SVM

Interface Presentation

V0.0.3

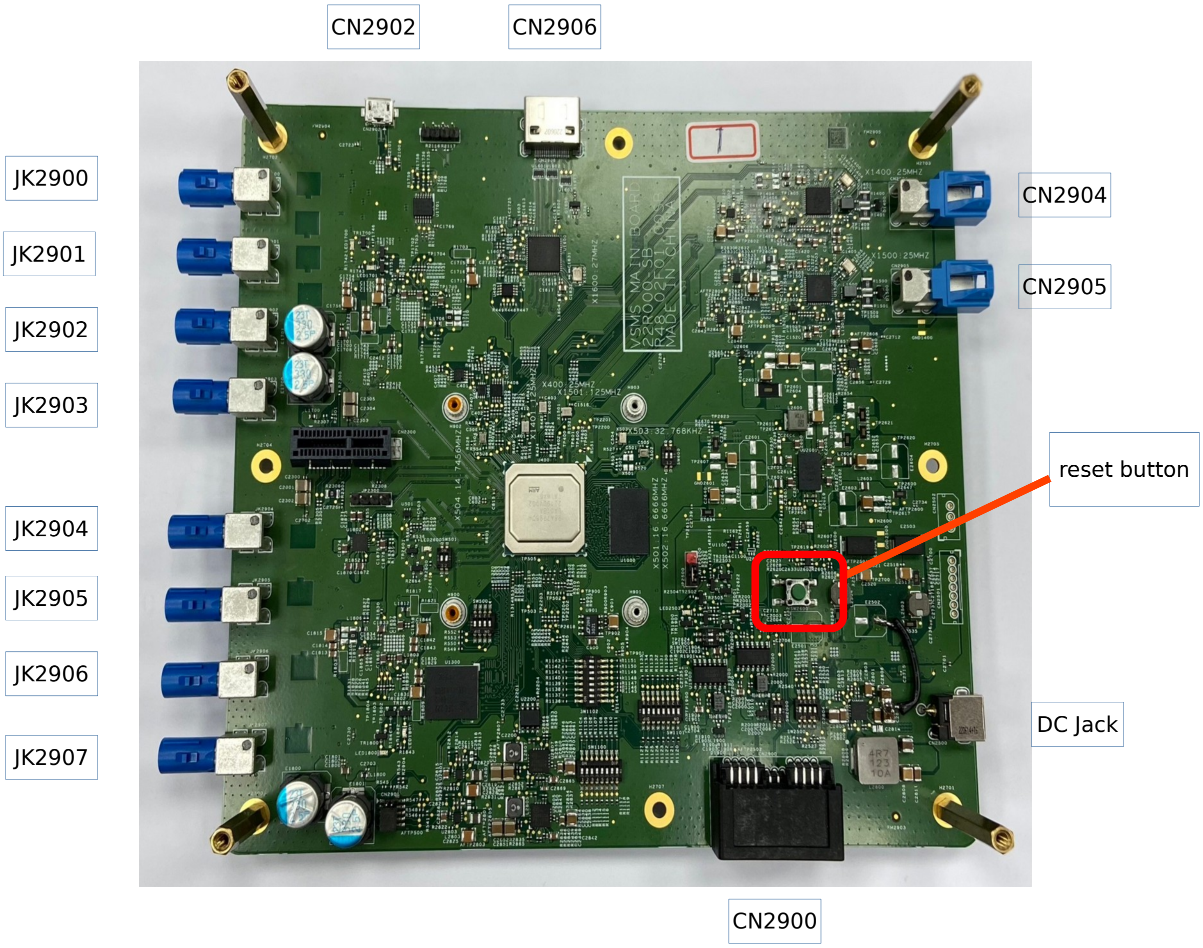
內容

i VERSION LIST

描述	文件版本	日期	修改
SVM VerifyFuncMethod 0.0.1	V0.0.1	2023/1/30	Jerry Lin
Add chapter of GPIO(Push Button)	V0.0.2	2023/4/19	Artie Wang
Update can screenshot	V0.0.3	2023/11/1	Jerry Lin

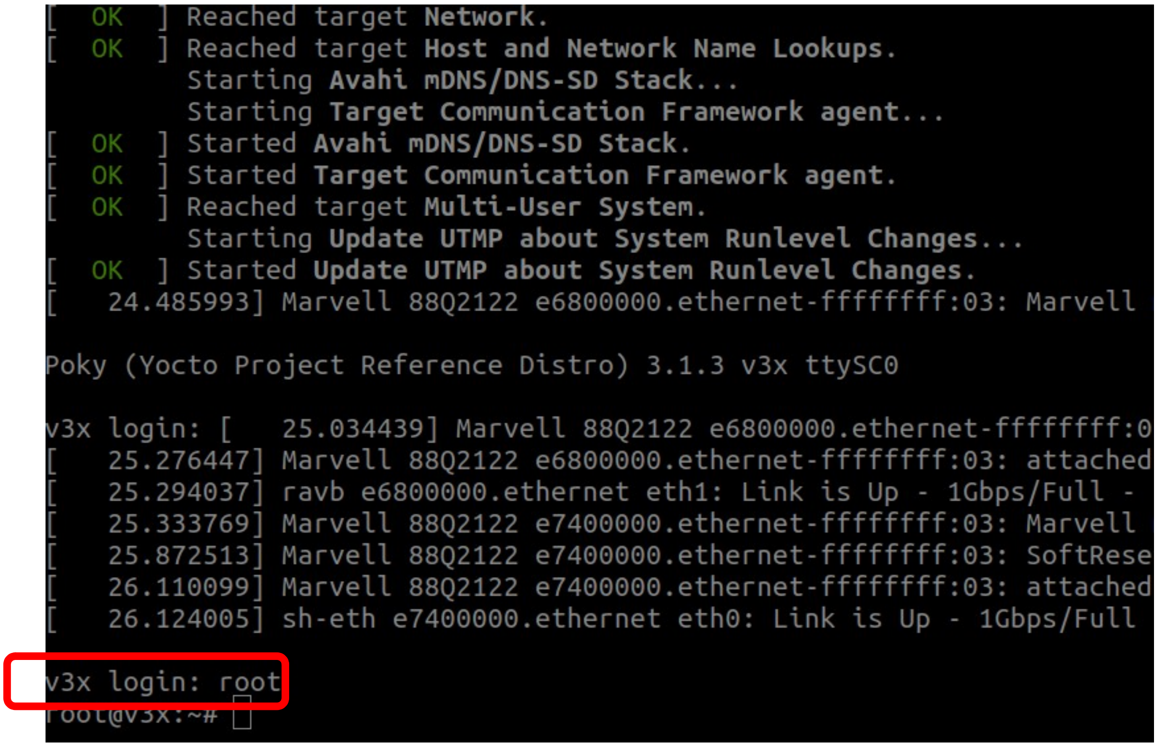
Interface

Interface 介紹



- # DEBUG Port

插入 Micro USB 線連接電腦, 開啟 terminal 程式並將 baud rate 設為 115200, 即可看到 terminal 訊息.開機完成後, 可用 root 登入.



```
[  OK  ] Reached target Network.
[  OK  ] Reached target Host and Network Name Lookups.
         Starting Avahi mDNS/DNS-SD Stack...
         Starting Target Communication Framework agent...
[  OK  ] Started Avahi mDNS/DNS-SD Stack.
[  OK  ] Started Target Communication Framework agent.
[  OK  ] Reached target Multi-User System.
         Starting Update UTMP about System Runlevel Changes...
[  OK  ] Started Update UTMP about System Runlevel Changes.
[   24.485993] Marvell 88Q2122 e6800000.ethernet-ffffffff:03: Marvell

Poky (Yocto Project Reference Distro) 3.1.3 v3x ttySC0

v3x login: [   25.034439] Marvell 88Q2122 e6800000.ethernet-ffffffff:0
[   25.276447] Marvell 88Q2122 e6800000.ethernet-ffffffff:03: attached
[   25.294037] ravb e6800000.ethernet eth1: Link is Up - 1Gbps/Full -
[   25.333769] Marvell 88Q2122 e7400000.ethernet-ffffffff:03: Marvell
[   25.872513] Marvell 88Q2122 e7400000.ethernet-ffffffff:03: SoftRese
[   26.110099] Marvell 88Q2122 e7400000.ethernet-ffffffff:03: attached
[   26.124005] sh-eth e7400000.ethernet eth0: Link is Up - 1Gbps/Full

v3x login: root
root@v3x:~# 
```

- FAKRA

JK2900 ~ JK2907

與 BSP node 對應如下：

JK2900 -> /dev/video0
JK2901 -> /dev/video1
JK2902 -> /dev/video2
JK2903 -> /dev/video3

JK2904 -> /dev/video4
JK2905 -> /dev/video5
JK2906 -> /dev/video6
JK2907 -> /dev/video7

驗證方式 ：

default 設定為

JK2900 ~ JK2904 支援 OV10635 camera
JK2904 ~ JK2907 支援 AR0233 camera
(以上可由 Device Tree 做更改)

在 Git Hub (https://github.com/RetronixTechInc/rcar-bsp/tree/main_sb)提供 Camera Test Script (rtx_camera_display.sh), 指令如下, N 為 0 ~ 7, 對應到 video0 ~ video7:

```
root@v3x:~# ./rtx_camera_display.sh N
```

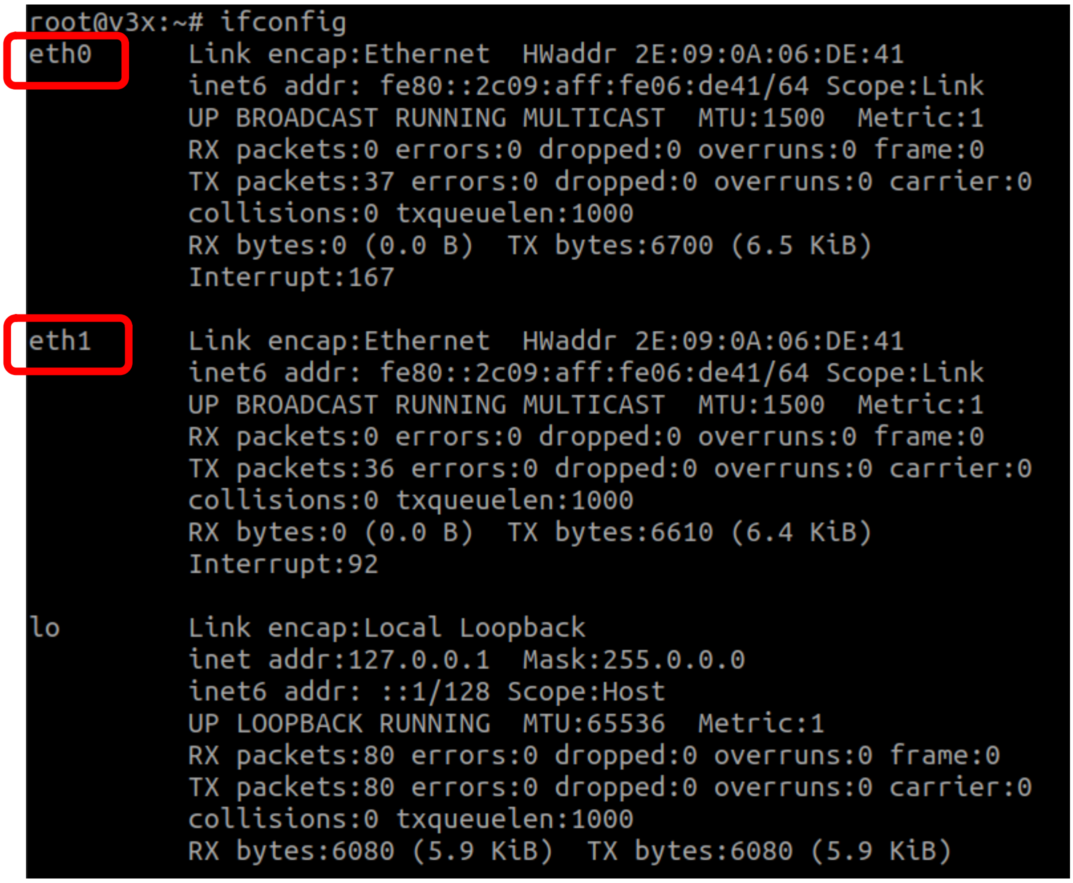
即可透過 HDMI Port 將 Camera 影像顯示至螢幕中.(如果要改變 Video Size 或 Format 等可修改此 Script)

- H-MTD

CN2904 ~ CN2905

與 BSP node 對應如下：

CN2904 -> eth1 (Master)
CN2905 -> eth0 (Slave)



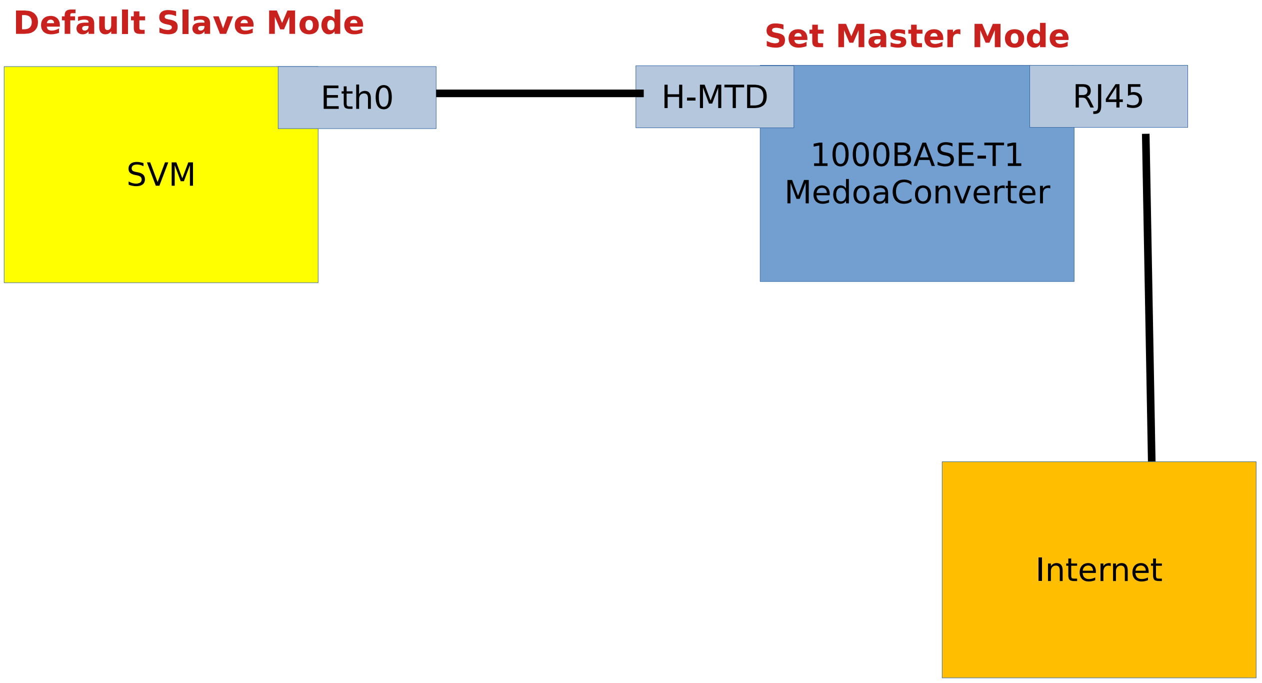
```
root@v3x:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 2E:09:0A:06:DE:41
          inet6 addr: fe80::2c09:aff:fe06:de41/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:37 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:6700 (6.5 KiB)
          Interrupt:167

eth1      Link encap:Ethernet  HWaddr 2E:09:0A:06:DE:41
          inet6 addr: fe80::2c09:aff:fe06:de41/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:36 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:6610 (6.4 KiB)
          Interrupt:92

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:80 errors:0 dropped:0 overruns:0 frame:0
          TX packets:80 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6080 (5.9 KiB)  TX bytes:6080 (5.9 KiB)
```

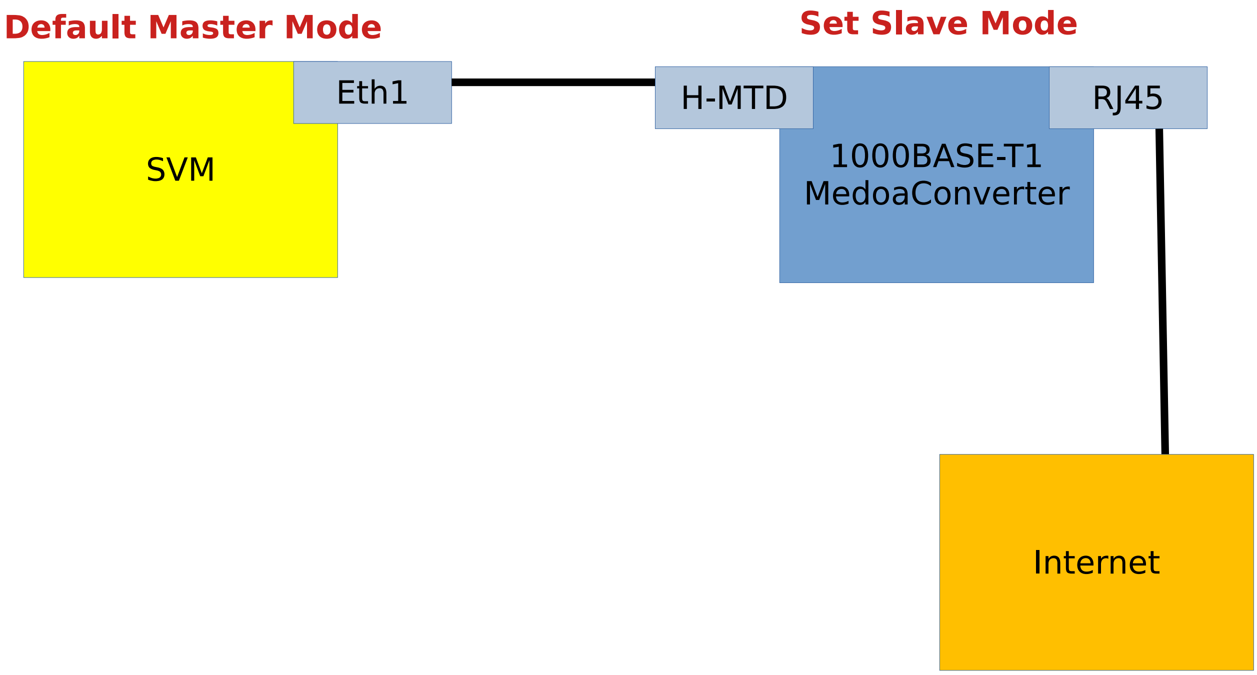
1. Eth0 連線

如須透過 RJ45 取得網路需要一台 1000BASE-T1 MediaConverter, 如要使用 Eth0, 因為 Eth0 預設是 Slave Mode, 所以要將 Converter 設為 Master Mode, 如下



2. Eth1 連線

如須透過 RJ45 取得網路需要一台 1000BASE-T1 MediaConverter, 如要使用 Eth1, 因為 Eth1 預設是 Master Mode, 所以要將 Converter 設為 Slave Mode, 如下

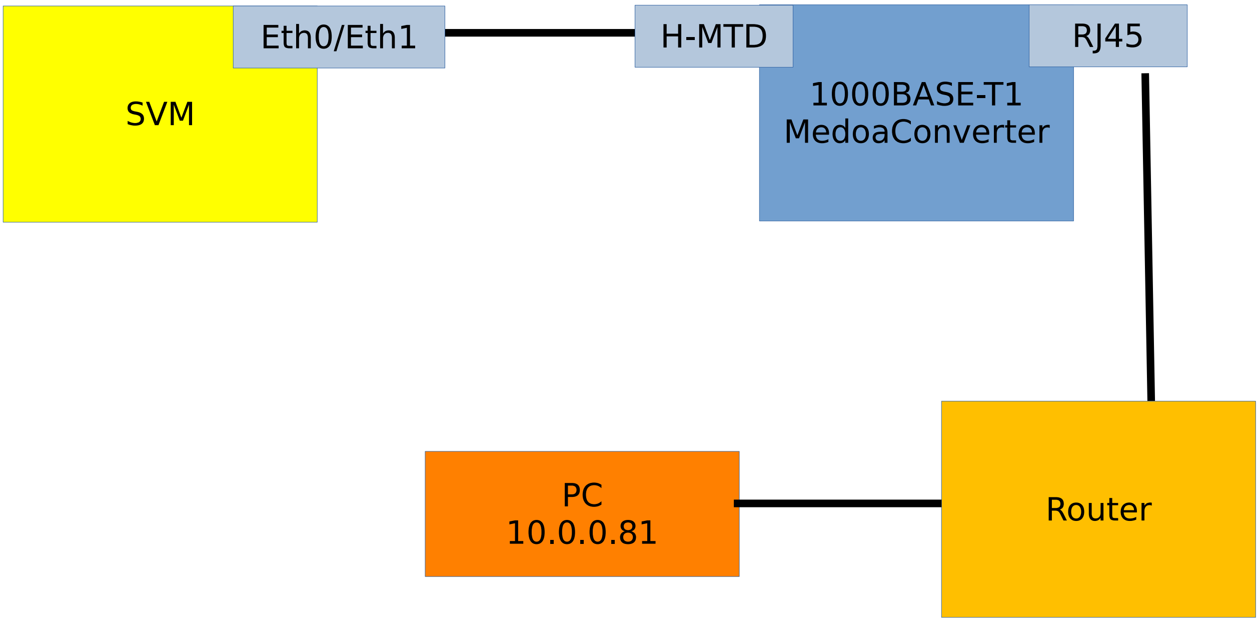


PS: **在 u-boot 中預設僅支援 eth1 (avb port)**

驗證方式 :

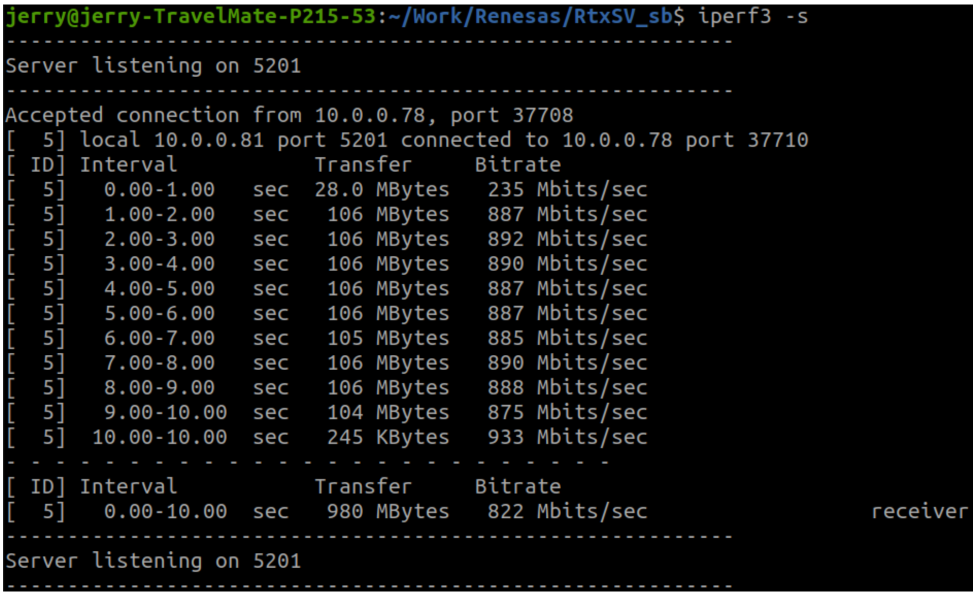
eth0 / eth1

環境設定如下:



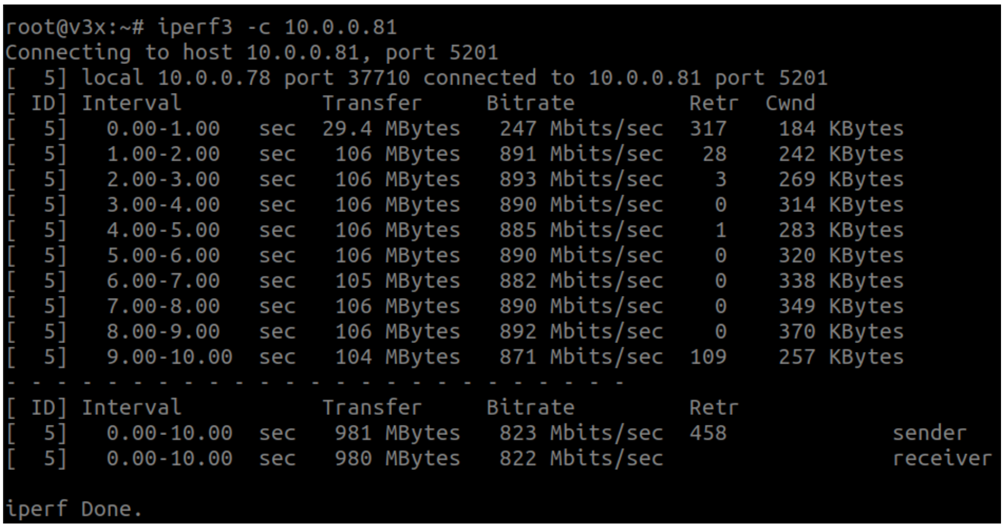
pc 當 Server 端, 執行

iperf3 -s



```
jerry@jerry-TravelMate-P215-53:~/Work/Renesas/RtxSV_sb$ iperf3 -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 10.0.0.78, port 37708
[  5] local 10.0.0.81 port 5201 connected to 10.0.0.78 port 37710
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-1.00   sec  28.0 MBytes   235 Mbits/sec
[  5]   1.00-2.00   sec   106 MBytes   887 Mbits/sec
[  5]   2.00-3.00   sec   106 MBytes   892 Mbits/sec
[  5]   3.00-4.00   sec   106 MBytes   890 Mbits/sec
[  5]   4.00-5.00   sec   106 MBytes   887 Mbits/sec
[  5]   5.00-6.00   sec   106 MBytes   887 Mbits/sec
[  5]   6.00-7.00   sec   105 MBytes   885 Mbits/sec
[  5]   7.00-8.00   sec   106 MBytes   890 Mbits/sec
[  5]   8.00-9.00   sec   106 MBytes   888 Mbits/sec
[  5]   9.00-10.00  sec   104 MBytes   875 Mbits/sec
[  5]  10.00-10.00  sec   245 KBytes   933 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-10.00  sec   980 MBytes   822 Mbits/sec                  receiver
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

SVM 當 Client 端, 執行
iperf3 -c ${PC_IP}

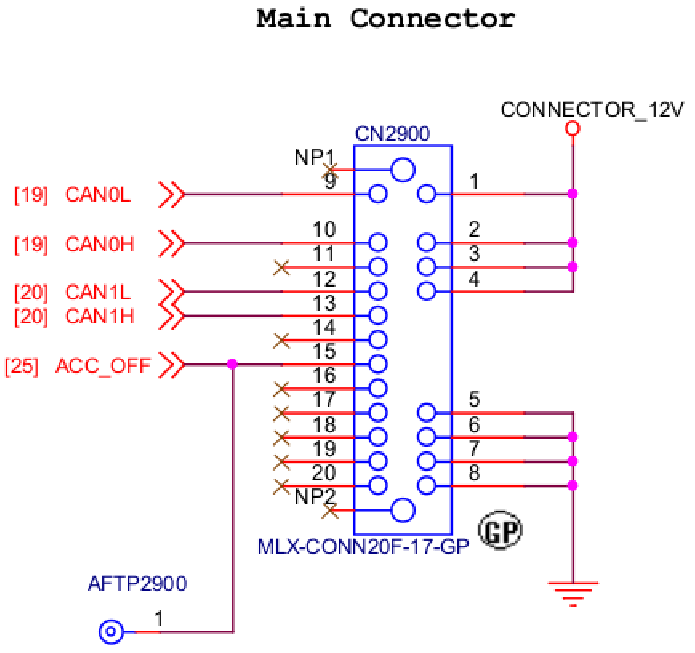


```
root@v3x:~# iperf3 -c 10.0.0.81
Connecting to host 10.0.0.81, port 5201
[  5] local 10.0.0.78 port 37710 connected to 10.0.0.81 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  29.4 MBytes   247 Mbits/sec  317    184 KBytes
[  5]   1.00-2.00   sec   106 MBytes   891 Mbits/sec   28    242 KBytes
[  5]   2.00-3.00   sec   106 MBytes   893 Mbits/sec    3    269 KBytes
[  5]   3.00-4.00   sec   106 MBytes   890 Mbits/sec    0    314 KBytes
[  5]   4.00-5.00   sec   106 MBytes   885 Mbits/sec    1    283 KBytes
[  5]   5.00-6.00   sec   106 MBytes   890 Mbits/sec    0    320 KBytes
[  5]   6.00-7.00   sec   105 MBytes   882 Mbits/sec    0    338 KBytes
[  5]   7.00-8.00   sec   106 MBytes   890 Mbits/sec    0    349 KBytes
[  5]   8.00-9.00   sec   106 MBytes   892 Mbits/sec    0    370 KBytes
[  5]   9.00-10.00  sec   104 MBytes   871 Mbits/sec  109    257 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec   981 MBytes   823 Mbits/sec  458             sender
[  5]   0.00-10.00  sec   980 MBytes   822 Mbits/sec                  receiver

iperf Done.
```

即可得到傳輸速率

- Main Connector (CAN bus)



SVM 有兩個 CAN bus, 如上圖, 對應到系統中的 can0 (CAN0L, CAN0H)與 can1 (CAN1L, CAN1H).

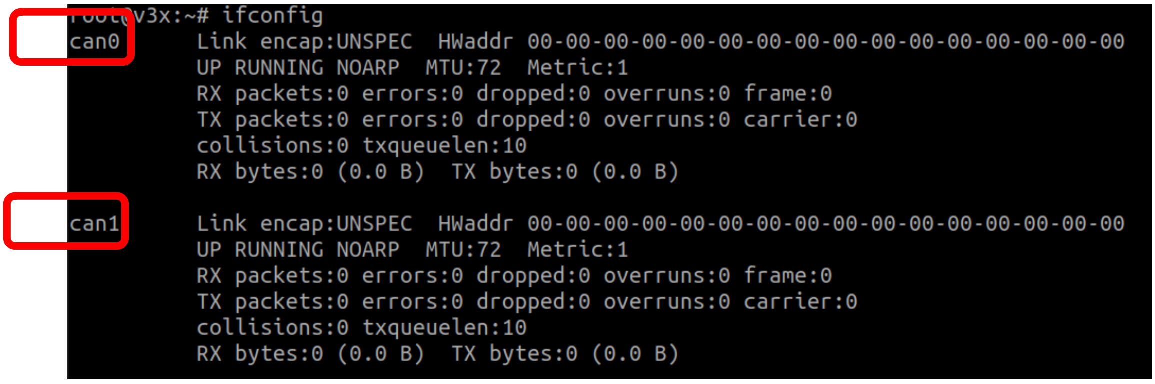
驗證方式：

1. 準備一條線材將 can0 與 can1 對接, CAN0H 接 CAN1H, CAN0L 接 CAN1L.
2. 在系統中 enable can0 與 can1, 執行以下指令

```
ip link set can0 down type can
ip link set can0 type can bitrate 100000 dbitrate 4000000 fd on
ip link set can0 up type can

ip link set can1 down type can
ip link set can1 type can bitrate 100000 dbitrate 4000000 fd on
ip link set can1 up type can
```

3. 執行 ifconfig 查看 can0/can1 是否已經 enable.



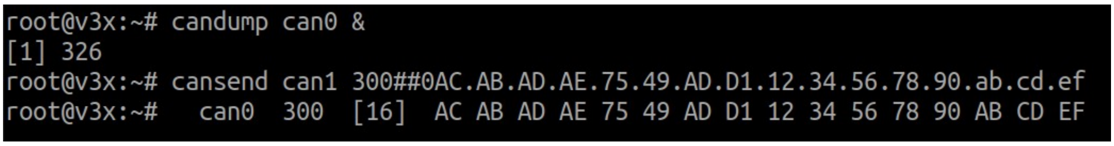
```
root@v3x:~# ifconfig
can0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          UP RUNNING NOARP  MTU:72  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:10
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

can1      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          UP RUNNING NOARP  MTU:72  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:10
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

4. can0 當接收端, can1 當傳送端, 實際傳送資料

```
candump can0 &

cansend can0 300##0AC.AB.AD.AE.75.49.AD.D1.12.34.56.78.90.ab.cd.ef
```



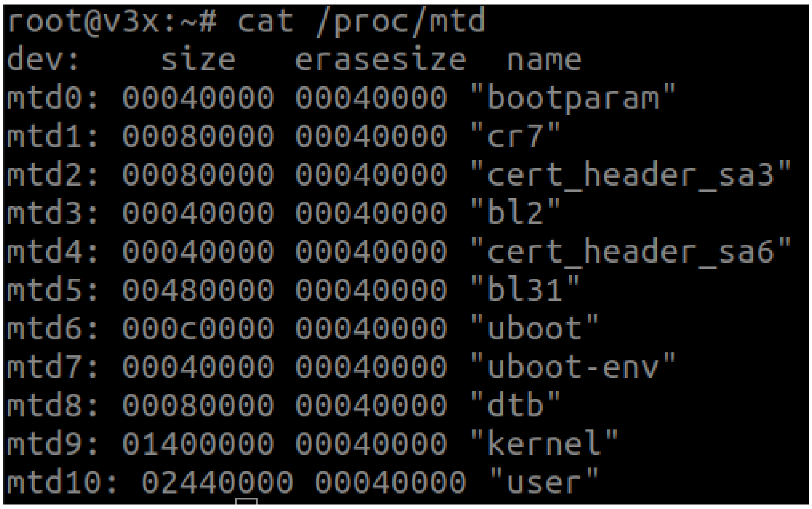
```
root@v3x:~# candump can0 &
[1] 326
root@v3x:~# cansend can1 300##0AC.AB.AD.AE.75.49.AD.D1.12.34.56.78.90.ab.cd.ef
root@v3x:~#   can0  300  [16]  AC AB AD AE 75 49 AD D1 12 34 56 78 90 AB CD EF
```

5. 如果溝通成功會看到 can0 收到 can1 所傳送的資料.

- ## SPI Flash (512 Mbits)

  目前 SPI Flash 裡面放 IPL, u-boot, uboot-env, ramdisk...,所以不建議更改, 但可以用以下指令去看 SPI Flash 的資訊.

```
cat /proc/mtd
```



```
root@v3x:~# cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00040000 00040000 "bootparam"
mtd1: 00080000 00040000 "cr7"
mtd2: 00080000 00040000 "cert_header_sa3"
mtd3: 00040000 00040000 "bl2"
mtd4: 00040000 00040000 "cert_header_sa6"
mtd5: 00480000 00040000 "bl31"
mtd6: 000c0000 00040000 "uboot"
mtd7: 00040000 00040000 "uboot-env"
mtd8: 00080000 00040000 "dtb"
mtd9: 01400000 00040000 "kernel"
mtd10: 02440000 00040000 "user"
```

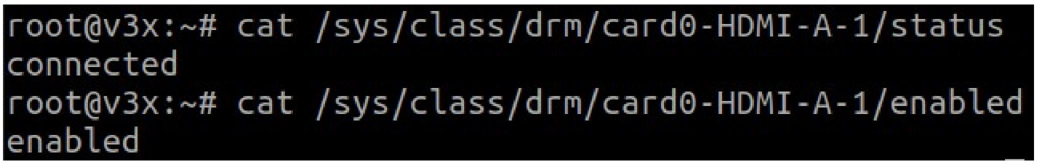
- ## HDMI

  透過 HDMI 插上螢幕可以看到我們將 terminal 資訊導到螢幕上, 也可以透過 cat 以下位置去看其他資訊.

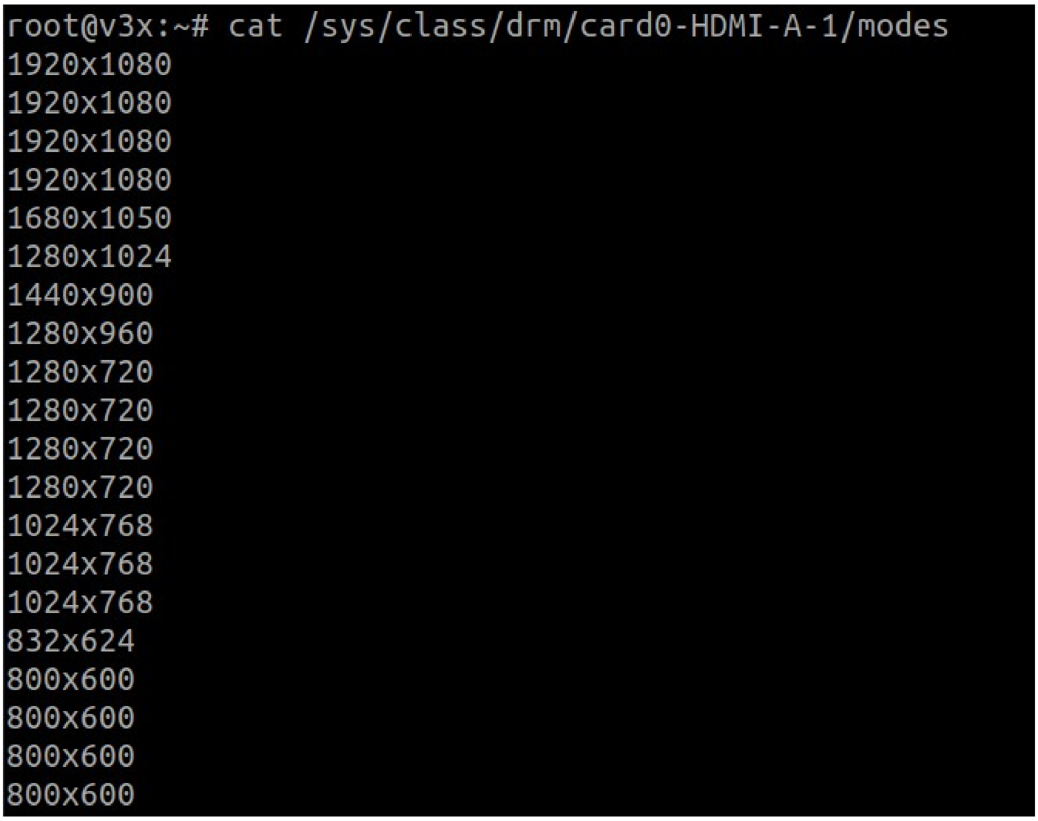
```
cat /sys/class/drm/card0-HDMI-A-1/status

cat /sys/class/drm/card0-HDMI-A-1/status

cat /sys/class/drm/card0-HDMI-A-1/modes
```



```
root@v3x:~# cat /sys/class/drm/card0-HDMI-A-1/status
connected
root@v3x:~# cat /sys/class/drm/card0-HDMI-A-1/enabled
enabled
```

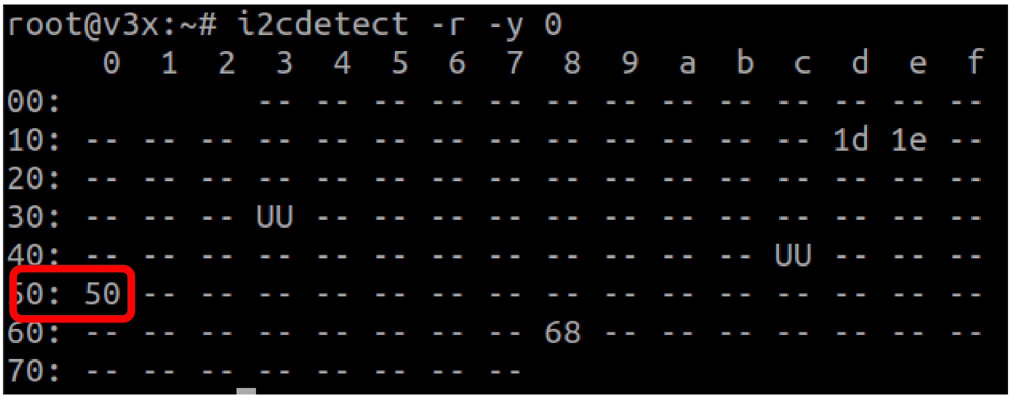


```
root@v3x:~# cat /sys/class/drm/card0-HDMI-A-1/modes
1920x1080
1920x1080
1920x1080
1920x1080
1680x1050
1280x1024
1440x900
1280x960
1280x720
1280x720
1280x720
1280x720
1024x768
1024x768
1024x768
832x624
800x600
800x600
800x600
800x600
```

- # EEPROM (1k bit)

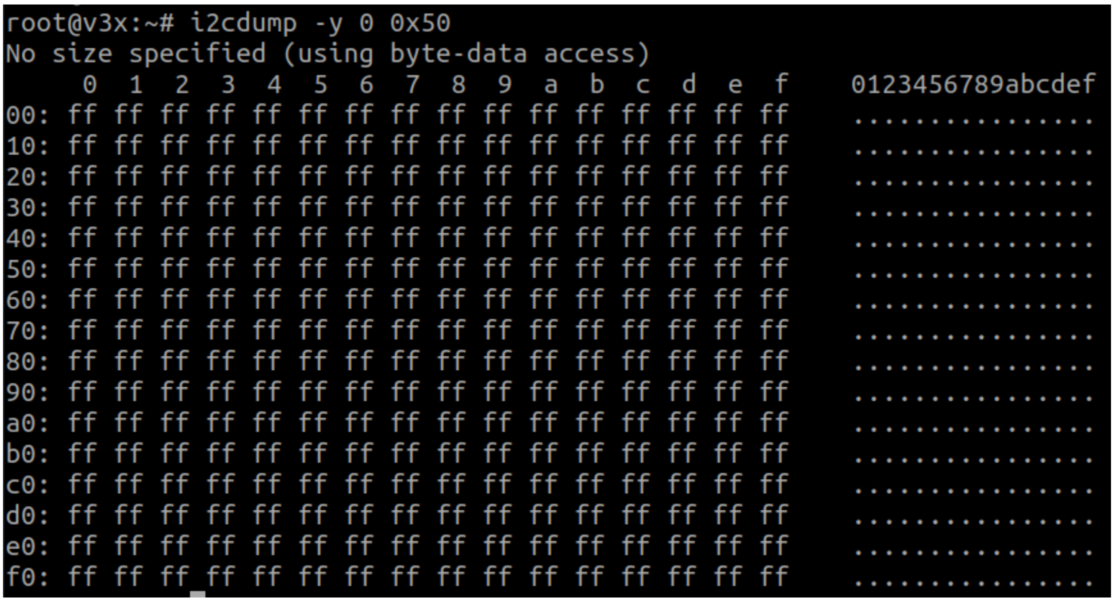
驗證方式：

1. EEPROM 的 i2c ID 為 0x50



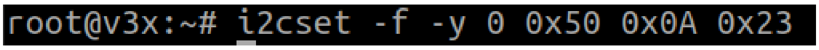
```
root@v3x:~# i2cdetect -r -y 0
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- 1d 1e --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- UU -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- UU -- -- --
50: 50 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

2.dump EEPROM 內容



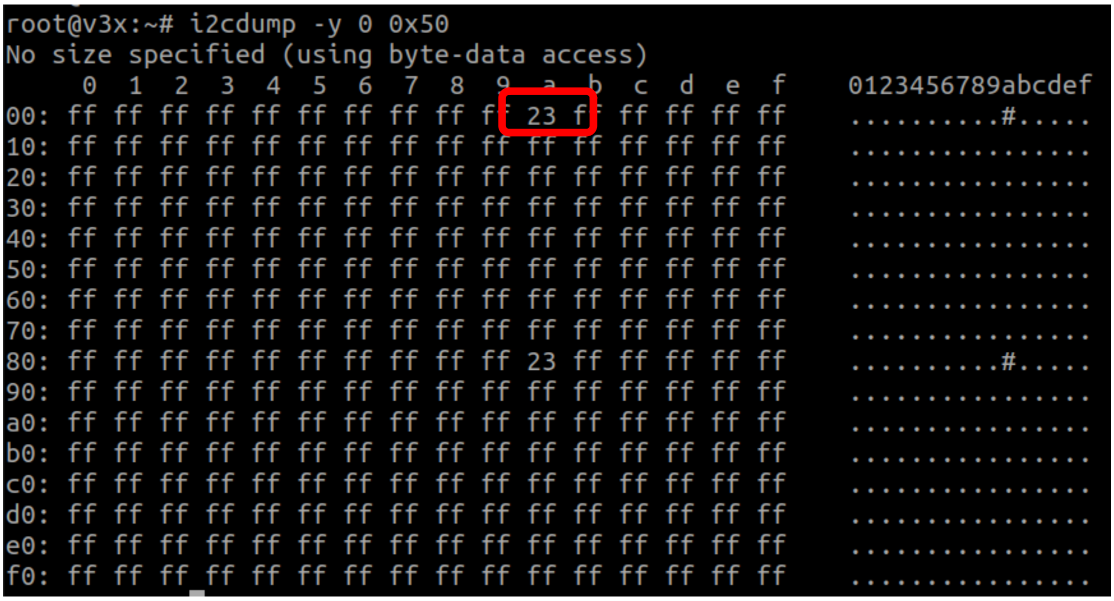
```
root@v3x:~# i2cdump -y 0 0x50
No size specified (using byte-data access)
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f    0123456789abcdef
00: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
10: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
20: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
30: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
40: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
50: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
60: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
70: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
80: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
90: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
a0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
b0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
c0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
d0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
e0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
f0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
```

3.對 0x0a 的位置寫入 0x23.



```
root@v3x:~# i2cset -f -y 0 0x50 0x0A 0x23
```

4. 再 dump 一次 EEPROM 內容, 確認 0x0a 已經變剛寫入的值即代表寫入成功.



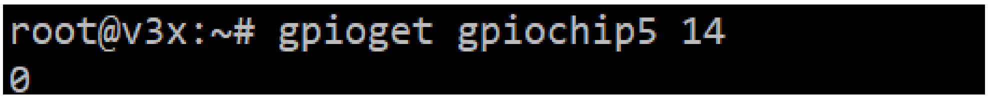
```
root@v3x:~# i2cdump -y 0 0x50
No size specified (using byte-data access)
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f    0123456789abcdef
00: ff ff ff ff ff ff ff ff ff ff 23 ff ff ff ff ff    ..........#.....
10: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
20: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
30: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
40: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
50: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
60: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
70: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
80: ff ff ff ff ff ff ff ff ff ff 23 ff ff ff ff ff    ..........#.....
90: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
a0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
b0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
c0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
d0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
e0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
f0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
```

• GPIO (Push Button)

SW2100 -> GP5_14
SW2101 -> GP5_12
SW2102 -> GP5_13

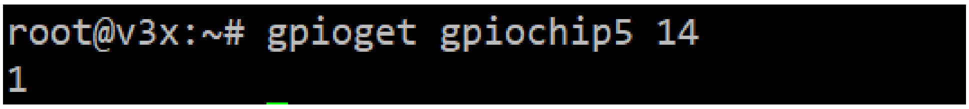
以下範例為讀取 button SW2100 的狀態 :

1. press SW2100 (GP5_14=0)



```
root@v3x:~# gpioget gpiochip5 14
0
```

2. release SW2100 (GP5_14=1)



```
root@v3x:~# gpioget gpiochip5 14
1
```