



Marmara Üniversitesi
Teknoloji Fakültesi
Mekatronik Mühendisliği Bölümü

Gömülü Sistemler Dersi
Final Proje Ödevi

Danışman:

Yrd. Doç. Dr. Hüseyin Yüce

Hazırlayanlar :

Mert Alkan 170214030

Envar Rashidov 170214902

CANLI NABIZ

(Nabız Sensörü İle Gerçek Zamanlı Grafik Uygulaması)

Yaptığımız sistem temel olarak Nabız sensöründen aldığımız nabız bilgisini gerçek zamanlı olarak grafiğinin çıkarılmasıdır.

Anemi, Ritim bozukluğu gibi hastalıklar için önemli olan nabız değerinin canlı olarak alınması ve ekrana grafik olarak verilmesi ile kullanıcı bunu kendisi takip eder hale gelecektir.

Sistemimizin en önemli bileşenlerinden olan nabız sensörümüz parmağtan aldığı dataları Raspberry pi ye gönderir. Daha önceden yazdığımız ve Raspberry pi ye gömdüğümüz Python kodu sayesinde sensör den alınan datalar işlenir ve gerçek zamanlı olarak grafik elde edilir.

- **Gerekli donanım bileşenleri**

1. 1 adet Raspberry Pi
2. 1 adet Nabız Darbe Sensoru
3. 1 adet Analog-Dijital çevirici (MCP3008 ADC)
4. 6 adet dişi-dişi kablo
5. 3 adet erkek-dişi kablo

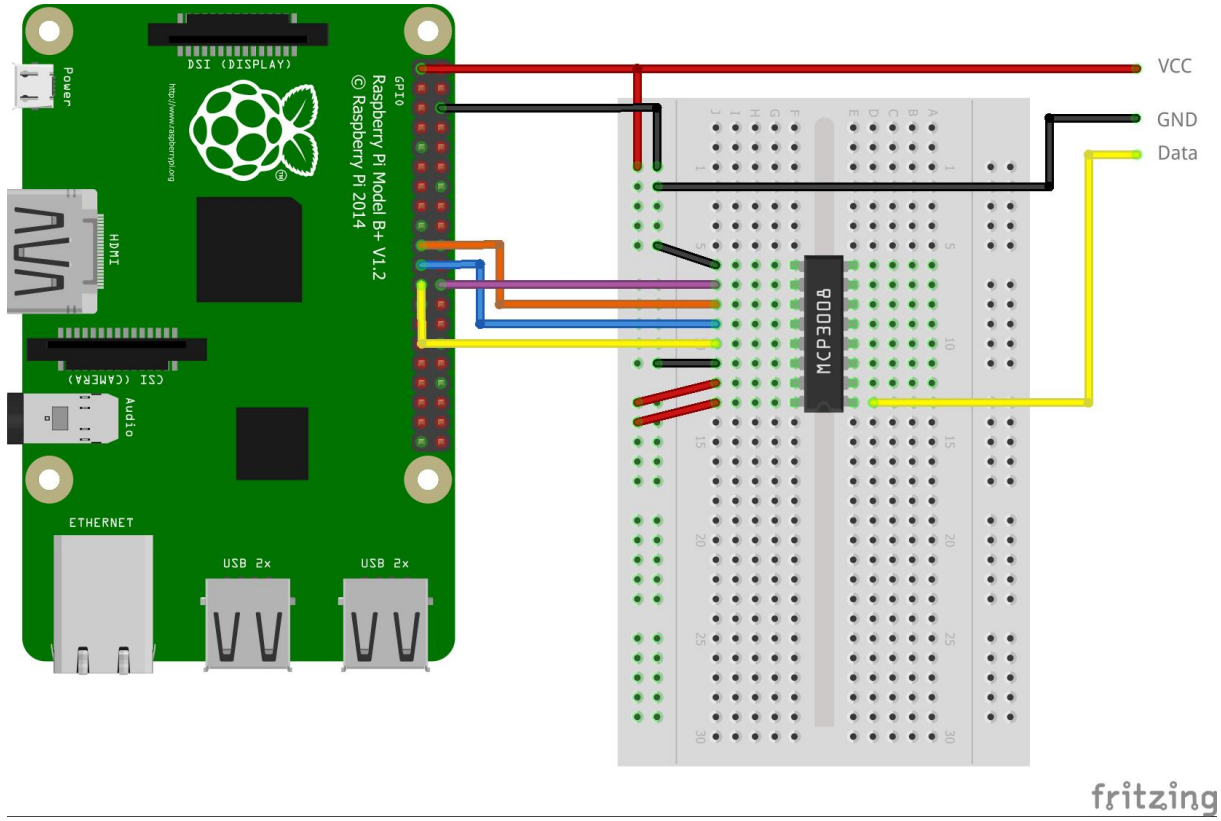
- **Yararlanılan Yazılım Bileşenleri**

1. RASPBIAN STRETCH LITE işletim sistemi
(<https://www.raspberrypi.org/downloads/raspbian/>)
2. Pycharm (<https://www.jetbrains.com/pycharm/>)
3. Matplotlib Tutorial
(<https://www.youtube.com/watch?v=UlygQI2eSdg&list=PLQVvva0QuDfefDfXb9Yf0la1fPDKluPF&index=23>)
4. Python programming (<https://pythonprogramming.net/live-graphs-matplotlib-tutorial/>)
5. Fritzing (www.fritzing.org/)
6. Raspberry Pi Heartbeat / Pulse measuring(<https://tutorials-raspberrypi.com/raspberry-pi-heartbeat-pulse-measuring/>)

- **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. Nem Sensörü: Nabız Değerlerinin alındığı sensör.
(<https://www.robotistan.com/pulse-nabiz-olcer>)
3. Analog – Dijital Dönüştürücü(MCP3002): Analog - Dijital dönüştürme (ADC) özelliğini barındırmaktadır.
4. Jumper Kablo: gerekli bağlantıları yapmak için kullanılır. Erkek-erkek ve erkek dişi iki tipi vardır.
5. 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



3.3V PWR	1		2	5V PWR
I2C1 SDA	3		4	5V PWR
I2C1 SCL	5		6	GND
GPIO 4	7		8	Reserved
GND	9		10	Reserved
SPI1 CS0	11		12	GPIO 18
GPIO 27	13		14	GND
GPIO 22	15		16	GPIO 23
3.3V PWR	17		18	GPIO 24
SPI0 MOSI	19		20	GND
SPI0 MISO	21		22	GPIO 25
SPI0 SCLK	23		24	SPI0 CS0
GND	25		26	SPI0 CS1
Reserved	27		28	Reserved
GPIO 5	29		30	GND
GPIO 6	31		32	GPIO 12
GPIO 13	33		34	GND
SPI1 MISO	35		36	GPIO 16
GPIO 26	37		38	SPI1 MOSI
GND	39		40	SPI1 SCLK

- **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.

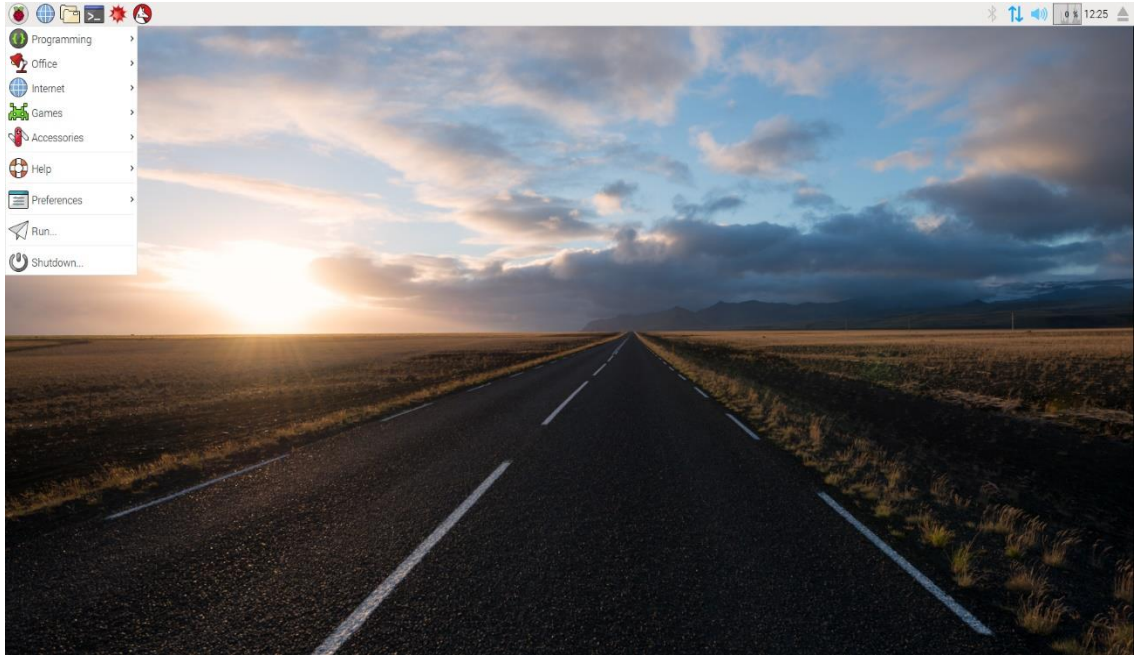
3.Bu işlemden sonra indirdiğimiz Raspberry pi yazılımı

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**

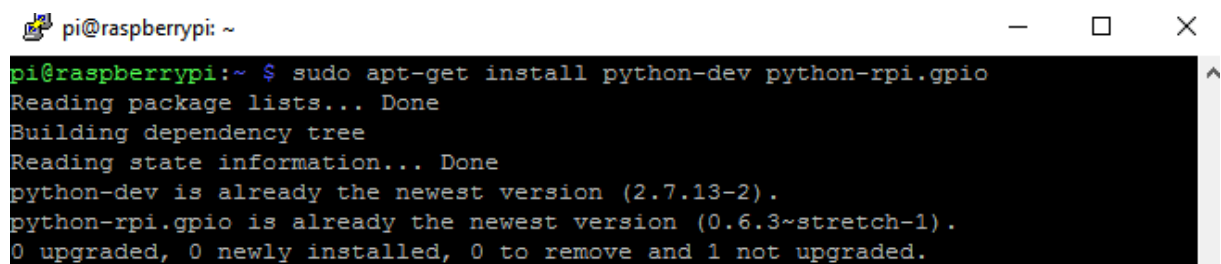
Kullanılacak olan kod için kütüphane tanımlama işlemi yapılırken arayüzden terminal kısmına giriş yapılır.

Buradan internet vasıtasıyla kütüphane yüklemeleri gerçekleştirilir.

Bu işlemlerin ardından arayüz vasıtasıyla kod yazımına başlanır.



- **RPI.GPIO kütüphanesinin tanımlanması**



Bu kütüphane ile pin giriş-çıkışlarındaki ayarlamalar yapılabilir.

- **Adafruit MCP3008 Kütüphanesinin Yüklenmesi:**

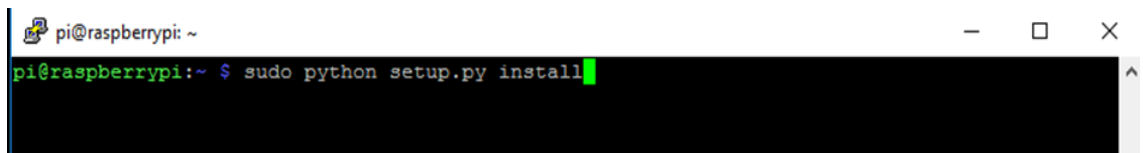
- Adafruit MCP3008 Kütüphanesini yüklerken Github sitesinden (https://github.com/adafruit/Adafruit_Python_MCP3008) direkt olarak indirme işlemi yapılabilir. İndirilen dosya içinde kod yazmaya başlanır.

- Ayrıca Adafruit MCP3008 kütüphanesini;



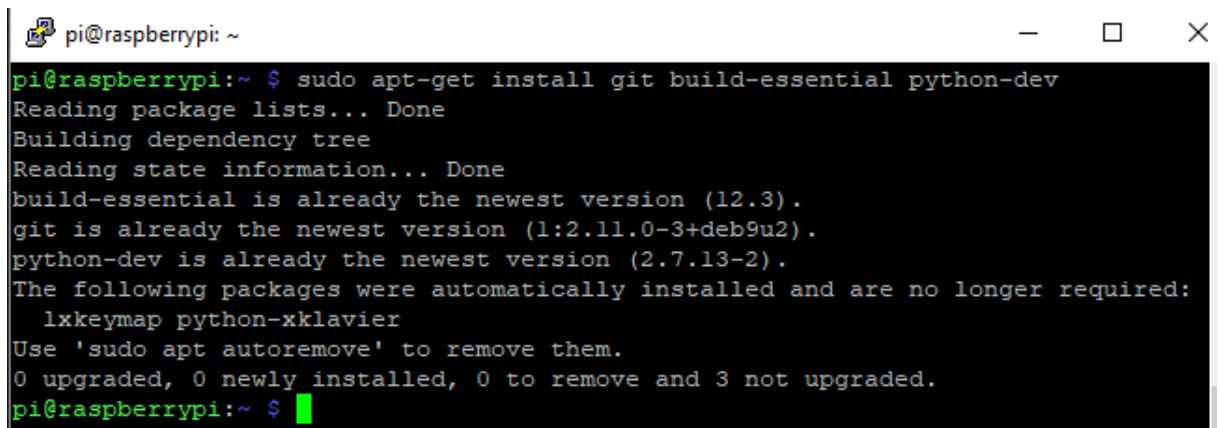
```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ sudo apt-get update
```

Komutu ile Raspberry pi kütüphane yükseltme,



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

Komutu Raspberry pi kütüphane kurulumu,



```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ sudo apt-get install git build-essential python-dev  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
build-essential is already the newest version (12.3).  
git is already the newest version (1:2.11.0-3+deb9u2).  
python-dev is already the newest version (2.7.13-2).  
The following packages were automatically installed and are no longer required:  
  lxkeymap python-xklavier  
Use 'sudo apt autoremove' to remove them.  
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.  
pi@raspberrypi:~ $
```

Komutu ile MCP3008(ADC) 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/dist-packages (from matplotlib)  
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/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/python2.7/dist-packages (from matplotlib)  
Requirement already satisfied: six>=1.10 in /usr/lib/python2.7/dist-packages (from matplotlib)  
Requirement already satisfied: pytz in /usr/local/lib/python2.7/dist-packages (from matplotlib)  
Requirement already satisfied: subprocess32 in /usr/local/lib/python2.7/dist-packages (from matplotlib)  
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python2.7/dist-packages (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 (from 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-2>) adresinden de ulaşabilirsiniz.

- **Video:**

Yaptığımız Çalışmanın Videosuna;

(<https://www.youtube.com/watch?v=kJpCEzoXnW4&t=38s>) adresinden ulaşabilirsiniz.

- **Kod yazımı:**

Öncelikle Adafruit MCP3008 kütüphanesi .lib uzantılı olmadığı için indirilen Adafruit MCP3008 klasörünün içine gidilir. Klasöre “home\pi\Adafruit MCP3008” 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.

- **Python Kodu :**

Bu kod ile nabız sensorundan den alınan verinin real-time olarak ekrana çıktısı verilmesi sağlanır. Bunun için Matplotlib kütüphanesini kullanır.

```
1. #!/usr/bin/env python
2.
3. import time
4. import collections
5. from threading import Thread
6. import matplotlib.pyplot as plt
7. import matplotlib.animation as animation
8.
9. # Import SPI library (for hardware SPI) and MCP3008 library.
10. import Adafruit_GPIO.SPI as SPI
11. import Adafruit_MCP3008
12.
13. import time
14.
15.
16. timeout = 1.5
17. last_touch = 0
18.
19. SPI_PORT = 0
20. SPI_DEVICE = 0
21. mcp = Adafruit_MCP3008.MCP3008(spi=SPI.SpiDev(SPI_PORT, SPI_DEVICE))
22.
23.
24. class heartbeadPlot:
25.     def __init__(self, plotLength = 100, dataNumBytes = 2): # gerekli tanımlamalar
26.         self.plotMaxLength = plotLength
27.         self.dataNumBytes = dataNumBytes
28.         self.rawData = bytearray(dataNumBytes)
29.         self.data = collections.deque([0] * plotLength, maxlen=plotLength)
30.         self.isRun = True
31.         self.isReceiving = False
32.         self.thread = None
33.         self.plotTimer = 0
34.         self.previousTimer = 0
35.
36.
37.     def getData(self, frame, lines, lineValueText, lineLabel, timeText):
38.         currentTime = time.clock()
39.         self.plotTimer = int((currentTime - self.previousTimer) * 1000) #ilk ok
uma hatalidir
40.         self.previousTimer = currentTime
41.         timeText.set_text('Plot Interval = ' + str(self.plotTimer) + 'ms')
42.         value = float(mcp.read_adc_difference(0))# veri alindi
43.         last_touch = time.time()
44.         self.data.append(value) # son datayi alip arraya ekliyoruz
45.         lines.set_data(range(self.plotMaxLength), self.data)
46.         lineValueText.set_text('[' + lineLabel + '] = ' + str(value))
47.
48.     def backgroundThread(self): # dataya tekrar bak
```

```

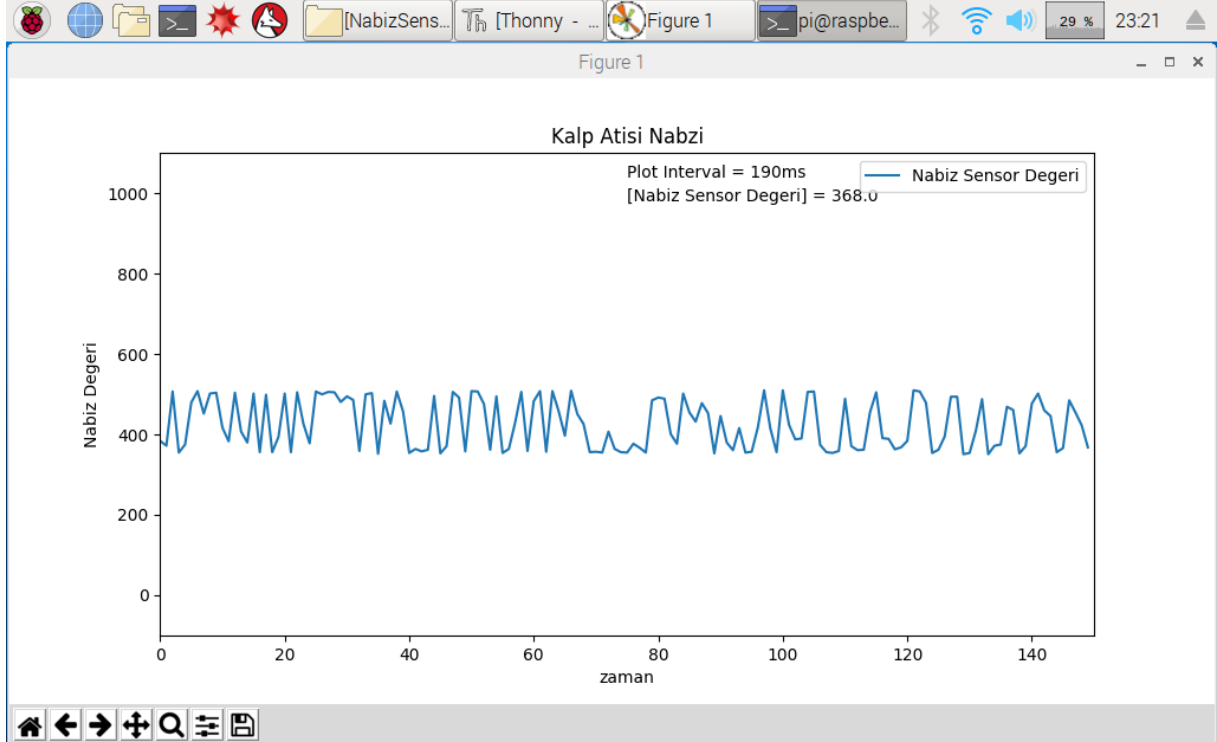
49.         time.sleep(1.0)
50.         while (self.isRun):
51.             self.serialConnection.readinto(self.rawData)
52.             self.isReceiving = True
53.
54.         def close(self):
55.             self.isRun = False
56.             self.thread.join()
57.             print('Baglanti kesildi...')
58.
59.
60.
61.     def main():
62.
63.         maxPlotLength = 150
64.         dataNumBytes = 2 #4          # bir data doktasinin byte degeri
65.         s = heartbeatPlot(maxPlotLength, dataNumBytes) # gerekli degiskenler orneklend
66.         i
67.         # grafikleme baslangici
68.         pltInterval = 1 # grafik animasyonunun peryodu [ms]
69.         xmin = 0
70.         xmax = maxPlotLength
71.         ymin = 0
72.         ymax = 1000
73.         fig = plt.figure()
74.         ax = plt.axes(xlim=(xmin, xmax), ylim=(float(ymin - (ymax - ymin) / 10), float(y
75.         max + (ymax - ymin) / 10)))
76.         ax.set_title('Kalp Atisi Nabzi')
77.         ax.set_xlabel("zaman")
78.         ax.set_ylabel("Nabiz Degeri")
79.         lineLabel = 'Nabiz Sensor Degeri'
80.         timeText = ax.text(0.50, 0.95, '', transform=ax.transAxes)
81.         lines = ax.plot([], [], label=lineLabel)[0]
82.         lineValueText = ax.text(0.50, 0.90, '', transform=ax.transAxes)
83.         anim = animation.FuncAnimation(fig, s.getData, fargs=(lines, lineValueText, line
84.         Label, timeText), interval=pltInterval) # animasyon baslasin
85.
86.         plt.legend(loc="sol ust")
87.         plt.show()
88.         s.close()
89.
90.
91. if __name__ == '__main__':
92.     main() # main fonksiyonu cagrilir

```

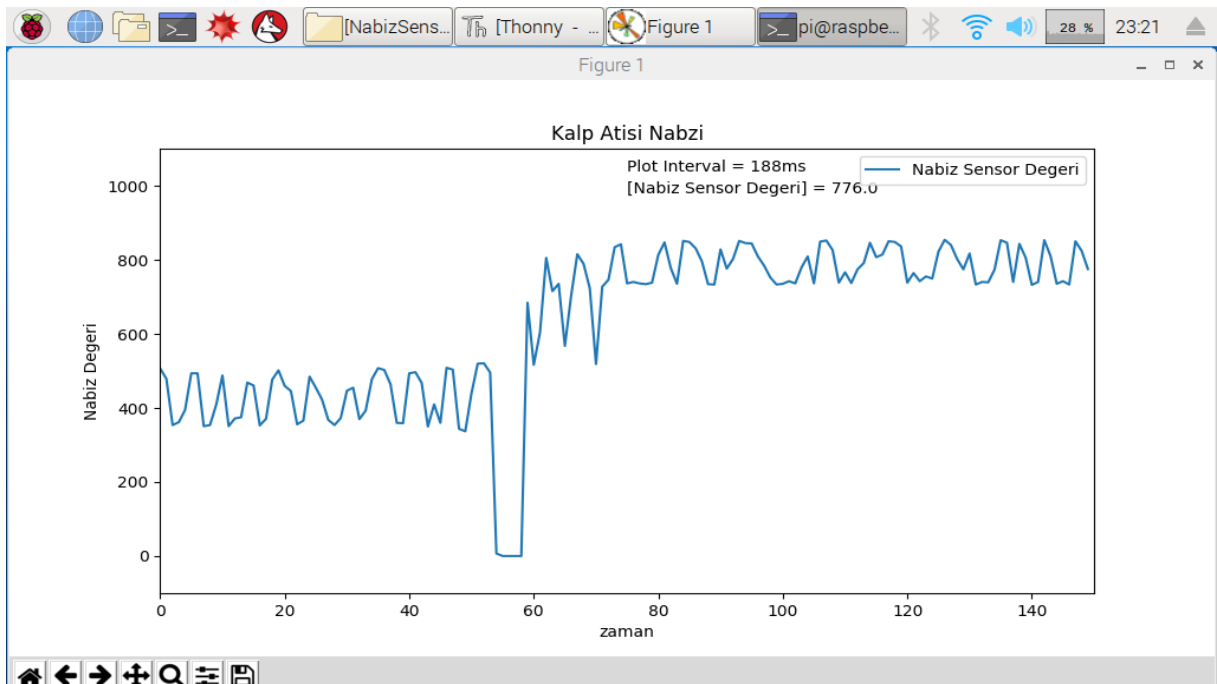
Matplotlib ile Real-Time grafik Ekran Görüntüleri:

- Nabız Sensörüne bir etki olmadığında çıkan grafik:

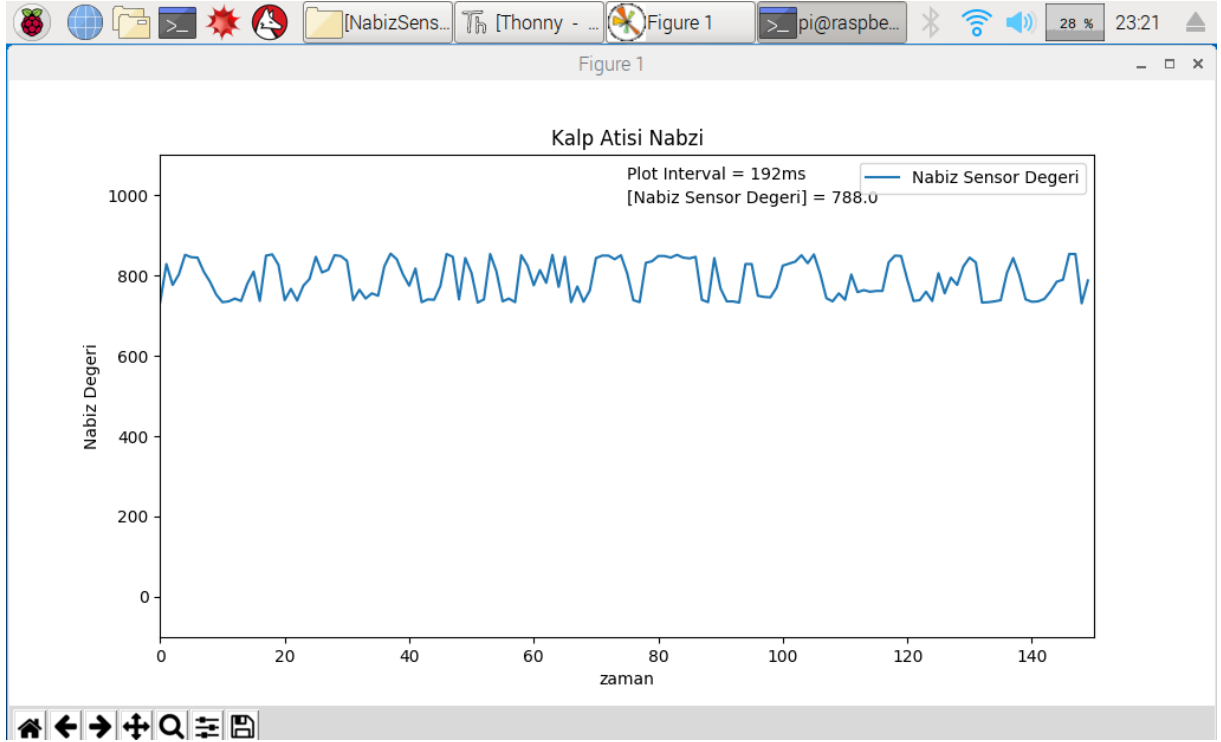
(Not: Dalgalanma sensorun yapısından kaynaklanmaktadır.)



- Nabız Sensörüne parmak ile dokunulduğundaki grafikteki değişim:

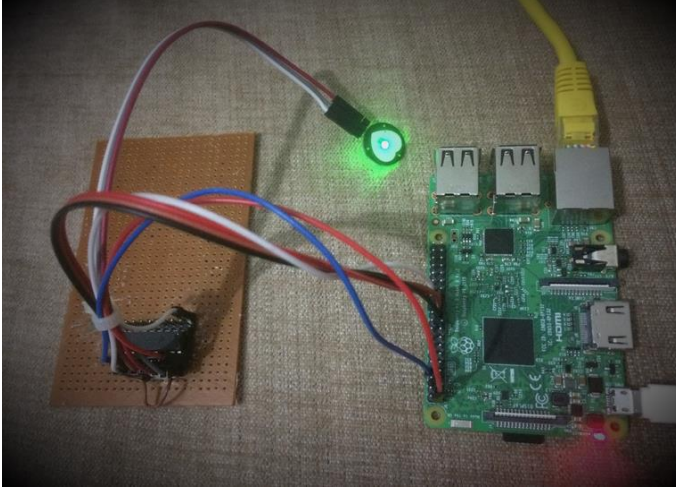


- Nabız Sensöründen alınan parmaktaki nabız bilgisi :

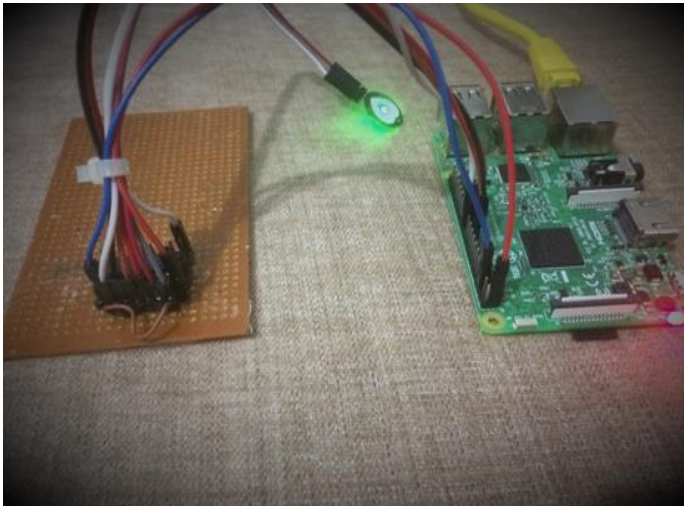
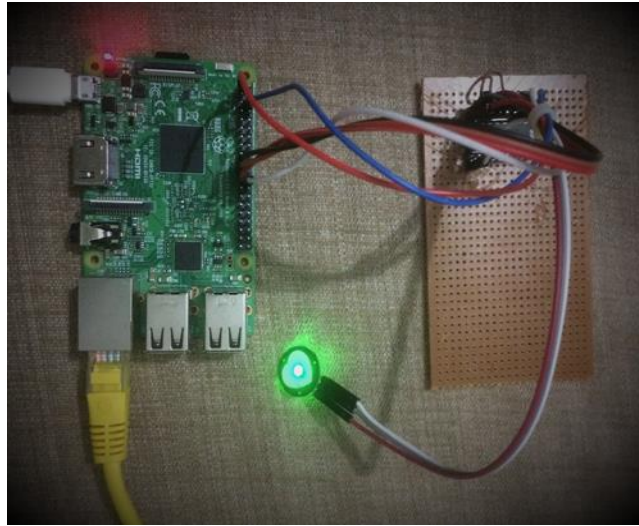


Gerçek zamanlı olarak ekrana verilmektedir.

- Proje Resimleri:



Raspberry
Pi ve
Board
Kurulumu



Proje Geliştirme Önerileri:

- Proje board üzerinden daha estetik bir yapıya kavuşturulabilir.
- Projenin python kodu daha da geliştirilerek daha net ve doğru grafikler alınması sağlanabilir.
- Kütüphane olarak matplotlib kütüphanesi yerine pythondaki başka grafik kütüphaneleri kullanılabilir.
- Nabız değerlerinin kullanıcıyı daha iyi anlaması için çeşitli arayüz çalışmaları yapılabilir.
- Ürünün bir satış durumu olması için çeşitli tasarımsal çalışmalar yapılabilir.
- Projeye farklı sağlık ile ilgili sensorler ilave ederek (ekg, ritim vb.) daha fazla işlevsellik kazandırılabilir.

