|  |  |  |
| --- | --- | --- |
| Project No&Topic: HAND CONTROLLED RC CAR |  | Name-Surname: ÖMER FARUK MALI  KUBİLAY ŞENGÜL  EGE HÜDÜREFE |
| Group No : 7 |  | ID# : 160412033  190412066  190412079 |

**1. Introduction**

This week our goal was to improve the maze solver robot and configure it to hand gesture control robot with wireless gloves. In that regard, we used 2 different modules that we have not used before. One of them was acceleration and gyro sensor and other one was wireless module. No changes have been made to the chassis.

**2. Equipment**

Arduino UNO

Arduino NANO

2 DC motor

3 ultrasonic displacement sensors

2 9V battery

MPU6050 gyro sensor

2 NRF24L01 wireless communication component

Miniboard

Car chassis

2 wheels

Jumper cables

Computer (for programming Arduino)

Glove (for snatching mini board)

**3. Software**

**RECEIVER CODE:**

#include <SPI.h>

#include <RF24.h>

RF24 radio(7, 8); // CE, CSN

int data[2];// X ve Y düzlemi için dizi tanımlama

#define sol\_motor\_hiz 3 // Sol motor hız pini

#define sag\_motor\_hiz 10 // Sağ motor hız pini

#define sol\_motor\_1 4 // Sol motor ileri

#define sol\_motor\_2 9 // Sol motor geri

#define sag\_motor\_1 5 //Sağ motor ileri

#define sag\_motor\_2 6 // Sağ motor geri

int sag\_sensor\_echo = A0;// sağ sensör

int sag\_sensor\_trigger = A1;// sağ sensör

int sol\_sensor\_echo = A2;// sol sensör

int sol\_sensor\_trigger = A3;// sol sensör

int on\_sensor\_echo = A4; // Ön sensör

int on\_sensor\_trigger = A5; // Ön sensör

int PWM=255;

void setup()

{

radio.begin();

radio.openReadingPipe(1,1234);

radio.startListening();

pinMode(on\_sensor\_trigger, OUTPUT);

pinMode(on\_sensor\_echo, INPUT);

pinMode(sol\_sensor\_trigger, OUTPUT);

pinMode(sol\_sensor\_echo, INPUT);

pinMode(sag\_sensor\_trigger, OUTPUT);

pinMode(sag\_sensor\_echo, INPUT);

pinMode(sol\_motor\_hiz, OUTPUT);

pinMode(sag\_motor\_hiz, OUTPUT);

pinMode(sol\_motor\_1, OUTPUT);

pinMode(sol\_motor\_2, OUTPUT);

pinMode(sag\_motor\_1, OUTPUT);

pinMode(sag\_motor\_2, OUTPUT);

Serial.begin(9600);

delay(5000);

}

void loop()

{

long on\_sensor\_zaman, sol\_sensor\_zaman, sag\_sensor\_zaman, sag\_mesafe, sol\_mesafe, on\_mesafe;

digitalWrite(on\_sensor\_trigger, LOW);

delayMicroseconds(2);

digitalWrite(on\_sensor\_trigger, HIGH);

delayMicroseconds(5);

digitalWrite(on\_sensor\_trigger, LOW);

on\_sensor\_zaman = pulseIn(on\_sensor\_echo, HIGH);

on\_mesafe = on\_sensor\_zaman/29/2;

digitalWrite(sol\_sensor\_trigger, LOW);

delayMicroseconds(2);

digitalWrite(sol\_sensor\_trigger, HIGH);

delayMicroseconds(5);

digitalWrite(sol\_sensor\_trigger, LOW);

sol\_sensor\_zaman = pulseIn(sol\_sensor\_echo, HIGH);

sol\_mesafe = sol\_sensor\_zaman/29/2;

digitalWrite(sag\_sensor\_trigger, LOW);

delayMicroseconds(2);

digitalWrite(sag\_sensor\_trigger, HIGH);

delayMicroseconds(5);

digitalWrite(sag\_sensor\_trigger, LOW);

sag\_sensor\_zaman = pulseIn(sag\_sensor\_echo, HIGH);

sag\_mesafe = sag\_sensor\_zaman/29/2;

analogWrite(sol\_motor\_hiz, 0);

analogWrite(sag\_motor\_hiz, 0);

analogWrite(sol\_motor\_1, 0);

analogWrite(sol\_motor\_2, 0);

analogWrite(sag\_motor\_1, 0);

analogWrite(sag\_motor\_2, 0);

/\* Serial.print("ön=");Serial.print(on\_mesafe);

Serial.print(" sağ=");Serial.print(sag\_mesafe);

Serial.print(" sol=");Serial.print(sol\_mesafe);

Serial.println(""); \*/

Serial.print("x axis: ");Serial.print(data[0]);Serial.print("y axis: ");Serial.println(data[1]);

if (radio.available()) {

radio.read(data, sizeof(data));

Serial.print("x axis: ");Serial.print(data[0]);Serial.print("y axis: ");Serial.println(data[1]);

if(data[0]> 50 && on\_mesafe > 10)//ileri

{

analogWrite(sol\_motor\_hiz, PWM);

analogWrite(sag\_motor\_hiz, PWM);

analogWrite(sol\_motor\_1, data[0]);

analogWrite(sol\_motor\_2, 0);

analogWrite(sag\_motor\_1, data[0]);

analogWrite(sag\_motor\_2, 0);

}

if(data[0]< -50)//geri

{

analogWrite(sol\_motor\_hiz, PWM);

analogWrite(sag\_motor\_hiz, PWM);

analogWrite(sol\_motor\_1, 0);

analogWrite(sol\_motor\_2, data[0]);

analogWrite(sag\_motor\_1, 0);

analogWrite(sag\_motor\_2, data[0]);

}

if(data[1] > 50)//sol

{

analogWrite(sol\_motor\_hiz, PWM);

analogWrite(sag\_motor\_hiz, PWM);

analogWrite(sol\_motor\_1, 0);

analogWrite(sol\_motor\_2, 0);

analogWrite(sag\_motor\_1, data[1]);

analogWrite(sag\_motor\_2, 0);

}

if(data[1] < -50)//sağ

{

analogWrite(sol\_motor\_hiz, PWM);

analogWrite(sag\_motor\_hiz, PWM);

analogWrite(sol\_motor\_1, -data[1]);

analogWrite(sol\_motor\_2, data[1]);

analogWrite(sag\_motor\_1, 0);

analogWrite(sag\_motor\_2, 0);

}

if(data[0] > -50 && data[0] < 50 && data[1] > -50 && data[1] < 50)//dur

{

analogWrite(sol\_motor\_hiz, 0);

analogWrite(sag\_motor\_hiz, 0);

analogWrite(sol\_motor\_1, 0);

analogWrite(sol\_motor\_2, 0);

analogWrite(sag\_motor\_1, 0);

analogWrite(sag\_motor\_2, 0);

}

}

}

**TRANSMITTER CODE:**

#include <SPI.h>

#include <RF24.h>

#include <Wire.h>

#include <MPU6050.h> //Mpu6050 kütüphanesi

MPU6050 ivme\_sensor;

int x, y, z; //ivme tanımlama

RF24 radio(7, 8); // CE, CSN

int data[2]; // X ve Y düzlemi için dizi tanımlama

void setup()

{

Wire.begin();

ivme\_sensor.initialize();

radio.begin();

radio.openWritingPipe(1234);

Serial.begin(9600);

}

void loop()

{

ivme\_sensor.getAcceleration(&x, &y, &z); // ivme ve gyro değerlerini okuma

data[0] = map(x, -17000, 17000, -255, 255 ); //X düzleminin verisi (ileri-geri)

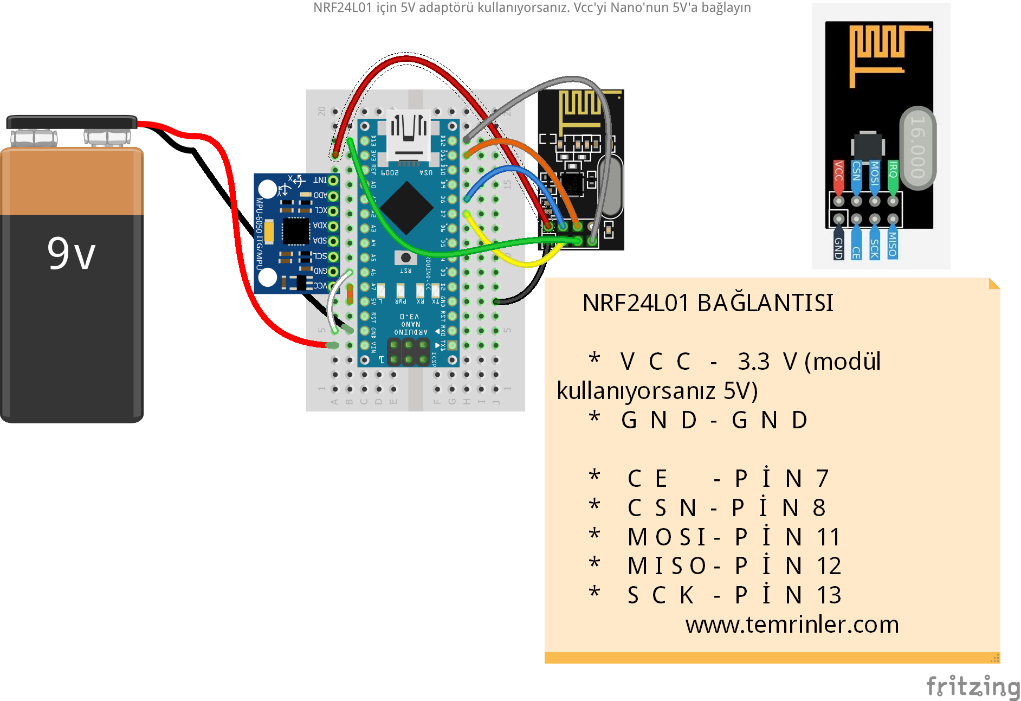
data[1] = map(y, -17000, 17000, -255, 255); //Y düzeleminin verisi (sağ-sol)

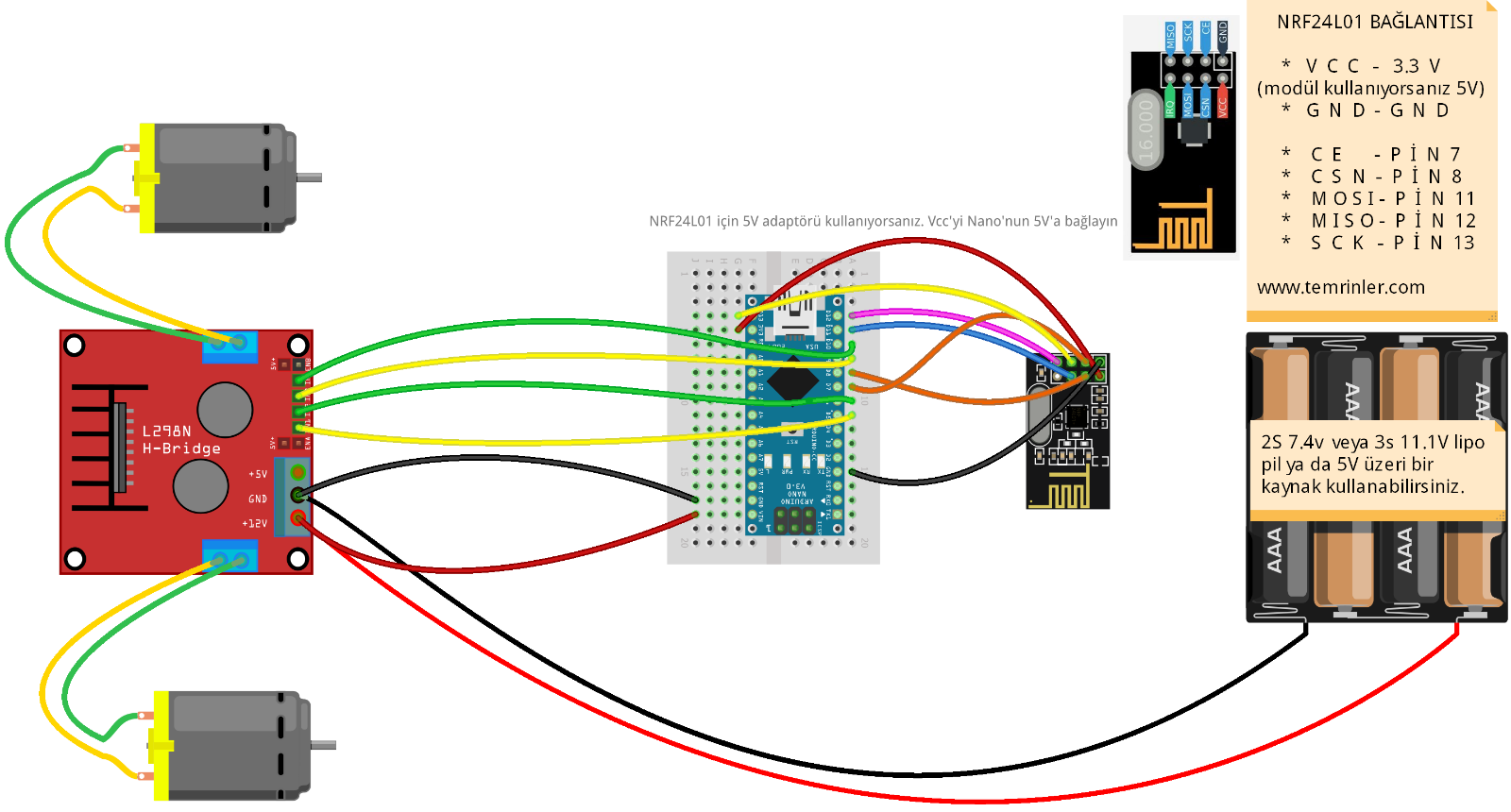
radio.write(data, sizeof(data));

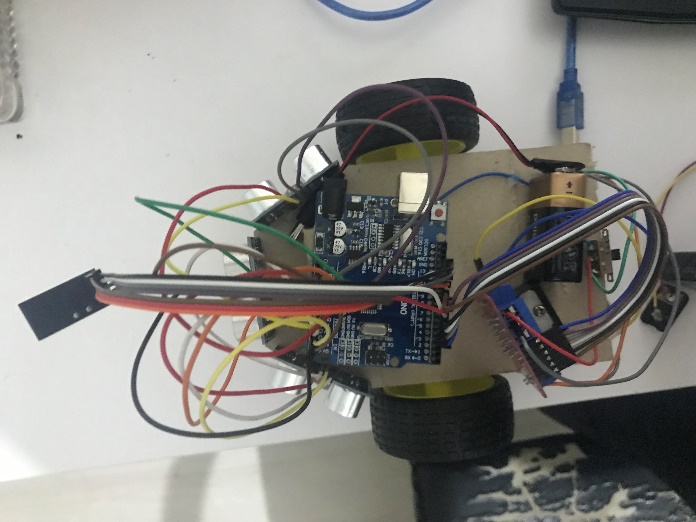
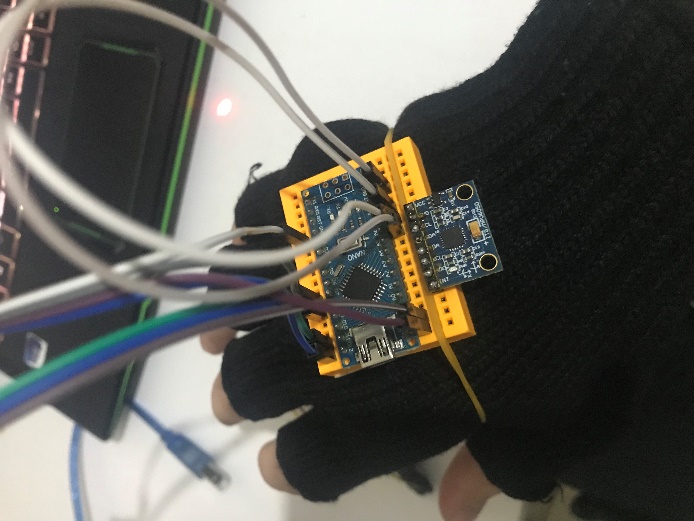
Serial.print("x axis: ");Serial.print(data[0]);Serial.print(" y axis: ");Serial.println(data[1]);

}

**4. Design and Assembly**

Transmitter part: We place the pieces on the miniboard as shown in the figure.

Receiver Part: We place the pieces on the miniboard as shown in the figure.



**5. Results and Discussions**

Throughout the process, we had to change our codes since we implemented new devices. So that was our main struggle during the assignment. Second Arduino was needed to use hand gestures to command the robot so we bought Arduino uno for this job. Since Arduino uno and Arduino nano have different type of port, we also found another connection cable to encode the Arduino nano. After all the encoding and assembling stuff is done, we carried out our first try. After setting the switch on which is on the robot, suddenly a smoke came out from battery. So we immediately remove the battery. Then we recognized the failure which is the two cables are touching each other and it is causing short circuit. So we fixed it. The other problem was in rotation of the robot. To achieve rotation motion, our design was like that: to turn left, left wheel should have stopped and vice versa. Instead of it, our wheels were rotating reverse directions therefore it was spinning. After reviewing the codes again, we successfully fixed that problem as well.

**6. Conclusions**

In this lab session; thanks to the controllers we added, we transferred the control of our robot, which came out of the labyrinth we made in the previous session, to our own gloves. As an achievement, this allowed us to better use the controllers we use and to do more detailed research on their software. The small command mistakes we made in the software part made us be more careful because this situation negatively affected the movement of our robot.

A camera setup we will add to our system can improve our robot in order to collect information more easily in the research area and in places that are difficult to enter. We can also make our design more comprehensive with the parts we will add to the control glove.