refresh the displays every 10 ms. During the timer interrupt service routine, the microcontroller refreshes the displays and reads the ADC data.

Acknowledge. The call can be acknowledged by using the Acknowledge key. When a call is acknowledged, the display shows 000 and the buzzer (PZ) snoozes.

Installation of the call indicator

Depending on the number of call points, connect the call points in a single circuit or arrange them in two circuits. The display indication will vary accordingly.

Normally, the call points are in different rooms. Rext is the resistor that decides the call point number. It is connected in series with the keys. For making the call, Bell-type push switches are used. Resistor Rext is placed inside the switch. The change in call point number can be implemented just by changing Rext.

EFY note. The software program Callnew.asm, along with the Callnew.S19 file and relevant datasheet, are included in the CD.

An actual-size, single-side PCB of Figs 3 and 4 is shown in Fig. 5, with its component layout in Fig. 6

CALLNEW.ASM

```
callnew.asm        Assembled with CASMW  10/17/03  10:33:13 AM  PAGE 1
                   1 * Call Indicator Using Motorola Micro-controller
                     MC68HC705J1A.
                   2 * Developed By : Uday B.Mujumdar,Lecturer,
                     Shri Ramdeobaba Kamla Nehru Engineering
                   3 *          College,Nagpur.
                   4 ***********************************************
00C0               5                  org  $00c0
00C0               6 digit_1          rmb  1  ;
00C1               7 digit_2          rmb  1  ;
00C2               8 digit_3          rmb  1  ;
00C3               9 position_1 rmb  1  ;
00C4               10 position_2      rmb  1  ;
00C5               11 position_3      rmb  1  ;
00C6               12 adc_data1       rmb  1  ;
00C7               13 adc_data2       rmb  1  ;
00C8               14 address         rmb  1  ;
00C9               15 disp_Address    rmb  1  ;
00CA               16 count1          rmb  1  ;
00CB               17 count2          rmb  1  ;
00CC               18 number1         rmb  1  ;
00CD               19 number2         rmb  1  ;
00CE               20 data_Out1       rmb  1  ;
00CF               21 data_Out2       rmb  1  ;
00D0               22 buzzer          rmb  1  ;
00D1               23 debounce rmb  1  ;
                   24  * Pending call storing :From D3 to F6.
                   25
                   26  *memory  area equates
00D2               27 ramstart   equ   $00c0
00D2               28 romstart   equ   $0300
00D2               29 vectors    equ   $07f8
                   30
                   31  *interrupt &reset  vector area
                   32
07F8               33        org  $07f8
07F8      0497     34 timvec   fdb  timer
07FA      0517     35 irqvec   fdb  snooze
07FC      0300     36 swivec   fdb  start
07FE      0300     37 resvec   fdb  start
                          38
                          39
0800                      40  porta    equ  $00
0800                      41  pa7      equ  7
0800                      42  pa6      equ  6
0800                      43  pa5      equ  5
0800                      44  pa4      equ  4
0800                      45  pa3      equ  3
0800                      46  pa2      equ  2
0800                      47  pa1      equ  1
0800                      48  pa0      equ  0
0800                      49  pa7.     equ  $80
0800                      50  pa6.     equ  $40
0800                      51  pa5.     equ  $20
0800                      52  pa4.     equ  $10
0800                      53  pa3.     equ  $08
0800                      54  pa2.     equ  $04
0800                      55  pa1.     equ  $02
0800                      56  pa0.     equ  $01
                   57
0800                      58  portb    equ  $01
                   59
0800                      60  pb5      equ  5
0800                      61  pb4      equ  4
0800                      62  pb3      equ  3
0800                      63  pb2      equ  2
0800                      64  pb1      equ  1
0800                      65  pb0      equ  0
                   66
0800                      67  pb5.     equ  $20
0800                      68  pb4.     equ  $10
0800                      69  pb3.     equ  $08
0800                      70  pb2.     equ  $04
0800                      71  pb1.     equ  $02
0800                      72  pb0.     equ  $01
                   73
                   74
0800                      75  ddra     equ  $04
0800                      76  ddra7    equ  7
0800                      77  ddra6    equ  6
0800                      78  ddra5    equ  5
0800                      79  ddra4    equ  4
0800                      80  ddra3    equ  3
0800                      81  ddra2    equ  2
0800                      82  ddra1    equ  1
0800                      83  ddra0    equ  0
0800                      84  ddra7.   equ  $80
0800                      85  ddra6.   equ  $40
0800                      86  ddra5.   equ  $20
0800                      87  ddra4.   equ  $10
0800                      88  ddra3.   equ  $08
0800                      89  ddra2.   equ  $04
0800                      90  ddra1.   equ  $02
0800                      91  ddra0.   equ  $01
                   92
                   93
0800                      94  ddrb     equ  $05
0800                      95  ddrb5    equ  5
0800                      96  ddrb4    equ  4
0800                      97  ddrb3    equ  3
0800                      98  ddrb2    equ  2
0800                      99  ddrb1    equ  1
0800                      100 ddrb0    equ  0
0800                      101 ddrb5.   equ  $20
0800                      102 ddrb4.   equ  $10
0800                      103 ddrb3.   equ  $08
0800                      104 ddrb2.   equ  $04
0800                      105 ddrb1.   equ  $02
0800                      106 ddrb0.   equ  $01
                   107
                   108
0800                      109 tscr     equ  $08
0800                      110 tof      equ  7
0800                      111 rtif     equ  6
0800                      112 toie     equ  5
0800                      113 rtie     equ  4
0800                      114 tofr     equ  3
0800                      115 rtifr    equ  2
0800                      116 rt1      equ  1
0800                      117 rt0      equ  0
                   118
                   119
0800                      120 tof.     equ  $80
0800                      121 rtif.    equ  $40
0800                      122 toie.    equ  $20
0800                      123 rtie.    equ  $10
0800                      124 tofr.    equ  $08
0800                      125 rtifr.   equ  $04
0800                      126 rt1.     equ  $02
0800                      127 rt0.     equ  $01
                   128
0800                      129 tcr      equ  $09
                   130
```

```
0800                 eprog   equ  $18
0800                 elat    equ  2
0800                 mpgm    equ  1
0800                 epgm    equ  0
0800                 elat.   equ  $04
0800                 mpgm.   equ  $02
0800                 epgm.   equ  $01

0800                 copr    equ  $07f0
0800                 copc    equ  0
0800                 copc.   equ  $01

0800                 mor     equ  $07f1
0800                 cop     equ  0
0800                 copen.  equ  $01

*************************************************
07F1                         org  mor
07F1        01               fcb  $01             ;Watchdog Timer
*************************************************
0300                         org  $0300
* Crystal Frequency is 3.2768MHz.This gives the
  Internal Clock Frequency of
* Crystal Frequency/2 = 1.6384MHz.
* The Timer interrupt can be programmed to give
  interrupt after every 16,384,
* cycles by selecting rt1 and rt2 in timer status
  and control register.
* Here the Timer is programmed to provide an
interrupt after every 10 miliseconds.
* i.e. 16384 cycles.For this option rt0 = rt1 = 0.
*************************************************
0300 [02] 9A         start   cli                  ; clear interrupt
0301 [05] 1808               bset  rtie,tscr      ; Activate the Timer Interrupt.
0303 [05] 1308               bclr  rt1,tscr
0305 [05] 1108               bclr  rt0,tscr
*************************************************
* Initilization :-In initialization; the port pins
  are assigned as input or output
*       as per the circuit connections.
*       Port A pins are used for Display of data
            as well as for  reading the ADC
*       Data.
*       Port B pins are used for controlling the
          ADC and Multiplexer.The Port B
*       pins are connected as:
*       Pb5: Read of ADC; Pb4 : Write of ADC;
  Pb3 : Interrupt from  ADC;
*       Pb2: For Channel Selection of 4051; Pb1
        :Mode Selection;
*       Pb0 : Latch Enable.
*       Keep the display and Buzzer off initially.
*************************************************
0307 [02] A6BF       InitA   lda   #%10111111
0309 [04] B700               sta   porta
030B [02] A6FF               lda   #%11111111
030D [04] B704               sta   ddra       ; Port A O/P
                                    Port, Display Off,Buzzer off

030F [02] A635       InitB   lda   #%110101   ; Pb5,Pb4,Pb2
                                    and Pb0 in O/P Mode.
                                              ; Pb3 and Pb1 in I/P
Mode.
0311 [04] B701               sta   portb
0313 [02] A635               lda   #%110101   ; RD,WR and Latch

    Enable High,
0315 [04] B705               sta   ddrb       ; Latch Transparant
0317 [05] 1101               bclr  Pb0,portb  ; Latch Latched
*************************************************
* Clear : Clear all memory locations.Later the
  locations are used for storing the
* pending calls.
*************************************************
0319 [03] 4F         Clear1  clra             ; Ram claring
031A [05] C707F0             sta   copr       ; Cick WatchDog Timer
031D [02] AEC0       Clear2  ldx   #$c0
031F [04] F7         Clear3  sta   ,x
0320 [03] 5C                 incx
0321 [02] A3FF               cpx   #$ff ; Check all the locations are
                                    cleared?
0323 [03] 25FA               blo   clear3
0325 [04] F7                 sta   ,x
*************************************************
* Assign : Assigns the memory location for storing
  the recent and  pending calls.
*************************************************
0326 [02] A6D3       Assign  lda   #$D3
0328 [04] B7C8               sta   Address
032A [04] B7C9               sta   Disp_Address
*************************************************
* Identify : This part of the programm identifies
  the location of calling point
*        from the adc data. The module is
 divided in two parts.In first part
*        the circuit1(adc_data1) output is
 analysed while in second part the
*        the circuit2(adc_data2) output is analysed.
*        The call point numbers will be stored
   in ram starting from address
*        D3 Hex.
*        ADC data is compared with already
   stored calculated values and
*        accordingly the calling Point destinatio
      n will be  confirmed.
*        The Calling Point destination will be
  confirmed if the data persists
*        100 miliseconds.
*************************************************
032C [03] B6C8       Ident00  lda   Address  ; If Address=F6 indicates
                                              that all the
032E [02] A1F6                cmp   #$F6      ; 36 memory locations are full
0330 [03] 230F                bls   Ident03   ;

0332 [03] B6C9       Ident01  lda   Disp_Address  ; Wait till all the
                                              calling points
0334 [02] A1F6                cmp   #$F6      ; are displayed.
0336 [03] 2203                bhi   Ident02
0338 [03] CC03E5              jmp   Mode00
*************************************************
* When Disp_Address points the memory location
   F7,it indicates that no call
* is pending and the addresss pointers are re-initia
    lised at starting address
* i.e. D3 hex.
*************************************************
033B [02] A6D3       Ident02  lda   #$D3
033D [04] B7C8                sta   Address
033F [04] B7C9                sta   Disp_Address
*************************************************
* Ident03 : Scan circuit1 output.
*************************************************
0341 [03] B6C6       Ident03  lda adc_data1  ; adc_data1 stores circuit1
                                              output.
0343 [02] A107                cmp   #!07
0345 [03] 2206                bhi   Ident05

0347 [05] 3FCA       Ident04  clr   Count1    ; No call  is there.
0349 [05] 3FCC                clr   Number1
034B [03] 2043                bra   Ident20   ; Check other circuit
*************************************************
* Ident05 : Adc data is greater than 07,Check for
    the calling point number.
* The range of data for each calling point is
 stored at memory locations from
* 0700hex to 0712hex.
*************************************************
034D [03] 5F         Ident05  clrx          ; Clear the Register X.Reg X acts as
                                            ; memory pointer.

034E [03] 5C         Ident06  incx

034F [05] D60700     Ident07  lda  $0700,x ; Check  if the ciccuit1 output lies
0352 [03] B1C6                cmp   adc_data1   ; in the range?
0354 [03] 2205                bhi   Ident10     ; Range is found

0356 [02] A311       Ident08  cpx   #!17   ; Is all the ranges are checked?
0358 [03] 25F4                blo   Ident06
035A [03] 5C         Ident09  incx             ; increment the memory pointer
*************************************************
* Ident10 : The range in which the adc_data lies is
   found. Confirm the perticular
* key press if the data persists for 100 miliseconds
* reg Count1 stores the number of scanning times
 for which the same data persists.
* reg number1 temporaly stores the calling point
   number of circuit1.The number
* will be confirmed if the data persists for 100
    milisecond(10 scannings )
*************************************************
035B [03] B6CA       Ident10  lda   Count1    ; Is it  a first key press?
035D [02] A100                cmp   #!00
035F [03] 2606                bne   Ident12

0361 [04] BFCC       Ident11  stx   Number1        ; store
the calling point number temporaly.
0363 [05] 3CCA                inc   Count1
0365 [03] 2029                bra   Ident20
*************************************************
* Ident12 : Check if the Key Press persists for 100
      Miliseconds or not.
* Also check whether it is the  same key press?
*************************************************
0367 [05] 3CCA       Ident12  inc   Count1

0369 [03] B3CC       Ident13  cpx   Number1    ; Check is it  a same
                                              key press?
036B [03] 2706                beq   Ident15    ; Yes,
```

```
036D [05] 3FCC    Ident14  clr   Number1      ; Key Press is
                                                different; start again.
036F [05] 3FCA             clr   Count1
0371 [03] 201D             bra   Ident20      ; Check the  other circuit

0373 [03] B6CA    Ident15  lda   Count1
0375 [02] A10A             cmp   #!10
0377 [03] 2317             bls   Ident20      ; 10 scannings
  are not over,check other circut.
                  ***********************************************
                  *Ident16 : If the Call point number is already
                   stored, Do not accept it again
                  ***********************************************
0379 [02] AED3    Ident16  ldx   #$D3         ; Memory pointer is
                                                initiated at D3hex

037B [03] F6      Ident17  lda   ,x           ; Check the data stored
                                                memory pointed by
037C [03] B1CC             cmp   Number1  ; the memory pointer.
037E [03] 270C             beq   Ident19  ; The call  point is already stored

0380 [03] 5C      Ident18  incx          ; Increment  the Memoty pointer
0381 [02] A3F6    cpx   #$F6             ; Is it a last  memory location?
0383 [03] 23F6             bls   Ident17
                  ***********************************************
                  *  A fresh call is there,store the call point
                         number in ram
                  ***********************************************
0385 [03] B6CC    lda   Number1   ; Number 1 stores the call no.data
0387 [03] BEC8    ldx   Address
0389 [04] F7      sta   ,x
038A [05] 3CC8    inc   Address
                  ***********************************************
                  * Ident19 : Get ready to read new data.
                  ***********************************************
038C [05] 3FCC    Ident19  clr   Number1
038E [05] 3FCA             clr   Count1
                  ***********************************************
                  ***********************************************
                  ***********************************************
                  * Ident20 : Scanning of Circuit2
                  *         : The output of circuit2 is stored in  adc_data2.
                  ***********************************************
0390 [03] B6C7    Ident20  lda   adc_data2
0392 [02] A107             cmp   #!07
0394 [03] 2206             bhi   Ident22

0396 [05] 3FCB    Ident21  clr   Count2
0398 [05] 3FCD             clr   Number2
039A [03] 2049             bra   Mode00       ; Check  other circuit
                  ***********************************************
                  * Ident22 : Adc data is greater than 07,Check for
                     the calling point number.
                  ***********************************************
039C [03] 5F      Ident22  clrx               ; Clear the Register X

039D [03] 5C      Ident23  incx

039E [05] D60700  Ident24  lda   $0700,x
03A1 [03] B1C7             cmp   adc_data2
03A3 [03] 2205             bhi   Ident26

03A5 [02] A311    Ident25  cpx   #!17
03A7 [03] 25F4             blo   Ident23
03A9 [03] 5C               incx
                  ***********************************************
                  * Ident26 : The range in which the adc_data lies is
                   found. Confirm the perticular
                  *      key press if the data persists for 100  miliseconds.
                  *      reg Count1 stores the number of scanning times
                           for which the same data
                  *       persists.
                  *        reg number1 temporaly stores the calling  point
                           number of circuit1.
                  *       The number will be confirmed if the same
                           data persists for 100 milisecond
                  *       (10 scannings )
                  ***********************************************
03AA [03] B6CB    Ident26  lda   Count2       ; Is it  a first key press?
03AC [02] A100             cmp   #!00
03AE [03] 2609             bne   Ident28

03B0 [02] 9F      Ident27  txa   ; Set msb high to indicate circuit2 data
03B1 [02] AA80             ora   #%10000000
03B3 [04] B7CD             sta   Number2
03B5 [05] 3CCB             inc   Count2
03B7 [03] 202C             bra   Mode00
                  ***********************************************
                  * Ident28 : Check if the Keypress persists for 100
                         Miliseconds or not.
                  * Also check whether it is a same key press?
                  ***********************************************
03B9 [05] 3CCB    Ident28  inc   Count2

03BB [02] 9F      Ident29  txa
03BC [02] AA80             ora   #%10000000
03BE [03] B1CD             cmp   Number2
03C0 [03] 2706             beq   Ident31

03C2 [05] 3FCD    Ident30  clr   Number2      ; Not valid  key press
03C4 [05] 3FCB             clr   Count2
03C6 [03] 201D             bra   Mode00       ;

03C8 [03] B6CB    Ident31  lda   Count2
03CA [02] A10A             cmp   #!10
03CC [03] 2317             bls   Mode00  ; 10 scannings are not finished.
                  ***********************************************
                  * Ident32 : If the Call point number is already
                   stored, Do not accept it again
                  ***********************************************
03CE [02] AED3    Ident32  ldx   #$D3

03D0 [03] F6      Ident33  lda   ,x
03D1 [03] B1CD             cmp   Number2
03D3 [03] 270C             beq   Ident36

03D5 [03] 5C      Ident34  incx
03D6 [02] A3F6             cpx   #$F6
03D8 [03] 23F6             bls   Ident33

03DA [03] B6CD    Ident35  lda   Number2      ; Number2 stores the
                                                call no.data
03DC [03] BEC8             ldx   Address
03DE [04] F7               sta   ,x
03DF [05] 3CC8             inc   Address
                  ***********************************************
                  * Ident36 : The number is already stored in
                      memory,Do not repeat it.
                  ***********************************************
03E1 [05] 3FCD    Ident36  clr   Number2
03E3 [05] 3FCB             clr   Count2
                  ***********************************************
                  * Mode : This part of the programme reads the
                   status of the Mode key.Accordingly
                  * the format of the display will be decided.
                  * For Mode 0: The Call Points will be displayed as
                       001 to 018 for circuit1
                  *          and 019 to 036 for circuit2.
                  * For Mode 1: The call points will be displayed as
                     001 to 018  for circuit1 and
                  *   and 101 to 118 for circuit2.
                  * Mode selector switch is connected to pin Pb1 of PortB.
                  ***********************************************
03E5 [05] 02012B  Mode00  brset  Pb1,Portb,Mode07    ; check
                            is it a  mode 1or 2.
                  ***********************************************
                  * Mode1 : Call points will be decided from 001 to  036
                  *      Display the calling point number pointed
                        by register Disp_Address.
                  ***********************************************
03E8 [03] BEC9    Mode01  ldx    Disp_Address
03EA [03] F6              lda    ,x
03EB [02] A100            cmp    #!00         ; Is it 00?
03ED [03] 271C            beq    Mode06
                  ***********************************************
                  * Mode02 : Data conditioning of circuit2 display.(Display
                   001 to 018)
                  ***********************************************
03EF [03] BEC9    Mode02  ldx    Disp_Address
03F1 [03] F6              lda    ,x
03F2 [04] B7CE            sta    Data_Out1

03F4 [05] 0ECE06  Mode03  brset  7,Data_Out1,Mode05     ; Msb
                    of the data decides whether
                                       ; it is circuit1 or circuit2 data
03F7 [05] 3FCF    Mode04  clr    Data_Out2          ;
03F9 [05] 1CD0            bset   6,Buzzer           ;  Buzzer on
03FB [03] 203B            bra    Discon00
                  ***********************************************
                  * Mode05 : Data conditioning of circuit2 display.(Di
                          splay 019 to 036)
                  ***********************************************
03FD [05] 1FCE    Mode05  bclr   7,Data_Out1
03FF [03] B6CE            lda    Data_Out1
0401 [02] AB12            add    #!18     ; Add 18 so that display will
                                            be from 19
0403 [04] B7CE            sta    Data_Out1
0405 [05] 3FCF            clr    Data_Out2
0407 [05] 1CD0            bset   6,Buzzer
0409 [03] 202D            bra    Discon00
                  ***********************************************
                  * Mode06 : The data is 00.It indicates that no key
                          press is found.
                  *      Dispaly 000 and Buzzer off.
                  ***********************************************
040B [05] 3FCE    Mode06  clr    Data_Out1
040D [05] 3FCF            clr    Data_Out2
```

```
040F [05] 3FD0          clr     Buzzer
0411 [03] 2025          bra     Discon00
                  ************************************************
                  * Mode07 : For Mode 1 display.
                  *       : The Call points will be displayed as 001
                        to 018  and 101 to118.
                  ************************************************
0413 [03] BEC9    Mode07    ldx     Disp_Address
0415 [03] F6                lda     ,x
0416 [02] A100              cmp     #!00
0418 [03] 2718              beq     Mode12
                  ************************************************
                  * Mode08 : For 001 to 018
                  ************************************************
041A [03] BEC9    Mode08    ldx     Disp_Address
041C [03] F6                lda     ,x
041D [04] B7CE              sta     Data_Out1

041F [05] 0ECE06  Mode09    brset   7,Data_Out1,Mode11

0422 [05] 3FCF    Mode10    clr     Data_Out2     ; Display will be 001 to
                                                    018
0424 [05] 1CD0              bset    6,Buzzer
0426 [03] 2010              bra     Discon00
                  ************************************************
                  * Mode11 : For 101 to 118
                  ************************************************
0428 [05] 1FCE    Mode11    bclr    7,Data_Out1
042A [02] A601              lda     #!01          ; Display will be 101 to 118.
042C [04] B7CF              sta     Data_Out2
042E [05] 1CD0              bset    6,Buzzer
0430 [03] 2006              bra     Discon00
                  ************************************************
                  * Mode12: No Key press is found; Display 000,Buzzer
                            off.
                  ************************************************
0432 [05] 3FCE    Mode12    clr     Data_Out1
0434 [05] 3FCF              clr     Data_Out2
0436 [05] 3FD0              clr     Buzzer
                  ************************************************
                  * Discon:- This part of the programme gets the BCD
                    equivalant of the hex data.
                  *     The data in all the stages is in hex. For
                   display purpase,the data should
                  *     be in BCD format.
                  * Data_Out1 and Data_Out2 stores the data to be
                        displayed in hex.
                  * Digit_1,Digit_2 and Digit_3 stores the data in
                          BCD format.
                  *     First the hex data is converted to decimal
                   equivalant by adding 06 or its
                  *     multiple ( for 0 to 9 hex add 00, for 0ahex
                   to 13hex add 06, for14hex to
                  *     1d hex add 0c hex and for 1e to 27 hex add 12hex.)
                  ************************************************
0438 [03] B6CE    Discon00  lda     Data_Out1     ;
043A [05] 3FC0              clr     Digit_1

043C [02] A00A    Discon01  sub     #$0a          ; Substaract 10 decimal
043E [03] 2504              bcs     Discon02
0440 [05] 3CC0              inc     digit_1
0442 [03] 20F8              bra     Discon01

0444 [03] B6C0    Discon02  lda     digit_1       ; Get the multiple of 6
0446 [02] AE06              ldx     #$06
0448 [11] 42                mul
0449 [03] BBCE              add     Data_Out1
044B [04] B7CE              sta     Data_Out1     ; equivalant  of hex in decimal.

044D [05] 3FC0    Discon03  clr     Digit_1
044F [05] 3FC1              clr     Digit_2
0451 [05] 3FC2              clr     Digit_3
                  ************************************************
                  * Discon04 : Convert the decimal to bcd one.
                  ************************************************
0453 [03] B6CE    Discon04  lda     Data_Out1
0455 [02] A40F              and     #%00001111
0457 [04] B7C0              sta     digit_1       ; bcd equivalant of lsb of
                                                    Data_Out1

0459 [03] B6CE    Discon05  lda     Data_out1
045B [02] A4F0              and     #%11110000
045D [03] 44                lsra
045E [03] 44                lsra
045F [03] 44                lsra
0460 [03] 44                lsra
0461 [04] B7C1              sta     digit_2       ; bcd equivalant of Msb of
                                                    Data_Out1

0463 [03] B6CF    Discon06  lda     Data_out2
0465 [02] A40F              and     #%00001111
0467 [04] B7C2              sta     digit_3       ; bcd equivalant
                        of lsb of Data_Out1
                  ************************************************
                  *Discon07 : Get the Display equivalant of each digit.
                  * The bcd of each digit is fed to the BCD to Seven
                      segement convertor 7447.
                  * The display equivalant( as per the hard-ware
                  arrengement ) is stored from 7c0 hex
                  * onwords.
                  * This part of the program gets the display
                    equivalant of each bcd number.
                  * Position_1,Position_2 and Position_3 stores the
                      data to be displayed.
                  ************************************************
0469 [03] B6C0    Discon07  lda     digit_1
046B [02] A40F              and     #%00001111
046D [02] 97                tax
046E [05] D607C0            lda     $07c0,x
0471 [02] AA06              ora     #%00000110
0473 [03] BAD0              ora     Buzzer
0475 [04] B7C3              sta     position_1    ; Digit1 data
                  ************************************************
0477 [03] B6C1    Discon08  lda     digit_2
0479 [02] A40F              and     #%00001111
047B [02] 97                tax
047C [05] D607C0            lda     $07c0,x
047F [02] AA05              ora     #%00000101
0481 [03] BAD0              ora     Buzzer
0483 [04] B7C4              sta     position_2    ; Digit2 data
                  ************************************************
0485 [03] B6C2    Discon09  lda     digit_3
0487 [02] A40F              and     #%00001111
0489 [02] 97                tax
048A [05] D607C0            lda     $07c0,x
048D [02] AA03              ora     #%00000011
048F [03] BAD0              ora     Buzzer
0491 [04] B7C5              sta     position_3    ; Digit3 data
                  ************************************************
                  * Wait :- As scanning is done after 10 miliseconds,
                    Controller is in low power mode
                  *       till fresh data is available.
                  ************************************************
0493 [02] 8F      Wait      wait
0494 [03] CC032C            jmp     Ident00
                  ************************************************
                  * Timer :- This is a Timer interrupt service
                  routine.The 16 bit internal Timer of
                  *     the Microcontroller is software programmed
                    to give interrupt after every
                  *     10 miliseconds.During the Timer interrupt
                    service routine two tasks are
                  *     completed.
                  *     i) Refreshing of multiplexed displays.
                  *     As the it very essential to refresh the
                   multiplexed display at a frequency
                  *     of 50Hz or more;during this interrupt
                  routine displays will be refreshed.
                  *     This gives a refreshing frequency of 100Hz.
                  *     ii) Scanning of Calling Points.
                  *     Both the circuits are scanned and the
                          digital equivalant of output voltages
                  *     will be stored in two registers.
                  ************************************************
0497 [05] 1408    Timer     bset    rtifr,tscr
0499 [03] 4F                clra
049A [05] C707F0            sta     Copr          ; kick the watchdog timer
                  ************************************************
                  * Timer01 : Take care of the debounce time.
                  ************************************************
049D [03] B6D1    Timer01   lda     Debounce
049F [02] A164              cmp     #!100
04A1 [03] 2404              bhs     Timer03

04A3 [05] 3CD1    Timer02   inc     Debounce
04A5 [03] 2004              bra     Disp00
04A7 [02] A665    Timer03   lda     #!101
04A9 [04] B7D1              sta     Debounce
                  ************************************************
                  * Refreshing of Displays.
                  ************************************************
04AB [02] A6BF    Disp00    lda     #%10111111
04AD [03] BAD0              ora     Buzzer
04AF [04] B700              sta     Porta
04B1 [02] A6FF              lda     #$ff
04B3 [04] B704              sta     ddra          ; Assign  Porta in Output mode.
04B5 [05] 1001              bset    Pb0,portb     ; Make the
                            Latch transparant.
                  ************************************************
                  * Disp01 : Refresh the digit1.Keep the digit1 on
                          for 1 milisecond.
                  ************************************************
04B7 [03] B6C3    Disp01    lda     position_1    ; Digit 1 Display
```

```
04B9 [04] B700              sta     porta
04BB [06] CD052B    Disp02  jsr     Delay
                    ****************************************************
                    * Disp03 : Refresh the digit2.Keep the digit2 on
                              for 1 milisecond.
                    ****************************************************
04BE [03] B6C4      Disp03  lda     position_2      ; Digit 2 Display
04C0 [04] B700              sta     porta
04C2 [06] CD052B    Disp04  jsr     Delay
                    ****************************************************
                    * Disp06 : Refresh the digit3.Keep the digit3 on
                              for 1 milisecond.
                    ****************************************************
04C5 [03] B6C5      Disp06  lda     position_3
04C7 [04] B700              sta     porta
04C9 [06] CD052B    Disp07  jsr     Delay
                    ****************************************************
                    * Disp08: Refreshing is over. Switch of all the
                      digits to save the power.Also Make
                    *       the latch Non Transperant so any changes
                      on the Port A bus will not change
                    *       the status of displays.
                    ****************************************************
04CC [02] A6BF      Disp08  lda     #%10111111
04CE [03] BAD0              ora     Buzzer
04D0 [04] B700              sta     Porta
04D2 [05] 1101              bclr    Pb0,portb   ; Latch Non Transparant
                    ****************************************************
                    * Adc: Scanning of the Calling Points.
                    *     ADC is used for reading the output voltages
                          of Circuit 1 and Circuit2.
                    *Multiplexer 4051 is used for selecting the circuit 1 or 2.
                    *     While reading the Adc data,Port A is
                    assigned as input port. Port B pins are
                    *     used for providing the control signals. End
                          of Conversion is indicated by
                    *     Intr signal.
                    *     The digital equivalants of circuit1 and 2
                      are stored in registers adc_data1
                    *     and adc_data2.
                    *     At the end of conversion,the Port A is
                          assigned as output port again.
                    ****************************************************
                    * Adc00 : Reading of Circuit1 output.
                    ****************************************************
04D4 [02] A600      Adc00   lda     #$00
04D6 [04] B704              sta     ddra        ; Take Port  A in Input mode.
04D8 [05] 1901              bclr    pb4,portb   ; Ensure Write signal to low

04DA [06] CD0533    Adc01   jsr     Delay2      ; Keep it low.
04DD [05] 1801      Adc02   bset    pb4,portb; Write high,Conversion starts.

04DF [03] B601      Adc03   lda     Portb       ; Wait for End of
                                                  conversion.Intr signal
                                                ; goes low at the end of conversion.
04E1 [02] A408              and     #%00001000
04E3 [02] A100              cmp     #%00000000
04E5 [03] 26F8              bne     Adc03

04E7 [05] 1B01      Adc04   bclr    pb5,portb           ; Read low

04E9 [06] CD0533    Adc05   jsr     Delay2

04EC [03] B600      Adc06   lda     porta           ; Data is available on
                                                      data bus of adc.
04EE [04] B7C6              sta     adc_data1       ; ADC data  is stored

04F0 [05] 1A01      Adc07   bset    pb5,portb       ; Read high
                    ****************************************************
                    * Adc08 : Reading of Circuit2 output.
                    ****************************************************
04F2 [05] 1501      Adc08   bclr    Pb2,Portb   ; Select circuit2  using
                                                  multiplexer.
04F4 [05] 1901              bclr    pb4,portb           ; Write low

04F6 [06] CD052B    Adc09   jsr     Delay
04F9 [05] 1801      Adc10   bset    pb4,portb       ; Write high, Conversion
                                                      starts.

04FB [03] B601      Adc11   lda     Portb       ; Read Intr signal from adc
04FD [02] A408              and     #%00001000
04FF [02] A100              cmp     #%00000000
0501 [03] 26F8              bne     Adc11

0503 [05] 1B01      Adc12   bclr    pb5,portb           ; Read low

0505 [06] CD0533    Adc13   jsr     Delay2
0508 [03] B600      Adc14   lda     porta           ; Data is available on
                                                      data bus.
050A [04] B7C7              sta     adc_data2       ; ADC data is stored

050C [05] 1A01      Adc15   bset    pb5,portb           ; Read high
050E [05] 1401              bset    Pb2,Portb       ; Circuit 1
                                                      Selected for next data read.

0510 [02] A6FF      Adc16   lda     #$ff            ; Port A in Input mode.
0512 [04] B700              sta     porta
0514 [04] B704              sta     ddra
0516 [09] 80                rti
                    ****************************************************
                    * Snooze:Here the Buzzer can be snoozed by receiving
                      the call.A Call Acknowledge key
                    * is used for this purpose. The Call Acknowledge
                      key generates the Interrupt
                    * request.Following Interrupt service routine
                      snoozes as well as displays any
                    * pending call.A key debounce time of 1 second is
                      provided.
                    ****************************************************
0517 [05] 0DD010    snooze  brclr   6,Buzzer,snooze4

051A [03] B6D1      snooze1 lda     Debounce
051C [02] A164              cmp     #!100
051E [03] 230A              bls     snooze4         ; Wait for debounce period
                                                      of 1 second.

0520 [03] BEC9      snooze2 ldx     Disp_Address
0522 [03] 4F                clra
0523 [04] F7                sta     ,x
0524 [05] 3CC9              inc     Disp_Address    ; Display the pending call

0526 [05] 3FD0      snooze3 clr     Buzzer          ; Snooze the buzzer.
0528 [05] 3FD1              clr     Debounce

052A [09] 80        snooze4 rti
                    ****************************************************
                    * Delay : Provides delay of
                            ********************************************
********
052B [02] AEFA      Delay   ldx     #!250

052D [03] 5A        Delay1  decx
052E [02] A300              cpx     #$00
0530 [03] 26FB              bne     Delay1
0532 [06] 81                rts
                            ********************************************
********
                    * Delay2: Provides delay of
                    ****************************************************
0533 [02] AE96      Delay2  ldx     #!150

0535 [03] 5A        Delay3  decx
0536 [02] A300              cpx     #$00
0538 [03] 26FB              bne     Delay3
053A [06] 81                rts
                            ********************************************
********

0701                                        org $0701
0701     0F1A2632           fcb  !15,!26,!38,!50,!62,!72,!82,!92,!103,!1
                                 14
          3E48525C
          6772
070B 7D89939C               fcb  !125,!137,!147,!156,!170,!188,!200
          AABCC8
                    ****************************************************
07C0                                        org $07c0
07C0          00801090      fcb  $00,$80,$10,$90,$08,$88,$18,$98,$20,$a0
              08881898
              20A0
```

AUTOMATIC WATER-LEVEL CONTROLLER

NIZAR P.I.

Here's an automatic water-level controller for overhead tanks. It uses an infrared (IR) transmitter and a receiver to control the operation of the centrifugal water pump. The pump controller circuit is built around dual-timer IC NE556 and NAND gate CD4011. IC NE556 contains equivalent of two NE555 timers. The IR transmitter transmits 38kHz signals and relay driver transistor SL100 controls the motor operation.

The system

The automatic water-level controller comprises an electronic circuitry and a mechanical capillary arrangement.

Electronic circuitry. Fig. 1 shows the circuit of automatic water-level controller. The components used in this circuit are low-cost and readily available in the market. The power supply is built around a 9V-0-9V, 500mA step-down transformer (X1), rectifier comprising diodes D1 and D2, and a filter capacitor (C1). The 6V regulator provides regulated supply to the circuit.

Both the timers of NE556 (IC2) are used in the monostable mode. Trigger input pins 6 and 8 of IC2 are connected to output pins F and E of sensors RX1 and RX2, respectively. (The capillary tube with sensors arrangement is shown in Fig. 3). Output pins 5 and 9 of IC2 are connected to the inputs of NAND gates N3 and N4. The outputs of NAND gates N3 and N4 are further connected to the RS

PARTS LIST

Semiconductors:	
IC1	- 7806 +6V regulator
IC2	- NE556 dual timer
IC3	- CD4011 quad 2-input NAND gate
IC4	- NE555 timer
D1, D2, D5	- 1N4001 rectifier diode
D3, D4	- 1N4148 diode
LED1, LED2	- Infrared transmitter LED
RX1-RX2	- Infrared receiver module (TSOP1738)
T1	- SL100 npn transistor
Resistors (all ¼-watt, ±5% carbon, unless stated otherwise):	
R1, R2, R7, R10	- 100-ohm
R3, R4	- 33-kilo-ohm
R5, R6, R11	- 1-mega-ohm
R12	- 4.7-kilo-ohm
VR1	- 10-kilo-ohm preset
Capacitors:	
C1	- 1000µF, 25V electrolytic
C2-C8	- 0.1µF ceramic disk
C9, C10	- 4.7µF, 16V electrolytic
C11, C12	- 10µF, 16V electrolytic
C13	- 100µF, 16V electrolytic
C14	- 0.001µF ceramic disk
C15	- 0.01µF ceramic disk
Miscellaneous:	
S1	- Push-to-on tactile switch
X1	- 230V AC primary to 9V-0-9V, 500mA secondary transformer
RL1, RL2	- 6V, 100Ω, 1C/O relay
	- Light-weight opaque float
	- Transparent tube for capillary

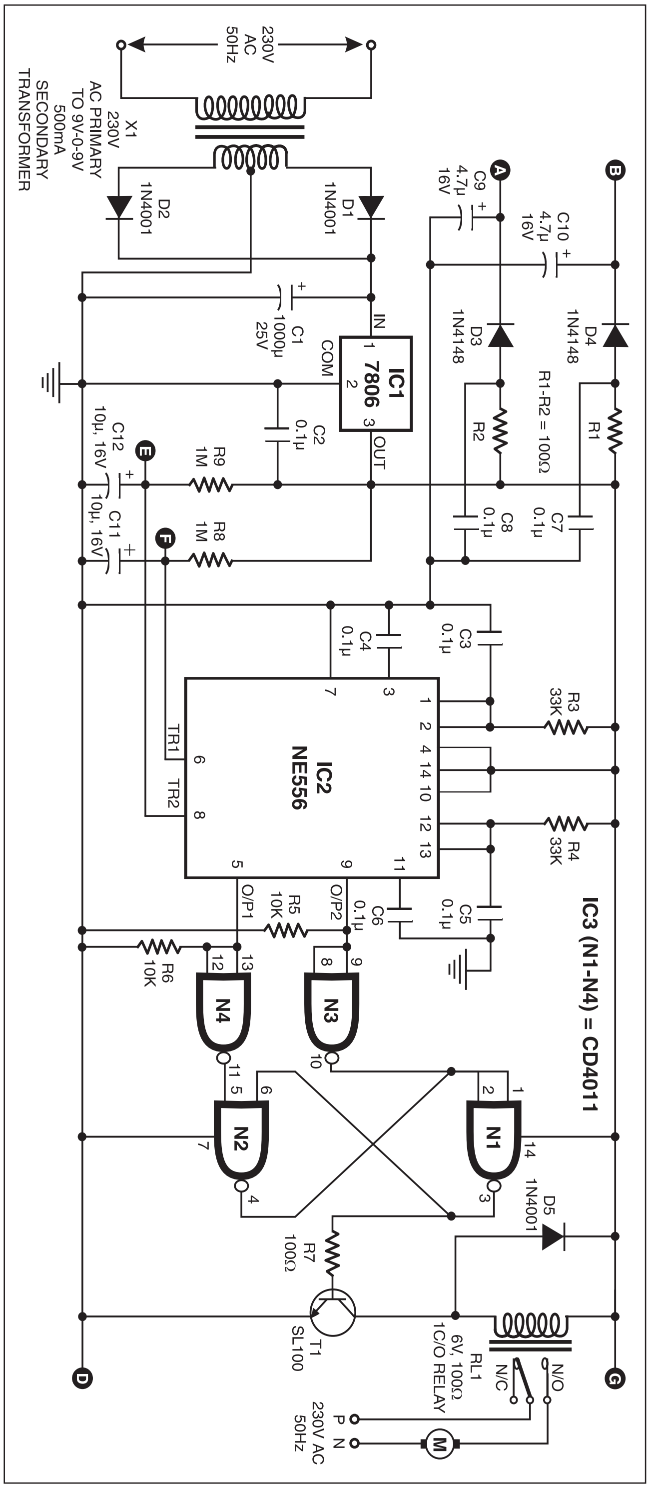


Fig. 1: Circuit diagram of water-level controller

flip-flop built around NAND gates N1 and N2.

Power supply terminals A and B are connected to +Vcc pins of RX1 and RX2 sensors (marked A and B), respectively. If you don't want to use a 9V battery for the transmitter circuit, connect points G and D in the pump controller circuit to the respective points (G and D) in the transmitter circuit.

Fig. 2 shows the transmitter circuit built around timer NE555. Timer NE555 (IC4) is wired as an astable multivibrator producing a frequency of about 38 kHz. When switch S1 is pressed, the circuit gets supply and the two infrared transmitter LEDs connected at the output of IC4 transmit IR beams at a frequency of 38 kHz.

Mechanical capillary tube arrangement. The capillary tube arrangement with sensors is shown in Fig. 3. IR transmitter LED1 and IR receiver sensor RX1 are connected face to face both on the top and the bottom of the capillary tube. Using an adhesive, fix IR receiver modules (TSOP1738) such that their front side is oriented towards IR transmitters. A very-light-weight float made of an opaque material is placed into the transparent capillary tube. It moves along the tube, depending on the level of water, crossing IR beams from the top-level and bottom-level sensors on reaching the top and bottom level limits. The capillary tube can be made of glass or any transparent material. The sensor arrangement for the overhead tank is shown in Fig. 4.

Working of the system

The two pairs of the IR LEDs and the IR receiver modules are used for the minimum (empty) and the maximum (full) water level positions in the tanks.

When the moving object is at level A, the motor is switched on. At B and C levels also, the motor remains on and water continues to fill the tank. When the float crosses the upper IR beam to reach level D, the motor turns off, as the tank is full, and water supply to the tank stops.

As the water is consumed, its level in the tank falls from D to E, C and then to B. At these levels also, the motor remains 'off.' However, when the object crosses the lower IR beam to reach level A, the system recognises that the tank is almost empty, or water is at the minimum level, and the motor turns on again.

In case of power failure, if the object was at D, E, C or B level at the time of power failure, the motor will not start on power resumption. If it was at or below level A, the motor starts on power resumption and starts filling the tank until the float reaches level D.

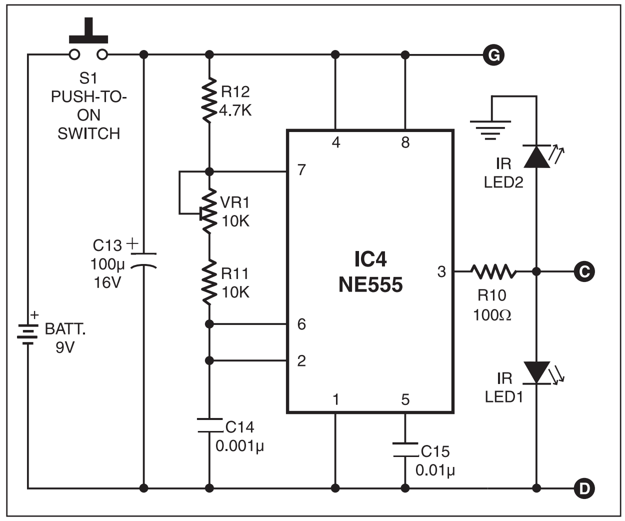


Fig. 2: Transmitter circuit

Overhead tank and reservoir automation

In many houses, water is first stored in a reservoir at or near ground level and from there it is pumped up to the overhead tank on the rooftop. People generally switch on the pump when their taps go dry and switch it off when the overhead tank starts overflowing. In case the reservoir is empty and the motor is switched on, it may damage the motor.

The complete arrangement for the overhead tank and the reservoir for automatic operation is shown in Fig. 5. It comprises two similar arrangements of the pump controller circuit, transmitter circuit and capillary tube assembly: one for the overhead tank and the other for the reservoir. In the capillary tube arrangements, 'M' represents the top-level sensing unit and 'N' the bottom-level sensing unit for the overhead tank. The connections of relays RL1 and RL2 to the pump are not identical. This arrangement prevents the motor from working when the reservoir is empty. The control circuit 2 recognises whether water is at the minimum level of the reservoir or not.

When the reservoir is empty, the float crosses sensor N to interrupt the IR beams emanating from it, which triggers IC2 at its pin 8. The triggering of IC2 makes its output pin 9 high, which energises the relay (RL2) via IC3

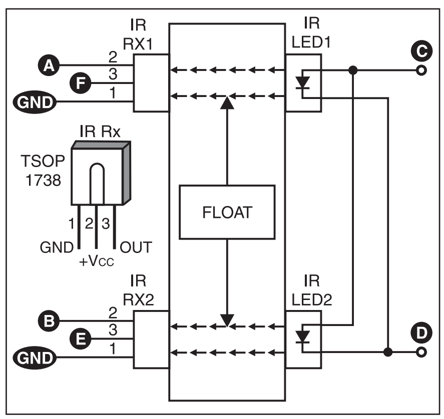


Fig. 3: Capillary tube arrangement with sensors

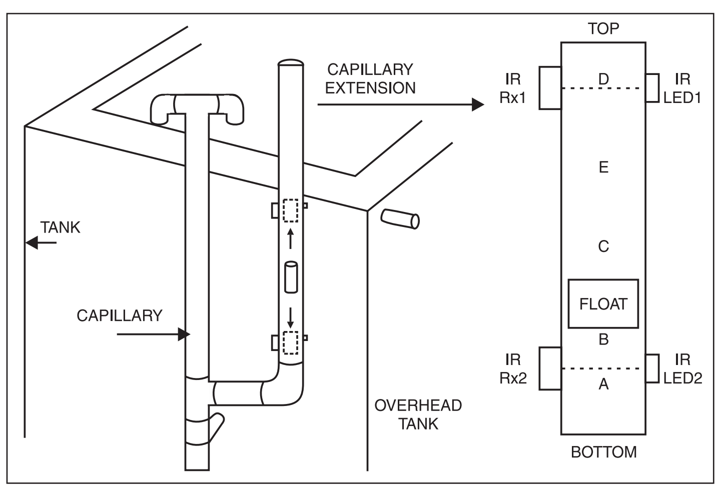


Fig. 4: Placing of sensors in the overhead tank

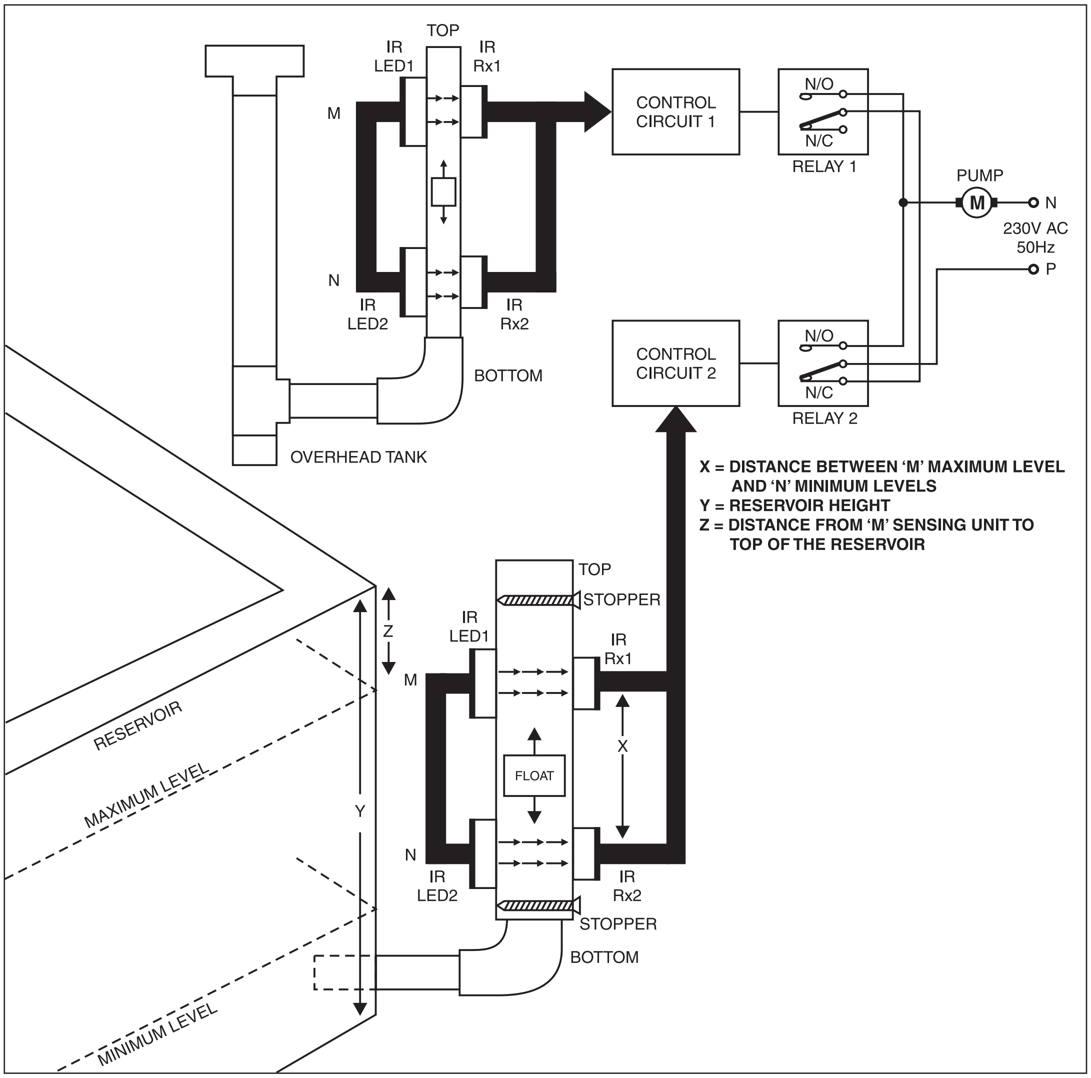


Fig. 5: The complete arrangement for the overhead tank and the reservoir for automatic operation

and driver transistor SL100. Now the motor starts to fill the tank up to the maximum level.

When the reservoir is full, the object crosses sensor M to interrupt the IR beams emanating from it, which triggers IC2 at its pin 6. The triggering of IC2 makes its output pin 5 high, which de-energises the relay (RL2) via IC3 and the driver transistor. Now the motor turns off and relay RL2 provides mains supply to relay RL1 connected to the control circuit 1 for the overhead tank.

X and Z distances (refer Fig. 5) in the sensor assembly depend on the height Y of the tank/reservoir. The distance X should not be below 20 cm. Otherwise, the IR beams from one sensor may interfere with IR

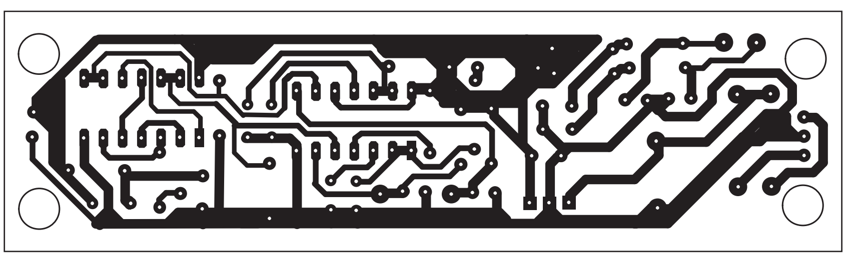


Fig. 6: Actual-size single side PCB layout for the circuit in Fig. 1

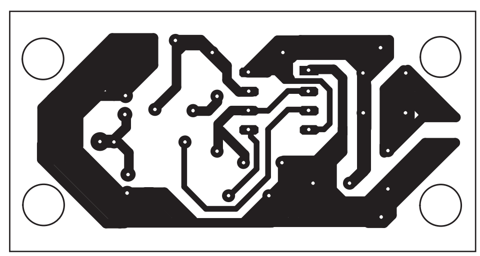


Fig. 8: Actual-size, single side PCB layout for the circuit in Fig. 2

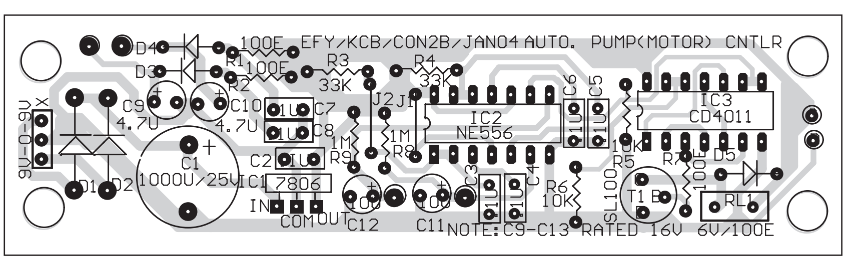


Fig. 7: Component layout for the PCB of Fig. 6

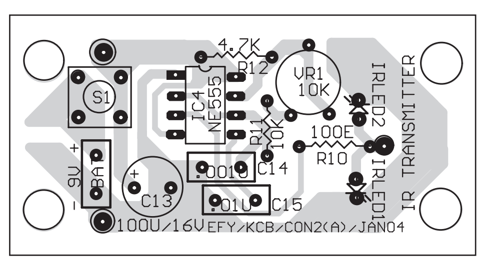


Fig. 9: Component layout for the PCB of Fig. 8

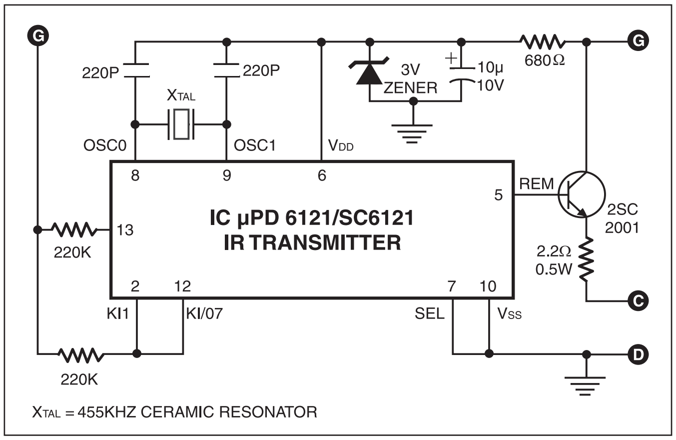


Fig. 10: Optional IR transmitter circuit

beams from the other sensor, affecting the sensing operation.

The complete arrangement is shockproof. The electronic circuit should be encapsulated in a plastic case, while the sensor arrangement should be housed in a PVC pipe. To protect the motor from high or low voltage, use a low-/high-voltage cut-off circuit.

An actual-size, single-side PCB for the circuits in Fig. 1 and Fig. 2 is shown in Fig. 6 and Fig. 7 with its comp-onent layout in Fig. 8 and Fig. 9, respectively.

Fig. 10 shows an optional IR transmitter circuit that is built around IC µPD6121. It transmits modulated pulses with carrier frequency of 38 kHz. Simply replace the transmitter given in Fig. 2 with this circuit and connect its points G, C and D to respective points of pump controller circuit. ❑

DIGITAL WATER-LEVEL INDICATOR CUM PUMP CONTROLLER

PARMAR LATESH B.

Many circuits of water-level controller have appeared in EFY. What sets this circuit apart from all of them is that it shows the level of water far away from the location of the overhead tank. Its other features include:

1. Up to five levels of water are indicated on LED display along with beep sound.
2. DTMF receiver section controls the on/off function of the motor.
3. No battery is required to store the water level when power fails.
4. The water-level scanning section scans the water level with beep sound after power resumes.
5. When water reaches the full level, the motor turns off and provides a beep sound for about a minute.
6. When water goes below the empty level, the motor starts with beep sound.

Fig. 1 shows the remote water-level sensing and DTMF transmitter circuit. At the heart of the circuit is NAND gate CD4093 with resistor-capacitor combination and diode network that senses the water level in the overhead tank. Water inside the tank is divided into five levels, namely, Empty, 1/4th, Half, 3/4th and Full. The DTMF codes used to indicate Empty, 1/4th, Half, 3/4th and Full levels are 1, 2, 3, 4 and 5, respectively. Different levels are indicated by different colour LEDs at the DTMF receiver end.

Suppose water level goes below Empty mark. Transistor T1 stops conducting and the output of NAND gate N1 goes low through resistor R1, capacitor C1 and diode D1. At the same time, the scanning output of NAND gate N12 also goes low. So trigger pin 6 of dual-timer NE556 (IC5), which is wired as a monostable, goes low to drive its output pin 5 high. As a result, column C1 and row R1 of DTMF dialler UM91214B (IC 10) short through analogue switch CD4066 (IC8) and dial the number corresponding to the Empty level. The DTMF output at pin 7 of IC10 is transmitted through wire link to the receiver (Fig. 2). The output of dialer is connected to DTMF decoder CM8870 (IC13) to decode the received signal. The decoded output sets flip-flop CD4013 (IC15) through BCD to decimal decoder IC14 to switch on the motor with LED indication and beep sound.

As water goes up and touches different level-sensing probes, NAND gates N1, N3, N5, N7 and N9 go low one by one and the corresponding differentiator networks activate to trigger IC5 through IC7, respectively, to produce a high output and transmit the corresponding DTMF code by dialer IC10. DTMF codes are transmitted one by one as the water level goes up and touches the different sensing probes. The sensor probes should be made of stainless steel to avoid corrosion.

Timers IC5 through IC7 are wired in monostable mode. The output of the monostable goes high for about 2.4 seconds when its trigger pin goes low.

As water is consumed, its level in the tank falls below different sensor probes and the outputs of NAND gates N2, N4, N6, N8 and N10, with resistor-capacitor combination and diode network, go low one by one. This low output is applied to the trigger pin of dual-timer IC5 through IC7 and the DTMF code corresponding to the level is generated by IC10. The output of the corresponding toggle flip-flop in the receiver section goes low to turn off the related LED, which indicates that water level is below that particular level.

The main purpose behind adding the level-scanning section is to avoid malfunctioning of the receiver section due to power failure (as no battery is added to the receiver to latch the present level of the water). In case we add a battery and the power fails, the water level is latched but during this period if the water level goes below any probe, there is no way to transmit the signal from the transmitter. This leaves us with no other option but to add the level-scanning section.

When power resumes, the level-scanning section scans and checks all the levels one by one (from Empty to Full) and transmits the corresponding codes to the receiver to show the water level in the overhead tank. So when the power resumes, the output of NAND gate N11 goes low after a delay of about 7 seconds, which is set by the combination of resistor R36 and capacitor C27, and trigger pin 6 of dual-timer IC11 goes low.

One timer of IC11 is used as a monostable whose output pin 5 remains high for about 23 seconds. Since pin 5 is directly connected to reset (pin 10) of the other timer of IC11, it is also activated for 23 seconds to generate the pulse. The second timer of IC11 is wired as an astable multivibrator to generate 1-second 'on' time and 3-second 'off' time signals at its pin 9. The 1-second pulse is fed to clock pin 14 of CD4017 (IC12), which scans one of the two inputs of NAND gates N12 through N16 each one by one. The other inputs of these NAND gates are connected to the cathodes of diodes D1 through D5 from the corresponding water level. The outputs of NAND gates N12 through N16 are connected, via diodes D19 through D23 and related resistor-capacitor networks, to the cathodes of diodes D6 through D10, respectively. As a result, trigger pins of IC5 through IC7 go low one by one and the outputs of corresponding timer sections go high, which shorts the related columns and rows of DTMF tone generator IC10 through analogue switch CD4066.

Fig. 2 shows the details of receiver and level indicator circuit. In the receiver section DTMF decoder CM8870 (IC13) is used to decode the received tone signal. This

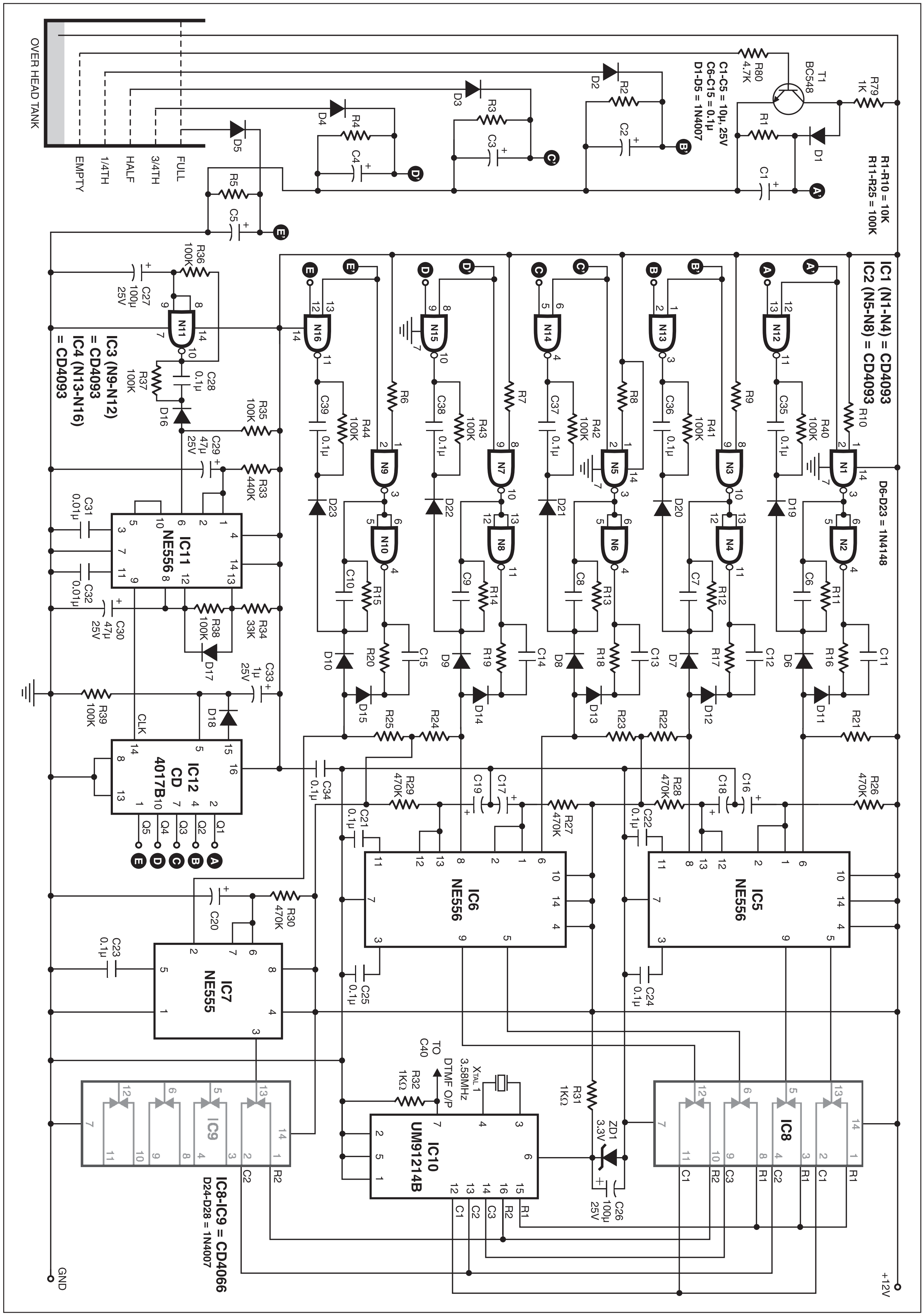


Fig. 1: Remote water level sensing and DTMF transmitter circuit

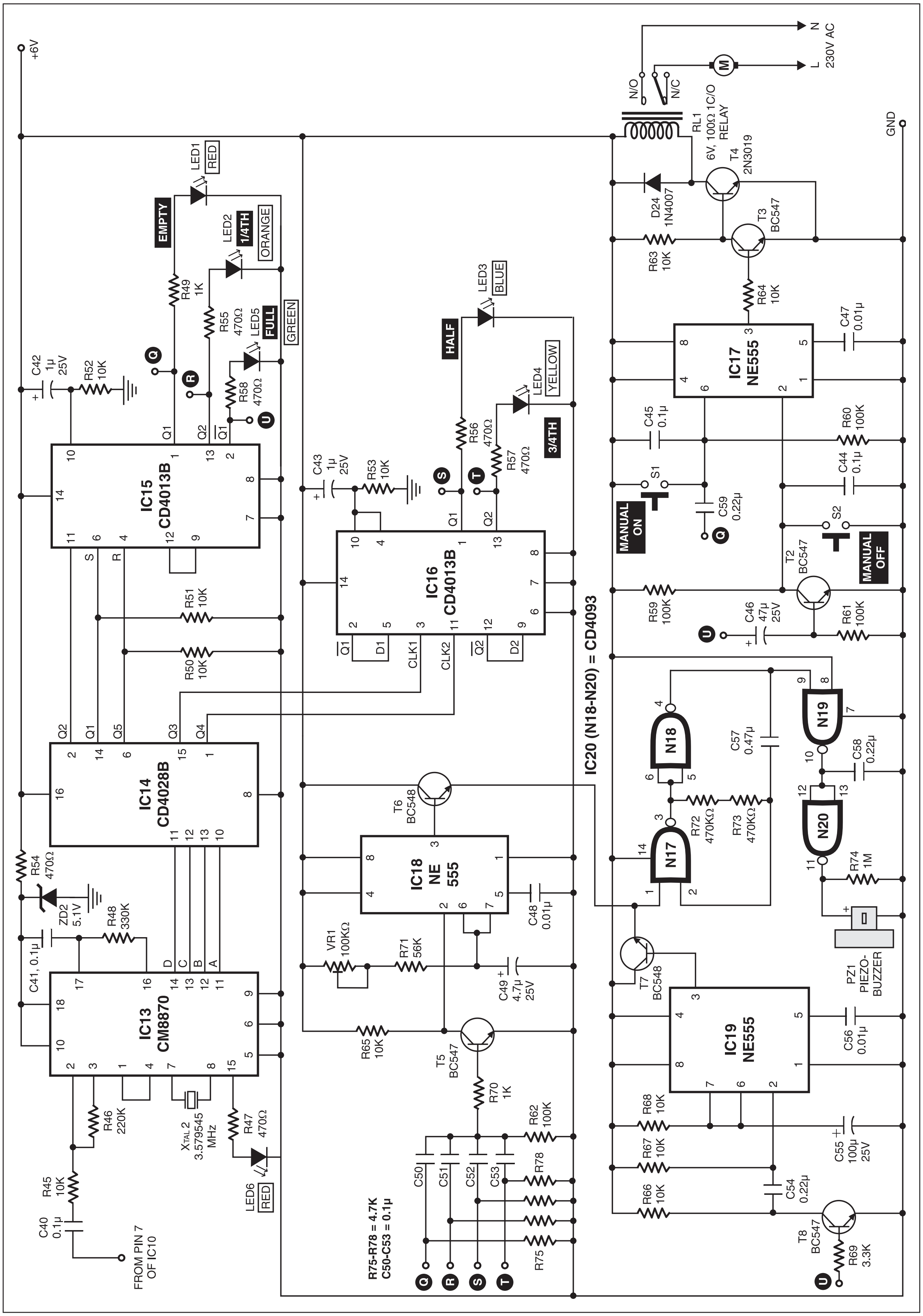


Fig. 2: Receiver and level indicator circuit

PARTS LIST

Semiconductors:

IC1-IC4, IC20	- CD4093 quad NAND gate
IC5, IC6, IC11	- NE556 dual timer
IC7, IC17-IC19	- NE555 timer
IC8, IC9	- CD4066 quad analogue switch
IC10	- UM91214B DTMF tone generator
IC12	- CD4017 decade counter
IC13	- CM8870 DTMF decoder
IC14	- CD4028 BCD-to-decimal decoder
IC15, IC16	- CD4013 dual D-type flip-flop
IC21	- 7812 12V regulator
IC22	- 7806 6V regulator
T1, T6, T7	- BC548 npn transistor
T2, T3, T8	- BC547 npn transistor
T4	- 2N3019 npn transistor
D1-D5, D24-D28	- 1N4007 rectifier diode
D6-D23	- 1N4148 switching diode
ZD1	- 3.3V, 0.5W zener diode
ZD2	- 5.1V, 0.5W zener diode
LED1, LED6	- Red LED
LED2	- Orange LED
LED3	- Blue LED
LED4	- Yellow LED
LED5	- Green LED

Resistors (all ¼-watt, ±5% carbon, unless stated otherwise):

R1-R10, R45, R50-R53, R63-R68	- 10-kilo-ohm
R11-R25, R35-R44, R59-R62	- 100-kilo-ohm
R26-R30, R72, R73	- 470-kilo-ohm
R31, R32, R49, R70, R79	- 1-kilo-ohm
R33	- 440-kilo-ohm
R34	- 33-kilo-ohm
R46	- 220-kilo-ohm
R47, R54-R58	- 470-ohm
R48	- 330-kilo-ohm
R69	- 3.3-kilo-ohm
R71	- 56-kilo-ohm
R74	- 1-mega-ohm
R75-R78, R80	- 4.7-kilo-ohm
VR1	- 100-kilo-ohm preset

Capacitors:

C1-C5	- 10µF, 25V electrolytic
C6-C15, C28, C34-C41, C44, C45, C50-C53, C61-C66	- 0.1µF ceramic disk
C16-C20, C49	- 4.7µF, 25V electrolytic
C21-C25, C31, C32, C47, C48, C56	- 0.01µF ceramic disk
C26, C27, C55	- 100µF, 25V electrolytic
C29, C30, C46	- 47µF, 25V electrolytic
C33, C42, C43	- 1µF, 25V electrolytic
C54, C58, C59	- 0.22µF ceramic disk
C57	- 0.47µF ceramic disk
C60	- 1000µF, 25V electrolytic

Miscellaneous:

X1	- 230V AC primary to 7.5V-0-7.5V, 1A secondary transformer
X_{TAL1}, X_{TAL2}	- 3.578MHz crystal
RL1	- 6V, 1C/O relay
PZ1	- Piezobuzzer
S1, S2	- Push-to-on switch

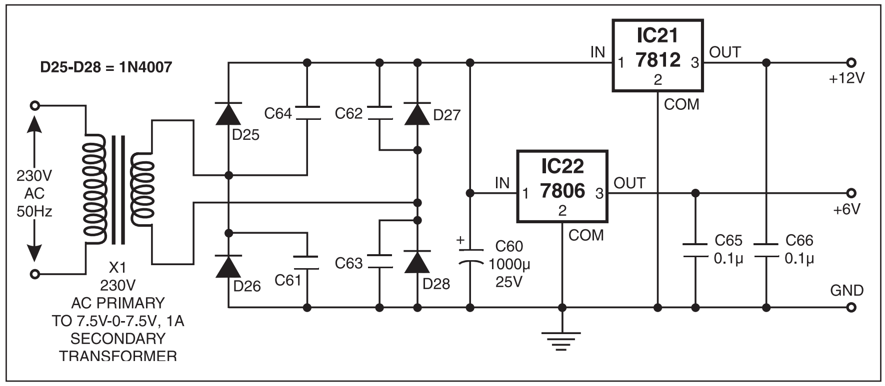


Fig. 3: Power supply

IC converts the received DTMF code into equivalent binary form. BCD-to-decimal decoder CD4028B (IC14) converts this binary code into decimal. Its Q1 through Q5 outputs are connected to 'D' flip-flop CD4013 to control the motor and indicate water level in the overhead tank through the LED. The present water level in the tank is indicated by glowing of the respective LED. When the LED goes off, it means water in the tank is below the indicated level.

Initially, when the power is switched on or the power resumes, all flip-flops of CD4013B (IC15 and IC16), except one (whose reset pin 4 is connected to pin 6 of IC14), are reset through the resistor-capacitor network at pins 4 and 10 of the two ICs. Pins 1 and 2 of IC15 are connected to pin 6 of IC17 via capacitor C59 and the base of transistor T2 via capacitor C46, respectively, to control the motor. The remaining flip-flop of IC15 is wired in set/reset mode.

When water goes below the Empty level, the set input of IC15 (as per the received signal) goes high to make outputs Q1 and Q1 high and low, respectively. The high Q1 output of IC15 energises relay RL1 and the motor is switched on automatically with the help of IC17 and transistors T3 and T4; the motor is connected through the contacts of relay RL1.

For manually switching on the motor, press switch S1.

When water level touches the 'Full' probe, the reset input (as per the received signal) of IC15 goes high to make Q1 and Q1 outputs low and high, respectively. The high Q1 output of IC15 de-energises relay RL1 and the motor turns off automatically with the help of IC17 and transistors T3 and T4. You can also manually switch off the motor by pressing switch S2.

The motor-off state is indicated by a one-minute beep sound. NE555 (IC17) is wired as a bistable multivibrator. When pin 2 of IC17 goes low, its output goes high to drive transistor T3 and transistor T4 de-energises relay RL1. When pin 6 of IC17 gets a high pulse, its output goes low and transistor T3 doesn't conduct while transistor T4 conducts to energise relay RL1. The motor on/off (for empty/full tank) is indicated by the respective LED.

The beeper section generates beep each time the LED indicates a new water level. At the output of flip-flop CD4013B, points Q, R, S, T and U for Empty, 1/4th, Half, 3/4th and Full level indications are connected to capacitors C50 through C53 and R67 at the base of transistor T8, respectively.

When the new water level is latched, the corresponding output of the flip-flop goes high and the LED lights up. At the same time, a small spike is passed to saturate transistor T5 to trigger pin 2 of monostable IC18. The output of monostable goes high for about 500 ms, which is connected (through transistor T6) to NAND gate N17 to activate the NAND gate oscillator. IC20 is wired as an oscillator.

When the tank is full, pin 2 of IC15 goes high to trigger IC19 through transistor T8. IC19 is wired as a one-minute monostable and transistor T7 activates the oscillator during this period. One-minute beep indicates that the tank is full and the motor has turned off.

The power supply circuit is shown in Fig. 3. The AC mains supply is stepped down by transformer X1 to deliver a secondary output of 7.5V-0-7.5V AC (15V AC), 1A. The output of the transformer is rectified by a full-wave bridge rectifier comprising diodes D25 through D28. Capacitor C60 acts as a filter to eliminate ripples. IC12 and IC22 provide regulated terminated on connector Con-1(A). These are to be extended to corresponding points of connector Con-1(B). Pads have been

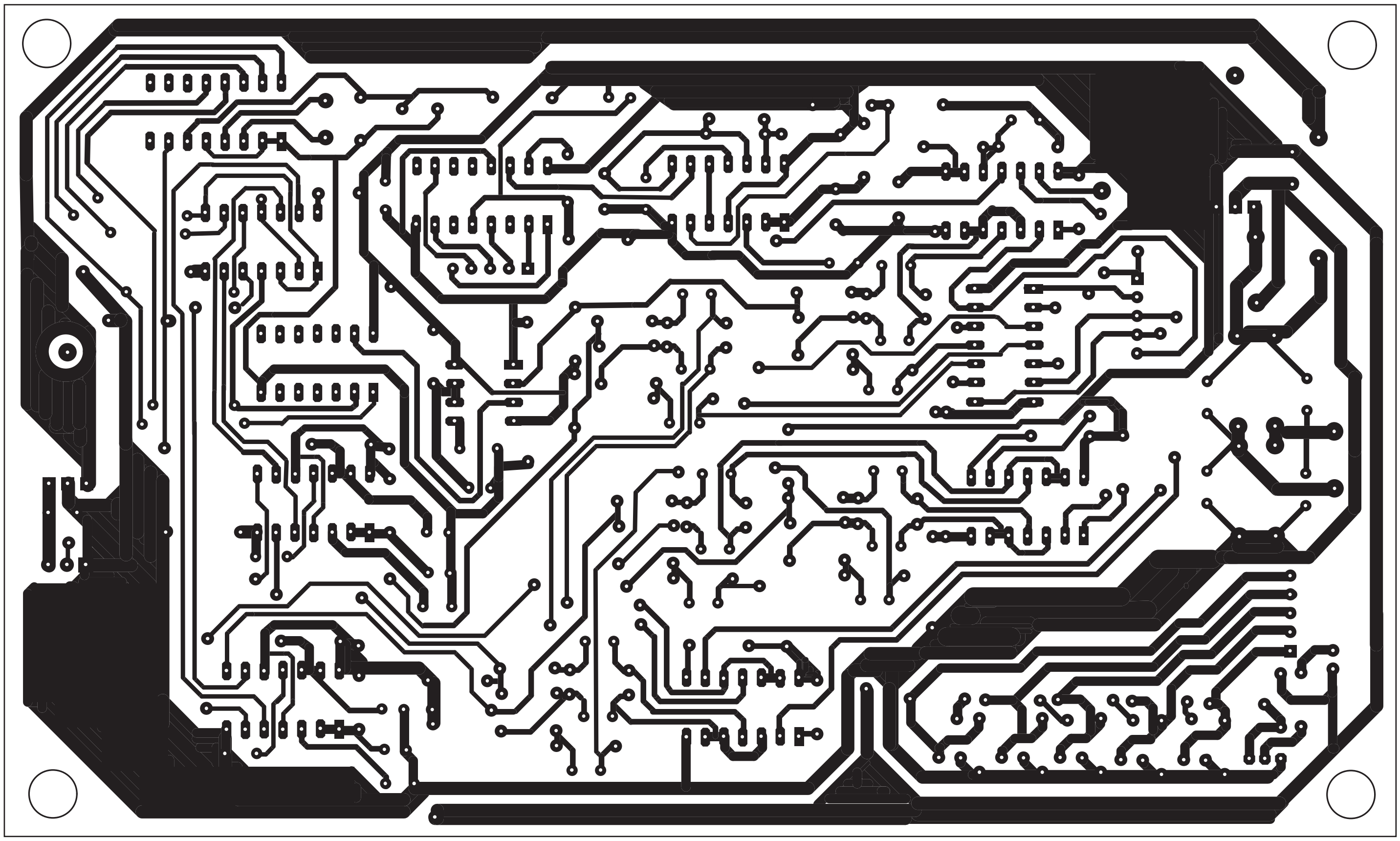


Fig. 4: Actual-size, single-side PCB for circuits of Figs 1 and 3 (PCB-1)

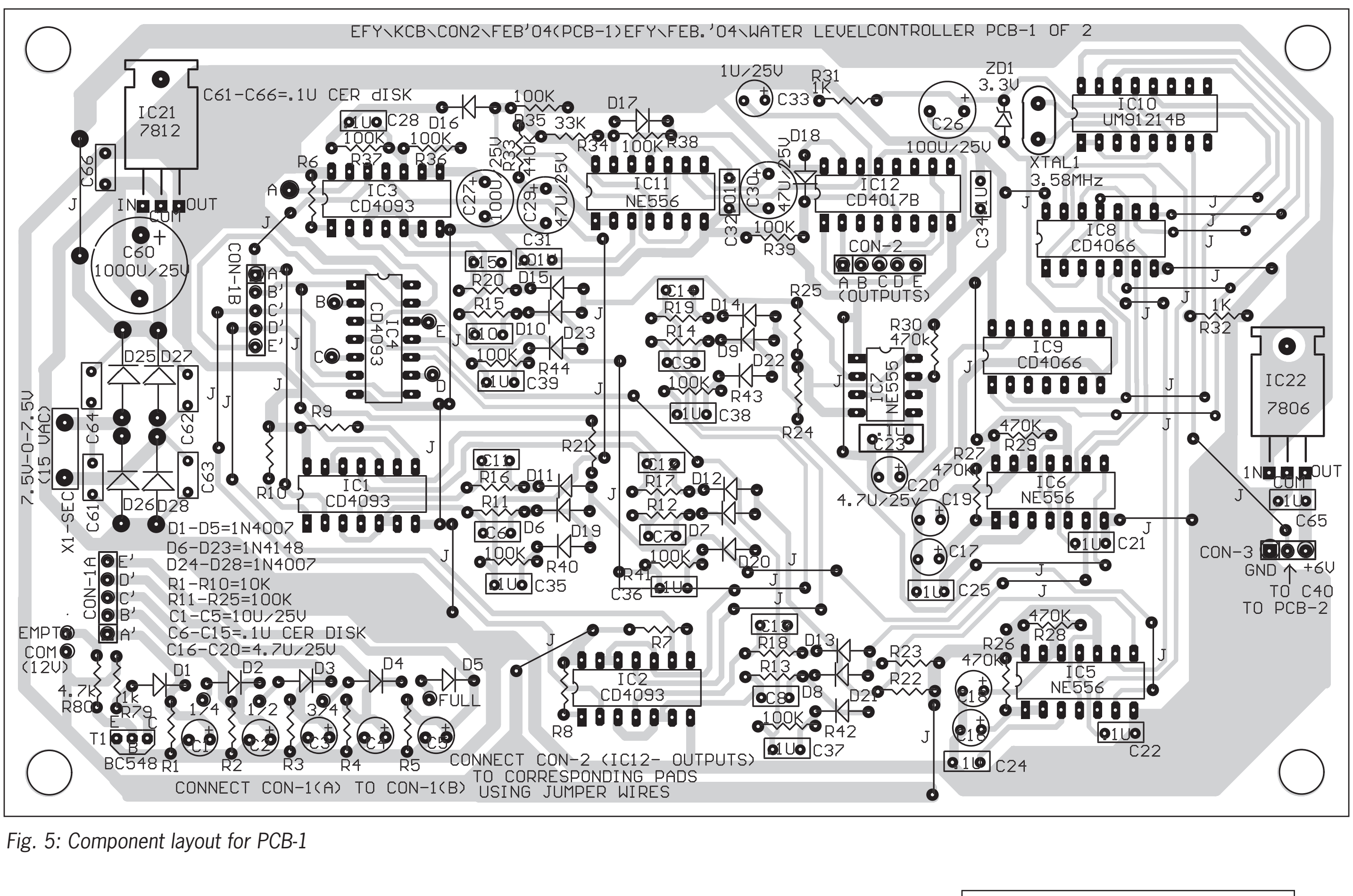


Fig. 5: Component layout for PCB-1

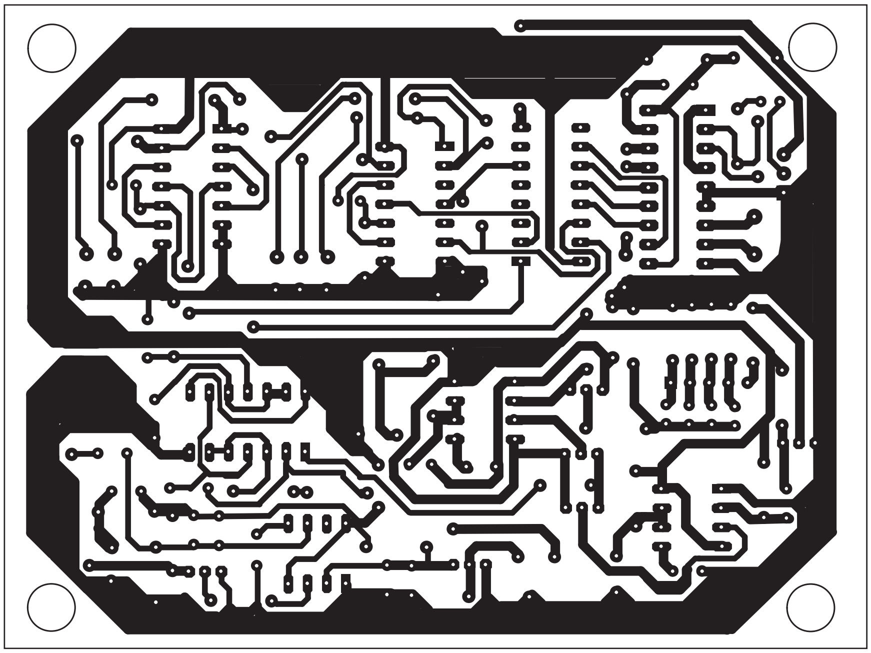


Fig. 6: Actual-size, single-side PCB for Fig. 2 (PCB-2)

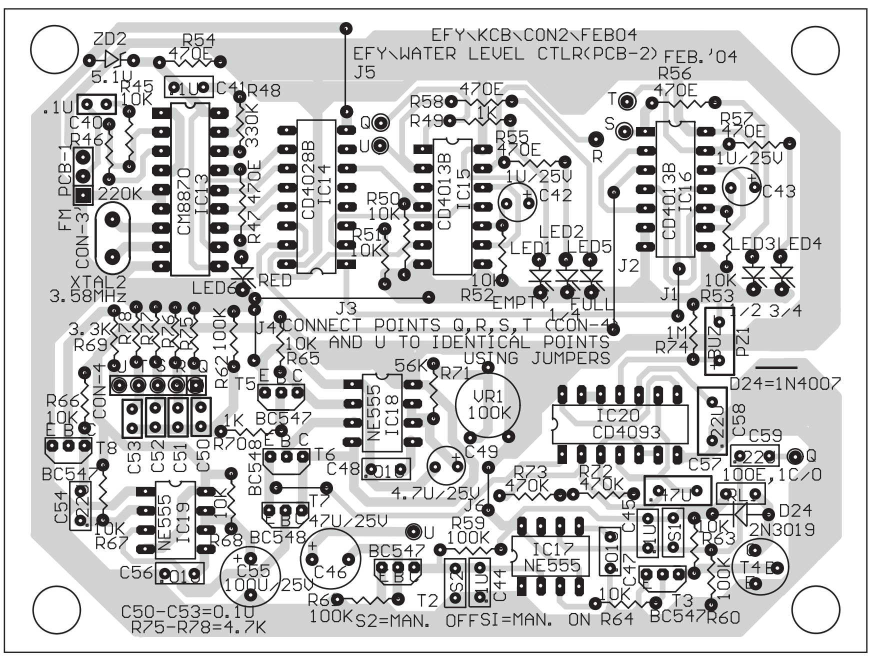


Fig. 7: Component layout for PCB-2

provided (and indicated) for connecting the probes using wire jumpers.

Similarly, PCB for Fig. 2 is shown in Fig. 6 with its component layout in Fig. 7. Identical points (Q, R, S, T and U) terminated on connector/pads need to be connected together using wire jumpers. 6V power supply including ground and DTMF output from connector Con-3 in PCB-1 is to be connected to Con-3 on PCB-2. ❑

Readers' comments:

Q1. IC6 (NE556) gets heated excessively within 5 to 6 seconds. As a result, I had to stop at the very first stage of testing. Note that the manual 'on'/'off' switch (S1) is functioning properly.

Q2. When 12V power supply to the circuit is switched on, after 5 to 10 seconds, a 'tick-tick' sound comes from the buzzer. Is it the sound of DTMF generated by IC10 when scanning the water level?

Q3. In Fig. 2, what is the function of variable resistor VR1 (100k)?

Q4. I have used 0.1pF and 0.01pF ceramic disk capacitors in place of 0.1µF and 0.01µF capacitors. The vendor says these will do the job. Please give your suggestion.

Q5. If I switch on mains when the water tank is empty, what time will it take to scan and start the motor relay?

Somnath Roy
Through e-mail

Reply to Somnath Roy by the author Parmar Latesh B.:

I am very thankful to Mr Roy for his keen interest in my circuit. The clarifications to his doubts are as follows:

A1. IC6 is getting heated due to some wrong connection around it or shorting of its two adjacent pins. Check properly. If everything is okay; the power supply may be faulty. Replace the transformer with one having a rating of 12-0-12V, 750 mA.

A2. The 'tick-tick' sound is not a DTMF tone generated by IC10. It may be due to the water level sensed by the probes (sensors) in the tank. So whenever the power to the transmitter section is switched on, the tick-tick sound is heard after 3 to 4 seconds as the level is scanned one by one.

A3. VR1 is used to adjust the time duration up to which the output of IC18 should remain high. This output activates NAND gate N17 of the buzzer section.

A4. The values of 0.1pF and 0.01pF cannot be replaced with microfarads (µF). These values are much less than microfarads. The values of ceramic or polyester capacitors are always marked in picofarads (pF). For example, the values of ceramic capacitors marked as 104 and 103 are read as 100,000 pF and 10,000 pF, respectively. But these values can be converted into microfarad units. Thus, when converted into microfarad units, 104 and 103 become 0.1 µF and 0.01 µF, respectively. (For conversion into different units, please refer to page 24 of Q&A section in April 2004 issue.)

A5. The level-scanning section will take approx. 23 seconds to scan all the levels. This time delay is provided by IC11 and R33 and C29. There is no predetermined time to switch on the motor. One can do it at any time manually, or soon after switching on the circuit. The water level is scanned only after the power to the circuit is switched on.

PC-BASED DATA LOGGER

M. DEEPAK

Here's a simple PC-based data logger to acquire slowly varying signals through the parallel port of a PC. It uses a 'C' program for data acquisition and plotting a voltage vs time graph on the monitor screen. This data logger can be used for automating simple experiments in physics laboratories and or monitoring slowly varying physical variables such as temperature in industries. Its range of operation, resolution and the maximum data rate are –5V to +5V, 39.2 mV and 18 samples per second, respectively.

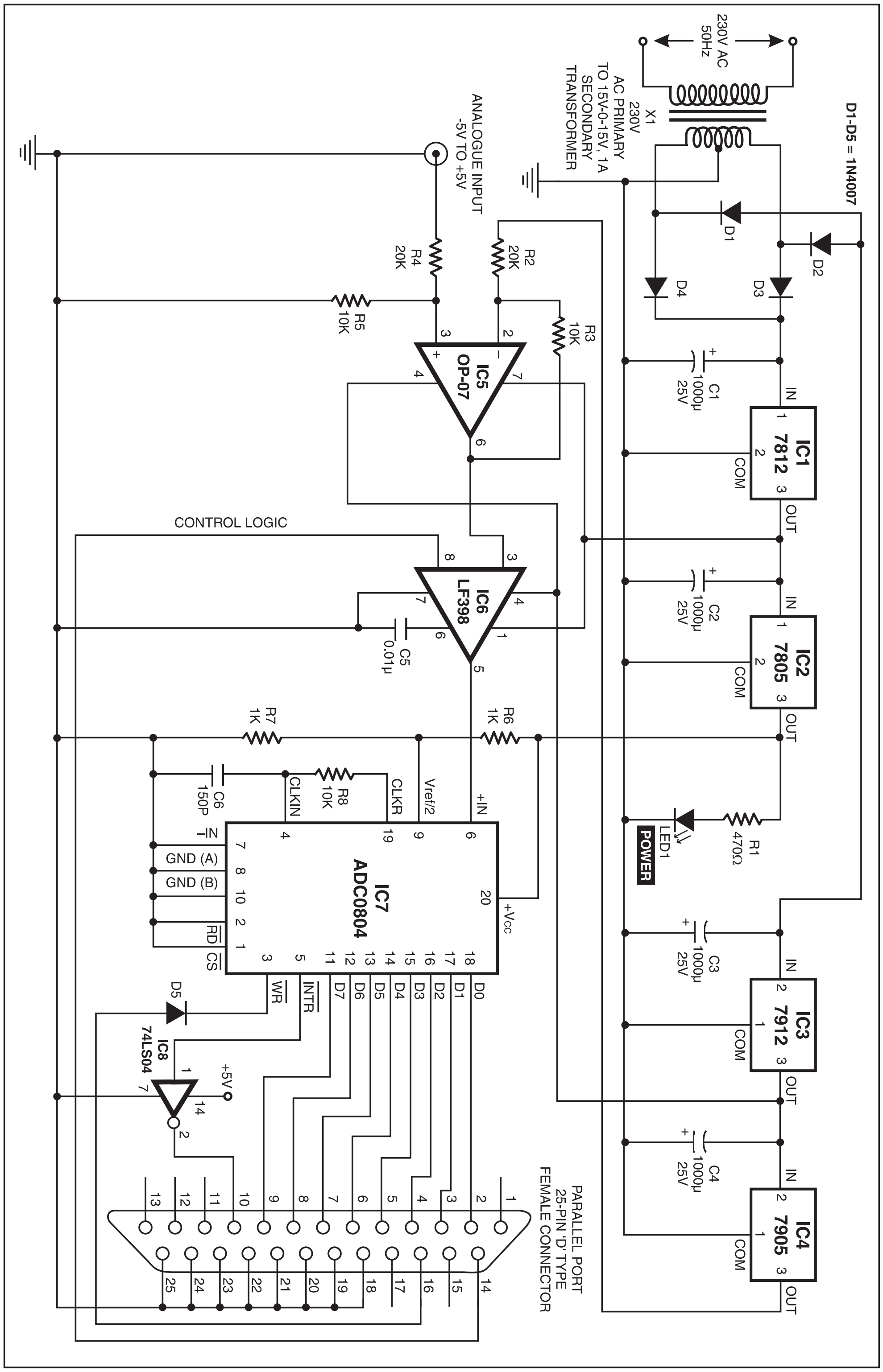


Fig. 1: Circuit of PC-based data logger

Power supply

The data logger requires regulated +5V, –5V, +12V and –12V DC supplies, which are obtained using regulator ICs 7805, 7905, 7812 and 7912, respectively. The power supply circuit uses a 15V-0-15V centre-tapped transformer. The outputs of the secondary of the transformer are applied to two full-wave rectifiers. The output of full-wave rectifier comprising diodes D3 and D4 is fed to positive DC regulator ICs 7812 (IC1) and 7805 (IC2), and the output of full-wave rectifier comprising diodes D1 and D2 is fed to negative DC regulator ICs 7912 (IC3) and 7905 (IC4). The outputs of rectifiers are pulsating DCs. Each rectifier output is filtered by capacitors C1 and C3 (1000 µF, 25V), respectively. Regulator ICs 7812 and 7912 provide regulated +12V and –12V DC. The outputs of ICs 7812 and 7912 are also given to the inputs of regulator ICs 7805 and 7905 to obtain +5V and –5V DC, respectively.

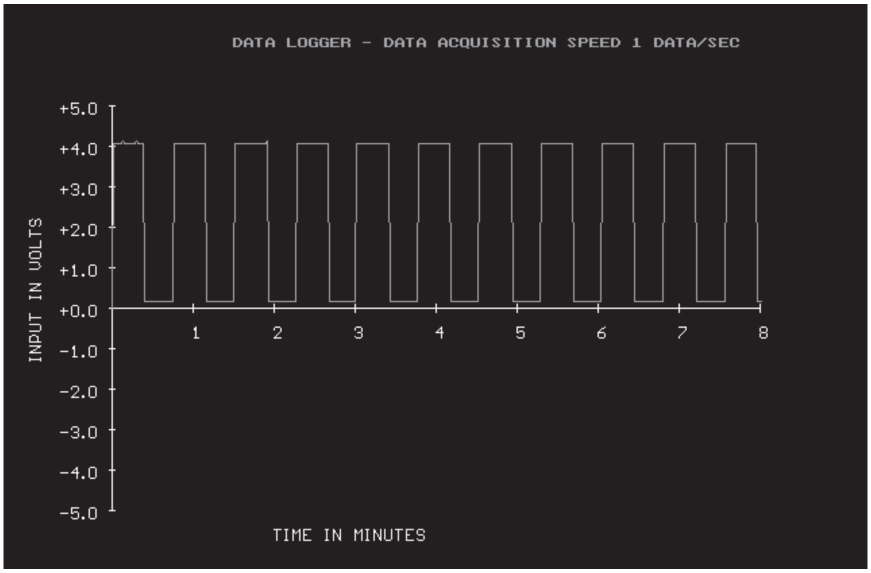


Fig. 2: Squarewave output for a squarewave input of 1Hz, 4V

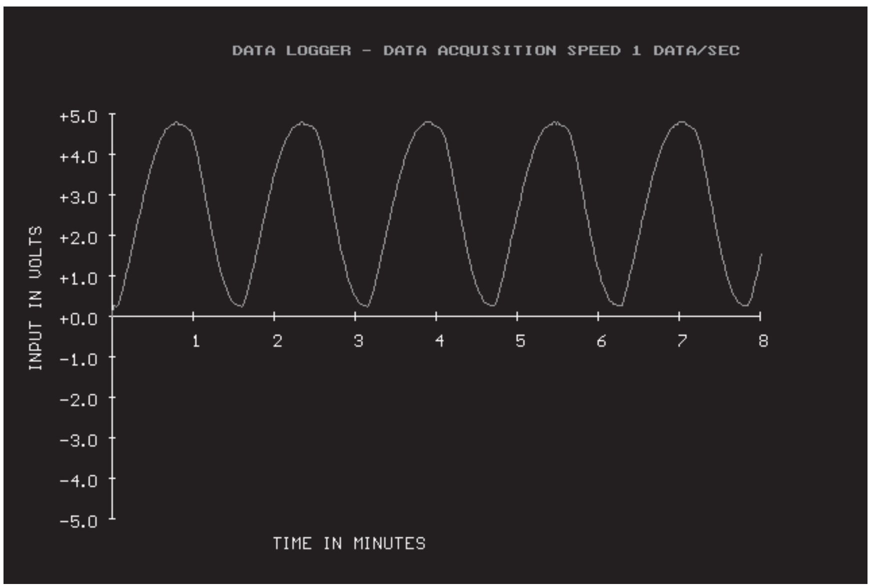


Fig. 3: Sinewave output for a sinewave input of 1Hz, 5V

Circuit description

Fig. 1 shows the circuit of the PC-based data logger. It uses analogue-to-digital converter ADC0804 (IC7), sample-and-hold IC LF398 (IC6) and op-amp IC OP-07 (IC5). The op-amp is in the differential amplifier configuration and transforms the input voltage in the range of –5V to +5V to the range 0 to +5V. It operates off +12V and –12V DC supplies. Resistors R2 and R4 (each 20 kilo-ohms) are input series resistors. Feedback resistor R3 and biasing resistor R5 (each 10 kilo-ohms) along with the input resistors set the gain of the amplifier to 0.5. The output voltage at pin 6 of IC5 is given by the following relationship:

Therefore, for the inputs of –5V, 0V

$$V_o = \frac{V_{in}+5}{2}$$

and +5V, the outputs are 0V, +2.5V and +5V, respectively. The –5V at the inverting input of IC5 shifts its output level to above 0V. Thus for a swing of input between –5V and +5V, the output swings from 0 to +5V.

The output of the op-amp is applied to input pin 3 of IC6 (LF398). IC LF398 also operates off +12V and –12V DC. It has a small acquisition time (10 ms) and less output noise in hold mode. Droop rate is low at 10^{-5} V/ms with 0.01µF polypropylene hold capacitor C5 connected to its pin 6. The control logic signal from pin 14 of the parallel port to pin 8 of this IC controls sample and hold operation. Logic 1 puts the device in sample mode and logic 0 puts it in hold mode. The output is obtained at pin 5 while pin 7 is grounded.

IC ADC0804 is an 8-bit, successive approximation type ADC that requires 5V DC regulated power supply. It has an in-built clock generator whose operating frequency (f) is given by:

f = 1/1.1RC

The frequency of clock generation is set to approximately 610 kHz by resistor R8 (10 kilo-ohms) and capacitor C6 (150 pF). The ADC converts analogue signals in the range of 0 to +5V to 8-bit digital data. The conversion time is approximately 100 ms.

The output of sample-and-hold IC6 at pin 5 is applied to the +IN (pin 6) of the ADC. The –IN (pin 7) of the ADC is grounded. Positive 5V is applied to pin 20 and +2.5V is applied to $V_{ref}/2$ input (pin 9) of the ADC through divider network comprising resistors R6 and R7 (each 1 kilo-ohm).

PARTS LIST

Semiconductors:	
IC1	- 7812 +12V regulator
IC2	- 7805 +5V regulator
IC3	- 7912 –12V regulator
IC4	- 7905 –5V regulator
IC5	- OP-07 op-amp
IC6	- LF398 sample-and-hold amplifier
IC7	- ADC0804 analogue-to-digital converter
IC8	- 74LS04 hex inverter
D1-D5	- 1N4007 rectifier diodes
LED1	- Power-indicator red LED
Resistors (all ¼-watt, ±5% carbon, unless stated otherwise):	
R1	- 470-ohm
R2, R4	- 20-kilo-ohm
R3, R5, R8	- 10-kilo-ohm
R6, R7	- 1-kilo-ohm
Capacitors:	
C1-C4	- 1000µF, 25V electrolytic capacitors
C5	- 0.01µF ceramic disk capacitor
C6	- 150pF ceramic disk capacitor
Miscellaneous:	
X1	- 230V AC primary to 15V-0-15V, 1A secondary step-down transformer
	- 25-pin D type female connector
	- Two 25-pin D type male connectors (for connecting the circuit to the female connector at the back of the PC)

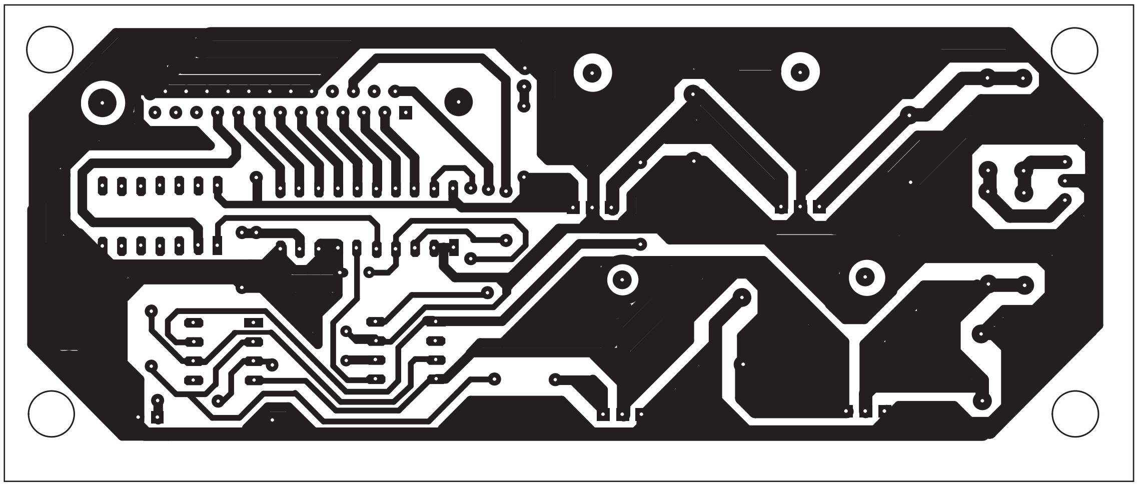


Fig. 4: Actual-size, single-side PCB conductor layout for PC-based data logger

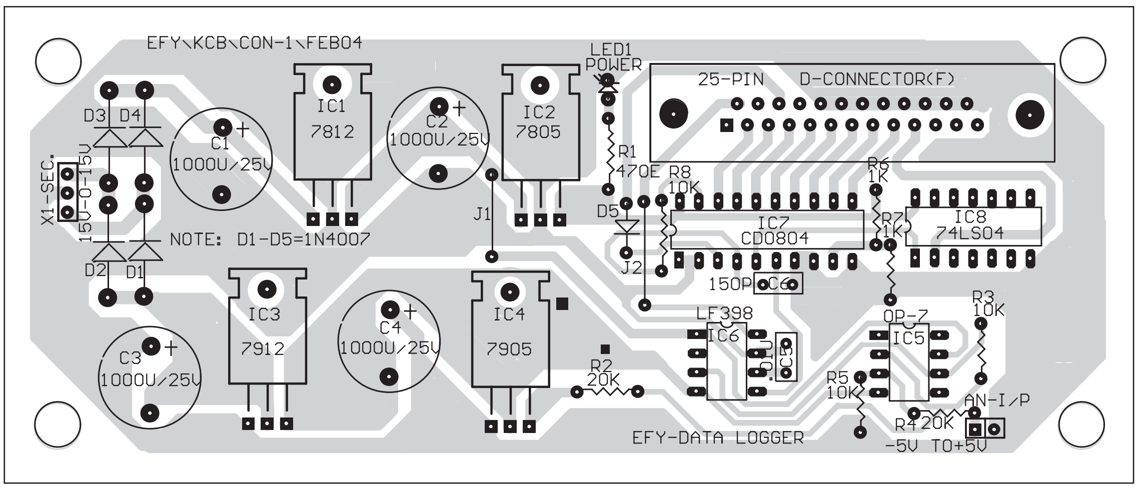


Fig. 5: Component layout for the PCB

The ADC operation is controlled by chip-select ($\overline{CS}$), read ($\overline{RD}$) and write ($\overline{WR}$) inputs. Logic 0 on $\overline{CS}$ (pin 1) keeps the ADC enabled. Logic 0 on $\overline{RD}$ (pin 2) enables the converted data to appear on digital output pins 11 through 18 of the ADC. The start-of-conversion pulse is applied to $\overline{WR}$ (pin 3). When a low-to-high transition occurs on $\overline{WR}$ pin, the ADC starts conversion. The interrupt $\overline{INTR}$ is used as end-of-conversion signal.

After the conversion is over, the $\overline{INTR}$ signal (pin 5) goes low. The $\overline{INTR}$ signal goes high when the conversion starts and remains high during conversion. The signal is inverted by IC8 and given to pin 10 of the parallel port, which provides the required positive edge for generating hardware interrupt on end of conversion. When the $\overline{INTR}$ output goes low, it indicates that A to D conversion is completed. Digital outputs D0 through D7 of the ADC are connected to data pins 2 through 9 of the parallel port, respectively. Since the ADC converts analogue inputs in the range of 0 to +5V to 8-bit digital data, the resolution of the ADC is 19.60 mV. (Resolution = V_{ref}/counts, where V_{ref} is 5V and counts are 256 for 8-bit digital data.)

The ADC (IC7) and the sample-and-hold IC (IC6) are controlled through the parallel port of the PC. The input/output (I/O) addresses of data, status and control registers of the parallel port LPT1 are 0378H, 0379H and 037AH, respectively. The data, status and control bits are designated as D_n, S_n and C_n in the following discussion.

Pin details of the three registers of the parallel port are given in the table. The 'n' prefix to the signal name denotes that the signal is active-low. IC LF398 is set to sample/hold mode by setting C1 bit (pin 14) of the control register of the parallel port. Resetting C1 bit to low provides logic 1 and setting C1 bit to high provides logic 0 on the control input (pin 8) of the LF398 as C1 bit is internally inverted and made available on pin 14 of the parallel port. Low-to-high transition of C2 bit that appears on pin 16 of the parallel port is applied to $\overline{WR}$ (pin 3) of the ADC to initiate data conversion. The falling edge of $\overline{INTR}$ signal from the ADC, which is inverted and applied to pin 10 of the parallel port, generates hardware interrupt thro-ugh IRQ7 line (not shown in Fig. 1). On most systems, the IRQ7 line is used to drive the first parallel port, normally for the use of a printer.

Control bit 4 (C4) of the control port is a PC output line. Making this bit high enables the interrupt circuitry associated with the ACK input (pin 10) of the parallel port. The parallel port is enabled to use IRQ7 line for interrupt by setting C4 bit of the control register to high. Note that C4 bit is not associated with the parallel port connector, rather it controls logic on the printer card or the PC motherboard.

The digitised data is read from the data register, which is configured to operate in input mode by setting C5 bit of the control register high.

The outputs for 1Hz squarewave and sinewave inputs with amplitudes of 4V and 5V, respectively, are shown in Figs 2 and 3.

The actual-size, single-side PCB for PC-based data logger is shown in Fig. 4 and its component layout in Fig. 5.

Register Pin Details of the PC's Parallel Port				
Parallel port pin No.	**Signal name**	**Direction**	**Register bit**	**Inverted**
1	nStrobe	Out	Control-0	Yes
2	Data0	In/Out	Data-0	No
3	Data1	In/Out	Data-1	No
4	Data2	In/Out	Data-2	No
5	Data3	In/Out	Data-3	No
6	Data4	In/Out	Data-4	No
7	Data5	In/Out	Data-5	No
8	Data6	In/Out	Data-6	No
9	Data7	In/Out	Data-7	No
10	nAck	In	Status-6	No
11	Busy	In	Status-7	Yes
12	Paper-Out	In	Status-5	No
13	Select	In	Status-4	No
14	Linefeed	Out	Control-1	Yes
15	nError	In	Status-3	No
16	nInitialize	Out	Control-2	No
17	nSelect-Printer	Out	Control-3	Yes
18-25	Ground	—	—	—

Software program

The C program (datalog.c) acquires data at the rate of one sample per second and plots a simulated time vs voltage graph on the monitor screen. It uses two interrupt service routines (ISRs): one is invoked through IRQ0 (system timer) interrupt and the other is invoked through IRQ7 (parallel port) interrupt. The ISRs are invoked by modifying the respective vectors in the interrupt vector table. However, before calling up the ISRs of application program to the interrupts, the existing vectors for the interrupts should be read from interrupt vector table and saved. The ISRs are invoked when the interrupt occurs, only if the interrupts are enabled.

The priority interrupt controller that occupies addresses 0020H and 0021H in the system I/O map is programmed to enable or disable the interrupts. The IRQ0 and IRQ7 interrupts are enabled by resetting D0 and D7 bits of the interrupt mask register at I/O address 0021H without affecting interrupt masks of other IRQs. When an ISR is invoked on an interrupt, the ISR should first execute the previous ISR which is chained to the interrupt. At the end of the ISR, it is necessary to issue an end-of-interrupt command to the interrupt controller. It is issued by sending control byte 20H to I/O address 0020H. Before terminating the application, the vectors are restored into the vector table to restore the status of the system.

The IRQ0 interrupt provides timing to start of conversion (at pin 16 of the parallel port) and the IRQ7 interrupt provides timing to end of conversion. The system timer generates 18.2 IRQ0 interrupts in one second. To set the required data rate, the counter is initialised with an appropriate value in the program; here the counter is set to '18'. The ISR for IRQ0 on each interrupt decrements the counter. When the counter reaches zero, the program sends start-of-conversion pulse to the ADC. The ISR for IRQ7 sets a flag to indicate that conversion is complete and the digitised data may be read from the data register. Before providing start-of-conversion pulse, the sample-and-hold IC is set to hold mode. Similarly, after the digital data is read from the data register, the sample-and-hold IC is set back to sample mode.

Execution of the program

The screen is initialised to graphics mode and a graphic chart is simulated. It requires the graphics initialiser file EGAVGA.BGI to be in the directory C:\TC\BGI\.

The vectors of existing timer and parallel port ISR are saved in variables 'oldintr' and 'oldtimer' using the getvect(....) statements. The vectors of ISRs of the application program, newintr() and newtimer(), are loaded into the interrupt vector table using the setvect(....) statements.

The status of interrupt masks is obtained and saved. The interrupt requests IRQ0 and IRQ7 are enabled.

The variable count that determines the data rate is initialised to the required rate: 1, 2, 3, 6, 9 and 18 data samples per second, respectively.

The program then enters the main loop and keeps monitoring the timerflag and the intrflag until a key is pressed. If the intrflag is 1, the program:

1. Reads the digitised data from the data register
2. Puts the sample-and-hold IC to sample mode by resetting C1 bit to low
3. Computes coordinates of the pixel corresponding to the data
4. Plots data point and draws a line joining the previous data point on the monitor screen and completes acquisition and plotting of one data
5. Updates variables for acquiring next data

If the timerflag is 1, the program determines whether it is the time to issue SOC (start of conversation) pulse. If so, the program:

1. Puts the sample-and-hold IC to hold mode by setting C1 bit to high
2. Issues start-of-conversion pulse by setting, resetting and setting C2 bit

If a key is pressed, the program restores interrupt mask and interrupt vectors, and terminates.

ISR for IRQ0 interrupt. The timer ISR performs the following tasks on each interrupt:

1. Calls the previous ISR in the chain
2. Enters 1 into the timerflag variable to indicate the main program that a timer interrupt has occurred
3. Sends end-of-interrupt command to the interrupt controller

ISR for IRQ7 interrupt. The ISR for the IRQ7 interrupt performs the following tasks on each interrupt:

1. Calls the previous ISR in the chain
2. Enters 1 into the intrflag variable to indicate the main program that an interrupt on IRQ7 line has occurred
3. Sends end-of-interrupt command to the interrupt controller

Note. 1. The range of operation and the resolution can be improved by using 12-bit ADCs operating on a wider range of analogue inputs (such as AD574A), but this will make design of the system more complicated.

2. The data rate can be improved using a separate clock circuit on-board. However, there is limit for the same as conversion time of the ADC is 100 ms.

SOURCE CODE FOR DATA LOGGER (DATALOG.C)

```
/* DATA LOGGER - BY M DEEPAK */
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<dos.h>

#define CONT 0x37A
#define STATUS 0x379
#define DATA 0x378

void interrupt(*oldintr)();
void interrupt newintr();
void interrupt(*oldtimer)();
void interrupt newtimer();
void drawchart();

int intrflag;
int timerflag;

void main()
{
        int count,i,time = 80,newvolt=0,oldvolt = 0;
        unsigned char d=0,intmask;
        float yold,ynew;
        int gd=DETECT,gm;

        clrscr();
        initgraph(&gd,&gm,"C:\\TC\\BGI");
/* initialize graphics mode                              */
        drawchart();
/* simulate graphics chart                               */

        oldintr = getvect(0x0f);
/* save vector of old ISR for IRQ7                       */
setvect(0x0f,newintr);
/* load vector of new ISR for IRQ7                       */
oldtimer = getvect(0x08);
/* save vector of old ISR for IRQ0                       */
        setvect(0x08,newtimer);
/* load vector of new ISR for IRQ7                       */
        intmask = inportb(0x21);
/* get the masking status of IRQ7 and IRQ0               */
        intmask &= 0x7e;

        outportb(0x21,intmask);
/* enable IRQ7 and IRQ0 interrupts                       */

        intrflag=0; timerflag = 0; count = 18;

        do{
                if(intrflag)
/* if digitized data is ready                            */
                {
                d = inportb(DATA);
/* read the data                                         */
        outportb(CONT, inportb(CONT) & 0xfd);


/* place S/H to sample mode                  */
        oldvolt = newvolt;
/*      find coordinates of pixel for the data */
                        newvolt = d-128;
                        setcolor(2);
        yold = (oldvolt*150.0)/(127.0);
        ynew = (newvolt*150.0)/(127.0);
        if(time<=560) line(time,250 -
(int)yold,time+1,250 - (int)ynew);


/* plot data and draw line                          */
                        time+=1;
                        intrflag = 0;
                }
                if(timerflag)
   /* if time to initiate SOC                       */
                {
                        timerflag = 0;
                        count--;
                        if(!count)
                        {

outportb(CONT, inportb(CONT) | 0x02);  /* place S/H to hold
mode */

outportb(CONT, inportb(CONT) | 0x34);  /* start of conver-
sion pulse */

outportb(CONT, inportb(CONT) & 0xfb);
                for (i=0; i<6000; i++);
                /* delay */

outportb(CONT, inportb(CONT) | 0x04);
        count = 18;
        /* set counter for next round               */
                        }
                }
        }while(!kbhit());
/* acquire and plot the data till key is pressed */

        intmask |= 0x80;
        /* restore the status of interrupts */
        outportb(0x21,intmask);
        setvect(0x0f,oldintr);
/* restore old vectors of ISRs for IRQ7 and IRQ0 */
        setvect(0x08,oldtimer);
        getch();
        closegraph();
        /* terminate the program     */
  }
   void drawchart()
        /* simulate graphics chart    */
   {
        int x;
        float q;
        char b[10];
        settextstyle(0,0,1);
        setcolor(3);
        outtextxy(170,50,"DATA LOGGER - DATA
ACQUISITION SPEED 1 DATA/SEC");
        settextstyle(2,1,5);
        setcolor(14);
        outtextxy(15,180,"INPUT IN VOLTS");
        settextstyle(2,0,5);
        outtextxy(200,410,"TIME IN MINUTES");
        setcolor(15);
        line(80,100,80,400);
/* draw the x and y axis */
        line(80,250,560,250);
        for(q=10.0;q>=0.0;q--)
/* draw y-axis graduation and calibration  */
        {
        line(78,100+(10.0-q)*30,82,100+(10-q)*30);
                sprintf(b,"%+.1f",q-5.0);
                outtextxy(42,94+(10-q)*30,b);
        }
        for(x=1;x<=8;x++)
/* draw x-axis graduation and calibration  */
        {
                line(80+x*60,247,80+x*60,253);
                sprintf(b,"%d",x);
                outtextxy(80+x*60,260,b);
        }
   }

   void interrupt newtimer()
        /* new timer ISR                      */
   {
        oldtimer();
/* execute old ISR      for IRQ0 interrupt */
        timerflag = 1;
        outportb(0x20,0x20);
/* issue end of interrupt command        */
   }

   void interrupt newintr()
/* new IRQ7 ISR                       */
   {
        oldintr();
        /* execute old ISR for IRQ7 interrupt */
        intrflag=1;
        outportb(0x20,0x20);
/* issue end of interrupt command */
   }
```
❑

Readers' comments:

Q1. Does the 'PC-based Data Logger' accept 230V AC, 50 Hz as input and plot a sinusoidal graph of 5V, 1Hz as the output on the screen? If the input voltage is less than 230V, will the output voltage be less than 5V? Please clarify.

Subhabrata Gupta
Jorhat Engineering College

The author M. Deepal replies:

A1. In this project, 230V AC is not the input. It is down-converted to 5V and 12V for use as power supplies for the circuit. It has nothing to do with the voltage that is plotted on the screen. The signal plotted on the screen is the corresponding input given as the input to IC OP-07. The range of input signal is -5 to +5V.

LIFT OVERLOAD PREVENTER

NANDHA KUMAR T.

Here's a lift safety system that stops lift operation when the number of persons inside the lift exceeds the lift capacity. It can be installed in factories or other large establishments where lift is used. The circuit can also be used as visitor counter or room power control.

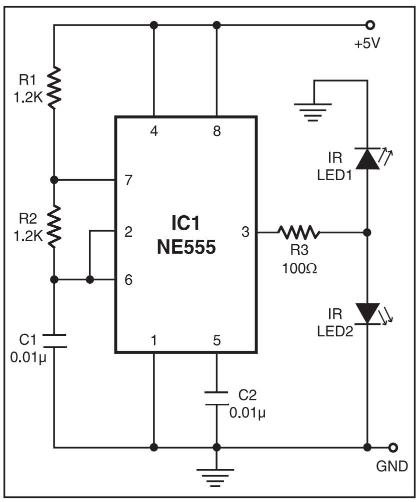


Fig. 1: Transmitter

Two transmitter and receiver pairs are used at the entry gate of the lift: one pair comprising IR LED1 and IR RX1 is installed outside the gate, while the other pair comprising IR LED2 and IR RX2 is installed inside the gate. Proper orientation of receiver and transmitter pairs is very important. The display section displays the number of persons inside the lift.

The circuit

Basically, the circuit comprises the following four sections:

1. Transmitter
2. Receiver (pulse generator)
3. Lift safety control
4. Display

1. The transmitter. Fig. 1 shows the transmitter section, where timer NE555 (IC1) is used as an astable multivibrator to produce 38kHz transmitting frequency for IR LED1 and LED2.

2. The receiver (pulse generator). The IR beams transmitted by LED1 and LED2 are incident on the corresponding infrared receiver modules RX1 and RX2 of the receiver section (refer Fig. 2), which produce a low output if the IR beam is interrupted.

When a person enters the lift room, the first and the second IR beams get interrupted in that order. On the other hand, when a person leaves the lift, the second beam is cut first and then the first beam. When the two IR beams are interrrupted one after another, a pulse is generated at pin 3 of timers IC2 and IC3 each and then both the pulses combine to form a single pulse at pin 2 of IC4 or IC5 (depending on whether a person enters or leaves the lift), which provides a clock for up or down counting.

When a person enters the lift, timers IC2 and IC3 get triggered in that order due to interruption of the first beam followed by interruption of the second beam. Triggering of timer IC2 charges capacitor C13 to drive transistor T1. At the same time, a high output appears across diode D6 due to triggering of timer IC3. This high output triggers IC4. The high output

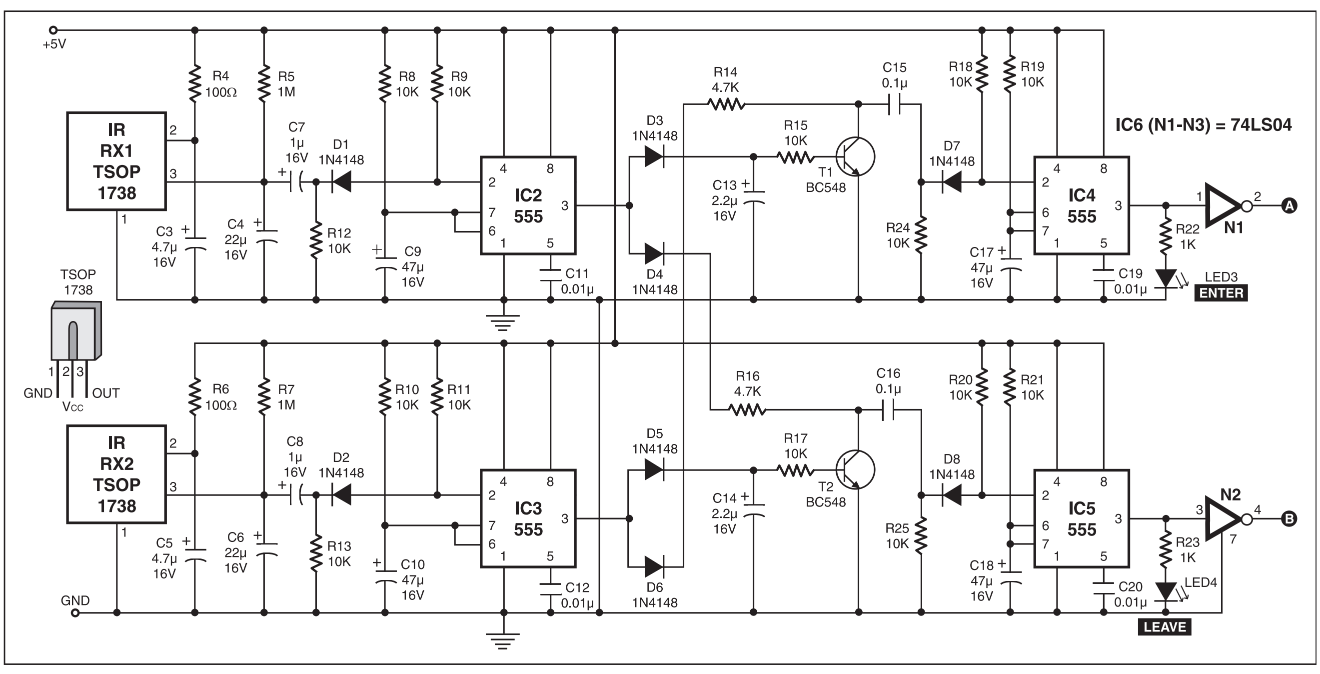


Fig. 2: Receiver (pulse generator) circuit

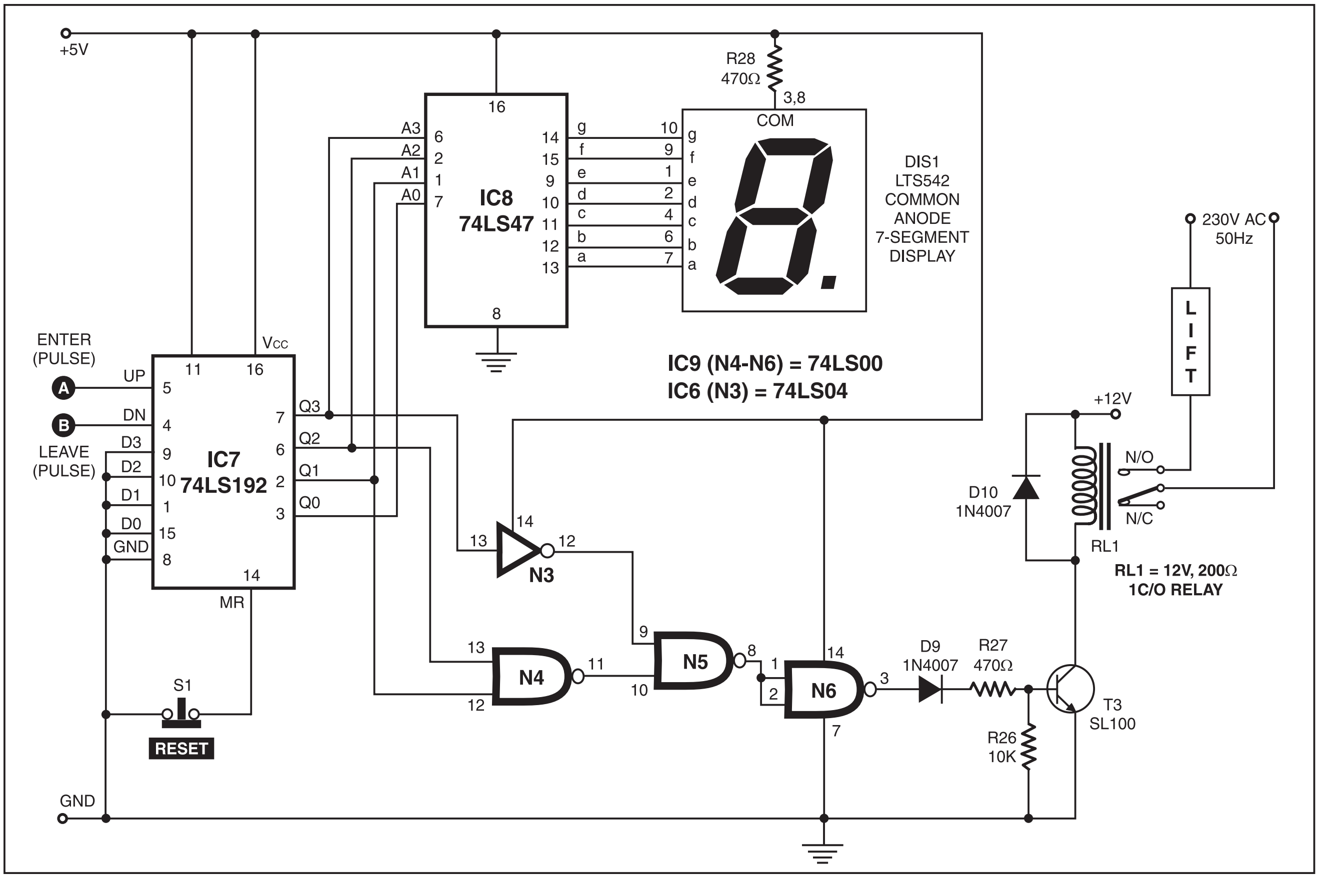


Fig. 3: Lift safety control circuit

of IC4 at its pin 3 is further given (via inverter N1) to pin 5 of IC7 for up-counting. Capacitor C14 also gets charged by timer IC3 but there is no high output across diode D4. So no pulse is available at pin 2 of IC5.

Similarly, when a person leaves the lift, timer IC3 gets triggered due to interruption of the second beam and then timer IC2 gets triggered due to interruption of the first beam. Triggering of timer IC3 charges capacitor C14 to drive transistor T2. At the same time, a high output appears across diode D4 due to triggering of timer IC2. This high output is used to trigger IC5. The high output of IC5 at its pin 3 is further given (via inverter N2) to pin 4 of IC7 for down-counting. Capacitor C13 also gets charged by timer IC2 but there is no high output across diode D6. So there is no pulse at pin 2 of IC4.

Thus, when a person enters the lift a high pulse is available at terminal A and LED3 blinks, and when a person leaves the lift, a high pulse is available at terminal B and LED4 blinks.

3. The lift safety control section. Points A and B of the receiver circuit (Fig. 2) are connected to the corresponding points of the lift safety contol circuit (Fig. 3). The lift safety control section is built around up-/down-counter IC 74LS192 (IC7), inverter N3 and

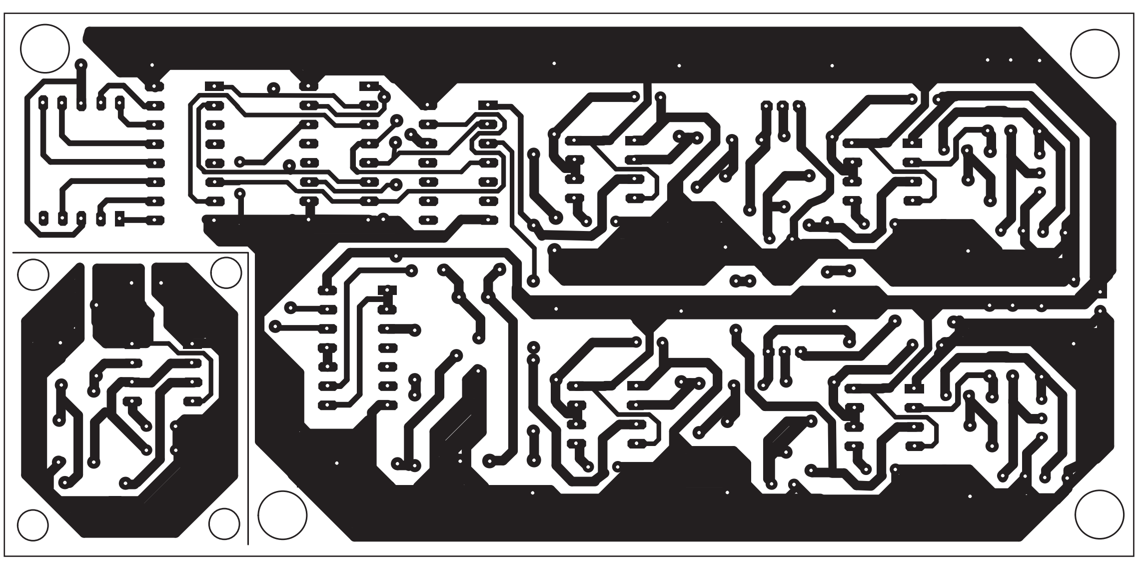


Fig. 4: Actual-size, single-side PCB for Figs 1, 2 and 3

quad NAND gate IC74LS00 (IC9). The output of NAND gate N6 is fed to relay driver transistor T3 for activating the lift via N/O contacts of relay RL1 as shown in Fig. 3. The relay requires 12V to operate.

This circuit is designed for a lift capacity of nine persons with safety limit of five persons. When the safety limit is crossed, i.e. the number of persons inside the room exceeds five, the lift controller is switched off. When the number of persons inside the room reduces to five, the lift control is restored as shown in Truth Table I.

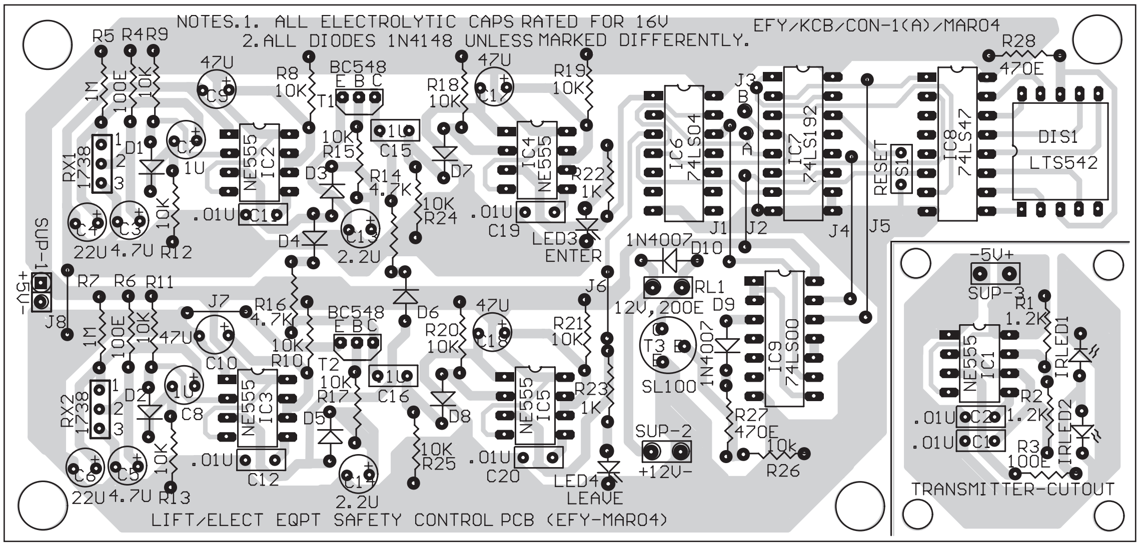


Fig. 5: Component layout of the PCB in Fig. 4

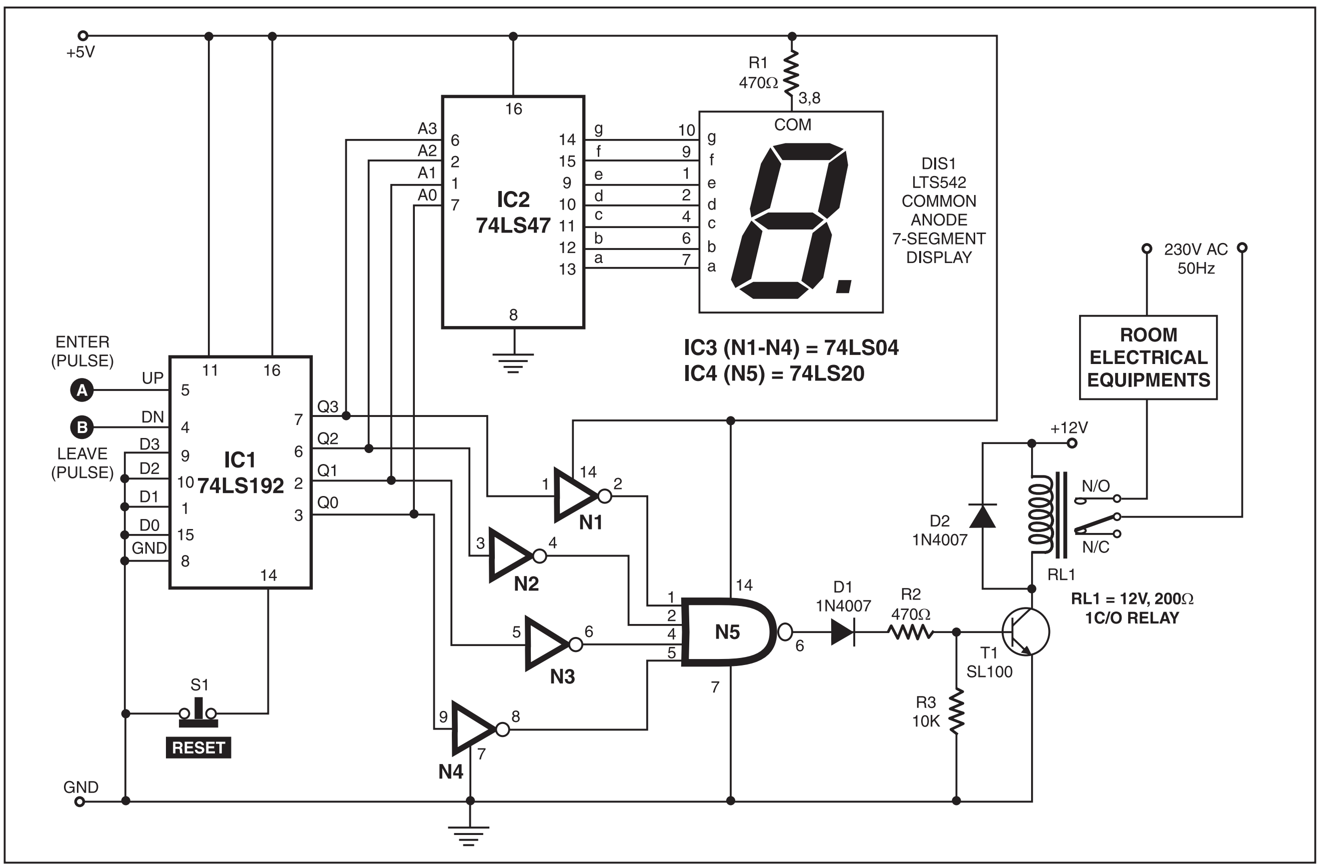


Fig. 6: Room power control

The safety limit can be extended by changing the circuit's logic part (comprising inverter N3 and NAND gate IC9) accordingly.

Counter IC 74LS192 is capable of both up-counting and down-counting if configured properly. When a person enters the lift the count of IC7 is incremented by one, which provides a pulse at its pin 5. When a person leaves the lift the count of IC7 is decremented by one, which provides a pulse at its pin 4. The counter counts up to 9 only. When a person enters the lift, LED3 glows, and when a person leaves, LED4 glows.

4. Display section. The display section consists of BCD-to-7-segment decoder/driver 74LS47 (IC8) and common-anode,

TRUTH TABLE I

Counter output of IC7 Q3	Q2	Q1	Logic output at pin 3 of N6 $= \overline{Q3}.(\overline{Q1}+\overline{Q2})$
0	0	0	1
0	0	0	1
0	0	1	1
0	0	1	1
0	1	0	1
0	1	0	1
0	1	1	0
0	1	1	0
0	0	0	0
1	0	0	0

TRUTH TABLE II

Counter output of IC7 Q3	Q2	Q1	Q0	Dec. Equ.	Logic output at pin 6 of N5 $=\overline{(\overline{Q0}.\overline{Q1}.\overline{Q2}.\overline{Q3})}$
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	2	1
0	0	1	1	3	1
0	1	0	0	4	1
0	1	0	1	5	1
0	1	1	0	6	1
0	1	1	1	7	1
1	0	0	0	8	1
1	0	0	1	9	1

7-segment display LTS542 (DIS1). The four BCD outputs (Q0 through Q3) of up/down counter IC7 are fed to decoder/driver IC8. The active-low outputs of the decoder are connected to the corresponding pins of the 7-segment, common-anode display.

Construction

The circuit (excluding relay) works off a 5V regulated power supply. The actual-size, single-side PCB for the lift safety control system comprising transmitter (Fig. 1), receiver (Fig. 2) and lift control (Fig. 3) sections is shown in Fig. 4 and its component layout in Fig. 5. The combined PCB can be cut to separate the transmitter section from the rest of the PCB.

Room power control

To replace the lift control circuit (Fig. 3) with the room power control circuit (Fig. 6), simply remove the lift safety control circuit connected between points A and B of the receiver (pulse generator) circuit. Now connect points A and B of the room power control circuit to the corresponding points of the receiver circuit.

Fig. 6 shows the circuit for automatic room power control along with display. When nobody is present in the room, the light in the room is automaticallly switched off. The circuit consists of up-/down-counter 74LS192 (IC1), display driver 74LS47 (IC2), common-anode display LTS542 (DIS1), inverter (IC3) and 4-input NAND gate (IC4). The output of NAND gate N5 is conneted to relay driver transistor T1 for power control of the room via N/O contact of relay RL1. The counter (IC1) counts up to 9. The 4-bit output of IC1 is inverted and fed to the dual 4-input NAND gate (IC4). If all the four bits of IC1 (Q0 through Q3) are zero, the output of IC4 is zero. Otherwise, the output of IC4 is high (logic 1).

When the output of IC4 is high, transistor T1 conducts to energise the relay, which provides mains power supply to the room. Thus only when someone is inside the room, the NAND gate output will be high and hence the power supply of the room will be 'on.' The logic (comprising IC3 and IC4) for maximum nine persons are summarised in Truth Table II.

PARTS LIST

Semiconductors:	
IC1, IC2-IC5	- NE555 timer
IC6	- 74LS04 hex inverter
IC7	- 74LS192 up/down decade counter
IC8	- 74LS47 BCD to 7-segment decoder/driver
IC9	- 74LS00 quad 2-input NAND gate
RX1, RX2	- Infrared receiver module TSOP (1738)
LED1, LED2	- Infrared transmitter LED
LED3, LED4	- 3mm red LED
T1, T2	- BC548 npn transistor
T3	- SL100 npn transistor
D1-D8	- 1N4148 diode
D9, D10	- 1N4007 rectifier diode
DIS1	- LTS542 common-anode display
Resistors (all ¼-watt, ±5% carbon, unless stated otherwise):	
R1, R2	- 1.2-kilo-ohm
R3, R4, R6	- 100-ohm
R5, R7	- 1-mega-ohm
R8-R13, R15, R17, R18-R21 R24-R26	- 10-kilo-ohm
R14, R16	- 4.7-kilo-ohm
R22, R23	- 1-kilo-ohm
R27, R28	- 470-ohm
Capacitors:	
C1, C2, C11, C12, C19, C20	- 0.01µF ceramic disk
C3, C5	- 4.7µF, 16V electrolytic
C4, C6	- 22µF, 16V electrolytic
C7, C8	- 1µF, 16V electrolytic
C9, C10, C17, C18	- 47µF, 16V electrolytic
C13, C14	- 2.2µF electrolytic
C15, C16	- 0.1µF ceramic disk
Miscellaneous:	
Power supply	- 5V regulated DC, 12V regulated DC
Relay	- 12V, 200-ohm, 1c/o relay

As the circuit uses IC 74LS192, it works only for rooms having a capacity of nine persons. However, it can be made to work for rooms having a capacity of 15 persons by using IC 74LS193 in place of IC 74LS192.

An actual-size, single-side PCB for the room power control circuit is shown in Fig. 7 and its component layout in Fig. 8. Points A and B marked on this PCB need to be connected to the corresponding points in the PCB shown in Fig. 5. ❑

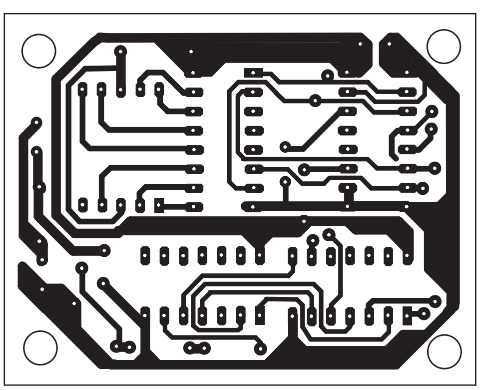


Fig. 7: Actual-size, single side PCB for automatic power control (Fig. 6)

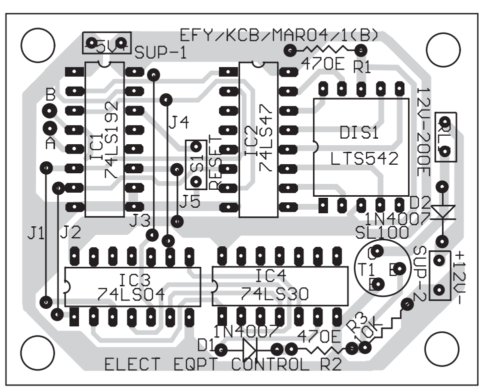


Fig. 8: Component layout for the PCB in Fig. 7

SOUND-OPERATED ON/OFF SWITCH

PRADEEP G.

Most sound-operated remote control devices use condenser microphone as the sensor. Since the microphone senses any sound or vibration, these remote controls may give a false response. The sound is generated by clapping or you can use your voice to activate the remote control. As the tone frequency generated through clapping or voice command is not constant, designing a tuned receiver for an ordinary clap or voice-operated switch is very difficult.

Here we've described a unique sound-operated on/off switch that responds only to a particular frequency of sound (4.5 kHz). A suitable receiver can be easily designed to receive and detect this tone. An electronic circuit is used to generate 4.5kHz audible sound. The circuit works with a sound generated from a distance of up to 4.6 metres (15 feet).

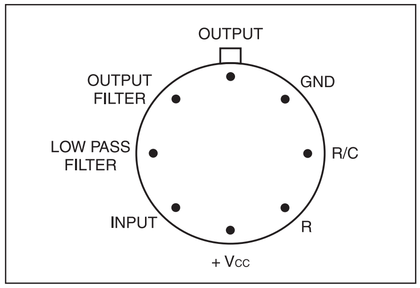


Fig. 4: Top view of IC LM567 (metal package)

The circuit

The sound-operated on/off switch comprises an electronic clapper (sound generator) and a receiver unit to activate the relay.

Electronic clapper. Fig. 1 shows the block diagram of electronic clapper (sound generator). It comprises tone generator, speaker driver and speaker sections.

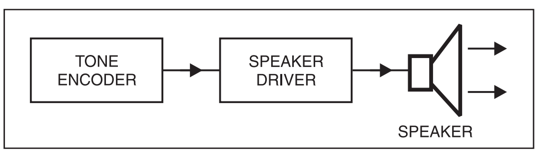


Fig. 1: Block diagram of electronic clapper

The circuit of electronic clapper (Fig. 2) is built around phase-locked loop (PLL) tone decoder LM567 (IC1). The voltage-controlled oscillator (VCO) section inside IC1 is configured to generate 4.5kHz signals. A pnp transistor SK100 (T1) is used to drive an 8-ohm, 0.5-watt loudspeaker (LS1). In order to obtain identical waveshapes of the signals, both the encoder (electronic clapper) and the decoder (receiver) must use the same IC. This is the precise reason why we've used IC LM567 in place of popular timer IC 555 here. When you press switch S1, the elec-

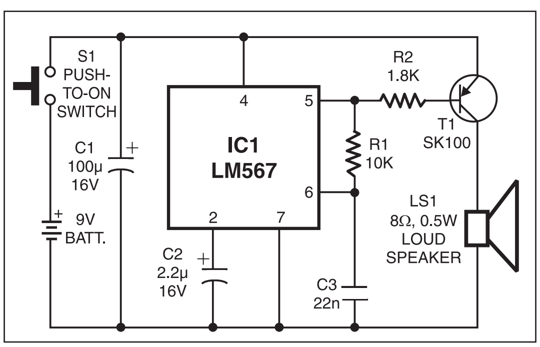


Fig. 2: Circuit of electronic clapper (sound generator)

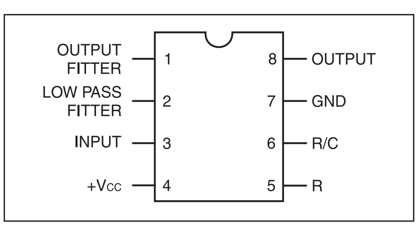


Fig. 3: Top view of IC LM567 in plastic package

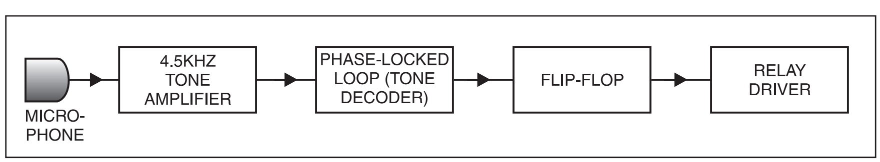


Fig. 5: Block diagram of the receiver unit

PARTS LIST

Semiconductors:	
IC1, IC2	- LM567 PLL tone decoder
IC3	- CD4027 dual JK flip-flop
IC4	- 7809 +9V regulator
T1	- SK100 pnp medium-power transistor
T2, T3	- BC549C npn signal transistor
T4	- BC557 pnp signal transistor
T5	- BC547 npn signal transistor
T6	- SL100 npn medium-power transistor
LED1	- Red LED
D1-D3	- 1N4001 rectifier diode
Resistors (all ¼-watt, ±5% carbon, unless stated otherwise):	
R1, R3, R9, R13, R15, R17,	- 10-kilo-ohm
R2	- 1.8-kilo-ohm
R4	- 4.7-kilo-ohm
R5	- 560-kilo-ohm
R6, R16	- 2.2-kilo-ohm
R7	- 2.7-kilo-ohm
R8	- 680-ohm
R10	- 1-mega-ohm
R11	- 180-kilo-ohm
R12	- 100-kilo-ohm
R14	- 18-kilo-ohm
R18, R19, R20	- 1-kilo-ohm
Capacitors:	
C1, C6, C14	- 100µF, 16V electrolytic
C2, C10	- 2.2µF, 16V electrolytic
C3, C9	- 22nF ceramic disk
C4	- 0.01µF ceramic disk
C5	- 56pF ceramic disk
C7, C8, C16, C17	- 0.1µF ceramic disk
C11	- 4.7µF, 16V electrolytic
C12	- 1µF, 16V electrolytic
C13	- 0.22µF ceramic disk
C15	- 1000µF, 25V electrolytic
Miscellaneous:	
Relay	- 9V, 150-ohm
S1	- Push-to-on tactile switch
LS1	- 8-ohm, 0.5W loudspeaker
Battery	- 9V
	- IC bases
	- Condenser mic

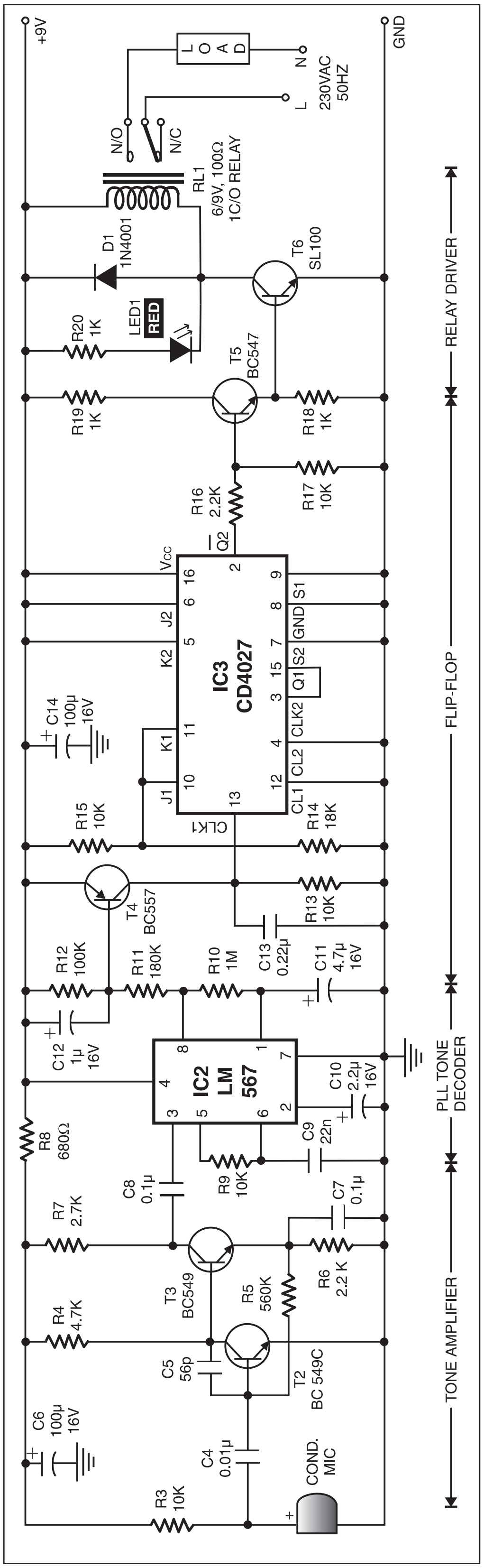


Fig. 6: Receiver circuit

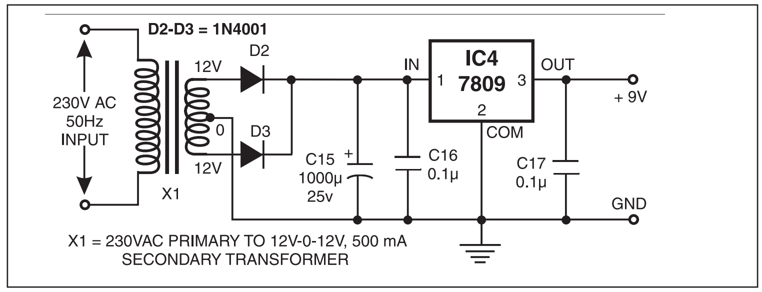


Fig. 7: Power supply circuit

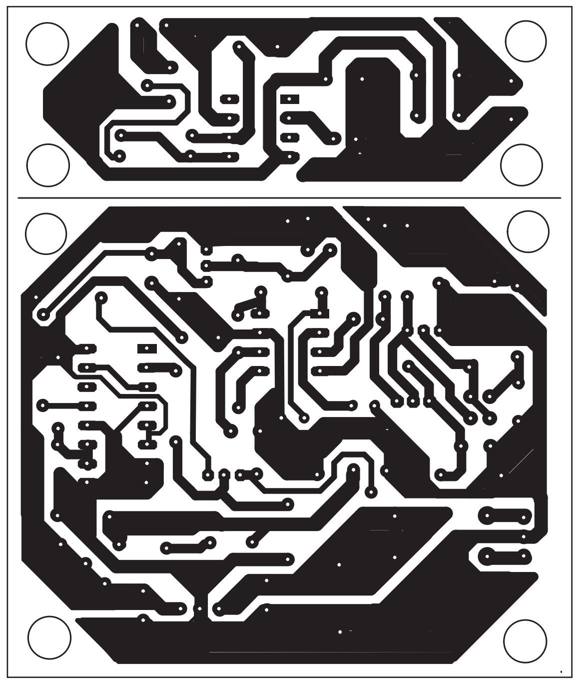


Fig. 8: Actual-size, single-side PCB for sound-operated on/off switch

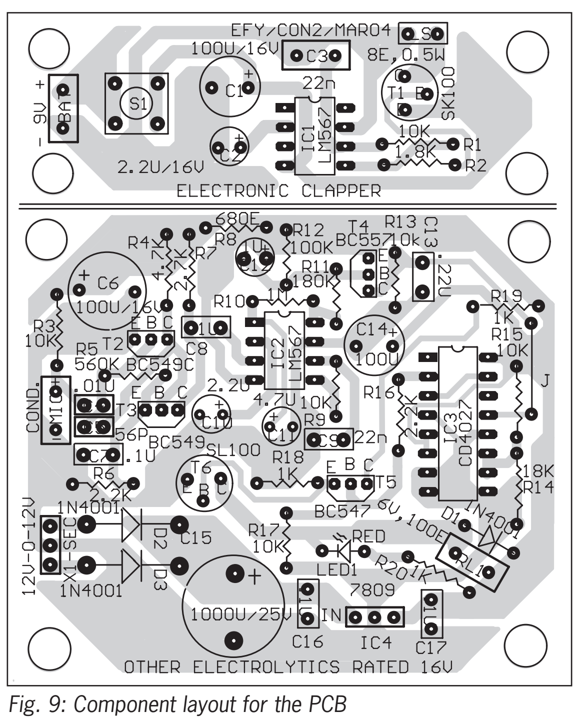


Fig. 9: Component layout for the PCB

tronic clapper generates 4.5kHz sound.

IC LM567 is available in small plastic and metal pakages. The pin configurations of both the packages are shown in Figs 3 and 4, respectively. The IC is a highly stable phase-locked loop with synchronous AM lock detection and power output circuitry. It is primarily used as a tone and frequency decoder where it is required to drive a load whenever a sustained frequency within its detection band is present at its self-biased input. The centre frequency of the band and the output delay are independently determined by external components.

The sailent features of IC LM567 are:

1. Wide frequency range (0.01 Hz to 500 kHz)
2. Highly stable centre frequency
3. Independently controlled bandwidth
4. High out-band signal and noise rejection
5. Low-voltage (5-10V) operation
6. 100mA output current sink capability
7. Inherent immunity to false

signals

Receiver unit. Fig. 5 shows the block diagram of the receiver unit. It comprises condenser microphone, 4.5kHz tone amplifier, PLL (tone decoder), flip-flop and relay driver stages. Fig. 6 shows the receiver circuit.

Tone amplifier. When you press switch S1, the electronic clapper generates 4.5kHz sound. The condenser microphone in the receiver unit converts this sound into an electrical pulse, which is given to a two-stage, high-gain AF preamplifier comprising transistors T1 and T2.

PLL tone decoder. The amplified 4.5kHz signals from the tone amplifier stage are given to PLL tone decoder IC LM567 (IC2) that is tuned for centre frequency of 4.5 kHz. As a result, the output of IC2 goes low.

Flip-flop section. The high-to-low pulse from PLL tone decoder is given to the clock input of the dual JK flip-flop wired around CMOS IC CD4027 (IC3). One of the two flip-flops inside IC3 acts as a squarewave shaper. The squarewave pulse generated by this flip-flop is coupled to the second flip-flop of the IC. This eliminates the need for an extra monostable multivibrator IC.

Relay driver. The output of the flip-flop section (IC3) is given to the relay driver, which drives the load connected to N/O contacts of the relay as shown in Fig. 6.

Power supply. Fig. 7 shows the power supply circuit for the receiver unit. The mains AC supply is stepped down by transformer X1. The output of the secondary transformer is rectified by a full-wave rectifier comprising diodes D1 and D2 and filtered by capacitor C15. The regulated 9V from regulator 7809 (IC4) powers the entire receiver circuit.

Construction

Assemble the electronic clapper and the receiver circuits on two separate PCBs. Check all the connections thoroughly. Connect a 9V battery to the clapper circuit and 9V regulated power supply to the receiver circuit. Since IC LM567 works off a maximum of 10 volts, a 9V regulated power supply is recommended.

Now if you press switch S1 momentarily, the clapper produces a sharp audio tone to energise the relay in the receiver circuit to activate the relay/load connected via relay contacts. To deactivate the relay, again press clapper switch S2.

An actual-size, single-side PCB for the sound-operated on/off switch comprising electronic clapper, receiver and power supply circuits is shown in Fig. 8 with its component layout in Fig. 9. The combined PCB can be cut along the double line to separate the clapper and receiver sections.

Note. ST Microelectronics CD4027 IC is recommended for momentary toggle operation in the receiver unit. ❑

DIGITAL CLOCK USING DISCRETE ICs

A. KANNABHIRAN &
R. JEYARAMAN

This digital clock can be easily constructed using readily-available ICs and components.

The block diagram of the digital clock is shown in Fig. 1. The basic 1Hz clock pulse signal is obtained from the clock pulse generator using a 4.194304MHz crystal. It is divided by 60 by the second's section to produce one clock pulse every minute, which is further divided by 60 to produce one clock pulse every hour. Both the seconds and minutes sections use divide-by-10 and divide-by-6 counters. The clock pulse from the minute's section is applied to the hour's section, which is a divide-by-12 counter to control the hour and AM/PM indication with the help of the code converter circuit and J-K flip-flop. The outputs of all the counters are displayed on 7-segment displays after suitable decoding.

Fig. 2 shows the circuit diagram of digital clock with AM and PM indication. The heart of the circuit is the precision 1-second oscillator section that is built around 14-stage counters CD4060 (IC1 and IC2). The clock accuracy depends upon the 1-second oscillator, which divides the crystal frequency (4.194304 MHz) by 16,348 to output 256 Hz at pin 3 of IC1, which is further divided by 256 to output one pulse per second at pin 14 of IC2. Resistors R1 and R2 are biasing and power-limiting resistors, respectively.

The one-second pulse is applied to the clock input of decade counter 74LS90 (IC3), which is a 4-stage ripple counter containing a master/slave flip-flop acting as a divide-by-2 counter and three flip-flops connected as a divide-by-5 counter. Clock input $\overline{CP1}$ of the divide-by-5 section must be externally connected to Q0 output of the divide-by-2 section. $\overline{CP0}$ clock input of the divide-by-2 section receives the clock signal from the oscillator output and a BCD count sequence is produced.

Q0 through Q3 outputs of the decade counter (IC3) are connected to A0 through A3 input pins of the BCD to 7-segment decoder/driver 74LS47 (IC9), respectively. IC9 accepts the 4-line input data, generates their complements internally and decodes the data with seven AND/OR gates having open-collector outputs to drive LED segments directly. The 'a' through 'f' outputs of IC 74LS47 (IC9) are connected to the corresponding inputs of 7-segment display DIS1. All the 7-segment displays work in the same fashion. Resistors R3 through R8 are used as current-limiting resistors for displays DIS6 down to DIS1, respectively. Each display comprises seven light-emitting diodes with their common anodes connected together. This configuration is known as the common-anode, 7-segment display.

ICs 74LS90 (IC3) and 74LS92 (IC4) are cascaded to produce units' and tens' digits of the seconds' display. Decade counter IC3 is reset to start counting from 0 after ninth count. Pin 11 (Q3) of IC3 is connected to clock input pin 14 ($\overline{CP0}$) of IC4. After ninth count, Q3 output of IC3 goes from high to low and provides a clock signal to $\overline{CP0}$ (pin 14) of IC4.

IC4 contains a flip-flop acting as a divide-by-2 counter and three flip-flops connected as a divide-by-6 counter. After fifth

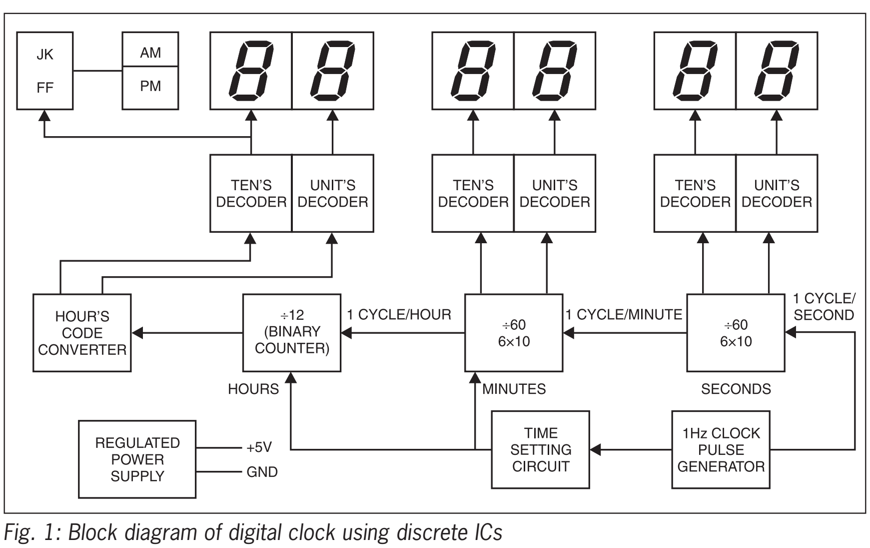


Fig. 1: Block diagram of digital clock using discrete ICs

PARTS LIST

Semiconductors:	
IC1, IC2	- CD4060 14-stage counter/ divider and oscillator
IC3, IC5	- 74LS90 decade counter
IC4, IC6	- 74LS92 divide-by-12 counter
IC7	- 74LS93 divide-by-16 counter
IC8	- CD4017 5-stage Johnson counter
IC9-IC14	- 74LS47 BCD to 7-segment decade counter/driver
IC15	- 74LS76 dual JK flip-flop
IC16, IC17	- 74LS04 hex inverter
IC18, IC19	- 74LS08 quad two-input AND gate
IC20	- 74LS32 quad two-input OR gate
IC21	- 7805, 5V regulator
T1-T4	- BC548 npn transistor
D1	- 1N4148 switching diode
D2-D5	- 1N4007 rectifier diode
DIS1-DIS6	- LTS542 common-anode 7-segment display
LED1	- Green LED
LED2	- Red LED
Resistors (all ¼-watt, ±5% carbon, unless stated otherwise):	
R1	- 470-kilo-ohm
R2	- 1.2-kilo-ohm
R3-R8, R16	- 220-ohm
R9-R14	- 2.2-kilo-ohm
R15	- 10-kilo-ohm
Capacitors:	
C1	- 22pF ceramic disk
C2	- 47pF ceramic disk
C3	- 1000µF, 25V electrolytic
C4	- 0.1µF ceramic disk
Miscellaneous:	
XTAL	- 4.194304MHz
S1-S3	- Push-to-on switch
X1	- 230V AC primary to 6V-0-6V, 300mA secondary transformer

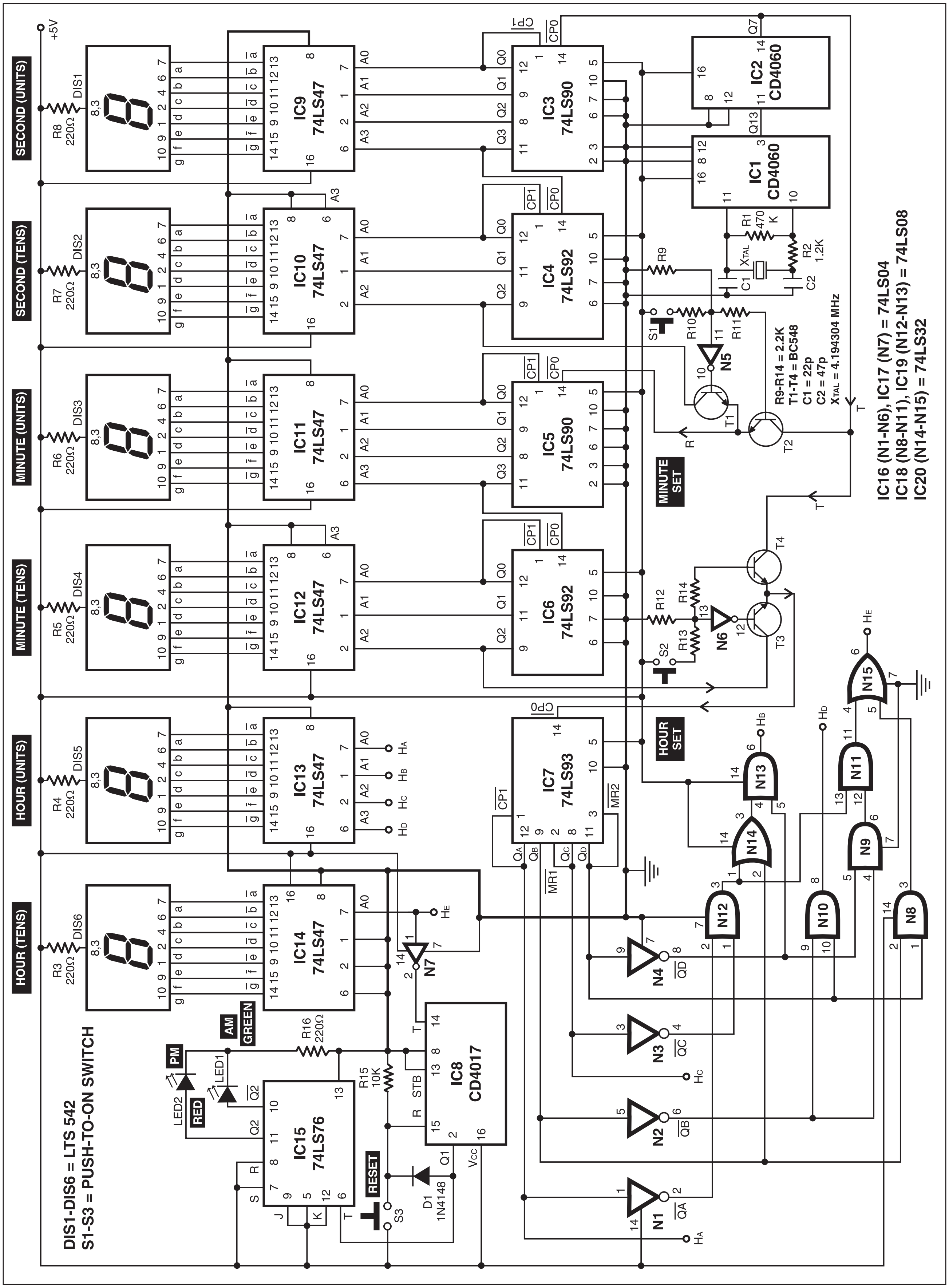


Fig. 2: Circuit diagram of digital clock using discrete ICs

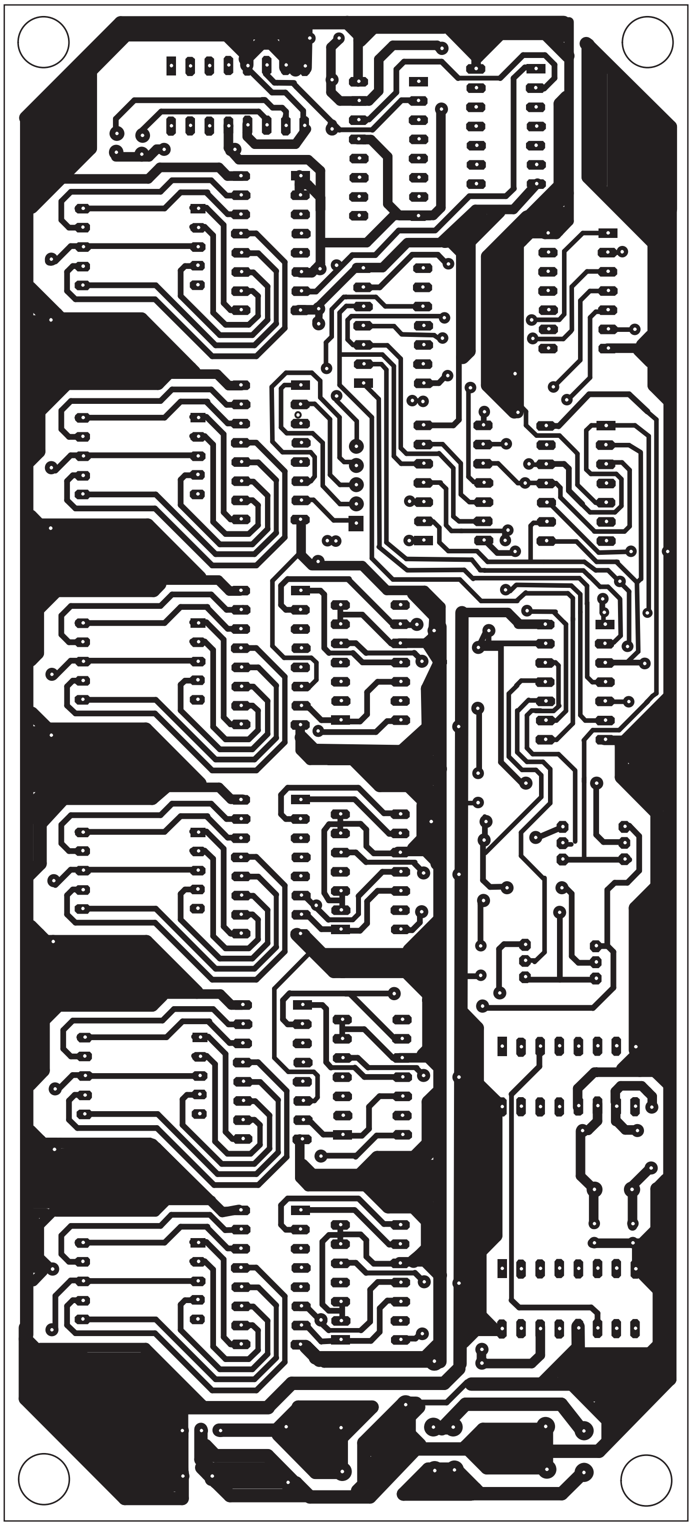


Fig. 3: Actual-size, single-side PCB for the digital clock using discrete ICs

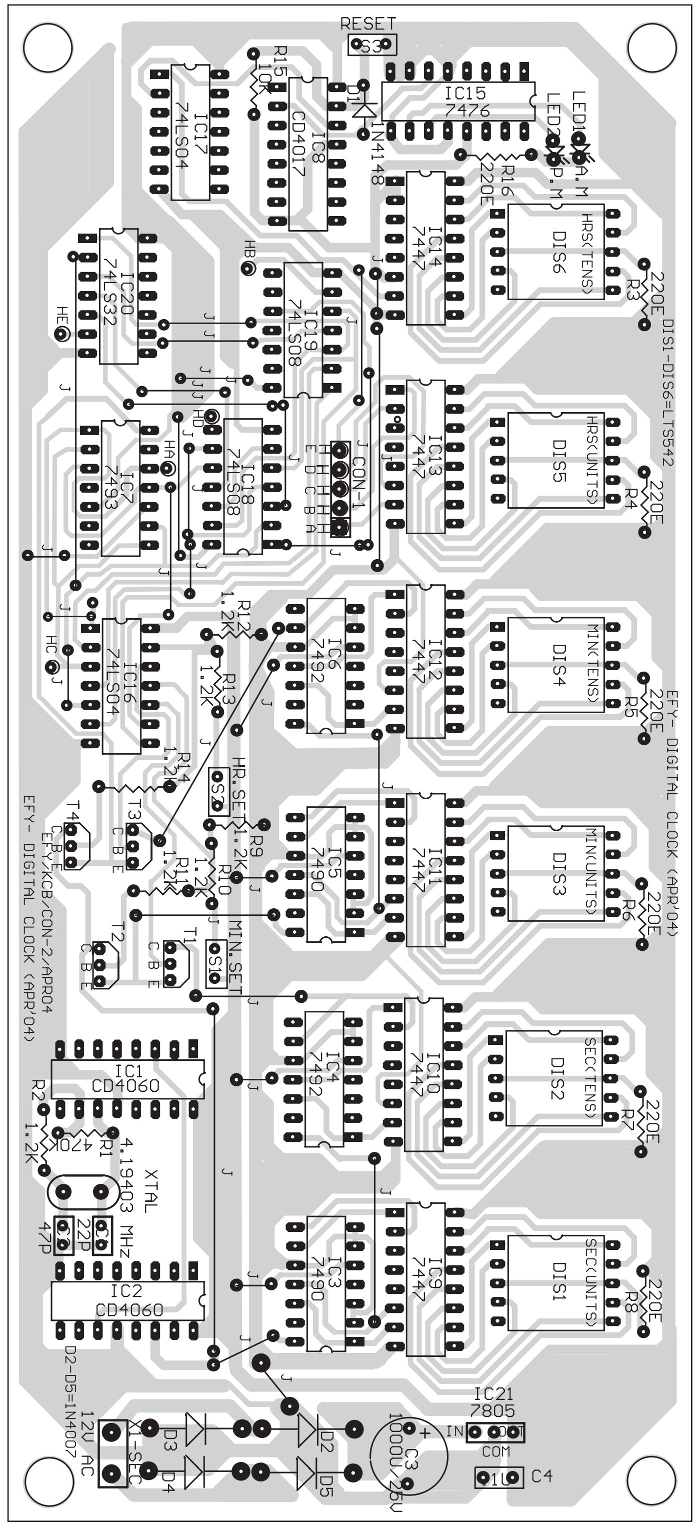


Fig. 4: Component layout for the PCB

count, Q2 output of IC4 goes from high to low and IC4 starts counting from 0.

The next clock pulse resets the seconds section after it counts up to 59 seconds and provides a clock pulse to the minutes section. IC5 and IC6 are used for generation of units' and tens' digits of the minutes' display with the help of IC11 and IC12, respectively. Q2 output of IC4 is connected to the clock input (CP0) of IC5 through transistor T1. Resistor R9 is pulled low and the high output of inverter N5 provides forward bias to transistor T1. Q2 output of IC4 is available at pin 14 of IC5 through the low-resistance path of transistor T1. The emitters of both transistors T1 and T2 are connected to pin 14 of IC5.

Switch S1 is used for setting the minutes time. When switch S1 is pressed, transistor T1 is reverse biased and transis-

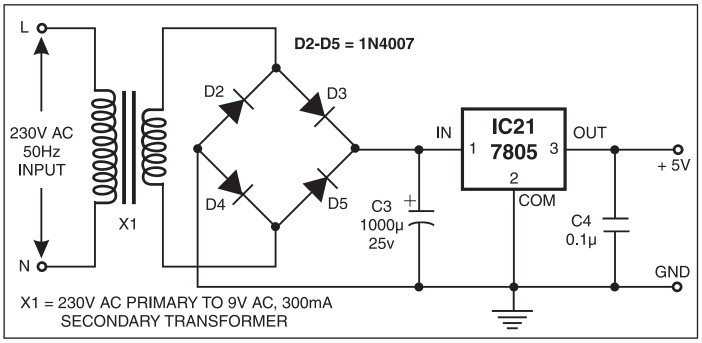


Fig. 5: Power supply circuit

Binary Input Conversion into 5-bit Code

Binary input to code converter				Converted output from code converter					Display	
Q_D	Q_C	Q_B	Q_A	H_E	H_D	H_C	H_B	H_A	Hrs. (Tens)	Hrs. (Units)
0	0	0	1	0	0	0	0	1	0	1
0	0	1	0	0	0	0	1	0	0	2
0	0	1	1	0	0	0	1	1	0	3
0	1	0	0	0	0	1	0	0	0	4
0	1	0	1	0	0	1	0	1	0	5
0	1	1	0	0	0	1	1	0	0	6
0	1	1	1	0	0	1	1	1	0	7
1	0	0	0	0	1	0	0	0	0	8
1	0	0	1	0	1	0	0	1	0	9
1	0	1	0	1	0	0	0	0	1	0
1	0	1	1	1	0	0	0	1	1	1
0	0	0	0	1	0	0	1	0	1	2

tor T2 is forward biased. Forward-biased transistor T2 provides a low-resistance path for 1Hz clock signal and, at the same time, transistor T1 blocks the signal from Q2 output of IC4.

The minutes section works the same way as the seconds section. After 59th count, the next clock pulse resets the minutes section and provides a clock pulse (through transistor T3) to clock input pin 14 of IC 74LS93 (IC7) of the hours section.

IC 74LS93 is a 4-bit binary counter that consists of four master/slave flip-flops which are internally connected as a divide-by-2 counter section and a divide-by-8 counter section. Each section has a separate clock input, which initiates counting on receiving a high-to-low clock pulse. Q_A output of the divide-by-2 section must be externally connected to $\overline{CP1}$ (pin 1) clock input of the divide-by-8 counter section. The input count pulse is applied to $\overline{CP0}$ (pin 14) clock input of the divide-by-2 counter section. This configuration acts as a divide-by-16 counter in normal condition.

Binary counter 74LS93 (IC7) is used as a divide-by-12 counter in this circuit (Fig. 2). This is achieved by connecting its Q_C and Q_D outputs to MR1 and MR2 asynchronous master reset inputs, respectively. When both Q_C and Q_D outputs become 1, the counter is reset to 0000 and as a divide-by-12 counter. It counts the clock pulse and gives the binary output from 0000 to 1011.

The outputs of IC7 are given to the code converter section. The code converter section converts the 4-bit binary input (Q_A through Q_D) into 5-bit code (H_A through H_E) as shown in the table. For inputs from 0001 through 1001, it produces the same outputs. But when inputs are 1010, 1011 and 0000, the code converter section converts these into 10000, 10001 and 10010, respectively. The code-converter circuit comprises NOT gates N1 through N4, AND gates N8 through N13 and OR gates N14 and N15. H_A through H_E outputs of the code converter are simplified by using Karnaugh map as follows:

$H_A = Q_A$

$H_B = \overline{Q_D} . (Q_B + \overline{Q_A} . \overline{Q_C})$

$H_C = Q_C$

$H_D = Q_D . \overline{Q_B}$

$H_E = Q_D . Q_B + \overline{Q_A} . \overline{Q_B} . \overline{Q_C} . \overline{Q_D}$

H_A through H_D outputs of the code converter are connected to 7-segment decoder 74LS47 (IC13) to display the units' digit of hour and H_E is connected to IC14 to display the tens' digit of hour. After ninth count, tens' digit of the hour display becomes '1' (H_E goes high) and units' digit resets to '0.' To display 01.00.00 after 12:59:59, the code converter circuit resets the tens' digit to '0' and the units' digit to '1'.

Edge-trigger flip-flop 74LS76 (IC15) is used for AM and PM indications in conjunction with CD4017 (IC8). H_E output of the code converter controls the AM/PM display. It is connected to clock input pin 14 of IC8 via NOT gate N7.

Every twelve hours, H_E output goes from high to low. The high clock input of IC8 takes its output pin 2 (Q1) high, which, in turn, triggers the flip-flop and resets IC8 via diode D1.

Initially, $\overline{Q2}$ output of IC15 is high as Q2 output is low. Thus AM LED1 (green) is on. After twelve hours, the first clock pulse turns Q2 high and its complement $\overline{Q2}$ goes low. As a result, PM LED2 (red) glows.

Again after twelve hours, H_E output of the code converter goes from high to low and gives another clock pulse to the flip-flop with help of CD4017. Now Q2 output goes low and its complement $\overline{Q2}$ becomes high. Thus AM LED glows.

Push-to-on switches S1 and S2 are used to manually set minute and hour, respectively. The 1Hz clock from the output of IC2 is used to advance the minutes counters (IC5 and IC6) or the hours counter (IC7) at a fast rate by pressing switch S1 (of the minutes' set) or switch S2 (of the hours' set). Switch S3 is used for initial resetting of IC8.

The power supply circuit is shown in Fig. 5. The AC mains supply is stepped down by transformer X1 to deliver a secondery output of 9V AC, 300 mA. The output of the transformer is rectified by a full-wave rectifier comprising diodes D2 through D5. Capacitor C3 acts as a filter to eliminate ripple. Regulator 7805 (IC21) provides regulated 5V power supply to the digital clock circuit.

An actual-size, single-side PCB for the digital clock is shown in Fig. 3 and its component layout in Fig. 4. H_A through H_E inputs of ICs 13 and 14 have been terminated on Con-1 and suitably marked on the PCB. These pins are to be connected to code converter outputs with identical marking and terminated on pads using jumpers. ❑

A BIDIRECTIONAL VISITORS COUNTER

MILIND GUPTA

This counter can be used to know the number of visitors present in a room at any given time. It is useful for places such as movie halls, buildings and offices. To keep the cost low, it uses a simple calculator instead of a counter-and-display circuit. The calculator can be used as a normal calculator any time by plugging it off from the circuit. All the components are readily available in the market and the circuit is easy to build.

Circuit description

Two transmitter-receiver pairs are used at the passage: One pair comprising light source A (transmitter) and light-dependent resistor LDR1 (receiver) is installed at entry side of the passage, while the other pair comprising light source B (transmitter) and LDR2 (receiver) is installed at exit side of the passage. Light from the two light sources (torches) should continuously fall on the respective light-dependent resistors (LDRs), so proper orientation of light beams and LDRs is essential. Fig. 1 shows the transmitter-receiver set-up at the entrance-cum-exit of the passage.

Fig. 2 shows the circuit of the bidirectional visitors counter, wherein sections A and B are light-detection circuits. The logic control circuit is built around AND gate IC3, NOR gate IC4 and flip-flops IC5 and IC6. The time delay circuit comprises timers IC11 and IC12. Optocouplers IC7 through IC10 are used to isolate the calculator from the circuit voltage.

The power supply circuit is shown between sections A and B in Fig. 2. The mains AC supply is stepped down by transformer X1 to 12V AC and the same is rectified by a bridge rectifier comprising diodes D1 through D4 and then filtered by capacitor C1. The regulated 9V from regulator IC 7809 (IC1) powers the entire circuit.

Working

Initially, when the power is switched on, flip-flops IC5 and IC6 are in reset state because of power-on-reset components comprising resistor R5 and capacitors C3 and C4. Thus transistors T3 and T4 are initially in cut-off state. At the same time, transistor T5 also is in cut-off state. In brief, when power is switched on, all the terminal keys including '1' , '+', '–' and '=' of the calculator remain open.

The two similar sections A (comprising LDR1, transistor T1 and NAND gate N1) and B (comprising LDR2, transistor T2 and NAND gate N2) detect the interruption of light and then generate clocks at pin 3 of NAND gate N1 and pin 4 of NAND gate N2, respectively.

When nobody is passing through the passage, light falls on both LDR1 and LDR2, which thus have low resistance. Since the resistance of LDR1 is low, transistor T1 conducts and the voltage at its collector is low. This low voltage is fed to NAND gate N1, which gives a high output at its pin 3. As the outputs of NOR gates N7 and N8 are low, the LED inside optocoupler IC8 is in 'off' state and the positive terminals (+) of the calculator remain open. Similarly, the resistance of LDR2 is also low, transistor T2 conducts and the voltage at its collector is low. This low voltage is further given to NAND gate N2, which gives a high output at its pin 4. As the outputs of NOR gates N9 and N10 are low, the LED inside optocoupler IC9 is in 'off' state and the negative key (–) of the calculator remains open.

Now if somebody enters the passage (to room/hall), first light A is interrupted and then light B. When light A is interrupted, the resistance of LDR1 increases to provide a low output at pin 3 of NAND gate N1. This low voltage is fed to AND gate N3 and NOR gates N7 and N8. Since pin 6 of NOR gate N8 is low, the output of

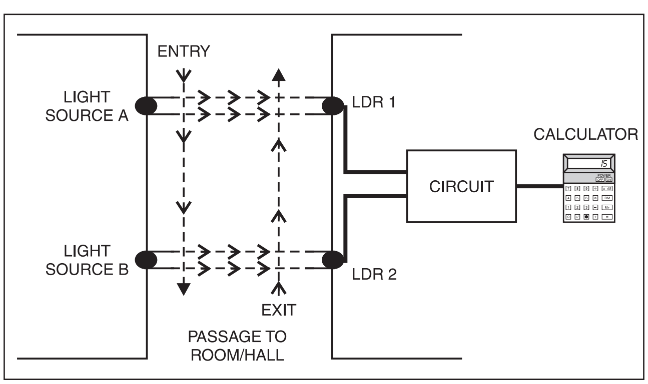


Fig. 1: Light beam set-up at the entrance-cum-exit of the passage

PARTS LIST

Semiconductors:	
IC1	- 7809, 9V regulator
IC2	- CD4093 quad 2-input Schmitt trigger
IC3	- CD4081 quad 2-input AND gate
IC4	- CD4001 quad 2-input NOR gate
IC5, IC6	- CD4013 dual D flip-flop
IC7-IC10	- MCT2E optocoupler
IC11, IC12	- NE555 timer
T1-T5	- BC547 npn transistor
D1-D4	- 1N4007 rectifier diode
D5, D6	- 1N4148 switching diode
Resistors (all ¼-watt, ±5% carbon, unless stated otherwise):	
R1, R3	- 3.3-kilo-ohm
R2, R4	- 39-kilo-ohm
R5-R9, R12, R13, R18, R19	- 1-kilo-ohm
R10, R11	- 1.2-kilo-ohm
R14, R16	- 100-kilo-ohm
R15, R17, R20, R21	- 10-kilo-ohm
VR1, VR2	- 200-kilo-ohm preset
LDR1, LDR2	- Light-dependent resistor
Capacitors:	
C1	- 1000µF, 35V electrolytic
C2, C5, C8	- 0.1µF ceramic disk
C3, C4	- 100pF ceramic disk
C6, C9	- 10µF, 25V electrolytic
C7, C10	- 0.01µF ceramic disk
Miscellaneous:	
S1	- On/Off switch
X1	- 230V AC primary to 0-12V, 300/500 mA secondary transformer
	- Calculator
	- Light sources (2 torches)

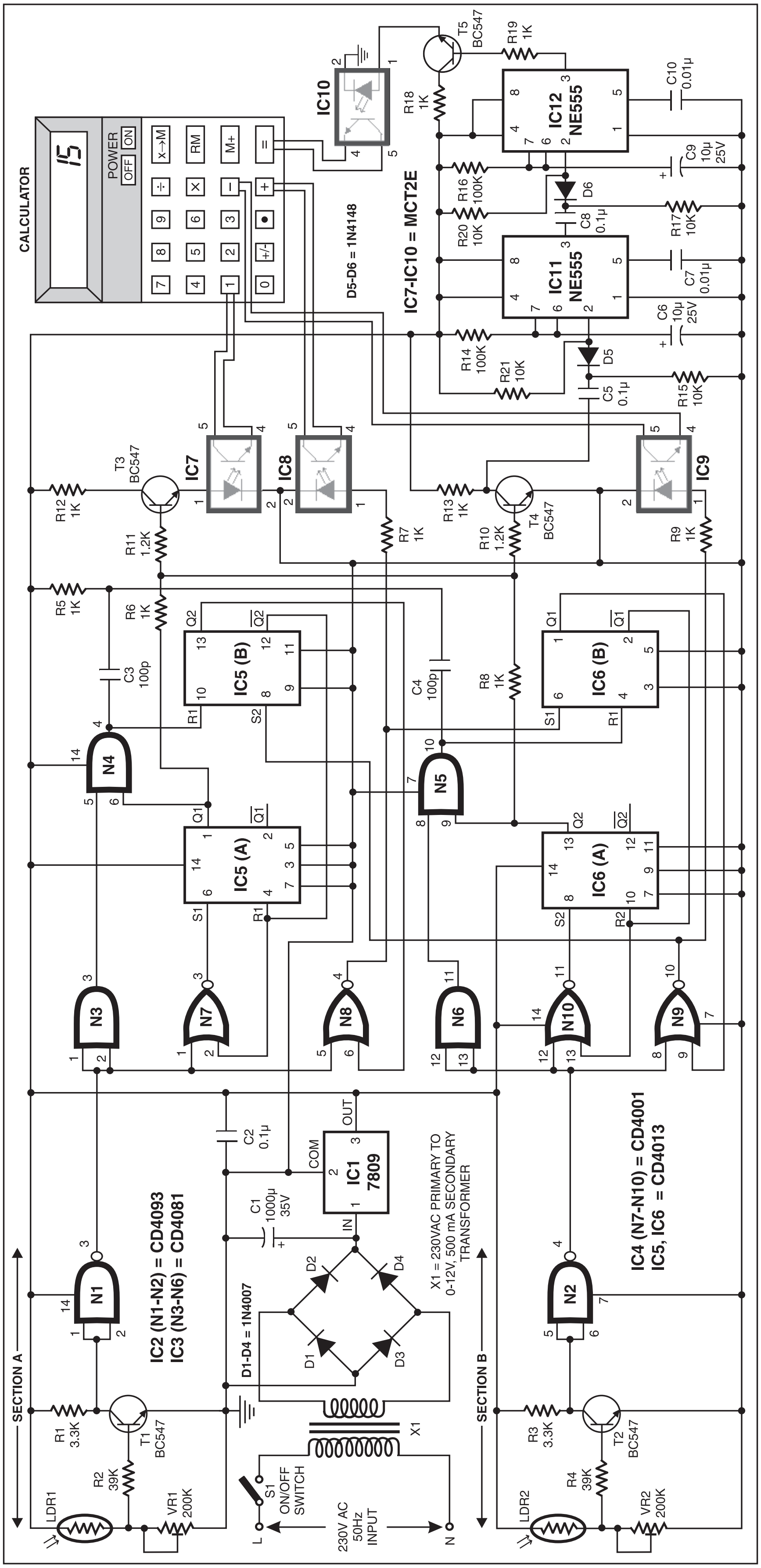


Fig. 2: Circuit diagram of bidirectional visitor counter

N8 goes high, setting Q1 output (pin 1) of IC6(B) to high state. Simultaneously, IC8 activates and its internal transistor shorts the '+' key of the calculator.

When the person moves further to interrupt light B, the resistance of LDR2 increases to provide a low output at pin 4 of NAND gate N2. Since IC6(B) is in set condition and pin 13 of NOR gate N10 is low, its output goes high. This, in turn, sets IC6(A) and its Q2 output (pin 13) is driven high. As a result, transistor T3 conducts to activate IC7 and its internal transistor shorts terminals of key '1' of the calculator. The high output of IC6(A) also triggers monostable IC11 via transistor T4, capacitor C5 and diode D5, which, in turn, triggers monostable IC12 after approximately one second. The output of IC12 activates IC10 via transistor T5 and its internal transistor shorts the terminals of the '=' key of the calculator. Thus the '=' key of the calculator shorts about one second after the '1' key shorts.

The output of AND gate N5 is still low because its pin 8 is at low level due to obstruction of light B. As the person moves past light source B, light again falls on LDR2 and the output of N5 goes high. This resets flip-flop IC6(B) to make its $\overline{Q1}$ output (pin 2) high, which, in turn, resets flip flop IC6(A) to make its Q2 output (pin 13) low and hence the output of AND gate N5 again goes low. As a result, transistors T3 and T4 stop conducting and keys '1' and '=' of the calculator get open. Thus the circuit returns to its original state after shorting the '+,' '1' and '=' keys of the calculator and it is ready for another count.

The above explanation can be summarised as follows: When light falling on LDR1 is interrupted first, followed by light falling on LDR2, the calculator keys '+,' '1' and '=' are automatically shorted consecutively. This adds '1' to the total on the calculator, indicating that a person is entering, and upcounting takes place.

Similarly, when somebody exits, first light B is interrupted and then light A. When light B is interrupted, the resistance of LDR2 increases to provide a low output at pin 4 of NAND gate N2. This low voltage is fed to AND gate N6 and NOR gates N9 and N10. Since pin 9 of NOR gate N9 is initially low, its output goes high to set IC5(B). Simultaneously, IC9 activates and its internal transistor shorts the '–' terminals of the calculator.

Now as the person crosses light B to interrupt light A, the resistance of LDR1 increases to provide a low output at pin 3 of NAND gate N1. Since IC5(B) is in 'set' condition and pin 2 of NOR gate N7 is low, its output goes high. This, in turn, sets IC5(A) and its Q1 (pin 1) is driven high. As a result, transistor T3 conducts to activate IC7 and its internal transistor shorts the terminals of '1' key of the calculator. The high output of IC5(A) also triggers monostable IC11 via transistor T4, capacitor C5 and diode D5, which, in turn, triggers monostable IC12 after a delay of approximately one second. The output of IC12 activates IC10 via transistor T5 and its internal transistor shorts the terminals of the '=' key of the calculator. Thus the '=' key of the calculator shorts about one second after the '1' key shorts.

The output of AND gate N4 is still low because its pin 5 is low due to obstruction of light A. As the person moves past light beam A, light again falls on LDR1 and the output of AND gate N4 goes high. This resets flip-flop IC5(B) and its $\overline{Q2}$ output (pin 12) goes high, which, in turn, resets flip-flop IC5(A) to make its Q1 output (pin 1) low and hence the output of AND gate N4 again goes low. As a result, transistors T3 and T4 stop conducting and keys '1' and '=' of the calculator become open. Thus the circuit returns to its original state after shorting the '–,' '1' and '=' keys of the calculator and it is ready for another count.

In brief, when light falling on LDR2 is interrupted first followed by light falling on LDR1, the calculator keys '–,' '1' and '=' are automatically shorted consecutively. This substracts 1 from the total on the calculator, indicating that a person is exiting, and downcounting takes place.

The total number of persons present in the room/hall, at any time, can be seen on the display of the calculator.

Construction

An actual-size, single-side PCB for the bidirectional visitor counter, including the power supply (Fig. 2), is shown in Fig. 3 and its component layout in Fig. 4. This circuit can also be assembled on any general-purpose PCB if you don't have the PCB shown in Fig. 3.

Once the circuit has been soldered, connect the calculator. You can use any simple calculator and connecting it to the circuit does not harm it. If you use jumper plugs attached to the calculator for connection to the circuit, you can plug off the calculator from the circuit at any time and use it as a general calculator.

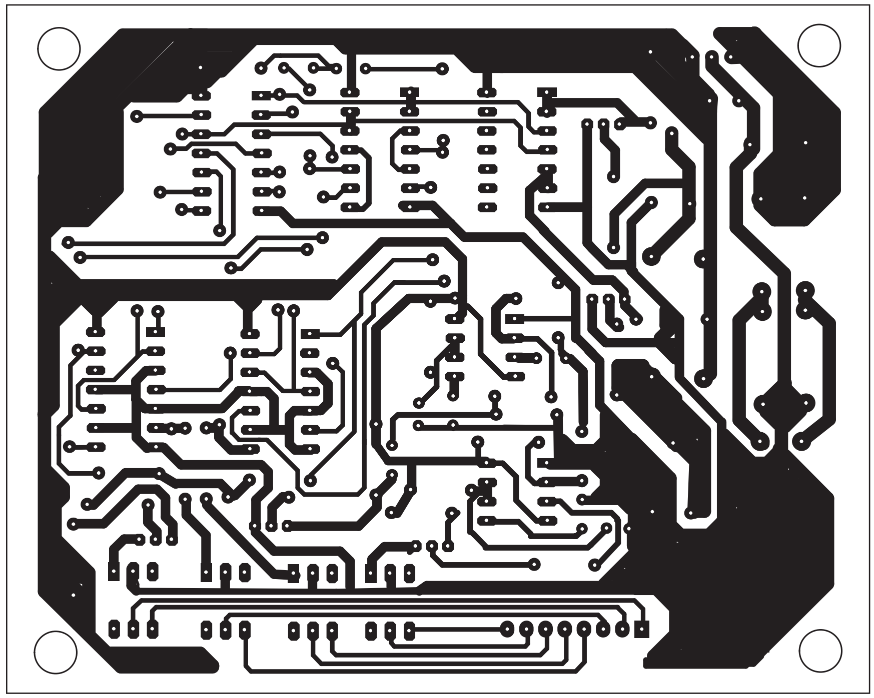


Fig. 3: Actual-size, single-side PCB for bidirectional visitor counter

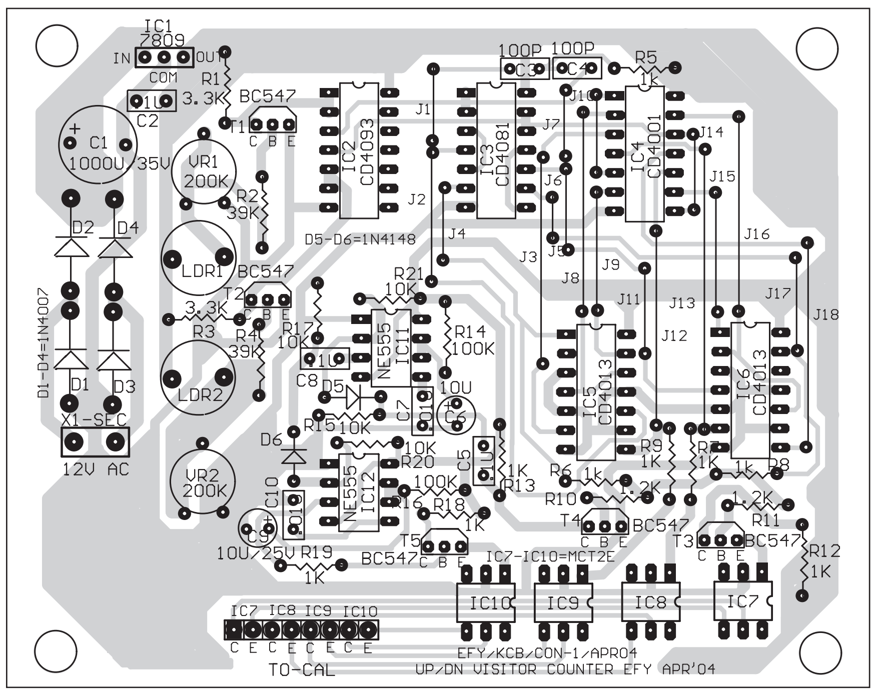


Fig. 4: Component layout for the PCB

To make the connections, open the calculator and solder fine wires on the two contacts beneath the '1', '=', '–' and '+' keys each. Make a hole on the back side of the calculator case to allow the wires to come out. Close the calculator after putting back the contact pads and the keys. Now switch on the calculator.

Use a digital multimeter to measure the DC voltage across the wires coming from the '1' key and identify their polarity. Now connect the negative-polarity wire to emitter pin 4 of optocoupler IC7 and the positive-polarity wire to its collector pin 5. Similarly, find out the polarities of wires connected to the '+', '–' and '=' keys and connect them to optocouplers IC8, IC9 and IC10, respectively. The circuit is now ready for use.

As shown in Fig. 1, mount the assembled system on the entrance-cum-exit of the passage to the room/hall where visitors are to be monitored. Use a 9V battery for back-up.

Testing

To test the circuit, expose both the LDRs to light sources of the same intensities. Switch on power to the circuit and measure the voltage at the collector of transistor T1 with respect to ground. If the voltage is more than 5V, set it to approximately one volt (or less) by adjusting preset VR1. Now obstruct light and measure the voltage again. If the voltage is below 5V, adjust it to more than 5V using VR1.

For 200-kilo-ohm VR1, the resistance of the LDR in no-light condition should be higher than 100k and under light, it should be as low as 10k or so. Otherwise, change the value of VR1 accordingly.

Now calibrate transistor T2 as explained above. Once the adjustments are completed, switch off power to the circuit and again switch it on after 5 to 10 seconds. Now switch on the calculator and press the AC (all clear) button on it. The display will show '0'.

Momentarily obstruct light A followed by light B. Once the path for light B is clear, the calculator display should show '1.' On repeating the procedure, '1' is added to display '2,' indicating that two people have entered. In the same manner, the displayed figure will increase by '1' for every obstruction of lights A followed by light B. This indicates that up-counter is working well.

For testing down-counting, press the AC (all clear) button on the calculator. The display will show '0'. Now momentarily obstruct light B followed by light A. Once the path for light A is clear, the calculator will display '–1'. On repeating the procedure, '1' is substracted from the existing total (–1) to display '–2', indicating that two persons have exited. In the same manner, the displayed figure will decrement by '1' for every obstruction of lights B and A in that order. This indicates that the down-counter is working well.

If the counter is not working properly, check the soldering for any loose connection. Check the connections to the calculator by manually shorting the wires of the calculator.

Precautions

To make sure that ambient light doesn't fall on the LDRs, house the LDRs in black tubes pointing towards the light sources.

The ICs should be soldered carefully. It is better to use IC bases and plug-in the ICs later. The solder to the IC pins should not be dry or loose.

To solder wires to the calculator, use a fine soldering tip. ❑

Readers' comments

Q1. we are getting continuous pulses on the calculator but except '0,' nothing is being displayed on the calculator. Section A of the circuit is responding to the obstructions in the path of the laser beam but section B is not responding. Why so?

Praveen Chowdhary
Through e-mail

The author, Milind Gupta, replies:

A1. Sections A and B are exactly the same. So the problem could be that your light-dependent resistor (LDR) is not giving enough change in resistance due to the obstruction. You can check the voltage at the LDR by varying preset VR2 (200k) and see whether the change is big enough to cause switching in the transistor. If not, replace LDR2.

As regards pulses in the calculator, do you mean to say that you are getting continuous pulses without any light obstruction? If yes, one of your 555 ICs might be wired as an astable rather than a monostable, i.e., pins 2 and 6 are shorting. If you are getting appropriate pulses but no increment in the calculator, it's the problem of the calculator. You should have selected a calculator that accepts the key input as described.

PROGRAMMER FOR 89C51/89C52/ 89C2051 MICROCONTROLLERS

S. ANANTHI, K. PADMANABHAN, P. ARVIND KUMAR, M. SHYAM, M. SHAKTIVEL

The 8051 family of microcontrollers, initially introduced by Intel, are now offered by a host of manufacturers such as Atmel, Philips and Dallas. Atmel 89C51, 89C52 and 89C2051 microcontrollers happen to be the workhorses today. These microcontrollers contain internal flash memory (EEPROM), which makes it possible to store the program internally inside the chip. For developing any application using these microcontrollers, one needs to have access to a programmer board.

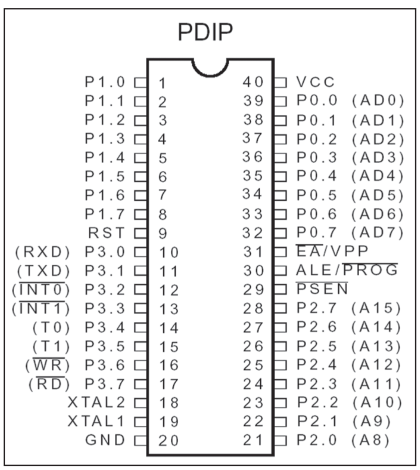


Fig. 1: Pin assignments for 89C51/52

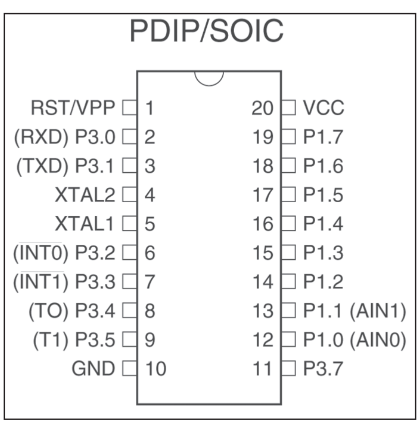


Fig. 2: Pin assignments for 89C2051

Here is a simple programmer circuit that can be used to program 89C51, 89C52 and 89C2051 microcontrollers (refer Figs. 1 and 2 for their pin assignments). The fancy here is that the programmer itself deploys an 89C51 chip containing the necessary firmware.

Operational modes

The programmer can operate in any of the following two modes:

1. Direct keyboard-entry mode
2. Serial-port interface mode

Direct keyboard-entry mode. In this mode, the programmer is connected to an IBM PC keyboard. The program data is entered byte by byte and the same gets programmed into the microcontroller which is inserted into the appropriate ZIF socket on the programmer board. The bitwise contents of any given location of an already programmed microcontroller can also be read and displayed on an 8-LED display provided on the programmer. There is also a provision for erasing the contents of an already programmed device. This mode is useful for developing simple applications by users who do not have ready access to a PC and want the code to be manually entered without the hassle of a computer.

Serial-port interface mode. The programming board can be connected to Com port of a PC, using a 3-wire cable, terminating on a 9-pin D connector on the board. A simple serial port program run on the PC starts the dialogue and you can program an 89C51, 89C52 or 89C2051 microcontroller in this programming mode by using program data in an ASCII file on the PC as well as byte-by-byte from the PC's keyboard. The locking of the entered code in the micro-

PARTS LIST

Semiconductors:	
IC1	- 89C51 microcontroller
IC2	- MAX232 RS-232 level converter
IC3	- 74LS04 hex inverter
IC4	- 7805 +5V regulator
T1, T2	- BC548 npn transistor
T3	- 2N2907 pnp transistor
T4	- BC557 pnp transistor
D1	- 1N4148 switching diode
ZD1	- 5V zener diode
LED1-LED8	- Red LED
LED9	- Yellow LED
LED10	- Green LED
Resistors (all ¼-watt, ±5% carbon, unless stated otherwise):	
R1, R5, R8, R9	- 4.7-kilo-ohm
R2-R4, R6, R12	- 10-kilo-ohm
R7	- 390-ohm
R10	- 1-kilo-ohm
R11	- 330-ohm
RNW1	- 4.7-kilo-ohm x 8-resistor network
RNW2	- 1-kilo-ohm x 8-resistor network
Capacitors:	
C1	- 1µF, 16V electrolytic
C2 - C5	- 22µF, 16V electrolytic
C6- C9	- 22pF ceramic disk
C10	- 0.1µF ceramic disk
Miscellaneous:	
X_{TAL1}	- 3.57MHz Crystal
X_{TAL2}	- 12MHz Crystal
S1, S2	- Push-to-on switch
ZIF Socket 1	- 40-pin ZIF socket
ZIF Socket 2	- 20-pin ZIF socket
	- 9-pin 'D' connector
	- 5-pin keyboard connector

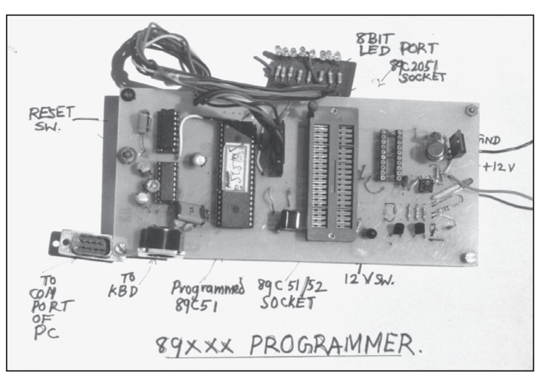


Fig. 3: Authors' working model of programming board

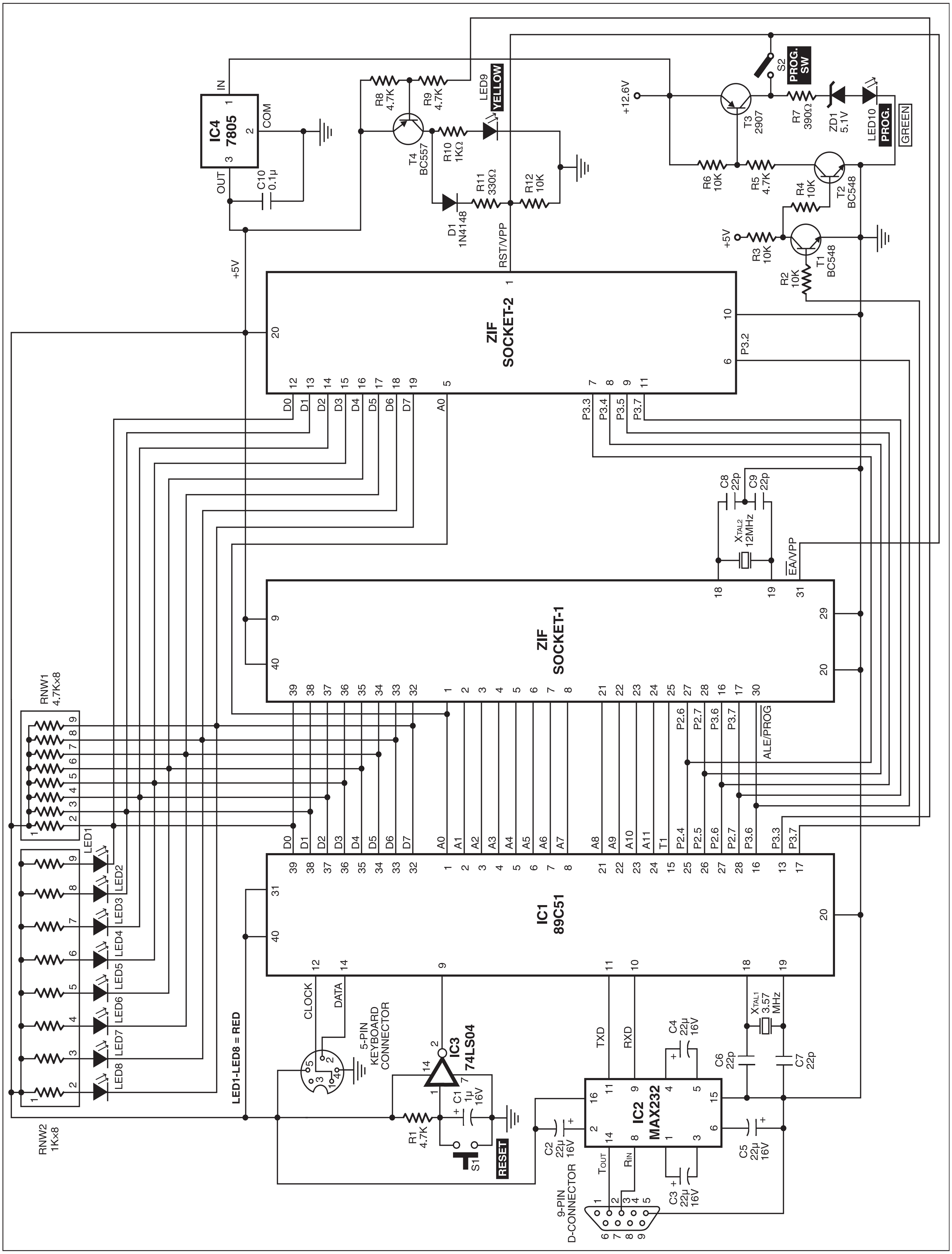


Fig. 4: Circuit diagram of the programming board

controller is also feasible.

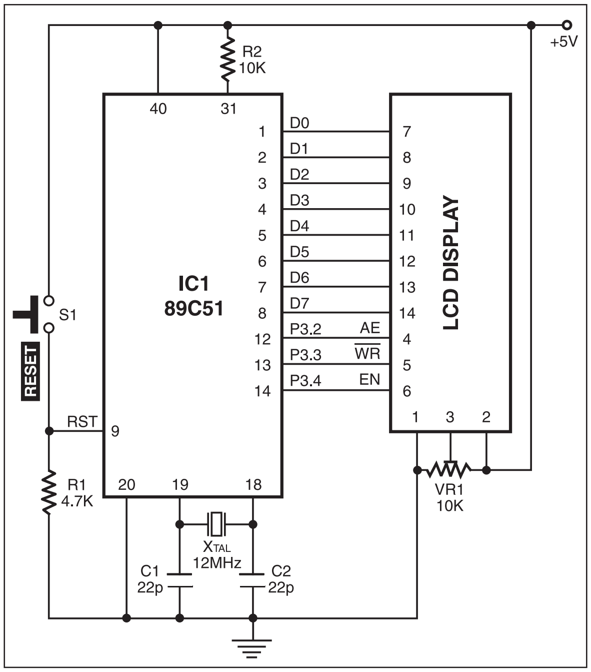


Fig. 5: Interface circuit between an 89C51 microcontroller and LCD module

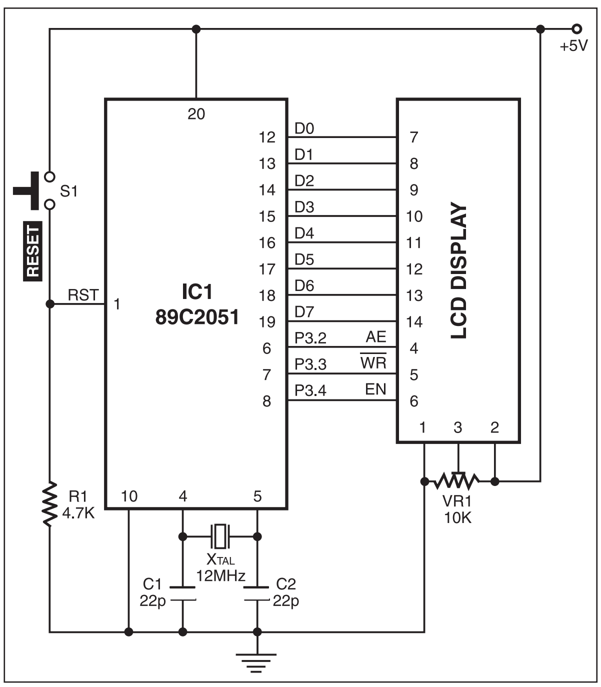


Fig. 6: Interface circuit between an 89C2051 microcontroller and LCD module

Circuit description

Figs 3 and 4 show the authors' working model and the schematic circuit diagram of the programmer board, respectively. In Fig. 4, microcontroller 89C51 (IC1) is preprogrammed for programming other microcontrollers inserted into the appropriate ZIF socket on the right—either into a 40-pin or a 20-pin IC to the socket at a time. The 40-pin ZIF socket-1 and 20-pin ZIF socket-2 are used for inserting an 89C51 (or 89C52) and 89C2051, respectively. IC MAX232 (IC2) is the voltage level converter, which converts TTL-level signals into RS-232C compatible signals and vice-versa. The power-on-reset signal is generated by R1-C1 combination in conjunction with NOT gate 74LS04 (IC3), which provides a high-going reset pulse to pin 9 of IC1. Manual resetting is also possible by shorting capacitor C1 momentarily using push-to-on tactile switch S1.

Port 0 of IC1 serves as the data bus for the IC to be programmed via ZIF socket-1 or ZIF socket-2. This port needs pull-up resistors. Therefore pin numbers 39 down to 32 of port 0 are pulled up to +5V through a (4.7k×8) resistor network RNW-1. These pins are also connected to LED1 to LED8 via current-limiting resistors (1k×8) of RNW-2. Thus, port-0 data can be viewed on these eight LEDs as complement of the actual data, at a specific memory location. A 3.57 MHz crystal (X_{TAL1}) is connected to pins 18 and 19 of IC1, which provides a low baud rate of 1200 for this application. However, a 12MHz crystal (X_{TAL2}) is used for the 89C51/52 IC (being programmed) in ZIF socket-1, to meet its internal timing requirements.

The address bus, data bus and control signals are required for programming a new microcontroller IC. Port 0 and Port 1 pins of IC1 provide 8-bit data bus and eight low-order address lines (A0 through A7), respectively. Four higher-order address lines (A8 through A11) are taken from pins 21 through 24 of port 2 (P2.0 through P2.3). To get A12 address line needed for the 89C52 higher memory IC, pin P3.5 (pin 15) is connceted to pin 25 of ZIF socket-1.

Pins 25 through 28 of IC1 (P2.4 through P2.7) are used for program control functions for the new IC (to be programmed in ZIF socket-1 or ZIF socket-2). The program control signals are given in Table.

When the programming (write) mode is invoked, control pin P2.4 is at logic 0, while pins P2.5 through P2.7 of IC1 are at logic 1 (0111H). During programming mode, the data received from the serial port is routed through the accumulator to the port pins 39 down to 32 of IC1 and hence given to the data pins of the sockets. Data lines D0 through D7 of IC1 are connected to pins 39 down to 32 of the ZIF socket-1 and to pins 19 down to 12 of ZIF socket-2. Pin 30 ($\overline{PROG}$) of ZIF socket-1 or pin 6 (INT0) of ZIF socket-2 is required to be pulsed low for about 100 microseconds during the programming operation. Also, V_{PP} pin 31 of ZIF socket-1 or V_{PP} pin-1 of ZIF socket-2 gets a pulsed 12V supply a few microseconds before the $\overline{PROG}$ pin goes low, which lasts for 2 milliseconds (ms) after $\overline{PROG}$ pin goes high again. This timing is needed by the internal logic of the microcontroller to keep the voltage applied to the oxide gate of the memory for suitable duration, thereby writing into the flash memory by turning a 1 into 0. (Erasing does the opposite of turning a 0 into a 1 bit). The memory bits, after being programmed, will remain non volatile, until the same is erased, which can be done only totally for the entire chip's flash memory.

However, the entire flash memory (and not any one location individually) can be erased by using a proper combination (1000H) of control signals via pins P2.4 through P2.7 by holding PROG pin low for about 10 ms, with 12V applied to V_{PP} pin.

For reading of the signature byte by IC1, control pins P2.4 through P2.7 are at logic 0 (0000H). The signature is present at address 30H of 89C51 (or 89C52) and address 00H of 89C2051 microcontroller.

Transistors T1 through T3 are used for generating the pulsed 12V supply using pins 13 (P3.3) and 17 (P3.7) of IC1. When pin 17 (P3.7) goes low, npn transistor T1 is cut off and its collector voltage rises to drive npn transistor T2 into conduction. As a result the base of pnp transistor T3 goes low, thereby transistor T3 conducts and its collector voltage rises to around

TABLE
Flash Programming Modes

Mode		RST	$\overline{PSEN}$	ALE/$\overline{PROG}$	EA/V_{PP}	P2.6	P2.7	P3.6	P3.7
Write code data		H	L	‾_/‾	H/12V	L	H	H	H
Read code data		H	L	H	H	L	L	H	H
Write lock	Bit-1	H	L	‾_/‾	H/12V	H	H	H	H
	Bit-2	H	L	‾_/‾	H/12V	H	H	L	L
	Bit-3	H	L	‾_/‾	H/12V	H	L	H	L
Chip erase		H	L	‾_/‾(1)	H/12V	H	L	L	L
Read signature byte		H	L	H	H	L	L	L	L

Note. 1. Chip erase requires a 10ms $\overline{PROG}$ pulse.

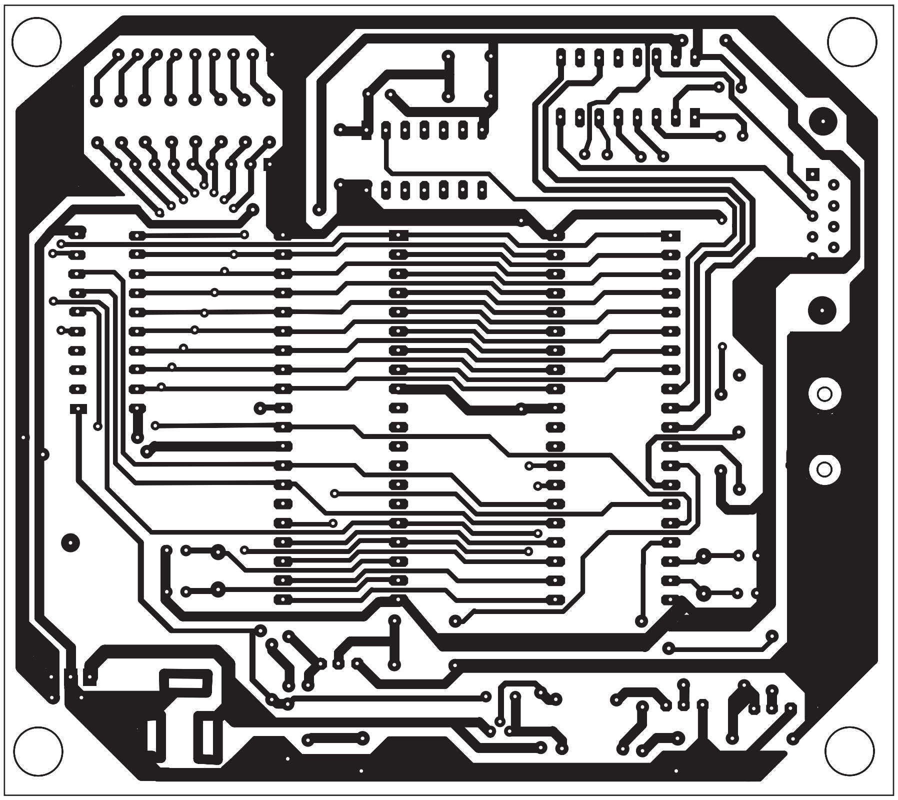


Fig. 7: Actual-size, single-side PCB for the programming board

12V. The collector of transistor T3 is connected to V_{PP} pins of the ZIF sockets 1 and 2. A green LED (LED10) connected to the collector of transistor T3 via current-limiting resistor R7 and zener diode ZD1 lights up to provide a visual indication of the programming voltage.

A voltage of 5V initially and 12V during erasing/programming is applied to V_{PP} pin of the microcontroller IC to be programmed. The availability of 5V at V_{PP} pins of ZIF sockets in absence of programming/erasing pulse period is ensured by circuitry around Transistor T4, in conjunction with pin 13 (P3.3) of IC1. When pin 13 goes low, transistor T4 conducts to provide nearly 5V at V_{PP} pin. LED9 gives visual indication of 5V at V_{PP} pins of sockets 1 and 2. Switch S2 is used for applying 12V pulses to V_{PP} pins of sockets 1 and 2. It protects the IC (to be programmed) from accidentally getting 12V upon power-on and thereby damaging it since 12V should be applied only when control signals are active for erasing or programming functions, and that too for limited duration. If 12V is applied for a longer duration, V_{PP} pin internally gets shorted to ground and further programming is not possible.

IC MAX232 (IC2) is used as an RS-232 level converter. Pins 10 (RXD) and 11 (TXD) of IC1 are connected to pins 3 and 2 of 9-pin D connector, respectively via IC2. The PC keyboard is connected to pins 12 and 14 of IC1.

For compactness, a single 12.6V DC supply is used for the programmer board. While Vpp pulse generation circuitry makes use of 12.6V, however all ICs deployed on the board need regulated 5V DC for their operation. Regulator IC 7805 (IC4) generates 5V supply from the 12.6V DC to meet this requirement.

Port 0 of IC1 connected to the ZIF socket-1 and ZIF socket-2 serves as a bi-directional port for writing (programming) and reading of data to/from the ICs being programmed.

1. In the direct keyboard-entry mode, data entered via the keyboard has to be output to the LEDs for viewing. During write operation control signal P2.4 would be logic 0 while P2.5 through P2.7 would be at logic 1 (0111H). For reading data from the programmed IC, port 0 is converted into an input port by outputting FF hex before outputting the control signals (0011H) via pins P2.4 through P2.7 respectively. Thus, the function of port 0 is bidirectional.

2. In the serial-port interface mode, data transfer from PC to programming board occurs serially and after two bytes of ASCII code are received at the programming board, the same are converted into

DATA.BAS

```
input “Filename= “;q$
 open q$ for random as #3 len=1
 field #3,1 as m$
open “Com1:1200,n,8,1,cs,ds,cd” as #1
 flag=0
pause=false:on error goto 9000
open “scrn:” for output as #2
OPEN “CAPTURE.DAT” FOR OUTPUT AS #4
locate ,,1
xoff$=chr$(19):xon$=chr$(17)

510 n$=inkey$: if n$=“s” or n$=“S” then flag=1 :
goto 800
 IF N$=”Y” THEN FLAG2=1
520 if n$<> “” then print #1,n$;:
if eof(1) then 510
570 if loc(1)>128 then pause=true:print #1, xoff$:
 n$=input$(loc(1),#1)
 lfp=0
630 lfp=instr(lfp+1,n$,chr$(10))
 if lfp>0 then mid$(n$,lfp,1)=” “:goto 630
 print #2,n$;
 IF FLAG2=1 THEN PRINT #4, N$;
 if loc(1)>0 then 570
 if pause then pause=false: print #1,xon$;
 goto 510
800 for kk=1 to 30000:next kk:get #3
 print #1,m$; : rem print m$;
 get #3: print #1,m$; :rem print m$;
830 if eof(1) then 830
 :REM ;data received
835 NUMB=NUMB+1 :REM;no. of bytes
 n$=input$(loc(1),#1)
 lfp=0
850 lfp=instr(lfp+1,n$,chr$(10))
 if lfp>0 then mid$(n$,lfp,1)=” “:goto 850
 print #2,n$;
 IF N$=”R” THEN PRINT #2, “stopped on error”:
GOTO 9001
 IF NUMB>=5 THEN numb=0 :goto 860
855 if eof(1) then 855
 goto 835
860 if not eof(3) then 800
 end
9000 print “err.no:”,err:resume
     9001 END
```

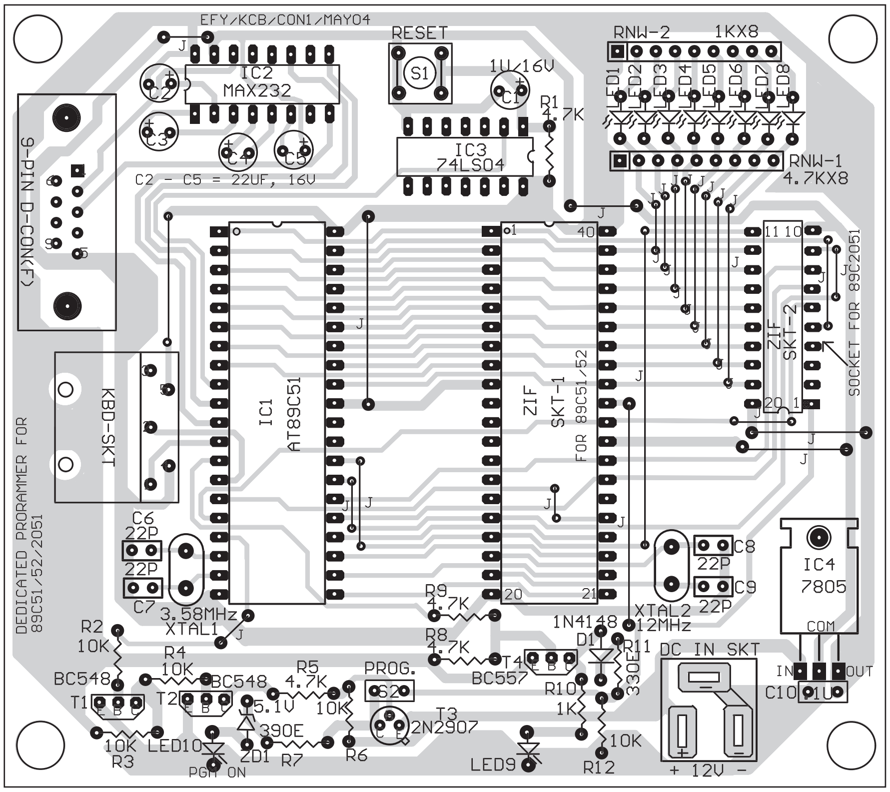


Fig. 8: Component layout for the PCB

one hex byte before being programmed into the new IC (in ZIF socket). The programmed data is then verified before sending the same, along with its address to the PC for its display on PC monitor. (The data followed by address is output serially to the computer.) The above procedure is repeated for programming and displaying of the next data byte.

Proper handshaking between the PC and programming board is essential for successful operation. The program (data.bas) for interfacing the PC to the programming board is written in Turbo Basic.(Turbo Basic TB.EXE file is included in current EFY-CD along with other software) The source code of data.bas is given below:

Start the Turbo Basic program (TB.exe) on the PC, select 'Key Break–On' in the Option menu bar, load the program (data.bas) for interfacing the PC and the programming board and run it. The program works in the non-compiled mode also. (*Note.* At line #800, the maximum value of variable kk may be varied, as necessary, depending on your PC's speed, so that the program works smoothly)

Programming

We shall discuss programming aspects relating to both the modes of operation namely, the direct keyboard-entry mode and the serial-port interface mode. For each mode a separate preprogrammed 89C51 microcontroller chip with different codes (for monitor program) is required.

Programming using direct keyboard-entry mode. In this configuration, a PC keyboard is connected to the board via keyboard connector provided on the programming board. The preprogrammed (with pgrmod1 data code) 89C51 microcontroller chip is put into the socket for IC1 and connected to a 12.6V supply. The IC to be programmed is inserted only after ensuring that 'Program' LED10 is off and resetting the circuit using push-to-on switch S1. Now programming can be started. Of course, only one of the two ZIF sockets is to be used at a time.

The keyboard is used for entering the address location, program data and commands for programming, verifying (reading) the programmed data bytes and erasing of the entire chip. The 'on' and 'off' status of the display LEDs indicate low (0) and high (1) logic levels, respectively (i.e. complement of the data). The software program takes into consideration the keys used for entering hexadecimal numbers 0 through 9 and letters A through F as also the keys used for high-address selection, low-address selection, incrementing, decrementing, programming and erasing as per the following details:

1. **Enter** key is used for incrementing the address.
2. **Backspace** key is used for decrementing the address.
3. **H** key is used for making the data field value as the high address.
4. **L** key is used for making the data field value as the low address.
5. **T** key is used for programming data at the current location.
6. **R** key is used for erasing the programmed IC.
7. **S** key is used for signature verification. It shows 1E on the LEDs.

When data, say, 75 is entered from the keyboard by first pressing '7' followed by '5' , the display LEDs show the entered data. If a mistake occurs during entry; say, '6' is entered after '7', re-entering '7' and '5' shows 75 Hex on the LED display.

To program this data into the microcontroller at location 0000H, press T key while keeping the 12V supply switch S2 pressed. This results in 75H to be programmed at location 0000H.

To advance to the next location, press Enter key. (To go back, press Backspace key.) Now enter the next byte to be programmed, say, 90H. If needed, correct as before. (Do not press Enter key or keys other than 0 through 9 and A through F.)

Then press T key again along with switch S2 to store 90H at location 0001H.

Every time T key is pressed, the 12V LED (green LED) blinks. This shows that the 12V pulse is applied to the EA/VPP pin. In this way data can be entered and programmed into the new IC byte-by-byte.

If data is to be entered at a location other than start address 0000H, the starting address can be set by using H and L keys as follows:

Supposing that you want to start programming from address 0250H. Enter 0 followed by 2 and then press H key. The high address is set to 02 Hex. Now Enter '5' followed by'0' and then press L key. The low address is set to 50 Hex. Thus, the programming start address is set to 0250H. The data is shown on the eight LEDs. (Please note that in a new good IC, all memory locations should read FF hex.) With an 89C2051 microcontroller (in ZIF socket-2) programming and verification (reading) cannot start from location other than origin (0000H) since 89C2051 has no provision for address input directly, but only by counting pulses applied into its pin 5, and the address is advanced by pulsing pin 5 (address line A0).

To read/verify data in an already programmed device starting with address 0000H, press Reset switch S1 and keep on pressing the Enter key to read data on the LED display byte-by-byte.

Erasing is done simply by resetting and pressing 12V switch S2 followed by R key. The 12V LED glows for a fraction of a second. The 12V LED should glow momentarily only when T (Program) key or R (Erase) key is pressed.

Programming using the serial-port interface mode. For this mode of operation the keyboard is not connected to the circuit board, but a 3-core cable is connected to the PC's spare Com port 1 or Com port 2 from the 9-pin 'D' connector on the programming board. Operation in this mode is feasible using DOS or Windows operating system. Programming the microcontroller IC in this mode requires ASCII code file (with extension .ASC), or the programming can be done byte-by-byte, using the PC's keyboard under 'P' option as explained later. The ASCII file is developed as follows:

1. The source program file (with extension .ASM) is developed using Assembly language, for which one can use X8051. exe cross-assembler program. The same program also generates its object code file (with extension .OBJ).

2. The code is converted into binary format (with .BIN extension) using the LINK151.exe program.

3. The binary file (with .BIN extension) is then converted into ASCII file (with .ASC extension) using the BIN4ASC. exe program.

As stated earlier, a different monitor program (with pgrmod2 data code), burnt into 89C51 IC is placed in the socket of IC1. Switch on the 12.6 V supply to the circuit board, insert the IC to be programmed in the ZIF socket and connect the programming board to the PC's Com port. Then, press Reset switch. If the 12V LED glows inadvertently at power-on, pressing the Reset button will put it out. Ensure that the 12V switch S2 is initially off.

Now you may run the data.bas program on the PC. This program sets the Com port for 1200 bauds, 8 bits, no parity. The program then prompts for the name of the ASCII file that is to be programmed into the fresh IC (in ZIF socket), as follows :

Filename?

Enter the file name interactively. For example, if the ASCII file name is EFY. ASC, type the same and press Enter key. (The code in EFY.ASC file contains code to display a message 'ElectronicForYou' on a 16X1 LCD module, which uses a Hitachi controller or equivalent that considers the single row as a configuous address from 0 to 15 for its 16 characters.) Now press Reset switch S1 on the board, momentarily. The following message should appear on PC monitor via the RS-232 Com port:

READY, Which Device, 8951 Or 52 Or 2051?

If the message doesn't appear on pressing Reset, check RS-232 connecting wires. Also check whether TXD pin 11 of the preprogrammed IC 89C51 is pulsing, using a logic probe. (It should pulsate.)

Enter 1, 2 or 3 to select the device. If you press key 1, the following message appears on the screen:

8951 choice

However, if you enter 2, the choice is 8952, and if you enter 3, the choice is 89C2051.

Now the program prompts:

Want to Erase, or Read or Prog or Lock? (E/R/P/L)

Make sure that 12V LED is 'off,' then press the 12V switch S2 and enter 'E' for erasing (if desired) the chip. The 12V Program LED10 glows for a while. You need not press Enter key after 'E' key is pressed.

After erasing is over, the following message comes up:

ERASE OVER, Now Send Data

For sending data (for programming) to the programming board from the PC, an ASCII file is needed. Simply enter 's' from the PC's keyboard. Of course, before entering 's' key, you need to ensure that, prior to pressing of the s key, the 12V LED does not glow. Now, keep the 12V switch pressed for the entire programming duration.

The data transfer takes place and the address gets incremented by one after programming each memory location. The 12V LED keeps on blinking during the programming process. The board sends the currently programmed address along with data to the PC for display on the monitor screen and we can watch this programming process.

The address gets incremented until either the entire chip has been programmed or else the data in the ASCII file has ended. Now open the 12V switch S2 and press Reset switch S1, on the board.

Data is programmed into the IC correctly, because after each byte is sent and programmed, the same is verified there. Then the next address is output to the PC to inform at what address programming is proceeding. In case data sent and data verified do not match, the following error message comes up:

ERR AT xxxx (Address)

and the program halts. Press Reset on the programming board to resume programming of the IC.

If the chip is programmed completely, the following message appears:

OVER

The read chip (R) option allows you to read the contents of a programmed microcontroller in the socket. This is useful to read already programmed chips, but if the lock bit is programmed, no data can be read.

In response to the prompt "Want to Erase, or Read or Prog or Lock? (E/R/P/L)" message, if you press R key, data is output to the computer (Binary data and so not in readable form directly.) The program automatically creates the capture.dat file containing data captured from the microcontroller in the current directory.

After data transfer is over, the program again prompts:

WANT TO ERASE, or READ or PROG or LOCK? (E/R/P/L)

If you want to exit from the program, press Control key in combination with Scroll Lock key followed by Escape key. All menu bars get enabled. Now, press Reset key on the programmer board. The captured data in capture.dat file can be viewed using debug utility. For this, first go to the DOS prompt command line and then go to the directory that contains the capture.dat file. Type debug capture.dat on the command prompt, press Enter key and then type 'd.' The captured data is displayed on the screen. The dump 'd' command shows data in the chip, which has been captured by the capture.dat file.

On the other hand, if you select programming (P) option in response to the "Want to Erase, or Read or Prog or Lock? (E/R/P/L)" prompt, you can program starting with any chosen memory address location on the target microcontroller chip (in ZIF socket-1) or from location 0000H(in ZIF socket-2)using the PC's keyboard. It first verifies the signature byte on the target microcontroller IC and checks whether it is correct. (The signature byte for an Atmel IC is 1E hex.) If the signature tallies, the board sends the following message to the PC:

SIGNATURE OK

If the signature does not tally, the following error message appears on the screen:

ERROR SIGNATURE, HALTED

If the signature is correct, the program prompts:

ENTER ADDRESS

For example, if the starting address is 0010H (for IC in ZIF socket-1) and data to be stored at 0010H is 75H, then proceed as follows.

Enter the four-digit hex address 0010 (do not press ‘Enter’ key). Now press ‘7.’ Then press ‘5’ in combination with the 12V switch (S2). Data 75H gets programmed at location 0010H. Release switch S2 after a single byte (75) is programmed. The PC’s screen shows ‘75.’

The address automatically increments to the next address (0011). The desired data for the new address can be entered by following the above procedure. Further data can be programmed in the same manner byte-by-byte. When all the data has been entered in this manner, press Reset switch S1.

If you want to program all the data contained in EFY.ASC file, starting from location 0000H, then after typing the four-digit hex address (0000), hold 12V switch S2 in pressed state and press s key. Thereupon, the green LED (LED10) flashes fast to indicate data transfer byte-by-byte. The screen shows data being programmed at the current address.

If there is an error in verification at any location, the following message, as stated earlier, appears:

ERR AT ….

and the program halts. Press Reset on the programming board to resume programming of the IC.

In this case, erase the chip and again try to program it.

The address gets incremented until either the chip has been programmed completely or else the data in the ASCII file has ended. After the programming is complete, release the 12V switch S2 and press Reset switch S1 on the programming board.

There is also a provision to write lock bit-1(refer Table). For this, choose the lock bit option ‘L’ in response to the “Want to Erase, or Read or Prog or Lock? (E/R/P/L)” prompt. Before pressing L key, the 12V LED should not glow inadvertently. If it does, press Reset to put it off. If the LED is ‘off,’ hold the 12V switch pressed and enter ‘L’ from the keyboard. The following message appears on the screen:

LOCKED,…, CAN’T, READ

When the microcontroller is locked, the capture.dat file can’t capture data in read option.

Practical demo circuit and programming example

The LCD module can be directly interfaced with an 89C51 (or 89C52) or 89C2051 chip as follows:

1. Fig. 5 shows interface circuit between an 89C51 microcontroller and an LCD module. Here eleven lines of microcontroller 89C51 are interfaced to the LCD module. Port 1 (8 lines comprising pin numbers 1 through 8) are used as data lines and three lines (P3.2 through P3.4 of port 3 of the 89C51 microcontroller) are used as control lines, respectively, for the LCD module. Accordingly, pins 1 through 8 of 89C51 are connected to data pins 7 through 14 of the LCD module. Port pins P3.2 through P3.4 (pins 12 through 14) are connected to pins 4 through 6 of the LCD module. Pin 31 of IC 89C51 is pulled high. The message “ElectronicForYou” gets displayed on the LCD module after you press Reset switch shown in Fig. 5. The listing file containing the source code burnt into 89C51 is given here as EFY.LST.

2. Fig. 6 shows interface circuit of the LCD module to 89C2051 microcontroller. Here pins 12 through 19 of the 89C2051 are connected to data pins 7 through 14 of the LCD module. Pins 6 through 8 of Port 3 (P3.2 through P3.4) are connected to pins 4 through 6 of the LCD module. The message “ElectronicForYou” gets displayed on the LCD module after you press Reset switch shown in Fig. 6. Both ICs (89C51 and 89C2051) contain identical program and as such EFY.LST is common for both the circuits of Figs. 5 and 6.

Note. For above examples**,** use only Hitachi HD44780U controller based on 16-character X 1-line LCD module.

An actual-size single-side PCB layout for the programmer circuit of Fig. 4 is shown in Fig. 7 with its component layout in Fig. 8.

The programs for the direct keyboard entry mode (pgrmod1.lst) and the serial-port interface mode (pgrmod2.lst) are self-explanatory. All relevant files, pertaining to this article, are included in the CD.

PGRMOD1.LST

```
        2500 A.D. 8051 CROSS ASSEMBLER   -   VERSION 3.41f
        ----------------------------------------------

                INPUT  FILENAME :  PGRMOD1.ASM
                OUTPUT FILENAME :  PGRMOD1.OBJ

00 32     TEMP      .EQU 32H
00 B6     ALE       .EQU 0B6H          ;P3.6
00 B7               VOLTS              .EQU 0B7H
00 50               SMALL              .EQU 50H
0000                         .ORG 0000H
0000      01 30     RESET: AJMP MONI
0003                         .ORG 0003H
0003      E1 03     AJMP 0703H         ; EXTERNAL INT. VECTOR 0
000B                .ORG 000BH
000B      E1 0B     AJMP 070BH         ; TO TIMER/COUNTER INTERRUPT  '0'
000F                         .ORG 000FH
000F      E1 23     AJMP 0723H         ; SERIAL INTERRUPT
0013                .ORG 0013H
0013      E1 13     AJMP 0713H         ; EXT. INT. 1 ADDRESS
001B                .ORG 001BH
001B      E1 1B     AJMP 071BH         ; EXT. TIMER COUNTER 1 INT. VEC.
0023                .ORG 0023H
0023      E1 23     AJMP 0723H         ; SERIAL PORT INTERRUPT VECTOR
0030                         .ORG 30H
0030                MONI:
0030      75 B0 FF  ST: MOV P3,#0FFH   ; ALL BITS IN PORT 3 SET CLR P3.6
0033      75 81 60  MOV SP,#060H
0036      75 80 FF  MOV P0,#0FFH
0039      C2 90     CLR P1.0           ; PIN 5 OF 2051 GND
003B      D2 B3     SETB P3.3          ; SET 5 v TO EA VPP PIN LOW (PIN 1
                                         2051 LOW)
003D      7F 28     MOV R7,#040
003F      DF FE     DJNZ R7,$
0041      C2 B3     CLR P3.3           ; LOW TO P3.3 GIVES 5 v TO EA VPP PINS
0043      C2 50     CLR SMALL          ; ASSUME BIG IC
0045      D2 B6     BEG:               SETB P3.6
0047      D2 B7     SETB P3.7
0049      90 00 00  MOV DPTR,#0
004C      7A 00     MOV R2,#0
004E      E5 82     A0: MOV A,DPL
0050      F5 90     MOV P1,A
0052      E5 83     A1: MOV A,DPH
0054      44 C0     ORL A,#0C0H        ; CONTROL CODE READ
0056      F5 A0     MOV P2,A           ; SO ADDRESS IS SET TO 00 00
0058      EA        A11: MOV A,R2
0059      F5 80     MOV P0,A
005B      12 00 78  AA: CALL KBD1
005E      12 01 E8  CALL CONVERT
0061      FB        MOV R3,A
0062      94 40     SUBB A,#040H
0064      50 1C     JNC D
0066      EA        MOV A,R2
0067      C4        SWAP A
0068      54 F0     ANL A,#0F0H
006A      4B        ORL A,R3
006B      FA        MOV R2,A
006C      E5 83     MOV A,DPH
006E      44 E0     ORL A,#0E0H        ; CONTROL CODE WRITE SO NEW IC
                                         DATA DOES NOT CLASH WITH THIS
0070      F5 A0     MOV P2,A
0072      E5 82     MOV A,DPL
0074      F5 90     MOV P1,A
0076      01 58     AJMP A11
0078      12 01 C4  KBD1: CALL KBD
007B      B4 F0 FA  CJNE A,#0F0H,KBD1
007E      12 01 C4  CALL KBD
0081      22        RET
0082      EB        D: MOV A,R3
```

```
0083   B4 41 03   CJNE A,#041H,A2     ; HIGH ADDRESS SET KEY
0086   02 00 C0   JMP H1              ; THIS KEY DONT WORK FOR 2051
0089   B4 42 03   A2:CJNE A,#042H,A3 ; LOW ADDRESS SET KEY
008C   02 00 CE   JMP L1              ; THIS KEY TOO WONT WORK FOR 2051 IC
008F   B4 43 03   A3: CJNE A,#043H,A4      ; GO KEY
0092   02 00 DE   JMP GO
0095   B4 48 03   A4: CJNE A,#048H,A5;SIGNATURE  s KEY
0098   02 00 E0   JMP SIGNATURE
009B   B4 47 03   A5: CJNE A,#047H,A6; ENTER KEY
009E   02 01 16   JMP NEXT_ADDR
00A1   B4 44 03   A6: CJNE A,#044H,A7      ; BACKSPACE KEY
00A4   02 01 34   JMP BACKADDR
00A7   B4 4A 03   A7: CJNE A,#04AH,A8      ; t KEY
00AA   02 01 4F   JMP PROG
00AD   B4 4B 03   A8: CJNE A,#04BH,A9      ; - RIGHTMOST KEY
00B0   02 01 7D   JMP ERASE
00B3   B4 49 03   A9: CJNE A,#049H,A10     ; k  KEY
00B6   02 01 9F   JMP LOCKBIT1
00B9   B4 4D 02   A10: CJNE A,#04DH,A101
00BC   D2 50      SETB SMALL
00BE   01 5B      A101: AJMP AA       ;
00C0   75 80 FF   H1: MOV P0,#0FFH    ; PORT 0 SHOULD BE INPUT PORT NOW!
00C3   EA         MOV A,R2
00C4   54 0F      ANL A,#00FH
00C6   F5 83      MOV DPH,A
00C8   44 C0      ORL A,#0C0H         ; READ MODE ONLY
00CA   F5 A0      MOV P2,A
00CC   01 5B      AJMP AA ;DISPLAY NOW SHOWS THE DATA THERE!
00CE   75 80 FF   L1: MOV P0,#0FFH    ; PORT 0 SHOULD BE INPUT PORT NOW!
00D1   EA         MOV A,R2
00D2   F5 82      MOV DPL,A
00D4   F5 90      MOV P1,A
00D6   E5 83      MOV A,DPH
00D8   44 C0      ORL A,#0C0H         ; READ MODE ONLY
00DA   F5 A0      MOV P2,A
00DC   01 5B      AJMP AA ;DISPLAY NOW SHOWS THE DATA THERE!
00DE   80 FE      GO: SJMP $          ; FOR FUTURE USE NOW HALTS HERE !
00E0   D2 B6      SIGNATURE: SETB P3.6
00E2   75 80 FF   MOV P0,#0FFH
00E5   74 00      MOV A,#0
00E7   F5 A0      MOV P2,A
00E9   30 50 04   JNB 50H,SIG ;00F1H
00EC   74 00      MOV A,#00H
00EE   01 F3      JMP 00F3H
00F0   74 30      SIG: MOV A,#30H
00F2   F5 90      MOV P1,A
00F4   7F 30      MOV R7,#30H
00F6   DF FE      DJNZ R7,$
00F8   E5 80      MOV A,P0
00FA   B4 1E 02   CJNE A,#01EH,ERR
00FD   01 5B      JMP AA              ; IF OK, THEN PROCEED WITH FURTHER
                                        KEY ENTRY
00FF   12 01 04   ERR: CALL BEEP
0102   01 FF      AJMP ERR            ; ELSE SHOW THE WRONG DATA NOT
                                        1EH THERE ON LEDS & HALT
0104   D2 97      BEEP: SETB P1.7     ; USE PORT 1 BIT 7 (A7 ADDR. LINE!)
                                        TO PULSE
0106   12 01 10   CALL DEL2
0109   C2 97      CLR P1.7
010B   12 01 10   CALL DEL2
010E   21 04      AJMP BEEP
0110   7F FF      DEL2: MOV R7,#0FFH
0112   00         BPN: NOP
0113   DF FD      DJNZ R7,BPN
0115   22         RET
0116   75 80 FF   NEXT_ADDR: MOV P0,#0FFH           ; PORT 0 SHOULD BE
                                                      INPUT PORT NOW!
0119   05 82      INC DPL
011B   E5 82      MOV A,DPL
011D   F5 90      MOV P1,A
011F   B4 00 02   CJNE A,#00,NN1
0122   05 83      INC DPH
0124   E5 83      NN1: MOV A,DPH
0126   44 C0      ORL A,#0C0H         ; CONTROL CODE READ
0128   F5 A0      MOV P2,A            ; WRITE THERE
012A   30 50 05   JNB SMALL,NM2
012D   D2 90      SETB P1.0           ; 2051 NEEDS ADDRESS IN CR. BY PULSE
012F   00         NOP
0130   C2 90      CLR P1.0
                                      ; NOW DISPLAY SHOWS
DATA AT THAT                            ADDRESS AUTOMATI-
CALLY
0132   01 5B      NM2: JMP AA
0134   30 50 02   BACKADDR: JNB SMALL,POSS          ; NOT POSSIBLE FOR
                                                      2051 TO DEC. ADDR !
0137   01 5B      JMP AA
0139   75 80 FF   POSS: MOV P0,#0FFH       ; PORT 0 SHOULD BE INPUT PORT
                                             NOW!
013C   15 82      DEC DPL
013E   E5 82      MOV A,DPL
0140   F5 90      MOV P1,A
                                      ; NOW DISPLAY SHOWS
DATA AT THAT                            ADDRESS AUTOMATI-
CALLY
0142   B4 FF 02   CJNE A,#0FFH,NM1
0145   15 83      DEC DPH
0147   E5 83      NM1: MOV A,DPH
0149   44 C0      ORL A,#0C0H         ; CONTROL CODE READ
014B   F5 A0      MOV P2,A            ; WRITE THERE
014D   01 5B      JMP AA
014F   C2 B3      PROG: CLR P3.3
0151   7F 30      MOV R7,#48          ; 48 CLCL
0153   DF FE      DJNZ R7,$
0155   C2 B7      CLR P3.7            ; VOLTS 12
0157   7F 0A      MOV R7,#10
0159   DF FE      DJNZ R7,$           ; DELAY 10 MICRO
015B   C2 B6      CLR P3.6            ; ALE LOW
015D   7F 0A      MOV R7,#10
015F   DF FE      DJNZ R7,$           ; DELAY 10 MICRO
0161   D2 B6      SETB P3.6           ; ALE MADE HIGH
0163   12 01 96   CALL DELAY          ; 2 MS DELAY FOR PROGRAMMING GIVEN
0166   D2 B7      SETB P3.7           ; VOLTS  ;ALE PIN MADE HIGH AGAIN
                                        AND VOLTS NOT 12 V
0168   D2 B6      SETB P3.6           ; ALE
016A   75 80 FF   MOV P0,#0FFH        ; MAKE PORT 0 AS INPUT PORT
016D   E5 83      MOV A,DPH
016F   44 C0      ORL A,#0C0H         ; CONTROL CODE FOR READ
0171   F5 A0      MOV P2,A
0173   E5 80      MOV A,P0            ; READ THE DATA
0175   F5 32      MOV TEMP,A
0177   EA         MOV A,R2
0178   B5 32 84   CJNE A,TEMP,ERR
017B   01 5B      JMP AA
017D   74 10      ERASE: MOV A,#10H
017F   F5 A0      MOV P2,A            ; ERASE CODE
0181   7F 50      MOV R7,#50H
0183   DF FE      DJNZ R7,$
                                      ; ALE  LOW  VOLTS
HIGH
0185   C2 B6      CLR P3.6
0187   C2 B7      CLR P3.7
0189   7D 32      MOV R5,#32H
018B   12 01 96   DEL1: CALL DELAY
018E   DD FE      DJNZ R5,$
0190   D2 B6      SETB P3.6
0192   D2 B7      SETB P3.7
0194   01 45      JMP BEG
0196   7F 0A      DELAY: MOV R7,#10
0198   7E 3C      TH: MOV R6,#60      ; 600 X 12/3.57 = 2 MS
019A   DE FE      DJNZ R6,$
019C   DF FA      DJNZ R7,TH
019E   22         RET
019F   74 F0      LOCKBIT1: MOV A,#0F0H
01A1   F5 A0      MOV P2,A
01A3   7F 0A      MOV R7,#10
01A5   DF FE      DJNZ R7,$
01A7   7F 30      MOV R7,#48          ; CLCL
01A9   DF FE      DJNZ R7,$
01AB   C2 B7      CLR P3.7            ; VOLTS 12
01AD   7F 0A      MOV R7,#10
01AF   DF FE      DJNZ R7,$           ; DELAY 10 MICRO
01B1   C2 B6      CLR P3.6            ; ALE LOW
01B3   7F 0A      MOV R7,#10
01B5   DF FE      DJNZ R7,$           ; DELAY 10 MICRO
01B7   D2 B6      SETB P3.6           ; ALE MADE HIGH
01B9   31 96      CALL DELAY          ; 2 MS DELAY FOR PROGRAMMING GIVEN
01BB   D2 B7      SETB P3.7           ; VOLTS  ; ALE PIN MADE HIGH AGAIN
                                        AND VOLTS NOT 12 V
01BD   D2 B6      SETB P3.6           ; ALE
01BF   75 80 FF   MOV P0,#0FFH        ; MAKE PORT 0 AS INPUT PORT
01C2   01 5B      JMP AA
01C4   7B 08      KBD: MOV R3,#8
01C6   7F 00      MOV R7,#0
01C8   A2 B2      KP1: MOV C,P3.2
01CA   40 FC      JC KP1
01CC   A2 B2      K4: MOV C,P3.2
01CE   50 FC      JNC K4
01D0   A2 B2      K5: MOV C,P3.2
01D2   40 FC      JC K5
01D4   A2 B4      MOV C,P3.4
01D6   EF         MOV A,R7
01D7   13         RRC A
01D8   FF         MOV R7,A
01D9   A2 B2      K6: MOV C,P3.2
01DB   50 FC      JNC K6
01DD   DB F1      DJNZ R3,K5
01DF   31 E3      ACALL DELAY1
01E1   FD         MOV R5,A
01E2   22         RET
01E3   7C 80      DELAY1: MOV R4,#80H
01E5   DC FE      DJNZ R4,$
01E7   22         RET
01E8                 CONVERT:
01E8   B4 45 03   CODECHK: CJNE A,#45H,K1
01EB   74 00      MOV A,#0            ; "0" KEY
01ED   22         RET
01EE   B4 16 03   K1: CJNE A,#16H,K2
01F1   74 01      MOV A,#1            ; "1" KEY
01F3   22         RET
01F4   B4 1E 03   K2: CJNE A,#01EH,K3
01F7   74 02      MOV A,#2            ; "2" KEY
01F9   22         RET
01FA   B4 26 03   K3: CJNE A,#26H,K41
01FD   74 03      MOV A,#3            ; "3" KEY
01FF   22         RET
0200   B4 25 03   K41: CJNE A,#25H,K51
0203   74 04      MOV A,#4            ; "4" KEY
0205   22         RET
0206   B4 2E 03   K51: CJNE A,#2EH,K61        ; "5" KEY
0209   74 05      MOV A,#5
020B   22         RET
020C   B4 36 03   K61: CJNE A,#36H,K7         ; "6" KEY
020F   74 06      MOV A,#6
0211   22         RET
0212   B4 3D 03   K7: CJNE A,#03DH,K8         ; "7" KEY
0215   74 07      MOV A,#7
0217   22         RET
0218   B4 3E 03   K8:    CJNE A,#03EH,K9
021B   74 08      MOV A,#8                    ; "8" KET
021D   22         RET
021E   B4 46 03   K9: CJNE A,#46H,KA          ; "9" KEY
0221   74 09      MOV A,#9
0223   22         RET
0224   B4 70 03   KA: CJNE A,#70H,KB          ; "0" KEY
0227   74 00      MOV A,#0
0229   22         RET
022A   B4 69 03   KB: CJNE A,#69H,KD          ; "1" KEY
022D   74 01      MOV A,#1
022F   22         RET
0230   B4 72 03   KD: CJNE A,#72H,KE          ; "2" KEY
0233   74 02      MOV A,#2
0235   22         RET
0236   B4 7A 03   KE: CJNE A,#07AH,KF         ; "3" KEY
0239   74 03      MOV A,#3
023B   22         RET
023C   B4 6B 03   KF: CJNE A,#06BH,KG         ; "4" KEY
023F   74 04      MOV A,#4
0241   22         RET
0242   B4 73 03   KG: CJNE A,#73H,KH          ; "5" KEY
0245   74 05      MOV A,#5
0247   22         RET
0248   B4 74 03   KH: CJNE A,#74H,KI          ; "6" KEY
024B   74 06      MOV A,#6
024D   22         RET
```

```
024E   B4 6C 03   KI: CJNE A,#06CH,KJ       ; "7" KEY
0251   74 07      MOV A,#7
0253   22         RET
0254   B4 75 03   KJ: CJNE A,#75H,KK        ; "8" KEY
0257   74 08      MOV A,#8
0259   22         RET
025A   B4 7D 03   KK: CJNE A,#75H,KL        ; "9" KEY
025D   74 09      MOV A,#9
025F   22         RET
0260   B4 5A 03   KL: CJNE A,#05AH,KM       ; "ENTER" KEY
0263   74 47      MOV A,#47H
0265   22         RET
0266   B4 33 03   KM: CJNE A,#33H,KN        ; "H" KEY
0269   74 41      MOV A,#41H
026B   22         RET
026C   B4 4B 03   KN: CJNE A,#04BH,KO       ; "L" KEY
026F   74 42      MOV A,#42H
0271   22         RET
0272   B4 66 03   KO: CJNE A,#66H,KP        ; bACKSPACE
0275   74 44      MOV A,#44H           ; TO DECREMENT
0277   22         RET
0278   B4 1B 03   KP: CJNE A,#01BH,KQ
027B   74 48      MOV A,#48H           ; INCREMENT ONLY
                                                      ; KEY IS S KEY
027D   22         RET
027E   B4 2C 03   KQ: CJNE A,#02CH,KR
0281   74 4A      MOV A,#04AH          ; REGISTER STORE
0283   22         RET                       ; IS T KEY

0284   B4 42 03   KR: CJNE A,#42H,KT
0287   74 49      MOV A,#49H           ; K KEY FOR NO
0289   22         RET                       ; ACCESS
028A   B4 1C 03   KT: CJNE A,#01CH,KS
028D   74 0A      MOV A,#0AH           ; KEY A ;
028F   22         RET
0290   B4 32 03   KS: CJNE A,#32H,KU
0293   74 0B      MOV A,#0BH           ; KEY B ;
0295   22         RET
0296   B4 21 03   KU: CJNE A,#21H,KV
0299   74 0C      MOV A,#0CH           ; KEY C ;
029B   22         RET
029C   B4 23 03   KV: CJNE A,#23H,KW
029F   74 0D      MOV A,#0DH           ; KEY D ;
02A1   22         RET
02A2   B4 24 03   KW: CJNE A,#24H,KX
02A5   74 0E      MOV A,#0EH
02A7   22         RET                            ; e KEY
02A8   B4 2B 03   KX: CJNE A,#02BH,KY
02AB   74 0F      MOV A,#0FH
02AD   22         RET                            ; F KEY

02AE   B4 2D 03   KY: CJNE A,#02DH,KZ       ; 07BH
02B1   74 4B      MOV A,#04BH          ; SMALL R KEY
02B3   22         RET                            ; ERASE

02B4   B4 34 03   KZ: CJNE A,#34H,KZ1
02B7   74 43      MOV A,#043H
02B9   22         RET
02BA   B4 2A 03   KZ1: CJNE A,#02AH,KZ2
02BD   74 0B      MOV A,#0BH
02BF   22         RET
02C0   B4 3A 03   KZ2: CJNE A,#3AH,KZ3
02C3   74 4D      MOV A,#4DH
02C5   22         RET
02C6   74 FF      KZ3: MOV  A,#0FFH
02C8   22         RET
02C9                   END
LINES ASSEMBLED :  350        ASSEMBLY ERRORS :   0
```

PGRMOD2.LST

```
2500 A.D. 8051 CROSS ASSEMBLER   -   VERSION 3.41f
--------------------------------------------------
     INPUT  FILENAME :  PGRMOD2.ASM
     OUTPUT FILENAME :  PGRMOD2.OBJ

0000              .ORG 0
0000   01 30      JMP 0030H

0023              .ORG      23H
0023   02 03 95   JMP SERINT           ; ISR SERIAL

       00 40      PC      .EQU 40H
       00 41      MAXADR              .EQU 41H
       00 43      SER_DATA            .EQU 43H
       00 30      FLAG                .EQU 30H
       00 31      SMALL               .EQU 31H
       00 B7      VOLTS               .EQU B7H          ;P3.7
       00 B6      ALE                 .EQU B6H          ;P3.6

0030              .ORG 30H


0030   75 81 60   BEGIN: MOV SP,#60H
0033   75 B0 FF   MOV P3,#FFH          ;all bits of port 3 set
0036   75 90 00   MOV P1,#0
0039   75 80 FF   MOV P0,#FFH          ;TEST

003C   12 03 89   CALL SER_INIT
003F              KK2:
003F   75 A8 90   MOV IE,#90H          ;ENABLE SER INTERRUPT

0042   90 00 4B   MM1: MOV DPTR,#MES1
0045   12 03 7E   CALL MESDISP
0048   02 00 7C   JMP S2


004B              MES1:
004B   52 45 41 44   .DB "R","E","A","D","Y"," ","W","h","i","c","h","","D","e","v",
                  "i","c","e"
004F   59 20 57 68
0053   69 63 68 20
0057   44 65 76 69
005B   63 65
005D   38 39 35 31   .DB    "8","9","5","1"," ","O","R"," ","5","2",
                        ","O","r"," ","2","0","5","1","?"
0061   20 4F 52 20
0065   35 32 20 4F
0069   72 20 32 30
006D   35 31 3F
0070   45 4E 54 45   .DB "E","N","T","E","R"," ","1","2","3",10,13,FFH
0074   52 20 31 32
0078   33 0A 0D FF
                                                   ;READY, WHICH DEVICE,
8951, OR 52 OR
                                     2051 ? ENTER 1,2,3"

007C   30 30 FD   S2:     JNB FLAG,$   ;flag bit high indicates a data byte received
007F   E5 43      MOV A,43H            ;serial bufer data
0081   C2 30      CLR FLAG             ;WE HAVE NOW USED THE LAST RECEIVED
                                        DATA
0083   B4 31 1D   CJNE A,#31H,N1       ;for entry "1"
0086   74 10      MOV A,#10H           ; MAX ADDR FOR 51 IS 0F H ONLY SO 10 H
                                       IS NEXT
0088   F5 41      MOV MAXADR,A
008A   C2 31      CLR SMALL            ;BIG IC
008C   90 00 95   MOV DPTR,#MES2
008F   12 03 7E   CALL MESDISP

0092   02 00 F4   JMP S4
0095   38 39 35 31   MES2: .DB "8","9","5","1"," ","C',"H","O","I","C","E",10,13,FFH
0099   20 43 48 4F
009D   49 43 45 0A
00A1   0D FF
00A3   B4 32 1D   N1: CJNE A,#32H,N2 ;for entry "2",Max address for 8052 is 20H
00A6   74 20      MOV A,#20H
00A8   F5 41      MOV MAXADR,A
00AA   C2 31      CLR SMALL            ; BIG IC
00AC   90 00 B5   MOV DPTR,#MES3
00AF   12 03 7E   CALL MESDISP
00B2   02 00 F4   JMP S4
00B5   38 39 35 32   MES3: .DB "8","9","5","2"," ","C',"H","O","I","C","E",10,13,FFH
00B9   20 43 48 4F
00BD   49 43 45 0A
00C1   0D FF
                        N2:            ;Max addr. for SMALL ic IS 2K
00C3   B4 33 2C   CJNE A,#33H,N3
00C6   75 41 08   MOV MAXADR,#08H
00C9   D2 31      SETB SMALL
00CB   90 00 E4   MOV DPTR,#MES4
00CE   12 03 7E   CALL MESDISP
00D1   C2 B3      CLR P3.3             ; MAKE 5 V AVAILABLE FOR VPP PIN 1
00D3   D2 B7      SETB P3.7            ; DONT APPLY 12 V
00D5   D2 B5      SETB P3.5            ; MAKE RST PIN GROUND VIA 7406
00D7   C2 90      CLR P1.0             ; MAKE PIN XTAL 1 LOW
00D9   7F 30      MOV R7,#48
00DB   DF FE      DJNZ R7,$
00DD   C2 B5      CLR P3.5             ; MAKES RST PIN1 TO 5 volts
00DF   D2 B6      SETB P3.6
00E1
00E1   02 00 F4   JMP S4

00E4   32 30 35 31   MES4: .DB "2","0","5","1"," ","C',"H","O","I","C","E",10,13,FFH
00E8   20 43 48 4F
00EC   49 43 45 0A
00F0   0D FF
00F2   01 42      N3:     JMP MM1
                                       ; PROGRAM FOR 8951 BEGINS

00F4   C2 B3      S4:     CLR P3.3     ;5 v TO EA VPP PIN AND ALSO TO PIN 1
                                        OF 2051 IC SKT
00F6   90 00 FF   MOV DPTR,#ERASEMES
00F9   12 03 7E   CALL MESDISP
00FC   02 01 2D   JMP S5
00FF              ERASEMES:
00FF   57 41 4E 54   .DB    "W","A","N","T"," ","T","O","
                        ","E","R","A","S","E","?","O","R"," ","R","E","A","D","
                  ","O","R"," ","P","R","O","G"," ","O","r","
                        ","L","O","C","K"," ","E","/","R","/","P"/"L",10,13,FFH
0103   20 54 4F 20
0107   45 52 41 53
010B   45 3F 4F 52
010F   20 52 45 41
0113   44 20 4F 52
0117   20 50 52 4F
011B   47 20 4F 72
011F   20 4C 4F 43
0123   4B 20 45 2F
0127   52 2F 01 0A
012B   0D FF

012D   30 30 FD   S5:     JNB FLAG,S5
0130   E5 43              MOV A,43H
0132   C2 30              CLR FLAG
0134   B4 45 03           CJNE A,#"E",S6


0137   02 03 E1           JMP ERASE
013A   B4 52 03           S6: CJNE A,#"R",S7
013D   02 01 7D           JMP READ
0140   B4 50 03           S7: CJNE A,#"P",S8
0143   02 01 F0           JMP S6_1
0146   B4 4C AB           S8: CJNE A,#"L",S4
0149   02 01 4C           JMP LOCK
014C   D2 B7              LOCK: SETB P3.7
014E   74 F0              MOV A,#F0H


0150   F5 A0              MOV P2,A     ;OUTPUT TO PORT 2
0152   7F 17              MOV R7,#17H
0154   DF FE              DJNZ R7,$
                                       ;ALE LOW VOLTS HIGH
0156   C2 B7              CLR P3.7     ;VOLTS
0158   C2 B6              CLR P3.6     ;ALE
015A   7F 1C              MOV R7,#28   ;DELAY 10O MICROSEC
```

```
015C    DF FE          DJNZ R7,$
015E    D2 B6          SETB P3.6      ;ALE
0160    D2 B7          SETB P3.7      ;VOLTS
0162    90 01 6A       MOV DPTR,#MESLOCK

0165    12 03 7E       CALL MESDISP

0168    01 F4          JMP S4
016A    4C 4F 43 4B                   MESLOCK: .DB "L","O","C","K","E","D",".",
                       "C","A","N","T"," ","R","E","A","D",10,13,FFH
016E    45 44 2E 43
0172    41 4E 54 20
0176    52 45 41 44
017A    0A 0D FF
017D    90 01 90       READ: MOV DPTR,#MESREAD
0180    12 03 7E       CALL MESDISP
0183    30 30 FD       RR12: JNB FLAG,RR12
0186    C2 30          CLR FLAG
0188    E5 43          MOV A,SER_DATA  ;43H
018A    B4 59 F6       CJNE A,#"Y",RR12
018D    02 01 AD       JMP READ1
0190    43 41 50 54    MESREAD: .DB "C","A","P","T","U","R","E","
                       ","D","A","T","A"," ","s","a","y"," ","Y"," ","f","o","r"," "
0194    55 52 45 20
0198    44 41 54 41
019C    20 73 61 79
01A0    20 59 20 66
01A4    6F 72 20
01A7    79 65 73 0A    .DB "y","e","s",10,13,FFH
01AB    0D FF
01AD    C2 B3          READ1: CLR P3.3      ; 5V to pin EA\VPP
01AF    D2 B7          SETB P3.7
01B1    D2 B6          SETB P3.6
01B3    C2 B5          CLR P3.5       ;HIGHEST ADDR. IS SET LOW
01B5    90 00 00       MOV DPTR,#0

01B8    E5 83          CONT1: MOV A,DPH
01BA    44 C0          ORL A,#C0H     ;READ
01BC    F5 A0          MOV P2,A
01BE    30 31 03       JNB SMALL,CONT2
01C1    02 01 C8       JMP CONT3
01C4    E5 82          CONT2:    MOV A,DPL
01C6    F5 90          MOV P1,A
01C8    7F 0A          CONT3:    MOV R7,#10
01CA    DF FE          DJNZ R7,$
01CC    E5 80          MOV A,P0       ;READ DATA   AND OUTPUT TO SERIAL
PORT
01CE    71 A9          ACALL TOUT
01D0    A3             INC DPTR
01D1    30 31 07       JNB SMALL,CCC
01D4    C2 90          CLR P1.0
01D6    D2 90          SETB P1.0      ;PULSE PIN 5 OF 2051 FOR ADDR. INCR.
01D8    00             NOP
01D9    C2 90          CLR P1.0
01DB    E5 83          CCC:     MOV A,DPH
01DD    30 31 06       JNB SMALL,CC2
01E0    B4 08 D5       CC1:   CJNE A,#08H,CONT1
01E3    02 01 EE       JMP CC3
01E6    B4 10 CF       CC2:   CJNE A,#10H,CONT1
01E9    D2 B5          SETB P3.5      ; HIGHEST ADDRESS MADE HIGH
01EB    B5 41 CA       CJNE A,MAXADR,CONT1
01EE    01 F4          CC3:     JMP S4



01F0    12 03 5F       S6_1:      CALL DISPLAY

01F3    C2 B3          CLR  P3.3      ;SWITCH ON 5 v TO EA\VPP
                                      ;BITS PORT 2   17 16 28 27  A11  A10 A9 A8
01F5    D2 B7          SETB P3.7      ;VOLTAGE  NOT 12 V
01F7    D2 B6          SETB P3.6      ;HIGH ALE




01F9    90 00 00       MOV DPTR,#0  ;FIRST ADDRESS OF NEW IC IS POINTED
TO.
01FC    C2 90          CLR P1.0
01FE    85 82 30       MOV 30H,DPL   ;ALSO SAVE IN 30,31 INT. RAM
0201    85 83 31       MOV 31H,DPH
0204    12 02 CE       CALL SIGNATURE
0207                   MORE:
                                      ;GET ADDRESS
0207    E5 31          MOV A,31H      ;HIGH ADDRESS
0209    54 0F          ANL A,#0FH
020B    44 E0          ORL A,#E0H     ;CONTROL CODE FOR WRITING
020D    F5 40          MOV PC,A       ; PC IS STORE FOR PORT 2 data
                                                            ;WRITE CONTROL CODE
FOR WRITE AND
                                      ALSO THE PROGRAM HIGH ADDRESS
020F    F5 A0          MOV P2,A       ; OUTPUT TO PORT 2

0211    E5 30          MOV A,30H      ;LOW ADDRESS
0213    30 31 03       JNB SMALL,YES_OUT  ; DONT OUTPUT ADDR. FOR
                                              89C2051 !
0216    02 02 1B       JMP NO_OUT
0219    F5 90          YES_OUT:MOV P1,A  ;LOW ADDRESS
                                      ;LOOK DATA FROM SERIAL AND PUT TO
                                      DATA PORT
021B    12 03 3E       NO_OUT:     CALL GETBYTE
021E    F5 80          MOV P0,A       ; PORT 0 IS DATA
                                      ; DELAY 48 CLCL
0220    7F 30          MOV R7,#48
0222    DF FE          DJNZ R7,$
                                      ;12 v APPLIED
0224    C2 B7          CLR  P3.7      ;THIS MAKES 12 v AVAILABLE TO PIN 31
                                                            ;DELAY 10 MICROS:
0226    7F 0A          MOV R7,#10
0228    DF FE          DJNZ R7,$
                                                            ;ALE PULSE LOW:
022A    C2 B6          CLR  P3.6      ;MAKE ALE PIN LOW
                                      ;DELAY 10 MICROS
022C    7F 0A          MOV R7,#10
022E    DF FE          DJNZ R7,$
                                      ;ALE MADE HIGH
0230    D2 B6          SETB P3.6      ; ALE
0232    12 04 1C       CALL DELAY  ;DELAY 2 MILLISECONDS
                                      ;JNB P3.5,$  ;CHECK FOR READY - NOT
                                      USED NOW
                                                            ;VOLTS LOW TO 5 V
0235    D2 B7          SETB P3.7      ;VOLTS
                                                            ;VERIFY PHASE"

0237    75 80 FF       MOV P0,#FFH
023A    E5 31          MOV A,31H
023C    54 0F          ANL A,#0FH
023E    44 C0          ORL A,#C0H     ;READ CODE
0240    F5 A0          MOV P2,A       ;HI ADDRESS IN PORT 2
0242    E5 30          MOV A,30H
0244    30 31 03       JNB SMALL,VV1
0247    02 02 4C       JMP VV2
024A    F5 90          VV1: MOV P1,A   ; P1 LO ADDR.
024C    7F 20          VV2: MOV R7,#20H
024E    DF FE          DJNZ R7,$      ;DELAY
                                      ;READ DATA:
0250    E5 80          MOV A,P0       ;READS THE DATA ON PORT 0
0252    F5 32          MOV 32H,A      ;SAVE IN DATA LOCATION

0254    EA             MOV A,R2       ;GET DATA PROGRAMMED

0255    B5 32 61       CJNE A,32H,ERR
0258    12 02 8C       CALL ADDROUT

025B    85 30 82       MOV DPL,30H  ;ADDR INCR.
025E    85 31 83       MOV DPH,31H
0261    A3             INC DPTR
0262    85 82 30       MOV 30H,DPL
0265    85 83 31       MOV 31H,DPH
                                      ;ADDRESS INCREMENT FOR NEXT USE
0268    30 31 08       JNB SMALL,PV1
026B    C2 90          CLR P1.0
026D    00             NOP
026E    D2 90          SETB P1.0      ;PULSE ADDRESS LINE FOR 2051
0270    00             NOP
0271    C2 90          CLR P1.0
0273                   PV1:
0273    E5 83          MOV A,DPH
0275    30 31 06       JNB SMALL,RR13
0278    B4 08 3C       CJNE A,#08H,MORE1
027B    02 03 CF       JMP END
027E    B4 10 08       RR13: CJNE A,#10H,MORE11
0281    D2 B5          SETB P3.5      ;HIGHEST ADDRESS OF 8052
0283    B4 41 31       CJNE A,#MAXADR,MORE1

0286    02 03 CF       JMP END
0289    02 02 B7       MORE11:   JMP MORE1
028C                                  ADDROUT:
028C    E5 31          MOV A,31H
028E    C4             SWAP A
028F    54 0F          ANL A,#0FH
0291    12 03 C2       CALL HEXASC
0294    71 A9          ACALL TOUT

0296    E5 31          MOV A,31H
0298    54 0F          ANL A,#0FH
029A    12 03 C2       CALL HEXASC
029D    71 A9          ACALL TOUT

029F    E5 30          MOV A,30H
02A1    C4             SWAP A
02A2    54 0F          ANL A,#0FH
02A4    12 03 C2       CALL HEXASC
02A7    71 A9          ACALL TOUT

02A9    E5 30          MOV A,30H
02AB    54 0F          ANL A,#0FH
02AD    12 03 C2       CALL HEXASC
02B0    71 A9          ACALL TOUT


02B2    74 0D          MOV A,#0DH
02B4    71 A9          ACALL TOUT
02B6    22             RET

02B7    41 07          MORE1: JMP MORE

                       ERR:   ;SEND ERROR MESSAGE

02B9    90 02 C3       MOV DPTR,#MESERR
02BC    12 03 7E       CALL MESDISP
02BF    51 8C          CALL ADDROUT

02C1    80 FE          SJMP $
02C3    45 52 52 2E    MESERR: .DB "E","R","R","."," ","A","T"," ",10,13,FFH
02C7    20 41 54 20
02CB    0A 0D FF

02CE                                  SIGNATURE:
                                      ;WRITE ADDR 30H AND CONTROL CODE
                                      00 TO READ BYTE AS 1EH
02CE    C2 B3          CLR P3.3       ;PIN 1 HIGH
02D0    D2 B6          SETB P3.6      ;ALE HIGH
02D2    D2 B7          SETB P3.7      ;12 V IS NOT APPLIED

02D4    74 00          MOV A,#00      ;READ CODE
02D6    F5 A0          MOV P2,A       ;HI ADDRESS IN PORT 2
02D8    30 31 05       JNB SMALL,SIG1
02DB    74 00          MOV A,#00H
02DD    02 02 E4       JMP SIG2
02E0    74 30          SIG1:    MOV A,#30H
02E2    F5 90          MOV P1,A       ; P1 LO ADDR.
02E4    7F 20          SIG2:       MOV R7,#20H
02E6    DF FE          DJNZ R7,$      ;DELAY
02E8    E5 80          MOV A,P0       ;DATA READ INTO A
02EA
02EA    B4 1E 32       CJNE A,#1EH,ERRSIG
02ED    90 03 00       MOV DPTR,#SIGMES
02F0    12 03 7E       CALL MESDISP
02F3    12 03 3E       CALL GETBYTE
02F6    EA             MOV A,R2
02F7    F5 31          MOV 31H,A
02F9    12 03 3E       CALL GETBYTE
02FC    EA             MOV A,R2
02FD    F5 30          MOV 30H,A
02FF    22             RET
```

```
0300    53 49 47 4E        SIGMES: .DB "S","I","G","N","A","T","U","R","E","
                                  ","O",".","K",10,13
0304    41 54 55 52
0308    45 20 4F 2E
030C    4B 0A 0D
030F    45 4E 54 45        .DB "E","N","T","E","R"," ","A","D","D","R","E","S","S",
                         10,13,FFH
0313    52 20 41 44
0317    44 52 45 53
031B    53 0A 0D FF
031F    90 03 27           ERRSIG:   MOV DPTR,#SIGMESER
0322    12 03 7E           CALL MESDISP
0325    80 FE              SJMP $            ;HALT ON ERROR
0327                                 SIGMESER:
0327    45 52 52 4F        .DB "E","R","R","O","R"," "
032B    52 20
032D    53 49 47 4E        .DB "S","I","G","N","A","T","U","R","E","
                             ","H","A","L","T",10,13,FFH
0331    41 54 55 52
0335    45 20 48 41
0339    4C 54 0A 0D
033D    FF
033E    30 30 FD           GETBYTE: JNB FLAG,$     ;read new serial byte
0341    E5 43              MOV A,SER_DATA     ;DATA IN A
0343    C2 30              CLR FLAG
0345    71 A9              ACALL TOUT
0347    12 04 25           CALL ASCI_HEX

034A    C4                 SWAP A
034B    54 F0              ANL A,#F0H
034D    FA                 MOV R2,A
034E    30 30 FD           JNB FLAG,$     ;read new serial byte
0351    E5 43              MOV A,SER_DATA  ;DATA IN A
0353    71 A9              ACALL TOUT
0355    12 04 25           CALL ASCI_HEX
0358    C2 30              CLR FLAG

035A    54 0F              ANL A,#0FH
035C    4A                 ORL A,R2          ; SECOND NIBBLE READ AND PACKED
035D    FA                 MOV R2,A          ;DATA BYTE IS IN R2 NOW
035E    22                 RET
                                         DISPLAY:        ;POINTS  TO  MESSAGE
AND SEND THE
                                 DATA OF PROGRAMMING ADDRESS
035F    90 03 66           MOV DPTR,#MESPROG
0362    12 03 7E           CALL MESDISP
0365    22                 RET
0366                       MESPROG:
0366    0D 4F 2E 4B        .DB 0DH,"O",".","K"," ","A","T"," ",10,13,20H,FFH
036A    20 41 54 20
036E    0A 0D 20 FF


0372                                 SER_INI2:
0372    75 98 52           MOV SCON,#52H
0375    75 89 20           MOV TMOD,#20H ;20
0378    75 8D FD           MOV TH1,#FDH   ;FDH
037B    D2 8E             SETB TR1
037D    22                 RET

037E                       MESDISP:
037E    E4                 NEXT:   CLR A
037F    93                 MOVC A,@A+DPTR
0380    B4 FF 01           CJNE A,#FFH,OUTP
0383    22                 RET
0384    71 A9              OUTP:   ACALL TOUT
0386    A3                 INC DPTR
0387    61 7E              JMP NEXT


0389                       SER_INIT:
0389    75 98 52           MOV SCON,#52H
038C    75 89 20           MOV TMOD,#20H ;20
038F    75 8D F8           MOV TH1,#F8H   ;FDH
0392    D2 8E              SETB TR1
0394    22                 RET

0395    C0 E0              SERINT:   PUSH A
0397    C0 D0              PUSH PSW
0399    30 98 08           JNB RI,RETPT
039C    E5 99              MOV A,SBUF
039E    F5 43              MOV 43H,A
03A0    C2 98              CLR RI
03A2    D2 30              SETB FLAG
03A4    D0 D0              RETPT:    POP PSW
03A6    D0 E0              POP A
03A8    32                 RETI

03A9    30 99 FD           TOUT: JNB TI,$
03AC    C2 99              CLR TI
03AE    F5 99              MOV SBUF,A
03B0    22                 RET
03B1    74 48 49 53        MES:  .DB   "t","H","I","S",20H,"I","S","
                                       ","A"," ","T","E","S","T",10,13,255
03B5    20 49 53 20
03B9    41 20 54 45
03BD    53 54 0A 0D
03C1    FF

03C2                                 HEXASC:
03C2    FB                 A1:     MOV R3,A

03C3    94 0A              SUBB A,#0AH

03C5    40 04              JC NUMKEY
03C7    EB                 A-F:    MOV A,R3
03C8    24 37              ADD A,#37H
03CA    22                 RET
03CB    EB                 NUMKEY:  MOV A,R3
03CC    24 30              ADD A,#30H
03CE    22                 RET

03CF    C2 B5              END:    CLR P3.5
03D1    90 03 D8           MOV DPTR,#FINALMES
03D4    71 7E              CALL MESDISP
03D6    80 FE              SJMP $
03D8                                 FINALMES:
03D8    0D 0A 4F 56                  .DB  0DH,0AH,"O","V","E","R",10,13,FFH
03DC    45 52 0A 0D
03E0    FF
03E1    D2 B7              ERASE:    SETB P3.7
03E3    74 10              MOV A,#10H


03E5    F5 A0              MOV P2,A        ;OUTPUT TO PORT 2
03E7    7F 17              MOV R7,#17H
03E9    DF FE              DJNZ R7,$
                                                     ;ALE LOW VOLTS HIGH
03EB    C2 B7              CLR P3.7          ;VOLTS
03ED    C2 B6              CLR P3.6          ;ALE
03EF    7D 03              MOV R5,#3H
03F1    12 04 1C           DEL1:       CALL DELAY
03F4    DD FB              DJNZ R5,DEL1
03F6    D2 B6              SETB P3.6         ;ALE
03F8    D2 B7              SETB P3.7         ;VOLTS
03FA    90 04 01           MOV DPTR,#MESER1
03FD    71 7E              CALL MESDISP
03FF    21 F0              JMP S6_1
0401    45 52 41 53        MESER1: .DB  "E","R","A","S","E"," ","O","V","E","R","
                  ","N","o","w"," ","S","e","n","d"," ","d","a","t","a",10,13,FFH
0405    45 20 4F 56
0409    45 52 20 4E
040D    6F 77 20 53
0411    65 6E 64 20
0415    64 61 74 61
0419    0A 0D FF

041C                       DELAY:
041C    7F 0A              MOV R7,#10
041E    7E 46              TH:  MOV R6,#70
0420    DE FE              DJNZ R6,$
0422    DF FA              DJNZ R7,TH
0424    22                 RET
0425    C3                 ASCI_HEX:   CLR C
0426    FF                 MOV R7,A
0427    94 40              SUBB A,#40H
0429    40 05              JC ZTO9
042B    C3                 CLR C
042C    EF                 MOV A,R7
042D    94 37              SUBB A,#37H
042F    22                 RET
0430    C3                 ZTO9:     CLR C
0431    EF                 MOV A,R7
0432    94 30              SUBB A,#30H
0434    22                 RET
0435                                 .END
LINES ASSEMBLED : 456                ASSEMBLY ERRORS :    0
```

EFY.LST

```
2500 A.D. 8051 CROSS ASSEMBLER   -   VERSION 3.41f
-------------------------------------------------

       INPUT  FILENAME :  EFY.ASM
       OUTPUT FILENAME :  EFY.OBJ


0000               .ORG 0000H
0000    01 30      RESET: AJMP PROG
0030               .ORG 0030H
0030    75 B0 FF         PROG: MOV P3,#FFH
0033    75 81 60         MOV SP,#60H
0036    11 5D            ACALL INIT_LCD
0038    7C 10            MOV R4,#10H
003A    90 00 4D         MOV DPTR,#MESG
003D    74 01            MOV A,#01H
003F    12 00 8A         CALL CMD
0042    74 00            K1: MOV A,#0H
0044    93               MOVC A,@A+DPTR
0045    12 00 A8         CALL LCDWR
0048    A3               INC DPTR
0049    DC F7            DJNZ R4,K1
004B    80 FE            SJMP $
004D    45 6C 65 63      MESG: DB       "E","l","e","c","t","r","o","n","i","c",
                         "F","o","r","Y","o","u"
0051    74 72 6F 6E
0055    69 63 46 6F
0059    72 59 6F 75
005D
005D    74 38            INIT_LCD:MOV A,#38H
005F    11 8A            ACALL CMD
0061    12 00 DD         CALL LONGDELAY
0064    74 0E            MOV A,#0EH
0066    11 8A            ACALL CMD
0068    12 00 DD         CALL LONGDELAY
006B    74 06            MOV A,#06H
006D    11 8A            ACALL CMD
006F    12 00 DD         CALL LONGDELAY
0072    74 80            MOV A,#80H
0074    11 8A            ACALL CMD
0076    12 00 DD         CALL LONGDELAY
0079    E5 01            MOV A,01H
007B    11 8A            ACALL CMD
007D    12 00 DD         CALL LONGDELAY
0080    22               RET
0081    7D FF            DELAY: MOV R5,#FFH
0083    00               NOP
0084    00               NOP
0085    00               NOP
0086    00               NOP
0087    DD FE           DJNZ R5,$
0089    22              RET
008A    C2 B2            CMD: CLR P3.2
008C    C2 B3            CLR P3.3
008E    C2 B4            CLR P3.4
0090    F5 90            MOV P1,A
```

```
0092    D2 B4       SETB P3.4
0094    00          NOP
0095    00          NOP
0096    00          NOP
0097    00          NOP
0098    00          NOP
0099    00          NOP
009A    00          NOP
009B    C2 B4       CLR P3.4
009D    11 81       CALL DELAY
009F    11 81       CALL DELAY
00A1    11 81       CALL DELAY
00A3    11 81       CALL DELAY
00A5    11 81       CALL DELAY
00A7    22          RET
00A8    C2 B2       LCDWR:CLR P3.2
00AA    C2 B3       CLR P3.3
00AC    C2 B4       CLR P3.4
00AE    D2 B2       SETB P3.2
00B0    F5 90       MOV P1,A
00B2    D2 B4       SETB P3.4
00B4    00          NOP
00B5    00          NOP
00B6    00          NOP
00B7    00          NOP
00B8    00          NOP
00B9    C2 B4       CLR P3.4
00BB    12 00 C7    CALL BUSY
00BE    11 81       CALL DELAY
00C0    11 81       CALL DELAY
00C2    11 81       CALL DELAY
00C4    11 81       CALL DELAY
00C6    22          RET
00C7    C2 B2       BUSY: CLR P3.2
00C9    D2 B3       SETB P3.3
00CB    C2 B4       CLR P3.4
00CD    D2 B4       SETB P3.4
00CF    20 97 FD    JB P1.7,$
00D2    C2 B4       CLR P3.4
00D4    11 81       CALL DELAY
00D6    11 81       CALL DELAY
00D8    11 81       CALL DELAY
00DA    11 81       CALL DELAY
00DC    22          RET
00DD    7E FF       LONGDELAY: MOV R6,#FFH
00DF    00          L1: NOP
00E0    00          NOP
00E1    00          NOP
00E2    00          NOP
00E3    00          NOP
00E4    DE F9       DJNZ R6,L1
00E6    22          RET
00E7                END
LINES ASSEMBLED :   99      ASSEMBLY ERRORS :   0
```

LASER-BASED COMMUNICATION LINK

ANJAAN NANDI

An optical communication system for inter- and intra-building com muni-cations, closed-circuit TVs, PC LANs, etc can be built using the following three basic components:

1. A light-emitting element, which could be a laser diode or light-emitting diode (LED)
2. Transmission media, such as optical fibre or free space
3. A light-receiving element, which could employ avalanche photodiode, PIN photodiode (PIN-PD) or any other light sensor

Since the communications performance of the system depends on the overall characteristics of the above elements, the characteristics of the individual elements should match.

Here we present a one-/two-way optical communications system using a short-wavelength visible laser diode (say, RLT6505 or the laser module of a laser pointer) as the light-emitting element, free space as the transmission media and a light-dependent resistor as the light-receiving element. This system is ideal for speech communication between two adjacent offices or between homes on the opposite sides of a road.

The system has the following features:

1. Communication is possible up to several hundred metres. The communication range can be extended up to several kilometers by using a parabolic light reflector.
2. It transmits high-quality audio.
3. It ensures privacy, since a laser beam is very narrow and the link is virtually impossible for someone to tap into.
4. Alignment/orientation of the transmitter and the receiver is easy because the laser beam is visible.

It also offers the following facilities:

1. When someone intercepts the beam, the communication link breaks and the receiver circuit provides an audio-visual indication of the interruption by sounding an alarm and incrementing the count of a 7-segment display.
2. The person at the receiver end is alerted of an impending audio message through a buzzer sound by depressing a call switch at the transmitter end.
3. The voice output from the microphone in the transmitter is reproduced through a loudspeaker in the receiver section after suitable amplification.

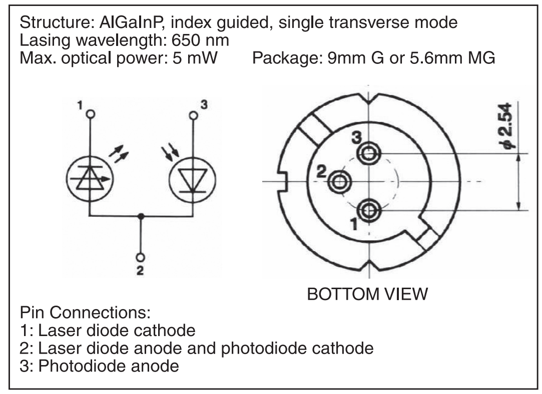


Fig. 1: Technical data of RLT6505G visible wavelength laser diode

Opto-Electrical Characteristics of RLT6505G (Tc=25°C)

Characteristic	Symbol	Test condition	Min.	Typ.	Max.	Unit
Optical output power	P_o	Kink free	—	—	5	mW
Threshold current	I_{th}	—	20	30	40	mA
Operation current	I_{op}	P_o=5 mW	—	45	70	mA
Operating voltage	V_{op}	P_o=5 mW	—	2.2	2.7	V
Lasing wavelength	I_p	P_o=5 mW	—	650	655	nm
Beam divergence	q_1	P_o=5 mW	5	8	11	°
Beam divergence	q_2	P_o=5 mW	25	31	37	°
Astigmatism	As	P_o=5 mW, NA=0.4	—	11	—	µm
Monitor current	I_m	P_o=5 mW, V_r=5V	—	10	—	µA

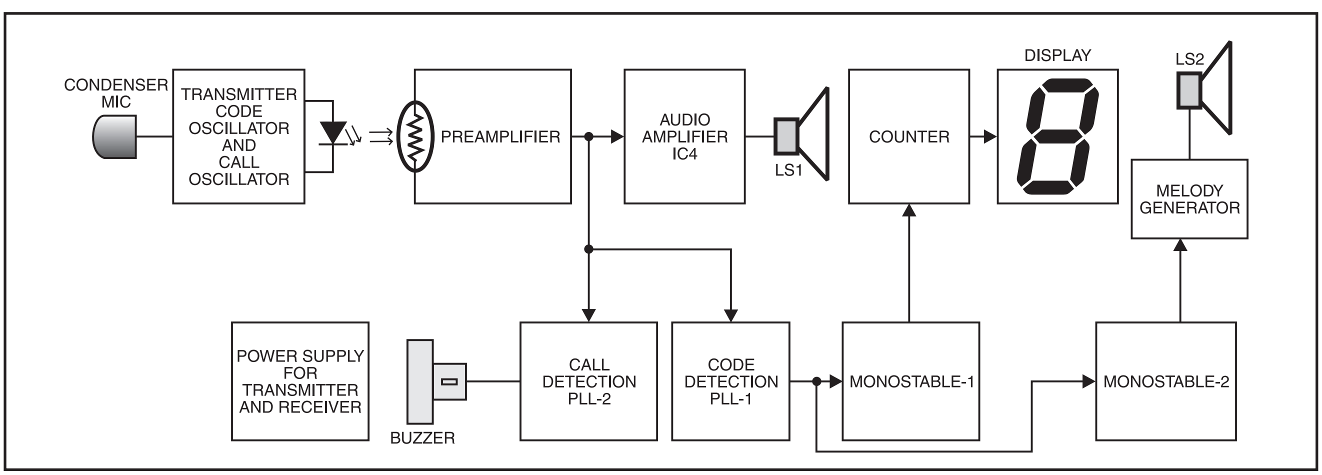


Fig. 2: Block diagram of the laser-based system for one-way speech communication

System description

Fig. 2 shows the block diagram of the laser-based system for one-way speech communication. It comprises transmitter, receiver and a common DC power supply section. The power supply section, at one end of the link, provides regulated 6V to the receiver-transmitter circuit. For two-way communication, you need to use an identical system, with the positions of the receiver and the transmitter reversed, with this system.

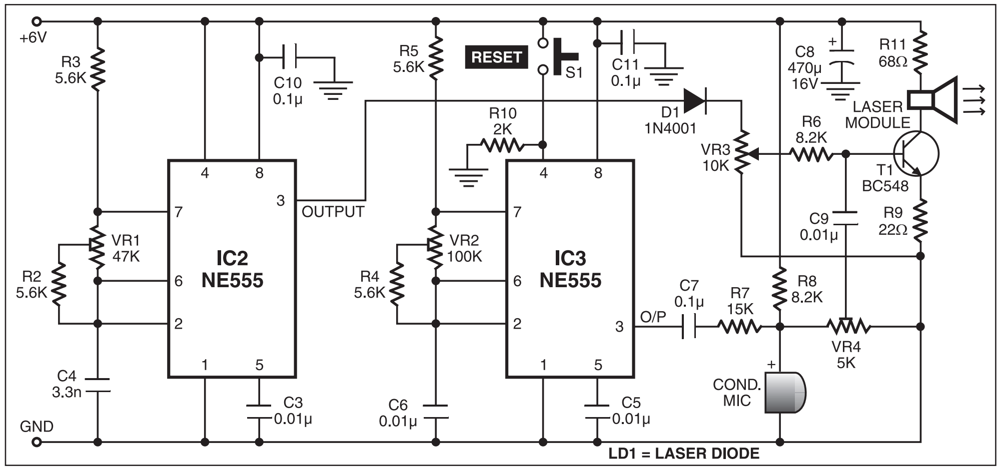


Fig. 3: Transmitter circuit

In the transmitter, the intensity of the laser beam is modulated by the output of an always-on code oscillator (operating at 10-15 kHz). Using a push-to-on switch, the tone oscillator (operating at 1-2 kHz) is momentarily activated to alert the person at the receiver end before starting a voice communication using the microphone.

The receiver receives the intensity-modulated light signals through a light sensor and outputs the code and 1kHz tone/voice.

The circuit for detecting the code signal is built around a phase-locked loop (PLL-1). The absence of the code signal indicates interception of the laser beam and activates an audio-visual warning at the remote receiver. For detecting the 1kHz call/tone signal, another phase-locked loop (PLL-2) is used. The call detection is indicated by a buzzer sound and an LED.

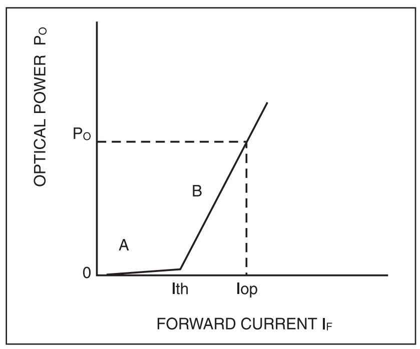


Fig. 4: Optical output vs forward current characteristics of laser diode

PARTS LIST

Semiconductors:	
IC1	- 7806 5V regulator
IC2, IC3, IC7, IC8	- 555 timer
IC4	- LM386 low-power audio amplifier
IC5, IC6	- NE567 phase-locked loop
IC9	- CD4033 decade counter/ 7-segment decoder
IC10	- UM66 melody generator
BR1	- 1A bridge rectifier
D1	- 1N4001 rectifier diode
ZD1	- 3.3V zener diode
LED1—LED3	- 5mm red LED
DIS1	- LTS543 common-cathode display
Resistors (all ¼-watt, ±5% carbon, unless stated otherwise):	
R1, R19, R20, R27, R32	- 1-kilo-ohm
R2-R5	- 5.6-kilo-ohm
R6, R8, R18, R21, R28	- 8.2-kilo-ohm
R7, R12	- 15-kilo-ohm
R9	- 22-ohm
R10	- 2-kilo-ohm
R11	- 68-ohm
R13, R17, R26	- 2.2-kilo-ohm
R14	- 2.7-kilo-ohm
R15	- 390-ohm
R16	- 390-kilo-ohm
R22	- 33-kilo-ohm
R23	- 4.7-ohm
R15	- 390-ohm
R24	- 36-kilo-ohm
R25	- 560-kilo-ohm
R29	- 4.7-kilo-ohm
R31	- 10-kilo-ohm
R30	- 220-ohm
VR1	- 47-kilo-ohm preset
VR2	- 100-kilo-ohm preset
VR5, VR6	- 10-kilo-ohm preset
VR3	- 10-kilo-ohm potmeter
VR4	- 5-kilo-ohm preset
VR7	- 10-kilo-ohm potmeter (log.)
Capacitors:	
C1	- 2200µF, 25V electrolytic
C2, C12, C13, C40	- 100µF, 16V electrolytic
C3, C5, C6, C9, C19, C22, C32, C35	- 0.01µF ceramic disk
C4,	- 3.3nF ceramic disk
C7, C10, C11, C14, C15, C17, C20, C26, C29	- 0.1µF ceramic disk
C8, C36, C38	- 470µF, 16V electrolytic
C16	- 56pF ceramic disk
C18, C27, C33, C39	- 10µF, 16V electrolytic
C21, C24	- 2.2µF, 16V electrolytic
C25	- 1nF ceramic disk
C23, C28	- 0.22µF ceramic disk
C30	- 47µF, 16V electrolytic
C31	- 1µF, 16V electrolytic
C34	- 6.8µF, 16V electrolytic
C37	- 3.3µF, 16V electrolytic
Miscellaneous:	
S1, S2	- Push-to-on tactile switch
LS1, LS2	- 8-ohm, 1W loudspeaker
Mic	- Condenser microphone
PZ1	- Piezobuzzer
X1	- 230V AC primary to 0-9V, 500mA secondary transformer
	- Laser module

The transmitter circuit

The transmitter circuit (shown in Fig. 3) consists of a code oscillator, tone/call oscillator, condenser microphone and an AF mixer stage.

The code oscillator comprising IC NE555 (IC2) is wired as an astable multivibrator operating at 10-15 kHz frequency. The actual oscillation frequency is decided by the timing components including resistors R2 and R3, preset VR1 and capacitor C4. We can adjust VR1 to vary the oscillation frequency to match with the centre frequency of PLL-1 at the remote receiver end. The output of IC2 is fed to the base of mixer transistor T1 via diode D1 and level-control potmeter VR3 and resistor R6.

Similarly, the tone/call oscillator comprising IC NE555 (IC3) is wired as an astable multivibrator to provide a 1-2kHz tone when tactile switch S1 is depressed. We can adjust VR2 to change the tone frequency to match with the centre frequency of PLL-2 at the remote receiver end. Resistor R10 is used to pull reset pin 4 of IC3 low when switch S1 is open.

The output of IC3 is also coupled to the base of the mixer transistor via capacitor C7, resistor R7, preset VR4 and capacitor C9. Preset VR4 is connected across the condenser microphone to adjust the audio signals when someone speaks into the microphone. Preset VR4 is used to vary the biasing signals.

The outputs of IC2 and IC3 and

voice signals are mixed by transistor T1 to drive the laser-pointer LED. The mixer output modulates the intensity of light signals emitted by the laser diode module in accordance with the level of the code oscillator and tone or audio signals available at the base of the mixer transistor.

Laser. The laser diodes can be constructed using a variety of different materials to produce distinctive wavelengths. Semiconductor laser diodes produce a much higher output power and highly directional beams compared to the LEDs. The laser must be operated with a large drive current to get a high density of ready-to-combine electrons at the p-n junction. Fig. 4 shows the optical output vs forward current characteristics of a laser diode. We can divide it into spontaneous-emission region A and laser-oscillation region B. The current required for starting oscillations is called the threshold current (I_{th}), while the forward (excitation) current necessary for maintaining the diode's specified optical output is called its operating current (I_{op}).

For the 5mW laser shown in Fig. 1, the typical values of threshold and operating currents are 30 mA and 45 mA, respectively. Keychain laser pointers available in the market have a power output of about 5 mW with forward current limited to 20 to 25 mA. Thus, a laser diode module of keychain-type visible laser pointer may be used for this transmitter circuit.

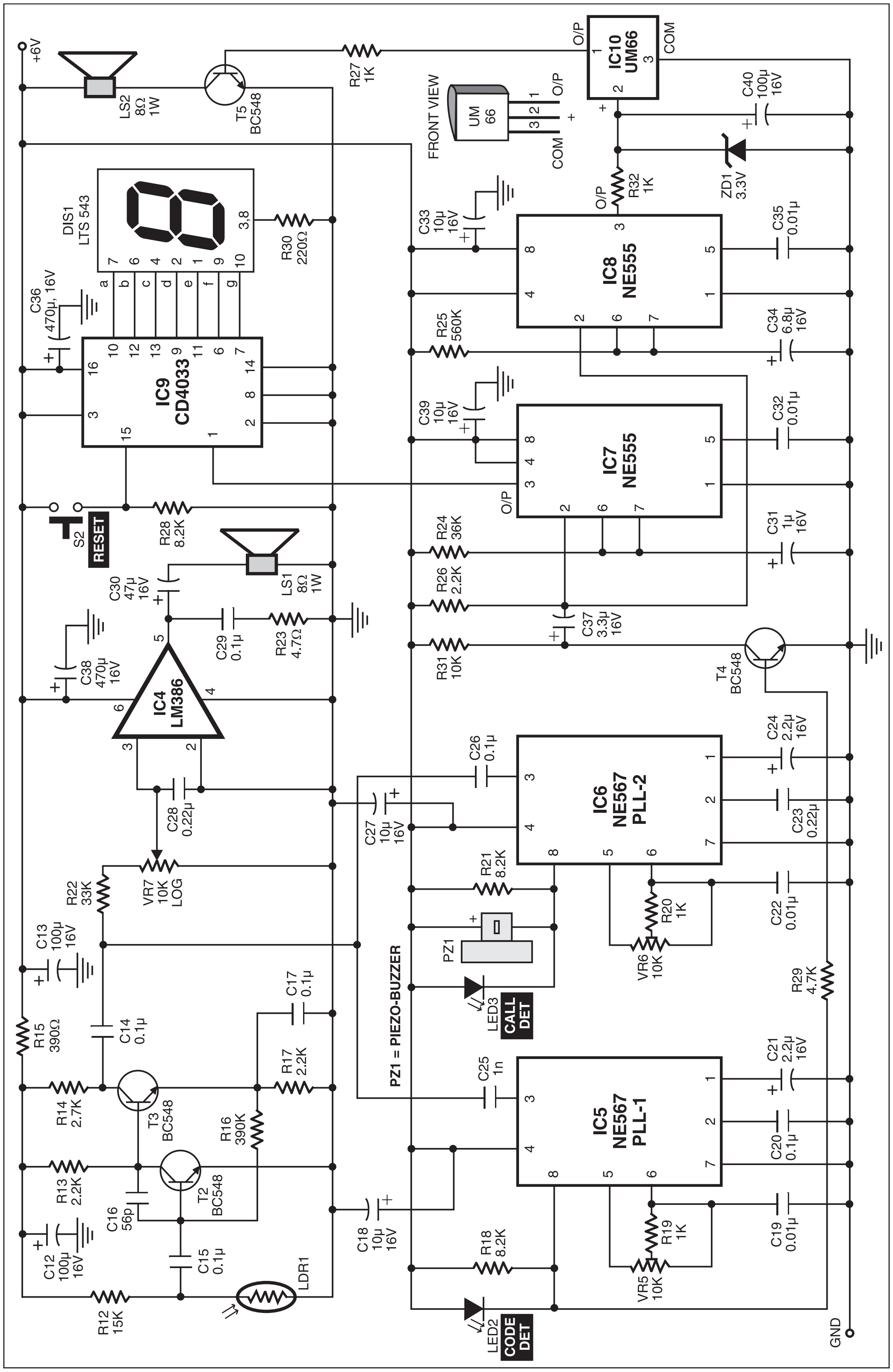


Fig. 5: Receiver circuit

The receiver

The receiver (Fig. 5) consists of a light sensor, a signal preamplifier, audio amplifier, code detector (with audio/visual alarm) and call/tone detector with buzzer indication. It uses a light-dependent resistor (LDR) as the light sensor. The resistance of LDR varies depending on the incident

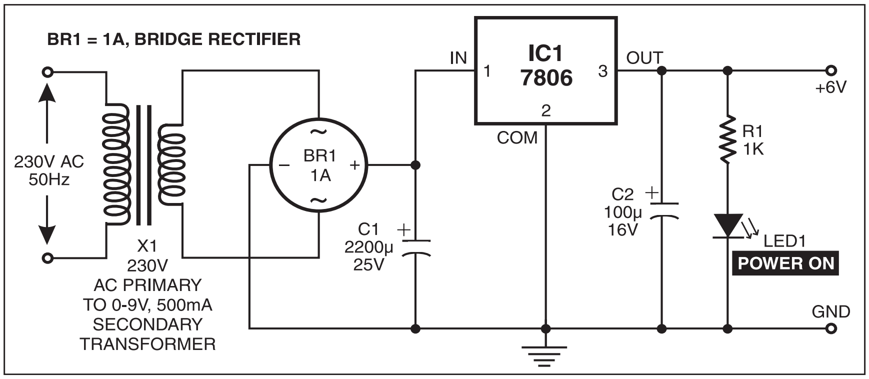


Fig. 6: Power supply circuit

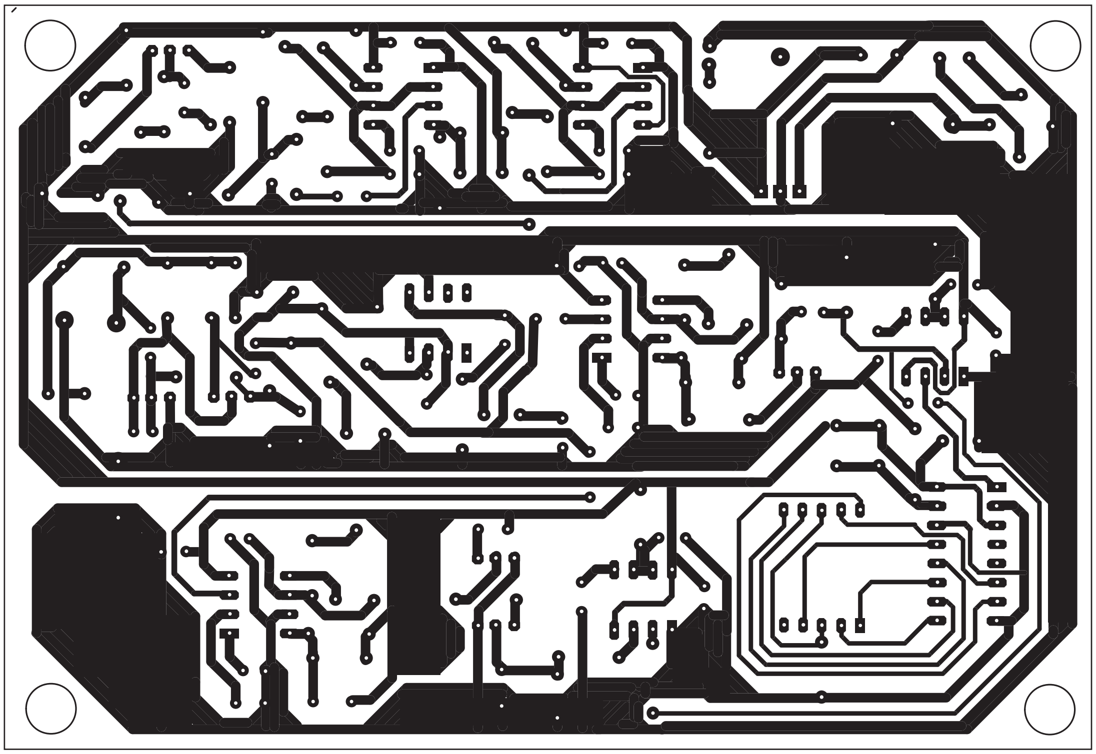


Fig. 7: Actual-size, single-side PCB layout of one-way speech communication circuit

light intensity, which, in turn, is a function of its modulation by the mixed output of code and tone or audio signals at the transmitter mixer stage. The output of the LDR sensor is amplified by a two-stage transistorised preamplifier.

The preamplifier output is coupled (via DC blocking capacitor C14) to:

1. The audio power amplifier built around IC LM386
2. Phase-locked loop (PLL-1) IC5
3. Phase-locked loop (PLL-2) IC6

The preamplifier output is fed to input pin 3 of audio power amplifier LM386 (IC4) through volume-control potmeter VR7. Capacitor C28 bypasses the noise signal and higher-order frequencies representing the code signal (10-15 kHz). The audio output (comprising voice or tone signals) from pin 5 of IC4 is coupled to loudspeaker LS1 through capacitor C30. A snubber network comprising capacitor C29 and resistor R23 is used for output stability. IC LM386 is a low-voltage audio power amplifier. Its gain is internally set to 20 to keep external part count low.

The preamplifier output, as stated earlier, is also connected to phase-locked loop IC5 and IC6 (each NE567) through capacitors C25 and C26, respectively. IC NE567 is a highly stable phase-locked loop with synchronous AM lock detection and power output circuitry. It is primarily used as a frequency decoder, which drives a load whenever a sustained frequency falling within its detection band is present at its self-biased input. The centre frequency of the band and output delay are independently determined by external components.

Link continuity/discontinuity indication. IC5 is used to detect the 10-15kHz code. In the absence of any input signal, the centre frequency of its internal free-running, current-controlled oscillator is determined by resistor R19 and capacitor C19. Preset VR5 is used for tuning IC5 to the desired centre frequency in the 10-15kHz range, which should match the frequency of the code generator in the transmitter.

The output at pin 8 of IC5 remains low as long as the transmitted code is detected by IC5. As a result, LED1 lights up to indicate continuity of the optical link/path for communication.

When the laser beam is interrupted due to any reason, the output at pin 8 of IC5 goes high to drive transistor T4 and its collector voltage falls to trigger monostable circuits built around IC7 and IC8 (each NE555), respectively. As a result, the output at pin 3 of these ICs goes high for the predetermined time period. The time periods of timers IC7 and IC8 depend on the values of resistor-capacitor combinations R26-C31 and R25-C34, respectively. Since output pin 3 of IC7 is connected to pin 1 of decade counter CD4033 (IC9), it provides a clock pulse to counter IC9 to increment its count, indicating interruption of the laser-light beam. The current count is shown on a 7-segment display (DIS1) connected to the 7-segment decoded outputs of counter IC9. Resistor R30 is used as a current-limiting resistor in the common-cathode path of DIS1.

For frequent interruptions of the light beam, the output of decade counter IC9 keeps incrementing the count. After the count reaches '9,' the next interruption resets the counter and it starts afresh. The counter/display can also be reset manually by momentarily depressing press-to-on

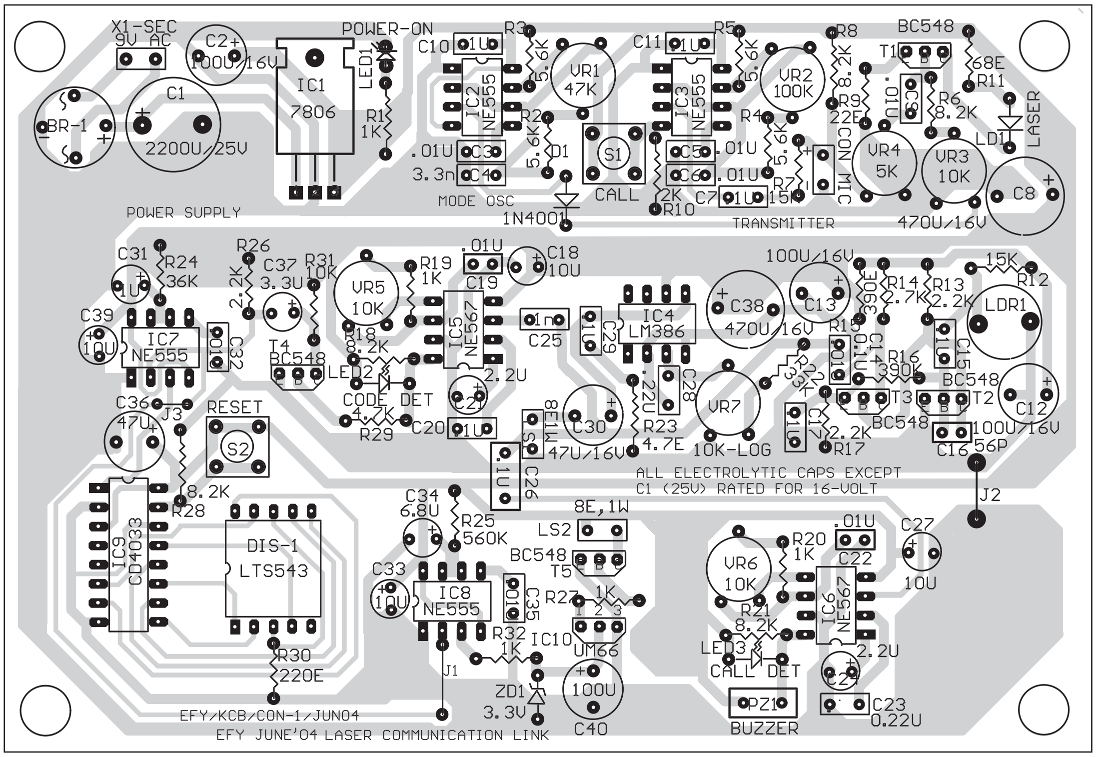


Fig. 8: Component layout for the PCB

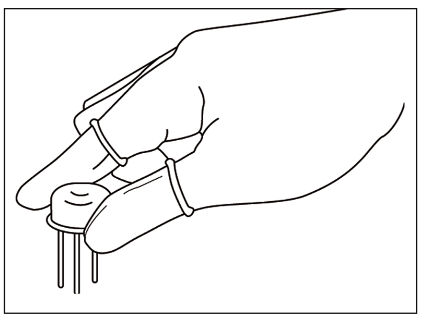


Fig. 9: Laser diode handling

switch S2.

As stated earlier, IC7 and IC8 are triggered simultaneously. Thus with each interruption of the light beam, the output of IC8 is pulsed high for a predetermined time to provide around 3V (determined by the output of zener diode ZD1) to melody IC UM66 (IC10). Thus IC10 generates a melodious tune whenever the light beam is interrupted. The output of IC10 is amplified by transistor T5 to drive loudspeaker LS2.

For initiating a call, the person at the transmitter end depresses switch S1 to alert the remote-end person of an impending voice communication. Thus the modulated light output from the transmitter contains 1-2kHz tone component in addition to the 10-15kHz code oscillator output. After detection and preamplification, 1-2kHz tone is decoded by PLL-2 circuit built around IC6, whose centre frequency is adjusted to match the frequency of tone/call oscillator in the transmitter.

IC6 is thus used as the call detector. Resistor R20 and capacitor C22 decide the centre frequency of its inbuilt oscillator in the absence of an input signal. Capacitors C23 and C24 serve as low-pass filter and output filter, respectively. Preset VR6 is used for tuning the inbuilt oscillator.

Thus when the 1-2kHz tone component is detected by IC6, its output pin 8 goes low to light up LED3 as also sound piezobuzzer PZ1 to alert the receiver-end person. Since the 1-2kHz tone component at the output of the preamplifier also passes through LM386 power amplifier, the tone is heard from loudspeaker LS-1 as well.

Voice communication. For voice communication, the person at the transmitter end speaks into the mike while call switch S1 is open. The modulated light beam contains the 10-15kHz code frequency and voice components. After demodulation at the receiver, the 10-15kHz code component is largely bypassed by capacitor C28 at the input of LM386, while the voice component (up to 3400 Hz) is attenuated insignificantly. Thus speech is reproduced at the output of LM386 via loudspeaker LS1. The code component (10-15 kHz) is detected by PLL IC5 signifying uninterrupted light path which is indicated by LED2, as explained earlier.

Construction

Fig. 6 shows the power supply circuit. The AC mains is stepped down by transformer X1 to deliver a secondary output of 9V AC at 500 mA. The transformer output is rectified by bridge rectifier BR-1. Capacitor C1 bypasses ripple and smoothes the rectifier output before regulation by 6V regulator 7806 (IC1). LED1 indicates power-on state. Resistor R1 acts as the current-limiting resistor for LED.

An actual-size, single-side PCB layout of the laser-based one-way speech communication circuit (comprising the transmitter, receiver and power supply units) is shown in Fig. 7 and its component layout in Fig. 8. For two-way (duplex) communication, you will need two PCBs.

Precautions

Take the following precautions while handling laser diodes:

1. For observing laser beams, always use safety goggles that block laser beams. Laser diodes up to 5mW output are ranked as Class III A products.

2. Laser diodes use gallium-arsenide (GaAs), which is potentially hazardous to the human body. Therefore, never crush, heat to the maximum storage temperature or put the laser diode in your mouth.

3. Semiconductor laser diodes are highly sensitive to electrostatic discharge, so be extremely careful while handling these. Don't touch the leads of the laser diode directly. Wear cotton gloves or ESD-protection gloves and handle the laser diode as shown in Fig. 9. ❑

DEVICE SWITCHING USING PASSWORD

CHARLS JOSEPH

Here's a password-based device switching circuit that stops unauthorised persons from switching on/off the devices. The circuit can switch on only one device at a time, out of a maximum of nine connected devices. To switch on/off the device, you need to enter a correct 4-digit password via the keypad.

Fig. 1 shows the block diagram of the device switching system using password. It mainly comprises a keypad, DTMF tone generator, DTMF decoder, demultiplexer and password circuit. Four DIP switches (DIP1 through DIP4) are used to set up the password.

The circuit

Fig. 2 shows the circuit for device switching using password. It can be divided into two sections, namely, the transmitter section and the DTMF decoder-and-password setup section.

The DTMF decoder-and-password setup unit is connected to the devices to be controlled. The DTMF generator (transmitter) is connected to the rest of the circuit through a two-core cable to enable device switching from a remote location.

1. The transmitter section. The transmitter circuit is built around DTMF encoder IC UM91214B (IC1). The DTMF encoder is commonly used as a dialler IC in telephone sets to generate DTMF tones. For its time base, IC UM91214B requires a 3.58MHz quartz crystal, which is connected between pins 3 and 4 of IC1 to form an internal oscillator. The oscillator output is converted into an appropriate DTMF signal through frequency division and multiplexing by the control logic of IC1.

A telephone type keypad is connected to ICI via 4-row and 3-column lines. Pins 15 through 18 of IC1 are row pins and pins 12 through 14 are column pins. Of the twelve keys on the keypad, we've used keys '1' through '9', '0' and '*.' The '#' key is not used here. Keys '1' through '9' are used for controlling the device, key '0' is used to switch off the device and key '*' is used to reset the circuit.

As stated earlier, we've used here a telephone-type keypad (also used for cash/debit card purchases) with twelve push-to-on switches to enter the password to control the devices.

When you press any key on the keypad, a unique pair of sinewave tones is produced, which is called dual-tone multifrequency (DTMF). These tone pairs lie within the audible frequency band of 300 to 2400 Hz and are chosen such that interference with any other frequency existing in the normal speech simultaneously is minimised. To minimise interference, a lower frequency from the rows (697 Hz, 770 Hz, 852 Hz or 941 Hz) is paired with a higher frequency from the columns (1209 Hz, 1336 Hz, 1477 Hz or 1633 Hz).

Thus a valid DTMF tone is the sum of a lower-frequency tone (697 Hz, 770 Hz, 852 Hz or 941 Hz) and

PARTS LIST

Semiconductors:	
IC1	- IC UM91214B DTMF dialler
IC2	- KT3170/MM8870 DTMF decoder
IC3	- 74LS154 4-to-16-line decoder/demultiplexer
IC4, IC5	- CD4015 dual 4-bit static shift register
IC6-IC9	- CD4030 quad Exclusive-OR gate
IC10	- 7408 quad 2-input AND gate
IC11-IC13	- CD4072 dual 4-input OR gate
IC14, IC15	- 74LS04 hex inverter
D1-D9	- 1N4007 rectifier diode
T1-T9	- BC548 npn transistor
ZD1	- 3.3V zener diode
Resistors (all ¼-watt, ±5% carbon, unless stated otherwise):	
R1	- 330-ohm
R2-R4	- 100-kilo-ohm
R5	- 330-kilo-ohm
R6-R14	- 4.7-kilo-ohm
Capacitors:	
C1	- 10µF, 10V electrolytic
C2, C3	- 39pF ceramic disk
C4, C5	- 0.01µF ceramic disk
Miscellaneous:	
X_{TAL1}, X_{TAL2}	- 3.58MHz crystal oscillator
S1	- On/off switch
DIP1-DIP4	- 4-way DIP switch
	- Keypad

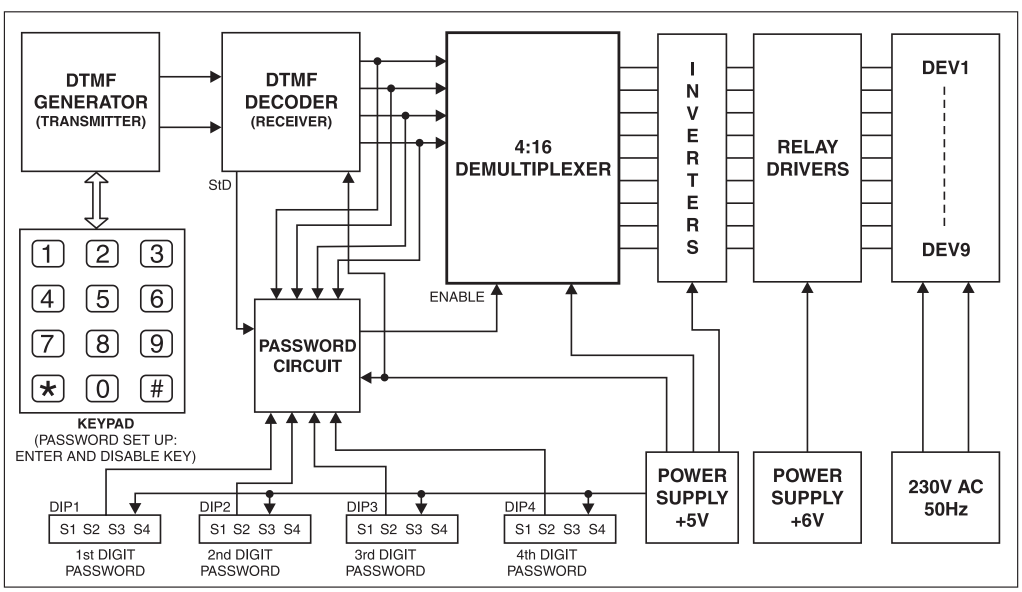


Fig. 1: Block diagram of the device switching system using password

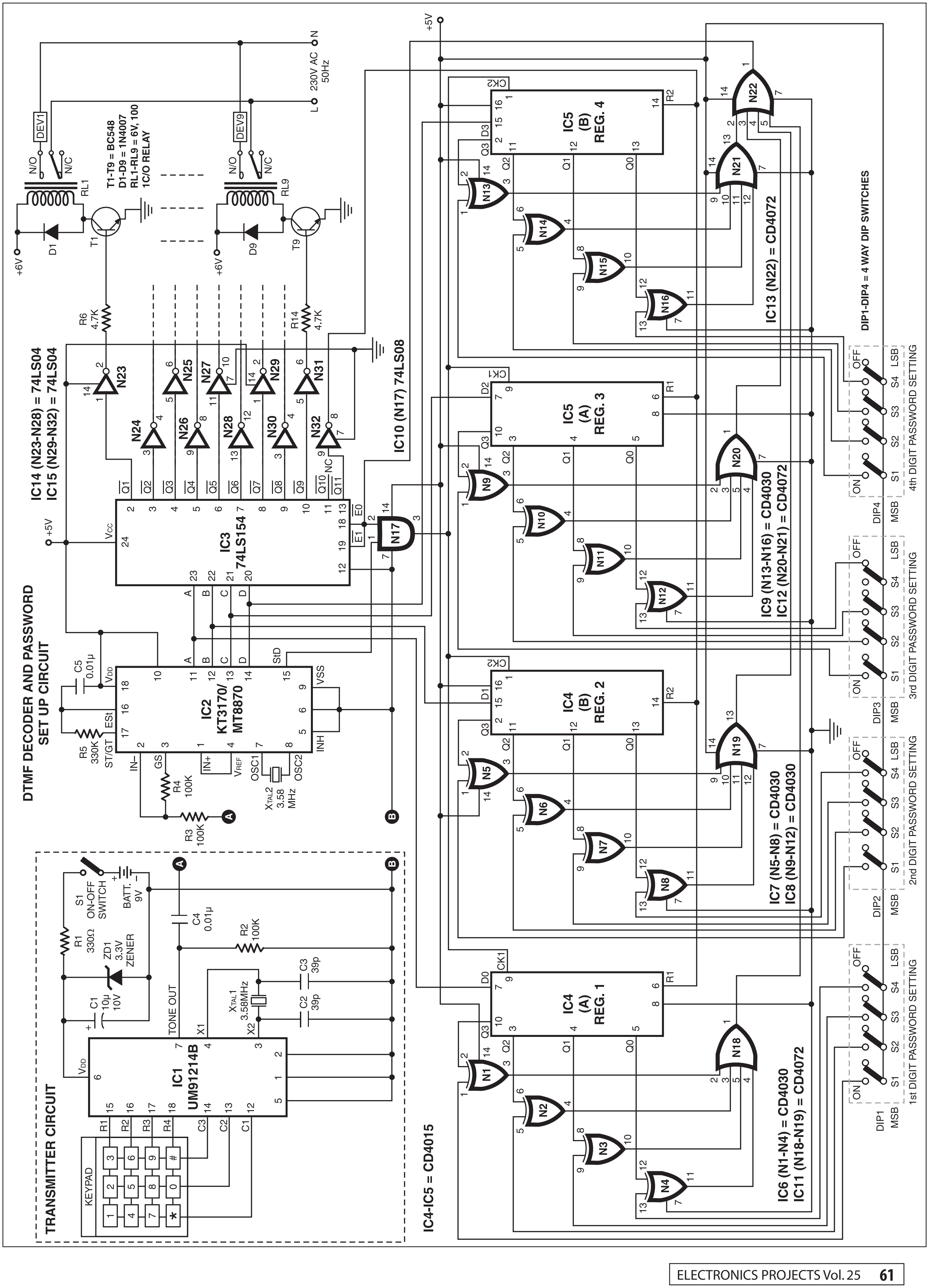


Fig. 2: Circuit for device switching using password

a higher-frequency group (1209 Hz, 1336 Hz, 1477 Hz or 1633 Hz). The DTMF dialling scheme allows 16 unique combinations of tones (codes), of which eleven codes have been used here. DTMF tones are so chosen that none of the tones is harmonic of the other tones. Therefore there is no chance of distortion caused by harmonics. Each tone is sent as long as a key remains pressed.

The DTMF coding scheme simplifies decoding because the composite DTMF signal may be separated using a bandpass filter into single frequency components, which may be handled individually.

If you press any key on the keypad, the corresponding DTMF tone pair output is available at pin 7 of the DTMF encoder (IC1). The tone output of IC1 is used as the input for the DTMF decoder (IC2).

2. The DTMF decoder. DTMF decoder KT3170/MT8870 (IC2) is used here. It uses a 3.58MHz crystal for providing clock for its internal circuitry. The DTMF decoder decodes the signal received from IC1 and provides a binary output corresponding to the key pressed in the transmitter circuit.

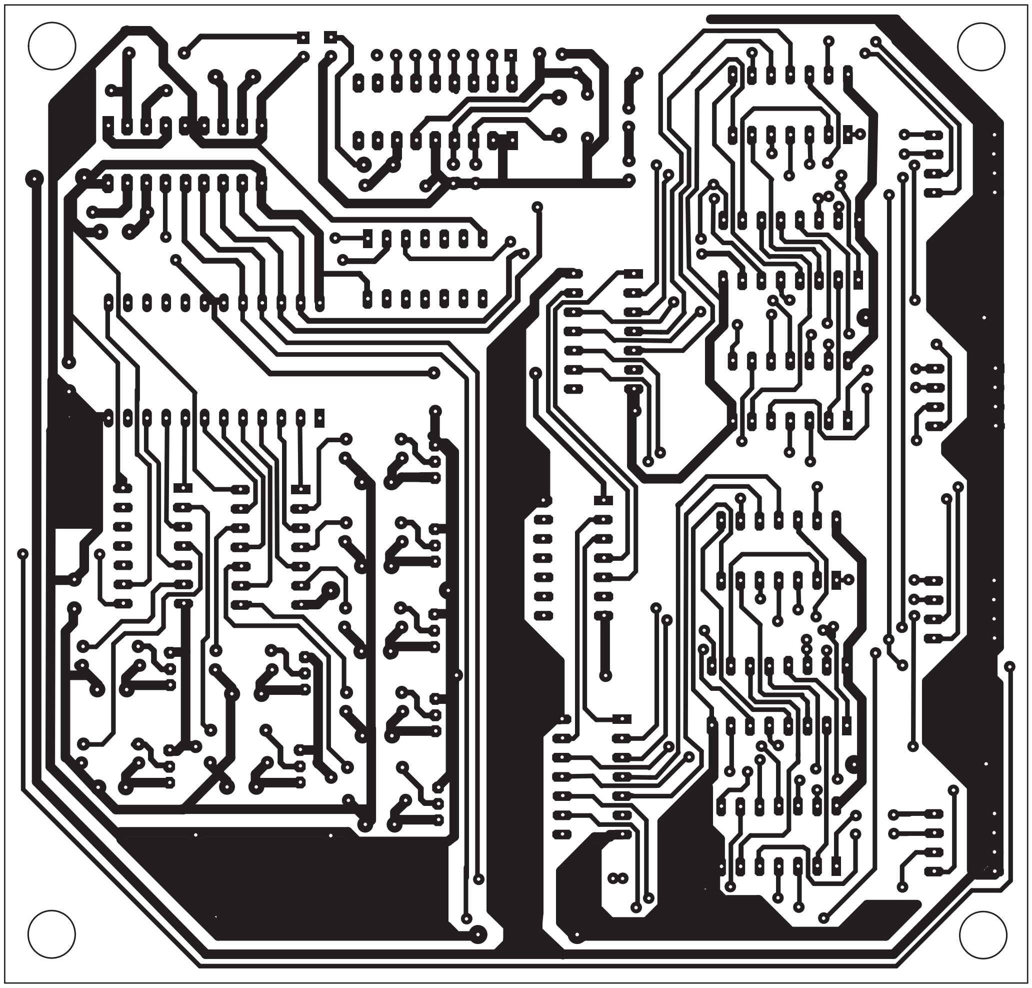


Fig. 3: Actual-size, single-side PCB for the circuit in Fig. 2

When you press any key on the keypad, IC2 receives a valid DTMF tone pair and decodes it into the corresponding 4-bit binary output, which is available at its pins 11 through 14. At the same time, its delayed steering output (StD) pin 15 goes high on receiving the tone pair, pulsing the clock pins of IC4 and IC5. The StD pulse is thus used to shift the data in dual 4-bit static shift registers of IC4 and IC5.

3. Password circuit. The password circuit is built around two dual 4-bit static shift registers (IC4 and IC5) and Ex-OR ICs (IC6 through IC9). IC6 through IC9 check whether the entered password is correct. The shift registers store the entered password in binary form. The stored number is cross-checked with the preset password with the help of Ex-OR ICs. The password is set by sliding the respective DIP switches DIP1 through DIP4, which are connected to the inputs of XOR gates.

When you press any key on the keypad, its binary code is output by decoder IC2. Bit one of the binary code is fed to shift register IC4(A), the second bit is fed to shift register IC4(B), the third bit is fed to shift register IC5(A) and the fourth bit is fed to shift register IC5(B). The clocks for IC4 and IC5 are generated by StD pin 15 of decoder IC2 via AND gate N17. The StD clocks shift data into shift registers IC4 and IC5.

The password in decimal numbers is set through the keypad. The corresponding binary numbers are fed through the DIP switches. The data across the outputs of IC4 and IC5, along with the password set by DIP switches, should result into a low output across the outputs of IC11 through IC13.

Once all the four digits of the password are entered, the 4-bit static data is available at the outputs of the shift registers (IC4 and IC5).

Password setting (refer table). Suppose you want to set up the password 9765. Press digits 9, 7, 6 and 5 sequentially and the password gets stored into shift registers IC4 and IC5 in binary format.

For the first binary digit (A bit), data stored into IC4(A) is 1101, so you have to set 1101 through switch DIP1 by making its S1 on (1), S2 on (1), S3 off (0) and S4 on (1).

For the second binary digit (B bit), data stored into IC4(B) is 0110, so

Password Setting Example

Keypad No. Seq.	Decoder output D	C	B	A	Register output
9	1	0	0	1	Q3
7	0	1	1	1	Q2
6	0	1	1	0	Q1
5	0	1	0	1	Q0
Reg. No.	4	3	2	1	

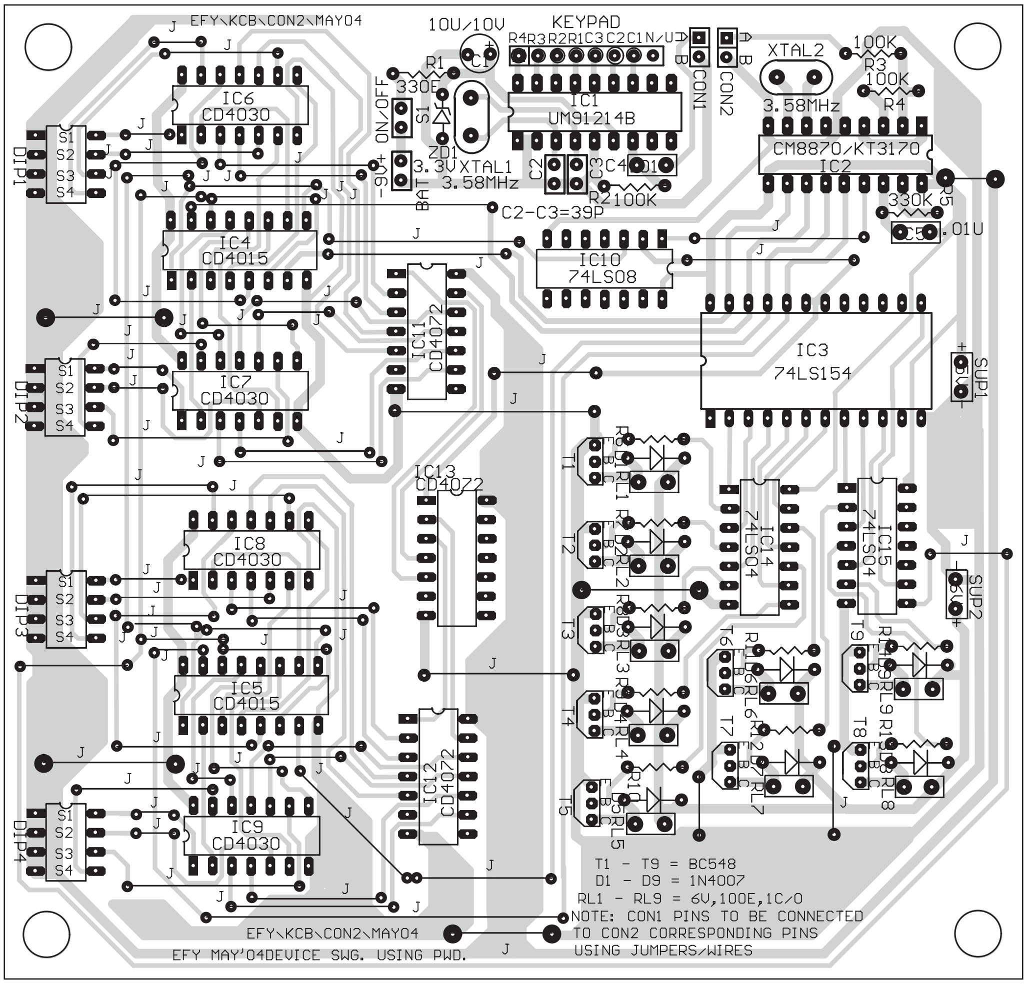


Fig. 4: Component layout for the PCB in Fig. 3

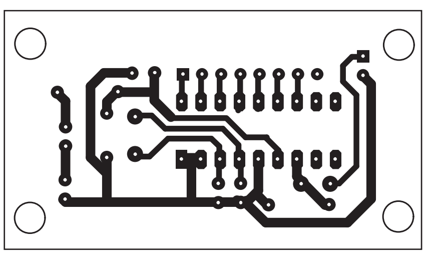


Fig. 5: PCB layout for transmitter

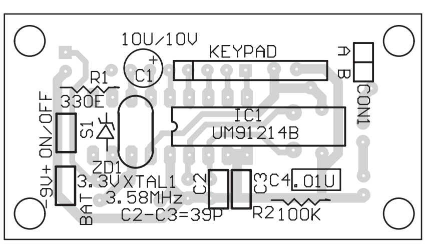


Fig. 6: Component layout for transmitter PCB

you have to set 0110 through switch DIP2 by making its S1 off, S2 on, S3 on and S4 off.

For the third binary digit (C bit), data stored into IC5(A) is 0111, so you have to set 0111 through switch DIP3 by making its S1 off, S2 on, S3 on and S4 on.

For the fourth binary digit (D bit), data stored into IC5(B) is 1110, so have to set 1110 through switch DIP4 by making its S1 on, S2 on, S3 on and S4 off.

Now your password is set and the circuit is ready to control the devices.

Password checking. IC6 through IC9 are used to check the password. If the password fed through DIP switches is correct, all the outputs of IC6 through IC9 go low and these are ORed by dual 4-input OR gates IC11 and IC12. Thus the outputs of gates N18 through N21 are low. The outputs of IC11 and IC12 are fed to IC13.

4. Enabling/disabling demultiplexer. The password is correct means that the inputs of gate N22 are low as these are connected to the outputs of gates N18 through N21. As a result, the output of gate N22 goes low, which enables the demultiplexer (IC3) for switching the appliance. Since the output of gate N22 is also connected to pin 2 of gate N17, it disables the clock signals of IC4 and IC5 at the same time.

If the password is wrong, the output of any one of gates N18 through N21 goes high. As a result, the output of gate N22 also goes high, which disables the demultiplexer (IC3). Since the output of gate N22 is also connected to pin 2 of gate N17, it enables the clocks signals of IC4 and IC5 at the same time. As a result, the appliance cannot be controlled.

5. Appliance on/off control circuit. The first four decimal digits you enter through the keypad form the password. Pressing the fifth decimal digit on the keypad switches on the device. Note that a particular device can be turned on only if you enter the corresponding decimal number on the keypad as the fifth number; for example, if you want to turn on device No. 1, press digit '1' on the keypad. Digit '0' turns off the device. Key '*' resets the circuit.

Suppose the 4-digit password you entered is correct. Now if you press '9,' which is the fifth digit entered by you, the respective device No. 9 turns on via relay RL9 and inverter N31. When you further press '0' key, which is the sixth key pressed by you, $\overline{Q10}$ output of IC3 goes low and device No. 9 turns off. When you press the '*' key, the $\overline{Q12}$ output of IC3 goes low and the circuit resets via inverter N32.

Fabrication

An actual-size, single-side PCB for the device switching circuit comprising both the transmitter (encoder) and the decoder (including password setting circuitry) is shown in Fig. 3 and its component layout in Fig. 4. However, if the transmitter circuit is to be used from a remote location, it needs to be separated. To meet this requirement, a separate PCB for the transmitter circuit is given in Fig. 5 and its component layout in Fig. 6. ❑

REMOTE-CONTROLLED SOPHISTICATED ELECTRONIC CODE LOCK

ARUP KUMAR SEN

Locking up valuables is a common practice to protect them from thieves. Various types of locks have been built for greater security. Among these is an electronic code lock, which ensures additional security by demanding a secret number (code) for opening the lock. Different types of circuits using different techniques have been developed for entering the code and its consequent processing.

Here's a sophisticated electronic code lock using the dual-tone multi-frequency (DTMF) signalling technique. The DTMF signalling technique improves signal readability even in a noisy environment.

This code lock has the following features:

1. The standard 12-digit telephone keypad is used for inputing the code.

2. The code here comprises only two digits. For greater security, the circuit can be modified to accommodate up to nine digits. However, this will require additional components.

3. The opener (operator) gets only two chances to input the code number for opening a lock. However, there is no limitation on closing the lock.

4. Two separate relays are provided: Relay A is used for opening the lock and relay B is used for closing the lock. The same code number is used for gaining access to the circuit for activating any of the relays.

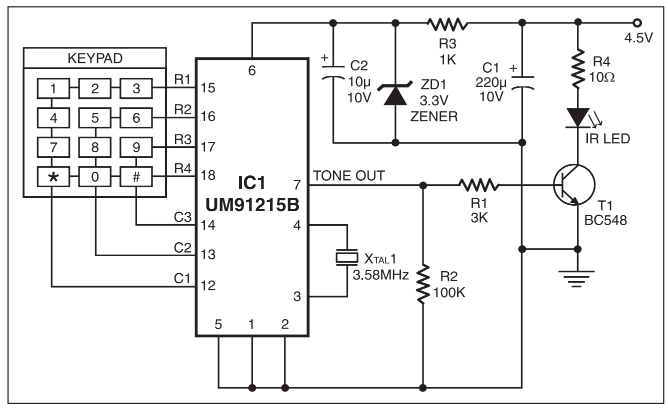


Fig. 2: Circuit of DTMF signal generator and transmitter

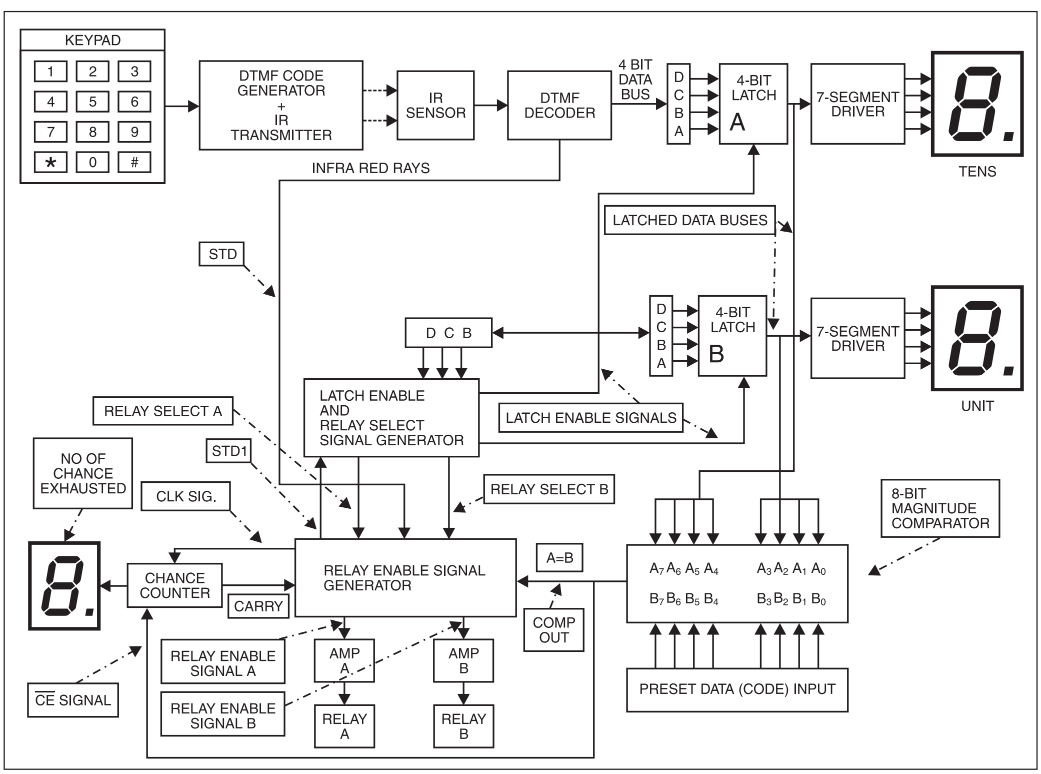


Fig. 1: Block diagram of remote-controlled sophisticated electronic code lock

Principle

When you press any key on the DTMF encoder, a DTMF signal is generated, which is first converted into a 4-bit equivalent binary/hexadecimal number by the DTMF decoder and then stored in a 4-bit latch. The two numbers generated due to pressing of two keys in sequence are stored in two different latches. The two latched numbers as a whole form the higher and lower nibbles of an 8-bit number.

Using a magnitude comparator, the resulting number is compared with another 8-bit number (code) applied to the comparator through two thumbwheel switches. If the two numbers match, the result of comparison is logic 1, which would allow the operator to switch on a relay by pressing a particular key from the keypad. The relay contacts would then activate a motor or a solenoid to open/close the door.

In case the numbers entered via keypad and thumbwheel switches don't match, pressing that very key would only advance a counter to decrease the allowed number of maximum chances for inputing the correct code. Once the maximum number of allowed attempts is over, the chance counter disables the input system, so pressing any key doesn't have any effect over the relays used for opening and closing the lock until and unless the chance counter is reset and correct code is entered

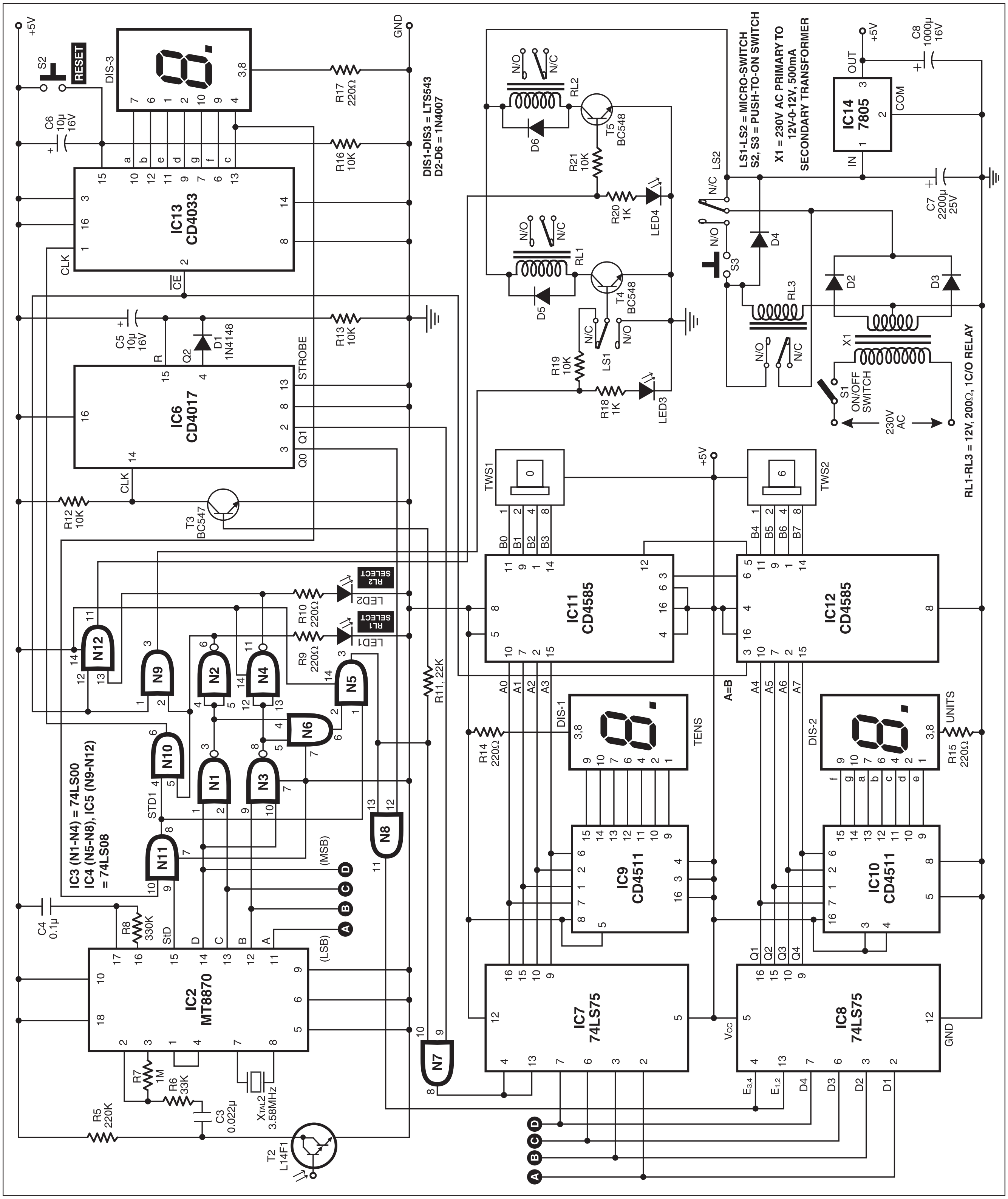


Fig. 3: The receiver, chance counter and relay drive circuit

PARTS LIST	
Semiconductors:	
IC1	- UM91215B DTMF dialler
IC2	- MT8870 DTMF decoder
IC3	- 74LS00 quad NAND gate
IC4, IC5	- 74LS08 quad AND gate
IC6	- CD4017 decade counter
IC7, IC8	- 74LS75 4-bit bistable latch
IC9, IC10	- CD4511 BCD-to-7-segment decoder/driver
IC11, IC12	- CD4585 4-bit magnitude comparator
IC13	- CD4033 7-segment decoder/driver
IC14	- 7805 +5V regulator
T1, T4, T5	- BC548 npn transistor
T2	- L14F1 phototransistor
T3	- BC547 npn transistor
ZD1	- 3.3V zener diode
D1	- 1N4148 switching diode
D2-D6	- 1N4007 rectifier diode
DIS1-DIS3	- LTS543 common-cathode 7-segment display
LED1, LED3	- Green LED
LED2, LED4	- Red LED
	- IR LED
Resistors (all ¼-watt, ±5% carbon, unless stated otherwise):	
R1	- 3-kilo-ohm
R2	- 100-kilo-ohm
R3, R18, R20	- 1-kilo-ohm
R4	- 10-ohm
R5	- 220-kilo-ohm
R6	- 33-kilo-ohm
R7	- 1-mega-ohm
R8	- 330-kilo-ohm
R9, R10, R14, R15, R17	- 220-ohm
R11	- 22-kilo-ohm
R12, R13, R16, R19, R21	- 10-kilo-ohm
Capacitors:	
C1	- 220µF, 10V electrolytic
C2	- 10µF, 10V electrolytic
C3	- 0.022µF ceramic disk
C4	- 0.1µF ceramic disk
C5, C6	- 10µF, 16V electrolytic
C7	- 2200µF, 25V electrolytic
C8	- 1000µF, 16V electrolytic
Miscellaneous:	
S1	- On/off switch
S2, S3	- Push-to-on switch
LS1, LS2	- Microswitch
TWS1, TWS2	- Thumbwheel switch
RL1-RL3	- 12V, 200-ohm, 1C/O relay
X1	- 230V AC to 12V-0-12V, 500mA secondary transformer
	- Reversible motor

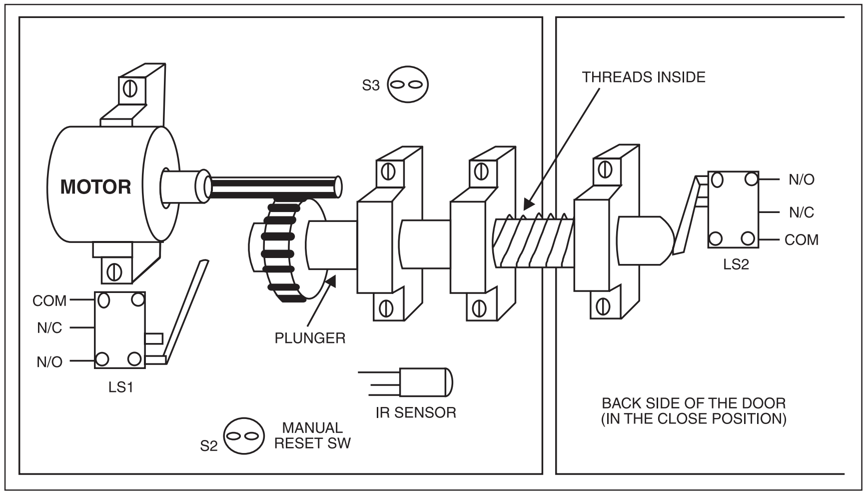


Fig. 4: Mechanical arrangement

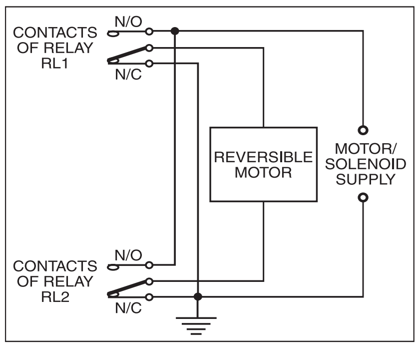


Fig. 5: Connection of reversible motor

via keypad.

Circuit description

Fig. 1 shows the block diagram of remote-controlled sophisticated electronic code lock. The entire circuit can be divided into two sections:

1. DTMF signal generator and transmitter
2. DTMF signal receiver, comparator and output relay driver

The DTMF signal generator and transmitter section is shown in Fig. 2.

Telephone tone/pulse dialler IC UM91215B is used for generating the DTMF signals. A DTMF signal is the algebraic sum of two different audio frequencies, and can be expressed as follows:

$f(t) = A.Sin(2pf_a t) + B.Sin(2pf_b t)$....(1)

where f_a and f_b are two different audio frequencies, with A and B as their respective peak amplitudes, and ¦ is the resultant DTMF signal. f_a belongs to low-frequency group and f_b belongs to high-frequency group.

Each of low- and high-frequency groups comprise four frequencies. From the various keys present on the telephone keypad, two different frequencies, one from the high-frequency group and another from the low-frequency group, are used to produce a DTMF signal to represent the pressed key. The amplitudes A and B of the two sine waves should be such that:

$0.7<(A/B)<0.9$...........(2)

The frequencies are chosen such that these are not the harmonics of each other. The frequencies associated with various keys on the keypad are given in Table I.

From Table I it is clear that if key 3 is pressed, 1477 Hz from the high-frequency group and 697 Hz from the low-frequency group produce the corresponding DTMF signal.

The DTMF signals generated due to pressing of different keys modulate the infrared (IR) rays generated by an IR LED. Transistor T1 (BC548) acts as the modulator. Normally, the LED is off. But when a DTMF signal is applied at the base of the transistor, the LED starts emitting IR rays due to varying collector current of transistor T1.

Dialler IC UM91215B (IC1) needs only 3 volts for its operation, but at least 4 volts should stay across the IR LED for effective transmission. Hence a supply of 4.5 volts is used. Three pencil cells in series can provide the required voltage. The supply for IC1 is regulated by zener diode ZD1.

Fig. 3 shows the DTMF signal receiver, chance counter and relay driver circuit.

When Darlington phototransistor T2 (L14F1) receives the modulated IR rays from IR LED, it converts the IR pulse train into equivalent electrical signal and couples the same to DTMF decoder IC

TABLE I
Frequencies Associated With Various Keys on the Keypad

		High-frequency group			
		1209 Hz	1336 Hz	1477 Hz	1633 Hz
Low-frequency group	697 Hz	1	2	3	A
	770 Hz	4	5	6	B
	852 Hz	7	8	9	C
	941 Hz	*	0	#	D, I

TABLE II
Decoded 4-bit Output of IC2 Corresponding to Keys Pressed

Key pressed	Pin 14 (MSB) D	Pin 13 C C	Pin 12 B	Pin 11 (LSB) A
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
0	1	0	1	0
*	1	0	1	1
#	1	1	0	0
A	1	1	0	1
B	1	1	1	0
C	1	1	1	1
D	0	0	0	0

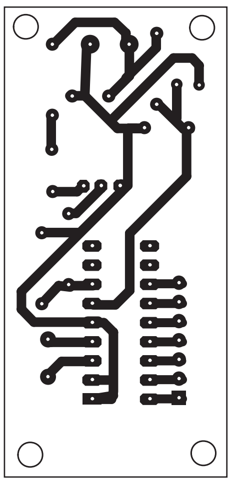


Fig. 6: PCB layout of DTMF signal generator and transmitter

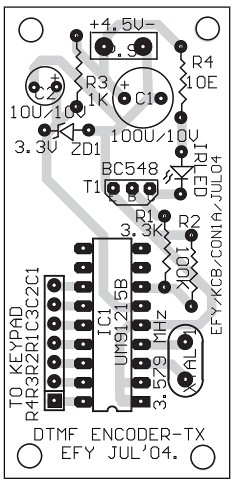


Fig. 7: Component layout for the PCB

CM8870 (IC2). If the signal is of sufficient amplitude and duration greater than the length of time predetermined by R8-C4 time constant, IC2 detects the signal and outputs a high-going pulse (StD) at its pin 15. The outputs at pins 11 through 14 of IC2 are the hexadecimal equivalent of the detected signal. Different decoded 4-bit numbers that would be generated due to pressing of different keys are shown in Table II.

The decoded number is latched in IC7 or IC8 depending upon the conditions governed by the latch-enable and relay-select signal generator logic circuit built around IC3, IC4, IC6 and transistor T3. The latched data from IC8 (74LS75) goes to BCD-to-7-segment decoder-cum-driver CD4511 (IC10). The decoded data at the output of IC10 is displayed on 7-segment display DIS2 (LT543). Similarly, data from another latch (IC7) is decoded by IC9 (CD4511) and displayed on 7-segment display DIS1. The two outputs together represent the 2-digit number entered from the keypad.

The outputs of latches IC7 and IC8 are also connected to 4-bit magnitude comparators IC11 and IC12 (each CD4585), respectively. Here, the combined output of the two latches is used as one of the two 8-bit numbers required by the magnitude comparator. Thumbwheel switches TWS1 and TWS2 are connected to comparators IC11 and IC12, respectively, for setting the 8-bit code. If the latched data inputs A0 through A7 from keypad and B0 through B7 from the thumbwheel switches are equal, the composite comparator outputs logic 1 at pin 3 of IC12. Output pin 3 is designated as A=B. When A=B is high, either relay A or relay B can be energised depending upon the signal from the relay-enable signal generator built around IC5.

The circuit is powered by 230V AC mains using switch S1. The AC mains is stepped down by transformer X1 to deliver a secondary output of 12V-0-12V at 500 mA. The transformer output is rectified by diodes D2 and D3 and smoothed by capacitor C7. Regulator 7805 (IC14) provides regulated 5V supply, which is connected to the entire circuit via normally closed (N/C) contacts of limit microswitch LS2. Another limit microswitch LS1 is connected to the base of transistor T4.

The status of limit microswitches LS1 and LS2 depends upon the position of the door-locking plunger. In the unlocked condition, the plunger stays in its retarded state remote from limit microswitch LS2, and the N/C contact of LS2 allows current to the circuit. On the other hand, the N/C contacts of limit microswitch LS1 are cut-off by the plunger and hence relay RL1 cannot be energised. However, relay RL2 can be energised.

If the plunger is moved forward to lock the door (using relay RL2), the plunger pushes limit microswitch LS2. When the plunger is completely advanced, it breaks the N/C contacts of microswitch LS2 and hence the connection of the circuit with +12V power supply. Being disconnected with the power supply, relay RL2, and consequently the motor/solenoid driving the plunger, goes off.

To resume the supply for unlocking, one has to press push-to-on switch S3. Consequently, the relay RL3 gets supply and pulls its armature. Even if S3 is released now, relay RL3 would still be in the energised condition, getting supply through its N/O contacts and providing supply to the circuit.

When the plunger is moved forward from its retarded position, microswitch LS1 frees itself and reconnects to the base of transistor T4, allowing relay RL1 to be activated. If the plunger is moved back to open the door (using relay RL1), limit microswitch LS1 would again be pushed and disconnect from the base of transistor T4, stopping the supply to the motor/solenoid. Thus the two microswitches also act as the limit switches for the motor.

Working of the circuit

When the circuit is switched on, counters IC6 (CD4017) and IC13 (CD4033) are reset by the power-on-reset citcuits comprising R13 and C5, and R16 and C6, respectively. So pin 13 of IC13 and Q0 output of IC6 both go high.

Now, if any key is pressed, and the generated IR ray having sufficient amplitude falls on phototransistor T2, the decoded data would be available at the outputs of IC2. The StD pulse from pin 15 of IC2 goes to pin 9 of IC5. Since pin 10 of IC5 is already high due to pin 13 of IC13, the output of AND gate N11 would be a high-going pulse having duration equal to StD. This output pulse would make pin 1 of AND gate N5 high. AND gates N5 and N6 of IC4 together form a 3-input AND gate, which receives inputs at pins 4 and 5 from NAND gates N1 and N3 of IC3, respectively.

Normally, the outputs of NAND gates N1 and N3 are high if none of the keys '0', '*' and '#' is pressed (refer Table II), so pin 2 of AND gate N5 is also high. Pin 3 of AND gate N5 goes high whenever pin 8

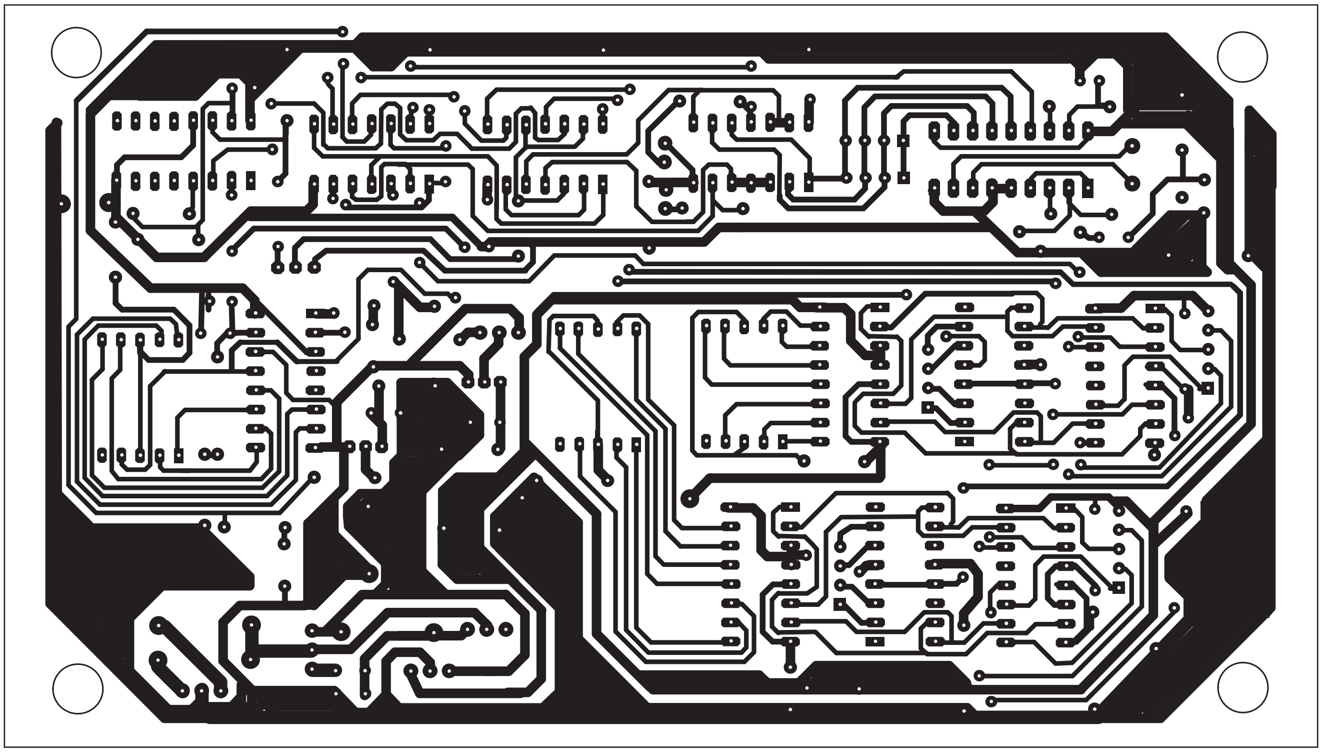


Fig. 8: Actual-size, single-side PCB layout of the receiver, chance counter and relay driver circuit

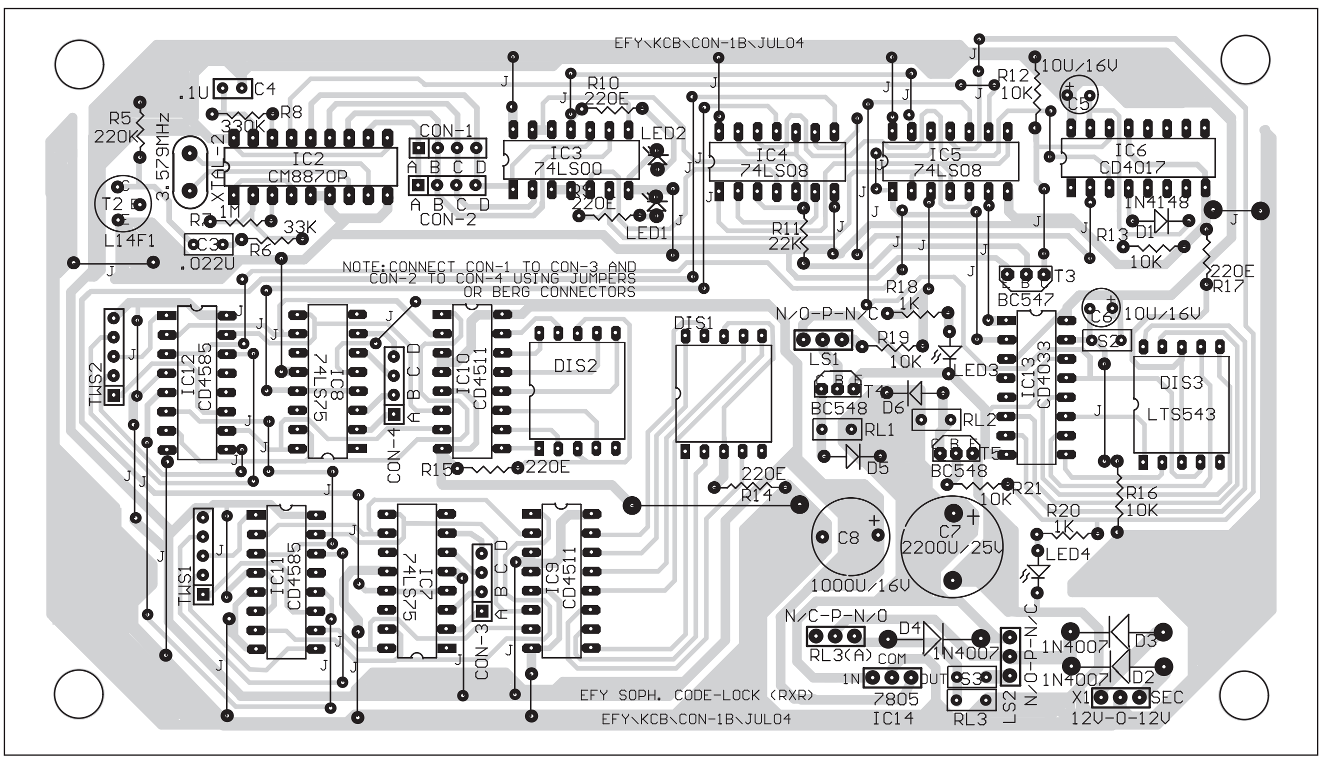


Fig. 9: Component layout for the PCB

of N11 goes high. Since pin 10 of N7 and pin 13 of N8 are tied to pin 3 of N5, these would also go high. Pin 12 of N8 is already high by the Q0 output of IC6 (CD4017). The output of N8 goes to latch-enable pins 4 and 13 of IC8.

The 4-bit data output of IC2 goes to latches IC7 and IC8. The StD pulse from IC2 forward biases transistor T3 to generate a clock pulse at its collector. This pulse being applied to the clock input of counter IC6 (CD4017), the counter advances by one and its Q1 output toggles from low to high state.

Now, if another key is pressed, the corresponding hexadecimal number is latched to IC7. At the end of this latching process, transistor T3 comes out of its saturated state and again applies a clock pulse to counter IC6. The counter advances by one to make its Q2 output high. Since Q2 output is tied to reset pin 15 of IC6, it immediately resets IC6 and its Q0 output goes high again. The process continues as long as one goes on pressing keys, except '0,' '*' and '#.'

When '#' key is pressed, the output of NAND gate N1 goes low as both of its inputs get high pulse from IC2. The 3-input AND gate formed by N5 and N6 is disabled, hence STD1 pulse is not allowed to change the counter CD4033 state or enable any latch to change its previously latched data.

The relay RL1 driving signal would be high if both the inputs of N11 are high. Pin 1 of N9 is fed by the composite comparator's output at pin 3 (A=B) of IC12. So pin 1 of N9 would be high if the numbers latched in IC7 and IC8 are equal to the number preset by thumbwheel switches TWS1 and TWS2. The input at pin 2 of gate N9 will be high when '#' key is pressed. Output at pin 3 of gate N9 is used to generate the relay RL1 select signal and clock for IC13.

So for driving relay RL1, one has to enter the correct code, then press '#' key on the keypad. On the other hand, for driving relay RL2, one needs to press '0' key after entering the correct code from the keypad. The magnitudes of the relay drive signals from gates N9 and N12 are boosted by transistors T4 and T5, respectively.

Since the lock-opening code comprises only two decimal digits, the number of chances to open the lock has been limited to two to ensure security. This is achieved with a chance counter built around decade counter-cum-7-segment decoder CD4033 (IC13).

The power-on reset signal to counter IC13 is provided by capacitor C6 and resistor R16. The counter remains reset until '#' key is pressed. When '#' key is pressed, pin 5 of gate N10 goes high by the relay RL1 select signal. Pin 4 of the same gate also goes high by STD1 pulse if output pin 13 of IC13 is high. So the counter would get a clock pulse only when '#' key is pressed and its output pin 13 is high. But the clock pulse would advance the counter by one only if the counter's chip-enable input (pin 2) is low.

Pin 2 of counter IC13 is connected to the output of the composite comparator (at pin 3 (A=B) of IC12). So if the correct code is entered from the keypad, the high A=B output would inhibit the counter from advancing. But if the entered code is wrong, the low A=B output would allow the counter to advance by one.

In this way, the counter tracks the number of failed attempts and displays the same on 7-segment display DIS3. If display DIS3 shows '1,' it means that one of the allowed chances have been exhausted.

The segment-c output (pin 13) of IC13 goes low with the exhaustion of two chances, which disables gate N11 and no STD1 pulse is generated further. So the input system would have no control over relay RL1 or RL2.

However, you can retry opening the lock by either of the following two ways:

1. Switching off the power supply to the circuit and then switching it on again to apply a power-on-reset to the chance counter.

2. Pushing manual reset switch S2 of the chance counter.

Construction

The transmitter part (acting as the key) is powered by a battery, so one can carry the same along with him. The lock system, including the IR receiver and relay driver circuit, is fitted on the back side of the door to be locked. The mechanical arrangement for the same is shown in Fig. 4. The manual reset switch, which you can use in the case of emrgency, must be kept hidden. You can mount it on the back side of the door such that in the case of emergency, you can access it from the front of the door by drilling a hole on the door. Drill a hole in front of the IR sensor (phototransistor T2) so that when the IR LED of transmitter is brought in front of the door, the emitted IR ray falls on the sensor. Mount the 7-segment displays on the front side of the door, so you can view the entered data code.

Alternatively, you can mount the entire transmitter-receiver combination on the back of the door. But, in that case, the keypad must be kept exposed for code entry from the front side of the door. The output of the transmitter can be connected directly to the receiver input, eliminating the need for infrared radiator. For the purpose, connect resistor R1 of the transmitter section directly to capacitor C3 of the receiver section after removing IR diode, transistor T1 (transmitter section), phototransistor T2 and resistor R5 (receiver section).

Whatever be the mounting option, it must be borne in mind that although IC2 (CM8870) is capable of detecting/decoding all the DTMF codes shown in Table I, only digits 1 through 9 can be used for formation of a code. The numbers representing '0,' '#,' and '*' keys haven't been used to form the code. Hence, the thumbwheel switches must be set to form a code between numbers 1 to 9 only.

Fig. 5 shows the connections of relays RL1 and RL2 to drive a single reversible-type AC motor. Instead of the motor, a solenoid can also be used to drive the plunger for opening or closing the door. If you use the solenoid, limit switch LS1 can be dispensed with to directly drive the base of transistor T4.

Steps for locking the door

1. Switch on power to the circuit using toggle switch S1.

2. Set the two thumbwheel switches to the desired code.

3. Align the two shutters of the door such that the plunger can move freely from one shutter to the other through the holes of the supports.

4. Switch on the DTMF transmitter.

5. Hold the IR LED transmitter close to the door such that the emitted IR ray falls on the IR sensor (phototransistor T2).

6. Enter code digits from the keypad and then press '0' key.

7. The motor starts running to rotate the plunger. The plunger moves forward due to screwing action of the threads over the surface of the plunger and inside the surface of supports. At the end of its journey, the plunger pushes limit microswitch LS2, cutting its N/C contact and hence the power supply to the receiver. Relay B goes off to cut power supply to the motor and hence the motor stops. The door is now locked.

Steps for unlocking the door

1. Push S3 momentarily. Relay RL3 immediately energises to power the circuit.

2. Switch on the DTMF transmitter and hold it close to the door such that the emitted IR ray falls on the sensor (phototransistor T2).

3. Enter the code from the keypad.

4. Press '#' key.

5. If the entered code is correct, relay RL2 energises and the motor starts running to rotate the plunger in reverse direction to disengage it from the supports. As soon as the plunger pushes limit microswitch LS1, the motor stops. Now you can push the shutters to open the door.

6. Switch off power to the circuit using switch S1.

7. If the entered code is not correct, the circuit gives you one more chance to unlock the door. Enter the code and press '#' again.

Emergency blocking

If you fail to enter the correct code in the allowed two chances, the input system would not accept any more signal from the IR transmitter until and unless the receiver is reset. Resetting can be done by either momentarily cutting the power to the circuit by using power switch S1 or by pressing manual reset switch S2. These switches should be kept hidden and used only in the case of emergency.

For greater security, you can increase the number of digits forming the code with some changes in the circuit. For a 3-digit code, you need to add another CD4585.

Actual-size, single-side PCBs for the transmitter and the receiver, chance counter and relay driver circuit are given in Figs 6 and 8, respectively, and their component layouts in Figs 7 and 9, respectively. ❑

Readers' comments

I have the following queries:

1. The code lock is working well but its range is 12.7 cm to 15 cm (5 to 6 inches) only. Why?

2. Why is '0' not used for formation of code?

3. How can I increase the range of the circuit?

Vivek
Through e-mail

The author, Arup Kumar Sen, replies:

Although IC CM8870 is capable of detecting/decoding all the 16 DTMF codes shown in the table here, only digits '1' through '9' can be used for formation of a code. Digit '0,' '#' and '*' are not used here to form the code. Since pressing '0' key produces the binary equivalent of decimal number '10' at the decoder output, formation of a code comprising decimal '0' is not possible, as it can't be compared by a standard thumbwheel switch that sets binary '000' for decimal '0'. Moreover, '0' key is used here for sending control signals.

Binary Output Across Pins 11 through 14 of IC CM8870

Key pressed (UM91215)	Pin 14 MSB D	Pin 13 C	Pin 12 B	Pin 11 LSB A
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	0
8	1	0	0	0
9	1	0	0	1
0	1	0	1	0
*	1	0	1	1
#	1	1	0	0
A	1	1	0	1
B	1	1	1	0
C	1	1	1	1
D	0	0	0	0

The low range could be due to improper orientation of the IR LED and the sensor (2N5777). Fig. 1 here shows the sensitive surface on the devices that transmits/receives the IR radiation.

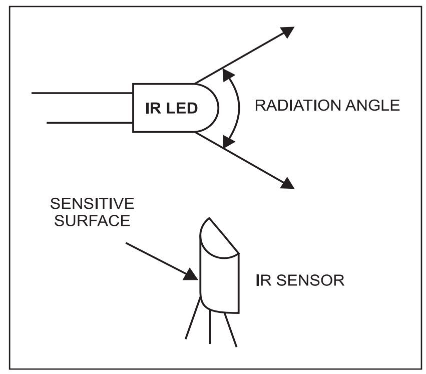


Fig. 1: Sensitive areas of the IR transmitter and receiver devices

IC 91215B can stand supply voltage of up to 5.5V, so supply voltage to the dialler may be increased slightly to extend the range. However, keep in mind that the gadget is not meant for use as a remote-controlled device. So a range of 12.7 to 15 cm is sufficient for opening or closing the door.

TEMPERATURE INDICATOR USING AT89C52

ADITYA RANE

Here's a microcontroller-based temperature indicator that displays the temperature in the range of –55°C to 125°C. Besides AT89C52 microcontroller, it uses a temperature sensor chip and an LCD module. The indicator outputs the calibrated data in digital form. The program for the microcontroller is written in C and not in Assembly language. Since C program has well-defined syntax, it far outweighs the merits of the Assembly language program.

The circuit

Fig. 1 shows the block diagram of the temperature indicator using microcontroller AT89C52. The power supply for the circuit is regulated by IC 7805 and supplied to different parts of the unit. DS1621 is the temperature sensor chip. The microcontroller unit (MCU) reads the temperature from the sensor. The temperature data is compared with certain user-defined temperature values and processed inside the MCU as per the program and then sent to the LCD for display.

Fig. 2 shows the circuit of temperature indicator using microcontroller AT89C52. Working of each section of the circuit is covered in the following paragraphs.

Power supply. The power supply unit consists of a step-down transformer (230V AC primary to 0-9V, 250mA secondary), bridge rectifier and voltage regulator. The output of the transformer is fed to bridge rectifier diodes D1 through D4 (each 1N4007). The ripple from the output bridge rectifier is filtered by capacitor C1 and fed to regulator IC 7805. The regulated output is given to the temperature sensor, microcontroller unit and LCD module, respectively.

When switch S1 is closed, LED1 glows to indicate the presence of power in the system.

Temperature sensor. Temperature sensor chip DS1621 (IC3) is an 8-pin DIP IC. Its pin details are shown in Fig. 3 and the internal block diagram in Fig. 4. The chip can measure temperatures from –55°C to +125°C in 0.5°C increments, which are read as 9-bit values. It can operate off 2.7V to 5.5V. Data is read/written via a 2-wire serial interface. Pins 1 and 2 of the temperature IC are connected to pins 11 and 10 of the microcontroller, respectively.

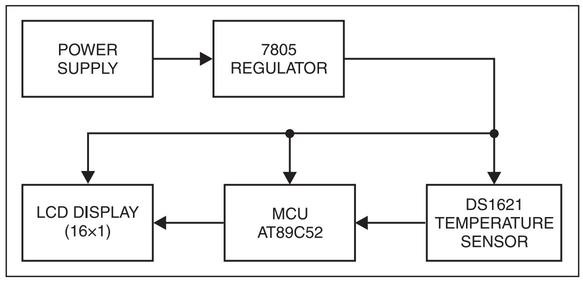


Fig. 1: Block diagram of temperature indicator using AT89C52

The thermal alarm output (T_{out}) of IC DS1621 activates when the temperature exceeds user-defined high temperature TH. The output remains active until the temperature drops below user-defined low temperature TL. User-defined temperature settings are stored in the non-volatile memory. Temperature settings and temperature readings are all communicated to/from IC DS1621 over a 2-wire serial cable. The most significant bit (MSB) of the data is transmitted first and the last significant bit (LSB) is transmitted last.

Addressing. The chip address of DS1621 comprises internal preset code nibble '1001' (binary) followed by externally configurable address pins/bits A2, A1 and A0. The eighth bit of the address byte is determined by the type of operation (either read or write) that is to be performed. For writing to the device the eighth bit is '0' and for reading from the device the eighth bit is '1.' In our case, A2, A1 and A0 pins are grounded and hence the device address for writing is '1001000b' or 90(hex) and for reading the device address is '10010001b' or 91(hex).

Configuration/status register. This register can be accessed for reading or writing by issuing command byte AC(hex) from the master (82C52). This register is particularly required if DS1621 is used for thermostat control, since it contains flag bits THF (high-temperature flag) and TLF (low-temperature flag) which are set to '1' when temperature crosses the respective limits earlier written into TH and TL registers. It also contains the flag bit (Done), which is set to '1' when results of conversion are available after issuing of start conversion command EE(hex). The other bits of configuration register are defined below:

'NVB' is the non-volatile memory busy flag, '1' is write to an E^2 memory cell in progress, '0' indicates that non-volatile memory is not busy, 'POL' is non-volatile output polarity bit ('1'=active-high and '0'=active-low) and '1SHOT' is one-shot mode. A copy to E^2 may take up to 10 ms. If 1SHOT is '1,' DS1621 will perform one temperature conversion upon reception of the Start Convert T protocol. If 1SHOT is '0', DS1621 will continuously perform temperature conversions. This bit is non-volatile.

Command Set. Complete command instruction set for accessing various internal registers as well as for starting and stopping of conversion process are given in Table I. For understanding the exact sequence in which Start bit, address byte, acknowledgement bit, command byte(s) and data byte(s) are to be sent along the I^2C bus, please refer to the datasheet of DS1621, wherein these aspects have been explained in proper detail. This will help

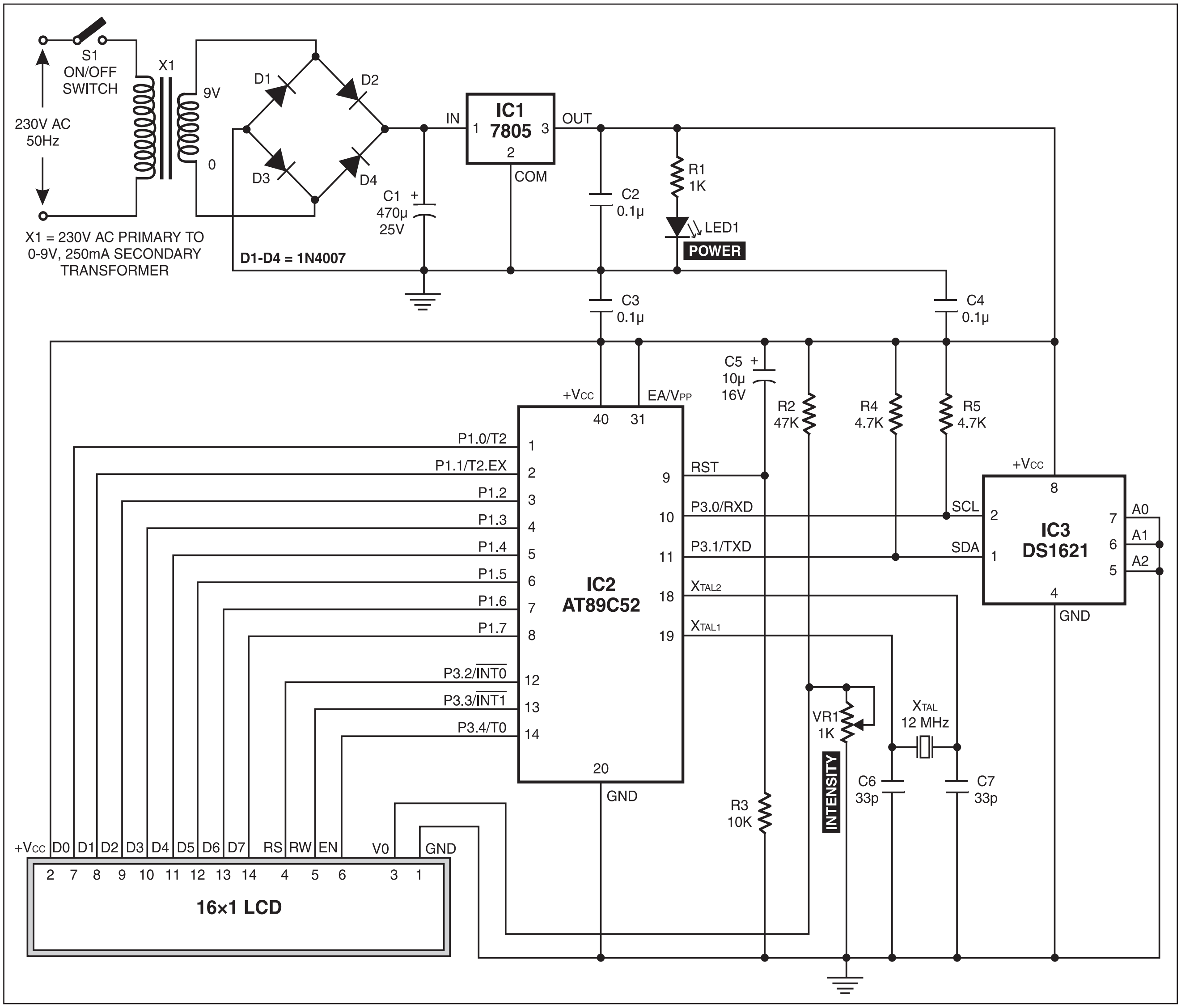


Fig. 2: Circuit diagram of temperature indicator using AT89C52

in understanding the contents of the main program.

Microcontroller unit. Microcontroller AT89C52 (IC2) is a 40-pin IC from Atmel. Its pin details are shown in Fig. 5. Like AT89C51, it also belongs to the 8031/8051 family. Microcontroller AT89C52 has a 256×8-bit internal random-access memory (RAM), eight interrupt sources and 8 kB of flash memory compared to 128x8-bit internal RAM, six interrupt sources and 4 kB of flash memory in AT89C51. By combining a versatile 8-bit CPU with flash memory on a monolithic chip, Atmel AT89C52 is a powerful, highly flexible and cost-effective solution to many embedded control applications.

Ports 0 and 2 are 8-bit bidirectional input/output (I/O) ports. These ports haven't been used in this temperature indicator.

Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. Ports 1.0 through 1.7 are connected to pins 7 through 14 of the LCD. Port-1 output buffers can sink/source four TTL inputs.

Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Ports 3.0 and 3.1 of IC2 are connected to serial clock line (SCL) and serial data line (SDA) of IC3, respectively. Ports 3.2 through 3.4 are connected to pins 4 through 6 of the LCD, respectively. Port-3 output buffers can sink/source four TTL inputs.

A 12MHz crystal oscillator is connected to X_{TAL1} and X_{TAL2} pins for operation of the microcontroller. A high pulse on RST pin (pin 9) while the oscillator is running resets the microcontroller. In this circuit, this pin is connected to +Vcc through capacitor C5 (10 µF, 16V). The external-access enable pin (EA) is connected to +Vcc for internal program executions. This pin also receives the 12V programming-enable voltage (V_{pp}) during flash programming when 12V programming is selected.

The program

The C-language program for microcontroller AT89C52 is compiled using cross-compiler C51 Version 7.10 from Keil Software. The demo version of this compiler is available for free on the Website 'www.keil.com.' It can compile programs up to 2 kB only, which is sufficient for writing most programs.

For testing the display, the program Hello.c is given here. This program,

PARTS LIST

Semiconductors:

IC1	- 7805 regulator IC
IC2	- AT89C52 microcontroller
IC3	- DS1621 temperature sensor
D1-D4	- 1N4007 rectifier diodes
LED1	- Red LED

Resistors (all ¼-watt, ±5% carbon, unless stated otherwise):

R1	- 1-kilo-ohm
R2	- 47-kilo-ohm
R3	- 10-kilo-ohm
R4, R5	- 4.7-kilo-ohm
VR1	- 1-kilo-ohm preset

Capacitors:

C1	- 470µF, 25V electrolytic capacitor
C2, C3, C4	- 0.1µF ceramic disk
C5	- 10µF, 16V electrolytic capacitor
C6,C7	- 33pF ceramic capacitor

Miscellaneous:

Transformer	- 230V AC primary to 0-9V, 250mA secondary
Crystal	- 12 MHz
LCD	- 16×1 LCD module
S1	- On/Off SPST switch

when loaded to AT89C52, displays "Hello! How R U?" on the LCD. The Hello.c program has nothing to do with temperature. It just guarantees a perfect communication between the LCD and the microcontroller. For temperature indication, the program Temp52.c is used. The programs Hello.c and Temp52.c, along with the hex files, are given at the end of this article.

The communication interface between the temperature sensor and the microcontroller chip follows the I²C (Inter Integrated Circuit) standard, which is implemented in 'C' here. I²C is a simple master/slave type interface. Simplicity of the I²C system is primarily due to the bidirectional 2-wire (SDA and SCL) design and the protocol format. Bidirectional communication is through 2-wire lines (which are either active-low or passive-high). In the program, the i2c_stop, i2c_start, i2c_write and i2c_read functions are used for communicating Clock and Data from DS1621 to P3.0 and P3.1 of AT89C52, respectively. Such functions as command, ready and display in the program are used for driving the LCD.

Program compilation for 8051 family controller. Keil C51 can compile C programs for most of the Atmel family microcontrollers. It also supports other devices. Unlike other cross-compilers (Hi-Tech, IAR, SDCC, etc), Keil C51 offers such features as fast code generation, strong multitasking environment, real-time operating system and inbuilt code optim-isation. To enjoy these features, you'll need full version of the compiler.

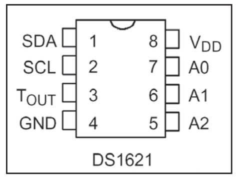


Fig. 3: Pin details of IC DS1621

Keil C51 has options to generate Assembly code and all the code listing supported by 8051 family, but Assembly language generated cannot be recompiled on any other assembler. As far as code generation is concerned, it uses minimum RAM and on-chip flash, allowing faster and optimised program in Intel-Hex format, which can be loaded to the microcontroller using any programmer. Conversion of C program into Intel-Hex format takes only a few seconds. In fact, you don't require all that long Assembly program in order to generate the output hex file.

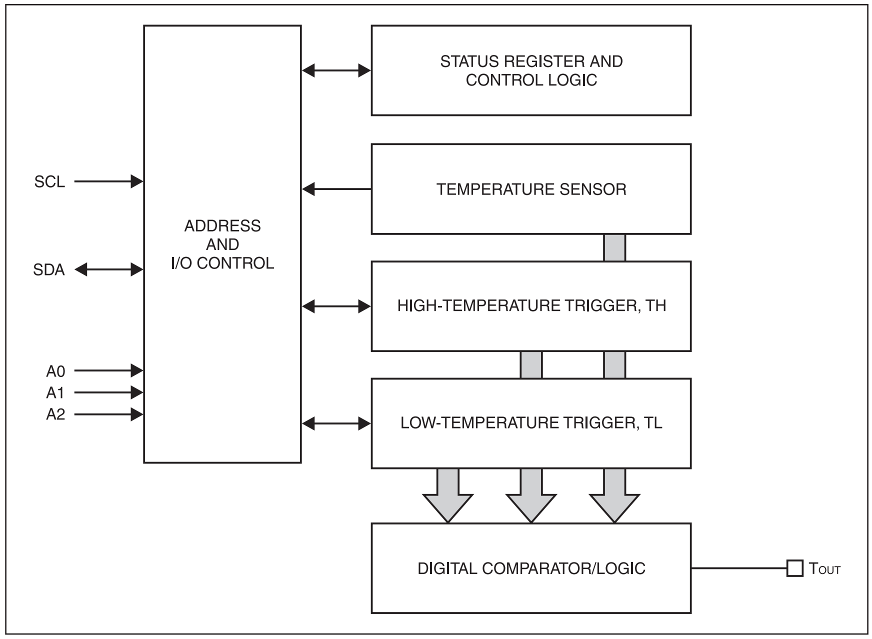


Fig. 4: Internal block diagram of IC DS1621

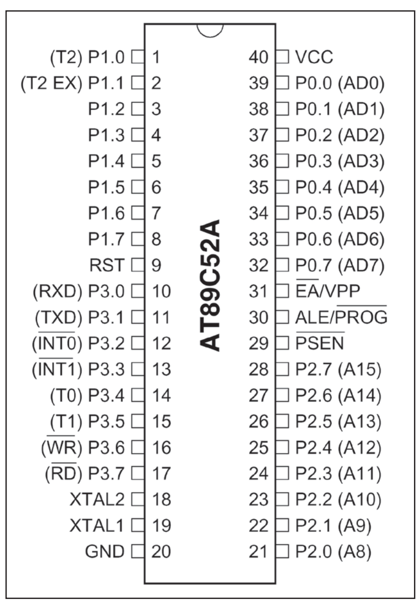


Fig. 5: Pin details of IC AT89C52

LCD

For display, a Lampex make 16x1 LCD (model GDM1601A) was used. Pin connections of this LCD are given in Table II. Pins 15 and 16 haven't been used. Pin 3 is connected to the circuit ground through a 1-kilo-ohm preset that is used to control the light intensity of the LCD. Note that the Hitachi make 16×1 LCD (HD44780A00) will not work in this project.

Construction

The circuit of this temperature indicator using microcontroller AT89C52 can be assembled on any general-purpose, single-side PCB. The microcontroller chip and the temperature sensor chip are mounted on the respective IC bases. Ensure a proper contact between pins of the IC bases and

DONE	THF	TLF	NVB	1	0	POL	1SHOT
MSB							LSB

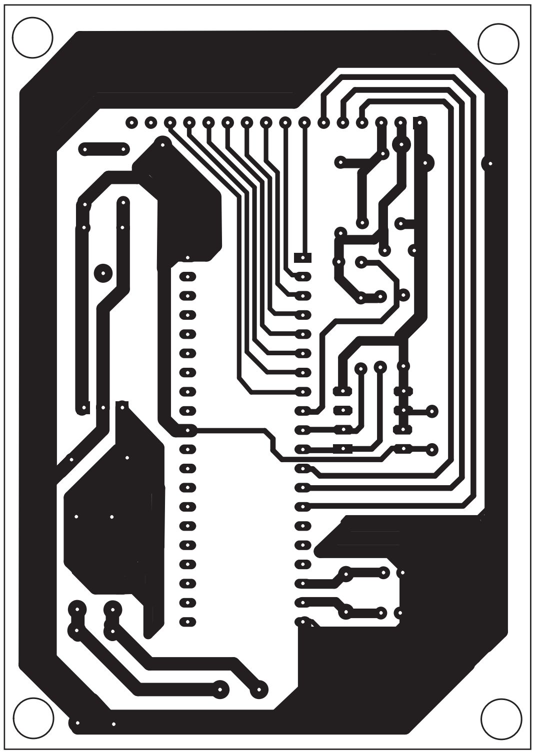


Fig. 6: Solder-side PCB layout for temperature indicator using AT8952

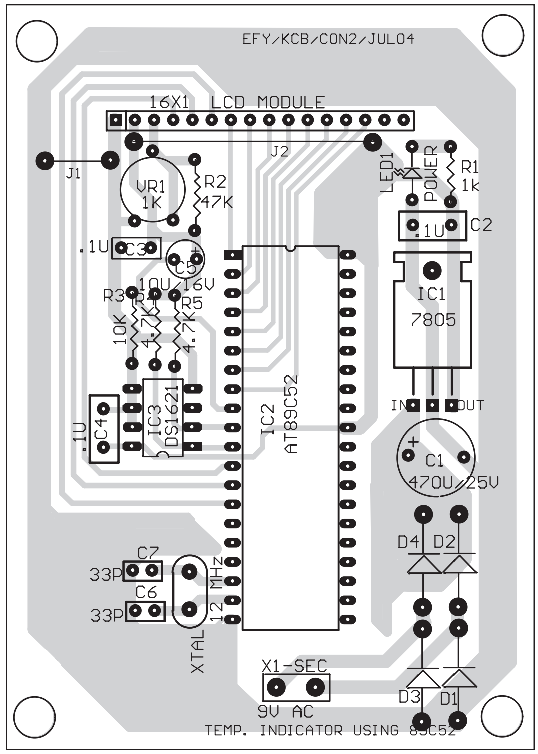


Fig. 7: Component layout for the PCB

the solder points on the PCB. Capacitors C3 and C4 must be connected near IC2 and IC3, respectively. The actual-size, single-side PCB layout for the circuit and its component layout are shown in Figs 6 and 7, respectively.

Program compilation

After you've installed Keil C51 in your PC, you can compile C program and generate hex file in either DOS or Windows mode. Here, program compilation for the program Hello.c has been explained. The same procedure is to be followed for the temperature indication program Temp52.c. For more example programs, refer to the directory in your hard drive where Keil is installed in the example folder.

DOS mode. 1. Installation of Keil C51 automatically generates 'Keil' folder in your computer's C drive.

2. Go to 'C:\Keil\C51\Bin' folder inside 'Keil' folder.

3. Copy 'Hello.c' into 'Bin' folder.

4. Copy 'Regx52.h' from 'C:\Keil\C51\Inc\Atmel' folder into 'C:\Keil\C51\Bin' folder.

5. Type 'c51 Hello.c' against the prompt and press Enter key.

6. Type 'bl51 Hello.obj.' This command is used for linking the Hello.obj file created by Keil C51.

7. Type 'oh51 Hello.' This command is used for creating the hex file.

Windows mode. 1. Installation of Keil C51 software automatically creates the icon 'Keil uVision2' on the desktop.

2. Double-click 'Keil uVision2.'

3. Suppose you have kept 'Hello.c' under 'C:\Windows \ Desktop\Hello' folder. Open 'Hello.c' from the 'File' menu.

4. From the menu bar, select 'Project/New Project.' Name the new project and save it with extension '.uv2.'

5. Select CPU as Atmel/AT89C52.

6. Choose 'Yes' in the option "Copy standard 8051 code to current project folder."

7. Choose 'View/Project Window.' A 'Project Workspace' window appears.

8. Double-click 'Target 1.'

9. Right-click 'Source Group1' and select "Add files to Group 'Source Group1.'" A window appears.

10. Add 'Hello.c' and close this window.

TABLE I
DS1621 Command Set

Instruction	Description	Protocol
Read Temperature	Reads last converted temperature value from temperature register.	Aah
Read Counter	Reads value of count remaining from counter.	A8h
Read Slope	Reads value of the slope accumulator.	A9h
Start Convert T	Initiates temperature conversion.	EEh
Stop Convert T	Halts temperature conversion.	22h
Access TH	Reads or writes high temperature limit value into TH register.	A1h
Access TL	Reads or writes low temperature limit value into TL register.	A2h
Access Configuration	Reads or writes configuration data to configuration register.	ACh

TABLE II
Pin Connections of the LCD

Pin No.	Functions
Pin 1	Ground (Gnd)
Pin 2	+Vcc
Pin 3	V0 (display intensity control)
Pin 4	RS (connected to P3.2 of AT89C52)
Pin 5	R/W (connected to P3.3 of AT89C52)
Pin 6	EN (connected to P3.4 of AT89C52)
Pin 7	D0 (connected to P1.0 of AT89C52)
Pin 8	D1 (connected to P1.1 of AT89C52)
Pin 9	D2 (connected to P1.2 of AT89C52)
Pin 10	D3 (connected to P1.3 of AT89C52)
Pin 11	D4 (connected to P1.4 of AT89C52)
Pin 12	D5 (connected to P1.5 of AT89C52)
Pin 13	D6 (connected to P1.6 of AT89C52)
Pin 14	D7 (connected to P1.7 of AT89C52)
Pin 15	Backlight +Vcc (not used)
Pin 16	Backlight Gnd (not used)

11. Double-click 'Source Group1' on the 'Project Workspace' window. Now the file name 'Hello.c' appears.

12. From 'Project' menu, select 'Options for File 'Hello.c.' In 'Properties,' choose file type as 'C source file.'

13. Again from 'Project' menu, select 'Options for Target 'Target1.'" A screen appears.

14. Choose 'Output' and tick on 'Hex File' for generating the hex file. Again choose 'Listing' option and tick on 'Conditional and Assembly Code'.

15. Open the Project menu and select 'Build Target' or press F7. The compiler shows ""Hello" 0 Error(s), 0 Warning(s)" in the output window just below the project window.

16. Close the screen and go to the 'Hello' folder to see the generated hex file and listing file.

Load the hex file into the microcontroller chip using a programmer. (Here' we've used Atmel Flash Programmer from Frontline Electronics.) Now integrate the microcontroller chip into the populated PCB comprising the temperature sensor and the LCD module.

Troubleshooting

1. Check the COM port on your PC before programming.

2. In case there is no message even if all the connections are correct, adjust the intensity control potentiometer (VR1) for display.

3. Check whether your hex file matches with the hex file given below in the article.

4. If the LCD shows wrong characters, replace it with another make LCD.

5. If DS1621 is not connected properly to AT89C52, the display will be completely blank.

Note: All the source codes and relevant files of this article have been included in CD.

TEMP52.C

```
/* Written By: Aditya Rane
T.E Computer Engg, Lokmanya Tilak College of Engineering, New Bombay, Vashi
E-mail: aditya@orionengg.com
Program for temperature indicator compiled under keil 'C' */

#include<stdio.h>
#include<string.h>
#include<Regx52.h>

//------------------------------------------------------------------------
//Global Variable
//------------------------------------------------------------------------
int temperature;
#define     HIGH 0x01   // Active High Signal
#define     LOW 0x00    // Active Low Signal
#define     TRUE 0x01   // Active High State
#define     FALSE 0x00  // Active Low State

//------------------------------------------------------------------------
// Functions Prototyping
//------------------------------------------------------------------------
void ready (void);
void command (int);
void display (char *);
void i2c_stop (void);
void i2c_start (void);
void i2c_write (unsigned char);
unsigned char i2c_read (void);
void convert (unsigned char);

//------------------------------------------------------------------------
// Port Defination
//------------------------------------------------------------------------
#define     DATA P3_1           // Serial data
#define     CLOCK P3_0          // Serial clock

//Begining of Main Program
void main (void)
{
int tmp;
char str[16];
bit flag = FALSE;
unsigned char ch;
void command (int);
void display (char *);
command(0x3c);
command(0x0c);
command(0x06);
while(1)
{
        i2c_start();
        i2c_write(0x90);
        i2c_write(0xEE);
        i2c_stop();

        i2c_start();
        i2c_write(0x90);
        i2c_write(0xAA);
        i2c_start();
        i2c_write(0x91);
        ch = i2c_read();
        i2c_stop();
        temperature = 0;
        convert(ch);
        if(flag == FALSE)
        {
                flag = TRUE;

                tmp = temperature;
        }
        else
        {
                if(tmp != temperature)
                {
                tmp = temperature;
                        sprintf(str,"%d%
s",temperature," Centigrade");
                        command(0x01);
                        command(0x80);
                        display(str);
                }
        }
}
}

//Delay Servive Routine
void delay_time (void)
{
unsigned int i;
for(i=0;i<100;i++);
}

//I2C Start Function
void i2c_start (void)
{
DATA = HIGH;
delay_time();
CLOCK = HIGH;
delay_time();
DATA = LOW;
CLOCK = LOW;
}

//I2C Stop Function
void i2c_stop (void)
{
unsigned char i;
CLOCK = LOW;
DATA = LOW;
CLOCK = HIGH;
delay_time();
DATA = HIGH;
i = DATA;
}

//I2C Data Write Function
void i2c_write (unsigned char j)
{
unsigned char i;
for(i=0;i<8;i++)
{
        DATA = ((j & 0x80) ? 1 : 0);
        j <<= 1;
        CLOCK = HIGH;
        delay_time();
        CLOCK = LOW;
}
i = DATA;
CLOCK = HIGH;
delay_time();
CLOCK = LOW;

}

//I2C Data Read Function
unsigned char i2c_read (void)
{
unsigned char i,j;
j = 0;
i = DATA;
for(i=0;i<8;i++)
{
        j <<= 1;
        CLOCK = HIGH;
        j |= DATA;
        delay_time();
        CLOCK = LOW;

}
return j;
}

//Binary to Decimal Conversion Function
void convert (unsigned char ch)
{
char x;
unsigned char arr[8]={128,64,32,16,8,4,2,1};
if(((ch & 0x80) ? 1 : 0)==0)
{
        for(x=0;x<8;++x)
        {
                if(((ch & 0x80) ? 1 : 0))
                temperature = temperature +
arr[x] * ((ch & 0x80) ? 1 : 0);
                ch <<= 1;
        }
}
else
{
        ch=~ch;
        ch=ch+1;
        for(x=0;x<8;++x)
        {
                if(((ch & 0x80) ? 1 : 0))
                temperature = temperature +
arr[x] * ((ch & 0x80) ? 1 : 0);
                ch <<= 1;
        }
        temperature=-temperature;
}
}

//Display Ready Check Function
void ready (void)
{
P3_4 = 0x00;
P1 = 0xff;
P3_2 = 0x00;
P3_3 = 0x01;
```

```
while(P1_7)
{
		P3_4 = 0x00;
		P3_4 = 0x01;
}
P3_4 = 0x00;
}

//Display Command Function void command (int a)
{
ready();
P1 = a;
P3_2 = 0x00;
P3_3 = 0x00;
P3_4 = 0x01;
P3_4 = 0x00;
}

//Display Write Function
void display (char *str)
{
unsigned int i;
for(i=0;i<=strlen(str)-1;++i)
{
		if(i == 8)
		command(0xc0);
		if(i == 16)
		command(0x80);
		ready();
		P1 = str[i];
		P3_2 = 0x01;
		P3_3 = 0x00;
		P3_4 = 0x01;
		P3_4 = 0x00;
}
}
```

TEMP52.HEX

```
:100F270025642573002043656E7469677261646583
:090F3700008040201008040201B2
:100DA800C2007F3C7E00120F9A7F0C7E00120F9
AC1
:100DB8007F067E00120F9A120F8B7F90120EB87F5B
:100DC800EE120EB8120F6A120F8B7F90120EB87
FB8
:100DD800AA120EB8120F8B7F91120EB8120F098
F3C
:100DE80034120F6AE4F508F509AF34120CF220004A
:100DF8000AD20085082285092380BCE5236509708D
:100E080004E522650860B08508228509237538FF46
:100E180075390F753A2785083B85093C753DFF757F
:100E28003E0F753F2C7B007A00792412085C7F0105
:100E38007E00120F9A7F807E00120F9A7B007A0044
:080E48007924120E50020DBFC7
:0E0F7C00E4FFFE0FBF00010EEF64644E70F53F
:010F8A002244
:0F0F8B00D2B1120F7CD2B0120F7CC2B1C2B02211
:100F6A00C2B0C2B1D2B0120F7CD2B1A2B1E433
F591
:010F7A003541
:010F7B002253
:020EB800AD0784
:100EBA00E4FCED30E703D38001C392B1ED25E0FDF8
:100ECA00D2B0120F7CC2B00CBC08E7A2B1E433FC6A
:070EDA00D2B0120F7CC2B080
:010EE10022EE
:100F0900E4FDA2B1E4FCED25E0FDD2B0A2B1E433E9
:0D0F19004205120F7CC2B00CBC08EBAF0506
:010F260022A8
:020CF2008F353C
:100CF40078377C007D007BFF7A0F79387E007F088F
:100D0400120C2CE53530E7047F0180027F00EF7080
:100D140041F536E53530E7047F0180027F00EF605E
:100D24002374372536F8E6FD7C00E5357E0030E790
:100D3400047F0180027F00120C7FEF2509F509EE84
:100D44003508F508E53525E0F5350536E536B4080A
:100D5400C2226335FF0535E4F536E53530E7047F17
:100D64000180027F00EF602374372536F8E6FD7CAE
:100D740000E5357E0030E7047F0180027F00120C1D
:100D84007FEF2509F509EE3508F508E53525E0F589
:100D9400350536E536B408C2C3E49509F509E4958A
:030DA40008F50847
:010DA7002229
:100F4000C2B47590FFC2B2D2B3309706C2B4D2B465
:050F500080F7C2B4228D
:0E0F9A00120F408F90C2B2C2B3D2B4C2B422C2
:060E50008B358A3689375C
:100E5600E4F538F539AB35AA36A937120F55EF2424
:100E6600FFFFEE34FFFED3E5399FE5389E5042E59D
:100E7600396408453870067FC0FE120F9AE539645A
:100E860010453870067F80FE120F9A120F40AB3560
:100E9600AA36A937853982853883120C52F590D245
:100EA600B2C2B3D2B4C2B40539E53970A8053880E8
:010EB600A497
:010EB7002218
:03000000020FA844
:0C0FA800787FE4F6D8FD758148020DA8A2
:100B5C00E709F608DFFA8046E709F208DFFA803
E7B
:100B6C0088828C83E709F0A3DFFA8032E309F60868
:100B7C00DFFA8078E309F208DFFA807088828C83D0
:100B8C00E309F0A3DFFA806489828A83E0A3F60884
:100B9C00DFFA805889828A83E0A3F208DFFA804C5E
:100BAC0080D280FA80C680D4806980F28033801035
:100BBC0080A680EA809A80A880DA80E280CA80339E
:100BCC0089828A83ECFAE493A3C8C582C8CCC58316
:100BDC00CCF0A3C8C582C8CCC583CCDFE9DEE780E6
:100BEC000D89828A83E493A3F608DFF9ECFAA9F065
:100BFC00EDFB2289828A83ECFAE0A3C8C582C8CCBB
:100C0C00C583CCF0A3C8C582C8CCC583CCDFEADED3
:100C1C00E880DB89828A83E493A3F208DFF980CC35
:100C2C0088F0EF60010E4E60C388F0ED2402B4042E
:100C3C000050B9F582EB2402B4040050AF232345D5
:060C4C008223900BAC7343
:100C5200BB010CE58229F582E5833AF583E0225057
:100C620006E92582F8E622BBFE06E92582F8E222A1
:0D0C7200E58229F582E5833AF583E49322BB
:100C7F00EF8DF0A4A8F0CF8CF0A428CE8DF0A42E89
:020C8F00FE2243
:10080000E5442438F8E60544227835300802783883
:10081000E475F001120CBC020C912001EB7F2ED28A
:10082000018018EF540F2490D43440D4FF30050BCE
:10083000EF24BFB41A0050032461FFE545600215A0
:10084000450548E5487002054730080D7835E475E0
:10085000F001120CBCEF020CAA020EE27403D208E3
:100860008003E4C208F5448B358A368937E4F545C0
:10087000F547F548E54560077F2012083B80F57590
:1008800046FFC202C201C203C204C206C207C209B5
:10089000120809FF700D3008057F0012084CAF48A0
:1008A000AE4722B4255FC2D5C205120809FF24D085
:1008B000B40A00501A75F00A784530D50508B6FF1D
:1008C0000106C6A426F620D5047002D20480D924DD
:1008D000CFB41A00EF5004C2E5D205020A4CD2028E
:1008E00080C6D20180C0D20380BCD2D580BAD206E5
:1008F00080B47F2012083B2003077401B5450040F7
:10090000F1120800FF12083B020874D209D20780D6
:1009100095120800FB120800FA120800F94A4B7001
:1009200006791D7A0B7BFF20032EE545602A7E00A9
:100930008E82758300120C5260060EEE654670F0D2
:10094000C2D5EBC0E0EAC0E0E9C0E0EE120A93D005
:10095000E0F9D0E0FAD0E0FB120C91FF60AAEBC006
:10096000E0EAC0E0E9C0E012083BD0E02401F9D0A1
:10097000E03400FAD0E0FBE5460460DCD546D980DF
:10098000877BFF7A0A798FD203809C791080027965
:1009900008C207C2098008D2D5790A8004790AC240
:1009A000D5E546047002F546E4FAFDFEFF120800A4
:1009B000FC7B08200213120800FD7B1030010A1294
:1009C0000800FE120800FF7B20EC3382D592D55040
:1009D00013C3E43001069FFFE49EFEE42002039D62
:1009E000FDE49CFCE4CBF8C202EC700CCFCECDCC85
:1009F000E824F8F870F38017C3EF33FFEE33FEED11
:100A000033FDEC33FCEB33FB994002FB0FD8E9EBF1
:100A1000300205F8D0E0C448B202C0E00AEC4D4E06
:100A20004F78207B0070C2EAB5460040BCC0E0129F
:100A30000A95D0F0D0E0200204C4C0E0C4B202C0E5
:100A4000F0120824D0F0D5F0EB020874120CCC0997
:100A50001153098B5808E24C08DE42098F4F099761
:0F0A60004409974908F743099D550981460981C3
:100A6F00450981470B3D5008E62D08EA2E090D2B4D
:100A7F0008EE23090B200B262A08A64800000905BB
:100A8F003F3F3F00790AA2D5200414300609B91060
:100A9F00020404B9080104A2D52007025001042062
:100AAF0003689203B545005034C0E07F203004192D
:100ABF007F30A20372077206500F120AECC203C2F4
:100ACF0007C206C2097F30800F300603E9C0E0126B
:100ADF00083B300603D0E0F9D0E0B545CC3006171F
:100AEF007F30B9100C12083B7F583005077F788094
:100AFF0003B9080312083B3003057F2D02083B7F23
:100B0F00202009F87F2B2007F322920380CF286E35
:100B1F00756C6C2900D2021208003002F8C20278FC
:100B2F004530D50108F60208A62D50434958120842
:100B3F00002403B405004001E4900B389312082CF5
:0D0B4F00743A12082CD20475450402098B7B
:100F5500E4FFFE120C91600C0FEF70010E09E970B1
:050F6500F20A80EF22FA
:100C9100BB010689828A83E0225002E722BBFE0261
:090CA100E32289828A83E4932294
:100CAA00BB010689828A83F0225002F722BBFE0129
:020CBA00F32223
:100CBC00FAE6FB0808E6F925F0F618E6CA3AF62239
:100CCC00D083D082F8E4937012740193700DA3A3B7
:100CDC0093F8740193F5828883E4737402936860CB
:060CEC00EFA3A3A380DFCB
:100EE200EFB40A07740D120EED740A309811A89926
:100EF200B8130CC2983098FDA899C298B811F63070
:070F020099FDC299F5992247
:00000001FF
```

HELLO.C

```
#include<stdio.h>
#include<string.h>
#include<Regx52.h>

void ready(void);
void command(int);
void display(char *);
void main (void)
{
command(0x3c);
command(0x0c);
command(0x06);
command(0x01);
command(0x80);
display("Hello! How R U ?");
while(1);
}

void command(int a)
{
void ready(void);
ready();
P1=a;
P3_2=0x00;
P3_3=0x00;
P3_4=0x01;
P3_4=0x00;
}

void display(char *str)
{
unsigned int i;
for(i=0;i<=strlen(str)-1;++i)
{
		if(i == 8)
		command(0xc0);
		if(i == 16)
		command(0x80);
		ready();
		P1 = str[i];
		P3_2 = 0x01;
		P3_3 = 0x00;
		P3_4 = 0x01;
		P3_4 = 0x00;
}
}

void ready(void)
{
P3_4=0x00;
P1=0xff;
P3_2=0x00;
P3_3=0x01;
while(P1_7)
{
		P3_4=0x00;
		P3_4=0x01;
}
P3_4=0x00;
}
```

HELLO.HEX


```
:0300000002092AC8
:0C092A00787FE4F6D8FD75810E0208AE5F
:1009190048656C6C6F2120486F7720522055203F25
:0109290000CD
:1008AE007F3C7E001209067F0C7E001209067F0631
:1008BE007E001209067F017E001209067F807E00EF
:0E08CE001209067BFF7A09791912080080FED4
:100906008E0D8F0E1208DC850E90C2B2C2B3D2B421
:03091600C2B42246
:060800008B088A09890A39
:10080600E4F50BF50CAB08AA09A90A1208F1EF24C6
:10081600FFFFEE34FFFED3E50C9FE50B9E5042E54D
:100826000C6408450B70067FC0FE120906E50C64D1
:1008360010450B70067F80FE1209061208DCAB0815
:10084600AA09A90A850C82850B83120868F590D23D
:10085600B2C2B3D2B4C2B4050CE50C70A8050B80C5
:01086600A4ED
:01086700226E
:1008DC00C2B47590FFC2B2D2B3309706C2B4D2B4D0
:0508EC0080F7C2B422F8
:10086800BB010CE58229F582E5833AF583E0225045
:1008780006E92582F8E622BBFE06E92582F8E2228F
:0D088800E58229F582E5833AF583E49322A9
:1008F100E4FFFE120895600C0FEF70010E09E9701C
:05090100F20A80EF2264
:10089500BB010689828A83E0225002E722BBFE0261
:0908A500E32289828A83E4932294
:00000001FF
```

❑

Readers' comments

I have purchased the complete kit, While assembling it, I found that the J2 label shown on the PCB is missing in the kit. As such, the LCD module could not be attached to the PCB. Also, J1 label shown on the PCB having two holes has neither been shown in the circuit diagram nor it was found in the kit. Please clarify.

Arun Rana
Meerut

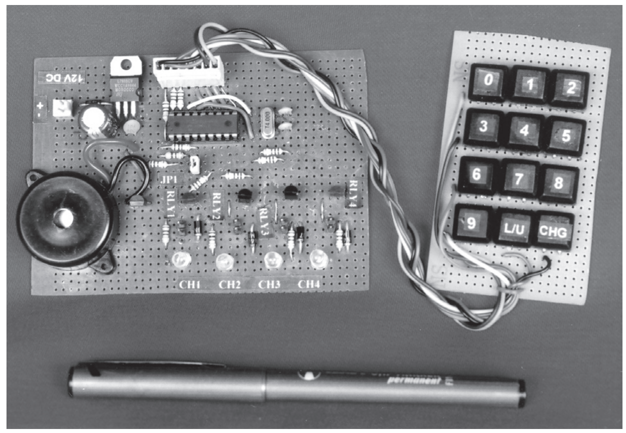
EFY: J1 and J2 are nothing but jumper connectors. You can use any conductor wire to connect them. The respective holes for connecting these jumpers are provided in the PCB.

PIC16F84-BASED CODED DEVICE SWITCHING SYSTEM

VIJAYA KUMAR P.

Here's a microcontroller-based code lock that can be used for preventing unauthorised access to devices or solenoid-operated locks/electrical devices. This code lock is built around Microchip's PIC16F84 microcontroller. Different passwords are used to access/operate different devices. So the code lock can be used as a multiuser code lock, where the users can access respective devices by entering the device number followed by the password. The password can be changed by the user and no external back-up supply is needed to retain the password. The password length for each device can be between 4 and 15 digits, as desired by the user.

A buzzer has been added to provide suitable feedback with respect to the data entered via the keypad. The number of beeps indicates whether the data has been entered correctly or not. When anyone trying to access the device enters the incorrect password three times, the circuit sounds an alarm.



Working model of PIC16F84-based coded device switching system

The alarm can be configured to work in two modes: auto-reset and latch-up. In the auto-reset alarm mode, all the keys pressed are ignored and the buzzer keeps beeping continuously for one minute, and thereafter the code lock resets automatically.

However, if you want additional security, you can enable the latch-up mode. In this mode the code lock never switches to the normal mode from the alarm mode and the only way to reset the code lock is to interrupt the power. When not in use, the code lock goes into sleep mode, and it wakes up if any key is pressed. This feature reduces the power consumption by the microcontroller.

The main features of PIC16F84 microcontroller are:

1. Program and data memory are in separate blocks, with each having its own bus connecting to the CPU
2. Reduced instruction set controller (RISC) with only 35 instructions to learn
3. 1024 words (14-bit wide) of program memory
4. 68 bytes of data RAM
5. 64 bytes of data EEPROM
6. 8-bit wide data bus
7. 15 special-function registers (SFRs)
8. 13 input/output (I/O) pins with individual direction control
9. Code protection
10. Built-in power-on-reset, power-up timer, oscillator start-up timer
11. Power-saving sleep mode

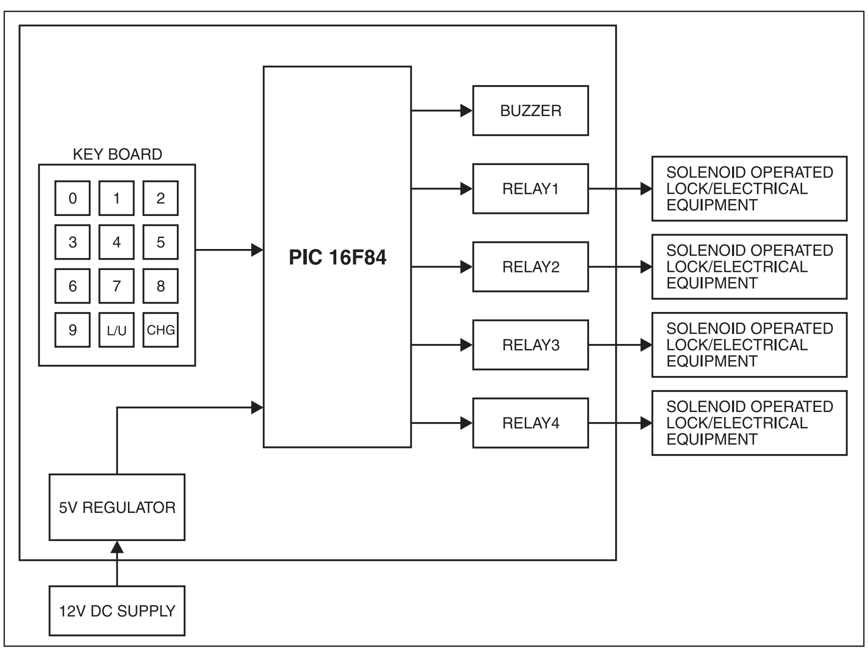


Fig. 1: Block diagram of PIC16F84-based coded device switching system

Circuit description

Fig. 1 shows the block diagram of the microcontroller-based code lock. Pin diagram of PIC16F84 microcontroller is shown in Fig. 2. Basically, the circuit

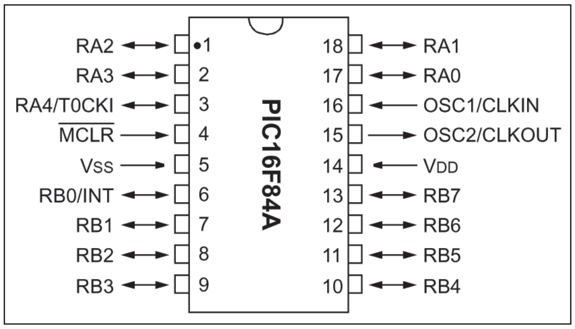


Fig. 2: Pin details of PIC18F84 microcontroller

PARTS LIST	
Semiconductors:	
IC1	- 7805 +5V regulator
IC2	- PIC16F84 microcontroller
T1-T5	- BC547 npn transistor
D1-D5	- 1N4007 rectifier diode
LED1-LED4	- Red LED
Resistors (all ¼-watt, ±5% carbon, unless stated otherwise):	
R1	- 10-kilo-ohm
R2	- 4.7-kilo-ohm
R3-R5	- 220-ohm
R6-R10	- 2.2-kilo-ohm
R11-R14	- 1-kilo-ohm
Capacitors:	
C1	- 470µF, 35V electrolytic
C2, C3	- 0.1µF ceramic disk
C4, C5	- 33pF ceramic disk
Miscellaneous:	
RL1- RL4	- 12V, 285-ohm, 1C/O relay (OEN58 type 1C)
X_{TAL}	- 4MHz crystal
PZ1	- Piezobuzzer
S1-S12	- Push-to-on tactile switch

(shown in Fig. 3) comprises PIC16F84 microcontroller (IC2), 4x3 matrix keyboard, relays and buzzer.

The microcontroller. PIC16F84 is an 8-bit CMOS microcontoller. Its internal circuitry reduces the need for external components, thus reducing the cost and power consumption and enhancing the system reliability. The microcontroller has two ports, namely, Port A and Port B. Out of the available 13 bidirectional I/O pins of Ports A and B, seven pins are used for keyboard interfacing, four pins are used to drive the relays corresponding to the four devices and one pin is used to read the jumper status for selecting the alarm mode. One can reset the microcontroller only by interrupting the power.

The password is stored in the internal 64-byte EEPROM memory of the microcontroller at addresses 0x00 through 0x3F. The memory can be programmed and read by both the device programmer and the CPU when the device is not code

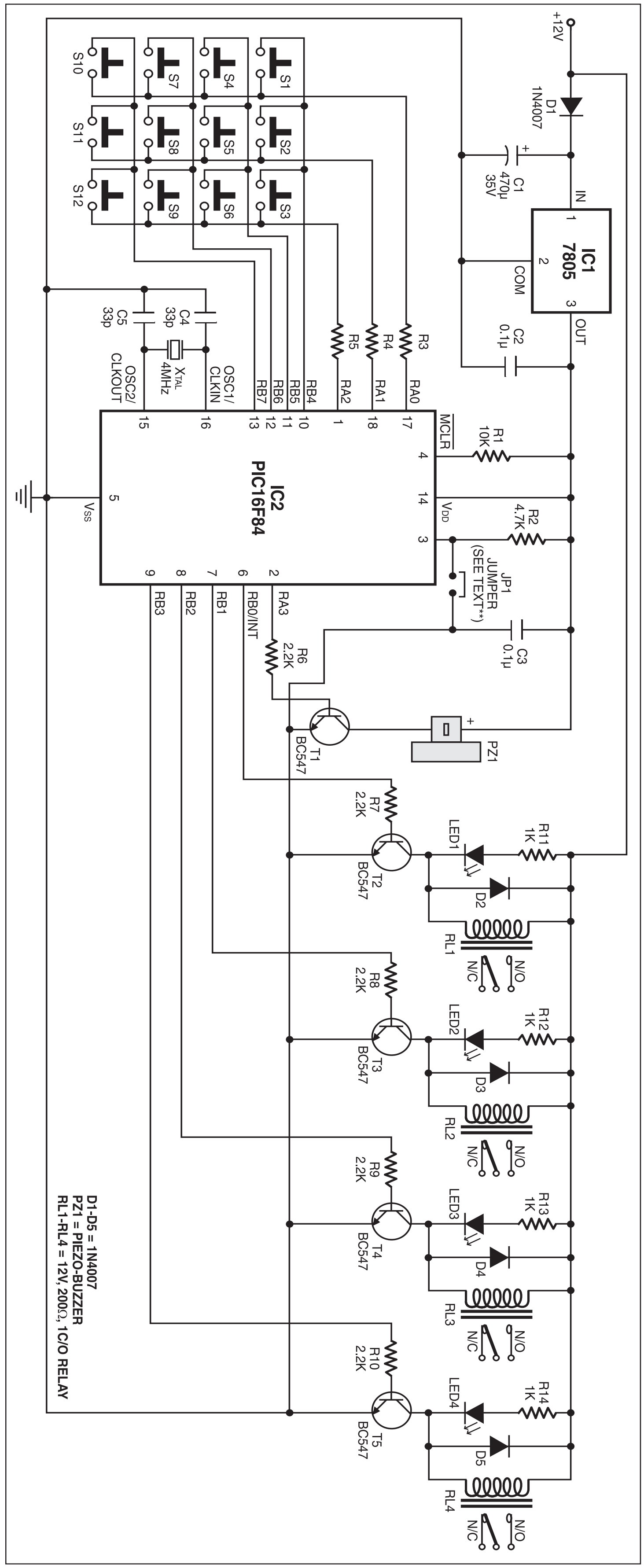


Fig. 3: Circuit diagram of PIC16F84-based coded device switching system

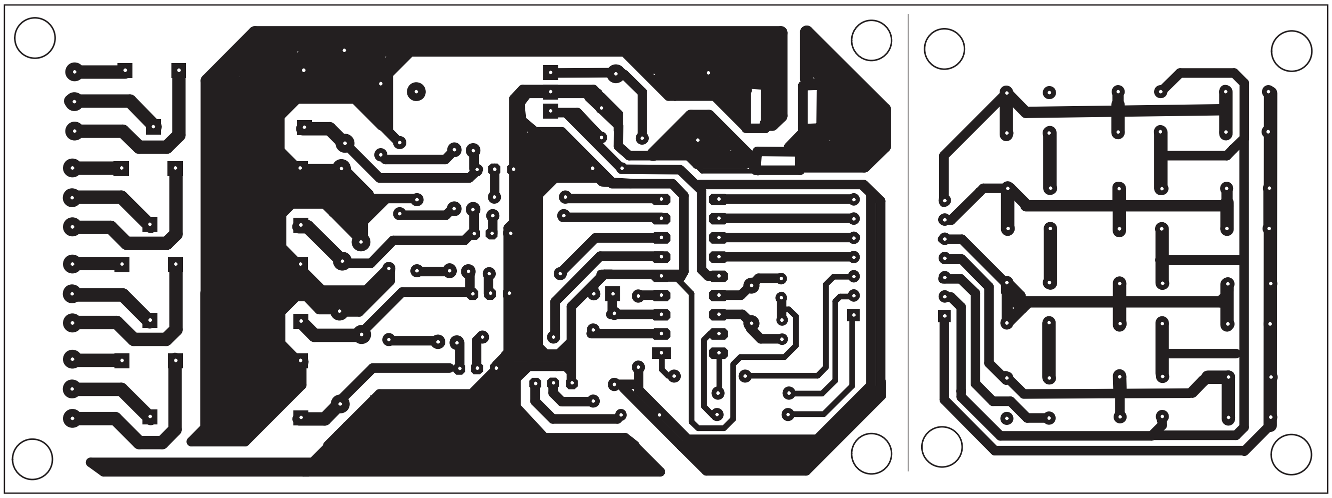


Fig. 4: Actual-size, single-side PCB layout for PIC16F84-based coded device switching system

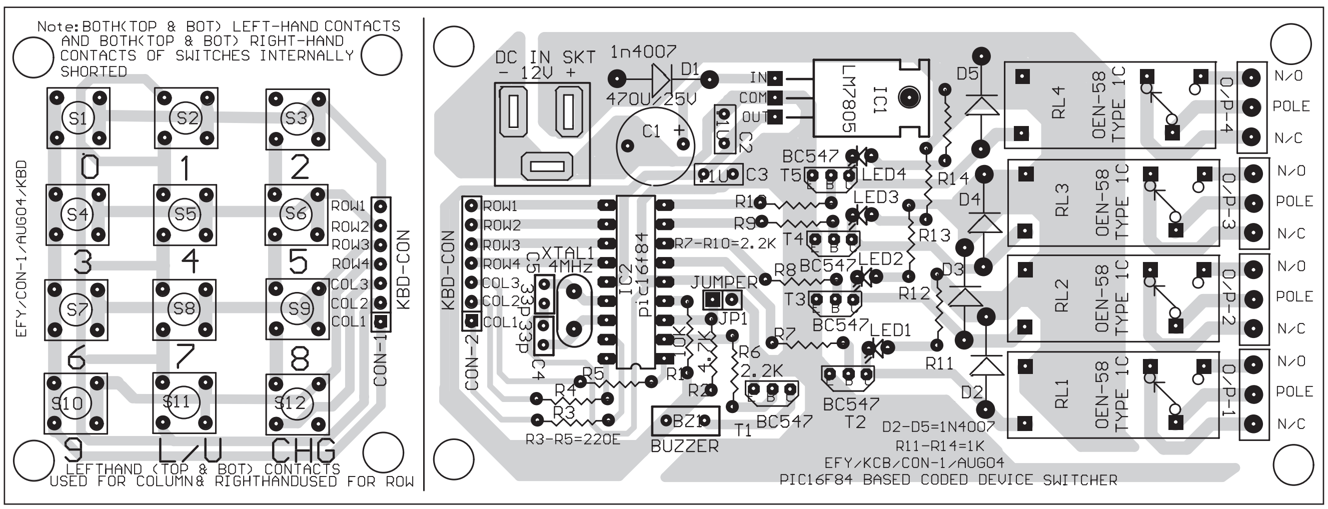


Fig. 5: Component layout for the PCB

protected. It is non-volatile and can retain data for more than 40 years.

Four special-function registers are used to read and write the EEPROM. These registers are named as EECON1, EECON2, EEDATA and EEADR, respectively. Register EEDATA holds 8-bit data for read/write and register EEADR holds the address of the EEPROM location being accessed. Register EECON1 contains the control bits, while register EECON2 is used to initiate the read/write operation.

Oscillator. The internal oscillator circuitry of the microcontroller generates the device clock. The microcontroller can be configured to work in one of the four oscillator modes:

1. External resistor-capacitor
2. Low-power crystal (oscillation frequency up to 200 kHz)
3. Crystal/resonator (oscillation frequency up to 4 MHz)
4. High-speed crystal/resonator (oscillation frequency up to 10 MHz)

In this circuit, the oscillator is configured to operate in crystal mode with a 4MHz crystal along with two 33pF capacitors.

Reset circuit. The built-in power-on reset circuitry of the microcontroller eliminates the need for the external power-on reset circuit. In the circuit, MCLR pin is tied to V_{DD} through resistor R1 (10 kilo-ohms) to enable power-on reset. The internal power-up timer (PWRT) provides a nominal 72ms delay from power-on reset. This delay allows V_{DD} to rise to an acceptable level when the microcontroller is powered on. The oscillator start-up timer (OST) provides 1024-oscillator cycle delay after the power-up timer delay is over. This ensures that the crystal oscillator has started and is stable.

Power supply. The 12V DC supply for the circuit is obtained from a 12V adaptor with 500mA rating. Any other source such as a 12V lead-acid battery can also be used. This 12V DC is used for operation of the relays used in the circuit. The regulated +5V supply for the microcontroller is derived using regulator IC 7805 (IC1). Diode D1 protects the circuit from reverse supply connections. Capacitor C1 filters out the ripples present in the incoming DC voltage.

Keyboard. The 12-key matrix keyboard comprises 12 tactile pushbutton switches arranged in four rows and three columns as shown in Fig. 3. Data is entered via this keyboard.

Ports A and B of the microcontroller are bidirectional I/O ports. Three lines of Port A (RA0 through RA2) are used as the output-scan lines and four lines of Port B (RB4 through RB7) are used as the input-sense lines. Port B of IC2 has weak

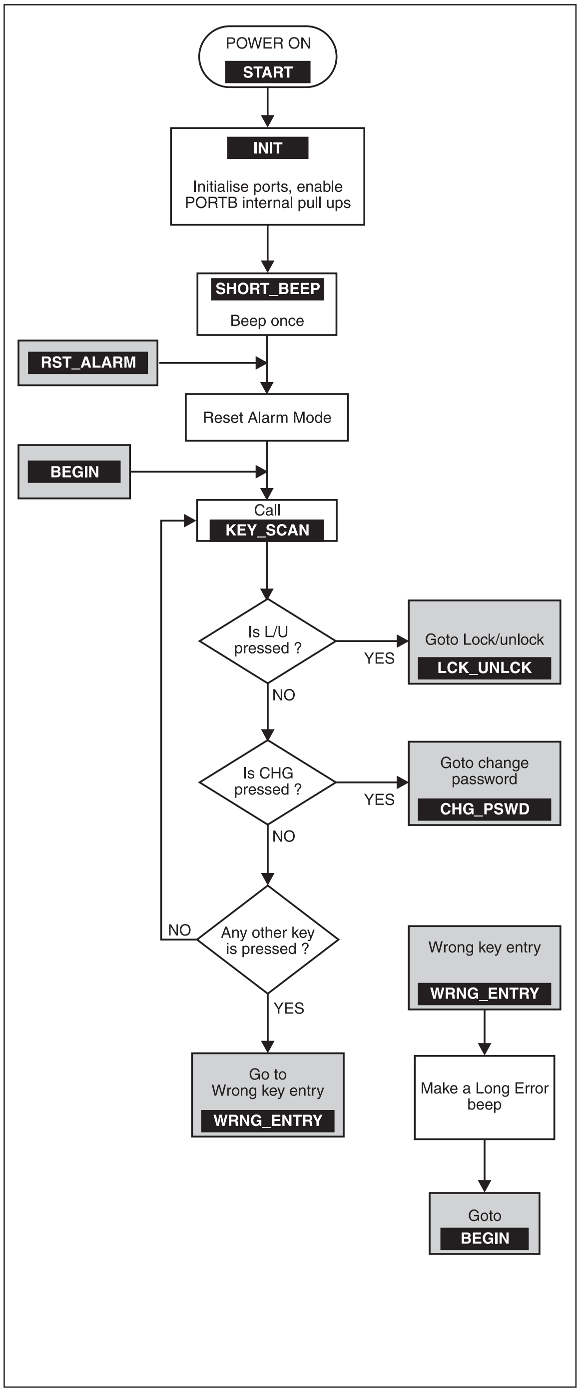


Fig. 6: Flow-chart of the main program

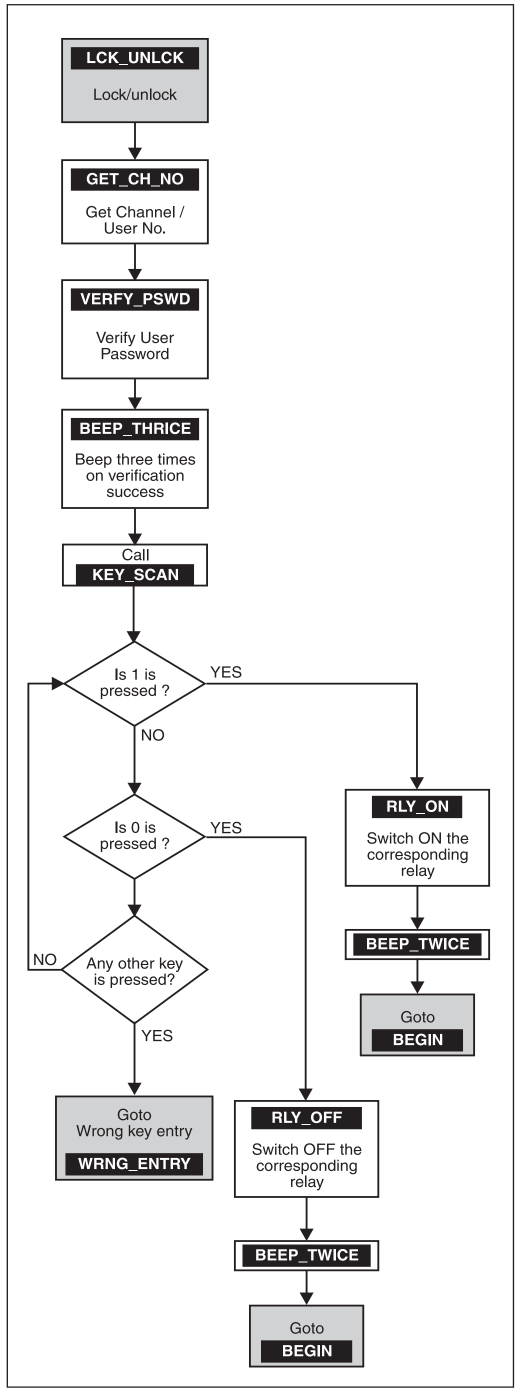


Fig. 6(a): Flow-chart for locking/unlocking the code lock

internal pull-ups, which can be enabled through the software. This eliminates the need for connecting external pull-up resistors to pins 10 through 13. Resistors R2 through R4 protect Port A's output drivers from shorting together when two keys of the same row are inadvertantly pressed simultaneously.

In the scanning routine, initially all the scan lines are made low and it is checked whether all the keys are in released state. If all the keys are in released state, the processor is put into sleep (power-down) mode. The interrupt-on-change feature of Port-B pins RB4 through RB7 is used to wake up the processor from sleep.

When any key is pressed, one of the sense lines becomes low. This change in the pin status causes an interrupt to wake up the microcontroller (IC2) from sleep.

Now each scan line is made low while keeping the remaining scan lines in high state. After making a scan line low, the status of the sense lines is read. If any of the sense lines is found low, it means that a key at the intersection of the current scan line and the low sense line has been pressed. If no key is found to be pressed, the next scan line is made low and again scan lines are checked for low state. This way all the twelve keys are checked for any pressed key by the microcontroller.

Since mechanical tactile switch keys are used, pressing of a single key may be considered by the microcontroller as pressing of many keys due to the bouncing of the keys. To avoid this, the processor is made to wait up to a debounce delay of 20 ms during the pressing or releasing of a key. Within this debounce delay, all the bounces get settled out, thus debouncing the key.

In sleep (power-down) mode, the device oscillator is turned off and the microcontroller is placed in its lowest-current consumption state. Also note that the microcontroller's I/O pin status remains unaltered during sleep mode.

Relays. To turn on/off the equipment or to lock/unlock the solenoid-operated locks, four relays (RL1 through RL4) are provided—one for each channel. Since the current-driving capacity of the port pins of PIC16F84 (IC2) is not enough to drive the relays directly, transistors T2 through T5 are used to boost the current to drive relays RL1 through RL4, respectively.

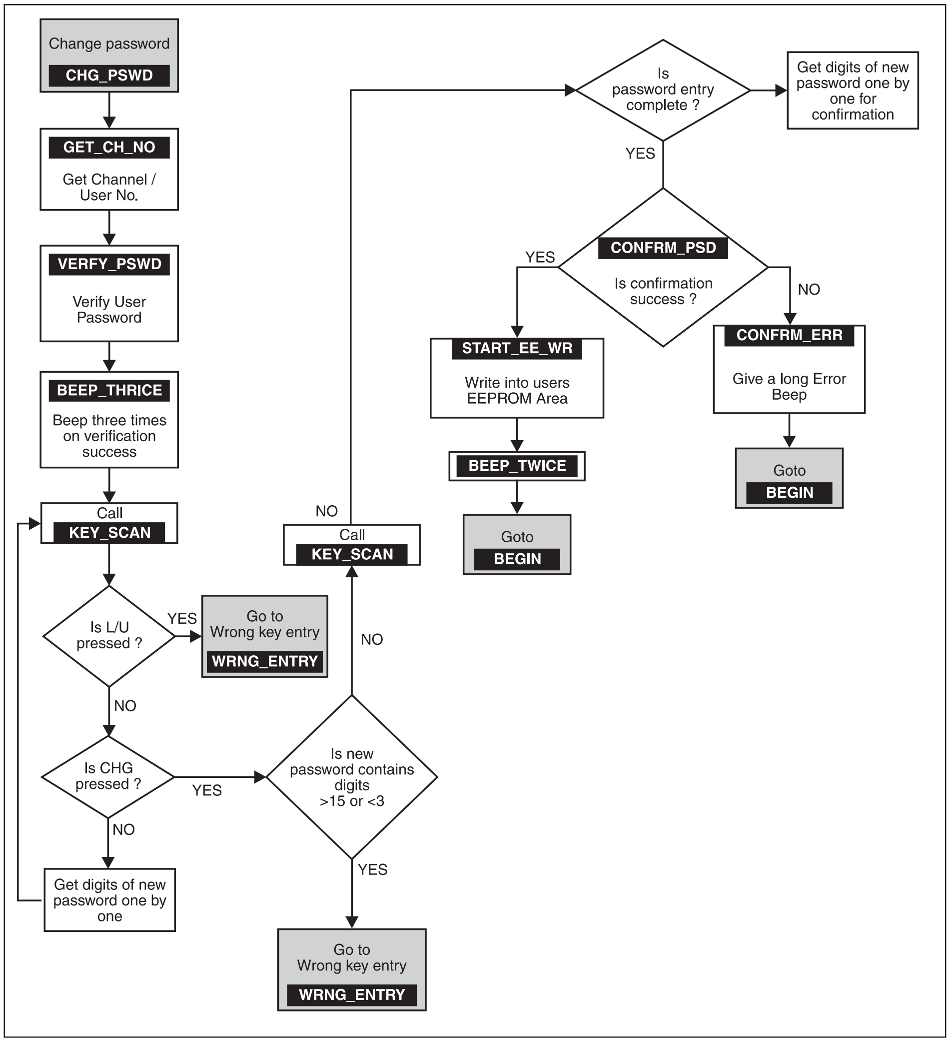


Fig. 6(b): Flow-chart for changing the password of the code lock

The bases of transistors T2 through T5 are connected to Port-B pins 6 through 9 (RB0 through RB3) through base-current-limiting resistors R7 through R10, respectively. The equipment or solenoid-operated locks can be connected to the normally open (N/O) contacts of these relays. Diodes D2 through D5 are used as freewheel clamp diodes. The series combination of a red LED (LED1 through LED4) and a current-limiting resistor (R11 through R14) is connected across each relay coil.

Buzzer. Pin 2 (RA3) of IC2 is connected via resistor R6 and transistor T1 to piezobuzzer PZ1. The buzzer gives a short beep when any key is pressed. In the case of a wrong data entry, the buzzer gives a long beep to indicate the error. On successful password verification, it gives three short beeps, and after successful password change, it gives two short beeps. When a wrong password is entered consecutively for three times, the buzzer sounds an alarm.

Construction and testing

An actual-size, single-side, PCB layout for PIC16F84-based coded device switching system is shown in Fig. 4 and its component layout in Fig. 5.

The main circuit and the matrix keyboard can be assembled on separate PCBs. First check the assembled PCBs for proper connections as per the circuit diagram. Then connect the main PCB to the matrix keyboard PCB using 7-pin SIP connectors and wires, ensuring one-to-one connection between the two PCBs. Connect the external 12V DC supply with the correct polarity, without inserting the PIC microcontroller into the socket, and follow these steps:

1. Check whether +5V is available at output pin 3 of regulator IC1 (7805).

2. Now check the availability of +5V at pins 4 and 14 of IC2 before placing IC2 into the socket.

3. To check the buzzer operation, connect pin 2 of IC2 socket to +5V available at pin 3 of IC1. Now the buzzer should beep continuously.

4. Check the operation of the four relays by connecting pins 6 through 9 of IC2 socket one by one to +5V.

5. Before placing jumper JP1, check the voltage at pin 3 of IC2 using a multimeter. The meter should read +5V or logic 1. Now on placing jumper JP1, the meter should read 0V or logic 0 at pin 3.

Now remove the supply and insert the programmed PIC16F84 microcontroller into the socket and switch on the supply. After power-on, the buzzer beeps once to indicate that the microcontroller is ready to take the user data. Now you can lock/unlock or change the password as described below. Initially the four channels can be accessed using the default password '1234.'

Operating procedure

For unlocking/switching on the equipment:

1. Press the lock/unlock button (L/U) on the keypad.

2. Now enter the device number by pressing the button corresponding to the device number. The valid device numbers are 1 to 4. For example, if you want to access device No. 1 (RL1), press button '1.'

3. Now enter your password digits one by one. Note that the default password is '1234.'

4. The buzzer gives three short beeps to indicate successful verification of the password. If the entered password is incorrect, the buzzer gives a long beep to indicate error. To try again, repeat the procedure from step 1.

5. If the entered password is correct, you can unlock or switch on device No. 1 by pressing button '1.' When you press the key, the relay corresponding to this device gets energised and it remains in this state until you lock/switch it off again.

For locking/switching off the equipment:

Follow the aforesaid steps 1 through 4 and press button '0.' Now the relay corresponding to the device you want to

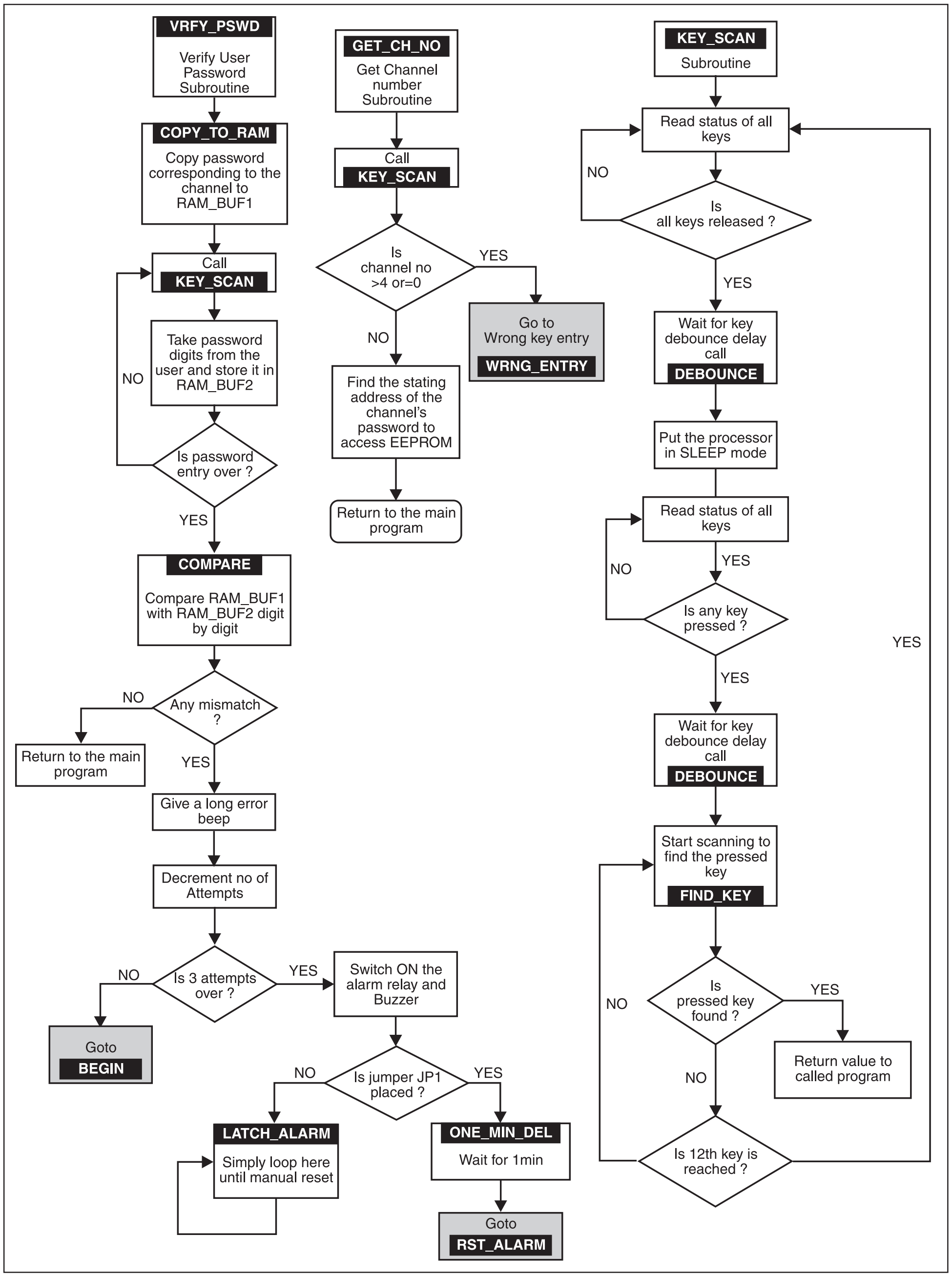


Fig. 6(c): Flow-chart for password verification, device (channel) selection and key scanning

turn off de-energises and it remains in this state until you unlock/switch it on again.

For changing the password:

1. Press the password change button (CHG) on the keypad.

2. Now press the device number.

3. Enter your current password.

4. On successful verification of the password, the buzzer gives three short beeps. If the entered password is wrong, the buzzer will give a long beep. Now if you want to try again, repeat the procedure from step 1.

5. Enter your new password. The length of the password should be between 4 and 15 digits.

6. End the password entry by pressing again CHG button.

7. Again enter your new password for confirmation. On successful confirmation, your new password gets replaced by the old password and the buzzer beeps twice to indicate successful password change. In case the password entered for confirmation is wrong, the buzzer gives a long beep to indicate error and the old password remains unaltered.

So whether you're locking, unlocking or changing the device, wrong password entry makes the buzzer to give a long error beep and the users are required to start afresh from step 1. In case you forget the password of the device, it can't be controlled until you reprogram the microcontro-ler.

Mode of operation. When anyone fails to enter the correct password in three attempts, the code lock circuit switches to alarm mode and the buzzer starts beeping continuously. All the keys pressed (for further attempts) are ignored by the code lock during alarm mode.

Placing the jumper between pin 3 (RA4) of IC2 and Ground enables the auto-reset alarm mode. Whereas removing the jumper enables the latch-up mode (see Fig. 3). If the auto-rest alarm mode is enabled, the code lock automatically resets after about one minute. If the latch-up alarm mode is enabled, the code lock never resets from the alarm mode until the user manually resets it by interrupting the power. Note that in the alarm mode the status of device-controlling relays remains unaltered.

Software

The software is written in Microchip's Assembly language. Fig. 6 shows the flow-chart for the program. In the flow-chart, important labels and subroutine names used in the program are also mentioned within the corresponding process boxes to enable easy understanding of the program. For instructions, you may refer to the PIC16F84 datasheet. The code is compiled and hex file is generated using MPLAB IDE. You can generate the hex file by using the MPASM.exe assembler also. The hex file generated can be burnt into the microcontroller using any PIC programmer that supports PIC16F84. We've used here PICburner to program the PIC. It is published in Electronics Project Vol-23.

CODLOCK.LST

```
MPASM 03.20 Released        CODLOCK.ASM   7-1-2004  16:25:54        PAGE  1

LOC  OBJECT CODE    LINE    SOURCE TEXT
 VALUE
                            00001 ;*************************************
                            00002 ;
                            00003 ;   TITLE: "MICROCONTROLLER BASED 4
                                        CHANNEL CODE LOCK"
                            00004 ;   PROCESSOR PIC16F84
                            00005 ;   Oscillator:XT 4MHz crystal Oscillator
                            00006 ;   Default passward:1234 for ch1 - ch4
                            00007 ;
                            00008 ;   Author:VIJAYA KUMAR.P
                            00009 ;   EMAIL:vijay_kum_p@yahoo.co.in
                            00010 ;
                            00011 ;*************************************
                            00012
                            00013 ;---------------------------------------------------------
                            00014
                            00015     #INCLUDE "p16f84.inc" ;Header file
                                      inclusion directive.
                            00001     LIST
                            00002 ;   P16F84.INC  Standard Header File,
                                      Version 2.00    Microchip Technology, Inc.
                            00136     LIST
                            00016
                            00017
                            00018 ;   NOTE: This header file consists of
                                      definations of all special function
                            00019 ;   registers (SFRs) and their associated bits.
                            00020
                            00021 ;---------------------------------------------------------
                            00022
                            00023 ;*********Configuration bit settings*******
                            00024
                            00025  LIST P=PIC16F84  ;processor type
                                               PIC16F84A
                            00026
2007  0001                  00027     __CONFIG  _XT_OSC  &_PWRTE_ON    &
                            _CP_ON & _WDT_OFF
                            00028
                            00029 ;   SETTING : XT oscillator mode,power up
                            timer ON, code protect on,watch dog
                            00030 ;   timer OFF
                            00031
                            00032 ;---------------------------------------------------------
                            00033 ;   Defining Default  passward. First time after
                            programming 16f84 you need
                            00034 ;   to use default passward 1234 for all
                            4 channels.
                            00035 ;---------------------------------------------------------
                            00036
2100                        00037     ORG  0X2100 ;Starting adderss of ch1's
                            passward
2100  0001 0002 0003        00038     DE 1,2,3,4 ;default passward for ch 1
       0004
210F  00039                 ORG 0X210F
210F  0004                  00040     DE D'04' ;Default passward length = 4 digits
                            00041
2110                        00042     ORG  0X2110 ;Starting adderss of ch2's
                                                          passward
2110  0001 0002 0003        00043     DE 1,2,3,4 ;Default passward for ch 2
      0004
211F                        00044     ORG 0X211F
211F  0004                  00045     DE D'04' ;Default passward length=4 digits
                            00046
2120                        00047     ORG 0X2120 ;Starting adderss of ch3's
                                                          passward
2120  0001 0002 0003        00048     DE 1,2,3,4 ;Default passward for ch 3
      0004
212F                        00049     ORG 0X212F
212F  0004                  00050     DE D'04' ;Default passward length=4 digits
                            00051
2130                        00052     ORG 0X2130 ;Starting adderss of ch4's
                                      passward
2130  0001 0002 0003        00053     DE 1,2,3,4 ;Default passward for ch 4
      0004
213F                        00054     ORG 0X213F
213F  0004                  00055     DE D'04' ;Default passward length=4 digits
                            00056
                            00057
                            00058 ;*************************************
                            00059     ;VARIABLE AND CONSTANT DATA
                                      DECLARATIONS
                            00060
                            00061
                            00062 ;   variables
                            00063
 0000000C                   00064     DEL_COUNT1 EQU  0X0C ;Counters used to
                                      obtain software delay.
 0000000D                   00065     DEL_COUNT2 EQU  0X0D
 0000000E                   00066     DEL_COUNT3 EQU  0X0E
 0000000F                   00067     KEY_IN   EQU  0X0F ;Holds the value of
                                      pressed key.
 00000010                   00068     KEY_NO   EQU  0X10 ;Holds key no.
 00000011                   00069     SCAN_CODE  EQU    0X11    ;Holds  scan
code.
 00000012                   00070     KB_TEMP  EQU  0X12  ;Temporary
                                      variable to hold key value
 00000013                   00071     RAM_BUF1_PNT EQU  0X13  ;Pointer reg
                                      to RAM_BUF1
 00000014                   00072     RAM_BUF2_PNT  EQU  0X14  ;Pointer reg
                                      to RAM_BUF2
 00000015                   00073     DIGIT_COUNT  EQU  0X15 ;Holds no of
                                      digits
 00000016                   00074     PSD_DIGIT EQU 0X16  ;Holds passward
                                      digit
 00000017                   00075     NO_OF_ATTEMPTS EQU 0X17 ;Holds no
                                      of attempts
 00000018                   00076     CH_NO  EQU 0X18 ;Holds channel/user no
 00000019                   00077     EEADDR_TEMP EQU 0X19 ;Temporary
                                      store to hold  EEPROM addr
 00000020                   00078     NO_OF_BEEPS EQU 0X20 ;Holds the
                                      number of beeps
 00000021                   00079     BUZ_DEL_CNT EQU 0X21 ;Counters used
                                      to obtain 1min delay
 00000022                   00080     TEN_SEC_CNT EQU 0X22
 00000023                   00081     ONE_MIN_CNT EQU 0X23
 00000024                   00082     NO_OF_DIGITS EQU 0X24 ;No of digits
                                      in a passward
                            00083
                            00084 ;   constant data declarations
                            00085
 00000030                   00086     RAM_BUF1  EQU 0X30 ;Starting address
                                      of RAM_BUF1
 00000040                   00087     RAM_BUF2  EQU 0X40  ;Starting address
                                      of RAM_BUF2
                            00088
                            00089
                            00090 ;*************************************
                            00091 ;   program starts from here as soon as you
                                      switch on the code lock circuit.
                            00092
 0000                       00093     ORG 0X0000  ;Reset vector
 0000                       2823            00094         GOTO START
                            00095
                            00096 ;*************************************
                            00097 ;   Interrupt service routine ISR for timer0
                                      starts from here.
                            00098 ;   This ISR is encountered for every 50ms.
                            00099 ;   NOTE:This ISR is used only to obtain 1
                                      minute delay.
                            00100
 0004                               00101         ORG 0X0004  ;Interrupt vector
 0004                       138B            00102         BCF  INTCON,GIE
;Dissable all interupts
 0005  1D0B                 00103     BTFSS INTCON,T0IF ;Is T0IF ==1?
 0006  0009                 00104     RETFIE   ;If No return form ISR
 0007  110B                 00105     BCF INTCON,T0IF ;If YES clear it
 0008  0BA2                 00106     DECFSZ TEN_SEC_CNT,F ;Decrement
                                      TEN_SEC_CNT and test if 0
 0009  280D                 00107     GOTO LOAD_TMR0 ;If !0 goto
                                      LOAD_TMR0,if 0,
 000A  0BA3                 00108     DECFSZ ONE_MIN_CNT,F ;Decrement
                                      ONE_MIN_CNT and test if 0
 000B  2810                 00109     GOTO  LOAD_TEN_SEC ;If !0 goto
                                      LOAD_TENS_SEC
 000C  2825                 00110     GOTO  RST_ALARM  ;If  0  goto  RST_
ALARM                       00111
 000D                       303F    00112  LOAD_TMR0 MOVLW 0X3F;Count for
                                      50ms
 000E                       0081            00113         MOVWF TMR0
 000F                       0009            00114         RETFIE
                                    00115
 0010   30C8      00116     LOAD_TEN_SEC MOVLW 0XC8  ;Count for 10sec
 0011   00A2                00117     MOVWF TEN_SEC_CNT
 0012   0009                00118     RETFIE
                            00119
                            00120 ;*************************************
                            00121 ;   INITIALISATION SUBROUTINE
                            00122
                            00123 ;   This part of the program intialises the
```

```
                required ports and SFRs.
0013  0183      INIT  CLRF STATUS  ;Switch to bank0
0014  0185      CLRF    PORTA  ;Clear PORTA
0015  0186      CLRF  PORTB  ;Clear PORTB
0016  1683      BSF  STATUS,RP0 ;Switch to bank1
0017  30F0      MOVLW  B'11110000'  ;Sets pins of portb
                as iiiioooo
0018  0086      MOVWF TRISB ;Where i=input &  o=output
0019  3010      MOVLW  B'00010000' ;Sets pins of porta
                as oooioooo
001A  0085      MOVWF  TRISA
001B  3007      MOVLW 0X07 ;Enable weak internal pull ups,
001C  0081      MOVWF  OPTION_REG ;asigns prescalar
                to TMR0 with
                ;     1:256 ratio.
001D  1283      BCF  STATUS,RP0  ;Switch to bank 0
001E  158B      BSF  INTCON,RBIE ;Enable portb int on
                change
001F  138B      BCF  INTCON,GIE ;Dissable all the
                interrupts
0020  3003      MOVLW  0X03  ;Max no of atempts = 3
0021  0097      MOVWF  NO_OF_ATTEMPTS
0022  0008      RETURN   ;Return from sub routine
               ;**************************************
               ;  The main program starts from here
0023  2013      START    CALL  INIT  ;Call initalization
                subroutine
0024  216C      CALL  SHORT_BEEP ;Now the buzzer
                beeps once
0025  1185      RST_ALARM BCF PORTA,3  ;Switch off
                buzzer
                ; here the program waits until L/U
                or CHG key is pressed.
0026  2033      BEGIN  CALL KEY_SCAN ;Call kb
                scanning routine
0027  0092      MOVWF KB_TEMP  ;W -->KB_TEMP
0028  3A0A      XORLW 0X0A   ;W XOR H'0A' -->W
0029  1903      BTFSC STATUS,Z ;Is L/U key is pressed ?
002A  2875      GOTO  LCK_UNLCK ;If yes goto
                LCK_UNLCK
002B  0812      MOVF  KB_TEMP,W ;KB_TEMP -->W
002C  3A0B      XORLW 0X0B   ;W XOR 0B -->W
002D  1903      BTFSC  STATUS,Z ;Else Is CHG key is
                pressed ?
002E  28FB      GOTO  CHG_PSWD ;If yes goto
                CHG_PSWD
002F  2831      GOTO    WRNG_ENTRY ;Give a long error
                beep on wrng key
0030  2826      GOTO  BEGIN ;Else simply LOOP_HERE
               ;**************************************
               ;  the program control comes here when any
                  wrong data entry is made.
0031  2172      WRNG_ENTRY CALL  LONG_BEEP
0032  2826      GOTO BEGIN
               ;**************************************
               ;  KEYBOARD SCANING ROUTINE
               ;
               ;  This subroutine when called returns the
                  value of key pressed in
               ;  w register and makes the buzzer to beep
                  once for every key press.
               ;  This routine uses the wake up on key
                  press feature and reduces power
               ;  consumption by the PIC while not in use.
               ;**************************************
0033            KEY_SCAN
0033  3010      KEY_RELEASE  MOVLW B'00010000'
                ;Clearing PORTA pins but
0034  0585      ANDWF PORTA,F; Retaining the RA4 status
0035  0806      MOVF  PORTB,W ;Read PORTB into W reg
0036  39F0      ANDLW B'11110000';Mask the lower nibble
0037  3AF0      XORLW   B'11110000' ;W Xor 11110000 --
                >W
0038  1D03      BTFSS  STATUS,Z  ;Is all keys are
                released ?
0039  2833      GOTO  KEY_RELEASE ;If not goto
                                  KEY_RELEASE
003A  206C      CALL DEBOUNCE ;If yes debounce the  key
003B  0806      MOVF PORTB,W;Clear previous mismatch
                condition
003C  100B      BCF INTCON,RBIF ;Clear RBIF
003D  0063      SLEEP   ;Put the processor in Sleep mode
003E  3010      ANY_KEY MOVLW B'00010000' ;Clearing
                PORTA pins but
003F  0585      ANDWF  PORTA,F    ;Retaining  the  RA4
                status
040   0806      MOVF PORTB,W  ;PORTB -->W reg
0041  3AFF      XORLW 0XFF  ;W XOR 0XFF -->W reg
0042  1903      BTFSC STATUS,Z ;Is any key pressed ?
0043  283E      GOTO ANY_KEY  ;If no goto ANY_KEY
0044  206C      CALL DEBOUNCE;If yes debounce the key
0045  3000      MOVLW  0X00
0046  0090      MOVWF KEY_NO ;Initialise KEY_NO to 0
0047  3010      FIND_KEY MOVLW B'00010000'
0048  0585      ANDWF  PORTA,F    ;Retaining  the  RA4
status
0049  205E      CALL  SCAN_TABLE ;Get the scan code
004A  0091      MOVWF  SCAN_CODE  ;Move
                SCAN_CODE to W reg
004B  3907      ANDLW  B'00000111' ;Mask 5 MSB's
004C  0485      IORWF  PORTA,F  ;w --> porta while
               ;  Retaining the RA4 status
004D  0806      MOVF  PORTB,W  ;Read PORTB to W reg
004E  39F0      ANDLW  B'11110000'  ;Mask the lower
                nibble of PORTB
004F  008F      MOVWF  KEY_IN  ;Move the key value to
                key_in
0050  0811      MOVF SCAN_CODE,W ;SCAN_CODE -->
                W reg
0051  39F0      ANDLW  B'11110000' ;Mask lower nibble
                of scan code
0052  060F      XORWF  KEY_IN,W ;compare read key
                with scan code
0053  1903      BTFSC  STATUS,Z ;Test for Z flag
0054  285B      GOTO  RET  ;If Z=1 goto RET else continue
0055  0A90      INCF KEY_NO,F ;Increment key no
0056  0810      MOVF  KEY_NO,W ;KEY_NO -->W REG
0057  3C0C      SUBLW  0X0C  ; W - 12 -->W
0058  1D03      BTFSS  STATUS,Z ;Test whether key
                no=12th key
0059  2847      GOTO  FIND_KEY ;If no goto FIND_KEY
005A  2833      GOTO  KEY_SCAN ;If yes goto start new
                scan
005B  216C      RET  CALL    SHORT_BEEP ;Now the
buzzer will beep once
005C  0810      MOVF  KEY_NO,W ;Pressed Key no-->w
005D  0008      RETURN   ;Return from key scan
               ;**************************************
               ;  LOOK UP TABLE FOR KEY CODE
               ;  This look up table is used by the keyboard
                  scan subroutine and look up
               ;  table returns the scancode in w register
                  when called by placing key number
               ;  in KEY_NO
               ;**************************************
005E  0810      SCAN_TABLE MOVF KEY_NO,W ;KEY_NO
                -->W reg
005F  0782      ADDWF  PCL,F  ;PCL+W -->PCL reg
0060  34E6      RETLW  B'11100110'  ;Scan code for key0
0061  34E5      RETLW  B'11100101' ;Scan code for key1
0062  34E3      RETLW  B'11100011' ;Scan code for key2
0063  34D6      RETLW  B'11010110' ;Scan code for key3
0064  34D5      RETLW  B'11010101' ;Scan code for key4
0065  34D3      RETLW  B'11010011' ;Scan code for key5
0066  34B6      RETLW  B'10110110' ;Scan code for key6
0067  34B5      RETLW  B'10110101' ;Scan code for key7
0068  34B3      RETLW  B'10110011' ;Scan code for key8
0069  3476      RETLW  B'01110110' ;Scan code for key9
006A  3475      RETLW B'01110101' ;Scan code for L/U key
006B  3473      RETLW B'01110011;Scan code for CHG key
               ;**************************************
               ;  DELAY FOR DEBOUNCING THE KEY
               ;  This delay routine produces a key board
                  debounce delay of 20ms
               ;**************************************
006C  301C      DEBOUNCE  MOVLW  0X1C
006D  008D      MOVWF  DEL_COUNT2
006E  30F0      KB_DLOOP1  MOVLW  0XF0
006F  008C      MOVWF  DEL_COUNT1
```

```
0070 0B8C   KB_DLOOP  DECFSZ DEL_COUNT1,F
0071 2870             GOTO  KB_DLOOP
0072 0B8D             DECFSZ  DEL_COUNT2,F
0073 286E   GOTO  KB_DLOOP1
0074 0008   RETURN

            ;*************************************
            ;  ROUTINE FOR LOCKING /UNLOCKING
            ;  When you press L/U key the program control comes here.
            ;*************************************

0075 20A1   LCK_UNLCK CALL  GET_CH_NO ;Get channel/user no
0076 20B4   CALL  VRFY_PASWD ;Call verify password subroutine
0077 0398   DECF CH_NO,F  ;Decrement CH_NO
0078 0818   MOVF CH_NO,W  ;CH_NO -->W reg
0079 0798   ADDWF CH_NO,F  ;CH_NO x 2 -->CH_NO
007A 3003   MOVLW 0X03  ;Reset no_of_attempts to 3
007B 0097   MOVWF NO_OF_ATTEMPTS
007C 217D   CALL BEEP_THRICE;Now the buzzer will beep 3 times

007D 2033   SWITCH_RELAY CALL KEY_SCAN ;Call Key scan subroutine
007E 0092   MOVWF KB_TEMP  ;Store the key val in KB_TEMP
007F 3A01   XORLW 0X01
0080 1903   BTFSC STATUS,Z ;Is key 1 is pressed ?
0081 2887   GOTO RLY_ON  ;If yes goto RLY_ON
0082 0812   MOVF KB_TEMP,W
0083 3A00   XORLW 0X00
0084 1903   BTFSC STATUS,Z ;Is key 0 is pressed ?
0085 288A   GOTO RLY_OFF  ;If yes goto RLY_OFF
0086 2831   GOTO WRNG_ENTRY ;If no goto WRNG_ENTRY
0087 208D   RLY_ON  CALL RLY_ON_TBL ;Call RLY_ON table
0088 2178   CALL BEEP_TWICE ;Now the buzzer will beep twice
0089 2826   GOTO BEGIN  ;Goto BEGIN

008A 2097   RLY_OFF  CALL RLY_OFF_TBL ;Call RLY_OFF table
008B 2178   CALL BEEP_TWICE ;Now the buzzer will beep twice
008C 2826   GOTO BEGIN  ;Goto BEGIN

            ;*************************************
            ;  RELAY_ON_TABLE
            ;*************************************

008D 0818   RLY_ON_TBL MOVF CH_NO,W
008E 0782   ADDWF PCL,F
008F 1406   BSF PORTB,0  ;Switches ON ch1's relay
0090 0008   RETURN
0091 1486   BSF PORTB,1  ;Switches ON ch2's relay
0092 0008   RETURN
0093 1506   BSF PORTB,2  ;Switches ON ch3's relay
0094 0008   RETURN
0095 1586   BSF PORTB,3  ;Switches ON ch4's relay
0096 0008   RETURN

            ;*************************************
            ;  RELAY_OFF_TABLE
            ;*************************************

0097 0818   RLY_OFF_TBL MOVF CH_NO,W
0098 0782   ADDWF PCL,F
0099 1006   BCF PORTB,0  ;Switches OFF ch1's relay
009A 0008   RETURN
009B 1086   BCF PORTB,1  ;Switches OFF ch2's relay
009C 0008   RETURN
009D 1106   BCF PORTB,2  ;Switches OFF ch3's relay
009E 0008   RETURN
009F 1186   BCF PORTB,3  ;Switches OFF ch4's relay
00A0 0008   RETURN

            ;*************************************
            ;  This sub routine is used to take channel/user number and it also finds the staring
            ;  address of ch's/user's password stored in EEPROM using Lookup table and places it
            ;  in EEADDR_TEMP. This address will be used by COPY_TO_RAM subroutine.
            ;
            ;*************************************

00A1 2033   GET_CH_NO CALL  KEY_SCAN ;Ch/user no -->w
00A2 0098   MOVWF CH_NO  ;[W] --> CH_NO
00A3 3A00   XORLW  0X00
00A4 1903   BTFSC  STATUS,Z ;Is entered key is 0 ?
00A5 2831   GOTO  WRNG_ENTRY  ;If  yes   WRNG_ENTRY
00A6 0818   MOVF  CH_NO,W  ;If no CH_NO -->W
00A7 3C04   SUBLW  0X04  ;Is entered key > 4 ?
00A8 1C03   BTFSS  STATUS,C
00A9 2831   GOTO  WRNG_ENTRY ;If YES goto WRNG_ENTRY
00AA 20AD   CALL  EEADDR_LOOKUP ;If no CALL EEADDR look up table
00AB 0099   MOVWF EEADDR_TEMP ;[W] -->EEADDR_TEMP
00AC 0008   RETURN

            ;*************************************
            ;  LOOK UP TABLE FOR EEADDRESS
            ;  This Lookup table returns the staring address of the ch's/user's password in
            ;  EEPROM data memory when the channel/user number is passed into it.
            ;*************************************

00AD 0818   EEADDR_LOOKUP MOVF  CH_NO,W
00AE 0782   ADDWF  PCL,F
00AF 0008   RETURN
00B0 3400   RETLW 0X00  ;Starting address of ch1's Passward
00B1 3410   RETLW  0X10  ;Starting address of ch2's Passward
00B2 3420   RETLW  0X20  ;Starting address of ch3's Passward
00B3 3430   RETLW  0X30  ;Starting address of ch4's Passward

            ;*************************************
            ;
            ;  SUBROUTINE TO VERIFY PASSWARD
            ;
            ;  This subroutine copies the passward saved in EEPROM into RAM_BUF1 then reads the
            ;  passward digits entered by the user and stores into RAM_BUF2 then compares
            ;  RAM_BUF1 with RAM_BUF2 digit by digit.
            ;  Returns to the called program if the match occures for all the digits. On mismatch it
            ;  gives an long error beep and decrements the NO_OF_ATTEMPTS by one. If
            ;  NO_OF_ATTEMPTS == 0 switches the code lock into alarm mode. and further
            ;  key presses will be ignored.The codelock comes to the normal working after 1  minute.
            ;  NOTE:the NO_OF_ATTEMPTS will not be
            ;  decremented if the jumper is placed
            ;  between RA4 and Gnd and hence will not switch into the alarm mode.
            ;*************************************

00B4 20EB   VRFY_PASWD CALL  COPY_TO_RAM ;Call COPY_TO_RAM sub routine
00B5 3030   MOVLW  RAM_BUF1
00B6 3E0F   ADDLW 0X0F  ;Initialize FSR to
00B7 0084   MOVWF  FSR  ;the end of RAM_BUF1
00B8 0800   MOVF  INDF,W  ;[INDF] -->W
00B9 00A4   MOVWF  NO_OF_DIGITS ;[W] -->NO_OF_DIGITS
00BA 0095   MOVWF  DIGIT_COUNT ;[W] -->DIGIT_COUNT
00BB 3040   MOVLW  RAM_BUF2 ;Initialise FSR to
00BC 0084   MOVWF  FSR  ;the starting of RAM_BUF2

00BD 2033   SCAN_NXT_BYTE CALL KEY_SCAN ;Call scan key routine
00BE 0080   MOVWF INDF   ;[W]-->INDF
00BF 3C09   SUBLW 0X09
00C0 1C03   BTFSS STATUS,C ;Is L/U or CHG key pressed ?
00C1 2831   GOTO  WRNG_ENTRY ;If yes goto WRNG_ENTRY
00C2 0A84   INCF  FSR,F  ;Increment FSR by 1
00C3 0B95   DECFSZ  DIGIT_COUNT,F ;Decrement DIGIT_COUNT by one,is it 0?
00C4 28BD   GOTO  SCAN_NXT_BYTE ;If no go back to SCAN_NXT_BYTE
```

```
                00399
00C5 3030       00400   COMPARE  MOVLW  RAM_BUF1
                        ;RAM_BUF1 pointer initialisation
00C6 0093       00401   MOVWF  RAM_BUF1_PNT
00C7 3040       00402   MOVLW  RAM_BUF2
00C8 0094       00403   MOVWF  RAM_BUF2_PNT ;RAM_BUF2
                        pointer initialisation
00C9 0824       00404   MOVF  NO_OF_DIGITS,W
00CA 0095       00405   MOVWF  DIGIT_COUNT
                        ;[NO_OF_DIGITS] --> DIGIT_COUNT
                        00406
00CB 0813       00407   COMP_CONT MOVF  RAM_BUF1_PNT,W
                        ;[RAM_BUF1_PNT] -->W
00CC 0084       00408   MOVWF  FSR  ;[W]-->FSR
00CD 0800       00409   MOVF  INDF,W  ;passward digit --> w
                        reg 1 by 1
00CE 0096       00410   MOVWF  PSD_DIGIT ;[W] -->PSD_DIGIT
00CF 0814       00411   MOVF  RAM_BUF2_PNT,W
                        ;[RAM_BUF2_PNT] -->W
00D0 0084       00412   MOVWF  FSR   ;[W]-->FSR
00D1 0816       00413   MOVF  PSD_DIGIT,W ;[PSD_DIGIT] -->W
00D2 0600       00414   XORWF  INDF,W  ;[W] xor [RAM_BUF2]
                        -->W
00D3 1D03       00415   BTFSS  STATUS,Z ;Is Z==1 ?
00D4 28DA       00416   GOTO  WARN  ;If no goto WARN
00D5 0A93       00417   INCF  RAM_BUF1_PNT,F ;If yes increment
                        RAM_BUF1_PNT by 1
00D6 0A94       00418   INCF  RAM_BUF2_PNT,F ;Increment
                        RAM_BUF2_PNT by 1
00D7 0B95       00419   DECFSZ DIGIT_COUNT,F ;Decrement
                        DIGIT_COUNT by 1, is it 0 ?
00D8 28CB       00420   GOTO  COMP_CONT  ;If no goto compare
                        nxt digit
00D9 0008       00421   RETURN   ;If yes Return back
                00422
                00423
00DA 2172       00424   WARN CALL LONG_BEEP ;Make a long
                        beep
00DB 0B97       00425   DECFSZ NO_OF_ATTEMPTS,F; Decrement
                        NO_OF_ATTEMPTS,is it 0 ?
00DC 2826       00426   GOTO  BEGIN  ;If no goto BEGIN
00DD 1585       00427   ALARM  BSF  PORTA,3  ;Switch ON the
                        buzzer
00DE 1A05       00428   BTFSC PORTA,4  ;Is the jumper placed?
00DF 28EA       00429   GOTO LATCH_ALARM ;If not goto
                        latch_alarm
                00430 ; If yes auto reset after 1 min
                00431 ;*************************************
                00432 ; program now inactivates the codelock for
                        1 minute
                00433 ; 1min = 1uS(instuction cycle) x
                        256(prescalar count) x(195)tmr0 counts
                        x200 x6
                00434 ; count to be loaded in TMR0 = (256 -195)
                        +2 =H'3F'
                00435 ; 2 is added because after moving a value to
                        TMR0 reg the actual
                00436 ; incremetation of TMR0 delays by 2 TMR0
                        clock cycles.
                00437 ;-----------------------------------------------------------
                00438
00E0 110B       00439   ONE_MIN_DEL BCF INTCON,T0IF ;Clear
                        TMR0 interrupt flag
00E1 168B       00440   BSF  INTCON,T0IE  ;Enable  TMR0
                        interrupt feature
00E2 3006       00441   MOVLW 0X06  ;Count for one minute
00E3 00A3       00442   MOVWF ONE_MIN_CNT
00E4 30C8       00443   MOVLW 0XC8  ;Count required to obtain
                        10s delay
00E5 00A2       00444   MOVWF TEN_SEC_CNT
00E6 303F       00445   MOVLW  0X3F  ;Count required to obtain
                        50ms delay
00E7 0081       00446   MOVWF TMR0
00E8 178B       00447   BSF INTCON,GIE
                00448
00E9 28E9       00449   INFI_LOOP  GOTO  INFI_LOOP ;Simply
                        loop here until 1 min
                00450 ;-----------------------------------------------------------
                00451 ; The program control comes here only if
                        the jumper is not placed.(see ckt dia)
                00452
00EA 28EA       00453   LATCH_ALARM GOTO  LATCH_ALARM
                        ;Simply lopp here until manual reset
                00454 ; by power interruption.
                00455
                00456
                00457 ;*************************************
                00458 ; ROUTINE TO COPY EEPROM CONTENT
                        TO RAM
                00459 ;*************************************
                00460
00EB            00461   COPY_TO_RAM
00EB 3030       00462   MOVLW  RAM_BUF1 ;Initialize FSR to the
00EC 0084       00463   MOVWF  FSR  ;Staring address of
                              RAM_BUF1
00ED 3010       00464   MOVLW D'16'
00EE 0095       00465   MOVWF  DIGIT_COUNT ;NO_OF_DIGITS
                        = 16 digits
00EF 0819       00466   MOVF EEADDR_TEMP,W
                        ;[EEADDR_TEMP] --> W
00F0 0089       00467   MOVWF  EEADR   ;[W] -->EEADR
                00468
00F1 1683       00469   COPY_NXT_BYTE BSF  STATUS,RP0
                        ;Select bank1
00F2 1408       00470   BSF  EECON1,RD ;Enable Read mode
00F3 1283       00471   BCF  STATUS,RP0 ;Select bank0
00F4 0808       00472   MOVF  EEDATA,W ;[EEDATA]-->w
00F5 0080       00473  MOVWF  INDF  ;[W]-->INDF
00F6 0A84       00474   INCF  FSR,F  ;Increment FSR by 1
00F7 0A89       00475   INCF  EEADR,F  ;Increment EEADR by 1
00F8 0B95       00476   DECFSZ  DIGIT_COUNT,F ;Decrement
                        DIGIT_COUNT by 1,is it 0 ?
00F9 28F1       00477   GOTO  COPY_NXT_BYTE  ;If no goto
                        COPY_NXT_BYTE
00FA 0008       00478   RETURN   ;If yes return
                00479
                00480
                00481 ;*************************************
                00482 ; ROUTINE TO CHG PASSWARD
                00483 ;
                00484 ; The program control comes here when you
                        press CHG key.First this subroutine asks
                00485 ; for channel no then old passward if the
                        entered information is correct, it takes the
                00486 ; new passward.then again takes the new
                        passward for confirmation. on confirmation
                00487 ; on confirmation success old pasward will
                        be replaced by the  new passward. On
                00488 ; confirmation error the old passward will
                        not be altered.
                00489 ;*************************************
                00490
00FB 20A1       00491   CHG_PSWD CALL  GET_CH_NO ;Get the
                        user/channel no
00FC 20B4       00492   CALL  VRFY_PASWD ;Veryfy the old
                        passward
00FD 217D       00493   CALL  BEEP_THRICE ;Beep thrice on
                        verificatin success
00FE 3003       00494   MOVLW 0X03 ;Reset NO_OF_ATTEMPTS
                        to 3
00FF 0097       00495   MOVWF NO_OF_ATTEMPTS
0100  3040      00496   MOVLW  RAM_BUF2 ;Initialise FSR to the
101   0084      00497   MOVWF  FSR  ;Starting address of
                              RAM_BUF2
0102  01A4      00498   CLRF NO_OF_DIGITS ;NO_OF_DIGITS=0
                00499
0103  2033      00500   GET_NXT_BYTE CALL  KEY_SCAN ;Call
                        key scan routine
0104  0080      00501   MOVWF  INDF   ;[W] -->INDF
0105  0092      00502   MOVWF KB_TEMP  ;[W] --> KB_TEMP
0106  3A0A      00503   XORLW 0X0A
0107  1903      00504   BTFSC STATUS,Z ;Is L/U key pressed ?
0108  2831      00505   GOTO WRNG_ENTRY ;If yes goto
                        WRNG_ENTRY
0109  0812      00506   MOVF KB_TEMP,W ;If no KB_TEMP-->W
010A  3A0B      00507   XORLW 0X0B
010B  1903      00508   BTFSC STATUS,Z ;Is CHG key pressed ?
010C  2910      00509   GOTO PROCEDE  ;If yes goto PROCEDE
010D  0AA4      00510   INCF  NO_OF_DIGITS,F ;If no increment
                        NO_OF_DIGITS by 1
010E  0A84      00511   INCF FSR,F  ;Increment FSR by 1
010F  2903      00512   GOTO  GET_NXT_BYTE ;Goto
                        GET_NXT_BYTE
                00513
0110  0824      00514   PROCEDE  MOVF NO_OF_DIGITS,W
                        ;[NO OF DIGITS] -->W
0111  0095      00515   MOVWF  DIGIT_COUNT ;[W]
                        -->DIGIT_COUNT
0112  3C03      00516   SUBLW 0X03  ;Is new password
0113  1803      00517   BTFSC STATUS,C ;contains < 4 digits ?
0114  2831      00518   GOTO WRNG_ENTRY ;If yes goto
                        WRNG_ENTRY
0115  0824      00519   MOVF NO_OF_DIGITS,W ;If no W
                        --> NO_OF_DIGITS
0116  3C0F      00520   SUBLW D'15'  ;Is new password
0117  1C03      00521   BTFSS STATUS,C ;contains --> >15 digits?
0118  2831      00522   GOTO WRNG_ENTRY ;If yes goto
                        WRNG_ENTRY
```

```
0119 2178    CALL BEEP_TWICE ;If no beep twice
011A 3030    MOVLW RAM_BUF1 ;Initialise FSR to the
011B 0084    MOVWF FSR ;starting address of RAM_BUF1
011C 2033    GET_NXT_BYTE2 CALL KEY_SCAN ;Call scan key routine
011D 0080    MOVWF INDF ;[W] -->INDF
011E 3C09    SUBLW 0X09 ;[W] - 0x09 -->W
011F 1C03    BTFSS STATUS,C ;Is L/U key is pressed ?
0120 2831    GOTO WRNG_ENTRY ;If yes goto WRNG_ENTRY
0121 0A84    INCF FSR,F ;If no increment FSR by 1
0122 0B95    DECFSZ DIGIT_COUNT,F ;Decrement DIGIT_COUNT by 1,is it 0 ?
0123 291C    GOTO GET_NXT_BYTE2 ;If yes goto GET_NXT_BYTE2
0124 3030    MOVLW RAM_BUF1 ;RAM_BUF1_PNT initialisation
0125 0093    MOVWF RAM_BUF1_PNT
0126 3040    MOVLW RAM_BUF2 ;RAM_BUF2_PNT initialisation
0127 0094    MOVWF RAM_BUF2_PNT
0128 0824    MOVF NO_OF_DIGITS,W ;[No of digits] -->W
0129 0095    MOVWF DIGIT_COUNT ;[W] -->DIGIT_COUNT
012A 0813    CONFRM_PSD MOVF RAM_BUF1_PNT,W
012B 0084    MOVWF FSR ;[RAM_BUF1_PNT] -->FSR
012C 0800    MOVF INDF,W ;[RAM_BUF1]-->W
012D 0096    MOVWF PSD_DIGIT ;[W]-->PSD_DIGIT
012E 0814    MOVF RAM_BUF2_PNT,W ;[RAM_BUF2_PNT] -->W
012F 0084    MOVWF FSR ;[W]-->FSR
0130 0816    MOVF PSD_DIGIT,W ;[PSD_DIGIT] -->W
0131 0200    SUBWF INDF,W ;[W]-[RAM_BUF2]-->W
0132 1D03    BTFSS STATUS,Z ;Is [RAM_BUF1]==[RAM_BUF2] ?
0133 295D    GOTO CONFRM_ERR ;If no goto CONFRM_ERR
0134 0A93    INCF RAM_BUF1_PNT,F ;If yes increment RAM_BUF1_PNT by 1
0135 0A94    INCF RAM_BUF2_PNT,F ;Increment RAM_BUF2_PNT by 1
0136 0B95    DECFSZ DIGIT_COUNT,F ;Decrement DIGIT_COUNT by 1,is it 0?
0137 292A    GOTO CONFRM_PSD ;If no goto CONFRM_PSD
0138 3040    MOVLW RAM_BUF2 ;If yes point to the
0139 3E0F    ADDLW 0X0F ;end of RAM_BUF2
013A 0084    MOVWF FSR
013B 0824    MOVF NO_OF_DIGITS,W ;Store the no of digits
013C 0080    MOVWF INDF ;in the password at the end of
013D 3040    MOVLW RAM_BUF2 ;RAM_BUF2
013E 0084    MOVWF FSR
013F 3010    START_EE_WR MOVLW D'16' ;No of bytes to write = 16
0140 0095    MOVWF DIGIT_COUNT
0141 0819    MOVF EEADDR_TEMP,W ;Set initial EEPROM address
0142 0089    MOVWF EEADR
0143 1283    BCF STATUS,RP0 ;Select bank0
0144 138B    BCF INTCON,GIE ;Dissable all interrupts
0145 0800    WR_EEPROM MOVF INDF,W ;[INDF] --> W
0146 0088    MOVWF EEDATA ;W -->EEDATA
0147 1683    BSF STATUS,RP0 ;Select bank1
0148 1508    BSF EECON1,WREN ;Enable write mode
0149 3055    MOVLW 0X55
014A 0089    MOVWF EECON2 ;H'55' must be written to eecon2
014B 30AA    MOVLW 0XAA ;to start write sequence
014C 0089    MOVWF EECON2 ;followed by H'AA'
014D 1488    BSF EECON1,WR ;Set WR bit to start writing
014E 1E08    POLL_EEIF BTFSS EECON1,EEIF ;Is write complete ?
014F 294E    GOTO POLL_EEIF ;If no goto POLL_EEIF
0150 1208    BCF EECON1,EEIF ;If yes clear EEIF bit
0151 1988    BTFSC EECON1,WRERR ;Is WRERR is set?
0152 2945    GOTO WR_EEPROM ;If set write again
0153 0A84    INCF FSR,F ;Increment FSR by 1
0154 1283    BCF STATUS,RP0 ;Select bank0
0155 0A89    INCF EEADR,F ;Increment EEADR by 1
0156 0B95    DECFSZ DIGIT_COUNT,F ;Decrement DIGIT_COUNT by1 ,is it 0 ?
0157 2945    GOTO WR_EEPROM ;If NO go to write next digit.
0158 1683    BSF STATUS,RP0 ;If yes select bank0
0159 1108    BCF EECON1,WREN ;Dissable Write mode
015A 1283    BCF STATUS,RP0 ;Select bank0
015B 217D    CALL BEEP_THRICE ;Beep thrice
015C 2826    GOTO BEGIN ;Goto BEGIN
015D 2172    CONFRM_ERR CALL LONG_BEEP;Give a long beep on confirm Error
015E 2826    GOTO BEGIN ;Goto BEGIN
             ;**************************************
             ; DELAY SUBROUTINE FOR BUZZER ON AND OFF TIME
             ;**************************************
015F 0821    BUZ_DELAY MOVF BUZ_DEL_CNT,W
0160 008C    MOVWF DEL_COUNT1
0161 3040    BUZ_LOOP1 MOVLW 0X40
0162 008D    MOVWF DEL_COUNT2
0163 30FE    BUZ_LOOP2 MOVLW 0XFE
0164 008E    MOVWF DEL_COUNT3
0165 0B8E    BUZ_LOOP3 DECFSZ DEL_COUNT3,F
0166 2965    GOTO BUZ_LOOP3
0167 0B8D    DECFSZ DEL_COUNT2,F
0168 2963    GOTO BUZ_LOOP2
0169 0B8C    DECFSZ DEL_COUNT1,F
016A 2961    GOTO BUZ_LOOP1
016B 0008    RETURN
             ;**************************************
             ; SUBROUTINES TO SOUND BUZZER
             ;**************************************
016C 3001    SHORT_BEEP MOVLW 0X01 ; Subroutine to produce a short beep
016D 00A1    MOVWF BUZ_DEL_CNT
016E 1585    BSF PORTA,3
016F 215F    CALL BUZ_DELAY
0170 1185    BCF PORTA,3
0171 0008    RETURN
0172 300A    LONG_BEEP MOVLW 0X0A ;Subroutine to produce a long beep
0173 00A1    MOVWF BUZ_DEL_CNT
0174 1585    BSF PORTA,3
0175 215F    CALL BUZ_DELAY
0176 1185    BCF PORTA,3
0177 0008    RETURN
0178 3005    BEEP_TWICE MOVLW 0X05
0179 00A1    MOVWF BUZ_DEL_CNT
017A 215F    CALL BUZ_DELAY
017B 3002    MOVLW 0X02 ;Subroutine to produce 2 short beeps
017C 2982    GOTO BEEP_NOW
017D 3005    BEEP_THRICE MOVLW 0X05
017E 00A1    MOVWF BUZ_DEL_CNT
017F 215F    CALL BUZ_DELAY
0180 3003    MOVLW 0X03 ;Subroutine to produce 3 short beeps
0181 2982    GOTO BEEP_NOW
0182 00A0    BEEP_NOW MOVWF NO_OF_BEEPS
0183 3004    BEEP_AGAIN MOVLW 0X04
0184 00A1    MOVWF BUZ_DEL_CNT
0185 215F    CALL BUZ_DELAY
0186 216C    CALL SHORT_BEEP
0187 0BA0    DECFSZ NO_OF_BEEPS,F
0188 2983    GOTO BEEP_AGAIN
0189 0008    RETURN
             END ;The progam ends here ❑
```

Readers' comments

I have the following queries:

Q1. Can I changeover from PIC16F84A to PIC16F628? PIC16F628 is a cheaper microcontroller that is pin-compatible with PIC16F84A. It is readily available from Microchip, which is already phasing out PIC16F84A for the last three years.

Q2. Is it possible to change the length of the password?

Q3. Can only one output be used?

Q4. Can an alphanumeric keypad be used?

Jatinder Chawla
Through e-mail

The author, Vijaya Kumar P., replies:

A1. I think you are talking of older PIC16F84 and PIC16C84. At present, PIC16F84A is widely available in India. I have been informed by Microchip's technical support (e-mail: taiwan.techhelp@microchip.com) that PIC16F84A is still available in the '2004 Products Selector Guide.' I haven't seen any phase-out note on this device. For reference, you may check out the 'Products Selector Guide' in 'Product Document List' on Microchip's website 'www.microchip.com'. I have used PIC16F84A microcontroller because EFY readers can find its programmer in Sept. 2002 issue of the magazine. Of course, you can also use PIC16F628 and PIC16F627 microcontrollers, which are pin-compatible with PIC16F84A, but this requires a few modifications in the program.

A2. It is clearly mentioned in the article that the password length can be changed from four digits to upto 15 digits as desired by the user.

A3. Yes, if you want only one channel, one output can be used. You can use any one of the channels, say, Channel 1. Then you don't need relays RL2 through RL4 and the associated components, i.e., transistors T3 through T5, resistors R8 through R10, diodes D2 through D5, and LED2 through LED4.

A4. An alphanumeric keypad cannot be used with the circuit because, to include alphanumeric characters, we'll have to use separate keys for each alphabet and numeral. This requires more number of input/output (I/O) port pins to implement the matrix keyboard. Since there are no additional free pins available in this application, the method you proposed is not possible.

Another way is to use a keypad similar to the one used in mobile handsets. Here, each key is multiplexed with a digit and one or more characters. The digit/character inputted depends upon the number of pressing actions within fixed time duration. But this type of keypad implementation requires a display device, such as LCD, to ensure that the correct key is pressed. Again, this requires more I/O pins to interface the LCD and the project becomes costly. Moreover, displaying the password entered in this case is not a secure way!

LOAD PROTECTOR WITH REMOTE SWITCHING FACILITY

S. SIVARAMAKRISHNAN

For inverters and UPS systems, the load should not be much below or above the rated power since it can cause excess heating of the output transformer windings and the active driving device and thereby damage them. Some domestic appliances also need to be protected against under-/over-voltage.

Here's an under-/over-voltage protector to protect devices from fluctuations in the mains. It also allows you to turn on/off the load through a remote handset. Its main features are:

1. It shuts down the load at under-/over-voltage.

2. After under-/over-voltage, the load is automatically restarted.

3. A visual indication is given for normal voltage, which goes off at under-/over-voltage.

Circuit description

Fig. 1 shows the block diagram of the remote-controlled load protector. Basically, it comprises a transformer, rectifier, filter, regulator and comparator along with remote switching transmitter and receiver circuitry. The remote signal transmitter is used for remote switching of the device during the normal voltage.

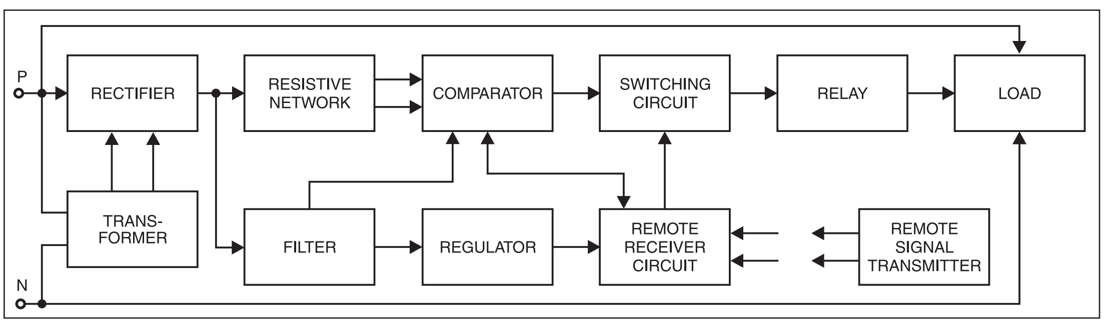


Fig. 1: Block diagram of the load protector with remote switching facility

The AC mains is stepped down by the transformer, rectified, filtered and then applied to the comparator as well as the regulator. The regulator provides 12V regulated power supply to the circuit excluding the comparator and the timer.

The circuit comprises two sections, namely, the transmitter (remote handset) and the receiver-cum-load protector.

The remote transmitter. Fig. 2 shows the remote transmitter built around astable multivibrator IC NE555 (IC1). Powered by a 9V battery, the remote transmitter transmits a preset frequency when push-to-on switch S1 is pressed. The modulated IR beam is received by phototransistor T1 of the receiver-cum-load protector unit.

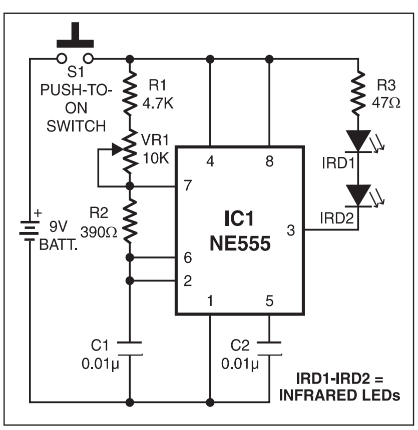


Fig. 2: Remote handset

The receiver-cum-load protector. Fig. 3 shows the receiver-cum-load protector circuit comprising power supply, receiver and protector sections.

Power supply. The circuit is powered by AC mains through fuse F1. The AC mains is stepped down by transformer X1 to deliver a secondary output of 18V-0-18V, 250mA. The transformer output is rectified by diodes D1 and D2 and filtered by capacitor C4. The filtered output is fed to IC2 and also the junction of resistor R17 and preset VR3. IC2 provides 12V regulated supply to the circuit. The output of IC2 is smoothed by capacitor C3.

Receiver. The receiver section is built around transistors T1 through T3 and IC3 through IC5. Darlington-pair phototransistor T1 is used to sense the infrared signals. The phototransistor is sensitive to the incident radiation. The incident photons result in a base current, which is amplified by the gain of the photo-darlington.

The frequency signals from the phototransistor are amplified by npn transistors T2 and T3 and applied to phase-locked loop IC NE567 (IC5) through capacitor C10. IC NE567 is a highly stable phase-locked loop with synchronous AM lock detection and power output circuitry. It is primarily used as a tone decoder, which drives a load whenever a sustained frequency falling within its detection band is present at its self-biased input. The centre frequency of the band and the output delay are independently determined by external components.

IC5 detects the code frequency. In the absence of any input signal, the centre frequency of its internal free-running, current-controlled oscillator is determined by preset VR4 and capacitor C9. Preset

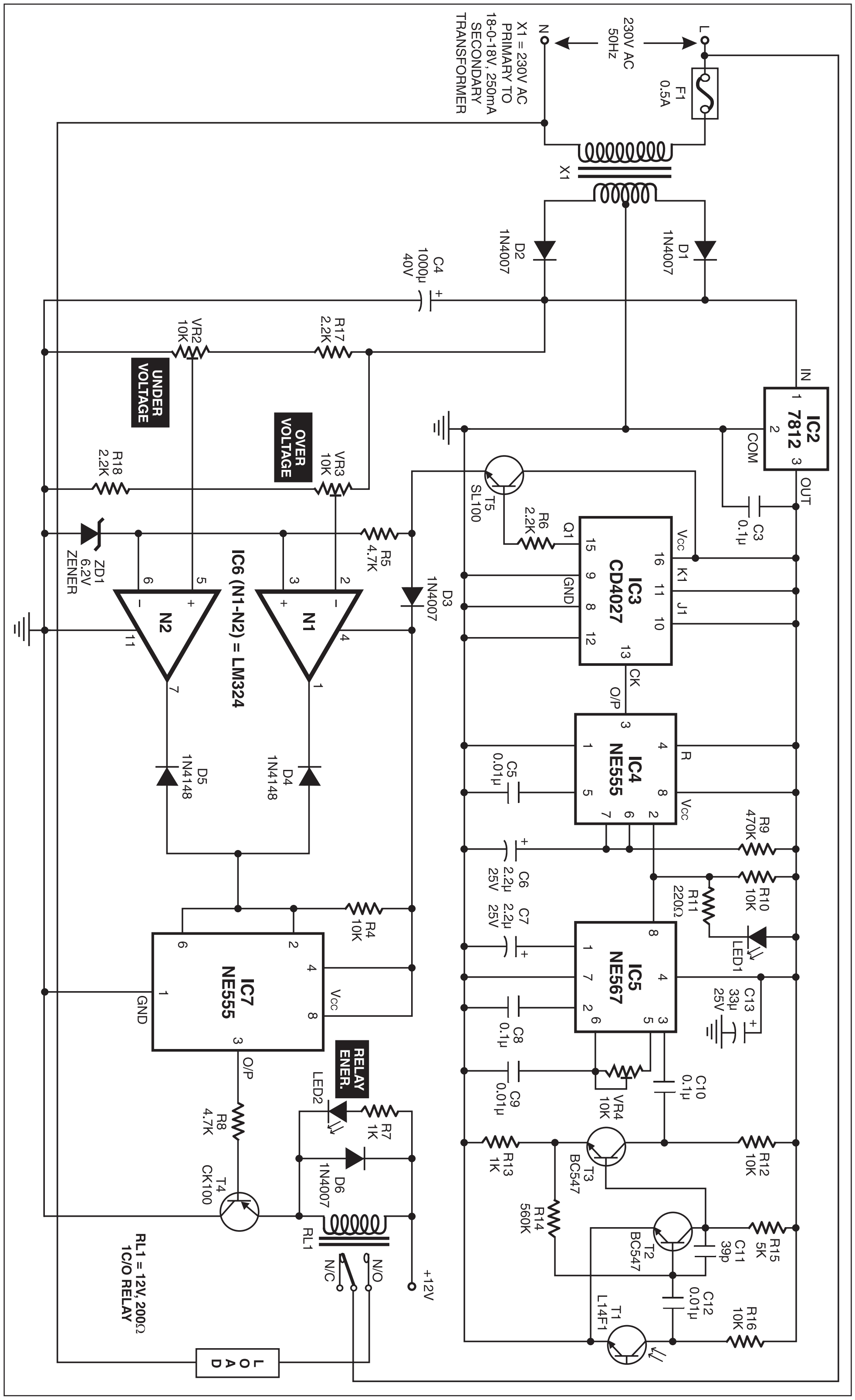
VR4 is used for tuning IC5 to the desired centre frequency in the code frequency range, which should match the frequency of the code generator in the transmitter. Low-pass frequency is determined by capacitor C8. Capacitor C7 attenuates frequencies outside the detection band to eliminate spurious outputs.

The output at pin 8 of IC5 remains low as long as the transmitted code frequency is detected by IC5. LED1 lights up to indicate detection of the transmitted signal. The output of IC5 triggers monostable multivibrator IC4, whose time period is about one second. IC4, in turn, provides clock signal to the JK flip-flop IC CD4027 (IC3) wired in toggle configuration.

When IC3 gets the first clock signal, its Q1 output (pin 15) goes high. On the next clock pulse, Q1 output goes low. Q1 output of IC3 is fed to the base of transistor T5 through resistor R6. Transistor T5 provides supply to comparator IC6 and timer IC7 only when Q1 is high.

Load protector. The load protector unit is built around diodes D1 and D2, comparator IC6 and timer IC7. The comparator is built around operational amplifier IC LM324 (IC6). It consists of four independent high-gain, frequency-compensated operational amplifiers that are designed specifically to operate from a single supply over a wide range of the voltages. The reference voltage (6.2V) generated by resistor R5 and zener diode ZD1 is provided to non-inverting pin 3 and inverting pin 6 of operational amplifiers N1 and N2, respectively. Zener diode ZD1 stabilises the reference voltage. Presets VR2 and VR3 are used for setting the under- and over-voltage at non-inverting pin 5 and inverting pin 2 of operational amplifiers N2 and N1, respectively.

Fig. 3: Circuit of the receiver-cum-load-protector



Pins 2 and 6 of IC7 are pulled high through resistor R4. Diodes D4 and D5 are used for wired-OR operation. Whenever the output of any one of the comparators (N1 or N2) goes low, the output coupled to pin 2 of IC7 goes low to trigger it. This happenes when under-/over-voltage conditions are encountered. As a result, the output of IC7 goes high to cut off transistor T4 and de-energise relay RL1.

IC NE555 (IC7) behaves like a level-sensing device. In the normal voltage condition, its low output drives pnp transistor

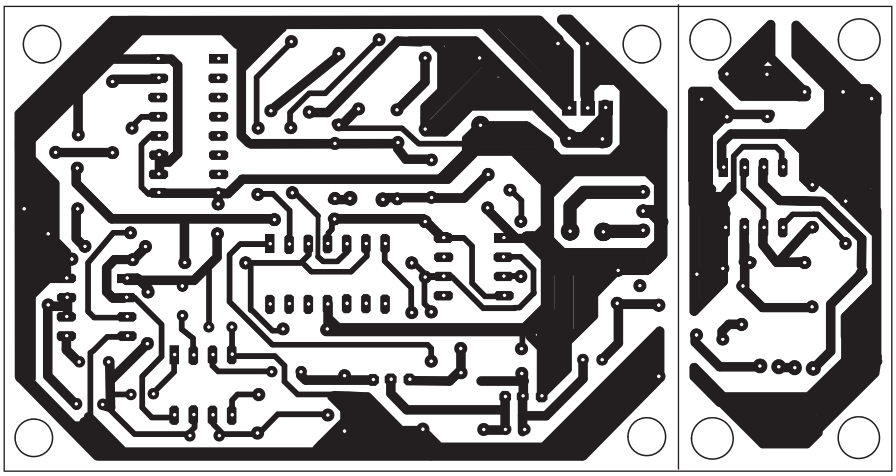


Fig. 4: Actual-size, single-side combined PCB layout for the remote handset (Fig. 2) and receiver-cum-load protector (Fig. 3)

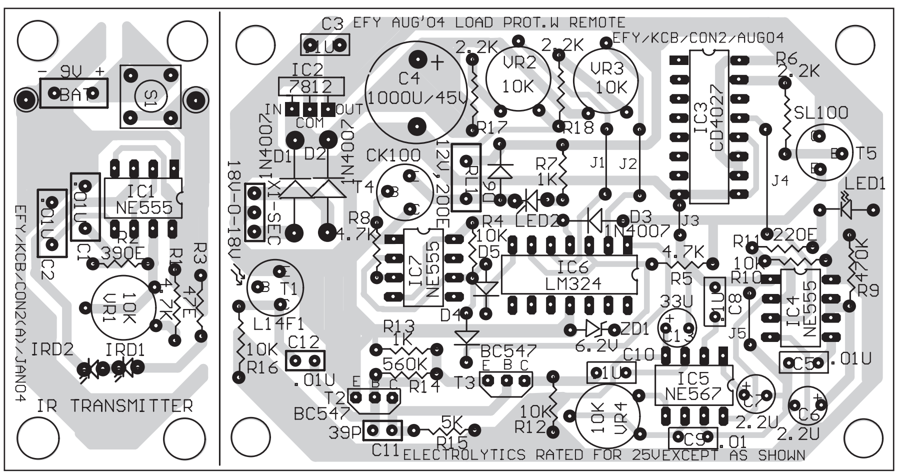


Fig. 5: Component layout for the PCB

PARTS LIST

Semiconductors:

IC1, IC4, IC7	- NE555 timer
IC2	- 7812, 12V regulator
IC3	- CD4027 dual JK flip-flop
IC5	- NE567 phase-locked loop
IC6	- LM324 comparator
ZD1	- 6.2V zener diode
D1-D3	- 1N4007 rectifier diode
D4, D5	- 1N4148 switching diode
T1	- L14F1 phototransistor
T2, T3	- BC547 npn transistor
T4	- CK100 pnp transistor
T5	- SL100 npn transistor
IRD1, IRD2	- Infrared diodes/LEDs
LED1, LED2	- Red LED

Resistors (all ¼-watt, ±5% carbon, unless stated otherwise):

R1, R5, R8	- 4.7-kilo-ohm
R2	- 390-ohm
R3	- 47-ohm
R4, R10, R12, R16	- 10-kilo-ohm
R6, R17, R18	- 2.2-kilo-ohm
R7, R13	- 1-kilo-ohm
R9	- 470-kilo-ohm
R11	- 220-ohm
R14	- 560-kilo-ohm
R15	- 5-kilo-ohm
VR1-VR4	- 10-kilo-ohm preset

Capacitors:

C1-C2, C5, C9, C12	- 0.01µF ceramic disk
C3, C8, C10	- 0.1µF ceramic disk
C4	- 1000µF, 40V electrolytic
C6, C7	- 2.2µF, 25V electrolytic
C11	- 39pF ceramic disk
C13	- 33µF, 25V electrolytic

Miscellaneous:

E1	- 500mA fuse
RL1	- 12V, 200Ω 1C/O relay
X1	- 230V AC to 18V-0-18V, 250mA secondary transformer

T4 into conduction to energise relay RL1 and operate the device connected to the contacts of the relay. Diode D6 is used as a free-wheeling diode. LED2 indicates relay energisation and device 'on' condition.

Working

If mains voltage is less than 245V but more than 200V, the output of IC2 is low and relay RL1 energises via relay-driver pnp transistor T5 to provide mains to the load (device) to be protected.

When mains voltage increases beyond 245V, which also means that the sampled voltage at pin 2 becomes higher than the reference voltage (6.2V), the output of N1 at pin 1 goes low to trigger IC7. As a result, the output of IC7 goes high to de-energise the relay via relay-driver pnp transistor T4 and LED1 stops glowing to indicate that the device is switched off (protected from over-voltage).

Similarly, when mains voltage goes below 200V, which also means that the voltage at pin 5 goes below the reference voltage (6.2V), the output of N2 at pin 7 goes low to trigger IC7. The triggered IC7 provides a high output to de-energise the relay via relay-driver pnp transistor T4 and LED1 stops glowing to indicate that the load (device) is protected from under-voltage.

Remote switching of the load

At the normal mains voltage, the load (device) connected across the normally-opened (N/O) contacts of relay RL1 is in 'on' condition. Now if you want to switch off the load, simply press switch S1 on the remote handset momentarily. As a result, relay RL1 de-energises to disconnect the load from mains. This happens because the output of IC3 (pin 15) goes low on pressing switch S1 on the remote transmitter, which inhibits the power supply for IC6 and IC7, and relay RL1 de-energises. Similarly, you can switch on the load again by pressing S1 momentarily, which toggles IC3 to re-establish supply for IC6 and IC7. Thereafter, the cycle repeats if switch S1 on the remote is pressed again and again. The remote handset can control devices from a distance of up to 8 metres.

Construction

The circuit of the load protector with remote switching facility can be assembled on any general-purpose PCB. However, the actual size, single-side combined PCB layout for the remote handset (Fig. 2) and the receiver-cum-load protector (Fig. 3) is shown in Fig. 4 and its component layout in Fig. 5.

It would help to rectify any problem if you use IC bases instead of directly soldering the ICs onto the PCB. Ensure proper contacts between pins of the IC bases and the solder points on the PCB. ❑

VOICE RECORDING AND PLAYBACK USING APR9600 CHIP

K. KRISHNA MURTY

Digital voice processing chips with different features and coding techniques for speech compression and processing are available on the market from a number of semiconductor manufacturers. Advanced chips such as Texas instruments' TMS320C31 can implement various voice-processing algorithms including code-excited linear prediction, adaptive differential pulse-code modulation, A law (specified by California Council for International Trade), µ law (specified by Bell Telephone) and vector sum-excited linear prediction.

On the other hand, APR9600 single-chip voice recorder and playback device from Aplus Integrated Circuits makes use of a proprietary analogue storage technique implemented using flash non-volatile memory process in which each cell is capable of storing up to 256 voltage levels. This technology enables the APR9600 to reproduce voice signals in their natural form.

The APR9600 is a good standalone voice recorder or playback IC with non-volatile storage and playback capability for 32 to 60 seconds. It can record and play multiple messages at random or in sequential mode. The user can select sample rates with consequent quality and recording time trade-off. Microphone amplifier, automatic gain control (AGC) circuits, internal anti-aliasing filter, integrated output amplifier and messages management are some of the features of the APR9600 chip.

Fig. 1 shows the functional block diagram of IC APR9600. Complete chip management is accomplished through the device control and message control blocks.

Voice signal from the microphone (see Fig. 2) is fed into the chip through a

PARTS LIST

Semiconductors:		
IC1	-	APR9600 voice processor
IC2	-	LM386 low-power audio amplifier
T1-T3	-	BC557 pnp transistor
D1	-	1N4001 rectifier diode
LED1-LED3	-	Red LED
Resistors (all ¼-watt, ±5% carbon, unless stated otherwise):		
R1, R2, R4-R8, R16, R17	-	100 kilo-ohm
R3, R10	-	390-ohm
R9	-	220-kilo-ohm
R11	-	1-ohm
R12(A)	-	33-kilo-ohm
R12(B)	-	5-kilo-ohm
R13, R14	-	4.7-kilo-ohm
R15	-	1-kilo-ohm
Capacitors:		
C1, C3, C4, C6, C8, C9, C11	-	0.1µF ceramic disk
C2	-	4.7µF, 16V electrolytic
C5	-	22µF, 16V electrolytic
C7	-	100µF, 16V electrolytic
C10	-	0.47µF, 63V electrolytic
C12	-	220µF, 25V electrolytic
C13	-	100µF, 10V electrolytic
Miscellaneous:		
S1-S9	-	Push-to-on switch
S10-S12	-	On/off switch
LS1	-	8-ohm, 0.5W speaker
	-	Condenser microphone

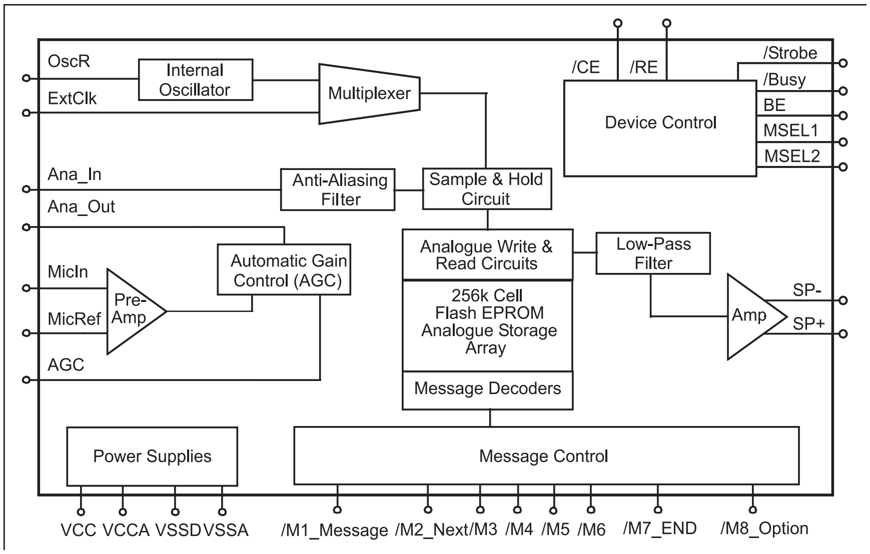


Fig. 1: Functional block diagram of IC APR9600

TABLE I
Modes Selection

Mode	MSEL1	MSEL2	/M8_Option
Random-access, 2 fixed-duration messages	0	1	Pull this pin to Vcc through 100k resistor
Random-access, 4 fixed-duration messages	1	0	Pull this pin to Vcc through 100k resistor
Random-access, 8 fixed-duration messages	1	1	Becomes the /M8 message trigger input pin
Tape-mode, normal operation	0	0	0
Tape-mode, auto-rewind operation	0	0	1

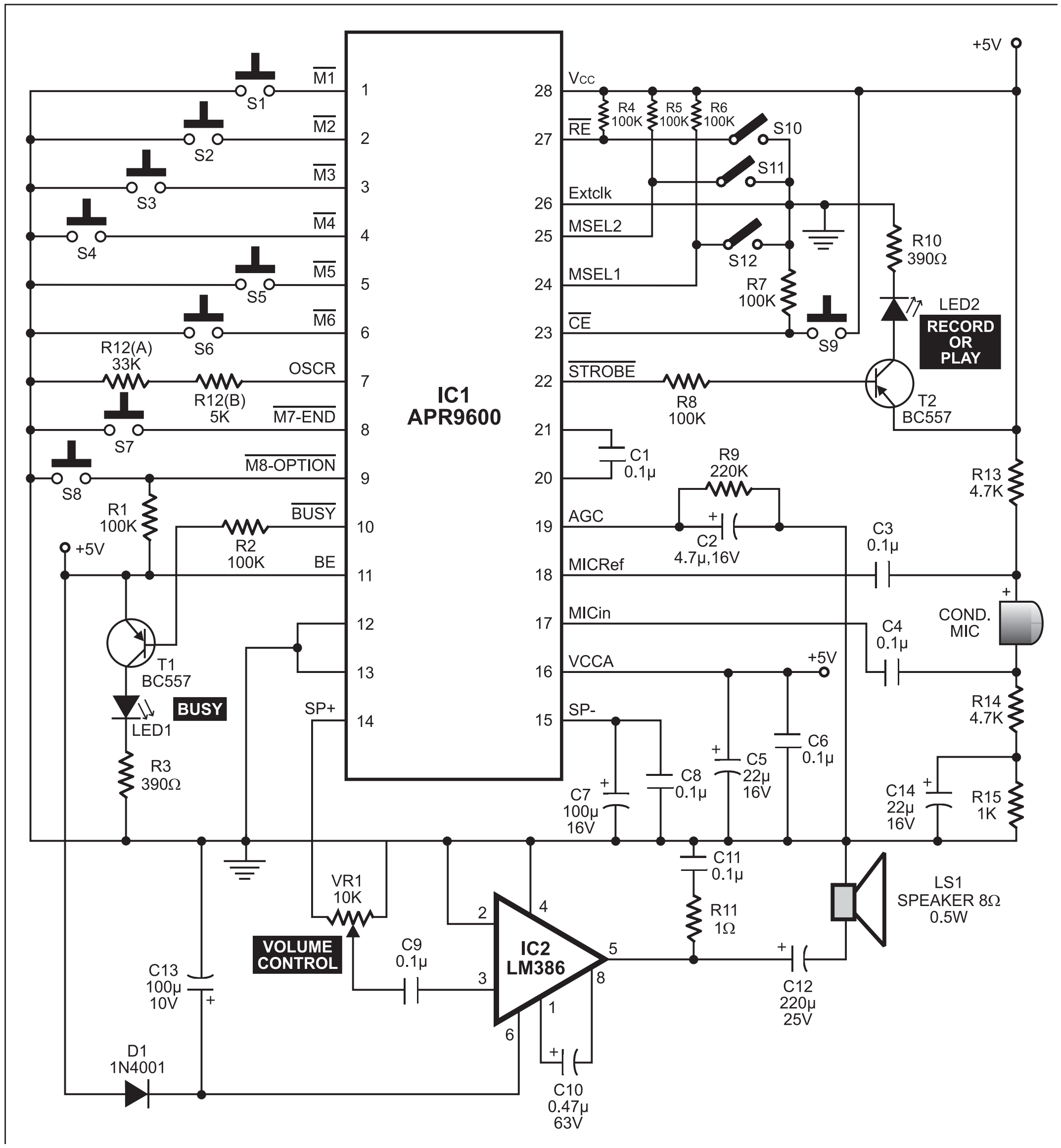


Fig. 2: Random-access mode configuration

differential amplifier. It is further amplified by connecting Ana_Out (pin 21) to Ana_In (pin 20) via an external DC blocking capacitor C1. A bias signal is applied to the microphone and to save power during playback, the ground return of this bias network can be connected to the normally open side of the record switch. Both Mic.$_{in}$ and Mic.$_{Ref}$ (pins 18 and 19) must be coupled to the microphone network through capacitors.

Recording signal from the external source can also be fed directly into the chip using Ana_In (pin 20), but the connection between Ana_In (pin 20) and Ana_out (pin 21) is still required for playback. An internal anti-aliasing filter automatically adjusts its response according to the sampling frequency selected. Then the signal is processed into the memory array through a combination of the sample-and-hold circuit and analogue read/write circuit. The incoming voice signals are sampled and the instantaneous voltage samples are stored in the non-volatile flash memory cells in 8-bit binary encoded format.

During playback, the stored signals are retrieved from the memory, smoothed to form a continuous signal, low-pass filtered and then amplified. The signal level at the speaker terminals SP+ and SP- (pins 14 and 15, respectively) is at about 12mW power into 16-ohm impedance. The output from pin 14 (SP+) is further amplified by the low-power amplifier using LM386 (IC2) as shown in the figure. The recorded message is reproduced into speaker LS1.

TABLE II
Reference Rosc Values and Corresponding Sampling Frequencies

Ref Rosc	Sampling frequency	Input bandwidth	Duration
84k	4.2 kHz	2.1 kHz	60 sec
38k	6.4 kHz	3.2 kHz	40 sec
24k	8.0 kHz	4.0 kHz	32 sec

Note. *Rosc table above is for reference only, different lots of ICs will have somewhat different Rosc value performance*

An internal oscillator provides sampling clock to the APR9600. The frequency of the oscillator and sampling rate depend on the value of resistor R12 [R12(A)+R12(B)] connected across OSCR (pin 7) of the chip and the ground.

Table II shows the sampling frequencies corresponding to different resistor values, as well as the resulting input bandwidth and duration of recording. Higher sampling rates improve the voice quality but they also increase the bandwidth requirement and thus reduce the duration. Lower sampling rates use fewer memory cells and effectively increase the recording/playback duration of the device. The RC network (comprising resistor R9 and capacitor C2 connected) at pin 19 sets the AGC attack time. (The attack time is defined as the delay present before the AGC circuit begins to adjust gain.)

Message management. The APR9600 chip supports the following message modes:

1. Random-access mode with 2, 4 or 8 messages within the total recording time.
2. Tape mode with two options: auto rewind and normal operation.

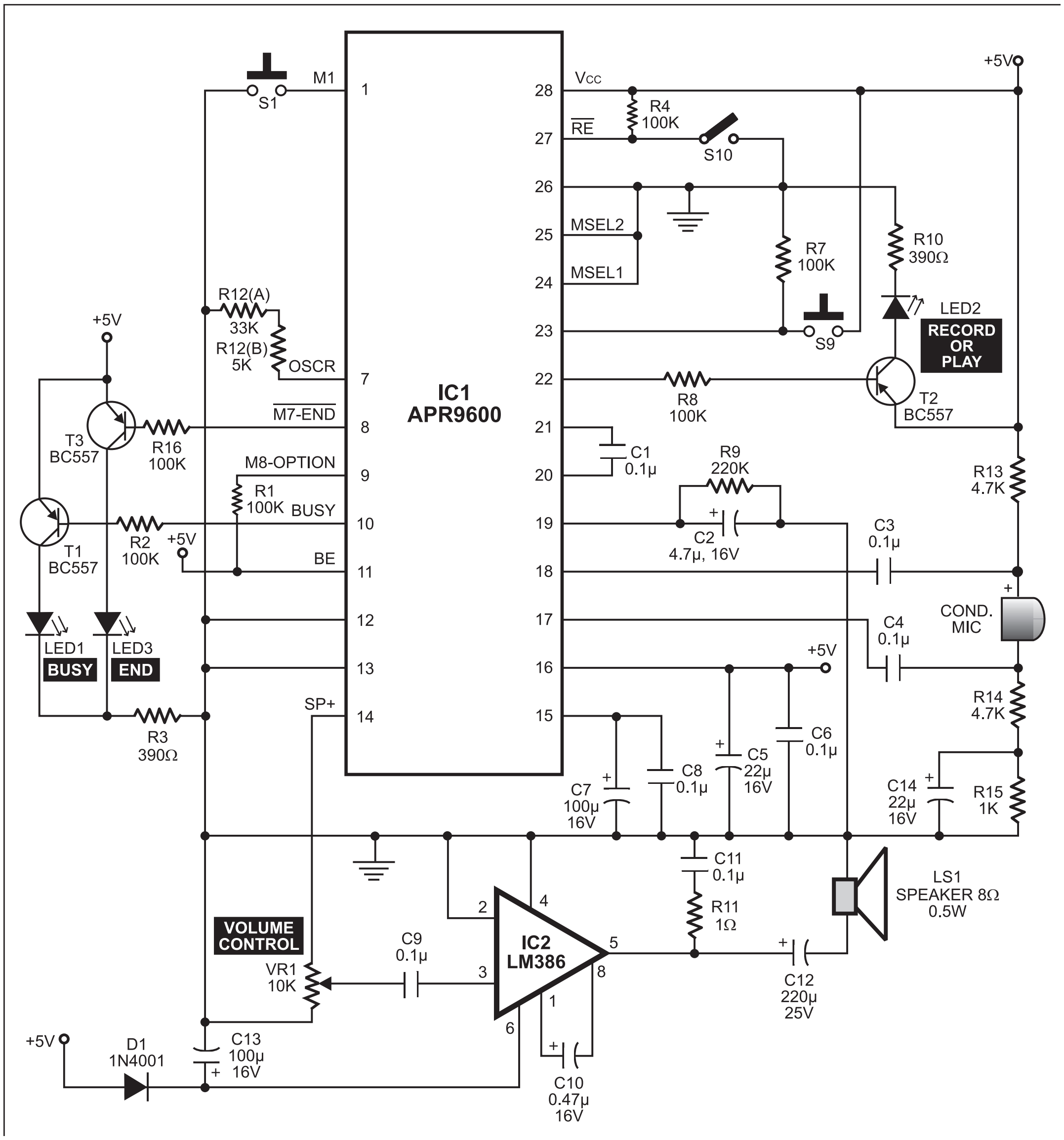


Fig. 3: Circuit for recording/playback in tape mode with auto-rewind option

The modes are defined by pins 24 (MSEL1), 25 (MSEL2) and 9 (/M8_Option) as shown in Table I, and cannot be mixed.

An important feature of the APR9600 chip is indication of changes in the device status through beeps superimposed on the device output; for example, the start of recording is indicated by a beep, so the person can now start speaking into the microphone. This feature is enabled by making pin 11 (BE) high.

General functional description. On power up, pin 23 (CE) is pulled low through resistor R7 to enable the device for operation. Toggling this pin by switch S9 also resets several message management features. Pin 27 (RE) is pulled low to enable recording and it is pulled high for playback. To start recording/playback, switch the appropriate trigger pin as described later. Glowing of LED1 indicates that the device is busy and no commands can be currently accepted. The LED is driven by pnp transistor T1, which is connected to pin 10 (Busy) of the chip. LED2 indicates recording in each individual memory segment. It is driven by pin 22 (strobe) through transistor T2.

Random-access mode

As mentioned earlier, the random-access mode supports 2, 4 or 8 messages of fixed durations. It allows easy indexing of messages as they can be recorded or played randomly. The length of each message is the total recording length available (as defined by the selected sampling rate) divided by the total number of memory segments/tracks enabled (as per Table I).

Recording of sound. The circuit for recording/playback of eight fixed-duration messages in random-access mode is shown in Fig. 2. Pins 9 (M8_Option), 24 (MSEL1) and 25 (MSEL2) are pulled high through resistors R1, R6 and R5, respectively. When switch S10 is closed, record pin 27 (RE) goes low to enable recording of the message from the microphone. The maximum length of the eight sound tracks is 7.5 seconds each.

Now to start recording the first message, press switch S1 and hold it in this position. A beep sound is heard and LED2 blinks. You can now speak into the condenser mic. The recording will terminate if switch S1 is released or if the recording time exceeds 7.5 seconds. Similarly, press switches S2 through S8 to record other sound tracks. For recording of two or four sound tracks of fixed duration, the status of pins 9, 24 and 25 is as per Table I.

Playback of sound tracks. Open switch S10 to make pin 27 high while keeping other switches in the same positions as in recording. Toggling switches S1 through S8 causes playback of particular sound tracks. Pressing the same switch again or switch S9 terminates playback of the sound track. Pressing any other switch (S1 through S8) while a sound track is being played causes a new sound track to be played. If the switch is held pressed, the particular sound track will play continuously.

Tape mode

The tape mode operation is much like the conventional cassette tape

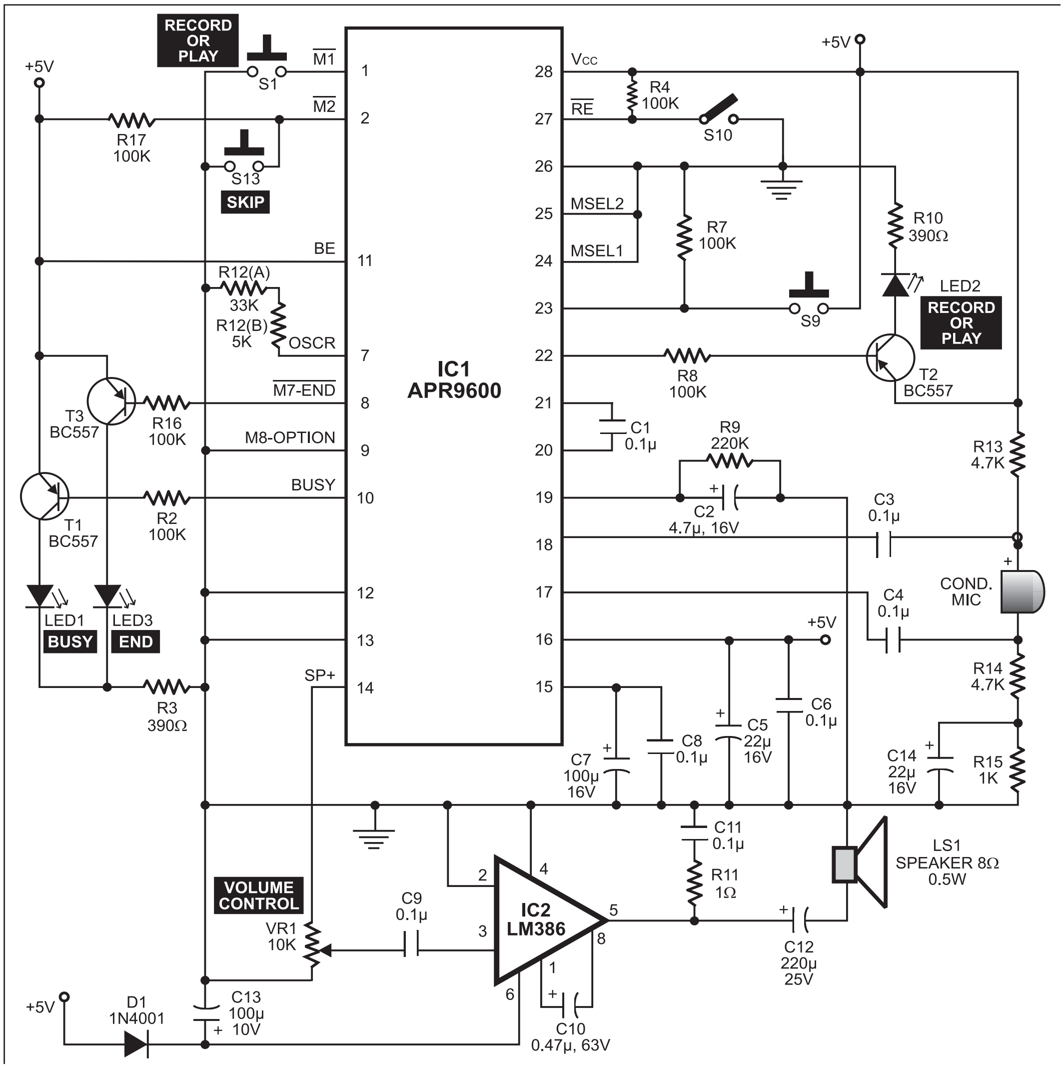


Fig. 4: Circuit for recording/playback in tape mode with normal option

recorder, but with auto-rewind and normal operation options. In auto-rewind mode, the device automatically rewinds to the beginning of the message immediately after recording or playing the message. In normal mode, it must be switched for rewind.

Sound recording in tape mode with auto-rewind option. Fig. 3 shows the circuit for recording/playback in tape mode with auto-rewind option. In this configuration, pins 24 (MSEL1) and 25 (MSEL2) are connected to ground, whereas pin 9 is pulled high through resistor R1. Close switch S10 to enable the recording of message. Press switch S9 to reset the sound track counter to zero. Now press switch S1 and hold it in this position. A beep sound is heard and LED2 starts blinking. This means you can speak into the mic. Recording will terminate when switch S1 is released or if the recording time exceeds 60 seconds. Press switch S1 again and again to record second, third, fourth and other consecutive sound tracks. Each sound track may have a different length but the total length of all sound tracks cannot exceed 60 seconds. When LED3 lights up during recording, it indicates the end of memory array.

Playback in tape mode with auto-rewind option. Open switch S10 to pull pin 27 high while keeping other switches in the same positions as applicable during recording. Toggle switch S1 repeatedly to play consecutive sound tracks. Press switch S9 to reset the sound track counter to zero. During playback, LED3 indicates that all recorded messages have been played.

Recording in tape mode with normal option. Fig. 4 shows the circuit for recorcding/playback in tape mode with normal option. Connect pins 24 (MSEL1), 25 (MSEL2) and 9 (M8_ option) to ground. Close

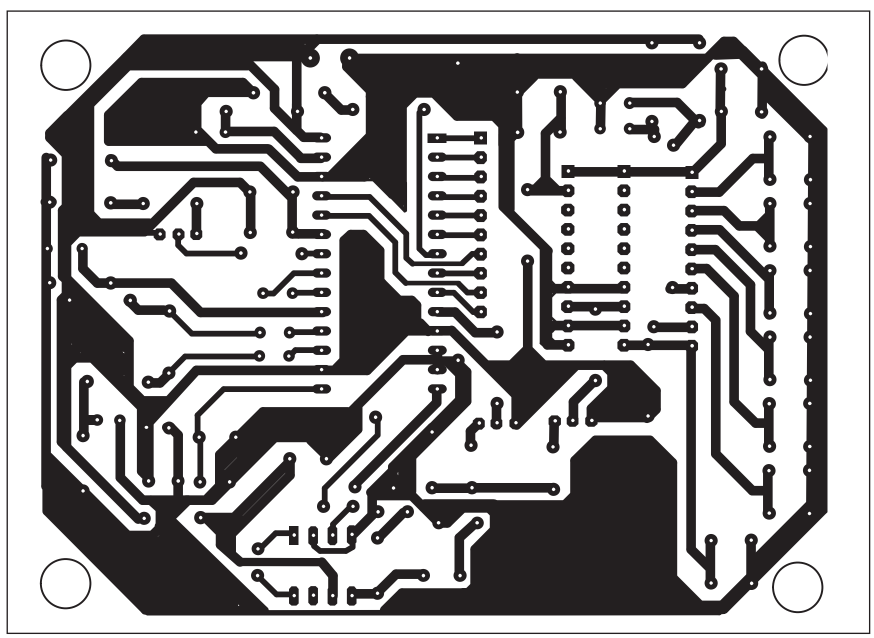


Fig. 5: Combined actual-size, single-side PCB for circuits of Figs 2, 3 and 4

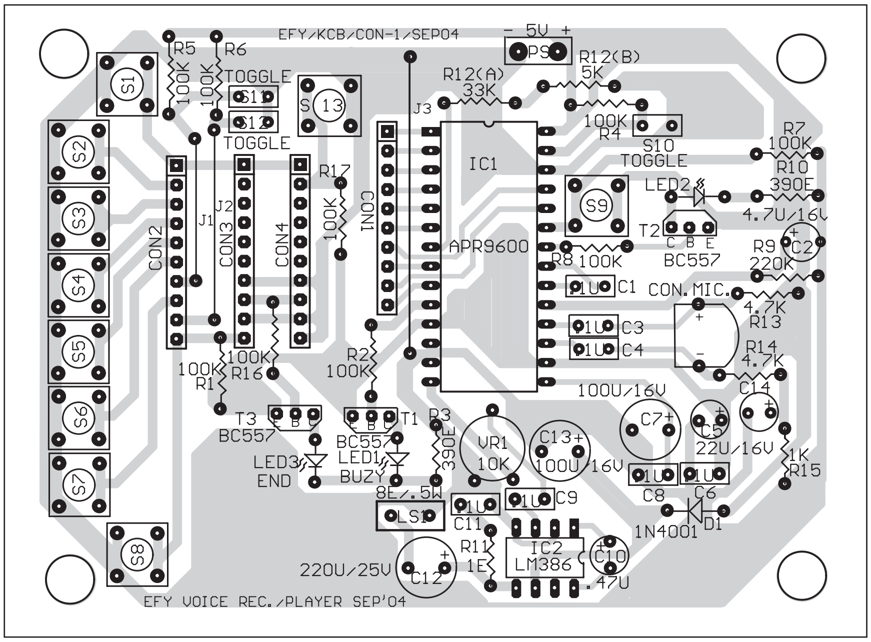


Fig. 6: Component layout for the PCB

switch S10 to enable the recording of message. Press switch S9 to reset the sound track counter to zero. The normal-mode recording is similar to the rewind-mode recording, except that after swich S1 is released, the sound counter doesn't increment itself to the next sound track location. To record the first sound track, press switch S1 and hold it in this position. A beep sounds and LED2 blinks. Now you can speak into the microphone. To record the next message, release switch S1 and toggle switch S13. Now press switch S1 again and hold in this position. A beep sounds and LED2 blinks. This means you can speak into the microphone to record the message. In case you press switch S1 without toggling switch S13 to record the message, the message will be recorded at the location of the first message.

Playback in tape mode with normal option. Open switch S10 to pull pin 27 high while keeping other switches in the same positions as during recording operation. First, press switch S9 to reset the sound track counter to zero. Now momentarily press switch S1 to play the first sound track. Momentarily pressing of switch S1 again and again will still play the first sound track. The sound track counter can be incremented to play the next sound track by momentarily pressing switch S13.

The combined actual-size, single-side PCB for the circuits of Figs 2, 3 and 4 is shown in Fig. 5 and its component layout in Fig. 6.

To obtain the configuration of Fig. 2, connect connector Con1 to Con2 using burgstick connectors with ribbon cable or simply using jumper wires. Similarly, configuration of Fig. 3 or Fig. 4 can be realised by connecting Con1 to Con3 or Con4. Note that switch S1 is common for all configurations. ❑

DYNAMIC TEMPERATURE INDICATOR AND CONTROLLER

NIRANJANA ASHOK AND SREEJA MENON

Here's a standalone digital thermometer that also controls the temperature of the heating element of a device according to its requirement. Use of embedded technology makes this closed-loop feedback control system efficient and reliable. Microcontroller (PIC16F73) allows dynamic and faster control. A temperature-controller knob and liquid crystal display (LCD) make the system user-friendly. The sensed and set temperature values are simultaneously displayed on the LCD panel in Kelvin scale.

The circuit is programmed for 'on'/'off' control. It is very compact using few components and can be implemented for several applications including air-conditioners, water-heaters, snow-melters, ovens, heat-exchangers, mixers, furnaces, incubators, thermal baths and veterinary operating tables.

PIC16F73 microcontroller is the heart of the circuit as it controls all the functions. Fig. 1 shows the pin configuration of PIC16F73 microcontroller.

The circuit

Fig. 2 shows the functional block diagram of the PIC16F73-based dynamic temperature controller. The temperature transducer (AD590) senses the temperature and converts it into an electrical signal, which is applied to the microcontroller. The analogue signal is converted into digital format by the inbuilt analogue-to-digital converter (ADC) of the microcontroller. The sensed and set values of the temperature are displayed on the 16x2-line LCD. The microcontroller drives a transistor to control the heating element with the help of an electromagnetic relay.

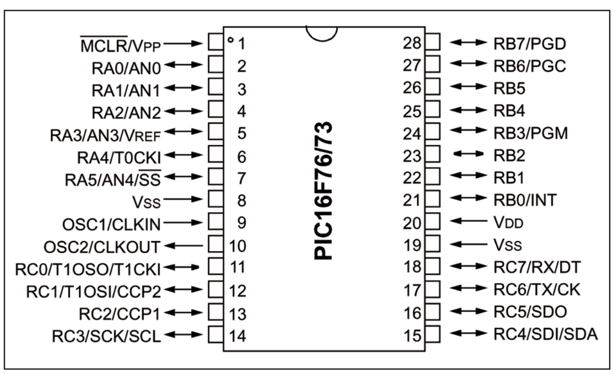


Fig. 1: Pin configuration of PIC16F73 microcontroller

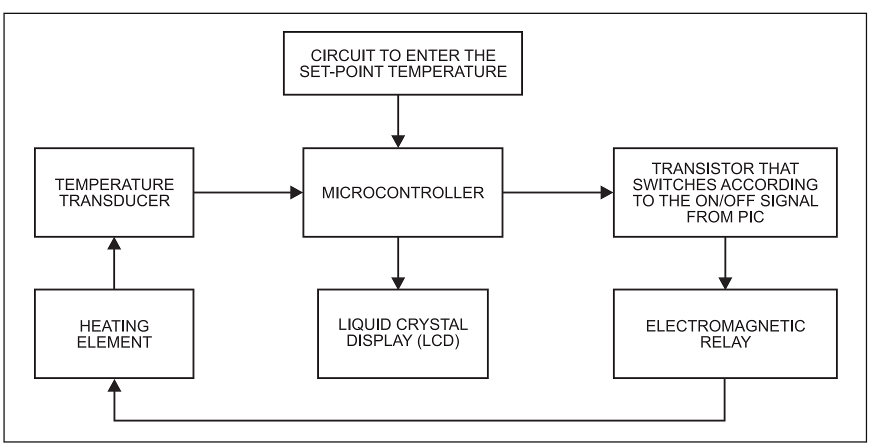


Fig. 2: Block diagram of the PIC16F73-based dynamic temperature controller

PARTS LIST

Semiconductors:	
IC1	- 7812, 12V regulator
IC2	- 7805, 5V regulator
IC3	- PIC16F73 microcontroller
T1	- SL100 npn transistor
D1-D5	- 1N4007 rectifier diode
AD590	- Temperature sensor
LED1	- Red LED
LED2	- Green LED
	- 16×2-line LCD
Resistors (all ¼-watt, ±5% carbon, unless stated otherwise):	
R1, R8	- 1-kilo-ohm
R2, R3, R6, R7	- 10-kilo-ohm
R4	- 3.9-kilo-ohm
R5	- 15-kilo-ohm
VR1	- 10-kilo-ohm potmeter
VR2	- 10-kilo-ohm preset
Capacitors:	
C1	- 1000µF, 35V electrolytic
C2	- 0.33µF ceramic
C3-C6	- 0.33µF ceramic
C7	- 100µF, 100V electrolytic
Miscellaneous:	
X1	- 230V AC primary to 7.5V-0-7.5V, 250mA secondary transformer
X_{TAL}	- 5MHz crystal
RL1	- 12V, 200-ohm, 1 C/O relay
S1	- On/off switch

PIC16F73 is an 8-bit, low-cost, high-performance flash microcontroller. Its key features are 4k words of flash program memory, 192 bytes of data RAM, eleven interrupts, three I/O ports, 8-bit ADC and only 35 powerful single-cycle instructions (each 14-bit wide). The ADC simplifies the overall embedded system design by providing a direct interface for temperature, pressure, motion and other sensors. The set temperature value can be varied from 253°K to 430°K using an external knob on the front panel of the cabinet.

Fig. 3 shows the circuit of PIC16F73 microcontroller-based dynamic temperature controller. The temperature sensor (AD590) outputs a current of 1 µA/°K.

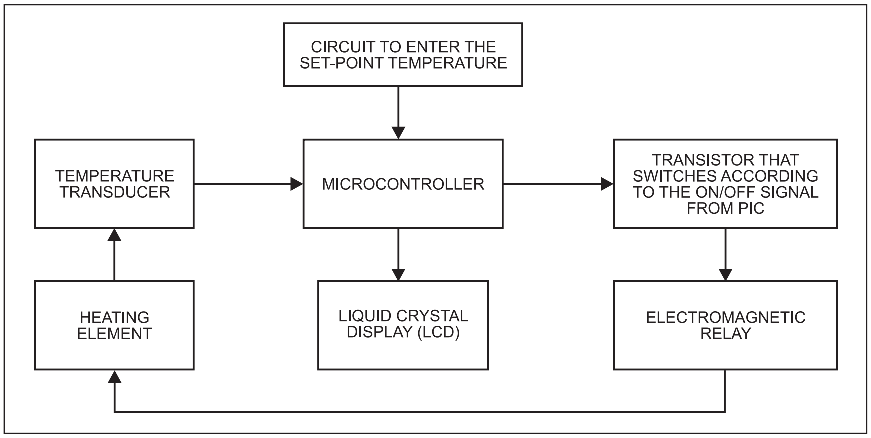


Fig. 2: Block diagram of the PIC16F73-based dynamic temperature controller

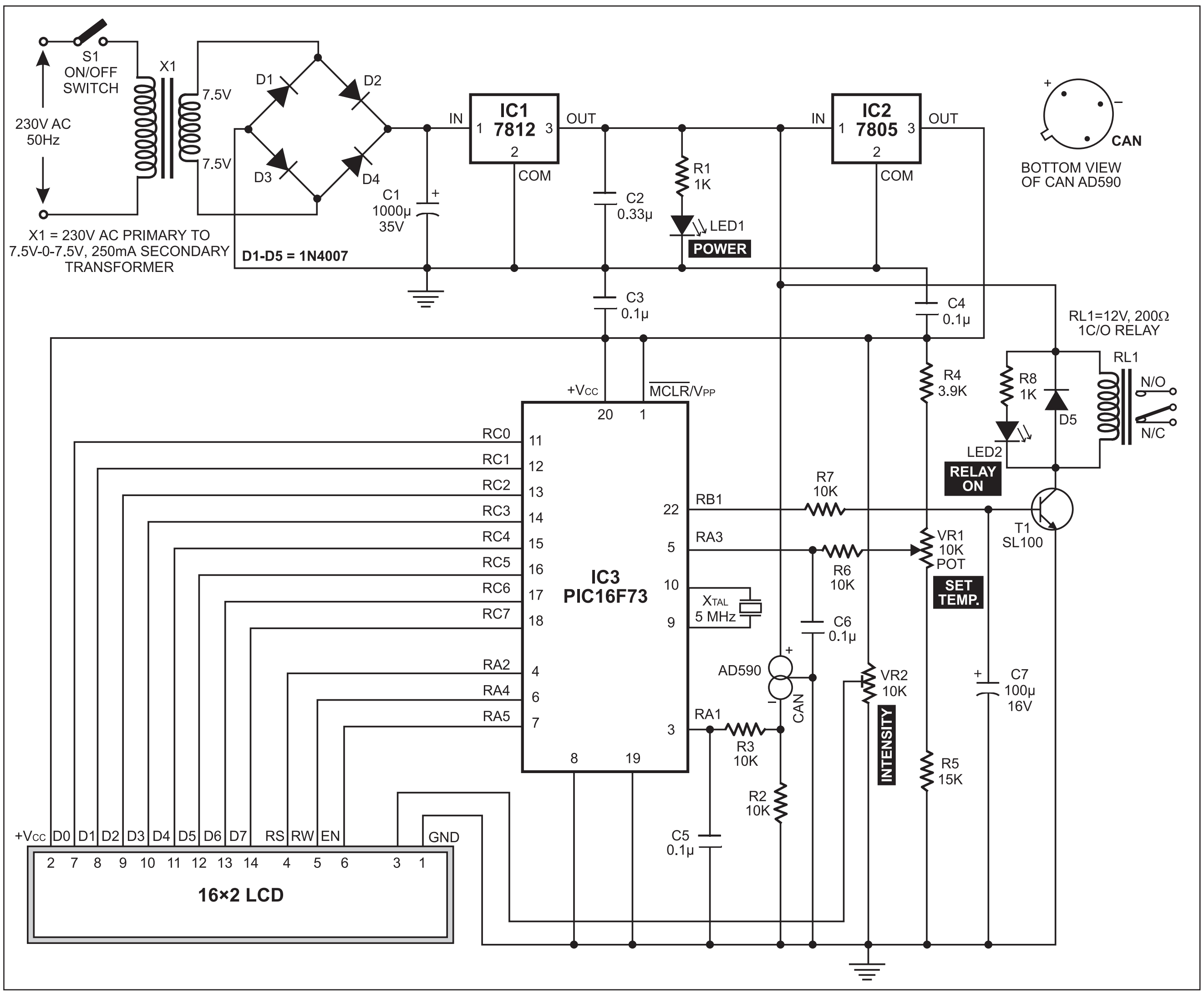


Fig. 3: Circuit of PIC16F73 microcontroller-based dynamic temperature controller

A high-impedance constant current is delivered for a supply voltage between 4V and 30V. The sensing range is linear from 218°K (–55°C) to 423°K (+150°C). A 10-kilo-ohm resistor is used to convert the current from the sensor into voltage with a sensitivity of 1V/°K (1 µA/°K×1000). Hence, the voltage range is 2.18V to 4.23V. This voltage is fed to pin 3 (RA1) of the microcontroller.

PIC16F73 microcontroller is a 28-pin IC with three input/output ports: port A (RA0 through RA5), port B (RB0 through RB7) and port C (RC0 through RC7).

Port-A pins 3 (RA1) and 5 (RA3) are programmed as analogue inputs. The inbuilt 8-bit ADC converts the analogue input signal into 8-bit digital equivalent output. Its analogue reference voltage is software-selectable to either the positive supply voltage of the device (Vcc) or the voltage level of RA3 pin. Here, Vcc (5V) is selected as the analogue reference voltage.

Pins 3 (RA1) and 5 (RA3) are programmed to sense the analogue voltages corresponding to the sensed and set temperature values, respectively. The voltage corresponding to the set temperature is obtained by means of a potential divider network comprising a potentiometer (VR1) and two fixed resistors (R4 and R5). The variable terminal of the potentiometer is connected to pin 5 (RA3) of the microcontroller and the shaft is rotated by the user to vary the set-point temperature that is visible on the LCD.

The microcontroller has been programmed to sense the analogue voltages corresponding to the sensed and set temperature values. The sensed voltages are manipulated such that the corresponding temperature values are displayed on the LCD by sending out the corresponding data signals through pins 11 through 18 (RC0 through RC7) and control signals through pins 4, 6 and 7 (RA2, RA4 and RA5) of the microcontroller. Then the sensed temperature value is compared with the set-point temperature value. Pin 22 (RB1) of the microcontroller goes high if the set-point temperature is higher than the sensed temperature. This pin has been programmed as an output to control the relay through transistor T1. The relay contacts are connected to the heating element.

Data is sent to the LCD's data pins 7 through 14. Control signals required before each data transmission are sent to pins 4, 5 and 6 (RS, R/W and Enable) of the LCD.

Working of the circuit

The mains supply is stepped down by transformer X1 to deliver a secondary output of 7.5V-0-7.5V AC, 250 mA. The transformer output is rectified by a full-wave rectifier comprising diodes D1 through D4 and filtered by capacitor C1. ICs 7812 (IC1) and 7805 (IC2) provide regulated 12V and 5V power supplies. Capacitors C2 and C4 bypass any ripple in the regulated outputs. LED1 gives power-'on' indication when current flows through resistor R1.

The 12V regulated supply is used for driving the temperature sensor (AD590). AD590 has three terminals, namely, '+', '–' and 'CAN.' The '+' terminal is connected to the 12V power supply and the 'CAN' terminal is grounded. The current output obtained from the '–' terminal is converted into voltage using resistor R2 (10 kilo-ohms). This voltage is applied to pin 3 (RA1) of the microcontroller.

The potential divider network comprising resistor R4 (4-kilo-ohm), potentiometer VR1 (10-kilo-ohm) and resistor R5 (15-kilo-ohm) is connected across regulated 5V supply. The variable terminal of potentiometer VR1 is connected to pin 5 of the microcontroller.

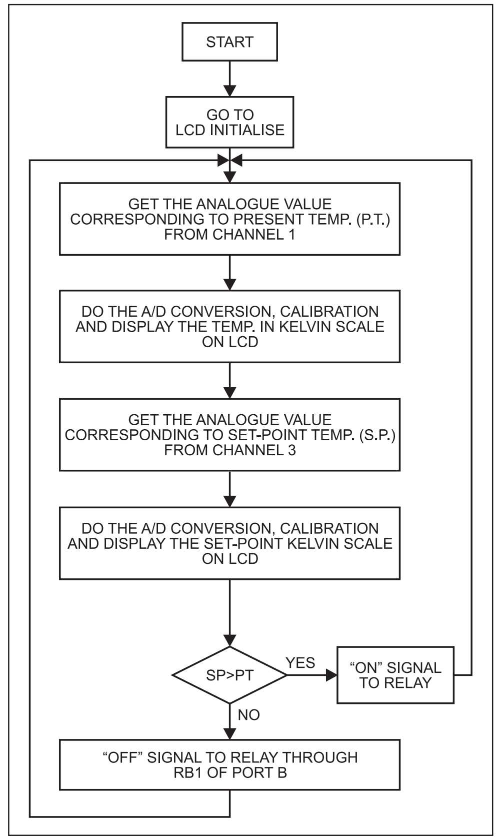


Fig. 4: Flow-chart of the program

Capacitors C5 through C7 filter out the noise. A 5MHz crystal (X_{TAL}) connected between pins 9 and 10 of the microcontroller provides clock frequency.

Register-select pin 4, R/W pin 5 and Enable pin 6 of the LCD are connected to pins 4, 6 and 7 of the microcontroller, respectively, and data pins 7 through 14 are connected to pins 11 through 18, respectively. Pin 3 of the LCD is used to control the contrast by using preset VR2.

The relay is connected between +12V and the collector of transistor T1. When pin 22 of the microcontroller is high, transistor T1 saturates and the relay energises to

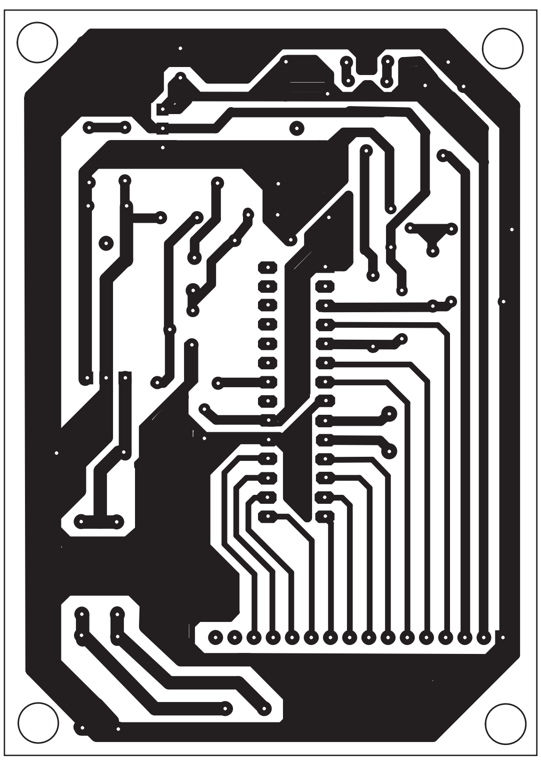


Fig. 5: Actual-size, single-side PCB layout for temperature indicator using PIC16F73

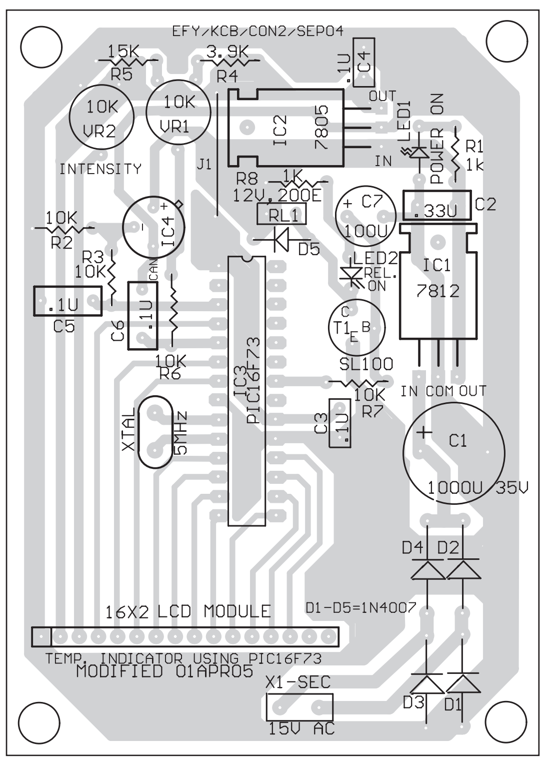


Fig. 6: Component layout for the PCB

switch the device 'on.' When pin 22 is low, transistor T1 cuts off and the relay de-energises to turn the device 'off.' Diode D5 is used here as a free-wheeling diode.

Programming the microcontroller

An El Cheapo programmer circuit (available on 'www.myke.com elcheapo .htm') has been used to program the micro-controller. The program is written as '.asm' file and assembled using MPLAB IDE for generating '.hex' file. The MPLAB IDE assembler can be downloaded from the Website of Microchip (www.microchip.com), the manufacturer of the PIC microc-ontroller. (***Note.*** The datash-eet of PIC16F73 and other relevant files have been included in CD.) The simulator gives the hex code of the program, which is then burnt into the microcontroller using the parallel port of the computer interfaced to the PIC programmer. The programmed micr-ocontroller is then placed in the PCB.

Fig. 4 shows the flow-chart of the program. The microco-ntroller is programmed to give various functional commands with delays for proper initialisation of the LCD. The control signals for the LCD are given from Port A (RA1) of the microcontroller. The analogue voltage corresponding to the sensed temperature given to Port A (RA1) is converted into a digital value and stored in the micro-controller. A binary value of '255' corresponds to 5V (500 kilo-ohms). Based on this relation, calibration is done to extract the digits of the sensed temperature value in degree Kelvin. These digits are then sent from Port C of the microcontroller to the data lines of the LCD.

Similarly, the set temperature is displayed on the LCD. The difference between the sensed and set temperature values is calculated and accordingly RB1 pin of Port B goes high or low to control the relay.

Construction and testing

An actual-size, single-side PCB layout for the dynamic temperature controller using PIC16F73 is shown in Fig. 5 and its component layout in Fig. 6. After making the PCB, check whether all the tracks are as per the circuit diagram. If the tracks are correct, solder the components to the board. Place AD590 close to the soldering iron. Now switch on the power supply and check voltages at various points before placing the microcontroller into the circuit. Taking into consideration the sizes of the various components and the way they have been placed, select the dimensions of the cabinet for the device. Put the entire circuit inside the cabinet and test the working of the circuit.

When burning the program into the microcontroller, use power supply with a proper current limiter to prevent damage to the parallel port of the computer as well as the microcontroller. The analogue voltage to the microcontroller should not be given directly from the power supply, as occasional spikes in the power supply may damage the microcontroller. Instead, you can provide the analogue voltage by means of a potentiometer connected across the required voltage.

Fluctuations visible on the LCD, especially when the sensed temperature value equals the set temperature value, can be eliminated by connecting capacitors between the supply and the ground to bypass the AC interference. Make sure that a pin configured as output is not given an input signal by chance.

Note. In EFY Lab, we used soldering iron as the heating element. The device was modeled to give an 'on'/'off' signal corresponding to the sensed and set-point temperature. When the sensed temperature was below the set temperature, the soldering iron got switched 'on,' and when the sensed temperature crossed the set temperature value, the soldering iron got switched off.

TEMP.LST

```
LOC      OBJECT CODE      LINE SOURCE TEXT      VALUE

                    00001   LIST P=16F73
                    00002   INCLUDE "p16f73.inc"
                    00001   LIST
                    00002   ; P16F73.INC Standard Header File, Version 1.00
                            Microchip Technology, Inc.
                    00320   LIST
2007     3FF2       00003 __CONFIG _HS_OSC & _WDT_OFF & _PWRTE_ON
 00000020           00004   BANK0RAM EQU H'20'
                    00005   CBLOCK BANK0RAM
  00000020          00006   AD1
  00000021          00007   ADUSER
  00000022          00008   TIME1
  00000023          00009   TIME2
  00000024          00010   TEMP
  00000025          00011   FIN
  00000026          00012   CONFU
  00000027          00013   A
  00000028          00014   B3
  00000029          00015   C3
  0000002A          00016   REM
  0000002B          00017   COUNT1
  0000002C          00018   COUNT
  0000002D          00019   COUNT0
  0000002E          00020   COUNTER
  0000002F          00021   COUNTER1
  00000030          00022   A1
  00000031          00023   B1
  00000032          00024   C1
  00000033          00025   C2
  00000034          00026   B2
  00000035          00027   A2
  00000036          00028   RANGE
                            00029     ENDC
 00000002          00030    RS   EQU   H'02'
 00000005          00031    E    EQU   H'05'
 00000004          00032    RW   EQU   H'04'
00000000            00033   RC0  EQU  H'00'
0000                00034  ORG 0X000
0000     2805       00035   GOTO MAINLINE
0004                00036   ORG 0X004
0004     29CA       00037   GOTO INT
0005                00038   MAINLINE
0005     1683       00039   BSF STATUS,RP0
0006     300A       00040   MOVLW B'00001010' ;to set ra1(present)& ra3(user-defined) as
                    00041                     ;i/p pins&ra2,ra4,ra5 as o/p control to lcd
0007     0085       00042   MOVWF TRISA
0008     0187       00043   CLRF TRISC        ;PORTC AS OUTPUT DATA PORT TO LCD
0009     0186       00044   CLRF TRISB
000A     3004       00045   MOVLW B'00000100'     ;to set analog i/p(ra1&ra3),vref(Vcc)
                    00046                          ;dig i/p(ra3,ra4&ra5)
000B     009F       00047   MOVWF ADCON1
000C     1283       00048   BCF STATUS,RP0        ;to go to bank0
000D                00049   INITIALIZE            ;*** LCD Initialization
000D     01AD       00050   CLRF COUNT0           ;all these are delay loops
000E     01AE       00051   CLRF  COUNTER
000F     201C       00052   CALL   MSCOUNTER
0010     2026       00053   CALL T48US
0011     203A       00054   CALL FUNCTIONSET
0012     202C       00055   CALL CHARACTER_ENTRY
0013     2033       00056   CALL  CLEAR_DISPLAY
0014     2041       00057   CALL DISPLAYON
0015                00058   BEGIN                      ;beginning the proper function
0015     2048       00059   CALL  DISPLAY
0016     2068       00060   CALL  PRESENT
0017     206D       00061   CALL  CHECK
0018     1803       00062           BTFSC  STATUS,0
0019     207E       00063   CALL  HUNDREDS
001A     2089       00064   CALL  TENS
001B     2895       00065   GOTO  SUBTRACT
                    00066
                    00067
001C                00068   MSCOUNTER             ;*** 150ms counter before LCD initializaion
001C     302C       00069   MOVLW   D'300'
001D     00A2       00070   MOVWF   TIME1
001E     307C       00071   LOOP1   MOVLW   D'124'
001F     00A3       00072   MOVWF   TIME2
0020     0000       00073   LOOP2   NOP
0021     0BA3       00074           DECFSZ      TIME2,F
0022     2820       00075           GOTO        LOOP2
0023     0BA2       00076           DECFSZ      TIME1,F
0024     281E       00077           GOTO        LOOP1
0025     0008       00078   RETURN
0026                00079   T48US                 ;*** 48us Delay Loop for LCD initialization
0026     3090       00080           MOVLW   D'400'
0027     00AD       00081   MOVWF   COUNT0
0028                00082   T_LOOP
0028     2062       00083   CALL    T12US
0029     0BAD       00084   DECFSZ  COUNT0,F
```

```
002A    2828    GOTO T_LOOP
002B    0008    RETURN
002C            CHARACTER_ENTRY    ;*** Character Entry Command for LCD
002C    1105    BCF PORTA,RS
002D    1205    BCF PORTA,RW
002E    3006    MOVLW H'06'
002F    0087    MOVWF PORTC
0030    2055    CALL PULSE_E
0031    2026    CALL T48US
0032    0008    RETURN
0033            CLEAR_DISPLAY      ;*** Clear Display Command for LCD
0033    1105    BCF PORTA,RS
0034    1205    BCF PORTA,RW
0035    3001    MOVLW H'1'
0036    0087    MOVWF PORTC
0037    2055    CALL PULSE_E
0038    2026    CALL T48US
0039    0008    RETURN
003A            FUNCTIONSET        ;*** Function Set Command for LCD
003A    1105    BCF PORTA,RS
003B    1205    BCF PORTA,RW
003C    3038    MOVLW  H'38'
003D    0087    MOVWF PORTC
003E    2055    CALL  PULSE_E
003F    2026    CALL  T48US
0040    0008    RETURN
0041            DISPLAYON     ;*** Display On/Off & Cursor Command for LCD
0041    1105    BCF PORTA,RS
0042    1205    BCF PORTA,RW
0043    300C    MOVLW  D'12'
0044    0087    MOVWF PORTC
0045    2055    CALL  PULSE_E
0046    2026    CALL  T48US
0047    0008    RETURN
0048            DISPLAY
0048    3008    MOVLW D'8'
0049    00AE    MOVWF COUNTER
004A            MESSAGE
004A    082E    MOVF COUNTER,W
004B    3C08    SUBLW D'8'            ;Subtract character count from 19
                                      ;& store result in W
004C    2059    CALL TEXT
004D    1505    BSF PORTA,RS          ;RS  line to 1 to i/p Data
004E    1205    BCF  PORTA,RW         ;R/W line to 0 to write
004F    0087    MOVWF PORTC           ;send character to LCD
0050    2055    CALL  PULSE_E         ;Clock the LCD
0051    2026    CALL T48US            ;delay for LCD busy
0052    0BAE    DECFSZ COUNTER,F      ;counter - 1 = 0 ?
                                      ;Are all characters displayed ?
0053    284A    GOTO MESSAGE          ;No.  Display the next character
0054    0008    RETURN                  ;Yes. Goto Initialize
0055            PULSE_E        ;*** Display On/Off & Cursor Command for LCD
0055    1685    BSF PORTA,E
0056    0000    NOP
0057    1285    BCF PORTA,E
0058    3400    RETLW H'0'
0059            TEXT                  ;*** Initialization Display Data for LCD
0059    0782    ADDWF 02,F            ;Store (PC+W) in PC(addr $02) to jump
                                      ;forward
005A    3454    RETLW    H'54'        ;ascii for t
005B    3465    RETLW    H'65'        ;ASCII for e
005C    346D    RETLW    H'6d'        ;ASCII for m
005D    3470    RETLW    H'70'        ;ASCII for p
005E    34A5    RETLW    H'a5'        ;ASCII for .
005F    34FE    RETLW    H'fe'        ;ASCII for blank
0060    343A    RETLW    H'3a'        ;ASCII for :
0061    34FE    RETLW    H'fe'        2;ASCII for blank
0062            T12US                 ;***  12 microseconds timer ***
0062    2863            GOTO   $+1
0063    2864            GOTO   $+1
0064    2865            GOTO   $+1
0065    2866            GOTO   $+1
0066    0000            NOP
0067    0008            RETURN
0068            PRESENT
0068    3049    MOVLW B'01001001'     ;set clk 2 fosc/8,ADON,i/p channel ra1
0069    009F            MOVWF ADCON0
006A    201C            CALL  MSCOUNTER
006B    151F            BSF ADCON0,2          ;set GO bit to start ADC
006C    0008            RETURN
006D            CHECK
006D    191F    BTFSC ADCON0,2      ;when conversion is complete ADCON0 will
                                    ;be cleared and control will come out of loop
006E    286D    GOTO CHECK
006F    081E    MOVF ADRES,W          ;the ADC value is found in ADRES
0070    00A0    MOVWF AD1
0071    3CFF    SUBLW D'255'          ;255-ADRES
0072    00A5    MOVWF FIN
0073    1003    BCF STATUS,0
0074    0D25    RLF FIN,W             ;(255-ADRES)*2
0075    00A6    MOVWF CONFU
0076    3064    MOVLW D'100'          ;2 extract hundreds value
0077    0226    SUBWF CONFU,W
0078    01A7    CLRF A
0079    01A8    CLRF B3
007A    01A9    CLRF C3
007B    01AC    CLRF COUNT
007C    01AB    CLRF COUNT1
007D    0008    RETURN
007E            HUNDREDS
007E    0FAB    INCFSZ COUNT1,F     ;2 count hundreds subtract from 100 until
                                      ;***  12 microseconds timer ***borrow
                                      is generated.
```

```
007F    3064    MOVLW D'100'
0080    02A6    SUBWF CONFU,F
0081    1803    BTFSC STATUS,0
0082    287E    GOTO HUNDREDS
0083    3001    MOVLW D'1'
0084    022B    SUBWF COUNT1,W
0085    00A7    MOVWF A
0086    3064    MOVLW D'100'
0087    07A6    ADDWF CONFU,F   ;the difference obtained should be added with
                                ;100 so as to extract the tens place
0088    0008    RETURN
0089            TENS
0089    0FAC    INCFSZ COUNT,F            ;2 count tens subtract from 10
008A    300A    MOVLW D'10'
008B    02A6    SUBWF CONFU,F
008C    1803    BTFSC STATUS,0
008D    2889    GOTO TENS
008E    300A    MOVLW D'10'
008F    0726    ADDWF CONFU,W
0090    00A9    MOVWF C3          ;ones place-difference+10 gives ones place
0091    3001    MOVLW D'1'
0092    022C    SUBWF COUNT,W
0093    00A8    MOVWF B3                  ;tens place-count gives tens place
0094    0008    RETURN
                                          ;temp in kevin =500 -(255-ADRES)*2
                                          ;******subtraction from 500 is done by
                                          digitwise subtraction
                                          ;various cases are to be considered inorder
                                          to initiate subtraction
0095            SUBTRACT
0095    300A    MOVLW D'10'
0096    00B2    MOVWF C1
0097    3009    MOVLW D'9'
0098    00B1    MOVWF B1
0099    3004    MOVLW D'4'
009A    00B0    MOVWF A1
009B    3000    MOVLW D'0'
009C    0629    XORWF C3,W
009D    1903    BTFSC STATUS,Z
009E    28AD    GOTO UNIT                 ;if ones place is 0,B1 should be made 10
009F    0829    MOVF C3,W
00A0    0232    SUBWF C1,W
00A1    00B3    MOVWF C2
00A2    3000    MOVLW D'0'
00A3    0628    XORWF B3,W
00A4    1903    BTFSC STATUS,Z
00A5    28C0    GOTO TEN                  ;if tens place is 0
00A6    0828    MOVF B3,W
00A7    0231    SUBWF B1,W
00A8    00B4    MOVWF B2
00A9    0827    MOVF A,W
00AA    0230    SUBWF A1,W
00AB    00B5    MOVWF A2
00AC    28C7    GOTO DISPLAY1
00AD            UNIT                      ;*when units place is zero and tens place
                                          is nonzero
00AD    01B3    CLRF C2
00AE    3000    MOVLW D'0'
00AF    0628    XORWF B3,W
00B0    1903    BTFSC STATUS,Z
00B1    28BA    GOTO UNITY                ;in case tens place is 0,A1 should be made 5
00B2    0AB1    INCF B1,F
00B3    0828    MOVF B3,W
00B4    0231    SUBWF B1,W
00B5    00B4    MOVWF B2
00B6    0827    MOVF A,W
00B7    0230    SUBWF A1,W
00B8    00B5    MOVWF A2
00B9    28C7    GOTO DISPLAY1
00BA            UNITY                       ;*when units place and tens place are
                                          both zeroes
00BA    01B4    CLRF B2
00BB    0AB0    INCF A1,F                   ;make A1=5
00BC    0827    MOVF A,W
00BD    0230    SUBWF A1,W
00BE    00B5    MOVWF A2
00BF    28C7    GOTO DISPLAY1
00C0            TEN                       ;*when tens place is zero and units place
                                          is nonzero
00C0    0828    MOVF B3,W
00C1    0231    SUBWF B1,W
00C2    00B4    MOVWF B2
00C3    0827    MOVF A,W
00C4    0230    SUBWF A1,W
00C5    00B5    MOVWF A2
00C6    28C7    GOTO DISPLAY1
00C7            DISPLAY1                    ;*to display the nos. on LCD
00C7    1505    BSF PORTA,RS              ;RS  line to 1 to i/p Data
00C8    1205    BCF PORTA,RW              ;R/W line to 0 to write
00C9    3030    MOVLW D'48'
00CA    07B5    ADDWF A2,F
00CB    0835    MOVF A2,W
00CC    0087    MOVWF PORTC               ;send character to LCD
00CD    2055    CALL PULSE_E              ;Clock the LCD
00CE    2026    CALL T48US                ;delay for LCD busy
00CF    1505    BSF PORTA,RS              ;RS  line to 1 to i/p Data
00D0    1205    BCF PORTA,RW              ;R/W line to 0 to write
00D1    3030    MOVLW D'48'
00D2    07B4    ADDWF B2,F
00D3    0834    MOVF B2,W
00D4    0087    MOVWF PORTC               ;send character to LCD
00D5    2055    CALL PULSE_E              ;Clock the LCD
00D6    2026    CALL T48US
00D7    1505    BSF PORTA,RS              ;RS  line to 1 to i/p Data
00D8    1205    BCF PORTA,RW              ;R/W line to 0 to write
00D9    3030    MOVLW D'48'
00DA    07B3    ADDWF C2,F
00DB    0833    MOVF C2,W
00DC    0087    MOVWF PORTC               ;send character to LCD
```

```
00DD 2055 00297 CALL PULSE_E ;Clock the LCD
00DE 2026 00298 CALL T48US
00DF 1505 00299 BSF PORTA,RS ;RS line to 1 to i/p Data
00E0 1205 00300 BCF PORTA,RW ;R/W line to 0 to write
00E1 30DF 00301 MOVLW B'11011111' ;to print symbol for degree
00E2 0087 00302 MOVWF PORTC ;send character to LCD
00E3 2055 00303 CALL PULSE_E ;Clock the LCD
00E4 2026 00304 CALL T48US
00E5 1505 00305 BSF PORTA,RS ;RS line to 1 to i/p Data
00E6 1205 00306 BCF PORTA,RW ;R/W line to 0 to write
00E7 304B 00307 MOVLW 'K'
00E8 0087 00308 MOVWF PORTC ;send character to LCD
00E9 2055 00309 CALL PULSE_E ;Clock the LCD
00EA 2026 00310 CALL T48US
          00311 ;*to give the set-point value follow the
                same procedure as above user_defined
00EB 3059 00312 MOVLW B'01011001' ;set clk 2 fosc/8,ADON,i/p channel ra3
00EC 009F 00313 MOVWF ADCON0
00ED 201C 00314 CALL MSCOUNTER
00EE 151F 00315 BSF ADCON0,2
00EF      00316 CHECK1
00EF 191F 00317 BTFSC ADCON0,2
00F0 28EF 00318 GOTO CHECK1
00F1 081E 00319 MOVF ADRES,W
00F2 00A1 00320 MOVWF ADUSER
00F3 3CFF 00321 SUBLW D'255'
00F4 00A5 00322 MOVWF FIN
00F5 1003 00323 BCF STATUS,0
00F6 0D25 00324 RLF FIN,W
00F7 00A6 00325 MOVWF CONFU
00F8 3064 00326 MOVLW D'100' ;2 extract hundreds value
00F9 0226 00327 SUBWF CONFU,W
00FA 01A7 00328 CLRF A
00FB 01A8 00329 CLRF B3
00FC 01A9 00330 CLRF C3
00FD 01AC 00331 CLRF COUNT
00FE 01AB 00332 CLRF COUNT1
00FF 1803 00333 BTFSC STATUS,0
0100 2103 00334 CALL HUNDREDS1
0101 210E 00335 CALL TENS1
0102 2919 00336 GOTO SUBTRACT1
0103      00337 HUNDREDS1
0103 0FAB 00338 INCFSZ COUNT1,F ;2 count hundreds
0104 3064 00339 MOVLW D'100'
0105 02A6 00340 SUBWF CONFU,F
0106 1803 00341 BTFSC STATUS,0
0107 2903 00342 GOTO HUNDREDS1
0108 3001 00343 MOVLW D'1'
0109 022B 00344 SUBWF COUNT1,W
010A 00A7 00345 MOVWF A
010B 3064 00346 MOVLW D'100'
010C 07A6 00347 ADDWF CONFU,F
010D 0008 00348 RETURN
010E      00349 TENS1
010E 0FAC 00350 INCFSZ COUNT,F ;2 count tens
010F 300A 00351 MOVLW D'10'
0110 02A6 00352 SUBWF CONFU,F
0111 1803 00353 BTFSC STATUS,0
0112 290E 00354 GOTO TENS1
0113 300A 00355 MOVLW D'10'
0114 0726 00356 ADDWF CONFU,W
0115 00A9 00357 MOVWF C3 ;ones place
0116 3001 00358 MOVLW D'1'
0117 022C 00359 SUBWF COUNT,W
0118 00A8 00360 MOVWF B3
0119      00361 SUBTRACT1
0119 300A 00362 MOVLW D'10'
011A 00B2 00363 MOVWF C1
011B 3009 00364 MOVLW D'9'
011C 00B1 00365 MOVWF B1
011D 3004 00366 MOVLW D'4'
011E 00B0 00367 MOVWF A1
011F 3000 00368 MOVLW D'0'
0120 0629 00369 XORWF C3,W
0121 1903 00370 BTFSC STATUS,Z
0122 2931 00371 GOTO UNIT1 ;in case tens place is 0,B1 should be made 10
0123 0829 00372 MOVF C3,W
0124 0232 00373 SUBWF C1,W
0125 00B3 00374 MOVWF C2
0126 3000 00375 MOVLW D'0'
0127 0628 00376 XORWF B3,W
0128 1903 00377 BTFSC STATUS,Z
0129 2944 00378 GOTO TEN1 ;in case tens place is 0 ,A1 should be made 5
012A 0828 00379 MOVF B3,W
012B 0231 00380 SUBWF B1,W
012C 00B4 00381 MOVWF B2
012D 0827 00382 MOVF A,W
012E 0230 00383 SUBWF A1,W
012F 00B5 00384 MOVWF A2
0130 294B 00385 GOTO DISPLAY2
0131      00386 UNIT1 ;*when units place is zero
0131 01B3 00387 CLRF C2
0132 3000 00388 MOVLW D'0'
0133 0628 00389 XORWF B3,W
0134 1903 00390 BTFSC STATUS,Z
0135 293E 00391 GOTO UNITY1
0136 0AB1 00392 INCF B1,F
0137 0828 00393 MOVF B3,W
0138 0231 00394 SUBWF B1,W
0139 00B4 00395 MOVWF B2
013A 0827 00396 MOVF A,W
013B 0230 00397 SUBWF A1,W
013C 00B5 00398 MOVWF A2
013D 294B 00399 GOTO DISPLAY2
013E      00400 UNITY1 ;*when units place and tens place are both zeroes
013E 01B4 00401 CLRF B2
013F 0AB0 00402 INCF A1,F
0140 0827 00403 MOVF A,W
0141 0230 00404 SUBWF A1,W
0142 00B5 00405 MOVWF A2
0143 294B 00406 GOTO DISPLAY2
0144      00407 TEN1 ;*when tens place is zero and units place
                is nonzero
0144 0828 00408 MOVF B3,W
0145 0231 00409 SUBWF B1,W
0146 00B4 00410 MOVWF B2
0147 0827 00411 MOVF A,W
0148 0230 00412 SUBWF A1,W
0149 00B5 00413 MOVWF A2
014A 294B 00414 GOTO DISPLAY2
014B      00415 DISPLAY2
014B 1105 00416 BCF PORTA,RS
014C 1205 00417 BCF PORTA,RW
014D 30A8 00418 MOVLW H'a8' ;TO GOTO SECOND LINE
014E 0087 00419 MOVWF PORTC
014F 2055 00420 CALL PULSE_E
0150 2026 00421 CALL T48US
0151 1505 00422 BSF PORTA,RS ;RS line to 1 to i/p Data
0152 1205 00423 BCF PORTA,RW ;R/W line to 0 to write
0153 3053 00424 MOVLW 'S'
0154 0087 00425 MOVWF PORTC ;send character to LCD
0155 2055 00426 CALL PULSE_E ;Clock the LCD
0156 2026 00427 CALL T48US
0157 1505 00428 BSF PORTA,RS ;RS line to 1 to i/p Data
0158 1205 00429 BCF PORTA,RW ;R/W line to 0 to write
0159 3065 00430 MOVLW 'e'
015A 0087 00431 MOVWF PORTC ;send character to LCD
015B 2055 00432 CALL PULSE_E ;Clock the LCD
015C 2026 00433 CALL T48US
015D 1505 00434 BSF PORTA,RS ;RS line to 1 to i/p Data
015E 1205 00435 BCF PORTA,RW ;R/W line to 0 to write
015F 3074 00436 MOVLW 't'
0160 0087 00437 MOVWF PORTC ;send character to LCD
0161 2055 00438 CALL PULSE_E ;Clock the LCD
0162 2026 00439 CALL T48US
0163 1505 00440 BSF PORTA,RS ;RS line to 1 to i/p Data
0164 1205 00441 BCF PORTA,RW ;R/W line to 0 to write
0165 30FE 00442 MOVLW H'FE'
0166 0087 00443 MOVWF PORTC ;send character to LCD
0167 2055 00444 CALL PULSE_E ;Clock the LCD
0168 2026 00445 CALL T48US
0169 1505 00446 BSF PORTA,RS ;RS line to 1 to i/p Data
016A 1205 00447 BCF PORTA,RW ;R/W line to 0 to write
016B 3050 00448 MOVLW 'P'
016C 0087 00449 MOVWF PORTC ;send character to LCD
016D 2055 00450 CALL PULSE_E ;Clock the LCD
016E 2026 00451 CALL T48US
016F 1505 00452 BSF PORTA,RS ;RS line to 1 to i/p Data
0170 1205 00453 BCF PORTA,RW ;R/W line to 0 to write
0171 306F 00454 MOVLW 'o'
0172 0087 00455 MOVWF PORTC ;send character to LCD
0173 2055 00456 CALL PULSE_E ;Clock the LCD
0174 2026 00457 CALL T48US
0175 1505 00458 BSF PORTA,RS ;RS line to 1 to i/p Data
0176 1205 00459 BCF PORTA,RW ;R/W line to 0 to write
0177 3069 00460 MOVLW 'i'
0178 0087 00461 MOVWF PORTC ;send character to LCD
0179 2055 00462 CALL PULSE_E ;Clock the LCD
017A 2026 00463 CALL T48US
017B 1505 00464 BSF PORTA,RS ;RS line to 1 to i/p Data
017C 1205 00465 BCF PORTA,RW ;R/W line to 0 to write
017D 306E 00466 MOVLW 'n'
017E 0087 00467 MOVWF PORTC ;send character to LCD
017F 2055 00468 CALL PULSE_E ;Clock the LCD
0180 2026 00469 CALL T48US
0181 1505 00470 BSF PORTA,RS ;RS line to 1 to i/p Data
0182 1205 00471 BCF PORTA,RW ;R/W line to 0 to write
0183 3074 00472 MOVLW 't'
0184 0087 00473 MOVWF PORTC ;send character to LCD
0185 2055 00474 CALL PULSE_E ;Clock the LCD
0186 2026 00475 CALL T48US
0187 1505 00476 BSF PORTA,RS ;RS line to 1 to i/p Data
0188 1205 00477 BCF PORTA,RW ;R/W line to 0 to write
0189 303A 00478 MOVLW ':'
018A 0087 00479 MOVWF PORTC ;send character to LCD
018B 2055 00480 CALL PULSE_E ;Clock the LCD
018C 2026 00481 CALL T48US
018D 1505 00482 BSF PORTA,RS ;RS line to 1 to i/p Data
018E 1205 00483 BCF PORTA,RW ;R/W line to 0 to write
018F 30FE 00484 MOVLW H'fe'
0190 0087 00485 MOVWF PORTC ;send character to LCD
0191 2055 00486 CALL PULSE_E ;Clock the LCD
0192 2026 00487 CALL T48US
0193 1505 00488 BSF PORTA,RS ;RS line to 1 to i/p Data
0194 1205 00489 BCF PORTA,RW ;R/W line to 0 to write
0195 3030 00490 MOVLW D'48'
0196 07B5 00491 ADDWF A2,F
0197 0835 00492 MOVF A2,W
0198 0087 00493 MOVWF PORTC ;send character to LCD
0199 2055 00494 CALL PULSE_E ;Clock the LCD
019A 2026 00495 CALL T48US ;delay for LCD busy
019B 1505 00496 BSF PORTA,RS ;RS line to 1 to i/p Data
019C 1205 00497 BCF PORTA,RW ;R/W line to 0 to write
019D 3030 00498 MOVLW D'48'
019E 07B4 00499 ADDWF B2,F
019F 0834 00500 MOVF B2,W
01A0 0087 00501 MOVWF PORTC ;send character to LCD
01A1 2055 00502 CALL PULSE_E ;Clock the LCD
01A2 2026 00503 CALL T48US
01A3 1505 00504 BSF PORTA,RS ;RS line to 1 to i/p Data
01A4 1205 00505 BCF PORTA,RW ;R/W line to 0 to write
01A5 3030 00506 MOVLW D'48'
01A6 07B3 00507 ADDWF C2,F
01A7 0833 00508 MOVF C2,W
01A8 0087 00509 MOVWF PORTC ;send character to LCD
01A9 2055 00510 CALL PULSE_E ;Clock the LCD
01AA 2026 00511 CALL T48US
01AB 1505 00512 BSF PORTA,RS ;RS line to 1 to i/p Data
```

```
01AC   1205   00513   BCF PORTA,RW          ;R/W line to 0 to write
01AD   30DF   00514   MOVLW B'11011111'     ;to print symbol for degree
01AE   0087   00515   MOVWF PORTC           ;send character to LCD
01AF   2055   00516   CALL PULSE_E          ;Clock the LCD
01B0   2026   00517   CALL T48US
01B1   1505   00518   BSF PORTA,RS          ;RS  line to 1 to i/p Data
01B2   1205   00519   BCF PORTA,RW          ;R/W line to 0 to write
01B3   304B   00520   MOVLW 'K'
01B4   0087   00521   MOVWF PORTC           ;send character to LCD
01B5   2055   00522   CALL PULSE_E          ;Clock the LCD
01B6   2026   00523   CALL T48US
              00524                         ;****to generate the ON/OFF signal
                                            subtract ADUSER from AD1
01B7          00525   CONTROL
01B7   0820   00526           MOVF AD1,W
01B8   0221   00527           SUBWF ADUSER,W
01B9   1803   00528           BTFSC STATUS,0
01BA   29BD   00529           GOTO CONTROL1
01BB   1086   00530           BCF PORTB,1
01BC   29C3   00531           GOTO CURSOR1
01BD          00532   CONTROL1
01BD   1903   00533   BTFSC STATUS,Z
01BE   29C1   00534   GOTO CLEAR            ;To give the OFF signal
01BF   1486   00535           BSF PORTB,1           ;To give the ON signal
01C0   29C3   00536           GOTO CURSOR1
01C1          00537   CLEAR
01C1   1086   00538   BCF PORTB,1
01C2   29C3   00539   GOTO CURSOR1
01C3          00540   CURSOR1     ;*********to give cursor home command to LCD
                      00541
01C3   1105   00542   BCF  PORTA,RS
01C4   1205   00543   BCF  PORTA,RW
01C5   3003   00544   MOVLW H'03'
01C6   0087   00545   MOVWF PORTC
01C7   2055   00546   CALL PULSE_E
01C8   2026   00547   CALL T48US
01C9   2815   00548   GOTO BEGIN
01CA          00549   INT
01CA   2805   00550   GOTO MAINLINE
                      00551   END
Program Memory Words Used:  456
Program Memory Words Free:  3640
Errors   :    0
Warnings :    0 reported,    2 suppressed
Messages :    0 reported,    4 suppressed
```

Readers' comments

I have the following queries:

1. In line No. 00117 of the program code (temp.lst), the result of analogue-to-digital conversion (ADC) has been subtracted from '255.' In the explanation, it is mentioned that '255' stands for '5V,' so what's the purpose behind doing so?

2. Again, the result of the above subtraction is multiplied by '2' in line No. 00180. Why?

3. At the start of the 'Subtract' subroutine (line No. 00220), the comment states that "temp in Kelvin 500-(255-ADRES)*2." In this equation, what is the reason behind using '500' and not any other number? What is the significance of the given formula?

4. In the same subroutine, i.e., 'Subtract,' the author has used Nos. 10, 9, 4 and 0 (line Nos 00221, 00223, 00225 and 00227, respectively) for digit-wise subtraction. Here, why specifically only these numbers are used?

Nirmit Dudhia
Through e-mail

The authors, Sreeja Menon and Niranjana Ashok, reply:

The logic behind the formula 500–(255–ADRES)*2 follows. The ADC inside the PIC is 8-bit and hence the maximum digital value is '255' corresponding to the analogue voltage value of 5V, as given in the specification of the PIC. Thus, we arrive at the relationship that '5V' corresponds to '255' (digital value).

Again, the temperature sensor used has the sensitivity of 1 µA/K, and the temperature sensing range is 218K to 423K. Since the output from the temperature sensor is a current corresponding to the sensed temperature, we used a 10-kilo-ohm resistor to convert the current into voltage. Thus, the voltage sensitivity of our set-up will be 1mV(0.01V)/K resulting in the input voltage range to the ADC of the PIC as 2.18V to 4.23V.

Now, in order to arrive at the formula, we made an approximation to the linearity in the relationship between the temperature and voltage.

That is,

255	5V	500K
254	4.98V	498K
253	4.96V	496K

and so on.

Since the sensitivity is 0.01V/K, for each unit change in the digital value the voltage value changes by 0.01V. The formula was derived to approximate the above relationship between the digital value and the temperature value in Kelvin scale.

We want the temperature in Kelvin value to energise/de-energise the relay and also to display on the LCD panel. Hence, when the ADC shows '255' the display should show '500K,' when the ADC shows '254' the display should show '498K,' and so on. The relay should energise/de-energise accordingly.

So based on the relationship mentioned above, we arrived at a linear relationship that is an approximation for the actual non-linear relationship between the sensed temperature and the digital value.

For example, when ADRES is '248,' according to the formula, the temperature is calculated as follows:

(255–248)x2=7x2=14

Now, 500–14=486K

Thus we arrive at the temperature in Kelvin.

This explanation suffices for the first three queries of Mr Dudhia.

Regarding the fourth query, since the PIC is a RISC processor, with 35 instructions, we had to arrive at a complicated logic to do the subtraction from '500' as the maximum possible digital value was '255.' Hence, to do an operation like 500–238, we need to do digit-wise subtraction. For the units place we perform 10–8=2, for the tens place we perform 9–3=6, and for the hundreds place we perform 4–2=2, resulting in '262' as the correct answer. The algorithm is based on the fact that we need to do subtraction from '500' and hence the numbers 10, 9 and 4. Again, if the units place of the number to be subtracted is '0,' we need to subtract the tens digit from '10' and the units digit from '0' and so on.

For example, 500-240 will be done as follows:

Units digit of the result=0–0=0

Tens digit of the result=10–4=6

Hundreds digit of the result=4–2=2

We have put 'A1' as the number from which the units digit is to be subtracted, 'B1' for the tens digit and 'C1' for the hundreds digit.

If the number to be subtracted has non-zero units, tens and hundreds digits:

A1=10, B1=9, C1=4

If only the units digit is zero:

A1=0, B1=10, C1=4

If the tens digit is zero:

A1=10, B1=9, C1=5

and so on for different combinations.

STEPPER MOTOR CONTROL USING 89C51 MICROCONTROLLER

MANDEEP SINGH WALIA

Here's a stepper motor controller based on 89C51 microcontroller to control the rotation of a DC stepper motor in clockwise and anti-clockwise directions. The controller is simple and easy-to-construct, and can be used in many applications including machine control and robotics for controlling the axial rotation in XY plane. A similar circuit can be added to control the rotation of the motor in either XZ or YZ plane.

Fig. 1 shows the block diagram of the stepper motor control system. The power supply section (in Fig. 2) consists of a step-down transformer (7.5V AC, 1A), bridge rectifier (comprising diodes D1 through D4), filter capacitors (C1 and C2) and regulator IC 7805.

We have used here an Atmel make low-power, high-performance, 8-bit CMOS microcontroller AT89C51 with 4 kB of Flash programmable and erasable read-only memory (PEROM). It has a 128x8-bit internal RAM, 32 programmable input/output (I/O) lines and two 16-bit timer/counters. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional non-volatile memory programmer.

By combining a versatile 8-bit CPU with Flash on a monolithic chip, Atmel AT89C51 is a powerful, highly flexible and cost-effective solution to many embedded control applications. From traffic control equipment to input devices, computer networking products and stepper motor controllers, 89C51 microcontrollers deliver a high performance with a choice of configurations and options matched to the specific needs of each application.

IC AT89C51 features:

1. 8-bit CPU with math registers A and B
2. 16-bit program counter (PC) and data pointer (DPTR)
3. 8-bit program status word (PSW)
4. 8-bit stack pointer (SP)

The control switches for the motor are connected to Reset and Port $P_{0.7}$ pins of the microcontroller.

Circuit description

Fig. 2 shows the complete circuit of the stepper motor controller. When power supply switch S1 is closed, LED1 glows to indicate the presence of power in the circuit. Capacitor C3 connected to pin 9 (RST) provides the power-on reset to the microcontroller.

The stepper motor is connected to port pins P2.4 through P2.7 of the microcontroller (IC2) through the motor-driver circuit consisting of four Darlington pairs comprising transistors BC548 and SL100 (T1-T2, T3-T4, T5-T6 and T7-T8). Coils 1 through 4 are the stepper motor coils.

When transistors conduct, 5V (Vcc) is applied to the coils and the currents flowing through them create magnetic fields and the motor starts rotating. The magnetic field energy thus created is stored in the coils.

When transistors stop conducting, power to the coils is cut off, the magnetic field collapses and a reverse voltage (called inductive kickback or back emf) is generated in the coils. The back emf can be more than 100 volts. The diodes connected across the coils absorb the reverse voltage spike. This voltage, if not absorbed by the diodes, may produce opposite torque and cause improper rotation of the motor and

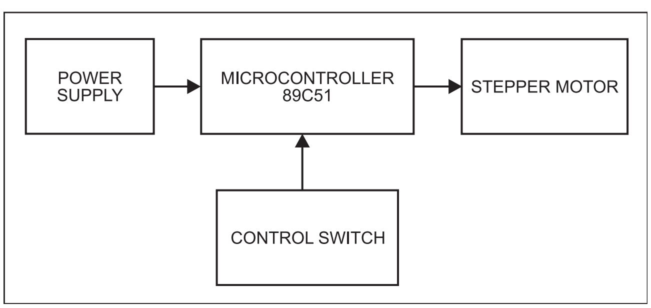


Fig. 1: Block diagram of the stepper motor control system

TABLE I
Power Consumption of Microcontrollers

IC	V_{oh}	I_{oh}	V_{oi}	I_{oi}	V_{il}	I_{il}	V_{ih}	I_{ih}	P_t
CMOS	2.4V	–60 µA	0.45V	1.7 mA	0.9V	10 µA	1.9V	10 µA	50 mW
NMOS	2.4V	–80 µA	0.45V	1.7 mA	0.8V	–800 mA	2.0V	10 µA	800 mW

PARTS LIST

Semiconductors:	
IC1	- 7805 5V regulator
IC2	- AT89C51 microcontroller
T1, T3, T5, T7	- BC548 npn transistors
T2, T4, T6, T8	- SL100 npn transistors
D1-D8	- 1N4001 rectifier diodes
LED1	- Red LED (5mm dia.)
Resistors (all ¼-watt, ±5% carbon):	
R1	- 100-ohm
R2	- 10-kilo-ohm
R3, R5, R7, R9	- 1-kilo-ohm
R4, R6, R8, R10, R11	- 470-ohm
Capacitors:	
C1	- 220µF, 25V electrolytic
C2	- 100µF, 16V electrolytic
C3	- 10µF, 16V electrolytic
C4, C5	- 33pF ceramic disk
C6	- 100µF, 16V electrolytic
Miscellaneous:	
X1	- 230VAC primary to 0-7.5V, 1A secondary step-down transformer
	- 5V DC stepper motor
S1, S3	- on/off switch
S2	- push-to-on switch

also damage the transistors. You can use virtually any type of rectifier or switching diodes of appropriate current and reverse voltage breakdown rating.

Clock and reset circuit. Two 33pF capacitors (C4 and C5) are connected to pins 18 and 19 of the microcontroller, respectively, with an 11.059MHz piezoelectric crystal (X_{TAL1}) across them. The clock frequency of the microcontroller depends on the frequency of the crystal oscillator used. Typically, the maximum and minimum frequencies are 1 MHz and 16 MHz, respectively, so we should use a piezoelectric crystal with a frequency in this range. The speed of the stepper motor is proportional to the frequency of the input pulses or it is inversely proportional to the time delay between pulses, which can be achieved through software by making use of instruction execution time.

The time taken by any instruction to get executed can be computed as follows:

$$\text{Time}=\frac{C\times 12}{F}$$

where 'C' is the number of cycles an instruction takes to execute and 'F' is the crystal frequency.

The crystal frequency in this circuit is 11.059 MHz, so the time taken to execute, say, ADD A, R1 (single-cycle instruction), is about one microsecond (µs). Use of a 6MHz crystal will bring down the instruction execution speed to to 2 µs.

When power is applied, the reset input must first go high and then low. A resistor-capacitor combination (R1-C3) is used to achieve this until the capacitor begins to charge. At a threshold of about 2.5V, the reset input reaches a low level and the microcontroller begins to function normally. Reset switch (S2) allows you to reset the program without having to interrupt the power.

One major feature of 89C51 microcontroller is the versatility built into the I/O circuits that connect the microcontroller to the outside world. Ports P0 through P3 of the microcontroller are not capable of driving loads that require tens of milliamperes (mA). Logic level current, voltage and power requirement for different versions of microcontrollers are given in Table I.

Driver circuit design. The microcontroller outputs a current of 1.7 mA. To drive the coil of a stepper motor requiring a torque of 7 kg-cm, 12V DC and 2 amp/phase, we have to use a driver circuit that amplifies the current from 1.7 mA to 3 amp.

As mentioned earlier, we have used

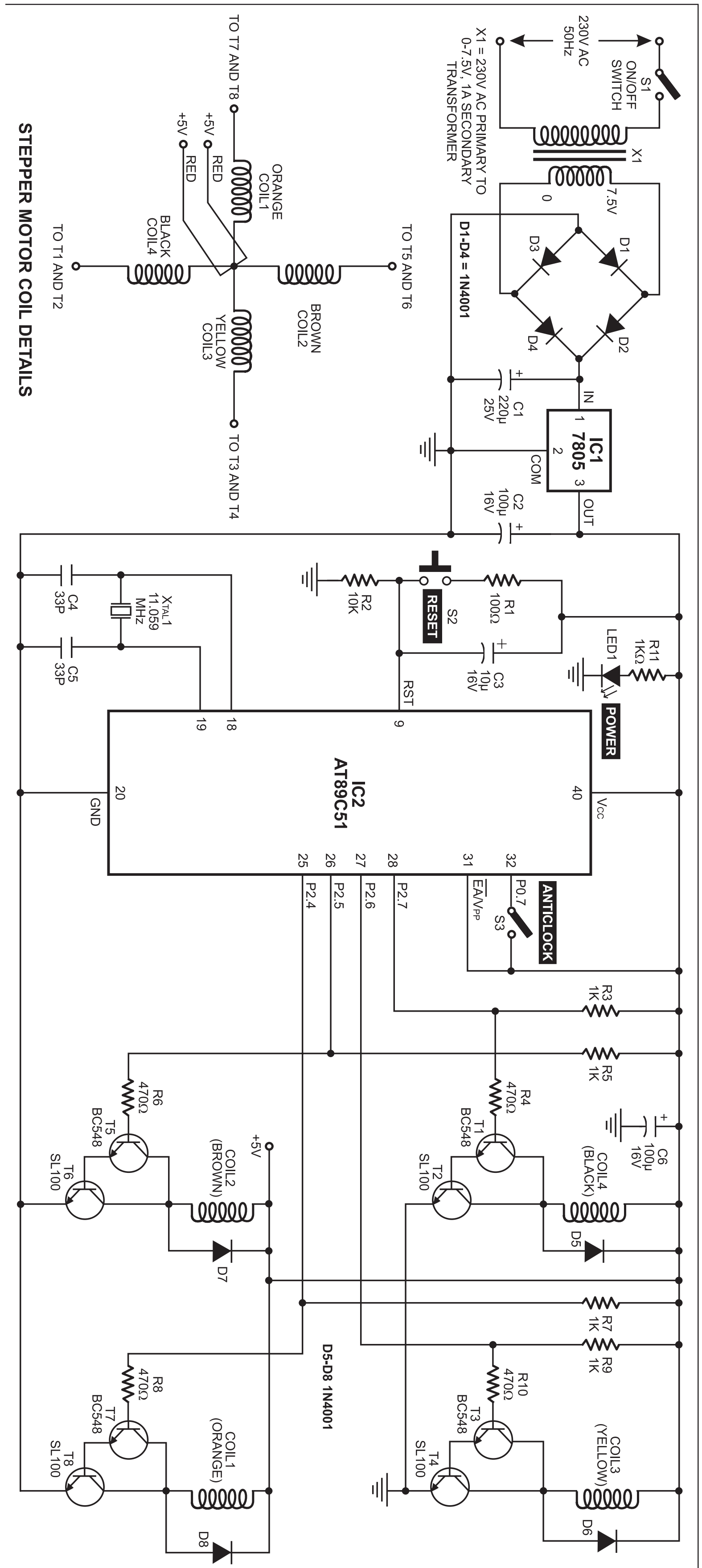


Fig. 2: Circuit of stepper motor control system

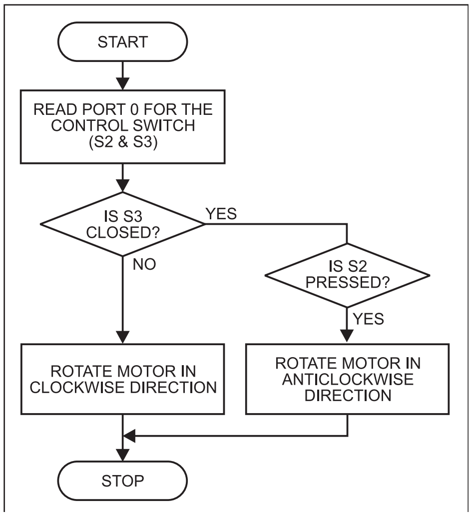


Fig. 3: Flow-chart of the program

TABLE II
Clockwise Step Sequence of the Motor

A1	A2	B1	B2	A1	A2	B1	B2	Hex value
0	0	1	1	0	0	1	1	=33h
0	1	1	0	0	1	1	0	=66h
1	1	0	0	1	1	0	0	=CCh
1	0	0	1	1	0	0	1	=99h

Anti-clockwise Step Sequence of the Motor

A1	A2	B1	B2	A1	A2	B1	B2	Hex value
0	0	1	1	0	0	1	1	=33h
1	0	0	1	1	0	0	1	=99h
1	1	0	0	1	1	0	0	=CCh
0	1	1	0	0	1	1	0	=66h

BC548 and SL100 as the driver transistors for driving a low-power rated stepper motor such as the one used in earlier 14cm (5.5-inch) floppy drives. But for a 7 kg-cm stepper motor, a driver circuit using transistors SL100 and 2N3055 would be needed to amplify the current to 2.72 amp. Typically, SL100 and 2N3055 each has a gain (h_{fe}) of 40, but 2N3055 can handle larger current since it belongs to the family of power transistors. So a heat-sink is required to dissipate the heat generated.

The output gain of the Darlington pair of SL100 and 2N3055 transistors is:

$$A_{V_o} = A_{V1} \times A_{V2}$$
$$= 40 \times 40$$
$$= 1600$$
$$A_{V_o} = Io/Iin = 1600$$

where Io is the output current and Iin is the input current of the Darlington pair.

$$Io = 1600 \times 1.7 \text{ mA}$$
$$= 2.72 \text{ A}$$

Since the stepper motor has four coils, we need to use four Darlington pairs.

Programming

The program is written in Assembly language and compiled using ASM51 cross-assembler. The listing file is given at the end of this article. 89C51 microcontroller is programmed using Atmel's Flash programmer.

One-step rotation of the stepper motor used in this project equals 1.8°. When you program the motor for 200 steps, the motor makes one complete revolution, i.e. 360°. In the program, the line 'MOV R7, #0CAH' defines the rotation by 202 steps. The hex number '0CAH' equals the decimal number '202.' However, one can change the number of steps in the program as per one's requirement.

The step sequence is defined by the line 'MOV A, #033H.' Table II shows the step sequence for 100 steps to energise the windings of the stepper motor in clockwise

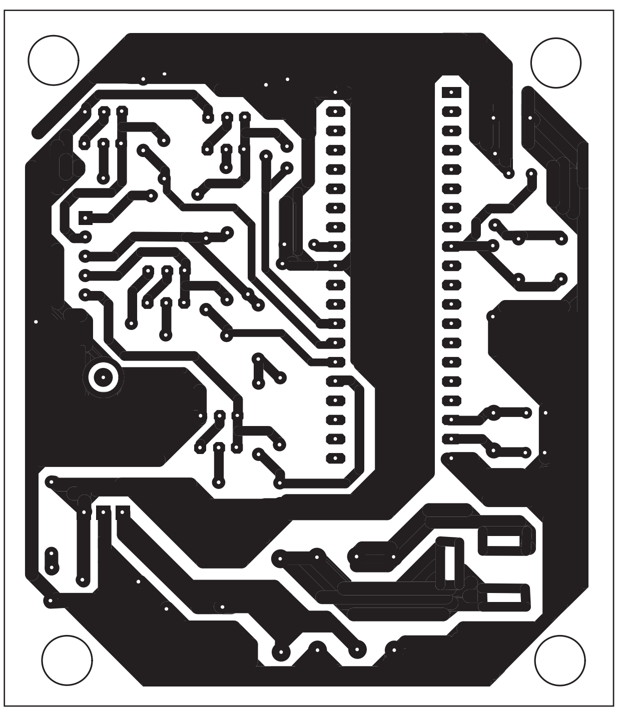


Fig. 4: Actual-size, single-side PCB for stepper motor control system using 89C51 microcontroller

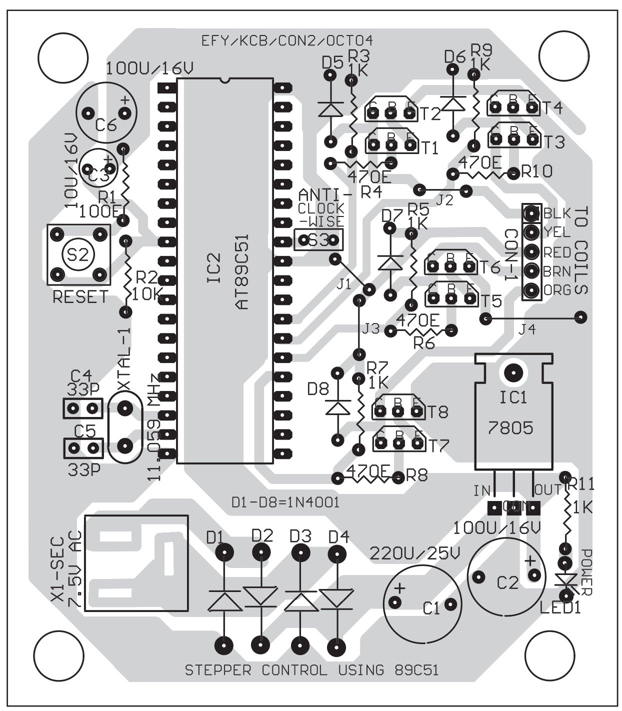


Fig. 5: Component layout for the PCB

and anti-clockwise directions. The rotor of the stepper motor is in a position of minimum reluctance and maximum flux. Thus by energising the windings (represented by A1, A2, B1 and B2), the rotor takes the position accordingly. In the program, the instructions 'RR A' and 'RL A' are used for clockwise and anti-clockwise, respectively.

S1 and S3 are toggle switches, while S2 is a tactile switch. Switch S3 interfaced to pin 32 of the microcontroller determines the direction of rotation. When the switch is opened the motor rotates in clockwise direction, and when the switch is closed the motor rotates in anti-clockwise direction.

For anti-clockwise rotation of the motor, reset switch S2 is to be pressed momentarily after S3 is closed (see Fig. 3). In case you observe an abnormal motion of the motor either in clockwise or anti-clockwise direction, pressing reset switch S2 momentarily will make the motor run smoothly.

Construction and working

You can assemble the circuit on any general-purpose PCB. An actual-size, single-side PCB for the stepper motor controller is shown in Fig. 4 and its component layout in Fig. 5.

Mount a 40-pin IC base for the microcontroller on the PCB, so you can remove the chip easily when required. Normally, six wires of different colours (two being red) are available for connection to the stepper motor. The sequence for connecting the stepper motor coils to the driver card is shown in Fig. 2.

After you are done with the hardware part, assemble the program (stpb1.asm) using ASM51 assembler. Load the hex file generated by ASM51 into a programmer and burn it into the chip. Now put the programmed chip on the IC base on the PCB.

Switch on the power supply to the circuit using switch S1. If motor rotation is not stable, press S2 momentarily. If the motor does not move at all, check the connections.

Note. The source code and the relevant files for this article have been included in CD.

STPB1.LST

```
              $MOD51
0000          ORG 0000H
0000  E580    MOV A, P0
0002  33      RLC A
0003  500B    JNC P12
0005  7FCA    MOV R7, #0CAH;
0007  7433    MOV A, #033H;
0009  F5A0    P13:  MOV P2, A;
000B  23      RL A;
000C  111B    ACALL DELAY
000E  DFF9    DJNZ R7, P13
0010  7FCA    P12:  MOV R7, #0CAH;
0012  7433    MOV A, #033H;
0014  F5A0    P11:  MOV P2, A;
0016  03      RR A;
0017  111B    ACALL DELAY
0019  DFF9    DJNZ R7, P11
001B  758910  DELAY: MOV TMOD, #10H
001E  7B05    MOV R3, #05
0020  758B08  Z:    MOV TL1, #8D
0023  758D01  MOV TH1, #1D
0026  D28E    SETB TR1
0028  308FFD  BACK:  JNB TF1, BACK
002B  C28E    CLR TR1
002D  C28F    CLR TF1
002F  DBEF    DJNZ R3, Z
0031  22      RET
              END

VERSION 1.2k ASSEMBLY COMPLETE, 0 ERRORS FOUND
```

Readers' comments

What changes are to be made if I use a 12V DC stepper motor instead of the 5V DC stepper motor?

Jitendra Savaliya
Through e-mail

The author, Mandeep Singh Walia, replies:

Mr Savaliya should use the Darlington pair of transistors SL100 and 2N3055. It has been mentioned in the text also under the heading 'driver circuit design.'

The point to be noted is that the power supply to the motor driver circuitry and the controller circuitry should be different. The controller works off +5V and the power supply to the driver circuit will be +12V. The grounds for both the supplies will be the same. This configuration works well with the 12V DC stepper motor.

MICROPROCESSOR-BASED HOME SECURITY SYSTEM

B.B. MANOHAR

Don't take the chance of becoming a victim of burglary, which is often accompanied by violence. Protect your family and valuables with this microprocessor-based home security system that will let you rest your head knowing that should any one try to break into your home, an alarm will go off and the police will be alerted immediately.

The 8085 microprocessor-based home security system, as shown in Fig. 1, consists of transmitter, receiver, phase-locked loop (PLL) and processing sections.

The transmitter section continuously transmits infrared (IR) rays, which are received by the receiver section. The received signal is further amplified and given to the PLL section, where its frequency is locked to the transmitted frequency.

When the IR signal is interrupted, the microprocessor starts working as per the program burnt into the erasable programmable read-only memory (EPROM) and controls the siren, telephone (via cradle and redial switches) and cassette player (in which the alert message is recorded already) via the respective relays.

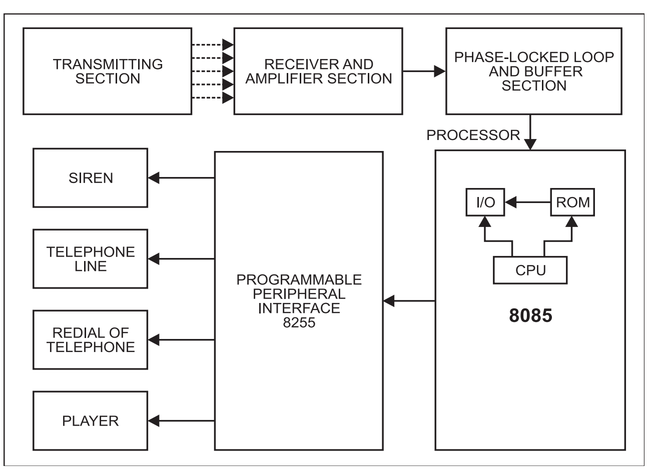


Fig. 1: Block diagram of the 8085 microprocessor-based home security system

Circuit description

Fig. 2 shows the complete circuit of the 8085 microprocessor-based home security system.

In the transmitter section, NE555 (IC1) is wired as an astable multivibrator whose oscillating frequency is decided by resistors R1 and R2, preset VR1 and capacitor C1. Capacitor C3 bypasses the noise to ground, preventing any change in the calculated pulse-width.

The output of IC1 is fed to the base of transistor T1, which drives an infrared light-emitting diode (IR LED) to transmit the modulated IR signal. Resistor R4 limits the current flowing through the IR LED. Preset VR1 is used to vary the modulating frequency.

The transmitter and the receiver are arranged such that the transmitted IR rays fall directly onto phototransistor L14G1 (T2) of the receiver. The signal received by T2 is amplified by transistor T3 and operational amplifier µA741 (IC2). Series input resistor R8 and feedback resistor R9 determine the gain of op-amp IC2. The amplified signal is applied to pin 3 of PLL LM567 (IC3) through capacitor C4.

IC LM567 is a highly stable PLL with synchronous AM lock detection and power output circuitry. It is primarily used as a frequency decoder, which drives the load whenever a sustained frequency

PARTS LIST

Semiconductors:

IC1	- NE555 timer
IC2	- µA741 operational amplifier
IC3	- LM567 phase-locked loop
IC4	- 8085 microprocessor
IC5	- 2732A EPROM (4k)
IC6	- 74LS373 octal transparent latch
IC7	- 8255A programmable peripheral interface
IC8, IC9	- MCT2E optocoupler
IC10	- 7805 5V regulator
IC11	- 7809 9V regulator
IC12	- 74LS00 NAND gate
T1, T3-T9	- BC548 npn transistor
T2	- L14G1 phototransistor
D1	- 1N4148 switching diode
D2-D10	- 1N4007 rectifier diode
LED1-LED3	- Red LED
IR LED1	- Infrared LED

Resistors (all ¼-watt, ±5% carbon, unless stated otherwise):

R1, R2	- 5.6-kilo-ohm
R3, R16, R18-R22, R25	- 4.7-kilo-ohm
R4	- 100-ohm
R5	- 3.9-kilo-ohm
R6, R8, R12, R15, R17	- 1-kilo-ohm
R7, R10, R11, R13, R14	- 10-kilo-ohm
R9	- 100-kilo-ohm
R23	- 120-ohm
R24	- 470-ohm

Capacitors:

C1	- 3.3nF ceramic disk
C2, C6, C13, C14	- 0.1µF ceramic disk
C3, C8	- 0.01µF ceramic disk
C4	- 1nF ceramic disk
C5	- 10µF, 25V electrolytic
C7	- 2.2µF, 25V electrolytic
C9	- 10µF,10V electrolytic
C10, C11	- 10pF ceramic disk
C12	- 1000µF, 50V electrolytic

Miscellaneous:

X1	- 230V AC primary to 12V-0-12V, 300mA secondary transformer
X_{TAL}	- 3.5MHz crystal
S1	- Push-to-on switch
S2	- On/off switch
RL1, RL2, RL4, RL5	- 12V, 200-ohm, 1C/O relay
RL3	- 12V, 200-ohm, 2C/O relay

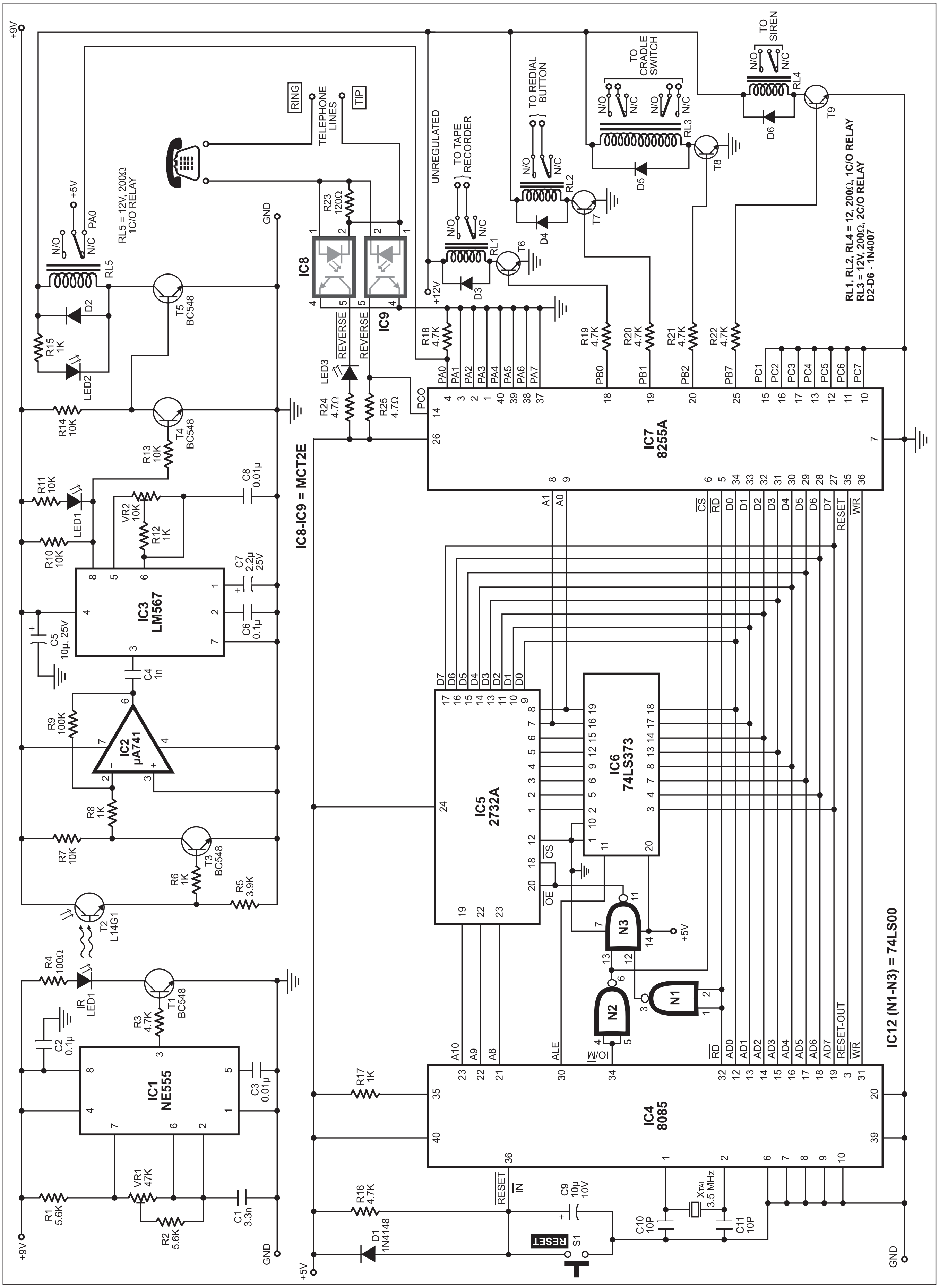


Fig. 2: Circuit of the 8085 microprocessor-based home security system

falling within its detection band is present in its self-biased input. The centre frequency of the detection band and output delay are independently determined by the external components.

In the absence of any input signal, the centre frequency of the PLL's internal free-running, current-controlled oscillator is determined by resistor R12 and capacitor C8. Preset VR2 is used for tuning IC3 to the desired centre frequency in the 6-10kHz range, which should match the modulating frequency of the transmitter. Capacitors C6 and C7 are used as low-pass filter (LPF) and output filter, respectively.

When the received signal is locked to the frequency of the transmitted signal, pin 8 of IC3 goes low and LED1 glows. Since pin 8 is connected to the base of transistor T4 through resistor R13, it is cut off and its collector voltage rises. As a result, transistor T5 is forward biased to energise relay RL5. The pole and normally-closed (N/C) contact of relay RL5 are connected to +5V and pin 4 (PA0) of IC7, respectively.

Normally, the transmitted IR signal falls on phototransistor T2, so relay RL5 is energised and there is no input to the processor via IC7. When the IR signal is interrupted, relay RL5 de-energises to provide a high (TTL-level) signal to the processor via port A of the programmable peripheral interface (PPI).

The processing section consists of an 8-bit 8085 microprocessor (IC4), EPROM IC 2732A (IC5), octal transparent latch IC 74LS373 (IC6) and programmable peripheral interface IC 8255A (IC7).

When the microprocessor gets a high signal from port A of IC7, it starts working as per the code loaded in the EPROM (IC5).

EPROM IC 2732A is a UV erasable and electrically programmable memory. It is organised as 4096 words×8 bits. The transparent window allows the user to expose the chip to ultraviolet light to erase the chip. After erasing the chip, a new program can be burnt into it.

IC 8085 (IC4) is an 8-bit, general-purpose microprocessor capable of addressing 64k of memory. It includes most of the logic circuitry required for performing computing tasks and communicating with the peripherals.

The low-order multiplexed address and data lines AD0 through AD7 of IC4 are connected to the EPROM (IC5) through the octal latch (IC6), while its high-order address lines A8 through A10 are directly connected to the EPROM. Address lines A0 through A7 are separated from data lines D0 through D7 by latch-enable signal (ALE).

Address latch-enable (ALE) pin 30 of the microprocessor is connected to latch-enable pin 11 of IC6. When ALE is high, the latch is transparent, i.e. the output changes according to the input data. When ALE goes low, the low-order address is latched at the output of IC6.

Data lines D0 through D7 of the microprocessor are connected to the data lines of IC5 and IC7 each. Chip-select signal ($\overline{CS}$) for IC5 is generated by $\overline{RD}$ and IO/M lines with the help of a NAND gate. The inverted IO/$\overline{M}$ signal provides CS signal to IC7.

IC 8255A (IC7) is a general-purpose programmable device compatible with most microprocessors. It has three programmable ports, any of which can be used for bidirectional data transfer. The 24 I/O pins can be grouped in two 8-bit ports (ports A and B) and the remaining eight bits as port C. The eight bits of port C can be used as individual bits or grouped in two 4-bit ports, namely, C_{UPPER} and C_{LOWER}. Ports A and C are configured as the input ports, and port B is configured as the output port. Port A is used for intruder detection, port B for activating the siren, cassette player, telephone cradle switch and redial button, and port C for polarity-reversal detection.

PB0 (pin 18), PB1 (pin 19), PB2 (pin 20) and PB7 (pin 25) of IC7 are connected to the bases of transistors T6 through T9 via resistors R19 through R22, respectively. A high signal on these pins energises relays RL1 through RL4. Switch S1 is used to reset IC4.

As you may be aware, telephone exchanges provide DC voltage reversal facility to PCOs (and other subscribers for a fee) to indicate call maturity. The same is assumed to have been incorporated in our telephone.

The circuit for detecting the polarity reversal in the telephone line is built around optocouplers IC8 and IC9. Normally, TIP is positive with respect to the RING lead of the telephone line. With the handset in off-hook position, a nominal loop current of 10 mA is assumed to flow through the telephone lines. Resistor R23

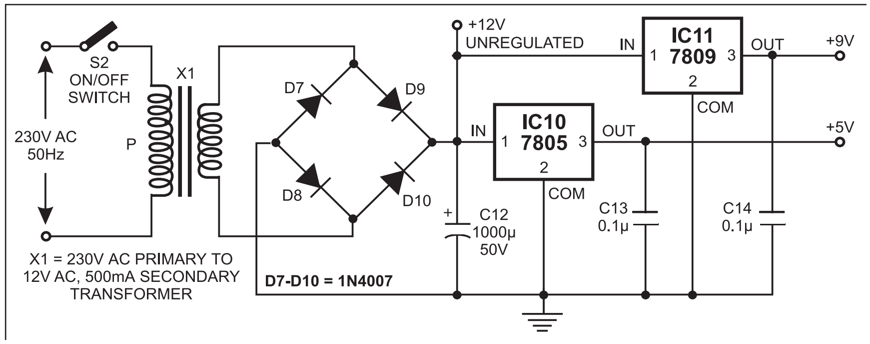


Fig. 3: Power supply circuit

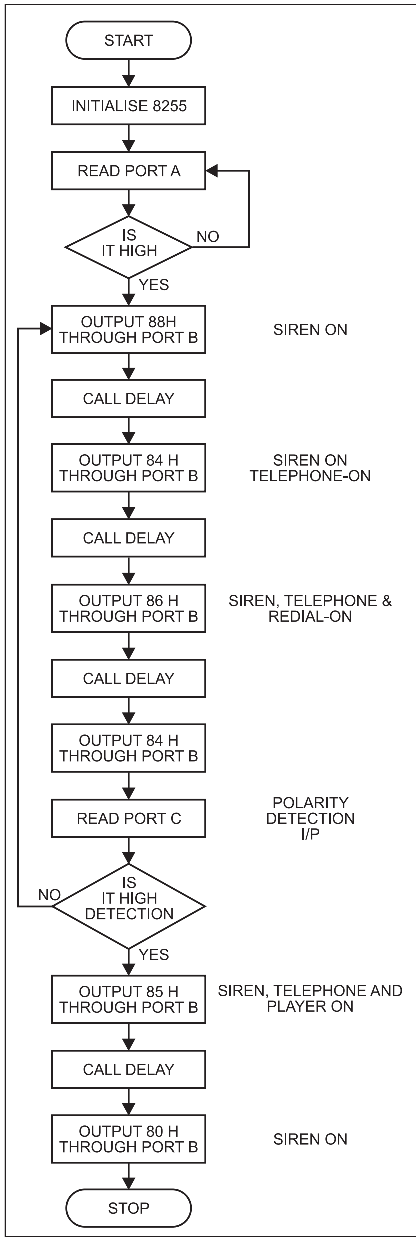


Fig. 4: Flow-chart of the program

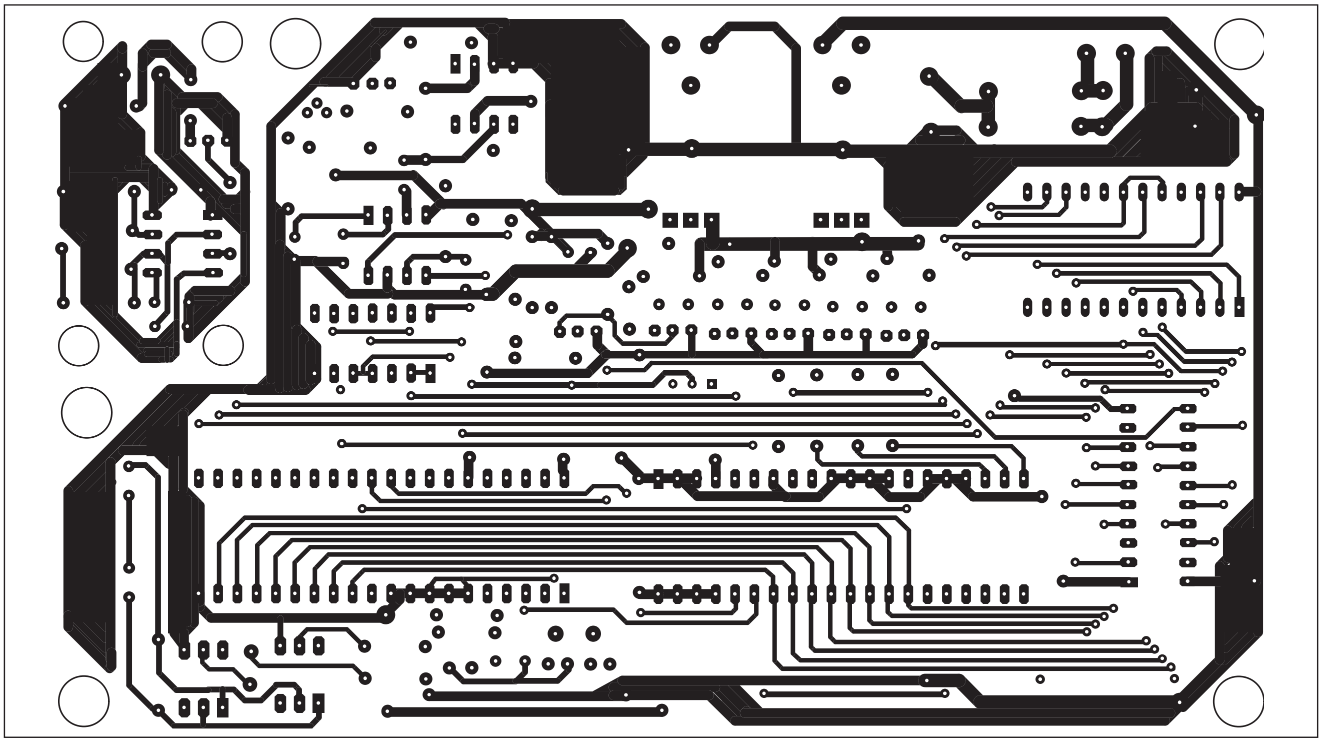


Fig. 5: Actual-size, solder-side PCB layout for the home security system

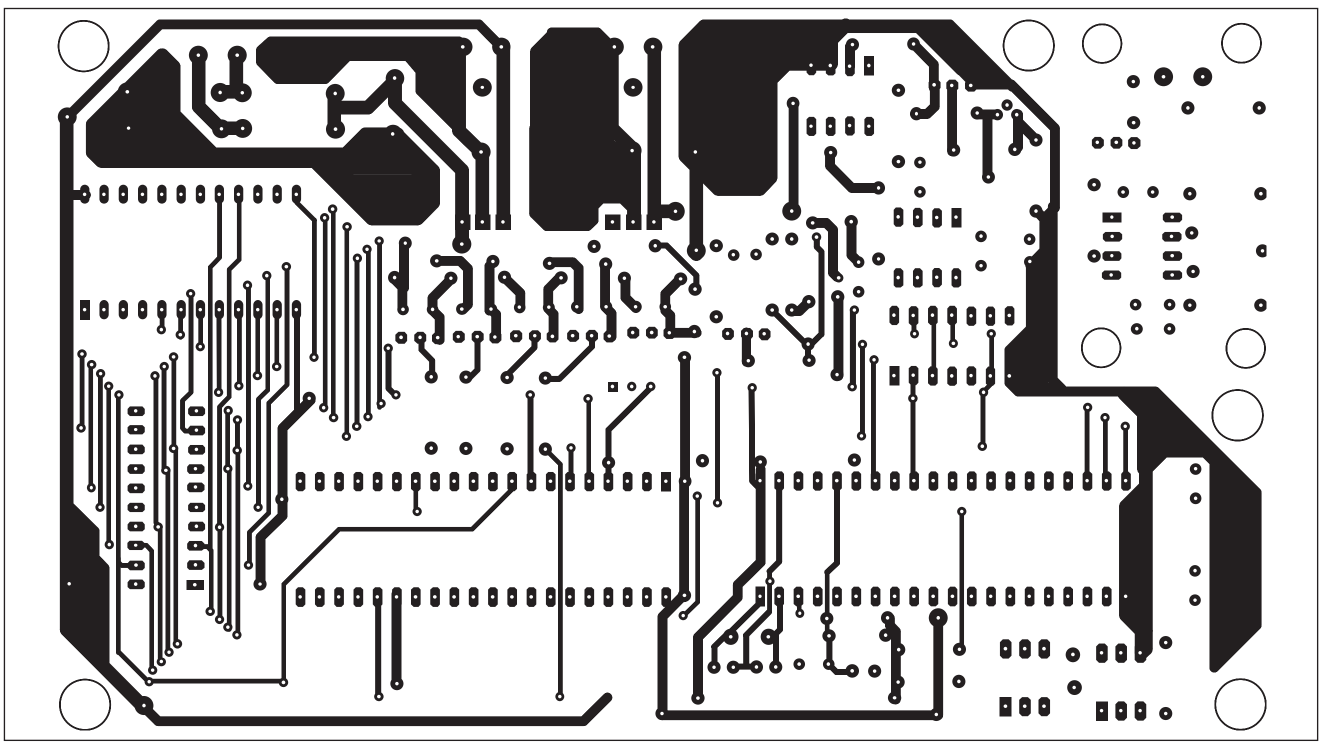


Fig. 6: Actual-size, component-side PCB layout for the home security system

is selected as 120 ohms to develop a voltage of 1.2V (which is adequate for an LED to turn on fully). When DC line voltage polarity reversal occurs, optocoupler IC8's internal LED conducts and LED3 glows to indicate polarity reversal. Simultaneously, optocoupler IC9's internal LED goes off and its pin 5 (collector) goes high to provide line-reversal sense signal to 8085 via pin 14 of 8255 PPI.

Fig. 3 shows the power supply circuit. The AC mains is stepped down by transformer X1 to deliver a secondary output of 12V AC at 300 mA. The transformer output is rectified by a full-wave bridge rectifier comprising diodes D7 through D10. Capacitor C12 acts as a filter to eliminate ripples. IC10 and IC11 provide regulated 5V and 9V power supplies, respectively. Capacitors C13 and C14 bypass any ripple present in the regulated outputs. Switch S2 acts as an 'on'/'off' switch.

Relay connections. The cradle switch in the telephone instrument is a double-

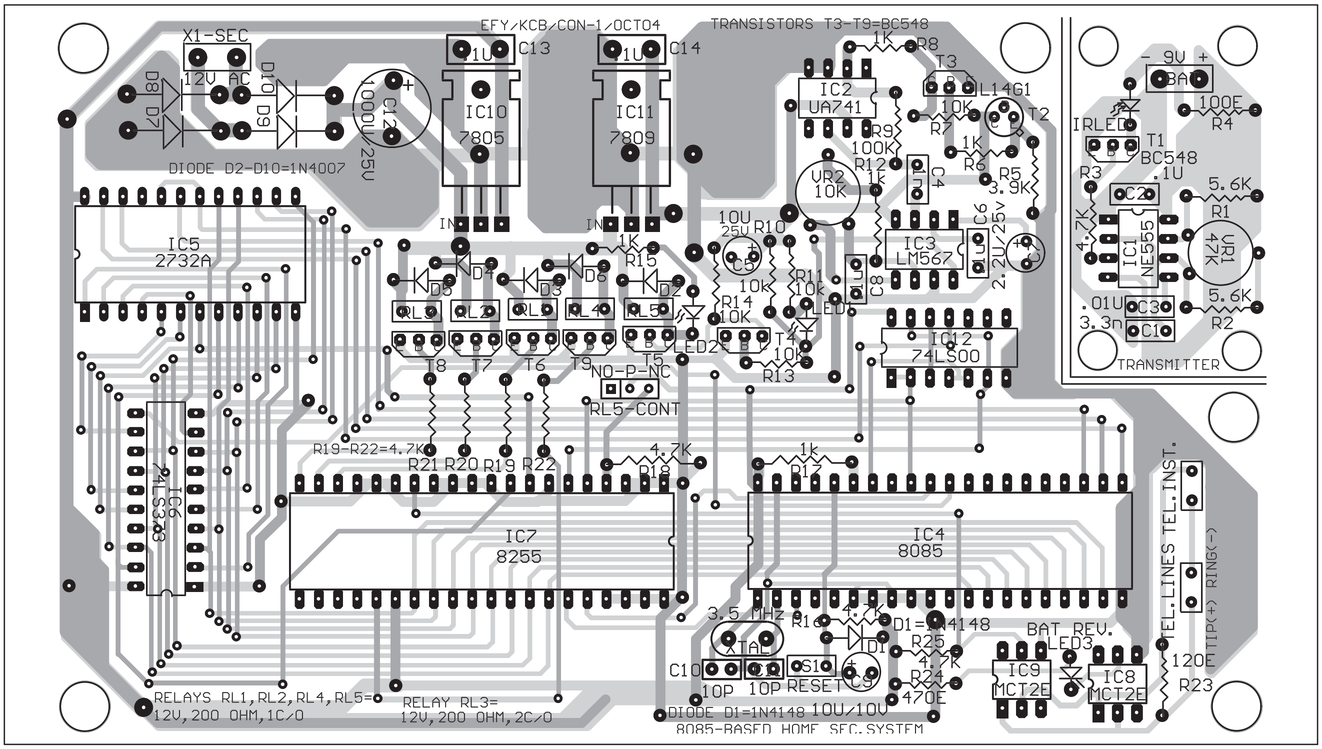


Fig. 7: Component layout for the PCB

pole, two-way switch. Replace this cradle switch with the contacts of DPDT relay RL3 as shown in Fig. 2. Now relay RL3 is used to implement the action of lifting the telephone handset.

There are four pads on the PCB of the telephone instrument where cradle switch is connected. The two pads which are shorted when the telephone handset is placed on the cradle are connected to the normally closed (N/C) contacts of relay RL3, while the other two pads which are shorted when the handset is off-hook are connected to the normally open (N/O) contacts of relay RL3.

Relay RL2 is connected in parallel to the redial button of the telephone instrument. When relay RL3 energises to emulate lifting of the handset, relay RL2 is energised to switch on the redial button and the already loaded telephone number of the police station or any other help provider is automatically dialled.

Relay RL4 activates the siren whenever the IR signal being received is interrupted. The siren sounds continuously until the user presses the reset button.

Relay RL1 is used to switch on the audio cassette player, in which the user's residential address and alert message to be conveyed to the police station are prerecorded. The speaker output of the cassette player is connected to the telephone's microphone to convey the alert message to the police station. The player gets switched off when the message is over.

Working of the circuit

The transmitting IR LED1 and phototransistor T2 of the receiver are fitted to the opposite pillars of the gate such that the IR rays emitted by the LED directly fall on the phototransistor.

The IR LED transmits a train of IR pulses. These pulses are received by the receiver and amplified by IC2. Output pin 8 of the PLL (IC3) is low when the PLL network is locked to the transmitter frequency and relay RL5 energises to make PA0 line of IC7 low.

When someone walks through the gate to enter your home, the transmitted signal is interrupted. Output pin 8 of the PLL network goes high and relay RL5 de-energies to make PA0 line of IC7 high. Now the microprocessor starts working as per the program loaded in the EPROM.

Relay RL4 energises to activate the siren. At the same time, relay RL3 energises to emutate lifting the telephone handset off the cradle to provide the dial tone. After a few seconds, relay RL2 energises to short the redial button contacts. After the loaded number is dialled, it switches off relay RL2. Then relay RL1 turns on the audio player.

Here we have provided the same polarity-reversal detection facility so that the audio player turns on only when polarity-reversal is detected.

The actual-size, double-side track layouts for solder and component sides of the PCB for the 8085 microprocessor-based home security system are shown in Figs 5 and 6, respectively, and their component layout in Fig. 7.

Software program

Fig. 4 shows the flow-chart of the Assembly language program. The device interface IC (IC7) is initialised with control word 99H. Ports A and C of IC7 act as input ports, while port B becomes the output port.

After initialisation, the 8085 microprocessor reads the status of port A. If port A is high, siren is activated. The telephone goes in off-hook condition and the emergency number is dialled through the redial button. Redial button gets switched off after the number is dialled. Now the microprocessor reads the status of port C and checks for the polarity reversal of the telephone line. When polarity reversal is detected, the audio player turns on to play the message. Otherwise, the process repeats from activation of the siren followed by emergency number dialling and so on. After delivering the message, the player automatically gets turned off. The siren sounds until the reset switch is pressed.

SECURITY.LST

```
2500 A.D. 8085 CROSS ASSEMBLER   -   VERSION 3.00b
--------------------------------------------------
     INPUT  FILENAME :  SECURITY.ASM
     OUTPUT FILENAME :  SECURITY.
0000                           ORG 0000H
0000   3E 99      MVI A,99H       ;Move control word to accumulator.
0002   D3 03      OUT 03H         ;O/P control word to control registor.
0004   DB 00      L1:IN 00H       ;Read port-A.
0006   FE 01      CPI 01H         ;Accumulator value compared with 01H
0008   C2 04 00   JNZ L1          ;Jump to L1 if it is not equal.
000B   3E 88      L2:MVI A,88H    ;Move 88H to accumulator.
000D   D3 01      OUT 01H         ;O/P the accumulator content to port-B (Siren ON).
000F   06 FF      MVI B,FFH;
0011   0E FF      LA:MVI C,FFH ;Delay Routine.
0013   0D         LB:DCR C
0014   C2 13 00   JNZ LB
0017   05         DCR B
0018   C2 11 00   JNZ LA
001B   3E 84      MVI A,84H       ;Move 84H to accumulator.
001D   D3 01      OUT 01H ;O/P the accumulator content to port-B.
001F   06 FF      MVI B,FFH
0021   0E FF      LAA:MVI C,FFH          ;Delay Routine.
0023   0D         LBB:DCR C
0024   C2 23 00   JNZ LBB
0027   05         DCR B
0028   C2 21 00   JNZ LAA
002B   3E 86      MVI A,86H   ;Move 86H to accumulator.
002D   D3 01      OUT 01H    ;O/P the accumulator content to port-B.
002F   11 FF FF   LXI D,FFFFH
0032   1B         LOOP1:DCX D
0033   7A         MOV A,D
0034   B3         ORA E
0035   C2 32 00   JNZ LOOP1
0038   3E 84      MVI A,84H    ;Move 84H to accumulator.
003A   D3 01 OUT 01H;O/P the accumulator content to port-B.
003C   06 40      MVI B,40H
003E   11 FF FF   SEC:LXI D,FFFFH
0041   DB 02      IN 02H
0043   FE 01      CPI 01H
0045   CA 59 00   JZ OFF
0048   1B         LOOP:DCX D
0049   7A         MOV A,D
004A   B3         ORA E
004B   C2 48 00   JNZ LOOP
004E   05         DCR B
004F   C2 3E 00   JNZ SEC
0052   DB 02      IN 02H    ;Read port-C.
0054   FE 01      CPI 01H ;Accumulator value compared with 01H
0056   C2 0B 00   JNZ L2             ;Jump to L2 if it is not equal.
0059   3E 85      OFF:MVI A,85H            ;Move 85H to accumulator.
005B   D3 01      OUT 01H            ;O/P the accumulator content to port-B.
005D   01 FF 01   LXI B,1FFH
0060   11 FF FF   SEC0:LXI D,FFFFH
0063   1B         LOOP0:DCX D
0064   7A         MOV A,D
0065   B3         ORA E
0066   C2 63 00   JNZ LOOP0
0069   0B         DCX B
006A   78         MOV A,B
006B   B1         ORA C
006C   C2 60 00   JNZ SEC0
006F   3E 80      MVI A,80H
0071   D3 01      OUT 01H
0073   11 FF FF   LP1: LXI D,FFFFH
0076   1B         LP:DCX D
0077   7A         MOV A,D
0078   B3         ORA E
0079   C2 76 00   JNZ LP
007C   C3 73 00                 JMP LP1
007F   76         HLT
0080                            END
************* S Y M B O L I C   R E F E R E N C E   T A B L E *************
L1  0004    L2     000B    LA     0011    LAA     0021
LB  0013    LBB    0023    LOOP   0048    LOOP0   0063
LOOP1       0032   LP      0076   LP1     0073    OFF     0059
SEC         003E   SEC0           0060
     LINES ASSEMBLED :   67          ASSEMBLY ERRORS :   0
```

SAFETY GUARD FOR THE BLIND

P. MURALI KUMAR

For the blind, it's difficult to step out without someone's help. To make the life simpler for them, here's an electronic safety guard system that alerts them of any obstacle or object in their path. The system can detect obstacles within one metre.

The system comprises transmitter and receiver sections (see Fig. 1). The receiver section uses an embedded system that tells the voice processor to play the recorded message in case any obstacle is detected. Here the embedded system is a microcontroller programmed to take the appropriate action.

System operation

The transmitter is built around timer IC 555, which is designed to operate at a frequency of 38 kHz. This signal is amplified by a current amplifier and transmitted through infrared (IR) diodes.

The receiver section consists of an IR

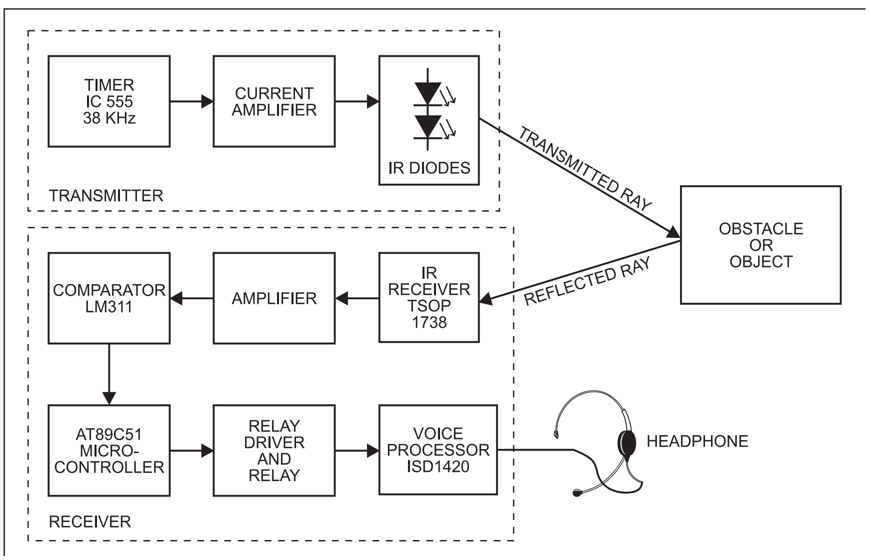


Fig. 1: Block diagram of the safety guard for the blind

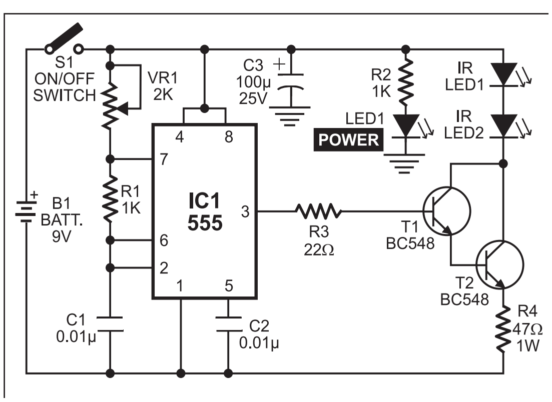


Fig. 2: Transmitter circuit

sensor TSOP1738, power amplifier, comparator IC LM311, microcontroller IC AT89C51, relay driver and voice processor IC ISD1420.

The IR rays reflected back from any obstacle are received by the IR receiver. The received signal is amplified by the amplifier stage, so even the weak signals can be picked up by the receiver. The amplified signal voltage is compared with a fixed threshold voltage at comparator LM311. The

PARTS LIST

Semiconductors:	
IC1	- 555 timer
IC2	- 7805 5V regulator
IC3	- LM311 comparator
IC4	- AT89C51 microcontroller chip
IC5	- ISD1420 voice processor
T1-T5, T7	- BC 548 npn transisor
T6	- BC558 pnp transisor
IR LED1, IR LED2	- Infrared LEDs (5mm dia.)
IRX1	- TSOP1738 IR receiver module
ZD1	- 2.2V, 1/4W zener diode
D1	- 1N4001 rectifier diode
LED1, LED2	- Red LED (5mm dia.)
LED3	- Green LED (5mm dia.)
LED4	- Yellow LED (5mm dia.)
LED5	- Red LED (5mm dia.)
Resistors (all ¼-watt, ±5% carbon):	
R1, R2, R9, R18, R19, R20, R21, R28	- 1-kilo-ohm
R3, R6, R11, R13	- 22-ohm
R4	- 47-ohm
R5, R7	- 100-ohm
R8	- 15-ohm
R10	- 68-ohm
R12	- 4.7-kilo-ohm
R14, R15, R16	- 470-ohm
R17a	- 620-ohm
R22, R23, R17b	- 100-kilo-ohm
R24	- 5.1-kilo-ohm
R25	- 470-kilo-ohm
R26, R27	- 10-kilo-ohm
VR1	- 2-kilo-ohm preset
VR2	- 4.7-kilo-ohm preset
Capacitors:	
C1, C2	- 0.01µF ceramic disk
C3, C4	- 100µF, 25V electrolytic
C5, C6	- 33pF ceramic disk
C7	- 0.001µF ceramic disk
C8, C14	- 4.7µF, 16V electrolytic
C9, C10, C11, C13	- 0.1µF ceramic disk
C12	- 220µF, 16V electrolytic
Miscellaneous:	
B1, B2	- 9V battery
X_{TAL1}	- 3.579MHz crystal oscillator
S1, S2	- On/off SPST switch
S3, S4	- Push-to-on tactile switch
MIC	- Condensor microphone
JACK1	- Jack for headphone connector
RL1	- 5V, 100-ohm, single-changeover relay

microcontroller.

When the comparator output goes high because of reflection of signal from an object, the microcontroller energises a relay via the relay driver. The relay contacts are used by a voice processor to play a prerecorded warning message (such as "hey, there's an obstacle"). The user can hear the played message using a headphone.

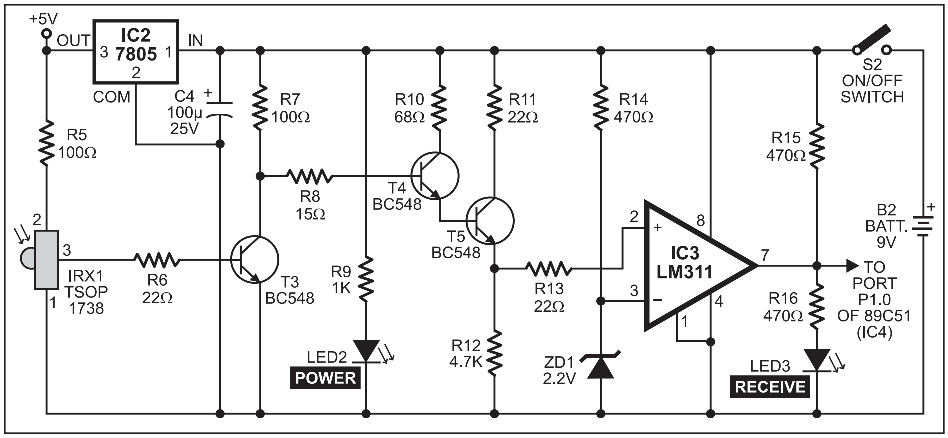


Fig. 3: Receiver circuit

Circuit description

Transmitter section. Fig. 2 shows the transmitter circuit powered by a 9V battery. When switch S1 is closed, LED1 glows to indicate the presence of power in the circuit. Timer IC 555 (IC1) is wired as an astable multivibrator. The output frequency (38 kHz) of IC1 at its pin 3 can be varied using VR1 (2k). The output of IC1 is given to the base of npn transistor T1 (BC548) via resistor R3. Transistors T1 and T2 (each BC548) form a Darlington pair that boosts the output current to drive the two infrared diodes connected in series at the collector of the Darlington pair (IR LED1 and IR LED2). The output signal frequency of 38 kHz is transmitted by the IR LEDs.

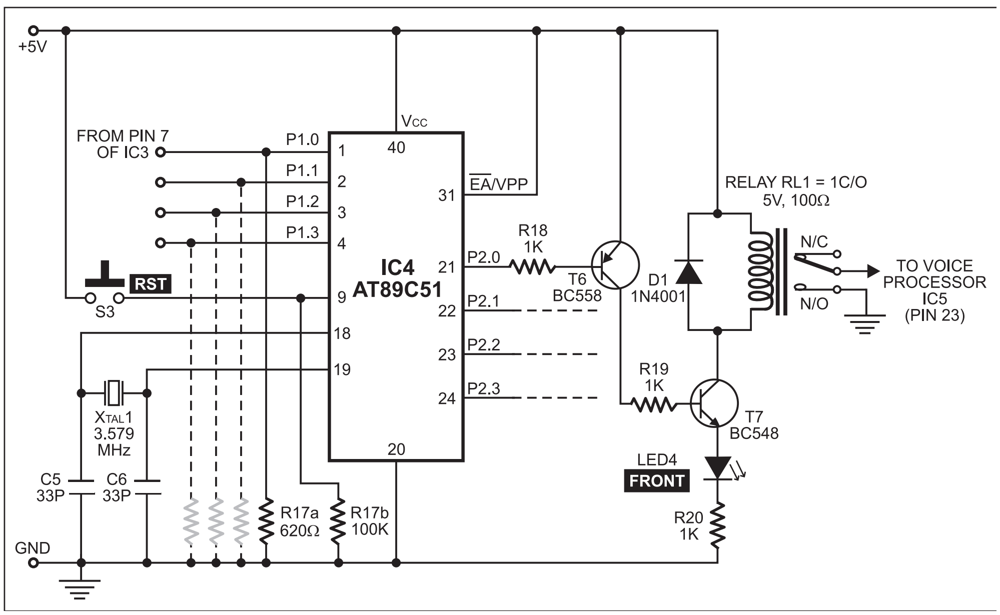


Fig. 4: Connections for the microcontroller

Receiver section. Fig. 3 shows the receiver circuit powered by a 9V battery. When switch S2 is closed, LED2 glows to indicate the presence of power in the circuit. The 9V supply is down-converted to 5V using regulator IC 7805 (IC2) to drive the IR receiver module (TSOP1738), microcontroller and voice processor sections.

The IR rays reflected from any object in the path of the user are received by the IR receiver module. This signal is amplified by the power amplifier stage comprising transistors T3, T4 and T5 (each BC548). The amplified output at the emitter of transistor T5 is given to the non-inverting input (pin 2) of comparator IC LM311 (IC3) through resistor R13.

A reference voltage of 2.2V developed across zener diode ZD1 is connected to the inverting input (pin 3) of IC3. When the voltage level at pin 2 increases beyond the reference voltage, output pin 7 of IC3 goes high, which is indicated by the glowing of LED3. This output is given to the I/O port P1.0 of microcontroller IC4.

Microcontroller section. Microcontroller chip AT89C51 (IC4) acts as a switching hub only and can be replaced by any other switching circuit. The use of this chip in this circuit is to show how to interface an embedded system in a home made project. The program burnt into this chip decides the action when a signal is received at its input.

As shown in Fig. 4, ports P1.0 through P1.3 of IC4 are used as the input ports. The corresponding outputs are available at ports P2.0 through P2.3. The output of the comparator is fed to port P1.0 and the corresponding output at port P2.0 is fed to the base of transistor T6 (BC558) through resistor R18.

Normally, when no signal is applied at input port P1.0, output port P2.0 is high. When input P1.0 becomes high, output P2.0 goes low and transistor T6 conducts. This, in turn, drives transistor T7 (BC548) to energise relay RL1, which is indicated by glowing of LED4. In case the circuit behaves abnormally, press reset switch S3 momentarily to reset the circuit.

Voice processor. The voice processor section receives regulated 5V DC supply from regulator IC2. Voice processor IC ISD1420 (IC5) used here is a 28-pin chip from Winbond. It can record a voice message up to 20 seconds long. The recorded message can be played at the press of a button connected to one of its pins.

As shown in Fig. 5, pushbutton switch S4 connected to pin 27 of IC5 is used for recording the message in the processor. Pin 23 is used for playing the recorded message. The condenser microphone for inputing the voice message is connected to pins 17 and 18 of IC5 via capacitors C13 and C10, respectively. The message is output via pins 14 and 15. A loudspeaker or headphone can be directly connected to these pins through a coupling capacitor. Here, we've used an output jack (JACK1) at these pins for headphone connection. Preset VR2 is used to control the volume and C14 acts as a coupling capacitor.

Keep switch S4 pressed (maximum for 20 seconds) as you speak into the microphone for recording the message. Release switch S4 after recording is done. To listen to the recorded message through the speaker or the headphone, playback pin 23 ($\overline{\text{PLAYL}}$) must be held down to ground. Here, energisation of relay RL1 pulls pin 23 to ground and thus enables playback of the recorded message.

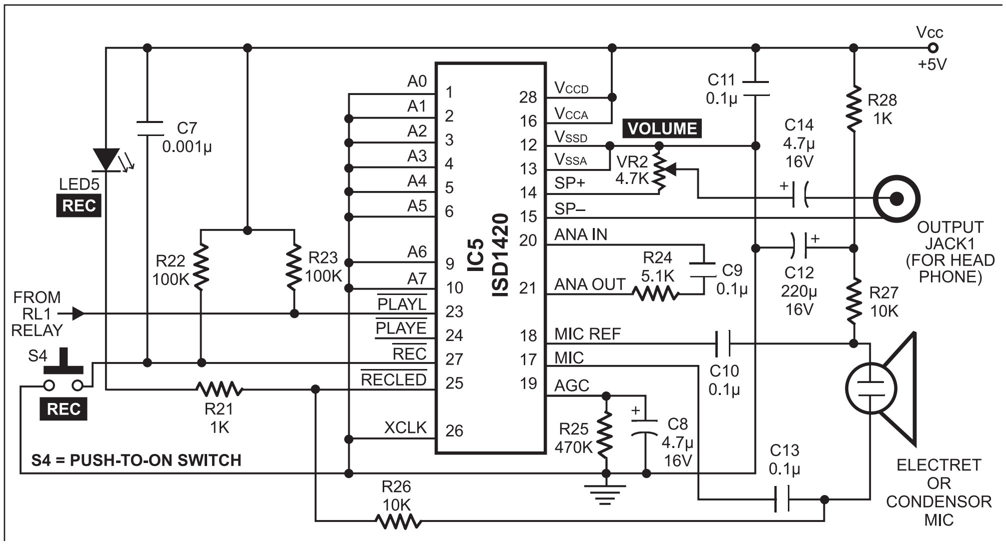


Fig. 5: Single-chip record/playback circuit using ISD1420 chip

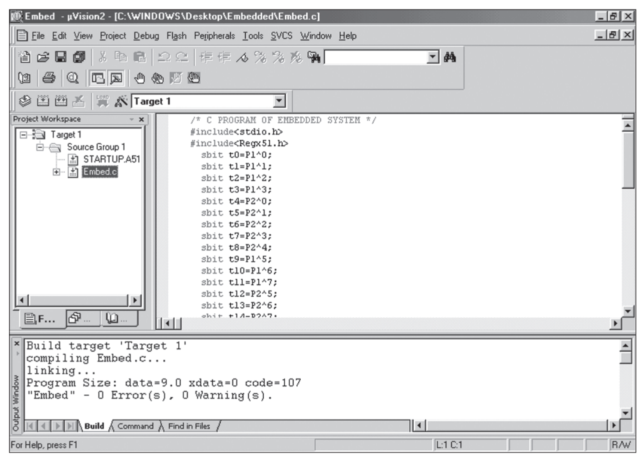
The pole of relay RL1 is connected to pin 23 of IC5, while the normally open (N/O) contact is grounded. When relay RL1 energises, the pole of the relay connects to the N/O contact enabling the voice processor to play the recorded message and the message can be heard from the headphone.

Software program

Written in C language, the software program (Embed.c) for the microcontroller is simple and easy to understand. You don't have to write long Assembly language program for this operation. The program is converted into Intel-Hex format for loading to the microcontroller. Here, we've used cross-compiler C51 version 7.10 from Keil Software for conversion. The demo version of this compiler is available for free on the Website 'www.keil.com.'

The C program includes '<Regx51.h>' file, which defines pseudo-variables to interact with memory-mapped devices and I/O ports of the microcontroller. The I/O ports P1.0 through P2.7 of AT89C51 are defined in the program. The program line "{if(t0==0) { t4=1;" instructs the microcontroller that if its input port P1.0 is low, its output port P2.0 should be high. Otherwise, P2.0 must be low.

The listing file (Embed.lst) generated by Keil compiler is given at the end of this article. Note that this listing file cannot be recompiled in any other assembler.



Screenshot of editing window

Program compilation

After installing Keil C51 in your system, you can compile C program and generate hex file in either DOS or Windows mode. For DOS-mode operation, refer to the 'Temperature Indicator Using AT89C52' article published in July issue. Every time you create a new program in Keil C51 version 7.10, you must create a project file with '.uv2' extension. Then write the program in 'Edit' window, compile it and link it. (The compiled program with '.hex' and '.lst' files has been included in CD.)

If you want to create your own program, the steps for Windows mode are:

1. Install Keil software in 'C' drive. After installation, 'Keil uVision2' icon is automatically created on the desktop.

2. Double-click 'Keil uVision2' on the desktop.

3. Suppose you have kept 'Embed.c' under 'C:\Windows\Desktop\Embedded' folder. Open 'Embed.c' from 'File' menu.

4. From 'Project' menu bar, select 'New Project.' Name the new project and save it with '.uv2' extension.

5. Select CPU as 'Atmel/AT89C51.'

6. Choose 'Yes' in the option 'Copy standard 8051 code to current project folder.'

7. From 'View' menu, select 'Project Window.' 'Project Workspace' window appears on the left-hand side of the PC screen.

8. Double-click 'Target 1.'

9. Right-click 'Source Group1' and select 'Add files to Group 'Source Group1." A window appears.

10. Add 'Embed.c' and close this window.

11. Double-click 'Source Group1' in 'Project Workspace' window to get 'Embed.c' (see the screenshot).

12. Right-click 'Embed.c' in 'Project' window, select 'Options for File Embed.c' and choose 'File Type' as 'C source file' under 'Properties.'

13. Again from 'Project' menu, select 'Options for Target 1.' An 'Option' screen appears.

14. Choose 'Output' and tick 'Hex File'

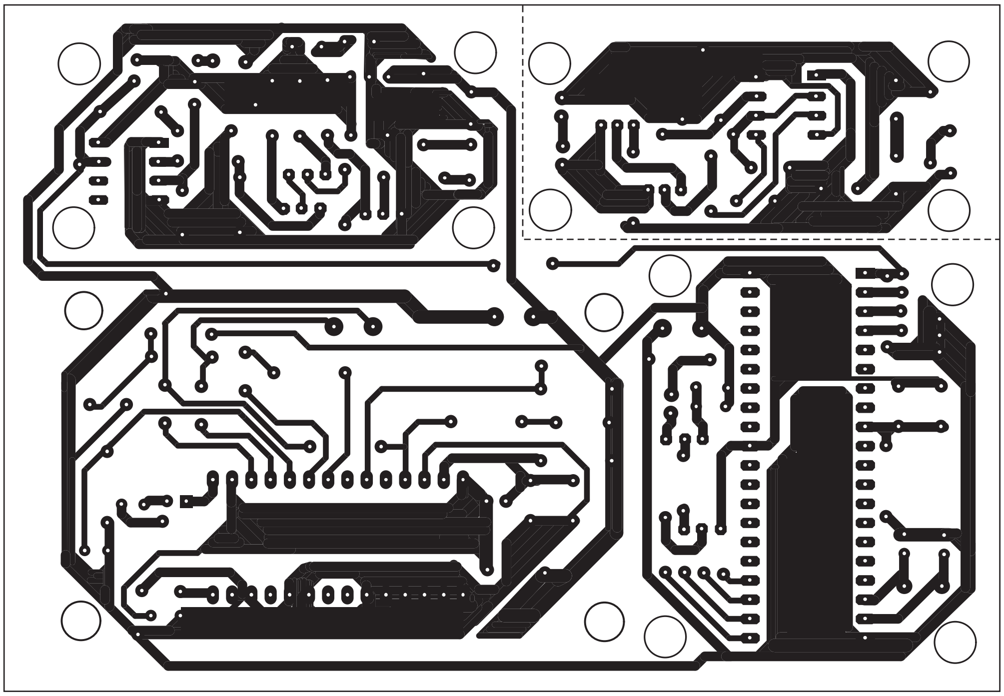


Fig. 6: Actual-size, single-side PCB for transmitter and receiver circuits of the safety guard

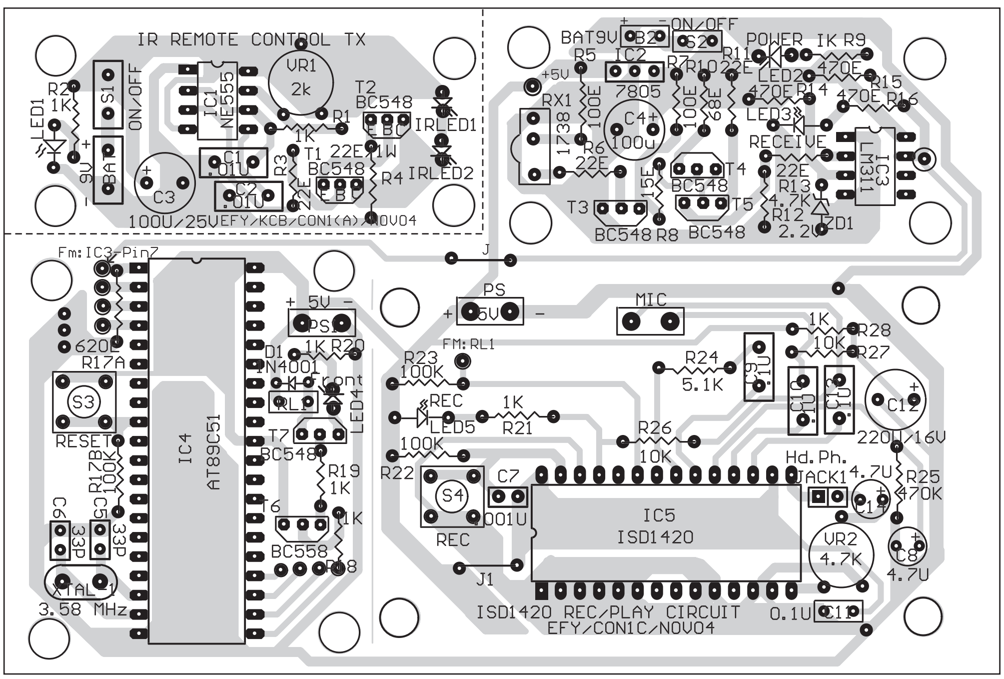


Fig. 7: Component layout for the PCB

for generating the hex file. Again choose 'Listing' option and tick 'Conditional' and 'Assembly Code.'

15. Open 'Project' menu and select 'Build Target' or press 'F7' key.

For any syntax error, the window will contain a list of errors with the line numbers. Double-clicking an error message will cause the corresponding line in the source edit window to be highlighted. Correct errors, if any, and press 'F7' key again.

If compilation is successful, an object file with '.obj' extension is created for the source file. The compiler shows '"Embed"-0 Error(s), 0 Warning(s)' in the output window just below the project window. It also produces a listing file with '.lst' extension for the source file. Then the system links all the generated files into a combined hex file (with '.hex' extension) suitable for loading into the chip. In this example, the files created are 'startup.obj,' 'startup.lst,' 'Embed.obj,' 'Embed.lst' and 'Embed.hex.'

16. Close the screen and go to 'Embedded' folder to see the generated hex and listing files.

17. Load the hex file into the microcontroller chip using a programmer.

We've used Atmel Flash Programmer V.1 from Frontline Electronics. It is a serial port programmer. The procedure for loading the hex file into Atmel Flash programmer is given below:

1. Double-click the icon of the programmer.

2. Select the appropriate COM port from 'Settings' menu bar.

3. Select '89C51' from 'Selection' option in 'Device' menu bar.

4. Load the hex file from 'File' menu.

5. Choose 'Auto' in 'Device' menu bar. This will automatically erase the previous program, if any, and load the new program into the chip.

Construction and working

The transmitter and receiver circuits can be assembled on separate general-purpose PCBs. Both the circuits must be separated or covered by some opaque, non-conducting material so that no stray signal from the transmitter falls on the receiver. The units can be carried in a bag, with the IR transmitting LEDs and the receiver sensor (IRX1) mounted on the front side of the user's belt by extending their leads using shielded wires.

This system uses only one pair of transmitter and receiver circuits for alerting against the obstacles in the path of the user, but it can be extended to use three more pairs for detection of objects on the right, left and back side. As shown in Fig. 4, input ports P1.1, P1.2 and P1.3 of AT89C51 and their corresponding output ports P2.1, P2.2 and P2.3 are left unused. These ports can be used for detection of objects on the right, left and back side. Separate pairs of transmitter and receiver circuits are required for all the sides.

If all the eight ports are to be used, use the same circuits as used at ports P1.0 and P2.0 for all the ports (shown by dotted lines in Fig. 4). Apart from adding these circuits to the microcontroller section, you also need to use the same but separate transmitter (Fig. 2), receiver (Fig. 3) and voice processor circuits (Fig. 5) for each input-output port combination. (Separate voice processor is not required if you make use of the multiple-message record/play capability of IC APR9600 as in 'Voice Recording and Playback Using APR9600 Chip' construction project published in EFY's September '04 issue. Relays can also be replaced by transistor switches.)

The combined actual-size, single-side PCB for the transmitter and receiver circuits is shown in Fig. 6 and its component layouts in Fig. 7. The transmitter and receiver PCBs can be separated by cutting the PCB along the dotted lines.

EFY note. The source code and other relevant files of this project have been included in CD.

EMBED.C

```
/* C PROGRAM OF EMBEDDED SYSTEM */
#include<stdio.h>
#include<Regx51.h>
sbit t0=P1^0;
sbit t1=P1^1;
sbit t2=P1^2;
sbit t3=P1^3;
sbit t4=P2^0;
sbit t5=P2^1;
sbit t6=P2^2;
sbit t7=P2^3;
sbit t8=P2^4;
sbit t9=P1^5;
sbit t10=P1^6;
sbit t11=P1^7;
sbit t12=P2^5;
sbit t13=P2^6;
sbit t14=P2^7;
void main()
  {
t9=t10=t11=t12=t13=t14=0;
t0=1; t1=1;t2=1;t3=1;//t3=t2=t1=t0=1;
 for(; ;)
   {
l1: if(t0==0)
     {
    t4=1;
t5=t6=t7=t8=0;
 goto l1;
     }
l2:if(t1==0)
     {
    t5=1;
t4=t6=t7=t8=0;
 goto l2;
   }
 l3:if(t2==0)
       {
     t6=1;
 t4=t5=t7=t8=0;
     goto l3;
         }
   l4:if(t3==0)
       {
     t7=1;
 t4=t5=t6=t8=0;
     goto l4;
       }
     t8=1;
 t4=t5=t6=t7=0;
               }
               }
```

EMBED.LST

```
C51 COMPILER V7.10   EMBED                    10/11/2004 10:52:09 PAGE 1


C51 COMPILER V7.10, COMPILATION OF MODULE EMBED
OBJECT MODULE PLACED IN embed.OBJ
COMPILER INVOKED BY: C:\KEIL\C51\BIN\C51.EXE embed.c BROWSE
DEBUG OBJECTEXTEND CODE

line level    source

1              /* C PROGRAM OF EMBEDDED SYSTEM */
2              #include<stdio.h>
3              #include<Regx51.h>
4              sbit t0=P1^0;
5              sbit t1=P1^1;
6              sbit t2=P1^2;
7              sbit t3=P1^3;
8              sbit t4=P2^0;
9              sbit t5=P2^1;
10             sbit t6=P2^2;
11             sbit t7=P2^3;
12             sbit t8=P2^4;
13             sbit t9=P1^5;
14             sbit t10=P1^6;
15             sbit t11=P1^7;
16             sbit t12=P2^5;
17             sbit t13=P2^6;
18             sbit t14=P2^7;
19             void main()
20             {
21    1        t9=t10=t11=t12=t13=t14=0;
22    1        t0=1; t1=1;t2=1;t3=1;//t3=t2=t1=t0=1;
23    1         for(; ;)
24    1            {
25    2        l1: if(t0==0)
26    2              {
27    3             t4=1;
28    3        t5=t6=t7=t8=0;
29    3         goto l1;
30    3              }
31    2        l2:if(t1==0)
32    2              {
33    3             t5=1;
34    3        t4=t6=t7=t8=0;
35    3         goto l2;
36    3            }
37    2          l3:if(t2==0)
38    2                {
39    3              t6=1;
40    3          t4=t5=t7=t8=0;
41    3              goto l3;
42    3                  }
43    2            l4:if(t3==0)
44    2                {
45    3              t7=1;
46    3          t4=t5=t6=t8=0;
47    3              goto l4;
48    3                }
49    2              t8=1;
50    2          t4=t5=t6=t7=0;
51    2                        }
52    1                        }

ASSEMBLY LISTING OF GENERATED OBJECT CODE

        ; FUNCTION main (BEGIN)
                                        ; SOURCE LINE # 19
                                        ; SOURCE LINE # 20
```

```
                                        ; SOURCE LINE # 21
0000  C2A7     CLR   t14
0002  C2A6     CLR   t13
0004  C2A5     CLR   t12
0006  C297     CLR   t11
0008  C296     CLR   t10
000A  C295     CLR   t9
                                        ; SOURCE LINE # 22
000C  D290     SETB  t0
000E  D291     SETB  t1
0010  D292     SETB  t2
0012  D293     SETB  t3
                                        ; SOURCE LINE # 23
                                        ; SOURCE LINE # 24
                                        ; SOURCE LINE # 25
0014           l1:
0014  20900C   JB    t0,l2
                                        ; SOURCE LINE # 26
                                        ; SOURCE LINE # 27
0017  D2A0     SETB  t4
                                        SOURCE LINE # 28
0019  C2A4     CLR   t8
001B  C2A3     CLR   t7
001D  C2A2     CLR   t6
001F  C2A1     CLR   t5
                                        ; SOURCE LINE # 29
0021  80F1     SJMP  l1
                                        ; SOURCE LINE # 30
                                        ; SOURCE LINE # 31
0023           l2:
0023  20910C   JB    t1,l3
                                        ; SOURCE LINE # 32
                                        ; SOURCE LINE # 33
0026  D2A1                              SETB    t5
                                        ; SOURCE LINE # 34
0028  C2A4     CLR   t8
002A  C2A3     CLR   t7
002C  C2A2     CLR   t6
002E  C2A0     CLR   t4
                                        ; SOURCE LINE # 35
0030  80F1     SJMP  l2
                                        ; SOURCE LINE # 36
                                        ; SOURCE LINE # 37
0032           l3:
0032  20920C   JB    t2,l4
                                        ; SOURCE LINE # 38
                                        ; SOURCE LINE # 39
0035  D2A2     SETB  t6
                                        ; SOURCE LINE # 40
0037  C2A4     CLR   t8
0039  C2A3     CLR   t7
003B  C2A1     CLR   t5
003D  C2A0     CLR   t4
                                        ; SOURCE LINE # 41
003F  80F1     SJMP  l3
                                        ; SOURCE LINE # 42
                                        ; SOURCE LINE # 43
0041           l4:
0041  20930C   JB    t3,?C0010
                                        ; SOURCE LINE # 44
                                        ; SOURCE LINE # 45
0044  D2A3     SETB  t7
                                        ; SOURCE LINE # 46
0046  C2A4     CLR   t8
0048  C2A2     CLR   t6
004A  C2A1     CLR   t5
004C  C2A0     CLR   t4
                                        ; SOURCE LINE # 47
004E  80F1     SJMP  l4
                                        ; SOURCE LINE # 48
0050        ?C0010:
                                        ; SOURCE LINE # 49
0050  D2A4     SETB  t8
                                        ; SOURCE LINE # 50
0052  C2A3     CLR   t7
0054  C2A2     CLR   t6
0056  C2A1     CLR   t5
0058  C2A0     CLR   t4
                                        ; SOURCE LINE # 51
005A  80B8     SJMP  l1
                                        ; FUNCTION main (END)

MODULE INFORMATION:   STATIC OVERLAYABLE
   CODE SIZE        =    92      ----
   CONSTANT SIZE    =    ----    ----
   XDATA SIZE       =    ----    ----
   PDATA SIZE       =    ----    ----
   DATA SIZE        =    ----    ----
   IDATA SIZE       =    ----    ----
   BIT SIZE         =    ----    ----
END OF MODULE INFORMATION.

C51 COMPILATION COMPLETE.  0 WARNING(S),  0 ERROR(S)
```

❑

DIGITAL COMBINATION LOCK

SREEKUMAR V.

We've seen in movies highly secured dens that require one to press certain number combination to gain entry. These locking systems use expensive microprocessors and PCs, which a common man can't afford.

Here is a digital combination lock using solidstate memory ICs that costs much less. As shown in Fig. 1, the system uses two key sets (user and security key sets), D-type flip-flops, comparators and solenoid. The user code comprising eight bits is compared with the preset security code of the same length (eight bits). If the user code matches with the security code, access is granted for opening the code lock by pressing an 'Enter' key. The lock can be closed/reset by using the reset key.

Circuit description

Fig. 2 shows the power supply circuit for the lock. The AC mains is stepped down by transformer X1 to deliver a secondary output of 9V AC at 300 mA. The transformer output is rectified by a full-wave bridge rectifier comprising diodes D1 through D4. Capacitor C2 acts as a filter to eliminate ripples. Regulator IC 7805 (IC9) provides regulated 5V power supply to the circuit.

Fig. 3 shows the circuit of the digital combination lock. The user key set comprising switches is connected to D-type flip-flop 74LS74 ICs (IC1 through IC4), which act as the storage devices for the sequence entered by pressing push-to-on tactile switches S1 through S8.

Pressing any of the user keys results in logic 1 to be clocked to the 'Q' output of the respective flip-flop of IC 74LS74. Else, the 'Q' outputs of the flip-flops of IC1 through IC4 remain at logic 0. The outputs of IC1 through IC4 are fed to 'A' inputs of two 4-bit magnitude comparator 74LS85 ICs (IC5 and IC6). The 'B' inputs of IC5 and IC6 are connected to the security key set (S9 through S16). Output pin 6 of IC5 ($O_{A=B}$) and input pin 3 of IC6 ($I_{A=B}$) are cascaded to obtain the 8-bit sequence.

Output pin 6 of comparator IC5 goes high if the input bit sequence is the same as the preset bit sequence, i.e. 'A3A2A1A0' is equal to 'B3B2B1 B0.' Similarly, output pin 6 of comparator IC6 goes high if the input bit sequence is the same as the preset bit sequence, i.e. 'A3A2A1A0' is equal to 'B3B2B1B0.' The high output of IC6 is fed to flip-flop IC7. Pressing 'Enter' key (S17) causes a clock transition at the input of IC7 and its Q1 output (pin 5) goes high. As a result, transistor T1 conducts and relay RL1 energises. At the same time, the solenoid connected to the relay contacts moves back to unlock the door.

In case the user input bit sequence doesn't match with the preset security bit sequence, the output of IC6 remains low and therefore pressing 'Enter' key doesn't activate the relay driver transistor. Consequently, the solenoid doesn't move back to unlock the door.

Solenoid connections are shown in Fig. 4. Driving the solenoid with DC is very simple. Just switch on the DC supply to it using a relay or transistor, and the solenoid operates. However, when the solenoid is driven, flywheel diodes are necessary. The large inductance of the coil can cause large voltage spikes to appear across the switching element (relay or

PARTS LIST

Semiconductors:	
IC1-IC4, IC7	- 74LS74 dual D-type flip-flop
IC5, IC6	- 74LS85 4-bit magnitude comparator
IC8	- 74LS00 quad 2-input NAND gate
IC9	- 7805 5V regulator
LED1	- 5mm red LED
LED2	- 5mm green LED
T1	- SL100 npn transistor
D1-D6	- 1N4001 rectifier diode
Resistors (all ¼-watt, ±5% carbon):	
R1-R10, R13	- 1-kilo-ohm
R11, R12	- 220-ohm
R14	- 2.2-kilo-ohm
Capacitors:	
C1	- 100µF, 16V electrolytic
C2	- 1000µF, 25V electrolytic
C3	- 0.1µF ceramic disk
Miscellaneous:	
X1	- 230V, AC primary to 9V AC, 300mA secondary transformer
S1-S8, S17, S18	- Push-to-on tactile switch
S9-S16	- SPDT switch
S19	- On/off switch
PZ1	- Piezobuzzer
RL1	- 5V, 200-ohm 1C/O relay
	- Solenoid or equivalent

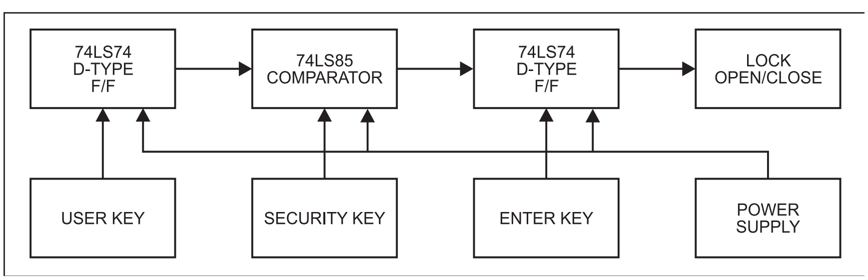


Fig. 1: Block diagram of digital combination lock

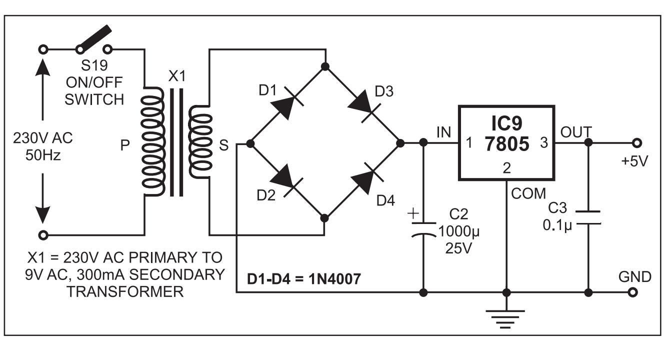


Fig. 2: Power supply circuit

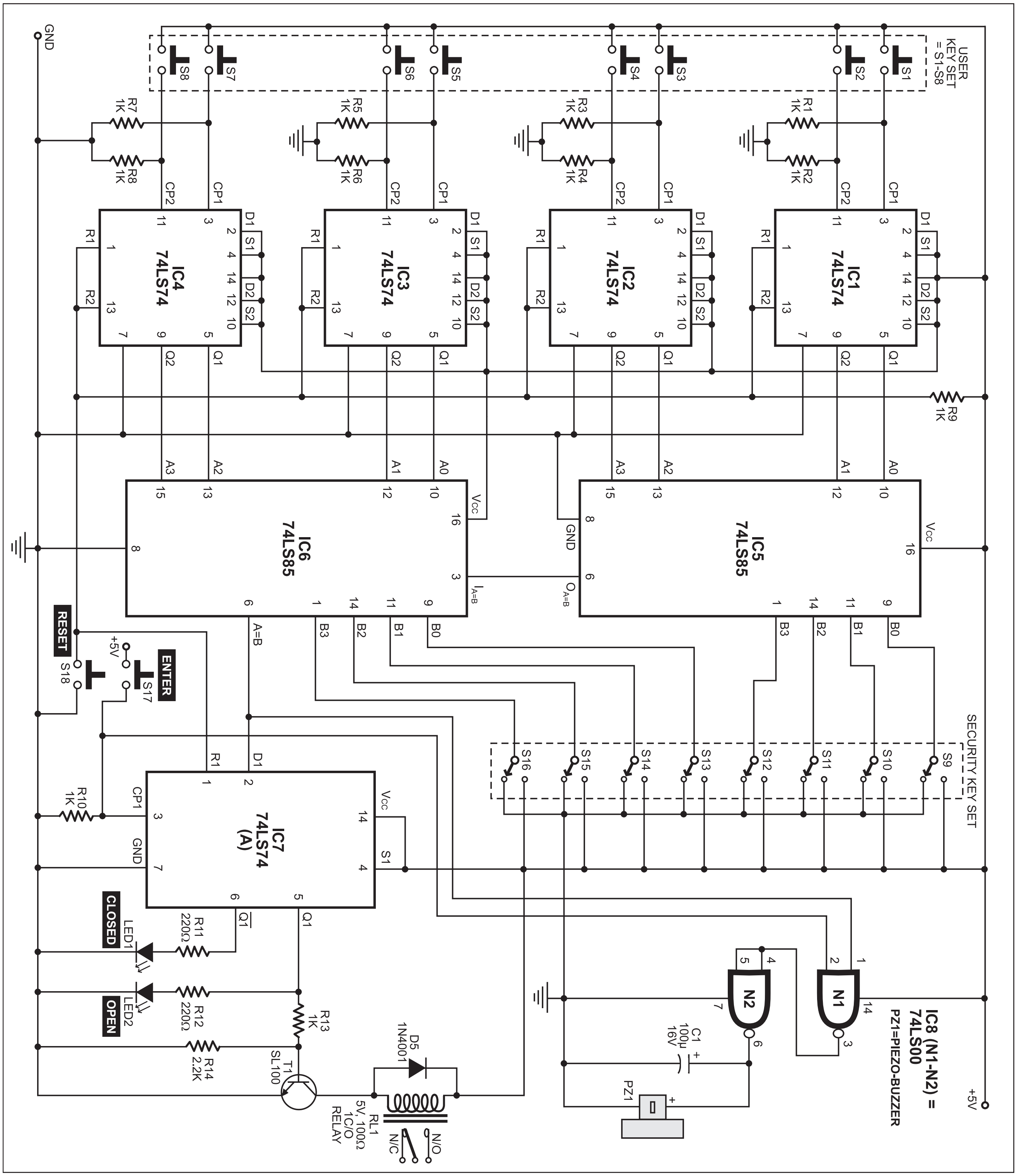


Fig. 3: Circuit of the digital combination lock

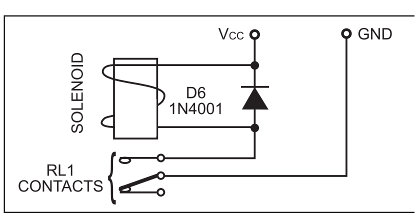


Fig. 4: Solenoid connections

transistor doing the switching), unless the current flowing through the coil is allowed to dissipate slowly.

When relay RL1 energises, the current flowing down through the solenoid coil is limited by the resistance of the coil. The inductors tend to oppose the quick change in the current flowing through them and generate a voltage of their own to stop this happening. When relay contacts open, the inductor generates a voltage to make the current to continue down through the coil, and the current flows up through the diode and back into the inductor. This is the reason why a freewheeling diode (D6) is

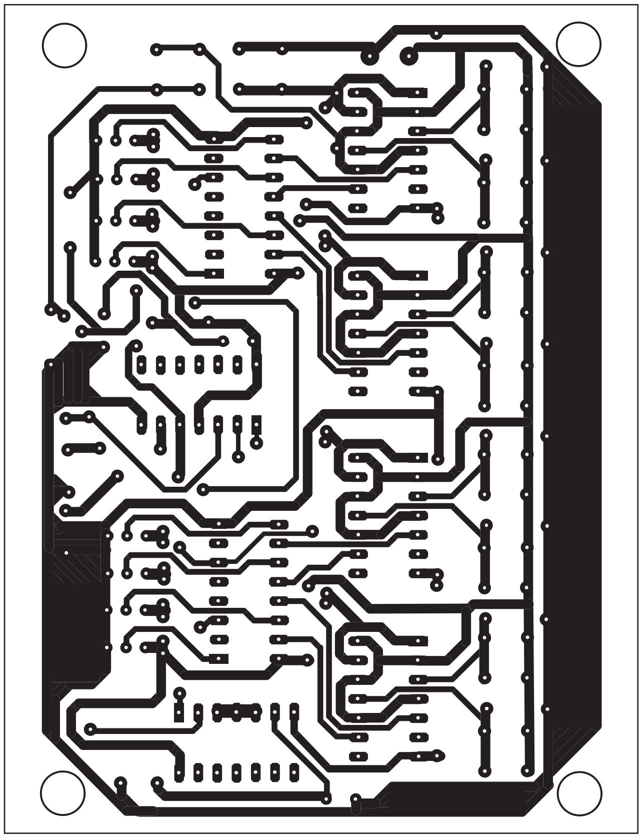


Fig. 5: Actual-size, single-side PCB for digital combination lock

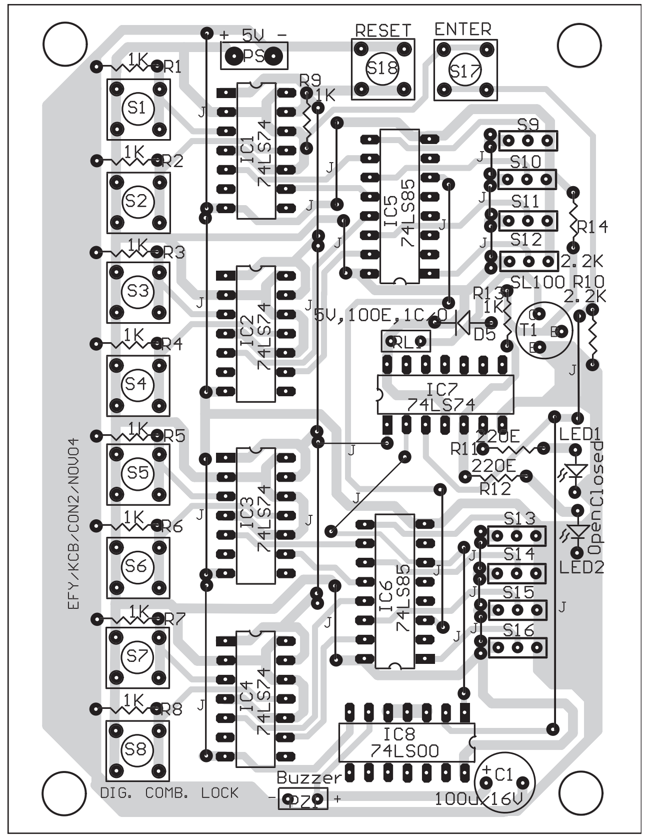


Fig. 6: Component layout for the PCB

used here. The logic built around NAND gates N1 and N2 enables the buzzer when the sequence matches and 'Enter' key is pressed. Capacitor C1 prolongs the buzzer sound.

Operation

This circuit is designed for 8-digit binary codes and can be divided into two parts, namely, the user key panel and the security key panel. Switches S1 through S8 shown within the rectangular dotted lines form the user key panel. Similarly, switches S9 through S16 shown within another rectangular dotted lines form the security key panel.

Suppose you want to set the password as '1578.' For this, connect the first switch (S9), fifth switch (S13), seventh switch (S15) and eighth switch (S16) of the security key set to +5V and ground all the remaining switches. To open the lock, you'll have to momentarily press the first switch (S1), fifth switch (S5), seventh switch (S7) and eighth switch (S16) of the user key set to match with the preset code in the security key set and then press 'Enter' key (S17).

If the entered sequence matches with the preset sequence, the buzzer sounds to indicate the correct entry and LED2 glows to indicate that the lock has opened. If the sequence doesn't match, the buzzer doesn't sound and LED1 glows to indicate that the door is not opening. For the next trial, press reset key S18.

Pressing 'Enter' key obviates fooling of the system by random entries when someone is trying to open the lock. With eight digits, up to 2^8 combinations are possible, which makes it very difficult for a person to keep on trying by pressing 'Enter' every time. After each entry, reset switch S18 should also be pressed to clear all the flip-flops (IC1 through IC4 and IC7).

Fabrication

An actual-size, single-side PCB for the digital combination lock (including the user and security key sets) is shown in Fig. 5 and its component layout in Fig. 6. If you want to install the user and security key sets away from the gate, you can separate them from the main circuit by using extended wires. An electromechanical device such as relay, magnetic bell or solenoid can be used to open the lock. The power supply circuit can be easily wired on a separate general-purpose PCB.

Precautions

1. Use a TTL logic gate such as 74LS74, 74LS85 or 74LS00 to minimise power consumption.

2. The solenoid must move smoothly to lock and unlock.

3. Check the security key terminals using multimeter before connecting into the PCB board. ❑

ULTRASONIC LAMP-BRIGHTNESS CONTROLLER

PRADEEP G.

Here is a low-cost, wireless lamp-brightness controller. It uses ultrasonic sound waves for remote control of the lamp's brightness.

As with any other remote control, the system basically comprises a transmitter and a receiver circuit. Frequencies above 20 kHz are inaudible (ultrasonic). The transmitter circuit generates ultrasonic sound of 40-50kHz frequency. The receiver senses the ultrasonic sound from the transmitter and enables a unijunction transistor (UJT) based relaxation oscillator, which, in turn, controls the lamp brightness by phase control of a silicon-controlled rectifier (SCR).

Fig. 1 shows the block diagram of the ultrasonic lamp-brightness controller. The received signals are amplified and given to the comparator after rectification and filtering. The comparator provides clock pulse to the decade counter. The output of the decade counter enables the UJT oscillator to control the phase angle of the current through the load via the SCR.

Fig. 2 shows the circuit of the ultrasonic transmitter. The transmitter uses a free-running astable multivibrator built around NOR gates of CD4001B that oscillates at a frequency of 40 to 50 kHz. An ultrasonic transducer is used here to transmit the ultrasonic sound.

The transmitter is powered from a 9V PP3 cell. Preset VR1 is used for setting the frequency to 40 kHz. When switch S1 is pressed, the signal is given to the transmitter transducer and inaudible 40kHz sound is transmitted.

Fig.3 shows the receiver circuit of the ultrasonic lamp-brightness controller. The 9.1V power supply for the receiver circuit is derived from 230V, 50Hz AC mains. The AC mains is rectified by diodes D13 through D16 and limited to 9.1V by using zener diode ZD1. Resistor R3 is used as the current limiter. Capacitor C8 acts as a filter to eliminate

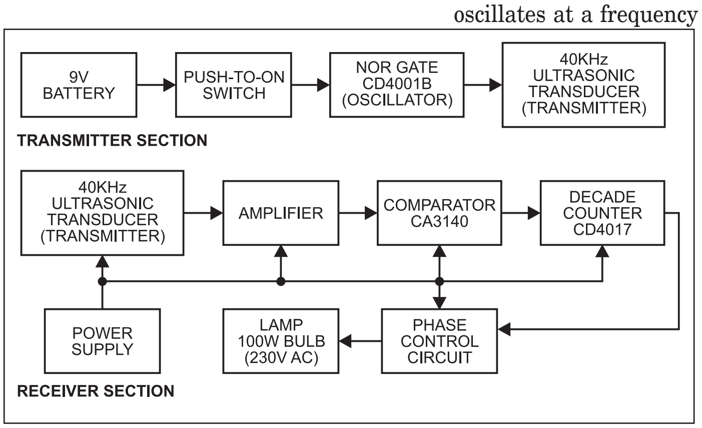


Fig. 1: Block diagram of the ultrasonic lamp-brightness controller

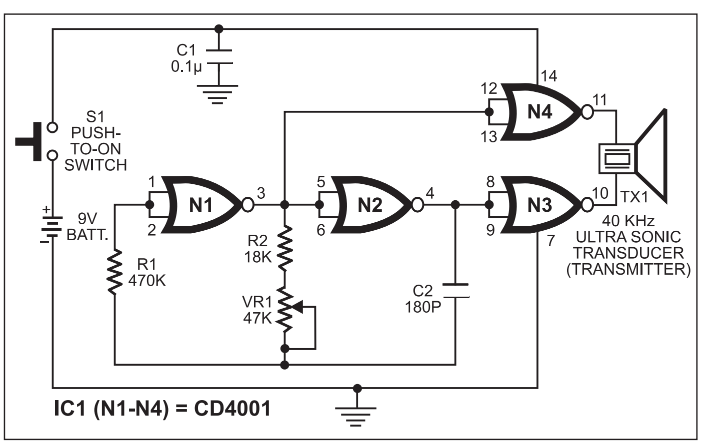


Fig. 2: Circuit of the ultrasonic transmitter

PARTS LIST	
Semiconductors:	
IC1	- CD4001 NOR gate
IC2	- CA3140 operational amplifier
IC3	- CD4017 decade counter
T1, T2	- BC549C npn transistor
T3	- 2N2646 unijunction transistor
SCR1	- TYN6004 silicon-controlled rectifier
D1-D12	- 1N4148 switching diode
D13-D16	- 1N4007 rectifier diode
ZD1	- 9.1V, 0.5W zener diode
Resistors (all ¼-watt, ±5% carbon, unless mentioned otherwise):	
R1	- 470-kilo-ohm
R2, R4	- 18-kilo-ohm
R3	- 56-kilo-ohm
R5	- 8.2-kilo-ohm
R6, R10	- 1.2-kilo-ohm
R7	- 10-kilo-ohm
R8, R9, R14	- 100-kilo-ohm
R11	- 120-kilo-ohm
R12	- 4.7-kilo-ohm
R13	- 10-kilo-ohm, 10W
VR1	- 47-kilo-ohm preset
VR2	- 20-kilo-ohm preset
VR3-VR12	- 2.2-mega-ohm preset
Capacitors:	
C1	- 0.1µF ceramic disk
C2	- 180pF ceramic disk
C3	- 1nF ceramic disk
C4, C5	- 1µF, 25V electrolytic
C6	- 470nF ceramic disk
C7	- 0.01µF ceramic disk
C8	- 100µF, 25V electrolytic
Miscellaneous:	
S1	- Push-to-on switch
TX1	- 40kHz ultrasonic transmitter
RX1	- 40kHz ultrasonic receiver
	- 230V, 60W lamp

ripples.

The receiver transducer senses 40kHz signals from the transmitter and converts them into equivalent electrical variation of the same frequency. These signals are amplified by transistors T1 and T2, then rectified and filtered.

The filtered DC voltage is given to the inverting pin 2 of operational amplifier CA3140 (IC2). The non-inverting pin 3 of IC2 is connected to a variable DC voltage via preset VR2 that determines the threshold value of the ultrasonic signal received by the receiver for controlling the lamp brightness.

Operational amplifier CA3140 has gate-protected MOSFET transistors in the input circuit to provide very high input impedance, very low input current and high-speed performance. It is internally phase-compensated to achieve stable operation.

The clock pulse from IC2 is applied to 5-stage Johnson decade counter IC 4017 (IC3). Johnson counters are a variation of standard ring counters, with the inverted output of the last stage fed back to the input of the first stage. They are also known as twisted ring counters. An n-stage Johnson counter yields a count sequence of $2n$ length, so it may be considered to be a mod-2n counter.

For each pulse from the op-amp, the output of IC3 changes sequentially from Q0 to Q9. Q0 through Q9 outputs of IC3 are connected to presets VR3 through VR12 via diodes D3 through D12. The other ends of presets are shorted and connected to capacitor C7 and the emitter of the UJT (T3). The preset-capacitor combination at the emitter of the UJT forms a relaxation oscillator around the UJT.

Initially, the UJT is in cut-off region and its internal input diode is reverse-biased. When Q0 output of decade counter CD4017 (IC3) goes high, capacitor C7 starts charging through preset VR3. When the voltage across the capacitor becomes high enough, it forward biases the internal input diode of the UJT, and the capacitor discharges into the low-resistance region between the UJT's emitter and resistor R14. Discharging continues until the voltage across the capacitor becomes zero and the internal diode of the UJT is again reverse-biased. When the diode is reverse-biased, capacitor C7 starts charging again.

The process of charging and discharging produces a sawtooth pulse. This pulse triggers SCR1 to control the phase angle of the current through the lamp. The capacitor-preset combinations determine the oscillation frequency of the UJT. At Q0 through Q9 outputs of IC3, presets are set at different values to obtain different phase angles. SCR1

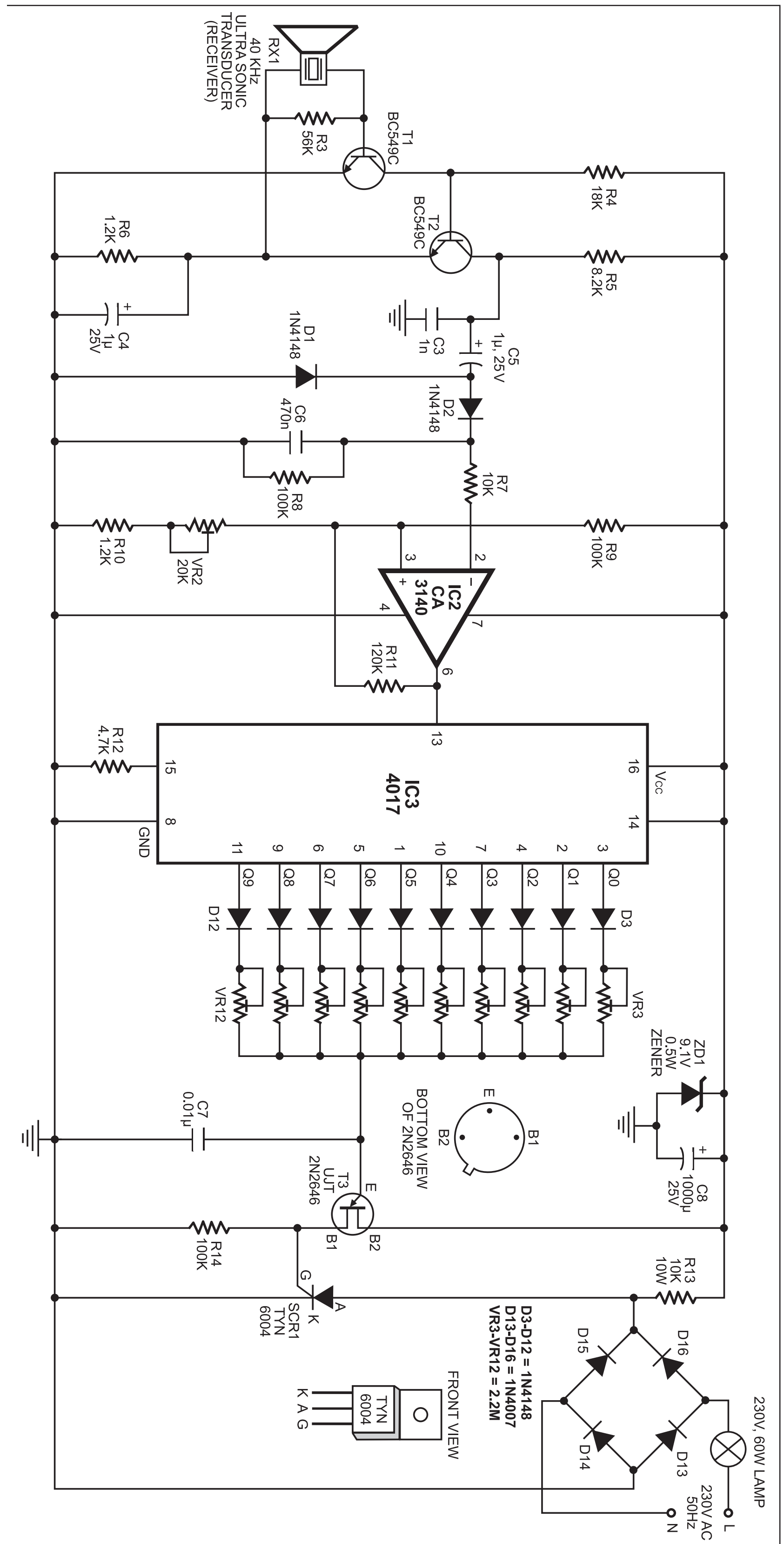


Fig. 3: Receiver circuit of the ultrasonic lamp-brightness controller

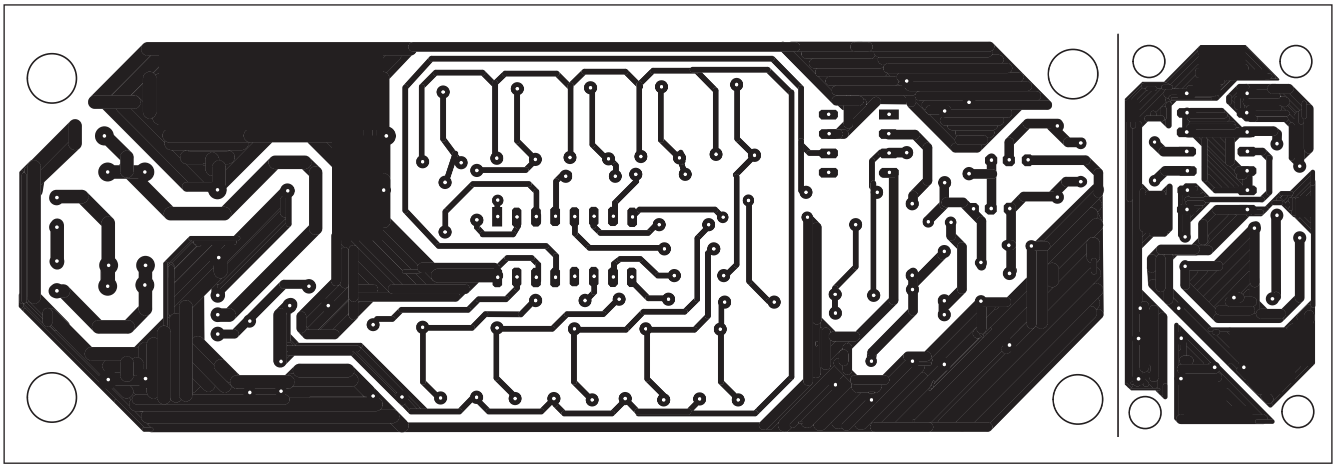


Fig. 4: Actual-size, single-side PCB for transmitter and receiver units of the lamp-brightness controller

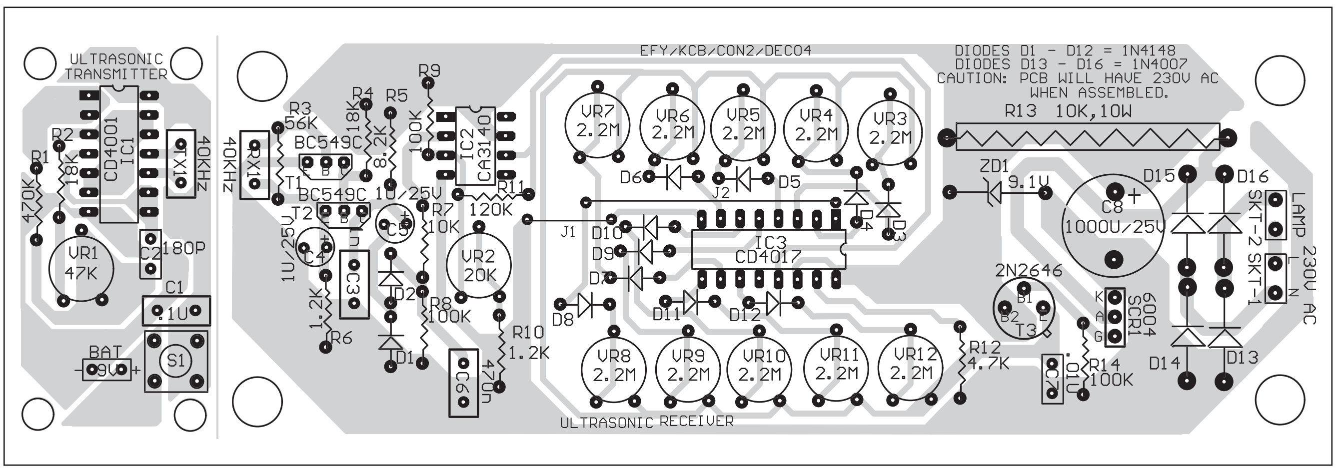


Fig. 5: Component layout for the PCB

directly drives the lamp.

After assembling the circuit, adjust the frequency of the transmitter to exactly 40 kHz. Orient the ultrasonic transducer transmitter towards the receiver transducer such that the receiver can directly receive the ultrasonic waves from the transmitter. Press switch S1 to switch on the transmitter for each operation of switch S1. The brightness level of lamp varies due to the phase control by the UJT.

The combined actual-size, single-side PCB for the transmitter and receiver units of the lamp-brightness controller is shown in Fig. 4 and its component layout in Fig. 5. The two PCBs can be separated by cutting along the vertical line. ❑

MOVING MESSAGE OVER DOT-MATRIX DISPLAY

A. KANNABHIRAN

Controlling electronic devices from a PC is a real fun. Here is a mov ing message display that makes use of the PC's parallel port. The message typed from the keyboard of the PC is displayed on the 5×7 dot-matrix display in moving format.

Moving message employing 5×7 (or 8×8) dot-matrix displays are used in many public places including railway stations and general stores for announcements. These can display any symbol of any language. In cheaper type of moving message displays, the message is stored in ROM/EPROM and the same cannot be changed easily. The costlier ones do provide the facility for changing the message.

The dot-matrix display circuit presented here has the following advantages:

1. The message to be displayed forms part of the program, so we can change the message whenever required.
2. Up to sixteen 5×7 dot-matrix displays can be used.
3. The program can be easily modified to display characters of other scripts.
4. The cost of the circuit will depend on the number of displays used in the circuit.

Here, the circuit is designed for English characters using four 5×7 dot-matrix displays. The message display speed can be varied by changing the display rate in the program.

PARTS LIST

Semiconductors:	
IC1	- 74LS138 1-of-8 demultiplexer
IC2	- 74LS154 1-of-16 demultiplexer
IC3, IC4	- 74LS04 hex inverter
IC5-IC8	- 74LS244 octal buffer
IC9	- 7805 +5V regulator
T1-T27	- BC548 npn transistor
D1, D2	- 1N4007 rectifier diode
Capacitors:	
C1	- 1000µF, 25V electrolytic
Miscellaneous:	
X1	- 230V AC to 7.5V-0-7.5V, 500mA transformer
	- 25-pin 'D' connector

TABLE I
Parallel-Port Pin Details

Pin number	Traditional use	Port name	Read/Write	Port address	Port bit
2-4	Data out	Data port	W	Base	D0-D2
5-9	Data out	—	W	Base	D3-D7
1	Strobe	Control port	R/W	Base+2	C0
14	Auto feed	—	R/W	Base+2	C1
16	Initialise	—	R/W	Base+2	C2
17	Select input	—	R/W	Base+2	C3
15	Error	Status port	R	Base+1	S3
13	Select	—	R	Base+1	S4
12	Paper end	—	R	Base+1	S5
10	ACK	—	R	Base+1	S6
11	Busy	—	R	Base+1	S7
18-25	Ground	—	—	—	—

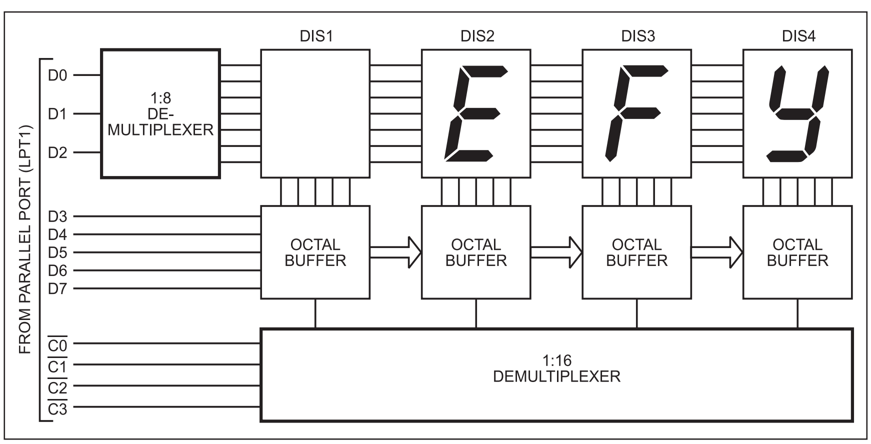


Fig. 1: Block diagram for moving message display using PC's parallel port

PC's parallel port

The PC's parallel port (LPT port) is used to output the display code and control signals for the moving message display. The parallel port is terminated into a 25-pin D-type female connector at the back of your PC. IBM PCs usually come with one or two LPT ports.

Each parallel port is actually made up of three ports, namely, data port, status port and control port.

Pins 2 through 9 form the 8-bit data port. This is purely a write-only port, which means it can be used only to output data.

Pins 1, 14, 16 and 17 form the control port. This port is read-/write-capable, which means it can be used both for outputing and inputing some data to/from the external hardware.

Pins 10 through 13 and pin 15 to-

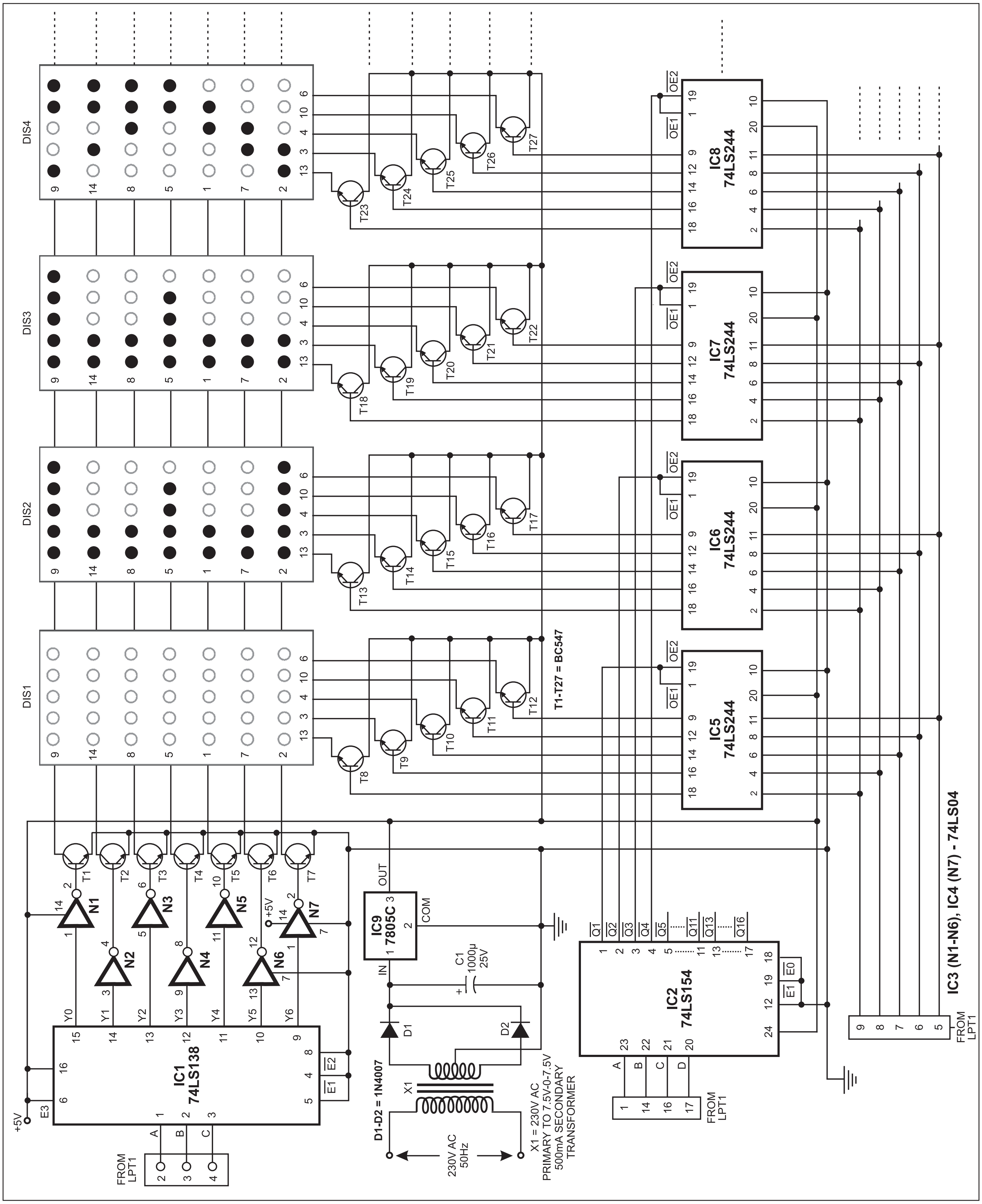


Fig. 2: Circuit diagram for moving message over dot-matrix display

gether form the status port. This is a read-only port, which means it can be used only to read data from the external hardware.

Table I shows pin details of the standard parallel port (SPP), including their traditional usage. The base address of the first parallel port (LPT1) is 378 (hex) or 888 (decimal). The data port of the parallel port can be accessed by its base address. The status port can be accessed using base address+1, i.e. 0379 hex (or 889 decimal). The control port can be accessed using base address+2, i.e. 037A hex (or 890 decimal).

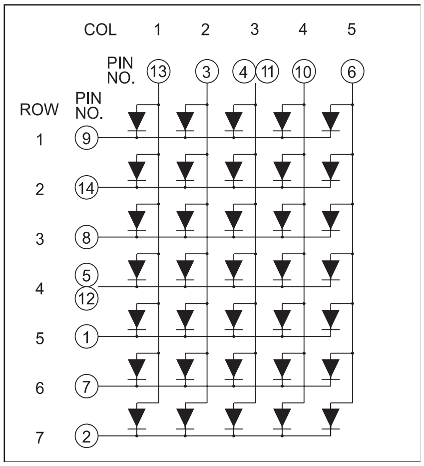


Fig. 3: Pin configuration of 5×7 dot-matrix display

Similar method can be followed for LPT2, whose base address is 0278 in hex. In the present application, we only need to output data. Since status port is a read-only port, the same is not used. Pins 18 through 25 are grounded.

The circuit

Fig. 1 shows the block diagram of the moving message display. The data to be output from the PC's parallel port (LPT1) is first processed by the program. Data lines D0 through D2 of the parallel port are used to enable the seven rows of the dot-matrix display using the 1:8 demultiplexer (IC1). Data lines D3 through D7 and control lines C0, C1, C2 and C3 are used (as output lines) to enable the columns of the dot-matrix display via the 1:16 demultiplexer (IC2).

Fig. 2 shows the circuit diagram of the moving message display using the PC's parallel port. The circuit comprises 1-of-8 demultiplexer 74LS138 (IC1), octal tristate buffers 74LS244 (IC5 through IC8), 1-of-16 demultiplexer 74LS154 (IC2), transistors and four 5×7 dot-matrix displays. Discrete light-emitting diodes (LEDs) can also be arranged in a matrix format to make an alphanumeric display, with each diode representing a pixel. However, it is advantageous to use a 5×7 matrix display which can be obtained in a single package such as FYM-2057IAX from Ningbo Foryard Opti-Electronics Co. Ltd (refer Fig. 3).

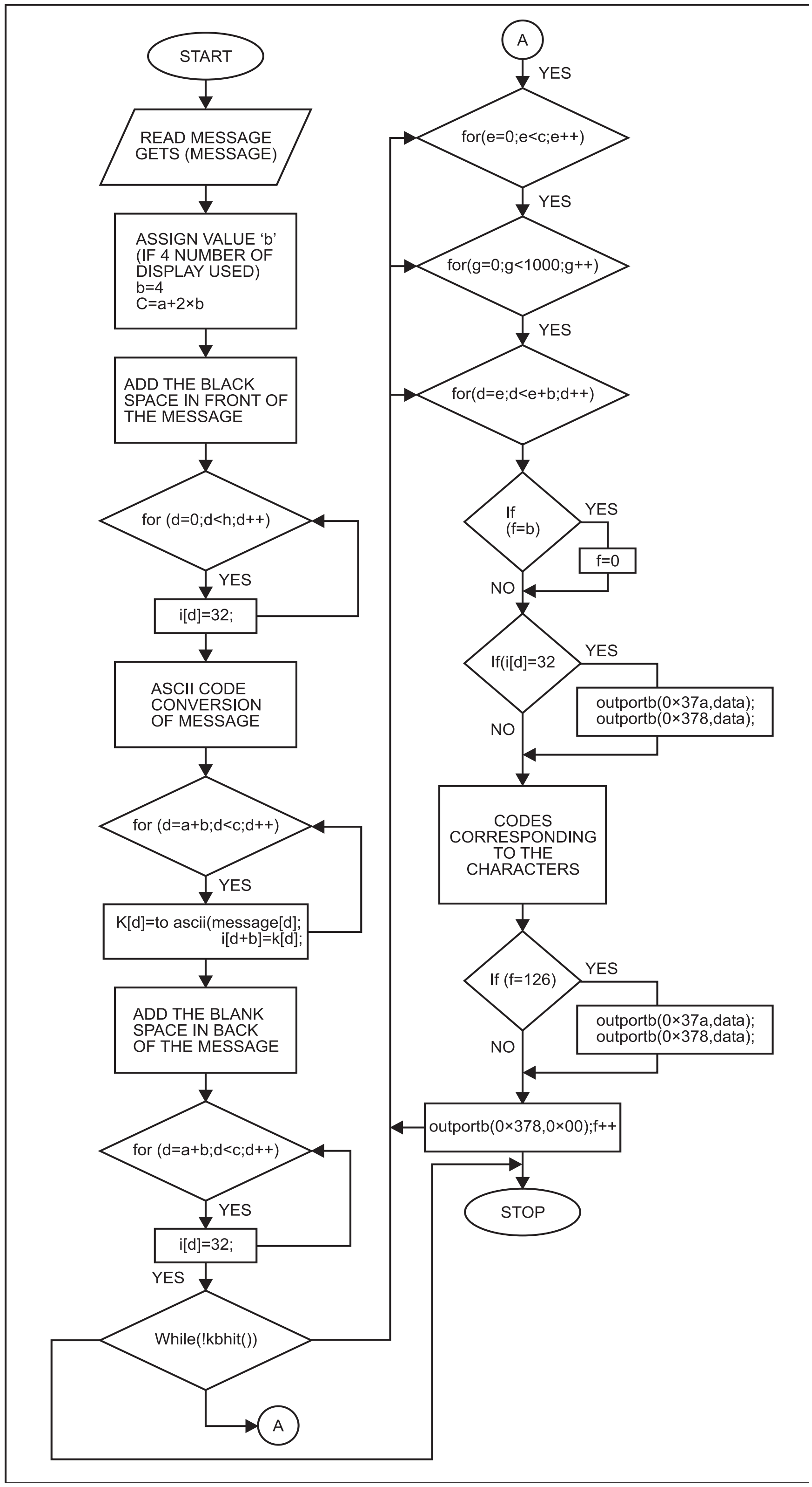


Fig. 4: Flow-chart of the program

The AC mains is stepped down by transformer X1 to deliver a secondary output of 7.5V-0-7.5V AC at 500 mA. The transformer output is rectified by a full-wave rectifier comprising diodes D1 and D2, filtered by capacitor C1, then regulated by regulator 7805C (IC9) to provide regulated 5V power supply to the circuit.

1-of-8 demultiplexer 74LS138 (IC1) provides ground path to the cathodes of all the LEDs of the dot-matrix display through inverters and transistors by using the time-division multiplexing technique. Pins 1 through 3 of IC1 are connected to pins 2 through 4 of LPT1. The outputs of

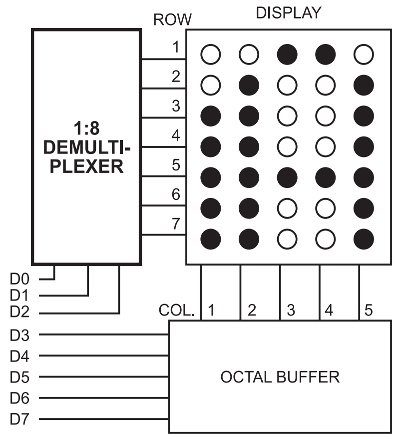


Fig. 5: Diagram of 'A' in 5×7 dot-matrix pattern

IC1 are inverted by NOT gates N1 through N7 and fed to transistors T1 through T7.

IC 74LS138 (IC1) has only eight active-low outputs. Enable pins E1 and E2 have been made permanently low, while enable pin E3 has been made permanently high. Any of the outputs of IC1 can be made low by inputing a 3-bit binary address. The low output of IC1 is made high by the inverter to forward bias the corresponding transistor (T1 through T7). This provides ground to the cathode of the respective LED of the dot-matrix display as shown in the schematic.

Pins 5 through 9 of LPT1 are connected to the non-inverting input pins of all the tristate buffers 74LS244 (IC5 through IC8). The input data of any buffer becomes available at its output when a low enable signal is provided by IC2. Demultiplexer IC 74LS154 (IC2) provides the enable signal to IC5 through IC8 using time-division multiplexing technique. There is provision for connecting twelve additional 74LS244 ICs to control another twelve 5×7 dot-matrix displays.

IC 74LS154 (IC2) has 16 active-low outputs. Its active-low enable pins E1 and E2 have been made permanently low. Any of the sixteen outputs of IC2 can be made low by inputing a 4-bit binary address.

Output pins 1 through 4 of IC2 are connected to enable pins 1 and 19 of buffers IC5 through IC8, respectively. The outputs of IC5 through IC8 are fed to the transistors connected to displays DIS1 through DIS4. The high output of buffer forward biases the connected transistor to provide + 5V supply to the anodes of the corresponding LEDs of the dot-matrix display.

The actual-size, single-side PCB for the moving message over dot-matrix dis-

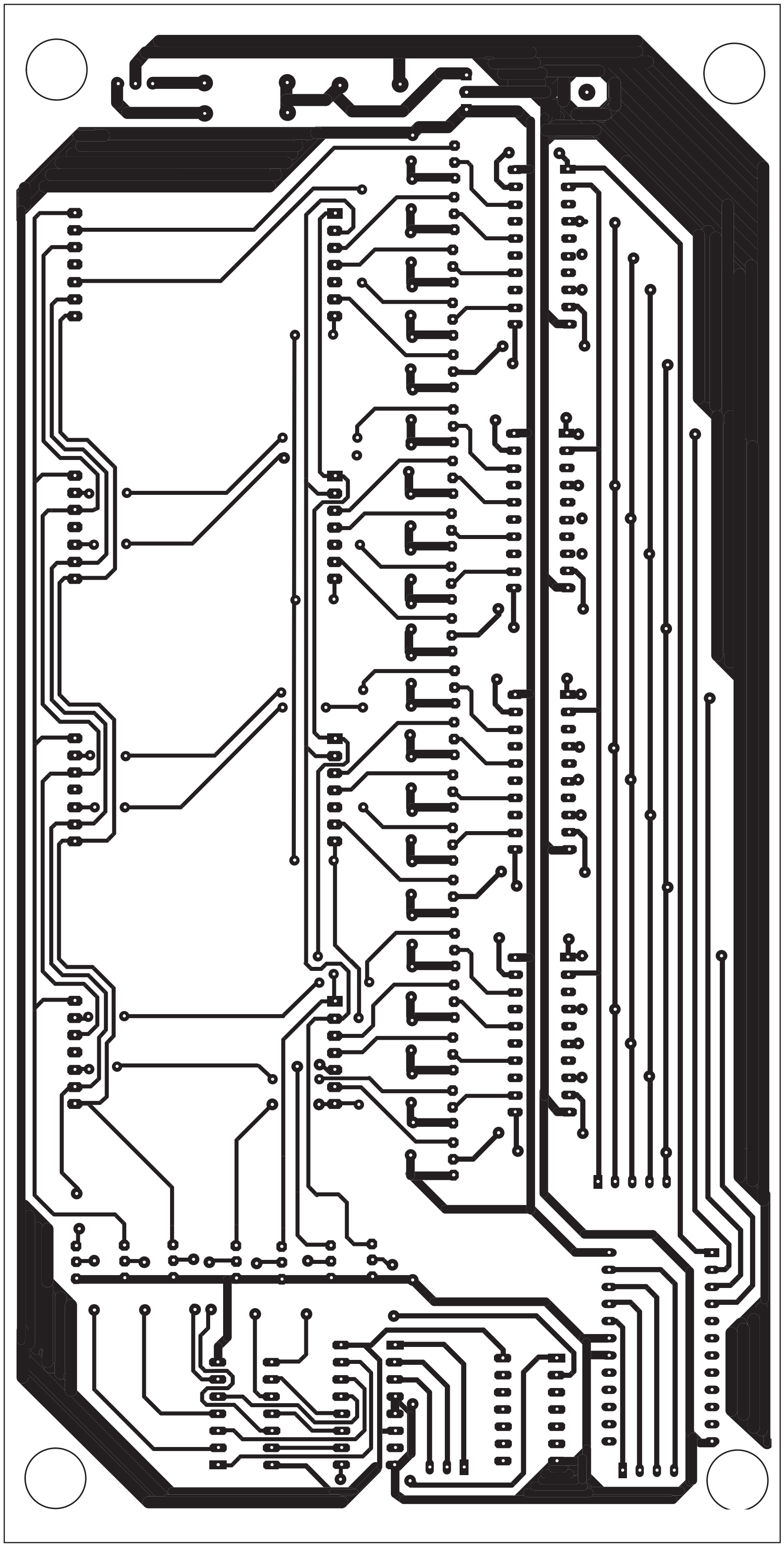


Fig. 6: Actual-size, single-side PCB for moving message over dot-matrix display

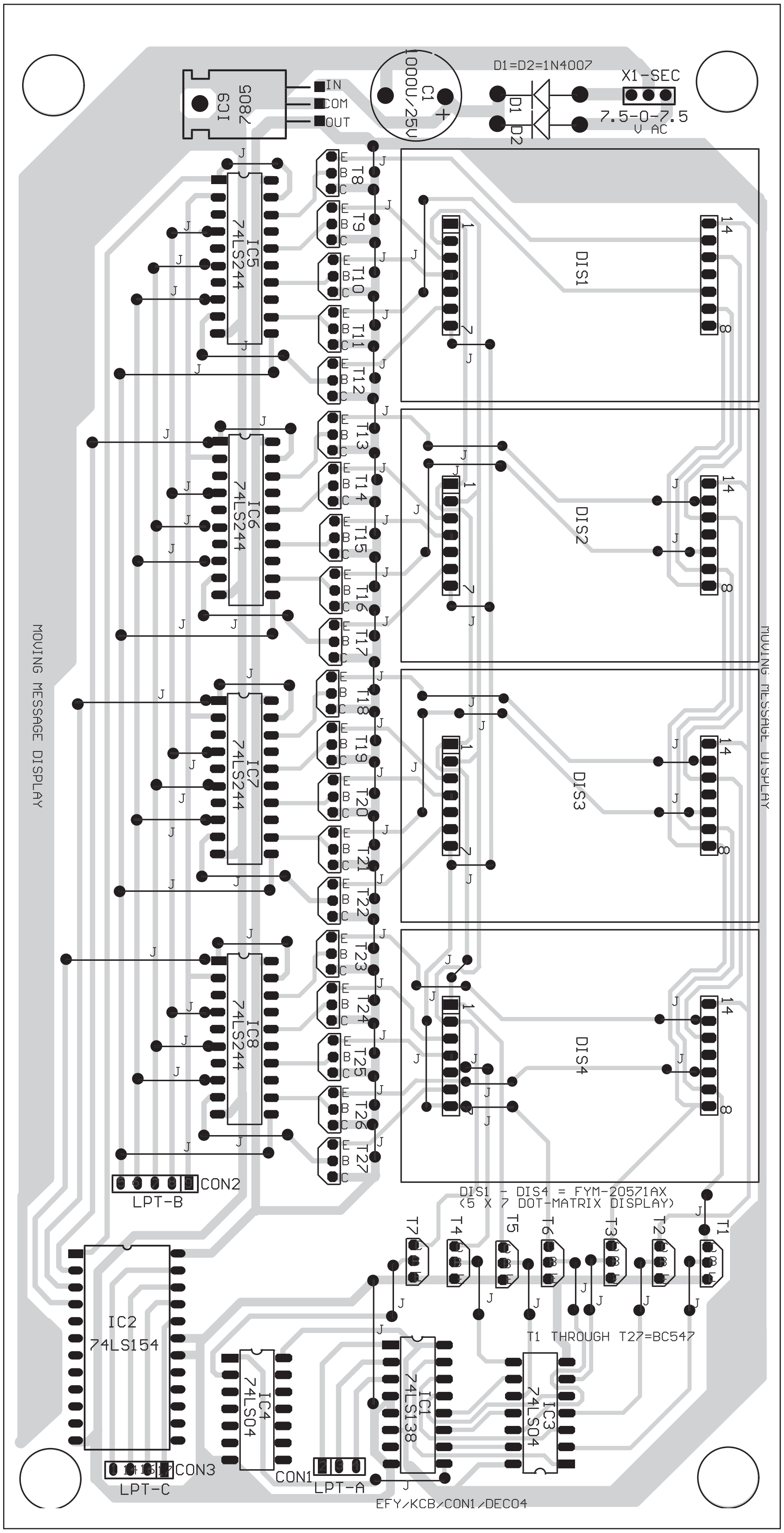


Fig. 7: Component layout for the PCB

TABLE II
Hex Equivalent of the Data Bits for Display of Columns and Rows

Bits for display of column					Bits for display of row			Equivalent hex code
D7	D6	D5	D4	D3	D2	D1	D0	
0	0	1	1	0	0	0	0	30
0	1	0	0	1	0	0	1	49
1	1	0	0	1	0	1	0	CA
1	1	0	0	1	0	1	1	CB
1	1	1	1	1	1	0	0	FC
1	1	0	0	1	1	0	1	CD
1	1	0	0	1	1	1	0	CE

play is shown in Fig. 6 and its component layout in Fig. 7.

The software

The software program for the moving message display is written in 'C++' language. It works as per the flow-chart shown in Fig. 4. The message (data) entered from the keyboard of the PC gets stored into an array.

Variable 'b' in the program signifies the number of blank spaces to be added before and after the message. Its value is determined from the number of dot-matrix displays used in the circuit. Since we have used four displays, assign a value of '4' to variable 'b.' The program now adds four blank spaces before and after the message. In case you are using all the sixteen displays, assign a value of '16' to variable 'b'.

The stored data is converted into the equivalent ASCII code and stored in the new array. ASCII code conversion is performed by including the header file '<ctype.h>' in the software.

The length of the message (including characters, numbers and blank spaces) is measured by string function, which is performed by the header file '<string.h>.' The message is converted into hex code and sent to the parallel port for 5x7 dot-matrix display. At the parallel port, data output is available in time-division multiplexing format. The speed of operation depends on the value of 'g' used in the program.

Suppose you want to display the message "Electronics For You." The length of this message is calculated by the string function as '19.' Since we've used four displays, four blank spaces get added before and after the message. Thus the length of the message now increases to 19+8=27.

The 8-bit data available (through data lines D0 through D7) from the parallel port's address 0×378 (base address+0) is used to display a single letter. Three bits (D0 through D2) from base address+0 are given to demultiplexer IC1 and the remaining five bits (D3 through D7) are given to the buffers (IC5 through IC8). Data flow from the buffers is controlled by demultiplexer IC2. Rows and columns of all the four dot-matrix displays are controlled by D0 through D2 and D3 through D7 with the help of control pins C0, C1, C2 and C3. The four control bits (C0 through C3) from base address+2 of the parallel port are given to IC2 to provide active-low output to enable the buffers.

Designing a character or symbol

Suppose you want to display letter 'A.' Draw 'A' on the 5×7 dot-matrix display as shown in Fig. 5. Now, to convert the letter 'A' into column bits (D3 to D7) for each row, the 'off' LEDs represent '0' and dark or lighted LEDs represent '1.' Thus bits D3 through D7 are converted into their binary and hex formats in Table II for each of the rows. For activation of rows, we use bits D0 through D2 and they simply change sequentially to output binary '000' through '110' (refer Table II). Hex equivalent of the eight data bits for each row is shown in the last column of Table II.

The rows of the display are selected through demultiplexer IC1 by using the three bits D0 through D2 from the parallel port and the columns of the display are selected by the remaining five bits D3 through D7 from the parallel port. These eight bits are converted into the equivalent hexadecimal value and sent through the parallel port by the program. Similarly, you can convert any letter of a message into its equivalent hex output for a blank space. Digits D3 through D7 of Table II will be all zeros for all rows, while activating bits D0 through D2 will change sequentially from '000' through '110.'

EFY note. The source code and other relevant files of this project have been included in CD. ❑

SECTION B :
CIRCUIT IDEAS

INTRUDER ALARM

PRAVEEN KUMAR

This circuit, fitted to the door of your house, sounds an alarm if anyone pushes the door. This way it alarms you against thieves or intruders. The circuit (refer Fig. 1) comprises transmitter and receiver sections. The transmitter is fitted onto the inside of the doorframe and the receiver is fitted to the door panel.

The transmitter section comprising a laser diode is powered by a 9V battery. When switch S1 in the transmitter section is closed, the laser diode glows. Closing switch S2 provides power supply to the receiver section. Light falling from the laser diode on the light-dependent resistor (LDR) in the receiver section provides base current to transistor T1 and it starts conducting. This grounds the base of transistor T2, so it doesn't conduct and the alarm remains off.

When somebody pushes the door, light incident on the LDR is interrupted and transistor T1 stops conducting since the LDR offers a high resistance in the absence of light. Transistor T2 receives base current and starts conducting. The pulse from the emitter of transistor T2 is connected to the inputs of AND gate N1 (IC1). The high ouput of AND gate is connected to a JK flip-flop (IC2) that works as a latch. As a result, output pin 12 (Q1) of IC2 goes high to cause conduction of transistor T3 and consequent sounding of the alarm. The alarm can be turned off by switch S2.

Arrange the laser diode and the LDR such that when the circuit is 'on' and the door is closed, light from the laser diode falls on the LDR to keep the alarm off. In order to make sure that ambient light is not incident on the LDR, make the arrangement as shown in Fig. 2.

EFY note. While testing at EFY Lab, a laser torch in place of the transmitter was used.

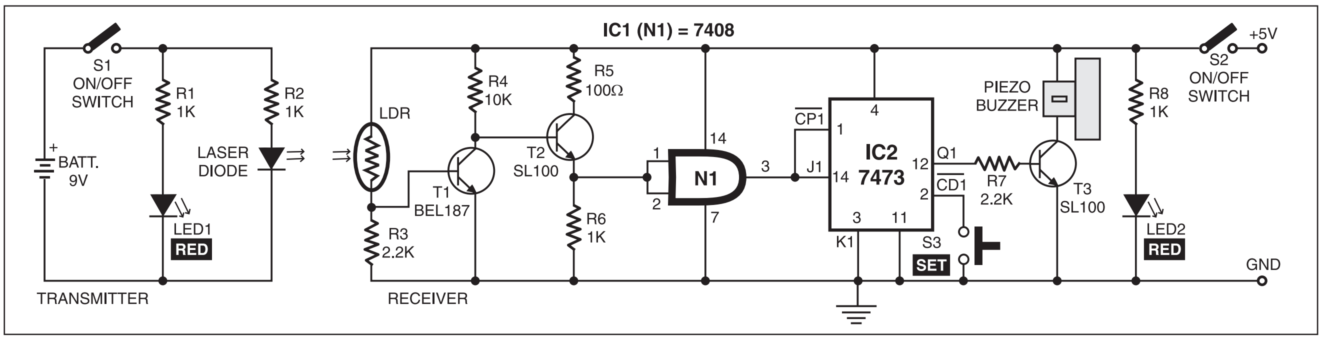


Fig. 1: Circuit diagram of intruder alarm

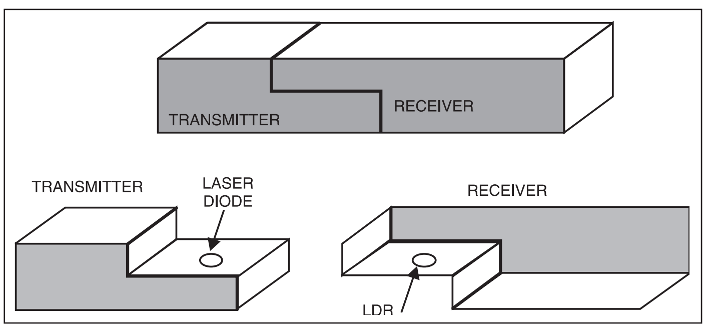


Fig. 2: Transmitter and receiver cabinets with holes for laser LED and LDR, respectively

LED-BASED MESSAGE DISPLAY

S.C. DWIVEDI

This LED-based message display is built around readily availble, low-cost components. It is easy to fabricate and makes use of 3mm red LEDs. A total of 172 LEDs have been arranged to display the message "HAPPY NEW YEAR 2004."

The arrangement of LED1 through LED11 is used to display 'H' as shown

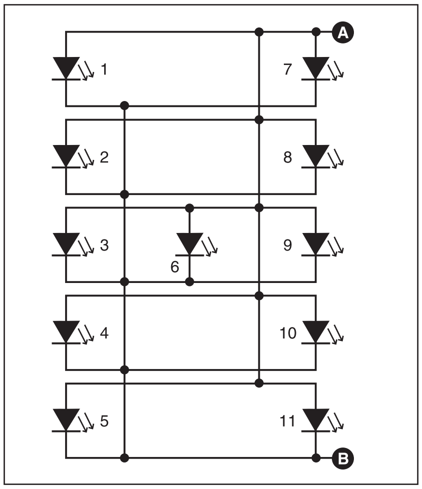


Fig. 1: LED arrangement for word 'H'

in Fig. 1. The anodes of LED1 through LED11 are connected to point A and the cathodes of these LEDs are connected to point B. Similarly, letter 'A' is built using LED12 through LED21. All the anodes of LED12 through LED21 are connected to point A, while the cathodes of these LEDs are connected to resistor R8 (not shown in the circuit diagram). Other letters/words can also be easily arranged to make the required sentence.

The power supply for the message display circuit (Fig. 2) comprises a 0-9V, 2A step-down transformer (X1), bridge rectifier comprising diodes D1 through D4, and a filter capacitor (C1). IC 7806 (IC1) provides regulated 6V DC to the display circuit comprising timer 555 (IC2) and decade counter CD4017 (IC3). The astable multivibrator built around IC2 produces 1Hz clock at its output pin 3. This output is connected to clock pin (pin 14) of the decade counter.

The decade counter can count up to 10. The output of IC3 advances by one count every second (depending on the time period of astable multivibrator IC2).

When Q1 output of IC3 goes high, transistor T1 conducts and the current flows through LED1 through LED48 via resistors R7 through R11. Now the word 'HAPPY' built around LED1 through LED48 is displayed on the LED arrangement board.

Next, when Q2 output of IC3 goes high, transistor T2 conducts and the current flows through LED49 through LED87 via resistors R12 through R14. Now the word 'NEW' is displayed on the LED arrangement board.

Again, when Q3 output goes high, transistor T3 conducts and the current

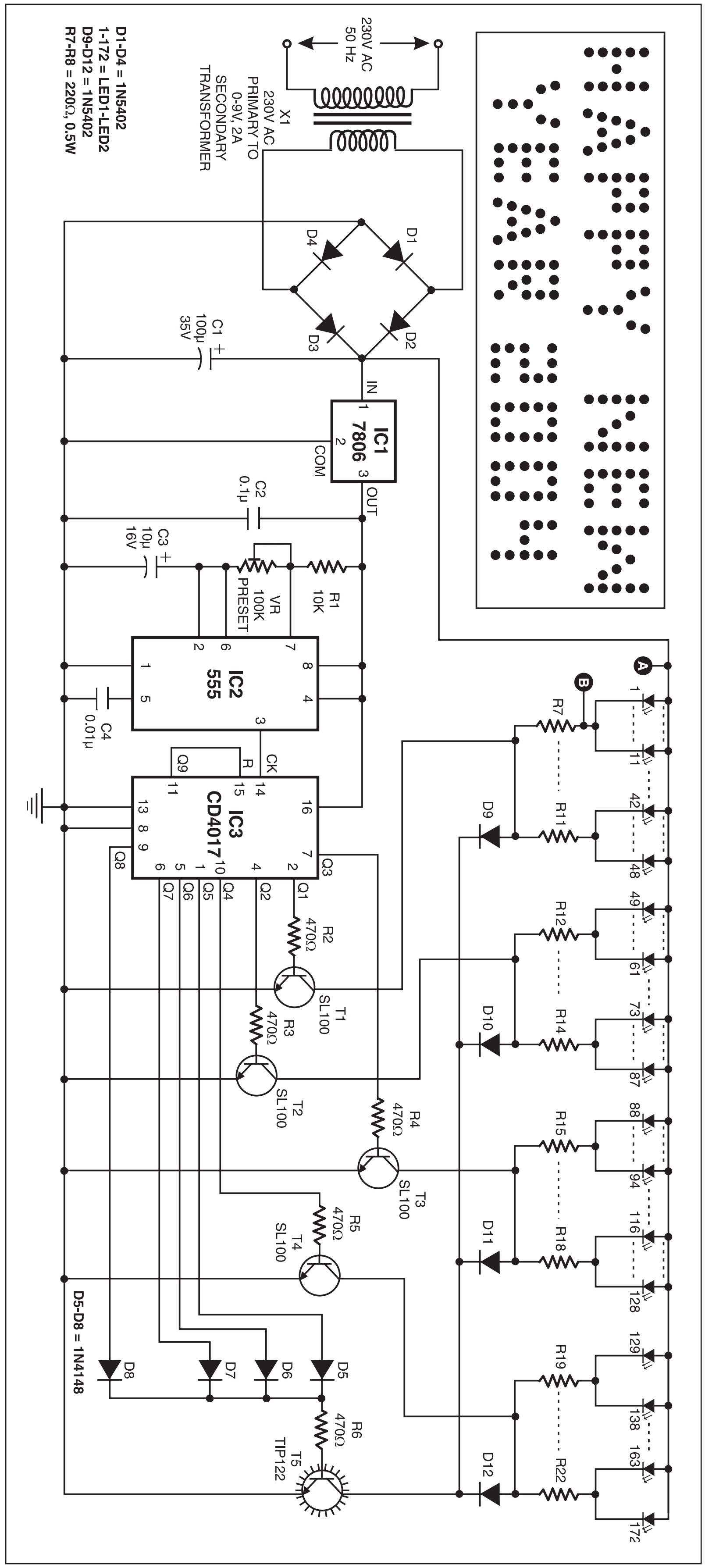


Fig. 2: Circuit diagram of LED-based message display

flows through LED88 through LED128 via resistors R15 through R18. Now the word 'YEAR' is displayed on the LED arrangement board.

Similarly, when Q4 output goes high, transistor T4 conducts and the current flows through LED129 through LED172 via resistors R19 through R22. Now digits '2004' are displayed on the LED arrangement board.

During the entire period when Q5, Q6, Q7, or Q8 output go high, transistor T5 conducts and the current flows through all the LEDs via diodes D9 through D12 and resistors R7 through R22. Now the complete message "HAPPY NEW YEAR 2004" is displayed on the LED arrangement for four seconds.

Thus, the display board displays 'HAPPY,' 'NEW,' YEAR' and '2004' one after another for one second each. After that, the message "HAPPY NEW YEAR 2004" is displayed for 4 seconds (because Q5 through Q8 are connected to resistor R6 via diodes D5 through D8).

At the next clock input output Q9 goes high, and IC3 is reset and the display is turned off for one second. Thereafter the cycle repeats.

DC-TO-DC CONVERTER

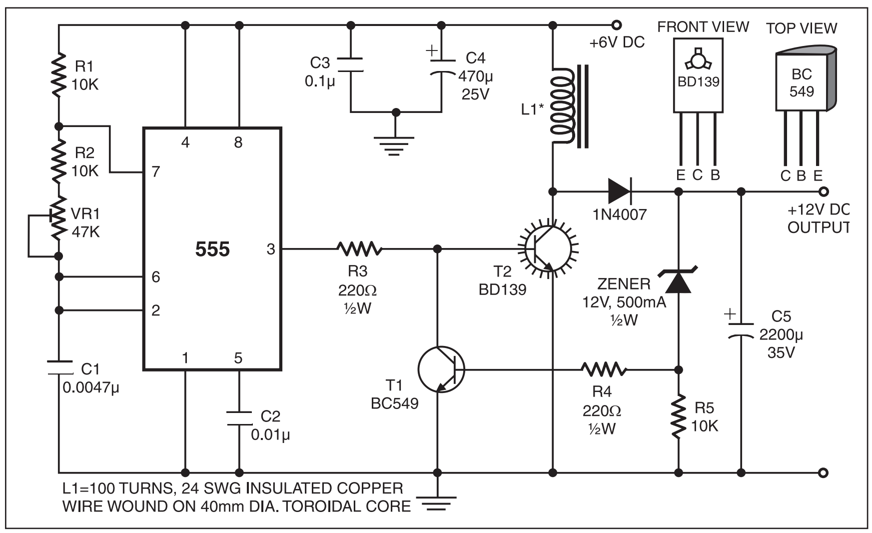
PRINCE PHILLIPS

Here's a low-cost circuit to convert 6V DC into 12V DC. It uses no transformer and is easy to construct with few components.

The circuit is built around IC 555, which generates the required frequency of around 2 to 10 kHz to drive power transistor BD139 (T2). The output frequency of the IC can be adjusted by a 47k potmeter (VR1) and given to the base of transistor T2 via resistor R3. Transistor T2 is mounted on an aluminium heat-sink. Inductor L1 and capacitor C5 (2200µF, 35V) are energy storage components. The 12V zener diode regulates the voltage across the output of the circuit.

The inductor comprises 100 turns of 24SWG enamelled copper wire wound on a 40mm dia. toroidal ferrite core. The more the turns on the core, the higher the current delivering capability of the circuit to the load at the output.

The output current is controlled by transistor BC549 (T1) with the help of resistors R4 and R5. The output voltage is controlled by the zener diode and smoothed by capacitor C5.



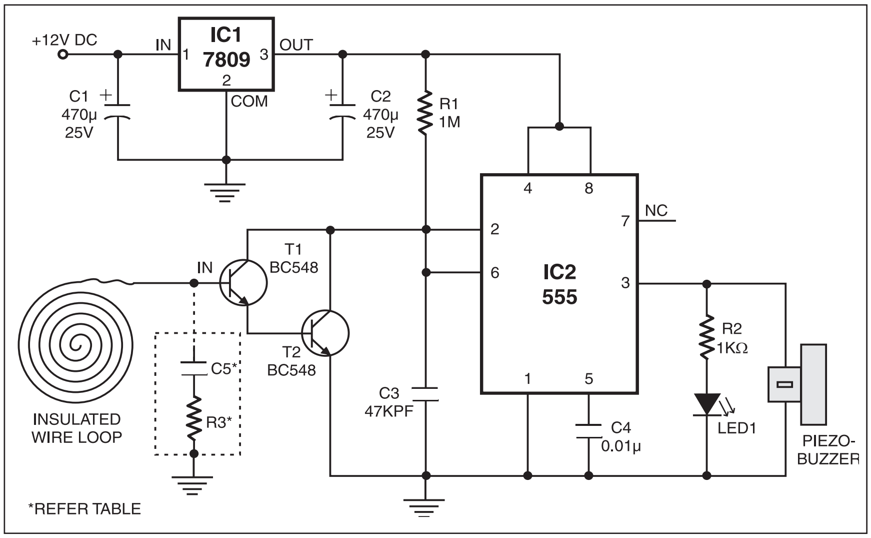
You can obtain regulated 12V DC, 120 mA across the output of this circuit. At higher loads (below 100 ohms), the circuit might not perform well and deliver as much current. Use a large capacitor (C5) and inductor for higher voltages and higher currents, respectively. Different output voltages can be obtained by using zener diodes of other ratings.

VERSATILE PROXIMITY DETECTOR WITH AUTO RESET

KAUSHIK HAZARIKA

Electrochemical processes taking place in our body generate complex signals (hum) that are continuously being passed along the nerve fibres throughout the body. Any physical activity such as muscle movement increases hum.

Here's a circuit that operates when it detects hum generated by the human body



Resistor and Capacitor Values for Optional Circuit				
R3 (kilo-ohms)	**C5 (pF)**	**Approx. triggering distance (mm)**	**Approx. follow-up distance (mm)**	**Approx. loop wire length (mm)**
22	220	3	30	68
220	82	10	50	68
220	10	5	20	68

in proximity. Its versatility lies in the fact that you don't need to touch the metal plates for detection. Just the presence of your hand/body within 1 cm of the sensing loop triggers the circuit. The activation of the circuit is indicated by the glowing of an LED and an audible beep. The circuit continues to glow and beep until the hand is within 5 cm of the loop. Beyond 5 cm, it resets automatically.

Here IC2 (555) simplifies the circuitry otherwise needed to achieve this. Regulator 7809 (IC1) supplies 9V DC to the circuit.

When power is turned on, capacitor C3 (47 kpF) charges through resistor R1 (1 mega-ohm). Output pin 3 of IC2 remains high as long as the voltage at its pin 2 is below 2/3Vcc; the buzzer beeps for this period. Beyond that voltage, the output resets (goes low).

Transistors T1 and T2 (each BC548) form a Darlington pair. As long as T1 and T2 remain in cut-off condition, capacitor C3 retains the charge and the buzzer is off. When you take your hand within 1 cm of the loop wire, T1 conducts due to the noise picked up by its base. So capacitor C3 gets a discharge path, and the voltage at pin 2 of IC2 going below 1/3Vcc sets output pin 3 high. As a result, the buzzer sounds.

The beep continues until C3 charges to 2/3Vcc due to gradual withdrawal of the hand from vicinity of the loop wire. The series combination of capacitor C5 and resistor R3 within dotted lines is optional and reduces hum at the base of T1. The values of C5 and R3 to be used for varying the sensitivity of the circuit are given in the table.

For calibration, wire the circuit and use a 7cm hook-up wire at the base of T1. When you place your hand over the wire insulation, the buzzer should beep. If it doesn't, check connections. Now connect the loop wire. If beep continues even when there is no person within 20 cm, use a suitable combination of C5 and R3 from the table to reduce the circuit sensitivity.

The suggested PCB size for the circuit (excluding power supply) is 4 cm×3 cm. Solder the loop wire directly. A small hook-up wire was used in the prototype.

Do not remove insulation of the wire. Keep the circuit away from mains wiring and large metal objects.

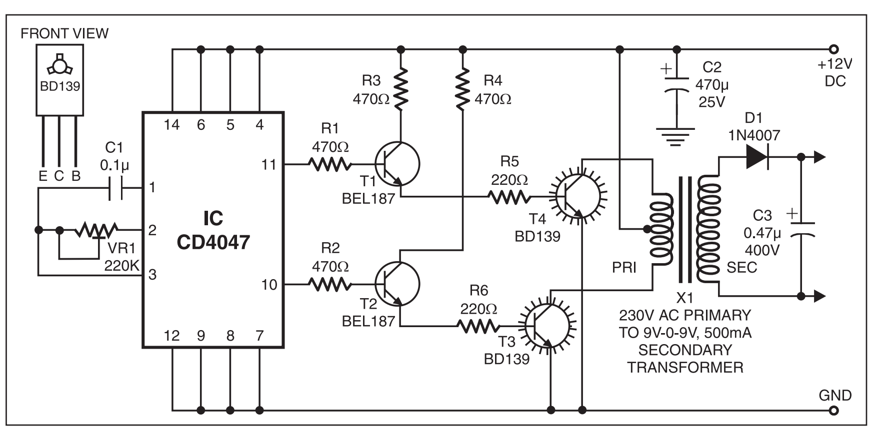
WINDOW CHARGER

PRADEEP G.

Keep away intruders with this compact electrified window charger. The charger produces non-lethal shocks that are strong enough to threaten intruders.

The circuit uses IC CD4047 as a free-running astable multivibrator. Capacitor C1 and preset VR1 are timing components. The pulse repetition rate is determined by the value of 4.4C1×VR1. The frequency can be varied with the help of preset VR1.

The IC generates complementary squarewave signals at pins 10 and 11. Transistors T1 and T2 serve as drivers



for the following push-pull amplfier stage. A high-voltage generator, realised using step-up transformer X1 and medium-power transistors T3 and T4, follows the astable multivibrator. The stepdown transformer is used for reverse function (step-up) and its output is rectified by diode D1, filtered by capacitor C3 and then given to window (made of metal frame).

MULTIBAND CW TRANSMITTER

REJIMON G. VU2RGQ

A radio frequency oscillator is at the heart of all radio transmitters and receivers. It generates high-frequency oscillations, which are known as carrier waves. Here's a continuous-wave (CW) transmitter for transmitting Morse code signals in the shortwave band (see Fig. 1). It is basically a variable frequency oscillator (VFO) whose frequency can be varied from 5.2 MHz to 15 MHz. The signal can be received in the shortwave band by any radio receiver. The circuit works off a 9V battery.

Connect the Morse key (S1) across capacitor C5 as shown in the figure. Attach a telescopic antenna (capable of transmitting over a short distance) at the output terminal. The coil and gang capacitor C2 form the tank circuit. The coil (L) has a total of 60 turns. Winding details are given in Fig. 2. Tappings on the coil allow selection of the required band. The frequency can be varied using C2 (main tuning).

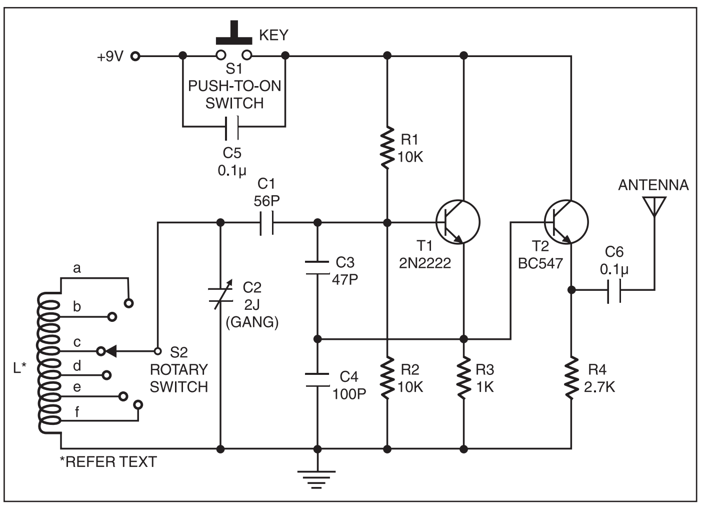


Fig. 1: Circuit of multiband CW transmitter

On reducing turns of the coil (using selector switch S2), the oscillator's frequency increases because frequency is inversely proportional to inductance. Capacitor C1 couples the signal from the tank circuit to the base of transistor T1 (2N2222). Transistor T1 provides the required positive feedback for oscillation and transistor T2 (BC547) functions as the emitter follower. The output is taken from the emitter of T2.

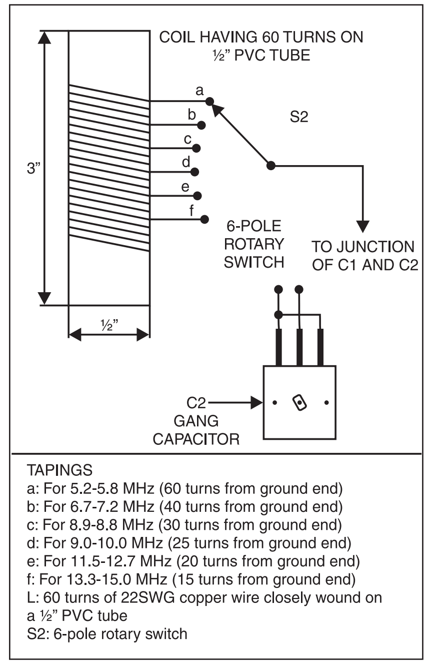


Fig. 2: Details of the inductor

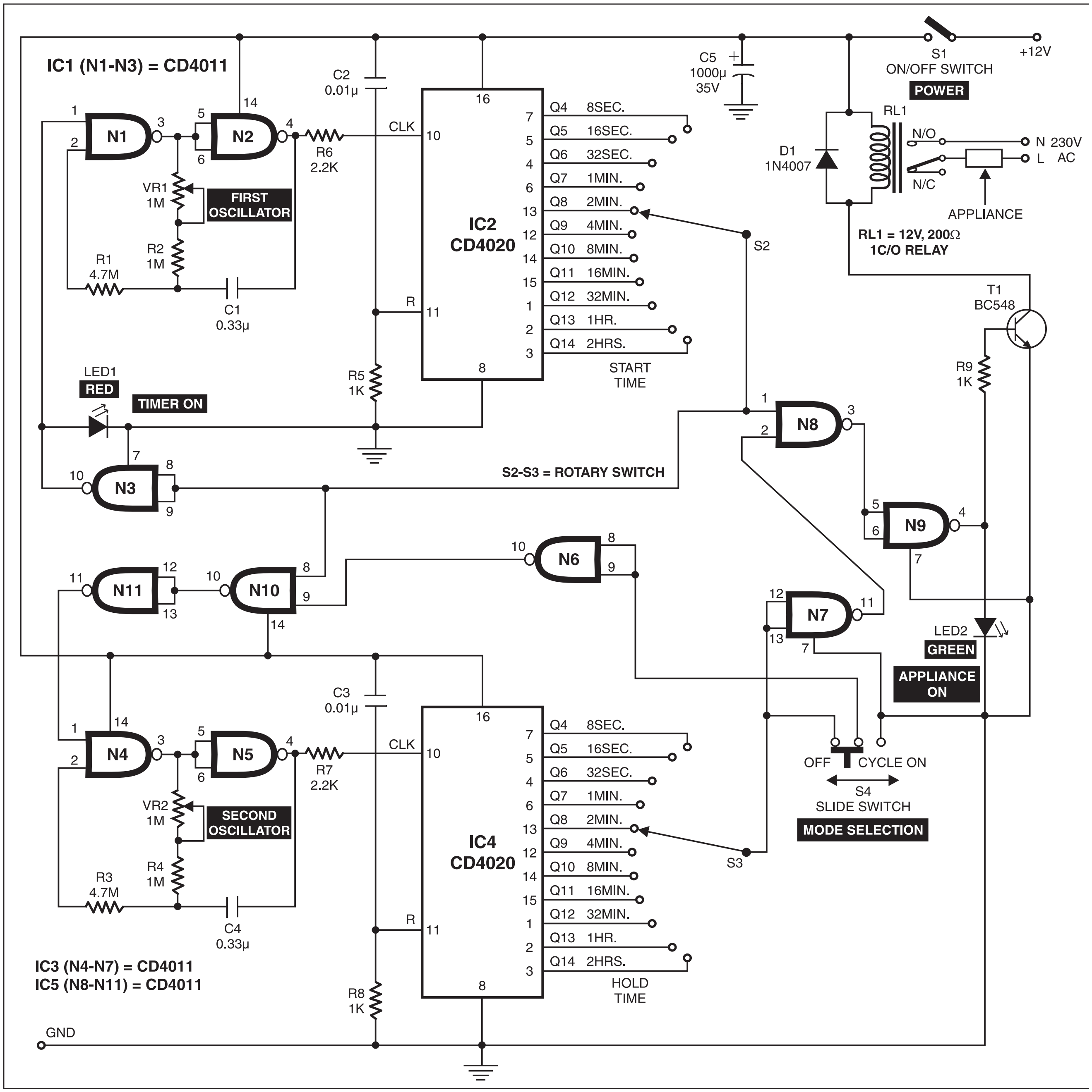
For stable oscillations, use a polystyrene capacitor as C1. All other capacitors may be ceramic disk type. Enclose the circuit in a metal box for better shielding.

PROGRAMMABLE TIMER FOR APPLIANCES

MITESH P. PARIKH

This programmable timer is useful for domestic, commercial as well as industrial applications. It automatically turns the appliance on/off after a preset time. The time period can be varied from 8 seconds to 2 hours with the help of rotary switches S2 and S3. The circuit works in two modes: off mode and cyclic mode. Slide switch S4 is used for mode selection.

In the off mode, the appliance turns on after a preset time (set by rotary switch



S2), remains on for another preset time (set by rotary switch S3) and then turns off. In the cyclic mode, this process repeats again and again.

The circuit is built around three quad two-input NAND gate ICs CD4011 (IC1, IC3 and IC5), two 14-bit binary ripple counters CD4020 (IC2 and IC4) and a relay driver transistor (T1). It works off a 12V DC, 500mA power supply. You can also power the circuit from mains by using a 12V DC, 500mA adaptor in place of the 12V DC power supply.

Let's assume that you want an appliance to turn on after two minutes and keep it on for another two minutes. For this set the rotary switches S2 and S3 to positions as shown in the figure.

Initially, when power switch S1 is closed, a small charging current pulse through capacitors C2 and C3 resets both the counters (IC2 and IC4) to make all their outputs (Q4 through Q14) low. The high output at pin 10 of NAND gate N3 starts the first oscillator comprising NAND gates N1 and N2, which provides clock pulses to IC2 at the rate of one pulse per second. The glowing of red LED (LED1) indicates that this oscillator is working well and timer is 'on.'

During the first 2 minutes, relay RL1 remains de-energised by the control circuit formed by NAND gates N7, N8 and N9 and LED2 is off, which indicates that the appliance is in 'off' codition. The second oscillator built around NAND gates N4 and N5 (which provides clock pulses to IC4 at the rate of one pulse per second) is inhibited by the timing control circuit formed by NAND gates N6, N10 and N11.

After 128 pulses (approximately two minutes), the Q8 output of IC2 goes high to perform the following three functions:

1. Make the output at pin 10 of NAND gate N3 low via rotary switch S2, which

inhibits the first oscillator

2. Energise relay RL1 via NAND gates N8 and N9 and relay driver transistor T1 to make appliance 'on'

3. Make the output at pin 10 of NAND gate N10 low, which is connected to the inputs of NAND gate N11 to make its output at pin 11 high. This high output is further connected to the input (pin 1) of NAND gate N4.

Now the second oscillator starts oscillating and provides clock pulses to pin 10 of IC4 at the rate of one pulse per second.

Now, after 128 pulses (approximately two minutes), the Q8 output of IC4 goes high. This de-energises the relay via NAND gates N7 and N9 and relay driver transistor T1, provided the mode-selection slide switch S4 is towards off position. The high Q8 output will inhibit the second oscillator via NAND gates N6, N10 and N11 to stop clock pulses to pin 10 of IC4. Thus the relay is energised only once (for 2 minutes) since clock pulses to both IC2 and IC4 are stopped altogether and their outputs get latched.

In case the mode-selector switch S4 is towards 'cycle on' side, clock pulses to IC4 would continue and the relay is alternately energised and de-energised for two minutes each. This continues until the circuit is switched off and started again, or the mode-selector switch is slided towards 'cycle off' side.

Rotary switch S2 is used for start time selection and rotary switch S3 is used for hold time selection. The start and hold time can be increased up to 24 hours by changing the values of R and C components of the oscillator circuit of first and second oscillator.

For heavier load, use a relay of a higher current rating. The circuit can be made on a multipurpose PCB and put in a plastic or metal cabinet with proper ventilation.

Readers' comments

1. What should be the values of VR1, R2 and C1 for making the timer for 12-hour and 24-hour operation in the circuit? Variable resistors are not available above the mega-ohm value. If non-polar type C1 of a higher value is not available, what type of capacitor can be used and how it should be connected?

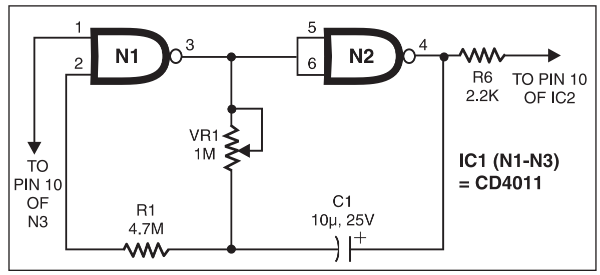
2. Can this timer be used for 15A to 20A loads, and if any spark develops, is there any method to eliminate spark?

Balakrishnan K. Nair
Mumbai

The author, Mitesh P. Parikh, replies:

1. I am thankful to Mr Nair for showing interest in my circuit. Here are the replies to his queries:

1. For 12-hour and 24-hour operation, the values of various components are VR1=528 kilo-ohms (using a 1-mega-ohm preset), R2=0 ohm (R2 unused) and C1=10µF, 16V electrolytic capacitor with positive terminal connected to pin 4 of N2 of the first oscillator and negative terminal connected to the junction of resistors R1 and VR1 as shown in the figure.



Modified circuit of programmable timer for appliances

Similarly, this can be done for the second oscillator also. Using these values, at pins 2 (Q13) and 3 (Q14) of IC4 (IC CD4020), we will get a delay of 12 hours and 24 hours, respectively.

2. As mentioned in the article, the timer can be used for 15A to 20A loads. The relay used in the timer should be of a higher current capacity, such as an industrial relay. Relays that can carry loads with heavy currents (such as 15A) are available in the market. These are enclosed in a black plastic cover, with only relay terminals being out, which can be mounted on the PCB.

For elimination of sparking, you should use a good-quality relay. Else, you can use a silicon-controlled rectifier (SCR) for switching heavy loads. In this case, there is no spark, but an additional circuit is required between the timer and the SCR for triggering and turning off the SCR at the correct time.

ANTI-BAG-SNATCHING ALARM

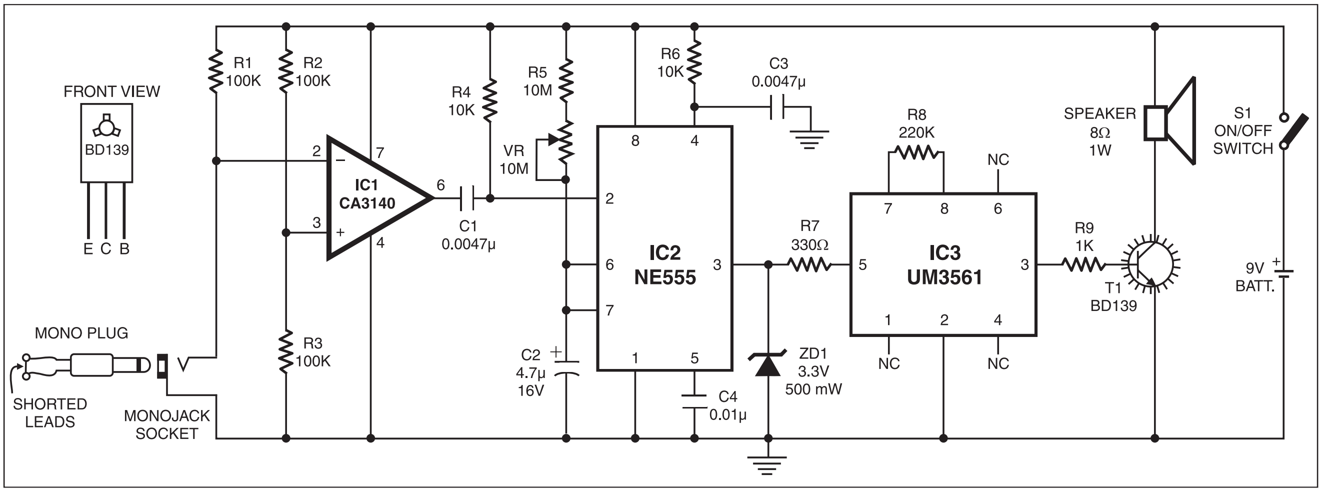
D. MOHAN KUMAR

Here is a simple alarm circuit to thwart snatching of your valuables while travelling. The circuit kept in your bag or suitcase sounds a loud alarm, simulating a police horn, if someone attempts to snatch your bag or suitcase. This will draw the attention of other passengers and the burglar can be caught red handed.

In the standby mode, the circuit is locked by a plug and socket arrangement (a mono plug with shorted leads plugged into the mono-jack socket of the unit). When the burglar tries to snatch the bag, the plug detaches from the unit's socket to activate the alarm.

The circuit is designed around op-amp IC CA3140 (IC1), which is configured as a comparator. The non-inverting input (pin 3) of IC1 is kept at half the supply voltage (around 4.5V) by the potential divider comprising resistors R2 and R3 of 100 kilo-ohms each. The inverting input (pin 2) of IC1 is kept low through the shorted plug at the socket. As a result, the voltage at the non-inverting input is higher than at the inverting input and the output of IC1 is high.

The output from pin 6 of IC1 is fed to trigger pin 2 of IC NE555 (IC2) via coupling capacitor C1 (0.0047 µF). IC2 is configured as a monostable. Its trigger pin 2 is held high by resistor R4 (10 kilo-ohms). Normally, the output of IC2 remains low and the alarm is off. Resistor R6, along with capacitor C3 connected to reset pin 4 of IC2, prevents any false



triggering. Resistor R5 (10 mega-ohms), preset VR (10 mega-ohms) and capacitor C2 (4.7 µF, 16V) are timing components. With these values, the output at pin 3 of IC2 is about one minute, which can be increased by increasing either the value of capacitor C2 or preset VR.

When there is an attempt at snatching, the plug connected to the circuit detaches. At that moment, the voltage at the inverting input of IC1 exceeds the voltage at the non-inverting input and subsequently its output goes low. This sends a low pulse to trigger pin 2 of IC2 to make its output pin 3 high. Consequently, the alarm circuit built around IC UM3561 (IC3) gets the supply voltage at its pin 5.

IC UM3561 is a complex ROM with an inbuilt oscillator. Resistor R8 forms the oscillator component. Its output is fed to the base of single-stage transistor amplifier BD139 (T1) through resistor R9 (1 kilo-ohm).

The alarm tone generated from IC3 is amplified by transistor T1. A loudspeaker is connected to the collector of T1 to produce the alarm. The alarm can be put off if the plug is inserted into the socket again. Transistor T1 requires a heat-sink.

Resistor R7 (330 ohms) limits the current to IC3 and zener diode ZD1 limits the supply voltage to IC3 to a safe level of 3.3 volts. Resistor R9 limits the current to the base of T1.

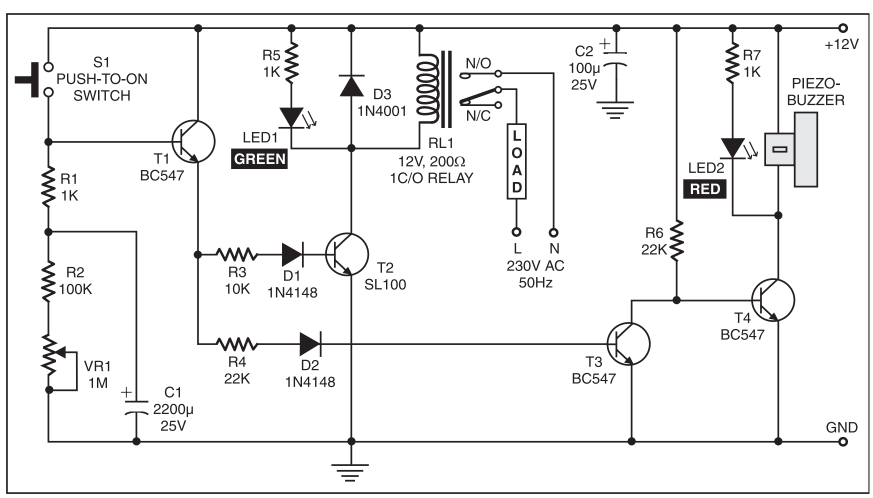
The circuit can be easily constructed on a vero board or general-purpose PCB. Use a small case for housing the circuit and 9V battery. The speaker should be small so as to make the gadget handy. Connect a thin plastic wire to the plug and secure it in your hand or tie up somewhere else so that when the bag is pulled, the plug detaches from the socket easily.

OFF TIMER WITH ALARM

PRADEEP G.

Here's an inexpensive transistorised timer that automatically switches off TV and other appliances after the set time. It works off a 12V DC, 300mA power supply.

Using preset VR1, you can set the time period from a few minutes up to half an hour. After connecting the power supply, momentarily press tactile switch S1. Transistors T1 and T2 conduct to energise relay RL1 and green LED (LED1) glows. The load/appliance connected via N/O contact of relay RL1 is switched on.



At the same time, transistor T3 conducts and transistor T4 stops conducting. So the buzzer doesn't sound and also red LED (LED2) doesn't glow.

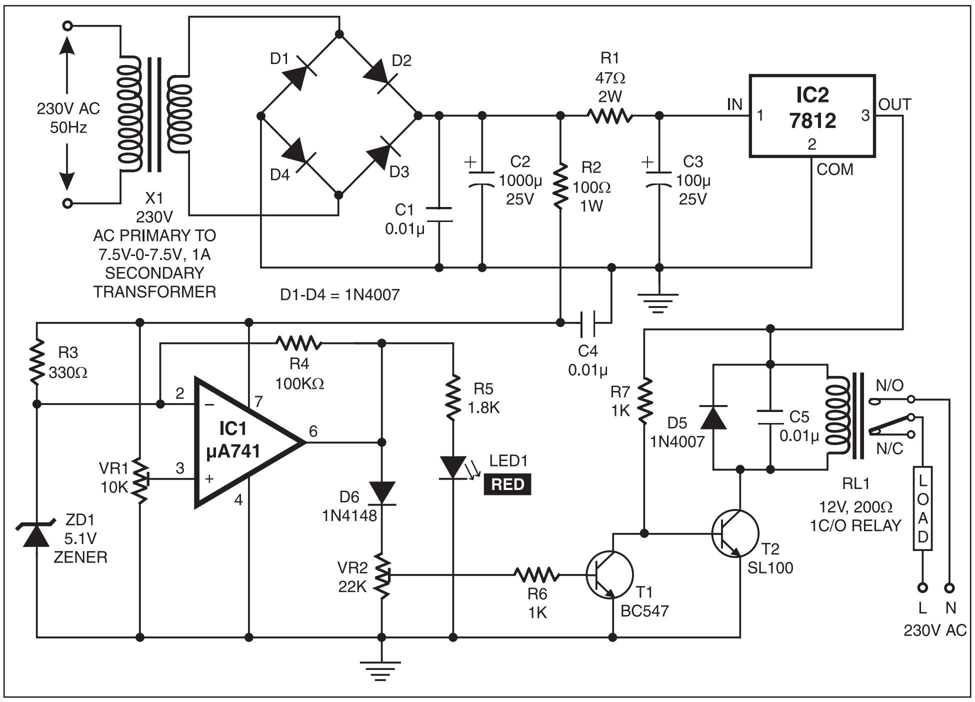
When the 'off' time period is over, relay RL1 de-energises and the appliance connected via N/O contact of the relay is switched off. The buzzer sounds and LED2 glows to indicate that the set time period is over.

OVER-VOLTAGE PROTECTOR

P.V. VINOD KUMAR

This circuit protects your television as well as other electrical appliances from over-voltage. It uses operational amplifier µA741 (IC1) as a comparator. The unregulated power supply is connected to resistor R3 and preset VR1 through resistor R2. Zener diode ZD1 provides reference voltage of 5.1V to the inverting input (pin 2) of IC1.

The non-inverting input (pin 3) of IC1 senses voltage fluctuation in the mains. Preset VR1 is adjusted such that for mains supply below 240V AC, the voltage at the non-inverting terminal of IC1 is less than 5.1V. Hence the output of IC1 is zero and transistor T1 is in non-conducting state. At the same time transistor T2 conducts to energise relay RL1 to connect the mains to the load.



When AC mains is beyond 240V, the voltage at pin 3 of IC1 goes above 5.1V. The high output of IC1 drives transistor T1 and transistor T2 stops conducting to de-energise the relay. Hence the appliance turns off.

Preset VR2 is used for proper biasing of transistor T1. The AC mains supply is stepped down by transformer X1 to deliver a secondary output of 7.5V-0-7.5V AC, 1A. The output of the transformer is rectified by a full-wave rectifier comprising diodes D1 through D4. Capacitors C1 and C2 act as filters to eliminate ripples. Regulator IC 7812 is used to provide regulated 12V supply.

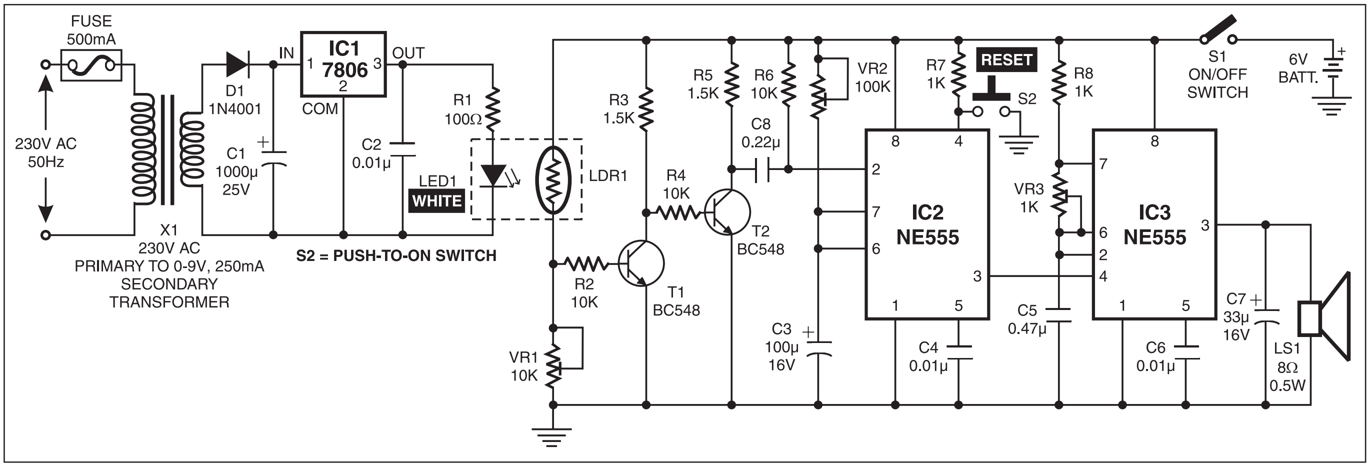
FUSE-CUM-POWER FAILURE INDICATOR

V. GOPALAKRISHNAN

This fuse-cum-power failure indicator comprises an LED (LED1), light-dependent resistor (LDR1), inverter circuit and two timer circuits built around IC NE555. LDR1 and preset VR1 form a voltage divider at the input of the cascaded amplifier comprising two BC548 transistors (T1 and T2). The base of transistor T1 is connected to the junction of LDR1 and the preset through resistor R2. The base of transistor T2 is connected to the collector of T1. The trigger pin of timer IC NE555 (IC2), which is configured as a monostable, is connected to the collector of T2. The output of transistor T1 is inverted by transistor T2. The inverted output of T2 triggers the monostable circuit.

LED1 gets power supply from the AC mains through transformer X1. The secondary output of the transformer is rectified and fed to regulator IC 7806 (IC1). The 6V regulated output drives LED1. As LDR1, enclosed in a cabinet, is kept illuminated by the light from LED1, the output of transistor T2 is normally high.

The transformer has a fuse on the input side of primary winding. When power supply goes off due to power cut or fuse blown off, no light falls on LDR1 and the output of transistor T2 goes low.



This high-to-low transition triggers the monostable (IC2) and its output pin 3 goes high for about 7 seconds. The output of the monostable is connected to reset pin 4 of IC3 (NE555), which is configured in astable mode. The output of the astable circuit is connected to a loudspeaker. IC3, along with the loudspeaker, forms an alarm circuit. Triggering of the monostable activates the alarm circuit, indicating the power failure.

LDR1, cascaded amplifier, monostable and astable circuits get power supply from a 6V battery.

LED-BASED READING LAMP

PRANAB KUMAR ROY

This lamp circuit using ultra-bright white LEDs provides sufficient light for reading purposes while consuming approximately 3 watts of power. In the case of AC mains failure, the battery backup circuit instantly lights up the LEDs. When the power resumes, the battery supply is automatically disconnected and the lamp circuit again works off AC mains.

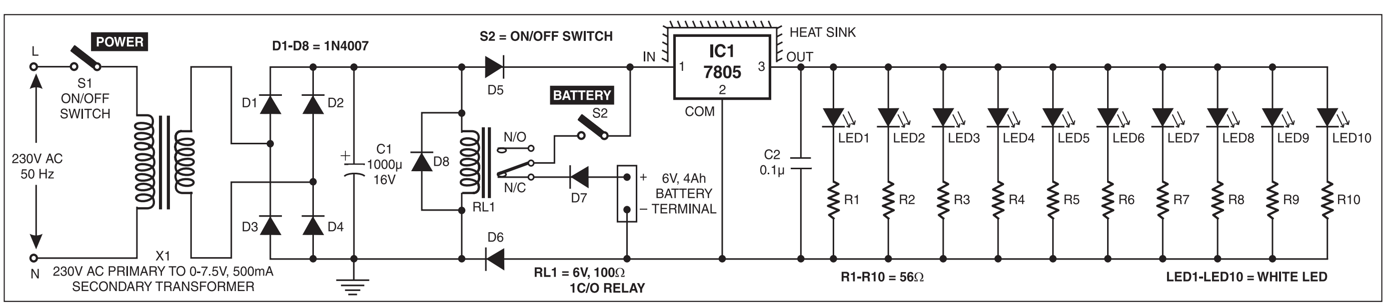
The power supply circuit consists of 0-7.5V, 500mA step-down transformer X1, rectifier diodes D1 through D4 and filter capacitor C1. Regulator IC 7805 (IC1) provides regulated 5V to LEDs, so there is no variation in the intensity of the lamp light even if the mains power supply fluctuates. A total of ten white LEDs (LED1 through LED10) are connected in parrallel across the 5V power supply. Resistors R1 through R10 (each 56 ohms) are connected in series with the white LEDs to limit the current. To increase the intensity of the lamp light, you can add more LEDs in the same manner; a maximum of 15 LEDs can be used for the lamp.

When power switch S1 is closed, relay RL1 energises to disconnect the 6V, 4Ah battery (connected across N/C contact of relay RL1) from input to regulator IC1 if battery switch S2 is closed. When power switch S1 is open, relay RL1 de-energises and connects the battery to the input of IC1 via N/C contacts of the relay.

Diodes D5 and D6 are reverse-current protection diodes that don't allow the battery current to flow towards the power supply section. Diode D7 is for reverse polarity protection of the battery. Before connecting the battery, make sure that it is fully charged.

The circuit can be assembled on a general-purpose PCB. Arrange all white LEDs (LED1 through LED10) on the PCB. Now remove the bulb holder from the lamp and fix the PCB (where bulb holder was mounted) such that LED light falls on your book properly. No separate reflectors are required for LEDs as the LEDs have inbuilt lens reflectors. Use a heat-sink for IC1 as indicated in the figure.

Caution. Though you can read for hours without eye strain in this lamp light, don't directly look at white LEDs for long.



MOBILE CELLPHONE CHARGER

D. MOHAN KUMAR

Charging of the cellphone battery is a big problem while travelling as power supply source is not generally accessible. If you keep your cellphone switched on continuously, its battery will go flat within five to six hours, making the cellphone useless. A fully charged battery becomes necessary especially when your distance from the nearest relay station increases. Here's a simple charger that replenishes the cellphone battery within two to three hours.

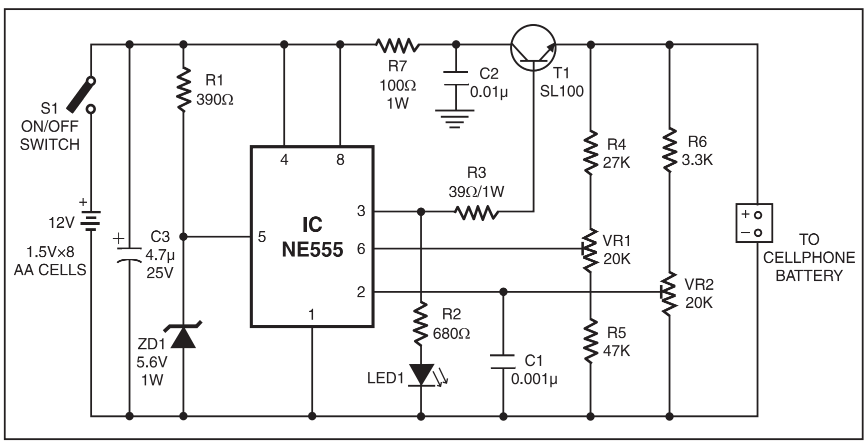
Basically, the charger is a current-limited voltage source. Generally, cellphone battery packs require 3.6-6V DC and 180-200mA current for charging. These usually contain three NiCd cells, each having 1.2V rating. Current of 100mA is sufficient for charging the cellphone battery at a slow rate. A 12V battery containing eight pen cells gives sufficient current (1.8A) to charge the battery connected across the output terminals. The circuit also monitors the voltage level of the battery. It automatically cuts off the charging process when its output terminal voltage increases above the predetermined voltage level.

Timer IC NE555 is used to charge and monitor the voltage level in the battery. Control voltage pin 5 of IC1 is provided with a reference voltage of 5.6V by zener diode ZD1. Threshold pin 6 is supplied with a voltage set by VR1 and trigger pin 2 is supplied with a voltage set by VR2.

When the discharged cellphone battery is connected to the circuit, the voltage given to trigger pin 2 of IC1 is below 1/3Vcc and hence the flip-flop in the IC is switched on to take output pin 3 high. When the battery is fully charged, the output terminal voltage increases the voltage at pin 2 of IC1 above the trigger point threshold. This switches off the flip-flop and the output goes low to terminate the charging process. Threshold pin 6 of IC1 is referenced at 2/3Vcc set by VR1. Transistor T1 is used to enhance the charging current. Value of R3 is critical in providing the required current for charging. With the given value of 39-ohm the charging current is around 180 mA.

LED Status for Different Charging Conditions

Load across the output	Output frequency (at pin 3)	LED1
No battery connected	765 kHz	On
Charging battery	4.5 Hz	Blinks
Fully charged battery	0	Off



The circuit can be constructed on a small general-purpose PCB. For calibration of cut-off voltage level, use a variable DC power source. Connect the output terminals of the circuit to the variable power supply set at 7V. Adjust VR1 in the middle position and slowly adjust VR2 until LED1 goes off, indicating low output. LED1 should turn on when the voltage of the variable power supply reduces below 5V. Enclose the circuit in a small plastic case and use suitable connector for connecting to the cellphone battery.

Note. At EFY lab, the circuit was tested with a Motorola make cellphone battery rated at 3.6V, 320 mAH. In place of 5.6V zener, a 3.3V zener diode was used. The charging current measured was about 200 mA.The status of LED1 is shown in the table.

Readers' comments

The circuit is not working. I tried by changing the values of 3.3-kilo-ohm resistor R6 to 33 kilo-ohms and some other components but to no avail. In this regard, please clarify:

1. Whether the output of the circuit is to be connected to the mobile cell-phone charging socket or directly to the battery after taking it out from the cell phone.

2. Can we use a 6V supply using 1.5V AAA cells? If yes, what changes are to be made?

Y. Diwakar
Principal, ITI
Medchal

The author, D. Mohan Kumar, replies:

I thank Mr Diwakar for showing interest in my circuit. I have designed the circuit for use during long journeys. My prototype is performing well and the circuit was also found to be working satisfactorily at the EFY lab. It requires no modification if a 12V power supply is used.

However, while checking the circuit, Mr Diwakar may note that the performance of the circuit depends on the voltage settings at pins 2 and 6 of IC 555 using VR2 and VR1. Resistor R6 and VR2 form a potential divider to give a voltage below 1/3Vcc at pin 2 to switch on the IC. Resistors R4 and R5 and VR1 provide a reference voltage of 2/3Vcc at pin 6. The voltage at pins 2 and 6 is to be adjusted to around 3.8V and 7.5V, respectively, after connecting a variable power supply to the output terminals.

The charging current is available from the emitter of T1 when the output of IC1 is high. T1 gets bias from the output of IC1 through R3. If the output of IC is correct, change the value of R3 to give proper bias to T1.

The circuit is designed to provide sufficient voltage and current to charge a cell phone two or three times during the journey. That is why a 12V power source with 1.6A current is used in the circuit as power source. Check the outputs of IC1 and T1 and measure the voltages at pins 2 and 6 after proper adjustments of VR1 and VR2. The circuit will work if all the connections and components are correct.

Here are my replies to the specific queries of Mr Diwakar:

1. The output can be directly connected to the cell phone socket using a suitable connector. It is current-regulated. The batteries can also be charged separately (after taking these out from the cell phone) if a suitable holder is available.

2. The circuit is designed to give an output voltage of 3V to 6V to charge different makes of cell phone batteries. Most cellphone batteries require 3.6V to 6V for charging. Each Ni-Cd cell (1.2V) requires 1V extra for proper charging. So if a power source of 6V is used for the circuit, it is just sufficient for charging since the circuit and also the LED consume some power. An AC adaptor providing 6V and 500mA current can be used as the power source for the circuit if the cell-phone battery is of 3.6V. For using a 6V supply, suitable values of ZD1, R1 and R3 need to be used. Rechargeable batteries capable of holding more than 1A current can also be used as the power source.

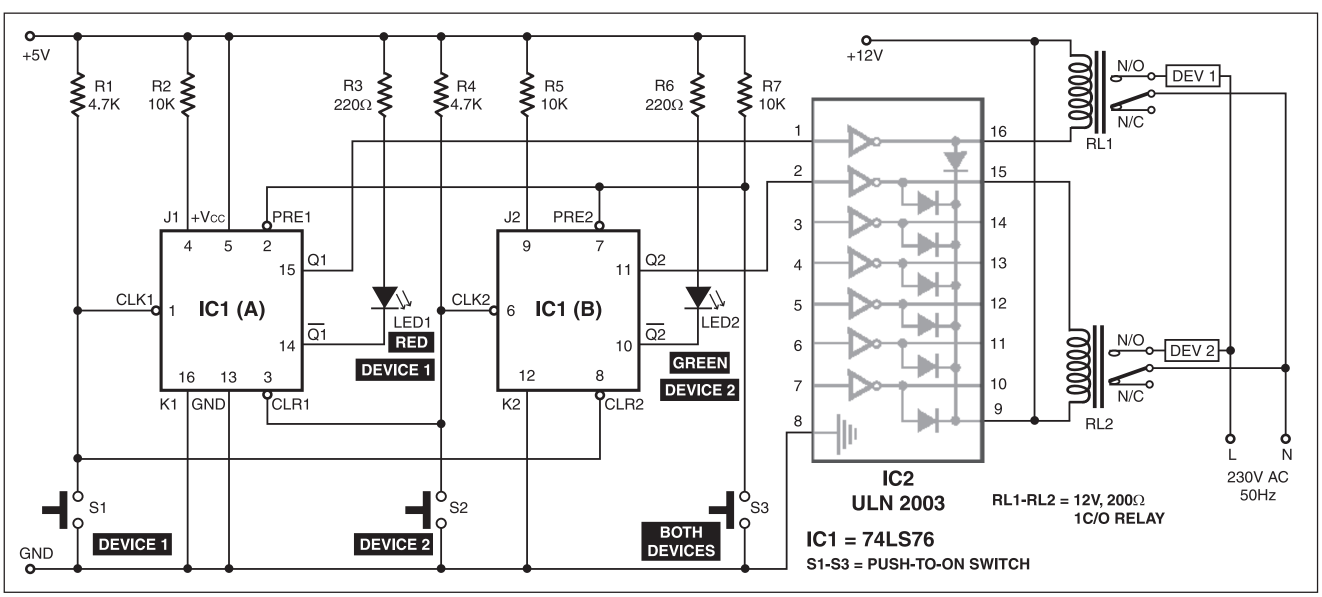
SMART FOOT SWITCH

JAYAN A.R.

Such jobs as jewel cutting and polishing require the workers to switch on/off two electrical appliances one after another repeatedly for two different services on the same workpiece. This is cumbersome as they need to fully concentrate on delicate handwork on precious jewels. Switching in such situations cannot be done by hand, and doing it by foot using ordinary switches is too tedious. This is mainly because of the difficulty in sensing and controlling the switch position by foot. Ordinary pushbutton switches make or break a contact momentarily, and they cannot hold the keypress status. You need a bistable multivibrator with two independent trigger inputs to solve this problem.

Here's a smart foot switch based on dual negative-edge triggered master slave JK flip-flop IC 74LS76 (IC1). J1 and J2 inputs are conneted to 5V through resistors R2 and R5 (each 10k), respectively. K1 and K2 inputs are grounded. Preset pins 2 and 7 are shorted and connected to 5V via resistor R7 (10k). Push-to-on switch S3 connected to the preset inputs is also grounded. Clock and clear inputs of the two flip-flops are cross-connected, i.e. CLK1 (pin 1) is conneted to CLR2 (pin 8) and CLR1 (pin 3) is connected to CLK2 (pin 6). Clock input pins 1 and 6 are pulled up high through resistors R1 and R4 (each 4.7k), respectively.

Push-to-on switches S1 and S2 are connected between clock and ground of the flip-flops. Switch S1 activates device 1, while switch S2 activates device 2. Switch S3 activates both device 1 and device 2 simultaneously. Device status is indicated by LED1 and LED2. Glowing of LED1 and LED2 indicates that device 1 and device 2, respectively, are in on condition. The LEDs are connected from +5V to Q1 (pin 14) and Q2 (pin 10) of IC1 through resistors R3 and R6, respectively.



Initially when the power supply is switched on, Q1 and Q2 outputs of the JK flip-flops are at low level (logic 0). When switch S1 is pressed for the first time, the high level (logic 1) present at J1 input is transferred to Q1 output on the trailing edge of clock (CLK1). The high level (logic 1) at Q1 activates relay RL1 through pin 16 of IC ULN2003 (IC2), turning on device 1 via its normally-opened (N/O) contacts. Clock CLK1 of flip-flop IC1(A) is also connected to clear input CLR2 of flip-flop IC1(B) so as to clear it asynchronously. Switch debounces don't affect the circuit as the same J1 state is being transferred to Q1 output on succeeding trailing edges. At the same time, device 2 is switched off.

When switch S2 is pressed, flip-flop IC1(A) gets cleared via CLR1 and the high state of J2 input of flip-flop IC1(B) is transferred to its Q2 output on the trailing edge of clock (CLK2). This high level (logic1) activates relay RL2 through pin 15 of IC2, turning on device 2 via its N/O contacts. At the same time, device 1 is switched off.

Now if you want to turn on both the devices simultaniously, press switch S3 momentarily. Switch S3 provides ground to preset inputs PRE1 and PRE2 of flip-flops IC1(A) and IC1(B), making their Q1 and Q2 outputs high, which energises both the relays turning on the two devices. LEDs glow to indicate that both the devices are 'on.'

Place all the three switches (S1 through S3) where you can easily press them by foot when required. The LEDs can also be mounted at a convenient location to know whether the devices are turned on.

DOORBELL-CONTROLLED PORCHLIGHT

T.A. BABU

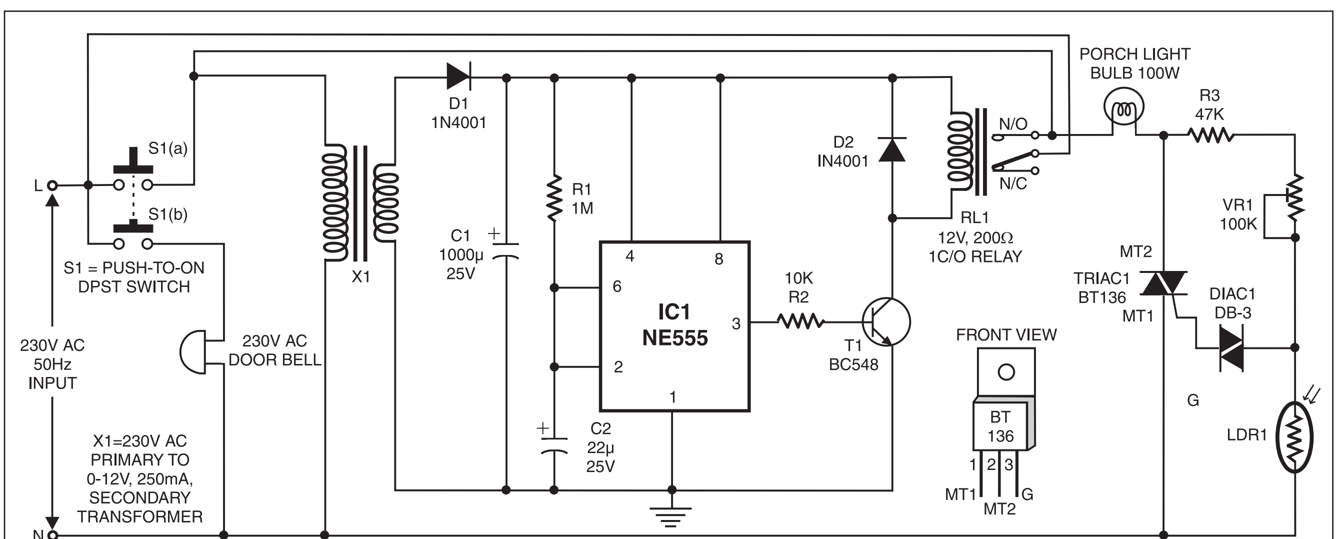
This add-on circuit automatically turns on your porchlight when your doorbell rings, so you can see the person ringing the doorbell at the doorstep. This also helps to guard against burglars, who usually press the doorbell switch to confirm that there is no one at home. By turning on the porchlight, the circuit will trick them into believing that someone is inside the home.

You can easily connect the circuit to your doorbell. The light remains on for around 20 seconds and then turns off. This duration is enough for you to find your way in the dark to open the door. However, duration can be varied by changing the RC components (R1 and C2).

When you momentarily press push-to-on DPST switch (S1), the AC mains is supplied to:

1. The doorbell via S1(b) and it rings.
2. Stepdown transformer X1 via S1(a) and it delivers 12V AC at its secondary. The secondary output is rectified by diode D1 and filtered by capacitor C1 to provide the required DC. The DC voltage triggers timer 555 (IC1) and its output at pin 3 goes high for the preset time. Simultaneosly, the relay energises and AC mains flows via its N/O contacts to switch on the porchlight bulb.

Triac 1 is wired as an automatic light controller to switch on the porchlight at night and switch it off during day. The conduction angle of triac 1 depends on the bias provided to the gate of the triac through diac 1, which, in turn, is controlled by preset VR1 and the light falling on LDR1.



AC MAINS VOLTAGE INDICATOR

P. VENKATA RATNAM

Here's a simple AC mains voltage indicator that uses three LEDs to indicate low, normal and high levels of AC mains voltage. The 5mm red LEDs are connected between the collectors of transistors T1, T2 and T3 and resistors R2, R4 and R6, respectively. Presets VR1, VR2 and VR3 are used to adjust the base voltages of transistors T1, T2 and T3, respectively.

The AC mains supply is stepped down by transformer X1 to deliver a secondary output of 18V AC, 250 mA. The secondary output is rectified by diode D1 and smoothed by C1 to give about 25V DC.

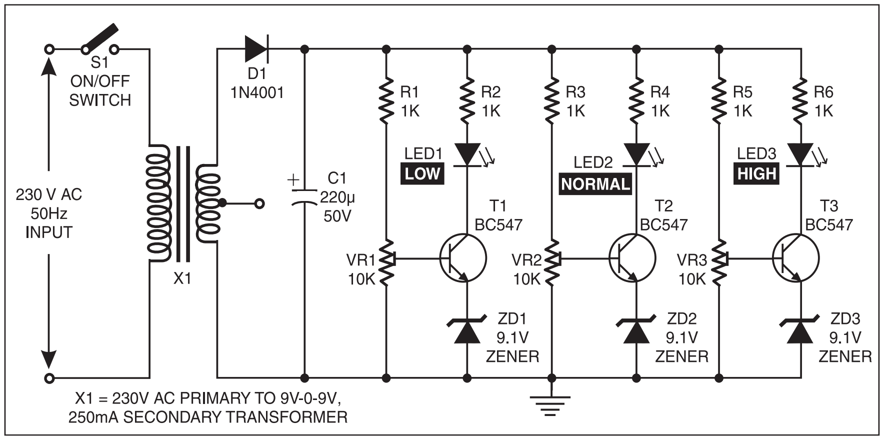


Fig. 1: Mains voltage indicator

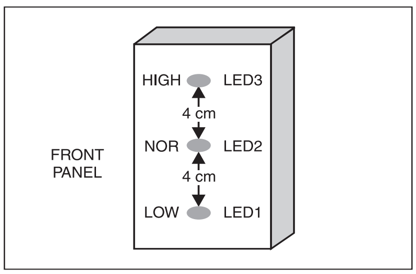


Fig. 2: Proposed panel for LEDs

This DC voltage varies proportionately with AC mains voltage, which is sensed by transistors T1 through T3.

Initially, set presets VR1 through VR3 towards ground to provide a low-resistance path across the base of transistors T1 through T3, respectively.

For setting the low voltage level, connect a manual AC voltage regulator (MVR) to the primary of transformer X1 and switch on power supply to the circuit by flipping switch S1 to 'on' position. Set the AC voltage of MVR to about 175V and slowly adjust VR1 until LED1 starts illuminating. When voltage across the base of transistor T1 reaches 9.7V (zener voltage 9.1V plus base emitter voltage 0.6V) by adjusting preset VR1, transistor T1 starts conducting. This causes LED1 to light up. LED1 stops glowing abruptly when the base voltage drops below the preset value. For setting the normal voltage level, set the AC voltage of MVR to about 200V and adjust VR2 slowly until LED2 starts illuminating.

For setting the high voltage level, set the AC voltage of MVR to about 230V and adjust VR3 slowly until LED3 starts illuminating. Now remove the MVR from the primary of step-down transformer X1 and connect the AC mains voltge to the monitor. Now the unit is ready for use.

If the mains voltage is above 230 volts, all the three LEDs continue to glow, indicating that the voltage is above 230 volts (high). If the voltage drops below 230 volts, LED3 goes off but LED2 and LED1 continue to glow, indicating that the voltage is above 200 volts but below 230 volts. If the voltage drops further below 200 volts, LED2 goes off but LED1 continues to glow, indicating that the voltage is above 175 volts but below 200 volts.

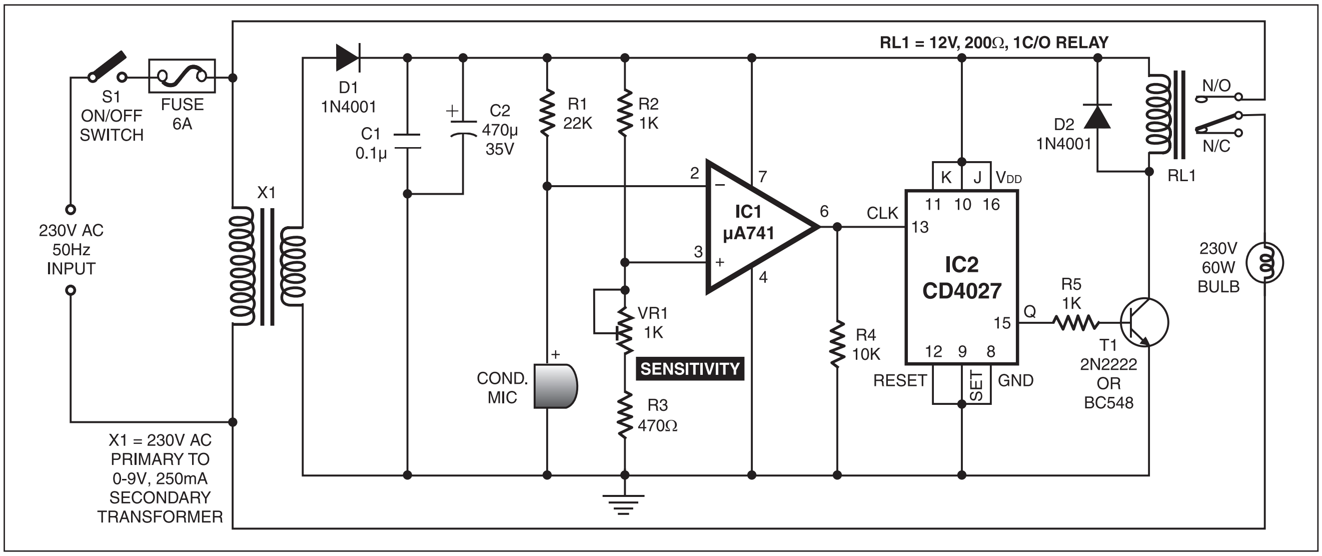
If the voltage drops below 175 volts, LED1 also stops glowing. At this stage, all the three LEDs are off, indicating that the voltage is below 175 volts.

To sum up, first, a high voltage (more than 230V) is indicated by glowing of all the three LEDs (LED1, LED2 and LED3). Second, normal voltage (200V-230V) is indicated by glowing of two LEDs (LED1 and LED2). Third, a low voltage (175V-200V) is indicated by the glowing of LED1 only. The circuit draws a total current of about 40 mA when all the LEDs glow. Mount all the LEDs on the front panel of the enclosure vertically in ascending order with a spacing of 4 cm between them as shown in Fig. 2. Fix the unit at a convenient place in the house to monitor the mains voltage.

SOUND-OPERATED LIGHT

RAJ K. GORKHALI

Here's a circuit that turns on your roomlight on detecting the sound produced when someone claps, tries to open your door or even inserts a key in the doorlock. This helps to guard against burglars as they assume that you are awake. The sensitivity control lets you adjust the sensitivity of the circuit to detect the intended sound level. The circuit switches the light alternately, i.e. if one sound pulse switches the light on, the next one switches it off. So you don't



need to go to the switchboard to switch off the light.

The AC mains supply is steppped down by transformer X1 to deliver a secondary output of 9V AC, 250 mA. The secondary output is rectified by diode D1 and filtered by capacitor C2 to give about 12V DC. The non-inverting input (pin 3) of op-amp IC1 is used as a reference voltage, fixed by adjusting preset VR1. The voltage at the inverting input (pin 2) of IC1 is same as that across the microphone. Switch S1 is power-on/off switch.

The sound sensitivity is adjusted by preset VR1. A high value of reference voltage at pin 3 of IC1 means a subtle sound is needed to change its output at pin 6. A low value of reference voltage at pin 3 of IC1 means a loud sound is needed to change its output at pin 6. Fix the reference voltage such that the output state of IC1 doesn't change with unwanted sounds.

In the absence of any sound, the inverting input voltgae is almost equal to the full DC voltage (about 12V DC), which ensures that output pin 6 of the op-amp is initially low. Since the JK flip-flop (IC2) has been wired as a toggle flip-flop and its output pin 15 is initially low, transistor T1 is in cut-off mode and relay RL1 remains de-energised. The AC power connected to the bulb via relay contact thus does not reach the bulb and it remains 'off.'

Now when you produce some sound near the condenser microphone, the current flows through the microphone and the voltage across the microphone goes down from 12V DC via the potential divider formed by resistor R1 and the microphone. If the sound is loud enough to bring the voltage at the inverting input below the reference voltage at the non-inverting input, output pin 6 of the op-amp (IC1) goes from low to high. This low-to-high going pulse triggers the flip-flop (IC2) at clock pin 13 and its output pin 15 goes high. Now the relay energises and the bulb glows via its N/O contacts.

Producing anothor sound causes a low-to-high transition at output pin 6 of the op-amp (IC1). This low-to-high going pulse triggers the flip flop at clock pin 13 and its output pin 15 goes low. Now the relay de-energises and the bulb goes off via its N/C contacts.

This way, the bulb glows alternately if there are recurrent sound pulses.

LOW-COST ELECTRONIC QUIZ TABLE

VINOD C.M.

Here is a simple, low-cost quiz table for four game participants. It de-termines the contestant who first presses the switch (S1 through S4) to answer a question and locks out the remaining three entries. Simultaneously, the respective audio alarm sounds and the bulb glows. The quiz table can be used for more number of contestants simply by adding buzzers, bulbs, MOSFETs and diodes. Besides, it provides an option for varying the time for which an individual buzzer and the corresponding bulb should be 'on' after a particular competitor has pressed the pushbutton. These timings can be set by presets VR1 through VR4 as required.

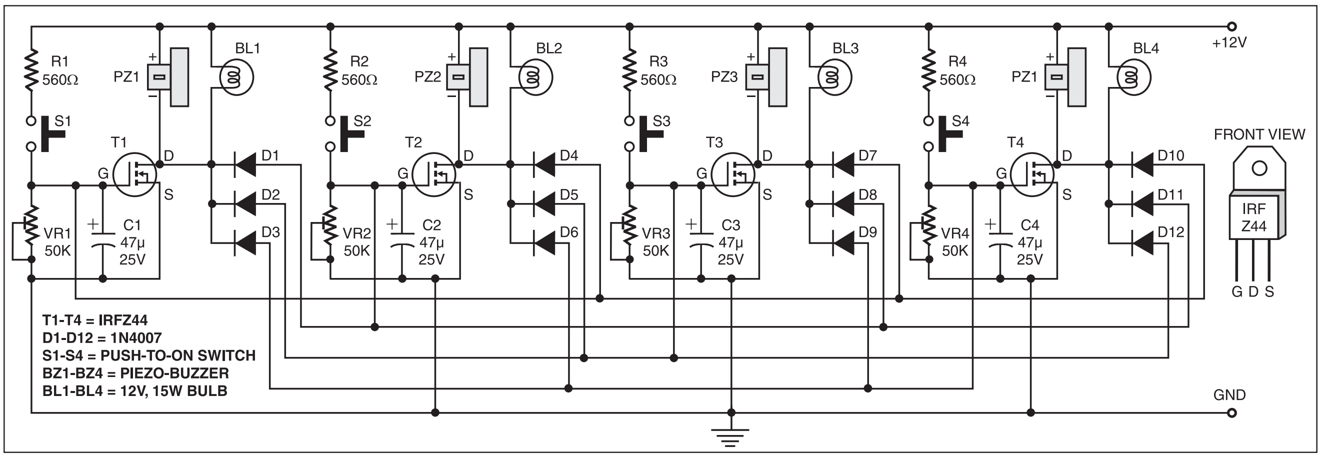


Fig. 1: Schematic of low-cost electronic quiz table

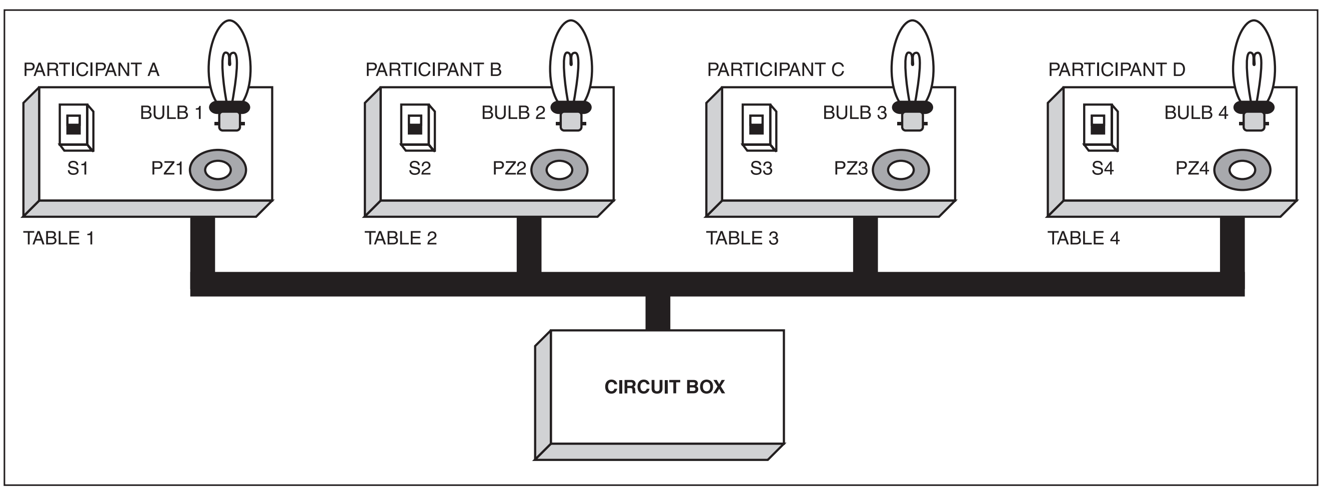


Fig. 2: Set-up for electronic quiz table

The circuit works off 12V, 1.5A power supply. The current rating of the power supply should be according to the load (wattage of bulbs). For higher-wattage bulbs, use power supply of a higher current rating.

If participant A presses switch S1, MOSFET T1 is triggered and the corresponding bulb BL1 (connected between drain of the MOSFET and 12V supply) glows and simultaneously piezobuzzer PZ1 connected in parallel to bulb BL1 sounds for the preset time. At the same time, capacitor C1 charges up to 12V, which then discharges through preset VR1. The discharging time of capacitor C1 is decided by preset VR1. For example, if preset VR1 is set for a resistance of 4.7k, it will give a delay of approximately 4 seconds, meaning that buzzer PZ1 and bulb BL1 will be 'on' for 4 seconds. It also indicates that participant A is the first to press his switch. Even if any other participant, say, participant B, presses switch S2 after participant A has already pressed switch S1, buzzer PZ2 and bulb BL2 will not function since MOSFET T2 has no gate voltage to trigger because it is grounded through R2 and D1.

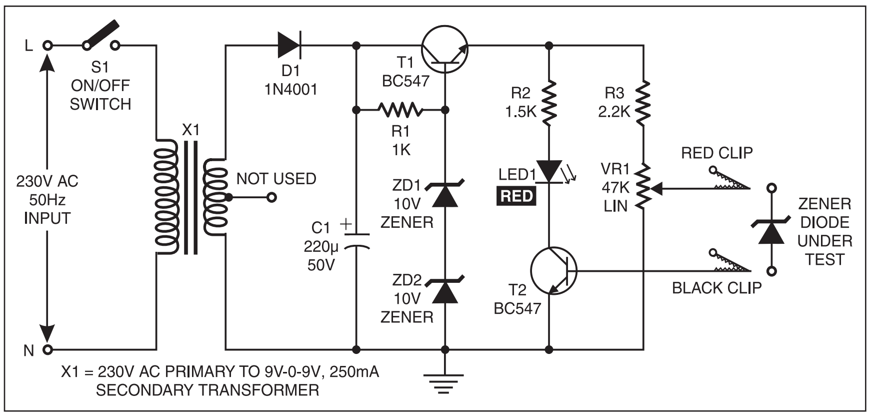
The same principle applies for other contestants as well. Instead of bulbs, you can also use a group of LEDs. Fig. 2 shows the set-up for electronic quiz table.

ZENER DIODE TESTER

P. VENKATA RATNAM

This zener diode tester can be used to check zener diodes of 3.3V to 18V. The breakdown voltage of the unknown zener diode is indicated on the precalibrated dial of potmeter VR1. The tester can also identify the polarity of zener diodes.

The power supply section comprising transformer X1, rectifier diode D1, filter capacitor C1, resistor R1, transistor T1 and zener diodes ZD1 and ZD2 provides approximately 20V DC stabilised voltage to the sensor section. The sensor circuit comprises resistors R2 and R3, potmeter VR1, red LED1 and transistor T2.



When linear potmeter VR1 is adjusted such that the voltage at its wiper arm (red crocodile clip) exceeds the breakdown voltage of the zener diode, the zener diode conducts and applies the bias voltage at the base of transistor T2, which causes red LED1 to light up. When the voltage at the wiper arm (red clip) is less than the breakdown voltage, the zener diode does not conduct and red LED1 does not glow.

For calibration of the zener diode tester, initially set the pointer knob of potmeter VR1 towards zero-resistance position. Short red clip of the potmeter and black clip of the transistor and switch on the tester. Rotate the pointer knob of potmeter VR1 slowly in clockwise direction until LED1 just starts to glow. Mark this setting of the knob on the paper dial as 0V.

Now connect a known zener diode of 3.3V between both the clips (red clip to the cathode and black clip to the anode of the zener diode) as shown in the figure. Rotate the knob of potmeter VR1 further in clockwise direction until LED1 just starts to glow. Mark this setting of the knob on the paper dial as 3.3V. Likewise, calibrate the dial of potmeter VR1 for other values of zener diodes by connecting known zener diodes to the tester. Now the tester is ready for use.

For testing an unknown zener diode, connect it across the clips in correct polarity and rotate the knob of potmeter VR1 until red LED1 just starts to glow. The voltage shown by the pointer knob on the dial at this setting is the breakdown voltage value of the zener diode under test.

If the zener diode is connected in reverse polarity (red clip to the anode and black clip to the cathode), the LED glows brightly at all settings of the knob above the zero reading, indicating that the zener diode is wrongly connected. The anode and cathode terminals of rectifier diodes can also be identified in this way. Do not touch the clips while testing.

HIGHWAY ALERT SIGNAL LAMP

D. MOHAN KUMAR

Here is a signal lamp for safe highway driving. The lamp automatically emits brilliant tricolour light when a vehicle approaches the rear side of your vehicle. It emits light for 30 seconds that turns off when the approaching vehicle overtakes. The ultra-bright blue, white and red LEDs of the signal lamp emit very bright light to alert the approaching vehicle's driver even during the day, giving addi-

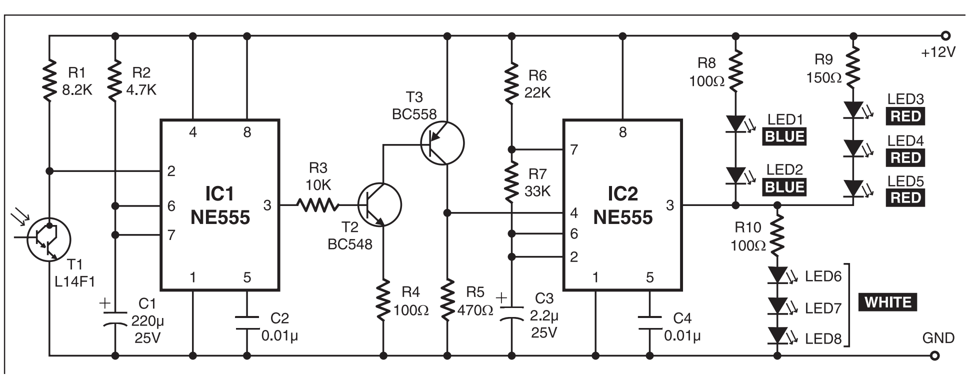


Fig. 1: Circuit diagram of highway alert signal lamp

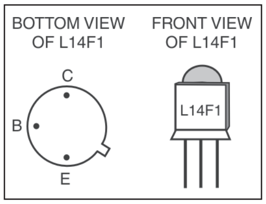


Fig. 2: Pin configuration

tional safety during night, or when you need to stop your vehicle on side of the highway. The circuit saves considerable battery power.

The circuit is built around two timer ICs NE555 (IC1 and IC2). IC1 is designed as a standard monostable, while IC2 is designed as an astable. Darlington phototransistor L14F1 (T1) is used as a photosensor to activate the monostable. The collector of phototransistor T1 is connected to trigger pin 2 of IC1, which is normally

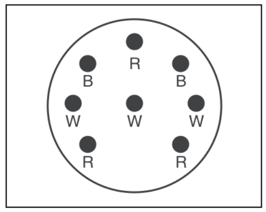


Fig. 3: Suggested arrangement of LEDs

kept high by resistor R1.

When headlight from an approaching vehicle illuminates the phototransistor, it conducts to give a short pulse to IC1, and the output of IC1 goes high for a period determined by resistor R2 and capacitor C1. The output of IC1 is fed to the base of transistor T2 via resistor R3. Transistor T2 conducts to drive transistor T3 and its collector goes high to take reset pin 4 of IC2 to high level. This activates astable IC2, which switches on and off the LED chain alternately. The intermittent flashing of LEDs gives a beautiful tricolour flashlight effect.

The circuit can be easily constructed on a small piece of general-purpose PCB. Fig. 2 shows the bottom and front views of Darlington phototransistor L14F1. The proposed arrangement of LEDs, which are soldered in a circular fashion on a general-purpose PCB, is shown in Fig. 3. Use a circular reflector for the LEDs to get brighter light. Fix the LED arrangement on the rear side of your vehicle, and the phototransistor where it is illuminated directly by the headlight of the approaching vehicle. 12V DC supply to the circuit, can be provided by your vehicle battery with proper polarity.

VARIABLE POWER SUPPLY WITH DIGITAL CONTROL

MANESH T. MATHEW

The most frequently used device in electronic workshops and laboratories is a universal power supply that provides a variable, fluctuation-free output. Here we present a variable power supply with digital control that is simple and easy to construct.

The circuit is built around an adjustable 3-terminal positive-voltage regulator IC LM317, CMOS decade counter IC CD4017, timer IC NE555 and 3-terminal fixed negative-voltage regulator LM7912.

The AC mains supply is stepped down by transformer X1 to deliver a secondary output of 12V-0-12V AC, 1A. The output of the transformer is rectified by a

full-wave rectifier comprising diodes

D1 through D4. Capacitors C1 through C4 are connected in parallel to rectifier

diodes to bypass undesired spikes and provide smooth and fluctuation-free power. Capacitors C5 and C13 are used as

filters to eliminate ripple. Here both negative and positive half cycles are used to obtain positive as well as negative

DC output. LED1, along with current-limiting resistor R1, is used for mains 'on' indication.

Timer IC NE555 (IC1) is wired as an astable multivibrator. It generates clock pulses when switch S2 is pressed. The output of IC1 is connected, via an RC network, to the clock input of counter IC CD4017 (IC2).

IC CD4017 is a decade ring counter. Each of its ten outputs goes high one by one when a clock pulse is received. The outputs of IC CD4017 are connected to the bases of transistors T1 through T10, respectively, as shown in the figure. LED3 through LED11 are used here to indicate the voltage levels. The collectors of transistors T2 through T10 are connected to presets VR1 through VR9, respectively, which are used to set the output voltage.

Adjustable voltage regulator IC LM317 (IC4) develops 1.25V nominal reference voltage (VREF) between its output and the adjustable terminal. The reference voltage appears across resistor R16. When the voltage is constant, a constant current flows through one of the output-setting variable resistors (VRset, VR1 through VR9), giving an output voltage at pin 2 of IC4 as follows:

VOUT=1.25(1+VRset/R16).

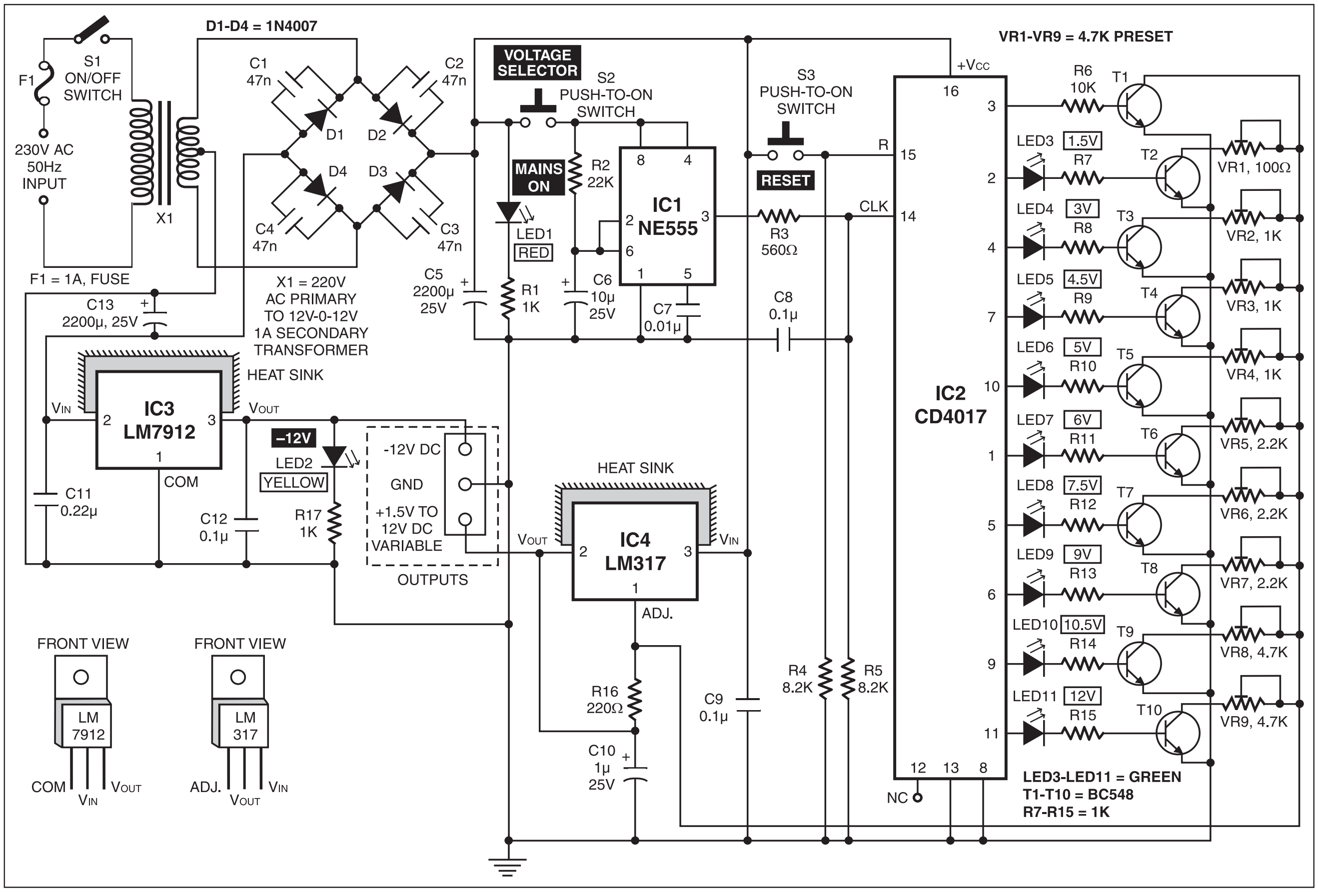
Presets VR1 through VR9 are adjusted to get the desired output voltage. The collector of transistor T1 is directly connected to ADJ terminal (pin 1) of IC4, so the output voltage of IC4 will be the voltage across fixed resistor R16, which is equal to 1.25V. When switch S3 is pressed, pin 3 of IC2 goes high and the output voltage becomes 1.2V.

When switch S2 is pressed, the output of IC1 goes high. As a result, the outputs of IC2 go high one by one as a ring counter. Since presets VR1 through VR9 are connected at the collectors of transistors T2 through T10, respectively, different output resistances appear between the adjustable and ground terminals of IC4, resulting in different output voltages. By using a properly calibrated digital multimeter you can easily adjust the presets to obtain 1.5V to 12V.

A fixed, negative 12V DC can be obtained by using fixed, negative-voltage regulator IC LM7912 (IC3). Thus the power supply unit can be used for circuits requiring both negative and positive DC voltages.

When CD4017 is reset by pressing switch S3, the output voltage becomes

1.2V and all the voltage-indication LEDs turn off.



Assemble the circuit on any general-purpose PCB and enclose it in a suitable cabinet. Use suitable heat-sinks for regulators IC3 and IC4. Since pin configurations of the regulators are different, never fix both regulators on the same heat-sink. For S2 and S3, using microswitches will enhance the beauty of the unit. LED2 is used to indicate the negative 12V DC voltage.

Readers' comments

I am very happy about your suggestion of the use of regulator LM317. However, there is no provision of applying different input voltages to get different output voltages. The input voltage supply from mains transformer after rectification is directly connected to pin 3 of regulator LM317. I tried with an output load of 10.5V, 350 mA for 12V input (fixed) to regulator LM317 and the regulator was heated normally. Please tell me a simple way to apply different input voltages to get different output voltages at approx. 1A load.

Ankana Mukherjee
Through email

The author, Manesh Mathew, replies: LM317 is used as a variable voltage regulator to achieve different output voltages at pin 2 according to different voltages applied at pin 1. This is done through a digital control, as explained in the circuit. At pin 3, apply an input voltage that is approx. 3V above the maximum output voltage one requires, i.e., to get a regulated voltage of 12V at 1A, one has to apply a minimum input voltage of 15V, 1A. (Refer to the specifications of the IC for the maximum input that can be applied to get the regulated output.) This is done to compensate for the voltage drop in the regulator and input voltage variations. In this circuit, the input voltage of the regulator is kept constant at 15V for getting 12V and below, as per one's requirements.

SIMPLE SECURITY SYSTEM

PRAVEEN KUMAR M.P.

Here's a simple home security system that sounds an alarm whenever somebody enters your house through the gate. The circuit consists of transmitter, receiver and alarm sections. The transmitter and receiver sections are fitted on the compound wall pillars to which the gate is attached, while the alarm circuit is mounted inside the house. The transmitter continuously transmits IR rays, which are incident on the receiver. When anyone passes

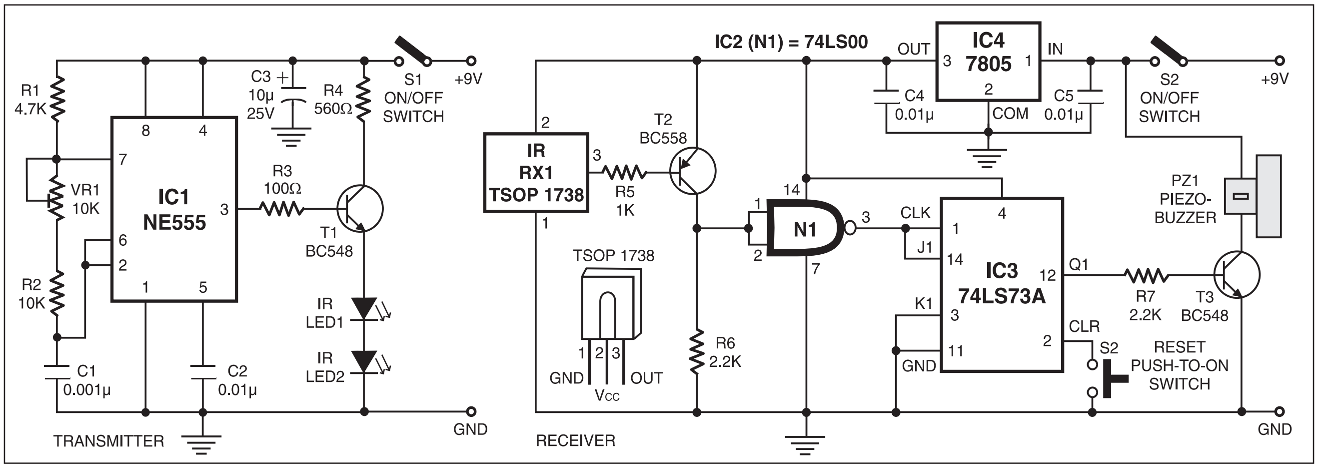


Fig. 1: Circuit diagram of transmitter and receiver of simple security system

through this continuous flow of IR beam, this is sensed and the alarm sounds, indicating that somebody has opened the front gate and entered the compound. The alarm will be 'on' until the reset pushbutton is pressed.

The transmitter is built around timer NE555 (IC1), which is wired as an astable multivibrator to oscillate at a frequency of 38 kHz. The output of IC1 is connected (via resistor R3) to the base of transistor T1. Transistor T1 drives both IR LEDs (LED1 and LED2). VR1 is used for adjusting the transmitting frequency.

The IR beams transmitted by LED1 and LED2 are incident on infrared receiver module RX1 of the receiver section, which produces a low output if the IR beam is interrupted by someone. Transistor T2 becomes forward biased and the output of IC2 goes low. The low output of IC2 is fed to the clock input of the JK flip-flop (IC3). The JK flip-flop acts as a latch. Its high output drives piezobuzzer PZ1 via transistor T3 and the buzzer sounds. To stop the alarm, you have to press reset switch S2.

Mount the transmitter and receiver units on the pillars of the gate. Ensure that ambient light does not reach the units to cause false alarm.

Readers' comments

Circuit is not working even though I have used the same components as given in the article. What may be the problem in my circuit? Does it require any change or correction?

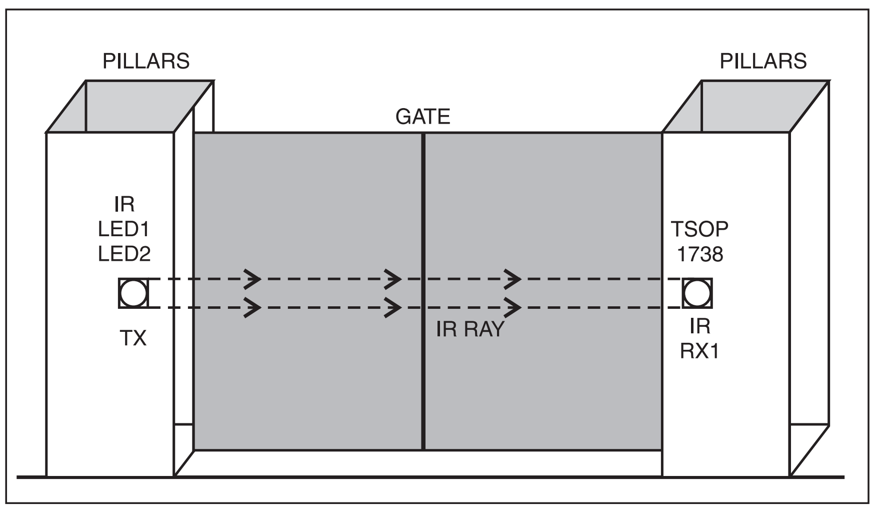
Akhilesh Mogra
Udaipur

The author, Praveen Kumar M.P., replies:

This may be due to the misalignment of transmitter and receiver sections. In order to eliminate this problem, the following steps may be taken:

1. Construct the transmitter as given in the article and a portion of the receiver as shown in the figure here. Use a shielded cable for connecting the IR eye (IR RX1) and a regulated 5V supply for the receiver section.

2. IR RX1 (TSOP1738) works in the IR region of light, i.e., at about 38 kHz. The output of the transmitter is tuned to this frequency by adjusting preset VR1.



3. Place IR RX1 near IR LEDs such that they face each other and are in line of sight. This will allow the IR rays emitted by IR LEDs to fall on IR RX1. Adjust preset VR1 slowly, using a screwdriver, until LED3 glows. Now the transmitter and the receiver are aligned correctly. When RX1 is taken away from IR LEDs, LED3 must stop glowing.

4. Now construct the remaining portion of the receiver circuit as given in the article.

The circuit should now work satisfactorily.

Fig. 2: Fitting of transmitter and receiver at the gate

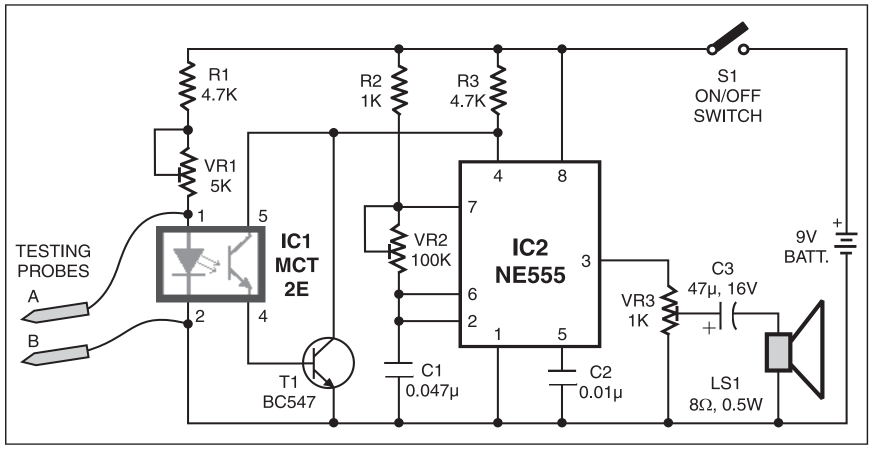
LOW-RESISTANCE CONTINUITY TESTER

PRADEEP G.

Using this circuit you can check con-tinuity of low-resistance paths such as PCB tracks, small coils, intermediate-frequency transformers (IFTs) and low-resistance transformers. However, you can't check semiconductors with this tester. The tester works off a 9V battery.

The tester is built around optocoupler MCT2E (IC1) and timer IC 555 (IC2) that is wired in free running mode. Optocoupler MCT2E is used here as a continuity sensor. Testing probes A and B are connected to pins 1 and 2 of optocoupler IC1. The phototransistor inside the optocoupler is connected to transistor BC547 (T1) to form a Darlington pair, which improves the performance of the circuit.

When the probes are not shorted, the LED inside the optocoupler glows and the Darlington pair conducts to keep reset pin 4 of IC 555 at ground level and thus no sound is produced.



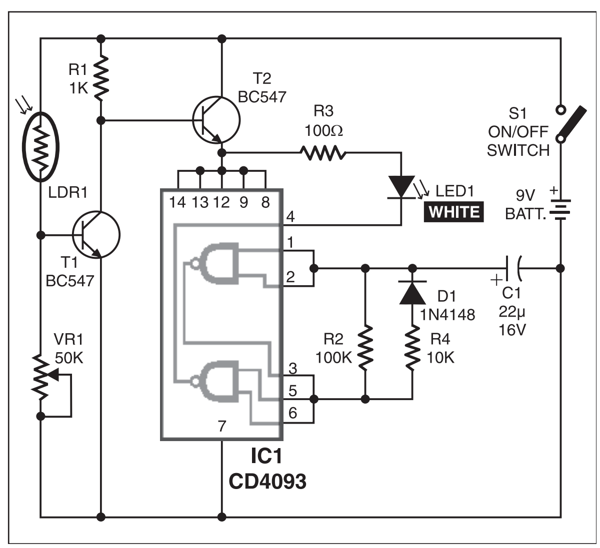
When probes are shorted via a low resistance, the LED stops glowing and the Darlington pair doesn't conduct. As a result, reset pin 4 of IC2 goes high to activate the loudspeaker, which generates a sharp audio tone. To minimise the current through IC1 when probes are not shorted, adjust VR1 until the circuit just stops sounding. The output tone and loudness can be varied by adjusting presets VR2 and VR3, respectively.

CHILD'S LAMP

D. MOHAN KUMAR

Here is a mini emergency lamp that you can use as a tabletop lamp in your child's study room. It is battery-operated and gives sufficient light for the child to move out of the room when power fails. The white LED in the circuit automatically turns on when light in the room goes off following a power cut. The LED gives a flashing light instead of glowing continuously to reduce power consumption.

The circuit comprises a light sensor and an LED flasher designed around CMOS IC CD4093 (IC1). The light sensor switch comprises a light-dependent resistor (LDR) and npn transistors T1 and T2. When ambient light is present, the low resistance of LDR1 drives transistor T1 into conduction. This keeps transistor T2 cut-off due to low base bias. The flasher circuit does not get power as long as ambient light falls on LDR1. When the resistance of LDR1 becomes high in darkness, transistor T1 stops conducting and transistor T2 starts conducting to turn on the LED lamp.



IC1 is designed as a simple oscillator using its gate 1 (comprising input pins 1 and 2 and output pin 3). The oscillator's external components comprise resistor R2 and capacitor C1. Diode D1 and resistor R4 help in rapid charging of capacitor C1. When capacitor C1 charge to around 50% of Vcc, output of gate 1 of IC1 goes low to discharge capacitor C1. The output from pin 3 of IC1 again goes high to charge capacitor C1 again. This cycle repeats and sets up an oscillation, which is given to gate 2 (comprising input pins 5 and 6 and output pin 4) of IC1. Gate 2 serves as a buffer to drive the white LED (LED1).

For the given values of resistor R2 and capacitor C1, the flashing rate of LED1 is one per second (1 Hz). It can be increased by decreasing the value of capacitor C1. Pin 14 of IC1 is Vcc and all the unused input pins are tied to the positive rail (pin 14) to prevent floating.

The circuit can be constructed on a small veroboard. Use a reflective holder for LED1, which should be directed downwards at an angle of 45 degrees to prevent direct viewing of LED1 which gives a high-intensity light that is harmful for eyes. Preset VR1 can be adjusted to control the sensitivity of LDR1. You can enclose the circuit in a plastic doll with LED1 as its headlamp to make it an attractive gadget for your child. Mount LDR1 such that ambient light falls on it directly.

CLAP-OPERATED ELECTRONIC SWITCH

DIPANJAN BHATTACHARJEE

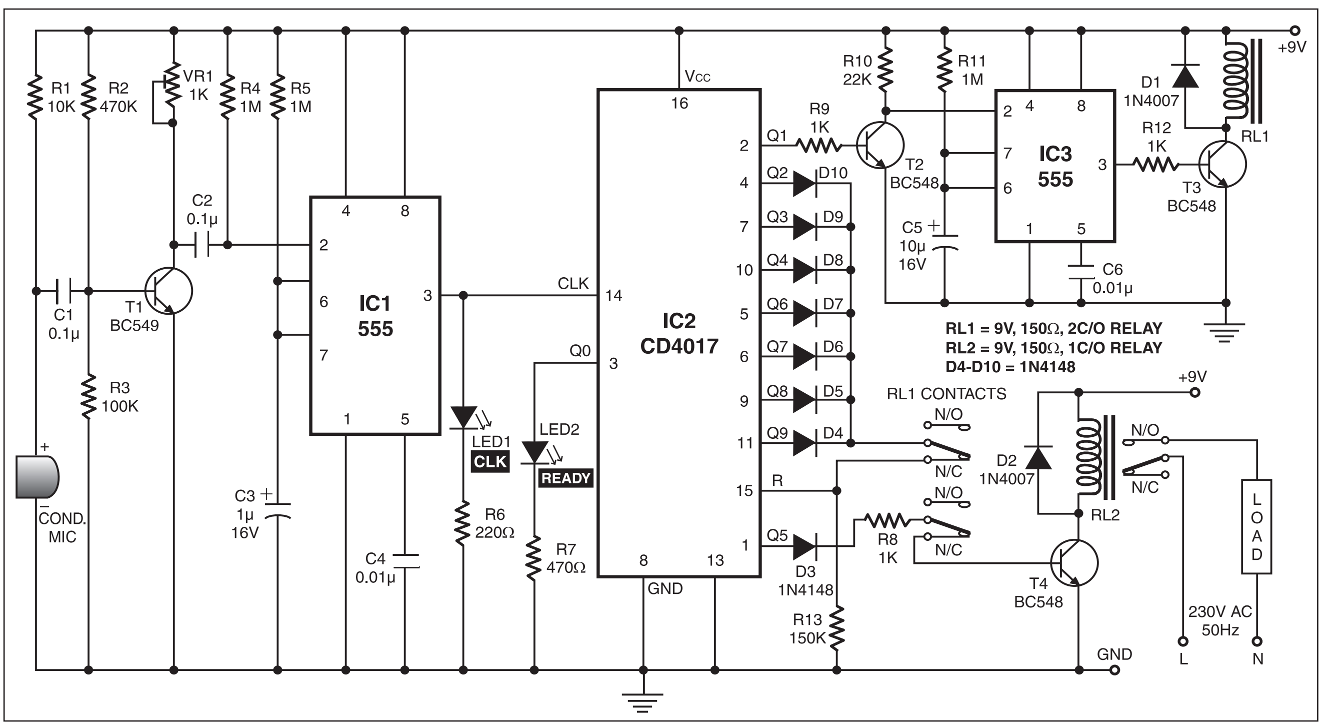
Here's a simple clap-operated electronic switch. Using this switch, you can turn on any appliance by clapping five times and turn it off by a single clap. The switch activates the appliance only if you apply the right clap code (five claps here) within the preset time (10 seconds). If you apply a wrong clap code (other than five claps) or you are unable to apply five claps within 10 seconds, the switch does not activate the appliance.

The circuit works off a 9V DC power supply. The condenser microphone converts the clap sound into an electrical signal. This electrical signal is amplified by transistor BC549 (T1). The amplified output is given to trigger pin 2 of monostable IC1, which produces a clock pulse at its output pin 3. The output of IC1 is fed to clock pin 14 of decade counter IC2.

On/Off Status of the Appliance for Every Clap

Claps	RL1	RL2	LED2	Appliance status
1	En	De	Off	Off
2	En	De	Off	Off
3	En	De	Off	Off
4	En	De	Off	Off
5	En	De	Off	Off
6	Off	Off	On	Off after 10 seconds

Note: En = energised; RL1 = Relay 1; De = de-energised; RL2= Relay 2



Initially, when the power is switched on, the Q0 output of IC2 is high and glowing of LED2 indicates that the switch is ready for use. All others outputs of IC2 (Q1 through Q9) are low. The Q5 output is used for activating the appliance via relay RL2.

At each clock (generated with each clap), the output of IC2 gets incremented. The Q1 output of IC2 is used to trigger the monostable (IC3). The output of IC3 is used to drive relay RL1. IC3 acts as a monostable multivibrator with a time period of approximately 10 seconds, which provides delay time to turn on an appliance even after completing the five claps before the preset time of 10 seconds. Contacts of relay RL1 separate the other output pins of IC2 (except Q5) from reset pin 15 for the preset time of 10 seconds.

When all the five claps are applied within 10 seconds, the Q5 output of IC2 goes high and relay RL2 energises to turn on the appliance just after the de-energisation of relay RL1. The table shows the on/off condition of the appliance for every clap.

When you apply a wrong clap code, the high output of IC2 resets it via its pin 15 and the appliance doesn't turn on.

LIGHT-CONTROLLED DIGITAL FAN REGULATOR

V. GOPALAKRISHNAN

It is very difficult to trace the switchboard in a dark room at night. Here is a torchlight-operated switch that allows you to control the fan speed remotely from your bed. The fan speed can be varied by the number of times you focus the torch light on the light-dependent resistor of the circuit—same way as you control the fan speed by rotating the regulator to different number positions.

Fig. 1 shows the circuit of light-controlled digital fan regulator. It comprises timer NE555 (IC1), decade counter 7490 (IC2), BCD-to-7-segment decoder/driver 7447 (IC3), common-anode 7-segment display (DIS1), BCD-to-decimal decoder 7442 (IC4) and a hex inverter (IC5). The fan regulator is triggered when torchlight falls on light-dependent resistor LDR1 and its resistance goes low.

The monostable (IC1) is wired such that its time period is adjusted to 1.3 seconds. The monostable clocks are counted with decade counter IC2. The Q0 through Q3 outputs of decade counter IC2 are

Fan Control With Torch Light Focused on the LDR

No. of focus on LDR1	Display DIS1	Energised relay No	Fan speed
0	0	—	Off
1	1	1	1 (min)
2	2	—	Off
3	3	2	2
4	4	—	Off
5	5	3	3
6	6	—	Off
7	7	4	4

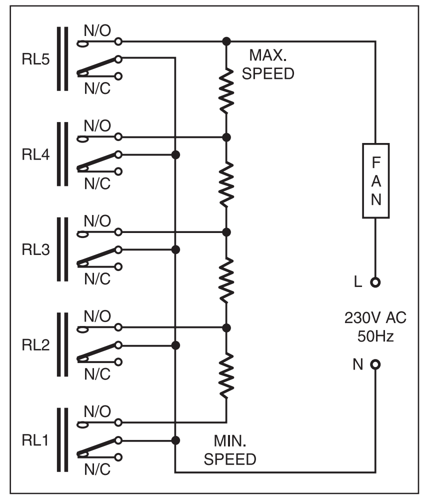


Fig. 2: Relay contacts of fan regulator resistors

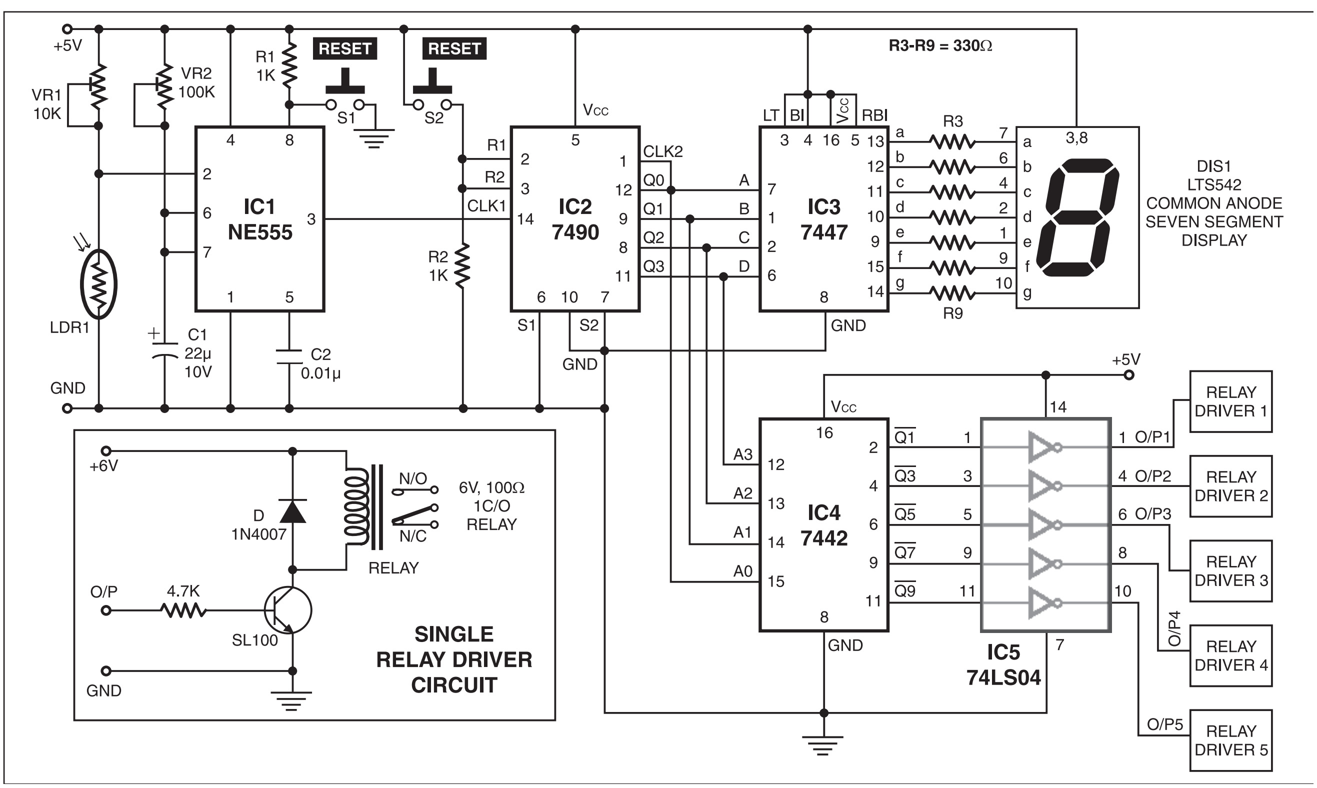


Fig. 1: Circuit of light-controlled digital fan regulator

given to IC3 and IC4.

The outputs of IC3 are given to the 7-segment display, while the outputs of IC4 are given to inverters. A common-anode, 7-segment display (DIS1) shows the number of times you focus the torchlight on LDR1. If it is an even number, the fan will be off. With the increase in odd numbers, the speed of fan increases.

The odd-numbered outputs of decoder IC4 (Q1, Q3 Q5, Q7 and Q9) go to the corresponding relay driver circuits via hex inverter IC5. The normally-opened (N/O) contacts of relays RL1 through RL5 are connected to regulating resistors as shown in Fig. 2.

The even-numbered outputs of decoder IC4 (Q0, Q2 Q4, Q6 and Q8) are not used. At these outputs, the fan turns off. Push-to-on switches S1 and S2 are used for initial resetting of monostable IC1 and decade counter IC2, respectively. For the circuit to work even in the presence of ambient light, for example, during daytime, LDR1 is made dark by covering it with an inked paper.

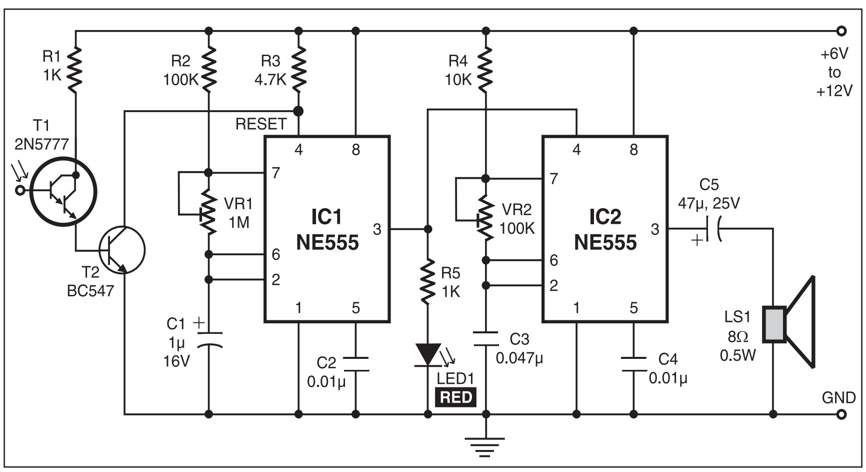
The digital fan regulator circuit (except relay driver circuits) works off 5V DC, while the relay driver circuits work off 6V DC.

SENSITIVE OPTICAL BURGLAR ALARM

PRADEEP G.

This optical burglar alarm uses two 555 timer ICs. Both the ICs are wired as astable multivibrators. The first astable multivibrator built around IC1 produces low frequencies, while the second astable multivibrator built around IC2 produces audio frequencies.

General-purpose Darlington phototransistor 2N5777 (T1) is used as the light sensor. To increase the sensitivity of the circuit, npn transistor BC547 (T2) is used.



Place phototransistor T1 where light falls on it continuously. Phototransistor T1 receives light to provide base voltage to transistor T2 . As a result, transistor T2 conducts to keep reset pin 4 of IC1 at low level. This disables the first multivibrator (IC1) and hence the second multivibrator (IC2) also remains reset so the alarm (loudspeaker LS1) does not sound.

When light falling on Darlington phototransistor T1 is obstructed, transistor T2 stops conducting and reset pin 4 of IC1 goes high. This enables the first multivibrator (IC1) and hence also the second multivibrator (IC2). As a result, a beep tone is heard from speaker LS1. The beep rate can be varied by using preset VR1, while the output frequency of IC2 can be varied by using another preset VR2.

The circuit works off a simple 6V-12V DC power supply.

WATCHMAN WATCHER

JAYAN A.R.

Here is a circuit that can be used in offices, stores, warehouses, etc dur-ing night to check whether the watchman of your establishment is on duty. For operation, it uses an existing telephone (e.g. in office or store) closest to the watchman's post. The watchman is given an audio alert signal by just ringing the office/store telephone once (minimum) from your residence or any other place, preferably using your mobile phone. The ring is detected by the given circuit and the watchman is also given a visual alert signal by a glowing lamp. The lamp remains 'on' for a duration of nearly 60 seconds soon after the ringtone. The watchman is given an instruction to register his presence by simply pointing his torch-light beam towards a wall-mounted LDR sensor unit (without lifting the handset off-cradle of the ringing telephone).

This is to be done within the time period during which the alert lamp glows. If he fails to do it within the permissible time, the circuit registers his absence by incrementing a count. If he does, the count remains unaltered.

Up to nine separate alert rings are considered here. The count displayed is the number of times the watchman failed to register his presence. The mobile phone records the called number and call time, and it can be used with the displayed count to get the timing details.

The telephone lines (TIP and RING) in the circuit are connected across optocoupler MCT2E (IC1) through a resistor-capacitor (R1-C1) combination. The diode in the optocoupler conducts only during ring pulses. The collector of the optocoupler transistor is normally off and a 5V signal is available here. This signal is connected to the trigger input of IC 555 (IC2) configured in monostable mode. The time constant of IC2 is set to nearly one minute (1.1RxC). Its output pin 3 is low during normal mode of operation and the relay is de-energised.

When the phone rings, the internal transistor of the optocoupler conducts to cause a high-to-low transition at trigger pin 2 of monostable IC2. Timer IC2 gets triggered on this trailing edge to energise relay RL1. This relay is used to switch on alert lamp L1. The circuit doesn't respond to additional trigger inputs for the set duration of the monostable. The caller may cut the phone call after hearing ringback tone from the called phone.

The sensor circuit formed using LDR1 activates another monostable 555 (IC6). LDR1 has a resistance of 2.2 kilo-ohms in daylight, which drops below 50 ohms when torchlight beam falls on it. (An LDR of nearly 2cm diameter has been used in this circuit.) Comparator LM358 (IC5) compares the level set at pin 3 (nearly 1V, set using a 10k pot) with the level at pin 2.

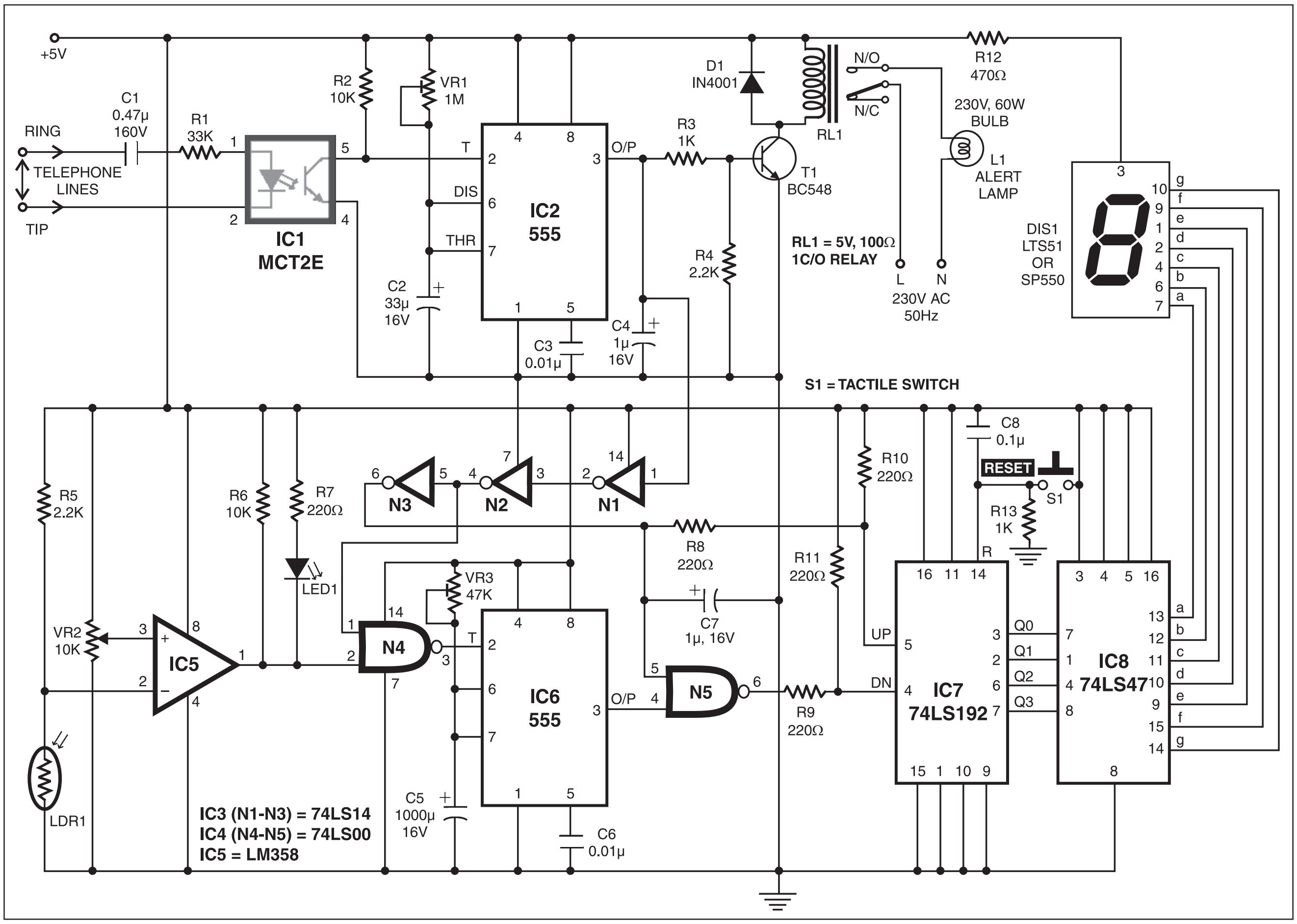
When no light is falling on LDR1, its voltage is above 1V and IC5 has a low output at its pin 1. When light is falling on LDR1, its voltage drops below 1V and IC5 output at its pin 1 becomes high. This low-to-high transition is NANDed with the output of monostable IC2 (via inverters gates N1 and N2) to form the trigger signal for monostable IC6. So the trigger input is normally high, which falls when torchlight beam is focused on LDR1. It returns to high state when torchlight is switched off. So the torch is used as a remote for triggering monostable IC6 and this triggering is enabled only when alert lamp L1 is 'on.'

Monostable IC6 has a time constant of nearly one minute (1.1RxC). It is used to form a down clock signal for 4-bit up-

Mode-Select Table of 74LS192

MRpin 14	PL pin 11	UPpin5	DNpin4	Mode
H	X	X	X	Reset
L	L	X	X	Preset
L	H	H	H	No change
L	H	_⌐	H	Count up
L	H	H	_⌐	Count down

Note: X = Don't care



/down-counter 74LS192 (IC7). Counter IC7 has two separate clocks for up and down counts (refer to the table). For correct counting, it needs one clock line to be high during high-to-low transition of the other clock line. Otherwise, it counts erratically.

To operate counter IC7, the voltage levels and timings of the two clock inputs (up and down) are to be properly adjusted. Both trigger inputs, i.e. up and down clocks, are asynchronous.

The output of monostable IC2 is filtered using capacitor C4 to remove unwanted transitions and inverted using Schmitt trigger inverter 74LS14 (IC3). This forms a signal with correct rising and falling edges. The inverted signal from pin 6 of gate N3 is used as the up clock.

Counter 74LS192 (IC7) is reset to zero state by making its reset pin 14 high through reset switch S1. The 7-segment, common-anode display DIS1 is driven through IC 74LS47 (IC8). When the phone rings, count '1' is displayed after nearly one minute. This happens if the watchman fails to focus the torchlight beam on LDR1.

If LDR1 receives light from the torch of the watchman within the allowed time period, the down clock remains high until the up clock is high. The counter counts up and then down, so, in effect, the count remains unchanged.

All components, except LDR1, are kept in a sealed cabinet with locking arrangement. Only LDR1 is wall-mounted and visible outside. This is done to avoid manual resetting of the counter. The circuit is to be powered by a battery to avoid resetting of the count during power failure.

The working procedure can be summarised as follows:

1. Initially, when the power supply is switched on, power-on-reset components C8 and R13 reset counter IC7 and the display shows '0.'

2. Now dial the telephone number (where parallel system is installed) from outside or from your mobile. For the first ring, relay RL1 energises and alert lamp L1 glows.

3.When alert lamp L1 is off, the counter is incremented by '1.'

4. If the watchman focuses the torchlight beam on LDR1 within the glowing time of alert lamp L1, the counter first counts up and then counts down and finally the display shows 0. This indicates that the watchman is present.

5. If the watchman focuses the torchlight beam on LDR1 after alert lamp L1 goes off, up-counting takes place and the display shows '1.' This indicates that the watchman is absent.

CELL-PHONE-CONTROLLED AUDIO/VIDEO MUTE SWITCH

T.K. HAREENDRAN

This cell-phone-controlled audio/video mute switch is highly useful in automobiles. The circuit automatically disconnects power supply to the audio/video system whenever the mobile handset is lifted off the holder for making or receiving a call. You can use any readily available cell-phone holder with some minor alterations or fabricate it yourself as shown in Fig. 1.

The circuit is wired around IC LM555 (IC1), the CMOS version of timer NE555, as shown in Fig. 2. IC1 is used as a medium-current line driver with either an inverting or non-inverting output. It can sink (or source) current of up to 50 mA only, so take

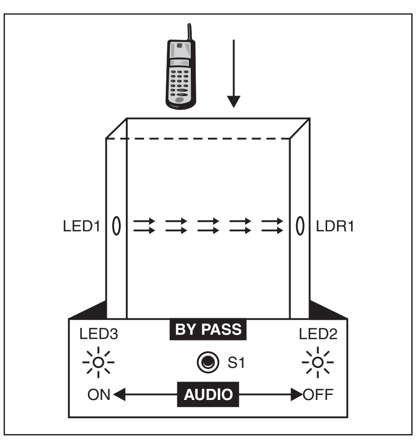


Fig. 1: Proposed cell-phone holder

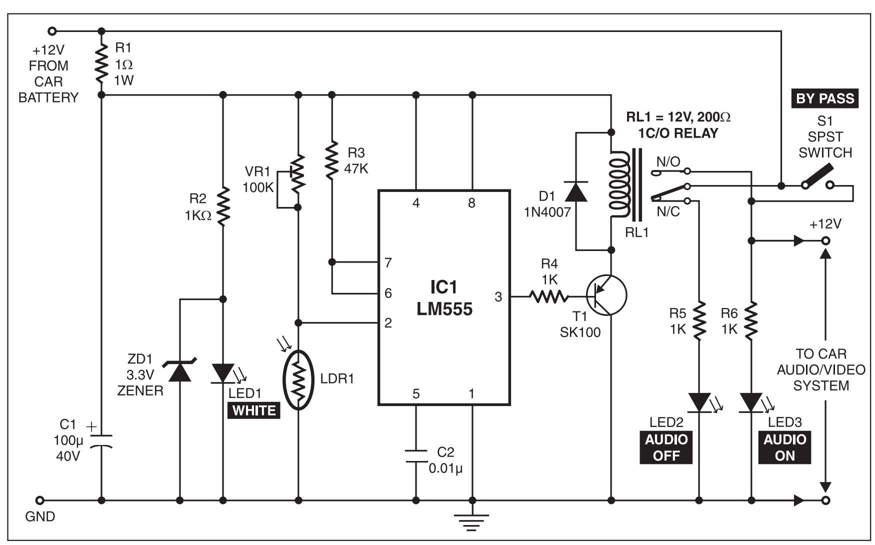


Fig. 2: The circuit of the cell phone-controlled audio/video mute switch

care while handling it. The audio/video system is connected to the circuit via normally opened (N/O) contacts of the relay.

When the cell phone is in its holder, LDR1 does not receive any light from white LED1 and its resistance is high. As a result, the voltage at pin 2 of IC1 remains high to provide a low output at pin 3. The low output of IC1 activates relay RL1 and the audio/video system gets power supply via its N/O contacts. LED3 glows to indicate that the audio/video system is 'on.'

When the handset is taken off the holder, light rays from LED1 fall on LDR1 and its resistance decreases. As a result, the voltage at pin 2 of IC1 decreases to provide a high output at its pin 3. The high output of IC1 deactivates relay RL1 and the audio/video system does not get power supply. LED2 glows to indicate that the audio/video system is 'off.'

Preset VR1 is used to control the sensitivity of the circuit. Zener diode ZD1 is used for protecting white LED1 from the higher voltage. The circuit works off a 12V car battery. Switch S1 can be used to manually switch on/off the audio/video system.

PANEL FREQUENCY METER

V. DAVID

Here's a simple panel frequency meter to measure the frequency of 230V AC mains. When you connect it to the 230V AC line, the display shows the line frequency. Generally, the line frequency is 50 Hz, which may vary from 48 Hz to 52 Hz. Beyond this frequency range, sensitive equipment may start malfunctioning.

The AC mains supply is stepped down by transformer X1 to deliver a secondary output of 9V-0-9V AC, 250 mA. The secondary output of the transformer is rectified by diodes D1 and D2, filtered by capacitor C1 and given to regulator IC1 to produce regulated 6V DC. 9V AC is also connected to pins 2 and 6 of IC2 via resistor R1. Timer IC2 converts the sinewave frequency sample of AC mains into a square wave that is more suitable for the circuit operation.

IC CD4093 (IC3) is used as an

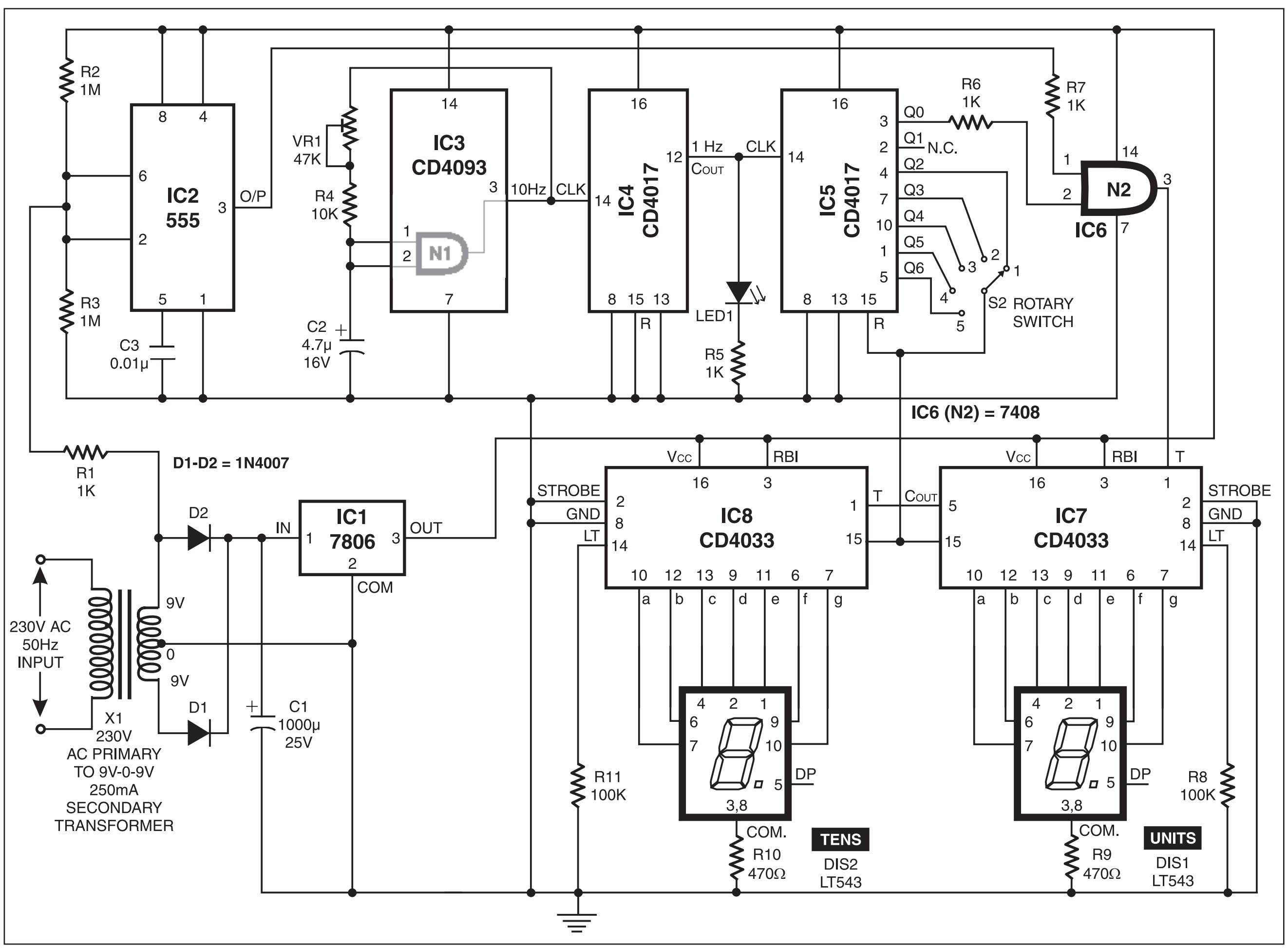


Fig. 1: The circuit of the panel frequency meter

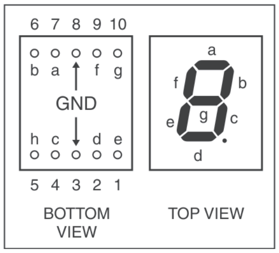


Fig. 2: Top and bottom views of LTS543 common-cathode, 7-segment displays

oscillator-cum-divider. The oscillator, wired around gate N1, produces 10Hz clock. Decade counter IC4 divides 10Hz clock by 10 to produce 1Hz clock. The output of gate N1 is fed back to its inputs via potentiometer VR1 and resistor R4. Capacitor C2 connected between the inputs of gate N1 and ground charges/discharges depending on the logic level at the output of gate N1. The values of VR1, R4 and C2 are selected to produce accurate 10Hz clock.

Decade counter IC CD4017 (IC4) divides the output of IC3 by 10 to provide one pulse per second. LED1 connected to pin 12 of IC4 gives one flash per second to indicate that the oscillator and the counter are working properly.

This 1Hz clock is fed to clock pin 14 of decade counter IC CD4017 (IC5), whose Q0 output is given to pin 2 and the square wave produced by IC2 is given to pin 1 of AND gate N1. Therefore, the unknown frequency of AC mains line, applied to pin 1 of AND gate N1, passes through it for only one second and the number of clocks per second are counted by IC7 and IC8.

Decade counters/7-segment decoders IC7 and IC8 are cascaded to drive common-cathode, 7-segment displays DIS1 and DIS2 (each LTS543). DIS1 shows units place of the frequency and DIS2 shows tens place. The top and bottom views of LTS543 common-cathode, 7-segment displays are shown in Fig. 2.

This is an auto-reset circuit. You can select the reset time of 1 second through 5 seconds using rotary switch S2, which is connected to reset pins of IC5, IC7 and IC8. For long-time display of the frequency, keep the knob of rotary switch S2 towards fifth position. Keeping rotary switch S2 to first position (minimum reset time) allows you to instantly see any variation in the supply frequency on the display. Also, while adjusting the generator frequency to mains frequency, keep rotary switch S2 towards first position.

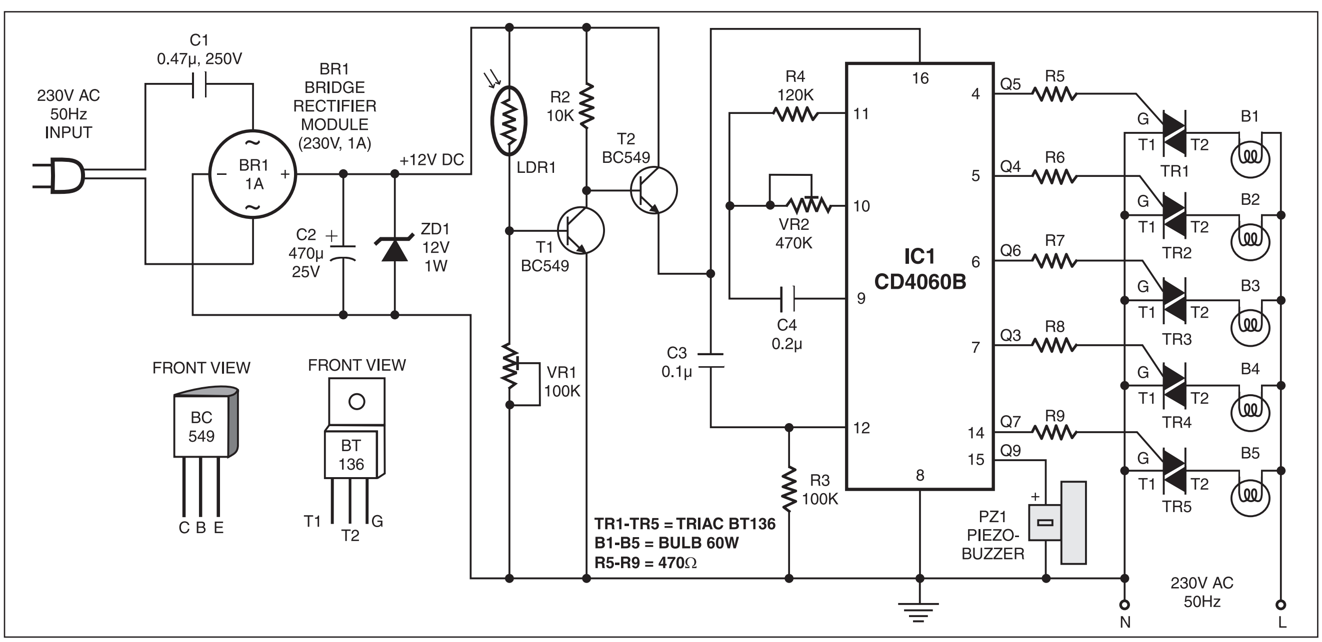
RANDOM FLASHING X-MAS STARS

D. MOHAN KUMAR

This novel colour display adds glitter to your Christmas and New Year celebrations. Unlike the usual running or flickering pattern, the lamps flash randomly to give a more attractive lighting effect. The display is fully automatic: the bulbs remain switched off during day and turn on like twinkling stars in the evening.

The circuit is designed around 14-stage, ripple-carry binary counter IC CD4060B. This IC has an internal oscillator and 14 bistable stages that are cascaded in series fashion. When the first bistable gets the clock signal from the oscillator, it turns on to drive the second bistable, which, in turn, drives the third bistable and so on. Each bistable divides the input signal by two. Out of a total of 14 possible outputs only ten outputs have been brought out on the external pins. This circuit uses only six outputs. Values of resistor R4, VR2 and capacitor C4 determine the basic frequency of the built-in oscillator. By adjusting VR2, the flashing rate of the lamp can be changed.

Five outputs of IC1 are connected to the gates of triacs TR1 through TR5 (BT136) via current-limiting resistors R5 through R9, respectively. When the outputs of IC1 go high, the triacs get gate current to switch on the lamps (230V, 60W). The lamps turn on/off in a random



fashion, giving a display pattern that is more attractive than the monotonous pattern of chaser lamps.

The use of a light-dependent resistor (LDR) automates the working of the circuit, so the user doesn't have to manually switch on/off the lamps daily. LDR1 and npn transistors T1 and T2 (each BC549) form the automatic switch. The resistance of LDR1 is low in daylight and increases in darkness. So during daytime, LDR1 gives base current to transistor T1, which conducts to pull the base of transistor T2 to low and hence transistor T2 remains cut-off. As a result, the rest of the circuit remains inactivated. During night, transistor T1 turns off to drive transistor T2 and provide supply voltage to IC1. Potmeter VR1 is used to adjust the sensitivity of the LDR.

The circuit is powered directly from the AC mains via capacitor and bridge rectifier module (BR1). Absence of transformer for the power supply reduces the cost as well as the size of the unit. Capacitor C1 drops the AC voltage to a safer level. The bridge rectifier module rectifies the AC and capacitor C2 smoothes the resulting DC. Zener diode ZD1 regulates the output voltage to 12V DC.

The circuit can be assembled on a small PCB or a breadboard. Mount the triacs with sufficient space in between to avoid short circuit. You can add a musical tone generation circuit to this circuit so that it sings a musical song when the lamps flash. It can be connected directly between output pin 15 of IC1 and ground, replacing the piezobuzzer. Connect ground rail (negative) of the circuit to the neutral line. Enclose the circuit in a shockproof plastic case.

Caution. Take extreme care while testing the circuit since most of its parts are at mains potential and hence lethal.

PC-BASED DC MOTOR SPEED CONTROLLER

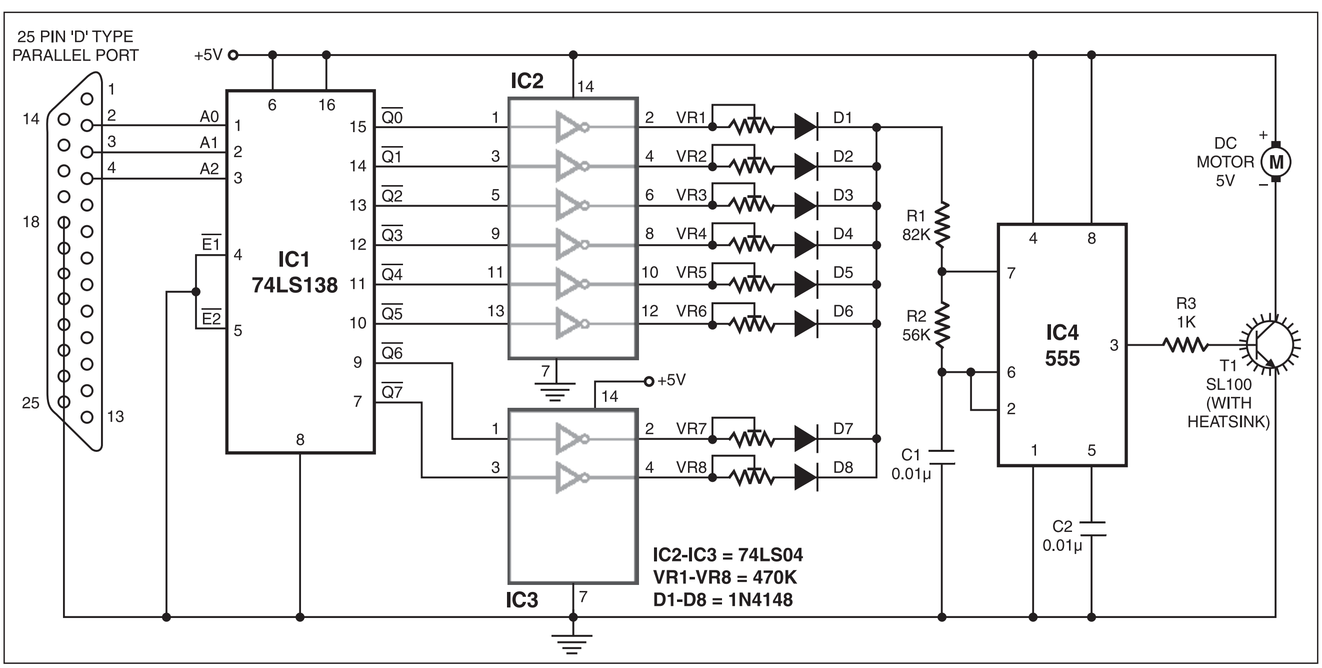
R. KARTHICK

This circuit allows you to control the speed of a DC motor (in eight lev-els) from your PC's parallel port. The PC uses a software program to control the speed of the motor.

The motor is connected to the PC through an interface circuit. The interface circuit consists of 1-of-8 decoder IC 74LS138 (IC1), hex inverter ICs 74LS04 (IC2 and IC3), resistor networks, timer IC 555 (IC4) and motor driver transistor SL100 (T1). The decoder IC accepts binary weighted inputs A0, A1 and A2 at pins 1, 2 and 3, respectively. With active-low enable input pins 4 and 5 of the decoder grounded, it provides eight mutually exclusive active-low outputs (Q0 through Q7). These outputs are inverted by hex inverters IC2 and IC3.

The resistor network comprising presets VR1 through VR8, resistors R1 and R2 and capacitor C1 are the timing components of timer IC 555 (IC4), which is configured in astable mode. The output of IC4 is a square wave, which is fed to the base of transistor T1 via current-limiting resistor R3. Transistor T1 is used to drive the motor.

The pulse-width modulation (PWM) method is used for efficient control of the



motor. The output of the PC is decoded to select a particular preset (VR1 through VR8). The value of the selected preset, along with resistors R1 and R2 and capacitor C1, changes the output pulse width at pin 3 of IC4. Thus the motor speed can be increased/decreased by choosing a particular resistance. For high-power motors, the transistor can be replaced by an IGBT or a power MOSFET.

The software (speedM.c) is written in 'C' language and compiled using Turbo C compiler.

Initially, when the motor is 'off,' the program prompts you to press 'Enter' key to start the motor. Once you press the key, the motor starts running at full speed. After a few seconds, the program asks you to press any key from the keyboard to go to the next screen for controlling the speed of the motor. This screen has options for increasing and decreasing the motor speed and also for exiting from the program. For increasing the speed enter choice 1 and press 'Enter' key, and for decreasing the speed enter choice 2 and press 'Enter' key. This action changes the speed by one step at-a-time and the message "Speed decreased" or "Speed increased" is displayed on the screen. To go to the main menu, again press 'Enter' key.

Note: The source code of the article is included in the CD.

SPEEDM.C

```
//R.KARTHICK,III ECE,K.L.N.C.E.,MADURAI
//karthick_klnce@rediffmail.com
#include<stdio.h>
#include<conio.h>
int a[7],i,c;
void start(void);
void main(void)
 {
 int P=0x0378,j,c=7,c1,x,y;
 clrscr();
 outportb(P,0);
 textbackground(9);
 textcolor(3);
 for(x=0;x<=80;x++)
   for(j=0;j<=25;j++)
    {
    gotoxy(x,j);
    cprintf(" ");
    }
 for(i=0;i<8;i++)
   a[i]=i;
 gotoxy(23,11);
 printf("Press Enter to start the motor");
 getch();
 gotoxy(28,13);
 printf("WAIT STARTING MOTOR");
 start();
 gotoxy(25,15);
 printf("Motor started sucessfully");
 gotoxy(22,17);
 printf("Press any key for speed control");
 getch();
 while(1)
  {
  clrscr();
  gotoxy(25,3);
    for(j=0;j<79;j++)
   {
   gotoxy(j+1,2);
   printf("*");
   }
  gotoxy(23,3);
    printf("DC MOTOR SPEED CONTROL USING
PC");
   for(j=0;j<79;j++)
    {
    gotoxy(j+1,4);
    printf("*");
    }
    printf("\n");
       printf("\t\t\t1.INCREASE SPEED\n\t\t\
t2.DECREASE SPEED\n\t\t\t3.EXIT") ;
    for(j=0;j<79;j++)
    {
    gotoxy(j+1,8);
    printf("*");
    }
     for(j=0;j<79;j++)
    {
    gotoxy(j+1,10);
    printf("*");
    }
    gotoxy(1,9);
    printf("Enter your choice:");
    scanf("%d",&c1);
    switch(c1)
     {
     case 1:if(c==7)
                      {
                      clrscr();
                      gotoxy(23,13);
                         printf("MOTOR IS RUNNING IN
FULL SPEED");
                                getch();
                                }
                    if(c<7)
                      {
                      clrscr();
                      c++;
                      outport(P,a[c]);
                      gotoxy(33,13);
                      printf("SPEED INCREASED");
                      getch();
                      }
                      break;
     case 2: if(c==0)
                      {
                      clrscr();
                      gotoxy(23,13);
         printf("MOTOR IS RUNNING IN LOW
SPEED");
                      getch();
                             }
                    if(c>0)
                     {
                     clrscr();
                     c--;
                     outport(P,a[c]);
                     gotoxy(33,13);
                     printf("SPEED DECREASED");
                     getch();
                     }
                     break;
     case 3  :
                    for(j=c;j>=0;j--)
                     {
                    outportb(0X0378,j);
                    delay(100);
                     }
                    outportb(P,0);
                    clrscr();
                    gotoxy(17,13);
                    textcolor(2);
                         cprintf("KARTHICK.R\nECE\
nK.L.N.COLLEGE OF ENGG\nMADURAI.");
                    getch();
                    exit(1);
     }
   }
  }

  void start()
   {
   outportb(0x0378,0);
   for(i=0;i<8;i++)
    {
    outportb(0X0378,i);
    delay(1000);
    }
```

FREQUENCY DIVIDER USING 7490 DECADE COUNTER

SRINIVAS MARYALA

Here is a low-cost circuit for generating different square-wave signals. The circuit is built around a 10MHz crystal oscillator, hex inverter IC 7404 and seven decade counter ICs 7490.

IC 7490 is a 4-bit, ripple-type decade counter. It consists of four master/slave flip-flops, which are internally connected to form a divide-by-two section and a divide-by-five section. Each section has a separate clock input to change the

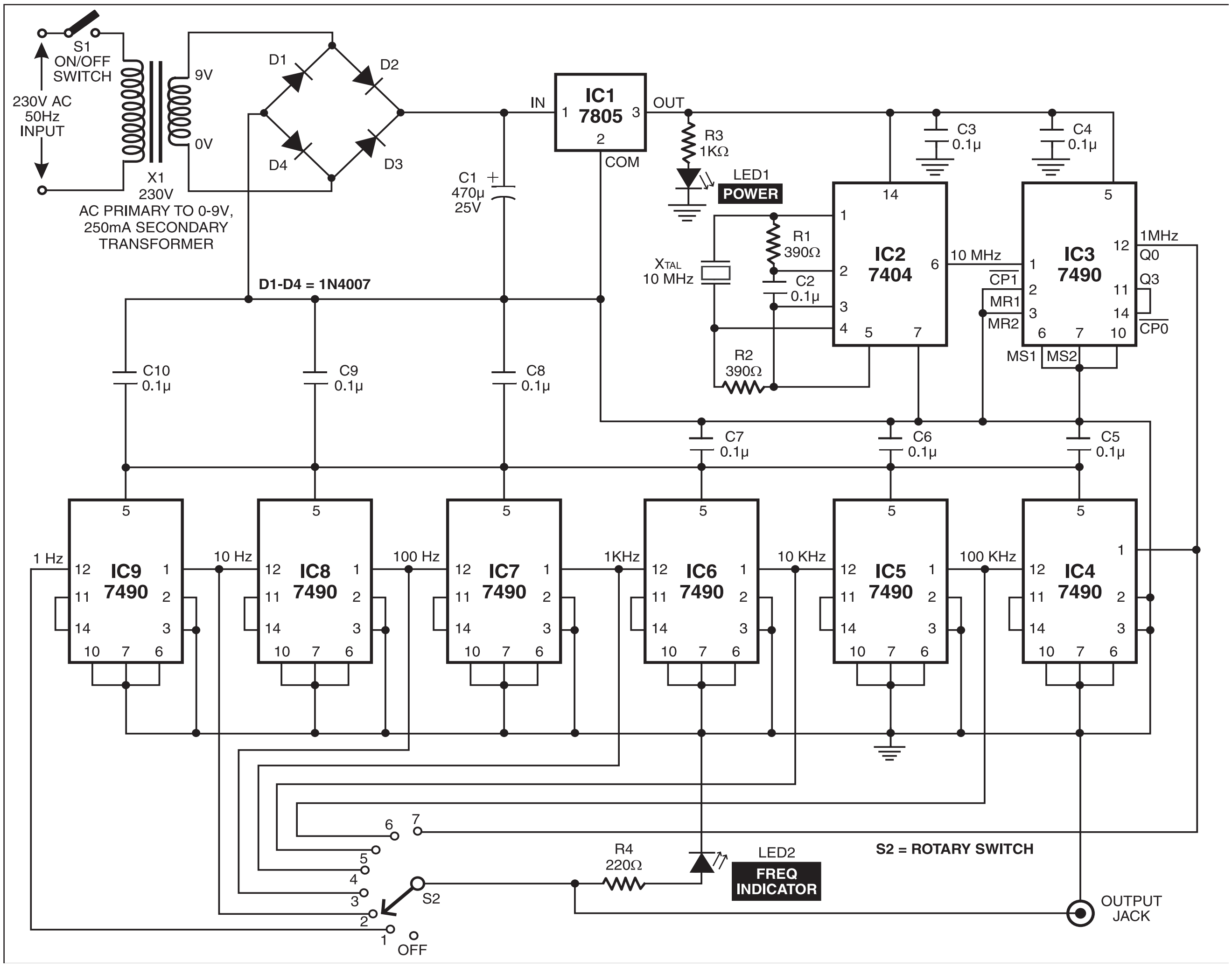


Fig. 1: Circuit of frequency divider using 7490 decade counter

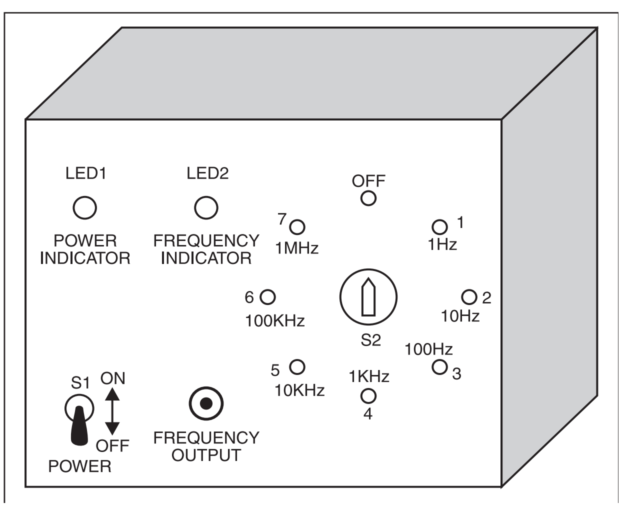


Fig. 2: Proposed control panel for the frequency divider using 7490 decade counter

output states of the counter on a high-to-low clock transition. The output states do not change simultaneously due to the internal ripple delays. Therefore the decoded output signals are subject to decoding spikes and should not be used for clocks or strobes.

Since the output of the divide-by-two section is not internally connected to the succeeding stages, IC 7490 can be operated in various counting modes.

In this circuit, ICs 7490 are configured as divide-by-10 counters. The power supply to the circuit is regulated by IC 7805 (IC1). LED1 indicates power on/off to the circuit. A 10MHz clock pulse is generated by the crystal and the associated circuit consisting of IC2 (7404). This clock pulse is fed to pin 1 of IC3 (IC 7490), which divides it by 10 to give a 1MHz clock pulse at its output pin 12. The 1MHz clock pulse is fed to the input of the next stage and so on up to IC9.

Thus at all the seven counter stages, we get unique output pulses (1 MHz, 100 kHz, 10 kHz, 1 kHz, 100Hz, 10Hz and 1 Hz, respectively). These output pulses are selected by rotary switch S2 and fed to an output jack. The blinking/flash rate of LED2 indicates the output frequency. However, you can identify output frequencies of 1 Hz and 10 Hz only. Above 10 Hz, the LED blinks so fast that it's not possible to estimate the frequency.

Assemble the circuit on a general-purpose PCB and enclose it in a cabinet as shown in Fig. 2.

DOME LAMP DIMMER

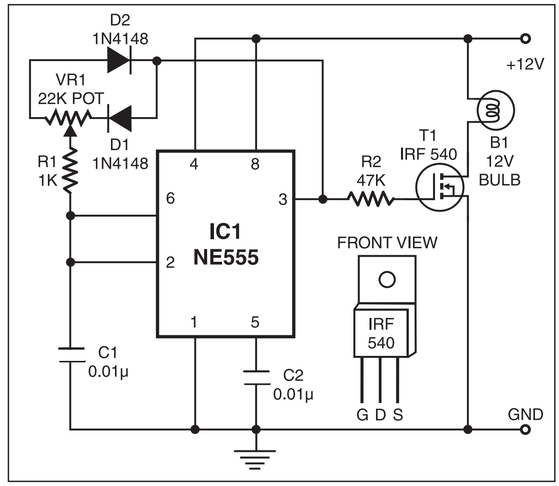
T.A. BABU

A reading light inside the car greatly assists passengers during night, but often the interior dome lamp is too bright and distracting to the driver. A linear regulator such as a rheostat can be used to control the brightness of the dome lamp but it consumes a lot of power.

Here is a dome lamp dimmer that gives you a fairly linear control over the lamp brightness from low to high intensity while consuming little power. Since it is a pulse-width modulated chopper circuit, you can also use it to dim a halogen bulb or control the speed of a mini drill, etc.

In the circuit, timer NE555 (IC1) is wired as an astable multivibrator to produce square wave at its output pin 3. The output of timer IC1 charges/discharges capacitor C1 via diodes D1 and D2. Adjust potmeter VR1 to control the RC time constant during the charge-discharge cycle and get the timer output with the desired pulse width. Thus the brightness of lamp B1 can be varied from low to high by adjusting potmeter VR1.

Most cars run only one wire to power the lamp and use the car body for the return current path. Connect ground path of the circuit to the car body. Use a suitable heat-sink for the MOSFET to handle the load current.



OFFSET TUNING INDICATOR FOR CW

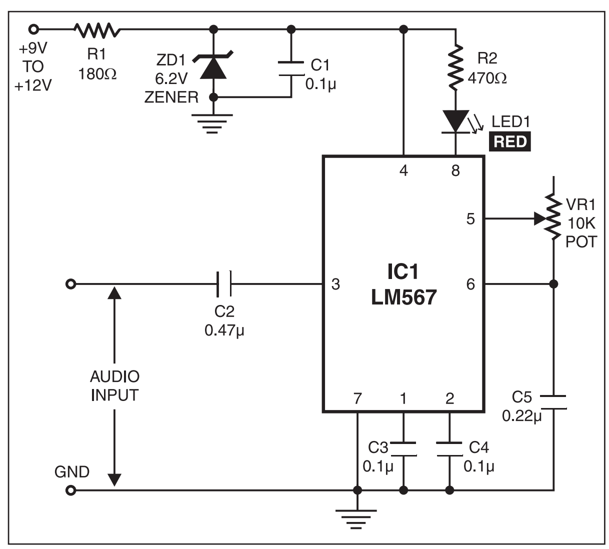
D. PRABAKARAN

Before the transceivers became popu-lar, the receiver and the transmitter were separate entities. Zero beating your continuous-wave (CW) transmitter to the called station was a simple matter. When transceivers appeared, difficulties arose. So modifications followed: first, a fixed CW offset of 600-800 Hz was provided. When a signal is tuned in, the local transmitter frequency is near that of the station being received. Next, to make the setting more accurate, a sidetone monitor producing a tone exactly equal to the count of offset appeared in most transceivers. By matching the incoming signal to the sidetone, transmitter frequency would be equal to the received signal frequency. However, matching the two tones is difficult and time-consuming.

Here's an easy-to-build offset tuning indicator for CW that provides a visual indication when the signal and sidetone match. It is built around tone decoder IC LM567, which is an 8-pin PLL IC. A lock between the IC's internal voltage-controlled oscillator (VCO) and the applied signal makes its output pin low. The VCO is set to an offset frequency of 600-800 Hz.

The received audio tone is monitored. When it matches with the frequency of the VCO, a red LED (LED1) turns on. Component values are optimised for the 600-800 Hz range.

For powering the circuit, a 9V or 12V DC source can be used. The operating voltage is regulated to 6.2 volts by zener diode (ZD1).



The audio input to the circuit is taken from the speaker or headphone output of the transceiver. One-time adjustment of VCO tuning control VR1 is required. The VCO must be accurately set to the transceiver offset. With the sidetone activated, adjust VR1 for the maximum LED indication at the lowest level that provides response.

8-DIGIT CODE LOCK FOR APPLIANCE SWITCHING

MANEESH CHADHA

This code lock is useful for appliances requiring exclusive or authorised use by those who know the preset code. If desired, the code can be changed.

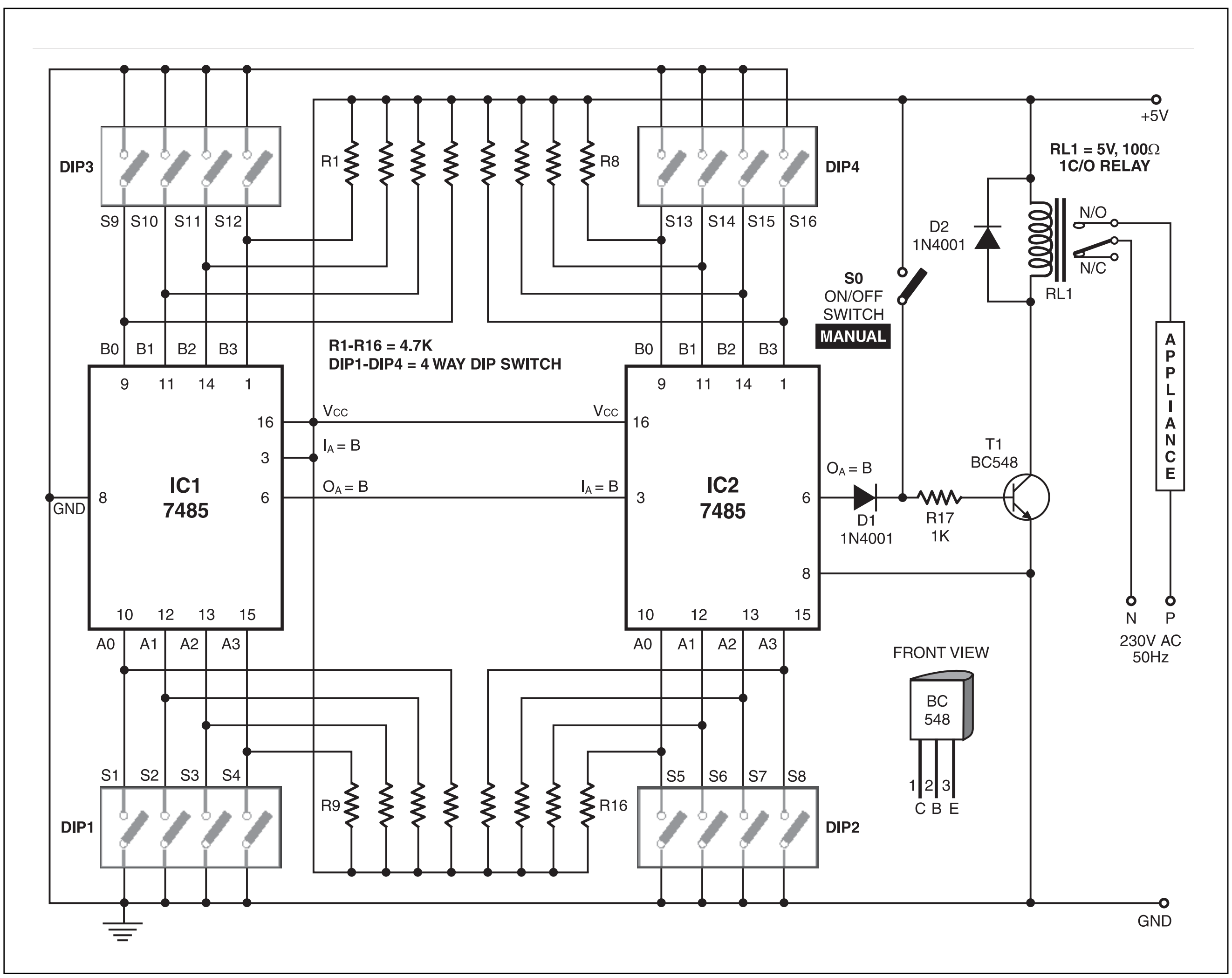
The circuit doesn't require additional AND or NOT gate operations at the outputs. It uses two pairs of 4-way DIP switches. The code is set using DIP switches DIP3 and DIP4. Then these two switches are hidden inside the assembly. With DIP3 and DIP4, up to 256 code combinations are possible. The unlocking code is set by the user using DIP switches DIP1 and DIP2, which is compared with the preset code entered earlier via DIP3 and DIP4. If the two codes match, transistor T1 conducts.

The codes are compared using two cascaded 4-bit magnitude comparator ICs (IC1 and IC2). If the input nibble present at DIP1 matches with preset DIP3 nibble, output pin 6 of IC1 (connected to input pin 3 of IC2) goes high. Now if nibble present at DIP2 matches with the preset nibble at DIP4, pin 6 of IC2 also goes high. This high output drives transistor T1 and the appliance turns on via relay contacts.

After use, disturb the positions of DIP1 and DIP2 so that the appliance can't be operated by unauthorised persons. This will also switch the appliance off.

The circuit works off a 5V DC power supply. Hidden switch S0 can be used to manually turn on/off the appliance if you have forgotten the preset code.

Caution. You may use this code lock at your own risk. After all, a clever intruder will try all 256 possible combinations one after the other to break the secret code.



STABILISED POWER SUPPLY WITH SHORT-CIRCUIT INDICATION

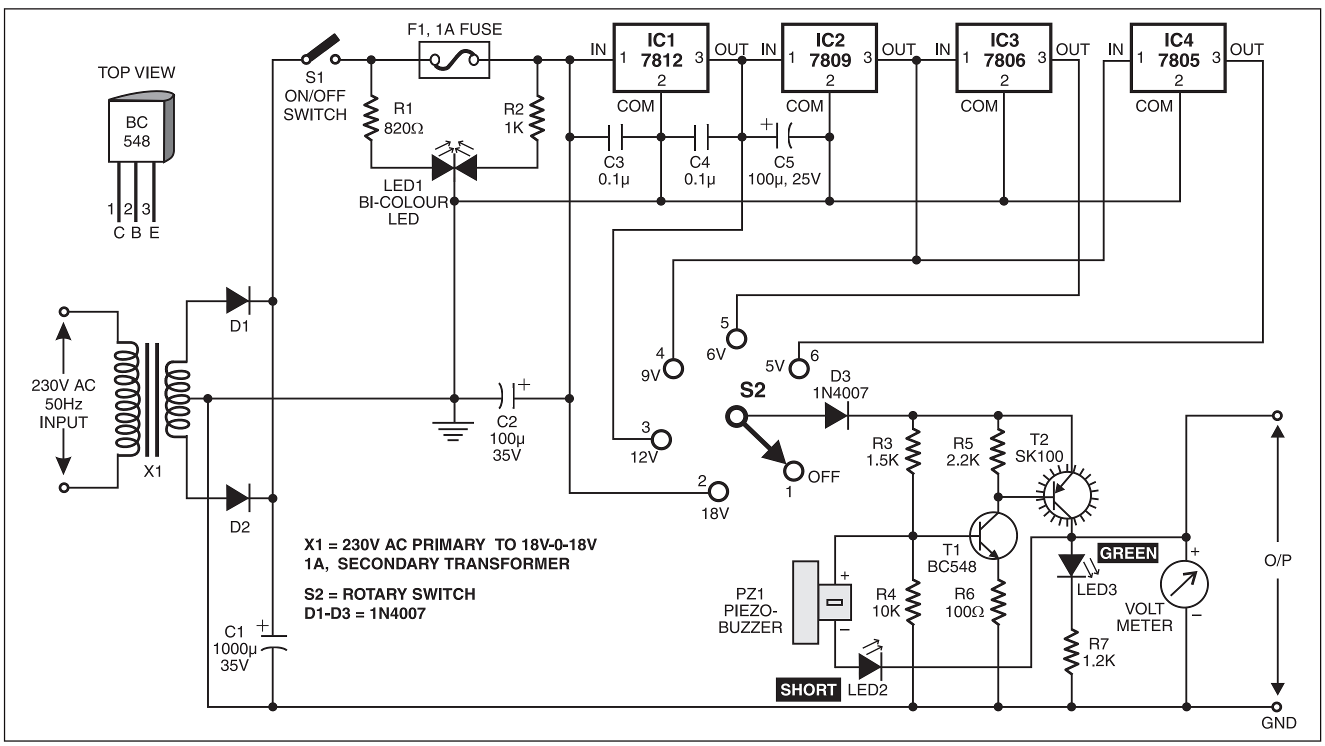
D. MOHAN KUMAR

Here is an efficient 4-stage stabi lised power supply unit for test ing elec-tronic circuits. It provides well regulated and stabilised output, which is essential for most electronic circuits to give proper results. The circuit provides an audio-visual indication if there is a short circuit in the PCB under test, so the power supply to the circuit 'under test' can be cut-off immediately to save the valuable components from damage.

The circuit provides four different regulated outputs (12V, 9V, 6V and 5V) and an unregulated 18V output, which are selectable through rotary switch S2. The selected output is indicated on the analogue voltmeter connected to the outputs rails.

The circuit uses a standard 18V-0-18V, 500mA step-down transformer to generate 18V AC. A rectifier diode comprising diodes D1 and D2 provides 18V DC, which is smoothed by capacitor C1 and given to the combination of regulator ICs (IC1 through IC4). The regulator ICs produce fixed, regulated outputs of 12V, 9V, 6V and 5V, respectively, which are connected to the rotary switch contacts. This power supply is useful for loads requiring up to 200mA current.

Complementary transistors T1 and T2 conduct when the power to the circuit is switched on. Full selected supply voltage is available at the collector of transistor T2, which is used to power the load. LED3 indicates the presence of output voltage. The negative terminal of piezobuzzer PZ1 is connected to the output rail via LED2, so the piezobuzzer remains silent as its negative terminal is also at full supply voltage (selected). If there is a short circuit at the output, LED2 glows to activate the piezobuzzer.



A fuse-failure indicator distinguishes short circuit at the output and input failure. It consists of a bicolour LED (LED1) and resistors R1 and R2. When power is available and the fuse is intact, red and green halves of LED1 are effectively in parallel to output a yellowish light. When fuse fails, green LED goes off and red LED lights up to indicate fuse breakdown.

The circuit can be easily constructed on a general-purpose PCB. Use small heat-sinks for all ICs to dissipate heat. The output voltage can be read on a voltmeter. Enclose the circuit in a metal box with provisions for voltmeter, LEDs, rotary switch, etc.

LIGHT-OPERATED INTERNAL DOOR LATCH

V. GOPALAKRISHNAN

Using this light-operated circuit, you can close or open the door of your room remotely from your bed. You just have to focus the torchlight on the light-dependent resistor of the circuit, which you can install inside your room at a suitable position.

The circuit comprises a control unit and a driver unit for the stepper motor circuit used to latch/open the door.

The control unit comprises two timer 555 ICs (IC1 and IC2), NAND gate IC (IC3), 4-bit bidirectional universal shift register (IC4), OR gate (IC5), NOR gate (IC6), hex inverter (IC7) and dual D-type positive-edge triggered flip-flop (IC8) as shown in Fig. 1. The driver circuit shown in Fig. 2 uses four Darlington pair transistors (T1 through T8) to increase the current carrying capability for operating the stepper motor.

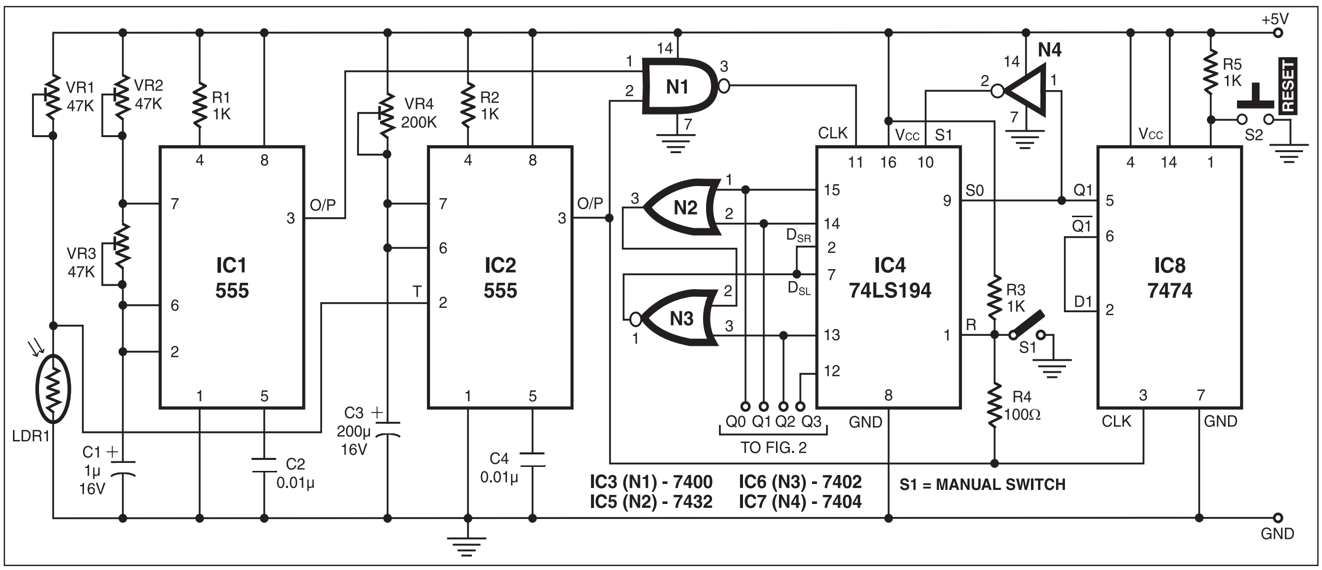


Fig. 1: Circuit of the control unit

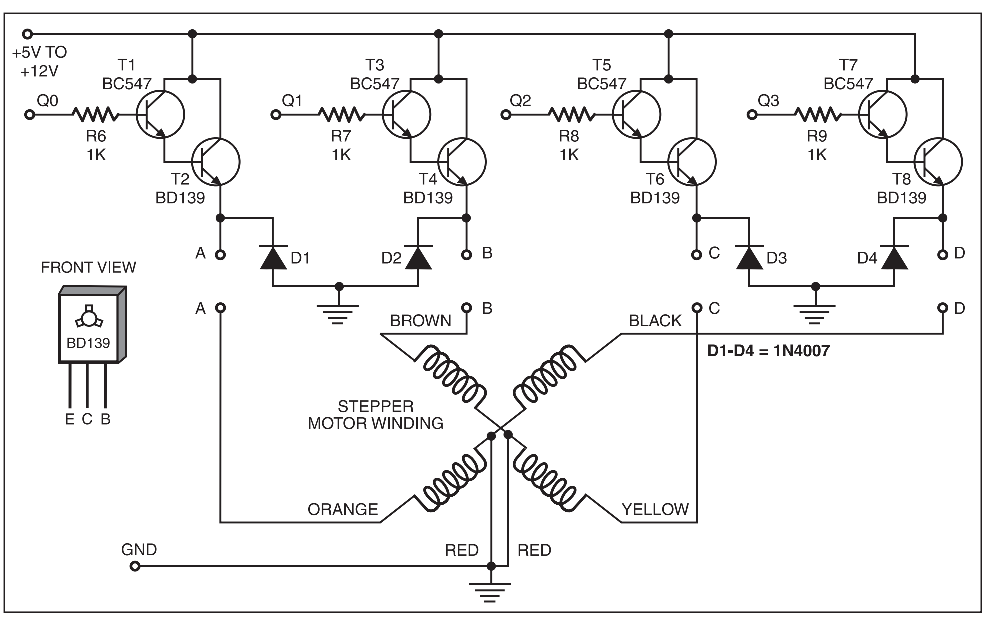


Fig. 2: Driver circuit for the stepper motor

The astable multivibrator built around timer 555 (IC1) has a time period of 1.5 seconds. The monostable built around IC2 is triggered when torchlight is focused on light-dependent resistor LDR1. Sensitivity potentiometer VR1 is adjusted to ambient light. Normally, the LDR is kept covered to avoid its activation by ambient light.

When torchlight is focused on the LDR, the monostable (IC2) is triggered. The 'on' time of IC2 is adjusted to 15 seconds by potentiometer VR4. The outputs at pin 3 of astable IC1 and monostable IC2 are fed to NAND gate N1 of IC3. The Q0 and Q1 outputs of shift register IC4 are ORed by OR gate N2 and its output

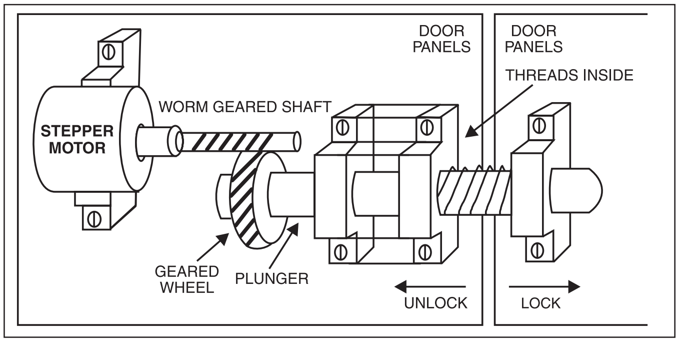


Fig. 3: Mechanical arrangement for the lock

Shift Register Output Waveform

Clock	Q3	Q2	Q1	Q0
1	0	0	0	1
2	0	0	1	0
3	0	1	0	0
4	1	0	0	0

is fed to NOR gate N3. The Q2 output of IC4 forms the second input for NOR gate N3. The output of NOR gate N3 goes to shift-right and shift-left serial data inputs (pins 2 and 7) of IC4.

Mode-control inputs S0 and S1 are used for direction changing of shift register IC4. The Q1 output of dual D-type flip-flop IC8 is fed to S0 directly and to S1 after inversion by N4.

The output of monostable IC2 resets IC4 via resistor R4, which stops the stepper motor. You can also manually stop the stepper motor by pushing switch S1 to 'on' position.

Switch S2 is used for resetting dual D-type flip-flop IC8. The monostable output also provides clock to the D flip-flop operating in toggle mode by connecting Q1 to D1 of IC8.

Fig. 2 shows the driver unit for the stepper motor, along with windings details of the stepper motor. Connect Q0 through Q3 outputs of IC4 in the control unit (Fig. 1) to positive and ground power supply terminals of the driver unit (Fig. 2). The waveform drive pattern of shift outputs of IC4 is shown in the table.

When you direct torchlight on the LDR, the stepper motor runs in one direction to latch the door. If you again focus torchlight on the LDR, the stepper motor runs in reverse direction to open the latch.

Fig. 3 shows the locking arrangement operated by the stepper motor.

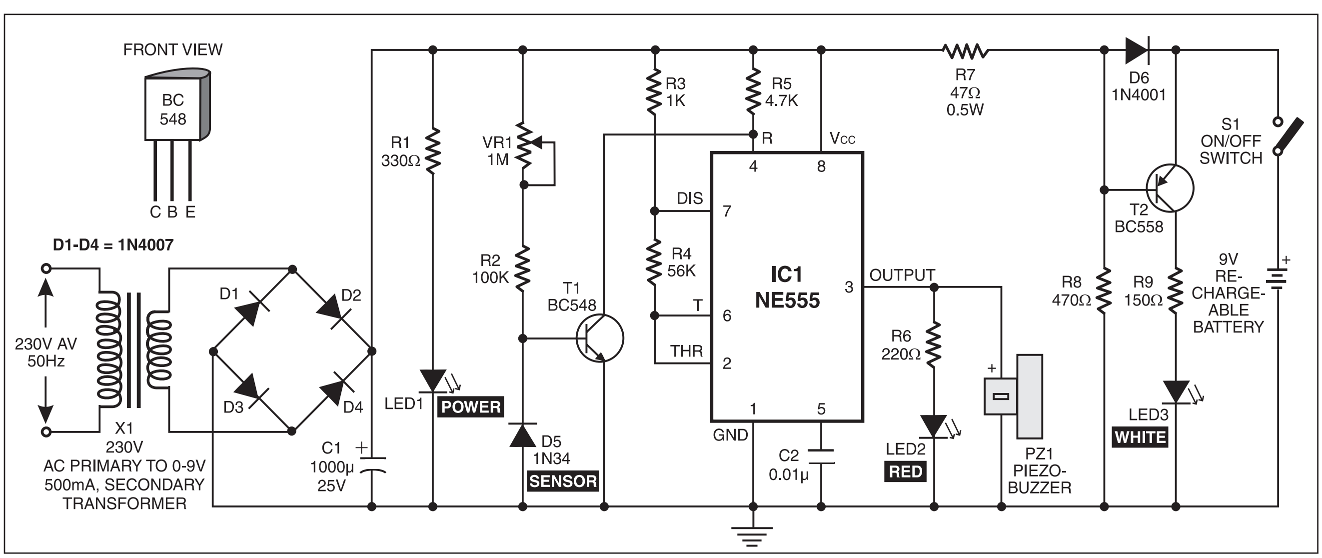
Note. During testing at EFY lab, a stepper motor for read/write head positioning in a 1.2MB floppy disk drive unit, operating off 12V with 3.6-degree revolution per step, was used. Connect the coloured terminal wires of the motor to the driver unit as shown in Fig. 2.

MAINS BOX HEAT MONITOR

D. MOHAN KUMAR

This simple circuit monitors the mains distribution box constantly and sounds an alarm when it senses a high temperature due to overheating, helping to prevent disasters caused by any sparking in the mains box due to short circuits. It also automatically switches on a bright white LED when the power fails. The LED gives ample light to check the mains box wiring or fuses in darkness. The circuit beeps once when power fails and again when power resumes.

The AC mains is stepped down by transformer X1 to deliver a secondary output of 9V AC at 500 mA. The transformer output is rectified by diodes D1 through D4. Capacitor C1 bypasses the ripple. LED1 indicates the power-on condition.



Resistor R1 acts as the current limiter for LED1.

Germanium diode D5 (1N34) is the temperature-sensing element, which is connected in the reverse bias mode. At normal temperature, the resistance of the diode is high and, as a result, transistor T1 conducts to hold reset pin 4 of IC1 in low state.

NE555 (IC1) is wired as an astable multivibrator. When the temperature around diode D5 rises due to overheating of the fuse, the resistance of the diode decreases and transistor T1 stops conducting. This enables IC1 and the oscillator starts to sound an alarm. By adjusting preset VR1, you can set the temperature level at which the alarm circuit is activated.

The emergency light circuit uses pnp transistor BC558 (T2) and a few passive components. It is powered by a 9V rechargeable battery, which is constantly charged via forward-biased diode D6 when mains power is present. Resistor R7 reduces the charging current to a safer level. The forward biasing of diode D6 results in reverse biasing of transistor T2 and thus the white LED (LED3) is off. When the power fails, transistor T2 is forward biased and lights up the LED. When the power resumes, transistor T2 stops conducting and the LED doesn't glow.

The circuit can be easily constructed on any general-purpose PCB. Diode D5 should be placed close to the fuse to sense the heat. It can be connected to the PCB using a short piece of shielded wire. The white LED should be directed towards the fuse such that the maximum light falls on the fuse.

To test the circuit, take the hot tip of the soldering iron near diode D5. The buzzer will sound to indicate the high temperature.

DIGITAL STOP WATCH

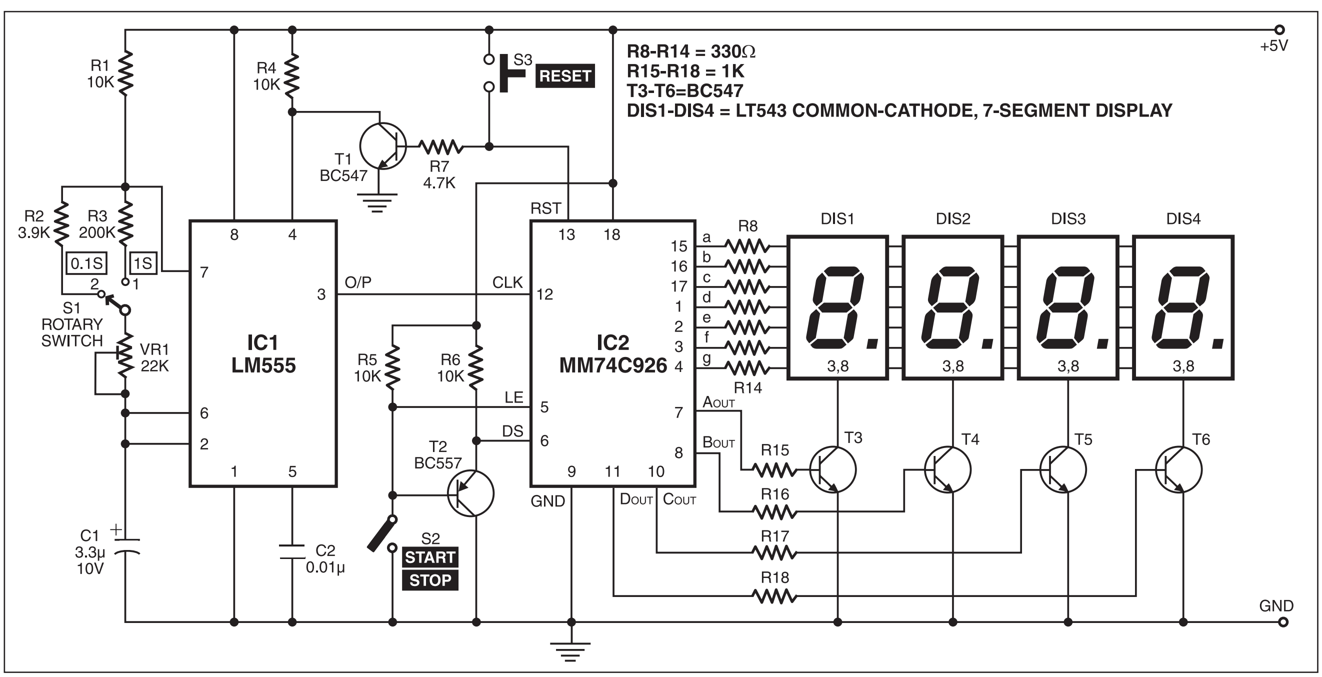
C.H. VITHALANI

Here's a digital stop watch built around timer IC LM555 and 4-digit counter IC with multiplexed 7-segment output drivers (MM74C926).

IC MM74C926 consists of a 4-digit counter, an internal output latch, npn output sourcing drivers for common-cathode, 7-segment display and an internal multiplexing circuitry with four multiplexing outputs. The multiplexing circuit has its own free running oscillator, and requires no external clock. The counter advances on negative edge of the clock. The clock is generated by timer IC LM555 (IC1) and applied to pin 12 of IC2.

A high signal on reset pin 13 of IC2 resets the counter to zero. Reset pin 13 is connected to +5V through reset push-on-switch S3. When S2 is momentarily pressed, the count value becomes 0, transistor T1 conducts and it resets IC1. Counting starts when S2 is in 'off' condition.

A low signal on the latch-enable input pin 5 (LE) of IC2 latches the number in the counter into the internal output latches. When switch S2 is pressed, pin 5 goes low and hence the count value gets stored in the latch. Display-select pin 6 (DS) decides whether the number on the counter or the number stored in the latch is to be displayed. If pin 6 is low the number in the output latch is displayed, and if pin 6 is high the number in the counter is displayed.



When switch S2 is pressed, the base of pnp transistor T2 is connected to ground and it starts conducting. The emitter of T2 is connected to DS pin of IC2. Thus, when switch S3 is pressed, reset pin 13 of IC2 is connected to ground via transistor T1 and the oscillator does not generate clock pulses. This is done to achieve synchronisation between IC1 and IC2.

First, reset the circuit so that the display shows '0000.' Now open switch S2 for the stop watch to start counting the time. If you want to stop the clock, close switch S2.

Rotary switch S1 is used to select the different time periods at the output of the astable multivibrator (IC1). The circuit works off a 5V power supply. It can be easily assembled on a general-purpose PCB. Enclose the circuit in a metal box with provisions for four 7-segment displays, rotary switch S1, start/stop switch S2 and reset switch S3 in the front panel of the box.

FLASHING-CUM-RUNNING LIGHT

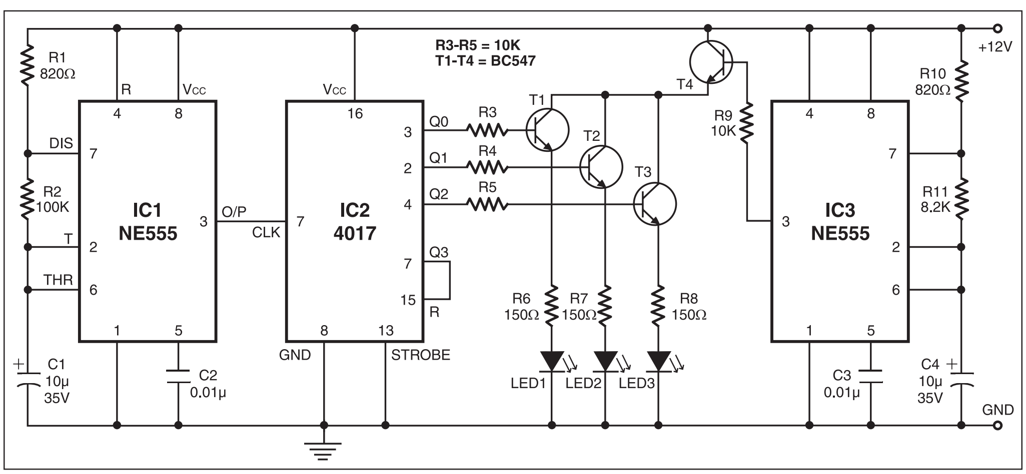
A. SIVASUBRAMANIAN

This circuit generates flashing lights in running pattern. In conventional running lights, the LEDs glow one by one. In this circuit, the LEDs flash a number of times one by one.

The circuit comprises two astable multivibrators (IC1 and IC3) and a decade counter (IC2). Astable multivibrator IC1 produces approximately 0.72Hz clock frequencies, which are given to decade counter IC2. The decade counter is designed to count Q0, Q1 and Q2 outputs, while its fourth output (Q3) is used to reset it. The Q0, Q1 and Q2 outputs of IC2 are fed to npn transistors T1, T2 and T3, respectively. The collectors of transistors T1, T2 and T3 are connected to the emitter of transistor T4, while their emitters are connected to LED1, LED2 and LED3 via 150-ohm resistors R6, R7 and R8, respectively. The LEDs are activated one by one by the decade counter outputs.

Astable multivibrator IC3 produces approximately 8.4Hz clock, which is given to transistor T4 via resistor R9 to switch on the supply to transistors T1 through T3 for each positive half cycle of IC3 output.

Now for each output period of IC2, a particular LED blinks at the rate of 8.4 Hz. The blinking then shifts to the next LED when the output of IC2 advances by one count (after about 1.3 seconds). Similarly, the blinking effect shifts to the next LED after another 1.3 seconds and the cycle repeats thereafter.



Flashing frequencies can be changed by changing the values of R10 and R11 and capacitor C4. The circuit can be easily assembled on any general-purpose PCB. It works off a 12V regulated power supply. You can also add more LEDs in series with LED1, LED2 and LED3, respectively.

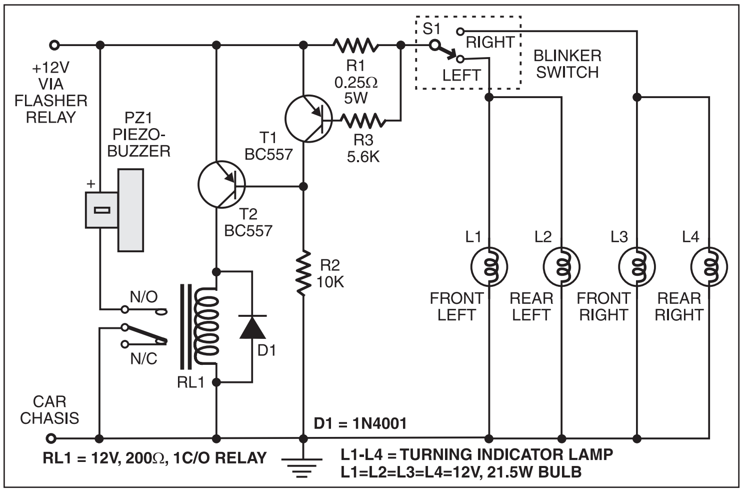
FAULTY CAR INDICATOR ALARM

DEBARAJ KEOT

Before taking a turn, either left or right, car drivers need to switch on the car's turn-indicator lamps so that the approaching vehicle drivers can take precaution accordingly. An accident is likely to occur in case your car's turn-indicator lamps fail to glow due to some reason or the other. Here's a circuit that sounds an alarm if your turn-indicator lamps don't glow, helping you to safeguard against any accident.

When both the the front and rear

turn-indicator lamps (either right or left) glow, the current through the lamps (L1-L2 or L3-L4) causes a voltage drop across series resistor R1. This voltage drives pnp transistor T1 into saturation. In this condition, pnp transistor T2 does not conduct and hence relay RL1 does not energise. No sound from piezobuzzer PZ1 (connected to normally-opened (N/O) contacts of relay RL1) means that the turn-indicator lamps are working satisfactorily.



When one or both of the turn-indicator bulbs are fused, the voltage drop across R1 is insufficient and pnp transistor T1 remains cut-off. In this condition, pnp transistor T2 conducts to energise relay RL1 and piezobuzzer PZ1 sounds to indicate that one or both the turn-indicator bulbs are fused.

Install the circuit (excluding turn-indicator lamps L1 through L4, which are already fitted in your car) near the driver's seat so that the driver has easy access to blinker switch S1. To turn left, the driver needs to connect blinker switch S1 to left position to flash front and back left-turn-indicator lamps (L1 and L2). Similarly, to turn right, he needs to connect blinker switch S1 to right position to flash front and back right-turn indicator lamps (L3 and L4).

The value of resistor R1 is to be changed according to the bulb wattages.

QUALITY FM TRANSMITTER

TAPAN KUMAR MAHARANA

This FM transmitter for your stereo or any other amplifier provides a good signal strength up to a distance of 500 metres with a power output of about 200 mW. It works off a 9V battery.

The audio-frequency modulation stage is built around transistor BF494 (T1), which is wired as a VHF oscillator and modulates the audio signal present at the base. Using preset VR1, you can adjust the audio signal level. The VHF frequency is decided by coil L1 and variable capacitor VC1. Reduce the value of VR2 to have a greater power output.

The next stage is built around transistor BC548 (T2), which serves as a Class-A power amplifier. This stage is inductively coupled to the audio-frequency modulation stage. The antenna matching network consists of variable capacitor VC2 and capacitor C9. Adjust VC2 for the maximum transmission of power or signal strength at the receiver.

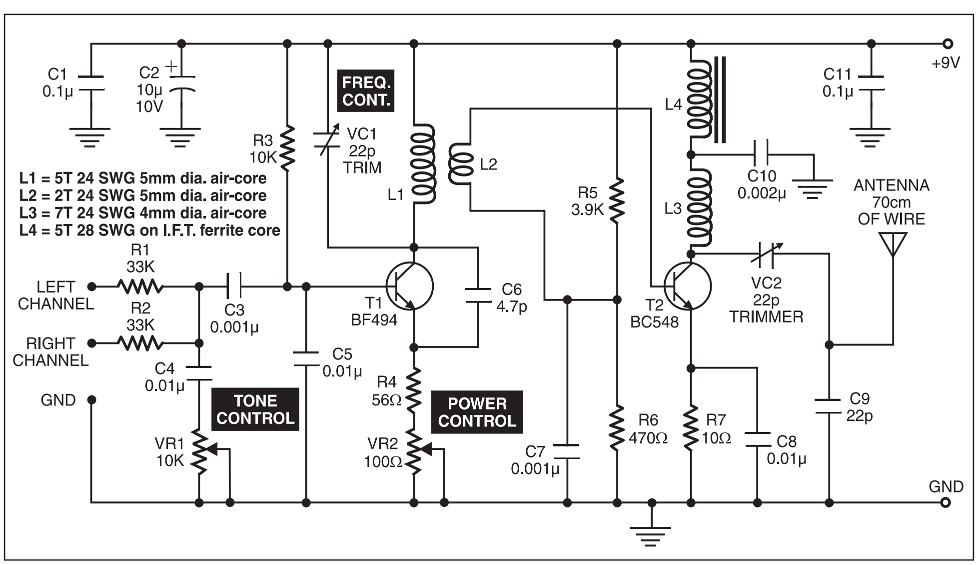
For frequency stability, use a regulated DC power supply and house the transmitter inside a metallic cabinet. For higher antenna gain, use a telescopic antenna in place of the simple wire. Coils L1 and L2 are to be wound over the same air core such that windings for coil L2 start from the end point for coil L1. Coil winding details are given below:

L1: 5 turns of 24 SWG wire closely wound over a 5mm dia. air core

L2: 2 turns of 24 SWG wire closely wound over the 5mm dia. air core

L3: 7 turns of 24 SWG wire closely wound over a 4mm dia. air core

L4: 5 turns of 28 SWG wire on an intermediate-frequency transmitter (IFT) ferrite core



SIMPLE KEY-OPERATED GATE LOCKING SYSTEM

DIPANJAN BHATTACHARJEE

This simple key-operated gate locking system allows only those persons who know the preset code to open the gate. The code is to be entered from the keypad within the preset time to operate the motor fitted in the gate. If anyone trying to open the gate presses a wrong key in the keypad, the system is disabled and, at the same time, sounds an alarm to alert you of an unauthorised entry.

Figs 1 and 2 show the block and circuit diagrams of the key-operated code locking system, respectively. Connect points A, B, C, D, E, F and ground of the circuit to the respective points of the keypad. Keys S7, S16, S14 and S3 are used here for code entry, and the remaining keys are used for disabling the system. It is very important to press the keys in that order to form the code. To start the motor of the gate, press switches S7, S16, S14 and S3 sequentially. If the keys are pressed in a different order from the preset order, the system will lock automatically and the motor will not start.

Initially, 6V is not available at pin 14 of AND gate IC6, so no pulse reaches the base of npn transistor T1 to trigger timer IC5 and, as a result, the gate doesn't open. To enable the system, first you have to trigger IC4. Pressing switch S7 triggers timer IC4 to provide 6V to IC6 for approximately 17 seconds. Within this time, you have to press switches S16, S14 and S3 sequentially. As a result, the outputs of timers IC1, IC2 and IC3 sequentially go high. These high outputs are further given to gates N1 and N2 of IC6 to trigger IC7 via npn transistor T1. The time durations for the high outputs of IC1, IC2 and IC3 are preset at 13.5, 9.43 and 2.42 seconds, respectively.

When all the four switches (S7, S16, S14 and S3) are pressed sequentially, timer IC7 triggers to start the motor for the preset period to open the gate. Once the time elapses, the motor stops automatically. The 'on' time for the motor can be selected by adjusting preset VR5. Here, the minimum 'on' time is

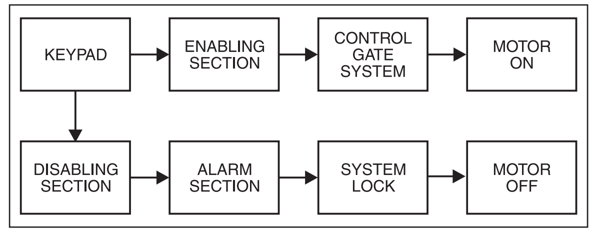


Fig. 1: Block diagram of simple key-operated gate locking system

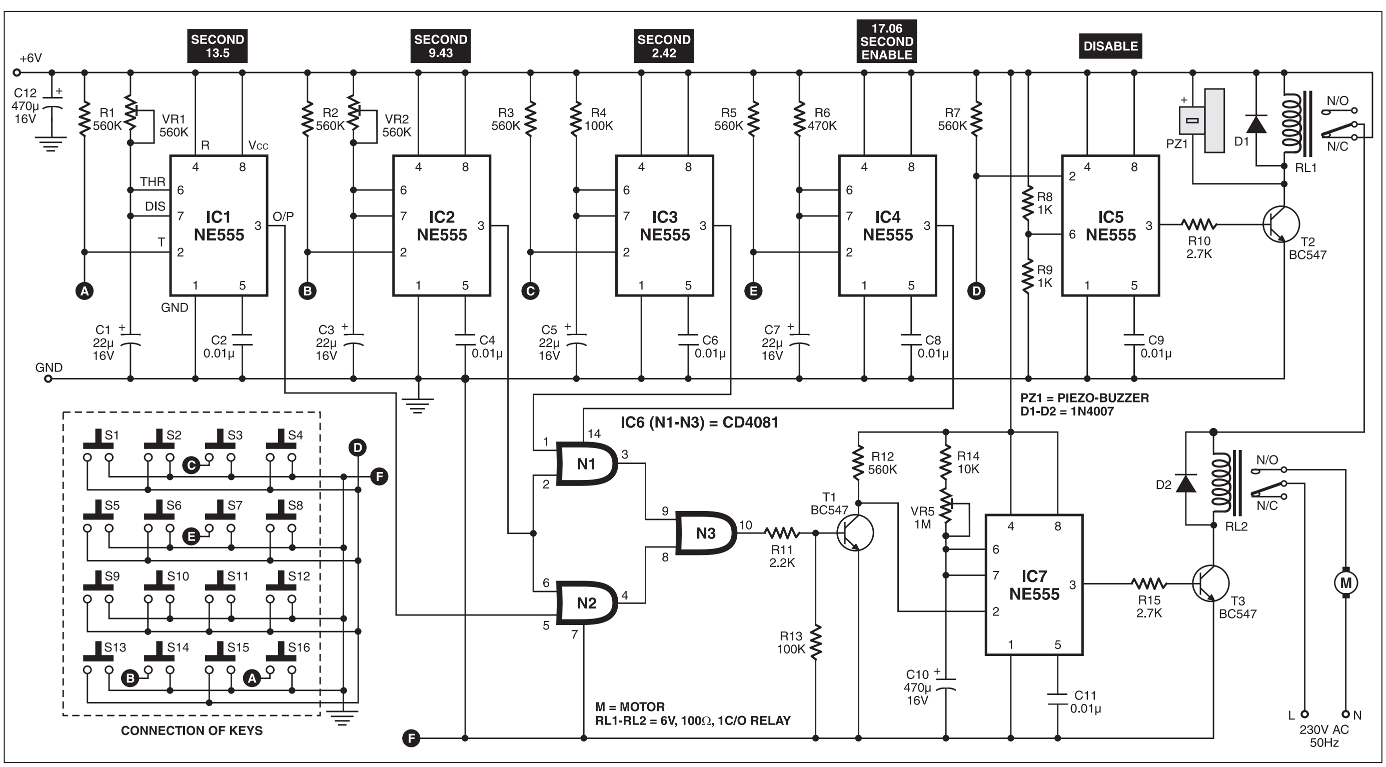


Fig. 2: Circuit of simple key-operated gate locking system

5.17 seconds and the maximum 'on' time is 517 seconds.

If a switch other than S7, S16, S14 and S3 is pressed, IC5 triggers to energise relay RL1, which disconnects the power supply of the second relay and the system gets locked and piezobuzzer PZ1 sounds an alarm to alert you that somebody is trying to open the gate lock.

Now to stop the sound and reset the system again press any key (other than S7, S16, S14 and S3) from the keypad.

The circuit works off 6V DC regulated power supply and can be easily assembled on a general-purpose PCB.

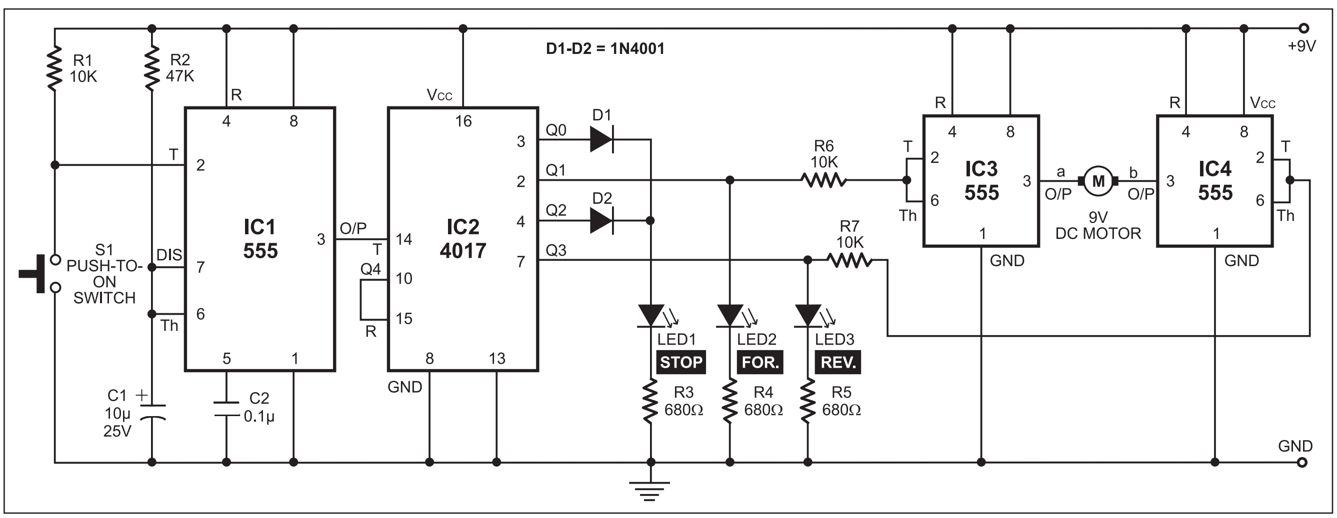
DC MOTOR CONTROL USING A SINGLE SWITCH

V. DAVID

This simple circuit lets you run a DC motor in clockwise or anti-clockwise direction and stop it using a single switch. It provides a constant voltage for proper operation of the motor. The glowing of LED1 through LED3 indicates that the motor is in stop, forward rotation and reverse conditions, respectively.

Here, timer IC1 is wired as a monostable multivibrator to avoid false triggering of the motor while pressing switch S1. Its time period is approximately 500 milliseconds (ms).

Suppose, initially, the circuit is in reset condition with Q0 output of IC2 being high. Since Q1 and Q3 outputs of IC2 are low, the outputs of IC3 and IC4 are



high and the motor doesn't rotate. LED1 glows to indicate that the motor is in stop condition.

When you momentarily press switch S1, timer 555 (IC1) provides a pulse to decade counter CD4017 (IC2), which advances its output by one and its high state shifts from Q0 to Q1. When Q1 goes high, the output of IC3 at pin 3 goes low, so the motor starts running in clockwise (forward) direction. LED2 glows to indicate that the motor is running in forward direction.

Now if you press S1 again, the high output of IC2 shifts from Q1 to Q2. The low Q1 output of IC2 makes pin 3 of IC3 high and the motor doesn't rotate. LED1 glows (via diode D2) to indicate that the motor is in stop condition.

Pressing switch S1 once again shifts the high output of IC2 from Q2 to Q3. The high Q3 output of IC2 makes pin 3 of IC4 low and the motor starts running in anti-clockwise (reverse) direction. LED3 glows to indicate that the motor is running in reverse direction.

If you press S1 again, the high output of IC2 shifts from Q3 to Q4. Since Q4 is connected to reset pin 15, it resets decade counter CD4017 and its Q0 output goes high, so the motor does not rotate. LED1 glows via diode D1 to indicate that the motor is in stop condition. Thereafter, the cycle repeats.

If you don't want to operate the motor in reverse direction, remove timer IC4 along with resistors R5 and R7 and LED3. And connect 'b' terminal of the motor to +Vcc.

Similarly, if you don't want to run the motor in forward direction, remove timer IC3 along with resistors R4 and R6 and LED2. And connect 'a' terminal of the motor to +Vcc.

The circuit works off a 9V regulated power supply for a 9V DC motor. Use a 6V regulated power supply for a 6V DC motor.

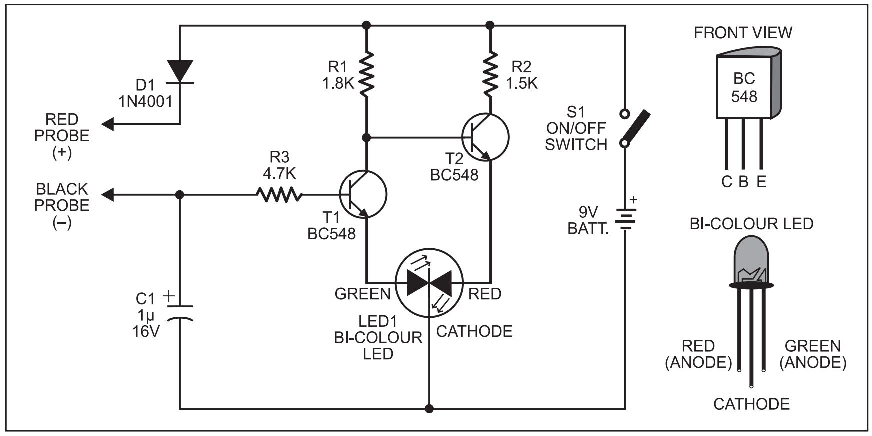
HANDY TESTER

D. MOHAN KUMAR

For beginners, here's a low-cost multitester that can be used to test the condition of almost all the electronic components from resistors to ICs. It uses only a few components but can also detect polarity, continuity, logic states and activity of multivibrators.

The circuit is extremely simple and exploits the biasing property of bipolar transistors. Transistors T1 and T2 act as transistor switches driving the red and green halves of bicolour LED1 independently to give results of the test.

When power is applied by pressing switch S1, transistor T1 stops conducting due to the lack of forward bias. At the same time, transistor T2 takes base bias voltage from the battery through resistor R1 and conducts. This allows the red half of bicolour LED1 to illuminate.



When the base of transistor T1 gets positive voltage through resistor R3, it conducts to light up the green half of bicolour LED1. When transistor T1 conducts, the base of transistor T2 is grounded and it cuts off to turn off the red half of bicolour LED1. The functioning of the circuit thus depends on the signal obtained at the base of transistor T1. The table gives the testing procedures for various components with the expected indications/results.

Bi-Colour LED Status for Various Tests

Component/test	Test procedure	LED1 status	Result	Note
Continuity	Red and black probes to the test points	Green 'on'	Continuity	
		Red 'on'	No continuity	
Polarity	Red probe to the positive of the circuit and black probe to the test point	Green 'on'	Positive	Circuit should be 'on'
		Red 'on'	Negative or no power	
Logic	Red probe to the circuit's positive and black probe to the output	Green 'on'	High	Circuit should be 'on'
		Red 'on'	Low	
IC	Red probe to the circuit's positive and black probe to the output	Green 'on'	High	Circuit should be 'on'
		Red 'on'	Low	
Multivibrator IC 555	Red probe to the circuit's positive and black probe to the output	Colour changes from red to yellow to green cyclically	IC oscillating	Circuit should be 'on'
		Red 'on'	No oscillation	
Electrolytic capacitor	Red probe to the positive and black probe to the negative lead	Green gradually turns red Red 'on'	Capacitor good Capacitor faulty	Capacitor should be discharged
Diode (LED/ Photodiode/IR diode)	Red probe to the anode and black probe to the cathode	Green 'on'	Good	1-kilo-ohm resistor should be connected to the anode of LEDs
	Red probe to the cathode and black probe to the anode	Red 'on'	Good	
	In both conditions	Colour remains the same (either green or red)	Open/short	
Resistor (1 ohm to 500 kilo-ohms)	Red and black probes to the ends of the resistor	Green 'on'	Good	
		Red 'on'	Faulty	
Transistor	Red probe to the base of the transistor and black probe first to the collector and then to the emitter	Green 'on' and again green 'on'	Transistor conducts	Circuit should be 'on'
	Black probe to the base of the transistor and red probe first to the collector and then to the emitter	Green 'on' and then red 'on'	Transistor doesn't conduct	

PROGRAMMABLE ELECTRONIC DICE

MANEESH CHADHA

Here's a simple programmable electronic dice with numeric display. This dice can be programmed using a 4-way DIP switch to display any random number between '1' and '2,' '1' and '3,' ….. or '1' and '9.'

To obtain the desired dice range, inner switches A, B, C and D of DIP switch are to be set as per the table. For example, if you want the electronic dice to count from 1 to 8, close switches A and D and keep B and C open. On pressing switch S1, the display varies fast between '1' and '8.' When you release S1, the display stops shuffling and the last (latest) number remains on it.

IC1 is a dual 4-input Schmitt trigger NAND gate 74LS13. Gate N1 is used as an oscillator built using resistor R2 and capacitor C1 to produce approximately 70kHz clock frequency, which is fed to IC2. Gate N2 loads data at the inputs of IC2.

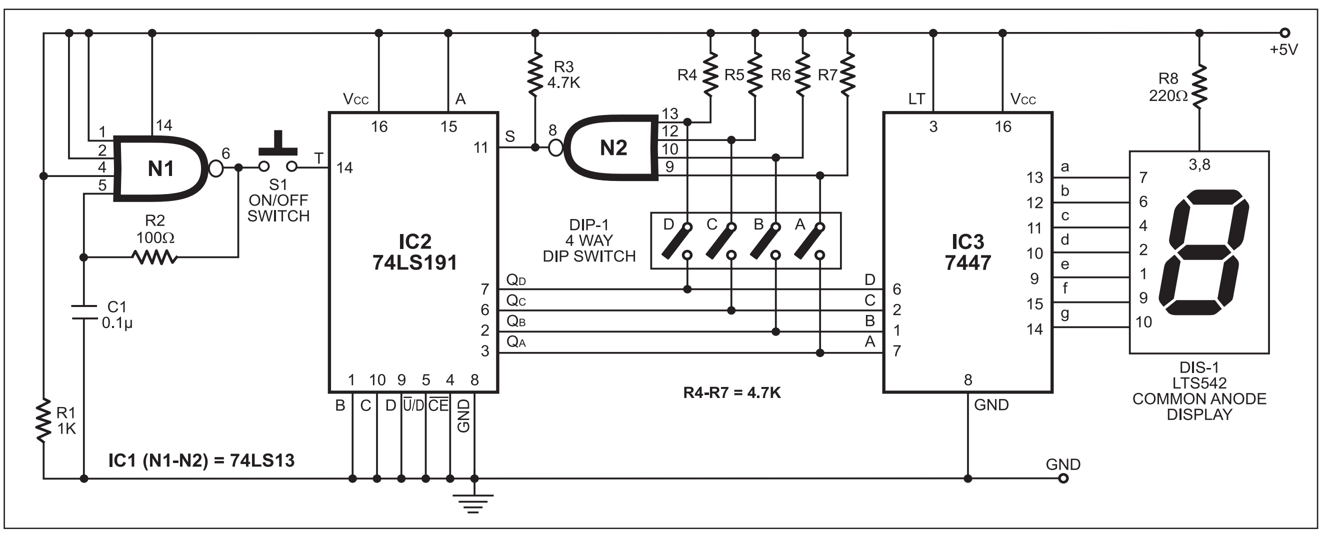
IC2 is a presettable binary counter (74LS191) with parallel loading facility. Whenever its pin 11 goes low, the data present at its inputs D through A (which is '0001') appears at its outputs Q_D through Q_A when all the inner switches of DIP switch are open and DIS1 shows the minimum count as '1' (and not '0').

With inner switches of DIP switch in positions shown in the table, the count output can go from '0001' to the maximum count shown under 'Dice Range' in the table when switch S1 is depressed. On releasing switch S1, the last count within the dice range gets displayed.

The outputs of IC2 are displayed on common-anode, 7-segment display LTS542 (DIS1). BCD-to-7-segment decoder IC 7447 (IC3) is used to drive the display. Resistor R8 limits the current through DIS1.

Setting of the 4-way DIP Switch For Display Ranges

Dice range	Close the inner Switch	Open the inner switch
1 to 2	B and A	D and C
1 to 3	C only	A, B and D
1 to 4	A and C	B and D
1 to 5	B and C	A and D
1 to 6	A, B, and C	D only
1 to 7	D only	A, B and C
1 to 8	A and D	B and C
1 to 9	B and D	A and C



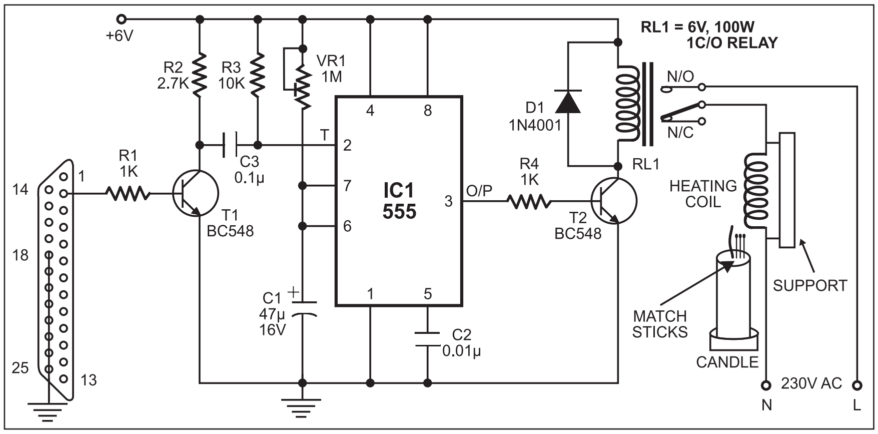
PC-BASED CANDLE IGNITOR

R. KARTHICK

Here's a PC-based lighting system that lets you light up a candle us-ing matchsticks by just pressing the 'Enter' key on the PC's keyboard. It is especially useful when celebrating such occasions as birthdays and anniversaries.

The number of matchsticks required to light up the candle is placed on the candle (alongside its wick) as shown in the figure. The heating coil for igniting the matchsticks is kept near them.

The interface circuitry between the PC and the heating coil for the candle-matchsticks arrangement comprises an inverter,



monostable and relay driver. Transistor BC548 (T1) acts as the inverter, IC 555 (IC1) is configured as the monostable circuit and transistor SL100 (T2) is the relay driver.

When you press ‘Enter’ key on the keyboard, the inverted output at the collector of transistor T1 goes low to trigger IC1 through its pin 2. Output pin 3 of the monostable goes high and transistor T2 conducts for around 50 seconds.

The conduction of transistor T2 energises relay RL1, which, in turn, connects the heating coil to 230V AC through the normally opened (N/O) contact. In place of the heating coil, you can also use an electric cigarrette lighter. The heating coil becomes red hot when connected across the 230V AC and ignites the matchsticks. The flames of the matchsticks light up the candle.

The program, written in ‘C’ language, is simple and easy to understand. The parallel-port D-type female connector normally available on the back of the PC is used for outputting the data to the interfacing circuitry. The address 378H of parallel-port LPT1 is used in the program. The parallel-port pin 2 corresponding to data bit D0 sends the control signal to energise the relay, which, in turn, connects the load to AC mains.

This circuit uses only one output of the PC’s parallel port to light up the candle, but it can be extended to light up up to eight *diyas*/candles in *thiruvillaku* (as called in South India) by using eight outputs with a slight change in the program and adding seven similar circuits.

Note: The source code of this article is included in the CD.

DEEPAM.C

```
/* PC Based Lighting System */
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
void main()
 {
 int n;
 clrscr();
 outportb(0x0378,0);
 _setcursortype(_NOCURSOR);
 randomize();
 textcolor(2);
 gotoxy(40,25);
  cprintf("By KARTHI,K.L.N.COLLEGE OF ENGG,Madurai");
 while(!kbhit())
 {
 n=random(10);
 textcolor(n);
 gotoxy(23,13);
 cprintf("Press enter to light up the DEEPAM");
 delay(100);
 }
 outportb(0x0378,1);
 getch();
 getch();
}
```

SOLIDSTATE REMOTE CONTROL SWITCH

SEEMANT SINGH

Here is a solidstate remote control switch which uses readily available electronic components. The control circuit comprises the transmitter and receiver sections. The range of the transmitter is around seven metres.

The transmitter circuit (shown in Fig. 1) is built around a timer IC (555) wired as an astable multivibrator. It works off a 9V battery. When remote control switch S1 is pressed, the astable multivibrator built around IC1 starts oscillating at a frequency of about 38 kHz. The signal frequency at output pin 3 of IC1 is transmitted through two infrared diodes (IR LED1 and IR LED2). A green LED (LED1) connected to pin 3 glows whenever S1 is pressed, indicating the presence of a signal for transmission at the output of the multivibrator.

The output frequency F at pin 3 of IC1 depends on the timing components, viz, resistors R1 and R2 and capacitor C2. It is given by the following relationship:

$$F = 1.443/(R1+2R2)C2$$

This frequency is fed to npn transistors T1 and T2 (each BC547) through resistor R4 (470-ohm) to drive the IR LEDs. Resistor R5 limits the current flowing through the IR LEDs.

The receiver circuit (shown in Fig. 2) consists of regulator IC 7806 (IC4), IR receiver module (TSOP1738), timer 555 (IC2) and decade counter CD4017 (IC3). Timer 555 (IC2) is wired as a monostable multivibrator.

The 9V DC power supply for the receiver circuit is regulated by regulator IC 7806. The presence of power in the circuit is indicated by glowing of the red LED (LED2).

The IR receiver module (TSOP1738), which gets 5.1V power supply through zener diode ZD1, receives the transmitted signal of about 38 kHz. The signal is amplified by transistor BC558 (T3) and given to triggering pin 2 of IC2 through coupling capacitor C6.

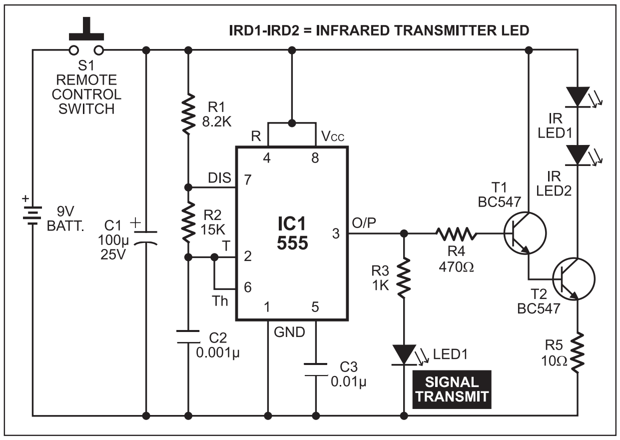


Fig. 1: Transmitter circuit

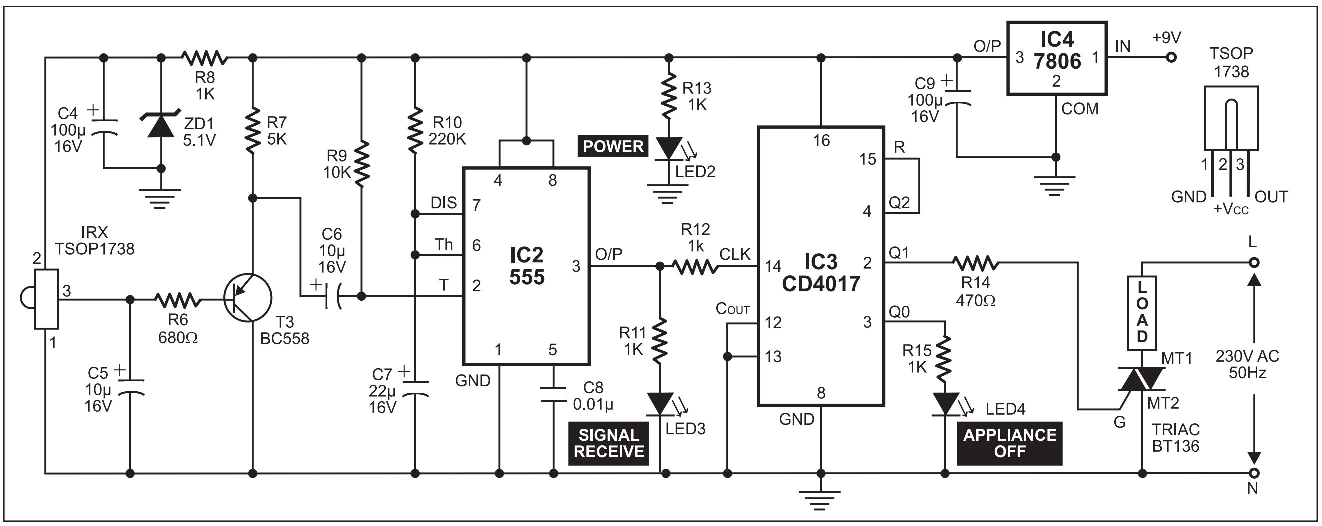


Fig. 2: Receiver circuit

Initially, when no signal is received from the transmitter, the output of the IR receiver module is high (approx. 5V).

When the transmitter is pointed at the receiver and switch S1 is momentarily pressed, the transmitted IR rays are sensed by the receiver module and its output pulses low to trigger the monostable (IC2). The output of IC2 goes high for about five seconds. Thus, even if you press the remote switch more than one time by mistake, there won't be any change in the output of the receiver within this period and hence no undesired switching of the appliance. The signal reception is indicated by glowing of the green LED (LED3).

The output of IC2 is given to the clock input (pin 14) of IC3. Here, IC3 is wired as a bistable circuit. For every clock input, pins 2 and 3 of IC3 alternately go high.

Initially, when the power to the receiver circuit is switched on, pin 3 of IC3 is high and therefore the yellow LED (LED4) connected to it glows. The glowing of LED4 indicates that the appliance is in 'off' condition.

When a clock pulse is received at pin 14 of IC3, pin 3 goes low to turn off LED4, while pin 2 becomes high. The high output at pin 2 triggers the gate of triac BT136, which, in turn, controls the appliance.

Precautions. Don't touch the leads of the triac as it is connected across the 230V AC mains. Also, make sure that the neutral point of mains is connected to the ground line of the circuit and not vice versa.

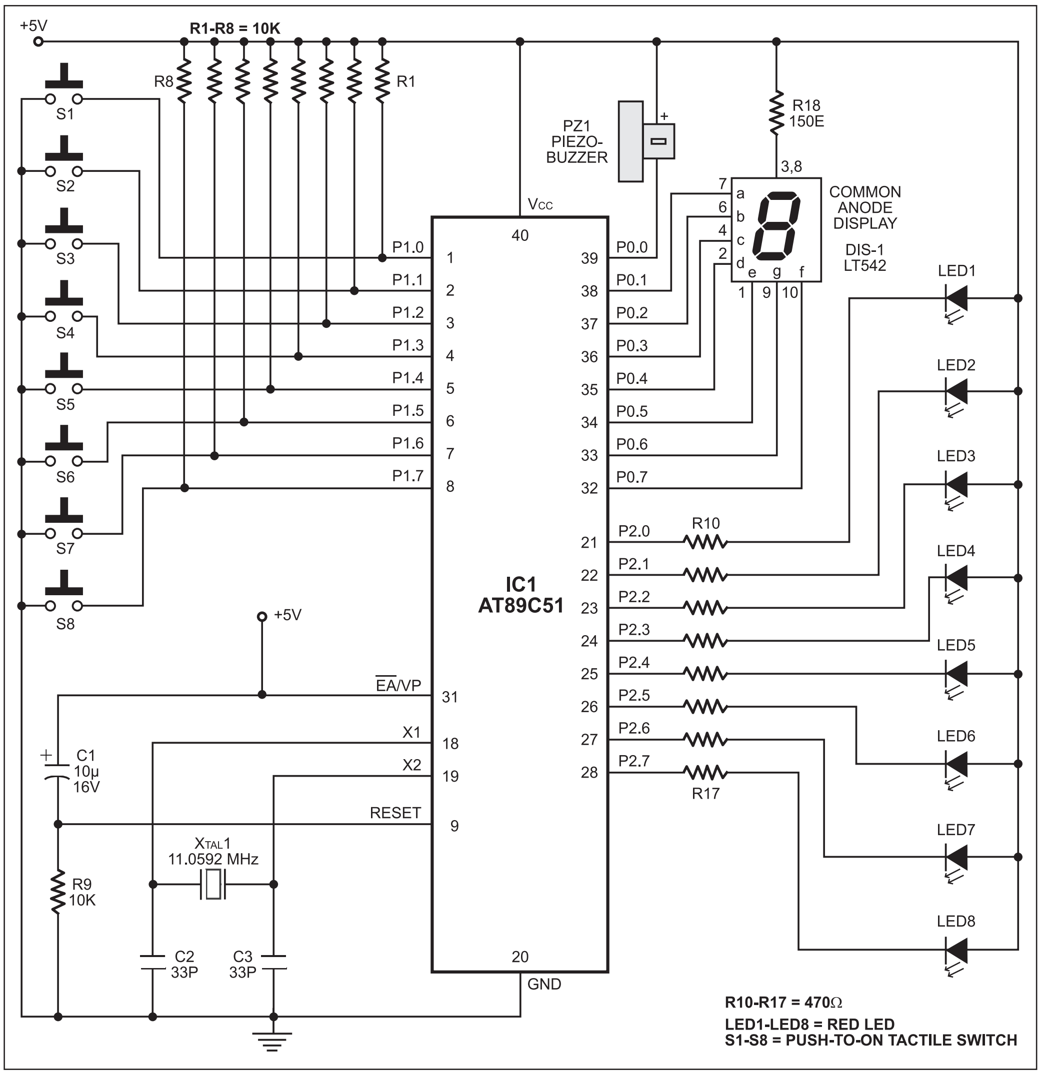
MICROCONTROLLER-BASED MONITORING SYSTEM

MANEESH CHADHA

In establishments such as small hotels, small offices and clinics, intercoms or calling bells prove to be a costlier option for communication between inmates and the assisting staff since such a provision can be made only for a limited number of points. Here's a simple and economical room-monitoring system that provides audio-visual identification of

Codes for Display Generation along with RAM Locations for Their Storage

LT542 pin no.		10	9	1	2	4	6	7	Buzzer	Hexa-	RAM
LT542 segment		g	F	e	D	C	B	A	input	decimal	memory
Inputs	Display	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	Code	Location
S1(P1.0)	1	1	1	1	1	0	0	1	0	F2H	68H
S2(P1.1)	2	0	1	0	0	1	0	0	0	48H	67H
S3(P1.2)	3	0	1	1	0	0	0	0	0	60H	66H
S4(P1.3)	4	0	0	1	1	0	0	1	0	32H	65H
S5(P1.4)	5	0	0	1	0	0	1	0	0	24H	64H
S6(P1.5)	6	0	0	0	0	0	1	0	0	04H	63H
S7(P1.6)	7	1	0	1	1	0	0	0	0	0BH	62H
S8(P1.7)	8	0	0	0	0	0	0	0	0	00H	61H



the call point. The system also provides feedback to the caller (in the form of busy signal). Using minimal hardware and software, it's a clean and easy way to communicate.

Flash-based microcontroller IC AT89C51 (IC1) is at the heart of the monitoring circuit. Ports 0, 1 and 2 of IC1 are used as output, input and output ports, respectively. Switches S1 through S8 are interfaced as inputs to controller IC1 via port 1 (p1.0 through p1.7). These switches, along with the respective LEDs, are to be installed in eight different rooms, while the remaining circuit is to be placed in the control room. Resistors R1 through R8 are pull-up resistors, while resistors R10 through R17 are current-limiting resistors. Other passive components constitute the reset and clock circuitry for operating the microcontroller.

When any of the switches is pressed, the corresponding call-point number is displayed on 7-segment, common-anode display DIS1 (LTS542) in the control room. This display is directly interfaced to output port 0 (pins P0.1 through P0.7) of controller IC1. Port pin P0.0 is connected to piezobuzzer PZ1, which sounds to indicate that someone needs help.

The audio-visual indication continues for a few seconds. During this time, if any other switch is also pressed, the controller doesn't recognise that request but gives busy signal to all the eight LEDs connected at its output port 2. As a result, irrespective of any switch being pressed, all the LEDs connected to the controller start blinking for this duration, so a caller gets to know that he has to wait for some time. When the LED stops blinking, he can press the switch for help. The LED again blinks to indicate that the request is being processed.

Written in Assembly language, the program for the microcontroller (MS. ASM) is simple and easy to understand. It is given at the end of the article. The table provides codes for generating the 7-segment display and the RAM locations of the microcon-troller where these are to be stored.

The complete Assembly program can be written using any text editor. Save the file as ‘MS.ASM.’ Generate the hex file (MS.HEX) by using the ASM51.EXE assembler. (This assembler was included in EFY-CD of March 2003 issue at path ‘E:\Software \Efysoftware\TEMar03\89C51Pgmr\Test Setup.’)

The complete procedure can be summarised as:

1. Program the micro-controller with the MS.HEX file using AT89C51 programmer.

2. After successfully programming the code, take the microcontroller out from the programmer and connect it to its IC base on the PCB of the circuit.

3. After assembling and soldering all other components, connect a 5V DC external power supply.

4. Now if you press any switch, the corresponding call-point number should display for a few seconds. At the same time, LEDs at all the call points should blink and the buzzer in the control room should sound for this duration.

5. If the kit is working properly, install the main unit with 7-segment display and buzzer in the control room and all the eight switches with LEDs beside them at different call points.

Note: The source code of this article is included in the CD.

MS.LST

```
0000                ORG     00H         ;locate reset routine at 00H
0000    0143        AJMP START          ;jump to START of main program
0003                ORG     03H         ;locate interrupt 0
0003    32          RETI                ;returns from external interrupt 0
000B                ORG     0BH         ;locate timer 0 interrupt
000B    32          RETI                ;returns from timer 0 interrupt
0013                ORG     13H         ;locate interrupt 1
0013    32          RETI                ;returns from external interrupt 1
001B                ORG     1BH         ;locate timer 1 interrupt
001B    32          RETI                ;returns from timer 1 interrupt
0023                ORG     23H         ;locate serial port interrupt
0023    32          RETI                ;returns from serial port interrupt
0025                ORG     25H         ;locate beginning of delay program
0025                DELAYMS:
0025    7F00        MOV     R7,#00H     ;delay of millisecond
0027                LOOPA:
0027    0F          INC R7              ;increment R7 by one
0028    EF          MOV     A,R7        ;store R7 value to Accumulator (A)
0029    B4FFFB      CJNE    A,#0FFH,LOOPA
002C    22                  RET
002D                DELAYHS:
002D    7E00        MOV     R6,#00H     ;half second delay
002F    7D08        MOV     R5,#008H    ;initialize R5
0031                LOOPB:
0031    0E          INC R6              ;to call milliseconds delay
0032    1125        ACALL   DELAYMS     ;call ms delay routine above
0034    EE          MOV     A,R6        ;store R6 value to A
0035    70FA        JNZ     LOOPB           ;go to LOOPB unless R6=OO
0037    E5A0        MOV     A,0A0H      ;store port-2 value to A
0039    F4          CPL     A           ;complement A and output A to port 2 to
003A    F5A0        MOV     0A0H,A      ;blinks all port LEDs
003C    7400        MOV     A,#00H      ;initialize A
003E    1D          DEC     R5          ;decrement R5
003F    ED          MOV     A,R5        ;move R5 value to A
0040    70EF        JNZ     LOOPB       ;if A is not 0 then go to LOOPB
0042    22          RET                 ;delay routine eight times
0043                START:
0043    756100      MOV     61H,#00H    ;store 1st code at 61H
0046    7562B0      MOV     62H,#0B0H   ;store 2nd code at 62H
0049    756304      MOV     63H,#04H    ;store 3rd code at 63H
004C    756424      MOV     64H,#24H    ;store 4th code at 64H
004F    756532      MOV     65H,#32H    ;store 5th code at 65H
0052    756660      MOV     66H,#60H    ;store 6th code at 66H
0055    756748      MOV     67H,#48H    ;store 7th code at 67H
0058    7568F2      MOV     68H,#0F2H   ;store 8th code at 68H
005B                LOOP:
005B    E590        MOV     A,90H       ; continuously read the inputs
005D    B4FF02      CJNE    A,#0FFH,L1  ;detect any switch press
0060    015B        AJMP    LOOP        ;again jump to LOOP to read port-1
0062                        L1:
0062    7C00        MOV R4,#00H         ;scan for any switch is pressed
0064                L2:
0064    D3          SETB C              ;loop L2 unless carry=0 detected
0065    33          RLC     A           ;rotate A with carry flag
0066    0C          INC     R4          ;increment R4, each time loop L2 is run
0067    40FB        JC      L2
0069    EC          MOV     A,R4        ;R4 value for switch number
006A    2460        ADD     A,#60H      ;add A
006C    F8          MOV     R0,A        ;move Address pointer to register R0
006D    E6          MOV     A,@R0       ;address pointer R0 points to correct
                                        code stored
006E    F580        MOV     80H,A       ;code stored at A transferred to port-0
0070    75A000      MOV     0A0H,#00H   ;initialize port-2 to start LEDs blinking
0073    112D        ACALL   DELAYHS     ;call four seconds delay
0075    7580FF      MOV     80H,#0FFH
0078    75A0FF      MOV     0A0H,#0FFH
007B    015B        AJMP    LOOP        ;delay returns here
                    END                 ;end program
VERSION 1.2k ASSEMBLY COMPLETE, 0 ERRORS FOUND
```

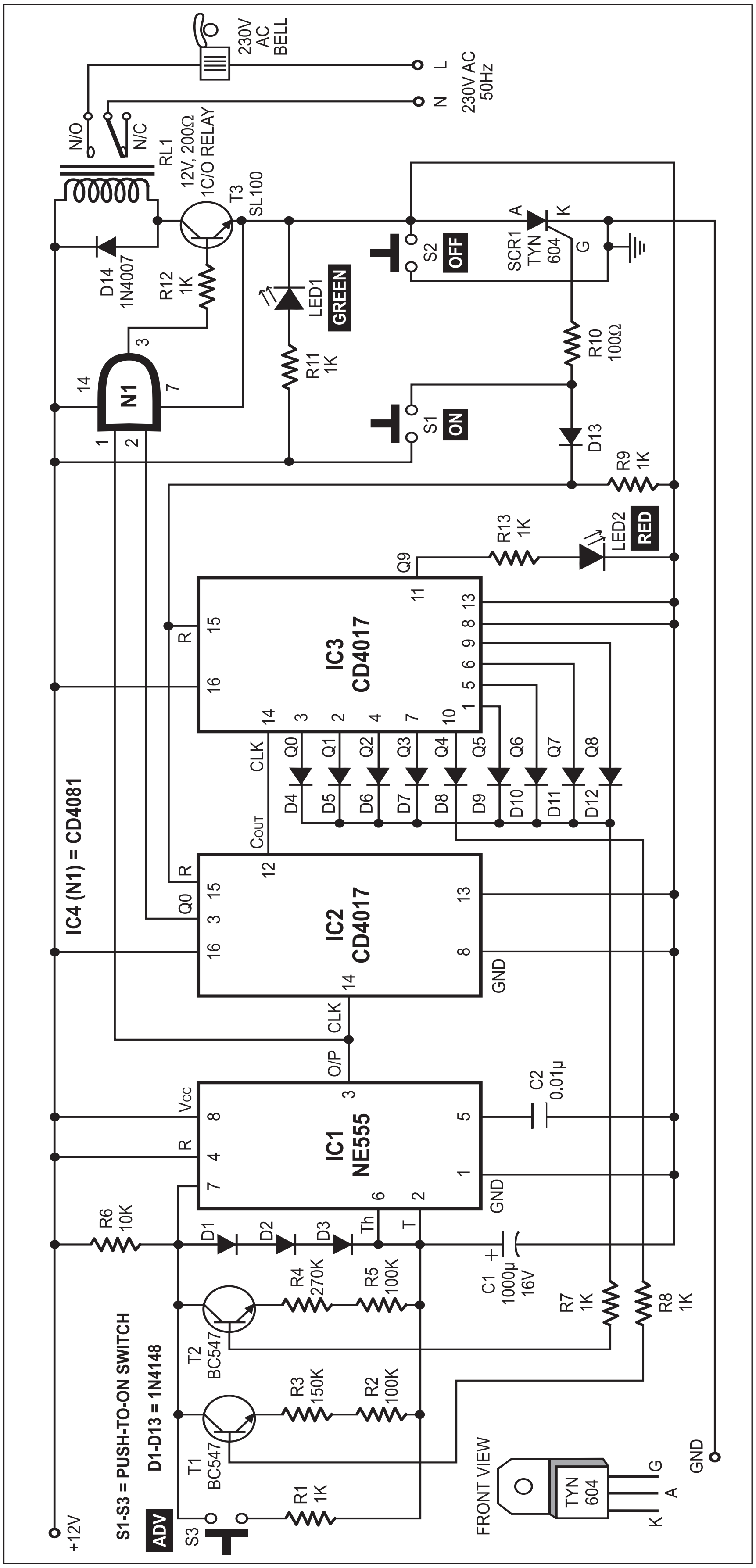
AUTOMATIC SCHOOL BELL

RAJ KUMAR MONDAL

Consider that a school has a total of eight periods with a lunch break after the fourth period. Each period is 45 minutes long, while the duration of the lunch break is 30 minutes.

To ring this automatic school bell to start the first period, the peon needs to momentarily press switch S1. Thereafter, the bell sounds every 45 minutes to indicate the end of consecutive periods, except immediately after the fourth period, when it sounds after 30 minutes to indicate that the lunch break is over. When the last period is over, LED2 glows to indicate that the bell circuit should now be switched off manually.

In case the peon has been late to start the school bell, the delay in minutes can be adjusted by advancing the time using switch S3. Each pushing of switch S3 advances the time by 4.5 minutes. If the



school is closed early, the peon can turn the bell circuit off by momentarily pressing switch S2.

The bell circuit contains timer IC NE555 (IC1), two CD4017 decade counters (IC2 and IC3) and AND gate CD4081 (IC4). Timer IC1 is wired as an astable multivibrator, whose clock output pulses are fed to IC2. IC2 increases the time periods of IC1 (4.5 and 3 minutes) by ten times to provide a clock pulse to IC3 every 45 minutes or after 30 minutes, respectively. When the class periods are going on, the outputs of IC3 switch on transistors T1 and T2 via diodes D4 through D12.

Resistors R4 and R5 connected in series to the emitter of npn transistor T2 decide the 4.5-minute time period of IC1. The output of IC1 is further connected to pin 14 of IC2 to provide a period with a duration of 45 minutes. Similarly, resistors R2 and R3 connected in series to the emitter of npn transistor T1 decide the 3-minute time period of IC1, which is further given to IC2 to provide the lunch-break duration of 30 minutes.

Initially, the circuit does not ground to perform its operation when 12V power supply is given to the circuit.

When switch S1 is pressed momentarily, a high enough voltage to fire silicon-controlled resistor SCR1 appears at its gate. When SCR1 is fired, it provides ground path to operate the circuit after resetting both decade counters IC2 and IC3. At the same time, LED1 glows to indicate that school bell is now active.

When switch S2 is pressed momentarily, the anode of SCR1 is again grounded and the circuit stops operating. In this condition, both LED1 and LED2 don't glow.

When the eighth period is over, Q9 output of IC3 goes high. At this time, transistors T1 and T2 don't get any voltage through the outputs of IC2. As a result, the astable multivibrator (IC1) stops working.

The school bell sounds for around 8 seconds at the end of each period. One can increase/decrease the ringing time of the bell by adding/removing diodes connected in series across pins 6 and 7 of IC1.

The terminals of the 230V AC electric bell are connected to the normally-open (N/O) contact of relay RL1. The circuit works off a 12V regulated power supply. However, a battery source for back-up in case the power fails is also recommended.

AUTOMATIC WATER PUMP CONTROLLER

R. ARAVIND & V. PRADEEP KUMAR

Here's a circuit that automatically controls the water pump motor. The motor gets automatically switched on when water in the overhead tank (OHT) falls below the lower limit. Similarly, it gets switched off when the tank is filled up. Built around only one NAND gate IC (CD4011), the circuit is simple, compact and economical. It works off a 12V DC power supply and consumes very little power.

The circuit can be divided into two parts: controller circuit and indicator circuit.

Fig. 1 shows the controller circuit. Let us consider two reference probes 'A' and 'B' inside the tank, where 'A' is the lower-limit probe and 'B' is the upper-limit probe. The 12V DC power supply is given to probe C, which is the limit for minimum water always stored in the tank.

The lower limit 'A' is connected to the base of transistor T1 (BC547), the collector of which is connected to the 12V power supply and the emitter is connected to relay RL1. Relay RL1 is connected to pin 13 of NAND gate N3.

Similarly, the upper-limit probe 'B' is connected to the base of transistor T2 (BC547), the collector of which is connected to the 12V power supply and the emitter is connected to pins 1 and 2 of NAND gate N1 and ground via resistor R3. The output pin 4 of NAND gate N2 is connected to pin 12 of NAND gate N3. The output of N3 is connected to input pin 6 of N2 and the base of transistor T3 via resistor R4. Relay RL2 connected to the emitter of transistor T3 is used to drive the motor.

If the tank is filled below probe A, transistors T1 and T2 do not conduct and the output of N3 goes high. This high output energises relay RL2 to drive the motor and it starts pumping water into the tank.

When the tank is filled above probe A but below probe B, water inside the tank provides base voltage to drive transistor T1 and relay RL1 energises to make pin 13 of gate N3 high. However, water inside the tank does not provide base voltage to transistor T2, so it does not conduct and the logic built around NAND gates N1 and N2 outputs low to pin 12 of gate N3. The net effect is that the output of N3 remains high and the motor continues pumping water into the tank.

When the tank is filled up to probe B level, water inside the tank still provides base voltage to transistor T1 and relay RL1 energises to make pin 13 of gate N3 high. At the same time, water inside the tank also provides base voltage to drive transistor T2 and the logic built around NAND gates N1 and N2 outputs high to pin 12 of gate N3. The net effect is that the output at pin 11 of N3 goes low and the motor stops pumping water into the tank.

Water-level Indication by LEDs

Level of water inside the tank	Glowing of LEDs
Full tank	LED1, LED2, LED3, LED4, LED5
¾ Tank	LED1, LED2, LED3, LED4
½ Tank	LED1, LED2, LED3
¼ Tank	LED1, LED2
Min. level	LED1

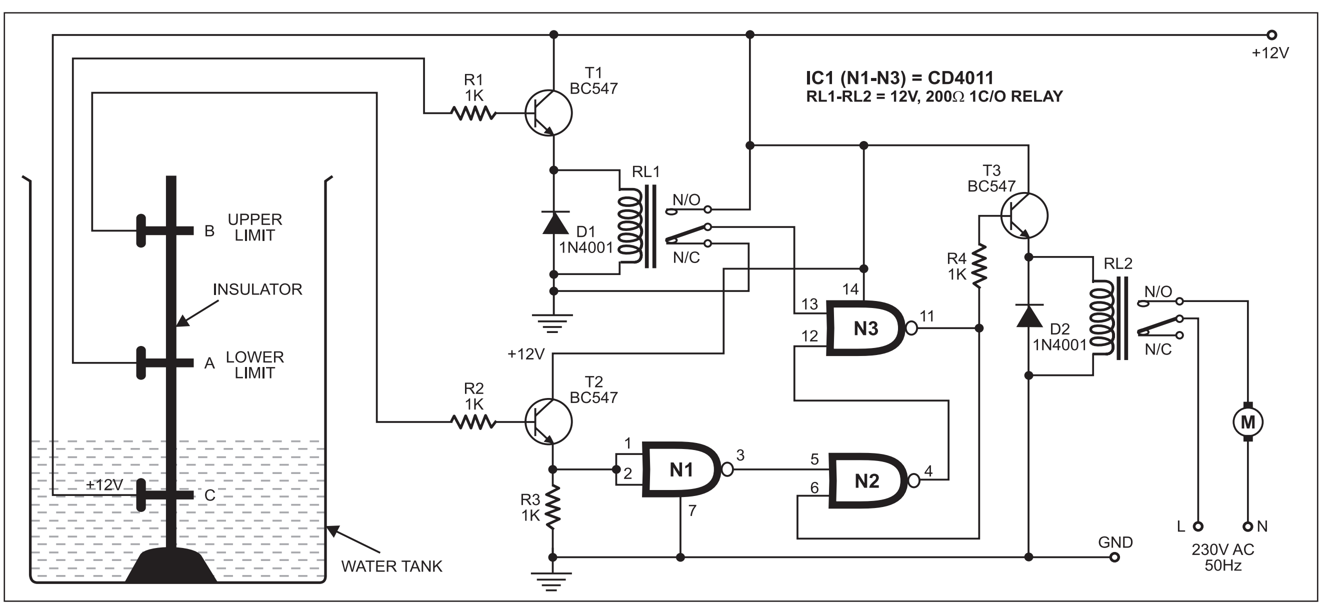


Fig. 1: Controller circuit

When water level falls below probe B but above probe A, water inside the tank still provides base voltage to transistor T1 and relay RL1 remains energised to make pin 13 of gate N3 high. However, transistor T2 doesn't conduct and the logic built around NAND gates N1 and N2 outputs high to pin 12 of N3. As a result, the output of N3 remains low and motor remains stopped.

When water level falls below probe A, both transistors T1 and T2 do not conduct. NAND gate N3 gives a high output to drive relay RL2 and the motor restarts pumping water into the tank.

Fig. 2 shows the indicator/monitoring circuit. It consists of five LEDs, which glow to indicate the level of water in the overhead tank. Since 12V power supply is given to water at the base of the tank, transistors T3 through T7 get base voltage and conduct to light up the LEDs (LED5 down through LED1).

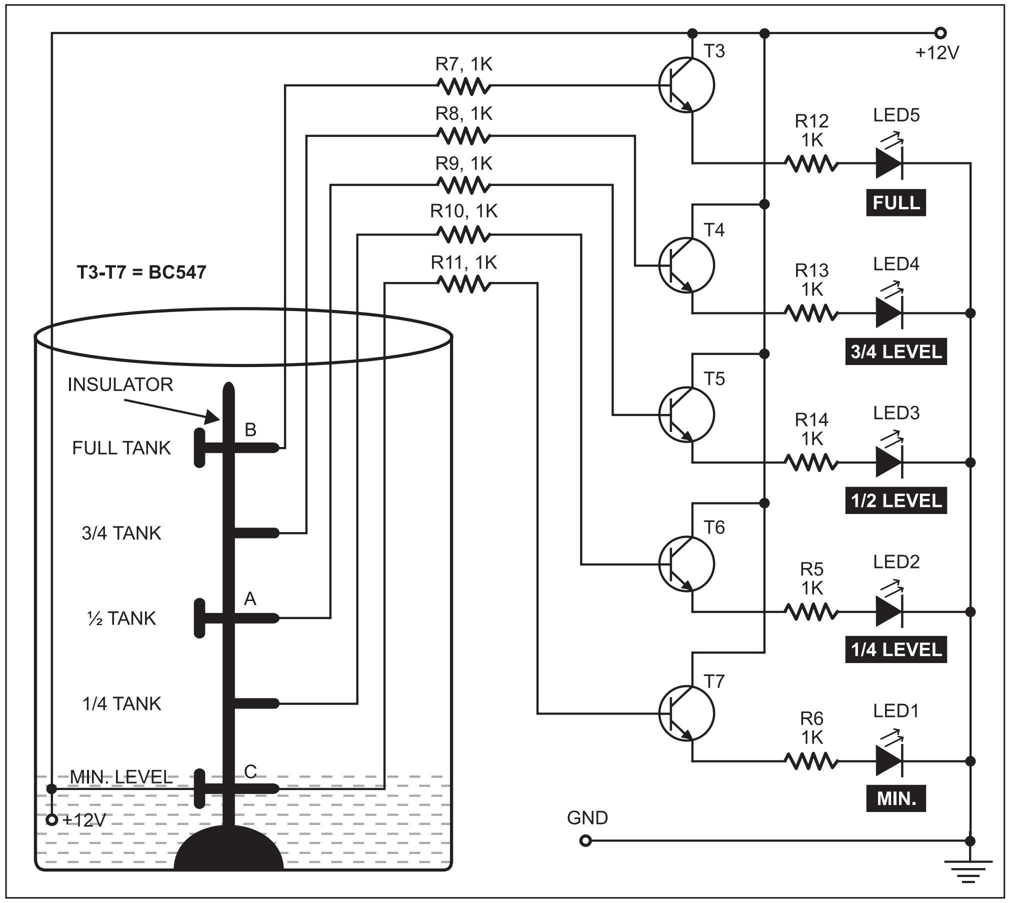


Fig. 2: Indicator/monitoring circuit

When water in the tank reaches the minimum at level C, transistor T7 conducts and LED1 glows. When water level rises to one-fourth of the tank, transistor T6 conducts and LED1 and LED2 glow. When water level rises to half of the tank, transistor T5 conducts and LED1, LED2 and LED3 glow. When water level rises to three-fourth of the tank, transistor T4 conducts and LED1 through LED4 glow. When the tank is full, transistor T3 conducts and all the five LEDs glow. So, from glowing of LEDs, one can know water level in the tank (see the table). The LEDs can be mounted anywhere for easy monitoring.

Note. The user can adjust the level to which water must be filled in the tank by adjusting the heights of probes A and B. The stand and adjusting screws should be insulated to avoid shorting.

NOISE METER

D. MOHAN KUMAR

Normally, sound intensity up to 30 dB is pleasant. Above 80 dB, it becomes annoying. And if it goes beyond 100 dB, it may affect your psychomotor performance, detracting your attention and causing stress. Noise pollution may also affect your hearing ability.

Noise intensity level in households is around 47 dB. But hi-fi music systems and TV sets operated at high volumes add to this sound, posing a health hazard.

Here's a simple circuit that senses and displays the noise intensity level in your room. It also gives a warning beep when noise crosses the safe level of 30 dB.

The circuit comprises a sound intensity sensor and a display unit. The regulator circuit built around regulator IC 7809 (IC1) provides regulated 9V power supply to the circuit.

The sound intensity sensor is built around a condenser microphone, op-amp IC CA3130 (IC2) and associated components. Op-amp IC2 is configured as a high-gain inverting amplifier. The voltage supply to IC2 at its non-inverting pin 3 is divided to half by resistors R3 and R4, which is also used as the reference voltage. Resistor R1 determines the sensitivity of the condenser microphone.

The microphone picks up sound vibrations and converts them into the corresponding electric pulses, which are fed to the inverting input of IC2 (pin 2) via capacitor C4 and resistor R2. Capacitor C4 blocks any DC entering the op-amp, since it may affect the functioning of the op-amp. The output of IC2 is connected to the inverting input through resistor R5 (10 mega-ohms) for negative feedback. Since the input impedance of IC2 is very high, even a small current can activate the op-amp.

The output of IC2 is given to preset VR1 via capacitor C5, which is used to control the volume. Capacitor C5 blocks DC, allowing only AC to pass through preset VR1. The AC signals from the wiper of VR1 are fed to a diode pump comprising diodes Dl and D2. The diode pump rectifies the AC and maintains it at the output level of IC2. Capacitor C6 acts as a reservoir capacitor for DC and resistor R6 provides the path for its discharge.

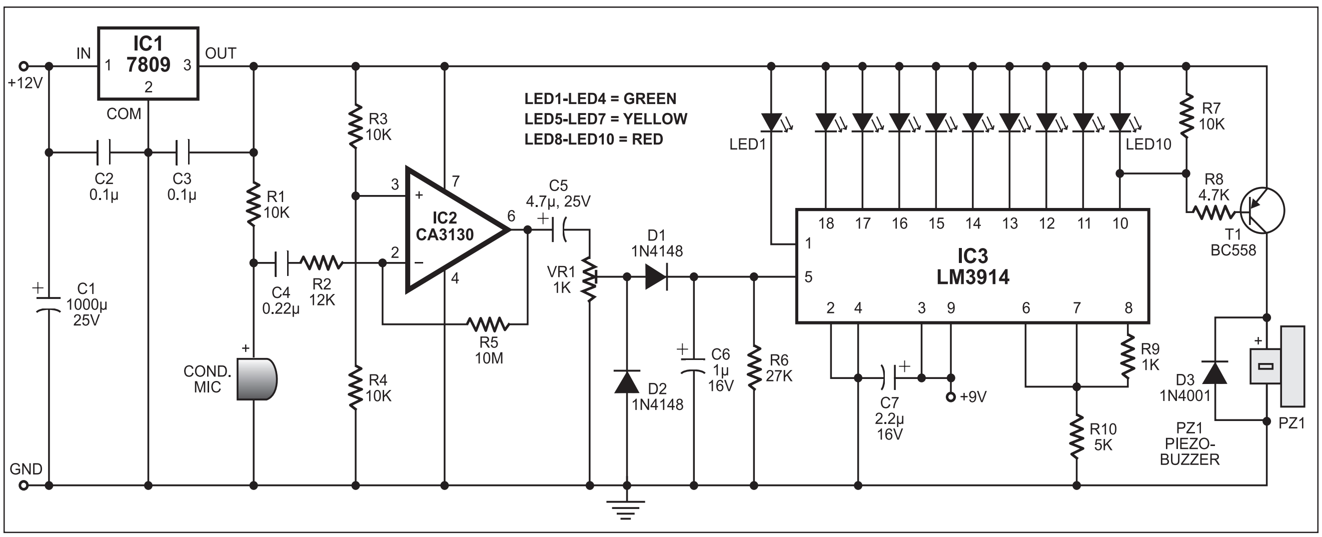
The display circuit is built around monolithic IC LM3914 (IC3), which senses the analogue voltage and drives ten LEDs to provide a logarithmic analogue display. Current through the LEDs is regulated by the internal resistors of IC3, eliminating the need for external resistors. The built-in low-bias input buffer of IC3 accepts signals down to ground potential and drives ten individual comparators inside IC3. The outputs of IC3 go low in a descending order from 18 to 10 as the input voltage increases.

Each LED connected to the output of IC3 represents the sound level of 3 dB, so when all the ten LEDs glow, it means the sound intensity is 30 dB.

Pin 9 of IC3 is connected to 9V to get the dot-mode display. In the dot-mode display, there is a small amount of overlap between segments. This assures that at no time will all LEDs be 'off.'

When output pin 10 of IC3 goes low, pnp transistor T1 gets base bias (normally cut-off due to resistor R7) to sound the piezobuzzer (PZ1) connected to its collector.

The circuit can be constructed on any general-purpose PCB. Condenser microphone should be connected using a shield wire and enclosed in a tube to increase its sensitivity. For audiovisual indications, use a small DC piezobuzzer and transparent LEDs. Adjust preset VR1 until only the first LED (LED1) lights up. Keep the circuit near the audio equipment or TV set to monitor the audio level.



ANTI-THEFT ALARM FOR BIKES

PRAVEEN KUMAR M.P.

If anybody tries to steal your bike, this circuit turns on the horn of the bike to alert you of the impending theft.

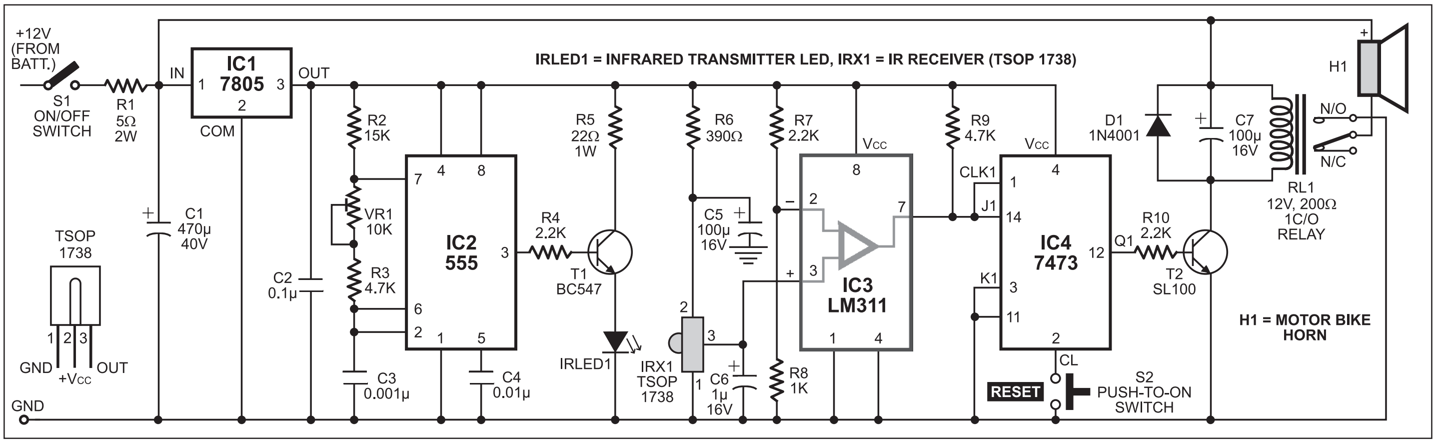
Usually, a handle lock is used on the handle bar for the safety of bikes, with the front mudguard in a slanted position. When the handle lock is freed, the front mudguard can be aligned with the body of the bike.

This circuit consists of transmitter and receiver sections. The transmitter (IR LED1) is fitted on the back end of the front mudguard and the receiver sensor (IRX1) is fitted on the central portion of the crash guard of the bike such that IR rays from the transmitter directly fall on the sensor when the front mudguard comes in line with the body of the bike.

The transmitter section is built around timer 555 (IC2), which is wired as an astable multivibrator with a frequency of around 38 kHz. The output of IC2 is further amplified by transistor T1 and given to an infrared light-emitting diode (IR LED1), which continuously transmits the IR frequency.

The receiver section uses IR receiver module TSOP 1738 (IRX1), which is normally used in TV receivers. The receiver module senses the IR modulated frequency transmitted by the IR LED.

When no IR rays are incident on the sensor, its output is high. But the output of the IR sensor goes low when it senses the modulated IR signal. The output of the receiver module is given to a negative-voltage comparator built around IC LM311 (IC3). The input voltage at pin 2 of IC3 is



fixed by using the voltage-divider network comprising resistors R7 and R8.

When IR rays are not incident on the IR receiver module, the voltage at pin 3 of IC3 is greater than the voltage at pin 2. As a result, the output of comparator IC3 is low. But when the receiver senses IR rays from IR LED1, the voltage at pin 3 of IC3 is lower than the voltage at pin 2. As a result, the output of the comparator goes high.

The output of the comparator is given to a latch made up of JK flip-flop (IC4). The low-to-high going pulse from the comparator makes the output of IC4 high until it is reset. The output of IC4 is latched and used to energise relay RL1 via transistor T2. The relay is connected to the negative terminal of the mobike's horn, while the positive terminal of the horn is connected to the positive terminal of the battery via resistor R1. The energised relay drives the horn, which continues sounding until you press reset switch S2 momentarily.

At night, lock your bike using the handle lock and switch on the circuit using switch S1. Since the IR transmitter (IR LED1) and the receiver (IRX1) will not be in line of sight, IR rays from IR LED1 will not be incident on the sensor. When anyone tries to move the bike away, the IR transmitter and the IR receiver will come in line of sight and the IR rays from the IR transmitter will be incident on the receiver. This will make the output of the comparator (IC3) high. The pulse from the comparator will make the output of latch IC4 high and transistor T2 will conduct to sound the horn via relay RL1.

Note. The circuit excluding the transmitter and the receiver can be housed in a small metal box and kept inside the tool box of the bike.

Before you start your bike, make sure that the circuit is switched off using switch S1.

TIMER WITH MUSICAL ALARM

PRADEEP G.

This low-cost timer can be used for introducing a delay of one minute to two hours. After the timing period is over, a musical song is heard.

The circuit is built around popular CMOS oscillator/divider CD4060 (IC1). It works off a 9V PP3 battery and its standby current drain is very low.

By adjusting preset VR1, the time delay can be adjusted. After time delay is over, output pin 3 of IC1 goes high and npn transistor T1 conducts to provide positive power supply to melody generator IC UM66 (IC2) at its pin 2. Zener diode ZD1 reduces this power supply to 3.3V required for operation of IC2. The output of IC2 is fed to the loudspeaker (LS1) via driver transistor T1.

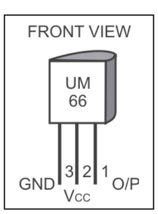


Fig. 1: Pin configuration of melody generator IC UM66

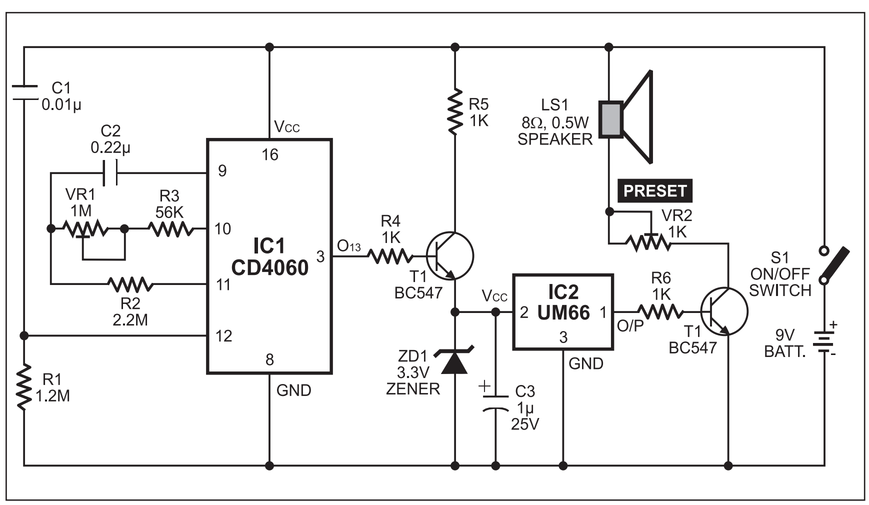


Fig. 2: The circuit of timer with musical alarm

Preset VR2 is used to control the volume of the loudspeaker.

The timer gets activated when power is supplied by pressing switch S1. To switch off the alarm, you need to switch off the power supply.

MAINS FAILURE/RESUMPTION ALARM

V. DAVID

This mains indicator sounds an alarm whenever AC mains fails or re-sumes. It is very useful in industrial installations, cinema halls, hospitals, etc.

The mains detector circuit is built around capacitors C1 and C2, resistor R1, and diodes D1 and D2. It provides sufficient voltage for the glowing of internal LED of optocoupler MCT2E (IC1).

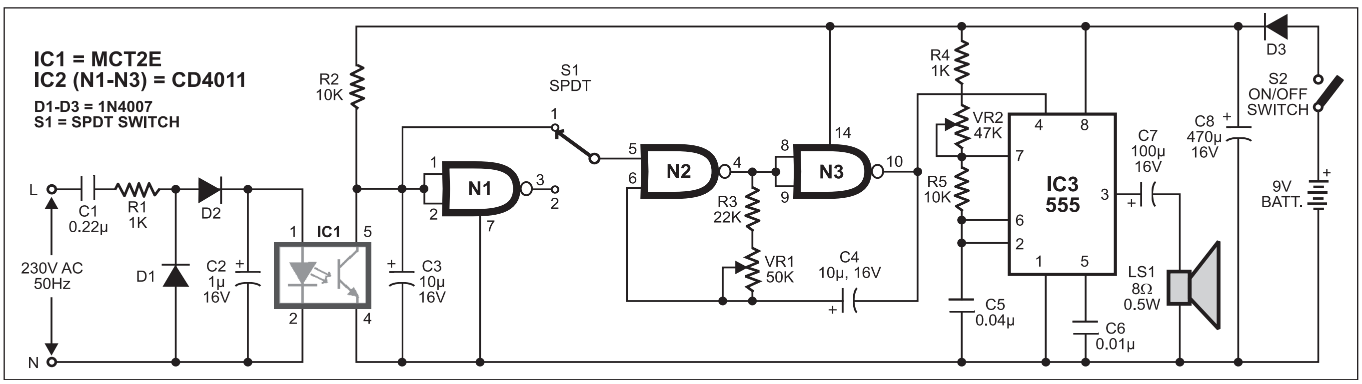
Initially SPDT switch S1 is at position 1. When mains fails, pin 5 of gate N2 goes high and the oscillator built around gates N2 and N3 of IC2 produces low-frequency oscillations at pin 10, which are further given to pin 4 of IC 555 (IC3). The oscillation frequency can be varied from 0.662 Hz to 1.855 kHz using preset VR1.

IC 555 (IC3) is wired as an audio tone generator. The tone of this audio oscillator can be varied from 472 Hz to 1.555 kHz using preset VR2. The low-frequency input activates IC3 to generate audio tones and loudspeaker LS1 connected to its output pin 3 sounds an alarm indicating mains failure.

To turn off the alarm, slide the pole of switch S1 to position 2. Now the circuit is ready for sensing the mains resumption.

When mains resumes, pin 5 of gate N2 goes high and the oscillator built around gates N2 and N3 of IC2 produces low-frequency oscillations at pin 10, which are given to reset pin 4 of IC3. As a result, loudspeaker LS1 again sounds to indicate that mains has resumed. To turn off the alarm, slide the pole of switch S1 back to position 1. Now the circuit is again ready for sensing the mains failure.

The circuit works off a 9V battery. It can be housed in a box and installed where you want to monitor the status of mains.

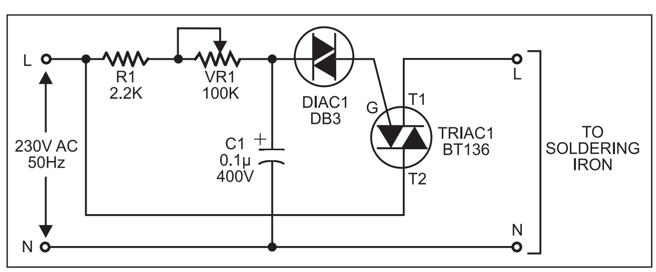


SOLDERING IRON TEMPERATURE CONTROLLER

P.V. VINOD KUMAR

Here is a simple circuit to control the temperature of a soldering iron. It is especially useful if the soldering iron is to be kept on for long since you can control the heat dissipation from the iron. When a soldering iron is switched on, the iron takes time to reach the solder's melting point. Simply connect this circuit to the soldering iron as shown in the figure and the iron reaches the solder's melting point quickly.

Triac BT136 is fired at different phase angles to get temperatures varying from zero to maximum. A diac is used to control the triac firing in both directions. Potentiometer VR1 is used for setting the temperature of the soldering iron.



The circuit can be housed in a box with the potentiometer fixed on the side such that its knob can be used from outside the box to adjust the soldering iron's temperature.

MULTIPURPOSE WHITE-LED LIGHT

N.S. HARISANKAR, VU3NSH

Standard fluorescent lamps and their smaller versions called compact fluo-rescent lamps (CFLs) radiate light in all directions (360°) and tend to increase the room temperature. In emergency lights using these lamps, the battery lasts only a few hours due to the power loss during conversion of DC into AC. These limitations can be overcome by using ultra-bright white LEDs.

Here is a torch-cum-table lamp using white LEDs that can also be modified to act as an emergency-cum-bedroom light. Its main features are long and continuous operation, very low power consumption, selectable light angle, very long life and negligible heat radiation.

Fig. 1 shows the circuit of white LEDs-based torch-cum-table lamp. The circuit is very simple and uses a battery charger unit built around IC LM317 (IC1) and a combination of white LEDs. Resistor R3 (4.7-ohm, 2W) limits the current through the battery. The radiation angles selected for white LEDs are 60° and 20°. Three columns of LED clusters (A, B and C) are made on separate transparent acrylic sheets, with each sheet having a total of twelve LEDs affixed to it.

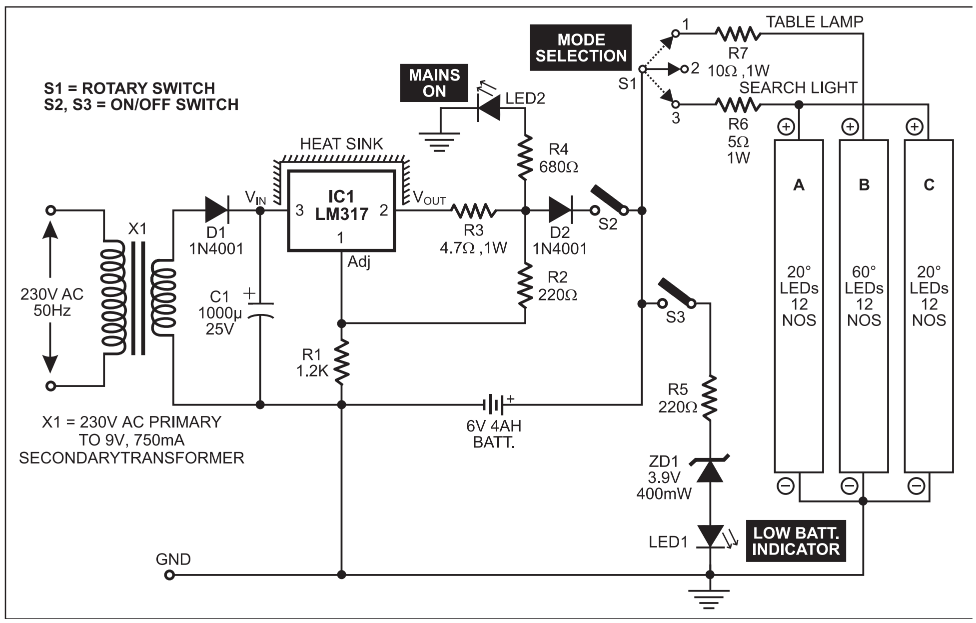


Fig. 1: Cluster LED searchlight/table lamp

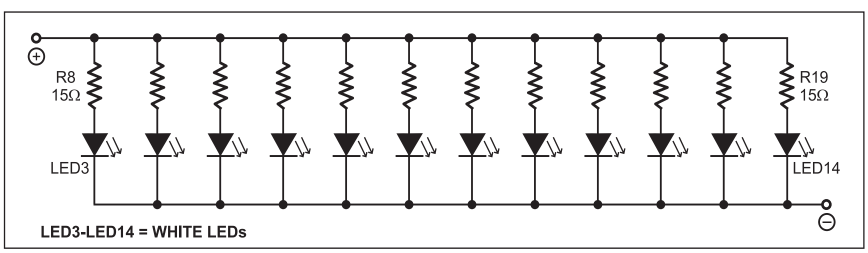


Fig. 2: Arrangement of LEDs for column A, B or C

The left (A) and right (C) columns use 20° LEDs, while the middle column (B) uses 60° LEDs. All the twelve LEDs of each column are connected in series to separate 15-ohm current-equalisation resistors (R8 through R19) as shown in Fig. 2, and to current-limiter resistors R7 (10-ohm, 1W) and R6 (5-ohm, 1W) as shown in Fig. 1. The entire unit is powered by a 6V, 4Ah maintenance-free rechargeable battery.

The continuous lighting life is around 7 hours in torchlight mode and around 14 hours in table lamp mode, depending on the battery capacity and quality. For the torch mode, only the left and right LED columns are used. These LEDs beam light up to 6 metres. In table lamp (spread light) mode, only the middle column of LEDs is used.

You can select between the table lamp and torch modes by using rotary switch S1, which is a single-pole, 3-way switch. When the pole of switch S1 is set at position 1, the C column of 60° LEDs lights up and the system acts as a table lamp. When the pole of switch S1 is set at position 3, columns A and C light up and the system acts as a torch. When the pole of switch S1 is at position 2, both the table lamp and the torch modes remain off.

When mains is switched on, LED2 glows. To charge the battery, flip switch S2 to 'on' position. To check the status of the battery, flip switch S3 to 'on' position. This will give an indication of battery charge. If low-battery indicator LED1 turns off, the battery needs to be charged.

Fig. 3 shows the circuit of emergency lamp with brightness control, which is derived from Fig. 1 with slight modification in the combination of LEDs. Built around four multichip (MC) LEDs, it is very compact and simple, and can work in two modes, namely, bedroom lamp and emergency lamp.

In bedroom lamp mode, only one blue LED glows. This LED is mounted at the top in upside down position to avoid direct viewing of the blue light. The arrangement gives a pleasant, well-spread light.

In emergency lamp mode, 8mm, 80° bright-white multichip LEDs give 80° spread light, which is sufficient for indoor uses. Circular PCBs for multichip LEDs have four internal junctions each. Solder LED17 through LED20 in the first PCB, LED21 through LED24 in the second PCB, LED25 through LED28

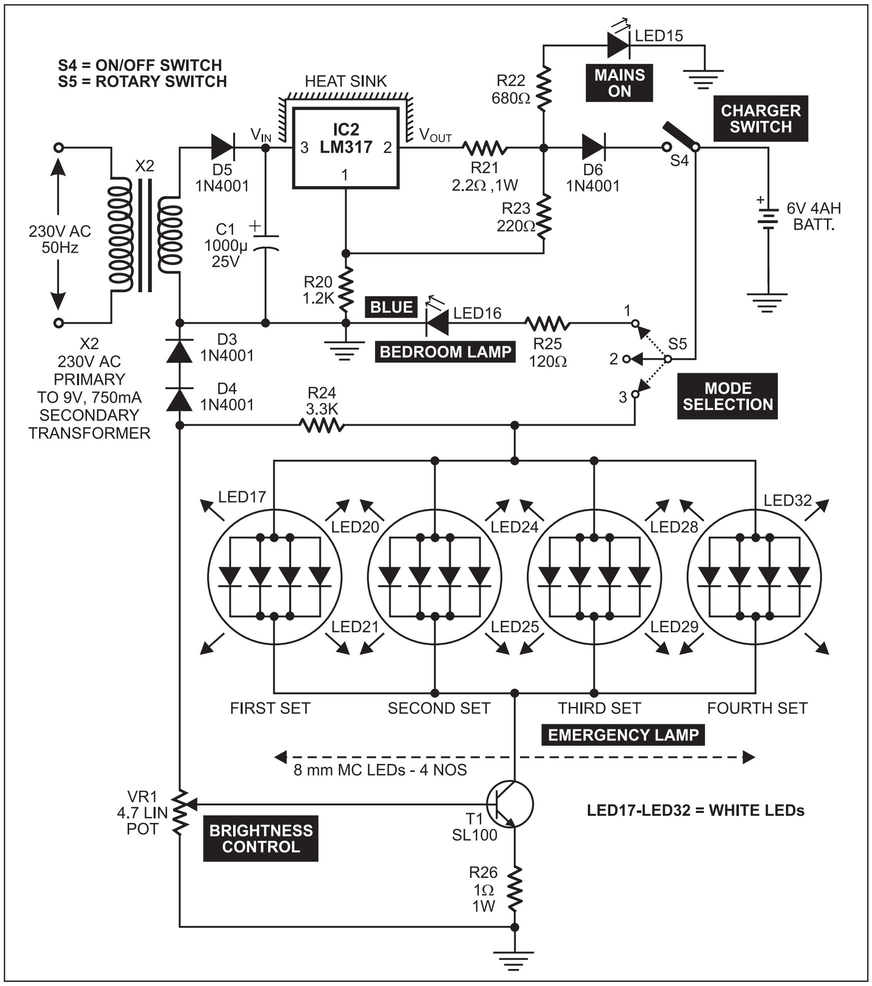


Fig. 3: Emergency lamp with brightness control

in the third PCB and LED29 through LED32 in the fourth PCB, with a spacing of 3 to 4 cm between two adjacent LEDs. Finally, house all the four circular PCBs in a compact cabinet along with the reflector such that light can spread out in the room.

Each multichip LED gives a power of 32 candles. Therefore use of four 8mm multichip LEDs will give a total power of 128 candles.

In emergency lamp mode (selected through rotary switch S5), all the four multichip LEDs (including LED17 through LED32) glow. The DC power source is a 6V, 4Ah chargeable battery, with charging circuit built around popular IC LM317 (IC2). Resistor R21 (2.2-ohm, 1W) acts as the current limiter for the battery.

You can control the candle power (brightness) of LEDs as per your requirements. Transistor SL100 (T1) and its associated components form the candle controller (brightness controller). The base biasing voltage of the transistor is stabilised by resistor R24 and diodes N3 and N4 (1N4001). This constant voltage is given to the base of the transistor through a potentiometer VR1 (4.7k lin.). By adjusting the potentiometer, you can control the intensity of the multichip LEDs. No heat-sink is required for the transistor.

ELECTRONIC WATCHDOG

TAPAN KUMAR MAHARANA

Here's an electronic watchdog for your house that sounds to inform you that somebody is at the gate. The circuit comprises a transmitter unit and a receiver unit, which are mounted face to face on the opposite pillars of the gate such that the IR beam gets interrupted when someone is standing at the gate or passing through it.

The transmitter circuit (see Fig. 1) is built around timer NE555 (IC1), which is wired as an astable multivibrator producing a frequency of about 38 kHz. The infrared (IR) beam is transmitted through IR LED1.

The receiver circuit is shown in Fig. 2. It comprises IR sensor TSOP1738 (IR RX1), npn transistor BC548 (T1), timer NE555 (IC2) and some resistors and capacitors. IC2 is wired as a monostable multivibrator with a time period of around 30 seconds. The melody generator section is built around melody generator IC UM66 (IC3), transistor T2 and loudspeaker LS1. Fig. 3 shows pin configurations of IR sensor TSOP1738 and melody generator IC UM66.

The power supply for the transmitter is derived from the receiver circuit by connecting its points A

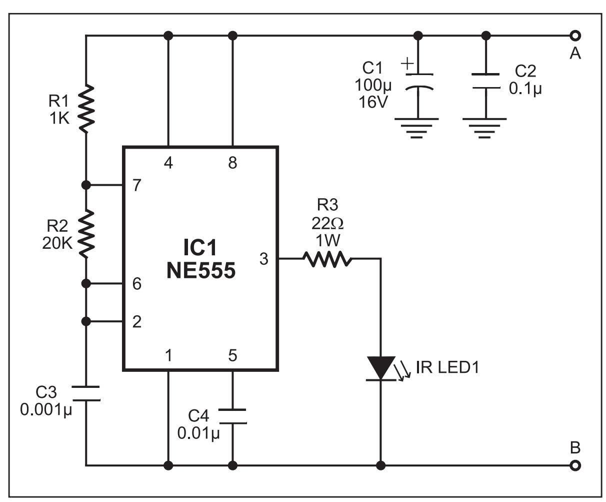


Fig. 1: 38kHz IR transmitter circuit

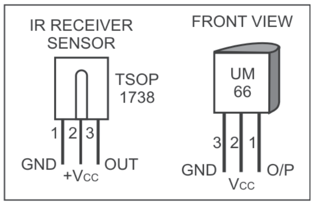


Fig. 3: Pin configurations of TSOP1738 and UM66

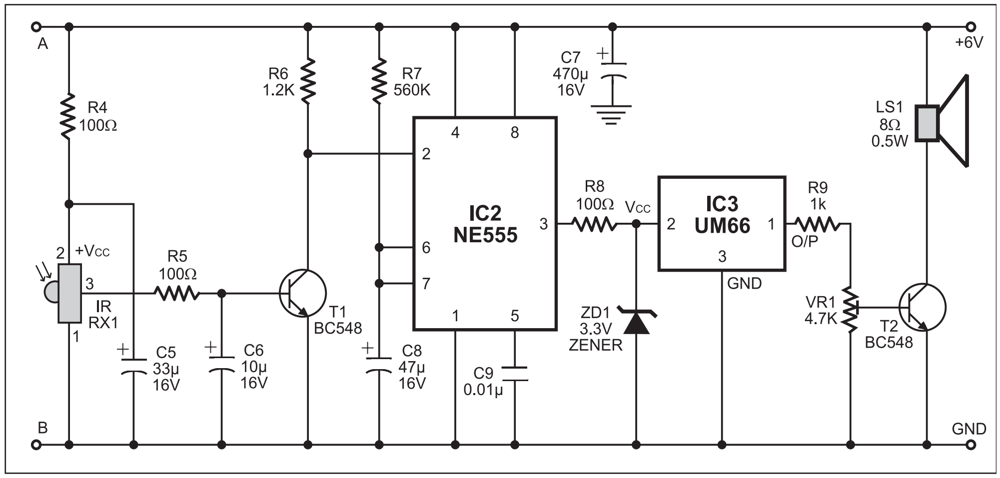


Fig. 2: Receiver circuit

and B to the respective points of the receiver circuit. The receiver is powered by regulated 6V DC. For the purpose, you can use a 6V battery.

The transmitter and receiver units are aligned such that the IR beam falls directly on the IR sensor. As long as IR beam falls on the sensor, its output remains low, transistor T1 does not conduct and trigger pin 2 of IC2 remains high.

When anyone interrupts the IR beam falling on the sensor, its output goes high to drive transistor T1 into conduction and pin 2 of IC2 goes low momentarily. As a result, IC2 gets triggered and its pin 3 goes high to supply 3.3V to melody generator IC3 at its pin 2, which produces a sweet melody through the speaker fitted inside the house. Output pin 3 of IC2 remains high for around 30 seconds.

Fig. 4 shows mounting arrangement for both the transmitter and receiver units on the gate pillars. To achieve a high directivity of the IR beam towards the sensor, use a reflector behind the IR LED.

After both the units have been built, connect 6V power supply to the receiver circuit. You should hear a continuous melody from the speaker. Now connect 6V power to the transmitter also and orient IR LED1 towards IR receiver. The melody should stop after about 30 seconds. Now the transmitter and the receiver units are ready for use.

When somebody enters through the door, the IR beam is interrupted and the alarm sounds for 30 seconds. The alarm keeps sounding as long as one stands between the transmitter and receiver units. Using preset VR1, you can set the volume of the loudspeaker.

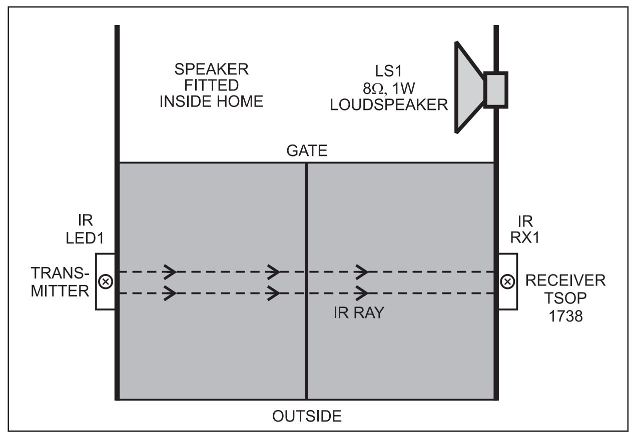


Fig. 4: Mounting arrangement for transmitter and receiver units

This circuit can also be used as a doorbell or burglar alarm.

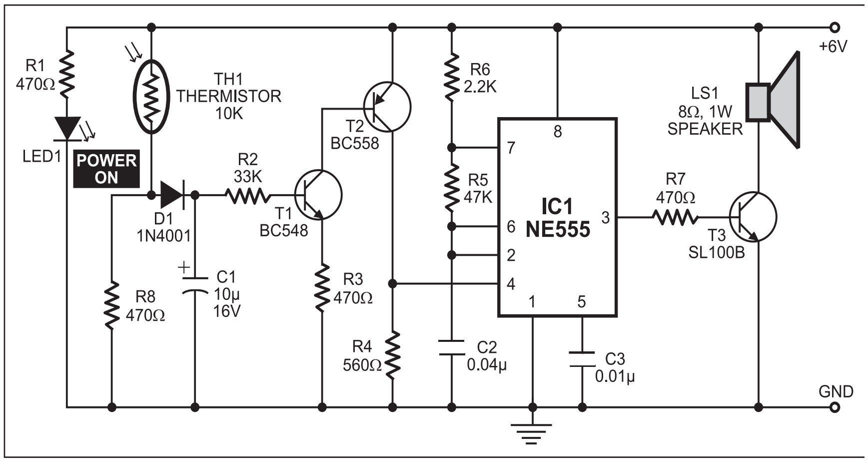
FIRE ALARM USING THERMISTOR

PRINCE PHILLIPS

In this fire alarm circuit, a thermistor works as the heat sensor. When temperature increases, its resistance decreases, and vice versa. At normal temperature, the resistance of the thermistor (TH1) is approximately 10 kilo-ohms, which reduces to a few ohms as the temperature increases beyond 100°C. The circuit uses readily available components and can be easily constructed on any general-purpose PCB.

Timer IC NE555 (IC1) is wired as an astable multivibrator oscillating in audio frequency band. Switching transistors T1 and T2 drive multivibrator NE555 (IC1). The output of IC1 is connected to npn transistor T3, which drives the loudspeaker (LS1) to generate sound. The frequency of IC1 depends on the values of resistors R5 and R6 and capacitor C2.

When thermistor TH1 becomes hot, it provides a low-resistance path to extend positive voltage to the base of transistor T1 via diode D1 and resistor R2. Capacitor C1 charges up to the positive voltage and increases the 'on' time of alarm. The higher the value of capacitor C1, the higher the forward voltage applied to the base of transistor



T1 (BC548). Since the collector of transistor T1 is connected to the base of transistor T2, transistor T2 provides positive voltage to reset pin 4 of IC1 (NE555). Resistor R4 is used such that IC1 remains inactive in the absence of positive voltage. Diode D1 stops discharging of capacitor C1 when the thermistor connected to the positive supply cools down and provides a high-resistance (10-kilo-ohm) path. It also stops the conduction of T1. To prevent the thermistor from melting, wrap it up in mica tape.

The circuit works off a 6V-12V regulated power supply. LED1 is used to indicate that power to the circuit is switched on.

TWILIGHT LAMP BLINKER

T.K. HAREENDRAN

During sunset or sunrise, the ambient light is not adequate to lead you through the open doorway or make your way around obstructions. To avoid any mishap, here is a twilight lamp blinker that you can place near obstructions.

Fig. 1 shows the circuit of the twilight lamp blinker. For powering the circuit, the mains input (230V AC) is down-converted by resistors R1 and R2, capacitor C1 and diodes D1 and D2 into a DC voltage that is low enough to safely charge the back-up battery pack. Resistor R2 across capacitor C1 functions as a bleeder resistor. Zener diode ZD2 protects against over-voltage.

Miniature Ni-Cd battery packs for cordless telephones are easily available at reasonable rates. Use such a battery pack with 4.8V, 500mAh rating for efficient and long-time back-up. The pole of switch S1 should be in position 2 if you use a battery. If you are not interested in the back-up facility, flip switch S1 to position 1.

The rest of the circuit includes a light-detector resistor (LDR1), IC CD4093 (IC1) and a preset (VR1) for brightness control. LDR1 is used as a sensor that has a low resistance during daytime and a high resistance at night.

When light falls on the LDR, its low resistance provides low level at the inputs of NAND gate N1. The high input from N1 makes the output of N2 low and the relaxation oscillator (built around NAND gates N3 and N4 of IC1, capacitor C3 and

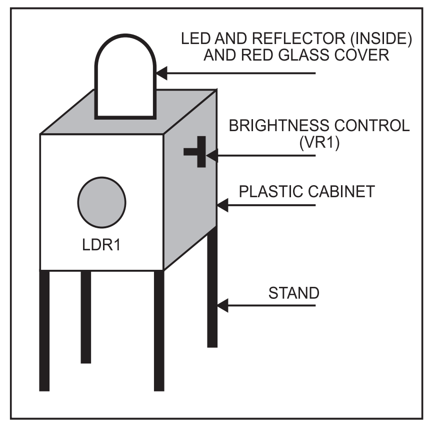


Fig. 2: Proposed enclosure

resistor R3) does not oscillate. As a result, transistor T1 does not conduct and LED1

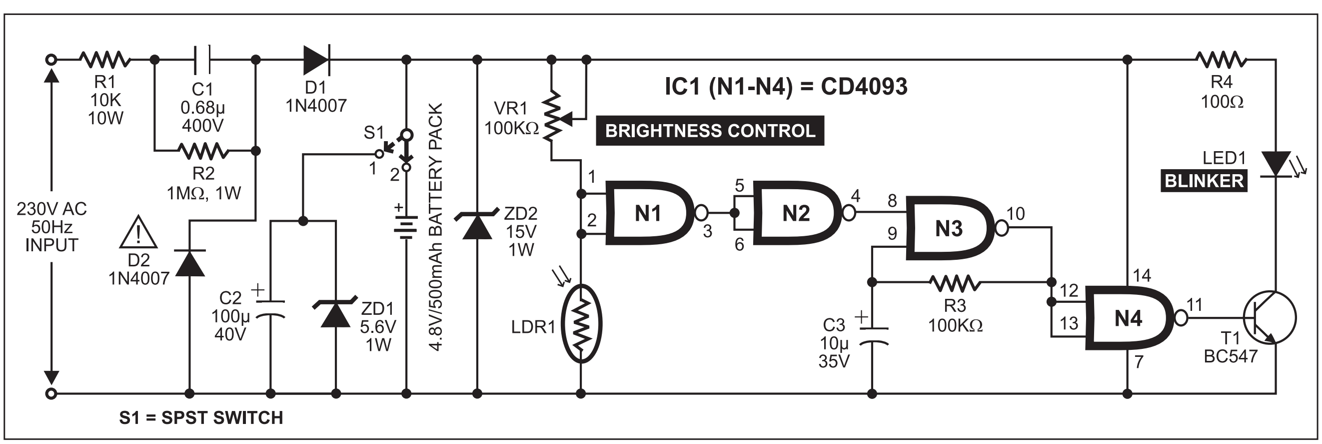


Fig. 1: Circuit diagram of twilight lamp blinker

does not blink.

On the other hand, in darkness, the high resistance of LDR1 provides a high level at the input pins of NAND gate N1. The low output from N1 makes the output of N2 high and the relaxation oscillator oscillates. As a result, transistor T1 conducts and LED1 blinks.

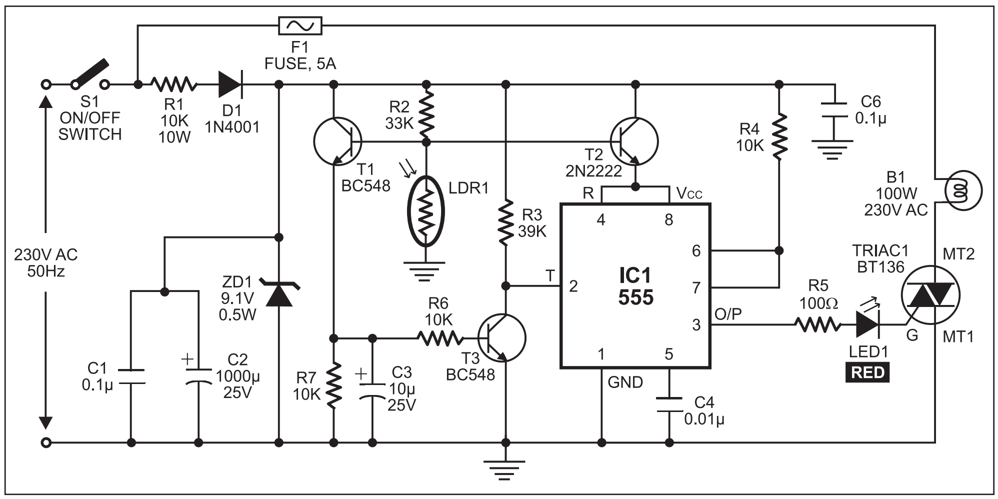
Transistor T1 is the LED driver. Resistor R4 limits the current flowing through LED1 and hence its brightness. You may connect one or two additional LEDs in series with LED1 to get more light. The low brightness of LED1 will extend the battery back-up time.

Since the circuit is directly connected to the high-voltage AC supply, enclose it in a plastic case (shown in Fig. 2) to avoid any fatal electric shock. On the front side of the cabinet, leave a hole for LDR1 so that light can easily fall on it. Fix preset VR1 on the other side. You can place the gadget anywhere you want, provided ambient light falls directly on the LDR.

ELECTRONIC STREET LIGHT SWITCH

PRINCE PHILIPS

Here's a simple and low-cost street light switch. This switch automati-cally turns on the light at sunset and turns it off at sunrise. The automatic function saves electricity besides manpower.



Broadly, the circuit can be divided into power supply and switching sections.

Pressing switch S1 connects mains to power the circuit. Mains is stepped down to 9.1V DC by resistor R1, diode D1 and zener diode ZD1. The output across ZD1 is filtered by capacitors C1 and C2. The output voltage can be increased up to 18V or decreased to 5V by changing the value of zener diode ZD1.

The switching circuit is built around light-dependent resistor LDR1, transistors T1 through T3 and timer IC1. The resistance of LDR1 remains low in daytime and high at night. Timer IC1 is designed to work as an inverter, so a low input at its pin 2 provides a high output at pin 3, and vice versa. The inverter is used to activate triac 1 and turn street bulb B1 on.

During daytime, light falls on LDR1 and transistors T1 and T2 remain cut-off to make pins 4 and 8 of IC1 low. Since transistor T3 is also cut-off, IC1 is not triggered. As a result, output pin 3 of IC1 (connected to the gate of triac 1 via resistor R5 and red LED1) remains low and the street bulb does not glow.

At night, no light falls on LDR1 and transistors T1 and T2 conduct to make pins 4 and 8 of IC1 high. Due to the conduction of transistor T3, trigger pin 2 of IC1 remains low. The high output of IC2 at its pin 3 turns the street bulb 'on.'

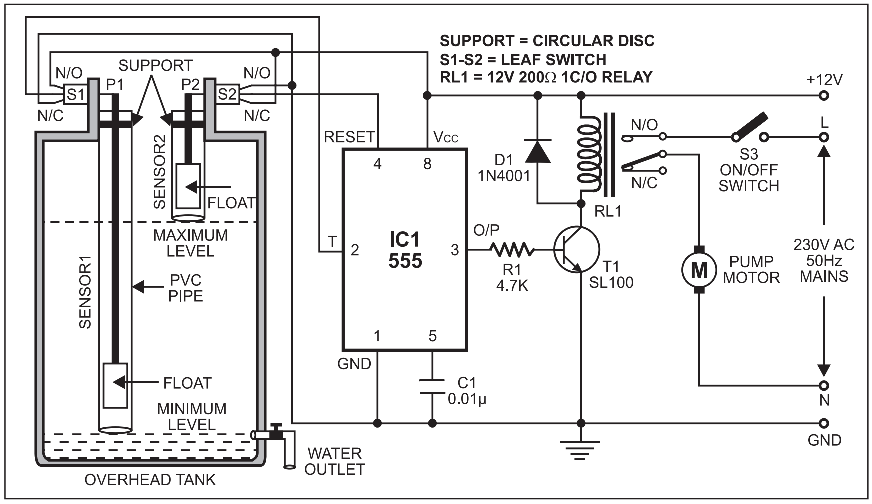
Assemble the circuit, except LDR1, on any general-purpose PCB. Use long wires for LDR1 connections so that it can be mounted at a place where sufficient light falls on it.

WATER-LEVEL CONTROLLER

K.P. VISWANATHAN

Here is a simple, automatic water-level controller for overhead tanks that switches on/off the pump motor when water in the tank goes below/above the minimum/maximum level. The water level is sensed by two floats to operate the switches for controlling the pump motor.

Each sensors float is suspended from above using an aluminium rod. This arrangement is encased in a PVC pipe and fixed vertically on the inside wall of the water tank. Such sensors are more reliable than induction-type sensors. Sensor 1 senses the minimum water level, while sensor 2 senses the maximum water level (see the figure).



Leaf switches S1 and S2 (used in tape recorders) are fixed at the top of the sensor units such that when the floats are lifted, the attached 5mm dia. (approx.) aluminium rods push the moving contacts (P1 and P2) of leaf switches S1 and S2 from normally closed (N/C) position to normally open (N/O) position. Similarly, when the water level goes down, the moving contacts revert back to their original positions.

Normally, N/C contact of switch S1 is connected to ground and N/C contact of switch S2 is connected to 12V power supply. IC 555 is wired such that when its trigger pin 2 is grounded it gets triggered, and when reset pin 4 is grounded it gets reset. Threshold pin 6 and discharge pin 7 are not used in the circuit.

When water in the tank goes below the minimum level, moving contacts (P1 and P2) of both leaf switches will be in N/C position. That means trigger pin 2 and reset pin 4 of IC1 are connected to ground and 12V, respectively. This triggers IC1 and its output goes high to energise relay RL1 through driver transistor SL100 (T1). The pump motor is switched on and it starts pumping water into the overhead tank if switch S3 is 'on.'

As the water level in the tank rises, the float of sensor 1 goes up. This shifts the moving contact of switch S1 to N/O position and trigger pin 2 of IC1 gets connected to 12V. This doesn't have any impact on IC1 and its output remains high to keep the pump motor running.

As the water level rises further to reach the maximum level, the float of sensor 2 pushes the moving contact of switch S2 to N/O position and it gets connected to ground. Now IC1 is reset and its output goes low to switch the pump off.

As water is consumed, its level in the overhead tank goes down. Accordingly, the float of sensor 2 also goes down. This causes the moving contact of switch S2 to shift back to NC position and reset pin 4 of IC1 is again connected to 12V. But IC1 doesn't get triggered because its trigger pin 2 is still clamped to 12V by switch S1. So the pump remains switched off.

When water level further goes down to reach the minimum level, the moving contact of switch S1 shifts back to N/C position to connect trigger pin 2 of IC1 to ground. This triggers IC1 and the pump is switched on.

The float sensor units can be assembled at home. Both the units are identical, except that their length is different. The depth of the water tank from top to the outlet water pipe can be taken as the length of the minimum-level sensing unit. The depth of the water tank from top to the level you want the tank to be filled up to is taken as the length of the maximum-level sensing unit. The leaf switches are fixed at the top of the tank as shown in the figure.

Each pipe is closed at both the ends by using two caps. A 5mm dia. hole is drilled at the centre of the top cap so that the aluminium rod can pass through it easily to select the contact of leaf switches. Similarly, a hole is to be drilled at the bottom cap of the pipe so that water can enter the pipe to lift the float.

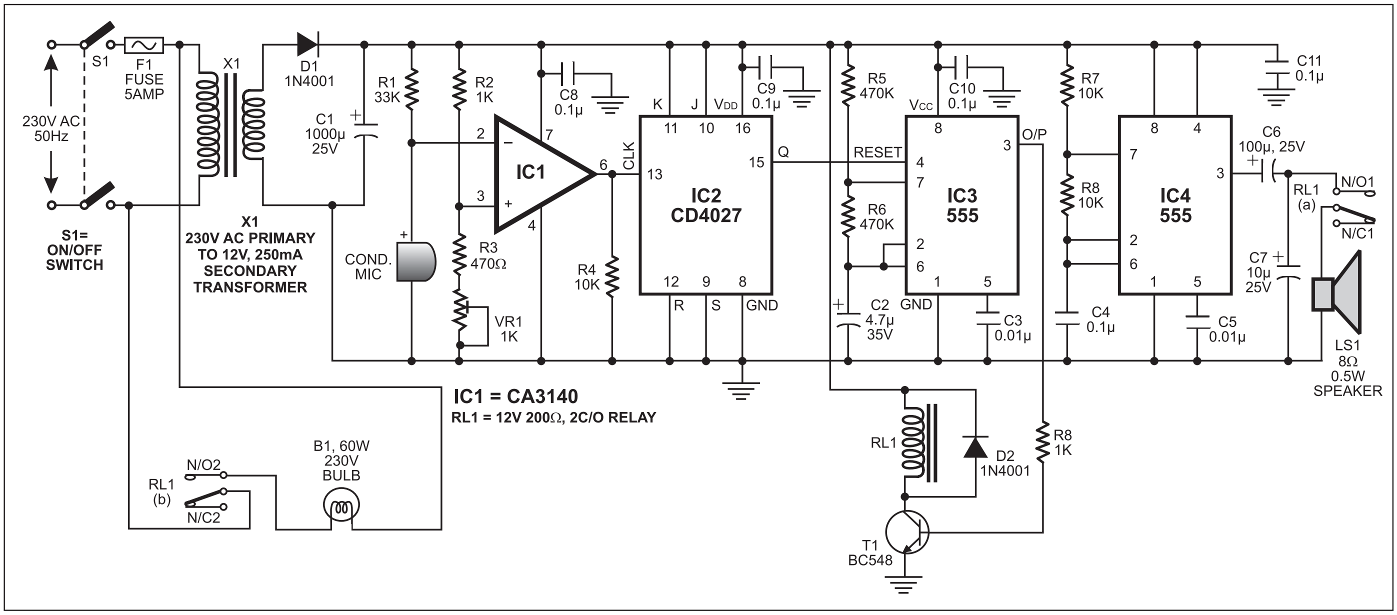
When water reaches the maximum level, the floats should not go up more than the required distance for pushing the moving contact of the leaf switch to N/O position. Otherwise, the pressure on the float may break the leaf switch itself. The length of the aluminium rod is to be selected accordingly. It should be affixed on the metal/thermocole float using some glue (such as Araldite).

SOUND-OPERATED INTRUDER ALARM

RAJ K. GORKHALI

When this burglar alarm detects any sound, such as that created by opening of a door or inserting a key into the lock, it starts flashing a light as well as sounding an intermittent audio alarm to alert you of an intruder. Both the light and the alarm are automatically turned off by the next sound pulse.

230V AC mains is stepped down by transformer X1, rectified by diode D1 and filtered by capacitor C1 to give 12V DC. The voltage at the non-inverting input (pin 3) of op-amp CA3140 (IC1) is treated as the reference voltage and it can be set using preset VR1. The voltage at the inverting input (pin 2) is the same as that across the condenser microphone. The condenser microphone should be carefully set for a high sensitivity of the sound. A high reference value means a subtle sound is enough to change the output of IC1 at pin



6. Fix the reference voltage such that the output remains unchanged during any false triggering.

In the absence of any sound, the voltage at input pin 2 of IC1 is almost equal to the full DC voltage and therefore the output of IC1 remains low. Since IC CD4027 is wired in toggle mode, its output pin 15 is also low. This makes reset pin 4 of IC3 low to reset the astable multivibrator built around timer 555 (IC3). As a result, transistor T1 is cut-off and relay RL1 remains de-energised. In de-energised state, both the N/O contacts of relay RL1, i.e. RL1(a) and RL1(b), remain open. RL1(a) contacts keep the lamp turned off, whereas RL(b) contact disconnects the output of the astable multivibrator built around IC 555 (IC4) to disable the speaker.

In the case of any noise, a current flows through the microphone and the voltage at pin 2 reduces to make the output of op-amp IC1 high. IC2 gets triggered by the pulse available at its pin 13 and its output at pin 15 goes high to enable astable multivibrator IC3. The output of IC3 goes high for three seconds and then goes low for 1.5 seconds. This repeats until pin 15 of IC2 remains high. The high output of IC3 energises the relay via driver transistor T1, while the low output de-energies the relay.

When relay RL1 is energised, relay contact RL1(a) passes on the AC power to bulb B1 and it lights up. At the same time, relay contact RL1(b) allows the output of astable multivibrator IC4 to the speaker and an audio tone is generated. The frequency of this audio tone is approximately 480 Hz. Both the flashing of the bulb and the audio tone continue as long as the output of flip-flop IC2 remains high.

Now if the circuit detects any further sound, the output of flip-flop IC2 goes low. This makes reset pin 4 of astable multivibrator IC3 low and IC3 stops oscillating. The low output of IC3 de-energises the relay to turn the bulb and the tone off.

HIT SWITCH

T.A. BABU

This versatile hit switch is the electronic equivalent of a conventional switch. It can be used to control the switching of a variety of electronic devices.

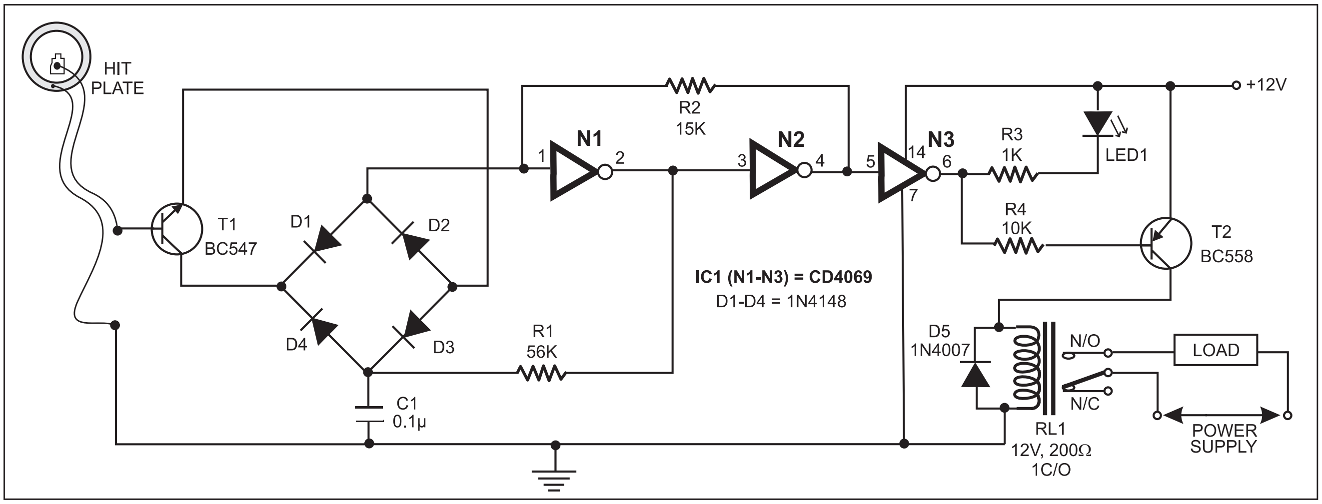
The circuit of the hit switch uses a piezoelectric diaphragm (piezobuzzer) as the hit sensor. A piezoelectric material develops electric polarisation when strained by an applied stress. The hit sensor makes use of this property.

When you hit or knock the piezo element (hit plate) with your fingertip, a small voltage developed by the piezo element is amplified by transistor BC547 (T1). The combination of transistor T1 and the bridge rectifier comprising diodes D1 through D4 acts as a voltage-control switch. The inverter gates of IC CD4069 (IC1) together with associated components form a bistable switch.

IC CD4069 is a CMOS hex inverter. Out of the six available inverter gates, only three are used here. IC1 operates at any voltage between 3V and 15V and offers a high immunity against noise. The recommended operating temperature range for this IC is –55°C to 125°C. This device is intended for all general-purpose inverter applications.

Initially, the input of gate N1 is low, while the input of gate N2 is high. Triggering the voltage-control switch by hitting the sensor pulls the input of gate N1 to high level and causes the bistable to toggle. The capacitor gets charged via resistor R1 and the circuit changes its state. This latch continues until the bistable switch gets the next triggering input.

Every time the hit plate receives a



hit, the voltage-control switch triggers the bistable circuit. That means every subsequent hit at the sensor will toggle the state of the switch. The red LED (LED1) connected at the output of gate N3 indicates 'on'/'off' position of the switch.

Relay RL1 is activated by the hit switch to control the connected load.

The circuit works off 12V DC. It can be constructed on any general-purpose PCB. For the desired results, proper connections and installation of the hit sensor are necessary. Remove the cover of the piezobuzzer and connect its two leads to the circuit. Mount the plate such that it receives the hit properly. The piezoelectric material on the plate can easily get damaged, so hit the switch gently.

CHANTING PLAYER

K.N. GHOSH

Chanting combines singing and music with *mantras*. The sweetness of chanting stills the mind, dissolves worries and opens the heart. Chanting forms an integral part of the practice schedule at *siddha yoga* retreats, centres and *ashrams*. Here are a few electronic chanting players for some popular *mantras* and *artis*.

At the heart of these circuits is a preprogrammed read-only memory (ROM) chip bonded on a hylam board. (The ROM chip is a complementary metal-oxide semiconductor (CMOS), large-scale integrated (LSI) chip.) Known as chip-on-board (COB), these boards are available in different sizes, under a blob of epoxy, with chips programmed with single or multiple *mantras*/*artis* such as *gayatri mantra*, *ganapati mantra*, *krishna mantra*, *om namah shivaye*, *shri ram jai ram* and *satnaam wahe guru*.

The COBs are available in 7-, 8-, 9- and 16-pin pad configurations. Pin connections of these COBs are shown in Fig. 6, Figs 1, 2 and 4, Figs 3 and 5, and Fig. 7, respectively. Some manufacturers make these COBs with different pad configurations, so their specifications should be strictly followed.

Besides a preprogrammed data ROM, the COBs contain an inbuilt oscillator, counter, shift register, adaptive differential pulse-code modulation (ADPCM) synthesiser and digital-to-analogue converter (DAC).

The timing pulses generated by the

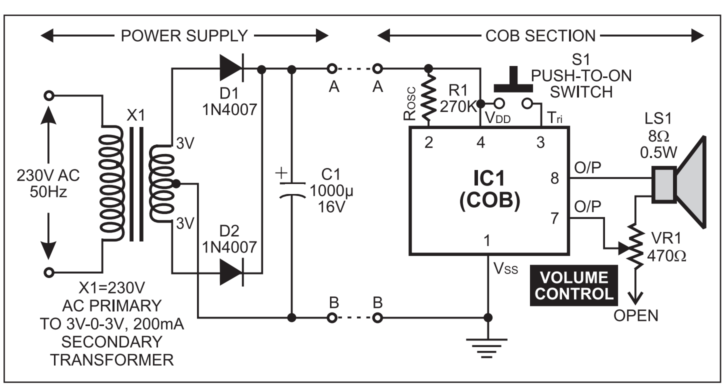


Fig. 1: The circuit for 3-in-1 mantra player including the power supply

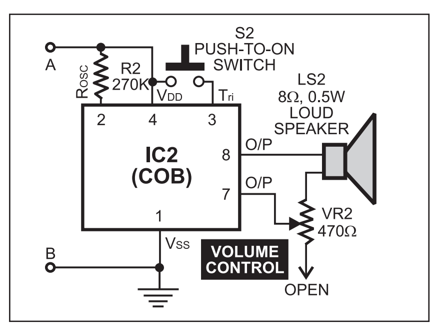


Fig. 2: The COB circuit for 2-in-1 mantra player

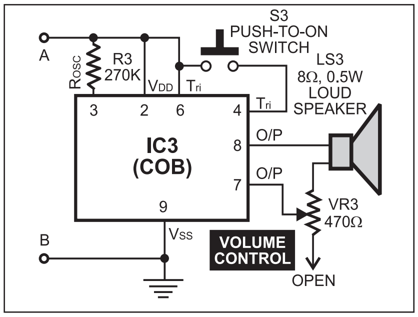


Fig. 3: The COB circuit for 6-in-1 mantra player

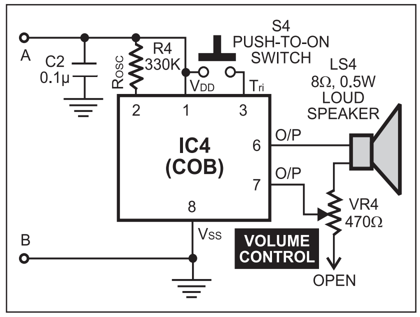


Fig. 4: The COB circuit for 5-in-1 mantra player

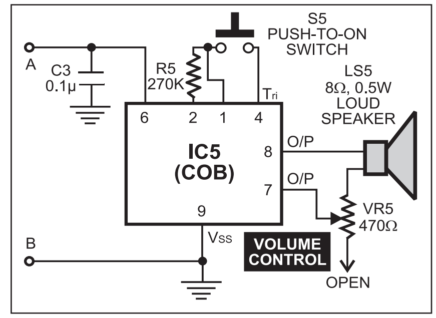


Fig. 5: The COB circuit for another 2-in-1 mantra player

oscillator regulate the pace of the *mantra* and other activity inside the chip. Its frequency is decided by an external resistor (Rosc) connected between its two input pins. The controller controls all the activities inside the chip. It sends appropriate signals to the counter and the shift register to read the data in the ROM. The output of the ROM is fed back to the controller, which directs it to the ADPCM synthesiser. The synthesiser's output is sent to the DAC, which converts it into audio. The audio output from the DAC is reproduced by the loudspeaker. The potentiometer connected to the input of the loudspeaker acts as a volume controller.

The COB works off 3V DC and is capable of driving the loudspeaker directly.

Fig. 1 can be divided into power supply and COB sections. The same power supply section is to be used for the COB circuits shown in Figs 2 through 5 as well. The 3V power supply for the COB is derived by using a 3V-0-3V center-tapped transformer (X1). The secondary output of the transformer is applied to a full-wave rectifier comprising diodes D1 and D2. The output of the full-wave rectifier is filtered by capacitor C1 to provide 3V DC to the COB.

For 3-in-1 *mantra* player, connect A and B terminals of the power supply section to the corresponding points of the COB section as shown in Fig. 1. Then connect 230V AC mains to the primary of transformer X1. Now the circuit is ready to play.

The desired *mantra* can be selected by applying positive supply to trigger pin 3 of IC1 by pressing push-to-on switch S1 momentarily. When you press switch S1 for the first time, "wahe guru" is played. When you press switch S1 second time, "satnam wahe guru" is played. When you press switch S1 third time, "satnam karta purush" is played. Using preset VR1, the volume of the sound can be controlled.

For 2-in-1 *mantra* player, connect the power supply section of Fig. 1 to the COB section shown in Fig. 2. The desired *mantra* can be selected by applying positive supply to trigger pin 3 of IC2 by pressing push-to-on switch S2 momentarily. When you press switch S2 for the first time, "jai ganesh jai ganesh deva" is played. When you press switch S2 second time, "aarti kijje hanuman lala ki" is played.

For 6-in-1 *mantra* player, connect the power supply section of Fig. 1 to the COB section shown in Fig. 3. The desired *mantra* can be selected by applying positive supply to trigger pin 4 of IC3 by pressing push-to-on switch S3 momentarily. When you press switch 3 for the first time, the circuit starts playing "om bhurbhua swaha" When you press switch S3 second time, "om namah shivaye" is played. When you press switch S3 third time, "jai ganesh, jai ganesh deva" is played. When you press switch S3 fourth time, "govind bolo hari gopal bolo" is played. When you press switch S3 fifth time, "shriman narayan narayan" is played. When you press switch S3 sixth time, "om krishna yadhamah" is played.

For 5-in-1 *mantra* player, connect the power supply section of Fig. 1 to the COB section shown in Fig. 4. When you press switch S4 for the first time, "om bhurbhua swaha" is played. On consequent pressing of switch S4, "om namo shivaye," "jai ganesh, jai ganesh deva" "jai siya ram" and "govind bolo hari gopal bolo" are played in that order.

For another 2-in-1 *mantra* player, connect the power supply section of Fig. 1 to the COB section shown in Fig. 5. When you press switch S5 for the first time, "om bhurbhua swaha" is played. When switch S5 is pressed second time, "om namah shivaye" is played.

The circuit for playing a single *mantra* with loud sound is shown in Fig. 6. The circuit comprises power supply, COB (shown within dotted lines) and low-power audio amplifier sections. Low-power audio

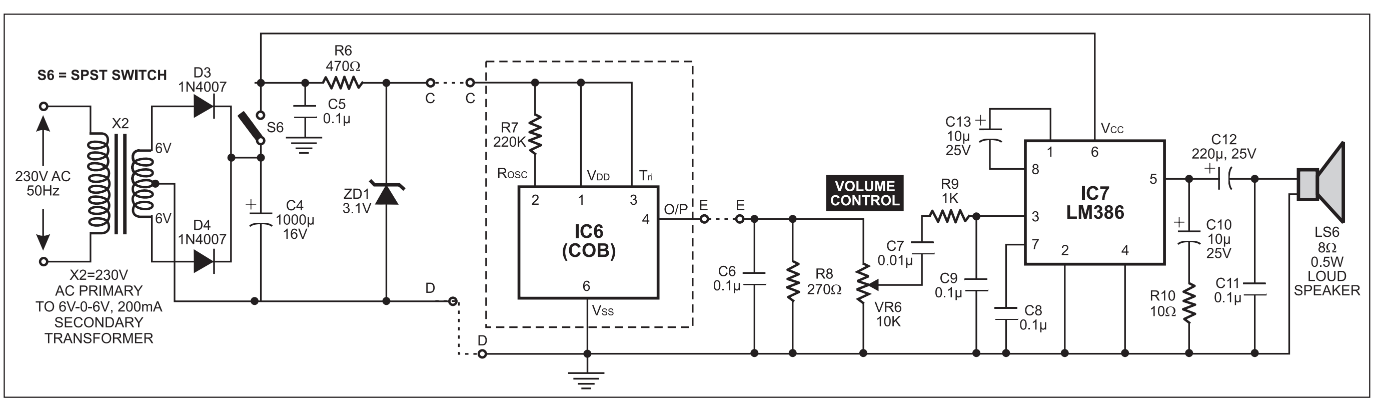


Fig. 6: The circuit (including power supply) for playing a single mantra with amplified sound

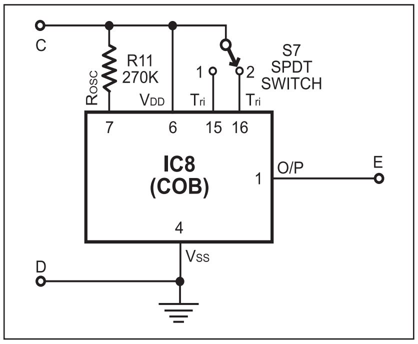


Fig. 7: The COB circuit for 2-mantra player

amplifier IC LM386 (IC6) is used here to get louder sound.

The power supply section uses a 6V-0-6V centre-tapped transformer (X2) instead of the 3V-0-3V centre-tapped transformer. The secondary output of the transformer is rectified by a full-wave rectifier comprising diodes D3 and D4, and filtered by capacitor C4 to provide 6V DC to the power amplifier (IC6). Zener diode ZD1 in series with resistor R6 reduces the supply voltage to 3V for the COB section.

Connect all the three sections together by connecting their identical terminals. Then connect 230V AC mains to the primary of transformer X2. Now the circuit is ready to work. Simply press switch S6 to provide the power supply to IC6 and IC7 and "om namah shivaye" start playing loudly. Using preset VR6, you can control the volume of the sound.

For a 2-*mantra* player with loud sound, disconnect the COB circuit shown within dotted lines in Fig. 6 and replace it with the COB circuit shown in Fig. 7. The desired *mantra* can be selected by applying positive supply to trigger pin 15 or 16 of IC8 by changing the position of switch S7. Note that switch S6 should be kept pressed.

When switch S7 is in position 1, "shri krishanah sharnam namah" is played. The *mantra* repeats continuously. To stop it, either release switch S6 or shift switch S7 to position 2. If you choose to shift switch S7, "shri krishana" stops playing but "hari krishana, hari krishana" starts playing. The *mantra* repeats continuously. To stop it, either release switch S6 or shift switch S7 to position 1.

For ease of construction, assemble a small printed circuit board (PCB) for the amplifier and power supply circuits. Various types of plastic enclosures for electronic chanting players are available in the market. Use a suitable enclosure for this player. Take care while handling and soldering the COBs as the CMOS chips can get damaged due to static charge.

Top 20 Projects (Out of 89)

- Lift Overload Preventor
- A Bi-directional Visitor Counter
- Programmer for 89C51/89C52/89C2051 Microcontroller
- Laser-based Communication Link
- Remote Controlled Sophisticated Electronic Code Lock
- PIC16F84-based Coded Device Switching System
- Voice Recording and Playback Using APR9600 Chip
- Microprocessor-based Home Security System
- Ultrasonic Lamp Brightness Controller
- Moving Message Over Dot-matrix Display
- LED-Based Message Display
- Programmable Timer for Appliances
- Antibag Snatching Alarm
- Fuse cum Power Failure Indicator
- Doorbell-controlled Porchlight
- Highway Alert Signal Lamp
- Watchman Watcher
- PC-based DC Motor Speed Controller
- Microcontroller-based Monitoring System
- Electronic Watchdog

EFY SINCERELY YOURS

EFY Enterprises Pvt Ltd
D-87/1, Okhla Industrial Area, Phase 1, New Delhi 110 020; Ph: +91-11-26810601-03
Fax: +91-11-26817563; E-mail: info@efyindia.com; Website: www.efyindia.com