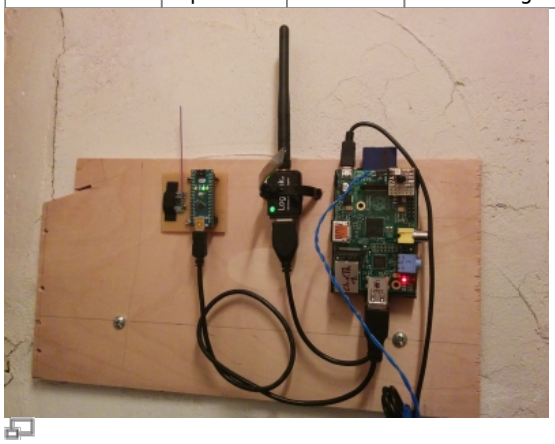
FHEM zu Hause

Erfassung verschiedener Raumtemperaturen und der jeweiligen Luftfeuchtigkeit mittels Funk Thermo-/Hygrometern, einem Raspberry Pi B und einem selbstgebauten 868 MHz Transceiver sowie dem Gasverbrauch mittels Reedkontakt direkt am Gaszähler.

Sensoren

Folgende Funk Temperatur- und Luftfeuchtesensoren habe ich erfolgreich getestet:

Typ / Bezeichnung	Datenrate	mehrfach nutzbar	Funktion(en)
30.3187.IT	17.241 kbps	nein, da identische ID für alle Geräte	Temperatur & Luftfeuchtigkeit
TX-38 IT	9.579 kbps	ja	Temperatur
TX29DTH-IT	17.241 kbps	ja	Temperatur & Luftfeuchtigkeit



Fertiger Aufbau neben der Heizung

Informationsquellen

1. http://www.fhemwiki.de/wiki/Raspberry_Pi
2. http://www.fhemwiki.de/wiki/DbLog
3. https://blog.moneybag.de/fhem-flashen-von-jeelinks-unter-fhem/
4. https://blog.moneybag.de/lacrosse-temperatursensor-an-arduino-nano-und-rfm12b-als-jeelink-ersatz/
5. https://blog.moneybag.de/fhem-guenstige-temperatur-und-luftfeuchte-sensoren-von-lacrosse/ - mehrere Sensoren gleichzeitig
6. http://www.meintechblog.de/2016/05/fhem-server-auf-dem-raspberry-pi-in-weniger-als-einer-stunde-einrichten/
7. http://www.meintechblog.de/2015/01/fhem-mit-jeelink-luftfeuchte-und-temperatur-zum-low-cost

-tarif-messen/ - unten verwendet! ,siehe auch Kommentare
8. https://steigerbalett.wordpress.com/2015/05/24/lacrosse-proficelltfa-dostmanntechnoline-temperatursensoren-an-fhem-anlernen/ - sehr wichtig!
9. http://blog.bubux.de/gaszaehler-auslesen/
10. https://forum.fhem.de/index.php?topic=20506.0
11. http://voizchat.de/gaszaehler-verbrauch-erfassen-mit-fhem-und-raspberry-gpio/
12. http://www.elinux.org/RPi_Low-level_peripherals

Implementierung

Den Raspberry Pi vorbereiten

- Raspbian Lite herunterladen
- Entpacken
- auf eine SD-Karte schreiben



Der von mir verwendete Raspberry Pi B (1. Generation) hat eine Polyfuse direkt hinter der Micro-USB Buchse. Diese begrenzt den Strom auf unter 1A so dass stärkere Netzteile wirkungslos bleiben. Diese Strombegrenzung führt unter bestimmten Bedingungen zum Einbruch der Versorgungsspannung (5V) und in der Folge zu Fehlfunktionen der Peripherie wie z.B. des EdiMax WLAN-Sticks. Ausführlich diskutiert hier.
Lösung: Polyfuse entfernen und einen 0 Ohm Widerstand einbauen!

```
wget https://downloads.raspberrypi.org/raspbian_lite_latest
unzip 2016-05-27-raspbian-jessie-lite.zip
dd bs=4M if=2016-05-27-raspbian-jessie-lite.img of=/dev/sdc
# oder für Ubuntu 16.04 LTS
dd bs=4M if=2016-05-27-raspbian-jessie-lite.img of=/dev/mmcblk0
```

- Raspberry Pi mit der neuen SD-Karte booten
- eine serielle Verbindung per FTDI-Kabel herstellen (SSH ist standardmäßig ab 11/2016 in Raspbian Jessie deaktiviert)
- SSH Server mittels `raspi-config` aktivieren damit die weitere Konfiguration per SSH erfolgen kann
- Systemtools installieren
- WLAN einrichten

Um eine serielle Verbindung zum Raspberry Pi (Modell B) herstellen zu können benötigt man z.B. ein sog. FTDI Kabel für einen Logik-Pegel von 3,3V. Ein original FTDI Kabel wird dann wie folgt verbunden:

Pin Nummer	Pin Name Rev1	Pin Name Rev2	Hardware Hinweis	Alt 0 Funktion	andere alternative Funktion	FTDI Kabel Farbe
P1-02	5V0		Supply through input poly fuse			

Pin Nummer	Pin Name Rev1	Pin Name Rev2	Hardware Hinweis	Alt 0 Funktion	andere alternative Funktion	FTDI Kabel Farbe
P1-04	5V0		Supply through input poly fuse			
P1-06	GND					Schwarz
P1-08	GPIO 14		Boot to Alt 0 →	UART0_TXD	ALT5 = UART1_TXD	Gelb
P1-10	GPIO 15		Boot to Alt 0 →	UART0_RXD	ALT5 = UART1_RXD	Orange

Nach dem erstmaligen booten der neu erstellten SD-Karte wird das Dateisystem automatisch auf den gesamten zur Verfügung stehenden Speicherplatz ausgedehnt. Dieser Vorgang kann etwas dauern. Anschließend kann man sich über die serielle Konsole anmelden und die erste Konfiguration vornehmen.

SSH Server einschalten

```
sudo minicom -b 115200 -o -D /dev/ttyUSB0
 --> User: Pi
 --> PW: Raspberry
sudo su
stty rows 48 cols 200
raspi-config
  --> enable SSH Server
apt-get update
apt-get -y upgrade
apt-get -y autoremove
shutdown -r && exit #reboot
```

wechsel auf SSH Verbindung

```
ssh -l pi <RASPBERRYPI-IP>
sudo su
# Tool installieren:
aptitude install mc apt-transport-https ifplugd ntpdate

# NTP konfigurieren:
sudo nano /etc/ntp.conf
 -> server 192.168.100.1

### Nameserver eintragen:
###echo "nameserver 192.168.100.1" >> /etc/resolv.conf

# DNS-Dienst systemd-resolved aktivieren und starten:
systemctl enable systemd-resolved.service
systemctl start systemd-resolved.service
```

```
# Prüfen, ob der Dienst laeuft:
systemctl status systemd-resolved

# Damit beim einstecken oder herausziehen des Netzwerkkables das WLAN
deaktiviert bzw. aktiviert wird muss ''Ifplugd'' konfiguriert wwerden:
nano /etc/ifplugd/action.d/ifupdown

#!/bin/sh
set -e
case "$2" in
up)
  /sbin/ifconfig wlan0 down
  /sbin/ifup $1
  ;;
down)
  /sbin/ifdown $1
  /sbin/ifconfig wlan0 up
  /sbin/route add default gw 192.168.100.1 wlan0
  ;;
esac

# ... und noch die grundlegenden Ifplugd Einstellungen:
nano /etc/default/ifplugd
INTERFACES="eth0"
HOTPLUG_INTERFACES="all"
ARGS="-q -f -u0 -d10 -w -I"
SUSPEND_ACTION="stop"

# Stromsparfunktion des WLAN Adapters deaktivieren
echo "options 8192cu rtw_power_mgnt=0 rtw_enusbss=0" >
/etc/modprobe.d/8192cu.conf

# Netzwerkschnittstellen (eth0 und wlan0) mit festen IP Adressen
konfigurieren:
sudo nano /etc/network/interfaces
...
auto lo
iface lo inet loopback

allow-hotplug eth0
iface eth0 inet static
  address 192.168.100.12
  netmask 255.255.255.0
  broadcast 192.168.100.255
  gateway 192.168.100.1
  dns-nameservers 192.168.100.1

auto wlan0
iface wlan0 inet static
  address 192.168.100.13
  netmask 255.255.255.0
```

```
  broadcast 192.168.100.255
  gateway 192.168.100.1
  dns-nameservers 192.168.100.1
  wpa-ap-scan 1
  wpa-scan-ssid 1
  wpa-ssid "WLAN-SSID"
  wpa-psk "WLAN-PASSWORT"

# Netzwerkschnisstellen vom DHCP CLient Dienst ausschließen:
echo "denyinterfaces eth0 wlan0" >> /etc/dhcpcd.conf

# DHCP Client Daemon deaktivieren & deinstallieren:
sudo systemctl disable dhcpcd
# oder
sudo service dhcpcd stop
apt-get purge isc-dhcp-client dhcpcd5 dhcpcd isc-dhcp-common

# Ifplugd als Systemdienst aktivieren:
sudo systemctl enable ifplugd

# abschließender Reboot
shutdown -h now && exit
```

FHEM Installieren

```
sudo su
apt-get -f install
apt-get -y install perl libdevice-serialport-perl libio-socket-ssl-perl
libwww-perl libxml-simple-perl libdigest-md5-file-perl libgd-graph-perl
libimage-librsvg-perl libjson-perl liblwp-protocol-http-socketunix-perl
liblwp-protocol-https-perl libnet-ssleay-perl libsoap-lite-perl
msttcorefonts python-dev python-rpi.gpio
apt-get install avrdude
wget http://fhem.de/fhem-5.7.deb
dpkg -i fhem-5.7.deb
apt-get -f install
```

Browserfenster öffnen: http://<IP_DES_R-PI>:8083/fhem

```
# Auf Updates prüfen:
update check

# Update durchführen:
update

# Passwortschutz abschalten:
attr global motd none

# Neustart nach update durchführen:
```

```
shutdown restart

# Schnittstelle für Arduino Nano mit Empfängermodul festlegen:
define myJeeLink JeeLink /dev/ttyUSB0@57600

# manuelles Programmieren, nur zur Info, hier nicht nötig ;-)
attr myJeeLink flashCommand avrdude -p atmega328P -c arduino -P [PORT] -b
57600 -D -U flash:w:[HEXFILE] 2>[LOGFILE]

# Baudrate festlegen, 1 = 17.241 kbps, 2 = 9.579 kbps, 4 = 8.842 kbps
# Beispiele:
# 3m -> wechselt zwischen 17.241 kbps und 9.579 kbps
# 5m -> wechselt zwischen 17.241 kbps und 8.842 kbps
# 7m -> wechselt zwischen allen drei data rates
# attr myJeeLink initCommands 1m 30t

# Da meine Sensoren alle mit der Datenrate 17.241bit/s funken benötige ich
keinen automatischen Wechsel, also nur 1m ohne xxt (xx = Wartezeit in
Sekunden zw. Wechsel)
attr myJeeLink initCommands 1m

# Firmware in Arduino Nano programmieren:
set myJeeLink flash LaCrosse

#set myJeeLink LaCrossePairForSec 60
#Weitere Einrichtung siehe hier:
http://www.meintechblog.de/2013/11/fhem-logfiles-und-graphen-datenlast-reduz
ieren-und-werte-ordentlich-visualisieren/
#define FileLog_Sensor1 FileLog ./log/Sensor1-%Y.log Sensor1.Temp:*
#attr Sensor1.Temp event-min-interval
state:600,battery:3600,temperature:300,humidity:300
#attr myJeeLink initCommands 3m 15t
```

Loggen in eine MySQL Datenbank

```
./etc/init.d/fhem stop
apt-get install mysql-server mysql-client libdbd-mysql libdbi1 libdbd-mysql-
perl libclass-dbi-mysql-perl

nano /opt/fhem/contrib/dblog/db_create_mysql.sql
###
CREATE DATABASE `fhem` DEFAULT CHARACTER SET utf8 COLLATE utf8_bin;
CREATE USER 'fhemuser'@'%' IDENTIFIED BY 'MYSQL-USER-PASSWORD';
CREATE TABLE `fhem`.`history` (TIMESTAMP TIMESTAMP, DEVICE varchar(64), TYPE
varchar(64), EVENT varchar(512), READING varchar(64), VALUE varchar(128),
UNIT varchar(64));
CREATE TABLE `fhem`.`current` (TIMESTAMP TIMESTAMP, DEVICE varchar(64), TYPE
varchar(64), EVENT varchar(512), READING varchar(64), VALUE varchar(128),
UNIT varchar(64));
GRANT SELECT, INSERT, DELETE, UPDATE ON `fhem`.* TO 'fhemuser'@'%';
```

```
CREATE INDEX Search_Idx ON `fhem`.`history` (DEVICE, READING, TIMESTAMP);
###
mysql -p -u root < /opt/fhem/contrib/dblog/db_create_mysql.sql
nano /opt/fhem/contrib/dblog/db.conf
###
## for MySQL
################################################################
%dbconfig= (
    connection => "mysql:database=fhem;host=localhost;port=3306",
    user => "fhemuser",
    password => "MYSQL-USER-PASSWORD",
);
#
./etc/init.d/fhem start
###
```

in FHEM:

```
define logdb DbLog ./contrib/dblog/db.conf .*:.*
```

neuen/zusätzlichen Sensor anlernen

```
set myJeeLink LaCrossePairForSec 60
rename LaCrosse_23 Sensor_<nn>.Temp_Humid
attr Sensor_<nn>.Temp_Humid event-min-interval
state:600,battery:3600,temperature:300,humidity:300
--> Filelog löschen
shutdown restart
kurz warten damit neue daten gesammelt werden können...
Klick auf "Everything" -> "logdb" -> "Create SVG plot from DbLog" -> neuen
Plot zusammenklicken ;-)
```

Batteriewechsel

```
set Sensor_03.Temp_Humid replaceBatteryForSec 60
# Batterien wechseln
# Aufzeichnung wird weiter geführt ...
```

Gaszähler-Impulse einlesen

```
sudo su
adduser fhem gpio
apt-get install wiringpi
```

Browserfenster öffnen: http://<IP_DES_R-PI>:8083/fhem

```
define GPIOGaszaehler RPI_GPIO 4
```

```
attr GPIOGaszaehler direction input
attr GPIOGaszaehler interrupt both
attr GPIOGaszaehler pud_resistor down

define Gasverbrauch HourCounter GPIOGaszaehler:on GPIOGaszaehler:off
attr Gasverbrauch userReadings Tagesverbrauchinqm:countsPerDay.*
{ReadingsVal("Gasverbrauch","countsPerDay",0)*0.01;;}
define myGasCalculator GasCalculator Gasverbrauch:countsOverall.*
setreading myGasCalculator GasCounterOffset nnnn.nn <-- durch aktuellen
Zählerstand ersetzen!

# über die GUI unter myGasCalculator -> GasCounterOffset 6088.89
# Readings zurücksetzen:
#  setreading Gasverbrauch Tagesverbrauchinqm 0.0
#  setreading GPIOGaszaehler Counter 0
```

From:
https://von-thuelen.de/ - **Christophs DokuWiki**

Permanent link:
https://von-thuelen.de/doku.php/wiki/projekte/fhem/uebersicht

Last update: **2018/10/13 08:19**