

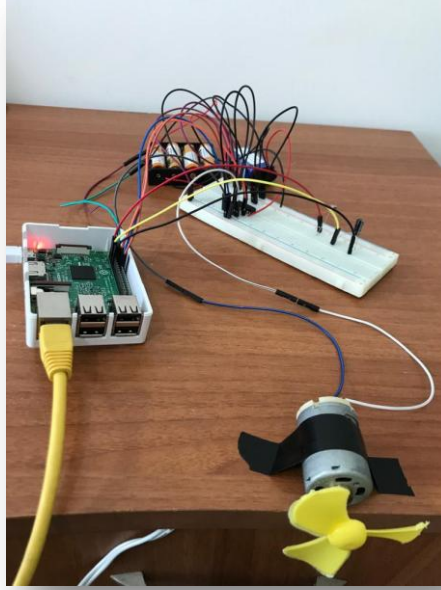


MARMARA
ÜNİVERSİTESİ

Teknoloji Fakültesi

Mekatronik Mühendisliği

Gömülü Sistemler Dersi Proje Ödevi



Rafet GÜVEN

170215012

Danışman:

Yrd. Doç. Hüseyin YÜCE

Raspberry Pi 3B ve LDR Sensörü ile DC Motor Kontrolü

Yaptığım tasarımda temel olarak Güneş ışığının şiddetinin fazla olduğu günlerde ışıktan maksimum seviyede yararlanmak adına perdeleri açarak zaman ve iş kaybının önüne geçilmesi düşünülmüştür. Sınıf, iş yeri gibi kalabalık ortamlar için tasarlanmıştır. Işık şiddeti çok ise DC motor çalışsın perde açılsın (motorun + yönde dönmesi), ışık şiddeti az ise DC motor çalışsın perde kapansın (motorun - yönde dönmesi). Böylece çalışanların zaman kayıplarının olmaması ve C vitamininden faydalanmaları amaçlanmıştır.

Projenin Genel Çalışma Prensibi

Kurduğum sistemdeki en önemli bileşenlerden biri olan LDR sensörü, ona seri bağlı olan kondansatörün dolma süresiyle birlikte çıkış vererek Raspberry Pi'a bilgi gönderir. Bu verileri sayısal bir çıktı olarak Raspberry Pi ekranından görebiliriz. Işık şiddeti yüksek ise okunan değer daha düşük (örn. 1325), eğer ışık şiddeti az ise okunan değer daha yüksektir (örn. 6854). LDR üzerinden geçen akımın kondansatörü doldurup boşaltma hızıyla doğrudan ilişkilidir. Bu aldığımız değerler Python kodunda IF bloğuna sokularak DC motorumuzun hangi seviyelerde çalışması veya durması gerektiği rahatça ayarlanabilir.

Benim projemde Raspberry Pi ve bilgisayarım arasındaki bağlantı fiziksel olarak Ethernet kablosuyla sağlanmıştır. Raspberry kontrolü buradan yapılarak harici klavye ve fare ihtiyaçları ortadan kaldırılmıştır. LDR'den alınan bilgiler doğrultusunda Güneş ışığının şiddetli olduğu değerler 0-2000 arasında, ışık şiddetinin normal olduğu ve herhangi bir eylemin gerektirmediği değerler 2000-5500 arasında, Güneş ışığının az olduğu ve içeride perde açılmasının gerekli olduğu durumlar 5500-7000 arasında belirlenmiştir. Bu durumlara göre motor kontrolü başarılı bir şekilde yapılmıştır.

IF bloğu 0-2000 arasında girdiği zaman, DC motor (+) yönde 2sn kadar döner ve otomatik şekilde durur.

IF bloğu 2000-5500 arasında girdiği zaman, motor herhangi bir eylem gerçekleştirmez ve stabil kalır.

IF bloğu 5500-7000 arasına girdiği zaman, motor (-) yönde 2sn kadar döner ve otomatik şekilde durur.

Kullanılan Donanım Bileşenleri:

- Raspberry Pi 3 Model B (1 adet)
- microSD Kart (1 adet)
- 12v DC Motor (1 adet)
- Pervane (1 adet)
- LDR (1 adet)
- 47 μ F Kondansatör (1 adet)
- L293D Motor Sürücü Entegre (1 adet)
- 12v DC kaynak (1 adet)
- Breadboard (1 adet)
- Jumper Kablo (20 adet)
- Ethernet Kablosu (1 adet)



Fotoğraf 1

Yararlanılan Kaynaklar:

- <https://github.com/marmara-mekatronik/mrm3006/blob/master/CC2019.ipynb>
- <https://gist.github.com/yasinkuyu/34ea9da052c55c4986f9d8e883310050>
- https://www.tutorialspoint.com/python/python_basic_syntax.htm
- <https://raspi.tv/2013/rpi-gpio-basics-4-setting-up-rpi-gpio-numbering-systems-and-inputs>
- <https://www.raspberrypi.org/forums/viewtopic.php?t=111353>
- <https://www.aranacorp.com/en/control-a-dc-motor-with-raspberry-pi/>

Kullanılan Bileşenler ve Özellikleri:

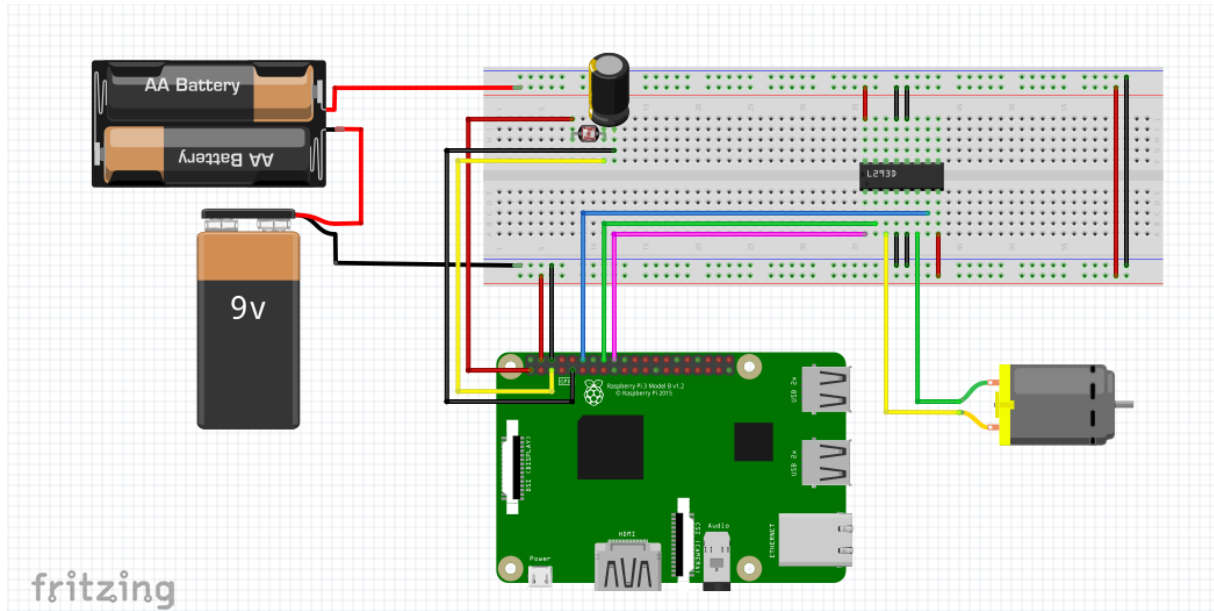
- Raspberry Pi: Tek bir boardtan oluşan mini bir bilgisayar olan Raspberry pi özgür bir yazılıma sahiptir. Raspberry pi daha çok gömülü sistemler projelerinde ve işletim sistemi uygulamalarında kullanılmaktadır. Projemde de Raspberry Pi 3 Model B kullanılmıştır.

- LDR: Işık şiddetini ölçmeye yarayan iki uçlu bir devre elemanıdır. Üzerine düşen ışık miktarına göre iki ucun arasındaki direnç değeri değişir. Bu özelliği sayesinde ışık miktarı ölçülebilir. Projemde de Raspberry Pi 3 Model B’de analog giriş mevcut olmadığı için bu duruma bir çözüm olarak bir RC devresi kurmamız gerekti.

LDR Ölçümü İçin RC Devresi: RC devresi, bir kondansatör ve dirençten oluşur. Kondansatör, elektrik devrelerinde yük depolama görevinde kullanılırlar. Akım, kondansatörü şarj ederek geriliminin yükselmesine sebep olur. Direnç ise kondansatöre giden akımı sınırlandırarak, kondansatörün şarj olmasını yavaşlatır. Yani kondansatörün dolacağı süreyi, direncin değeri belirler. LDR’nin direnci üzerine düşen ışığa göre değişeceği için, kondansatörün şarj olma süresini ışık miktarı belirleyecektir.

- DC Motor: Işık şiddetinin bilgisine göre ileri geri hareket eder. 12v DC kaynak ile çalışır.
- DC Motor Entegresi (L293D): Motor sürücü olarak kullanılır, motorun ileri geri hareketini sağlar.
- Kondansatör: Kondansatörler elektrik yüklerini kısa süreliğine depo etmeye yarayan devre elemanlarıdır. Projemde, LDR'den alınacak analog sinyali dijitalle çevirmeye yaramıştır.
- Jumper Kablo: Bileşenler arasında fiziksel bağlantıyı sağlamak için kullanılmıştır.
- Board: Devrenin üzerine kurulduğu iletken.
- 12v DC kaynak: DC motoru çalıştırmak için kullanılmıştır.

Devre Şeması



Şekil 5 – Devre Şeması

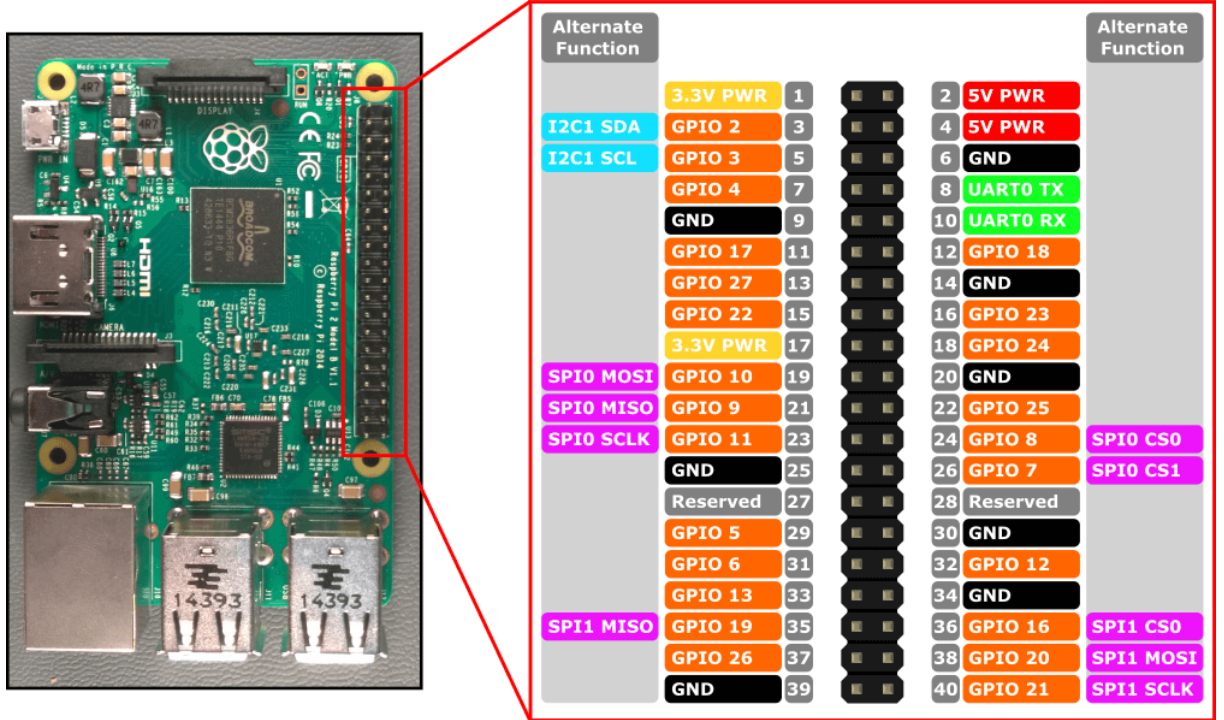
Projenin Yapım Aşamaları

1. Board kurulumu

Board kurulumu yaparken üstte gösterilen şematik gösterim dikkate alınır.

Gerekli bağlantılar erkek-erkek, dişi-erkek jumper kablolar yardımıyla yapılır.

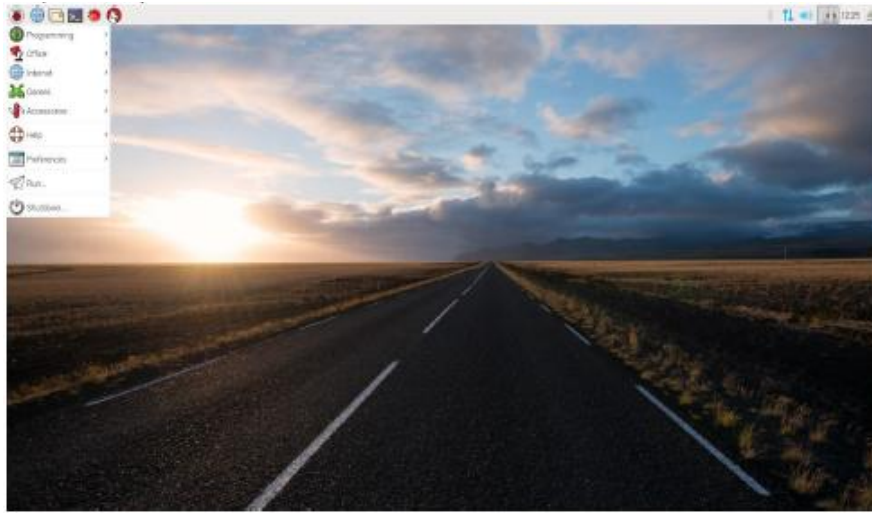
Kurulum yaparken kabloların nereye bağlanacağı çok önemlidir çünkü kodlar buna göre yazılmıştır ve sistemin istenilen şekilde çalışmasını engelleyebilir.



Şekil 1 - Raspberry Pi 3 Model B Pin Düzeni

2. Raspberry Pi 3 Model B Kurulumu

1. Raspberry Pi yazılımını SD karta yüklemek için önce SD kartın ‘SD Card Formatter’ programı ile biçimlendirilmesi gerekir.
(<https://www.sdcard.org/downloads/formatter/>)
2. Raspberry Pi işletim sistemi (Raspbian) indirilir.
(<https://www.raspberrypi.org/downloads/raspbian/>)
3. İndirilen işletim sistemi ‘Win32 Disk Imager’ programı ile SD karta yazılır. (<https://sourceforge.net/projects/win32diskimager/>)
4. Yükleme işleminden sonra SD kart Raspberry Pi’ya takılır ve sistem başlatılır.
5. Harici ekran, klavye ve fare bağlantılarından sonra Raspbian arayüzüyle karşılaşılır.



Şekil 2 - Raspbian Arayüzü

3. RPI.GPIO Kütüphanesinin Tanımlanması

Yapılan projeye alakalı kullanılacak kodlar için kütüphane tanımlama işlemi arayüzde bulunan terminal ekranından yapılır.

İnternet yardımıyla yüklemeler gerçekleştirilir.

RPI.GPIO kütüphanesiyle pin giriş-çıkış ayarlamaları yapılır.

pi@raspberrypi: ~

```
pi@raspberrypi:~ $ sudo apt-get install python-dev python-rpi.gpio
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-dev is already the newest version (2.7.13-2).
python-rpi.gpio is already the newest version (0.6.3~stretch-1).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
```

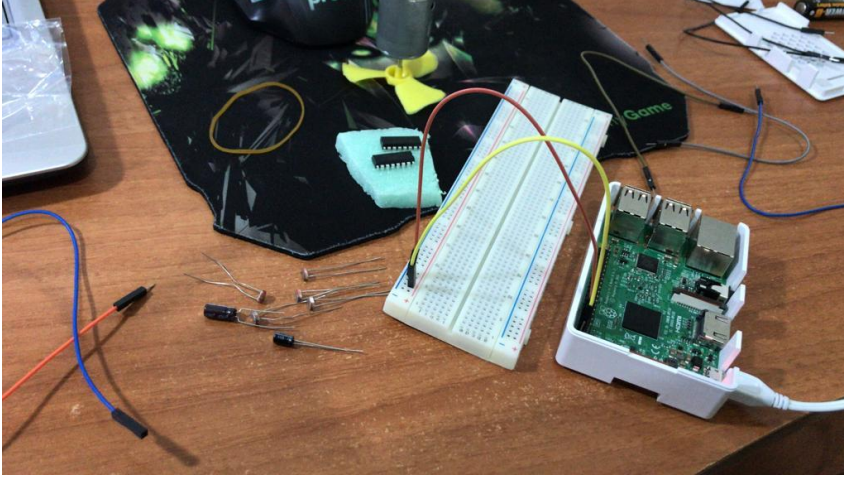
Şekil 3 - Terminalde RPI.GPIO Kütüphanesinin Yüklenmesi

Python Kodu

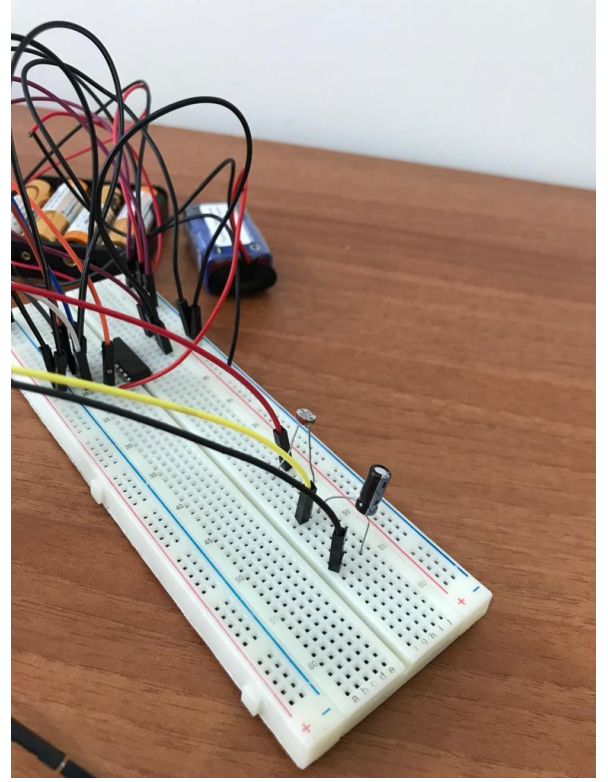
Kaynak koduna (<https://github.com/rafetguven/Project1>) adresinden de ulaşabilirsiniz.

```
1 import time
2 import RPi.GPIO as GPIO
3 from time import sleep
4
5 GPIO.setwarnings(False)
6 GPIO.setmode(GPIO.BCM)
7
8 ldr_pin = 3
9
10 def Rctime (RCpin):
11     reading = 0
12     GPIO.setup(RCpin, GPIO.OUT)
13     GPIO.output(RCpin, GPIO.LOW)
14     time.sleep(.1)
15
16     GPIO.setup(RCpin, GPIO.IN)
17     while (GPIO.input(RCpin) == GPIO.LOW):
18         reading += 1
19     return reading
20     GPIO.cleanup()
21 while True:
22     LDRReading = Rctime(3)
23     if 2000>LDRReading:
24
25         Motor1E = 18
26         Motor1A = 23
27         Motor1B = 24
28         print LDRReading
29         GPIO.setup(Motor1A,GPIO.OUT)
30         GPIO.setup(Motor1B,GPIO.OUT)
31         GPIO.setup(Motor1E,GPIO.OUT)
32         GPIO.output(Motor1A,GPIO.LOW)
33         GPIO.output(Motor1B,GPIO.HIGH)
34         GPIO.output(Motor1E,GPIO.HIGH)
35         sleep(2)
36
37     if 4000<LDRReading<5500:
38
39         Motor1E = 18
40         Motor1A = 23
41         print LDRReading
42         GPIO.setup(Motor1E,GPIO.OUT)
43         GPIO.output(Motor1E,GPIO.LOW)
44         GPIO.setup(Motor1A,GPIO.OUT)
45         GPIO.output(Motor1A,GPIO.LOW)
46
47     if 5500<LDRReading<7000:
48
49         Motor1E = 18
50         Motor1A = 23
51         Motor1B = 24
52         print LDRReading
53         GPIO.setup(Motor1A,GPIO.OUT)
54         GPIO.setup(Motor1B,GPIO.OUT)
55         GPIO.setup(Motor1E,GPIO.OUT)
56         GPIO.output(Motor1A,GPIO.HIGH)
57         GPIO.output(Motor1B,GPIO.HIGH)
58         GPIO.output(Motor1E,GPIO.LOW)
59         sleep(2)
60         GPIO.output(Motor1A,GPIO.LOW)
61
62     time.sleep(1)
```

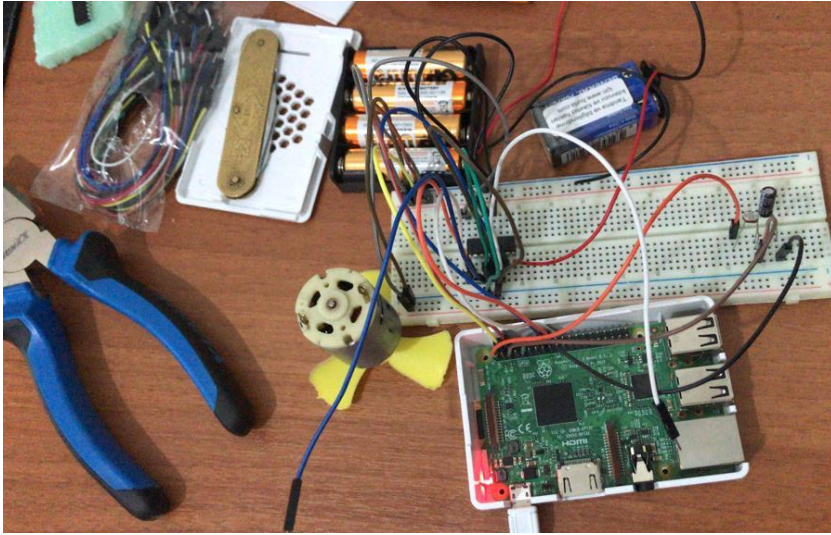

Proje Fotoğrafları



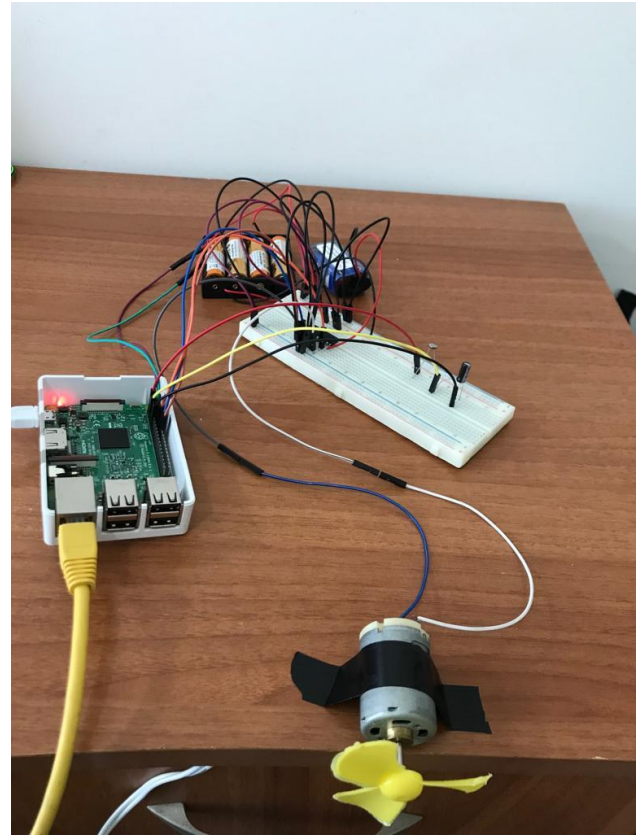
Fotoğraf 2



Fotoğraf 3



Fotoğraf 4



Fotoğraf 5 – Projenin Son Hali