

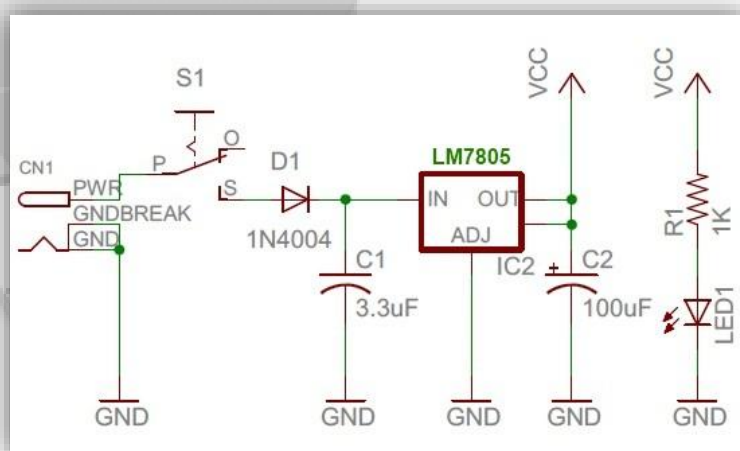
دلیل استفاده از سیستم کنترل نور :

الان در بسیاری از تلفن های هوشمند ، تلویزیون ها و رایانه ها از سیستم تنظیم روشنایی صفحه نمایش در هنگام شرایط نور مختلف استفاده میشود . دلیل آن هم بهبود کیفیت تصویر و همچنین جلوگیری از اذیت شدن چشم است . هنگامی که نور محیط زیاد است نور صفحه نمایش زیاد می شود تا بتوانیم بهتر صفحه نمایش را ببینیم و همچنین هنگامی که نور محیط کم است نور صفحه نمایش کم می شود تا نور شدید چشم ها را اذیت نکند . اگر شما از این مدار در اتاق خواب خود استفاده می کنید در هنگام شب و تاریکی نور سون سگمنت ها کم می شود و روشنایی سون سگمنت ها باعث روشنایی اتاق و همچنین اذیت شدن شما نمی شود . فایده دیگر مصرف کم انرژی است در صورتی که از منبع تغذیه مستقل مثل باتری استفاده می کنید بسیار مهم است .

مدار از سه قسمت زیر تشکیل شده است :

1 - منبع تغذیه ، 2 - میکروکنترلر و حسگرها ، 3 - صفحه نمایش و درایو آن

برای کنترل ولتاژ مدار از رگولاتور 7805 استفاده شده است که باعث ثبات ولتاژ و همچنین و محافظت مدار در برابر ولتاژ های بالا تر از 7 ولت می شود . همچنین یک ال ای دی برای نشان دادن تغذیه مدار قرار داده شده است .

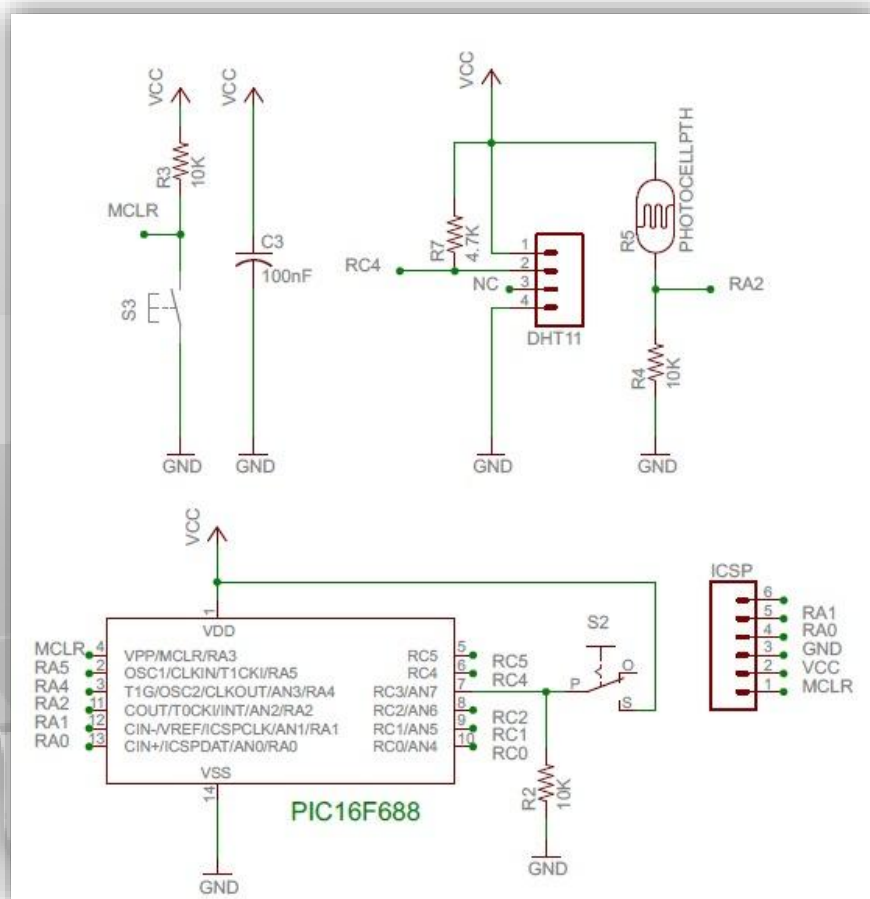


در مدار زیر ارتباط سنسور DHT11 و فتوسل (به آن مقاومت وابسته به نور یا LDR نیز گفته می شود) با میکرو کنترلر PIC16F688 نشان داده شده است .

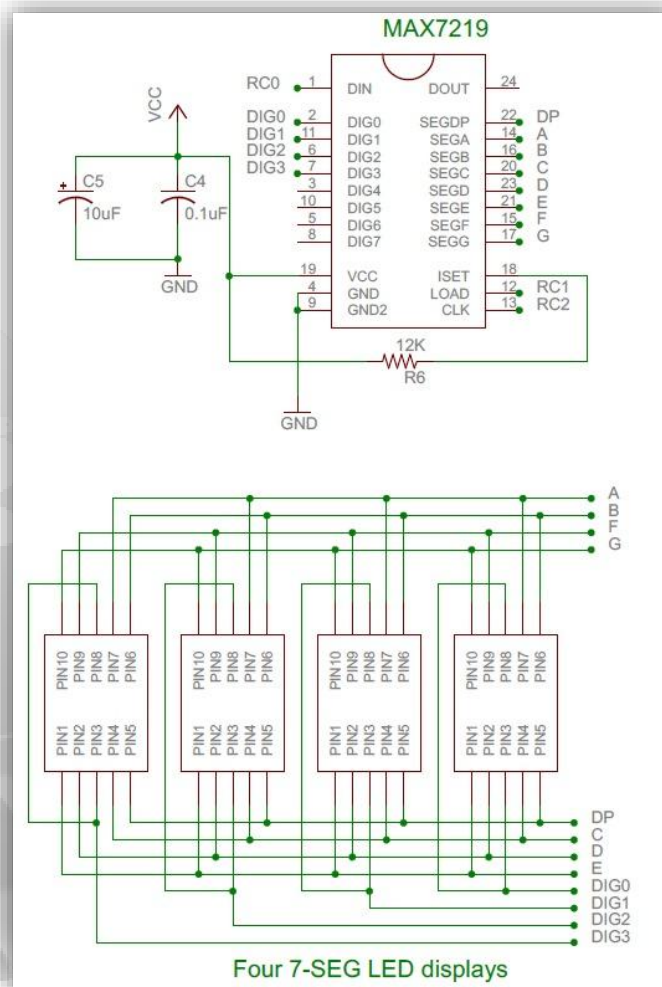
DHT11 یک سنسور دیجیتال کم هزینه برای اندازه گیری درجه حرارت $0-50^{\circ}\text{C}$ با دقت $\pm 2^{\circ}\text{C}$ و رطوبت نسبی در محدوده 20-95٪ با دقت $\pm 5\%$ است . خروجی سنسور بطور کامل کالیبره شده تا برای استفاده از پروتکل 1-Wire به نهایت بازده خود برسد کلید S2 برای تغییر حالت نمایش واحد دما بین درجه فارنهایت ($^{\circ}\text{F}$) و درجه سانتیگراد ($^{\circ}\text{C}$) است . در حالت پیش فرض که کلید باز است درجه فارنهایت است . برای تبدیل آن به سانتیگراد کلید را بالا بکشید .

با R5 (فتوسل) و R4 یک شبکه تقسیم ولتاژ را میسازیم مقدار ولتاژ در سراسر R4 متناسب با کم شدن نور در فتوسل تغییر می کند . مقاومت فتوسل (photoresistor) در شرایط نوری معمولی کمتر از $1\text{K}\Omega$ است و در تاریکی تا چند کیلو می تواند تغییر کند . بنابراین برای سنجش نور

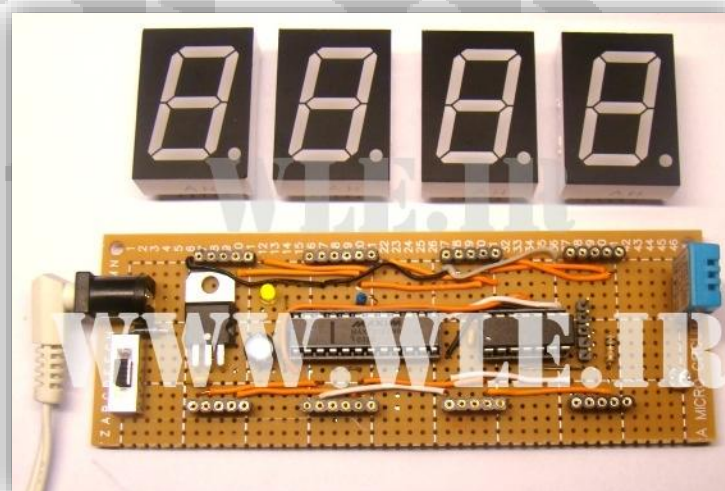
در مقاومت R4 در شرایط نور کم ولتاژ کمتر از 0.1 ولت و در شرایط روشنایی به بیش از 4.0 ولت تغییر می کند . با استفاده از adc میکرو این مقادیر خوانده می شوند و شرایط تشخیص داده می شود .



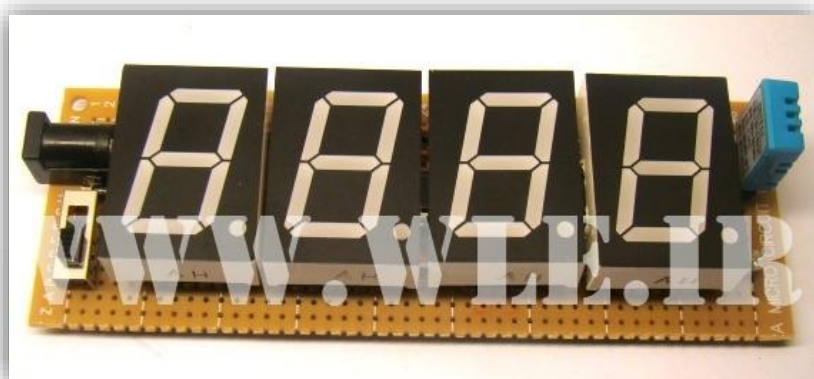
قسمت نمایشگر شامل تراشه MAX7219 است که میتواند تا 8 تا سون سگمنت (کاتد مشترک) را به تنهایی از طریق یک رابط سه سیم درایو کند . در حال حاضر برای ال ای دی همه ی سگمنت ها تنها از یک مقاومت خارجی بین پین ISET و منبع تغذیه استفاده شده است . دستگاه 16 مرحله را برای کنترل سطح روشنایی از طریق pwm میکرو فراهم می کند .



در شکل زیر مدار کامل را می بینید که بر روی برد سوراخ دار پیاده سازی شده است



شکل زیر هم که مدار کامل و آماده شده می باشد



برنامه نویسی :

برنامه نویسی برای میکروکنترلر PIC به زبان سی و با استفاده از کامپایلر mikroC ورژن 5.30 صورت گرفته است . میتونید این کامپایلر رو از [اینجا](#) دانلود کنید .

خطوط تعریف شده برای مقدار دهی به MAX7219 به حدی ساده و قابل فهم هستند که با هر زبان برنامه نویسی دیگری قابل اجرا هستند . این تراشه این قابلیت رو داره که میزان نور سگمنت ها را از طریق برنامه نویسی میکرو فراهم کند و لزومی به استفاده از سخت افزار جانبی نیست . برای خواندن مقدار آنالوگ فتوسل از adc ده بیتی (0-1023) میکرو استفاده شده است ولتاژ خروجی فتوسل با تقسیم بر 50 به 0-19 بخش کوچکتر تقسیم می شود . (20 مرحله روشنایی بین 0 تا 1023)

میکروکنترلر برای نمایش حرارت و رطوبت بطور مداوم بین این دو سویچ می کند . چند ثانیه دما و چند ثانیه رطوبت

این هم برنامه میکروکنترلر PIC به زبان سی :

```
// Define Soft-SPI connections for LED display
#define MOSI_Pin    RC0_bit
#define CS_Pin      RC1_bit
#define CLK_Pin     RC2_bit

// Define Data pin for DHT11
#define Data         RC4_bit
#define DataDir      TRISC4_bit
#define FC_Select    RC3_bit    // Fahrenheit or Celcius select

unsigned short TOUT = 0, CheckSum, i, check, u, NumSamples;
unsigned short T_Byte1, T_Byte2, RH_Byte1, RH_Byte2;
unsigned long int brightness=0x0b;
unsigned int counter = 0, TempF, TempC, RH, ADC_OP, TempSum, RHSum;
//Brightness look-up table
unsigned short Brightness_Table[21] = {0, 1, 2, 3, 3, 4, 4, 5, 5, 6, 7, 8, 9,
10, 11, 12, 12, 13, 14, 15, 15};
```

```
void SPI_Write_Byte(unsigned short num){
    unsigned short t, Mask, Flag;
    CLK_Pin = 0;
    Mask = 128;
    for (t=0; t<1; t++)
    {
        SPI_Write_Byte(num & Mask);
        Mask = Mask >> 1;
    }
}

void MAX7219_INIT() {
    // Disable Shutdown mode
    CS_Pin = 0; // CS pin is pulled LOW
    SPI_Write_Byte(0x0C); // Select Shutdown register
    SPI_Write_Byte(0x01); // Set D0 bit to return to normal operation
    CS_Pin = 1; // CS pin is pulled HIGH

    // Set BCD decode mode for digits DIG1-DIG3, and DIG5-DIG7
    CS_Pin = 0; // CS pin is pulled LOW
    SPI_Write_Byte(0x09); // Select Decode Mode register
    SPI_Write_Byte(0b00001100); // Disable BCD mode for digits DIG0, DIG1
    CS_Pin = 1; // CS pin is pulled HIGH
    // Set display brightness
    CS_Pin = 0; // CS pin is pulled LOW
    SPI_Write_Byte(0x0A); // Select Intensity register
    SPI_Write_Byte(brightness); // Set brightness
    CS_Pin = 1; // CS pin is pulled HIGH

    // Set display refresh
    CS_Pin = 0; // CS pin is pulled LOW
    SPI_Write_Byte(0x0B); // Select Scan-Limit register
    SPI_Write_Byte(0x03); // Select digits DIG0-DIG3
    CS_Pin = 1; // CS pin is pulled HIGH

    // Enable Display-Test
    CS_Pin = 0; // CS pin is pulled LOW
    SPI_Write_Byte(0x0F); // Select Display-Test register
    SPI_Write_Byte(0x01); // Enable Display-Test
    CS_Pin = 1; // CS pin is pulled HIGH
    Delay_ms(1000);
    // Disable Display-Test
    CS_Pin = 0; // CS pin is pulled LOW
    SPI_Write_Byte(0x0F); // Select Display-Test register
    SPI_Write_Byte(0x00); // Disable Display-Test
    CS_Pin = 1; // CS pin is pulled HIGH
}

void Display_Value(unsigned int j, unsigned short k){
    CS_Pin = 0; // CS pin is pulled LOW
    SPI_Write_Byte(4); // Send thousands digit
    SPI_Write_Byte((j/100)%10);
    CS_Pin = 1; // CS pin is pulled HIGH

    CS_Pin = 0; // CS pin is pulled LOW
    SPI_Write_Byte(3); // Send hundreds digit
    SPI_Write_Byte((j/10)%10);
    CS_Pin = 1; // CS pin is pulled HIGH
}
```

```
CS_Pin = 0; // CS pin is pulled LOW
SPI_Write_Byte(2); // Send tens digit
if(k 40) num |= 1<40us, Data is 1
}
return num;
}

void interrupt(){
    if(PIR1.TMR1IF){
        TOUT = 1;
        T1CON.TMR1ON = 0; // stop timer
        PIR1.TMR1IF = 0; // Clear TMR0 interrupt flag
    }
}

void Display_Brightness(){
    ADC_OP = ADC_Read(2);
    Brightness = ADC_OP/50;
    CS_Pin = 0; // CS pin is pulled LOW
    SPI_Write_Byte(0x0A); // Select Intensity register
    SPI_Write_Byte(Brightness_Table[brightness]); // Update brightness
    CS_Pin = 1; // CS pin is pulled HIGH
}

void Wait_Nsec(){
    Delay_ms(3000);
}

void main() {
    ANSEL = 0b00000100; // RA2/AN2 is analog input
    CMCON0 = 0x07; // Disbale comparators
    TRISC = 0b00001000; // PORTC All Outputs except RC3
    TRISA = 0b00001110; // PORTA All Outputs, Except RA3 and RA2

    INTCON.GIE = 1; //Enable global interrupt
    INTCON.PEIE = 1; //Enable peripheral interrupt
    // Configure Timer2 module
    PIE1.TMR1IE = 1; // Enable Timer2 interrupt
    T1CON = 0; // Prescaler 1:1, and Timer2 is off initially
    PIR1.TMR1IF = 0; // Clear TMR INT Flag bit
    TMR1H = 0;
    TMR1L = 0;
    MAX7219_INIT(); // initialize max7219
    if(!FC_Select) Display_Value(0, 0);
    if(FC_Select) Display_Value(0, 1);

    do{
        NumSamples = 0;
        TempSum = 0;
        RHSum = 0;
        for (u=0; u
```

سرعت میکرو را روی کریستال داخلی با سرعت 4 مگاهرتز قرار دهید . MCLR و Power-Up تایمر (PWRT) هر دو باید فعال باشند .
مدار کاملاً تست شده می باشد و در شرایط بسیار روشن و شرایط بسیار تاریک بخوبی جواب می دهد در تاریکی مطلق نورش بسیار کم میشود و
اصلاً چشم را اذیت نمی کند و از طرف دیگر اعداد آن به راحتی قابل خواندن هستند .



وب سایت الکترونیک و رباتیک

WWW.WLE.IR