Group 2

2a) Wouldn’t it be great if you could connect your Raspberry Pi from your mobile phone or tablet and display all the images of your mobile on the personal computer?

Aim: To display the data of the phone on the raspberry pi Opearting System.

Hardware Requirements: Raspberry pi 3, Ethernet Cable, HDMI adapter, Keyboard, Mouse, Monitor, USB cable, smart phone, Raspberry pi Adapter.

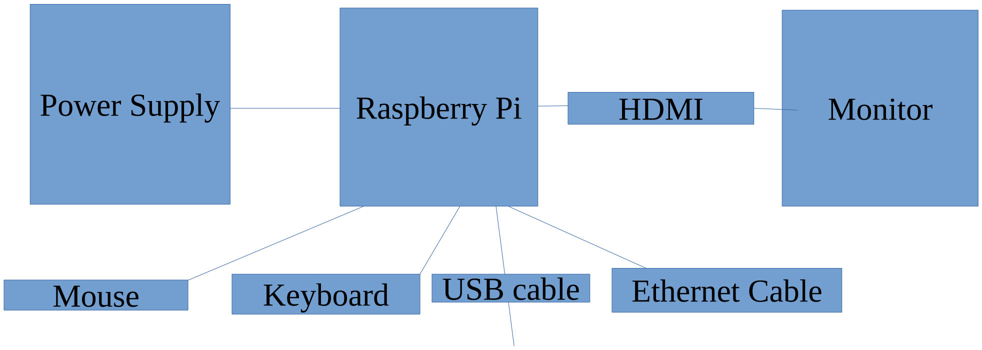
Software Requirements: Ubuntu Operating system, Python, Raspberry Pi Operating system

Description: Raspberry Pi is a minuture computer that performs the basic functions of a real time PC. It contains the ethernet cable port that used to communicate with the internet. It also contians a wifi module in order to connect to the internet. HDMI adapter allows the raspberry to display it’s Opearting system on the monitor. Raspberry Pi has multiple built in 2x2 I/O ports. Raspberry pi runs on Scratch and Python to run the programs in the OS. The RaspOS can be manipulated through the use of basic keyboard and a mouse. Raspberry Pi is a computer powered by the Broadcom BCM2835 system-on-a-chip (SoC). This SoC includes a 32-bit ARM1176JZFS processor, clocked at 700MHz, and a Videocore IV GPU. It also has 256MB of RAM in a POP package above the SoC. The Raspberry Pi is powered by a 5V micro USB AC charger or at least 4 AA batteries.

Steps/Procedure:

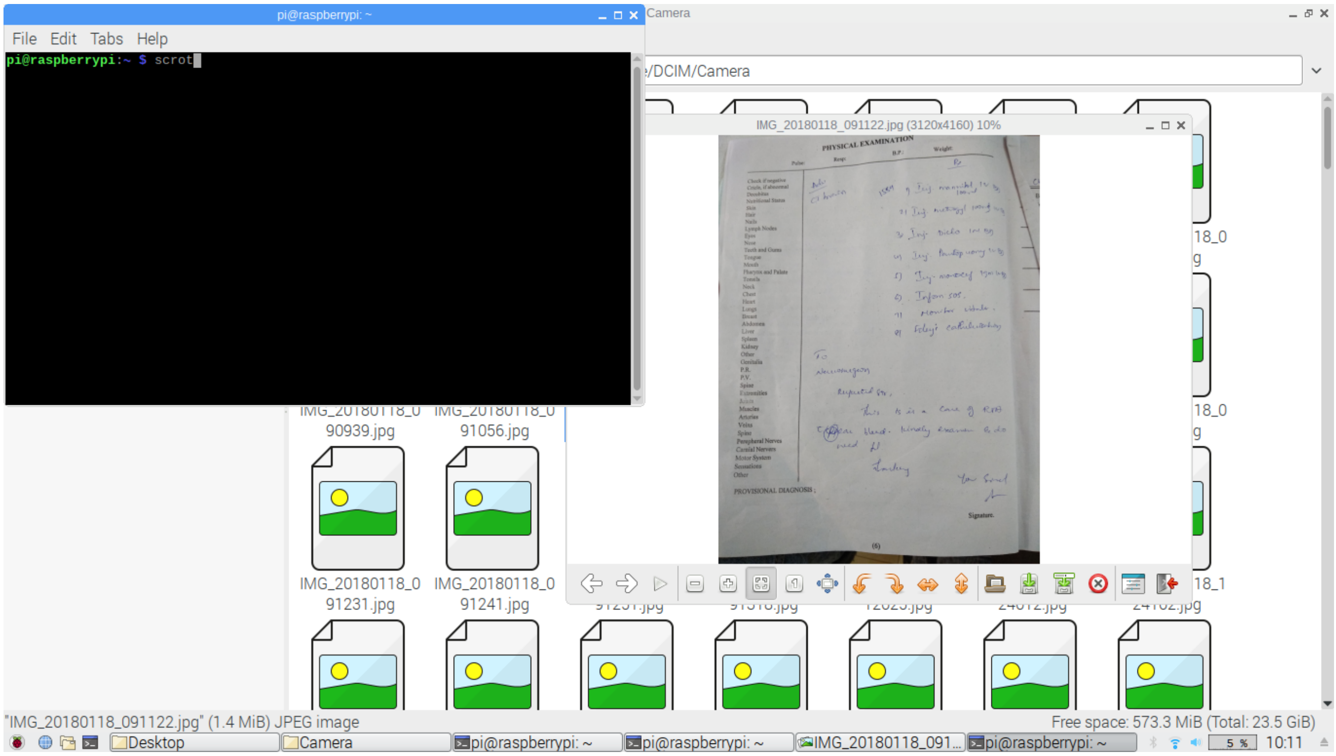
- Acquire Raspberry pi.
- Connect the keyboard and Mouse to the Raspberry pi Device
- Connect the Ethernet cable to the Raspberry pi Device
- Power the Raspberry pi Device the Power supply Adapter.
- Connect the hdmi to the monitor from Raspberry Pi.
- Connect the Power supply to the CPU.
- Turn on the Raspberry pi.
- Connect the Mobile phone to the Rapsberry pi
- Access data from the Data.

Circuit Diagram:



Smart Phone

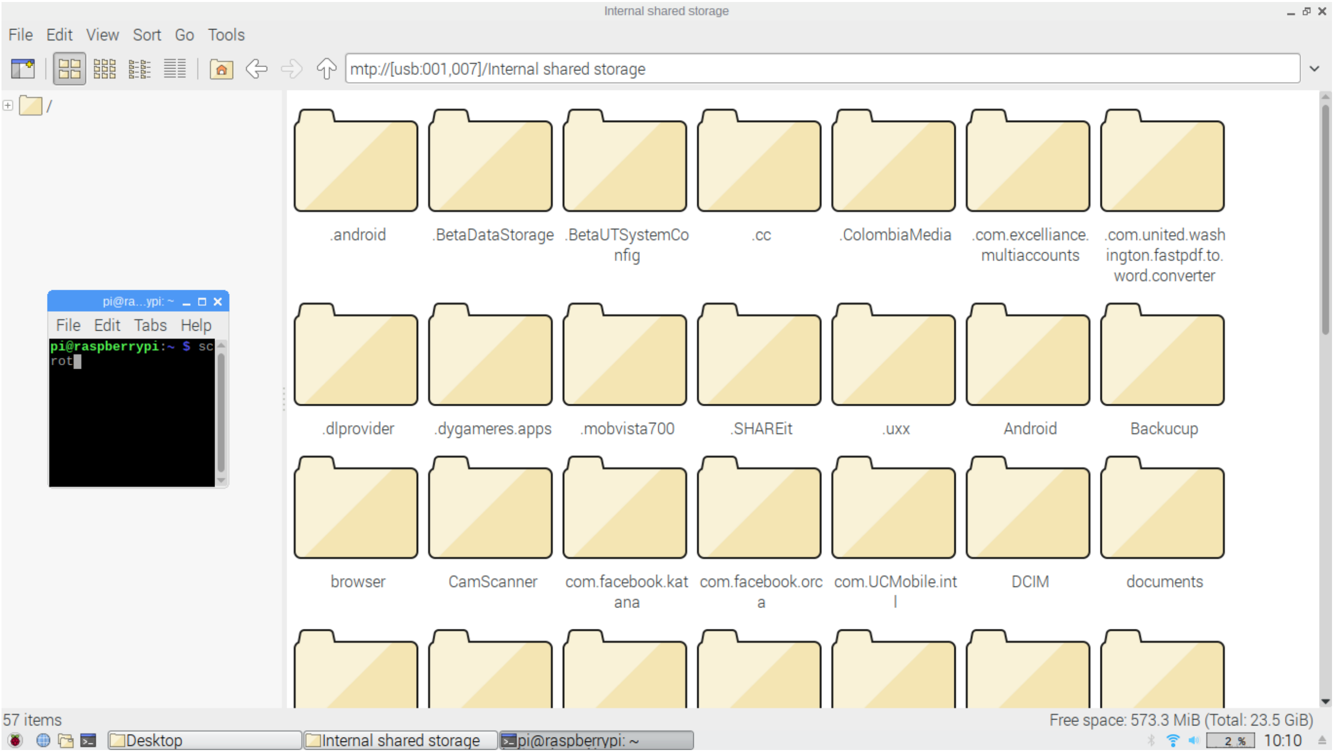
Output:



```
pi@raspberrypi:~ $ scrot
```

"IMG_20180118_091122.jpg" (1.4 MiB) JPEG image

Free space: 573.3 MiB (Total: 23.5 GiB)

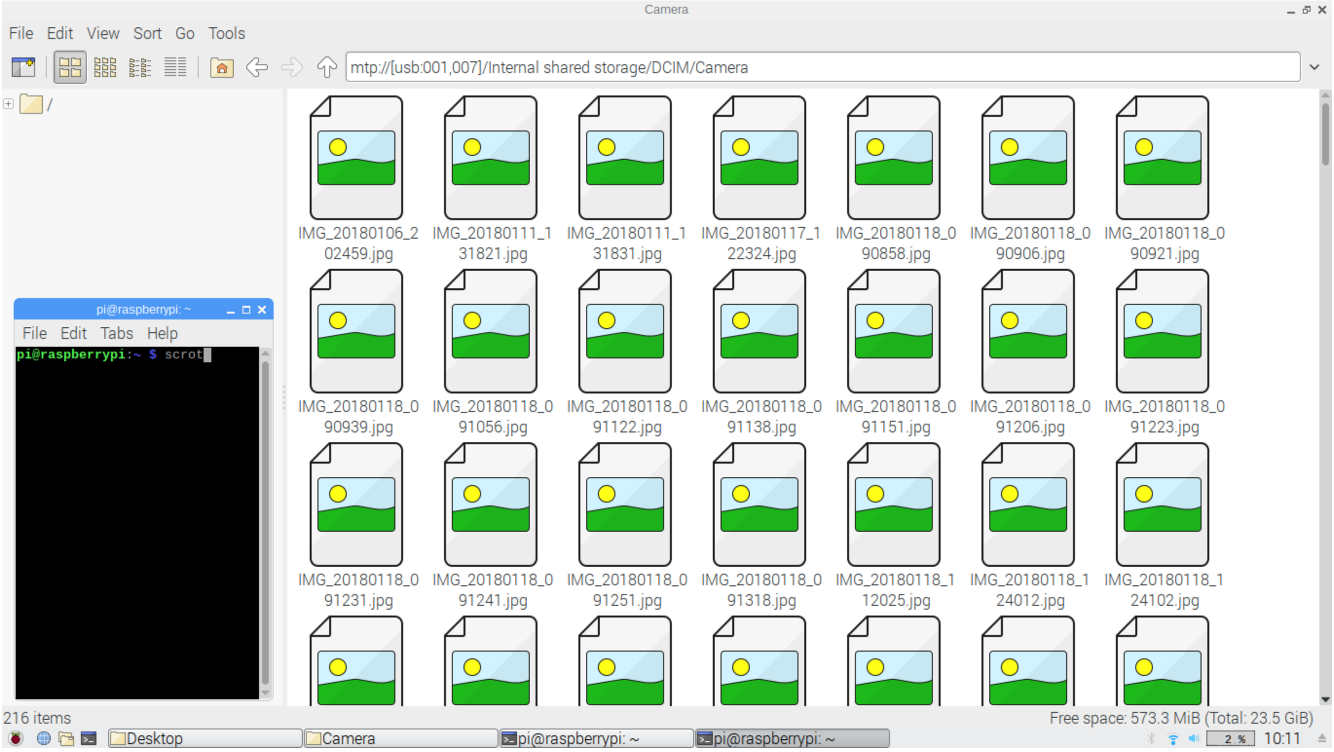


mtp://[usb:001,007]/Internal shared storage

57 items

Free space: 573.3 MiB (Total: 23.5 GiB)

```
pi@raspberrypi:~ $ scrot
```



mtp://[usb:001,007]/Internal shared storage/DCIM/Camera

```
pi@raspberrypi:~ $ scrot
```

IMG_20180106_202459.jpg
IMG_20180111_131821.jpg
IMG_20180111_131831.jpg
IMG_20180117_122324.jpg
IMG_20180118_090858.jpg
IMG_20180118_090906.jpg
IMG_20180118_090921.jpg
IMG_20180118_090939.jpg
IMG_20180118_091056.jpg
IMG_20180118_091122.jpg
IMG_20180118_091138.jpg
IMG_20180118_091151.jpg
IMG_20180118_091206.jpg
IMG_20180118_091223.jpg
IMG_20180118_091231.jpg
IMG_20180118_091241.jpg
IMG_20180118_091251.jpg
IMG_20180118_091318.jpg
IMG_20180118_112025.jpg
IMG_20180118_124012.jpg
IMG_20180118_124102.jpg

216 items

Free space: 573.3 MiB (Total: 23.5 GiB)

2b)

Aim : to display the location of the raspberry pi through the GPS module.

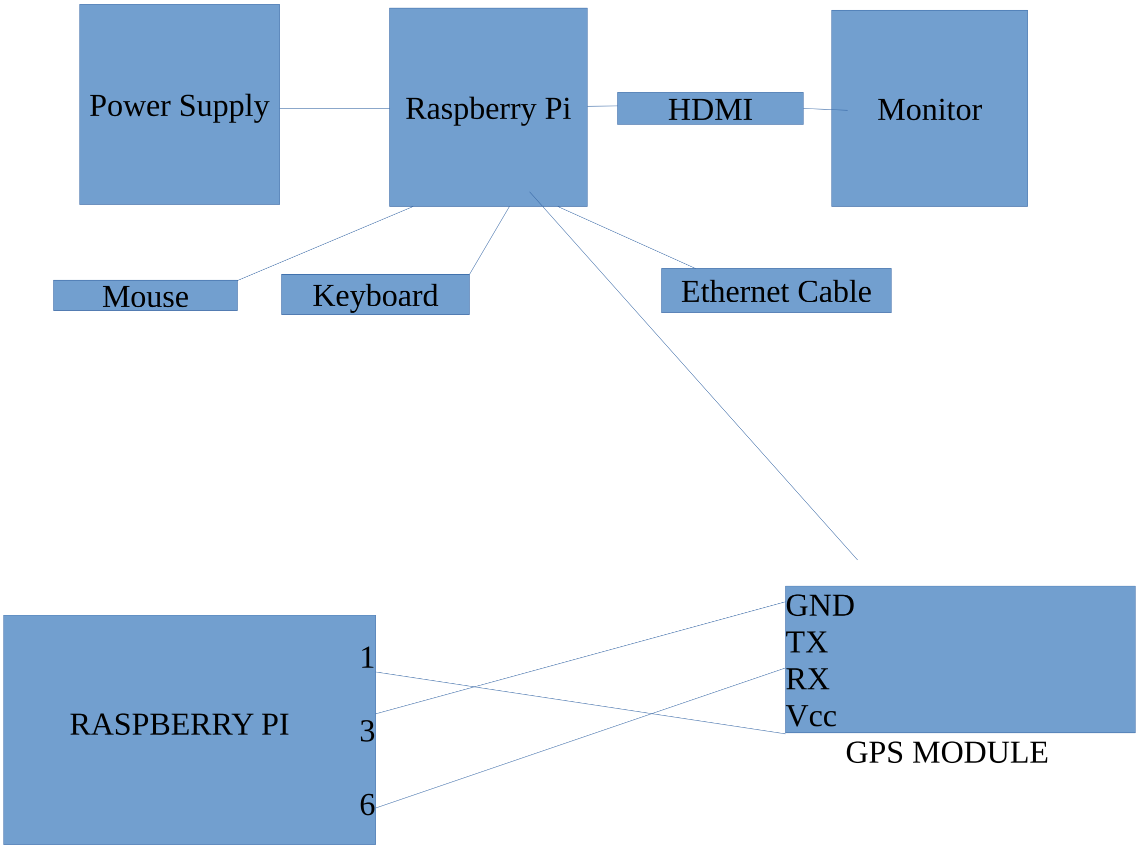
Equipments: Raspberry pi 3, Ethernet Cable, HDMI adapter, Keyboard, Mouse, Monitor, USB cable, smart phone, Raspberry pi Adapter, GPS module, Wire connections.

Description: Raspberry Pi is a minuture computer that performs the basic functions of a real time PC. It contains the ethernet cable port that used to communicate with the internet. It also contians a wifi module in order to connect to the internet. HDMI adapter allows the raspberry to display it's Opearting system on the monitor. Raspberry Pi has multiple built in 2x2 I/O ports. Raspberry pi runs on Scratch and Python to run the programs in the OS. The RaspOS can be manipulated through the use of basic keyboard and a mouse. Raspberry Pi is a computer powered by the Broadcom BCM2835 system-on-a-chip (SoC). This SoC includes a 32-bit ARM1176JZFS processor, clocked at 700MHz, and a Videocore IV GPU. It also has 256MB of RAM in a POP package above the SoC. The Raspberry Pi is powered by a 5V micro USB AC charger or at least 4 AA batteries.**G**lobal **P**ositioning **S**ystem (GPS) is a satellite-based system that uses satellites and ground stations to measure and compute its position on Earth.
Gps module allows the raspberry pi to send data in the form of longitude and latitude.

Steps/Procedure:

- Acquire Raspberry pi.
- Connect the keyboard and Mouse to the Raspberry pi Device
- Connect the Ethernet cable to the Raspberry pi Device
- Power the Raspberry pi Device the Power supply Adapter.
- Connect the hdmi to the monitor from Raspberry Pi.
- Connect the Power supply to the CPU.
- Turn on the Raspberry pi.
- Connect the Mobile phone to the Rapsberry pi
- Access data from the Data.
- Connect the GPS module to the Raspberry pi
- Execute program
- Acquire the GPS data.

Circuit Diagram:

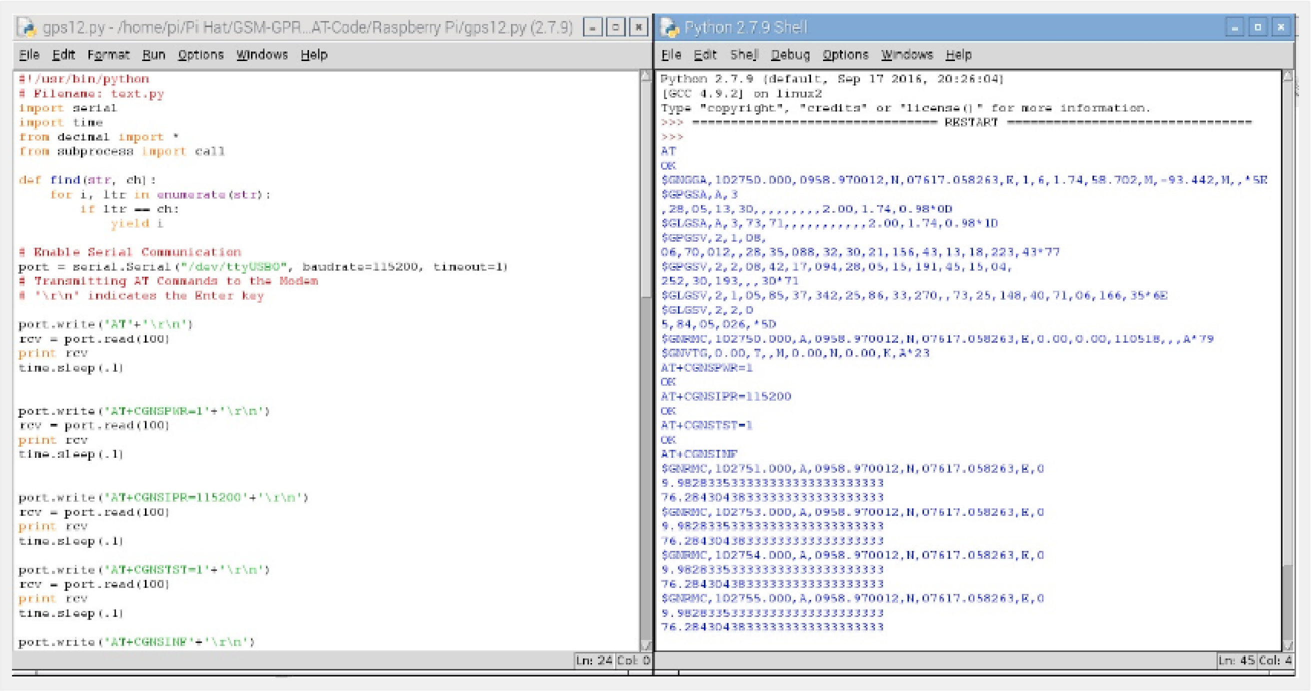


CODE:

```
GPS Interfacing with Raspberry Pi using Pyhton

http://www.electronicwings.com

import serial               #import serial pacakge

from time import sleep

import webbrowser           #import package for opening link in browser

import sys                  #import system package


def GPS_Info():
    global NMEA_buff
    global lat_in_degrees
    global long_in_degrees
    nmea_time = []
    nmea_latitude = []
    nmea_longitude = []
    nmea_time = NMEA_buff[0]                    #extract time from GPGGA string
    nmea_latitude = NMEA_buff[1]                #extract latitude from GPGGA string
    nmea_longitude = NMEA_buff[3]               #extract longitude from GPGGA string

    print("NMEA Time: ", nmea_time,'\n')
    print ("NMEA Latitude:", nmea_latitude,"NMEA Longitude:", nmea_longitude,'\n')

    lat = float(nmea_latitude)                  #convert string into float for calculation
    longi = float(nmea_longitude)               #convertr string into float for calculation

    lat_in_degrees = convert_to_degrees(lat)    #get latitude in degree decimal format
    long_in_degrees = convert_to_degrees(longi) #get longitude in degree decimal format

#convert raw NMEA string into degree decimal format
def convert_to_degrees(raw_value):
    decimal_value = raw_value/100.00
    degrees = int(decimal_value)
    mm_mmmm = (decimal_value - int(decimal_value))/0.6
    position = degrees + mm_mmmm
    position = "%.4f" %(position)
    return position
gpgga_info = "$GPGGA,"
ser = serial.Serial ("/dev/ttyS0", baudrate=9600)              #Open port with baud rate
```

```
GPGGA_buffer = 0

NMEA_buff = 0

lat_in_degrees = 0

long_in_degrees = 0


try:

    while True:

        received_data = (str)(ser.readline())                   #read NMEA string received

        GPGGA_data_available = received_data.find(gpgga_info)   #check for NMEA GPGGA string

        #print(received_data)

        if (GPGGA_data_available>0):

            GPGGA_buffer = received_data.split("$GPGGA,",1)[1]  #store data coming after "$GPGGA,"
string

            NMEA_buff = (GPGGA_buffer.split(','))               #store comma separated data in buffer

            print(NMEA_buff)

            GPS_Info()                                          #get time, latitude, longitude


            print("lat in degrees:", lat_in_degrees," long in degree: ", long_in_degrees, '\n')

            map_link = 'http://maps.google.com/?q=' + lat_in_degrees + ',' + long_in_degrees    #create link
to plot location on Google map

            print("<<<<<<<<press ctrl+c to plot location on google maps>>>>>>\n")               #press ctrl+c
to plot on map and exit

            print("------------------------------------------------------------\n")


except KeyboardInterrupt:

    webbrowser.open(map_link)        #open current position information in google map

    sys.exit(0);
```

Output:


gps12.py - /home/pi/Pi Hat/GSM-GPR...AT-Code/Raspberry Pi/gps12.py (2.7.9)
File Edit Format Run Options Windows Help
Ln: 24 Col: 0
Python 2.7.9 Shell
File Edit Shell Debug Options Windows Help
Ln: 45 Col: 4

```python
#!/usr/bin/python
# Filename: text.py
import serial
import time
from decimal import *
from subprocess import call

def find(str, ch):
    for i, ltr in enumerate(str):
        if ltr == ch:
            yield i

# Enable Serial Communication
port = serial.Serial("/dev/ttyUSB0", baudrate=115200, timeout=1)
# Transmitting AT Commands to the Modem
# '\r\n' indicates the Enter key

port.write('AT'+'\r\n')
rcv = port.read(100)
print rcv
time.sleep(.1)


port.write('AT+CGNSPWR=1'+'\r\n')
rcv = port.read(100)
print rcv
time.sleep(.1)


port.write('AT+CGNSIPR=115200'+'\r\n')
rcv = port.read(100)
print rcv
time.sleep(.1)

port.write('AT+CGNSTST=1'+'\r\n')
rcv = port.read(100)
print rcv
time.sleep(.1)

port.write('AT+CGNSINF'+'\r\n')
```

```
Python 2.7.9 (default, Sep 17 2016, 20:26:04)
[GCC 4.9.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
AT
OK
$GNGGA,102750.000,0958.970012,N,07617.058263,E,1,6,1.74,58.702,M,-93.442,M,,*5E
$GPGSA,A,3
,28,05,13,30,,,,,,,,,2.00,1.74,0.98*0D
$GLGSA,A,3,73,71,,,,,,,,,,,2.00,1.74,0.98*1D
$GPGSV,2,1,08,
06,70,012,,28,35,088,32,30,21,156,43,13,18,223,43*77
$GPGSV,2,2,08,42,17,094,28,05,15,191,45,15,04,
252,30,193,,,30*71
$GLGSV,2,1,05,85,37,342,25,86,33,270,,73,25,148,40,71,06,166,35*6E
$GLGSV,2,2,0
5,84,05,026,*5D
$GNRMC,102750.000,A,0958.970012,N,07617.058263,E,0.00,0.00,110518,,,A*79
$GNVTG,0.00,T,,M,0.00,N,0.00,K,A*23
AT+CGNSPWR=1
OK
AT+CGNSIPR=115200
OK
AT+CGNSTST=1
OK
AT+CGNSINF
$GNRMC,102751.000,A,0958.970012,N,07617.058263,E,0
9.98283353333333333333333333
76.28430438333333333333333333
$GNRMC,102753.000,A,0958.970012,N,07617.058263,E,0
9.98283353333333333333333333
76.28430438333333333333333333
$GNRMC,102754.000,A,0958.970012,N,07617.058263,E,0
9.98283353333333333333333333
76.28430438333333333333333333
$GNRMC,102755.000,A,0958.970012,N,07617.058263,E,0
9.98283353333333333333333333
76.28430438333333333333333333
```