RS422/RS485 HAT

Application Note: How to use Modbus

1 Introduction

The RS422/RS485 HAT for the Raspberry Pi can be used for Modbus systems. In Modbus RTU and Modbus ASCII RS485 is used as the physical layer.
Our RS422 / RS485 Serial HAT is a fully galvanic isolated serial communication HAT designed for use with the Raspberry Pi and the perfect choice for such kind of applications. The objective of this document is to show how to configure the HAT and the Raspberry Pi to be used for Modbus serial communication.

2 Bill of Material

- Raspberry Pi B+, B2 or B3
- RS422/RS485 serial HAT (available in our webstore)

3 Wiring

The wiring is very simple. You have to connect only the A and B terminals of the HAT with the A and B line of the Modbus system. Y and Z terminals are not used for this kind of application.
For long distances it is recommend to use twisted pairs for A and B.

5 DIP Switch Settings

The transmitting/receiving direction can be switched automaticaly:

SW1	
1	OFF
2	ON
3	ON
4	OFF

SW2	
1	OFF
2	OFF
3	ON
4	ON

Depending of the position of the RS422/RS485 HAT in the Modbus line you have to switch the terminating resistor ON or OFF. Please switch the resistor to ON position only if the HAT is on one the end of the bus line. In all other cases switch the terminating resistor OFF:

SW3	
1	ON/OFF
2	OFF
3	OFF
4	OFF

6 Free up Serial Line and Enable UART

6.1 Raspberry Pi 1 and 2

On the Raspberry Pi 1 and 2 ttyAMA0 might be used by the serial console. To free it up do the following. Remove any references to *ttyAMA0* from */etc/inittab* and */boot/cmdline.txt*.

Our /boot/cmdline.txt looks like this:

```
dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait
```

Please follow this link for more information:
http://elinux.org/RPi_Serial_Connection#Preventing_Linux_using_the_serial_port

6.2 Raspberry Pi 3

First we have to config the Raspberry Pi:

```
sudo raspi-config
```

```
Goto -> 5 Interfacing Options -> P6 Serial -> choose YES
```

Additionally remove any references to `console=serial0,115200` and `/dev/ttyAMA0` from */boot/cmdline.txt.*

Our file looks like this:

```
dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait
```

Update the Raspbian Jessie operating system:

```
sudo apt-get update
sudo apt-get upgrade
```

On the Raspberry Pi 3 /dev/ttyAMA0 is used by the Wifi and Bluetooth module. There is a “mini UART” available on /dev/ttyS0 by default. It is better though, to use the hardware UART and switch the Wifi/Bluetooth module to mini UART. To do that, add this line at the end of */boot/cmdline.txt*

```
dtoverlay=pi3-miniuart-bt
enable_uart=1
force_turbo=1
```

Optional: Disable the serial interface in Raspbian Jessie:

```
systemctl disable serial-getty@ttyAMA0.service
```

Reboot your Raspberry Pi.

6.3 UART test

You can easily test the function of the internal UART with Minicom or any other terminal program without a Modbus system.

If the UART /dev/ttyAMA0 is free and connected to the GPIO14/15 pins of the 2x20 header of the Raspberry Pi then the TX Led of the HAT blinks when you send data from the terminal.

7 Software for Modbus

You will find an easy-to-use and tested Modbus RTU and Modbus ASCII implementation for Python on:

MinimalModbus