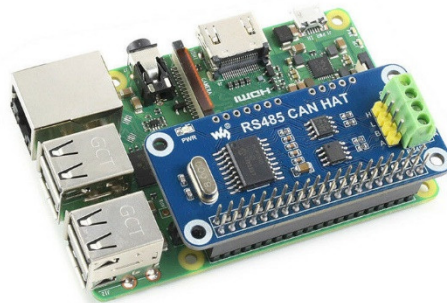


Interfacing TF03-485 Single Point LiDAR with Raspberry Pi HAT



Written by: Ibrahim (FAE)

www.benewake.com
Benewake (Beijing) Co., Ltd.

This article explains how to interface Benewake TF03-485 single point LiDAR with Raspberry Pi 485-CAN HAT. The HAT is compatible with Raspberry Pi Zero/Zero W/ZeroWH/2B/3B/3B+/4B. The following assumptions are made while writing this article:

1. The end-user has Raspberry Pi single board computer and 485-CAN HAT¹ (manufactured by Waveshare). I have added four different links from where 485-CAN HAT can be purchased
2. Benewake TF03-485 single point LiDAR in hands
3. Raspbian operating system. At the time of writing this article the version *2021-03-04-raspbian-buster-armhf*² was tested
4. All the necessary cables and 5V power supply. TF03-485 industrial version supports wide voltage range 5-24V
5. The default interface mode of LiDAR is 485 so we don't need any prior settings
6. Any editor for writing and modifying the script (VS Code is recommended)
7. **UART port is enabled:**
 - I. Start raspi-config(in terminal type): `sudo raspi-config`.
 - II. Select option 3 - Interface Options.
 - III. Select option P6 - Serial Port.
 - IV. At the prompt "Would you like a login shell to be accessible over serial"? answer is 'No'
 - V. At the prompt "Would you like the serial port hardware to be enabled"? answer is 'Yes'
 - VI. Exit **raspi-config** and reboot the Pi for changes to take effect.

¹ <https://www.waveshare.com/product/raspberry-pi/rs485-can-hat.htm>
<https://www.ebay.com/itm/263753270299>
<https://www.cytron.io/p-rs485-can-hat-for-raspberry-pi>
<https://botland.store/raspberry-pi-hat-connection/12532-rs485-can-hat-overlay-for-raspberry-pi-waveshare-14882-5904422319502.html>

² https://downloads.raspberrypi.org/raspbian_armhf/images/raspbian_armhf-2021-03-25/2021-03-04-raspbian-buster-armhf.zip

Pin configuration of 485-CAN HAT:

We are only interested in two pins of HAT which are **A** and **B** as only these two pins are needed for 485 communication. The HAT can directly be inserted into 40-pin header of Raspberry Pi. In this tutorial we are not discussing other pins. The following image shows 485 pins marked in red:



Figure 1: Pinouts

Schematic diagram:

The connection diagram for interfacing TF03 LiDAR to HAT and using a separate power supply is given below:

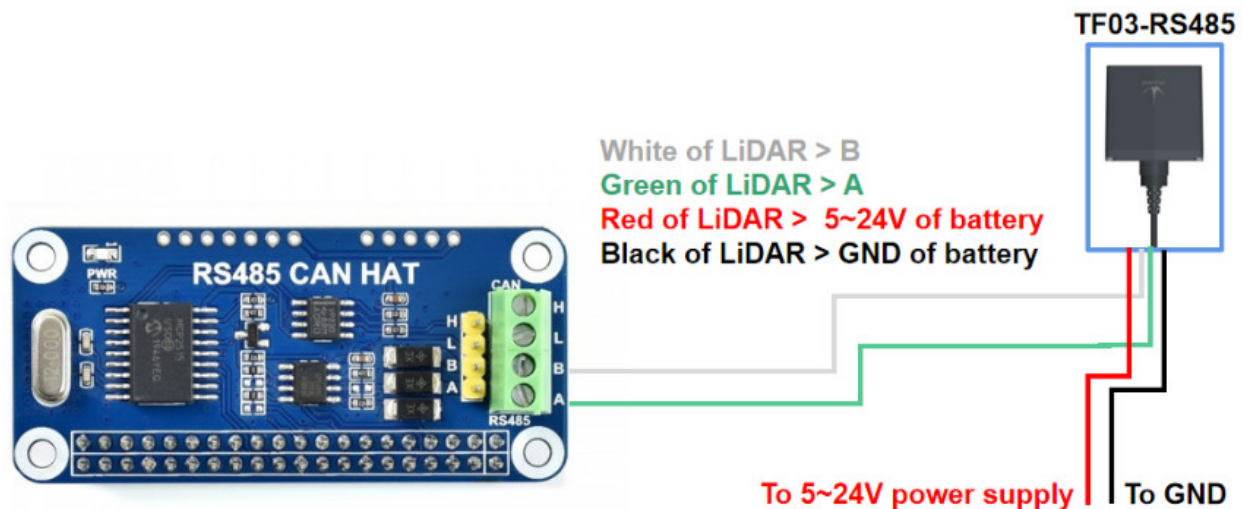


Figure 2: Connection Diagram

For exact details about LiDAR current rating and your supply capacity, please refer to their respective data-sheets. It should be noted that if you are using the same power supply for LiDAR and Raspberry Pi + HAT, then your supply should be able to provide enough current to all these devices otherwise some parts may behave abnormally.

Default 485 Communication Parameters of TF03:

According to the data-sheet of TF03 LiDAR the parameters of 485 communication:

Item	Parameter
Communication interface	RS485/RS232
Baud rate	115200
Data bit	8
Stop bit	1
Parity	None

Figure 3: 485 Communication

So 485 communication in code script is established based on this information. The baud rate and other parameters must match on two sides (HAT and LiDAR) otherwise communication cannot be established.

Cable and Connector Selection:

All Benewake Single Points LiDARs have either 4-pin or 7pin (for TF03) Molex connector, so the required mating connector is JST socket 1.25mm. Please refer to the link³. You can choose either 4-pin or 7-pin depending upon the LiDAR that you have. On the HAT side will need male-DuPont connector or any other suitable male connector that can be inserted into HAT:

³ <https://www.aliexpress.com/i/4000580232835.html>



Figure 4: Male DuPont cable

Table III: Data format (for TF03 Series)

Data bit	Definition	Description
Byte0	Frame header	0x59
Byte1	Frame header	0x59
Byte2	DIST_L	DIST low 8-bits
Byte3	DIST_H	DIST high 8-bits
Byte4	Strength_L	Signal strength low 8-bits
Byte5	Strength_H	Signal strength high 8-bits
Byte6	Reserved bit	/
Byte7	Reserved bit	/
Byte8	Checksum	Low 8 bits of Checksum bit. Checksum = Byte0 + Byte2+...+Byte7

Setting Development Environment:

1. [Installing BCM2835]: in terminal type the following commands step by step:

```
$ wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.60.tar.gz
$ tar zxvf bcm2835-1.60.tar.gz
$ cd bcm2835-1.60/
$ sudo ./configure
$ sudo make && sudo make check && sudo make install
```



2. [Installing Python libraries]: Type the following commands.

```
$ sudo pip3 install pillow
$ sudo pip3 install numpy
$ sudo apt-get install libopenjp2-7
$ sudo apt install libtiff
$ sudo apt install libtiff5
$ sudo apt-get install libatlas-base-dev
```

I. [For Python2]:

```
$ sudo apt-get update
$ sudo apt-get install python-pip
$ sudo pip install RPi.GPIO
$ sudo pip install smbus
```

II. [For Python3]:

```
$ sudo apt-get update
$ sudo apt-get install python3-pip
$ sudo pip3 install RPi.GPIO
$ sudo pip3 install smbus
```

3. [Installing WiringPi Library]:

```
$ sudo apt-get install wiringpi
$ wget https://project-downloads.drogon.net/wiringpi-latest.deb
$ sudo dpkg -i wiringpi-latest.deb
$ gpio -v
```

Note: The 2.52 version will appear (in my case it appeared). If it does not appear, it means there is an installation error

4. [Installing Serial Library]:

```
$ sudo apt-get install python-serial
```

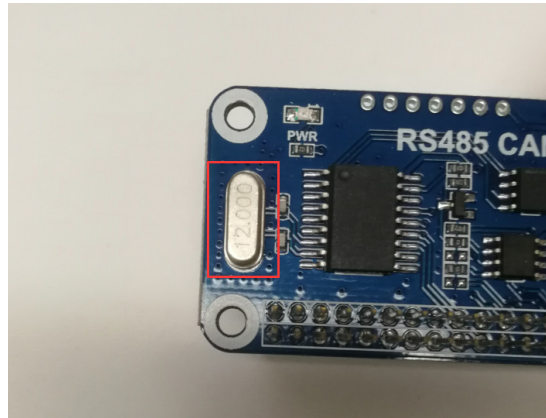
5. [Configure Raspberry Pi system files]:

```
$ sudo nano /boot/config.txt (any other text editor can be used like gedit, vim
etc. instead of nano)
```

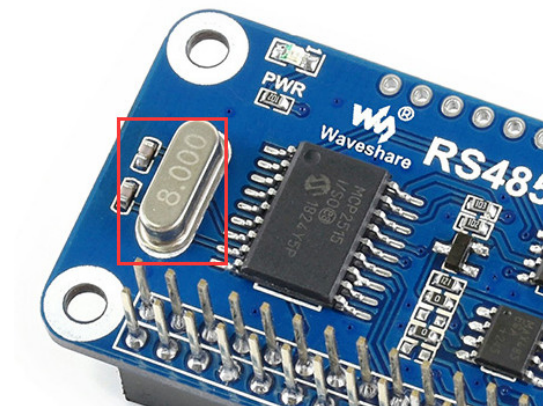
Enter the following contents at the end of config file:

```
dtoverlay=spi=on
dtoverlay=mcp2515-
can0,oscillator=12000000,interrupt=25,spimaxfrequency=2000000
```

Among them, **oscillator=12000000**, is the on-board crystal oscillator size of 12M, as shown in the figure below (my board oscillator has also 12M value):



Older version of HAT has different oscillator installed:



So the settings will be as follows:

```
dtparam=spi=on
dtoverlay=mcp2515-can0,oscillator=8000000,interrupt=25,spi_maxfrequency=1000000
```

6. After saving and exiting, restart the Raspberry Pi: `sudo reboot`
7. Run the command to check whether the initialization is successful: `dmesg | grep -i '\(can\|spi\)'`

If everything goes successful then it should print the following:

```
pi@raspberrypi:~/485-CAN-HAT/485/python $ dmesg | grep -i '\(can\|spi\)'
[ 6.049175] CAN device driver interface
[ 6.097586] mcp251x spi0.0 can0: MCP2515 successfully initialized.
```

If the module is not connected, it may prompt as follows: Please check whether the module is connected. Whether to enable SPI and enable MCP2515 kernel driver. Whether to restart.

```
pi@raspberrypi:~$ dmesg | grep -i '\\(can\\|spi\\)'\n[ 16.300731] systemd[1]: Cannot add dependency job for unit regenerate_ssh_host_keys.service, ignoring: Unit regenerate_ssh_host_keys.service failed to load: No such file or directory.\n[ 16.499602] systemd[1]: Cannot add dependency job for unit display-manager.service, ignoring: Unit display-manager.service failed to load: No such file or directory.\n[ 20.661718] CAN device driver interface\n[ 20.680261] mcp251x spi0.0: Cannot initialize MCP2515. Wrong wiring?\n[ 20.680293] mcp251x spi0.0: Probe failed, err=19
```

8. [Downloading example code⁴ and grant permission]:

```
$ sudo chmod 777 -R Python-code-folder/
```

If everything is successful and there is no issue with connection, and system settings etc. then you should be able to run the code and see the data being printed to the terminal.

For Python2:

```
$ sudo python TF03-485.py
```

```
pi@raspberrypi:~/485-CAN-HAT/485/python$ python TF03-485.py\n[INFO] Serial port: /dev/ttyS0\n[INFO] You will continuously receive data, press Ctrl + C to exit\n[INFO] This code works with Python2 and Python3\n\nPrinting python2 portion\nDistance:88\nStrength:293\n\nPrinting python2 portion\nDistance:88\nStrength:292\n\nPrinting python2 portion\nDistance:88\nStrength:291\n\nPrinting python2 portion\nDistance:88\nStrength:292\n\nPrinting python2 portion\nDistance:88\nStrength:292
```

⁴ <https://github.com/ibrahimgazi/TF03-485-Python-code-for-485-CAN-HAT>



For Python3:

```
$ sudo python3 TF03-485.py
```

```
pi@raspberrypi:~/485-CAN-HAT/485/python $ python3 TF03-485.py
[INFO] Serial port: /dev/ttyS0
[INFO] You will continuously receive data, press Ctrl + C to exit
[INFO] This code works with Python2 and Python3

Printing python3 portion
Distance:88
Strength:291

Printing python3 portion
Distance:88
Strength:292

Printing python3 portion
Distance:88
Strength:293

Printing python3 portion
Distance:88
Strength:292
```