

# Quick Start Guide iSwitchPi

## Intelligent Power Switch for Raspberry Pi

English and German Version

## Inhaltsverzeichnis

1.	Before you start .....	1
2.	Overview Installation.....	1
3.	Installation Details .....	1

# Quick Start Guide to iSwitchPi (English)

## 1. Before you start

Checkout Layout of iSwitchPi-Board at end of this document.

Before installing the iSwitchPi board you have to define a GPIO pin to be used for communication. This is called the **GPIO communication pin**. The design of the board allows you to choose between the following GPIO pins: 13,19,20 or 26 (corresponding to physical pins 33,35,37 and 38). If you already use these pins in your application you need to redesign it to make one of these pins available. If possible, use GPIO pin 20 since this is the default value.

Furthermore, if you plan to use the square wave output of the iSwitchPI board as an external interrupt line to the Pi you need to choose another GPIO pin for the interrupt signal. For this, the design of the board allows you to choose between these GPIO pins: 17,22, 23, or 27 (corresponding to physical pins 11,15,16 and 13). This is called the **GPIO interrupt pin**.

Take a note of both pin numbers, you will need them during installation.

Also locate the Python script **iswitchpi.py** that needs to be transferred to the Pi later on. You can find the script (and the full documentation) on [GitHub](#). Download the repository to your local machine. Locate the script in this folder ../sources/python.

## 2. Overview Installation

The Installation is done in several steps:

- Define the GPIO pins you intend to use for iSwitchPi.
- Install the Python script in the Pi and make sure it is running ok by checking the console messages during boot.
- Stack the board onto the Pi, set the dip-switches and use the board in **TestMode** - this allows you to actually test the **GPIO communication pin**.
- Disable **TestMode** and have fun using this clever device.

## 3. Installation Details

After defining the GPIO pins to be used, follow these steps to get your iSwitchPi board up and running.

### 3.1 Install the iSwitchPi Python script in the Pi

- Connect a power supply to the Micro USB connector on the Pi board. Make sure you run the Pi **without** the iSwitchpi board at this stage.
- Login to the Pi and create a folder in the home folder of user pi. Call this folder iswitchpi or any other name you can remember.

- Copy the Python script iswitchpi.py into this folder. Use your preferred method to do this.
- Edit the file /etc/rc.local and add a line to start this script at boot time (replace myfolder with your folder name)  
python /home/pi/myfolder/ishwitchpi.py &

If you want to specify a GPIO pin **other** than the default (GPIO 20) use commandline parms:

- python /home/pi/myfolder/ishwitchpi.py [-d N] [-p nn] &

*The commandline parameter -d specifies debug output (default is 1):*

- 0: no output (quiet)
- 1: output at start and when a shutdown comand is issued (HALT oder REBOOT) (default)
- 2: full debug output for testing.

*The commandline parameter -p specifies the **GPIO communication pin**:*

- no parameter -p: GPIO 20 is used by default (physical pin 38).
- choose between - p 13 or - p 19 or -p 26 or - p 20
- Reboot the Pi and connect a display to the HDMI output. Check the boot messages and make sure the messages from script iswitchpi are visible. Check the output and make sure the script reports the correct GPIO pin it will use. This confirms that the script is running properly. Later, you can check if the board receives the pulses from the Pi (use the boards **TestMode**).
- Shutdown the Pi with sudo halt.
- The Pi is now ready for the board.

## 3.2 Configure the board

- Stack the board onto the Pi but **do not connect a power supply just yet.**

*Configure the board - communication Line*

- The **GPIO communication pin** you defined previously (either 13,19, 20 or 26) needs to be set on **Dip-Switch 1 right part. This must be the same pin the Python script uses.**
  - put ONLY ONE switch into position ON - thereby selecting GPIO 13,19,20 or 26
  - Double check the pin number with what you selected on the commandline for the Python script. Make sure it is the same pin ! The script uses GPIO pin 20 if no commandline parm is given.

*Configure the interrupt line (pulses)*

- If you want to interrupt the Pi using the square wave output of the board you need to do the following:
  - Locate **Dip-Switch 2** on the board. Select the desired frequency by setting the 4 switches as indicated on the component diagramm. If you do **NOT** need the interrupt line put all 4 switches in the OFF position - that disables the generation of the square wave.
  - Locate **Dip-Switch 1 left part** on the board. This selects the **GPIO interrupt pin**. Put ONLY one switch into the ON position - thereby selecting GPIO 17,22,23 or 27
  - Double check the pin number with what you use in your **own interrupt handling script**. Make sure it is the same pin! A working example of an interrputt-handling script can be found in the example

folder.

- If you do **NOT** need the interrupt line put all 4 switches of **Dip-Switch 2** and **Dip-Switch 1 left part** into the OFF position.

#### *External led and/or pushbutton*

- If needed, connect an external led and/or external pushbutton to the 4x2 pin-header as indicated on the component diagramm. No resistor is needed for the led. You might want to do this if the Pi is in a box.

### 3.3 Check if the board receives pulses from Pi

This step helps you to find out if the pulses from the Pi are received by the board.

Put the board into the TestMode - set **Dip Switch 3 position 1** to ON

- Connect a 5 Volt power supply to one of three possible inputs on the iSwitchPi board:
  - Barrel Jack 5.5/2.1 mm - make sure that the polarity is ok: center pin is +5Volt. **There is currently no protection against reverse polarity. So double check** the polarity. The 5 Volt + is on the center pin of the barrel jack.
  - Micro-USB connector
  - 4x2 pin-header as indicated on the component diagramm. Again: check the polarity.

Pressing the pushbutton once powers the pi, green led is on and the orange led blinks just once. Wait until the Pi has booted and the Python scripts start sending pulses. Watch the orange led - it should blink every 5 seconds or so. This tells you: ok, pulses from Pi are coming. If orange led does not blink check the GPIO communication pin used by the script and also check Dip switch 1 right part.

Another press on the pushbutton cuts power to the Pi. Clear the TestMode by setting **Dip Switch 3 position 1 to OFF**.

### 3.4 Using the board

If everything is ok you should see the green led flash every second for 50 ms. That indicates the **stand-by mode** of the iSwitchPi. Your lucky: you are ready to start the Pi.

A short press on the pushbutton should result in this:

- The Pi will receive its 5 Volt power and will start to boot.
- The green led will blink fast. This indicates: ok, I have given power to the Pi and I am now waiting for the Python script to send pulses on the chosen communication line telling me that the Pi is running.
- If 3 pulses are received from the Pi the led will turn steady on. This indicates **normal operating mode**. If no pulses are received from the Pi (script not running or wrong **GPIO communication pin** configured) power to the Pi will be cut after 30 seconds.

#### *Shutting down the Pi*

- A short press on the pushbutton during **normal operating mode** will result in this:
- The green led starts to blink slow and the 20 sec power off timer is started
- The Python script will be notified (using the GPIO communication line) and it will issue a sudo halt

command. The Pi will shut down within 10 sec (on a Pi model 2 or 3).

- After the timer on the board (within the MCU) expires 5 Volt power will shut of. The led will go back to short pulse blinking - the stand by mode.

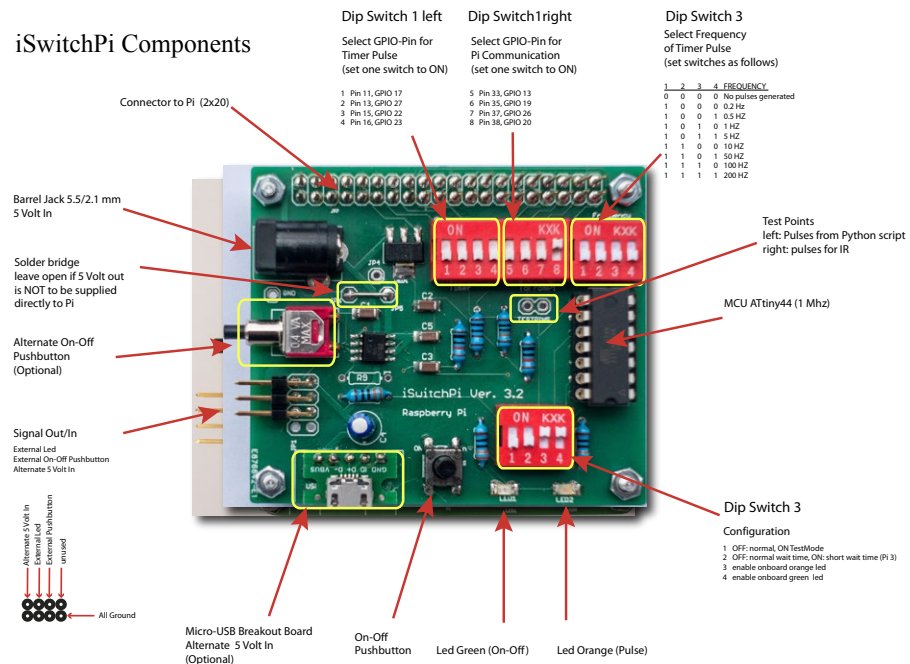
Study the state diagram to find out exactly what modes are possible.

**Happy computing with the clever iSwitchPi.**

# Quick Start Guide to iSwitchPi (German)

to be done.

# Layout iSwitchPi PCB



Peter K. Boxler, Dec. 2016

End of document.

Peter K. Boxler, December 2016