

Welcome!

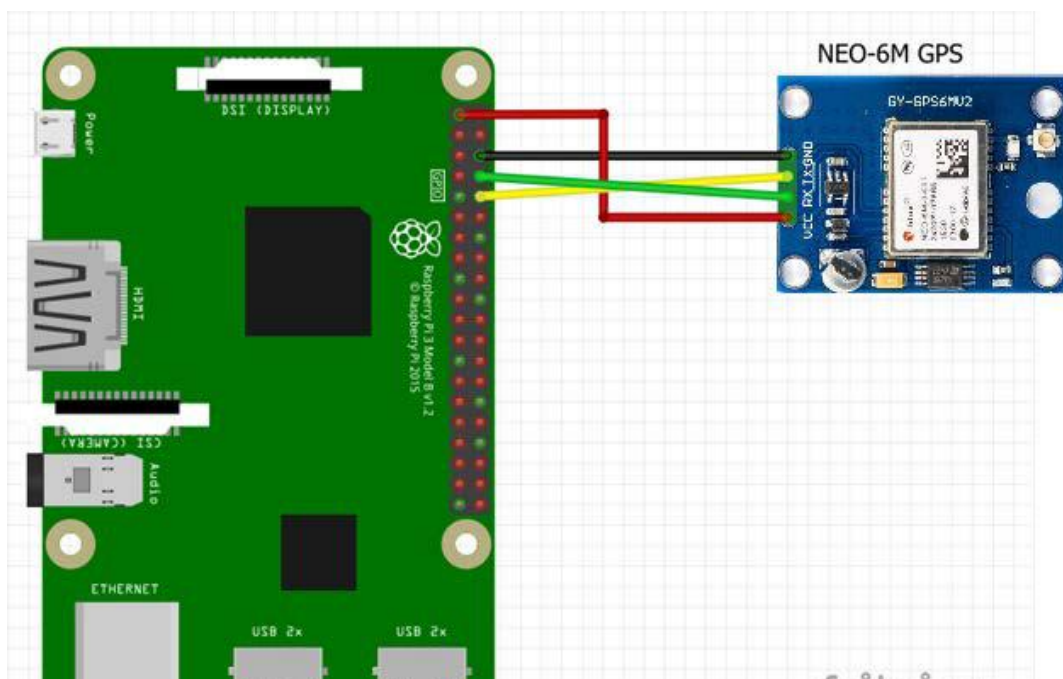
And thank you for purchasing our **AZ-Delivery NEO-6M GPS Module** for the Raspberry Pi, Arduino or PC! On the following pages, we will go through the first steps of the installation process of the Raspberry Pi.

We wish you a lot of fun!



The GPS module from Ublox comes with an active antenna, which provides a strong signal. The power supply must be between 3 and 5V. The built-in data backup battery provides a fast GPS fix.

Wiring up the GPS receiver with the Raspberry Pi:



The GPS module has only 4 Pins: VCC, GND, RX and TX.

VCC is connected to **PIN 1 (3,3V)** on the Raspberry

Red wire

GND is connected to **PIN 6 (GND)**

Black wire

TX and RX are crosswise connected:

TX is connected to **PIN 10 (RX)**

Yellow wire

RX is connected to **PIN 8 (TX)**

Green wire

After everything has been wired, the Raspberry Pi can be started.

Additional information: These instructions are based on Raspberry Pi Image from 29.11.2017 (Stretch - Lite) – updates may require slight modifications to the instructions.

The GPS receiver sends its data via the serial interface. You can activate this on the Raspberry Pi in the Raspberry configuration menu:

sudo raspi-config

```
Raspberry Pi Software Configuration Tool (raspi-config)

1 Change User Password Change password for the current user
2 Network Options       Configure network settings
3 Boot Options          Configure options for start-up
4 Localisation Options  Set up language and regional settings to match your location
5 Interfacing Options   Configure connections to peripherals
6 Overclock             Configure overclocking for your Pi
7 Advanced Options      Configure advanced settings
8 Update                Update this tool to the latest version
9 About raspi-config    Information about this configuration tool

<Select>                                <Finish>
```

After, select sub-item P6 (Serial), located in menu 5 (Interfacing Options):

```
Raspberry Pi Software Configuration Tool (raspi-config)

P1 Camera      Enable/Disable connection to the Raspberry Pi Camera
P2 SSH         Enable/Disable remote command line access to your Pi using SSH
P3 VNC         Enable/Disable graphical remote access to your Pi using RealVNC
P4 SPI         Enable/Disable automatic loading of SPI kernel module
P5 I2C         Enable/Disable automatic loading of I2C kernel module
P6 Serial      Enable/Disable shell and kernel messages on the serial connection
P7 1-Wire      Enable/Disable one-wire interface
P8 Remote GPIO Enable/Disable remote access to GPIO pins

<Select>                                <Back>
```

We will be asked a few questions:

Would you like a login shell to be accessible over serial?	->	<No>
Would you like the serial port hardware to be enabled?	->	<Yes>
The serial login shell is disabled		
The serial interface is enabled	->	<Ok>

With **<Finish>** we exit the *raspi-config* menu.

Now the Raspberry Pi should be updated:

```
sudo apt-get update
sudo apt-get upgrade
```

Do you want to continue? [Y/n] -> **y** (enter Y and confirm with *Enter*)

Now that the Raspberry Pi is updated, we can install the software.

```
sudo apt-get install minicom gpsd gpsd-clients
```

minicom:	terminal program
gpsd:	GPS Deamon
gpsd-clients	GPS viewer program

Do you want to continue? [Y/n] -> **y**

If an error occurs,

E: Unable to fetch some archives, maybe run apt-get update or try with -- fix-missing?

Then simply enter the following command:

```
sudo apt-get update --fix-missing
```

And after that start again the installation:

```
sudo apt-get install minicom gpsd gpsd-clients
```

The Raspberry Pi 3 will then use the serial interface for the Bluetooth module and put a software interface on the GPIO pins. Since these are very inaccurate, we will have to put the hardware interface back on the GPIO pins:

In order to do that, we have to go to the boot configuration:

```
sudo nano /boot/config.txt
```

And fill in the following lines, right at the end:

```
dtoverlay=pi3-miniuart-bt  
enable_uart=1  
force_turbo=1
```

Explanation:

```
dtoverlay=pi3-miniuart-bt  
enable_uart=1  
force_turbo=1
```

Put Bluetooth on mini UART
Clock frequency upon constant frequency
Baud rate is not dependent on the System Clock

With CTRL + O, we store the file again and we exit NANO

With CTRL + X, we reboot the Raspberry:

sudo reboot

After a reboot, we should set the serial interface to a baud rate of 9600:

stty -F /dev/ttyAMA0 9600

We can check if that has already been taken up:

stty -F /dev/ttyAMA0

The following output should now appear:

```
pi@raspberrypi:~ $ stty -F /dev/ttyAMA0
speed 9600 baud; line = 0; -brkint -imaxbel
pi@raspberrypi:~ $
```

Have a look at the terminal program and check if the GPS receiver also sends data:

minicom -b 9600 -o -D /dev/ttyAMA0

```
Welcome to minicom 2.7

OPTIONS: I18n
Compiled on Apr 22 2017, 09:14:19.
Port /dev/ttyAMA0, 17:41:46

Press CTRL-A Z for help on special keys

+ÁÁÁÁÖÄÝÁÑÁÝÁ-ÉÑÁ±ÉÁÁÁÁÁ±±±ÖÝ 5) ÇAYQ±±Q±±5±Á-ÉÑÁ±9±Á-ÑÜÁ±-±©É5) αA±ÄÝÑÜ
$GPGSV,3,1,11,02,09,129,,06,16,097,17,10,05,274,10,12,70,263,26*70
$GPGSV,3,2,11,14,08,325,,15,17,185,20,17,25,044,15,19,41,059,17*7F
$GPGSV,3,3,11,24,79,136,13,25,28,254,22,32,26,312,24*4A
$GPGLL,4926.93245,N,01151.78417,E,174616.00,A,A*63
$GPRMC,174617.00,A,4926.93252,N,01151.78447,E,0.194,,210118,,A*7F
$GPVTG,,T,,M,0.194,N,0.359,K,A*20
$GPGGA,174617.00,4926.93252,N,01151.78447,E,1,06,2.14,390.2,M,46.4,M,,*5B
$GPGSA,A,3,15,24,12,32,10,25,,,,,,2.90,2.14,1.96*06
$GPGSV,3,1,11,02,09,129,,06,16,097,14,10,05,274,10,12,70,263,26*73
$GPGSV,3,2,11,14,09,325,,15,17,185,20,17,25,044,15,19,41,059,17*7E
$GPGSV,3,3,11,24,79,136,10,25,28,254,22,32,26,312,24*49
$GPGLL,4926.93252,N,01151.78447,E,174617.00,A,A*61
$GPRMC,174618.00,A,4926.93252,N,01151.78442,E,0.176,,210118,,A*79
$GPVTG,,T,,M,0.176,N,0.325,K,A*27
$GPGGA,174618.00,4926.93252,N,01151.78442,E,1,06,2.14,390.5,M,46.4,M,,*56
$GPGSA,A,3,15,24,12,32,10,25,,,,,,2.90,2.14,1.96*06
$GPGSV,3,1,11,02,09,129,,06,16,097,11,10,05,274,10,12,70,263,26*76
$GPGSV,3,2,11,14,09,325,,15,17,184,21,17,25,044,14,19,41,059,17*7F
$GPGSV,3,3,11,24,79,136,11,25,28,254,22,32,26,312,24*48
$GPGLL,4926.93252,N,01151.78442,E,174618.00,A,A*6B
$GPRMC,174619.00,A,4926.93225,N,01151.78384,E,0.360,,210118,,A*70
$GPVTG,,T,,M,0.360,N,0.668,K,A*2E
$GPGGA,174619.00,4926.93225,N,01151.78384,E,1,06,2.14,390.7,M,46.4,M,,*58
```

If your output is similar to the image above, then everything has worked out.

Press first CTRL + A, and then Q; and by confirming with *YES*, you will exit the minicom.

```
+-----+
| Leave without reset? |
|  Yes      No        |
+-----+
```

Now we should configure the GPS-Deamon and let it report our position.

```
sudo nano /etc/default/gpsd
```

The last line reads: **GPSD_OPTIONS=""**
We change this line to: **GPSD_OPTIONS="/dev/ttyAMA0"**

Save again with CTRL + O and exit the Nano with CTRL + X

Then we determine the socket and the serial interface:

```
sudo gpsd /dev/ttyAMA0 -F /var/run/gpsd.sock -n
```

and then we reboot the Raspberry:

```
sudo reboot
```

After the reboot is complete, we can start the GPS client:

cgps -s

The following outcome, in addition to other supplementary information, comes from the GPS receiver:

[illegible]

You did it! Your GPS receiver is up and running!

Now it is time to learn and put your own projects into practice.

And for more hardware, our online store is always at your disposal:

<https://az-delivery.de>

Enjoy!

Imprint

<https://az-delivery.de/pages/about-us>