Heart Monitor

Aya Farag 900160580

TABLE OF CONTENTS

01 About the Project

02 Hardware/Pins

03 ADC

04 Obtaining Data for One minute

04 Calculating BPM

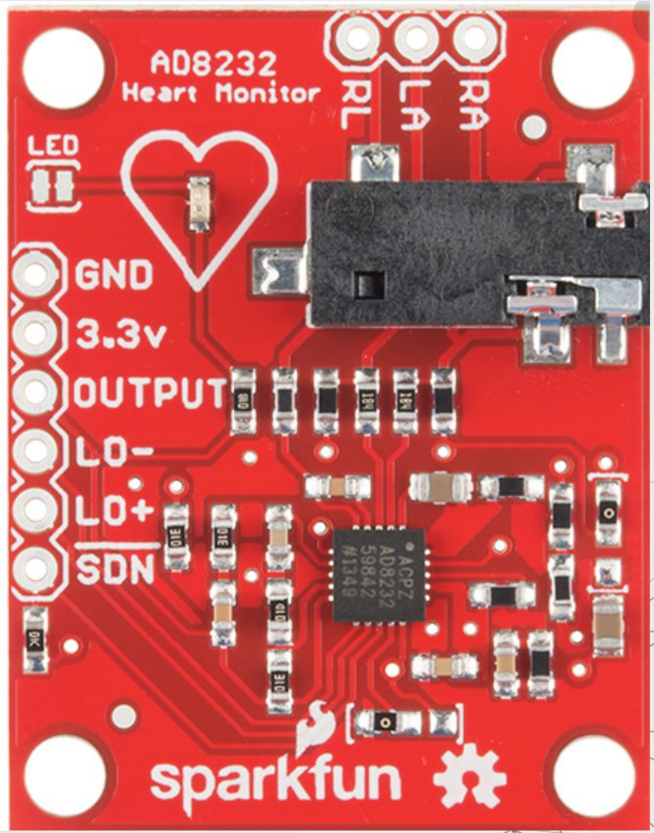
05 Code Walkthrough

06 Pyserial

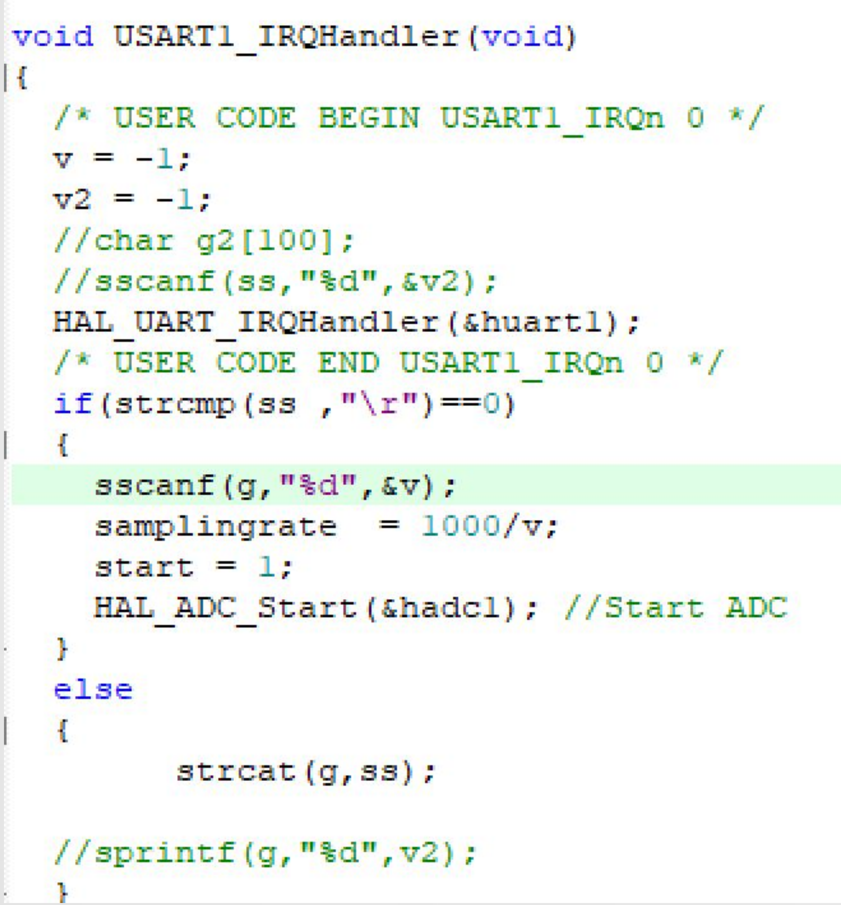
07 Result

About the Project

For my project I’ve chosen to implement a simple heart monitor. The set requirements is that I have to develop it using the STM32 Module and the ECG Sensor, and using both I should be able to collect ECG signal, and report it to the PC over USB link, and the data received should be graphed , I have to report the heart beat rate, set a sampling rate and collect one minute worth of data. The ECG data must be graphed and I should provide UI elements, to select COM port, and baud rate, and UI elements to set a sampling rate.



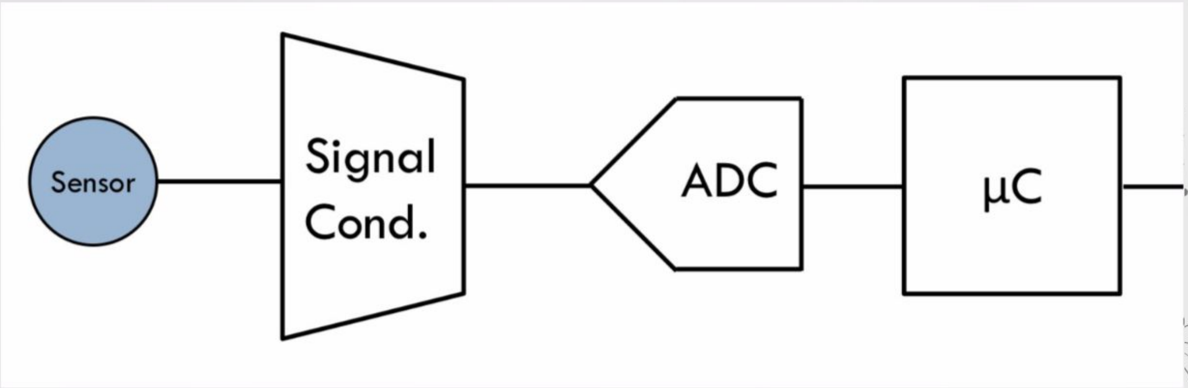
Code Walkthrough



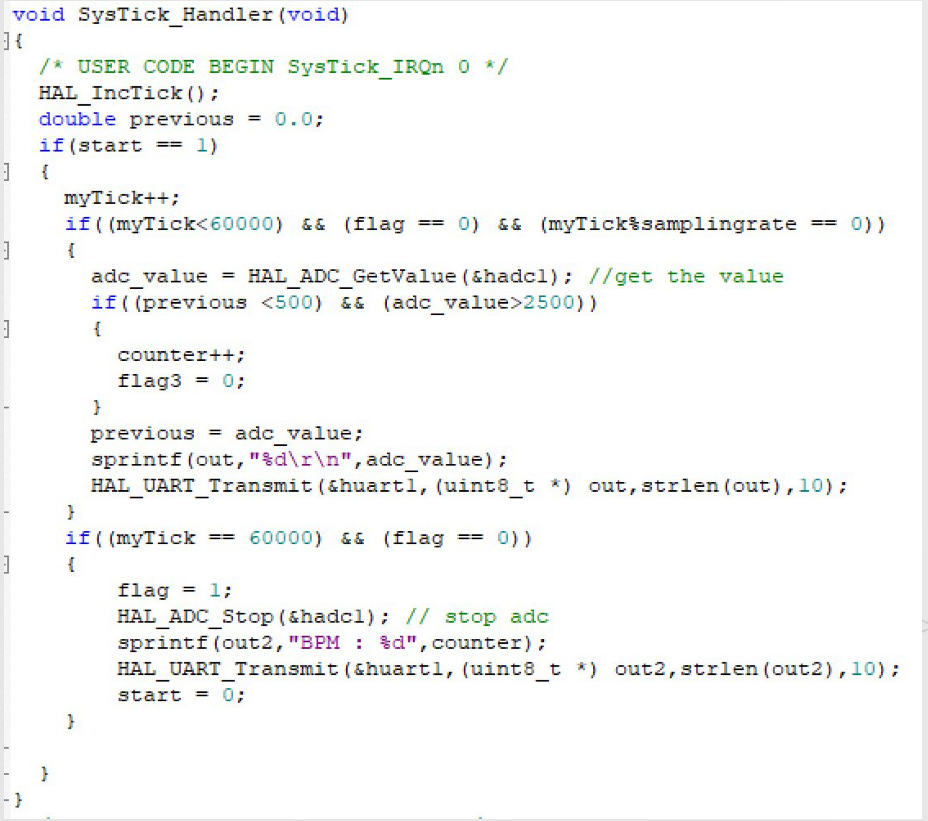
```
void USART1_IRQHandler(void)
{
  /* USER CODE BEGIN USART1_IRQn 0 */
  v = -1;
  v2 = -1;
  //char g2[100];
  //sscanf(ss,"%d",&v2);
  HAL_UART_IRQHandler(&huart1);
  /* USER CODE END USART1_IRQn 0 */
  if(strcmp(ss ,"\r")==0)
  {
    sscanf(g,"%d",&v);
    samplingrate  = 1000/v;
    start = 1;
    HAL_ADC_Start(&hadc1); //Start ADC
  }
  else
  {
        strcat(g,ss);

  //sprintf(g,"%d",v2);
  }
```

Setting sampling Rate

- I first wait for an input value to set the sampling rate , with \r as a stopping condition
- Once the user presses enter I set the sampling rate and start the ADC.



Code Walkthrough

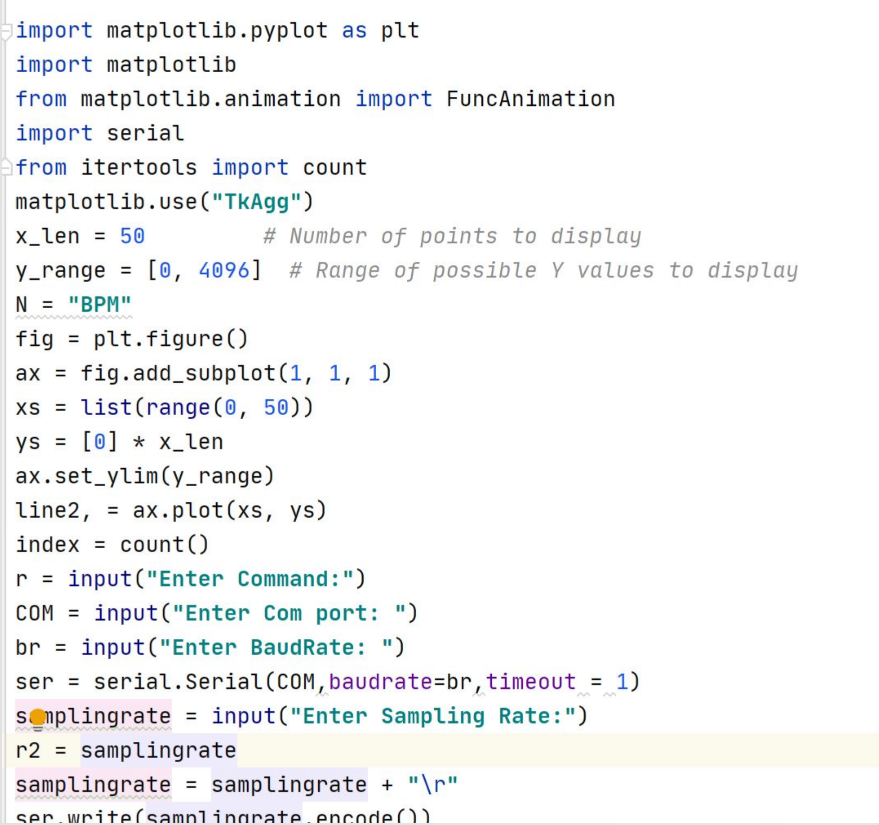


```
void SysTick_Handler(void)
{
  /* USER CODE BEGIN SysTick_IRQn 0 */
  HAL_IncTick();
  double previous = 0.0;
  if(start == 1)
  {
    myTick++;
    if((myTick<60000) && (flag == 0) && (myTick%samplingrate == 0))
    {
      adc_value = HAL_ADC_GetValue(&hadc1); //get the value
      if((previous <500) && (adc_value>2500))
      {
        counter++;
        flag3 = 0;
      }
      previous = adc_value;
      sprintf(out,"%d\r\n",adc_value);
      HAL_UART_Transmit(&huart1,(uint8_t *) out,strlen(out),10);
    }
    if((myTick == 60000) && (flag == 0))
    {
        flag = 1;
        HAL_ADC_Stop(&hadc1); // stop adc
        sprintf(out2,"BPM : %d",counter);
        HAL_UART_Transmit(&huart1,(uint8_t *) out2,strlen(out2),10);
        start = 0;
    }

  }
}
```

Reading From ECG & Calculating Beats Per Minute

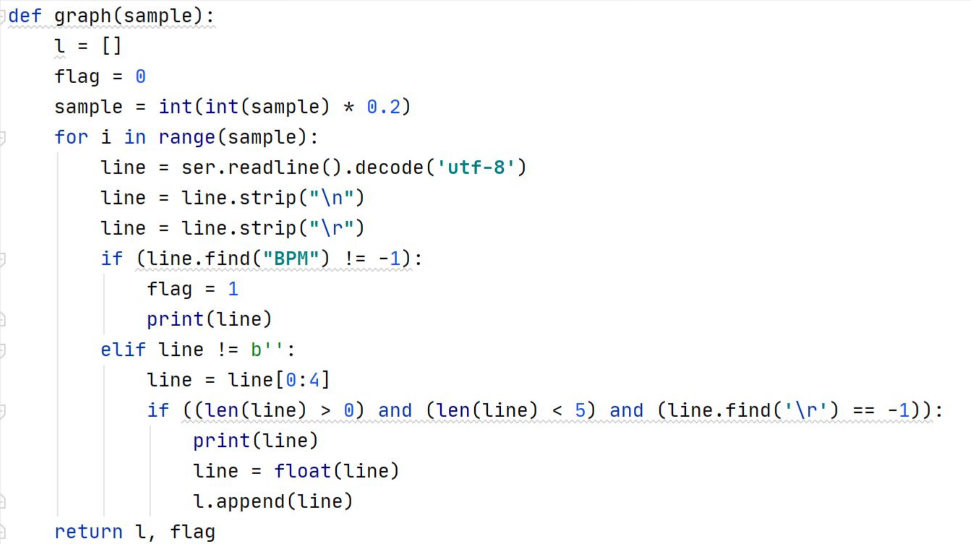
- To Start reading from the ECG, I set a flag in the previous function that states I started the ADC, so Once this flag is set, I start reading.
- I keep reading for one minute using the systick timer.
- Along the way to calculate BPM, every value I get I look for 2 consecutive values that go from low to high, so I save the previous read value and the current value, and if they pass a certain threshold I increment the BPM counter, this is done for one minute, then I transmit BPM.
- Along the way I'm transmitting all the values I'm receiving in order to be able to use them in my Python code to graph.

PySerial- Com Ports and Baud Rate

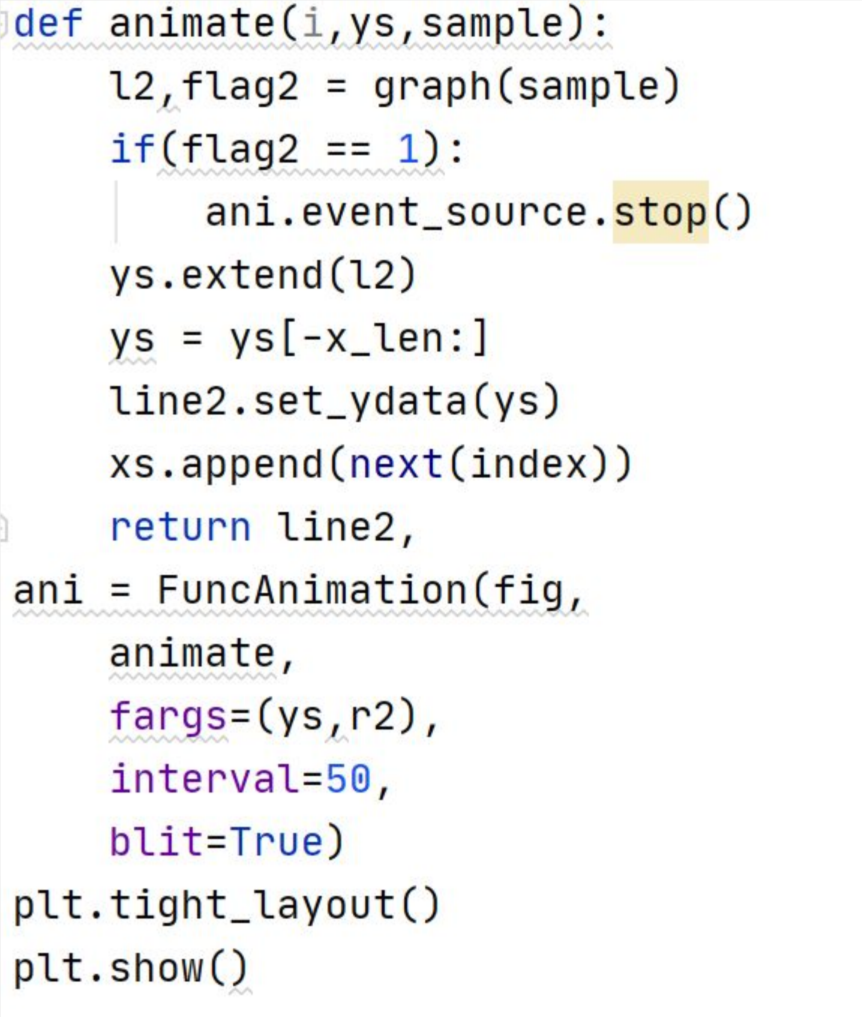


```
import matplotlib.pyplot as plt
import matplotlib
from matplotlib.animation import FuncAnimation
import serial
from itertools import count
matplotlib.use("TkAgg")
x_len = 50          # Number of points to display
y_range = [0, 4096]  # Range of possible Y values to display
N = "BPM"
fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
xs = list(range(0, 50))
ys = [0] * x_len
ax.set_ylim(y_range)
line2, = ax.plot(xs, ys)
index = count()
r = input("Enter Command:")
COM = input("Enter Com port: ")
br = input("Enter BaudRate: ")
ser = serial.Serial(COM,baudrate=br,timeout = 1)
samplingrate = input("Enter Sampling Rate:")
r2 = samplingrate
samplingrate = samplingrate + "\r"
ser.write(samplingrate.encode())
```

- I take as a user input, the starting command “S”, Baus Rate, Com port
  .
- After I’ve received the above 3 inputs, I open the port to start reading from it.
- I then take the sampling rate as a user input and I write the sampling rate in the port with an enter to start sampling and reading.



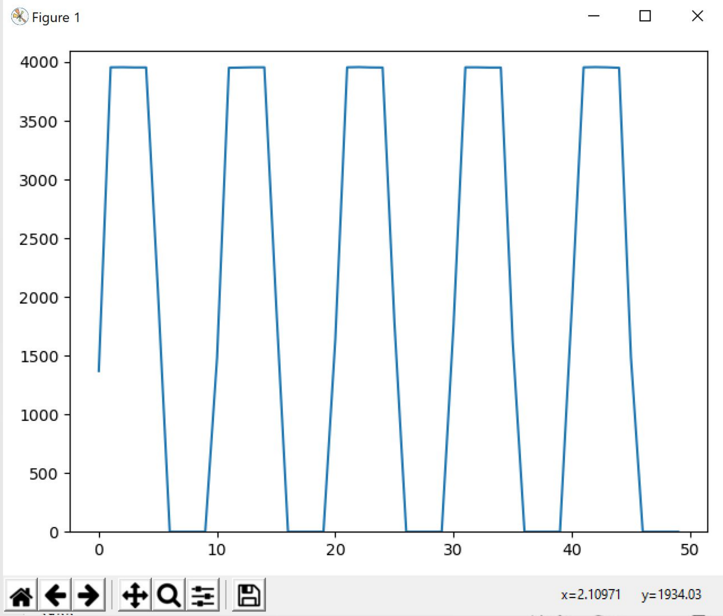
```
def graph(sample):
    l = []
    flag = 0
    sample = int(int(sample) * 0.2)
    for i in range(sample):
        line = ser.readline().decode('utf-8')
        line = line.strip("\n")
        line = line.strip("\r")
        if (line.find("BPM") != -1):
            flag = 1
            print(line)
        elif line != b'':
            line = line[0:4]
            if ((len(line) > 0) and (len(line) < 5) and (line.find('\r') == -1)):
                print(line)
                line = float(line)
                l.append(line)
    return l, flag
```



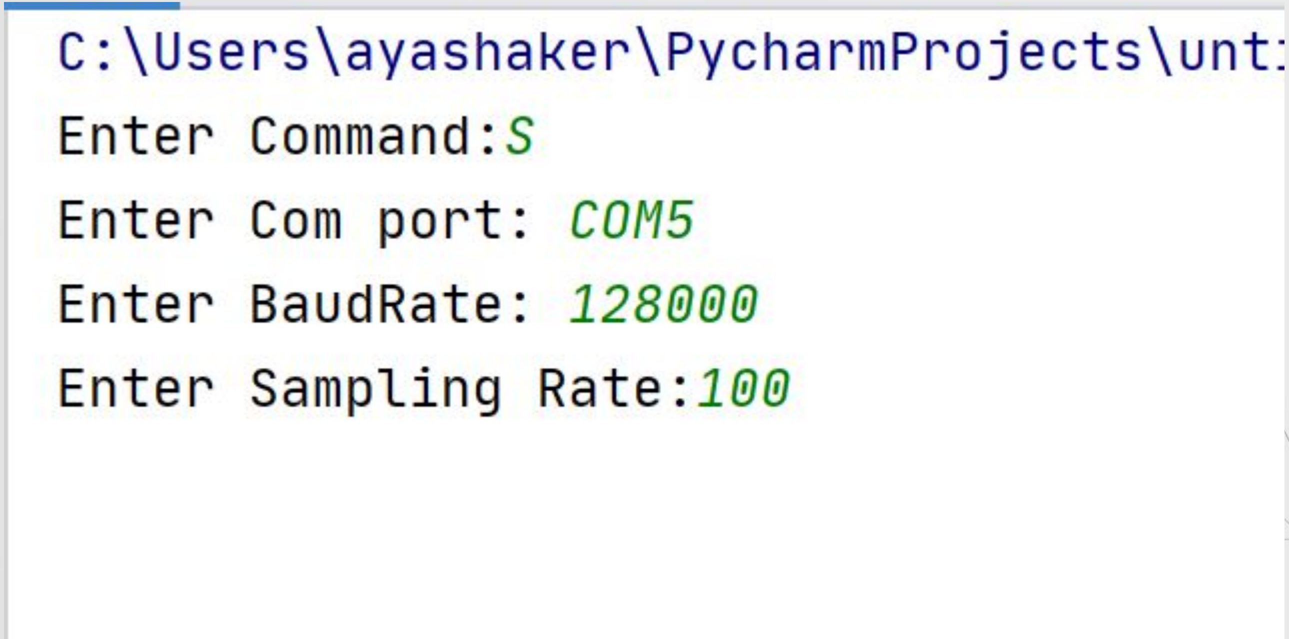
```
def animate(i,ys,sample):
    l2,flag2 = graph(sample)
    if(flag2 == 1):
        ani.event_source.stop()
    ys.extend(l2)
    ys = ys[-x_len:]
    line2.set_ydata(ys)
    xs.append(next(index))
    return line2,
ani = FuncAnimation(fig,
    animate,
    fargs=(ys,r2),
    interval=50,
    blit=True)
plt.tight_layout()
plt.show()
```

Graphing

- I plot the data I receive from the port live, so my plot is animated.
- Since when the sampling rate is high, the plot is delayed, therefore
- I dont plot each sample every time, I concatenate 25% of the
- inputted data and I graph them at once, this helped reduce the delay

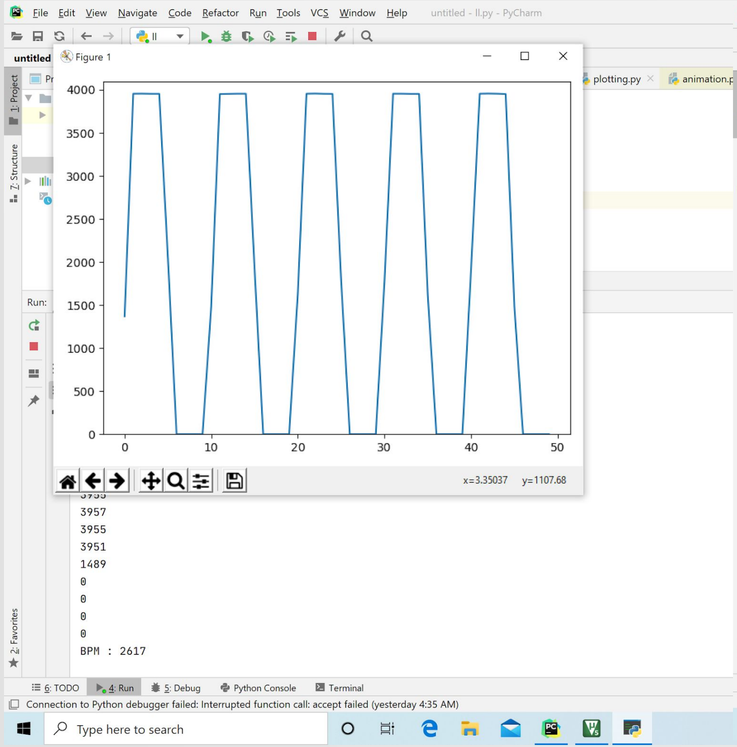


User input:



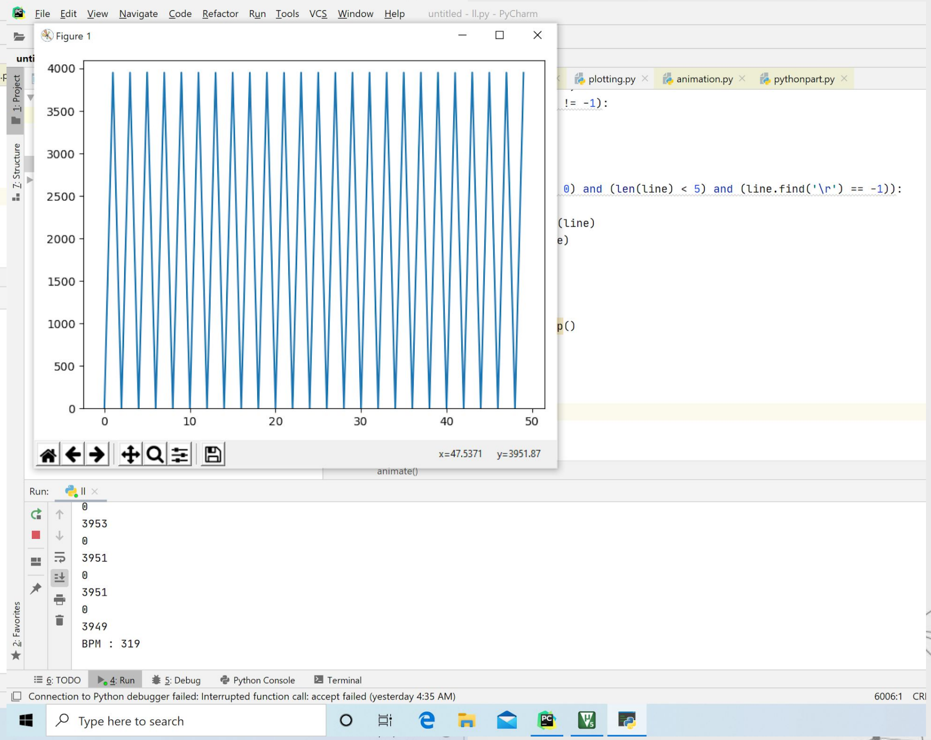
```
C:\Users\ayashaker\PycharmProjects\unt
Enter Command:S
Enter Com port: COM5
Enter BaudRate: 128000
Enter Sampling Rate:100
```

Results



```
3957
3955
3951
1489
0
0
0
0
BPM : 2617
```

Connection to Python debugger failed: Interrupted function call: accept failed (yesterday 4:35 AM)



```
 != -1):

 0) and (len(line) < 5) and (line.find('\r') == -1)):

(line)
e)

p()

animate()
```

```
0
3953
0
3951
0
3951
0
3949
BPM : 319
```

Connection to Python debugger failed: Interrupted function call: accept failed (yesterday 4:35 AM)

To Do	Set Due Date
Get enough Data for a minute to get Beats per minute, and view it + Taking COM Rate and Baud Rate from user	Phase1
View Heart Beat Signal-Graphed	Phase2
Set sampling Rate / Report/ Presentation	Phase2
Documentation on GitHub	Phase2
Finish Project/Finalized	Phase2

DEMO

Github Link: https://github.com/ayashaker98/Embbeded-Project-REP

Thank you