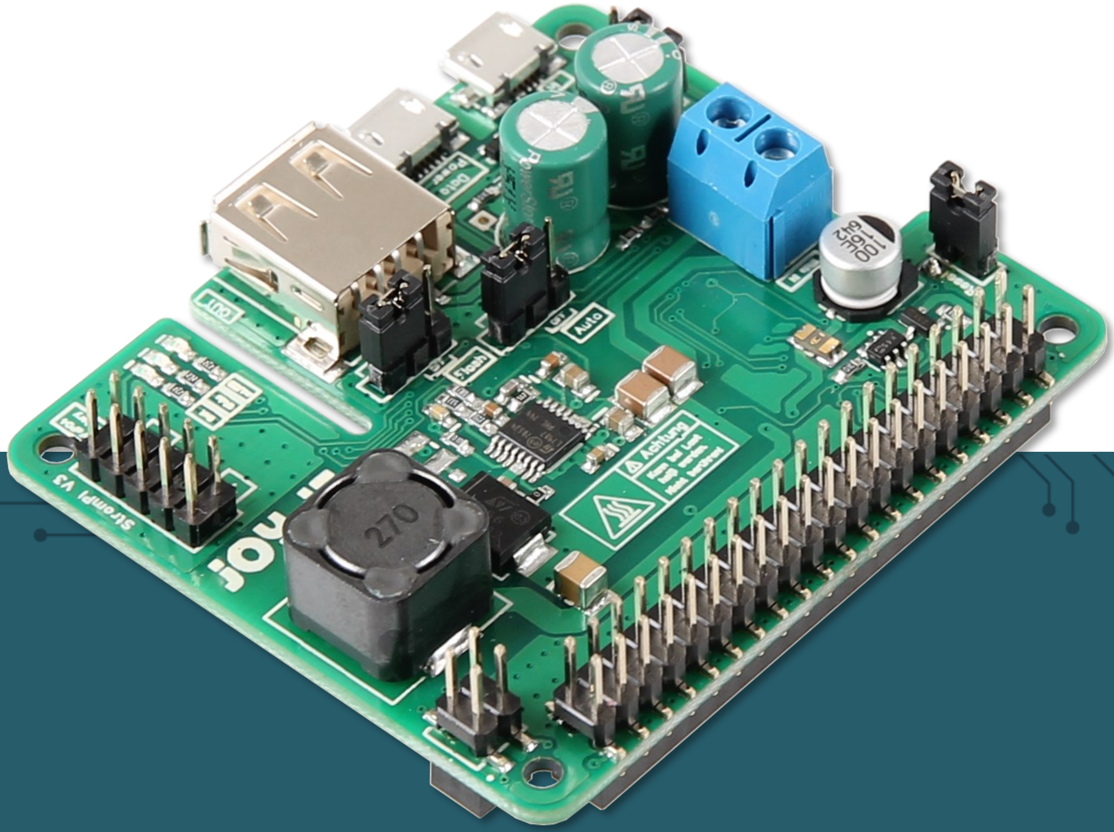


STROMPI 3

Power-Solution

1. TABLE OF CONTENT

2. GENERAL INFORMATION

Dear customer,

Thank you for purchasing our product. The structure and functions of the StromPi3 are explained in the following chapters. We will show you which settings should be made so that the Raspberry Pi will work smoothly with the StromPi.
If you have any difficulties with the usage of the device, do not hesitate to contact us.

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

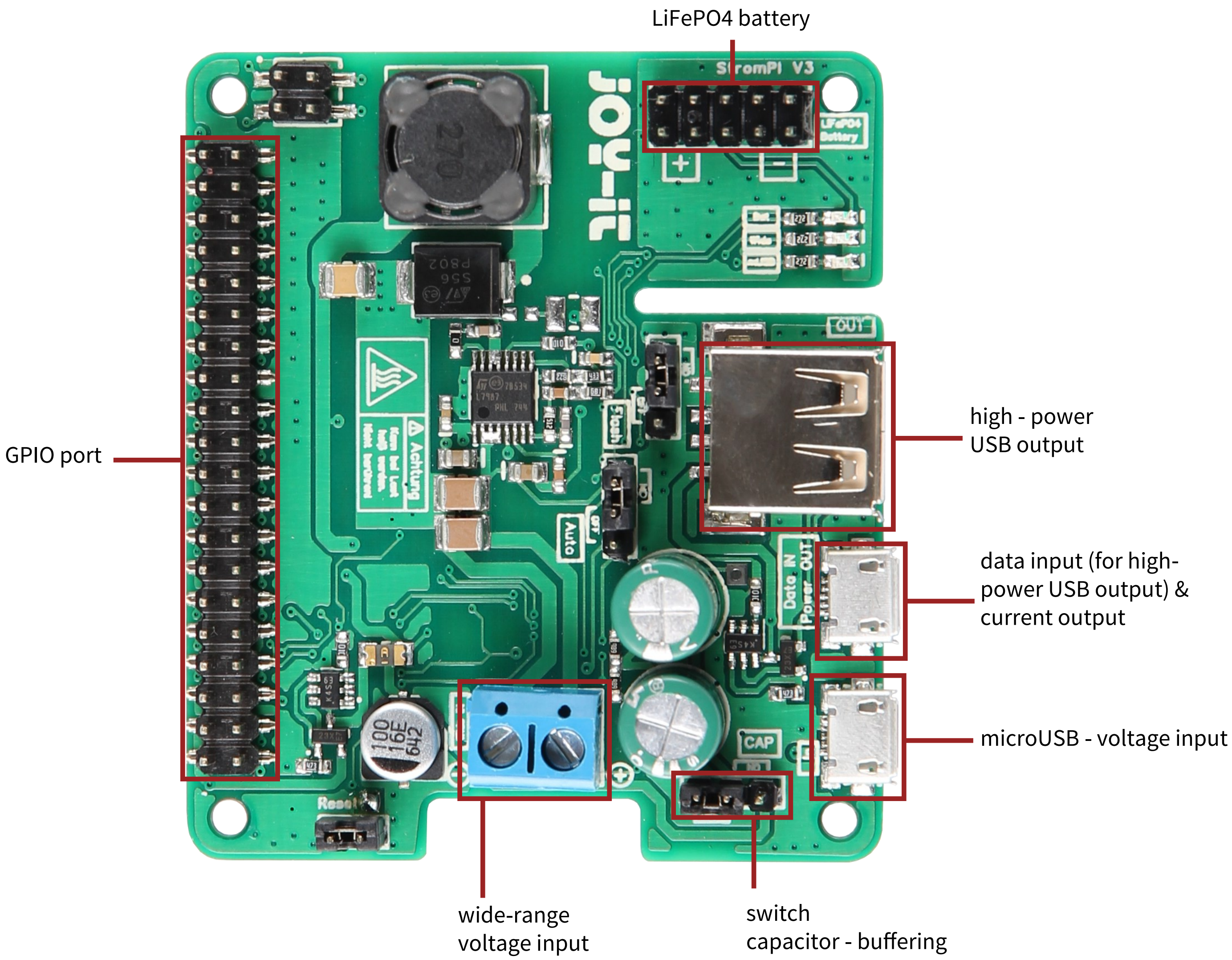
The prioritized inputs can easily be changed which means the StromPi 3 will fit perfectly into your project. With the voltage range from 6V to 61V and a current with up to 3A, there is enough power for even bigger projects. In addition, the StromPi 3 has an own configurable, autonomously operating microsystem unit which can be used for a programmable start-stop behaviour:

Let your system boot and shut down at pre-set 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 of 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) which makes the StromPi3 to an absolute allrounder.

The StromPi 3 is perfectly suitable for every application area!

⚠ This manual is made for the firmware version **1.72**. We recommend upgrading the firmware of the StromPi regularly. You can download the newest version of the firmware here. There is a separate manual how to flash the firmware in the download register.

3. OVERVIEW



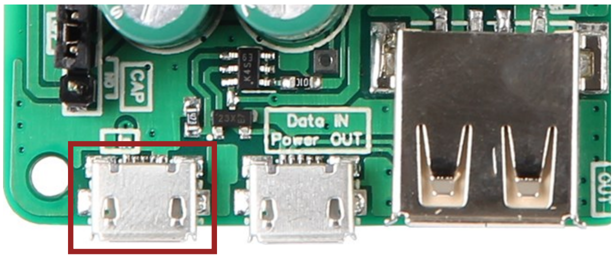
⚠ With heavy load the StromPi 3 can heat up. Do not ouch the circuit board during operation to avoid burns. Also allow a good supply of air to prevent the unit from overheating.

4. POWER SOURCES

The StromPi 3 can get its power from three different power sources. In case of a power failure, it switches to the power source which is configurated as the second and if necessary to the third.

microUSB

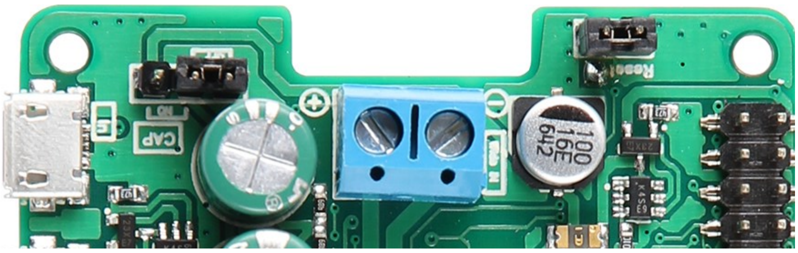
The StromPi 3 can of course, as usual, be powered via the microUSB port.



Wide - Range

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

Attention! The voltage 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 the StromPi3.



Battery unit

The StromPi 3 can be expanded with an optional 1000 mAh or 2000 mAh LiFe04 battery, which is a rechargeable emergency power source. You only have to plug in the separately available battery unit, which then can be configurated as the second or third current source. Because of the implemented GPIO-bar on the battery unit, you do not have any restrictions with the usage of it.

If the orange LED on the battery unit lights up, it means the battery is charging.

If the LED flashes, it is an indication that the battery unit is defective.

In order to be able to use all functions of the StromPis to the full extent, you must use our Python scripts. These are listed and explained in chapter 9 StromPi 3 scripts.

5. QUICKSTART

In this chapter, you will find the necessary steps to use the StromPi 3.

Activation of the serial communication

More details can be found in chapter 8.1
Use the following commands to install the required dependencies and then open the file **config.txt** .

```
sudo apt-get update
sudo apt-get install python-serial python3-serial
sudo nano /boot/config.txt
```

Now append the following lines to the end of the file:

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

Open the Raspberry configuration and disable in
5 Interfacing Options → *P6 Serial* the serial login shell and activate the serial hardware.

Opening of the serial console

More details can be found in chapter 11
Use the following commands to install Minicom and connect the Raspberry Pi to the StromPi 3.

```
sudo apt-get update
sudo apt-get install minicom
sudo minicom -D /dev/serial0 -b 38400
```

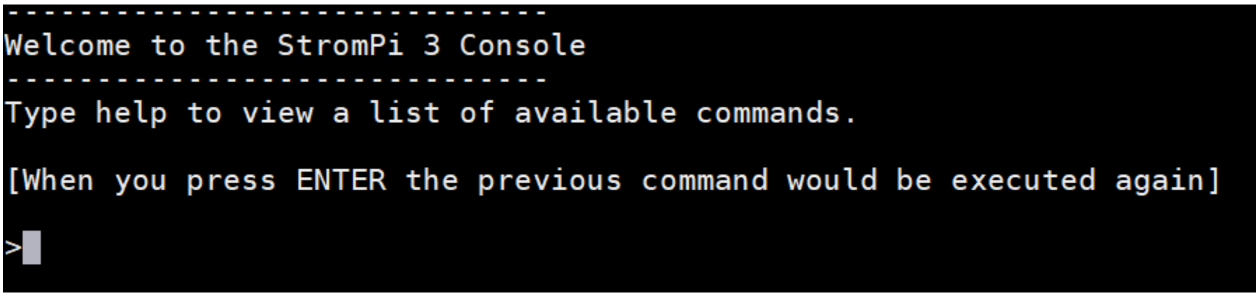
Open the console of the StromPi 3 with the following command :
(the typed characters are not displayed)

```
startstrompiconsole
```

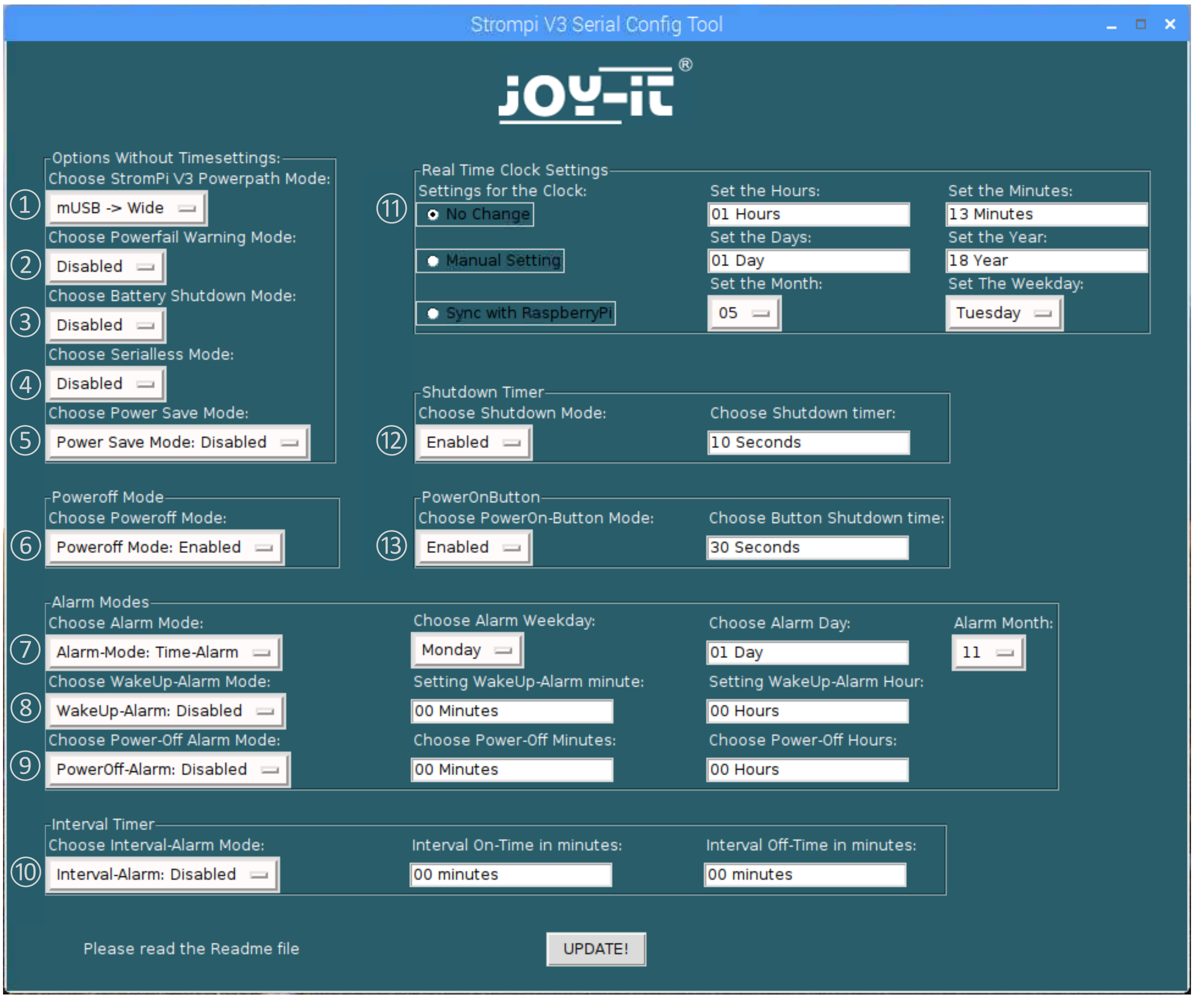
- ... or the shortcut:

```
sspc
```

The command **show-status** opens the current configuration setting and **help** gives you an overview of the commands.
The start screen of the StromPi 3 console looks like this:



```
--------------------------------
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]

>
```



Strompi V3 Serial Config Tool

Options Without Timesettings:

① Choose StromPi V3 Powerpath Mode: mUSB -> Wide

② Choose Powerfail Warning Mode: Disabled

③ Choose Battery Shutdown Mode: Disabled

④ Choose Serialless Mode: Disabled

⑤ Choose Power Save Mode: Power Save Mode: Disabled

Poweroff Mode

⑥ Choose Poweroff Mode: Poweroff Mode: Enabled

Alarm Modes

⑦ Choose Alarm Mode: Alarm-Mode: Time-Alarm

Choose Alarm Weekday: Monday

Choose Alarm Day: 01 Day

Alarm Month: 11

⑧ Choose WakeUp-Alarm Mode: WakeUp-Alarm: Disabled

Setting WakeUp-Alarm minute: 00 Minutes

Setting WakeUp-Alarm Hour: 00 Hours

⑨ Choose Power-Off Alarm Mode: PowerOff-Alarm: Disabled

Choose Power-Off Minutes: 00 Minutes

Choose Power-Off Hours: 00 Hours

Interval Timer

⑩ Choose Interval-Alarm Mode: Interval-Alarm: Disabled

Interval On-Time in minutes: 00 minutes

Interval Off-Time in minutes: 00 minutes

Real Time Clock Settings

⑪ Settings for the Clock:

● No Change

○ Manual Setting

○ Sync with RaspberryPi

Set the Hours: 01 Hours

Set the Minutes: 13 Minutes

Set the Days: 01 Day

Set the Year: 18 Year

Set the Month: 05

Set The Weekday: Tuesday

Shutdown Timer

⑫ Choose Shutdown Mode: Enabled

Choose Shutdown timer: 10 Seconds

PowerOnButton

⑬ Choose PowerOn-Button Mode: Enabled

Choose Button Shutdown time: 30 Seconds

Please read the Readme file

① Choose the StromPi V3 Powerpath Mode
② Enable or disable the Powerfail Warning
③ Adjust the Battery Shutdown Mode
④ Enable or disable the Serialless Mode
⑤ Enable or disable the Power Save Mode
⑥ Enable or disable the Poweroff Mode
⑦ Adjust the Alarm Mode
⑧ Adjust the WakeUp-Alarm Mode
⑨ Adjust the PowerOff-Alarm Mode
⑩ Adjust the Interval - Alarm Mode
⑪ Adjust the Real Time Clock
⑫ Enable or disable the Shutdown-Timers and adjust the time (in seconds)
⑬ Enable or disable the PowerOnButton and adjust the time (in seconds)

⚠ A serial connection is required to make all settings described in the following chapters. Chapter 8 explains how to configure the serial port for the Raspberry Pi.

StromPi V3 Powerpath Mode:
The StromPi V3 Powerpath Mode determines the prioritisation of the separate voltage sources.

Powerfail Warning Mode:
The PowerFail Warning Mode will warn in case that the prioritised voltage source is failing.

Battery Shutdown Mode:
The Battery Shutdown Mode determines at which battery charge level the Raspberry Pi should be shut down.

Serialless Mode:
The Serialless Mode allows using the serial ports for other purposes by communicating via a GPIO pin. [Modification required! see chapter 10 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 set up the time and the date of the real time clock manually or automatically with comparing and synchronising the time with the Raspberry Pi.

Shutdown Timer:
If you have activated the Shutdown Timer and the StromPi detects a power failure at the primary power source, the StromPi3 will send a signal to the Raspberry Pi through the serial interface to start the shutdown (according to the script). After the counter has expired, the StromPi 3 will disconnect the power supply of the Raspberry Pi.

PowerOnButton:
With the PowerOnButton, you can restart the switched off Raspberry Pi by closing the contact between the reset pins. [Modification required! See chapter 10 Modification].

Interval Timer:
The interval timer can be set up to shutdown or boot the system after a selected time.

Alarm Mode:
The alarm mode refers to the WakeUp-Alarm and PowerOff-Alarm Mode. It selects if the alarm goes off daily, weekly or only at a specific date.

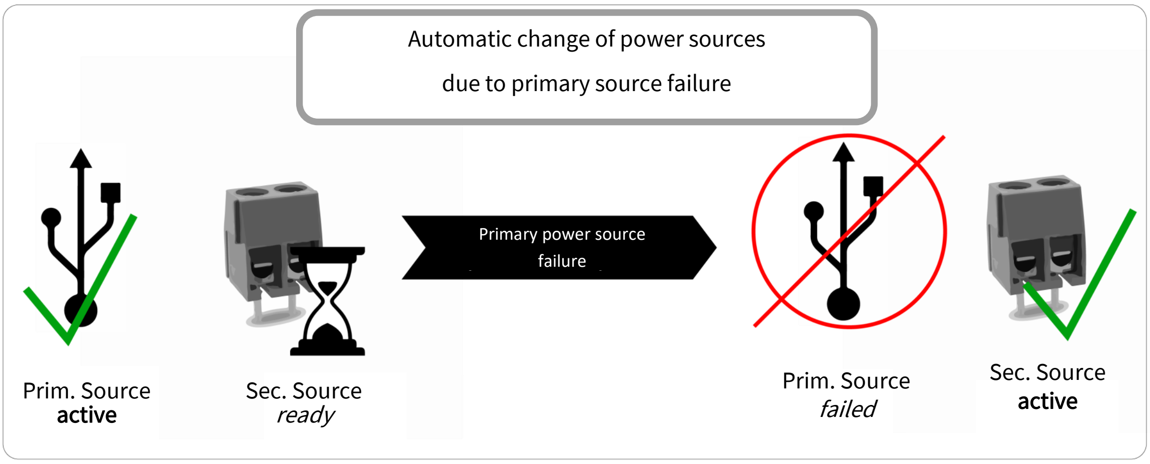
WakeUp-Alarm Mode:
The WakeUp-Alarm Mode starts the Raspberry Pi at a specific time.

PowerOff -Alarm Mode:
The PowerOff-Alarm Mode shuts down the Raspberry Pi at a specific time.

Poweroff Mode:
The Poweroff Mode requires the Raspberry Pi to be switched on with the PowerOnButton after a shutdown by the ShutDown Timer. Additionally, the Poweroff Mode prevents detection if the primary power supply returns.

6.1 Prioritisation of input

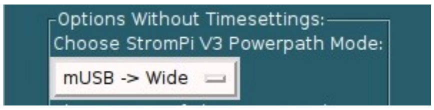
The voltage source from which the StromPi 3 obtains its energy can be adjusted according to your wishes and the intended use.
In addition, a second or even a third voltage source can be selected, which automatically takes over the supply in case the first source should fail.
The factory setting is configured so that the microUSB input serves as the main power source. If this fails, the wide-range input is automatically added as an alternative source.



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 (from version 1.6 on)	microUSB	Wide - Range	battery
Mode 6 (from version 1.6 on)	Wide - Range	microUSB	battery

You can set the corresponding mode via the config script under "Choose StromPi V3 Powerpath Mode".

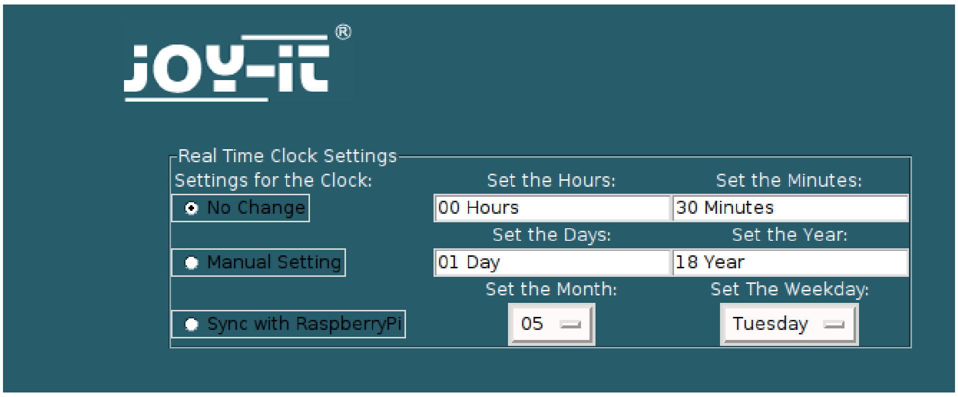

Options Without Timesettings:
Choose StromPi V3 Powerpath Mode:
mUSB -> Wide

6.2 Time-controlled applications

6.2.1 Real Time Clock

The StromPi 3 also has a real-time clock. Should the power fail, even time-controlled applications are not endangered.
Time controlled starts and shutdowns of the Raspberry Pi is also not a problem!
Please note that for continuous operation of the RTC at least one of the power sources (mUSB, Wide Range or battery) must remain connected to the StromPi 3. If all power sources are disconnected, the current time is not stored.

The real time clock (RTC) of the StromPi 3 can be set with the config script:



Real Time Clock Settings

Settings for the Clock:

● No Change

○ Manual Setting

○ Sync with RaspberryPi

Set the Hours: 00 Hours

Set the Minutes: 30 Minutes

Set the Days: 01 Day

Set the Year: 18 Year

Set the Month: 05

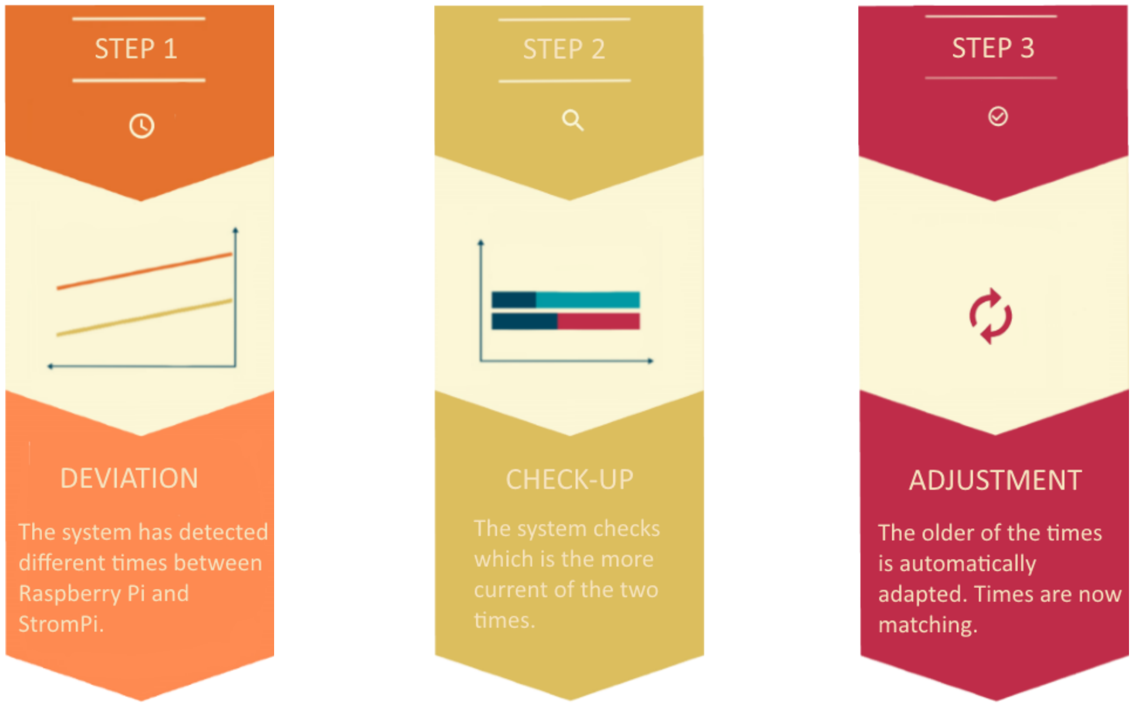
Set The Weekday: Tuesday

Setting the time and date manually
To set the time and date manually, click on **Manual Setting**. Now, you can enter the time and date in the corresponding fields to the right side. Click on **Update** to apply the settings.

Synchronise time and date with the Raspberry Pi
To synchronize the time and date with the Raspberry Pi, click on **Sync with RaspberryPi**. Clicking on **Update** compares the time of the RTC of the StromPi with the Raspberry Pi. The older time is overwritten with the more recent one. The pre-set time of the StromPi is the 01. May 2018 and the Raspberry Pi updates its time when it is connected to the Internet.

6.2.2 Automatic configuration of time

In addition, we provide a script which synchronises the time after the boot of the Raspberry. This script can be downloaded here in the section „Scripts“. Download the archive, unzip and copy the script to your Raspberry Pi.
In chapter 9.4 Automatic execution of scripts, you can read how to add scripts to the Autostart so that they perform automatically after a boot.



STEP 1

DEVIATION

The system has detected different times between Raspberry Pi and StromPi.

STEP 2

CHECK-UP

The system checks which is the more current of the two times.

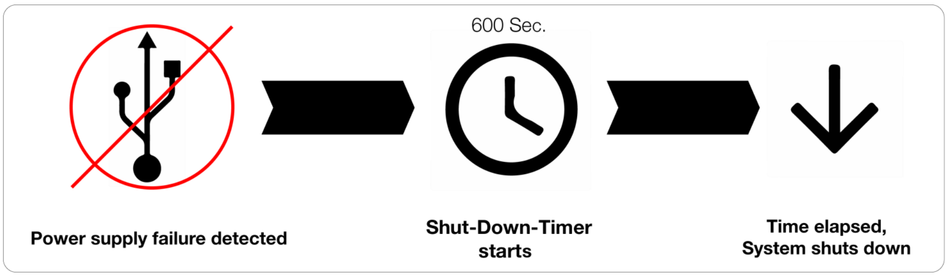
STEP 3

ADJUSTMENT

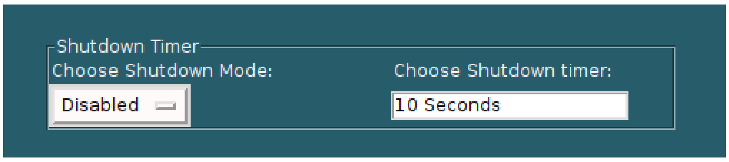
The older of the times is automatically adapted. Times are now matching.

6.2.3 Shutdown Timer

The StromPi 3 has a shutdown timer which automatically interrupts the power supply of your Raspberry Pi after a failure of the primary power supply is detected. You can choose the period of time, when the shutdown takes place after the failure of the power supply, individually, according to your needs. For example, if you have set the shutdown timer to 600 (600 seconds -> 10 minutes), the system operates like the following:


600 Sec.
Power supply failure detected
Shut-Down-Timer starts
Time elapsed, System shuts down

If the StromPi 3 now detects a power failure, the timer starts. After 10 minutes the StromPi 3 will then disconnect the power supply to the Raspberry Pi. Additionally, we provide a script that can detect a power failure either via the serial console or via a GPIO pin and then shut down the Raspberry Pi safely. The shutdown timer can be set in the config script by **Shutdown Timer**.



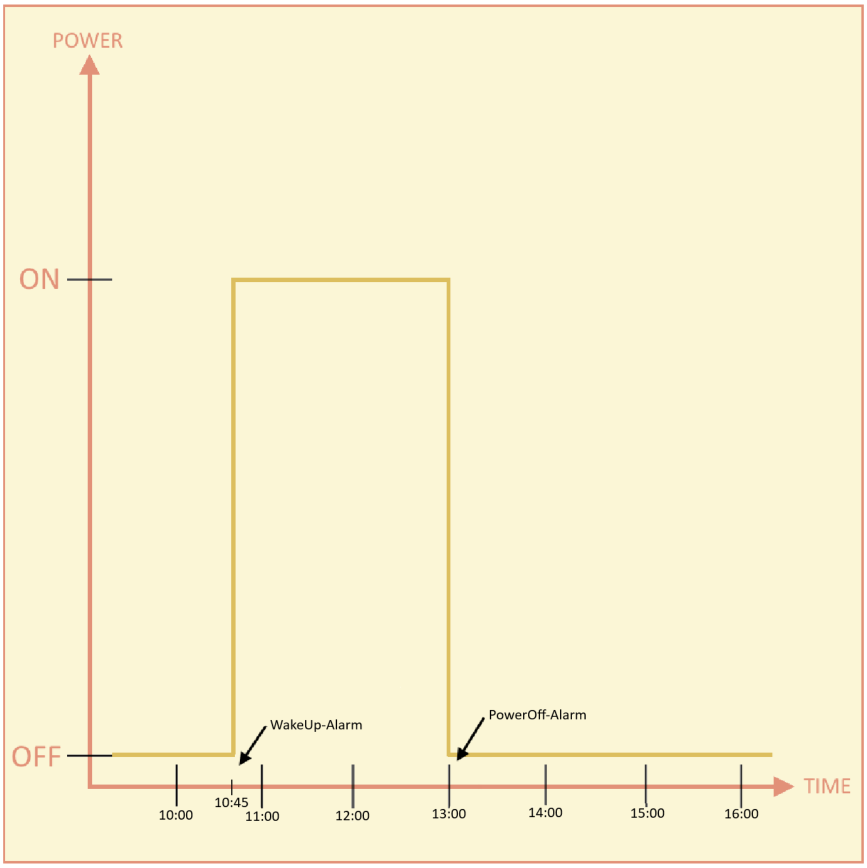
Shutdown Timer
Choose Shutdown Mode: Disabled
Choose Shutdown timer: 10 Seconds

To enable the Shutdown Timer, select **Enable** at **Choose Shutdown Mode** and enter the desired shutdown time in seconds into the adjacent field. Finally, confirm with **Update**.

6.2.4 Alarm Modes

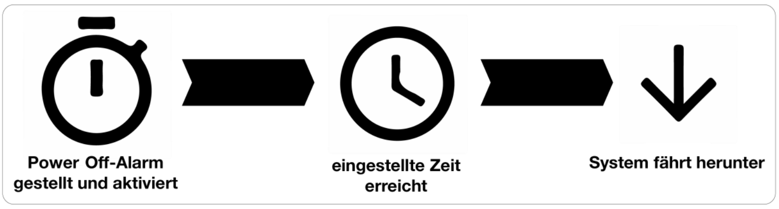
The StromPi3 has the possibility to start and shut down the Raspberry time-controlled. With the combination of WakeUp-Alarm and PowerOff-Alarm Mode, you can configure an individual start-stop behaviour for your Raspberry Pi (e.g. start at 10:45 every day and shut down at 13:00).

Your Raspberry Pi will only be active during the time window in which the incurring tasks have to be completed.

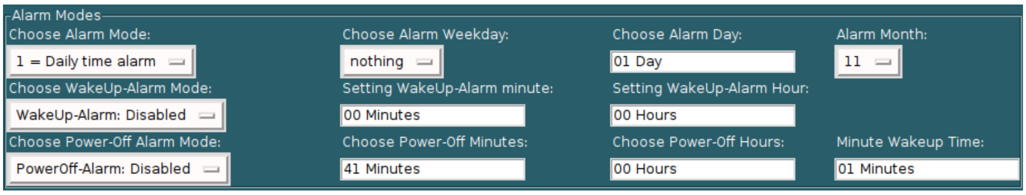


6.2.5 PowerOff - Alarm Mode

The PowerOff alarm allows you to use the Raspberry Pi every day at a preset time. However, the Raspberry Pi can only be shut down once a day using this mode.


Power Off-Alarm gestellt und aktiviert
eingestellte Zeit erreicht
System fährt herunter

If you want to shut down the StromPi3 time-controlled, start the config script.



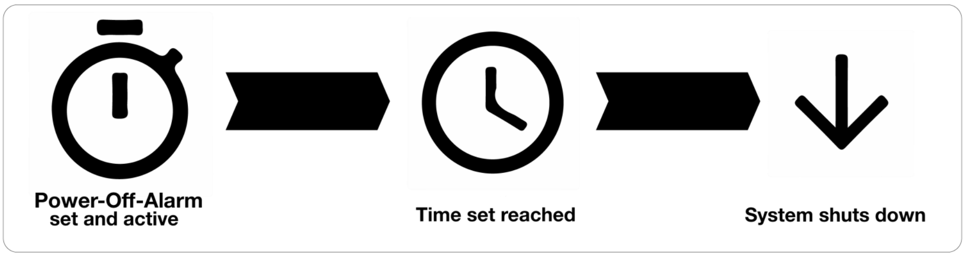
Alarm Modes

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

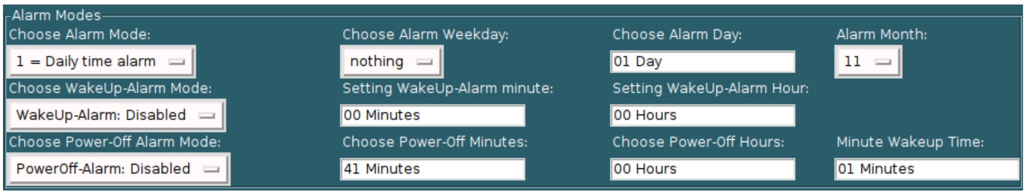
Switch ***Choose PowerOff-Alarm Mode*** to **Enable** to activate the PowerOff-Alarm. Enter the time when the Raspberry Pi should be shut down into the two fields on the right side. Enter the minutes in the field ***Choose Power-Off Minutes*** and the hours in the field ***Choose Power-Off Hours.***

6.2.6 WakeUp - Alarm Mode

The WakeUp-Alarm allows you to start the Raspberry time-controlled. In addition, you have the possibility to activate these functions for a specific date or a weekday.



If you want to start the StromPi 3 time-controlled, start the config script.



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

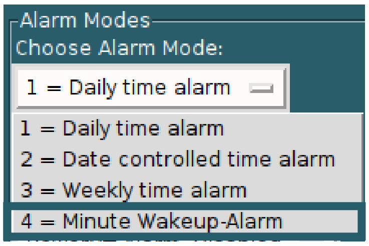
Switch ***Choose WakeUp-Alarm Mode*** to **Enable** to activate the WakeUp-Alarm. Enter the time, when the Raspberry Pi should start, in the two fields to the right side. Enter the minutes in the field ***Choose WakeUp -Alarm Minutes*** and the hours in the field ***Choose WakeUp-Alarm hours***. The Raspberry Pi can only be started once a day using this mode.

Depending on what you have set in ***Choose Alarm Mode***, your Raspberry Pi will now boot daily, weekly or just at a set date at the set time. With Alarm Mode 1, your Raspberry starts at the set time every day. With Alarm Mode 2, your Raspberry starts on a specific date at the time you set. With Alarm Mode 3, your Raspberry will start at the specific weekday you set once a week. With Alarm Mode 4 you activate the Minute WakeUp Alarm. The alarm mode refers exclusively to the WakeUp-Alarm, it has no influence on the PowerOff-Alarm.

⚠ **Attention!** Your Raspberry Pi can only be started via the WakeUp-Alarm if it was previously shut down automatically via the PowerOff-Alarm or it was shut down manually with the serial command *poweroff*.

6.2.7 Minute WakeUp-Alarm Mode

The Minute WakeUp Alarm allows you to set an automatic timer, to start the Raspberry Pi after the Raspberry Pi has been switched off by the PowerOff command or the PowerOff-Alarm. This timer is given in minutes. To activate this mode, the alarm mode must be set to **Minute WakeUp-Alarm** in the config script.



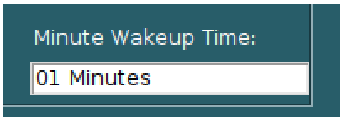
Alarm Modes

Choose Alarm Mode:

1 = Daily time alarm

- 1 = Daily time alarm
- 2 = Date controlled time alarm
- 3 = Weekly time alarm
- 4 = Minute Wakeup-Alarm

The time must then be set (specified in minutes):

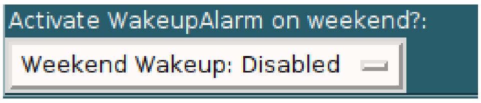


Minute Wakeup Time:

01 Minutes

6.2.8 Weekend WakeUp-Alarm

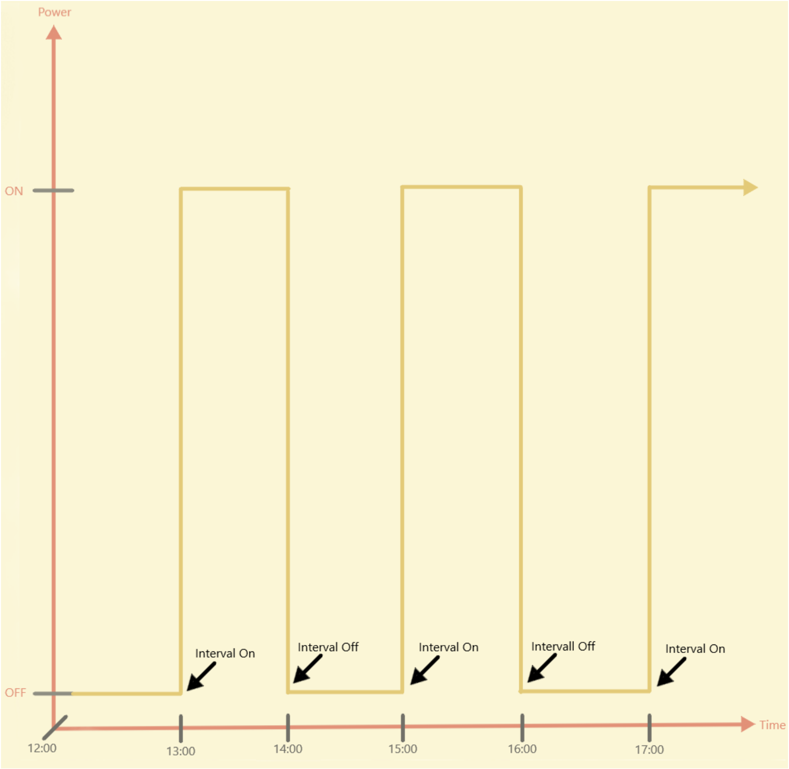
The Weekend WakeUp-Alarm allows you to switch the WakeUp-Alarm on or off during the weekend. If you set this mode to **Disabled**, the StromPi 3 will be switched off during the weekend (Saturday and Sunday) and will not be launched by the WakeUp-Alarm. If you continue to start the Raspberry Pi with the WakeUp-Alarm during the weekend, the WakeUp-Alarm must be set to **Enabled**.



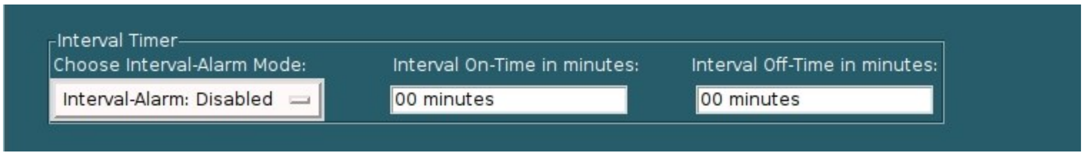
Activate WakeupAlarm on weekend?:

Weekend Wakeup: Disabled

6.2.9 Interval Function



The interval function allows you to define time intervals in which the Raspberry Pi is switched off and on again with the help of the StromPi 3.You can activate and configure this function in the config script by **Timer**.



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

Switch ***Choose Interval-Alarm Mode*** to **Enable** to enable the Interval function.

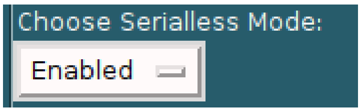
At ***Interval On-Time in minutes,*** you can set the time that the Raspberry Pi should remain on in minutes.

At ***Interval Off-Time in minutes***, you can set the time that Raspberry Pi should remain off in minutes.

7. ADDITIONAL FUNCTIONS

7.1 Serialless Mode

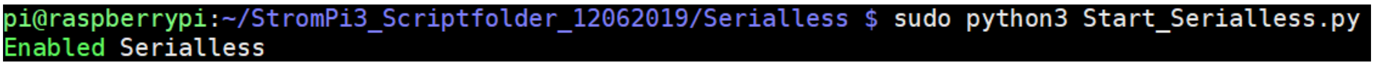
The Serialless Mode allows you to suppress any serial communication of the StromPi, you can e.g. use another HAT, which uses the serial interface, can be used simultaneously with the StromPi. However, this also prevents the StromPi 3 from being configured, which is why it is useful to configure the StromPi before you activate the Serialless Mode.
In addition, this mode causes that the shutdown warning is not transmitted via the serial interface as before, but via a free GPIO pin.
To activate the Serialless Mode, you have to set the Serialless Mode in the config script by under ***Choose Serialless Mode*** to **Enable**.



Choose Serialless Mode:

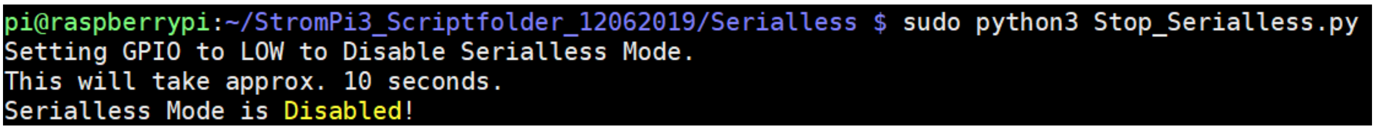
Enabled

You must also run the *Start_Serialless.py* script.



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

When you have done this, the Serialless Mode is activated and the serial communication is now completely suppressed. To end the Serialless Mode again, you have to execute the script *Stop_Serialless.py* .This script sets the GPIO pin to low if the GPIO pin is on low for more than 5 seconds, the Serialless Mode is deactivated again.



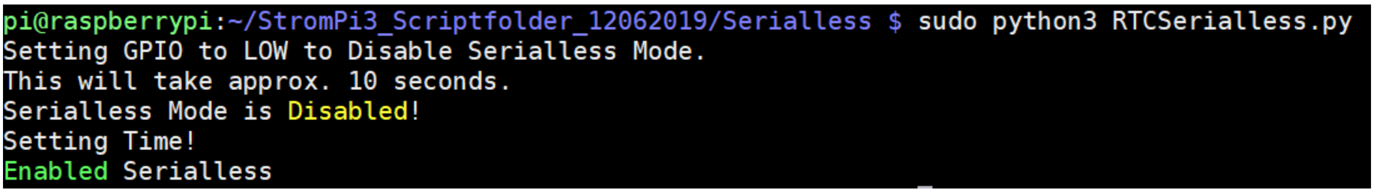
```
pi@raspberrypi:~/StromPi3_Scriptfolder_12062019/Serialless $ sudo python3 Stop_Serialless.py
Setting GPIO to LOW to Disable Serialless Mode.
This will take approx. 10 seconds.
Serialless Mode is Disabled!
```

⚠ For this mode the Serialless - modification is necessary which is described in detail in chapter 10 Modifications.

⚠ The scripts used here require Python3.

7.2 Example of usage with real time clock

Now to show you one of the many possible applications, we have created a script (*RTCSerialless.py*) which after starting the script, deactivates the Serialless Mode, then synchronizes the real time clock, and activates the Serialless Mode again. Here is the console output after successful execution of the scripts:



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

7.3 PowerOn-Button Mode

The PowerOn-Button Mode allows the StromPi to be started again after shutdown with the *poweroff* command or the PowerOff-Alarm without disconnecting the power supply or removing the BatteryHATs.

Only a small modification is necessary for this function. A cable bridge must be soldered in between two solder pads. Then it is possible to restart the StromPi 3 by bridging the reset pin.

If the StromPi 3 is already modified for the Serialless Mode or used in Serialless Mode with the PowerOn-Button modification a further small modification must be made.

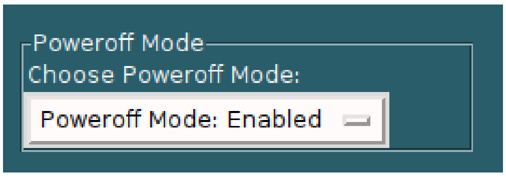
In chapter 10 Modifications, we will explain exactly what is required.

⚠ **IMPORTANT: The set time in the configuration script is the initialization of the PowerOn-Button. We recommend a time of at least 30 seconds.**

7.4 Poweroff Mode

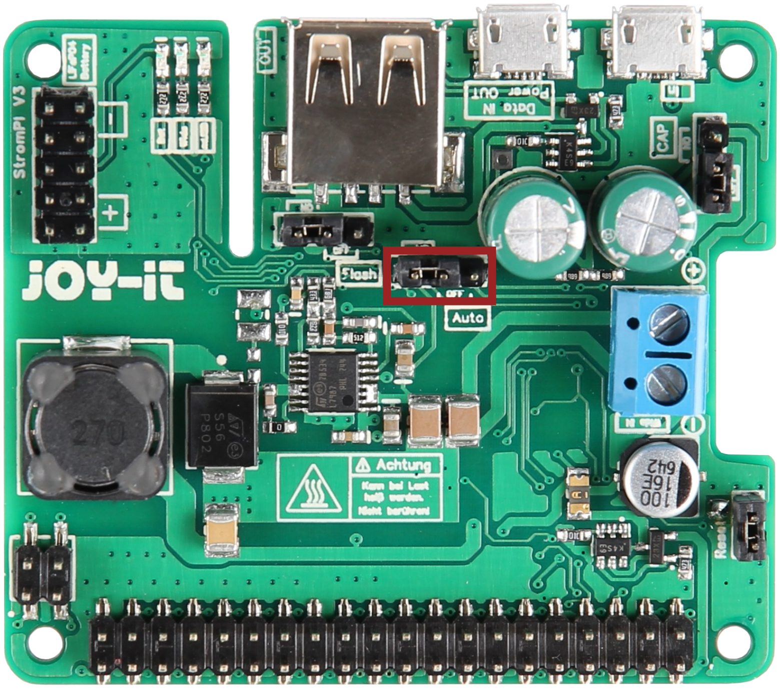
The Poweroff Mode ensures that after a power failure your Raspberry Pi remains switched off until you press the PowerOn-Button
(see chapter 7.3).

You lose the option of recognising a powerback so that the Raspberry will shut down in any case and the StromPi will switch off the power supply to the Raspberry after the set time. To switch on the Poweroff Mode, you must set ***Choose Poweroff Mode*** in the config script to **Enable**.

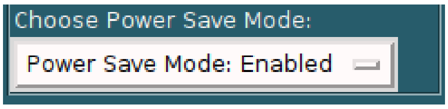

Poweroff Mode
Choose Poweroff Mode:
Poweroff Mode: Enabled

7.5 Power Save Mode

The Power Save Mode allows you to drastically reduce the power consumption of the StromPi 3 by switching off the voltage converter when the Wide port is not in use and switching it only on in case of a power failure of the primary power supply. This is only possible in low current mode (below 2.5 A).
If the current is too high, there could be complications when switching the power paths.
To activate the Power Save Mode, the jumpers are set **OFF** as shown in the following picture.



Additionally, you have to set ***Choose Power Save Mode*** to **Enable** in the config script.


Choose Power Save Mode:

Power Save Mode: Enabled

8. CONFIGURATION OF STROMPI 3

Here, we show you how to set up the serial communication for the StromPi. This manual is based on Raspbian Buster from December 2019.

We recommend that you use the latest version of the Raspbian Buster image.

8.1 Activation of the serial communication

In order to use the functions of the StromPi and to make settings, a few adjustments must be made to your Raspberry Pi.

First, you need to install the Python3-Serial library by typing the following command into the console of the Raspberry Pi:

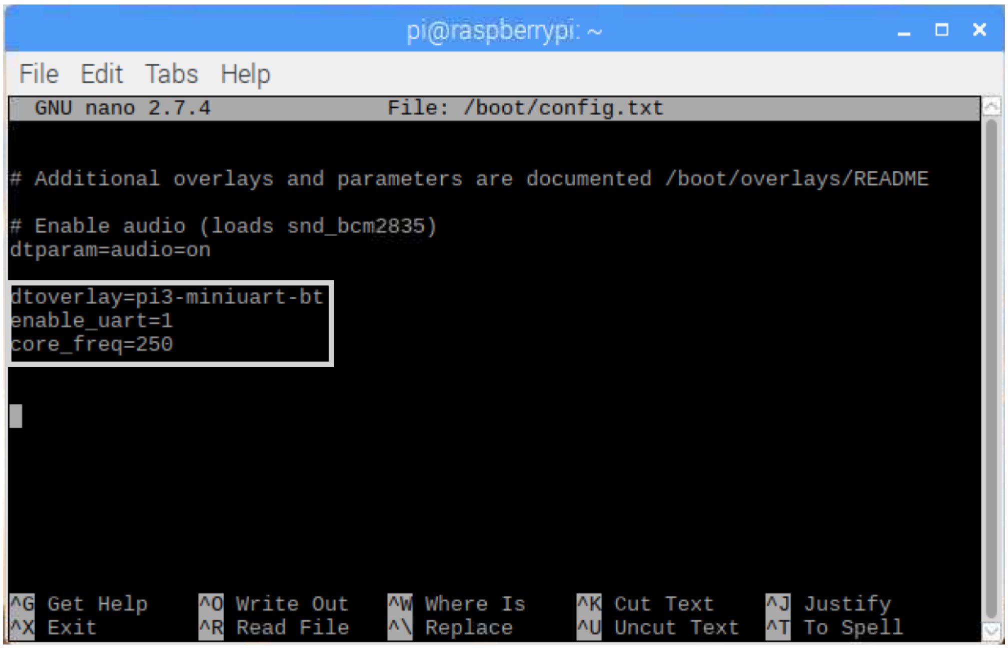
```
sudo apt-get install python-serial python3-serial
```

Then open the console on your Raspberry Pi and enter
the following command:

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

Append the following lines to the end of the file:

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



```
  GNU nano 2.7.4              File: /boot/config.txt

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

# Enable audio (loads snd_bcm2835)
dtparam=audio=on

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


^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify
^X Exit         ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell
```

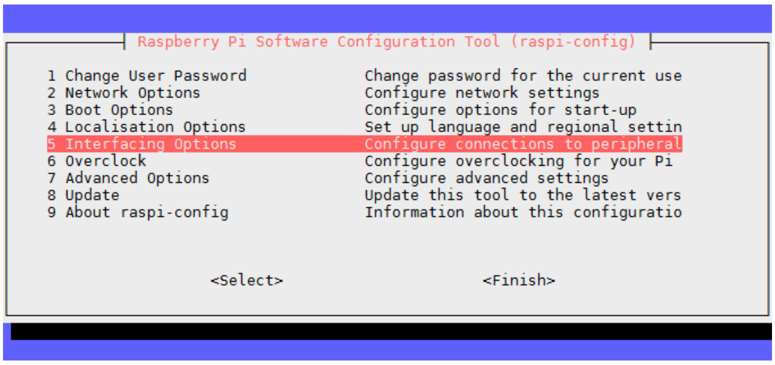
Save the file with the key combination **Ctrl+O** and exit the file with the combination **Ctrl+X**.

In addition to this, it is necessary to activate the serial port of the Raspberry Pi and deactivate the serial console. You can do this either via the Raspberry Pi *Configuration* or via the *Raspi-Config*.

To do this via the *Raspi-Config*, enter the following command into the terminal window on your Raspberry Pi :

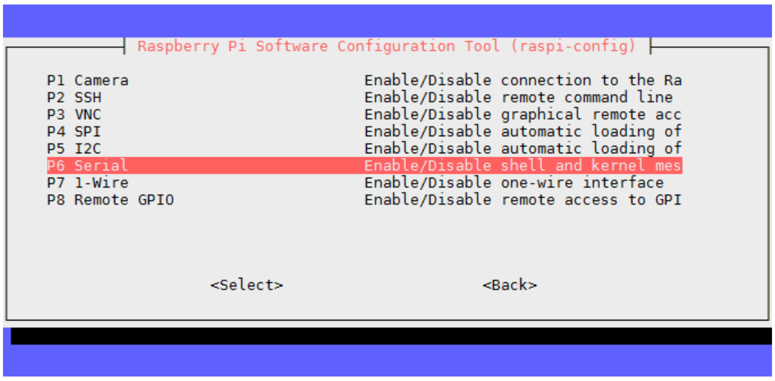
```
sudo raspi-config
```

Use the arrow keys on your keyboard to navigate to menu **5 Interfacing Options.**



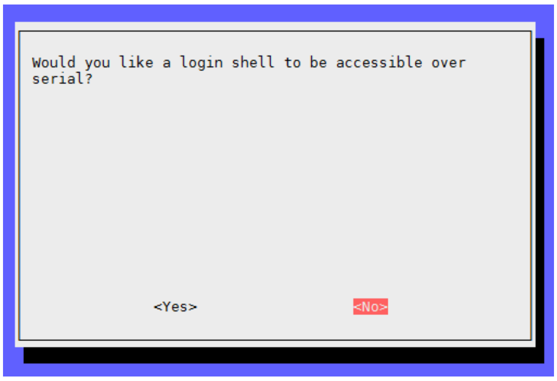
```
┤ Raspberry Pi Software Configuration Tool (raspi-config) ├

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


            <Select>                      <Finish>
```

Select the point **P6 Serial**.



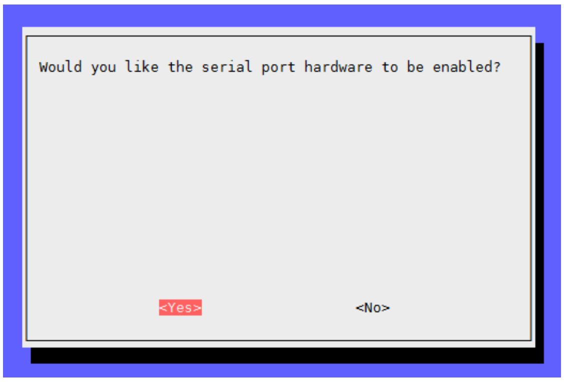
```
┤ Raspberry Pi Software Configuration Tool (raspi-config) ├

P1 Camera                       Enable/Disable connection to the Ra
P2 SSH                          Enable/Disable remote command line
P3 VNC                          Enable/Disable graphical remote acc
P4 SPI                          Enable/Disable automatic loading of
P5 I2C                          Enable/Disable automatic loading of
P6 Serial                       Enable/Disable shell and kernel mes
P7 1-Wire                       Enable/Disable one-wire interface
P8 Remote GPIO                  Enable/Disable remote access to GPI


            <Select>                      <Back>
```

The question *Would you like a login shell to be accessible over serial?* answer with **No**.



```
Would you like a login shell to be accessible over
serial?




          <Yes>                    <No>
```

The question *Would you like the serial port hardware to be enabled?* answer with **Yes**.



```
Would you like the serial port hardware to be enabled?




          <Yes>                    <No>
```

After confirming again, you can exit the menu with **Finish**. Answer the question after restarting the device with **Yes**. The serial interface is now activated and the serial console
deactivated.

8.2 Execution of the GUI script

Attention! The StromPi config script does not work with the CLI version of the StromPi firmware, only with the main version of the firmware. In addition, the serial communication must be activated
[see chapter 8.1 Automatic execution of scripts].

To execute the configuration script on your Raspberry Pi, you have to install a library. You can do this via the following command.

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

Now there are 3 possibilities how you can open the graphic
user interface.

8.3 VNC for remote desktop connection

To use VNC you have to use the command

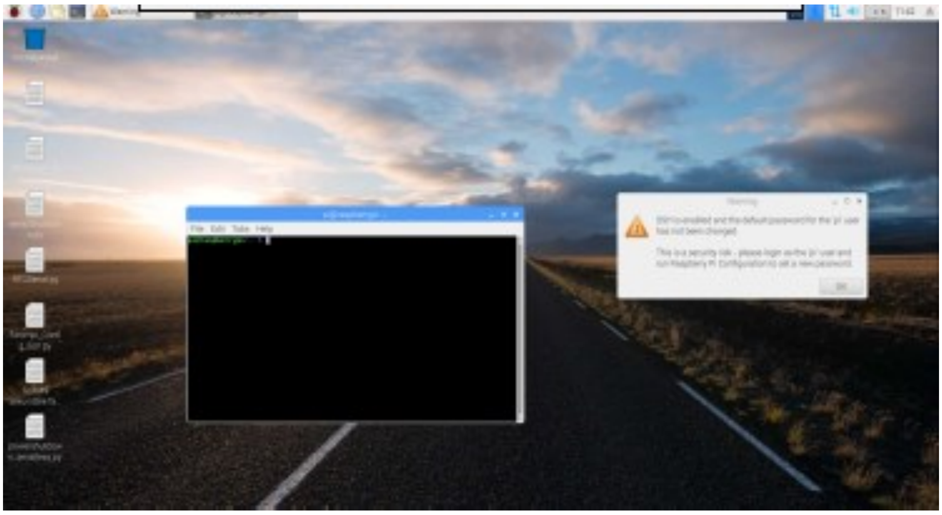
```
sudo raspi-config
```

to activate *Interfacing options*→ *VNC* . Then try to find out your IP address with the following command.

```
ifconfig
```

Download the Raspberry Pi RealVNC on the device from which you want to control and connect them by entering the IP address.

Now, the following window should appear:



SSH is enabled and the default password for the 'pi' user has not been changed.

This is a security risk - please login as the 'pi' user and run Raspberry Pi Configuration to set a new password.

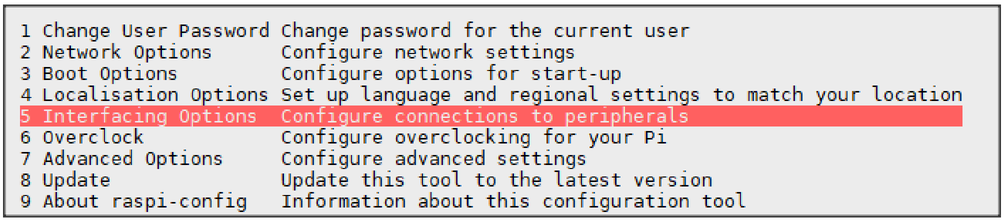
Here you can then use the console as usual with your Linux system.

8.4 SSH-connection via PuTTY and Xming

Execute the following command:

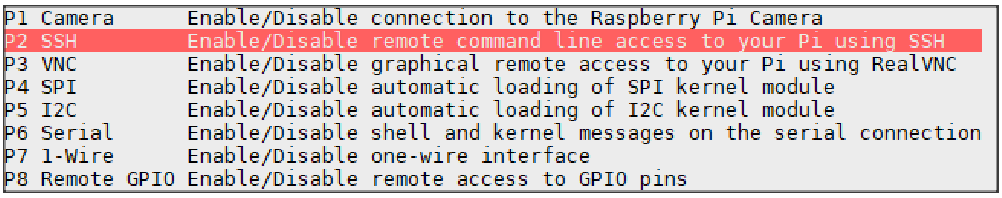
```
sudo raspi-config
```

Go to *Interfacing Options*

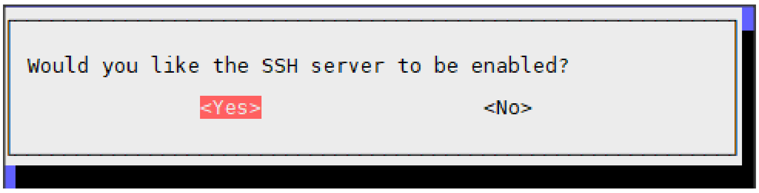


```
1 Change User Password Change password for the current user
2 Network Options      Configure network settings
3 Boot Options         Configure options for start-up
4 Localisation Options Set up language and regional settings to match your location
5 Interfacing Options  Configure connections to peripherals
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
```

and activate *SSH* afterwards.



```
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
```



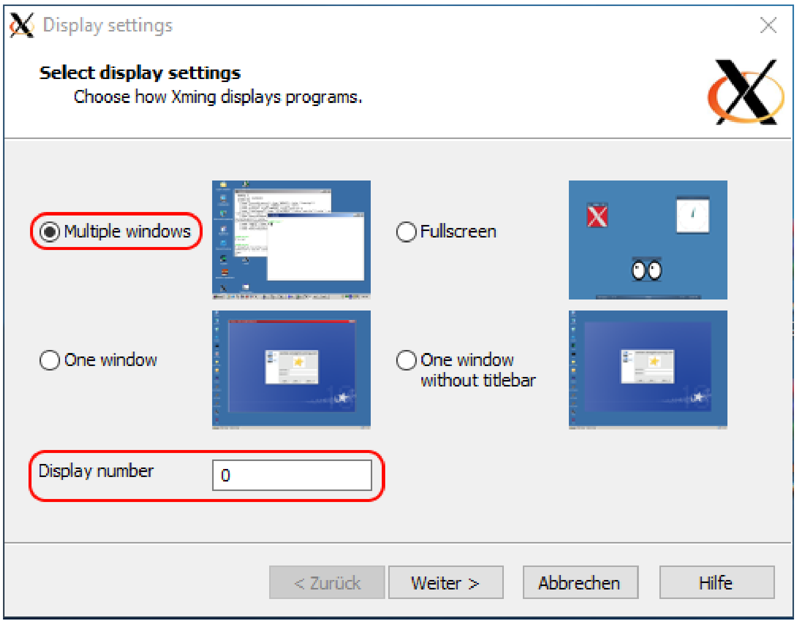
```
Would you like the SSH server to be enabled?

                 <Yes>                  <No>
```

Choose *YES.* Now you have to download PuTTY and Xming.

Xming

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

Select ***Multiple Windows*** and ***Display number*** as **0**.



Display settings

Select display settings

Choose how Xming displays programs.

● Multiple windows

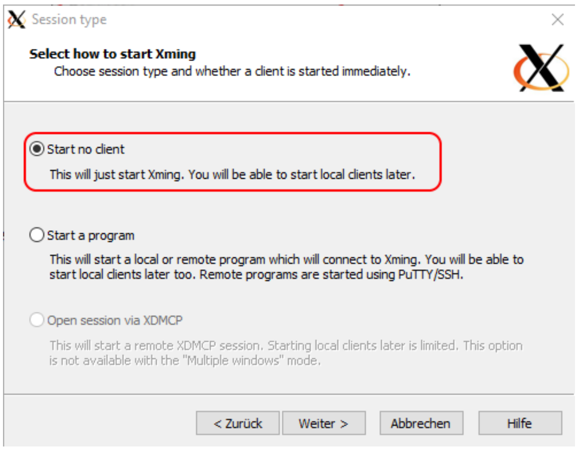
○ Fullscreen

○ One window

○ One window without titlebar

Display number: 0

Make sure that *Start no client* is selected:



Select how to start Xming

Choose session type and whether a client is started immediately.

● Start no client

This will just start Xming. You will be able to start local clients later.

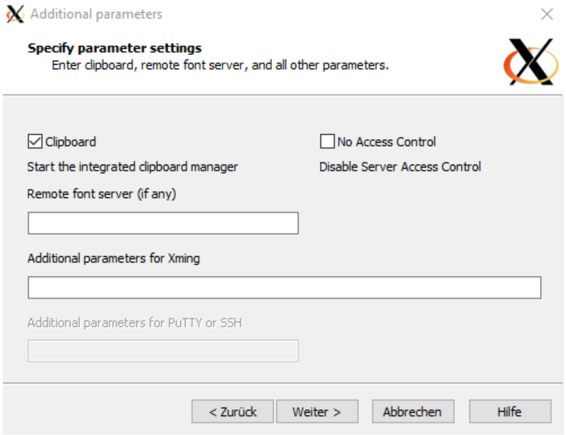
○ Start a program

This will start a local or remote program which will connect to Xming. You will be able to start local clients later too. Remote programs are started using PuTTY/SSH.

○ Open session via XDMCP

This will start a remote XDMCP session. Starting local clients later is limited. This option is not available with the "Multiple windows" mode.

Press **Next** without making any changes:



Specify parameter settings

Enter clipboard, remote font server, and all other parameters.

☑ Clipboard

Start the integrated clipboard manager

☐ No Access Control

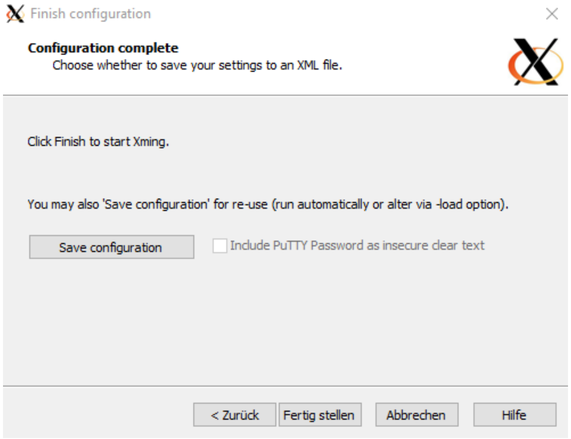
Disable Server Access Control

Remote font server (if any)

Additional parameters for Xming

Additional parameters for PuTTY or SSH

Now press **Finish**:



Configuration complete

Choose whether to save your settings to an XML file.

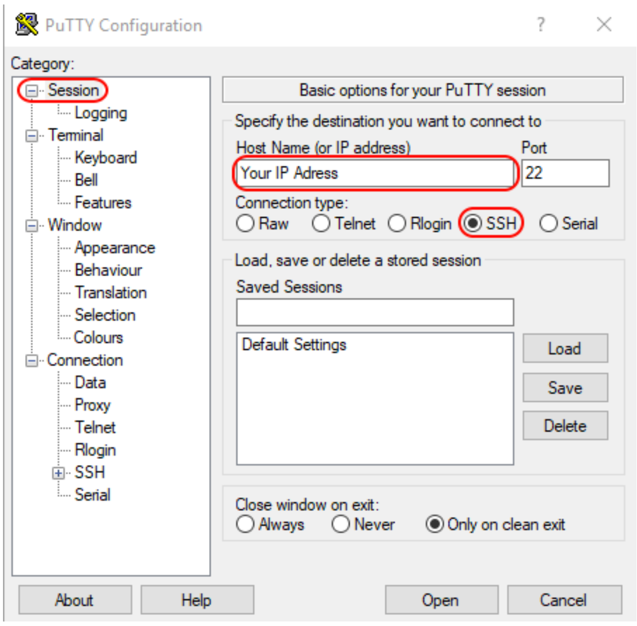
Click Finish to start Xming.

You may also 'Save configuration' for re-use (run automatically or alter via -load option).

☐ Include PuTTY Password as insecure clear text

PuTTY

Now you also have to set PuTTY.
Make sure that you have selected in *Session* → *SSH* and have entered the IP address of your Raspberry Pi.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): Your IP Adress

Port: 22

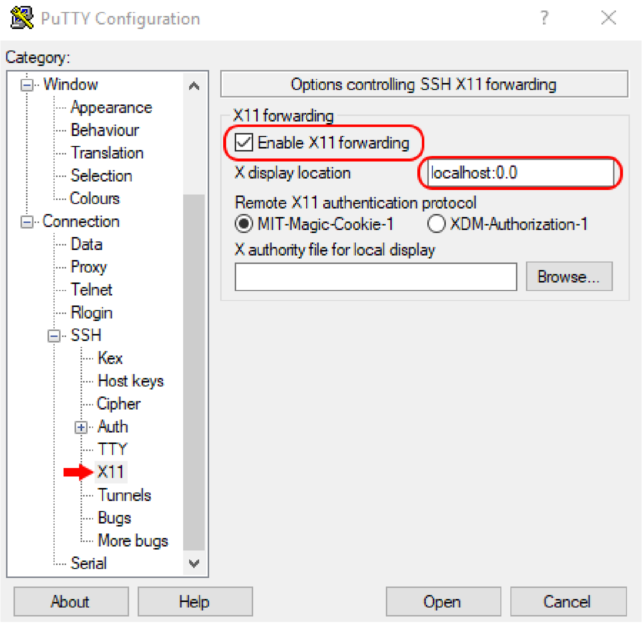
Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions: Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Now make sure that the *X11* → **Enable X11 fowarding** is switched on and enter *Xdisplay location* → **localhost:0.0** .



PuTTY Configuration

Options controlling SSH X11 forwarding

X11 forwarding

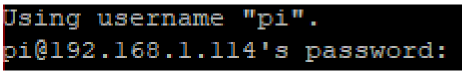
☑ Enable X11 forwarding

X display location: localhost:0.0

Remote X11 authentication protocol: ● MIT-Magic-Cookie-1 ○ XDM-Authorization-1

X authority file for local display:

Now press **Open** and you should see the following:



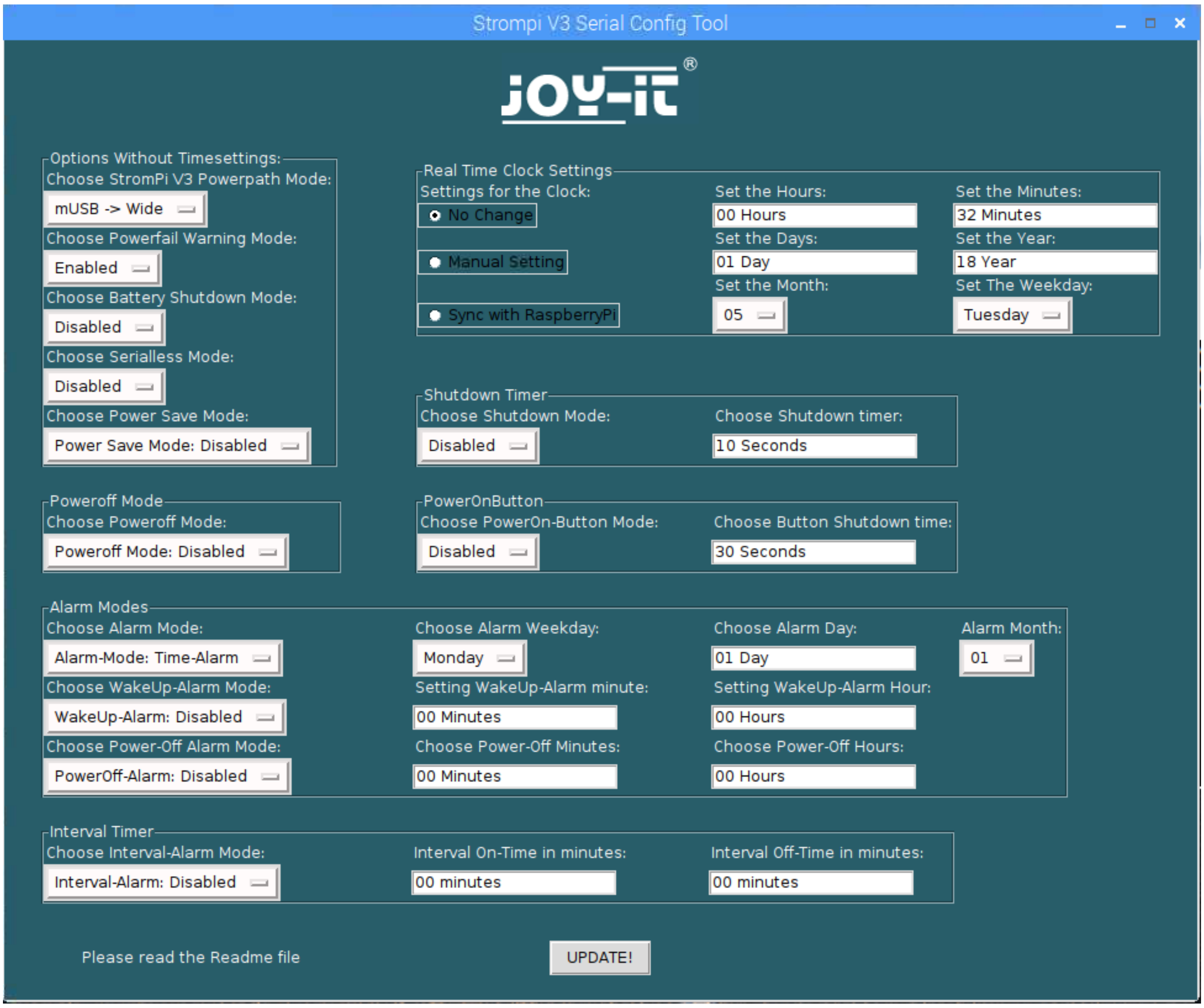
```
Using username "pi".
pi@192.168.1.114's password:
```

Now enter your login data.
You have successfully established the SSH connection.

Now enter the following command:

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

Now the graphical interface opens, which looks as the following:



Strompi V3 Serial Config Tool

Options Without Timesettings:

Choose StromPi V3 Powerpath Mode: mUSB -> Wide

Choose Powerfail Warning Mode: Enabled

Choose Battery Shutdown Mode: Disabled

Choose Serialless Mode: Disabled

Choose Power Save Mode: Power Save Mode: Disabled

Real Time Clock Settings

Settings for the Clock: ● No Change ○ Manual Setting ○ Sync with RaspberryPi

Set the Hours: 00 Hours

Set the Minutes: 32 Minutes

Set the Days: 01 Day

Set the Year: 18 Year

Set the Month: 05

Set The Weekday: Tuesday

Shutdown Timer

Choose Shutdown Mode: Disabled

Choose Shutdown timer: 10 Seconds

Poweroff Mode

Choose Poweroff Mode: Poweroff Mode: Disabled

PowerOnButton

Choose PowerOn-Button Mode: Disabled

Choose Button Shutdown time: 30 Seconds

Alarm Modes

Choose Alarm Mode: Alarm-Mode: Time-Alarm

Choose Alarm Weekday: Monday

Choose Alarm Day: 01 Day

Alarm Month: 01

Choose WakeUp-Alarm Mode: WakeUp-Alarm: Disabled

Setting WakeUp-Alarm minute: 00 Minutes

Setting WakeUp-Alarm Hour: 00 Hours

Choose Power-Off Alarm Mode: PowerOff-Alarm: Disabled

Choose Power-Off Minutes: 00 Minutes

Choose Power-Off Hours: 00 Hours

Interval Timer

Choose Interval-Alarm Mode: Interval-Alarm: Disabled

Interval On-Time in minutes: 00 minutes

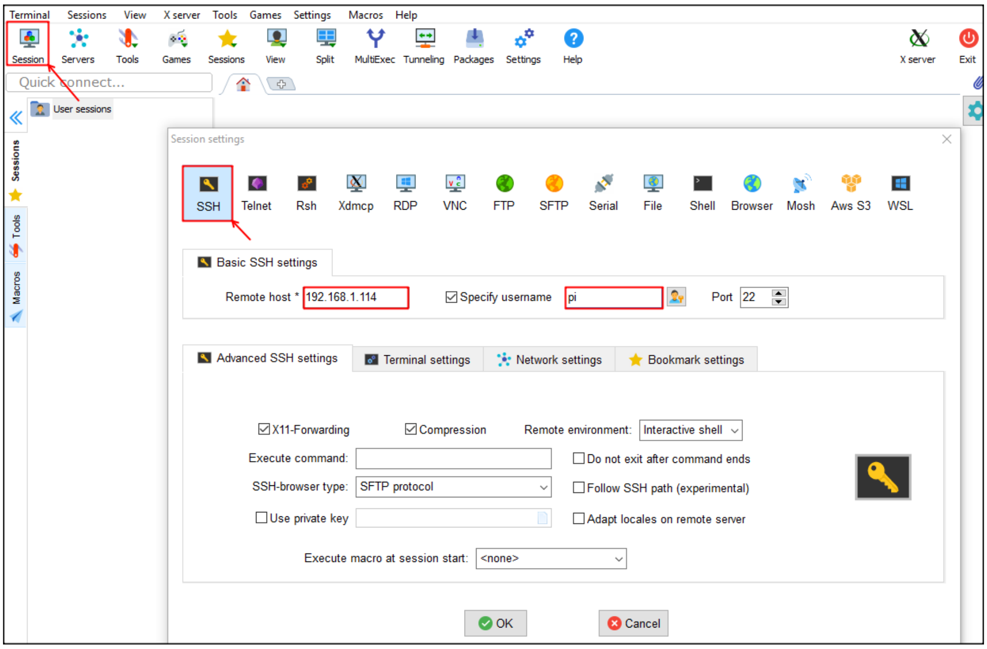
Interval Off-Time in minutes: 00 minutes

Please read the Readme file

The displayed values are synchronised with the values you set in the StromPi when the script is started. You can now change any values and then click **Update** to change the settings of your StromPi.

8.5 SSH-connection via MobaXterm

Make sure that SSH is enabled, if it is not enabled already, refer to the last chapter on how to enable SSH. Next, download MobaXTerm.



Remote host * 192.168.1.114 ☑ Specify username pi Port 22

☑ X11-Forwarding ☑ Compression Remote environment: Interactive shell

Execute command: ☐ Do not exit after command ends

SSH-browser type: SFTP protocol ☐ Follow SSH path (experimental)

☐ Use private key ☐ Adapt locales on remote server

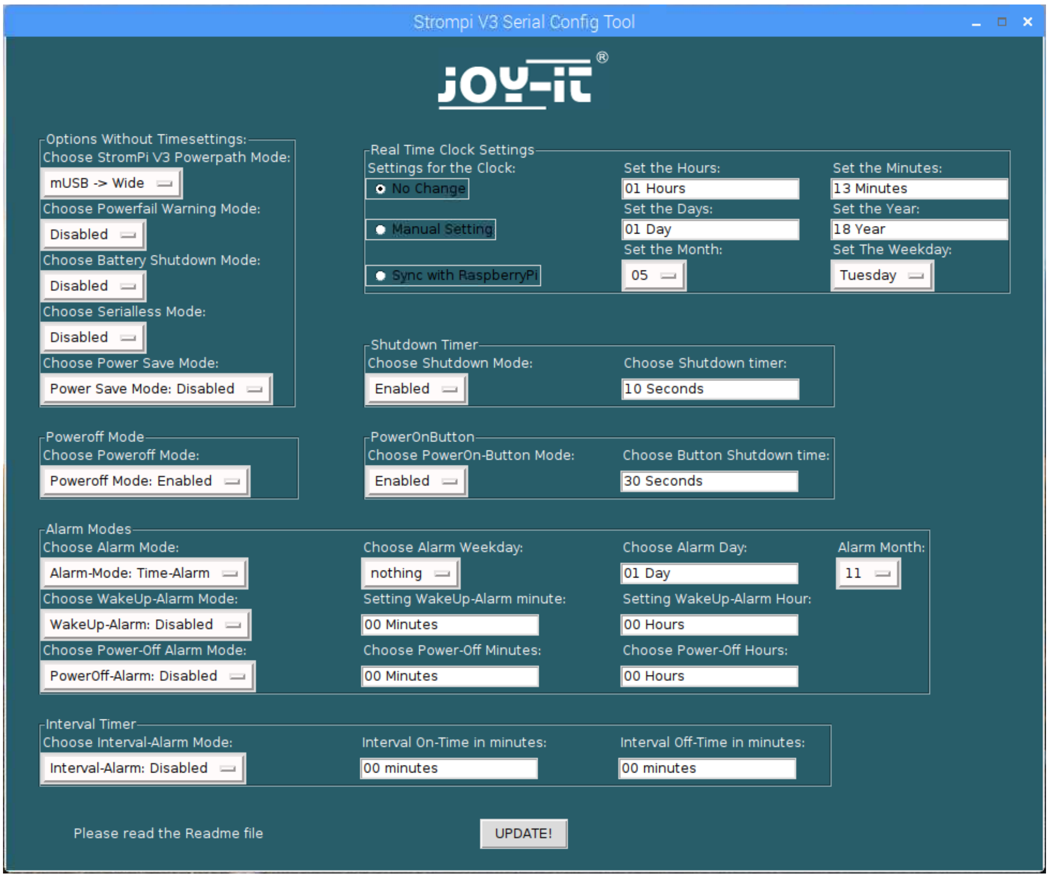
Execute macro at session start: <none>

Now press Session and then select SSH. Now enter the IP address of your Raspberry Pi at ***Remote host*** and **pi** at ***Specify username***. Then press **OK** and enter the password of your Raspberry Pi when prompted.

You can call the graphical interface with the following command:

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

The interface will look like this:



Options Without Timesettings:
Choose StromPi V3 Powerpath Mode: mUSB -> Wide
Choose Powerfail Warning Mode: Disabled
Choose Battery Shutdown Mode: Disabled
Choose Serialless Mode: Disabled
Choose Power Save Mode: Power Save Mode: Disabled

Real Time Clock Settings
Settings for the Clock: ● No Change ○ Manual Setting ○ Sync with RaspberryPi
Set the Hours: 01 Hours
Set the Minutes: 13 Minutes
Set the Days: 01 Day
Set the Year: 18 Year
Set the Month: 05
Set The Weekday: Tuesday

Shutdown Timer
Choose Shutdown Mode: Enabled
Choose Shutdown timer: 10 Seconds

Poweroff Mode
Choose Poweroff Mode: Poweroff Mode: Enabled

PowerOnButton
Choose PowerOn-Button Mode: Enabled
Choose Button Shutdown time: 30 Seconds

Alarm Modes
Choose Alarm Mode: Alarm-Mode: Time-Alarm
Choose Alarm Weekday: nothing
Choose Alarm Day: 01 Day
Alarm Month: 11
Choose WakeUp-Alarm Mode: WakeUp-Alarm: Disabled
Setting WakeUp-Alarm minute: 00 Minutes
Setting WakeUp-Alarm Hour: 00 Hours
Choose Power-Off Alarm Mode: PowerOff-Alarm: Disabled
Choose Power-Off Minutes: 00 Minutes
Choose Power-Off Hours: 00 Hours

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

Please read the Readme file

The displayed values are synchronised with the values you set in the StromPi when the script is started. You can now change any values and then click on **Update** to change the settings of your StromPi.

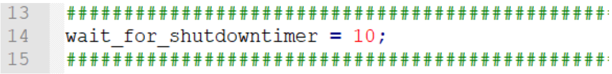
It is possible that your script looks different. The Poweroff Mode and the Power Save Mode were added on a later firmware version.

9. STROMPI 3 SCRIPTS

You can download all StromPi scripts published by us here. The serial scripts read and send the required information via the serial port, the serialless scripts use a GPIO pin instead. Besides the ConfigSkript, which you already know, there are the following scripts:

9.1 Serial scripts

serialShutdown.py
This script shuts down the Raspberry Pi when the primary power supply fails. The script has its own shutdown timer. It specifies the time to wait if the primary power supply gets back before the Raspberry Pi is shut down. The preset is set to 10 seconds. You can adjust the time in line 14 of the script.



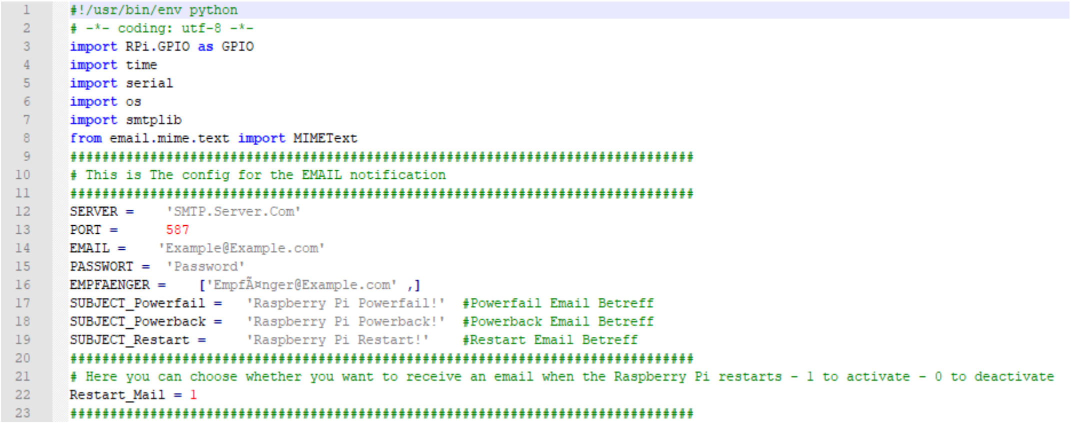
```
13  ################################################
14  wait_for_shutdowntimer = 10;
15  ################################################
```

The shutdown timer in the script must be smaller than the Shutdown Timer you set in the config script, otherwise, the power supply of the Raspberry Pis will be interrupted before it has been shut down. You must have this script in the Autostart of your Raspberry Pi to make it work. You will find instructions on how to do this in chapter 9.4 Automatic execution of scripts.

shutdown.py
This script shuts down your StromPi and Raspberry Pi via the Poweroff command when executed.
The Poweroff command allows starting the StromPi via the WakeUp-Alarm.

RTCSerial.py
This script compares the time of the Raspberry Pi with the time of the RTC of the StromPi at the start and the older one will be overwritten by the more recent one. If you put this script into the Autostart of your Raspberry Pi, the time will be compared automatically every time you start the Raspberry Pi. For instructions, refer to chapter 9.4 Automatic execution of scripts.

Poweralarm_serial.py
This script sends an email to the email address specified in the script if the primary power supply fails. Another email will be sent when the primary power supply is available again or when the Raspberry Pi is restarted.



```
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  PASSWORT =  'Password'
16  EMPFAENGER =    ['EmpfÃ¤nger@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 you can turn the notification mail on (1) or off (0) when restarting the Switch system.

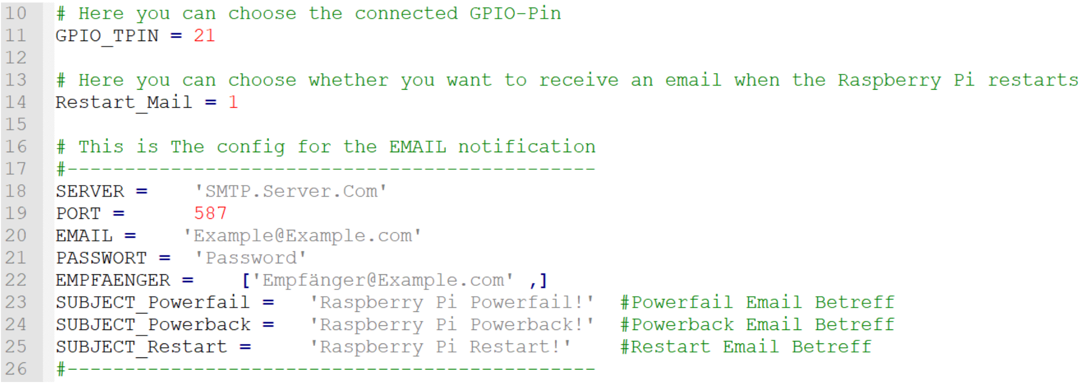
SERVER is your SMTP e-mail server (e.g. smtp.your-e-mail-server.com)
PORT is your server port
EMAIL is your email address
PASSWORT is your email password
EMPFAENGER are the email addresses that receive the emails
SUBJECT_Powerfail is entered in the subject of the power loss e-mail
SUBJECT_Poweback is entered in the subject of the powerback e-mail

You must have this script in the Autostart of your Raspberry Pi to make it work.
You will find instructions on how to do this in chapter 9.4 Automatic execution of scripts.

In chapter 9.3 Raspberry Pi Pin-Layout, you can take a look at the pin assignment of the Raspberry Pi.

9.2 Serialless scripts

poweralarm_serialless.py
This script compares the time of the Raspberry Pi with StromPi and overwrites the older with the newer time. Additionally, it sends an email to the email address given in the script if the primary power supply fails. Another email will be sent when the primary power supply is available again or when the Raspberry Pi has been restarted.



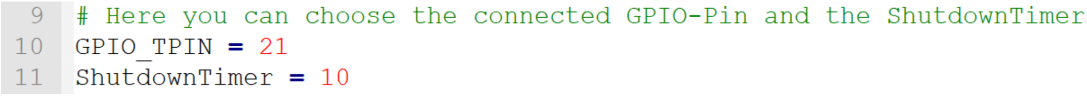
```
10  # Here you can choose the connected GPIO-Pin
11  GPIO_TPIN = 21
12
13  # Here you can choose whether you want to receive an email when the Raspberry Pi restarts
14  Restart_Mail = 1
15
16  # This is The config for the EMAIL notification
17  #-----------------------------------------------
18  SERVER =     'SMTP.Server.Com'
19  PORT =       587
20  EMAIL =     'Example@Example.com'
21  PASSWORT =  '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 you use for alternative communication for the serialless mode. In line 14 you can turn the notification mail on (1) or off (0) at system reboot.

SERVER is your SMTP e-mail server (e.g. smtp.your-e-mail-server.com)
PORT is your server port
EMAIL is your email address
PASSWORT is your email password
EMPFAENGER are the email addresses that receive the emails
SUBJECT_Powerfail is entered in the subject of the power loss e-mail
SUBJECT_Poweback is entered in the subject of the power back e-mail

You must have this script in the autostart of your Raspberry Pi to make it work. See chapter 9.4 Automatic execution of scripts.
In chapter 9.3 Raspberry Pi Pin-Layout you can view the pin assignment of the Raspberry Pi.

powershutdown_serialless.py
This script shuts down the Raspberry Pi when the primary power supply fails. The script has its own shutdown timer, it specifies the time to wait if the primary power supply is back before the Raspberry Pi
is shut down. This is pre-set to 10 seconds. You can adjust the time in line 11 of the script, as well as the GPIO pin that you can use for alternative communication for the serialless mode use (line 10).



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

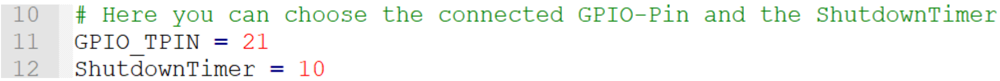
The shutdown timer in the script must be smaller than the Shutdown Timer which you specify in the config script, otherwise, the power supply to the Raspberry Pi will be interrupted before it is shut down.

In chapter 9.3 Raspberry Pi Pin-Layout you can take a look at the pin assignment of the Raspberry Pi.
You must have this script in the autostart of your Raspberry Pi to make it work. You will find instructions on how to do this in chapter 9.4 Automatic execution of scripts.

RTC_serialShutdown_serialless.py

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

The Shutdown Timer and the GPIO pin that you use for the alternative communication in the Serialless Mode, can be changed in lines 11 and 12.



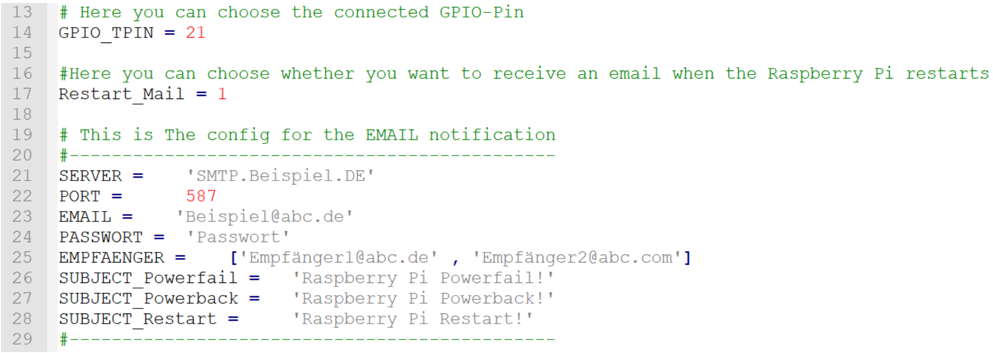
```
10 # Here you can choose the connected GPIO-Pin and the ShutdownTimer
11 GPIO_TPIN = 21
12 ShutdownTimer = 10
```

The shutdown timer in the script must be smaller than the Shutdown Timer you specify in the config script, otherwise the power supply to the Raspberry Pi will be interrupted before it is shut down.

In chapter 9.3 Raspberry Pi Pin Layout you can take a look at the pin assignment of the Raspberry Pi.
You must have this script in the autostart of your Raspberry Pi for it to work. You will find instructions on how to do this in chapter 9.4 Automatic execution of scripts.

RTC_Email_serialless.py

This script compares the time of the Raspberry Pi with StromPi and overwrites the older with the newer time. In addition, if the primary power supply fails, it sends an email to the email address given in the script. Another email is sent when the primary power supply is available again, or when the Raspberry Pi has been restarted.



```
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 PASSWORT =  '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 you use for alternative communication for the Serialless Mode. In line 17 you can turn the notification mail on (1) or off (0) at system reboot.

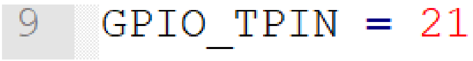
SERVER is your SMTP e-mail server (e.g. smtp.your-e-mail-server.com)
PORT is your server port
EMAIL is your email address
PASSWORT is your email password
EMPFAENGER are the email addresses that receive the emails
SUBJECT_Powerfail is entered in the subject of the power loss e-mail
SUBJECT_Poweback is entered in the subject of the power back e-mail

In chapter 9.3 Raspberry Pi Pin-Layout you can take a look at the pin assignment of the Raspberry Pi.
You must have this script in the autostart of your Raspberry Pi for it to work. You will find instructions on how to do this in chapter 9.4 Automatic execution of scripts.

RTCSerialless.py

This script was written especially for the serialless mode. When the script is started, the serialless mode is terminated, so that the time of the Raspberry Pi and the StromPi can be compared and the older one will be overwritten with the more current time. Afterwards, the serial mode is reactivated.

Because the serial communication is now deactivated, you cannot make further changes to the configuration of the StromPi. In line 9 of the script, you can change the GPIO pin used for alternative communication for the serialless mode.



```
9   GPIO_TPIN = 21
```

To end the serialless mode the communication pin must be set to low for at least 5 seconds. In chapter 9.3 Raspberry Pi Pin-Layout, you can view the pin assignment of the Raspberry Pi.

If you put this script into the autostart of your Raspberry Pi, the time will be compared automatically every time you start the Raspberry Pi. And the serialless mode will be activated afterwards.

You will find instructions on how to do this in chapter 9.4 Automatic execution of scripts.

Start_Serialless.py

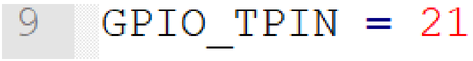
This script starts the serialless mode to suppress all serial communication between the Raspberry Pi and the StromPi. In line 9 of the script, you can change the GPIO pin you use for the alternative communication in the Serialless Mode.

In chapter 9.3 Raspberry Pi Pin-Layout, you can view the pin assignment of the Raspberry Pi.

Stop_Serialless.py

This script stops the serialless mode to enable serial communication between the Raspberry Pi and the StromPi again.

In line 9 of the script, you can change the GPIO pin you use for alternative communication in the Serialless Mode.



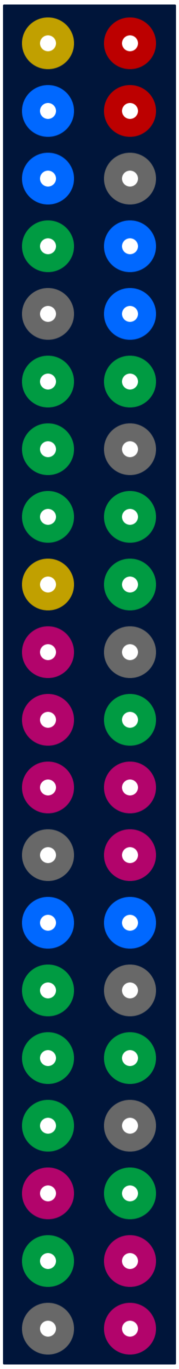
```
9   GPIO_TPIN = 21
```

In chapter 9.3 Raspberry Pi Pin-Layout, you can take a look at the pin assignment of the Raspberry Pi.

9.3 Raspberry Pi Pin-Layout

Here you can see the pin assignment of the Raspberry Pi:

1	3.3 V DC	**2**	5 V DC
3	GPIO 2 (SDA1, I2C)	**4**	5 V 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	3.3 V	**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



9.4 Automatic execution of scripts

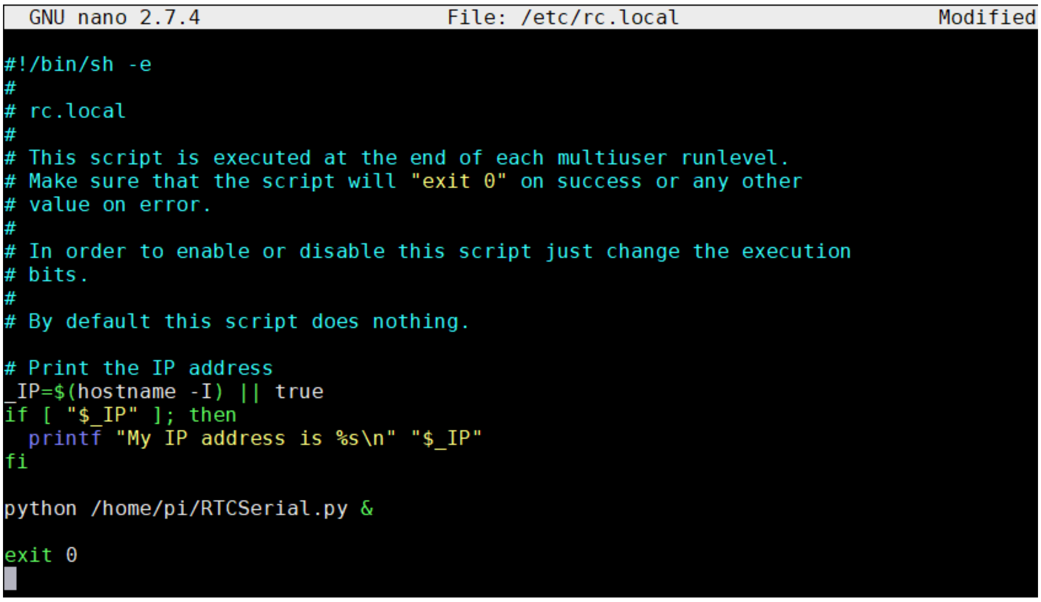
Scripts can be added to the autostart of your Raspberry Pis to be executed when the Raspberry Pis is started:

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

Here you can add the script to the autostart. The following command must be inserted before the ***exit 0*** command.

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

Example with 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
```

If you have not copied the script to the */home/pi* folder, you will need to change the path accordingly.
Save the file with the key combination **CTRL + O**, confirm with **Enter** and leave the editor with the combination **CTRL + X**.

The changes will take effect after a restart.

10. MODIFICATIONS

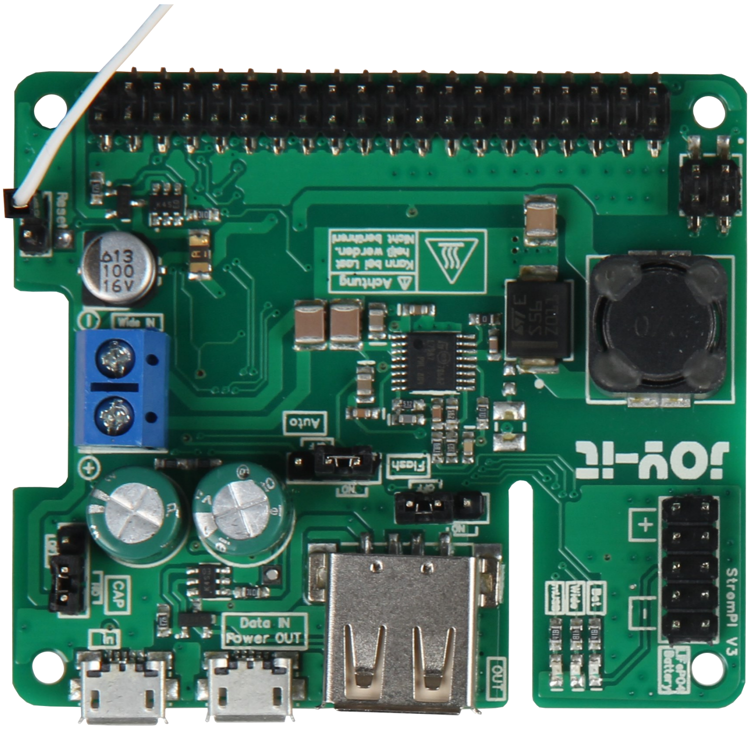
In this chapter, we describe all modifications that are necessary for some functions of the StromPi.

Serialless Mode

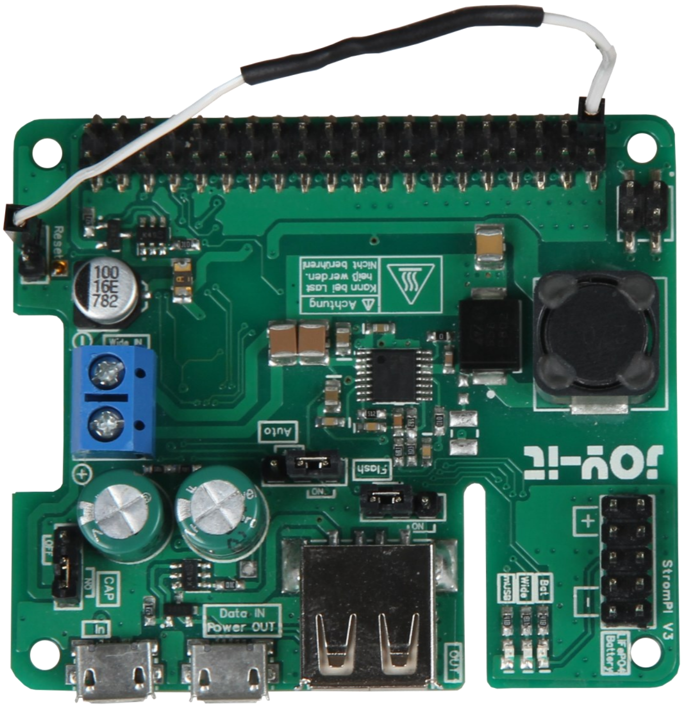
For the modification for the Serialless Mode, there are 3 different variants, which are explained in the following.

Variant1

Variant 1 is the fastest solution, but can only be used if you are not using a battery hat.
Just plug a cable to the reset pin as shown in the following picture.
To do this, you have to remove the jumper.

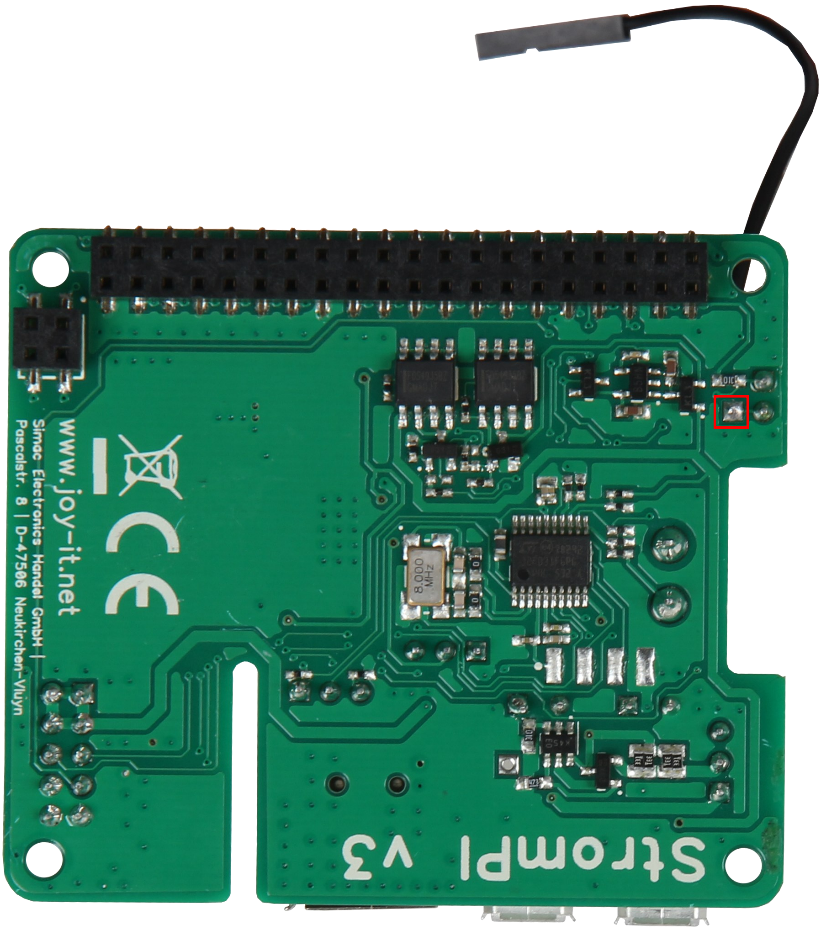


The other end of the cable must be connected to a GPIO pin.

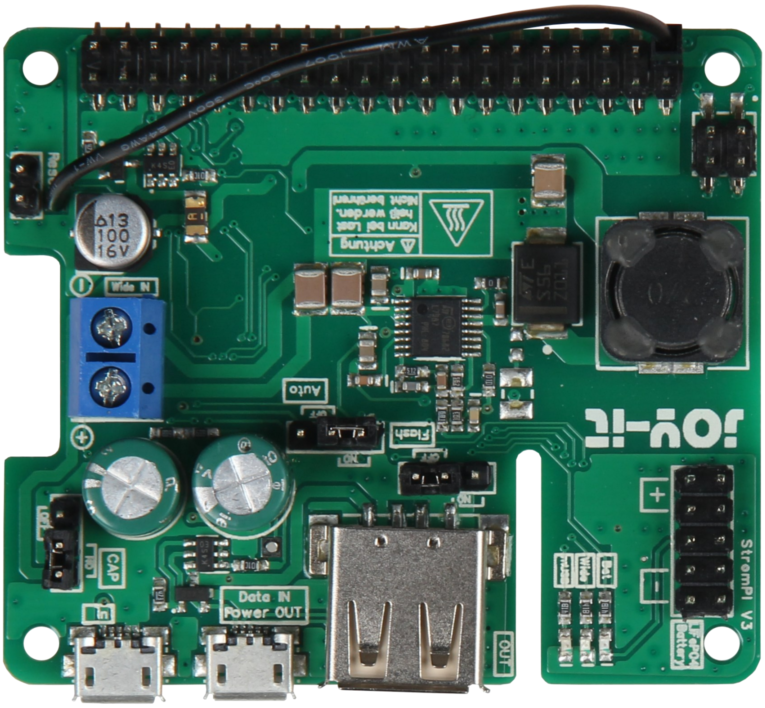


You can use any other GPIO pin, but the software must be adapted accordingly.

Variant 2
Solder one end of the cable to the free pad as shown in the picture. Note that the jumper on the reset pin must be set.

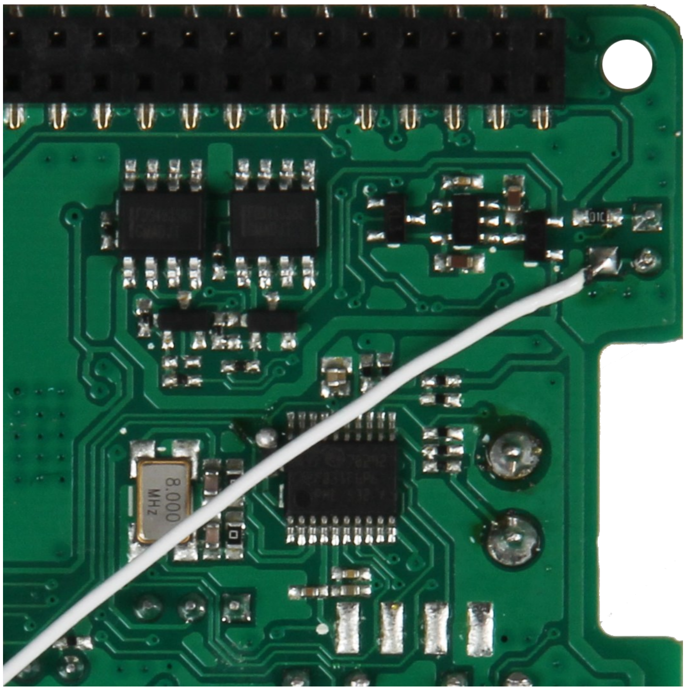

www.joy-it.net
Simac Electronics Handel GmbH | Pascalstr. 8 | D-47506 Neukirchen-Vluyn
StromPI v3

The other end of the cable is connected to a free GPIO pin.

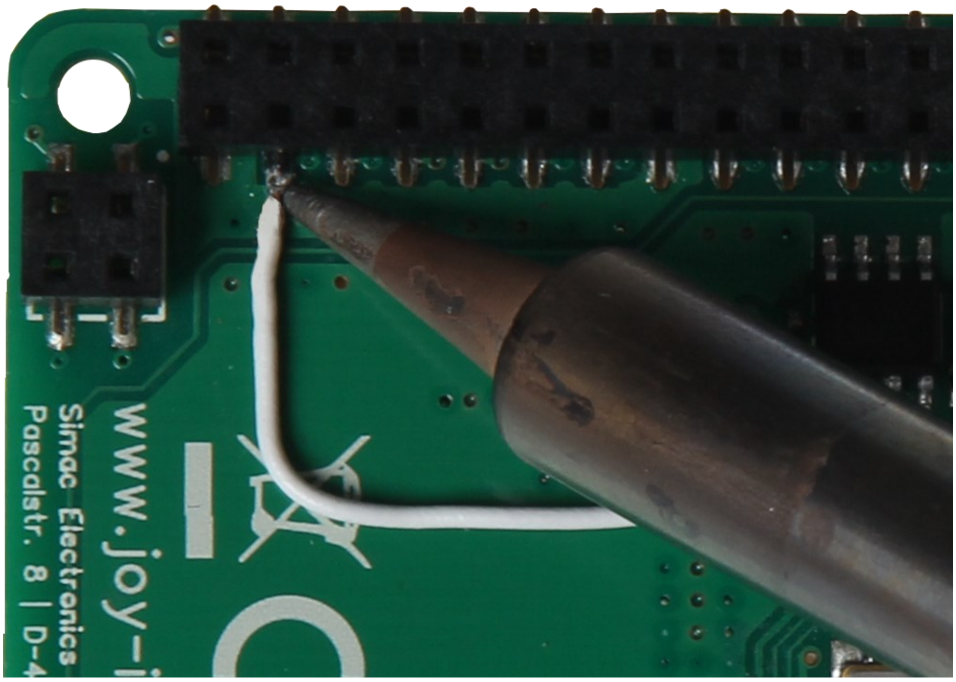


Variant 3

For variant 3, solder one end of the cable to the free pad of the reset pin, like in variant 2. Again, the jumper must remain set.

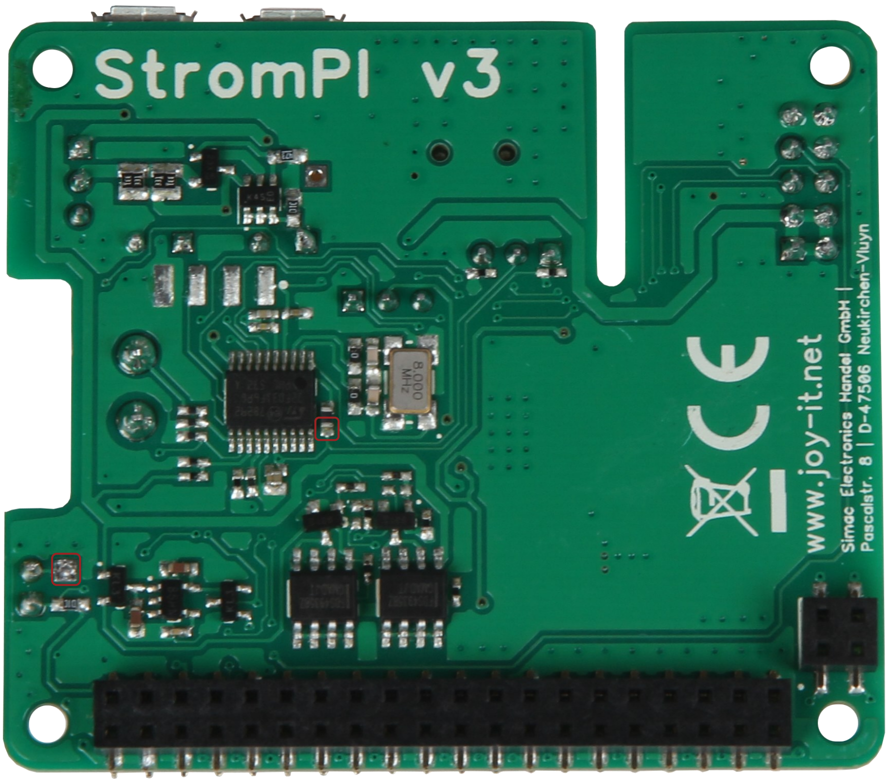


Solder the other end of the cable as shown in the picture below to one of the GPIO pins.

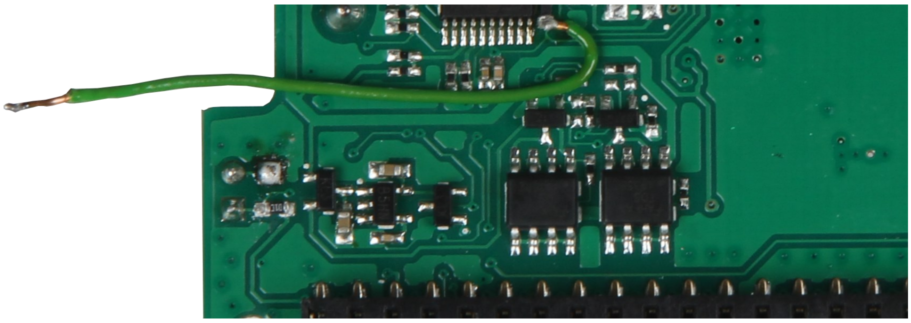


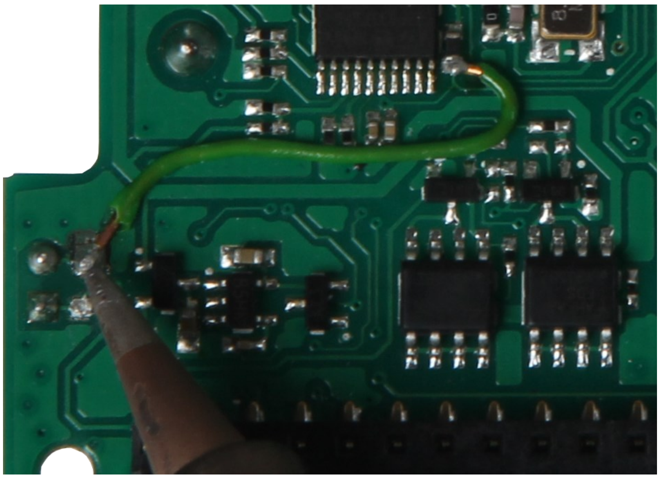
PowerOn-Button

The two solder pads marked in red must be bridged with a cable.


StromPI v3
www.joy-it.net
Simac Electronics Handel GmbH |
Pascalstr. 8 | D-47506 Neukirchen-Vluyn
8.000 MHz

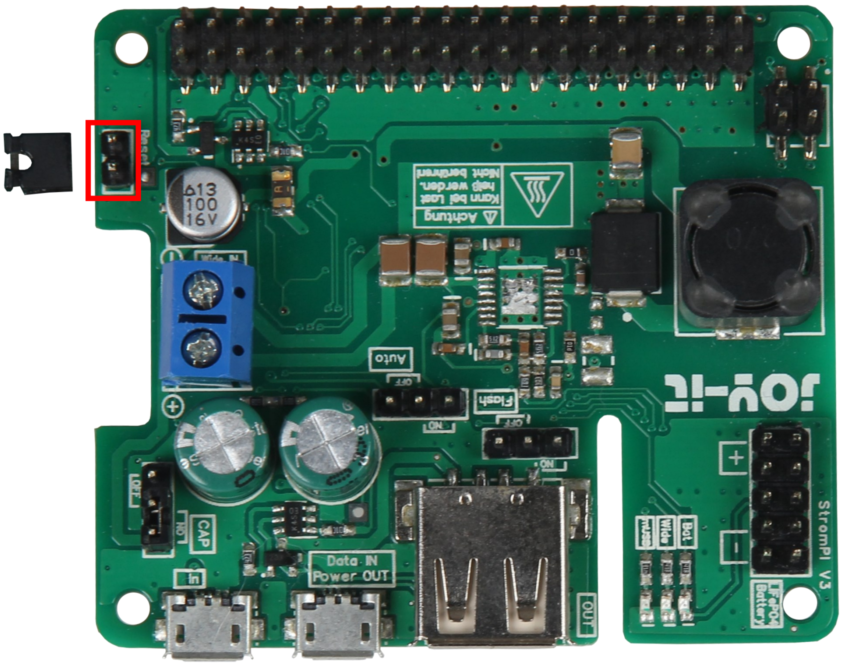
First apply some solder to both pads. Now solder one end of the cable to the upper pad.





Finally the other end of the cable is soldered to the lower solder pad.

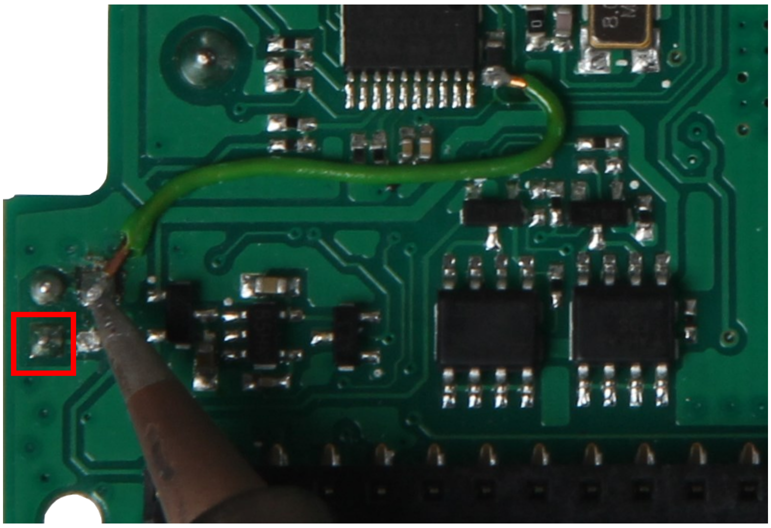
Now you can start the system after a Poweroff - command by short-circuit the reset pins marked in red and restart. In operation the pins must not be bridged. For more comfort it is possible connect a push button (normally open contact) to the pins in order to short circuit them at the restart.

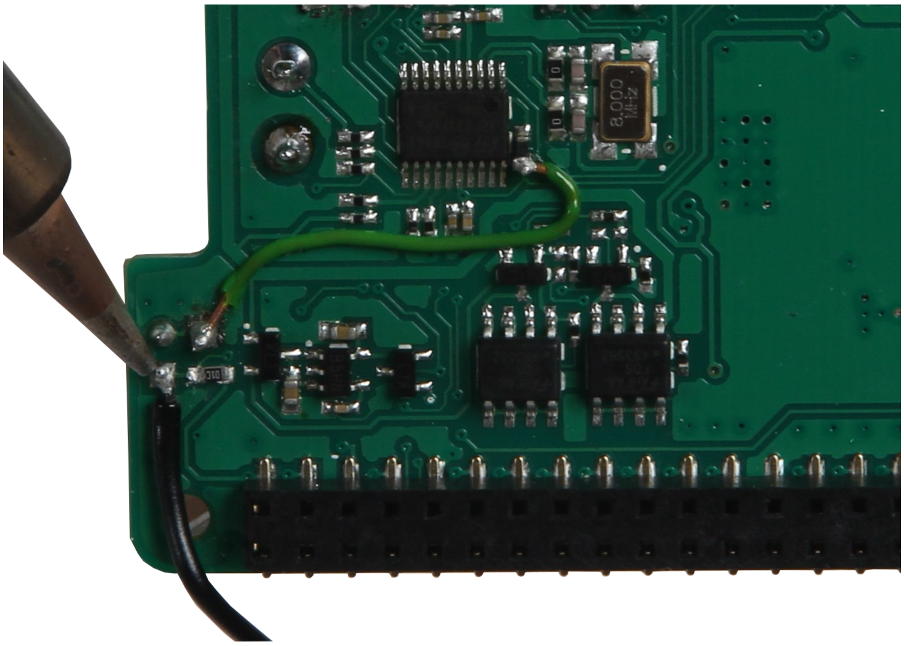


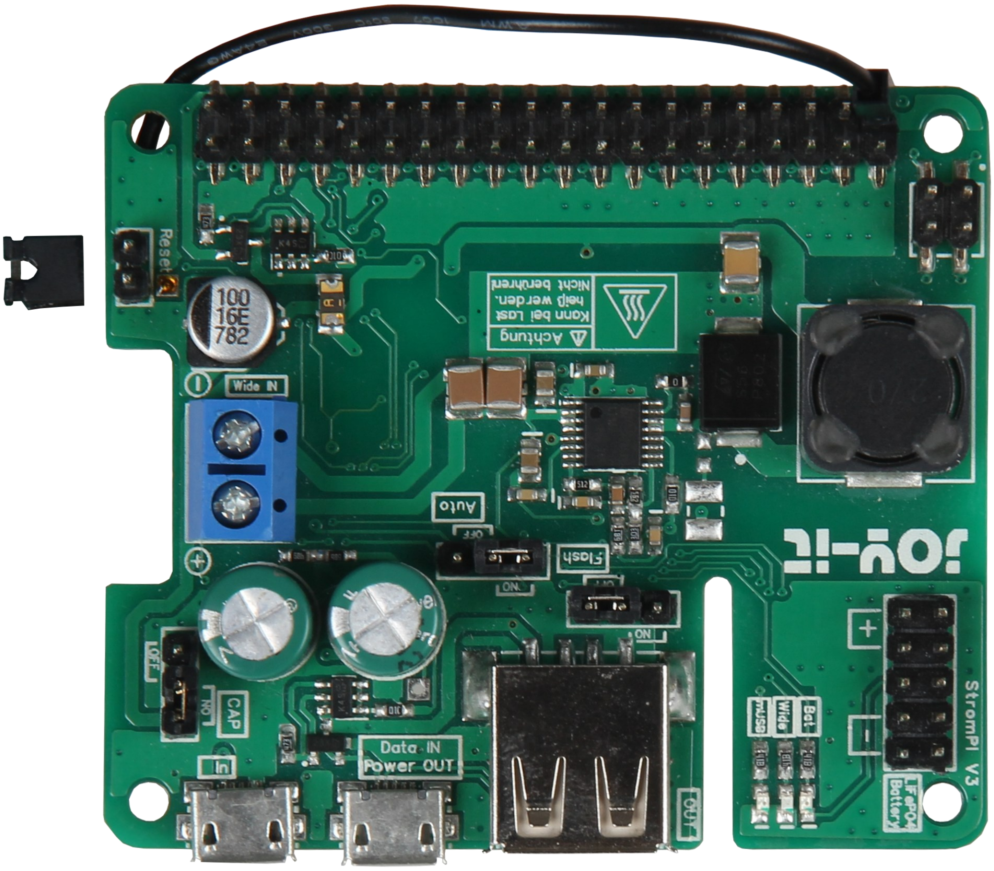
IMPORTANT! There must be at least 30 seconds between Poweroff and restart.

Serialless and PowerOn-Button

Since the pad which is normally used for the serialless modification is occupied by the poweroff modification, the cable for the serialless mode must now be soldered to the red marked pad.







For more information about the Serialless Mode please read the **Serialless instructions**.

11. OPENING THE SERIAL CONSOLE

To make settings manually via the StromPi console, you must download and install Minicom. To do so, enter the following commands:

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

After Minicom has been successfully installed, you can establish a serial connection to the StromPi 3 with the following command:

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

Attention! The system is now waiting for another input and may not display your next keystrokes.

Enter the following command and confirm it with **Enter**, even though you may not be able to see it:

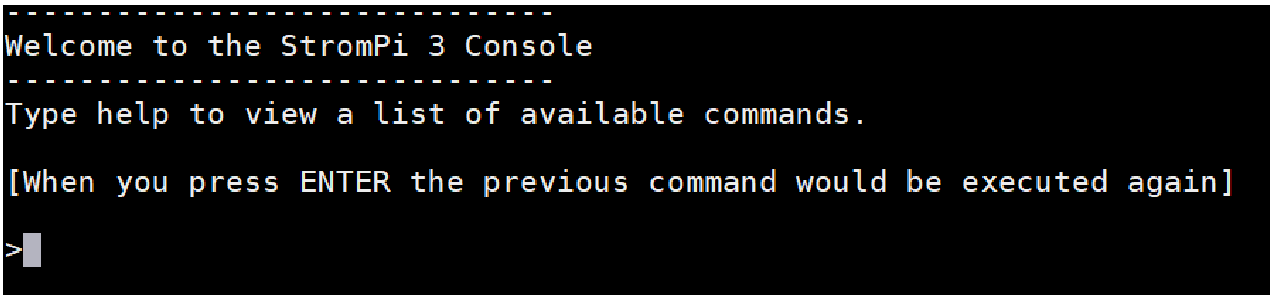
Open the serial console of the StromPi 3 with following command:

```
startstrompiconsole
```

- … or the abbreviation :

```
sspc
```

The serial console of the StromPi 3 now opens. The current configuration setting of your device can be seen 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]

>
```

The **help** command provides a general overview.

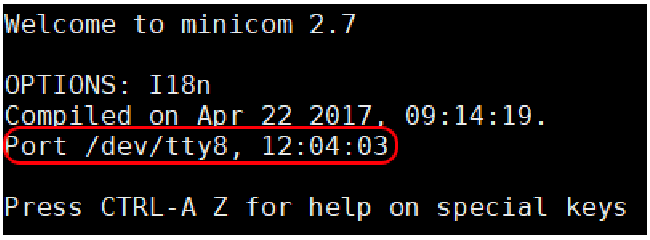
Use the key combination **CTRL + A / X / Enter** to close Minicom.

You can scroll through the terminal window with the key combinations **SHIFT + Page up / Page down**. The mouse wheel also works.

After entering the following command, ...

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

… should there be displayed *Port/dev/tty8* (like in the picture) and not *Port/dev/serial0,* then you have to configure Minicom manually.

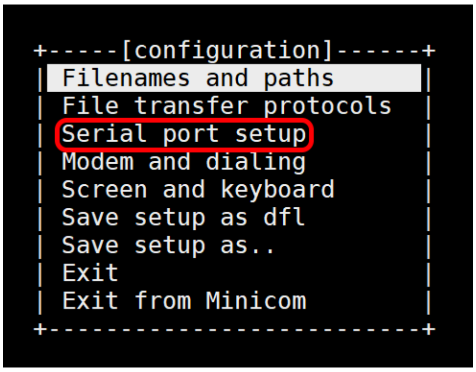


```
Welcome to minicom 2.7

OPTIONS: I18n
Compiled on Apr 22 2017, 09:14:19.
Port /dev/tty8, 12:04:03

Press CTRL-A Z for help on special keys
```

To achieve this, enter the following command:

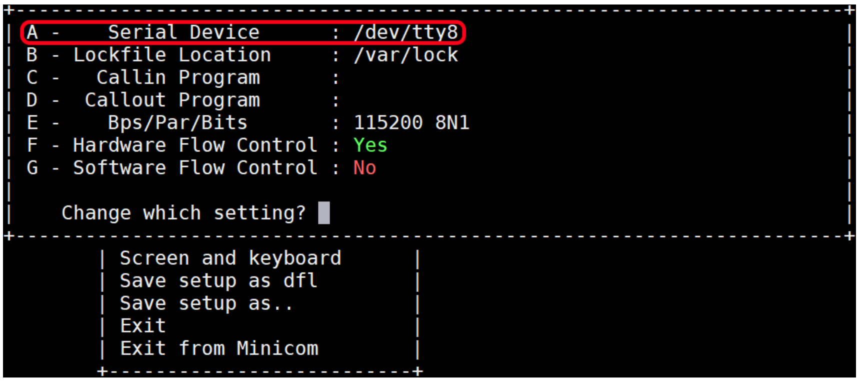
```
sudo minicom -s
```

You will now see the following window:



```
+-----[configuration]------+
| Filenames and paths      |
| File transfer protocols  |
| Serial port setup        |
| Modem and dialing        |
| Screen and keyboard      |
| Save setup as dfl        |
| Save setup as..          |
| Exit                     |
| Exit from Minicom        |
+--------------------------+
```

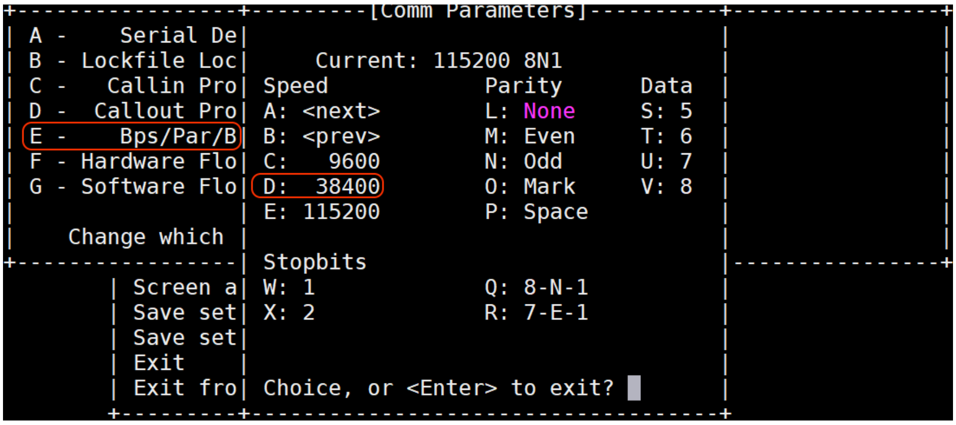
Select ***Serial port setup*** here. Then the following window appears:



```
+-----------------------------------------------------------------------+
| A -    Serial Device      : /dev/tty8                                 |
| B - Lockfile Location     : /var/lock                                 |
| C -   Callin Program      :                                           |
| D -  Callout Program      :                                           |
| E -    Bps/Par/Bits       : 115200 8N1                                |
| F - Hardware Flow Control : Yes                                       |
| G - Software Flow Control : No                                        |
|                                                                       |
|    Change which setting?                                              |
+-----------------------------------------------------------------------+
        | Screen and keyboard      |
        | Save setup as dfl        |
        | Save setup as..          |
        | Exit                     |
        | Exit from Minicom        |
        +--------------------------+
```

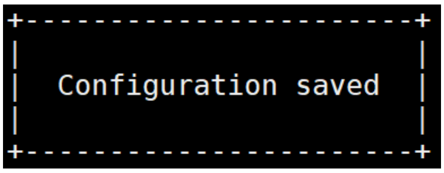
Now press **A** and change */dev/tty8* to */dev/serial0*.
Then confirm with **Enter**.

Now enter **E** and the following window appears:



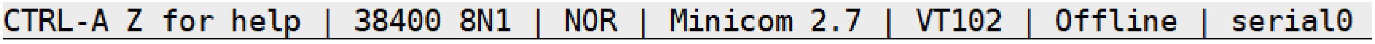
```
+-----------------+---------[Comm Parameters]----------+----------------+
| A -     Serial De|                                    |                |
| B - Lockfile Loc|     Current: 115200 8N1            |                |
| C -    Callin Pro| Speed            Parity      Data  |                |
| 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? █      |
        +---------+------------------------------------+
```

Now press **D** to set a baud rate of 38400.Then confirm twice with **Enter** to enter the configuration menu. Then press ***Save as dfl*** and this window appears as confirmation:



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

You can now exit the configuration menu with **ESC** and then press **CTRL + A**. Now you see in the lower part of the console the following:



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

If everything looks like in the picture above, Minicom is configured correctly. Now you can type the following command to start Minicom:

```
sudo minicom
```

Now you can open the StromPi 3 console with following command...

```
startstrompiconsole
```

- … or the abbreviation .

```
sspc
```

On the following pages all StromPi console commands are listed.

12. COMMANDS FOR THE CONSOLE

Commands for the CLI and Main Firmware versions

StromPi - Start console:

```
startstrompiconsole
```

or

```
sspc
```

StromPi - Exit console:

```
quit
```

List all commands:

```
help
```

Setting the time of the RTC of the StromPi:

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

Setting the date of the RTC of the StromPi:

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

Output of the measured voltages:

```
adc-output
```

Setting the StromPi mode:
(<1>: mUSB → Wide, <2>: Wide → mUSB, <3>: mUSB → Bat, <4>: Wide → Bat, <5>: mUSB → Wide, <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
```

Switch off the Raspberry Pi with the StromPi:

```
poweroff
```

Commands only for the CLI firmware version

Set the Shutdown Timer in seconds:

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

Enable <1> or disable <0> the Raspberry Pi Shutdown:

```
shutdown-enable <option>
```

Activate <1> or deactivate <0> the warning when the primary power supply fails:

```
warning-enable <option>
```

Setting the WakeUp-Alarm:
(<1>: time, <2>: date, <3>: weekday)

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

Enable <1> or disable <0> the WakeUp-Alarm:

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

Set the time of the WakeUp-Alarm:

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

Set the weekday of the WakeUp-Alarm:

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

Set the date of the WakeUp alarm:

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

Set the time of the Poweroff-Alarm:

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

Enable <1> or disable <0> the Poweroff-Alarm:

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

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

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

Set the battery voltage, at which the Raspberry Pi should be switched off:

(<0>: deactivated, <1>: < 10 %, <2>: < 25 %, <3>: < 50 %)

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

Enable <1> or disable <0> the PowerOff Interval-Alarm:

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

Set the **On time** of the Interval-Alarm:

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

Set the **Off time** of the Interval-Alarm:

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

Enable <1> or disable <0> the PowerOn-Button:

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

Set the time of the initialisation of the PowerOn-Button:

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

13. EU DECLARATION OF CONFORMITY

Hereby, SIMAC Electronics GmbH declares that the product RB-StromPi3 is in compliance with the Directives EMV-Richtlinie 2014/30/EU (EMV) & 2011/65/EU (Rohs).
The norms listed below were applied to assess the device:

EN 55022:2010+ AC:2011
EN55024:2010
61000-3-2:2014
EN 61000-3-3:2013

Date	Name of the person in charge	Signature	Operational status
24.08.2018	Xu Lu		Managing director

14. FURTHER INFORMATION

Our information and redemption obligation according to the the Electrical and Electronic Equipment Act (ElektroG)

Symbol on electrial and electronic products :

This crossed-out bin means that electrical and electronic products do **not** belong into the household waste. You must hand over your old appliance to a registration office. Before you can hand over the old appliance, you must remove used batteries and accumulators which are not enclosed by the device.

Return options :
As the end user, you can hand over with the purchase of a new device your old appliance (which has essentially the same functions as the new one purchased from us) free of charge for disposal. Small devices which do not have outer dimensions greater than 25 cm can be submitted independently of the purchase of a new product in normal household quantities.

Possibility of restitution at our company location during our opening hours :
Simac GmbH, Pascalstr. 8, D-47506 Neukirchen-Vluyn

Possibility of restitution nearby :
We send you a parcel stamp with which you can send us your old appliance free of charge. For this possibility, you must contact us via e-mail at service@joy-it.net or via telephone.

Information about packaging:
Please package your old appliance safe during transport. Should you not have a suitable packaging material or you do not want to use your own material, you can contact us and we will send you an appropriate package.

15. SUPPORT

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

E-Mail: service@joy-it.net
Ticket-System: http://support.joy-it.net
Phone: +49 (0)2845 98469 – 66 (11- 17 o‘clock)

For further informations, please visit our website:
www.joy-it.net

Published: 19.03.2020