KARABÜK ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

BLM439 - ROBOT TEKNOLOJİLERİ
PROJE: LABİRENT ÇÖZEN ROBOT

DANIŞMAN

DR.ÖĞR.ÜYESİ FERHAT ATASOY

HAZIRLAYANLAR

AHMET TALHA ÇAKIROĞLU - 2014010206076

İBRAHİM HALİL HAĞİ – 2014010206044

ÖNSÖZ

Biz, proje seçerken robot teknolojileri ve robotik anlamda bize iyi kazanımlar sağlayacak hedefler belirledik. Bu anlamda yaptığımız bu proje, bize dersin amacını hatırlatmış oldu. Bireysel olarak bu alanda nelerin yapıldığını ve nasıl yapıldığını araştırarak, gözlemleyerek öğrenmiş ve uygulamış olduk. Bunun yanı sıra mesleki anlamda olmayan ama bir projede yapılması gereken çözüm analizlerini de yaşayarak öğrenmiş olduk.

Bu projeden çıkaracağımız sonuçlar aslında bitmedi daha yeni başlıyor diyebiliriz. Oluşturduğumuz bu projeyi daha da geliştirerek insanoğlunun yaşadığı problemlere çözüm olacağını düşünmekteyiz. Nitekim yaptığımız bu proje daha önce yapılmamış spesifik bir projedir.

Bu bağlamda bize yol gösteren ve araştırmaya sevk eden değerli hocalarımıza teşekkür ederiz.

Ahmet Talha ÇAKIROĞLU

İbrahim Halil HAĞİ

İÇİNDEKİLER

ÖZET

Projemizde Arduino UNO kullanarak dengede duran ve çizgi izleyen robot oluşturduk. Bu iki kriteri birleştirerek bir çizgi labirenti çözdük. Bu projeye başlamadan önce Arduino'nun yapısını ve temel elektronik bilgilerin ışığında hareket ettik.

Ardından dengede duran ve çizgi izleme için malzemeleri araştırdık ve en uygun olanları seçerek onların data sheetlerini inceleyerek robotumuzun taslağını oluşturduk.

Projemizde 2 adet sensör kullandık. Bunlardan biri denge için diğeri ise çizgi izleme için kullanıldı.

Denge için MPU6050 3 eksenli Gyro ve Accelometer'ı tercih ettik. Bu sensör eğim verilerini alarak motorları döndürmemizi sağladı.

Çizgi için ise TCRT5000L kızılötesi sensörünü kullandık. Bu sensör ise siyah çizgi gördüğünde 1 sinyal değerini üreterek robotun çizgi izlenmesi sağlanmıştır. Böylece beyaz zemin üzerinde siyah çizgi ile oluşturduğumuz labirenti çözme imkanı verdi.

Motor sürücü olarak L298N entegresini kullandık. Bu motor sürücüsü bize 2 DC motor sürebileceğimiz anlamına geldiğinden bunu tercih ettik.

Bu projenin kullanım açısından örnek vermek gerekirse, insan elinin giremediği dar alanlara koyarak sorunu çözmesi amaçlanmaktadır. Örneğin göçük, su boruları içi vb.

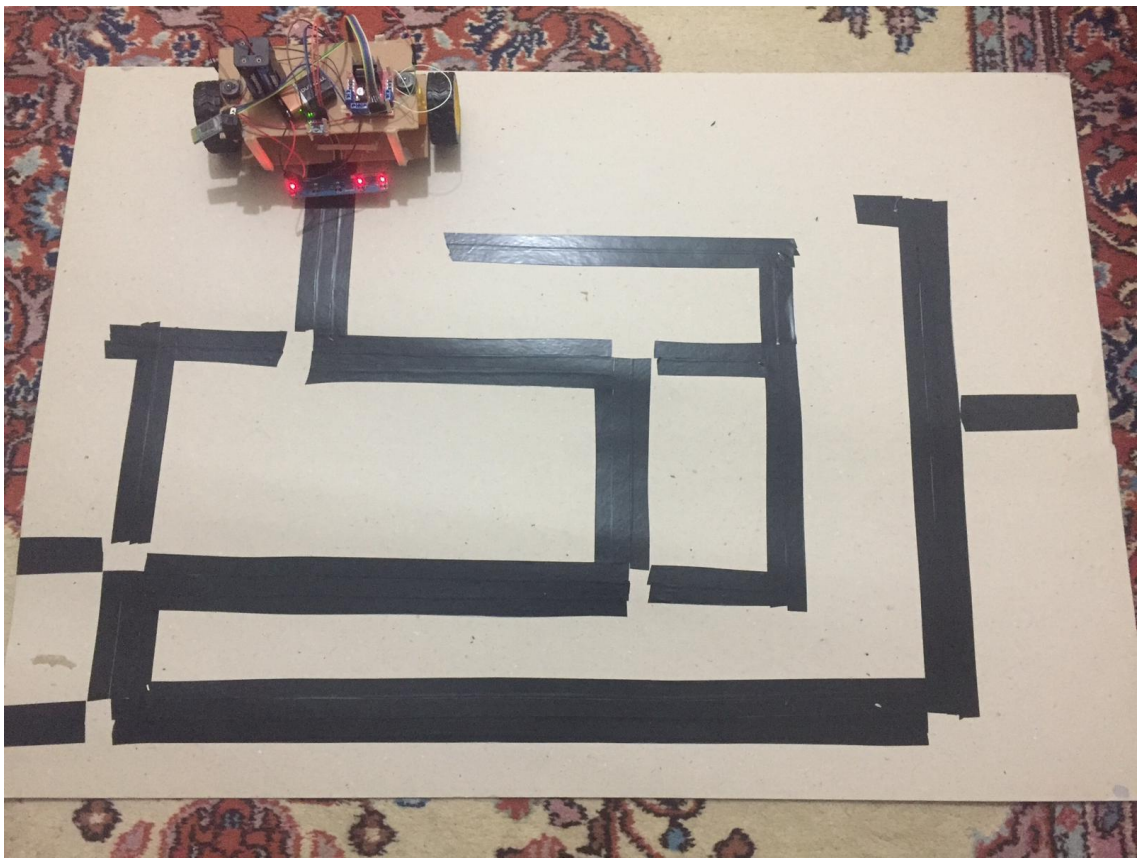
BAŞARI KRİTERLERİMİZ

1. Robotun kendi dengesini sağlaması.
2. Oluşturan Çizgi labirentini çözmesi.
3. Robotun labirentte gittiği yolu bilgisayar ekranında çizmesi.

1.GENEL BİLGİLER

1.1. GİRİŞ

Projemizin temel konusu aslında labirent çözmektir. Bu labirenti çizgi olarak tasarladık. Bunun yanı sıra robot labirentte giderken aynı zamanda kendi dengesini de sağlayacak ve böylece daha esnek bir yapıya sahip olacaktır. Diğer labirent çözen robotlardan farkı budur.



1.2 KULLANILAN MALZEMELER

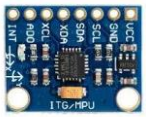
1.2.1 ARDUİNO UNO

Arduino Uno'yu kullanma sebebimiz, açık kaynak olan bir platform olmasıdır. Ayrıca robot alanında çok tecrübeye sahip olmadığımızdan kullanması kolay ve ucuz olanı seçmek zorundaydık.

Arduino' yu programlarken Arduino IDE kullandık.

1.2.2 MPU6050 SENSÖR

MPU6050 3 eksenli bir jiroskop(gyro) ve 3 eksenli açısal ivme ölçen(accelometer) barındıran 6 eksenli bir karttır. I^2C haberleşmesi kullanarak bu verileri Arduino'ya gönderir ve programlanır. Programın içerisinde bize motorları kontrol etme becerisi sağlar.



1.2.3 BLUETOOTH MODÜLÜ

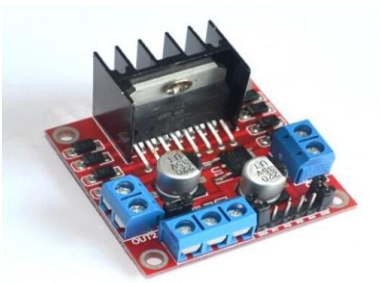
Kullanacağımız bluetooth modülüyle robotun eş zamanlı olarak konum bilgisini alarak bunu C#'ta yapacağımız uygulama ile gittiği yolu ekrana çizecek. Modül olarak HC-06 kullandık.

1.2.4 MOTOR (DC)

İki adet DC motor kullanıldı. Bunların çalışma voltajı 6V. Dönüş hızı ise 250RPM. Burada motorların dönüş hızı daha fazla olan bir DC motor seçebilirdik. Ama bu bir prototip olacağından çok masraflı bir robot yapmayı düşünmedik. Ayrıca bu motorlara uygun tekerlekle de birleştirip güzel bir takım oluşturduk.

1.2.5 L298N MOTOR SÜRÜCÜSÜ

Bu motor sürücüsünü seçmemizdeki amaç, 2 tane DC Motor kullanmamıza olanak sağladığıdan seçtik. Ayrıca motor bağlantıları kolay ve zahmetsiz.



1.2.6 BATARYA

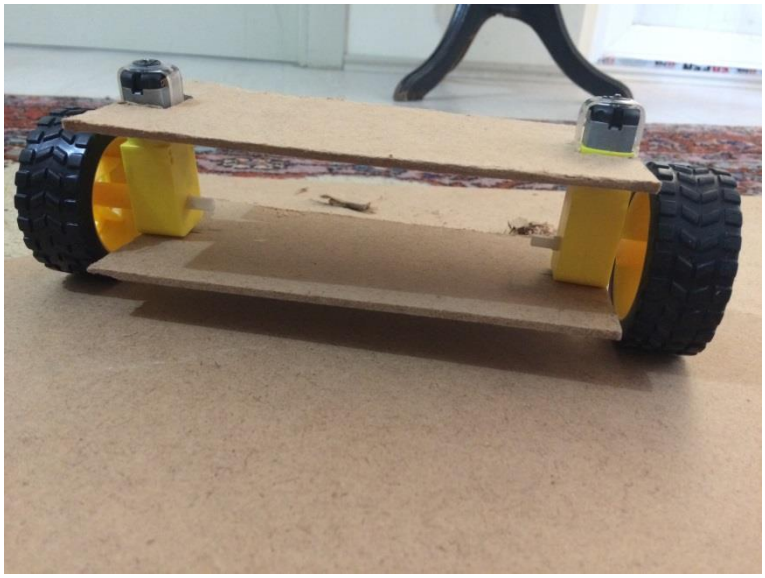
Devremize gücü 2 şekilde sağlıyoruz. Bunlardan birincisi Arduino UNO'ya 6V güç sağlayan piller kullandık. Motor sürücümüze ise 9V'luk pil kullandık.

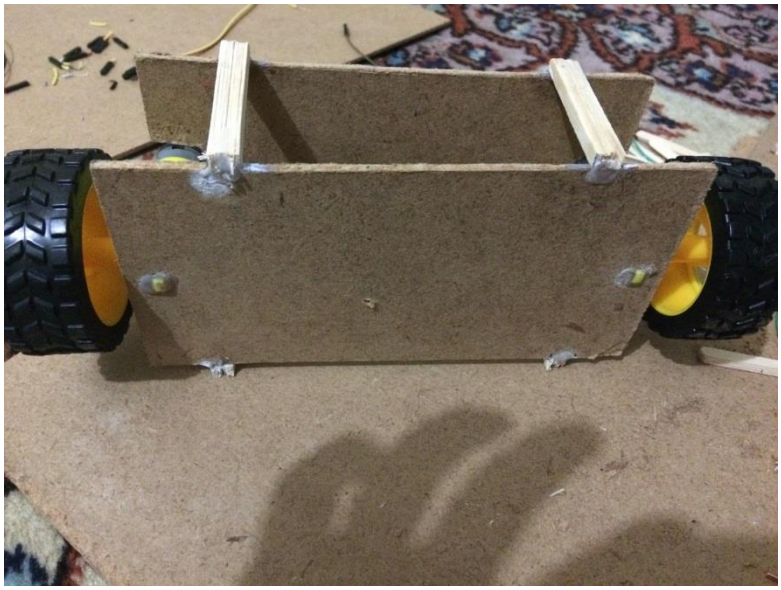
2.ÇALIŞMALAR

2.1 DONANIM

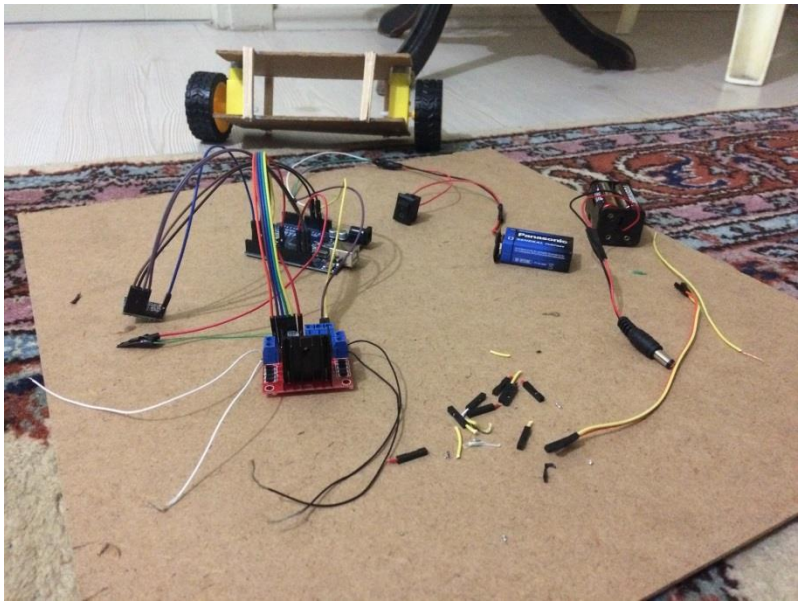
Robotun tasarımı belki de bizi en çok zorlayan kısımların başında yer aldı. Sebebi hazır kit kullanmadık. Bunun yerine robotumuzun tasarımını kendimiz yapmak istedik. Bu kit için ilk önce sunta bulduk ve bu suntadan 9cm genişliğinde 18cm uzunluğunda 2 parça kestik.

Bunları 2 kat olacak şekilde yerleştirdik. Bu iki suntayı tahta çubuklar ile yapıştırdık. Kestiğimiz bu parçalardan bir tanesinin sağına ve soluna motorlarımızı yerleştirdik.

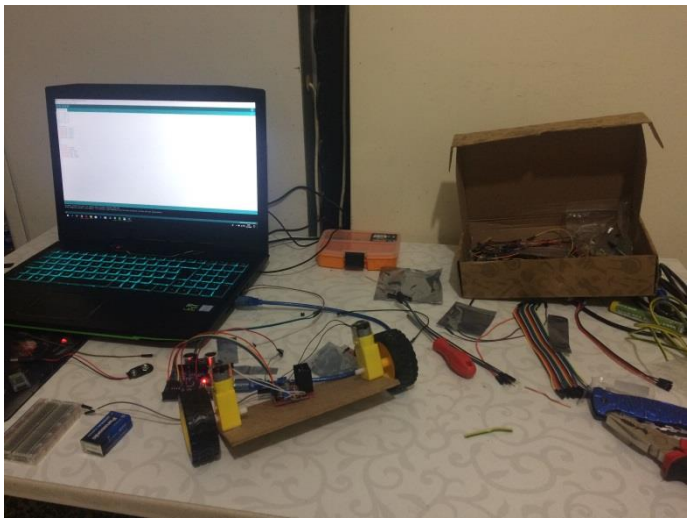




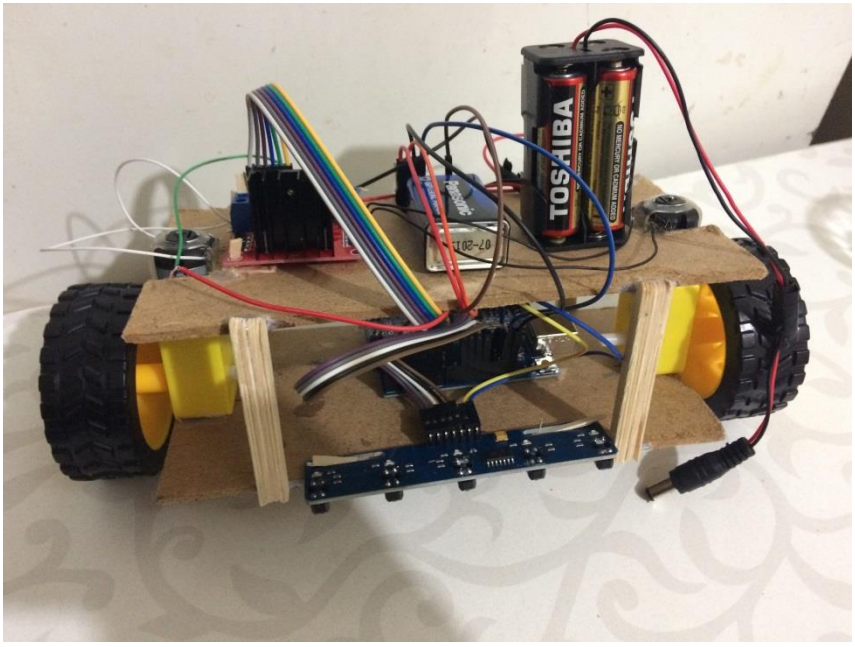
Bundan sonraki aşamada denge robotu yaptığımız için kullandığımız malzemelerin ağırlıklarına göre yerleştirmek istedik fakat bu o kadar kolay olmadı.



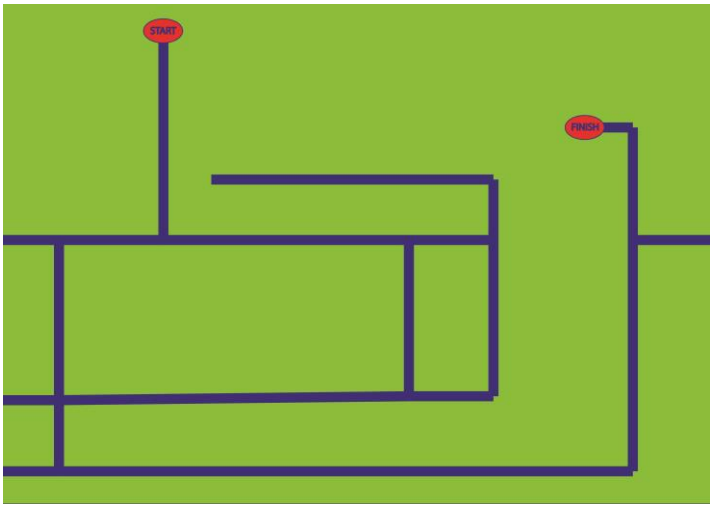
Bunu zor kılan olay ise Arduino'ya giden kabloların geçişlerini hesaplayamadık.

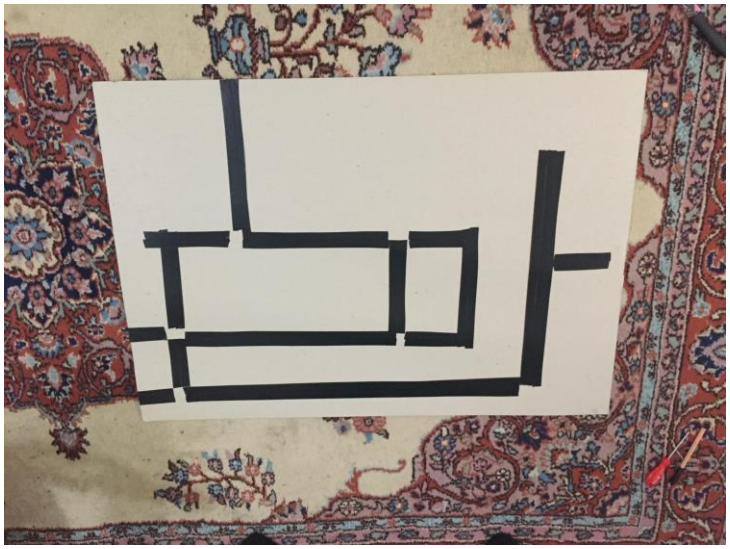


Buna rağmen robotun 1. Katına Arduino ve kızılötesi sensörlerimizi yerleştirdik. 2.Kat'a ise motor sürücümüzü, jiroskopumuzu, 1 adet motor sürücüsüne bağlı buton, Arduino'yu beslemek için 4 tane 1.5V'luk pil ve yatağı, motor sürücünü beslemek için 9V'luk pil ve son olarak bluetooth modülümüzü yerleştirdik. Bu aşamaların bazılarını fotoğraflama imkanı bulduk ve onlarda şekillerde görüldüğü gibidir.



Son olarak robotumuzun çözmesini isteğmiz bir labitent tasarlayıp bir platform oluşturduk. Elde imkanlarımızla biraz küçük platform oldu fakat prototip için gayet yeterli bir platform olduğumu düşünmekteyiz.





2.2 YAZILIM

Bu aşamada robotumuzun yapacağı işlemleri programladık. Yazım işi bizi en çok zorlayan kısımlardan biri oldu. Çünkü normalde var olan iki farklı programı birleştirmek için sıfırdan bir algoritma kullanmamız gerekti. Bunun için fazla mesai harcadık. Ama sonun da kodumuzu yazdık.

Kontrol sistemi olarak PID'yi kullandık ve bunu Arduino içerisine ekledik.

2.3.1 DENGE İÇİN MPU6050 SENSÖRÜNÜN TANIMLAMA KODU

```
//Gerekli kütüphaneler
#include <PID_v1.h>
#include <LMotorController.h>
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"
#define min_speed 30
SoftwareSerial bluetoothIletisim(10,9);//RX,TX -> Arduino da ters
MPU6050 dengee;

bool dmpDurum = true; //veri paketi
uint8_t mpuIntStatus;
uint8_t devStatus; //çalışırsa 0 çalışmazsa 1 değeri dönmesi sağlanacak
uint8_t fifoBuffer[64];
uint16_t packetSize;
uint16_t fifoSayac;


Quaternion q;
VectorFloat gravity;
float ypr[3];

double originalSetpoint = 178;
double setpoint = originalSetpoint;
double movingAngleOffset = 0.08;
double giris, cikis;

//PID tanımlaması
double Kp = 500;
double Kd = 100;
double Ki = 0.5;
PID pid(&giris, &cikis, &setpoint, Kp, Ki, Kd, DIRECT);

//jiroskop kesmesini ilk başta false olarak tanımlıyoruz
volatile bool mpuInterrupt = false;

void dmpDataReady()
{  //eğer veri paketi gönderimi sağlanırsa jiroskop kesmesini true yapıyoruz
   mpuInterrupt = true;
}
```

2.3.2 DENGE İÇİN MPU6050 SENSÖRÜNÜN SETUP KODU

```
void setup()
{
Wire.begin();
   dengee.initialize();
   devStatus = dengee.dmpInitialize();
   //jiroskop x,y,z varsayılan değerlerimiz
   dengee.setXGyroOffset(200);
   dengee.setYGyroOffset(85);
   dengee.setZGyroOffset(-65);
   dengee.setZAccelOffset(1750);

   if (devStatus == 0)//çalışır
   {
     dengee.setDMPEnabled(true); //veri transferi -dmp- başlat
     attachInterrupt(0, dmpDataReady, RISING);
     mpuIntStatus = dengee.getIntStatus();
     dmpDurum = true;
     packetSize = dengee.dmpGetFIFOPacketSize();
     pid.SetMode(AUTOMATIC);
     pid.SetSampleTime(10);
     pid.SetSampleTime(5);
     pid.SetOutputLimits(-255, 255);
   }

   else
   {
     Serial.print(("DMP başlatılamadı..."));
     Serial.print(devStatus);
   }
}
```

2.3.3 DENGE FONKSİYONU

```
void denge(){
 if (!dmpDurum)//veri transferi yoksa başlama
   {
    return;
   }
   while (!mpuInterrupt && fifoSayac < packetSize)
   {
      pid.Compute();
      motorController.move(cikis, min_speed);
   }

   mpuInterrupt = false;
   mpuIntStatus = dengee.getIntStatus();
   fifoSayac = dengee.getFIFOCount();
  //taşma kontrol
 if ((mpuIntStatus & 0x10) || fifoSayac == 1024)
   {
    dengee.resetFIFO();
    Serial.println(("FIFO taştı!"));
   }
 else if (mpuIntStatus & 0x02)
   {
    while (fifoSayac < packetSize) fifoSayac = dengee.getFIFOCount();
    dengee.getFIFOBytes(fifoBuffer, packetSize);
    fifoSayac -= packetSize;
    dengee.dmpGetQuaternion(&q, fifoBuffer);
    dengee.dmpGetGravity(&gravity, &q);
    dengee.dmpGetYawPitchRoll(ypr, &q, &gravity);
    giris = ypr[1] * 180/M_PI + 180;
   }
}
```

2.3.4 ÇİZGİ İÇİN KIZILÖTESİ SÖNSÖR TANIMLAMA ve SETUP KODU

```
double solMotor = 0.8;
double sagMotor = 0.6;

//arduino pin bağlantıları
const int sag_i=7;
const int sag_g=6;
const int sol_i=5;
const int sol_g=4;
const int enb=3;
const int ena=8;
const int sol_sensor = 11;
const int sag_sensor = 12;
const int bitirme_sensor = 13;
int sol_durum, sag_durum, bitirme_durum;
LMotorController motorController(ena, sag_i, sag_g, enb, sol_i, sol_g, solMotor, sagMotor);

void setup()
{
//TCRT5000L setup ve Dc motor setup
  pinMode(sag_i, OUTPUT);
  pinMode(sag_g, OUTPUT);
  pinMode(sol_i, OUTPUT);
  pinMode(sol_g, OUTPUT);
  pinMode(enb, OUTPUT);
  pinMode(ena, OUTPUT);
  pinMode(sag_sensor, INPUT);
  pinMode(sol_sensor, INPUT);
  pinMode(bitirme_sensor, INPUT);
}
```

2.3.5 LABİRENT FONKSİYONU

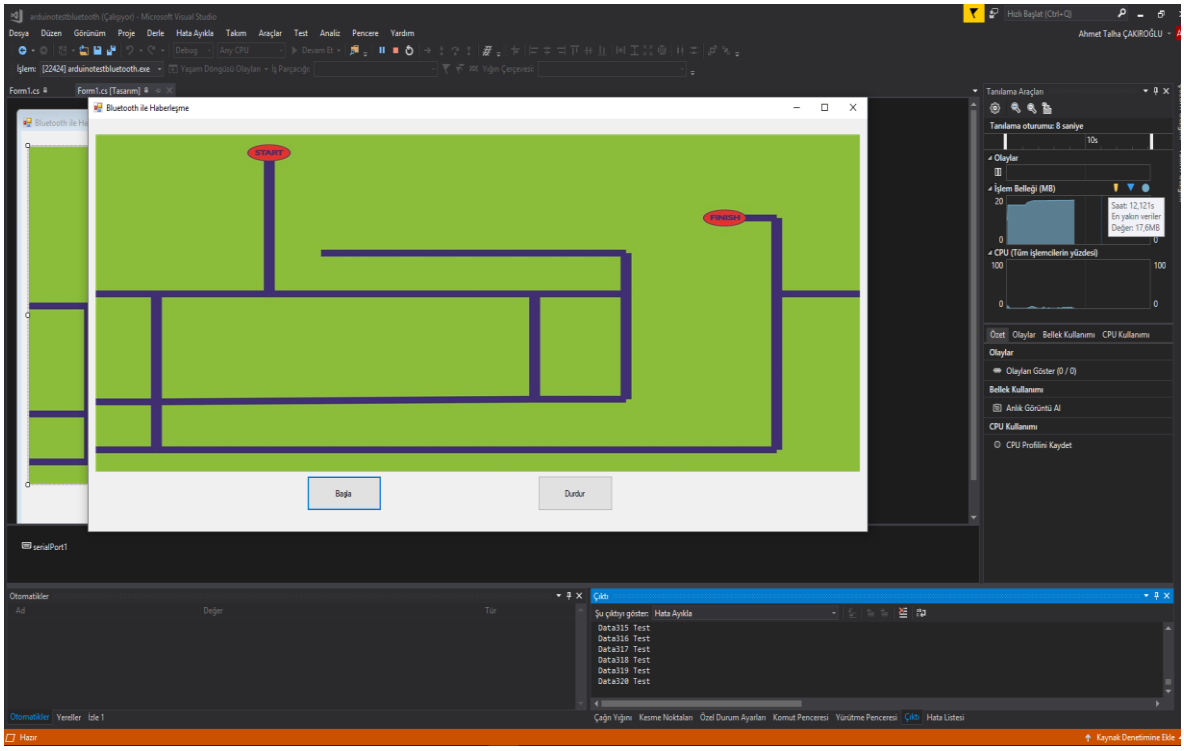
```
void labirent(){
  //kızılötesi sensörlerden okunan değerleri kaydetme
  sol_durum = digitalRead(sol_sensor);
  sag_durum = digitalRead(sag_sensor);
  bitirme_durum = digitalRead(bitirme_sensor);
  digitalWrite(enb, HIGH);
  digitalWrite(ena, HIGH);

  if (sol_durum == LOW && sag_durum == LOW) //ikiside siyah okursa
  { digitalWrite(enb, HIGH);
    digitalWrite(ena, HIGH);
    digitalWrite(sag_i, HIGH);
    digitalWrite(sag_g, LOW);
    digitalWrite(sol_i, HIGH);
    digitalWrite(sol_g, LOW);
  }
  else if (sol_durum == LOW && sag_durum == HIGH)  // sol siyah okursa
  { digitalWrite(enb, HIGH);
   digitalWrite(ena, HIGH);
   digitalWrite(sag_i, HIGH);
   digitalWrite(sag_g, HIGH);
   digitalWrite(sol_i, HIGH);
   digitalWrite(sol_g, LOW);
  }
  else if (sol_durum == HIGH && sag_durum == LOW) //sağ siyah okursa
  { digitalWrite(enb, HIGH);
   digitalWrite(ena, HIGH);
   digitalWrite(sag_i, HIGH);
   digitalWrite(sag_g, LOW);
   digitalWrite(sol_i, HIGH);
   digitalWrite(sol_g, HIGH);
  }
  else if (sol_durum == HIGH && sag_durum == HIGH && bitirme_durum == LOW) //sağ, sol siyah okumaz bitirmeSensör okursa dur.
  {
   digitalWrite(enb, HIGH);
   digitalWrite(ena, HIGH);
   digitalWrite(sag_i, LOW);
   digitalWrite(sag_g, LOW);
   digitalWrite(sol_i, LOW);
   digitalWrite(sol_g, LOW);
  }
  else  //hiçbirine girmezse yine dur
  { digitalWrite(enb, HIGH);
   digitalWrite(ena, HIGH);
   digitalWrite(sag_i, LOW);
   digitalWrite(sag_g, LOW);
   digitalWrite(sol_i, LOW);
   digitalWrite(sol_g, LOW);
  }
}
```

2.3.6 BLUETOOTH MODÜLÜ TANIMI VE SETUP KODU

```
#include <SoftwareSerial.h> // seri  haberleşme kütüphanesi

SoftwareSerial bt_iletisim(10,9);//RX,TX bağlantı pinleri Arduinoda  ters olmalıdır.

void setup()
{
  Serial.begin(9600);
  bluetoothIletisim.begin(9600);
}
```

2.3.7 BLUETOOTH MODÜLÜ PROGRAMI VISUAL STUDİO ÇIKTISI

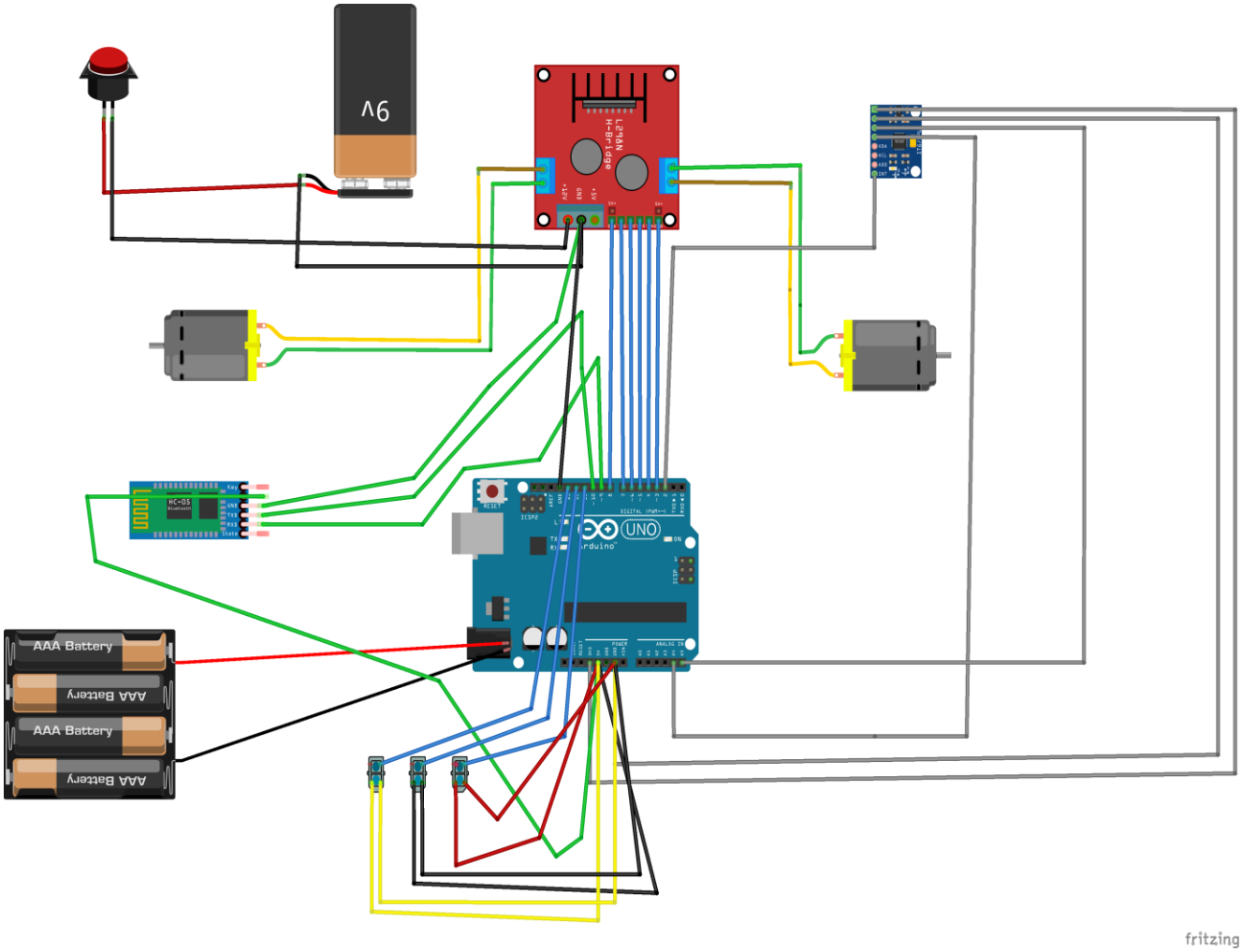
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO.Ports;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace arduinotestbluetooth
{    public partial class Form1 : Form
    {   int count = 0;
        public Form1()
        {   InitializeComponent();
            serialPort1.PortName = "COM9";
            serialPort1.BaudRate = 9600;
            try
            {  if (!serialPort1.IsOpen)
                    serialPort1.Open();}
            catch(Exception ex)
            {MessageBox.Show(ex.Message);}
              serialPort1.DataReceived += new System.IO.Ports.SerialDataReceivedEventHandler(DataReceived);
        }
        private void DataReceived(object sender, System.IO.Ports.SerialDataReceivedEventArgs e)
        {  //throw new NotImplementedException();
            try{SerialPort sp1 = (SerialPort)sender;
                Console.Write("Data"+count+" "+sp1.ReadLine()+ "\n");count++;}
            catch(Exception ex)
            {}}}}
```



2.3.8 LOOP FONKSİYONU

```
void loop() {
      denge();
      if(mpuIntStatus  == false) //kesme bilgisi yoksa, robot dengededir bu yüzden labirent çöz
       {
         labirent();
         bluetoothIletisim.println("Veri Gönderimi");
       }
      denge();
 }
```

2.3.9 DEVRE ŞEMASI – SİMULASYON (FRTİZİNG)



3. KARŞILAŞILAN SORUNLAR VE ÇÖZÜMLERİ

Robotumuzun malzemelerini seçerken karşılaştığımız sorunlardan bir tanesi DC motor seçimiydi. Burada motorların çalışma voltajı ve dönüş hızlarının önemli bir seçim kriteri olduğunu fark ettik. Ve bu motorlara uygun tekerleiği seçerken ikisi bir arada olan hazır motor kitini tercih ederek ve hocalarımıza danışarak bu sorunu çözdük.

Motor sürücüsünü seçerken de çok ikilemde kaldık. Zira piyasada bir sürü motor sürücüsü var ama biz hangisini kullanacağımız bilmiyorduk. Araştırdık ve öğrendik ve bu bağlamda L298N sürücüsünün 2 adet DC motor çallıştırabileceğini gördük ve onu kullandık.

Robotumuzun kitinin dengesini bulamadık. Bunun için kullandığımız malzemelerin ağırlıklarıyla bir denge oluşturduk.

Yazılımda karşılaştığımız sorunlar belkide bu projeyi yaparken en çok zorlandığımız kısımlardan bir tanesidir. Bu sorun denge ve labirent fonksiyonlarını harmanlayıp tek bir fonksiyonda yazma işlemydi. Loop fonksiyonunda direk yazdığımızda motorlar kararsız halde kalıyordu. Bunun için bir algoritma geliştirerek bu sorundan kurtulduk ve motorlar kararlı halde çalışmaya başladı.

ÖZEL TEŞEKKÜR

Başta bu projemizde bize her türlü konuda ve robot alanında bize yardımcı olan;

DR. ÖĞR. Ferhat ATASOY'a

Geçmiş senelerde bize mikroişlemci bilgilerini sunan;

Öğr. Gör. Kürşat Mustafa KARAOĞLAN'a

Robotun mekaniği ile ilgili bize bilgilerini sunan;

Özer KILIÇ'a

Teşekkürlerimizi iletiyoruz. Saygılarımızla.

KAYNAKÇA

https://lezzetlirobottarifleri.com

https://www.robotistan.com

http://www.robotdevreleri.com/

http://www.temrinler.com

http://elektronikhobi.net/pid-kontrol-nedir-basitlestirilmis-anlatim/

https://www.projehocam.com/pid-kontrol-algoritmasi-nedir/

https://www.youtube.com/user/MerakliMaymunn

https://www.youtube.com/channel/UCLqQ5u1b5Q-LFfcUCYiKd4Q

https://www.robimek.com

http://www.csharpuygulamalar.com/

https://drive.google.com/drive/folders/1RI94t2C6TeIP9wSpZUBWLX24QaZefY4N