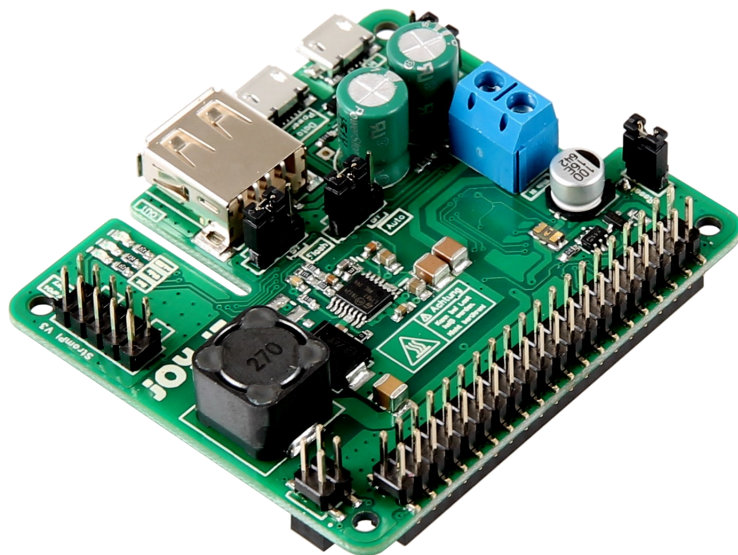


joy-it



StromPi 3

TABLE OF CONTENTS

1. Introduction
2. Overview
3. Current sources
4. Functions of the StromPi 3
 - 4.1 Prioritization of input
 - 4.2 Time-controlled applications
 - 4.2.1 Real time clock
 - 4.2.2 Automatic configuration of time
 - 4.2.3 ShutDown timer
 - 4.2.4 Alarm mode
 - 4.2.5 PowerOff alarm
 - 4.2.6 WakeUp alarm
 - 4.2.7 Minute Wakeup Alarm
 - 4.2.8 Interval function
5. Additional functions
 - 5.1 Serialless mode
 - 5.2 Example of usage with real-time clock
 - 5.3 PowerOn button
 - 5.4 PowerOff mode
 - 5.5 Powersave mode
6. Configuration of StromPi 3
 - 6.1 Activation of serial communication
 - 6.2 Execution of the GUI script
 - 6.3 VNC for remote desktop connection
 - 6.4 SSH-Connection with PuTTY and Xming
 - 6.5 SSH-Connection with MobaXterm
7. StromPi 3 scripts
 - 7.1 Serial scripts
 - 7.2 Serialless scripts
 - 7.3 Raspberry Pi Pin-Layout
 - 7.4 Automatic execution of scripts
8. Modifications
9. Opening of the serial console of the Strompi 3
10. console commands
11. Support
12. EU-Declaration of conformity

1. INTRODUCTION

Dear customer,

Thank you for buying our product. The structure and functions of the StromPi3 are explained in the following chapters. We show you which settings must be adjusted that the Raspberry Pi will work with the StromPi perfectly.

If you get any difficulties with the usage with the device, do not hesitate to contact us.

The StromPi3 enables the Raspberry Pi® and other single-board computer new areas of application. Additionally, the StromPi 3 has next to the same voltage inputs (mUSB, WideRange 6-61V) as the StromPi2 an optional plug-in unit for batteries. This expands the Raspberry with a LiFePO4 battery which is a rechargeable emergency power source. Unexpected power failures are now in the past.

The prioritized inputs can easily be changed which means the StromPi 3 is able to fit in your project perfectly. With the voltage range from 6V to 61V and a current with up to 3A, there is enough power for bigger projects.

In addition the StromPi 3 has an own configurable autonomously acting microsystem unit which can be used for a programmable start-stop behaviour:

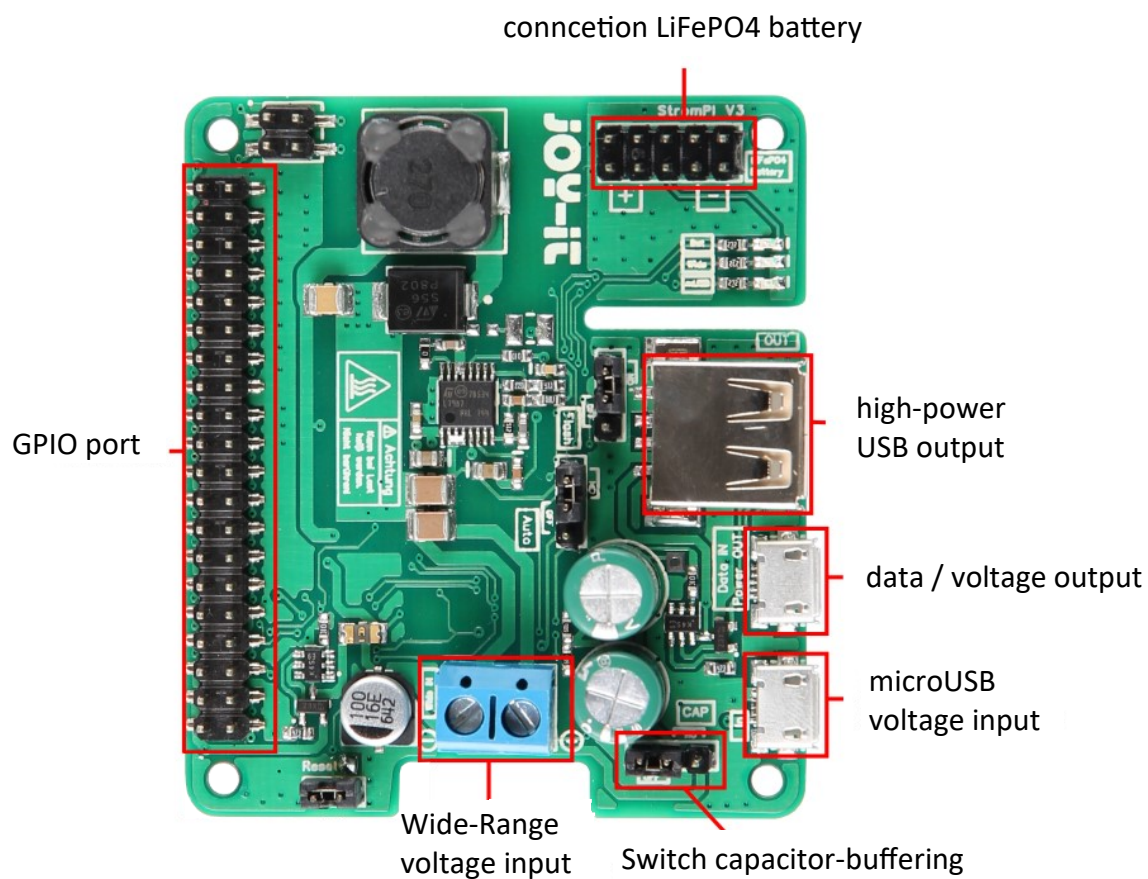
Let your system boot and shutdown at dictated times to do measurements or to switch devices on and off (ideal for digital signals or for controlling machines). The included real time clock can be used to synchronize the time with the Raspberry Pi without internet access. There is also an expansion to control the serial interface (state battery charge, voltage of the in-/ outputs, control and configuration) this makes the StromPi3 to an absolute allrounder.

The StromPi 3 suits every application area you choose perfectly.



This Manual is made for the firmware version 1.71. We recommend upgrading the firmware of the StromPi regularly. You can download the newest version [here](#). There is also a manual to flash the firmware in the download register.

2. OVERVIEW



The StromPi 3 can heat up if it is under heavy load. Do not touch the board when it is in operation you could burn yourself. Pay attention that the device has a good air flow to prevent any damage from overheating.

3. CURRENT SOURCES

The StromPi 3 can get its power from three different current sources. In case of a power failure it switches steplessly to the current source which is configured as the second and after that to the third.

MICRO USB

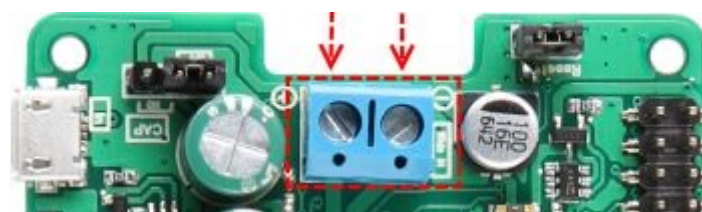
The StromPi 3 can obviously, as usual, be powered via the microUSB port.



WIDE-RANGE

The Wide mode can be used with a current source from 6V - 61V. Connect the marked port to an arbitrary current source, as power supply or a vehicle electrical system.

Attention! The current source must be switched on within the range of 6 - 61 V. Please pay attention to the polarity to avoid damaging the Raspberry Pi or StromPi3.



BATTERY-UNIT

The StromPi 3 can be expanded with an optional 1000mAh LiFe04 battery, which is a rechargeable emergency power source. You only have to plug in the separate available battery unit, which then can be configured as the second or third current source. Because of the implemented GPIO-bar, you do not have any restrictions with the usage of the battery unit.

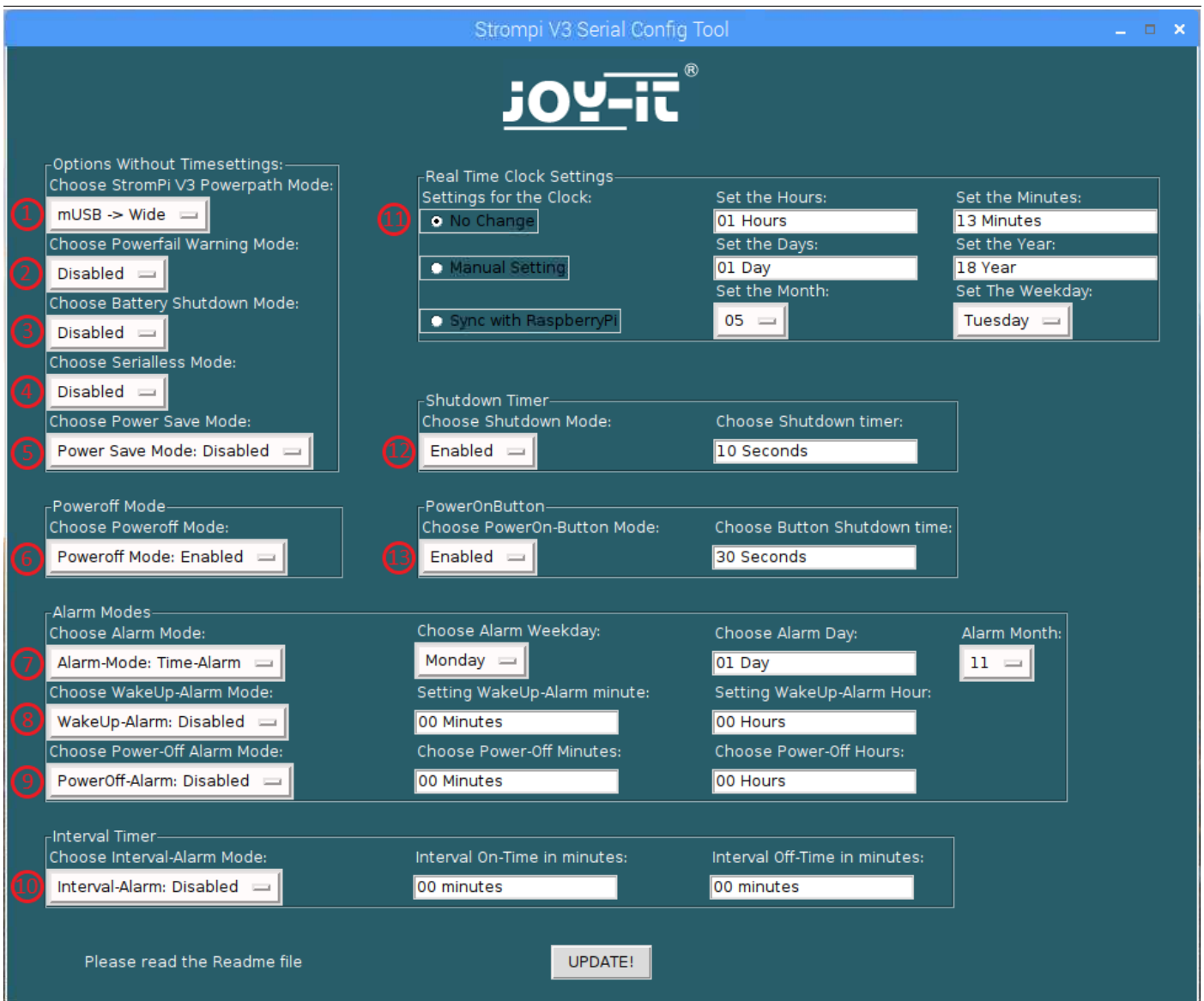


4. FUNCTIONS OF THE STROMPI 3

The StromPi 3 has a huge amount of configuration possibilities. You can configure this device after your desires like for example the prioritization of input, optional shutting down as well as the WakeUp and PowerOff mode. More functions can be setup with our graphical user interface script.

To use all of the functions that the StromPi provides you have to use our Python scripts. These are listed and explained in chapter 7 "StromPi 3 scripts".

4. FUNKTIONEN DES STROMPI 3



- 1: Choose StromPi mode.
- 2: Activate or deactivate the powerfail warning
- 3: Setup of the BatteryShutdown mode
- 4: Activate or deactivate the Serialless mode
- 5: Activate or deactivate the Power Save mode
- 6: Activate or deactivate the Poweroff mode
- 7: Setup of the Alarm mode
- 8: Setup of the Wakeup alarms
- 9: Setup of the PowerOff alarms
- 10: Setup of the Interval alarms
- 11: Setup of the real time clock
- 12: Activate or deactivate the Shutdown timers and setup the time (in seconds)
- 13: Activate or deactivate the PowerOn buttons and setup the time (in seconds)



To use all the settings which are described in the following chapters you have to use the serial connection. In chapter 6 is an explanation how to configure a serial interface for the Raspberry Pi.

4. FUNCTIONS OF STROMPI 3

StromPi mode:

The StromPi mode determines the prioritization of the separate current sources.

Powerfail warning:

The PowerFail warning will show when the primary power source is failing.

Battery shutdown mode:

The battery shutdown mode determines with which remaining battery level of the RaspberryPi the device will shut down.

Serialless mode:

The serialless mode allows a further usage of the serial ports by communication on a GPIO pin. [modification required! Look at chapter 8 "Modifications"]

Power save mode:

The power save mode will reduce the power consumption if Wide is set up as the secondary power source. However, this is only recommended when the device is in operation with a low amperage (<2,5A / for example an Raspberry Pi without periphery).

Real time clock settings:

In the RTC settings you can setup the time and the date of the real time clock manually or automatically with comparing and synchronizing the time with the Raspberry Pi.

Shutdown timer:

If you have activated the shutdown mode and the StromPi detects a power failure at the primary power source the StromPi3 will send a signal to the raspberry through the serial interface to start the shutdown. After that the counter starts that will disconnect the powersupply to the raspberry.

PowerOn button:

With the PowerOnButton you can start the RaspberryPi if it is in the poweroff state through short-circuiting the two reset pins. [modification needed! See chapter 8 "modifications"]

Interval timer:

The interval timer can be setup to shutdown or boot the system after a selected time.

Alarm mode:

The alarm mode refers to the WakeUp and PowerOff alarm. It selects if the alarm goes of daily, weekly or only at a specific date.

WakeUp alarm:

The WakeUp alarms start the Raspberry Pi at a specific time.

PowerOff alarm:

The PowerOff alarms shut the Raspberry Pi down at a specific time.

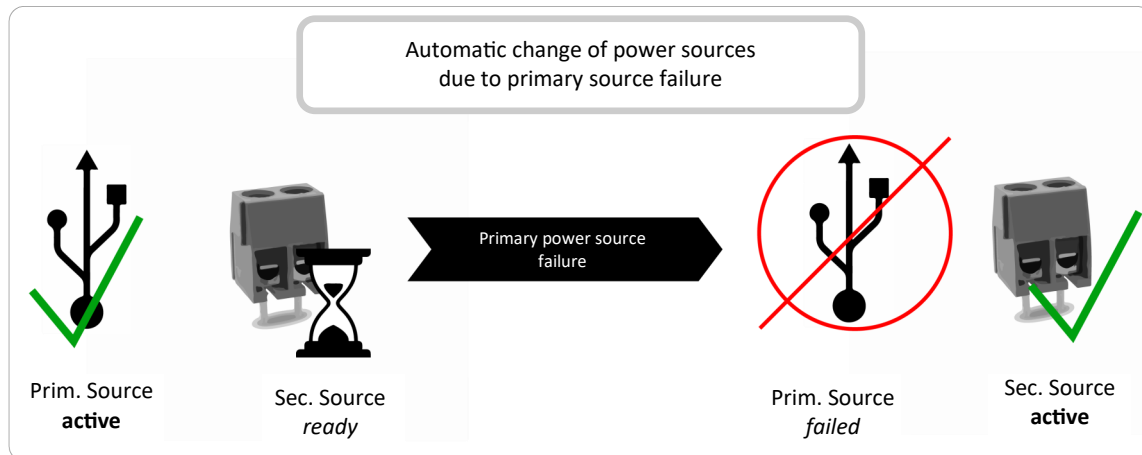
Poweroff mode:

The Poweroff mode requires the activation of the RaspberryPi with the PowerOn button after a shutdown through the Shutdown timer. Additionally, this mode prevents the detection of a reversion of the primary power supply.

4.1 PRIORITIZATION OF INPUT

The power source where the StromPi draws its power from can be set up as desired. In addition, you can select a second and a third power source which will take over if the first source fails.

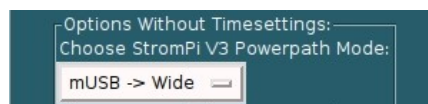
In the default settings microUSB is the primary power source and Wide is the secondary.



In total there are six different modes possible. The primary power source is the main power source. The secondary power source will take over when the primary source fails.

MODE	PRIMARY	SECONDARY	TERTIARY
Mode 1	microUSB	Wide-Range	-
Mode 2	Wide-Range	microUSB	-
Mode 3	microUSB	Battery	-
Mode 4	Wide-Range	Battery	-
Mode 5 (V1.6 or higher)	microUSB	Wide-Range	Battery
Mode 6 (V1.6 or higher)	Wide-Range	microUSB	Battery

The corresponding mode can be setup with the configscript in “choose StromPi V3 Powerpath Mode”.



4.2 TIME-CONTROLLED APPLICATIONS

4.2.1 REAL TIME CLOCK



The StromPi 3 has a additional real time clock.

If the power fails, the time-controlled applications are not endangered. Even time-controlled shutdowns and boots are still possible!

Please note that you have to use atleast one power source connected (mUSB, Wide or battery) - If all sources are seperated, the current time will not be saved.

The real time clock (RTC) of the StromPi 3 can be setup with the Configscript:

ADJUST TIME AND DATE MANUALLY

To adjust time and date manually click on **Manual Setting**. Now you can fill the fields in the right. If you click on Update it will set your setting.

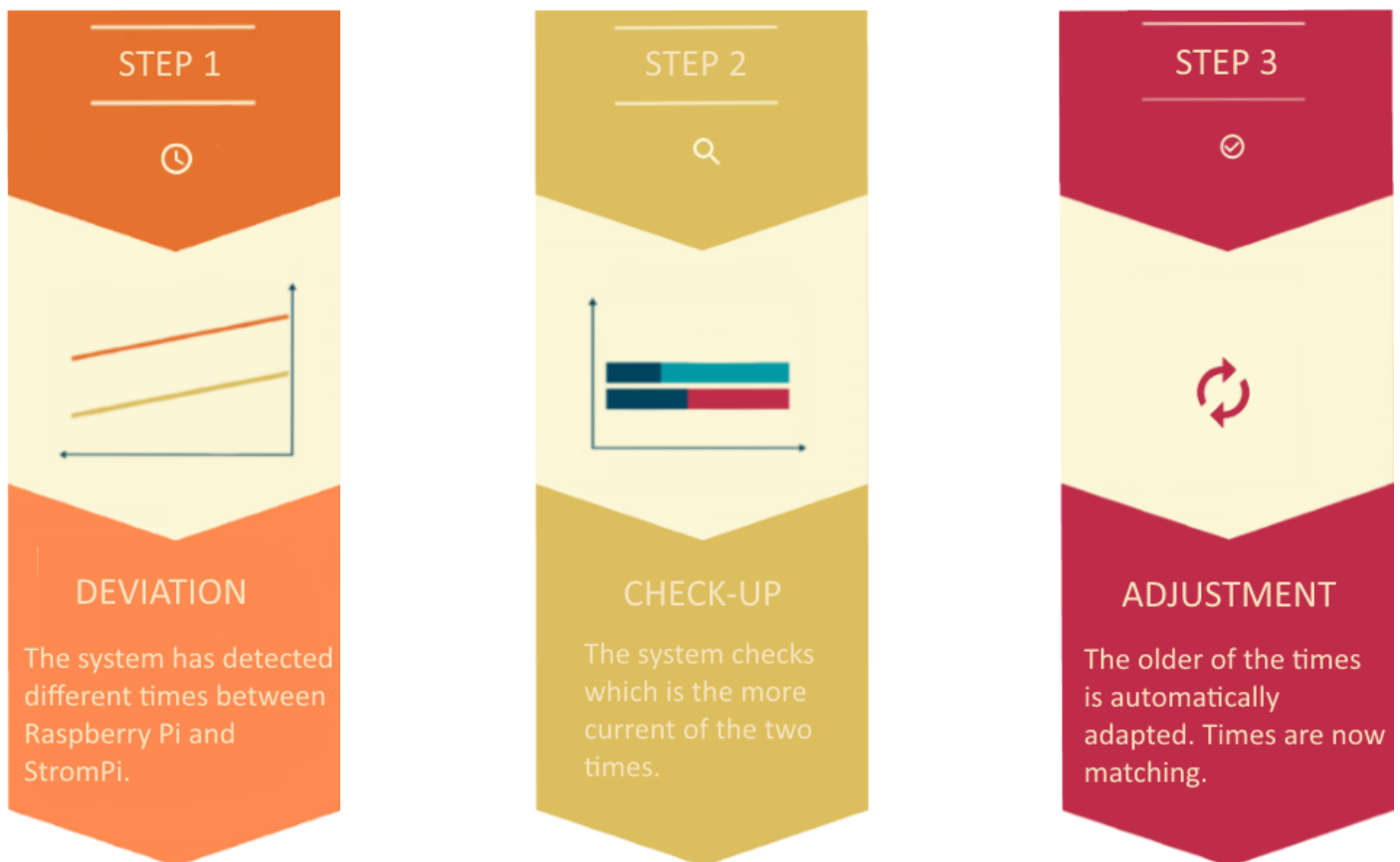
SYNCHRONIZING TIME AND DATE WITH RASPBERRY PI

To synchronize time and date with the Raspberry click on **Sync with Raspberry Pi**. With the click on Update the time from the RTC of the StromPi will be compared to the time on the Raspberry. The newer time will be overwritten with the current one. The default time of the StromPi is the 01.May 2018 and the Raspberry Pi updates its time automatically if it is connected to the internet.

4.2.2 AUTOMATIC CONFIGURATION OF TIME

In addition we provide a script which synchronizes the time after the boot of the Raspberry. This script can be downloaded [here](#) in the download area in section „Scripts“. Download the archive, unzip and copy the script to your Raspberry Pi.

In chapter 7.4 “Automatic execution of scripts” you can read how to add scripts to the autostart that they perform automatically after the boot.

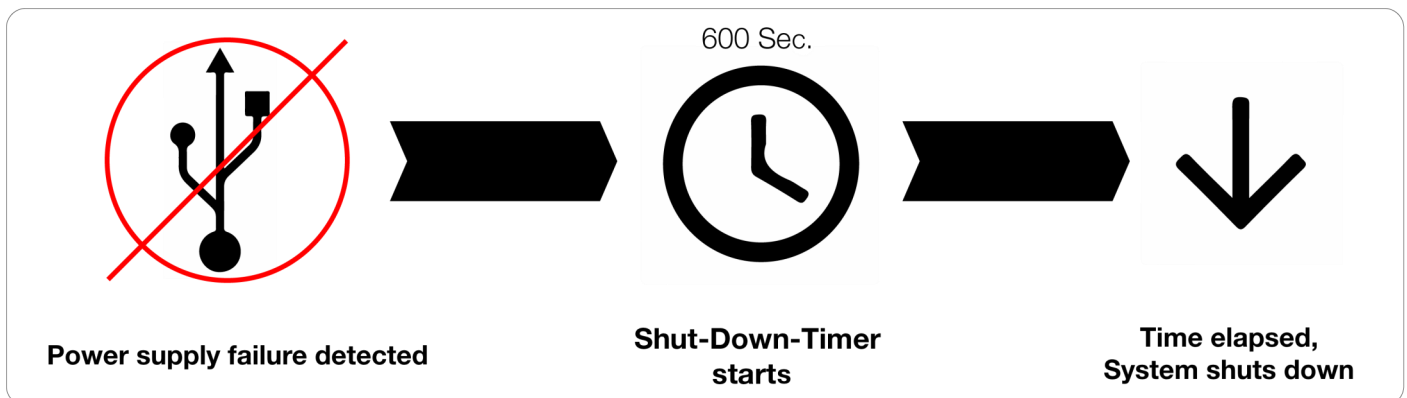


4.2.3 SHUTDOWN-TIMER

The StromPi 3 has a Shutdown timer which cuts the power supply to the Raspberry Pi automatically if the primary power source fails.

You can change the time that will pass before the StromPi cuts the power to the raspberry.

If you have set the shutdown timer to 600 (600 → 10 minutes) the system reacts like:



If the StromPi 3 detects a power failure on the main power supply it will start the timer. After 10 minutes the StromPi cuts the power to the Raspberry Pi. Additionally, we offer you a script which can detect a power failure through the serial console or with a GPIO pin and begin the safe shutdown of the Raspberry Pi.

The Shutdown timer can be setup with the Configscript in point "Shutdown Timer".

The screenshot shows the "Shutdown Timer" configuration window. It contains two settings:

- Choose Shutdown Mode:** A dropdown menu currently set to "Disabled".
- Choose Shutdown timer:** A text input field containing "10 Seconds".

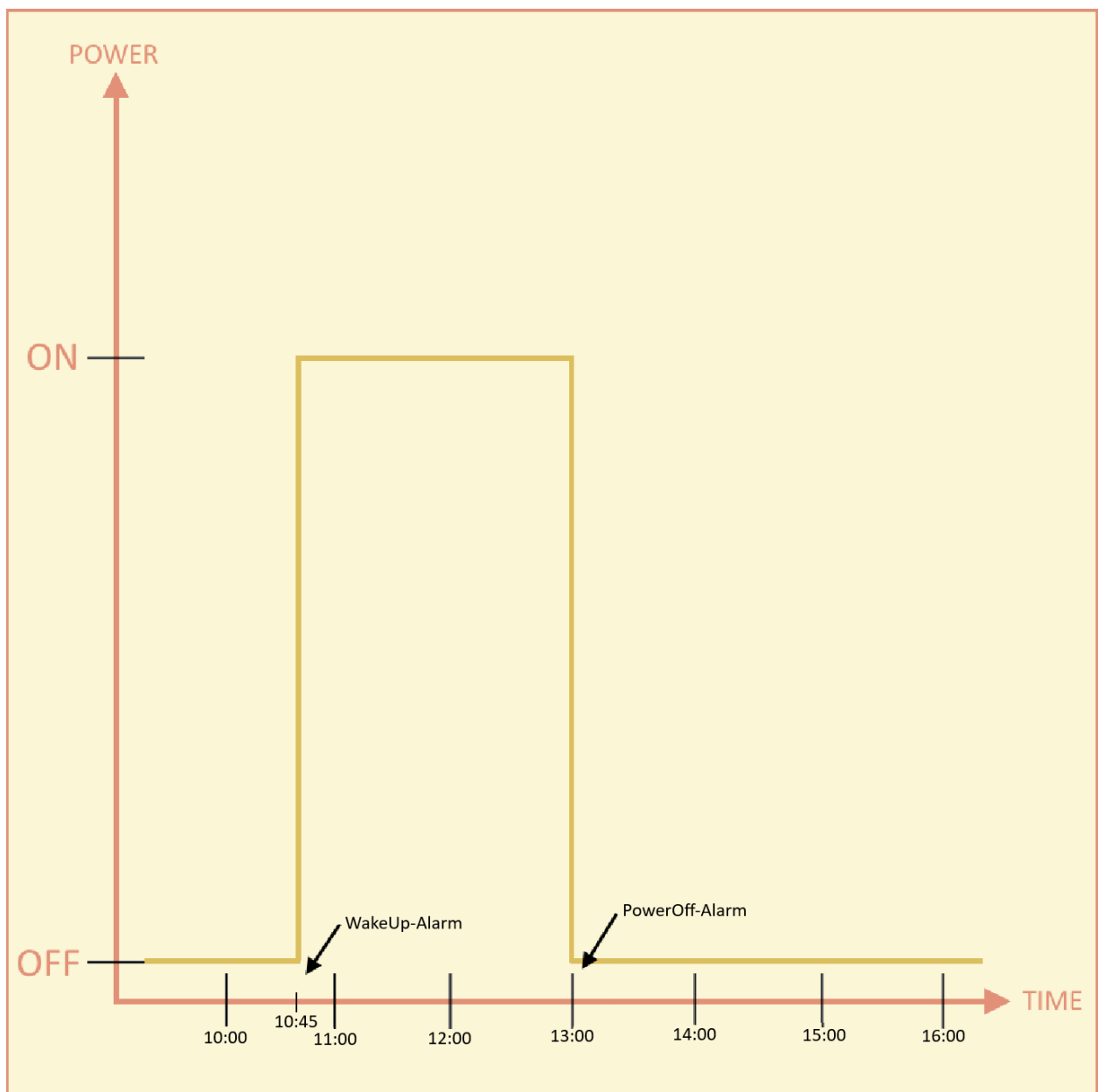
To activate the shutdown timer set "Choose Shutdown Mode" to **Enable** and write down the desired time in seconds in the field next to it.

At the end confirm with **Update**.

4.2.4 ALARM-MODE

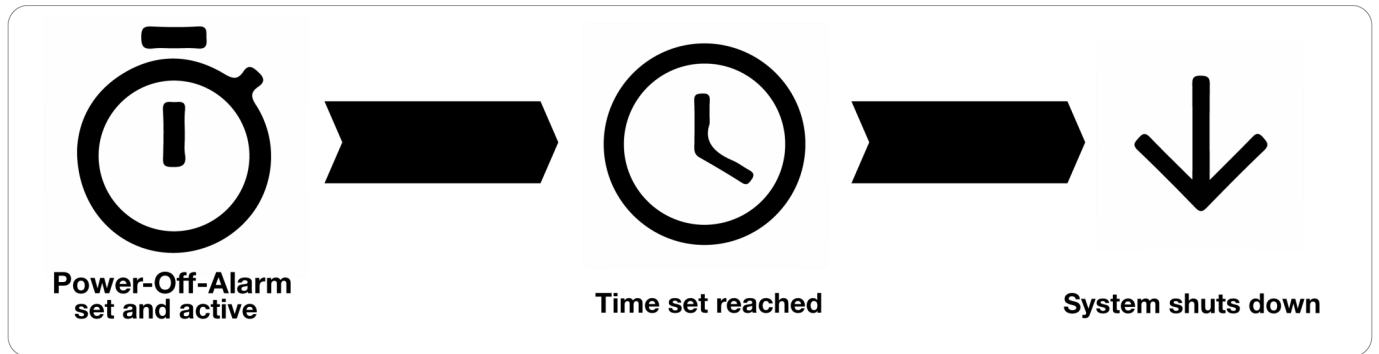
The StromPi 3 has the possibility to boot and shutdown the Raspberry time-controlled.

The combination of WakeUp and PowerOff alarm lets you configurate the start-stop-behaviour for the Raspberry Pi (for example every day at 10:45 AM is the boot and the shutdown is at 1:00PM). This way your Raspberry is only running when the occuring duties have to get processed in this time window.



4.2.5 POWER-OFF-ALARM

The PowerOff alarm allows the Raspberry to shut down at the preset time every day. Although, the Raspberry Pi can be only shutdown with this mode once a day.



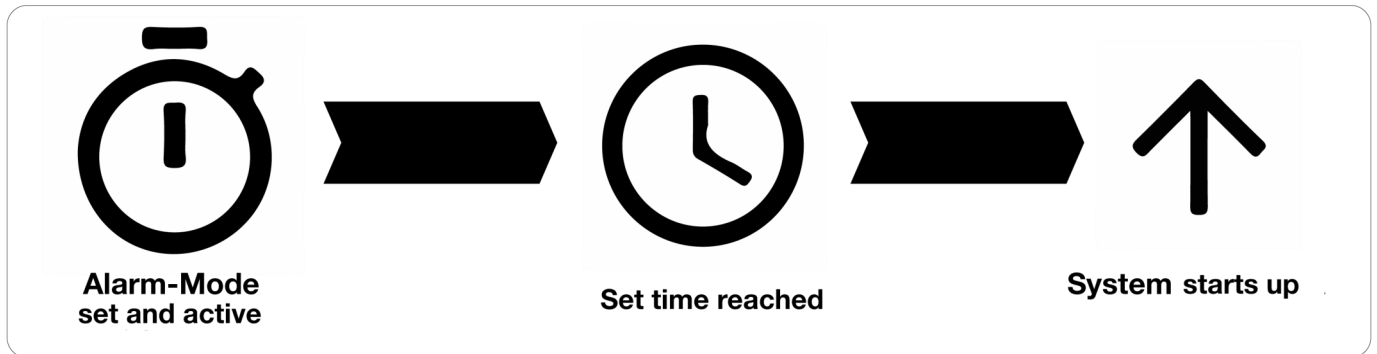
To shut down the StromPi 3 time-controlled start the Configscript:

Alarm Modes			
Choose Alarm Mode:	Choose Alarm Weekday:	Choose Alarm Day:	Alarm Month:
1 = Daily time alarm	nothing	01 Day	11
Choose WakeUp-Alarm Mode:	Setting WakeUp-Alarm minute:	Setting WakeUp-Alarm Hour:	
WakeUp-Alarm: Disabled	00 Minutes	00 Hours	
Choose Power-Off Alarm Mode:	Choose Power-Off Minutes:	Choose Power-Off Hours:	Minute Wakeup Time:
PowerOff-Alarm: Disabled	41 Minutes	00 Hours	01 Minutes

Switch the “Choose PowerOff alarm mode” to **Enable** to activate the PowerOff alarm. The time to shutdown the Raspberry can be put in the 2 boxes on the right next to it. The „Choose Power-Off Minutes“ have to be set in minutes and the „Choose Power-Off Hours“ in hours.

4.2.6 WAKE-UP-ALARM

The Wake-Up alarm allows you to start the Raspberry time-controlled. Additionally, you have the opportunity to start it at a particular date or at a particular weekday.



To start the StromPi 3 time-controlled start the Configscript:

Alarm Modes			
Choose Alarm Mode:	Choose Alarm Weekday:	Choose Alarm Day:	Alarm Month:
1 = Daily time alarm	nothing	01 Day	11
Choose WakeUp-Alarm Mode:	Setting WakeUp-Alarm minute:	Setting WakeUp-Alarm Hour:	
WakeUp-Alarm: Disabled	00 Minutes	00 Hours	
Choose Power-Off Alarm Mode:	Choose Power-Off Minutes:	Choose Power-Off Hours:	Minute Wakeup Time:
PowerOff-Alarm: Disabled	41 Minutes	00 Hours	01 Minutes

Switch the „Choose WakeUp-Alarm Mode“ to **Enable** to activate the WakeUp-alarm. The time when the Raspberry Pi has to start can be filled in the boxes on the right next to it. „Choose WakeUp-Alarm Minutes“ has to be set in minutes and „Choose WakeUp-Alarm Hours“ in hours. The Raspberry Pi is only able to start once with this mode a day.

Depending what you have setup by „Choose Alarm Mode“ the Raspberry starts daily, weekly or only to particular dates at the chosen time

Alarm mode1 starts your Raspberry every daily at the preset time.

Alarm mode2 starts your Raspberry at a particular date at the preset time.

Alarm mode3 starts your Raspberry at a particular weekday at the preset time.

Alarm mode4 activates the Minute Wakeup Alarm

The Alarm-Mode refers only to the WakeUp alarm, it does not have any influence on the PowerOff alarm.

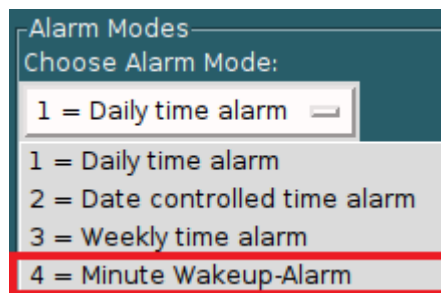


Attention! Your Raspberry Pi is only capable to start with the WakeUp-alarm, if it got shutdown with the PowerOff alarm or manually with the serial command „**poweroff**“.

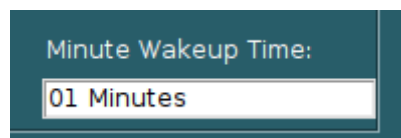
4.2.7 MINUTE WAKEUP ALARM

The Minute wakeup alarm allows you to set a timer that will start the raspberry after it got shutdown through the Poweroff Alarm or the „poweroff“ command.

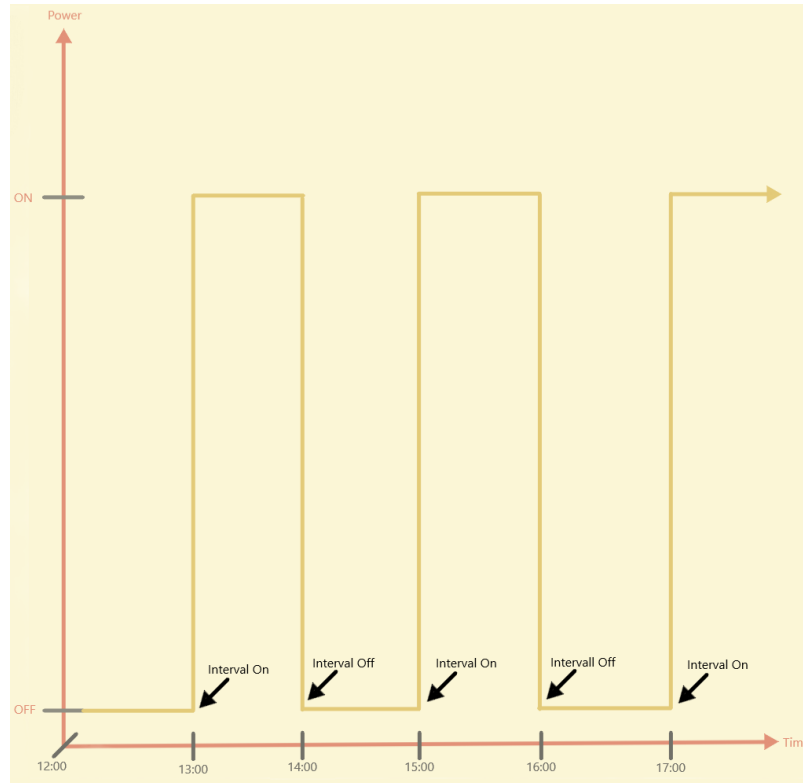
You have to activate the mode in the Configscript.
Click on „Choose Alarm Mode“ and select „4 = Minute Wakeup-Alarm“.



After that you have to choose the time (the timer is set to minutes).



4.2.8 INTERVAL FUNCTION



The interval function allows to define time intervals when the Raspberry shuts down and starts with the StromPi 3.

This function can be activated and configured with the configscript in point “interval timer”.

Interval Timer		
Choose Interval-Alarm Mode:	Interval On-Time in minutes:	Interval Off-Time in minutes:
Interval-Alarm: Disabled	00 minutes	00 minutes

Switch the “Choose Interval-Alarm Mode:” to **Enable** to activate the interval function.

The “On-Time in minutes” will determine how long the raspberry will be running.

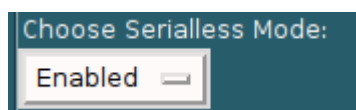
Specify in “Interval On-Time in minutes” the time how long the Raspberry Pi has to be switched off in minutes.

5.1 SERIALLESS MODE

The Serialless mode allows to suppress any serial communication of the StromPi with which you can use another HAT which uses the serial interface at the same time as the StromPi. However, it inhibits the configuration of the StromPi 3. For that reason it is advisable to configure the StromPi before activating the Serialless mode.

Additionally, the Shutdown-Warning will not occur on the serial interface as before with this mode it will occur on a free GPIO pin.

To activate the Serialless mode you have to set "Choose Serialless Mode" to **Enable**.



Then you have to run the script "Start_Serialless.py".

```
pi@raspberrypi:~/StromPi3_Scriptfolder_12062019/Serialless $ sudo python3 Start_Serialless.py
Enabled Serialless
```

If this is done, the Serialless mode is activated and the serial communication is suppressed.

To terminate the Serialless mode, you have to run the script "Stop_Serialless.py". This script sets the GPIO pin to low. If it is low for more than 5 seconds, the Serialless mode gets deactivated.

```
pi@raspberrypi:~/StromPi3_Scriptfolder_12062019/Serialless $ sudo python3 Stop_Serialless.py
Setting GPIO to LOW to Disable Serialless Mode.
This will take approx. 10 seconds.
Serialless Mode is Disabled!
```



For this mode a Serialless modification is necessary which is described in chapter 8 "Modifications".



The used scripts need Python3.

5.2 EXAMPLE OF USAGE WITH REAL-TIME CLOCK

To show you one of the many possibilities of application we created a script (RTCSerailles.py) which deactivates the Serialless mode after starting the script until the real-time clock is synchronized. After that the Serialless mode is activated again.

Here is the console output after a successful running of the script:

```
pi@raspberrypi:~/StromPi3_Scriptfolder_12062019/Serialless $ sudo python3 RTCSerailless.py
Setting GPIO to LOW to Disable Serialless Mode.
This will take approx. 10 seconds.
Serialless Mode is Disabled!
Setting Time!
Enabled Serialless
```

5.3 POWERON-BUTTON

The function of the PowerOn button allows after shut down of the StromPi with a “PowerOff“-command or with the PowerOff-alarm without separating the voltage source or without removing the BatteryHATs to restart it.

For this function is a minor modification necessary. There has to be a cable bridge between two solder pads. After this it is possible for the StromPi 3 to restart by shorting the Reset pins.

If the StromPi 3 is already modified for the Serialless mode or if the Serialless mode want to be used with the PowerOn button modification, you must perform another minor modification.

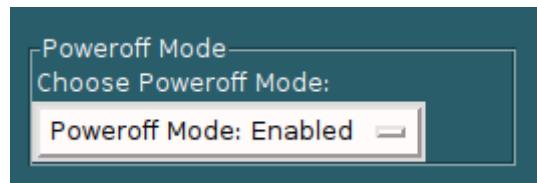
In chapter 8 “Modifications” is the explanation for the modification.



IMPORTANT: The adjusted time in the configuration script is the initialization time of the PowerOn button. We recommend a minimum time of 30 seconds

5.4 POWEROFF MODE

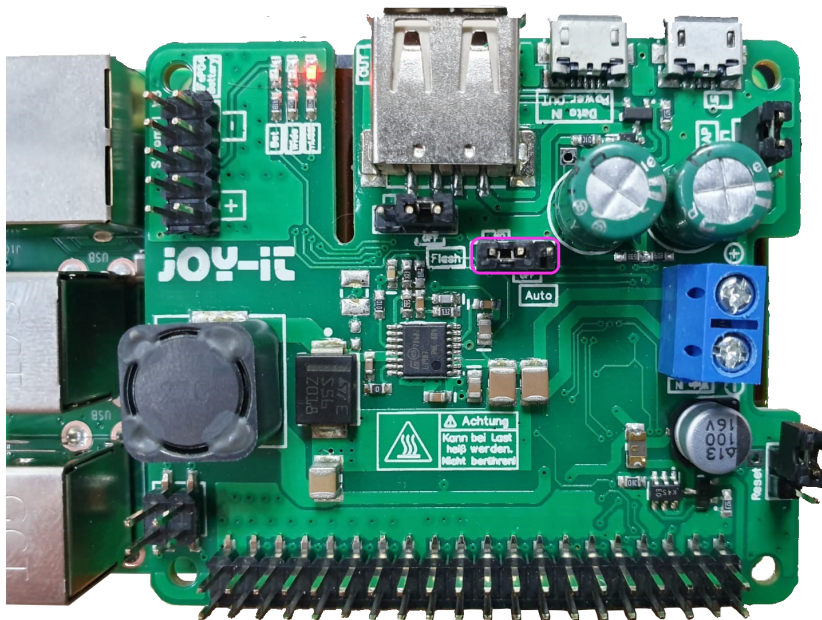
The Poweroff mode ensures that after a power failure your Raspberry remains off until you press the PowerOn button (look chapter 5.3).with this mode you lose the option to recognize a power back. The Raspberry will always shutdown and the StromPi will separate the power supply after the preset time. To activate the PowerOff mode you must setup **Enable** in the Configscript in point "Choose Poweroff Mode".



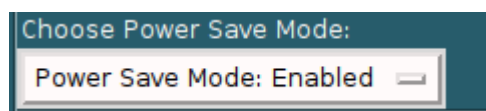
5.5 POWERSAVE MODE

The Powersave mode allows to reduce the power consumption of the StromPi 3 through a voltage transformer which will be deactivated if the Wide port is not in use and is only activated if a power failure is detected on the primary power source. This is only possible if the amperage is low. (under 2.5 ampere)

If the voltage level is too high, it is possible that the powerpath switching could fail. To Activate the PowerSave mode the shown jumper must be switched on **OFF**.



Additionally, in point "Choose Powersave Mode in the Configscript must be on **Enable**.



6. CONFIGURATION OF THE STROMPI 3

In this chapter we show you how to setup the serial communication of your StromPi. This manual is based on the Raspbian Stretch version of April 2019.

We recommend the use of the newest version of the Raspbian Stretch images.

6.1 ACTIVATION OF THE SERIAL COMMUNICATION

To use all the functions of the StromPi and change the settings you have to configure your Raspberry Pi.

First you must install the python3-serial library with the following command:

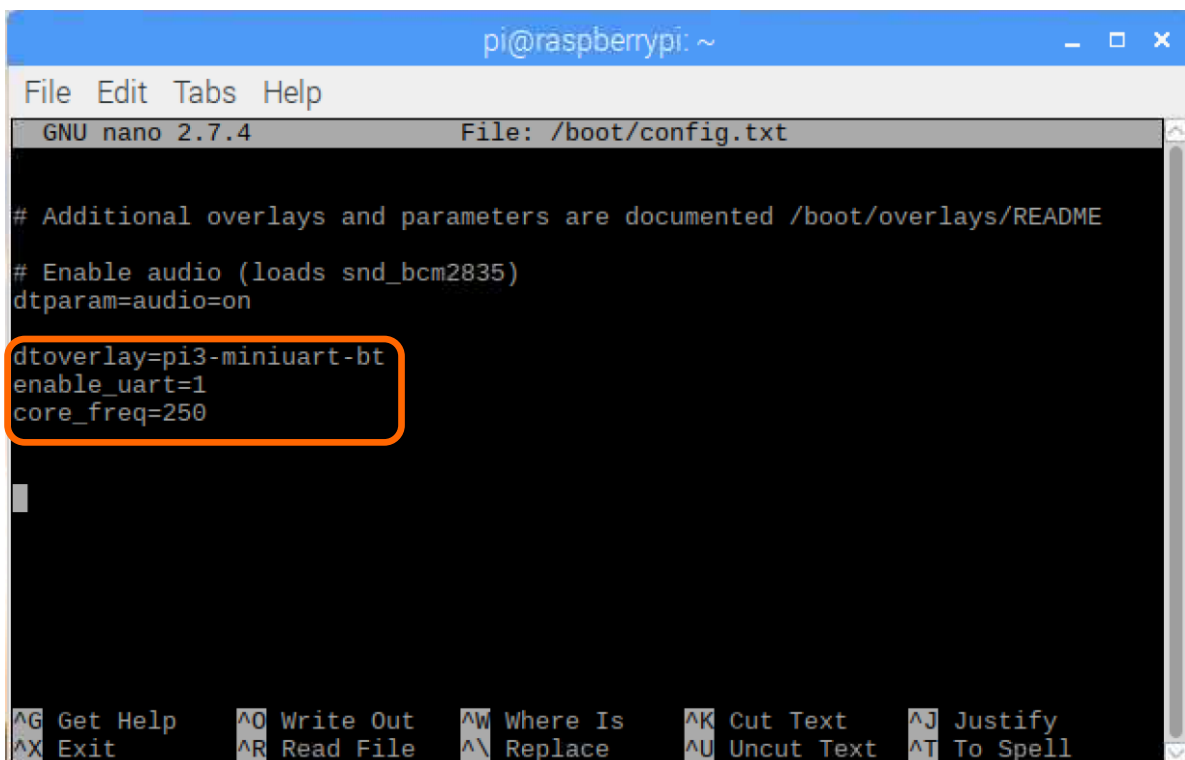
```
sudo apt-get install python-serial python3-serial
```

Open the console of the Raspberry and enter the following command:

```
sudo nano /boot/config.txt
```

Add the following lines at the end of the file:

```
dtoverlay=pi3-miniuart-bt
enable_uart=1
core_freq=250
```



Save the file with the key combination of **Strg+O** and leave the file with the combination of **Strg+X**.

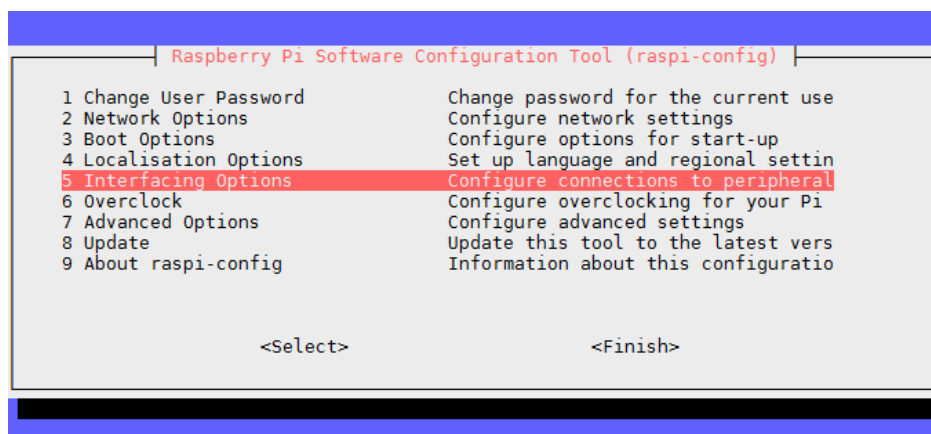
6.1 ACTIVATION OF THE SERIAL COMMUNICATION

Additionally it is necessary to activate the Raspberry Pi's serial interface and deactivate the serial console. You can do this with the Raspberry Pi configuration or with the Rasp-Config.

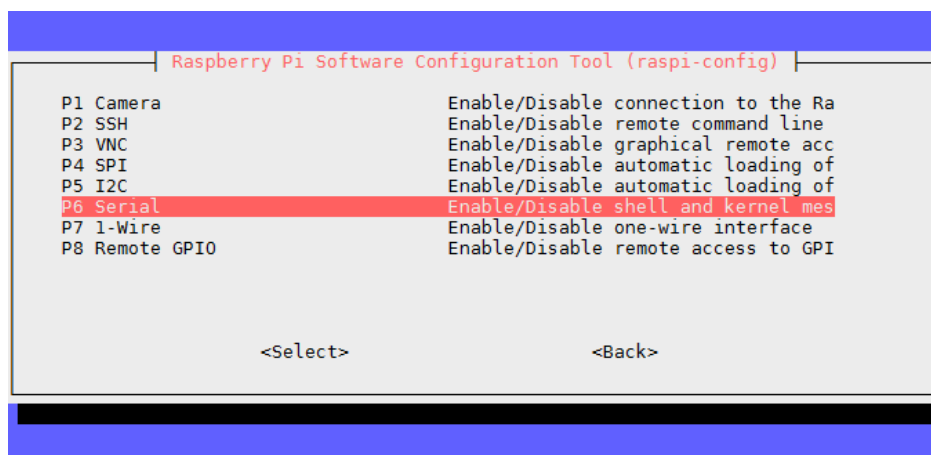
To do this with the Raspi-Config enter the following command in the terminal window at your Raspberry:

```
sudo raspi-config
```

Navigate with the arrow keys to the point **"5 Interfacing Options"**.

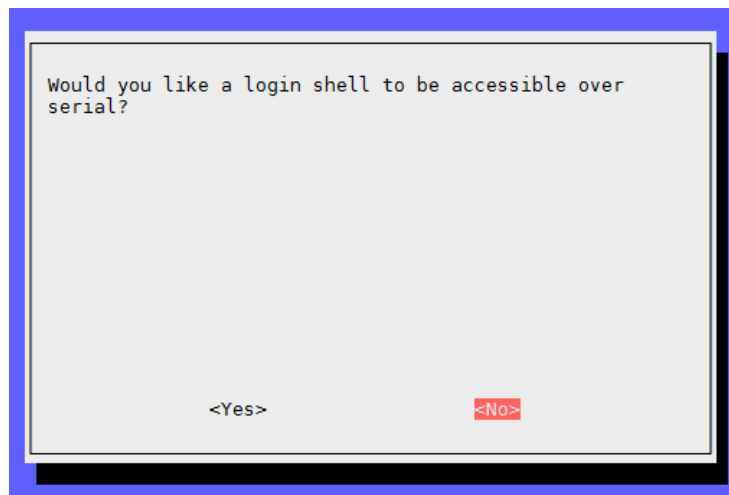


Choose point **„P6 Serial“**.



6.1 AKTIVIERUNG DER SERIELLEN KOMMUNIKATION

Answer the question **“Would you like a login shell to be accessible over serial?”** with **“No”**.



Answer the question **“Would you like the serial port hardware to be enabled?”** with **“Yes”**.



After a confirmation you can leave the menu with **“Finish”**.
The question if you want to restart the device has to be answered with **“Yes”**.
The serial interface is now activated and the serial console deactivated.

6.2 EXECUTION OF THE GUI SCRIPT



Attention! The StromPi configscript does not work with the CLI version of the StromPi firmware, only the main version of the firmware supports the script. Additionally the serial communication has to be activated [look chapter 6.1 “Activation of serial communication”]

To run the configurationscript on your Raspberry Pi you have to install a library first which you can do with the following command:

```
sudo apt-get install python3-pil python3-pil.imageTk
```

There are 3 possibilities to run the grafical interface.

6.3 VNC FOR REMOTE DESKTOP CONNECTION

For the usage of VNC you have to run the following command:

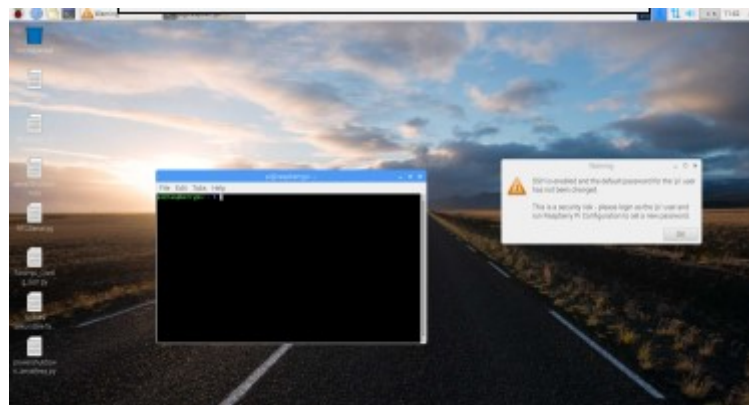
```
sudo raspi-config
```

under “Interfacing options” → “VNC” you have to set it on. After that you have to get your IP address with the following command.

```
ifconfig
```

Download RealVNC on your device from which you want to controll the Raspberry Pi and connect them through entering the IP address.

Now following window should appear:



With the console you can work as usual with your linux system.

6.4 SSH-CONNECTION WITH PUTTY AND XMING

Run the following command:

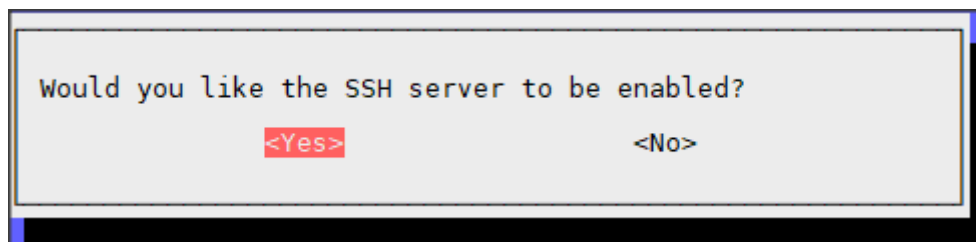
```
sudo raspi-config
```

Choose “Interfacing Options”

```
1 Change User Password Change password for the current user
2 Network Options       Configure network settings
3 Boot Options          Configure options for start-up
4 Localisation Options  Set up language and regional settings to match your location
5 Interfacing Options   Configure connections to peripherals
6 Overclock             Configure overclocking for your Pi
7 Advanced Options      Configure advanced settings
8 Update               Update this tool to the latest version
9 About raspi-config    Information about this configuration tool
```

After that activate “SSH”

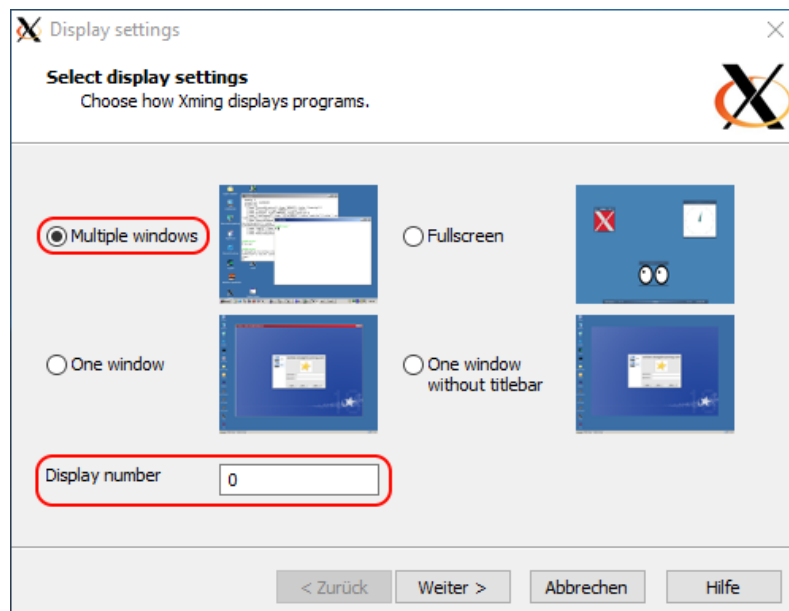
```
P1 Camera      Enable/Disable connection to the Raspberry Pi Camera
P2 SSH         Enable/Disable remote command line access to your Pi using SSH
P3 VNC         Enable/Disable graphical remote access to your Pi using RealVNC
P4 SPI         Enable/Disable automatic loading of SPI kernel module
P5 I2C         Enable/Disable automatic loading of I2C kernel module
P6 Serial      Enable/Disable shell and kernel messages on the serial connection
P7 1-Wire      Enable/Disable one-wire interface
P8 Remote GPIO Enable/Disable remote access to GPIO pins
```



Now you have to download [PuTTY](#) and [Xming](#) .

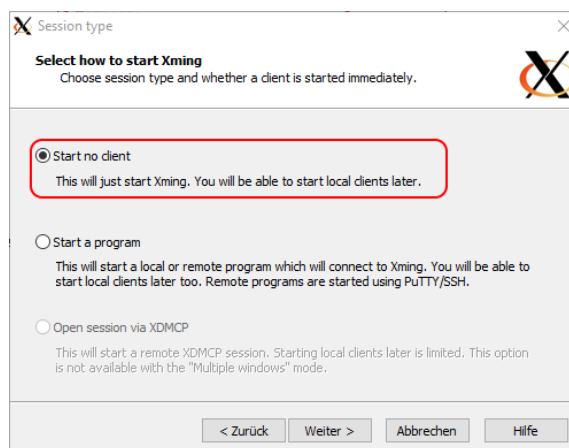
Start the file XLaunch.exe in the installation directory of Xming.

Choose “Multiple Windows” and after that enter in “Display number” 0.

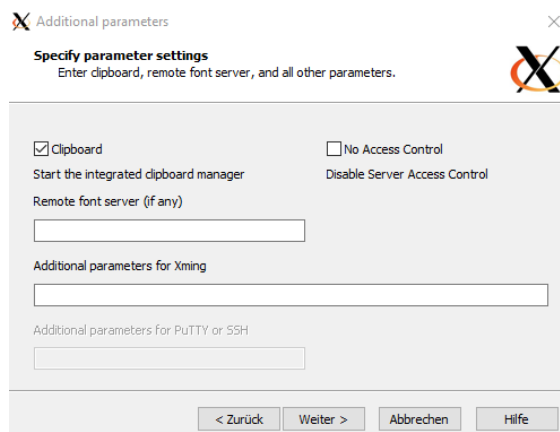


6.4 SSH-CONNECTION WITH PUTTY AND XMING

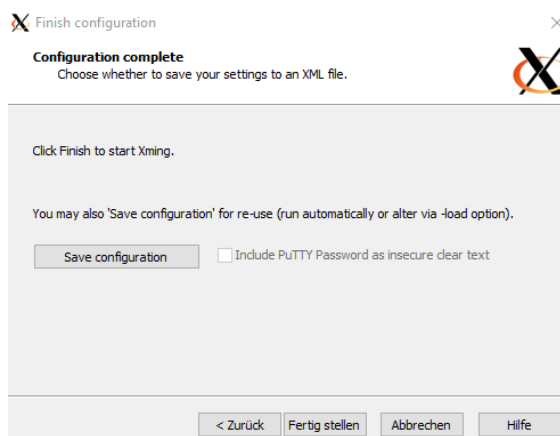
Ensure that **Start no client** is chosen:



Press **Next** without any changes:



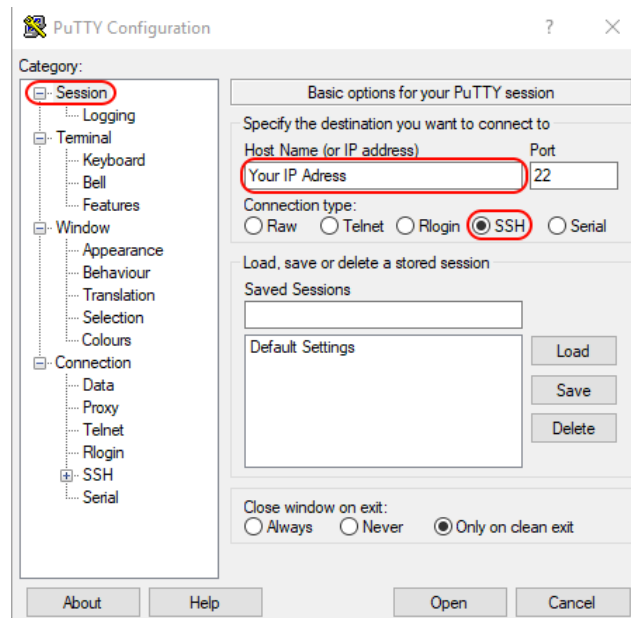
Press **Finish**:



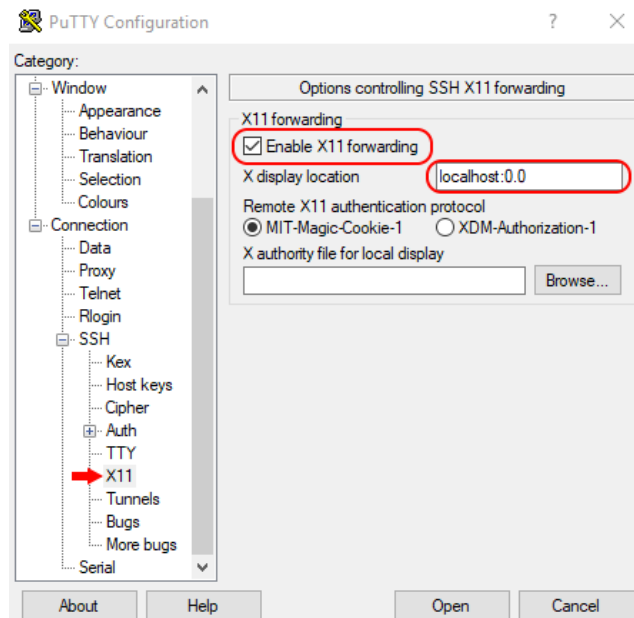
6.4 SSH-CONNECTION WITH PUTTY AND XMING

Now you have to setup PuTTY.

Ensure you have selected “Session” “SSH” and put the IP address of you Raspberry Pi in the Host Name text box.



Ensure you have turned “Enable X11 forwarding” in point “X11” to on and put „localhost:0.0” in “X display location”.



Press on “Open” and you should see the following:

```
Using username "pi".
pi@192.168.1.114's password: █
```

Fill in your login data.

You have successfully established a SSH connection.

6.4 SSH CONNECTION WITH PUTTY AND XMING

Enter following command:

```
sudo -E python3 strompi_config_gui.py
```

Now the grafical surface should appear. It should look like this:

Strompi V3 Serial Config Tool

JOY-IT®

Options Without Timesettings:

Choose StromPi V3 Powerpath Mode:

Choose Powerfail Warning Mode:

Choose Battery Shutdown Mode:

Choose Serialless Mode:

Choose Power Save Mode:

Poweroff Mode

Choose Poweroff Mode:

Real Time Clock Settings

Settings for the Clock:

- ☒ No Change
- ☐ Manual Setting
- ☐ Sync with RaspberryPi

Set the Hours:

Set the Days:

Set the Month:

Set the Minutes:

Set the Year:

Set The Weekday:

Shutdown Timer

Choose Shutdown Mode:

Choose Shutdown timer:

PowerOnButton

Choose PowerOn-Button Mode:

Choose Button Shutdown time:

Alarm Modes

Choose Alarm Mode:

Choose WakeUp-Alarm Mode:

Choose Power-Off Alarm Mode:

Choose Alarm Weekday:

Setting WakeUp-Alarm minute:

Choose Power-Off Minutes:

Choose Alarm Day:

Setting WakeUp-Alarm Hour:

Choose Power-Off Hours:

Alarm Month:

Interval Timer

Choose Interval-Alarm Mode:

Interval On-Time in minutes:

Interval Off-Time in minutes:

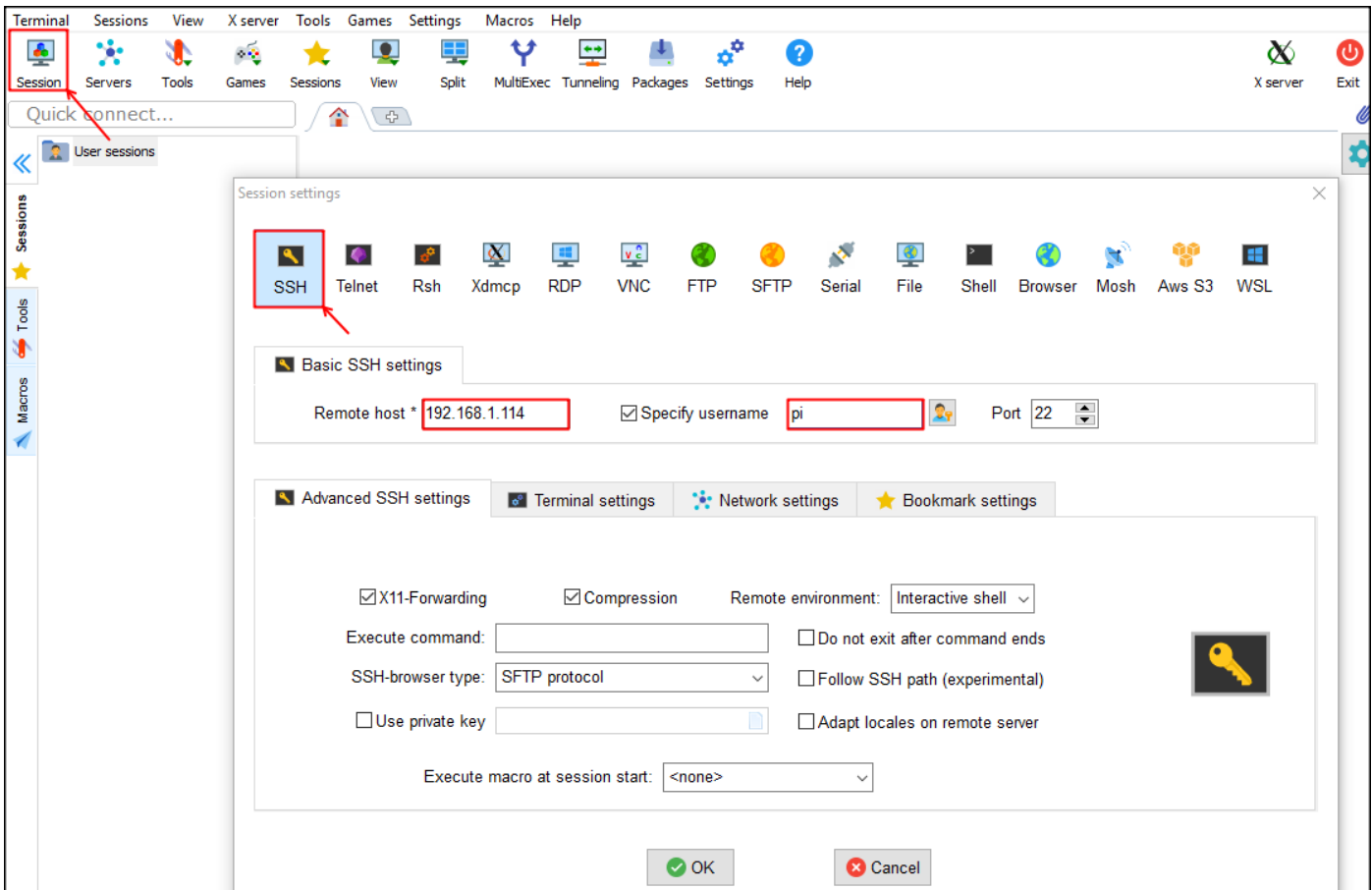
Please read the Readme file

The shown values will be synchronized at the start of the script with the setting of the StromPi. You can now change values individually and with the Press on "Update" the settings will be saved.

6.5 SSH CONNECTION WITH MOBAXTERM

Ensure that SSH is activated. If it is not setup already look at the last chapter how to activate the SSH.

As the next step download [MobaXTerm](#).



Press on “Session” now and choose SSH.

Put your IP address in the remote host textfield and in next to “Specify username” you put „pi”. After that press “OK” and enter the password of the Raspberry Pi when it is asked.

With this command:

```
sudo -E python3 strompi_config_gui.py
```

You can start the grafical interface script.

6.5 SSH CONNECTION WITH MOBAXTERM

The grafical surface should look like the following:

The shown values will be synchronized at the start of the script with the setting in the StromPi. You can now change values individually and with the Press on "Update" the settings will be saved.



It is possible that your script will look different. The PowerOff mode and the Powersave mode were added at a more recent firmware version.

7 STROMPI 3 SCRIPTS

All our published scripts for the StromPi 3 can be downloaded [here](#).

Additionally to the configscript which you already know there are the following scripts:
The serial scripts read and send necessary information through the serial interface instead of the Serailless scripts which will use a GPIO pin.

7.1 SERIAL SCRIPTS

serialShutdown.py

This script shuts down the Raspberry Pi if the primary power source fails. The script has his own shutdown timer which sets the time to wait primary power source to return prior to the Raspberry Pi shuts down. The default time is 10 seconds. You can customize the time in line 14 of the script.

```
13 #####  
14 wait_for_shutdowntimer = 10;  
15 #####
```

The shutdown timer in the script must be lower than the shutdown timer which you can set in the configscript because otherwise the power supply of the Raspberry will be seperated prior to the shut down.

You want to have this script in the autostart of your Raspberry Pi. A manual can be found in chapter 7.4 “Automatic execution of scripts”.

shutdown.py

This script shuts down your StromPi and Raspberry Pi with the PowerOff command.
This command enables the StromPi to start with the WakeUp alarm.

RTCSerial.py

This script compares the time of the Raspberry Pi with the time of the RTC of the StromPi and the older one gets overwritten with the more recent one.

If this script is in the autostart of your Raspberry Pi, the time gets automatically compared with every restart of the Raspberry Pi.

A manual can be found in chapter 7.4 “Automatic execution of scripts”.

7 STROMPI 3 SCRIPTS

7.1 SERIAL SCRIPTS

Poweralarm_serial.py

This script sends a email at the stated email in the script if the primary voltage source fails. A further email will be send if the primary voltage source comes back or if the Raspberry restarts.

```

1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3  import RPi.GPIO as GPIO
4  import time
5  import serial
6  import os
7  import smtplib
8  from email.mime.text import MIMEText
9  #####
10 # This is The config for the EMAIL notification
11 #####
12 SERVER = 'SMTP.Server.Com'
13 PORT = 587
14 EMAIL = 'Example@Example.com'
15 PASSWORD = 'Password'
16 EMPFAENGER = ['EmpfÃnger@Example.com' ,]
17 SUBJECT_Powerfail = 'Raspberry Pi Powerfail!' #Powerfail Email Betreff
18 SUBJECT_Powerback = 'Raspberry Pi Powerback!' #Powerback Email Betreff
19 SUBJECT_Restart = 'Raspberry Pi Restart!' #Restart Email Betreff
20 #####
21 # Here you can choose whether you want to receive an email when the Raspberry Pi restarts - 1 to activate - 0 to deactivate
22 Restart_Mail = 1
23 #####

```

In line 22 can be the notification email at the restart be turned on (1) or off (0).

„SERVER“ is your SMTP E-Mail server (for example smtp.your-e-mail-server.com)

„PORT“ is your server port

„EMAIL“ is your email address

„PASSWORD“ is your email password

„EMPFAENGER“ are the email addresses which get the emails

„SUBJECT_Powerfail“ is the reference by power failure in the email

„SUBJECT_Poweback“ is the reference by return of the power in the email

The script must be at autosart of your raspberry Pi that it can work. A manual can be found in chapter 7.4 “Automatic execution of scripts“.

In chapter 7.3 „Raspberry Pi Pin-layout“ is the layout of the pins of the Raspberry Pi shown.

7 STROMPI 3 SCRIPTS

7.2 SERIALLESS SCRIPTS

poweralarm_serialless.py

This script sends a email at the stated email in the script if the primary voltage source fails. A further email will be send if the primary voltage source comes back or if the Raspberry restarts. With the Serialless mode the power failure will be detected through a GPIO pin.

```

10 # Here you can choose the connected GPIO-Pin
11 GPIO_TPIN = 21
12
13 # Here you can choose whether you want to receive an email when the Raspberry Pi restarts
14 Restart_Mail = 1
15
16 # This is The config for the EMAIL notification
17 #-----
18 SERVER = 'SMTP.Server.Com'
19 PORT = 587
20 EMAIL = 'Example@Example.com'
21 PASSWORD = 'Password'
22 EMPFAENGER = ['Empfänger@Example.com' ,]
23 SUBJECT_Powerfail = 'Raspberry Pi Powerfail!' #Powerfail Email Betreff
24 SUBJECT_Powerback = 'Raspberry Pi Powerback!' #Powerback Email Betreff
25 SUBJECT_Restart = 'Raspberry Pi Restart!' #Restart Email Betreff
26 #-----

```

In line 11 of the script you can change the GPIO-pin which you use as an alternative communication for the serialless mode. In line 14 the notification email at the restart can be turned on (1) or off (0).

„SERVER“ is your SMTP E-Mail server (for example smtp.ihr-e-mail-server.de)

„PORT“ is your server port

„EMAIL“ is your email address

„PASSWORD“ is your email password

„EMPFAENGER“ are the email addresses which get the emails

„SUBJECT_Powerfail“ is the reference by power failure in the email

„SUBJECT_Poweback“ is the reference by return of the power in the email

The script must be at autosart of your raspberry Pi that it can work. A manual can be found in chapter 7.4 “Automatic execution of scripts”.

In chapter 7.3 „Raspberry Pi Pin-layout“ is the layout of the pins of the Raspberry Pi shown.

7 STROMPI 3 SCRIPTS

7.2 SERIALLESS SCRIPTS

powershutdown_serialless.py

This script shuts the raspberry down if the primary power source fails. The script has its own shutdown timer which determines the time which will be waited if the power supply returns prior to the shutdown of the Raspberry Pi. The default time is 10 seconds. You can customize the time in line 11 as well as the GPIO-pin for the alternative communication for the Serialless mode (line 10).

```
9 # Here you can choose the connected GPIO-Pin and the ShutdownTimer
10 GPIO_TPIN = 21
11 ShutdownTimer = 10
```

The shutdown timer in the script must be lower than the shutdown timer which you can set in the configscript because otherwise the power supply of the Raspberry will be separated prior to the shut down.

In chapter 7.3 „Raspberry Pin-layout“ you can see the layout of the pins of the Raspberry Pi. You need this script at the autostart of your Raspberry Pi that it will work. A manual can be found in chapter 7.4 “Automatic execution of scripts“.

RTC_serialShutdown_serialless.py

This script has all functions of the powershutdown_serialless.py script and compares additionally the time of the StromPi and the Raspberry Pi at the start. The older one gets overwritten with the more recent one.

The shutdown-timer and the GPIO-pin which you use as an alternative communication for the serialless mode can be changed in line 11 and 12.

```
10 # Here you can choose the connected GPIO-Pin and the ShutdownTimer
11 GPIO_TPIN = 21
12 ShutdownTimer = 10
```

The shutdown timer in the script must be lower than the shutdown timer which you can set in the configscript because otherwise the power supply of the Raspberry will be separated prior to the shut down.

In chapter 7.3 „Raspberry Pin-layout“ you can see the layout of the pins of the Raspberry Pi. You need this script at the autostart of your Raspberry Pi that it will work. A manual can be found in chapter 7.4 “Automatic execution of scripts“.

7 STROMPI 3 SCRIPTS

7.2 SERIALLESS SCRIPTS

RTC_EMail_serialless.py

This script compares the time of the Raspberry with the time of the StromPi and the older one gets overwritten from the more recent one. Additionally it sends an email to the set email address if the power fails. Another email will be sent if the primary voltage source returns or the Raspberry Pi restarts.

```

13 # Here you can choose the connected GPIO-Pin
14 GPIO_TPIN = 21
15
16 #Here you can choose whether you want to receive an email when the Raspberry Pi restarts
17 Restart_Mail = 1
18
19 # This is The config for the EMAIL notification
20 #-----
21 SERVER = 'SMTP.Beispiel.DE'
22 PORT = 587
23 EMAIL = 'Beispiel@abc.de'
24 PASSWORD = 'Passwort'
25 EMPFAENGER = ['Empfänger1@abc.de' , 'Empfänger2@abc.com']
26 SUBJECT_Powerfail = 'Raspberry Pi Powerfail!'
27 SUBJECT_Powerback = 'Raspberry Pi Powerback!'
28 SUBJECT_Restart = 'Raspberry Pi Restart!'
29 #-----

```

In line 14 of the script you can change the GPIO-pin which you use as an alternative communication for the serialless mode. In line 17 the notification email at the restart can be turned on (1) or off (0).

„SERVER“ is your SMTP E-Mail server (for example smtp.ihr-e-mail-server.de)

„PORT“ is your server port

„EMAIL“ is your email address

„PASSWORD“ is your email password

„EMPFAENGER“ are the email addresses which get the emails

„SUBJECT_Powerfail“ is the reference by power failure in the email

„SUBJECT_Poweback“ is the reference by return of the power in the email

In chapter 7.3 „Raspberry Pi Pin-layout“ is the layout of the pins of the Raspberry Pi shown.

The script must be at autosart of your raspberry Pi that it can work. A manual can be found in chapter 7.4 “Automatic execution of scripts”.

7 STROMPI 3 SCRIPTS

7.2 SERIALLESS SCRIPTS

RTCSerialless.py

This script was written particular for the Serialless mode.

At the start of the script the Serialless mode will be closed so that the time of the Raspberry Pi and the StrompiPi can be compared and the older one can be overwritten with the more recent one. After that the Serialless mode is activated again.

Because of the deactivated serial communication is it not possible to make changes with the configuration of the StromPi.

In line 9 of the script you can customize the GPIO-pin which is used as an alternative communication for the Serialless mode.

```
9 GPIO_TPIN = 21
```

To close the Serialless mode you have to set the communication pin for a minum of 5 seconds on "Low". In chapter 7.3 „Raspberry Pi Pin-Layout“ you cann see the layout of the pins.

If the script is at autostart of the Raspberry Pi, the time will be automatically set by every start of the Raspberry and the Serialless mode activated after that.

A manual can be found in chapter 7.4 "Automatic execution of scripts".

Start_Serialless.py

This script starts the Serialless mode to suppress all serial communication between the Raspberry Pi and the StromPi.

In line 9 of the script the GPIO-pin can be customized which is used as an alternative communication of the Serialless mode.

In chapter 7.3 "Raspberry Pi Pin-Layout" you can see the layout of the pins.

Stop_Serialless.py

This script closes the Serialless mode to enable the communication between StromPi and Raspberry Pi again.

In line 9 you can customize the GPIO-pin which is usedas an alternative communication for the Serialless mode

```
9 GPIO_TPIN = 21
```

In chapter 7.3 "Raspberry Pi Pin-Layout" you can see the layout of the pins.

7.3 RASPBERRY PI PIN-LAYOUT

Here you can see here the layout of the pins:

GPIO-Board number:			GPIO-Board Number:	
1	3.3V DC		2	5V DC
3	GPIO 2 (SDA1, I2C)		4	5V DC
5	GPIO 3 (SCL1, I2C)		6	Ground
7	GPIO 4		8	GPIO 14 (TXD0)
9	Ground		10	GPIO 15 (RXD0)
11	GPIO 17		12	GPIO 18
13	GPIO 27		14	Ground
15	GPIO 22		16	GPIO 23
17	3.3V		18	GPIO 24
19	GPIO 10 (SPI, MOSI)		20	Ground
21	GPIO 9 (SPI, MISO)		22	GPIO 25
23	GPIO 11 (SPI, CLK)		24	GPIO 8 (SPI)
25	Ground		26	GPIO 7 (SPI)
27	ID_SD (I2C, EEPROM)		28	ID_SC
29	GPIO 5		30	Ground
31	GPIO 6		32	GPIO 12
33	GPIO 13		34	Ground
35	GPIO 19		36	GPIO 16
37	GPIO 26		38	GPIO 20
39	Ground		40	GPIO 21

7.4 AUTOMATIC EXECUTION OF SCRIPTS

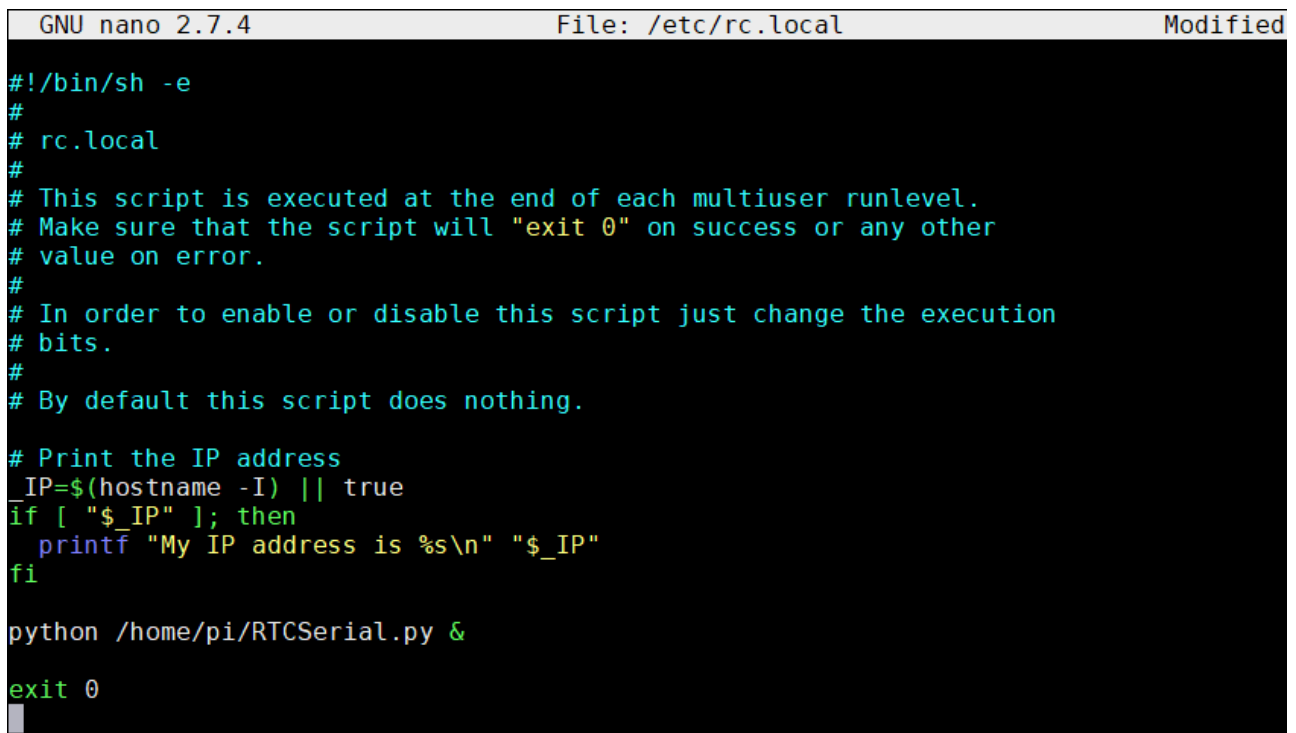
Scripts can be copied to the autostart of the Raspberry Pi to get executed at the start:

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

Here the script can be added. The following command must be added for the command **"exit 0"**.

```
python /home/pi/Skriptname.py &
```

Example for the RTCSerial.py-Script:



```
GNU nano 2.7.4 File: /etc/rc.local Modified

#!/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

python /home/pi/RTCSerial.py &

exit 0
```

Should you have not copied the script into the folder **"/home/pi"** you must adapt the path. Save the file with the key combination **STRG+O**, confirm with **Enter** and leave the editor with the combination **STRG+X**.

The changes are after a restart effective.

8 MODIFICATIONS

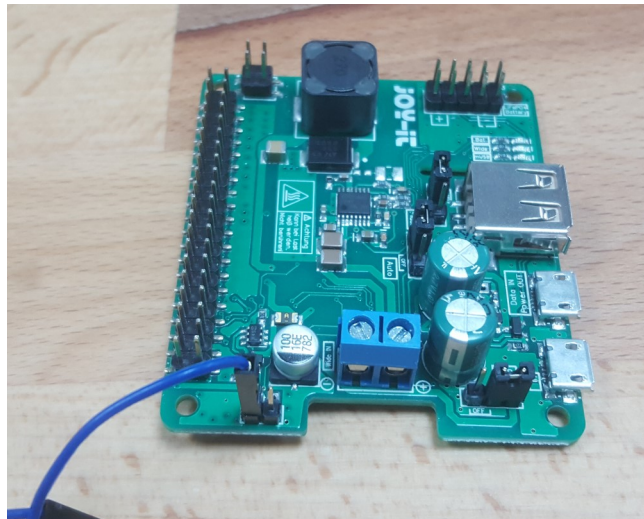
In the following chapters are all modifications for some function of the StromPi and how to use them described.

8.1 Serialless Mode

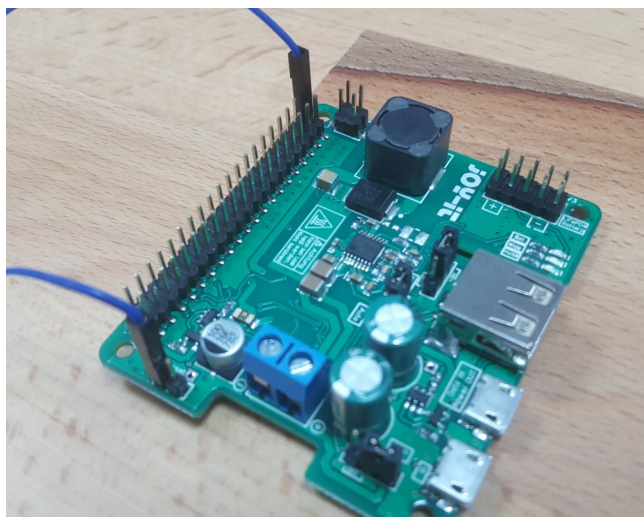
For the modification for the Serialless mode are 3 different variation which are explained in the following.

Variation 1

Variation 1 is the fastest solution but can only be used if you do not use the battery hat. You only have to plug a cable on to the Reset-pin which is shown on the following picture. To do that you only have to remove the Jumper



The other end of the cable must be plugged on the GPIO-pin.



YOU can use every other pin either but the software must be adjusted to the change.

8.1 Serialless Mode

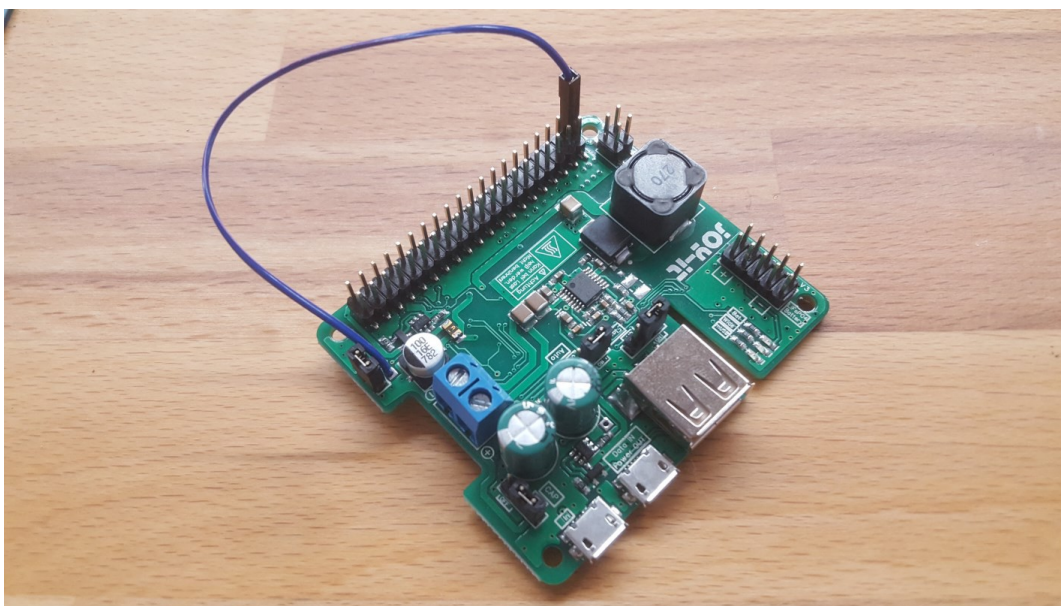
VARIATION 2

Solder one end of the cable on to the free pad like shown in the picture.

Note that the Jumper at the Reset-pin have to be plugged in for that.



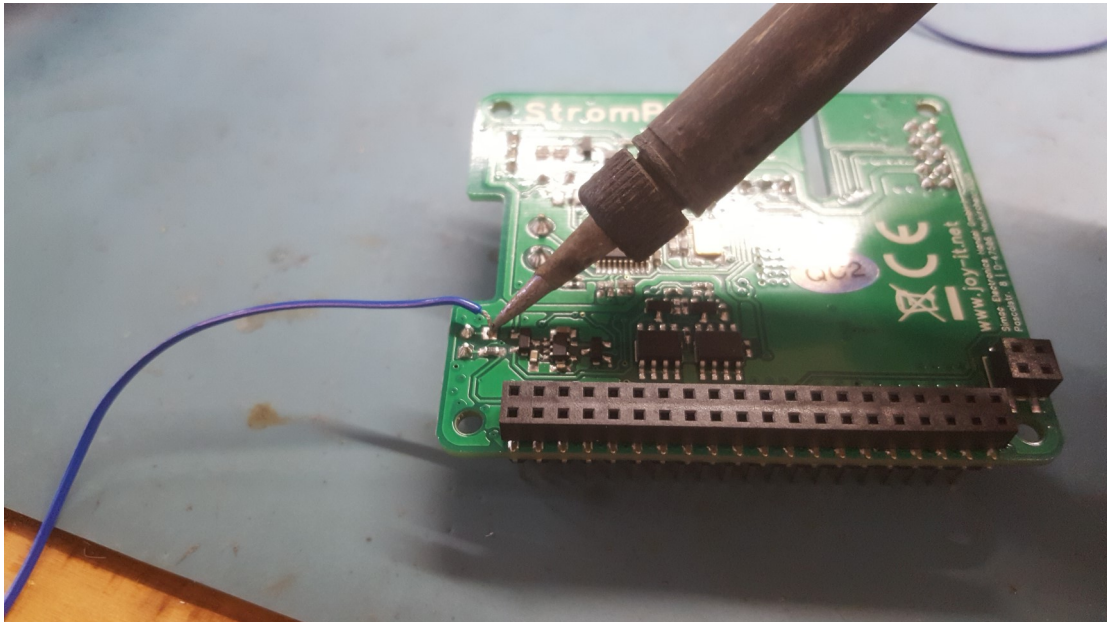
The other end of the cable is like in picture 1 plugged on to the free GPIO-pin.



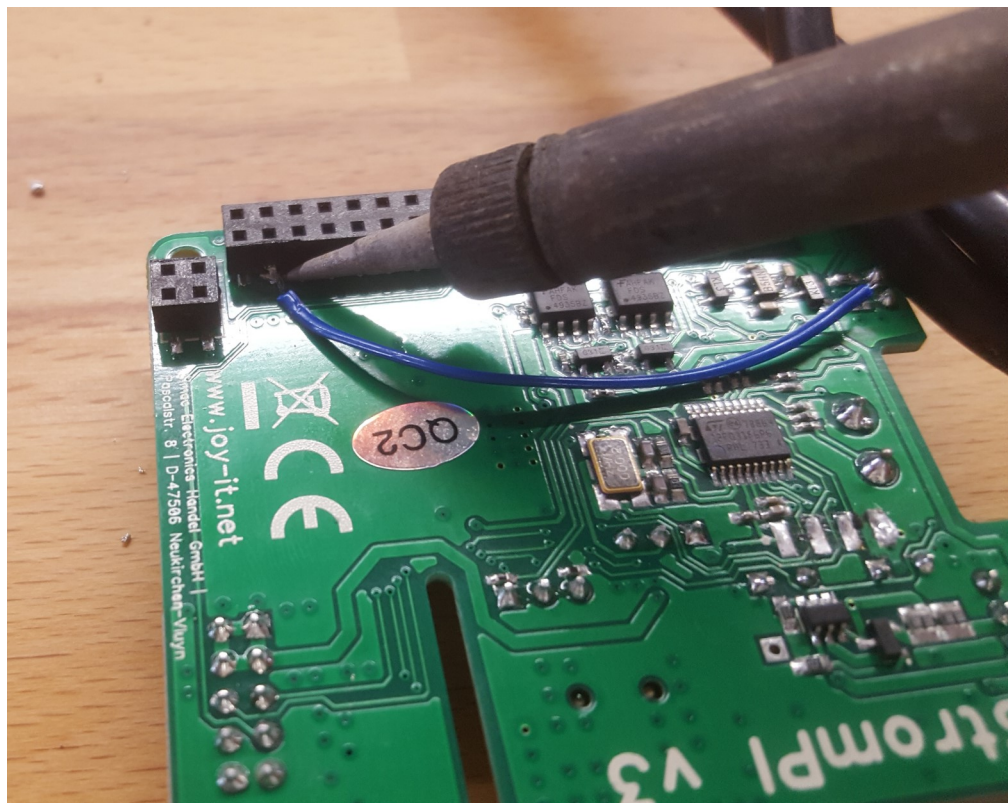
8.1 Serialless Mode

VARIATION 3

For variation 3 you have to solder one end of the cable on the free pad at the Reset-pins. The jumper still has to be set here.



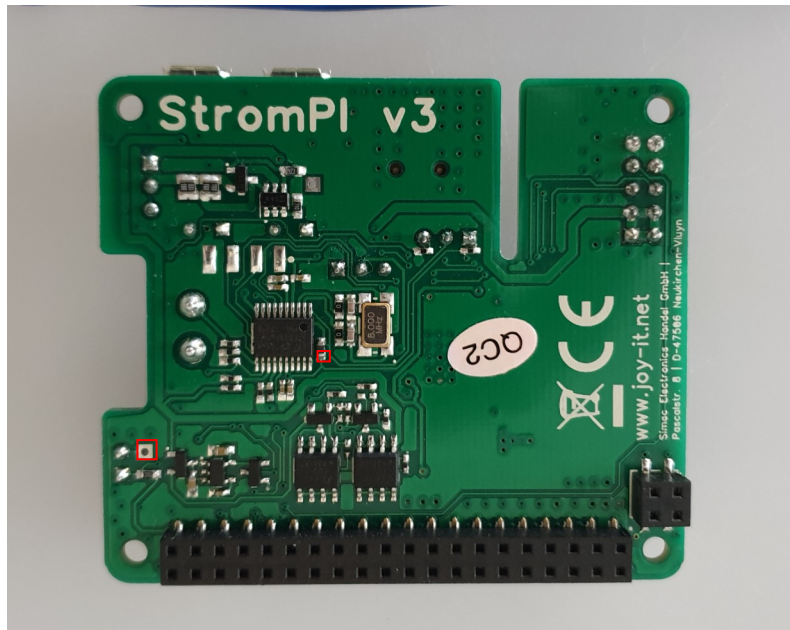
The other end of the cable have to be soldered at one GPIO-pin like shown in the picture.



8 MODIFICATION

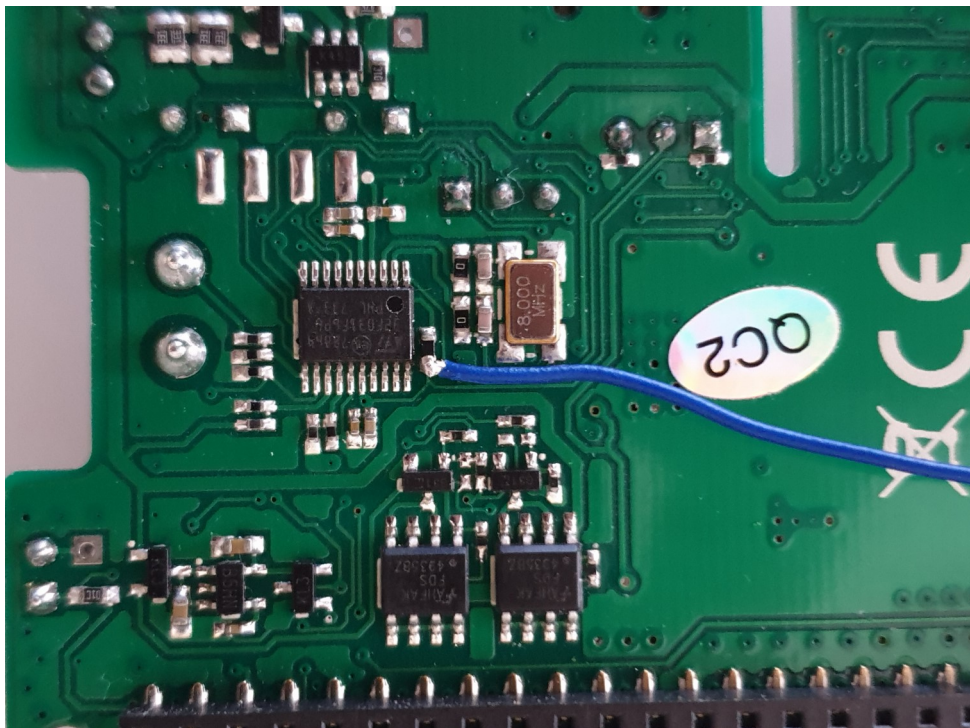
8.2 PowerOn button

The both red-marked pads to solder have to be bypassed with a cable.

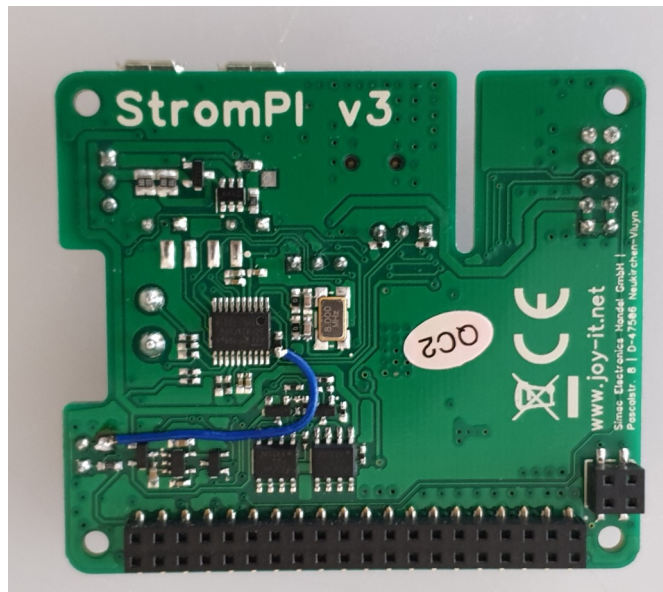


Give some solder on both pads.

Solder one end of the cable on the upper pad.

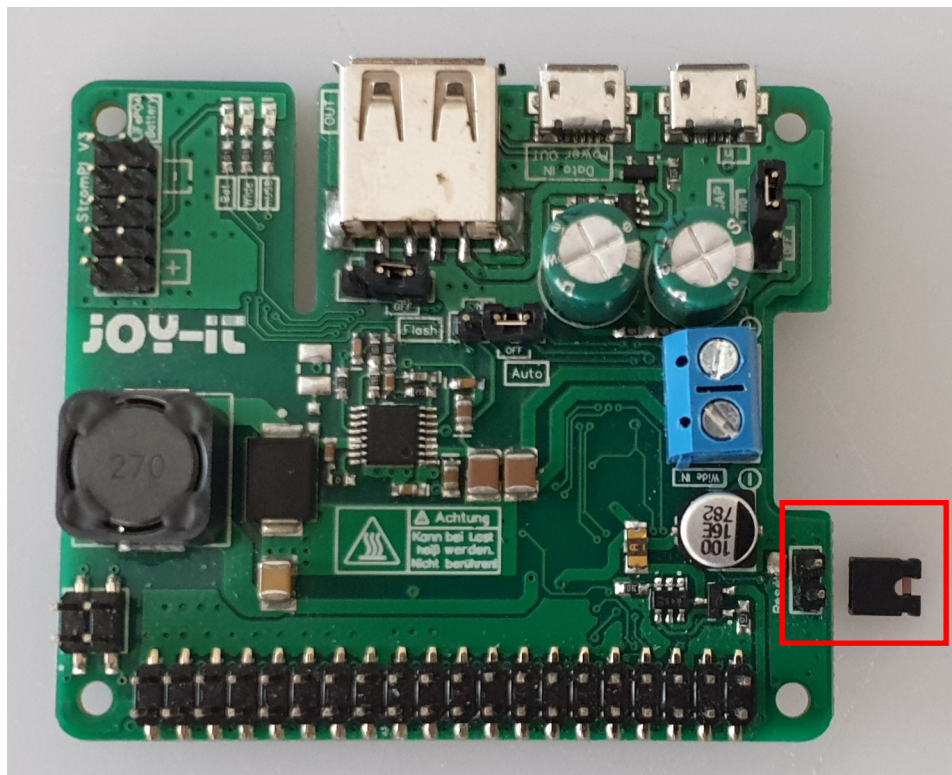


8.2 PowerOn button



At the last step, the other end of the cable must be soldered at the lower pad.

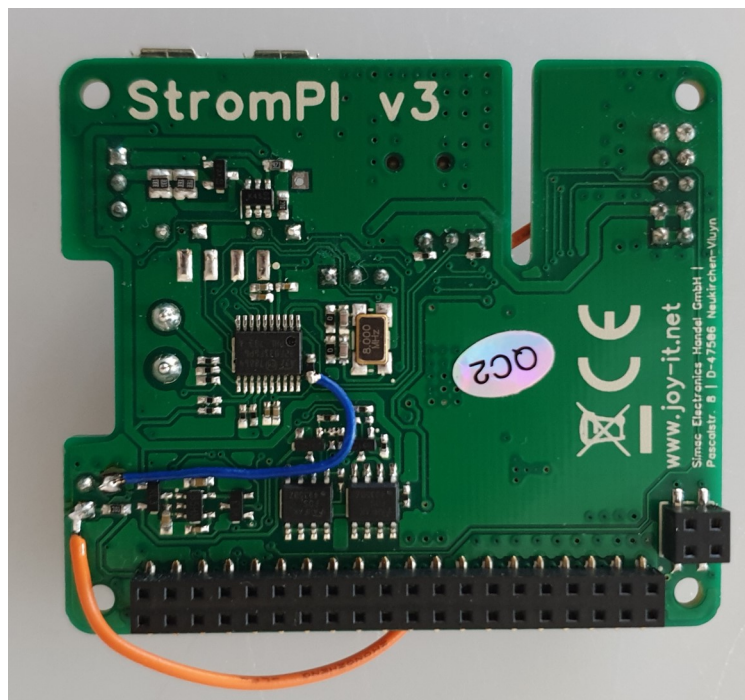
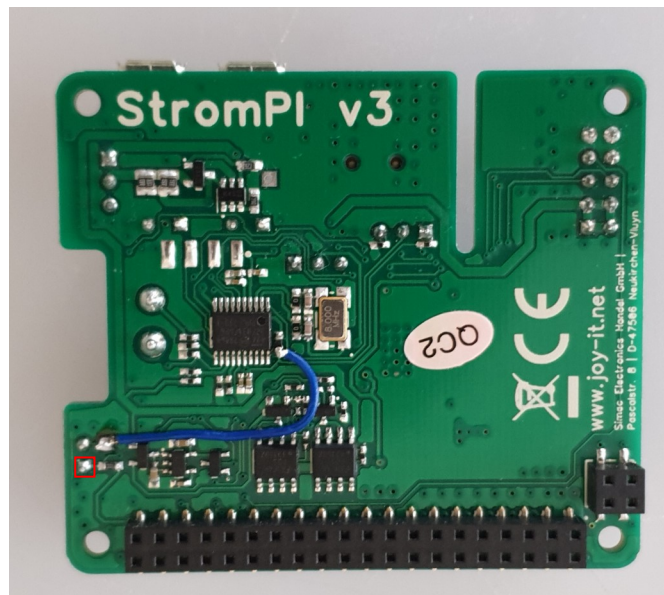
Now the system is able to restart after a "PowerOff" command through short of the red-marked Reset-pins. When it is running the jumper should not be set. To make it more comfortable it is possible to add a switch at the pins to short-circuit them for a restart.



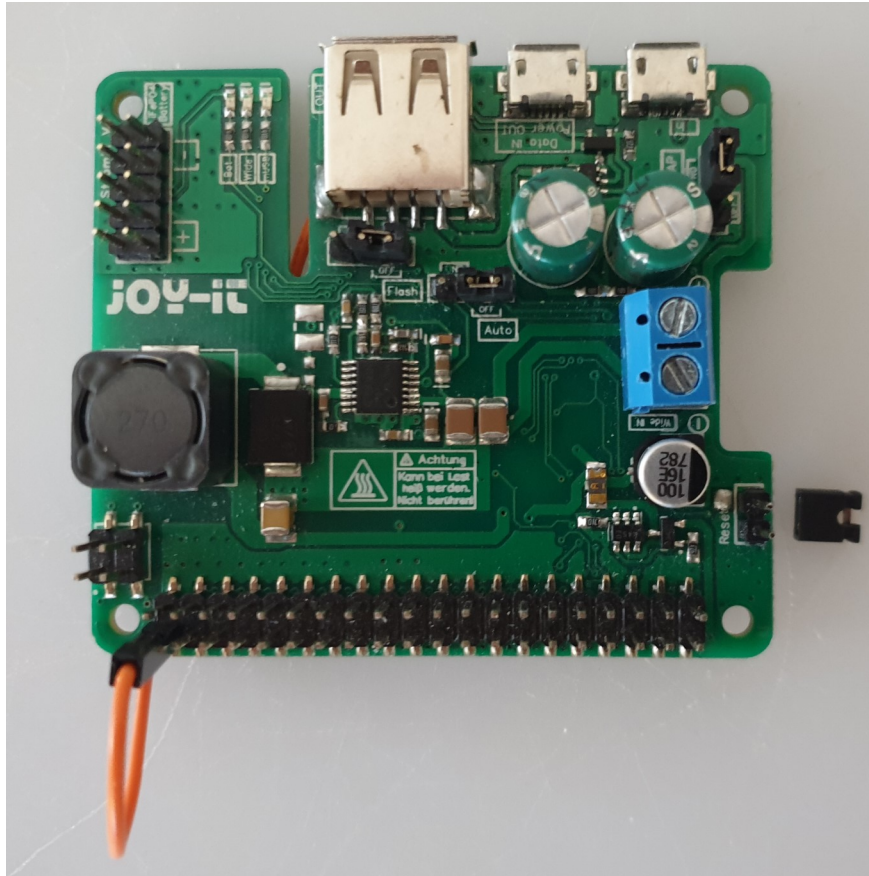
IMPORTANT: Between "PowerOff" and a restart must be a minimum of 30 seconds.

8.3 SERIALLESS + POWERON BUTTON MODIFICATION

Because the pad that is used for the serialless modification is occupied the cable for the serialless mode has to be soldered at the red-marked pad for the Serialless mode.



8.3 SERIALLESS + POWERON BUTTON MODIFICATION



For more information about the Serialless mode please read the chapter about the Serialless mode.

9 OPENING OF THE SERIAL CONSOLE OF THE STROMPI 3

To change settings manually with the StromPi console you must download Minicom and install it. After that enter the following commands:

```
sudo apt-get update
sudo apt-get install minicom
```

After Minicom is successfully installed, you can start a serial connection to the StromPi3 with the following command:

```
sudo minicom -D /dev/serial0 -b 38400
```



Attention! The system is waiting for an input and it is likely that you don't see your keystrokes. Enter the following command and confirm this with **Enter** even if you do not see your input.

You open the serial console with:

```
startstrompiconsole
```

Or the shortage:

```
sspc
```

The serial console of the StromPi 3 should open now. The current configuration settings of your device can be looked up with the command **show-status**.

```
-----
Welcome to the StromPi 3 Console
-----
Type help to view a list of available commands.
[When you press ENTER the previous command would be executed again]
>|
```

A general overview gives you the command **help**.

With the key combination of **Strg + A + X** , **Enter** you are closing Minicom.

With the key combination of **Shift+ Page up / down** you can scroll through the terminal window. The mouse wheel works either.

9 OPENING OF THE SERIAL CONSOLE OF THE STROMPI 3

Enter an “E” and this window appears:

```

+-----[Comm Parameters]-----+
| A - Serial De |                               | | |
| B - Lockfile Loc | Current: 115200 8N1 |
| C - Callin Pro | Speed | Parity | Data |
| D - Callout Pro | A: <next> | L: None | S: 5 |
| E - Bps/Par/B | B: <prev> | M: Even | T: 6 |
| F - Hardware Flo | C: 9600 | N: Odd | U: 7 |
| G - Software Flo | D: 38400 | O: Mark | V: 8 |
|                               | E: 115200 | P: Space |
|                               |                               |
| Change which |                               |
|                               | Stopbits |                               |
| | Screen a | W: 1 | Q: 8-N-1 |
| | Save set | X: 2 | R: 7-E-1 |
| | Save set |                               |
| | Exit |                               |
| | Exit fro | Choice, or <Enter> to exit? █ |
+-----+-----+
  
```

Press “D” to set a baud rate of 38400.

Confirm this two times with Enter to return to the configuration menu.

After that press on “Save as dfl” and the following window should appear:

```

+-----+
| Configuration saved |
+-----+
  
```

You can leave with “ESC” the configuration menu and afterwards press “Strg + A”.

Now you see at the console the following:

```
CTRL-A Z for help | 38400 8N1 | NOR | Minicom 2.7 | VT102 | Offline | serial0
```

If it looks the same like the picture above, your Minicom is successfully configured.

You can enter the following commands now to start Minicom:

```
sudo minicom
```

Now you can open the Strompi Console with:

```
startstrompicconsole
```

Or with the shortage:

```
sspc
```

On the following two sites are all StromPi console commads listed.

10 STROMPI 3 CONSOLE COMMANDS

Commands for the CLI and main firmwareversions

Start StromPi console:

```
startstrompiconsole
```

oder

```
sspc
```

Close StromPi console:

```
quit
```

List all commands:

```
help
```

Setup the time of the RTC of the StromPi:

```
set-clock <hour> <minutes> <seconds>
```

Setup of the date of the RTC of the StromPi:

```
set-date <day> <month> <year> <weekday>
```

Output of the measured voltage:

```
adc-output
```

Setup of the StromPi modes (<1>: mUSB -> Wide, <2>: Wide -> mUSB, <3>: mUSB -> Bat, <4>: Wide -> Bat, <5>: mUSB -> Wide -> Bat, <6>: Wide -> mUSB -> Bat):

```
strompi-mode <number>
```

Output of the current time of the RTC of the StromPi:

```
time-output
```

Display the current global settings:

```
show-status
```

Display the current alarm settings:

```
show-alarm
```

Shut down the Raspberry P with the StromPi:

```
poweroff
```

Commands for the CLI firmwareversion

Set the shutdown timer in seconds:

```
set-timer <shutdown-seconds>
```

Activate <1> or deactivate <0> the Raspberry Pi shutdown:

```
Shutdown-enable <option>
```

10 STROMPI 3 CONSOLE COMMANDS

Activate <1> or deactivate <0> the warning by failure of primary power supply:

```
Warning-enable <option>
```

Setup of Wakeup alarms (<1>: time, <2>: date, <3>: weekday):

```
alarm-mode <mode-number>
```

Activate <1> or deactivate <0> the Wakeup alarms:

```
alarm-enable <option>
```

Setup of the time of the Wakeup alarms:

```
alarm-set-time <hour> <minutes>
```

Setup of the weekday of the Wakeup alarms:

```
alarm-set-weekday <weekday>
```

Setup of the date of the Wakeup alarms:

```
alarm-set-date <date> <month>
```

Setup of the time of the PowerOff alarms:

```
poweroff-set-time <hour> <minutes>
```

Activate <1> or deactivate <0> the PowerOff alarms:

```
poweroff-enable <option>
```

Activate <1> or deactivate <0> the Serialless modes:

```
serialless-mode <option>
```

Setup of the voltage of the battery when the Raspberry Pi is shutdown
(<0>: deactivated, <1>: < 10%, <2>: < 25%, <3>: < 50%):

```
batlevel-shutdown <level-number>
```

Activate <1> or deactivate <0> the PowerOff interval alarms:

```
interval-enable <option>
```

Setup of the Ontime of the interval alarms:

```
interval-ontime <min>
```

Setup of the Offtime of the interval alarms:

```
interval-offtime <min>
```

Activate <1> or deactivate <0> the PowerOn buttons:

```
poweronbutton-enable <option>
```

Setup of the time of the PowerOn-Button initialization:

```
poweronbutton-timer <seconds>
```

11 SUPPORT

We also support you after your purchase. If you have any questions left or encounter any problems, feel free to contact us by email, phone or with our ticket-supportsystem on our website.

E-Mail: service@joy-it.net

Ticket-System: <http://support.joy-it.net>

Telefon: +49 (0)2845 98469 – 66 (10 - 17 o'clock)

For further information please visit our website:

www.joy-it.net

12 EU-DECLARATION OF CONFORMITY

Manufacturer: Simac GmbH
Pascalstr. 8
47506 Neukirchen-Vluyn

Article: JOY-IT StromPi 3

Description: USV / Wide power supply
Raspberry PI 3 & other SBC (SingleBoardComputer) 5V

Purpose: Application in Prototyping / Experimental setup

Confirms that the product "JOY-IT StromPi 3" if it is used as intended, the requirements of the following guideline are fulfilled:

2014/30/EU (EMC) & 2011/65/EU (RoHS)

The following norms have been used to assess this device:

EN 55022:2010+ AC:2011, EN 55024:2010, EN 61000-3-2:2014, EN 61000-3-3:2013

Date	Name of person in charge	Signature	Position in business
24.08.2018	Xu Lu		Manager