

## ABH3 CAN 通信 Linux 系の注意点

### 1, ハードウェアを有効にする : Raspberry Pi の場合

製品名 : PiCAN2 CAN-Bus Board for Raspberry Pi 2/3

会社名 : SK Pang Electronics Ltd

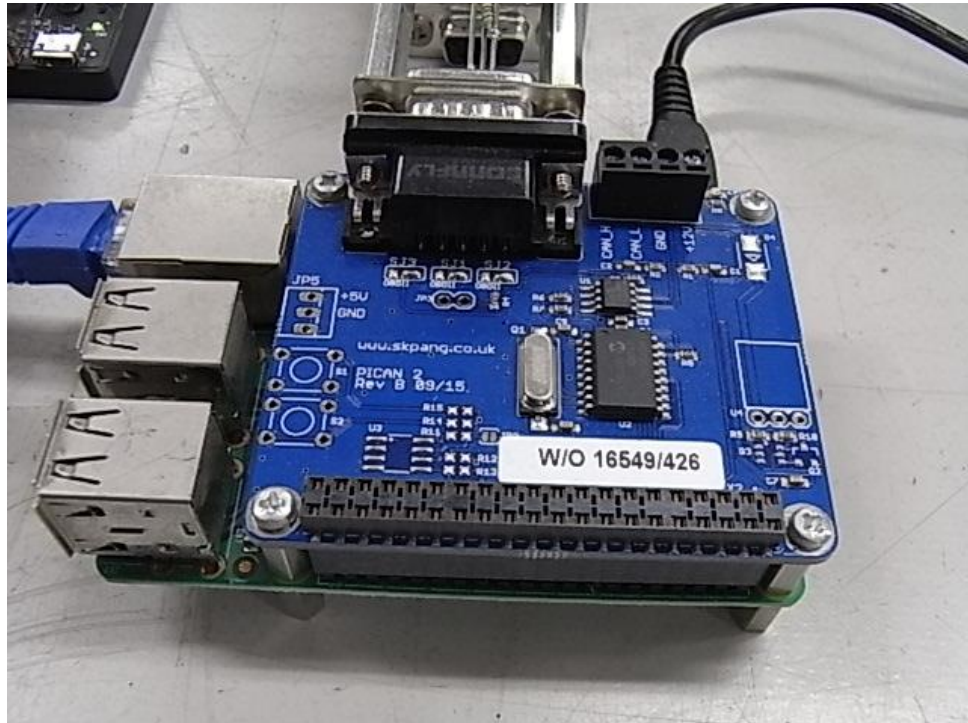
参考 HP : <http://skpang.co.uk/catalog/pican2-canbus-board-for-raspberry-pi-23-p-1475.html>

ハードウェアのマニュアルに従いソフトウェアをインストール下さい。※1 次ページ参照  
linux-can-utils フォルダにある lib.c / lib.h ファイルと include フォルダが必要になりますので適時コピーしてください。

起動毎に以下を実行する必要があります。systemd 等を使用して起動時に実行すると便利です。

```
sudo /sbin/ip link set can0 up type can bitrate 500000
```

500000 はボーレートになります。使用する機器に合わせて変更してください。



※ 1

2021/6/14 現在、V1.3 マニュアルにある以下のダウンロードページが見えなくなっています。

[http://www.skpang.co.uk/dl/can-test\\_pi2.zip](http://www.skpang.co.uk/dl/can-test_pi2.zip)

<https://bitbucket.org/hardbyte/python-can/get/4085cffd2519.zip>

以下の手順に従ってください。

- ・ マニュアルの通り「/boot/config.txt」を編集する。以下 3 行を追加。

```
dtoverlay=spi=on
```

```
dtoverlay=mcp2515-can0,oscillator=16000000,interrupt=25
```

```
dtoverlay=spi-bcm2835-overlay
```

- ・ リブートする

```
reboot あるいは sudo reboot
```

- ・ CAN ポートを有効にする。

```
sudo /sbin/ip link set can0 up type can bitrate 500000
```

500000 はボーレートになります。使用する機器に合わせて変更してください。

- ・ can-utils をインストールする。

```
git clone https://github.com/linux-can/can-utils.git
```

```
cd can-utils
```

```
./autogen.sh
```

```
./configure
```

```
make
```

```
sudo make install
```

## 2, ハードウェアを有効にする：Jetson Xavier NX の場合

本ボードは CAN インターフェースが 1 系統ありますが、CAN ドライバ IC は搭載されておりません。本サンプルでは弊社製作の基板を使用しますが、同基板をご利用いただく場合は営業部まで連絡ください。

can-utils をインストールする必要があります。任意のディレクトリで実行してください。

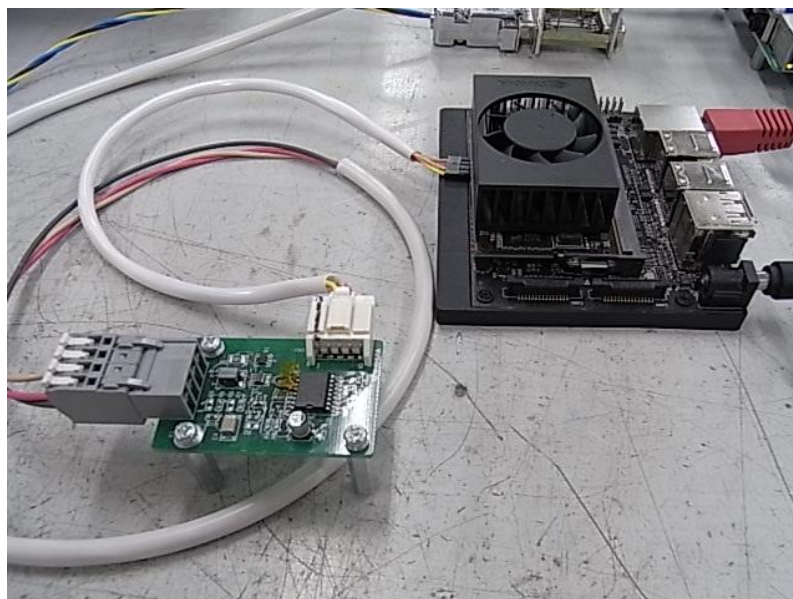
```
git clone https://github.com/linux-can/can-utils
cd can-utils/
./autogen.sh
./configure
make
sudo make install
```

can-utils フォルダにある lib.c / lib.h ファイルと include フォルダが必要になりますので適時コピーしてください。

起動毎に以下を実行する必要があります。systemd 等を使用して起動時に実行すると便利です。

```
sudo busybox devmem 0x0c303000 32 0x0000C400
sudo busybox devmem 0x0c303008 32 0x0000C458
sudo modprobe can
sudo modprobe can_raw
sudo modprobe mttcan
sudo ip link set can0 type can bitrate 500000
sudo ip link set can0 up
```

500000 はボーレートになります。使用する機器に合わせて変更してください。



### 3、ソースのコンパイル

GitHub よりダウンロードしたフォルダに lib.c / lib.h と include ディレクトリをコピーし、make することで実行ファイルが作成されます。

実行結果：

```
takahisa@XavierNX:~/Downloads/abh3/abh3_CAN-Bus_Linux$ make
cc -Wl,--no-as-needed -c -o lib.o lib.c
cc -Wl,--no-as-needed -c -o canABH3.o canABH3.c
cc -Wl,--no-as-needed -c -o pack_float.o pack_float.c
cc -Wl,--no-as-needed -lm single_canABH3.o canABH3.o pack_float.o lib.o -o single_canABH3
cc -Wl,--no-as-needed -lm bcast_canABH3.o canABH3.o pack_float.o lib.o -o bcast_canABH3
```

#### ① single\_canABH3.c

アラーム解除後、サーボオンを行い、一定回転で 10 秒間モータを回転させ、サーボオフする。  
回転中は回転速度(Y 軸、X 軸、A 軸、B 軸)を表示する。

実行結果：

```
takahisa@XavierNX:~/src/testCAN/sample$ ./single_canABH3
open
Single Pcket DP0 Send ID: 00ef0102
                Recv ID: 00ef0201
Single Pcket DP1 Send ID: 01ef0102
                Recv ID: 01ef0201
Broad Packet Send Resuest ID: 00ea0102
                Recv Base ID: 00ff0001

Abnormal: 00400000 Warning: 00400000
Abnormal: 00000000 Warning: 00000000
-4.600    -4.600    -9.200      0.000
 7.200     1.400     8.600     6.000
-6.400    -1.600    -8.000    -4.800
-4.200    -4.200    -8.600     0.000
-1.200    -7.200    -8.400     5.800
-0.400    -0.400    -0.800     0.000
 0.000     0.000     0.000     0.000
 3.800    -13.200    -9.400    17.200
13.400    -20.800    -7.200    34.200
35.800    -26.400     9.400    62.000
48.800    -47.600     1.000    96.400
62.400    -69.600    -7.400   132.000
83.200    -57.000    26.200   140.200
80.000    -63.200    16.800   143.200
91.400    -47.000    44.400   138.400
92.000    -56.400    35.600   148.400
105.600   -52.600    53.000   158.200
103.600   -58.400    45.200   161.800
117.600   -55.200    62.600   172.800
112.800   -45.600    67.200   158.200
114.800   -42.200    72.600   157.000
(以下略)
```

## ② bcast\_canABH3.c

ブロードキャストパケット 0～6のリクエストを行い、戻り値を表示する。

実行結果：

```
takahisa@XavierNX:~/src/testCAN/sample$ ./bcast_canABH3
open
Single Pcket DP0 Send ID: 00ef0102
                        Recv ID: 00ef0201
Single Pcket DP1 Send ID: 01ef0102
                        Recv ID: 01ef0201
Broad Packet Send Resuest ID: 00ea0102
                        Recv      Base ID: 00ff0001

Abnormal: 00400000 Warning: 00400000
control: 08070070 in_out: 000000c3
velCmdAY:  0.000000 velCmdBX:  0.000000 velFbkAY: -2.600000 velFbkBX:  3.000000
curCmdAY:  0.000000 curCmdBX:  0.000000 loadA:  0.600000 loadB:  0.200000
pulseA: 141803 pulseB: 430787
analog0:  0.210000 analog1:  0.140000 mainVolt: 23.700001 controlVolt: 23.700001
monitor0:  0.011930 monitor1:  0.106096
```

## 4, ソース一覧

ファイル名	内容
single_canABH3.c	ABH3 の CAN サンプルライブラリを使用したシングルパケットのサンプル
single_canABH3++.cpp	上記サンプルの C++バージョン
bcast_canABH3.c	ABH3 の CAN サンプルライブラリを使用したブロードキャストパケットのサンプル
bcast_canABH3++.cpp	上記サンプルの C++バージョン
canABH3.c	ABH3 の CAN サンプルライブラリの c ソースファイル ※本サンプルライブラリは異常時の処理等を十分には考慮しておりません。
canABH3.h	同 C ヘッダファイル
canABH3++.cpp	上記ライブラリの C++ラッパーソフト
canABH3++.hpp	同 C++ヘッダファイル
lib.c	Volkswagen Group Electronic Research の c ソースファイル
lib.h	同 C ヘッダファイル
include/	CAN ヘッダファイルのフォルダ
makefile	上記 make ファイル
set_can_pins_raspi	Raspberry Pi 用ハードウェア設定シェルフファイル
set_can_pins_jetson	Jetson Xavier NX 用ハードウェア設定シェルフファイル