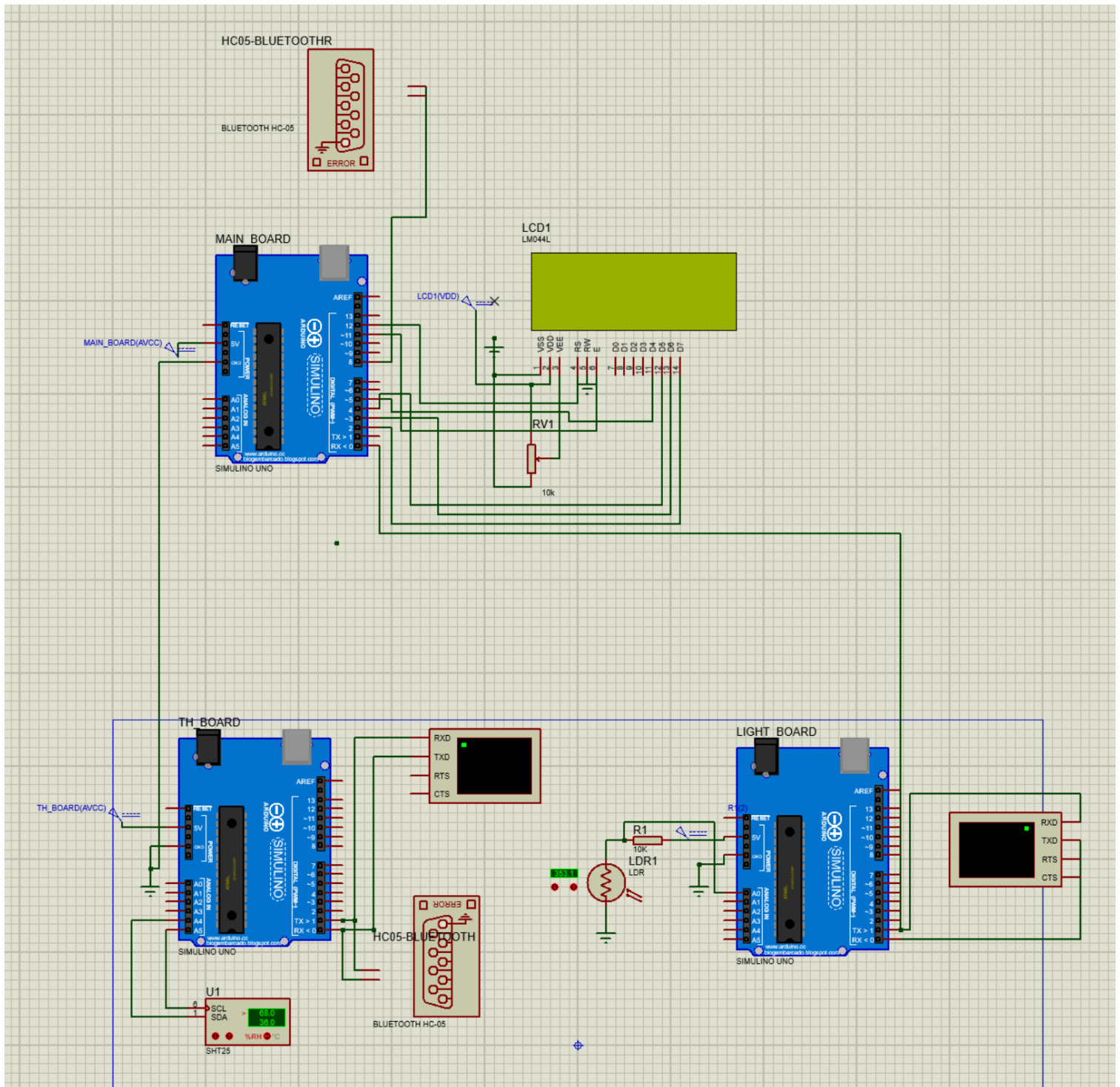
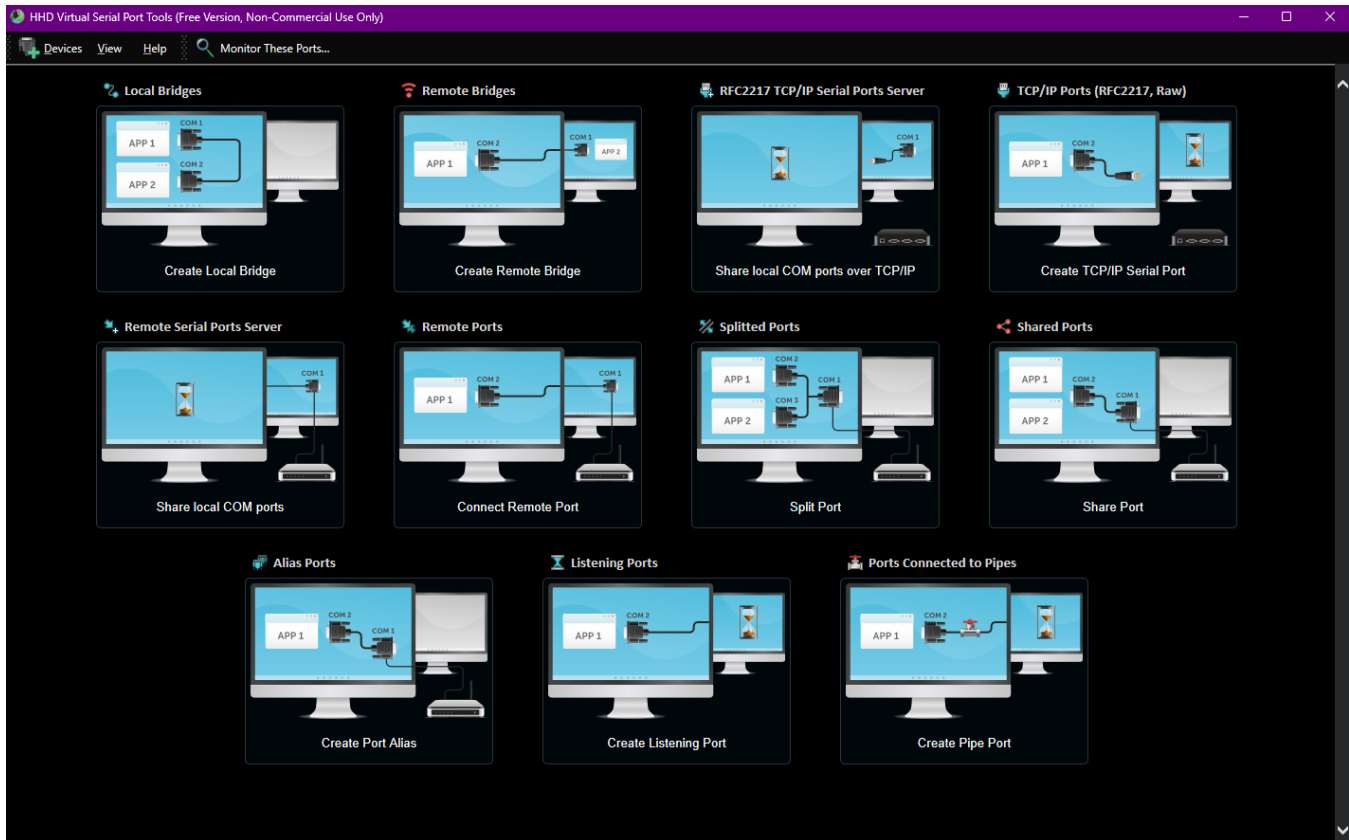


در این بخش ابتدا به بررسی مدار رسم شده در خود پروتئوس میپردازیم:



در این شماتیک از سه اردیو نو UNO استفاده کردم که به دو سنسور رطوبت و ودما متصل است، همچنین با استفاده از یک ماژول بلوتوث ارتباط بین ماژول رطوبت و دما (که خود متصل به اردوینو است) را برقرار کرده ام. در طی این روند دو نکته حائز اهمیت وجود دارد:

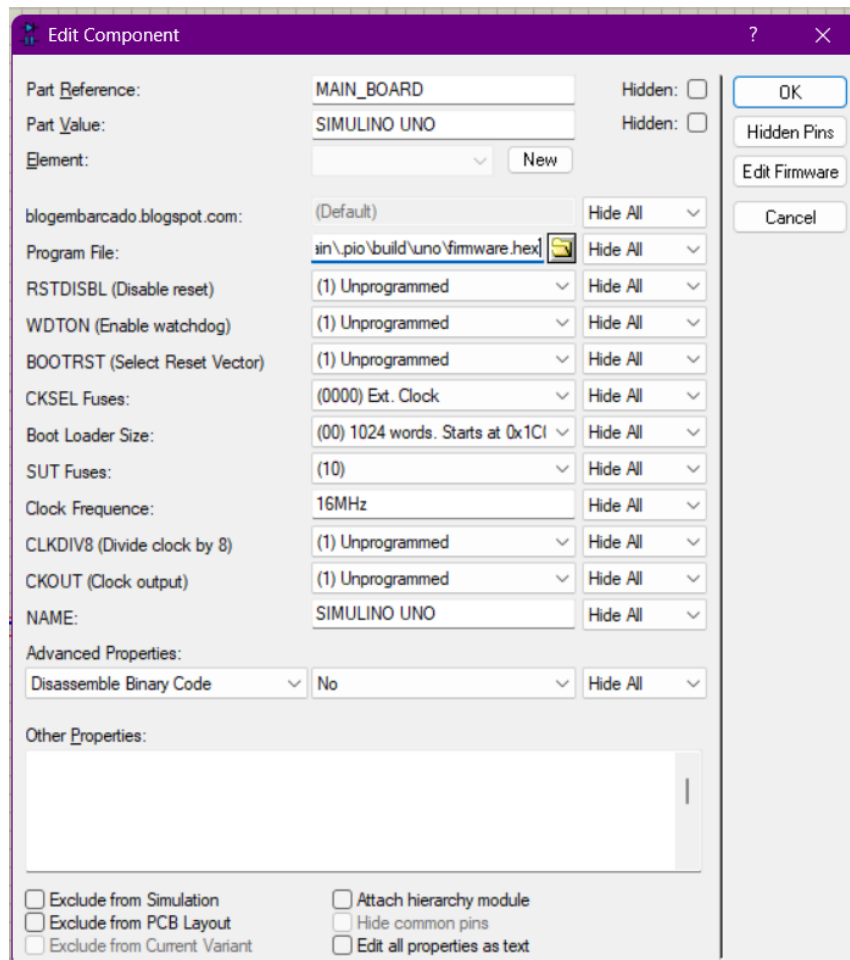
1. از آنجا که در Main-Board ارتباط توسط پورت سریال دیگری نمیتوانیم داشته باشیم (چرا که این مدل آردوینو یک پورت سریال بیشتر ندارد). پس باید راهی جایگزین برای آن پیدا کنیم که طبق پیشنهاد سوال Software Serial میتواند گزینه مناسبی باشد.
2. با توجه به اینکه انتقال بلوتوث در شبیه سازی انجام نمیشود، باید با استفاده از یک نرم افزار پورت کام مجازی بسازیم و به شبیه ساز پروتوس معرفی کنیم سپس اطلاعات دو مازول بلوتوث از طریق این پورت ها به صورت کامل نرم افزاری بدون نیاز به سخت افزار فیزیکی منتقل میشود. در این تمرین بنده از نرم افزار HHD Virtual Serial Port Tools استفاده کرده ام که محیطی مانند شکل زیر دارد:



و نهایتا آخرین قطعه ای که در بخش طراحی نیاز است به آن اشاره کنیم:

LCD است که با اتصال به main-board خروجی و وضعیت را به ما نشان میدهد. صفحه نمایش LCD با مدل LM044L یک نمایشگر ماتریسی مایع است که به طور خاص برای استفاده در دستگاه های الکترونیکی و سیستم های کنترلی طراحی شده است که رابط ارتباطی آن UART بوده که در Arduino UNO پشتیبانی میشود. از نقاط مثبت آن میتوان به کتاب خانه های آماده برای چاپ های کاراکتری روی آن اشاره کرد.

به طور خاص لازم به ذکر است که در محیط های شبیه سازی، برای پروگرام کردن میکرو ها امکان ارتباط مستقیم آن با IDE هایی مانند platformio یا Arduino Studio نمیشود از این رو در بین فایل های تولیدی توسط این نرم افزار ها، کافی است فایل firmware.hex را به برد موجود در محیط پروتوس بدهید که با دوبار کلیک بر روی برد پنجره زیر باز میگردد:



این کار باید برای دو برد دیگر نیز انجام شود.

اکنون به سراغ بررسی کدهای استفاده شده در سه برد میرویم هر قسمت را به طور جداگانه بررسی میکنیم:

:MainBoard

```
#include <LiquidCrystal.h>
#include <Arduino.h>
#include <AltSoftSerial.h>

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

float humidity = 0;
float temprcher = 0;
float lux = 0;
bool C = 0;
float value = 0;

AltSoftSerial altSerial1;
```

در این بخش از فایل صرفاً یک سری از متغیرهایی که در ادامه به آن‌ها نیاز داریم را معرفی کردم.

متغیرهایی که با d شروع میشوند مربوط به LCD هستند.

کتابخانه LiquidCrystal برای کار با LCD و کتابخانه AltSoftSerial برای ایجاد یک سریال مجازی است.

سایر متغیرها از اسمشون فعالیتشون مشخص است. (متغیر شرطی C یک فلگ برای چک کردن شرایط است.)

```
void setup() {
  altSerial1.begin(9600); // Initialize AltSoftSerial for communication
  Serial.begin(9600);      // Initialize Serial for communication

  lcd.begin(20, 4);        // Initialize LCD: 20 columns, 4 rows
}
```

در این بخش ابتدا یک پورت سریال مجازی با Baudrate برابر 9600 ایجاد میکنیم و سپس سریال موجود در داخل آردوینو را نیز با Baudrate برابر 9600 باز میکنیم. سپس تعداد خانه های LCD را مشخص کردیم.

```
void loop() {
  // Read data from Serial (from PC)
  if (Serial.available() > 0) {
    String receivedValue = Serial.readStringUntil('\n');
    Serial.println(receivedValue);

    // Check if received value starts with 'n' and ends with 'n'
    if (receivedValue.substring(0, 1) == "n" && receivedValue.substring(receivedValue.length()-1) == "n") {
      String v = receivedValue.substring(1, receivedValue.length()-1);
      value = v.toFloat();
      lux = value; // Store lux value
    }
  }

  // Read data from AltSoftSerial (another device)
  if (altSerial1.available() > 0) {
    String receivedValue = altSerial1.readStringUntil('\n');

    // Extract temperature and humidity values separated by ','
    int commaIndex = receivedValue.indexOf(',');
    String number1 = receivedValue.substring(0, commaIndex);
    String number2 = receivedValue.substring(commaIndex + 1);

    temprcher = number1.toFloat(); // Store temperature value
    humidity = number2.toFloat(); // Store humidity value
  }
}
```

در این بخش برای هر دو پورت سریال سخت افزاری و مجازی بررسی میکنیم که آیا داده جدیدی ثبت شده است یا خیر. در صورت آنکه ثبت شده بود متغیر های مربوط به دما، رطوبت و یا نور را آپدیت میکنیم تا در خطوط بعدی روی صفحه نمایش داده شوند.

```
// Display sensor readings on the LCD
lcd.setCursor(0, 0);
lcd.print("LUX= ");
lcd.print(lux);

lcd.setCursor(0, 1);
lcd.print("temperature= ");
lcd.print(temprcher);

lcd.setCursor(0, 2);
lcd.print("humidity= ");
lcd.print(humidity);
```

در این بخش، به هر یک از متغیر هایی که داریم یک سطر اختصاص میدهم.

```
// Determine and display conditions based on sensor readings
C = 1;
if (humidity >= 80) {
  lcd.setCursor(0, 3);
  lcd.print("conditions = dont");
  C = 0;
} else if (humidity <= 50) {
  lcd.setCursor(0, 3);
  lcd.print("conditions = 15 cc/min ");
  C = 0;
} else if (C && temprcher <= 25 && lux <= 600) {
  lcd.setCursor(0, 3);
  lcd.print("conditions = 10 cc/min ");
  C = 0;
} else if (C && temprcher <= 25 && lux >= 600) {
  lcd.setCursor(0, 3);
  lcd.print("conditions = 5 cc/min ");
  C = 0;
} else if (C && temprcher >= 25 && lux >= 600) {
  lcd.setCursor(0, 3);
  lcd.print("conditions = 5 cc/min ");
  C = 0;
} else if (C && temprcher >= 25 && lux <= 600) {
  lcd.setCursor(0, 3);
  lcd.print("conditions = 10 cc/min ");
  C = 0;
}
}
```

در این خطوط صرفاً شرط‌هایی گذاشته شده تا خواسته‌های سوال برقرار شود.

:LDR-Board

```
void setup() {
  pinMode(A0, INPUT); // Sets pin A0 as input for LDR
  Serial.begin(9600); // Initializes serial communication at baud rate 9600
}
```

در این بخش پین متصل به سنسور را فعال میکنیم تا ADC آن ورودی آنالوگ سنسور را دریافت و آن را به ما بدهد. توجه کنید آردوینو یک ADC ده بیتی دارد که یعنی ولتاژ سورس تا زمین را به 1024 گام تقسیم میکند و از آنجا که منبع تغذیه ما 5 ولتی است در واقع تقریباً هر 5 میلی ولت را میتوانیم تمایز قائل شویم.

```
void loop() {
  LDR_value = analogRead(A0); // Read analog value from LDR (0-1023)

  // Calculate lux value using the formula based on LDR_value and ADC_value
  lux = (250.0 / (ADC_value * LDR_value)) - 50.0;

  // Prepare data to send over serial in the format "n lux_value n"
  String data = "n" + String(lux) + "n";

  // Send data over serial
  Serial.println(data);

  delay(25); // Delay for stability (adjust as needed)
}
```

توجه داشته باشید که خروجی سنسور یا ورودی اردوینو برابر LUX نمیباشد و همانطور که در کد نوشته شده است $(lux = \frac{250}{5 * LDR\ val} - 50)$. و نهایتاً از طریق سریال به main-board منتقل میشود.

:THRBoard

```
#include "Wire.h"      // Includes the Wire library for I2C communication
#include "SHT2x.h"      // Includes the SHT2x library for the sensor

SHT2x sht;             // Creates an instance of the SHT2x class
```

از آنجا که سنسور SHT2x از I2C استفاده میکند برای استفاده از آن باید wire.h را اضافه کنیم و همچنین برای سهولت کار با سنسور از لایبریری رسمی آن یعنی SHT2x.h استفاده میکنیم.

```
void setup() {
    Serial.begin(9600); // Initializes serial communication at baud rate 9600
}
```

پورت سریال را برای اتصال به main-board باز میکنیم.

```
void loop() {
    bool success = sht.read(); // Attempts to read data from the sensor

    if (!success) {
        Serial.println("Failed to read from sensor"); // Prints an error message if the sen
    } else {
        float temperature = sht.getTemperature(); // Retrieves the temperature value from
        float humidity = sht.getHumidity();       // Retrieves the humidity value from the

        // Prepares the data string in the format "temperature,humidity"
        String dataToSend = String(temperature) + "," + String(humidity);
        Serial.println(dataToSend); // Sends the data string over the serial port
    }
}
```

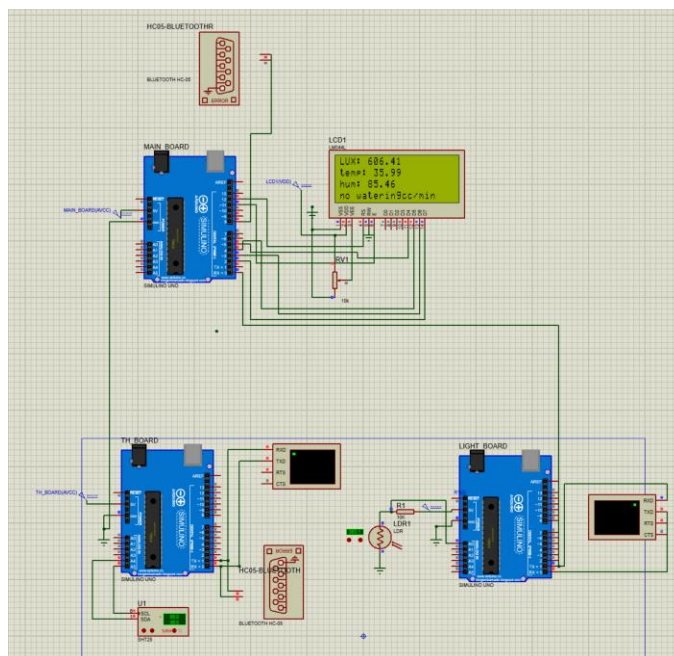
اکنون داده را از روی سنسور خوانده و با فرمت CSV روی سریال ارسال میکنیم.

خروجی شبیه سازی:

پیش از شروع شبیه سازی تنظیمات مربوط به HHD Virtual Serial Port Tools را انجام میدهیم.

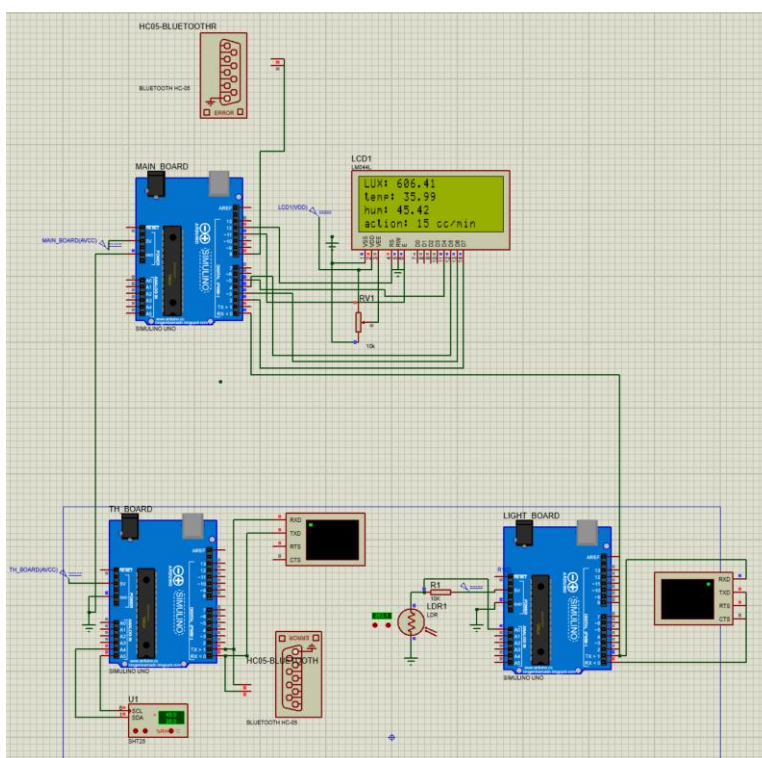
حالت اول:

رطوبت بالای هشتاد درصد:



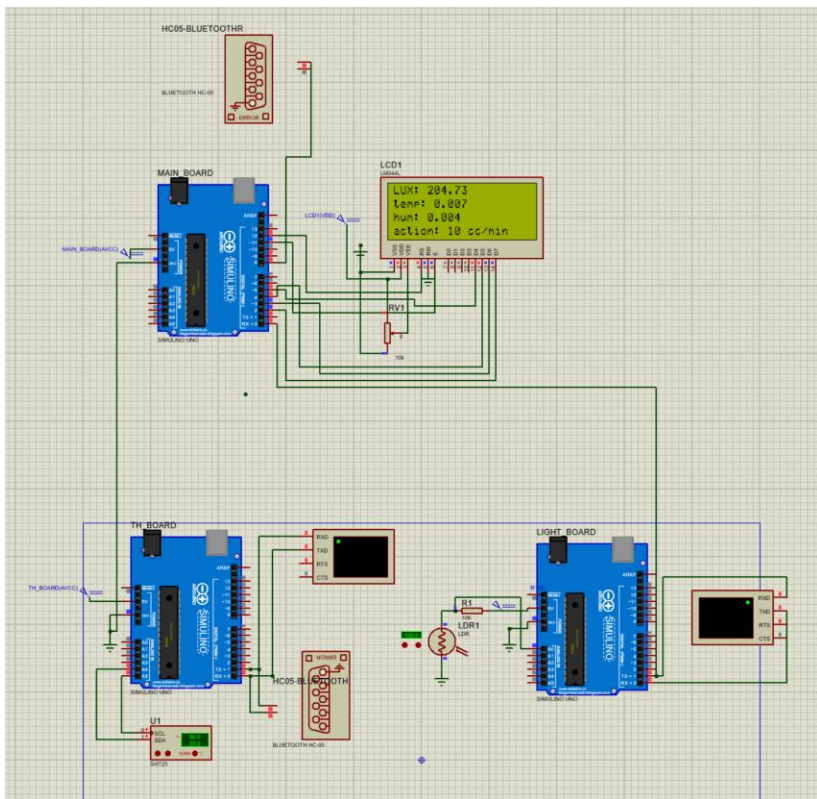
حالت دوم:

رطوبت کمتر از 50 درصد



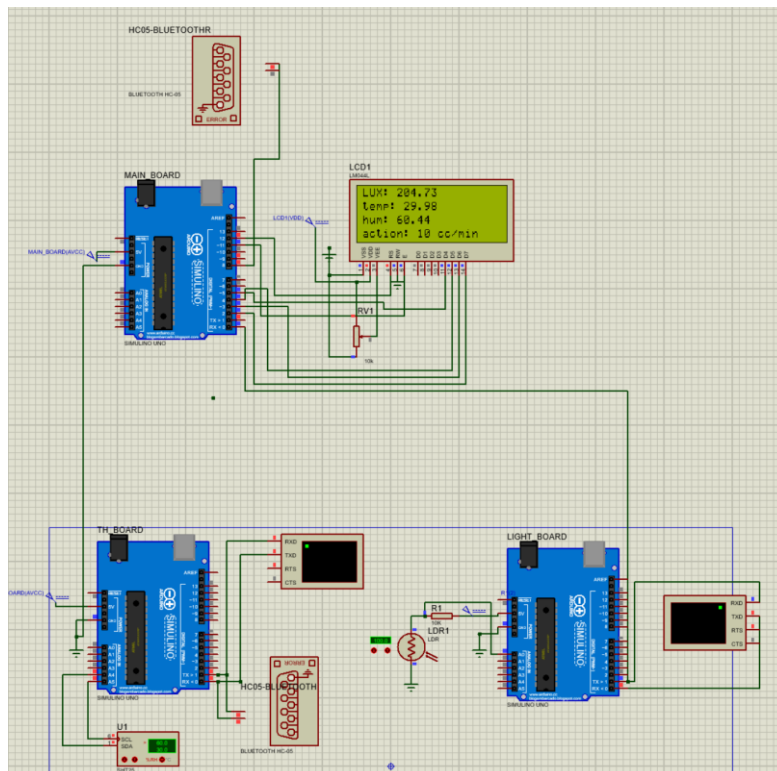
حالت سوم:

دما کمتر از 25 درجه و نور کمتر از 600:



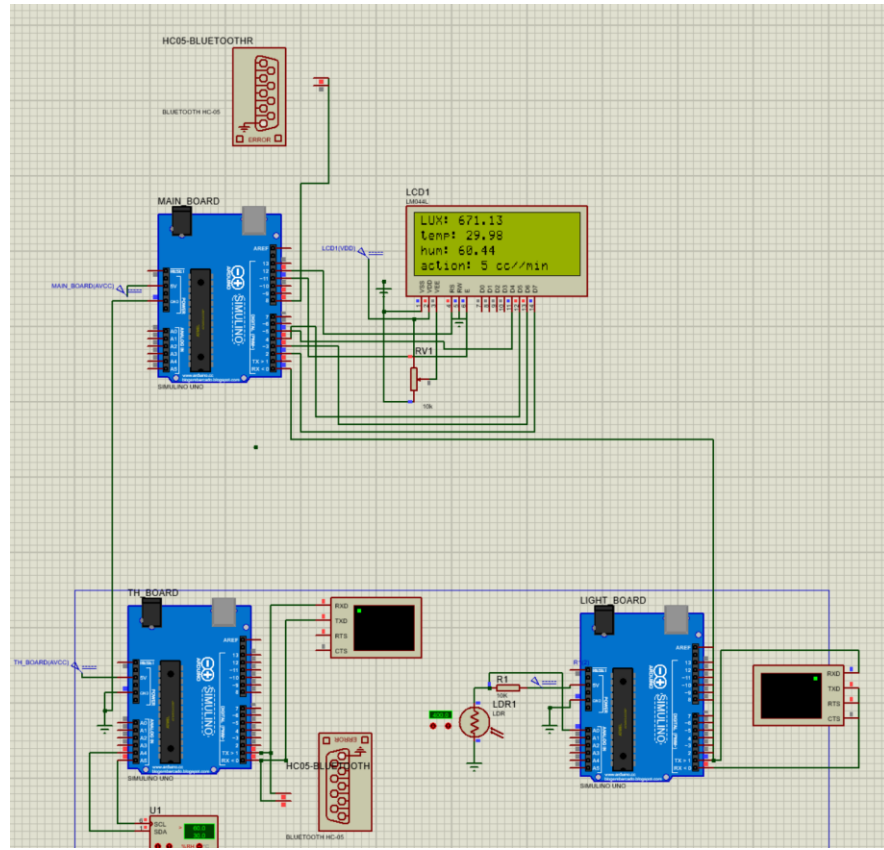
حالت چهارم:

دما بیش از 25 درجه و نور کمتر از 600:



حالت پنجم:

دما بیش از 25 نوز بیشتر از 600:



حالت ششم:

دما کمتر از 25 نور بیشتر از 600:

