



# M-Bus Master Hat Datasheet

Issue Date 1 November 2020  
 Issue Number 1.7d.2  
 Board Revision 1.7d+  
 Page 1/3

## M-Bus Support

M-Bus Specification Supported	EN 13757-2 (Wired M-Bus, Physical Layer)
Max Unit Loads (ULs) Supported	3
Max Slaves Supported	3 (See note 1)
Max Current Supplied on M-Bus	4.5mA (steady state) 25mA (peak, during mark from slave)
M-Bus Voltage supplied	26-28V (27.5V typical)
Max Bus Length	100m
Supported M-Bus Baud Rates	300, 600, 1200, 2400, 4800, 9600
Optical Isolation between Host and M-Bus	Yes

## Raspberry Pi Support

### Models Supported

Raspberry Pi 1 Model A+	Raspberry Pi 3 Model A+	Raspberry Pi 4 Model B
Raspberry Pi 1 Model B+	Raspberry Pi 3 Model B	Raspberry Pi Zero
Raspberry Pi 2 Model B	Raspberry Pi 3 Model B+	Raspberry Pi Zero W(H)

### GPIOs Used

Physical Pin Number	Broadcom Pin Number	wiringPi Pin Number	Purpose
1	-	-	3.3V – Logic
2	-	-	5V – M-Bus Power
4	n/a	n/a	5V – M-Bus Power
8	14	15	Serial TX – M-Bus Transmit
9	-	-	Ground
10	15	16	Serial RX – M-Bus Receive
27	0	30	ID_SD – Read Hat EEPROM
28	1	31	ID_SC – Read Hat EEPROM
37	26	25	GPIO – Enable/Disable M-Bus power
39	-	-	Ground

### Pi HAT Support (See Note 2)

Pi HAT Mechanical Specification Compliant	Yes
ID EEPROM support for HAT auto-detect and auto-configuration	Yes
Full-sized 40W GPIO Connector	Yes
Spacer height from Pi to HAT	11mm
Pi back-power support	No
Pi camera cable cut-out	No

## Physical Dimensions

Full-sized	65mm x 56mm ( $\pm 2\%$ )
Micro	65mm x 30mm ( $\pm 2\%$ )
Mounting holes	2.6mm ( $\pm 3\%$ ) M2.5
PCB thickness	1.7mm ( $\pm 5\%$ )



# M-Bus Master Hat Datasheet

Issue Date 1 November 2020  
Issue Number 1.7d.2  
Board Revision 1.7d+  
Page 2/3

## Theory of Operation

### HAT

The M-Bus Master Hat supports the Raspberry Pi HAT Specification (See Note 2).

A Raspberry Pi running a recent version of Raspbian will read the M-Bus Master Hat's EEPROM at boot time and correctly configure the necessary GPIOs:

- Serial TX and RX pins are configured to use the PL011 UART, using device `/dev/ttyAMA0` (disabling the default use of `/dev/ttyAMA0` for Bluetooth).
- GPIO26 (pin 37, wiringPi GPIO 25) which controls the M-Bus power, is set to output mode, low state, disabling the M-Bus power by default.

The M-Bus Master Hat EEPROM configuration can be examined by inspecting the contents of the following files:

File	Sample Contents
<code>/proc/device-tree/hat/product</code>	M-Bus Master
<code>/proc/device-tree/hat/product_id</code>	0x0001
<code>/proc/device-tree/hat/product_ver</code>	0x0005
<code>/proc/device-tree/hat/uuid</code>	e1f1239d-4567-0123-abcd-ff86f7ec1234
<code>/proc/device-tree/hat/vendor</code>	packom.net

### M-Bus Power

M-Bus power is created using a standard boost converter circuit, based around the TI TPS61040/TPS61041 ICs. A peak M-Bus voltage of 28V can be supplied, with a typical value being around 27.5V.

While the M-Bus specification calls for a nominal voltage of 36V, voltages down to 21-24V are required by the specification to be supported. As the M-Bus Master Hat supports a bus length of up to 100m and 4.5mA maximum steady state current draw, the voltage drop over the bus will be negligible, and the voltage within the required range.

M-Bus power is enabled by driving GPIO 26 (pin 37, wiringPi GPIO 25) high. The red M-Bus Power LED illuminates when the bus is powered.

### M-Bus Communication

The M-Bus Master Hat converts between serial communication to/from the Raspberry Pi, and the M-Bus physical communication layer.

To transmit data on the M-Bus a space is transmitted as voltage drop of 12V on the M-Bus, whereas a mark is transmitted as no voltage drop on the M-Bus.

For receiving data from a slave connected to the M-Bus, an additional current draw of 11-20mA on the M-Bus is interpreted as a space, and no additional current draw as a mark.

The M-Bus Master Hat uses optocouplers to isolate communication between the Pi and the M-Bus.

### Serial Settings

The appropriate serial port must be used for communicating on the M-Bus (`/dev/ttyAMA0`, connected to the PL011 UART). The PL011 rather than the mini-UART must be used as the mini-UART does not support parity modes.

The speed must be a supported M-Bus baud rate, and a rate supported by the M-Bus Master Hat. Serial settings must be 8-bit, even parity, 1 start bit, 1 stop bit.



# M-Bus Master Hat Datasheet

Issue Date 1 November 2020  
Issue Number 1.7d.2  
Board Revision 1.7d+  
Page 3/3

## Compatible Open Source Software (OSS) (See Note 3)

Project	Link	Details
libmbus	<a href="https://github.com/rscada/libmbus">https://github.com/rscada/libmbus</a>	M-Bus Library from Raditex Control (See Note 4)
pyMeterBus	<a href="https://github.com/ganehag/pyMeterBus">https://github.com/ganehag/pyMeterBus</a>	Meter-Bus for Python (See Note 4)
pyMbusHat	<a href="https://github.com/packom/pyMbusHat">https://github.com/packom/pyMbusHat</a>	Sample Python app to drive M-Bus Master Hat, using pyMbusHat
mbus-httpd	<a href="https://github.com/packom/mbus-httpd">https://github.com/packom/mbus-httpd</a>	M-Bus HTTP Server supporting M-Bus Master Hat, using libmbus

## Warranty

The M-Bus Master Hat is warranted to be free of manufacturing or design defects for a period of 1 year from purchase.

## Attribution

The M-Bus Master Hat is based on an M-Bus Master design available as part of libmbus.

## Notes

1. Assuming each slave draws 1UL
2. For more information on Pi HAT support and specification see <https://github.com/raspberrypi/hats>
3. No software support is provided with the M-Bus Master Hat
4. The M-Bus Master Hat is not endorsed by the creators of libmbus or PyMeterBus