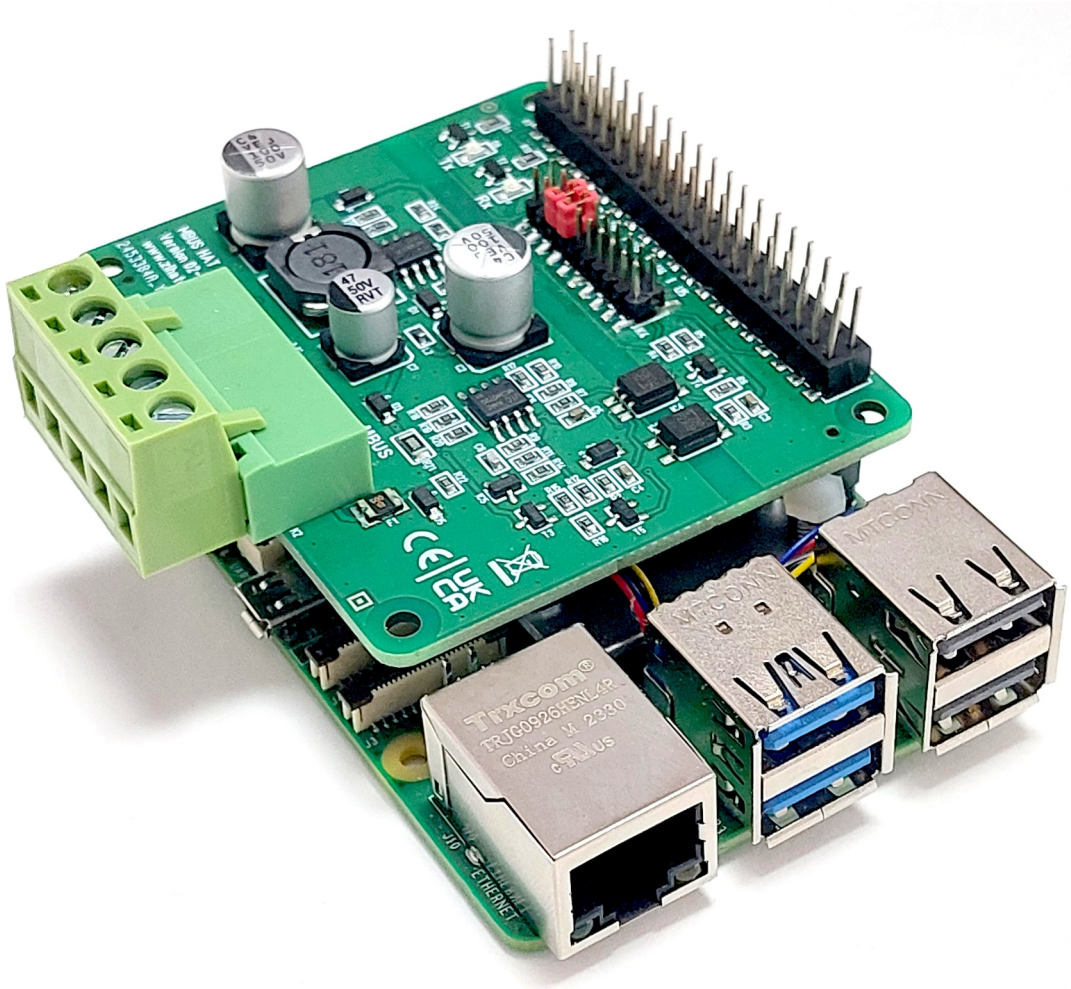
M-Bus HAT

for Raspberry Pi and compatible



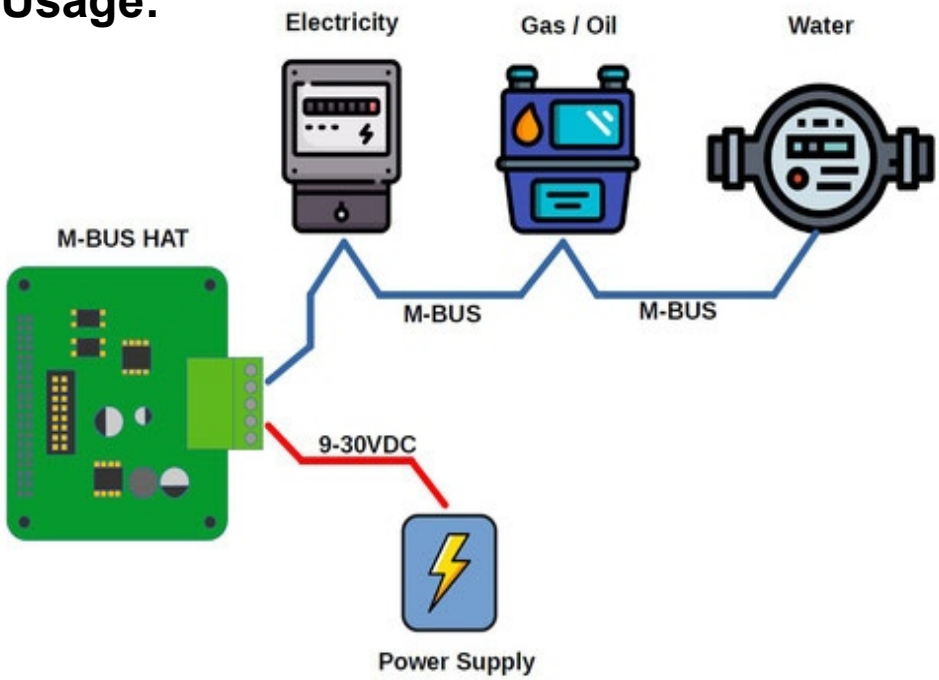
Features:

- M-Bus (Meter-Bus) master
- Compatible to European standard EN 13757-2
- For the remote reading of consumption meters
- For up to 6 unit-loads (9mA)
- Power supply by Raspberry Pi or external DC power supply (9...30V)
- Galvanicaly isolated interface
- Removable screw terminals for bus and power supply connection
- Stacked header version available
- Indicator LEDs for RX and TX signals
- For Raspberry Pi 2 / 3 / 4 / 5 Modell B, Raspberry Pi Zero and compatible SBC

M-Bus HAT

for Raspberry Pi and compatible

Usage:



Compatibility :

Raspberry Pi B+, 2 B, 3 B, 3 B+

Raspberry Pi 4 B, Pi 5 B

Raspberry Pi A+, 3 A+

Raspberry Pi Zero (w) & Zero2

M-Bus HAT

for Raspberry Pi and compatible

Part number table:

Part-No.	EAN	Version
RPIHTMBS	0676424951411	With **stacked** header

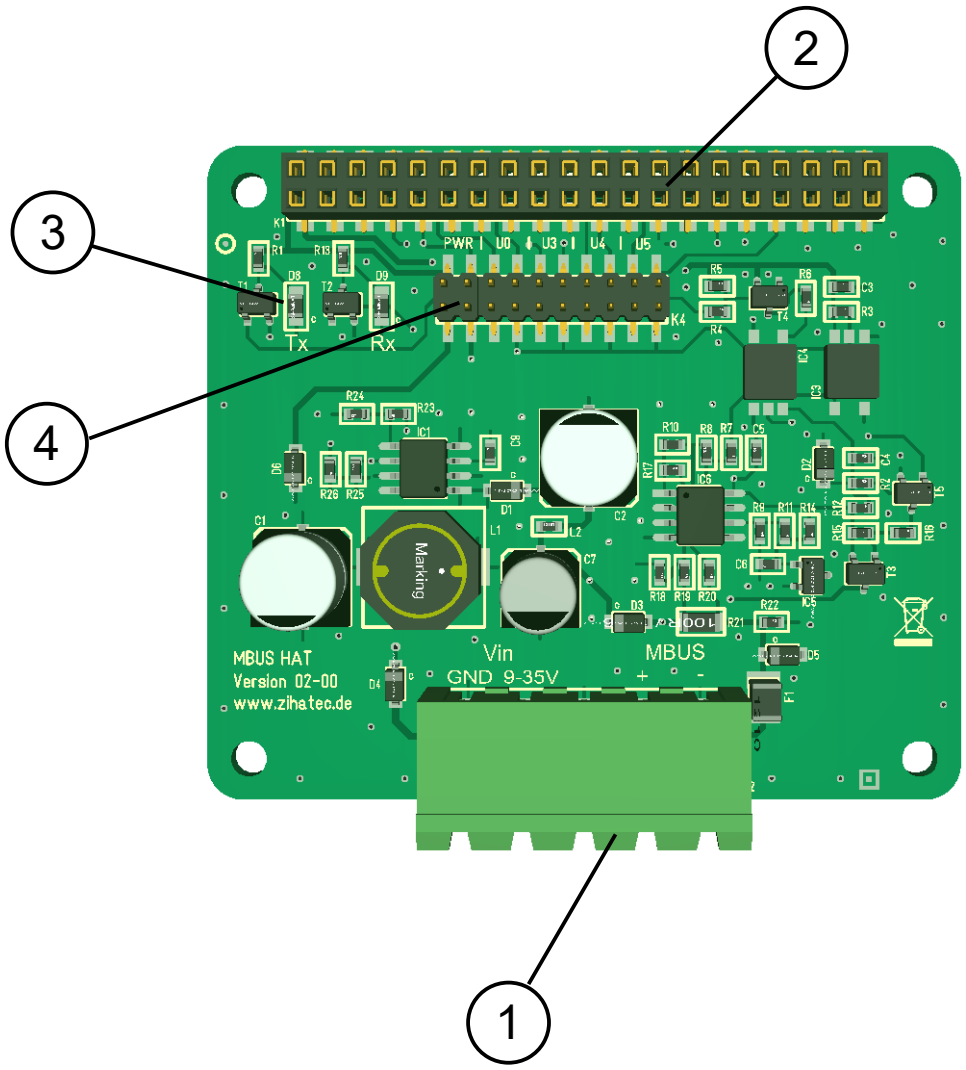
A version with standard header is no longer offered from version 02-0x for production reasons

Used Raspberry Pi Pins:

Depending on the selected UART via jumper K5 different pins are used:

Function	UART0	UART3	UART4	UART5
GND	PIN 6, 9, 14, 25, 39			
3,3V	PIN 1			
5V	PIN 2, 4			
TX	GPIO14 (8)	GPIO4 (7)	GPIO8 (24)	GPIO12 (32)
RX	GPIO15 (10)	GPIO5 (29)	GPIO9 (21)	GPIO13 (33)

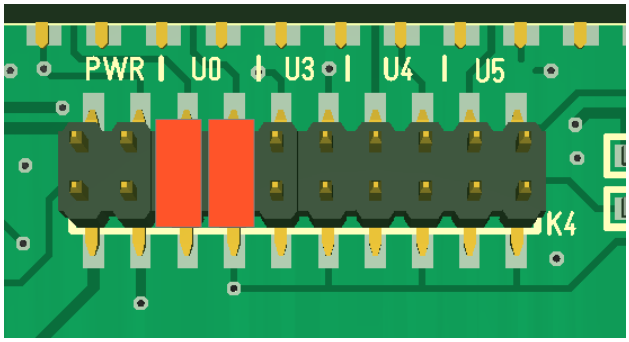
Control Elements:



① Power and M-Bus terminal
② header for Raspberry Pi (on backside)
③ Indicator LEDs
④ Jumper K4 for PWR and UART configuration

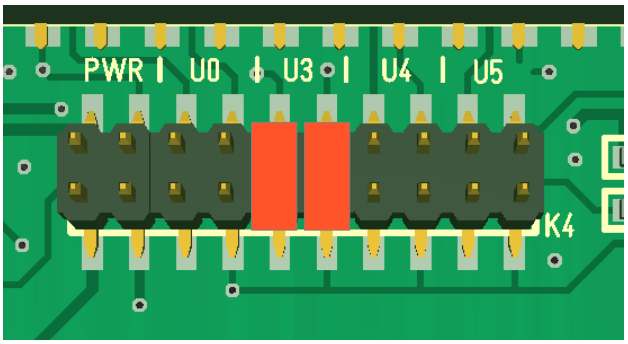
UART configuration via Jumper K4:

When using a Raspberry Pi 4 or Pi5, other UARTs can be selected Alternatively via jumper K4 besides UART0:



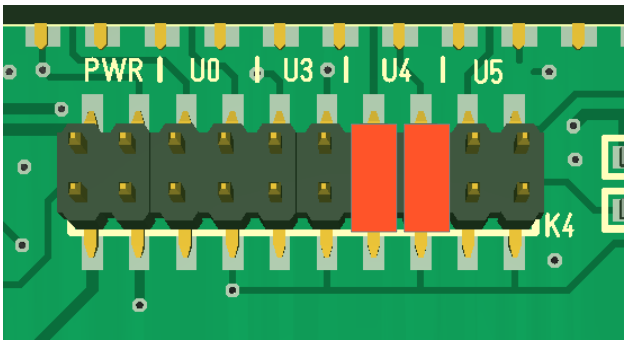
UART0 – default

(for all Raspberry Pi models, Not recommend for Pi5)



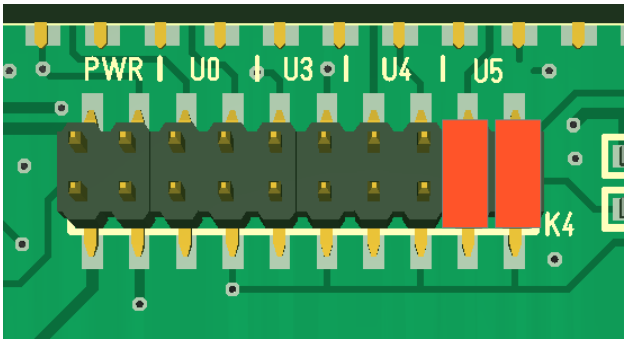
UART3 (Raspberry Pi 4 only)

UART2 (Raspberry Pi 5 only)



UART4 (Raspberry Pi 4 only)

UART3 (Raspberry Pi 5 only)



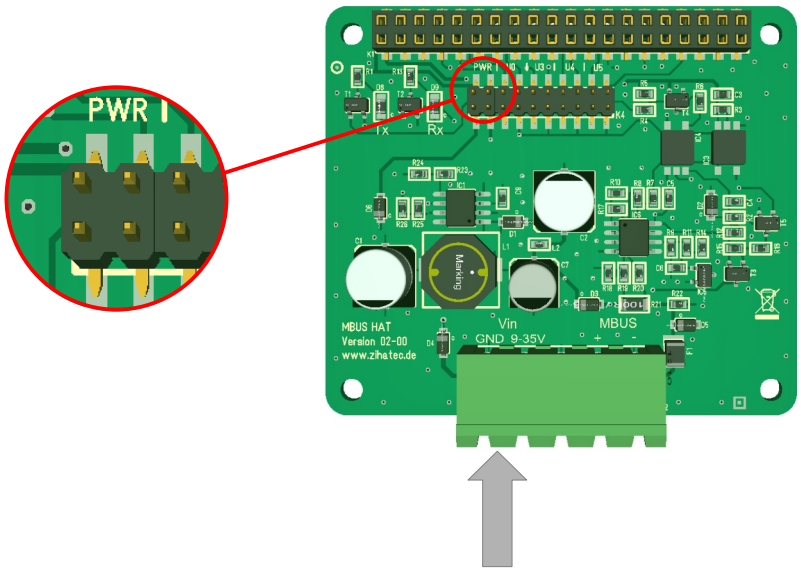
UART5 (Raspberry Pi 4 only)

UART4 (Raspberry Pi 5 only)

Power configuration via Jumper K4:

The voltage source for the MBUS supply can be selected via the two jumpers marked PWR on K4:

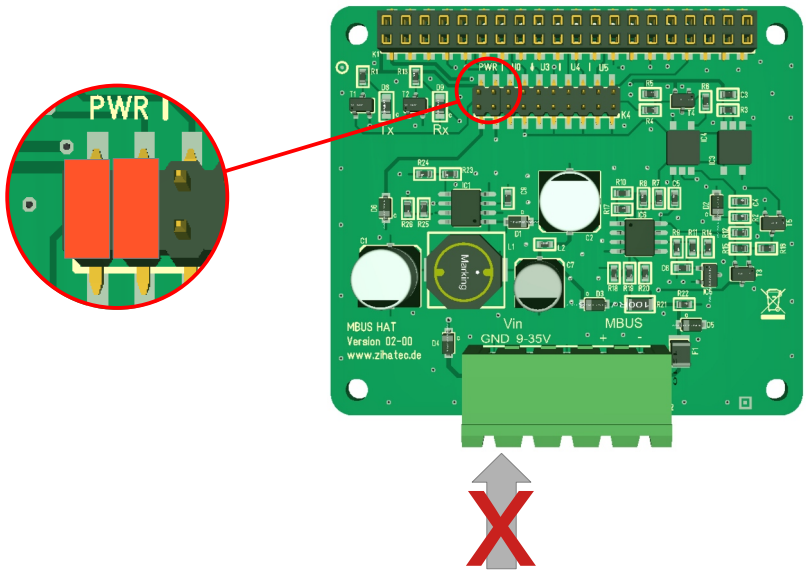
a) External power supply (recommended)



The two jumpers marked PWR on K4 are NOT plugged in.

Connect an external supply voltage of 9V to 30V DC to terminal K2.

b) Internal power supply from Raspberry Pi



Insert 2 jumpers at the position of K4 marked with PWR.

In this case, no external voltage source needs to be connected to K2.

Attention: the galvanic isolation between MBUS and Raspberry Pi is ineffective in this configuration.

UART(0) Configuration under Bullseye:

The easiest way is to use the raspi-config tool to enable the UART to the GPIO14/15 pins.

take a fresh Raspbian Bullseye image

```
sudo raspi-config
```

goto '3 Interfacing Options'

goto 'I6 Serial Port'

'Would you like a login shell to be accessible over serial?' --> NO

'Would you like the serial port hardware to be enabled?' --> YES

Finish raspi-config

```
sudo echo "dtoverlay=disable-bt" | sudo tee -a /boot/config.txt
```

```
sudo systemctl disable hciuart
```

reboot the Raspberry Pi

Now you can access the UART via **/dev/serial0**

UART Configuration under Bookworm (Raspberry Pi 4 only):

In our experience, libmbus does not run stable with UART0 (GPIO14,15) on Bookworm due to higher latency times. We therefore recommend using the remaining 3 UARTs.

UART	PCB Print	Access via
3	U3	/dev/ttyAMA3
4	U4	/dev/ttyAMA4
5	U5	/dev/ttyAMA5

To enable the corresponding UART, please proceed as follows:

take a fresh Raspbian image

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

For e.g. UART3 add the following line at the end of the file:

```
dtoverlay=uart3
```

If you want to enable more or another UART change this line to uart4 or uart 5 or add some additional lines.

reboot the Raspberry Pi!

UART Configuration under Bookworm (Raspberry Pi 5 only):

In our experience, libmbus does not run stable with UART0 (GPIO14,15) on Bookworm due to higher latency times. We therefore recommend using the remaining 3 UARTs.
Please note, however, that in contrast to the Pi4, the connections of the UARTs are interchanged:

UART	PCB Print	Access via
2	U**3**	/dev/ttyAMA**2**
3	U**4**	/dev/ttyAMA**3**
4	U**5**	/dev/ttyAMA**4**

To enable the corresponding UART, please proceed as follows:

take a fresh Raspbian image

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

For e.g. UART2 add the following line at the end of the file:

```
dtoverlay=uart2
```

If you want to enable more or another UART change this line to uart3 or uart 4 or add some additional lines.

reboot the Raspberry Pi!

Installation of libmbus (C++) library and utility:

Libmbus by Raditex Control is an open-source library for Linux.
For more information see http://www.rscada.se/libmbus

Installation:

```
sudo apt install git libtool autoconf cmake build-essential
```

```
sudo apt-get install -y cmake
```

```
sudo git clone https://github.com/rscada/libmbus.git
```

```
cd libmbus
```

```
sudo ./build.sh
```

```
sudo make install
```

```
cd bin
```

```
sudo ln -s /usr/local/lib/libmbus.so.0 /usr/lib/libmbus.so.0
```

Using libmbus (C++) library and utility:

Goto /libmbus/bin directory first:

```
cd libmbus/bin
```

Usage for UART0 (Bullseye):

```
./mbus-serial-scan -d -b 2400 /dev/serial0
```

(will list all connected M-Bus slave devices)

```
./mbus-serial-request-data -d -b 2400 /dev/serial0 10
```

(will read out the M-Bus device with address 10)

Usage for e.g. UART3 (Bookworm):

```
./mbus-serial-scan -d -b 2400 /dev/ttyAMA3
```

(will list all connected M-Bus slave devices)

```
./mbus-serial-request-data -d -b 2400 /dev/ttyAMA3 10
```

(will read out the M-Bus device with address 10)

Using pyMeterBus (Python) library and utility:

PyMeterBus is a pure Python implementation of the Meter-Bus by Mikael Ganehag Brorsson. For more information see https://gitlab.com/ganehag/pyMeterBus

Installation under Bullseye:

```
pip3 install pyMeterBus
```

```
sudo git clone https://gitlab.com/ganehag/pyMeterBus
```

```
cd pyMeterBus/tools
```

Usage for UART0 (Bullseye):

```
python3 mbus-serial-scan.py -d /dev/serial0
```

(will list all connected M-Bus slave devices)

```
python3 mbus-serial-request-data.py -d -a 10 /dev/serial0
```

(will read out the M-Bus device with address 10)

Installation under Bookworm:

```
python3 -m venv mbus
```

```
source mbus/bin/activate
```

```
cd mbus
```

```
pip install pyMeterBus
```

```
sudo git clone https://gitlab.com/ganehag/pyMeterBus
```

```
cd pyMeterBus/tools
```

Usage for UART3 (Bookworm):

```
python3 mbus-serial-scan.py -d /dev/ttyAMA3
```

(will list all connected M-Bus slave devices)

```
python3 mbus-serial-request-data.py -d -a 10 /dev/ttyAMA3
```

(will read out the M-Bus device with address 10)