

## سیتم‌های نهفته و بی‌درنگ

### گزارش تمرین اول

آشنایی با پروتکل‌های ارتباطی انتقال داده و سنسورها

اعضای گروه:

آریان سلطانی : 810198558

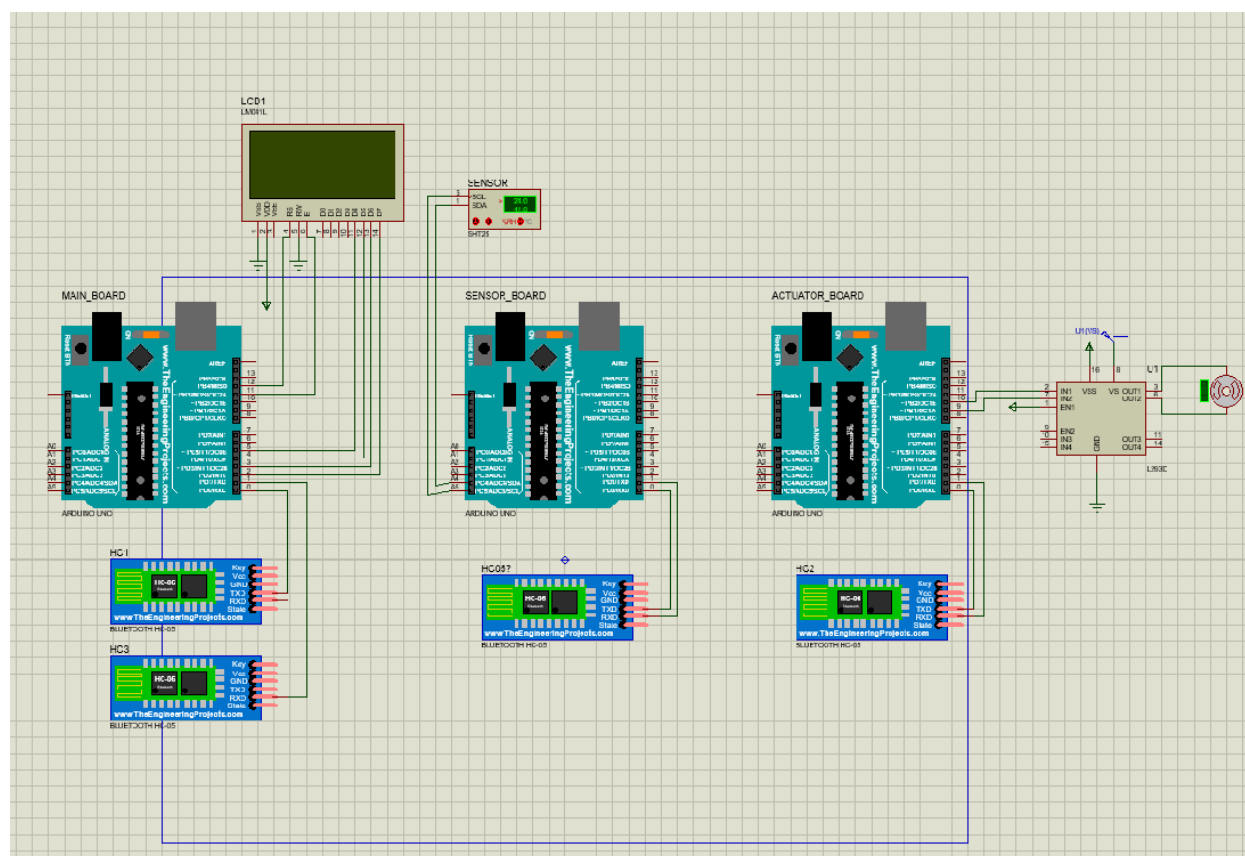
صبا شهبواری: 810198420

معین کرمی: 810198540

محمد هنرجو: 810198487

## شرح پروژه:

این پروژه از سه گره حسگر، عملگر و گره مرکزی تشکیل شده است. گره حسگر به یک سنسور اندازه گیری دما و رطوبت متصل است که داده‌ها را به صورت آنالوگ به برد حسگر می‌دهد. برد حسگر این داده‌ها را با استفاده از بلوتوث به برد مرکزی می‌فرستد. برد مرکزی داده‌های دریافت شده از گره حسگر پردازش می‌کند و با توجه به آن‌ها **duty cycle** موتور را تعیین می‌کند. این برد از طریق بلوتوث سرعت موتور را به گره عملگر ارسال می‌کند. گره مرکزی علاوه بر برد **arduino** یک صفحه نمایشگر نیز دارد که اطلاعاتی اندازه‌گیری شده را نمایش می‌دهد. در گره عملگر یک برد **arduino** و یک مازول بلوتوث وجود دارد که بتواند با گره مرکزی ارتباط برقرار کند و سرعت موتور را دریافت کند. برد عملگر داده‌های دریافتی را به درایور موتور **DC** می‌فرستد.



در ادامه جزئیات هر گره و نحوه پیاده‌سازی آن توضیح داده شده است:

### گره مرکزی:

همان طور که در تصویر زیر مشاهده می‌شود در ابتدا ماژول LCD و اتصالات آن به برد مرکزی را مشخص کردیم. سپس در تابع `setUp` با استفاده از دستور `serial.begin` یک ارتباط سریال با `baud rate` برابر 9600 برای اتصال به ماژول بلوتوث ایجاد می‌کنیم. در ادامه نیز ابعاد صفحه LCD را مشخص کرده‌ایم.

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

void setup() {
  Serial.begin(9600);
  lcd.begin(20,4);
}
```

برد مرکزی به دو ماژول بلوتوث وصل است که از یکی از آن‌ها برای دریافت اطلاعات از گره حسگر استفاده می‌شود و از دیگری برای ارسال اطلاعات به گره عملگر استفاده می‌شود. یکی از این ماژول‌ها روی `com` دو و دیگری روی `com` سه `configure` شده‌است در نتیجه برد می‌تواند فقط روی یکی از آن‌ها بنویسد و فقط از روی دیگری بخواند و تداخلی در ارتباطات به وجود نمی‌آید.

در تابع `loop` بدنه اصلی برنامه را ایجاد می‌کنیم. طبق این کد هر وقت که داده‌ای رو پورت سریال دریافت شود، برد مرکزی ابتدا در تابع `readDataFromSerialPort` داده‌ها را از پورت سریال می‌خواند و با توجه به فرمتی که برای انتقال داده‌ها در نظر گرفتیم مقدار اندازه گیری شده دما و رطوبت را مشخص می‌کند.

سپس در تابع `calculateActuatorSpeed` که شامل چند شرط ساده‌است با توجه به مقدار دما و میزان رطوبت، سرعت مناسب موتور تعیین می‌شود و روی پورت سریال نوشته می‌شود تا به گره عملگر برسد. در نهایت تمام اطلاعات روی `lcd` چاپ می‌شوند.

```
void loop() {
  if(Serial.available() > 4)
  {
    readDataFromSerialPort();
    calculateActuatorSpeed();
    writeToSerialPort(String(motorDutyCycle));
    writeToLCDPort();
  }
}
```

### گره حسگر:

در تابع `setUp` مشابه برد مرکزی یک ارتباط `serial` برای اتصال به ماژول بلوتوث تعریف می‌شود. برای اتصال برد حسگر به سنسور از ارتباط `wire` استفاده می‌کنیم. تابع `readDataFromSensor` به عنوان پارامتر یک `command` می‌گیرد که تعیین می‌کند اندازه‌گیری دما خواسته شده یا رطوبت. سپس یک `transmission` به آدرس I2C آغاز می‌کند و دستور اندازه‌گیری را ارسال می‌کند. بعد از یک `delay` مقادیر نوشته شده روی `wire` را می‌خواند و مقدار رطوبت (یا دما) را محاسبه و ثبت می‌کند.

```
float readDataFromSensor(int addr, float scale, float offset){
    unsigned int data[2];
    Wire.beginTransmission(Addr);
    Wire.write(addr);
    Wire.endTransmission();
    Wire.requestFrom(Addr,2);
    while (Wire.available() < 2)
        delay(10);

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

در تابع `loop` مقدار رطوبت اندازه‌گیری شده با آخرین اندازه ارسال شده مقایسه می‌شود و اگر مقدار جدید بیشتر از ۵ درصد تغییر کرده باشد، اندازه‌های جدید روی پورت سریال نوشته می‌شوند تا از طریق بلوتوث به برد مرکزی برسند.

```
void loop() {
    currTemperatureData.humidity = readDataFromSensor(HUMIDITY_ADDR, 125.0, 6.0);
    currTemperatureData.temp = readDataFromSensor(TEMP_ADDR, 175.72, 46.85);

    if(firstReport || humidityHasMajorChange()){
        sendDataToMainBoard();
        updateLastReport();
        firstReport = false;
    }
}
```

گره عملگر:

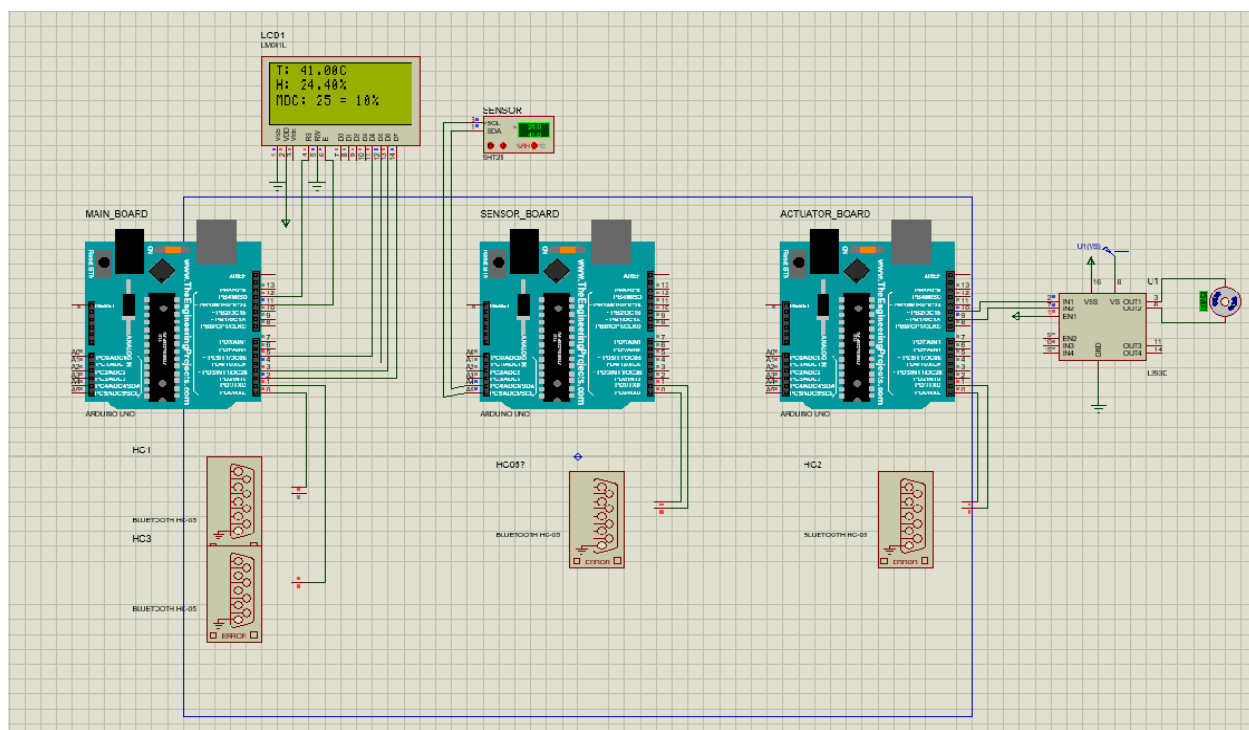
این گره مشابه سایر بورد ها به مازول بلوتوث وصل می‌شود و **duty cycle** موتور را که توسط بورد مرکزی ارسال شده از روی سریال پورت می‌خواند و به صورت **analog** به موتور DC می‌دهد. برای اتصال این بورد به موتور DC از پورت‌های ۹ و ۱۰ استفاده کردیم که برای انتقال سیگنال PWM هستند.

```
void setup() {
  Serial.begin(9600);
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
}

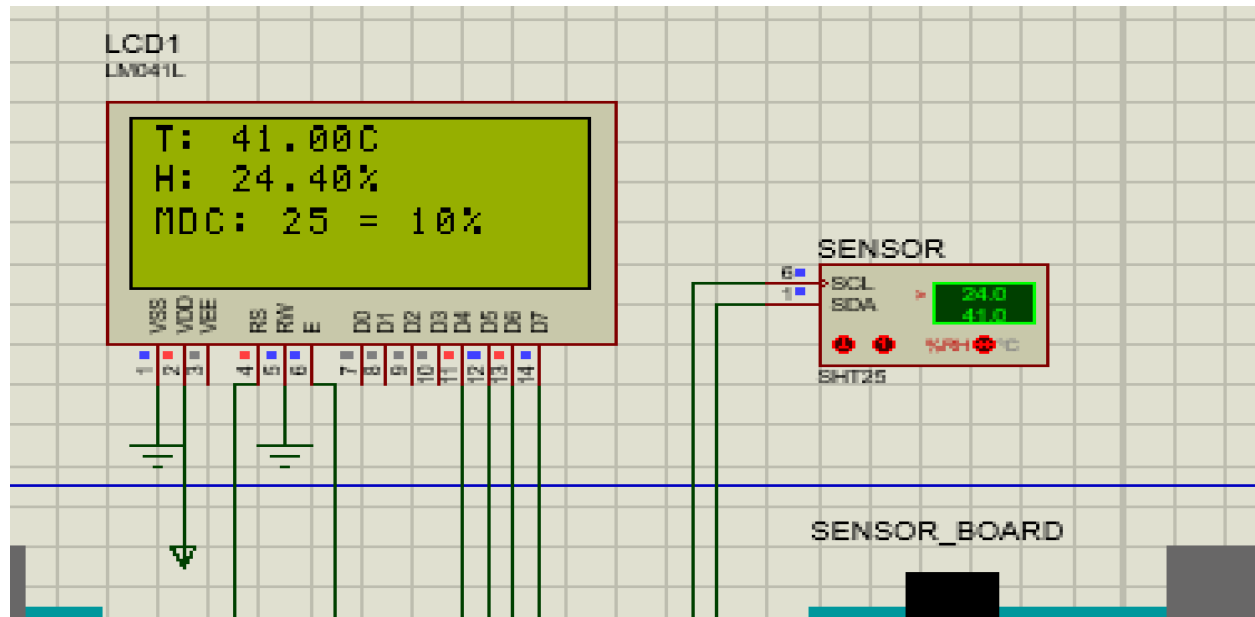
void setMotorSignals()
{
  analogWrite(10, motorSpeed);
  analogWrite(9, 0);
}
```

### نتایج:

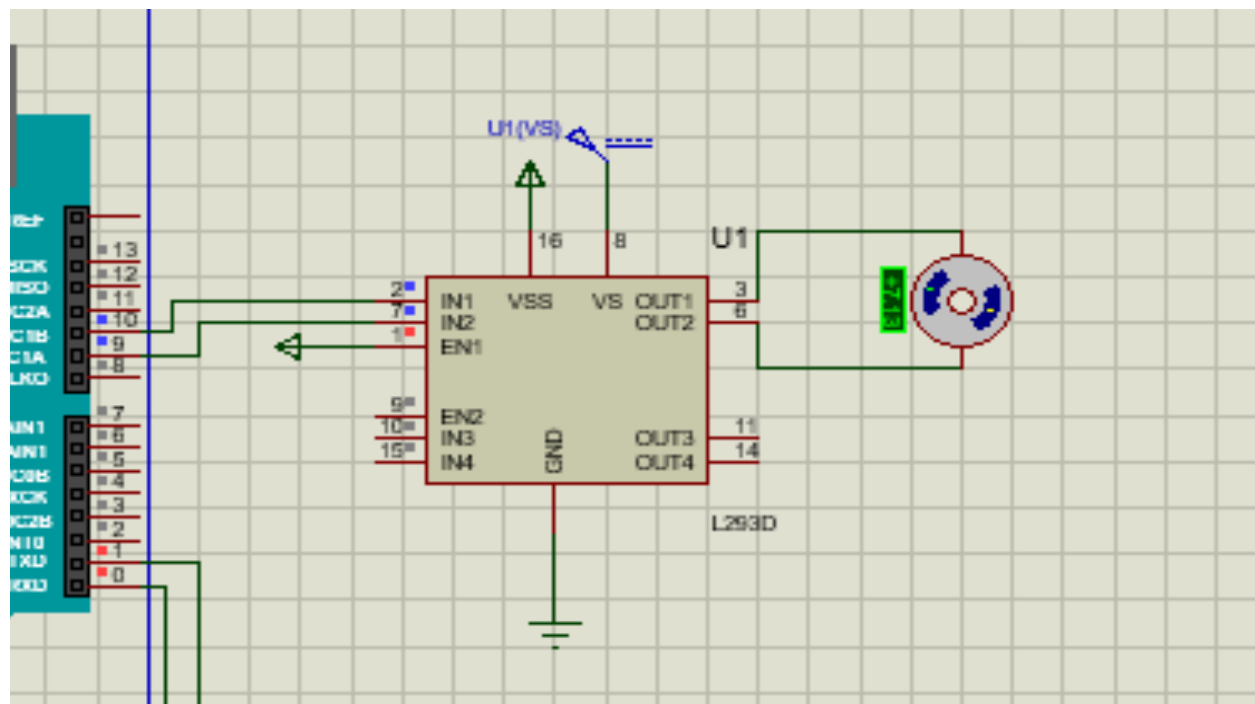
همان طور که در تصویر زیر مشاهده می‌شود اطلاعات اندازه‌گیری شده توسط سنسور به درستی به بورد مرکزی انتقال داده شده‌اند و روی نمایشگر بورد مرکزی قابل مشاهده هستند. سرعت موتور نیز متناسب با این اندازه‌ها تعیین شده و موتور با دریافت سیگنال از بورد عملگر شروع به کار می‌کند:



تصویر نمایشگرها:



تصویر موتور:



## پرسش‌ها:

(1)

پروتکل بلوتوث در محدوده فرکانسی 2400 تا 2483.5 مگاهرتز کار می‌کند. برای آن که سیگنال‌های بلوتوث دستگاه‌هایی که در یک منطقه هستند با هم تداخل نکنند روش‌های مختلفی وجود دارد. یکی از این روش‌ها frequency hopping است. در این روش فرستنده و گیرنده اطلاعات به طور همزمان بر روی یک مجموعه خاص از فرکانس‌ها ارتباط برقرار می‌کنند و بین این فرکانس‌ها به طور مداوم طبق یک الگوریتم شبه تصادفی پرش می‌کنند. برای مثال محدوده بلوتوث به ۷۹ فرکانس مختلف در باند فرکانسی 2.4 گیگاهرتز تقسیم می‌شود و سیگنال‌ها با نرخ 1600 بار در ثانیه بین این فرکانس‌ها پرش می‌کنند. استفاده از این روش باعث می‌شود احتمال تداخل با سایر فرکانس‌ها بسیار کم شود.

(2)

پروتکل I2C برای ارسال و دریافت داده بین دستگاه‌های مختلف از ۲ پایه استفاده می‌کند. این پایه‌ها عبارتند از SDA (Serial Data Line) که داده‌ها بر روی آن جابه‌جا می‌شوند و دیگری SCL (Serial Clock Line) است که برای انتقال سیگنال کلاک استفاده می‌شود. کلاک I2C توسط Master تولید می‌شود و بین تمام slave ها پخش می‌شود. دستگاه‌های slave هر کدام یک آدرس مشخص دارند و فقط در صورت درخواست دستگاه مستر پاسخ می‌دهند و خودشان هیچ وقت نمی‌توانند ارتباط را شروع کنند. در header هر فریم داده که توسط master ارسال می‌شود ۷ بایت که آدرس slave مورد نظر را تعیین می‌کند قرار می‌گیرد. هر دستگاهی که پیامی را دریافت می‌کند آدرس موجود در هدر آن را با آدرس خودش مقایسه می‌کند تا بفهمد باید پاسخ دهد یا نه. همچنین در هدر هر فریم تعدادی بیت به عنوان ack, nack و برای مشخص کردن آغاز و اتمام ارتباط وجود دارند که باعث می‌شوند از انتقال درست داده‌ها بتوانیم اطمینان حاصل کنیم.

(3)

موتور stepper:

موتور stepper نوعی موتور DC است که به صورت گسسته حرکت می‌کند. درون این موتور چندین سیم پیچ وجود دارد که در گوه‌هایی به نام فاز قرار دارند و با عبور الکتریسیته از هر یک از این فازها موتور با سرعت یک گام به چرخش در می‌آید. به همین دلیل سرعت چرخش این نوع موتور ثابت است. با کنترل کردن گام‌های یک stepper می‌توان سرعت آن را با دقت بالایی تعیین کرد. به طور کلی استپ موتور ها برای مواردی که نیاز به سرعت کم در مسافت کوتاه باشد، مناسب هستند. این موتور برخلاف موتور servo قابلیت تنظیم گشتاور ندارد.

موتور servo :

ساختار این موتور مشابه موتور stepper می‌باشد اما تعداد قطب‌های مغناطیسی بیشتری دارد که باعث می‌شود حداکثر سرعت و دقت بیشتری داشته باشند. برای همین این موتور در مواردی که نیاز به دقت بالا در کنترل موقعیت، حرکت در مسافت طولانی و در مواردی که نیاز به تولید گشتاور زیاد با سرعت بالا است (مانند شناورهای دریایی)، مناسب است. اما از طرفی قیمت و ابعاد این موتور به دلیل پیچیدگی ساختار آن بیشتر از موتور stepper است

موتور DC:



موتور DC نوعی موتور القایی است که با قرار دادن سیم های رسانا در یک میدان مغناطیسی جریان الکتریکی تولید می کند و با تاثیر این جریان روی میدان مغناطیسی، یک نیروی چرخشی ایجاد می شود. از جمله مزایای موتور DC می توان به این اشاره کرد که برای شروع چرخش نیاز به توان بالایی ندارد و اندازه کوچک و قیمت مناسبی دارد. اما این موتور در مقایسه با موتورهای stepper و servo دقت پایینی دارد. همچنین در زمان بارگذاری بالا سرعت چرخش و نیروی ایجاد شده توسط آن کاهش میابد.