Recording Power Supply

Version 6

Jed Marti KI7NNP

February 23, 2026

Abstract

Operating manual for a triple voltage power supply and recorder.

Contents

List of Figures

List of Tables

1 Introduction

The power supply provides three independent voltages from 12 - 25 volt input sources. Each section has a current and temperature monitor connected to an I^2C bus.

Voltage A From a buck regulator drops the input voltage to from 3 to 5.5 volts up to 3 amps. Normally set to 5.1 volts to power Raspberry Pi's. Powers on-board sensors that are limited to 5.5 volts.

Voltage B From a buck/boost regulator provides output from 3 to 30 volts at up to 3 amps. Normally set to 12 volts to power monitors and AREDN transceivers.

Voltage C From a buck/boost regulator provides output from 3 to 30 volts at up to 3 amps. Normally set to 19 or 24 volts.

To lower cost, not all components need be populated. The supply can function without the recording monitor. Figure 1 shows a small ammunition box with a 30 Ah $LiFePO_4$ battery and recording monitor.

Figure 1: Power supply and monitor/recorder with 30 Ah battery

Figure 2 shows the 4 layer PC board with the A, B, and C regulator sections from the top to the bottom. Each regulator section is protected by a fuse. Each regulator section has a potentiometer to set the desired output voltage.

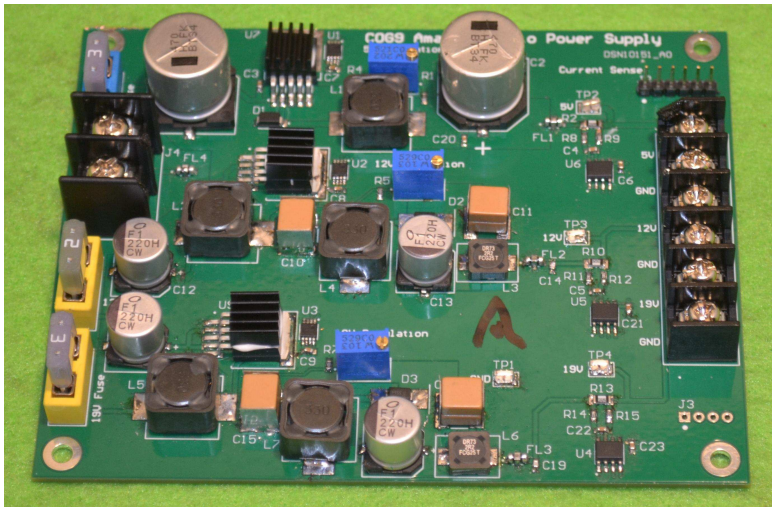

Figure 2: Populated power board.

The recording monitor in Figure 3 is powered by an ATMEGA1284 with 128k flash and 16k RAM. A 128 x 64 OLED, time and date clock, 128k serial EEPROM, and push buttons support its operation. It is normally powered by the supply board but can be operated independently (mostly for debugging).

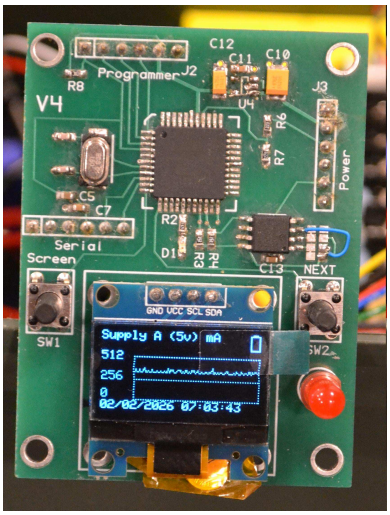

Figure 3: Populated monitor board.

2 The Power Board

The 4 layer power supply board is divided into 3 areas: A, B, and C as shown in Figure 4 Each section has its own mini-fuse, (on the left hand side), and 20 turn potentiometer for voltage (blue rectangle).

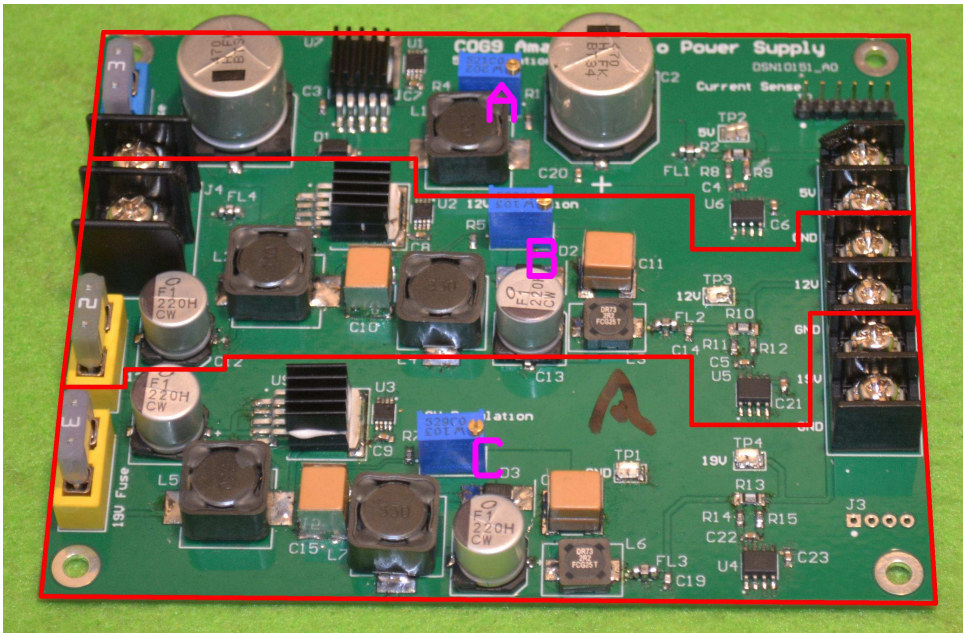

Figure 4: Power board areas.

2.1 General

Battery input is the two screw terminal on the left. Positive on the top, ground on the bottom. The supply is not protected from reverse wiring. **The magic smoke will come out!** The 6 pin header at the top right corner goes to the recorder monitor supplying power, battery voltage, and I^2C from the sensors. A 4 pin header on the lower right corner is for the auxiliary battery temperature sensor.

2.2 A: 5 volt section

The 5 volt regulator in section A, must be set before the board is fully populated as its voltage drives the 6 I^2C sensors. It can be adjusted down but should never output more than 5.5 volts

or these sensors will be fried. The 20 turn potentiometer just above the magenta letter A sets the voltage.

The board may be assembled without the sensors in which case the output voltage may be run up to somewhat less than the battery output.

This section must be populated for the sensors to work but is not necessary. Removing or blowing its fuse will still allow the 12 and 19 volt sections to operate but the recorder will not operate.

2.3 B: 12 volt section

The 12 volt section is a buck/boost regulator with adjustable output. A typical value is slightly more than 12 volts to account for voltage drops at higher amperages. The output is adjustable with the 20 turn blue box potentiometer just above the magenta letter B. It has a fuse on its input typically the maximum output current expected.

The board may be assembled without this section or if its output is not needed, the fuse can be pulled to save idle current draw.

2.4 C: 19 volt section

The 19 volt section has the same circuitry as section B. It could be set to the same as section B to expand 12 volt amperage or to 5 volts to expand that capability. It need not be populated or if not used, the fuse can be pulled to save the idle current. Its output voltage is adjusted by the 20 term potentiometer next to the magenta C.

2.5 Connection to Recorder/Monitor

The 6 pin header in the upper right corner of Figure 4 on page 5 has 5 active pins. The ground connection is toward the inside of the power board. The blue wire is a direct, non-fused connection to the battery for the monitor to read its current output voltage using an ADC.

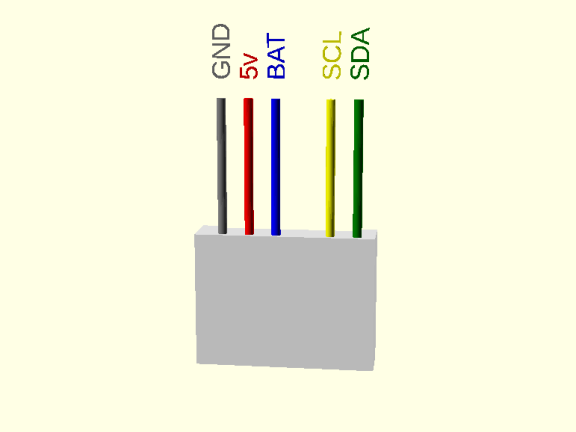

Figure 5: 6 pin header to recorder monitor

2.6 Connection to Auxiliary I^2C sensors

The 4 pin header in zone C provides power and 400 kHz I^2C for one or more auxiliary sensors. Typically this is for a temperature sensor attached to the battery to help warn of over temperature.

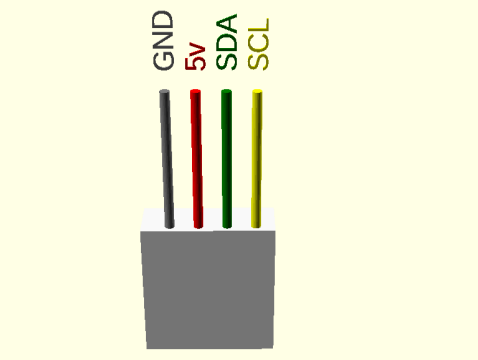

Figure 6: 4 pin header for auxiliary sensors.

3 Recorder/Monitor

The recorder is based on the ATMEL Atmega 1284 MCU with 128k flash, 16k RAM, and 4k EEPROM. I/O includes two push buttons, a bright red LED, 115.2k baud serial port, a 128x64 OLED, 128k serial EEPROM and a battery backup date/time clock.

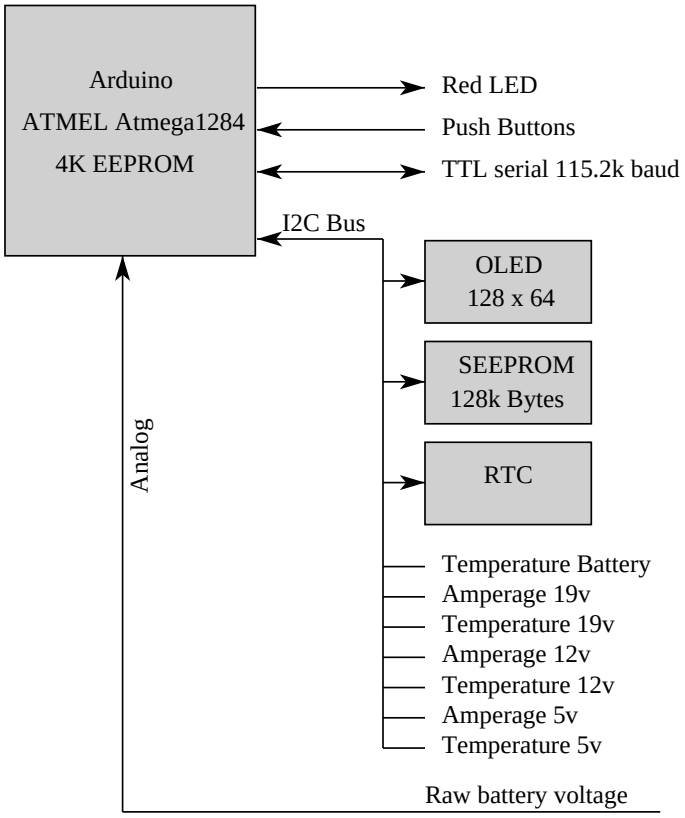

Figure 7: Recorder/monitor structure

The push buttons are used during power on to start recording or select between available screens during operation. The serial port provides fine control of its operation.

3.1 Startup

The recorder can be triggered during power on startup or at any later time through the serial port command line.

Figure 8: Recorder/monitor Control Panel

To start recording on power up without recourse to the serial interface, push one or both of the buttons before powering on holding them until the power up screen appears. Then release, recording starts automatically overwriting any previously stored values.

Prev	Next	Recording
up	up	Do not start recording.
up	down	Start recording current on all 3 channels.
down	up	Start recording current and voltage on all 3 channels.
down	down	Start recording all values: current, voltage, temperature.

Table 1: Push buttons during power up.

3.2 Screens

During power on, various activities are scrolled across the screen when they are completed or not. Afterward, a selected screed is updated every so often. Pressing and releasing the **Next** button moves to the next screen and **Previous** to the previous one, wrapping around when= the end or beginning is reached.

There are 12 screens only one of which can be displayed at a time.

1	Start page	Version, release, owner.
2	Power page	Shows current, voltage, temperature numerically.
3	Temp. page	Shows the current temperature values.
4	Error page	If errors are encountered, a message will appear here. If no errors, then the page is shown but blank.
5	Current A	Shows a graph of A current draw, updated in real time.
6	Current B	Shows a graph of B current draw, updated in real time.
7	Current C	Shows a graph of C current draw, updated in real time.
8	Temp. A	Shows a graph of A regulator's temperature in real time.
9	Temp. B	Shows a graph of B regulator's temperature in real time.
10	Temp. C	Shows a graph of C regulator's temperature in real time.
11	Temp. Aux	Shows a graph of the auxiliary temperature in real time.
12	Battery	Shows a graph of battery voltage in real time.

Table 2: OLED screen types.

3.3 Serial Port Interface

Trying to do too much with two buttons is a mistake. The serial port interface offers fine grain control and communication with more capable devices. The default configuration is 115.2k baud, 8 bits no parity. A 3.5mm stereo plug provides ground, transmit and receive at TTL levels.

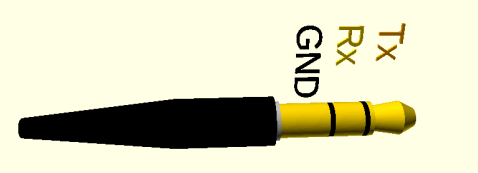

Figure 9: Serial port 3.5mm audio connector

Serial data can be accomplished with **PUTTY**, **Terraterm**, or **cterm**. The AREDN specific **kserver** program will use the serial port to supply SVG output and download collected data.

3.3.1 Command Line

The serial port is read until a new line character is entered. Backspaces are allowed. The maximum input line length is 80 characters[1].

Commands are case insensitive and are a keyword optionally followed by values to help their case. The recorder is ready to accept a command when a ? appears.

Command	Arguments	Description
amps		Show voltages and current draws 3.3.2.
batcalib		Calibrate battery reading 3.3.3.
csv		Dump recorded data in CSV format 3.3.4.
cterm	*file-name*	Dump saved data to a **cterm** monitor 3.3.5.
date	[*date*]	Get the current date and time or set it 3.3.6.
eeprom	[*name value*]	Dump EEPROM contents or set a value 3.3.7.
low	*volts*	Set the battery low voltage trigger3.3.8.
off	*volts*	Set the battery voltage to trigger power down 3.3.9.
owner	*call-sign*	Set the owner call sign 3.3.10.
over	*volts*	Set the battery over-voltage warning 3.3.11.
regs		Dump register values from INA219's 3.3.12.
rate	[*integer*]	Set the sample rate per second 3.3.13.
reset		Reset EEPROM values to default 3.3.14.
samples	[*integer*]	Set the number of sample to average 3.3.15.
serve		Dump the current status as HTML to kserver 3.3.16.
set	ABC *reg value*	Set an INA219 register value 3.3.17.
svg	...	Generate SVG graph for kserver 3.3.18.
start	[**amps**,**volts**,**temps**,**all**]	Start recording 3.3.19.
stop		Stop recording 3.3.20.

Table 3: Serial Port Commands.

[1]Useful if your input comes from 80 column punch cards. Tradition dies hard

3.3.2 amps command

Show the three raw readings from each current monitor: the shunt voltage, output voltage and current draw. Only devices found will be displayed.

```
?amps
A: shunt 2.29 mv, bus 5.05 V, draw 40 ma
B: shunt 0.42 mv, bus 12.18 V, draw 8 ma
C: shunt 0.60 mv, bus 19.44 V, draw 0 ma

?
```

3.3.3 Battery Calibration

This command needs to only be done once.

3.3.4 CSV output

Dump the contents of the recorded data onto the serial port in CSV format. See section 3.3.19 on page 19 for details.

3.3.5 cterm output

Dump the stored data into a CSV file using the **cterm** protocol. The command must be followed by the file name on the host machine to store the data in.

```
?cterm foo.csv
Starting upload
Uploading 'foo.csv'
waiting
Sending foo.csv
.............................................................
.............................................................
......................
'foo.csv' uploaded
foo.csv sent
?
```

The first few lines of stored data might look like the following:

```
month,day,year,hour,minute,second,maA,maB,maC
2,18,2026,15,41,38,34,8,0
2,18,2026,15,41,40,34,8,0
2,18,2026,15,41,42,34,8,0
...
```

Current values are in milliamps, voltages in volts, temperatures in Fahrenheit.

3.3.6 Date

The time/date clock's backup battery will discharge over the course of a week or so if you don't turn the device on. Likewise, the date and time will drift and need to be reset. You query the current values with just the command.

```
?date
02/22/2026 19:04:06
?
```

To set the date/time, use the following format:[2]

date *day-of-week month day year hours*:*minutes*:*seconds*

where:

day-of-week Spell out the day of the week: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday.

month Spell out the month: January, February, March, April, May, June, July, August, September, October, November, December.

day Day of the month, from 1 to 31.

year The full year number.

hours*:*minutes*:*seconds For 24 hour clock.

For example:

[2] Got carried away with this.

```
?date Sunday February 22 2026 19:10:00

Date set
?date
02/22/2026 19:10:58
?
```

3.3.7 EEPROM contents

System parameters are saved across power off periods in the Atmega's EEPROM. You access the current values with the **eeprom** command.

```
?eeprom
EEPROM configuration
 Owner: KI7NNP
 Battery M=0.023 B=-0.120
 Battery overvoltage 14.5, low 10.0, done 5.0
 Max temp 125 f
 Sample Rate 10/sec, Size 20
 Graph MS update 2000
 Battery ovf Wh
    Last ovf Wh
Record currrents
?
```

The values stored are:

owner Your call sign, used when generating HTML and SVG files and on the startup screen. Up to 7 characters.

M Slope of line for ADC reading of battery voltage.

B Zero intercept for ADC reading of battery voltage. $v = Mi + B$ where i is the ADC reading.

over Battery over voltage - exceed this and get warnings.

low Battery lower than this value triggers warnings about low battery.

done Battery lower than this value triggers recording stop - soon going away.

maxtemp Maximum temperature in Fahrenheit of any temperature reading.

rate The time in milliseconds between current/voltage/temperature readings (not the number of readings to take per second).

samples How many samples to take and average before changing graphs or recording.

graph Number of milliseconds between graph updates.

battery The watt hour battery capacity.

last The last number of watt hours used.

which Which data structure is being recorded. Value is set by the **start** recording command or push buttons at startup.

Record currents Only recording the three milliamp readings.

Record currents, voltages Record the three milliamp and voltage readings.

Record currents, temperatures Record milliamp and temperature readings.

Record everything Currents, voltages, temperatures,

You can change these by:

eeprom *item value*

For example to take only 5 samples per second, we set the rate to 200 milliseconds.

```
?eeprom rate 200
rate now 200
?eeprom
EEPROM configuration
 Owner: KI7NNP
 Battery M=0.023 B=-0.120
 Battery overvoltage 14.5, low 10.0, done 5.0
 Max temp 125 f
 Sample Rate 5/sec, Size 20
 Graph MS update 2000
 Battery ovf Wh
    Last ovf Wh
Record currrents
?
```

3.3.8 Low Voltage Trigger

The battery voltage is monitored at the sample rate. If it falls below this value, the blinking LED goes faster, and a warning appears on the serial port and the error screens yells to finish work.

You can change the level with:

low *voltage*

3.3.9 Trigger power down

The monitor can't turn off the power supply, the battery management system will do it for you. But when the battery voltage falls below this level, a serial port warning message is triggered and recording is stopped.

You can change the voltage with:

off *voltage*

3.3.10 Owner

Your call sign goes here. It's used to identify graphical data and HTML output and appears on the startup screen. Up to 7 characters can be saved and are stored in the EEPROM.

3.3.11 Over Voltage Trigger

Set the voltage that shouldn't happen. Perhaps the battery is running wild. A warning appears on the serial port, the LED blinks like crazy and the error screen signals the problem.

3.3.12 INA219 Registers

Dump the contents of all six INA219 registers. For debugging.

```
?regs
  Config  A:0x199F B:0x399F C:0x399F
  Shunt   A:0xF4 B:0x2B C:0x3A
  Bus     A:0x277A B:0x5F2A C:0x97D2
  Power   A:0x0 B:0x0 C:0x0
  Current A:0x0 B:0x0 C:0x0
  Calib   A:0x0 B:0x0 C:0x0
?
```

3.3.13 Set Sample Rate

Set the number of samples taken per second. Samples are collected and averaged before storage though that shown on screens will be the latest value, not the average. The system is probably not fast enough to do more than 20 per second. See also 3.3.15.

3.3.14 Reset EEPROM values to defaults

Reset the EEPROM values to their defaults, erasing the owner and so on. Try not to use.

3.3.15 Number of Samples to Average

Sets the number of samples to collect and average before writing to the serial EEPROM. These are collected at the **rate** per second, see 3.3.13.

For example, if the sample rate is 5 and you set the samples to 10, an average will occur every 2 seconds.

3.3.16 Send HTML status to kserver

The **kserver** web server program can request data from the a serial port. This command creates an HTML file.

1. Send ASCII character 0x01 (↑A) to indicate start of serial HTML.
2. Send the HTML document.
3. Send ASCII character 0x04 (↑D) to indicate end of serial HTML.

For example:

```
?serve
^Aa<html>
<head>
<style>
table, th, td { border-spacing: 2px; border-color: gray; border: 1px solid black; }
</style>
<title> KI7NNP Power Supply </title>
</head>
<body>
<h1 align="center"> KI7NNP Power Status</h1>
```

```
<p><b>POWERARD</b> V6.00.001<p> Battery 13.28 volts
<p> <b>A: </b> 5.05V   40ma  66F   0.48 Wh
<p> <b>B: </b>12.18V    0ma  66F   0.00 Wh
<p> <b>C: </b>19.42V    0ma  64F   0.00 Wh
</body>
</html>
^D
?
```

3.3.17 Set INA register Values

Set one of the INA219 registers (for debugging).

set [**A** — **B** — **C**] *register-number value*

3.3.18 Send SVG graph to kserver

When the serial port is connected to the **kserver** program, send an SVG version of one of the screen graphs. As the **serve** command, this starts with a 0x01 (↑A) and ends with a 0x04 (↑D). The command must be followed by one of:

tempA The temperature of regulator A in Fahrenheit.

tempB The temperature of regulator B in Fahrenheit.

tempC The temperature of regulator C in Fahrenheit.

tempAUX The temperature monitored by the auxiliary sensor in Fahrenheit.

ampsA Graph current consumption of power A.

ampsB Graph current consumption of power B.

ampsC Graph current consumption of power C.

battery Graph battery voltage.

The time duration depends upon the settings of **rate** (3.3.13 on page 17) and **samples** (3.3.15 on page 17).

```
?svg ampsA

<svg version="1.1" viewBox="0 0 127 70" xmlns="http://www.w3.org/2000/s
vg">
<style>
.f0 { font: normal 0px sans-serif; }
.f1 { font: normal 1px sans-serif; }
.f2 { font: normal 2px sans-serif; }
.f3 { font: normal 3px sans-serif; }
.f4 { font: normal 4px sans-serif; }
.f5 { font: normal 5px sans-serif; }
.f6 { font: normal 6px sans-serif; }
.f7 { font: normal 7px sans-serif; }
.f8 { font: normal 8px sans-serif; }
.f9 { font: normal 9px sans-serif; }
.f10 { font: normal 10px sans-serif; }
</style>
<text x="0" y="7" fill="0" class="f10">KI7NNP Supply A (5v) mA</text>
<line x1="26" x2="127" y1="54" y2="54" stroke="#C0C0C0" /><line x1="26" x2="127"
 y1="38" y2="38" stroke="#C0C0C0" /><line x1="26" x2="127" y1="20" y2="20" strok
e="#C0C0C0" /><line x1="26" x2="26" y1="54" y2="20" stroke="#C0C0C0" /><line x1=
"127" x2="127" y1="54" y2="20" stroke="#C0C0C0" /><text x="0" y="55" fill="#0000
00" class="f6">0</text>
<text x="0" y="40" fill="#000000" class="f6">32</text>
<text x="0" y="22" fill="#000000" class="f6">64</text>
<line x1="27" y1="32" x2="28" y2="32" stroke="#FF0000"/>
.
.
.
```

3.3.19 Start recording

You can control recording from the push buttons at start up or by command on the serial port. There are 5 options to the **start** command and auto start are shown in Table 4.

Command option	Prev button	Next button	Records
none	↓	↓	Record everything.
amps	↓	↑	Record time and three current values.
volts	↑	↓	Record time, currents and voltages.
temps	X	X	Record currents and temperatures. Can't be selected at startup.
all	↓	↓	Record everything.
...	↑	↑	At startup, don't record anything. Don't use **start**.

Table 4: Recording types

Recording columns have uniform names but only those selected as above appear.

Heading	Type	Description
month	all	Month of the year, 1-12.
day	all	Day of the month, 1 - 31.
year	all	4 digits of the year.
hour	all	Hour of 24 hour day, 0-23.
minute	all	Minute of hour, 0-59.
maA	all	Milli-amps for power A.
maB	all	Milli-amps for power B.
maC	all	Milli-amps for power C.
...add for volts		
vA	volts	Output voltage for power A.
vB	volts	Output voltage for power B.
vC	volts	Output voltage for power C.
bat	volts	Supply battery voltage.
Time, amperage and ...		
tA	temps	Fahrenheit temperature power A.
tB	temps	Fahrenheit temperature power B.
tC	temps	Fahrenheit temperature power C.
tX	temps	Fahrenheit temperature battery.

Table 5: CSV file headings.

3.3.20 Stop recording

If you're recording, stop and write end of file. Just powering off will have the same effect as an end of file is written each time something is recorded.