



DL3NDD 2015

## A/D Wandler PCF8591 Raspberry pi

Analogue Werte messen (z.B.Spannung) incl. Temperatur-und Lichtsensor

Aufgabe

Übersicht

Volt messen  
mcp3008

Skripte

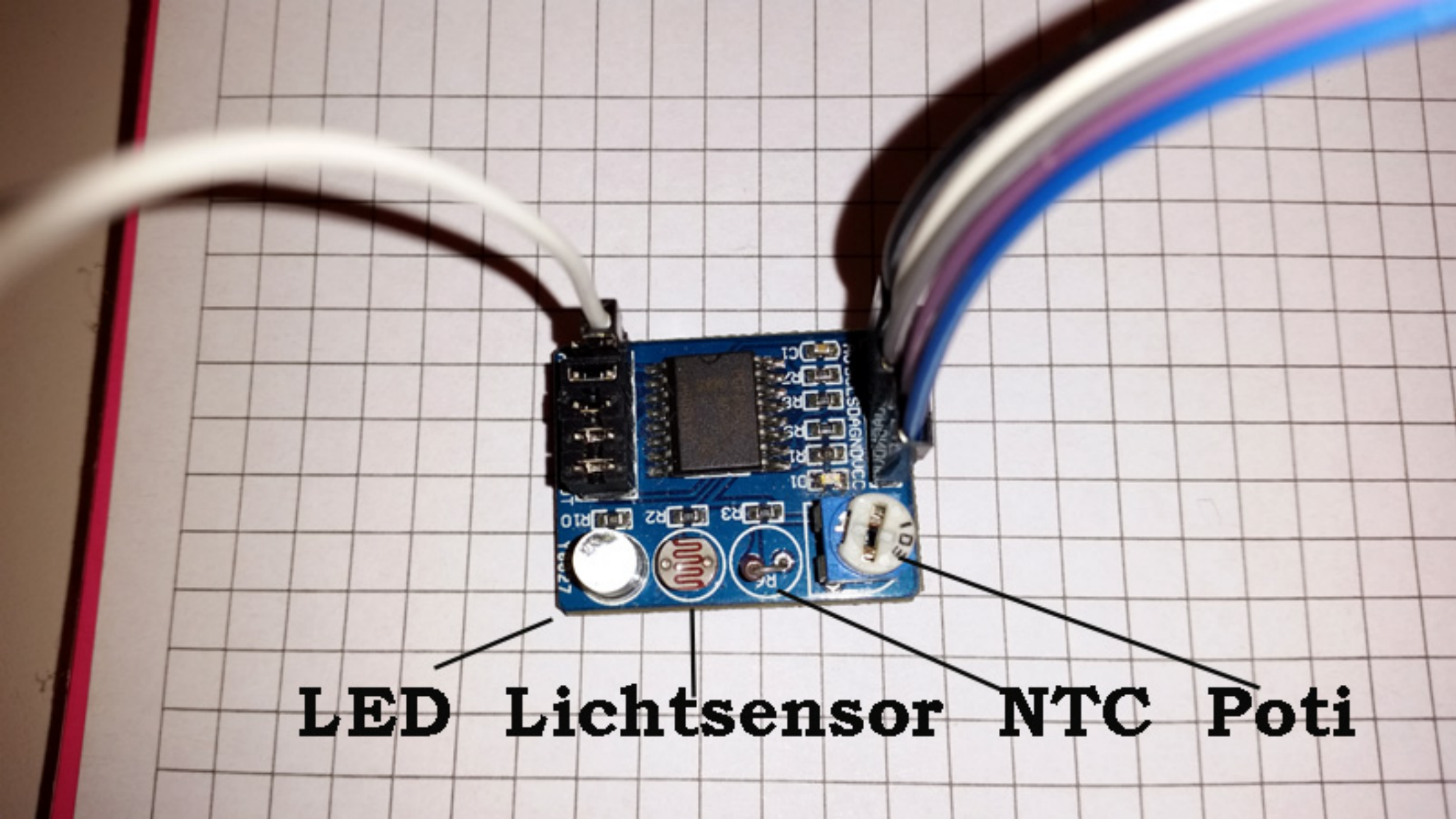
HF-Messen

Lüfter  
Steuerung

Links

Hardware:

PCF8591 Modul, bei Ebay 3 Euro incl. Versand! Keine Lötarbeiten



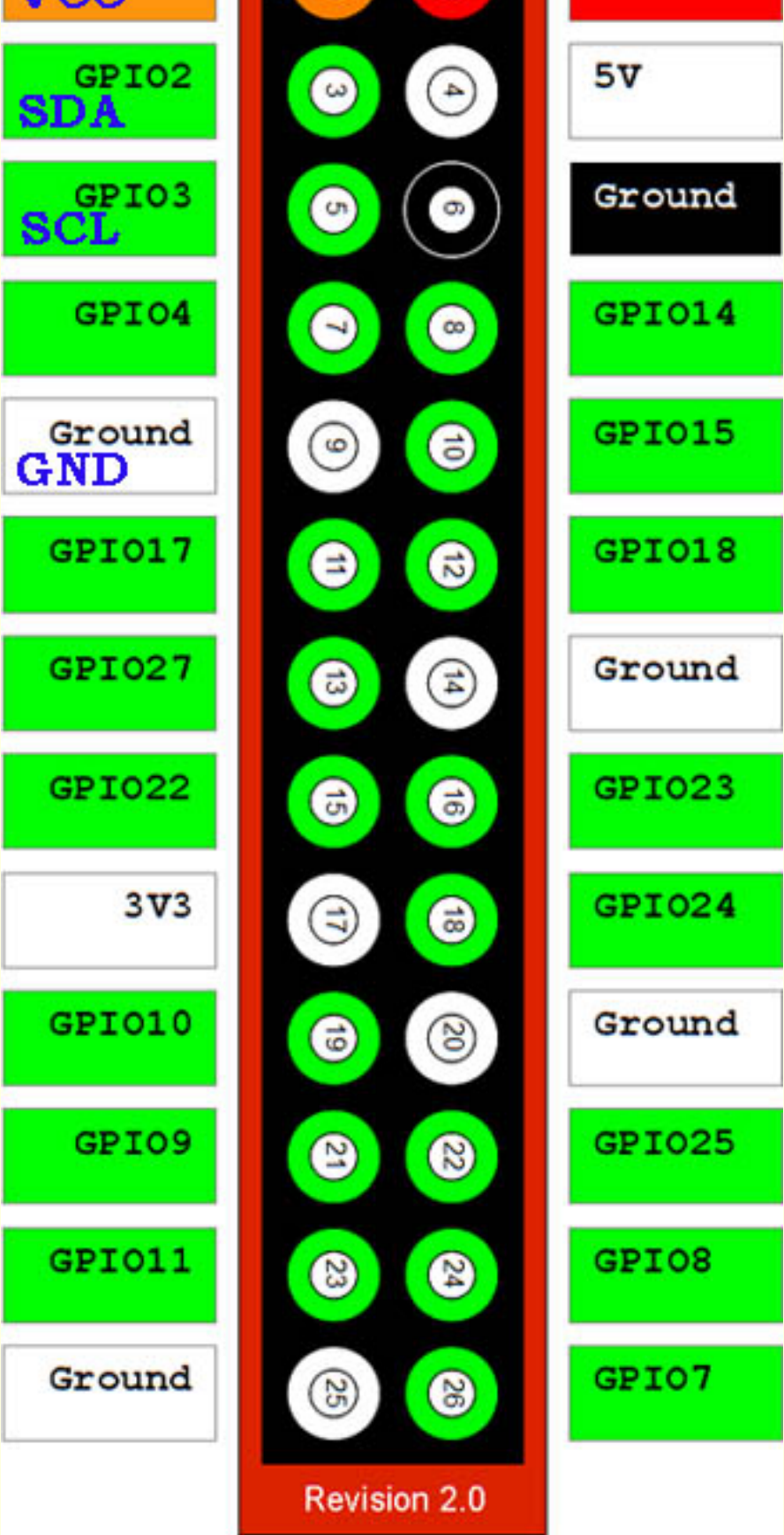
Das Modul hat vier Eingänge (AN0, AN1, AN2, AN3) und einen digitale Ausgang (out). Die Jumper verbinden die Eingänge mit den Sensoren auf dem Board.

Anschluss:

Der PCF8591 Modul arbeitet mit dem i2c-Bus. **A0 wird mit Plus der externen zu messenden Spannung verbunden.** [Achtung! Diese darf nicht 3.3Volt überschreiten. Wer höhere Spannungen messen will, z.B. an einem Solarakku o.ä. muss mit einem Spannungsteiler arbeiten. \[s.hier\]\(#\)](#)



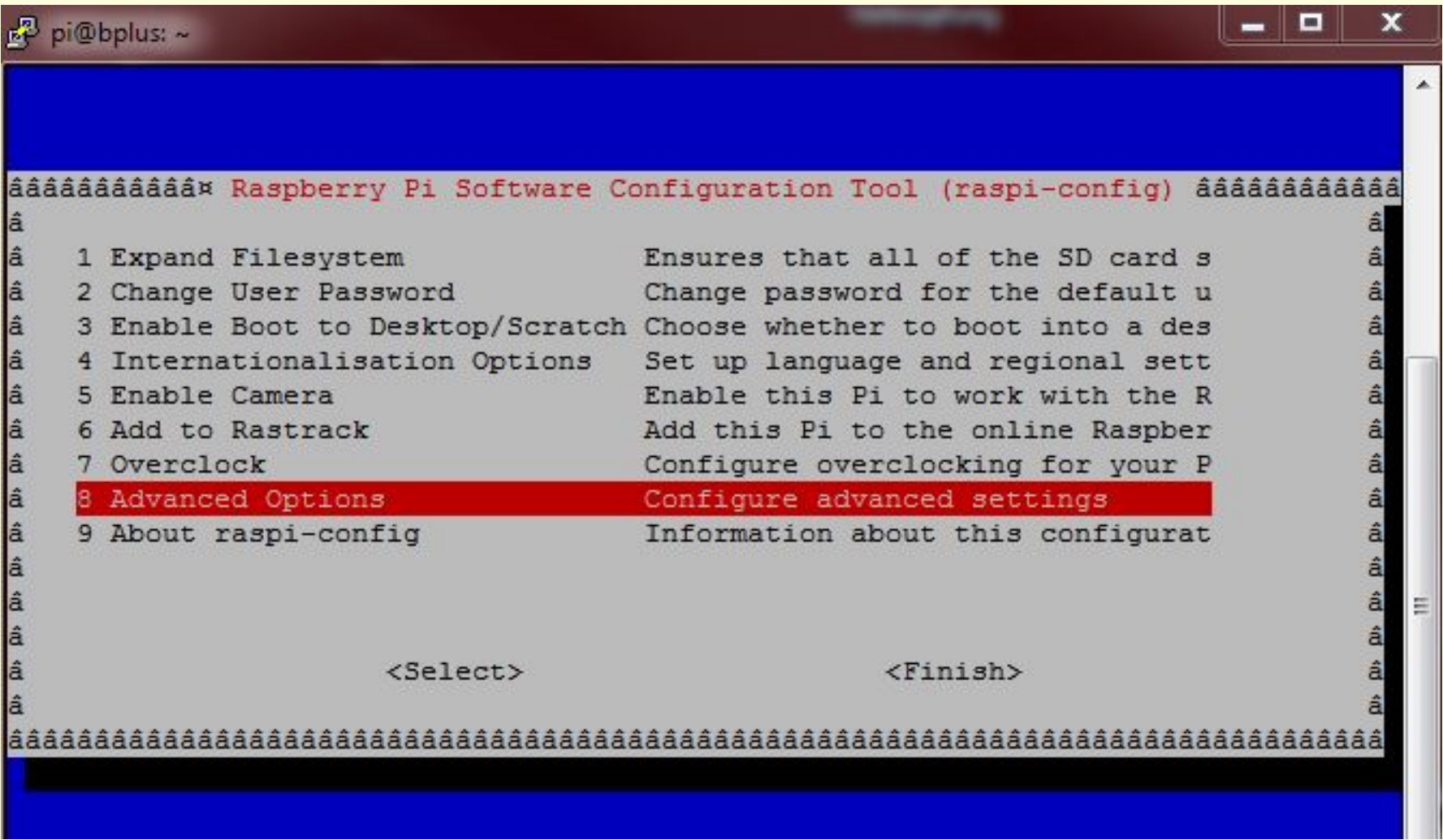




i2c einrichten:

Hier wird es etwas tricky. Dazu gibt es im Netz jede Menge Info. Ich fasse es mal zusammen wie es bei mir funktionierte.

Ab Kernel 3.18 findest du unter `sudo raspi-config` Advanve-i2C aktivieren



dann

```
sudo apt-get install python-smbus
```

und

```
sudo apt-get install i2c-tools
```

```
sudo reboot
```

Dann überprüfen wir und fügen evtl.hinzu

```
sudo nano /etc/modules
```

```
pi@bplus: ~
GNU nano 2.2.6      Datei: /etc/modules

/etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.

snd-bcm2835
i2c-dev

[ 8 Zeilen gelesen ]
^G Hilfe      ^O Speichern ^R Datei Öffnen ^Y Seite zur Seite ^K Ausschneiden ^C Cursor
^X Beenden    ^U Ausrichten ^W Wo ist      ^V Seite vor      ^I Ausschn. rücken Rechtschr.
```

Wichtig: bei meinem neuen B+ funktionierte es erstmal **nicht**, nach langem suchen die Lösung im Netz gefunden.

in die config.txt muss noch hinzugefügt werden:

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

```
# Additional overlays and parameters are documented /boot/overlays/README
dtparam=i2c1=on
dtparam=i2c_arm=on
```

Jetzt sollte alles gehen :-), test mit:

```
sudo i2cdetect -y 1
```

das Ergebnis sieht dann so aus:

```
pi@bplus ~ $ sudo i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40:  -- -- -- -- -- -- -- -- 48 -- -- -- -- -- --
50:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
pi@bplus ~ $
```

Einfaches Python 2 Skript zum auslesen:

```
#!/usr/bin/python2
import smbus
import time

bus = smbus.SMBus(1)
add = 0x48

ch0=0x00
ch1=0x01
ch2=0x02
ch3=0x03

def readAD():
    analog = bus.read_byte(add)
    return analog

def writeDA(value):
    bus.write_byte_data(add, 0x44, value)

while(0==0):
    an0 = readAD()
    an1 = readAD()
    an2 = readAD()
    an3 = readAD()
    print 'Licht = %3d Volt = %3d Poti = %3d Temp = %3d' % (an0, an1, an2, an3)
    writeDA(an0)
    time.sleep(0.5)
    writeDA(0x00)
```



Geht wunderbar...das Skript muss dann noch erweitert werden zum Sammeln und Darstellen der Werte, und evtl. Transport auf eine Webseite.

```
1  #!/usr/bin/python2
2  import smbus
3  import time
4
5  bus = smbus.SMBus(1)
6  add = 0x48
7
8  ch0=0x00
9  ch1=0x01
10 ch2=0x02
11 ch3=0x03
12
13 def readAD():
14     analog = bus.read_byte(add)
15     return analog
16
17 def writeDA(value):
18     bus.write_byte_data(add, 0x44, value)
19
20 while(0==0):
21     an0 = readAD()
22     an1 = readAD()
23     an2 = readAD()
24     an3 = readAD()
25     print 'Licht = %3d Volt = %3d Poti = %3d Temp = %3d' % (an0, an1, an2, an3)
26     writeDA(an0)
27     time.sleep(0.5)
28     writeDA(0x00)
29
30
```

```
pi@bplus: ~
Licht = 116 Volt = 120 Poti = 101 Temp = 211
Licht = 114 Volt = 120 Poti = 101 Temp = 211
Licht = 111 Volt = 120 Poti = 101 Temp = 211
Licht = 116 Volt = 120 Poti = 101 Temp = 211
Licht = 114 Volt = 120 Poti = 101 Temp = 211
Licht = 111 Volt = 120 Poti = 101 Temp = 211
Licht = 116 Volt = 120 Poti = 101 Temp = 211
Licht = 113 Volt = 120 Poti = 101 Temp = 211
Licht = 111 Volt = 120 Poti = 101 Temp = 211
Licht = 117 Volt = 120 Poti = 101 Temp = 211
Licht = 112 Volt = 120 Poti = 101 Temp = 211
Licht = 112 Volt = 120 Poti = 101 Temp = 211
Licht = 117 Volt = 120 Poti = 101 Temp = 211
Licht = 112 Volt = 120 Poti = 101 Temp = 211
Licht = 112 Volt = 120 Poti = 101 Temp = 211
Licht = 117 Volt = 120 Poti = 101 Temp = 211
Licht = 112 Volt = 120 Poti = 101 Temp = 211
Licht = 116 Volt = 120 Poti = 101 Temp = 211
Licht = 112 Volt = 120 Poti = 101 Temp = 211
Licht = 113 Volt = 120 Poti = 101 Temp = 211
Licht = 115 Volt = 120 Poti = 101 Temp = 211
Licht = 112 Volt = 120 Poti = 101 Temp = 211
Licht = 115 Volt = 120 Poti = 101 Temp = 211
Licht = 115 Volt = 120 Poti = 101 Temp = 211
Licht = 112 Volt = 120 Poti = 101 Temp = 211
Licht = 115 Volt = 120 Poti = 101 Temp = 211
Licht = 115 Volt = 120 Poti = 101 Temp = 211
Licht = 112 Volt = 120 Poti = 101 Temp = 211
Licht = 117 Volt = 120 Poti = 101 Temp = 211
Licht = 112 Volt = 120 Poti = 101 Temp = 211
Licht = 113 Volt = 120 Poti = 101 Temp = 211
^CTraceback (most recent call last):
  File "/home/pi/simplepcf.py", line 27, in <module>
    time.sleep(0.5)
KeyboardInterrupt
pi@bplus ~ $
```