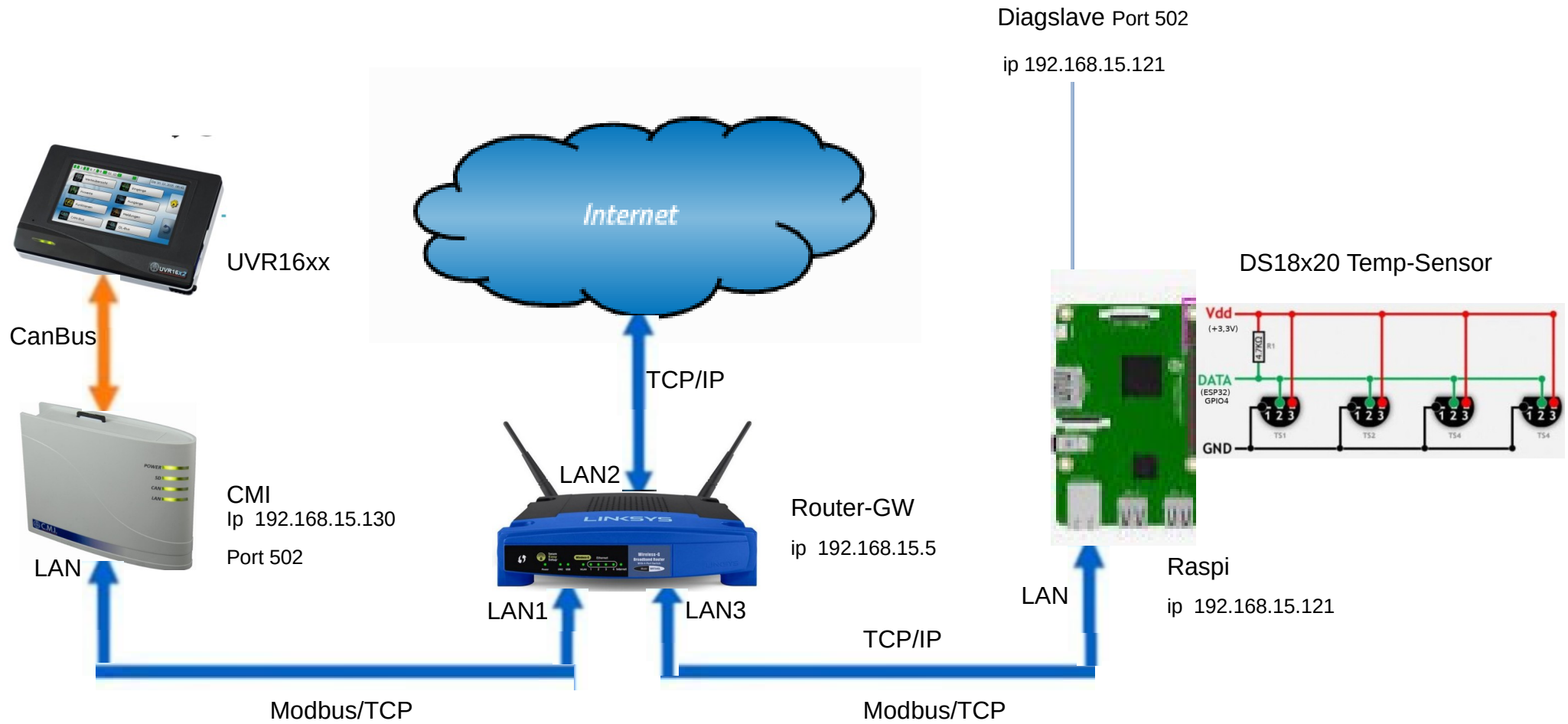
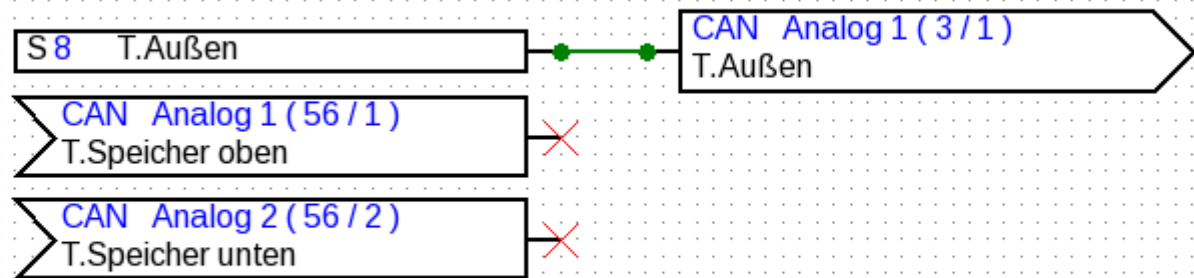


Modbusverbindung Raspi mit UVR16x2



CAN:



CAN-Bus

CAN-Analogausgänge

1: T.AußenAuto-83.2 °C

2: unbenutzt

3: unbenutzt

3

Ethernet

CAN

Meldungen

Kontakte

Passwörter

Datenlogging

Zeit

Eingänge

Ausgänge

Eingänge

CAN-BUS

Analog

1: S1

2: s2

3: CAN_AN_I_03

4: s4

5: s5

6: s6

7: s7

8: s8

9: s9

10: s10

11: s11

12: s12

13: s13

14: s14

CAN-BUS

Bezeichnung:

CAN_AN_I_03

Knotennummer:

3

Netzwerkausgang:

1

Timeout (Min):

20

Einheit:

Temperatur

Wert bei Timeout:

Unverändert

Ausgabewert:

0.00000

aktueller Wert:

-82.7 °C

Ethernet

CAN

Meldungen

Kontakte

Passwörter

Datenlogging

Zeit

Eingänge

Ausgänge

Ausgänge

CAN-BUS

Analog

Digital

Modbus Master

Analog

1: s1

2: s2

3: MOD_AN_O_03

4: s4

5: s5

6: s6

7: s7

8: s8

9: s9

10: s10

Modbus

Bezeichnung:

MOD_AN_O_03

Eingang:

CAN-Bus

3: CAN_AN_I_03

Messwert

IP:

192.168.15.121

Gerät:

1

Funktion:

06-preset single register

Adresse:

110

Datentyp:

32-bit signed

Byte-Reihenfolge:

Big-endian

Faktor:

1.00000

Sendebedingung:

bei Änderung >

1.00000

Blockierzeit (Sek):

5

Intervallzeit (Min):

1

aktueller Wert:

-827

Python-script mit ip-adresse vom diagslave

- → lesen adresse 100 - 115

...

```
# z.B. werden 15 Modbus-Register mit der Adresse 100-115 gelesen
regs = c.read_holding_registers(100, 15)
# if success display registers
if regs:
    print("reg ad #100 to 115: "+str(regs))
```

...

Ausgaben vom python-script

```
pi@raspi4red:~/daten/projekte/py_ds18b20 $ python3 Modbus_ds18s20_3write_read_single_register.py
define modbus localhost:502
```

...

```
Tx
[64 3A 00 00 00 06 01] 03 00 64 00 0F
Rx
[64 3A 00 00 00 21 01] 03 1E 03 09 00 00 59 5B 00 00 00 00 00 00 00 00 00 00 00 00 FC C5 00 00 00 00 00 00 00 00
reg ad #100 to 115: [777, 0, 22875, 0, 0, 0, 0, 0, 0, 0, 64709, 0, 0, 0, 0]
```

...