ACTIVITY_2_SENSORS

August 28, 2018

In [66]:
```
import csv
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import math
from gmplot import gmplot
```

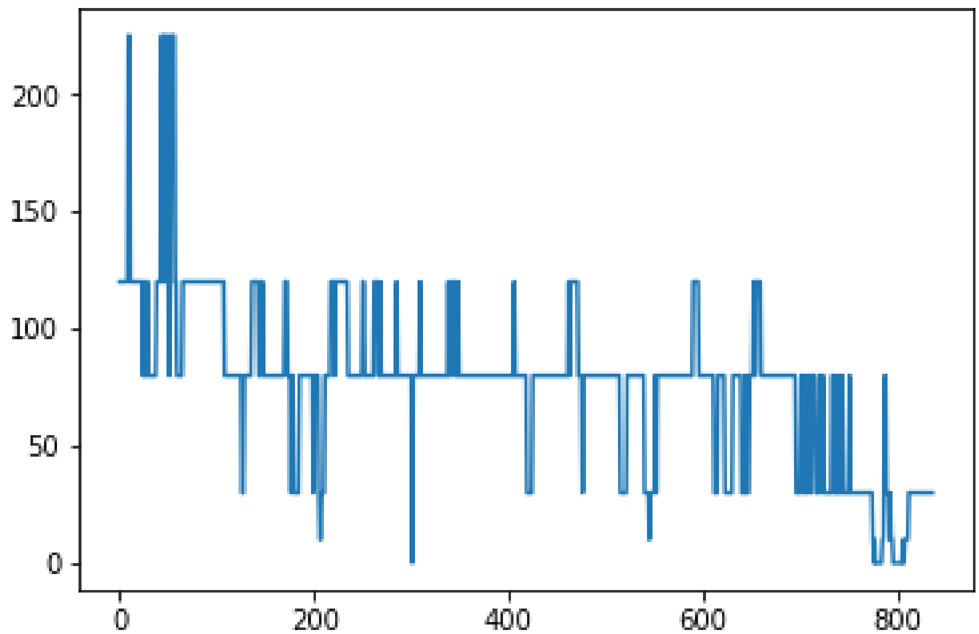
In [75]:
```
df = pd.read_csv('Sensor_record_20180807_185126_AndroSensor.csv',sep=',',header=1)
df=df.set_index('Time since start in ms ')
```

In [68]:
```
e=df['LOCATION Latitude : ']
g=df['LOCATION Longitude : ']
gmap1 = gmplot.GoogleMapPlotter(float(e.iloc[1]), float(g.iloc[1]), 13)

gmap1.plot(e, g, 'cornflowerblue', edge_width = 2.5)

#gmap1.scatter(e, g, '#3B0B39', size=40, marker=False)
gmap1.draw("/home/kpit/Downloads/map3.html")
```

1 CALCULATING NUMBER OF STREET LIGHTS AND COCONUT TREES

In [69]:
```
#Taking LUX values from dataframe.
a=df['LIGHT (lux)']
plt.plot(a)
cou_street_lights=0
for i in a:
    if i >=80:
        cou_street_lights=cou_street_lights+1
print("NUMBER OF STREET LIGHTS:")
print(int(cou_street_lights/30))
cou_coconut_trees=0
for i in a:
    if i<=40:
        cou_coconut_trees+=1
print("NUMBER OF COCONUT TREES:")
print(int(cou_coconut_trees)/6)
```

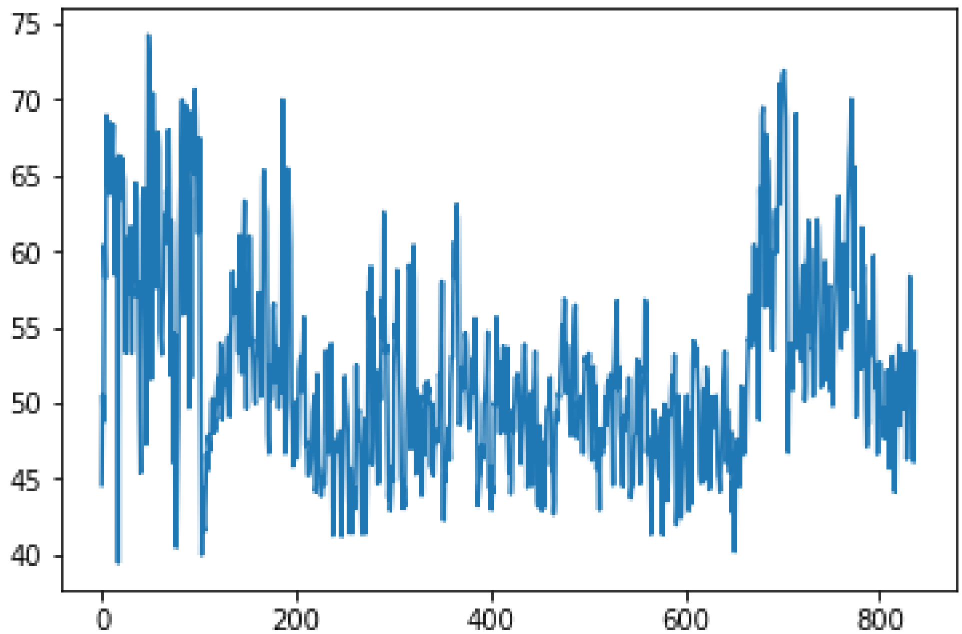
```
NUMBER OF STREET LIGHTS:
21
NUMBER OF COCONUT TREES:
30.0
```



2 PASSING SIGNAL TO MOBILE PHONE TO INCREASE VOLUME BASED ON SURROUNDING NOISE.

```
In [70]: c=df['SOUND LEVEL (dB)']
         plt.plot(c)
         print("MAXIMUM SOUND INTENSITY:")
         max_sound=max(c)
         print(max(c))
         mea=abs(c).mean()
         print("MEAN SOUND INTENSITY:")
         print(mea)
         if max_sound <40:
             print("INCREASE VOLUME TO 45 db")
         elif max_sound>40 and max_sound<60:
             print("INCREASE VOLUME TO 65 dB")
         else:
             print("INCREASE VOLUME TO 80 dB")
```

```
MAXIMUM SOUND INTENSITY:
74.273
MEAN SOUND INTENSITY:
52.268205495818385
INCREASE VOLUME TO 80 dB
```

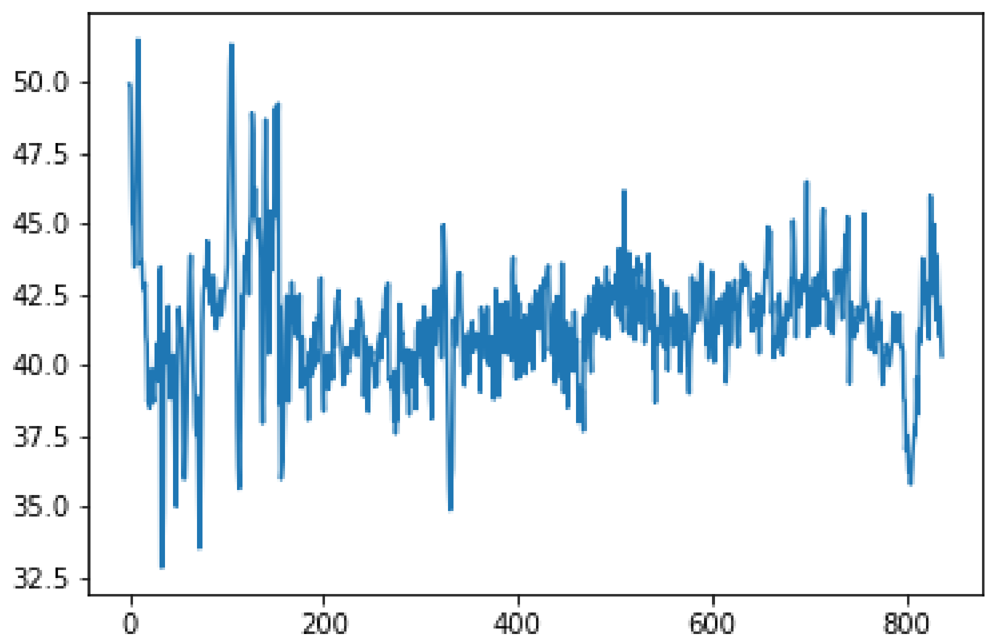


3 MAGNETIC FIELD

```
In [71]: m=df['MAGNETIC FIELD X (T)']
         MX = np.array([])
         MY = np.array([])
         MZ = np.array([])
         for i in range(len(df['MAGNETIC FIELD X (T)'])):
             MX = np.append(MX, float(df['MAGNETIC FIELD X (T)'][i].tolist()))
             MY = np.append(MY, float(df['MAGNETIC FIELD Y (T)'][i].tolist()))
             MZ = np.append(MZ, float(df['MAGNETIC FIELD Z (T)'][i].tolist()))
         MT = np.sqrt(np.square(MX)+np.square(MY)+np.square(MZ))
         plt.plot(MT)
         print("MAXIMUM MAGNETIC FIELD:")
         print(max(MT))
         print("MEAN MAGNETIC INTENSITY:")
         mea=abs(MT).mean()
         print(mea)
```

```
DF1=df.loc[df['MAGNETIC FIELD X (T)'] >=30]
lat=DF1['LOCATION Latitude : ']
long=DF1['LOCATION Longitude : ']

gmap2 = gmplot.GoogleMapPlotter(float(lat.iloc[1]), float(long.iloc[1]), 13)

gmap2.heatmap(lat, long, radius=25,opacity=0.6, dissipating=True, threshold=10)

gmap2.draw("/home/kpit/Downloads/magnetic_intensity.html")
```

```
MAXIMUM MAGNETIC FIELD:
51.48736835380111
MEAN MAGNETIC INTENSITY:
41.448902866685565
```



4 CALCULATING EUCLIDIAN ACCELERATION

```
In [72]: AX = np.array([])
         AY = np.array([])
         AZ = np.array([])
         for i in range(len(df['LINEAR ACCELERATION Y (m/sš)'])):
             AX = np.append(AX, float(df['LINEAR ACCELERATION X (m/sš)'][i].tolist()))
             AY = np.append(AY, float(df['LINEAR ACCELERATION Y (m/sš)'][i].tolist()))
             AZ = np.append(AZ, float(df['LINEAR ACCELERATION Z (m/sš)'][i].tolist()))
```

```python
A_TOT = np.sqrt(np.square(AX)+np.square(AY)+np.square(AZ))
print("AVERAGE EUCLIDIAN ACCELERATION:")
print(A_TOT.mean())
plt.plot(A_TOT)
```

```
AVERAGE EUCLIDIAN ACCELERATION:
2.0447466272235886
```

```
Out[72]: [<matplotlib.lines.Line2D at 0x7ff91023f710>]
```