

Grove - GPS SKU: 113020003



这个 Grove - GPS 模块是一款低成本的现场可编程小配件，其配备了 SIM28 (u-blox 6 是旧版本) 和串行通信配置。它配备了一个具有 22 个跟踪通道，66 个采集通道的 GPS 接收器。它的跟踪和采集的灵敏度可达到 -160dBm，这使其成为个人导航项目和定位服务的最佳选择，同时也是同类产品中的性价比佼佼者。

产品特性

- 支持 NMEA 和 u-blox 6 协议。(截至 2014 年 1 月 10 日，之后 SIM28)
- 低功耗
- 可配置波特率
- Grove 兼容接口

!!!Tip 关于 Grove 模块的更多信息请点击 [Grove System](#)

规格参数

项目	范围/值
----	------

项目	范围/值
输入电压	3.3/5V
波特率	4800 - 57600 (u-blox 版本)
波特率	9600 - 115200 (SIM28 版本)
默认波特率	9600

Platforms Supported

入门指导

遵循 [Grove system](#) 可以帮助用户实现 Grove 入门。

与 Arduino 一起使用

该示例仅使用软件串行从 GPS 读取，并将其发送回串行端口。

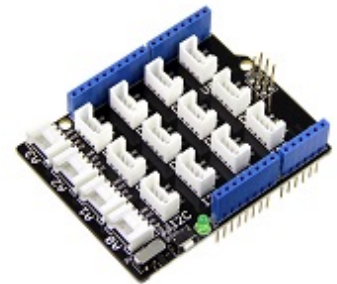
连接

在这里，我们将通过一个简单的演示向您展示这个 Grove - GPS 的工作原理。首先，我们需要准备以下内容：

Seeeduino V4

Grove - GPS

Base Shield



- 使用 Grove 线缆将 Grove-GPS 连接到 Grove - Base Shield 上的 数字端口2。
- 把 base Shield 插入 Seeeduino-V4。
- 使用 USB 线缆将 Arduino 连接到 PC。

软件部分

!!!Note 请注意，u-center 软件仅适用于 Windows。

- 安装 [u-center](#)。
- 将下面的代码上传到您的 Arduino / Seeeduino。

```

#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
}

```

- 打开 U-center.
- 点击 **Receiver -> Port**, 然后选择 Arduino 正在使用的 COM 端口。
- 点击 **Receiver -> Baudrate** 确保选到 9600。
- 点击 **View -> Text Console** 然后您会得到一个 NMEA 数据流的窗口。
- 打开串行监视器, 您可以看到如下所示:

软件部分

- 跳转到演示目录：

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

- 找到代码。

```
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
    https://raw.githubusercontent.com/SeeedDocument/Grove-GPS/master/res/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 individual
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)
```

- 运行代码。

```
sudo python grove_gps.py
```

- 结果如下。

```

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

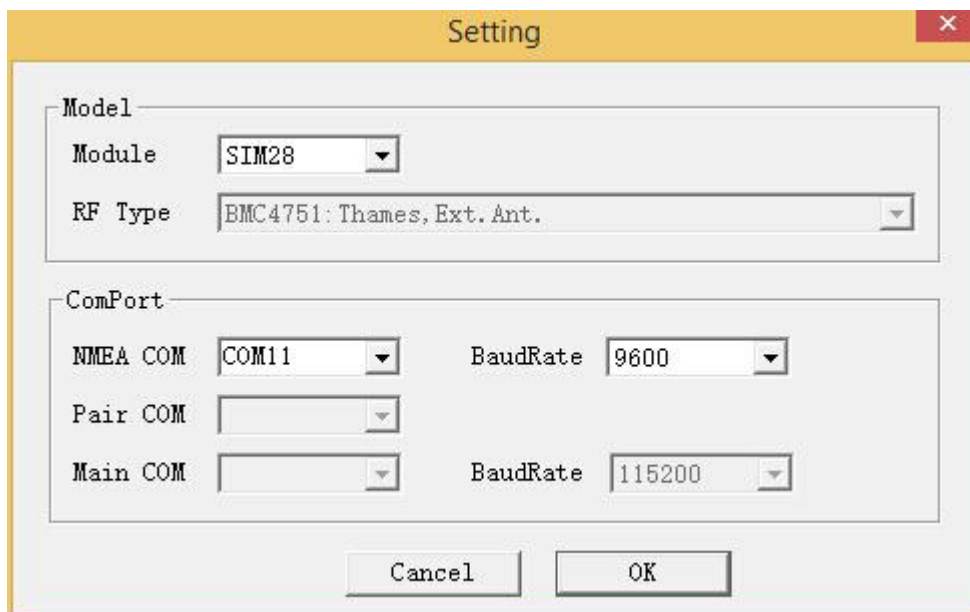
```

Note

GPS 更适合户外使用。建议将您的 raspberry pi 放在窗外或室外任何地方。

SIM28 模块注意：

1. Grove-GPS 将模块更改为 SIM28，与原始版本相同。
2. 我们应该使用 "SIMCom GPS DEMO" 工具来接收 SIM28 模块数据。
3. 打开 SIMCom_GPS_DEMO 工具，转到 **Module->properties->module->select SIM28**。
4. SIMCom_GPS_DEMO_V1.07 仅适用于 Windows。



5. 打开 SIMCom_GPS_DEMO 工具，转到 **Module->connect**。选择 GPS 模块使用的串行端口。

The screenshot displays the SIMCom GPS DEMO V1.07 software interface. The main window is titled "SIMCom GPS DEMO V1.07 Module: SIM28". It features a menu bar (Module, Windows, Tools, Help) and a toolbar. The interface is divided into several sections:

- General info:** Shows UTC Time (07:47:27), BJ Time (15:47:27), Latitude (N 2235.2652), Longitude (E 11356.4763), Altitude (85.8M), and Speed (3.0558Km/h). It also displays DOP values (PDOP: 2.52, HDOP: 2.34, VDOP: 0.95) and signal status (GLN: No Signal, BD: No Signal).
- Signal:** A table showing signal strength for various satellites. The columns are GPS [1-32] [33-64] (+67), GLONASS [65-96], and BD [201-214]. The rows list satellite IDs (e.g., 28, 04, 20, 30, 01, 08, 17, 07, 11, 06, 32, 124, 193) with corresponding signal strength bars.
- Position:** A circular diagram showing the current position on a grid. The center is marked with '90' and '160'. Various satellite IDs are plotted around the grid, including 193, 11, 32, 17, 28, 4, 20, 6, 30, 8, and 7.
- Output:** A text area displaying NMEA data, including \$GPGGA, \$GPRMC, \$GPGSV, and \$GPGLW messages. The status bar at the bottom left indicates "就绪" (Ready).
- Command & TTF:** A section for sending commands, with fields for RestartType (WARM), CycleTimes(T) (20), UnfixTimeOut(S) (60), and FixedTimeOut(S) (5). It includes Stop and Start buttons, a Command input field, and a Send button. The Command Result field shows "Without CheckSum".

资源下载

- **[Eagle文件]** [Grove-GPS Eagle File](#)
- **[原理图PDF]** [GPS Schematic\(PDF\)](#)
- **[芯片数据手册]** [E-1612-UB Datasheet](#)
- **[芯片数据手册]** [U-Blox6 Receiver Description Protocol Spec](#)
- **[其他文件]** [U-Blox u-center GPS evaluation software](#)
- **[其他文件]** [SIM28_DATA_File](#)
- **[其他文件]** [SIMCom_GPS_DEMO_V1.07](#)