SKU:TEL0137 (https://www.dfrobot.com/product-2216.html)

(https://www.dfrobot.com/product-2216.html)

(https://www.dfrobot.com/product-2212.html)

Introduction

USB GPS Receiver

This GPS receiver brings much more accurate and faster positioning performance with stronger signal compared with traditional GPS receivers. It is compact, lightweight and portable. Integrated with built-in receiving antenna, this product adopts a high-precision positioning chip and industrial-grade manufacturing process so that it can meet the positioning requirements of both industrial-grade and personal use. The GPS receiver has battery inside for supplying power for storing satellite data, such as satellite signal status, final position and time that would greatly increase the positioning speed at the module’s next boot. In addition, this receiver adopts USB interface, which can be conveniently used in maincontrollers like Raspberry Pi, NVIDIA, LattePanda. It is suitable for vehicle navigation, handheld positioning, wearable devices and other fields.

USB GPS Receiver (2m Extension Cable)

This is an Ublox chip-based GPS receiver module of low power and high sensitivity that can receive 56 channels satellite signal. Compared with traditional GPS receivers, it brings much more accurate and faster positioning performance with stronger signal. Integrated with built-in receiving antenna, this product adopts a high-precision positioning chip and industrial-grade manufacturing process so that it can meet the positioning requirements of both industrial-grade and personal use. The GPS receiver has battery inside for supplying power for storing satellite data, such as satellite signal status, final position and time that would greatly increase the positioning speed at the module’s next boot. Besides, this receiver adopts USB interface, which can be conveniently used in maincontrollers like Raspberry Pi, NVIDIA, LattePanda. The module has a 2m USB cable and its bottom is magnetic which makes it able to be easily attached to metal objects. It is suitable for vehicle navigation, handheld positioning, wearable devices and other fields.

Specification

- Chip: UBX-G7020-KT
- Frequency: L1, 1575.42MHz/L2,1561.10MHZ/L3,1602.00MHZ
- Baud Rate: 4800,9600,19200,38400,57600,115200bps
- Receiving Channel: 56CH
- Sensitivity: tracking -162dBm; Acquisition: 160dBm; cold start: -148dBm
- Cold Start: 29s average
- Warm Start: 3s average
- Hot Start: 1s average
- Accuracy: horizontal position accuracy<2.5M, SBAS<2.0M
- Timing Accuracy: 30ns
- Update Rate: 1Hz

- Operating Temperature: -40℃~85℃
- Storage Temperature: -40℃~85℃

Tutorial

1. NMEA0183 Protocol Introduction

Connect USB GPS receiver to your computer, and place in an open space. Check sensor data on via serial assistant.

The sensor will output the following data:

```
$GPGGA,061831.000,2236.9152,N,11403.2422,E,2,07,1.1,144.0,M,-2.2,M,4.8,0000*60
$GPGSA,A,3,18,22,25,12,14,21,24,15,,,,,1.93,1.04,1.63*01
$GPGSV,3,1,11,12,40,089,45,14,37,314,46,15,10,078,44,18,77,096,43*72
$GPGSV,3,2,11,21,27,192,31,22,60,330,43,24,24,037,45,25,42,142,41*71
$GPGSV,3,3,11,31,21,230,27,42,51,128,37,50,46,122,39*4D
$GPRMC,061831.000,A,2236.9152,N,11403.2422,E,0.00,,130214,,,D*76
$GPVTG,309.62,T, ,M,0.13,N,0.2,K*6E
```

1.1 GGA

Sample Data: $GPGGA,061831.000,2236.9152,N,11403.2422,E,2,07,1.1,144.0,M,-2.2,M,4.8,0000*60

Name	Example	Unit	Description
Message ID	$GPGGA		GGA protocol header
UTC time	061831.000		hhmmss.sss
Latitude	2236.9152		ddmm.mmmm
N/S Indication	N		N=North, S=South

Name	Example	Unit	Description
N/S Indication	N		N=North, S=South
Longitude	11403.2422		dddmm.mmmm
E/W Indication	E		W=West, E=East
Position Indication	2		0:unpositioned 1:SPS mode, position valid 2:Differential, SPS mode, position valid, 3:PPS mode, position valid
Satellites Used	07		Range: 0 to 12
HDOP	1.1		Horizontal Precision
MSL Altitude	144.0	Meters	-
Unit	M	Meters	
Geoidal	-2.2	Meters	-
Units	M		-
Differential Time	4.8	Second	Invalid when DGPS is not used
Differential ID	0000		
Checksum	*60		

1.2 GSA

Sample Data: $GPGSA,A,3,18,22,25,12,14,21,24,15,,,,,1.93,1.04,1.63*01

Name	Sample	Unit	Description
Message ID	$GPGSA		GSA protocol Header

Name	Sample	Unit	Description
Message ID	$GPGSA		GSA protocol Header
Mode 1	A		M=Manual(forced to operate in 2D or 3D mode), A=Automatic
Mode 2	3		1:Position invalid, 2:2D Position, 3:3D Position
Satellite used	18		Channel 1
Satellite used	22		Channel 2
Satellite used	25		Channel 3
Satellite used	12		Channel 4
Satellite used	14		Channel 5
Satellite used	21		Channel 6
Satellite used	24		Channel 7
Satellite used	15		Channel 8
,,,	,,,	,,,	,,,
Satellite used			Channel 12
PDOP	1.93		Position Precision
HDOP	1.04		Horizontal Precision
VDOP	1.63		Vertical Precision
Checksum	*01		

1.3 GSV

Sample Data:

```
$GPGSV,3,1,11,12,40,089,45,14,37,314,46,15,10,078,44,18,77,096,43*72
$GPGSV,3,2,11,21,27,192,31,22,60,330,43,24,24,037,45,25,42,142,41*71
$GPGSV,3,3,11,31,21,230,27,42,51,128,37,50,46,122,39*4D
```

Name	Sample	Unit	Description
Message ID	$GPGSV		GSV protocol header
Number of messages	3		Range 1 to 3
Message Number	1		Range 1 to 3
Number of satellites	11		
Satellite ID	12		Range 1 to 32
Elevation	40	Degrees	Max 90°
Azimuth	089	Degrees	Range 0 to 359°
SNR（C/No）	45	dBHz	Range 0 to 99, null when not tracking
Satellite ID	14		Range 1 to 32
Elevation	37	Degrees	Max 90°
Azimuth	314	Degrees	Range 0 to 359°
SNR（C/No）	46	dBHz	Range 0 to 99, null when not tracking
Satellite ID	15		Range 1 to 32

Name	Sample	Unit	Description
Elevation	10	Degrees	Max 90°
Azimuth	078	Degrees	Range 0 to 359°
SNR（C/No）	44	dBHz	Range 0 to 99, null when not tracking
Satellite ID	18		Range 1 to 32
Elevation	77	Degrees	Max 90°
Azimuth	096	Degrees	Range 0 to 359°
SNR（C/No）	43	dBHz	Range 0 to 99, null when not tracking
Checksum	*72		

1.4 RMC

Sample Code: $GPRMC,061831.000,A,2236.9152,N,11403.2422,E,0.00,,130214,,,D*76

Name	Sample	Unit	Description
Message ID	$GPRMC		RMC Protocol Header
UTC Time	061831.000		hhmmss.ss
Status	A		A=Data valid; V=Data invalid
Latitude	2236.9152		ddmm.mmmmm
N/S Indication	N		N=North, S=South
Longitude	11403.2422		dddmm.mmmm

Name	Sample	Unit	Description
E/W Indication	E		W=West, E=East
Speed over Ground	0.00	Knot（Knots）	
Course over Ground		Degrees	
Date			ddmmyy
Magnetic variation			-
Checksum	*76		

1.5 VTG

Sample Data: $GPVTG,309.62,T, ,M,0.13,N,0.2,K*6E

Name	Sample	Unit	Description
Message ID	$GPVTG		VTG protocol header
Course over Ground	309.62	Degrees	
Reference	T		True
Course over Ground	309.62	Degrees	
Reference	M		Magnetic
Speed	0.13	Knot（Knots）	
Unit	N		Knots
Speed	0.2	km/hr	

Name	Sample	Unit	Description
Unit	K		km/hr
Checksum	*6E		

On Windows or LattePanda

1. Download and install Google Earth (https://www.google.co.uk/earth/download/gep/agree.html).

2. Connect USB GPS module to your Computer or LattePanda, and open the Google Earth.

3. Configure as follows.

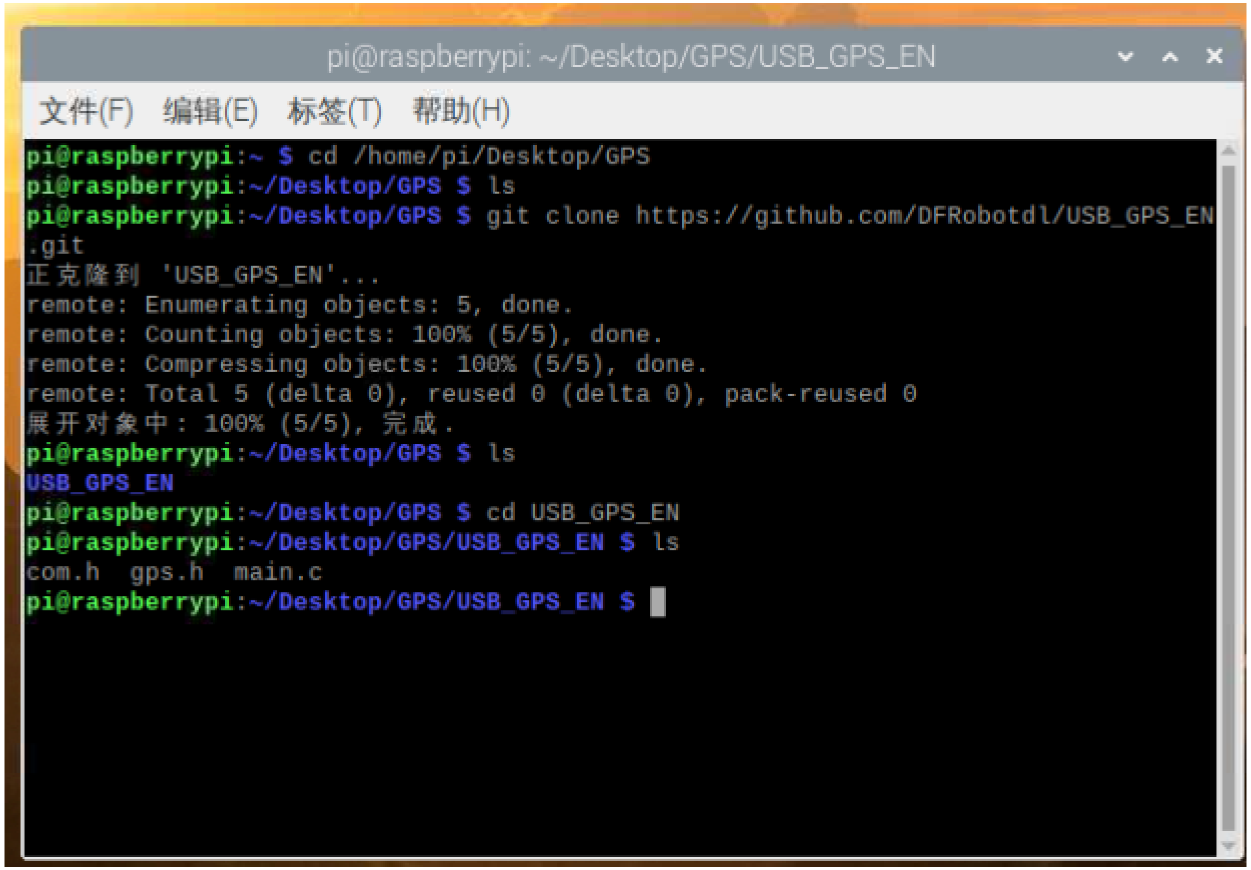
If the picture has been jumping far and near, you can click stop, and then click start after it is stable.

Read GPS Data on Raspberry Pi, Jetson Nano, or Linux

1. Download program, and plug in USB GPS receiver.

Taking Raspberry Pi as an example, same to Nano and Linux.

```
Input the following commands at the terminal
cd .....                                                      //Directory of the file you want to save
git clone https://github.com/DFRobotdl/USB_GPS_EN.git     //Download program from Github
cd USB_GPS_EN
```



pi@raspberrypi: ~/Desktop/GPS/USB_GPS_EN
文件(F) 编辑(E) 标签(T) 帮助(H)

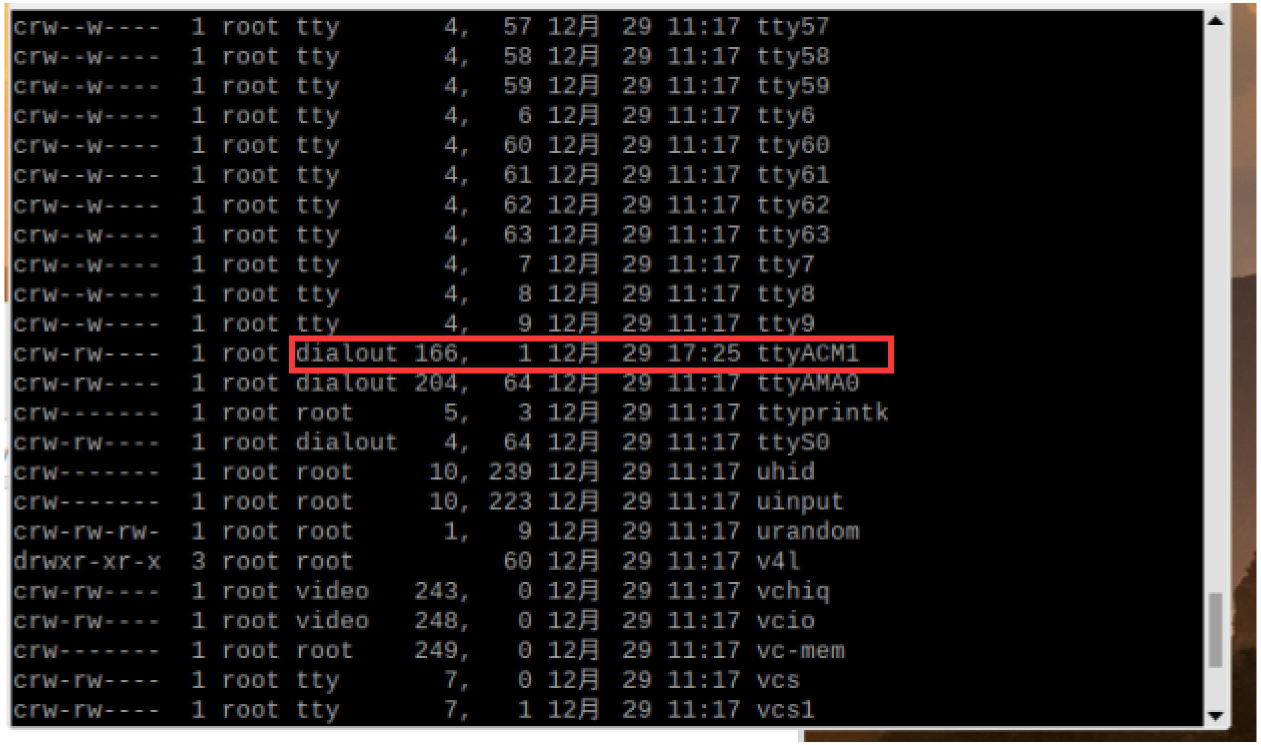
```
pi@raspberrypi:~ $ cd /home/pi/Desktop/GPS
pi@raspberrypi:~/Desktop/GPS $ ls
pi@raspberrypi:~/Desktop/GPS $ git clone https://github.com/DFRobotdl/USB_GPS_EN
.git
正克隆到 'USB_GPS_EN'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
展开对象中: 100% (5/5), 完成.
pi@raspberrypi:~/Desktop/GPS $ ls
USB_GPS_EN
pi@raspberrypi:~/Desktop/GPS $ cd USB_GPS_EN
pi@raspberrypi:~/Desktop/GPS/USB_GPS_EN $ ls
com.h  gps.h  main.c
pi@raspberrypi:~/Desktop/GPS/USB_GPS_EN $ 
```

2. Check device

Input:

```
sudo ls -l /dev
```

Find the device you just connected.

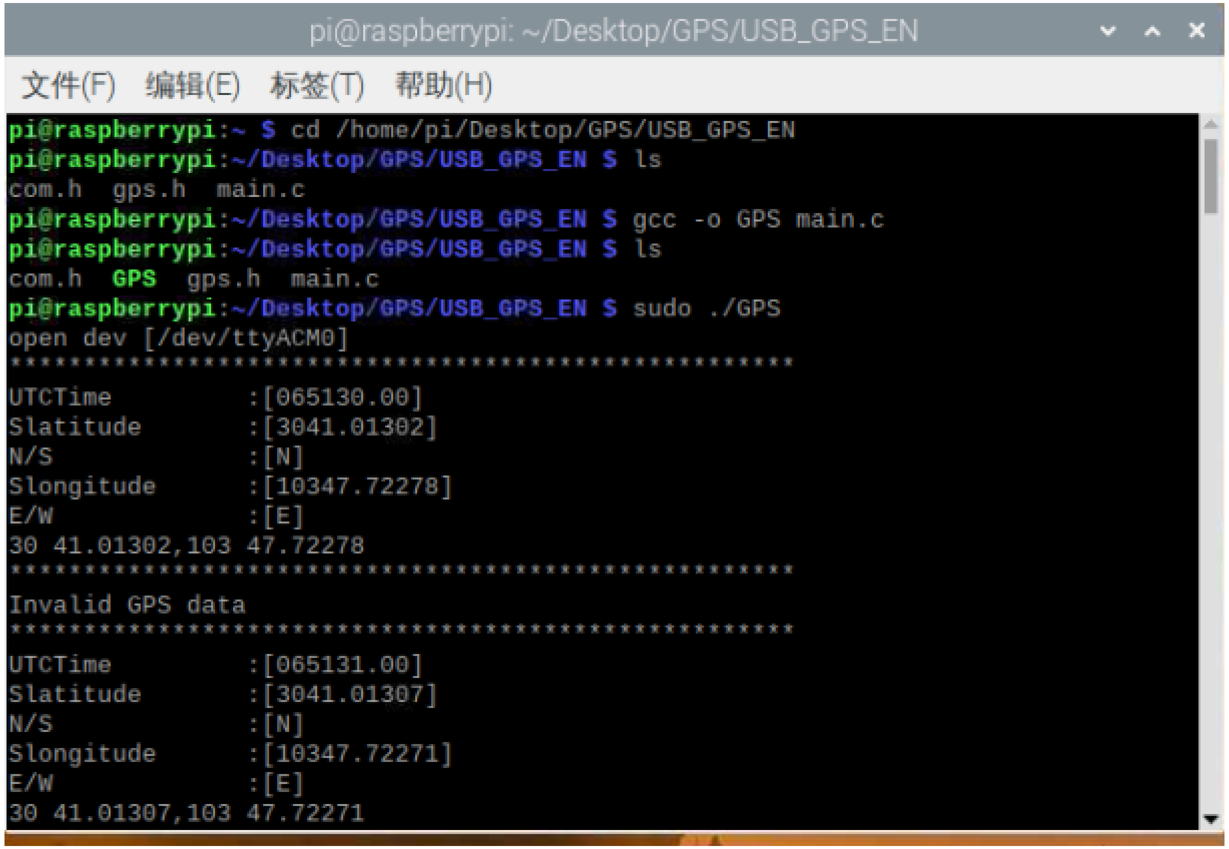


```
crw--w----  1 root tty       4,  57 12月  29 11:17 tty57
crw--w----  1 root tty       4,  58 12月  29 11:17 tty58
crw--w----  1 root tty       4,  59 12月  29 11:17 tty59
crw--w----  1 root tty       4,   6 12月  29 11:17 tty6
crw--w----  1 root tty       4,  60 12月  29 11:17 tty60
crw--w----  1 root tty       4,  61 12月  29 11:17 tty61
crw--w----  1 root tty       4,  62 12月  29 11:17 tty62
crw--w----  1 root tty       4,  63 12月  29 11:17 tty63
crw--w----  1 root tty       4,   7 12月  29 11:17 tty7
crw--w----  1 root tty       4,   8 12月  29 11:17 tty8
crw--w----  1 root tty       4,   9 12月  29 11:17 tty9
crw-rw----  1 root dialout 166,   1 12月  29 17:25 ttyACM1
crw-rw----  1 root dialout 204,  64 12月  29 11:17 ttyAMA0
crw-------  1 root root      5,   3 12月  29 11:17 ttyprintk
crw-rw----  1 root dialout   4,  64 12月  29 11:17 ttyS0
crw-------  1 root root     10, 239 12月  29 11:17 uhid
crw-------  1 root root     10, 223 12月  29 11:17 uinput
crw-rw-rw-  1 root root      1,   9 12月  29 11:17 urandom
drwxr-xr-x  3 root root          60 12月  29 11:17 v4l
crw-rw----  1 root video   243,   0 12月  29 11:17 vchiq
crw-rw----  1 root video   248,   0 12月  29 11:17 vcio
crw-------  1 root root    249,   0 12月  29 11:17 vc-mem
crw-rw----  1 root tty       7,   0 12月  29 11:17 vcs
crw-rw----  1 root tty       7,   1 12月  29 11:17 vcs1
```

3. Open the previously downloaded min. C file, change the device port in the program to be consistent with the actual one, and save it.

4. Open the folder of the program on the terminal, compile and run.

```
cd ..... /USB_GPS_LINUX
gcc -o GPS main.c
sudo ./GPS
```



```
pi@raspberrypi:~ $ cd /home/pi/Desktop/GPS/USB_GPS_EN
pi@raspberrypi:~/Desktop/GPS/USB_GPS_EN $ ls
com.h  gps.h  main.c
pi@raspberrypi:~/Desktop/GPS/USB_GPS_EN $ gcc -o GPS main.c
pi@raspberrypi:~/Desktop/GPS/USB_GPS_EN $ ls
com.h  GPS  gps.h  main.c
pi@raspberrypi:~/Desktop/GPS/USB_GPS_EN $ sudo ./GPS
open dev [/dev/ttyACM0]
*****************************************************
UTCTime         :[065130.00]
Slatitude       :[3041.01302]
N/S             :[N]
Slongitude      :[10347.72278]
E/W             :[E]
30 41.01302,103 47.72278
*****************************************************
Invalid GPS data
*****************************************************
UTCTime         :[065131.00]
Slatitude       :[3041.01307]
N/S             :[N]
Slongitude      :[10347.72271]
E/W             :[E]
30 41.01307,103 47.72271
```

5. Copy the GPS data to the Google Map (https://www.google.com/maps)

FAQ

For any questions, advice or cool ideas to share, please visit the **DFRobot Forum** (https://www.dfrobot.com/forum/).

More Documents

DFshopping_car1.png Get **USB GPS Receiver** (https://www.dfrobot.com/product-203.html) from DFRobot Store or **DFRobot Distributor**. (https://www.dfrobot.com/index.php?route=information/distributorslogo)

Turn to the Top