



# **Benutzerhandbuch S.USV pi solutions**

Kompatibel zu S.USV pi basic und S.USV pi advanced

Revision 1.4 | Datum 19.04.2016

# Inhaltsverzeichnis

<b>1 Funktionen</b>	3
<b>2 Technische Spezifikationen</b>	4
2.1 Übersicht	5
2.2 Ausführung	5
2.3 LED - Statusanzeige	6
<b>3 Installationsanleitung</b>	7
<b>3.1 Hardware</b>	7
3.1.1 Inbetriebnahme S.USV	7
3.1.2 Anschluss des Akkus	8
3.1.3 Anschluss der externen Stromversorgung	8
3.1.4 Verwendung der Taster	9
3.1.5 GPIO – Port	9
<b>3.2 Software</b>	10
3.2.1 Raspbian	10
3.2.2 I <sup>2</sup> C	10
3.2.3 S.USV	13
3.2.3.1 Register	14
3.2.4 RTC – Real Time Clock	15
3.2.4.1 Zeitgesteuertes Ein- und Ausschalten	17
<b>4 Client Software</b>	18
<b>4.1 S.USV – Daemon</b>	18
4.1.1 Daemon Konfiguration	18
4.1.2 Daemon Steuerung	20
4.1.3 User Shutdown Script	20
<b>4.2 S.USV – Client</b>	21
4.2.1 Client Optionen	21

# 1 Funktionen

---

Die S.USVpi solutions ist ein erweitertes Energieversorgungs-Zusatzmodul für Raspberry Pi, mit dem Hauptfokus auf die unterbrechungsfreie Stromversorgung des Einplatinencomputers. Nebenbei liefert das Modul noch weitere Zusatzfunktionen, um die Bedienung des Raspberry Pi durch den Benutzer zu optimieren.

Das Modul ist eine voll funktionsfähige Plug & Play Lösung. Die Stromversorgung der S.USVpi solutions erfolgt direkt über den J8-Anschluss des Raspberry Pi und nutzt somit eine gemeinsame Spannungsquelle, dadurch sind keine weiteren Verkabelungen oder Stromversorgungen notwendig. Zusätzlich ist das Modul mit einem LiPo-Akku ausgestattet, der über einen Boost-Schaltleistungswandler den notwendigen Spannungsbereich abdeckt, um die Raspberry Pi bei Fehlverhalten sicher herunterzufahren und Datenverluste zu verhindern. Die „Advanced“ Variante bietet außerdem einen Netzeingang für den erweiterten Spannungsbereich von 7 -24 Volt ( Solarzellen, KFZ – Bereich, etc. ).

- HAT konformes USV Modul
- Kompatibel mit Pi 3 and Pi 2 Model B, Pi A+, Pi B+
- Adapterlösung für Pi Model A und B
- Unterbrechungsfreie Stromversorgung
- Plug & Play ( Stromversorgung über Raspberry Pi )
- Monitoring – Überwachungssystem ( V/mA )
- Intelligente Softwarelösung inklusive Mobiler Applikation
- Integrierter LiPo-Akku ( 300mAh ) mit intelligenter automatischer Ladefunktion
- Battery Management Controller ( Überlast-, Tiefentlade- und Überhitzungsschutz )
- Battery Monitoring System
- Netzeingang mit erweitertem Spannungsbereich von 7-24 Volt
- Software simulierte Real Time Clock mit Battery Back-Up
- Zeitgesteuertes Ein- und Ausschalten des Raspberry Pi
- Raspberry Pi Supply Switch ( Ein- und Ausschalttaster / File Safe Shutdown )
- LED – Statusanzeige
- Bootloader für Live – Firmware Updates

## 2 Technische Spezifikationen

	<i>S.USV pi basic</i>	<i>S.USV pi advanced</i>
Plug & Play	✓	✓
Stromversorgung	5 Volt/2500 mA	5 Volt/2500 mA
Erweiterter Spannungsbereich	x	7-24 Volt/2500 mA
Schnittstellen	I <sup>2</sup> C	I <sup>2</sup> C
ID EEPROM	✓	✓
Monitoring - System	✓	✓
Mobile Applikation	✓	✓
LiPo - Akku	✓	✓
Battery Management	✓	✓
Battery Monitoring	✓	✓
Real Time Clock	✓	✓
Raspberry Pi Supply Switch	✓	✓
Kompatibilität	Pi 3, Pi 2, Pi A+, Pi B+	Pi 3, Pi 2, Pi A+, Pi B+
Abmessungen	65x56,5 mm	65x56,5 mm

	<b>300 mAh</b>	<b>3000 mAh</b>
Normal Voltage	3.7 V	3.7 V
Working Voltage	3.0 - 4.2 V	3.0 - 4.2 V
Capacity	300mAh	3000mAh
Internal Impedance	≤60mΩ	≤30mΩ
Constant Charge   Discharge Current	2C   15C	1C   2C
Working Temperature	charge: 0-45°C; discharge: -20-60°C	charge: 0-45°C; discharge: -20-60°C
Connector	JST-XH-2P	JST-XH-2P
Leadwire	UL1571#28 / 50mm	UL1571#28 / 50mm
Wrapping Way	blue PVC	blue PVC
Line Drawn Out Direction	Center	Center
Dimensions	30.0 x 20.0 x 6.7 mm	60.0 x 50.0 x 9.0 mm

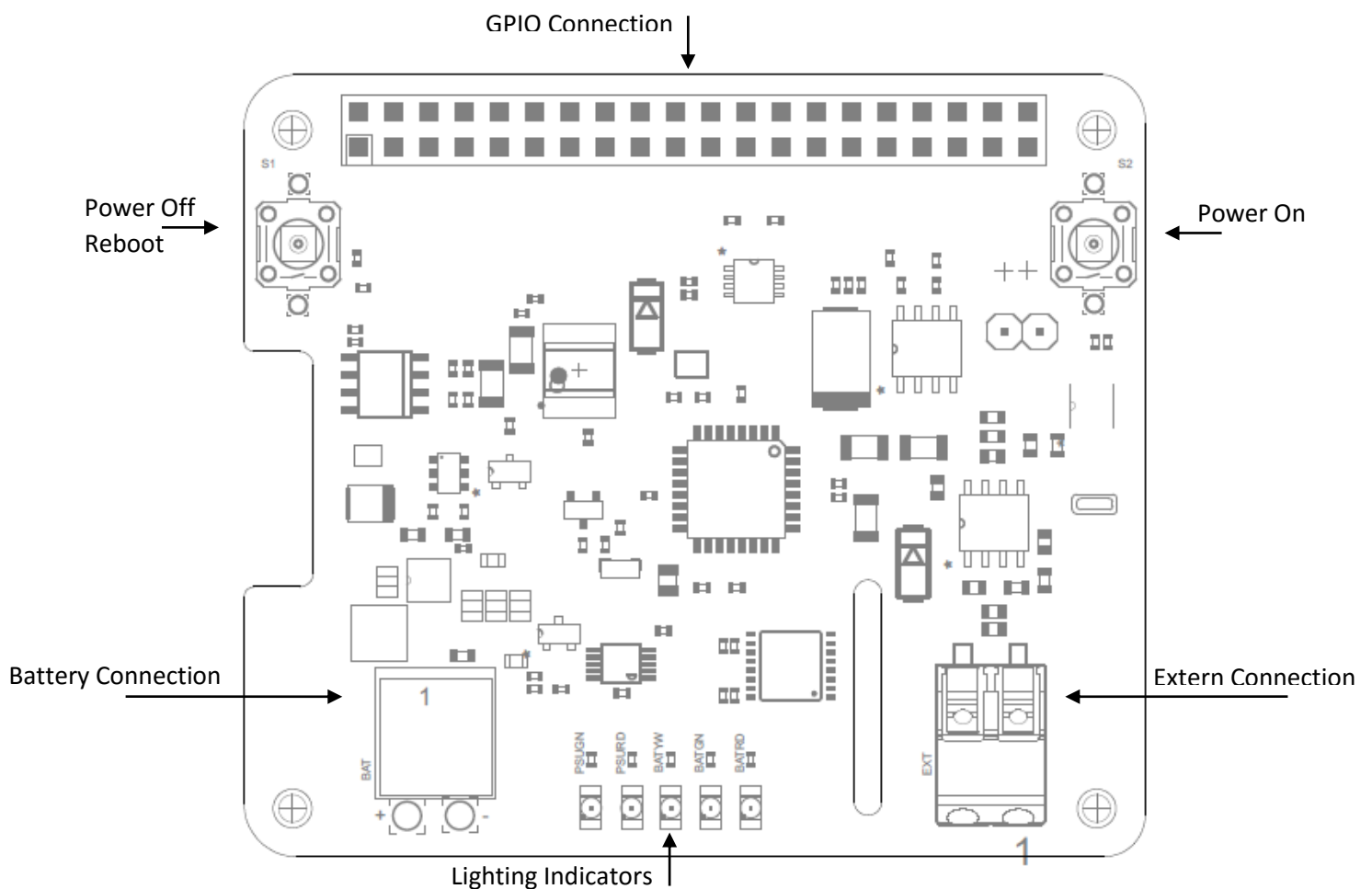
### Überbrückungszeiten bei Stromausfall (Batteriebetrieb):

Die folgende Tabelle soll einen Einblick in die Gesamtleistung bzgl. der Überbrückungszeit der S.USV bei Stromausfall geben, gemessen wurde bei einer Akku-Restkapazität von 70-80%.

	Stromverbrauch +/-	300 mAh	3000 mAh
Raspberry Pi A+	500mA	45 Minuten	11 Stunden
Raspberry Pi B+	500mA	30 Minuten	8 Stunden
Raspberry Pi 2B	500mA	30 Minuten	8 Stunden

(Hinweis: Die gemessenen Überbrückungszeiten in Bezug auf den gemessenen Stromverbrauch sind keine Garantieangaben und können je nach Peripherie abweichen.)

## 2.1 Übersicht



## 2.2 Ausführung

- **Battery Connection:** Verbindungsbuchse für den Anschluss des mitgelieferten LiPo-Akkus.  
**Battery Connector:** (Würth Elektronik 620 002 113 322)
- **Extern Connection:** Verbindungsbuchse für den Anschluss des erweiterten Spannungsbereichs (7-24V).
- **GPIO PORT:** GPIO Connection zur Raspberry Pi.
- **Power Off / Reboot:** Drücken Sie den Taster für einen Neustart oder halten Sie diesen mindestens 3 Sekunden lang zum Herunterfahren.
- **Power On:** Drücken Sie den Taster zum Hochfahren des Raspberry Pi.
- **Lighting Indicators:** LED-Statusanzeige für die S.USV / Raspberry Pi.

## 2.3 LED - Statusanzeige

LED	Anzeige
PSU <b>GREEN</b> (Blinkend)	Startup – Initialisierung der S.USV Firmware
PSU <b>GREEN</b>	RPi Power Supply Unit ist online (Voltage present)
PSU <b>RED</b>	RPi Power Supply Unit ist offline (Voltage loss) – Battery Powering ist online
BAT <b>YELLOW</b>	Charging Circuit Online – Akku wird geladen
BAT <b>GREEN</b>	Charging Circuit Online – Akku ist vollständig geladen
BAT <b>RED</b>	Charging Circuit Offline – Akku nicht verbunden oder fehlerhaft
BAT <b>RED</b> (Blinking)	Charging Circuit Offline – Akkurestkapazität im kritischen Bereich

## 3 Installationsanleitung

---

### 3.1 Hardware

Um über die vollständige Funktionalität der S.USV zu verfügen, empfehlen wir den Akku im ersten Schritt vollständig zu laden.

Darüber hinaus empfehlen wir ein Netzteil mit mindestens 2 Ampere, um den Raspberry Pi zu betreiben.

In den folgenden Schritten wird Ihnen die Inbetriebnahme der S.USV nochmals im Detail beschrieben:

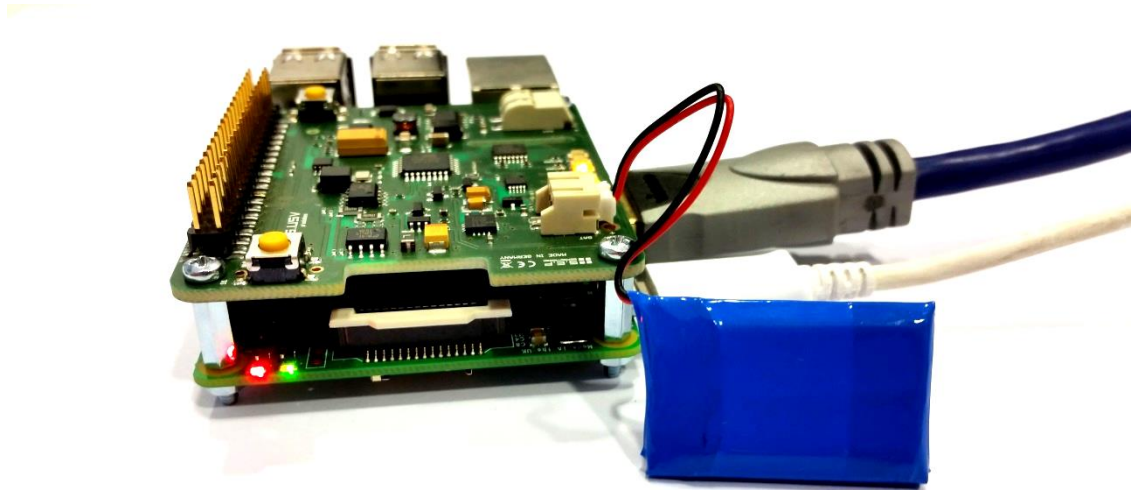
#### 3.1.1 Inbetriebnahme S.USV

- Die primäre Stromversorgung der S.USV erfolgt über den GPIO Port – Pin 2 (+5V) des Raspberry Pi. Bitte stecken Sie nun die Platine wie dargestellt auf den RPi, um die benötigte Verbindung herzustellen und befestigen Sie diese mit dem beigegebenen Mounting Kit.



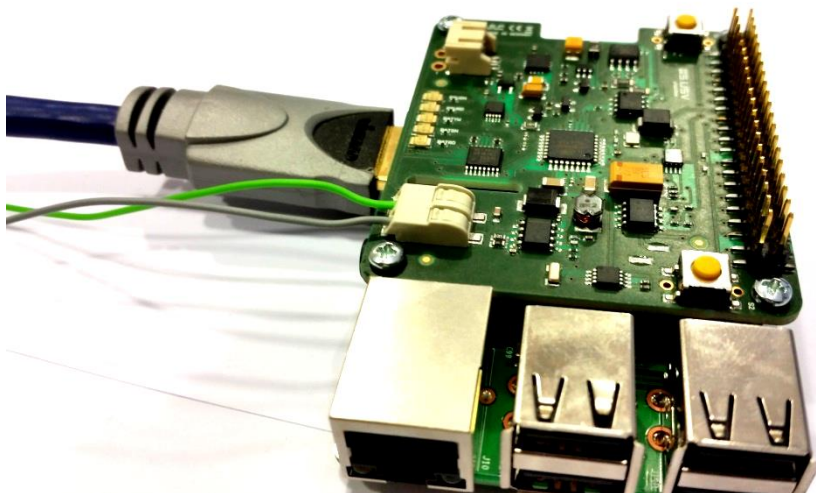
### 3.1.2 Anschluss des Akkus

- Bitte stellen Sie für die Inbetriebnahme sicher, dass der mitgelieferte LiPo - Akku in die dafür vorgesehene JST-Buchse auf der Vorderseite der Platine eingesteckt ist.



### 3.1.3 Anschluss der externen Stromversorgung

- Für die Verwendung des Wide-Range Eingangs befindet sich die dargestellte Klemmbuchse auf der Vorderseite der Platine. Bitte beachten Sie hierbei die +/- Kennzeichnung auf der Platine, um einen Kurzschluss zu vermeiden.

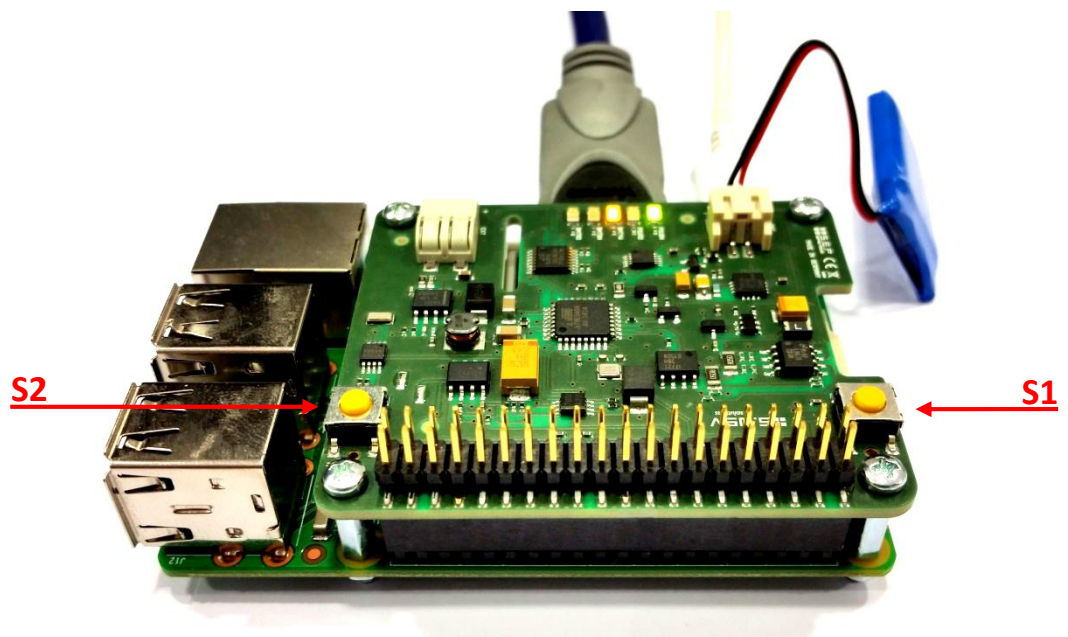




### 3.1.4 Verwendung der Taster

- Für das Ein- und Ausschalten, sowie das Neustarten des Raspberry Pi, stehen Ihnen folgende Taster zur Verfügung:
  - **S1** : Power Off / Reboot
  - **S2** : Power On

Beide Taster verfügen über entsprechende Durchkontaktierungen, somit können diese an jede beliebige Stelle Ihres Gehäuses geführt werden.



### 3.1.5 GPIO – Port

- Für die Stromversorgung, sowie die Datenübertragung der S.USV stehen folgende GPIO – Pins in Verwendung:
  - **Pin #02:** DC Power +5V - Stromversorgung
  - **Pin #03:** GPIO 02 (SDA1, I<sup>2</sup>C) – I<sup>2</sup>C Datenleitung
  - **Pin #05:** GPIO 03 (SCL1, I<sup>2</sup>C) – I<sup>2</sup>C Taktleitung
  - **Pin #13:** GPIO 27 (GPIO\_GEN2) – Zustandsüberwachung S.USV
  - **Pin #27:** ID\_SD (I<sup>2</sup>C ID EEPROM) – ID Datenleitung
  - **Pin #28:** ID\_SC (I<sup>2</sup>C ID EEPROM) – ID Taktleitung

## 3.2 Software

(Hinweis: Die aktuelle Software wurde speziell für Raspbian optimiert. In den nächsten Wochen werden weitere Ausführungen für alle gängigen Betriebssysteme verfügbar sein.)

### 3.2.1 Raspbian

- Für die Installation und Konfiguration der Raspberry Pi, empfehlen wir folgende Kurzanleitung:

<https://www.raspberrypi.org/help/quick-start-guide/>

- Für die Installation des Betriebssystems empfehlen wir die Image Installation Guides von Raspberry Pi:

<https://www.raspberrypi.org/documentation/installation/installing-images/>

- Das Image des Raspbian Betriebssystems finden Sie auf folgender Seite:

<https://www.raspberrypi.org/downloads/>

### 3.2.2 I<sup>2</sup>C

Der verwendete ID EEPROM enthält Daten für die Identifizierung des Add-On Boards, dadurch weiß der Raspberry Pi welche Hardware sich auf dem Board befindet und wie die GPIOs eingerichtet werden müssen. Dies ermöglicht es der S.USV pi solutions bei Systemstart der Pi Software erfolgreich identifiziert zu werden, einschließlich das Laden aller erforderlichen Treiber.

Benutzer der Vorgängermodelle A und B folgen bitte der manuellen Anleitung:

- Die Kommunikation zwischen der S.USV und dem Raspberry Pi erfolgt über die I<sup>2</sup>C-Schnittstelle, aktivieren und konfigurieren Sie diese zunächst bitte mit den folgenden Schritten:

1. Zuerst müssen Sie die entsprechenden I<sup>2</sup>C-Tools installieren, dadurch sind Sie in der Lage alle Geräte zu überprüfen, die mit der Raspberry Pi verbunden sind. Um die I<sup>2</sup>C-tools Utility zu installieren, führen Sie bitte folgende Befehle im Terminal aus:

***sudo apt-get install python-smbus***

```
pi@raspberrypi ~ $ sudo apt-get install python-smbus
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

***sudo apt-get install i2c-tools***

```
pi@raspberrypi ~ $ sudo apt-get install i2c-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

2. Als nächstes müssen Sie LXTerminal, Konsole oder SSH öffnen, um folgenden Befehl auszuführen:

***sudo nano /etc/modules***

fügen Sie nun folgende Zeilen an das Ende der Datei:

***i2c-bcm2708***

***i2c-dev***

```
GNU nano 2.2.6 File: /etc/modules

# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.

snd-bcm2835
i2c-dev
i2c-bcm2708
rtc-ds1307
```

Speichern Sie die Datei mit **Control-X Y** <return>

3. Abhängig von Ihrer Distribution, verfügen Sie möglicherweise auch über eine Datei namens:

***/etc/modprobe.d/raspi-blacklist.conf***

Sollten Sie nicht über diese Datei verfügen, kann dieser Schritt übersprungen werden. Bei vorhandener Datei, sollten Sie folgende Zeilen auskommentieren, indem sie ein # davor setzen:

***blacklist spi-bcm2708***

***blacklist i2c-bcm2708***

Mit folgendem Befehl öffnen Sie die Datei im Editor:

***sudo nano /etc/modprobe.d/raspi-blacklist.conf***

```
GNU nano 2.2.6 File: /etc/modprobe.d/raspi-blacklist.conf
#blacklist spi-bcm2708
#blacklist i2c-bcm2708
```

bearbeiten Sie hier die Datei wie oben beschrieben, speichern und schließen Sie anschließend die Datei mit **Control-X Y** <return>

4. Wenn Sie eine aktuelle Raspberry Pi (3.18 Kernel oder höher) verwenden, müssen Sie zusätzlich ein Update der Datei ***/boot/config.txt*** durchführen. Editieren Sie diese mit ***sudo nano /boot/config.txt*** und fügen Sie folgende Zeilen an das Ende der Datei:

***dtparam=i2c1=on***

***dtparam=i2c\_arm=on***

```
GNU nano 2.2.6 File: /boot/config.txt
# uncomment if hdmi display is not detected and composite is being output
#hdmi_force_hotplug=1

# uncomment to force a specific HDMI mode (this will force VGA)
#hdmi_group=1
#hdmi_mode=1

# uncomment to force a HDMI mode rather than DVI. This can make audio work
# DMT (computer monitor) modes
#hdmi_drive=2

# uncomment to increase signal to HDMI, if you have interference, blanking,
# no display
#config_hdmi_boost=4

# uncomment for composite PAL
#sdtv_mode=2

#uncomment to overclock the arm. 700 MHz is the default.
#arm_freq=800

# Uncomment some or all of these to enable the optional hardware interfaces
#dtparam=i2c_arm=on
#dtparam=i2s=on
#dtparam=spi=on

# Uncomment this to enable the lirc-rpi module
#dtoverlay=lirc-rpi

# Additional overlays and parameters are documented /boot/overlays/README

dtparam=i2c1=on
dtparam=i2c_arm=on
```

5. Sobald die oben genannten Schritte getan sind, führen Sie folgenden Befehl aus ***sudo reboot*** um das System neu zu starten.

6. Wenn Sie sich nun wieder einloggen, können Sie mit folgendem Befehl alle angeschlossenen Geräte anzeigen lassen:

***sudo i2cdetect -y 1***

```
pi@raspberrypi ~ $ sudo i2cdetect -y 1
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  UU  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

Zwei I<sup>2</sup>C-Adressen sind im Einsatz – 0x0F für die S.USV und 0x68 für die Real Time Clock, die sich auf der S.USV befindet.

Beachten Sie bitte bei den Vorgängermodellen den Befehl in ***sudo i2cdetect -y 0*** abzuändern.

### 3.2.3 S.USV

1. Bitte laden Sie das bereit gestellte Debian-Package aus unserem Download-Bereich und speichern Sie es auf einer lokalen Speicheradresse auf dem Raspberry Pi.
2. Um das Debian-Package auf Ihrem Raspberry Pi zu installieren, wechseln Sie in das gewählte Verzeichnis und führen Sie folgende Befehle in der Befehlszeile aus:

***sudo tar -xvzf susvd-en-x.x-all.tar.gz*** (um die tar-Datei zu entpacken)

***sudo dpkg -i susvd-en-x.x-all.deb*** (um das Debian-Package zu installieren)

```
pi@raspberrypi ~ $ cd /home/pi/S.USV-Install/
pi@raspberrypi ~/S.USV-Install $ ls
susvd-en-1.1-all.deb
pi@raspberrypi ~/S.USV-Install $ sudo dpkg -i susvd-en-1.1-all.deb
```

3. Der S.USV-Client und -Daemon sind nun vollständig installiert und einsatzbereit. (Die installierten Dateien befinden sich in folgendem Pfad: ***/opt/susvd***)

- Im Falle einer erfolgreichen Installation wechseln Sie in das Verzeichnis **/opt/susvd** und führen folgenden Befehl als Superuser aus, um den Daemon zu starten und ein korrektes Arbeiten der S.USV sicherzustellen (siehe Abschnitt 4 für eine Beschreibung aller Befehle):

**sudo ./susvd -start**

```
pi@raspberrypi ~ $ cd /opt/susvd
pi@raspberrypi /opt/susvd $ sudo ./susvd -start
S.USV Daemon started..
pi@raspberrypi /opt/susvd $
```

Der erfolgreiche Start des Daemon wird in der Konsolenausgabe angezeigt.

- Wenn alles erledigt ist, können Sie die volle Funktionalität der S.USV verwenden.

### 3.2.3.1 Register

Command	Array - Field	Returned Value	Mode	Comment
<b>0xD0</b>	0	Voltage in [mV]	Read	first 8 bit
	1		Read	last 8 bit
<b>0xD1</b>	0	Power Extern [mA]	Read	first 8 bit
	1		Read	last 8 bit
<b>0xD2</b>	0	Power Battery [mA]	Read	first 8 bit
	1		Read	last 8 bit
<b>0xD3</b>	0	Battery Voltage [mV]	Read	first 8 bit
	1		Read	last 8 bit
<b>0xD4</b>	0	Battery Status	Read	0 = Charge 1 = Full 2 = Failure
<b>0x35</b>	0	Command	Read	0x35
	1	Charge Status	Read	1 = Activated <0 = Deactivated
	2	Charge Current	Read	0 = 1000mA 1 = 500mA 2 = 300mA
<b>0x22</b>	0	Command	Read	0x22
	1	Firmware	Read	Major Version
	2	Firmware	Read	Minor Version
	3	Model	Read	0 = Advanced 1 = Basic
<b>0x45</b>	0	Command	Read	0x45
	1	Power Status	Read	0 = Secondary 1 = Primary

Command	Parameter	Mode	Comment
<b>0x37</b>	0 = 1000mA 1 = 500mA 2 = 300mA	Write	Change charging current
<b>0x41</b>	i2c-Address	Write	Change i2c-Address
<b>0x29</b>	-	Write	Activate Charging circuit
<b>0x27</b>	-	Write	Deactivate Charging circuit

### 3.2.4 RTC – Real Time Clock

Die integrierte Real Time Clock ist eine nützliche Zugabe für den Raspberry Pi. Bei bestehender Ethernet-Verbindung wird die aktuelle Uhrzeit mit dem Internet über den NTP(Network Time Protocol)-Dienst synchronisiert. In diversen Szenarien ist die Verbindung zu einem Netzwerk jedoch nicht möglich, dies kann u.a. im Auto, an einer Solar- oder Windkraftanlage oder auch bei Verwendung des RPi im Schaltschrank der Fall sein. Zu diesem Zweck kommt die RTC zum Einsatz, die auch bei fehlender Netzwerk Verbindung die Systemzeit aktuell hält.

Der ID EEPROM auf der S.USV konfiguriert das RTC-Modul automatisch.

(Hinweis: Siehe Punkt 2 zum Einrichten der aktuellen Zeit.)

Benutzer der Vorgängermodelle A und B folgen bitte der manuellen Anleitung:

- Bitte überprüfen Sie die korrekte Verbindung des RTC-Chip Moduls indem sie den Befehl ***sudo i2cdetect -y 1*** (i2cdetect -y 0 für das Vorgängermodell Rev. 1 Pi) in der Befehlszeile ausführen. Die DS1307 Real Time Clock sollte sich auf der I<sup>2</sup>C-Adresse ID #68 befinden.

1. Laden Sie das RTC-Modul mit dem Befehl ***sudo modprobe rtc-ds1307*** im Kernel.  
( Bitte verwenden Sie diese Befehle als Superuser )

Für einen Rev. 2 Pi oder später:

```
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
```

Für die ältere Rev.1 Pi:

```
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-0/new_device
```

- Überprüfen Sie nun die Zeit auf dem RTC-Modul mit:

***sudo hwclock -r***

```
pi@raspberrypi ~ $ sudo hwclock -r
Mon 19 Oct 2015 13:33:23 CEST -0.885224 seconds
```

Sollte dies der erste Einsatz der Real Time Clock sein, wird Sie das Datum 1. Januar 2000 zurück melden.

Bitte konfigurieren und bestätigen Sie nun die aktuelle Uhrzeit mit:

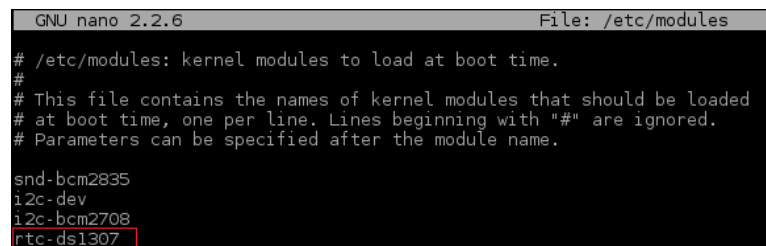
***sudo hwclock -w*** um die Systemzeit auf das RTC-Modul zu schreiben.

- Um die Real Time Clock nun auf der Raspberry Pi einzurichten, müssen Sie das RTC-Modul in die Datei ***/etc/modules*** hinzufügen.

- Gehen Sie in das Terminalfenster und führen folgenden Befehl aus:

***sudo nano /etc/modules***

fügen Sie nun die Zeile ***rtc-ds1307*** an das Ende der Datei.



```
GNU nano 2.2.6 File: /etc/modules
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.
snd-bcm2835
i2c-dev
i2c-bcm2708
rtc-ds1307
```

- Nächster Schritt ist es, die DS1307 RTC durch Editieren der Datei ***/etc/rc.local*** zur Boot Sektion hinzuzufügen.

Führen Sie folgenden Befehl aus:

***sudo nano /etc/rc.local***

und fügen Sie die folgenden Zeilen am Ende der Datei ein:

***echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new\_device***

***sudo hwclock -s***

***date***



```
GNU nano 2.2.6 File: /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
#
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
sudo hwclock -s
date
exit 0
```

Um die Datei zu speichern, drücken Sie **Ctrl+O**, **Y** <return>.

4. Beim nächsten Neustart Ihres Raspberry Pi, wird die aktuelle Uhrzeit von der Real Time Clock auf der S.USV ausgelesen.

#### 3.2.4.1 Zeitgesteuertes Ein- und Ausschalten

In diversen Szenarien ist der Raspberry Pi ein fest verdrahtetes Modul einer Gesamtkomponente. Teilweise leidet die Zugänglichkeit hierzu unter verschiedensten Bedingungen, manchmal wird diese auch ganz bewusst vermieden.

In vielen Fällen soll der Einplatinencomputer zu einer definierten Zeit aufgeweckt werden, diverse Sensordaten o.a. abrufen, diese entsprechend weiterführen um anschließend wieder automatisch in den Shutdown zu fahren.

Auch für den Einsatz als Media-Center ist das zeitgesteuerte Ein- und Ausschalten eine sinnvolle Ergänzung.

Um genau diese Einsatzbereiche durch eine nützliche Funktion zu erweitern, haben wir das Zeitgesteuerte Ein- und Ausschalten des RPi implementiert. Der User kann problemlos die gewünschten Uhrzeiten zum Hoch- und Herunterfahren in die Config übergeben, alles Weitere wird durch den integrierten RTC-Chip und der damit verbundenen Software geregelt.

Weitere Informationen zur Konfiguration und Verwendung der Funktion finden Sie im Kapitel 4.2.1 Client Optionen.

## 4 Client Software

---

Die Kommunikation zwischen S.USV und Raspberry Pi erfolgt über die I<sup>2</sup>C-Schnittstelle unter der Adresse 0x0F.

Grundsätzlich besteht das Softwarepaket der S.USV aus zwei Werkzeugen:

1. Der S.USV Daemon (susvd), überwacht und kontrolliert die S.USV durch konstantes Überprüfen des S.USV Status und reagiert entsprechend auf verschiedene Ereignisse. Der S.USV Daemon wird einmal gestartet und fortan im Hintergrund ausgeführt.
2. Der S.USV Client (susv), ermöglicht dem Benutzer den aktuellen Status der S.USV auszulesen, sowie diese zu kontrollieren. Zum Beispiel, das manuelle Aktivieren oder Deaktivieren der Ladeschaltung. Der S.USV Client ist ebenso für das Bearbeiten der config-Variablen des S.USV Daemon verantwortlich, z.B. des Shutdown-Timers.

### 4.1 S.USV – Daemon

Der S.USV Daemon ist für die Überwachung und Steuerung der S.USV in Verbindung mit dem Raspberry Pi verantwortlich.

Der S.USV Daemon erstellt ein Protokoll in der Datei: ***/var/log/susvd.log***

In den folgenden Abschnitten werden die einzelnen Optionen aufgezeigt.

#### 4.1.1 Daemon Konfiguration

Um den S.USV Daemon zu konfigurieren wechseln Sie in das Verzeichnis ***/opt/susvd*** und führen folgende Befehle als Superuser aus:

***sudo ./susv -timer <Zeit in Sekunden>***

(Default Wert = 10)

Dieser Wert gibt an, wie lange das System weiterläuft bevor der File Safe Shutdown durch die S.USV initiiert wird, nachdem die Spannungsversorgung auf Akkubetrieb umgeschaltet hat.

Werte ">=0" sind möglich.

```

pi@raspberrypi ~ $ sudo ./susv -timer 60
*****
*
* S.USV pi solutions
* www.s-usv.de
*
* Model: Advanced
* Firmware Version: 1.3
* Software Version: 1.3
*
* Mail notification: Enabled
*
* Timed Boot: Enabled
* Boot time: 07:00:00
*
* Timed Shutdown: Enabled
* Shutdown time: 18:00:00
*
* Sun Jan 10 09:00:10 2016
*
*****
* Shutdown timer set to: 60
*
* Please restart S.USV Daemon
*
*****

```

Diese Funktion kann durch den Wert "-1" deaktiviert werden.

In diesem Fall läuft der Raspberry Pi weiter, bis die Akkurestkapazität einen Wert von 10% erreicht. Von diesem Zeitpunkt an wird die S.USV automatisch einen File Safe Shutdown durchführen, um den Akku vor Beschädigungen zu schützen.

***sudo ./susv -auto <0/1>***

(Default Wert = 1)

Dieser Wert bestimmt das Startverhalten des S.USV Daemon. Der Wert "1" aktiviert den Autostart, der Wert "0" deaktiviert den Autostart.

```

pi@raspberrypi ~ $ sudo ./susv -auto 1
*****
*
* S.USV pi solutions
* www.s-usv.de
*
* Model: Advanced
* Firmware Version: 1.3
* Software Version: 1.3
*
* Mail notification: Enabled
*
* Timed Boot: Enabled
* Boot time: 07:00:00
*
* Timed Shutdown: Enabled
* Shutdown time: 18:00:00
*
* Sun Jan 10 09:00:31 2016
*
*****
* Autostart enabled
*
* Please restart S.USV Daemon
*
*****

```

Beachten Sie bitte, dass der S.USV Daemon Service bei deaktiviertem Autostart manuell gestartet werden muss, um eine Korrektheit der S.USV Funktion zu garantieren.

***sudo ./susv -sleep <Zeit in Sekunden>***

(Default Wert = 1)

Dieser Wert bestimmt die Wiederholung in dem der S.USV Daemon die Ausgangsspannung der S.USV überprüft. Werte "> = 0" sind möglich.

```

pi@raspberrypi ~ $ sudo ./susv -sleep 1
*****
* S.USV pi solutions *
* www.s-usv.de *
* *
* Model: Advanced *
* Firmware Version: 1.3 *
* Software Version: 1.3 *
* *
* Mail notification: Enabled *
* *
* Timed Boot: Enabled *
* Boot time: 07:00:00 *
* *
* Timed Shutdown: Enabled *
* Shutdown time: 18:00:00 *
* *
* Sun Jan 10 09:01:02 2016 *
* *
*****
* Sleep timer set to: 1 *
* *
* Please restart S.USV Daemon *
* *
*****

```

#### 4.1.2 Daemon Steuerung

Zur Steuerung des S.USV Daemon wechseln Sie in das Verzeichnis ***/opt/susvd*** und führen folgende Befehle als Superuser aus:

***sudo ./susvd -start***

Startet den S.USV Daemon-Service und dessen Konfiguration.

***sudo ./susvd -stop***

Stoppt den S.USV Daemon Service.

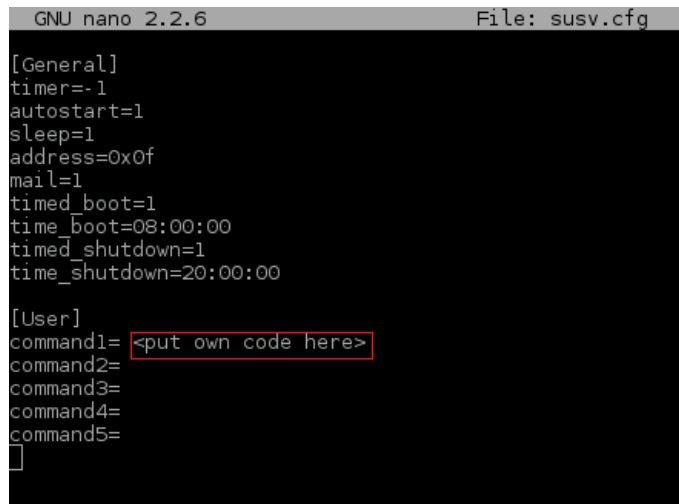
***sudo ./susvd -restart***

Startet den S.USV Daemon Service neu.

#### 4.1.3 User Shutdown Script

Innerhalb der S.USV Config ist es möglich, diverse Befehle oder ganze Scripts einzubinden. Diese werden entsprechend beim Shutdown durch den Daemon ausgeführt, so hat der User die Möglichkeit den Shutdown Prozess ganz nach seinen Wünschen und Bedingungen zu konfigurieren.

Wechseln Sie hierzu in das Verzeichnis **/opt/susvd** und editieren Sie die zugehörige config Datei mit **sudo nano susv.cfg**

A screenshot of a terminal window showing the GNU nano 2.2.6 text editor editing the file susv.cfg. The editor has a dark background with light green text. The content is divided into two sections: [General] and [User]. The [General] section contains configuration options for timer, autostart, sleep, address, mail, timed boot, and timed shutdown. The [User] section contains five command slots, with the first one containing the placeholder text '<put own code here>' which is highlighted with a red rectangular box. The cursor is at the end of the first command line in the [User] section.

```
GNU nano 2.2.6 File: susv.cfg

[General]
timer=-1
autostart=1
sleep=1
address=0x0f
mail=1
timed_boot=1
time_boot=08:00:00
timed_shutdown=1
time_shutdown=20:00:00

[User]
command1= <put own code here>
command2=
command3=
command4=
command5=

```

## 4.2 S.USV – Client

Der S.USV-Client ermöglicht dem Benutzer die Zustandsüberwachung und Funktionskontrolle der S.USV.

In den folgenden Abschnitten werden die einzelnen Optionen aufgezeigt.

### 4.2.1 Client Optionen

Zur Steuerung des S.USV-Clients wechseln Sie in das Verzeichnis **/opt/susvd** und führen folgende Befehle aus:

**./susv -help**

Mit diesem Befehl werden alle möglichen Optionen aufgelistet.

#### Übersicht und Notifikation:

**./susv -status**

S.USV Status auslesen.

Dieser Befehl erlaubt es, den Status der S.USV auszulesen. Hierfür werden alle möglichen Zustände bereitgestellt, sowie die aktuelle Spannungsquelle und dessen Stromverbrauch.

```
pi@raspberrypi ~ $ sudo ./susv -status
*****
*                               *
* S.USV pi solutions            *
* www.s-usv.de                  *
*                               *
* Model: Advanced                *
* Firmware Version: 1.3          *
* Software Version: 1.3          *
*                               *
* Mail notification: Enabled     *
*                               *
* Timed Boot: Enabled            *
* Boot time: 07:00:00            *
*                               *
* Timed Shutdown: Enabled        *
* Shutdown time: 18:00:00        *
*                               *
* Sun Jan 10 09:39:11 2016       *
*                               *
*****
*                               *
* Powering Source: Primary       *
* Charging circuit: ONLINE       *
* Charging current: 300 mA       *
*                               *
* Voltage in: 5.29 V             *
* Battery capacity: 96.20%       *
* Battery voltage: 4.18V         *
* Power Battery: 000.00 mA       *
* Power Extern: 372.82 mA        *
*                               *
* Shutdown timer: -1             *
* Autostart: enabled             *
* Sleep timer: 1                 *
*                               *
*****
```

***./susv -mail <1/0>***

(Default Wert = 0)

Notifikation über E-Mail aktivieren/deaktivieren.

Bei Aktivierung der Notifikation über E-Mail, wird bei Spannungsverlust eine entsprechende Notifikation an die konfigurierte Adresse übermittelt.

Wechseln Sie hierzu in das Verzeichnis ***/opt/susvd/scripts*** und editieren Sie das entsprechende Python-Script mit ***sudo nano mail.py***

```

GNU nano 2.2.6                               File: mail.py
import smtplib
from email.MIMEText import MIMEText
from email.MIMEBase import MIMEBase
from email import encoders

fromaddr = "YOUR EMAIL"
toaddr = "EMAIL ADDRESS YOU SEND TO"

msg = MIMEText('')

msg['From'] = fromaddr
msg['To'] = toaddr
msg['Subject'] = "SUBJECT OF THE EMAIL"

body = "TEXT YOU WANT TO SEND"

msg.attach(MIMEText(body, 'plain'))

filename = "susvd.log"
attachment = open("/var/log/", "rb")

part = MIMEBase('application', 'octet-stream')
part.set_payload((attachment).read())
encoders.encode_base64(part)
part.add_header('Content-Disposition', "attachment; filename= %s" % filename)

msg.attach(part)

server = smtplib.SMTP('smtp.gmail.com', 587)
server.starttls()
server.login(fromaddr, "YOUR PASSWORD")
text = msg.as_string()
server.sendmail(fromaddr, toaddr, text)
server.quit()

```

### Zeitgesteuertes Ein- und Ausschalten:

***./susv -boot <1/0>***

(Default Wert = 0)

Zeitgesteuertes Einschalten aktivieren/deaktivieren.

Über diesen Befehl wird das zeitgesteuerte Einschalten des Raspberry Pi konfiguriert. Der Wert "1" steht für die Aktivierung, der Wert "0" für die Deaktivierung.

```

pi@raspberrypi ~ $ sudo ./susv -boot 1
*****
*
* S.USV pi solutions
* www.s-usv.de
*
* Model: Advanced
* Firmware Version: 1.3
* Software Version: 1.3
*
* Mail notification: Enabled
*
* Timed Boot: Enabled
* Boot time: 07:00:00
*
* Timed Shutdown: Enabled
* Shutdown time: 18:00:00
*
* Sun Jan 10 09:39:57 2016
*
*****
*
* Timed boot enabled
*
* Please restart S.USV Daemon
*
*****

```

### ***./susv -setboot <HH:mm:ss>***

Uhrzeit für das zeitgesteuerte Einschalten konfigurieren.

Konfigurieren Sie über diesen Befehl die gewünschte Uhrzeit, um den Raspberry Pi zur definierten Zeit hochzufahren.

```

pi@raspberrypi ~ $ sudo ./susv -setboot 08:00:00
*****
*
* S.USV pi solutions
* www.s-usv.de
*
* Model: Advanced
* Firmware Version: 1.3
* Software Version: 1.3
*
* Mail notification: Enabled
*
* Timed Boot: Enabled
* Boot time: 08:00:00
*
* Timed Shutdown: Enabled
* Shutdown time: 18:00:00
*
* Sun Jan 10 09:40:15 2016
*
*****
*
* Boot time 08:00:00
*
* Please restart S.USV Daemon
*
*****

```

### ***./susv -shutdown <1/0>***

(Default Wert = 0)

Zeitgesteuertes Ausschalten aktivieren/deaktivieren.

Über diesen Befehl wird das zeitgesteuerte Ausschalten des Raspberry Pi konfiguriert. Der Wert "1" steht für die Aktivierung, der Wert "0" für die Deaktivierung.



```

pi@raspberrypi ~ $ sudo ./susv -shutdown 1
*****
*
* S.USV pi solutions
* www.s-usv.de
*
* Model: Advanced
* Firmware Version: 1.3
* Software Version: 1.3
*
* Mail notification: Enabled
*
* Timed Boot: Enabled
* Boot time: 08:00:00
*
* Timed Shutdown: Enabled
* Shutdown time: 18:00:00
*
* Sun Jan 10 09:40:33 2016
*
*****
* Timed shutdown enabled
*
* Please restart S.USV Daemon
*
*****

```

### ***./susv -setshutdown <HH:mm:ss>***

Uhrzeit für das zeitgesteuerte Ausschalten konfigurieren.

Konfigurieren Sie über diesen Befehl die gewünschte Uhrzeit, um den Raspberry Pi zur definierten Zeit herunterzufahren.

```

pi@raspberrypi ~ $ sudo ./susv -setshutdown 20:00:00
*****
*
* S.USV pi solutions
* www.s-usv.de
*
* Model: Advanced
* Firmware Version: 1.3
* Software Version: 1.3
*
* Mail notification: Enabled
*
* Timed Boot: Enabled
* Boot time: 08:00:00
*
* Timed Shutdown: Enabled
* Shutdown time: 20:00:00
*
* Sun Jan 10 09:40:46 2016
*
*****
* Shutdown time 20:00:00
*
* Please restart S.USV Daemon
*
*****

```

## **Stromquellen und Versorgung:**

### ***./susv -vin [0]***

Auslesen der Eingangsspannung.

Mit diesem Befehl ist es möglich, die aktuelle Eingangsspannung zu überprüfen.

```
pi@raspberrypi ~ $ sudo ./susv -vin
*****
*                               *
* S.USV pi solutions           *
* www.s-usv.de                 *
*                               *
* Model: Advanced               *
* Firmware Version: 1.3        *
* Software Version: 1.3        *
*                               *
* Mail notification: Enabled    *
*                               *
* Timed Boot: Enabled           *
* Boot time: 07:00:00           *
*                               *
* Timed Shutdown: Enabled       *
* Shutdown time: 18:00:00       *
*                               *
* Sun Jan 10 09:01:47 2016      *
*                               *
*                               *
* Voltage in: 5.31 V            *
*                               *
*****
```

(Hinweis: Eine angehängte „0“ an den Befehl gibt lediglich den Wert aus.)

#### ***./susv -pwrext [0]***

Auslesen des externen Stromverbrauchs.

Bei Versorgung über den externen Spannungseingang kann der aktuelle externe Stromverbrauch mit diesem Befehl überprüft werden.

```
pi@raspberrypi ~ $ sudo ./susv -pwrext
*****
*                               *
* S.USV pi solutions           *
* www.s-usv.de                 *
*                               *
* Model: Advanced               *
* Firmware Version: 1.3        *
* Software Version: 1.3        *
*                               *
* Mail notification: Enabled    *
*                               *
* Timed Boot: Enabled           *
* Boot time: 07:00:00           *
*                               *
* Timed Shutdown: Enabled       *
* Shutdown time: 18:00:00       *
*                               *
* Sun Jan 10 09:03:22 2016      *
*                               *
*                               *
* Power Extern: 333.28 mA       *
*                               *
*****
```

(Hinweis: Eine angehängte „0“ an den Befehl gibt lediglich den Wert aus.)

#### ***./susv -pwrbat [0]***

Auslesen des Stromverbrauchs der Batterie.

Bei Versorgung über Akku kann der aktuelle Stromverbrauch des Akkus mit diesem Befehl überprüft werden.

```

pi@raspberrypi ~ $ sudo ./susv -pwrbat
*****
*
* S.USV pi solutions
* www.s-usv.de
*
* Model: Advanced
* Firmware Version: 1.3
* Software Version: 1.3
*
* Mail notification: Enabled
*
* Timed Boot: Enabled
* Boot time: 07:00:00
*
* Timed Shutdown: Enabled
* Shutdown time: 18:00:00
*
* Sun Jan 10 09:37:23 2016
*
*****
*
* Power Battery: 349.40 mA
*
*****

```

(Hinweis: Eine angehängte „0“ an den Befehl gibt lediglich den Wert aus.)

## Akku- und Ladeschaltung:

### ***./susv -capbat [0]***

Auslesen der Batterie – Restkapazität.

Dieser Befehl erlaubt es, die aktuelle Batteriespannung sowie die verbleibende Akkukapazität auszulesen.

```

pi@raspberrypi ~ $ sudo ./susv -capbat
*****
*
* S.USV pi solutions
* www.s-usv.de
*
* Model: Advanced
* Firmware Version: 1.3
* Software Version: 1.3
*
* Mail notification: Enabled
*
* Timed Boot: Enabled
* Boot time: 07:00:00
*
* Timed Shutdown: Enabled
* Shutdown time: 18:00:00
*
* Sun Jan 10 09:37:44 2016
*
*****
*
* Battery capacity: 96.20%
* Battery voltage: 4.18V
*
*****

```

(Hinweis1: Eine Restkapazität von <25% wird durch die LED BATRD signalisiert. Bei einer Restkapazität von <10% wird der Raspberry Pi automatisch heruntergefahren.)

(Hinweis2: Eine angehängte „0“ an den Befehl gibt lediglich den Wert aus.)

### ***sudo ./susv -chrgpwr <300/500/1000>***

(Bitte verwenden Sie diesen Befehl als Superuser)

Konfiguriert den Ladestrom für den Akku.

Verwenden Sie diesen Befehl, um die aktive Ladestromstärke zu konfigurieren. Um die Ladezeit der Akkus zu minimieren stehen folgende Stromstärken zur Verfügung:

- 300mA
- 500mA
- 1000mA

```
pi@raspberrypi ~ $ sudo ./susv -chrgpwr 1000
*****
* S.USV pi solutions          *
* www.s-usv.de               *
*                             *
* Model: Advanced             *
* Firmware Version: 1.3       *
* Software Version: 1.3       *
*                             *
* Mail notification: Enabled   *
*                             *
* Timed Boot: Enabled         *
* Boot time: 07:00:00         *
*                             *
* Timed Shutdown: Enabled     *
* Shutdown time: 18:00:00     *
*                             *
* Sun Jan 10 09:38:02 2016    *
*                             *
*****
* Charging current: 1000 mA    *
*                             *
*****
```

(Hinweis: Der konfigurierte Ladestrom wird im EEPROM gespeichert und bei Systemstart geladen.)

### ***./susv -chrgon***

Aktiviert die Ladeschaltung.

Erlaubt das manuelle Aktivieren der Akku Ladeschaltung.

```
pi@raspberrypi ~ $ sudo ./susv -chrgon
*****
* S.USV pi solutions          *
* www.s-usv.de               *
*                             *
* Model: Advanced             *
* Firmware Version: 1.3       *
* Software Version: 1.3       *
*                             *
* Mail notification: Enabled   *
*                             *
* Timed Boot: Enabled         *
* Boot time: 07:00:00         *
*                             *
* Timed Shutdown: Enabled     *
* Shutdown time: 18:00:00     *
*                             *
* Sun Jan 10 09:01:21 2016    *
*                             *
*****
* Charging circuit: Online     *
*                             *
*****
```

(Hinweis: Die Konfiguration der Ladeschaltung wird im EEPROM gespeichert und bei Systemstart geladen.)

### ***./susv -chrgoff***

Deaktivieren der Ladeschaltung.

Erlaubt das manuelle Deaktivieren der Akku Ladeschaltung.

```

pi@raspberrypi ~ $ sudo ./susv -chrgoff
*****
*                               *
* S.USV pi solutions           *
* www.s-usv.de                 *
*                               *
* Model: Advanced              *
* Firmware Version: 1.3        *
* Software Version: 1.3        *
*                               *
* Mail notification: Enabled   *
*                               *
* Timed Boot: Enabled          *
* Boot time: 07:00:00          *
*                               *
* Timed Shutdown: Enabled     *
* Shutdown time: 18:00:00     *
*                               *
* Sun Jan 10 09:01:33 2016     *
*                               *
*****
* Charging circuit: Offline    *
*                               *
*****

```

(Hinweis: Die Konfiguration der Ladeschaltung wird im EEPROM gespeichert und bei Systemstart geladen.)

### Firmware und Adressierung:

#### ***./susv -flash <Pfad der HEX Datei>***

Aktualisieren der Firmware.

Benützen Sie diesen Befehl um die aktuelle Firmware zu aktualisieren.

```

pi@raspberrypi ~ $ ./susv -flash /home/pi/S.USV-Install/susv_fw_11.hex
device       : /dev/i2c-1      (address: 0x30)
version      : TWIBOOT m8v2.1 (sig: 0x1e 0x93 0x07 => AVR Mega 8)
flash size   : 0x1c00 / 7168   (0x40 bytes/page)
eeprom size  : 0x0200 / 512
writing flash : [*****] (4228)
pi@raspberrypi ~ $ 

```

(Hinweis: Bitte nach Abschluss des Flash-Vorgangs und erneuter Initialisierung der S.USV einen Neustart des Daemons durchführen.)

#### ***sudo ./susv -chgadd <0x..>***

(Default Adresse = 0x0f)

(Bitte verwenden Sie diesen Befehl als Superuser)

Konfiguriert die I<sup>2</sup>C-Adresse der S.USV.

Um mögliche Kompatibilitätsprobleme zu vermeiden, besteht die Möglichkeit, die S.USV I<sup>2</sup>C-Adresse über diesen Befehl zu verändern. Überprüfen Sie die verwendete Adresse über den Befehl **i2cdetect -y 1**.

```

pi@raspberrypi ~ $ sudo ./susv -chgadd 0x32
*****
*
* S.USV pi solutions
* www.s-usv.de
*
* Model: Advanced
* Firmware Version: 1.3
* Software Version: 1.3
*
* Mail notification: Enabled
*
* Timed Boot: Enabled
* Boot time: 07:00:00
*
* Timed Shutdown: Enabled
* Shutdown time: 18:00:00
*
* Sun Jan 10 09:38:30 2016
*
*****
*
* I2C-Address set to: 0x32
*
* Please restart S.USV Daemon
*
*****
pi@raspberrypi ~ $ cd /opt/susvd
pi@raspberrypi /opt/susvd $ sudo ./susvd -restart
S.USV Daemon stopped!
S.USV Daemon started..
pi@raspberrypi /opt/susvd $ sudo i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  32  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  UU  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --

```

(Hinweis: Die konfigurierte I<sup>2</sup>C-Adresse wird im EEPROM gespeichert und bei Systemstart geladen.)

