SIM808 GSM GPRS GPS Dev Board IPX SMA with GPS Antenna for Arduino Raspberry Pi

Introduction:

SIM808 GSM GPRS GPS Dev Board IPX SMA with GPS Antenna is a complete Quad-Band GSM/GPRS module, which combines GPS technology for satellite navigation. The compact design, which integrated GPRS and GPS in a SMT package, will significantly save time and make it easier for users to develop GPS enabled applications. Featuring an industry-standard interface and GPS function, it allows variable assets to be tracked at any location and anytime with signal coverage. The module can also be controlled via AT commands.

Specifications:

- 5-18V power supply supported.
- USB port can be very convenient.
- Serial USB TTL compatible for 2.85V, 3.3V and 5v signal.
- GSM, GPRS, GPS {text messaging, data transmission, positioning} triple function.
- Quad-band communications, globally available. , AT command operation, can support SMS, phone, GPRS data, GPS data is automatically output, embedded TCP protocol to support transparent mode and command mode, supports HTTP protocol to support DTMF decoding, support MMS, support recording function, and so on.
- IPX antenna connector.
- Reserved USB port (not solder) available for changing firmware. So you can update for Bluetooth.
- Quad-band 850/900/1800/1900MHz.
- GPRS multi-slot class 10 standard.
- GPRS mobile station class B standard.
- Meet the GSM phase 2/2 + standards:-
  Class 4 (2 W @ 850/900 MHz).
  Class 1 (1 W @ 1800/1900MHz).
- AT command control (GSM 07.07, 07.05 and SIMCom enhanced AT command set).
- SIM Application Toolkit.
- Supply voltage range:
  GPRS: 3.2 ~ 4.8 V.
  GPS: 3.0 ~ 4.5V.
- Low power consumption.
- Size: 30 * 30 * 3.2mm.
- Weight: SIM908: 5.2g.
- Working temperature: -40 ° Cto +85 ° C.
- SMS.
  Point MO and MT.
  SMS Cell Broadcast.
  Text and PDU mode.
- Voice, Three kinds of rate.

Half Rate (HR).

Full-rate (FR).

Enhanced Full Rate (EFR).

- Handsfree function.
- Support echo suppression algorithm.
- GPRS data.
- GPRS class 8/10: Maximum 85.6 kbps (downlink rate).
- Support PBCCH.
- Coding schemes CS 1, 2, 3, 4.
- PPP-stack.
- CSD data.
- CSD up to 14.4 kbps.
- USSD.
- Nontransparent.
- Compatibility: AT cellular command interface.
- GPS Receive Type

  Receive 42 channels

  GPS L1 C / A code,

  High-performance STE engine.
- Sensitivity.

  Tracking: -160 dBm.

  Cold starts: -143 dBm.
- Time-To-First-Fix.

  Cold start: 30s (typ.)

  Hot start: 1s (typ.)
- Accuracy.

  Accurate positioning: <2.5 m CEP.
- Power consumption (GSM idle mode).

  Capture 77mA.

  Tracking 76mA.

Applications:

- Tracker device.
- Controlling or data gathering device via text messages.

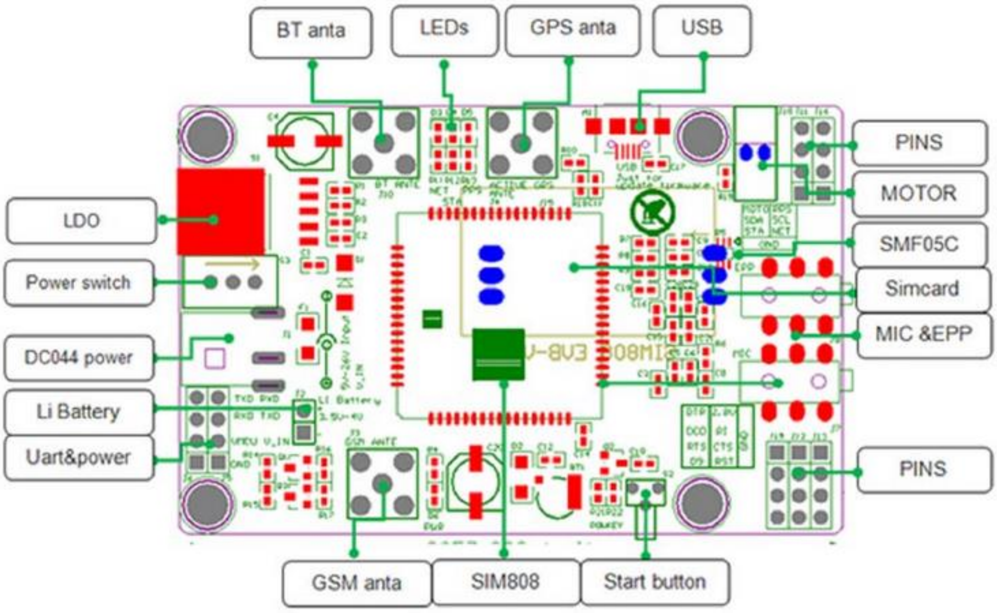
Objectives:

This tutorial will shows you a few simple steps about how to use SIM808 GSM GPRS GPS Dev Board IPX SMA with GPS Antenna by using Arduino Uno. At the end of this tutorial, you will get a result of comparison when the module can detect its location via satellite GPS coordinate.

Components needed:

- Arduino Uno and USB Cable.
- Jumper wires.
- SIM808 GSM GPRS GPS Dev Board IPX SMA with GPS Antenna.
- 2V 1A Power Adapter.

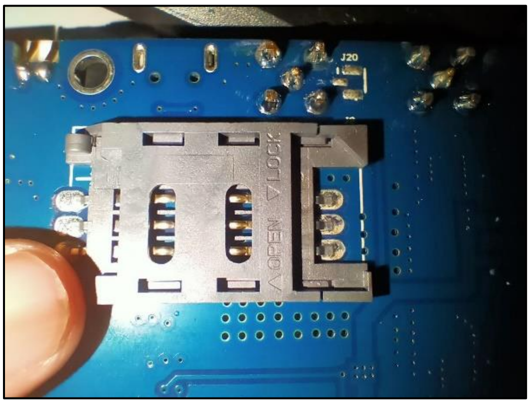
Procedures:



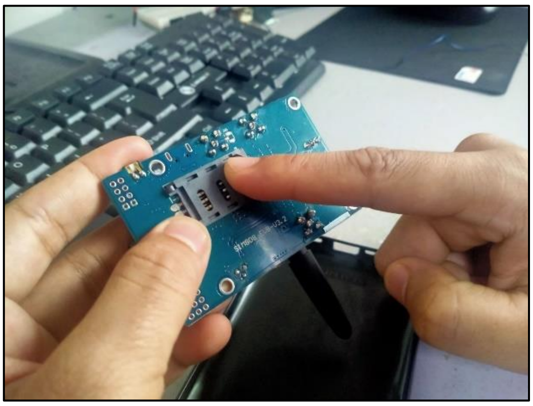
SIM808 GSM GPRS GPS Dev Board IPX SMA with GPS Antenna Schematic

Step 1: Inserting SIM card & wiring connection to use only GSM part of the SIM808 GSM GPRS GPS Dev Board IPX SMA with GPS Antenna.

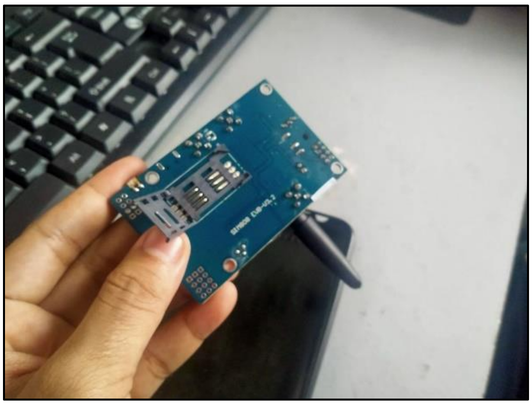
1. Inserting the SIM card into SIM808.



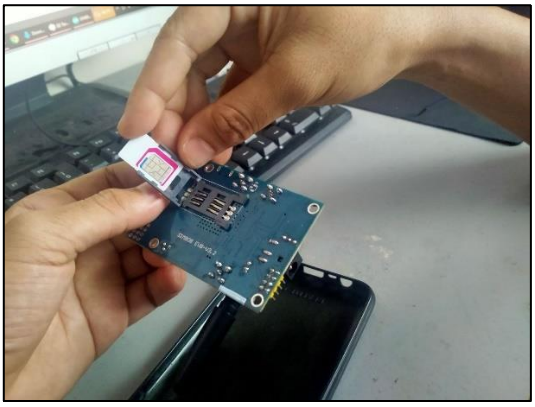
Slide to left to open & slide to right to close.



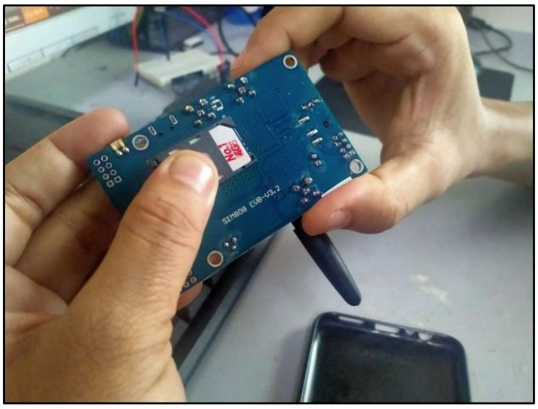
Slide the SIM holder to the left to open.



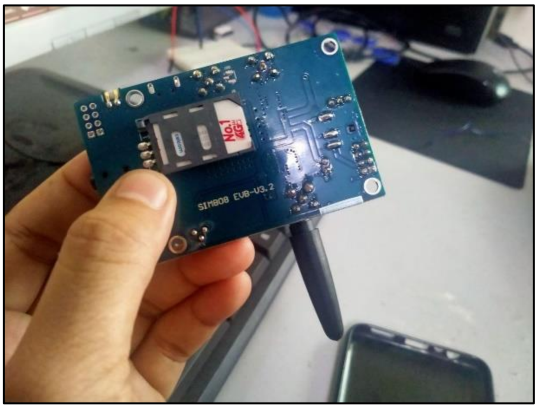
Pull up the SIM holder & then slide in the SIM card into the holder opening.



Slide in the SIM card into correct position.



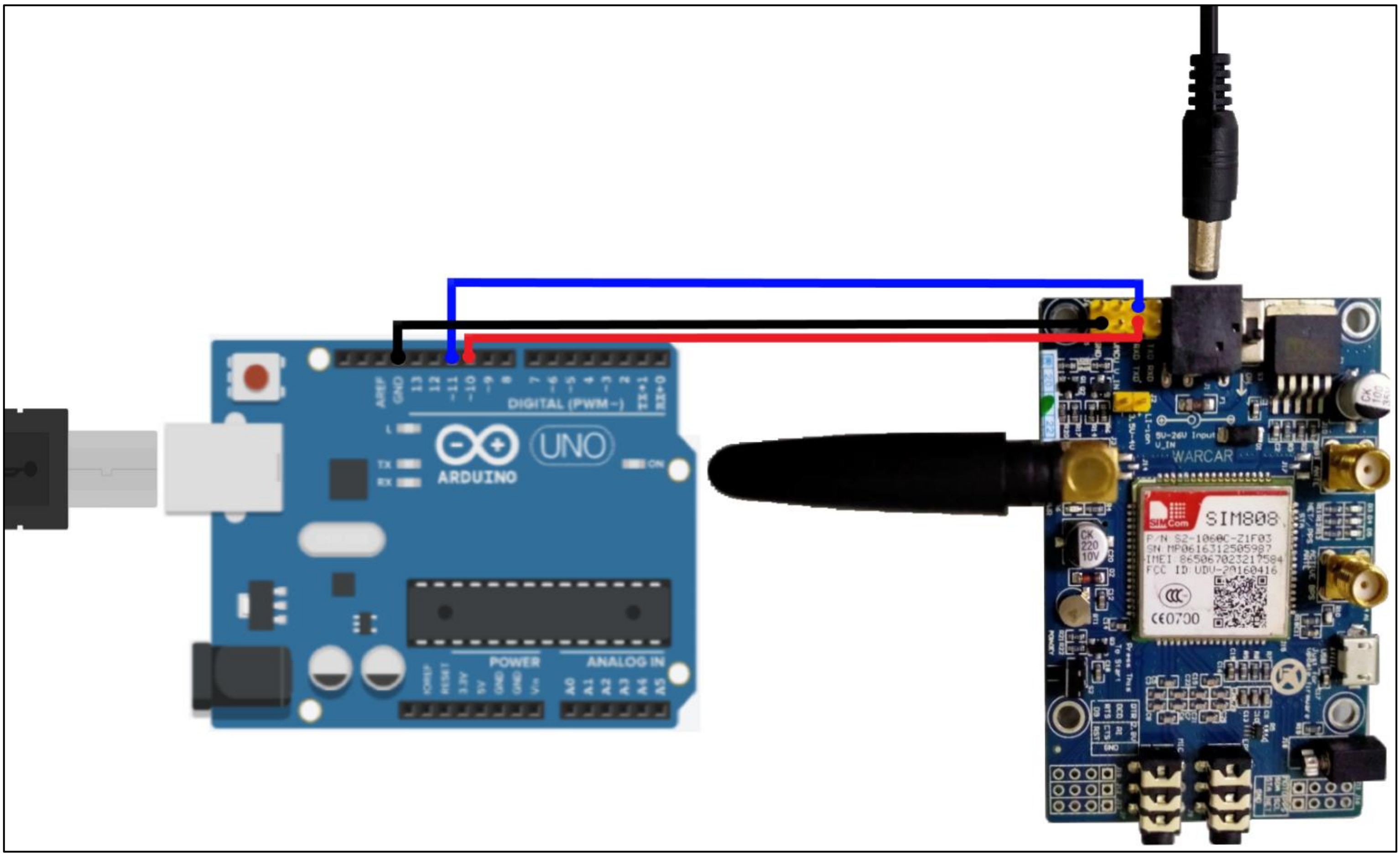
Push down the holder & then slide it to the right to close.



Finished inserting the SIM card.

2. Use jumper wires to connect Arduino Uno to SIM808 like the picture below.

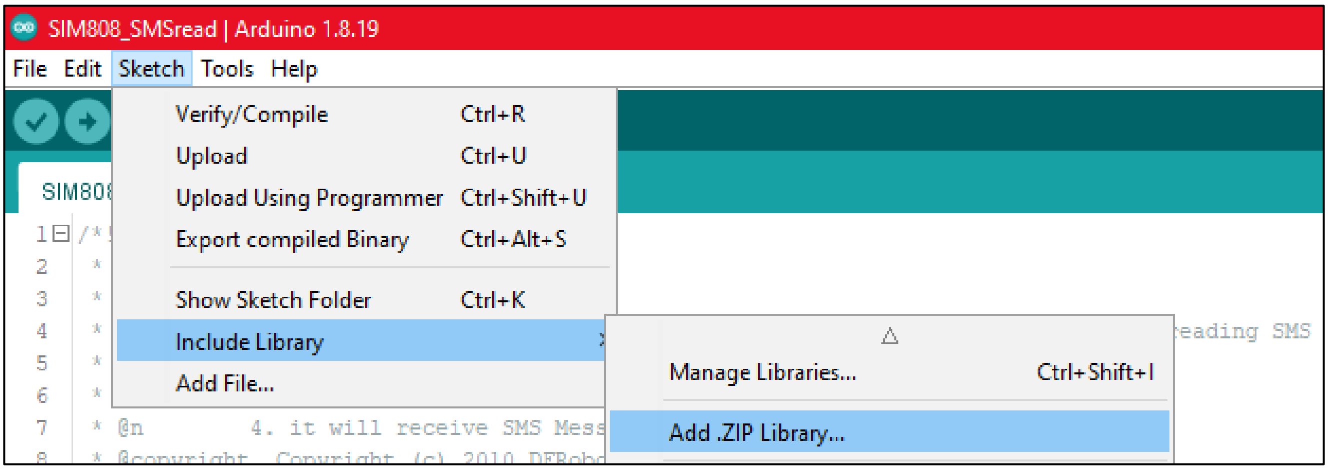
Arduino Uno	SIM808
10	TX
11	RX
GND (GROUND)	GND (GROUND)



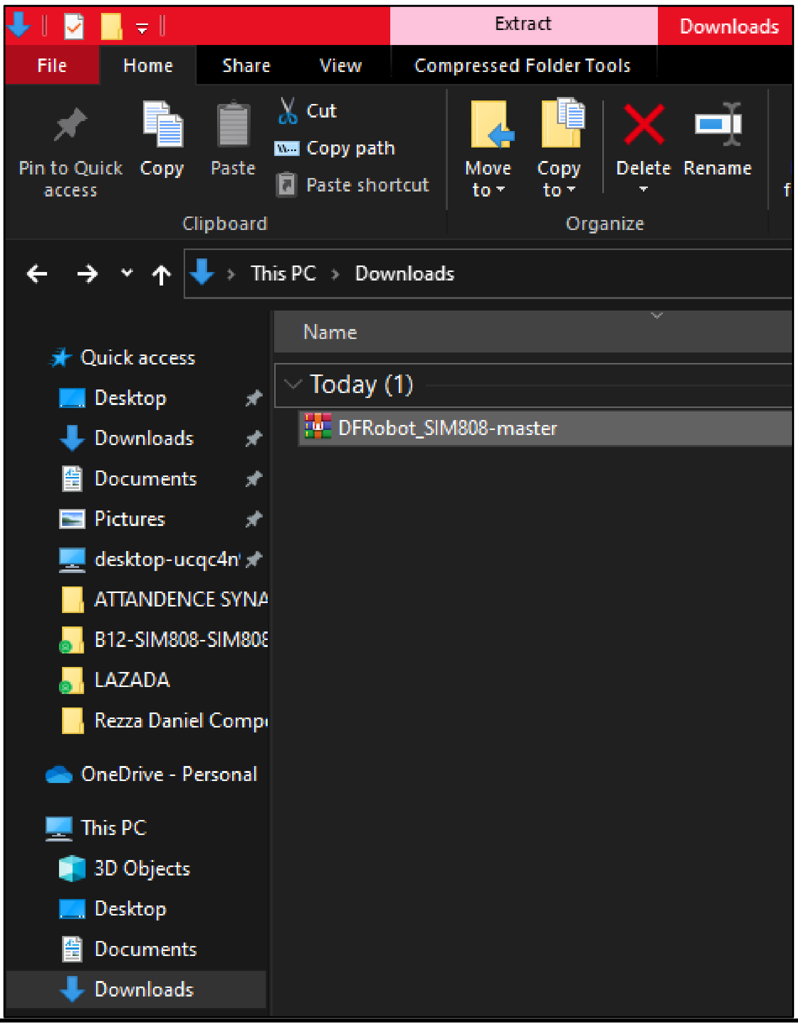
Step 2: Installing SIM808 library into Arduino IDE.

1. Restart Arduino IDE after Board installation complete. Download the SIM808 library (https://github.com/DFRobot/DFRobot_SIM808).

2. Click on **Sketch > Include Library > Add .ZIP Library…**


SIM808_SMSread | Arduino 1.8.19
File Edit Sketch Tools Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...
Manage Libraries... Ctrl+Shift+I
Add .ZIP Library...
SIM808
/*
@n 4. it will receive SMS Mess
@copyright Copyright (c) 2010 DFRobo
eading SMS

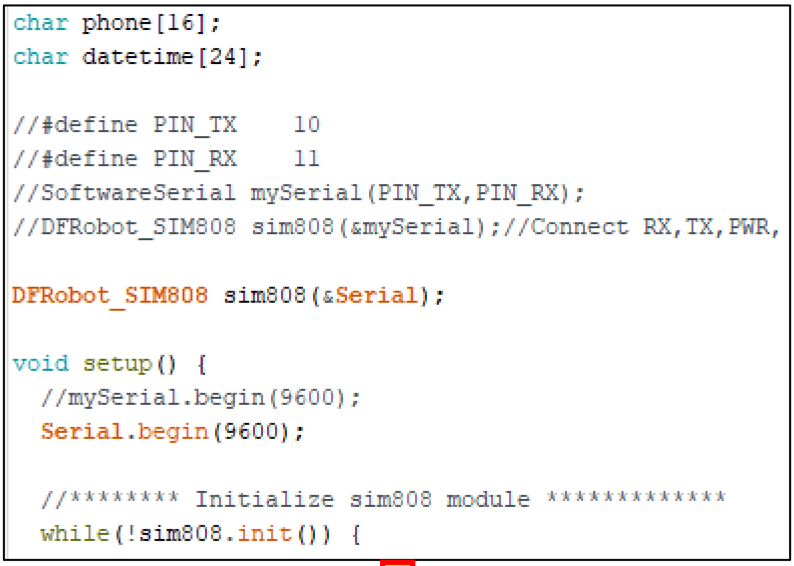
3. New windows will pop-up Navigate & Select downloaded SIM808 library earlier.



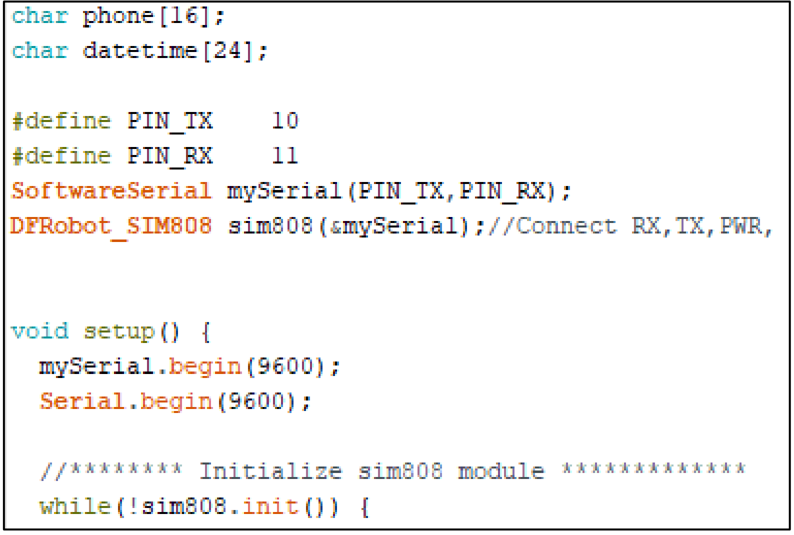
Step 3: Uploading Program into SIM808 GSM GPRS GPS Dev Board IPX SMA with GPS Antenna to test GSM function.

1. On Arduino IDE click on **File > Examples > DFRobot_SIM808 > SIM808_SMSread.**

2. Edit part of the coding to match with the pictures below.

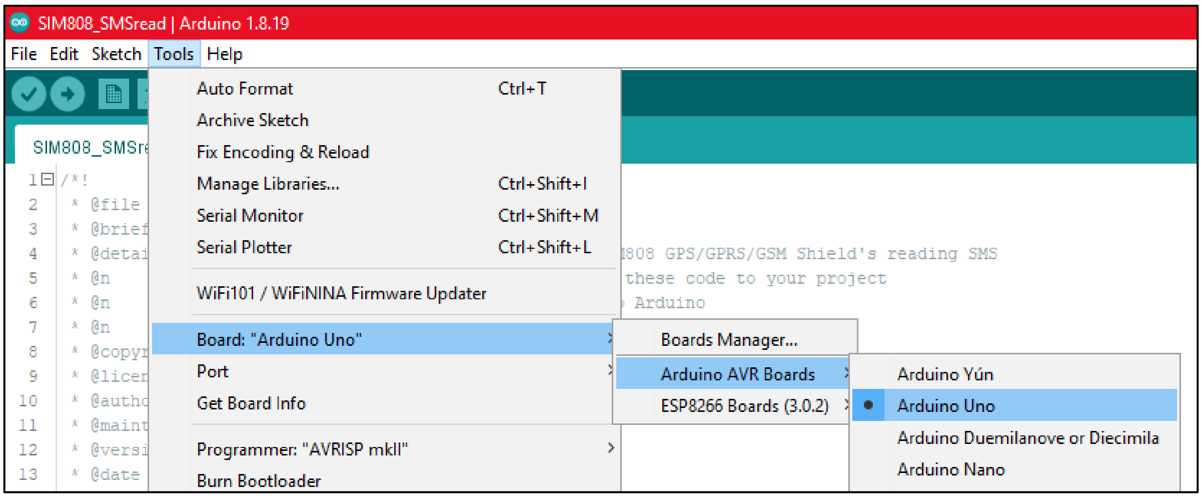


```
char phone[16];
char datetime[24];

//#define PIN_TX    10
//#define PIN_RX    11
//SoftwareSerial mySerial(PIN_TX,PIN_RX);
//DFRobot_SIM808 sim808(&mySerial);//Connect RX,TX,PWR,

DFRobot_SIM808 sim808(&Serial);

void setup() {
  //mySerial.begin(9600);
  Serial.begin(9600);

  //******** Initialize sim808 module *************
  while(!sim808.init()) {
```



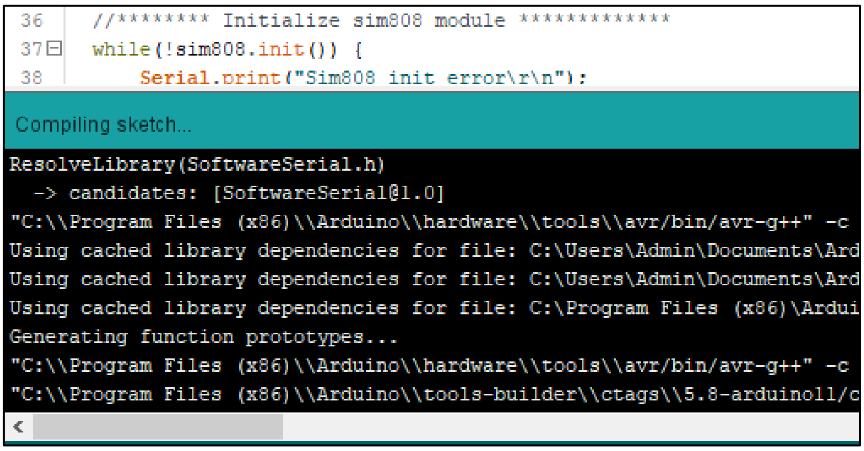
```
char phone[16];
char datetime[24];

#define PIN_TX    10
#define PIN_RX    11
SoftwareSerial mySerial(PIN_TX,PIN_RX);
DFRobot_SIM808 sim808(&mySerial);//Connect RX,TX,PWR,


void setup() {
  mySerial.begin(9600);
  Serial.begin(9600);

  //******** Initialize sim808 module *************
  while(!sim808.init()) {
```

3. Make sure Arduino UNO already connected with PC. Select the right board for Arduino UNO by navigate to Tools > Board: > Arduino AVR Boards > Arduino Uno.



4. After selecting board, make sure to select the right COM PORT for your Arduino Uno.

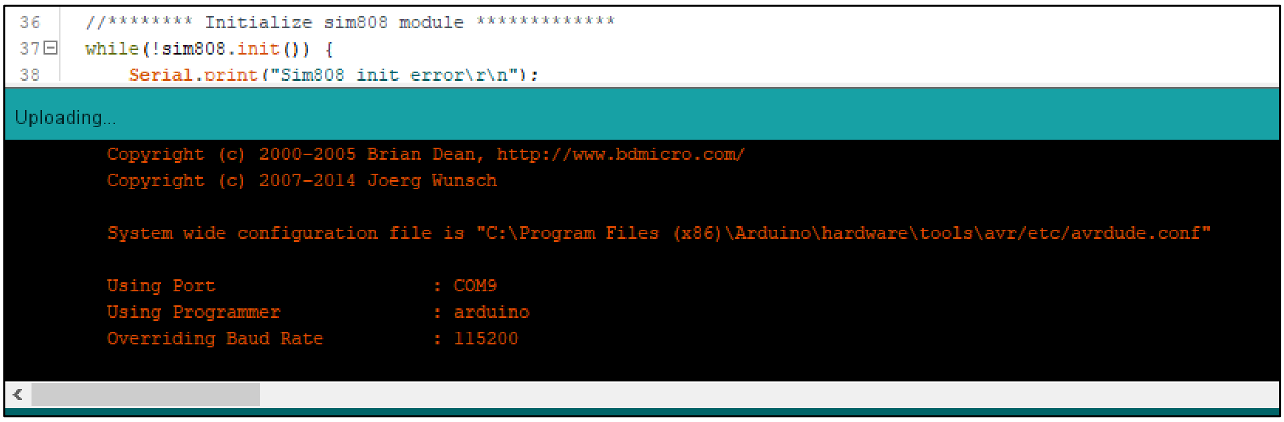
5. After selecting **PORT,** click on **UPLOAD** to start uploading the code into board. Wait until done Uploading message appear.



```
//********* Initialize sim808 module *************
while(!sim808.init()) {
    Serial.print("Sim808 init error\r\n");
```

Compiling sketch...

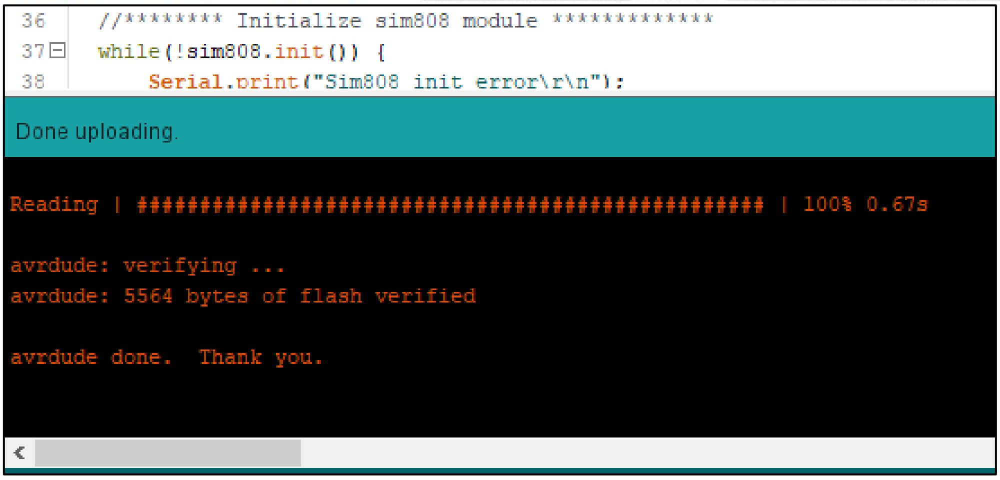
```
ResolveLibrary(SoftwareSerial.h)
  -> candidates: [SoftwareSerial@1.0]
"C:\\Program Files (x86)\\Arduino\\hardware\\tools\\avr/bin/avr-g++" -c
Using cached library dependencies for file: C:\Users\Admin\Documents\Ard
Using cached library dependencies for file: C:\Users\Admin\Documents\Ard
Using cached library dependencies for file: C:\Program Files (x86)\Ardui
Generating function prototypes...
"C:\\Program Files (x86)\\Arduino\\hardware\\tools\\avr/bin/avr-g++" -c
"C:\\Program Files (x86)\\Arduino\\tools-builder\\ctags\\5.8-arduino11/c
```



```
//********* Initialize sim808 module *************
while(!sim808.init()) {
    Serial.print("Sim808 init error\r\n");
```

Uploading...

```
        Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com/
        Copyright (c) 2007-2014 Joerg Wunsch

        System wide configuration file is "C:\Program Files (x86)\Arduino\hardware\tools\avr/etc/avrdude.conf"

        Using Port                    : COM9
        Using Programmer              : arduino
        Overriding Baud Rate          : 115200
```

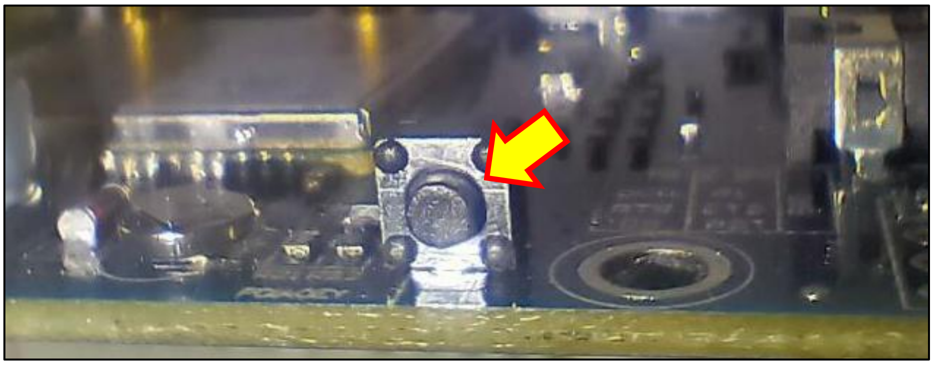


```
//******** Initialize sim808 module *************
while(!sim808.init()) {
    Serial.print("Sim808 init error\r\n");
```

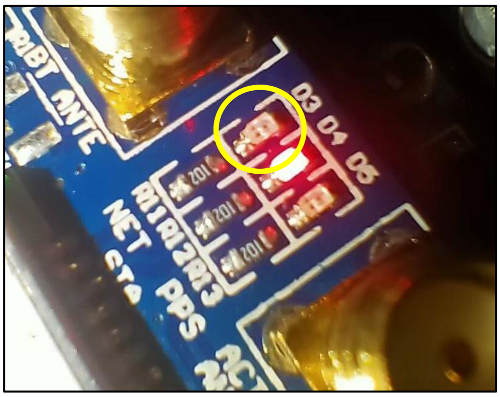
Done uploading.

```
Reading | ################################################## | 100% 0.67s

avrdude: verifying ...
avrdude: 5564 bytes of flash verified

avrdude done.  Thank you.
```

6. Powered on your Arduino Uno & SIM808 and then press the start button on your SIM808 to start.

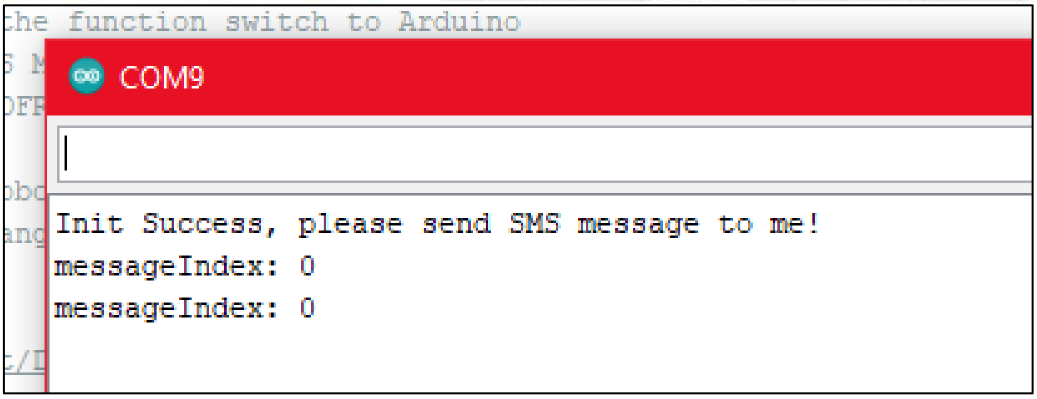


7. Observe the D3 Led. When successfully contact to service provider the LED will blink. It takes 3 seconds for each blink to occur, less than that means it still not able to contact service provider. (**Side note: Sometime it take a while to contact service provider. Depends on the SIM card service provider signal strength**).



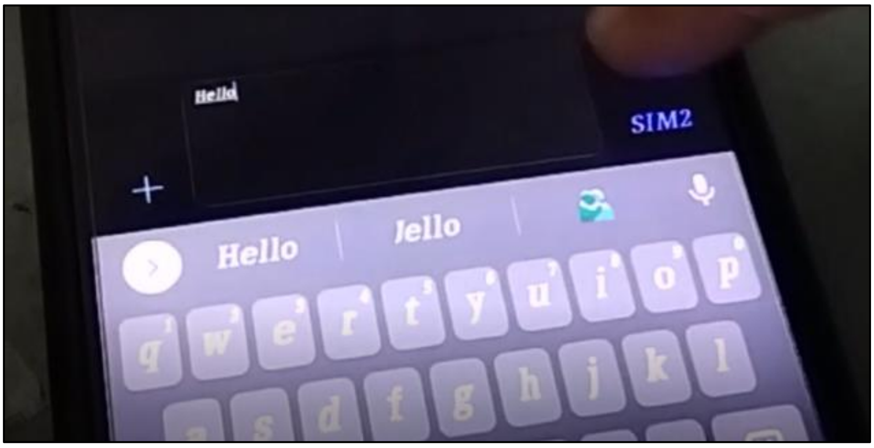
It take 3 seconds for each blink to light up.

8. Open serial monitor in Arduino IDE and select 9600 baud and wait for the output to appear like the picture below.

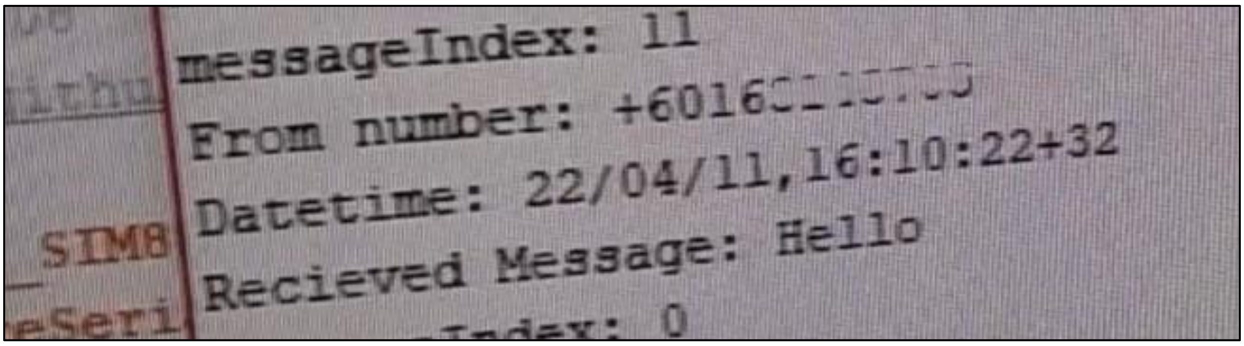


```
Init Success, please send SMS message to me!
messageIndex: 0
messageIndex: 0
```

9. Open a smartphone and then send a simple message to the SIM card numbers used.



10. Then, observe the serial monitor for the message to appear.

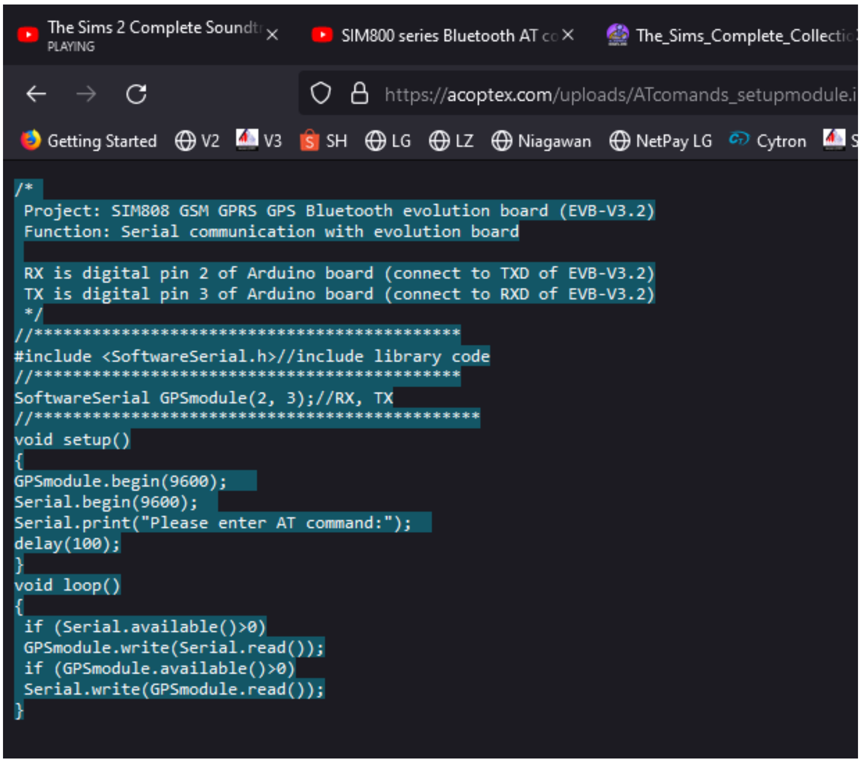


```
messageIndex: 11
From number: +6016[illegible]
Datetime: 22/04/11,16:10:22+32
Recieved Message: Hello
 Index: 0
```

Step 4: Uploading Program into SIM808 GSM GPRS GPS Dev Board IPX SMA with GPS Antenna to test Bluetooth function.

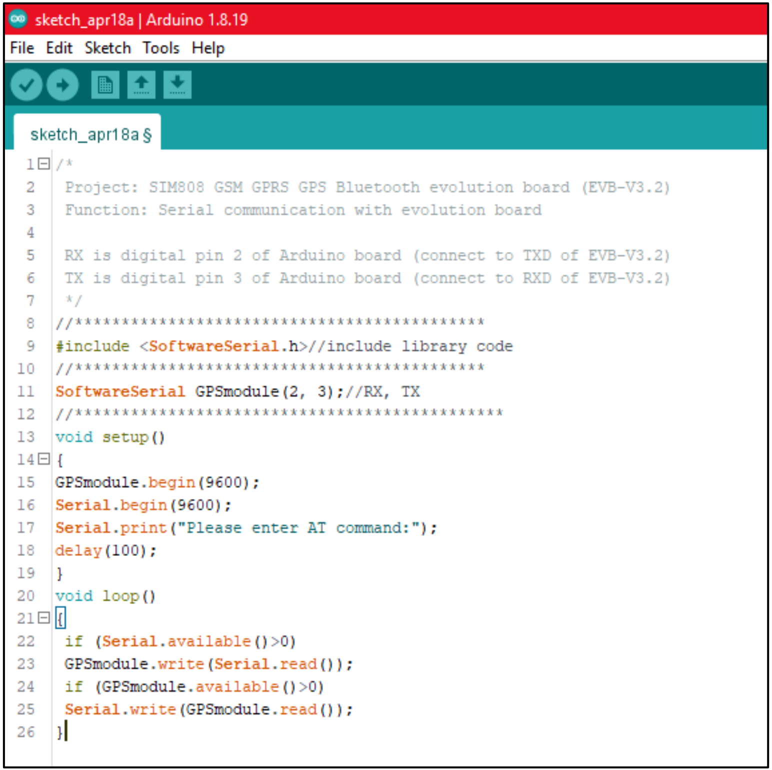
1. To test Bluetooth function, we will use the same exact circuit but with different coding.

2. Go to this link https://acoptex.com/uploads/ATcomands_setupmodule.ino and copy the coding.


The Sims 2 Complete Soundt
PLAYING
SIM800 series Bluetooth AT c
The_Sims_Complete_Collectic
https://acoptex.com/uploads/ATcomands_setupmodule.i
Getting Started
V2
V3
SH
LG
LZ
Niagawan
NetPay LG
Cytron

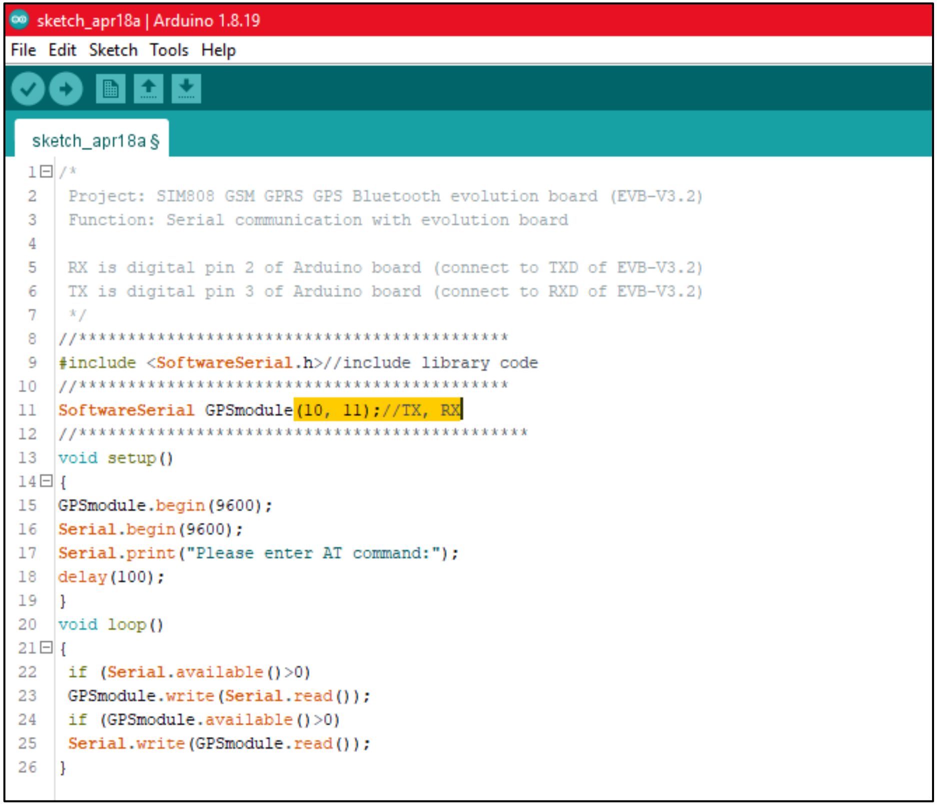
```
/*
 Project: SIM808 GSM GPRS GPS Bluetooth evolution board (EVB-V3.2)
 Function: Serial communication with evolution board

 RX is digital pin 2 of Arduino board (connect to TXD of EVB-V3.2)
 TX is digital pin 3 of Arduino board (connect to RXD of EVB-V3.2)
 */
//*********************************************
#include <SoftwareSerial.h>//include library code
//*********************************************
SoftwareSerial GPSmodule(2, 3);//RX, TX
//***********************************************
void setup()
{
GPSmodule.begin(9600);
Serial.begin(9600);
Serial.print("Please enter AT command:");
delay(100);
}
void loop()
{
 if (Serial.available()>0)
 GPSmodule.write(Serial.read());
 if (GPSmodule.available()>0)
 Serial.write(GPSmodule.read());
}
```

3. Create a new sketch on Arduino IDE and paste the coding into it.



```
/*
 Project: SIM808 GSM GPRS GPS Bluetooth evolution board (EVB-V3.2)
 Function: Serial communication with evolution board

 RX is digital pin 2 of Arduino board (connect to TXD of EVB-V3.2)
 TX is digital pin 3 of Arduino board (connect to RXD of EVB-V3.2)
 */
//*********************************************
#include <SoftwareSerial.h>//include library code
//*********************************************
SoftwareSerial GPSmodule(2, 3);//RX, TX
//***********************************************
void setup()
{
GPSmodule.begin(9600);
Serial.begin(9600);
Serial.print("Please enter AT command:");
delay(100);
}
void loop()
{
 if (Serial.available()>0)
 GPSmodule.write(Serial.read());
 if (GPSmodule.available()>0)
 Serial.write(GPSmodule.read());
}
```

4. Edit a line of the code to be the same as the highlight line of code in the picture below.

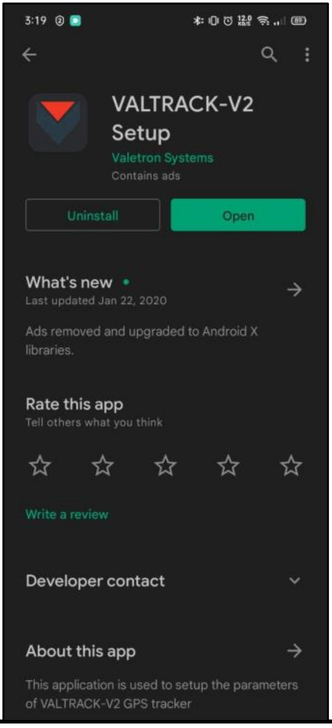


```
/*
 Project: SIM808 GSM GPRS GPS Bluetooth evolution board (EVB-V3.2)
 Function: Serial communication with evolution board

 RX is digital pin 2 of Arduino board (connect to TXD of EVB-V3.2)
 TX is digital pin 3 of Arduino board (connect to RXD of EVB-V3.2)
 */
//*********************************************
#include <SoftwareSerial.h>//include library code
//*********************************************
SoftwareSerial GPSmodule(10, 11);//TX, RX
//***********************************************
void setup()
{
GPSmodule.begin(9600);
Serial.begin(9600);
Serial.print("Please enter AT command:");
delay(100);
}
void loop()
{
 if (Serial.available()>0)
 GPSmodule.write(Serial.read());
 if (GPSmodule.available()>0)
 Serial.write(GPSmodule.read());
}
```

5. When you finish edit the code, upload the new coding into your Arduino Uno.

6. Take out your smartphone and turn on the Bluetooth.

7. Open your app store in your smartphone to download and install this Valtrack app: https://play.google.com/store/apps/details?id=com.valetron.valtrack&hl=en&gl=US.



VALTRACK-V2 Setup

Valetron Systems

Contains ads

What's new •

Last updated Jan 22, 2020

Ads removed and upgraded to Android X libraries.

Rate this app

Tell others what you think

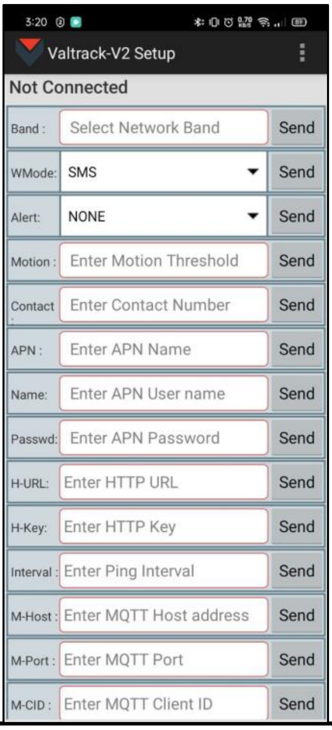
☆ ☆ ☆ ☆ ☆

Write a review

Developer contact

About this app

This application is used to setup the parameters of VALTRACK-V2 GPS tracker



Valtrack-V2 Setup

Not Connected

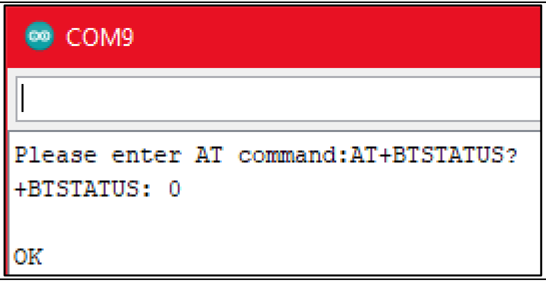
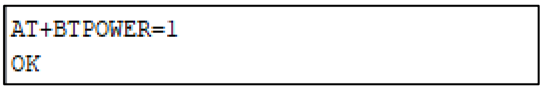
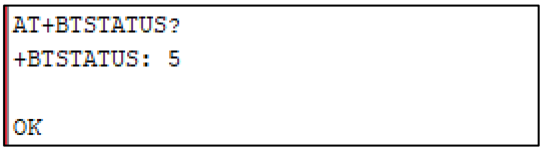
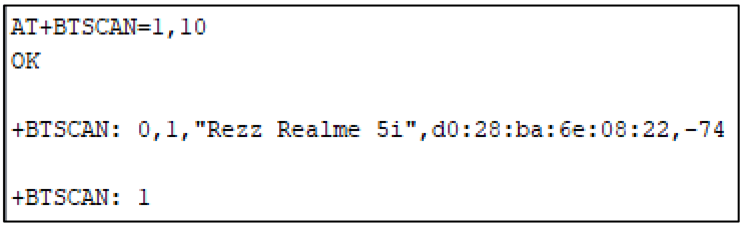
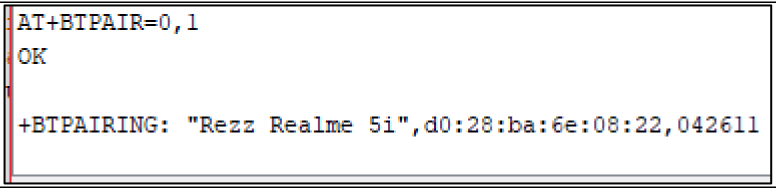
Band :	Select Network Band	Send
WMode:	SMS	Send
Alert:	NONE	Send
Motion :	Enter Motion Threshold	Send
Contact	Enter Contact Number	Send
APN :	Enter APN Name	Send
Name:	Enter APN User name	Send
Passwd:	Enter APN Password	Send
H-URL:	Enter HTTP URL	Send
H-Key:	Enter HTTP Key	Send
Interval :	Enter Ping Interval	Send
M-Host :	Enter MQTT Host address	Send
M-Port :	Enter MQTT Port	Send
M-CID :	Enter MQTT Client ID	Send

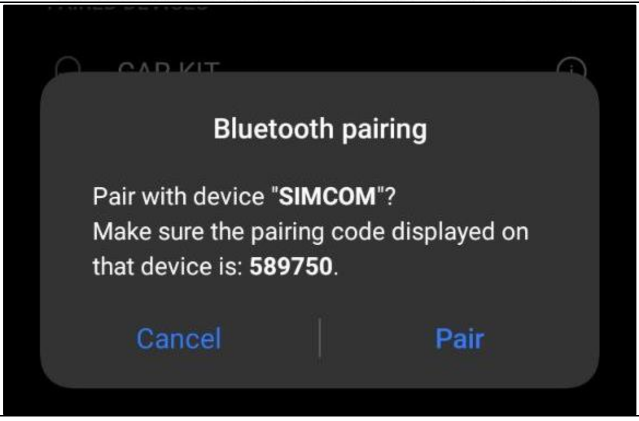
7. Keep the Valtrack app open on your smartphone.

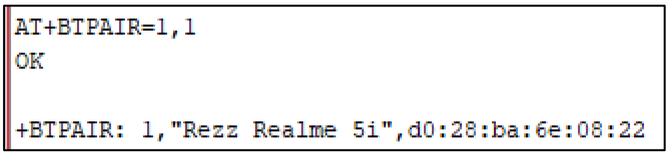
8. Power on your SIM808 and press the start button again to start.

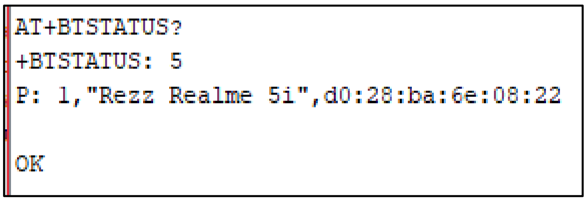
9. Open serial monitor and set it on 9600-baud frequency.

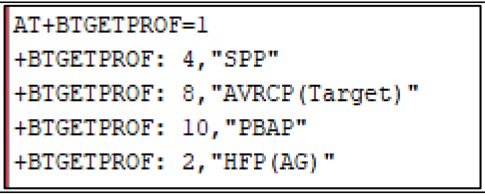
10. Use this step-by-step table AT commands below to set up and test the SIM808 Bluetooth function. (Side note; Use the link provided below for more SIM808 Bluetooth AT commands). https://cdn-shop.adafruit.com/product-files/2637/SIM800+Series_Bluetooth_Application_Note_V1.04.pdf

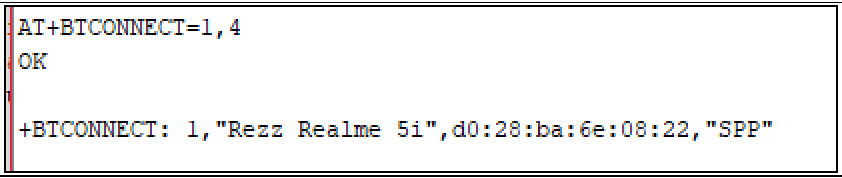
AT+BTSTATUS? To check the Bluetooth status. 0 means that the Bluetooth is turn OFF in SIM808.	 COM9 Please enter AT command:AT+BTSTATUS? +BTSTATUS: 0 OK
AT+BTPOWER=1 To power on Bluetooth in SIM808. Parameter **1** means ON **0** means OFF	 AT+BTPOWER=1 OK
AT+BTSTATUS? To check the Bluetooth status again. 5 means that it is power on and in idle.	 AT+BTSTATUS? +BTSTATUS: 5 OK
AT+BTSCAN=1,10 To scan for any near Bluetooth device available. Parameter for **1** means start and **10** means how many second you want to take to search for a device.	 AT+BTSCAN=1,10 OK +BTSCAN: 0,1,"Rezz Realme 5i",d0:28:ba:6e:08:22,-74 +BTSCAN: 1
AT+BTPAIR=0,1 To pair with my smartphone with a device ID of **0,1**	 AT+BTPAIR=0,1 OK +BTPAIRING: "Rezz Realme 5i",d0:28:ba:6e:08:22,042611

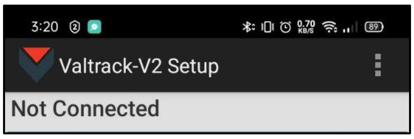
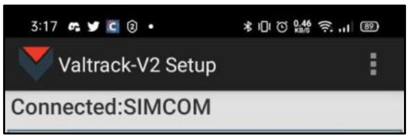
Your smartphone will pop up this message asking permission to be pair with SIMCOM/SIM808. Click pair.	 Bluetooth pairing Pair with device "**SIMCOM**"? Make sure the pairing code displayed on that device is: **589750**.

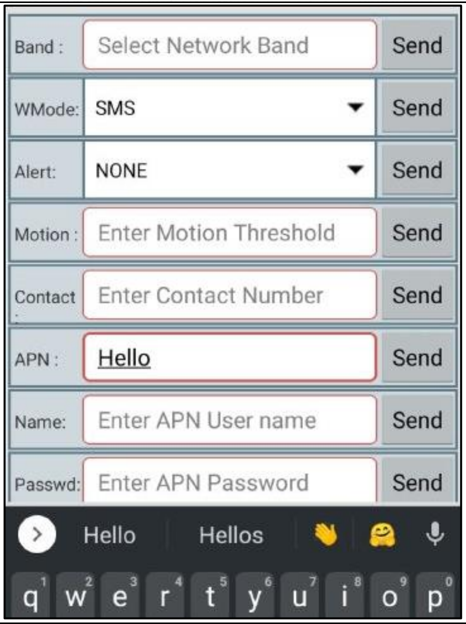
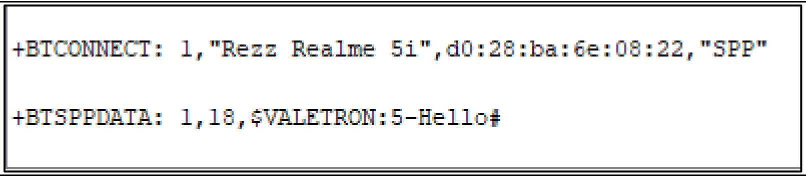
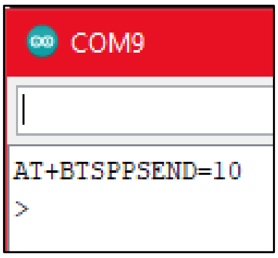
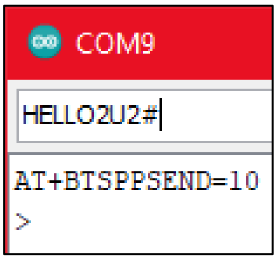
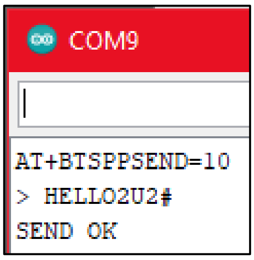
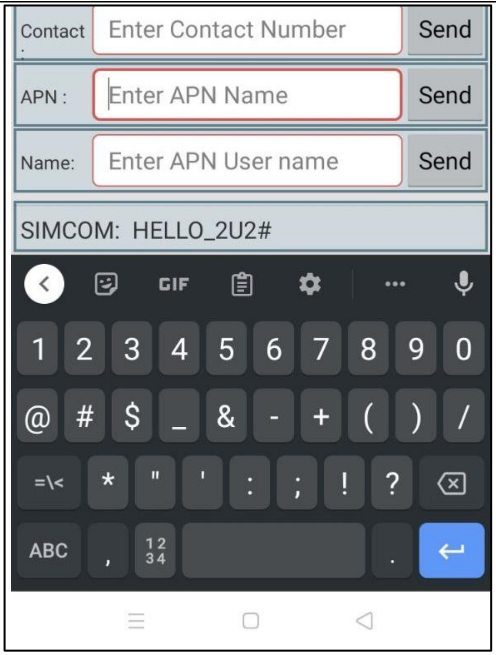
AT+BTPAIR=1,1
To accept the pairing request sent by the smartphone. Parameter 1,1 is to accept pair with my smartphone device ID.
	
	`AT+BTPAIR=1,1`
	`OK`
	
	`+BTPAIR: 1,"Rezz Realme 5i",d0:28:ba:6e:08:22`

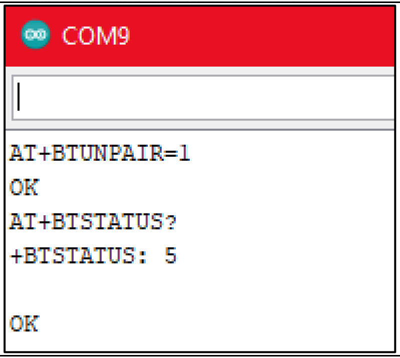
AT+BTSTATUS?
To check the Bluetooth status again. The output show that it already pair with my smartphone.
	
	`AT+BTSTATUS?`
	`+BTSTATUS: 5`
	`P: 1,"Rezz Realme 5i",d0:28:ba:6e:08:22`
	
	`OK`

AT+BTGETPROF=1
To find what profile ID supported. Parameter **1** is my device given ID.
	
	`AT+BTGETPROF=1`
	`+BTGETPROF: 4,"SPP"`
	`+BTGETPROF: 8,"AVRCP(Target)"`
	`+BTGETPROF: 10,"PBAP"`
	`+BTGETPROF: 2,"HFP(AG)"`

AT+BTCONNECT=1,4
To connect using SPP profile label **4**.
	
	`AT+BTCONNECT=1,4`
	`OK`
	
	`+BTCONNECT: 1,"Rezz Realme 5i",d0:28:ba:6e:08:22,"SPP"`

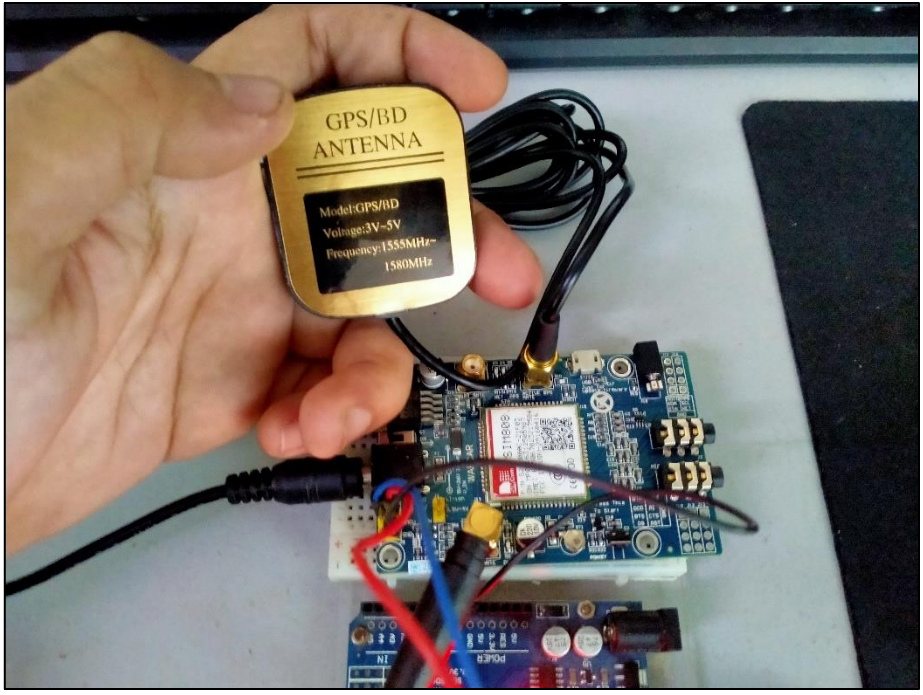
Valtrack app on your smartphone will show that it is connected to SIMCOM/SIM808	 Not Connected  Connected:SIMCOM

Use APN box to send a short message to SIM808	 Band : Select Network Band Send WMode: SMS Send Alert: NONE Send Motion : Enter Motion Threshold Send Contact : Enter Contact Number Send APN : Hello Send Name: Enter APN User name Send Passwd: Enter APN Password Send
The sent message will appear in serial monitor shortly after.	 +BTCONNECT: 1,"Rezz Realme 5i",d0:28:ba:6e:08:22,"SPP" +BTSPPDATA: 1,18,$VALETRON:5-Hello#
AT+BTSPPSEND=10 To send message from SIM808 to your smartphone. Parameter **10** is the length of text you want to use. Remember to put a hashtag (#) at the end of the message to end the sentence to send.	 COM9 AT+BTSPPSEND=10 >  COM9 HELLO2U2# AT+BTSPPSEND=10 >  COM9 AT+BTSPPSEND=10 > HELLO2U2# SEND OK
The sent message will appear in Valtrack app shortly after.	 Contact : Enter Contact Number Send APN : Enter APN Name Send Name: Enter APN User name Send SIMCOM: HELLO_2U2#

AT=BTUNPAIR=1 To un pair SIMCOM/SIM808 from your smartphone. Parameter **1** is your device ID.		 AT+BTUNPAIR=1 OK AT+BTSTATUS? +BTSTATUS: 5 OK

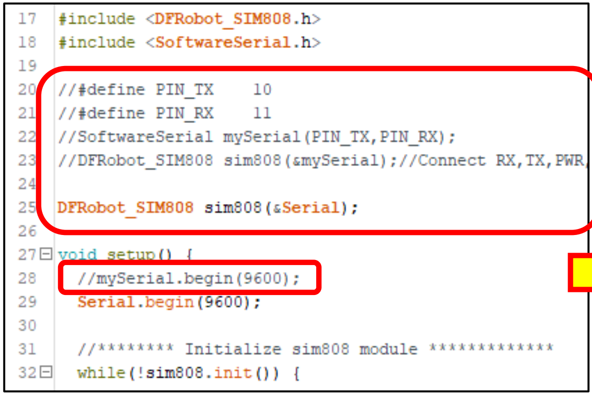
Step 5: Uploading program into SIM808 GSM GPRS GPS Dev Board IPX SMA with GPS Antenna to test GPRS GPS function.

1. To test Bluetooth function, we will use the same exact circuit but with different coding and also attaching the GPS/BD Antenna to SIM808.

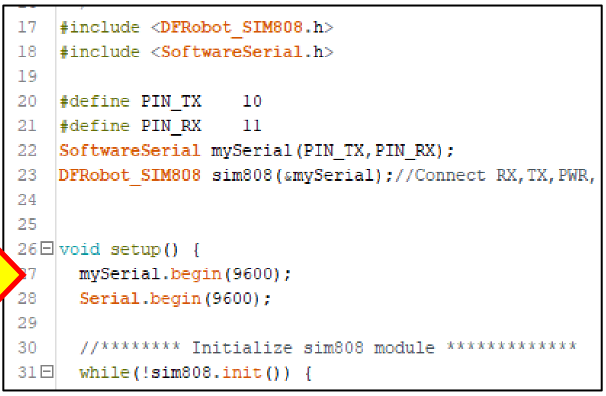


2. On Arduino IDE click on **File > Examples > DFRobot_SIM808 > SIM808_GetGPS.**

3. Edit part of the coding to match with the pictures below.



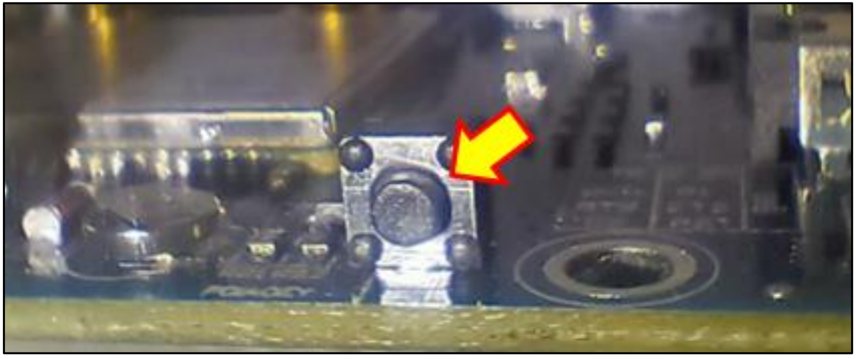
```
#include <DFRobot_SIM808.h>
#include <SoftwareSerial.h>

//#define PIN_TX    10
//#define PIN_RX    11
//SoftwareSerial mySerial(PIN_TX,PIN_RX);
//DFRobot_SIM808 sim808(&mySerial);//Connect RX,TX,PWR,

DFRobot_SIM808 sim808(&Serial);

void setup() {
  //mySerial.begin(9600);
  Serial.begin(9600);

  //******** Initialize sim808 module *************
  while(!sim808.init()) {
```



```
#include <DFRobot_SIM808.h>
#include <SoftwareSerial.h>

#define PIN_TX    10
#define PIN_RX    11
SoftwareSerial mySerial(PIN_TX,PIN_RX);
DFRobot_SIM808 sim808(&mySerial);//Connect RX,TX,PWR,


void setup() {
  mySerial.begin(9600);
  Serial.begin(9600);

  //******** Initialize sim808 module *************
  while(!sim808.init()) {
```

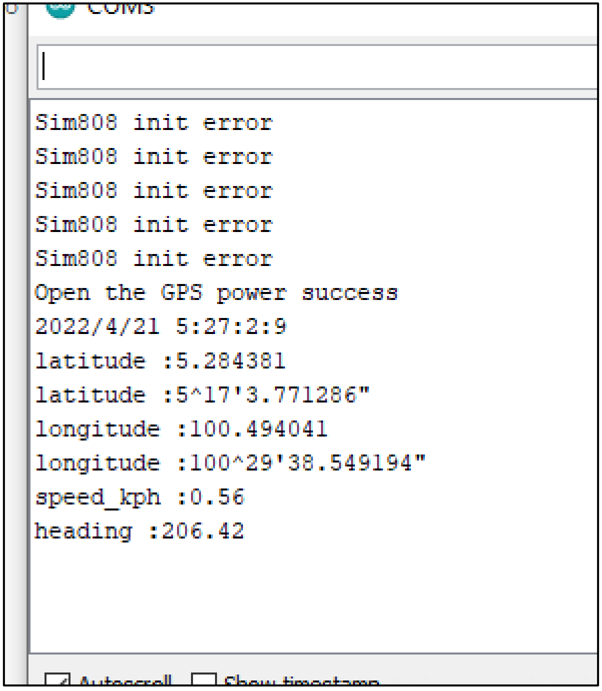
4. Upload the coding into your Arduino Uno.

5. When done uploading the code, power ON your SIM808 and then press the start button.



6. Open up your serial monitor in Arduino IDE and set it to 9600 baud.

7. It takes a few minutes to get the coordinates but it should output like in the picture below.



```
Sim808 init error
Sim808 init error
Sim808 init error
Sim808 init error
Sim808 init error
Open the GPS power success
2022/4/21 5:27:2:9
latitude :5.284381
latitude :5^17'3.771286"
longitude :100.494041
longitude :100^29'38.549194"
speed_kph :0.56
heading :206.42
```

Conclusion:

SIM808 GSM GPRS GPS Dev Board IPX SMA with GPS Antenna, can connect using GSM to mobile service provider to send/receive SMS message and also GPRS GPS function that can be locate to it exact coordinate receiving signal using GPS/BD antenna from satellite. Furthermore, SIM808 can also be connected via Bluetooth to any device that support it to relay any data/information from SIM808 to a third party device for example like a smartphone. To test all this feature is pretty simple just by using the examples from SIM808 library installed in Arduino IDE to test each features separately.