

私の覚書なのだ

AMBEserverインストール手順(AMBE3000R基板用)

Raspberry Pi の GPIO 操作でお世話になる「Wiring Pi」は、Bullseye ベースの Rasperry Pi OS のリポジトリから無くなりました。

とにかく古い 3000R基板は標準でシリアル通信速度は230400bpsです。

3000R基板を改造して460800bpsにした場合、起動スクリプトはAMBEserverGPIOを使用します。ここで私には！！「Wiring Pi」が必要なのです。

起動スクリプトをAMBEserverに変更すれば、USB接続タイプのThumbDV等でも動作すると思います。(私は持っていないので理論上の想像です。)

OS周りのバージョン、外部接続機器の違いにより動作不良がでるかも知れません。本手順による作業においてトラブル、損害等が発生しても責任は負えません。

※raspbian-bullseyeが入った状態で既にラズパイが起動していることが前提です。

===== for RaspberryPi3 B+を使用する場合 =====

GUIの場合、RaspberryPiデスクトップ画面のメニューバーより

設定-->RaspberryPiの設定-->インターフェースで

Serial Portを有効、Serial Consoleを無効にした後、再起動した方が

/boot/cmdline.txtの console=serial0,115200

の部分の削除が不要となり楽かも！？。

下記コマンドでserial-gettyを無効化する。

\$ sudo systemctl disable serial-getty@ttyAMA0.service

\$ sudo reboot

ttyAMA0シリアル通信コンソール無効化

\$ sudo nano /boot/cmdline.txt

#dwc_otg.lpm_enable=0 console=serial0,115200 console=tty1 root=.....省略.....plymouth.ignore-serial- consoles

dwc_otg.lpm_enable=0 console=tty1 root=.....省略.....plymouth.ignore-serial- consoles

rootwait net.ifnames=0

1行目を #でコメントアウト 2行目にconsole=serial0,115200 の記述を削除したものを追加する。か もしくは、console=serial0,115200 の記述を削除する。

RaspberryPi3で旧networkデバイス名使用可能にするために最終行に

rootwait net.ifnames=0 を追加、

saveしてnanoを終了

```
$ sudo nano /boot/config.txt
```

最終行に下記を追加してRaspberryPi3でttyAMA0を使用するBluetoothを使用不可にする
(これを行わないとGPIOでttyAMA0が使用出来ない)

```
dtoverlay=pi3-disable-bt
```

と

```
enable_uart=1
```

もあることを確認して、無ければ追加記入した上で、
saveしてnanoを終了。再起動する。

```
$ sudo reboot
```

定番通りUPDATE <--時間がかかる

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

再起動する。(変更がなければ必要ない)

```
$ sudo reboot
```

AMBEserverのソースダウンロード

```
$ sudo apt install -y git
```

```
$ sudo git clone https://github.com/dl5di/OpenDV
```

Wiring Piのインストール

Wiring Piは、ラズパイでGPIOを制御するのに必要なC言語用のライブラリ。
C言語でGPIOを使うのにこのライブラリが必要。

WiringPiをインストールするとgpioというコマンドもインストールされる。
このコマンドを使うと、シェル上でGPIOを制御することができる。

Wiring Pi のソースコードを Git で入手するための下準備

```
$ sudo apt-get install libi2c-dev
```

```
$ sudo apt-get install git-core
```

ユーザーのホームディレクトリに移動し、ソースコードを入手

```
$ cd
```

```
$ sudo git clone https://github.com/WiringPi/WiringPi.git
```

Wiring Pi のソースコードが wiringPiディレクトリとしてクローンされるので、ディレクトリ内に移動してビルドする。

```
$ cd WiringPi
$ sudo ./build
```

gccコマンドでコンパイルして実行。

※ 「-lwiringPi」で、ライブラリをリンクする必要がある。

(例) LIBS := -lwiringPi (下記のMakefile)

完了後DV3000ディレクトリに移動

```
-----
$ cd ~/OpenDV/DummyRepeater/DV3000
-----
```

Makefileの作成

```
-----
$ sudo nano Makefile
```

-----Makefileの内容(以下を記述し、保存する)-----

```
all:      dv3000d AMBEServer AMBEServerGPIO
```

```
GCC := gcc
```

```
CFLAGS := -O2 -DRASPBERRY_PI -Wall
```

```
CFLAGS2 := -O2 -Wall
```

```
LIBS := -lwiringPi
```

```
dv3000d: dv3000d.o
```

```
$(GCC) $(LD_FLAGS) -o dv3000d dv3000d.o $(LIBS)
```

```
dv3000d.o:      dv3000d.c
```

```
$(GCC) $(CFLAGS) -c dv3000d.c
```

```
AMBEServer:     AMBEServer.o
```

```
$(GCC) $(LD_FLAGS) -o AMBEServer AMBEServer.o
```

```
AMBEServer.o:   AMBEServer.c
```

```
$(GCC) $(CFLAGS2) -c AMBEServer.c
```

```
AMBEServerGPIO: AMBEServerGPIO.o
```

```
$(GCC) $(LD_FLAGS) -o AMBEServerGPIO AMBEServerGPIO.o $(LIBS)
```

```
AMBEServerGPIO.o:      AMBEServer.c
```


dv3000dの起動スクリプトを編集

(AMBEserverGPIO/AMBEserverで動かす場合は、最後の方の参考事項参照)

```
-----
$ sudo nano /etc/init.d/dv3000d
-----
```

/usr/bin/dv3000d -d の行を下記に変更(port:2465の場合)

```
/usr/bin/dv3000d -d -p 2465
```

saveしてnanoを終了

手動で起動して見る

```
-----
$ sudo /usr/bin/dv3000d -d -p 2465
$ ps -C dv3000d
-----
```

下記の表示が出れば起動OK

```
PID TTY      TIME CMD
1234 ?        00:00:02 dv3000d <---1234は例
```

起動確認後

```
-----
sudo kill PID番号      (例 $ sudo kill 1234)
-----
```

自動起動の設定

```
-----
$ sudo apt-get install chkconfig
$ sudo chkconfig dv3000d on
$ sudo chkconfig dv3000d --list
-----
```

下記の表示が出れば自動起動設定OK

```
dv3000d                0:off 1:off 2:on  3:on  4:on  5:on  6:off
```

再起動する

```
-----
$ sudo reboot
-----
```

立ち上がったら起動確認

```
-----  
$ ps -C dv3000d  
-----
```

下記の表示が出れば起動OK

```
PID TTY      TIME CMD  
1234 ?      00:00:02 dv3000d
```

うまく立ち上がらなければ起動スクリプト等をチェックする

<参考>

dv3000d停止

```
sudo service dv3000d stop
```

dv3000d開始

```
sudo service dv3000d start
```

dv3000d再起動

```
sudo service dv3000d restart
```

自動起動の停止

```
sudo service dv3000d stop
```

```
sudo chkconfig dv3000d off
```

以上、

===== 参考事項1 AMBEServerGPIOで動かす場合 =====

ホームディレクトリに移動

```
$ cd
```

AMBEServerGPIOの起動スクリプトを編集

```
$ sudo nano /etc/init.d/AMBEServerGPIO
```

/usr/bin/AMBEServerGPIO -d -i /dev/ttyAMA0 の行を下記に変更(port:2465、speed:460800bpsの場合)

```
/usr/bin/AMBEServerGPIO -d -p 2465 -s 460800 -i /dev/ttyAMA0
```

saveしてnanoを終了

手動で起動して見る

```
$ sudo /usr/bin/AMBEServerGPIO -d -p 2465 -s 460800 -i /dev/ttyAMA0
$ ps -C AMBEServerGPIO
```

下記の表示が出れば起動OK

```
PID TTY      TIME CMD
1234 ?      00:00:02 AMBEServerGPIO <---1234は例
```

起動確認後

```
sudo kill PID番号      (例 $ sudo kill 1234)
```

自動起動の設定 (bullseye 64bitでは使えないchkconfig)

```
$ sudo apt-get install chkconfig
$ sudo chkconfig AMBEServerGPIO on
$ sudo chkconfig AMBEServerGPIO --list
```

chkconfig(apt-get install chkconfig)を使用すると、起動時にデーモンを自動的に起動できる。 しかし、最新のOSでは、「chkconfig」は使用できない。(rc.local or systemctl)

下記の表示が出れば自動起動設定OK

```
AMBEServerGPIO      0:off 1:off 2:on  3:on  4:on  5:on  6:off
```

再起動する

```
-----
$ sudo reboot
-----
```

立ち上がったら起動確認

```
-----
$ ps -C AMBEServerGPIO
-----
```

下記の表示が出れば起動OK

```
PID TTY      TIME CMD
1234 ?      00:00:02 AMBEServerGPIO
```

うまく立ち上がらなければ起動スクリプト等をチェックする。

起動方法はport=2465の場合

```
sudo /usr/bin/AMBEServerGPIO -d -p 2465 -s 460800 -i /dev/ttyAMA0
```

「chkconfig」での自動起動が上手くいかない場合には、

「sudo nano /etc/rc.local」 に次の起動コマンドを記述すれば自動起動する。

```
sudo service AMBEServerGPIO start
```

<参考>

AMBEServerGPIO停止

```
$ sudo service AMBEServerGPIO stop
```

AMBEServerGPIO開始

```
$ sudo service AMBEServerGPIO start
```

AMBEServerGPIO再起動

```
$ sudo service AMBEServerGPIO restart
```

自動起動の停止

```
$ sudo service AMBEServerGPIO stop
```

```
$ sudo chkconfig AMBEServerGPIO off
```

=====参考事項2 USB接続タイプのThumbDV等で動かす場合=====

ホームディレクトリに移動

```
cd
```

AMBEserverの起動スクリプトを編集

```
sudo nano /etc/init.d/AMBEserver
```

/usr/bin/AMBEserver -d -i /dev/ttyUSB0 の行を下記に変更(port:2460の場合)

```
/usr/bin/AMBEserver -d -p 2460 -i /dev/ttyUSB0
```

saveしてnanoを終了

手動で起動して見る

```
sudo /usr/bin/AMBEserver -d -p 2460 -i /dev/ttyUSB0
ps -C AMBEserver
```

下記の表示が出れば起動OK

```
PID TTY      TIME CMD
1234 ?      00:00:02 AMBEserver <---1234は例
```

起動確認後

```
sudo kill PID番号      (例 $ sudo kill 1234)
```

自動起動の設定

```
sudo apt-get install chkconfig
sudo chkconfig AMBEserver on
sudo chkconfig AMBEserver --list
```

下記の表示が出れば自動起動設定OK

```
AMBEserver      0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

再起動する

```
-----
sudo reboot
-----
```

立ち上がったら起動確認

```
-----
ps -C AMBEServer
-----
```

下記の表示が出れば起動OK

```
PID TTY      TIME CMD
1234 ?      00:00:02 AMBEServer
```

うまく立ち上がらなければ起動スクリプト等をチェックする

起動方法はport=2460の場合

```
sudo /usr/bin/AMBEServer -d -p 2460 -i /dev/ttyUSB0
```

「chkconfig」での自動起動が上手くいかない場合には、
「sudo nano /etc/rc.local」 に次の起動コマンドを記述すれば自動起動します。

```
sudo service AMBEServer start
```

<参考>

AMBEServer停止

```
sudo service AMBEServer stop
```

AMBEServer開始

```
sudo service AMBEServer start
```

AMBEServer再起動

```
sudo service AMBEServer restart
```

自動起動の停止

```
sudo service AMBEServer stop
```

```
sudo chkconfig AMBEServer off
```

=====参考事項終わり=====

Wiring Piのインストール&設定

Wiring Piとは、ラズパイでGPIOを制御するのに必要なC言語用のライブラリ。
C言語でGPIOを使うのにこのライブラリが必要なので最初にインストール。
WiringPiをインストールするとgpioというコマンドもインストールされる。
このコマンドを使うと、シェル上でGPIOを制御することができる。

Wiring Pi のソースコードを Git で入手するための下準備

```
$ sudo apt-get install libi2c-dev
$ sudo apt-get install git-core
```

ユーザーのホームディレクトリに移動し、Wiring Pi のソースコードを入手

```
$ cd ~
$ git clone https://github.com/WiringPi/WiringPi.git
```

Wiring Pi のソースコードが wiringPiディレクトリとしてクローンされますので、ディレクトリ内に移動してビルドする。

```
$ cd WiringPi
$ ./build
```

C言語を使ってGPIO端子を制御する。

gccコマンドでコンパイルして実行する。

※ 「-lwiringPi」で、ライブラリをリンクする必要がある。

-----Makefileの内容(例1)-----

```
all:          dv3000d AMBEserver AMBEserverGPIO
```

```
GCC := gcc
```

```
CFLAGS := -O2 -DRASPBERRY_PI -Wall
```

```
CFLAGS2 := -O2 -Wall
```

```
LIBS := -lwiringPi
```

-----Makefileの内容(例2)-----

```
$ gcc -o exe_LED LED.c -lwiringPi
```