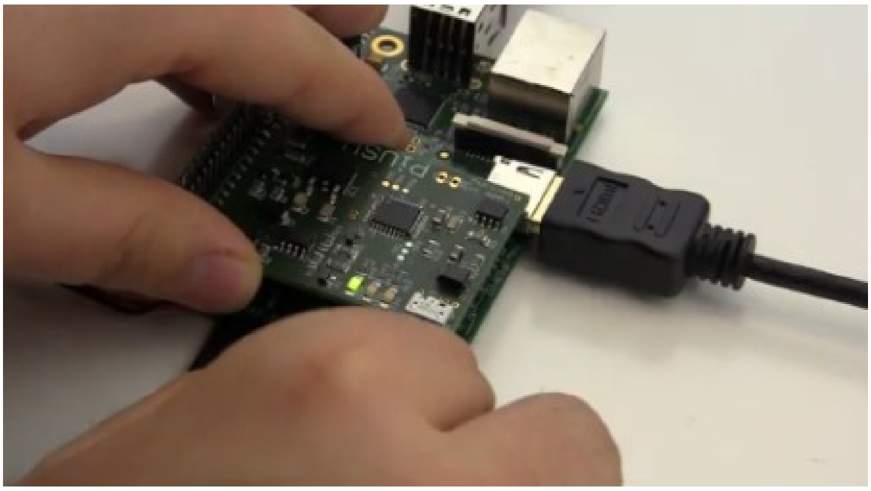

Title:

Documentation for the Pi-UPS-Monitor-App

Version:
0.01

Date:
09.01.14

Table of Content

1. PURPOSE, OR WHAT PIUSVMONITOR DOES

The **piusvmonitor software** monitors:

1. changes of the power supply of the CW2. Pi UPS device
2. writes them to a logfile (default location: /usr/share/piusvmonitor/log.piusv) and
3. also reacts to changes in the power supply.

You are free to configure *piusvmonitor*'s behavior.

2. PACKAGE CONTENTS

- *piusvmonitor* program
- wiringPi library (licensed under GNU LESSER GENERAL PUBLIC LICENSE)

3. INSTALLATION

1. Download the package.
2. If you have the piusvmonitor.deb already on your system, change to *piusvmonitor*'s location with: `cd /location/of/`
3. then install with: `dpkg --install piusvmonitor.deb`

4. USAGE

- Use the console commands (see below, chapter 5) to start and stop *piusvmonitor* software
- Edit the config.piusv file to configure *piusvmonitor*'s behavior

5. CONSOLE COMMANDS

Hint: The commands can be used regardless of the directory or the user.

The console command "`piusvd start`" will start the *piusvmonitor* daemon "piusvd", if it is not already running.

If "piusvd" is already running you will receive a message and nothing else will be done.

The console command "`piusvd stop`" will stop the *piusvmonitor* daemon "piusvd" if it is running.

If "piusvd" was not running, nothing will be done.

The console command "`piusvd restart`" will execute a "piusvd stop" and then a "piusvd start" command afterwards.

6. CONFIGURATION

Hint: Do not change the config.piusv's location!

The config.piusv offers the following configuration options:

`Autostart`: default: 1

Effect: Decides if "piusvd" is started on boot.

Any value other then "0" will start the piusvdaemon on boot.

`Logfile`: default: /usr/share/piusvmonitor/log.piusv

Effect: specifies the location of the logfile, where changes in the power supply are logged.

Enter the absolute path of the logfile's location. If the path is invalid no data will be saved.

`TurnOffTime`: default: 10 (seconds)

Effect: defines the time interval, in seconds, that has to pass before the shutdown script is started, the Raspberry Pi and the CW2. Pi UPS are shut down.

Details:

When PRI_POW is low and SEC_POW is high, the countdown is started.

If the countdown is greater then 0 and PRI_POW is high again the timer is reset to its defined value, and Raspberry Pi and CW2. Pi UPS will keep running.

If the countdown is equal or less 0, shutdown of Raspberry Pi and CW2. Pi UPS is inevitable, even if PRI_POW gets high again.

The shutdown script (details below) and a 20 seconds countdown to shut CW2. Pi UPS down are started.

When the shutdown script is executed, Raspberry Pi will be shut down.

When the 20 seconds have passed CW2. Pi UPS will be shutdown.

Example:

Lets assume "TurnOffTime" is set to a value of 7.

First incident: PRI_POW gets low, and gets high again after 5 seconds: CW2. Pi UPS keeps running, nothing significant happens.

Second incident: PRI_POW gets low, and gets high again after 9 seconds:

After 7 seconds (the value set for TurnOffTime), the "ShutDownScript" will be executed and the 20 seconds countdown to shut down Raspberry Pi and CW2. Pi UPS.

The shutdown of Raspberry Pi and CW2. Pi UPS is inevitable. The fact that PRI_POW gets high after 9 seconds does not stop the shutdown.

`Sleep`: default: 1 (second)

> Effect: specifies the time interval when *piusvmonitor* checks the CW2. Pi UPS power supply state.
>
> Example: If "Sleep" is set to 0.5, *piusvmonitor* will check CW2. Pi UPS power state every 0.5 seconds (2 times a second).

For each signal (PRI_POW, SEC_POW, BAT_LOW) a shell script can be passed to *piusvmonitor.*

Hint: Use the absolute path always!

The script will be executed anytime the signal changes from low to high.

PRI_POW: Default:

> Effect: *piusvmonitor* will execute the PRI_POW-shell script anytime PRI_POW changes from low to high

SEC_POW: Default:

> Effect: *piusvmonitor* will execute the SEC_POW-shell script anytime SEC_POW changes from low to high

BAT_LOW: Default:

> Effect: *piusvmonitor* will execute the BAT_LOW-shell script anytime BAT_LOW changes from low to high

ShutDownScript: default:

> Effect: This script will be executed if PRI_POW was low for "TurnOffTime" seconds continuous, and the countdown to (inevitable) shut down the Raspberry Pi and CW2. Pi UPS has started

7. LOGFILE(ENTRYS)

As mentioned before you can configure the logfile's location inside the config.piusv.

A logfile entry contains a time stamp and a signal info. *piusvmonitor* only records signal changes.

The power supply state signals are:

- PRI_POW: Primary Power Supply
  Indicates that CW2. Pi UPS supplies the Raspberry Pi via micro USB, which is usually connected to the grid.

- SEC_POW: Secondary Power Supply
  Indicates that CW2. Pi UPS supplies the Raspberry Pi via the battery pack connected to the CW2. Pi UPS.

- BAT_LOW: Battery LOW
  Indicates that the battery pack connected to the CW2. Pi UPS is either low or not (correctly) connected.
  If the Battery pack is (correctly) connected, change or recharge the batteries.

Furthermore there is the shutdown entry which indicates that *piusvmonitor* has entered the countdown to its inevitable shutdown.

8. SOFTWARE STRUCTURE

- *piusvmonitor*: Monitors the power supply of the CW2. Pi UPS device
- piusvautostart: This script decides if *piusvmonitor* is started on boot or not.
- piusvd: provides the piusv-daemon.
- wiringPi library is used

9. KNOWN BUGS

None so far!

Some warnings which should not bother you.

“Startparbug” may occur if *piusvmonitor* is started on boot.