به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

سیستم های نهفته‌ی بی‌درنگ

گزارش طراحی و پیاده‌سازی گلدان هوشمند

سجاد گندم مالمیری

علیرضا توکلی

عرفان رزاقی

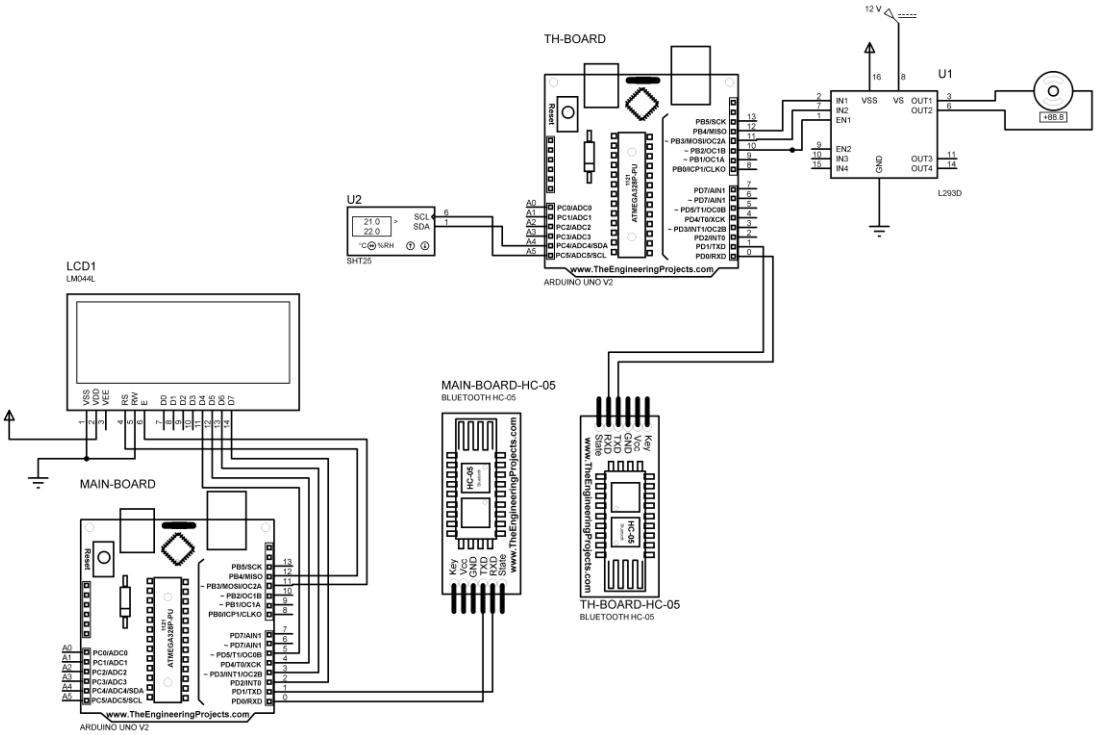
کاوه معصومی

1401/02/10

فهرست

بخش 1 – توضیح اجزای سیستم

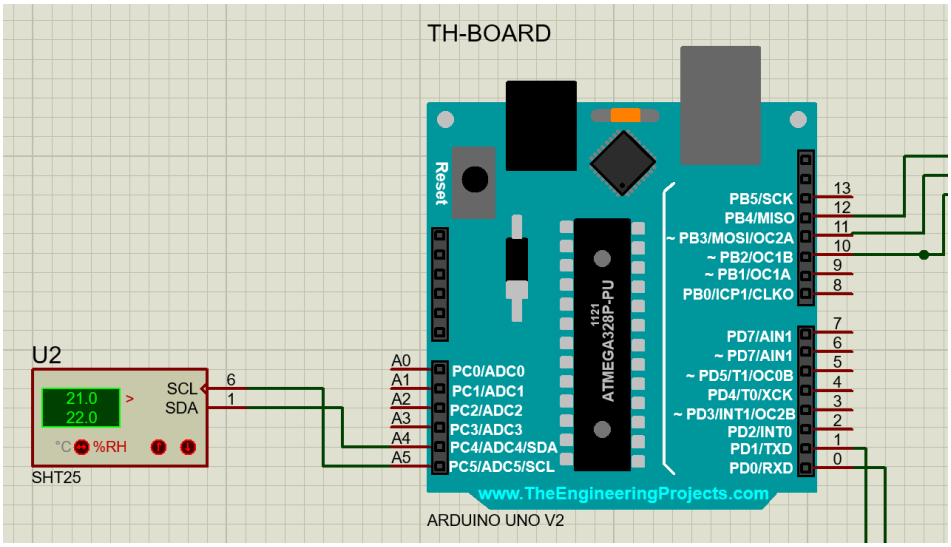
سیستم در یک نمای کلی به شکل زیر است:



این سیستم از دو بخش Main Board (پایین سمت چپ) و TH Board (بالا سمت راست) تشکیل شده است. بخش TH Board دما و رطوبت را هر ۵ ثانیه از سنسور SHT۲۵ (U۲ در شکل) میخواند و آن را با کمک ماژول بلوتوث HC-۰۵ به Main Board ارسال میکند. سپس Main Board تصمیم میگیرد که موتور متصل به TH Board با چه سرعتی آبیاری به گلدان را انجام دهد و نتیجه ی تصمیم را به TH Board برمیگرداند و خودش نیز وضعیتی از سیستم شامل دما و رطوبت و تصمیم گرفته شده را در LCD متصل به خود نمایش میدهد. در نهایت نیز موتور متصل به TH Board با سرعت مناسب چرخش را انجام میدهد.

حال به بررسی سورس کد هر کدام از بخش های سیستم بپردازیم:

- سنسور SHT۲۵: این سنسور با پروتکل I۲C و با کمک کتابخانه ی Wire.h با آردوییونوی TH Board صحبت میکند. همانطور که در سورس کد زیر میبینیم، این سنسور یک آدرس I۲C مخصوص دارد و برای دریافت هر کدام از دما و رطوبت لازم است یک درخواست با کد خاص به آن ارسال کنیم و منتظر نتیجه ی آن بمانیم:

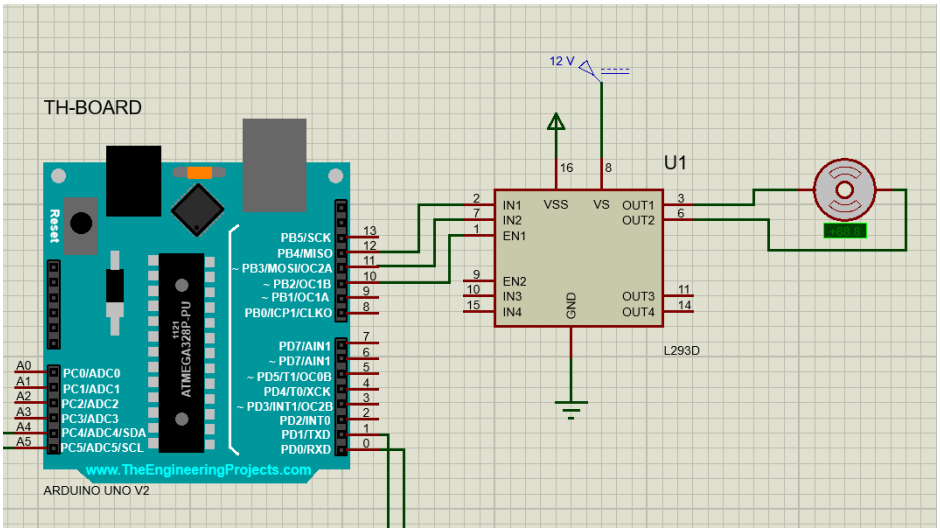



```
float getHumidity() {
  unsigned int data[2];
  Wire.beginTransmission(TH_SENSOR_ADDRESS);
  Wire.write(0xF5); // Send humidity measurement command
  Wire.endTransmission();
  delay(200);

  Wire.requestFrom(TH_SENSOR_ADDRESS, 2); // Read 2 bytes of data

  while (Wire.available() < 2)
    delay(200);

  data[0] = Wire.read();
  data[1] = Wire.read();

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

  return humidity;
}
```

```
float getTemperature() {
  unsigned int data[2];
  Wire.beginTransmission(TH_SENSOR_ADDRESS);
  Wire.write(0xF3); // Send temperature measurement command
  Wire.endTransmission();
  delay(200);

  Wire.requestFrom(TH_SENSOR_ADDRESS, 2); // Read 2 bytes of data

  while (Wire.available() < 2)
    delay(200);

  data[0] = Wire.read();
  data[1] = Wire.read();

  float tempCentigrade = (((data[0] * 256.0 + data[1]) * 175.72) / 65536.0) - 46.85;

  return tempCentigrade;
}
```

- موتور DC و درایور L۲۹۳D: چون جریانی که موتور DC میکشد بیشتر از جریانی است که آردوینو میتواند تامین کند، نیاز داریم که برای راه اندازی موتور DC از یک ماژول درایور مثل L۲۹۳D استفاده کنیم. طرز استفاده از آن نیز همانطور که در شکل معلوم است خیلی ساده است، کافیست

پین های IN۱ و IN۲ را به ترتیب به HIGH و LOW وصل کرده و سرعت چرخش موتور را با استفاده از PWM روی پین EN۱ کنترل کنیم. واضح است که باید منبع تغذیه ی قدرتمندتر از آردوینو را به پین VS درایور وصل نمود:

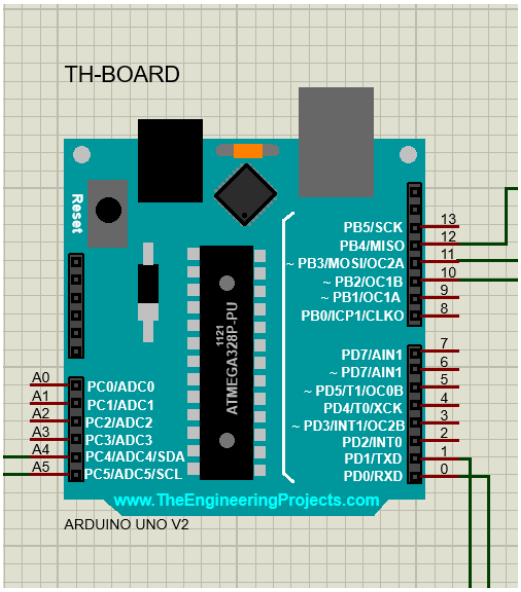



```
const int motorDriverPinInput1  = 12;
const int motorDriverPinInput2  = 11;
const int motorDriverPinEn1     = 10;

unsigned long previousMillis = 0;
const long samplingIntervalMillis = 5000;

byte motorSpeed;

void setup() {
  // Initialize DC Motor Driver
  pinMode(motorDriverPinInput1, OUTPUT);
  pinMode(motorDriverPinInput2, OUTPUT);
  pinMode(motorDriverPinEn1, OUTPUT);
  digitalWrite(motorDriverPinInput1, HIGH);
  digitalWrite(motorDriverPinInput2, LOW);
```

```
if (Serial.available() > 0) {
  char startComm = Serial.read();
  if (startComm == START_COMM_DELIM) {
    motorSpeed = (byte) Serial.read();
    analogWrite(motorDriverPinEn1, motorSpeed);
  }
}
```

PWM

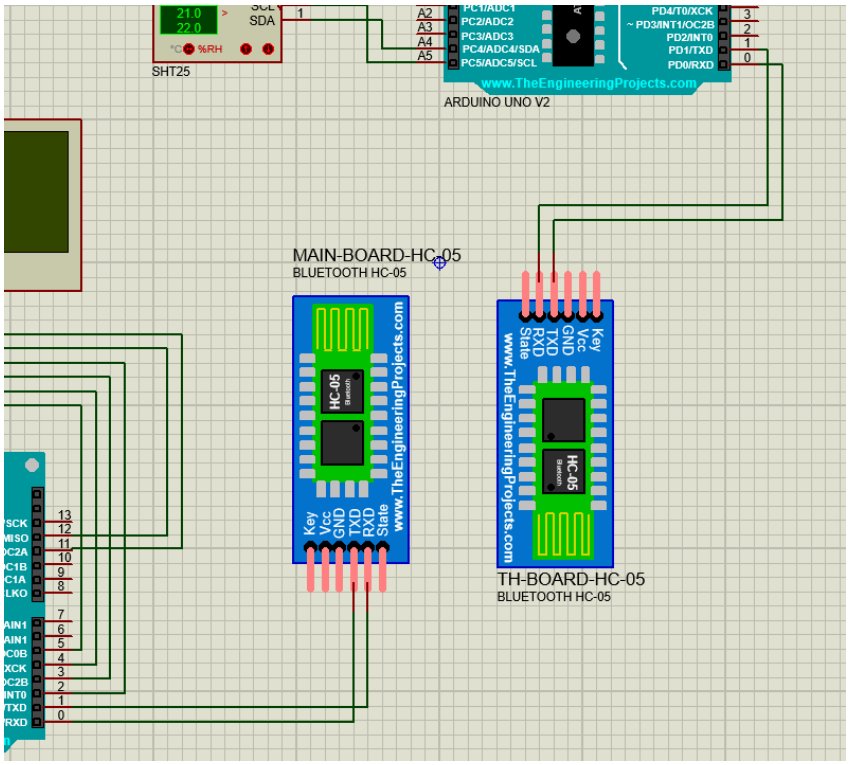
- برد TH-BOARD : این برد مهم، هر پنج ثانیه داده ها را از سنسور خوانده و آن را به برد اصلی ارسال میکند و منتظر رسیدن دستور از طرف آن (از طریق بلوتوث سریال) برای کنترل موتور میماند. دقت کنیم که این صبر پنج ثانیه ای به صورت Non Blocking است و از یک تایمر برای این کار استفاده میشود. برای ارسال داده ها به برد اصلی، ابتدا یک کاراکتر نشان دهنده ی شروع ارسال دیتا مانند ! ارسال شده و سپس به صورت متوالی دو float حاوی مقادیر سنسور ها را ارسال میکند.




```
39 void loop() {
40   if (Serial.available() > 0) {
41     char startComm = Serial.read();
42     if (startComm == START_COMM_DELIM) {
43       motorSpeed = (byte) Serial.read();
44       analogWrite(motorDriverPinEn1, motorSpeed);
45     }
46   }
47
48   unsigned long currentMillis = millis();
49   if (currentMillis - previousMillis >= samplingIntervalMillis) {
50     previousMillis = currentMillis;
51
52     float humidity = getHumidity();
53     float temperature = getTemperature();
54
55     Serial.print(START_COMM_DELIM);
56     sendFloat(humidity);
57     sendFloat(temperature);
58   }
59
60   delay(100);
61 }
```

Non Blocking

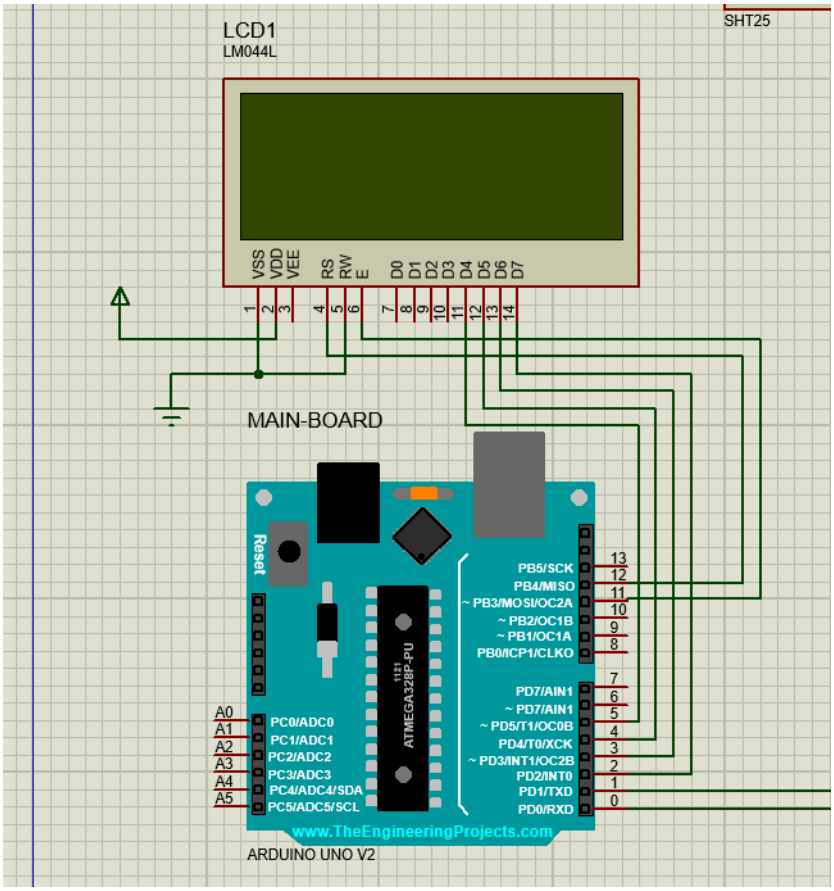
- ماژول بلوتوث HC-۰۵ : این ماژول از پروتکل UART و پین های Tx Rx برای ارتباط با آردوینو استفاده میکند. برای اتصال دو نسخه ماژول بلوتوث در این پروژه در شبیه سازی، کافیست یکی را به COM ۴ و دیگری را به COM ۳ وصل کرده و با نرم افزار Virtual Serial Port Driver این دو را به صورت مجازی به همدیگر وصل کنیم. در نتیجه استفاده از آن ها و انتقال داده ها به سادگی استفاده از رابط سریال آردوینو خواهد بود:




```
32
33    // Initialize Bluetooth Serial Communication
34    Serial.begin(9600);
35    while (!Serial);
36
```

```
54
55      Serial.print(START_COMM_DELIM);
56      sendFloat(humidity);
57      sendFloat(temperature);
58    }
59
60    delay(100);
61  }
62
63  void sendFloat(float num) {
64    byte* f_bytes = (byte*) &num;
65    for (int idx = 0; idx < sizeof(float); Serial.write(f_bytes[idx++]) );
66  }
67
```

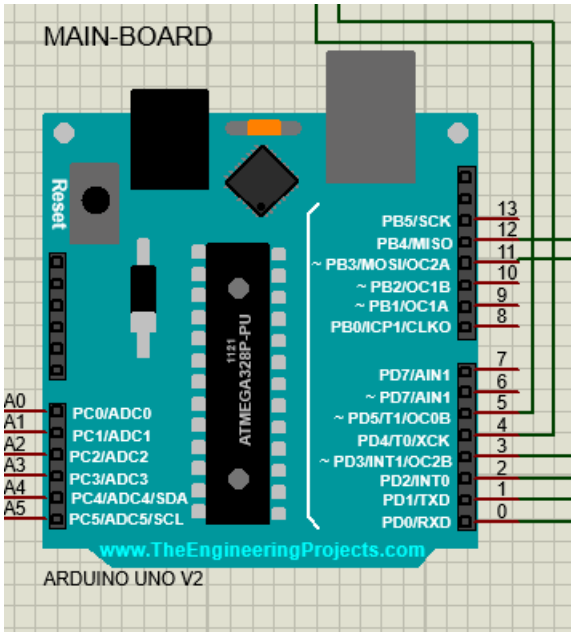
- ماژول LCD LM۰۴۴L : برای استفاده از این ماژول کافیست پین های لازم را به آردوینو وصل کرده و سپس به راحتی با کتابخانه ی LiquidCrystal.h از آن برای نشان دادن دما و تصمیمات استفاده کنیم:




```
10
11 const int lcdRs = 12, lcdEn = 11, lcdD4 = 5, lcdD5 = 4, lcdD6 = 3, lcdD7 = 2;
12 LiquidCrystal lcd(lcdRs, lcdEn, lcdD4, lcdD5, lcdD6, lcdD7);
13
14 void setup() {
15   // Initialze LCD With 20x4 Size
16   lcd.begin(20, 4);
```

```
93 void displayStatus(float humidity, float temperature, String action) {
94   lcd.clear();
95
96   lcd.setCursor(0, 0);
97   lcd.print("Humidity:  ");
98   lcd.print(humidity);
99   lcd.print("  %RH");
100
101  lcd.setCursor(0, 1);
102  lcd.print("Temp    :  ");
103  lcd.print(temperature);
104  lcd.print("  C");
105
106  lcd.setCursor(0, 2);
107  lcd.print("======|Action|======");
108  lcd.setCursor((20 - action.length())/2, 3);
109  lcd.print(action);
110 }
```

- Main Board: این برد اصلی ترین ماژول و تصمیم گیرنده ی نهایی است اما با این حال کار ساده ای به عهده دارد. او همواره منتظر پیغامی از برد TH مانده و در صورت دریافت پیغام و صحیح بودن مقادیر سنسور ها، آن ها را بررسی کرده و نتیجه ی تصمیم را به برد TH ارسال میکند:

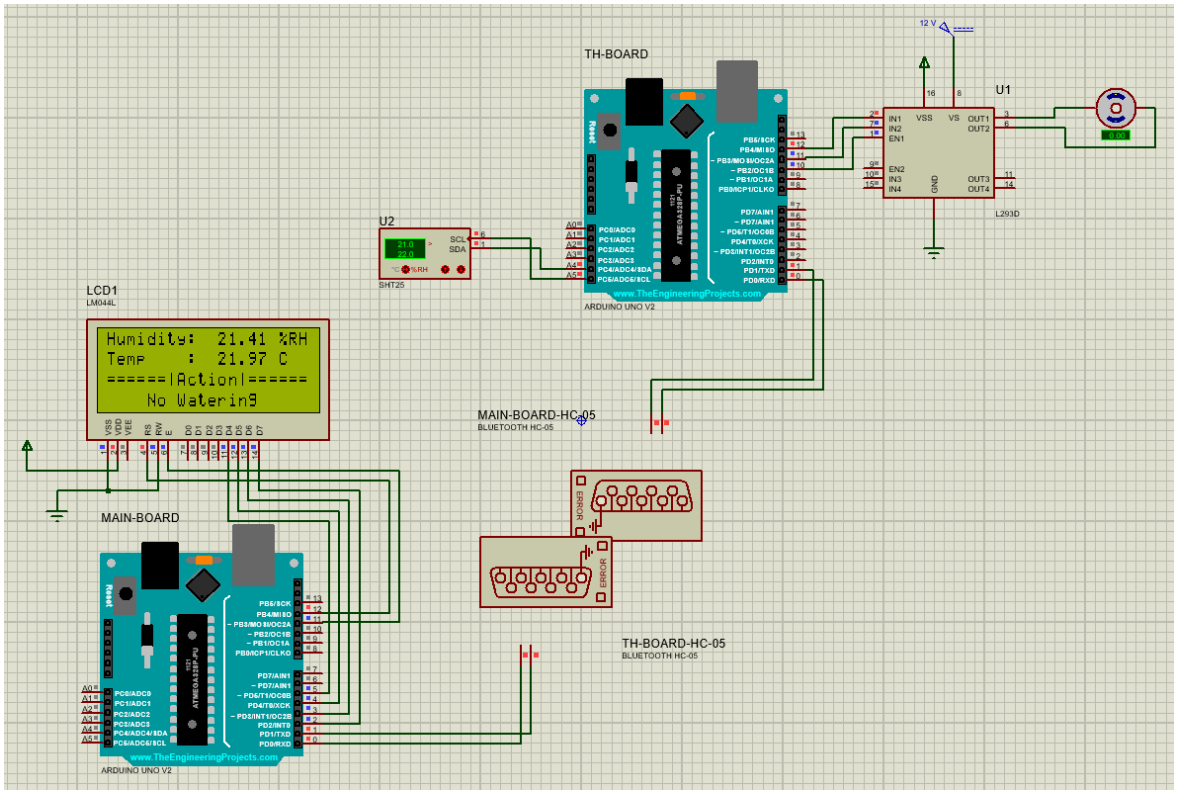



```
24
25  void loop() {
26    if (Serial.available() > 0) {
27      char startComm = (char) Serial.read();
28      if (startComm == START_COMM_DELIM) {
29        float humidity = readFloat();
30        float temperature = readFloat();
31        updateDisplayTH(humidity, temperature);
32        updateDisplayAction("Deciding...");
33
34        byte motorSpeed = decideMotorSpeed(humidity, temperature);
35        Serial.print(START_COMM_DELIM);
36        Serial.write(motorSpeed);
37
38      }
39
40    }
41    delay(100);
42  }
43
```

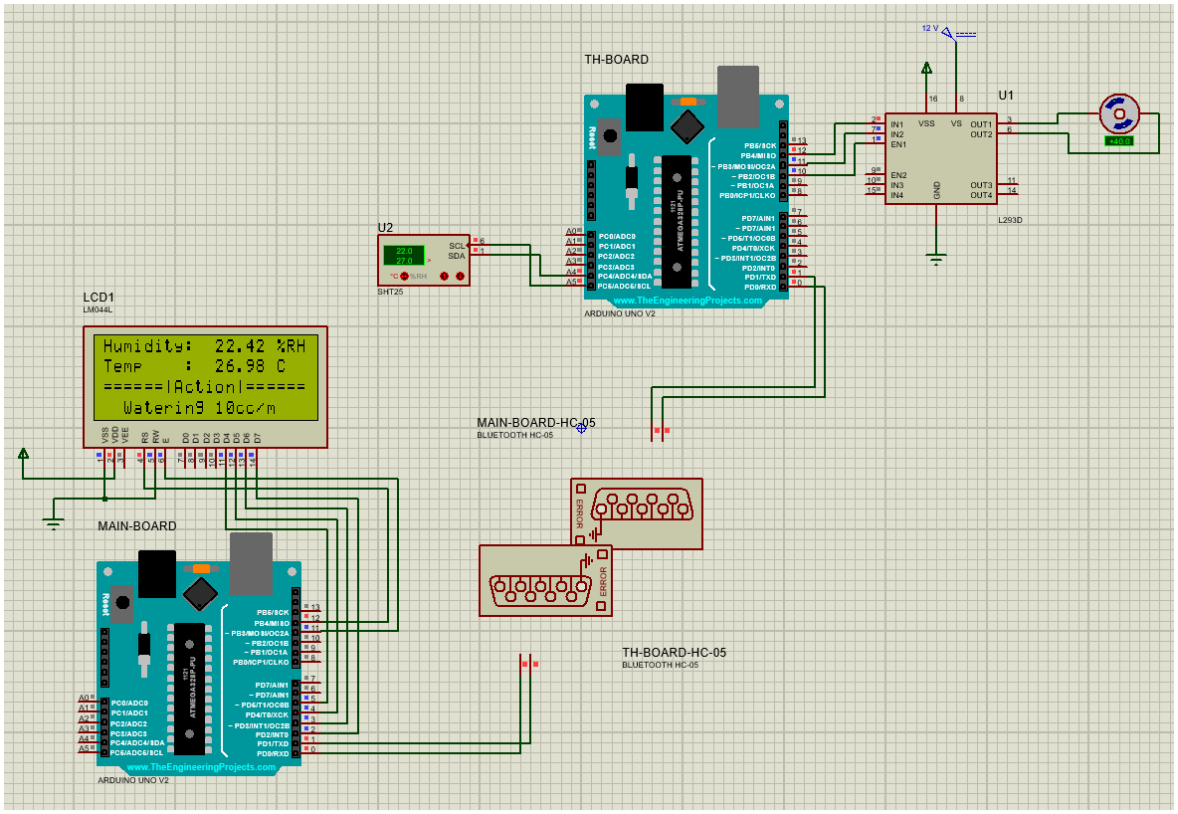
```
byte decideMotorSpeed(float humidity, float temperature) {
  byte motorSpeed;
  if (humidity > 50.0f) {
    updateDisplayAction("No Watering");
    motorSpeed = 0x00;
  }
  else if (humidity > 20.0f) {
    if (temperature < 25.0f) {
      updateDisplayAction("No Watering");
      motorSpeed = 0x00;
    }
    else {
      updateDisplayAction("Watering 10cc/m");
      motorSpeed = 0x19;
    }
  }
  else {
      updateDisplayAction("Watering 20cc/m");
      motorSpeed = 0x40;
  }
  return motorSpeed;
}

```

بخش 2 – نتایج سیمولیشن

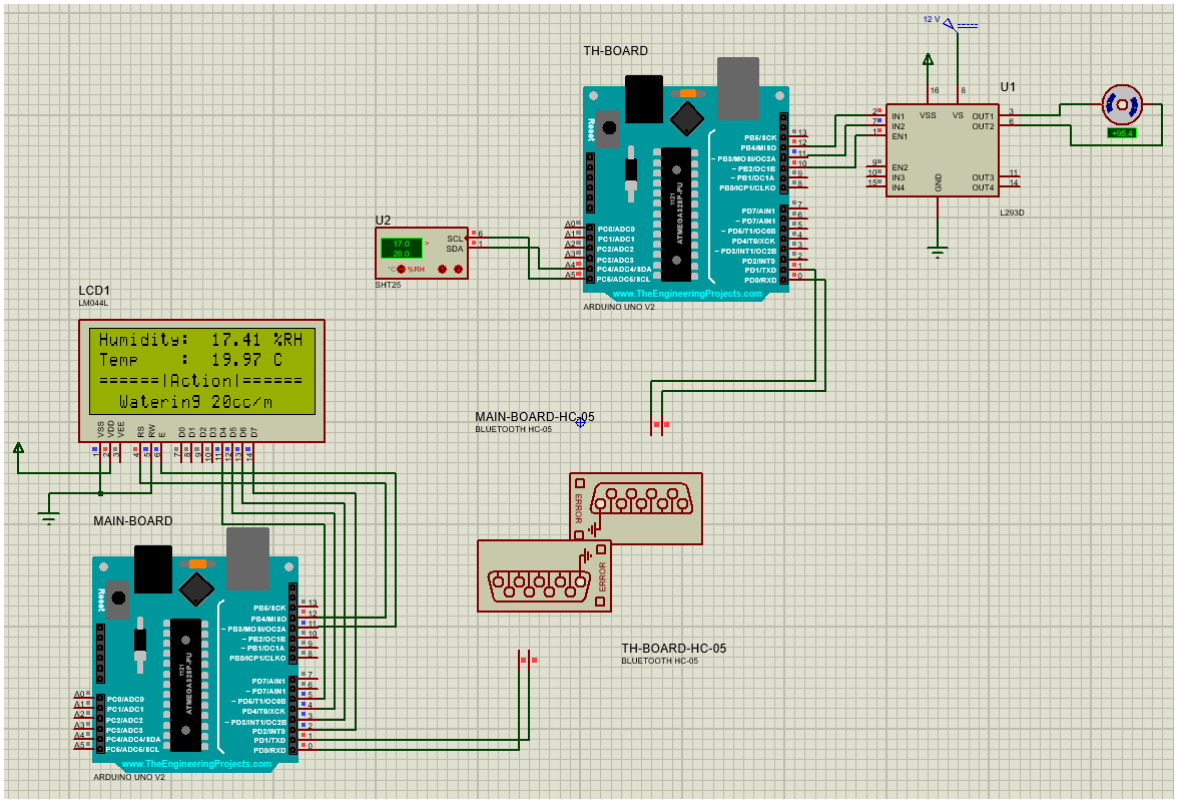
حالت اول: بدون آبیاری



حالت دوم: آبیاری با سرعت ۱۰ سی سی در دقیقه (duty cycle ۱۰%)



حالت سوم: آبیاری با سرعت ۲۰ سی سی در دقیقه (duty cycle ۲۵%)



بخش 3 – پاسخ سوالات

- در مورد بلوتوث، از چه فرکانسی برای ارتباط بی سیم استفاده میشود؟ در صورت وجود چند دستگاه بلوتوث در اطراف هم، چگونه از تداخل داده های ارسالی دستگاه ها جلوگیری میشود؟ نیازی به ارایه جزییات پروتوکل ارتباطی بلوتوث نیست. بیان مفاهیم کلی کافی است.

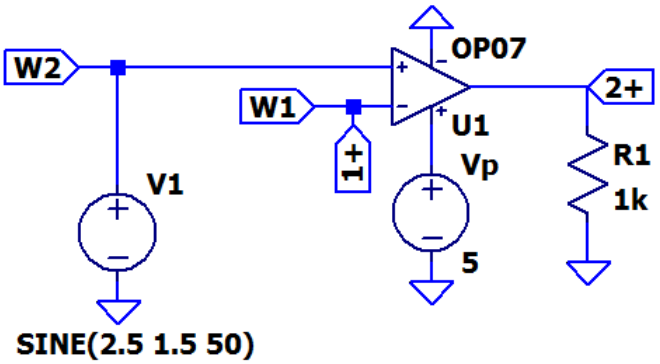
در بلوتوث از فرکانس ۲.۴۵ GHz برای ارتباط بی سیم استفاده میشود. برای جلوگیری از تداخل امواج نیز از Frequency-Hopping Spread Spectrum استفاده میکند به این صورت که با hop rate ۱۶۰۰ بار در ثانیه فرکانسی که روی آن در حال ارسال داده است را تغییر میدهد. باند فرکانسی بلوتوث به تعدادی کانال که هر کدام بخشی از باند کلی هستند تقسیم شده است و این پرش متمادی روی هر باند باعث جلوگیری از تداخل خواهد شد.

- اگر نیاز به اتصال چند سنسور مختلف که از پروتکل I۲C استفاده میکنند باشد آیا می توان همه را به تنها پورت موجود I۲C در AVR متصل کرد؟ در این صورت، چگونه تضمین میشود که داده های ارسال آنها با هم تداخل نمیکند؟

بله مشکلی ایجاد نمیکند، چرا که هر کدام از سنسورها در I۲C از یک آدرس خاص ۷ یا ۸ بیتی استفاده میکنند و در مسیج هایی که controller میفرستد، آدرس مقصد در ابتدای پیام ارسال میشود. بنابراین هر کسی که پیام را میبیند میتواند آدرس مقصد را با آدرس خود مقایسه کند و در صورتی که با آن ها فرق داشت دیگر به محتوای پیام گوش ندهد.

- نحوه ی ساخت PWM را شرح دهید.

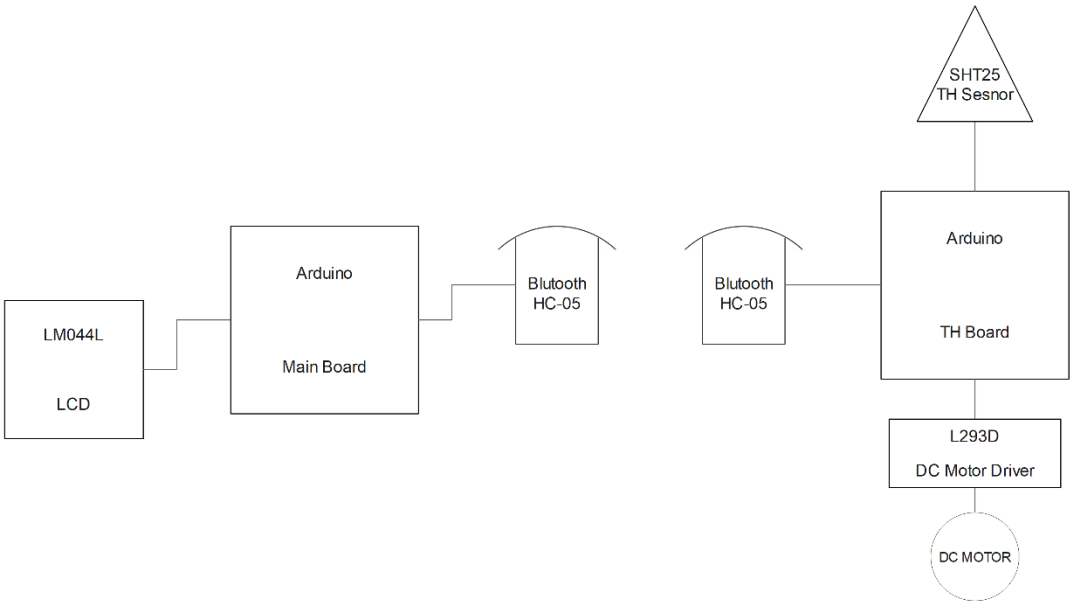
برای ساخت PWM چندین روش وجود دارد از جمله تکنیک های آنالوگی، sigma-delta modulation و direct digital synthesis.
یکی از ساده ترین روش های ساخت PWM این است که دو سیگنال کنترلی modulation و carrier را با یکدیگر مقایسه کنیم که به روش carrier-based PWM معروف است. سیگنال carrier یک موج مثلثی با فرکانس بالاست و سیگنال modulation میتواند به هر صورتی باشد. به مدار آن دقت کنیم:



برای ساخت PWM کافیست از ورودی منفی آمپلی فایر برای سیگنال carrier و از ورودی مثبت آن برای سیگنال modulation استفاده کنیم.

به این صورت، یک سیگنال modulation با ولتاژ بالاتر باعث یک خروجی PWM با duty cycle بزرگتر خواهد شد.

- طراحی مفهومی این تمرین را رسم کنید و تنها اتصالات و اجزای اصلی را نمایش دهید.



نسخه ی کمی دقیق تر این طراحی در این بخش آورده شده بود.