

پرسش‌ها

- ۱- از فرکانس 2.4 GHz استفاده می‌کند. این فرکانس را دیوایس‌های دیگر نیز می‌توانند استفاده کنند، اما چون سیگنال‌های بلوتوث سیگنال‌های ضعیفی نسبت به برای مثال تلفن همراه هستند، در عمل احتمال اینکه تداخل پیش بیاید کم است و همچنین بلوتوث‌های امروزی نیز از spread-spectrum frequency hopping استفاده می‌کنند که به این معنی است که 1600 بار در یک ثانیه بین 70 فرکانس رندوم در بازه‌ی مجاز خودشان جا به جا می‌شوند و به همین دلیل امکان تداخل کم می‌شود و حتی اگر تداخلی هم به وجود بیاید خیلی کم طول خواهد کشید.
 - ۲- $12c$ به این شکل ساخته شده است که هر زمان کسی بخواهد روی این پورت چیزی بنویسد باید اول کلاک را ایجاد کند و سپس یک استارت بیت روی خط دیتا بفرستد. اگر در زمانی که این دیتا دارد فرستاده می‌شود یک سنسور دیگر بخواهد دیتا بفرستد، باید منتظر stop bit بماند. در واقع با دیدن استاپ بیت بقیه سنسورها می‌فهمند که خط آزاد شده، پس هیچ وقت تداخل داده نخواهیم داشت.
 - ۳- موتورهای DC (جریان مستقیم)، Stepper و Servo به عنوان سه نوع از موتورهای الکتریکی پرکاربرد شناخته می‌شوند. هر یک از این سه نوع موتور برای کاربردهای خاص خودشان مناسب هستند و قابلیت‌ها و مزایای خاصی دارند.
- موتور DC:
- موتورهای DC از جمله پرکاربردترین نوع موتورهای الکتریکی هستند که برای بسیاری از کاربردها مورد استفاده قرار می‌گیرند. این موتورها با اعمال جریان DC، یک میدان مغناطیسی ایجاد می‌کنند که باعث چرخش شفت موتور می‌شود. مزیت اصلی موتورهای DC، سادگی و قابلیت کنترل بهتر است که این موتورها را برای کاربردهای متنوعی از جمله موتورهای خودرو، پمپ‌ها، فن‌ها و ... مورد استفاده قرار می‌دهند.

• موتور Stepper:

موتورهای Stepper با حرکت قطعه به قطعه ای عمل می کنند و برای بسیاری از کاربردهای دقیق از جمله پرینترها، دستگاه های CNC و ربات های صنعتی استفاده می شوند. این موتورها از یک تراشه کنترلر استفاده می کنند و تعداد دقیقی از گام ها را برای چرخش موتور تعیین می کنند. این قابلیت باعث می شود تا موتورهای Stepper برای کاربردهایی که نیاز به حرکت دقیق دارند، مانند تولید قطعات صنعتی و تست کالیبراسیون، بسیار مناسب باشند.

• موتور Servo:

موتورهای Servo با سیستم بازخورد خودکار که به عنوان یک حسگر موقعیت عمل می کند، کنترل موقعیت و سرعت دقیق را فراهم می کنند. به عنوان نمونه

مفاهیم مقدماتی این تمرین:

رابط آردوینو:

- آردوینو یک پلتفرم سخت افزاری و نرم افزاری متن باز است که برای پروژه های الکترونیکی طراحی شده است. این شامل یک برد میکروکنترلر و یک محیط توسعه یکپارچه (IDE) است که روی یک کامپیوتر اجرا می شود. برد آردوینو به عنوان رابط بین کامپیوتر و سخت افزار خارجی مانند سنسورها، موتورها و سایر قطعات الکترونیکی عمل می کند. برد آردوینو را می توان از طریق کابل USB به کامپیوتر متصل کرد که به IDE اجازه می دهد کد را آپلود کند و با برد ارتباط برقرار کند. IDE یک زبان برنامه نویسی ساده بر اساس C++ ارائه می دهد که به کاربران اجازه می دهد به راحتی کد بنویسند و روی برد آپلود کنند.

این برد همچنین دارای پایه هایی است که می توان از آنها برای اتصال به قطعات سخت افزاری خارجی استفاده کرد و این پین ها را می توان برای انجام عملکردهای خاص برنامه ریزی کرد. به عنوان مثال، یک پین را می توان برای خواندن ورودی از یک سنسور یا برای خروج ولتاژ برای کنترل موتور برنامه ریزی کرد.

به طور خلاصه، رابط آردوینو به روشی اطلاق می شود که برد آردوینو به کامپیوتر و اجزای سخت افزاری خارجی متصل می شود و به کاربران اجازه می دهد تا از طریق برنامه نویسی نرم افزاری، دستگاه های الکترونیکی را کنترل و با آنها تعامل داشته باشند.

پروتکل های ارتباط موازی :

- **UART، I2C و RS232** همگی پروتکل های ارتباطی سریال هستند که برای انتقال داده ها بین دستگاه ها استفاده می شوند. با این حال، آنها از نظر مشخصات، کاربردها و سازگاری متفاوت هستند.
- **UART (Universal Asynchronous Receiver-Transmitter)** یک پروتکل ارتباطی سریالی ساده و پرکاربرد است که داده ها را در یک جریان پیوسته از بیت ها بدون هیچ سیگنال همگام سازی ارسال و دریافت می کند. از یک خط داده واحد برای انتقال و دریافت استفاده می کند و هیچ مکانیزم تشخیص یا تصحیح خطا را ارائه نمی دهد. UART معمولاً در برنامه های کم سرعت مانند رابط با سنسورها، مودم ها و سایر دستگاه های تعبیه شده استفاده می شود.
- **I2C (Inter-Integrated Circuit)** یک پروتکل ارتباطی سریال همزمان چند استاد و چند بردی است که از دو خط - **SDA** (خط داده) و **SCL** (خط ساعت) - برای ارتباط استفاده می کند. این به چندین دستگاه اجازه می دهد تا در یک اتوبوس با هم ارتباط برقرار کنند و مکانیزمی برای تشخیص خطاها و ارسال مجدد داده ها در صورت لزوم فراهم می کند. I2C معمولاً در برنامه های با سرعت کم تا متوسط مانند برقراری ارتباط با حسگرها، دستگاه های حافظه و سایر تجهیزات جانبی استفاده می شود.

- **RS232** (استاندارد توصیه شده ۲۳۲) یک پروتکل ارتباطی سریال قدیمی است که از یک سیگنال ولتاژ تک سر برای انتقال داده ها بین دو دستگاه استفاده می کند. حداکثر سرعت داده ۱۱۵.۲ کیلوبیت بر ثانیه را پشتیبانی می کند و مکانیزمی برای تشخیص و تصحیح خطا فراهم می کند. **RS232** معمولاً در کاربردهای صنعتی و ارتباطی مانند اتصال رایانه به مودم یا چاپگر سریال استفاده می شود.

- به طور خلاصه، **UART** یک پروتکل ساده و کم سرعت است، **I2C** یک پروتکل پیچیده تر و چند دستگاهی است، و **RS232** یک پروتکل قدیمی است که معمولاً در برنامه های صنعتی و ارتباطی استفاده می شود.

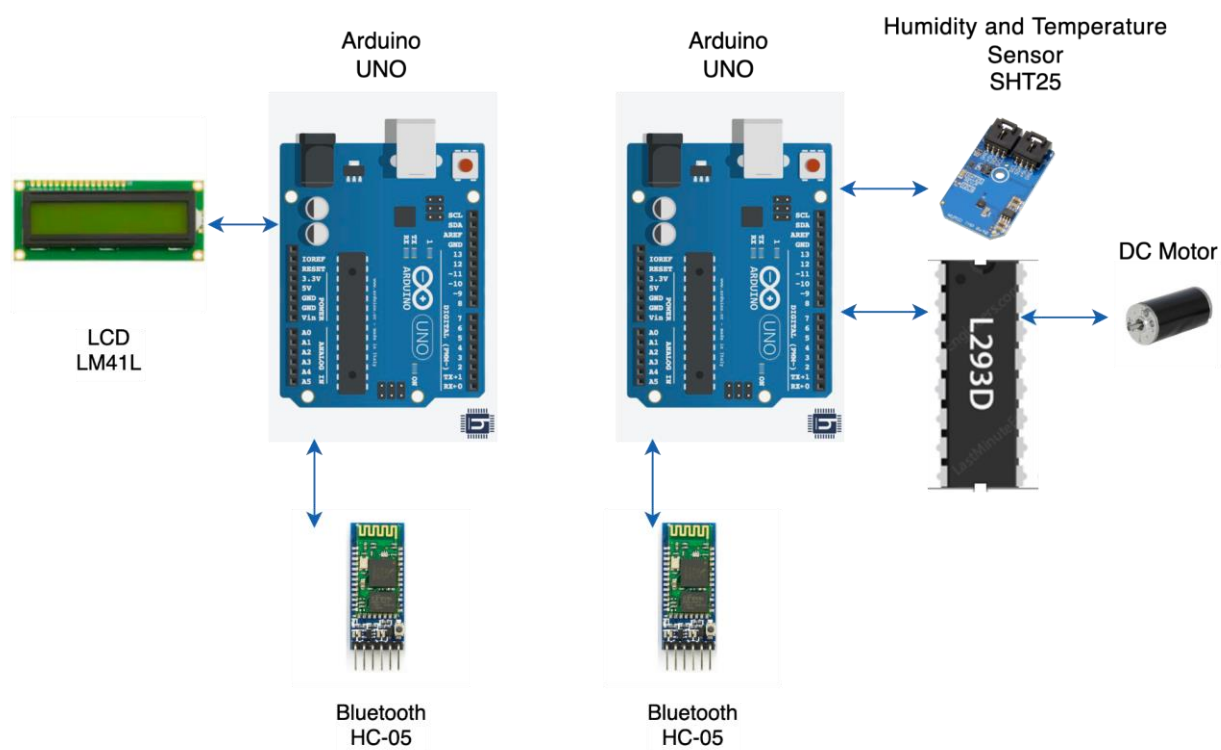
درباره Arduino UNO :

- **Arduino Uno** یک برد میکروکنترلر محبوب است که در طیف گسترده ای از پروژه های الکترونیکی استفاده می شود. این دستگاه بر اساس میکروکنترلر **ATmega328P** ساخته شده است و دارای ۱۴ پین ورودی/خروجی دیجیتال، ۶ پایه ورودی آنالوگ و سایر اجزای ضروری از جمله نوسانگر کریستالی، اتصال **USB** و جک برق است.

- **GPIO** مخفف عبارت **General Purpose Input/Output** است و به پین های دیجیتالی روی برد آردوینو اشاره دارد که می توانند به عنوان ورودی یا خروجی پیکربندی شوند. از این پین ها می توان برای خواندن سیگنال های دیجیتال از حسگرها یا سوئیچ ها، کنترل **LED** ها یا سایر خروجی ها یا برقراری ارتباط با دستگاه های دیگر از طریق پروتکل هایی مانند **I2C**، **SPI** یا **UART** استفاده کرد.

- **ADC** مخفف عبارت **Analog to Digital Converter** است و به پین های ورودی آنالوگ روی برد آردوینو اشاره دارد که می تواند سیگنال های آنالوگ را به مقادیر دیجیتالی تبدیل کند که توسط میکروکنترلر قابل خواندن و پردازش باشد. **Arduino Uno** دارای ۶ پایه ورودی آنالوگ با برچسب **A0** تا **A5** است که می تواند ولتاژ ۰ تا ۵ ولت را با وضوح ۱۰ بیت (یعنی ۱۰۲۴ مقدار ممکن) اندازه گیری کند. این به شما امکان می دهد با سنسورهای آنالوگ مختلف مانند سنسورهای دما، نور یا صدا ارتباط برقرار کنید.

- به طور خلاصه، پورت‌های GPIO و ADC در آردوینو Uno یک راه انعطاف‌پذیر و همه‌کاره برای ارتباط با طیف گسترده‌ای از قطعات و دستگاه‌های الکترونیکی ارائه می‌کنند که آن را به یک پلت فرم ایده‌آل برای نمونه‌سازی و آزمایش پروژه‌های الکترونیکی تبدیل می‌کند.



• توضیحات مربوط به بخش کد گره مرکزی

*در بخش گره مرکزی ابتدا مطابق کد زیر یک lcd تعریف می کنیم و به پورت های آردوینو وصل می کنیم.

کد بخش ساخت LCD

```
////create LCD  
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;  
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

*آردوینو مرکزی دارای یک بلوتوث هست که با کتابخانه SoftwareSerial یک بلوتوث که به پورت ۰ و ۱ آردوینو وصل است می خوانیم.

کد بخش ساخت بلوتوث

```
SoftwareSerial blueMain(rx, tx); //RX, TX
```

*در تصویر بالا rx به پورت ۰ و tx به پورت ۱ متصل است.

حال بعد از راه اندازی بلوتوث و lcd باید داده ای که از گره حسگر می آید را بخواند. از سمت گره حسگر ابتدا با علامت * شروع ارسال داده را مشخص می کنیم دما و ارسال می کنیم. سپس از علامت - برای جدا سازی دما و رطوبت استفاده می کنیم. بعد از آن رطوبت را قرار می دهد. در آخر برای اینکه متوجه شویم که یک رشته داده ای که می خواستیم تمام شده است، علامت % را قرار می دهیم. داده ارسال شده دارای ۱۳ بیت می باشد که هر عدد مربوط به دما و رطوبت دو بیت صحیح و اعشاری دارد. یک بن برای جدا سازی عدد اعشاری نیز داریم. دو بیت هم برای * و % که چمعا ۱۳ بیت می شود. در while که داریم ۱۳ کاراکتر را از serial زمانی که داده موجود باشد می خوانیم.

کد بخش دریافت داده

```
void readDataSensor() {  
  char buff[13];  
  //Get data from TH
```

```
byte i=0;
while (i < 13) {
    if (Serial.available() > 0) {
        buff[i] = Serial.read();
        i++; //Only increment after something was received
    }
}
int j = 0;
while (j < 13){
    Serial.print(buff[j]);
    j++;
}
Serial.println("");

char temp[10];
char humadity[10];
if (buff[0] == '*' && buff [12] == '%') {
    tempValue = atof(strncpy(temp, buff+1, 5));
    humidityValue = atof(strncpy(humadity, buff+7, 5));
    processData();
}
}
```

*زمانی که داده را گره مرکزی دریافت کرد در تابع `createOutput` بر اساس ورودی که دریافت کرده است باید خروجی را برای ارسال به گره عملگر آماده کند. ابتدا در کد زیر بررسی می کنیم که اگر رطوبت بیشتر از ۳۰ بود، `stop` بدهد، اگر کمتر از ۱۰ بود باید `duty` ۲۵ داشته باشد که به این منظور `string` با مقدار ۲۵ را ارسال می کنیم و اگر بین ۲۰ تا ۳۰ بود رشته با مقدار ۱۰ را ارسال می کنیم. برای مقادیر ارسالی به صورت زیر `define` کرده ایم:

```
#define STOP "*###%"
#define DUTY10 "*10%"
```

```
#define DUTY25 "*25%"
```

کد بخش تولید رشته خروجی

```
void createOutput() {  
    if(humidityValue > 30) {  
        data = STOP;  
    }  
    else if(humidityValue < 20){  
        data = DUTY25;  
    }  
    else if((20 <= humidityValue) && (humidityValue <= 30)){  
        if(tempValue < 25){  
            data = STOP;  
        }  
    }  
    else if((10 <= humidityValue) && (humidityValue <= 20)){  
        data = DUTY20;  
    }  
    else {  
        data = DUTY10;  
    }  
}
```

*بعد از ساخت رشته خروجی مقدار رطوبت، دما و پیغام خروجی را به کمک `print` توسط بلوتوث ارسال می‌کنیم

کد بخش ارسال رشته خروجی

```
void sendData() {  
    Serial.print(data);  
}
```


به نام خدا
تمرین اول cps

کیمیا فخاری کیسمی ۸۱۰۱۹۷۶۵۰
محمدحسین عطائی ۸۱۰۱۹۷۶۳۲
دانیال سعیدی ۸۱۰۱۹۸۵۷۱
محمد قره حسنلو ۸۱۰۱۹۸۴۶۱

*ابتدا دما، رطوبت و تصمیم درباره ی آبیاری را بر روی lcd چاپ می کنیم.

کد بخش چاپ بر روی lcd

```
void LCDShowSensorValues() {  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.println(String(tempValue));  
    lcd.println( String(humidityValue));  
    lcd.println(data);  
}
```

*در کد زیر ابتدا روی lcd دما، رطوبت و خروجی محاسبه شده را نشان می دهد.

خروجی روی ترمینال سمت چپ نشان می دهد که دما ۲۹.۹۸ و رطوبت ۲۰.۴۲ ارسال شده است و در سمت راست مقدار ۱۰٪ را گرفته است که از گره مرکزی آمده است و نشان می دهد که duty cycle باید ۱۰ باشد.

*۲۹.۹۸-۲۰.۴۲٪

*۱۰٪

به نام خدا
تمرین اول cps

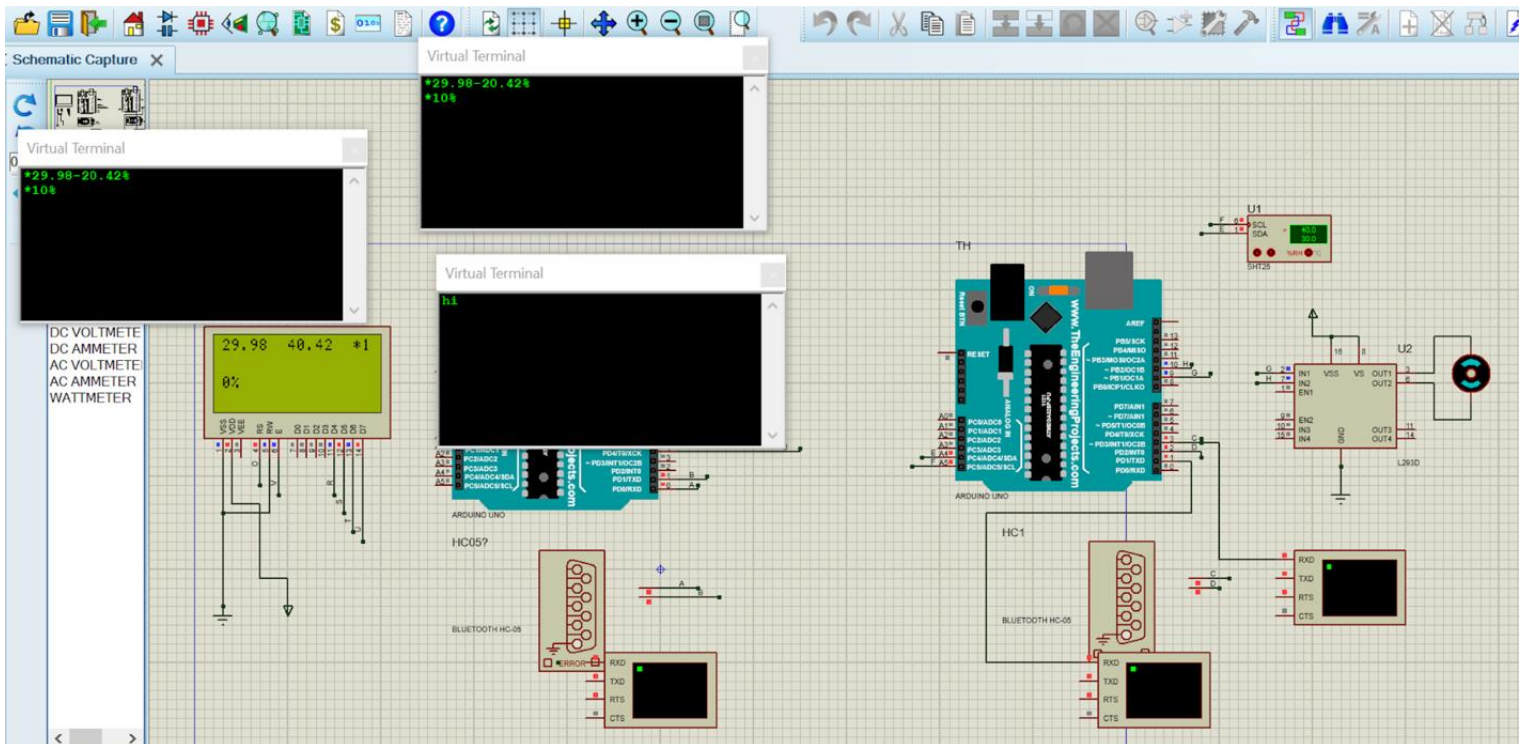
کیما فخاری کیسمی ۸۱۰۱۹۷۶۵۰

محمدحسین عطائی ۸۱۰۱۹۷۶۳۲

دانیال سعیدی ۸۱۰۱۹۸۵۷۱

محمد قره حسنلو ۸۱۰۱۹۸۴۶۱

خروجی چاپ روی lcd



به نام خدا
تمرین اول cps

کیمیا فخاری کیسمی ۸۱۰۱۹۷۶۵۰
محمدحسین عطائی ۸۱۰۱۹۷۶۳۲
دانیال سعیدی ۸۱۰۱۹۸۵۷۱
محمد قره حسنلو ۸۱۰۱۹۸۴۶۱

● توضیحات مربوط به بخش کد گرہ حسگر و عملگر

کد بخش بلوتوث

```
SoftwareSerial blueTH(rx,tx); //RX, TX
```

```
#define tx 2
```

```
#define rx 3
```

*برای دریافت داده از سنسور ابتدا برای هم رطوبت و هم دما باید **transmission** با سنسور داشته باشیم. از آنجایی که ارتباط آردوینو با سنسور از نوع **I2C** است از **wire** باید استفاده کنیم.

```
void sensorTransmission(int vale) {  
  Wire.beginTransmission(Addr);  
  Wire.write(vale);  
  Wire.endTransmission();  
  delay(500);  
}
```

*بعد از مرحله **transmission** داده را در ارایه می خوانیم و با فرمول

```
((data[0] * 256.0 + data[1]) * 125.0) / 65536.0 - 6;
```

به عدد رطوبت تبدیل می شود.

کد بخش دریافت رطوبت از سنسور

```
float getHumidityValue() {  
  sensorTransmission(humidityCMD);  
  //request  
  unsigned int data[2];  
  Wire.requestFrom(Addr, 2);  
  if(Wire.available() == 2)
```

```
{  
    data[0] = Wire.read();  
    data[1] = Wire.read();  
}  
return (((data[0] * 256.0 + data[1]) * 125.0) / 65536.0) - 6;  
}
```

*بعد از مرحله transmission داده را در ارایه می خوانیم و با فرمول

```
return (((data[0] * 256.0 + data[1]) * 175.72) / 65536.0) - 46.85;
```

به عدد دما تبدیل می شود.

کد بخش دریافت دما از سنسور

```
float getTempValue() {  
    sensorTransmission(tempCMD);  
    //request  
    unsigned int data[2];  
    Wire.requestFrom(Addr, 2);  
    if(Wire.available() == 2)  
    {  
        data[0] = Wire.read();  
        data[1] = Wire.read();  
    }  
    return (((data[0] * 256.0 + data[1]) * 175.72) / 65536.0) - 46.85;
```

● ابتدا * بعد دما با دو رقم اعشار و سپس - برای جدا سازی قرار می دهیم. بعد از - رطوبت و در آخر % را قرار

می دهیم.

کد بخش ارسال داده به گره مرکزی

```
void sendData (float temp,float humidity) {  
    blueTH.print("*" + String(temp, 2) + "-" + String(humidity, 2) + "%");
```

به نام خدا
تمرین اول cps

کیمیا فخاری کیسمی ۸۱۰۱۹۷۶۵۰
محمدحسین عطائی ۸۱۰۱۹۷۶۳۲
دانیال سعیدی ۸۱۰۱۹۸۵۷۱
محمد قره حسنیلو ۸۱۰۱۹۸۴۶۱

}

*داده ای که از سمت مرکزی می‌آید ۴ کاراکتر است که سر آن *و ته آن % وجود دارد. این مقدار ارسالی را از بلوتوث می‌خوانیم.

کد بخش دریافت داده از گره مرکزی

```
String readResponseData() {  
    char buff[4];  
    byte i=0;  
    while (i < 4) {  
        buff[i] = blueTH.read();  
    }  
}
```

*برای ارسال **duty cycle** به ماژول L293D از تابع **analogwrite** استفاده است. ورودی این ماژول به پورت ۹ و ۱۰ آردوینو وصل است. اگر **duty cycle** برابر ۱۰ باشد، مقدار ۲۵.۴ را ارسال می‌کنیم و اگر **duty cycle** برابر ۲۵ باشد مقدار ۶۳.۵ ارسال می‌شود. در صورتی هم که دستور **stop** آمده باشد مقدار ۰ را می‌فرستیم. Out ۱ پورت ۹ آردوینو می‌باشد.

کد بخش **duty cycle** برای DC

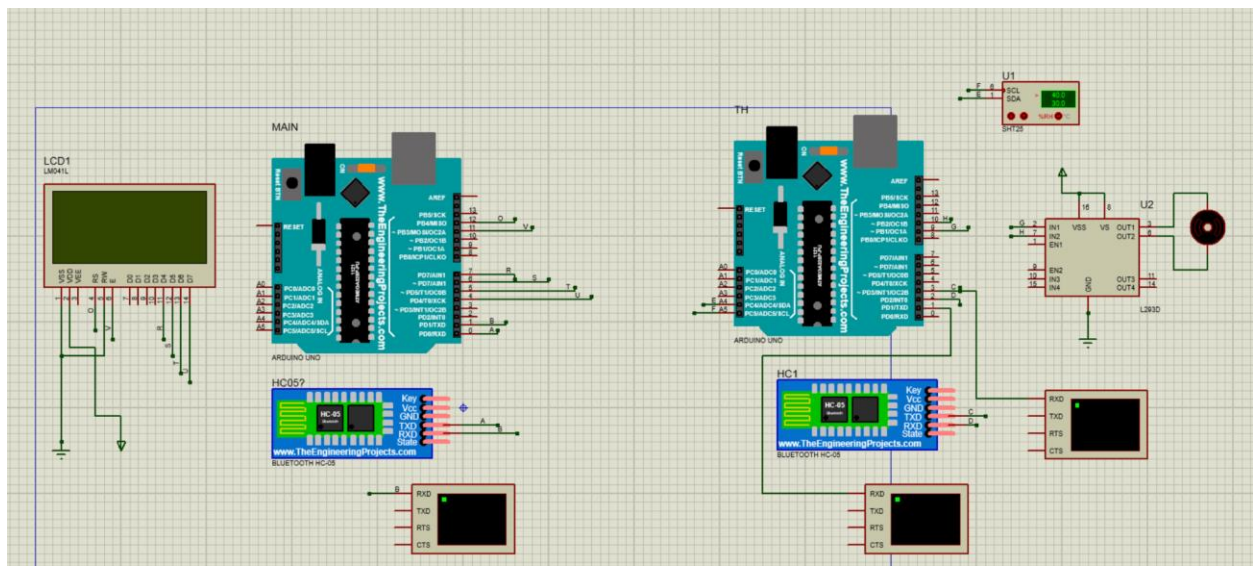
```
void sendDataToDC(String response){  
    Serial.print("in dc");  
    if(response == STOP) {  
        analogWrite(out1, 0);  
    }else if(response == DUTY10){  
        analogWrite(out1, 25.4);  
    }else if(response == DUTY25) {  
        analogWrite(out1, 63.5);  
    }
```

به نام خدا
تمرین اول cps

کیما فخاری کیسمی ۸۱۰۱۹۷۶۵۰
محمدحسین عطائی ۸۱۰۱۹۷۶۳۲
دانیال سعیدی ۸۱۰۱۹۸۵۷۱
محمد قره حسنلو ۸۱۰۱۹۸۴۶۱

```
}  
}
```

• تصویر ماژول های متصل شده به یکدیگر در proteus



برد TH :

هدف در این برد این است که اطلاعات مربوط به دما و رطوبت از سنسور گرفته شود و آنها را از طریق بلوتوث به برد اصلی ارسال کند. همچنین موتور DC نیز به این برد متصل است تا بعد از دریافت دستور لازم توسط برد اصلی به این برد، موتور حرکت لازم را انجام دهد. این برد از ماژول های SHT ۲۵، موتور DC و ماژول بلوتوث

HC-۰۵ تشکیل شده است. همچنین از ترمینال مجازی برای نمایش اطلاعات استفاده شده است. این برد از طریق پورت های ۹ و ۱۰ به موتور و از طریق دو پورت ۴A و ۵A به سنسور متصل شده است.

ماژول SHT۲۵ :

این ماژول مربوط به سنسور برای اندازه گیری دما و رطوبت است که از طریق ارتباط I2C اطلاعات را به برد ارسال می کند.

ماژول HC-۰۵ :

این ماژول مربوط به انتقال داده توسط بلوتوث است. یعنی با دریافت داده از برد اصلی و همچنین از طریق اتصال به بلوتوث مربوط به برد دیگر اطلاعات و داده ها را بین این دو ماژول و برد جا به جا می کند. در بلوتوث مربوط به برد اصلی پورت ۱ برد به RXD ماژول و پورت ۰ به TXD ماژول متصل است. در برد دیگر پورت ۳ به TXD و پورت ۲ به RXD متصل شده است. برای بلوتوث ها و اتصال به آنها از softwareSerial استفاده شده است.

برد اصلی: main

این برد اطلاعات مربوط به دما و رطوبت را از برد TH دریافت کرده و با پردازش و بررسی این اطلاعات دستور خروجی را مشخص کرده و همچنین نمایش این اطلاعات بر روی صفحه نمایش LCD را نیز انجام می دهد. این ماژول نیز به یک بلوتوث برای برقراری ارتباط با برد دیگر متصل است. این برد علاوه بر بلوتوث به ماژول LCD نیز متصل است.

ماژول LCD LM041L :

صفحه ی LCD با اندازه ی ۱۶x۲ می باشد که برای نمایش اطلاعات دریافتی از برد TH توسط برد اصلی استفاده شده است. برای اتصال LCD به برد Arduino پین های مشخصی از این برد به LCD متصل شده است. همچنین رابط کار با این LCD بر روی Arduino ، رابط LiquidCrystal می باشد.

موتور DC:

به نام خدا
تمرین اول cps

کیمیا فخاری کیسمی ۸۱۰۱۹۷۶۵۰

محمدحسین عطائی ۸۱۰۱۹۷۶۳۲

دانیال سعیدی ۸۱۰۱۹۸۵۷۱

محمد قره حسنلو ۸۱۰۱۹۸۴۶۱

پورت VS و VSS از این ماژول به یک منبع تغذیه و GND آن به زمین متصل شده اند. دو ورودی از Arduino یعنی پورت های ۹ و ۱۰ به این موتور وصل شده اند.