Grove - GPS

This Grove - GPS module is a cost-efficient and field-programmable gadget armed with a SIM28 (u-blox 6 is the old version) and serial communication configuration. It

features 22 tracking / 66 acquisition channels GPS receiver. The sensitivity of tracking and acquisition both reach up to -160dBm, making it a great choice for personal navigation projects and location services, as well as an outstanding one among products of the same price class.

Get One Now

[http://www.seeedstudio.com/depot/grove-gps-p-959.html]

Version

Product Version	Changes	Released Date
Grove - GPS V1.2	Initial	Oct 2015

Features

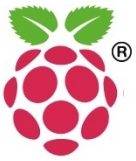
- Supports NMEA and u-blox 6 protocols. (Till Jan,10 2014, after that SIM28 instead)
- Low power consumption
- Baud rates configurable
- Grove compatible interface

Tip
More details about Grove modules please refer to Grove System [http://wiki.seeedstudio.com/Grove_System/]

Specifications

Parameter	Range/Value
Input Voltage	3.3⁄5V
BaudRate	4800 - 57600(u-blox version)
BaudRate	9600 - 115200(SIM28 version)
Default BaudRate	9600

Platforms Supported

Arduino	Raspberry Pi	BeagleBone	Wio	LinkIt ONE
				

Caution
The platforms mentioned above as supported is/are an indication of the module's software or theoritical compatibility. We only provide software library or code examples for Arduino platform in most cases. It is not possible to provide software library / demo code for all possible MCU platforms. Hence, users have to write their own software library.

Getting Started

Note
If this is the first time you work with Arduino, we firmly recommend you to see Getting Started with Arduino [http://wiki.seeedstudio.com/Getting_Started_with_Arduino/] before the start.

Play With Arduino

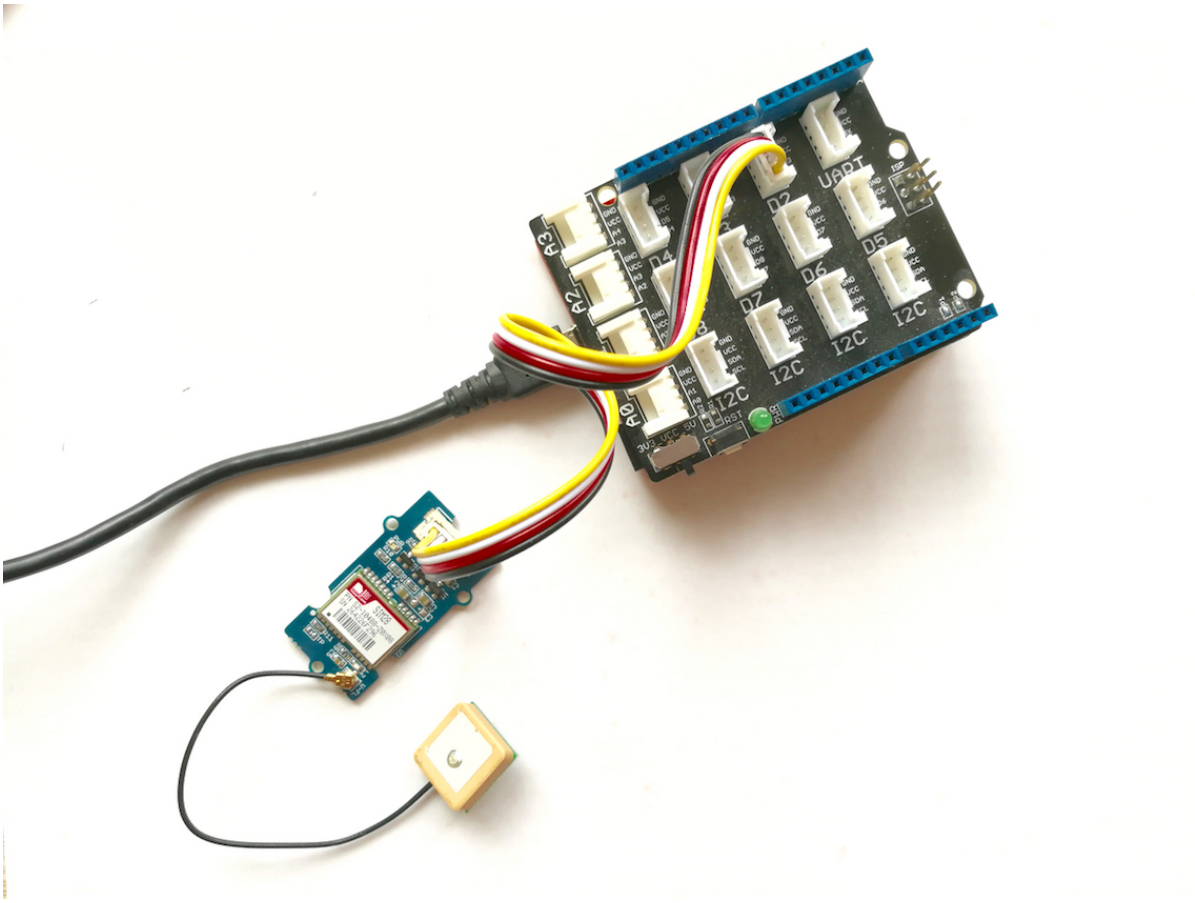
This sample simply reads from the GPS by using software serial and sends it back on the serial port.

Hardware

- **Step 1.** Prepare the below stuffs:

Seeeduino V4.2	Base Shield	Grove - GPS
		
Get One Now [http://www.seeedstudio.com/Seeeduino-V4.2-p-2517.html]	Get One Now [https://www.seeedstudio.com/Base-Shield-V2-p-1378.html]	Get One Now [https://www.s GPS-p-959.htm

- **Step 2.** Connect Grove - GPS to port **D2** of Grove-Base Shield.
- **Step 3.** Plug Grove - Base Shield into Seeeduino.
- **Step 4.** Connect Seeeduino to PC via a USB cable.



Note
If we don't have Grove Base Shield, We also can directly connect Grove - GPS to Seeeduino as below.

Seeeduino	Grove - GPS
5V	Red
GND	Black
D3	White
D2	Yellow

Software

Note
Please note that the u-center software is for windows only.

- **Step 1.** Install u-center [https://www.u-blox.com/en/product/u-center-windows] software.
- **Step 2.** Copy the code into Arduino IDE and upload. If you do not know how to upload the code, please check how to upload code [http://wiki.seeedstudio.com/Upload_Code/].

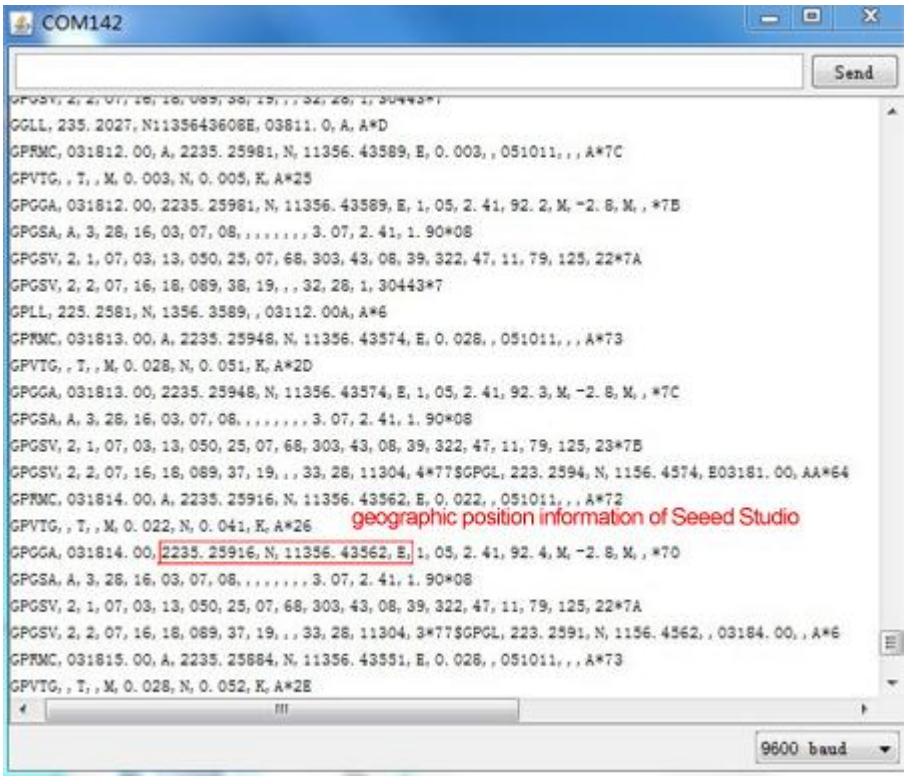
```
#include <SoftwareSerial.h>
SoftwareSerial SoftSerial(2, 3);
unsigned char buffer[64];                   // buffer array for dat
int count=0;                                // counter for buffer a
void setup()
{
    SoftSerial.begin(9600);                 // the SoftSerial baud
    Serial.begin(9600);                     // the Serial port of A
}

void loop()
{
    if (SoftSerial.available())                     // if date is c
    {
        while(SoftSerial.available())               // reading data
        {
            buffer[count++]=SoftSerial.read();      // writing data
            if(count == 64)break;
        }
        Serial.write(buffer,count);                 // if no data t
        clearBufferArray();                         // call clearBu
        count = 0;                                  // set counter
    }
    if (Serial.available())                 // if data is available
    SoftSerial.write(Serial.read());        // write it to the Soft
}


void clearBufferArray()                     // function to clear bu
{
    for (int i=0; i<count;i++)
    {
        buffer[i]=NULL;
```

```
    }                       // clear all index of array with commanc
}
```

- **Step 3.** Open U-center.
- **Step 4.** Click Receiver -> Port and select the COM port that the Arduino is using.
- **Step 5.** Click Receiver -> Baudrate and make sure 9600 is selected.
- **Step 6.** Click View -> Text Console and you should get a window that will stream NMEA data.
- **Step 7.** Open the serial monitor,You can see as show below:



We also can view data in Google Earth:

- **Step 1.** Click File -> Database Export -> Google Earth KML
- **Step 2.** This should launch Google Earth with the history that was captured by u-center.
- **Step 3.** Alternatively, data can be recorded by pressing the red circle on the toolbar which will then ask where you want to save the record.

- **Step 4.** When we have captured enough data, click the black square to stop recording.
- **Step 5.** We can then convert the .ubx file generated to KML by using uploading the ubx file to GPSVisualizer [http://www.gpsvisualizer.com/].

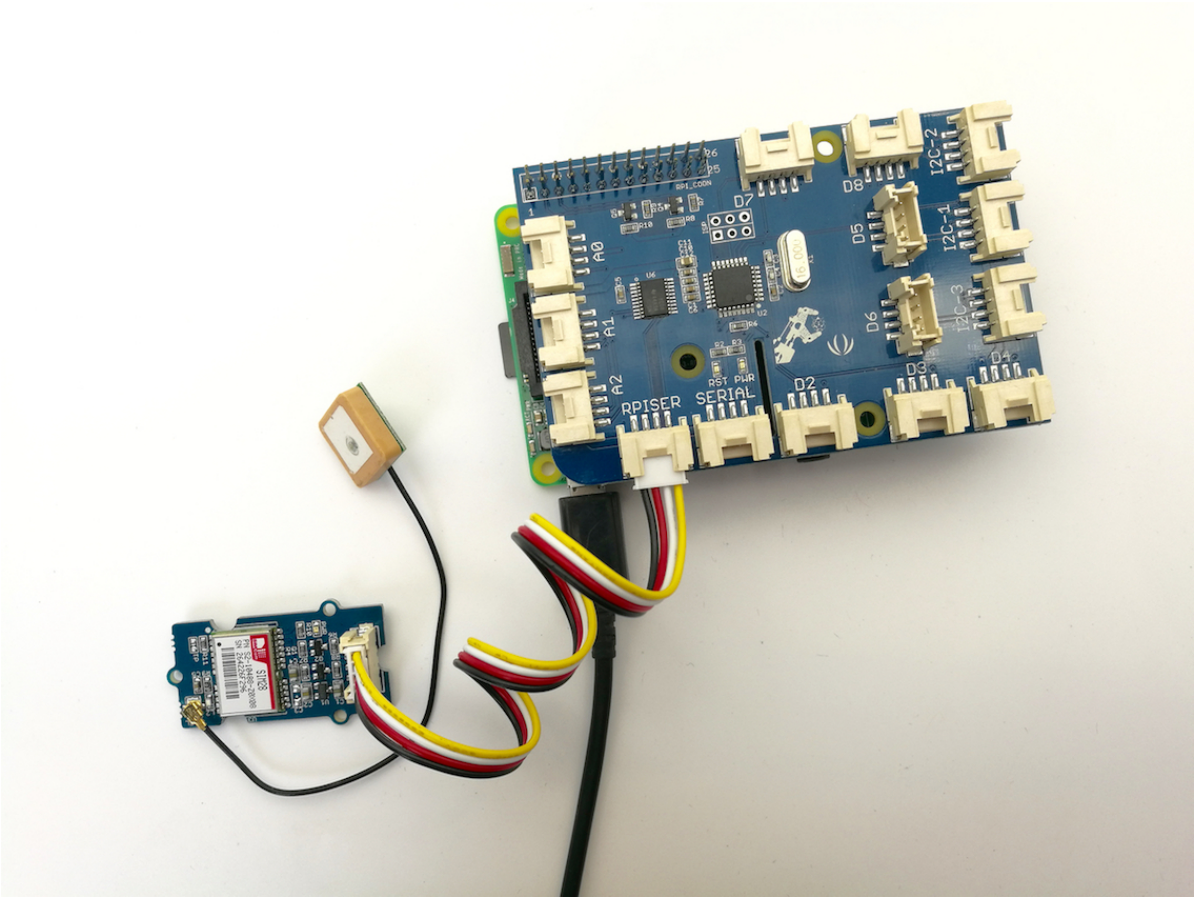
Play With Raspberry Pi

Hardware

- **Step 1.** Prepare the below stuffs:

Raspberry pi	GrovePi_Plus	Grove -
		
Get One Now [https://www.seeedstudio.com/Raspberry-Pi-3-Model-B-p-2625.html]	Get One Now [https://www.seeedstudio.com/GrovePi%2B-p-2241.html]	Get On [http:// gps-p-9

- **Step 2.** Plug the GrovePi_Plus into Raspberry.
- **Step 3.** Plug the sensor to grovepi+ socket **RPISER** by using a grove cable.
- **Step 4.** Connect the Raspberry to PC through USB cable.



Software

- **Step 1.** Follow Setting Software [https://www.dexterindustries.com/GrovePi/get-started-with-the-grovepi/setting-software/] to configure the development environment.
- **Step 2.** Navigate to the demos' directory:

```
cd ~
git clone https://github.com/DexterInd/GrovePi.git
cd GrovePi/Software/Python/grove_gps
```

- **Step 3.** To see the code

```
nano grove_gps_data.py   # "Ctrl+x" to exit #
```

```
import serial, time
```

```
import smbus
import math
import RPi.GPIO as GPIO
import struct
import sys
#import ir_receiver_check

enable_debug=1
enable_save_to_file=0

#if ir_receiver_check.check_ir():
#   print("Disable IR receiver before continuing")
#   exit()

ser = serial.Serial('/dev/ttyAMA0',  9600, timeout = 0) #Open the
ser.flush()

def cleanstr(in_str):
    out_str = "".join([c for c in in_str if c in "0123456789.-" ])
    if len(out_str)==0:
        out_str = "-1"
    return out_str

def safefloat(in_str):
    try:
        out_str = float(in_str)
    except ValueError:
        out_str = -1.0
    return out_str

class GPS:
    #The GPS module used is a Grove GPS module http://www.seeedstu
    inp=[]
    # Refer to SIM28 NMEA spec file http://www.seeedstudio.com/wik
    GGA=[]


    #Read data from the GPS
    def read(self):
        while True:
            GPS.inp=ser.readline()
            if GPS.inp[:6] =='$GPGGA': # GGA data , packet 1, has
                break
            time.sleep(0.1)     #without the cmd program will cras
        try:
```

```
            ind=GPS.inp.index('$GPGGA',5,len(GPS.inp))  #Sometimes
            GPS.inp=GPS.inp[ind:]
        except ValueError:
            print ("")
        GPS.GGA=GPS.inp.split(",")  #Split the stream into indivic
        return [GPS.GGA]

    #Split the data into individual elements
    def vals(self):
        if enable_debug:
            print(GPS.GGA)

        time=GPS.GGA[1]

        if GPS.GGA[2]=='':  # latitude. Technically a float
            lat =-1.0
        else:
            lat=safefloat(cleanstr(GPS.GGA[2]))

        if GPS.GGA[3]=='':  # this should be either N or S
            lat_ns=""
        else:
            lat_ns=str(GPS.GGA[3])

        if  GPS.GGA[4]=='':  # longitude. Technically a float
            long=-1.0
        else:
            long=safefloat(cleanstr(GPS.GGA[4]))

        if  GPS.GGA[5]=='': # this should be either W or E
            long_ew=""
        else:
            long_ew=str(GPS.GGA[5])

        fix=int(cleanstr(GPS.GGA[6]))
        sats=int(cleanstr(GPS.GGA[7]))

        if  GPS.GGA[9]=='':
            alt=-1.0
        else:
            # change to str instead of float
            # 27"1 seems to be a valid value
            alt=str(GPS.GGA[9])
        return [time,fix,sats,alt,lat,lat_ns,long,long_ew]

```

```
    # Convert to decimal degrees
    def decimal_degrees(self, raw_degrees):
        try:
            degrees = float(raw_degrees) // 100
            d = float(raw_degrees) % 100 / 60
            return degrees + d
        except:
            return raw_degrees


if __name__ == "__main__":
    g=GPS()
    if enable_save_to_file:
        f=open("gps_data.csv",'w')  #Open file to log the data
        f.write("name,latitude,longitude\n")    #Write the header
    ind=0
    while True:
        time.sleep(0.01)
        try:
            x=g.read()  #Read from GPS
            [t,fix,sats,alt,lat,lat_ns,longitude,long_ew]=g.vals()

            # Convert to decimal degrees
            if lat !=-1.0:
                lat = g.decimal_degrees(safefloat(lat))
                if lat_ns == "S":
                    lat = -lat

            if longitude !=-1.0:
                longitude = g.decimal_degrees(safefloat(longitude)
                if long_ew == "W":
                    longitude = -longitude

            # print ("Time:",t,"Fix status:",fix,"Sats in view:",s
            try:
                print("Time\t\t: %s\nFix status\t: %d\nSats in vie
            except:
                print("Time\t\t: %s\nFix status\t: %s\nSats in vie

            s=str(t)+","+str(safefloat(lat)/100)+","+str(safefloat

            if enable_save_to_file:
                f.write(s)  #Save to file
            time.sleep(2)
        except IndexError:
```

```
            print ("Unable to read")
        except KeyboardInterrupt:
            if enable_save_to_file:
                f.close()
            print ("Exiting")
            sys.exit(0)
```

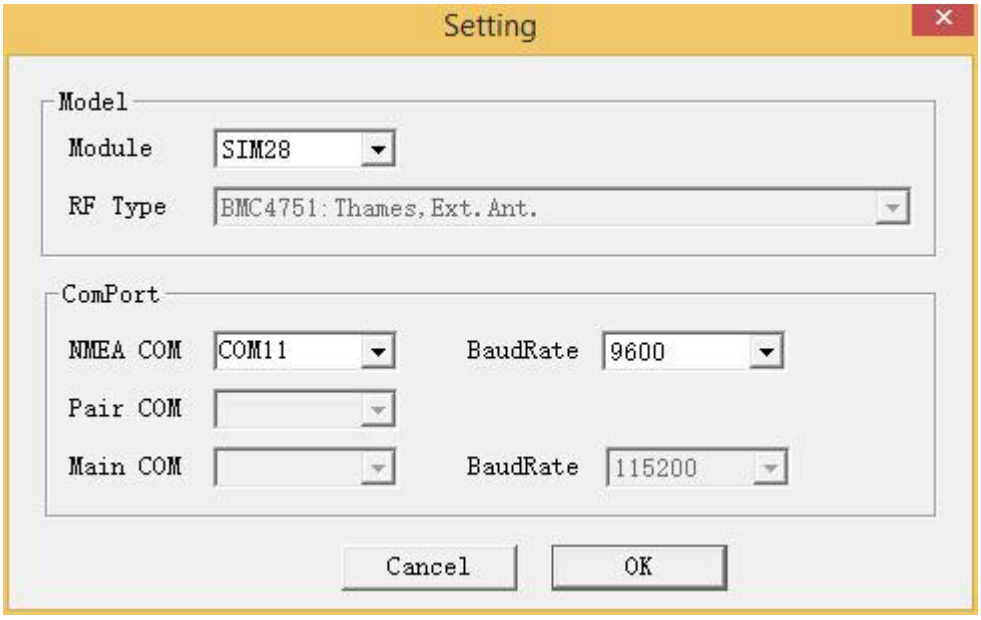
- **Step 4.** Run the demo.

```
sudo python grove_gps_data.py
```

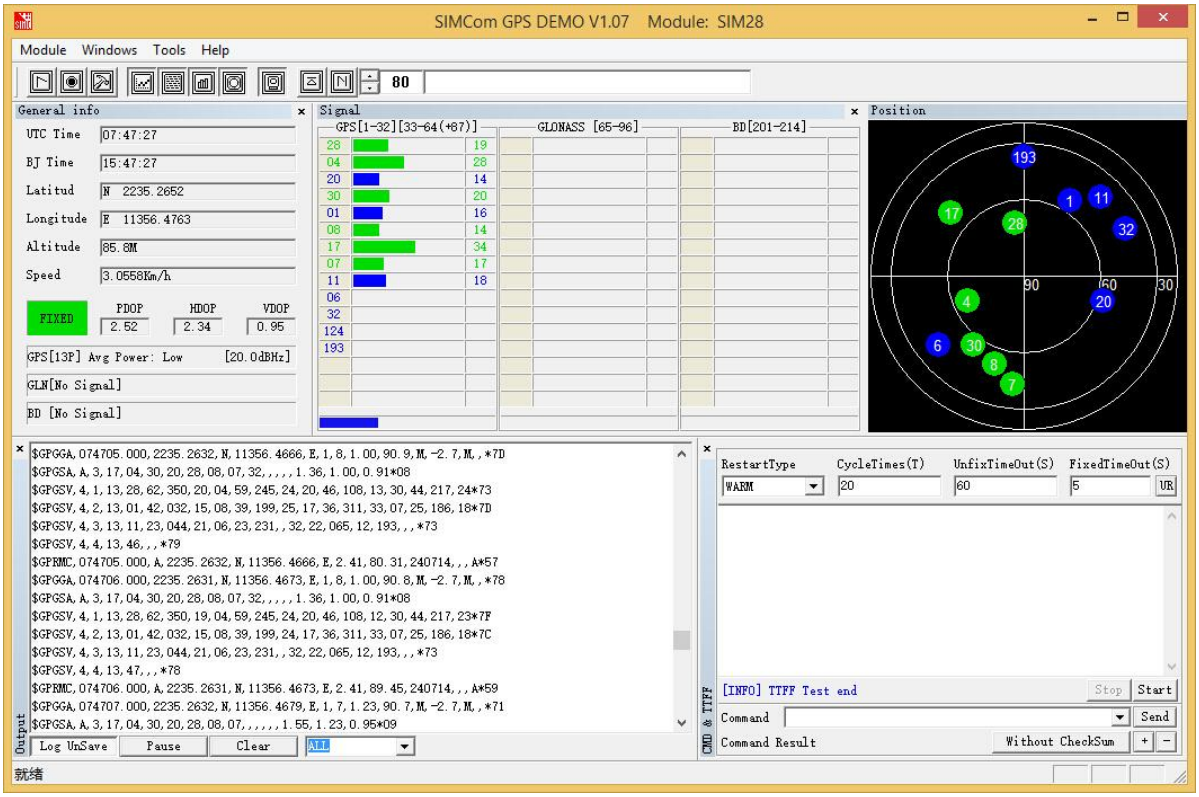
Note
GPS is better used outdoors. It is recommended to put your raspberry pi outside the window or any place outdoors.

SIM28 module Note:

- **Step 1.** Grove-GPS has change the module as SIM28 which the same footprint as origin version.
- **Step 2.** We should use "SIMCom GPS DEMO" [https://raw.githubusercontent.com/SeeedDocument/Grove-GPS/master/res/SIMCom_GPS_DEMO_V1.07.zip] tools to receive SIM28 module data.
- **Step 3.** Open SIMCom_GPS_DEMO tools, go to Module->properties->module->select SIM28.
- **Step 4.** SIMCom_GPS_DEMO_V1.07 is for Windows only.



- **Step 5.** Open SIMCom_GPS_DEMO tools, go to Module->connect. Select the serial port which the GPS module used.



Resources

- **[Eagle]** Grove-GPS Eagle File [https://raw.githubusercontent.com/SeeedDocument/Grove-GPS/master/res/GPS.zip]
- **[PDF]** GPS Schematic(PDF) [https://raw.githubusercontent.com/SeeedDocument/Grove-GPS/master/res/GPS.pdf]
- **[Datasheet]** E-1612-UB Datasheet [https://raw.githubusercontent.com/SeeedDocument/Grove-GPS/master/res/E-1612-UB_Datasheets_Sheet.pdf]
- **[Datasheet]** U-Blox6 Receiver Description Protocol Spec [https://raw.githubusercontent.com/SeeedDocument/Grove-GPS/master/res/U-blox-6-Receiver-Description-Including-Protocol-Specification.zip]
- **[Software]** U-Blox u-center GPS evaluation software [https://www.u-blox.com/en/product/u-center-windows]
- **[Document]** SIM28_DATA_File [https://raw.githubusercontent.com/SeeedDocument/Grove-GPS/master/res/SIM28_DATA_File.zip]
- **[Document]** SIMCom_GPS_DEMO_V1.07 [https://raw.githubusercontent.com/SeeedDocument/Grove-GPS/master/res/SIMCom_GPS_DEMO_V1.07.zip]

Projects

Project of GPS/GPRS Tracker: In this new project we will present our GPS Tracker connected using the GPRS technique and MQTT protocol.

(https://www.hackster.io/OHAlgerie/project-of-gps-gprs-tracker-36c425)

Tech Support

Please submit any technical issue into our forum [http://forum.seeedstudio.com/] or drop mail to techsupport@seeed.cc [mailto:techsupport@seeed.cc].