



دانشکده مهندسی مکانیک
پردیس دانشکده‌های فنی
دانشگاه تهران

گزارش تکالیف

جلسه اول

درس سیستم‌های اندازه‌گیری کارشناسی

دکتر صدیقی

مهدی عبدالله چالکی (۸۱۰۶۹۶۲۶۸)

نیم‌سال دوم

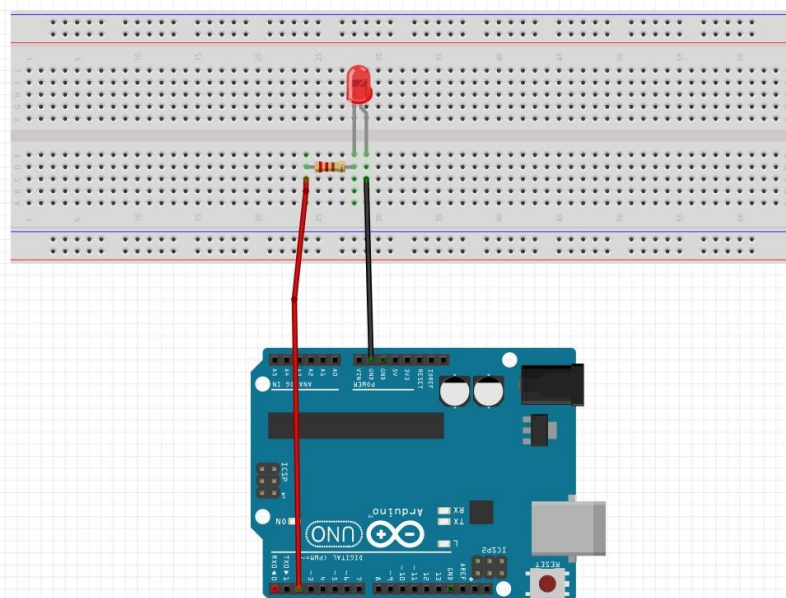
سال تحصیلی ۹۹-۰۰

۱- تکلیف اول

در ابتدا، خواسته‌ی هر تکلیف را به صورت خلاصه توضیح دهید. سپس روند انجام آن و رسیدن به پاسخ موردنظر را مفصلاً شرح دهید. در انتها موارد تحویلی تکلیف را ارائه دهید (تنها در صورتی که تکلیف موردنظر موارد تحویلی داشته باشد مثل نمودار داده‌ها و غیره). مدارهای الکترونیکی را ترجیحاً با استفاده از نرم‌افزار فریتزینگ^۱ طراحی کنید و تصویر آن‌را در گزارش قرار دهید.

کدهای مربوط به آردوینو^۲ و متلب^۳ (به همراه توضیحات موجود در کد) را در قسمت ضمائم بیاورید. در صورتی که در ویدئوی ضبط‌شده مربوط به هر جلسه، سوالاتی پرسیده شده باشند، پاسخ آن‌ها را بعد از تکلیف در قسمت «پاسخ به سوالات» شرح دهید.

در تکلیف اول، هدف این است که یک دیود نوری (LED) را طبق برنامه زمانی، خاموش و روشن کنیم. به این منظور، در مدار طراحی شده از یک لامپ ال ای دی، یک مقاومت (برای محدود کردن جریان عبوری) و سیم‌های جامپر برای اتصال قطعات استفاده می‌شود.



همان طور که در تصویر مشاهده می‌شود، سمت منفی ال ای دی به زمین مدار وصل شده و پایه مثبت نیز از طریق یک مقاومت، به پین شماره ۲ آردوینو متصل است. این پایه را در ابتدای کد، به عنوان خروجی تعریف می‌کنیم.

سپس برای خاموش و روشن کردن ال ای دی، به این صورت عمل شده است که ولتاژ پین شماره ۲ به مدت یک ثانیه high و یک ثانیه low باشد. زمان یک ثانیه نیز از طریق دستور delay ایجاد می‌شود. در نتیجه، با اتصال مدار به منبع تغذیه (در اینجا USB کامپیوتر) چراغ به صورت چشمک زن عمل خواهد کرد.

¹ Fritzing

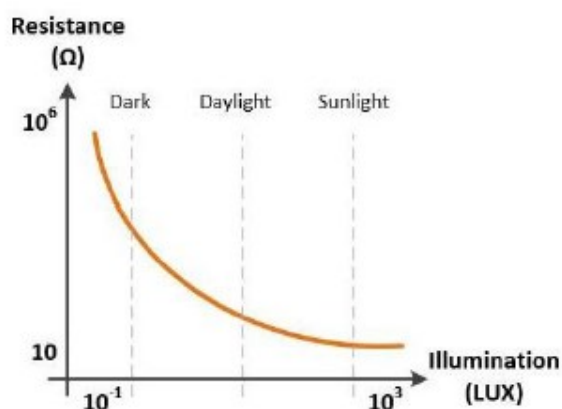
² Arduino

³ Matlab

۲- تکلیف دوم

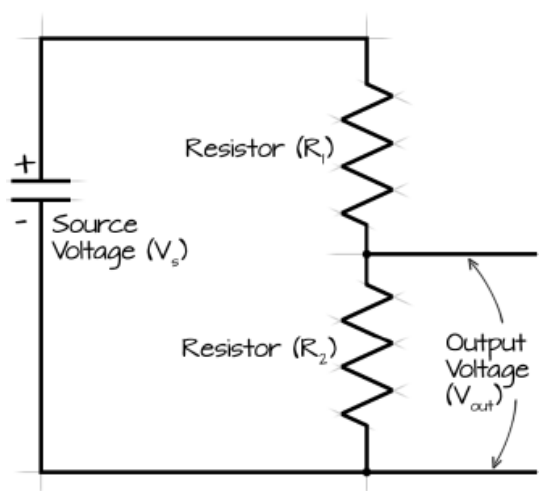
هدف تکلیف دوم، کنترل روشنایی یک لامپ به کمک حسگر فوتورزیستور است. بدین صورت که اگر نور محیط کم بود، فوتوسل روشن تر شده و اگر محیط روشن شد، لامپ کم نورتر شود.

فوتورزیستور دارای ساختاری است که با تابیدن نور بر روی آن، مقاومتش کم می‌شود.



برای بدست آوردن مقدار لحظه‌ای این مقاوت، از یک مدار تقسیم کننده ولتاژ و یک مقاومت معلوم کمک می‌گیریم.

$$\frac{V_{out}}{V_s} = \frac{R_2}{R_1 + R_2}$$



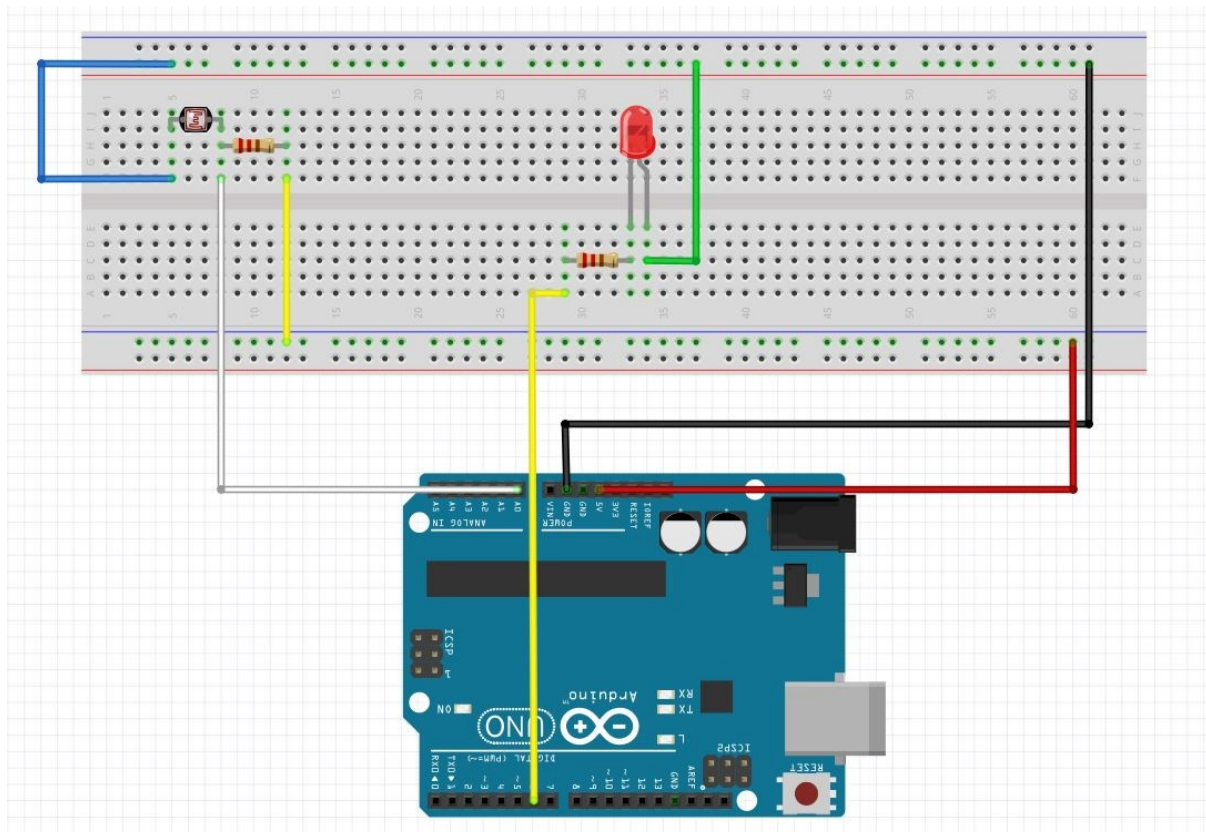
بدین ترتیب، مقاومت R_2 از رابطی زیر بدست می‌آید:

$$R_2 = \frac{R_1 * V_0}{V_s - V_0}$$

با توجه به استفاده از مقاومت ۲۲۰ اهم و ولتاژ منبع ۵ ولت، داریم:

$$R_2 = \frac{220 * V_0}{5 - V_0}$$

بدین ترتیب، نیازمند اندازه گیری ولتاژ V_0 هستیم. اینکار به کمک خواندن ولتاژ پایه مشترک مقاومت و فوتورزیستور انجام می‌گیرد.



سیم سفید در شکل بالا، به پایه A0 وصل شده و مقدار آن خوانده می‌شود. سپس مقدار آن از مقیاس ۰ تا ۱۰۲۳، به ۵ ولت تصویر می‌شود. بنابراین، با استفاده از رابطه بالا می‌توانیم مقدار مقاومت را تعیین کنیم. در بخش بعدی، یک جدول داریم که باید به ازای قرار گرفتن مقدار مقاومت در میان هر دو عدد آن، مقدار روشنایی در آن بازه درونیایی شده و اعلام شود. برای این کار، از دستورات if و else کمک گرفتیم.

پس از تعیین شدن مقدار روشنایی محیط، نوبت به کنترل نور ال ای دی می‌رسد. مجدداً ال ای دی را مانند بخش قبلی با یک مقاومت به پایه ۶ وصل کرده و سمت منفی نیز به زمین متصل می‌شود. سپس ولتاژ خوانده شده توسط A0، به ۰ تا ۲۵۵ تصویر شده و نور کنترل می‌شود.

نکته آخر هم اینکه برای کم کردن مقدار نویز، ابتدا به فاصله هر ۱۰ میلی ثانیه یک بار، یک سمپل گرفته می‌شود. ۲۰ تا سمپل با هم جمع شده و میانگین آن‌ها در نهایت مقدار خروجی را می‌سازد.

First Code:

```
// digital pin 2 has an LED attached to it.
int led = 2;

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second
  Serial.begin(9600);
  // make the LED pin an output:
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);  // Turns the LED on
  Serial.println(1);        // Prints out the state of the LED
  delay(1000);              // Waits for a second

  digitalWrite(led, LOW);   // Turns the LED off
  Serial.println(0);        // Prints out the state of the LED
  delay(1000);              // Waits for a second

}
```

Second Code:

```
#define NUM_SAMPLES 20          // Number of samples used for averaging

int LDR = A0;                   // Pin A0 has an LDR attached to it.
int led_pin = 6;                // Digital pin 6 has an LED attached to it.

int sum = 0;                    // sum of samples taken
unsigned char sample_count = 0; // current sample number
float V_out = 0.0;              // calculated V_out
float Resistance = 0.0;         // calculated Resistance
float Illumination = 0.0;       // calculated Illumination
float LED;                      // output to be sent to the LED

void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  pinMode(led_pin, OUTPUT);      // Define led_pin as an output
}

void loop() {

  // This loop is used for averaging and decreasing the noise
  while (sample_count < NUM_SAMPLES) {
    sum += analogRead(LDR);
    sample_count++;
    delay(10);
  }

  // Map output voltage to 0-5 scale
  V_out = ((float)sum / (float)NUM_SAMPLES * 5.0) / 1024.0;
```

```

// Calculate Resistance of the LDR
Resistance = (((float)V_out*220.0)/(5.0-V_out));

// Map output voltage to 0-255 scale to control LED light
LED = V_out * 255.0 / 1023.0;
analogWrite(led_pin, LED);

// Reset values
sample_count = 0;
sum = 0;

// Look up for the right category
if (Resistance<1.5e7 && Resistance>1e6) {
    Illumination = interp(1e6,1.5e7,0.1,0.001,Resistance);
    Serial.print(Illumination);
    Serial.println (" LUX");
}

else if (Resistance<1e6 && Resistance>7.5e4) {
    Illumination = interp(7.5e4,1e6,35,0.1,Resistance);
    Serial.print(Illumination);
    Serial.println (" LUX");
}

else if (Resistance<7.5e4 && Resistance>3e4) {
    Illumination = interp(3e4,7.5e4,72,35,Resistance);
    Serial.print(Illumination);
    Serial.println (" LUX");
}

```



```

else if (Resistance<3e4 && Resistance>1.7e4) {
    Illumination = interp(1.7e4,3e4,157,72,Resistance);
    Serial.print(Illumination);
    Serial.println (" LUX");
}

else if (Resistance<1.7e4 && Resistance>1.5e4) {
    Illumination = interp(1.5e4,1.7e4,200,157,Resistance);
    Serial.print(Illumination);
    Serial.println (" LUX");
}

else if (Resistance<1.5e4 && Resistance>3.8e3) {
    Illumination = interp(3.8e3,1.5e4,1380,200,Resistance);
    Serial.print(Illumination);
    Serial.println (" LUX");
}

else if (Resistance<3.8e3 && Resistance>2.8e3) {
    Illumination = interp(2.8e3,3.8e3,3700,1380,Resistance);
    Serial.print(Illumination);
    Serial.println (" LUX");
}

else if (Resistance<2.8e3 && Resistance>1.4e3) {
    Illumination = interp(1.4e3,2.8e3,7300,3700,Resistance);
    Serial.print(Illumination);
    Serial.println (" LUX");
}

```

```

else if (Resistance<1.4e3 && Resistance>47) {
    Illumination = interp(47,1.4e3,2.3e4,7300,Resistance);
    Serial.print(Illumination);
    Serial.println (" LUX");
}

else if (Resistance<47 && Resistance>10) {
    Illumination = interp(10,47,1e5,2.3e4,Resistance);
    Serial.print(Illumination);
    Serial.println (" LUX");
}

else {
    Illumination = 0;
    Serial.print(Illumination);
    Serial.println (" LUX");
}
}

// Float interpolation function
float interp(float A_1, float A_2, float B_1, float B_2, float x){
    return ((x-A_1)/(A_2-A_1))*(B_2-B_1)+B_1;
}

```