

# InnoEX Ethernet-CANbus UserGuide



**MILLITRONIC**

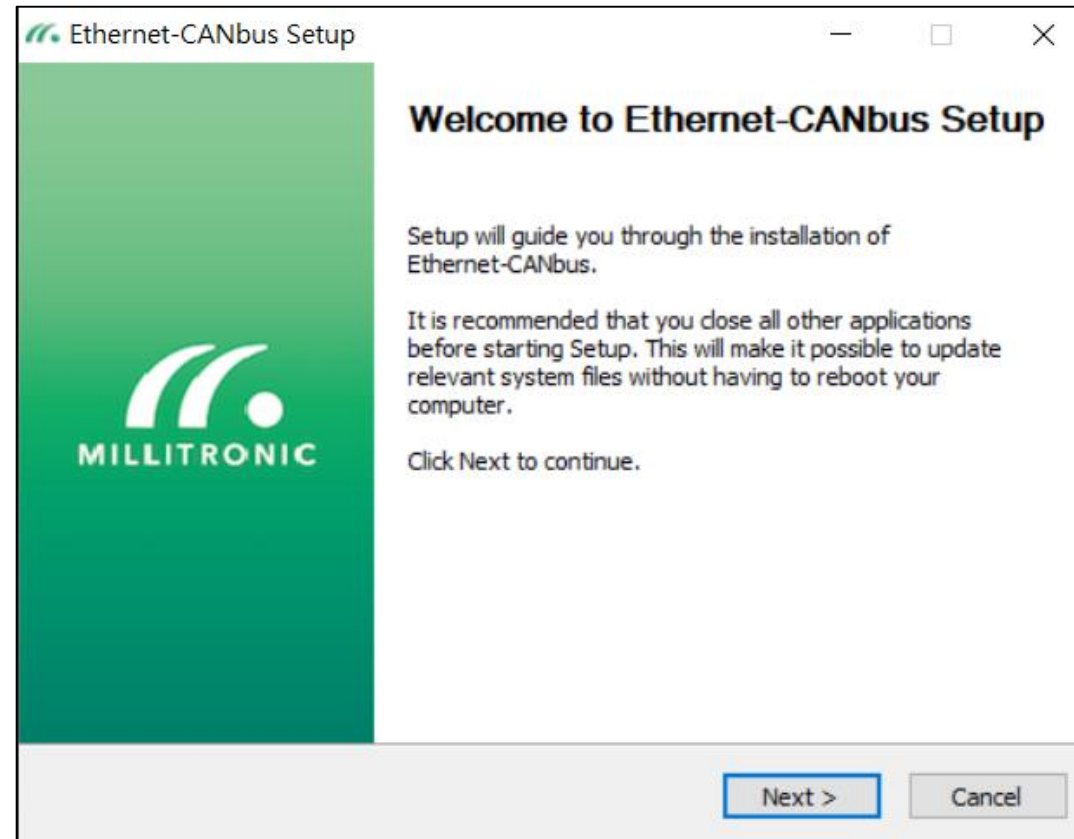
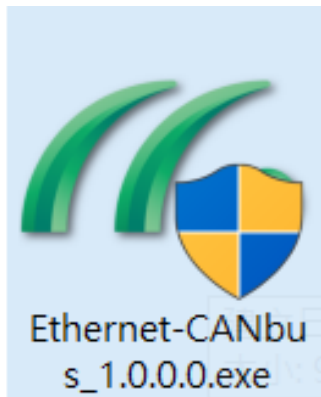
W I G I G   S O L U T I O N   P R O V I D E R

# Abstract

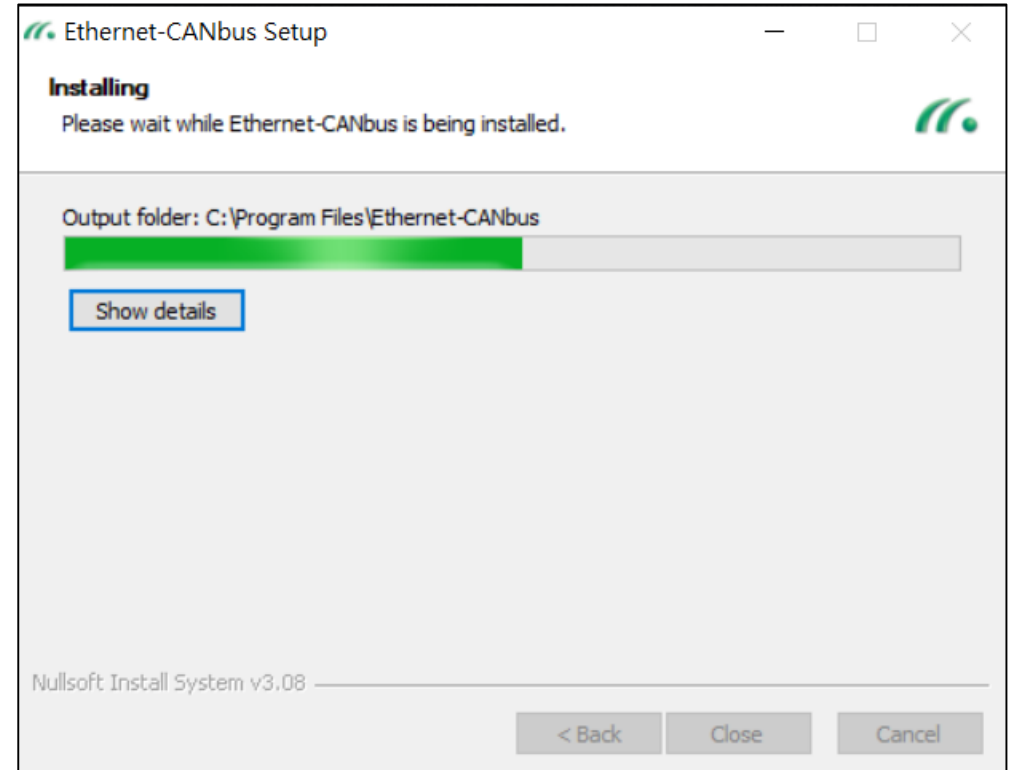
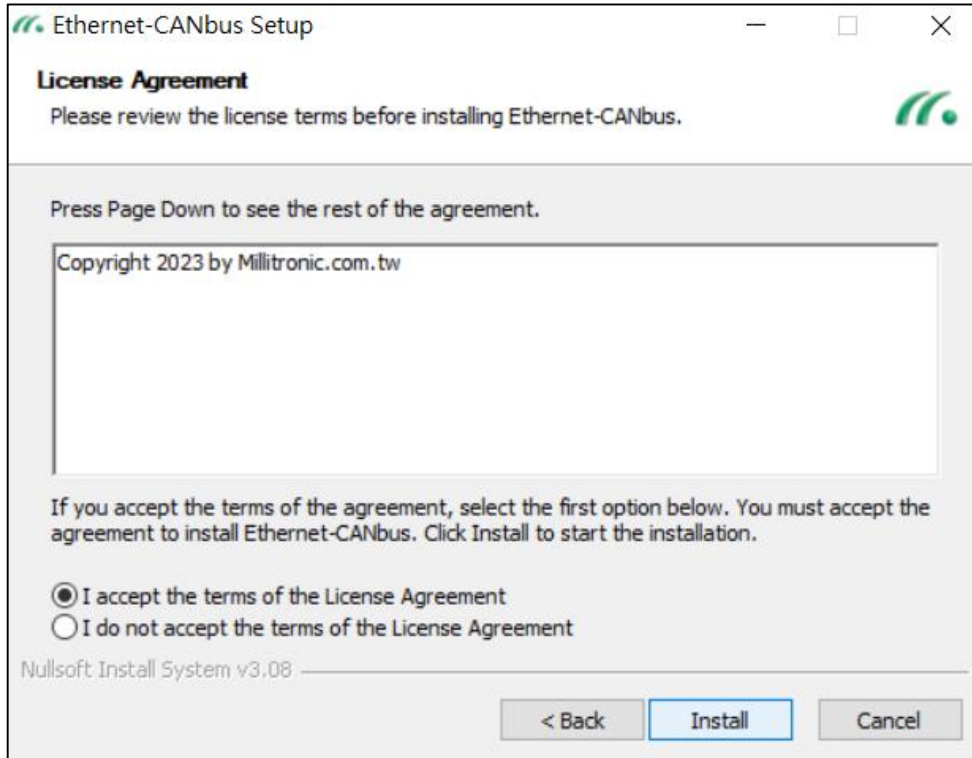
- Ethernet-CANbus provides a IP-base link to bridge CAN bus connection between the local host PC and a remote CAN device.
  - On **Windows**, it requires driver, connection-process(UI/App). To access the CAN device, we provide a test tool and a command line tool. There are source code and SDK of the command line tool so that user can build their own application on windows.
  - On **Linux**, it requires kernel driver, connection-process(command line) and can-utils. Once setup connection, user can access the target CAN device through can-utils.
- The document will guide two platforms OS (Windows and Linux-Ubuntu) for how to setup, how to connect to Ethernet-CANbus device and how to configure the Canbus tools.

# Install on Windows

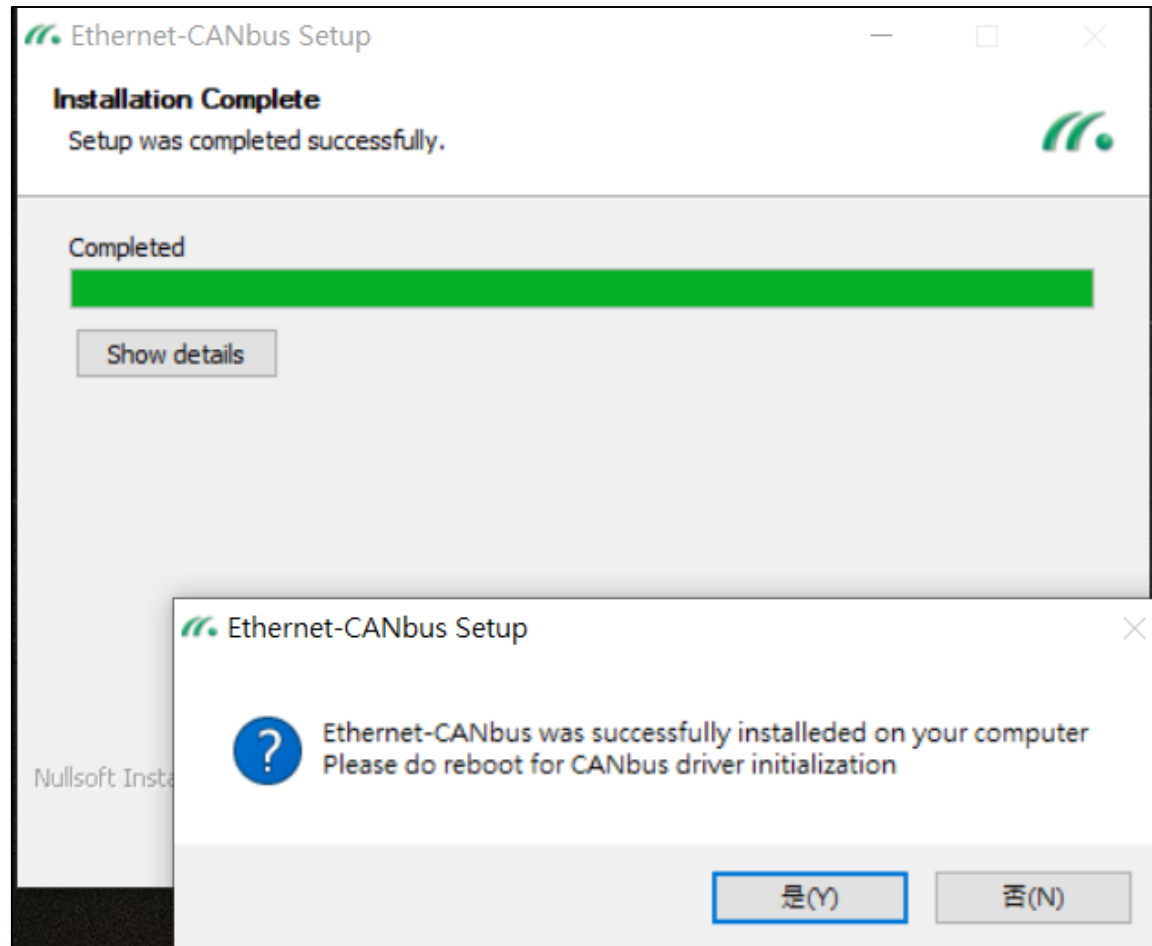
- Click “Ethernet-CANbus\_1.0.0.0.exe” to install APP
- Click “Next”



- Click “Install” to start installing

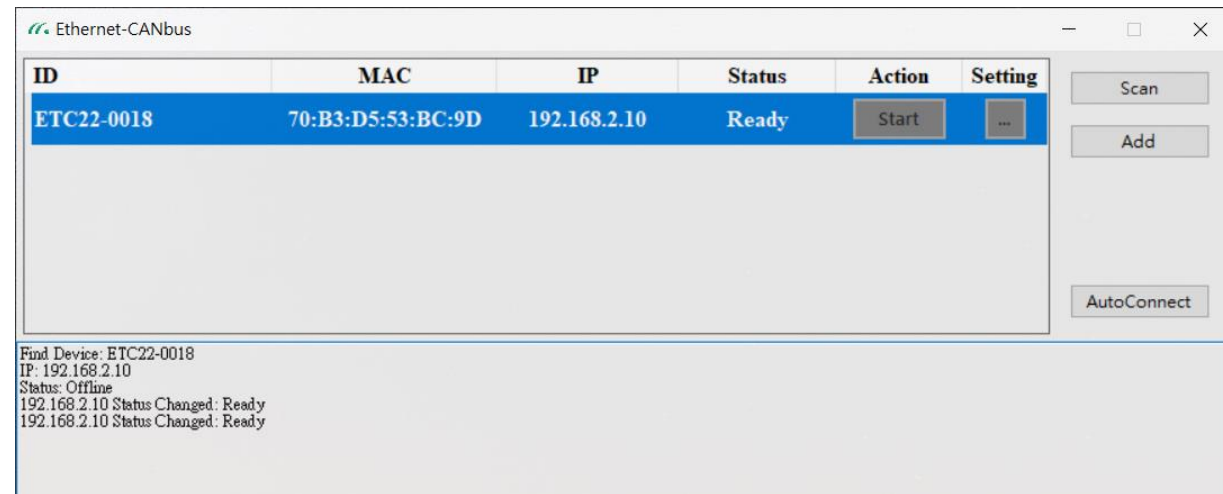


- When it finishes, it will ask to **reboot** PC for CANbus driver initialization
- If you click “**Yes**” it will reboot immediately

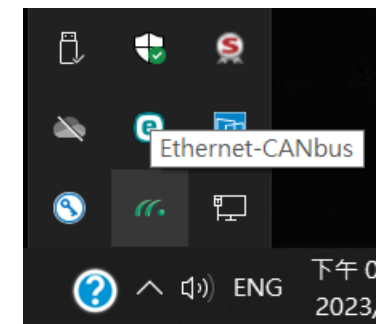
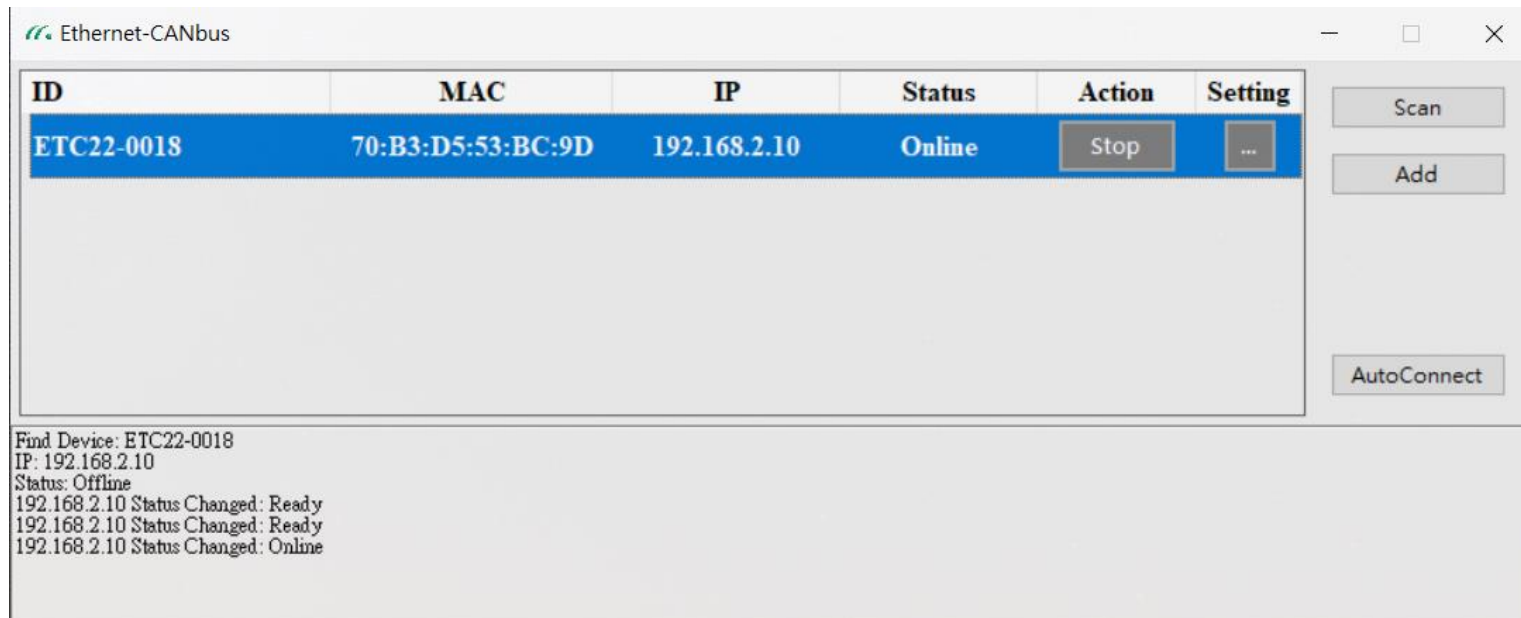


# Windows - Use App

- Click “Ethernet-CANbus” at Desktop
- It will shows the form and connect device automatically  
(Default device IP-address is 192.168.2.10)

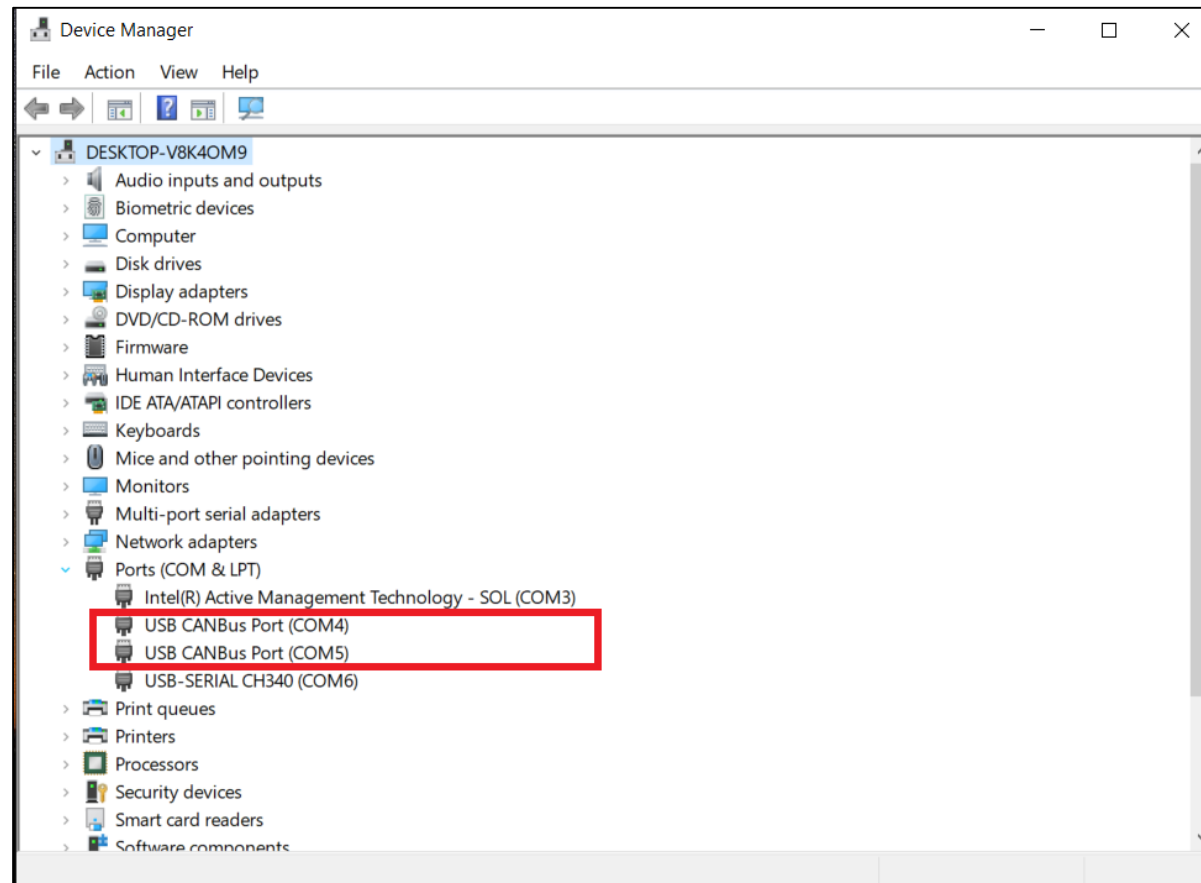


- When it connect successfully, it will show status as this
- Click “**Stop**” button, it will disconnect device
- If click “-” button, the form will hide, click the Notification Icon to show the form



# Windows - Check CANbus device

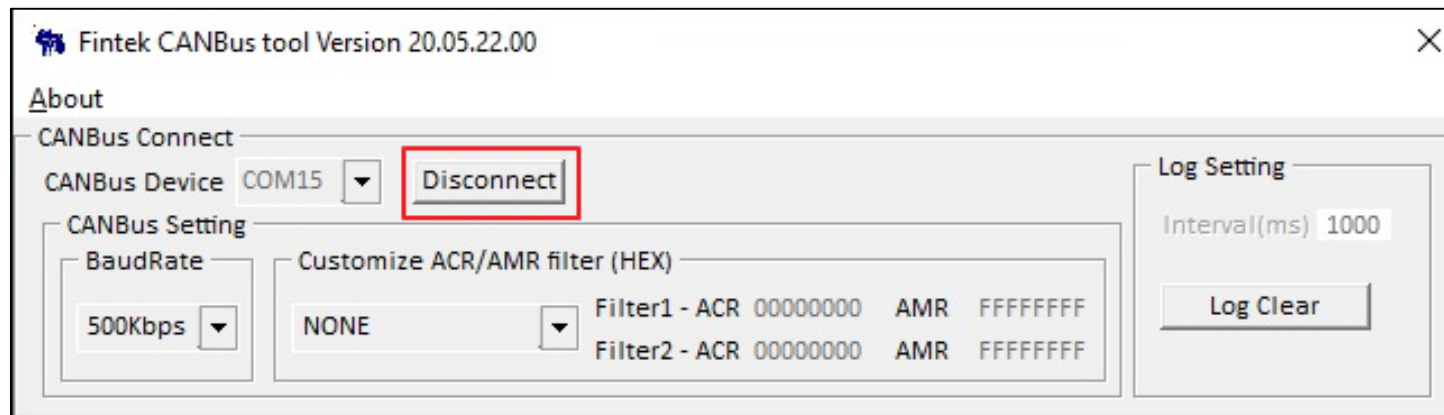
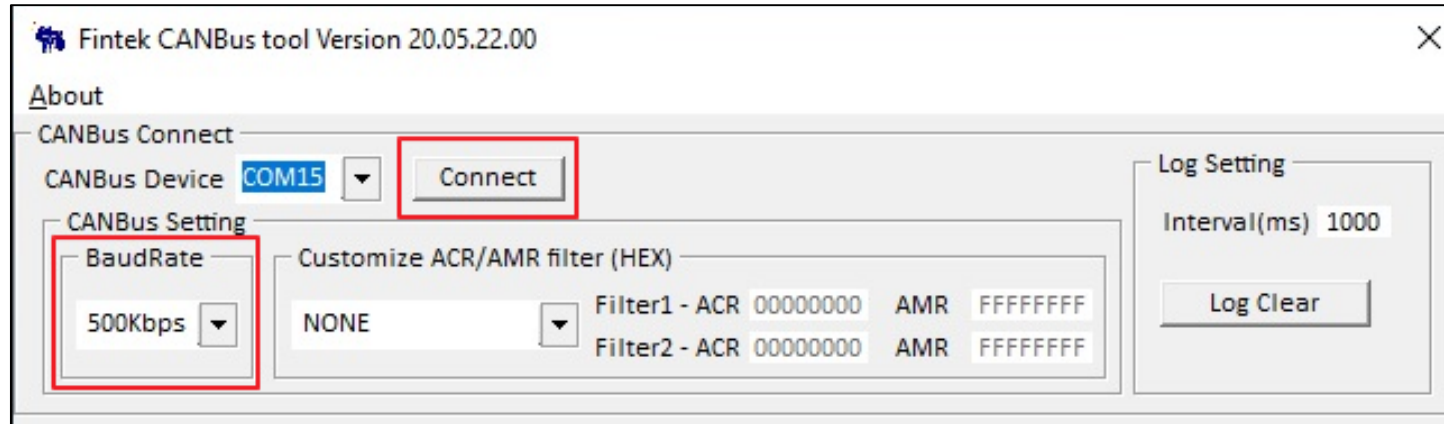
- If device connected, you can open “**Device Manager**” to check CANbus device Com-port







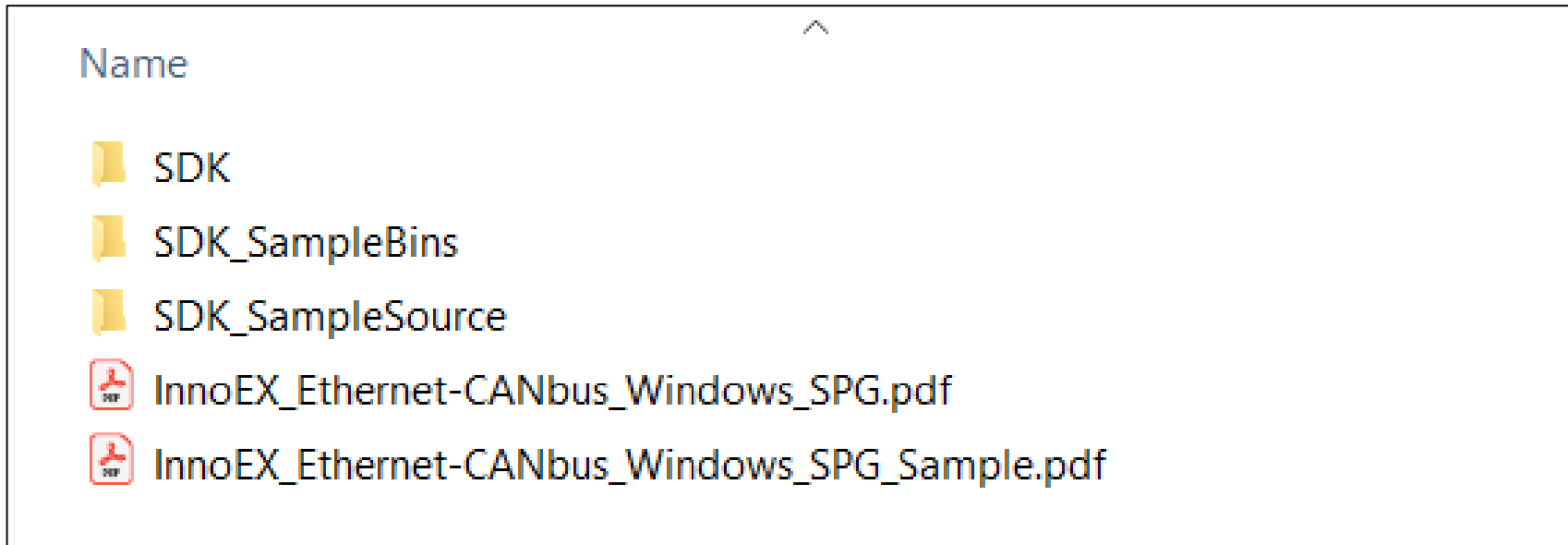
- Before testing, please set **BaudRate**, and then click "**Connect**" to start CANbus.
- After successfully starting CAN, the button will change from "**Connect**" to "**Disconnect**".





# Windows – CANbus SDK

- We provide CANbus SDK for programming development
- You can read these two guides for more detail.



# Install on Linux

- Prepare the kernel tree & compiler tools for your distribution:
  - `sudo su`
  - `apt-get update`
  - `apt-get install build-essential fakeroot gcc kernel-package libncurses5-dev`  
(if your target system is Debian/Ubuntu based)

# Build Kernel Module and Copy

- Cd to `./mlvc_v6.1.0.8-0.1.2` and do “make” to build **mlvc.ko**
- And then copy `mlvc.ko` to previous path

```
david@ubuntu:~/mlvc/mlvc_v.6.1.0.8-0.1.2$ make
make -C /lib/modules/`uname -r`/build M=`pwd` modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-136-generic'
CC [M] /home/david/mlvc/mlvc_v.6.1.0.8-0.1.2/main.o
CC [M] /home/david/mlvc/mlvc_v.6.1.0.8-0.1.2/hcd.o
CC [M] /home/david/mlvc/mlvc_v.6.1.0.8-0.1.2/link.o
CC [M] /home/david/mlvc/mlvc_v.6.1.0.8-0.1.2/urb.o
LD [M] /home/david/mlvc/mlvc_v.6.1.0.8-0.1.2/mlvc.o
Building modules, stage 2.
MODPOST 1 modules
CC [M] /home/david/mlvc/mlvc_v.6.1.0.8-0.1.2/mlvc.mod.o
LD [M] /home/david/mlvc/mlvc_v.6.1.0.8-0.1.2/mlvc.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-136-generic'
david@ubuntu:~/mlvc/mlvc_v.6.1.0.8-0.1.2$ ll
total 15252
drwxrwxr-x 2 david david 4096 Jan 30 23:46 ./
drwxrwxr-x 3 david david 4096 Jan 30 23:46 ../
-rw-rw-r-- 1 david david 2944 Jan 30 23:45 fthclink.h
-rw-rw-r-- 1 david david 2444 Jan 30 23:45 fthclink.h
-rw-rw-r-- 1 david david 424 Jan 30 23:45 ftrelease.h
-rw-rw-r-- 1 david david 18845 Jan 30 23:45 hcd.c
-rw-rw-r-- 1 david david 1622552 Jan 30 23:46 hcd.o
-rw-rw-r-- 1 david david 40458 Jan 30 23:46 hcd.o.cmd
-rw-rw-r-- 1 david david 12153 Jan 30 23:45 link.c
-rw-rw-r-- 1 david david 1610096 Jan 30 23:46 link.o
-rw-rw-r-- 1 david david 38469 Jan 30 23:46 link.o.cmd
-rw-rw-r-- 1 david david 6597 Jan 30 23:45 main.c
-rw-rw-r-- 1 david david 1606776 Jan 30 23:46 main.o
-rw-rw-r-- 1 david david 40469 Jan 30 23:46 main.o.cmd
-rw-rw-r-- 1 david david 791 Jan 30 23:45 Makefile
-rw-rw-r-- 1 david david 4450448 Jan 30 23:46 mlvc.ko
-rw-rw-r-- 1 david david 310 Jan 30 23:40 mlvc.ko.cmd
-rw-rw-r-- 1 david david 179 Jan 30 23:46 mlvc.mod
-rw-rw-r-- 1 david david 2284 Jan 30 23:46 mlvc.mod.c
-rw-rw-r-- 1 david david 299 Jan 30 23:46 mlvc.mod.cmd
-rw-rw-r-- 1 david david 5752 Jan 30 23:46 mlvc.mod.o
-rw-rw-r-- 1 david david 30968 Jan 30 23:46 mlvc.mod.o.cmd
-rw-rw-r-- 1 david david 4445864 Jan 30 23:46 mlvc.o
-rw-rw-r-- 1 david david 344 Jan 30 23:46 mlvc.o.cmd
-rw-rw-r-- 1 david david 46 Jan 30 23:46 modules.order
-rw-rw-r-- 1 david david 0 Jan 30 23:46 Module.symvers
-rw-rw-r-- 1 david david 148 Jan 30 23:45 releasenote.txt
-rw-rw-r-- 1 david david 5898 Jan 30 23:45 urb.c
-rw-rw-r-- 1 david david 1558808 Jan 30 23:46 urb.o
-rw-rw-r-- 1 david david 38286 Jan 30 23:46 urb.o.cmd
david@ubuntu:~/mlvc/mlvc_v.6.1.0.8-0.1.2$
```

```
david@ubuntu:~/mlvc/mlvc_v.6.1.0.8-0.1.2$ cp -f ./mlvc.ko ../
david@ubuntu:~/mlvc/mlvc_v.6.1.0.8-0.1.2$ cd ../
david@ubuntu:~/mlvc$ ll
total 5092
drwxrwxr-x 3 david david 4096 Jan 31 00:28 ./
drwxr-xr-x 18 david david 4096 Jan 30 23:45 ../
-rwxrwxr-x 1 david david 26968 Jan 30 23:46 mlvcctrl*
-rw-rw-r-- 1 david david 166 Jan 30 23:45 mlvcctrl.service
-rw-rw-r-- 1 david david 4450448 Jan 31 00:28 mlvc.ko
-rw-rw-r-- 1 david david 716416 Jan 30 23:45 mlvc-pkg_x64.tmp
drwxrwxr-x 2 david david 4096 Jan 30 23:46 mlvc_v.6.1.0.8-0.1.2/
david@ubuntu:~/mlvc$
```

# Use mlvctrl

- Enter command with -h or help to print the command usage
- Make sure to check using root permission (it will ask sudo permission and passwords automatically)

```
david@ubuntu:~/mlvc$ ./mlvctrl -h
Info : mlvctrl version is v0.0.7

Usage: mlvctrl  help
        mlvctrl  server_add                <Host_IP>
        mlvctrl  server_remove             <Host_IP>
        mlvctrl  server_all_connect        <Host_IP>
        mlvctrl  server_all_disconnect     <Host_IP>
        mlvctrl  show_list [-d]
        mlvctrl  setup
```

# Setup

- Use “mlvcctrl setup” to install mlvc-pkg

```
david@ubuntu:~/mlvc$ ./mlvcctrl setup
[sudo] password for david:
Do setup mlvc-pkg package, check the installing path...
david@ubuntu:~/mlvc$
```



# Mlvcctrld service

- Use “*sudo systemctl status mlvcctrld.service*” to check active
- After “*setup*” command, it will enable this service(need **reboot** or set **start** to active).

```
david@ubuntu:~$ sudo systemctl status mlvcctrld.service
[sudo] password for david:
● mlvcctrld.service - mlvc control autostart service
   Loaded: loaded (/lib/systemd/system/mlvcctrld.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
```

```
david@ubuntu:~$ sudo systemctl status mlvcctrld.service
● mlvcctrld.service - mlvc control autostart service
   Loaded: loaded (/lib/systemd/system/mlvcctrld.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2023-02-03 01:05:58 PST; 1s ago
     Main PID: 2592 (mlvcctrl)
       Tasks: 2 (limit: 4622)
      CGroup: /system.slice/mlvcctrld.service
              └─2592 /opt/mlvc-pkg/mlvcctrl start -d -a
                 └─2622 /opt/mlvc-pkg/sbin/mlvcd

Feb 03 01:05:58 ubuntu sudo[2598]: pam_unix(sudo:session): session closed for user root
Feb 03 01:05:58 ubuntu sudo[2611]: root : TTY=unknown ; PWD=/ ; USER=root ; COMMAND=/opt/mlvc-pkg/sbin/mlvcd
Feb 03 01:05:58 ubuntu sudo[2611]: pam_unix(sudo:session): session opened for user root by (uid=0)
Feb 03 01:05:58 ubuntu sudo[2611]: pam_unix(sudo:session): session closed for user root
Feb 03 01:05:58 ubuntu sudo[2627]: root : TTY=unknown ; PWD=/ ; USER=root ; COMMAND=/opt/mlvc-pkg/bin/mlvcioctl show
Feb 03 01:05:58 ubuntu sudo[2627]: pam_unix(sudo:session): session opened for user root by (uid=0)
Feb 03 01:05:58 ubuntu sudo[2627]: pam_unix(sudo:session): session closed for user root
Feb 03 01:05:58 ubuntu sudo[2630]: root : TTY=unknown ; PWD=/ ; USER=root ; COMMAND=/opt/mlvc-pkg/bin/mlvcioctl ping 5:1
Feb 03 01:05:58 ubuntu sudo[2630]: pam_unix(sudo:session): session opened for user root by (uid=0)
Feb 03 01:05:58 ubuntu sudo[2630]: pam_unix(sudo:session): session closed for user root
```

# Server add

- Use “*mlvctrl server\_add <Host\_IP>*” to add target host and check the host by show\_list

```
david@ubuntu:~/mlvc$ ./mlvctrl server_add 192.168.50.148
Initialize mlvcd process and clean all servers...
[Info] Add IP address : 192.168.50.148
server is added successfully
connecting... connecting... done
```

# Show Server List

- Check the host server numbers and check devices by **-d**

```
david@ubuntu:~/mlvc$ ./mlvctrl show_list  
Find 1 servers on the list:  
192.168.50.148:33000 - connected (10/10)
```

```
david@ubuntu:~/mlvc$ ./mlvctrl show_list -d  
Servers:  
192.168.50.148:33000 - connected (10/10)  
0001 - 090c:0768:0001 - avail - SMI USB Display (5-1.4, D0374542B7F0633D3C390000)
```

# Server all connect

- Use “*mlvctrl server\_all\_connect <Host IP>*” to connect host all devices
- Make sure to check status “**conn**” after doing connect command

```
david@ubuntu:~/mlvc$ ./mlvctrl show_list -d
Servers:
192.168.0.129:33000 - connected (10/10)
  0008 - 090c:0768:0001 - avail - SMI USB Display (4-1.4, D0374542B7F0633D3C390000)
  0006 - 04a5:b000:2019 - avail - BenQ ideaCam S1 (4-1.1)

david@ubuntu:~/mlvc$ ./mlvctrl server_all_connect 192.168.0.129
[Info] Start to connect all devices of the server ip : 192.168.0.129

david@ubuntu:~/mlvc$ ./mlvctrl show_list -d
Servers:
192.168.0.129:33000 - connected (8/10)
  0008 - 090c:0768:0001 - conn - SMI USB Display (4-1.4, D0374542B7F0633D3C390000)
  0006 - 04a5:b000:2019 - conn - BenQ ideaCam S1 (4-1.1)
```

# Server all disconnect

- Use “*mlvctrl server\_disconnect <Host IP>*” to disconnect all devices from the host

```
david@ubuntu:~/mlvc$ ./mlvctrl show_list -d
Servers:
  192.168.0.129:33000 - connected (8/10)
    0008 - 090c:0768:0001 - conn - SMI USB Display (4-1.4, D0374542B7F0633D3C390000)
    0006 - 04a5:b000:2019 - conn - BenQ ideaCam S1 (4-1.1)

david@ubuntu:~/mlvc$ ./mlvctrl server_all_disconnect 192.168.0.129
[Info] Disconnecting all devices of the server ip : 192.168.0.129

david@ubuntu:~/mlvc$ ./mlvctrl show_list -d
Servers:
  192.168.0.129:33000 - connected (10/10)
    0008 - 090c:0768:0001 - avail - SMI USB Display (4-1.4, D0374542B7F0633D3C390000)
    0006 - 04a5:b000:2019 - avail - BenQ ideaCam S1 (4-1.1)
```

# Server remove

- Use “mlvctrl server\_remove <Host\_IP>” to remove the target host and check using by show\_list

```
david@ubuntu:~/mlvc$  
david@ubuntu:~/mlvc$ ./mlvctrl server_remove 192.168.50.148  
david@ubuntu:~/mlvc$  
david@ubuntu:~/mlvc$  
david@ubuntu:~/mlvc$  
david@ubuntu:~/mlvc$ ./mlvctrl show_list  
Find 0 servers on the list:
```

# Operation Flow

- |   |    |   |
|---|----|---|
| 1. Set mlvcctrld.service is <b>Active</b> | => | After setup, it will need reboot OS or set start up mlvcctrld.service |
| 2. server_add <Host IP>                   | => | Add server Host IP at server list                                     |
| 3. show_list -d                           | => | Check Device ID from the list   |
| 4. Server_all_connect <Host IP>           | => | Connect server all devices  |
| 5. Server_all_disconnect <Host IP>        | => | Disconnect server all devices   |
| 6. server_remove <Host IP>                | => | Remove server by Host IP  |

## Note :

1. remove all servers before OS shutdown and suspend

# Linux - Building Driver

- Cd to **./driver\_v1.14/** Dir and start to build kernel object : **f81604.ko**
- Follow Steps :
  1. make clean && make
  2. sudo make install
  3. reboot

```
david@ubuntu:~/MLCanbus_release/driver_v1.14$ ll
total 72
drwxrwxr-x 2 david david 4096 Feb 13 22:07 ./
drwxrwxr-x 3 david david 4096 Feb 13 21:45 ../
-rwxrwxr-x 1 david david 50731 Feb 13 21:45 f81604.c*
-rwxrwxr-x 1 david david 776 Feb 13 21:45 Makefile*
-rwxrwxr-x 1 david david 5564 Feb 13 21:45 sja1000.h*
david@ubuntu:~/MLCanbus_release/driver_v1.14$ make clean && make
#make -C /lib/modules/5.4.0-137-generic/build M=/home/david/MLCanbus_release/driver_v1.14 clean
rm -rf *.~ *~ *.o *.ko *.mod.c *.cmd *.o.d .tmp_versions Module.symvers modules.order Module.markers
make -C /lib/modules/5.4.0-137-generic/build M=/home/david/MLCanbus_release/driver_v1.14 modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-137-generic'
CC [M] /home/david/MLCanbus_release/driver_v1.14/f81604.o
Building modules, stage 2.
MODPOST 1 modules
CC [M] /home/david/MLCanbus_release/driver_v1.14/f81604.mod.o
LD [M] /home/david/MLCanbus_release/driver_v1.14/f81604.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-137-generic'
#make -C /DataDisk/old/hpeter/allwinner/loboris/test/sunxi_jwrdegoede M=/home/david/MLCanbus_release/driver_v1.14 modules
#make -C /DataDisk/hpeter/DMA-210UII/samsung_android_kernel_3.0 M=/home/david/MLCanbus_release/driver_v1.14 modules
#make -C /DataDisk/freescale/android_x86-6.0-r2/out/target/product/x86/obj/kernel M=/home/david/MLCanbus_release/driver_v1.14 modules
david@ubuntu:~/MLCanbus_release/driver_v1.14$ sudo make install
make -C /lib/modules/5.4.0-137-generic/build M=/home/david/MLCanbus_release/driver_v1.14 INSTALL_MOD_DIR=updates modules_install
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-137-generic'
INSTALL /home/david/MLCanbus_release/driver_v1.14/f81604.ko
At main.c:160:
- SSL error:02001002:system library:fopen:No such file or directory: ../crypto/bio/bss_file.c:72
- SSL error:2006D080:BI0 routines:BI0_new_file:no such file: ../crypto/bio/bss_file.c:79
sign-file: certs/signing_key.pem: No such file or directory
DEPMOD 5.4.0-137-generic
Warning: modules_install: missing 'System.map' file. Skipping depmod.
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-137-generic'
depmod
david@ubuntu:~/MLCanbus_release/driver_v1.14$ reboot
```



# Linux - Check the mlvc and canbus interface

- Check the “*./mlvcctrl show\_list -d*” to check the connection
- If not in connection, please refer to the mlvc user guide for operation
- Use “*ls /sys/class/net/ -al*” command to check CANBUS device is available (can0/can1/... etc.)

```
david@ubuntu:~/MLCanbus_release$ ./mlvcctrl show_list -d
Servers:
  192.168.2.10:33000 - connected (9/10)
    0001 - 2c42:1709:0001 - conn - USB TO CANBUS BRIDGE (3-1, 88635600168801)
```

```
david@ubuntu:~/MLCanbus_release$ ls /sys/class/net/ -al
total 0
drwxr-xr-x  2 root root 0 Feb 13 00:13 .
drwxr-xr-x 69 root root 0 Feb 13 00:13 ..
lrwxrwxrwx  1 root root 0 Feb 13 18:51 can0 -> ../../devices/platform/fthc/usb3/3-1/3-1:1.0/net/can0
lrwxrwxrwx  1 root root 0 Feb 13 18:51 can1 -> ../../devices/platform/fthc/usb3/3-1/3-1:1.0/net/can1
lrwxrwxrwx  1 root root 0 Feb 13 00:13 ens33 -> ../../devices/pci0000:00/0000:00:11.0/0000:02:01.0/net/ens33
lrwxrwxrwx  1 root root 0 Feb 13 00:13 lo -> ../../devices/virtual/net/lo
```

# Linux - Configure Canbus and startup

- The following examples will configure “can0” to bit-rate 250000, sample-point 0.875 and error restart with 100ms.

1. `sudo su`
2. `ifconfig can0 down`
3. `ip link set can0 type can restart-ms 100`
4. `ip link set can0 type can bitrate 250000 sample-point 0.875`

This value should be fine-tune by customer or following table, and the clock should set with half clock source (16MHz / 2 = 8MHz)

Bit timing parameters for sjal000 with 8.000000 MHz ref clock														
nominal					real		Bitrt	nom	real	SampP				
Bitrate	TQ[ns]	PrS	PhS1	PhS2	SJW	BRP	Bitrate	Error	SampP	SampP	Error	BTR0	BTR1	
1000000	125	2	3	2	1	1	1000000	0.0%	75.0%	75.0%	0.0%	0x00	0x14	
800000	125	3	4	2	1	1	800000	0.0%	80.0%	80.0%	0.0%	0x00	0x16	
500000	125	6	7	2	1	1	500000	0.0%	87.5%	87.5%	0.0%	0x00	0x1c	
250000	250	6	7	2	1	2	250000	0.0%	87.5%	87.5%	0.0%	0x01	0x1c	
125000	500	6	7	2	1	4	125000	0.0%	87.5%	87.5%	0.0%	0x03	0x1c	
100000	625	6	7	2	1	5	100000	0.0%	87.5%	87.5%	0.0%	0x04	0x1c	
50000	1250	6	7	2	1	10	50000	0.0%	87.5%	87.5%	0.0%	0x09	0x1c	
20000	3125	6	7	2	1	25	20000	0.0%	87.5%	87.5%	0.0%	0x18	0x1c	
10000	6250	6	7	2	1	50	10000	0.0%	87.5%	87.5%	0.0%	0x31	0x1c	

5. `ip link set can0 type can berr-reporting on`
6. `ifconfig can0 txqueuelen 1000`
7. `tc qdisc add dev can0 root handle 1: pfifo`
8. `ifconfig can0 up`

- After config two canbus interfaces, you can use “**ifconfig**” for checking
- We also attach a script to bring up canbus, and you can edit and execute it by yourself.
- Command : **sudo sh ./canbus\_up.sh can0**

```
david@ubuntu:~/MLCanbus_release$ ifconfig
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 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

can1: 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 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.243.132 netmask 255.255.255.0 broadcast 192.168.243.255
    inet6 fe80::55e8:6c20:71bd:d486 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:ce:09:67 txqueuelen 1000 (Ethernet)
    RX packets 2479155 bytes 266355356 (266.3 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1446930 bytes 179142920 (179.1 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 44328 bytes 7404064 (7.4 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 44328 bytes 7404064 (7.4 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
MLCanbus_release > $ canbus_up.sh
1  #!/bin/bash -e
2  Tarcanbus=$1
3  ↓
4  if [ -z $(ls /sys/class/net/ | grep $Tarcanbus) ];
5  then
6  |... echo "$Tarcanbus not found!!" && exit
7  fi
8  ↓
9  echo "Set \"$Tarcanbus\" canbus initial and up..."
10 ifconfig $Tarcanbus down
11 ip link set $Tarcanbus type can restart-ms 100
12 ip link set $Tarcanbus type can bitrate 250000 sample-point 0.875
13 ip link set $Tarcanbus type can berr-reporting on
14 #ip link set $Tarcanbus type can loopback off
15 ifconfig $Tarcanbus txqueuelen 1000
16 tc qdisc add dev $Tarcanbus root handle 1: pfifo
17 ifconfig $Tarcanbus up
18 ↓
19 |
```

# Linux - Using can-utils to operate Canbus

- We can get can-utils with following command

Debian/Ubuntu :

`apt-get install can-utils`

Frdora/Centos/RHEL :

`yum install can-utils`

Source code download link :

<https://github.com/linux-can/can-utils>

- We'll use "candump" to receive data, "cangen" & "cansend" to send data. The "cangen" will send random data & ID and "cansend" will send specific data & ID to canbus.

## Cangen and cansend example

```
david@ubuntu:~$ cangen can1 -n 10  
david@ubuntu:~$ cansend can1 123#1122  
david@ubuntu:~$ cansend can1 456#7777888899994444  
david@ubuntu:~$
```

## Candump log

```
david@ubuntu:~/MLCanbus_release$ candump can0  
can0 769 [8] FE 73 FA 4C 91 5A 2F 07  
can0 388 [2] 87 AF  
can0 5F9 [8] BF BA 90 22 29 CB 75 23  
can0 15F [2] 0F 61  
can0 514 [8] 42 7F 55 07 80 A1 6F 23  
can0 3FC [8] 90 87 B9 59 B0 1C 50 4C  
can0 675 [3] E5 D7 9C  
can0 3DA [2] AD 99  
can0 08E [8] D5 B5 4F 5C 17 6C 3C 45  
can0 53D [8] 15 36 55 7B 37 5B 09 5E  
can0 123 [2] 11 22  
can0 456 [8] 77 77 88 88 99 99 44 44
```

A decorative graphic consisting of several concentric circles of varying shades of green, centered on the left side of the slide.

# THANK YOU

