

Raspberry Pi リフレクター開設体験記

Shu JA3GQJ

はじめに

6月21日 JF2ANH 阪野さんから頂いたリフレクターの image を SD カードに焼き付けて早速手持ちの 2~3 年前に購入した古いラズベリーパイに装着して電源を入れたら問題なく走りだし、指定した URL でダッシュボードも表示されました。そこで勉強を兼ねて阪野さんの image を参考にしながらラズベリーパイの OS のインストールからリフレクターソフトのインストールまで一から始めることにしました。作者、ラック (Luc) さんの資料では「The software packages for Linux are tested on Debian7 (Wheezy) 32 and 64bit or newer. Raspbian will work but is not recommended. 」とありDebian(Wheezy)のimageをSDカードに焼き付けて最近購入した新しいPi3でソフトのインストールを始めましたが途中でコマンドを受け付けなくなってimageの焼き直しを何度やっても同じコマンドのところでNGとなりどうしようもなかったもので、ラックさん「Raspbianは動作するかもしれないが推奨は出来ない」と言っていますが背に腹は代えられず今度はRaspbian JESSIEのimageを焼き付けたSDカードを使用したところ問題なくソフトをインストールすることが出来て動作もしましたのでその手順を以下にまとめてみます。

前提条件

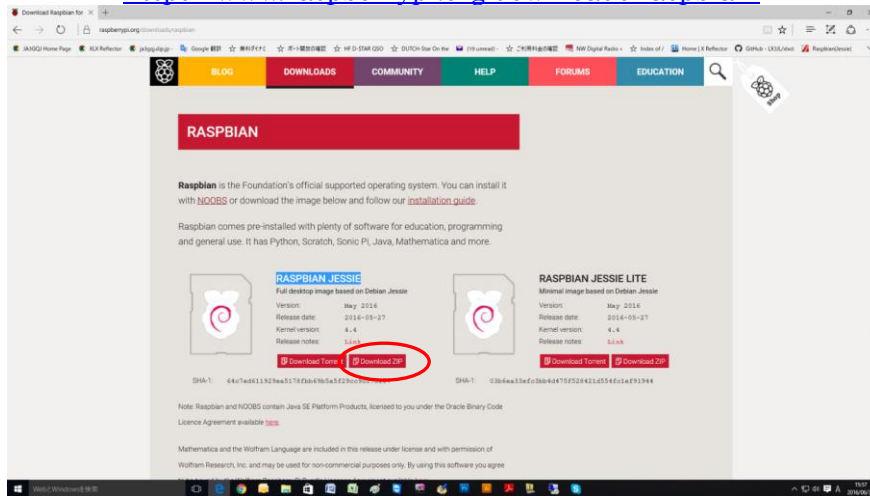
- ・ラズベリーパイにはキーボード、マウス、ディスプレイを接続して、LANは有線接続します。
- ・ラズベリーPi3を自宅サーバーにするためNTT光回線に付属するルーターのポートが次の通り開放（マッピングとかフォワーディングともいわれる）されている事。カッコ内は用途で、中には開放しなくてもよいポートが含まれているかも知れませんが当局は詳しいことが判らないのですべてのポートを開放しました。
TCP port 80 (http) 、TCP port 22 (ssh)、 TCP port 443 (https) 、TCP port 10022 (ssh)
TCP port 8080 (RepNet)
UDP port 10001 (json interface XLX Core)、UDP port 10002 (XLX interlink)
UDP port 20001 (DPlus protocol)、UDP port 30001 (DExtra protocol)
UDP port 30051 (DCS protocol)
- ・ルーターのポートを開放する方法はそれぞれのメーカーの取扱説明書やウェブページを参考にする。
- ・自宅サーバーのグローバル IP アドレスは <http://ieserver.net/> の無料ダイナミックDNS(DDNS)サービスを利用して取得する。また、ラズベリーパイのローカル IP アドレスは一般的な自動割り付け(DHCP)ではなく手動設定の固定アドレスにする。手動設定の方法は4ページ参照。
- ・プロバイダーの都合やルーターの電源が落ちると使用するグローバル IP アドレスが変わってしまうのでリフレクターの公開はドメイン名 (<http://ja3gq.dip.jp> など) で行う。
- ・リフレクター番号 (XLX081 等) は dvs@rls.lu にメールをして出来れば自分が希望する番号を取得する。
- ・image を SD カードに焼き付けるソフト Win32 Disk Imager を予め Windows パソコンにインストールしておく。
- ・SD カードによって稼働するラズベリーパイの連続運転の耐久時間の経験は皆無です。

- ・使用できる SD カードの容量は最低 8 GB ではないかと思いますが当局は現在 16GB を使用しています。

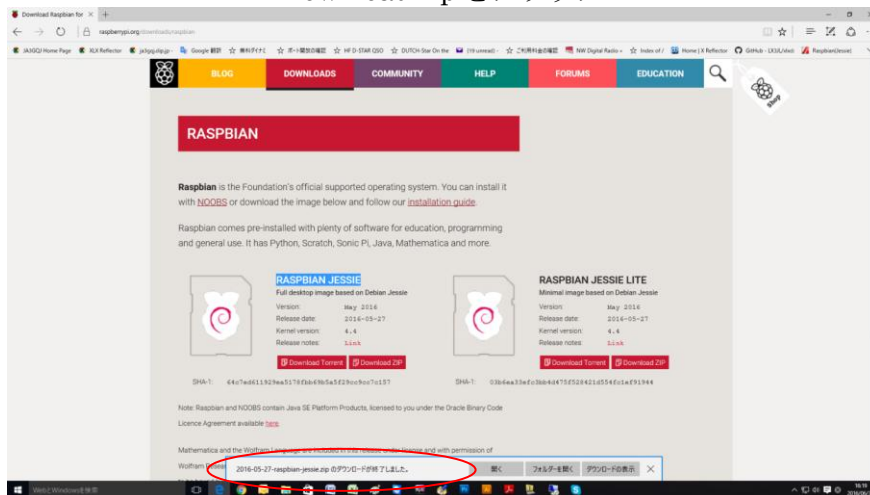
OS の image を焼く

下記から RASPBIAN JESSIE の ZIP ファイルをダウンロードする(当局は約 25 分かかった)

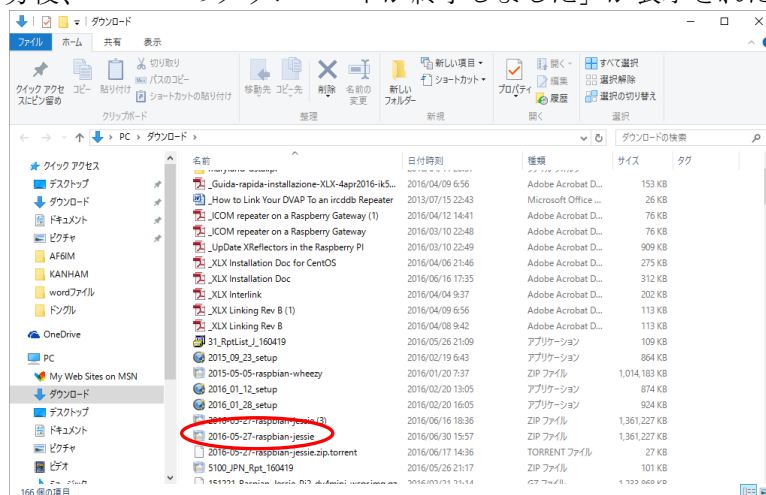
<https://www.raspberrypi.org/downloads/raspbian/>



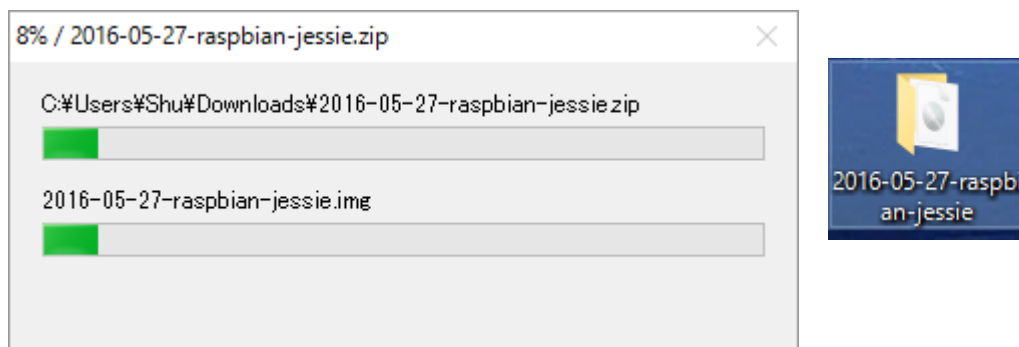
Download zip をクリック



約 25 分後、「・・・のダウンロードが終了しました」が表示されたら OK



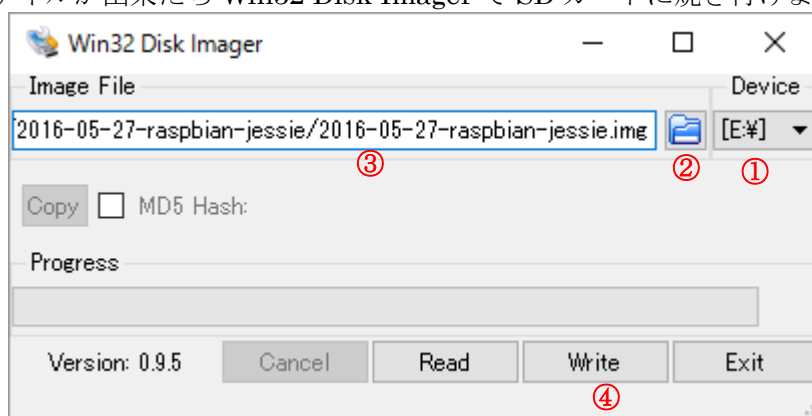
当局の場合ダウンロード先はデフォルトのダウンロードフォルダーなのでそこにある「2016-05-27-raspbian-jessie.zip」を解凍ソフト Lhasa 等を使って「2016-05-27-raspbian-jessie.img」という image ファイルを作る。



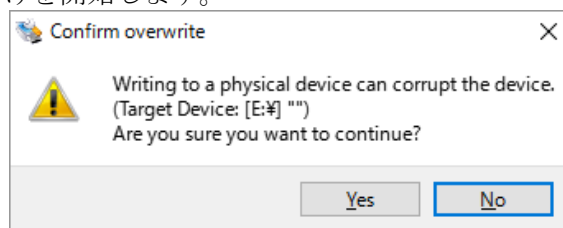
当局の場合「2016-05-27-raspbian-jessie.img」という image ファイルがデスクトップの「2016-05-27-raspbian-jessie」というフォルダー（右上）の中に生成されました。



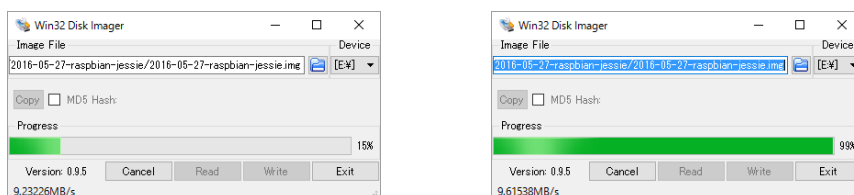
image ファイルが出来たら Win32 Disk Imager で SD カードに焼き付けます。



Win32 Disk Imager 立ち上げて SD カードを Windows パソコンに挿入すると自動でデバイス(この例では E:①)が認識されます。その左隣のフォルダーのアイコン(②)をクリックして先ほど生成した「2016-05-27-raspbian-jessie」というフォルダーの中にある「2016-05-27-raspbian-jessie.img」という image ファイル(③)を選択したら「Write」(④)をクリックして焼き付けを開始します。

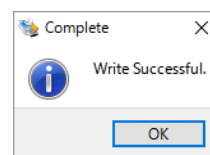


先に何か書き込まれた SD カードに焼き付ける場合は上の図のようなメッセージ画面が出ますので消してもよければ「Yes」をクリックして先に進みます



上の図のように緑色のプログレスバーが表示されて焼き付けの進行状態が示されます。暫く待つと緑色のプログレスバーが空白を埋めて（上の図右）焼き付けが終了します。

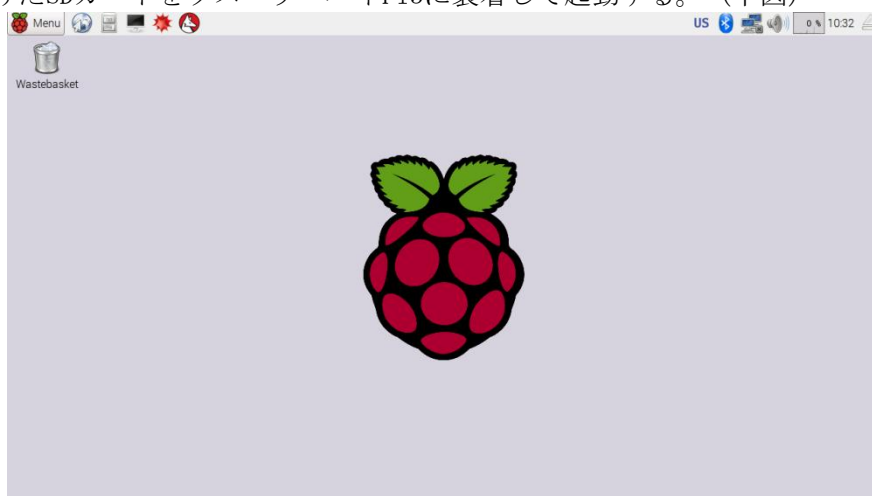
右の図は焼き付けが成功して完了したことを知らせるポップアップ画面です「OK」をクリックして閉じます



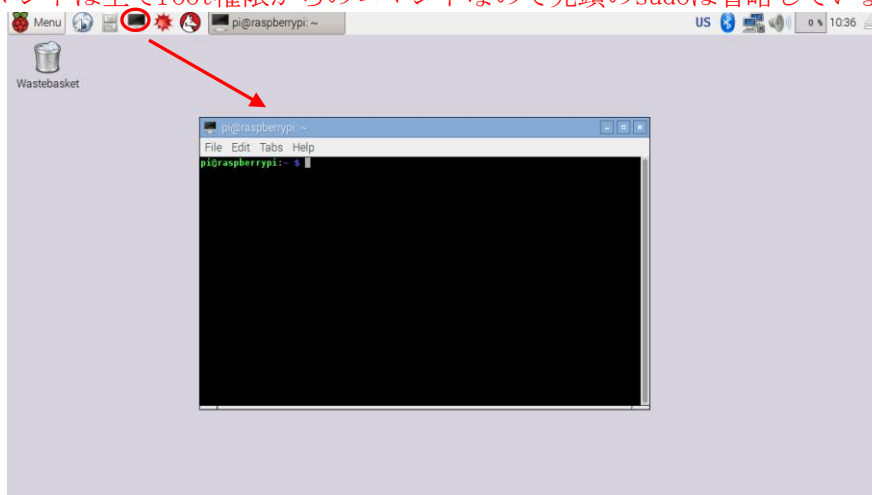
Win32 Disk Imager のダウンロードは 17 ページ参照

ラズベリーパイ固定 IP アドレス設定

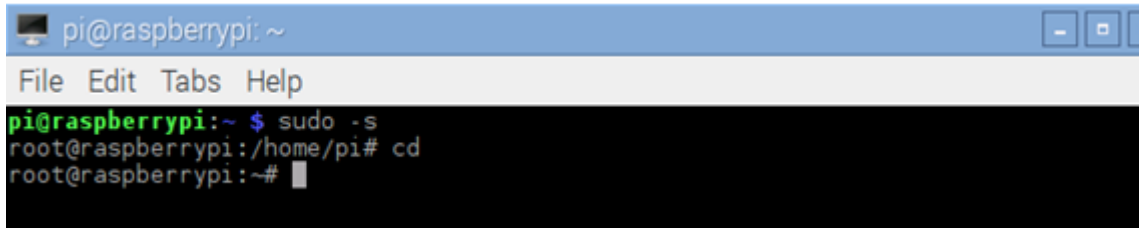
焼き付けたSDカードをラズベリーパイPi3に装着して起動する。（下図）



- 起動後コマンド入力画面のLXterminal（下図）を開いてroot権限に切り替える。
以降コマンドは全てroot権限からのコマンドなので先頭のsudoは省略しています。



sudo -s root権限に切り替えるコマンド



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ sudo -s  
root@raspberrypi:/home/pi# cd  
root@raspberrypi:~#
```

pico /etc/dhcpd.conf

開いた Terminal 画面の最終行の次に有線、無線のそれぞれを追記

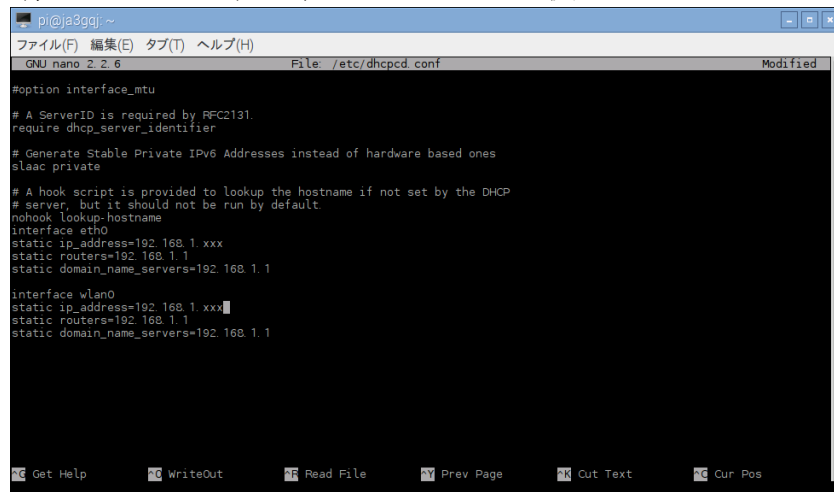
有線 LAN 用

```
Interface eth0  
Static ip_address=192.168. x. yyy  
Static routers=192.168. x. 1  
Static domain_name_servers=192.168. x. 1
```

無線 LAN 用

```
Interface wlan0  
Static ip_address=192.168. x. yyy  
Static routers=192.168. x. 1  
Static domain_name_servers=192.168. x. 1
```

有線、無線 にかかわらず通常 x には 0 か 1 が使われているケースが多い。



```
pi@jaggy: ~  
ファイル(F) 編集(E) タブ(T) ヘルプ(H)  
GNU nano 2.2.6 File: /etc/dhcpd.conf Modified  
#option interface_mtu  
# A ServerID is required by RFC2131.  
require dhcp_server_identifier  
# Generate Stable Private IPv6 Addresses instead of hardware based ones  
slaac private  
# A hook script is provided to lookup the hostname if not set by the DHCP  
# server, but it should not be run by default.  
nohook lookup-hostname  
interface eth0  
static ip_address=192.168.1.xxx  
static routers=192.168.1.1  
static domain_name_servers=192.168.1.1  
  
interface wlan0  
static ip_address=192.168.1.xxx  
static routers=192.168.1.1  
static domain_name_servers=192.168.1.1
```

XLX マルチプロトコルリフレクターゲートウェイをインストールする

- 次にOSのアップデートとアップグレードを行うが新しいソフトをインストールする前の呪いみたいな物と思っています。

```
# apt-get update
```

```
# apt-get upgrade
```

- 続いてギットをインストールする

```
# apt-get install git git-core
```

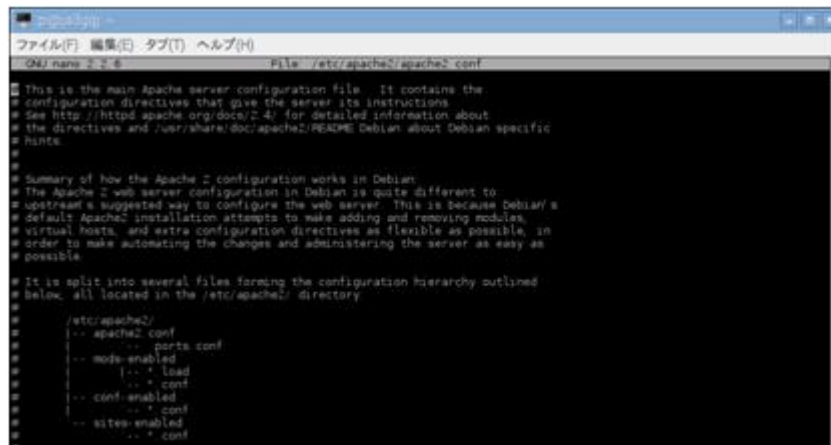
- 続いてウェブサーバーと PHP をインストールする。

```
# apt-get install apache2 php5
```

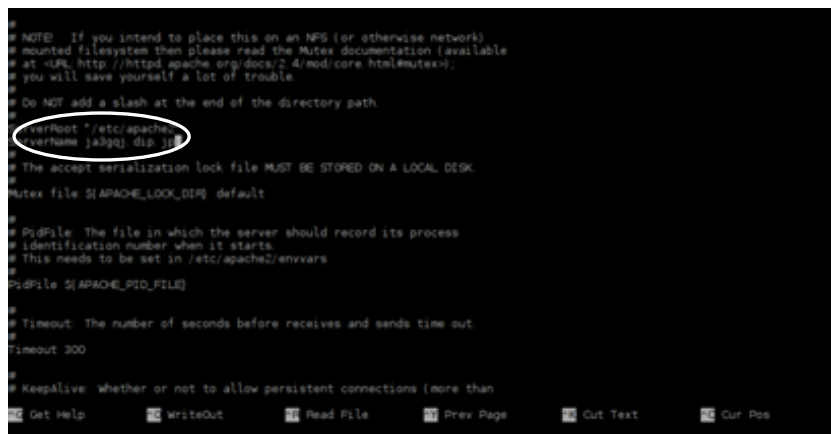
- 続いて apache2 にサーバー名を設定する。

```
# pico /etc/apache2/apache2.conf
```

Pico は当局にとって初めてのテキストエディターですが nano と同じような感じがします。そういえばピコは10のマエナス6乗ナノは10のマエナス12乗ですね。



```
File: /etc/apache2/apache2.conf
This is the main Apache server configuration file. It contains the
configuration directives that give the server its instructions
See http://httpd.apache.org/docs/2.4/ for detailed information about
the directives and /usr/share/doc/apache2/README.Debian about Debian specific
hints.
Summary of how the Apache 2 configuration works in Debian:
The Apache 2 web server configuration in Debian is quite different to
upstream's suggested way to configure the web server. This is because Debian's
default Apache2 installation attempts to make adding and removing modules,
virtual hosts, and extra configuration directives as flexible as possible, in
order to make automating the changes and administering the server as easy as
possible.
It is split into several files forming the configuration hierarchy outlined
below, all located in the /etc/apache2/ directory
/etc/apache2/
-- apache2.conf
-- ports.conf
-- mods-enabled
-- * load
-- * conf
-- conf-enabled
-- * conf
-- sites-enabled
-- * conf
```



```
NOTE! If you intend to place this on an NFS (or otherwise network)
mounted filesystem then please read the Mutex documentation (available
at <URL http://httpd.apache.org/docs/2.4/mod/core.html#mutex>);
you will save yourself a lot of trouble.
Do NOT add a slash at the end of the directory path
ServerRoot "/etc/apache2/"
# The accept serialization lock file MUST BE STORED ON A LOCAL DISK.
Mutex file:${APACHE_LOCK_DIR} default
PidFile: The file in which the server should record its process
identification number when it starts.
This needs to be set in /etc/apache2/envvars
PidFile ${APACHE_PID_FILE}
Timeout: The number of seconds before receives and sends time out
Timeout 300
KeepAlive: Whether or not to allow persistent connections (more than
Get Help WriteOut Read File Prev Page Cut Text Cur Pos
```

上の図で ServerRoot が” /etc/apache2” であることを確認して ServerName が設定されていなければ例えば ja3gqj.dip.jp のように入力して保存する。

- 続いて g++ コンパイラーをインストールする

```
# apt-get install build-essential
# apt-get install g++-4.7
```
- 続いて XLX ソースコードをインストールしてコンパイルする

```
# git clone https://github.com/LX3JL/xlxd.git
# cd xlxd/src/
# make
# make clean
# make install
```
- 続いてスタートアップスクリプト XLXD を/etc/init.d へコピーする

```
# cp ~/xlxd/scripts/xlxd /etc/init.d/xlxd
```
- 続いてルートディレクトリーに callhome フォルダを作って書き込み権限を与える。

```
# mkdir callhome
# chmod 777 callhome
```

- ・続いてデフォルトのスタートアップパラメータの記述を変更する
 - # pico /etc/init.d/xlxd このコマンドでスタートアップパラメーターを開く
 - ① ARGUMENTS を自分のローカル IP アドレスに変える

```

# start script by LX1IQ
### END INIT INFO

PATH=/sbin:/bin:/usr/sbin:/usr/bin

. /lib/lsb/init-functions

# change below settings according to your system
NAME="xlxd"
DAEMON="/usr/sbin/xlxd"
ARGUMENTS="XLX081 192.168.1.xxx"
PIDFILE="/var/log/xlxd.pid"
USER=root
GROUP=root

[ -f $DAEMON ] || exit 0

start() {

```

- ② 次のところにシャープマークを付けて別の赤字の記述を追加する
 - # start-stop-daemon --start --exec \$DAEMON --chuid \$USER:\$GROUP --background -- \$ARGUMENTS
\$DAEMON \$ARGUMENTS
 - # start-stop-daemon --stop --pidfile \$PIDFILE
killproc \$DAEMON

```

# start daemon

echo -n "Starting $NAME: "
start-stop-daemon --start --exec $DAEMON --chuid $USER:$GROUP --background -- $ARGUMENTS
$DAEMON $ARGUMENTS
RETVAL=$?
echo
sleep 4
echo pidof xlxd > $PIDFILE
return $RETVAL

# stop daemon

echo -n "Stopping $NAME: "
start-stop-daemon --stop --pidfile $PIDFILE
killproc $DAEMON
RETVAL=$?
echo

```

- ・続いて次のコマンドで config. in. php を開いて・・・
 - # pico /var/www/html/pgs/ config. in. php

```

<?php
/*
Possible values for IPModus
HideIP
ShowFullIP
ShowLast1ByteOfIP
ShowLast2ByteOfIP
ShowLast3ByteOfIP
*/
$Service = array();
$CallingHome = array();
$PageOptions = array();
$PageOptions['ContactEmail'] = 'scjr3vh[nifty.com]'; // Support E-Mail address
$PageOptions['DashboardVersion'] = '2.3.1'; // Dashboard Version
$PageOptions['PageRefreshActive'] = true; // Activate automatic refresh
$PageOptions['PageRefreshDelay'] = '10000'; // Page refresh time in milliseconds
$PageOptions['RepeatersPage'] = array();
$PageOptions['RepeatersPage']['LimitTo'] = 99; // Number of Repeaters to show
$PageOptions['RepeatersPage']['IPModus'] = 'ShowLast1ByteOfIP'; // See possible options above
$PageOptions['RepeatersPage']['MasqueradeCharacter'] = '*'; // Character used for masquerade
$PageOptions['PeerPage'] = array();
$PageOptions['PeerPage']['LimitTo'] = 99; // Number of peers to show
$PageOptions['PeerPage']['IPModus'] = 'ShowLast1ByteOfIP'; // See possible options above
$PageOptions['PeerPage']['MasqueradeCharacter'] = '*'; // Character used for masquerade
$PageOptions['ModuleNames'] = array();
$PageOptions['ModuleNames']['A'] = 'Int'; // Module nomination
$PageOptions['ModuleNames']['B'] = 'Regional';
$PageOptions['ModuleNames']['C'] = 'National';
$PageOptions['ModuleNames']['D'] = 'Local';

```

- ・・・次の事項を編集する。
 - \$PageOptions ContactEmail 連絡先の e メールアドレスを入れる
 - \$PageOptions RepeaterPage IPModus 表示する IP アドレスの桁数を決める。

\$PageOptions PeerPage IPModus 表示する IP アドレスの桁数を決める

```
$PageOptions['MetaDescription'] = 'XLX is a D-Star Reflector System for Ham Radio Operators.'; // Meta Tag Values, usefull$
$PageOptions['MetaKeywords'] = 'Ham Radio, D-Star, XReflector, XLX, XRF, DCS, REF, '; // Meta Tag Values, usefull$
$PageOptions['MetaAuthor'] = 'LX1IQ'; // Meta Tag Values, usefull$
$PageOptions['MetaRevisit'] = 'After 30 Days'; // Meta Tag Values, usefull$
$PageOptions['MetaRobots'] = 'index, follow'; // Meta Tag Values, usefull$
$Service['PIDFile'] = '/var/log/xlxd.pid';
$Service['XMLFile'] = '/var/log/xlxd.xml';
$CallingHome['Active'] = true; // xlx phone home, true or false
$CallingHome['MyDashboardURL'] = 'http://ja3gqj.dip.jp/index.php'; // dashboard url
$CallingHome['ServerURL'] = 'http://xlxapi.rlx.lu/api.php'; // database server, do not change !!!!
$CallingHome['PushDelay'] = 600; // push delay in seconds
$CallingHome['Country'] = 'Osaka JAPAN'; // Country
$CallingHome['Comment'] = 'First Japanese XLX Multiprotocol Reflector Gateway';
$CallingHome['HashFile'] = '/callhome/callinghome.php'; // Make sure the apache user has read and wr$
$CallingHome['OverrideIPAddress'] = ''; // Insert your IP address here. Leave blank for a$
$CallingHome['InterLinkFile'] = '/xlxd/xlxd.interlink'; // Path to interlink file
```

\$CallingHome Active 無条件で false を true にする。

\$CallingHome MyDashBoardURL 自分のドメイン名を URL にする。

この例では http://ja3gqj.dip.jp/index.php

\$CallingHome Country この例では Osaka JAPAN のように国名を入れる。

\$CallingHome Comment 100字以内の英文でコメントを書く。

- 続いて自動起動およびシャットダウンのためのサービスを宣言する
update-rc.d xlxd defaults
 - 続いて次のコマンドでサービスの起動停止を行う
service xlxd start
service xlxd stop
 - 続いてダッシュボードの内容を/var/wwwへコピーする
cp -r ~/xlxd/dashboard/* /var/www/html
 - 続いてダッシュボードにサーバログファイルへの読み取りアクセス権を与えます
chmod +r /var/log/messages
 - 続いて再起動して自動起動の有無を確認する
reboot
- 再起動後、次のコマンドを実行して稼働状況を確認する。
- # CD /etc/init.d
- # ./xlxd status このコマンドの後、下図のような画面が表示されれば正常に稼働しています

```
pi@ja3gqj: ~
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
root@ja3gqj: /home/pi# cd
root@ja3gqj: ~# cd /etc/init.d
root@ja3gqj: /etc/init.d# ./xlxd status
● xlxd.service - LSB: starts the dstar reflector as a linux service
   Loaded: loaded (/etc/init.d/xlxd)
   Active: active (running) since 水 2016-06-29 20:42:30 JST; 1 weeks 0 days ago
 Process: 31348 ExecStop=/etc/init.d/xlxd stop (code=exited, status=0/SUCCESS)
 Process: 31402 ExecStart=/etc/init.d/xlxd start (code=exited, status=0/SUCCESS)
CGroup: /system.slice/xlxd.service
└─31411 /xlxd/xlxd XLX081 192.168.1.200

7月 07 05:29:08 ja3gqj xlxd[31411]: Sending connect packet to XLX peer XL... C
7月 07 05:29:08 ja3gqj xlxd[31411]: XLX nack packet from XLX052 at 220... 31
7月 07 05:29:09 ja3gqj xlxd[31411]: Sending connect packet to XLX peer XL... C
7月 07 05:29:09 ja3gqj xlxd[31411]: XLX nack packet from XLX052 at 220... 31
7月 07 05:29:10 ja3gqj xlxd[31411]: Sending connect packet to XLX peer XL... C
7月 07 05:29:10 ja3gqj xlxd[31411]: XLX nack packet from XLX052 at 220... 31
7月 07 05:29:11 ja3gqj xlxd[31411]: Sending connect packet to XLX peer XL... C
7月 07 05:29:11 ja3gqj xlxd[31411]: XLX nack packet from XLX052 at 220... 31
7月 07 05:29:12 ja3gqj xlxd[31411]: Sending connect packet to XLX peer XL... C
7月 07 05:29:12 ja3gqj xlxd[31411]: XLX nack packet from XLX052 at 220... 31
Hint: Some lines were ellipsized, use -l to show in full.
root@ja3gqj: /etc/init.d#
```

・正しくて起動していてもLAN接続が出来ていないかLANが正しく接続されていてもルーターのポートが解放されていない場合は次のようなエラーが表示されます。

```
pi@ja3gqj: ~  
ファイル(F) 編集(E) タブ(T) ヘルプ(H)  
pi@ja3gqj: ~ $ sudo -s  
root@ja3gqj: /home/pi# cd  
root@ja3gqj: ~# cd /etc/init.d  
root@ja3gqj: /etc/init.d# ./xlxd status  
● xlxd.service - LSB: starts the dstar reflector as a linux service  
   Loaded: loaded (/etc/init.d/xlxd)  
   Active: active (exited) since 月 2016-07-04 07:51:21 JST; 2min 44s ago  
 Process: 450 ExecStart=/etc/init.d/xlxd start (code=exited, status=0/SUCCESS)  
  
7月 04 07:51:17 ja3gqj xlxd[450]: Starting xlxd:  
7月 04 07:51:17 ja3gqj xlxd[481]: Gatekeeper loaded 1 lines from /xlxd/xl...st  
7月 04 07:51:17 ja3gqj xlxd[481]: Gatekeeper loaded 0 lines from /xlxd/xl...st  
7月 04 07:51:17 ja3gqj xlxd[481]: Gatekeeper loaded 0 lines from /xlxd/xl...nk  
7月 04 07:51:17 ja3gqj xlxd[481]: Error opening socket on port UDP30001 o...00  
7月 04 07:51:17 ja3gqj xlxd[481]: Error opening socket on port UDP20001 o...00  
7月 04 07:51:17 ja3gqj xlxd[481]: Error opening socket on port UDP30051 o...00  
7月 04 07:51:17 ja3gqj xlxd[481]: Error opening socket on port UDP10002 o...00  
7月 04 07:51:17 ja3gqj xlxd[481]: Error starting reflector  
7月 04 07:51:21 ja3gqj systemd[1]: Started LSB: starts the dstar reflecto...e.  
Hint: Some lines were ellipsized, use -l to show in full.  
root@ja3gqj: /etc/init.d# cd  
root@ja3gqj: ~#
```

・起動されておらず、尚且つ、LAN接続が出来ていないかLAN接続が正しくてもルーターのポートが解放されていない場合は次のようなエラーが表示されます。

次のコマンドで手動による確実な起動が出来ます。

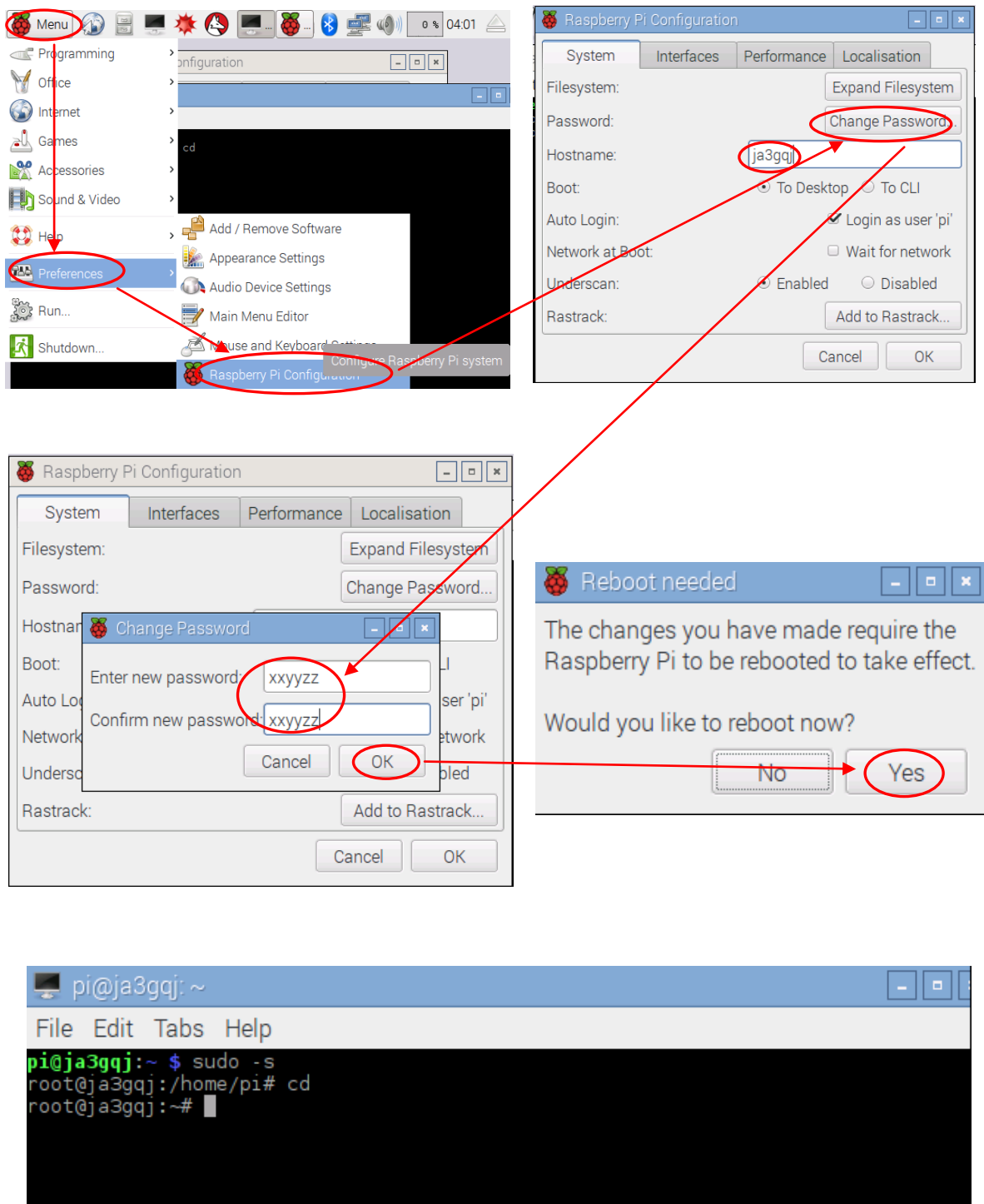
```
# CD /etc/init.d  
# ./xlxd start
```

```
pi@ja3gqj: ~  
ファイル(F) 編集(E) タブ(T) ヘルプ(H)  
Hint: Some lines were ellipsized, use -l to show in full.  
root@ja3gqj: /etc/init.d# cd  
root@ja3gqj: ~# cd /etc/init.d  
root@ja3gqj: /etc/init.d# ./xlxd stop  
[ ok ] Stopping xlxd (via systemctl): xlxd.service.  
root@ja3gqj: /etc/init.d# ./xlxd status  
● xlxd.service - LSB: starts the dstar reflector as a linux service  
   Loaded: loaded (/etc/init.d/xlxd)  
   Active: inactive (dead) since 月 2016-07-04 08:19:21 JST; 16s ago  
 Process: 1127 ExecStop=/etc/init.d/xlxd stop (code=exited, status=0/SUCCESS)  
 Process: 450 ExecStart=/etc/init.d/xlxd start (code=exited, status=0/SUCCESS)  
  
7月 04 07:51:17 ja3gqj xlxd[481]: Gatekeeper loaded 0 lines from /xlxd/xl...nk  
7月 04 07:51:17 ja3gqj xlxd[481]: Error opening socket on port UDP30001 o...00  
7月 04 07:51:17 ja3gqj xlxd[481]: Error opening socket on port UDP20001 o...00  
7月 04 07:51:17 ja3gqj xlxd[481]: Error opening socket on port UDP30051 o...00  
7月 04 07:51:17 ja3gqj xlxd[481]: Error opening socket on port UDP10002 o...00  
7月 04 07:51:17 ja3gqj xlxd[481]: Error starting reflector  
7月 04 07:51:21 ja3gqj systemd[1]: Started LSB: starts the dstar reflecto...e.  
7月 04 08:19:21 ja3gqj systemd[1]: Stopping LSB: starts the dstar reflecto...e.  
7月 04 08:19:21 ja3gqj xlxd[1127]: Stopping xlxd:  
7月 04 08:19:21 ja3gqj systemd[1]: Stopped LSB: starts the dstar reflecto...e.  
Hint: Some lines were ellipsized, use -l to show in full.  
root@ja3gqj: /etc/init.d#
```

ホスト名とパスワードの変更

次の手順でデフォルトのホスト名、Raspberryからja3gqjに変更しました。

矢印の順番に進み最後の「Yes」クリックで再起動します。



ホスト名が Rspaberrypi から ja3gqj に変わった

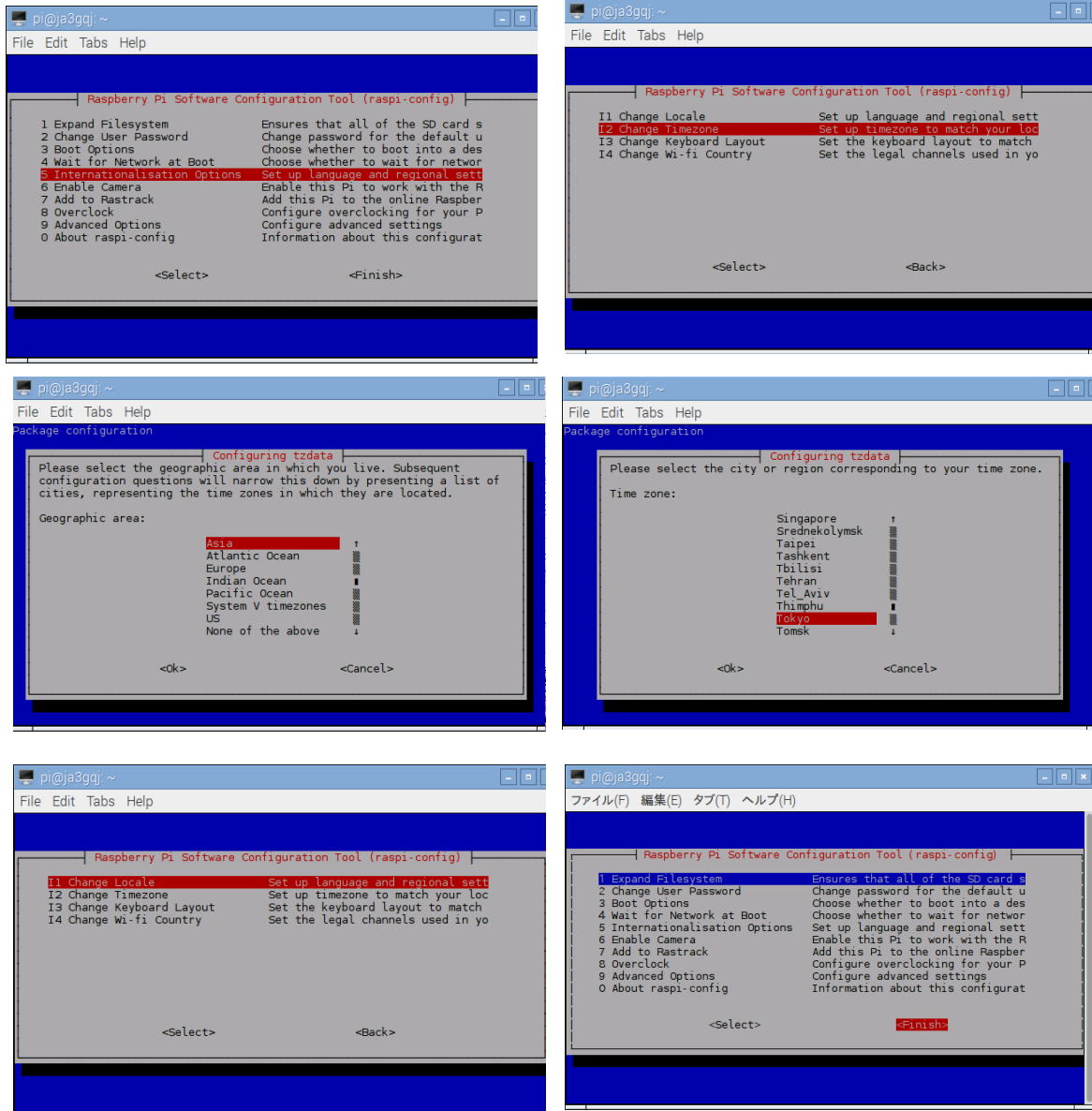
時間の変更

次の手順で日本時間を設定する

```
# raspi-config
```

このコマンドで最初に開いた画面の「Internationalisation Options」を選択して

次から下の画面の赤帯を順番に選択していく。右矢印で最後の「Finish」を選択する



日本語ソフトのインストール

次のコマンドを順番に実行していきますがかなり時間がかかるコマンドや作業中に「Do you want to continue?」と表示されるコマンドがありますこんなときは無条件で「y」を入力して次へ進みます。

```
# apt-get update
# apt-get upgrade
# apt-get install ibus-anthy
# apt-get install fonts-takao
```

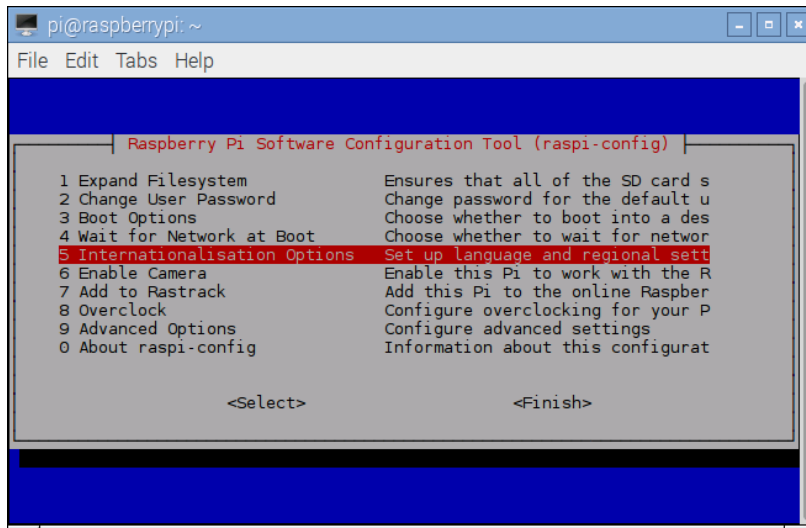
日本語仕様に変更

日本語ソフトのインストールが終わってからデフォルトの英語仕様から日本語仕様に変更します。

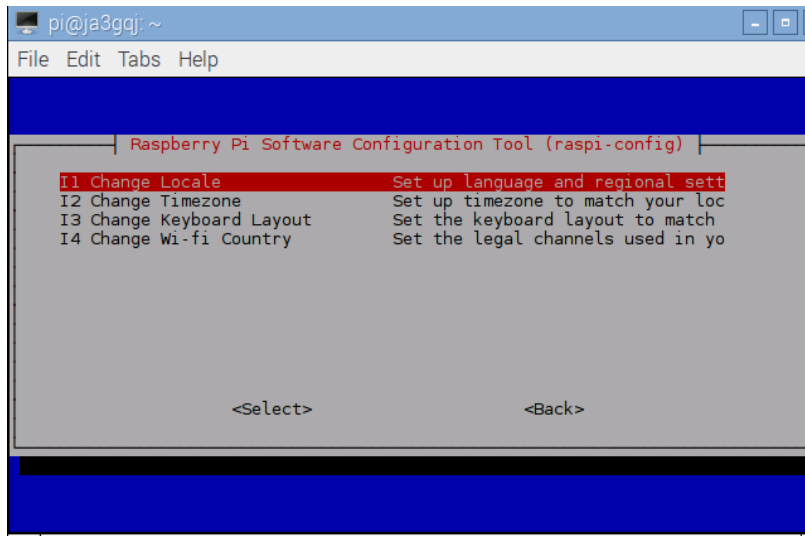
```
# raspi-config
```

このコマンドで最初に開いた画面の「Internationalisation Options」を選択して次から

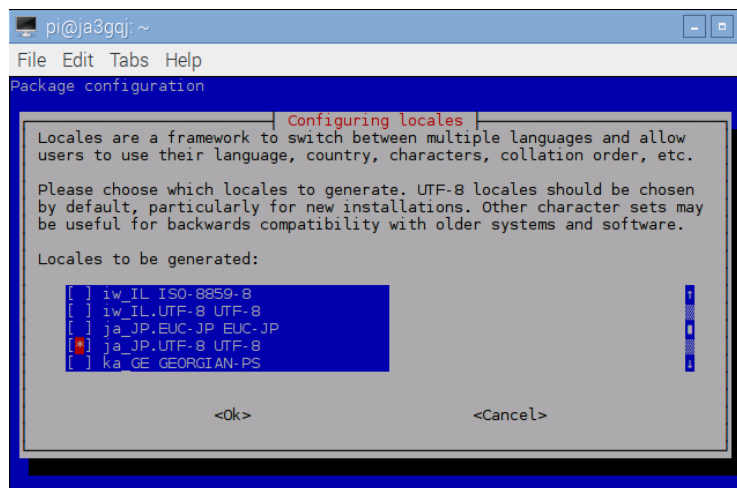
下の画面の赤帯を順番に選択していく。



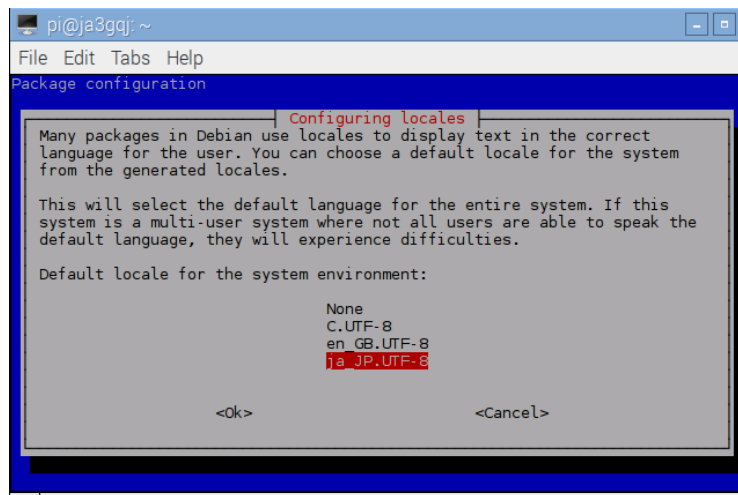
「Change Locale」を選択してエンターキーを押す。



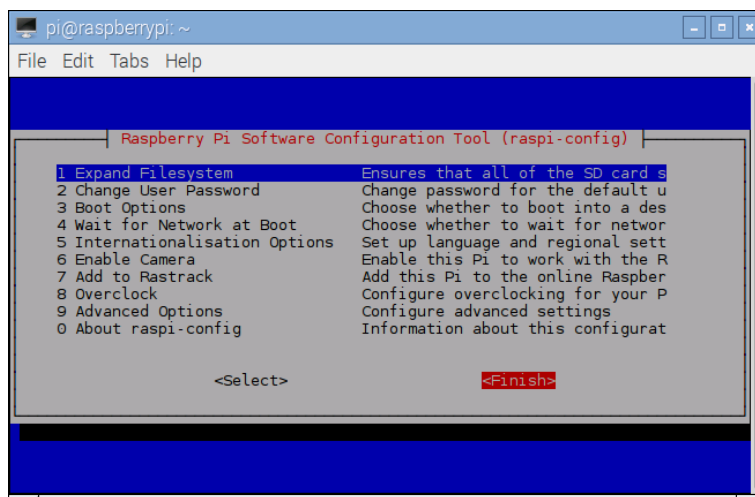
下向き矢印キーで「ja_JP.UTF-8 UTF-8」を選択し、スペースキーを押して米印を付けてエンターキーを押す。



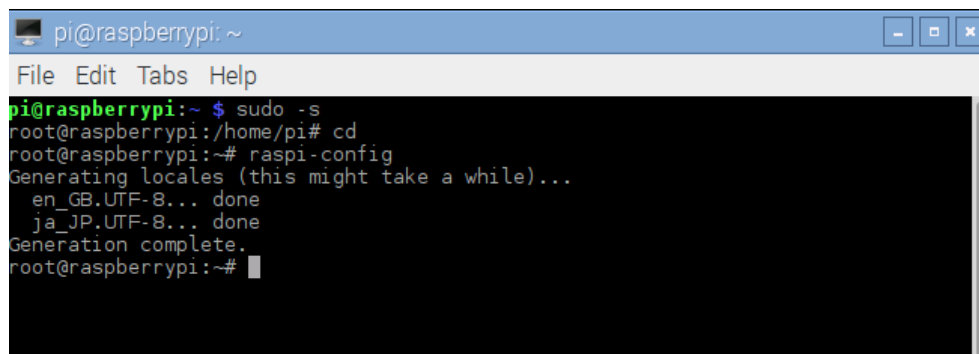
開いた画面で「ja_JP.UTF-8」を下向き矢印キーで選択してエンターキーを押す。



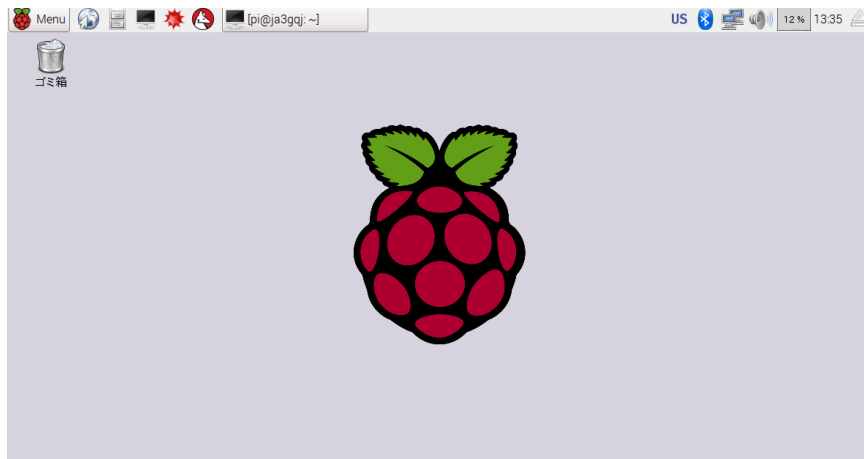
開いた画面で「Finish」を右矢印で選択してエンターキーを押す。



「Terminal」画面に戻って終了



和文でゴミ箱と表示されて日本語仕様になった事が分かる。

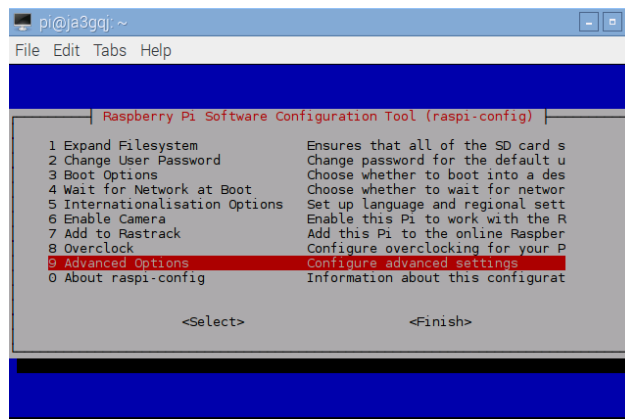


SSHを有効にする

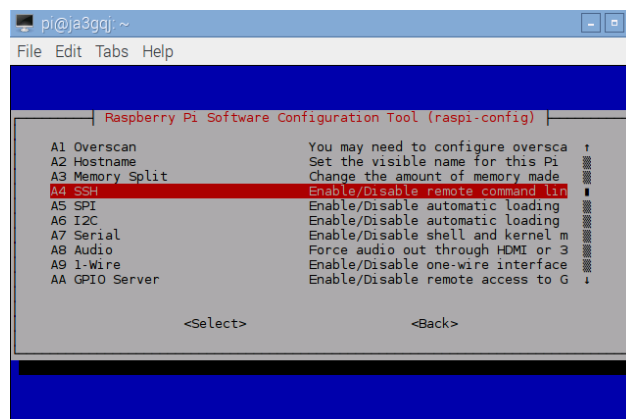
当局はこの SSH がよく理解できていませんが、有効にしておく Windows パソコンから Windows のリーモートディスク接続でラズベリーパイを操作できるのではないかと思います。

raspi-config

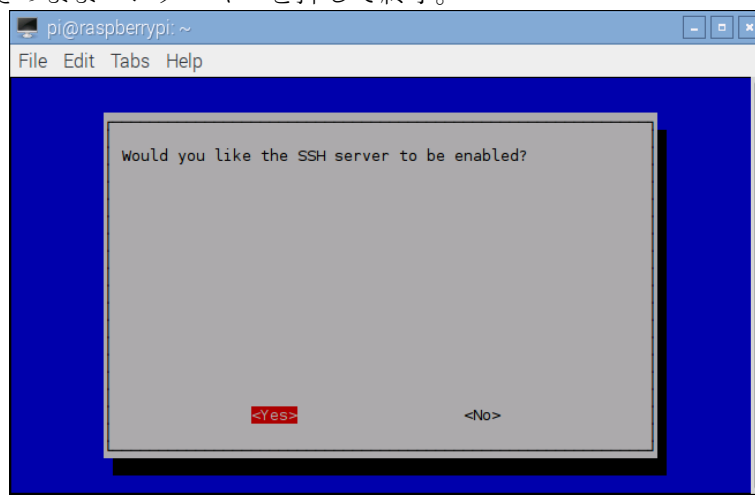
このコマンドで最初にかいた画面の「Advanced Options」を選択して次から下の画面の赤帯を順番に選択していく。



開いた画面で「SSH」を下向き矢印キーで選択してエンターキーを押す。



開いた画面でそのままエンターキーを押して終了。

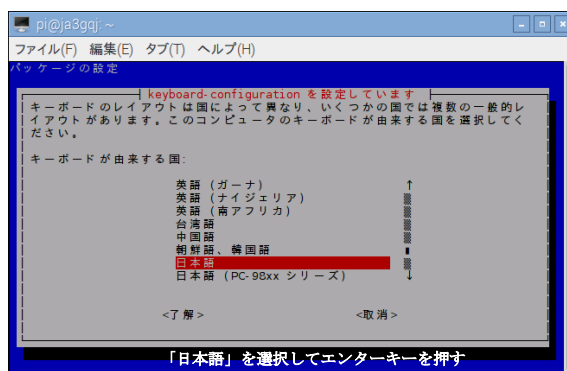
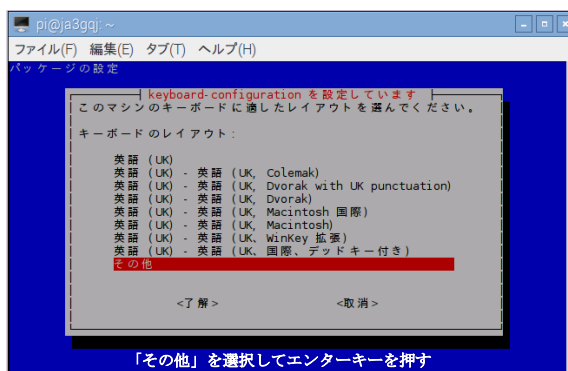
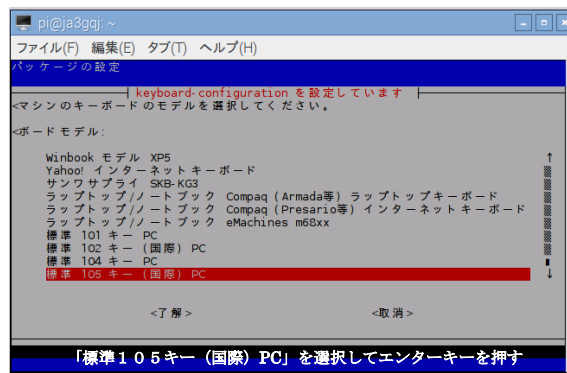
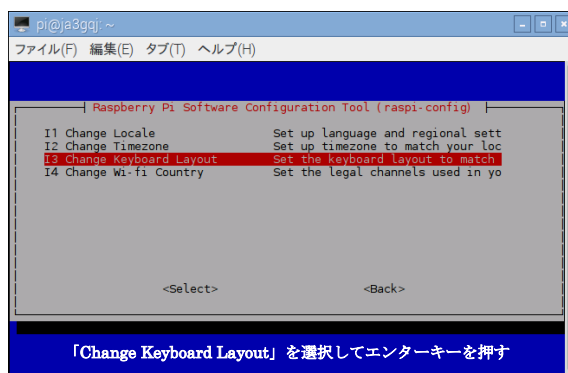


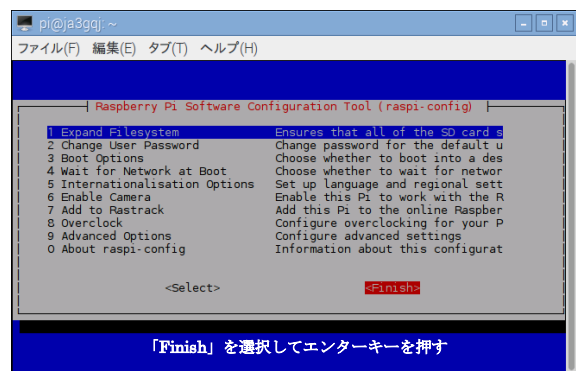
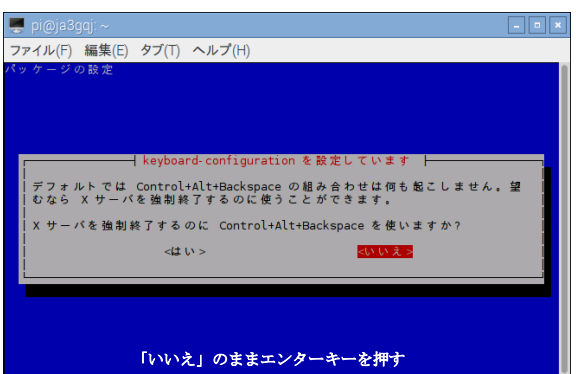
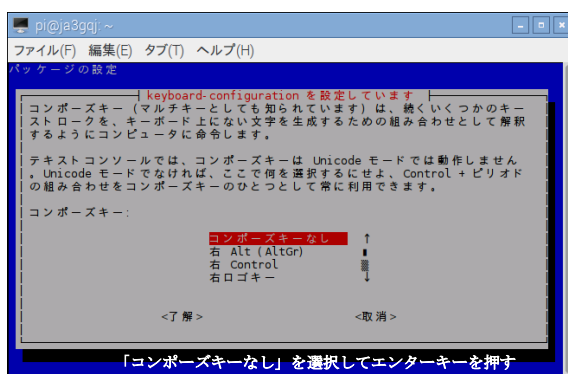
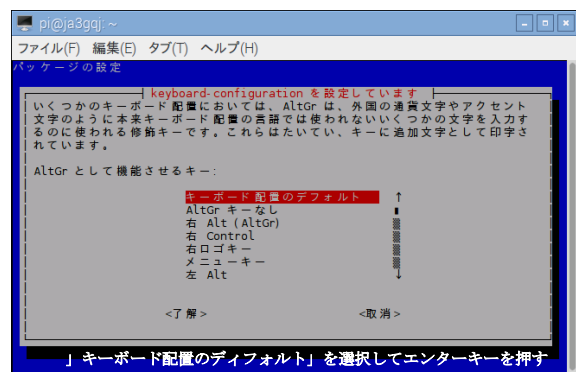
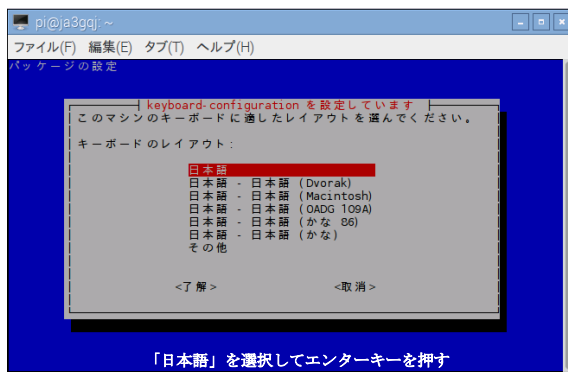
キーボード配列の変更

ディフォルトの英文キー配列は不便なので日本語キー配列に変更する。

```
# raspi-config
```

このコマンドで最初に開いた画面の「Internationalisation Options」(1 2 ページ 最上段) を選択して次から下の画面の赤帯を順番に選択していく。





Appendix

Win32 Disk Imager のダウンロード

https://osdn.jp/projects/sfnet_win32diskimager/

