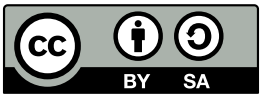
AuroraWatchNet magnetometer manual

Steve Marple,
Lancaster University.

October 1, 2013

Licence



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

Contents

List of figures

List of tables

Abbreviations and acronyms

Abbreviation / acronym	Meaning
ADC	Analogue to digital converter
ESD	Electro-static discharge
FET	field-effect transistor
GUI	Graphical user interface
IP	Internet Protocol
RFI	radio-frequency interference
RTC	Real-time clock
SAMNET	Sub-auroral magnetometer network
SSH	Secure shell

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-

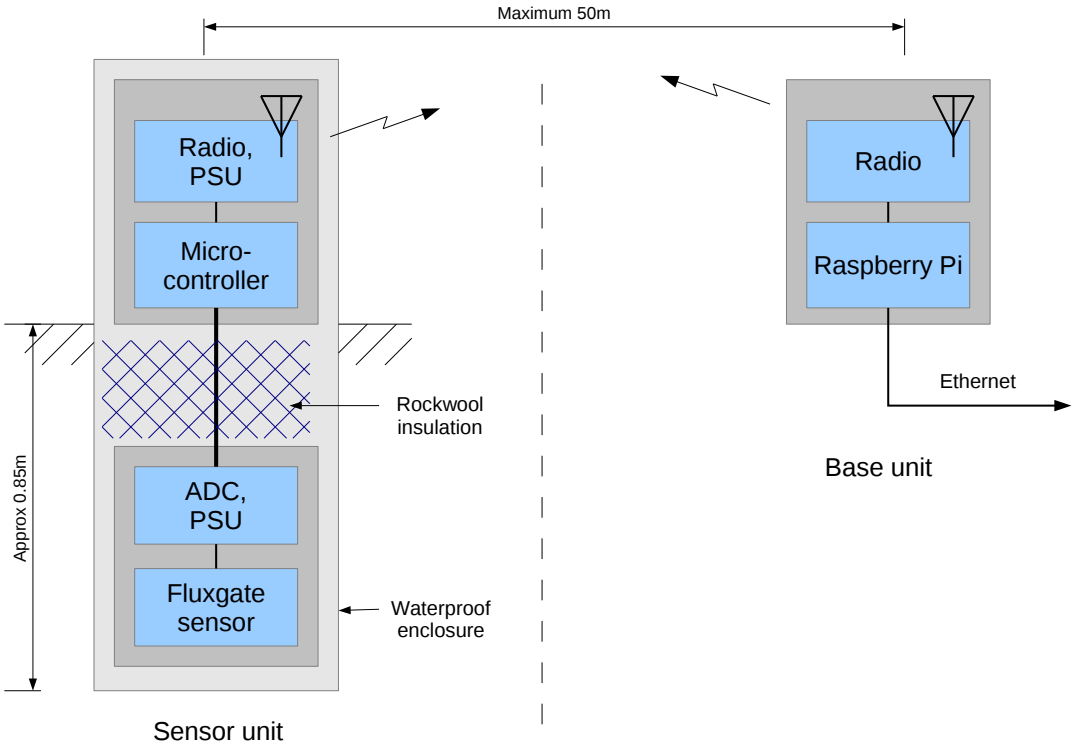


Figure 1.1: System overview.

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

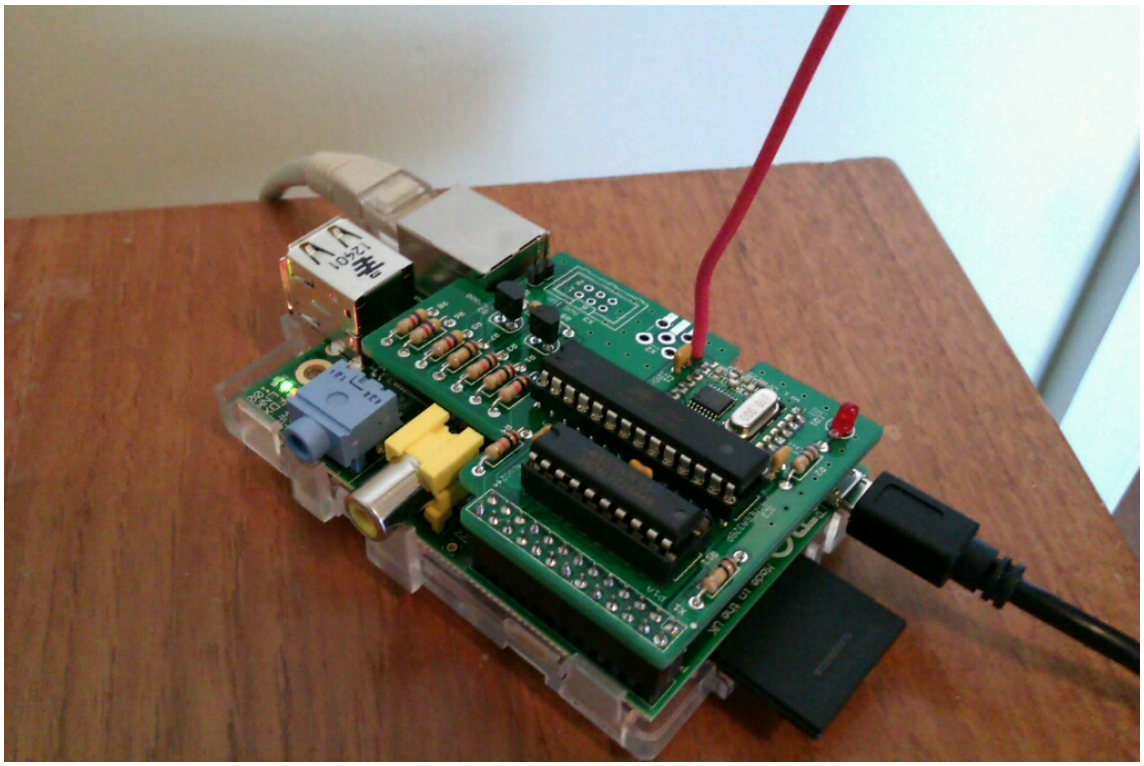


Figure 1.3: Base unit. ©Steve Marple. CC BY-SA 3.0.

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.
- Sidecutters.
- Small pliers.
- In-circuit serial programmer for Atmel AVR microcontrollers, e.g., Atmel AVR Dragon.
- 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 ver very difficult to access the pads of surface mount devies. It is also preferable that *passive* components (resistors, capacitors, inductors and crystals) are fitted before semiconductors (field-effect transistors, integreated 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 kΩ).
3. R11 (15 kΩ).
4. R10 (910 kΩ).
5. C15 (4.7 µF).
6. C16 (10 µF).
7. R1, R3, R4, R6, R8 (10 kΩ).

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).

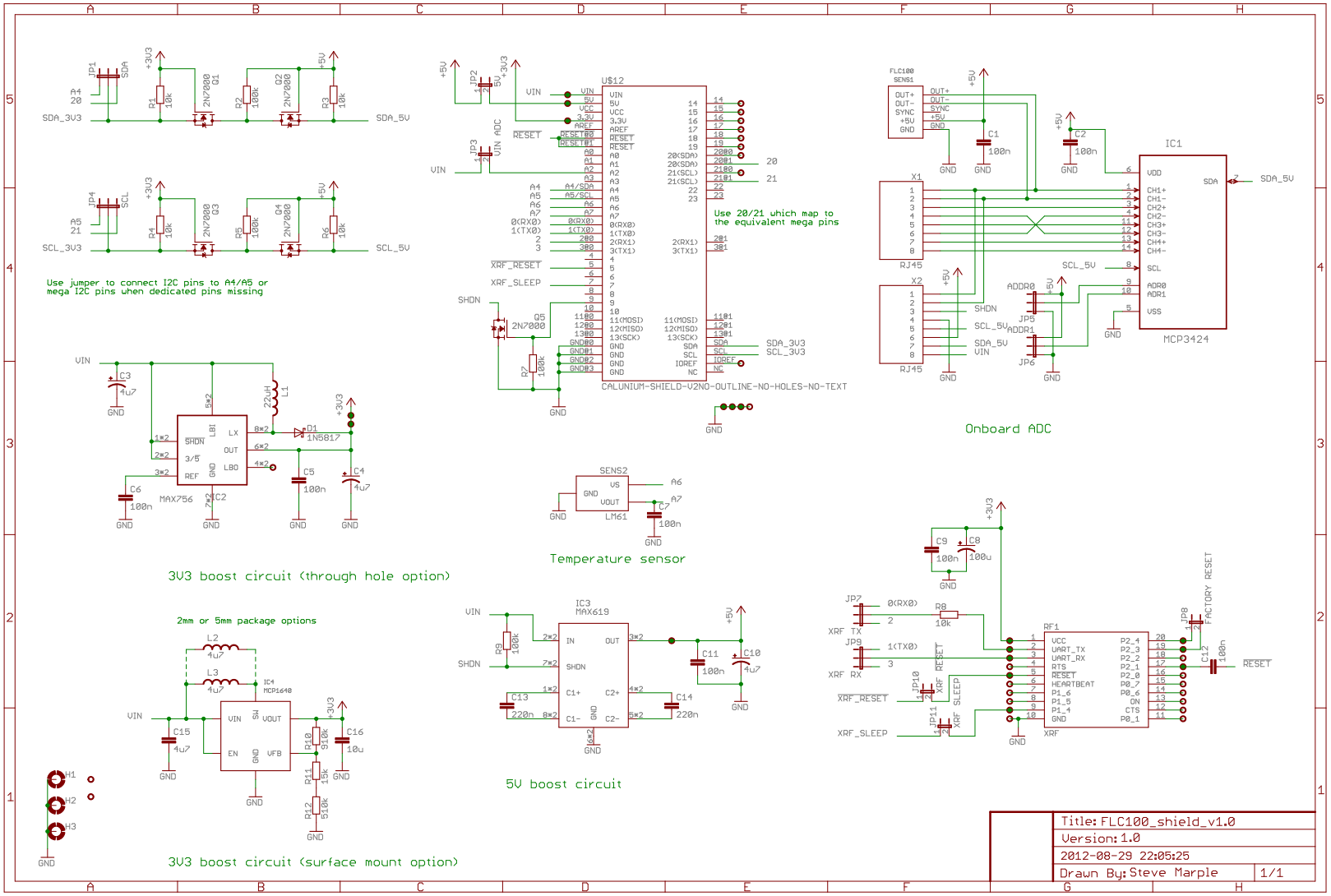


Figure 3.1: FLC100 shield v. 1.0 circuit diagram.

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 Ω resistor. Ignore the 1 kΩ marking; a lower value resistor is used to enable the green LED to be seen more clearly in daylight.
4. R7 (1 kΩ). 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 kΩ).
6. R3, R6 (10 kΩ).
7. L1 (10 µH).
8. 40 pin socket for IC4.
9. C4 (100 pF).
10. C3, C5, C7, C9, C12 (100 nF).
11. C2, C8 (1 µF).

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.
14. C6 (4.7 µF).
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 **??** 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.

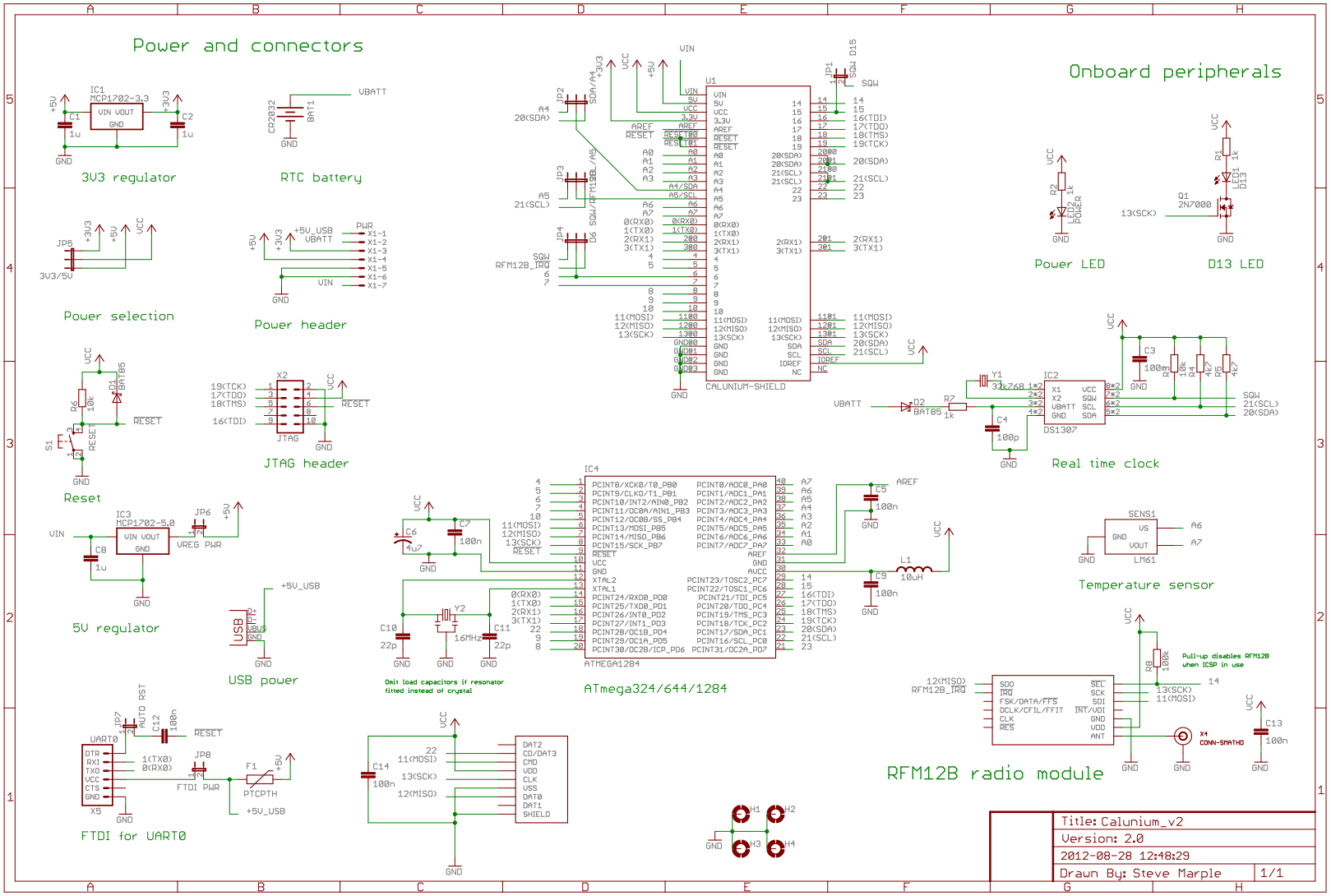


Figure 4.1: Calunium v. 2.0 circuit diagram.

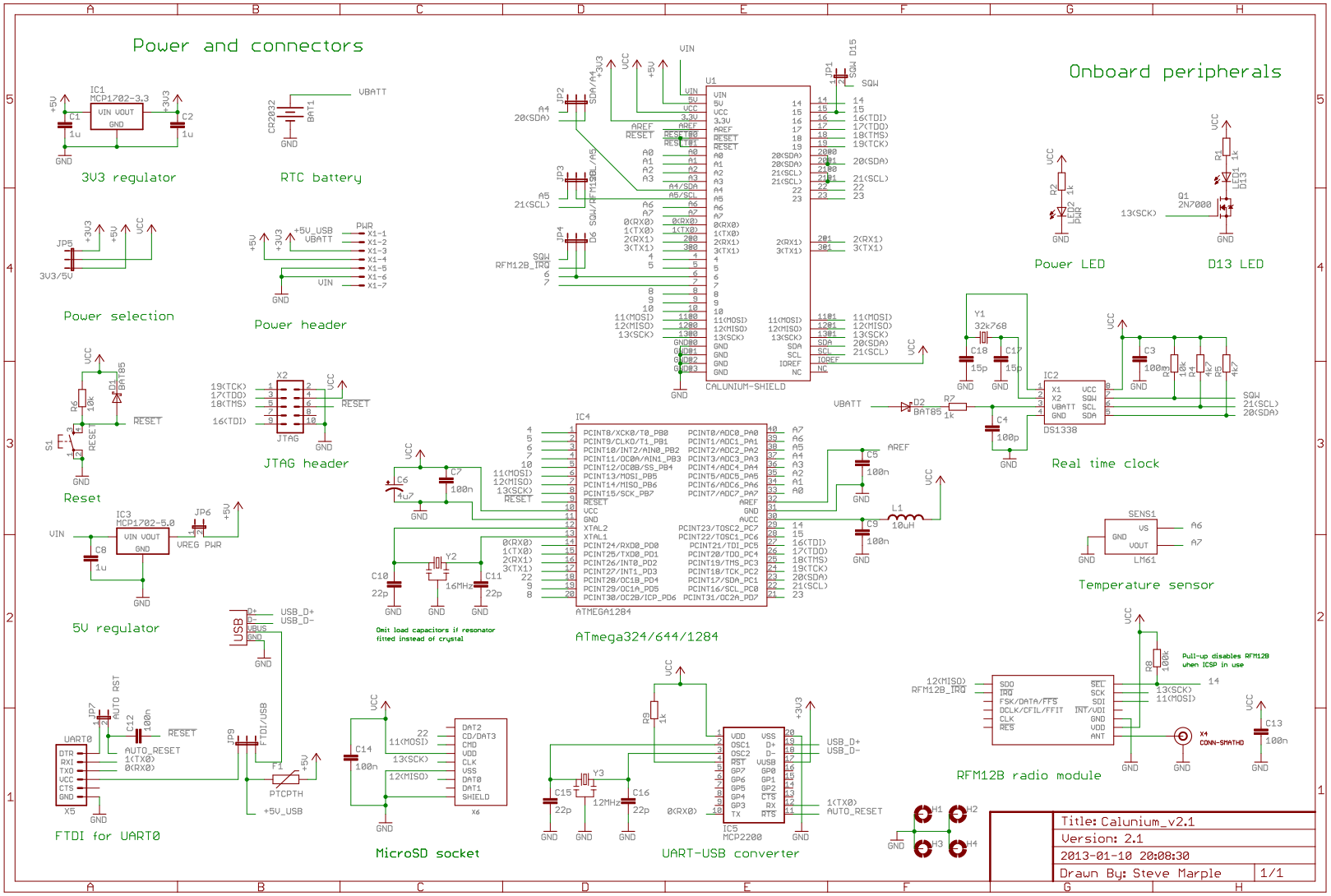


Figure 4.2: Calunium v. 2.1 circuit diagram.

4.3 Programming the firmware

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 peice of solderless breadboard to hold the male turned-pin headers (Figure 5.1). Insert the headers so that the conical part is pointing downwards. Fit the upside down turned-pin sockets onto the headers (Figure 5.2). Place the PCB onto the upside-down sockets (Figure 5.3) 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 kΩ).
5. R4 (100 kΩ).
6. R5 (4.7 kΩ).
7. C1, C2, C5, C8 (100 nF).
8. C9 (10 nF).
9. C6, C7 (220 nF).
10. C3, C4 (4.7 µF).
11. D1 (BAT85).
12. SENS2 (LM61).
13. JP5 and JP7 (fit as 2×3 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.

TO DO: Take photo and insert

Figure 5.1: Fit the male turned-pin headers.

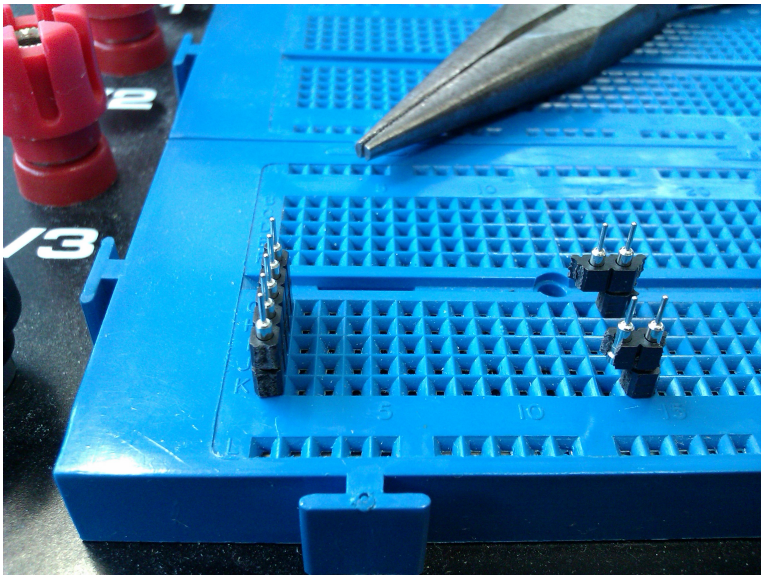


Figure 5.2: Fit the female turned-pin sockets.

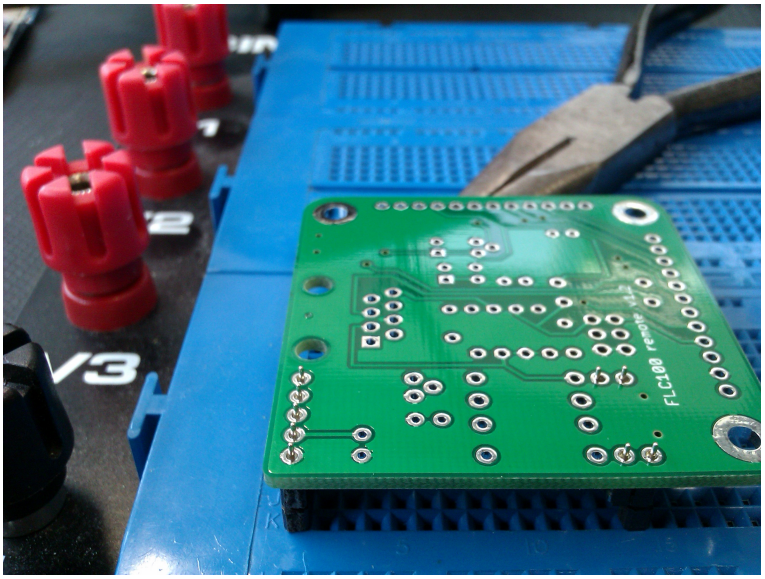


Figure 5.3: Place the sensor PCB onto the female turned-pin sockets. The pliers are used to support the other side of the PCB.

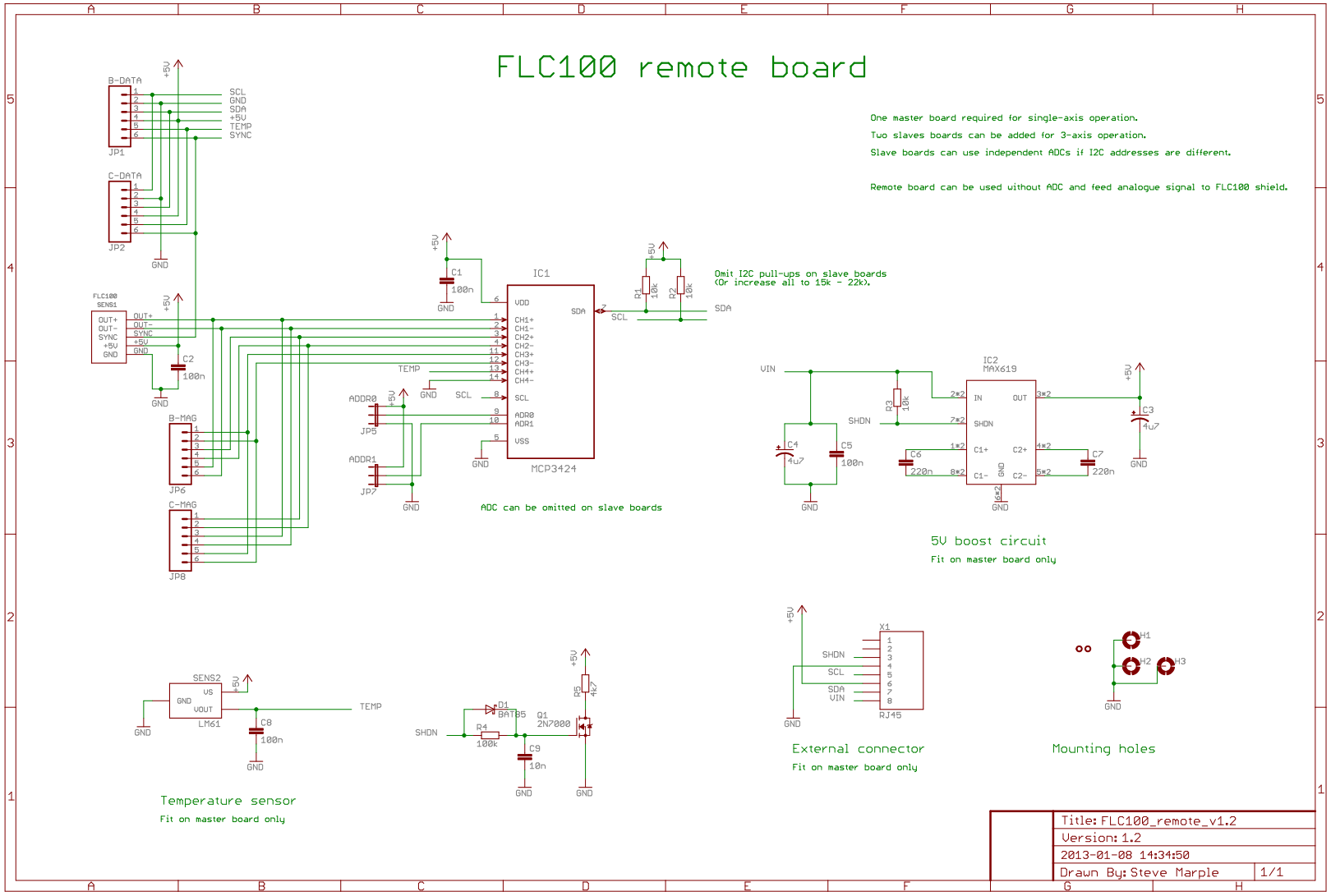


Figure 5.4: Sensor PCB version 1.2 circuit diagram.

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.

6.2 Network requirements

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

- Domain name system (DNS) resolution. This is normally provided as standard on most networks.
- Outgoing access on port 123 (NTP).
- Outgoing SSH access (port 22).

Chapter 7

Raspberry Pi setup

7.1 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**.

7.2 Configuring Raspbian

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.

7.2.1 /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/ttyAMA0`. Edit `/etc/inittab`. Find the line relating to `ttyAMA0`. Either delete the line entirely or comment it out by inserting a hash character (#) at the start of the line.

7.2.2 Raspbian configuration

Log in as `pi` and run

```
sudo raspi-config
```

7.2.2.1 Change user password

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

7.2.2.2 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`.

7.2.2.3 Advanced options

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

7.2.2.4 Expand Filesystem

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

7.3 Install missing software packages

As user `root`

```
apt-get install screen lsof
```

7.4 Installing the AuroraWatchNet server software

7.4.1 Install the Git repository

As user `pi`

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

7.4.2 Configure `cron`

As user `pi`

```
crontab -e
```

In the `nano` editor add the following lines: TO DO: Add lines for data transfer

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

Save the file, CTRL - x , y , ↵ .

7.4.3 Configure `ifplugd`

As user `root`

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

7.4.4 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

7.4.5 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
```

7.4.6 Configure `ntp`

TO DO: May not be necessary for most users, but all users should check that the time is correct

Part IV

Operation

Chapter 8

Raspberry Pi operation

8.1 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.

8.2 Shutting down the Raspberry Pi

The Raspberry Pi should be shutdown before power is removed:

```
sudo shutdown -h now
```

To reboot the Raspberry Pi use

```
sudo shutdown -r now
```

8.3 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
```

8.4 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 9

Software and firmware updates

9.1 Raspbian updates

TO DO: See Raspberry Pi web pages

9.2 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
```

9.3 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 9.2) and the recording process restarted (section 8.3). To update the firmware

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

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