

Tutorial Funksteckdosen für den Repetierserver

Inhalt

1. Vorwort	1
2. Hardware.....	1
3. Anschlussplan	3
4. Software	3
5. Git installieren	3
6. Raspberry-Remote installieren	4
7. Anpassungen	4
8. Befehle für Repetier Server einrichten	4
9. Weitere Dateien	5
10. Nutzung der zusätzlichen Funktionen im Gcode.....	5
11. Wichtiger Hinweis!	6
12. Dateien	6

1. Vorwort

Zuerst möchte ich anmerken, das sich mich an die Anleitung von Philipp Seidel gehalten habe. Diese Anleitung ist für den Octoprint Printserver gedacht und funktioniert auf dem Repetier Server aber auch so, benötigt aber einige zusätzliche Einstellungen. <https://blog.seidel-philipp.de/anleitung-3d-drucker-mit-octoprint-einschalten-und-ausschalten> . Herzlichen Dank an Philipp Seidel für die wichtige Vorarbeit!

2. Hardware

Benötigt wird an Hardware der Raspberry Pi natürlich und Funksteckdosen die sich per DIP-Schalter konfigurieren lassen.



Abbildung 1 Codierschalter Funksteckdose

2.1. Codierung der Funksteckdose



Abbildung 2 Rückseite der Funksteckdose

Den SystemCode (hausinterner Code) stellen wir auf 10101 stellen
Unitcode stellen wir auf 10000 (A) – Der Unitcode eine eindeutige Nr. der Steckdose. Dies ist die Steckdose D oder 4

Hier das entsprechende 433 Mhz Sendemodul mit Zusatzantenne



Abbildung 3 433Mhz Sendemodul

so sieht es dann komplett aus.

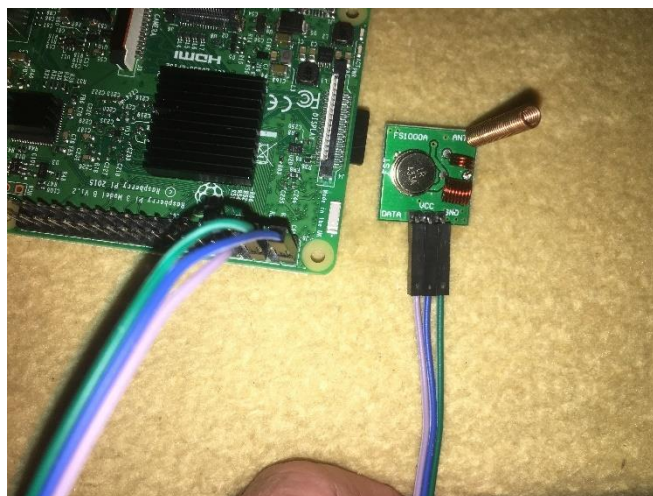


Abbildung 4 Raspberry mit angeschlossenem Sendemodul

3. Anschlussplan

Die Anschlüsse werden wie folgt verbunden

GND vom Sendemodul an -> GND des Raspberry

DATA vom Sendemodul an -> GPIO-PIN 17 des Raspberry

5V (VCC) vom Sendemodul an -> 5V des Raspberry

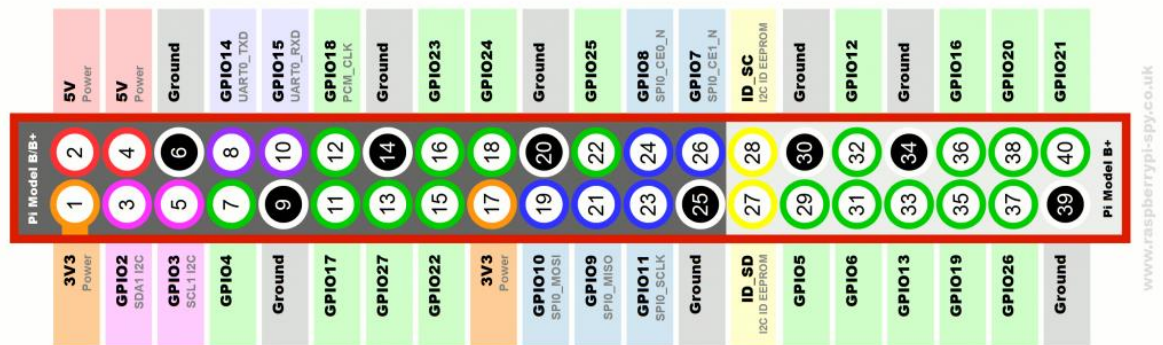


Abbildung 5 Pinbelegungsplan des Pi 3 Modell B

4. Software

Die benötigten Komponenten auf dem Raspberry werden wie folgt installiert. Software für die Installation ist Putty und WinSCP. Installation. Als Benutzer pi auf dem Raspberry, per Putty, anmelden.

5. Git installieren

WiringPi, mit dem die GPIO-Pins des Pi gesteuert werden

Da für die Installation noch einige elementare Module im Repetierserver Image fehlen, müssen diese vorher installiert werden und dies mit den folgenden Befehlen:

```
sudo apt-get update
sudo apt-get install gcc
sudo apt-get install g++
sudo apt-get install make
```

```
sudo apt-get update
sudo apt-get install gcc
sudo apt-get install make
asudo pt-get install git
sudo git clone git://git.drogon.net/wiringPi
cd wiringPi
sudo ./build
```

```
sudo apt-get update
sudo apt-get install git
sudo git clone git://git.drogon.net/wiringPi
cd wiringPi
sudo ./build
```

6. Raspberry-Remote installieren



```
cd
sudo git clone git://github.com/xkonni/raspberry-remote.git
cd raspberry-remote
sudo make send
```

```
cd
sudo git clone git://github.com/xkonni/raspberry-remote.git
cd raspberry-remote
sudo make send
```

7. Anpassungen

Nun kommen wir hier zu den benötigten Anpassungen, damit der Benutzer „repetierserver“ auch die Funktionen benutzen kann. Der User "repetierserver" muss als sogenannter "sudoer" deklariert werden und Rechte auf verschiedene Verzeichnisse und den "send"-Befehl bekommen. Das ist einiges an manueller Tipperei. Hier die Befehle und bitte darauf achten, das beim ersten "echo" nur ein ">"-Zeichen verwendet wird und ab dem zweiten "echo" dann bitte so ">>" : sudo -s echo "repetierserver ALL=NOPASSWD: /home/pi/raspberry-remote" > /etc/sudoers.d/repetierserver-remote echo "repetierserver ALL=NOPASSWD: /home/pi/raspberry-remote/send" >> /etc/sudoers.d/repetierserver-remote echo "repetierserver ALL=NOPASSWD: /home/pi/autopoweroff.sh" >> /etc/sudoers.d/repetierserver-remote echo "repetierserver ALL=NOPASSWD: /home/pi/autopoweron.sh" >> /etc/sudoers.d/repetierserver-remote echo "repetierserver ALL=NOPASSWD: /home/pi" >> /etc/sudoers.d/repetierserver-remote

Im Nachhinein habe ich festgestellt, das mit sudo visudo -f /etc/sudoers.d/repetierserver-remote die Datei auch erstellt werden kann. Üblicherweise öffnet sich dazu der „nano“ Editor. Innerhalb des Editors gibt es einige rudimentäre Befehle einer Textverarbeitung. Die aktuelle Zeile kann mit Ctrl-K ausgeschnitten und mit Ctrl-U wieder eingefügt werden. Damit sind dann nur noch einige Angaben zu ändern. Und mit Ctrl-O wird gespeichert und anschließend mit Ctrl-X der Editor verlassen.

```
repetierserver ALL=NOPASSWD: /home/pi/raspberry-remote/send
repetierserver ALL=NOPASSWD: /home/pi/raspberry-remote
repetierserver ALL=NOPASSWD: /home/pi/autopoweron
repetierserver ALL=NOPASSWD: /home/pi/autopoweron.sh
repetierserver ALL=NOPASSWD: /home/pi/autopoweroff
repetierserver ALL=NOPASSWD: /home/pi/autopoweroff.sh
repetierserver ALL=NOPASSWD: /home/pi
repetierserver ALL=NOPASSWD: /usr/bin/python
```

8. Befehle für Repetier Server einrichten

Auch das Schalten aus dem Gcode funktioniert mit selbstdefinierten Befehlen. Wichtige Grundlage ist die Anpassung der extcommands.xml, die ich hier angehängt habe.

```
<config>
<!-- If you want to run external commands, enter them in this configuration with
full path. You will see them in the main menu. Copy this file into the
<storage>/database directory.
Security consideration: The reason there is no online editor for this is simply
security. The commands you enter here are executed with the privileges of the
daemon running the server. If you would allow online configuration, a hacker could
simply add any command he needs to hack your system. -->

<command>
  <name>Shutdown Server</name>
  <execute>sudo /sbin/shutdown -h now</execute>
  <confirm>Really shut down the server?</confirm>
  <!-- Define if command should show up in local printer interface, default true.-->
```

```

        <local>true</local>
        <!-- Define if command should show up in remote printer interface, default true.-->
        <remote>true</remote>
        <!-- Define if command should show up only for users with print permission, default true.-->
        <print-permission>true</print-permission>
        <!-- Define if command should show up only for users with add files permission, default false.-->
        <add-permission>true</add-permission>
        <!-- Define if command should show up only for users with del files permission, default false.-->
        <del-permission>true</del-permission>
        <!-- Define if command should show up only for users with configuration permission, default false.-->
        <config-permission>true</config-permission>
    </command>
    <command>
        <name>Reboot Server</name>
        <execute>sudo /sbin/shutdown -r now</execute>
        <confirm>Really reboot the server?</confirm>
    <!-- Define if command should show up in local printer interface, default true.-->
    <local>true</local>
    <!-- Define if command should show up in remote printer interface, default true.-->
    <remote>true</remote>
    <!-- Define if command should show up only for users with print permission, default true.-->
    <print-permission>true</print-permission>
    <!-- Define if command should show up only for users with add files permission, default false.-->
    <add-permission>true</add-permission>
    <!-- Define if command should show up only for users with del files permission, default false.-->
    <del-permission>true</del-permission>
    <!-- Define if command should show up only for users with configuration permission, default false.-->
    <config-permission>true</config-permission>
    </command>
    <command>
        <name>Drucker_Strom_an</name>
        <execute>./home/pi/autopoweron.sh</execute>
    </command>
    <command>
        <name>Drucker_Strom_aus</name>
        <execute>./home/pi/autopoweroff.sh</execute>
    </command>
    <command>
        <name>Testmail</name>
        <execute>python /home/pi/Druckerfertig.py</execute>
    </command>
    <execute name="poweron" allowParams="true">./home/pi/autopoweron.sh</execute>
    <execute name="poweroff" allowParams="true">./home/pi/autopoweroff.sh</execute>
</config>

```

9. Weitere Dateien

Im Verzeichnis /home/pi müssen die Skriptdateien autopoweron.sh und autopoweroff.sh stehen. Diese müssen mit dem Befehl "chmod +X autopoweron.sh" bzw. "chmod +X autopoweroff.sh" ausführbar gemacht werden.

Die Datei autopoweron.sh hat folgenden Inhalt:

```
#!/bin/bash
sudo ./home/pi/raspberry-remote/send 10101 4 1
```

Die Datei autopoweroff.sh hat folgenden Inhalt:

```
#!/bin/bash
sudo ./home/pi/raspberry-remote/send 10101 4 0
```

10. Nutzung der zusätzlichen Funktionen im Gcode

```
@execute poweron
@execute poweroff
```

Diese Befehle veranlassen den Repetierserver dann entsprechend die Funksteckdose an oder auszuschalten.

Um die Befehle automatisch direkt von einem Slicer im Gcode einzubauen, werden diese Befehle im Scriptteil von z. B. Simplyfy 3D, eingetragen. Das Anschalten „poweron“ muss am Anfang des Scripts eingefügt werden {PREPEND "@execute poweron \n"} und wird durch den Postprozessor Befehle {PREPEND "xxx"} gesteuert. {APPEND "@execute poweroff\n"} für das Ausschalten muss dann am Ende stehen und wird über den Befehle {APPEND "xxx \n"}. Mit der Zeichenkombination „\n“ wird eine Zeilenschaltung gemacht. Im folgenden Screenshot ist die Anwendung zu sehen.

11. Wichtiger Hinweis!

Damit das Abschalten des Druckers nicht zu schnell stattfindet und das Druckbett vom Extruder weggefahren ist, muss auf die vollständige Abarbeitung des Gcodes geachtet werden. Damit dies gewährleistet werden kann, ist z.B. nach dem Fahrbefehl ein M400 notwendig. Damit wird der Controller angewiesen mit der Abarbeitung des Gcodes zu warten, bis der vorige Befehl abgearbeitet ist. Die Stelle im Endscript ist gelb markiert.

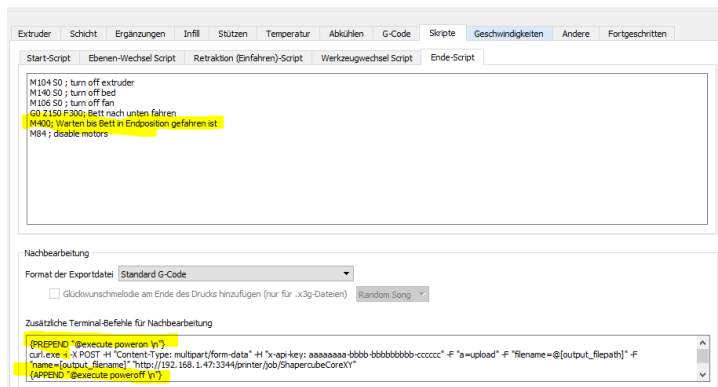


Abbildung 6 Einbinden der Befehle in Simplify 3D

12. Dateien



autopoweroff.sh



autopoweron.sh



extcommands.xml