Shri Ramdeobaba College of Engineering and Management, Nagpur



**Semester:** VI EC (Section B) **Session:** 2020-21

**Course:** Embedded Systems

**Course Code:** ECT

**Assignment-1**

**REPORT**

**Submitted By:**

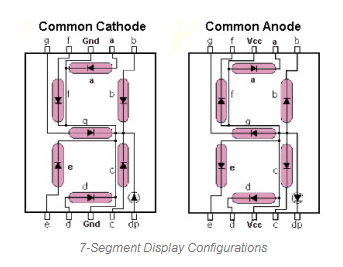
Mr. Adivardhan Maheshwari - 24

**Guided by:**

Prof. Deepak Khushalani

**1.Multiplex 7-segment display to display “HELLO EC”**

Here, we use a Multi-plexed 7-segment Common Cathode.



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| H | G | F | E | D | C | B | A | Digit |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | H |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | E |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | L |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | L |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | O |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | E |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | C |

The 7-segment display is interfaced at port1 and port2 of 8051.

Port1 is connected to the leds of 7-segment.

Port2 is used to select the 7-sement display out of 8.

Eg.: mov p2,#11111110b

This instruction is used to select the 1st Display.

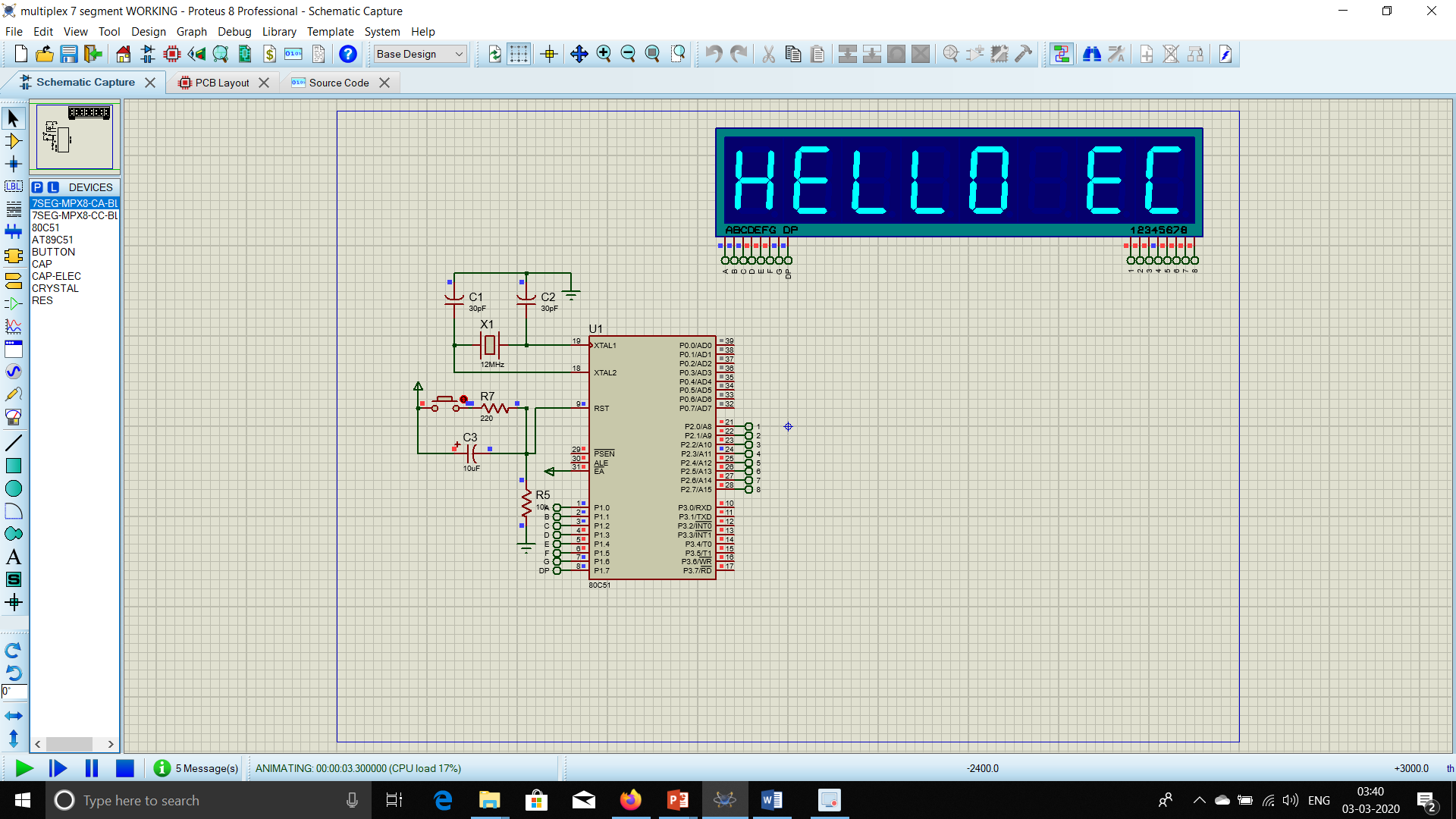
mov p1,#01110110b

Data is sent according to the above hex table for ‘E’

ACALL DELAY

Delay should be less than the minimum time required for the eye to perceive change. If delay is increased beyond that, then we will be able to see only one letter at a time.

The above process is done again and again for each display and it appears as if a single long text is printed continuously.



org 0000h

LOOP: mov p2,#11111110b

mov p1,#01110110b

ACALL DELAY

mov p2,#11111101b

mov p1,#01111001b

ACALL DELAY

mov p2,#11111011b

mov p1,#00111000b

ACALL DELAY

mov p2,#11110111b

mov p1,#00111000b

ACALL DELAY

mov p2,#11101111b

mov p1,#00111111b

ACALL DELAY

mov p2,#11011111b

mov p1,#00000000b

ACALL DELAY

mov p2,#10111111b

mov p1,#01111001b

ACALL DELAY

mov p2,#01111111b

mov p1,#00111001b

ACALL DELAY

LJMP LOOP

DELAY: MOV TH1,#0DCH

MOV TL1,#0

SETB TR1

HERE3: JNB TF1,HERE3

CLR TR1

CLR TF1

RET

End

**2. Interface LCD Display and 3x4 keypad. Display the key pressed at keypad on lcd display.**

Keypad is interfaced at port1 and lcd display at port2.

We continuously check the values after change in voltage at every row and column.

Eg.: label1: mov p1,#0feh

//here we sent #feh to make all pins hign except row1.

mov r1,p1

//we check the new value at port1(it will get changed if a key is pressed)

mov a,r1

cjne a,#0eeh,label2

//if key’1’ is pressed, the value at port1 will become #eeh

mov a,#01h

//#01h is an at command use to clear screen. This will clear

all the previous data.

acall cmnd

//used to send command instruction like set cursor or clear screen

mov a,#'1'

acall disp

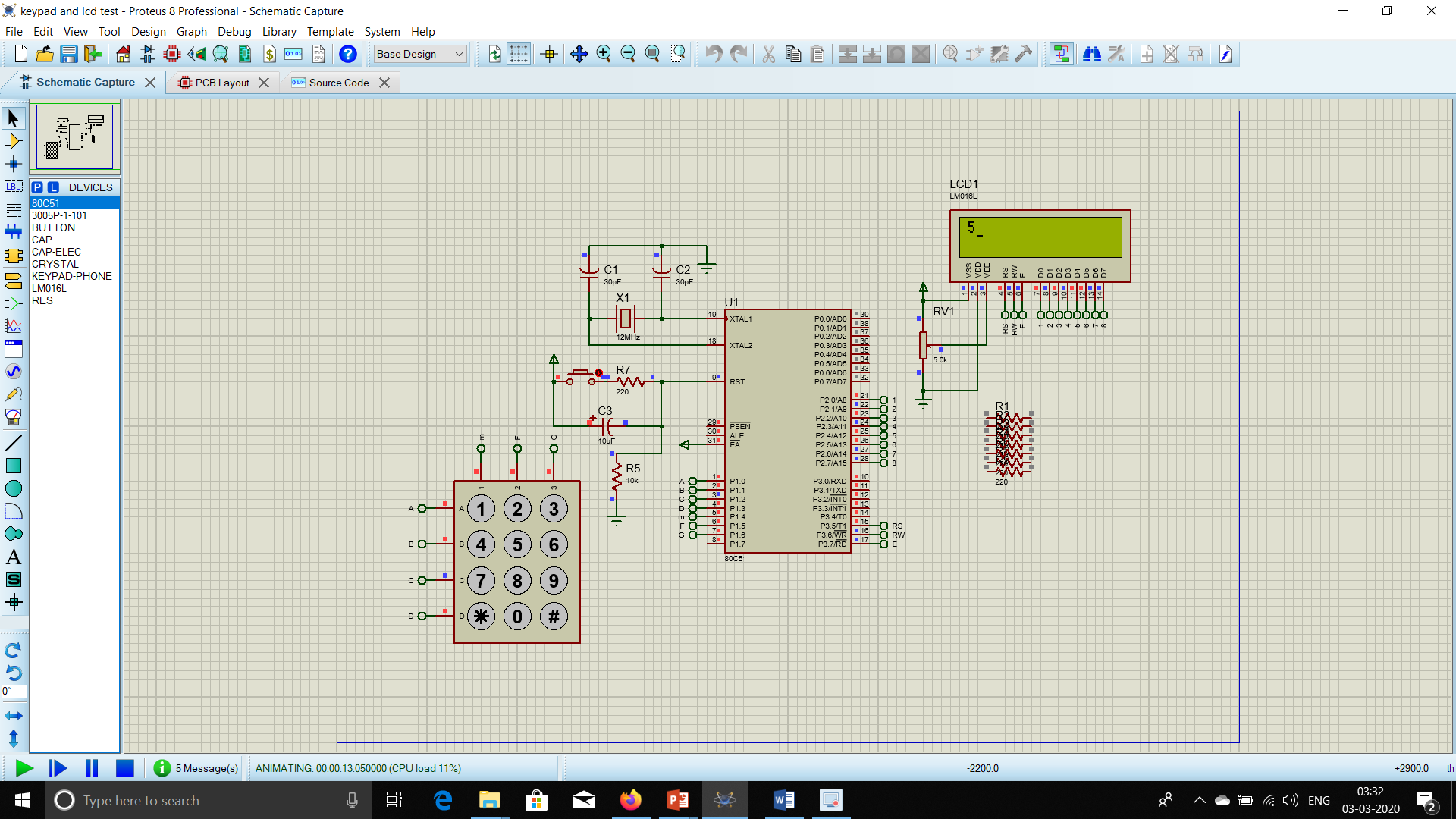
//used to display the data stored in accumulator

ljmp label1

//to recheck from the beginning

Whenever a key is pressed the instructions inside loop are executed.

Other than this, lcdint contains all the initialization instructions and disp contains the display instructions. We need delay so that when a key is pressed it is detected as a single press and not multiple presses.



data1 equ p2

rs equ p3.5

rw equ p3.6

en equ p3.7

org 0000h

ljmp main

main: nop

acall lcdint

label1: mov p1,#0feh

mov r1,p1

mov a,r1

cjne a,#0eeh,label2

mov a,#01h

acall cmnd

mov a,#'1'

acall disp

ljmp label1

label2: mov a,r1

cjne a,#0deh,label3

mov a,#01h

acall cmnd

mov a,#'2'

acall disp

ljmp label1

label3: mov a,r1

cjne a,#0beh,label4

mov a,#01h

acall cmnd

mov a,#'3'

acall disp

ljmp label1

label4: mov p1,#0fdh

mov r1,p1

mov a,r1

cjne a,#0edh,label5

mov a,#01h

acall cmnd

mov a,#'4'

acall disp

ljmp label1

label5: mov a,r1

cjne a,#0ddh,label6

mov a,#01h

acall cmnd

mov a,#'5'

acall disp

ljmp label1

label6: mov a,r1

cjne a,#0bdh,label7

mov a,#01h

acall cmnd

mov a,#'6'

acall disp

ljmp label1

label7: mov p1,#0fbh

mov r1,p1

mov a,r1

cjne a,#0ebh,label8

mov a,#01h

acall cmnd

mov a,#'7'

acall disp

ljmp label1

label8: mov a,r1

cjne a,#0dbh,label9

mov a,#01h

acall cmnd

mov a,#'8'

acall disp

ljmp label1

label9: mov a,r1

cjne a,#0bbh,label10

mov a,#01h

acall cmnd

mov a,#'9'

acall disp

ljmp label1

label10: mov p1,#0f7h

mov r1,p1

mov a,r1

cjne a,#0e7h,label11

mov a,#01h

acall cmnd

mov a,#'\*'

acall disp

ljmp label1

label11:mov a,r1

cjne a,#0d7h,label12

mov a,#01h

acall cmnd

mov a,#'0'

acall disp

ljmp label1

label12: mov a,r1

cjne a,#0b7h,label13

mov a,#01h

acall cmnd

mov a,#'#'

acall disp

label13 ljmp label1

lcdint: mov a,#38h

acall cmnd

mov a,#0fh

acall cmnd

mov a,#01h

acall cmnd

mov a,#06h

acall cmnd

mov a,#80h

acall cmnd

mov a,#3ch

acall cmnd

ret

cmnd: mov data1,A

clr rs

clr rw

setb en

acall delay

clr en

ret

disp: mov data1,A

setb rs

clr rw

setb en

acall delay

clr en

ret

delay: mov r3,#255

mov r0,#255

l1: mov r2,#255

l2: djnz r2,l2

djnz r3,$

djnz r0,l1

ret

END

**3. Interface a stepper motor at port1 using a motor driver IC as a bridge. WAP to rotate stepper motor at 45o clockwise at 100rpm and 180o anti-clockwise at 50rpm.**

**Calculations:**

1. Total no. of steps for 360 degree completion if step angle = 1.8 degree = 360/1.8 = 200

This means we require 200 steps are required to complete one rotation.

1. For 45o clockwise, total steps = 200/8=25
2. For 180o anti-clockwise, total steps = 200/2=100
3. RPM = revolution per minute ( i.e how many rotation per minute of 200 steps)
4. 100 RPM Clockwise

1 revolution time (in msec) = (1/RPM) \* 60\*1000  
 = (1/100)\*60\*1000

= 600msec

Therefore, 1 step time = 600/200 = 3ms.

There, delay1 is of 3ms.

1. 50 RPM Anti-Clockwise

1 revolution time (in msec) = (1/RPM) \* 60\*1000  
 = (1/50)\*60\*1000

= 1200msec

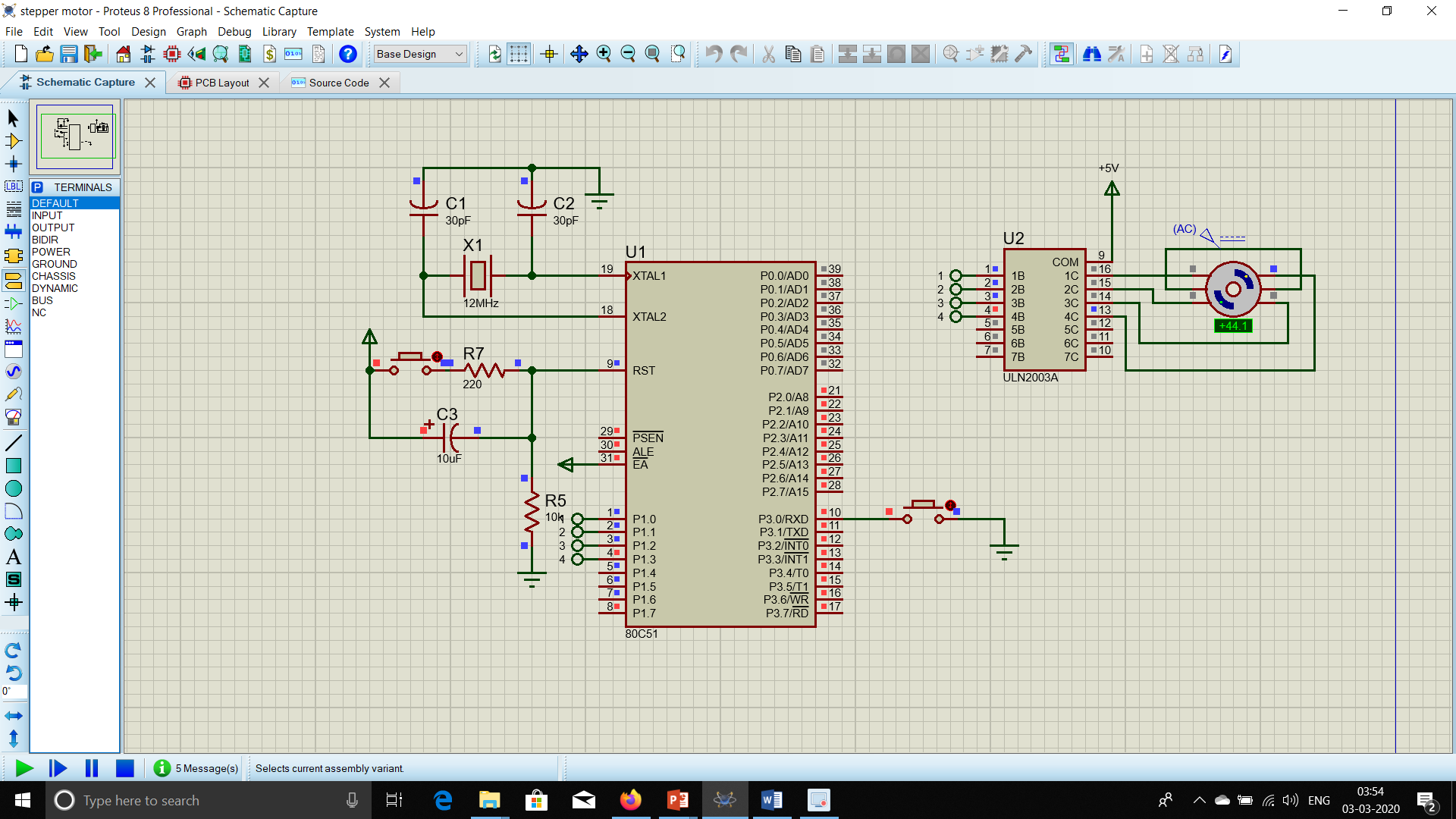
Therefore, 1 step time = 1200/200 = 6ms.

There, delay2 is of 6ms.

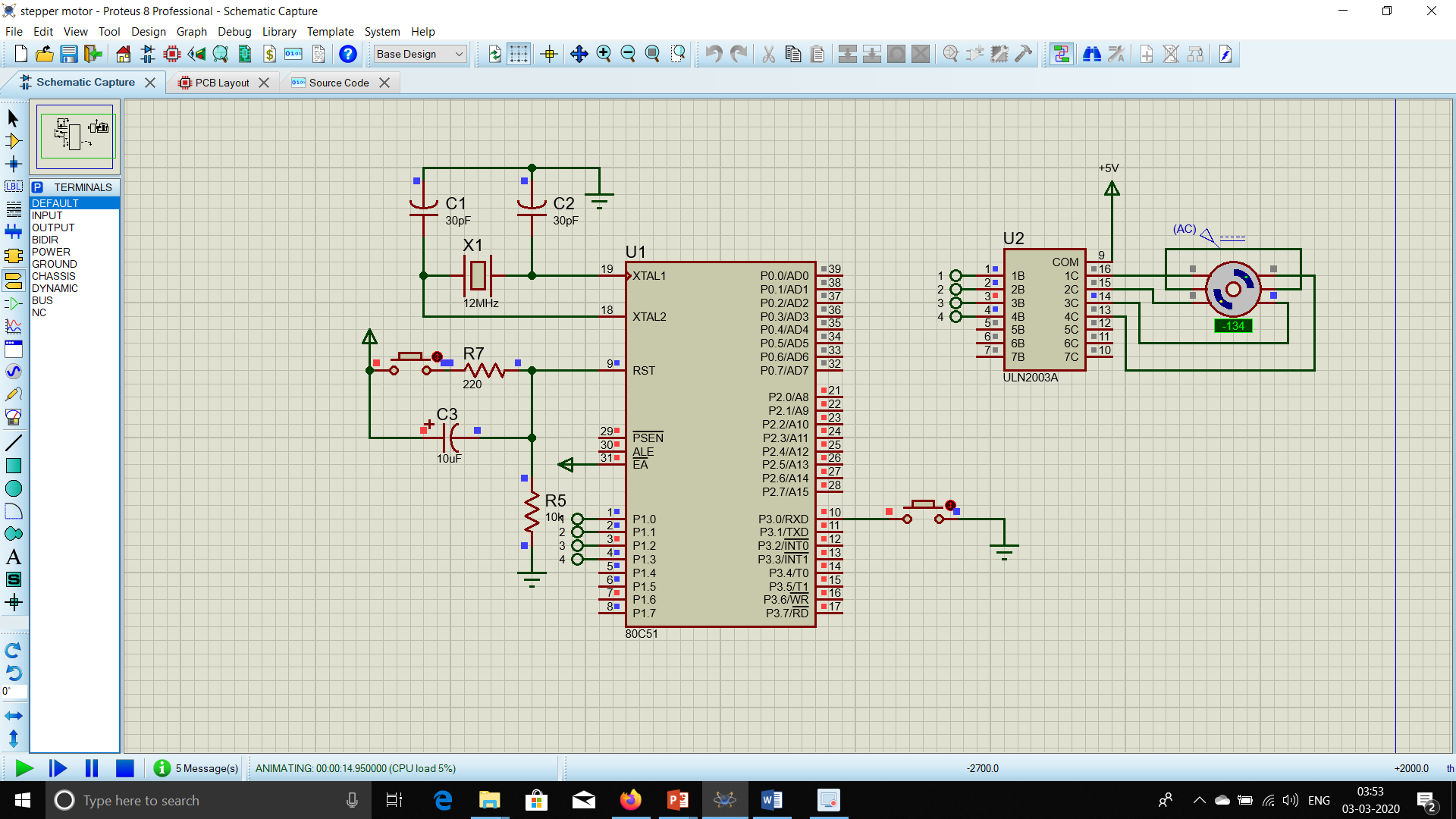
DELAY loop is of 1ms. So to generate 3 ms of delay we call delay loop 3 times inside delay1. Similarly, to generate 6 ms of delay we call delay loop 6 times inside delay2.

As we need to apply only one end of stepper motor high at a time, we load accumulator with 11h(00010001b).

To start the motor rotation, we interfaced a switch at p3.0. Whenever p3.0 is made low, the motor starts to 45o clockwise at 100rpm and 180o anti-clockwise at 50rpm.



Clockwise 45o

Clockwise 45o + AntiClockwise 180o

org 0000h

ljmp main

org 0100h

main: setb p3.0

jb p3.0,$

Start: mov a,#011h

mov r1,#025d

back1: rr a

mov p1,a

acall delay2

djnz r1,back1

mov r1,#0100d

back: mov p1,A

rl A

Acall delay1

djnz r1,back

ljmp main

delay1: MOV R6,#3D

LOOP1: ACALL DELAY

DJNZ R6,LOOP1

ret

delay2: MOV R6,#6D

LOOP2: ACALL DELAY

DJNZ R6,LOOP2

ret

DELAY: MOV TMOD,#00000001B

MOV TH0,#0FCH

MOV TL0,#018H

SETB TR0

JNB TF0,$

CLR TR0

CLR TF0

RET

END