

**دانشکده مهندسی برق**

**تمرین های شبیه سازی ریز پردازنده**

**تمرین سری 7**

**تهیه کننده و نویسنده:**

**رضا آدینه پور**

**استاد مربوطه:**

**جناب اقای دکتر حسین خسروی**

**تاریخ تهیه و اراﺋﻪ:**

**آذر ماه 1400**

1. **یک میکرو را به آیسی حافظه eeprom مدل AT24C512 با رابط i2c وصل کنید و یک صفحه کلید هم به میکرو وصل کنید.  
   برای اولین بار که برنامه اجرا می شود، آدرس خانه های 210 تا 213 را خوانده و اگر محتویات همگی صفر بود، پیام زیر را بدهد.  
   Welcome   
   Set Password:   
   بعد از دریافت کد چهار رقمی از کاربر، مجدد کد را دریافت کند (Verify Password) و سپس آن را در همان خانه های 210 تا 213 ذخیره کند.  
   برای دفعات بعد، پیام Enter Password را نشان دهد و پس از دریافت عدد چهار رقمی از کاربر، آن را با رشته مندرج در در خانه های حافظه eeprom مقایسه کند، اگر درست بود، Login Succeeded و اگر استباه بود، پیام Login Failed را نشان دهد و دوباره رمز دریافت کند.  
   اگر سه مرتبه رمز اشتباه وارد شد، سیستم به مدن 10 ثانیه قفل شده و ثانیه شمار به صورت معکوس از 10 تا صفر بشمارد و مجدد رمز دریافت کند.**
2. **(تمرین اختیاری): استفاده از ال سی دی گرافیکی و ساخت یک ساعت انالوگ-دیجیتال.**

**فرکانس کاری میکرو در CodeVision و Proteus، 8 مگاهرتز تنظیم شده است.**

**کد سوال اول به صورت زیر است:**

#include <mega32.h>

#include <i2c.h>

#include <alcd.h>

#include <stdio.h>

#include <delay.h>

unsigned int i = 210;

unsigned char str[2], j = 0, k = 0, e, z = 0, setPass, correctPass[4], code[4], flag = 2;

void byteWrite(unsigned char deviceAddres, unsigned int addres, unsigned char data)

{

unsigned char lowAddres, highAddres;

lowAddres = addres;

highAddres = (addres >> 8);

deviceAddres <<= 1;

i2c\_start();

i2c\_write(deviceAddres);

i2c\_write(highAddres);

i2c\_write(lowAddres);

i2c\_write(data);

i2c\_stop();

delay\_ms(10);

}

unsigned char randomRead(unsigned char deviceAddres, unsigned int addres)

{

unsigned char lowAddres, highAddres, read;

lowAddres = addres;

highAddres = (addres >> 8);

deviceAddres <<= 1;

i2c\_start();

i2c\_write(deviceAddres);

i2c\_write(highAddres);

i2c\_write(lowAddres);

i2c\_start();

i2c\_write(deviceAddres | 1);

read = i2c\_read(0);

i2c\_stop();

delay\_ms(10);

return read;

}

interrupt [EXT\_INT0] void ext\_int0\_isr(void)

{

if(setPass == 0)

{

correctPass[j] = PINC & 0x0f;

switch(correctPass[j])

{

case 0:

correctPass[j] = 7;

lcd\_putchar('7');

break;

case 1:

correctPass[j] = 4;

lcd\_putchar('4');

break;

case 2:

correctPass[j] = 1;

lcd\_putchar('1');

break;

case 3:

correctPass[j] = 0;

lcd\_putchar('0');

break;

case 4:

correctPass[j] = 8;

lcd\_putchar('8');

break;

case 5:

correctPass[j] = 5;

lcd\_putchar('5');

break;

case 6:

correctPass[j] = 2;

lcd\_putchar('2');

break;

case 7:

correctPass[j] = 0;

lcd\_putchar('0');

break;

case 8:

correctPass[j] = 9;

lcd\_putchar('9');

break;

case 9:

correctPass[j] = 6;

lcd\_putchar('6');

break;

case 10:

correctPass[j] = 3;

lcd\_putchar('3');

break;

case 11:

correctPass[j] = '=';

lcd\_putchar('=');

break;

case 12:

correctPass[j] = '/';

lcd\_putchar('/');

break;

case 13:

correctPass[j] = '\*';

lcd\_putchar('\*');

break;

case 14:

correctPass[j] = '-';

lcd\_putchar('-');

break;

case 15:

correctPass[j] = '+';

lcd\_putchar('+');

}

j++;

}

if(flag == 0)

{

code[k] = PINC & 0x0f;

switch(code[k])

{

case 0:

code[k] = 7;

lcd\_putchar('7');

break;

case 1:

code[k] = 4;

lcd\_putchar('4');

break;

case 2:

code[k] = 1;

lcd\_putchar('1');

break;

case 3:

code[k] = 0;

lcd\_putchar('0');

break;

case 4:

code[k] = 8;

lcd\_putchar('8');

break;

case 5:

code[k] = 5;

lcd\_putchar('5');

break;

case 6:

code[k] = 2;

lcd\_putchar('2');

break;

case 7:

code[k] = 0;

lcd\_putchar('0');

break;

case 8:

code[k] = 9;

lcd\_putchar('9');

break;

case 9:

code[k] = 6;

lcd\_putchar('6');

break;

case 10:

code[k] = 3;

lcd\_putchar('3');

break;

case 11:

code[k] = '=';

lcd\_putchar('=');

break;

case 12:

code[k] = '/';

lcd\_putchar('/');

break;

case 13:

code[k] = '\*';

lcd\_putchar('\*');

break;

case 14:

code[k] = '-';

lcd\_putchar('-');

break;

case 15:

code[k] = '+';

lcd\_putchar('+');

}

k++;

}

}

void main(void)

{

DDRA = 0x00;

PORTA = 0x00;

DDRB = 0x00;

PORTB = 0x00;

DDRC = 0x00;

PORTC = 0x00;

DDRD = 0x00;

PORTD = 0x00;

// External Interrupt(s) initialization

// INT0: On

// INT0 Mode: Falling Edge

// INT1: Off

// INT2: Off

GICR|=(0<<INT1) | (1<<INT0) | (0<<INT2);

MCUCR=(0<<ISC11) | (0<<ISC10) | (1<<ISC01) | (0<<ISC00);

MCUCSR=(0<<ISC2);

GIFR=(0<<INTF1) | (1<<INTF0) | (0<<INTF2);

// Bit-Banged I2C Bus initialization

// I2C Port: PORTA

// I2C SDA bit: 1

// I2C SCL bit: 0

// Bit Rate: 100 kHz

// Note: I2C settings are specified in the

// Project|Configure|C Compiler|Libraries|I2C menu.

i2c\_init();

lcd\_init(16);

#asm("sei")

for(i = 210; i <= 213; i++)

{

byteWrite(0x50, i, 0);

}

if((randomRead(0x50, 210) == 0) && (randomRead(0x50, 211) == 0) && (randomRead(0x50, 212) == 0) && (randomRead(0x50, 213) == 0))

{

setPass = 0;

lcd\_gotoxy(0, 0);

lcd\_putsf("Welcome");

lcd\_gotoxy(0, 1);

lcd\_putsf("SetPassword:");

}

while (1)

{

if(j == 4)

{

j = 0;

setPass = 1;

byteWrite(0x50, 210, correctPass[0]);

byteWrite(0x50, 211, correctPass[1]);

byteWrite(0x50, 212, correctPass[2]);

byteWrite(0x50, 213, correctPass[3]);

lcd\_clear();

lcd\_gotoxy(0, 0);

lcd\_putsf("Verify Password");

delay\_ms(3000);

lcd\_clear();

lcd\_gotoxy(1, 0);

lcd\_putsf("Enter Password:");

flag = 0;

}

if(k == 4)

{

k = 0;

flag = 1;

if( (code[0] == randomRead(0x50, 210)) && (code[1] == randomRead(0x50, 211)) && (code[2] == randomRead(0x50, 212)) && (code[3] == randomRead(0x50, 213)) )

{

lcd\_gotoxy(0, 1);

lcd\_putsf("Login Succeeded!");

delay\_ms(3000);

lcd\_clear();

lcd\_gotoxy(1, 0);

lcd\_putsf("Enter Password:");

flag = 0;

}

else

{

lcd\_gotoxy(0, 1);

lcd\_putsf("Login Failed!");

delay\_ms(3000);

lcd\_clear();

lcd\_gotoxy(1, 0);

lcd\_putsf("Enter Password:");

flag = 0;

z++;

if(z == 3)

{

z = 0;

lcd\_clear();

lcd\_gotoxy(0, 0);

lcd\_puts("SystemLocked!");

for(e = 10; e > 0; e--)

{

flag = 1;

lcd\_gotoxy(14, 0);

*sprintf*(str, "%2d", e);

lcd\_puts(str);

delay\_ms(1000);

}

lcd\_clear();

lcd\_gotoxy(1, 0);

lcd\_putsf("Enter Password:");

flag = 0;

}

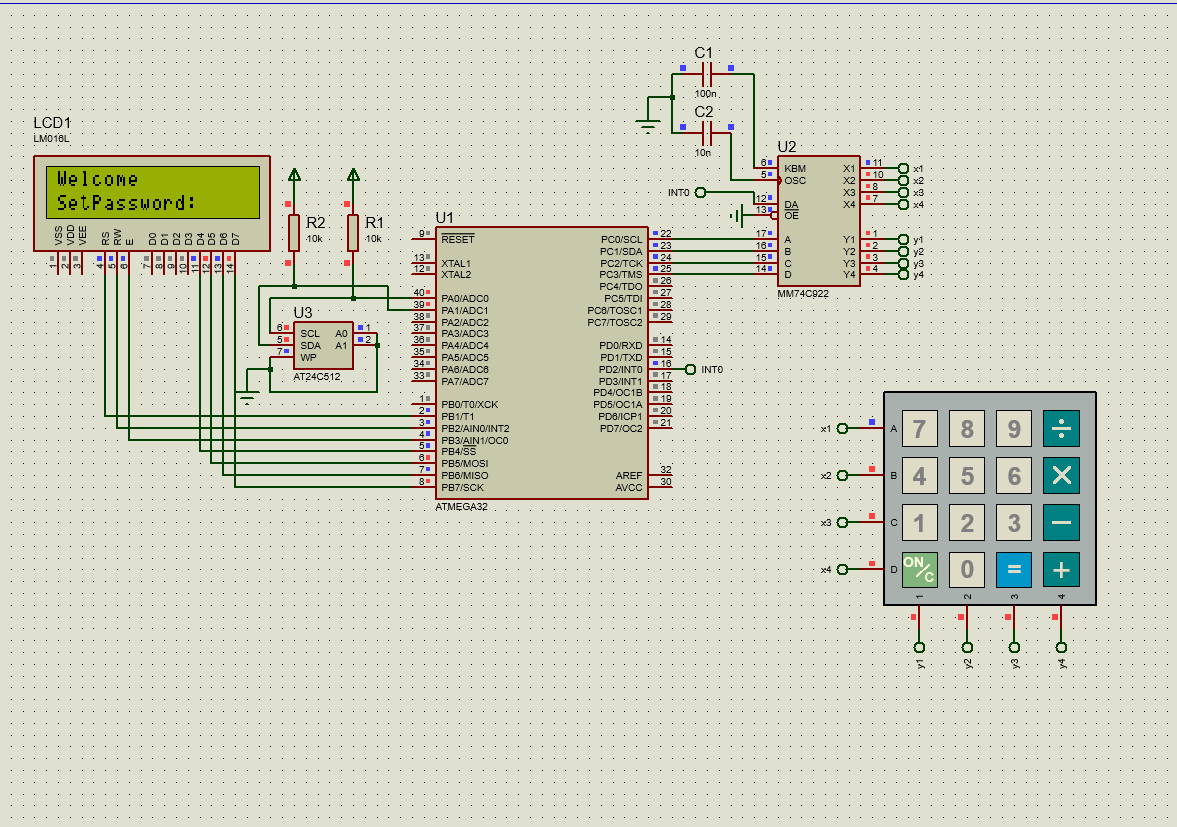
}

}

}//End While(1)

}//End main()

**تصویر شبیه سازی شده این تمرین به صورت زیر است:**

****

**کد تمرین اختیاری به صورت زیر است:**

#include <mega32.h>

#include <delay.h>

#include <stdio.h>

#include "Includes/GLCD.h"

flash unsigned char clock[]=

{

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x80,0x80,0x40,0x20,0x20,0x90,0x88,

0x08,0x00,0x04,0x04,0x02,0x02,0x02,0x02,0x01,0x01,0x01,0x01,0x01,0xF9,0x01,0x01,0x01,0x01,

0x01,0x01,0x02,0x02,0x02,0x02,0x04,0x04,0x80,0xC8,0x08,0x10,0x20,0x20,0x40,0x80,0x80,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0x20,0x18,0x0C,0x02,0x01,0x01,0x00,0x00,0x00,0x00,

0x00,0x01,0x06,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x06,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x01,0x01,0x02,0x0C,0x18,0x20,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0xC0,0x30,0x06,0x01,0x00,0x06,0x04,0x0C,0x08,0x10,0x10,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x08,0x08,0x04,0x04,0x02,0x02,0x00,0x00,0x01,0x06,0x38,0xC0,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0xFC,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x80,0x80,0x80,0x80,0x80,0x80,0x00,0x03,0xFC,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x3F,0xC0,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0x3F,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x0C,0x60,0x80,0x00,0x00,0x40,

0x60,0x20,0x10,0x18,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x08,0x08,0x10,0x30,0x20,0x00,0x00,0x80,0x60,0x1C,

0x03,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,

0x04,0x18,0x30,0x40,0x80,0x80,0x00,0x00,0x00,0x00,0x00,0x00,0x80,0xE0,0x30,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x80,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x30,0xC0,0x80,0x00,0x00,0x00,0x00,0x00,0x80,0x80,0x40,0x30,0x18,0x04,0x03,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x01,0x02,0x04,0x04,0x08,0x12,0x11,0x20,0x20,0x20,

0x40,0x40,0x40,0x40,0x80,0x80,0x80,0x80,0x80,0x80,0x9F,0x80,0x80,0x80,0x80,0x80,0x40,0x40,

0x40,0x40,0x20,0x20,0x20,0x10,0x11,0x08,0x04,0x04,0x02,0x01,0x01,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00

};

unsigned char str[10];

long int i = 0;

interrupt [TIM0\_OVF] void timer0\_ovf\_isr(void)

{

i++;

}

void main(void)

{

int hour = 10, min = 35, sec = 30, Rh = 15, Rs = 23, Rm = 20;

float Xh = 0, Yh = 0, Xs = 0, Ys = 0, Xm, Ym, H;

H = hour + (min/60.0);

DDRA=(1<<DDA7) | (1<<DDA6) | (1<<DDA5) | (1<<DDA4) | (1<<DDA3) | (1<<DDA2) | (1<<DDA1) | (1<<DDA0);

DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) | (0<<DDC1) | (0<<DDC0);

// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: 1000/000 kHz

// Mode: Normal top=0xFF

// OC0 output: Disconnected

// Timer Period: 0/256 ms

TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) | (1<<CS00);

TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (1<<TOIE0);

ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);

TCNT0=0x00;

OCR0=0x00;

glcd\_on();

glcd\_clear();

bmp\_disp(clock, 0, 0, 127, 7);

point\_at(32, 32, 1);

#asm("sei")

while (1)

{

if(i\*256 + TCNT0>=999999)

{

line(32, 32, Xs, Ys, 0, 0);

sec++;

i = 0;

TCNT0 = 0;

Xh = Rh\*cos((3-H)\*2\*3.1415/12)+32;

Yh=-Rh\*sin((3-H)\*2\*3.1415/12)+32;

line(32,32,Xh,Yh,0,1);

Xm=Rm\*cos((15-min)\*2\*3.1415/60)+32;

Ym=-Rm\*sin((15-min)\*2\*3.1415/60)+32;

line(32,32,Xm,Ym,0,1);

Xs=(Rs\*cos((15-sec)\*2\*3.1415/60))+32;

Ys=-Rs\*sin((15-sec)\*2\*3.1415/60)+32;

line(32,32,Xs,Ys,0,1);

*sprintf*(str,"%02d:%02d:%02d",hour,min,sec);

glcd\_puts(str,60,7,0,1,0);

if(sec == 59)

{

sec = 0;

min++;

line(32,32,Xm,Ym,0,0);

if(min == 59)

{

min = 0;

hour++;

line(32, 32, Xh, Yh, 0, 0);

if(hour == 24)

hour = min = sec = 0;

}

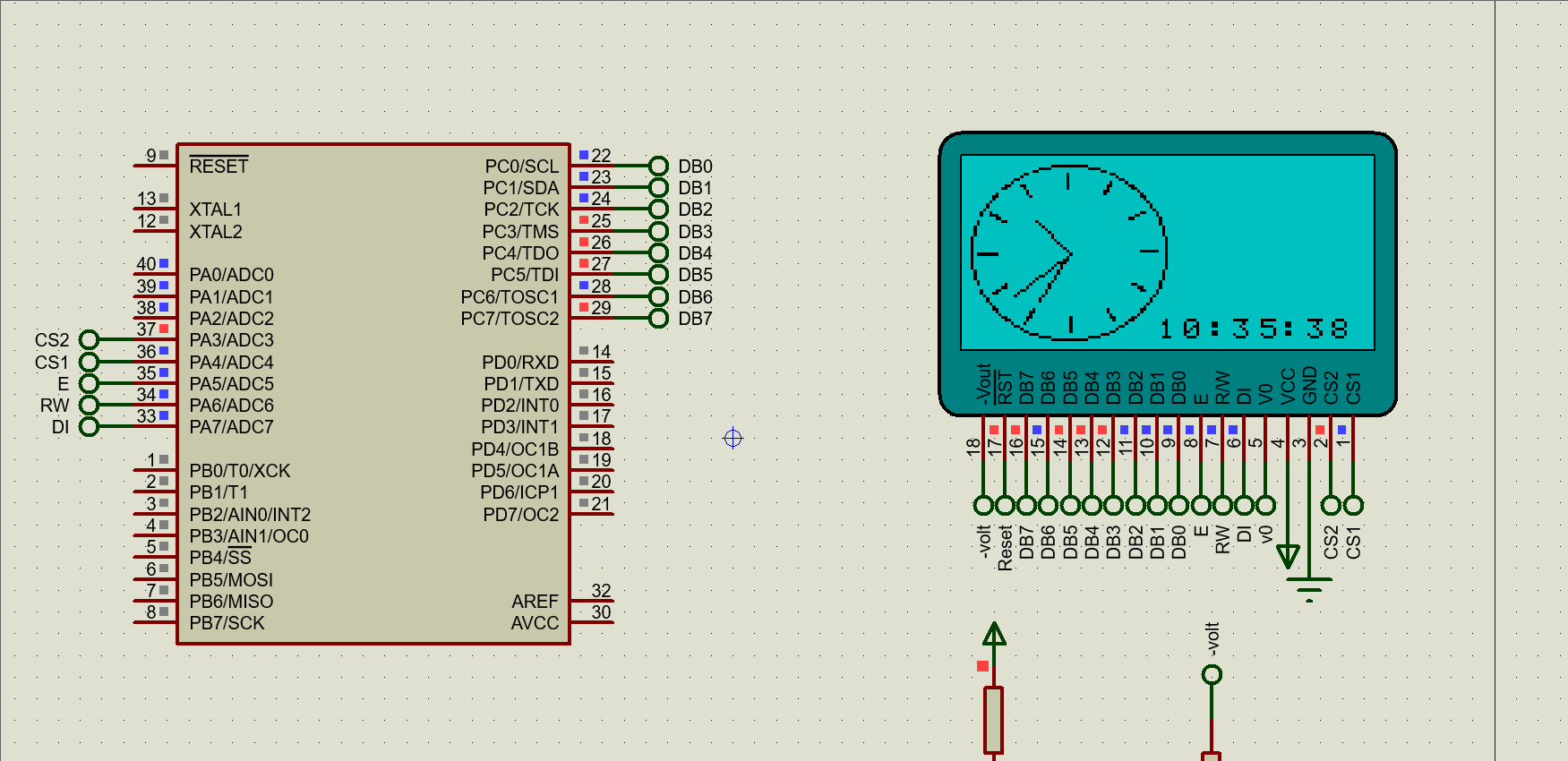
}

}

}//End While()

}//End Main()

**تصویر شبیه سازی شده این تمرین به صورت زیر است:**

****