B. Sc. III: Semester- V Paper- DSE 1005E1 Linear Integrated Circuits, 8051 Microcontroller Interfacing and Embedded C

Section II: 8051 Microcontroller Interfacing and Embedded C

UNIT 2 :Real World Interfacing of 8051

Presented By:
Dr. C. B. Patil
Vivekanand College (Autonomous), Kolhapur

UNIT 2 :Real World Interfacing of 8051

Syllabus: Interfacing to output devices – LED, Relay, LCD, seven segment display, seven segment display (multiplexing mode), DC Motor, Stepper Motor. Interfacing to input devices – Switch, 4X4 matrix keyboard, opto-coupler, thumb wheel switch. Interfacing to DAC0808 and ADC0804.

INTERFACING OF LED WITH 8051

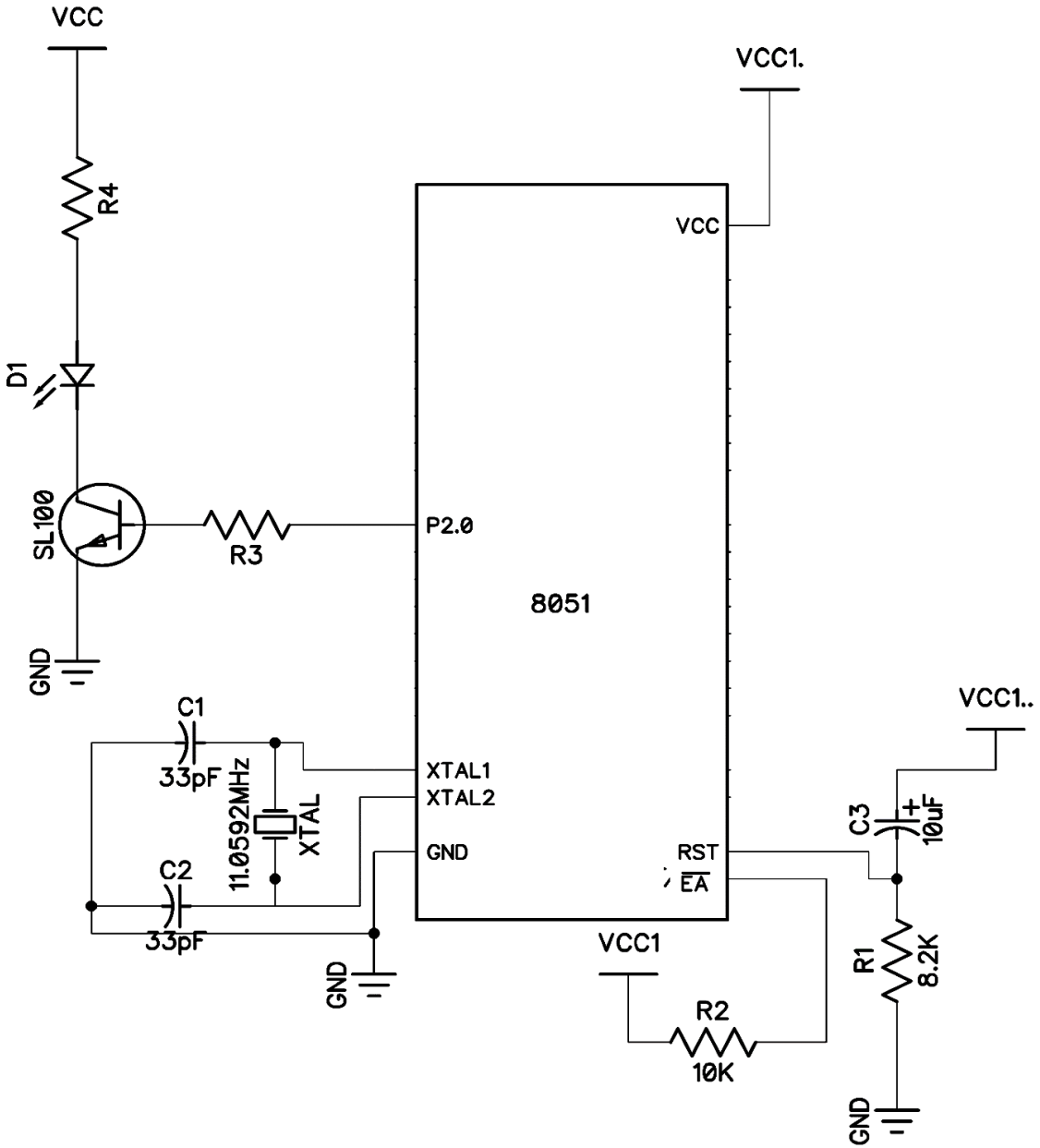
Commonly used LEDs has generally barrier potential of 1.5V and current of 10mA. If this voltage and current applied to the LED, it glows with full intensity.

Circuit Description: The power on reset circuit with R1_C3 is connected to RESET pin and for generating clock the crystal and capacitors C1 and C2 of 33pf are connected between XTAL1 and XTAL2 pin of microcontroller.

We cannot connect any pin of the 8051 to the LED directly because required current for LED is more than sinking/sourcing capacity of the 8051 and it is harmful. Therefore transistor (SL100) is used as buffer. It's base terminal is connected to port pin P2.0 through 47K resistor. This resistor limits the base current. The LED is connected in between Vcc and collector of transistor through resistor R4.This resistor limits current through LED (10 mA).

When we make Pin P2.0 high (by giving instruction the SETB P2.0) transistor will become ON, then current flows through LED-collector-emitter of transistor and hence LED turns ON .To make LED off we make pin P2.0 low by using instruction CLR P2.0.

INTERFACING OF LED WITH 8051



```
START :  SETB P2.0 ;        Make port pin P2.0 High (LED ON)
         ACALL DELAY ;      LED ON for some time
         CLR P2.0 ;         Make port pin P2.0 Low (LED OFF)
         ACALL DELAY ;      LED ON for some time
         SJMP START ;       Continues LED ON-OFF
;-------------------------------------------------------------------------------------------
DELAY:   MOV TMOD,# 01 H ;          Timer 0 mode 1 selection
         MOV TH0,# 00H;             Load Higher byte of timer 0 with initial count 00H
         MOV TL0,# 00H ;            Load Higher byte of timer 0 with initial count 00H
         SETB TR0 ;                 Start the timer 0
         AGAIN: JNB TF0,AGAIN ;     Monitor TF0 flag to check count finish or not
         CLR TR0 ;                  Stop the timer
         CLR TF0 ;                  Clear the overflow Flag
         RET ;                      Return from subroutine.
```

*Time delay = 65536 x 1.085 µS(1 machine cycle period for crystal freq.11.0592 MHz)= 71106 µS or 71.106 ms.

The value of resistor R4

R4= $(Vcc - VD - VCE)/IC$

Where, Vcc; Supply Voltage (+5V)

VD: LED barrier potential(+1.5V)

VCE: Collector to emitter (when transistor becomes on here 0.6V)

IC: Collector current(here LED Current=10mA)

R4 =330 Ohm

R3= $(VP2.0 - VBE)/IB$= $(VP2.0 - VBE)\ \beta\ /\ IC$

Where VP2.0=Maximum voltage at pin P2.0(Here it is +5V)

VBE=Base to Emitter Voltage when transistor ON(here 0.6V)

IB=base Current

IC: Collector current(here LED Current=10mA)

β= Current gain of transistor(Here 100)

R3=47K

The R1=8.2 K & C3=10 µF forms power on Reset.

The Crystal X= 11.0592 MHz and C1=C2=33 µF used as clock circuit of 8051

Interfacing of switch with 8051:

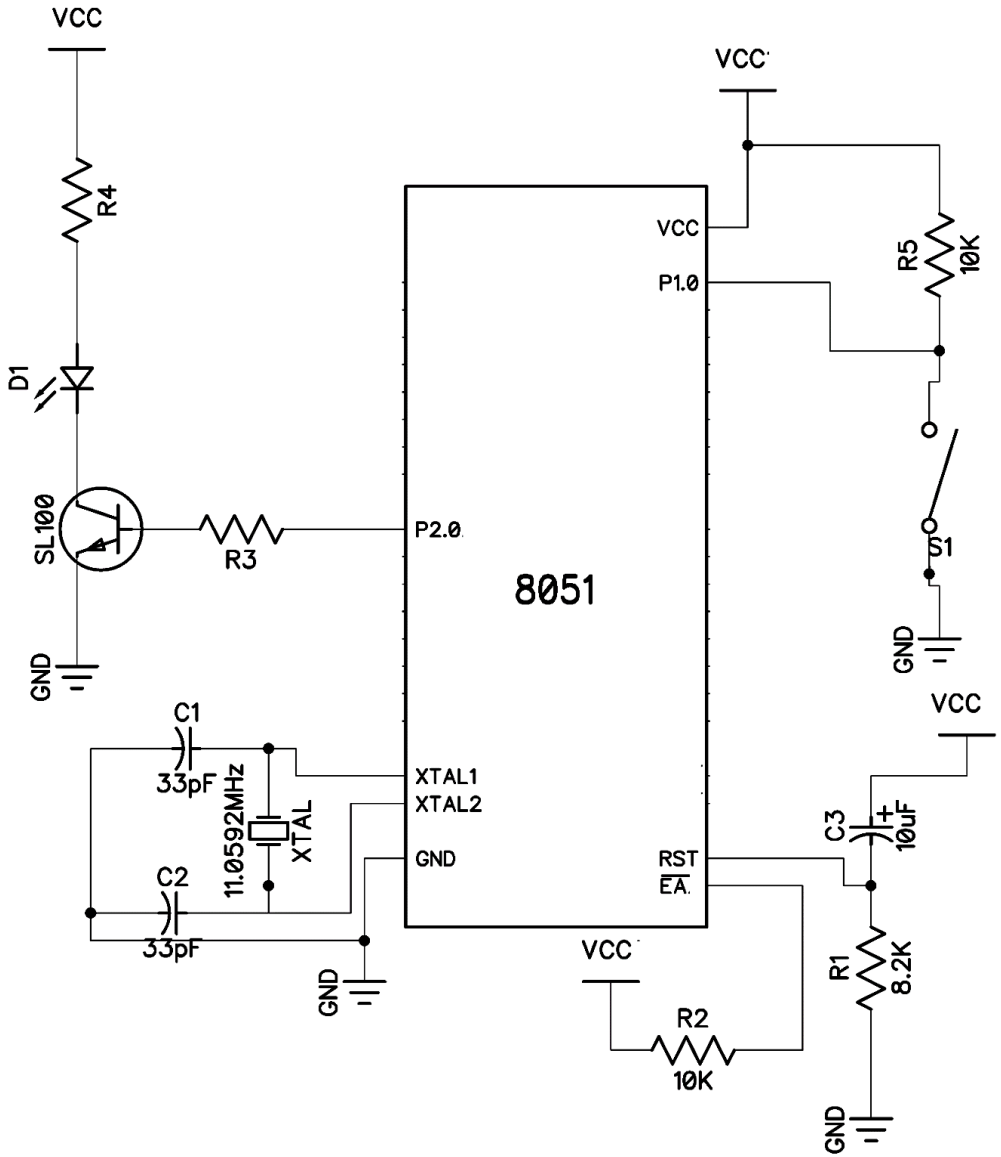
We can connect a number of switches at ports of the 8051. A switch is connected at the port pin P1.0 and we read status of switch is displayed on LED connected at P2.0

Circuit Description:

A Switch is connected at port pin P1.0 and it is normally open. One terminal of the switch is connected to +Vcc through pull up resister and one terminal is grounded. When switch is pressed port pin becomes low and when switch is open the port pin becomes high.

The statues of the switch is read by using MOV A,P1 and it display on LED. In this case port P1 must configured as an input port.LED is connected at pin P2.0 through buffer transistor. The data from the accumulator is transferred to port 2 by giving instruction MOV P2,A after complementing.

Interfacing of LED and switch with 8051



ALP:

```
START: MOV A,# 0FFH ;
       MOV P1, A;          Make all pins of port P1 high so
                           that port P1 is input port
AGAIN: MOV A,P1 ;          Read the statues of the switch and
                           transfer it to Accumulator
       CPL A ;             Compliment for switch closed LED
                           will be ON
       MOV P2,A ;          Display statues of the switch on LED
       SJMP AGAIN ;        Continuous
```

Interfacing of stepper motor to 8051

A Stepper Motor is a brushless, synchronous motor which divides a full rotation into a number of steps. DC motor which rotates continuously when a fixed DC voltage is applied to it, while a step motor rotates in discrete step angles. The number of steps required to complete one complete rotations known as steps per revolution. If stepper motor has 12, 24, 72, 144, 180 and 200 resulting stepping angles are 30, 15, 5, 2.5, 2, and 1.8 degrees per step (step angle= 360/steps).

Working (Stepper Motor)

Stepper motors consist of a permanent magnetic rotating shaft, called the rotor and electromagnets on the stationary portion that surrounds the motor, called the stator. Fig.3.12 illustrates one step rotation of a stepper motor. At position 1, we can see that the rotor is beginning at the upper electromagnet, which is currently active (has voltage applied to it). To move the rotor clockwise (CW), the upper electromagnet is deactivated and the right electromagnet is activated, causing the rotor to move 90 degrees CW, aligning itself with the active magnet. This process is repeated in the same manner step by step upto starting position. In this example step angle is 90 degree and it will requires 4 steps to complete one rotation. This is full stepping method.

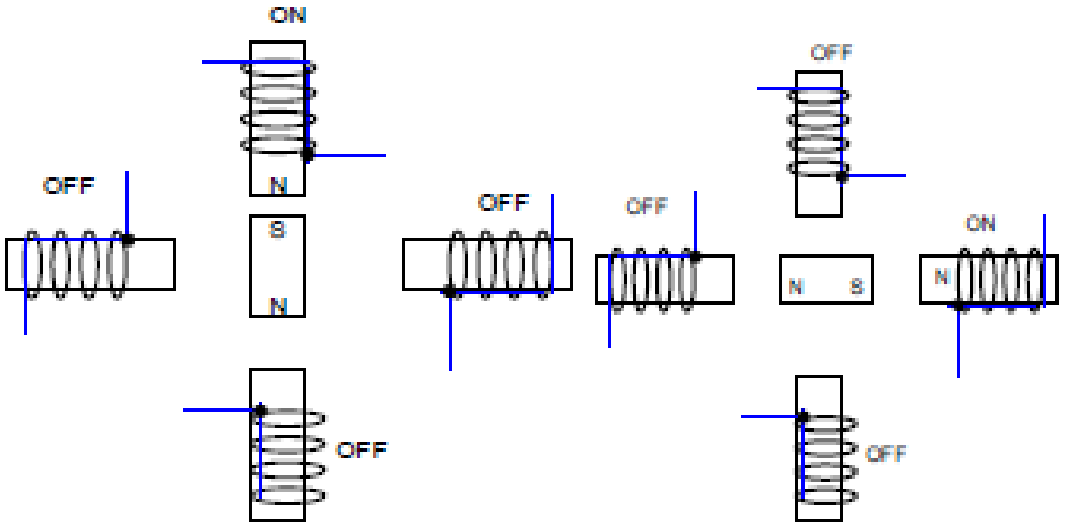


Fig 3.12 Full Stepping Method

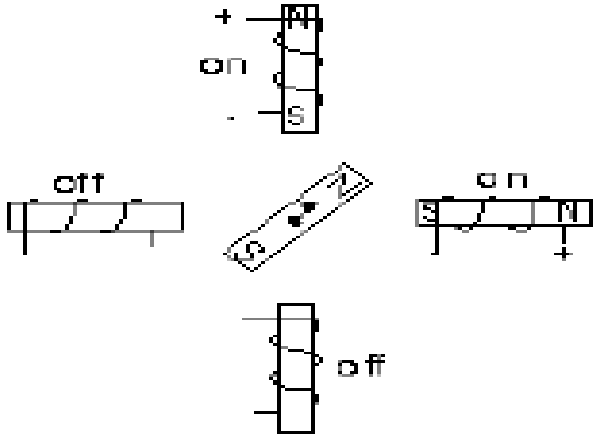


Fig. 3.13 Half stepping

Step	Winding A	Winding B	Winding C	Winding D
1	1	0	0	1
2	1	1	0	0
3	0	1	1	0
4	0	0	1	1

Table 3.1 Stepper Motor step sequence

ALP for Motor clockwise rotations

```
        MOV A,# 66 H ;Load bit pattern in accumulator
BACK:   MOV P1,A ;Transfer bit pattern to port P1 (only P1.0-P1.3 s are used)
        RR A ;Rotate accumulator right (Motor rotates in clock wise by one step )
        ACALL DELAY;delay (speed of motor depends on delay)
        SJMP BACK ;continues rotation
```

**To rotate motor anticlockwise instruction please use RL A instead of RR A*

```
;-------------------------------------------------------------------------------------------
DELAY:  MOV TMOD,# 01 H ;Selection of Timer 0 in mode 1
        MOV TH0,# 0EEH ;Load Higher byte of timer 0 with initial count EEH
        MOV TL0,# 00H ;Load Higher byte of timer 0 with initial count 00H
        SETB TR0 ;Start the timer 0
        AGAIN: JNB TF0,AGAIN ;Monitor TF0 flag to check count finish or not
        CLR TR0 ;Stop the timer
        CLR TF0 ;Clear the overflow Flag
        RET ;Return from subroutine.
```

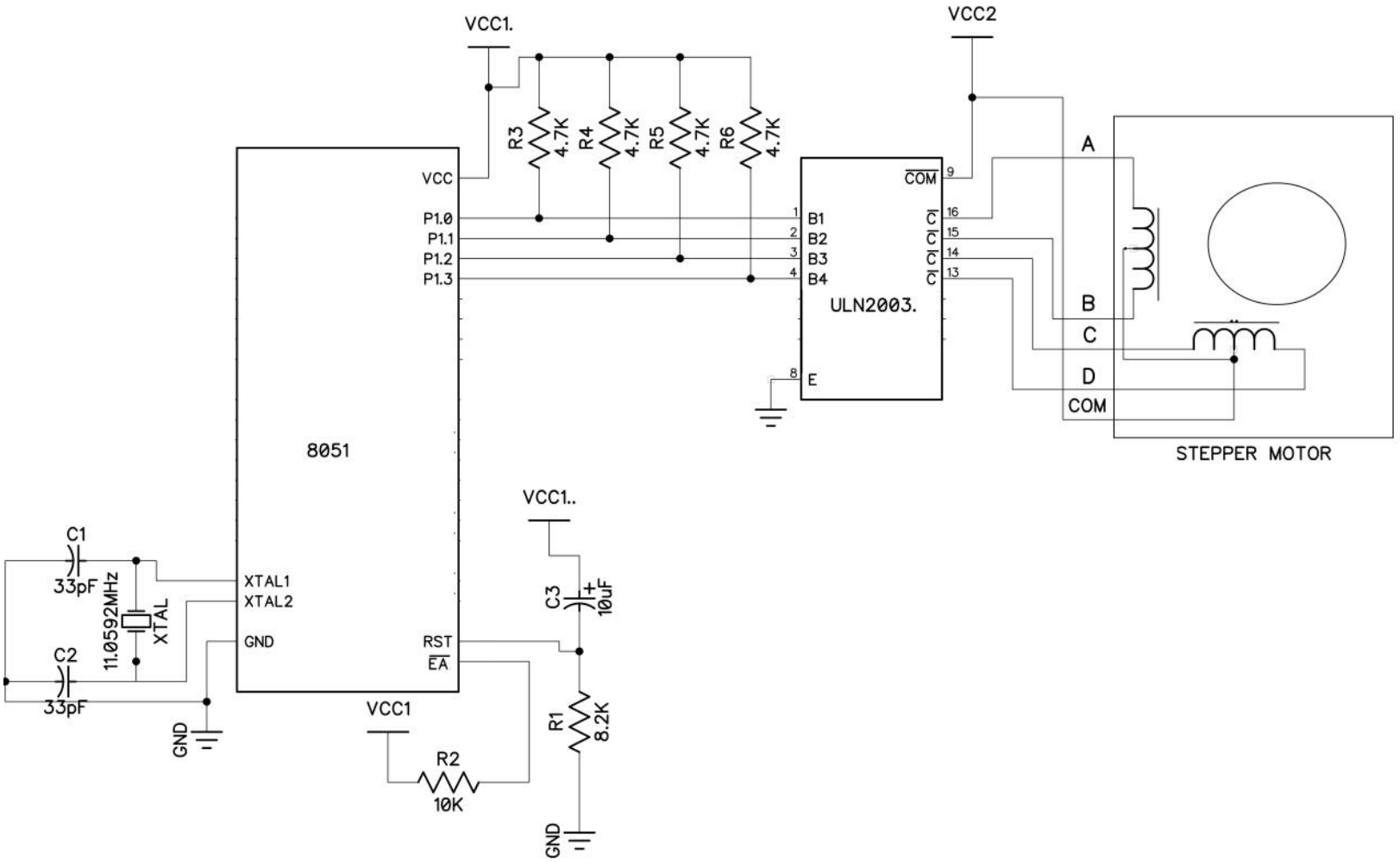
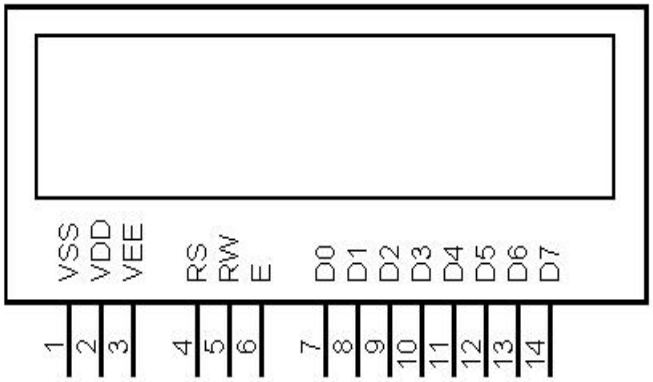


Fig. Circuit diagram to interface stepper motor with 8051

Interfacing of LCD with 8051



Pin No:	Name	Function
1	VSS	This pin must be connected to the ground
2	VCC	Positive supply voltage pin (5V DC)
3	VEE	Contrast adjustment
4	RS	Register selection RS=1 selects data register RS=0 Selects Command Register
5	R/W	Read or write
		To read from reg R/W=1 to write on reg. R/W=0
6	E	Enable High to low pulse enables command or data reg.
7-14	D0-D7	LCD Data Lines
15	LED+	Back light LED+
16	LED-	Back light LED-

Table 3.2 LCD Pins

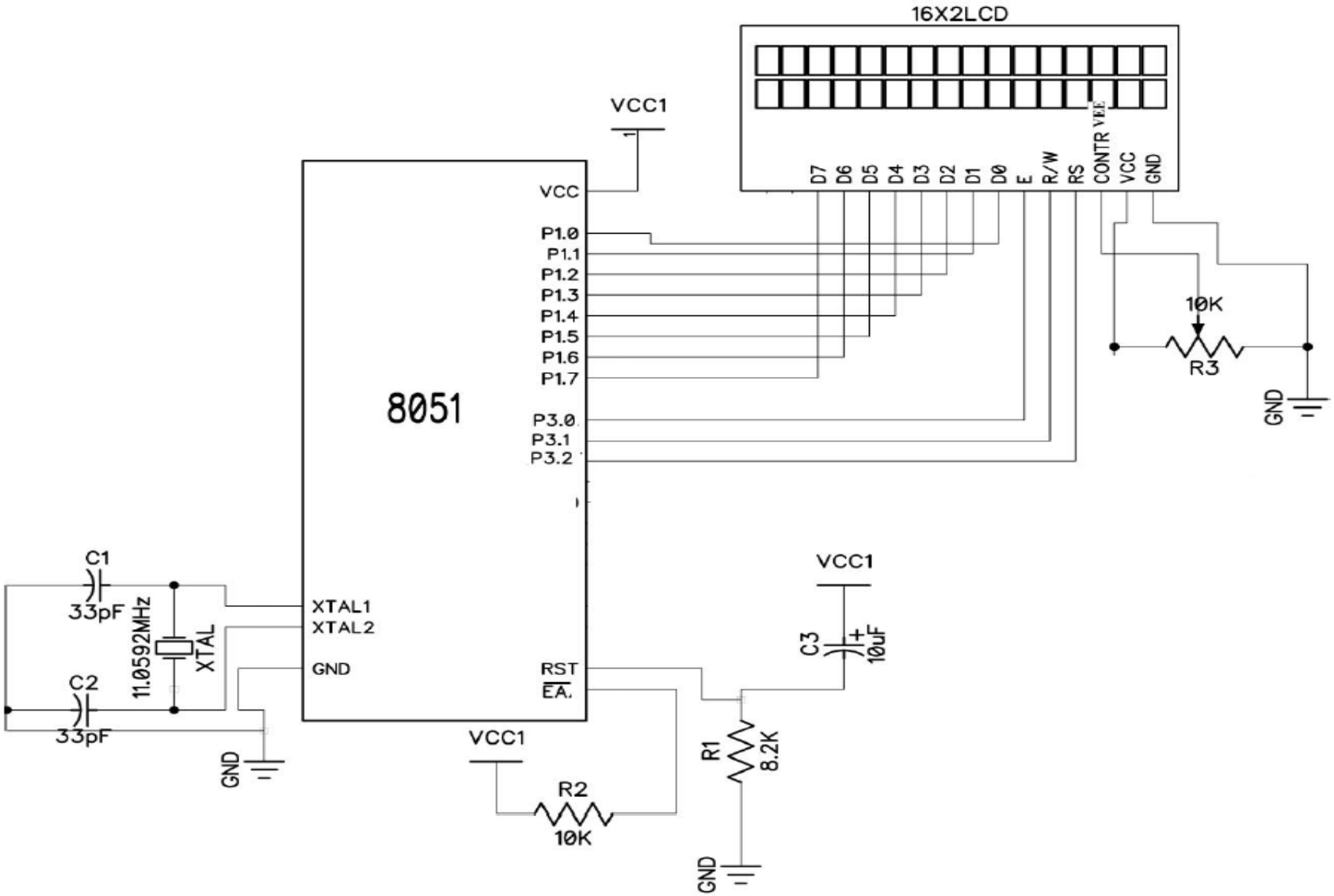


Fig 3.16. LCD interfacing with 8051

Command	Function
0FH	LCD ON, Cursor ON, Cursor blinking ON
01H	Clear screen
02H	Return home
04H	Decrement cursor
06H	Increment cursor
0EH	Display ON ,Cursor blinking
80H	Force cursor to the beginning of 1st line
C0H	Force cursor to the beginning of 2nd line
38H	Use 2 lines and 5×7 matrix
08H	Display OFF, Cursor OFF

Table 3.3 LCD Commands

ALP 3.9 ALP to show letters “YES” on LCD

To send command for LCD we have to make
RS=0 Pin P3.2=0
R/W=0 Pin P3.1=0
E= High to Low Pulse Pin P3.0=1 to 0
To send data for LCD we have to make
RS=1 Pin P3.2=1
R/W=0 Pin P3.1=0
E= High to Low Pulse Pin P3.0=1 to 0

```
;-------------------------------------------------------------------------------------------------------
LDATA EQU P1 ;LCD data lines connected at P3
RS EQU P3.2 ;LCD RS pin connected at P3.2
RW EQU P3.1 ;LCD RW pin connected at P3.1
E EQU P3.0 ;LCD E pin connected at P3.0
BUSY EQU P1.7 ;LCD BUSY bit connected at P1.7
;-------------------------------------------------------------------------------------------------------
LCD_INI:    MOV A,#38H ; Initialize LCD 2 lines 5 x 7 matrix
            ACALL COMMAND ; Command subroutine Call
            MOV A,#0EH ;Display On cursor On
            ACALL COMMAND ; Command subroutine Call
            MOV A,#01H ;Clear LCD
            ACALL COMMAND ; Command subroutine Call
            MOV A,#06H ;Shift cursor right
            ACALL COMMAND ; Command subroutine Call
            MOV A,#84H ; Cursor at line 1, position 4
            ACALL COMMAND ; Command subroutine Call
            MOV A, #‘Y’ ; display Letter Y on LCD
            ACALL DISPLAY ;display subroutine call
            MOV A, #‘E’ ; display Letter E on LCD
            ACALL DISPLAY ;display subroutine call
            MOV A, #‘S’ ; display Letter S on LCD
            ACALL DISPLAY ;display subroutine call
STOP:       SJMP STOP ;STOP
```

```
COMMAND: ACALL DELAY ; subroutine Checking display LCD status (ready/busy)
         MOV LDATA,A ; Send command to LCD on data pin
         CLR RS ; RS=0 for command
         CLR RW ; RW=0 for LCD write operation
         SETB E ; E=1    for H-to-L pulse
         NOP
         NOP
         CLR E ; E=0  for H-to-L pulse
         RET ; Return to main program
;-------------------------------------------------------------------------------------------------------------------------------
DISPLAY: ACALL DELAY ;subroutine Checking display LCD status (ready/busy)
         MOV LDATA,A ; Send data to LCD on data pin
         SETB RS ; RS=1 for data
         CLR RW ; RW=0 for write operation
         SETB E ; E=1    for H-to-L pulse
         NOP
         NOP
         CLR E ; E=0  for H-to-L pulse
         RET ; Return to main program
;-----------------------------------------------------------------------------------------
DELAY:   MOV TMOD,#0H; Selection of Timer 0 in mode 1
         MOV TH0,#0EEH; Load higher byte of timer 0 with count EEH
         MOV TL0,#00H; Load lower byte of timer 0 with count  00H
         SETB  TR0 ; Start the Timer
AGAIN:   JNB TF0; AGAIN; Monitor TF0 flag to check  count finish or not
         CLR  T0 ; stop the Timer
         CLR TF0; Clear over flow flag
         RET ; Return from subroutine
```

Example 12-2

Write an 8051 C program to send letters ‘M’, ‘D’, and ‘E’ to the LCD using the busy flag method.

Solution:

```
#include <reg51.h>
sfr ldata = 0x90; //P1=LCD data pins
sbit rs = P2^0;
sbit rw = P2^1;
sbit en = P2^2;
sbit busy = P1^7;
void main()
{
lcdcmd(0x38);
lcdcmd(0x0E);
lcdcmd(0x01);
lcdcmd(0x06);
lcdcmd(0x86); //line 1, position 6
lcddata(‘M’);
lcddata(‘D’);
lcddata(‘E’);
}
```

```
void lcdcmd(unsigned char value)
{
lcdready(); //check the LCD busy flag
ldata = value; //put the value on the pins
rs = 0;
rw = 0;
en = 1; //strobe the enable pin
MSDelay(1);
en = 0;
return;
}
void lcddata(unsigned char value)
{
lcdready(); //check the LCD busy flag
ldata = value; //put the value on the pins
rs = 1;
rw = 0;
en = 1; //strobe the enable pin
MSDelay(1);
en = 0;
return;
}
```

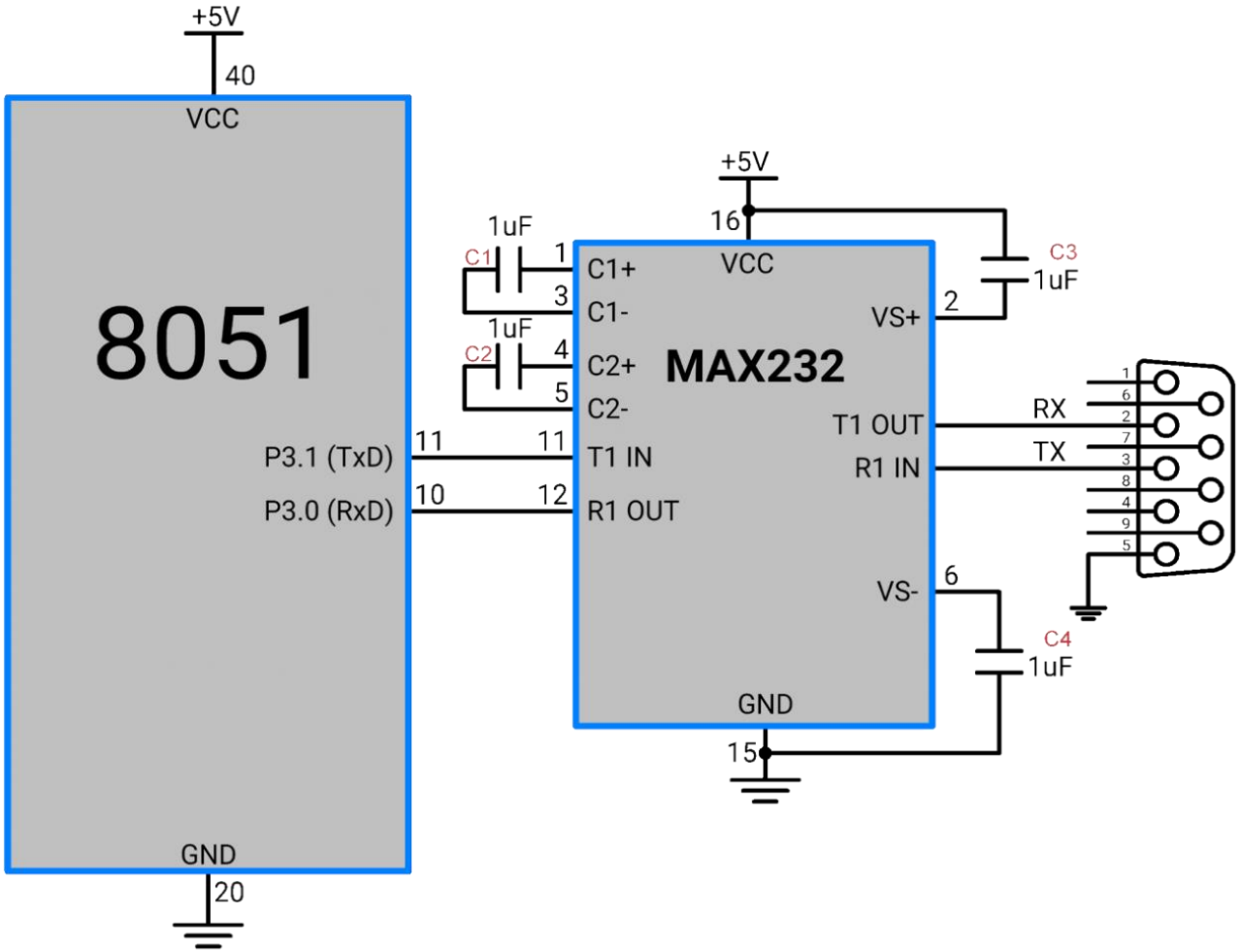
```
void lcdready()
{
busy = 1; //make the busy pin at input
rs = 0;
rw = 1;
en = 1; //strobe the enable pin
MSDelay(1);
en = 0;
while(busy==1){ //wait here for busy flag
}

void lcddelay(unsigned int itime)
{
unsigned int i, j;
for(i=0;i<itime;i++)
for(j=0;j<1275;j++);
}
```

Interfacing of MAX 232 WITH 8051

TTL	
LOGIC 0	0 V
LOGIC 1	5 V

RS232	
LOGIC 0	+3 to +25 V
LOGIC 1	-3 to -25 V



Write a program for the 8051 to transfer "YES" serially at 9600 baud, 8-bit data, 1 stop bit, do this continuously

Solution:

```
        MOV  TMOD,#20H   ;timer 1,mode 2(auto reload)
        MOV  TH1,#-3     ;9600 baud rate
        MOV  SCON,#50H   ;8-bit, 1 stop, REN enabled
        SETB TR1         ;start timer 1
AGAIN:  MOV  A,#"Y"      ;transfer "Y"
        ACALL TRANS
        MOV  A,#"E"      ;transfer "E"
        ACALL TRANS
        MOV  A,#"S"      ;transfer "S"
        ACALL TRANS
        SJMP AGAIN       ;keep doing it
;serial data transfer subroutine
TRANS:  MOV  SBUF,A      ;load SBUF
HERE:   JNB  TI,HERE     ;wait for the last bit
        CLR  TI          ;get ready for next byte
         RET
```

Write a program for the 8051 to receive bytes of data serially, and put them in P1, set the baud rate at 4800, 8-bit data, and 1 stop bit

Solution:

```
        MOV  TMOD,#20H  ;timer 1,mode 2(auto reload)
        MOV  TH1,#-6    ;4800 baud rate
        MOV  SCON,#50H  ;8-bit, 1 stop, REN enabled
        SETB TR1        ;start timer 1
HERE:   JNB  RI,HERE    ;wait for char to come in
        MOV  A,SBUF     ;saving incoming byte in A
        MOV  P1,A       ;send to port 1
        CLR  RI         ;get ready to receive next
                        ;byte
        SJMP HERE       ;keep getting data
```