GpsNtp-Pi

~

Installation and Operation Guide

See last page for copyright and document information

GpsNtp-Pi ~ Installation and Operation Guide

Table of Contents

GpsNtp-Pi Time Server Installation and Operation Guide

Whitham D. Reeve

1. Introduction

This document describes the installation and operation of the GpsNtp-Pi time server using the Raspberry Pi computer platform, a GPS receiver and the Network Time Protocol (NTP). An accompanying document, ***GPS Network Time Server on Raspberry Pi: GpsNtp-Pi*** {GpsNtp-Pi}, provides a general description. The system may operate as a standalone time server or in conjunction with external time servers and is used to synchronize PC real-time clocks to Coordinated Universal Time (figure 1).

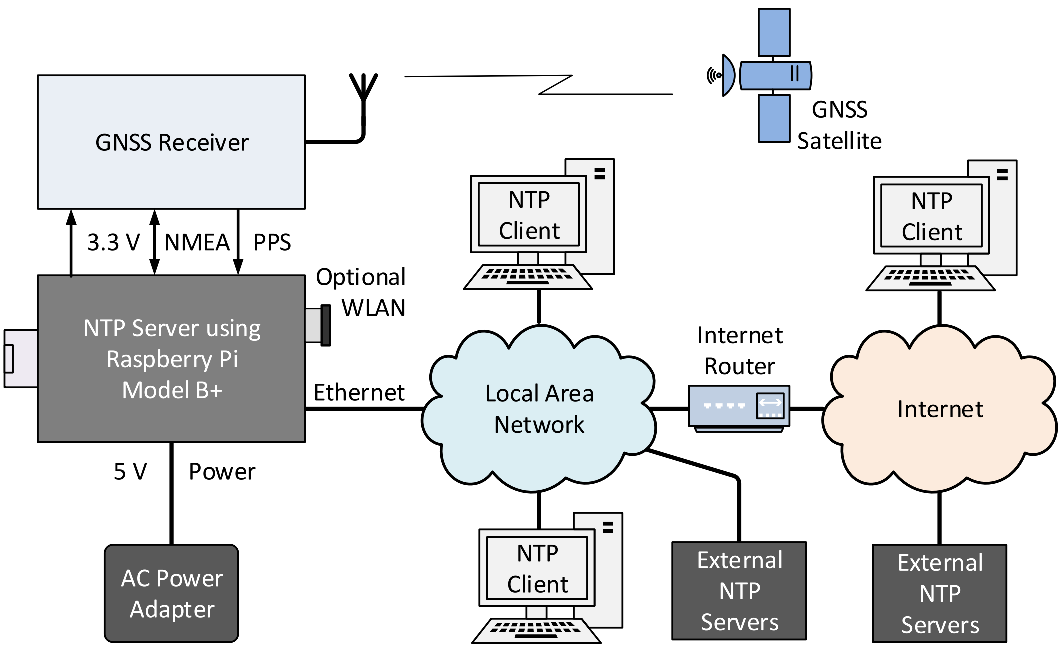


Figure 1 ~ Raspberry Pi time server block diagram. The Raspberry Pi platform with a GPS receiver on the left provides time service to PCs both on the local area network and on a wide area network such as the internet. The system can be operated with or without external time servers, but operation with external servers is recommended to improve system performance and reliability. The only connections required between the RPi platform and the GPS receiver are power, ground, serial transmit and receive and PPS.

2. Hardware Assembly

This project was developed on an unmodified Raspberry Pi model B+. It most likely will work on the model B but has not been tested. The RPi is operated "headless"; that is, it is used without a directly connected keyboard, mouse or monitor. All provisioning is done from a PC running an SSH terminal program and connected to the same LAN as the RPi. The hardware requirements and costs are shown in table 1.

Table 1 ~ Hardware Requirements and Costs

Description	Adafruit	HAB Supplies	Notes
Base system	Raspberry Pi B+, 35 USD	Raspberry Pi B+, 35 USD	Note 1
GPS receiver assembly	Ultimate GPS Hat for Raspberry Pi {GPSHat}, 45 USD	Raspberry Pi+ GPS Expansion Board {GPSHAB}, 70 USD	Note 1
Battery for GPS RAM	CR1220, 2 USD	CR2032, 2 USD	Note 3
Onboard antenna connector	U.FL-M	SMA-F	
External active antenna	Optional, 13 USD	Required, 15 USD	Note 4
Internal patch antenna	Yes	No	
Flexible jumper cable	Optional, 4 USD	Not required	Note 5
Soldering required	Yes	Yes	Note 6
Mounting hardware required	See hardware list	See hardware list	Note 2
RPi power supply	5 Vdc 5 W	5 Vdc 5 W	Note 7

Table 1 notes:

1. Prices are as-of February 2015 and do not include shipping. The HAB Supplies price includes an active antenna.
2. The Adafruit product does not include any hardware for rigidly mounting the GPS daughterboard to the RPi or for mounting the RPi to a chassis. The HAB Supplies product includes only 2 of 4 spacers needed for the GPS board and none for the chassis mounting. See Table 2..
3. Batteries are user supplied. The HAB Supplies GPS board is incorrectly marked CR2023.
4. Each vendor has related antenna products that can be purchased with the receiver.
5. A flexible jumper is needed between the RPi board connector and an external antenna cable to reduce stress on the connectors.
6. Soldering requirements are minor and described in more detail below.
7. The RPi uses a micro-USB connector for power but it cannot be powered from a PC USB port. Both GPS boards require 3.3 Vdc, which is obtained from the RPi onboard power supply.

Table 2 ~ Hardware list

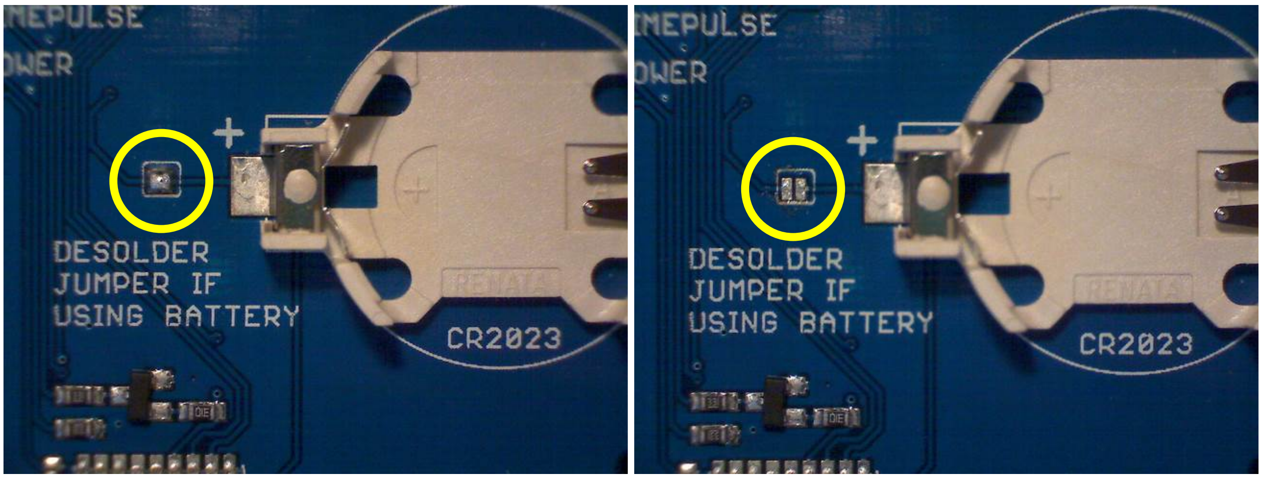
Item	Qty	Remarks
M2.5 x 11 mm stud spacer	4	Note 1
M2.5 x 6 mm stud spacer	4	Note 2
M2.5 hex nut	4	
Flat washer, 2.5 mm	8	
Lock washer, 2.5 mm	8	
M2.5 x 4 mm flat head screw	4	
Chassis, 100 x 70 x 1.6 mm	1	Aluminum
Rubber feet	4	

Hardware List Table notes:

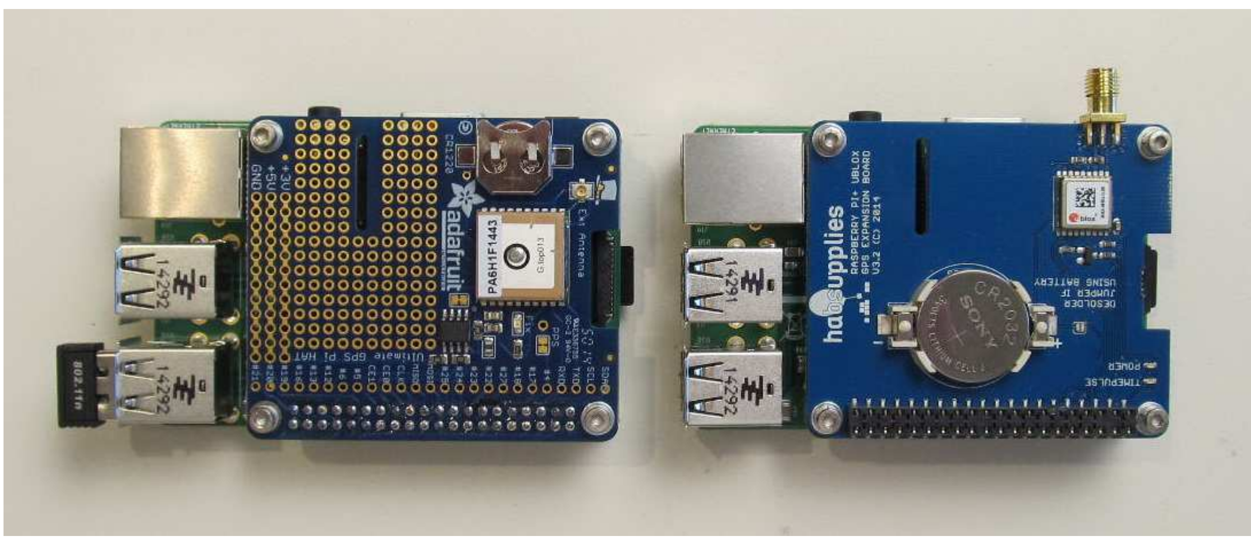
1. Qty can be reduced to 2 if the additional spacers exactly match the length of the spacers supplied.
2. These spacers may be any convenient length to ensure adequate spacing from the aluminum chassis.

Preliminary assembly: Prior to attaching the GPS daughterboard to the RPi board:

- Do not insert the battery until instructed
- **Adafruit GPS board**: Solder the 20x2 connector to the bottom of the PCB. Test fit the connector and when soldering, make sure the connector is flush to the board
- **HAB Supplies GPS board**: Plug the intermediate connector supplied with the board into the bottom of the board; the connector is not soldered. It is necessary to remove a jumper so the battery may be used to retain RAM data. The battery jumper is a blob of solder covering two small pads. It is circled in the images below before (left) and after (right) removal with a soldering iron and solder wick. Note the incorrect silkscreen text below the battery holder; it should read CR2032



Final platform assembly: Attach the GPS daughterboard to the RPi using M2.5 hardware and 11 mm threaded stud standoffs between the two boards. Use M2.5 hardware and 6 mm threaded stud standoffs to mount the RPi board on an aluminum chassis. The assembly should be mounted in a metal enclosure as discussed later.



Battery: After board assembly, insert the battery into the holder on the GPS board. The battery is oriented with the + terminal away from the board. Do not force the battery. It should easily snap or slide in.

Antenna: An outdoor external antenna is recommended but initial setup and testing can be performed indoors. For testing, the antenna should be placed on or near a window, preferably a south-facing window. If the Adafruit Hat is used with its internal patch antenna, the RPi assembly should be placed near a window or on a windowsill. The antenna must have a clear view of the sky. Generally, GPS performance is higher if it is installed to minimize multipath interference caused by reflections from rooftops and walls. The internal patch antenna and external antennas that may be purchased with the Adafruit and HAB Supplies GPS receivers are far from optimum for timing applications because of their irregular gain patterns and nonexistent multipath suppression capabilities. On the other hand, they are cheap and adequate for amateur radio astronomy applications.

The HAB Supplies antenna is shown in the image on the left side and the optional Adafruit antenna is shown right. Antenna dimensions are approximately 50 x 35 x 15 mm. Both antennas have a magnetic base for mounting to a ferromagnetic surface but neither is rated for outdoor service. Both are patch antennas with small coaxial cables and SMA-M connectors. Their manufacturers and specifications are not given by the vendors.

If a different antenna is contemplated, the following guide may be used to select it:

- Polarization: Right-hand circular
- Mode: Active
- Impedance: 50 ohm
- Voltage: 3.0 to 3.6 Vdc (supplied by receiver)
- Current: < 20 mA at 3.3 Vdc (the GTop receiver current limits the antenna bias-tee to 28 mA at 3.3 V)
- Gain: 15 to 50 dBi
- Noise figure: < 1.5 dB

Connecting an external antenna: If an external antenna is to be connected to the Adafruit GPS receiver board, use a 150± mm long flexible jumper between the onboard U.FL-M connector and the SMA-M antenna cable connector (a jumper is available from Adafruit). The flexible jumper is necessary so that no significant force is placed on the onboard connector during operation. If the optional antenna from HAB Supplies is used with their GPS board, there is no need for a flexible jumper (the supplied antenna has RG-174 coaxial cable, already quite small and flexible).

Network: Connect a high-quality Cat5 network cable between the RPi LAN connector and a LAN switch or router. The wired connection is used for provisioning. A WLAN connection can be installed later.

Power Supply: Before connecting a 5 V power supply, read this first: DO NOT attempt to power GpsNtp-Pi using a USB port on a desktop or laptop PC or handheld device. Power measurements of GpsNtp-Pi on the RPi model B+ showed < 1.5 W average dc load. However, when the RPi is booting and depending on connected peripherals, the measured load current can be > 670 mA (3.4 W), too high for a PC USB port.

The system should be powered by a well-regulated, low ripple, electrically quiet power source rated ≥ 1.0 A at 5.0 Vdc (≥ 5 W). Using a poor quality power supply will lead to unreliable operation so be sure it meets these requirements. One of the most common problems users have with the RPi is inadequate power supply current. To minimize electrical noise, it is recommended that ferrite beads be installed on the power cable (image below).

AC wall power adapter rated 12 W at 5 Vdc with North American ac input plug. One or more clamshell ferrite beads on the dc power lead may help reduce radio frequency interference from the switch-mode power supply. This image shows one bead on the right with three windings of the power lead. The Raspberry Pi uses a micro-USB connector for power. (Image © 2014 W. Reeve)

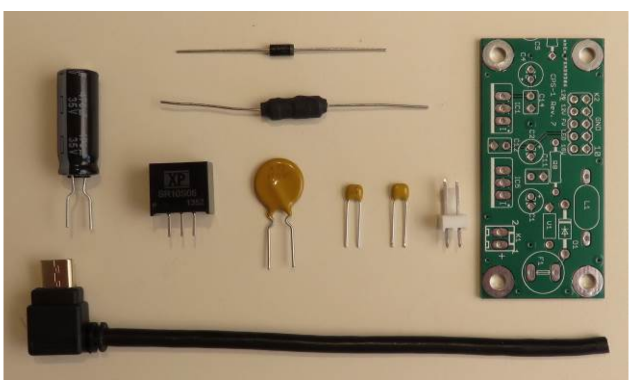
Enclosure: If the GpsNtp-Pi is to be used in a sensitive RF environment, the circuit board assembly should be installed in a metal enclosure. Some considerations are:

- When inside an enclosure, the LED indicators on the RPi and GPS circuit boards will be hidden. One way to handle this is to drill a small hole (~ 2 mm) in the enclosure above the GPS LEDs, so they will be visible through the hole.
- The 5 Vdc power input to the RPi is through a micro-USB connector, which will be inaccessible after the boards are enclosed. Use a jumper from the RPi board to the rear panel where a power connector is installed.
- A small dc-dc converter may be installed in the enclosure to convert from, say, 12 Vdc input to 5 Vdc for the RPi. This normally will require an input pi filter for the converter. As above, a jumper will be required from the RPi board to the converter board. If a converter is used, it will be necessary to check its specified maximum output filter capacitor size. The RPi board has a 220 µF power input filter capacitor, so the converter will have to handle at least that much. In addition to the pi filter at the converter input, MLCC capacitors rated 10 nF should be placed in parallel with the converter input and output.
- The RPi circuit board will have to be positioned in the enclosure so the USB and Ethernet connectors are accessible. A rectangular cutout in the front panel will allow the board to be installed flush with the panel.
- The antenna input on the GPS board will have to be extended to the front panel.

Below are pictures of the GpsNtp-Pi installed in a Box Enclosures B3-160 extruded aluminum enclosure and dc-dc converter power supply. Enclosure dimensions are 160 long x 108 mm wide x 45 mm high. All holes in the enclosure were cut using engineering drawings produced specifically for the enclosure. A 5 W dc-dc converter was installed on a CPS-1 printed circuit board with filter components and PTC resettable fuse and polarity guard diode. Power input at 12 Vdc is through a 2.1 x 5.5 mm coaxial dc power connector.


GNSS Antenna
GpsNtp-Pi
plugable


Power

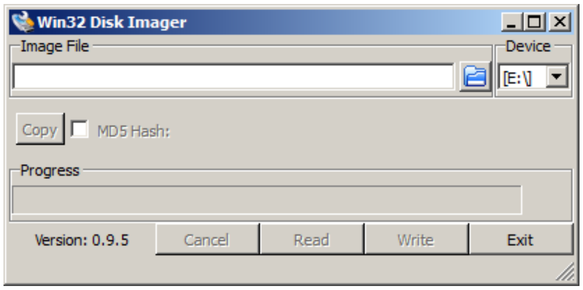


3. Operating System Installation

The system uses the Raspbian distribution, which is a version of Linux. It is necessary to download and install the distribution image and install it on a memory card in a format that is compatible with the RPi. It is not possible to simply copy the distribution image from a Windows PC to a memory card; instead, a special image writer program is needed. Suitable programs are Win32 Disk Imager {WDImg} and HDD Raw Copy Utility {HDDRaw}. This guide is based on Win32 Disk Imager. Download and install the program, and place a shortcut to the program on the Desktop. Mark the shortcut to run as Administrator.

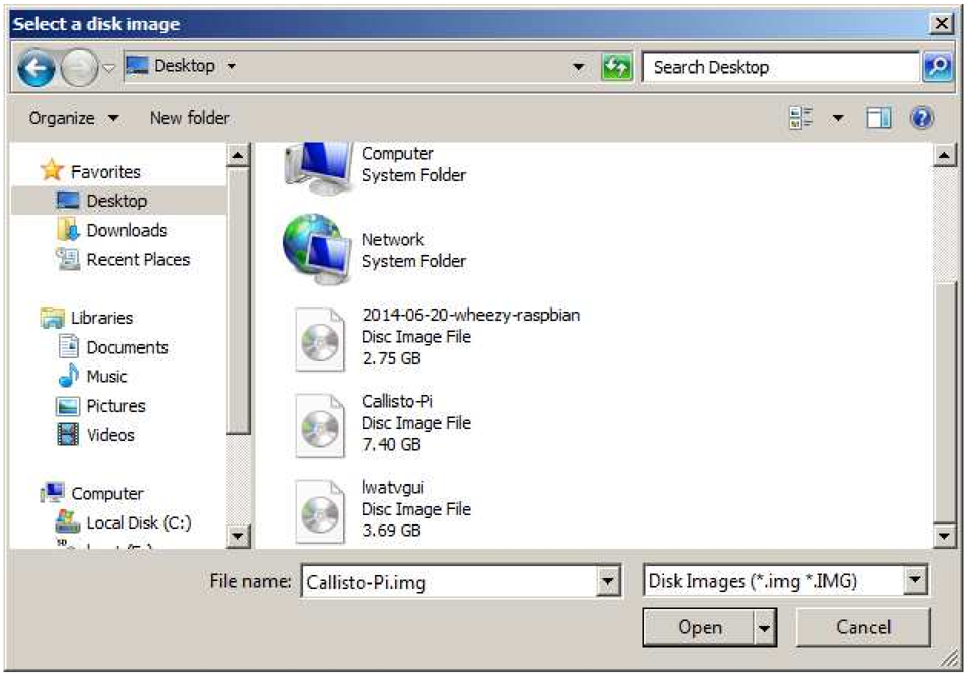
Now, download the Raspbian distribution from {Raspbian} and extract it to the Desktop. Write this image to a 4 GB or larger micro-SD memory card as follows:

Connect a card reader/writer suitable for a micro-SD memory card to the PC. Insert the SD or micro-SD memory card into the card reader/writer. Some older micro-SD cards and card reader/writers are incompatible, so it may be necessary to install the micro-SD card in a full-size SD card carrier or adapter and insert the carrier into the reader. Now, determine the memory card's drive letter from My Computer or Explorer. Open Win32 Disk Imager.

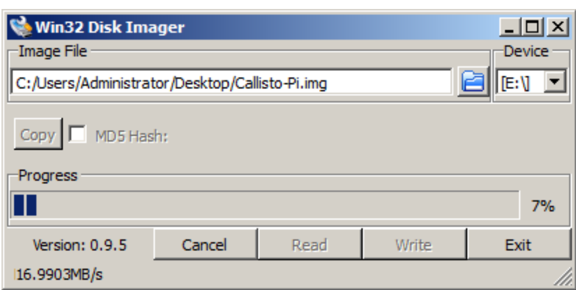


A drive letter is shown in the Device field in upper-right of Win32 Disk Imager. If you have more than one device plugged into the PC, select the letter corresponding to your memory card (drive letter E: in example above).

Click the folder icon to right of the Image File input field. Windows Explorer will open (below). Navigate to the location where the Raspbian image is located and select the file. In the example below, the image file is located on the Desktop but it has not yet been selected. Click Open.



At this time you will be returned to the Win32 Disk Imager user interface. Click the **Write** button. After a moment the Progress bar will show the Write operation, which requires several minutes. Be patient.



When the Write process finishes, a Success window will pop up.

Eject the memory card by right-clicking the drive letter in My Computer or Explorer and selecting Eject. Remove the memory card from the reader/writer.

<u>Memory card installation</u>: Insert the micro-SD memory card into the slot connector on the bottom of the RPi board. The side of the memory card with the contacts (see right) should face the board. The connector is push-to-insert/push-to-release and is self-latching. The card cannot be inserted the wrong way. Gently push the card all the way in and release finger pressure; the card should remain in the holder and not spring out. To release, push again and the card will spring out.

Power connection: Connect the 5 V power supply to the micro-USB connector on the RPi. Refer to table 2 and watch the LEDs on the RPi board. If the red LED does not illuminate, there is a problem with the power supply. If the green Activity LED turns on and stays on, the memory card may be corrupted.

Table 2 ~ RPi LED Indicators

Condition		
Power off	Power LED, Red: OFF	Activity LED, Green: OFF
Power on, Boot	Power LED, Red: ON	Activity LED, Green: Flashes during boot
Power on, Ready	Power LED, Red: ON	Activity LED, Green: Occasional flash
Ethernet connector	Left LED, Yellow: ON 100 Mb/s, OFF 10 Mb/s	Right LED, Green: ON link established, Flashes activity OFF if no link

The GPS receivers operate autonomously and will try to acquire satellites as soon as power is available from the RPi board. Even if the RPi itself is not setup to use the GPS receiver outputs, the receivers will operate in a normal way. The GPS receiver boards have one (Adafruit) or two (HAB Supplies) LED indicators. See table 3.

Table 3 ~ GPS Receiver LED Indicators

Receiver condition	**Adafruit ~ GTop Receiver**	**HAB Supplies ~ uBlox Receiver**
Power off	Status LED, Red: OFF	Power LED, Red: OFF Timepulse LED, Green: OFF
Power on, Acquiring satellites	Status LED, Red: Flash 1 per second	Power LED, Red: ON Timepulse LED, Green: OFF
Power on, Fix obtained, tracking	Status LED, Red: Flash 1 per 15 seconds	Power LED, Red: ON Timepulse LED, Green: Flash 1 per second

All LED indicators on the RPi board except the red power LED will extinguish when the shutdown is complete, requiring only a moment. The LEDs on the GPS receiver boards will not change. Power may then be removed from the RPi board.

Provisioning: In the following sections, the RPi will be provisioned over a wired network connection using an SSH terminal program. The network connection type can be changed later to a wireless connection as described in {WLAN}. However, it is expected that the jitter performance will be much worse when the time server is operated over a wireless connection. This caution would apply both when the GpsNtp-Pi uses a server pool for synchronization and when an NTP client on a PC accesses the GpsNtp-Pi for synchronization.

Each software function in the RPi software has a “manual”, which can be accessed by typing at the prompt **`man`** followed by the function name. For example, to read the manual for the NTP daemon, type **`man ntpd`** and for the GPS daemon, type **`man gpsd`**. These manuals will help you understand the changes specified.

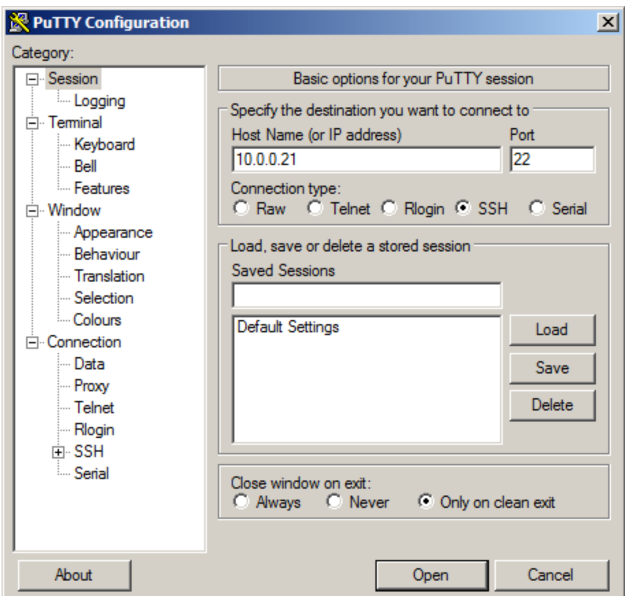
This operating system and software is modified and provisioned as described in the following sections.

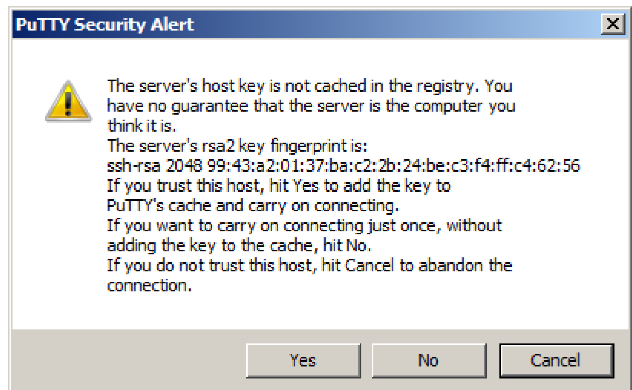
- **`Text in Blue is typed at the RPi prompt`**
- **`Text in Red indicates editing inside a configuration file`**

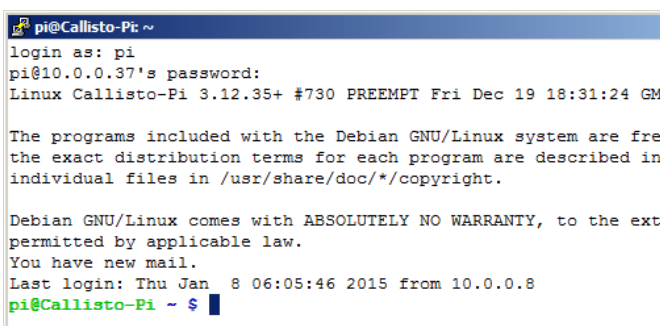
Backup your RPi software images after provisioning. See {BkRsRPi} for procedures.

4. Install a Secure Shell Terminal Program

This guide is based on PuTTY {PuTTY}, which should be downloaded and installed now. Open PuTTY and enter the RPi IP address into the Host **Name or IP address** field. Do not change the port (22) or Connection type (SSH). You can determine the RPi's IP address from your router LAN status screen or by using a program like Netscan {Netscan}. Enter a name for the session (example, GpsNtp-Pi) in the Saved Sessions field. You also can customize the user interface by clicking on the various entries in the Category window but you can do this later. Click the Save button to save the session settings.







Upper-left: Screenshot of PuTTY configuration window with the IP address for a test setup. Your installation will have use a different address. Upper-right: Upon first connection, PuTTY asks for authentication; click Yes. Lower-left: When PuTTY is connected, the window changes to a simple command line interface. After login, you can make changes to the RPi as described in this guide. (Images © 2014 W. Reeve)

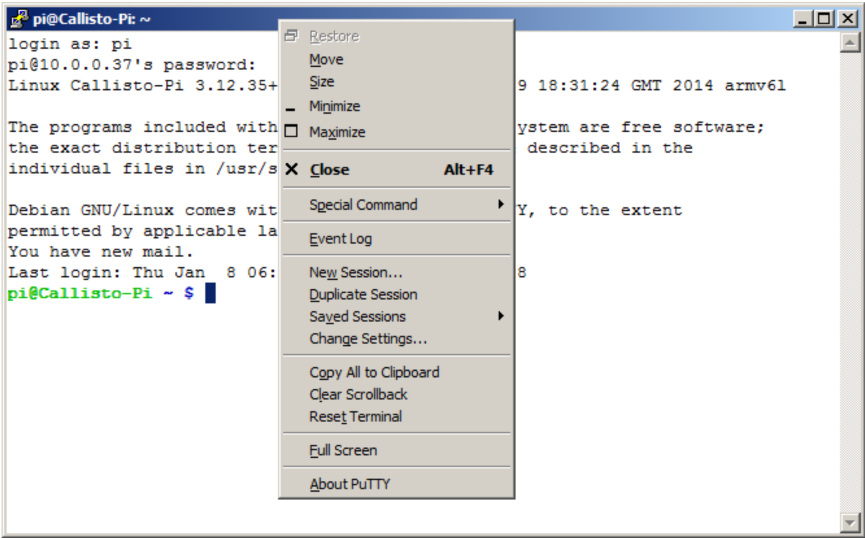
Click Open. After authentication you will see the RPi login prompt in the command line interface. Enter the username and then the password. You will not see characters on the console as you enter the password.

Default username: pi
Default password: raspberry

After logging into RPi, you will see the prompt

pi@raspberry – $

At any time you can right-click the bar at the top of the PuTTY window to see the PuTTY menu. Click Change Settings ... if you would like to revise the user interface and session settings.



In the next section, you will change the default password. You will need the new password to log into RPi for all setup and provisioning activities.

Note: If it becomes necessary to remove power from the RPi, the RPi must be properly shutdown before removing power (just like a Windows PC). Never remove power without first shutting down the RPi. To shutdown, log into RPi as described later using the SSH terminal. At the prompt enter:

```
sudo halt
```

or

```
sudo shutdown -h now
```

The next several sections install GPS and NTP software on the Raspberry Pi. This process requires approximately 2 hours, of which most time is absorbed by compiling operations.

5. Basic RPi Setup

The basic setup includes changing the host name and password and updating the operating system software.

Log into the RPi using PuTTY. First, set the default keyboard layout. Using the nano editor, replace the xkblayout=”gb” with xkblayout=”us”:

```
sudo nano /etc/default/keyboard
```

Type CTRL-X (Exit), Y (yes), Enter to save and close the editor.

Change host name: To change the host name to GpsNtp-Pi, edit the /etc/hosts file:

```
sudo nano /etc/hosts
```

Change the name associated with address 127.0.1.1 (do not change 127.0.0.1), as in:

```
127.0.1.1       GpsNtp-Pi
```

Type CTRL-X (Exit), Y (yes), Enter to save and close the editor.

Similarly, replace the name in the /etc/hostname file:

```
sudo nano /etc/hostname
```

Change the name from raspberry to GpsNtp-Pi, as in

```
GpsNtp-Pi
```

Type CTRL-X (Exit), Y (yes), Enter to save and close the editor.

Change password: Log into the RPi using the defaults:

User: pi
Password: raspberry

At the prompt enter the following command:

```
passwd
```

RPi responds with:

```
Changing password for pi
(current) UNIX password:
```

Enter the current (default) password (the default Raspberry Pi password is “raspberry” without quotes). No characters will be displayed. Enter the new password (the default GpsNtp-Pi password is “GpsNtp-Pi” without quotes) and confirm it when RPi responds with:

```
Enter new UNIX password:
Retype new UNIX password:
```

After RPi accepts the new password you will see the following. The new password will take effect immediately:

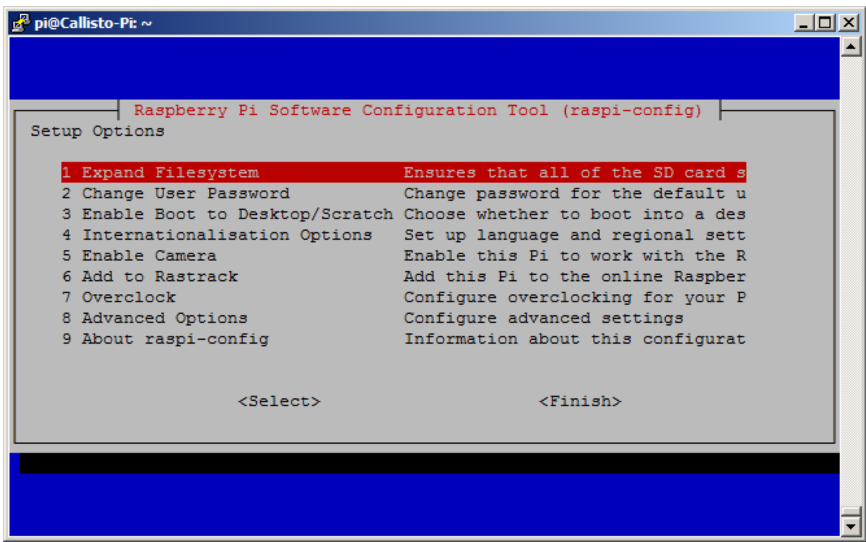
```
passwd: password updated successfully
```

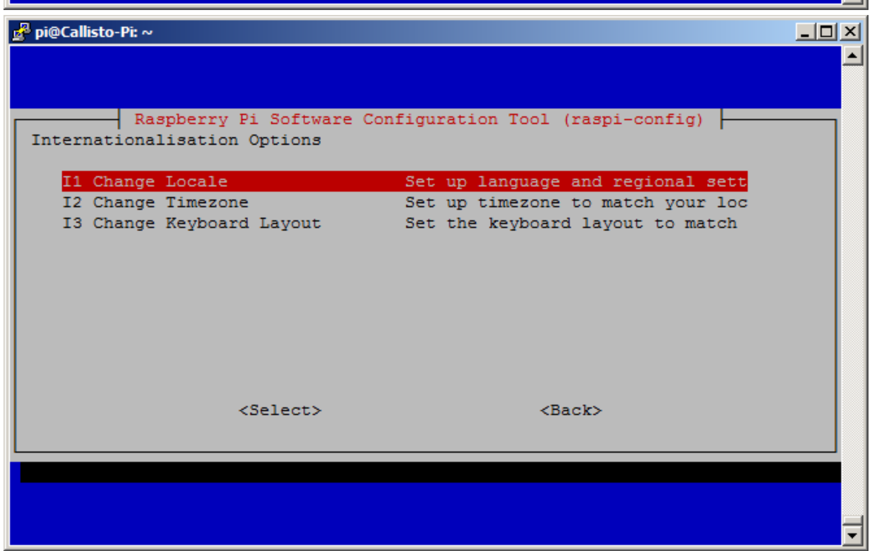
Enter the following command at the prompt. After a moment a window will appear with a list of settings. Use the arrow keys to select the desired option, press Tab to Select and then Enter:

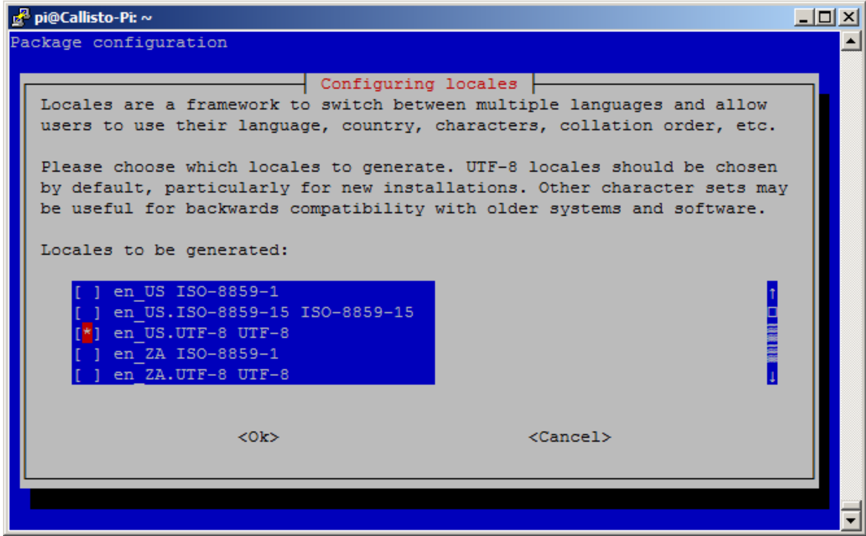
```
sudo raspi-config
```

- Select Option 1. Expand filesystem
- Select Option 4. Internationalization Options – Option 11. Change Locale – en_US.UTF-8 UTF-8. Use the spacebar to deselect en_GB.UTF-8 UTF-8. Scroll down to en_US.UTF-8 UTF-8 and select it with the spacebar. Scroll through the entire list to make sure only one Locale is selected
- Select Option 4. Internationalization Options – Option 13. Change Keyboard Layout
- Select Option 8. Advanced Options – A8 Enable or Disable Serial Shell – Accessible – No
- Finish

Example screenshots are shown below.







After initial setup:

```
sudo reboot
```

Now it is necessary to update the software and operating system. These steps may require several minutes:

```
sudo apt-get update
sudo apt-get dist-upgrade
sudo rpi-update
sudo reboot
```

6. Install and Verify Pulse Per Second ~ PPS

Install software to support PPS:

```
sudo apt-get install pps-tools
sudo apt-get install libcap-dev
```

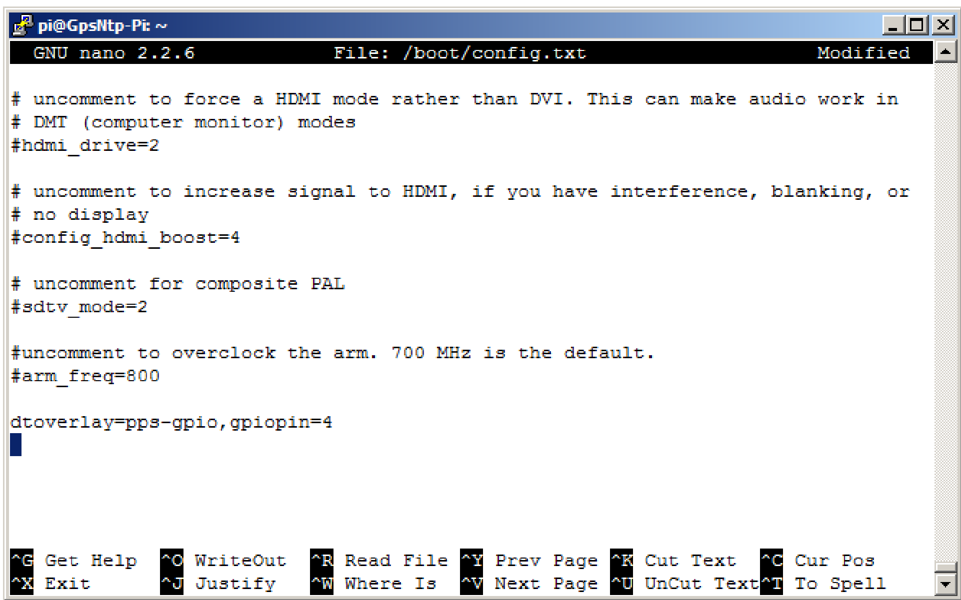
Configure the RPi to use the general purpose input/output (GPIO) for the PPS input from the GPS receiver:

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

Add a new line as shown below. The value X = 4 or 18 depending on GPS board. The HAB GPS uses GPIO pin 18 and the Adafruit GPS uses GPIO pin 4.

```
dtoverlay=pps-gpio,gpiopin=X
```

The PuTTY window should look similar to this (this example shows the pin used with the Adafruit GPS board):



Type CTRL-X (Exit), Y (yes), Enter to save and close the editor.

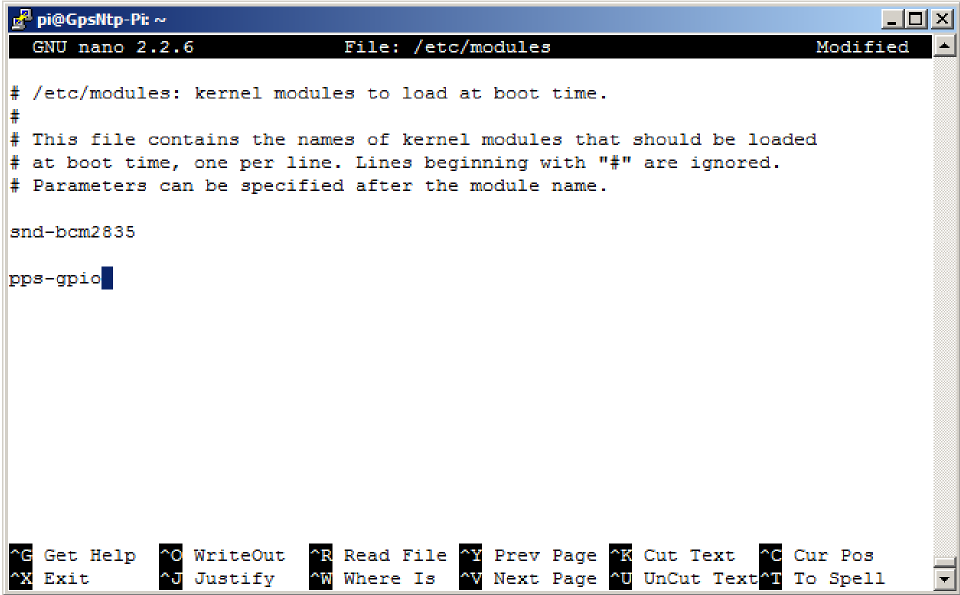
Additional changes are required to use PPS

```
sudo nano /etc/modules
```

Add a new line

```
pps-gpio
```

The PuTTY window should look similar to this:



Type CTRL-X (Exit), Y (yes), Enter to save and close the editor, and then reboot:

```
sudo reboot
```

After reboot, log into the RPi and check that the module is loaded:

```
lsmod | grep pps
```

The output should be similar to:

```
pps_gpio 2529 1
pps_core 7943 2 pps_gpio
```

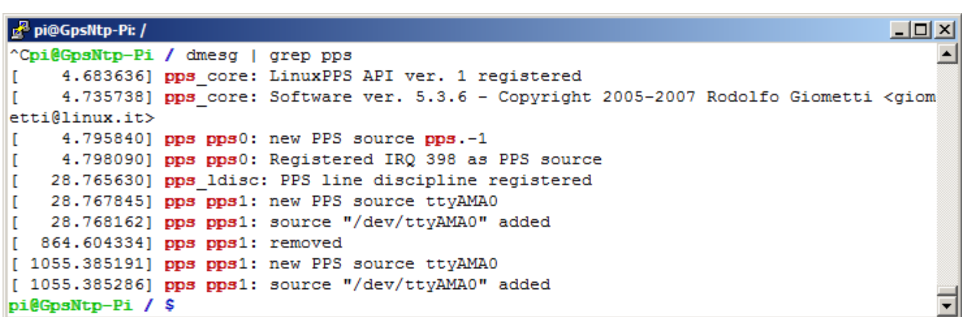
To determine if PPS is working, first ensure the GPS receiver is in the satellite tracking mode. This may require up to 15 minutes from a cold start:

- On the HAB Supplies GPS board, the green PPS LED flashes about 1 per second when tracking
- On the Adafruit GPS, the red GPS status LED flashes once every 15 seconds when tracking

Next, enter:

```
dmesg | grep pps
```

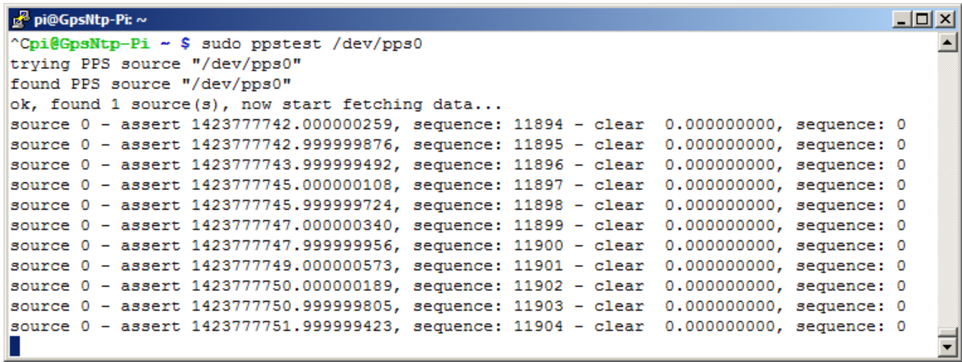
Output should be similar to shown below, indicating that the PPS software module is loaded. Your window may show only the first three lines. The third line indicates there is a new PPS source pps.-1:



Also, check:

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

If the PPS software module is working, output should be similar to shown below with the source 0 lines scrolling 1 line/s. Press CTRL-C to quit:



```
pi@GpsNtp-Pi: ~
^Cpi@GpsNtp-Pi ~ $ sudo ppstest /dev/pps0
trying PPS source "/dev/pps0"
found PPS source "/dev/pps0"
ok, found 1 source(s), now start fetching data...
source 0 - assert 1423777742.000000259, sequence: 11894 - clear  0.000000000, sequence: 0
source 0 - assert 1423777742.999999876, sequence: 11895 - clear  0.000000000, sequence: 0
source 0 - assert 1423777743.999999492, sequence: 11896 - clear  0.000000000, sequence: 0
source 0 - assert 1423777745.000000108, sequence: 11897 - clear  0.000000000, sequence: 0
source 0 - assert 1423777745.999999724, sequence: 11898 - clear  0.000000000, sequence: 0
source 0 - assert 1423777747.000000340, sequence: 11899 - clear  0.000000000, sequence: 0
source 0 - assert 1423777747.999999956, sequence: 11900 - clear  0.000000000, sequence: 0
source 0 - assert 1423777749.000000573, sequence: 11901 - clear  0.000000000, sequence: 0
source 0 - assert 1423777750.000000189, sequence: 11902 - clear  0.000000000, sequence: 0
source 0 - assert 1423777750.999999805, sequence: 11903 - clear  0.000000000, sequence: 0
source 0 - assert 1423777751.999999423, sequence: 11904 - clear  0.000000000, sequence: 0
```

7. Install GPS Daemon and Associated Tools

Install GPSD and GPSD-Clients:

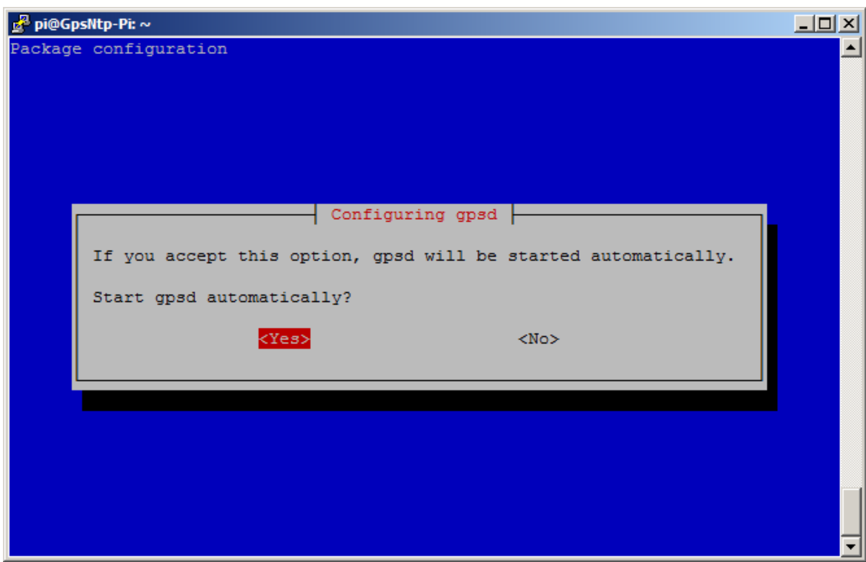
```
sudo apt-get install gpsd gpsd-clients
```

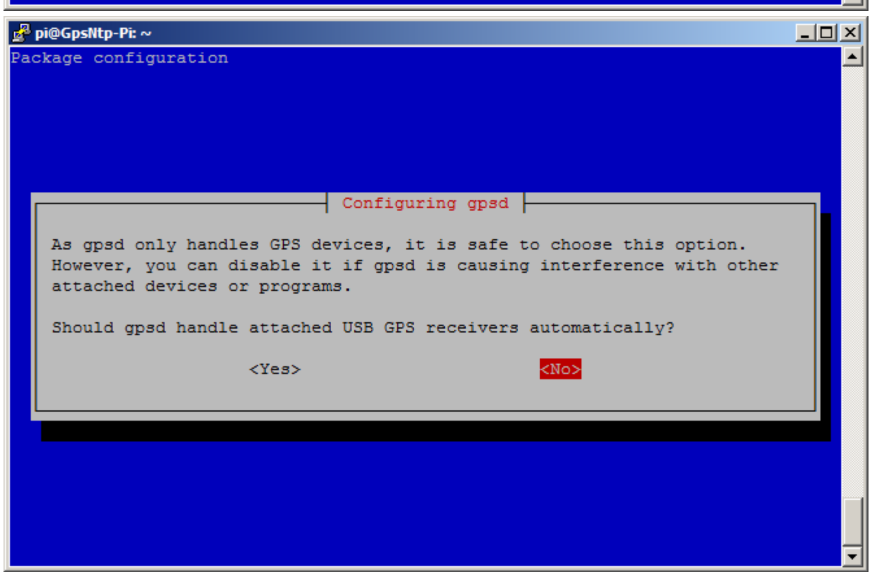
After installation, configure GPSD:

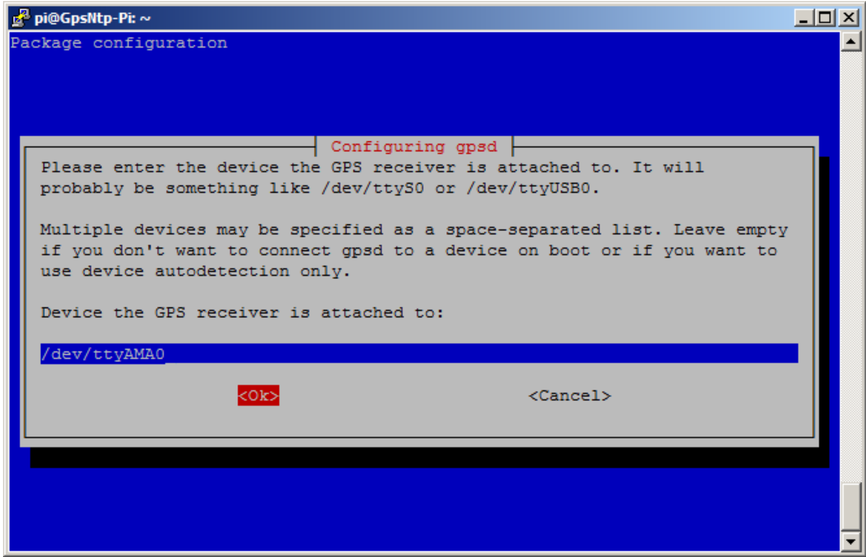
```
Sudo dpkg-reconfigure gpsd
```

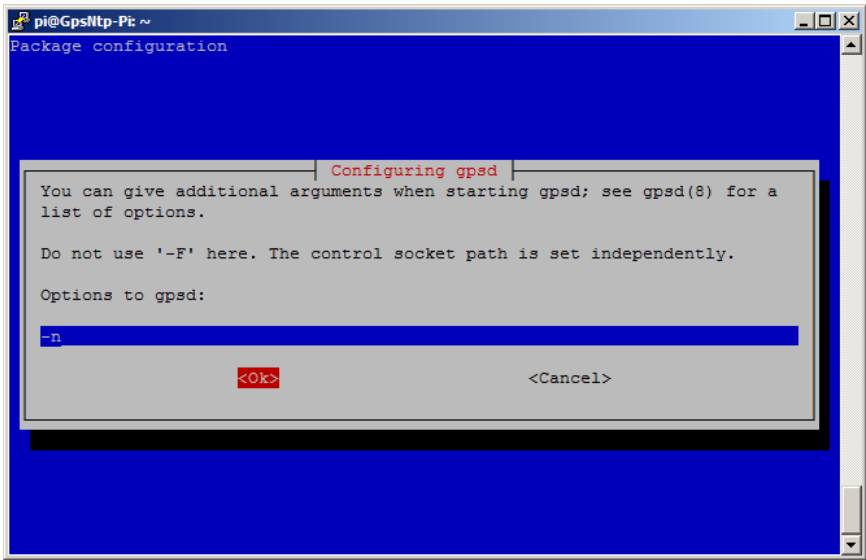
Follow the prompts to:

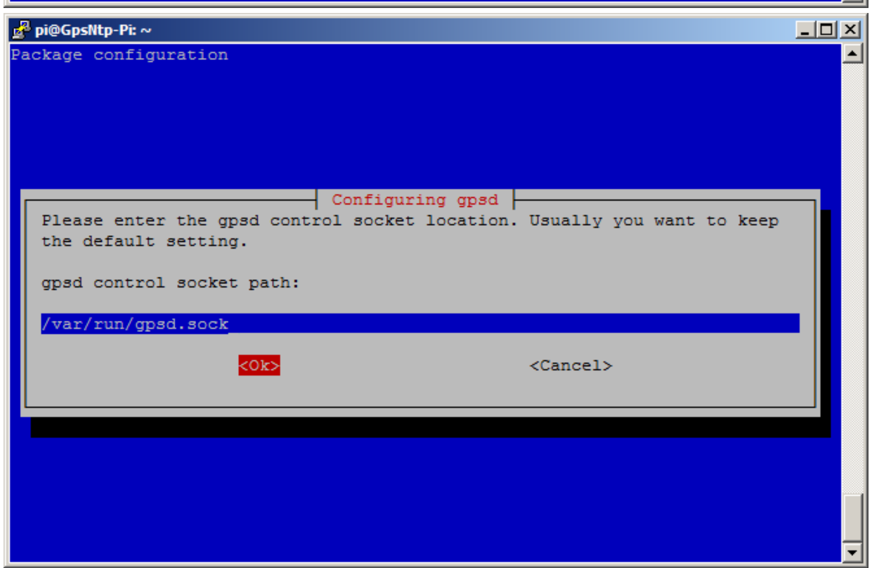
- Start gpsd automatically: Yes
- Automatically handle USB devices: No
- Assign GPS device: /dev/ttyAMA0 (trailing zero not "O")
- Add option: -n
- Assign GPSD control socket location: /var/run/gpsd.sock











After GPSD has been configured, reboot:

```
sudo reboot
```

Log into the RPi and test the GPS receiver time and position data output. Before running the next command, set the PuTTY window to 80 columns x 30 rows (in PuTTY Change Settings... – Window). To quit the GPS monitor, press CTRL-C:

```
gpsmon
```

The upper part of the console display will remain fixed while the enclosed data updates automatically. The lower part scrolls as incoming data are displayed. Of particular interest in the scrolling data are the $GPRMC parameters, which show position, time and date (if necessary, click and drag the window height to display more dynamic data but do not increase its width). The various parameters and how to interpret them are described in {NMEA}. The data in the GPS monitor continuously updates. Enter CTRL-S to freeze and CTRL-Q to unfreeze.

```
pi@GpsNtp-Pi: ~
localhost:2947:                    Generic NMEA>
lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
xTime: 2015-02-12T19:19:37.000Z Lat:  61 11' 57.947" N Lon: 149 57' 23.813" W  x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq Cooked PVT qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
x GPGGA GPGSA GPRMC GPVTG GPGSV                                                x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq Sentences qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
lqqqqqqqqqqqqqqqqqqqklqqqqqqqqqqqqqqqqqqqqqqqqqqqqqklqqqqqqqqqqqqqqqqqqqqqqqqqqqk
xCh PRN  Az El S/N xxTime:       191937.000       xxTime:      191938.000      x
x 0  32 148 80  20 xxLatitude:     6111.9658 N    xxLatitude:  6111.9658       x
x 1   1 221 68  28 xxLongitude:   14957.3969 W    xxLongitude: 14957.3968      x
x 2   4 168 48  18 xxSpeed:      0.09             xxAltitude:  0.4             x
x 3   3 259 46  32 xxCourse:     95.431           xxQuality:   1   Sats: 10    x
x 4  11 197 45  17 xxStatus:     A        FAA: A  xxHDOP:      0.92            x
x 5  14  64 42  26 xxMagVar:                      xxGeoid:     9.6             x
x 6  17 308 31  24 xmqqqqqqqqqqq RMC qqqqqqqqqqqqjmqqqqqqqqqqq GGA qqqqqqqqqqqqj
x 7  31 120 24  22 xlqqqqqqqqqqqqqqqqqqqqqqqqqqqqqklqqqqqqqqqqqqqqqqqqqqqqqqqqqk
x 8  12  27 10  27 xxMode: A 3                    xxUTC:           RMS:        x
x 9  22  92  8  14 xxSats: 11 32 4 31 17 1 14 22 xxMAJ:           MIN:        x
x10  23 214  4   0 xxDOP: H=0.92  V=0.85  P=1.25 xxORI:           LAT:        x
x11  41   0  0   0 xmqqqqqqqqqqq GSA qqqqqqqqqqqqjxLON:           ALT:        x
mqqqqqq GSV qqqqqqqj                              mqqqqqqqqqqq GST qqqqqqqqqqqqj
(62) $GPGSA,A,3,11,32,04,31,17,01,14,22,03,12,,,1.25,0.92,0.85*07\x0d\x0a
(71) $GPRMC,191937.000,A,6111.9658,N,14957.3969,W,0.09,95.43,120215,,,A*41\x0d\x
0a
(38) $GPVTG,95.43,T,,M,0.09,N,0.17,K,A*09\x0d\x0a
(71) $GPGGA,191938.000,6111.9658,N,14957.3968,W,1,10,0.92,0.4,M,9.6,M,,*4B\x0d\x
0a
(62) $GPGSA,A,3,11,32,04,31,17,01,14,22,03,12,,,1.25,0.92,0.85*07\x0d\x0a
```

Alternately, run a more compact monitor. The values will appear after a moment and then continuously update:

```
cgps -s
```

```
pi@GpsNtp-Pi: /
lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqklqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
x    Time:       2015-02-11T23:46:02.000Z     xxPRN:   Elev:  Azim:  SNR:  Used: x
x    Latitude:    61.199421 N                 xx   9    80    153    26      Y   x
x    Longitude:  149.956404 W                 xx  10    61    258    23      Y   x
x    Altitude:   5.6 ft                       xx  20    55    196    21      Y   x
x    Speed:      0.6 mph                      xx  23    51    104    18      Y   x
x    Heading:    227.8 deg (true)             xx   7    46    198    20      Y   x
x    Climb:      19.7 ft/min                  xx  16    36    079    23      Y   x
x    Status:     3D FIX (4 secs)              xx   2    30    284    29      Y   x
x    Longitude Err:   +/- 33 ft               xx   6    21    235    26      Y   x
x    Latitude Err:    +/- 31 ft               xx  30    17    210    23      Y   x
x    Altitude Err:    +/- 61 ft               xx   5    14    310    20      Y   x
x    Course Err:      n/a                     xx  29    12    345    17      Y   x
x    Speed Err:       +/- 45 mph              xx   3    05    148    00      N   x
x    Time offset:     0.855                   xx  38    00    000    00      N   x
x    Grid Square:     BP51ae                  xx                                 x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqjm                               qj
```

8. Enable PPS Support in NTP Daemon

The NTP daemon supplied with the Raspbian distribution for the RPi does not natively support PPS so it needs to be recompiled.

To recompile ntp, first check http://archive.ntp.org/ntp4/ for the latest production version of NTP. If different than shown below, modify the strings accordingly. The file will have the general form **ntp-4.2.8px.tar.gz**, where x is the latest patch number (patch 8 is shown below). Next, run the following commands at the RPi prompt. Note: These steps may require up to 30 minutes and there will be long periods with no apparent activity on the SSH terminal:

```
wget http://www.eecis.udel.edu/~ntp/ntp_spool/ntp4/ntp-4.2/ntp-4.2.8p8.tar.gz
```

~~wget http://archive.ntp.org/ntp4/ntp-4.2.8p8.tar.gz~~ [old address, do not use]

```
tar zxvf ntp-4.2.8p8.tar.gz
cd ntp-4.2.8p8/
./configure --enable-linuxcaps
```

`make` **(see note below)**

```
sudo make install
```

Note: For the RPi 2, which has 4 cores, use `make -j5` instead of `make` in the above series of statements.

Copy the recompiled files to their destination (enter the following very carefully to avoid typing mistakes):

```
sudo service ntp stop
sudo cp /usr/local/bin/ntp* /usr/bin/
sudo cp /usr/local/sbin/ntp* /usr/sbin/
```

Now, modify the NTP configuration file to use the GPSD and PPS servers:

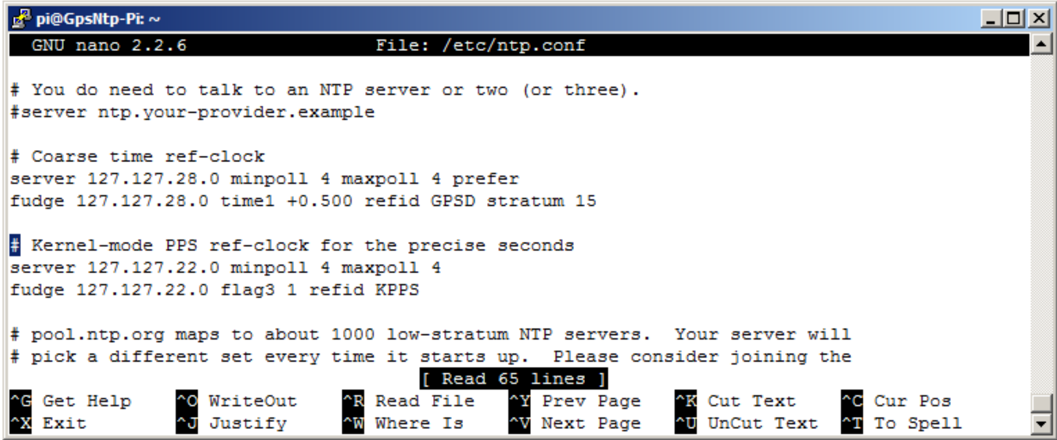
```
sudo nano /etc/ntp.conf
```

Add the following lines in the configuration file to specify the GPSD shared memory and kernel-mode PPS drivers. They can be placed above the server pool list (see screenshot below). The time1 parameter in the shared memory driver (127.127.28.0) probably will require adjustment but that will be covered later.

```
# Coarse time ref-clock using shared memory GPS serial data
server 127.127.28.0 minpoll 4 maxpoll 4 prefer
fudge 127.127.28.0 flag1 1 time1 +0.500 refid GPSD

# Precise time ref-clock using Kernel-mode PPS from GPS
server 127.127.22.0 minpoll 4 maxpoll 4
fudge 127.127.22.0 refid KPPS
```

Note the flag1 setting in the shared memory driver for GPSD; this setting is for standalone operation in which initial time is established by the GPS serial data. If true standalone operation (no access to any other time server), these lines should be entered as shown above. However, if the RPi time server will have access to external servers, the flag1 1 setting can be deleted. Additional information on the fudge factors for the shared memory and PPS drivers is given at the end of this section. Additional information on the various reference clock drivers is at {NTPRefClk}.



Next, scroll down to the list of four servers labeled *server 0.debian..., server 1.debian...* and so on. Delete the four servers (or comment them out with the # character).

Now, enter the NTP server pool closest to your country; see {NTPPool}. The United States NTP server pool is shown here. Note that the statement is preceded by the comment character #, which means the line will be ignored. This is necessary to test the standalone operation. The comment character can be removed later as described in **section 11**:

```
# pool us.pool.ntp.org iburst minpoll 6 prefer
```

To log statistics for later viewing and analysis, remove the comment character # from the beginning of the line that starts with *statdir*. If statistics are not required, leave the comment character:

```
statsdir /var/log/ntpstats/
```

The NTP configuration file has several restrict statements with default settings. Two, in particular, require modification. Find these statements (they may be separated by comments):

```
restrict -4 default kod notrap nomodify nopeer noquery
restrict -6 default kod notrap nomodify nopeer noquery
restrict 127.0.0.1
restrict ::1
```

Insert the following statement and comment out the first two lines, as in:

```
restrict source notrap nomodify nopeer query

# restrict -4 default kod notrap nomodify nopeer noquery
# restrict -6 default kod notrap nomodify nopeer noquery
```

Do not change:

```
restrict 127.0.0.1
restrict ::1
```

Depending on how the NTP server will be used, it may be necessary to implement additional access restrictions but that is not covered here. The restrict parameters are more fully documented here: http://www.eecis.udel.edu/~mills/ntp/html/accept.html.

Type CTRL-X (Exit), Y (yes), Enter to save and close the editor.

Restart the NTP service:

```
sudo service ntp restart
```

After a few minutes run NTP Query (all query command options are defined in {NTPQ}:

```
ntpq -p
```

```
pi@GpsNtp-Pi: ~
pi@GpsNtp-Pi ~ $ ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*SHM(0)          .GPSD.           0 l   15   16  377    0.000   43.460  59.293
oPPS(0)          .KPPS.           0 l   14   16  377    0.000    1.598   0.144
pi@GpsNtp-Pi ~ $
```

The * in the first column of the first line indicates the shared memory GPSD reference clock is being used. The o in the first column of the second line indicates the kernel-mode PPS source is selected and being used. Other symbols may appear in the first column; see table at end of this section. To see additional details, use more command line options in the query tool. In the example display below the remote column shows the local address of the KPPS (driver designated .22) rather than its name PPS(0):

```
ntpq -crv -pn
ntpq -crv -p
```

```
pi@GpsNtp-Pi: ~
pi@GpsNtp-Pi ~ $ ntpq -crv -pn
associd=0 status=0115 leap_none, sync_pps, 1 event, clock_sync,
version="ntpd 4.2.8p1@1.3265 Sun Feb  8 00:48:57 UTC 2015 (1)",
processor="armv6l", system="Linux/3.18.6+", leap=00, stratum=1,
precision=-18, rootdelay=0.000, rootdisp=1.150, refid=KPPS,
reftime=d887a4e6.79b0423e  Thu, Feb 12 2015 22:17:10.475,
clock=d887a4f0.b717f49a  Thu, Feb 12 2015 22:17:20.715, peer=5285, tc=4,
mintc=3, offset=0.001048, frequency=-13.350, sys_jitter=0.003815,
clk_jitter=0.004, clk_wander=0.000, tai=35, leapsec=201507010000,
expire=201512280000
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*127.127.28.0    .GPSD.           0 l   11   16  377    0.000    6.467  30.461
o127.127.22.0    .KPPS.           0 l   10   16  377    0.000    0.001   0.004
pi@GpsNtp-Pi ~ $
```

If you repeat the query command over periods of tens of minutes, you should see the PPS offset slowly decrease toward zero. If it does not, you may have a typing error in ntp.conf. The first query shown in the example above shows a PPS offset of 1.598 ms and the next query above, taken some time later, shows an offset of 0.001 ms (1 μs).

If you do not see the PPS source, it is possible the NTP server is picking up the NTP information via DHCP, which is overriding the settings above. Do this:

```
sudo rm /etc/dhcp/dhclient-exit-hooks.d/ntp
sudo rm /var/lib/ntp/ntp.conf.dhcp

sudo reboot
```

After reboot, log into the RPi. To view various statistics that were enabled in the ntp.conf file, use WinSCP to go to the root directory

`/var/log/ntpstats/`

This directory includes two log files, loopstats and peerstats. Right-click the file and select Open. For a description of the log data go to {NTPLog}.

Driver fudge factors in ntp.conf:
The following provides additional information on a few fudge factors.

The fudge factor for the shared memory driver (.28) discussed above uses the flag1 and time1 parameters. An additional parameter, not explicitly specified, is the time2 parameter. The time2 parameter specifies the allowable time difference between the internal time and time provided by a reference clock. It can be set between 1 s and 86 400 s (24 h). If not explicitly included in the shared memory driver fudge factor statement, the time2 default is 4 h.

The time2 and flag1 parameters interact. If the flag1 parameter is not explicitly included, the driver uses its default value of 0 and the time difference limit check is made. The time difference limit check is skipped when flag1 is set to 1. The stated purpose of setting flag1 to 1 is "*for systems where the RTC backup cannot keep the time over long periods without power and the SHM clock must be able to force long-distance initial jumps*."

The PPS driver (.22) normally does not require any fudge factor other than to assign a reference ID. However, if Allan deviation plots are to be made of the PPS offset data, flag4 1 can be specified, as in:

```
fudge 127.127.22.0 flag4 1 refid KPPS
```

When flag4 is set, the driver records a timestamp once for each second. The data are stored in clockstats file in directory /var/log/ntpstats/. The clockstats and other statistics may be accessed using the cat command or WinSCP.

Note for future NTP updates: NTP is occasionally updated. However, the `sudo apt-get upgrade` command used to upgrade the RPi does not automatically upgrade the NTP daemon. To update the NTP daemon, run the RPi update and upgrade and then recompile NTP as follows:

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

wget http://www.eecis.udel.edu/~ntp/ntp_spool/ntp4/ntp-4.2/ntp-4.2.8p8.tar.gz
tar zxvf ntp-4.2.8p8.tar.gz
cd ntp-4.2.8p8/
./configure --enable-linuxcaps
make
sudo make install

sudo service ntp stop
sudo cp /usr/local/bin/ntp* /usr/bin/
sudo cp /usr/local/sbin/ntp* /usr/sbin/
sudo service ntp start
```

It should not be necessary to reconfigure the NTP configuration file, ntp.conf, but the file should be checked anyway. Be sure to backup the memory card before updating and upgrading. Using NTP v4.2.8p9 and later may not work with the Raspbian Wheezy operating system on the RPi B+ but additional experimentation is necessary. Note: To be informed of new versions of NTP, join the *NTP Hackers* mailing list at {NTPHack}.

Columns shown in NTP query with the –p option

Remote: Peers specified in the NTP configuration (ntp.conf) file

*	Remote peer or server presently used as the primary reference
#	Source selected, distance exceeds maximum value
o	PPS peer; system synchronization is derived from a pulse-per-second (PPS) signal, either indirectly via the PPS reference clock driver or directly via the kernel interface
+	Source selected, included in final set
x	Source false ticker, out of tolerance
.	Source selected from end of candidate list
–	Source discarded by cluster algorithm
blank	Source discarded high stratum, failed sanity

refid: Remote synchronization source

stratum: Stratum level of the source

t: Types available

l	local (such as a GPS, WWVB)
u	unicast (most common)
m	multicast
b	broadcast
–	netaddr

when: When last polled (s)

poll: Source polling interval (s)

reach: Octal 8-bit left-shift shift register value recording polls (bit set = successful, bit reset = fail)

delay: Round trip delay to the remote peer or server (ms)

offset: Mean offset (phase) in the times reported between this local host and the remote peer or server, positive for 'remote time' ahead of 'local time' (ms)

jitter: Mean deviation (jitter) in the time reported for the remote peer or server, rms of difference between multiple time samples (ms)

9. Leap seconds

Download the most recent leap-second file from one of the following sources and place it on your Desktop. These files have an expiration date so always download the latest one. The filename will be of the form *leap-seconds.3629404800*, where the number string will vary. The web address of the necessary leap-second file varies over time but can be found by internet search ("NTP leap second file"). The following addresses are valid during February 2020:

- ftp://ftp.nist.gov/pub/time/
- ftp://ftp.boulder.nist.gov/pub/time/

Rename the downloaded file to **leap-seconds.file** (that is, replace the number string with the word "file"). Using WinSCP go to the /home/pi/ directory on the RPi, right-click in the right-hand window and select New – Directory.... Name the new directory ntp, as in /home/pi/ntp. Select the /ntp directory so that it is open in the right-hand window (it will be empty). Now, drag leap-seconds.file from the Desktop to the new /ntp directory.

Open a PuTTY shell session (if not already open) and log into the RPi. If desired, PuTTY can be open from WinSCP in Commands menu – Open in PuTTY or Ctrl-P. At the RPi prompt open the ntp.conf file in the nano editor:

```
sudo nano /etc/ntp.conf
```

Add a line that references the leap-second file at the above location:

```
# Location of leap seconds file
leapfile /home/pi/ntp/leap-seconds.file
```

Type CTRL-X (Exit), Y (yes), Enter to save and close the editor.

Restart the NTP service:

```
sudo service ntp restart
```

After a few minutes, query the NTPD

```
ntpq -crv
```

The" leapsec" parameter will be seen near the end of the list and its value should be the date and time of the next leap second as defined in the leap second file. In the screenshot below the date and time values are 1 July 2015 and 0000 UTC, which is the next leap second at the time this guide was written (February 2015). The expiration date of the leap second file also is shown, 28 December 2015 at 0000 UTC. If this guide is used after this expiration date, a newer leap second file will be available with a newer expiration date. The leap second file should be kept updated.

pi@GpsNtp-Pi: ~

```
pi@GpsNtp-Pi ~ $ ntpq -crv
associd=0 status=0115 leap_none, sync_pps, 1 event, clock_sync,
version="ntpd 4.2.8p1@1.3265 Sun Feb  8 00:48:57 UTC 2015 (1)",
processor="armv6l", system="Linux/3.18.6+", leap=00, stratum=1,
precision=-18, rootdelay=0.000, rootdisp=1.210, refid=KPPS,
reftime=d883c8eb.1274fda5  Tue, Feb 10 2015  0:01:47.072,
clock=d883c8fa.04b150ae  Tue, Feb 10 2015  0:02:02.018, peer=25722, tc=4,
mintc=3, offset=0.008819, frequency=-12.997, sys_jitter=0.003815,
clk_jitter=0.010, clk_wander=0.003, tai=35, leapsec=201507010000,
expire=201512280000
pi@GpsNtp-Pi ~ $
```

10. Final Adjustments and Checks

The convergence time constant of the RPi NTP server is about 1 h. Thus, assuming the GPS receiver has a fix, the server achieves close to 0 ms offset of the PPS reference clock in a couple hours. The offset can be followed as it changes by repeatedly using the ntpq commands described above.

When the RPi power is first applied and the GPS receiver obtains a fix, the PPS offset after a few minutes will be some positive or negative value. Over the next couple hours the offset will move closer to 0.000 ms as the NTP daemon learns about the GPS clock characteristics. Many factors are involved so, even after several hours, the offset may not be 0.000 ms and some variation will be noted with each query. My lab test systems typically run 0.000 ± 0.002 ms for the system with the GTop receiver and 0.000 ± 0.005 ms for the U-Blox receiver .

Note: The GPS receivers store the time and satellite ephemeris data in battery-backed RAM. In addition, the NTPD uses a *driftfile* (location specified in the NTP configuration file) to store the clock frequency offset determined by the NTPD. This offset is updated every hour. If the RPi and GPS receiver are power cycled, the stored clock data provides very close to actual date and time and the driftfile provides very close to actual frequency offset. With these two parameters, the convergence is faster. Therefore, on the first cold-start, convergence requires a couple hours after satellite tracking starts but on subsequent starts convergence will be much faster.

The PPS is a precise signal with around 10 ns of jitter. On the other hand, the offset of the GPS serial data output (GPSD) generally will have much more variation than the PPS because of the variables involved in sending data over an asynchronous serial port. However, the GPSD offset can be reduced somewhat by adjusting the GPSD reference clock fudge parameter time1 in the ntp.conf file. The initial value used in the ntp.conf file is 500 ms (0.500 s). Enter the following to see the present offset; repeat several times over a period of a few tens of minutes:

```
ntpq –p
```

Look for the row with remote SHM(0) and refid GPSD. The offset values probably will be different for each query. Do a mental average. For values –100 < offset < +100 ms, there probably is no need for adjustment. If you consistently see larger negative or positive offsets, the NTP statistics can be used to calculate the average. Let the RPi run for at least 4 hours and then at the RPi prompt enter the following series of statements to calculate the average:

```
awk `
/127\.127\.28\.0/ { sum += $5 * 1000; cnt++; }
END { print sum / cnt; }
` < /var/log/ntpstats/peerstats
```

A moment after the last line is entered, a positive or negative value will be returned. The value is the average GPSD reference clock offset in ms and is to be subtracted from the GPSD time1 value in ntp.conf. For example, say the value is –33.573. This means the time1 value for GPSD in ntp.conf needs to be increased by 34 ms from +0.500 to +0.534 s. If the value is 33.573, then time1 needs to be reduced by 34 ms to 0.466. To make the change, use the nano editor:

```
sudo nano /etc/ntp.conf
```

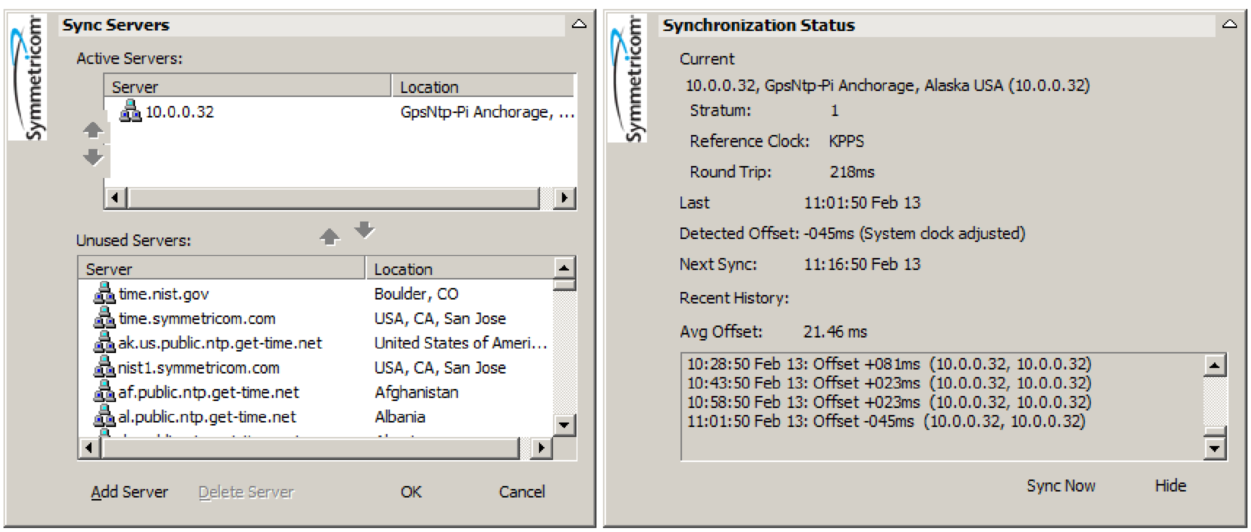
Scroll down to the line starting with *fudge 127.127.28.0* and adjust the time1 parameter. When finished type CTRL-X (Exit), Y (yes), Enter to save and close the editor. Now, restart the NTP service:

```
sudo service ntp restart
```

If you decide to recalculate the average offset using the above procedures, wait at least another day or two. Avoid unnecessarily changing the time1 value. In my lab systems the time1 value in the Adafruit GPSD driver is +0.534 s and in the HAB Supplies GPSD is +0.125 s.

The overall NTP server system can be tested by installing an NTP client program on a PC on the same LAN. Example clients are SymmTime (not compatible with Windows 7x64 or later), Dimension4 and Meinberg NTP (Note: Do not confuse the Meinberg NTP Monitor software tool with the NTP software; both are available on Meinberg's website). Download locations for these and similar programs may be found by ordinary internet search. The preferred client, by far, is NTP itself. The Windows version may be obtained at {NTP-Win}.

After installation, setup the NTP client to use the IP address of the RPi NTP server. For example, the SymmTime setup is shown below for a GpsNtp-Pi lab system with IP address 10.0.0.32. In this case, I clicked Add Server in the Sync Servers window (left), specified the lab system IP address and then removed all other servers from the upper frame with the down-arrow between the frames. The Synchronization Status window (right) shows the status of the new server. Note that the reference clock is KPPS and stratum level is 1, indicating proper operation of the RPi NTP server.



This completes final adjustments and checks of the RPi as a standalone NTP server.

<u>Tweak</u>: The PPS jitter in some systems is reduced by a small tweak to the cmdline.txt file in the /boot/ directory. At the command line prompt enter

```
sudo nano /boot/cmdline.txt
```

At the end of the single line in this file, add `nohz=off` with one space between the new text and existing text. When finished type CTRL-X (Exit), Y (yes), Enter to save and close the editor. Now, reboot the NTP server:

```
sudo reboot
```

11. Provision NTP Server Pool

When the shared memory mode and PPS drivers were setup, the pool statement in the ntp.conf file was commented out with the # character for testing purposes. If the RPi NTP server will not be used with any external LAN or WAN NTP servers whatsoever, the pool statement can be left as-is:

```
# pool us.pool.ntp.org iburst minpoll 6 prefer
```

However, if internet access is available it is recommended to use backup servers in case of, for example, GPS receiver or antenna failure or loss of signal because the antenna is in a poor location. Also, other servers can be used in a peer arrangement to improve overall system performance (peering is beyond the scope of this project but information can be found at {NTPOrg}).

A pool of external servers is accessed by the GpsNtp-Pi NTP server through an internet connection and access to them is enabled by uncommenting the pool statement in the ntp.conf file:

```
pool us.pool.ntp.org iburst minpoll 6 prefer
```

When the pool has been enabled, it is necessary to restart the protocol:

```
sudo service ntp restart
```

Wait several minutes after restart and then query the NTP. If the NTP server pool statement is enabled, you should see additional servers listed (for the US pool, typically 7 or more):

```
ntpq -p
```

```
pi@GpsNtp-Pi: ~
pi@GpsNtp-Pi ~ $ ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*SHM(0)          .GPSD.           0 l    6   16  377    0.000   16.057  22.306
oPPS(0)          .KPPS.           0 l    5   16  377    0.000   -0.049   0.004
 us.pool.ntp.org .POOL.          16 p    -   64    0    0.000    0.000   0.004
+lithium.constan 192.5.41.209     2 u   27   64  377  109.555    0.629 122.105
+tick.nullmodem. 128.10.19.24     2 u   17   64  377  101.200    0.480 227.782
+time.devry.net  10.154.2.2       2 u   22   64  357   98.353    1.775   3.031
+tock6.usshc.com .GPS.            1 u   12   64  377   93.901    2.684  60.808
+soft-sea-01.ser 209.51.161.238   2 u   18   64  377   46.623    1.001   2.339
+origin.towfowi. 204.9.54.119     2 u   13   64  377   67.173    1.605   4.497
+208.53.158.34 ( 130.173.91.58    2 u   20   64  377  101.652    3.515   0.449
pi@GpsNtp-Pi ~ $
```

The character in column 1 indicates the status. In the example above, the pool servers are marked with + character indicating they have been found to be acceptable. Since these are taken from a pool, the list will change over time. See {NTPQ-1} and {NTPQ-2} for a more complete description of the status indicators and parameters.

If you experience problems using external time servers, make sure that UDP port 123 is open in both directions on all firewalls between the GpsNtp-Pi network connection and the remote time servers that are to be used for synchronization.

Note: If you plan to add your RPi NTP server to the pool, you should not use the **`*.pool.ntp.org`** alias in the NTP configuration as described above. Instead, time servers should be hand-picked from good nearby (network-wise) servers. See {NTPJoin} for additional information.

12. References, Web Links and Further Reading

{BkRsRPi}	http://www.reeve.com/Documents/Articles%20Papers/Reeve_RPi_BackupRestore.pdf
{CPi}	http://www.reeve.com/Documents/Articles%20Papers/Reeve_Callisto-Pi.pdf
{GPSHAB}	http://ava.upuaut.net/store/index.php?route=product/product&path=59_60&product_id=117
{GPSHat}	https://blog.adafruit.com/2014/12/26/new-product-adafruit-ultimate-gps-hat-for-raspberry-pi-a-or-b-mini-kit/
{GTOP}	http://www.gtop-tech.com/en/product/LadyBird-1/MT3339_GPS_Module_04.html
{HDDRaw}	http://hddguru.com/
{LWATV}	http://www.reeve.com/Documents/Articles%20Papers/Reeve_RPi-LWATV.pdf
{Netscan}	https://www.softperfect.com/products/networkscanner/
{NMEA}	http://www.gpsinformation.org/dale/nmea.htm
{NTPAcc}	http://www.eecis.udel.edu/~mills/ntp/html/accopt.html
{NTPHack}	http://lists.ntp.org/listinfo/hackers
{NTPJoin}	http://www.ntppool.org/join/configuration.html#management-queries
{NTPLog}	http://www.ntp.org/ntpfaq/NTP-s-trouble.htm
{NTPOrg}	http://www.ntp.org
{NTPPool}	http://www.pool.ntp.org/en/
{NTPRefClk}	http://doc.ntp.org/4.1.2/refclock.htm
{NTPQ}	http://doc.ntp.org/4.1.0/ntpq.htm
{NTP-Win}	https://www.meinbergglobal.com/english/sw/ntp.htm
{PuTTY}	http://www.putty.org/
{Raspbian}	http://www.raspbian.org/
{RvTime}	http://www.reeve.com/Documents/Articles%20Papers/Reeve_LeapSec2015.pdf
{RPiInstall}	http://www.raspberrypi.org/documentation/installation/installing-images/README.md
{uBLOX}	http://www.u-blox.com/en/gps-modules/pvt-modules/max-m8-series-concurrent-gnss-modules.html
{WLAN}	http://www.reeve.com/Documents/Articles%20Papers/Reeve_RPi_WLANSetup.pdf

Document information

Author: Whitham D. Reeve

Revision:
0.0 (Original draft started, 06 Feb 2015)
0.1 (Corrections and updates per working system, 07 Feb 2015)
0.2 (Numerous updates, 12 Feb 2015)
0.3 (Minor edits for clarification, 16 Feb 2015)
0.4 (Revised ntp.conf settings for standalone operation, 24 Feb 2015)
0.5 (Split from system description, 24 Feb 2015)
0.6 (Comprehensive edits, 01 Mar 2015)
0.7 (Additional cleanup for distribution, 15 Mar 2015)
1.0 (Distribution, 1 Apr 2015)
1.1 (Added link to reference clock and enclosure descriptions, 21 Apr 2015)
1.2 (Revised make for ntp-4.2.8p2, 21 Jun 2015)
1.3 (Minor edits sect. 5, 8 and 9, 11 Oct 2015)
1.4 (Edits to clarify, 18 Oct 2015)
1.5 (Minor updates to gpsmon, 02 Nov 2015)
1.6 (Revised note in sect. 8, 30 Jan 2016)
1.7 (Added PuTTY login in sect. 9, 10 Dec 2016)
1.8 (Updated NTP archive URL and update procedures, 30 Mar 2018)
1.9 (Added listing of NTP query peer definitions and NTP link, 19 Nov 2018)
2.0 (Updated addresses for leap second file, 23 Sep 2019)
2.1 (Updated addresses for leap seconds file in sect. 9, 29 Feb 2020)