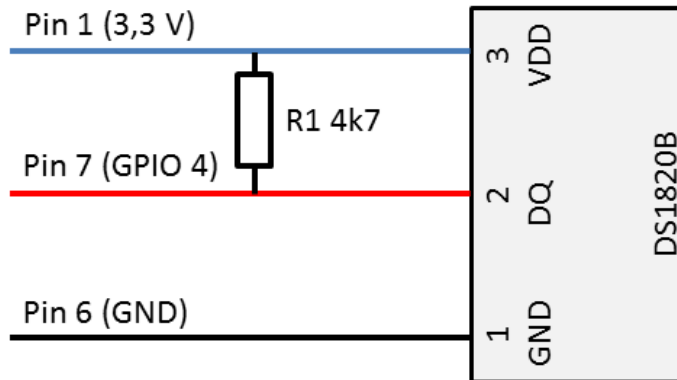


## 4.1 Temperaturerfassung mit 1-Wire auf RASPi ohne zusätzlichen 1-Wire Adapter

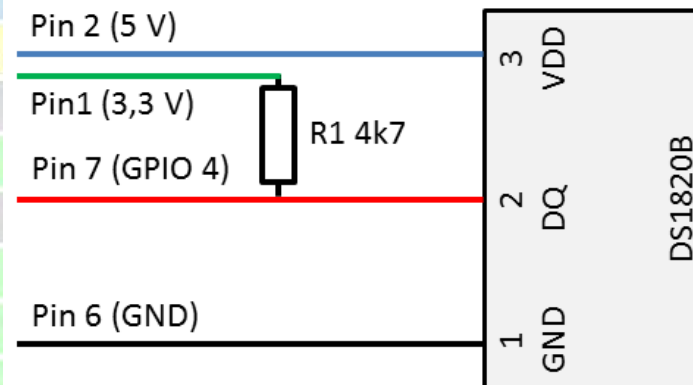


Pi Model B/B+			
3V3 Power	1	2	5V Power
GPIO2 SDA1 I2C	3	4	5V Power
GPIO3 SCL1 I2C	5	6	Ground
GPIO4	7	8	GPIO14 UART0_TXD
Ground	9	10	GPIO15 UART0_RXD
GPIO17	11	12	GPIO18 PCM_CLK
GPIO27	13	14	Ground
GPIO22	15	16	GPIO23
3V3 Power	17	18	GPIO24
GPIO10 SPI0_MOSI	19	20	Ground
GPIO9 SPI0_MISO	21	22	GPIO25
GPIO11 SPI0_SCLK	23	24	GPIO8 SPI0_CE0_N
Ground	25	26	GPIO7 SPI0_CE1_N
ID_SD I2C ID EEPROM	27	28	ID_SC I2C ID EEPROM
GPIO5	29	30	Ground
GPIO6	31	32	GPIO12
GPIO13	33	34	Ground
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
Ground	39	40	GPIO21

Für **kurze** Leitungen  
(parasitäre Versorgung)



Für **längere** Leitungen



## 4.2 Temperaturerfassung mit 1-Wire auf RASPi ohne zusätzlichen 1-Wire Adapter

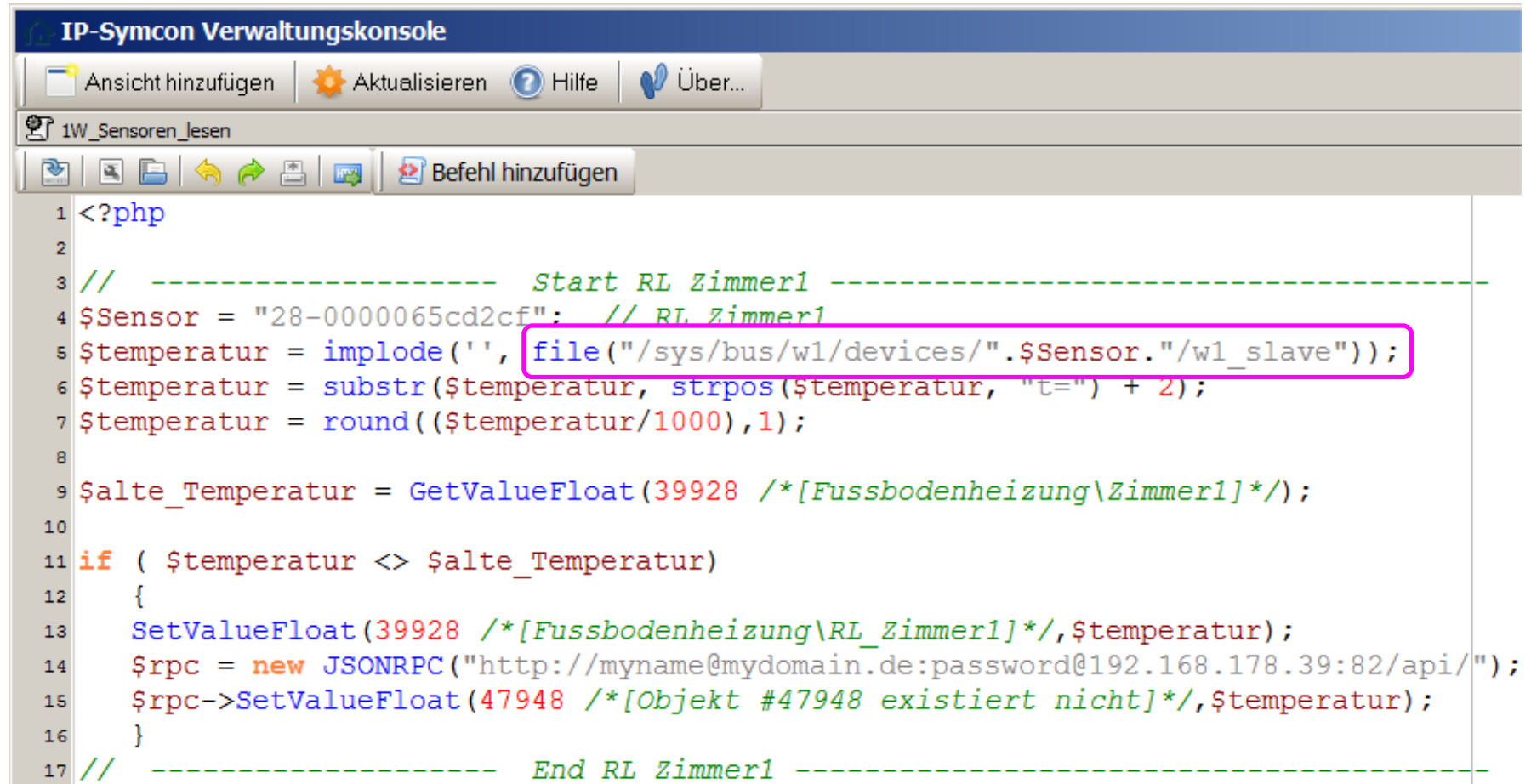
### Notwendige Kernel Module

```
sudo nano /etc/modules  
w1-gpio pullup=1  
w1-therm
```

### 1-Wire Filesystem auf dem RASP

```
cd /sys/bus/w1/devices  
  
ls -l  
28-0000065bff68 28-0000065c1c55 ... ..  
  
cd 28-0000065bff68  
  
cat w1_slave  
b8 01 4b 46 7f ff 08 10 8a : crc=8a YES  
b8 01 4b 46 7f ff 08 10 8a t=27500
```

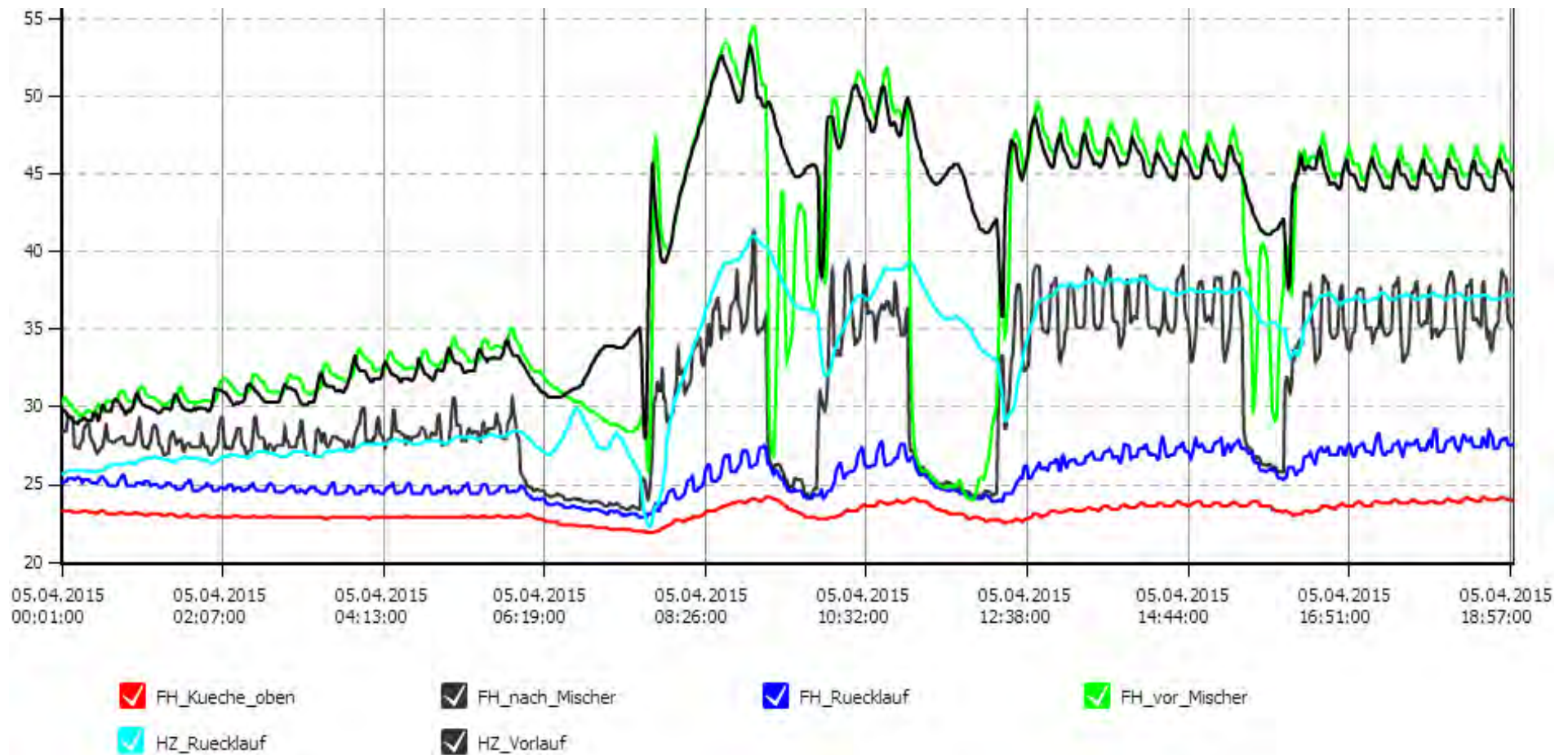
## 4.3 Temperaturerfassung mit 1-Wire auf RASPi ohne zusätzlichen 1-Wire Adapter



The screenshot shows the IP-Symcon Verwaltungskonsole interface. The title bar reads "IP-Symcon Verwaltungskonsole". Below the title bar are buttons for "Ansicht hinzufügen", "Aktualisieren", "Hilfe", and "Über...". The main window title is "1W\_Sensoren\_lesen". Below the title bar are icons for file operations and a "Befehl hinzufügen" button. The main content area displays PHP code for reading a 1-Wire sensor. The code is as follows:

```
1 <?php
2
3 // ----- Start RL Zimmer1 -----
4 $Sensor = "28-0000065cd2cf"; // RL Zimmer1
5 $temperatur = implode('', file("/sys/bus/w1/devices/". $Sensor . "/w1_slave"));
6 $temperatur = substr($temperatur, strpos($temperatur, "t=") + 2);
7 $temperatur = round(($temperatur/1000), 1);
8
9 $alte_Temperatur = GetValueFloat(39928 /*[Fussbodenheizung\Zimmer1]*/);
10
11 if ( $temperatur <> $alte_Temperatur)
12 {
13     SetValueFloat(39928 /*[Fussbodenheizung\RL_Zimmer1]*/, $temperatur);
14     $rpc = new JSONRPC("http://myname@mydomain.de:password@192.168.178.39:82/api/");
15     $rpc->SetValueFloat(47948 /*[Objekt #47948 existiert nicht]*/, $temperatur);
16 }
17 // ----- End RL Zimmer1 -----
```

## 4.4 Temperaturerfassung mit 1-Wire auf RASPi ohne zusätzlichen 1-Wire Adapter





## 5.2 Ölstandsmessung mit RASPi auf Ultraschallbasis (leider noch experimentell)

**HC-SR04**



**DYP-ME007Y**



## 5.4 Ölstandsmessung mit RASPi auf Ultraschallbasis (leider noch experimentell)

```
#GPIO Pins zuweisen
```

```
GPIO_TRIGGER = 23
```

```
GPIO_ECHO = 24
```

```
#Richtung der GPIO-Pins festlegen (IN / OUT)
```

```
GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
```

```
GPIO.setup(GPIO_ECHO, GPIO.IN)
```

```
def distanz():
```

```
    # setze Trigger auf HIGH
```

```
    GPIO.output(GPIO_TRIGGER, True)
```

```
    # setze Trigger nach 0.01ms aus LOW
```

```
    time.sleep(0.00001)
```

```
    GPIO.output(GPIO_TRIGGER, False)
```

```
    ....>
```

```
    StartZeit = time.time()
```

```
    StopZeit = time.time()
```

```
    # speichere Startzeit
```

```
    while GPIO.input(GPIO_ECHO) == 0:
```

```
        StartZeit = time.time()
```

```
    # speichere Ankunftszeit
```

```
    while GPIO.input(GPIO_ECHO) == 1:
```

```
        StopZeit = time.time()
```

```
    # Zeit Differenz zwischen Start und Ankunft
```

```
    TimeElapsed = StopZeit - StartZeit
```

```
    # mal der Schallgeschwindigkeit (34300 cm/s)
```

```
    # und durch 2 teilen, da hin und zurueck
```

```
    distanz = (TimeElapsed * 34300) / 2
```

```
    return distanz
```



## 5.4 Ölstandsmessung mit RASPi auf Ultraschallbasis (leider noch experimentell)

Anzahl

