

Installation von Hometop HT3

Für folgende Hard und Software:

Raspberry Pi 3b

ht_pitiny Adpater

Raspbian Stretch minimal Image stand September

Basis dient die Anleitung von Norbert, entsprechend von mir angepasst.

3.1 Betriebssystem

Aktualisierung des Betriebssystem mit:

```
$ sudo apt-get update
```

Den letzten Ausgabestand aktivieren:

```
$ sudo apt-get upgrade
```

Python3 installieren

```
$ sudo apt-get install python3
```

Seriellen Treiber für Python3 laden:

```
$ sudo apt-get install python3-serial
```

setuptools und GPIO Treiber für Python3 laden:

```
$ sudo apt-get install python3-setuptools
```

```
$ sudo apt-get install RPI.GPIO
```

TK (GUI) Treiber für Python3 laden:

```
$ sudo apt-get install python3-tk
```

Perl objekt-orientiertes RRDTool Interface installieren:

```
$ sudo apt-get install librrdtool-oo-perl
```

anschliessend

```
$ sudo apt-get autoremove
```

RRDTool Datenbank installieren:

```
$ sudo apt-get install rrdtool
```

User in Gruppe <dialog> aufnehmen:

```
$ sudo adduser pi dialog
```

Deaktivieren der default eingeschalteten TTY-Systemausgaben (RaspberryPI):

```
$ sudo nano /boot/cmdline.txt
```

von

```
dwc_otg.lpm_enable=0 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200 console=tty1
```

..

in

```
dwc_otg.lpm_enable=0 console=tty1 root=PARTUUID...
```

ändern. danach Datei speichern

Deaktivieren des default eingeschalteten Bluetooth-Moduls

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

hinzugefügt am Ende

```
dtoverlay=pi3-disable-bt
```

```
# reboot
```

3.1.1 Komponenten für MQTT

Key für zukünftige Updates holen / installieren

```
$ sudo wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
$ sudo apt-key add mosquitto-repo.gpg.key
$ cd /etc/apt/sources.list.d
$ sudo wget http://repo.mosquitto.org/debian/mosquitto-stretch.list
```

System aktualisieren
\$ sudo apt-get update

Installation des MQTT-Broker

Vorab müssen folgende Kommandos ausgeführt werden, da wir sonst auf folgenden Konflikt stoßen werden:

```
mosquitto : Depends: libssl1.0.0 (>= 1.0.1) but it is not installable
```

```
Depends: libwebsockets3 (>= 1.2) but it is not installable
```

Daher führen wir vorab folgendes aus:

```
$ wget
http://ftp.us.debian.org/debian/pool/main/libw/libwebsockets/libwebsockets3\_1.2.2-1\_armhf.deb
$ wget http://ftp.us.debian.org/debian/pool/main/o/openssl/libssl1.0.0\_1.0.2l-1~bpo8+1\_armhf.deb
$ sudo dpkg -i libwebsockets3_1.2.2-1_armhf.deb
$ sudo dpkg -i libssl1.0.0_1.0.2l-1~bpo8+1_armhf.deb
```

Dann wie gewohnt fortfahren

```
$ sudo apt-get install mosquitto
```

```
$ sudo apt-get autoremove
```

Installation der MQTT Clients (mosquitto_sub und mosquitto_pub)

```
$ sudo apt-get install mosquitto-clients
```

Testen ob der Broker läuft

```
$ ps -aux |grep mosquitto
```

Resultat aus der ps - Anzeige:

```
... .. /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf
```

Installation Tool 'pip'

```
$ sudo apt-get install python3-pip
```

Holen von paho-mqtt, git etc.

```
$ sudo pip3 install paho-mqtt
```

```
$ sudo apt-get install git
```

```
$ sudo apt-get install insserv
```

Neustart des Raspberry Pi:
\$ sudo reboot

Die aktuelle Software mit Dokumentation von github.com holen (als user 'pi'):
\$ cd
\$ git clone https://github.com/norberts1/hometop_HT3.git
\$ Folder: HT3 zu ~/. verschieben
\$ mv ~/hometop_HT3/HT3 ~/.

3.1.2 rrdtool_draw.pl anpassen

\$ sudo nano /HT3/sw/etc/ rrdtool_draw.pl

%2.lf\l ändern in %2.0lf\l (3 Positionen)

Beispiel:

```
format => '%2.lf\l',
```

```
draw => 'Brennerleistung_last',
```

ändern in

```
format => '%2.0lf\l',
```

```
draw => 'Brennerleistung_last',
```

%6.lf ändern in %6.0lf (1 Position)

```
gprint
```

```
=>
```

```
{
```

```
format => ' Brenner ein \: %6.lf Zähler',
```

```
draw => 'brennereinheiz_last',
```

Ändern in

```
gprint
```

```
=>
```

```
{
```

```
format => ' Brenner ein \: %6.0lf Zähler',
```

```
draw => 'brennereinheiz_last',
```

Wer will, kann die **Zeitleiste** zur besseren Übersicht **anpassen**:
Wir bleiben in der rrdtool_draw.pl

```
# Set
```

```
Starttime
```

```
my $start_time = time()-2880*60;
```

Der Wert 2880 kann nach belieben eingestellt werden.

Ich habe dort 720 stehen.

Erklärung:

Gewünschte Stunden, die im Zeitstrahl angezeigt werden sollen * 60

Beispiele:

2880 = 48 Stunden

720 = 12 Stunden

360 = 6 Stunden

```
# Set  
Starttime  
my $start_time = time()-720*60;
```

Speichern

Ab hier kann dann wie von Norbert beschrieben fortgefahren werden.

EXTRAS

Desktop Environment

WICHTIG!

Da ich das Minimalimage als Grundlage nutze, fehlt der Desktop! Ohne Desktop kann später kein

HT3_Systemstatus

Bzw.

HT3_Analyser

ausgeführt werden. Diese setzen einen aktiven Desktop zur Grafikausgabe voraus.

Dazu gehen wir wie folgt vor:

1. XOrg installieren:

```
$sudo apt-get install --no-install-recommends xserver-xorg
```

2a.Mate Gui installieren – nutze ich!

```
$ sudo apt-get install mate-desktop-environment-core  
oder
```

2b. XFCE GUI

```
$ sudo apt-get install xfce4 xfce4-terminal  
oder
```

2c. LXDE GUI

```
$ sudo apt-get install lxde-core lxappearance
```

3. LightDM login manager

```
$ sudo apt-get install lightdm
```

Zum Starten

```
$ startx
```

Damit der Desktop später auch aufgerufen werden kann, installieren wir uns noch einen

TightVNCserver

```
$ sudo apt-get update
```

```
$ sudo apt-get install tightvncserver
```

```
$ vncpasswd
```

Kommando zum Starten einer VNC-Session:

```
$ vncserver :1
```

Zum Aufruf über einen Client geben wir dann z.B. 192.168.0.2:5901

Die :1 ist der jeweilige VNC Server und der resultierende Port = 5901

Haben wir einen zweiten VNCServer mit vncserver :2 gestartet ist er über z.B.

192.168.0.2:5902 erreichbar

Der VNC Server stoppen wir über

```
$ vncserver :1 -kill
```

Erweiterung: Autostart von TightVNCServer per systemd-Service

Zuerst erstellen wir eine neue Datei im systemd-Systemverzeichnis:

```
$ sudo nano /etc/systemd/system/vncserver@.service
```

Und kopieren folgenden Inhalt in das Fenster:

```
[Unit]
```

```
Description=VNC mit TightVNCServer
```

```
After=syslog.target network.target
```

```
[Service]
```

```
Type=forking
```

```
User=pi
```

```
PAMName=login
```

```
PIDFile=/home/pi/.vnc/%H:%i.pid
```

```
ExecStartPre=-/usr/bin/vncserver -kill :%i > /dev/null 2>&1
```

```
ExecStart=/usr/bin/vncserver -depth 24 -geometry 1280x800 :%i
```

```
ExecStop=/usr/bin/vncserver -kill :%i
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Die Zeile "ExecStart..." enthält VNC-Parameter "-depth" und "-geometry", die man nach eigenen belieben anpassen kann.

Dann speichern und schließen: Strg + O, Return, Strg + X

Dann muss man systemd noch mitteilen, dass es einen neuen Service/Daemon gibt.

```
$ sudo systemctl daemon-reload
```

Automatisches Starten beim Booten einschalten:

```
$ sudo systemctl enable vncserver@1.service
```

Anzeigen, ob der VNC-Dienst beim Booten automatisch gestartet wird:

```
$ sudo systemctl is-enabled vncserver@1.service
```

Steuern kann man den Dienst mit folgenden Kommandos:

VNC-Dienst starten:

```
$ sudo systemctl start vncserver@1.service
```

Laufenden VNC-Dienst stoppen:

```
$ sudo systemctl stop vncserver@1.service
```

Status des VNC-Dienstes anzeigen:

```
$ sudo systemctl status vncserver@1.service
```

Automatisches Starten beim Booten ausschalten:

```
$ sudo systemctl disable vncserver@1.service
```

Das sollte es gewesen sein.

Viel Erfolg

URBANsUNITED