

تمرینها به صورت گروهی انجام شوند، استفاده از وقفه در تمام سوالها آزاد است.

۱. برای یک چهارراه چراغ راهنما طراحی کنید. همه فرآیند طرح و تصمیم گیری را خودتان انجام دهید. در پروتئوس پیاده سازی نمایید. (برای راحتی فرض کنید در هر مسیر چراغ سبز ۴۰ ثانیه و زرد ۱۰ ثانیه روشن است، چراغ ۳ مسیر دیگر به هنگام سبز یا زرد بودن یک مسیر قرمز است)

برای سادگی ۴ چراغ راهنما داریم که هر بار فقط یک مسیر اجازه حرکت دارد. ابتدا مسیر اول سبز می شود و بقیه قرمز هستند، سپس مسیر اول زرد می شود و بعد از ۱۰ ثانیه قرمز شده و مسیر دوم سبز می شود و به همین ترتیب ادامه می یابد. بهتر است برای زمان بندی دقیق از تایمر استفاده کنیم. به این منظور تایمر ۱ را روی ۱ ثانیه تنظیم کرده (مثلا با کلاک ۱ مگاهرتز و پیش تقسیم کننده ۱۰۲۴،  $n=1000$  خواهد شد که هم در مد نرمال و هم مد CTC میتوان قرار داد) و وقفه آن را فعال می کنیم. در وقفه یک شمارنده قرار می دهیم که در هر حالت با رسیدن به ۴۰ یا ۱۰، وضعیت چراغها را عوض کند و حالت را تغییر دهد. ما ۸ حالت داریم. حالت ۱: چراغ ۱ سبز، حالت ۲: چراغ ۱ زرد، حالت ۳: چراغ ۲ سبز و .... کد

## ۲. روی یک LED ماتریسی حروف اسم خودتان را به ترتیب نمایش دهید. در پروتئوس پیاده سازی کنید.

برای نمایش FARZANEH روی LED ماتریسی در ابتدا آرایه ALPHA را ساخته ایم. سپس برای نمایش هر کاراکتر، به ترتیب ستونهای LED را با توجه به محتویات آرایه PORT روشن می کنیم و با توجه به محتویات ALPHA برای آن کاراکتر، مطابق با ترتیب ستونها مقادیر لازم برای روشن کردن LED در آن ستون را روی سطرها می ریزیم. برای هر کاراکتر ۲۰۰ بار محتویات آن را نشان می دهیم تا چشم آن را به درستی درک کند. همچنین با یک ترفند، چهار ستون اول و دوم را جداگانه روشن می کنیم تا چشمک زدن ستونها را چشم کمتر ببیند.

```
#include <avr/io.h>

//header to enable data flow control over pins

#define F_CPU 1000000

//telling controller crystal frequency attached

#include <util/delay.h>

//header to enable delay function in program

int main(void)
{
    DDRD = 0xFF; //PORTB,C,D are set as output

    DDRB = 0xFF;

    DDRC = 0xFF;

    char PORT[8] = {1,2,4,8,16,32,64,128}; //pin values of PORTD

    static int ALPHA[26][8]={
        {0b01111111,0b11111111,0b11001100,0b11001100,0b11001100,0b11001100,0b11111111,0b01111111},
        {0b00111100,0b01111110,0b11011011,0b11011011,0b11011011,0b11011011,0b11111111,0b11111111},
        {0b11000011,0b11000011,0b11000011,0b11000011,0b11000011,0b11100111,0b01111110,0b00111100},
        {0b01111110,0b10111101,0b11000011,0b11000011,0b11000011,0b11000011,0b11111111,0b11111111},
        {0b11011011,0b11011011,0b11011011,0b11011011,0b11011011,0b11011011,0b11111111,0b11111111},
        {0,0b11011000,0b11011000,0b11011000,0b11011000,0b11011000,0b11111111,0b11111111},
        {0b00011111,0b11011111,0b11011000,0b11011011,0b11011011,0b11011011,0b11111111,0b11111111},
        {0,0b11111111,0b11111111,0b00011000,0b00011000,0b00011000,0b11111111,0b11111111},
        {0b11000011,0b11000011,0b11000011,0b11111111,0b11111111,0b11000011,0b11000011,0b11000011},
        {0b11000000,0b11000000,0b11000000,0b11111111,0b11111111,0b11000011,0b11001111,0b11001111},
        {0,0b11000011,0b11100111,0b01111110,0b00111100,0b00011000,0b11111111,0b11111111},
        {0b00000011,0b00000011,0b00000011,0b00000011,0b00000011,0b00000011,0b11111111,0b11111111},
        {0b11111111,0b11111111,0b01100000,0b01110000,0b01110000,0b01100000,0b11111111,0b11111111},
        {0b11111111,0b11111111,0b00011100,0b00011100,0b01110000,0b11110000,0b11111111,0b11111111},
        {0b01111110,0b11111111,0b11000011,0b11000011,0b11000011,0b11000011,0b11111111,0b01111110},
        {0b01110000,0b11111100,0b11001100,0b11001100,0b11001100,0b11001100,0b11111111,0b11111111},
        {0b01111110,0b11111111,0b11001111,0b11011111,0b11011011,0b11000011,0b11111111,0b01111110},
        {0b01111001,0b11111011,0b11011111,0b11011110,0b11011100,0b11011000,0b11111111,0b11111111},
```

```

{0b11001110,0b11011111,0b11011011,0b11011011,0b11011011,0b11011011,0b11111011,0b01110011},
{0b11000000,0b11000000,0b11000000,0b11111111,0b11111111,0b11000000,0b11000000,0b11000000},
{0b11111110,0b11111111,0b00000011,0b00000011,0b00000011,0b00000011,0b11111111,0b11111110},
{0b11100000,0b11111100,0b00011110,0b00000011,0b00000011,0b00011110,0b11111100,0b11100000},
{0b11111110,0b11111111,0b00000011,0b11111111,0b11111111,0b00000011,0b11111111,0b11111110},
{0b01000010,0b11100111,0b01111110,0b00111100,0b00111100,0b01111110,0b11100111,0b01000010},
{0b01000000,0b11100000,0b01110000,0b00111111,0b00111111,0b01110000,0b11100000,0b01000000},
{0b11000011,0b11100011,0b11110011,0b11111011,0b11011111,0b11001111,0b11000111,0b11000011}}; //c
haracters a,b,c,d,e,f,g,...z binary codecs

char NAME[]={5,0,17,25,0,13,4,7}; //FARZANEH character values

uint8_t l =0;

while(1)
{
    for (int m=0;m<sizeof NAME;m++)
    {
        l = NAME[m];

        for (int n=0;n<200;n++)//execute 200 times for the eye to catch
        {
            for (int j=0;j<5;j++)
            {
                PORTB = ~PORT[j]; // ROW
                PORTC = 0xFF;
                //_delay_us(8000);
                PORTD = ALPHA[l][j]; //show half of character (COLUMN)

                _delay_us(500);
            }

            PORTB=0x00; //clear screen after show.
            PORTC=0x00; //clear screen after show.
            PORTD=0x00; //clear screen after show.
            //_delay_us(5000);
            for (int k=0;k<5;k++)
            {
                PORTC = ~PORT[k]; // ROW
                PORTB = 0xFF;
                PORTD = ALPHA[l][k+4]; //show other half of character(COLUMN)

                _delay_us(500);
            }

            PORTC=0x00; //clear screen after show.
            PORTB=0x00; //clear screen after show.
            PORTD=0x00; //clear screen after show.
            //_delay_us(5000);
        }
    }
}

```

```
_delay_ms(220);
```

```
_delay_ms(220);
```

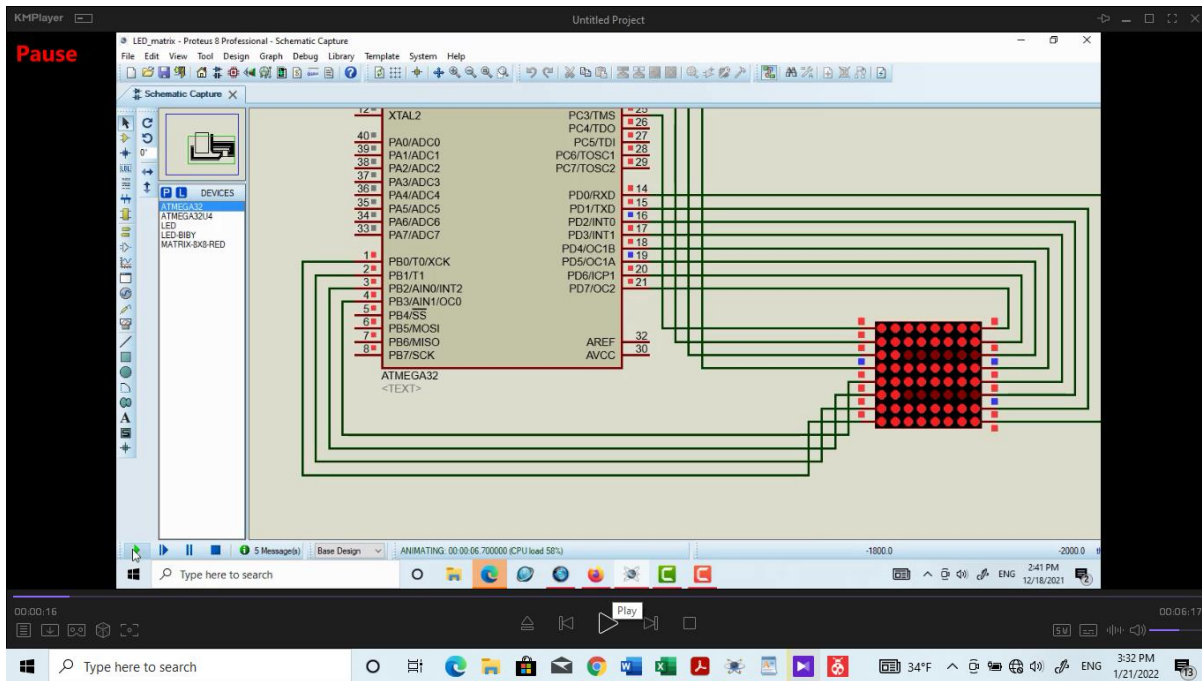
```
_delay_ms(220);
```

```
_delay_ms(220);
```

```
_delay_ms(220);
```

```
}
```

```
}
```



۳. با استفاده از یک LCD یک ساعت دیجیتال که زمان و تاریخ را نشان می‌دهد، بسازید. این ساعت قابلیت کوک کردن برای زنگ زدن در زمان مشخص هم داشته باشد. برای تنظیم ساعت و کوک کردن آن از صفحه کلید استفاده کنید. (استفاده از زمان real time اختیاری است و نمره اضافه دارد)

با استفاده از قطعه DS1307 و ارتباط سریال (مثلاً I2C) می‌توان زمان حقیقی را به میکرو اعلام کرد. سپس یک تایمر ۱ ثانیه‌ای باید تنظیم کرد (اگر از تایمر ۰ یا ۲ استفاده می‌شود روی 0.1 ثانیه تنظیم کنید و با ۱۰ بار شمارش ۱ ثانیه را ثبت کنید) و زمان را به صورت ثانیه/دقیقه/ساعت نمایش داد. یک بازر هم برای اعلام زنگ استفاده می‌شود. توابع rtc\_get\_date و rtc\_get\_time برای گرفتن زمان به کار می‌رود. تاریخ و زمان روی lcd نمایش داده می‌شود. اگر ساعت کوک شده باشد، زمان زنگ هم بر روی lcd به همراه یک ستاره نمایش داده می‌شود. با فشار کلید # تنظیم زنگ انجام می‌شود. بار اول ساعت و بار دوم دقیقه را تنظیم می‌کنیم. در این برنامه از سال کیبسه صرف‌نظر شده است.

```
#include <mega32a.h>
#include <i2c.h>
#include <ds1307.h>
#include <lwire.h>
#include <alcd.h>
#include <delay.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define Buzzer PORTD .0

char lcd_buff[16];
unsigned char Hour, Minute, Second, number = 0;
unsigned char ringHour = 0, ringMinute = 0, ringMode = 0;
unsigned char tmpHour = 0, tmpMinute = 0, tmpMode = 0;
unsigned char Year, Month, Day, Weekday = 0;
unsigned char menu = 0;

void time_date_functions();
void ring_menu();
void showMenu();
void main_loop();

interrupt[TIM2_OVF] void timer2_ovf_isr(void)
{
    if (number >= 0 && Second <= 8)
    {
        number++;
    }
    else if (number >= 9)
    {
        if (Second >= 0 && Second <= 58)
        {
            Second++;
        }
        else if (Second >= 59)
        {
            Second = 0;
            Minute++;
            if (Minute > 59)
            {
                Minute = 0;
                Hour++;
                if (Hour > 23)
                {

```

```

        Hour = 0;
        Day++;
        Weekday++;
    }
}
}

int key_pad(void)
{
    PORTA .3 = 0;
    PORTA .4 = 1;
    PORTA .5 = 1;
    PORTA .6 = 1;
    if (!PINA .0)
        return 3;
    if (!PINA .1)
        return 2;
    if (!PINA .2)
        return 1;
    PORTA .3 = 1;
    PORTA .4 = 0;
    PORTA .5 = 1;
    PORTA .6 = 1;
    if (!PINA .0)
        return 6;
    if (!PINA .1)
        return 5;
    if (!PINA .2)
        return 4;
    PORTA .3 = 1;
    PORTA .4 = 1;
    PORTA .5 = 0;
    PORTA .6 = 1;
    if (!PINA .0)
        return 9;
    if (!PINA .1)
        return 8;
    if (!PINA .2)
        return 7;
    PORTA .3 = 1;
    PORTA .4 = 1;
    PORTA .5 = 1;
    PORTA .6 = 0;
    if (!PINA .0)
        return 12;
    if (!PINA .1)
        return 0;
    if (!PINA .2)
        return 10;
    return 13;
}

int key;
void main(void)
{
    DDRA = 0b01111000;
    PORTA = 0b00000111;
    DDRB = 0xFF;
    DDRC = 0xFF;
    DDRD = 0xFF;

    ASSR = 1 << AS2;
    TCCR2 = (0 << PWM2) | (0 << COM21) | (0 << COM20) | (0 << CTC2) | (1 << CS22) |
(0 << CS21) | (1 << CS20);
    TCNT2 = 0x**;//این مقدار را باید محاسبه کرد برای یک دهم ثانیه
    OCR2 = 0x00;

```

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

```
i2c_init();
rtc_init(0, 0, 0);
wl_init();

lcd_init(16);
lcd_clear();
#asm("sei")
rtc_get_time(&Hour, &Minute, &Second);
rtc_get_date(&Weekday, &Day, &Month, &Year);
main_loop();
}
```

```
void main_loop()
{
    while (1)
    {
        key = key_pad();
        if (key == 12)
        {
            ring_menu();
        }

        time_date_functions();

        if (menu == 0)
        {
            sprintf(lcd_buff, "%i/%i/%i ", Year, Month, Day);
            lcd_gotoxy(0, 0);
            lcd_puts(lcd_buff);

            sprintf(lcd_buff, "%i:%i:%i ", Hour, Minute, Second);
            lcd_gotoxy(0, 1);
            lcd_puts(lcd_buff);

            sprintf(lcd_buff, "%i:%i", ringHour, ringMinute);
            lcd_gotoxy(10, 1);
            lcd_puts(lcd_buff);
            lcd_gotoxy(15, 1);
            if (ringMode == 1)
            {
                lcd_puts("*");
            }
            else
            {
                lcd_puts(" ");
            }
        }
        else
        {
            showMenu();
        }

        if (Hour == ringHour && Minute == ringMinute && ringMode == 1)
        {
            unsigned int i_sound;
            for (i_sound = 1; i_sound <= 10; ++i_sound)
            {
                Buzzer = 1;
                delay_us(800);
                Buzzer = 0;
                delay_us(800);
            };
            Buzzer = 0;
        }
        else
    }
}
```

```

        {
            Buzzer = 0;
        }
    }
}
void ring_menu()
{
    lcd_clear();
    if (menu >= 3)
    {
        ringHour = tmpHour;
        ringMinute = tmpMinute;
        ringMode = tmpMode;
        menu = 0;
    }
    else
        menu++;
    delay_ms(20);
}

void showMenu()
{
    if (tmpHour >= 24)
    {
        tmpHour = 0;
        lcd_clear();
    }
    if (tmpMinute >= 60)
    {
        tmpMinute = 0;
        lcd_clear();
    }
    switch (menu)
    {
        {
        case 1:
            lcd_gotoxy(0, 0);
            lcd_puts("Set Hour:");
            lcd_gotoxy(8, 1);
            if (key != 13 && key != 12)
            {
                tmpHour *= 10;
                tmpHour += key;
            }
            sprintf(lcd_buff, "%i", tmpHour);
            lcd_puts(lcd_buff);
            break;

        case 2:
            lcd_gotoxy(0, 0);
            lcd_puts("Set Minute:");
            lcd_gotoxy(8, 1);
            if (key != 13 && key != 12)
            {
                tmpMinute *= 10;
                tmpMinute += key;
            }
            sprintf(lcd_buff, "%i", tmpMinute);
            lcd_puts(lcd_buff);
            break;

        case 3:
            lcd_gotoxy(0, 0);
            lcd_puts("ON / OFF:");
            lcd_gotoxy(8, 1);
            if (tmpMode == 0)
            {
                lcd_puts("OFF");
            }
        }
    }
}

```



```

        else
        {
            lcd_puts("ON ");
        }

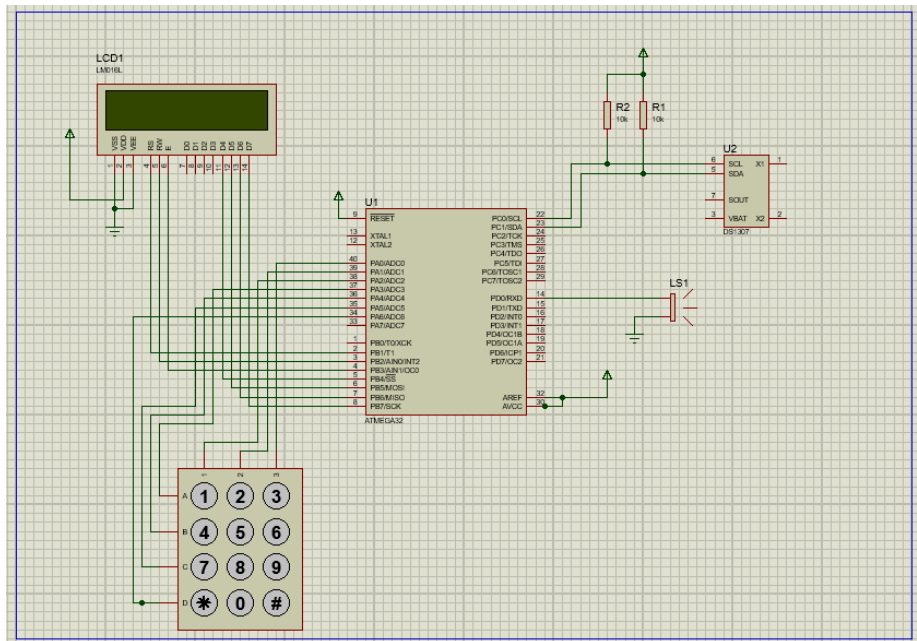
        if (key != 13 && key != 12)
        {
            if (tmpMode == 0)
            {
                tmpMode = 1;
                lcd_gotoxy(8, 1);
                lcd_puts("ON ");
            }
            else
            {
                tmpMode = 0;
                lcd_gotoxy(8, 1);
                lcd_puts("OFF");
            }
        }

        break;

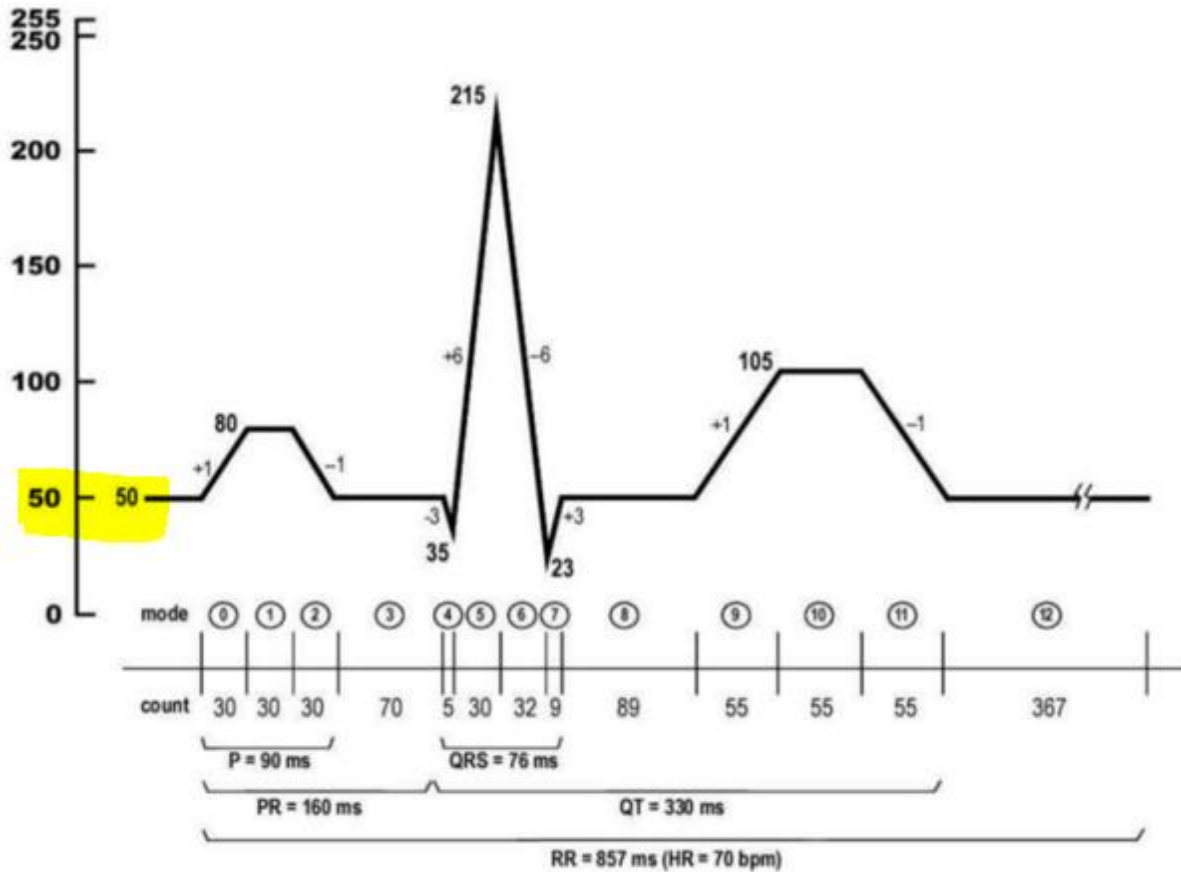
    case 4:
        break;
    }
    delay_ms(22);
}

void time_date_functions()
{
    if (menu == 0)
    {
        if (Weekday == 0 || Weekday > 7)
            Weekday = 1;
        if (Day == 0)
            Day = 1;
        if ((Day > 31) && (Month == 1 || Month == 3 || Month == 5 || Month == 7 ||
Month == 8 || Month == 10 || Month == 12))
        {
            Day = 1;
            Month++;
        }
        if ((Day > 30) && (Month == 4 || Month == 6 || Month == 9 || Month == 11))
        {
            Day = 1;
            Month++;
        }
        if (Day > 28 && Month == 2 && isLeapYear((Year_century * 100) + Year, 1) ==
0)
        {
            Day = 1;
            Month++;
        }
        if (Month == 0)
            Month = 1;
        if (Month > 12)
        {
            Month = 1;
            Year++;
        }
        if (Year > 99)
        {
            Year = 0;
        }
    }
}

```



۴. فرض کنید یک مدار DAC دارید که ۸ بیت ورودی شما را به مقدار آنالوگ تبدیل می کند، به این صورت که مقدار صفر را به صفر ولت و مقدار ۲۵۵ را به ۵ ولت تبدیل می کند. الف) برنامه ای برای تولید شکل موج یک نوار قلب مانند زیر بنویسید (پیاده سازی آن در پروتوس نمره اضافی محسوب می شود). ب) به جز DAC به نظر شما چه روش دیگری برای تولید این شکل موج وجود دارد. از دو دیدگاه سخت افزار و نرم افزار بحث کنید.



تنظیمات اولیه ی برنامه به شرح زیر است:

از میکروکنترلر Atmega32 استفاده می کنیم.

تایمر ۱ را روی یک میلی ثانیه تنظیم می کنیم.

فرکانس کاری میکرو را ۱ مگاهرتز قرار می دهیم.

PORTD را به عنوان خروجی تنظیم می کنیم و مقدار اولیه ی آن را (با توجه به قسمت زرد رنگ در شکل) ۵۰ قرار می دهیم.

سپس دو متغیر mode و count را به عنوان int در نظر می گیریم.

چهارچوب اصلی برنامه که در تایمر ۱ نوشته می شود در قالب برنامه ی switch case می باشد، قبل از پیاده سازی switch case در

تایمر، توابع مورد نیاز را تعریف می کنیم یک از آن توابع به فرم زیر است:

```
void mode_0()
{ count++;
  if(count==30)
  { mode++; count=0; return; }
  PORTD++; return; }
```

طبق شکل سه مد کاری mode 0, mode 1, mode 2 موج P را نمایش می دهد.

موارد مهم در بررسی هر mode:

شیب آن را که با PORTD تغییر می دهیم.

مدت زمانی که در آن mode قرار داریم که با count تعریف می شود.

برای مثال، همین کد را در نظر بگیرید، همان طور که مشاهده می کنید وقتی در mode 0 قرار داریم، مدت زمان در این mode برابر با ۳۰ میلی ثانیه می باشد و شیب نمودار در این قسمت ۱+ است PORTD (نشانگر شیب نمودار است). در mode 1 داریم:

```
void mode_1()
{ count++;
  if(count==30)
  { mode++; count=0; return; }
  return; }
```

حال mode 7 را در نظر بگیرید، مدت زمان این mode برابر با ۹ میلی ثانیه و شیب آن ۳+ است. برنامه ی آن به صورت زیر می باشد.

```
void mode_7(){
  count++;
  if(count==9)
  { mode++; count=0; return; }
  PORTD=PORTD+3; return;}
```

پس از نوشتن توابع mode به صورتی که در بالا بیان شد، تابع switch case و توابع داخلی آن را به صورت زیر در بدنه ی تایمر می نویسیم:

```
switch(mode){
case 0: mode_0(); break;
case 1: mode_1(); break;
case 2: mode_2(); break;
case 3: mode_3();
break;
.
.
.
}
```

در بدنه ی اصلی برنامه (قسمت main) فقط مقدار اولیه ی PORTD که همان ۵۰ است مشخص می کنیم.

قطعات مورد استفاده شامل:

• میکروکنترلر atmega 32

• مبدل DAC0800

• اسیلوسکوپ

• خازن با اندازه های ۱ / ۰ میکروفاراد.

• مقاومت با اندازه های ۱۰ و ۷ / ۴ و ۲ / ۱ کیلو اهم.

• صفحه کلید

• گیت های and و or

در نهایت آنچه که در نرم افزار پروتئوس پیاده سازی شد به شکل زیر می باشد:



۵. در هر مورد از کدام وقفه استفاده می کنید؟ پایه ای که سیگنال باید به آن وصل شود را مشخص کنید (وقفه را فقط نام ببرید، توضیح لازم نیست)

الف) فشار دیگ بخار صنعتی به صورت پیوسته توسط یک سنسور آنالوگ ثبت می شود. می خواهیم هر وقت فشار از حدی بالاتر بود، الارم دهد.

ب) دستگاه ثبت ضربان قلب در هر ضربان یک پالس ایجاد می کند، اگر ضربان قلب در هر ثانیه بیش از حدی شود (فاصله دو پالس کمتر از حدی شود)، باید الارم دهد.

ج) یک دستگاه در حالت stand by است و با فشار یک کلید یا لمس صفحه لمسی آن باید شروع به کار کند

د) برای کنترل فاصله گذاری اجتماعی، درب ورودی فروشگاه پس از ورود n نفر باید بسته شود، سنسور به ازای ورود هر شخص یک پالس ایجاد می کند.

الف) فشار دیگ بخار صنعتی به صورت پیوسته توسط یک سنسور آنالوگ ثبت می شود. می خواهیم هر وقت فشار از حدی بالاتر بود، الارم دهد. وقفه مقایسه گر آنالوگ با پایه AIN0 و AIN1، یا وقفه adc، پایه های آنالوگ

PA0 تا PA7

ب) دستگاه ثبت ضربان قلب در هر ضربان یک پالس ایجاد می کند، اگر ضربان قلب در هر ثانیه بیش از حدی شود (فاصله دو پالس کمتر از حدی شود)، باید الارم دهد. وقفه تایمر با پایه T0 یا T1 یا بهتر است وقفه capture با

پایه ICP

ج) یک دستگاه در حالت stand by است و با فشار یک کلید یا لمس صفحه لمسی آن باید شروع به کار کند وقفه

خارجی با پایه INT0 یا INT1

د) برای کنترل فاصله گذاری اجتماعی، درب ورودی فروشگاه پس از ورود n نفر باید بسته شود، سنسور به ازای ورود هر شخص یک پالس ایجاد می کند. وقفه تایمر (شمارنده) با پایه T0 یا T1

۶. یک سنسور آنالوگ داریم که فشار را اندازه گیری میکند، دامنه سیگنال خروجی فشارسنج از ۰ تا ۵ ولت متغیر است. آنچه برای ما مهم است مدت زمانی است که در یک شبانه روز فشار از حد آستانه مشخصی (مثلا ۴ ولت) بالاتر بوده است. مدار لازم و برنامه مورد نیاز را بنویسید، چه قابلیتی در میکرو به ما کمک می کند. این زمان را بر روی LCD به صورت ثانیه نشان دهد. (این زمان هرگز از ۱ دقیقه بالاتر نمی رود)

این مدار از خاصیت capture تایمر ۱ و از بخش مقایسه گر آنالوگ آن استفاده میکند. پایه PB2 را به خروجی سنسور و پایه PB3 را به ۴ ولت وصل می کنیم. بیت ACIC را در این ثابت ۱ می کنیم.

7	6	5	4	3	2	1	0	
ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	ACSR
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	

و سپس تایمر ۱ را در مد نرمال قرار می دهیم. پرچم AC1 را مدام چک می کنیم تا لحظه ای که مقایسه گر برابری خروجی سنسور با ۴ ولت را capture می کند، ثبت کنیم. دوباره هم این کار را انجام می دهیم و تفاضل زمانی را به دست می آوریم. اگر کلاک ورودی میکرو ۱ مگاهرتز باشد و پیش تقسیم کننده را روی ۱۰۲۴ تنظیم کنیم، کلاک تایمر ۱، تقریباً برابر با ۱ کیلوهرتز است، در این صورت ۱ دقیقه ۶۰۰۰۰ خواهد شد. با تقسیم مقدار حاصل بر ۱۰۰۰ زمان بر حسب ثانیه به دست می آید. کد بسیار ساده و شبیه کدهای جلسه ۱۹ است.