

# MICROCONTROLLER-BASED ULTRASONIC DISTANCE METER

■ K. PADMANABHAN

There are several ways to measure distance without contact. One way is to use ultrasonic waves at 40 kHz for distance measurement. Ultrasonic transducers measure the amount of time taken for a pulse of sound to travel to a particular surface and return as the reflected echo.

This circuit calculates the distance based on the speed of sound at 25°C ambient temperature and shows it on a 7-segment display. Using it, you can measure distance up to 2.5 metres.

For this particular application, the required components are AT89C2051 microcontroller, two 40kHz ultrasonic transducers (one each for transmitter



Fig. 1: Ultrasonic transmitter and receiver pair

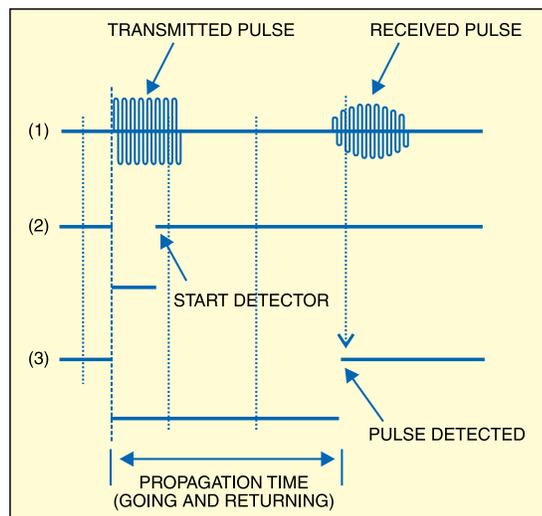


Fig. 2: The ultrasonic pulse, echo signal and time measurement

and receiver), current buffer ULN2003, operational amplifier LM324, inverter CD4049, four 7-segment displays, five transistors and some discrete components. The ultrasonic transmitter-receiver pair is shown in Fig. 1.

Ultrasonic generators use piezoelectric materials such as zinc or lead zirconium tartrate or quartz crystal. The material thickness decides the resonant frequency when mounted and excited by electrodes attached on either side of it. The medical scanners used for abdomen or heart ultrasound are designed at 2.5 MHz. In this circuit, a 40kHz transducer is used for measurement in the air medium. The velocity of sound in the air is around 330 m/s at 0°C and varies with temperature.

In this project, you excite the ultrasonic transmitter unit with a 40kHz pulse burst and expect an echo from the object whose distance you want to measure. Fig. 2 shows the transmitted burst, which lasts for a period of approximately 0.5 ms. It travels to the object in the air and the echo signal is picked up by another ultrasonic transducer unit (receiver), also a 40 kHz pre-tuned unit. The received signal, which is very weak, is amplified several times in the receiver circuit and appears somewhat as shown in Fig. 2 when seen on a CRO.

Weak echoes also occur due to the signals being directly received through the side lobes. These are ignored as the real echo received alone would give the correct distance. That is why we should have a level control. Of course, the signal gets weaker if the target is farther than 2.5 metres and will need a



higher pulse excitation voltage or a better transducer.

Here the microcontroller is used to generate 40kHz sound pulses. It reads when the echo arrives; it finds the time taken in microseconds for to-and-fro travel of sound waves. Using velocity of 333 m/s, it does the calculations and

## PARTS LIST

### Semiconductors:

IC1	- AT89C2051 microcontroller
IC2	- ULN2003 current buffer
IC3	- CD4049 hex inverting buffer
IC4	- LM324 quad operational amplifier
IC5	- 7815, 15V regulator
IC6	- 7915, -15V regulator
IC7	- 7805, 5V regulator
T1-T4	- BC557 pnp transistor
T5	- 2N2222 npn transistor
D1, D2	- 1N4148 switching diode
D3-D6	- 1N4007 rectifier diode
DIS1-DIS4	- LTS 542 common-anode, 7-segment display

### Resistors (all 1/4-watt, ±5% carbon):

R1, R2	- 2-mega-ohm
R3	- 82-kilo-ohm
R4, R7-R10	- 10-kilo-ohm
R5	- 33-kilo-ohm
R6	- 100-kilo-ohm
R11	- 1-kilo-ohm
R12-R15	- 1.2-kilo-ohm
R16	- 220-ohm
RNW1	- 10-kilo-ohm resistor network
VR1	- 1-kilo-ohm preset

### Capacitors:

C1, C2	- 3.3nF ceramic disk
C7, C10-C12	- 0.1µF ceramic disk
C3	- 2.2nF ceramic disk
C4	- 10µF, 16V electrolytic
C5, C6	- 22pF ceramic disk
C8, C9	- 1000µF, 50V electrolytic

### Miscellaneous:

X1	- 230V AC primary to 15V-0-15V, 500mA secondary transformer
X <sub>TAL</sub>	- 12MHz crystal
S1	- Push-to-on switch
S2	- On/off switch
TX1	- 40kHz ultrasonic transmitter
RX1	- 40kHz ultrasonic receiver

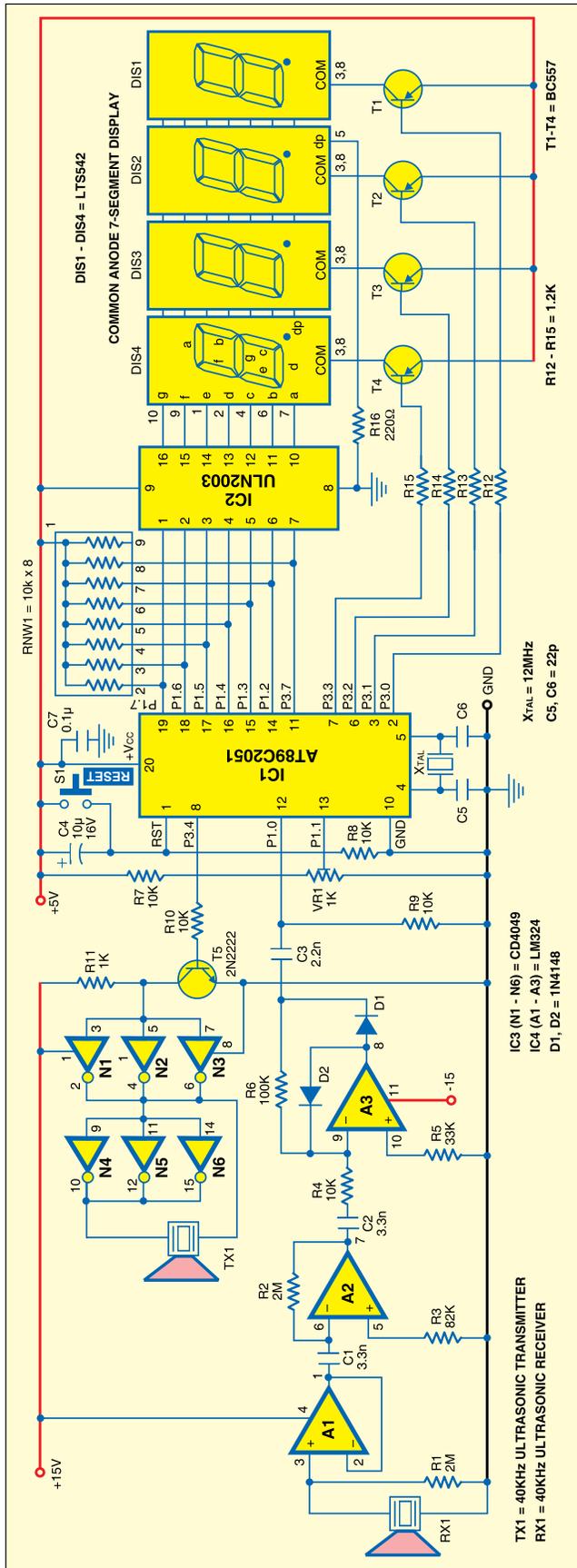


Fig. 3: Microcontroller based distance meter circuit

shows on the four 7-segment displays the distance in centimetres and millimetres (three digits for centimetres and one for millimetres).

## Circuit description

Fig. 3 shows the circuit of the microcontroller-based distance meter. The 40kHz pulse bursts from the microcontroller are amplified by transistor T5. Inverting buffer CD4049 drives the ultrasonic sensor used as the transmitter. Three inverters (N1, N2 and N3) are connected in parallel to increase the transmitted power. This inverted output is fed to another set of three inverters (N4, N5 and N6). Outputs of both sets of parallel inverters are applied as a push-pull drive to the ultrasonic transmitter.

The positive-going pulse is applied to one of the terminals of the ultrasonic sensor and the same pulse after 180-degree phase shift is applied to another terminal. Thus the transmitter power is increased for increasing the range.

If you want to increase the range up to 5 metres, use a ferrite-core step-

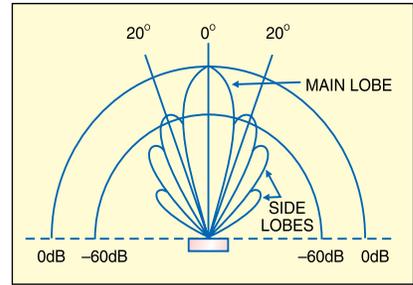


Fig. 4: Two dimensional beam pattern of ultrasonic signal showing main lobe and side lobe energy levels

up pulse transformer, which steps-up the transmitter output to 60V (peak-to-peak).

The echo signal received by the receiver sensor after reflection is very weak. It is amplified by quad operational amplifier LM324. The first stage (A1) is a buffer with unity gain. The received signal is directly fed to the non-inverting input (pin 3) of A1 and coupled to the second stage by a 3.3nF (small-value) capacitor. If you use the ubiquitous 0.01µF capacitor for coupling, there will be enormous hum at the output. The second stage of the inverting amplifier uses a 2-mega-ohm resistor for feedback. The third stage is a precision rectifier amplifier with a gain of 10.

The rectifier functions, unlike a simple diode, even for signal voltage of less than 0.6V. The output is filtered to accept 40kHz frequencies and fed to pin 12 of microcontroller AT89C2051, which is an analogue comparator. Pin 13 is the other pin of the comparator used for level adjustment using preset VR1.

The ultrasonic transducer outputs a beam of sound waves, which has more energy on the main lobe and less energy (60 dB below the main lobe) on the side lobes as shown in Fig. 4. Even this low side-lobe signal is directly picked up by the receiver unit. So you have to space the transmitter and receiver units about 5 cm apart. The two units are fixed by cellotape onto a cardboard, with the analogue circuit at one end.

Microcontroller AT89C2051 is at the heart of the circuit. Port-1 pins P1.7 through P1.2, and port-3 pin P3.7 are connected to input pins 1 through 7 of

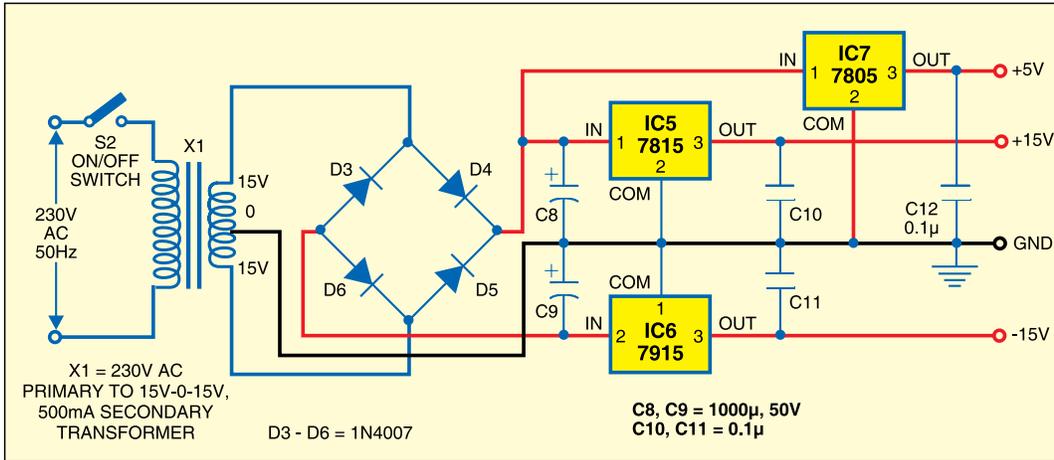


Fig. 5: Power supply circuit for distance meter

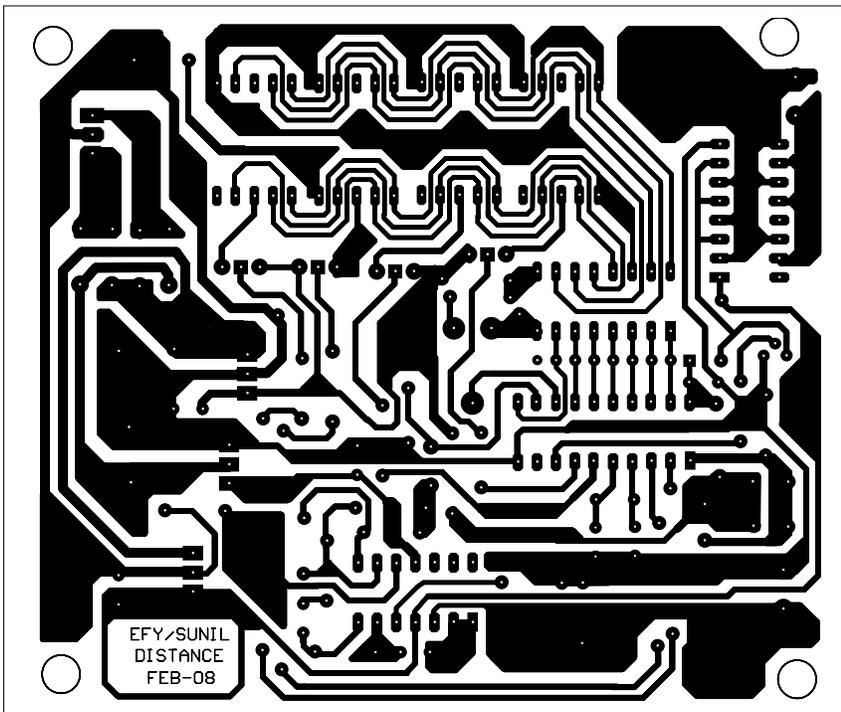


Fig. 6: Actual-size, single-side PCB for the microcontroller-based ultrasonic distance meter

IC2 (IC ULN2003), respectively. These pins are pulled up with a 10-kilo-ohm resistor network RNW1. They drive all the segments of the 7-segment display with the help of inverting buffer IC2.

Port-3 pins P3.0 through P3.3 of the microcontroller are connected to the base of transistors T1 through T4 to provide the supply to displays DIS1 through DIS4, respectively. Pin P3.0 of microcontroller IC1 goes low to drive transistor T1 into saturation, which provides supply to the common-anode pin (either pin 3 or 8) of

display DIS1. Similarly, transistors T2 through T4 provide anode currents to the other three 7-segment displays.

Microcontroller IC1 provides the segment data and display-enable signal simultaneously in time-division multiplexed mode for displaying a particular number on the 7-segment display unit.

Segment data and display-enable pulse for the display are refreshed every 5 ms. Thus the display appears to be continuous, even though the individual LEDs used in it light up one by one.

Using switch S1 you can manually reset the microcontroller, while the power-on reset signal for the microcontroller is derived from the combination of capacitor C4 and resistor R8. A 12MHz crystal is used to generate the basic clock frequency for the microcontroller. Resistor R16 connected to pin 5 of DIS2 enables the decimal point.

The comparator is inbuilt in microcontroller AT89C2051. The echo signal will make port-3 pin 3.6 low when it goes above the level of voltage set on pin 13. This status is sensed by the microcontroller as programmed.

When port-3 pin P3.6 goes high, we know that the echo signal has arrived; the timer is read and the 16-bit number is divided by twice the velocity of sound and then converted into decimal format as a 4-digit number.

**Power supply.** Fig. 5 shows the circuit of the power supply. The 230V AC mains is stepped down by transformer X1 to deliver the secondary output of 15V-0-15V, 500 mA. The transformer output is rectified by a full-wave bridge rectifier comprising diodes D3 through D6, filtered by capacitors C8 and C9 and then regulated by ICs 7815 (IC5), 7915 (IC6) and 7805 (IC7). Regulators 7815, 7915 and 7805 provide +15V, -15V and +5V regulated supply, respectively. Capacitors C10 through C12 bypass the ripples present in the regulated power supply.

## Construction and testing

An actual-size, single-side PCB for the microcontroller-based distance meter is shown in Fig. 6 and its component layout in Fig. 7.

Assemble the PCB and put the programmed microcontroller into the socket. After switching on the power supply and microcontroller automatically getting reset upon power-'on,'

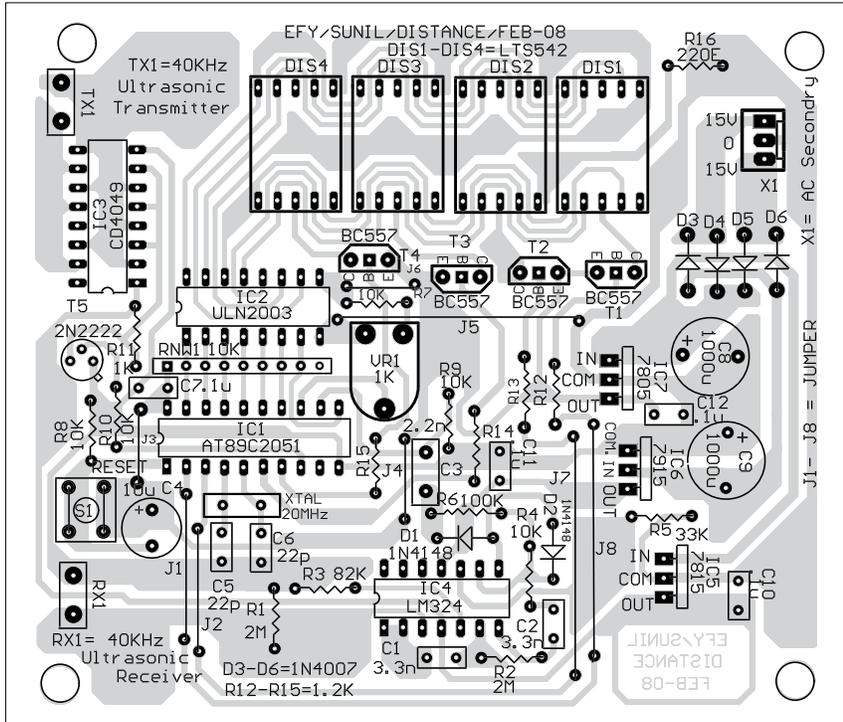


Fig. 7: Component layout for the PCB

pin 8 will pulse at 40kHz bursts. This can be seen using an oscilloscope. Give this signal to channel 1 of the oscilloscope. Adjust the time base to 2 ms per division and set it to trigger mode instead of normal mode. Adjust the potmeter on the oscilloscope labeled 'level' such that the trace starts with the burst and appears steady as shown in Fig. 2.

Connect the transmitter and receiver ultrasonic units either by a twisted pair of wire or by a shielded cable to the board. Give the received signal to channel 2 of the oscilloscope. Then, place an A4-size plastic sheet in front of the ultrasonic transducers and

observe the echo signal. It will appear as shown in Fig. 2.

The two transducers can be fixed to a thick cardboard with two wires leading to the circuit—two 40cm long shielded cables will do. The laser pointer is fixed such that it is axial to the transducers. Channel 2 is connected to pin 12, which is the positive non-inverting terminal of AT89C2051's comparator. The negative inverting terminal (pin 13) is connected to a preset reference. Adjust the preset such that the voltage is 0.1V-0.2V at pin 13. This will enable detection of weak echoes also.

When the echo signal goes above

the level of reference voltage set on pin 13, it will make P3.6 low; the arrival of echo is sensed by the program using jnb p3.6 (jump not bit) instruction.

## Software

The software is written in Assembly language and assembled using 8051 cross-assembler. It is well commented and easy to understand.

The pulse train for 0.5 ms is started by making pin 8 high and low alternately for 12.5 microseconds so that the pulse frequency is 40 kHz. After 25 such pulses have passed, a waiting time is given to avoid direct echoes for about 20  $\mu$ s. Then the signal is awaited, while the timer runs counting time in microseconds. When the echo arrives, port-3 pin P3.6 goes high, the timer reads and the 16-bit number is divided by twice the velocity and converted into decimal format as a 4-digit number.

If the echo does not arrive even after 48 milliseconds, the waiting loop is broken and the pulse train sequence is started once again. If the echo comes within this time, it is displayed for half a second before proceeding to another measurement. Thus, the display appears continuous and flicker-free.

## Other uses

Simply by changing this program, the same unit can be made to detect moving objects (such as cars racing on the street) and find their range and speed. It can also be used with suitable additional software as a burglar alarm unit for homes or offices.

**EFY note.** All the relevant files of this project are included in this month's EFY-CD.

## ULTRSD.ASM

Program listing:

```
$mod51
ORG 0H
AJMP 30H
ORG 0BH ;TIMER 0 INTERRUPT VECTOR
; AJMP TIMER0ISR ;Timer 0 Interrupt
service routine address
ORG 30H
MOV SP,#60H ;set stack pointer
MOV P3,#0FFH ;set all port 3 bits high
to enable inputs also
MOV P1,#03 ;set port 1 to all zeros
expect bits 0,1
```

```
MOV TMOD,#01100001B ;TIMER 1 - MODE 2
COUNTER,TIMR-0 TO MODE 1
BEG: MOV TH0,#0H ;TIMER REG.0 IS SET TO
0, GIVES 64ms
MOV TLO,#0 ; timer low reg. is also so
;TOTAL CYCLE TIME IS 64.6ms ,350m/s
gives 0.35mx65=22.5m
; up and down 10 metres say! .35 m/ms,
.35 mm/us, 1mm per 3 micros
; up and down .35/2 mm/us = 1/6 mm/us
; VELOCITY OF SOUND IN AIR IS 350 M/S
; AFTER 100 TIMES, WE HAVE TO STOP
TRANSMITTING FOR A TIME OF ABOUT .1 S
```

```
; SO WE STOP FOR THIS AMOUNT OF TIME
and expect an echo.
mov r2,#25 ; 25 pulses 26 us =.53 ms
(343m/s*.5ms=17cm)
pulse: setb p3.4 ;generates 40KHz
mov r1,#5
djnz r1,$
clr p3.4
mov r1,#5
djnz r1,$ ;wait for 13 us
djnz r2, pulse ;20pulses
setb tr0 ;start timer
```

```

mov r2,#10
djnz r2,$;wait 20 us
check_echo:
jnb p3.6,checktimeout
MOV 40h,TL0 ; read timer count
MOV 41h,TH0
mov r0,40h
mov r1,41h
mov r3,#0
mov r2,#6
call UDIV16 ;divide by 6
mov 40h,r0
mov 41h,r1
mov 50h,#25
disp: call displ ; show the value on
LED
    djnz 50h,disp ; so many times for a
visible time limit

    jmp beg

checktimeout: mov a,th0
    cjne a,#0c0h,check_echo ;upto 4 metres
    jmp beg
;subroutine UDIV16
;16 bit/16bit unsigned divide
;input r1,r0 =dividend X
;input r3,r2 =divisor Y
;output r1,r0 =quotient q of x/y
;output r3,r2 = remainder
; alters acc,r4-47,flags,dptr
UDIV16: mov r7,#0 ;clear partial re-
mainder
    mov r6,#0 ;
    mov B,#16 ;set loop count
div_loop: clr C ;clear carry flag
    mov a,r0 ; shift the highest bit of
dividend into
    rlc a
    mov r0,a
    mov a,r1
    rlc a
    mov r1,a
    mov a,r6 ;... the lowest bit of partial
remainder
    rlc a
    mov r6,a
    mov a,r7
    rlc a
    mov r7,a
    mov a,r6
    clr C
    subb a,r2
    mov dpl,a
    mov a,r7
    subb a,r3
    mov dph,a
    cpl C
    jnc div_1 ;update partial reaminder
if borrow
    mov r7,dph
    mov r6,dpl ; update parital reminder
div_1: mov a,r4
    rlc a
    mov r4,a
    mov a,r5
    rlc a
    mov r5,a
    djnz B,div_loop
    mov a,r5
    mov r1,a ; put qt. in r0,r1
    mov a,r4

```

```

mov r0,a
mov a,r7 ;get rem. saved before the
mov r3,a ;last subtraction.
mov a,r6
mov r2,a
ret

;16 Bit Hex to BCD Conversion for 8051
Microcontroller
; This routine is for 16 bit Hex to BCD
conversion;
;Accepts a 16 bit binary number in
R1,R2 and returns 5 digit BCD in
;R7,R6,R5,R4,R3(upto 64K )
Hex2BCD: ;r1=high byte ;r7 most sig-
nificant digit
;R2 = LSByte
MOV R3,#00D
MOV R4,#00D
MOV R5,#00D
MOV R6,#00D
MOV R7,#00D
MOV B,#10D
MOV A,R2
DIV AB
MOV R3,B ;
MOV B,#10 ; R7,R6,R5,R4,R3
DIV AB
MOV R4,B
MOV R5,A
CJNE R1,#0H,HIGH_BYTE ; CHECK FOR HIGH
BYTE
    SJMP ENDD
HIGH_BYTE: MOV A,#6
    ADD A,R3
    MOV B,#10
    DIV AB
    MOV R3,B
    ADD A,#5
    ADD A,R4
    MOV B,#10
    DIV AB
    MOV R4,B
    ADD A,#2
    ADD A,R5
    MOV B,#10
    DIV AB
    MOV R5,B
    CJNE R6,#00D,ADD_IT
    SJMP CONTINUE
ADD_IT: ADD A,R6
CONTINUE: MOV R6,A
    DJNZ R1,HIGH_BYTE
    MOV B, #10D
    MOV A,R6
    DIV AB
    MOV R6,B
    MOV R7,A
ENDD: ret

DISP1:
REFRESH: ; content of 18 to 1B memory
locations are output on LEDs
; only numbers 0 to 9 and A to F are
valid data in these locations
    mov r1,41h
    mov r2,40h

    CALL HEX2BCD
    MOV 18H,r3 ; least significant digit
    MOV 19H,r4 ; next significant digit
    MOV 1AH,r5

```

```

MOV 1BH,R6 ; most significant digit
(max:9999)

refresh1: MOV R0,#1bh ; 1b,1a,19,18,
holds values for 4 digits
    MOV R4,#8 ; pin p3.3_0 made low one by
one starts wth 18
    mov r7,#2 ; decimal pt.on 3rd digit
from left (2 nd fromright)
PQ2: CALL SEGDISP
deC R0
mov a,r4
rrc a
mov r4,a
jnc pq2
PV3:
RET

SEGDISP:
mov dptr,#ledcode
MOV A,@R0
ANL A,#0FH
MOVC A,@A+dptr

segcode:
MOV R5,A
ORL A,#03H ; WE WANT TO USE PORT 1 BITS
0 AND 1 FOR INPUT ANLOG
; so retain them high
S3: MOV P1,A ; SEGMENT_PORT

    MOV A,R5 ;we use p3.7 for the segment
'a' of display
    RRC A ;so get that bit D0into carry
; cpl c
; mov p3.5,c ; dec pt is D0 bit that is
wired to p3.5
    rrc a
    mov p3.7,c ;segment 'a';
    S1: MOV A,R4 ; get digit code from r4
00001000
    cpl a ;11110111
    rrc a ;11111011-1

    mov p3.0,c ; output to drive transsi-
tors for digit lighting
    rrc a ;11111101-1
    mov p3.1,c
    rrc a ;11111110-1
    mov p3.2,c
    rrc a ;11111111-0 yes low makes leftmost
digit show msdigit
    mov p3.3,c
S5:
    S4: ACALL DELAY1 ; let it burn
for some time
    MOV A,#0ffH ; extinguish
the digit after that time
    MOV P3,A ; to prevent shadow
    s6: RET
ledcode:
DB 7EH,0CH,0B6H,9EH,0CCH,0DAH,0FAH
DB 0EH,0FEH,0CEH,0EEH,0F8H,72H,0BCH,0
F6H,0E2H
;these are code for
the numbers 0 to 9 and A to F
    DELAY1: MOV R1,#0ffH
    N: NOP

    DJNZ R1,N
RET
END

```