



PIXIE BOARD PIXIEPOWER USER MANUAL

The information contained herein is believed to be accurate as of the date of this publication. AEL Microsystems Ltd assumes no liability for errors, or for any incidental, consequential, indirect, or special damages, including, without limitation, loss of use, loss or alteration of data, delays or lost profits or savings, arising from the use of this document, or use of any product, circuit or software described herein or the product which it accompanies.

This document may be reproduced or transmitted by means, electronic or mechanical, provided it is used for personal or educational use.

AEL Microsystems Ltd
Malvern
UK

Acknowledgements:

AEL Microsystems Ltd acknowledges the trademarks of other organisations for their respective products and services mentioned in this document.

This contact information is subject to change, for the latest details go to www.aelmicro.com

Web: <https://www.aelmicro.com>
Email: pixie.support@aelmicro.com

1. Contents

- 1. Contents 3
- 2. "PixiePower" application. 4
 - 2.1 Install the "PixiePower" application. 4
 - 2.2 Operation 4
 - 2.3 Options 5
 - 2.4 Daemon only options 7
 - 2.5 Useful applications 8
 - 2.5.1 Hardware GPIO driven power off 8
 - 2.5.2 Using the PWR On/Off input to start the shutdown 8
- 3. Notes 10

2. “PixiePower” application.

The “PixiePower” application can be used to easily change the settings of the PIXIE Power supply board and also provide a continuous power monitor allowing actions to be performed during changes in power levels etc.

2.1 Install the “PixiePower” application.

Using the Raspberry Pi web browser download the “PixiePower” application **pixie-power-x.y.tgz** found at **aelmicro.com > Support > Pixie Boards > Software > “PixiePower” Application** which will usually end up in the **/home/pi/Downloads** directory on the Raspberry Pi.

Open a command line window and enter the following:

```
sudo tar xzf ~/Downloads/pixie-power-x.y.tgz -C /usr/bin/
```

Test the installation by entering:

```
sudo PixiePower -?
```

this will result in the option help being displayed.

2.2 Operation

When the command is used without options the various voltage, current and other readings are displayed. The board identifier is always required as the 1st argument based on the board identifier switch setting on the power supply board.

Open a command line window and enter the following if the board identifier is set to 0:

```
sudo PixiePower 0
```

The PSU levels and statistics are displayed.

V1+: is the voltage at the input terminal V1+

V2+: is the voltage at the input terminal V2+

V+: is the voltage after the diode OR’ing of inputs V1+ and V2+, this may be 0.4V less due to the diode voltage drop.

+5V: is the voltage at the +5V terminals and power to pin 4 of the 40 pin Pi board.

+5I: is the current being draw from the 5V regulator by the Raspberry Pi and additional boards.

Temp: is the temperature of the power supply board in degrees C.

Status: is the boards current status bits.

Bit 0 = Power on/off input signal.

Bit 1 = T2P signal.

Bit 2 = CPU fan is on.

Bit 3 = Board cooling fan is on.

Bit 4 = Watchdog is in signalled time window.

PC: is the total number of power cycles performed due to the power on/off input or software control.

PCW: is the total number of power cycles performed due to a watchdog timeout.

PCOT: is the total number of power cycles performed due to a board over temperature shutdown.

To continually refresh the values until a key is pressed add the **-r** option:

```
sudo PixiePower 0 -r
```

2.3 Options

- fan=** This controls the CPU fan, **-fan=0** for off and **-fan=1** for on.
- i2c=** This changes the I2C device channel, the default is 1.
- i2cBase=** This changes the base I2C address, the default is 16.
- ipv1=** This sets the GPIO signal used to input the V1 fail signal to the Raspberry Pi.
Values between 5 and 27 are valid, use a minus value e.g. **-19** to invert the signal state.
- ippwr=** This sets the GPIO signal used to input the power on/off input signal to the Raspberry Pi.
Values between 5 and 27 are valid, use a minus value e.g. **-19** to invert the signal state.
- ipwf=** This sets the GPIO signal used to input the watchdog signalled input signal to the Raspberry Pi.
Values between 5 and 27 are valid, use a minus value e.g. **-19** to invert the signal state.
- opwr=** This sets the GPIO signal used to output a watchdog reset signal from the Raspberry Pi.
Values between 5 and 27 are valid, use a minus value e.g. **-19** to invert the signal state.
- opcf=** This sets the GPIO signal used to output a CPU fan control signal from the Raspberry Pi.
Values between 5 and 27 are valid, use a minus value e.g. **-19** to invert the signal state.
- oppof=** This sets the GPIO signal used to output a power off signal from the Raspberry Pi.
The signal is always active low.
- pwr=<off> <on>**
This option sets the time from issuing this option before the board turns the output supply off, followed by an optional on time from when the board then turns the output supply back on. The values of **<off>** and **<on>** are in seconds and are 32 bits in size so very long times can be set.
- Examples:
- pwr=5** This will turn the board off after 5 seconds and it remains off until the main input supply is cycled, or the power on/off input is actioned.
- pwr=5 10** This will turn the board off after 5 seconds, then after a further 10 seconds the output is turned back on.
- pwr=0 10** This will turn the board output on after 10 seconds if it has been turned off using a GPIO pin set by the **-oppof** option.
This would become useful if the PI had to shutdown completely then wished to be restarted sometime in the future, next day, week or month etc.
- r** Only used without any other option to refresh the current displayed readings until a key is pressed.
- save** Used to save the GPIO values set by **-ipv1**, **-ippwr**, **-ipwf**, **-opwr**, **-opcf**, **-oppof**, option to the board's EEPROM so they are preserved through a power off condition.

-v2=

This controls the V2 input switch and the V1>V2 power on detection.

If the V1>V2 feature is enabled, then the supply will turn on if the level of V1 becomes greater than V2. This feature is useful for battery backed conditions where the battery is connected to input V2 and the main supply on V1. When the main supply returns the power supply will turn back on if it had been turned off using software.

Bit 0 = V2 switch, 0=off, 1= on.

Bit 1 = V1>V2 turn on supply, 0= disabled, 1=enabled.

-wdr

This option resets the watchdog indicating the CPU is still in control.

As reset can also be performed using a GPIO pin, see the **-opwr** option.

-wdt=<tmo> <sig>

This option sets the watchdog timeout and signal values.

The values of **<tmo>** and **<sig>** are in seconds and are 8 bits in size.

The **<sig>** time is optional and provides an early warning via the status or GPIO signals of a pending watchdog fail, see **-ipwf**.

The **<tmo>** value is the timeout after which if a watchdog reset command or GPIO signal is not received the power output is cycled forcing a power on reset.

2.4 Daemon only options

The application can also be run as a daemon for monitoring functions.

All the previous options can be included on the command line and are actioned prior to the daemon running.

All the options below can be performed at the same time.

-d This is used to indicate it is to be run as a daemon.

-pwr=0/1 <cmd>

This option takes a value of 0 or 1 and is used to monitor the input power on/off for activity. When there is a change in state the **<cmd>** will be actioned.

Given a value = 0, the **<cmd>** will only be called once, if its = 1 then it will be called each time the power on/off is pressed.

Example:

```
sudo PixiePower-d -pwr=0 "sudo shutdown -h now"
```

This will instruct the OS to shutdown when the on/off input changes state.

-v2min=<level> <below> <above>

This option monitors the voltage **<level>** on **V2+**. When the voltage falls below the level the **<below>** command is actioned when it goes back above the level then the **<above>** command is actioned.

This can be used for battery backed applications to detect a point at say the power needs to be turned off because the battery level has become too low.

Example:

```
sudo PixiePower-d -v2min=9.6 "sudo shutdown -h now"
```

-wdm=0 <cmd>

This option takes a value of 0 which is for future use only and is not used at present, and is used to monitor status of the watchdog timer and pending timeout signal.

When there is a change in state of the watchdog to signalled then **<cmd>** will be actioned.

Example:

```
sudo PixiePower-d -wdt=60 30 -wdm=0 "sudo shutdown -h now"
```

This example activates the watchdog with a timeout of 60 seconds and will action the shutdown command after 30 seconds if no watchdog reset is issued.

2.5 Useful applications

These are some useful setup's for using the features of the power supply board, they can be set to run the PixiePower application during start up to setup the board feature, the easiest method is to edit the `/etc/rc.local` file and add the commands you wish to use.

Edit an `/etc/rc.local` file using:

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

put the commands in the file and always end the file with `exit 0`, the following example set the power off GPIO output pin and the power On/Off input pin

```
sudo PixiePower 0 -oppof=20 -ippwr=21
exit 0
```

2.5.1 Hardware GPIO driven power off

The cleanest way to turn the power off is to use a GPIO output pin which changes state at the same time as the Raspberry PI has entered its halt state, this method ensures that all the shutdown tasks are completed cleanly.

For this instance, we are using GPIO21 as the control output to turn off the power board output supply.

- 1) Edit the `/etc/rc.local` file and add in the following:

```
sudo PixiePower 0 -oppof=21
```
- 2) Edit the `/boot/config.txt` file using `sudo nano /boot/config.txt` and add:

```
dtoverlay=gpio-poweroff,gpiopin=21,active_low=1
```
- 3) Reboot the system

Whenever the system is shut down in the future the power supply will be turned off at the end of the shutdown sequence.

2.5.2 Using the PWR On/Off input to start the shutdown

This can be done in one of two ways, either by a GPIO pin or the use of the **PixiePower** daemon, both methods will issue a shutdown command to the operating system.

2.5.2.1 Using a GPIO pin

For this instance, we are using GPIO20 as the control input, whenever the **PWR on/off** input is closed, the shutdown should commence.

- 1) Edit the `/etc/rc.local` file and add in the following:

```
sudo PixiePower 0 -ippwr=-20
```

Note the use of `"-20"` to indicate a low output whenever the **PWR on/off** input is closed
- 2) Edit the `/boot/config.txt` file using `sudo nano /boot/config.txt` and add:

```
dtoverlay=gpio-poweroff,gpio_pin=20
```
- 3) Reboot the system

2.5.2.2 Using a PixiePower monitor daemon

For this instance, we are using a monitoring daemon waiting on the status of the power board for a change of state in the **PWR On/Off** input, when this occurs a command such as “shutdown -h now” can be executed.

- 1) Edit the `/etc/rc.local` file and add in the following:
`sudo PixiePower 0 -d -pwrmon=0 "shutdown -h now"`
- 2) Reboot the system

Using a combination of the power off and shutdown examples above a Raspberry Pi can be turned off using a push button connected to the **PWR On/Off** terminals.

3. Notes