

使用教程

简介

树莓派 GPS 模块是一个小型定位追踪器，可以实时监测当前所在位置。支持全系列树莓派微型电脑 Raspberry Pi Model A, B, A+, B+, Zero, 2, 3。同时也支持其它的控制器的如 51 单片机、香蕉派、香橙派、arduino、stm32 等等，该模块通过串口或者 USB 口与树莓派等进行通信。

模块采用 L80-39 模组方案，采用 NMEA0183 协议，集成小型 GPS 天线，具有超低功耗，快速定位等优势，在追踪和定位中，可以测量移动速度与方向等，具有非常好的性能特点。



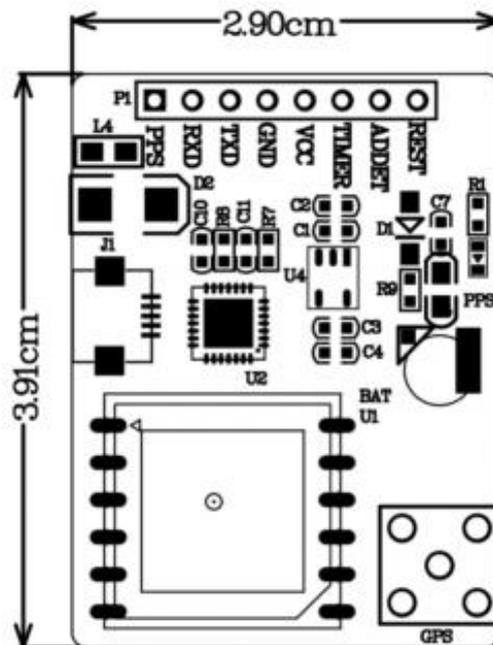
特性

- -165 dBm 灵敏度, 1Hz (默认), 最高 5Hz 数据更新率, 66 通道
- 极低的电流消耗：20ma
- PPS 信号输出
- 预留 SMA 天线接口，支持外接有源天线，实现更好的定位性能。
- LED 定位指示灯提醒，未定位时 PPS 指示灯长亮，定位后闪烁。
- 天线尺寸：15.0 x 15.0 x 4.0 mm
- 支持短路保护和天线侦测
- 支持自辅助 AGPS(EASY 技术，不需要外部的 memory 支持)
- LOCUS 技术：将日志信息自动记录到内部 Flash
- 支持 DGPS, SBAS(WAAS/EGNOS/MSAS/GAGAN)

特性

- 通道：22 (跟踪) /66 (捕获)
- 电压：3.3-5V
- 重量：4.35g
- 最大工作电流：100mA
- 工作温度：-40 ~ +85°C
- 尺寸：16.6x26x18.6(mm)
- GPS 型号 L80-39
- 天线尺寸：15x15x4(mm)
- 定位精度：<2.5m CEP
- 速度精度：<0.1 m/s
- @-130dBm 不使用 EASY™ 技术冷启动定位时间：<35s
- @-130dBm 使用 EASY™ 技术冷启动定位时间：<15s
- 捕获灵敏度：-148dm
- 追踪灵敏度：-165dm
- 更新率：默认 1HZ，最大 5HZ
- 数据格式：NMEA0183 协议
- 连接方式：1*8 2.54 间距排针, USB micro 数据线

尺寸



模块图片：



连接方式

1.UART

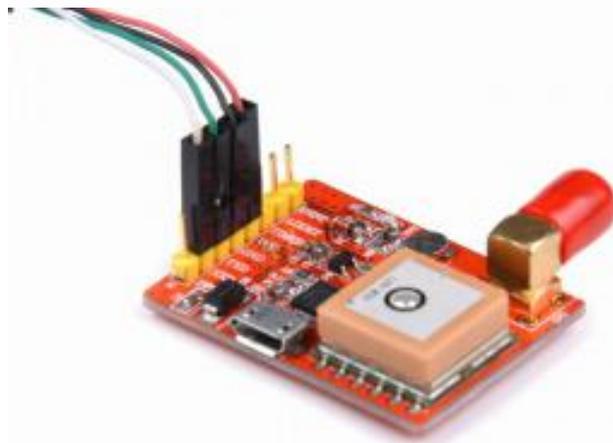
Raspberry Pi model A, B, A+, B+, Zero, 2

默认的串口为调试串口，为了使用串口作为 GPS 数据串口，我们需要使用 *raspi-config* 的配置工具来对使能串口。

Raspberry Pi model 3

由于树莓 3 的串口默认作为 BT 使用，不能再作为通讯串口使用，如果想通过串口连接 GPS，需要使用 USB-TTL 数据线，连接如下图所示

GPS	Wire color
VCC	Red wire
GND	Black wire
TXD	Green wire
RXD	White wire



2.USB(Raspberry Pi model A, B, A+, B+, Zero, 2, 3)

通过 microUSB 线连接树莓派和 GPS 模块，如下图所示



使用说明

1. 完成连接后，板子上电进入系统。首先升级内核(可忽略，但如果版本较老，可能影响设备读取)和安装 GPSD 工具

```
sudo apt-get update
sudo apt-get -y install gpsd
gpsd-clients python-gps
```

2.重启系统后，配置 gpsd 服务，使其能够开机自动运行读取串口数据

```
Sudo dpkg-reconfigure gpsd
```

3.启动 gpsd 服务

```
sudo gpsd /dev/ttyUSB0 -F /var/run/gpsd.sock
```

4.停止 gpsd 服务:

```
sudo killall gpsd
```

5.也可以使用 systemctl 命令来管理 gpsd 服务，首先修改配置文件，路径为 /etc/default/gpsd，修改设备名称，接口名称和路径。如下图所示

```
pi@raspberrypi:~$ grep -v "#" /etc/default/gpsd | grep -v "$"
START_DAEMON="true"
USBAUTO="true"
DEVICES="/dev/ttyUSB0"
GPSD_OPTIONS="-F /var/run/gpsd.sock"
pi@raspberrypi:~$
```

启动服务:

```
sudo systemctl enable gpsd.sock
sudo systemctl start gpsd.sock
```

停止服务:

```
sudo systemctl stop gpsd.sock
sudo systemctl disable gpsd.sock
```

输入 CPGS 命令获取 GPS 信息

```
sudo cgps -s
```

GPS 信息:

```
Time: 2016-06-21T07:08:09.000Z PRN: Elev: Azim: SNR: Used:
x Latitude: 31.101330 N xx 1 76 046 31 Y x
x Longitude: 121.440386 E xx 11 60 035 00 Y x
x Altitude: 435.4 ft xx 30 48 249 51 Y x
x Speed: 0.0 mph xx 137 45 139 37 Y x
x Heading: 105.8 deg (true) xx 193 44 138 41 Y x
x Climb: 0.0 ft/min xx 28 42 324 23 Y x
x Status: 3D FIX (2 sats) xx 7 39 209 48 Y x
x Longitude Err: +/- 12 ft xx 22 34 117 44 Y x
x Latitude Err: +/- 11 ft xx 8 28 063 22 Y x
x Altitude Err: +/- 37 ft xx 3 26 146 39 N x
x Course Err: n/a xx 17 23 281 24 N x
x Speed Err: +/- 16 mph xx x x
x Time offset: 0.683 xx x x
x Grid Square: PM01rc xx x x
```

使用 Python 解析 GPS 数据

库文件

1. serial

serial 是一个默认的库，将与 Raspbian 预装。不需要安装任何东西。

2. pynmea2

pynmea2 是一个易于使用解析 NMEA 语句的库。需要安装 (先安装 pip 再通过 pip 安装 pynmea2)。具体安装命令如下：

安装 PIP：

```
sudo apt-get install python-pip
```

通过“pip”安装 pynmea2:

```
sudo pip install pynmea2
```

Python 代码：

```
import serial
import pynmea2
def parseGPS(str):
    if str.find('GGA') > 0:
        msg = pynmea2.parse(str)
        print "Timestamp: %s -- Lat: %s %s -- Lon: %s %s --
Altitude: %s %s" %
(msg.timestamp,msg.lat,msg.lat_dir,msg.lon,msg.lon_dir,msg.altitude,msg.
altitude_units)
```

```
serialPort = serial.Serial("/dev/ttyUSB0", 9600, timeout=0.5)
```

```
while True:
```

```
    str = serialPort.readline()
```

```
    parseGPS(str)
```

运行代码，结果如下：

```
Timestamp: 14:19:56 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:19:57 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:19:58 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:19:59 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:20:00 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:20:01 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:20:02 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:20:03 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:20:04 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:20:05 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:20:06 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:20:07 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:20:08 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:20:09 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:20:10 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:20:11 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:20:12 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:20:13 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:20:14 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:20:15 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:20:16 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:20:17 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:20:18 -- Lat: 3413.7896 N -- Lon: 10851.0013 E -- Altitude: 514.1 M
Timestamp: 14:20:19 -- Lat: 3413.7901 N -- Lon: 10851.0013 E -- Altitude: 514.0 M
Timestamp: 14:20:20 -- Lat: 3413.7901 N -- Lon: 10851.0014 E -- Altitude: 514.1 M
```