Interface S88 Gleisbox Raspberry Pi Manual

Guido de Hek

December 6, 2023

Contents

1 Introduction **4**

2 Setup of the Raspberry Pi **5**

2.1 Install BCM2835 v1.73 5

2.2 Shutdown Button 5

2.3 Remote Desktop Protocol 6

3 Setup of CAN-interface **7**

3.1 Connection Scheme 7

3.2 Oscillator Settings 7

3.3 PiCanS88 9

4 Setup of S88N-interface **11**

4.1 Connection Scheme 11

4.2 S88UDP installation 11

5 Setup of Rocrail **12**

5.1 Installation 12

5.2 Rocrail Controller Settings 12

6 PCB Description **13**

Credits

Parts of this system have been developed by members of the model railway community. This is mainly applicable for the s88udp conversion, as well as srcpd. Many thanks and full credits to them!

1 Introduction

System for controlling a marklin track using a rpi

2 Setup of the Raspberry Pi

Prepare an SD-card with Raspberry Pi OS. I have used the version from October 10th 2023 (64-bit) with desktop but without recommended software. My system uses a Raspberry Pi 4B. The advantage of desktop support is that a standalone system can be created by attaching an external monitor, mouse and keyboard. This creates the possibillity of implementing a very small and modular control system.

Refer to the Raspberry Pi website for instructions. Follow the instructions to setup the raspberry pi OS. For remote control of the raspberry pi it is possible to enable SSH and/or VNC. Refer to the setup instructions for more information. There is a lot of information available about the setup and realizing SSH or VNC communication on Raspberry Pi support forums.

2.1 Install BCM2835 v1.73

Install BCM2835-1.73 so that the pin IO on the Raspberry Pi can be used:

```
wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.73.tar.gz

tar zxvf bcm2835-1.73.tar.gz

cd bcm2835-1.73

./configure

make

sudo make check

sudo make install
```

2.2 Shutdown Button

A button can be connected to the system to simplify the process of shutting the system down. First of all, a pushbutton must be connected between GPIO3 (header pin 5) and GND (e.g. header pin 6).

Next, the following script must be installed:

```
git clone https://github.com/Howchoo/pi-power-button.git

./pi-power-button/script/install
```

Uninstalling the script can be done via:

```
./pi-power-button/script/uninstall
```

Note: warning about pull-up resistor can be neglected.

2.3 Remote Desktop Protocol

To install RDP run:

```
sudo apt update
sudo apt upgrade
sudo apt install xrdp
```

3 Setup of CAN-interface

t.b.d.

3.1 Connection Scheme

The CAN-bus shall be connected to the Gleisbox as depicted in figure 1. The pin 1 (power supply) does not have to be connected.

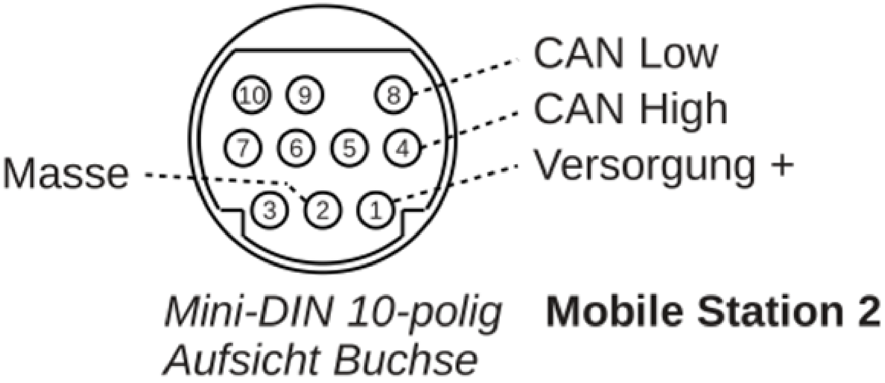


Figure 1: Pinout canbus.

3.2 Oscillator Settings

Oscillator settings for the CAN peripheral need to be set up.
Open config.txt:

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

In the section for optional hardware interfaces add (if not present):

```
dtparam=spi=on
dtoverlay=mcp2515-can0,oscillator=16000000,interrupt=25
dtoverlay=spi-lcs
```

Use ctrl+X to stop editing, press Y and enter to save the changes.
Reboot using:

```
sudo reboot
```

To test that can0 is working properly start the connection using:

```
sudo ip link set can0 up type can bitrate 250000
```

Check the connection status:

```
ifconfig
```

In the output, can0 should be visible. Output should be similar to this (UP, RUNNING):

```
can0: flags=193<UP,RUNNING,NOARP>  mtu 16
unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 10  (UNSPEC)
RX packets 0  bytes 0 (0.0 B)
RX errors 0  dropped 0  overruns 0  frame 0
TX packets 0  bytes 0 (0.0 B)
TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Now that we know that the communication with the MCP2515 is working properly. The next step is to check if the CAN transceiver is able to listen to the CAN-bus. To do this, the CAN helper tools are needed.

To avoid the error: /autogen.sh: line 20: autoreconf: command not found ; execute the following commands first:

```
sudo apt-get install autoconf
sudo apt-get install libtool
```

Install the CAN helper tools using:

```
git clone https://github.com/linux-can/can-utils.git

cd can-utils

./autogen.sh

./configure

make

sudo make install
```

Now we can test the board. Connect the Gleisbox with the Mobile Station 2 and the Raspberry Pi (via the board). Power on the Gleisbox and the Mobile Station.
Monitor the CAN bus messages using the candump function:

```
sudo ./candump can0
```

Press the stop button on the mobile station a few times. Messages should appear and they should look something like this:

```
can0  0036936E   [5]  00 00 00 00 11
can0  0030936E   [0]
can0  0031931F   [8]  47 44 D8 D6 01 27 00 10
can0  0000936E   [7]  00 00 00 00 09 00 09
can0  0000936E   [6]  00 00 00 00 08 07
can0  0001931F   [7]  00 00 00 00 09 00 09
can0  0000936E   [5]  00 00 00 00 01
can0  0001931F   [6]  00 00 00 00 08 07
can0  0001931F   [5]  00 00 00 00 01
can0  0000936E   [5]  00 00 00 00 00
can0  0001931F   [5]  00 00 00 00 00
can0  0000936E   [7]  00 00 00 00 09 00 09
can0  0000936E   [6]  00 00 00 00 08 07
can0  0001931F   [7]  00 00 00 00 09 00 09
can0  0000936E   [5]  00 00 00 00 01
can0  0001931F   [6]  00 00 00 00 08 07
can0  0001931F   [5]  00 00 00 00 01
can0  0030936E   [0]
can0  0031931F   [8]  47 44 D8 D6 01 27 00 10
```

When a similar output is visible, the connection with the Gleisbox and the track is established succesfully! If a retest of the CAN bus is needed; simply navigate to the folder 'can-utils' and use the command sudo ./candump can0. Exit the candump function using ctrl+C.
Now, make sure that can0 starts up automatically. Open /etc/network/interfaces:

```
sudo nano /etc/network/interfaces
```

Add the following lines:

```
auto can0
iface can0 can static
bitrate 250000
```

Use ctrl+X to stop editing, press Y and enter to save the changes.

3.3 PiCanS88

The next step is to install *PiCanS88* to establish the CAN and S88 connection. To install PiCanS88:

```
cd ..
wget http://www.ifoedit.com/PiCanS88.tar
tar xvf PiCanS88.tar
cd PiCanS88
make
```

Now you can start it with (where the IP would be the IP of your Rocrailserver):

```
sudo ./PiCanS88 -f -b 192.168.0.255 -v -d 192.168.0.58 -m 1
```

Explanation of possible options:

```
-f (stay in forground and show details on screen)
-v (show details on screen)
-b 192.168.2.255 (broadcast address of your local home network)
-d 192.168.2.96 (IP address of the Computer where your Rocrail server is running)
-m 1 (number of connected S88 modules with 16 signals each) - (If you have modules with only
```

Table 1: S88N pinout and description.

RJ45 pin	Colour in UTP cable	S88N Description
1	Orange-white	+5V (+12V not in this board)
2	Orange	Data
3	Green-white	GND
4	Blue	Clock
5	Blue-white	GND
6	Green	Load
7	Brown-white	Reset
8	Brown	Rail signal (not used in this design)

4 Setup of S88N-interface

The S88N interface is used to obtain data about occupied track sections. This interface is based on 5Vdc (according to the standard 12Vdc is also possible).

4.1 Connection Scheme

Table 1 shows the S88N pin definitions as well as the UTP cable colors.

4.2 S88UDP installation

Install the following libraries first (to prevent pcap.h compilation error):

```
sudo apt-get install zlib1g-dev libpcap-dev
```

Download and install s88udp-rpi:

```
cd

git clone https://github.com/GBert/railroad

cd railroad/can2udp/src

make
```

To start the interface:

```
sudo ./s88udp-rpi -v -f -c "17,22,23,24" -m 1
```

Arguments behind option -c are the gpio ports. The amount of S88 modules is set using option -m.

To test if the udp ports are assigned for use by Rocrail:

```
sudo netstat -autpn | egrep "Proto|157"
```

The PID ”Rocrail” should be displayed.

5 Setup of Rocrail

t.b.d.

5.1 Installation

To download and install rocrail execute:

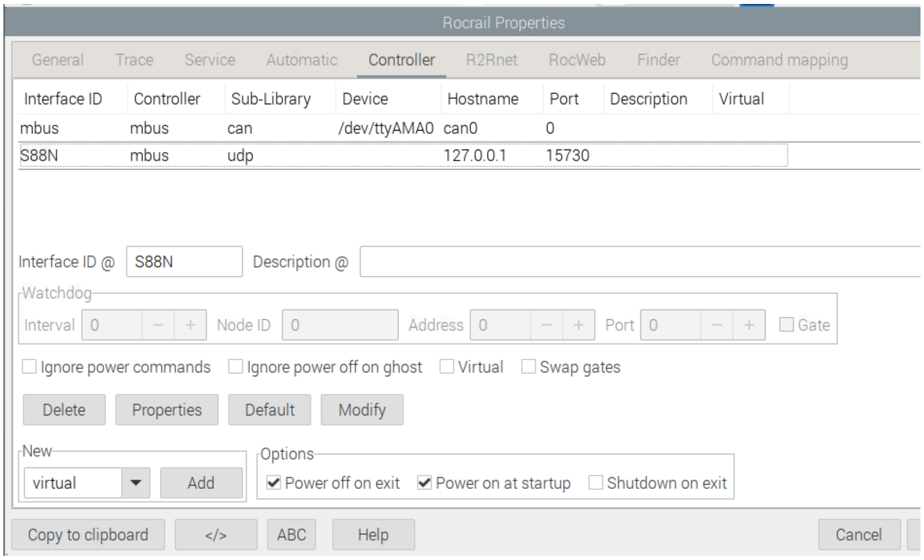
```
wget https://wiki.rocrail.net/rocrail-snapshot/Rocrail-PiOS11-ARM64.zip
unzip Rocrail-PiOS11-ARM64.zip -d ~/Rocrail
```

Run Rocrail server using:

```
./startrocrail.sh
```

5.2 Rocrail Controller Settings

Maak twee controllers, 1 voor can, en 1 voor S88N. screenshots van beide controllers staan hieronder.



Interface ID	Controller	Sub-Library	Device	Hostname	Port	Description	Virtual
mbus	mbus	can	/dev/ttyAMA0	can0	0		
S88N	mbus	udp		127.0.0.1	15730		

Interface ID @ S88N Description @

Watchdog

Interval 0 Node ID 0 Address 0 Port 0 □ Gate

□ Ignore power commands □ Ignore power off on ghost □ Virtual □ Swap gates

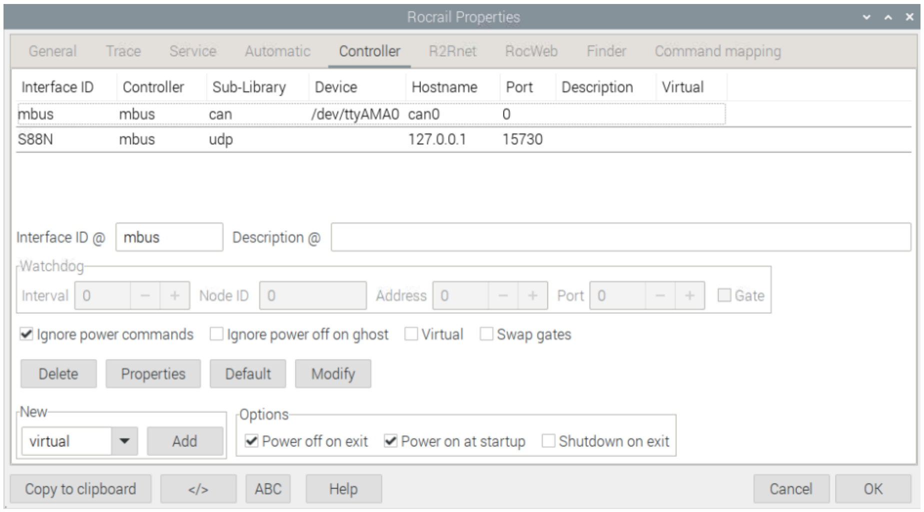
New

virtual

Options

☑ Power off on exit ☑ Power on at startup □ Shutdown on exit

Figure 2: General controller settings.



Interface ID	Controller	Sub-Library	Device	Hostname	Port	Description	Virtual
mbus	mbus	can	/dev/ttyAMA0	can0	0		
S88N	mbus	udp		127.0.0.1	15730		

Interface ID @ mbus Description @

Watchdog

Interval 0 Node ID 0 Address 0 Port 0 □ Gate

☑ Ignore power commands □ Ignore power off on ghost □ Virtual □ Swap gates

New

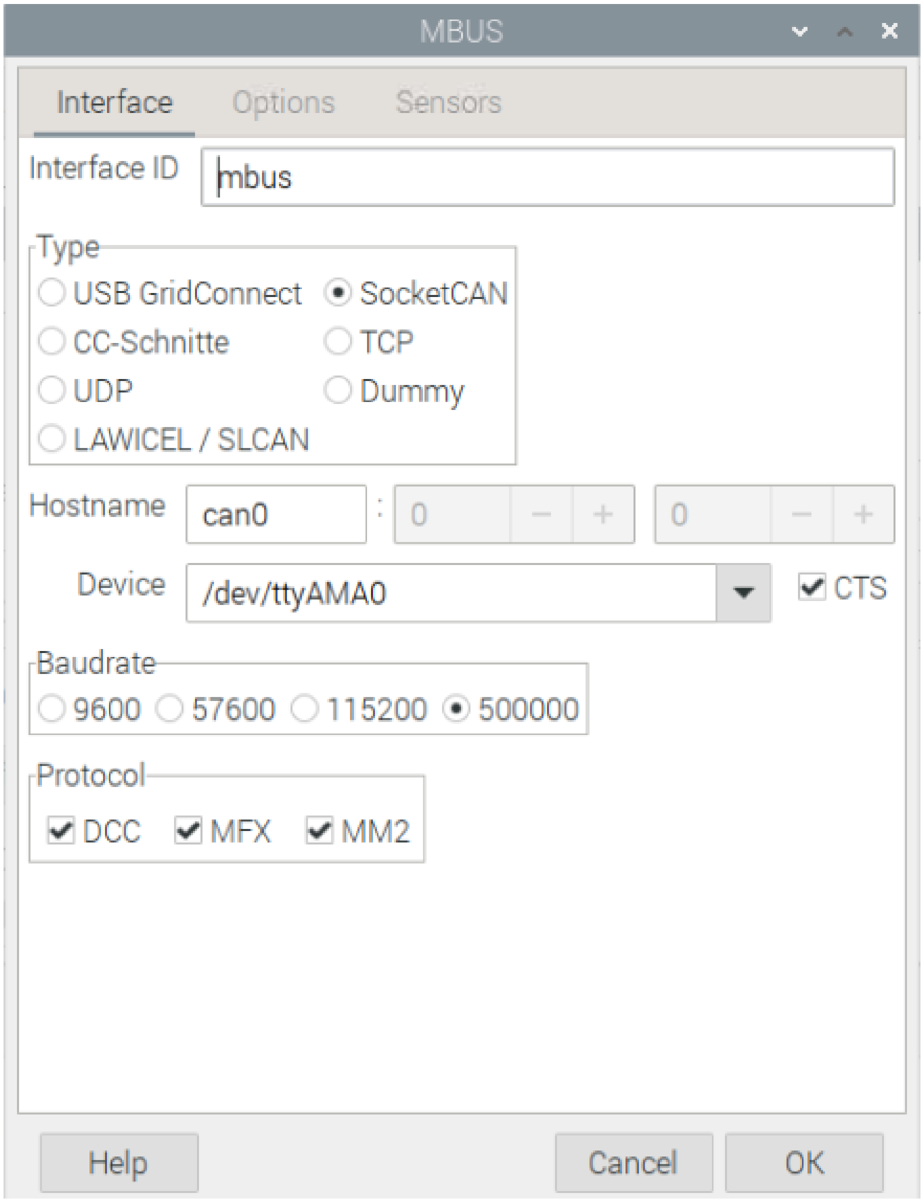
virtual

Options

☑ Power off on exit ☑ Power on at startup □ Shutdown on exit

Figure 3: General controller settings.

6 PCB Description



Interface ID: mbus

Type

○ USB GridConnect ● SocketCAN
○ CC-Schnitte ○ TCP
○ UDP ○ Dummy
○ LAWICEL / SLCAN

Hostname: can0 : 0 − + 0 − +

Device: /dev/ttyAMA0 ☑ CTS

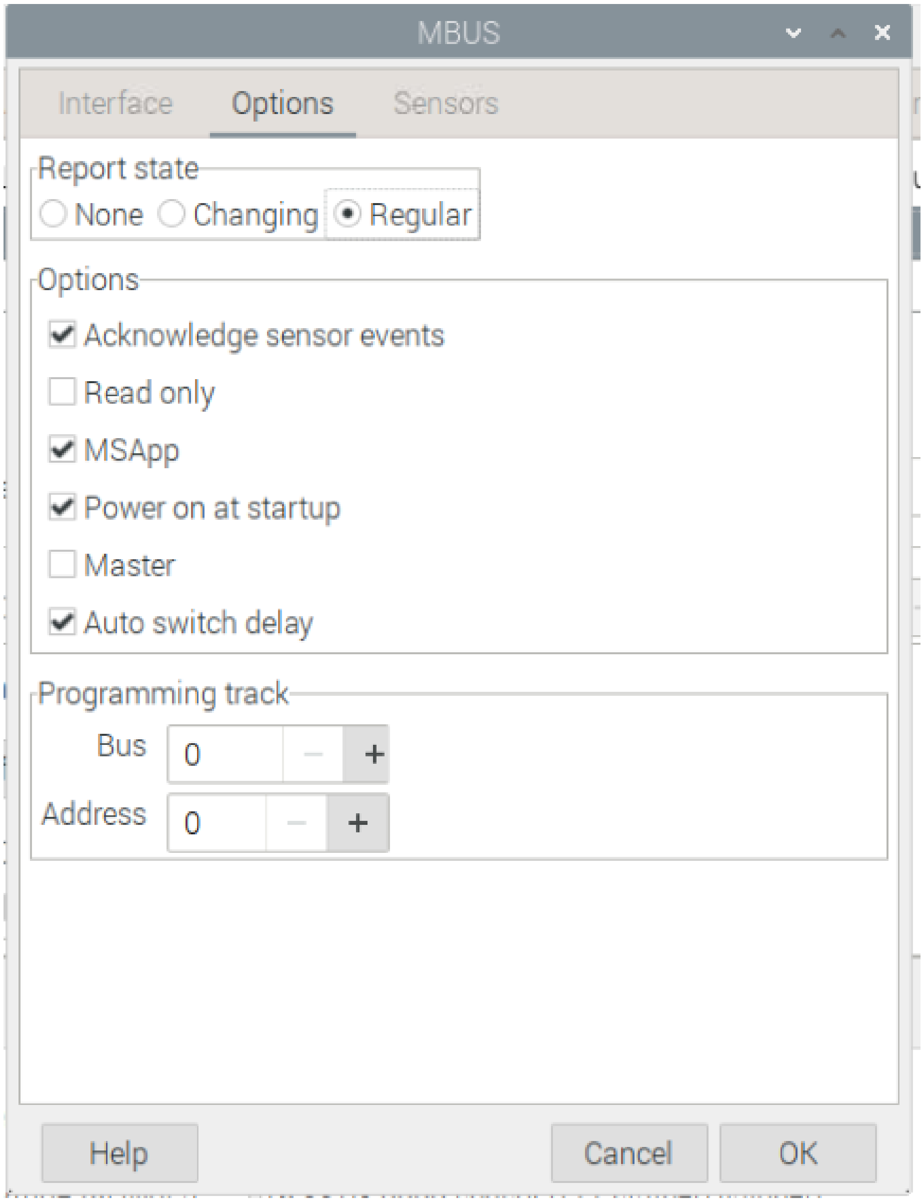
Baudrate

○ 9600 ○ 57600 ○ 115200 ● 500000

Protocol

☑ DCC ☑ MFX ☑ MM2

Figure 4: Controller settings mbus-can tab 1.



Report state

○ None ○ Changing ● Regular

Options

☑ Acknowledge sensor events

□ Read only

☑ MSApp

☑ Power on at startup

□ Master

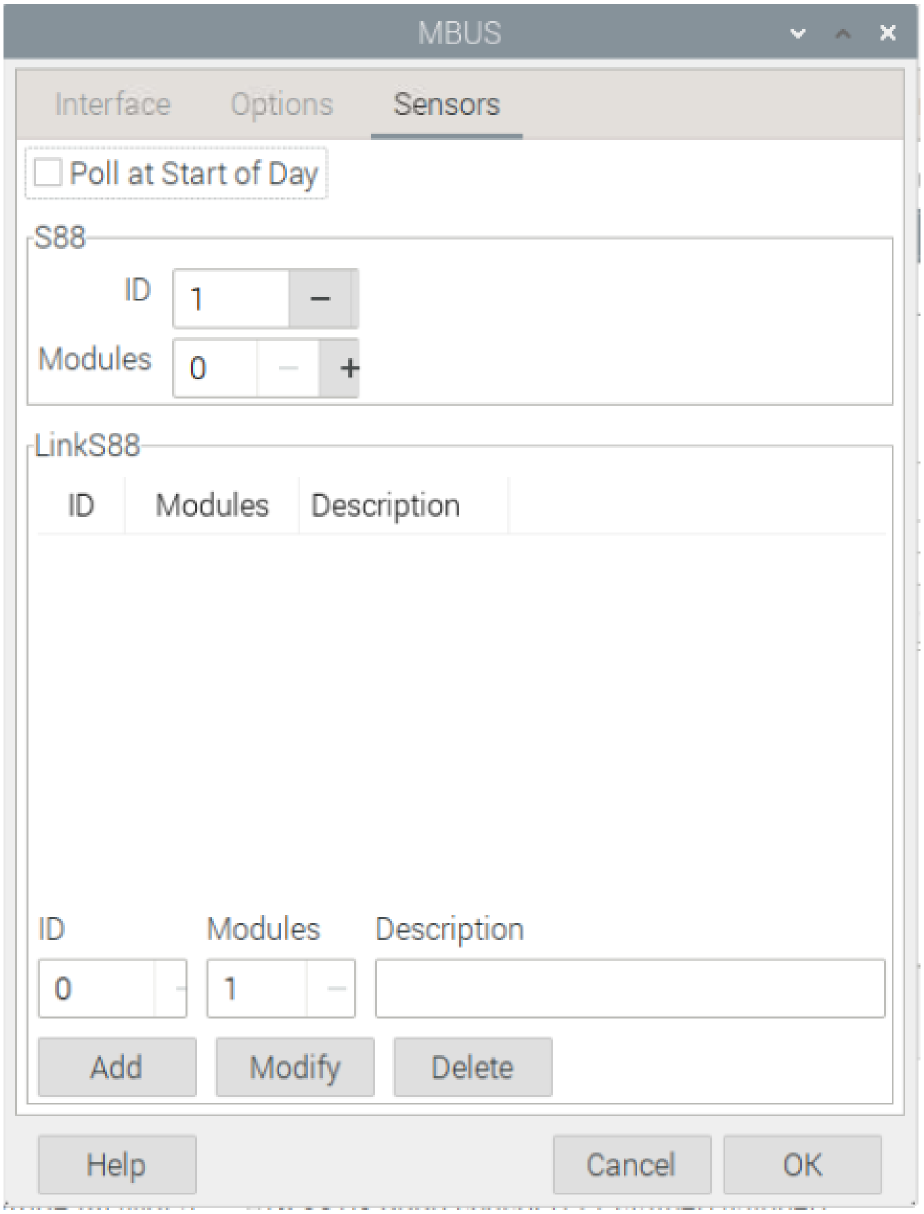
☑ Auto switch delay

Programming track

Bus 0

Address 0

Figure 5: Controller settings mbus-can tab 2.



□ Poll at Start of Day

S88

ID: 1

Modules: 0

LinkS88

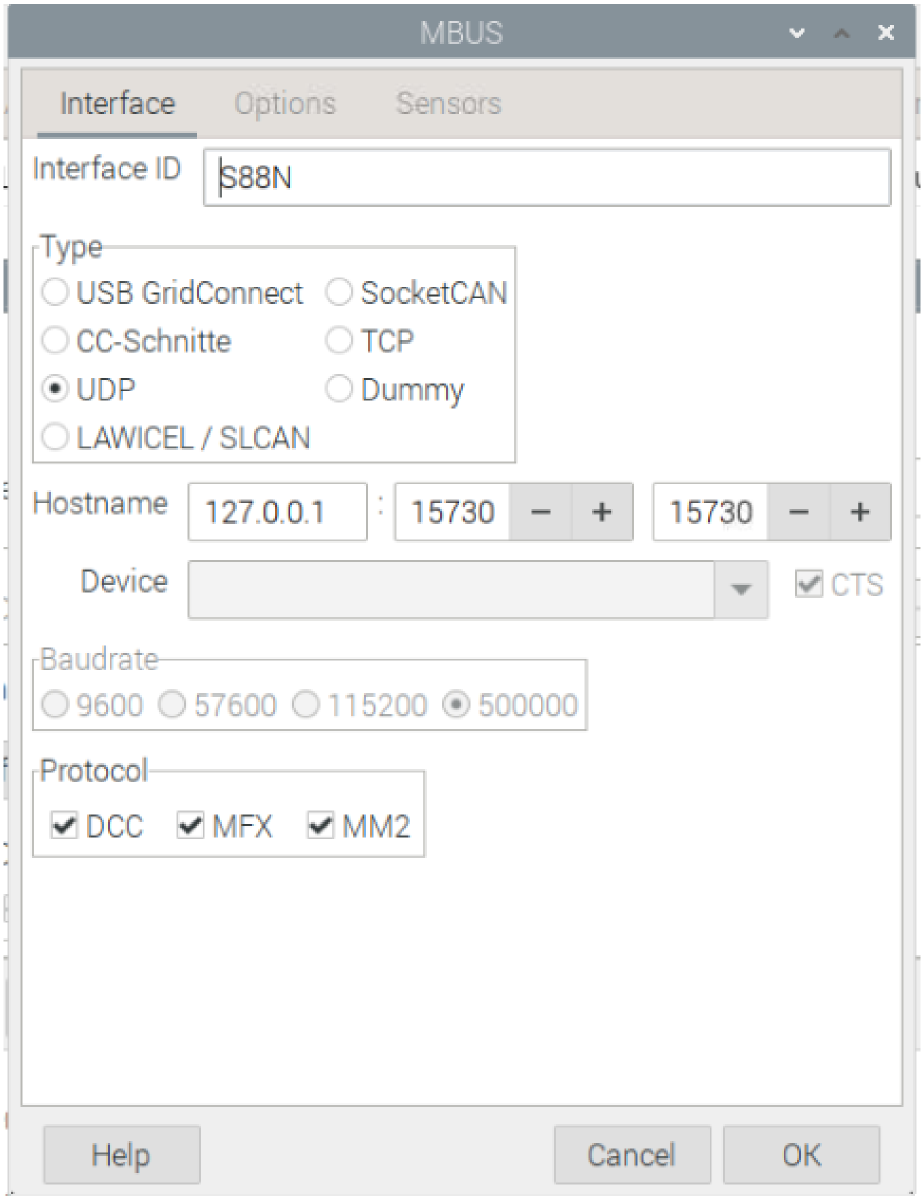
ID	Modules	Description

ID: 0

Modules: 1

Description:

Figure 6: Controller settings mbus-can tab 3.



Interface ID: S88N

Type
○ USB GridConnect ○ SocketCAN
○ CC-Schnitte ○ TCP
● UDP ○ Dummy
○ LAWICEL / SLCAN

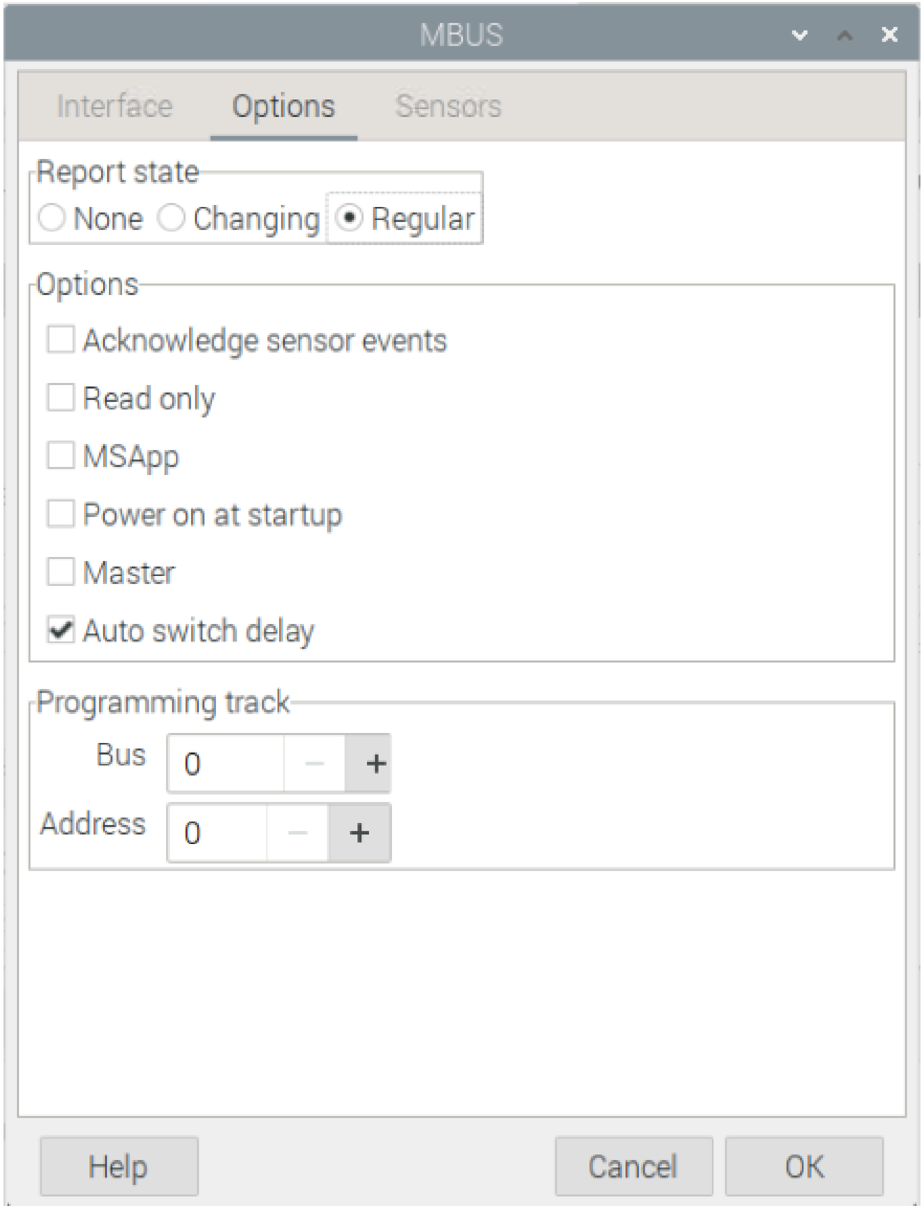
Hostname: 127.0.0.1 : 15730 − + 15730 − +

Device: ☑ CTS

Baudrate
○ 9600 ○ 57600 ○ 115200 ● 500000

Protocol
☑ DCC ☑ MFX ☑ MM2

Figure 7: Controller settings mbus-s88 tab 1.

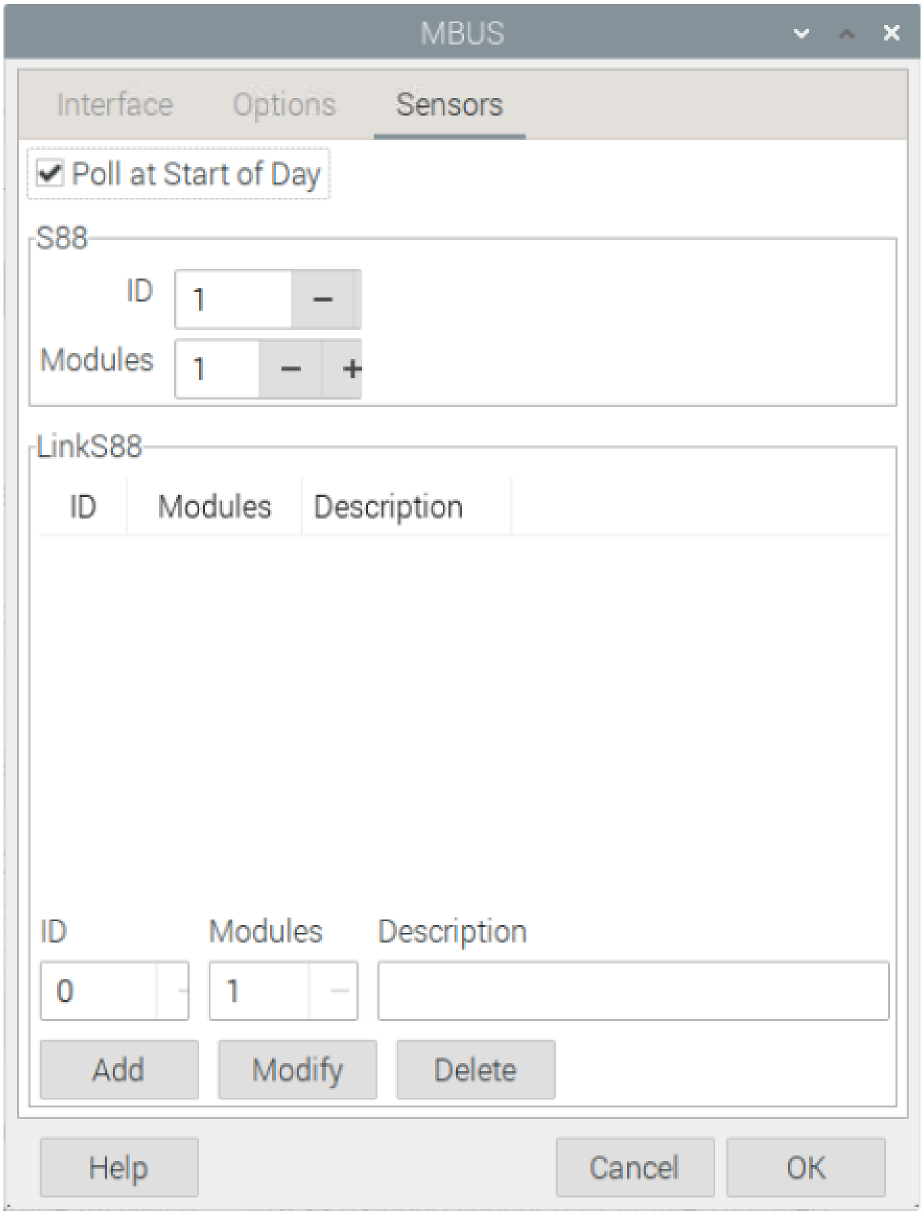


Report state
○ None ○ Changing ● Regular

Options
□ Acknowledge sensor events
□ Read only
□ MSApp
□ Power on at startup
□ Master
☑ Auto switch delay

Programming track
Bus 0
Address 0

Figure 8: Controller settings mbus-s88 tab 2.



☑ Poll at Start of Day

S88

ID 1

Modules 1

LinkS88

ID	Modules	Description

ID 0

Modules 1

Description

Figure 9: Controller settings mbus-s88 tab 3.

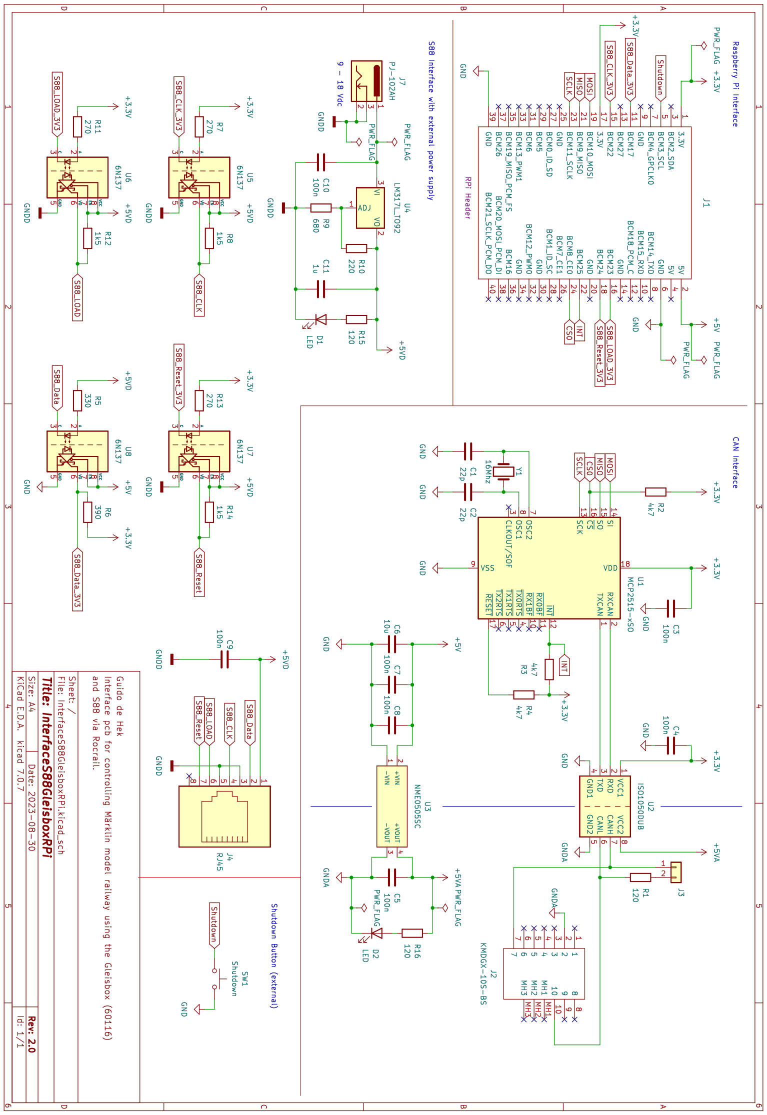


Figure 10: Schematic of the system.