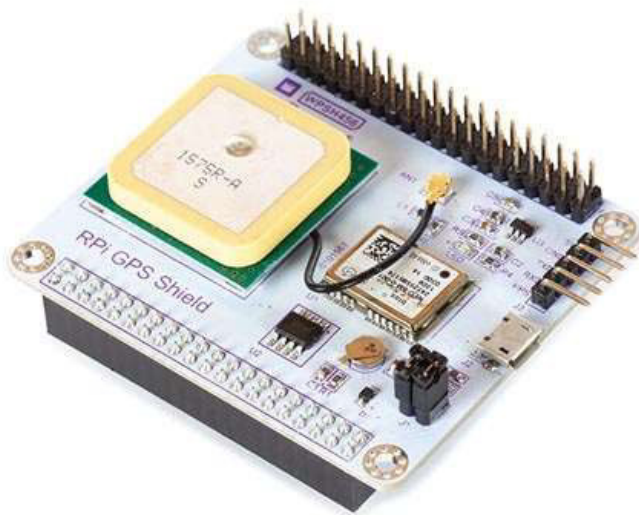**EN** NEO-6M GPS SHIELD FOR RAPSBERRY PI

WPSH456

Introduction

To all residents of the European Union
Important environmental information about this product
This symbol on the device or the package indicates that disposal of the device after its lifecycle could harm the environment. Do not dispose of the unit (or batteries) as unsorted municipal waste; it should be taken to a specialized company for recycling. This device should be returned to your distributor or to a local recycling service. Respect the local environmental rules.
If in doubt, contact your local waste disposal authorities.

Thank you for choosing Whadda! Please read the manual thoroughly before bringing this device into service. If the device was damaged in transit, do not install or use it and contact your dealer.

Safety Instructions

Read and understand this manual and all safety signs before using this appliance.

For indoor use only.

- This device can be used by children aged from 8 years and above, and persons with reduced physical, sensory or mental capabilities or lack of experience and knowledge if they have been given supervision or instruction concerning the use of the device in a safe way and understand the hazards involved. Children shall not play with the device. Cleaning and user maintenance shall not be made by children without supervision.

General Guidelines

- Refer to the Velleman® Service and Quality Warranty on the last pages of this manual.
- All modifications of the device are forbidden for safety reasons. Damage caused by user modifications to the device is not covered by the warranty.
- Only use the device for its intended purpose. Using the device in an unauthorised way will void the warranty.
- Damage caused by disregard of certain guidelines in this manual is not covered by the warranty and the dealer will not accept responsibility for any ensuing defects or problems.
- Nor Velleman nv nor its dealers can be held responsible for any damage (extraordinary, incidental or indirect) – of any nature (financial, physical…) arising from the possession, use or failure of this product.
- Keep this manual for future reference.

What is Arduino®

Arduino® is an open-source prototyping platform based on easy-to-use hardware and software. Arduino® boards are able to read inputs – light-on sensor, a finger on a button or a Twitter message – and turn it into an output – activating of a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so, you use the Arduino programming language (based on Wiring) and the Arduino® software IDE (based on Processing). Additional shields/modules/components are required for reading a twitter message or publishing online. Surf to www.arduino.cc for more information

Product overview

The Whadda NEO-6M GPS shield for Raspberry Pi allows you to easily add GPS functionality to your Raspberry Pi board. The shield simply plugs in to the 40-pin GPIO connector, no extra connections are required.

The shield is based on a U-Blox NEO-6M GPS module, a highly accurate 50 channel GPS receiver. The shield also comes with a built-in ceramic patch antenna suitable for outdoor reception of GPS satellite signals. The ceramic patch antenna is connected using a detachable u.FL cable, allowing the user to easily add a different type of antenna. Furthermore, the module also features a 5-pin header breaking out the power, serial and PPS lines to easily connect it to other boards.

Specifications

GPS chipset: U-Blox NEO-6M
Standard baudrate: 9600
Dimensions (W x L x H): 70 x 56 x 23 mm

Wiring description

Pin	Name
5 V	Supply voltage (5 V DC)
GND	Ground
TXD	GPS Serial Transmit (Device)
RXD	GPS Serial Receive (Device)
PPS	GPS Pulse Per Second output

Example program

Configuring the RPi to receive GPS data using gpsd

You can download the example Python program by going to the official Whadda github page: https://github.com/WhaddaMakers/Neo-6M-GPS-shield-for-RPi

1. If you don't have a recent version of Raspberry Pi OS installed on your Pi's microSD card, go to www.raspberrypi.org/software/ and download the latest version of the Raspberry Pi Imager. Use the Raspberry Pi Imager to flash the MicroSD card with the latest version of Raspberry Pi OS.
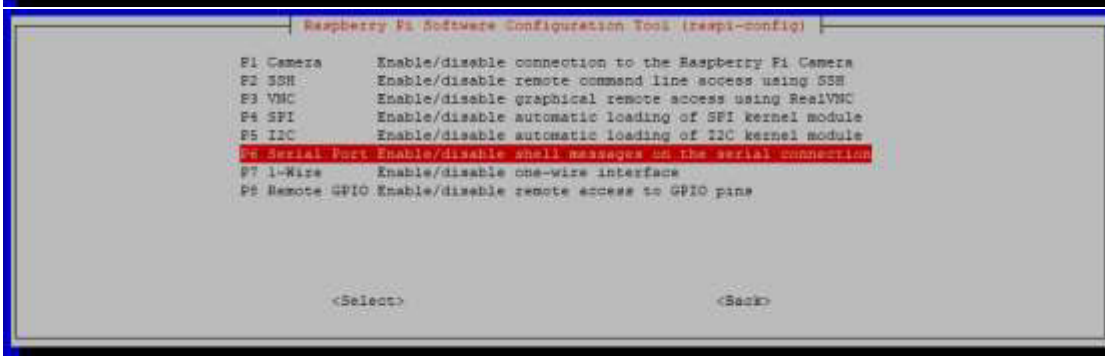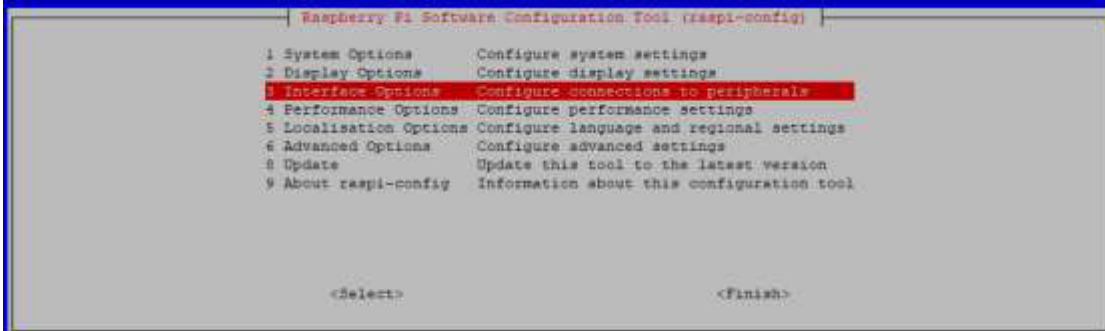
From this point on there are 2 ways to configure the pi: **on the pi itself** by connecting the necessary peripherals (mouse, keyboard and screen) to it directly, or **using your pc to connect to it remotely**. If you plan to use the 1st option you can skip step 2 and proceed directly to step 3.

2. If you want to use WiFi follow this guide to configure the WiFi access details before you proceed. Navigate to the SD Card partition named **BOOT** and add an empty file called **ssh**. Make sure that this file doesn't have a file extension. In Windows you might need to check

the File name extensions box in the View tab in file explorer to be able to do this:

3. Insert the MicroSD card into the Pi and connect all of your peripherals (if you're planning to use a separate monitor). Also plugin a network cable if you're planning to use a wired ethernet connection. Power up the pi by connecting it to the USB power supply adapter.

4. Wait 2-3 min until the pi is fully booted. If you're using a separate monitor, run through the initial Raspberry Pi OS setup wizard to configure network connections, etc... If you're connecting to the pi remotely, go to your network router setup webpage or use an IP-scanner (e.g. Angry IP Scanner) to find the assigned IP-address of your pi and connect to it by opening Powershell/Terminal and type in the following command:
ssh pi@<REPLACE WITH IP ADDRESS>. The default password is raspberry.

5. To make sure your pi is fully up-to-date, run the following command:
sudo apt update && sudo apt upgrade -y

6. Enable the serial interface (necessary to communicate with the GPS shield) by running **sudo raspi-config** and selecting **3 Interface Options** > **P6 Serial Port** > **No** (to have the login shell accessible over serial) > **Yes** (enabling the serial port hardware). It is advised to reboot the Pi after this step by typing **sudo reboot**.
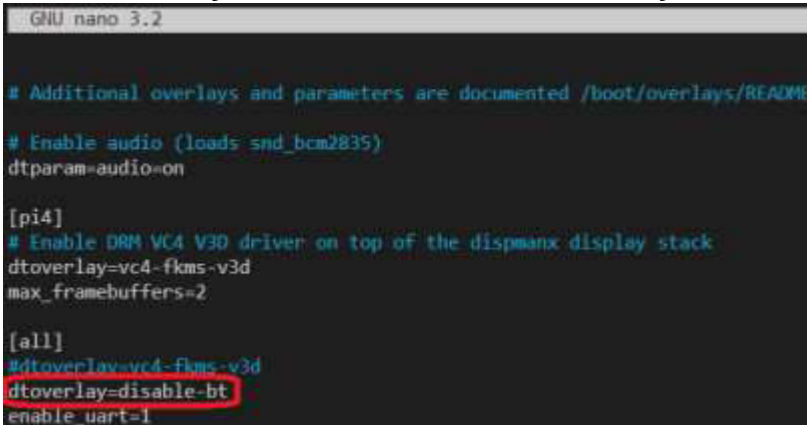
7. Open the boot config.txt configuration file by typing in: **sudo nano /boot/config.txt**

```
pi@raspberrypi:~ $ sudo nano /boot/config.txt
```

Add the following line at the end of the file: **dtoverlay=disable-bt**

Afterwards, **press ctrl-X**, type in **Y** and **hit enter**. This will save the edited file.

This change to the config file will disable the Bluetooth module, and frees up the hardware serial port for the GPS module.

8. Type in the following command to make sure the Bluetooth module is completely disabled: **sudo systemctl disable hciuart**

```
pi@raspberrypi:~ $ sudo systemctl disable hciuart
```

9. Perform a reboot by typing in **sudo reboot**
10. Connect the GPS shield to the Raspberry Pi if you haven't already done so.
11. Install minicom by typing in: **sudo apt install minicom**
12. Check if the GPS module can properly communicate with your Raspberry Pi by using the minicom terminal by typing in the following command:
minicom -b 9600 -o -D /dev/ttyAMA0

```
pi@raspberrypi:~ $ minicom -b 9600 -o -D /dev/ttyAMA0
```

If you see GPS messages coming in starting with a $ sign, everything is working properly:

```
Welcome to minicom 2.7.1

OPTIONS: I18n
Compiled on Aug 13 2017, 15:25:34.
Port /dev/ttyAMA0, 13:26:44

Press CTRL-A Z for help on special keys

$GPRMC,112658.00,A,5056.03554,N,00338.56300,E,0.708,,300321,,,A*7C
$GPVTG,,T,,M,0.708,N,1.312,K,A*2D
$GPGGA,112658.00,5056.03554,N,00338.56300,E,1,07,1.16,45.7,M,46.0,M,,*69
$GPGSA,A,3,24,19,32,12,25,06,31,,,,,,1.95,1.16,1.56*04
$GPGSV,3,1,11,02,28,094,,03,03,352,21,0



CTRL-A Z for help | 9600 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline | ttyAMA0
```

If this is not the case, check if all of the steps have followed correctly, reboot the Pi and check if the shield is connected properly.

Press **Ctrl-A**, then **X** and **hit enter** to exit minicom.

13. Install gpsd and gpsd-clients by running the following command:
sudo apt install gpsd gpsd-clients –y
```
pi@raspberrypi:~ $ sudo apt install gpsd gpsd-clients -y
```

14. Disable the gpsd service that started running automatically by running the following command:
sudo systemctl disable gpsd.socket
```
pi@raspberrypi:~ $ sudo systemctl disable gpsd.socket
```

15. Start a new gpsd instance that redirects the GPS data to the correct serial port by typing in the following command:
sudo gpsd /dev/ttyAMA0 -F /var/run/gpsd.sock

```
pi@raspberrypi:~ $ sudo gpsd /dev/ttyAMA0 -F /var/run/gpsd.sock
```

16. The GPS decoding deamon should be up and running now. To check if it is able to receive and parse the GPS data run the following command: **sudo gpsmon**
This will open a monitor program that will print out all of the received GPS parameters such as latitude, longitude, time, speed, course, satellite data, etc...

```
tcp://localhost:2947                NMEA0183>

Time: 2021-03-30T12:13:31.000Z Lat:   50 56' 00.04938" Non:    3 38' 34.00517" E
                          Cooked TPV

  GPZDA GPGBS GPRMC GPVTG GPGGA GPGSA GPGSV GPGLL GPGST
                          Sentences

Ch PRN  Az El S/N    Time:        121331.00      Time:        121331.00
0   2  71 34  33    Latitude:     5056.00823 N   Latitude:   5056.00823
1   3 335  1   0    Longitude:   00338.56753 E   Longitude: 00338.56753
2   6  36 17  33    Speed:        0.096           Altitude:  -1.8
3  12  79 45  18    Course:       355.14          Quality:   1    Sats: 05
4  24 143  9   0    Status:       A       FAA: A  HDOP:      4.97
5  25 358 84  20    MagVar:                       Geoid:     46.0
6  29 200 48  14         —— RMC ——                  —— GGA ——
7  31 304 34  39
8  32 242 31   0    Mode: A3 Sats: 2 6 12 25 31   UTC: 121331.00 RMS: 39
9                   DOP: H=4.97  V=5.42  P=7.35   MAJ:            MIN:
10                  TOFF:  0.133681956            ORI:           LAT: 18
11                  PPS:                          LON: 3.6       ALT: 26
    —— GSV ——          —— GSA + PPS ——               —— GST ——
(68) $GPGSV,3,2,09,24,09,143,,25,84,358,21,29,48,200,13,31,34,304,38*70
(29) $GPGSV,3,3,09,32,31,242,*47
(52) $GPGLL,5056.00858,N,00338.56762,E,121329.00,A,A*68
(37) $GPGST,121329.00,40,,,,18,3.6,26*51
(38) $GPZDA,121329.00,30,03,2021,00,00*6D
(39) $GPGBS,121329.00,17.8,3.6,26.0,,,,*44
(68) $GPRMC,121330.00,A,5056.00841,N,00338.56758,E,0.056,,300321,,,A*78
(35) $GPVTG,,T,,M,0.056,N,0.104,K,A*25
(74) $GPGGA,121330.00,5056.00841,N,00338.56758,E,1,05,4.97,-1.2,M,46.0,M,,*77
(52) $GPGSA,A,3,12,25,06,31,02,,,,,,,,7.35,4.97,5.42*08
(68) $GPGSV,3,1,09,02,34,071,32,03,01,335,,06,17,036,33,12,45,079,19*76
(68) $GPGSV,3,2,09,24,09,143,,25,84,358,21,29,48,200,13,31,34,304,38*70
(29) $GPGSV,3,3,09,32,31,242,*47
```

If the time data is not valid and no latitude and longitude are displayed, the GPS module is probably not able to receive satellite signals at its current position. Move the GPS module close to a window with the ceramic patch antenna pointed towards the sky. It may take several minutes for the GPS module to get its first fix.
Press **q** and then **enter** to exit gpsmon.

Using GPS data in a custom python script

1. Download the example python code from the Whadda githup page by entering the following command: **git clone** https://github.com/WhaddaMakers/Neo-6M-GPS-shield-for-RPi
2. Navigate to the folder with the example code using **cd Neo-6M-RPi-GPS-shield**
3. Install the required python libraries by running the following command:
pip3 install -r requirements.txt
4. Run the example code by using the following command:
python3 gps_3.py

If the GPS module has a fix, it prints out the current GPS position (latitude, longitude), time and the number of satellites in sight:

```
Number of sats in sight: 10
No fix...

Number of sats in sight: 10
Location (lat,lon): (50.933706667, 3.642983667)
GPS UTC Time: 13:01:01 2021-03-30

Number of sats in sight: 10
Location (lat,lon): (50.933720167, 3.642972833)
GPS UTC Time: 13:01:02 2021-03-30

Number of sats in sight: 10
Location (lat,lon): (50.933727833, 3.642960167)
GPS UTC Time: 13:01:03 2021-03-30

Number of sats in sight: 10
Location (lat,lon): (50.9337095, 3.642957333)
GPS UTC Time: 13:01:04 2021-03-30

Number of sats in sight: 10
Location (lat,lon): (50.933703, 3.6429565)
GPS UTC Time: 13:01:05 2021-03-30
```