

# PRJ001 Funksteckdosen

11. APRIL 2016

VERSION 1.00

Hausautomation

zum Nachrüsten:

Funksteckdosen auf 433MHz

+ erweiterbar

+ wartungsfreundlich

- unbefugter Zugriff möglich

(Das ist eine Eigenschaft

dieses Steckdosentyps und

nicht des hier beschriebenen

Projektes!)

TIP:

Geeignet für Leselampen,

oder die Weihnachtsbe-

leuchtung etc., solange das

unerwünschte Ein- bzw.

Ausschalten kein Risiko

darstellt.

...

## Inhaltsverzeichnis

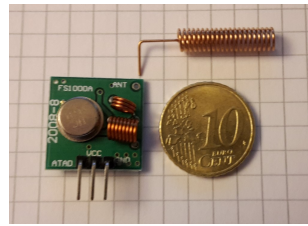
- 1 Controller: Raspberry PI GPIO
- 2 Aktor: Sender 433MHz
- 3 Anschluss: Breakout oder Lochrasterplatine
- 4 Software: Treiber und Kommandozeile
- 5 GUI: webbasiert für Intra- / Internet
- 6 Kompatibilität zu pilight
- 7 Vor- und Nachteile
- 8 Literaturverweise

## 1 Raspberry PI GPIO

(Steckverbinder für Model B+)

	3V Power	1	2	5V Power	
	GPIO2 SDA1 I2C	3	4	5V Power	
	GPIO3 SCL1 I2C	5	6	GND	
	GPIO4	7	8	GPIO14 UART0_TXD	
	GND	9	10	GPIO15 UART0_RXD	
	GPIO17	11	12	GPIO18 PCM_CLK	
	GPIO27	13	14	GND	
	GPIO22	15	16	GPIO23	
	3V Power	17	18	GPIO24	
	GPIO10	19	20	GND	
	GPIO09	21	22	GPIO25	
	GPIO11	23	24	GPIO08 SPI0_CEO_N	
	GND	25	26	GPIO07 SPI0_CE1_N	
	ID_SD I2C IDEEPROM	27	28	ID_SC I2C EEPROM	
	GPIO05	29	30	GND	
	GPIO06	31	32	GPIO12	
	GPIO13	33	34	GND	
	GPIO19	35	36	GPIO16	
	GPIO26	37	38	GPIO20	
	GND	39	40	GPIO21	

## 2. Aktor: Sender 433MHz



- Oben die Antenne  
ca. 16cm lang / Drahtstärke ca. 0,6mm
- Links der Sender FS1000A  
Anschlüsse Antenne / Data (spiegelverkehrt geschrieben) / VCC und GND

## 3. Anschluss ab Raspi

	3V Power	1	2	5V Power	VCC
	GPIO2 SDA1 I2C	3	4	5V Power	
	GPIO3 SCL1 I2C	5	6	GND	GND
	GPIO4	7	8	GPIO14 UART0_TXD	
	GND	9	10	GPIO15 UART0_RXD	
Data	GPIO17	11	12	GPIO18 PCM_CLK	
	...	.	.	...	

- Der Betrieb an 5V genügt für eine 2 Etagen, der Sender kann auch bis 12V betrieben werden – allerdings empfiehlt sich dann ein Netzteil oder eine abschaltbare Batterie / alternativ: täglicher Batteriewechsel..

## 4 Software: Treiber und Kommandozeile

- Hierzu existiert ausreichend Literatur (1) oder Anleitungen im www
- Konfiguration der Steckdosen nicht vergessen (syscode und unicode)
- Kommandozeile:  
Pfad z.B. /home/pi  
„./send \$syscode \$unicode 1“   Schalter An  
„./send \$syscode \$unicode 0“   Schalter Aus
- Als C-Code z.B. im CGI-Skript:  
Pfad z.B. /usr/lib/cgi-bin  
...  
System(„sudo ./send 11000 1 0“)

## 5 GUI: webbasiert für Intra- / Internet

Voraussetzung:

- Netzwerkanbindung
- Webserver: html und php, z.B. Apache

### GUI:

Webdesigner (Magix): erzeugt HTML-Code  
(als Erleichterung / Alternativ HTML und Tools wie Bootstrap)



Button, hier wird später ein PHP-Kommando hinterlegt

a) da HTML auf dem Browser läuft, die Kommandos jedoch auf dem Server (Raspi) Kommandos ausführen soll:

1) umbenennen: \*.htm in \*.php (Dateien und Links auf diese Dateien)  
z.B. index.htm nach index.php

2) Option für mobile Geräte den viewport umstellen (Darstellung Zoom)

von  
<meta name="viewport" content="width=device-width, initial-scale=1" />  
nach  
<meta name="viewport" content="width=480, initial-scale=1" />

b) Einbindung der Kommandozeilen

1) Im Header (am Ende):  
<?php  
include "ioctl.inc.php";  
?>

2) Austausch der Links durch PHP-Aufrufe

von  
<a href="Funk\_Couch\_an" onclick=...  
in  
<a href="<?php print(\$\_SERVER['PHP\_SELF']);?>?led=SD1ein" onclick=...

3) Die Datei „ioctl.inc.php“ mit Funkkommandoaufrufen

```
<?php
    $syscode_1 = "10000";
    ...
    $unicode_A = "1";
    ...
```

```

if(isset($_GET['led'])) {

    //LED ein
    if($_GET['led'] == '1ein') {
        exec("gpio mode 4 out");
        exec("gpio write 4 1");
        //echo "lokal IO GPIO4: Ausgang setzten + einschalten";
    }

    //LED aus
    elseif($_GET['led'] == '1aus') {
        exec("gpio mode 4 out");
        exec("gpio write 4 0");
        //echo "lokal IO GPIO4: Ausgang setzten + ausschalten";
    }

    //SD1 an
    elseif($_GET['led'] == 'SD1ein') {
        $_val = trim(@shell_exec("./send $syscode_1 $unicode_A 1"));
        echo "sd1 an";
    }
    //SD1 aus
    elseif($_GET['led'] == 'SD1aus') {
        $_val = trim(@shell_exec("./send $syscode_1 $unicode_A 0"));
        echo "SD7 aus";
    }
}
?>

```

## 6 Kompatibilität zu pilight:

```
$_val = trim(@shell_exec("pilight-send -p mumbi -p -s 3 -u 4 -t
```

Hier ist wird statt „0“ und „1“ „t“ und „f“ = true und false gesetzt  
Das verwendete Telegramm geht auf den Hersteller zurück (hier „mumbi“)  
Syscode und Unicode werden in Abkürzungen verwendet

Wählt man hier ein kompatible Kommandzeile, so läßt sich ein eigenes  
Web-GUI mit den Treibern und der GUI-App von pilight kombinieren.

## 7 Vor- und Nachteile

- Die Sender sind „oneway“; ob ein abgesetztes Kommando auch tatsächlich ausgeführt wird, wird nicht überprüft (daher wurde auf eine Status-Info wie bei Slider verzichtet)
- Funksteckdosen sind nicht vor Fernzugriffen geschützt, hier sind ein wenig komplexere Protokolle mit Autorisierungsmechanismen erforderlich; also nur eine Frage der Zeit...
- + Einfach zu verdrahten / einfaches Protokoll
- + Handsender können ebenfalls weiter / parallel verwendet werden
- + mit einem WebGUI erspart man sich eine App-Installation und ist damit flexibler

## 8 Literaturverweise

---

*Kofler, Kühnast, Scherbeck*  
*Raspberry Pi – Das umfassende Handbuch*  
*Galileo Computing / Rheinwerkverlag*

---

*Pilight, ein open Source Projekt zur Hausautomation*  
<https://www.pilight.org/>

---