Department of Computer Engineering

CSE4020 Embedded Systems
Project Report

Supervisor : WİSAM ELMASRY

Date : 16.01.2024

Multipurpose Reconnaissance Vehicle

Students :

	Name & Surname
	Semir Kimyonşen
	Alparslan Eravşar
	Adem Özkayıt

Project Description:

Project Title: Multipurpose Reconnaissance Vehicle

Project Description:

This project encompasses a Multipurpose Reconnaissance Vehicle designed for versatile exploration on the surface of Planet. The vehicle is equipped with various sensors and communication antennas to monitor diverse features and atmospheric conditions on Planet. Additionally, it includes a set of sensors to assess environmental conditions and respond to specific tasks.

Key Features:

1. Communication Antennas:
    - The vehicle is equipped with two communication antennas.
    - The control center is equipped with two communication antennas.
    - Antenna 1 and Antenna 2 establish 1-to-1 communication between the vehicle and the control center.
    - Antenna 3 and Antenna 4 communicate with other remote sensors or exploration equipment.

2. Soil Moisture Sensor (SOIL SENSOR):
    - Measures soil moisture levels on the Planet surface.

- Used for analyzing the potential for plant growth and assessing local environmental conditions.

3. Motion Sensor:
    - Detects movements in the vehicle's surroundings.
    - Identifies potential hazards and takes safety measures.
4. Temperature and Humidity Sensor (DHT11):
    - Measures temperature and humidity levels in the Planet atmosphere.
    - Evaluates atmospheric conditions and optimizes the vehicle's operational parameters.
5. Gas Detection Sensor (MQ-3):
    - Detects specific gases in the Planet atmosphere.
    - Identifies potential hazardous gases and sends to control center precautions.
6. Water Level Sensor (WATER SENSOR):
    - Measures precipitation on the Planet surface.
    - Used for detecting water and precipitation the potential presence of liquid water.
7. Light Detection Sensor (LDR):
    - Measures light levels on the Planet surface.
    - Monitors day and night conditions and optimizes the vehicle's energy management.

Project Objectives:

- Monitor and analyze environmental conditions on the Planet surface.
- Detect potential precipitations.
- Collect and assess atmospheric data.
- Ensure safety by monitoring movements in the vehicle's surroundings.
- Maintain continuous communication with the control center through communication antennas.

Components:

Components	Quantity
Arduino Uno Board	4

LDR	1
Rain Sensor	1
DHT 11 Sensor	1
MQ-3	1
SOIL	1
PIR	1
16x2 LCD	1
Big Size Breadboard	3
Middle Size Breadboard	1
Nrf24L01(Antenna)	4
Nrf24L01(Adapter)	4
330 Ohm Resistor	6
USB CABLE	4
LED GREEN	2
Buzzer	1
Potentiometer	1
Servo Motor	2
DC Motor	1
Joystick	1
Button	1
NFC	1
Jumper Cables (M-M)	21
Jumper Cables (F-M)	47
Jumper Cables (F-F)	14

Design

Measure

Lcd port 1 is connected to gnd Lcd Port 2 is connected to vcc Lcd Port3 is connected to potantimeter port 2 Lcd Port4 is connected to arduino port 7 Lcd Port5 is connected to gnd Lcd Port6 is connected to Arduino port 6 Lcd Port 11 is connected to Arduino port 5 Lcd Port 12 is connected to Arduino port 4 Lcd Port 13 is connected to Arduino port 3 Lcd Port 14 is connected to Arduino port 2 Lcd Port 15 is connected to vcc with resistor Lcd Port 16 is connected to gnd

Potantiometer Port 1 is connected to vcc Potantiometer Port 2 is connected to LCD Port3 Potantiometer Port 3 is connected to gnd Buzzer + is connected to Arduino 10 Buzzer - is connected to gnd

From Arduino volt 5 we transferred power to board. From Arduino GND we transferred gnd to board.

Antenna vcc is connected to 5v Antenna gnd is connected to gnd Antenna CE is connected to port

9 Antenna CSN is connected to port 8 Antenna Sck is connected to port 13 Antenna MO is connected to port 11
Antenna MISO is connected to port 12

Sensor Part

Antenna vcc is connected to 5v Antenna gnd is connected to gnd Antenna CE is connected to port 9 Antenna CSN is connected to port 8 Antenna Sck is connected to port 13 Antenna MO is connected to port 11
Antenna MISO is connected to port 12

Water Sensor S pin is connected to Arduino A1 Water Sensor + is connected to Arduino + Water Sensor - is connected to Arduino –

Motion Sensor + is connected to Arduino + Motion Sensor - is connected to Arduino – Motion Sensor data pin is connected to Arduino 4

MQ sensor DO pini is connected to Digital 3 MQ sensor GND pini is connected to GND MQ sensor VCC pini is connected to VCC

DHT11 sensor data pin is connected to Arduino 2 DHT11 sensor GND pini is connected to GND DHT11 sensor VCC pini is connected to VCC

SOIL sensor analog pin is connected to Arduino analog 2 SOIL sensor + pin is connected to Arduino + SOIL sensor - pin is connected to Arduino –

LDR sensor leg 1 is connected to gnd with resistor and connected A0 with jumper LDR sensor leg 2 is connected to vcc

From Arduino volt 5 we transferred power to board.
From Arduino GND we transferred gnd to board.

Moment Part

DC motor leg 1 is connected to diot and this leg connected to VIR DC motor leg 2 is connected to diot leg and this leg connected to transistor

transistor middle le connected to Arduino 3 with resistor.
Other transistor leg connected to gnd

Servor motor data pin connected to Arduino 6 Servor motor gnd connected to Arduino gnd Servor motor vcc connected to Arduino vcc

Servor motor data pin connected to ardiuno 5 Servor motor gnd connected to Arduino gnd Servor motor vcc connected to Arduino vcc

Led one leg connected to gnd with resistor We connected one leg of the leds to arduino 2 with serial connection

Antenna vcc is connected to 5v Antenna gnd is connected to gnd Antenna CE is connected to port 9 Antenna CSN is connected to port 8 Antenna Sck is connected to port 13 Antenna MO is connected to port 11
Antenna MISO is connected to port 12

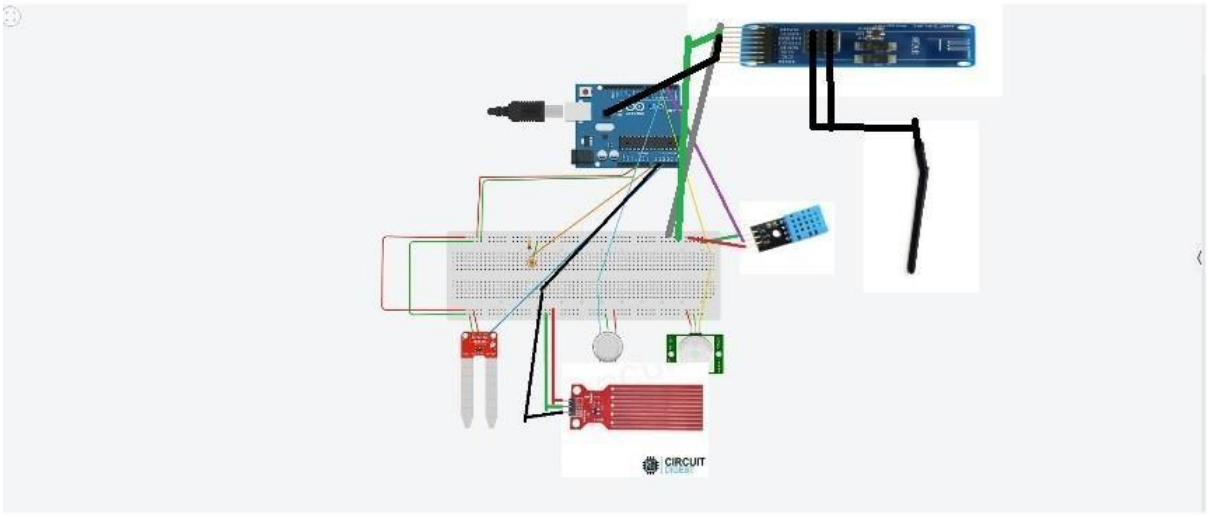
Moment-Controller Part

Antenna vcc is connected to 5v Antenna gnd is connected to gnd Antenna CE is connected to port 9 Antenna CSN is connected to port 8 Antenna Sck is connected to port 13 Antenna MO is connected to port 11
Antenna MISO is connected to port 12

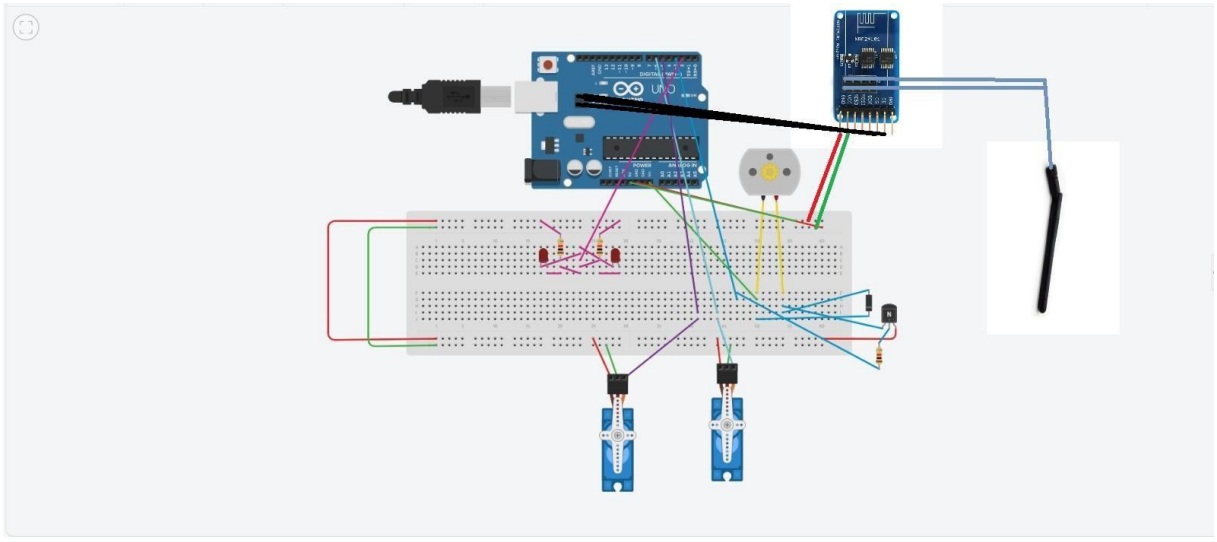
Button one leg connected to Arduino 4 Button leg connected to gnd with resistor Button leg connected to vcc

Joystick gnd pin connected to gnd to gnd Joystick +5v pin connected to vcc Joystick VRx pin connected to Arduino A0 Joystick VRy pin conencted to Arduino A1 Joystick SW pin connected to Arduino 2

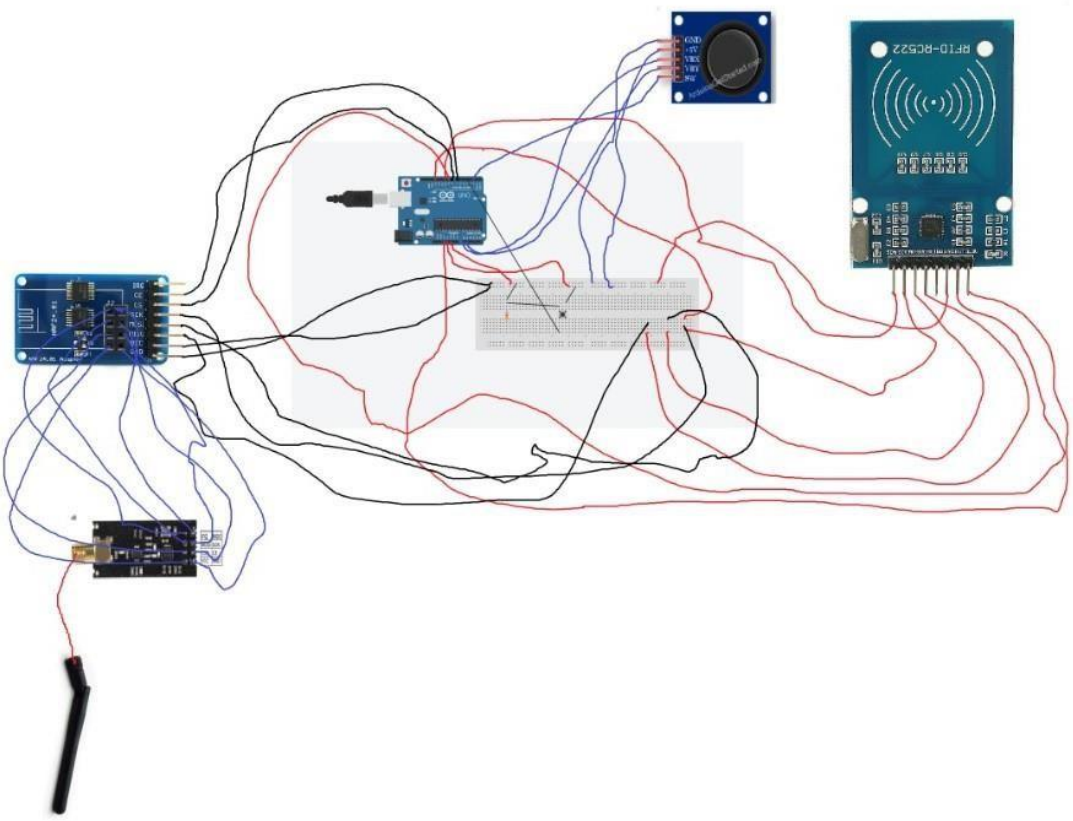
NFC 3.3v pin connected to Arduino 3.3V NFC reset pin connected to Arduino 9 NFC gnd pin connected to gnd NFC mıso mosı sck sda connected to ISCP pins Arduino



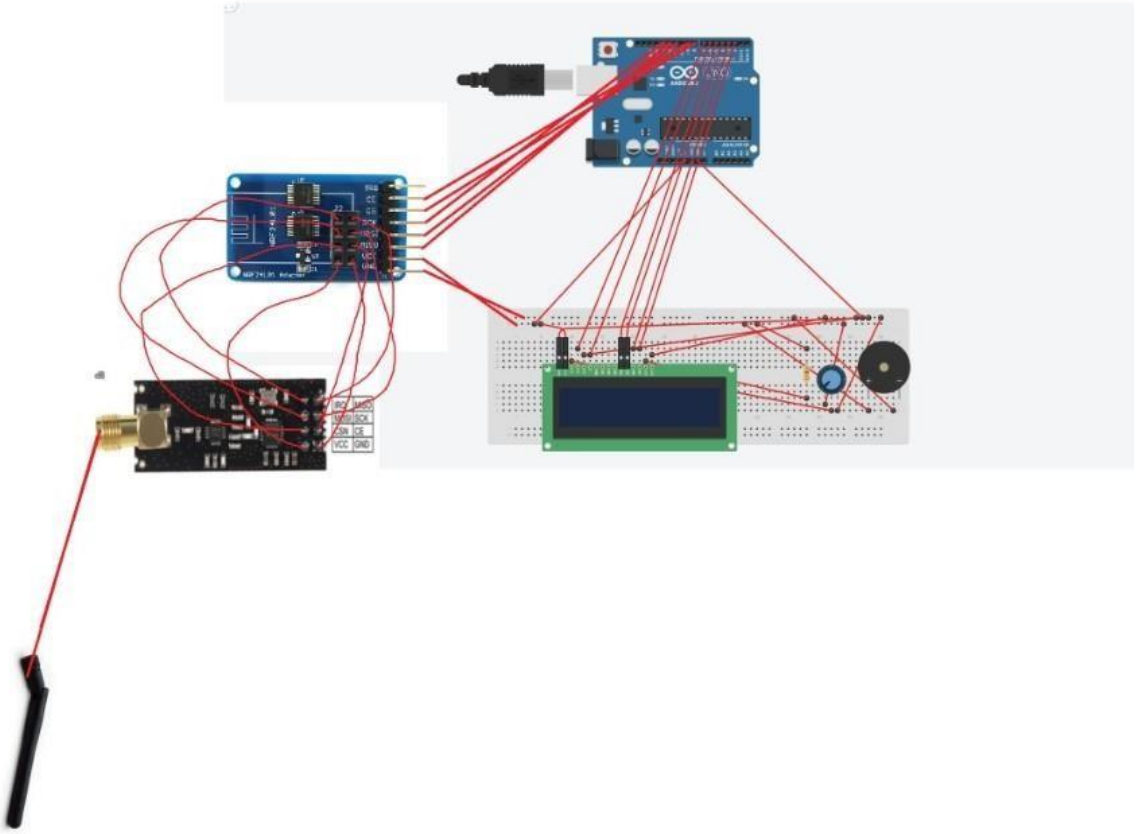
Connection diagram of sensors located at the top of the vehicle. (Figure-1)



The connection diagram of components that enable the vehicle to move. (Figure-2)

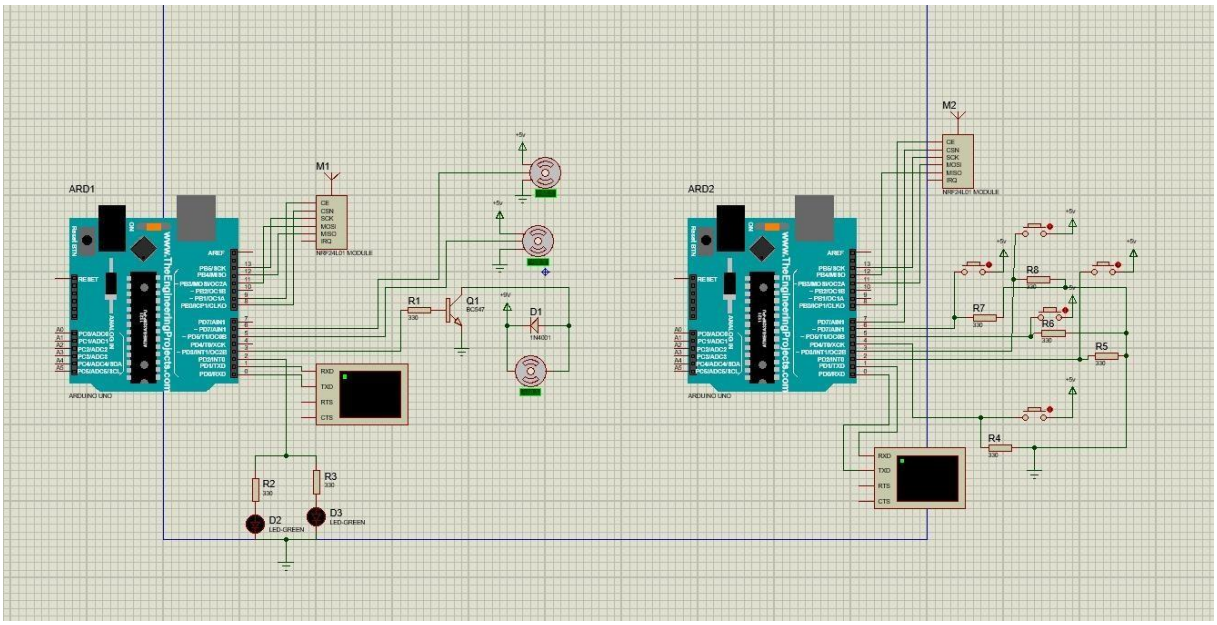


The connection diagram of the system controlling the vehicle's motion. (Figure-3)

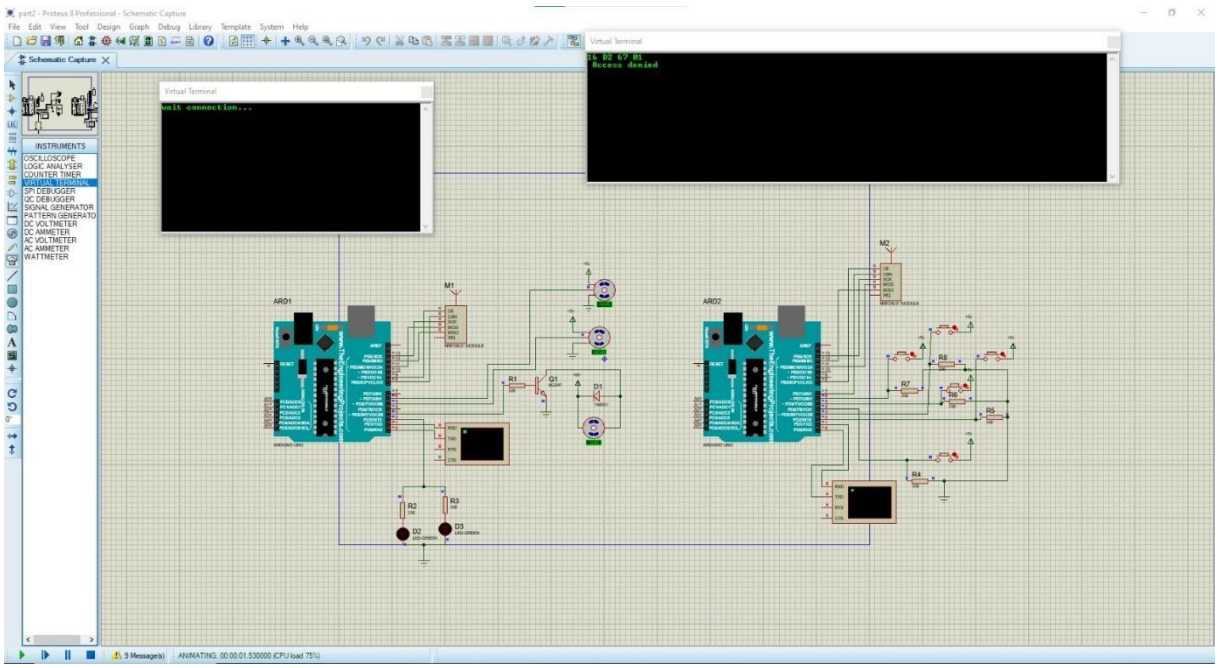


The connection diagram of the system where data from sensors is processed. (Figure-4)

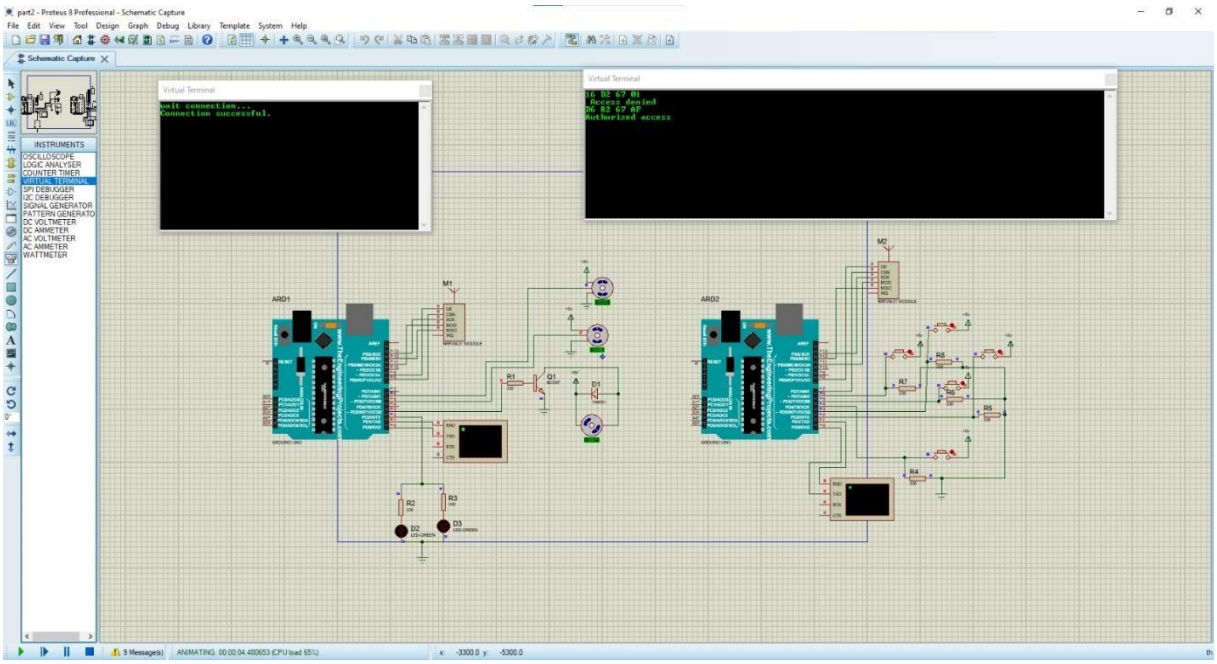
1 is connected to 4, and 2 is connected to 3.



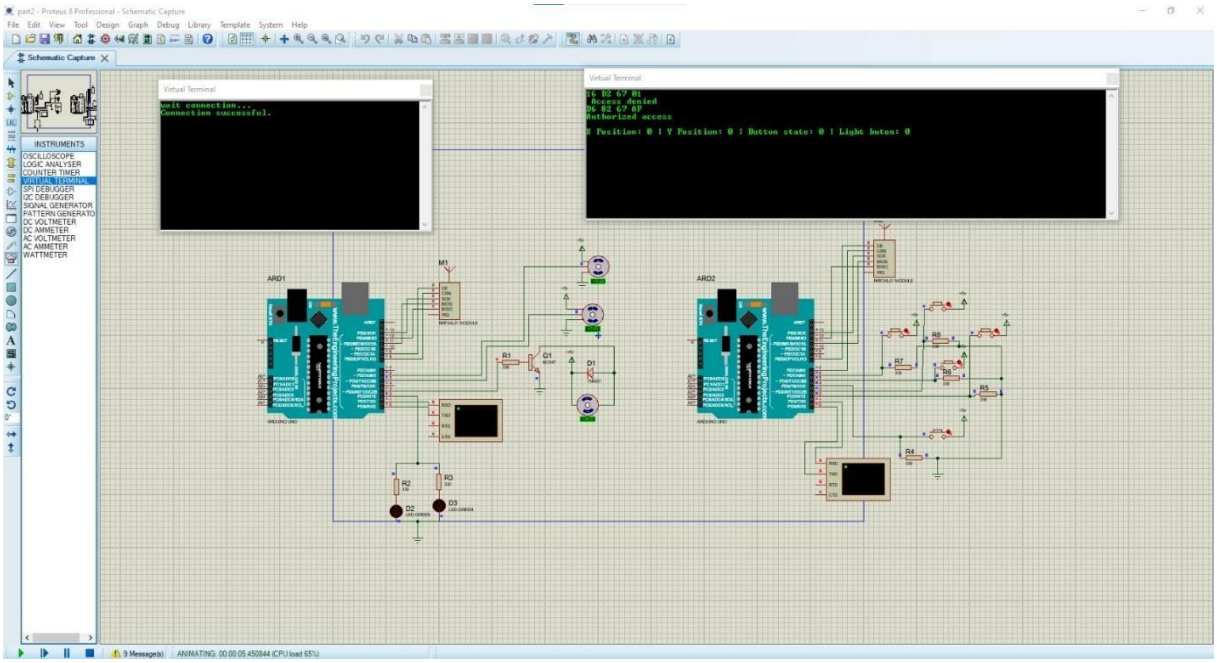
Proteus schematic of the system enabling the vehicle's motion.



The initial state when the system is started and when an invalid card is applied.

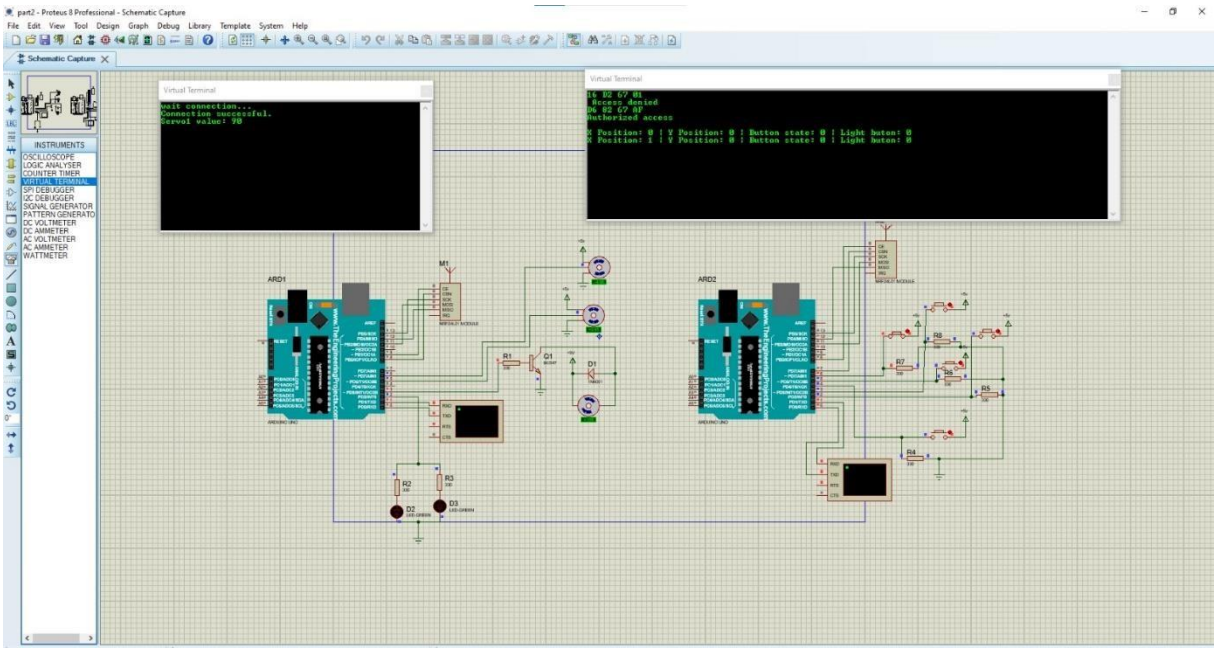


The initial state when the system is started and when a valid card is applied.

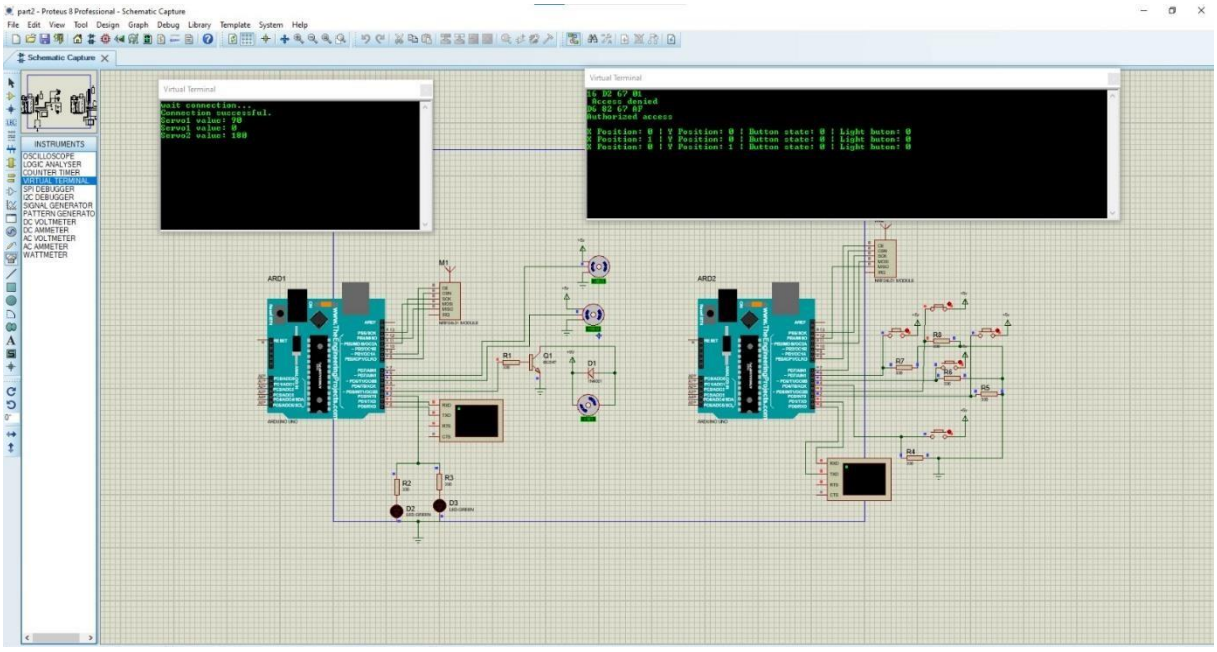


Buttons are not pressed,

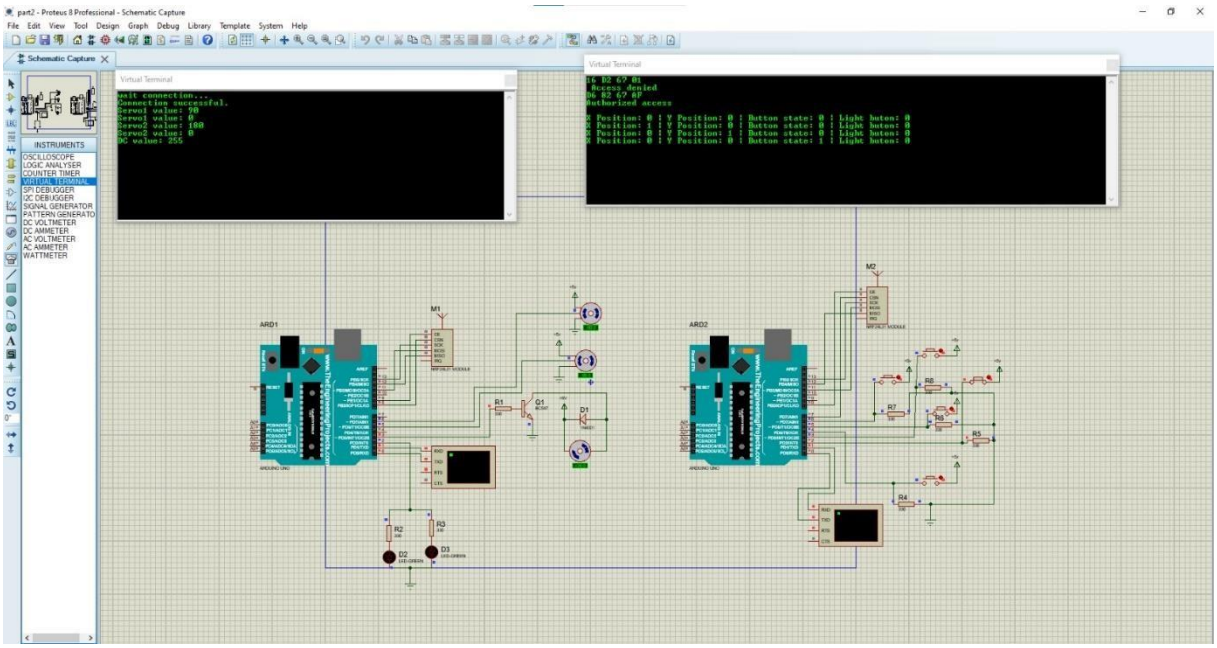
Note: Buttons are used to simulate the joystick.



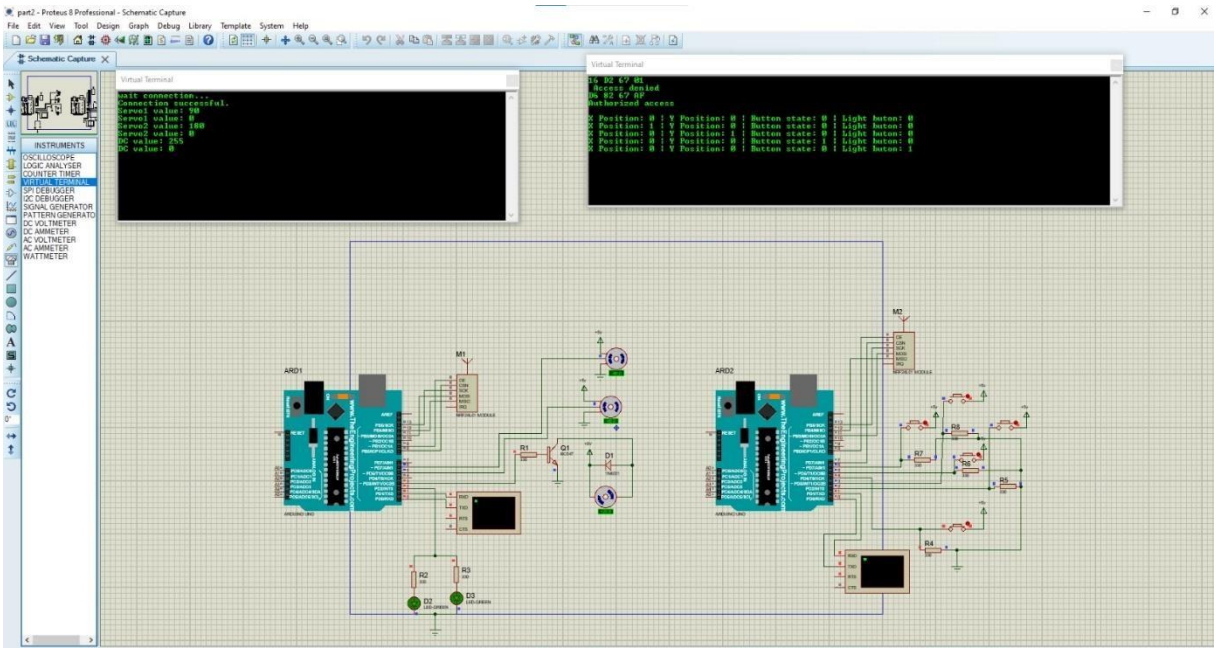
When the joystick is moved along the X-axis.
Servo motor start to 0-90 degree



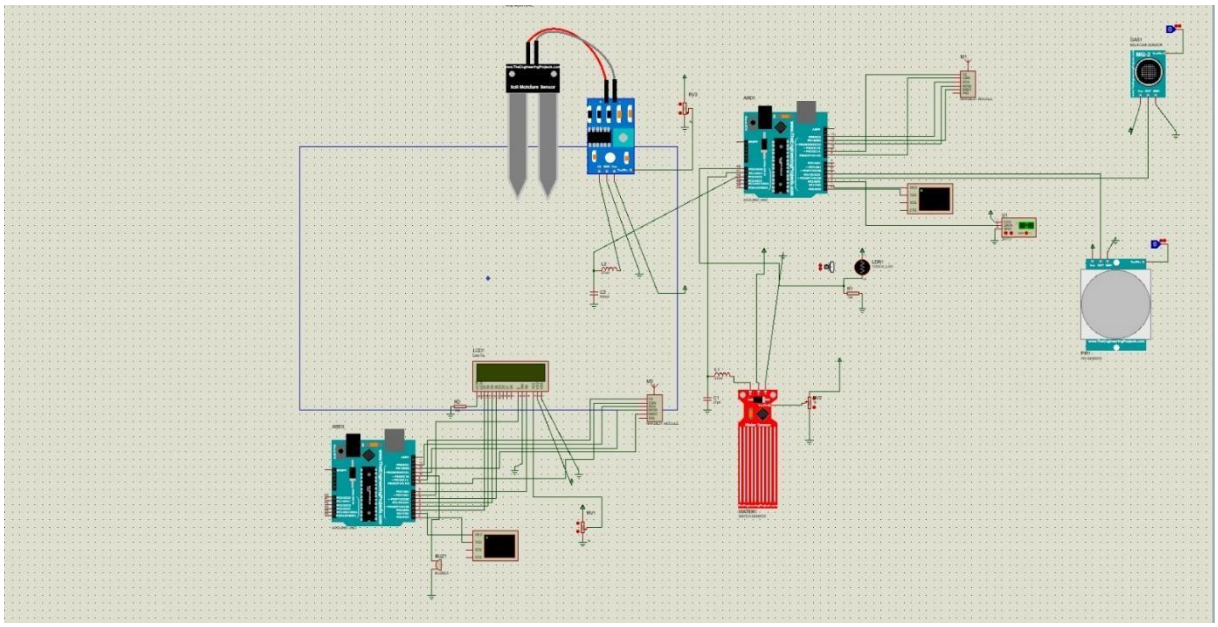
When the joystick button is pressed, it initiates the movement of the servo motor.



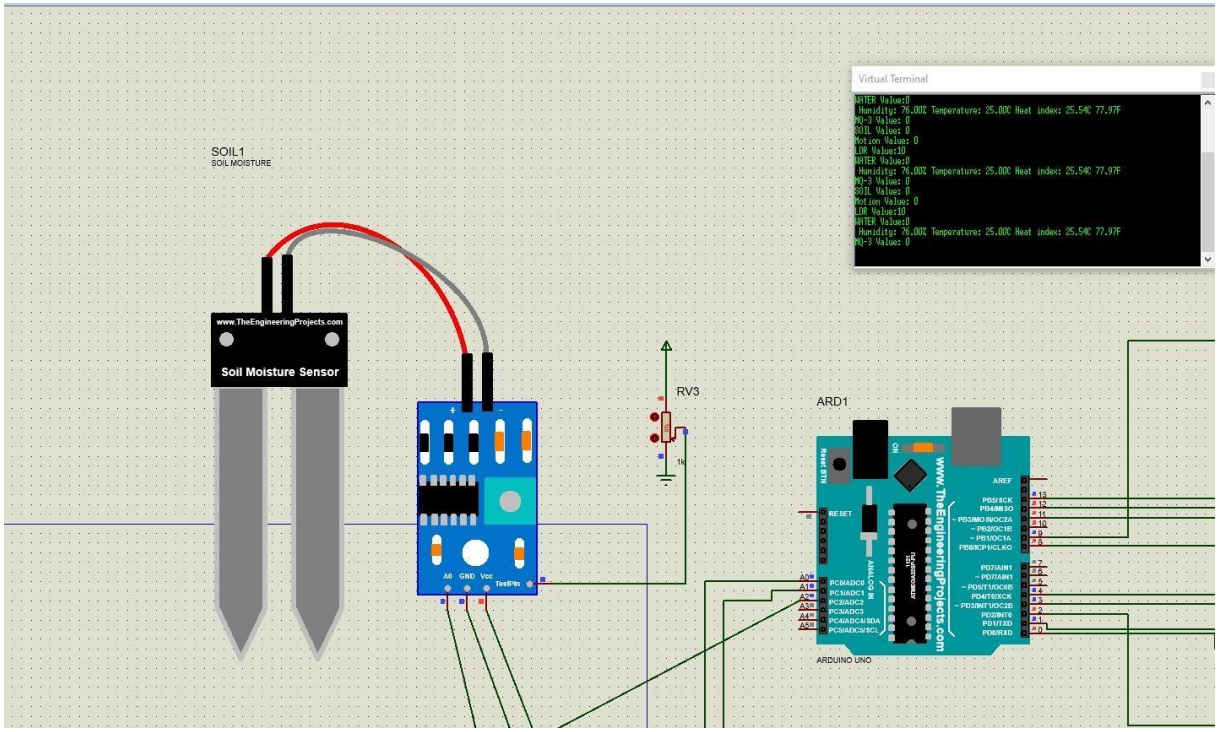
When the joystick is moved along the Y-axis. DC motor start to move



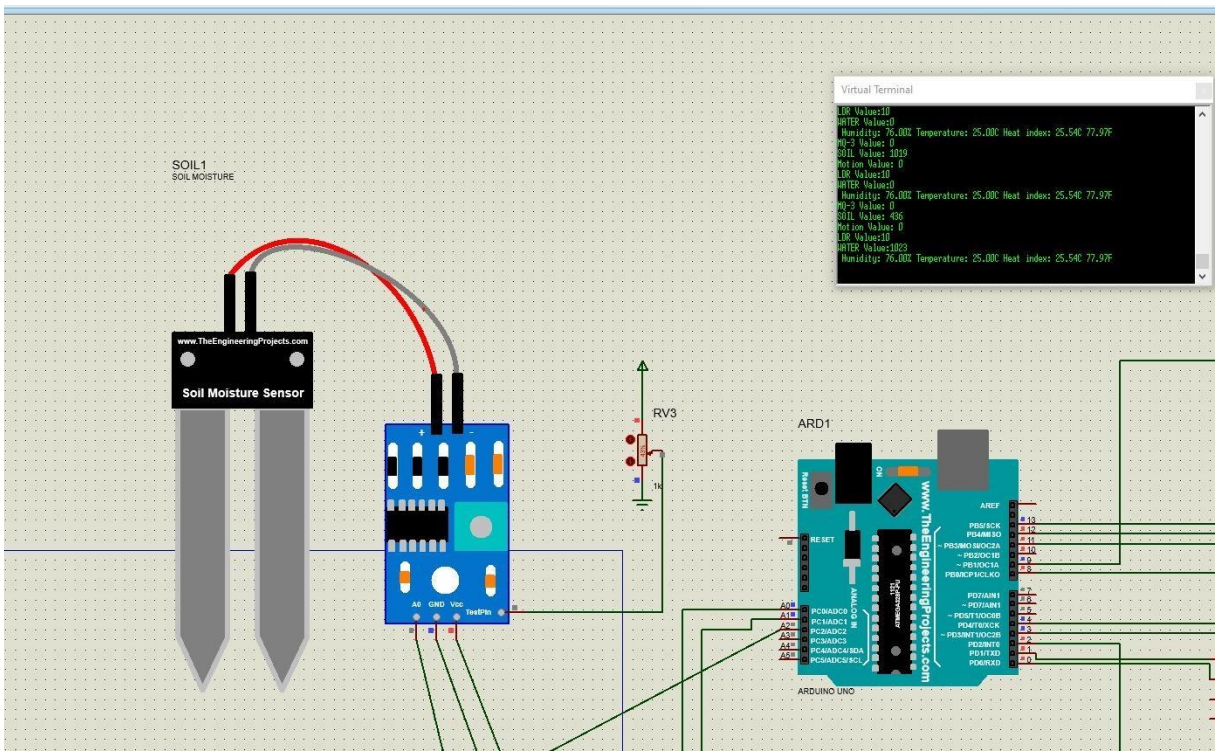
The lights turn on when the button is pressed.



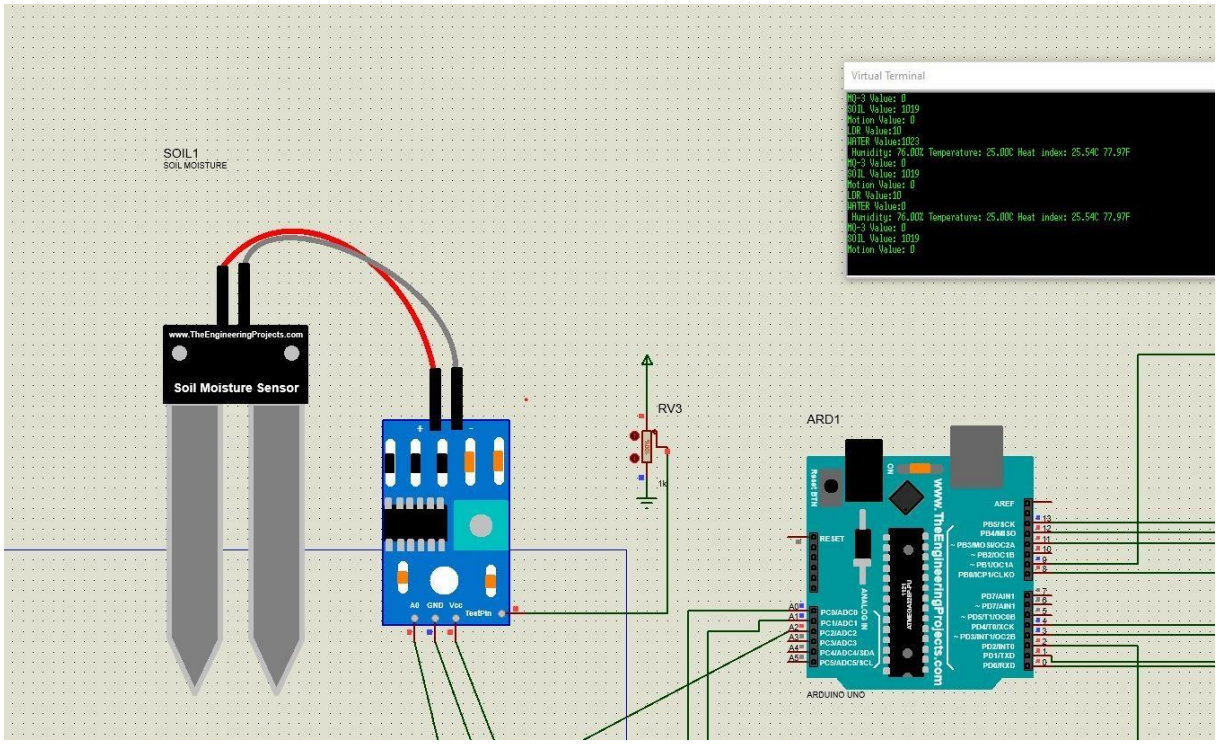
Overall image of our second system



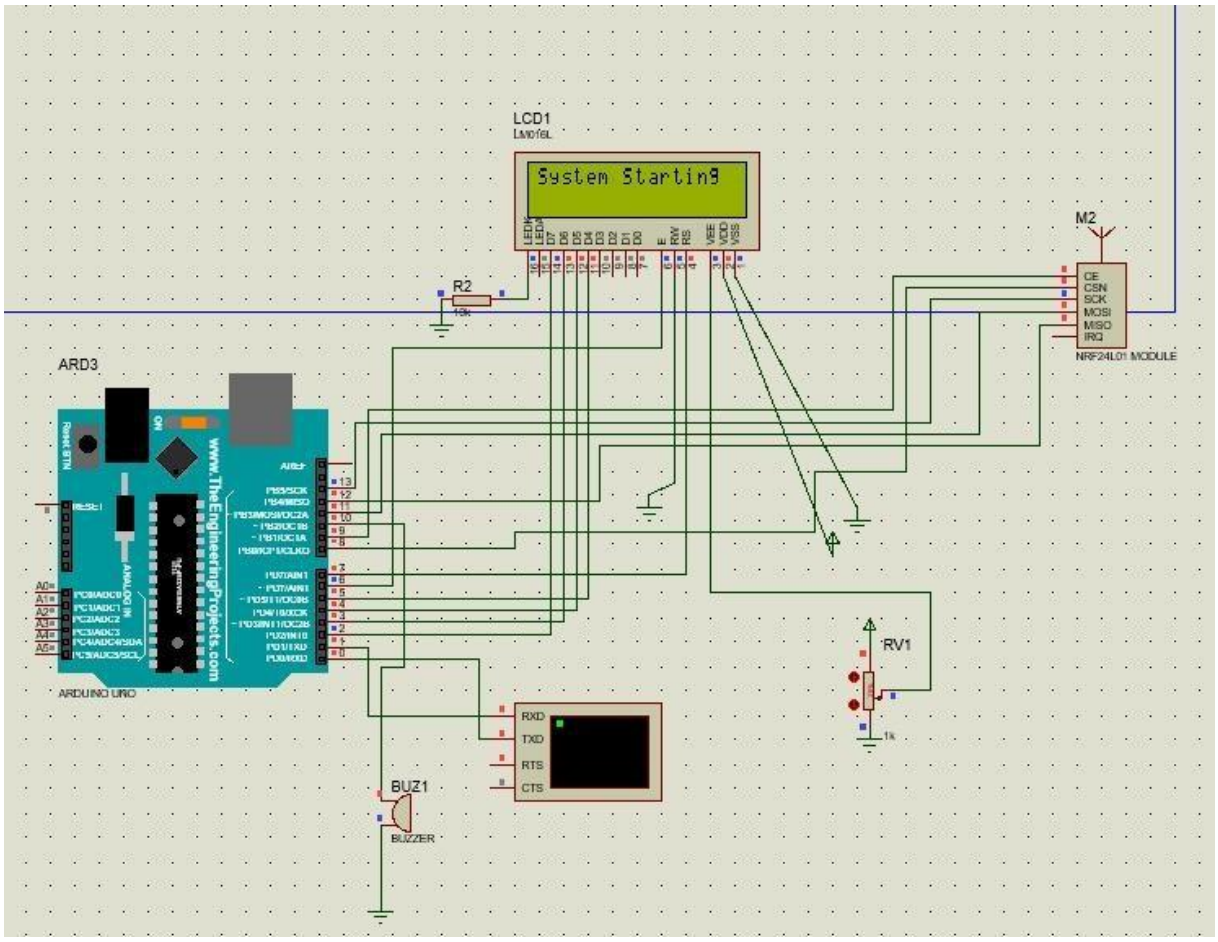
Initial state of the soil moisture sensor (value=0 state)



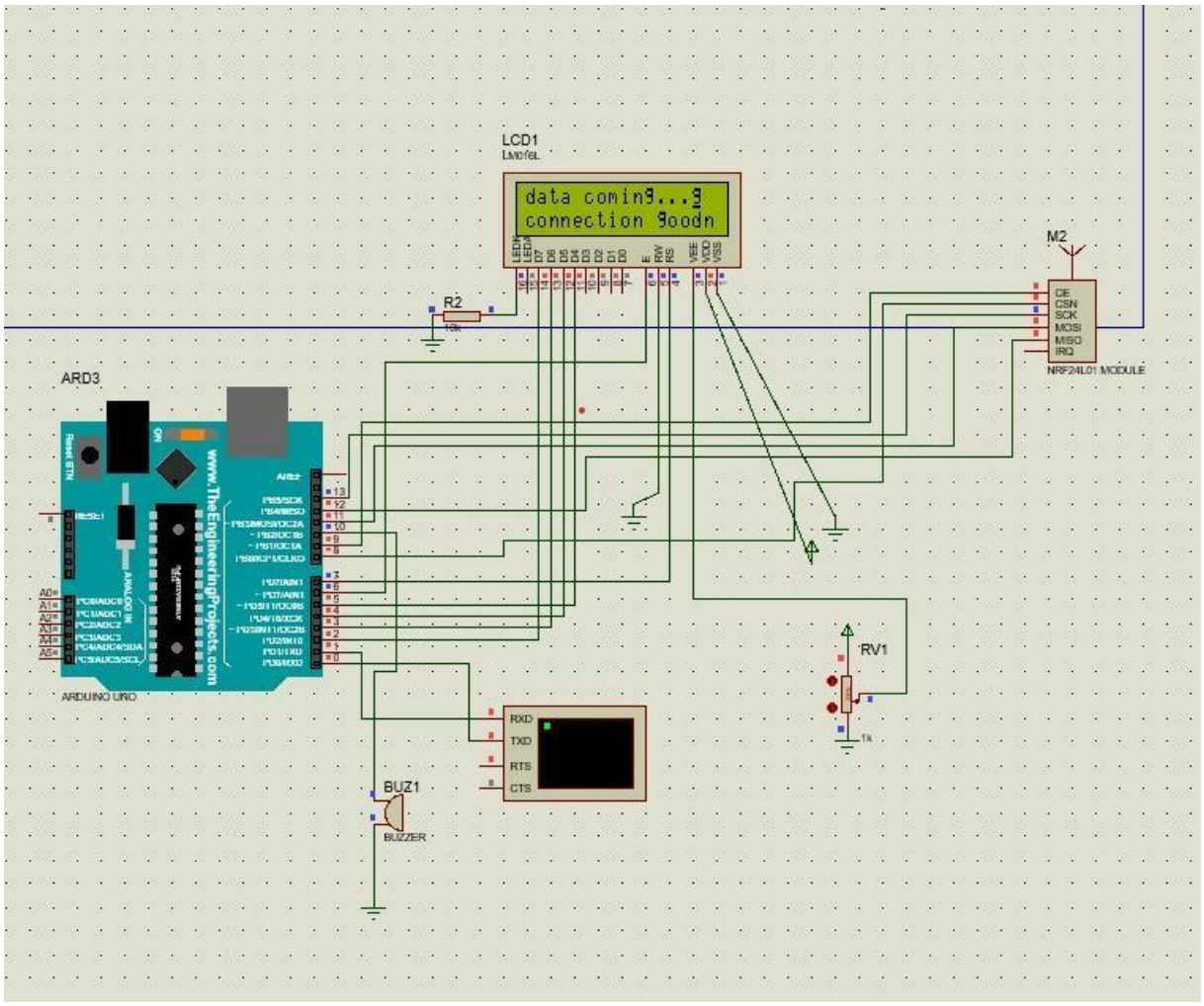
Soil moisture sensor takes a middle value (value=436 state)



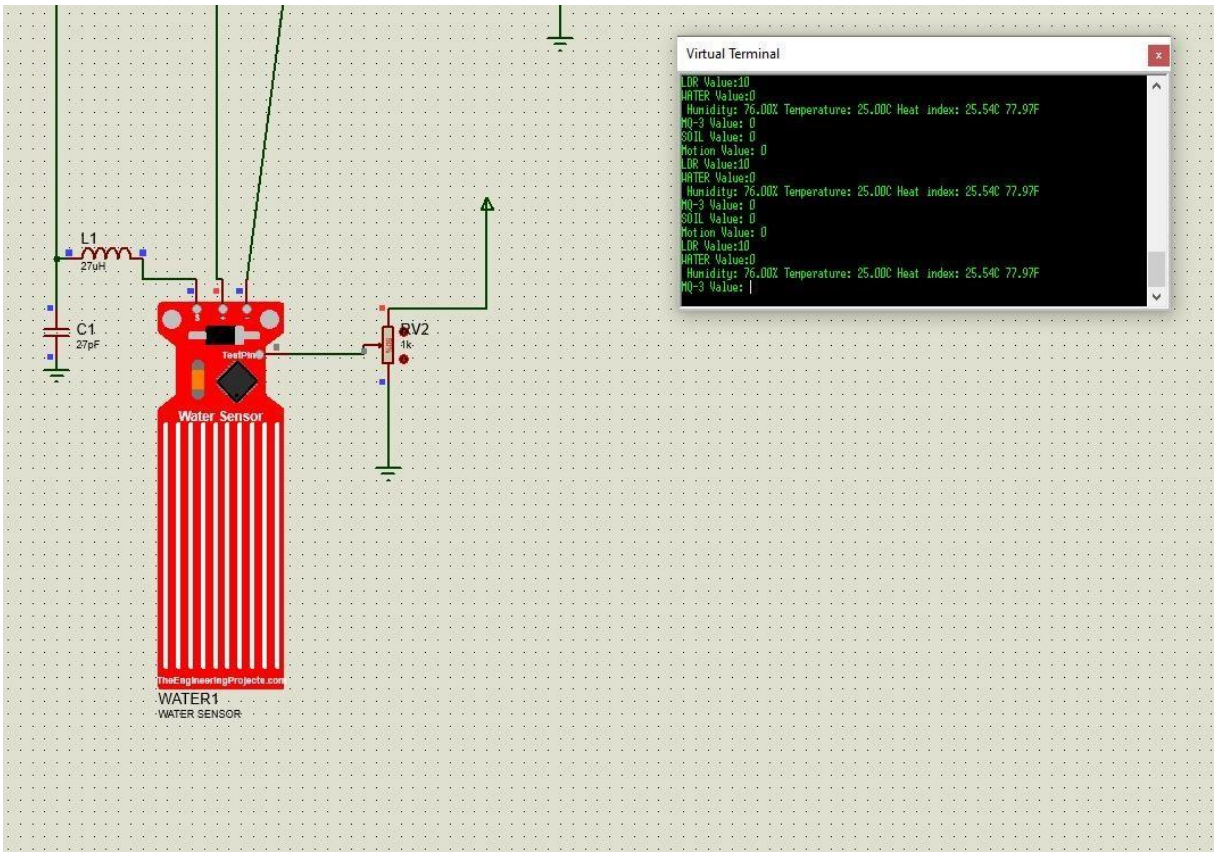
Soil moisture sensor maximum value state (value=1023)



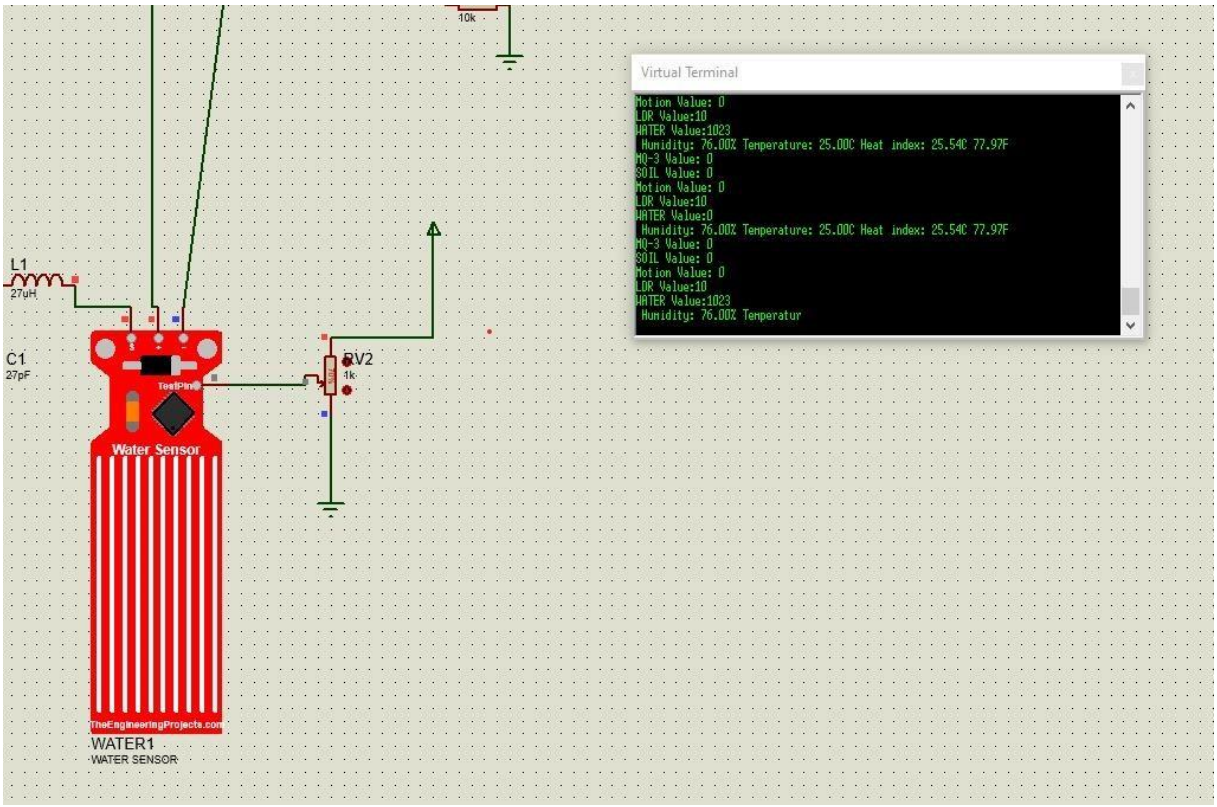
Receiver part of our second system while initialized



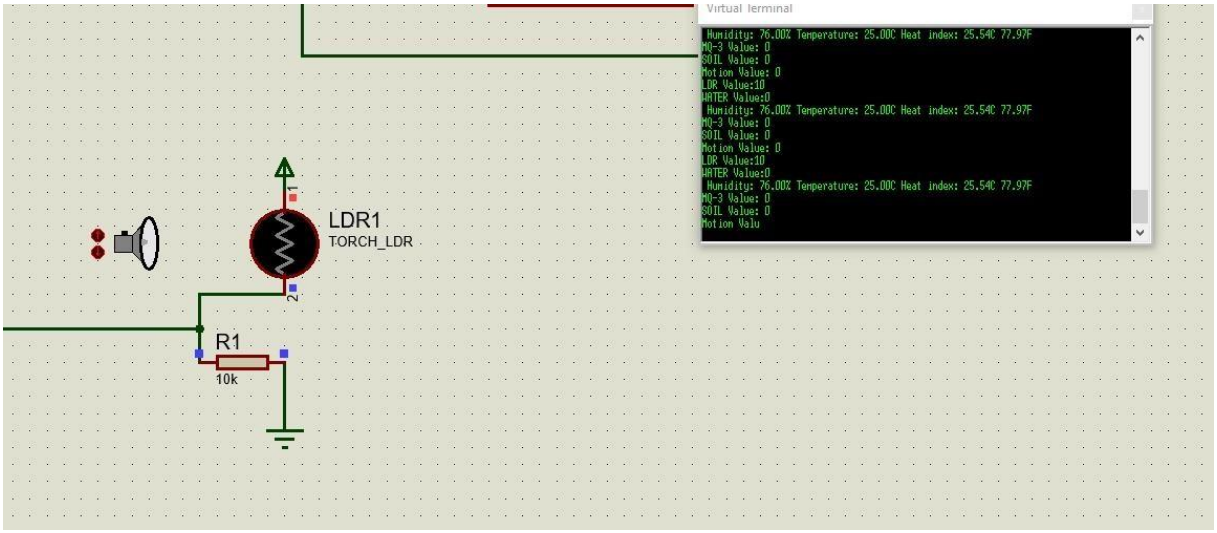
Receiver part of our second system while running



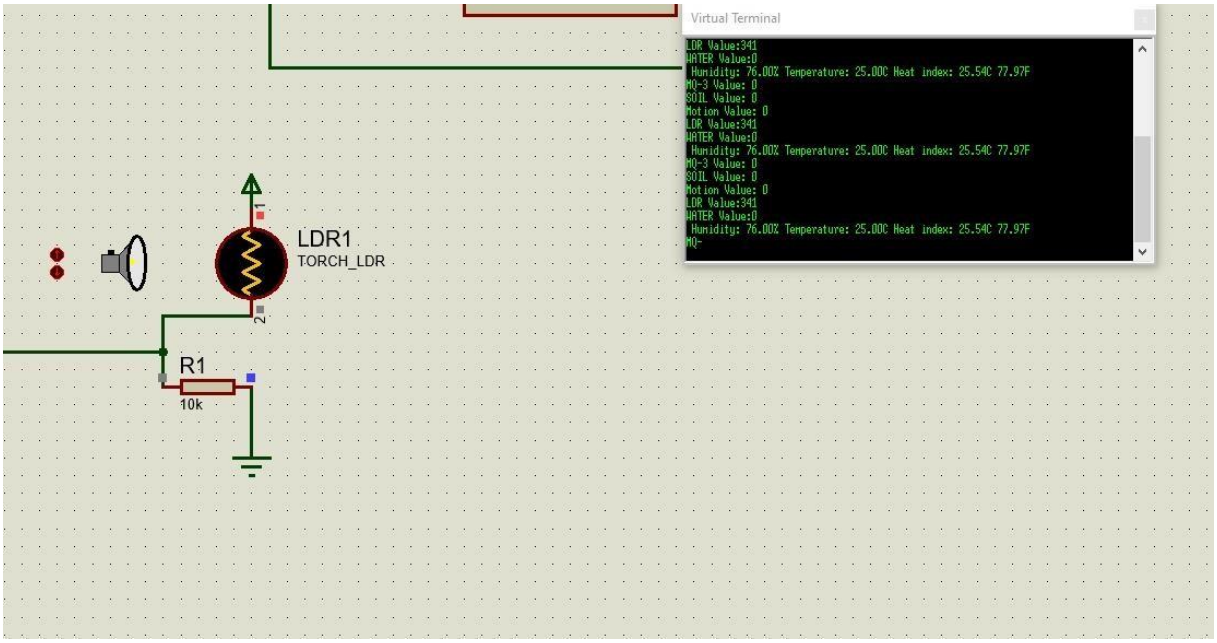
Water sensor when initialized (value=0 state)



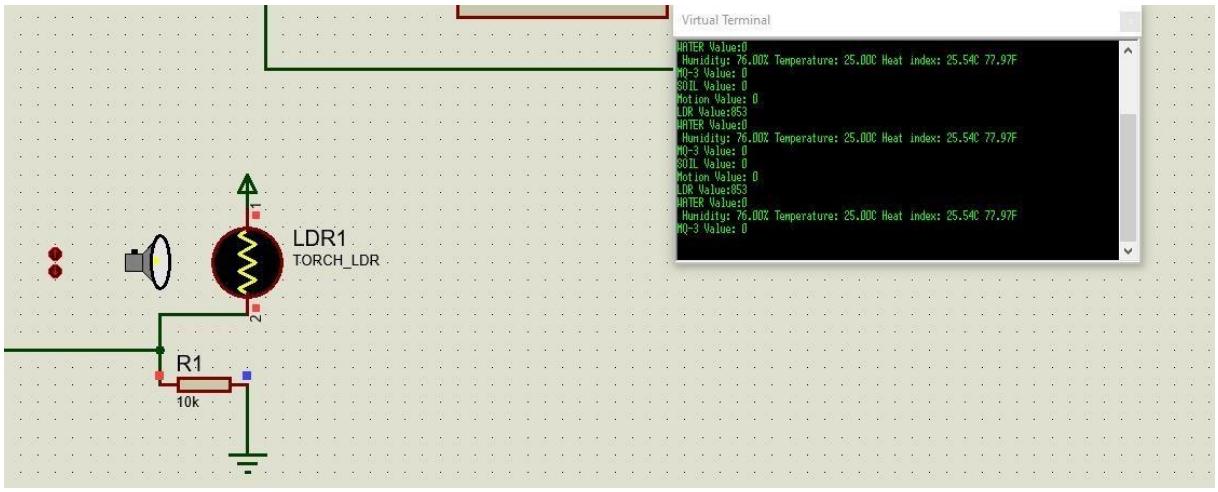
Water sensor in max value state (value=1023 state)



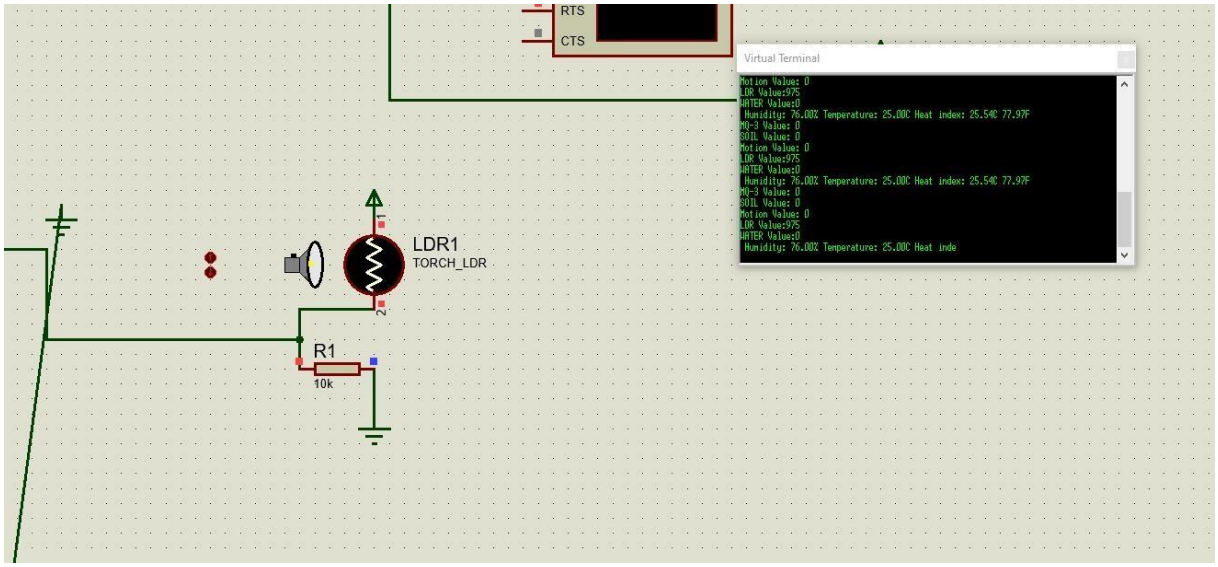
LDR Sensor when initialized (value=10 state)



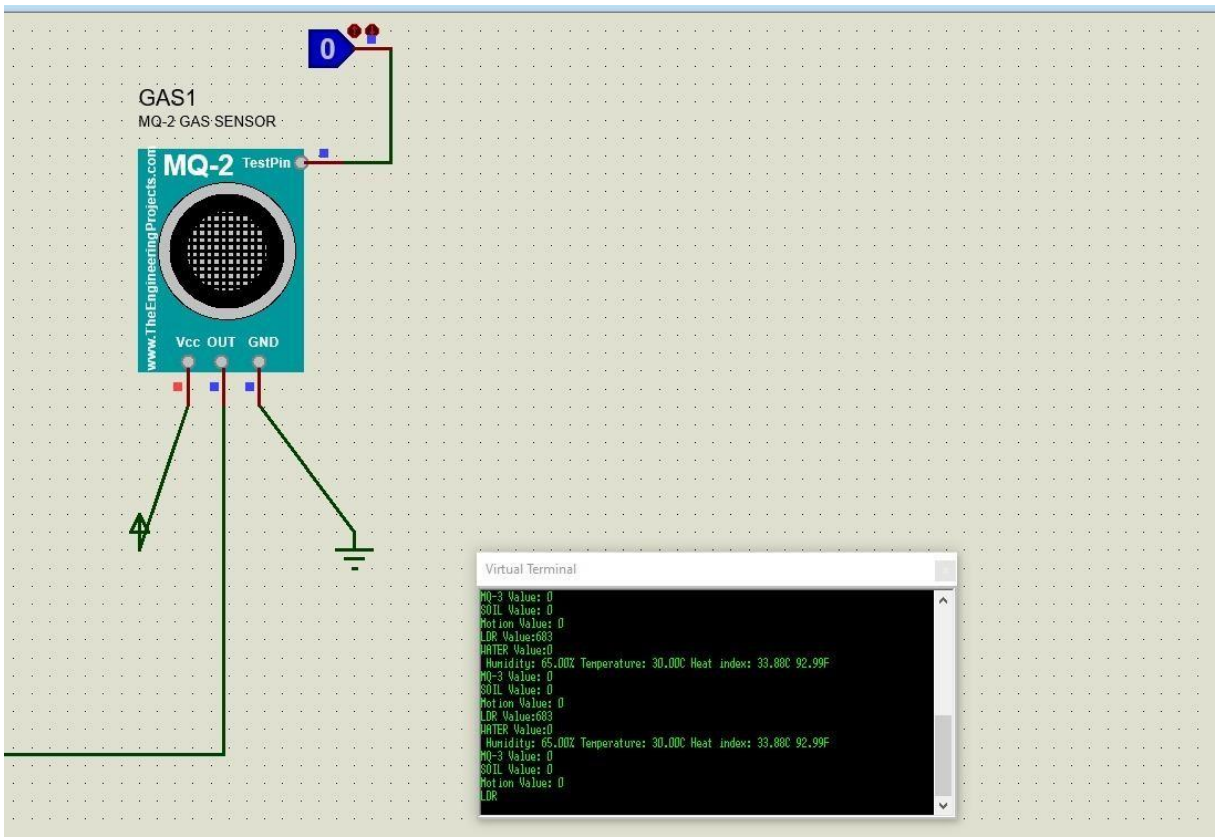
LDR Sensor when takes an average value (value=341 state)



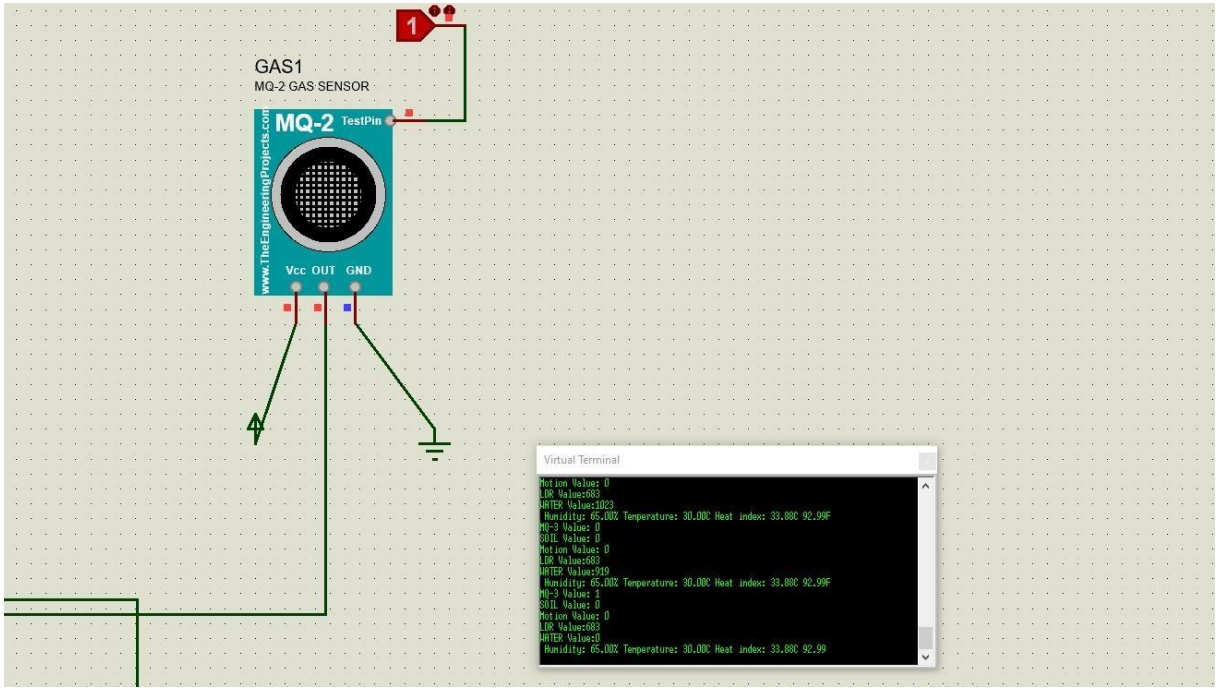
LDR Sensor when takes a upper-middle value between max value (value=853 state)



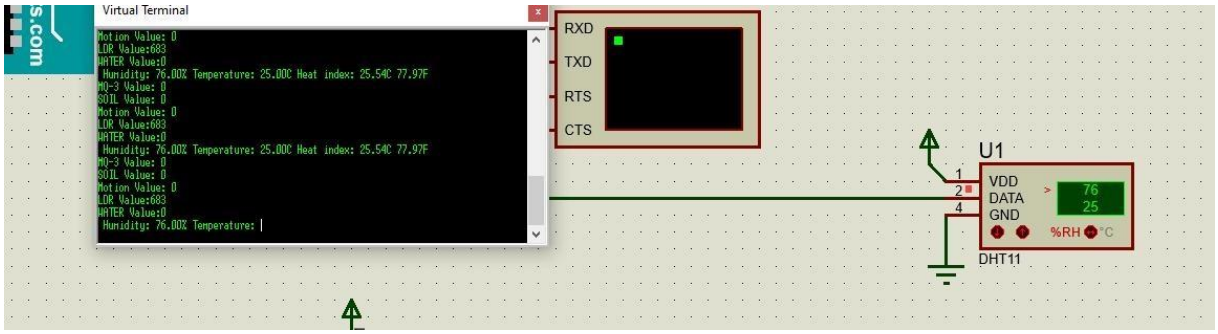
LDR Sensor when takes max value (value=975 state)



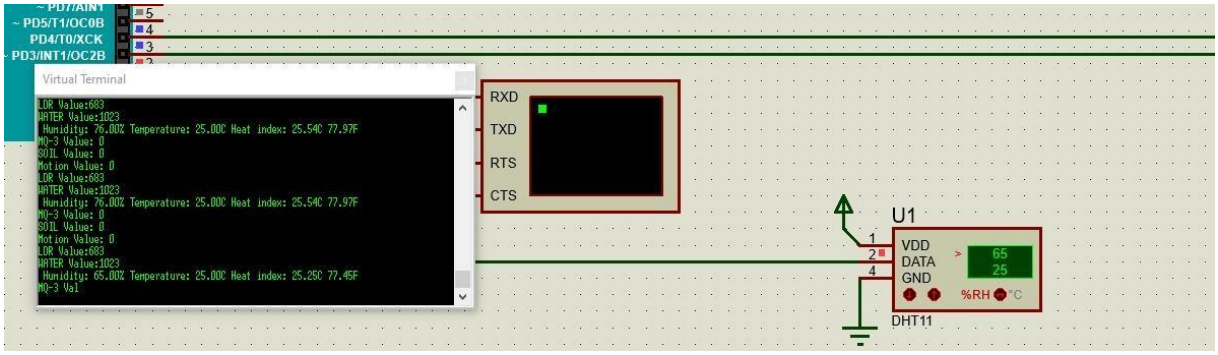
MQ-2 GAS Sensor when initialized (value=0 state)



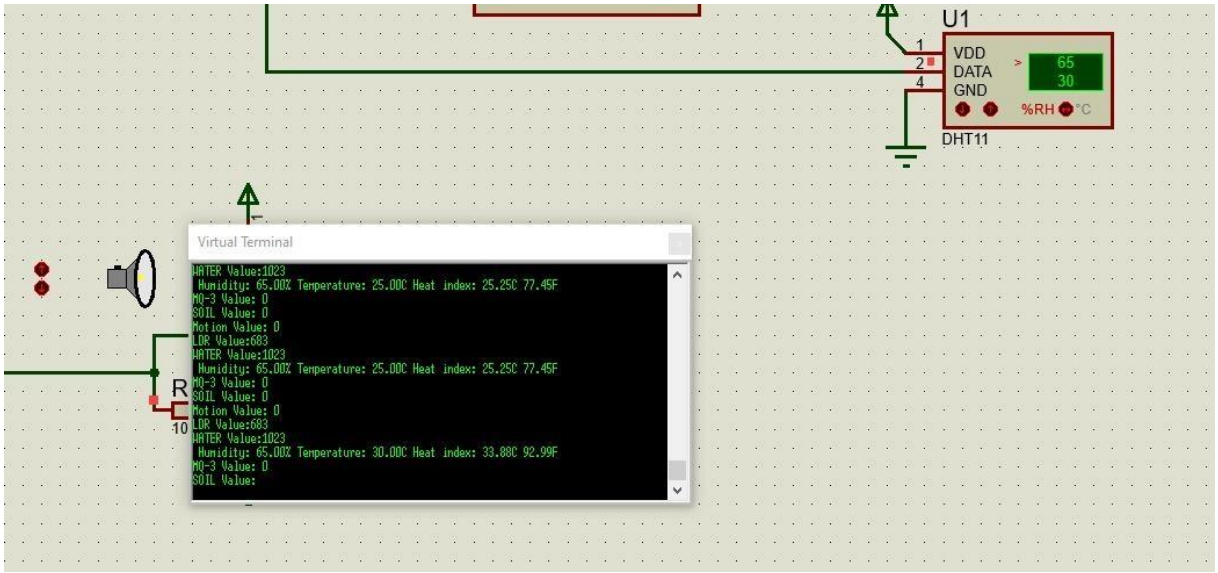
MQ-2 GAS Sensor when taking max value (value=1 state)



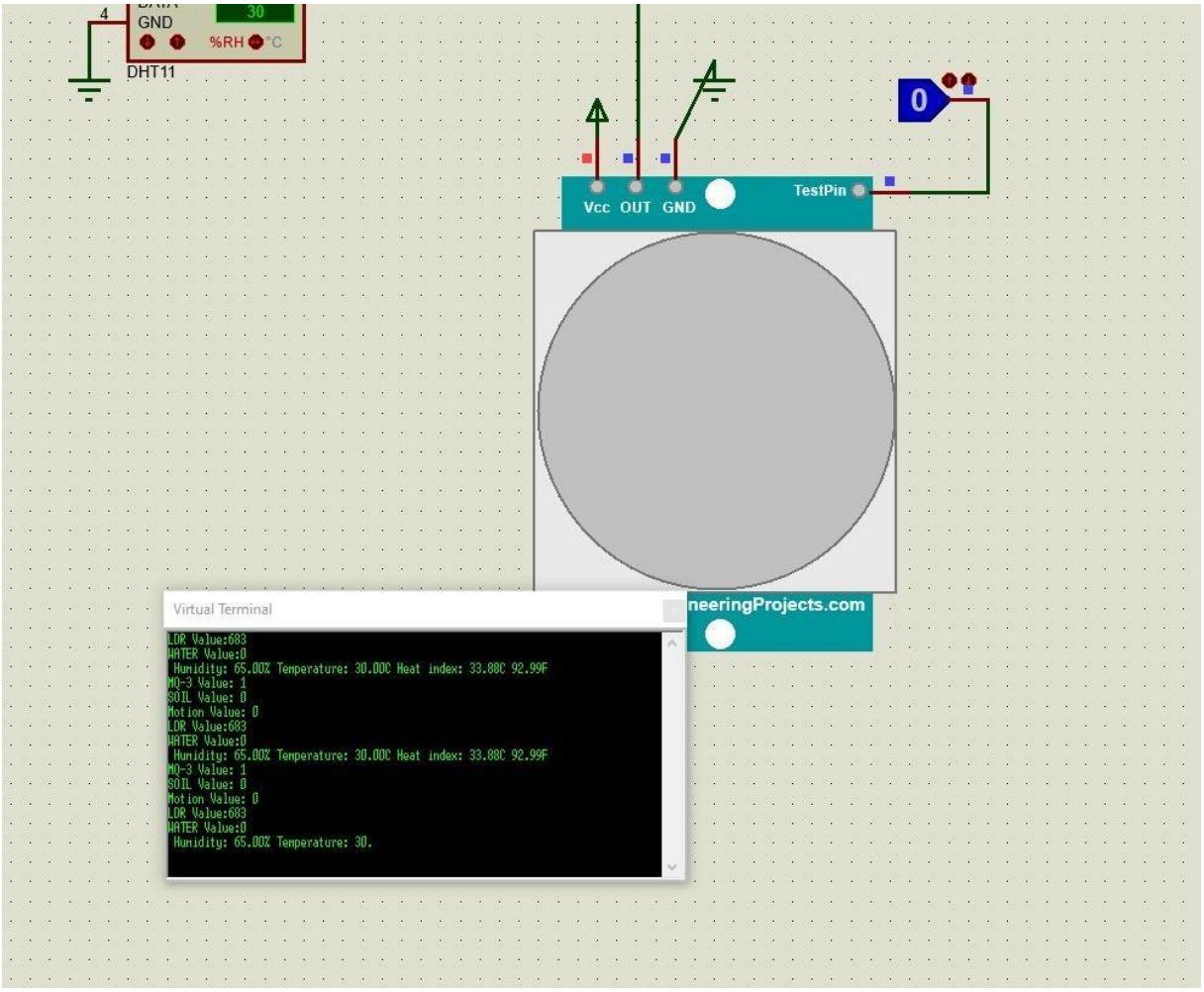
DHT11 Sensor when initialized (value= 76% humidity, 25.0 *C Temperature)



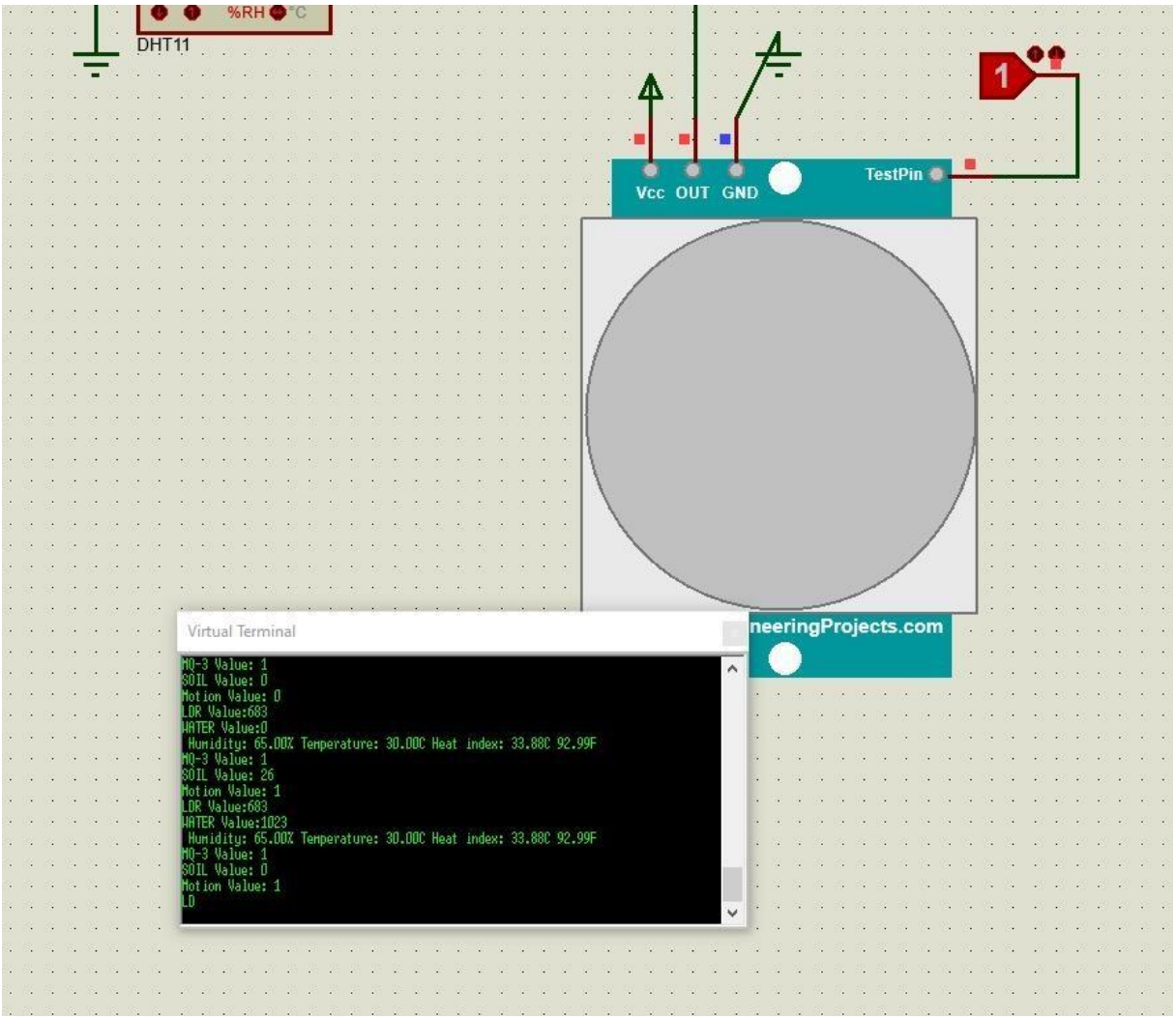
DHT11 Sensor when it’s humidity changed(value= 65% humidity, 25.0 *C Temperature)



DHT11 Sensor when it’s temperature changed(value=65% humidity, 30.0 *C Temperature)



PIR Sensor when initialized (value=0 state)



PIR Sensor when activated(value=1 state)

Part I Sensors

```
// Recever Module

//Include Libraries
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <LiquidCrystal.h>

#define CE_PIN 9
#define CSN_PIN 8
#define BUZZER 10

// We defined for variables coming from the array
#define KEY1 0
#define KEY2 0
#define KEY3 0
#define KEY4 0
#define KEY5 0
#define KEY6 0
#define KEY7 0

// create an RF24 object
RF24 radio(CE_PIN, CSN_PIN);  // CE, CSN

//address through which two modules communicate.
const byte address[6] = "11111";

int datas[] = { KEY1, KEY2, KEY3, KEY4, KEY5, KEY6, KEY7 };

const int rs = 7, en = 6, d4 = 5, d5 = 4, d6 = 3, d7 = 2; LiquidCrystal
lcd(rs, en, d4, d5, d6, d7);

void setup()
{
  while (!Serial);
Serial.begin(9600);
  Serial.println("Nrf24L01 Receiver Initializing");
radio.begin();

  //set the address
  radio.openReadingPipe(0, address);

  //Set module as receiver
```

```
  radio.startListening();

  lcd.begin(16, 2);
  lcd.print("System Starting"); // print to lcd screen   delay(300);   lcd.setCursor(0, 1);
  lcd.print("Lisining to signal...");
}

void loop()
{
  digitalWrite(BUZZER, LOW);
  lcd.cursor();   lcd.blink();
  // if radio signal available   if (radio.available())
  {
    lcd.setCursor(0, 0);
    lcd.print("connection good");     lcd.setCursor(0, 1);
    lcd.print("data coming...");     // reading data from the radio
radio.read(datas, sizeof(datas));

    Serial.print("LDR Value:");
    Serial.println(datas[0]);

    Serial.print("WATER Value:");
    Serial.println(datas[1]);

    Serial.print("Humidity Value:");
    Serial.println(datas[3]);

    Serial.print("Temperature Value:");
    Serial.println(datas[2]);

    Serial.print("MQ-3 Value:");
    Serial.println(datas[4]);

    Serial.print("SOIL Value:");
    Serial.println(datas[5]);

    Serial.print("Motion Value:");
    Serial.println(datas[6]);     // if motion detect     if(datas[6] == 1){
      digitalWrite(BUZZER, HIGH);
```

```
    delay(100);
   }
   else{
    digitalWrite(BUZZER, LOW);
   }
  }
  // if radio signal not available
else {
   Serial.println("waiting for connection...");
lcd.setCursor(0, 1);    lcd.print("waiting
pipe...");    lcd.setCursor(0, 0);
   lcd.print("Lost connection");
  }
}
```

```
// Transmitter Module

//Include Libraries
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include "DHT.h"

#define CE_PIN 9
#define CSN_PIN 8

#define LDR A0 // Analog LDR port
#define WATER A1 // Analog WATER SENSOR port
#define DHTPIN 2 // Digital pin connected to the DHT sensor
#define DHTTYPE DHT11 // DHT 11
#define MQ 3 // MQ-3 port
#define SOIL A2 // SOIL SENSOR port
#define MOTION 4 // Motion Sensor port

DHT dht(DHTPIN, DHTTYPE);

//create an RF24 object
RF24 radio(CE_PIN, CSN_PIN);  // CE, CSN
```

```
//address through which two modules communicate.
const byte address[6] = "111111";

int datas[7];
int ldrValue, waterValue, mq3Value, soilValue, motionValue; float
h, t, f, hif, hic;

void setup()
{
  radio.begin();

  //set the address
radio.openWritingPipe(address);   //
begin port 9600 serial port
  while (!Serial);
Serial.begin(9600);

  dht.begin();   pinMode(MQ,
INPUT);   pinMode(MOTION,
INPUT);   //Set module as
transmitter
radio.stopListening();
}
void loop()
{
  // LDR Start----------
ldrValue = analogRead(A0);
datas[0] = analogRead(LDR);
  Serial.print("LDR Value:");
  Serial.println(ldrValue);
  // LDR End----------

  // WATER Start----------
waterValue = analogRead(A1);
datas[1] = analogRead(WATER);
  Serial.print("WATER Value:");
  Serial.println(waterValue);
  // WATER End----------

  // --------- DHT Start ---------
  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
h = dht.readHumidity();
  // Read temperature as Celsius (the default)
t = dht.readTemperature();
```

```
  // Read temperature as Fahrenheit (isFahrenheit = true)
f = dht.readTemperature(true);
```

```
  // Check if any reads failed and exit early (to try again).
if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
  }


  // Compute heat index in Fahrenheit (the default)
hif = dht.computeHeatIndex(f, h);
  // Compute heat index in Celsius (isFahreheit = false)
hic = dht.computeHeatIndex(t, h, false);   datas[2] = t;
  datas[3] = h;

  Serial.print(F(" Humidity: "));
  Serial.print(h);
  Serial.print(F("% Temperature: "));
  Serial.print(t);
  Serial.print(F("C "));
  Serial.print(F("Heat index: "));
  Serial.print(hic);
  Serial.print(F("C "));
  Serial.print(hif);
  Serial.println(F("F"));
  // --------- DHT End -----------

  // --------- MQ-3 Start -----------
mq3Value = digitalRead(MQ);   datas[4]
= mq3Value;
  Serial.print("MQ-3 Value: ");
  Serial.println(mq3Value);
  // --------- MQ-3 End -----------

  // --------- SOIL Start -----------
soilValue = analogRead(SOIL);
datas[5] = soilValue;
  Serial.print("SOIL Value: ");
  Serial.println(soilValue);
  // --------- SOIL End -----------

  // --------- Motion Start -----------
  motionValue = digitalRead(MOTION);
  datas[6] = motionValue;
Serial.print("Motion Value: ");
  Serial.println(motionValue);
  // --------- Motion End -----------
```

```
// Send data via radio
```

```
  radio.write(datas, sizeof(datas));

  //delay(1);
}
```

Part II Motors

```
// Transmitter Module

//Include Libraries
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <SPI.h>
#include <MFRC522.h>
#define SS_PIN 10
#define RST_PIN 9

#define CE_PIN 8
#define CSN_PIN 7

#define yPosition A1 // yPosition Analog port
#define xPosition A0 // xPosition Analog port
#define swJoistic 2 // Joistic pus button port
#define ledButtons 4 // Led port

MFRC522 mfrc522(SS_PIN, RST_PIN);   // Create MFRC522 instance.
//create an RF24 object
RF24 radio(CE_PIN, CSN_PIN);  // CE, CSN

//address through which two modules communicate.
const byte address[6] = "11100";
int datas[4];

void setup()
{
 Serial.begin(9600);
  radio.begin(); // starting connection

  pinMode(yPosition, INPUT);
pinMode(xPosition, INPUT);
```

```
  pinMode(swJoistic, INPUT_PULLUP);
pinMode(ledButtons, INPUT);

  SPI.begin();      // Initiate  SPI bus
  mfrc522.PCD_Init();   // Initiate MFRC522
  Serial.println("Approximate your card to the reader...");
Serial.println();   //set the address
  radio.openWritingPipe(address);
}

void loop()
{
  // Look for new cards
  if ( ! mfrc522.PICC_IsNewCardPresent())
  {
    return;
  }
  // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial())
  {
    return;
  }
  //Show UID on serial monitor
  Serial.print("UID tag :");
String content= "";
  byte letter;
  for (byte i = 0; i < mfrc522.uid.size; i++)
  {
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
Serial.print(mfrc522.uid.uidByte[i], HEX);
    content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
content.concat(String(mfrc522.uid.uidByte[i], HEX));
  }
  Serial.println();
  Serial.print("Message : ");
content.toUpperCase();
  //to find card uid go to the serial moniter, scan your card and copy uid into the section
if (content.substring(1) == "13 C0 2B AD")
  {
    Serial.println("Authorized access");
    Serial.println();     delay(2000);
while(1){      // write datas
datas[0] = analogRead(xPosition);
datas[1] = analogRead(yPosition);
```

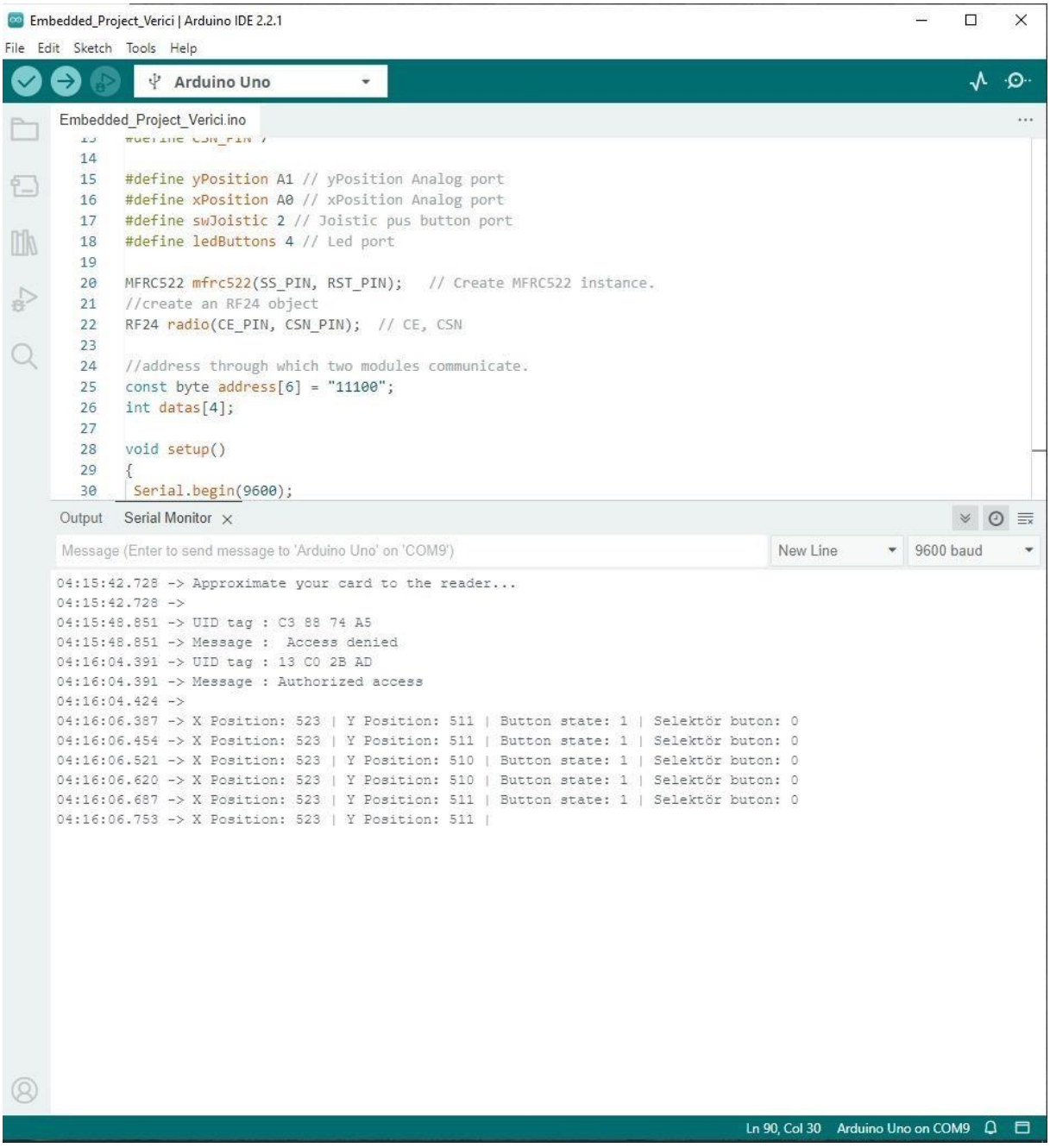
```
    datas[2] = digitalRead(swJoistic);
datas[3] = digitalRead(ledButtons);

    Serial.print("X Position: ");
    Serial.print(datas[0]);
    Serial.print(" | Y Position: ");
    Serial.print(datas[1]);
    Serial.print(" | Button state: ");
    Serial.print(datas[2]);
    Serial.print(" | Light buton: ");
    Serial.print(datas[3]);
    Serial.println("  ");

    radio.write( datas, sizeof(datas) ); //We print the data to the output to be sent
   }
  } else
{
  Serial.println(" Access denied");
delay(2000);
  }
delay(500);
}
```

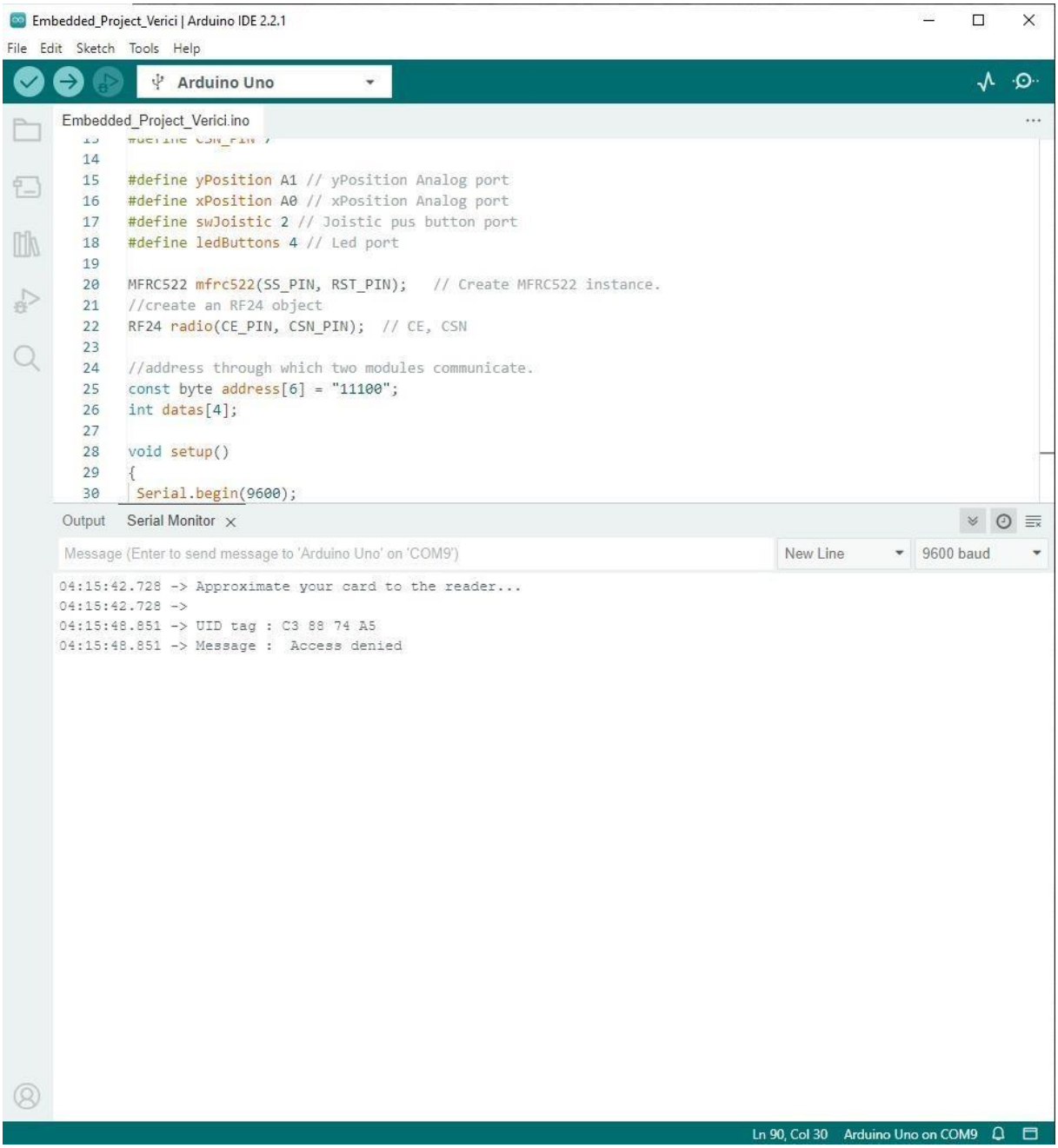
```
// Recever Module

//Include Libraries
#include <Servo.h>
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

#define CE_PIN 9
#define CSN_PIN 8

#define SERVO1 6 // Serveo 1 port
#define SERVO2 5 // Serveo 2 port
#define DCMOTOR 3 // DC Motor port
#define LEDS 2 // Leds port

// We defined for variables coming from the array
```

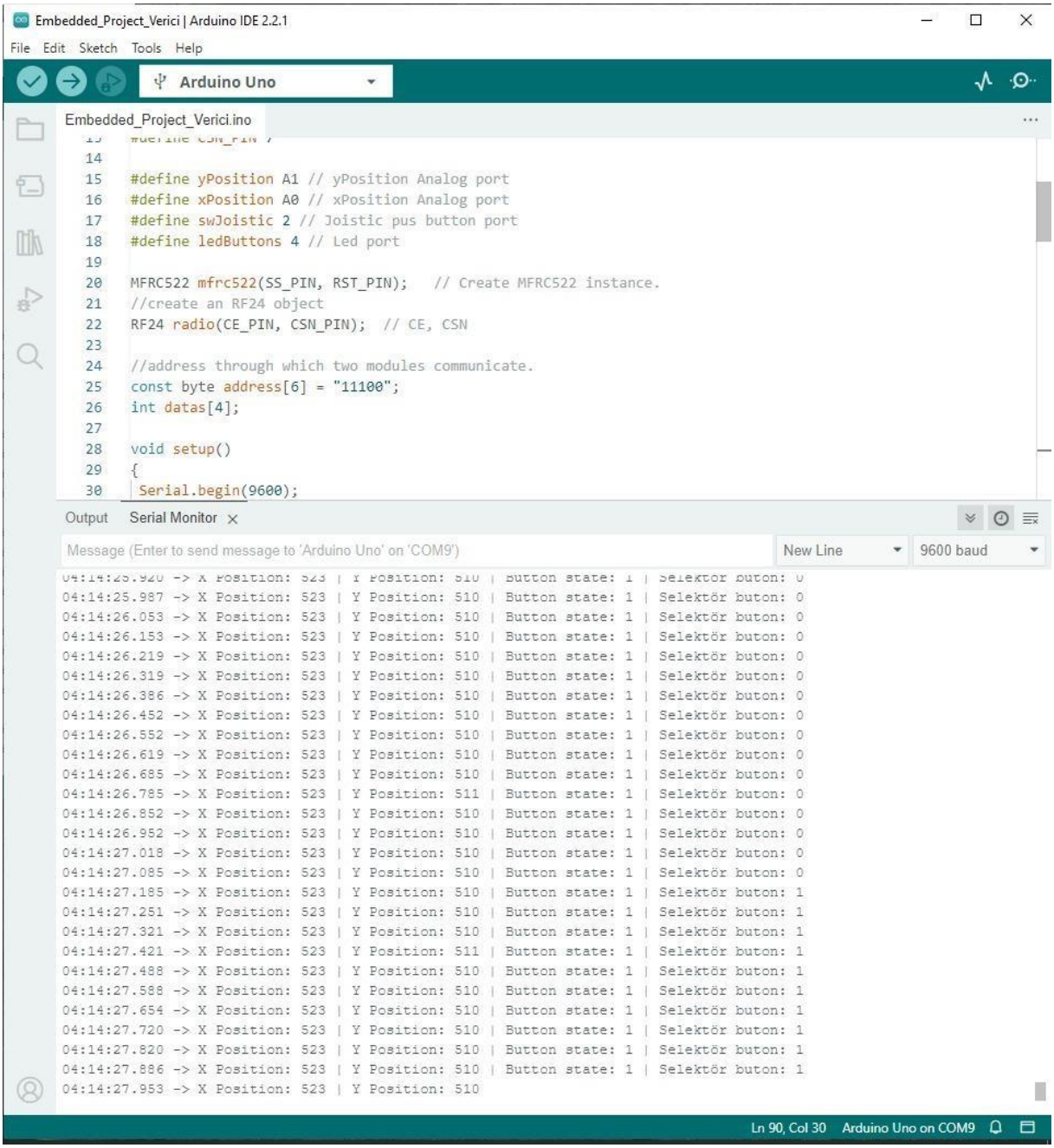
```
#define KEY1 0
#define KEY2 0
#define KEY3 0
#define KEY4 0

//create an RF24 object
RF24 radio(CE_PIN, CSN_PIN);  // CE, CSN

//address through which two modules communicate.
const byte address[6] = "11100";

int datas[] = { KEY1, KEY2, KEY3, KEY4 };
int servoValue1, servoValue2, swJoistic, ledStatus; bool
open = false; // this defined for car led

// Servo motor objects
Servo servoObje1;
Servo servoObje2;

void setup()
{
  // begin serial port 9600
  while (!Serial);
Serial.begin(9600);   delay(500);

  servoObje1.attach(SERVO2);
servoObje2.attach(SERVO1);
pinMode(DCMOTOR, OUTPUT);
pinMode(LEDS, OUTPUT);   //We
communicate via radio
  Serial.println("Nrf24L01 Receiver Initializing...");
radio.begin();    //set the address
  radio.openReadingPipe(0, address);
  //Set module as receiver
  radio.startListening();
}

void loop()
```

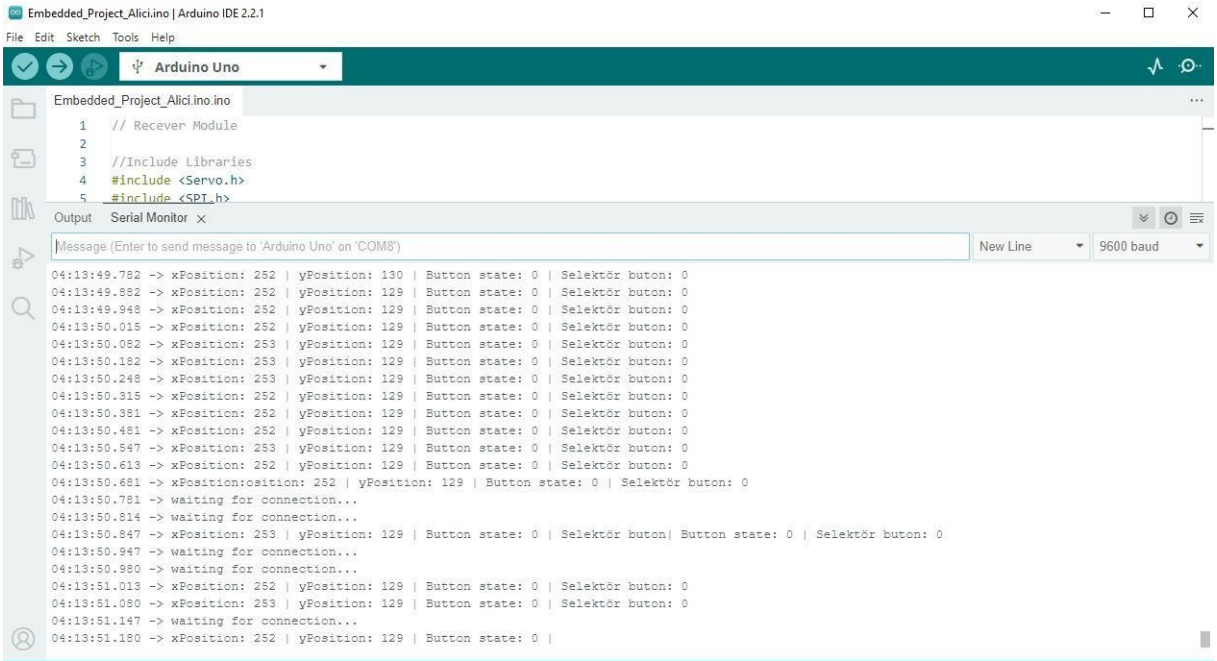
```
{
  if (radio.available()) // if connection available
  {
    // reading data from the radio
radio.read(datas, sizeof(datas));

    //We adapted the incoming information to a certain range
servoValue1 = map(datas[0], 522, 1023, 0, 254);
servoValue2 = map(datas[1], 0, 1023, 0, 179);     swJoistic =
datas[2];
    ledStatus = datas[3];

    Serial.print("xPosition: ");
    Serial.print(servoValue1);
    Serial.print(" | yPosition: ");
    Serial.print(servoValue2);
    Serial.print(" | Button state: ");
    Serial.print(swJoistic);
    // if joistic buton pressed servo moves
if(datas[2] == 0){       swJoistic = 180;
    }
    Serial.print(" | Light buton: ");
    Serial.print(ledStatus);
    Serial.println();

    //dc motor moves
if(servoValue1 >=0 ){
      analogWrite(DCMOTOR, servoValue1);
    }
    // If the button is pressed, the car light turns on.
    // open will be true or false
if(ledStatus == 1){       open =
!open;       delay(400);
    }
    if(open){
      digitalWrite(LEDS, HIGH);
    }
```

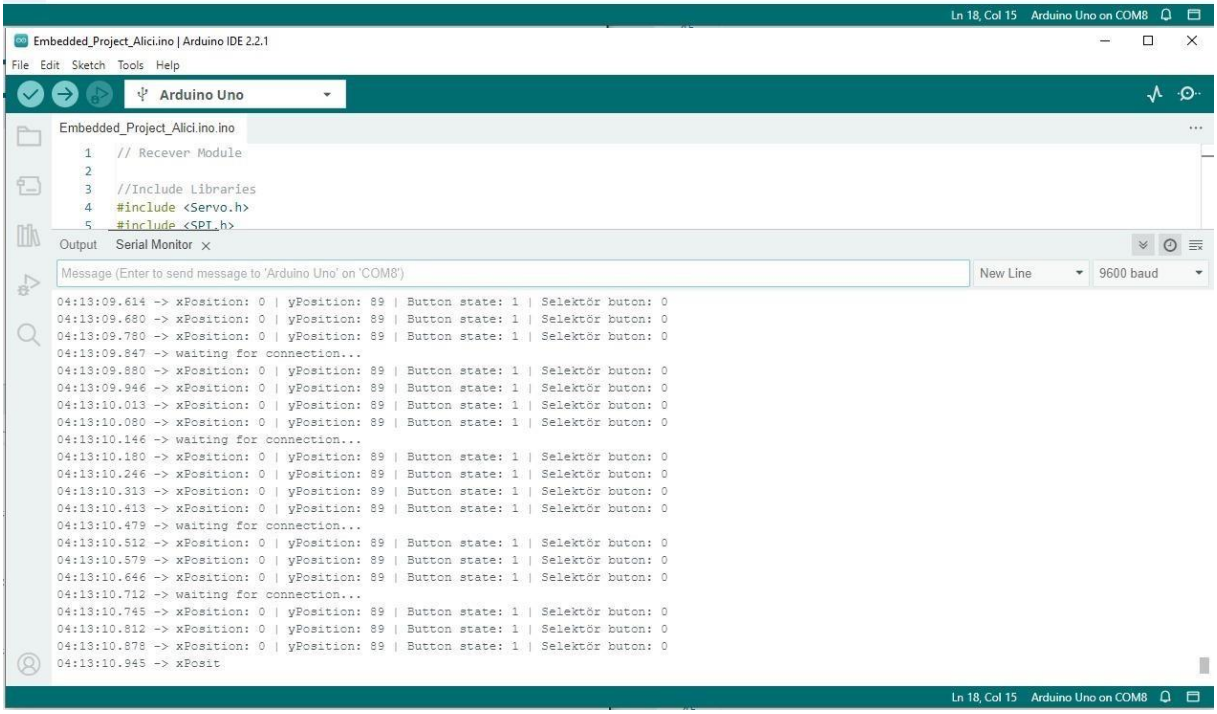
```
   else{
     digitalWrite(LEDS, LOW);
   }

   // adjusting the degree of the servo motor
servoObje1.write(swJoistic);
   servoObje2.write(servoValue2);
  }
  else { // if connection not available
Serial.println("waiting for connection...");
  }
}
```

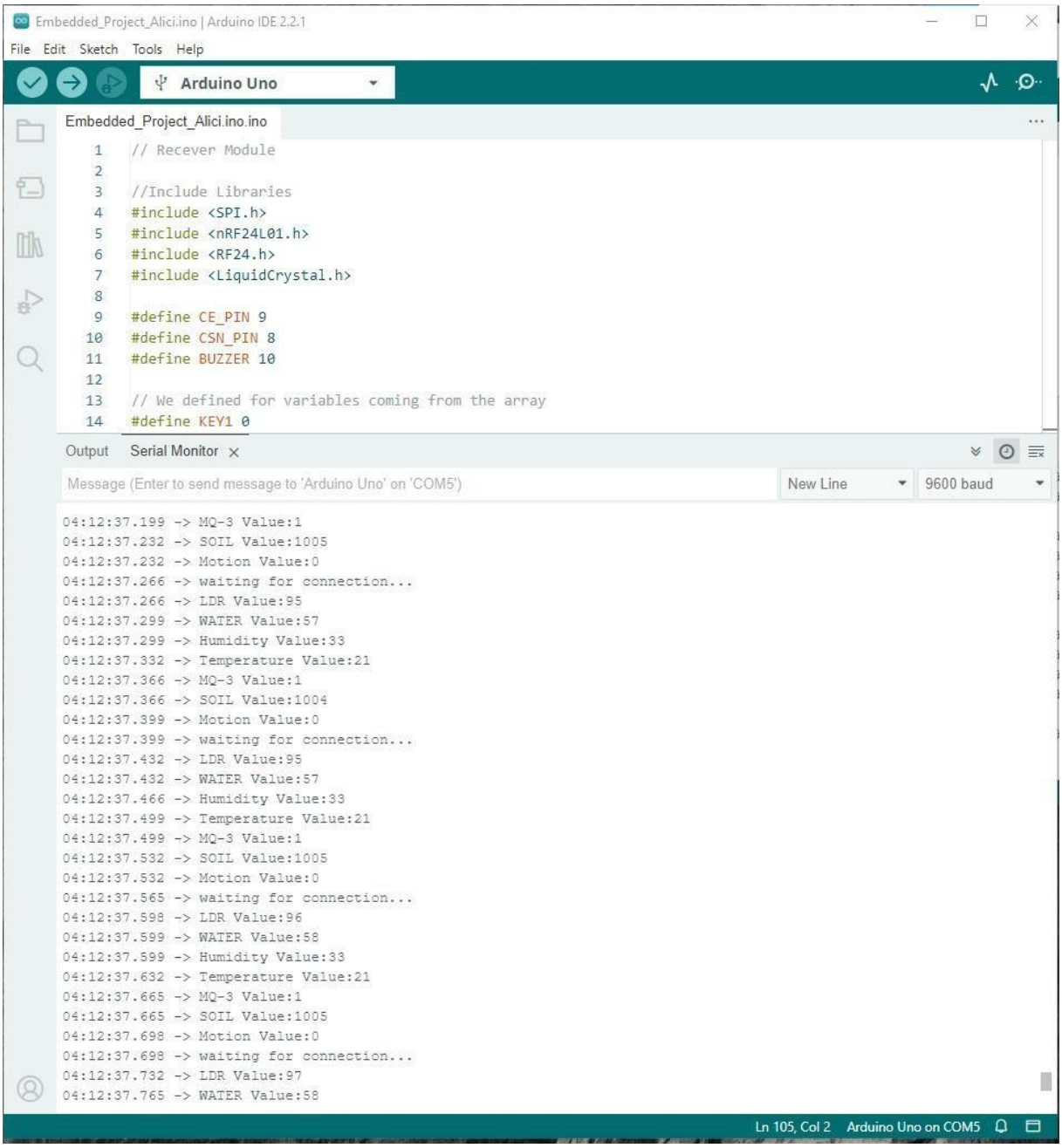

Embedded_Project_Verici | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Arduino Uno
Embedded_Project_Verici.ino
#define yPosition A1 // yPosition Analog port
#define xPosition A0 // xPosition Analog port
#define swJoistic 2 // Joistic pus button port
#define ledButtons 4 // Led port
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance.
//create an RF24 object
RF24 radio(CE_PIN, CSN_PIN); // CE, CSN
//address through which two modules communicate.
const byte address[6] = "11100";
int datas[4];
void setup()
{
Serial.begin(9600);
Output Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'COM9')
New Line
9600 baud
04:15:42.728 -> Approximate your card to the reader...
04:15:42.728 ->
04:15:48.851 -> UID tag : C3 88 74 A5
04:15:48.851 -> Message : Access denied
04:16:04.391 -> UID tag : 13 C0 2B AD
04:16:04.391 -> Message : Authorized access
04:16:04.424 ->
04:16:06.387 -> X Position: 523 | Y Position: 511 | Button state: 1 | Selektör buton: 0
04:16:06.454 -> X Position: 523 | Y Position: 511 | Button state: 1 | Selektör buton: 0
04:16:06.521 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:16:06.620 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:16:06.687 -> X Position: 523 | Y Position: 511 | Button state: 1 | Selektör buton: 0
04:16:06.753 -> X Position: 523 | Y Position: 511 |
Ln 90, Col 30 Arduino Uno on COM9

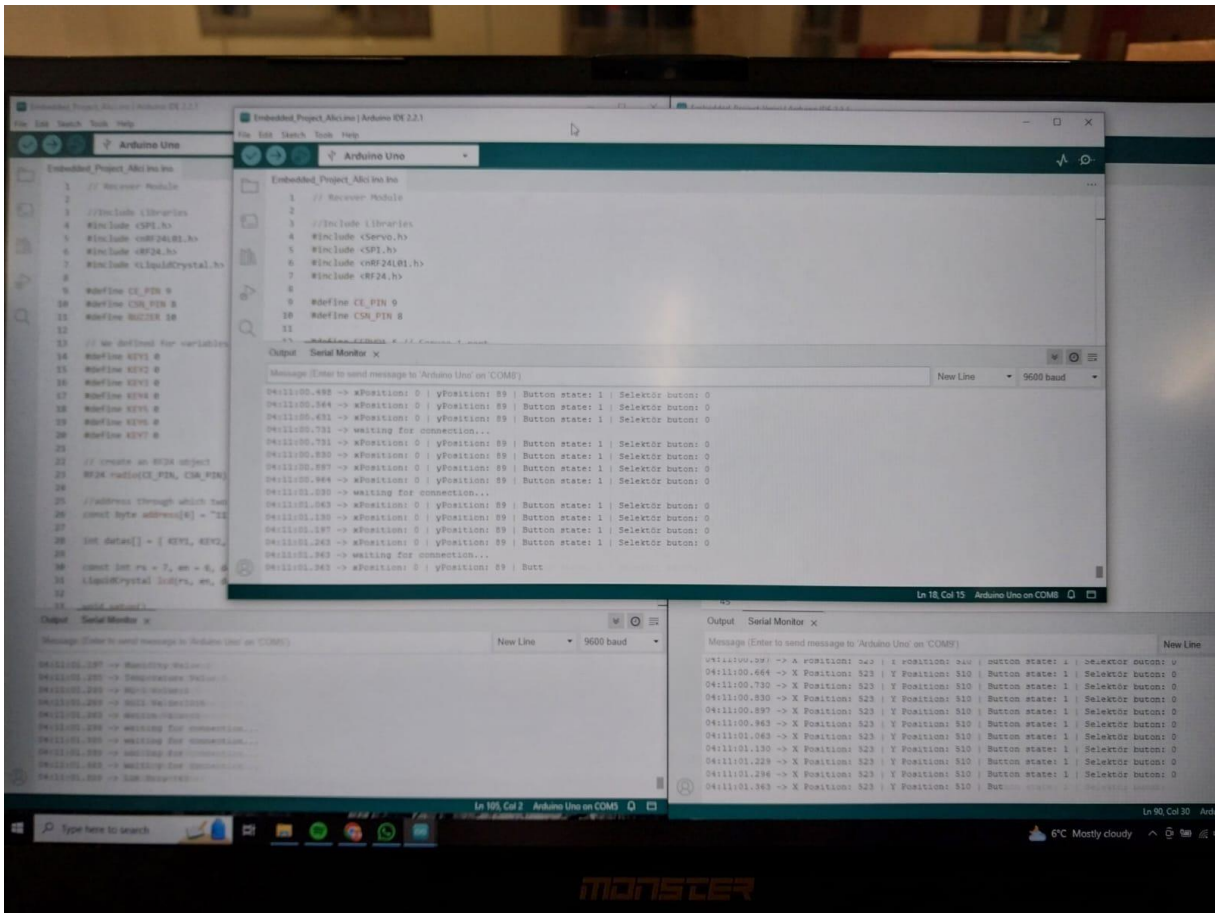

Embedded_Project_Verici | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Arduino Uno
Embedded_Project_Verici.ino
#define yPosition A1 // yPosition Analog port
#define xPosition A0 // xPosition Analog port
#define swJoistic 2 // Joistic pus button port
#define ledButtons 4 // Led port
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance.
//create an RF24 object
RF24 radio(CE_PIN, CSN_PIN); // CE, CSN
//address through which two modules communicate.
const byte address[6] = "11100";
int datas[4];
void setup()
{
Serial.begin(9600);
Output
Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'COM9')
New Line
9600 baud
04:15:42.728 -> Approximate your card to the reader...
04:15:42.728 ->
04:15:48.851 -> UID tag : C3 88 74 A5
04:15:48.851 -> Message : Access denied
Ln 90, Col 30
Arduino Uno on COM9

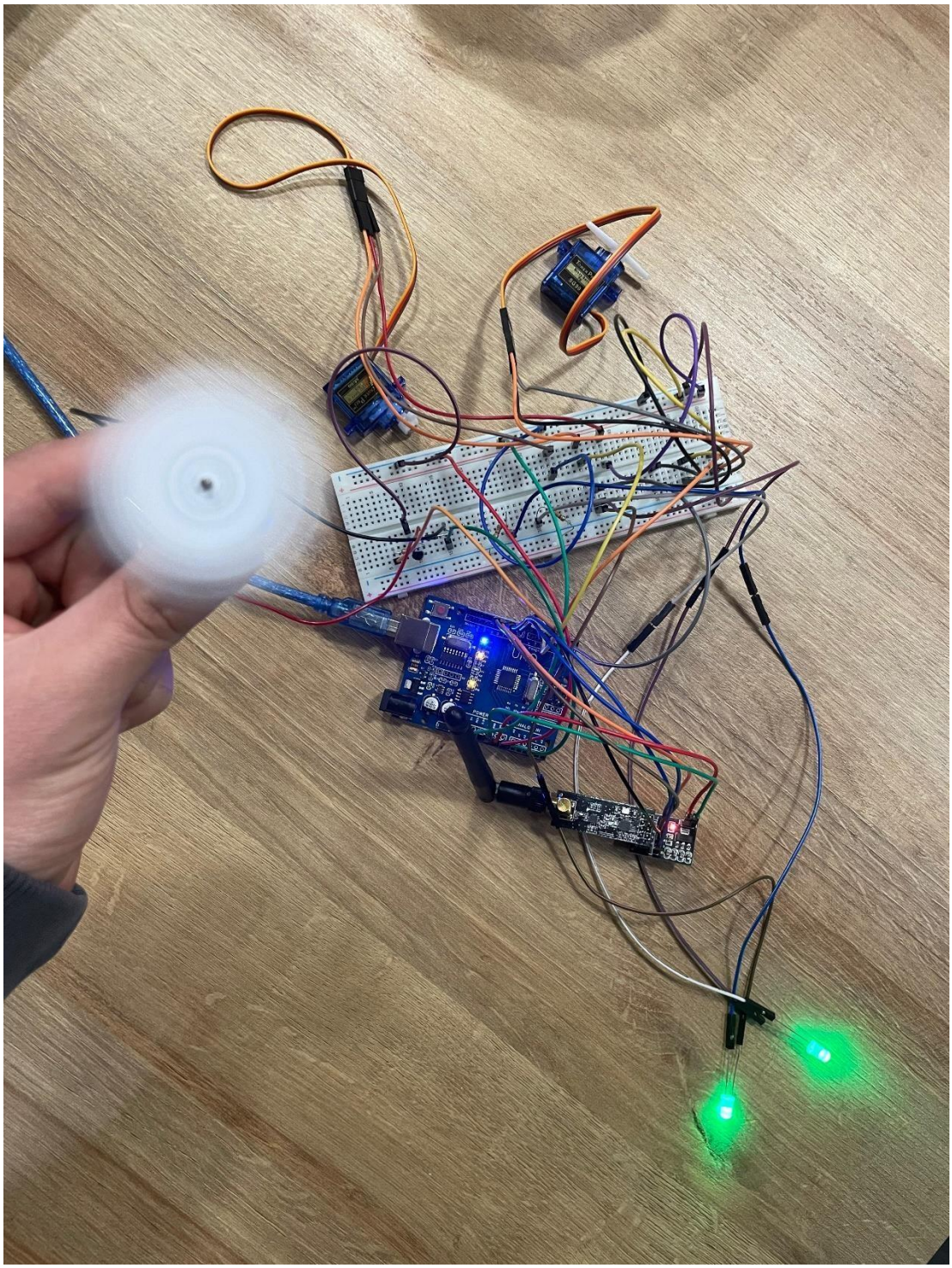

Embedded_Project_Verici | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Arduino Uno
Embedded_Project_Verici.ino
#define yPosition A1 // yPosition Analog port
#define xPosition A0 // xPosition Analog port
#define swJoistic 2 // Joistic pus button port
#define ledButtons 4 // Led port
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance.
//create an RF24 object
RF24 radio(CE_PIN, CSN_PIN); // CE, CSN
//address through which two modules communicate.
const byte address[6] = "11100";
int datas[4];
void setup()
{
Serial.begin(9600);
Output Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'COM9')
New Line
9600 baud
04:14:25.920 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:14:25.987 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:14:26.053 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:14:26.153 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:14:26.219 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:14:26.319 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:14:26.386 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:14:26.452 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:14:26.552 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:14:26.619 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:14:26.685 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:14:26.785 -> X Position: 523 | Y Position: 511 | Button state: 1 | Selektör buton: 0
04:14:26.852 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:14:26.952 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:14:27.018 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:14:27.085 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:14:27.185 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 1
04:14:27.251 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 1
04:14:27.321 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 1
04:14:27.421 -> X Position: 523 | Y Position: 511 | Button state: 1 | Selektör buton: 1
04:14:27.488 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 1
04:14:27.588 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 1
04:14:27.654 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 1
04:14:27.720 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 1
04:14:27.820 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 1
04:14:27.886 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 1
04:14:27.953 -> X Position: 523 | Y Position: 510
Ln 90, Col 30 Arduino Uno on COM9


Embedded_Project_Alici.ino | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Arduino Uno
Embedded_Project_Alici.ino.ino
1 // Recever Module
2
3 //Include Libraries
4 #include <Servo.h>
5 #include <SPI.h>
Output Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'COM8')
New Line
9600 baud
04:13:49.782 -> xPosition: 252 | yPosition: 130 | Button state: 0 | Selektör buton: 0
04:13:49.882 -> xPosition: 252 | yPosition: 129 | Button state: 0 | Selektör buton: 0
04:13:49.948 -> xPosition: 252 | yPosition: 129 | Button state: 0 | Selektör buton: 0
04:13:50.015 -> xPosition: 252 | yPosition: 129 | Button state: 0 | Selektör buton: 0
04:13:50.082 -> xPosition: 253 | yPosition: 129 | Button state: 0 | Selektör buton: 0
04:13:50.182 -> xPosition: 253 | yPosition: 129 | Button state: 0 | Selektör buton: 0
04:13:50.248 -> xPosition: 253 | yPosition: 129 | Button state: 0 | Selektör buton: 0
04:13:50.315 -> xPosition: 252 | yPosition: 129 | Button state: 0 | Selektör buton: 0
04:13:50.381 -> xPosition: 252 | yPosition: 129 | Button state: 0 | Selektör buton: 0
04:13:50.481 -> xPosition: 252 | yPosition: 129 | Button state: 0 | Selektör buton: 0
04:13:50.547 -> xPosition: 253 | yPosition: 129 | Button state: 0 | Selektör buton: 0
04:13:50.613 -> xPosition: 252 | yPosition: 129 | Button state: 0 | Selektör buton: 0
04:13:50.681 -> xPosition:osition: 252 | yPosition: 129 | Button state: 0 | Selektör buton: 0
04:13:50.781 -> waiting for connection...
04:13:50.814 -> waiting for connection...
04:13:50.847 -> xPosition: 253 | yPosition: 129 | Button state: 0 | Selektör buton| Button state: 0 | Selektör buton: 0
04:13:50.947 -> waiting for connection...
04:13:50.980 -> waiting for connection...
04:13:51.013 -> xPosition: 252 | yPosition: 129 | Button state: 0 | Selektör buton: 0
04:13:51.080 -> xPosition: 253 | yPosition: 129 | Button state: 0 | Selektör buton: 0
04:13:51.147 -> waiting for connection...
04:13:51.180 -> xPosition: 252 | yPosition: 129 | Button state: 0 |
Ln 18, Col 15 Arduino Uno on COM8

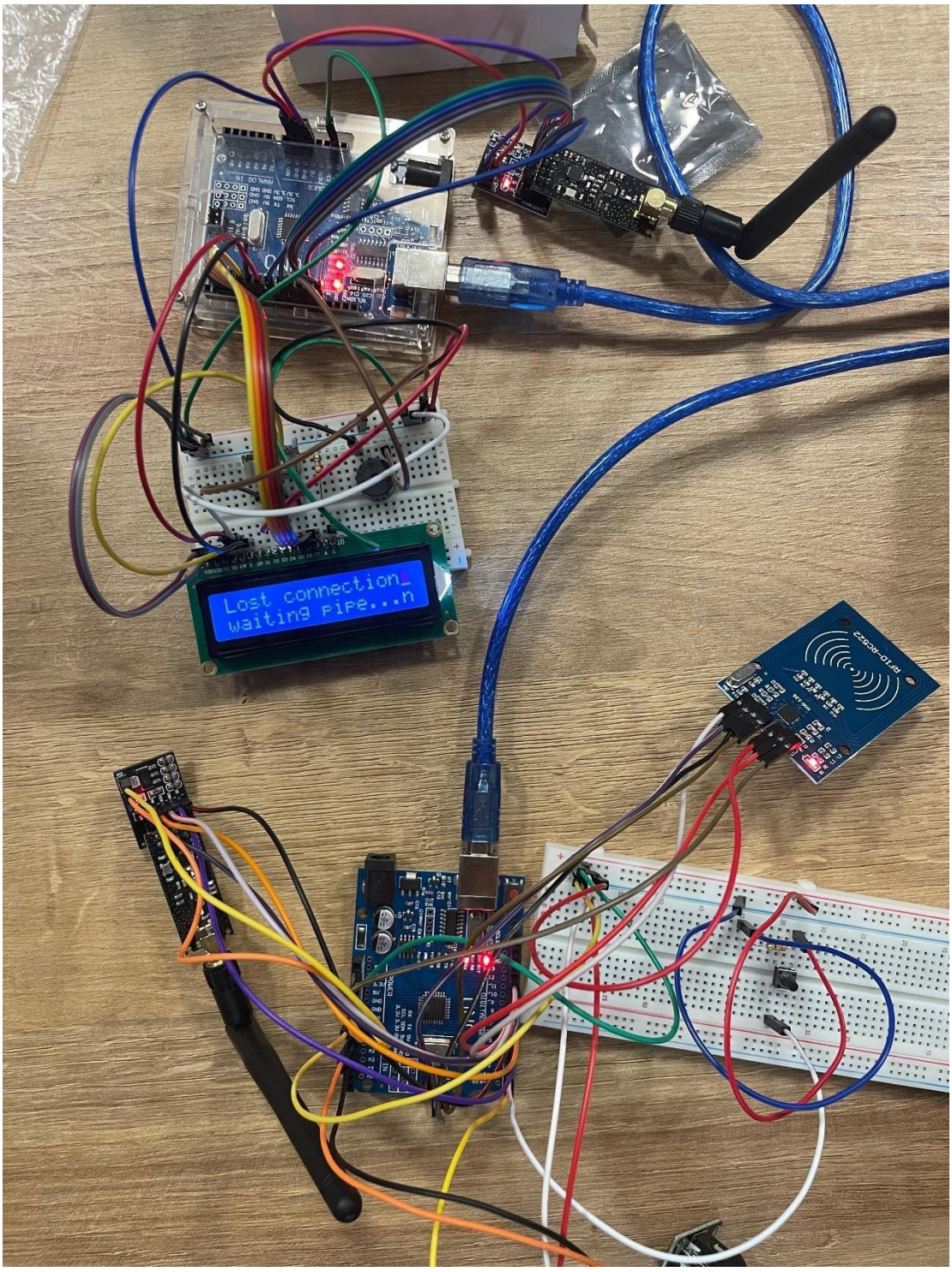

Embedded_Project_Alici.ino | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Arduino Uno
Embedded_Project_Alici.ino.ino
1 // Recever Module
2
3 //Include Libraries
4 #include <Servo.h>
5 #include <SPI.h>
Output Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'COM8')
New Line
9600 baud
04:13:09.614 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:13:09.680 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:13:09.780 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:13:09.847 -> waiting for connection...
04:13:09.880 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:13:09.946 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:13:10.013 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:13:10.080 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:13:10.146 -> waiting for connection...
04:13:10.180 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:13:10.246 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:13:10.313 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:13:10.413 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:13:10.479 -> waiting for connection...
04:13:10.512 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:13:10.579 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:13:10.646 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:13:10.712 -> waiting for connection...
04:13:10.745 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:13:10.812 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:13:10.878 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:13:10.945 -> xPosit
Ln 18, Col 15 Arduino Uno on COM8


Embedded_Project_Alici.ino | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Arduino Uno
Embedded_Project_Alici.ino.ino
1 // Recever Module
2
3 //Include Libraries
4 #include <SPI.h>
5 #include <nRF24L01.h>
6 #include <RF24.h>
7 #include <LiquidCrystal.h>
8
9 #define CE_PIN 9
10 #define CSN_PIN 8
11 #define BUZZER 10
12
13 // We defined for variables coming from the array
14 #define KEY1 0
Output Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'COM5')
New Line
9600 baud
04:12:37.199 -> MQ-3 Value:1
04:12:37.232 -> SOIL Value:1005
04:12:37.232 -> Motion Value:0
04:12:37.266 -> waiting for connection...
04:12:37.266 -> LDR Value:95
04:12:37.299 -> WATER Value:57
04:12:37.299 -> Humidity Value:33
04:12:37.332 -> Temperature Value:21
04:12:37.366 -> MQ-3 Value:1
04:12:37.366 -> SOIL Value:1004
04:12:37.399 -> Motion Value:0
04:12:37.399 -> waiting for connection...
04:12:37.432 -> LDR Value:95
04:12:37.432 -> WATER Value:57
04:12:37.466 -> Humidity Value:33
04:12:37.499 -> Temperature Value:21
04:12:37.499 -> MQ-3 Value:1
04:12:37.532 -> SOIL Value:1005
04:12:37.532 -> Motion Value:0
04:12:37.565 -> waiting for connection...
04:12:37.598 -> LDR Value:96
04:12:37.599 -> WATER Value:58
04:12:37.599 -> Humidity Value:33
04:12:37.632 -> Temperature Value:21
04:12:37.665 -> MQ-3 Value:1
04:12:37.665 -> SOIL Value:1005
04:12:37.698 -> Motion Value:0
04:12:37.698 -> waiting for connection...
04:12:37.732 -> LDR Value:97
04:12:37.765 -> WATER Value:58
Ln 105, Col 2 Arduino Uno on COM5

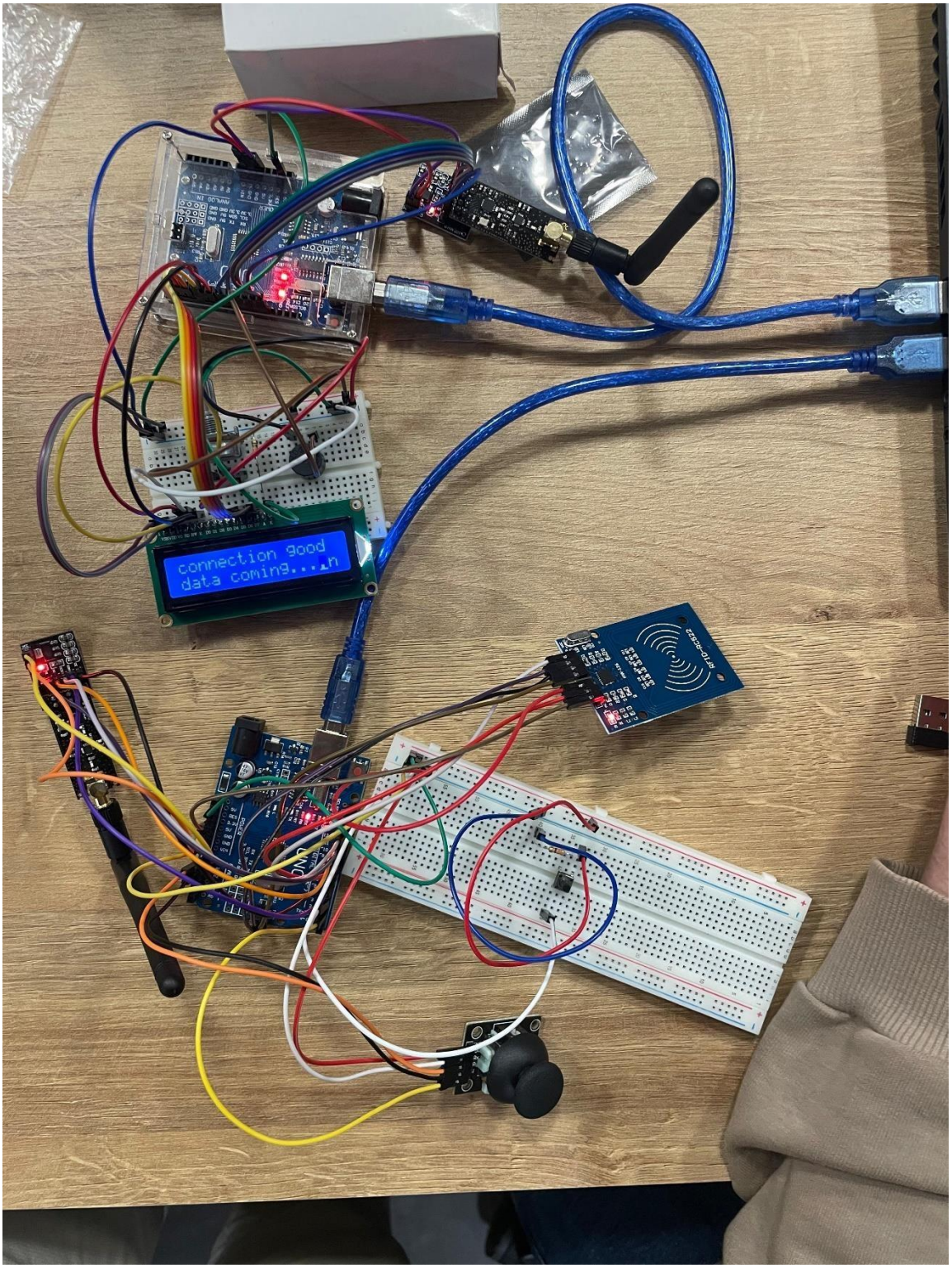

Embedded_Project_Alici.ino | Arduino IDE 2.2.1
Arduino Uno
Embedded_Project_Alici.ino.ino
// Receiver Module
//Include Libraries
#include <Servo.h>
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#define CE_PIN 9
#define CSN_PIN 8
Output Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'COM8')
New Line
9600 baud
04:11:00.498 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:11:00.564 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:11:00.631 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:11:00.731 -> waiting for connection...
04:11:00.731 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:11:00.830 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:11:00.897 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:11:00.964 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:11:01.030 -> waiting for connection...
04:11:01.063 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:11:01.130 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:11:01.197 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:11:01.263 -> xPosition: 0 | yPosition: 89 | Button state: 1 | Selektör buton: 0
04:11:01.363 -> waiting for connection...
Ln 18, Col 15 Arduino Uno on COM8
Output Serial Monitor
New Line
9600 baud
Ln 105, Col 2 Arduino Uno on COM5
Output Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'COM9')
New Line
04:11:00.664 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:11:00.730 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:11:00.830 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:11:00.897 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:11:00.963 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:11:01.063 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:11:01.130 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:11:01.229 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:11:01.296 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
04:11:01.363 -> X Position: 523 | Y Position: 510 | Button state: 1 | Selektör buton: 0
Ln 90, Col 30
Type here to search
6°C Mostly cloudy
MONSTER



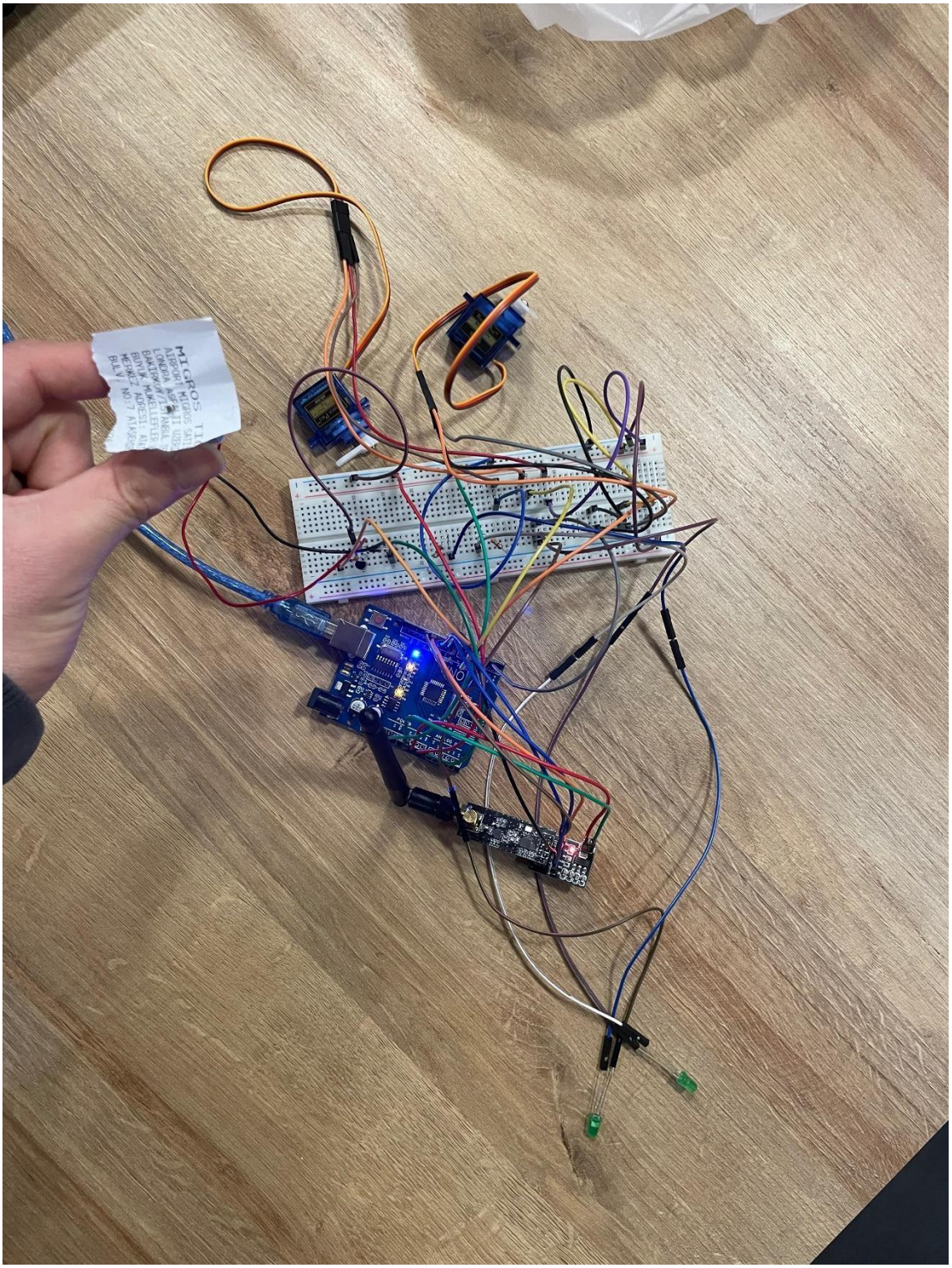
This is motor system



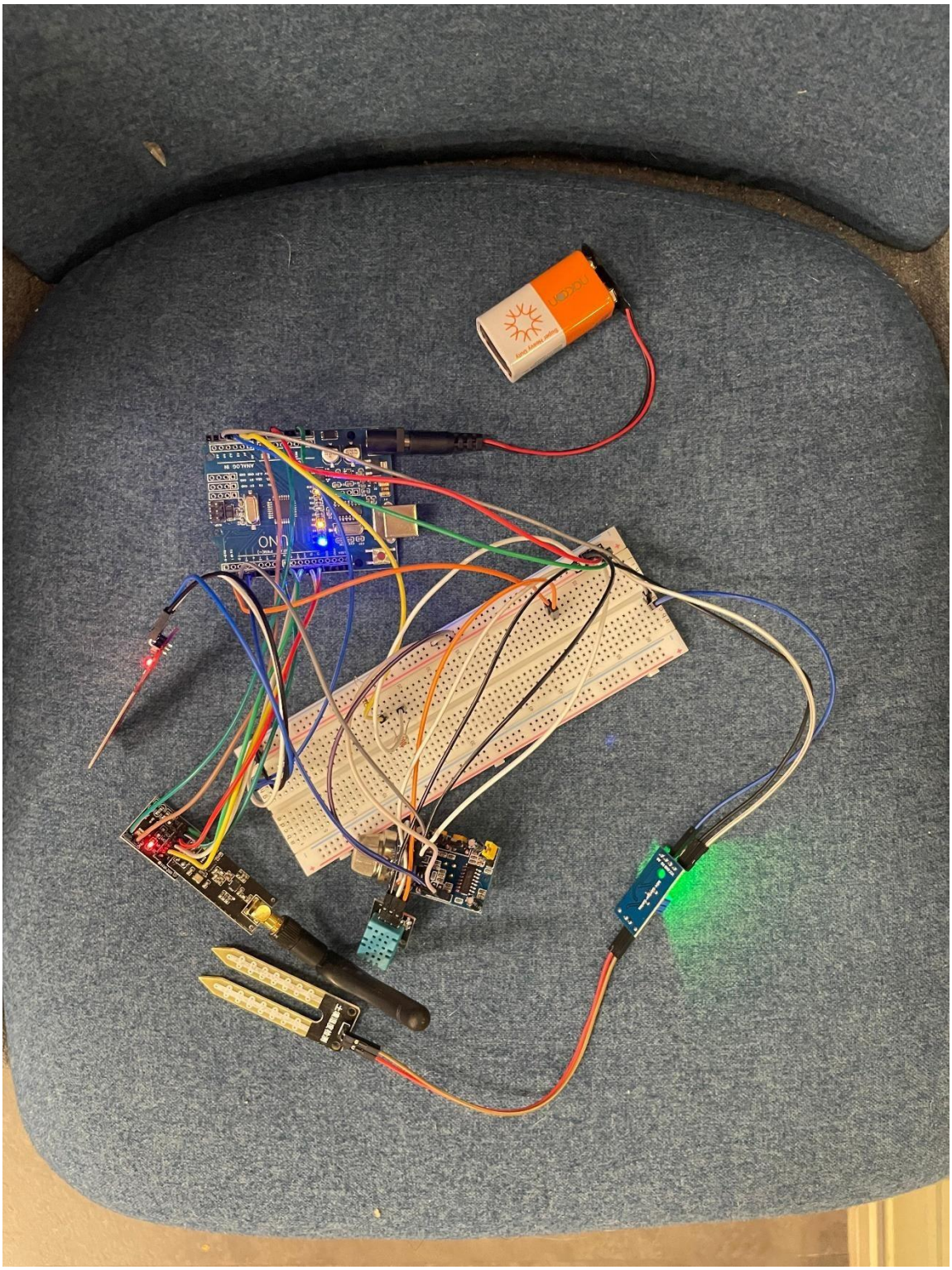
The top one is sensor controller, the bottom is motor controller system (connection lost)



The top one is sensor controller, the bottom is motor controller system (connection good)



Motor system



Sensor system