

**T.C.**

**MİLLİ EĞİTİM BAKANLIĞI**

**SİVAS MERKEZ**

**Sivas Fen Lisesi**

**Bilgisayar Bilimi Dersi**

**Rapor**

|  |  |
| --- | --- |
| **Rapor No** | Proje-1 |
| **Rapor Tarih** | 16.01.2018 |
| **Proje Adı** | Arduino ile akıllı otopark |

Bilgisayar Bilimi Öğretmeni

Ersin TÜTÜNCÜ

2017-2018



**T.C.**

**MİLLİ EĞİTİM BAKANLIĞI**

**SİVAS MERKEZ**

**Sivas Fen Lisesi**

**Proje Grup**

|  |  |  |
| --- | --- | --- |
| **Proje Görev** | **Numara** | **Ad-Soyad** |
| Proje Yönetimi | 151 | Emine GÖLLÜCE |
| Doküman Yönetimi | 317 | İrem DURAN |
| Lojistik Yöneti | 202 | Gülçin Yiğit |
| Yazılım Geliştirme | 164 | Bekir ALEGÖZ |
| Web ve GitHub Yönetimi | 303 | Hatice İNCE |
| Sunum Yönetimi | 438 | Nuri Mert GEDİKLİ |

**İÇİNDEKİLER**

[ÖZET 3](#_30j0zll)

[Anahtar Kelimeler 3](#_1fob9te)

[ABSTRACT 3](#_3znysh7)

[Key Words 3](#_2et92p0)

[Proje Görev Dağılımı Listesi ve Görev Dağılımı Açıklaması 4](#_tyjcwt)

[∙](#_3dy6vkm) Görev Dağılımı ve Sorumlusu 4

[∙](#_1t3h5sf) Görev süresince sürdürülen eylemler 5

[∙ Görevlerin iş yükü şeması](about:blank) 6

[∙](#_2s8eyo1) Yoklama Çizelgeleri 7

[∙](#_17dp8vu) Haftalık İş Katkı Cetvelleri 8

[GİRİŞ 8](#_3rdcrjn)

[1.Projenin Açıklaması 8](#_26in1rg)

[2.UML Diyagramlar 8](#_lnxbz9)

[3.Donanım Yapısı: 9](#_35nkun2)

[a.Gömülü Sistemler Mimarisi ve Devre Tasarımı 9](#_1ksv4uv)

[b.Mekanik Sistem Mimarisinin Tanıtılması: 12](#_44sinio)

[PID 17](#_2jxsxqh)

[4.Yazılım Yapısı 18](#_z337ya)

[SONUÇ 19](#_3j2qqm3)

[1.Bilgi Düzeyine Katkıları: 19](#_1y810tw)

[2.Teknolojik Katkıları: 19](#_4i7ojhp)

[3.Ekip Çalışması Katkıları 19](#_2xcytpi)

[4.Aksayan Yönler: 20](#_1ci93xb)

[5.Görüş ve Öneriler: 20](#_3whwml4)

**ÖZET**

Proje kapsamında elimizdeki malzemelerle minyatür bir akıllı otopark yaptık .

**Anahtar Kelimeler**

Arduino, akıllı otopark, RFID,sensör, led lamba

**ABSTRACT**

We did miniature parking with the supplies in the scope of the project

**Key Words**

Arduino, miniature parking,RFID.

**Proje Görev Dağılımı Listesi ve Görev Dağılımı Açıklaması**

* **Görev Dağılımı ve Sorumlusu**

|  |  |  |
| --- | --- | --- |
| Proje Yönetimi | Görev dağılımı ve takibinden sorumlu kişi, aynı zamanda proje grubunun çalışma takvimini ve düzenini ayarlamaktadır.Grupta bulunan kişilerle iletişim halinde olup projenin yönetimini sağlar. | Emine GÖLLÜCE |
| Döküman Yönetimi | Projenin tüm tasarım ve çizimlerinden,proje raporunun sunulmasından,dökümanların uygun forma getirilmesinden kodlamaya ait diagram ve modellerin hazırlanması ve web sitesi tasarımından sorumlu olan kişidir. | İrem DURAN |
| Lojistik Yönetimi | Projede kullanılacak tüm elemanların, malzemelerin belirlenmesi ve temin edilmesi,en uygun tasarımın yapılması için geliştirmelerin yapılmasıyla ve projenin donanımsal kısmının tanıtılması ile ilgilenen kişidir. | İrem DURNAN  Hatice İNCE  Nuri Mert GEDİKLİ  Gülçin Yiğit  Bekir ALEGÖZ  Emine GÖLLÜCE |
| Yazılım Geliştirme Yönetimi | Yazılım için araştırmaların yapılması, yazılım aşamalarının proje grubuna dağıtılması,Yazılım ile ilgili raporların hazırlanarak ilgili bölüme(döküman yönetimine) aktarılması yazılım ve süreç testlerinin gerçekleştirilmesi ile ilgilenen kişidir. | Bekir ALEGÖZ |
| WEB ve GitHub Yönetimi | Proje tanıtımı için WEB sayfasının hazırlanması, projenin GitHub yönetiminin yapılması,döküman yöneticisinden almış olduğu raporlar ile WEB sitesine ve GitHub'a işlemekle sorumlu olan kişidir. | Hatice İNCE |
| Sunum Yönetimi | Proje teslim zamanında sunumun, yapılan tüm işlemlerin uygun bir biçimde anlatılmasından,rapor ve evrakların eksiksiz bir şekilde sunulmasından ve önerilere,  sorulara uygun çözümler üretmekten sorumludur. | Gülçin Yiğit  Nuri Mert GEDİKLİ |

* **Görev süresince sürdürülen eylemler**

|  |  |
| --- | --- |
| 1.Hafta | Proje Hakkında Bilgi Edinme |
| 2.Hafta | Malzeme Seçimi |
| 3.Hafta | Mekanik ve Elektronik Tasarım |
| 4.Hafta | Yazılım |
| 5.Hafta | Grup elemanlarına ait iş yükünün tamamlanması (rapor,web,github) |

SUNUM

## Görevlerin iş yükü şeması

* **Yoklama Çizelgeleri**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tarihler**  **Grup** | **12.03.2018** | **19.03.2018** | **26.04.201** | **02.05.2018** | **09.05.2018** |
| **Emine GÖLLÜCE** | + | **+** | **+** | **+** | **+** |
| **Hatice İNCE** | **+** | ***+*** | **+** | **+** | **+** |
| **Nuri Mert GEDİKLİ** | **+** | **+** | **+** | **+** | **+** |
| **Gülçin YİĞİT** | **+** | **+** | **+** | **+** | **+** |
| **İrem DURAN** | **+** | **+** | **+** | **+** | **+** |

* **Haftalık İş Katkı Cetvelleri**

|  |  |
| --- | --- |
|  | **Yapılan iş** |
| **12.03.2018** | Proje araştırması |
| **19.03.2018** | Malzeme Teminatı |
| **26.04.2018** | Elektronik ve Mekanik Tasarım |
| **02.05.2018** | Yazılım |
| **09.05.2018** | Deneme ve Test Aşamaları |

**GİRİŞ**

**1.Projenin Açıklaması**

Yapılan projede amaç; ilk önce rfıd kartını okutarak kapının açılması eğer yanlış kartı okuttuysak kapının alarmının ötmesi daha sonra. Mesafe sensörleri ile arabanın yakınlığını anlayıp ona göre led lambalardan kırmızı ya da yeşil olanı yakması .

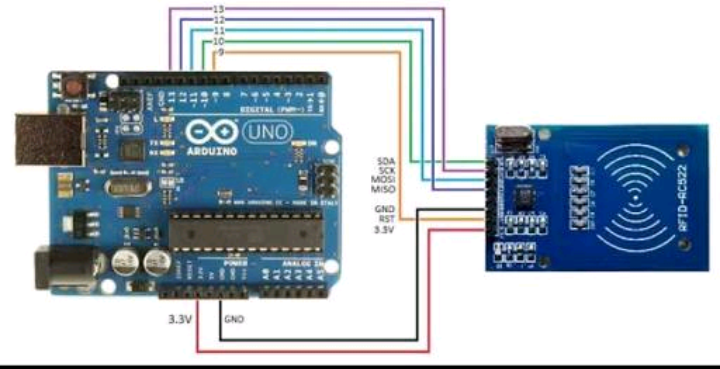
**3.Donanım Yapısı:**

**a.Gömülü Sistemler Mimarisi ve Devre Tasarımı**

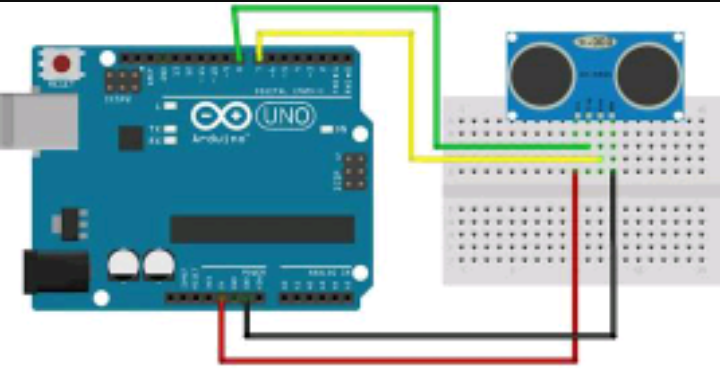
|  |  |
| --- | --- |
|  | **Arduino Uno R3**  Arduino Uno; Atmega328 temelli bir mikrodenetleyici kartıdır. Üzerinde 14 adet dijital giriş/çıkış pini (6 tanesi PWM çıkışı olarak kullanılabilir), 6 analog giriş, 16Mhz kristal, usb soketi, güç soketi, ICSP konektörü ve reset tuşu bulundurmaktadır. Kart üzerinde mikrodenetleyicinin çalışması için gerekli olan her şey bulunmaktadır. Kolayca usb kablosu üzerinden bilgisayara bağlanabilir, adaptör veya pil ile çalıştırılaiçindi |
|  | **Servo motor**  Servo mekanizmaların açısal doğrusal  Pozisyon hız ve ivme kontrolünü hatasız bir şekilde yapantahrik sistemi olarak tanımlanır yani hareket kontrolü yapan bir düzenektir |
|  | **Mesafe Sensörü**  Robot ve otomasyon projelerinde bazen sadece engeli algılamak yeterli olmayabilir. Mesafe algılaması gerektiren durumlardan SHARP mesafe algılayıcı sensörler kullanılır |
|  | **Miniboard**  Mavi mini breadboard 170 deliğe sahip ve çeşitli yüzeylere yapıştırabilmeniz için altı yapışkanlıdır. Ufak devre çalışmalarınızda ve devre kartlarının üzerine yapıştırarak hızlı bir şekilde prototipleme işlemini gerçekleştirebilir, devrelerinizi çalıştırabilirsiniz.  Boyutları: 45,72x35,56mm. |
|  | **Jumper Kablo (E-E/D-E)**  Devre elemanlarının bağlantılarını gerçekleştirmek için kullanabiliriz. |
|  | **Led**  Yeşil ve kırmızı ledler ardunio ile komuta göre çalışırlar . Yanma süreleri kodlar ile ayarlanır . |
|  | **RFID kart**  **:** |
|  |  |

**Devre Tasarımı:**

**RFID ardunio bağlantısı**

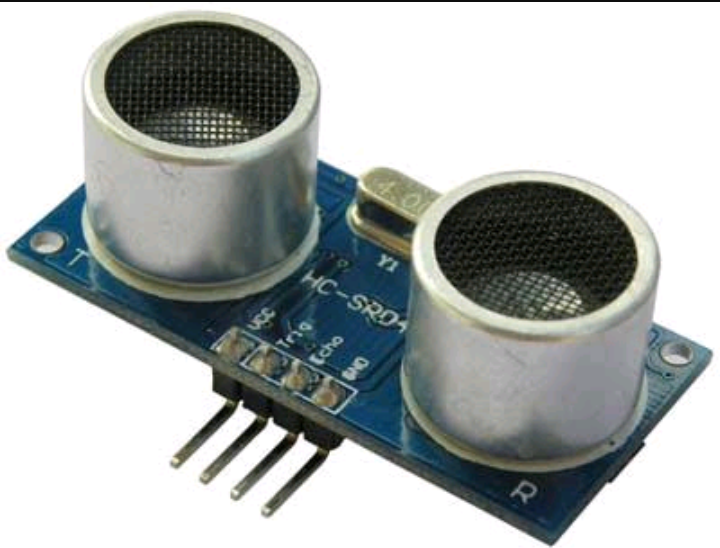
****

**Ardunio mesafe sensörü bağlantısı**

****

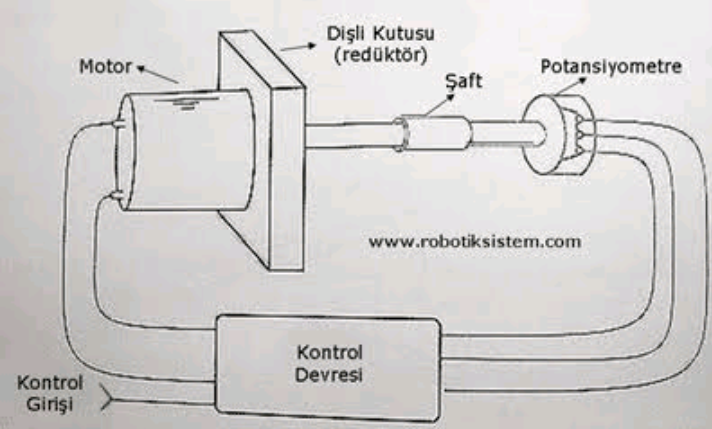
**b.Mekanik Sistem Mimarisinin Tanıtılması:**

Gömülü sistemlerin kapsamlı tanıtılması

**1.Mesafe sensörü**

**Çalışma voltajı 5 V DC  
Çalışma frekansı 40 Hz  
Çektiği akım 15 mA  
Boyutları 45 mm x 20 mm x 15 mm  
Görme açısı 15 derece  
Tetik bacağı giriş sinyali 10 us TTL Darbesi  
Echo çıkış sinyali Giriş TTL sinyali ve Mesafe Oranı**

**2.Servo motor**

****

**Servo, mekanizmalardaki açısal-doğrusal pozisyon, hız ve ivme kontrolünü hatasız bir şekilde yapan tahrik sistemi olarak tanımlanır. Yani hareket kontrolü yapılan bir düzenektir. Servo motorlar, robot teknolojilerinde en çok kullanılan motor çeşidi olmakla birlikte, RC (Radio Control) uygulamalarda da kullanılmaktadırlar. RC Servo Motorlar ilk olarak uzaktan kumandalı model araçlarda kullanılmışlardır. Servolar, istenilen pozisyonu alması ve yeni bir komut gelmediği sürece bulunduğu pozisyonu değiştirmemesi amacıyla tasarlanmıştır.  
  
Servo Motor Çalışma Prensibi  
Servo motorların içerisinde motorun hareketini sağlayan bir DC motor bulunmaktadır. Bu motorun dışında bir dişli mekanizması, potansiyometre ve bir motor sürücü devresi bulunmaktadır. Potansiyometre, motor milinin dönüş miktarını ölçmektedir. Servo içerisindeki DC motor hareket ettikçe potansiyometre döner ve kontrol devresi motorun bulunduğu pozisyon ile istenilen pozisyonu karşılaştırarak motor sürme işlemi yapar. Yani, servolar diğer motorlar gibi harici bir motor sürücüye ihtiyaç duymadan çalışmaktadırlar. Genellikle çalışma açıları 180 derece ile sınırlıdır fakat 360 derece çalışma açısına sahip özel amaçlı servo motorlar da vardır. Servolar genellikle 4.8-6V gerilim ile çalışmaktadırlar. 7.4V ve daha yüksek gerilimle çalışan servolar da bulunmaktadır.**

**3.Arduino UNO R3**

****

**Özellikler**

|  |  |
| --- | --- |
| Mikrodenetleyici | AT Mega 328 |
| Çalışma Gerilimi | 9V |
| Giriş Gerilimi(önerilen) | 7-12V |
| Dijital I/O Pinleri | * 14 (6 tanesi PWM çıkışı) |
| Analog Giriş Pinleri | 6 |
| Her I/O için Akım | 40mA |
| * 3.3V Çıkış için Akım | * 50 mA |
| EEPROM | * 1 KB (ATmega328) |
| SRAM | * 2KB (ATmega328) |
| Flash Hafıza | * 32KB (Atmega32bootloader) |
| Uzunluk | * 68.8 mm |
| Genişlik | 53.4mm |
| Ağırlık | * 25gr |

Arduino Uno gücünü usb üzerinden veya harici güç kaynağından alabilir. Harici güç kaynağı AC-DC adaptör olabileceği gibi bataryada olabilir. Adaptör kart üzerindeki 2.1mm merkez-pozitif güç soketinden bağlanabilir. Batarya kart üzerindeki GND ve Vin pinleri üzerinden bağlanabilir.

Kartın çalışması için sürekli olarak usb'nin bağlı olması şart değildir. Kart sadece adaptör veya batarya ile çalıştırılabilir. Bu sayede kart bilgisayardan bağımsız olarak çalıştırılabilir.

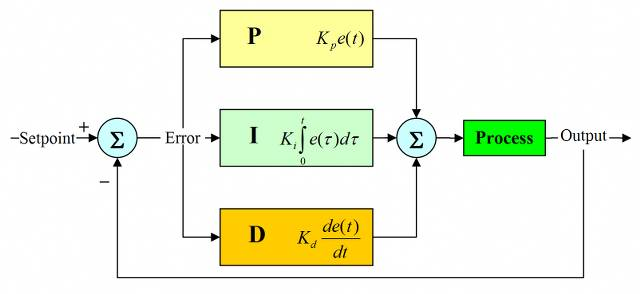
Harici güç kaynağı olarak 6-20V arası kullanılabilir. Ancak bu değerler limit değerleridir. Kart için önerilen harici besleme 7-12V arasıdır. Çünkü kart üzerinde bulunan regülatör 7V altındaki değerlerde stabil çalışmayabilir. 12V üstündeki değerlerde de aşırı ısınabilir.

Uno kartının üzerindeki mikrodenetleyicinin çalışma gerilimi 5V'dur.

Güç pinleri:

* **VIN:** Harici güç kaynağı kullanılırken 7-12V arası gerilim giriş pini.
* **5V:** Bu pin regülatörden çıkan 5V çıkışı verir. Eğer kart sadece usb (5V) üzerinden çalışıyor ise usb üzerinden gelen 5V doğrudan bu pin üzerinden çıkış olarak verilir. Eğer karta güç Vin (7-12V) veya güç soketi (7-12V) üzerinden veriliyorsa regülatörden çıkan 5V doğrudan bu pin üzerinden çıkış olarak verilir.
* **3V3:** Kart üzerinde bulunan 3.3V regülatörü çıkış pinidir. Maks. 50mA çıkış verebilir.
* **GND:** Toprak pinleridir.

**PID**

****

**Şekil 9: PID kontrol Diyagramı**

PID,Oransal İntegral Türev için kullanılan bir kısaltmadır.En genel tanımıyla bir kontrol geri bildirim mekanizmasıdır.PID yönteminin en temel amacı hatayı minimize etmek,en aza indirmektir.PID kontrolünü uygulamak ve kavramak oldukça zordur. PID kontrolünde öncelikle hata tanımlaması yapılmalıdır. Hata ise referans değere olan uzaklık olarak tanımlanabilir. Açıklamak gerekirse;

Referans=İstenilen değer

Gelen=Şuan ki Konum

HATA=Referans-Gelen

**Oransal Terim(P):**

Oransal terim, sistemden gelen hatayı bir katsayı ile çarparak hatayı küçültmeyi hedefler. Bozucu etkileri de mevcuttur. Projemizin yazılım kısmında oldukça ağırlık verdiğimiz PID kontrolünde deneme ve test aşamasında bu katsayıya büyük bir değer vermemiz gerektiğini anladık.

P=Kp\*HATA

**Integral Terimi(I):**

Integral hatanın alanını bulmak anlamına gelir. Integralin çok yükselmesini önlemek için sınırlandırmak gereklidir. Sürekli toplandığı için integral çok artarsa tekrar azalmasını beklemek zaman alır. Bu yüzden integrali sınırlamak sistemin çabuk toparlamasını sağlayacaktır.

I = I + (Ki \* HATA \* dt)

dt: PID fonksiyonuna her girdiğinde geçen zaman.

**Türev Terimi(D):**

Türev sistemdeki iki örnek arasındaki zamanı hesaplar.Eğer hatada bir değişim olmadıysa türev sıfır olur.

EHata=Bir önceki hatanın değeri

HD=HATA-EHata

D=(Kd\*HD)/dt

**PID Algoritması**

Kp, Ki ve Kd katsayılarından oluşur. Bu katsayılar deneme yanılma yöntemiyle bulunur. Yapacağınız sistemde optimum katsayıları bulmak için değerde değişiliklik yapıp sistemi gözlemlemeniz gerekmektedir.

HATA = Referans - Gelen

HD = HATA - EHata

P = Kp \* Hata

I = I + (Ki \* HATA\* dt)

D = (Kd \* HD)/dt

PID = P + I + D

EHata = Hata

**4.Yazılım Yapısı**

**a.Algoritmik olarak:**

**RFID nin dogru kartın okutulmasıyla doğru değerlerin oluşması ve ışık yanması ve buna bağlı olarak kapının açılması sonra Arduino ya bağlanmış olan mesafe sensörü nün mesafeye göre yani belirlenen değerler oluşunca ledlerin yanması.**

**b:Kod yapısı:**

#include <Servo.h>

#include <SPI.h>

#include <MFRC522.h>

#include <EEPROM.h>

#define RST\_PIN 9

#define SS\_PIN 10

MFRC522 mfrc522(SS\_PIN, RST\_PIN);

String lastRfid = "";

String kart1 = "";

String kart2 = "";

MFRC522::MIFARE\_Key key;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*buzzer\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

int buzzerPin = 8 ;

int notaSayisi = 8;

int C = 262;

int D = 294;

int E = 330;

int F = 349;

int G = 392;

int A = 440;

int B = 494;

int C\_ = 523;

int notalar[] = {C, D, E, F, G, A, B, C\_};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*servo\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

Servo sg90;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*mesafe\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

#define trigPin 4 //Sensörün Echo pini Arduinonun 4. pinine bağlanır

#define echoPin 5 //Sensorün Trig pini Arduinonun 5. pinine bağlanır

void setup()

{

Serial.begin(9600);

SPI.begin();

mfrc522.PCD\_Init();

pinMode(buzzerPin, OUTPUT);

Serial.println("RFID KART OKUMA UYGULAMASI");

Serial.println("--------------------------");

Serial.println();

//EEPROM'dan kart bilgisini oku

readEEPROM();

{

sg90.attach(6);

}

{

pinMode(trigPin, OUTPUT); //4. yani trigpini çıkış olarak ayarlıyoruz

pinMode(echoPin, INPUT); //5. yani echoPini giriş olarak ayarlıyoruz

Serial.begin(9600);

pinMode(3, OUTPUT);

pinMode(2 , OUTPUT);

}

}

void loop()

{

//yeni kart okununmadıkça devam etme

if ( ! mfrc522.PICC\_IsNewCardPresent())

{

return;

}

if ( ! mfrc522.PICC\_ReadCardSerial())

{

return;

}

//kartın UID'sini oku, rfid isimli string'e kaydet

String rfid = "";

for (byte i = 0; i < mfrc522.uid.size; i++)

{

rfid += mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ";

rfid += String(mfrc522.uid.uidByte[i], HEX);

}

//string'in boyutunu ayarla ve tamamını büyük harfe çevir

rfid.trim();

rfid.toUpperCase();

if (rfid == lastRfid)

return;

lastRfid = rfid;

Serial.print("Kart 1: ");

Serial.println(kart1);

Serial.print("Kart 2: ");

Serial.println(kart2);

Serial.print("Okunan: ");

Serial.println(rfid);

Serial.println();

//1 nolu kart okunduysa LED'i yak, 2 nolu kart okunduysa Buzzer'ı sustur.

if (rfid == kart1)

{

digitalWrite(buzzerPin, HIGH);

delay(100);

digitalWrite(buzzerPin, LOW);

Serial.println("Giris Basarılı!");

}

if (rfid == kart1)

{

sg90.write(85);

delay(2000);

sg90.write(5);

Serial.println("Kapı açıldı.");

}

if (rfid == kart1)

{

long duration, distance;

digitalWrite(trigPin, HIGH);

delayMicroseconds(1000);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = (duration / 2) / 29.1; //Ölçüm fonksiyonu

Serial.println(distance);

if (distance <= 5)

{

digitalWrite(3, HIGH);

digitalWrite(2, LOW);

delay(3000);

digitalWrite(3, LOW);

digitalWrite(2, LOW);

}

if (distance > 5 && distance < 200)

{

digitalWrite(2, HIGH);

digitalWrite(3, LOW);

delay(3000);

digitalWrite(2, LOW);

digitalWrite(3, LOW);

}

else

{

digitalWrite(22, LOW);

digitalWrite(23, LOW);

}

}

if (rfid == kart2)

{

Serial.println("Gecersiz Kart!");

digitalWrite(buzzerPin, HIGH);

delay(500);

digitalWrite(buzzerPin, LOW);

}

Serial.println();

delay(200);

}

void readEEPROM()

{

//EEPROM'dan ilk kartın UID'sini oku (ilk 4 byte)

for (int i = 0 ; i < 4 ; i++)

{

kart1 += EEPROM.read(i) < 0x10 ? " 0" : " ";

kart1 += String(EEPROM.read(i), HEX);

}

//EEPROM'dan ikinci kartın UID'sini oku

for (int i = 4 ; i < 8 ; i++)

{

kart2 += EEPROM.read(i) < 0x10 ? " 0" : " ";

kart2 += String(EEPROM.read(i), HEX);

}

//Okunan stringleri düzenle

kart1.trim();

kart1.toUpperCase();

kart2.trim();

kart2.toUpperCase();

}

**SONUÇ**

**1.Bilgi Düzeyine Katkıları:**

1-Öncelikle bir proje gerçekleştirirken nelere dikkat etmemiz gerektiğini öğrendik.

2-projenin , geliştirilme sürecini öğrendik. Bu aşamada nelerin yapılmaması gerektiğini öğrendik.

3-Kodlama ve kod geliştirme becerilerimizi arttırdı.

4-Arduino ide ile arduino programlama ve arduino kod yazma mantığını öğrendik.

5--Kızılötesi sensör, servo motor ve arduino çalışma mantığını öğrenerek farklı devrelere uygulayabildik.

**2.Teknolojik Katkıları:**

Kodlama teknoloji günümüzde gelişen teknolojiyle birlikte endüstriyel alanda bir süredir kullanılmaktadır. Genellikle gelişen sanayinin oluşturduğu büyük sanayi kuruluşlarında lojistik ve otomasyon bölümleri içerisinde fazlaca gereksinim duyulan niteliksiz insan gücü ile yapılan taşıma işlemlerini bir süredir çizgi takip eden robotlar yapmaya başlamışlardır.

**3.Ekip Çalışması Katkıları:**

Ekip çalışmasının bize kattıkları:

1-Grup olarak nasıl çalışmamız gerektiğini öğrendik.

2Çalışanların birbirleriyle iletişimini arttırdı.

3-Bir problemi grup olarak nasıl çözmemiz gerektiğini öğrendik.

5-Problem çöze yeteneğimizi geliştirdi.

6-Görev dağılımı yaparak görevlerimiz kolaylaştı ve birbirimize yardım ederek dayanışma içinde olduk.

**4.Aksayan Yönler:**

Projeyi gerçekleştirirken karşılaştığımız sorunlar:

Devre tasarımının en sade hale getirilmesinde sorun yaşadık.

Teknik aksaklılar sonucunda kodumuz silindi.

**5.Görüş ve Öneriler:**

Bu proje grubumuzdakilerin teknik bilgisini geliştirdi ve bize proje yapma konusunda birçok bilgi kattı. Bu bilgiler sayesinde ilerde yapacağımız projelere katkısını olacağını düşünüyoruz.

Birçok kişi üniversite döneminde bunu yaparken bizim lisede yapmamız birçok yönden bize avantaj sağladı

***Github adresi:***