



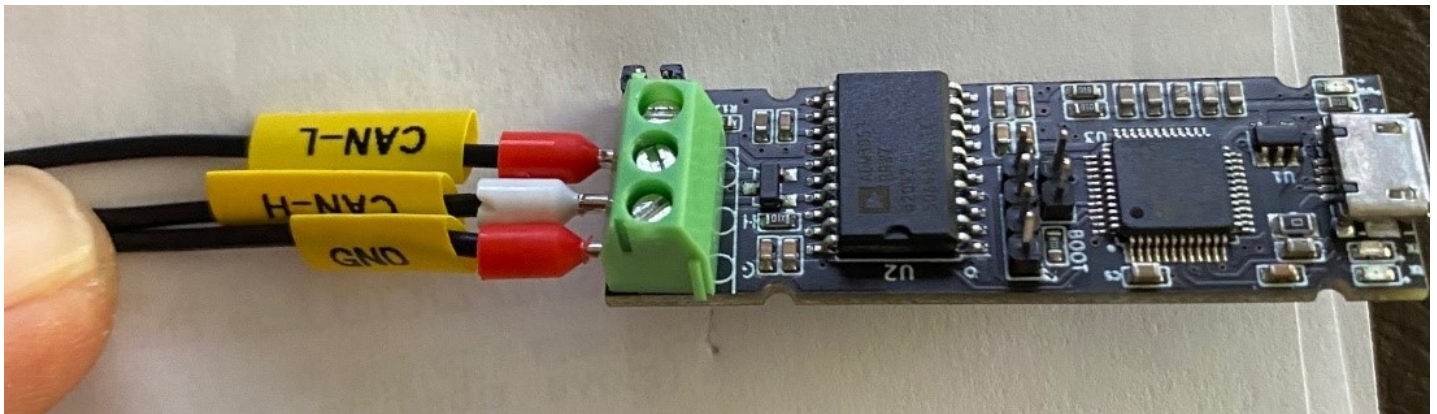
## CANBUS KIT INSTRUCTIONS v1.0

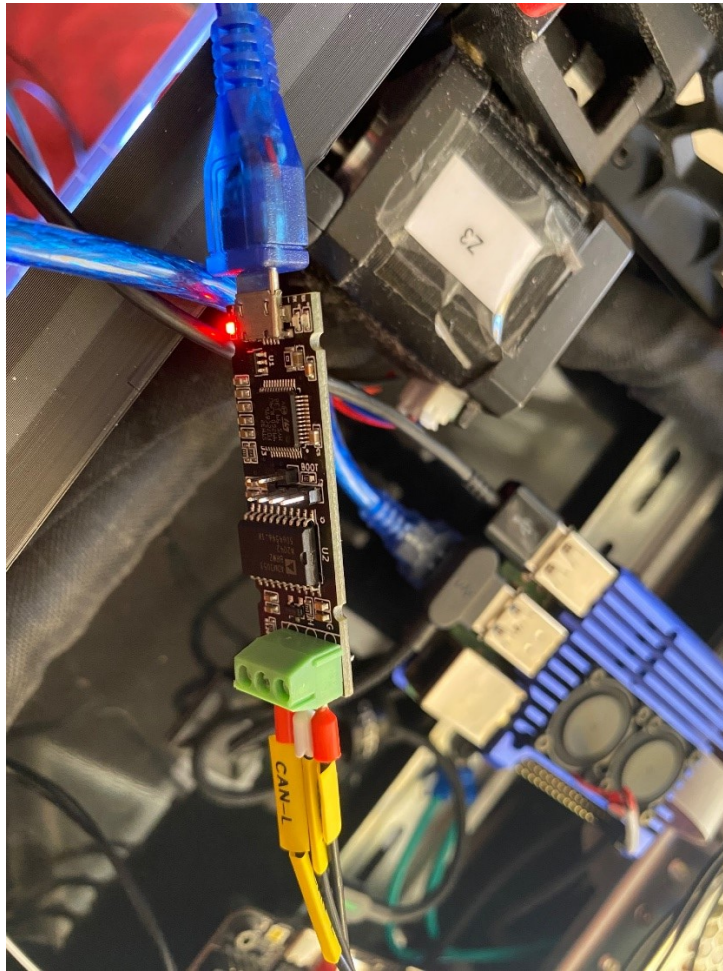
### DISCLAIMER:

There's no CAN BUS definitive guide available out there. **CAN BUS configuration requires some experience and knowledge to get it working.** Purpose of this guide is to help people in some basic steps of can bus configuration. We are not responsible if the following steps are different from your hardware and software configurations and we cannot provide support for your printer configuration or troubleshooting.

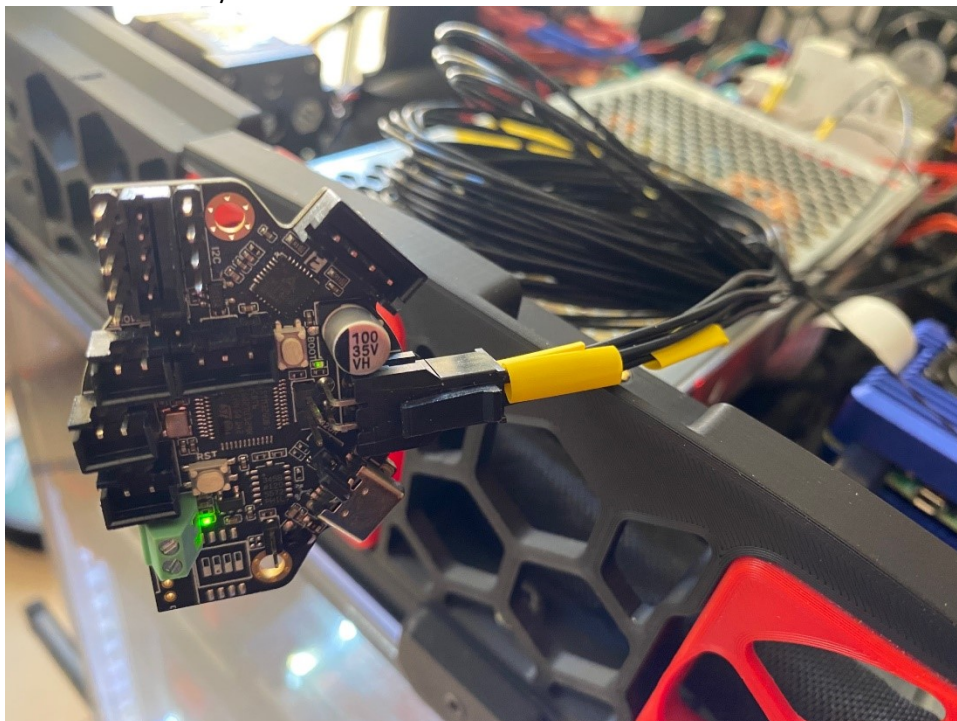
### CONNECTIONS

Connect CAN-L, CAN-H and GND ferrules terminal to Canable Pro. Also connect USB cable to Raspberry Pi:



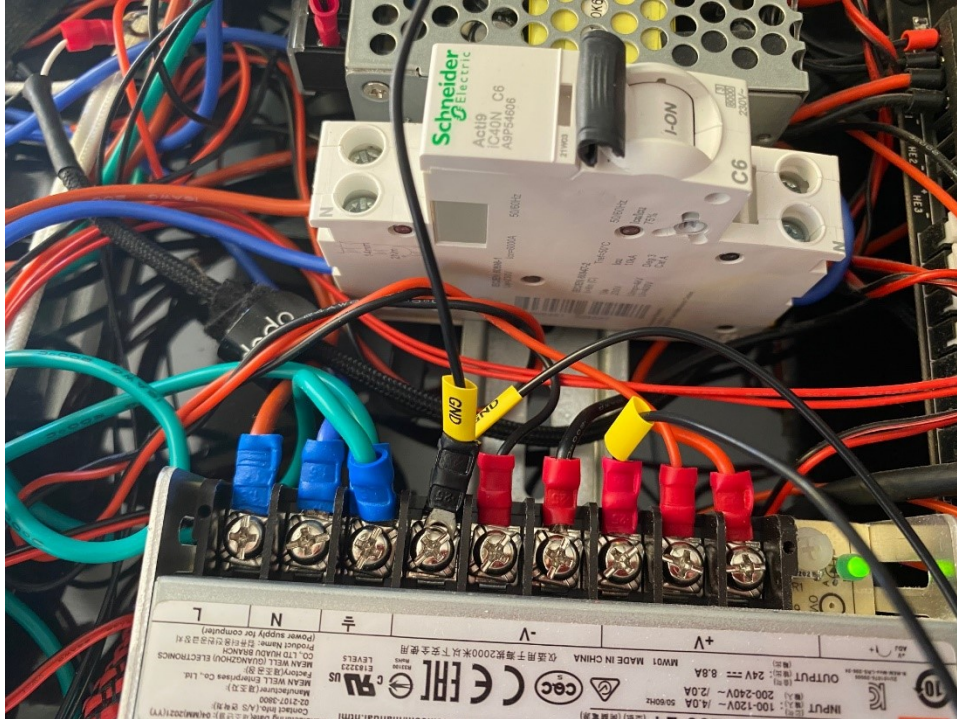


Connect Molex 4pin connector to EBB36/ EBB 42:



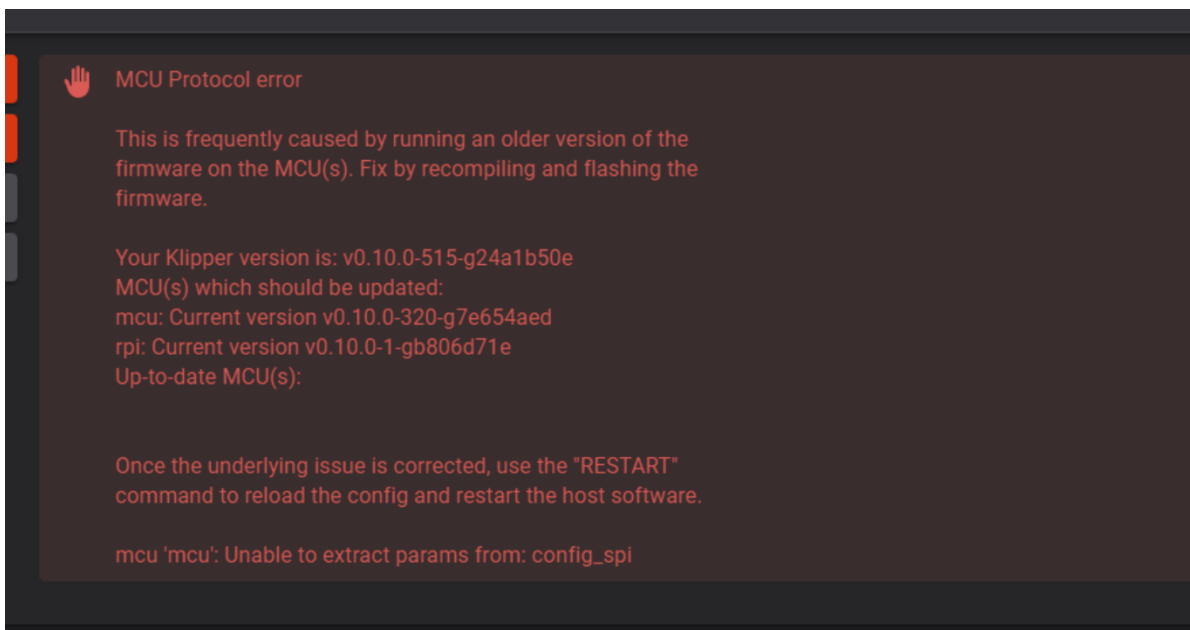


Connect the **2 GND** and the **VCC (24V)** fork connectors to GND and VCC on your PSU:



## FIRMWARE

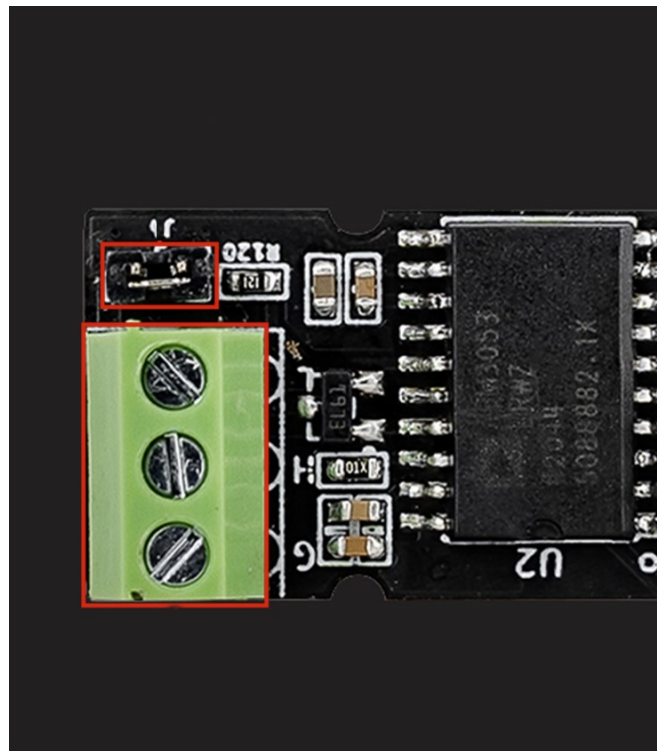
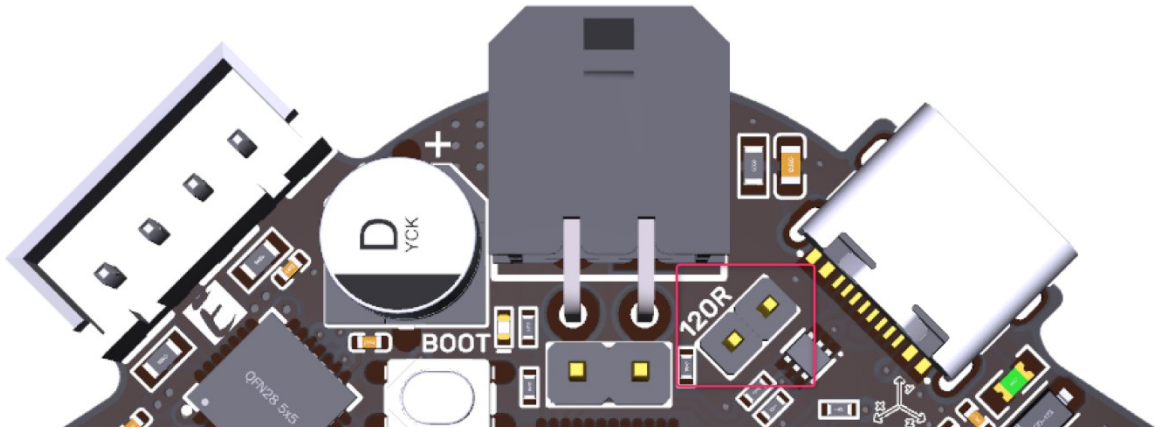
Boards are already flashed with latest versions of CANBOOT, Klipper and Candlelight firmwares; however Klipper should be updated on your MCU to same version of your CAN BOARD, otherwise CAN BUS will not work:



- *KLIPPER VERSION installed on your EBB36 CAN Board:* v0.10.0-515
- *Canbus\_UUID:* 028c2b2b0b37

## JUMPERS

Jumper caps on both EBB36/42 and Canable Pro should be on the 120R position for normal canbus operation:



## CONFIGS

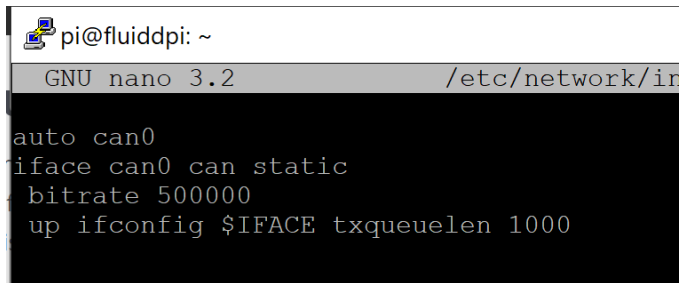
### On RASPBERRY Pi:

***sudo apt-get install can-utils***

***sudo nano /etc/network/interfaces.d/can0***

Insert these lines and save:

```
auto can0
iface can0 can static
    bitrate 500000
    up ifconfig $IFACE txqueuelen 1000
```



```
pi@fluiddpi: ~
GNU nano 3.2 /etc/network/ir
auto can0
iface can0 can static
    bitrate 500000
    up ifconfig $IFACE txqueuelen 1000
```

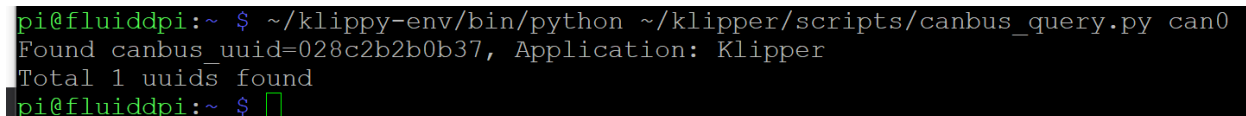
***sudo reboot***

***Sudo ifconfig can0***

***Interface can0 should not work at this stage.***

To find the UUID of your CAN bus controller (**UUID is shown ONLY when CAN BUS is not initialized**):

***~/klippy-env/bin/python ~/klipper/scripts/canbus\_query.py can0***



```
pi@fluiddpi:~ $ ~/klippy-env/bin/python ~/klipper/scripts/canbus_query.py can0
Found canbus_uuid=028c2b2b0b37, Application: Klipper
Total 1 uuids found
pi@fluiddpi:~ $
```

In your **printer.cfg** add line [**include sample-bigtreetech-ebb-canbus-v1.1.cfg**]

```

X printer.cfg

1 # This file contains common pin mappings for the BigTreeTech OctoPus V1.
2 # To use this config, the firmware should be compiled for the STM32F446 with a "32KiB bootloader"
3 # Enable "extra Low-Level configuration options" and select the "12MHz crystal" as clock reference
4
5 # after running "make", copy the generated "klipper/out/klipper.bin" file to a
6 # file named "firmware.bin" on an SD card and then restart the OctoPus with that SD card.
7
8 # See docs/Config_Reference.md for a description of parameters.
9
10 ## Voron Design VORON2 250/300/350mm BigTreeTech OctoPus V1 TMC2209 UART config
11
12 ## *** THINGS TO CHANGE/CHECK: ***
13 ## MCU paths [mcu] section
14 ## Thermistor types [extruder] and [heater_bed] sections - See 'sensor types' List at end of file
15 ## Z Endstop Switch location [safe_z_home] section
16 ## Homing end position [gcode_macro G32] section
17 ## Z Endstop Switch offset for Z0 [stepper_z] section
18 ## Probe points [quad_gantry_level] section
19 ## Min & Max gantry corner positions [quad_gantry_level] section
20 ## PID tune [extruder] and [heater_bed] sections
21 ## Fine tune E steps [extruder] section
22
23 [include /home/pi/klipper_config/nozzle_scrub.cfg]
24 [include print_area_bed_mesh.cfg]
25 [include Ellis_macros.cfg]
26 [include klicky-probe.cfg]
27 [include z_calibration.cfg]
28 [include stealthburner_leds.cfg]
29 [include sample-bigtreetech-ebb-canbus-v1.1.cfg]
30
31
32 [virtual_sdcard]
33 path: ~/gcode_files
34
35 [display_status]
36
37 [pause_resume]
38
39

```

Copy the file **sample-bigtreetech-ebb-canbus-v1.1.cfg** to your printer configs folder:



sample-bigtreetech-ebb-canbus-v1.1 (1).c

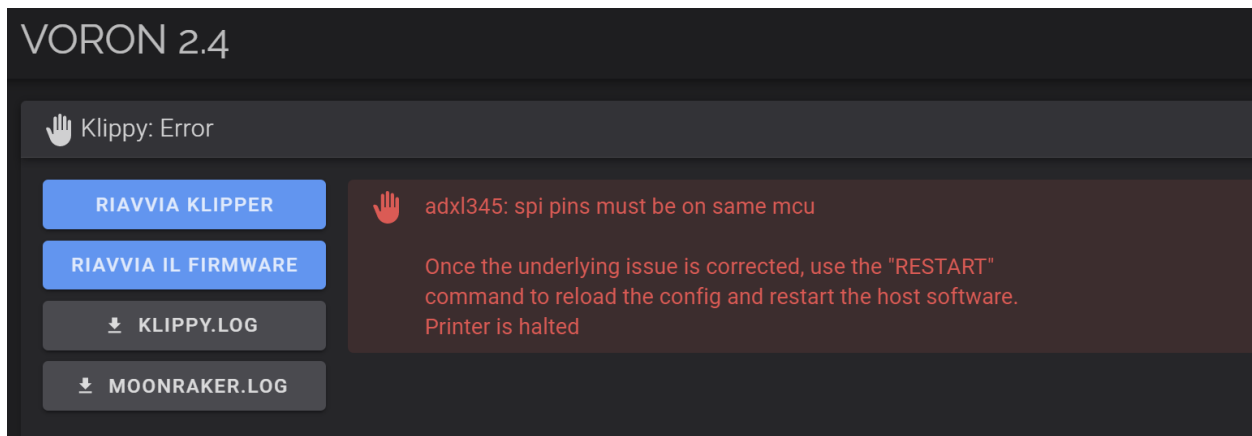
VORON 2.4			
{ } Files di Configurazione			
/			
Nome	Modificato ↓	Dimensione	
sample-bigtreetech-ebb-canbus-v1.1.cfg	Jul. 13, 2022 - 03:36 pm	1.5 kB	
printer.cfg	Jul. 13, 2022 - 02:59 pm	22.8 kB	
sample-bigtreetech-ebb-canbus-v1.2.cfg	Jun. 21, 2022 - 10:22 am	1.5 kB	
moonraker.conf.bkp	Jun. 21, 2022 - 09:00 am	1.0 kB	
moonraker.conf	Jun. 21, 2022 - 09:00 am	1.1 kB	
stealthburner_leds.cfg	Apr. 04, 2022 - 08:53 pm	7.5 kB	
Ellis_macros.cfg	Mar. 26, 2022 - 03:45 pm	34.9 kB	
printer-20220301_132359.cfg	Mar. 01, 2022 - 01:16 pm	22.8 kB	
printer-20220301_131622.cfg	Mar. 01, 2022 - 01:11 pm	22.8 kB	
printer-20220301_131123.cfg	Mar. 01, 2022 - 01:00 pm	22.8 kB	
printer-20220301_130056.cfg	Mar. 01, 2022 - 12:48 pm	22.8 kB	
printer-20220301_124821.cfg	Mar. 01, 2022 - 12:39 pm	22.8 kB	
printer-20220301_123926.cfg	Mar. 01, 2022 - 12:24 pm	22.8 kB	
printer-20220301_122427.cfg	Mar. 01, 2022 - 12:11 pm	22.8 kB	
printer-20220301_121146.cfg	Feb. 27, 2022 - 06:32 pm	22.8 kB	
z_calibration.cfg	Feb. 10, 2022 - 12:44 pm	4.4 kB	
printer-20220203_083135.cfg	Feb. 03, 2022 - 08:07 am	22.8 kB	
print_area_bed_mesh_Turge08.cfg	Jan. 25, 2022 - 03:14 pm	7.2 kB	

Check and replace **canbus\_uuid** number with your uuid:

```
× sample-bigtreetech-ebb-canbus-v1.1.cfg
1 # This file contains common pin mappings for the BIGTREETECH EBBCan
2 # Canbus board. To use this config, the firmware should be compiled for the
3 # STM32G0B1 with "8 MHz crystal" and "USB (on PA11/PA12)" or "CAN bus (on PB0/PB1)".
4 # The "EBB Can" micro-controller will be used to control the components on the nozzle.
5
6 # See docs/Config_Reference.md for a description of parameters.
7
8 View 'mcu' documentation
9 [mcu EBBCan]
10 #serial: /dev/serial/by-id/usb-Klipper_Klipper_firmware_12345-if00
11 canbus_uuid: 028c2b2b0b37
12
13 View 'adxl345' documentation
14 [adxl345]
15 cs_pin: EBBCan: PB12
16 spi_software_sclk_pin: EBBCan: PB10
17 spi_software_mosi_pin: EBBCan: PB11
18 spi_software_miso_pin: EBBCan: PB2
19 axes_map: x,y,z
```

Comment adxl section (if you have it) in your printer.cfg, otherwise you will have a conflict because adxl is also present on your CAN board:

```
[adxl345]
#cs_pin: rpi:None
```



To check if your config is correct, run the command:

```
~/klippy-env/bin/python ~/klipper/scripts/canbus_query.py can0
```

**if everything is working correctly, UUID is not shown anymore because CAN BUS is now initialized:**

```
pi@fluidpi:~ $ ~/klippy-env/bin/python ~/klipper/scripts/canbus_query.py can0
Total 0 uuids found
```

Also run the command: ***sudo ifconfig can0***

Interface can0 should be **UP and RUNNING**:

```
pi@fluidpi:~ $ sudo ifconfig can0
can0: flags=193<UP,RUNNING,NOARP> mtu 16
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 1000
    (UNSPEC)
    RX packets 3808 bytes 29362 (28.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2009 bytes 10792 (10.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

***That's it for canbus installation.***

***You should now proceed with connecting all your peripherals to can board and modify your config files accordingly.***