M-Bus HAT

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)
- External DC power supply (9...30V) needed
- Galvanically 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 Modell B, Raspberry Pi Zero and compatible SBC
M-Bus HAT

for Raspberry Pi and compatible

Usage:

![Diagram of M-Bus HAT usage with Raspberry Pi models]

Compatibility:

- Raspberry Pi B+, 2 B, 3 B, 3 B+
- Raspberry Pi 4 B
- Raspberry Pi A+, 3 A+
- Raspberry Pi Zero (w) & Zero2
M-Bus HAT
for Raspberry Pi and compatible

Part number table:

<table>
<thead>
<tr>
<th>Part-No.</th>
<th>EAN</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPIHTMB</td>
<td>0676424951404</td>
<td>With <strong>standard</strong> header</td>
</tr>
<tr>
<td>RPIHTMBS</td>
<td>0676424951411</td>
<td>With <strong>stacked</strong> header</td>
</tr>
</tbody>
</table>

Used Raspberry Pi Pins:

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

<table>
<thead>
<tr>
<th>Function</th>
<th>UART0</th>
<th>UART3</th>
<th>UART4</th>
<th>UART5</th>
</tr>
</thead>
<tbody>
<tr>
<td>GND</td>
<td></td>
<td>PIN 6, 9, 14, 25, 39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.3V</td>
<td></td>
<td></td>
<td></td>
<td>PIN 1</td>
</tr>
<tr>
<td>5V</td>
<td></td>
<td></td>
<td>PIN 2, 4</td>
<td></td>
</tr>
<tr>
<td>TX</td>
<td>GPIO14 (8)</td>
<td>GPIO4 (7)</td>
<td>GPIO8 (24)</td>
<td>GPIO12 (32)</td>
</tr>
<tr>
<td>RX</td>
<td>GPIO15 (10)</td>
<td>GPIO5 (29)</td>
<td>GPIO9 (21)</td>
<td>GPIO13 (33)</td>
</tr>
</tbody>
</table>
Control Elements:

1. Power and M-Bus terminal
2. Headers for Raspberry Pi (on backside)
3. Indicator LEDs
4. Jumper K5 for UART configuration
UART configuration via Jumper K5:

When using a Raspberry Pi 4, other UARTs can be selected alternatively via jumper K5 besides UART0:

- **UART0 – default**
  (for all Raspberry Pi models)

- **UART3**
  (Raspberry Pi 4 only)

- **UART4**
  (Raspberry Pi 4 only)

- **UART5**
  (Raspberry Pi 4 only)
UART(0) Configuration (all Raspberry Pi models):

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

take a fresh Raspbian 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`
UART3-5 Configuration (for Raspberry Pi 4 only):

The new Pi-4 features additional hardware UART ports. Our M-BUS HAT can use the additional UARTS 3 – 5. You can enable the additional UARTs by editing the /boot/config.txt file:

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

For UART3 add the following line at the end of the file:

```
dtovemlay=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!

```
ls /dev/ttyAMA*
```

You should see in minimum /dev/ttyAMA0. This is the first enabled UART. If you have more UARTs enabled, you will see more additional entries /dev/ttyAMA1 etc.

Now you can access the UART3 via `/dev/ttyAMA0`
Using 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:

```bash
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
```

Usage for UART0:

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

(will list all connected M-Bus slave devices)

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

```bash
pip3 install pyMeterBus

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

cd pyMeterBus/tools
```

Usage for UART0:

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

(Will list all connected M-Bus slave devices)

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

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