

AuroraWatchNet magnetometer manual

Steve Marple,
Lancaster University.

Monday 11th November, 2013

Licence

This document is made available under the Creative Commons Attribution-ShareAlike 3.0 Unported Licence.



Please attribute this work as “AuroraWatchNet magnetometer manual. Steve Marple, Lancaster University. 2013.”

Contents

Licence	i
Contents	ii
List of figures	v
Abbreviations	vi
I Introduction	1
1 Overview of the hardware	2
1.1 Introduction	2
1.2 Sensor unit	2
1.3 Base unit	2
II Construction	5
2 Beginning construction	6
2.1 Anti-static precautions	6
2.2 Tools required	6
2.3 Order of assembly	6
3 FLC100 shield assembly	8
3.1 Introduction	8

CONTENTS	iii
3.2 FLC100 shield version 1.0	8
3.2.1 Order of assembly	8
4 Calunium assembly	11
4.1 Introduction	11
4.2 Calunium version 2.0 and version 2.1	11
4.2.1 Order of assembly	11
4.3 Testing the board	17
4.4 Programming the firmware	17
4.4.1 Programming the bootloader	17
4.4.2 Programming the magnetometer firmware	17
5 Sensor PCB assembly	19
5.1 Sensor PCB version 1.2	19
5.1.1 Order of assembly	19
III Installation	24
6 Site requirements	25
6.1 Sensor requirements	25
6.2 Network requirements	25
7 Installation procedure	26
7.1 Determining the maximum range for the radio link	26
7.1.1 Factors which alter the maximum range	27
8 Raspberry Pi setup	28
8.1 SD card creation	28
8.2 Configuring Raspbian	28
8.2.1 /dev/ttyAMA0 serial port setup	28

8.2.2	Raspbian configuration	29
8.2.2.1	Change user password	29
8.2.2.2	Internationalisation options	29
8.2.2.3	Advanced options	29
8.2.2.4	Expand Filesystem	29
8.2.3	Install missing software packages	29
8.2.4	Remove swap file	30
8.2.5	Configure file system mount options	30
8.3	Installing the AuroraWatchNet server software	30
8.3.1	Install the Git repository	30
8.3.2	Configure python	31
8.3.3	Configure cron	31
8.3.4	Configure ifplugd	32
8.3.5	Create configuration file	32
8.3.6	Create init file for server daemon	32
8.3.7	Configure ntp	32
IV	Operation	34
9	Raspberry Pi operation	35
9.1	Introduction	35
9.2	Shutting down the Raspberry Pi	35
9.3	Starting and stopping the data recording daemon	35
9.4	Monitoring the data recording process	36
10	Software and firmware updates	37
10.1	Raspbian updates	37
10.2	AuroraWatchNet software updates	37
10.3	Sensor unit firmware updates	37

List of figures

1.1	System overview	3
1.2	Sensor unit	3
1.3	Base unit	4
3.1	Completed FLC100 shield	9
3.2	FLC100 shield v. 1.0 circuit diagram.	10
4.1	Completed Calunium v2.1	12
4.2	LED orientation	14
4.3	Real-time clock load capacitors (Calunium v 2.0)	14
4.4	Calunium v. 2.0 circuit diagram.	15
4.5	Calunium v. 2.1 circuit diagram.	16
5.1	Completed sensor PCB (single-axis)	20
5.2	Completed sensor PCB (three-axis)	21
5.3	Fit the male turned-pin headers.	21
5.4	Fit the female turned-pin sockets.	22
5.5	Place the sensor PCB onto the female turned-pin sockets.	22
5.6	Sensor PCB version 1.2 circuit diagram.	23

Abbreviations

ADC	analogue to digital converter
DHCP	dynamic host configuration protocol
DNS	domain name system
ESD	electro-static discharge
FET	field-effect transistor
GPU	graphics processing unit
HTTP	Hypertext transfer protocol
HTTPS	Hypertext transfer protocol secure
IP	internet protocol
ISM	Industrial, scientific and medical (radio band)
ISP	in-circuit serial programmer (sometimes abbreviated as ICSP)
LED	light emitting diode
NTP	network time protocol
PCB	printed circuit board
RFI	radio-frequency interference
RTC	real-time clock
SD	secure digital
SSH	secure shell
TTL	transistor-transistor logic
USB	universal serial bus
UTC	coordinated universal time

Part I

Introduction

Chapter 1

Overview of the hardware

1.1 Introduction

The magnetometer is designed for low-power operation, simple installation and ease of construction. The entire design is open source, allowing anyone with reasonable soldering ability to construct one.

The magnetometer has two major parts, the base unit and the sensor unit (Figure 1.1). The sensor unit is located outdoors, away from buildings, cars and other sources of human disturbance. It is battery powered and communicates with the base unit by a radio link (433 MHz or 868 MHz), enabling the sensor to be installed without any wiring to the base unit. The base unit is placed indoors and should be positioned such that there are the minimum number of walls between it and the sensor unit.

1.2 Sensor unit

The sensor unit is contained inside a waterproof enclosure approximately 1.1 m high which is partially buried to reduce temperature variations and to provide a stable foundation. The sensor itself is placed at the bottom of the enclosure, approximately 0.85 m below ground. The microcontroller, radio module and battery are positioned in the top part of the enclosure, above ground level. Insulating material (e.g. rockwool) is used to fill the space in-between.

The Calunium microcontroller board is based on the popular Arduino platform but uses the more powerful Atmel ATmega1284P microcontroller.

1.3 Base unit

The base unit is a Raspberry Pi single-board computer with a radio transceiver unit. The Ethernet interface of the Raspberry Pi is used to send the magnetic field measurements to Auro-

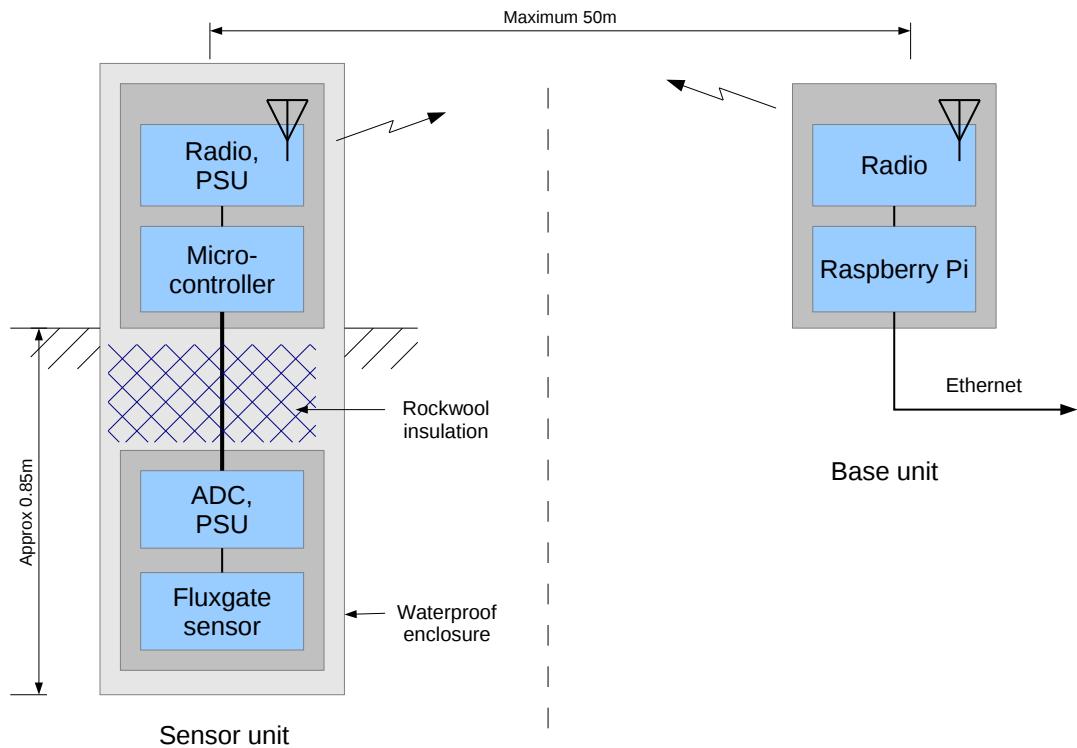


Figure 1.1: System overview.



Figure 1.2: Sensor unit. ©Steve Marple. CC BY-SA 3.0.



Figure 1.3: Base unit. ©Steve Marple. CC BY-SA 3.0.

raWatch UK. When the Raspberry Pi is accessed over the network with Secure Shell (SSH) a display and keyboard are not needed. The Raspberry Pi runs the Raspbian linux distribution. The receiving software is written in Python.

Part II

Construction

Chapter 2

Beginning construction

2.1 Anti-static precautions

2.2 Tools required

- Soldering iron.
- Side cutters.
- Small pliers.
- In-circuit serial programmer (ISP) for Atmel AVR microcontrollers. Instructions are given for the Atmel AVR Dragon but other programmers can be used.
- USB to TTL serial converter for 3.3 V operation, e.g., FTDI TTL-232R-3V3.
- Digital multimeter.
- Solderless breadboard (optional).

2.3 Order of assembly

For ease of access components should normally be fitted in order of increasing size, particularly increasing height. If this order is not observed it can be very difficult to access the pads of surface mount devices. It is also preferable that *passive* components (resistors, capacitors, inductors and crystals) are fitted before semiconductors (field-effect transistors, integrated circuits). This is because the semiconductors are easily damaged by electro-static discharge (sometimes this damage isn't immediately obvious). It is therefore more convenient to fit as many components as possible before fitting the semiconductors, at which point ESD precautions should be followed. As field-effect transistors are particularly vulnerable to damage by ESD it is recommended they are fitted as late as possible. From these guidelines the following order is recommended.

- Surface-mount passive components.
- Surface-mount semiconductors.
- Through-hole passive components.
- Through-hole semiconductors (FETs last).
- Switches.
- Connectors, battery holders.

The first PCB to be assembled is the FLC100 shield, this board provides the power to the system and will enable you to test each part correctly.

Chapter 3

FLC100 shield assembly

3.1 Introduction

The FLC100 shield is an Arduino “shield” which operates at 3.3 V. Do not attempt to use it with standard Arduino boards which are operated at 5 V. The shield houses the XRF radio module, the boost power supply (which creates the 3.3 V supply for the microcontroller and radio) and the 3.3 V – 5 V level shifters.

There are both through-hole and surface mount versions of the boost power supply. It is suspected that the through-hole version causes RFI since the radio module often fails to receive messages. This problem was not apparent on the prototype board. The surface-mount version has no such problems and is the option which should be used. It is also cheaper and more efficient.

There is an option to fit a FLC100 sensor directly to the circuit board. This option is not used because the FLC100 is slightly temperature sensitive and better performance is obtained by positioning the sensor below ground. Whilst the board provides an option to fit an MCP3424 ADC and MAX619 charge-pump power supply they are also fitted remotely, below ground, for reasons of temperature stability.

3.2 FLC100 shield version 1.0

3.2.1 Order of assembly

1. IC4 (MCP1640).
2. R12 (510 k Ω).
3. R11 (15 k Ω).
4. R10 (910 k Ω).
5. C15 (4.7 μ F).
6. C16 (10 μ F).
7. R1, R3, R4, R6, R8 (10 k Ω).

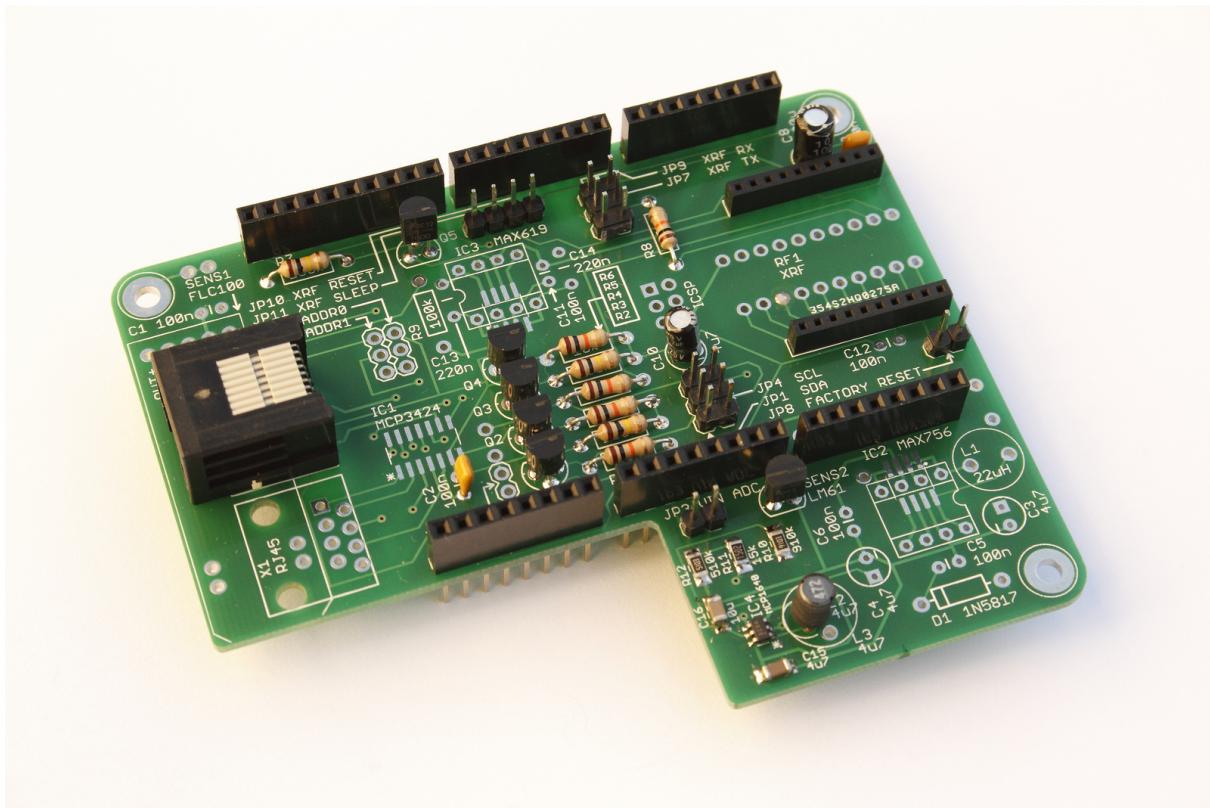


Figure 3.1: Completed FCL100 shield. ©Steve Marple. CC BY-SA 3.0.

8. R2, R5, R7 (100 kΩ).
9. C2, C7, C9 (100 nF). **TO DO: check list**
10. C10 (4.7 μF).
11. C8 (100 μF).
12. L2 (4.7 μH). The shorter lead should be fitted **TO DO:**. Although the orientation of inductors is normally ignored communication with the manufacturer revealed that the shorter lead indicates the start of the winding. This arrangement is preferred to help minimise RFI.
- 13.
- 14.
- 15.
- 16.
- 17.
18. Q1, Q2, Q3, Q4, Q5 (2N7000).

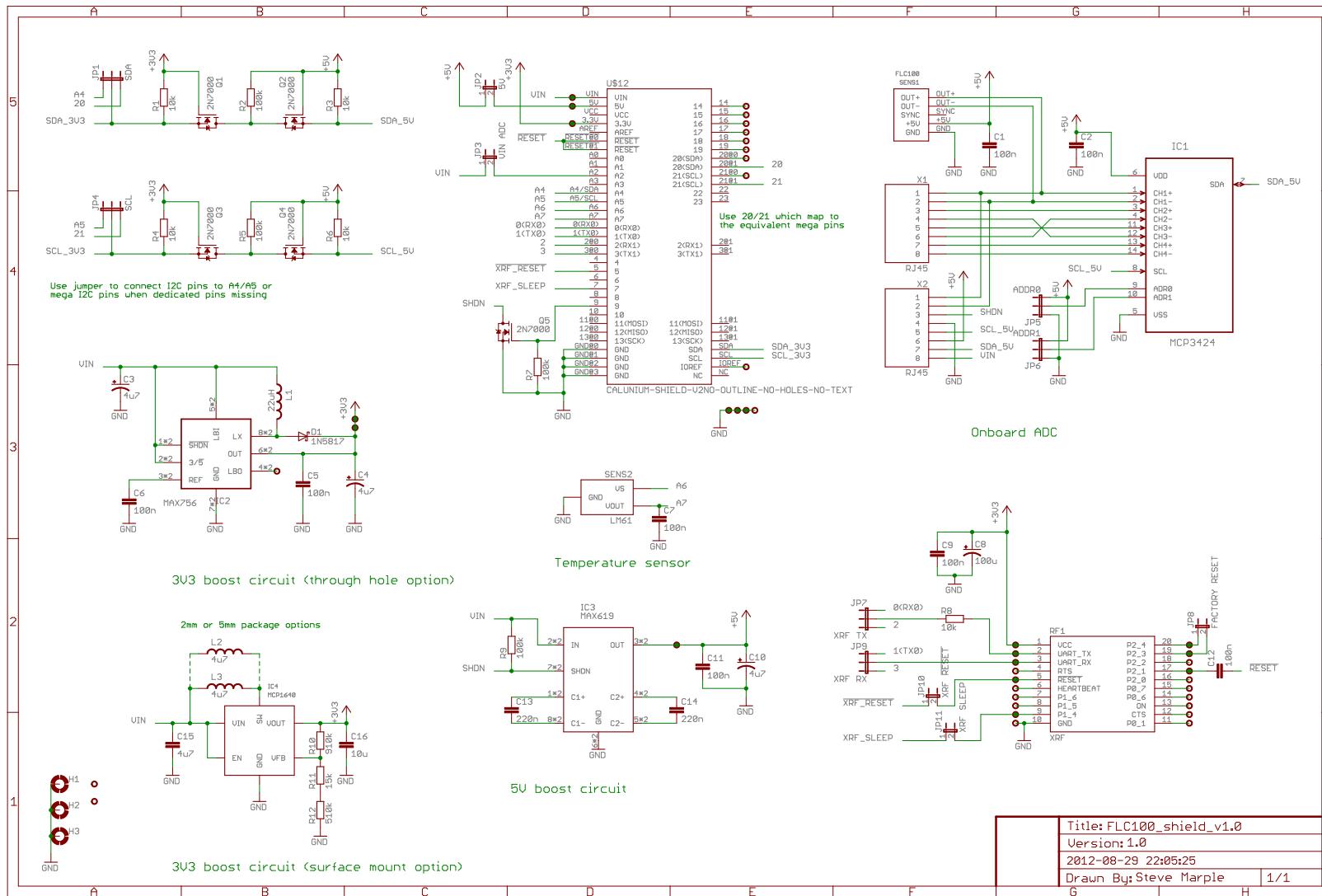


Figure 3.2: FLC100 shield v. 1.0 circuit diagram.

Chapter 4

Calunium assembly

4.1 Introduction

The Calunium microcontroller development board is intended to be a flexible system for both development and embedded use. As such it has various hardware options and careful attention must be paid to assembling it for optimum performance. Parts which are not needed are omitted to lower power consumption (e.g., power LED, USB controller).

4.2 Calunium version 2.0 and version 2.1

4.2.1 Order of assembly

Fit components in order:

1. IC2. The standard real-time clock is the MicroChip MCP90410 but MicroChip MCP79411 or MCP79412 can be used without any other changes. It is also possible to fit the Maxim DS1338-33 real-time clock, but see below for changes.
2. Y1 (32.768 kHz).
3. R1, fit a $680\ \Omega$ resistor. Ignore the $1\ k\Omega$ marking; a lower value resistor is used to enable the green LED to be seen more clearly in daylight.
4. R7 ($1\ k\Omega$). Do not fit if using the DS1338-33 real-time clock. Instead link between R7 and D2 at the end nearest the RTC battery, as indicated by the white line on the silkscreen. The wire will bypass both R7 and D2 which are not required for the DS1338-33.
5. R4, R5 ($4.7\ k\Omega$).
6. R3, R6 ($10\ k\Omega$).
7. L1 ($10\ \mu\text{H}$).
8. 40 pin socket for IC4.
9. C4 ($100\ \text{pF}$).
10. C3, C5, C7, C9, C12 ($100\ \text{nF}$).
11. C2, C8 ($1\ \mu\text{F}$).

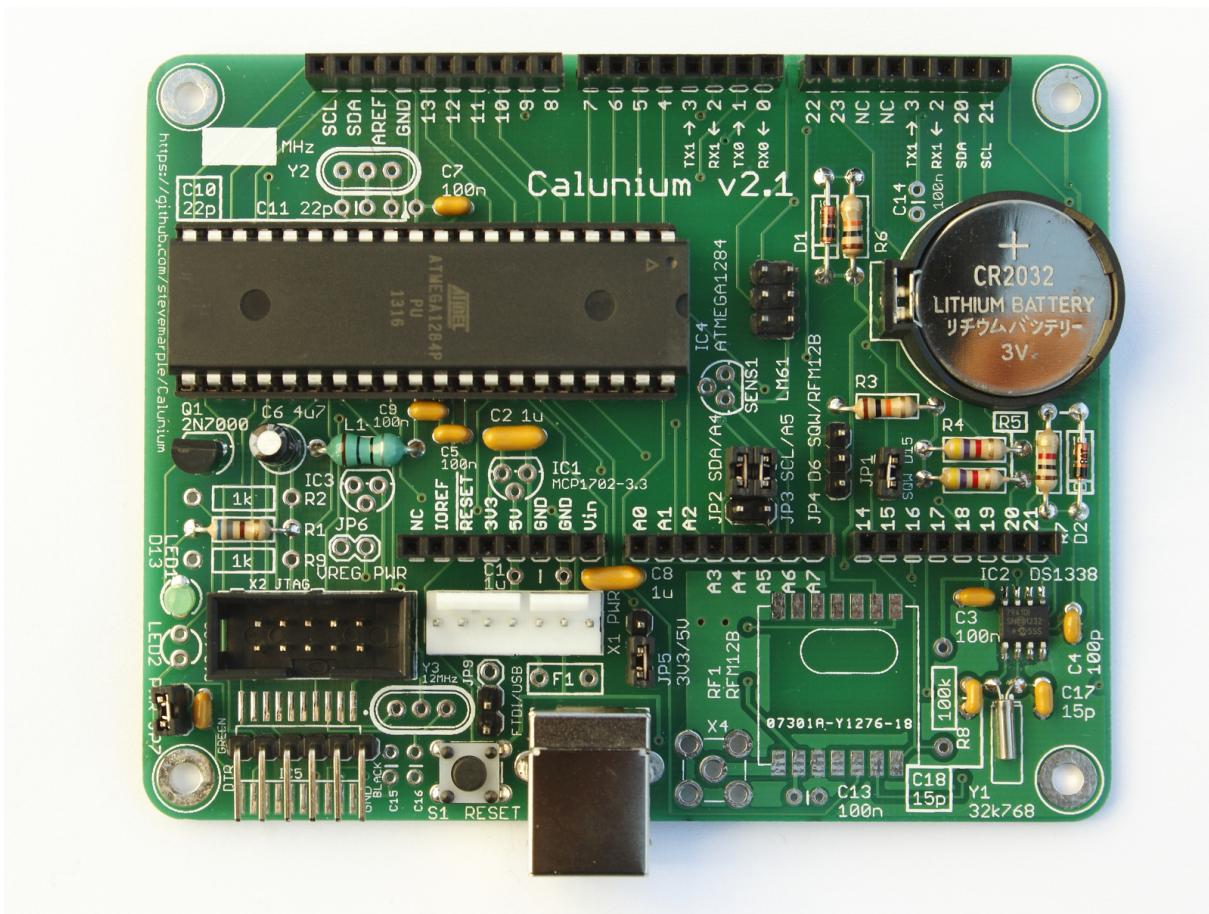


Figure 4.1: Completed Calunium v2.1. ©Steve Marple. CC BY-SA 3.0.

12. D1, D2 (BAT85). Do not fit D2 if using the DS1338-33 real-time clock.
13. LED1 (green LED). The cathode is nearest LED2, see figure ??.
14. C6 (4.7 μ F).
15. ICSP header (2×3 jumper block). See Figure ??.
16. JP2 and JP3. Fit as combined 2×3 jumper block.
17. JP1, JP7 (1×2 jumper).
18. JP4, JP5 (1×5 jumper).
19. X5 (1×6 right-angle or vertical header for UART0).
20. S1 (reset switch).
21. Arduino headers. **TO DO: Add description**
22. C17, C18 (15 pF). Do not fit if using DS1338-33 RTC. For Calunium verion 2.0 the capacitors must be fitted on the reverse side of the board (see Figure 4.3 as no specific mounting holes exist (error caused by using an earlier, incorrect datasheet which did not show the load capacitors)).
23. X1 (Molex power header). Ensure correct orientation, with the backplate closest to the Arduino headers.
24. **TO DO: Add component name** (RTC battery holder).
25. Q1 (2N7000). This item is very sensitive to damage by electrostatic discharge!
26. Battery (CR2032). Check that the battery backup pin (3) on the RTC measures 3.0 V.
27. **TO DO: Fit shunts to jumpers ...**

Do not fit the ATmega1284P microcontroller until after testing the board power supplies.

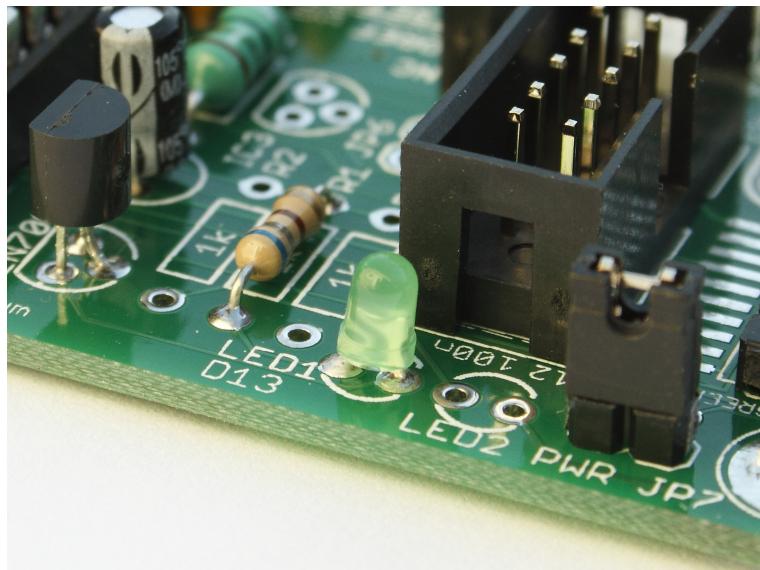


Figure 4.2: LED orientation. ©Steve Marple. CC BY-SA 3.0.

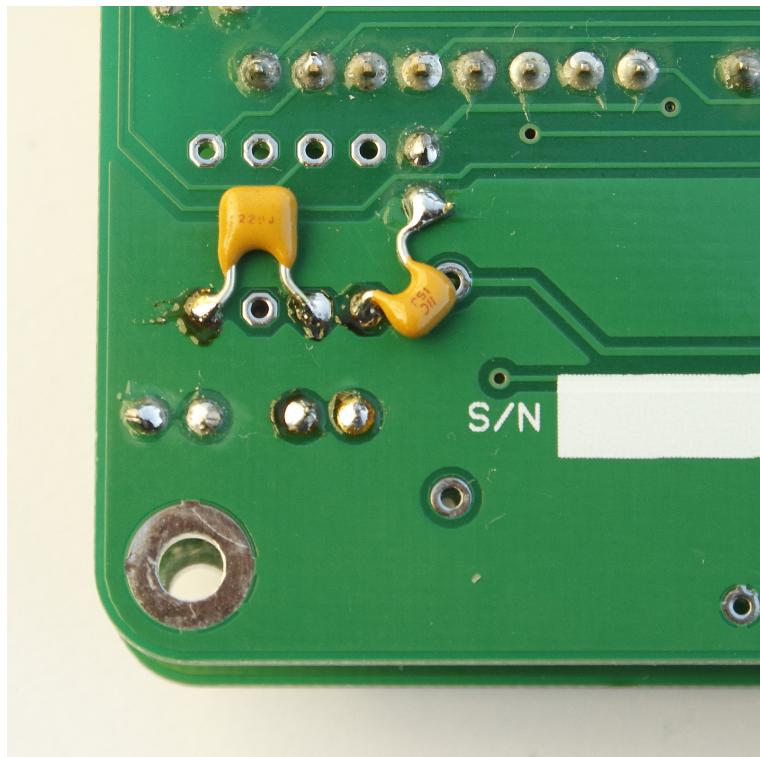
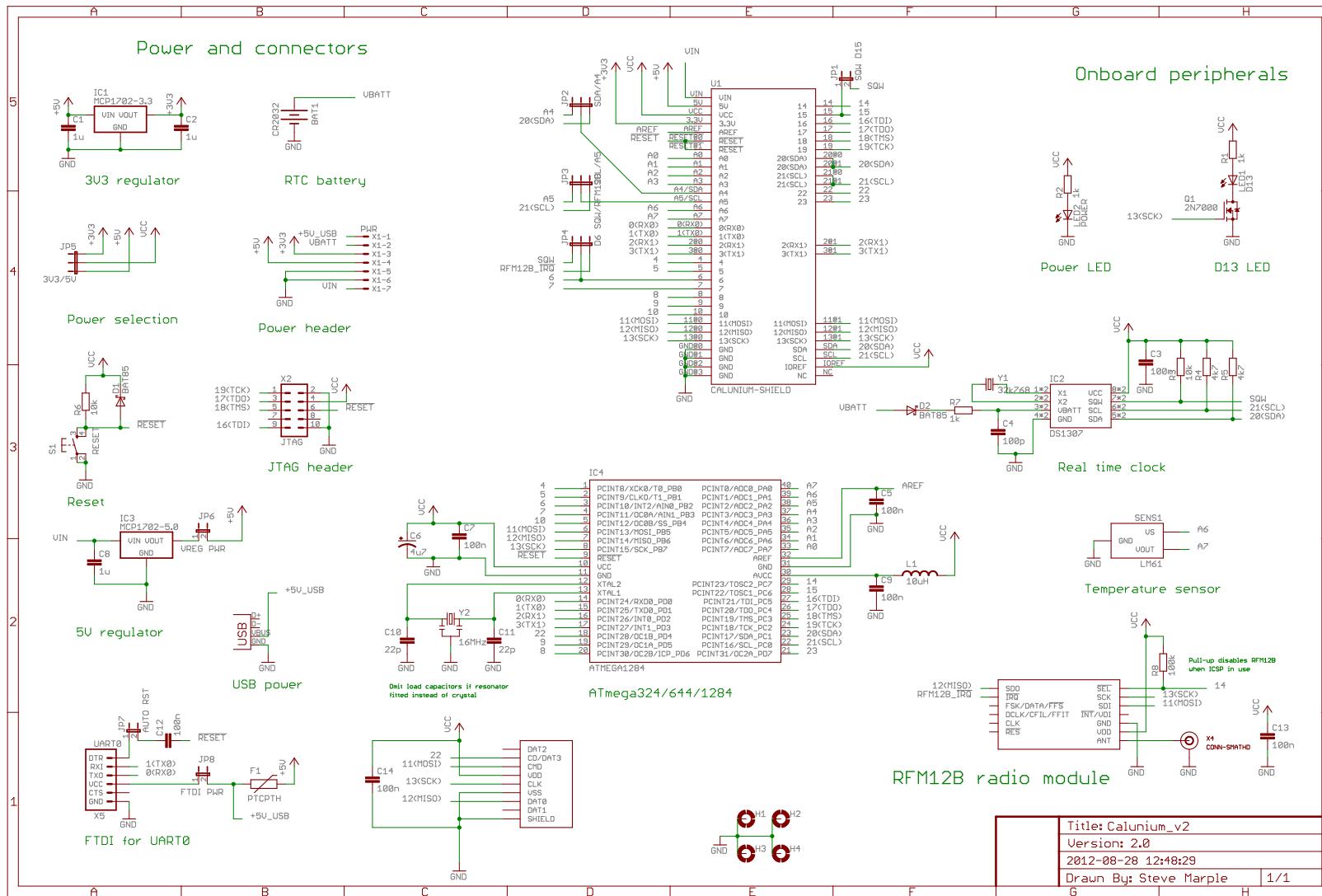


Figure 4.3: Real-time clock load capacitors (Calunium v 2.0). ©Steve Marple. CC BY-SA 3.0.



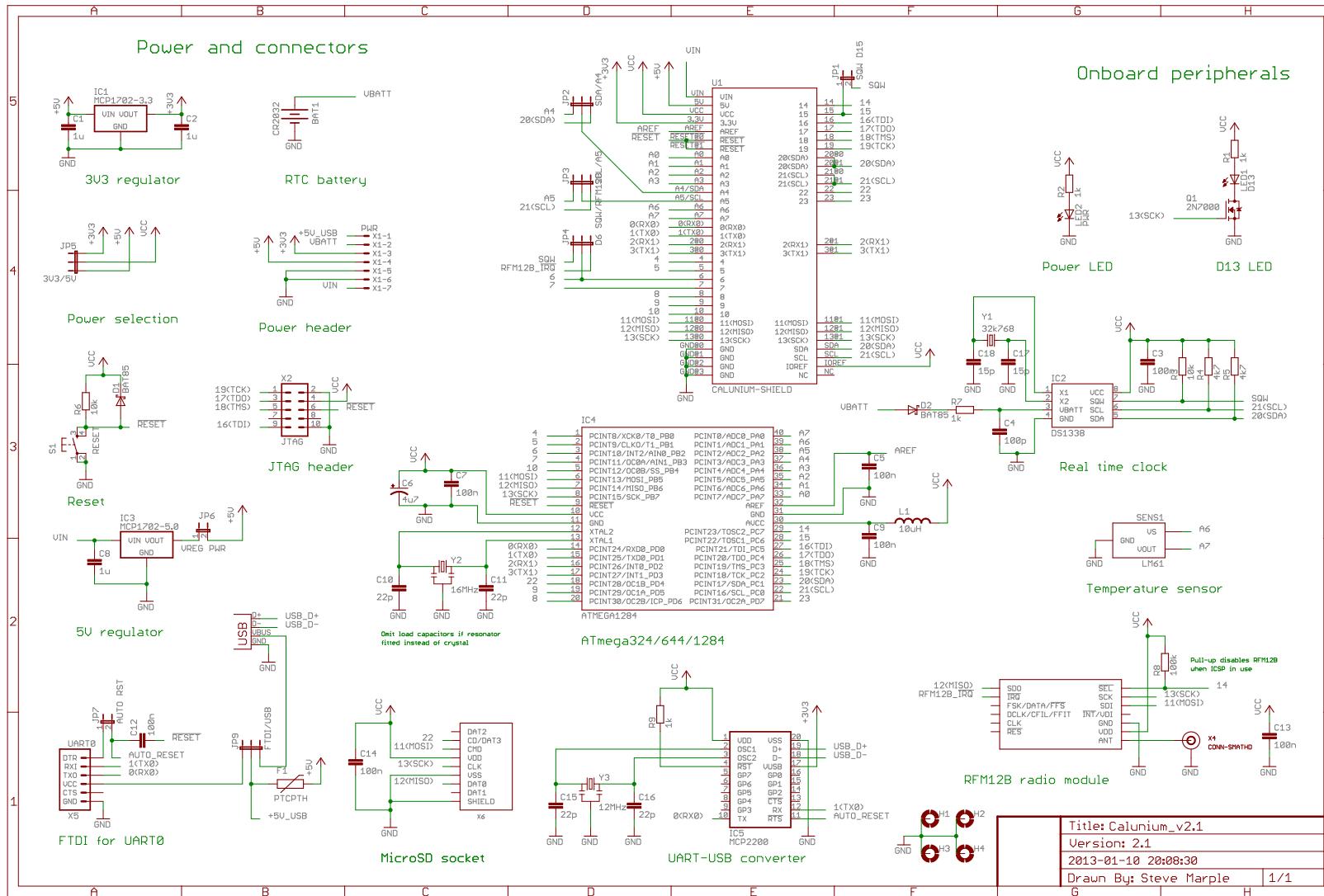


Figure 4.5: Calunium v. 2.1 circuit diagram.

4.3 Testing the board

4.4 Programming the firmware

These instructions assume you are using the Atmel AVR Dragon in ISP mode but adapting them to suit your programmer should be straightforward; see the `avrdude` manual page for further information.

4.4.1 Programming the bootloader

Power up the Calunium board and connect the programmer. Ensure the cable is correctly orientated at both ends. The bootloader can be compiled and programmed simply, as user pi:

TO DO: Check directory

```
cd /home/pi/xboot
make clean
make SHELL=bash calunium_8MHz_RC_ISP.conf.mk program
```

TO DO: ignore lock bit verification errors.

If the bootloader is correctly programmed the green LED connected to D13 on the Calunium PCB should flash at about 1 Hz.

4.4.2 Programming the magnetometer firmware

Ensure that the shunt marked “AUTO RST” is fitted, and that the shunt marked “FTDI PWR” is omitted. Connect the FTDI cable and identify the USB device file; as user pi:

```
dmesg | tail
```

Look for a line containing text similar to

```
FTDI USB Serial Device converter now attached to ttyUSB0
```

For the case above the device file `/dev/ttyUSB0`. Now program the microcontroller using the `xboot` bootloader. Replace `/dev/ttyUSB0` with the device file one your system. As user pi:

```
cd /home/pi/AuroraWatchNet/firmware/magnetometer  
avrdude -p atmega1284p -b 38400 -c avr109 -P /dev/ttyUSB0 \  
-U flash:w:xrf_rf12-0.10a.bin:r
```

- Whilst it is possible to program the firmware using the AVR Dragon alone this approach ensures that the xboot bootloader is present and functions correctly, allowing the microcontroller firmware to be updated over the radio link.

Chapter 5

Sensor PCB assembly

5.1 Sensor PCB version 1.2

5.1.1 Order of assembly

Fit components in order:

1. Turned pin sockets for the FLC100 sensor. Accurate alignment is important so use a piece of solderless breadboard to hold the male turned-pin headers (Figure 5.3). Insert the headers so that the conical part is pointing downwards. Fit the upside down turned-pin sockets onto the headers (Figure 5.4). Place the PCB onto the upside-down sockets (Figure 5.5) and solder all 7 connections. Remove from the breadboard, leaving the male headers in place. Carefully position the FLC100 sensor onto the male turned-pin headers. The top-side of the FLC100 has two yellow capacitors and the letters BS the circuit board. Solder the FLC100 sensor to the header. Gently remove the FLC100 sensor and place in an anti-static bag.
2. IC1 (MAXMCP3424).
3. IC2 (MAX619).
4. R1, R2, R3 (10 k Ω).
5. R4 (100 k Ω).
6. R5 (4.7 k Ω).
7. C1, C2, C5, C8 (100 nF).
8. C9 (10 nF).
9. C6, C7 (220 nF).
10. C3, C4 (4.7 μ F).
11. D1 (BAT85).
12. SENS2 (LM61).
13. JP5 and JP7 (fit as 2 \times 3 male header).
14. X1 (RJ45 vertical jack).
15. Q1 (2N7000). This item is very sensitive to damage by electrostatic discharge!
16. SENS1. Gently fit the FLC100 sensor into the turned-pin sockets. Apply pressure only on the circuit boards, not the components or coil.

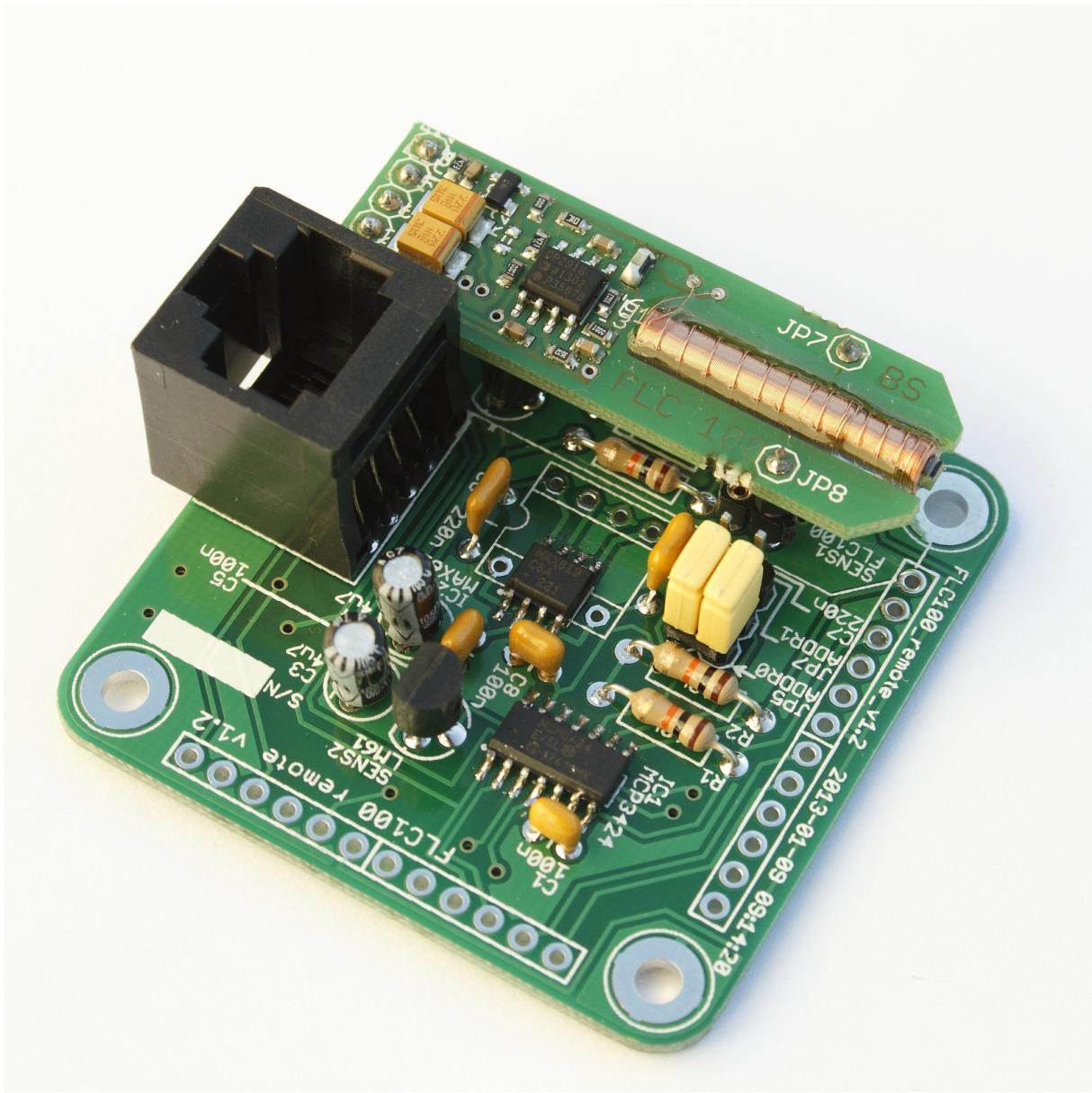


Figure 5.1: Completed sensor PCB (single-axis). ©Steve Marple. CC BY-SA 3.0.

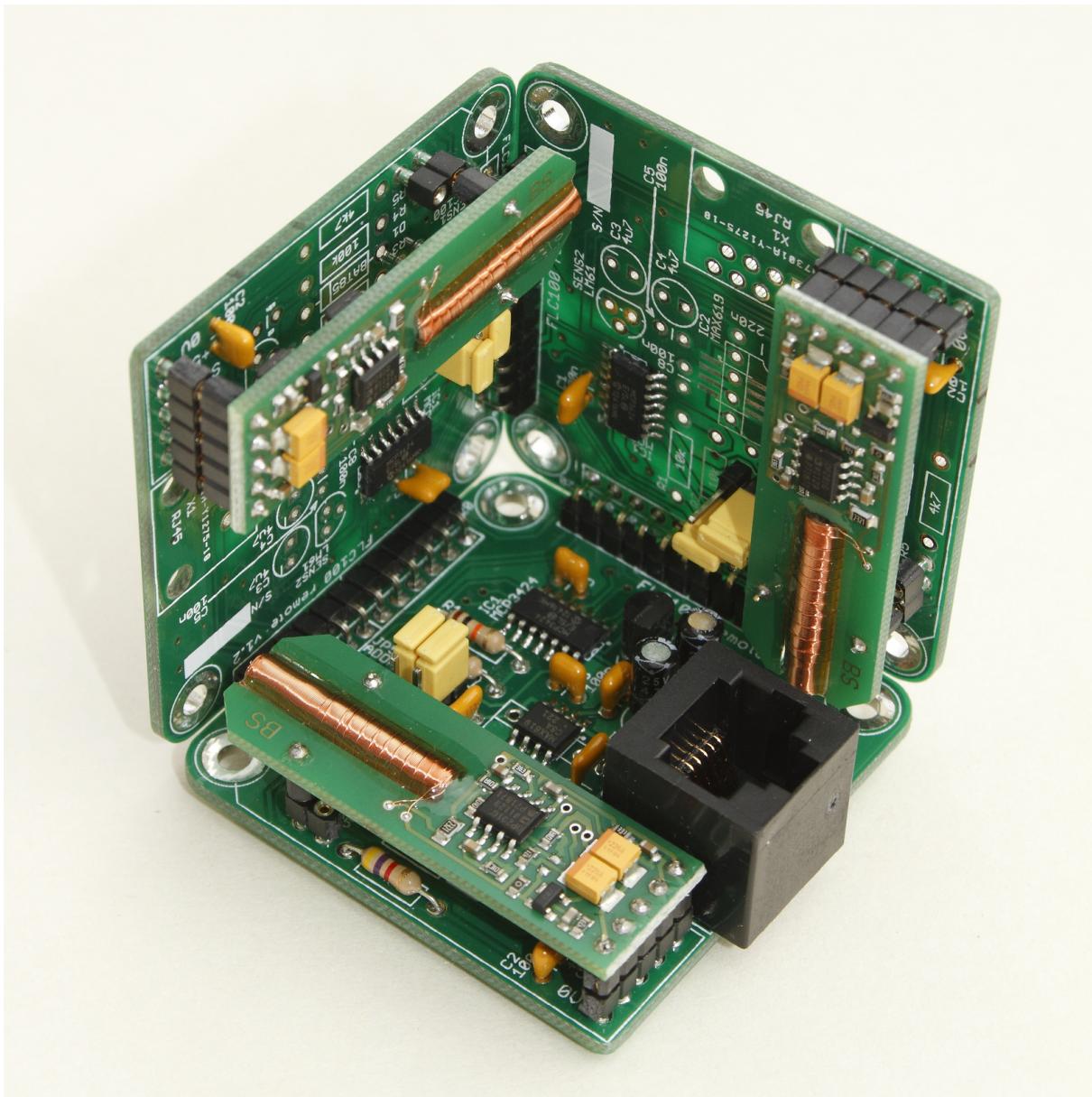


Figure 5.2: Completed sensor PCB (three-axis). ©Steve Marple. CC BY-SA 3.0.

TO DO: Take photo and insert

Figure 5.3: Fit the male turned-pin headers.

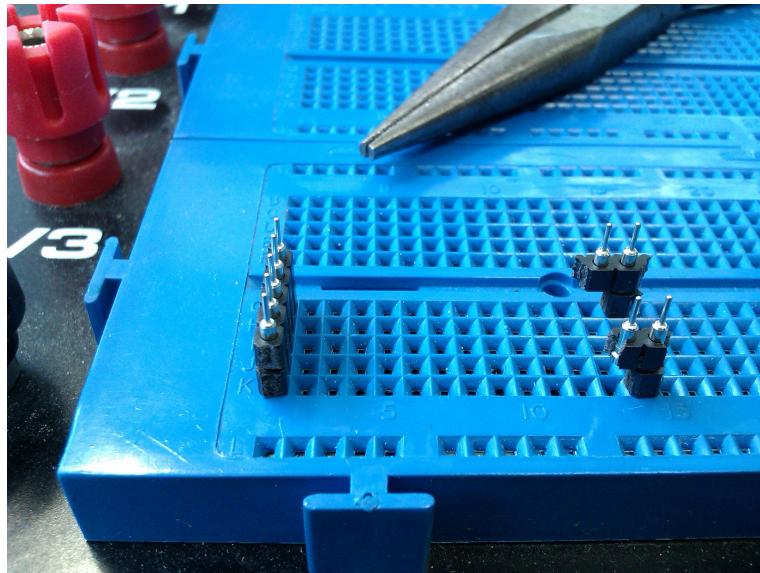


Figure 5.4: Fit the female turned-pin sockets.

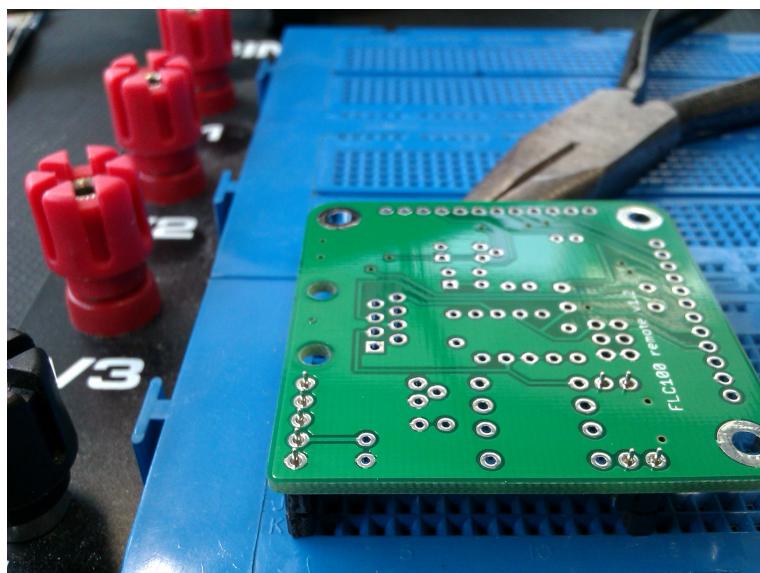


Figure 5.5: Place the sensor PCB onto the female turned-pin sockets. The pliers are used to support the other side of the PCB.

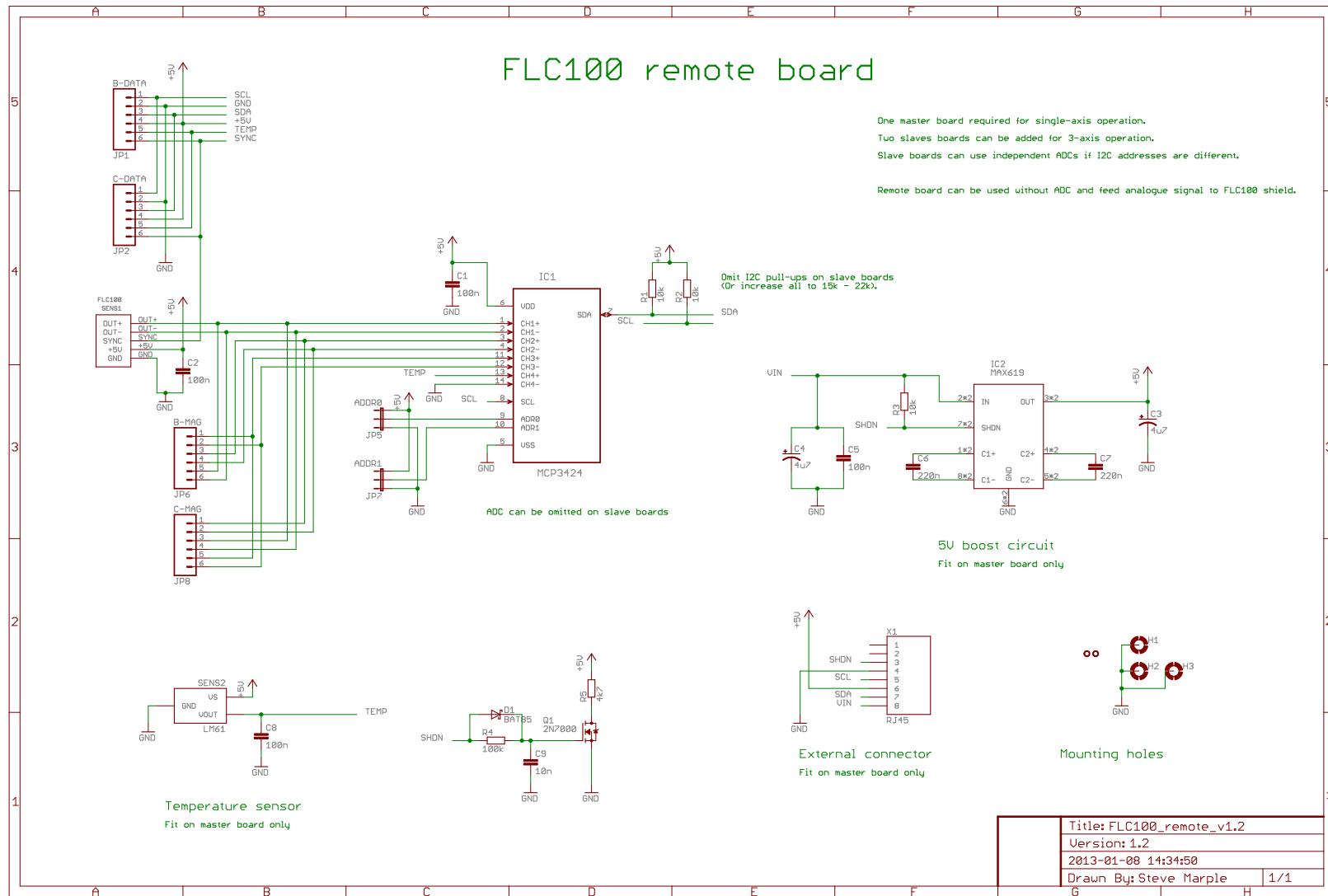


Figure 5.6: Sensor PCB version 1.2 circuit diagram.

Part III

Installation

Chapter 6

Site requirements

6.1 Sensor requirements

The site for the sensor should be chosen with regard to the following requirements (highest priority given first).

- Within range of the base unit.
- Away from moving metal objects, for example, trains (more than 50 m), cars (more than 20 m).
- Away from static metal objects, in particular those containing the *ferro-magnetic* materials iron, nickel and cobalt.

The FLC100 fluxgate magnetometer sensor is slightly sensitive to temperature variations. To ensure correct behaviour a stabilised temperature environment is required. This is obtained by burying the sensor.

6.2 Network requirements

The following network requirements are needed for the system to operate fully:

- DNS resolution. This is normally provided as standard on most networks.
- Outgoing access on port 80 (HTTP) and 443 (HTTPS). Required for software updates.
- Outgoing access on port 123 (NTP), or access to a local NTP server.
- Outgoing SSH access (port 22).

Chapter 7

Installation procedure

7.1 Determining the maximum range for the radio link

The maximum range can be determined with the following procedure:

1. Ensure that the Raspberry Pi is turned on and restart the recording daemon (p. 36). Monitor the recording process (p. 36).
2. Disconnect the remote sensor PCB from the FLC100 shield. Place the remaining circuitry about 5 m away from the Raspberry Pi and connect the battery. Keep the two units at least 2 m apart at all times to prevent overloading the radio receivers.
3. The green LED on the Calunium PCB will flash 3 times with a frequency of about 1 Hz (this indicates the bootloader is operating). The LED will then turn off.
4. About a second later the LED should light again, indicating the start of data transfer to the Raspberry Pi. If an acknowledgement is received the LED will turn off. This process should happen quickly, so that the LED flashes on momentarily. Just after LED turns on the Raspberry Pi should output the received data message, along with its response to the sensor unit. If the LED remains on it indicates the microcontroller has not received an acknowledgement; see section ??.

This step repeats itself following every data sampling operation, normally every 30 s. The time interval can be reduced by altering the sampling period (section ??). A functioning radio communication link is required to alter the sampling period.

5. Move the sensor unit to its intended location. Try to position it at the same height as it will be when buried, approximately 10 cm above ground. Keep the radio antenna vertical. If the LED continues to flash briefly every 30 s the radio link is operating correctly. If not then find another location.
6. If the radio link appears to work at correctly at the desired location then aim to find the maximum range by steadily increasing the distance between the sensor unit and base unit. Remember that the link is only tested after each sampling period. If the LED remains on move the sensor unit closer to the base unit and wait for the link to re-establish, indicated by the LED switching off.

- To save power the LED indicates the link status only for approximately 15 minutes after the power has been connected or the reset button pressed. If the LED has stopped indicating the link status press the reset button on the Calunium PCB.

7.1.1 Factors which alter the maximum range

The following factors can influence the range of the radio link:

- Obstructions. Try to find locations for the sensor unit and base unit which have direct line of sight. It is acceptable for walls to be in the path but note they will attenuate the signal. Avoid raised ground whenever possible. Placing the base unit on the first or second floor is likely to give improved range.
- Sources of radio noise. Some electrical equipment may cause RFI, which will reduce the range of the radio link. Try to keep both the sensor unit and base unit away from other electrical equipment, particularly those containing radio transmitters. “Smart” electric meters can contain radio transmitters which operate on the same 868 MHz ISM radio band.

Chapter 8

Raspberry Pi setup

8.1 SD card creation

Download the latest Raspbian image and copy to the SD card following the instructions on the Raspberry Pi web site. **Copying the compressed image to a FAT partition on the SD card will not work.**

8.2 Configuring Raspbian

 In the command window you can press the `Tab` key to have Linux complete the command or filename.

Raspbian is most easily configured by booting the new image. If you are able to discover the IP address (for instance, by checking the DHCP tables of your home router) you can do this over the network using SSH. Otherwise you must use attach a keyboard and monitor to the Raspberry Pi. If you are familiar with Linux it is also possible to edit the files by mounting the SD card on another Linux system.

8.2.1 /dev/ttyAMA0 serial port setup

Disable the console from running on `/dev/ttyAMA0`. Edit `/boot/cmdline.txt` to remove the parts which relate to `ttyAMA0`. Remove

```
console=ttyAMA0,115200 kgdboc=ttyAMA0,115200
```

Disable the getty process from running on /dev/ttymA0. Edit /etc/inittab. Find the line relating to ttymA0. Either delete the line entirely or comment it out by inserting a hash character (#) at the start of the line.

8.2.2 Raspbian configuration

Log in as pi and run

```
sudo raspi-config
```

8.2.2.1 Change user password

If the default password has not been changed then do so now to keep your system secure.

8.2.2.2 Internationalisation options

cron uses local time and the shift to and from daylight saving time complicates the cron tables. Set the Raspberry Pi's timezone to UTC to avoid daylight saving.

Select Internationalisation Options and then Change Timezone. For geographic area select None of the above, then select UTC. Select OK.

8.2.2.3 Advanced options

Select Advanced Options and then Memory Split. Set the GPU memory to 16 (MB).

8.2.2.4 Expand Filesystem

Finally select Expand Filesystem. Although the first option do this last. Choose Finish and then reboot.

8.2.3 Install missing software packages

As user root

```
apt-get install screen lsof python-pip ipython python-matplotlib \
    python-scipy avahi-daemon dnsutils
```

```
dpkg -i ~pi/python-numpy_1.7.1-3_armhf.deb
```

8.2.4 Remove swap file

To prolong the life of the SD card a swap file is not used. As user `root`

```
apt-get remove dphys-swapfile
```

8.2.5 Configure file system mount options

As user `root`

```
nano /etc/fstab
```

Find the line where the root file system is mounted, it will look similar to

```
/dev/mmcblk0p2 / ext4 defaults,noatime 0 1
```

Change the mount options (`defaults, noatime` in the example above) so that the mount options are now `noatime, nodiratime`. The line should look similar to the one below.

```
/dev/mmcblk0p2 / ext4 noatime,nodiratime 0 1
```

8.3 Installing the AuroraWatchNet server software

8.3.1 Install the Git repository

As user `pi`

```
git clone --recursive https://github.com/stevemarple/AuroraWatchNet.git
git clone --recursive https://github.com/stevemarple/auroraplot.git
git clone --recursive https://github.com/stevemarple/Calunium.git
git clone --recursive https://github.com/stevemarple/xboot.git
mkdir ~/bin
cd ~/bin
ln -s ../AuroraWatchNet/software/server/awnetd/awnetd.py
ln -s ../AuroraWatchNet/software/server/bin/log_ip
```

8.3.2 Configure python

As user pi

i The first line uses backticks, which on most keyboards can be found on the key above TAB.

```
user_site=`python -m site --user-site`
mkdir -p $user_site
cp ~/auroraplot/examples/example_auroraplot_custom.py \
    $user_site/auroraplot_custom.py
ln -t $user_site -s ~/auroraplot
```

8.3.3 Configure cron

As user pi

```
crontab -e
```

In the nano editor add the following lines to periodically report that the Raspberry Pi is operating, which helps the AuroraWatch administrators monitor which stations are active. This reporting step can be omitted if you prefer.

```
@reboot /home/pi/bin/log_ip reboot > /dev/null 2>&1
@hourly /home/pi/bin/log_ip > /dev/null 2>&1
```

Add the following lines to transfer data. **TO DO**

Save the file, **[CTRL]** - **[x]** , **[y]** , **[←]** .

8.3.4 Configure `ifplugd`

Configure `ifplugd` to report when the network interface has been assigned an IP address, which helps the AuroraWatch administrators monitor which stations are active. This step can be omitted if you prefer. As user `root`

```
cd /etc/ifplugd/action.d  
ln -s /home/pi/AuroraWatchNet/software/server/bin/log_ip
```

8.3.5 Create configuration file

As user `root`

```
mkdir /data  
chown pi.pi /data  
nano /etc/awnet.ini
```

TO DO: Create file contents, perhaps using a template copied from the repository

8.3.6 Create init file for server daemon

As user `root`

```
cd /etc/init.d  
ln -s /home/pi/AuroraWatchNet/software/server/awnetd/awnetd.sh awnetd  
update-rc.d awnetd defaults
```

8.3.7 Configure `ntp`

Check that the current time is correct by typing

```
date --utc
```

This will output the current date and time in UTC. If you aren't certain what the current time in UTC is then you can check at this web page, <https://www.google.co.uk/#q=utc+time>.

If the time is incorrect you will need to configure the NTP server. As user root edit /etc/ntp.conf:

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

Find the line(s) which start server and a insert # at the beginning of the line(s) to comment it(them) out. Insert a similar line which begins with the word server but is followed by your own NTP server hostname. Your network administrator should be able to provide this information. Finally restart NTP; as user root

```
/etc/init/ntp restart
```

Part IV

Operation

Chapter 9

Raspberry Pi operation

9.1 Introduction

For generic operation of the Raspberry Pi (setting the hostname, assigning a fixed IP address etc.) please see the Raspbian documentation, <http://www.raspbian.org/RaspbianDocumentation>.

9.2 Shutting down the Raspberry Pi

The Raspberry Pi must be shutdown cleanly before power is removed:

```
sudo shutdown -h now
```

Before removing the power wait until only the red power LED is lit; wait a further two seconds to ensure futher access to the SD card are not needed. If the power is removed whilst data is being written to the SD will corrupt it file system.

To reboot the Raspberry Pi use

```
sudo shutdown -r now
```

9.3 Starting and stopping the data recording daemon

Data is recorded on the Raspberry Pi using a *daemon* process, which is started and stopped by the Debian init scripts. The scripts must be started and stopped as user root, the actual data

recording process runs as user pi.

To start data recording

```
sudo /etc/init.d/awnetd start
```

To stop data recording

```
sudo /etc/init.d/awnetd stop
```

It is also possible to check the status of the data recording process

```
sudo /etc/init.d/awnetd status
```

The restart option forcibly stops recording (if running) and then starts it again:

```
sudo /etc/init.d/awnetd restart
```

9.4 Monitoring the data recording process

The data recording process directs its standard output and error streams to a virtual terminal using screen. It is possible to attach to this virtual terminal to monitor the output.

As user pi

```
screen -r awnet
```

To exit from screen type **[CTRL]** - **[a]** , **[d]** .

 Pressing **[CTRL]** - **[c]** will terminate the recording process.

Chapter 10

Software and firmware updates

10.1 Raspbian updates

TO DO: See Raspberry Pi web pages

10.2 AuroraWatchNet software updates

The software can be updated easily simply by *pulling* a new version from the Github repository.
As user pi

```
cd ~/AuroraWatchNet  
git pull
```

10.3 Sensor unit firmware updates

Only apply firmware updates if they are necessary. The firmware version selected must match the hardware, if it does not the sensor unit will be left inoperable and recovery will require an in-circuit serial programmer. First ensure that the AuroraWatchNet software has been updated (section 10.2) and the recording process restarted (section 9.3). To update the firmware

```
send_cmd.py --upgradeFirmware=name-version
```

Replace *name-version* with the firmware name/version string.