Version Change History

date	personnel	Manual	discription
2023-03-25	Makerbase	V1.00	Documentation is created.
2023-04-12	Makerbase	V1.00	update profile

Makerbase CANable V2.0 Use Manual

Purchaselink:
https://www.aliexpress.us/item/3256805268926264.html?spm

Part1. Introduction

1.1 Introduce

The CANable 2.0 is a small low-cost open source USB to CAN adapter. The CANable enumerates as a virtual serial port on your computer and acts as a serial-line to CAN bus interface. With the alternative candleLight firmware, the CANable enumerates as a native CAN interface on Linux. CANable 2.0 supports both standard CAN and CAN-FD.

CANable adapters are compatible with ARM-based embedded platforms such as the Raspberry Pi, Raspberry Pi Zero, ODROID, BeagleBone, etc. and are well suited for integration into OEM products.

1.2 Interface

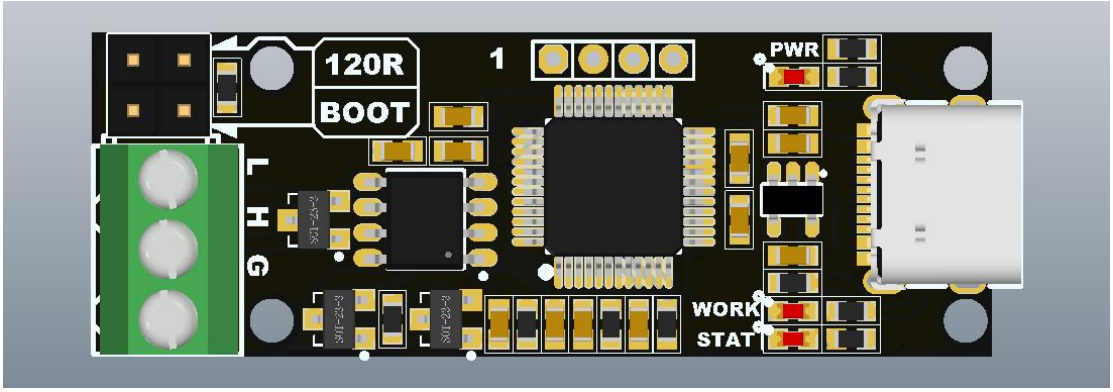
1. Supports CAN2.0A and B, baud rates up to 1M;
2. Initial slcan support for CAN-FD (beta, 2M/5Mbaud);
3. Compatible with socketcan through slcand;
4. Optional enable/disable termination resistor via shorting cap;
5. Support Type-C USB interface, online firmware upgrade and DFU mode upgrade firmware;
6. Using high-performance ARM Cortex-M4 STM32G431C8T6 processor, operating frequency up to 170MHz;

Comparison between MKS CANable V2.0 and MKS CANable V1.0		
Exterior	Makerbase® 	Makerbase 
model	MKS CANable V2.0	MKS CANable V1.0
MCU	STM32G431C8T6	STM32F072C8T6
Communication	TJA1051T/3	TJA1051T/3
Input voltage	DC3.3V-5V	DC3.3V-5V
CAN FD	YES	NO
CAN2.0A/B	YES	YES
Firmware	candleLight、slcan	candleLight、slcan、pcan
Baud	1M/2M/5M	1M
Size	45.0 x 16.2 x 1.6mm	50.0 x 16.0 x 1.6mm

Part2. Hardware

2.1 Schematic

Github: https://github.com/makerbase-mks/CANable-MKS



Part3. Application Software and Firmware

Application	CANable V2.0	CANable V1.0
cangaroo	slcan(support CAN FD)	candleLight/slcan/cantact
BUSMASTER V3.2.2	candleLight	candleLight/pcan/cantact
TSMaster	candleLight	candleLight/pcan/cantact
PCAN-Explorer 5、pcan view	NO	pcan
cantact	slcan(not support CAN FD)	slcan

Part4. Update firmware

This time we will introduce two methods to update the firmware, namely "Web application" and "ST DFU tool".

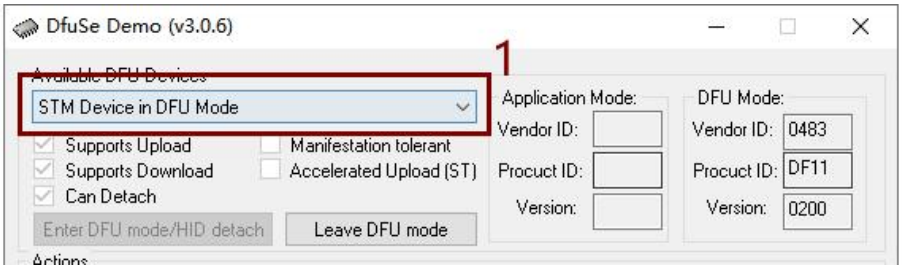
4.1 Hardware connection

1. Connect the short-circuit cap to the two pins of BOOT, and then connect the Type-C end to the USB port of the PC, as shown in the figure below.

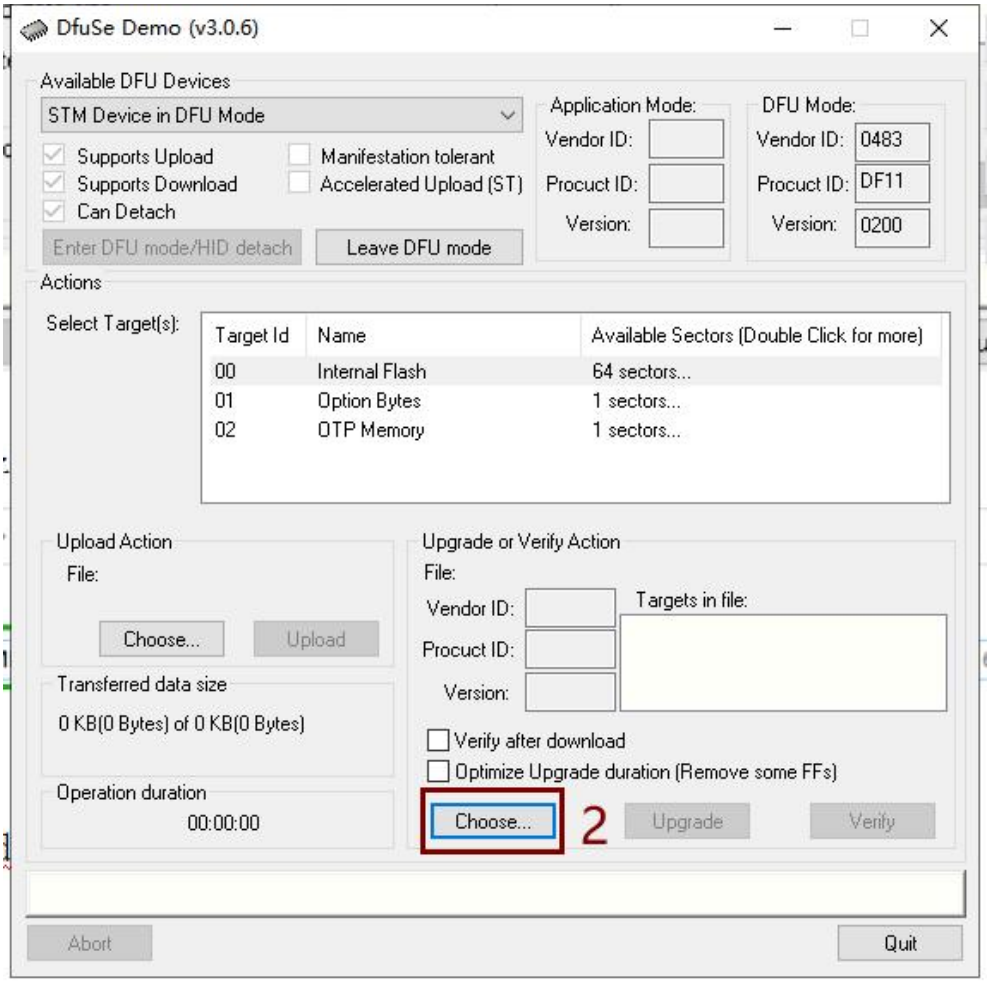
4.2 DfuSe

1. Open the DfuSe installation directory (C:\Program Files (x86)\STMicroelectronics\Software\DfuSe v3.0.6\Bin\Driver), choose to install the driver according to the computer system, as shown in the figure below.

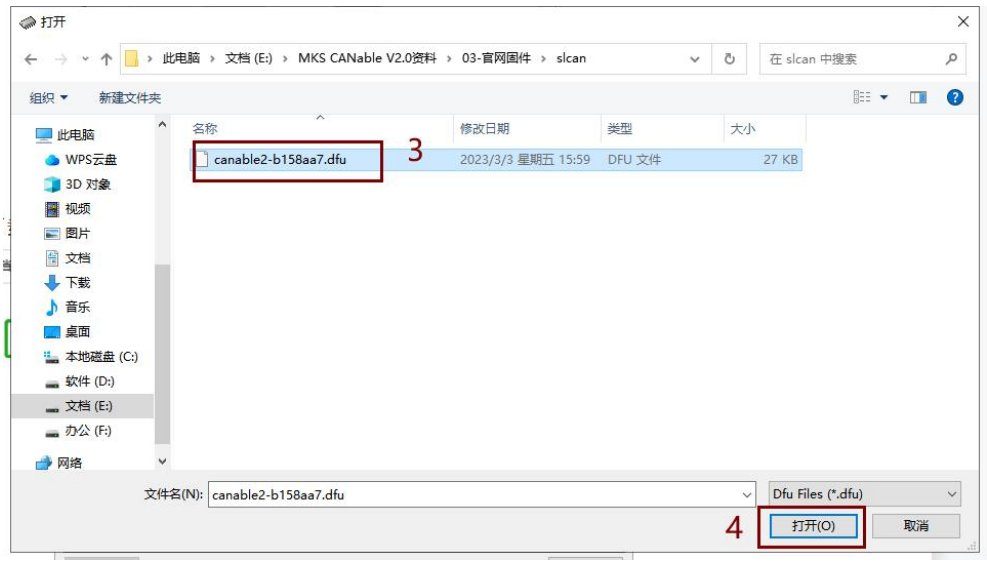
2. Open the DfuSe program and select the device that needs to update the firmware, as shown in Figure 1 below.



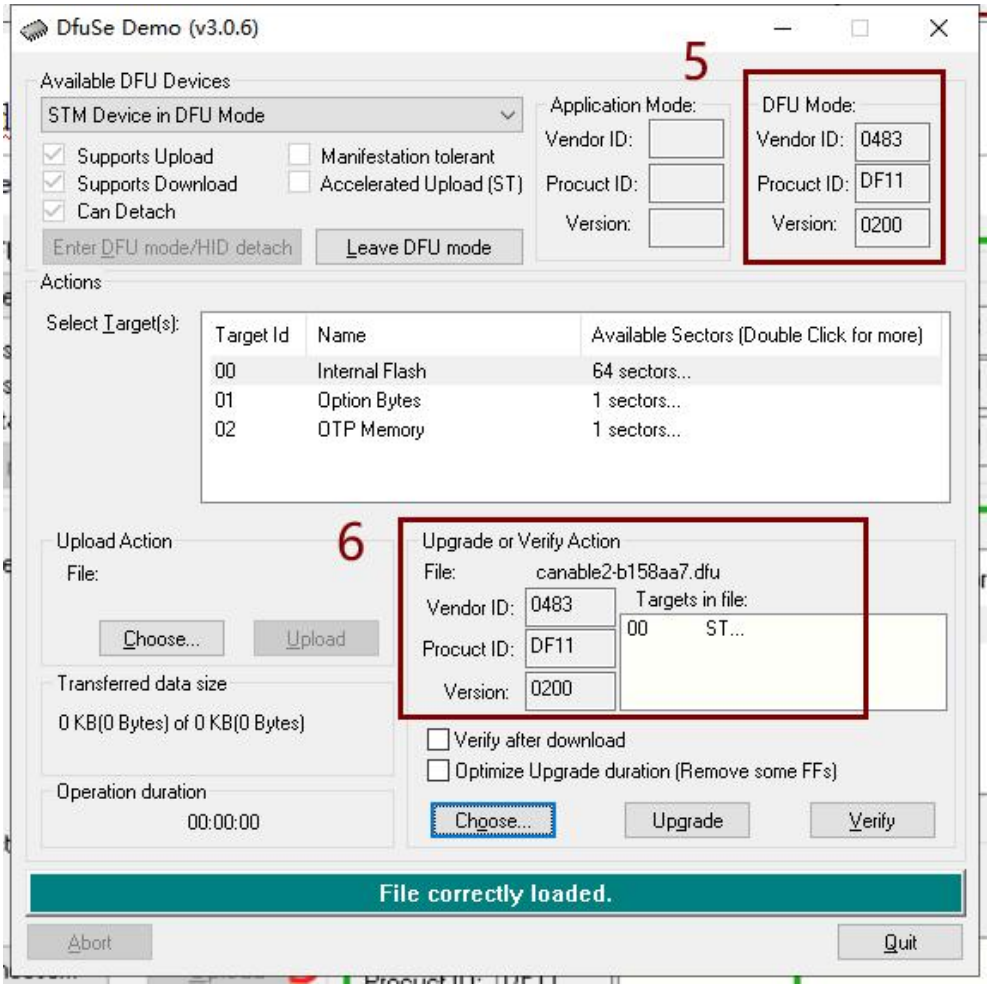
3. Click "Choose..." to select the firmware file, as shown in Figure 2 below.



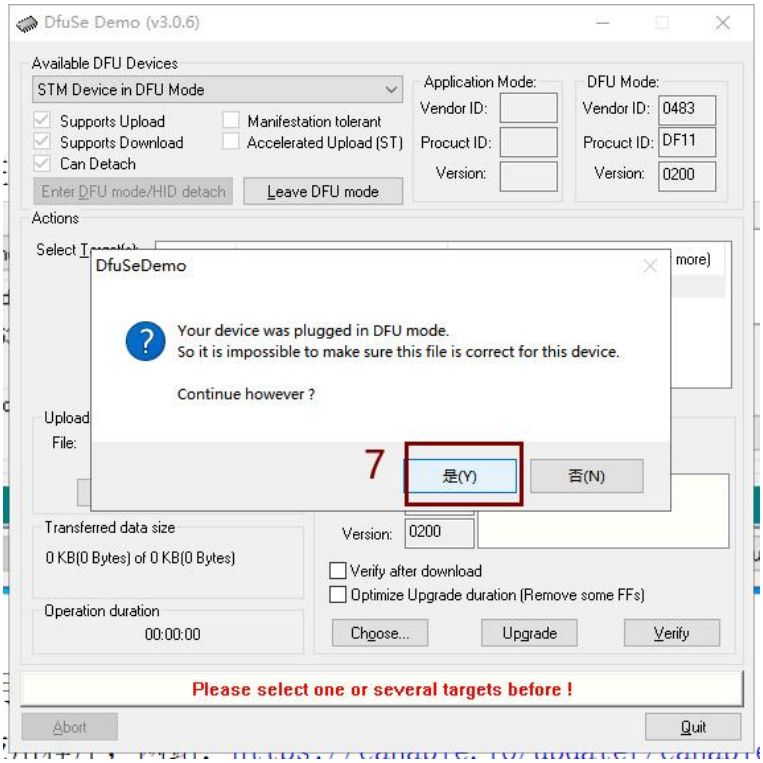
4. Select the downloaded firmware file "canable2-b158aa7.dfu", as shown in Figure 3 below.



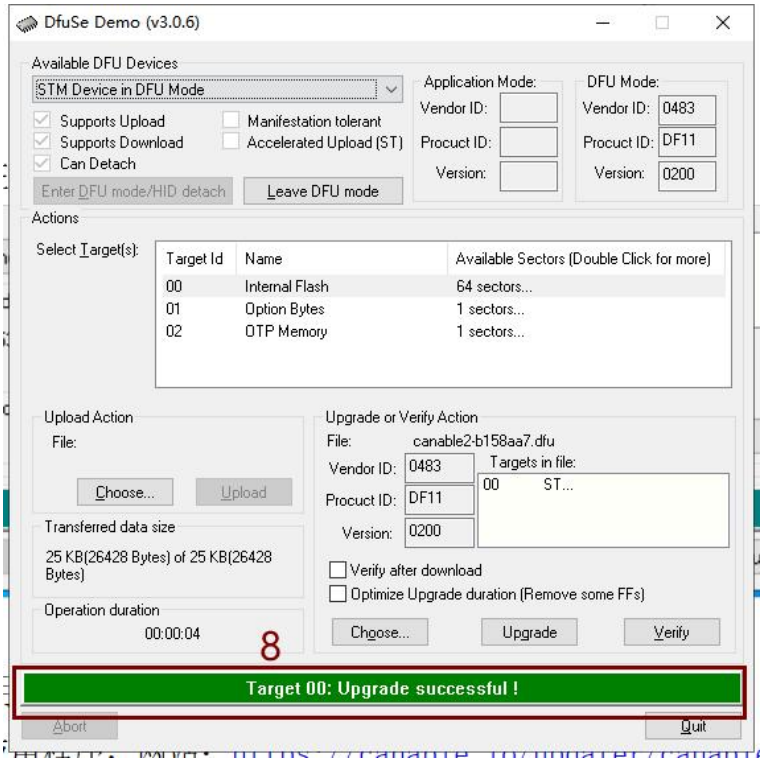
5. Check whether "Vender ID, Procuct ID, Version" are corresponding, as shown in Figure 5 and 6 below.



6. Click "Yes (Y)", as shown in Figure 7 below;

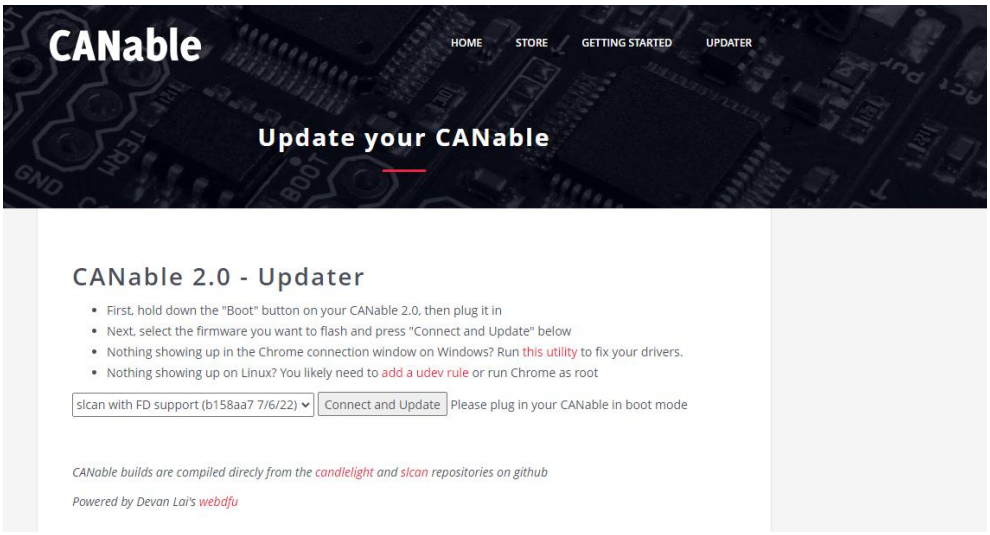


7. The upload is successful, as shown in Figure 8 below.

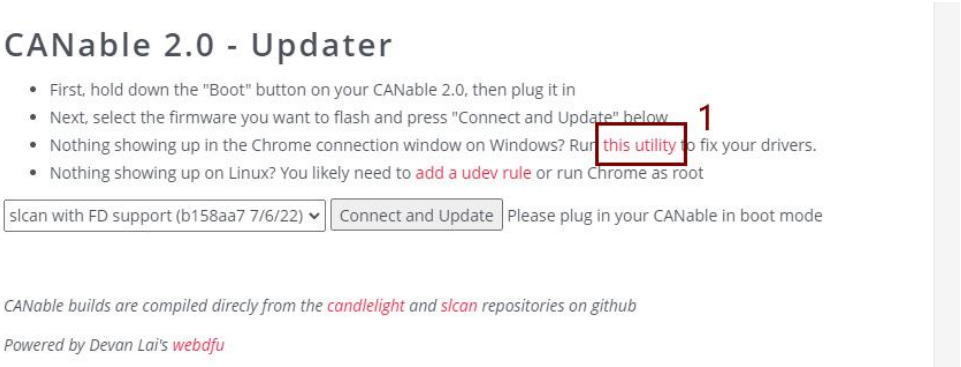


4.3 Web applications

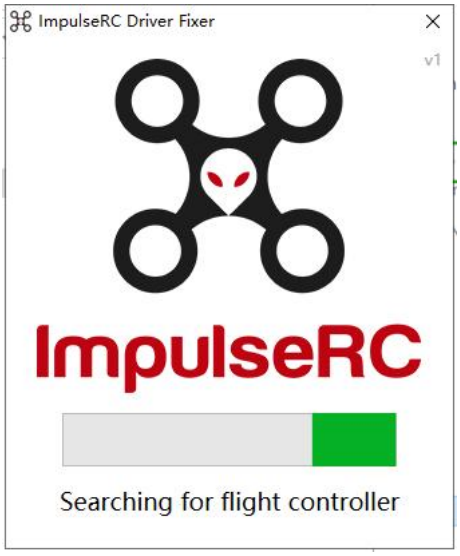
1. Open the web application, website: https://canable.io/updater/canable2.html.



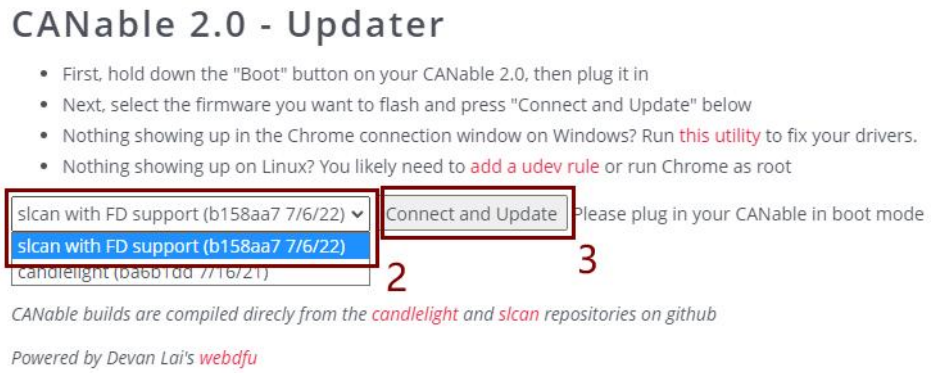
2. Click the picture 1 below to download the driver update program.



3. After the download is successful, open the program to update the driver, as shown in the figure below.

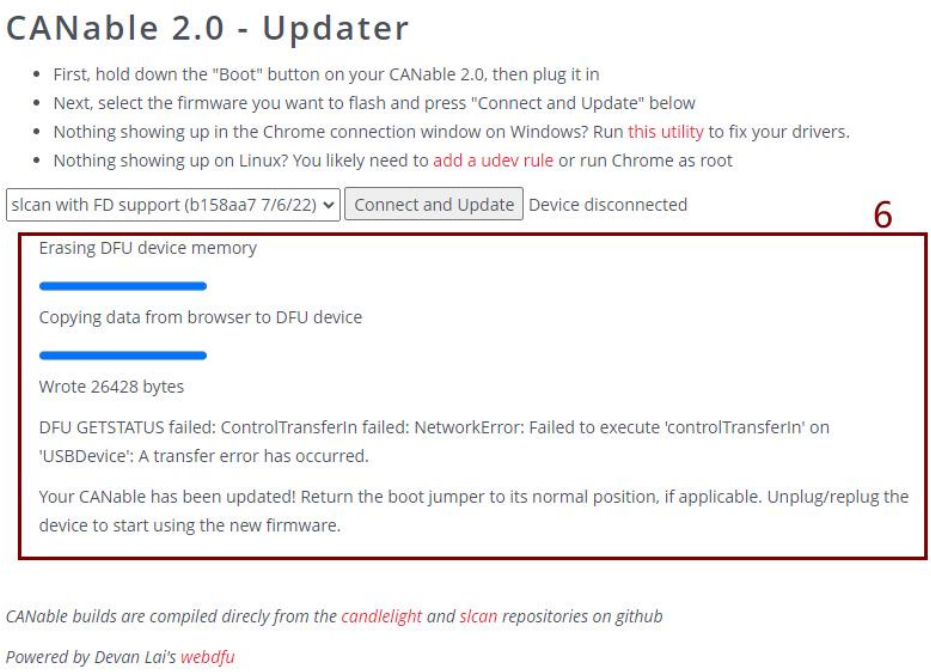


4. Select "slcan with FD support" or "candlelight", as shown in Figure 1 below; click "Connect and Update", as shown in Figure 2 below.



5. Select the device that needs to update the firmware, as shown in Figure 4 below; click "Connect" to upload the firmware, as shown in Figure 5 below.

6. Upload the firmware successfully, as shown in the figure below.

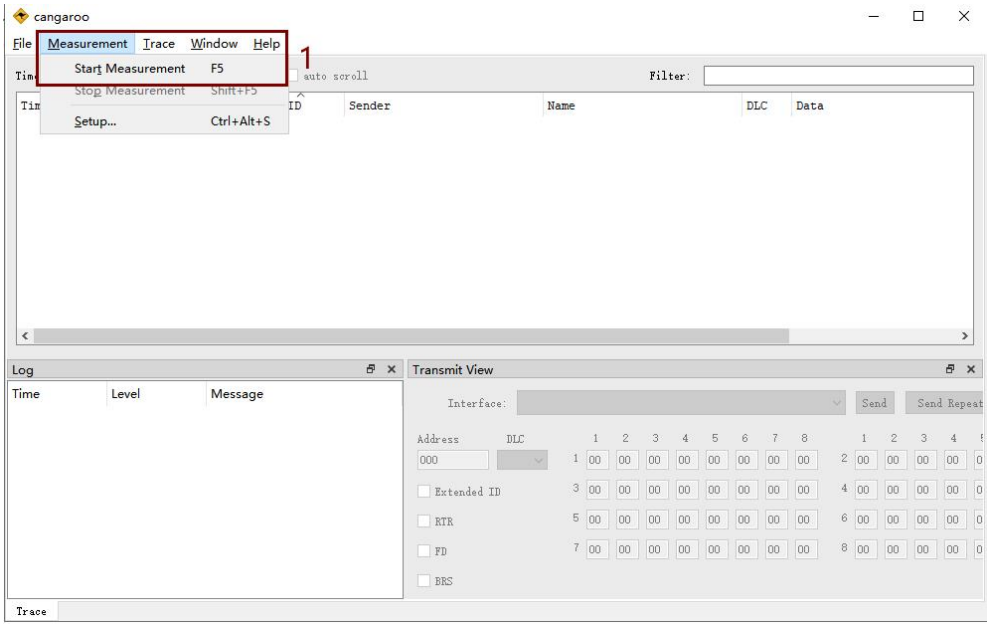


Part5. Window

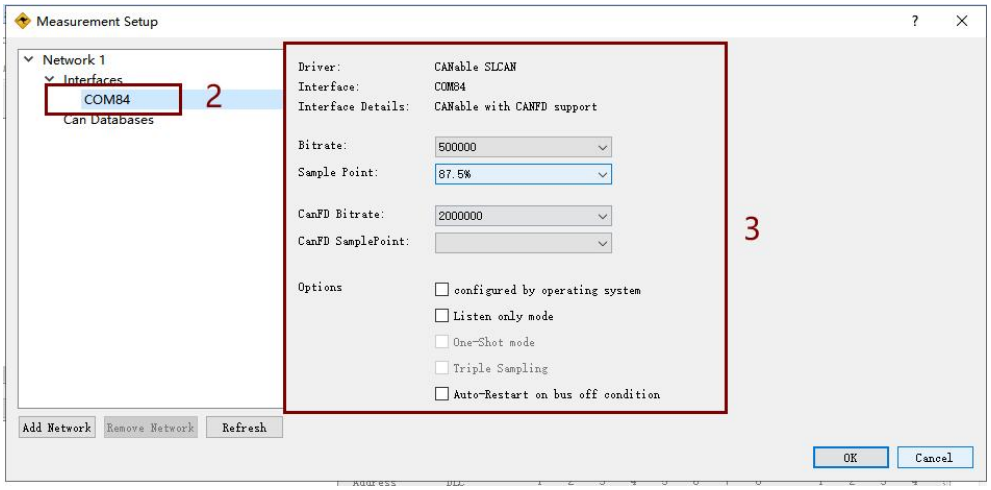
5.1 cangaroo

1. Open the cangaroo program, as shown in Figure 1 below.

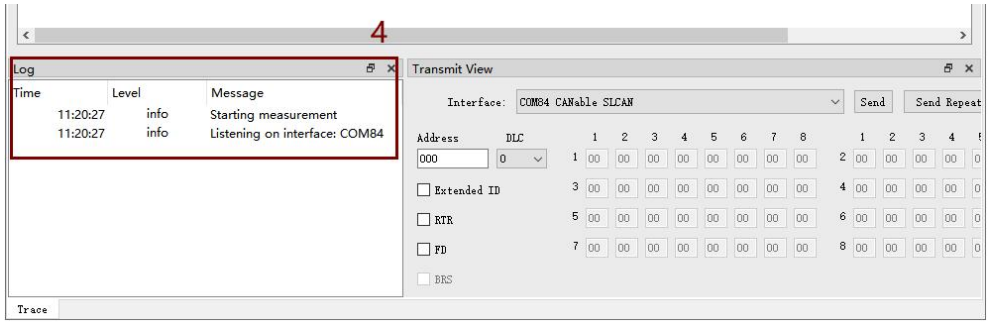
2. Select Measurement > Start Measurement or press F5 to enter the configuration page, as shown in Figure 1 below.



3. Select the port where the module is located, as shown in Figure 2 below; set the module communication protocol parameters as shown in Figure 3 below, click OK after setting.



4. The configuration is complete, the page is as shown in the figure below.



Part6. Linux

6.1 Use of candleLight under Linux

1. Connect the MKS CANable V2.0 module to the ubuntu system, a can0 device will come out, send commands to view:

```
# ifconfig -a
```

2. Install can-utils, after the installation is complete, you can set the can0 device.

```
# sudo apt-get install can-utils
```

Note: If it is an embedded system, you need to transplant the can-utils tool.

3. Set CAN communication bit rate:

```
# sudo ip link set can0 type can bitrate 500000
```

4. Enable CAN:

```
# sudo ifconfig can0 up
```

5. Configure the storage length of the transmission data buffer of txqueuelen:

```
# sudo ifconfig can0 txqueuelen 1000
```

6. can0 send data:

For example:ID:0x888, data:0xABCDEFAB:

```
# cansend can0 888#ABCDEFAB
```

7. can0 receive data:

```
# candump can0
```

8. close can0 device:

```
# sudo ifconfig can0 down
```

6.2 Use of slcan in LinuxLinux

1. Connect the MKS CANable V2.0 module to the ubuntu system, and send commands to query the device:

```
# ls /dev/ttyACMx
（x 可能是 0, 1, 2, 3, .......）
```

2. Map ttyACM0 to can0, and set the bit rate to 500K:

```
# sudo slcand -o -c -s6 /dev/ttyACM0 can0
```

The corresponding setting of the bit rate is as follows:

```
-s0 = 10k
-s1 = 20k
-s2 = 50k
-s3 = 100k
-s4 = 125k
-s5 = 250k
-s6 = 500k
-s7 = 750k
-s8 = 1M
```

3. Enable CAN:

```
# sudo ifconfig can0 up
```

4. Configure the storage length of the transmission data buffer of txqueuelen:

```
# sudo ifconfig can0 txqueuelen 1000
```

5. can0 send data:

For example:ID:0x888, data:0xABCDEFAB:

```
# cansend can0 888#ABCDEFAB
```

6. can0 receive data:

```
# candump can0
```

7. Calculate bus loading percentage on can0:

```
# canbusload can0 500000
```

8. Top level view showing can traffic can:

```
# cansniffer can0
```

9. Generate fixed data CAN messages:

```
# cangen can0 -D 11223344DEADBEEF -L 8
```

6.3 Use of SocketCAN on Linux

1. Install can-utils, after the installation is complete, you can set the can0 device:

```
sudo apt-get install can-utils
```

Note: If it is an embedded system, you need to transplant the can-utils tool.

2. Set can bitrate to 500K:
```
sudo ip link set can0 type can bitrate 500000
```

3. Enable CAN:
```
sudo ifconfig can0 up
```

4. Configure the storage length of the transmission data buffer of txqueuelen:
```
sudo ifconfig can0 txqueuelen 1000
```

5. can0 send data:
For example:ID:0x888, data:0xABCDEFAB:
```
cansend can0 888#ABCDEFAB
```

6. can0 receive data:
```
candump can0
```

7. close can0 device:
```
sudo ifconfig can0 down
```

8. Calculate bus loading percentage on can0:
```
canbusload can0 500000
```

9. Top level view showing can traffic:
```
cansniffer can0
```

10. Generate and send fixed data CAN messages for testing:
```
cangen can0 -D 11223344DEADBEEF -L 8
```

6.4 Slcan is used on embedded linux

1. Transplant can-utils:
(1)Download the latest version can-utils 4.0.6 source code:
http://www.pengutronix.de/software/socket-can/download/canutils;

(2) Compiling can-utils requires libsocketcan library support, download libsocketcan 0.0.10:
http://www.pengutronix.de/software/libsocketcan/download/;

(3)Unpack libsocketcan-0.0.10.tar.bz2. Execute the configure command:

```
./configure --host=arm-linux-gnueabi
--prefix=/home/can/install/libsocketcan

make

make install
```

libsocketcan compiled.

(4)Unzip canutils-4.0.6.tar.bz2, enter the decompression directory, and execute the configure command:

```
./configure --host=arm-linux-gnueabi
--prefix=/home/can/install/canutils    libsocketcan_LIBS=-lsocketcan
LDFLAGS=-L/home/can/install/libsocketcan/lib
libsocketcan_CFLAGS=-I/home/can/install/libsocketcan/include
CFLAGS=-I/home/can/install/libsocketcan/include
```

Among them, --host is to specify the cross tool chain, --prefix is to specify the location where the file is generated, libsocketcan_LIBS is to specify the library that canconfig needs to link, LDFLAGS is to specify the path of the external library, and CPPFLAGS is to specify the path of the external header file.

(5)Execute make, make install, generate four directories under /home/can/install/canutils/canutils, and copy them to the corresponding directories of the development board file system.

2. Kernel configuration driver support:

```
# make ARCH=arm64 menuconfig
```

Or `# make ARCH=arm menuconfig`

```
[*] Networking support  --->
        <*>  CAN bus subsystem support  --->
                CAN Device Drivers  --->
                        CAN USB interfaces  --->
                                <*> Serial / USB serial CAN Adaptors (slcan)
```

3. After connecting the module to the USB port of the embedded device, the returned message is as follows:

```
[root@RK356X:/]# [   23.890745] usb 3-1: new full-speed USB device number 2 using ohci-platform
[   24.109136] usb 3-1: device descriptor read/all, error -62
[   24.287404] usb 3-1: new full-speed USB device number 3 using ohci-platform
[   24.480738] usb 3-1: device descriptor read/64, error -62
[   24.819141] usb 3-1: New USB device found, idVendor=ad50, idProduct=60c4, bcdDevice= 2.00
[   24.819161] usb 3-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[   24.819169] usb 3-1: Product: CANable 2022 0726
[   24.819176] usb 3-1: Manufacturer: Protofusion Labs
[   24.819183] usb 3-1: SerialNumber: 004C002B5550531620303731
[   24.824832] cdc_acm 3-1:1.0: ttyACM0: USB ACM device
[   24.825910] pwm-backlight backlight1: backlight1 supply power not found, using dummy regulator
[   24.825983] pwm-backlight backlight1: Linked as a consumer to regulator.0
[   24.826039] pwm-backlight backlight1: Dropping the link to regulator.0
[   32.060758] vcc3v3_lcd0_n: disabling
[   32.060792] vcc3v3_lcd1_n: disabling
[   32.060801] vcc3v3_pcie: disabling
```

Please note the obtained ttyACMx device, here is ttyACM0.

```
[root@RK356X:/]# [   23.890745] usb 3-1: new full-speed USB device number 2 using ohci-platform
[   24.109136] usb 3-1: device descriptor read/all, error -62
[   24.287404] usb 3-1: new full-speed USB device number 3 using ohci-platform
[   24.480738] usb 3-1: device descriptor read/64, error -62
[   24.819141] usb 3-1: New USB device found, idVendor=ad50, idProduct=60c4, bcdDevice= 2.00
[   24.819161] usb 3-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[   24.819169] usb 3-1: Product: CANable 2022 0726
[   24.819176] usb 3-1: Manufacturer: Protofusion Labs
[   24.819183] usb 3-1: SerialNumber: 004C002B5550531620303731
[   24.824832] cdc_acm 3-1:1.0: ttyACM0: USB ACM device
[   24.825910] pwm-backlight backlight1: backlight1 supply power not found, using dummy regulator
[   24.825983] pwm-backlight backlight1: Linked as a consumer to regulator.0
[   24.826039] pwm-backlight backlight1: Dropping the link to regulator.0
[   32.060758] vcc3v3_lcd0_n: disabling
[   32.060792] vcc3v3_lcd1_n: disabling
[   32.060801] vcc3v3_pcie: disabling
```

4. Query device:

```
# ls /dev/ttyACM0
 (0 could be 0, 1, 2, 3, .......)
```

5. Map ttyACM0 to can0 and set the bitrate to 500K:

```
# slcand -o -c -s6 /dev/ttyACM0 can0
```

The corresponding setting of the bit rate is as follows:

```
-s0 = 10k
-s1 = 20k
-s2 = 50k
-s3 = 100k
-s4 = 125k
-s5 = 250k
-s6 = 500k
-s7 = 750k
-s8 = 1M
```

6. Enter the command to view the device:

```
# ifconfig -a
```

Return information, as shown in the figure below:

```
[root@RK356X:/]# ifconfig -a

can0    Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
        UP RUNNING NOARP  MTU:16  Metric:1
        RX packets:3845 errors:0 dropped:0 overruns:0 frame:0
        TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 KiB)  TX bytes:0 (0.0 B)
```

7. Enable can0:

```
# ifconfig can0 up
```

8. Configure the storage length of the transmission data buffer of txqueuelen:

```
# ifconfig can0 txqueuelen 1000
```

9. can0 send data,ID:0x888, data:0xABCDEFAB:

```
# cansend can0 888#ABCDEFAB
```

10. can0 accepts data:

```
# candump can0
```

11. Calculate the percent bus load on can0:

```
# canbusload can0 500000
```

12. Top level view showing can traffic:

```
# cansniffer can0
```

13. Generate and send fixed data CAN messages for testing:

```
# cangen can0 -D 11223344DEADBEEF -L 8
```

Welcome to order our products:
https://www.aliexpress.us/item/3256805268926264.html?spm