

GNSS Setup

In Linux, GNSS positioning is based on `gpsd` and time sync `ntpd`. The following guide expects that you have a compatible GNSS receiver.

Table of Contents

1. Prerequisites	3
2. Step-by-Step Instructions	3
3. Checking NTP Status	5
4. Glossary	7

1. Prerequisites

First, you need a compatible *GNSS* unit. Any GNSS with PPS signal that works in Linux should apply. However, if you did not purchase the GNSS unit from Kaitotek, modifications to the configuration may be needed to match with the wiring of the GNSS unit.

The following packages need to be installed (Ubuntu):

- `gpsd`
- `ntp`
- `setserial`
- `pps-tools` (optional, recommended for debugging problems)
- `gpsd-clients` (optional, recommended for debugging problems)

You need NTP with PPS clock support. By default, it is provided with latest updates of Ubuntu xenial (16.04 LTS) and yakkety (16.10) and zesty (17.04). In Ubuntu, the PPS is enabled in NTP 4.2.8p4 or later. If your system has a version of NTP without PPS support, compile it yourself with PPS clock (22) enabled. This can be done by enabling all clocks during configuration phase: `./configure --enable-all-clocks`.

This guide assumes that kernel level PPS is supported. If you have a modern distribution this should not be a problem, but for example some old CentOS distributions do not have kernel level PPS support. In this case special *shmpps* application with NMEA(20) and SHM(28) NTP drivers and special configuration might help you to get application level PPS support.



In order to reach a good result, your machine should have an integrated serial port or at least a PCI-based serial port adapter for the PPS signal. If, e.g., a USB serial port adapter is in use, the synchronization accuracy will be much worse.

2. Step-by-Step Instructions

This chapter assumes that the serial port location is `/dev/ttyS0` (an integrated serial port for PPS signal) and USB port is `/dev/ttyUSB0` (for NMEA message reception). Port numbers may vary depending on your system, so please check and change them accordingly.

To install the required packages, type:

```
sudo apt-get install gpsd gpsd-clients ntp pps-tools setserial
```

Connect the USB and serial cables. It might be useful to check in which port the USB device is attached:

```
dmesg
```

To start *gpsd* type:

```
sudo service gpsd start
```

Then, check that GNSS reception is working:

```
xgps
```

You should see satellites appearing. It might take few minutes initially (as long as 12 minutes in the worst case). Please wait until the fix is got. If nothing happens, exit the `xgps` and check that `gpsd` is using the correct USB port by typing:

```
sudo dpkg-reconfigure gpsd
sudo service gpsd restart
```

When the GNSS basic reception is working fine, enable the serial driver for PPS (18). In case there are several serial ports, please modify the port number `/dev/ttyS<port>` to match the correct one.

```
sudo ldatattach 18 /dev/ttyS0
```

`/dev/pps0` device should appear. If there are several `/dev/pps<port>` devices or you want to ensure that the PPS is working type:

```
sudo ppstest /dev/pps0
```

If you see output appearing every second, the PPS works. Press `Ctrl+C` to stop. If you don't see anything after `ok, found 1 source(s), now start fetching data...`, please check the device is properly connected and port number is correctly selected.

Enable serial port **low_latency** mode to ensure low jitter:

```
sudo setserial /dev/ttyS0 low_latency
```

Configure NTP by editing `/etc/ntp.conf`.

Disable the default servers by commenting them out, if you want to use only GPS time. Example:

```
# Use servers from the NTP Pool Project. Approved by Ubuntu Technical Board
# on 2011-02-08 (LP: #104525). See http://www.pool.ntp.org/join.html for
# more information.
#server 0.ubuntu.pool.ntp.org
#server 1.ubuntu.pool.ntp.org
#server 2.ubuntu.pool.ntp.org
#server 3.ubuntu.pool.ntp.org
```

Add the following lines:

```
#PPS
server 127.127.22.0 minpoll 4 maxpoll 4
fudge 127.127.22.0 flag2 0
#SHM
server 127.127.28.0 prefer
fudge 127.127.28.0 time1 0.55 refid GPS
```

Again, if the `/dev/pps<port>` device number is not 0, you should modify the number in PPS lines accordingly. Example (PPS device `/dev/pps1`):

```
#PPS
server 127.127.22.1 minpoll 4 maxpoll 4
fudge 127.127.22.1 flag2 0
```

Start the NTP

```
sudo service ntp start
```

3. Checking NTP Status

Wait for the clock to be synchronized. To check the status type:

```
ntpq -p
```

Typical outputs are presented next.

1. The clock is correctly synchronized (see the marks o and *):

```
remote          refid          st t when poll reach  delay  offset  jitter
=====
oPPS(1)         .PPS.         0 1  11  16  377  0.000  -0.011  0.019
*SHM(0)         .GPS.         0 1  42  64  377  0.000   6.741 15.645
```

2. Not (yet) synchronized. Please wait or recheck the configuration. Also, if the clock is off too much, NTP could refuse synching. In this case, do initial synching manually.

```
remote          refid          st t when poll reach  delay  offset  jitter
=====
PPS(1)         .PPS.         0 1  -  16  0  0.000  0.000  0.000
SHM(0)         .GPS.         0 1  -  64  0  0.000  0.000  0.000
```

3. SHM only synchronized. Please wait for the PPS sync or recheck the PPS configuration.

```
remote          refid          st t when poll reach  delay  offset  jitter
=====
PPS(1)         .PPS.         0 1  -  16  0  0.000  0.000  0.000
*SHM(0)         .GPS.         0 1  3  64  1  0.000 -14.525 0.000
```

4. Clock sources dropped as falseticker due to poor accuracy. Please check that the GPS signal is strong enough and restart NTP.

```
remote          refid          st t when poll reach  delay  offset  jitter
=====
xPPS(1)        .PPS.         0 1  -  16  0  0.000  0.000  0.000
xSHM(0)        .GPS.         0 1  -  64  0  0.000  0.000  0.000
```

5. No PPS clock. Please check that PPS (22) is enabled in *ntp.conf* and that NTP has PPS driver support.

```
remote          refid          st t when poll reach  delay  offset  jitter
=====
*SHM(0)         .GPS.          0 1  3  64  1  0.000 -14.525  0.000
```

6. NTP not running:

```
ntpq: read: Connection refused
```

4. Glossary

Global Navigation Satellite System

A general term under which all the different global satellite navigation systems (e.g., GPS, GLONASS, Galileo, BDS) fall.