

SUPER-WATCHDOG WITH BATTERY BACKUP

for RASPBERRY PI

USER'S GUIDE VERSION 6.0

SequentMicrosystems.com

GENERAL DESCRIPTION 2

FEATURES..... 3

WHAT IS IN YOUR KIT 4

QUICK START-UP GUIDE 4

BOARD LAYOUT 6

BLOCK DIAGRAM 6

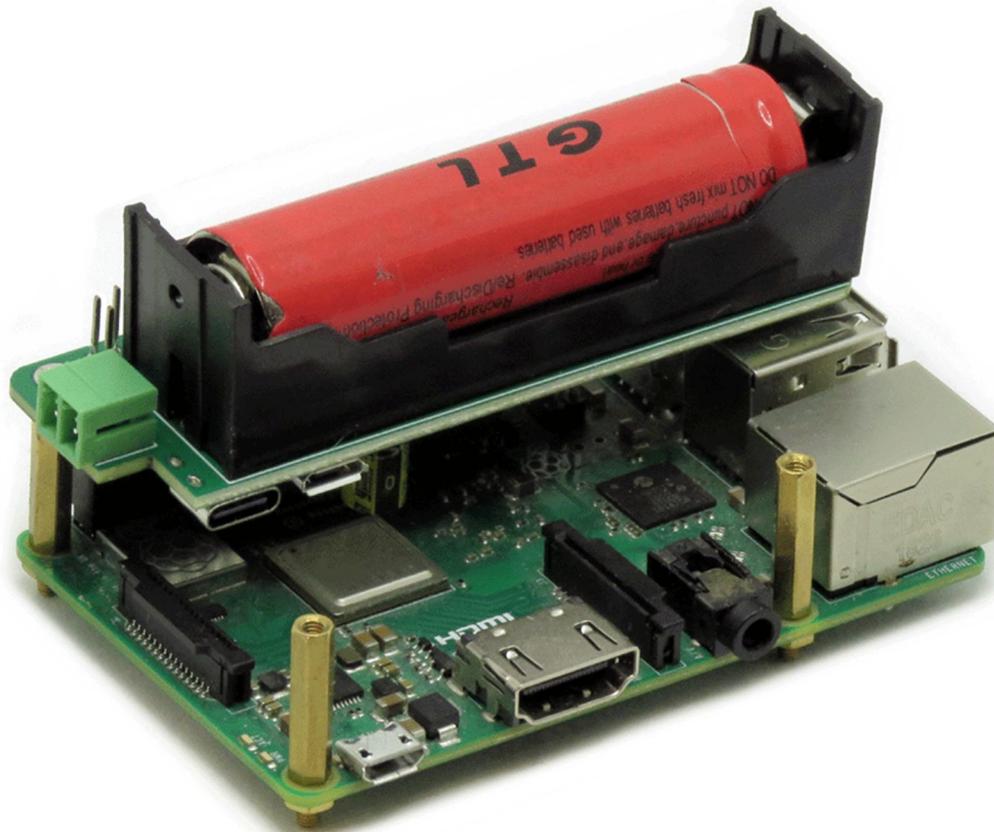
POWER REQUIREMENTS..... 7

MECHANICAL SPECIFICATIONS..... 8

SOFTWARE SETUP 8

PROGRAMING THE WATCHDOG 11

GENERAL DESCRIPTION



The Super-Watchdog for Raspberry Pi B+, 2, 3, 4 and Zero provides the safety required by mission critical projects, implementing two major functions. It power-cycle the Raspberry Pi in case of software lock-up, resetting not only the Pi but also any peripheral attached to it, and it prevents SD-Card failure in case of power loss, by allowing the Raspberry Pi to shut down itself safely. The 18650 Li-Ion battery can keep the Raspberry Pi running for hours, assuring continuous operation during power outages.

The watchdog has on board hardware to self-check all the major functions. Self-check is performed at each power up. The on-board LED is flashed rapidly (5 times per second) if a hardware error is detected, or slowly (1 time per second) to indicate normal functionality. The LED is turned on if the watchdog resets the Raspberry Pi, and off is the Raspberry Pi is powered down for a specific time interval.

The card has a default timeout of 120 seconds. Once installed, if it does not receive a reset from Raspberry Pi within 2 minutes, it cuts the power and restores it after 10 seconds.

Raspberry Pi needs to issue a reset command on the I2C port before the timer on the watchdog expires. The timer period after power up and the active timer period can be set from the command line. The number of resets is stored in flash and can be accessed or cleared from the command line.

When running on battery power, the watchdog processor monitors the battery voltage. To prevent permanent damage to the battery due to over-discharging, the processor will cut off the power to

Raspberry Pi when the battery drops below 2.8V. A software command is available to the user to also monitor the battery voltage. It is strongly recommended that Raspberry Pi performs a self-shutdown then the battery voltage drops to 3.0V.

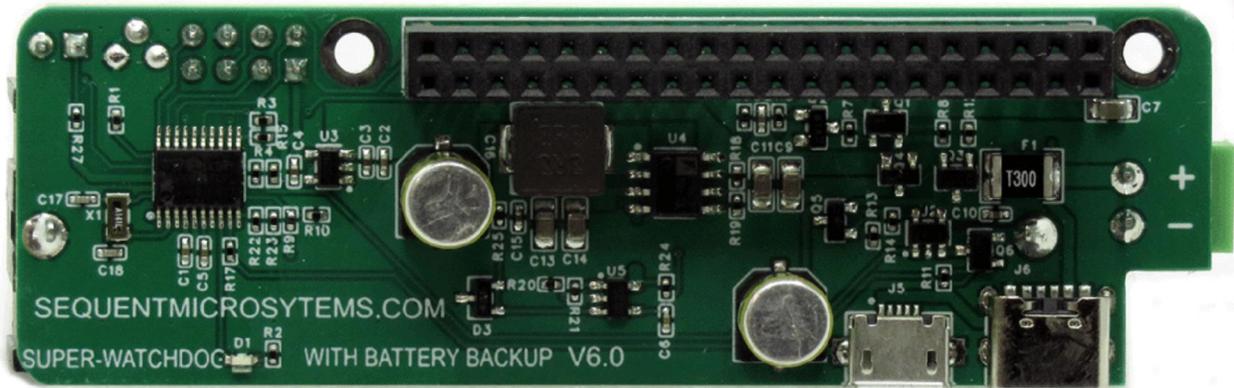
Command line software and Python drivers can be downloaded from [Github.com](https://github.com)

FEATURES

- A must for all RPI mission critical installation
- Cycle RPI power after software lock-up
- Prevent SD Card damage during power outages
- Run for hours on Li-Ion 18650 battery
- On board battery charger, step-up converter and intelligent power monitor
- Triple power supply jacks: micro USB, USB-C and 3.5mm Pluggable Connector
- Provide up to 3A/5V power to Raspberry Pi from Li-Ion Battery
- On board resettable fuse
- On board reset pushbutton
- All mounting hardware included: brass stand-offs, screws and nuts
- Command line, Node-RED, Python drivers

WHAT IS IN YOUR KIT

1. SUPER-WATCHDOG UPS HAT for Raspberry Pi

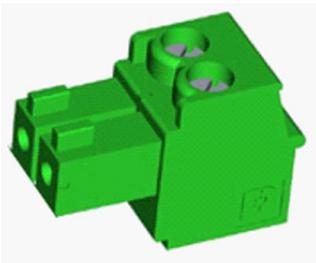


2. Mounting hardware



- a. Two M2.5x18mm male-female brass standoffs
- b. Two M2.5x5mm brass screws
- c. Two M2.5 brass nuts

3. 2-Pin plug for the power supply connector



QUICK START-UP GUIDE

1. Plug your SUPER-WATCHDOG card on top of your Raspberry Pi and power up the system from the watchdog.
2. Enable I2C communication on Raspberry Pi using raspi-config.
3. 3. Install the wdt software from github.com:

```
~$ git clone https://github.com/SequentMicrosystems/wdt-rpi.git
```

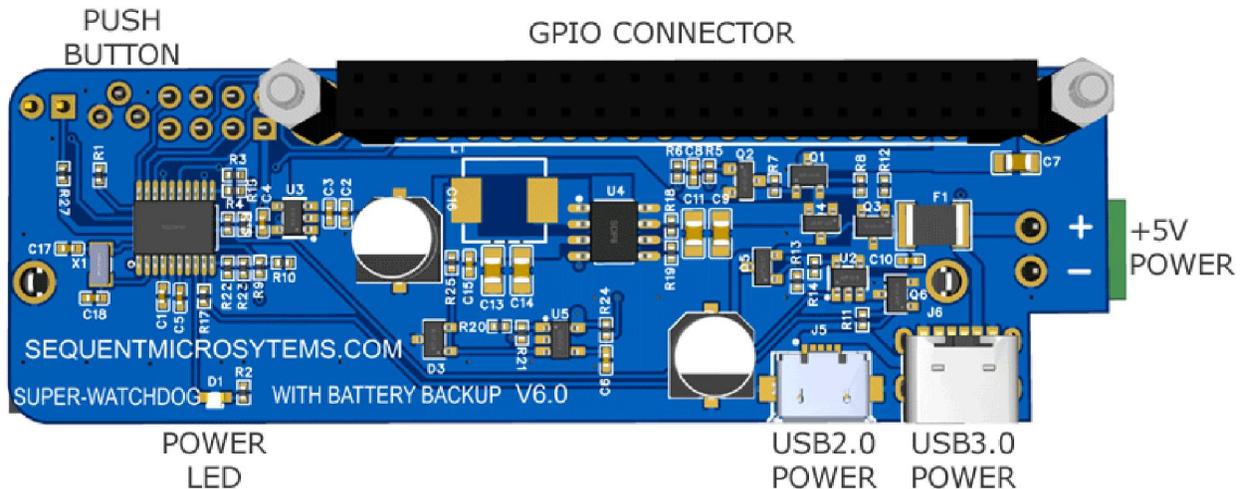
```
~$ cd /home/pi/wdt-rpi
```

```
~/wdt-rpi$ sudo make install
```

```
~/wdt-rpi$ wdt
```

The program will respond with a list of available commands.

BOARD LAYOUT



The Super-watchdog comes with appropriate mounting hardware. A power LED flashes when power is applied to the board. The top of the card is occupied by the 18650 battery holder (battery not included). All surface mount components are installed on the bottom.

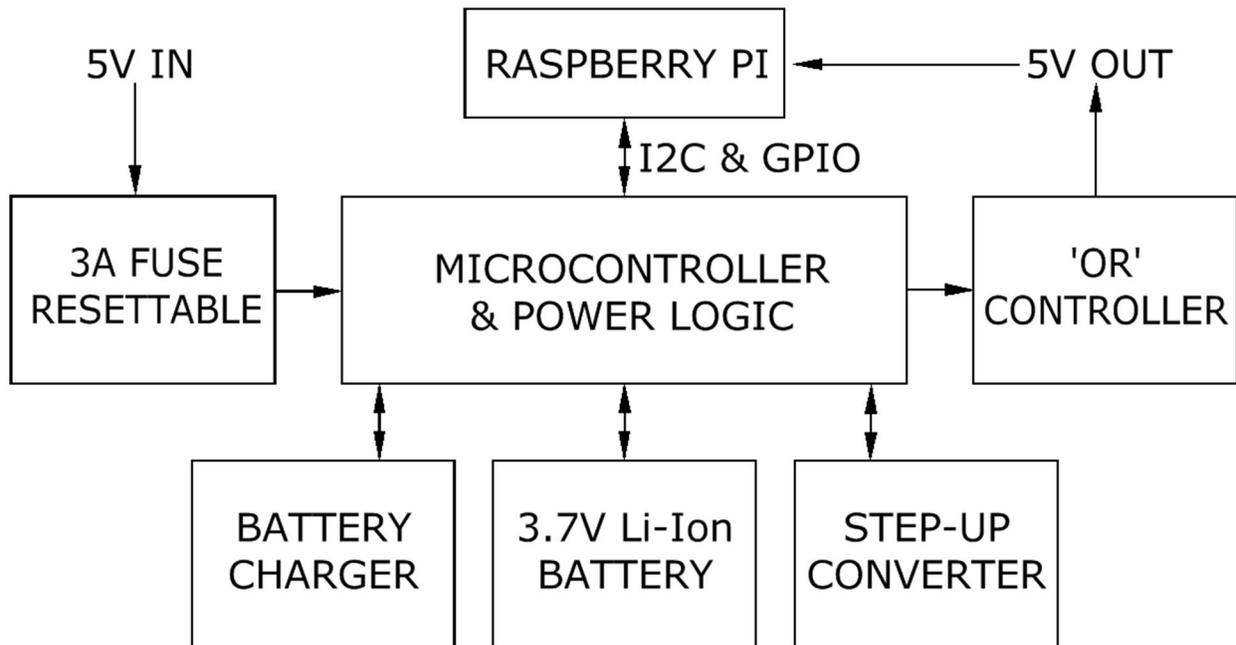
For your convenience, power can be applied to the board on any of the 3 power jacks provided. A pluggable 2-pin 3.5mm connector, a USB 2.0 if you use Raspberry Pi Zero or 3, or a USB 3.0 if you use Raspberry Pi 4.

On the upper left corner of the board, as shown in the above picture, is installed a pushbutton. The button is read by the processor, which can transfer the button state to pin 37 of Raspberry Pi.

You can write your own handler to perform any task desired on your Raspberry Pi. For example, you can shut down Raspberry Pi when the button is pressed twice.

A 2-pin header parallel to the button can be used to actuate the button from another location, when Raspberry Pi is installed in an enclosure and the button is not accessible.

BLOCK DIAGRAM



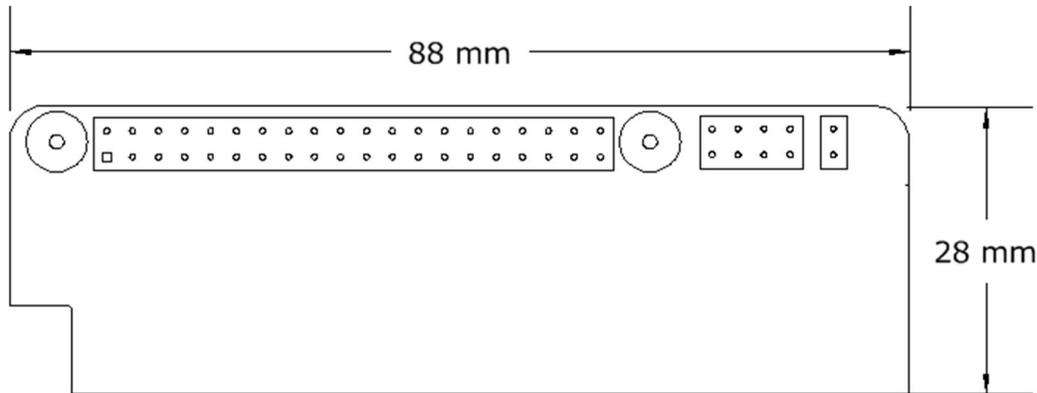
POWER REQUIREMENTS

In order for the watchdog to cycle the power to Raspberry Pi, +5V power needs to be supplied only to the watchdog. The watchdog has three power connectors: USB 2.0 and USB 3.0 connectors on the bottom and 2.1mm barrel connector on top. The USB 2.0 connector can supply up to 1.5A, USB 3.0 up to 3A while the barrel connector up to 5A. If your Raspberry Pi system requires more than 1.5A, do not power your system from the USB 2.0

If the 5V power supply fails, the watchdog switches the power supply to battery. When running on battery, the current supplied to Raspberry Pi is limited to 2A. Actual current drawn from battery will be about twice the current supplied to Raspberry Pi.

Watchdog current consumption:	10 mA @ +5V
Maximum power to Raspberry Pi:	3A
Battery charging current:	400mA
Raspberry Pi current supplied from battery:	2.0A (up to 10 seconds)
Raspberry Pi current until battery discharge:	1.5A

MECHANICAL SPECIFICATIONS



SOFTWARE SETUP

The watchdog board occupies the I2C address 0x30.

1. Have your Raspberry Pi ready with the [latest OS](#).
2. Enable I2C communication:

```
~$ sudo raspi-config
```

1. Change User Password	Change password for default user	
2. Network Options	Configure network settings	
3. Boot Options	Configure options for start-up	
4. Localisation Options	Set up language and regional settings to match..	
5. Interfacing Options	Configure connections to peripherals	
6. Overclock	Configure overclocking for your Pi	
7. Advanced Options	Configure advanced settings	
8. Update	Update this tool to the latest version	
9. About raspi-config	Information about this configuration	
P1	Camera	Enable/Disable connection to the Raspberry Pi Camera
P2	SSH	Enable/Disable remote command line access to your Pi
P3	VNC	Enable/Disable graphical remote access to your Pi using...
P4	SPI	Enable/Disable automatic loading of SPI kernel module
P5	I2C	Enable/Disable automatic loading of I2C kernel module
P6	Serial	Enable/Disable shell and kernel messages to the serial port
P7	1-Wire	Enable/Disable one-wire interface
P8	Remote GPIO	Enable/Disable remote access to GPIO pins

3. Install the wdt software from github.com:

```
~$ git clone https://github.com/SequentMicrosystems/wdt-rpi.git
```

```
~$ cd /home/pi/wdt-rpi
```

```
~/wdt-rpi$ sudo make install
```

```
~/wdt-rpi$ wdt
```

The program will respond with a list of available commands.

Type "`wdt -h`" for online help.

After installing the software, you can update it to the latest version with the commands:

```
~$ cd /home/pi/wdt-rpi
```

```
~/wdt-rpi$ git pull
```

```
~/wdt-rpi$ sudo make install
```

After installing the software, you can address the watchdog with the command "wdt". The watchdog will respond with a list of available commands. Any command send to the watchdog will reload the watchdog timer.

`wdt -r[eload]`: reload watchdog timer, prevent Raspberry power cycle.

`wdt -d[efault]`: set the default watchdog timeout (seconds). The watchdog loads this value after a power cycle. This value needs to be long enough for the boot process to complete.

`wdt -p[eriod]`: set the current watchdog period (seconds). Raspberry must address the watchdog faster than this value to prevent a power cycle. The period will be overwritten after a Raspberry power cycle with the default value.

`wdt -c[lear]`: clear the number of power cycles performed by the watchdog.

`wdt -off[interval]`: set the time the watchdog will keep the Raspberry power off (seconds). The default value is 10 seconds. After each power cycle, the off interval is reset to 10 seconds. Set this parameter to a large value to power down Raspberry Pi for a specific time interval.

`wdt -g[et] d[efault]`: get watchdog default parameter.

`wdt -g[et] p[eriod]`: get the watchdog period parameter.

`wdt -g[et] r[esets]`: get the number of power cycles performed by the watchdog.

`wdt -g[et] v[in]`: get the input voltage (mV).

`wdt -g[et] off[interval]`: get the current off interval (seconds).

wdt -g[et] vr[asp]: get the voltage applied to Raspberry Pi (mV).
wdt -g[et] vb[at]: get the battery voltage (mV).
wdt -g[et] c[harge]: read the battery charging status (0=off, 1=charged, 2=charging, 3=fault).
wdt -g[et] t[emp]: get the watchdog processor temperature.

PROGRAMING THE WATCHDOG

After the first power up, the watchdog cuts off the power to Raspberry Pi, performs a self-check and tries to turn on the power. The watchdog monitors the input power, the battery power and the power supplied to Raspberry Pi. If a hardware error is detected, the watchdog flashes rapidly (5 times per second) the on-board LED. If no error is detected, the LED flashes 1 time per second and the watchdog enters the stand-by state, waiting to be activated.

The watchdog is activated when is addressed first time on the I2C interface. The default timeout is set to 120 sec. If the timeout expires, the watchdog will turn off the Raspberry Pi power for 10 seconds, then turn it on again.

To disable the watchdog while setting it up, set the period to zero:

```
wdt -p 0
```

The watchdog can be reset either by writing a reset command from the command line ("wdt -r"), or by toggling pin 23 (GPIO 11) on the Raspberry Pi. The commands can be issued from a memory resident script or from cron. The script can have a resolution of 1 second, while cron can have a resolution of 1 minute.

1. RESET THE WATCHDOG FROM A SCRIPT USING THE COMMAND LINE

a. Use your favorite editor to create a script called `wdt_reload` with the following content:

```
#!/bin/sh
while :
do
    wdt -r
    sleep 60
done
```

b. Make the script executable:

```
~/wdt-rpi$ chmod +x wdt_reload
```

c. Launch the script memory resident:

```
~/wdt-rpi$ wdt_reload&
```

d. Set the default watchdog timer period to your desired value. In this example we use 120 sec. The timer period needs to be longer than the "sleep" command in the `wdt_reload` file.

```
~/wdt-rpi$ wdt -p 120
```

2. RESET THE WATCHDOG FROM A SCRIPT USING THE GPIO INTERFACE

Same as above, with the following `wdt_reload` file content:

```
#!/bin/sh
gpio -g mode 11 out
```

```
while :
do
  gpio -g toggle 11
  sleep 60
done
```

3. RESET THE WATCHDOG FROM CRON

Create the wdt_reload with the following content:

```
#!/bin/sh
/home/pi/wdt-rpi/wdt -r
```

Make it executable, then edit the crontab file:

```
~/wdt-rpi$ crontab -e
```

Add the following line at the end:

```
* * * * * /home/pi/wdt-rpi/wdt_reload
```

Save the crontab file. To check that the cron is running, add at the end of the wdt_reload the command

```
echo `date` >> /home/pi/wdt-rpi/logfile
```

After 1 minute you should see the logfile with the timestamp when it was called by cron:

```
~/wdt-rpi$ cat logfile
Fri Jun 28 18:04:02 PDT 2525
```

Remove the timestamp from the wdt_reload file. Set the watchdog period to anything shorter than 60 seconds, and wait for the watchdog to cycle the power on Raspberry Pi. Ten seconds before reboot the watchdog will accelerate flashing it's LED. During the 10 seconds reboot, the LED will stay on.

4. SCHEDULED POWER DOWN

To schedule a Raspberry Pi power down for a specific time T(seconds), do the following:

a. Set the watchdog off interval:

```
wdt -off T
```

b. Set the watchdog period long enough for Raspberry Pi to shut down (10 sec recommended).

```
wdt -p 10
```

c. Shutdown Raspberry Pi:

```
sudo shutdown now
```