

ラズパイの USB バスパワーを OFF/ON する方法

1. 概要

ラズパイを使用し、RFID リーダ・ライタ(USB 給電タイプ)を制御する場合、何らかの理由によりリーダ・ライタをリセットするために USB ケーブルを抜き差しして電源を切り入りしたいことがあるかもしれません。

ところが、直接 USB ケーブルを抜き差しできないときは、アプリケーション内からそれを行う必要があります。

本書では、アプリケーション内からラズパイの USB バスパワーを OFF/ON する2つの方法を記します。

1つ目は、ラズパイの4つの USB ポート全てを OFF/ON する方法、

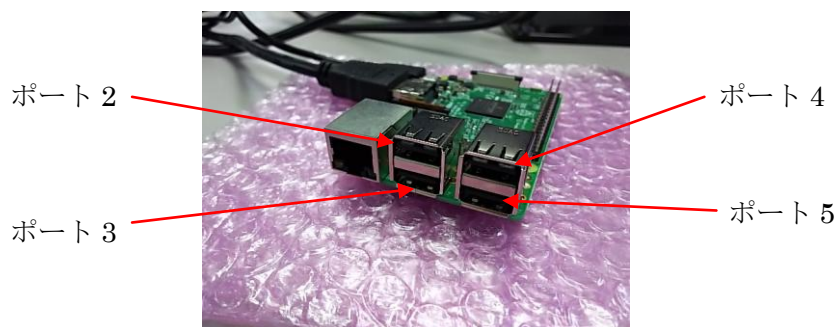
2つ目は、RFID リーダ・ライタのポートだけを OFF/ON する方法です。

想定している機器などは下のとおりです。

- ◆ラズパイ: Raspberry Pi 3 Model B
- ◆OS: Raspbian
- ◆言語: python3
- ◆RFID リーダ・ライタ: アートファイネックス社製 RFID リーダ・ライタ(USB 給電タイプ)

2. ラズパイの USB ポートについて

ラズパイには下図のように USB ポートが4つ付いており、ポート2~5と割り付けられています。(ポート1はLAN。)



例えば下図のように ポート2にキーボード、ポート3にマウス、ポート5にリーダ・ライタを接続して、`lsusb -f`を実行すると



```

$ lsusb -t
/: Bus 01.Port 1: Dev 1, Class=root_hub, Driver=dwc_otg/1p, 480M
  |__ Port 1: Dev 2, If 0, Class=Hub, Driver=hub/5p, 480M
    |__ Port 1: Dev 20, If 0, Class=Vendor Specific Class, Driver=smc95xx, 480M
    |__ Port 2: Dev 21, If 0, Class=Human Interface Device, Driver=usbhid, 1.5M } キーボード
    |__ Port 2: Dev 21, If 1, Class=Human Interface Device, Driver=usbhid, 1.5M }
    |__ Port 3: Dev 22, If 0, Class=Human Interface Device, Driver=usbhid, 1.5M   マウス
    |__ Port 5: Dev 23, If 0, Class= Vendor Specific Class, Driver=ftdi_sio, 12M   リーダ・ライター
  
```

キーボード/マウス/リーダ・ライターがそれぞれの USB ポートに接続されていることが分かります。

本書では、このポート 2~5 の全てのポートを OFF/ON する方法と、RFID リーダ・ライターのポートだけを OFF/ON する方法を記します。

3. 全ポートを OFF/ON

3.1. コマンド

USB バスパワーを切るときは下のコマンドを実行します。(unbind)

```
echo -n "1-1" > /sys/bus/usb/drivers/usb/unbind
```

また入れるときは下のコマンドを実行します。(bind)

```
echo -n "1-1" > /sys/bus/usb/drivers/usb/bind
```

特権が必要なため

```
sudo sh -c " * * * * * "
```

で行う必要があるので実際には下のようになります。

```
sudo sh -c "echo -n ¥"1-1¥" > /sys/bus/usb/drivers/usb/unbind"
```

```
sudo sh -c "echo -n ¥"1-1¥" > /sys/bus/usb/drivers/usb/bind"
```

※注意

コマンドラインで unbind すると、キーボードとマウスの操作ができなくなります。

3.2. アプリケーションで実行

アプリケーションで実行するには、subprocess を使います。

始めに

```
import subprocess
```

を宣言しておき、アプリケーション内で下のようになります。

OFF するときは

```
subprocess.call("sudo sh -c ¥"echo -n ¥"1-1¥" > /sys/bus/usb/drivers/usb/unbind¥"",  
shell=True)
```

ON するときは

```
subprocess.call("sudo sh -c ¥"echo -n ¥"1-1¥" > /sys/bus/usb/drivers/usb/bind¥"",  
shell=True)
```

3.3. サンプルアプリケーション

```

import serial
import time
import subprocess          # subprocess を使うという宣言

for n in range(100):      # 以下を 100 回繰り返す
    time.sleep(5)         # 5 行待ち
    ser = serial.Serial("/dev/ttyUSB0")
    ser.baudrate = 19200
    sendData = b'\xBB\x80\x22\x00\x02\x01\xA1\x7E'
    print(n)
    ser.write(sendData)
    while (True):
        recvData = ser.read(5)
        len1 = recvData[3]*256+recvData[4]+1
        temp1 = ser.read(len1)
        if (recvData[2] == 0xFF):
            print('No tag!!')
            break
        elif (recvData[2] == 0x27):
            break
        else:
            len2 = len(temp1)
            temp2 = temp1[2:len2-4]
            print(temp2.hex())
    ser.close()
    time.sleep(5)         # 5 行待ち
    subprocess.call("sudo sh -c 'echo -n '1-1' > /sys/bus/usb/drivers/usb/unbind'".
shell=True)             # USB バスパワーを切る
    print('\nOFF\n')
    time.sleep(5)         # 5 行待ち
    subprocess.call("sudo sh -c 'echo -n '1-1' > /sys/bus/usb/drivers/usb/bind'".
shell=True)             # USB バスパワーを入れる
    print('ON\n')
    
```

USB オープン
 タグ読取りコマンドを送信
 読取り結果を受信&表示
 USB クローズ

4. リーダ・ライタのポートだけを OFF/ON

4.1. はじめに

ラズパイの4つの USB ポートのどれか1つを OFF/ON する方法はありませんが、システムトークス社製の SUGOI HUB4X を使用すればその HUB のポート1またはポート2だけを OFF/ON できます

4.2. 事前準備

ラズパイをネットワークに接続して、以下の手順で libusb-dev をインストールして hub-ctrl を使えるようにします。

```
$ sudo apt-get install libusb-dev  
$ wget http://www.gniibe.org/oitoite/ac-power-control-by-USB-hub/hub-ctrl.c  
$ gcc -O2 hub-ctrl.c -o hub-ctrl-armhf-static -lusb -static  
$ sudo cp hub-ctrl-armhf-static /usr/local/bin/hub-ctrl
```

システムトークス社製の SUGOI HUB4X を用意します。

4.3. 接続方法

下図のように、ラズパイのポートと SUGOI HUB を接続し、SUGOI HUB の一番左のポートにリーダ・ライタを接続します。(左側の2ポートのいずれかでなければ個別に OFF/ON できません。)



4.4. デバイス番号の確認

SUGOI HUB とリーダー・ライタのデバイス番号を確認します。

```
$ lsusb
```

```
Bus 001 Device 007: ID 0403:6001 Future Technology Devices International, Ltd FT232 USB-serial (UART) IC          リーダ・ライタ
Bus 001 Device 006: ID 0409:005a NEC Corp. HighSpeed Hub          SUGOI HUB
Bus 001 Device 005: ID 413C:301a Dell Computer Corp.              マウス
Bus 001 Device 004: ID 413C:2113 Dell Computer Corp.             キーボード
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.     LAN
    SMSC9512/9514 Fast Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp. SMC9514 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 roothub
```

```
$ lsusb -t
```

```
/: Bus 01: Port 1: Dev 1, Class=root_hub, Driver=dwc_otg/1p, 480M
   |__ Port 1: Dev 2, If 0, Class=Hub, Driver=hub/5p, 480M
      |__ Port 1: Dev 3, If 0, Class=Vendor Specific Class, Driver=sm95xx, 480M
      |__ Port 2: Dev 4, If 0, Class=Human Interface Device, Driver=usbhid, 1.5M } キーボード
      |__ Port 2: Dev 4, If 1, Class=Human Interface Device, Driver=usbhid, 1.5M }
      |__ Port 3: Dev 5, If 0, Class=Human Interface Device, Driver=usbhid, 1.5M   マウス
      |__ Port 5: Dev 6, If 0, Class= Hub, Driver=hub/4p, 480M                SUGOI HUB
         |__ Port 1: Dev 7, If 0, Class= Vendor Specific Class, Driver=ftdi_sio, 12M リーダ・ライタ
```

4.5. コマンド

リーダー・ライタのポートを OFF するときは下のコマンドを実行します。

```
hub-ctrl -b 1 -d 6 -P 1 -p 0    (-b は Bus の番号、-d は Device の番号、-P は Port の番号、
                               -p は 0 が OFF で 1 が ON)
```

リーダー・ライタのポートを ON するときは下のコマンドを実行します。

```
hub-ctrl -b 1 -d 6 -P 1 -p 1
```

特権が必要なため

```
sudo sh -c " * * * * * "
```

で行う必要があるので実際には下のようになります。

```
sudo sh -c "hub-ctrl -b 1 -d 6 -P 1 -p 0"
```

```
sudo sh -c "hub-ctrl -b 1 -d 6 -P 1 -p 1"
```

4.6. アプリケーションで実行

アプリケーションで実行するには、`subprocess` を使います。

始めに

```
import subprocess
```

を宣言しておき、アプリケーション内で下のようにします。

OFF するときは

```
subprocess.call("sudo sh -c ¥"hub-ctrl -b 1 -d 6 -P 1 -p 0¥", shell=True)
```

ON するときは

```
subprocess.call("sudo sh -c ¥"hub-ctrl -b 1 -d 6 -P 1 -p 1¥", shell=True)
```

4.7. サンプルアプリケーション

```

import serial
import time
import subprocess          # subprocess を使うという宣言

for n in range(100):      # 以下を 100 回繰り返す
    time.sleep(5)         # 5 行待ち
    ser = serial.Serial("/dev/ttyUSB0")
    ser.baudrate = 19200
    sendData = b'\xBB\x80\x22\x00\x02\x01\xA1\x7E'
    print(n)
    ser.write(sendData)
    while (True):
        recvData = ser.read(5)
        len1 = recvData[3]*256+recvData[4]+1
        temp1 = ser.read(len1)
        if (recvData[2] == 0xFF):
            print('No tag!!')
            break
        elif (recvData[2] == 0x27):
            break
        else:
            len2 = len(temp1)
            temp2 = temp1[2:len2-4]
            print(temp2.hex())
    ser.close()
    time.sleep(5)         # 5 行待ち
    subprocess.call("sudo sh -c '\hub-ctrl -b 1 -d 6 -P 1 -p 0'", shell=True) # OFF
    print('\nOFF\n')
    time.sleep(5)         # 5 行待ち
    subprocess.call("sudo sh -c '\hub-ctrl -b 1 -d 6 -P 1 -p 1'", shell=True) # ON
    print('ON\n')
    
```

USB オープン
 タグ読み取りコマンドを送信
 読み取り結果を受信&表示
 USB クローズ

4.8. 注意事項

前述では SUGOI HUB がデバイス番号 6 になっていましたが、ラズパイの USB ポートから SUGOI HUB を抜き差しするとデバイス番号が変更されます。

そのため、抜き差しをしないようにするか、現在どのデバイス番号になっているかをその都度調べて対応します。

どうしても、自動的に SUGOI HUB のデバイス番号を入手してリーダー・ライタのポートを OFF/ON しなければならない場合は次章をご覧ください。

5. 自動的に SUGOI HUB のデバイス番号を入手してリーダ・ライタのポートだけを OFF/ON

5.1. 概要

```
$ lsusb
```

を実行すると下のように SUGOI HUB の情報が表示されます。

```
Bus 001 Device 006: ID 0409:005a NEC Corp. HighSpeed Hub
```

この ID (0409:005a) を手掛かりにデバイス番号を入手し、そのデバイス番号を使って OFF/ON します。

5.2. 事前準備

下の2つのシェルを作成します。

(1) off.sh --- SUGOI HUB のデバイス番号を入手してリーダ・ライタのポートを OFF するシェル

```
#!/bin/sh
```

```
DEV=`lsusb | grep '0409:005a' | tail -1 | cut --delimiter=' ' --fields=4 | tr --delete ':'`  
sudo sh -c "hub-ctrl -b 1 -d $DEV -P 1 -p 0"
```

(2) on.sh --- SUGOI HUB のデバイス番号を入手してリーダ・ライタのポートを ON するシェル

```
#!/bin/sh
```

```
DEV=`lsusb | grep '0409:005a' | tail -1 | cut --delimiter=' ' --fields=4 | tr --delete ':'`  
sudo sh -c "hub-ctrl -b 1 -d $DEV -P 1 -p 1"
```

パーミッションを変更します。

```
$ sudo chmod +x off.sh
```

```
$ sudo chmod +x on.sh
```

これらのシェルをアプリから呼びます。

5.3. サンプルアプリケーション

```

import serial
import time
import subprocess          # subprocess を使うという宣言

for n in range(100):      # 以下を 100 回繰り返す
    time.sleep(5)         # 5 行待ち
    ser = serial.Serial("/dev/ttyUSB0")
    ser.baudrate = 19200
    sendData = b'\xBB\x80\x22\x00\x02\x01\xA1\x7E'
    print(n)
    ser.write(sendData)
    while (True):
        recvData = ser.read(5)
        len1 = recvData[3]*256+recvData[4]+1
        temp1 = ser.read(len1)
        if (recvData[2] == 0xFF):
            print('No tag!!')
            break
        elif (recvData[2] == 0x27):
            break
        else:
            len2 = len(temp1)
            temp2 = temp1[2:len2-4]
            print(temp2.hex())
    ser.close()
    time.sleep(5)          # 5 行待ち
    subprocess.call("sh off.sh", shell=True)    # OFF
    print('\nOFF\n')
    time.sleep(5)          # 5 行待ち
    subprocess.call("sh on.sh", shell=True)     # ON
    print('ON\n')
    
```

USB オープン
 タグ読み取りコマンドを送信
 読み取り結果を受信&表示
 USB クローズ

以上