

Integration der Wärmepumpe ROTEX HPSU 518 (Bj. 2013) im ioBroker

Alles was hier steht hat bei mir („RaspiUser“) funktioniert ...
Hier nur als To-Do-Anleitung, als copy & paste (ggf. Änderungen, s. Text) ohne nähere Erläuterungen o.ä.
Wie immer: ALLES AUF EIGENE GEFAHR ! Ohne Gewähr.

Meine Wärmepumpe ist vom ioBroker räumlich getrennt. Da geht keine Kabelverbindung und es bietet sich ein WLAN Zugang an.

Ich habe hier

<https://forum.iobroker.net/topic/52171/rotex-hpsu-daikin-altherma-w%C3%A4rmepumpe-%C3%BCber-iobroker-canbus>

den Anstoss zur Umsetzung bekommen, dazu ergänzt durch die sehr guten Anleitungen

<https://crycode.de/can-bus-am-raspberry-pi>

und

<https://crycode.de/socketcan-ueber-ethernet-mit-cannelloni>

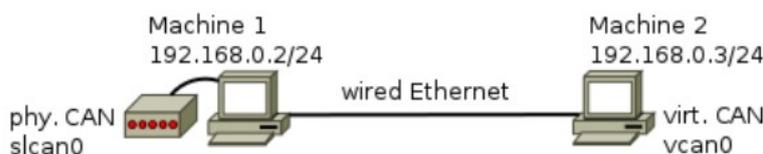
und dann für **meine** Hardware die Tätigkeiten in dieser Anleitung, ohne hin und her zwischen den Links, zusammengefasst ...

Wer genauere Info's zur Installation o.ä. möchte muss also (doch) dort nachlesen

Danke ! an crycode für die ausführliche Erläuterungen und seine Scripte u.v.a.!!!

Prinzipiell sieht der Aufbau so aus (<https://github.com/mguentner/cannelloni>):

Two machines 1 and 2 need to be connected:



Machine 2 needs to be connected to the physical CAN Bus that is attached to Machine 1.

„Machine 1“ ist der Pi Zero W mit CAN HAT an der Wärmepumpe, die reale „can0“ Adresse. Die „Machine 2“ ist der ioBroker-Raspberry Pi, also die virtuelle „vcan0“ Adresse, die dann via WLAN mit dem Zero „verbunden“ ist.

CAN Bus und CAN HAT auf dem Pi Zero W installieren bzw. einbinden

- x Mit dem Raspbian-Imager <https://www.raspberrypi.com/software/> die „Raspberry Pi OS Lite (Legacy)“ Variante auf die SD-Karte (16 GB) schreiben. Zuvor nicht vergessen mit dem „Zahnrad“ alle gewünschten Einstellungen (bei mir speziell das WLAN) zu wählen.

- x Die SD-Karte im Zero einstecken, starten und die Installation aktualisieren ...

```
sudo apt-get update  
sudo apt-get upgrade
```

- x Die CAN Software installieren (ist bereits im Image enthalten)...

```
sudo apt install can-utils
```

- x Die CAN HAT Hardware (meine ist https://www.waveshare.com/wiki/RS485_CAN_HAT) aktivieren (ggf. Frequenz und GPIO anpassen) ...

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

... am Ende diesen Text einfügen (ggf. Frequenz und GPIO anpassen) und speichern

```
dtoverlay=mcp2515-can0,oscillator=12000000,interrupt=25  
dtoverlay=spi-bcm2835-overlay
```

- x Den Zero neu starten.

- x Mit

```
ls /sys/bus/spi/devices/spi0.0  
ls /sys/bus/spi/devices/spi0.0/net  
ls /sys/class/net
```

die Installation prüfen. Hier sollten keine Fehlermeldungen auftauchen. Beim letzten Befehl kommt bei mir z.B. als Antwort

```
can0 wlan lo
```

- x Jetzt das can0 Interface für den automatischen Start einstellen ...

```
cd /etc/network
```

```
sudo nano interfaces
```

... und am Ende diesen Text einfügen (ggf. Bitrate anpassen) und speichern ...

```
# CAN-Bus  
auto can0  
iface can0 can static  
    bitrate 20000
```

- x Jetzt den Zero ausschalten um die Hardware der HPSU zu kontaktieren ...

```
sudo shutdown
```

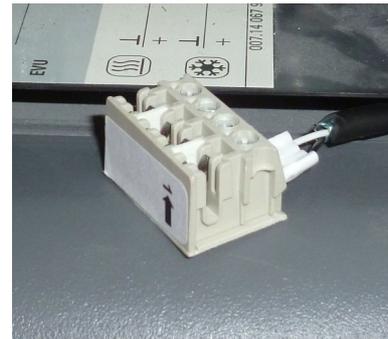
ALLE Sicherungen der HPSU ausschalten !!! LEBENSGEFAHR !!!

Die Haube abnehmen, das Bedienteil links und rechts mit einem Schraubendreher entclipsen und nach oben weg klappen. Die „Nasen“ in die dafür vorgesehenen Aussparungen einschieben damit nix abreißt.

Oben auf der Platine „RoCon BM1“ sehen wir nun den Stecker „J13“. Diesen nach oben abziehen. Auf der Steckerrückseite sind Schraubklemmen.

Ich verwende 1 verdrilltes (!!)

abgeschirmtes Adernpaar von einem abgeschirmten CAT-Kabel. „PIN 1“ ist CAN-H und der daneben CAN-L. Das Kabel nach außen führen und dann 1:1 mit dem CAN HAT des Zero kontaktieren. Alles wieder rück-bauen ...



- x Nachdem man den Zero gestartet hat und soweit alles geklappt hat sollte man mit diesem Befehl im Terminal ...

```
candump -tA -x can0
```

... nach kurzer Zeit (etwas Geduld ...) einige Werte von der Wärmepumpe sehen können, z.B.

```
(2023-04-01 15:54:44.888061) can0 RX - - 180 [7] 20 0A 0C 00 66 00 00  
(2023-04-01 15:54:48.820103) can0 RX - - 180 [7] 20 0A FA 01 D6 01 06
```

CAN Bus auf dem ioBroker Raspberry

Die CAN Software installieren (ist bereits im Image enthalten)...

```
sudo apt install can-utils
```

SocketCAN über Ethernet mit Cannelloni

Auf **beiden** „Machine“ muss die Cannelloni-Software installiert sein ...

```
sudo apt install build-essential cmake git

git clone https://github.com/mguentner/cannelloni.git
cd cannelloni
cmake -DCMAKE_BUILD_TYPE=Release
make

sudo make install
```

Permanenter Betrieb über SystemD-Services

Damit der CAN Bus immer arbeitet muss man noch ein paar Änderungen vornehmen

Einrichtung auf dem System mit virtuellem „vcan0“, hier ioBroker Raspi 192.168.178.36

```
sudo nano /usr/local/sbin/vcan0-setup.sh
```

x ... diesen Text (ggf. Bitrate ändern) einfügen und speichern ...

```
#!/bin/bash
set -e
IFACE=vcan0
RATE=20kbit

case "$1" in
  start)
    modprobe vcan
    ip link add name $IFACE type vcan
    ip link set dev $IFACE up
    tc qdisc add dev $IFACE root tbf rate $RATE latency 100ms
    burst 1000
    ;;
  stop)
    ip link del name $IFACE
    ;;
  *)
    echo "Usage: $0 <start|stop>" >&2
    exit 1
esac
```

x ... die Datei Rechte anpassen ...

```
sudo chmod u+x /usr/local/sbin/vcan0-setup.sh
```

x ... 1ten SystemD-Service einrichten ...

```
sudo nano /etc/systemd/system/vcan0-setup.service
```

x ... diesen Text einfügen & speichern ...

```
[Unit]
Description=Setup vcan0
After=network.target

[Service]
Type=oneshot
ExecStart=/usr/local/sbin/vcan0-setup.sh start
ExecStop=/usr/local/sbin/vcan0-setup.sh stop
RemainAfterExit=true

[Install]
WantedBy=multi-user.target
```

x ... 2ten SystemD-Service einrichten ...

```
sudo nano /etc/systemd/system/cannelloni-vcan0.service
```

x ... diesen Text (ggf. IP Adresse ändern) einfügen & speichern ...

```
[Unit]
Description=Cannelloni for vcan0
After=network.target
Requires=vcan0-setup.service

[Service]
ExecStart=/usr/local/bin/cannelloni -I vcan0 -R 192.168.178.92
Environment=LD_LIBRARY_PATH=/usr/local/lib
Restart=always
RestartSec=3s
User=nobody
Group=nogroup

[Install]
WantedBy=multi-user.target
```

x ... Services starten ...

```
sudo systemctl daemon-reload
sudo systemctl enable vcan0-setup.service
sudo systemctl enable cannelloni-vcan0.service
sudo systemctl start vcan0-setup.service
sudo systemctl start cannelloni-vcan0.service
```

Einrichtung auf dem System mit physischem „can0“, hier Zero W Raspi 192.168.178.92

```
sudo nano /etc/systemd/system/cannelloni-can0.service
```

- x ... diesen Text (ggf. IP Adresse ändern) einfügen & speichern ...

```
[Unit]
Description=Cannelloni for can0
After=network.target

[Service]
ExecStart=/usr/local/bin/cannelloni -I can0 -R 192.168.178.36
Environment=LD_LIBRARY_PATH=/usr/local/lib
Restart=always
RestartSec=3s
User=nobody
Group=nogroup

[Install]
WantedBy=multi-user.target
```

- x ... Service starten ...

```
sudo systemctl daemon-reload
sudo systemctl enable cannelloni-can0.service
sudo systemctl start cannelloni-can0.service
```

- x Jetzt sollten in einem Terminal auf dem **ioBroker Raspi** mit z.B. „candump vcan0“ die Werte der HPSU ankommen ...

ioBroker

- x Im ioBroker den CAN-Adapter installieren, eine Instanz anlegen und unter „Allgemein“ die Schnittstelle „vcan0“ eintragen.
- x Unter „Import / Export“ und „Konfigurationen von Github abrufen“ die config-Datei „Rotex HPSU Compact / Daikin Altherma 1.6.0“ (Stand 01.04.23) aus Github importieren.
- x Nach dem Speichern und dem Instanzen-Start füllen sich die „knapp 1000“ Objekte, zuerst nur mit „(Null)“.
Aktuelle Werte kommen nur, wenn man entweder aktiv an dem Heizungsbedienteil etwas abfragt / einstellt oder über die ID „680“ via Adapter die Aktualisierung anstößt.
- x **Sehr wichtig ist es nun, sich in den Beiträgen ...**

<https://forum.iobroker.net/topic/52166/adapter-can-bus-beschreibung-faq-diskussionen-etc> ,
<https://forum.iobroker.net/topic/39033/test-adapter-canbus-v1-1-x-latest>, und
<https://forum.iobroker.net/topic/52171/rotex-hpsu-daikin-altherma-w%C3%A4rmepumpe-%C3%BCber-iobroker-canbus/261>,

... über die Adapter Bedienung, Objekte (!) u.v.a. zu informieren bzw. !!!!!