



Benutzerhandbuch - S.USV solutions

Kompatibel mit Raspberry Pi, UP Board und Tinker Board
Revision 2.0 | Datum 15.03.2017

Inhaltsverzeichnis

1 Funktionen	3
2 Technische Spezifikationen	4
2.1 Übersicht	5
2.2 Ausführung	5
2.3 LED - Statusanzeige	6
3 Installationsanleitung	7
3.1 Hardware	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	10
3.2 Software	10
3.2.1 Raspberry Pi - Raspbian	10
3.2.2 UP Board – Ubilinux	11
3.2.3 Tinker Board – TinkerOS	11
3.2.4 I ² C	12
3.2.5 S.USV	13
3.2.5.1 Register	14
3.2.6 RTC – Real Time Clock	15
3.2.6.1 Zeitgesteuertes Ein- und Ausschalten	15
4 Client Software	16
4.1 S.USV – Daemon	16
4.1.1 Daemon Konfiguration	17
4.1.2 Daemon Steuerung	18
4.1.3 User Shutdown Script	19
4.2 S.USV – Client	20
4.2.1 Client Optionen	20
5 Support	29
5.1 Tutorials	29
5.1.1 RTC Konfiguration - SYSTEMD	29
5.2 FAQ – Frequently Asked Questions	30
5.3 Troubleshooting	35

1 Funktionen

S.USV solutions sind erweiterte Energieversorgungs-Zusatzmodule für Einplatinencomputer, mit dem Hauptfokus auf die unterbrechungsfreie Stromversorgung.

Nebenbei liefern die Module noch weitere Zusatzfunktionen, um die Bedienung des Einplatinencomputers durch den Benutzer zu optimieren.

Die Module sind voll funktionsfähige Plug & Play Lösungen. Die Stromversorgung der S.USV solutions Module erfolgt direkt über die GPIO Power Pins der Einplatinencomputer und nutzt somit eine gemeinsame Spannungsquelle, dadurch sind keine weiteren Verkabelungen oder Stromversorgungen notwendig. Zusätzlich sind die Module mit einem LiPo-Akku ausgestattet, der über einen Boost-Schaltleistungswandler den notwendigen Spannungsbereich abdeckt, um die Einplatinencomputer 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 konforme USV Module
- Kompatibel mit Raspberry Pi (S.USV pi), AAEON UP Board (S.USV UPs) und ASUS Tinker Board (S.USV tinker)
- Adapterlösung für Raspberry Pi Model A und B
- Unterbrechungsfreie Stromversorgung
- Plug & Play
- Monitoring – Überwachungssystem (V/mA)
- Intelligente Softwarelösung inklusive Mobiler Applikation
- Integrierter LiPo-Akku (300mA) mit intelligenter automatischer Ladefunktion
- Battery Management Controller (Überlast-, Tiefentlade- und Überhitzungsschutz)
- Battery Monitoring System
- Netzeingang mit erweitertem Spannungsbereich von 7-24 Volt
- Real Time Clock mit Battery Back-Up
- Zeitgesteuertes Ein- und Ausschalten des Einplatinencomputers
- Supply Switch (Ein- und Ausschalttaster / File Safe Shutdown)
- LED – Statusanzeige
- Bootloader für Live – Firmware Updates

2 Technische Spezifikationen

	S.USV pi		S.USV up	S.USV tinker
	basic	advanced	UPs	tinker
Plug & Play	√	√	√	√
Stromversorgung	5 Volt/2500 mA	5 Volt/2500 mA	5 Volt/2500 mA	5 Volt/2500 mA
Erweiterter Spannungsbereich	x	7-24 Volt/3500 mA	7-24 Volt/3500 mA	7-24 Volt/3500 mA
Sekundäre Ausgangsleistung	5 Volt/3500 mA	5 Volt/3500 mA	5 Volt/3500 mA	5 Volt/3500 mA
Schnittstellen	I ² C	I ² C	I ² C	I ² C
ID EEPROM	√	√	x	√
Monitoring - System	√	√	√	√
LiPo - Akku	√	√	√	√
Battery Management	√	√	√	√
Battery Monitoring	√	√	√	√
Real Time Clock	√	√	√	√
Supply Switch	√	√	√	√
Kompatibilität	Pi 3, Pi 2, Pi A+, Pi B+	Pi 3, Pi 2, Pi A+, Pi B+	Alle UP Boards	Alle tinker Boards
Abmessungen	65x56,5 mm	65x56,5 mm	65x56,5 mm	65x56,5 mm

	300 mAh	3000 mAh
Nennspannung	3.7 V	3.7 V
Betriebsspannung	3.0 - 4.2 V	3.0 - 4.2 V
Kapazität	300mAh	3000mAh
Interne Impedanz	≤60mΩ	≤30mΩ
Konstante Ladung Entladestrom	2C 15C	1C 2C
Arbeitstemperatur	charge: 0-45°C; discharge: -20-60°C	charge: 0-45°C; discharge: -20-60°C
Steckverbinder	JST-PH-2P	JST-PH-2P
Anschlusskabel	UL1571#28 / 50mm	UL1571#28 / 50mm
Wrapping Way	blue PVC	blue PVC
Line Drawn Out Direction	Center	Center
Maße	30.0 x 20.0 x 6.7 mm	60.0 x 50.0 x 9.0 mm

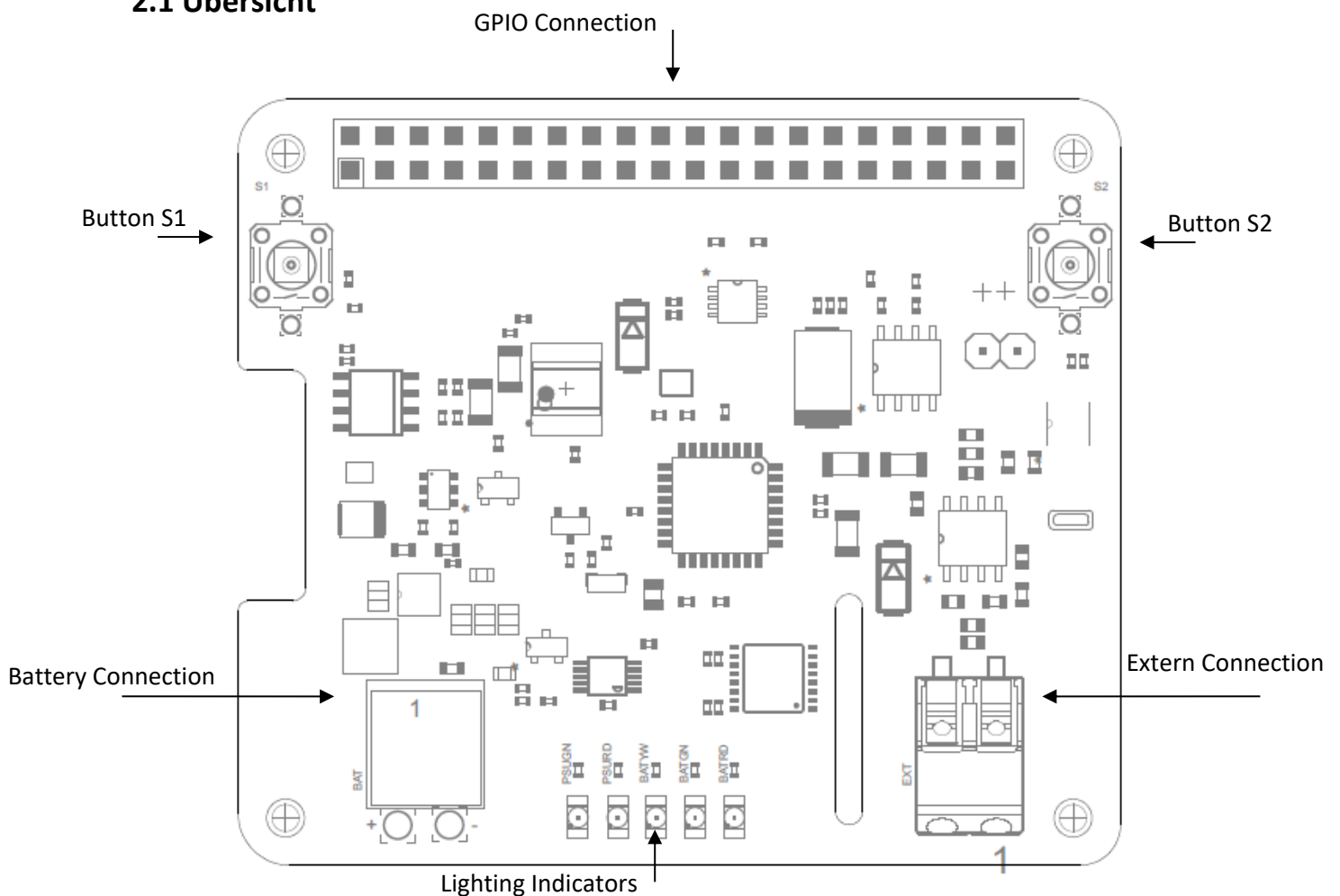
Überbrückungszeiten bei Stromausfall (Batteriebetrieb):

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	6-8 Stunden
Raspberry Pi 2B	500mA	30 Minuten	6-8 Stunden
Raspberry Pi 3B	500mA	30 Minuten	6-8 Stunden
AAEON UP Board	500mA	30 Minuten	4-5 Stunden
ASUS Tinker Board	500mA	30 Minuten	4-5 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 zum Einplatinencomputer.
- **Button S1:**
 - **Raspberry Pi / Tinker Board:** Drücken Sie den Taster für einen Neustart oder halten Sie diesen mindestens 3 Sekunden lang zum Herunterfahren.
 - **UP Board:** Drücken Sie den Taster für einen Shutdown oder halten Sie diesen mindestens 3 Sekunden lang für einen Neustart. Drücken Sie den Taster im ausgeschalteten Zustand, um den Einplatinencomputer hochzufahren.
- **Button S2:**
 - **Raspberry Pi / Tinker Board:** Drücken Sie den Taster zum Hochfahren des Einplatinencomputers.
- **Lighting Indicators:** LED-Statusanzeige für die S.USV.

2.3 LED - Statusanzeige

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

3 Installationsanleitung

3.1 Hardware

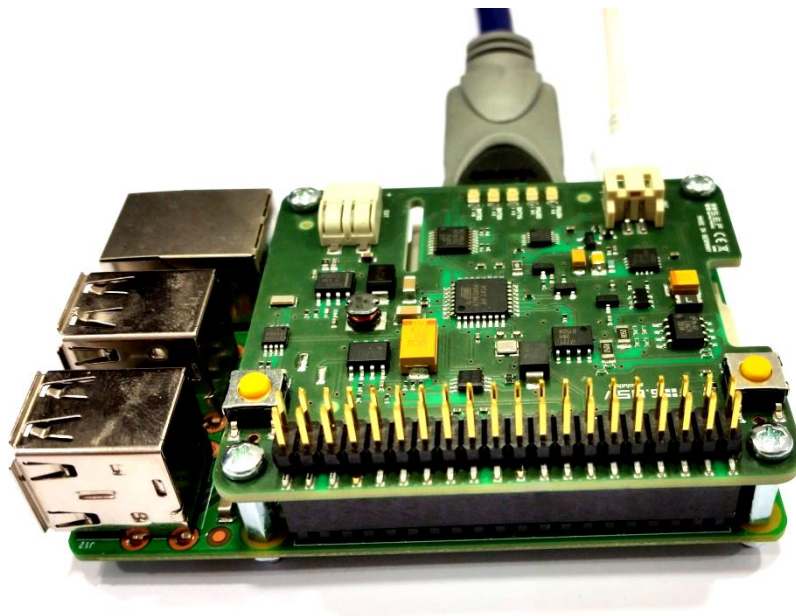
Um über die vollständige Funktionalität der S.USV zu verfügen, empfehlen wir den Akku im ersten Schritt vollständig zu laden.

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

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

3.1.1 Inbetriebnahme S.USV

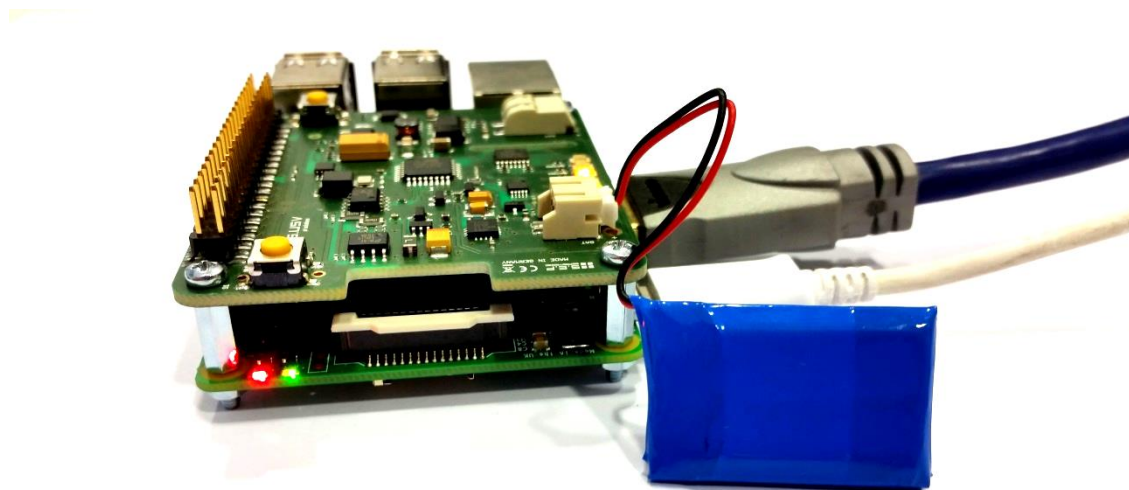
- Die primäre Stromversorgung der S.USV erfolgt über den GPIO Port – Pin 2 (+5V) des Einplatinencomputers. Bitte stecken Sie nun die Platine wie dargestellt auf den Einplatinencomputer, um die benötigte Verbindung herzustellen und befestigen Sie diese mit dem beigegebenen Mounting Kit.



3.1.2 Anschluss des Akkus

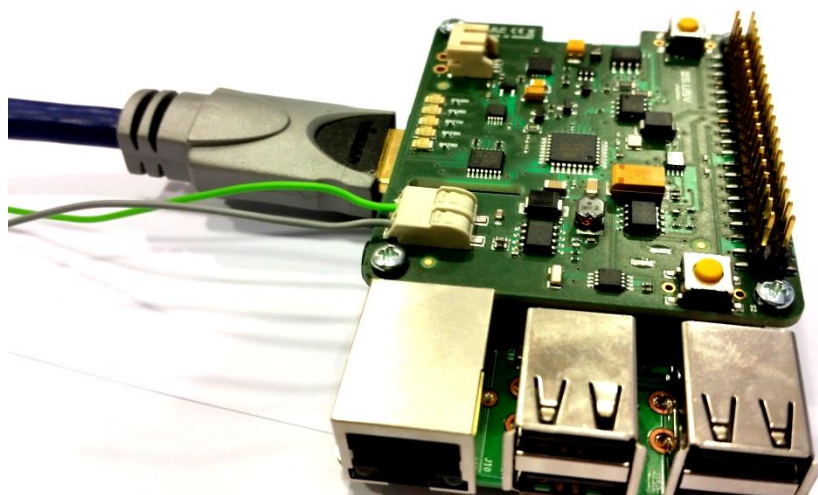
- Bitte stellen Sie für die Inbetriebnahme sicher, dass der mitgelieferte LiPo - Akku in die dafür vorgesehene JST-Buchse auf der Vorderseite der Platine eingesteckt ist.

Eine Inbetriebnahme des Systems ohne gesteckten Akku kann zu Beschädigung führen.



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.



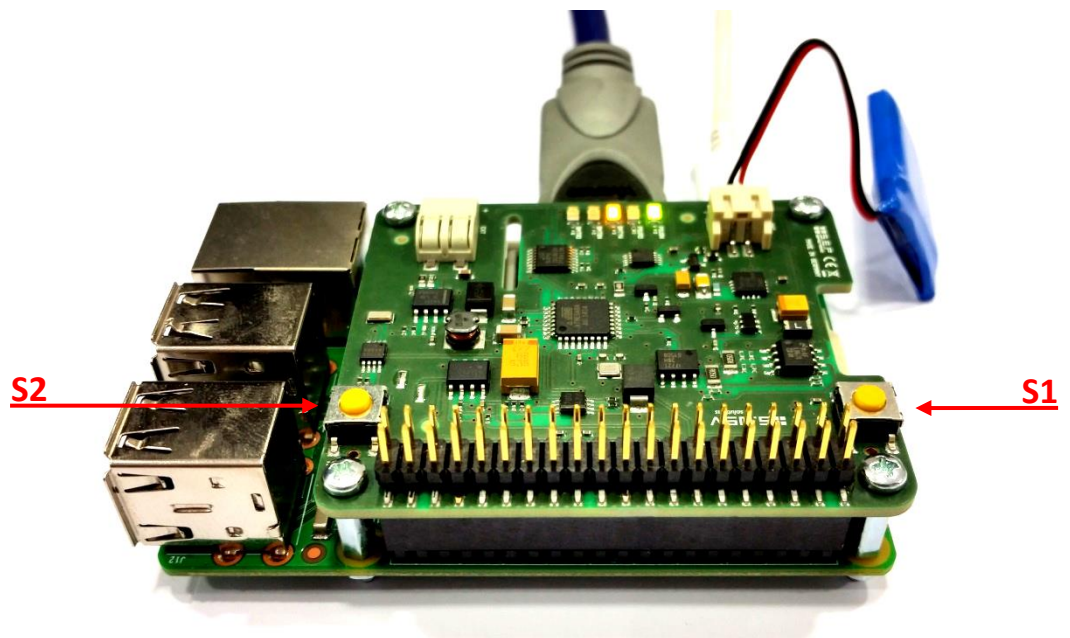
3.1.4 Verwendung der Taster

- **Raspberry Pi / Tinker Board:**

Für das Ein- und Ausschalten, sowie das Neustarten des Einplatinencomputers, 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.



- **UP Board:**

Für das Ein- und Ausschalten, sowie das Neustarten des UP Boards, verbinden Sie den Taster S1 über das beigegefügte PicoBlade Kabel mit **CN33 – Power Button Wafer** des UP Boards.

- **S1** : Power Off / Reboot

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, I²C) – I²C Datenleitung
- **Pin #05:** GPIO 03 (SCL1, I²C) – I²C Taktleitung
- **Pin #13:** GPIO 27 (GPIO_GEN2) – Zustandsüberwachung S.USV
- **Pin #27:** ID_SD (I²C ID EEPROM) – ID Datenleitung
- **Pin #28:** ID_SC (I²C ID EEPROM) – ID Taktleitung

3.2 Software

3.2.1 Raspberry Pi - 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 UP Board – Ubilinux

- Für die Installation und Konfiguration des UP Boards, empfehlen wir folgende Kurzanleitung:

<https://up-community.org/wiki/Setup>

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

https://up-community.org/wiki/Installing_ubilinux

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

<https://up-community.org/downloads/ubilinux>

3.2.3 Tinker Board – TinkerOS

- Für die Installation und Konfiguration des Tinker Boards, empfehlen wir folgende Kurzanleitung:

<https://www.asus.com/de/Motherboards/TINKER-BOARD/HelpDesk/>

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

<https://www.asus.com/de/Motherboards/TINKER-BOARD/HelpDesk/>

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

https://www.asus.com/de/supportonly/Tinker%20Board2GB/HelpDesk_Download/

3.2.4 I²C

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

- Die Kommunikation zwischen der S.USV und dem Einplatinencomputer erfolgt über die I²C-Schnittstelle, aktivieren und überprüfen Sie diese zunächst bitte mit den folgenden Schritten:
 - Zuerst müssen Sie die entsprechenden I²C-Tools installieren, dadurch sind Sie in der Lage alle Geräte zu überprüfen, die mit dem Einplatinencomputer verbunden sind. Um die I²C-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
```

- Sobald die oben genannten Schritte getan sind, führen Sie folgenden Befehl aus ***sudo reboot*** um das System neu zu starten.
- 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:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  -- 0f
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  UU  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

Zwei I²C-Adressen sind im Einsatz – 0x0F für die S.USV und 0x68 für die Real Time Clock, die sich auf der S.USV befindet.

3.2.5 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 Einplatinencomputer.
2. Um das Debian-Package auf Ihrem Einplatinencomputer zu installieren, wechseln Sie in das gewählte Verzeichnis und führen Sie folgende Befehle in der Befehlszeile aus:

sudo tar -xvf susvd-en-x.x-all.tar (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***)
4. 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.

3.2.5.1 Register

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

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

3.2.6 RTC – Real Time Clock

Die integrierte Real Time Clock ist eine nützliche Zugabe für den Einplatinencomputer. 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 Einplatinencomputer 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 1 zum Einrichten der aktuellen Zeit.)

- Bitte überprüfen Sie die korrekte Verbindung des RTC-Chip Moduls indem sie den Befehl ***sudo i2cdetect -y 1*** in der Befehlszeile ausführen. Die DS1308 Real Time Clock sollte sich auf der I²C-Adresse ID #68 befinden.

1. 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.

3.2.6.1 Zeitgesteuertes Ein- und Ausschalten

In diversen Szenarien ist der Einplatinencomputer 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 Einplatinencomputers 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 Einplatinencomouter erfolgt über die I²C-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 Einplatinencomputer 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 Einplatinencomouter 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`**

```
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.MIMEmultipart import MIMEmultipart
from email.MIMEtext import MIMEtext
from email.MIMEbase import MIMEbase
from email import encoders

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

msg = MIMEmultipart()

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 Einplatinencomputers 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 Einplatinencomputer 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 Einplatinencomputers 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 Einplatinencomputer 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 Einplatinencomputer 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 Systems durchführen.)

`sudo ./susv -chgadd <0x.>`

(Default Adresse = 0x0f)

(Bitte verwenden Sie diesen Befehl als Superuser)

Konfiguriert die I²C-Adresse der S.USV.

Um mögliche Kompatibilitätsprobleme zu vermeiden, besteht die Möglichkeit, die S.USV I²C-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 I²C-Adresse wird im EEPROM gespeichert und bei Systemstart geladen.)

5 Support

Bitte wenden Sie sich an support@s-usv.de, insofern Sie weitere Unterstützung benötigen.

5.1 Tutorials

5.1.1 RTC Konfiguration - SYSTEMD

Das entsprechende RTC Modul (DS1307) wird automatisch durch den ID EEPROM konfiguriert und in den Kernel geladen. Bitte führen Sie die folgenden Schritte durch, um die RTC als Hardware Clock zu verwenden:

1. Deaktivieren Sie die fake-hwclock des Einplatinencomputers, welche zu Komplikationen mit der RTC führen kann. Verwenden Sie hierzu folgende Befehle im Terminal:

```
sudo apt-get -y remove fake-hwclock  
sudo update-rc.d -f fake-hwclock remove
```

2. Nachdem die fake-hwclock deaktiviert wurde, sollte das Original Hardware-Clock Script entsprechend angepasst werden. Editieren der Datei */lib/udev/hwclock-set* und auskommentieren folgender Zeilen:

```
#if [ -e /run/systemd/system ] ; then  
#exit 0  
#fi
```

3. Führen Sie nun ein Neustart des Systems durch.

```
sudo reboot
```

Überprüfen Sie anschließend die korrekte Systemzeit und schreiben Sie diese in die RTC

```
sudo hwclock -w
```

5.2 FAQ – Frequently Asked Questions

❖ Welche Vorteile bringen die S.USV solutions Module?

- Während dem Arbeitsprozess auf den Einplatinencomputern werden Betriebssystem und Benutzerdaten kontinuierlich auf die SD-Karte geschrieben. Ein plötzlicher Ausfall oder eine Unterbrechung der Stromversorgung während des Schreibens von Dateien kann zu Verlust und/oder Korruption von wichtigen Daten führen. Die S.USV solutions überbrückt diese Spannungsschwankungen und verleiht den Einplatinencomputern mehr Stabilität in der Versorgung durch die unterbrechungsfreie Stromversorgung. Des Weiteren verfügen die S.USV solutions über weitere nützliche Zusatzfunktionen.

❖ Sind die S.USV solutions Module vollständig Plug & Play?

- Die S.USV solutions Module sind intelligente Plug & Play Lösungen. Um die Hauptfunktion der unterbrechungsfreien Stromversorgung zu verwenden, sind keine weiteren Konfigurationen notwendig. Die Schnittstelle und Kommunikation zwischen Einplatinencomputer und S.USV werden automatisch über den HAT EEPROM konfiguriert. Für die Überwachung und Steuerung der Module steht ein entsprechender S.USV Client und Daemon zur Verfügung.

❖ Mit welchen Einplatinencomputer sind die S.USV solutions Module kompatibel?

- Die S.USV solutions Reihe besteht aktuell aus drei unterschiedlichen Ausführungen:
 - S.USV pi (Kompatibel mit allen Raspberry Pi Modellen)
 - S.USV UPs (Kompatibel mit allen AAEON UP Boards)
 - S.USV Tinker (Kompatibel mit allen ASUS Tinker Boards)

❖ Wie erfolgt die Spannungsversorgung der S.USV?

- Die Stromversorgung der S.USV solutions erfolgt direkt über die GPIO Power Pins der Einplatinencomputer und nutzt somit eine gemeinsame Spannungsquelle, dadurch sind keine weiteren Verkabelungen oder Stromversorgungen notwendig. Die „Advanced“ Variante bietet außerdem einen Netzeingang für den erweiterten Spannungsbereich von 7 -24 Volt (Solarzellen, KFZ – Bereich, etc.).

❖ Welche Spannungsversorgungen können über den externen Eingang angeschlossen werden?

- Es können alle Spannungsquellen mit einem Spannungsbereich zwischen 7-24V und einem Mindestausgangsstrom von 2A verwendet werden.

❖ Welche Schutzvorkehrungen wurden bezüglich der Spannungsversorgung implementiert?

- Um die externe Spannungsversorgung, sowie das S.USV Modul und Einplatinencomputer vor Beschädigungen zu schützen, wurden sowohl auf primärer, sowie auf sekundärer Seite folgende Vorkehrungen getroffen:
 - ZVD - Circuit
 - TVS - Circuit
 - PPTC Fuse
 - Power Monitoring System

❖ Welche Ausgangsleistung stellen die S.USV solutions Module zur Verfügung?

- Die S.USV solutions Module stellen folgende Ausgangsleistung zur Verfügung:
 - Primary: 5V / 3.5A (+/- 5%)
 - Secondary: 5V / 3.5A (+/- 5%)

❖ Welche GPIO-Pins werden durch die S.USV belegt?

- 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, I²C) – I²C Datenleitung
 - **Pin #05:** GPIO 03 (SCL1, I²C) – I²C Taktleitung
 - **Pin #13:** GPIO 27 (GPIO_GEN2) – Zustandsüberwachung S.USV
 - **Pin #27:** ID_SD (I²C ID EEPROM) – ID Datenleitung
 - **Pin #28:** ID_SC (I²C ID EEPROM) – ID Taktleitung

❖ Wie erfolgt die automatische Umschaltung von primären in den sekundären Versorgungsmodus?

- Die Eingangsspannung des Gesamtsystems wird zyklisch durch die Software überwacht und mit einer intern konfigurierten Spannungsreferenz verglichen. Um die Versorgungsspezifikationen von 5V +/- 5% einzuhalten, wird bei Spannungseinbrüchen <4,75V automatisch in den sekundären Modus (Batteriebetrieb) geschaltet. Bei Wiederkehr der primären Stromversorgung schaltet die S.USV automatisch in den primären Modus zurück.

❖ **Welche Art von Akkus werden mit den S.USV solutions Modulen verwendet?**

- Es werden speziell für das System konzipierte Lithium-Polymer Akkumulatoren verwendet. Jedem Produkt liegt ein 300mAh Akku bei, optional kann ein Akku mit höherer Kapazität von 3000mAh für erhöhte Überbrückungszeiten über unseren Online-Shop bezogen werden.

Für weitere Daten bezüglich der Akku-Charakteristik sehen Sie S.4 – Kapitel 2 Technische Spezifikationen. Wir raten generell davon ab, Akkumulatoren mit abweichender Charakteristik zu verwenden.

❖ **Können auch Akkus mit höherer Kapazität verwendet werden?**

- Es besteht generell kein Kapazitätslimit bezüglich der Verwendung von Lithium-Polymer oder Lithium-Ionen Akkumulatoren. Positiv getestet wurde die aktuelle Revision 2.0 bis zu einer Kapazität von 10.000mAh.

Für weitere Daten bezüglich der Akku-Charakteristik sehen Sie S. 4 – Kapitel 2 Technische Spezifikationen. Wir raten generell davon ab, Akkumulatoren mit abweichender Charakteristik zu verwenden.

❖ **Wie funktioniert die Akku – Ladeschaltung auf den S.USV solutions Modulen?**

- Bei allen S.USV solutions Produkten wurde ein intelligentes Battery Management System implementiert. Die Ladeschaltung wurde speziell auf die Verwendung von Lithium-Polymer und Lithium-Ionen Akkumulatoren konzipiert. Die Ladeschaltung teilt die Ladung der Akkus in vier voneinander gesteuerten Ladezyklen um den Akku schonend zu laden und die Lebensdauer dadurch zu erhöhen. Die durch uns angebotenen Akkus können bis zu einer Ladestromstärke von 1A belastet werden, die Ladestromstärke kann der Benutzer über die Client Software konfigurieren (300, 500 und 1000mA).

❖ **Welche Schutzvorkehrungen wurden bezüglich der Spannungsversorgung durch die Akkus implementiert?**

- Durch das intelligente Battery Management System, sowie den Protection Circuit Modulen auf den von uns angebotenen Akkus, sind diese sowohl über die Software, sowie über die Hardware vor Beschädigung (Überspannung, Tiefenentladung und Überhitzung) gesichert. Bei einer Restkapazität von <25% wird dies durch die Statusindikatoren signalisiert, bei einer Restkapazität von <10% wird das System in den Shutdown-Prozess geführt, um eine Tiefenentladung zu vermeiden. Im sekundären Modus (Batteriebetrieb) wird die Ladeschaltung automatisch deaktiviert.

Software und Hardware arbeiten in diesem Szenario unabhängig voneinander, um die Akkumulatoren in jedem Falle vor Beschädigungen zu schützen.

❖ **Können die S.USV solutions Module auch in Verbindung mit weiteren HATs betrieben werden?**

- Ein Hauptfokus der S.USV solutions Reihe liegt in der Modularität und Variabilität des Systems. Alle GPIO-Pins sind 1:1 verbunden und können frei weiterverwendet werden, lediglich GPIO Pin 13 (GPIO27) ist für die Zustandsüberwachung durch die S.USV reserviert.

Positiv getestet wurden bisher folgende Produktpaletten:

- HiFi Berry
- FHEM
- HomeMatic
- EnOcean
- RPI WWCAM

❖ **Wie funktioniert der automatische Shutdown?**

- Beim Automatischen Shutdown der S.USV wird kontinuierlich die gemessene Eingangsspannung mit einem Referenzwert (Spannungsschwelle) verglichen. Fällt die Eingangsspannung unter den intern konfigurierten Schwellwert von 4,75V, schaltet die S.USV automatisch in den sekundären Modus (Batteriebetrieb). Der resultierende Shutdown-Prozess wird in diesem Modus je nach konfiguriertem Shutdown-Timer abgehandelt.

Für weitere Informationen und Konfigurationsmöglichkeiten folgen Sie bitte S. 17 - Kapitel 4.1.1 Daemon Konfiguration.

❖ **Folgt nach Rückkehr der Stromversorgung ein automatischer Neustart des Systems?**

- Eine Wiederkehr der primären Spannung führt im ausgeschalteten Zustand des Systems unmittelbar zu einem Neustart. Das System geht in den Reboot-Prozess. Auch während des Shutdown-Prozess führt eine Wiederkehr der primären Spannung zu einem direkten Reboot des Systems nach Beendigung des Shutdown Vorgangs.

❖ **Wie funktioniert das zeitgesteuerte Ein- Ausschalten der S.USV solutions Module?**

- Durch den Benutzer kann die Funktion des zeitgesteuerten Ein- Ausschaltens des Gesamtsystems aktiviert und konfiguriert werden. Nach erfolgreicher Konfiguration kann das System zur konfigurierten Shutdown Time in den Ruhezustand geführt werden. Die integrierte Real Time Clock wird in diesem Zustand durch entsprechende Firmwareroutinen ausgewertet und je nach konfiguierter Boot-Time darauf reagiert. Bei Übereinstimmung der konfigurierten Uhrzeit fährt das System automatisch in den Bootvorgang.

Für weitere Informationen und Konfigurationsmöglichkeiten folgen Sie bitte S. 20 – Kapitel 4.2.1 Client Optionen

❖ **Besteht die Möglichkeit einer vorkonfigurierten Mail Notifikation / Alarmierung?**

- Es besteht die Möglichkeit entsprechende Mail Notifikation durch die S.USV zu aktivieren. Hierfür speichert der Benutzer seine Maildaten in ein vordefiniertes Python Script, welches bei Verlust der primären Versorgungsspannung versandt wird, um den Kunden über den Ausfall der Versorgungsquelle zu informieren.

Für weitere Informationen und Konfigurationsmöglichkeiten folgen Sie bitte S. 20 – Kapitel 4.2.1 Client Optionen.

❖ **Besteht die Möglichkeit von vorkonfigurierten User Shutdown Scripts?**

- Über die Config der S.USV Module (susv.cfg in /opt/susvd) können entsprechende User Shutdown Scripts durch den Kunden angelegt werden, welche bei Einleitung des Shutdown-Prozess ausgeführt werden.

Für weitere Informationen und Konfigurationsmöglichkeiten folgen Sie bitte S. 20 – Kapitel 4.2.1 Client Optionen.

❖ **Wie erfolgt die Kommunikation zwischen S.USV und Einplatinencomputer?**

- Die Kommunikation zwischen S.USV und Einplatinencomputer erfolgt über die I2C-Schnittstelle (3.3V Pegel). Die S.USV Module arbeiten als Slaves, somit kann der I2C-Bus problemlos über weitere Peripherie weiterverwendet werden.

Alle verfügbaren Daten wie Eingangsspannung, Stromverbrauch oder Restkapazität des Akkus können über entsprechende I2C-Register oder direkt über die vorkonfigurierten Client-Befehle abgerufen weiterverarbeitet werden. Dadurch kann der Benutzer unabhängig auf unterschiedliche Ereignisse reagieren.

Für weitere Informationen oder Konfigurationsmöglichkeiten folgen Sie bitte S. 14 – Kapitel 3.2.5.1 Register und S. 4.2.1 Client Optionen.

5.3 Troubleshooting

❖ HW/SW Kompatibilität

- Bitte beachten Sie die entsprechenden Kompatibilitäten zwischen Hardware- und Softwarerevision.
Die nicht gegebene Funktionalität, sowie die fehlerhaften Parameter resultieren ggf. aus entsprechender Inkompatibilität.

Bitte verwenden Sie für die Hardware-Revision 1.X lediglich die 1.X Firmware- sowie Softwarerevisionen. Dies gilt ebenso für die aktuelle Hardware-Revision 2.X und den damit verbundenen Softwarerevisionen.

❖ I2C-Request timed out – I2C-Adresswechsel

- Bitte überprüfen Sie zunächst über den Befehl `sudo i2cdetect -a -y 1` den gesamten I2C-Bus, unter Umständen wurde die I2C-Adresse der S.USV auf eine andere Adresse verlegt (z.B. 0x01, 0x00).
- In diesem Fall übernehmen Sie bitte die Adresse manuell in die im Verzeichnis `/opt/susvd` befindliche `susv.cfg` und führen Sie einen Restart des Daemon Prozess durch
- Sofern die I2C-Kommunikation hierüber wieder aktiviert werden konnte, können Sie die I2C-Adresse wieder auf die ursprüngliche 0x0f zurücksetzen.

❖ S.USV schaltet nicht vollständig ab

- Die Problematik deutet auf eine fehlerhafte Konfiguration des S.USV Daemon Prozess.

Bitte überprüfen Sie zunächst ob dieser korrekt arbeitet – Wechseln Sie hierzu in das Verzeichnis `/opt/susvd` und führen Sie folgenden Befehl aus **`sudo ./susvd -status`**

❖ S.USV pi basic schaltet unerwartet in sekundären Modus – Boot Cycle

- Bitte beachten Sie, dass die Versorgung diverser Peripherie und aktivierter Ladeschaltung über den microUSB Eingang des Einplatinencomputers zu Spannungseinbrüchen an den GPIO Power Pins führen kann. Die S.USV schaltet bei einer Eingangsspannung $<4,75V$ automatisch in den sekundären Modus und mit aktiviertem Daemon Prozess in den Shutdown.

Gehen Sie bitte wie folgt vor, um die Problematik zu überprüfen:

- Herunterfahren des Systems
- Entfernen des S.USV solutions Modul
- Reboot des Systems und Beenden des S.USV Daemon Prozess
- Neustart des Systems mit S.USV solutions Modul
- Optional können Sie nun die Eingangsspannung direkt an den GPIO Pins messen

In den meisten Fällen konnte die Problematik durch ein leistungstärkeres Netzteil mit einer Ausgangsspannung von 5,25V behoben werden. Für Systeme mit erhöhtem Leistungsverbrauch empfehlen wir die Stromversorgung über den externen Eingang der S.USV zu realisieren, welcher bis zu 5A liefern kann.

❖ **S.USV ignoriert Wiederkehr der Stromversorgung:**

- Im sekundären Modus (Batteriebetrieb) der S.USV werden kontinuierlich Stromwerte ermittelt und in entsprechenden Firmwareroutinen weiterverarbeitet. Dadurch sind die Module in der Lage eine Wiederkehr der primären Stromversorgung, durch die daraus resultierend signifikante Reduzierung des Stromverbrauchs über den sekundären Eingang, zu erkennen. Die intelligente Software erkennt und ermittelt die Leistungsfähigkeit der primären Stromquelle und reagiert entsprechend darauf.

Sollte das S.USV Modul im sekundären Modus nach Wiederkehr der Stromquelle nicht zurück in den primären Modus schalten, empfehlen wir das Szenario mit leistungsfähigeren Netzteilen zu testen.

❖ **Shutdown/Reboot Button S1 zeigt keine Reaktion auf Betätigung**

- Für die Funktion des S1 Shutdown/Reboot Buttons ist ein korrektes Arbeiten des S.USV Daemon Prozess notwendig, welcher die entsprechenden Signale weiterverarbeitet.

Bitte überprüfen Sie zunächst ob dieser korrekt arbeitet – Wechseln Sie hierzu in das Verzeichnis ***/opt/susvd*** und führen Sie folgenden Befehl aus ***sudo ./susvd -status***

