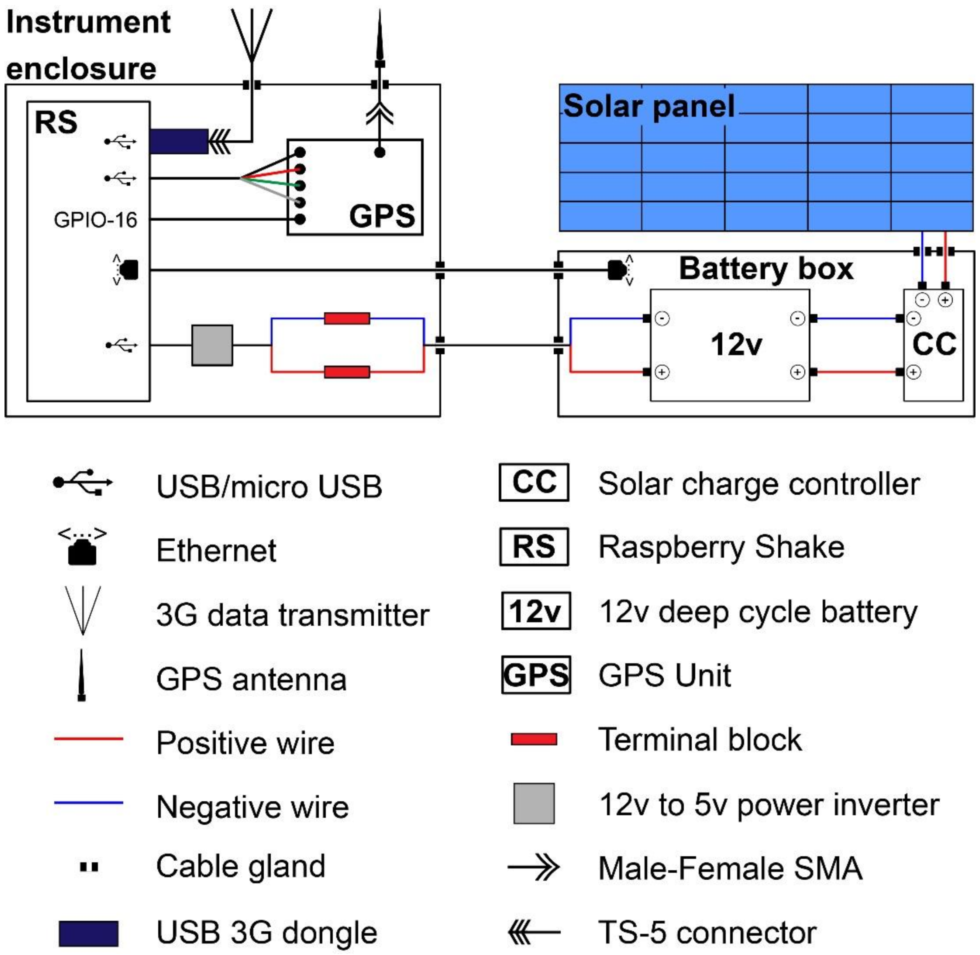
Remote installation of a Raspberry Shake

Instructions were compiled by Rupert Bainbridge from multiple web sources which are listed at the start of each section.

Contents:

1. Wiring diagram for installation with batteries and solar panel



Simplified wiring diagram for the installation of the Raspberry Shake 1D in a remote location with GPS for time synchronisation and 3G data connectivity

Raspberry Shake: https://raspberryshake.org/

Adafruit Ultimate GPS: https://learn.adafruit.com/adafruit-ultimate-gps/overview

3G data transmitter: https://www.amazon.co.uk/Huawei-Unlocked-Portable-Dongle-Genuine/dp/B01DML8GOK/ref=sr_1_3?s=electronics&ie=UTF8&qid=1523878171&sr=1-3&keywords=huwai+dongle

Solar panel & charge controller: https://www.amazon.co.uk/Photonic-Universe-battery-charging-controller/dp/B01N4DKAH4/ref=sr_1_4?srs=1906344031&ie=UTF8&qid=1525170961&sr=8-4&keywords=60w

12v deep cycle battery: http://www.halfords.com/motoring/bulbs-blades-batteries/car-batteries/halfords-3-year-guarantee-hb030-lead-acid-12v-car-battery

Terminal blocks: https://uk.rs-online.com

12v to 5v power inverter: https://www.amazon.co.uk/Universal-Converter-Regulator-Camcorder-HitCar/dp/B00Y8GD00E/ref=cm_sw_em_r_d_dp_daO_WKDKzb1QGQS7K_tt?th=1

2. Power consumption of the Shake

Shake with 3G dongle and GPS attached: 0.5A at 5V

3G dongle: 0.2A

GPS: 0.05A

3. Finding the Shake's dynamic IP address

https://manual.raspberryshake.org/fing.html

Shake must be connected to the router via an Ethernet cable and phone must be connected to the same router's wifi

1. Download the app 'Fing' onto a phone or tablet connected to a router through WiFi

2. Connect your Shake to the same router using an Ethernet connection

3. Open up Fing and select the WiFi network that you are connected to

4. A window will open telling you the IP of all devices currently connected to that network, write the IP address of the Shake down for future use.

4. Changing the password and data storage time

https://manual.raspberryshake.org/hacked.html#hacked
https://manual.raspberryshake.org/traces.html#waveforms-on-disk

This can only be done when connected to the Pi over an internet connection and can access the desktop screen

1. Connect the Shake to the router using an Ethernet cable

2. Open the Raspberry Shake's webpage at ***http://raspberryshake.local/*** or type ***http://*****<IP_Address_here>**

Changing the Password

1. Click on the gear icon and navigate to 'Actions'

2. Change ssh password
   Username: myshake
   Default ssh password: shakeme

Changing the length of time data is stored for

1. Click the settings icon and navigate to Settings/Advanced/Waveform Data Retention/Waveform Files

2. Change the 'Retained for ***x*** days' to the number of days you wish for data to be retained for

   Note: Be careful when configuring this parameter! You risk filling up the disk space. If this happens, you will no longer be able to access your Raspberry Shake. It is estimated that the OS and software consume ~3 Gb of disk space.

---

5. Installing the necessary services on the on the Shake

1. Plug in a screen using the HDMI port and a keyboard using a USB port

2. Power on shake using the mains plug

3. Plug shake into the router using an Ethernet cable and leave it to update for ~10 mins

4. Log into the Shake

   Default Username and Password:
   Username: myshake
   Password: shakeme

   Note: to make the Shakes more secure the password will likely have changed, see the '**Shake_details**' document for login details.

5. Update programme lists on the Shake using the following command:

   ```
   sudo apt-get update
   ```

6. Install gpsd (GPS daemon), gpsd-clients, python-gps, and pps-tools; a suite of tools which will enable the RPi to read the satellite data coming from the GPS receiver.

   ```
   sudo apt-get install gpsd gpsd-clients python-gps pps-tools
   ```

7. Install *sg3-utils* which will enable the use of a 3G dongle to send data in remote locations.

   ```
   sudo apt-get install sg3-utils
   ```

6. Installing the GPS for setting the time

https://area-51.blog/2012/06/18/getting-gps-to-work-on-a-raspberry-pi/
http://raspberrypi.tomasgreno.cz/ntp-client-and-server.html
http://psn.quake.net/shake/
http://www.satsignal.eu/ntp/Raspberry-Pi-NTP.html#no-soldering
http://doc.ntp.org/4.2.8/drivers/driver28.html
http://doc.ntp.org/4.2.8/drivers/driver22.html
https://www.raspberrypi.org/forums/viewtopic.php?t=140585

Here I use an Adafruit Ultimate GPS Breakout v3 (https://learn.adafruit.com/adafruit-ultimate-gps/overview), which requires some soldering of pins for setup and a female-female jumper cable.

Setting up the GPS time

1. Connect a screen (HDMI), keyboard (USB) and the GPS (USB) to the Pi and power up.

2. The GPS time will not synchronise with the NTP time if the date/time on the Shake is considerably different from that being received by the GPS. It is therefore necessary to crudely set the time before starting this procedure. To manually set the time type the following command using your date/time:

```
sudo date --set "11 Oct 2017 16:00:00"
```

3. Run the following command to see all currently connected USB devices; the USB device should be listed. If it is there then the GPS has been recognised as a serial device.

```
sudo lsusb
```

Example output:

```
Bus 002 Device 002: ID 8087:0024 Prolific Technology, Inc. PL2303 Serial
                    Port
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

e.g. UBlox (UBlox) or Adafruit GPS (Prolific Technology) may come up as shown above

4. View the system log to see how the GPS device has been listed:

```
dmesg | grep tty
```

This will search the ***dmesg*** output for any mention of ***tty***. This is used as it is the precursory information to either USB or ACM (see below) and will therefore narrow down the output to the information you require.

Near the bottom of the system log there will be lines confirming that the USB GPS device has been listed. The device should be listed as one of the following: ***ttyUSB0*** or ***ttyACM0***. Note which of these the device is listed as for later.

5. ***You s***hould have already installed the necessary services needed for the Shake to use the GPS. If not then install *gpsd* (GPS daemon), *gpsd-clients, python-gps,* and *pps-tools;* a suite of tools which will enable the RPi to read the satellite data coming from the GPS receiver.

```
$ sudo apt-get install gpsd gpsd-clients python-gps pps-tools
```

6. Start *gpsd*; use ACM or USB depending on how the device was listed in the system log.

```
$ sudo gpsd /dev/ttyUSB0
```

7. Input your device parameters into the gpsd configuration file by entering:

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

In the file there will be the following titles in CAPITALS, the information after the titles needs adding to the file. In the 'DEVICES' section ACM can be changed to USB if this does not work, however ACM should work if this is how the device was listed in the system log (*dmesg*)

IMPORTANT: In newer versions of the Shake image this information may already be in the file, check to make sure it's all there and correct!:

```
START_DAEMON=”true”
USBAUTO=”true”
DEVICES=”/dev/ttyUSB0 /dev/pps0”
GPSD_OPTIONS=”-n -G”
GPSD_SOCKET=”/var/run/gpsd.sock”
```

Save the changes using ctrl+x → y → enter.

8. Restart gpsd

```
$ sudo service gpsd restart
```

9. Check that the GPS has signal before proceeding. A GPS signal lock will be indicated by the flashing light on the Adafruit circuit board:

Flash every second = No lock
Flash every 1 seconds = GPS lock

With your RPi and GPS in a position where it can view the sky, check the data feed. **–s** tells the *cgps* to only show processed. Press '***q***' to exit the '*cgps*' screen

```
$ cgps –s
```

10. The ***ntpd*** (Network Time Protocol daemon) should already be installed on the Shake. You can view NTP information by using the command:

```
$ ntpq -p
```

11. Edit the ***ntp.conf*** file using the following command to access the file.

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

nano refers to the text editing programme installed on the Shake and ***/etc*** is the directory to the file.

IMPORTANT: In newer versions of the Shake Image the GPS clock server/fudge settings may already be in the file, scroll all the way to the bottom to check. Make sure they are all correct, usually there are some server and configuration setting which are incorrect – particularly in the PPS settings.

Enable the collection of statistics on the NTP driver. These stats can be used in future to troubleshoot NTP daemon problems or monitor the function of the NTP daemon. Stats files are logged to **/var/log/ntpstats**

Disable all pool servers except the preferred server option.

Add lines after the existing lines beginning with *server*, to tell NTP where to receive data for timekeeping from the Shared Memory Driver (SHM); this will enable the NTP clock to receive data from the GPS (NMEA) stream. ***127.127.*** indicates that you are using 'virtual' reference clocks. ***28.0*** is the location of the shared memory driver where the GPS reference clock information is available to NTP. ***refid*** is the command to name the stream with the label that you set (GPS or PPS).

Information on the shared memory driver can be found at the following web address (SHM - http://doc.ntp.org/4.2.8/drivers/driver28.html).

The ntp configuration file should look like the example below once finished. You may have to remove some # to enable certain services. **Green**/**red** indicates necessary additions to the file:

Example output:

```
tinker panic 0
# /etc/ntp.conf, configuration for ntpd; see ntp.conf(5) for help

driftfile /var/lib/ntp/ntp.drift

# Enable this if you want statistics to be logged.
statsdir /var/log/ntpstats

statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable
```

```
#pool.ntp.org maps to about 1000 low-stratum servers. Your server will
#pick a different set every time it starts up. Please consider joining
#the pool: <http://www.pool.ntp.org/join.html>
#server 10.18.100.236 iburst
#server 10.18.100.236 iburst
server 1.debian.pool.ntp.org iburst prefer
#server 2.debian.pool.ntp.org iburst
#server 3.debian.pool.ntp.org iburst

# GPS_time – GPS NMEA serial data reference clock
server 127.127.28.0 minpoll 4 prefer
fudge 127.127.28.0 refid GPS stratum 3 time1 0.510

#By default, exchange time with everybody, but don’t allow #configuration.
restrict -4 default kod notrap nomodify nopeer noquery
restrict -6 default kod notrap nomodify nopeer noquery

#Local users may interrogate the ntp server more closely.
restrict 127.0.0.1
restrict ::1
```

12. Restart *ntpd*

```
$ sudo service ntpd restart
```

13. Query the NTP server to see if the time has synchronised – this may take time

```
$ ntpq -p
```

Example output:

```
remote           refid      st t  when poll reach  delay  offset  jitter
=========================================================================
*SHM(0)          .GPS.       3  l   16   16   377   0.000  -10.318 26.352
```

14. If the time has not synced then type the following commands to restart gpsd and ntpd:

```
$ sudo killall gpsd

$ sudo gpsd /dev/ttyUSB0

$ sudo service ntpd restart
```

11. If necessary check the raw data streams from the GPS unit using the following commands:

Check processed GPS data:

```
$ gpsmon /dev/ttyUSB0
```

Press ***q*** and then ***enter*** to quit the *gpsmon* screen back to the command line

7. Setting up the Pulse Per Second (PPS) for better time accuracy

https://hackaday.io/project/15137/instructions
http://www.satsignal.eu/ntp/Raspberry-Pi-NTP.html#no-soldering
https://www.raspberrypi.org/forums/viewtopic.php?t=1970&start=50
https://www.novell.com/support/kb/doc.php?id=7012021

1. Shutdown the Shake and connect the PPS pin on the Adafruit GPS circuit board to the Shake board on GPIO pin 16. See Section 12 at the end of the document for a GPIO pin diagram. You can use an alternative pin to GPIO 16 but you need to change the number used in the boot configuration file later on.

2. Edit the ***ntp.conf*** file: Use the following command to access the file.

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

Enable the NTP clock to receive data from the PPS stream you are about to setup. PPS allows the Shake to keep the timing of the clock's seconds more accurate but does not supply actual time itself. ***22.0*** will use ***/dev/pps0*** to get PPS timestamps and correct the kernel clock.

The ***flag 4*** fudge factor used below records a timestamp once for each second allowing the construction of Allen deviation plots. Add the following lines after the # GPS_time added in the previous section:

```
# PPS_time - kernel-mode PPS ref-clock for the precise seconds
server 127.127.22.0 minpoll 4
fudge 127.127.22.0 refid PPS stratum 0 flag3 1 flag4 1
```

3. Make an ntp working directory

```
$ mkdir ntp
```

4. Enter that directory (*cd* = change directory)

```
$ cd ntp
```

5. Install the following to prevent a file not found error later on in the process

```
$ sudo apt-get install libcap-dev
```

6. Install the following

```
$ sudo apt-get install libssl-dev
```

7. Download the following NTP (network time protocol) tar file, if it does not install then check the ntp site for updates http://archive.ntp.org/ntp4/ntp-4.2 (check bottom of the list for an updated version number e.g. ntp-4.2.8p13):

```
$ wget http://archive.ntp.org/ntp4/ntp-4.2/ntp-4.2.8p13.tar.gz
```

8. Unpack tar file

```
$ tar xvfz ntp-4.2.8p13.tar.gz
```

9. Enter the directory that was just created:

```
$ cd ntp-4.2.8p13
```

10. Once in the directory run '*configure*' which will find the configuration script and build the '*makefile*'.

```
$./configure --enable-linuxcaps
```

11. Build the new NTP software from the 'makefile'

```
$ make
```

(use "**make -j5**" for faster execution on the four-core Raspberry Pi 2/3.)

```
$ sudo make install
```

12. Once you have a new set of NTP binaries, you need to copy them to the correct directory and then restart NTP.

Stop NTP:

```
$ sudo service ntpd stop
```

Copy files:

```
$ sudo cp /usr/local/bin/ntp* /usr/bin/  && sudo cp
  /usr/local/sbin/ntp* /usr/sbin/
```

Restart NTP:

```
$ sudo service ntpd restart
```

13. Move into the/usr/local/sbin and check the new files have been copied

Change directory:

```
$ cd /usr/local/sbin
```

Check listings:

```
$ ls -l
```

Check the listings, files labelled ntp* should have the date that you built the new ntp binaries.

14. Check for version and basic function

```
$ ntpq -crv -pn
```

15. Edit the boot configuration file:

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

Add the following on a new line:

```
dtoverlay=pps-gpio,gpiopin=16
```

Save and close (***Ctrl-x, y, Enter***)

16. Edit the modules file

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

Add the following on a new line

```
$ pps-gpio
```

Save and close (***Ctrl-x, y, Enter***)

17. Reboot

```
$ sudo reboot
```

18. PPS commands from this point on check functionality. To check that the module is loaded, you can use the lsmod command, for example:

```
$ lsmod | grep pps
```

The output should be similar to:

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

19. You should now be able to run the ppstest command and see that pps data is coming through to the Shake. ***You may need to wait upwards of 30 mins if you don't have good GPS signal.***

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

With an example output of:

```
trying PPS source "/dev/pps0"
found PPS source "/dev/pps0"
ok, found 1 source(s), now start fetching data...
source 0 - assert 1351501153.999956346, sequence: 47481 - clear
0.000000000, sequence: 0
source 0 - assert 1351501154.999954601, sequence: 47482 - clear
0.000000000, sequence: 0
```

press ***Ctrl-C*** to exit

20. Check the status of the pps input source in the kernel by using the following command:

```
$ dmesg | grep pps
```

Example output:

```
[ 0.000000] Linux version 3.2.27-pps-g965b922-dirty (root@bt) (gcc version
4.6.2 (Ubuntu/Linaro 4.6.2-14ubuntu2~ppa1) ) #1 PREEMPT Sat Sep 22 16:30:50
EDT 2012
[ 1.866364] usb usb1: Manufacturer: Linux 3.2.27-pps-g965b922-dirty
dwc_otg_hcd
[ 12.797224] pps_core: LinuxPPS API ver. 1 registered
[ 12.803850] pps_core: Software ver. 5.3.6 - Copyright 2005-2007 Rodolfo
Giometti <giometti@linux.it>
[ 12.824858] pps pps0: new PPS source pps-gpio.-1
[ 12.832182] pps pps0: Registered IRQ 194 as PPS source
[ 133.043038] pps_ldisc: PPS line discipline registered
```

21. Check that the GPS and PPS source are now being used to keep time on the Shake

```
$ ntpq -p
```

Example output:

```
remote           refid      st t  when poll reach  delay  offset  jitter
=========================================================================
*SHM(0)          .GPS.      3  l   16   16   377   0.000  -10.318 26.352
oPPS(0)          .PPS.      0  l   14   16   377   0.000   0.113  0.009
```

The * symbol next to SHM indicates that the time is being set from the GPS on the shared memory driver (SHM). The o symbol next to PPS indicates that there is a good PPS signal that is being used to keep the seconds of the SHM synchronised.

22. Statistics on the function of *ntpd* will be collected (enabled in the configuration file in Section 5), sometimes the folder to store the files is not set up.

Change directory to /var/log

```
$ cd /var/log
```

Check to see if the ntpstats folder is there and that the permissions are correct

```
$ ls -l
```

Example output where ‘ntp ntp’ is shown, correct permissions.

```
drwxr-xr-x  2  ntp  ntp     4096  May  14  2018  ntpstats
```

IF the output is different e.g. ‘ntp root’ then follow the next steps. If correct then ignore the following steps and move to the next section.

Create the directory ntpstats (this file may already exist, if so then do not use this command)

```
$ mkdir ntpstats
```

Change the ‘owner’ of the folder from root to ‘ntp’ so that *ntpd* can access the folder and save the statistics files

```
$ sudo chow ntp ntpstats
```

Check the ntpstats directory now has ntp listed as the owner

```
$ ls -l
```

Example output:

```
drwxr-xr-x  2  ntp  ntp     4096  May  14  2018  ntpstats
```

Change directory into ntpstats

```
$ cd ntpstats
```

Check that the *ntpd* statistics files are showing; only statistics activated in the *ntpd* configuration file will be shown.

```
$ ls -l
```

Example output:

```
Total 24
```

```
-rw-r--r--  2  ntp  ntp  2255   May  14  09:14  clockstats
-rw-r--r--  2  ntp  ntp  2255   May  14  09:14  clockstats.20180514
-rw-r--r--  2  ntp  ntp  232    May  14  09:14  loopstats
-rw-r--r--  2  ntp  ntp  232    May  14  09:14  loopstats.20180514
-rw-r--r--  2  ntp  ntp  743    May  14  09:14  peerstats
-rw-r--r--  2  ntp  ntp  743    May  14  09:14  peerstats.20180514
```

Additional useful tool:

Install the GPIO Zero tool to view the GPIO pin layout on screen in the command window:

```
$ sudo apt-get install python3-gpiozero
```

Use the ***`pinout`*** command to view the pin layout on the RPi.

8. Setting up 3G data connectivity using a USB dongle

Adapted from https://manual.raspberryshake.org/gprs.html for the type of dongle used

This setup was achieved using a **Huawei E3531 dongle**.

1. Plug the GPS modem dongle into the USB port on the RPi

2. SSH into the Shake

3. Determine what USB devices are connected and confirm you see the GPRS modem

```
$ lsusb
```

Example output:

```
Bus 001 Device 007: ID 12d1:1f01 Huawei Technologies Co., Ltd.
Bus 002 Device 002: ID 8087:0024 Prolific Technology, Inc. PL2303 Serial
                    Port
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

4. Install '*sg3-utils*'

```
$ sudo apt-get install sg3-utils
```

5. Restart the Shake

```
$ sudo reboot
```

6. SSH back into the Shake and re-determine GPRS modem still appears

```
$ lsusb
```

7. Use this command to adjust the configuration and tell the system to bring up the Huawei GPRS as eth1

```
$ sudo /usr/bin/sg_raw /dev/sg0 11 06 20 00 00 00 00 00 01 00
```

Example output:

```
SCSI Status: Good

sense Information:
sense buffer empty
```

8. Confirm device appears as eth1

```
$ sudo ifconfig -a
```

Example output:

```
eth0      Link encap:Ethernet  HWaddr b8:27:eb:44:65:26
  inet addr:10.18.101.108  Bcast:10.18.101.255  Mask:255.255.255.0
  inet6 addr: fe80::baa1:df69:36b5:ecb3/64 Scope:Link
  UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
  RX packets:126048 errors:0 dropped:1485 overruns:0 frame:0
  TX packets:46 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:6134619 (5.8 MiB)  TX bytes:6330 (6.1 KiB)

eth1      Link encap:Ethernet  HWaddr 58:2c:80:13:92:63
  inet addr:192.168.8.100  Bcast:192.168.1.255  Mask:255.255.255.0
  inet6 addr: fe80::3d2:b0ae:25df:f0c5/64 Scope:Link
  UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
  RX packets:1608 errors:0 dropped:0 overruns:0 frame:0
  TX packets:2091 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:230145 (224.7 KiB)  TX bytes:398195 (388.8 KiB)
```

eth1 now exists and IP 192.168.8.x is visible (This may vary from unit to unit)

9. Make this change permanent so that the GPRS modem comes up automatically at boot

```
$ sudo nano /etc/udev/rules.d/10-Huawei.rules
```

Add the following **on one line** to the new document:

```
SUBSYSTEMS=="usb", ATTRS{modalias}=="usb:v12D1p1F01*",
SYMLINK+="hwcdrom", RUN+="/usr/bin/sg_raw /dev/hwcdrom 11 06 20 00 00
00 00 00 01 00"
```

10. Adjust interfaces

```
$ sudo nano /etc/network/interfaces
```

At end, add:

```
allow-hotplug eth1
iface eth1 inet dhcp
```

11. Reboot

```
$ sudo reboot
```

9. Retrieving data remotely

https://iamkelv.in/blog/2017/06/zerotier.html
https://www.zerotier.com/download.shtml

To remotely access the Shake over the Internet you will need to set up Zerotier on both a work computer and the Shake.

On a Windows PC open a browser (e.g. Chrome):

1. Go to https://my.zerotier.com/ and create an account

2. Once logged in, go to the Networks tab and create a new network

3. Change the name of your new network to something relevant

4. Under *IPv4 Auto-Assign*, check *Auto-Assign from Range* and click on an IP range which you know will not conflict with the private IP ranges of any networks you use your devices on (ZeroTier will automatically assign your devices an IP from this range).

5. Under Access Control, ensure Certificate (Private Network) is checked

6. Take note of the Network ID that is produced (top of the page)

On the Raspberry Shake:

7. Run the shell command under the Zerotier download page. (you may have to run this command twice if it does not run first time. Simply press the up arrow to retrieve our last command on the command line and press Enter). This link may periodically change, if unsure check the download page and look for the Linux GPG download instructions.

```
curl -s
'https://raw.githubusercontent.com/zerotier/ZeroTierOne/master/doc/co
ntact%40zerotier.com.gpg' | gpg --import && \
if z=$(curl -s 'https://install.zerotier.com/' | gpg); then echo "$z"
| sudo bash; fi
```

8. Ensure Zerotier starts on system boot

```
$ sudo systemctl enable zerotier-one
```

9. Check if everything is working by running

```
$ sudo zerotier-cli status
```

Example output:

```
200 info [ID] [version] ONLINE
```

10. To join the network, run

```
sudo zerotier-cli join [Network ID]
```

The [Network ID] section should be replaced with your network ID number that was created on the Zerotier website.

On a Windows PC:

11. Open the networks tab on Zerotier, select your network and scroll to the bottom of the page. In the ‘*Members*’ section authorise the connection of the Shake to the network by clicking the tick box in the ‘*Auth?*’ column.

12. Add a suitable Name of the device, i.e. Shake [LOCATION], in the ‘*Name/Description*’ column.

On the Raspberry Shake:

13. Check that your Raspberry Pi is properly connected by running

```
$ sudo zerotier-cli listnetworks
```

Example output:

```
200 listnetworks [...] OK PRIVATE [...] [The IP(s) you assigned the
Raspberry Pi]/24
```

14. To have your Raspberry Pi automatically join the virtual network on system boot, simply run

```
$ sudo touch /var/lib/zerotier-one/networks.d/[Network ID].conf
```

On a Windows PC:

15. Navigate to the Account tab of your Zerotier account online

16. On ‘*API Access Tokens*’, click ‘*Create Access Token*’

17. Have this access token to hand and download the Zerotier One client for Windows from https://www.zerotier.com/download.shtml

    Once installed add a new network by right-clicking on the icon and selecting ‘*Join network*’

18. Enter your Network ID and click ‘Join’

19. Enter your newly created API key when requested

20. Install Filezilla from https://filezilla-project.org/

21. To remotely access the Shake, open Filezilla. In the QuickConnect toolbar use the following inputs:

Host: Managed IP of the Shake as noted on the Zerotier Network tab
Username: Username of the Shake
Password: Password of the Shake
Port: 22

22. Alternatively, use the Windows '*Remote Desktop Connection*' tool to SSH into the Shake and access the command line.

    When prompted for the 'Computer', use the Managed IP of the Shake as noted on the Zerotier Network tab

10. Setting up a Static IP address for the Shake – works sometimes…

http://www.circuitbasics.com/how-to-connect-to-a-raspberry-pi-directly-with-an-ethernet-cable/

Once setup the static IP will not allow further services to be installed via an Ethernet connection to a router, so make sure all necessary services (*xrdp*, *gpsd* etc.) have been installed before this process is undertaken. The Static IP that will be created is specific to the laptop from which it is set up, therefore this will need to be considered when taking a laptop into the field to download data.

1. Open Windows Command Prompt line by searching 'command prompt' in the windows search option.

2. Find the IPv4 IP address of the laptop's Ethernet adapter in Command Prompt:

```
$ ipconfig
```

3. Under the ***Ethernet adapter Ethernet*** configuration settings find the ***Autoconfiguration IPv4 Address***; it will look something like: 169.254.81.78

4. Power on the Shake with a screen and keyboard attached. Find the default gateway IP of the Shake using the following command:

```
$ route -ne
```

Write down the number from the ***Gateway*** column on the line which has ***eth0*** in the ***Iface*** column. It may look something like 10.0.0.1 or 0.0.0.0

5. Configure the Static IP in the ***dhcpcd.conf*** file. Open the file using the command:

```
$ sudo nano /etc/dhcpcd.conf
```

6. Scroll to the bottom of the file and add the following lines of code to the bottom of the file.

```
interface eth0
static ip_address=169.254.81.99

static routers=0.0.0.0
```

For the ***static ip_address*** use the first three numbers from the laptop's IPv4 address and select a number between 0-255 for the final number.

Scotland laptop: 169.254.218.XXX RB Laptop:

For the ***static routers*** use the ***Gateway*** number noted down earlier. Exit and save the file using ***ctrl + X*** followed by ***ctrl + Y***

7. Reboot the Shake

```
$ sudo reboot
```

After the reboot, connect your laptop to the Shake using an Ethernet cable. Open Remote Desktop Connection on the laptop and type in the static IP address that was set. Remote desktop should then prompt you for the Username and Password for the Shake and allow you to login to the Shake command line.

11. Downloading data from shake offline via Filezilla

https://filezilla-project.org/

1. Ensure Filezilla is installed on the laptop you want to use to download the data. This will be the laptop which the Static IP has been setup on.

2. Open Filezilla and enter the following information into the toolbar at the top of the screen

   Host: 169.254.XX.XX (this is your IP – Static if connected over Ethernet or Zerotier if connected over internet)
   Username: myshake
   Password: shakeme
   Port: 22

3. Once remote access is acquired through FileZilla you will need to navigate to the appropriate files on the Shake using the pane on the right hand side of the screen. The file directory for the waveform data is as follows:

   ```
   /opt/data/archive/YEAR/NETWORK/STATION/CHANNEL/<DAILY MINISEED FILES>
   ```

   In reality this may look something like:

   ```
   /opt/data/archive/2017/AM/RE**D/SHZ.D/AM.RE**D.00.SHZ.D.2017.311
   ```

   The section reading ***AM.RE**D.00.SHZ.D.2017.311*** is the daily miniseed file. The number at the end (311 in this case) refers to the sequential Day of the Year (DOY) e.g. 311 would be November 7th.

   FileZilla defaults to accessing ***/home/myshake*** folder on the Shake, you will need to go up two directories (the directory above home) in order to find the ***/opt*** directory and locate the data using the directory structure above.

4. Select all of the miniseed files in the folder and drag them across to the desired folder on the laptop. The directory structure for the laptop is displayed on the left hand side of the screen.

5. Once the transfer is complete you can close FileZilla and stop the remote connection.

12. Remote login to the Shake

Note: requires a laptop and Ethernet connection to the laptop

Use this function if you want to remotely login to the Shake using a laptop

1. Connect an Ethernet cable between the laptop and Shake

2. Open the Windows Remote Access software on the laptop and enter the Static IP address which has been set up.

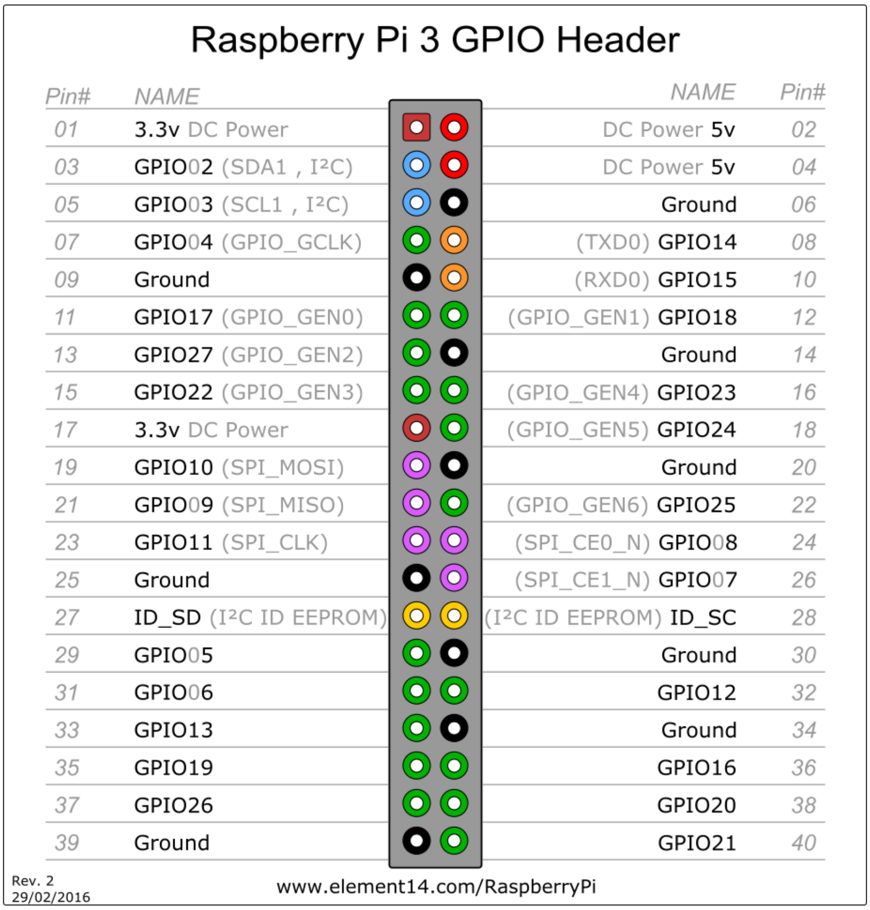
3. This should open up the login screen for the Shake, type in the Username and Password

4. The Shake will login and you will have the familiar command line view to enter commands.

___

13. Raspberry Shake GPIO pins

https://manual.raspberryshake.org/specifications.html



Raspberry Pi 3 GPIO Header

Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I²C)	DC Power 5v	04
05	GPIO03 (SCL1 , I²C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I²C ID EEPROM)	(I²C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

Rev. 2
29/02/2016

www.element14.com/RaspberryPi

The Raspberry Shake board ("hat") sits on pins 01 through 26. So GPIOs and grounds on pins 27 to 40 are easily available to end users for other applications. The Raspberry Shake does not, however, use all of the pins from 01 to 26. The following pins are used:

02, 04: 5v
06, 09, 14, 20, 25: Ground
08, 10: Tx/Rx
15: RESET
17: 3.3v

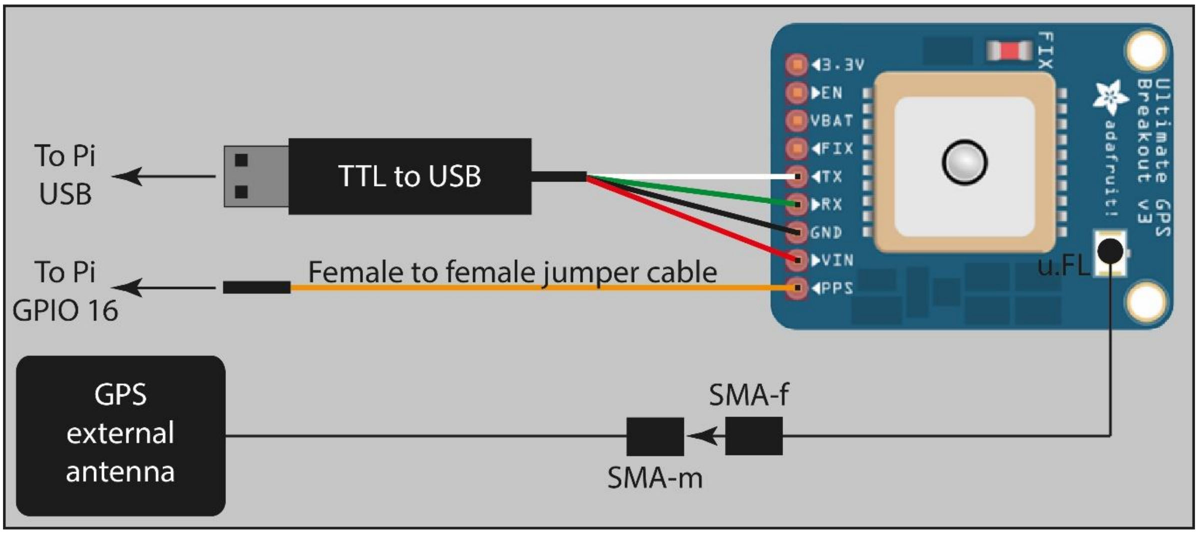
All other pins are available for use. Some have specific functions such as pins 27 and 28, but the others can all be used for general purpose or their specific functions.

---

14. Preparation of the Adafruit Ultimate GPS

https://learn.adafruit.com/adafruit-ultimate-gps-on-the-raspberry-pi/setting-everything-up

1. Cut the row of pins down to a set of 9 and insert them into the 9 pin holes on the GPS board.
2. Use a soldering iron and solder to affix the pins to the board
3. If wanted, solder the battery holder to the reverse of the GPS board
4. Attach the coloured USB-to-TTL cables and a female-to-female jumper cable in the following manner:



5. The USB cable should be put into one of the Shake’s USB slots, whilst the jumper cable should be attached to GPIO pin 16.
6. Connect the male-SMA connector of the external aerial to the female-SMA connector of the SMA-to-u.FL cable.
7. Connect the u.FL connector to the mount on the GPS board

15. Constructing Allen deviation plots for PPS

http://www.reeve.com/Documents/Articles%20Papers/Reeve_GpsNtp-Pi_Setup.pdf
http://doc.ntp.org/4.2.4/monopt.html

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 files may be accessed from a Windows computer using Filezilla to copy the files over to a PC.

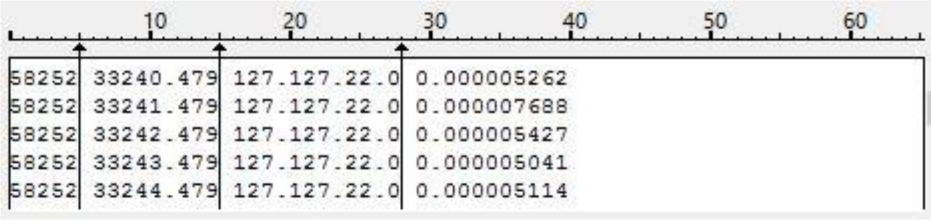
The statistics files can be opened using Excel

1. Open the clockstats file in Excel – all the data will be in one column

2. Highlight the column with the data and split it into the separate columns of information;

   Data → Text to Columns → Fixed width → Next

   Example output:



```
58252 33240.479 127.127.22.0 0.000005262
58252 33241.479 127.127.22.0 0.000007688
58252 33242.479 127.127.22.0 0.000005427
58252 33243.479 127.127.22.0 0.000005041
58252 33244.479 127.127.22.0 0.000005114
```

   Columns should already be split by the spaces in the data column, so click Finish.

3. Columns will now be ordered in the following manner:

   MJD – Modified Julian Day

   Time – Units in seconds (s) after midnight of that day in UTC

   Server – The server providing the data

   Offset -

MJD	Time (s)	Server	Offset
58252	33240.479	127.127.22.0	0.000005262

4. Allen deviation plots can now be made of the Time and Offset values

16. Useful commands

mkdir – make a new directory in the present working directory

```
$ mkdir NewDIR
```

rmdir – remove directory if it is empty

```
$ rmdir NewDIR
```

rm – removes the specified file (or recursively from a directory when used with –r). Files deleted like this are gone for good!

cp – makes a copy of a file and places it in the specified location (copy-paste)

```
$ cp ~/fileA /home/NewDir
```

mv – moves a file and places it in a specified location (cut-paste)

```
$ mv ~/fileA /home/NewDir
```

cat – lists the contents of a file(s)

```
$ cat FileA
```

can also be used on multiple files; e.g. to list the contents of all .txt files in the directory

```
$ cat *.txt.
```

head – lists the top of a specified file (default is 10 lines). Can be used with –n to specify the number of lines. E.g. to show the top 15 lines of FileA

```
$ head -15 FileA
```

tail – same as head but for the bottom of the file

sudo – run commands as a super user

df – displays the disk space available. Use –h to outputs the diskspace in kb, mb and gb

```
$ df –h
```

wget – download a file directly to the computer, e.g.

```
$ wget https://www.raspberrypi.org/documentation/linux/usage/commands.md
```

`sudo reboot` – restart the RPi

`sudo shutdown` – shutdown the RPi

`set date and time`

```
$ sudo date –set “11 Oct 2017 16:00:00”
```

Time should be set to UTC

`ifconfig` – configure network interface parameters

Keyboard functions

Shift + PageUp/PageDown - move the page up or down page by page

Shift + Ctrl + PageUp/PageDown – move the page up or down line by line

Ctrl + X – Exit

Ctrl + O – Save

Navigating the directories

Working directory - the directory which you are currently in:

`ls` – Lists the contents of the current working directory

https://www.tecmint.com/15-basic-ls-command-examples-in-linux/

`cd` – change current directory to the one specified

relative path = `cd directory`;
absolute path = `cd /home/pi/directory`)

`pwd` – display the name of the present working directory/give you the path to the working directory

`cd ../` - go up a folder in the directory tree

Search files

`grep` – searches specified files for specified arguments, e.g. to search for lines with the word *apple* in a file called *fruitlist.txt* you would execute the command:

```
$ grep “apple” fruitlist.txt
```

To search for *apple* in all text files in the working directory you can use a wildcard (*):

```
      $ grep “apple” *.txt
```

OR for example, to search for anything associated with pps information in the dmesg (dmesg - gives information about the status of the kernel)

```
      $ dmesg | grep pps
```

`Find` **–** looks for a file.

```
      $ Find / <file or folder name>
```

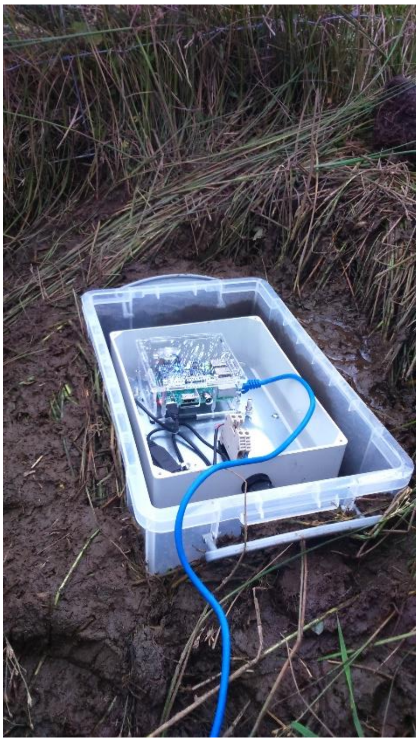
17. Directory functions

https://docs.gempa.de/seiscomp3/current/base/installation.html

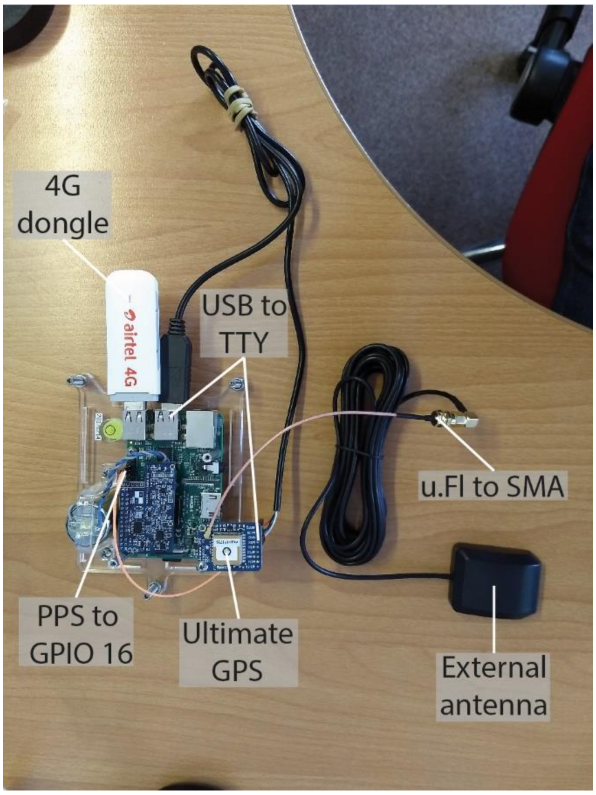
Directory	Description
bin	The user module binaries.
lib	The base library directory used by all modules.
man	The manual pages.
sbin	The system/service/server binaries such as seedlink.
var	Variable files whose content is expected to continually change.
var/log	Log files of started modules.
var/lib	Default directory for files created by modules
etc	Configuration directory.
etc/defaults	The default configuration files. This directory is read as first when a module starts.

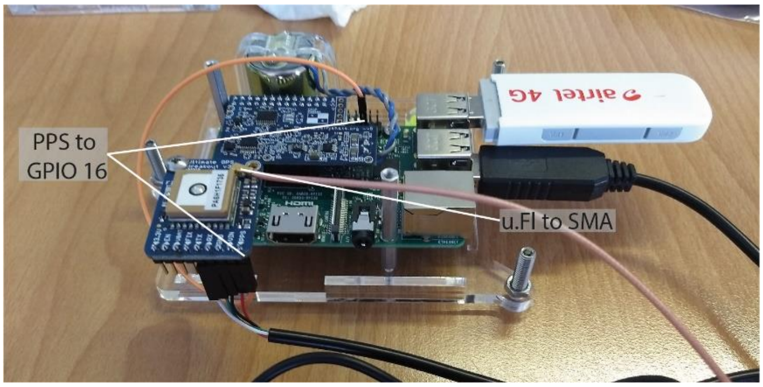
18. Pictures of the Shake setup

Version 1 - Early installation without permanent GPS or 3G connectivity. Manual connection to shake through Ethernet (Filezilla) to remove daily miniseed files for processing.



Version 2 – Not currently installed in the field – works in laboratory testing




PPS to GPIO 16
u.FI to SMA
airtel 4G

POWER SAVING – From Raspberry Shake manual - https://manual.raspberryshake.org/power.html?highlight=power%20save

Disable HDMI: Est. savings: ~20 mA

First, install this library:

```
$ sudo apt install libraspberrypi-bin
```

Enter the rc.local file

```
$ sudo nano /etc/rc.local
```

Add the following line near the end of file, **BEFORE “exit 0”**

```
/opt/vc/bin/tvservice -o
```

Shut off PWR and ACT LEDs: Est. savings: ~5 mA per LED

Add this at end of /boot/config.txt (only worked for us for shutting off green ACT LED):

```
# Disable the ACT LED.
dtparam=act_led_trigger=none
dtparam=act_led_activelow=off
```

Shut off LAN LEDs:

install libusb-dev on the Pi

```
$ sudo apt-get install libusb-dev
```

Download llctl program to a Windows PC: https://www.raspberrypi.org/forums/viewtopic.php?f=63&t=72070

Using Filezilla log in to the Shake and copy the downloaded file to /opt

Unpack the llctl.tgz file and move into the newly created folder using the following command:

```
$ mkdir llctl; tar -xvf llctl.tgz --directory llctl; cd llctl
```

When in the /opt/llctl folder, install the llctl program using the following command

```
$ make
```

Copy the new llctl programme in to /usr/local/bin

```
$ cp llctl /usr/local/bin/
```

Edit the rc.local file

```
$ sudo nano /etc/rc.local
```

Add the following to the file before the ‘exit 0’ line

```
/usr/local/bin/llctl f0 l0 d0
```

Reboot the Pi

```
$ sudo reboot
```