



Marmara Üniversitesi
Teknoloji Fakültesi
Mekatronik Mühendisliği Bölümü
Gömülü Sistemler Dersi Proje Ödevi

Danışman:
Yrd. Doç. Hüseyin Yüce

Hazırlayanlar :
Mert Alkan 170214030
Envar Rashidov 170214902

Raspberry Pi İle Sıcaklık Ve Nem Optimizasyonu

Yaptığımız sistem temel olarak ortamdaki sıcaklık ve nemi stabilize eden ve belirlenen değerlerin dışına çıkılmamasına yöneliktir. Seralarda sıcaklık veya nemin çok düşük olması bitki büyümesini ve gelişimini geriletir, nemin veya sıcaklığın çok yüksek olması ise bitkilerin tozlanması(üreme) üzerinde olumsuz etki yapar. Ortaya sunduğumuz bu tasarım bu problemi ortadan kaldıracak niteliktedir.

Sistemimizin en önemli bileşenlerinden olan sıcaklık ve nem sensörümüz ortamdaki nemi ve sıcaklığı algılayıp dataları Raspberry pi ye gönderir. Daha önceden yazdığımız ve Raspberry pi ye gömdüğümüz Python kodu sayesinde sensör den alınan datalar işlenir ve “stabil sıcaklık, ısıtma sistemi devrede, soğutma sistemi devrede, stabil nem, nem artırma devrede, nem kurutma devrede” şeklinde çıktılar elde edilir.

Bu çıktıların aktif olma durumuna göre sistemimizde bulunan cihazlar devreye girip ortamın sıcaklık ve nem değerinin stabil bir aralıkta olmasını sağlar. Cihazların çalışması dc motor ve led ler ile gösterilir.

Dc motor sıcaklığı, 3 led ise nem durumunu temsil etmektedir.

Bu şekilde kullanıcı rahat bir şekilde seranın durumunu öğrenir ve buna uygun olarak hareket eder.

Sistem kurulurken doğru yerlere sensörlerin yerleştirilmesi çok önemlidir. Bu sensörlerden gelen bilgiler doğrultusunda uygun aralıklar belirlenip çalışma durumu ortaya çıkar.

- **Gerekli donanım bileşenleri**

1. 1 adet Raspberry Pi
2. 1 adet DHT_11 sensor
3. 2 adet direnç 220 ohm
4. 8 adet erkek-erkek kablo
5. 11 adet erkek-dişi kablo
6. 4 adet led
7. 1 adet dc motor
8. 1 adet L293 motor sürücü entegresi

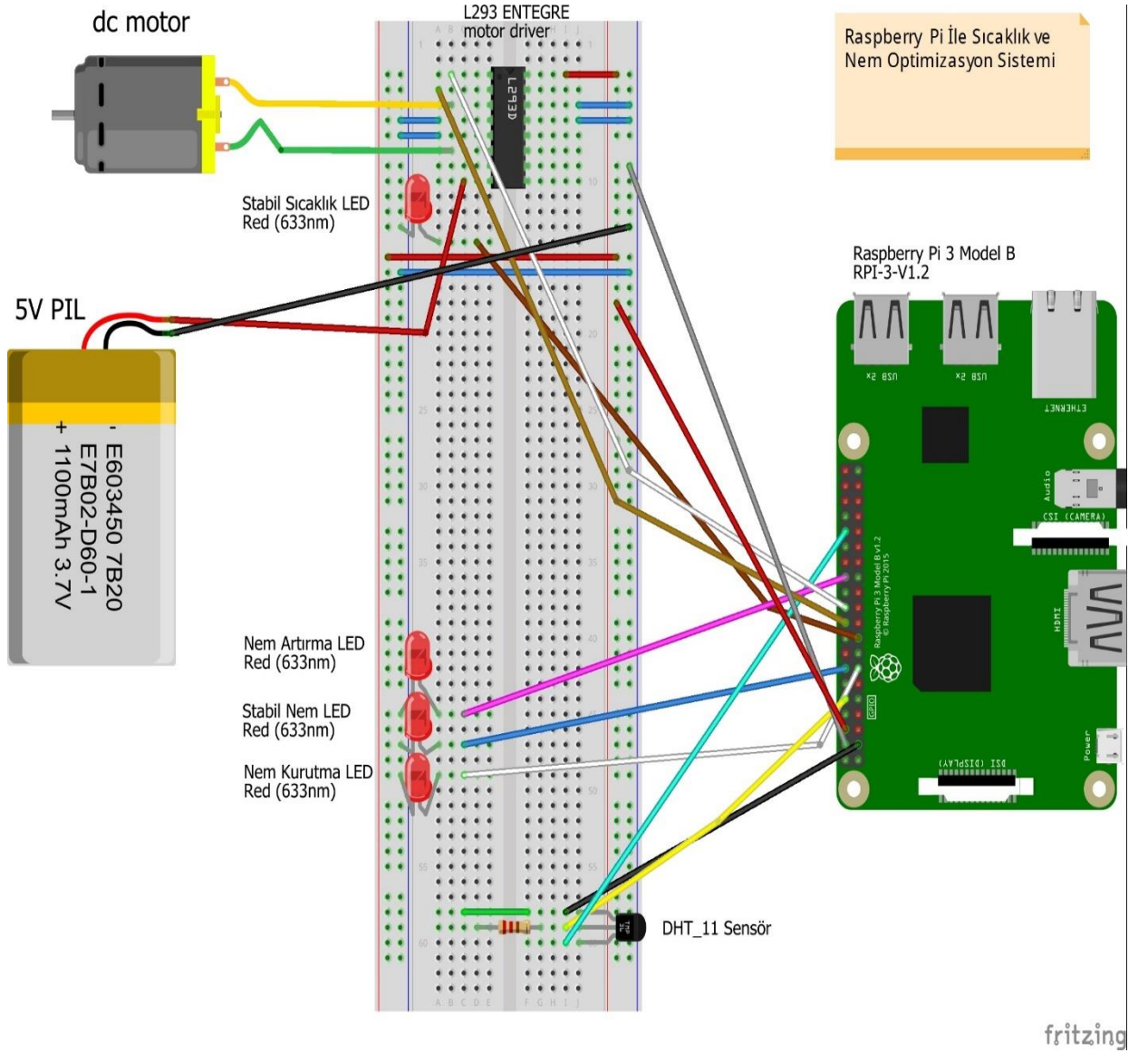
- **Yararlanılan Yazılım Bileşenleri**

1. RASPBIAN STRETCH LITE işletim sistemi
(<https://www.raspberrypi.org/downloads/raspbian/>)
2. Githup (https://github.com/szazo/DHT11_Python)
3. Pycharm (<https://www.jetbrains.com/pycharm/>)
4. Matplotlib Tutorial
(<https://www.youtube.com/watch?v=UlygQI2eSdg&list=PLQVvva0QuDfefDfXb9Yf0la1fPDKluPF&index=23>)
5. Python programming (<https://pythonprogramming.net/live-graphs-matplotlib-tutorial/>)
6. Fritzing (www.fritzing.org/)

- **Kullanılan fiziksel bileşenler ve özellikleri**

1. Raspberry pi 3 B modeli : Tek bir board tan oluşan mini bir bilgisayar olan Raspberry pi özgür bir yazılıma sahiptir. Raspberry pi daha çok gömülü sistemler projelerinde ve işletim sistemi uygulamalarında kullanılmaktadır. Ürün olarak Raspberry pi 3 model b kullanılmıştır.
2. DHT 11 Isı ve Nem Modülü: Sistemde kullanılan sıcaklık ve nem bilgisinin alındığı sensör. (<https://www.robotistan.com/dht11-isi-ve-nem-sensoru-kart>)
3. Dc Motor: Sıcaklık bilgisi ile ileri ve geri hareket eder, 5v pil ile çalışır. (<https://www.robotistan.com/motor>)
4. Dc Motor Entegresi (L293) : Motor sürücü görevini yapar, dc motorun hareketini sağlar.
5. Led Lamba: Nem değerlerini göstermek için kullanılır.
6. Jumper Kablo: gerekli bağlantıları yapmak için kullanılır. Erkek-erkek ve erkek dişi iki tipi vardır.
7. 5v Pil: Dc motoru çalıştırmak için kullanılır.
8. Board: Devrenin üzerine kurulduğu iletken.

- Şematik Çizim



Projenin Yapım Aşamaları

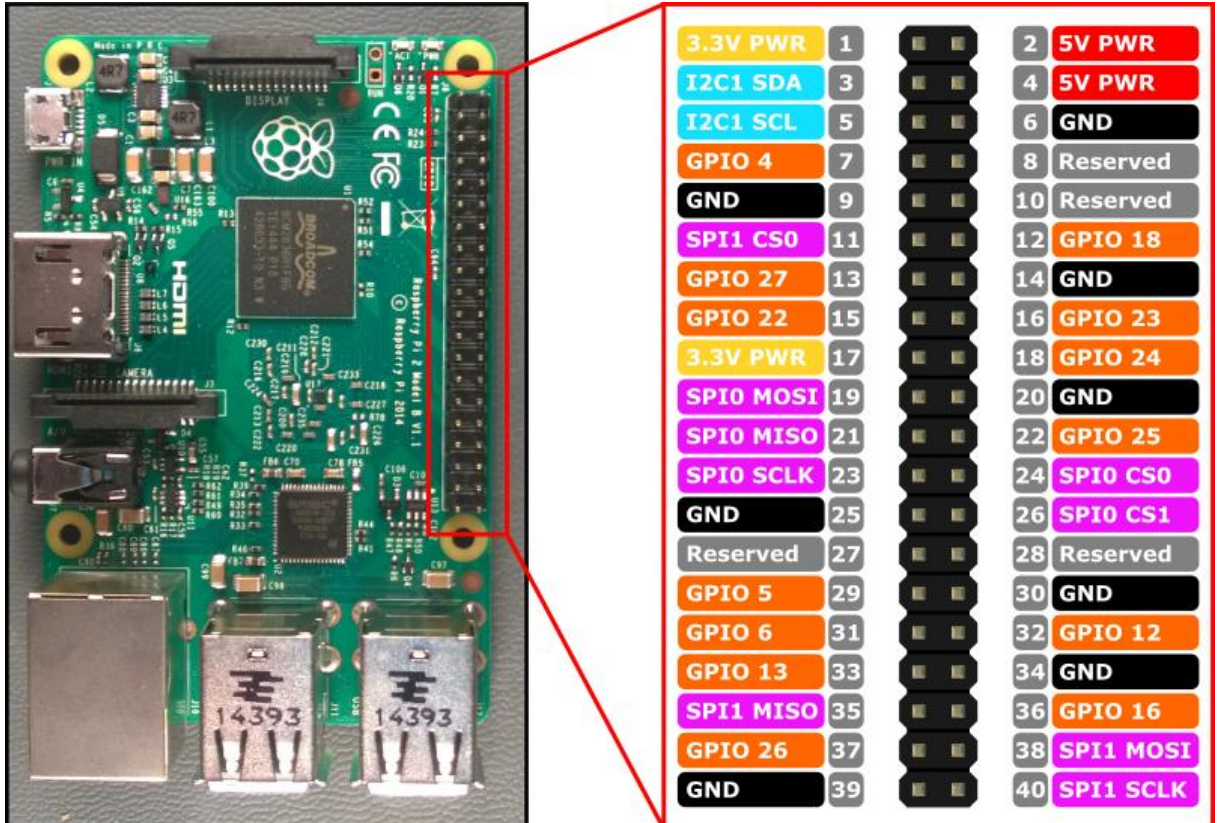
- **Board kurulumu:**

Board kurulumu yapılırken üstte gösterilen şematik şema dikkate alınır. Gerekli bağlantılar Raspberry pi pinlerinden yapılır.

Gerekli bağlantıları erkek-erkek ve erkek-dişi jumper kablolar ile yapabiliriz.

Board kurulumu yapılırken kabloların nereye bağlanacağına dikkat etmemiz gerekir. Bu sistemin düzgün çalışması için önemlidir.

Raspberry Pi 3 Model B Pin Düzeni



- **Raspberry Pi 3 Model B Kurulumu:**

(<https://www.raspberrypi.org/downloads/raspbian/>)

1.Öncelikle bu linkten indirdiğimiz Raspberry pi yazılımı sd karta yüklenir.

2.Bunun için önce sd kartın SD Card Formatter

(<http://bit.ly/1aIC3Wp>) programı ile biçimlendirilmesi gerekir.

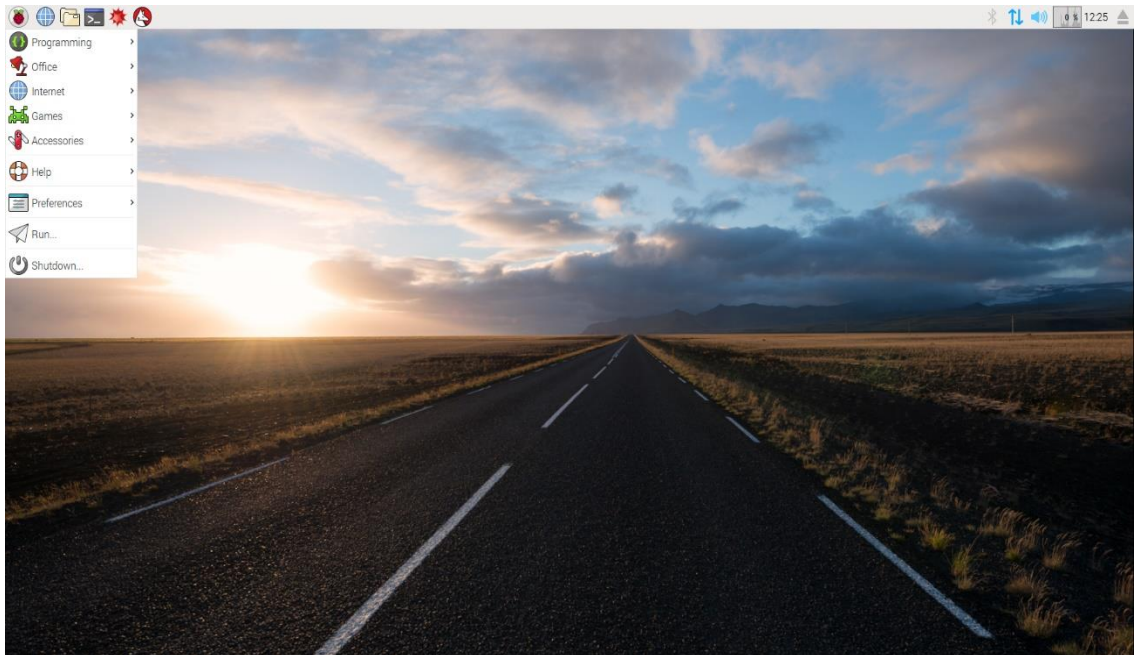
3.Bu işlemten sonra indirdiğimiz Raspberry pi yazılımı

Win32 Disk Imager: (<http://bit.ly/1UF6LFU>) programı ile sd karta yazılır.

4.Bu işlemlerden sonra sd kart Raspberry pi deki girişine takılarak Raspberry pi ye güç verilir.

5.Ekran bağlantılarını da sağladıktan sonra Raspbian arayüzü karşımıza çıkar.

Raspbian Arayüzü

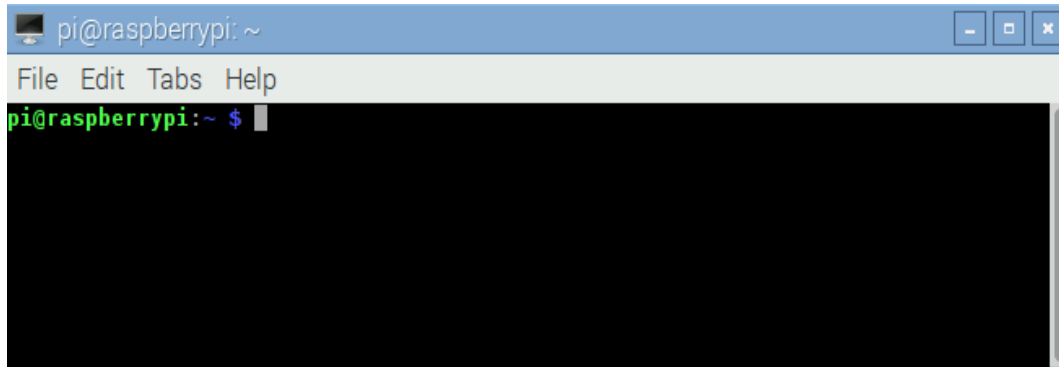


- **Kütüphanelerin Tanımlanması**

Kullanılacak olan kod için kütüphane tanımlama işlemi yapılırken arayüzden terminal kısmına giriş yapılır.

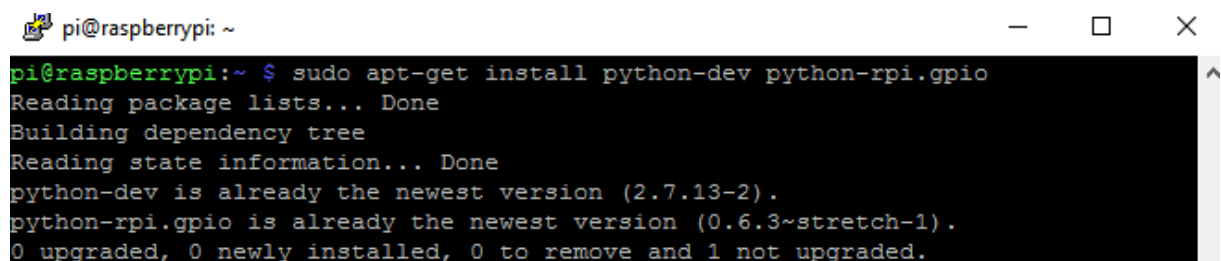
Buradan internet vasıtasıyla kütüphane yüklemeleri gerçekleştirilir.

Bu işlemlerin ardından arayüz vasıtasıyla kod yazımına başlanır.



```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $
```

- **RPI.GPIO kütüphanesinin tanımlanması**



```
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.
```

Bu kütüphane ile pin giriş-çıkışlarındaki ayarlamalar yapılabilir.

- **DHT_11 kütüphanesinin yüklenmesi**

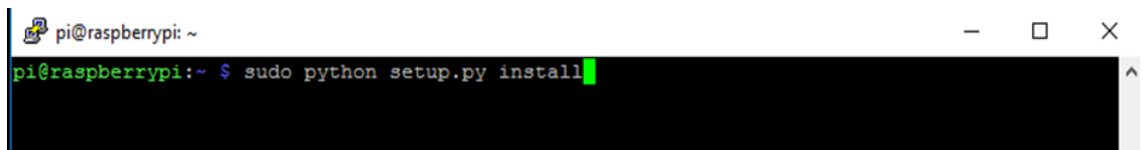
DHT_11 kütüphanesini yüklerken Github sitesinden (https://github.com/szazo/DHT11_Python) direkt olarak indirme işlemi yapılabilir. İndirilen dosya içinde kod yazmaya başlanır.

- Ayrıca dht_11 kütüphanesini;



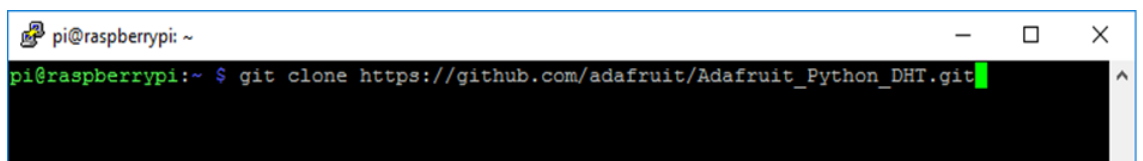
```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ sudo apt-get update
```

Komutu ile Raspberry pi kütüphane yükseltme,



```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ sudo python setup.py install
```

Komutu Raspberry pi kütüphane kurulumu,



```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ git clone https://github.com/adafruit/Adafruit_Python_DHT.git
```

Komutu ile dht_11 kütüphane kurulumu yapılabilir.

- Matplotlib kütüphanesinin yüklenmesi:

```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ sudo apt-get install python-matplotlib  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
python-matplotlib is already the newest version (2.0.0+dfsg1-2).  
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
```

Kodu ile Matplotlib kütüphanesi kurulumu gerçekleşir.

```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ pip install matplotlib  
Requirement already satisfied: matplotlib in /usr/local/lib/python2.7/dist-packages  
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python2.7/d  
ist-packages (from matplotlib)  
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/l  
ocal/lib/python2.7/dist-packages (from matplotlib)  
Requirement already satisfied: numpy>=1.7.1 in /usr/lib/python2.7/dist-packages (  
from matplotlib)  
Requirement already satisfied: backports.functools-lru-cache in /usr/local/lib/py  
thon2.7/dist-packages (from matplotlib)  
Requirement already satisfied: six>=1.10 in /usr/lib/python2.7/dist-packages (fro  
m matplotlib)  
Requirement already satisfied: pytz in /usr/local/lib/python2.7/dist-packages (fr  
om matplotlib)  
Requirement already satisfied: subprocess32 in /usr/local/lib/python2.7/dist-pack  
ages (from matplotlib)  
Requirement already satisfied: cycycler>=0.10 in /usr/local/lib/python2.7/dist-pack  
ages (from matplotlib)  
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python2.7/dist  
-packages (from matplotlib)  
Requirement already satisfied: setuptools in /usr/lib/python2.7/dist-packages (fr  
om kiwisolver>=1.0.1->matplotlib)  
You are using pip version 9.0.3, however version 10.0.0 is available.  
You should consider upgrading via the 'pip install --upgrade pip' command.
```

Kodu ile Matplotlib kütüphanesinin pip kullanarak modül olarak kurulması.

- **Kaynak Kodu:**

Kaynak koduna;
(<https://github.com/mertalkan/python-proje>) adresinden de ulaşabilirsiniz.

- **Kod yazımı:**

Öncelikle DHT_11 kütüphanesi .lib uzantılı olmadığı için indirilen DHT11_Python klasörünün içine gidilir. Klasöre “home\pi\DHT11_Python” ismi ile ulaşılabilir.

Klasörün içine girildikten sonra “.py” uzantılı bir dosya açılır, dosya açıldıktan sonra en başta kütüphanelerin tanımlamaları yapılır.

“import” ifadesiyle tanımlama işlemi yapılır.

Kütüphane tanımlamaları bittikten sonra kod yazımına başlanabilir.

- **Kod-1 (Ana Kod)**

Bu kod ile sensor den alınan verinin motor ve LED olarak çıkışı sağlanır.

Ayrıca grafik için “.txt” uzantılı dosyalar oluşturulup içine veri yazılır.

```
1. import RPi.GPIO as GPIO          # Kütüphane tanımlama işlemleri yapılır.
2. import dht11
3. import time
4. import datetime
5. from time import sleep
6. i=0
7. #.txt dosyalarının içine sıralı bilgi yazmak için kullanılacak i ye 0 atanır.
8.
9.
10. GPIO.setwarnings(False)          # Pinlere gerekli komutlar verilir.
11. GPIO.setmode(GPIO.BOARD)
12. GPIO.cleanup()
13.
14.
15. Motorileri = 16                   # Motor için gerekli pin giriş değerleri verilir.
16. Motorgeri = 18                   # Motorileri motorun sağa, motorgeri motorun sola dönüşünü sağlar.
17. Motordur = 22                    # Motor dur ise motorun durdurulmasını sağlar.
18.
19. GPIO.setup(15,GPIO.OUT)
20. #Ledler için (GPIO.setup) kullanılarak gerekli pin kurulumları yapılır.
21. GPIO.setup(23,GPIO.OUT)
22. GPIO.setup(24,GPIO.OUT)
23. GPIO.setup(12,GPIO.OUT)
24. GPIO.setup(13,GPIO.OUT)
25. GPIO.setup(15,GPIO.OUT)
26. GPIO.setup(8,GPIO.OUT)
27.
28.
29.
30. GPIO.setup(Motorileri,GPIO.OUT)   #Üste tanımlanan motor pin degerleri için kurulum yapılır.
31. GPIO.setup(Motorgeri,GPIO.OUT)
32. GPIO.setup(Motordur,GPIO.OUT)
33.
34. sensoroku = dht11.DHT11(pin=8)
35. #Sensör oku degiskenine "dht11.DHT11" ile sensörden alınan bilgi verilir.
36.                                     #bu verinin alımı için 8 nolu pin tanımlanır.
37.
38. open("nem.txt", "w")
39. #grafik kodu için kullanılacak olan nem.txt dosyasının açılma işlemi yapılır.
40. open("sicaklik.txt", "w")
41. #grafik kodu için kullanılacak olan sıcaklık.txt dosyasının açılma işlemi yapılır.
42. with open("nem.txt", "a") as log:
43.     #nem.txt dosyasını log komutu ile yazma için uygun hale getirilir.
44.     with open("sicaklik.txt", "a") as log1:
```

```

45. #sıcaklık.txt dosyasını log1 komutu ile yazma için uygun hale getirilir.
46.
47.     while True:                                #while true ile sonsuz bir döngü açılır.
48.
49.         deger = sensoroku.read()                #sensor oku daki data deger degiskenine atanır
50.
51.
52.
53.         if deger.is_valid():
54.             #if döngüsü ile uygun şartlarda yapılacak olan işlemler yazılır.
55.
56.             print("Son veri okuma saati: " + str(datetime.datetime.now()))
57.
58.             print("Sicaklik degeri: %d C" % deger.temperature)
59. #sensörden alınan datanın yazılması sağlanır.
60.
61.             print("Nem degeri: %d %" % deger.humidity)
62.
63.
64.
65.             if(deger.humidity>80):
66. #nemin %80 den büyük olma durumunda ekrana bilgi yazdırılır.
67.
68.                 print("Nem kurutma devrede")
69.
70.                 GPIO.output(24, GPIO.HIGH)
71. #nemin %80 den büyük olma durumunda 24 nolu pin high olur diğerleri low
72.                 GPIO.output(12, GPIO.LOW)
73.                 GPIO.output(13, GPIO.LOW)
74.
75.
76.
77.             elif(deger.humidity<70):
78. #nemin %70 den küçük olma durumunda ekrana bilgi yazdırılır.
79.
80.                 print("Nem artirma devreye girdi!")
81.
82.                 GPIO.output(12, GPIO.LOW)
83. #nemin %80 den büyük olma durumunda 13 nolu led high olur. Diğerleri 0
84.                 GPIO.output(13, GPIO.HIGH)
85.                 GPIO.output(24, GPIO.LOW)
86.
87.
88.             else:
89. #else ile kalan durumlar için ekrana verilen bilgi yazdırılır.
90.
91.                 print("Nem degeri sisteme uygun!")
92.
93.                 GPIO.output(12, GPIO.HIGH)        #led durumları pin çıkışları ile ayarlanı
r.
94.                 GPIO.output(13, GPIO.LOW)
95.                 GPIO.output(24, GPIO.LOW)
96.
97.
98.
99.
100.
101.             if(deger.temperature>25):
102.                 #sıcaklık için nem ile aynı şekilde şart durumları yapılır.
103.
104.                 print("Sogutma sistemi devreye girdi!") #ekrana bilgi yazdırılır
105.
106.                 GPIO.output(Motorileri,GPIO.HIGH)
107.                 #motorun sola dönmesi için motorgeriye low degeri verir...

```

```

108.             GPIO.output(Motorgeri,GPIO.LOW)
109.         #...diğerleri ise high olur
110.             GPIO.output(Motordur,GPIO.HIGH)
111.             GPIO.output(15, GPIO.LOW)
112.         #stabil durumu belirten led low degeri alır.
113.
114.             elif(deger.temperature<24):             # elif ile deger kontrol
        edilir.
115.
116.             print("Isitma sistemi devreye girdi!") #bilgi ekrana yazdırılır
117.
118.             GPIO.output(Motorileri,GPIO.LOW)
119.         #motor ve led için pin çıkışlarına high ve low degerleri verilir.
120.             GPIO.output(Motorgeri,GPIO.HIGH)
121.             GPIO.output(Motordur,GPIO.HIGH)
122.             GPIO.output(15, GPIO.LOW)
123.
124.
125.         else:
126.
127.             print("Sicaklik degeri sisteme uygun!")
128.         #else ile kalan durumlar için bilgi ekrana yazdırılır.
129.
130.             GPIO.output(Motordur,GPIO.LOW)
131.         #else durumu için pin çıkışları duzenlenir
132.             GPIO.output(15, GPIO.HIGH)
133.         #motor durdurulurken led degeri high yapılır. led stabil sıcaklığı belirtir
134.
135.         global i
136.         #globalkomutu ile iki.txt uzantılıdosyanında içineaynı yerdengelen datayazılır
137.
138.         i=i+1
139.         # işlemi ile .txt uzantılı dosyanının içine yazılacak grafik verisi düzenli h
        ale getirilir.
140.         if(int(deger.humidity)!=0):
141.             # gelen datanın sıfırdanfarklıdegerolması sağlanır.grafik kararsızhale geliyor
142.             log.write("{}{}\n".format(str(i),str(deger.humidity)))
143.             #logwrite ile ilk .txt ye yazma işlemi yapılır.
144.             if(int(deger.temperature)!=0):
145.                 #gelen datanın sıfırdan farklıdegerolması sağlanır.grafik kararsızhale geliyor
146.                 log1.write("{}{}\n".format(str(i),str(deger.temperature)))
147.                 #log1write ile ikinci .txt ye yazma işlemi yapılır.
148.                 time.sleep(2)
149.                 #sonsuz döngü için 2 saniye bekle.Daha kısa sürelerde grafik kararsız oluyor.

```

- **Grafik çizimi:**

Grafik çizimi için hem nem hem de sıcaklık için iki ayrı kod kullanılır.

Kod-1 ile oluşturulan ve içine data yazılan “.txt” uzantılı dosyalar kullanılarak alınan veriler grafiğe dökülür.

Grafik çiziminde x eksenini saniye cinsinden y eksenini ise okunan değer cinsinden olur.

Kod-1 çalışmasıyla veriler “.txt” dosyalarına kayıt olmaya başlar.

Daha sonra kod durdurulduğu anda kayıt biter.

Grafik kodunun çalıştırılmasıyla ise grafik ekrana yansır.

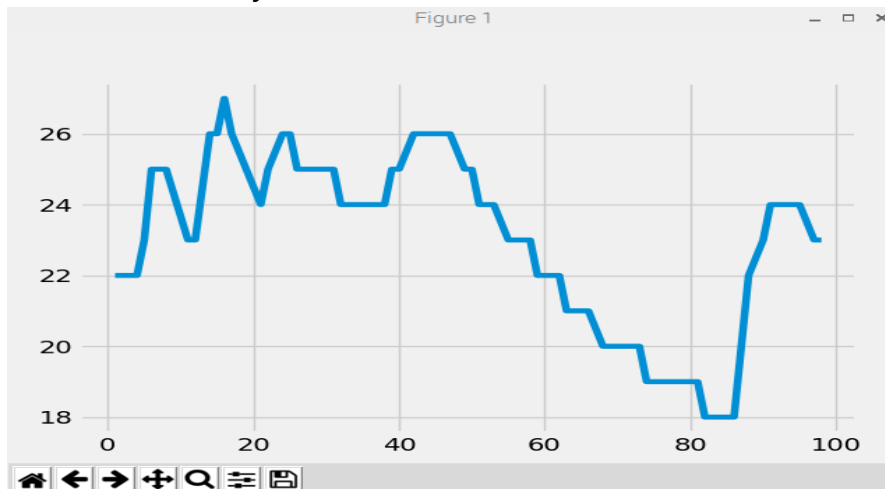
Kod-1 yeniden çalışması ile “.txt” uzantılı dosyaların içeri otomatik olarak silinir. Böylece her seferde istenilen bir grafik alınır.

- **Kod-2(SICAKLIK için Grafik)**

Kod-1 ile oluşturulan sıcaklik.txt uzantılı dosyadan veri okunarak grafik oluşturulması sağlanır.

```
1. def sıcaklıkoku():
2. #def komutu ile istediğim sıcaklıkoku() isminde fonsiyon tanımlarım.
3.
4.     import matplotlib.pyplot as plt
5.     #grafik ile ilgili kütüphane tanımlamaları yapılır.
6.     import matplotlib.animation as animation
7.     from matplotlib import style
8.
9.     style.use('fivethirtyeight')
10. #stle.use ile matplotlib için önceden tanımlanan stil tanımlanır.
11.
12.     fig = plt.figure()
13. #matplotlib için figur tanımlaması yapılır.grafik için kullanılacak
14.     ax1 = fig.add_subplot(1,1,1) #grafigin matematik mantığı için tanımlama yapılır
15.
16.     def animate(i): #anime adlı fonksiyon tanımlanır. i degeri içine verilir.
17.         graph_data = open('sıcaklik.txt','r').read()
18. #ana koddan alınan verilerin okunması sağlanır sıcaklık.txt ile.
19.         lines = graph_data.split('\n')
20.         xs = [] # x eksenini belirtir
21.         ys = [] # y eksenini belirtir.
22.         for line in lines: #grafığın sıra ile geçisi sağlanır.
23.             if len(line) > 1:
24.                 x, y = line.split(',')
25.                 xs.append(x) #degerler x ve y eksenlerine oturtulur.
26.                 ys.append(y)
27.         ax1.clear()
28.         ax1.plot(xs, ys) #eksenlere sıra ile degerler oturtulur.
29.         ani = animation.FuncAnimation(fig, animate, interval=10)
30.         plt.show() #grafigi ekrana ver komutu.
31. sıcaklıkoku() #sıcaklıkokudan alınan verilerle oluşturulan grafik ekrana verilir.
```

Sıcaklık-Zaman Grafik

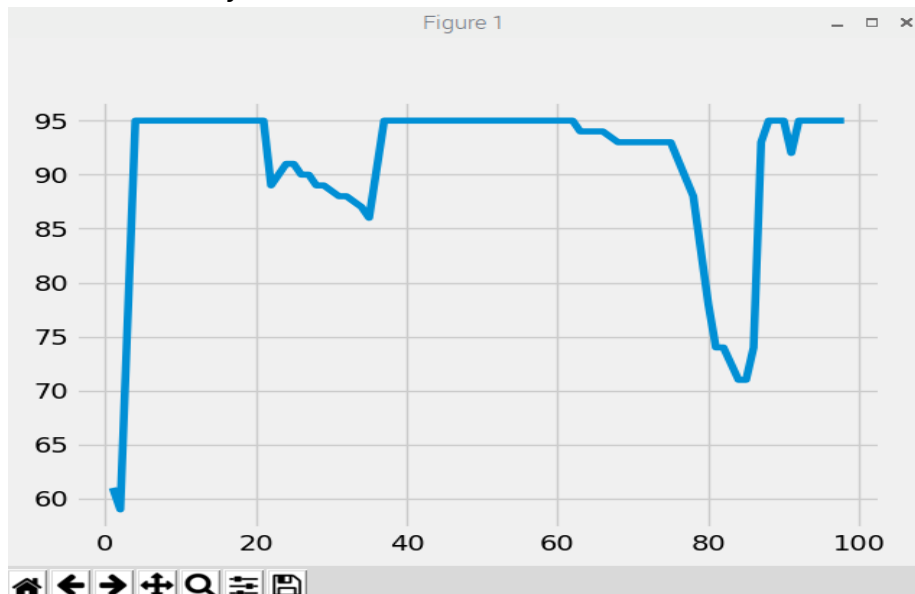


- **Kod-3(NEM için Grafik):**

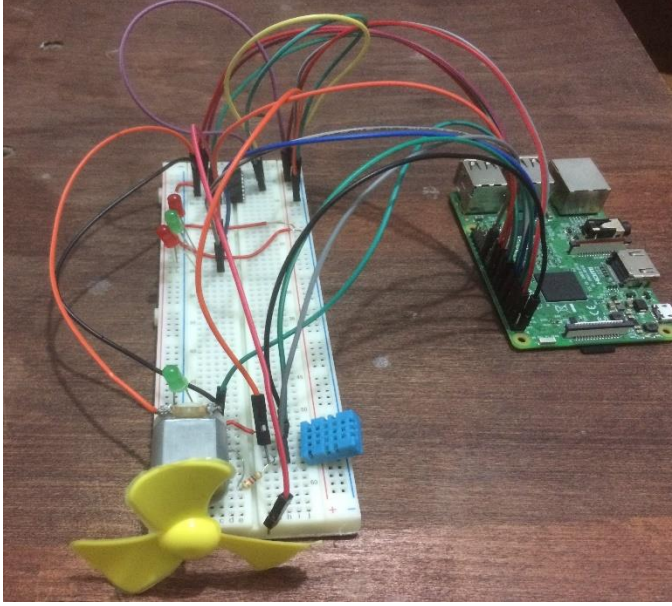
Kod-1 ile oluşturulan nem.txt uzantılı dosyadan veri okunarak grafik oluşturulması sağlanır.

```
1. def nemoku(): #def komutu ile istediğim nemoku() isminde fonsiyon tanımlarım.
2.     import matplotlib.pyplot as plt
3.     import matplotlib.animation as animation
4. #grafik ile ilgili kütüphane tanımlamaları yapılır.
5.     from matplotlib import style
6.
7.     style.use('fivethirtyeight')
8. #stle.use ile matplotlib için önceden tanımlanan stil tanımlanır.
9.
10.    fig = plt.figure()
11. #matplotlib için figur tanımlaması yapılır.grafik için kullanılacak
12.    ax1 = fig.add_subplot(1,1,1) #grafığın matematik mantığı için tanımlama yapılır.
13.
14.    def animate(i): #anime adlı fonksiyon tanımlanır. i degeri içine verilir.
15.        graph_data = open('nem.txt','r').read()
16. #ana koddan alınan verilerin okunması sağlanır nem.txt ile.
17.        lines = graph_data.split('\n')
18.        xs = [] # x eksenii belirtilir
19.        ys = [] # y eksenii belirtilir.
20.        for line in lines: #grafığın sıra ile geçisi sağlanır.
21.            if len(line) > 1:
22.                x, y = line.split(',')
23.                xs.append(x) #degerler x ve y eksenlerine oturtulur.
24.                ys.append(y)
25.        ax1.clear()
26.        ax1.plot(xs, ys) #eksenlere sıra ile degerler oturtulur.
27.        ani = animation.FuncAnimation(fig, animate, interval=10)
28.        plt.show() #grafigi ekrana ver komutu.
29. nemoku() #nemokudan alınan verilerle oluşturulan grafik ekrana verilir.
```

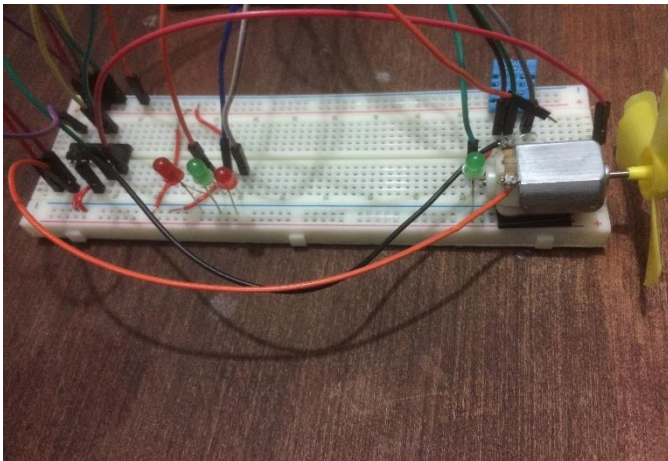
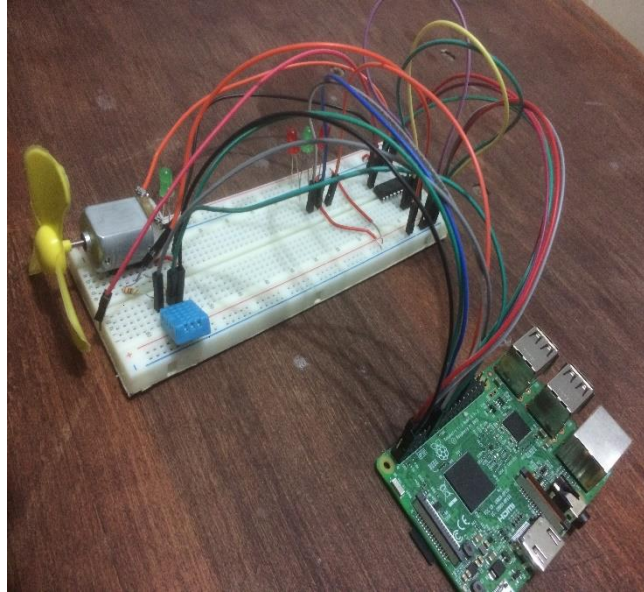
Nem-Zaman Grafik



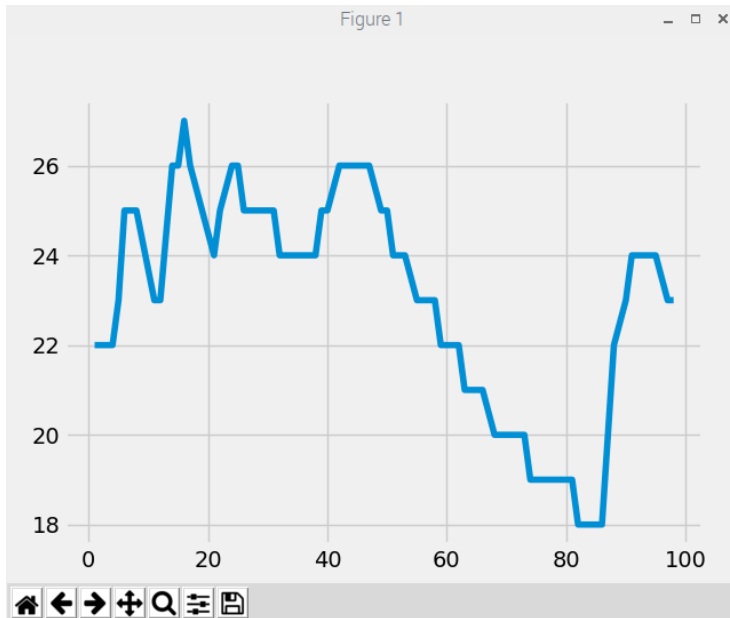
- Proje Resimleri:



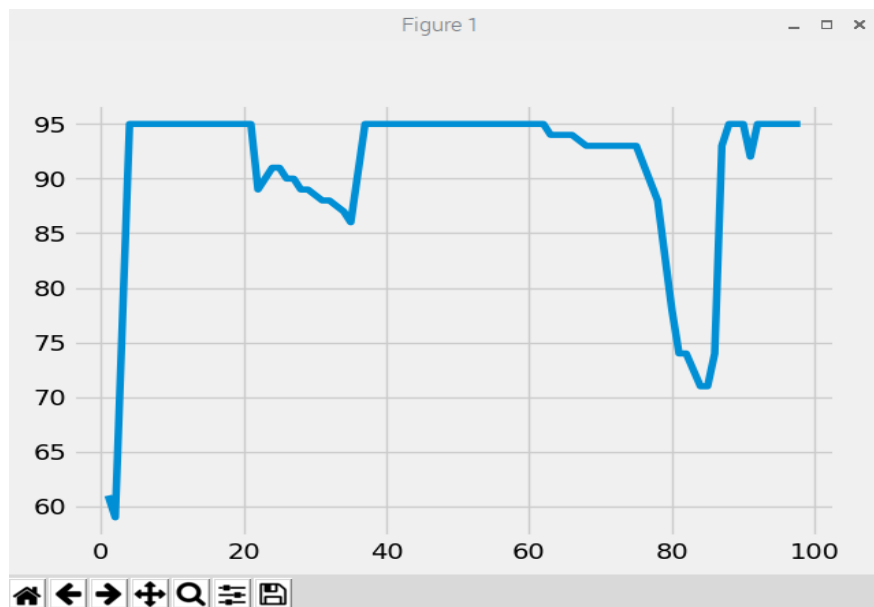
Raspberry
Pi ve
Board
Kurulumu



- **Grafik Resimleri**



Sıcaklık-Zaman Grafik



Nem-Zaman Grafik

