

Raspberry Pi XLX リフレクター開設体験記

(2017/06/改訂版)

Shu JA3GQJ

はじめに

XLX リフレクターは JR3VH クラブが 2016 年 3 月に国内で最初に試験的に運用を始めたのがきっかけで 2016 年 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 3~4 年前に購入した古い Raspberry Pi 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 Raspberry Pi の OS image の焼き付けからリフレクターソフトのインストール迄を一から始めて使用してきました。今年 5 月にルーターと Raspberry Pi の AC 電源側に小容量の UPS を設置しましたが効果はまだ確認出来ていません。2017 年 6 月現在 Raspberry Pi のバージョンも更に新しくなっているようです。近頃は、ダッシュボードのカスタマイズにはまってしまう、index.php と peers.php に力を入れています。インターリンクの相手先が XLX 番号だけでは国が分かりにくいので原始的ですが peers.php に記述を追加して国旗を表示するようにしました。index.php にはデフォルトの五つのボタン以外に新しくボタンを追加し、その内容の表示用に liveccs.php を模倣した新しいファイルを pgs フォルダに追加しました。これらカスタマイズをバージョンアップに対応する便利な方法を模索中です。(巻末 Appendix④参照)

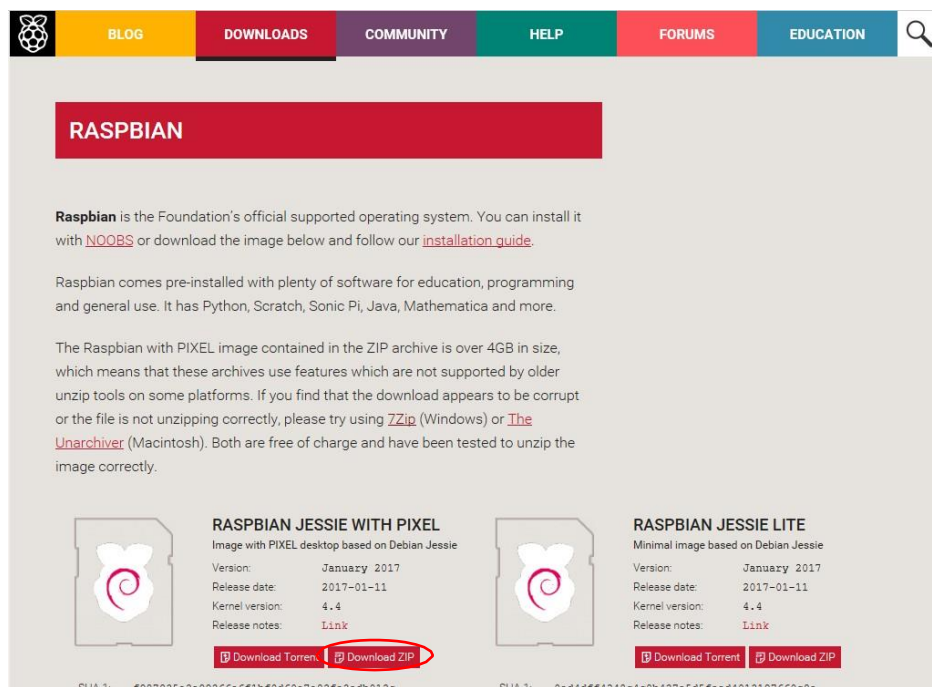
前提条件

- Raspberry Pi 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、GUI モードで使用する。(MSDOS のようなコマンドモードの文字入力方式の方が CPU への負荷が軽くなると言われていますが初心者は Windows の画面のような方が扱いやすいので GUI モードを使います。)
- Raspberry Pi 3 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)
- ポート開放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、Raspberry Pi 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。(手動設定方法は 13 ページ)
- 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://xlxreflector.jpn.ph> など) で行う。
- リフレクター番号 (XLX569 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。(Windows 10 パソコンでのダウンロードは巻末 Appendix②参照)
- SD カードによって稼働する Raspberry Pi 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。

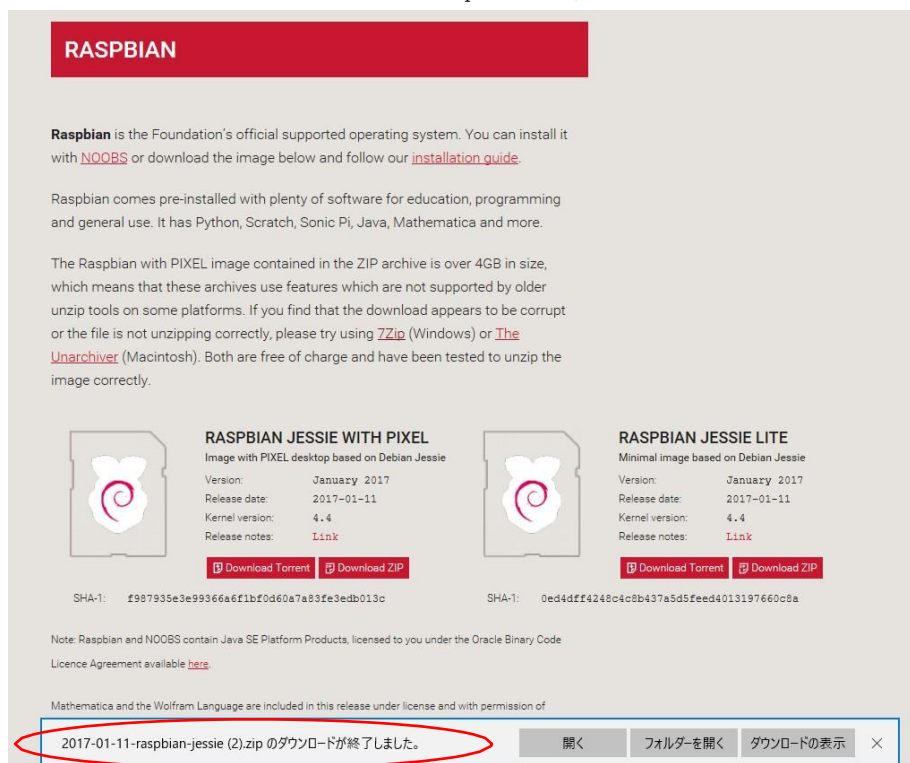
Raspberry Pi 用 OS の image を焼く

下記から RASPBIAN JESSIE WITH PIXEL の ZIP ファイル(2017年1月バージョン)をダウンロードしますがダウンロード時間が約30分必要であった。

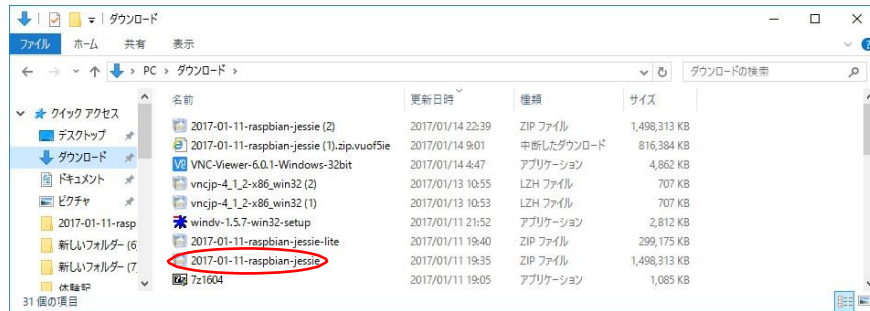
<https://www.raspberrypi.org/downloads/raspbian/>



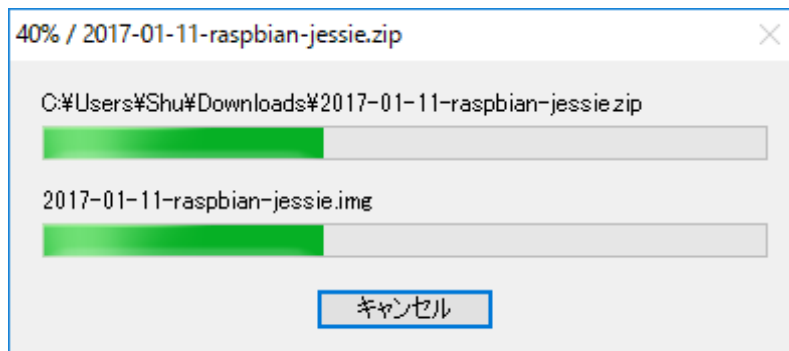
Download zip をクリック



約30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ィフォルトのダウンロードフォルダーなのでそこにある「2017-01-11-raspbian-jessie.zip」を解凍ソフト **Lhaplus** 等を使って「2017-01-11-raspbian-jessie.img」という image ファイルを作る。新しいバージョンはコメントにもあるように 4GB 以上のピクセル image の zip ファイルで 従来の Lhasa では解凍できず Lhaplus を使用した。



当局の場合「2017-01-11-raspbian-jessie.img」という image ファイルがデスクトップの「2017-01-11-raspbian-jessie」というフォルダー（右上）の中に生成されました。

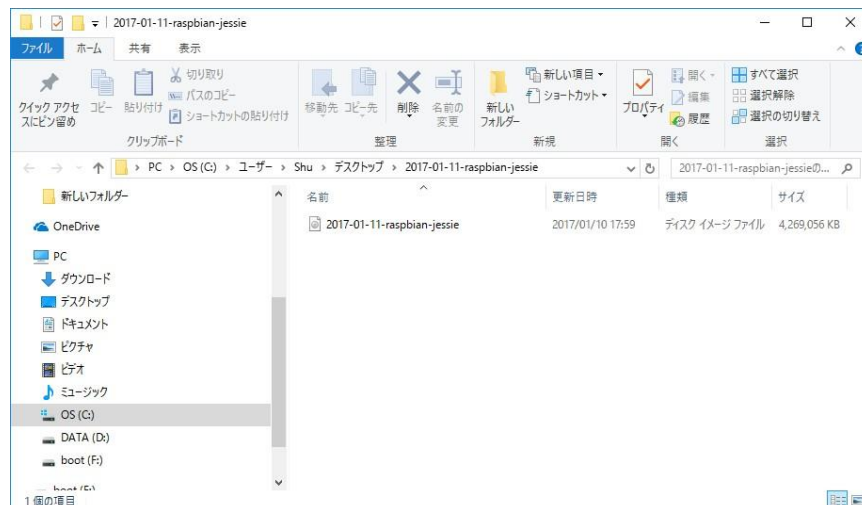
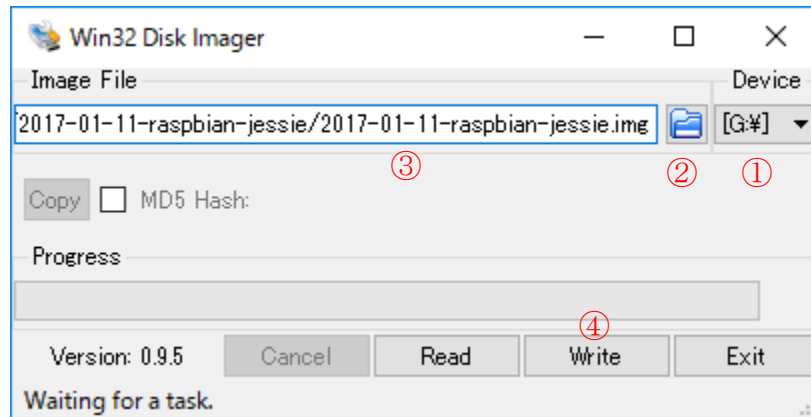
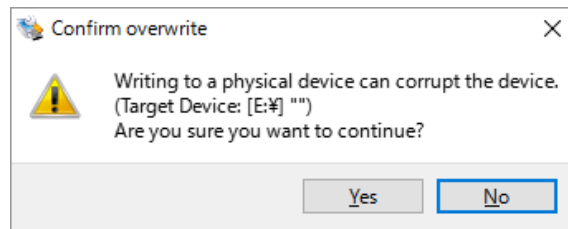


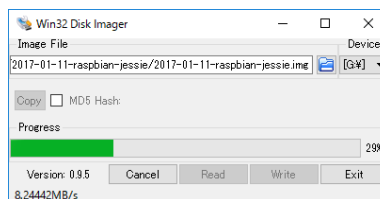
image ファイルが出来たら Win32 Disk Imager で SD カードに焼き付けます。



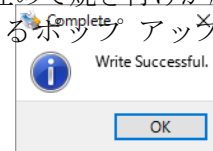
Win32 Disk Imager 立ち上げて SD カードを Windows パソコンに挿入すると自動でデバイス（この例では G¥①です）が認識されます。その左隣のフォルダーのアイコン（②）をクリックして先ほど生成した「2017-01-11-raspbian-jessie」というフォルダーの中にある「2017-01-11-raspbian-jessie.img」という image ファイル（③）を選択したら「Write」（④）をクリックして焼き付けを開始します。



先に何かが書き込まれた 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(当局の場合約 9 分)と緑色のプログレスバーが空白を埋めて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 のダウンロードは巻末 APPENDIX②参照

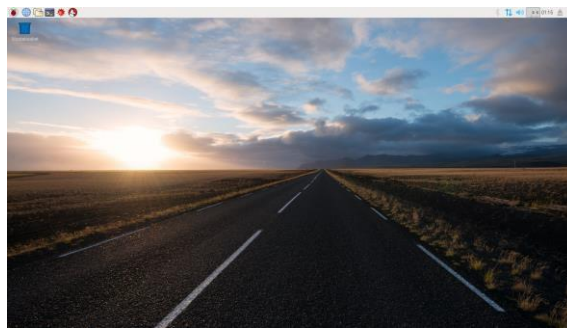
Raspberry Pi の起動

OS イメージの焼き付けが終わった SD カードを Raspberry Pi に装着して起動すると従来のバージョンと違った中身は英文仕様の GUI 画面がポップアップします。

起動準備完了

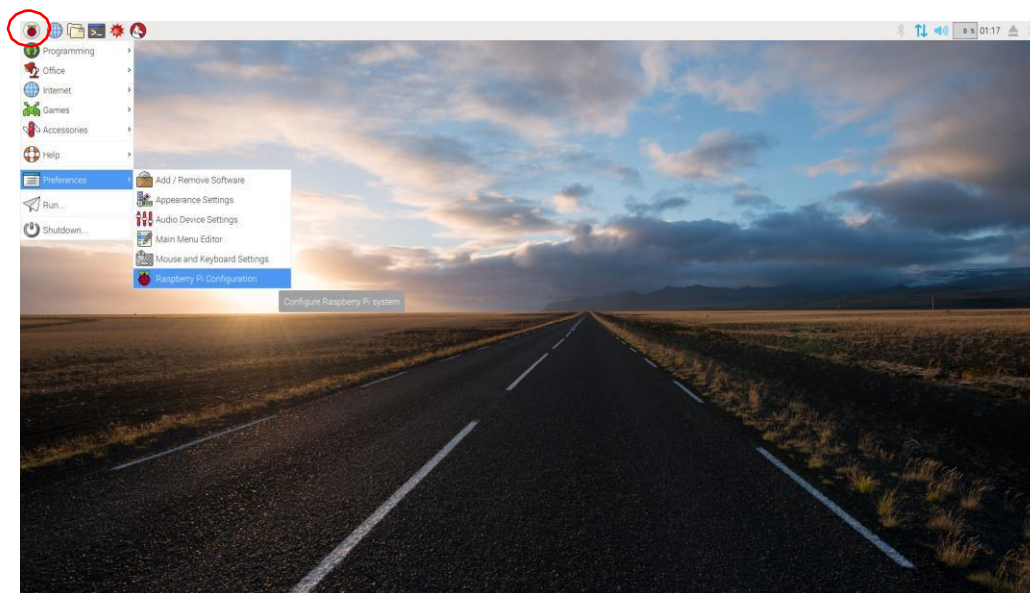


起動後のモニター画面



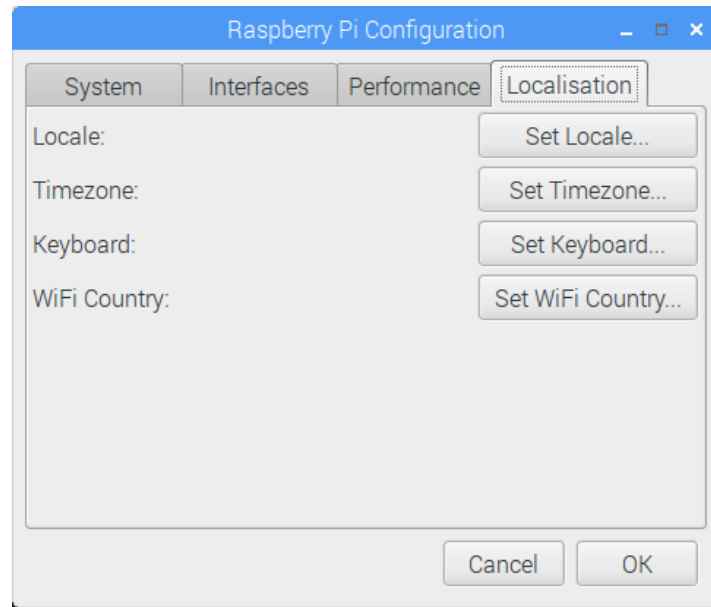
Raspberry Pi を日本語仕様にする

画面左上のラズベリーのアイコンをクリックしてメニューを開いて Preferences ➡ Raspberry Pi Configuration と進んで Raspberry Pi Configuration を開く。

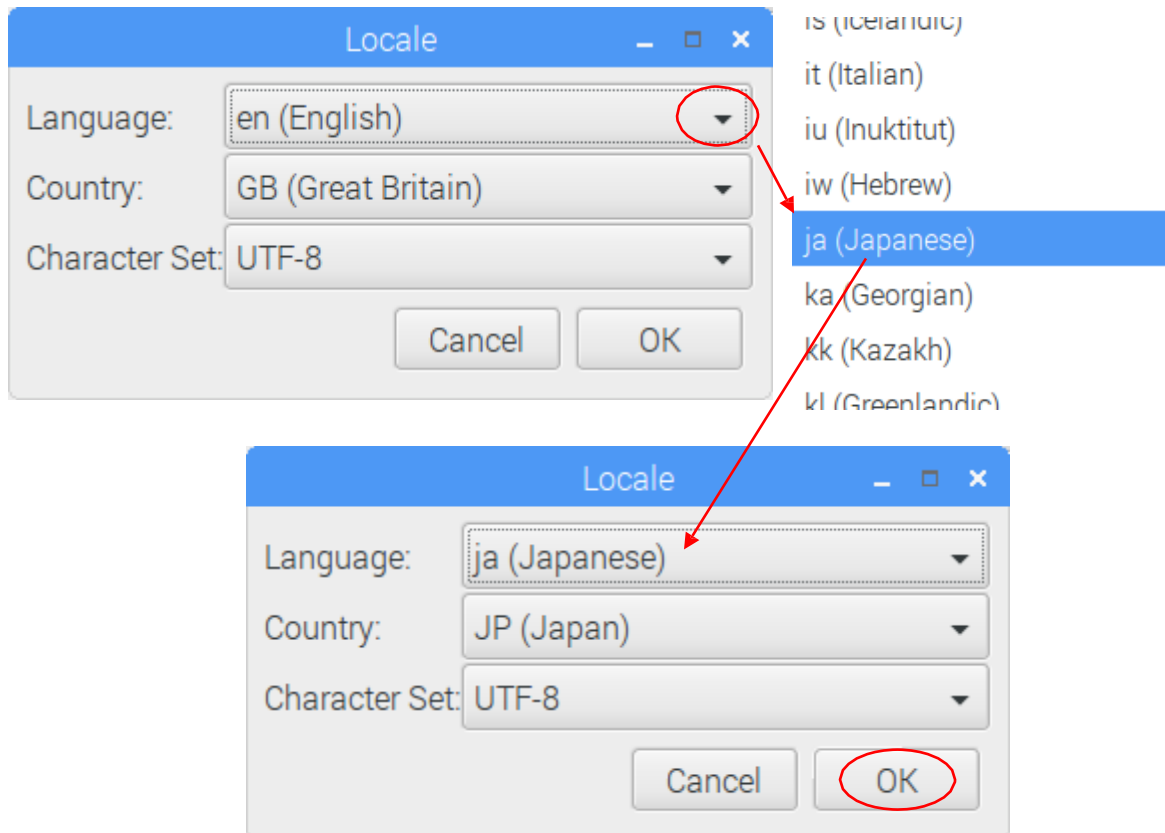


Raspberry Pi Configuration を開くと「System」、「Interfaces」、「Performance」、「Localisation」（ローカライゼーション）の合計4種類のタグがありますので先ず「Localisation」（ローカライゼーション）タグを選択する。

注：日本語仕様にするだけであれば Set Local でロケイルを日本にするだけでも良いが、ついなので「Localisation」（ローカライゼーション）タグの項目すべての設定をしました。



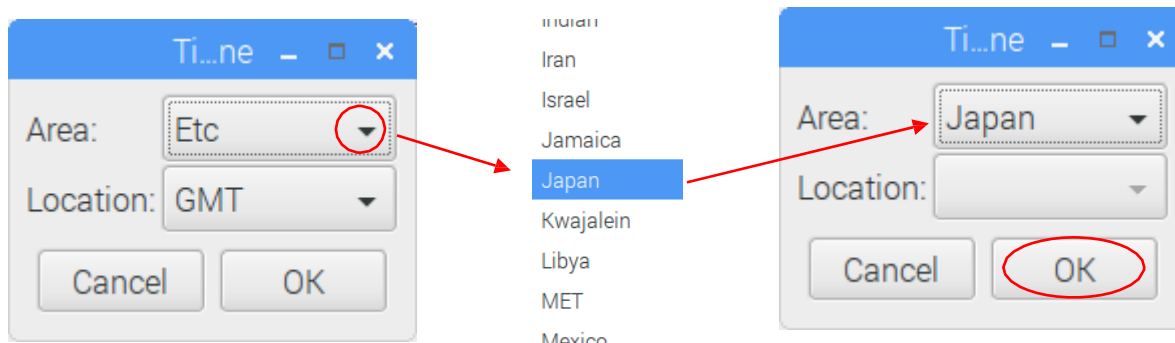
「Localisation」タグで「Set Locale…」をクリックするとデフォルトの画面がポップアップする。プルダウンボタン（右の▼印）を押して表示されるプルダウンメニューから ja (Japanese) を選択する。（Country は自動的に JP (Japan) が設定される）



タイムゾーンを変更する

次に「Localisation」タグで「Set Timezone...」をクリックして時間を UTC(GMT) から日本時間に変更する。

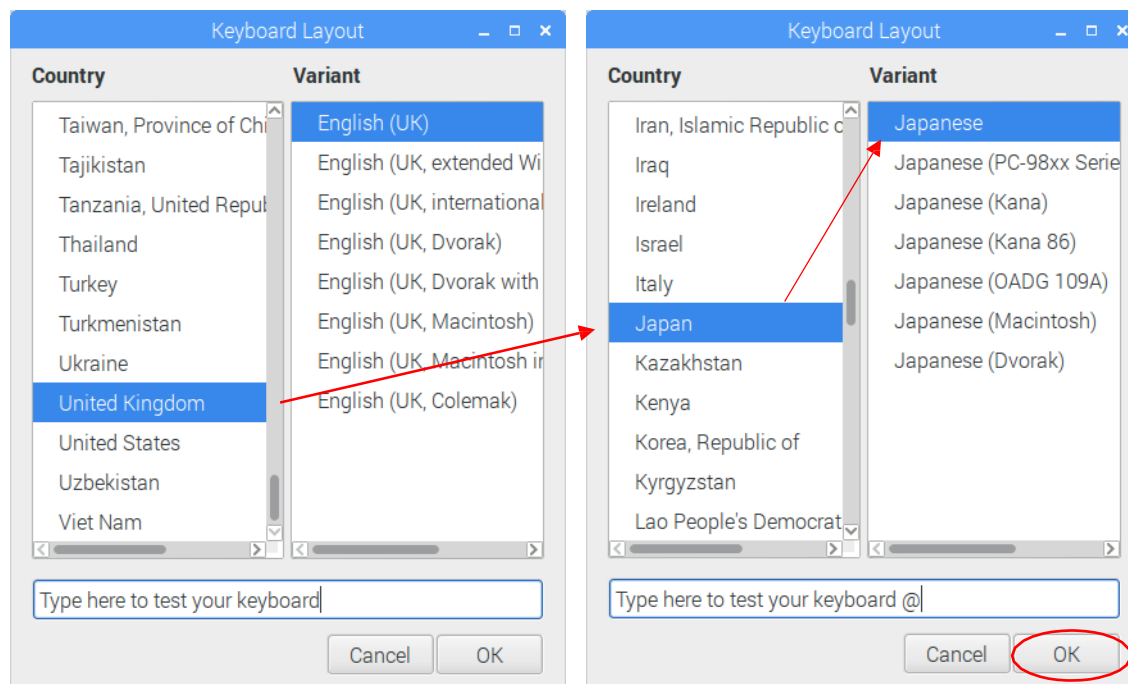
Area のプルダウンボタン（右の▼印）を押して表示されるプルダウンメニューから Japan を選択する。（Location の設定は不要）



キーボード配列を日本仕様に変更する

次に「Localisation」タグで「Set Keyboard...」をクリックしてキーボードを日ごろ使い慣れた配列にする。

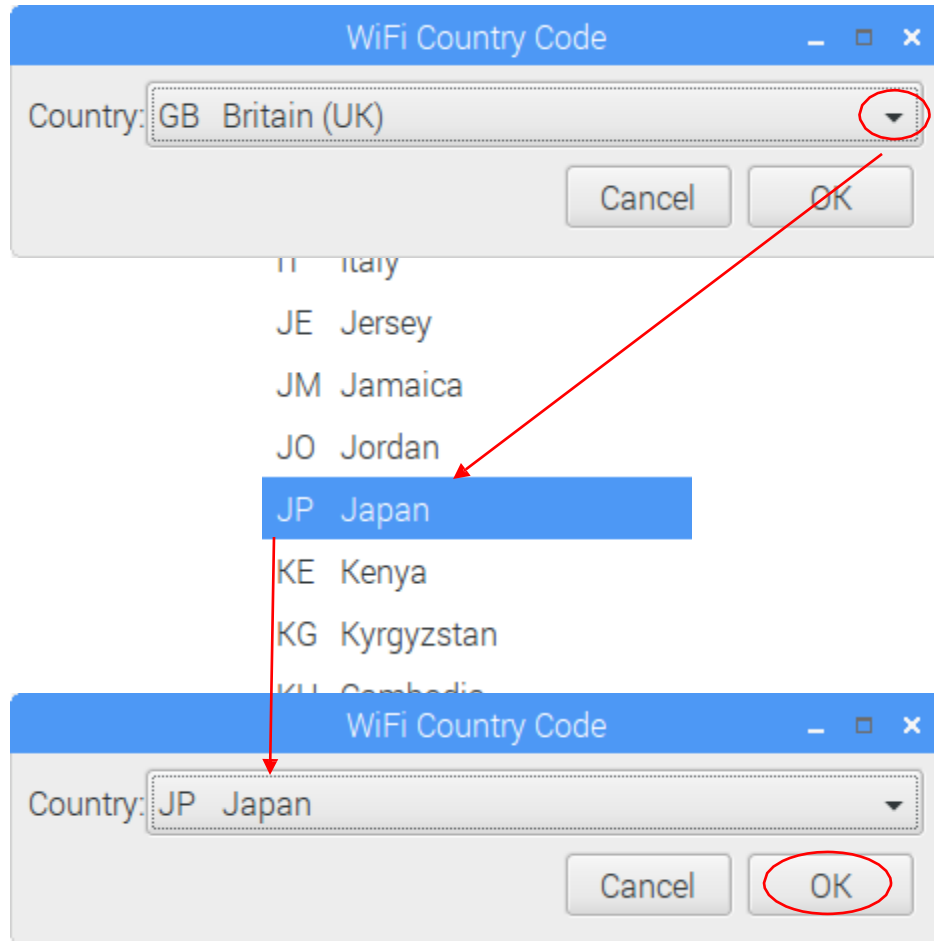
Country のデフォルトは United Kingdom になっているので Country の右のスクロールバーで Japan を選択した後さらに Variant で Japanese を選択して「Type here to test your keyboard」欄でキーボードの@マークキーを押して正しく@マークが表示されることを確認したら「OK」ボタンで設定終了。



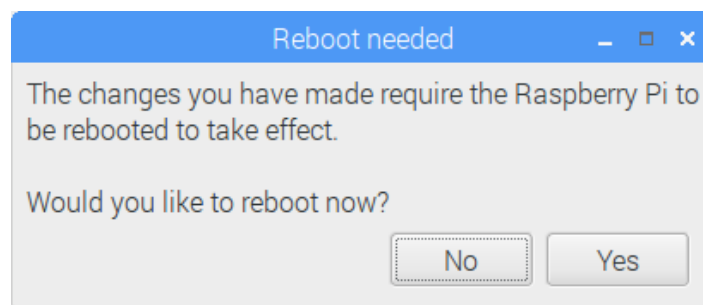
WiFi を日本仕様に変更する

「Localisation」 (ローカライゼーション) タグで「Set WiFi Country...」をクリックして Country のプルダウンボタン

(右の▼印) を押して表示されるプルダウンメニューから JP JAPAN を選択する。

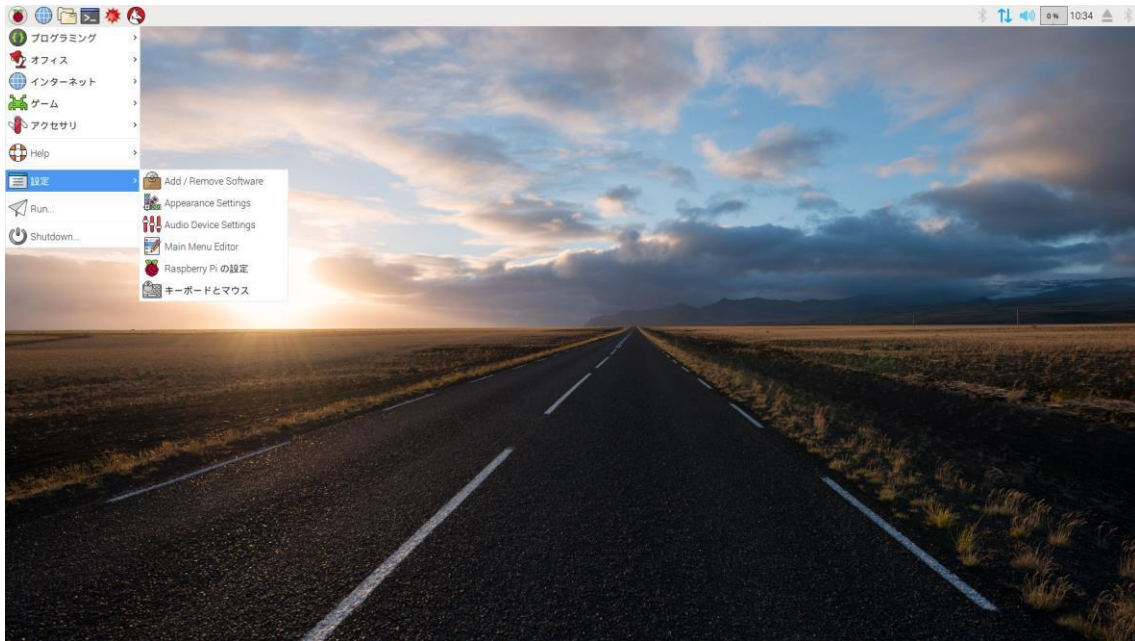


WiFi Country Code 画面の「OK」ボタンをクリックしたら「Localisation」 (ローカライゼーション) タグの全ての項目の設定が終了しますので「Localisation」タグの「OK」ボタンをクリックして和文仕様への設定が完了します。このとき次のようなコメント (設定を有効にするために再起動が必要) がポップアップするので「Yes」をクリックして Raspberry Pi を再起動します。

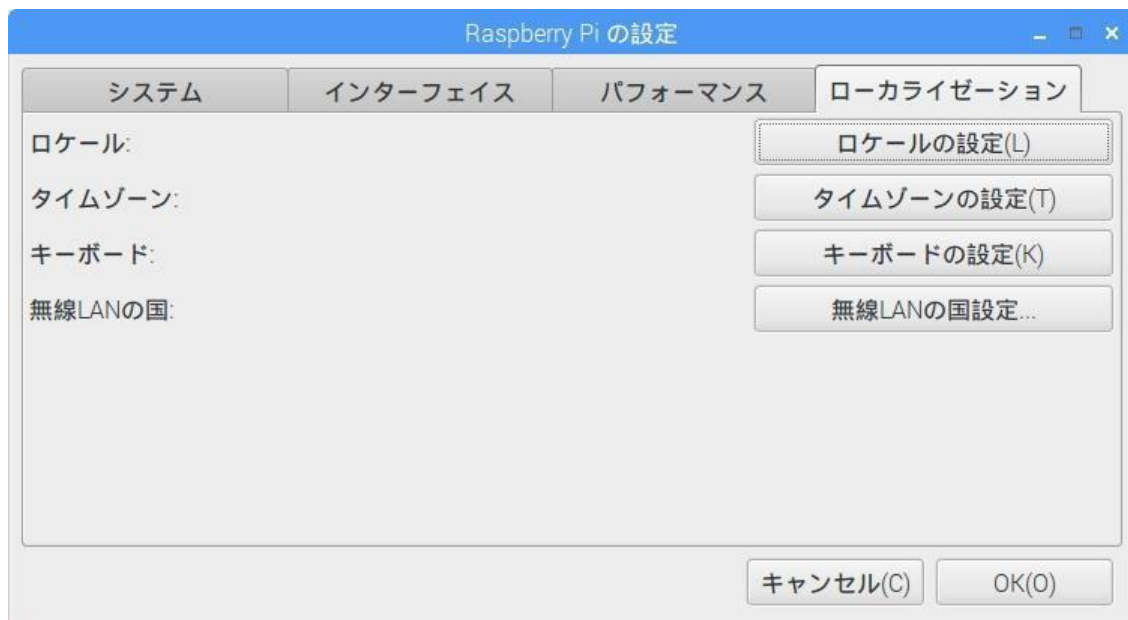


日本語仕様の確認

再起動後画面左上の Raspberry Pi アイコンをクリックしてメニューを開くと Preferences が「設定」、Raspberry Pi Configuration が「Raspberry Pi の設定」に変わっているので、

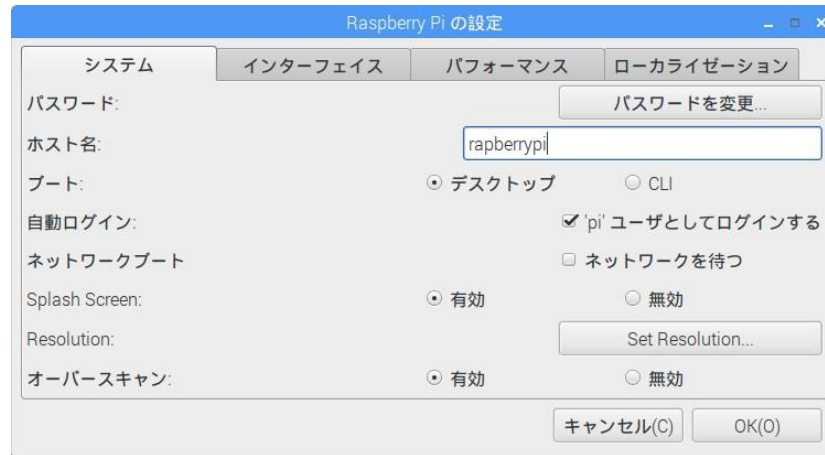


そこで「設定」⇒「Raspberry Pi の設定」と進んで「Raspberry Pi の設定」画面を開くとタグの表示が全てカタカナで表示されています。

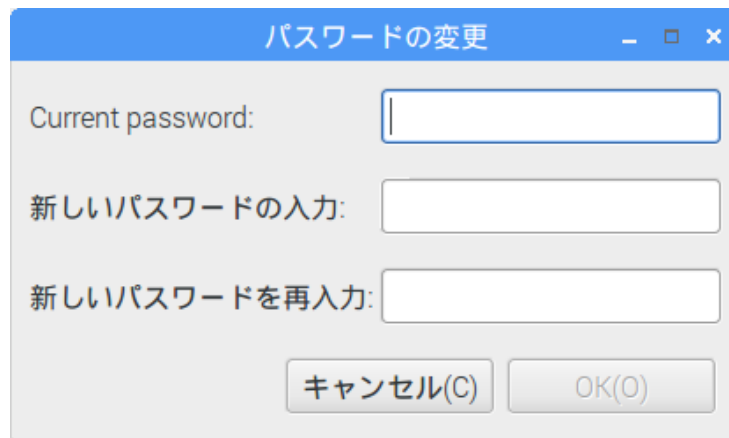


ホスト名、パスワードの変更

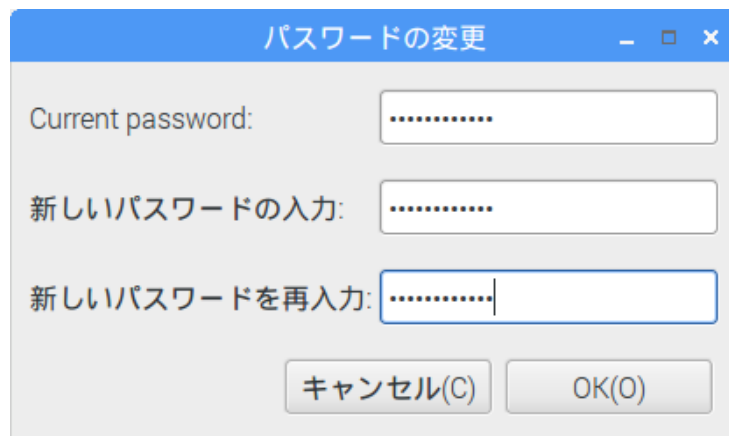
デフォルトのホスト名は「Raspberrypi」です。これを例えば「ja3gqj」等に変更します。



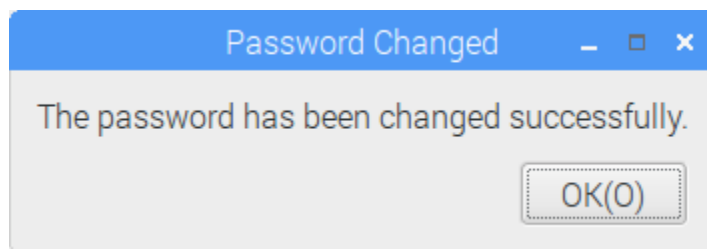
デフォルトのパスワードは「raspberry」です。これを「パスワード変更」ボタンをクリックしてポップアップした画面（下図）で自分用に変更します。



このとき、新しいバージョンから従来のパスワード (Current Password) が必要になりました。



空白欄をすべて埋めて「OK」ボタンをクリックするとパスワードが変更された旨のコメント画面がポップアップしますので「OK」ボタンをクリックして終了します。



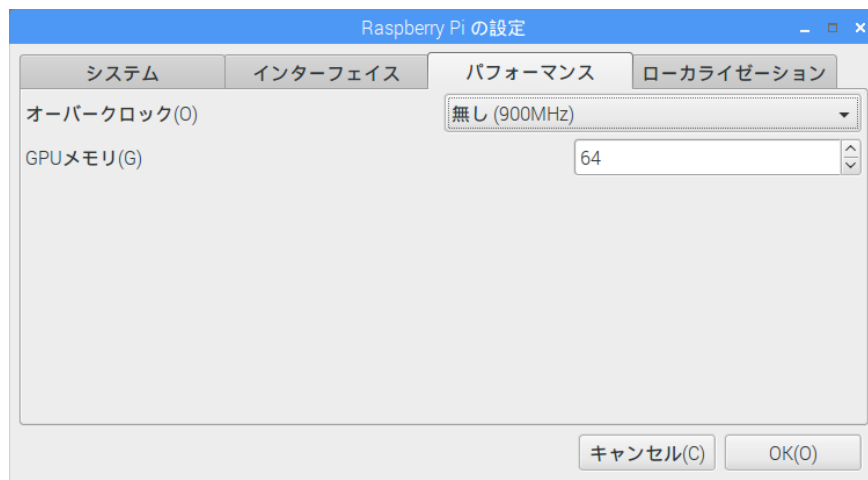
SSH と VNC を有効にする

今回は使いませんがこれらを有効にしておくと Raspberry Pi をウィンドウズパソコンなどから遠隔操作が出来るようなので便利かと思います。



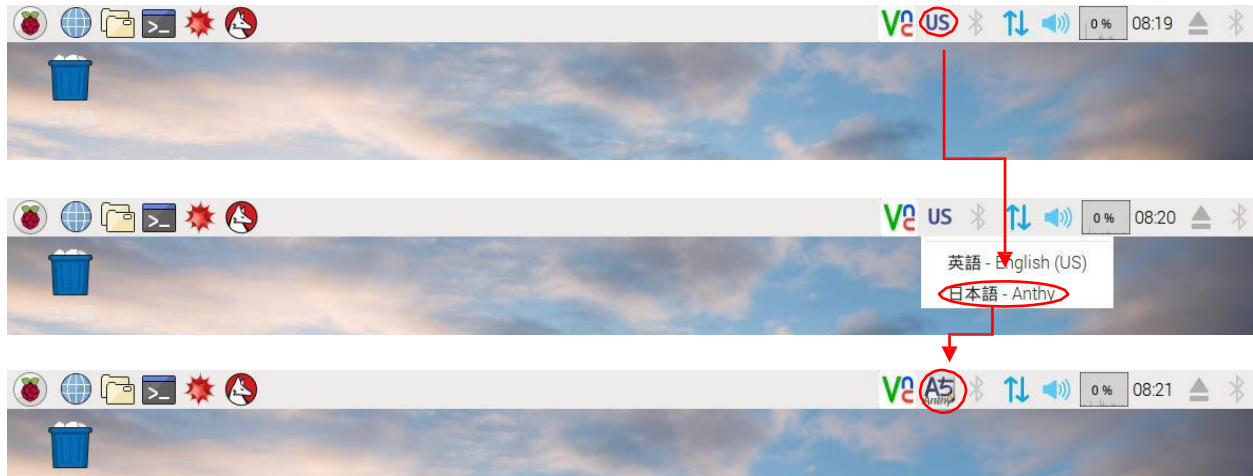
パフォーマンスは変更しない

これを操作することで動作速度が早くなる場合があるそうですが発熱や CPU の劣化を早めるなどの問題が有ると言う事なので、ここはデフォルトのままにしておきます。

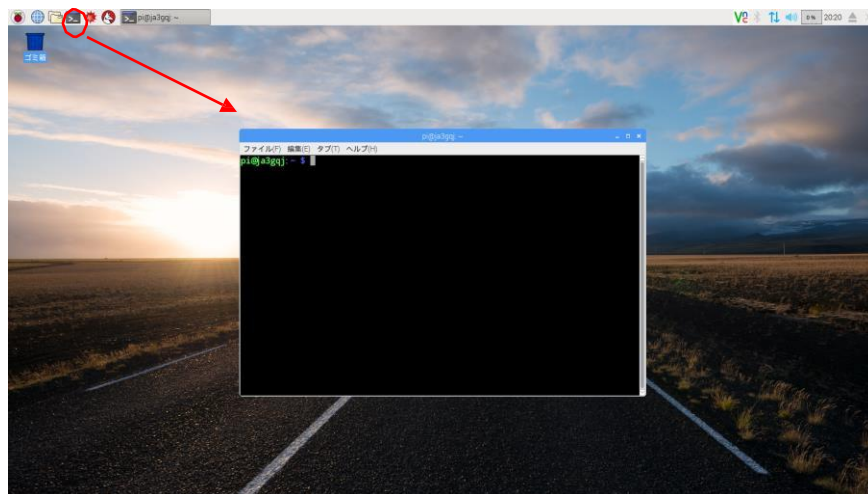


日本語ソフトのインストール

9 ページで日本語仕様に変更した確認をしました。ところが文章を編集するときの日本語の入力方法が分からないので別途次の要領で日本語ソフトをインストールしました。インストール後は再起動すると画面右上にアイコンが表示されます。



- ・ 起動後ターミナルエミュレーター（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（#印は含まない）

```
pi@ja3ggj: ~  
File Edit Tabs Help  
pi@ja3ggj:~ $ sudo -s  
root@ja3ggj:~/home/pi# cd  
root@ja3ggj:~#
```

次のコマンド（＃は含まない）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?」（続けるかどうか）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
```

エディターpicoについて

これ以降はエディターが必要になりますので当局はRaspberry Pi 付属の pico を使用します。編集や新しく作成するファイル名の前に pico /xx/xx.yy などと端末エミュレータ TXTerminal に 打ち込みます。カーソル移動は上下左右の矢印を使う必要がありますがその他は Windows パソコン と同じようなキーボード操作で OK です。ただ編集、新規作成後の保存は先ず、コントロールキーと「x」キーを同時に押した後「y」を入力してエンターキーで保存終了します。

Raspberry Pi に固定 IP アドレスを設定する

最近わかってきましたが XLX リフレクターソフトとはダッシュボードと称して一種のホームページを開設するようなものでホームページ公開と同じように IP アドレスは固定しなければならないので XLX リフレクターソフトをインストール前に IP アドレスを固定することにします。

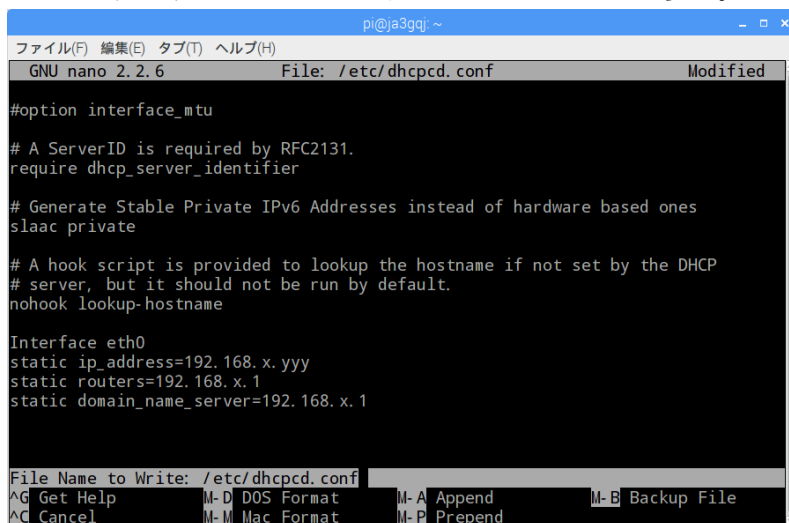
```
# pico /etc/dhcpd.conf
```

開いた Terminal 画面の最終行の次に有線か無線、どちらかを追記しますが入門用には有線のほうが楽です。

有線 LAN 用（無線 LAN 用は Interface eth0 を Interface wlan0 とする）

```
Interface eth0
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@ja3gqj: ~
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

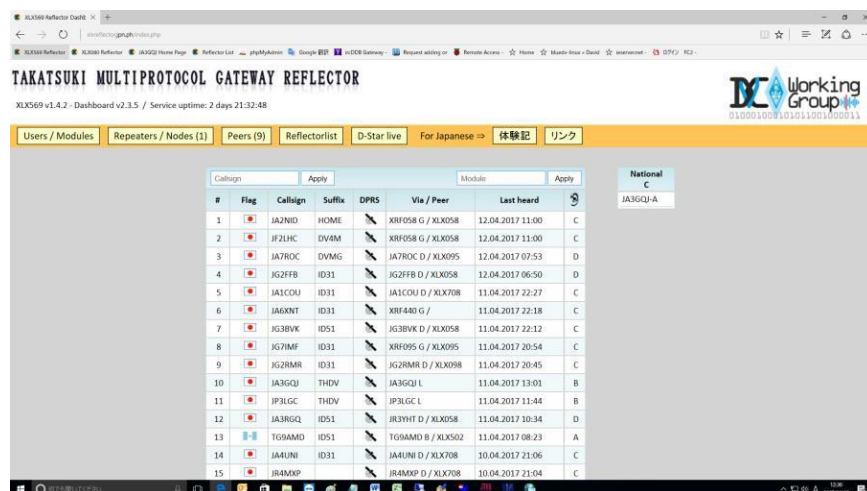
# A hook script is provided to lookup the hostname if not set by the DHCP
# server, but it should not be run by default.
nohook lookup-hostname

Interface eth0
static ip_address=192.168.x.yyy
static routers=192.168.x.1
static domain_name_server=192.168.x.1

File Name to Write: /etc/dhcpd.conf
^G Get Help      M-D DOS Format  M-A Append      M-B Backup File
^C Cancel        M-M Mac Format  M-P Prepend
```

XLX マルチプロトコルリフレクターゲートウェイソフトの概要

Dplus 系の REFxxx リフレクターは開設のための登録手続をすれば自分でリフレクター用のソフトを用意する必要はありませんがこの XLX リフレクターは強い登録手続の必要がなく基本的に自分でリフレクター用のソフトを立ち上げれば OK です。正常に立ち上がるとウィンドウズパソコン等から普通の WEB ページを見るようにその URL 例えば <http://xlxreflector.jpn.ph/index.php> 等と打ち込むと次のような画面が現れこれをダッシュボードと呼んでいます。一番の特徴は php の記述で画面をカスタマイズできることです。

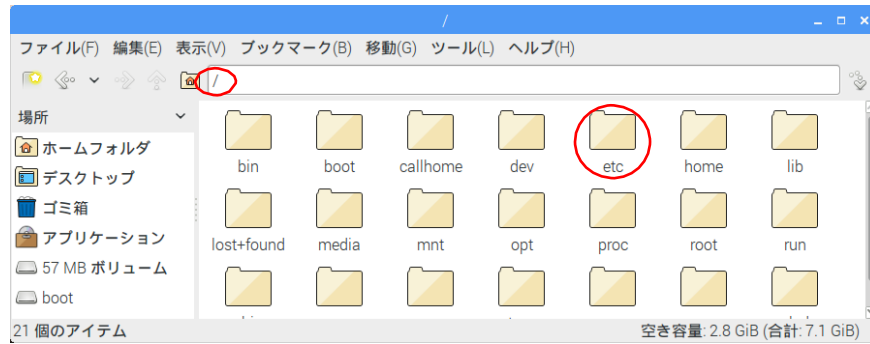


このダッシュボーには WEB ページ（ホームページ）を開設するときに使われる・・・/index.html に代えて・・・/index.php が使われていますが、WEB サーバー用 apach2 や PHP と言った WEB ページのプログラムを記述するソフトをインストールするようになっておりリフレクターソフトをインストールする過程でディフォルトの index.html ファイルが生成されています。

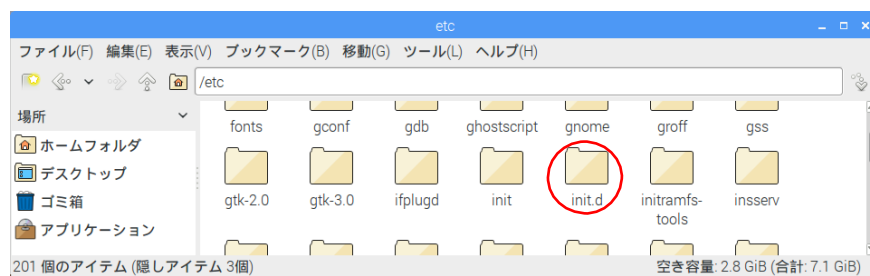
Raspberry Pi の OS とソフトのディレクトリー（Windows パソコンでフォルダーと言われている）の構成は次のようになっています。ファイルマネージャーで開くディフォルトのフォルダーは pi フォルダーで中身は下図の通りです。



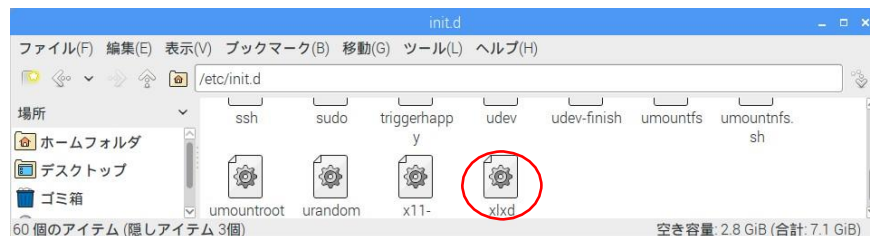
上向き矢印（上図）でルートディレクトリー（記号の「/」だけ下図）に行くと次の図のように bin, boot, callhome, dev, etc, home, lib, lost+found, media, mnt, opt, proc, root, run, sbinsrv, sys, tmp, usr, var, xlxld とする合計 21 種類のフォルダーが出来ています。此の内 callhome と xlxld はプログラムインストール中に生成されます。



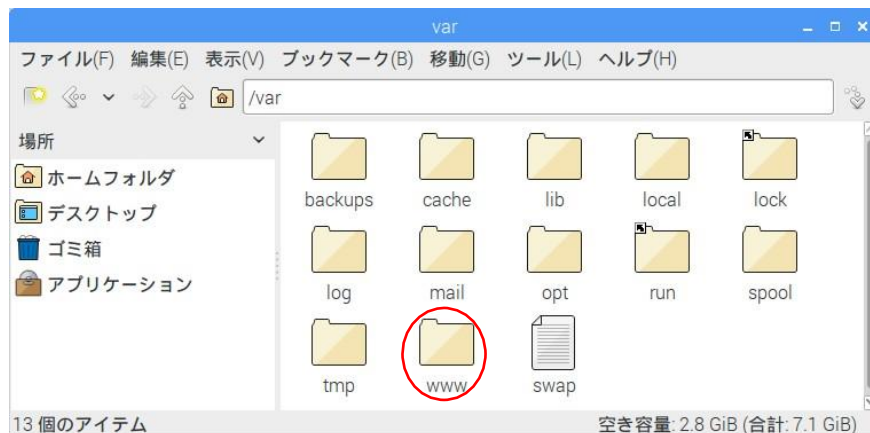
また etc フォルダーには合計 111 種類のフォルダーと合計 90 種類のファイルがあります。
(ファイルの数は変動するものと思われます。)



init.d フォルダーには合計 60 種類のファイルがあり、この中で重要なのが x1xd ファイルです。



更に var フォルダーには backups, cach, lib, local, lock, log, mail, opt, run, spool, tmp, **www** という
合計 12 種類のフォルダーがあり



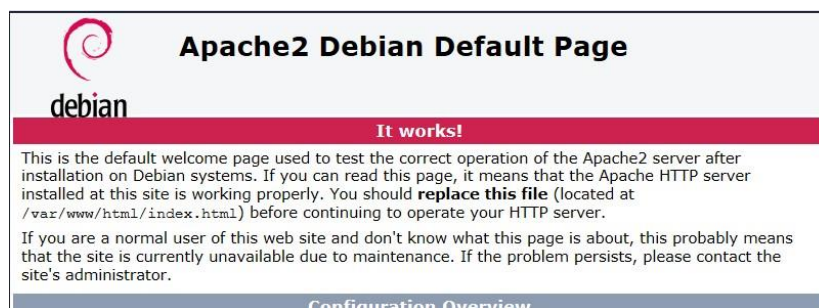
その中の **www** フォルダーにお馴染みの **html** フォルダ（下図）があります。



この **html** フォルダに **XLX** リフレクターに必要なすべてのフォルダとファイル（下図）が生成されています。



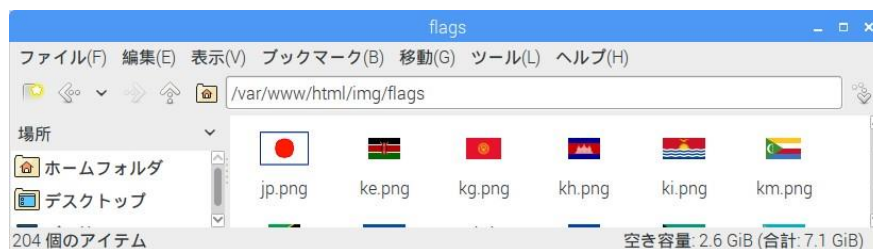
web ダッシュボードで表示するのは **index.php** です。ちなみに <http://xlxreflector.jpn.ph> だけを打ち込んだ場合はデフォルトの **index.html**（下図の Apache2 Debian Default Page）が表示されますので URL の打ち込みは <http://xlxreflector.jpn.ph/index.php> としなければなりません。



img フォルダには見慣れたアイコンの画像ファイルと各国の旗のフォルダ **flags** があります。此の中へ自分の好みの画像を入れて同じファイル名にするとダッシュボードに表示されるアイコンや画像が変更されます。



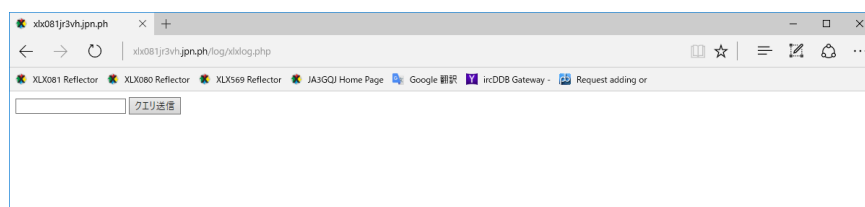
flags フォルダの日本国旗は当局で加工したものと交換しています。



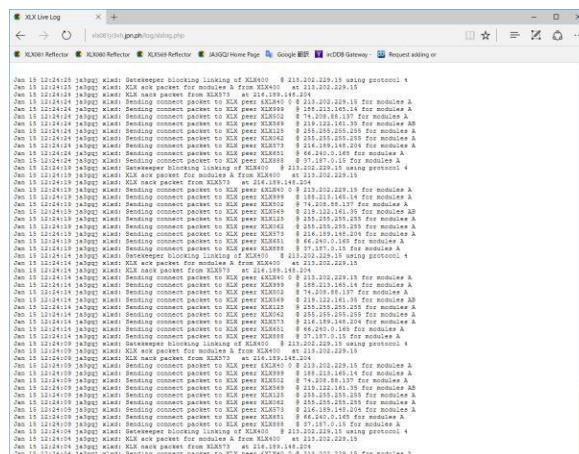
log フォルダにある xlxlog.php は web ページで <http://xlxreflector.jp.ph/log/xlxlog.php> と打ち込むと・・・



・・・次のようなページがポップアップします。



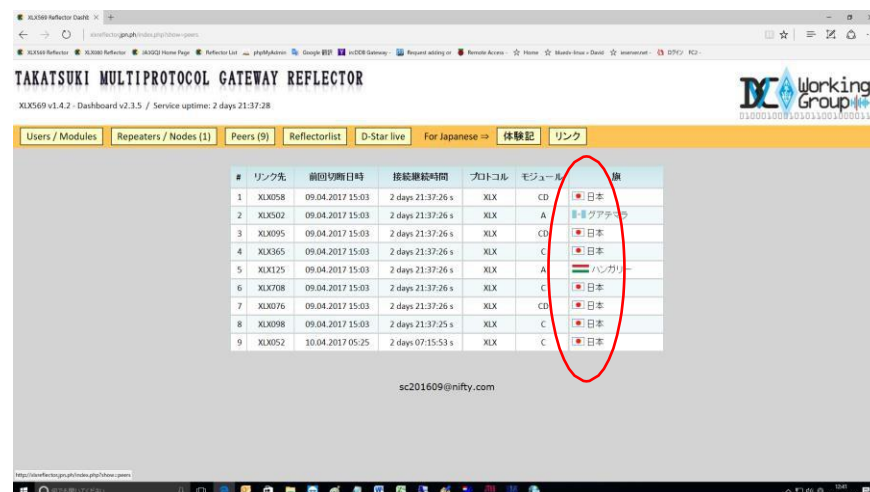
このページで「クエリー送信」にパスワード (-D-Star+) を入力するとリアルタイムのログが表示されいているようですが利用（解読）方法が分かりませんけど相互接続相手との状況が表示されているように見えます。



pgs フォルダには各種 php ファイルがあり config.inc.php はセットアップでその内容を変更する必要があるファイルです。



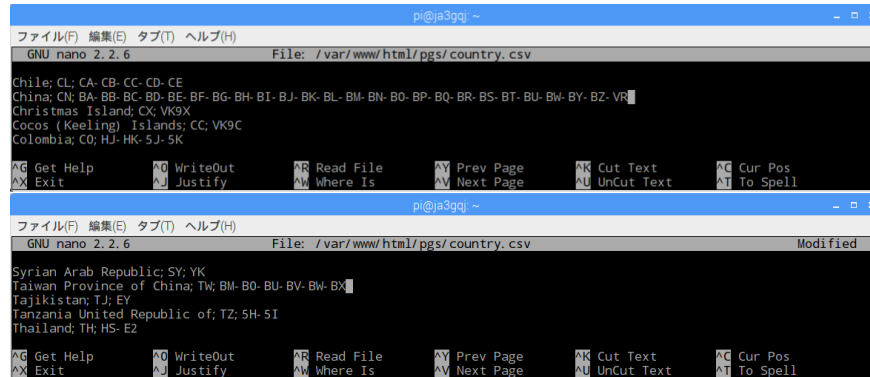
当局は pgs フォルダの peers.php ファイルを加工して相互接続相手の旗と国名を表示するようにしました。



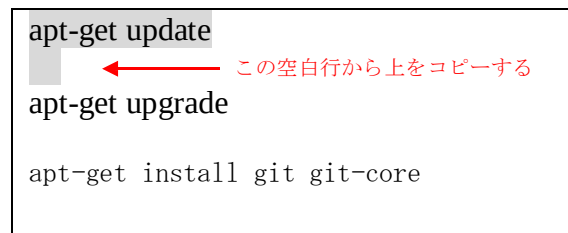
また下図のように旗が表示されないのは pgs フォルダの country.csv の国名リストにその国の新しいプリフィックスが記入されていないためなのでこのファイルを開いてプリフィックスを記入します。バージョンが 1.4.0 になってから台湾、香港、中国のプリフィックスリストと旗の関係に変化があり、香港の旗がなくなり、従来の台湾のプリフィックスが中国のプリフィックスにリストされています。（当局はバージョン 1.3.9 以前のものに編集しました。）

#	Flag	Settings	Info	IP	My Peer	Last Heard
1	日本	XLX058	CD	207.170.11.100	207.170.11.100	09.04.2017 15:03
2	台湾	XLX502	A	207.170.11.100	207.170.11.100	09.04.2017 15:03
3	日本	XLX095	CD	207.170.11.100	207.170.11.100	09.04.2017 15:03
4	日本	XLX365	C	207.170.11.100	207.170.11.100	09.04.2017 15:03
5	ハンガリー	XLX125	A	207.170.11.100	207.170.11.100	09.04.2017 15:03
6	日本	XLX708	C	207.170.11.100	207.170.11.100	09.04.2017 15:03
7	日本	XLX076	CD	207.170.11.100	207.170.11.100	09.04.2017 15:03
8	日本	XLX098	C	207.170.11.100	207.170.11.100	09.04.2017 15:03
9	日本	XLX052	C	207.170.11.100	207.170.11.100	09.04.2017 15:03

China のプリフィックス「VR」（香港）は当局が追加しましたが BV, BX など一部のプリフィックスは China と Taiwan の両方にダブってリストされており、ダッシュボードへの表示は China のほうが優先されるので当局はバージョン 1.3.9 以前のものに編集してダブっている China のプリフィックスを削除して従来通り台湾の旗が表示できるようにしました。



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 次のコマンド（#印は含まない）を Raspberry Pi を管理者権限（root）で順次実行していきますがこれらのコマンドをキーボードで打ち込むかコマンド一식을あらかじめ付属の LibreOffice Writer 等で一覧表にしておいてコピーして端末エミュレーター（LXTerminal）に貼り付けるか二通りの方法があります。一覧表方式の場合コマンドごとに必ずエンターキーを打って各コマンド間是一行開けておきコピーの時にコマンドの次の空欄の行からコピーして貼り付けると端末エミュレーター（LXTerminal）ではエンターキーを打つ必要がありませんのでインストール作業が早く終わります。（下図参照）



- ・ 先ず OS のアップデートとアップグレードを行います但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。ここでは先の日本語ソフトをインストールするときに実施しているので省略しました。
 - # 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apache2 と PHP をインストールする。
 - # apt-get install apache2 php5
- ・ 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 フォルダー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を開いて記述を変更する。

```
pico /etc/init.d/xlxd
```

**pico は OS 付属のエディターです。**カーソルの移動はキーボードの上下左右の矢印で行います。Delete キー、BackSpace キーなどカーソル移動以外はワードなどと変わりません。編集の終了は コントロールキーと「x」キーを同時に押した後「y」を入力してエンターキーをクリックしたら 保存して終了します。保存なしで終了するときは「n」を入力してエンターキーをクリックします。

①開いたらその中の ARGUMENTS を 13 ページで設定したローカル固定 IP アドレスに変える。

```
hamradio.dip.jp - pi@ham_radio: ~ - VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
GNU nano 2.2.6 File: /etc/init.d/xlxd Modified
#!/bin/sh
BEGIN INIT INFO
Provides: xlxd service
Processname: xlxd
Required-Start: $remote_fs $syslog
Required-Stop: $remote_fs $syslog
Default-Start: 2 3 4 5
Default-Stop: 0 1 6
Short-Description: starts the dstar reflector as a linux service
Description:
dstar reflector by LX3JL
start script by LX11Q
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="XL595 192.168.x.yyy"
PIDFILE="/var/log/xlxd.pid"
USER=root
GROUP=root

[-f $DAEMON] || exit 0

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

stop() {
 echo -n "Stopping $NAME: "
 start-stop-daemon --stop --pidfile $PIDFILE
 killproc $DAEMON
 RETVAL=$?
 echo
}
```

- ② 次に `start-stop-daemon --start --exec $DAEMON --chuid $USER:$GROUP -background-`  
`-$ARGUMENTS` 行にシャープマーク (#) を付けて次の行に `$DAEMON $ARGUMENTS` を記述する。
- ③ 次に `start-stop-daemon --stop --pidfile $PIDFILE` 行にシャープマーク (#) を付けて  
 次の行に `killproc $DAEMON` の記述を追加する。
- 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` へコピーするが二種類のダッシュボードがあるので  
 次のコマンドのどちらかを選ぶ。
  - ① `# cp -r ~/xlxd/dashboard2/* /var/www/html` ←
  - ② `# cp -r ~/xlxd/dashboard/* /var/www/html` ← 14 ページ上図
- 続いて次のコマンドで `config.inc.php` を開いて・・・  
`# pico /var/www/html/pgs/ config.inc.php`

```

php
/*
Possible values for IPModus
HideIP
ShowFullIP
ShowLast1ByteOfIP
ShowLast2ByteOfIP
ShowLast3ByteOfIP
*/
$Service = array();
$CallingHome = array();
$PageOptions = array();
$PageOptions['ContactEmail'] = 'sc201609@ifty.com'; // Support Es
$PageOptions['DashboardVersion'] = '2.3.1'; // Dashboard Ver$
$PageOptions['PageRefreshActive'] = true; // Activate autos
$PageOptions['PageRefreshDelay'] = '10000'; // Page refresh $
$PageOptions['RepeatersPage'] = array();
$PageOptions['RepeatersPage']['LimitTo'] = 99; // Number of Rep$
$PageOptions['RepeatersPage']['IPModus'] = 'ShowLast1ByteOfIP'; // See ps

```

- ・・・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。
- IP アドレスの桁数は端末エミュレーター画面の上から 5、6、7、8、9 行目にあるスクリプトをコピーして該当する 2 個所に貼り付けます。デフォルトは `ShowfullIP` になっています。国内では `ShowLast1ByteOfIP` か `HideIP` にすればセキュリティーが改善されるでしょう。 `HideIP` にすると IP アドレスは表示されません。

```

pi@ja3gqj: ~
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
GNU nano 2.2.6 File: /var/www/html/pgs/config.inc.php Modified
$CallingHome = array();
$PageOptions = array();
$PageOptions['ContactEmail'] = 'sc201609@ifty.com'; // Support Es
$PageOptions['DashboardVersion'] = '2.3.1'; // Dashboard Ver$
$PageOptions['PageRefreshActive'] = true; // Activate auto$
$PageOptions['PageRefreshDelay'] = '10000'; // Page refresh $
$PageOptions['RepeatersPage'] = array();
$PageOptions['RepeatersPage']['LimitTo'] = 99; // Number of Rep$
$PageOptions['RepeatersPage']['IPModus'] = ShowLast1ByteOfIP'; // See p$
$PageOptions['RepeatersPage']['MasqueradeCharacter'] = '*'; // Chara$
$PageOptions['PeerPage'] = array();
$PageOptions['PeerPage']['LimitTo'] = 99; // Number of pee$
$PageOptions['PeerPage']['IPModus'] = ShowLast1ByteOfIP'; // See p$
$PageOptions['PeerPage']['MasqueradeCharacter'] = '*'; // Character use$
$PageOptions['ModuleNames'] = array(); // Module nomina$
$PageOptions['ModuleNames']['A'] = 'Int.';
$PageOptions['ModuleNames']['B'] = 'Regional';
$PageOptions['ModuleNames']['C'] = 'National';
$PageOptions['ModuleNames']['D'] = '';

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

\$CallingHome Active: デフォルトの fales を ture にする。

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

この例では <http://xlxreflctor.jpn.ph/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。

```

GNU nano 2.2.6 File: /var/www/html/pgs/config.inc.php
$PageOptions['ModuleNames']['C'] = 'National';
$PageOptions['ModuleNames']['D'] = '';

$PageOptions['MetaDescription'] = 'XLX is a D-Star Reflector System for Ham Radio Operators.'; // Meta Tag Values, usef$
$PageOptions['MetaKeywords'] = 'Ham Radio, D-Star, XReflector, XLX, XRF, DCS, REF, '; // Meta Tag Values, usef$
$PageOptions['MetaAuthor'] = 'LX110'; // Meta Tag Values, usef$
$PageOptions['MetaRevisit'] = 'After 30 Days'; // Meta Tag Values, usef$
$PageOptions['MetaRobots'] = 'index, follow'; // Meta Tag Values, usef$

$PageOptions['UserPage']['ShowFilter'] = true; // Show Filter on Users $

$Service['PIDFile'] = '/var/log/xlxd.pid';
$Service['XMLFile'] = '/var/log/xlxd.xml';

$CallingHome['Active'] = true; // xlx phone home, true or fal$
$CallingHome['MyDashBoardURL'] = 'http://hamradio.dip.jp/index.php'; // dashboard url
$CallingHome['ServerURL'] = 'http://xlxapi.fix.ku.api.php'; // database server, do not change !!!!
$CallingHome['PushDelay'] = 600; // push delay in seconds
$CallingHome['Country'] = 'JAPAN'; // Country
$CallingHome['Comment'] = 'Welcome beginner'; // Comment. Max 100 charac$
$CallingHome['HashFile'] = 'callhome/callinghome.php'; // Make sure the apache user has read and$

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

\$CallingHome MyDashBoardURL: デフォルトの tmp を callhome に書き換える。

次の 3 項目の設定 (記述) が正しくないとダッシュボードが動作しません「」一つ  
欠けても NG です。

\$CallingHome Active、\$CallingHome MyDashBoardURL、  
\$CallingHome MyDashBoardURL

- 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
```

- 再起動する

```
reboot
```

- 再起動後、ターミナルエミュレーター（LXterminal）を開いて次のコマンドをコピー（[巻末 Appendix③参照](#)）して貼り付けるとコマンドが順次自動的に実行され最後に稼働状況が表示される。（コピー貼り付け用には必ず各コマンド間に空白行を入れて最後のコマンド行の次の空白行からコピーして這いつけること。）

```
再起動後の一括コマンド

sudo -s

cd

service xlxd stop

service xlxd start

cd /etc/init.d

./xlxd start

./xlxd status
```

これらコマンドが順に自動実行された後、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。

```
root@a3gqj: /home/p1# cd
root@a3gqj: ~# cd /etc/init.d
root@a3gqj: /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 ExecStart=/etc/init.d/xlxd start (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 XLX052 192.168.1.200

7月 07 05:29:08 ja3gqj xlxd[31411]: Sending connect packet to XLX peer XLX... 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 XLX... 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 XLX... 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 XLX... 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 XLX... 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@a3gqj: /etc/init.d#
```

- 正しく起動していても 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。

```
p1@a3gqj: ~$ sudo -s
root@a3gqj: /home/p1# cd
root@a3gqj: ~# cd /etc/init.d
root@a3gqj: /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[450]: Gatekeeper loaded 1 lines from /xlxd/xl... st
7月 04 07:51:17 ja3gqj xlxd[450]: Gatekeeper loaded 0 lines from /xlxd/xl... st
7月 04 07:51:17 ja3gqj xlxd[450]: Gatekeeper loaded 0 lines from /xlxd/xl... nk
7月 04 07:51:17 ja3gqj xlxd[450]: Error opening socket on port UDP30001 o... 00
7月 04 07:51:17 ja3gqj xlxd[450]: Error opening socket on port UDP30001 o... 00
7月 04 07:51:17 ja3gqj xlxd[450]: Error opening socket on port UDP30001 o... 00
7月 04 07:51:17 ja3gqj xlxd[450]: Error opening socket on port UDP10002 o... 00
7月 04 07:51:17 ja3gqj xlxd[450]: 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@a3gqj: /etc/init.d# cd
root@a3gqj: ~#
```

- 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（#印は含まない）で手動による確実な起動出来ます。

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

```
pi@ja3ggj: ~
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@ja3ggj: /etc/init.d# cd
root@ja3ggj: ~# cd /etc/init.d
root@ja3ggj: /etc/init.d# ./xlxd stop
[ok] Stopping xlxd (via systemctl): xlxd.service.
root@ja3ggj: /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)

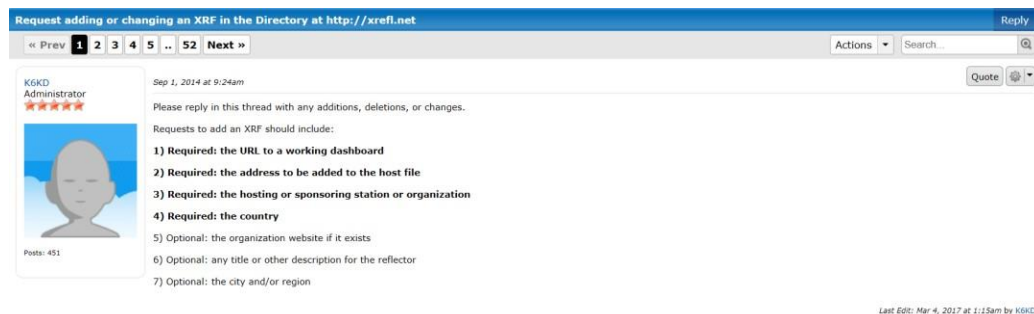
7月 04 07:51:17 ja3ggj xlxd[481]: Gatekeeper loaded 0 lines from /xlxd/xl...nk
7月 04 07:51:17 ja3ggj xlxd[481]: Error opening socket on port UDP30001 o...00
7月 04 07:51:17 ja3ggj xlxd[481]: Error opening socket on port UDP20001 o...00
7月 04 07:51:17 ja3ggj xlxd[481]: Error opening socket on port UDP30051 o...00
7月 04 07:51:17 ja3ggj xlxd[481]: Error opening socket on port UDP10002 o...00
7月 04 07:51:17 ja3ggj xlxd[481]: Error starting reflector
7月 04 07:51:21 ja3ggj systemd[1]: Started LSB: starts the dstar reflecto...e.
7月 04 08:19:21 ja3ggj systemd[1]: Stopping LSB: starts the dstar reflect....
7月 04 08:19:21 ja3ggj xlxd[1127]: Stopping xlxd:
7月 04 08:19:21 ja3ggj systemd[1]: Stopped LSB: starts the dstar reflecto...e.
Hint: Some lines were ellipsized, use -l to show in full.
root@ja3ggj: /etc/init.d#
```

XLX マルチプロトコル・リフレクター・ゲートウェイの広報

<http://xrefl.boards.net/thread/2/request-adding-changing-directory-xrefl?page=1>

(下図) のように X Reflector Forum で、次の 1) ~ 7) までの項目を記入して John K6KD へ「XLX Multiprotocol Reflector Gateway」で取得した XLXxxx の登録を依頼する。

- 1) Required: the URL to a working dashboard
- 2) Required: the address to be added to the host file
- 3) Required: the hosting or sponsoring station or organization
- 4) Required: the country
- 5) Optional: the organization website if it exists
- 6) Optional: any title or other description for the reflector
- 7) Optional: the city and/or region



## ソフトのバージョンアップ

バージョンアップ情報は <https://groups.yahoo.com/neo/groups/ircDDBGateway/info> のフォーラムから入手しますがバージョンアップの方法には二通りあると思います。

1. カスタマイズをしていないとき。



- d. xlx/xd/src の xlx/xd を /xlxd にコピーする。  
#cp xlx/xd /xlxd (コピーするときのディレクトリーは xlx/xd/src/)
- e. xlx/xd を起動する。  
# service xlx/xd start
- f. 新しいダッシュボードファイルを古いファイルに上書きする。  
# cp -r ~/xlxd/dashboard/\* /var/www/html
- g. /var/www/html/pgs の中ある config.inc.php を開いて新バージョンであることを確認する。  
# pico /var/www/html/pgs/config.inc.php

```
$PageOptions['DashboardVersion'] = '2.3.7';
$PageOptions['PageRefreshActive'] = true;
$PageOptions['PageRefreshDelay'] = '10000';
```

- h. ディスクトップのバックアップして置いた config.inc.php にダッシュボードの新しいバージョンを書き込んで、/var/www/html/pgs に上書きして貼り付ける。  
# pico /home/pi/Desktop/config.inc.php (バックアップしたファイルを開く)

```
$PageOptions['ContactEmail'] = 'hamradio@hera.eonet.ne.jp';$
$PageOptions['DashboardVersion'] = '2.3.5';$
$PageOptions['PageRefreshActive'] = true;$
$PageOptions['PageRefreshDelay'] = '10000';$
```

2.3.5 を確認したバージョン 2.3.7 に書き換える

書き換えたファイルを元のフォルダーに戻す。

# cp /home/pi/Desktop/config.inc.php /var/www/html/pgs

- i. callinghome.php を元に戻す。  
# cp /home/pi/Desktop/callinghome.php /callhome

## 2. カスタマイズをしているときは

- a. カスタマイズの仕方によりますがカスタマイズしたファイル (フォルダー) をバックアップする。(画像関係は別ホルダーにして置くとバックアップの必要がない)
- b. 前項の a. ~ i. を行う。
- c. 続いてカスタマイズしたファイルを元に戻す。 当局の例では  
/var/www/html/index.php, /var/www/html/pgs/peers.php と  
/var/www/html/pgs に追加した php ファイル (別ホルダーから元に戻す) が対象になります。

## おわりに

XLX リフレクターのバージョンアップに翻弄されてきましたがその方法が確認できてやっと落ち着きました。今後、勉強 (確認) しなければならないのは XLX リフレクターの仕組みで、我々が使用しているダッシュボードの位置づけです。2017/04/12 現在リフレクター一覧表には 160 のリフレクターが稼働していますがこれらリフレクターはどのように関わっているのか、何処かに親サーバーがあってこの親サーバーを通じて繋がっているのかそれとも親サーバーはリフレクターの自動登録の管理だけで個々のリフレクター同志が直接つながって分散型のサーバーを形成しているのか、そのいずれにしても何が原因で交信中に音声途切れるのか事があるのか、インターネット回線の影響を受けていることは他のインターネット通信ソフト (CQ-100 や HamsPhere など) を使用した体験から各国の回線事情や通信時間帯やインターネット利用者が多い地域などの影響があるように思います。昨年 3 月頃のリフレクター数は 100 未満でしたがその後増加して現在は 176 です。(その他ダッシュボードのリストに表示されないマルチではないシンプルな XRF リフレクターが 10 ほどあるようです) 最高 1000 のリフレクターが開設できるようですがインターネット回線を利用する関係でリフレクター数の増加は通信速度に影響を与える要因ではないかと思っています。現在のリフレクターでそのダッシュボードのアクセス時間に差がありますがこれも

音声伝達に何らかの影響があるような気がします。更に一つのリフレクターに接続している利用者が多いと音声伝達に遅れが出るのではないかと思います。最後にリフレクターに無線局をリンクしたときのデータが全くありません。一対一のリンクが最適な条件であることは推測できますが、リフレクター1に対して無線局は何局までリンクできるのか、インターネット回線内のデータ移動時間と無線通信でのデータ移動時間は異なりインターネット回線内のデータ移動時間は変動するが無線通信でのデータ移動時間は変動しないと考えられますのでこれらを意識して感覚的なデータがとれたらと思います。

## Appendix①

### Vsftpd ファイルサーバーの構築

XLX リフレクターソフトが働いている Raspberry Pi のエミュレーターTXTerminal を管理者(root) 権限にして次のコマンドで vsftpd をインストールします。(管理権限するとコマンドの前の sudo が省略できる、以下コマンド入力には#印は含まない。)

```
apt-get update
apt-get upgrade
apt-get install vsftpd
```

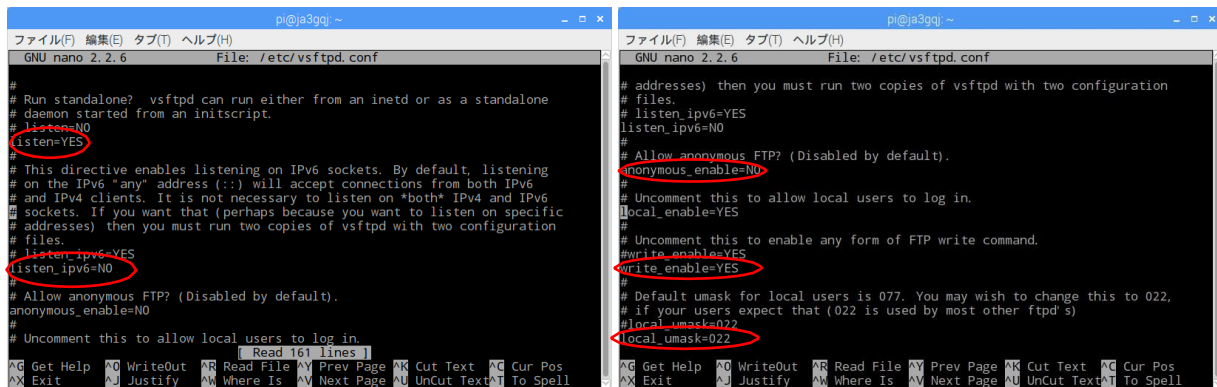
次のコマンド（#印は含まない）でポップアップした画面で以下の項目を編集する。

```
pico /etc/vsftpd.conf
```

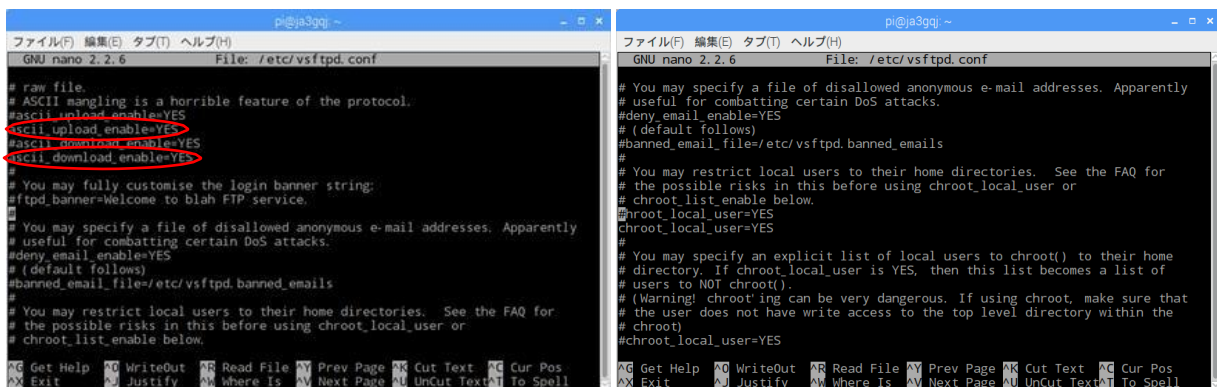
取り合えず次の項目を編集してお馴染みの ffftp を

パソコン (Windows 10) にインストール設定後、Raspberry Pi の接続が出来ました。ポイントは デフォルトの listen=NO を YES に同じく listen\_ipv6=YES を NO にしないと接続できなかったところ

です。  
listen=YES, listen\_ipv6=NO, anonymous\_enable=NO, write\_enable=YES, local\_umask=022,  
ascii\_upload\_enable=YES, ascii\_download\_enable=YES, chroot\_local\_user=YES,  
chroot\_local\_list=YES, chroot\_list\_file=/etc/vsftpd.chroot\_list



```
pi@ja3gq: ~
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
GNU nano 2.2.6 File: /etc/vsftpd.conf
Run standalone? vsftpd can run either from an inetd or as a standalone
daemon started from an initscript.
#listen=NO
listen=YES
This directive enables listening on IPv6 sockets. By default, listening
on the IPv6 "any" address (:::) will accept connections from both IPv6
and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
sockets. If you want that (perhaps because you want to listen on specific
addresses) then you must run two copies of vsftpd with two configuration
files.
#listen_ipv6=YES
listen_ipv6=NO
Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
Uncomment this to allow local users to log in.
#local_enable=YES
Uncomment this to enable any form of FTP write command.
#write_enable=YES
write_enable=YES
Default umask for local users is 077. You may wish to change this to 022.
if your users expect that (022 is used by most other ftpd's)
#local_umask=022
local_umask=022
AG Get Help AO WriteOut AR Read File AY Prev Page AX Cut Text AG Cur Pos
XX Exit AJ Justify AL Where Is AV Next Page AU UnCut Text AT To Spell
```



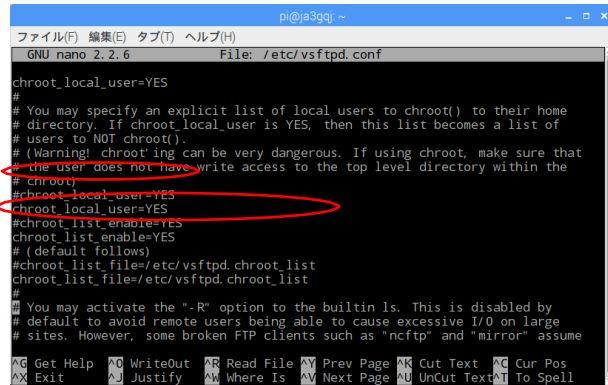
```
pi@ja3gq: ~
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
GNU nano 2.2.6 File: /etc/vsftpd.conf
raw file
ASCII mangling is a horrible feature of the protocol.
#ascii_upload_enable=YES
ascii_upload_enable=YES
#ascii_download_enable=YES
ascii_download_enable=YES
You may fully customise the login banner string:
#ftpd_banner=WELCOME to blah FTP service.
#
You may specify a file of disallowed anonymous e-mail addresses. Apparently
useful for combatting certain DoS attacks.
#deny_email_enable=YES
(default follows)
#banned_email_file=/etc/vsftpd.banned_emails
#
You may restrict local users to their home directories. See the FAQ for
the possible risks in this before using chroot_local_user or
chroot_list_enable below.
#chroot_local_user=YES
chroot_local_user=YES
You may specify an explicit list of local users to chroot() to their home
directory. If chroot_local_user is YES, then this list becomes a list of
users to NOT chroot().
(Warning! chroot'ing can be very dangerous. If using chroot, make sure that
the user does not have write access to the top level directory within the
chroot)
#chroot_local_list=YES
chroot_list_file=/etc/vsftpd.chroot_list
AG Get Help AO WriteOut AR Read File AY Prev Page AX Cut Text AG Cur Pos
XX Exit AJ Justify AL Where Is AV Next Page AU UnCut Text AT To Spell
```

ベテランの人には当たり前ですが vsftpd.conf の内容は#印で設定の意味が説明されていますのでそれが理解できれば上の編集で不要な 箇所があるかもしれません。またデフォルトで設定項目に#印がついていないところは そのままで上の項目は#印が付いている処 ばかりですがその行はそのまま残してその 次に#印が付かない行を追加します。

設定の編集が終わったら次に ffftp からアクセスさせるためのユーザーを指定する新しいファイルを作ります。

```
pico /etc/vsftpd.chroot_list
```

vsftpd.chroot\_list は新しく作るファイルの名前で上のコマンドでポップアップするブランクの 端末エミュレーターLXTeminal に「pi」と打ち込んで保存するだけです。保存方法は本文 13 ページ を参照ください。



```
pi@ja3gqj ~
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
GNU nano 2.2.6 File: /etc/vsftpd.conf

chroot_local_user=YES
#
You may specify an explicit list of local users to chroot() to their home
directory. If chroot_local_user is YES, then this list becomes a list of
users to NOT chroot().
(Warning! chrooting can be very dangerous. If using chroot, make sure that
the user does not have write access to the top level directory within the
chroot.)
#chroot_local_user=YES
chroot_local_user=YES
#chroot_list_enable=YES
chroot_list_enable=YES
(default follows)
#chroot_list_file=/etc/vsftpd.chroot_list
chroot_list_file=/etc/vsftpd.chroot_list
#
You may activate the "-R" option to the builtin ls. This is disabled by
default to avoid remote users being able to cause excessive I/O on large
sites. However, some broken FTP clients such as "ncftp" and "mirror" assume
#
AG Get Help AG WriteOut AR Read File AY Prev Page AK Cut Text AC Cur Pos
AX Exit AJ Justify AW Where Is AN Next Page AU UnCut Text AT To Spell

Read 1 line
```



```
pi@ja3gqj ~
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
GNU nano 2.2.6 File: /etc/vsftpd.chroot_list

pi

AG Get Help AG WriteOut AR Read File AY Prev Page AK Cut Text AC Cur Pos
AX Exit AJ Justify AW Where Is AN Next Page AU UnCut Text AT To Spell

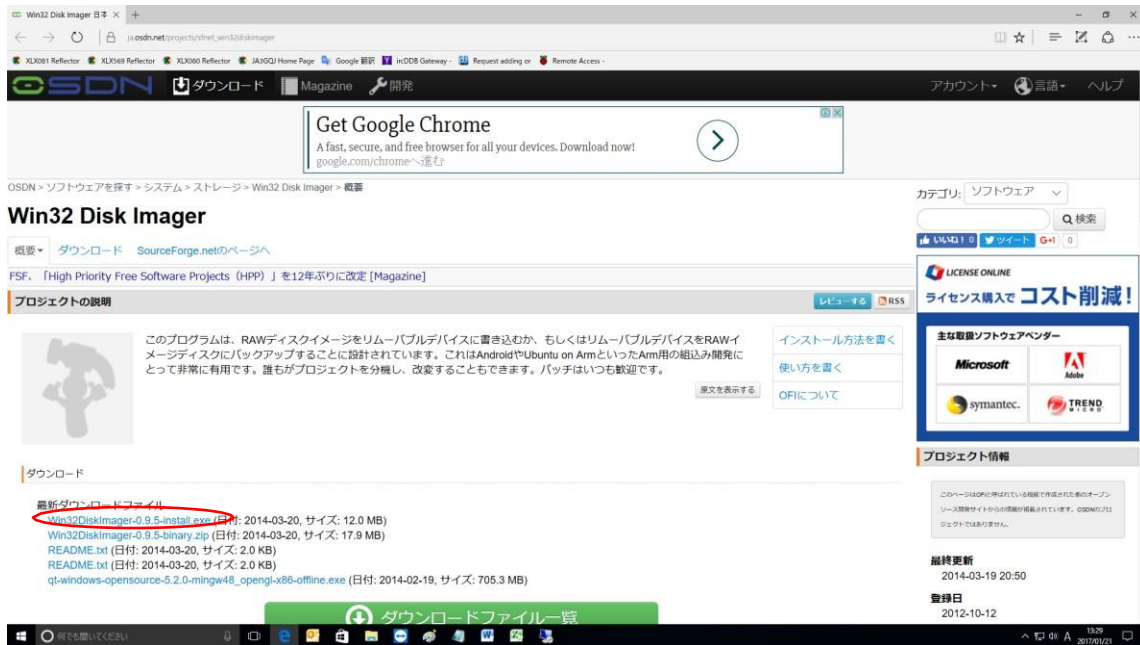
Read 1 line
```

以上

## Appendix②

### Win32 Disk Imager のダウンロード

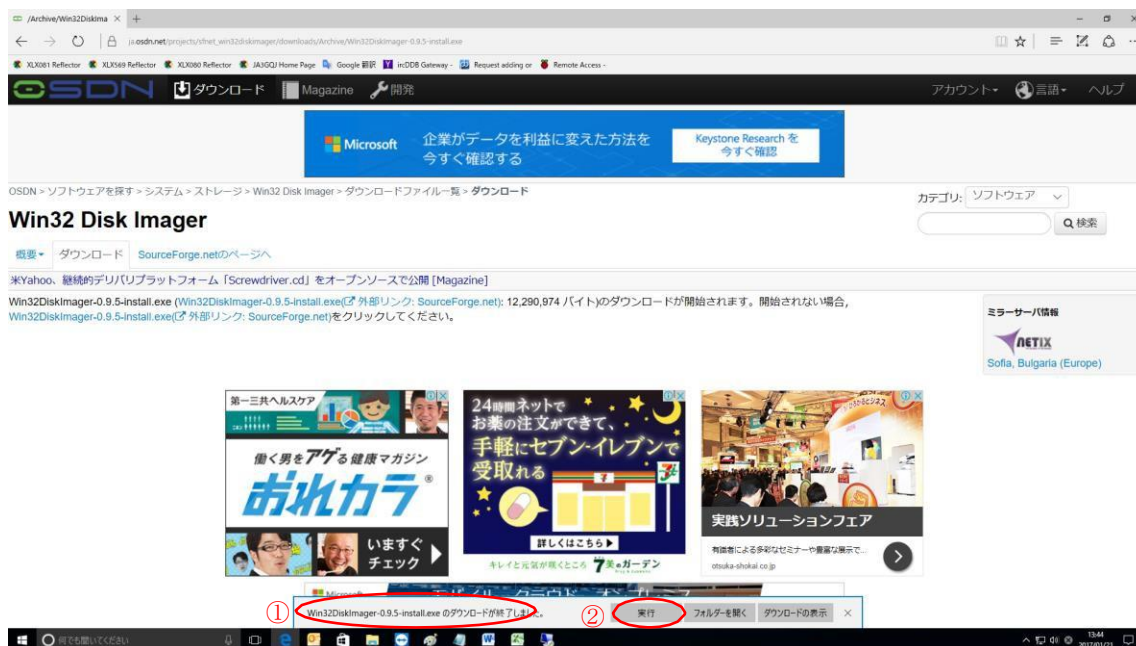
Window 10 パソコンで [https://osdn.jp/projects/sfnet\\_win32diskimager/](https://osdn.jp/projects/sfnet_win32diskimager/) を開いて Win32DiskImager-0.9.5-install.exe (日付: 2014-03-20, サイズ: 12.0 MB)をクリック (下図)



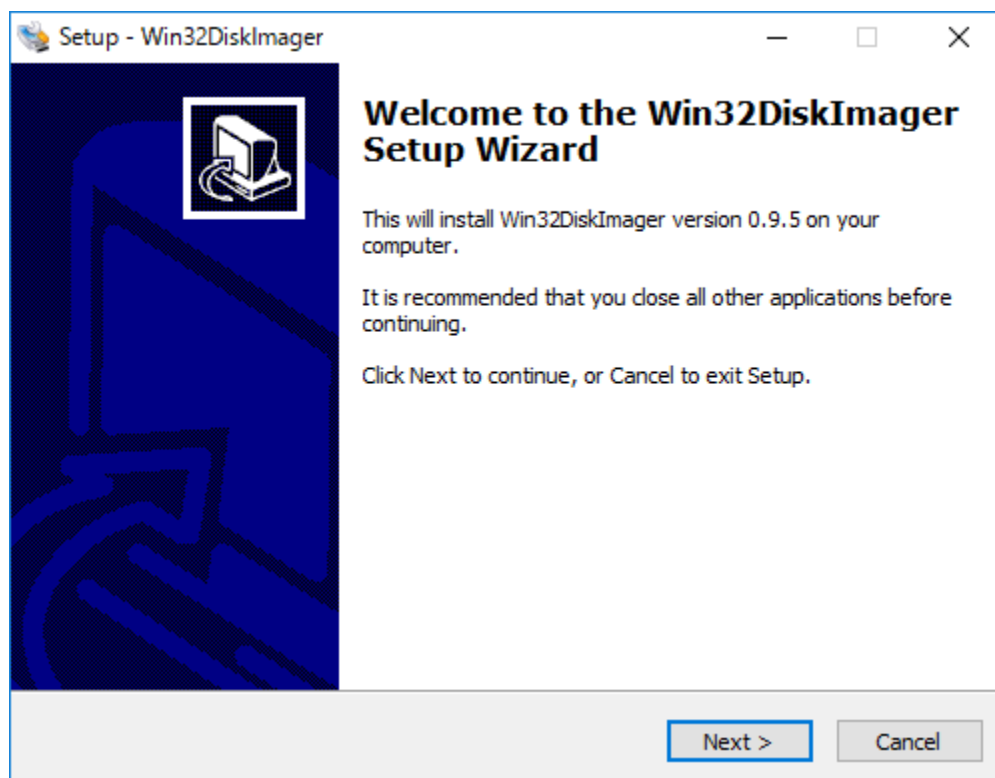
次に開いた画面 (下図) の下にある「Win32-DiskImager-0.9.5-install.exe(11.7MB)について行う操作を選んでください」①で「保存」ボタン②を押してダウンロードする。



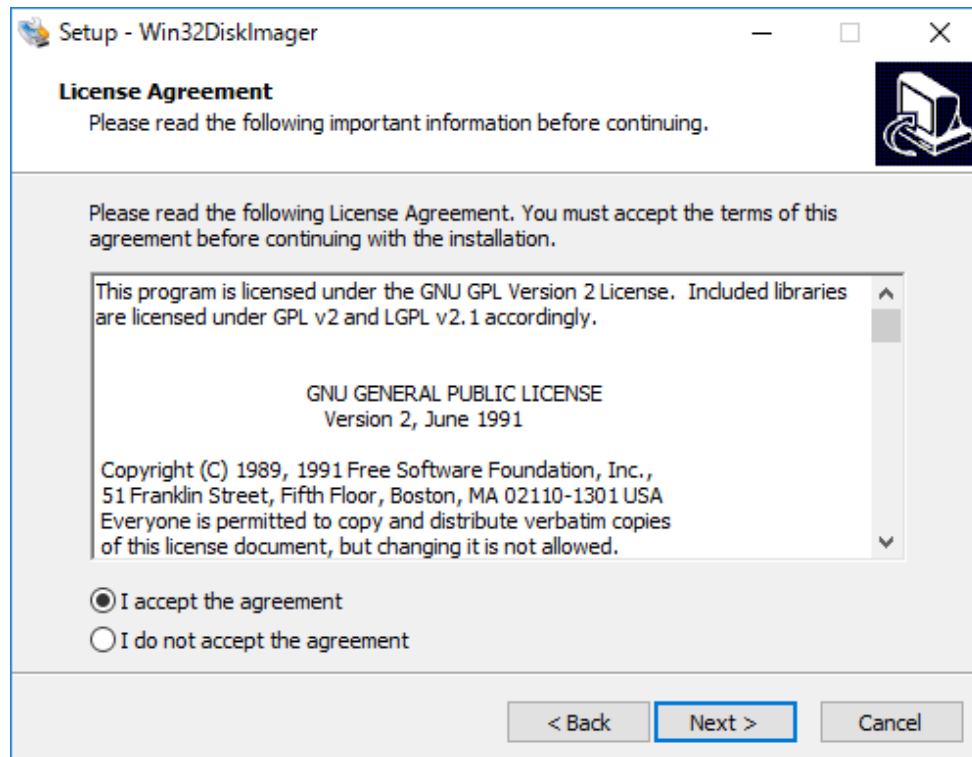
暫くして「Win32-DiskImager-0.9.5-install.exe のダウンロードが終了しました」①と表示が変わったらその右の「実行」②ボタンをクリックしてインストールを開始する。



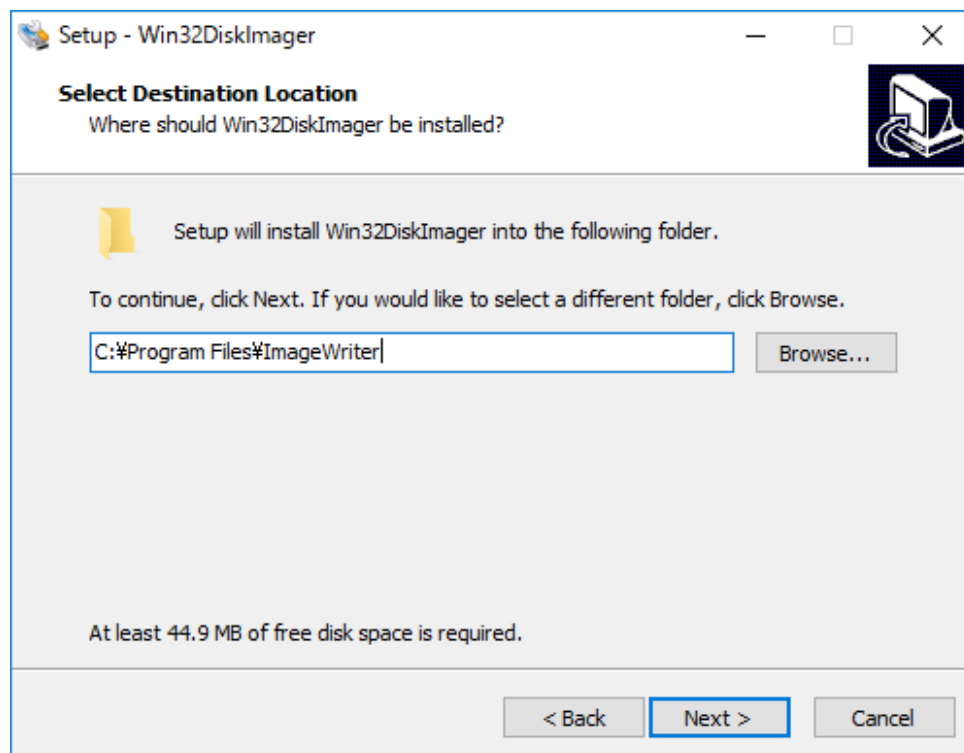
開始直後には「この不明な発行元からのアプリがデバイスに変更を加えることを許可しますか」とのコメント画面が出るので「はい」をクリックして次に進むと下の図がポップアップするので「Next」ボタンをクリックする。



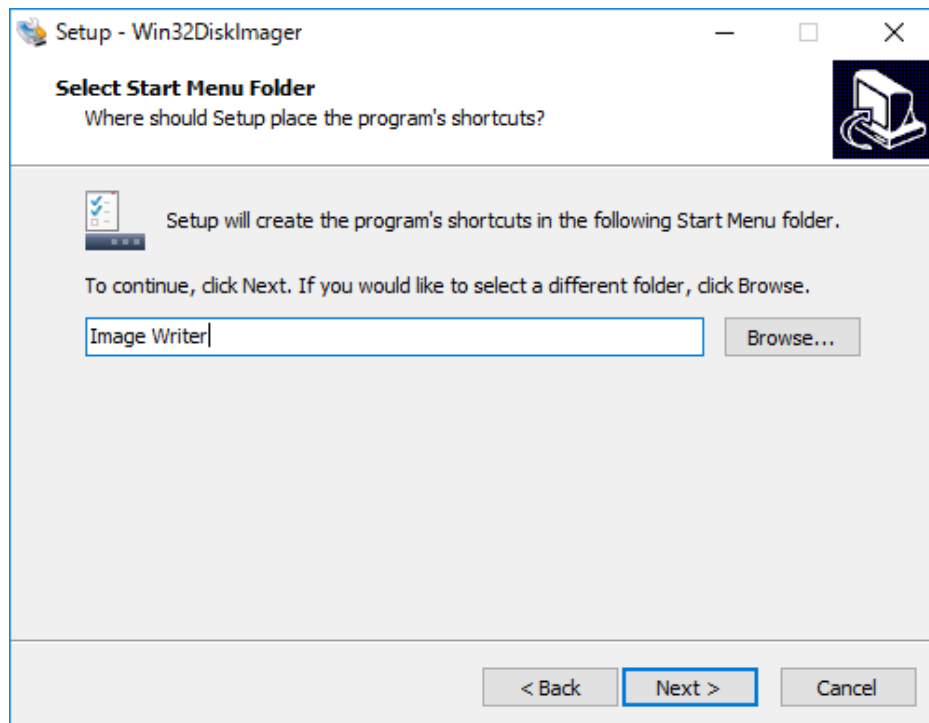
次にポップアップした画面で「I accept the agreement」を選択したら「Next>」ボタンで次に進む。



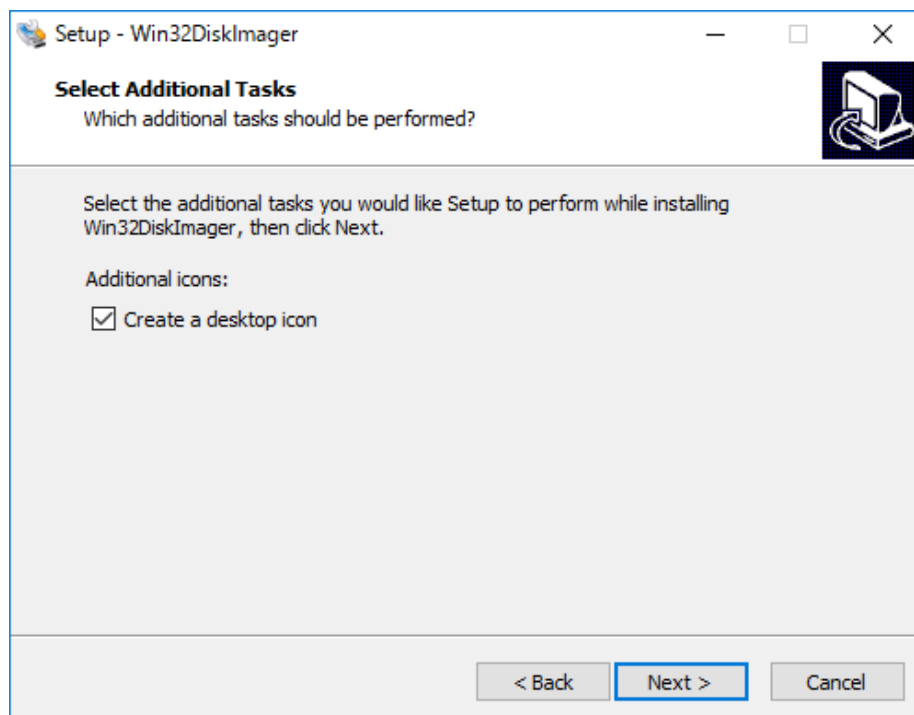
ファイルのインストール先は下図の通りデフォルトで「Next>」ボタンで次に進む。



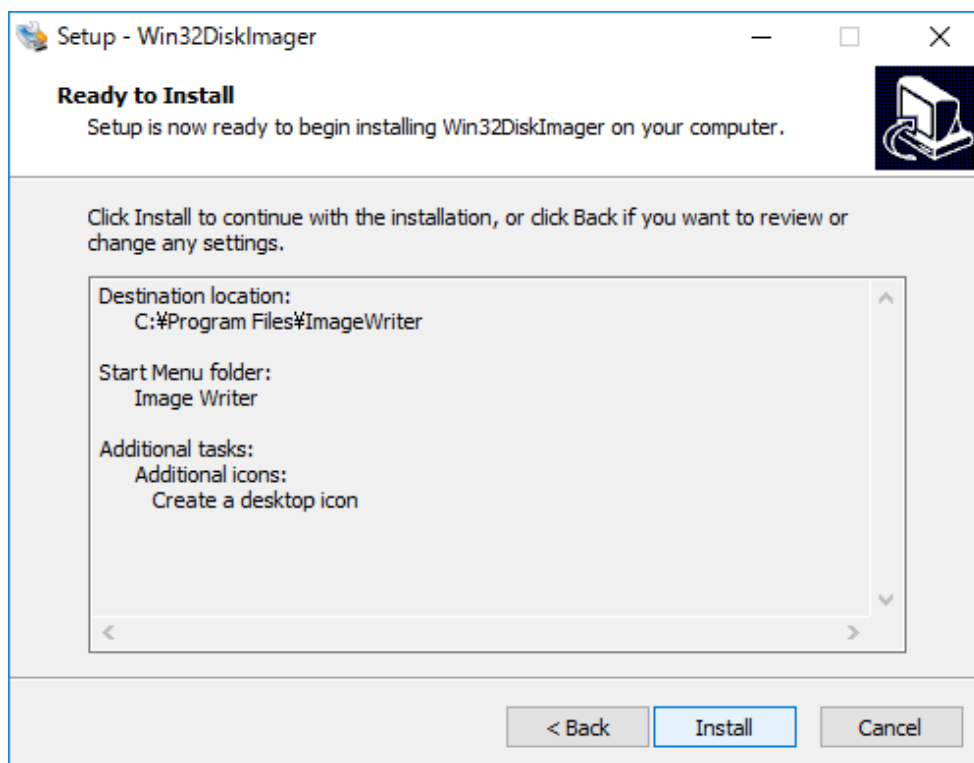
ファイルをインストールするフォルダ名は下図の通りデフォルトで「Next>」ボタンで次に進む。



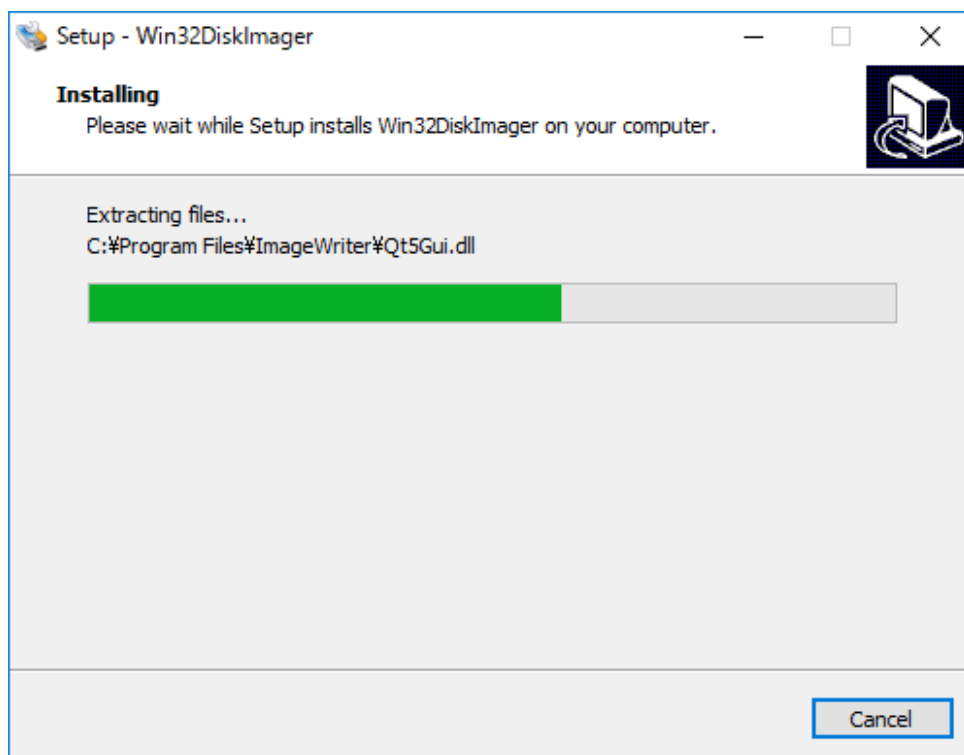
「Create a desktop icon」にチェックマークを付けて「Next>」ボタンで次に進む。



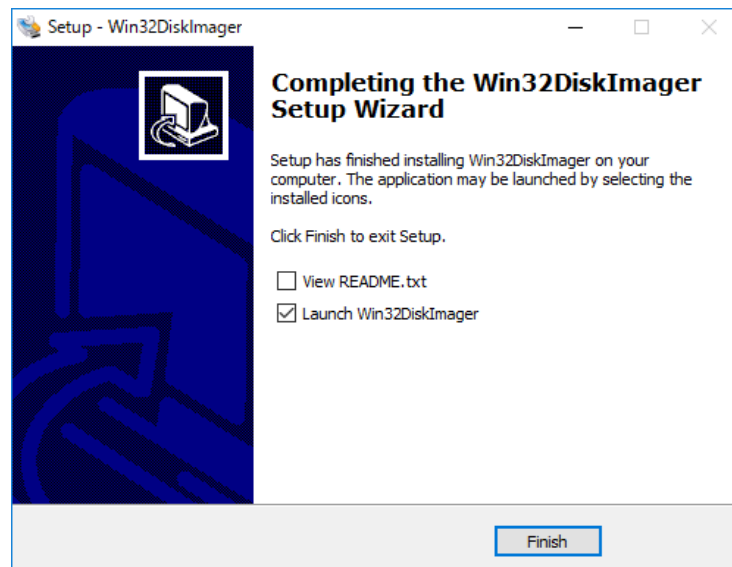
次にポップアップした画面は「Next>」ボタンをクリックするだけ



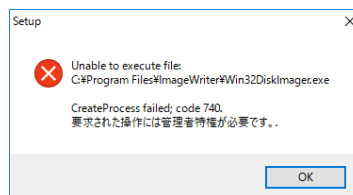
次の画面でグリーンのプログレスバーでインストールの進行状態が表示されて



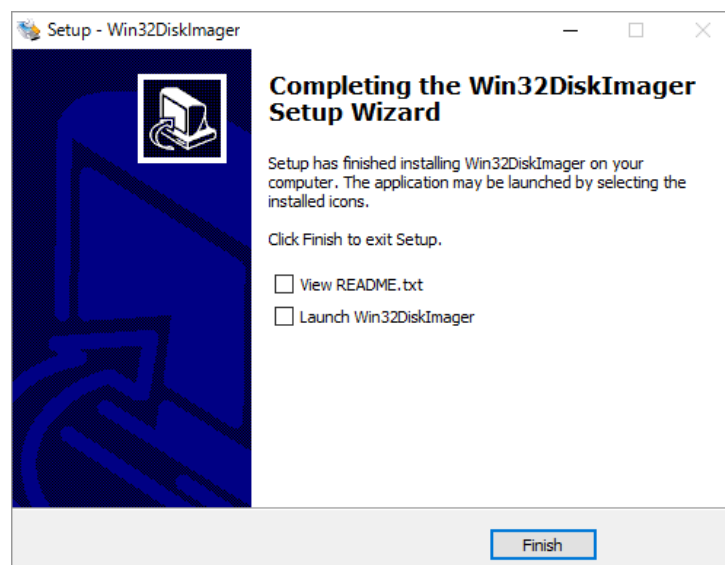
インストール終了前の画面で「View README.txt」のチェックマークを外して「Finish」ボタンで終了しようとしても



次のようなコメント画面がポップアップしてしまうので



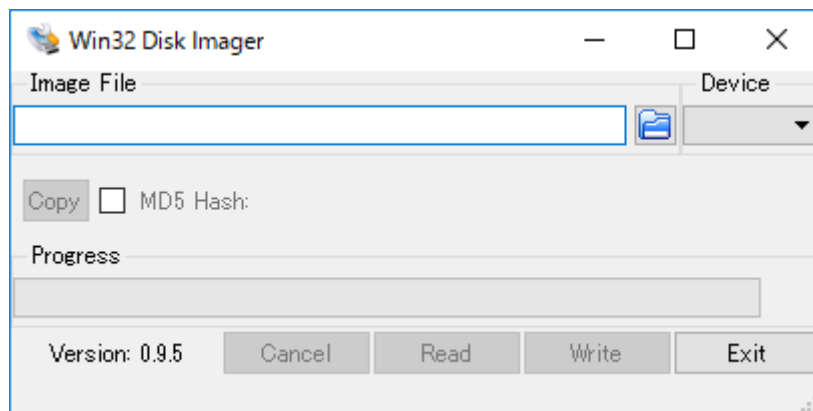
下図のように「View README.txt」と「Launch Win32DiskImager」両方のチェックマークを外して「Finish」ボタンで一旦終了して



デスクトップのアイコン（下図）をクリックして起動するが



起動前に再び「この不明な発行元からのアプリがデバイスに変更を加えることを許可しますか」のコメント画面がポップアップするので「はい」をクリックしてプログラムを起動（下図）する。

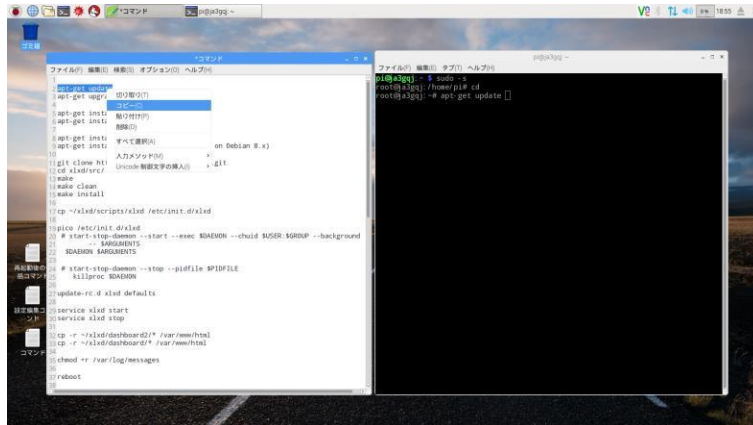


以上

## Appendix③

### 便利グッズ

- ① XLX マルチプロトコルリフレクターをインストールしてセットアップするときのコマンドの一覧表を作っておいてそれをコピーしてターミナルエミュレーターに貼り付けると便利。



#### コマンド一覧表の一例

「//」は原文のコメント。「#」はコマンドではないのでコピーしないこと。

「pico」はエディターです。「赤字」は当局が追加したコマンド

Debian 7 (Wheezy) 32 and 64bit: (原文)

```
// after a clean installation of debian make sure to run update and upgrade
apt-get update
apt-get upgrade
```

```
// install git
apt-get install git git-core
```

```
// install webserver with PHP5 support
apt-get install apache2 php5
```

```
//install g++ compiler
apt-get install build-essential
apt-get install g++-4.7 (原文のコメントはカット)
```

```
//download and compile the XLX sources
git clone https://github.com/LX3JL/xlxd.git
cd xlxd/src/
make
make clean
make install
```

```
// copy startup script "xlxd" to /etc/init.d
```

```
cp ~/xlxd/scripts/xlxd /etc/init.d/xlxd

mkdir callhome
chmod 777 callhome

// adapt the default startup parameters to your needs
pico /etc/init.d/xlxd
「start-stop-daemon --start --exec $DAEMON --chuid $USER:$GROUP --background
 -- $ARGUMENTS」 のところを
「 $DAEMON $ARGUMENTS 」に変えるのでコピーしてペーストするのは$DAEMON $ARGUMENTS
だけ
「start-stop-daemon --stop --pidfile $PIDFILE」のところを
「killproc $DAEMON」に変えるのでコピーしてペーストするのはkillproc $DAEMON だけ

// Last step is to declare the service for automatic startup and shutdown
update -rc.d xlxd defaults

// start or stop the service with:
service xlxd start
service xlxd stop

// copy dashboard to /var/www
cp -r ~/xlxd/dashboard/* /var/www/html

// give the dashboard read access to the server log file
chmod +r /var/log/messages

// reboot server to see if the auto-start is working
reboot
```

- ② 再起動後に必要なコマンドの一覧表を作っておいてそのすべてをコピーしてターミナルエミュレーターに貼り付ける。すべてのコマンドが自動的に実行されるので便利。各コマンド行間は必ず一行開けることと最後のコマンド 行の次の行までコピーすることがポイントです。

```
sudo -s

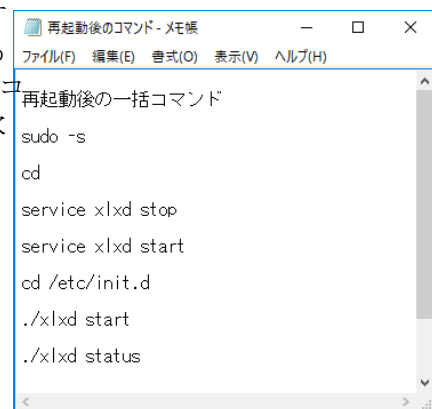
cd

service xlxd stop

service xlxd start

cd /etc/init.d

./xlxd stop
```



```
./xld start
```

```
./xld status
```

- ③ ソフトをバージョンアップするときのコマンド一覧表を作っておいてそのすべてをコピーしてターミナルエミュレーターに貼り付けるとすべてのコマンドが自動的に実行されるので便利。

```
sudo -s
```

```
cd
```

```
service xld stop
```

```
chmod 777 xld
```

```
rm -r xld
```

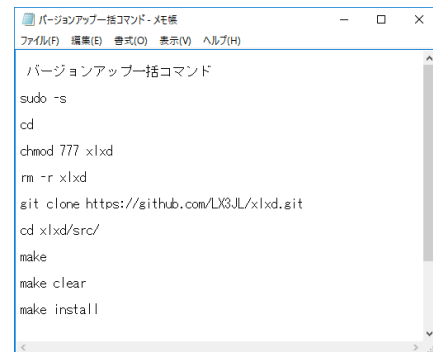
```
git clone https://github.com/LX3JL/xld.git
```

```
cd xld/src/
```

```
make
```

```
make clear
```

```
make install
```



以上

## Appendix④

### 1. index.php にボタンを追加する。

Pico や nano 等エディターで index.php を開き . . . . .

```
pi@ja3gqj:~$ sudo -s
root@ja3gqj:/home/pi# cd
root@ja3gqj:~# pico /var/www/html/index.php
```

まず、追加ボタン用に次の記述を書き加えます。（記述の位置①は下図参照） デフォルトの五つのボタンの記述は一行で大変長く最後のボタンの記述は画面のかなり左に 移動しないと見えませんがその後ろに追加ボタンを記述しますけど下図の通り<td>〜</td>の部分を改行して5行に分けて記述します。ホームページを開設している方はご存知の通り<tr>〜</tr>は表の行、<td>〜</td>表の行を作ります。

```
<td>説明</td>
<td>体験記 1</td>
<td>体験記 2</td>
<td>IC-7100 の活用</td>
<td>リンク</td>
```

```
<?
</head>
<body>
<?php if (file_exists("../tracking.php")) { include_once("tracking.php"); }?>
<div id="top">

<?php echo $Reflector->GetReflectorName(); ?><?php echo $Reflector->GetVersion(); ?><?php echo $Reflector->GetVersion(); ?>
<div id="menubar">
<div id="menu">
<table border="0">
<tr>
<td>Users / Modules</td>
<td>説明</td>
<td>体験記 1</td>
<td>体験記 2</td>
<td>IC-7100 の活用</td>
<td>リンク</td>
</tr>
</table>
</div>
</div>
<div id="content" align="center">
<?php
if ($CallingHome['Active']) {
if (!is_readable($CallingHome['HashFile']) && (!is_writable($CallingHome['HashFile']))) {
echo
<div class="error">
your private hash in ".$CallingHome['HashFile']. " could not be created, please check your config file and the permissions $
</div>
}
}

switch ($_GET['show']) {
case 'users' : require_once("../pgs/users.php"); break;
case 'repeaters' : require_once("../pgs/repeaters.php"); break;
case 'liveiroddb' : require_once("../pgs/liveiroddb.php"); break;
case 'peers' : require_once("../pgs/peers.php"); break;
case 'reflectors' : require_once("../pgs/reflectors.php"); break;
① case 'xlxinfo' : require_once("../pgs/xlsxinfo.php"); break;
case 'xlxdown1' : require_once("../pgs/xlsxdown1.php"); break;
case 'xlxdown2' : require_once("../pgs/xlsxdown2.php"); break;
case 'xlxdown3' : require_once("../pgs/xlsxdown3.php"); break;
case 'ja3gqj' : require_once("../pgs/ja3gqj.php"); break;
default : require_once("../pgs/users.php");
}
?>
```



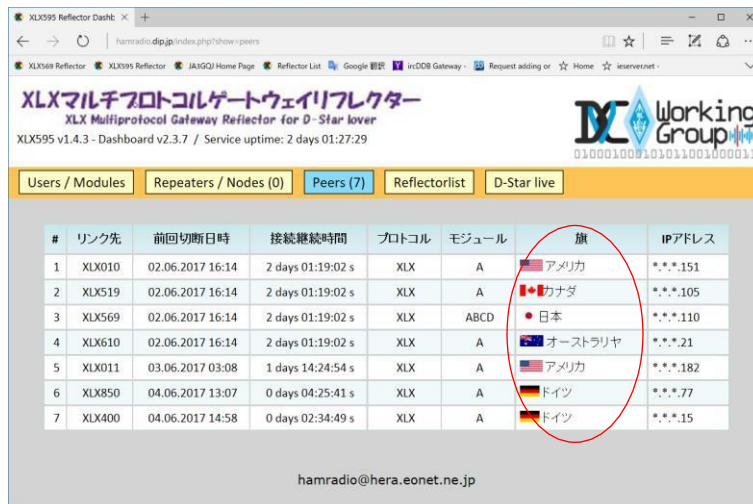
次の記述位置は前頁①参照、和文の処はデフォルトの英文でも良い。

```
<th width="25">#</th>
<th width="80">リンク先</th>
<th width="145">前回切断日時</th>
<th width="145">接続継続時間</th>
<th width="90">プロトコル</th>
<th width="90">モジュール</th>
<th width="164">旗</th>
```

次の記述位置は前頁②参照。記述行はリンク先の数だけ必要。旗はから該当画像を持ってくる。

```
<td>';
switch ($Reflector->Peers[$i]->GetCallSign()) {
 case 'XLX010' : echo '';echo ' アメリカ'; break;
 case 'XLX011' : echo '';echo ' アメリカ'; break;
 case 'XLX610' : echo '';echo ' オーストラリア'; break;
 case 'XLX888' : echo '';echo ' イタリア'; break;
 case 'XLX400' : echo ''; echo ' ドイツ'; break;
 case 'XLX850' : echo ''; echo ' ドイツ'; break;
 case 'XLX519' : echo ''; echo ' カナダ'; break;
 case 'XLX302' : echo ''; echo ' カナダ'; break;
 case 'XLX569' : echo ''; echo ' 日本'; break;
 case 'XLX303' : echo ''; echo ' カナダ'; break;
 default :
}';</td>
```

結果下図のような peers (インターリンク) の画面になる。



#	リンク先	前回切断日時	接続継続時間	プロトコル	モジュール	旗	IPアドレス
1	XLX010	02.06.2017 16:14	2 days 01:19:02 s	XLX	A	 アメリカ	*.*.151
2	XLX519	02.06.2017 16:14	2 days 01:19:02 s	XLX	A	 カナダ	*.*.105
3	XLX569	02.06.2017 16:14	2 days 01:19:02 s	XLX	ABCD	 日本	*.*.110
4	XLX610	02.06.2017 16:14	2 days 01:19:02 s	XLX	A	 オーストラリア	*.*.21
5	XLX011	03.06.2017 03:08	1 days 14:24:54 s	XLX	A	 アメリカ	*.*.182
6	XLX850	04.06.2017 13:07	0 days 04:25:41 s	XLX	A	 ドイツ	*.*.77
7	XLX400	04.06.2017 14:58	0 days 02:34:49 s	XLX	A	 ドイツ	*.*.15

4. ボタンに和文表記が出来るのは Raspberry Pi にフォントソフトを追加して和文が使用できるようにしているためです。

以上