

Industrial CAN FD Series Control Board

User Manual

Copyright

Copyright© 2024 SUNIX Co., Ltd. All Rights Reserved.

No part of this publication may be reproduced, transcribed, stored in a retrieval system, translated into any language, or transmitted in any form or by any means, photocopying, manual, or otherwise, without prior written permission from SUNIX.

Disclaimer

SUNIX shall not be liable for any incidental or consequential damages resulting from the performance or use of this equipment.

SUNIX makes no representations or warranties regarding the contents of this manual. Information in this manual has been carefully checked for reliability; however, no guarantee is given as to the correctness of this content. In the interest of continued product improvement, this company reserves the right to revise the manual or include change in the specifications of the product described within it at any time without notice and without obligation to notify any person of such revision or changes. The information contained in this manual is provided for general use by the customers.

Safety Information

1. Keep this User's Manual for future reference.
2. Always read the safety information carefully.
3. Keep this equipment away from direct sunlight, or in humid or damp places.
4. Do not place this equipment in an unstable position, or on vibrating surface before setting it up.
5. Do not use or place this equipment near magnetic fields, televisions, or radios to avoid electronic interface that affects device performance.



Table of Contents

1. HARDWARE INSTALLATION.....	3
1-1 PCI Express CAN FD Card	3
1-2 M.2 PCIe CAN FD Card	4
1-3 Mini PCIe CAN FD Card.....	5
2. PIN ASSIGNMENT	6
3. TERMINATION SETTING.....	7
3-1 PCIe Series card	7
3-2 M.2 PCIe & Mini PCIe Series card.....	7
4. DRIVER INSTALLATION	8
4-1 Windows platform	8
4-2 Linux platform	12
5. DRIVER VERIFICATION	16
5-1 Windows platform	16
5-2 Linux platform	18
6. DRIVER UNINSTALL	19
6-1 Windows platform	19
6-2 Linux platform	20
7. SUNIX SDC IO MANAGER.....	22
7-1 Check Interface Card details.....	22
7-2 Check single channel Basic details	23
7-3 Check single channel Nominal Bittate details	24
7-4 Check single channel Data Bittate details	25
8. SOCKETCAN	26
8-1 Socket CAN Configuration	26
8-2 CAN2.0/FD Communication.....	28
9. TROUBLESHOOTING.....	32

1. Hardware Installation

1-1 PCI Express CAN FD Card

The hardware installation of PCI Express Industrial I/O Control Board is easy to carry out. Before inserting the card into the PCIe bus, please follow the detailed steps given below to install the board in your computer.



SAFETY FIRST

To avoid damaging to the computer, make sure to remove any power connection before card installation.

Step 1: Turn your PC's power off, and shut off the power to any peripheral.

Step 2: Remove the power plug from the plug socket.

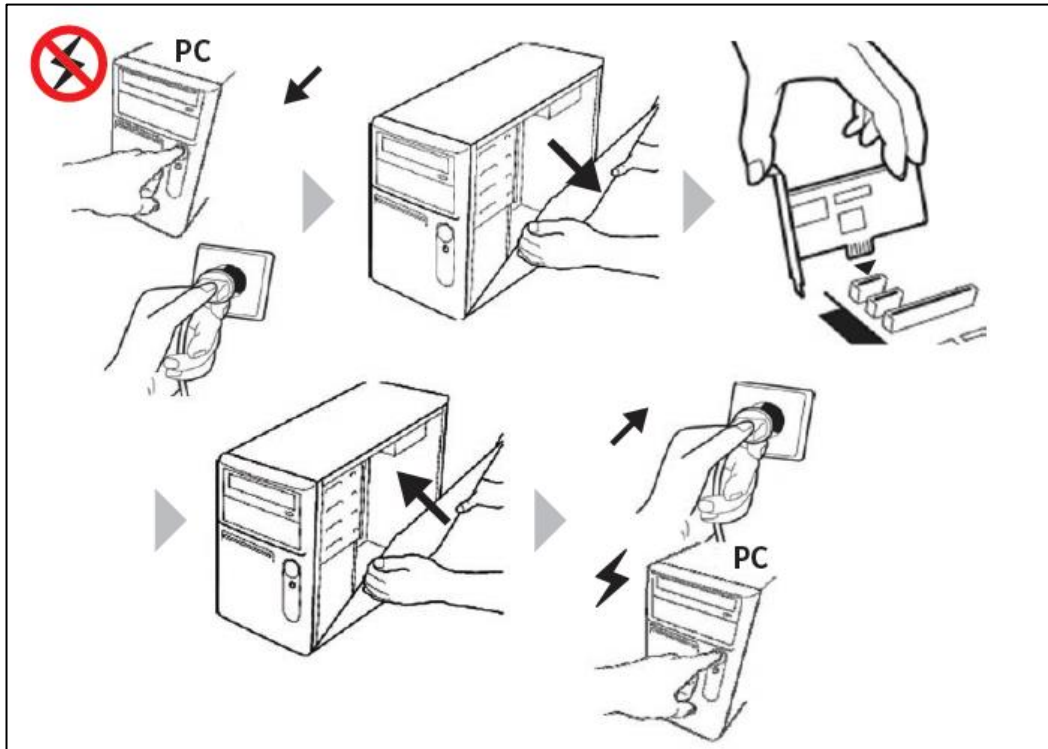
Step 3: Remove the cover from the computer case.

Step 4: If fitted. Remove the metal cover plate on the rear of a free PCIe slot.

Step 5: Insert PCI Express Industrial I/O Control Board into the free PCIe slot and screw it firmly on the bracket side.

Step 6: Place the cover back onto the computer.

Step 7: Insert the plug into the plug socket.



1-2 M.2 PCIe CAN FD Card

The hardware installation of M.2 PCI Express Industrial I/O Control Board is easy to carry out. Before inserting the card into M.2 slot, please follow the detailed steps given below to install the boards in your computer.



SAFETY FIRST

To avoid damaging to the computer, make sure to remove any power connection before card installation.

Step 1: Remove the screw attached to the motherboard.

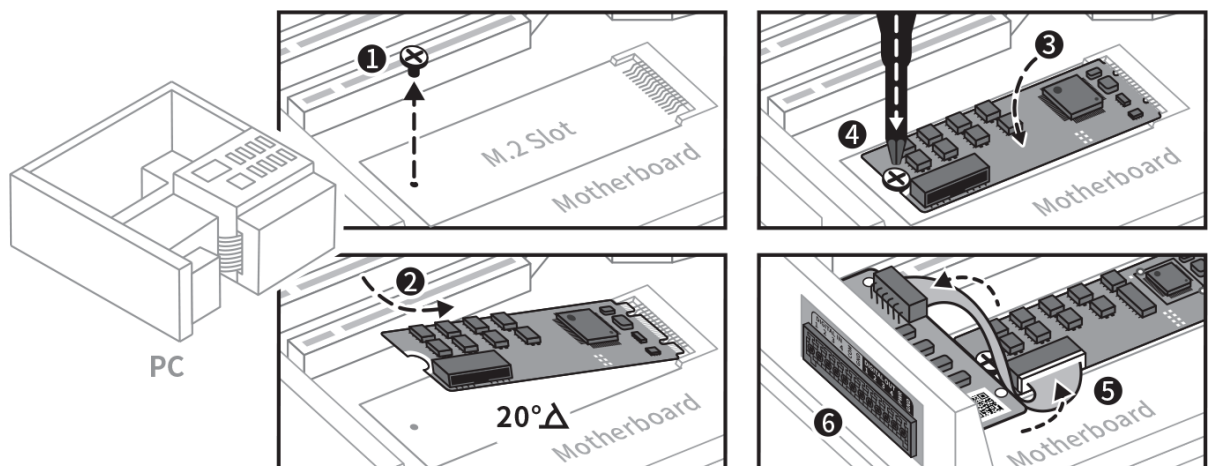
Step 2: Insert M.2 Card tightly into the M.2 slot with 20° angle. (SUNIX M.2 Card supports Key-M and Key-B slot)

Step 3: Push the M.2 card down to the screw hole.

Step 4: Lock screw on M.2 card that you remove from motherboard in step1.

Step 5: Connect ribbon cable

Step 6: Secure the extension board to PC chassis.



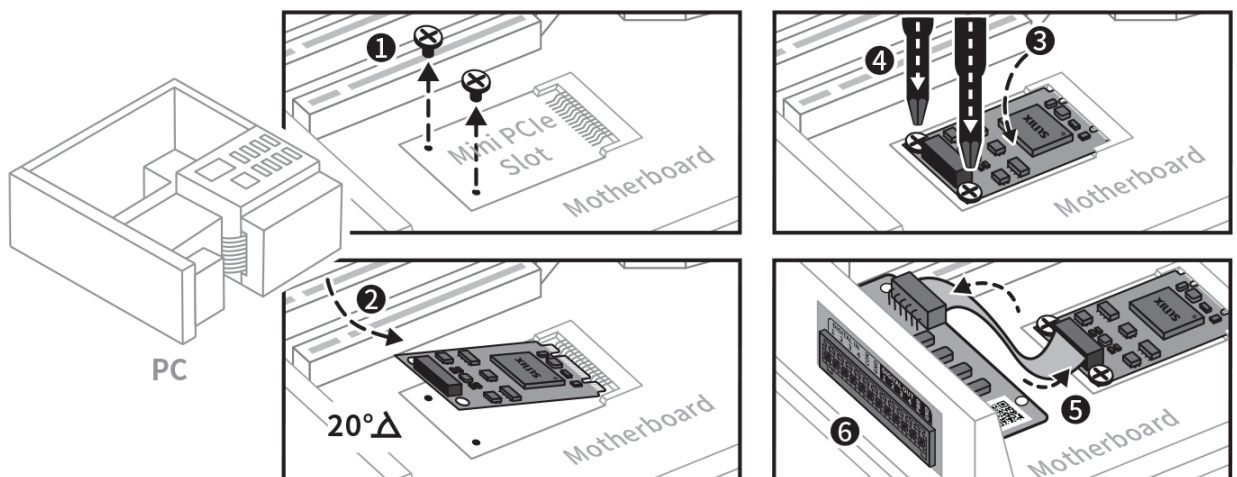
1-3 Mini PCIe CAN FD Card

The hardware installation of Mini PCI Express Industrial I/O Control Board is easy to carry out. Before inserting the card into Mini PCI-E slot, please follow the detailed steps given below to install the boards in your computer.



SAFETY FIRST To avoid damaging to the computer, make sure to remove any power connection before card installation.

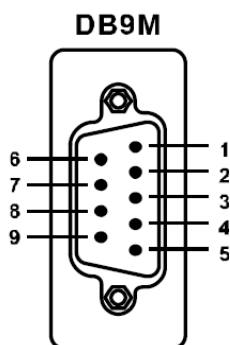
- Step 1: Remove the screw attached to the motherboard.
- Step 2: Insert Mini PCI-E Card tightly into the Mini PCI-E slot.
- Step 3: Push the Mini PCI-E card down to the screw hole.
- Step 4: Lock screw on Mini PCI-E card that you remove from motherboard in step1.
- Step 5: Connect ribbon cable
- Step 6: Secure the extension board to PC chassis.



2. Pin Assignment

PCI Express / M.2 / Mini PCIe Series

DB9 Pin define



PIN	Signal
1	-
2	CAN L
3	GND
4	-
5	-
6	GND
7	CAN H
8	-
9	-

3. Termination Setting

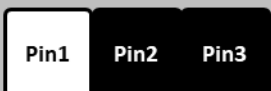
3-1 PCIe Series card

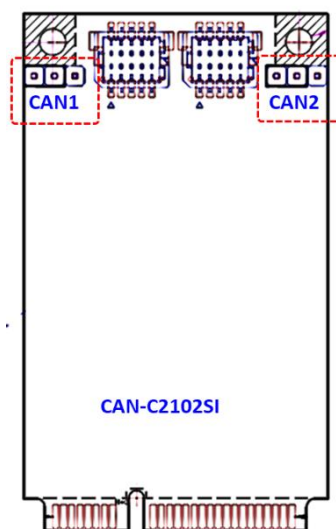
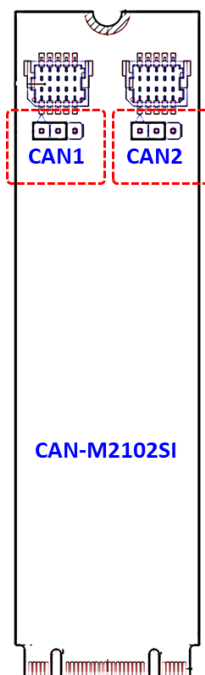
Windows platform : Termination enable by software API
(default disable)

Linux platform : Termination Always **disable**

3-2 M.2 PCIe & Mini PCIe Series card

Termination jumper default is disable

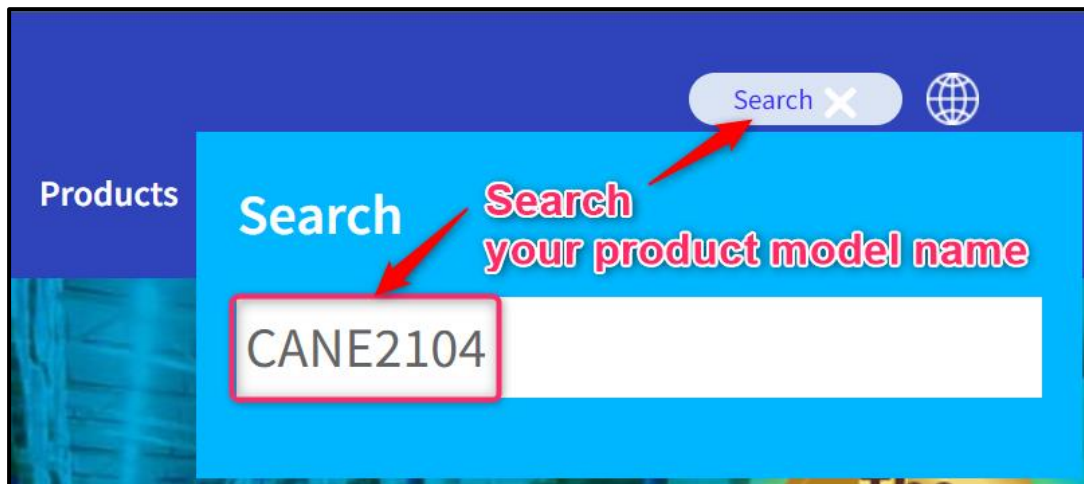
Termination Status	Jumper Setting	Description
Disable		Close Pin 1 & 2
Enable		Close Pin 2 & 3



4. Driver Installation

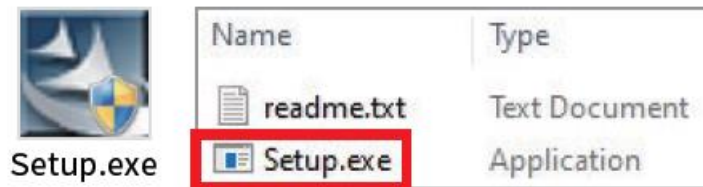
4-1 Windows platform

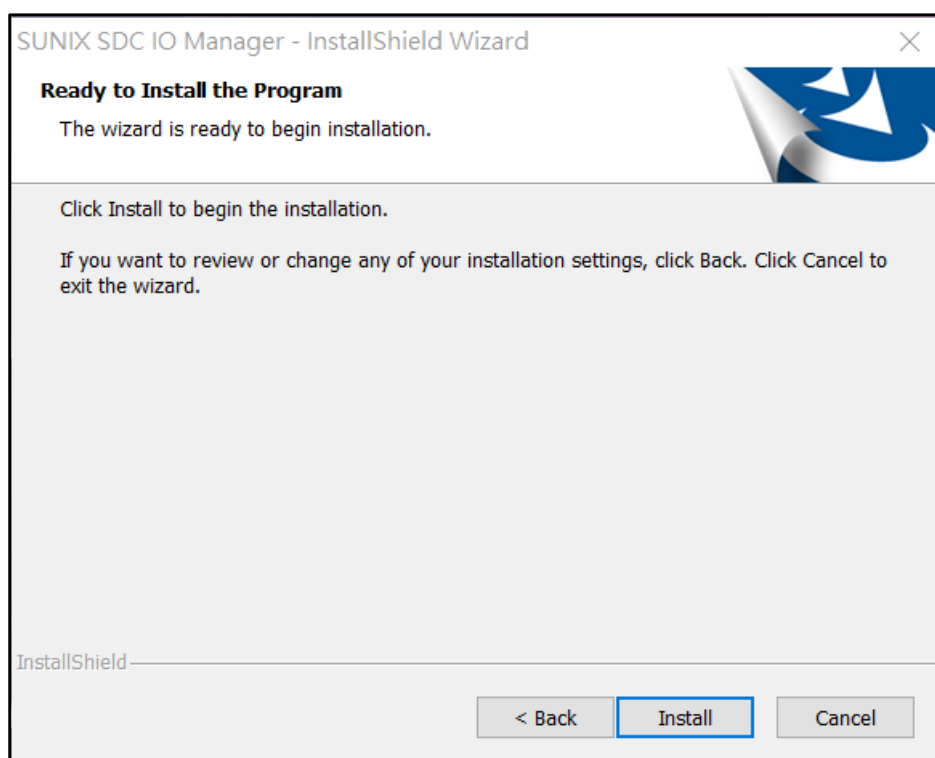
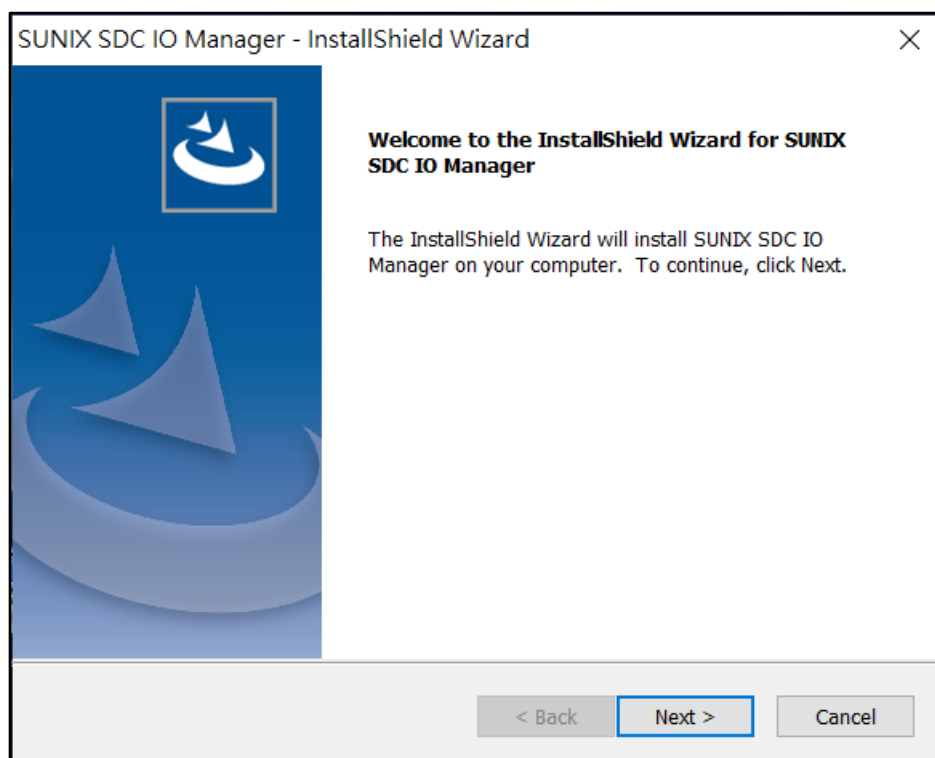
(1) You can download the latest driver from SUNIX official website (<http://www.sunix.com>)

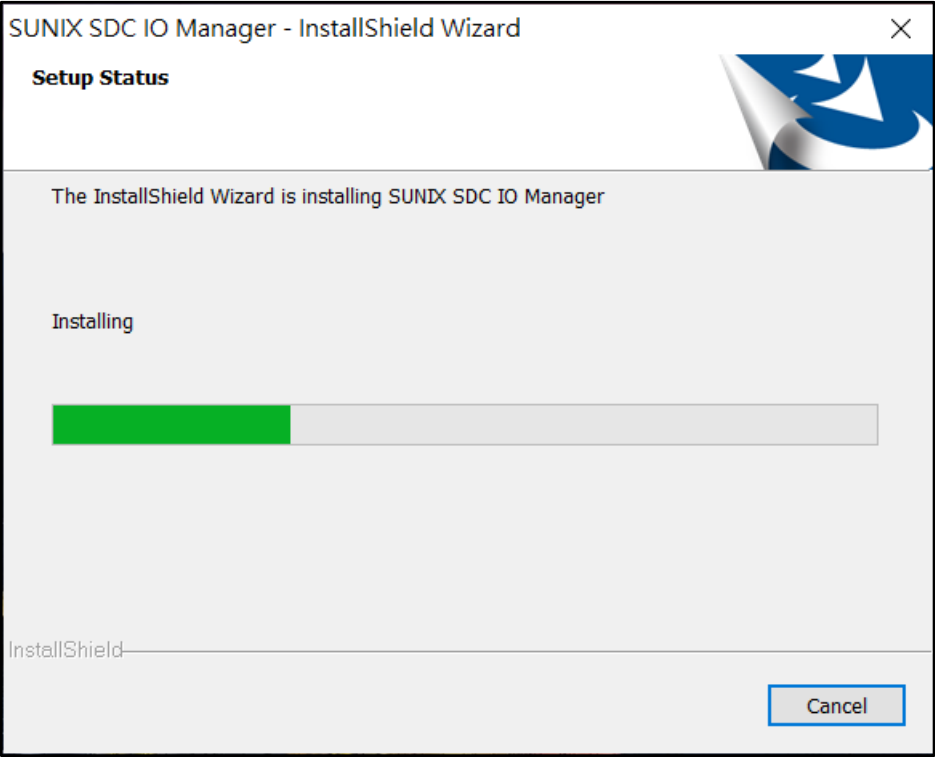


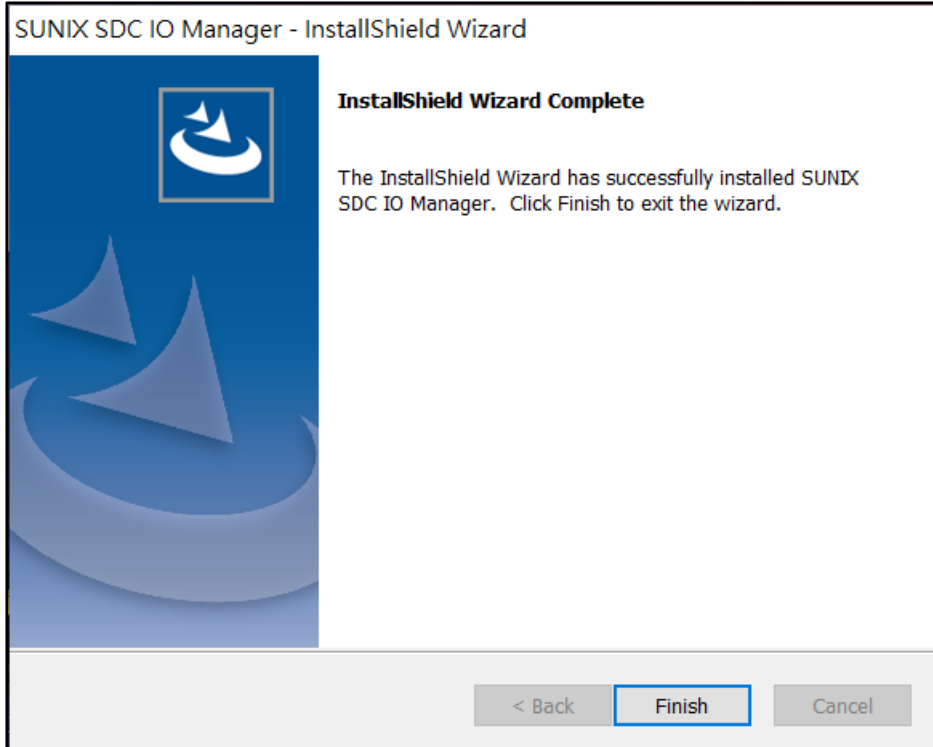
(2) Please execute Setup.exe to install driver. Driver InstallShield Wizard will show up and please wait for driver install procedure running.

Please plug the board in an available I/O slot first, before installing the driver.
Unzip the software file and run setup.exe under Windows operating system.









4-2 Linux platform

1. Driver installation

(1) Confirm the Kernel version is 4.11 or above, include 5.x and 6.x.

```
root@test-System-Product-Name: ~
root@test-System-Product-Name:~# uname -a
Linux test-System-Product-Name 6.5.0-45-generic #45~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC
Mon Jul 15 16:40:02 UTC 2 x86_64 x86_64 x86_64 GNU/Linux
root@test-System-Product-Name:~#
```

(2) Confirm that the following packages have been installed:

- make
- gcc
- gcc-12
- ncurses.h

If these packages are not installed, please confirm Chapter 9 Troubleshooting.

(3) Verify that the system detects the card by “lspci”:

```
# lspci
```

```
root@test-System-Product-Name:~# lspci -v -s '0000:05:00.0'
05:00.0 Communication controller: SUNIX Co., Ltd. Device 2000 (rev 01)
Subsystem: SUNIX Co., Ltd. Device 0001
Flags: medium devsel, IRQ 255
Memory at 84c02000 (32-bit, non-prefetchable) [disabled] [size=2K]
I/O ports at 4000 [disabled] [size=64]
Memory at 84c00000 (32-bit, non-prefetchable) [disabled] [size=8K]
Capabilities: [40] Power Management version 3
```

(4) Unzip “SDCLinuxExpansionBoardDriver_VX.X.X.X.zip”.

```
root@test-System-Product-Name:/home/test# ls
Desktop Documents Downloads Music Pictures Public SDCLinuxExpansionBoardDriver_V2.1.0.0.zip snap Templates Videos
root@test-System-Product-Name:/home/test# unzip SDCLinuxExpansionBoardDriver_V2.1.0.0.zip
Archive: SDCLinuxExpansionBoardDriver_V2.1.0.0.zip
```

(5) Enter SDCLinuxExpansionBoardDriver VX.X.X.X/driver driver” directory.

Use “make” and “make install” to build and install driver.

```
$make
```

```
root@test-System-Product-Name:/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/driver# make
Compile SDC driver. (kernel 6.5.0-45-generic)

make CFLAGS_MODULE+=-DNETCAN CFLAGS_MODULE+=-Wframe-larger-than=2048 EXTRA_CFLAGS+=-DSDC_CREATE_UART=1 -DSDC_CREATE_DIO=1 -DSDC_CREATE_SPI=1
-DSDC_CREATE_CAN=1 -DSDC_CREATE_PARPORT=0" -C /lib/modules/6.5.0-45-generic/build M=/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/driver
modules
make[1]: Entering directory '/usr/src/linux-headers-6.5.0-45-generic'
warning: the compiler differs from the one used to build the kernel
The kernel was built by: x86_64-linux-gnu-gcc-12 (Ubuntu 12.3.0-1ubuntu1~22.04) 12.3.0
You are using: gcc-12 (Ubuntu 12.3.0-1ubuntu1~22.04) 12.3.0
```

\$make install

```
root@test-System-Product-Name:/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/driver# make install
Install SDC driver. (kernel 6.5.0-45-generic)

sudo cp -f ./obj/sunix-sdc.ko /lib/modules/6.5.0-45-generic/kernel/drivers/mfd
sudo cp -f ./obj/sdc_config.ko /lib/modules/6.5.0-45-generic/kernel/drivers/char
sudo cp -f ./obj/8250_sdc.ko /lib/modules/6.5.0-45-generic/kernel/drivers/tty/serial/8250
sudo cp -f ./obj/sdc_dio.ko /lib/modules/6.5.0-45-generic/kernel/drivers/char
sudo cp -f ./obj/spi-sdc.ko /lib/modules/6.5.0-45-generic/kernel/drivers/spi
sudo cp -f ./obj/sdc_mcp251x.ko /lib/modules/6.5.0-45-generic/kernel/drivers/net/can/spi
sudo cp -f ./obj/sdc_mcp251xfd.ko /lib/modules/6.5.0-45-generic/kernel/drivers/net/can/spi
sudo cp -f ./obj/can_sdc.ko /lib/modules/6.5.0-45-generic/kernel/drivers/net/can
sudo depmod -a
```

(6) Reboot system, confirm the module is mounted

\$lspci -v -s "0000:xx:xx.x"

```
root@test-System-Product-Name:~# lspci -v -s '0000:05:00.0'
05:00.0 Communication controller: SUNIX Co., Ltd. Device 2000 (rev 01)
Subsystem: SUNIX Co., Ltd. Device 0001
Flags: medium devsel, IRQ 16
Memory at 84c02000 (32-bit, non-prefetchable) [size=2K]
I/O ports at 4000 [size=64]
Memory at 84c00000 (32-bit, non-prefetchable) [size=8K]
Capabilities: [40] Power Management version 3
Kernel driver in use: sunix_sdc
Kernel modules: sunix_sdc
```

(7) Check the SDC device nodes in /dev dev.

```
root@test-System-Product-Name:~# ls /dev
autofs      dri          hidraw4     i2c-4       loop1       men          nvme0n1p2  rtc          tty          tty19       tty3        tty40       tty51       tty62       ttyS14      ttyS25      ttyS8
block       drm_dp_aux0  hpet        i2c-5       loop2       queue        nvram       rtc          tty0         tty2        tty30       tty41       tty52       tty63       ttyS15      ttyS26      ttyS9
btrfs-control ecryptfs     hugepages   i2c-6       loop3       mtd0         port        sdc_config6 tty10        tty20       tty31       tty42       tty53       tty7        ttyS16      ttyS27      udmabuf
bus         fb0          hwrng       i2c-7       loop4       mtd0ro       ppp         sdc_dio0     tty11        tty21       tty32       tty43       tty54       tty8        ttyS17      ttyS28      uhid
char        fd           i2c-c0      i2c-8       loop5       mtd1         pps0        shm          tty12        tty22       tty33       tty44       tty55       tty9        ttyS18      ttyS29      uinput
console     full         i2c-c1      i2c-9       loop6       mtdi         ppsaux      snd          tty13        tty23       tty34       tty45       tty56       ttyprintk   ttyS19      ttyS30      urandom
core        fuse         i2c-10      initctl     loop7       net          ptp0        snapshot     tty14        tty24       tty35       tty46       tty57       ttyS0        ttyS2       ttyS30      usb
cpu         gpiochip0    i2c-11      input       loop8       ng0n1        ptp1        snd          tty15        tty25       tty36       tty47       tty58       ttyS1        ttyS20      ttyS31      userfaultfd
cpu_dma_latency hidraw0      i2c-12      kmsg        loop-control null          ptp1        stderr       tty16        tty26       tty37       tty48       tty59       ttyS10       ttyS21      ttyS32      userio
cuse        hidraw1      i2c-13      kvm          mapper      nvme0        pts         stdout       tty17        tty27       tty38       tty49       tty60       ttyS11       ttyS22      ttyS33      vcs
disk        hidraw2      i2c-2       log          mcelog      nvme0n1      random      tpm0         tty18        tty28       tty39       tty50       tty61       ttyS12       ttyS23      ttyS34      vcs1
dma_heap    hidraw3      i2c-3       loop0       mei0        nvme0n1p1    rfkill      tpmrm0       tty19        tty29       tty40       tty51       tty62       ttyS13       ttyS24      ttyS35      vcs2
```

2. Tool installation

- (1) Complete driver installation.
- (2) Enter “./SDCLinuxExpansionBoardDriverVX.X.X.X/tools tools” directory.
- (3) Enter the following commands in order.

```
$cp /x86-64/sdc-config /usr/sbin or cp /arm64/sdc-config /usr/sbin
$cp /x86-64/sdc-uart-config /usr/sbin or cp /arm64/sdc-uart-config /usr/sbin
```

```
root@test-System-Product-Name:/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/tools# ls
arm64  x86-64
root@test-System-Product-Name:/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/tools# cp x86-64/sdc-config /usr/sbin/
root@test-System-Product-Name:/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/tools# cp x86-64/sdc-uart-cfg /usr/sbin/
root@test-System-Product-Name:/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/tools#
```

- (4) Enter “./SDCLinuxExpansionBoardDriverVX.X.X.X/script/Ubuntu”, or
“./SDCLinuxExpansionBoardDriverVX.X.X.X/script/Kylin” directory

```
$cp sdc-board-cfg.sh /etc/init.d
$update-rc.d sdc-board-cfg.sh defaults
$/etc/init.d/sdc-board-cfg.sh start
$sdc-uart-cfg -s
```

```
root@test-System-Product-Name:/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/script/Ubuntu# cp sdc-board-cfg.sh /etc/init.d/
root@test-System-Product-Name:/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/script/Ubuntu# update-rc.d sdc-board-cfg.sh defaults
root@test-System-Product-Name:/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/script/Ubuntu# /etc/init.d/sdc-board-cfg.sh start
Starting sdc-board-cfg.sh (via systemctl): sdc-board-cfg.service.
root@test-System-Product-Name:/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/script/Ubuntu# sdc-uart-cfg -s
SAVE line=4;flags=00000000;
SAVE line=5;flags=00000000;
SAVE line=6;flags=00000013;
SAVE line=7;flags=00000013;
```

3. Terminal tool snxterm installation

- (1) Enter “./SDCLinuxExpansionBoardDriverVX.X.X.X/snxterm” directory.
- (2) Use “make install” to build and install snxterm.

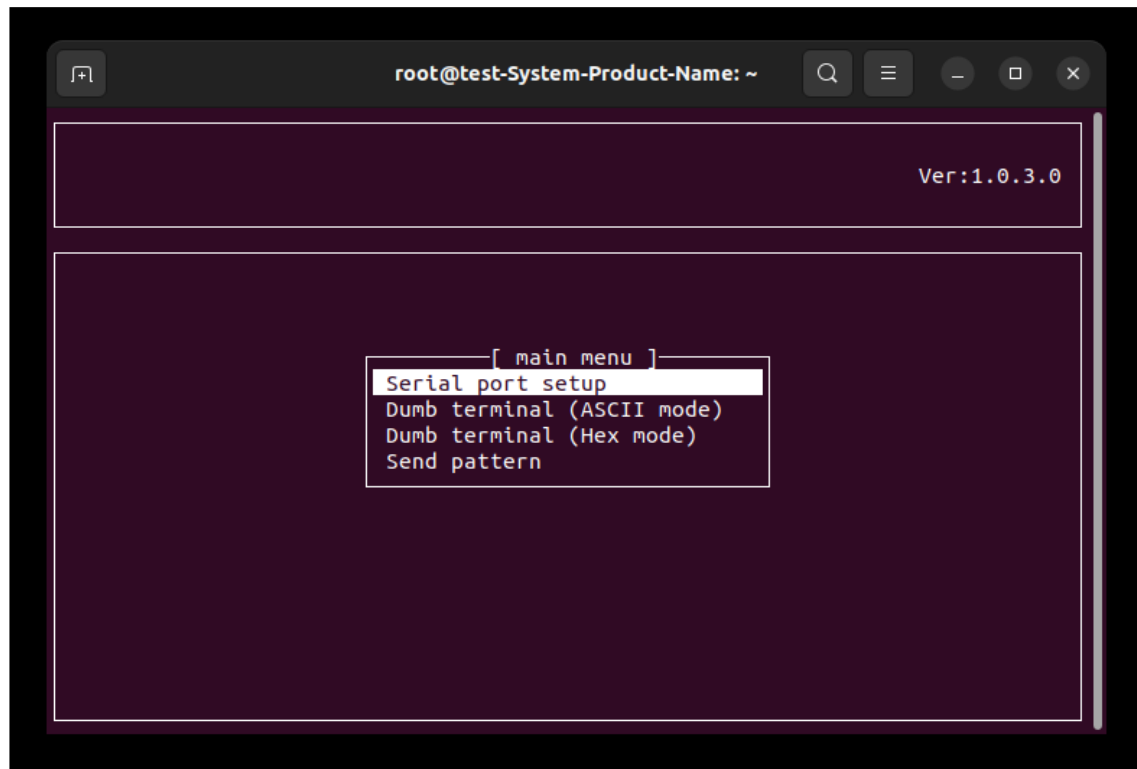
```
$make install
```

```
root@test-System-Product-Name:/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/snxterm# make install
gcc -O2 -pthread -pipe -o snxterm snxterm.c -lnurses
cp -p snxterm /usr/bin
```

- (3) snxterm is a terminal tool, the tool could test serial port communication,

Enter “snxterm” to execute the application.

\$snxterm



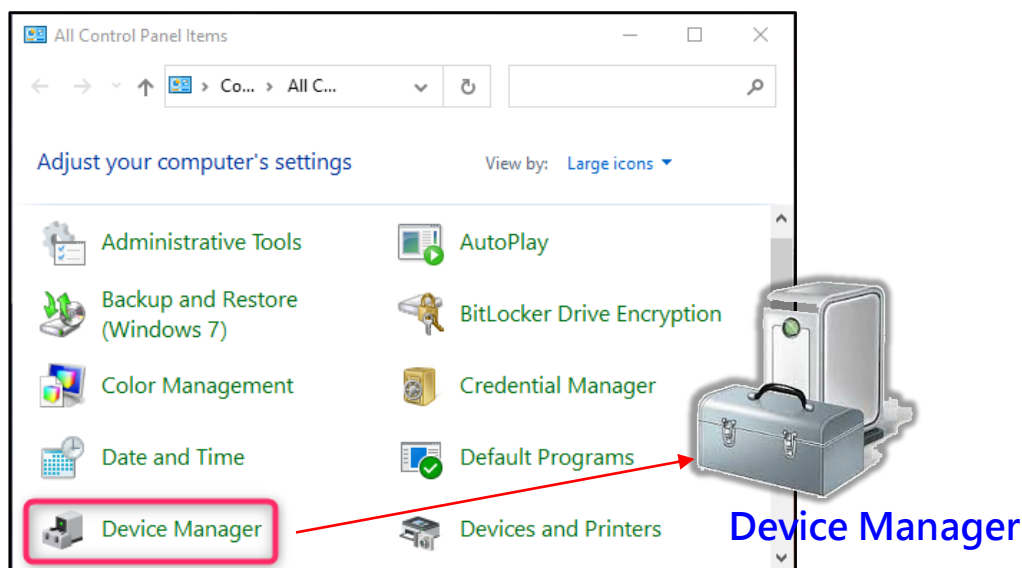
5.Driver Verification

5-1 Windows platform

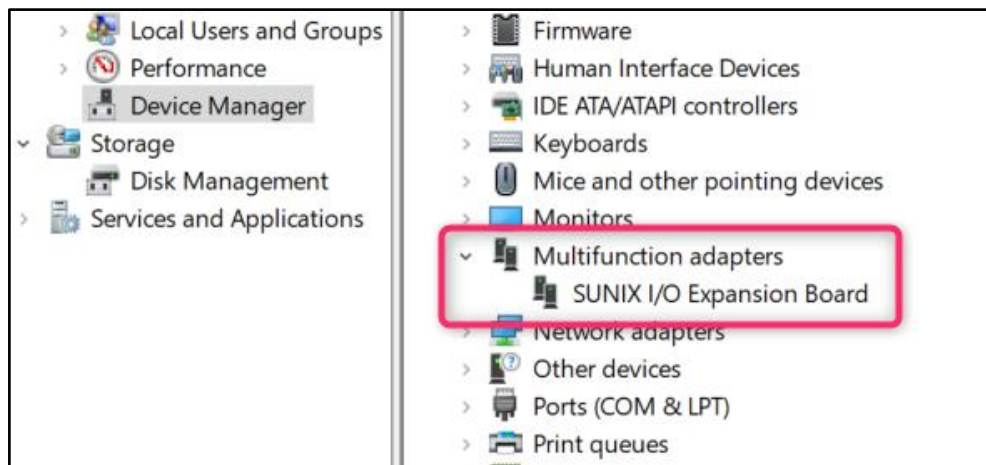
You can use Windows “**Device Manager**” to verify proper installation.

(1) Select Device Manager in the in the Windows Control Panel.

Controller Panel > All Control Panel Items > Device Manager



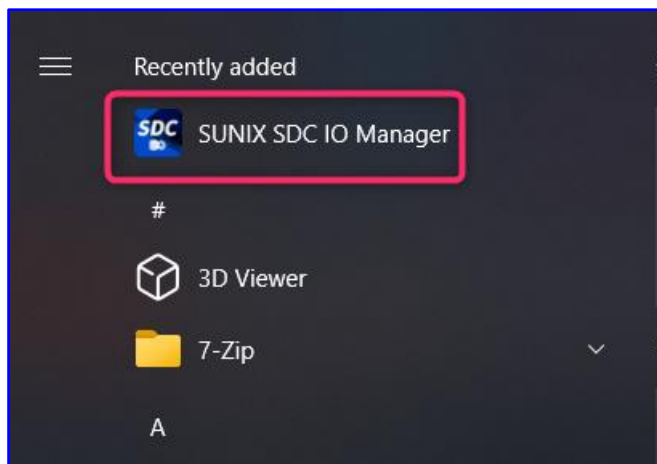
(2) In the Device Manager window, you would read **SUNIX I/O Expansion Board** under **Multifunction adapters** catalog



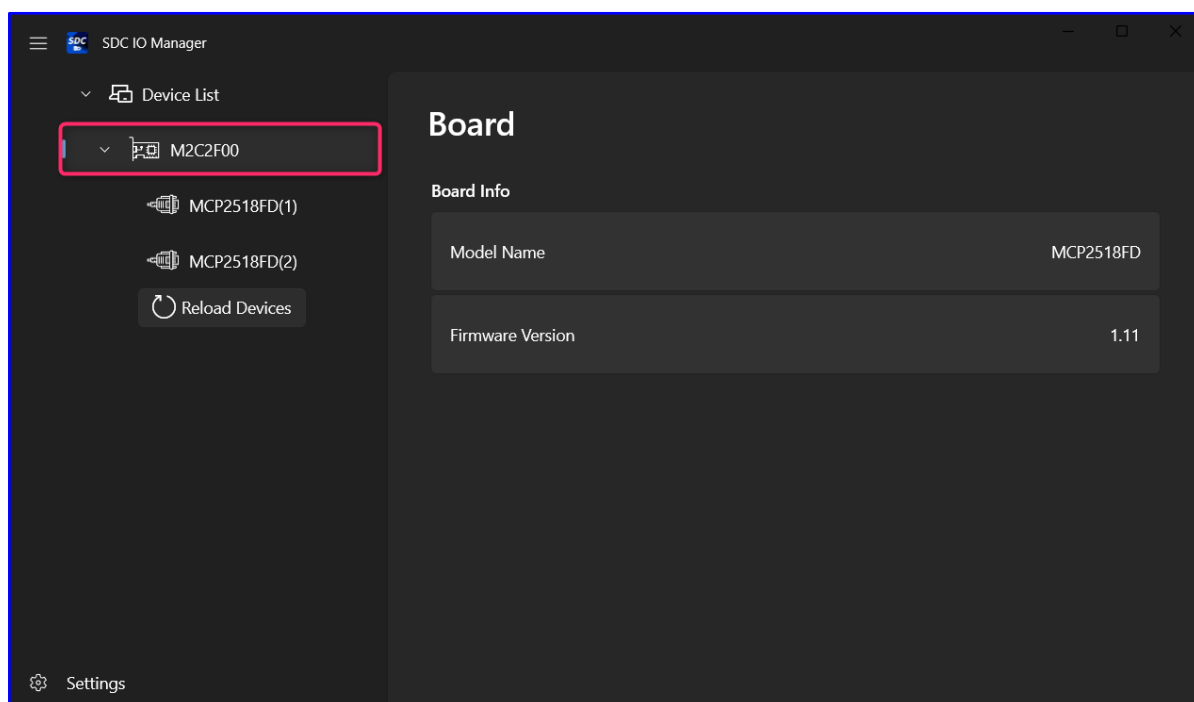
(3) Execute “**SUNIX SDC IO Manager**”.



Start > SUNIX SDC IO Manager



Check the “**Device List window**”. If the interface card appears, it means the driver has been installed successfully.



5-2 Linux platform

Reboot system and check the module is mounted.

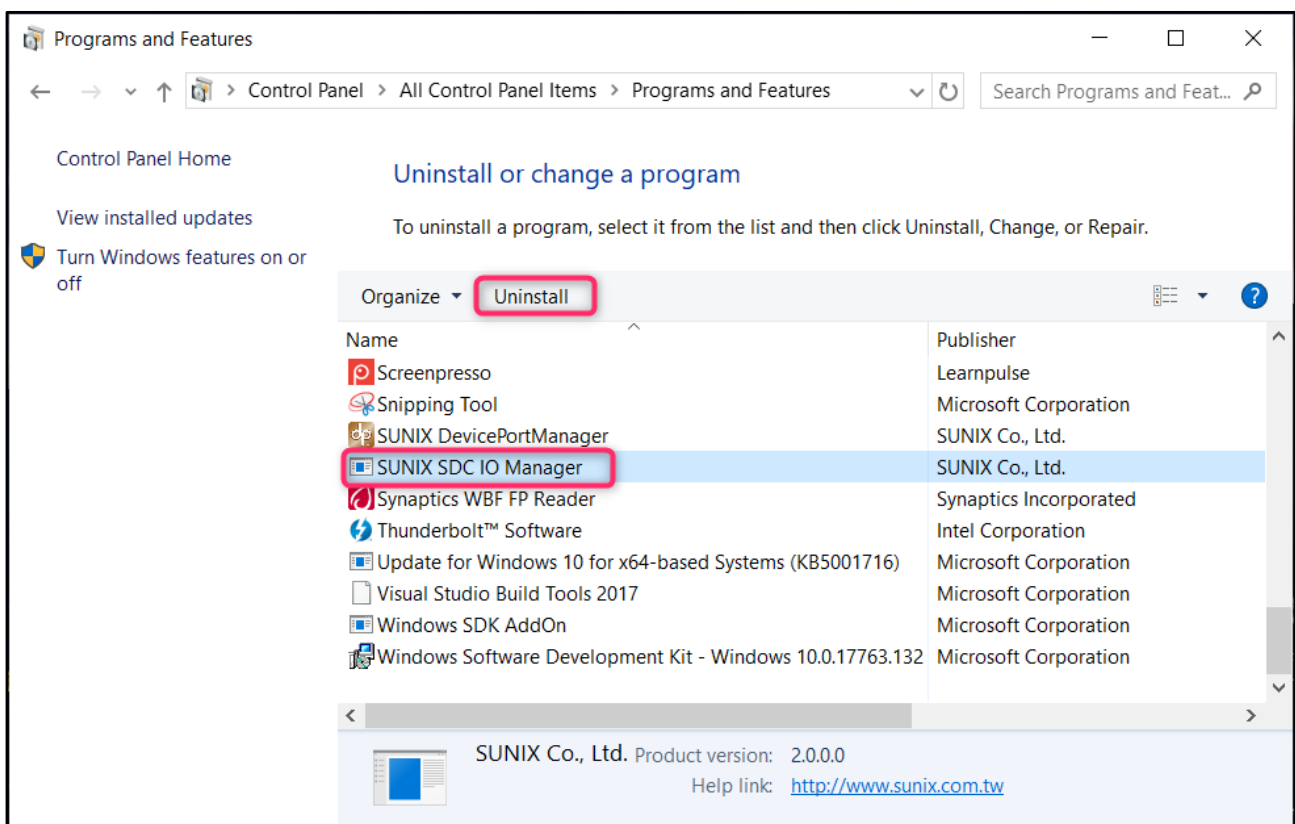
```
# lspci -v
```

```
root@test-System-Product-Name:~# lspci -v -s '0000:05:00.0'
05:00.0 Communication controller: SUNIX Co., Ltd. Device 2000 (rev 01)
Subsystem: SUNIX Co., Ltd. Device 0001
Flags: medium devsel, IRQ 16
Memory at 84c02000 (32-bit, non-prefetchable) [size=2K]
I/O ports at 4000 [size=64]
Memory at 84c00000 (32-bit, non-prefetchable) [size=8K]
Capabilities: [40] Power Management version 3
Kernel driver in use: sunix_sdc
Kernel modules: sunix_sdc
```

6. Driver Uninstall

6-1 Windows platform

- (1) Open Control Panel -> Programs and Features.
- (2) Select "SUNIX SDC IO Manager".
- (3) Click "Uninstall" to uninstall driver.



6-2 Linux platform

1. Driver uninstallation

- (1) Enter “./SDCLinuxExpansionBoardDriverVX.X.X.X/driver” directory.
- (2) Close all driver device node (/dev/sdc_dio0, /dev/ttyS4... etc.).
- (3) Use “make uninstall” to uninstall driver.

```
$make uninstall
```

```
root@test-System-Product-Name:/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/driver# make uninstall
Uninstall SDC driver. (kernel 6.5.0-45-generic)

sudo rm -f /lib/modules/6.5.0-45-generic/kernel/drivers/mfd/sunix-sdc.ko
sudo rm -f /lib/modules/6.5.0-45-generic/kernel/drivers/char/sdc_config.ko
sudo rm -f /lib/modules/6.5.0-45-generic/kernel/drivers/tty/serial/8250/8250_sdc.ko
sudo rm -f /lib/modules/6.5.0-45-generic/kernel/drivers/char/sdc_dio.ko
sudo rm -f /lib/modules/6.5.0-45-generic/kernel/drivers/spi/spi-sdc.ko
sudo rm -f /lib/modules/6.5.0-45-generic/kernel/drivers/net/can/spi/sdc_mcp251x.ko
sudo rm -f /lib/modules/6.5.0-45-generic/kernel/drivers/net/can/spi/sdc_mcp251xfd.ko
sudo rm -f /lib/modules/6.5.0-45-generic/kernel/drivers/net/can/can_sdc.ko
sudo depmod -a
```

- (4) Use “make clean” to remove all built files.

```
$make clean
```

```
root@test-System-Product-Name:/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/driver# make clean
Clean SDC driver. (kernel 6.5.0-45-generic)

make -C /lib/modules/6.5.0-45-generic/build M=/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/driver clean
make[1]: Entering directory '/usr/src/linux-headers-6.5.0-45-generic'
  CLEAN   /home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/driver/Module.symvers
make[1]: Leaving directory '/usr/src/linux-headers-6.5.0-45-generic'
rm -Rf .tmp_versions
rm -f .cache.*
```

2. Tool uninstallation

- (1) Complete driver uninstallation.
- (2) Enter the following commands in order to uninstall tools.

```
$/etc/init.d/sdc-board-cfg.sh stop
$update-rc.d -f sdc-board-cfg.sh remove
$rm /etc/init.d/sdc-board-cfg.sh
$rm -Rf /usr/share/sdc_8250/
$rm /usr/sbin/sdc-uart-cfg
$rm /usr/sbin/sdc-config
```

```
root@test-System-Product-Name:~# /etc/init.d/sdc-board-cfg.sh stop
Stopping sdc-board-cfg.sh (via systemctl): sdc-board-cfg.service.
root@test-System-Product-Name:~# update-rc.d -f sdc-board-cfg.sh remove
root@test-System-Product-Name:~# rm /etc/init.d/sdc-board-cfg.sh
root@test-System-Product-Name:~# rm -Rf /usr/share/sdc_8250/
```

```
root@test-System-Product-Name:~# rm /usr/sbin/sdc-config
root@test-System-Product-Name:~# rm /usr/sbin/sdc-uart-cfg
```

3. Terminal tool snxterm uninstallation

- (1) Enter “./SDCLinuxExpansionBoardDriverVX.X.X.X/snxterm” directory.
- (2) Complete tools uninstallation.
- (3) Enter the following command to uninstall snxterm.

```
$make clean
```

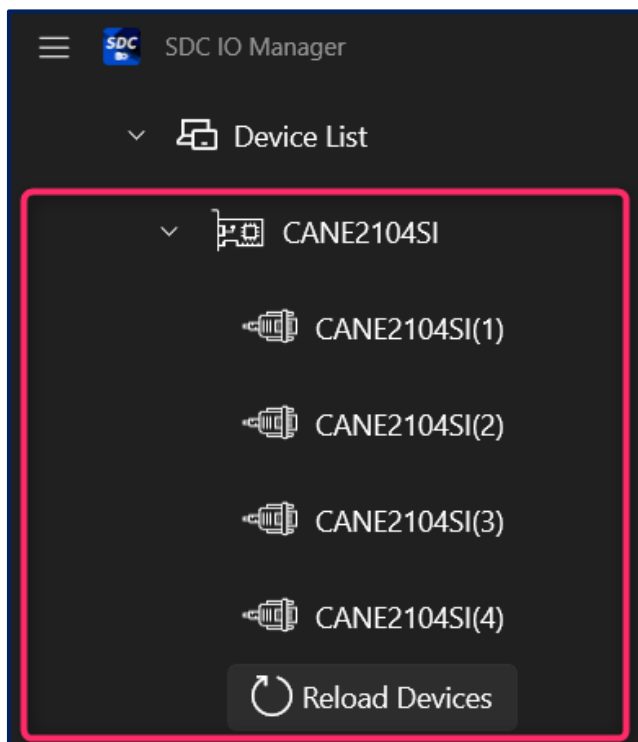
```
root@test-System-Product-Name:/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/snxterm# make clean
rm -f snxterm
rm -f *~ *.o
rm -f /usr/bin/snxterm
```

7. SUNIX SDC IO Manager

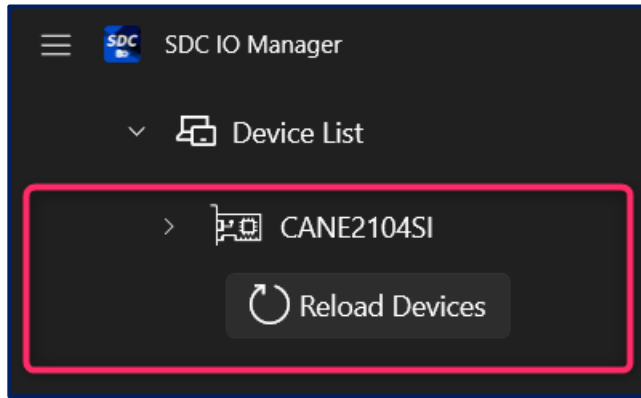
7-1 Check Interface Card details

After clicking on the interface card, the **Model Name** and **Firmware Version** will appear in the right window.

Click  : you can to expand communication port

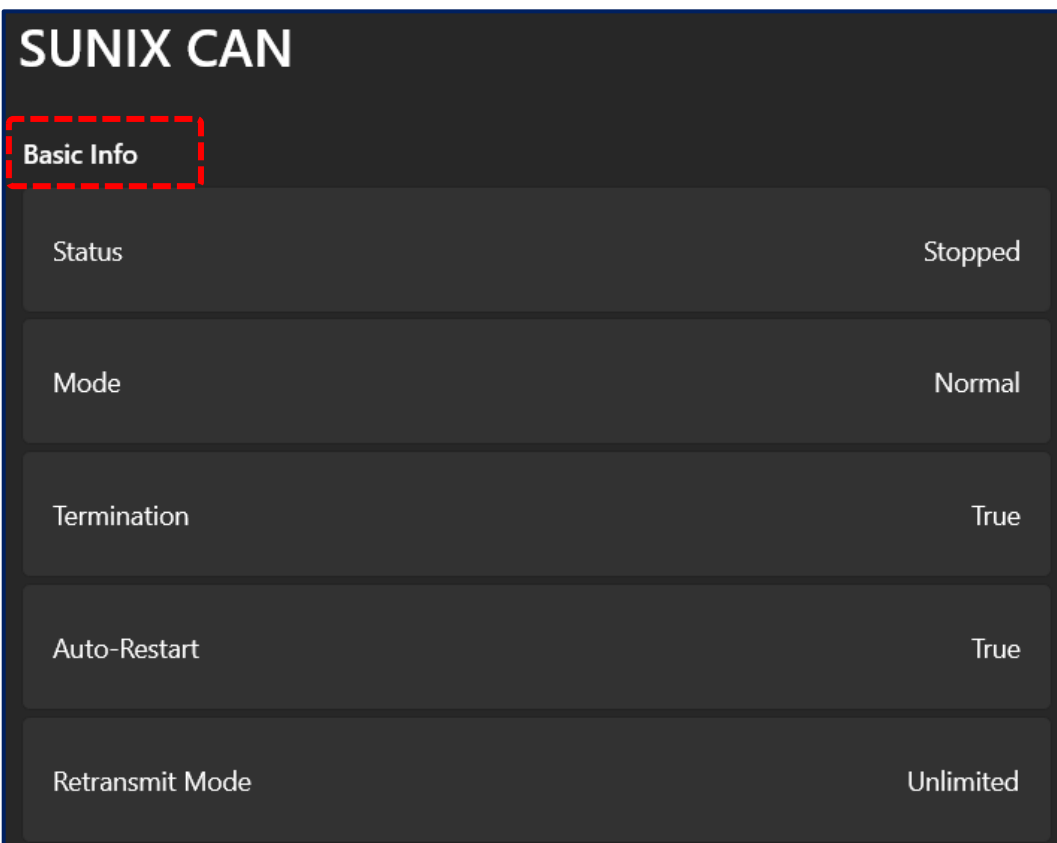
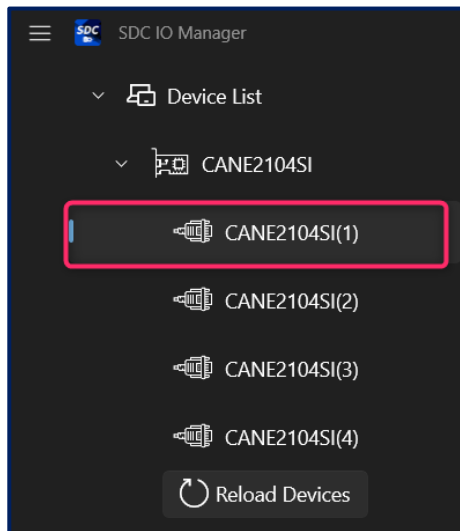


Click  : you can to collapse the communication port



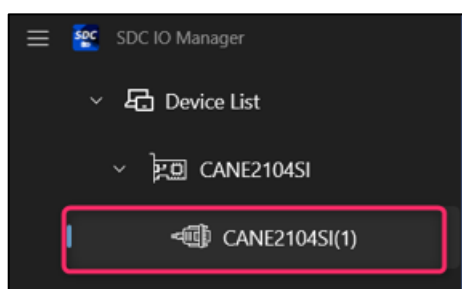
7-2 Check single channel Basic details

Click single channel port,
drag the right window scroll to view **Basic** information.



7-3 Check single channel Nominal Bitrate details

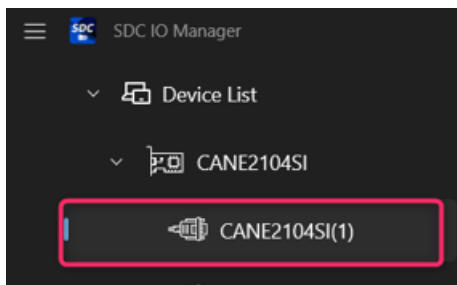
Click single channel port,
drag the right window scroll down to view **Nominal Bitrate** information.



SUNIX CAN	
Nominal Bitrate Info	
Bitrate	1000000
Sample Point	0
SJW, Synchronization Jump Width	0
Error	0
TQ, Time Quantum	0
Phase Buffer Segment 1	0
Phase Buffer Segment 2	0

7-4 Check single channel Data Bitrate details

Click single channel port,
drag the right window scroll down to view **Data Bitrate** information.



SUNIX CAN	
Data Bitrate Info	
Bitrate	2000000
Sample Point	0
SJW, Synchronization Jump Width	0
Error	0
TQ, Time Quantum	0
Phase Buffer Segment 1	0
Phase Buffer Segment 2	0

8. SocketCAN

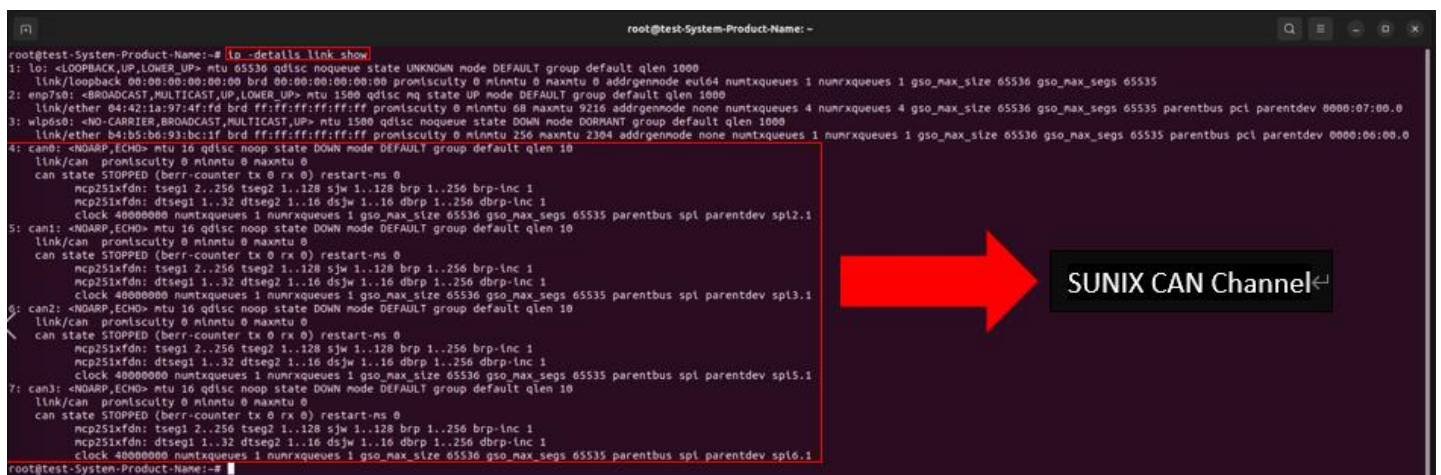
Confirm that the following packages have been installed

- Iproute2
- Can-utils

8-1 Socket CAN Configuration

1). Check all SocketCAN devices details.

\$ip -details link show



```

root@test-System-Product-Name:~# ip -details link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00 promiscuity 0 minmtu 0 maxmtu 0 addrgenmode eui64 numrxqueues 1 numtxqueues 1 gso_max_size 65536 gso_max_segs 65535
2: empi0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group default qlen 1000
    link/ether 04:42:1a:97:4f:1d brd ff:ff:ff:ff:ff:ff promiscuity 0 minmtu 68 maxmtu 9216 addrgenmode none numrxqueues 4 gso_max_size 65536 gso_max_segs 65535 parentbus pci parentdev 0000:07:00.0
3: wlpes0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DORMANT group default qlen 1000
    link/ether b4:b5:b6:93:bc:1f brd ff:ff:ff:ff:ff:ff promiscuity 0 minmtu 256 maxmtu 2304 addrgenmode none numrxqueues 1 numtxqueues 1 gso_max_size 65536 gso_max_segs 65535 parentbus pci parentdev 0000:06:00.0
4: can0: <NOARP,ECHO> mtu 16 qdisc noop state DOWN mode DEFAULT group default qlen 10
    link/can promiscuity 0 minmtu 0 maxmtu 0
    can state STOPPED (berr-counter tx 0 rx 0) restart-ms 0
    ncp251xfd: tseg1 2..256 tseg2 1..128 sjw 1..128 brp 1..256 brp-inc 1
    ncp251xfd: dtseg1 1..32 dtseg2 1..16 dsjw 1..16 dbrp 1..256 dbrp-inc 1
    clock 400000000 numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535 parentbus spi parentdev spi2.1
5: can1: <NOARP,ECHO> mtu 16 qdisc noop state DOWN mode DEFAULT group default qlen 10
    link/can promiscuity 0 minmtu 0 maxmtu 0
    can state STOPPED (berr-counter tx 0 rx 0) restart-ms 0
    ncp251xfd: tseg1 2..256 tseg2 1..128 sjw 1..128 brp 1..256 brp-inc 1
    ncp251xfd: dtseg1 1..32 dtseg2 1..16 dsjw 1..16 dbrp 1..256 dbrp-inc 1
    clock 400000000 numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535 parentbus spi parentdev spi3.1
6: can2: <NOARP,ECHO> mtu 16 qdisc noop state DOWN mode DEFAULT group default qlen 10
    link/can promiscuity 0 minmtu 0 maxmtu 0
    can state STOPPED (berr-counter tx 0 rx 0) restart-ms 0
    ncp251xfd: tseg1 2..256 tseg2 1..128 sjw 1..128 brp 1..256 brp-inc 1
    ncp251xfd: dtseg1 1..32 dtseg2 1..16 dsjw 1..16 dbrp 1..256 dbrp-inc 1
    clock 400000000 numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535 parentbus spi parentdev spi5.1
7: can3: <NOARP,ECHO> mtu 16 qdisc noop state DOWN mode DEFAULT group default qlen 10
    link/can promiscuity 0 minmtu 0 maxmtu 0
    can state STOPPED (berr-counter tx 0 rx 0) restart-ms 0
    ncp251xfd: tseg1 2..256 tseg2 1..128 sjw 1..128 brp 1..256 brp-inc 1
    ncp251xfd: dtseg1 1..32 dtseg2 1..16 dsjw 1..16 dbrp 1..256 dbrp-inc 1
    clock 400000000 numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535 parentbus spi parentdev spi6.1
root@test-System-Product-Name:~#
  
```

2). Check single channel details, at this time, bus state should be 'STOPPED'.

\$ip -details link show [device]

```
root@test-System-Product-Name:~# ip -details link show can0
4: can0: <NOARP,ECHO> mtu 16 qdisc noop state DOWN mode DEFAULT group default qlen 10
    link/can promiscuity 0 minmtu 0 maxmtu 0
    can state STOPPED (berr-counter tx 0 rx 0) restart-ms 0
    mcp251xfd: tseg1 2..256 tseg2 1..128 sjw 1..128 brp 1..256 brp-inc 1
    mcp251xfd: dtseg1 1..32 dtseg2 1..16 dsjw 1..16 dbrp 1..256 dbrp-inc 1
    clock 4000000 numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535 parentbus spi parentdev spi2.1
root@test-System-Product-Name:~#
```

3). Set CAN 2.0 or Set CAN FD.

✚ CAN2.0: Set can channel up and set arbitration bitrate.

```
$ip link set [channel] up type can bitrate [arbitration bitrate]
```

```
root@test-System-Product-Name:~# ip link set can0 up type can bitrate 1000000
```

➤ CAN2.0 arbitration bitrate $\leq$ 1Mbps

✚ CAN FD: Set can channel up, set arbitration bitrate and data bitrate, set FD on.

```
$ip link set [channel] up type can bitrate [arbitration bitrate] dbitrte [data bitrate] fd on
```

```
root@test-System-Product-Name:~# ip link set can1 up type can bitrate 1000000 dbitrte 8000000 fd on
```

➤ 8Mbps $\geq$ data bitrate $>$ arbitration bitrate

4). Make sure the bus status is ERROR-ACTIVE and bitrate configuration is correct.

```
$ip -details link show
```

```
root@test-System-Product-Name:~# ip -details link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00 promiscuity 0 minmtu 0 maxmtu 0 addrgenmode eui64 numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535
2: enp5s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group default qlen 1000
    link/ether 04:42:1a:97:4f:fd brd ff:ff:ff:ff:ff:ff promiscuity 0 minmtu 68 maxmtu 9216 addrgenmode none numtxqueues 4 numrxqueues 4 gso_max_size 65536 gso_max_segs 65535 parentbus pci parentdev 0000:06:00.0
3: can0: <NOARP,UP,LOWER_UP,ECHO> mtu 16 qdisc pfifo_fast state UP mode DEFAULT group default qlen 10
    link/can promiscuity 0 minmtu 0 maxmtu 0
    can state ERROR-ACTIVE (berr-counter tx 0 rx 0) restart-ms 0
    bitrate 1000000 sample-point 0.750
    tq 25 prop-seg 14 phase-seg1 15 phase-seg2 10 sjw 1
    mcp251xfd: tseg1 2..256 tseg2 1..128 sjw 1..128 brp 1..256 brp-inc 1
    mcp251xfd: dtseg1 1..32 dtseg2 1..16 dsjw 1..16 dbrp 1..256 dbrp-inc 1
    clock 4000000 numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535 parentbus spi parentdev spi2.1
4: can1: <NOARP,UP,LOWER_UP,ECHO> mtu 72 qdisc pfifo_fast state UP mode DEFAULT group default qlen 10
    link/can promiscuity 0 minmtu 0 maxmtu 0
    can <FD> state ERROR-ACTIVE (berr-counter tx 0 rx 0) restart-ms 0
    bitrate 1000000 sample-point 0.750
    tq 25 prop-seg 14 phase-seg1 15 phase-seg2 10 sjw 1
    mcp251xfd: tseg1 2..256 tseg2 1..128 sjw 1..128 brp 1..256 brp-inc 1
    dbitrte 8000000 dsample-point 0.600
    dtq 25 dprop-seg 1 dphase-seg1 1 dphase-seg2 2 dsjw 1
    mcp251xfd: dtseg1 1..32 dtseg2 1..16 dsjw 1..16 dbrp 1..256 dbrp-inc 1
    clock 4000000 numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535 parentbus spi parentdev spi3.1
5: wlp5s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DORMANT group default qlen 1000
    link/ether b4:b5:b6:93:bc:1f brd ff:ff:ff:ff:ff:ff promiscuity 0 minmtu 256 maxmtu 2304 addrgenmode none numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535 parentbus pci parentdev 0000:05:00.0
root@test-System-Product-Name:~#
```

5). Set can channel down

```
$ip link set [channel] down
```

```
root@test-System-Product-Name:~# ip link set can0 down
```

6). Make sure the bus status is STOPPED.

```
$ip -details link show
```

```
3: can0: <NOARP,ECHO> mtu 16 qdisc pfifo_fast state DOWN mode DEFAULT group default qlen 10
    link/can  promiscuity 0 minmtu 0 maxmtu 0
    can state STOPPED (berr-counter tx 0 rx 0) restart-ms 0
        bitrate 1000000 sample-point 0.750
        tq 25 prop-seg 14 phase-seg1 15 phase-seg2 10 sjw 1
        mcp251xfd: tseg1 2..256 tseg2 1..128 sjw 1..128 brp 1..256 brp-inc 1
        mcp251xfd: dtseg1 1..32 dtseg2 1..16 dsjw 1..16 dbrp 1..256 dbrp-inc 1
        clock 40000000 numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535 parentbus spi parentdev spi2.1
4: can1: <NOARP,ECHO> mtu 72 qdisc pfifo_fast state DOWN mode DEFAULT group default qlen 10
    link/can  promiscuity 0 minmtu 0 maxmtu 0
    can <FD> state STOPPED (berr-counter tx 0 rx 0) restart-ms 0
        bitrate 1000000 sample-point 0.750
        tq 25 prop-seg 14 phase-seg1 15 phase-seg2 10 sjw 1
        mcp251xfd: tseg1 2..256 tseg2 1..128 sjw 1..128 brp 1..256 brp-inc 1
        dbitrates 8000000 dsample-point 0.600
        dtq 25 dprop-seg 1 dphase-seg1 1 dphase-seg2 2 dsjw 1
        mcp251xfd: dtseg1 1..32 dtseg2 1..16 dsjw 1..16 dbrp 1..256 dbrp-inc 1
        clock 40000000 numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535 parentbus spi parentdev spi3.1
```

8-2 CAN2.0/FD Communication

can-utils is a public can bus test utility, The document just introduce 'cansend' and 'candump', The two commands could test the basic data transmission.

About can-utils other commands, user could check the github:

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

Debian can-utils Package:

<https://packages.debian.org/en/sid/can-utils>

➤ cansend direction:

```

root@test-System-Product-Name:~# cansend
cansend - send CAN-frames via CAN_RAW sockets.

Usage: cansend <device> <can_frame>.

<can_frame>:
<can_id>#{data}      for Classical CAN 2.0 data frames
<can_id>#R{len}      for Classical CAN 2.0 data frames
<can_id>#{data}_{dlc} for Classical CAN 2.0 data frames
<can_id>#R{len}_{dlc} for Classical CAN 2.0 data frames
<can_id>##<flags>{data} for CAN FD frames

<can_id>:
3 (SFF) or 8 (EFF) hex chars
{data}:
0..8 (0..64 CAN FD) ASCII hex-values (optionally separated by '.')
{len}:
an optional 0..8 value as RTR frames can contain a valid dlc field
_{dlc}:
an optional 9..F data length code value when payload length is 8
<flags>:
a single ASCII Hex value (0 .. F) which defines canfd_frame.flags

Examples:
SA1#11.2233.44556677.88 / 123#DEADBEEF / 5AA# / 123##1 / 213##311223344 /
1F334455#1122334455667788_B / 123#R / 00000123#R3 / 333#R8_E

```

➤ candump direction:

```

root@test-System-Product-Name:~# candump
candump - dump CAN bus traffic.

Usage: candump [options] <CAN interface>+
       (use CTRL-C to terminate candump)

Options:
-t <type>      (timestamp: (a)bsolute/(d)elta/(z)ero/(A)bsolute w date)
-H            (read hardware timestamps instead of system timestamps)
-c            (increment color mode level)
-i            (binary output - may exceed 80 chars/line)
-a            (enable additional ASCII output)
-S            (swap byte order in printed CAN data[] - marked with '' )
-s <level>     (silent mode - 0: off (default) 1: animation 2: silent)
-l            (log CAN-frames into file. Sets '-s 2' by default)
-f <fname>     (log CAN-frames into file <fname>. Sets '-s 2' by default)
-L            (use log file format on stdout)
-n <count>     (terminate after reception of <count> CAN frames)
-r <size>      (set socket receive buffer to <size>)
-D            (Don't exit if a "detected" can device goes down)
-d            (monitor dropped CAN frames)
-e            (dump CAN error frames in human-readable format)
-B            (display raw DLC values in {} for Classical CAN)
-x            (print extra message infos, rx/tx brs esi)
-T <msecs>    (terminate after <msecs> if no frames were received)

Up to 16 CAN interfaces with optional filter sets can be specified
on the commandline in the form: <ifname>[,filter]*

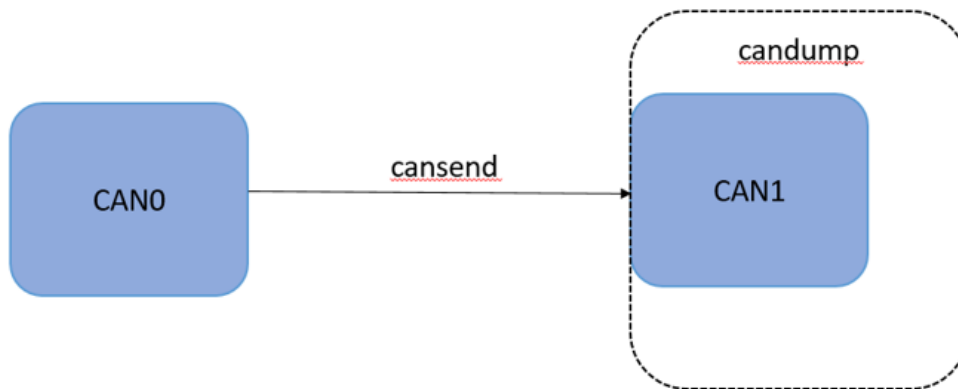
Filters:
Comma separated filters can be specified for each given CAN interface:
<can_id>:<can_mask>
    (matches when <received_can_id> & mask == can_id & mask)
<can_id>~<can_mask>
    (matches when <received_can_id> & mask != can_id & mask)
#<error_mask>
    (set error frame filter, see include/linux/can/error.h)
[j]j]
    (join the given CAN filters - logical AND semantic)

CAN IDs, masks and data content are given and expected in hexadecimal values.
When the can_id is 8 digits long the CAN_EFF_FLAG is set for 29 bit EFF format.
Without any given filter all data frames are received ('0:0' default filter).

Use interface name 'any' to receive from all CAN interfaces.

```

➤ Test Condition



✚ CAN 2.0

- 1). can0 send can frame by cansend

```

$ cansend [channel] [SID#data]

root@test-System-Product-Name: ~
File Edit View Search Terminal Help
root@test-System-Product-Name:~# cansend can0 1f123345#1122334455667788
root@test-System-Product-Name:~#
  
```

➤ Pattern Parsing

1f123345 → SID, send ID

→ CAN2.0

1122334455667788 → Data

- 2). can1 waiting the can frame by candump

```

$candump [channel]
  
```

➤ Received CAN frame

```

root@test-System-Product-Name:~# candump can1
can1 1F123345 [8] 11 22 33 44 55 66 77 88
  
```

If test is finish, enter “ctrl + c” to exit candump.

CAN FD

- 1). Send CAN FD frame by cansend

```
$cansend [channel] [SID##{flag}{data}]
```

```
root@test-System-Product-Name:~# cansend can0 7ff##31234567890abcdef1234567890abcdef
```

flag	RX/TX	BRS	ESI
0	RX	--	--
1	RX	✓	--
2	RX	--	✓
3	RX	✓	✓

➤ Pattern Parsing

7ff → SID, send ID

→ FD

3 → flag

1234567890abcdef1234567890abcdef → Data

- 2). can1 waiting the can frame(add -x to show flag) by candump

```
$candump [channel] -c -x
```

```
root@test-System-Product-Name:~# candump can1 -c -x
can1 RX B E 7FF [16] 12 34 56 78 90 AB CD EF 12 34 56 78 90 AB CD EF
```

If test is finish, enter “ctrl + c” to exit candump.

9. Troubleshooting

Errors often occur when compiling source code packages. The following is a compilation of some common errors.

1). “make” not found

```
root@test-System-Product-Name:/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/driver# make
Command 'make' not found, but can be installed with:
apt install make # version 4.3-4.1build1, or
apt install make-guile # version 4.3-4.1build1
```

Ubuntu command “apt-get install make-guile”

\$apt-get install make-guile

```
root@test-System-Product-Name:/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/driver# apt-get install make-guile
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  make-doc
The following NEW packages will be installed:
  make-guile
```

2). gcc-12 not found

```
root@test-System-Product-Name:/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/driver# make
Compile SDC driver. (kernel 6.5.0-45-generic)

make CFLAGS_MODULE+=-DNETCAN CFLAGS_MODULE+=-Wframe-larger-than=2048 EXTRA_CFLAGS+=-DSDC_CREATE_UART=1 -DSDC_CREATE_DIO=1 -DSDC_CREATE_SPI=1
-DSDC_CREATE_CAN=1 -DSDC_CREATE_PARPORT=0" -C /lib/modules/6.5.0-45-generic/build M=/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/driver
modules
make[1]: Entering directory '/usr/src/linux-headers-6.5.0-45-generic'
warning: the compiler differs from the one used to build the kernel
The kernel was built by: x86_64-linux-gnu-gcc-12 (Ubuntu 12.3.0-1ubuntu1~22.04) 12.3.0
You are using:
CC [M] /home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/driver/sunix-sdc.o
/bin/sh: 1: gcc-12: not found
```

Ubuntu command “apt-get install gcc-12”

```
$apt-get install gcc-12
```

```
root@test-System-Product-Name:/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/driver# apt-get install gcc-12
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu cpp-12 libasan8 libbinutils libc-dev-bin libc-devtools libc6-dev
  libisl-dev libtirpc-dev libtsan2 libubsan1 linux-libc-dev manpages-dev rpcsvc-proto
```

3). “snxterm” installation, gcc not found

```
root@test-System-Product-Name:/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/snxterm# make
gcc -O2 -pthread -pipe -o snxterm snxterm.c -lnurses
make: gcc: No such file or directory
make: *** [Makefile:13: snxterm] Error 127
```

Ubuntu command “apt-get install gcc”

```
$apt-get install gcc
```

```
root@test-System-Product-Name:/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/snxterm# apt-get install gcc
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu cpp-11 gcc-11 gcc-11-base libasan6 libbinutils libc-dev
  libgcc-11-dev libitm1 liblsan0 libnsl-dev libtirpc-dev libtsan0 libubsan1 linux-libc-dev manpages-dev rpcs
Suggested packages:
  binutils-doc gcc-11-locales gcc-multilib autoconf automake libtool flex bison gcc-doc gcc-11-multilib gcc-
Recommended packages:
  libnss-nis libnss-nisplus
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu gcc gcc-11 libasan6 libbinutils libc-dev-bin libc-devto
  libnsl-dev libtirpc-dev libtsan0 libubsan1 linux-libc-dev manpages-dev rpcsvc-proto
The following packages will be upgraded:
  cpp-11 gcc-11-base libc6 libc6-dbg
4 upgraded, 24 newly installed, 0 to remove and 437 not upgraded.
Need to get 39.0 MB/66.0 MB of archives.
After this operation, 131 MB of additional disk space will be used.
```

4). “snxterm” installation, ncurses.h: no such file

```
root@test-System-Product-Name:/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/snxterm# make
gcc -O2 -pthread -pipe -o snxterm snxterm.c -lncurses
snxterm.c:12:10: fatal error: ncurses.h: No such file or directory
 12 | #include <ncurses.h>
    | ^~~~~~
compilation terminated.
make: *** [Makefile:13: snxterm] Error 1
```

Ubuntu command “apt-get install libncurses5-dev”

```
$apt-get install libncurses5-dev
```

```
root@test-System-Product-Name:/home/test/SDCLinuxExpansionBoardDriver_V2.1.0.0/snxterm# apt-get install libncurses5-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libncurses-dev
Suggested packages:
  ncurses-doc
The following NEW packages will be installed:
  libncurses-dev libncurses5-dev
```