Grove - GPS

Release date： 9/20/2015

Version： 1.0

Wiki: http://www.seeedstudio.com/wiki/Grove_-_GPS

Bazaar: http://www.seeedstudio.com/depot/Grove-GPS-p-959.html

Document Revision History

Revision	Date	Author	Description
1.0	Sep 21, 2015	Victor.He	Create file

Contents

Disclaimer

For physical injuries and possessions loss caused by those reasons which are not related to product quality, such as operating without following manual guide, natural disasters or force majeure, we take no responsibility for that.

Under the supervision of Seeed Technology Inc., this manual has been compiled and published which covered the latest product description and specification. The content of this manual is subject to change without notice.

Copyright

1. Introduction

This Grove - GPS module is a cost-efficient and field-programmable gadget armed with a SIM28 (U-blox 6 is old version) and serial communication configuration. It features 22 tracking / 66 acquisition channel 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.

2. Features

- Input Voltage: 3.3/5V
- BaudRate: 4800 - 57600(u-blox version)
- BaudRate: 9600 - 115200(SIM28 version)
- Default BaudRate: 9600
- Supports NMEA and U-Blox 6 protocols. (Jan,10 2014 before, after that SIM28 instead)
- Low power consumption
- Baud rates configurable
- Grove compatible interface

3. Usage

3.1 With Arduino

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

- Connect the Grove-GPS to Digital I/O 2 on the Grove - Base Shield using a Grove Universal 4 pin cable.
- Upload the code below. Please click here if you do not know how to upload.

```
// link between the computer and the SoftSerial Shield
//at 9600 bps 8-N-1
//Computer is connected to Hardware UART
//SoftSerial Shield is connected to the Software UART:D2&D3

#include <SoftwareSerial.h>

SoftwareSerial SoftSerial(2, 3);
unsigned char buffer[64];                   // buffer array for data receive
over serial port
int count=0;                                // counter for buffer array

void setup()
{
    SoftSerial.begin(9600);                 // the SoftSerial baud rate
    Serial.begin(9600);                     // the Serial port of Arduino
baud rate.
}

void loop()
{
    if (SoftSerial.available())                     // if date is coming from
software serial port ==> data is coming from SoftSerial shield
    {
        while(SoftSerial.available())               // reading data into char
array
        {
            buffer[count++]=SoftSerial.read();      // writing data into
array
            if(count == 64)break;
        }
```

```
        Serial.write(buffer,count);                 // if no data
transmission ends, write buffer to hardware serial port
        clearBufferArray();                        // call clearBufferArray
function to clear the stored data from the array
        count = 0;                                 // set counter of while loop
to zero


    }
    if (Serial.available())                 // if data is available on
hardware serial port ==> data is coming from PC or notebook
    SoftSerial.write(Serial.read());        // write it to the SoftSerial
shield
}

void clearBufferArray()                     // function to clear buffer
array
{
    for (int i=0; i<count;i++)
    { buffer[i]=NULL;}                      // clear all index of array with
command NULL
}
```

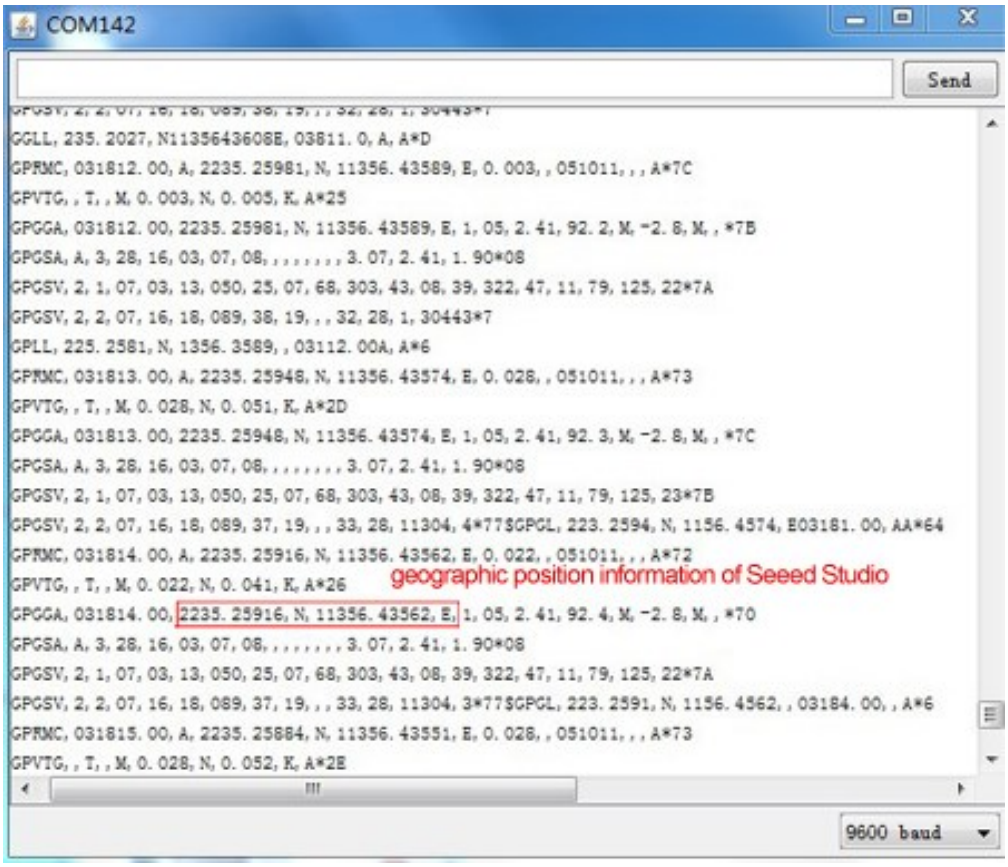
- Install u-center. Upload the code below to your Arduino/Seeeduino and then open u-center.

1) Click Receiver -> Port and select the COM port that the Arduino is using.

2) Click Receiver -> Baudrate and make sure 9600 is selected.

3) Click View -> Text Console and you should get a window that will stream NMEA data.

4) Open the serial monitor, you can see as show below:



- To View data in Google Earth:

1) Click File -> Database Export -> Google Earth KML

2) This Should launch Google Earth with the history that was captured by u-center.

3) Alternatively, data can be recorded by pressing the red circle on the toolbar which will then ask you where you want to save the record.

4) When you have captured enough data, click the black square to stop recording.

5) You can then convert the .ubx file generated to KML by using uploading the ubx file to GPSVisualizer

3.2 With Raspberry Pi

1. You should have got a raspberry pi and a grovepi or grovepi+.
2. You should have completed configuring the development environment, otherwise follow here.
3. Connection. Plug the sensor to grovepi socket D4 by using a grove cable.
4. Navigate to the demos' directory:

```
cd yourpath/GrovePi/Software/Python/
```

To see the code

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

```
import serial, time
import smbus
import math
import RPi.GPIO as GPIO
import struct
import sys

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

class GPS:
    #The GPS module used is a Grove GPS module
http://www.seeedstudio.com/depot/Grove-GPS-p-959.html
    inp=[]
    # Refer to SIM28 NMEA spec file
http://www.seeedstudio.com/wiki/images/a/a0/SIM28_DATA_File.zip
    GGA=[]

    #Read data from the GPS
    def read(self):
        while True:
            GPS.inp=ser.readline()
            if GPS.inp[:6] =='$GPGGA': # GGA data , packet 1, has all the
data we need
                break
            time.sleep(0.1)
        try:
            ind=GPS.inp.index('$GPGGA',5,len(GPS.inp))  #Sometimes multiple
GPS data packets come into the stream. Take the data only after the last
'$GPGGA' is seen
            GPS.inp=GPS.inp[ind:]
        except ValueError:
            print ""
        GPS.GGA=GPS.inp.split(",")  #Split the stream into individual
parts
        return [GPS.GGA]

    #Split the data into individual elements
    def vals(self):
        time=GPS.GGA[1]
        lat=GPS.GGA[2]
```

```
        lat_ns=GPS.GGA[3]
        long=GPS.GGA[4]
        long_ew=GPS.GGA[5]
        fix=GPS.GGA[6]
        sats=GPS.GGA[7]
        alt=GPS.GGA[9]
        return [time,fix,sats,alt,lat,lat_ns,long,long_ew]

g=GPS()
f=open("gps_data.csv",'w')  #Open file to log the data
f.write("name,latitude,longitude\n")    #Write the header to the top of
the file
ind=0
while True:
    try:
        x=g.read()  #Read from GPS
        [t,fix,sats,alt,lat,lat_ns,long,long_ew]=g.vals()   #Get the
individial values
        print "Time:",t,"Fix status:",fix,"Sats in
view:",sats,"Altitude",alt,"Lat:",lat,lat_ns,"Long:",long,long_ew
        s=str(t)+","+str(float(lat)/100)+","+str(float(long)/100)+"\n"
        f.write(s)  #Save to file
        time.sleep(2)
    except IndexError:
        print "Unable to read"
    except KeyboardInterrupt:
        f.close()
        print "Exiting"
        sys.exit(0)
```

5. Run the demo.

```
sudo python grove_gps.py
```

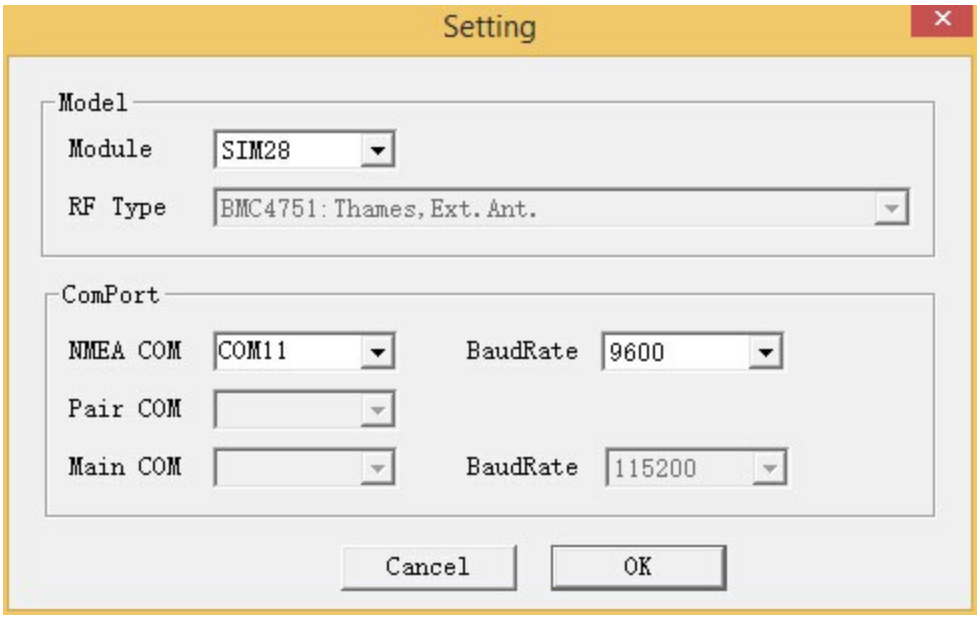
6.Result

- Note: GPS is better outdoor using, recommand you to put your raspberry pi to the window or any place outdoor.

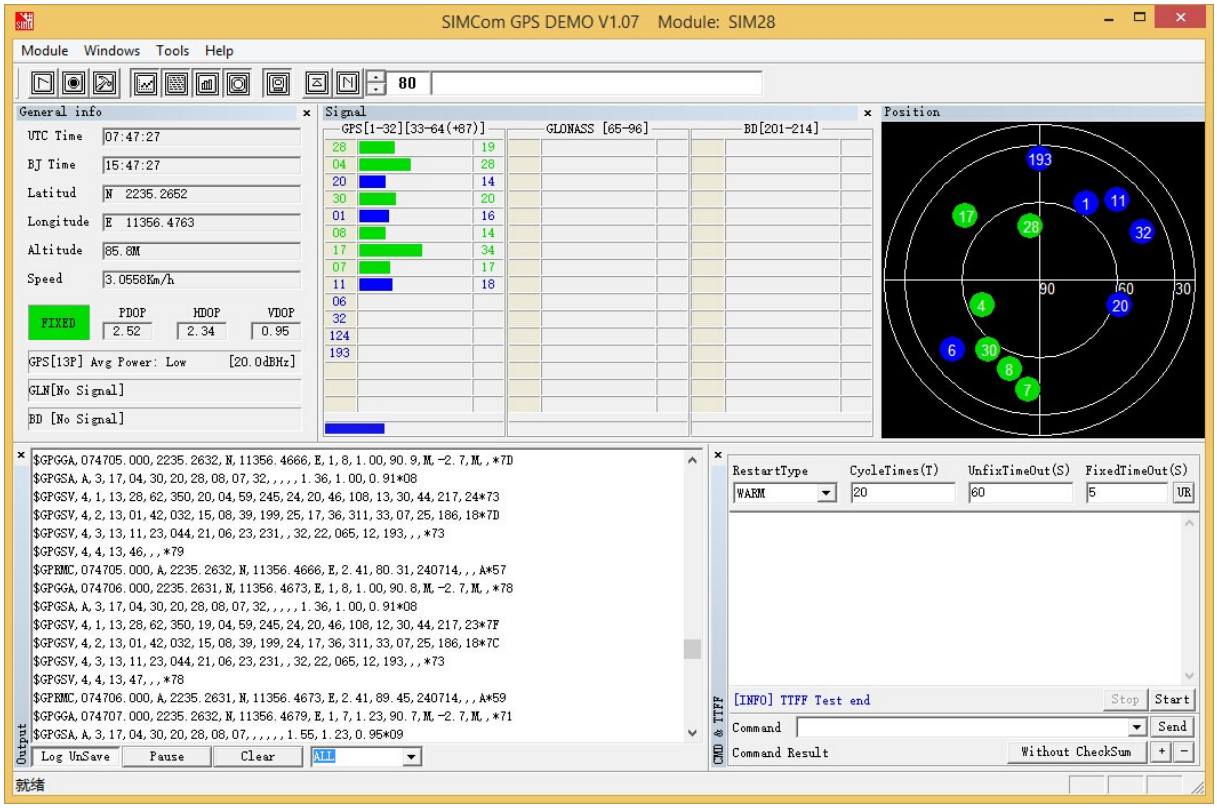
```
pi@192.168.18.111 [Disconnected]
Time: 094628.000 Fix status: 2 Sats in view: 9 Altitude 62.6 Lat: 2235.2487 N Long: 11356.4267 E
Time: 094629.000 Fix status: 2 Sats in view: 9 Altitude 62.5 Lat: 2235.2486 N Long: 11356.4267 E
Time: 094630.000 Fix status: 2 Sats in view: 9 Altitude 62.3 Lat: 2235.2482 N Long: 11356.4269 E
Time: 094631.000 Fix status: 2 Sats in view: 9 Altitude 62.1 Lat: 2235.2475 N Long: 11356.4270 E
Time: 094632.000 Fix status: 2 Sats in view: 9 Altitude 61.8 Lat: 2235.2471 N Long: 11356.4272 E
Time: 094633.000 Fix status: 2 Sats in view: 9 Altitude 61.5 Lat: 2235.2468 N Long: 11356.4274 E
Time: 094634.000 Fix status: 2 Sats in view: 9 Altitude 61.2 Lat: 2235.2468 N Long: 11356.4276 E
Time: 094635.000 Fix status: 2 Sats in view: 9 Altitude 61.0 Lat: 2235.2469 N Long: 11356.4279 E
Time: 094636.000 Fix status: 2 Sats in view: 10 Altitude 60.8 Lat: 2235.2469 N Long: 11356.4282 E
Time: 094637.000 Fix status: 2 Sats in view: 10 Altitude 60.8 Lat: 2235.2469 N Long: 11356.4282 E
Time: 094638.000 Fix status: 2 Sats in view: 10 Altitude 60.8 Lat: 2235.2469 N Long: 11356.4282 E
Time: 094639.000 Fix status: 2 Sats in view: 10 Altitude 60.8 Lat: 2235.2469 N Long: 11356.4282 E
```

4. SIM28 module Note:

1. GPS Bee has change the module as SIM28 which the same footprint as origin version.
2. You should use "SIMCom GPS DEMO" tools to receive SIM28 module data.
3. Open SIMCom_GPS_DEMO tools, go to Module->properties->module->select SIM28.



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


SIMCom GPS DEMO V1.07 Module: SIM28
Module Windows Tools Help
80
General info
UTC Time 07:47:27
BJ Time 15:47:27
Latitud N 2235.2652
Longitude E 11356.4763
Altitude 85.8M
Speed 3.0558Km/h
FIXED
PDOP 2.52
HDOP 2.34
VDOP 0.95
GPS[13P] Avg Power: Low [20.0dBHz]
GLN[No Signal]
BD [No Signal]
Signal
GPS[1-32][33-64(+87)]
GLONASS [65-96]
BD[201-214]
Position
193
1
11
17
28
32
90
60
30
4
20
6
30
8
7
$GPGGA,074705.000,2235.2632,N,11356.4666,E,1,8,1.00,90.9,M,-2.7,M,,*7D
$GPGSA,A,3,17,04,30,20,28,08,07,32,,,,,1.36,1.00,0.91*08
$GPGSV,4,1,13,28,62,350,20,04,59,245,24,20,46,108,13,30,44,217,24*73
$GPGSV,4,2,13,01,42,032,15,08,39,199,25,17,36,311,33,07,25,186,18*7D
$GPGSV,4,3,13,11,23,044,21,06,23,231,,32,22,065,12,193,,,*73
$GPGSV,4,4,13,46,,,*79
$GPRMC,074705.000,A,2235.2632,N,11356.4666,E,2.41,80.31,240714,,,A*57
$GPGGA,074706.000,2235.2631,N,11356.4673,E,1,8,1.00,90.8,M,-2.7,M,,*78
$GPGSA,A,3,17,04,30,20,28,08,07,32,,,,,1.36,1.00,0.91*08
$GPGSV,4,1,13,28,62,350,19,04,59,245,24,20,46,108,12,30,44,217,23*7F
$GPGSV,4,2,13,01,42,032,15,08,39,199,24,17,36,311,33,07,25,186,18*7C
$GPGSV,4,3,13,11,23,044,21,06,23,231,,32,22,065,12,193,,,*73
$GPGSV,4,4,13,47,,,*78
$GPRMC,074706.000,A,2235.2631,N,11356.4673,E,2.41,89.45,240714,,,A*59
$GPGGA,074707.000,2235.2632,N,11356.4679,E,1,7,1.23,90.7,M,-2.7,M,,*71
$GPGSA,A,3,17,04,30,20,28,08,07,,,,,,1.55,1.23,0.95*09
Output
Log UnSave
Pause
Clear
ALL
RestartType
CycleTimes(T)
UnfixTimeOut(S)
FixedTimeOut(S)
WARM
20
60
5
UR
[INFO] TTFF Test end
Stop
Start
Command
Send
Command Result
Without CheckSum
CMD & TTFF
就绪

5. Version Tracker

Revision	Descriptions	Release
GPS Bee kit (with Mini Embedded Antenna)	-	origin
v1.1	change the GPS module to SIM28	Dec 5,2013

6. Resources

GPS Eagle File

GPS Schematic(PDF)

E-1612-UB Datasheet

U-Blox6 Receiver Description Protocol Spec

U-Blox u-center GPS evaluation software

SIM28_DATA_File

SIMCom_GPS_DEMO_V1.07