

アプリケーションノート

RaspberryPI と CPI-UR001 を接続する

目次

1. 同梱されているソフトウェアとご注意	1
1.1. 同梱されているソフトウェア	1
1.2. ご注意	1
2. はじめに	2
2.1. 用意したもの	2
2.2. 構成	2
3. 必要なソフトウェアのダウンロード	3
3.1. SD カードに書き込む OS”Raspbian”をダウンロードする	3
3.2. SD カードに OS を書き込むツールをダウンロードする	3
3.3. SD カードへの書き込み	3
3.4. /boot/config.txt の初期設定	5
4. Raspi-config による初期設定	6
4.1. “Expand Filesystem”の実行	6
4.2. ユーザパスワードの変更	7
4.3. GUI デスクトップの表示抑止	8
4.4. Internationalisation Options の設定	9
4.5. タイムゾーンの設定	9
4.6. キーボードの設定	11
4.7. 再起動	14
5. IP アドレスの固定化	15
5.1. Raspbian を更新	16
6. 日本語環境の設定	17
6.1. Raspi-config でロケール設定	17
6.2. 日本語フォントのインストール	18
6.3. コンソールでの日本語表示	18
7. Apache (WEB サーバー) のインストール	20
8. ftp サーバのインストール	21
8.1. 設定ファイルの変更	21
8.2. ftp サーバの自動起動登録	25
9. ソースプログラムを PC からファイル転送	26
9.1. バイナリファイルの転送	27
10. 検知放射線量をデータとして格納	28
10.1. Raspberry Pi と CPI-UR001 の通信仕様とユーザーズマニュアル	28
10.2. 検出音 ON/OFF 制御方法	29
10.3. 測定停止・再開制御方法	29
10.4. 検出音の ON/OFF と測定停止・再開について	29
10.5. CSV ヘッダーファイルのコピー	29
11. データ格納用ソースプログラム	30
11.1. コンパイル	30
11.2. 実行前の準備	30
11.3. 試験実行	31
12. ブラウザでデータをダウンロードできるようにする	32
12.1. Mkhtml の Crontab への登録	33
12.2. CGI の組み込み	34
12.3. ボタン表示更新のタイミング	35
13. Radmon_pi を/etc/initab へ登録する	36
14. おわりに	37

1. 同梱されているソフトウェアとご注意

1.1. 同梱されているソフトウェア

ファイル名	ファイル内容
Application_note_UR001_01.pdf	本説明書 システムの構築・手順について述べている
favicon.ico	ダミーファイル。ホームページのルートフォルダ("/home/pi/")に格納する。一般的にブラウザがページ表示する際に、このファイル名を探して、見つからない場合、"/var/log/apache2/error.log"が無駄に大きくなることを避ける為に設けた。
head	計測結果の CSV ファイル用ヘッダファイル。このヘッダにより、CPI 製 radmon ソフトウェアへロードすることで波形表示が可能となる。Apache2 のインストール後、"/var/www/radmon.data/cmd/"のフォルダに格納される。
index.html	ホームページのルートフォルダ("/home/pi/")に格納される。このファイルは/home/pi/におかれる"mkhtml"というシェルスクリプトによって、1 回/分の間隔にて自動更新される。
jump.html	更新された index.html を再表示させる時に
mkbody	パイプのファイル名を HTML 化する"mkhtml"のサブプログラムシェル。
mkhtml	"mkbody"の上位シェルとして index.html を"/home/pi/"に生成。
radmon_pi	放射線検知状況を CSV ファイルとして格納する実行形式のプログラム。
radmon_pi.c	"radmon_pi"のソースプログラム。 "gcc -o radmon_pi radmon_pi.c"にてコンパイルすると"radmon_pi"が生成される。
radmon_pi.sh	検知ソフトウェアを OS に登録する際に使用するシェルスクリプト
Soundsw.cgi	検知音 On/Off 切り替え用 CGI。検知データのホームページでの表示中、検知音の On/Off 制御するボタンのための CGI。 "/usr/lib/cgi-bin/"に格納される。
StartStopSW.cgi	計測の一時停止・再開用 CGI。検知データのホームページでの表示中、計測を一時停止・再開の制御ボタンのための CGI "/usr/lib/cgi-bin/"に格納される。

1.2. ご注意

本アプリケーションノートに記載されている内容を利用して生じたトラブルや損害について、株式会社シーピーアイテクノロジーズは一切の責任を負いかねますので、予めご了承いただいた上でご利用願います。また本内容についてのご質問は原則としてお答えいたしかねますので予めご了承願います。

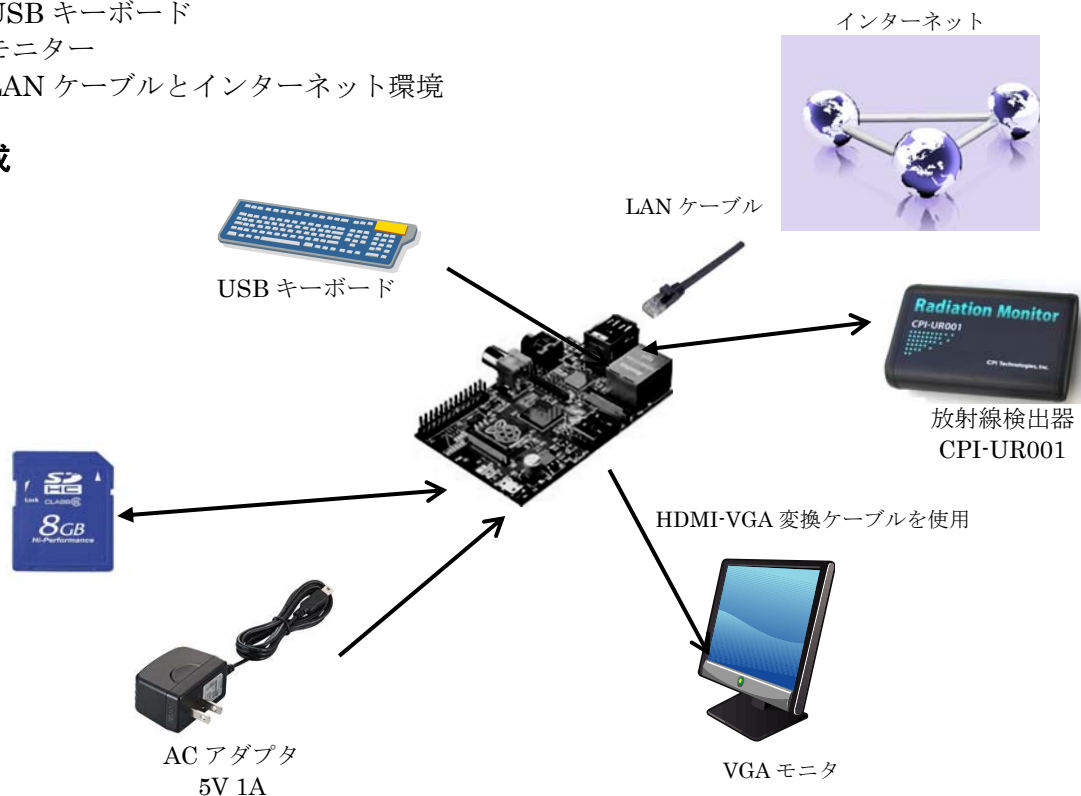
2. はじめに

安価に Linux サーバを構築できる Raspberry Pi にシーピーアイテクノロジー社の簡易放射線量検出器 CPI-UR001 を接続し、検知データサーバとして構築してみた。

2.1. 用意したもの

- Raspberry Pi
- 簡易放射線検出器 CPI-UR001 (シーピーアイテクノロジー社)
- 8GB SD メモリカード
- AC アダプタ (5V 0.7A 以上)
- USB キーボード
- モニター
- LAN ケーブルとインターネット環境

2.2. 構成



SD メモリカードには OS (Wheezy Raspbian) を書き込み、Raspberry Pi には、web サーバーと ftp をインストールし、検知データをネットワーク経由で PC にダウンロードできるようにする。

3. 必要なソフトウェアのダウンロード

「3. 必要なソフトウェアのダウンロード」は WindowsOS 下にて実行した例を示す。

3.1. SD カードに書き込む OS “Raspbian” をダウンロードする

「<http://www.raspberrypi.org/downloads>」の、Raspbian “wheezy” の項目にある、「yyyy-mm-dd-wheezy-raspbian.zip」をダウンロードし解凍する。

“yyyy-mm-dd”はバージョンを識別するための西暦の年月日のようで、今回ダウンロードしたのは、“2013-07-26”であった。約 2GB のファイルなので、完了までに時間を必要とする。

3.2. SD カードに OS を書き込むツールをダウンロードする

<http://sourceforge.net/projects/win32diskimager/>

より、「Win32DiskImager」をダウンロードする。

3.3. SD カードへの書き込み

図 1 に Win32 Disk Imager を実行したダイアログボックスを示す。

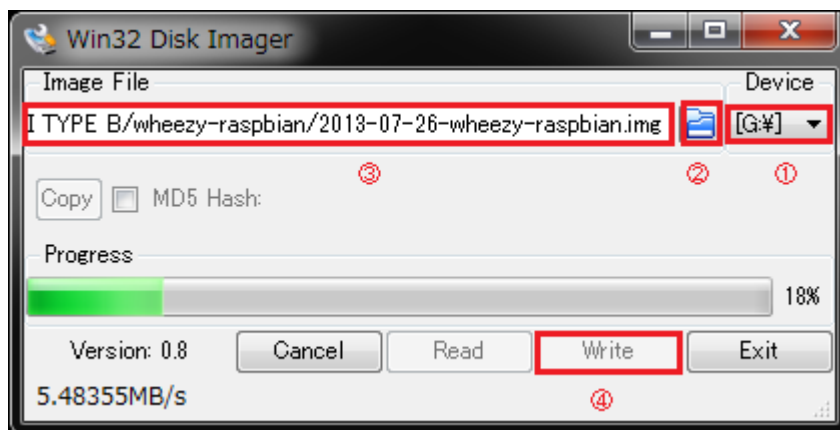


図 1

- ① PC に SD カードが装着されていれば、ドライブ名が表示されるので、書き込む SD カードのドライブかどうかを確認する。
- ② このフォルダのアイコンをクリックし、解凍した OS の img ファイルを指定する。
- ③ 指定した img ファイルのパスとファイル名が表示されるので、確認する。

- ④ 「Write」 ボタンをクリックすると図 2 のウィンドウが表示されるので書き込む SD カードのドライブ名が正しいことを確認してから「Yes」のボタンをクリックする。
- ⑤ 図 3 のウィンドウが表示されたら、正しく書き込まれたので、OK をクリックする。

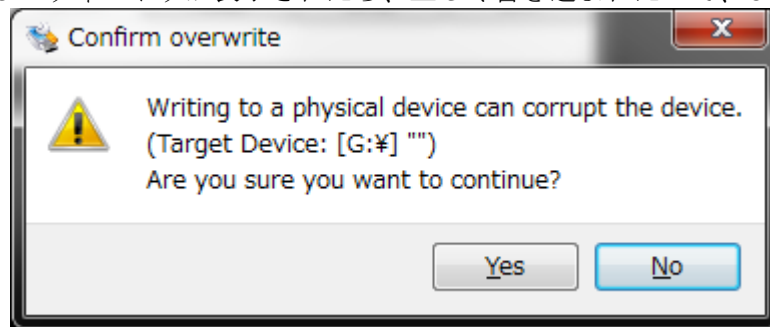


図 2

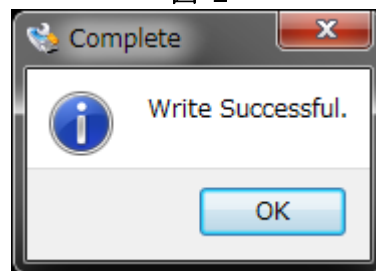


図 3

3.4. /boot/config.txt の初期設定

HDMI-VGA 変換ケーブルでモニタに接続したが、SD に書き込んだままの状態では画面が表示されない場合がある。その時は書き込んだ SD カードをエクスプローラで表示し

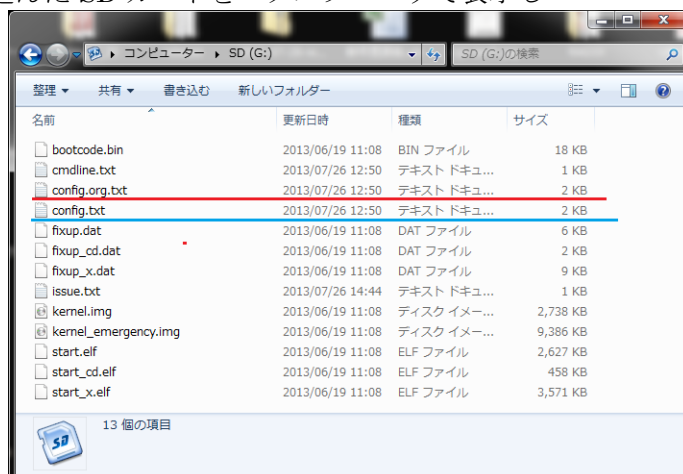


図 4

青い下線 config.txt”のバックアップを赤い下線の”config.org.txt”としてコピーアンドペーストと名前の変更などで作成する。

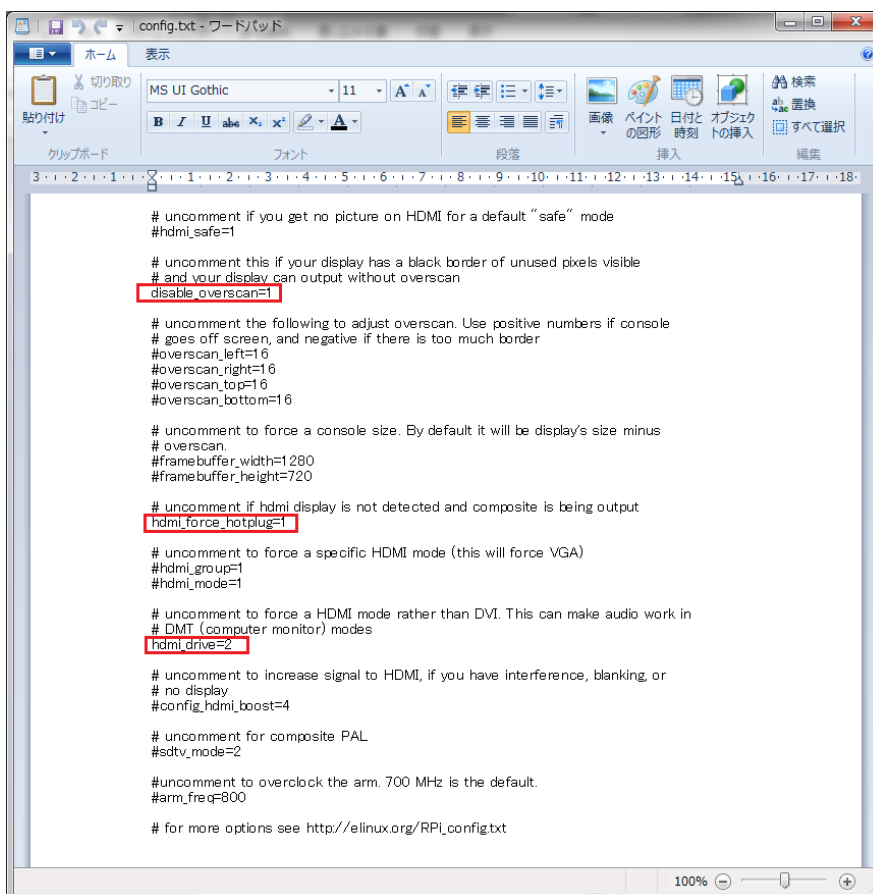


図 5

“config.txt”の内容を UNIX の改行コードをサポートしている適当なエディタによって図 5 に示す赤枠の部分を変更する。適当なエディタが無い場合は、ワードパッドで編集する。

ワードパッドは編集結果を UNIX 形式の改行コードで格納できないが、実際の動作には支障はなさそうだ。

1. “#disable_overscan=1”の先頭”#”を削除
2. “#hdmi_force_hotplug=1”の先頭”#”を削除
3. “#hdmi_drive=2”の先頭”#”を削除

使用するディスプレイによって、他の行のアンコメントや数値設定が必要な場合があるようなので、個別に確認のこと。

4. Raspi-config による初期設定

以上を設定し、SD カードを Raspberry Pi に実装後、電源投入するとしばらくして図 6 の画面が表示される。反転表示が現在選択されているメニューを示し、カーソルの上下キーで選択を移動し、タブキーでアクション (<Select>, <Finish>, <Ok>など) を選択し、Enter キーで実行する。

4.1. “Expand Filesystem”の実行

通常は約 2 GB のサイズとなってカードにコピーされるものが、8 GB や 16 GB のカードにコピーされた場合、その容量に合わせたサイズに領域を拡張する機能をもつ。

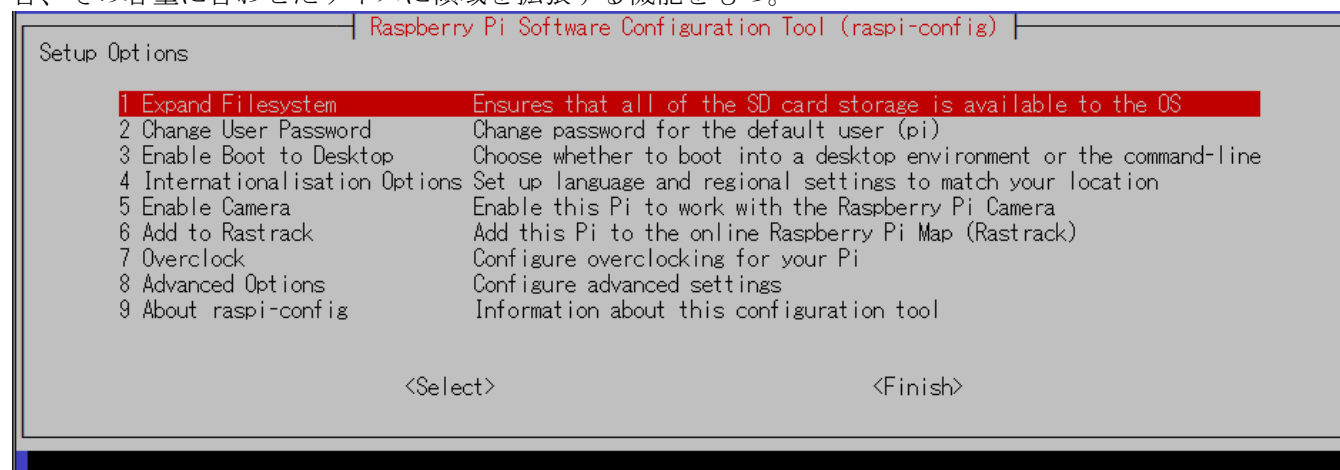


図 6

図 6 の状態から実行後、いくつかのメッセージを表示し、図 7 に示す画面が表示されるので、<Ok>で図 6 の画面に戻る。

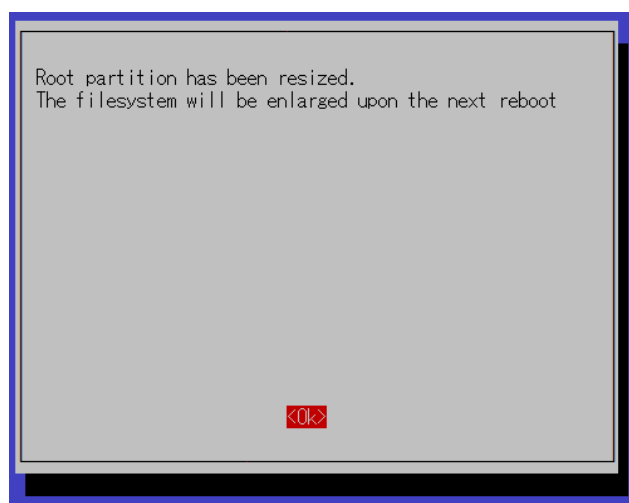


図 7

4.2. ユーザパスワードの変更

”2 Change User Password”を選択後、実行。

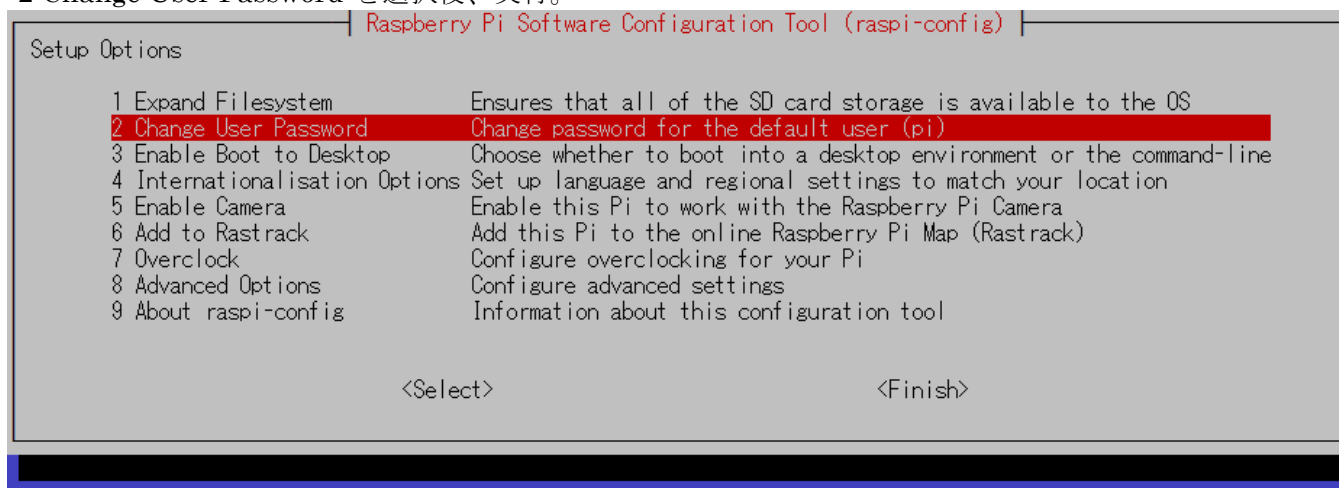


図 8

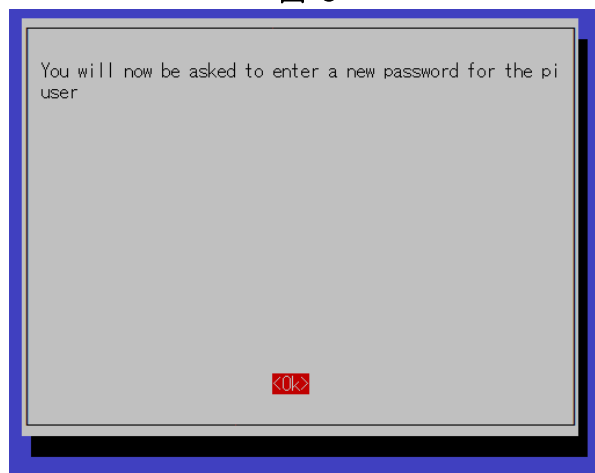


図 9

図 9 の画面が表示されたら<Ok>を実行。

新しいパスワードを2回入力すると、図 10 の画面が表示されるので実行する。

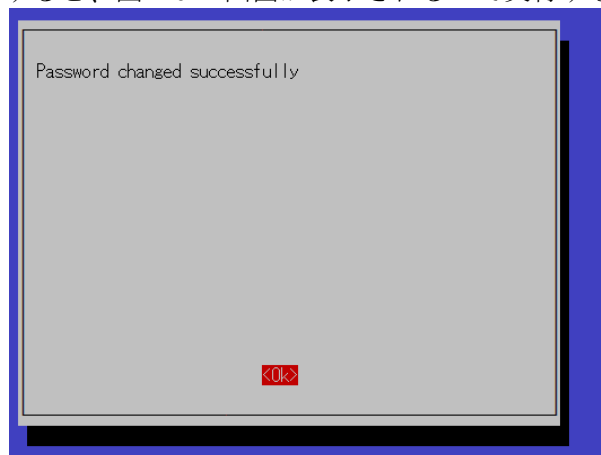


図 10

4.3. GUI デスクトップの表示抑止

GUI デスクトップの表示はこのアプリケーションでは使用しないので以下のようにする。

図 11 のように「3. Enable Boot to Desktop」を実行する。メッセージがいくつか表示され図 12 の画面が表示されたら「<No>」を選択し、実行する。

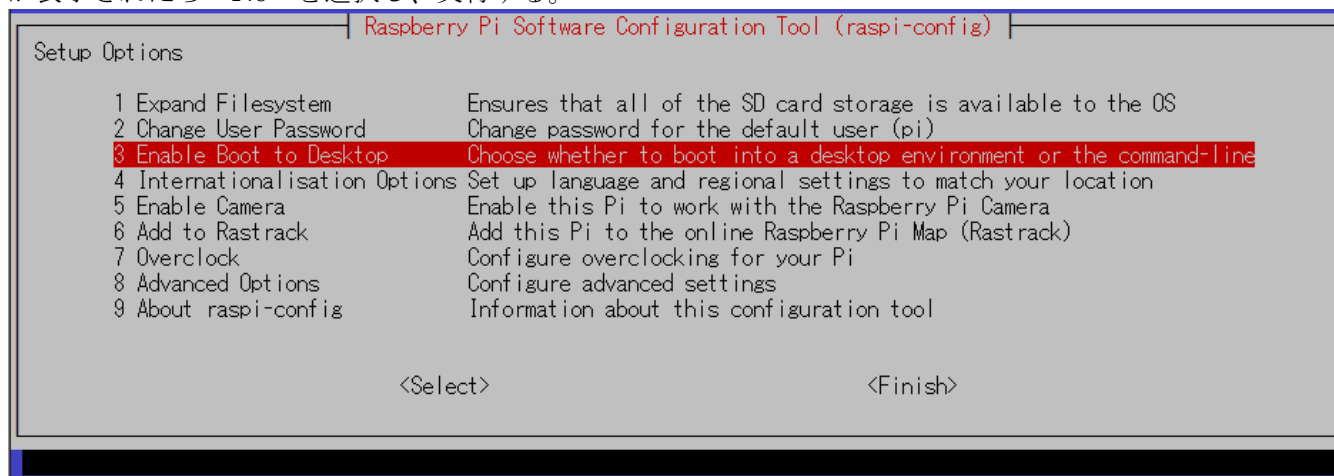


図 11

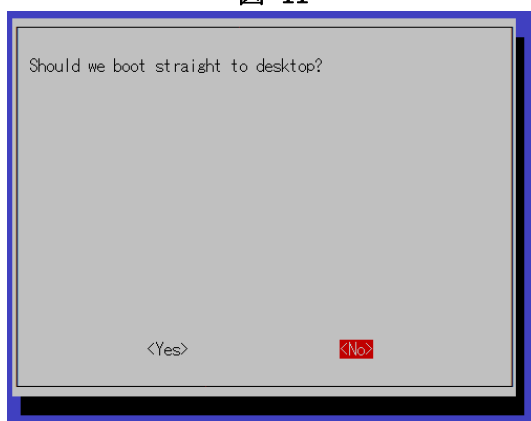


図 12

4.4. Internationalisation Options の設定

言語・時間関連設定です。

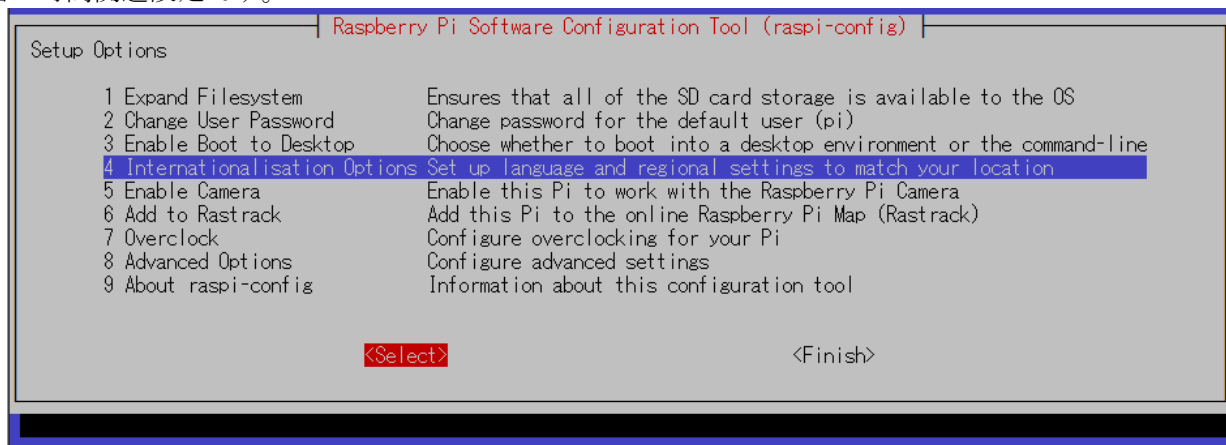


図 13

日本語サポートは、あとで設定するので、“I2 Chage Timezone”の設定を行う。

4.5. タイムゾーンの設定

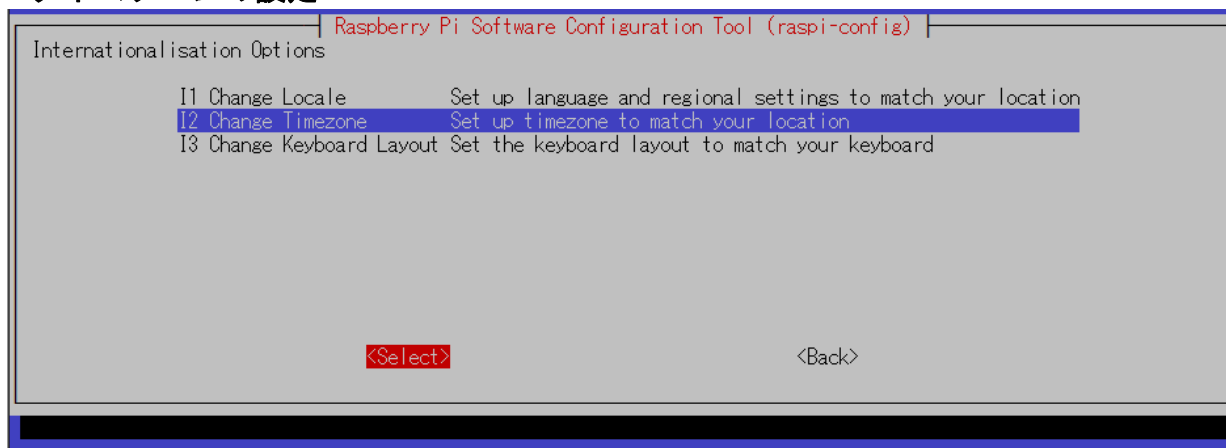


図 14

図 14 のように“I2 Change Timezone”を選択し実行する。



図 15

図 15 が表示されたら“Asia”を選択し、<Ok>で実行する。

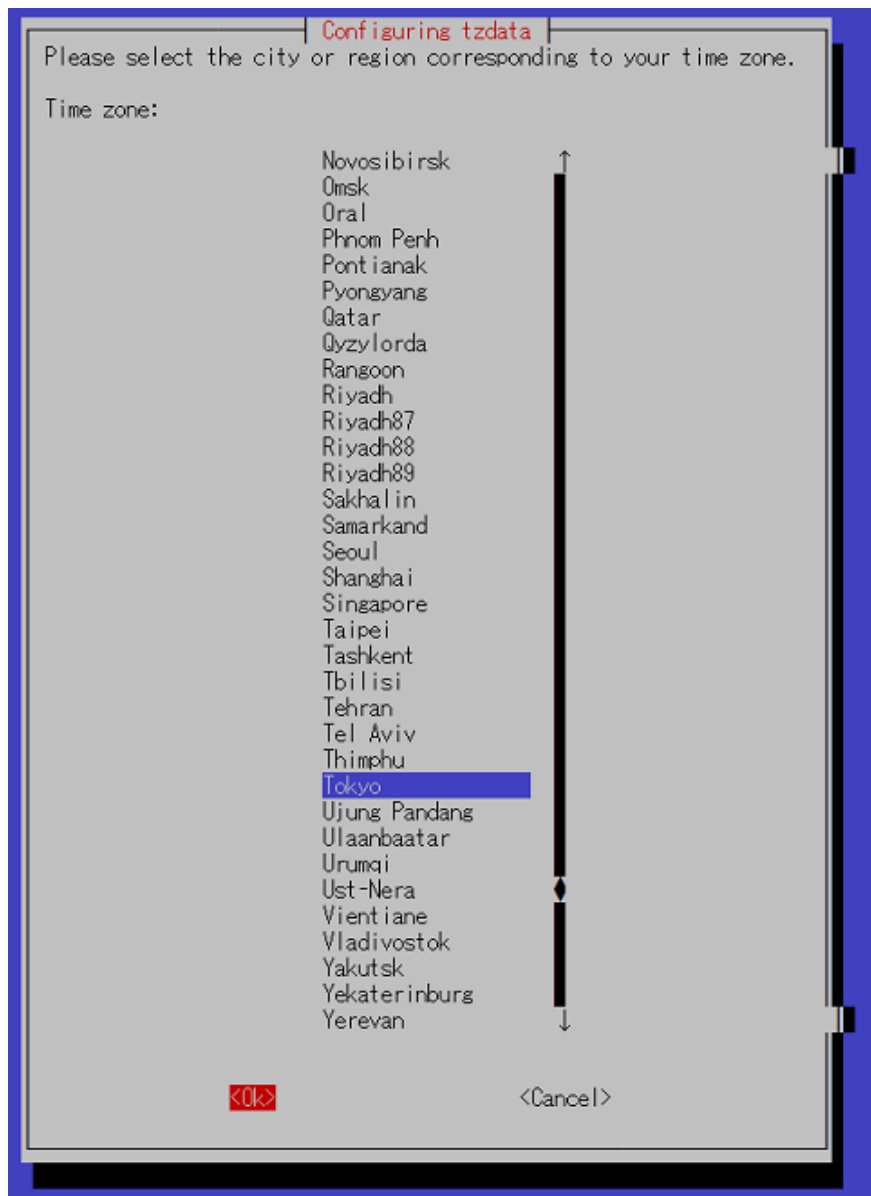


図 16

“Tokyo”を選択し、<Ok>で実行するとメッセージがいくつか表示され図 6 が表示され終了。

以上でタイムゾーンが東京に変更された。

4.6. キーボードの設定

図 13 の画面で実行により図 17 が表示される。

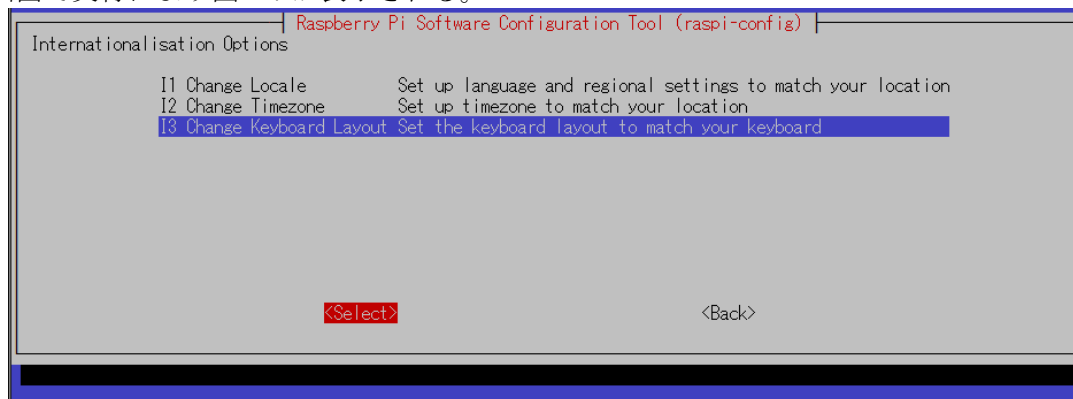


図 17

“I3 Change Keyboard Layout”を選択し実行。

しばらく待つと図 17 が表示されるので”Generic 105-Key (Intl) PC”を選択、実行。

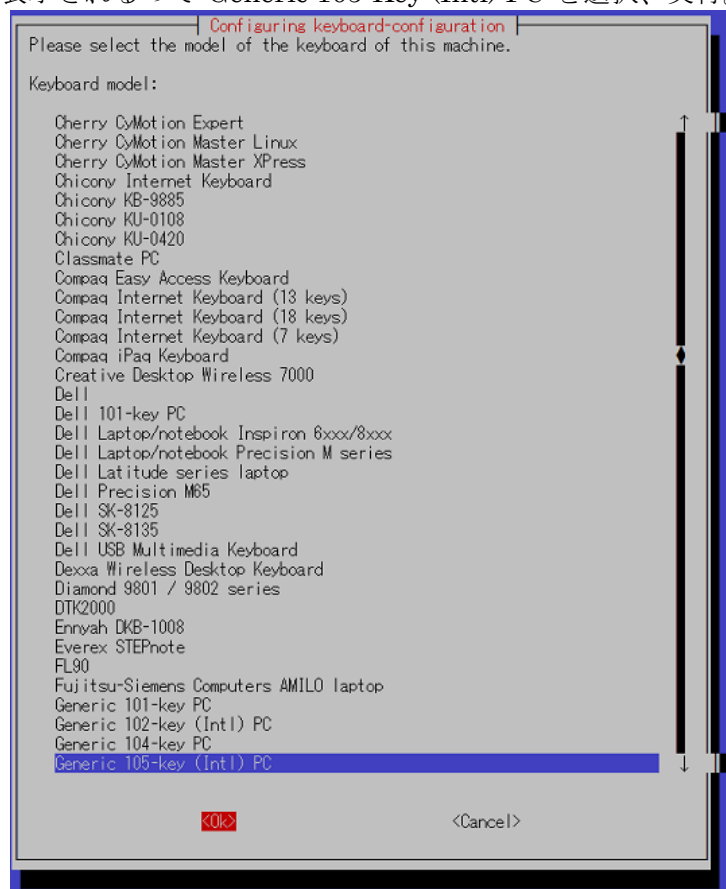


図 18

図 19 の画面で、“Other”を選択し<Ok>で実行。

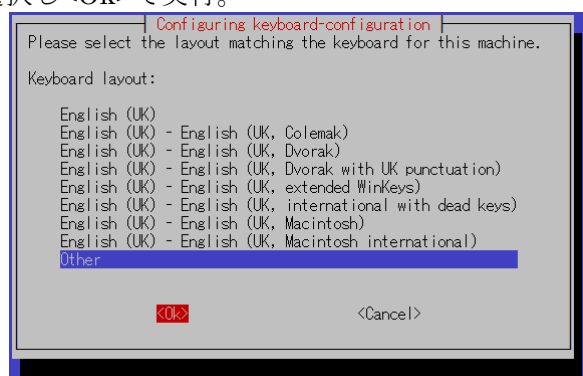


図 19

図 20 の画面で”Japanese”を選択し、<Ok>で実行。

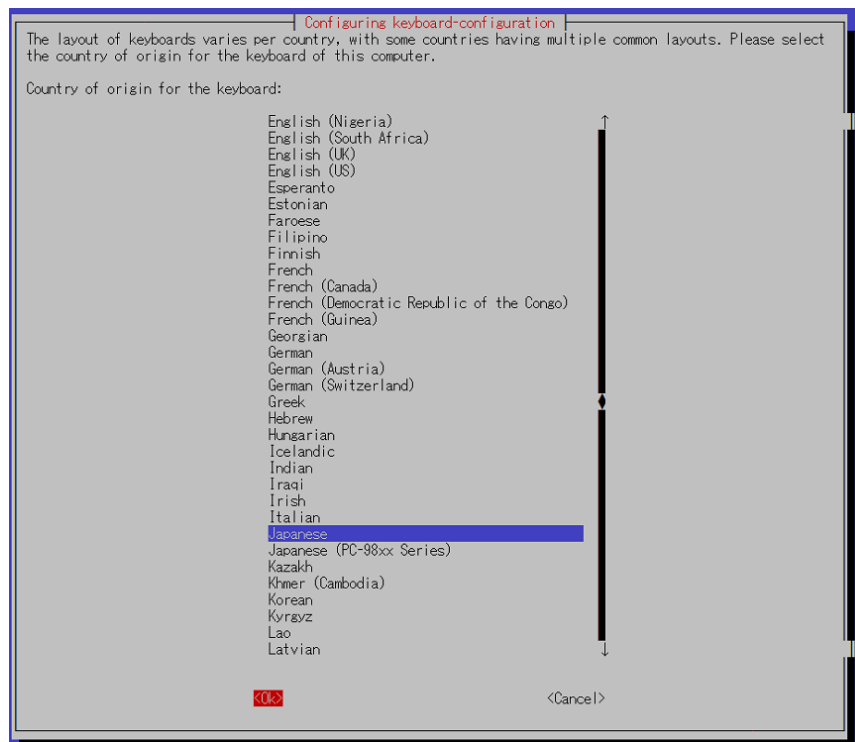


図 20

図 21 の画面で”Japanese”を選択し、<Ok>で Enter。



図 21

図 22 の画面で”The default for the keyboard layout”を選択し<Ok>で実行。



図 22

図 23 で”No compose key”を選択し、<Ok>で実行。



図 23

図 24 では<No>で実行。

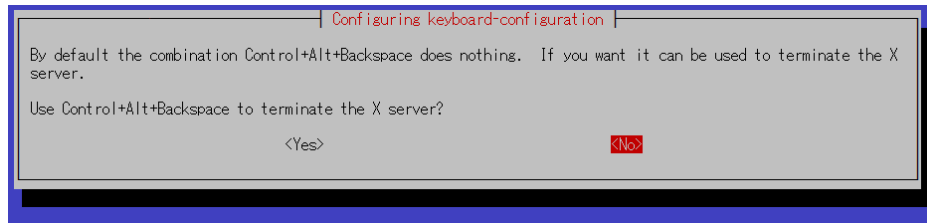


図 24

図 6 の画面が出てくるまでしばらく時間がかかる。
以上で”raspi-config”の必要最小限の設定が完了した。

4.7. 再起動

図 25 のように TAB で<Finish>を選択し実行。

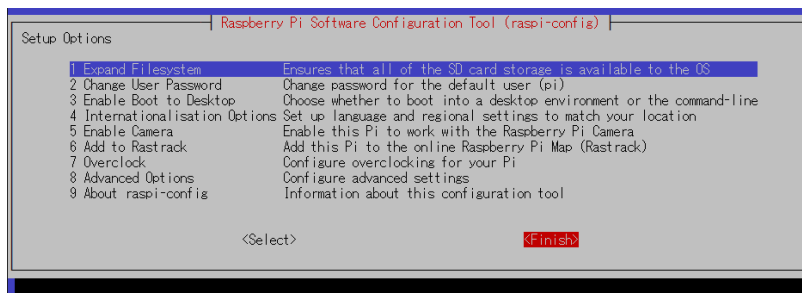


図 25

図 26 のようにリブートをするか聞いてくるので、<Yes>で実行。



図 26

リブート後は、ユーザ名を聞いてくるので、「pi」と入力し、「4.2 ユーザパスワードの変更」で設定したパスワードを入力してログインする。

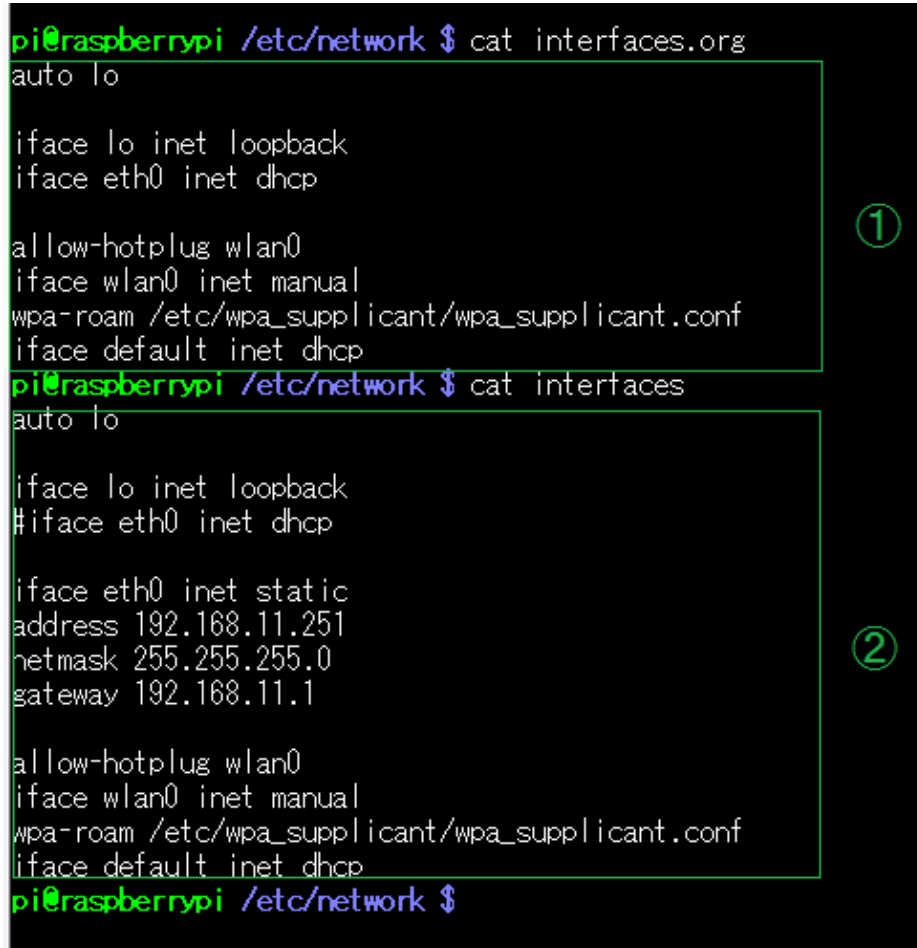
5. IP アドレスの固定化

通常は DHCP からの IP アドレスを利用するので、自身の IP アドレスを確認する場合は、`/sbin/ifconfig` で確認する。筆者はノート PC から LAN 経由で SSH を使って固定 IP アドレスでリモート接続したいのと、ホームページは固定アドレスで利用したいので IP アドレスを固定することにした。

コマンドラインで `sudo nano /etc/network/interfaces` を図 27 の①（オリジナル）を②に変更する。

実際のアドレス、マスク、ゲートウェイはそれぞれに合わせて変更が必要。

図 27 では固定アドレスを 192.168.11.251 に設定。



```
pi@raspberrypi /etc/network $ cat interfaces.org
auto lo

iface lo inet loopback
iface eth0 inet dhcp

allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp

pi@raspberrypi /etc/network $ cat interfaces
auto lo

iface lo inet loopback
#iface eth0 inet dhcp

iface eth0 inet static
address 192.168.11.251
netmask 255.255.255.0
gateway 192.168.11.1

allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp

pi@raspberrypi /etc/network $
```

図 27

```
pi@raspberrypi ~ $ sudo /etc/init.d/networking reload
```

で IP アドレスを更新。



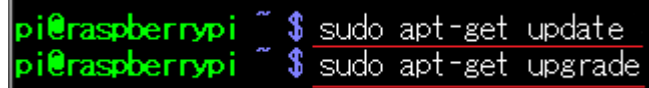
```
pi@raspberrypi ~ $ ifconfig
eth0      Link encap:イーサネット ハードウェアアドレス b8:27:eb:a4:6e:1c
          inetアドレス:192.168.11.251 ブロードキャスト:192.168.11.255 マスク:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  メトリック:1
          RXパケット:3538 エラー:0 損失:0 オーバラン:0 フレーム:0
          TXパケット:2040 エラー:0 損失:0 オーバラン:0 キャリア:0
          衝突(Collisions):0 TXキュー長:1000
          RXバイト:1742468 (1.6 MiB) TXバイト:344015 (335.9 KiB)

lo        Link encap:ローカルループバック
          inetアドレス:127.0.0.1 マスク:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  メトリック:1
          RXパケット:0 エラー:0 損失:0 オーバラン:0 フレーム:0
          TXパケット:0 エラー:0 損失:0 オーバラン:0 キャリア:0
          衝突(Collisions):0 TXキュー長:0
          RXバイト:0 (0.0 B) TXバイト:0 (0.0 B)
```

図 28

5.1. Raspbian を更新

Raspbian を最新版の OS に更新するために“\$sudo apt-get update”、“\$sudo apt-get upgrade”を実行



```
pi@raspberrypi ~ $ sudo apt-get update
pi@raspberrypi ~ $ sudo apt-get upgrade
```

図 29

図 29 の操作中に「続行しますか[Y/n]?」と尋ねてくる場合があるので、“Y”を入力する。

6. 日本語環境の設定

6.1. Raspi-config でロケール設定

“sudo raspi-config”の実行で図 30 の画面。”Internationalization Options”を選択し実行。

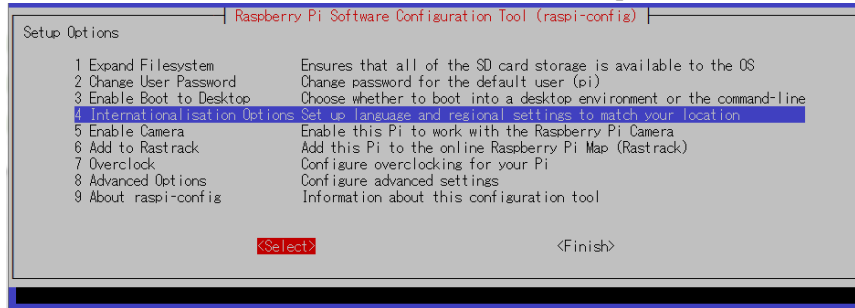


図 30

図 31 の画面で、”I1 Change Locale”を選択し実行。



図 31

上下のカーソルキーを使用し ja_JP.EUC-JP EUC-JP と ja_JP.UTF-8 UTF-8 を選択しスペースキーで”*”を表示させ、<Ok>で実行 (図 32)

既に”en_GB.UTF-8 UTF-8”は、選択されており、この選択はそのままにしておく。

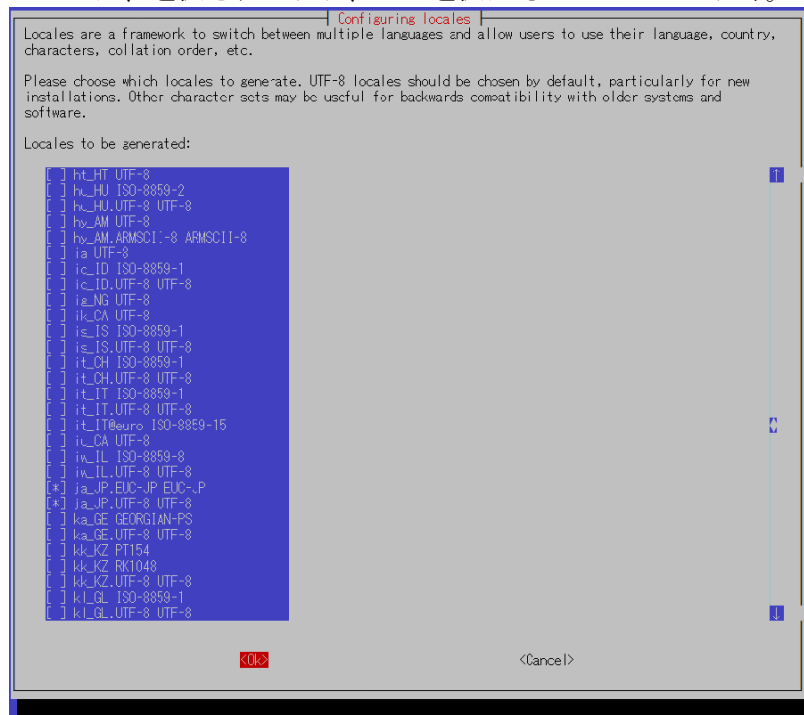


図 32

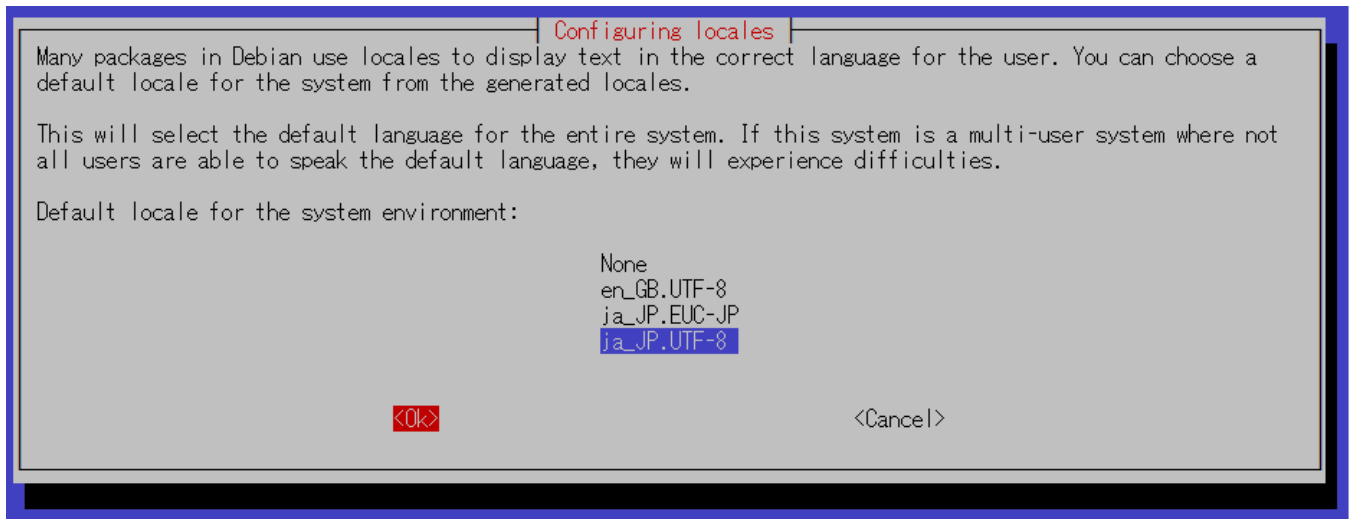


図 33

図 33 の画面が表示されるので、jp_JP.UTF-8 を選択後、実行。しばらく待った後、図 6 の画面が表示されたら、<Finish>を選択し実行。

6.2. 日本語フォントのインストール

```
pi@raspberrypi ~ $ sudo apt-get install ttf-kochi-gothic xfonts-intl-japanese xfonts-intl-japanese-big xfonts-kaname ibus-anthy
```

インストール中に「続行しますか[Y/n]?」と聞かれる場合があるので、「Y」で続行する。

以上で日本語インストールが終了。

6.3. コンソールでの日本語表示

以下を入力する。途中で「続行しますか[Y/n]?」と聞いてきた場合は「Y」で続行する。

```
pi@raspberrypi ~ $ sudo apt-get install jfbterm uim uim-anthy
```

終了したらインストール完了。全角モードと半角モードの切り替えはシフト+スペースとなっているが、コントロールキー+スペースに変更する場合はエディタで以下のように呼び出し

```
pi@raspberrypi ~ $ sudo nano /usr/share/uim/generic-key-custom.scm
```

図 34 赤の下線が”Shift”となっているので、“Control”に修正する

```
(define-custom 'generic-on-key '("zenkaku-hankaku" "Control" " )
  '(global-keys1)
  '(key)
  (N_ "[Global] on")
  (N_ "long description will be here"))

(define-custom 'generic-off-key '("zenkaku-hankaku" "Control" " )
  '(global-keys1)
  '(key)
  (N_ "[Global] off")
  (N_ "long description will be here"))
```

図 34

図 35 のように.bashrc”をエディタで編集。“~.bashrc”の末尾に jfbterm を自動起動するスクリプトを追加するために図 36 に示す赤枠を追加する。

```
pi@raspberrypi ~ $ sudo nano ~/.bashrc
```

図 35

```
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
    . /etc/bash_completion
fi

#
# enable Japanese terminal and FEP
#
case $TERM in
linux)
if [ -c /dev/fb0 ]; then
jfbterm -q -e uim-fep -u anthy
exit
fi
;;
esac
```

図 36

```
pi@raspberrypi ~ $ sudo shutdown -r now
```

で再起動する。

ログインするときのユーザ名は”pi”，パスワードは「4.2 ユーザパスワードの変更」にて設定したものを入力する。

コンソールでログインしたときに、画面左下に”anthy[An- R]” と表示されれば OK。

日本語入力に切替えたいときは<Control>キーとスペースキーを同時に押下、元に戻す時もおなじ方法で行う。

7. Apache (WEB サーバー) のインストール

図 37 の様に”sudo apt-get install apache2”と入力すると、「続行しますか?」と聞いてくるので”Y”で続行する。

```
pi@raspberrypi ~ $ sudo apt-get install apache2
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています
状態情報を読み取っています... 完了
以下の特別パッケージがインストールされます:
  apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap ssl-cert
提案パッケージ:
  apache2-doc apache2-suexec apache2-suexec-custom openssl-blacklist
以下のパッケージが新たにインストールされます:
  apache2 apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap ssl-cert
アップグレード: 0 個、新規インストール: 10 個、削除: 0 個、保留: 0 個。
1,350 kB のアーカイブを取得する必要があります。
この操作後に追加で 4,996 kB のディスク容量が消費されます。
続行しますか [Y/n]? Y
```

図 37

終了したら、”ps ax | grep apache”でプロセスが存在していることを確認する。(図 38)
実際の表示と異なる場合もあるかもしれないが、”/usr/sbin/apache2 -k start”が表示されていれば良い。

```
pi@raspberrypi ~ $ ps ax | grep apache
2942 ?        Ss        0:00 /usr/sbin/apache2 -k start
2945 ?        S         0:00 /usr/sbin/apache2 -k start
2947 ?        Sl        0:00 /usr/sbin/apache2 -k start
2948 ?        Sl        0:00 /usr/sbin/apache2 -k start
3055 pts/0    S+        0:00 grep --color=auto apache
```

図 38

更に、ホームページが表示されるかどうかを確認するために、Raspberry Pi に割り当てた IP アドレス(図 27)をブラウザに入力。(ここでは”192.168.11.251”)
図 39 は出力の例。

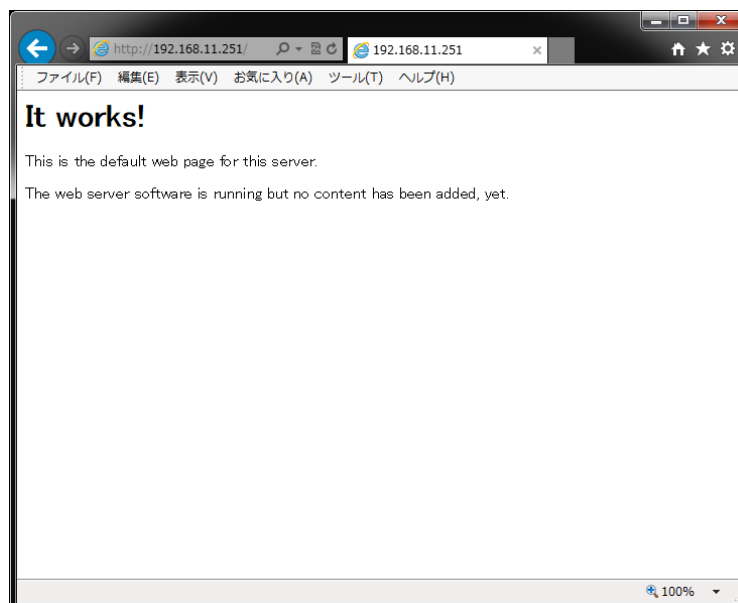


図 39

8. ftp サーバのインストール

(参考 URL : <http://assimane.blog.so-net.ne.jp/2013-04-14>)

図 29 に示す更新を定期的 (一週間に一度くらい) に実行していないのであれば、事前に実行する。

以下のコマンドライン入力で ftp サーバのインストールを行う。

```
pi@raspberrypi ~ $ sudo apt-get install vsftpd
```

インストールが終了したら、設定ファイルを変更する。

8.1. 設定ファイルの変更

```
pi@raspberrypi ~ $ sudo nano /etc/vsftpd.conf
```

にて”/etc/vsftpd.conf”を以下のように変更する。 変更は赤下線部の先頭の”#”を削除する。

```
# Example config file /etc/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=YES
#
# Run standalone with IPv6?
# Like the listen parameter, except vsftpd will listen on an IPv6 socket
# instead of an IPv4 one. This parameter and the listen parameter are mutually
# exclusive.
listen_ipv6=YES
#
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
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
#
# 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
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
anon_mkdir_write_enable=YES
```

図 40

```

#
# Activate directory messages - messages given to remote users when they
# go into a certain directory.
dirmessage_enable=YES
#
# If enabled, vsftpd will display directory listings with the time
# in your local time zone. The default is to display GMT. The
# times returned by the MDTM FTP command are also affected by this
# option.
use_localtime=YES
#
# Activate logging of uploads/downloads.
xferlog_enable=YES
#
# Make sure PORT transfer connections originate from port 20 (ftp-data).
connect_from_port_20=YES
#
# If you want, you can arrange for uploaded anonymous files to be owned by
# a different user. Note! Using "root" for uploaded files is not
# recommended!
chown_uploads=YES
chown_username=whoever
#
# You may override where the log file goes if you like. The default is shown
# below.
xferlog_file=/var/log/vsftpd.log
#
# If you want, you can have your log file in standard ftpd xferlog format.
# Note that the default log file location is /var/log/xferlog in this case.
xferlog_std_format=YES
#
# You may change the default value for timing out an idle session.
idle_session_timeout=600
#
# You may change the default value for timing out a data connection.
data_connection_timeout=120
#
# It is recommended that you define on your system a unique user which the
# ftp server can use as a totally isolated and unprivileged user.
nopriv_user=ftpsecure
#
# Enable this and the server will recognise asynchronous ABOR requests. Not
# recommended for security (the code is non-trivial). Not enabling it,
# however, may confuse older FTP clients.
async_abor_enable=YES
#
# By default the server will pretend to allow ASCII mode but in fact ignore
# the request. Turn on the below options to have the server actually do ASCII

```



```
# By default the server will pretend to allow ASCII mode but in fact ignore
# the request. Turn on the below options to have the server actually do ASCII
# mangling on files when in ASCII mode.
# Beware that on some FTP servers, ASCII support allows a denial of service
# attack (DoS) via the command "SIZE /big/file" in ASCII mode. vsftpd
# predicted this attack and has always been safe, reporting the size of the
# raw file.
# ASCII mangling is a horrible feature of the protocol.
ascii_upload_enable=YES
ascii_download_enable=YES
#
# You may fully customise the login banner string:
ftpd_banner=Welcome to my 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
#
# 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_user=YES
chroot_list_enable=YES
# (default follows)
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
# the presence of the "-R" option, so there is a strong case for enabling it.
ls_recurse_enable=YES
#
# Customization
#
# Some of vsftpd's settings don't fit the filesystem layout by
# default.
```

図 42

図 42 以降のテキストの変更はない。

“/etc/vsftpd.chroot_list”を設け、それにユーザ名“pi”を登録する。

```
pi@raspberrypi ~ $ sudo touch /etc/vsftpd.chroot_list
pi@raspberrypi ~ $ sudo nano /etc/vsftpd.chroot_list
```

図 43

で nano エディタで“pi”を入力し終了する。

終了後, cat コマンドで以下のように確認できれば OK.

```
pi@raspberrypi ~ $ cat /etc/vsftpd.chroot_list
pi
pi@raspberrypi ~ $
```

図 44

新しい設定でサーバーを再起動する。(図 45)

```
pi@raspberrypi ~ $ sudo service vsftpd restart
Stopping FTP server: vsftpd.
Starting FTP server: vsftpd.
pi@raspberrypi ~ $
```

図 45

8.2. ftp サーバの自動起動登録

/etc/init.d/vsftpd のバックアップファイルを作成し、パーミッションを変更する。

```
pi@raspberrypi ~ $ cd /etc/init.d
pi@raspberrypi /etc/init.d $ sudo cp -p vsftpd vsftpd.org
pi@raspberrypi /etc/init.d $ sudo chmod 644 vsftpd.org
```

図 46

エディタで vsftpd の先頭部分の赤線で示した行に変更する。オリジナルは”1”のみ記入されている。“sudo update-rc.d vsftpd remove”で一旦削除した後、“sudo update-rc.d vsftpd defaults”で登録し、再起動する。

```
pi@raspberrypi /etc/init.d $ sudo nano vsftpd
pi@raspberrypi /etc/init.d $ head vsftpd
#!/bin/sh

### BEGIN INIT INFO
# Provides:          vsftpd
# Required-Start:    $remote_fs $syslog
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Very secure FTP server
### END INIT INFO

pi@raspberrypi /etc/init.d $ sudo update-rc.d vsftpd remove
update-rc.d: using dependency based boot sequencing
pi@raspberrypi /etc/init.d $ sudo update-rc.d vsftpd defaults
update-rc.d: using dependency based boot sequencing
pi@raspberrypi /etc/init.d $ sudo shutdown -r now
```

図 47

9. ソースプログラムを PC からファイル転送

実際の制御プログラムを作成する場合、使い慣れた PC のエディタを使用の方が容易なので、ソースプログラムやテキストを WindowsPC から Raspberry Pi に転送してみる。

Raspberry Pi での漢字コードは UTF-8 で、PC の SJIS とは異なるため、WindowsPC 側では FTP の転送機能にコード変換機能を持つ、FFFTP を使用した。コード変換機能を持たない FTP を使用する場合は、転送後に UTF にコード変換が必要となる。

プログラムやテキストは付属の Sources フォルダに入っているなので、そのフォルダの中身で青色反転されている内容を、Raspberry Pi の "/home/pi" に転送する。

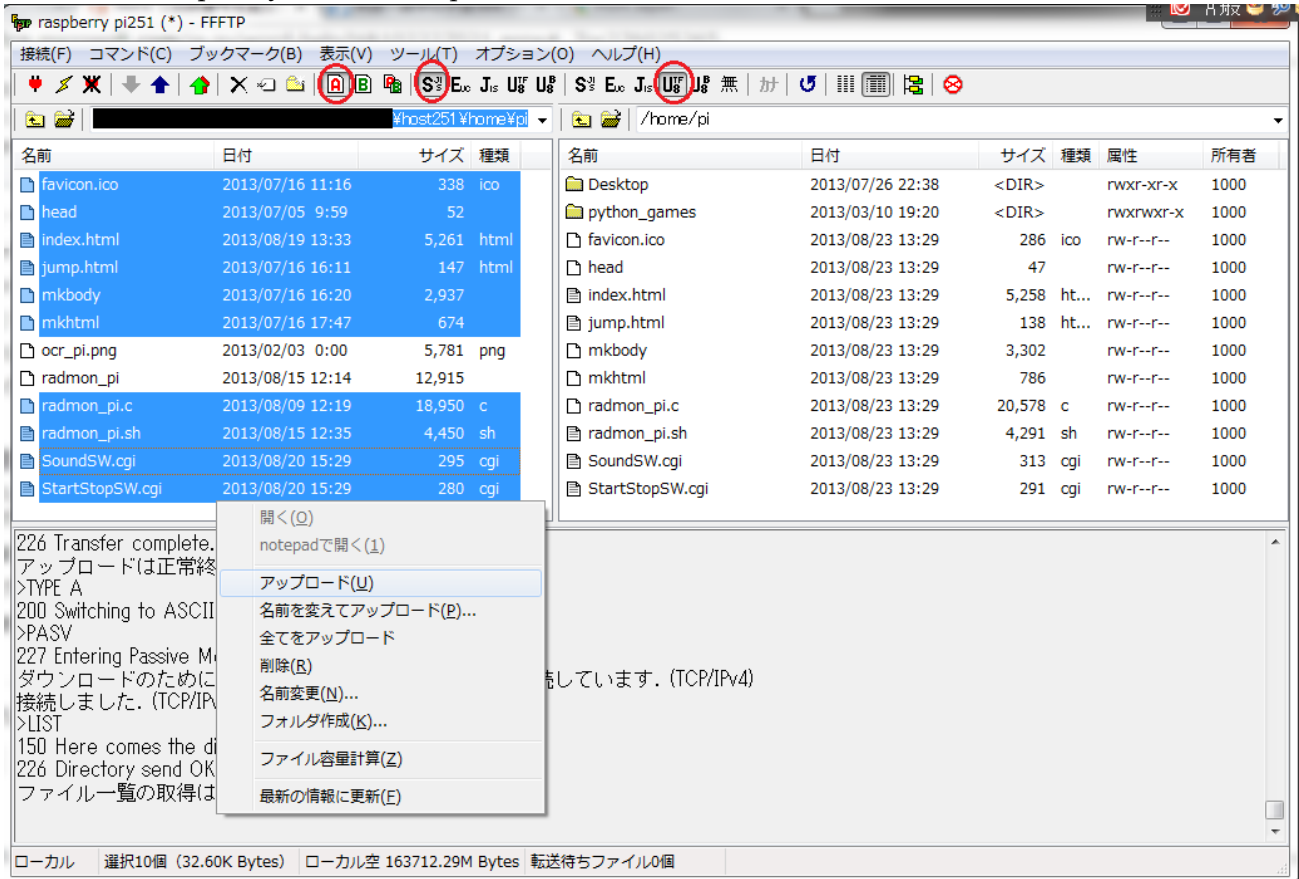


図 48

ソースプログラムの転送なのでツールバーの



のように、"A"を選択し、ローカル側 SJIS とホスト側 UTF8 を選択する。
この設定でソースプログラムを転送すれば、自動的に UTF-8 に変換する。

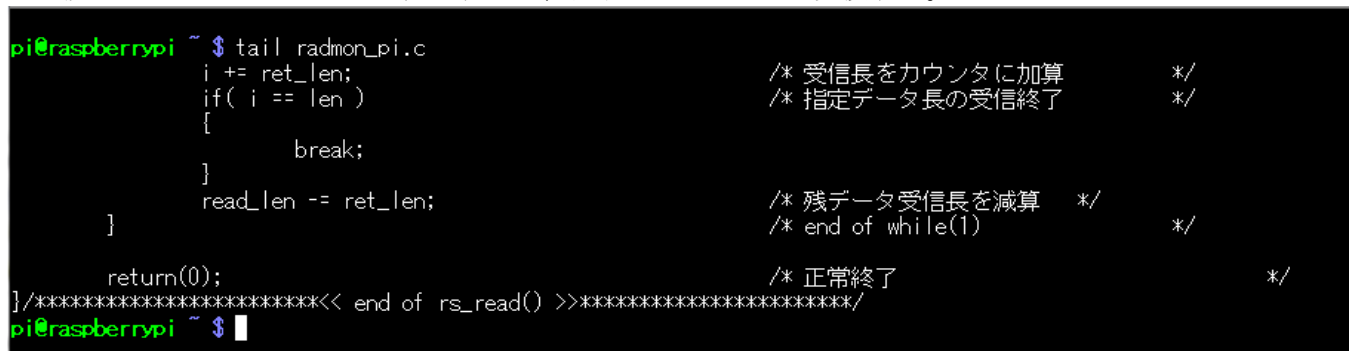


図 49 に示すように more, head, tail などのコマンドで転送したファイルを表示し、漢字が正常に表示されたら、正しく転送終了。

9.1. バイナリファイルの転送

続いて図 50 に示す青反転のファイルをバイナリ転送する。

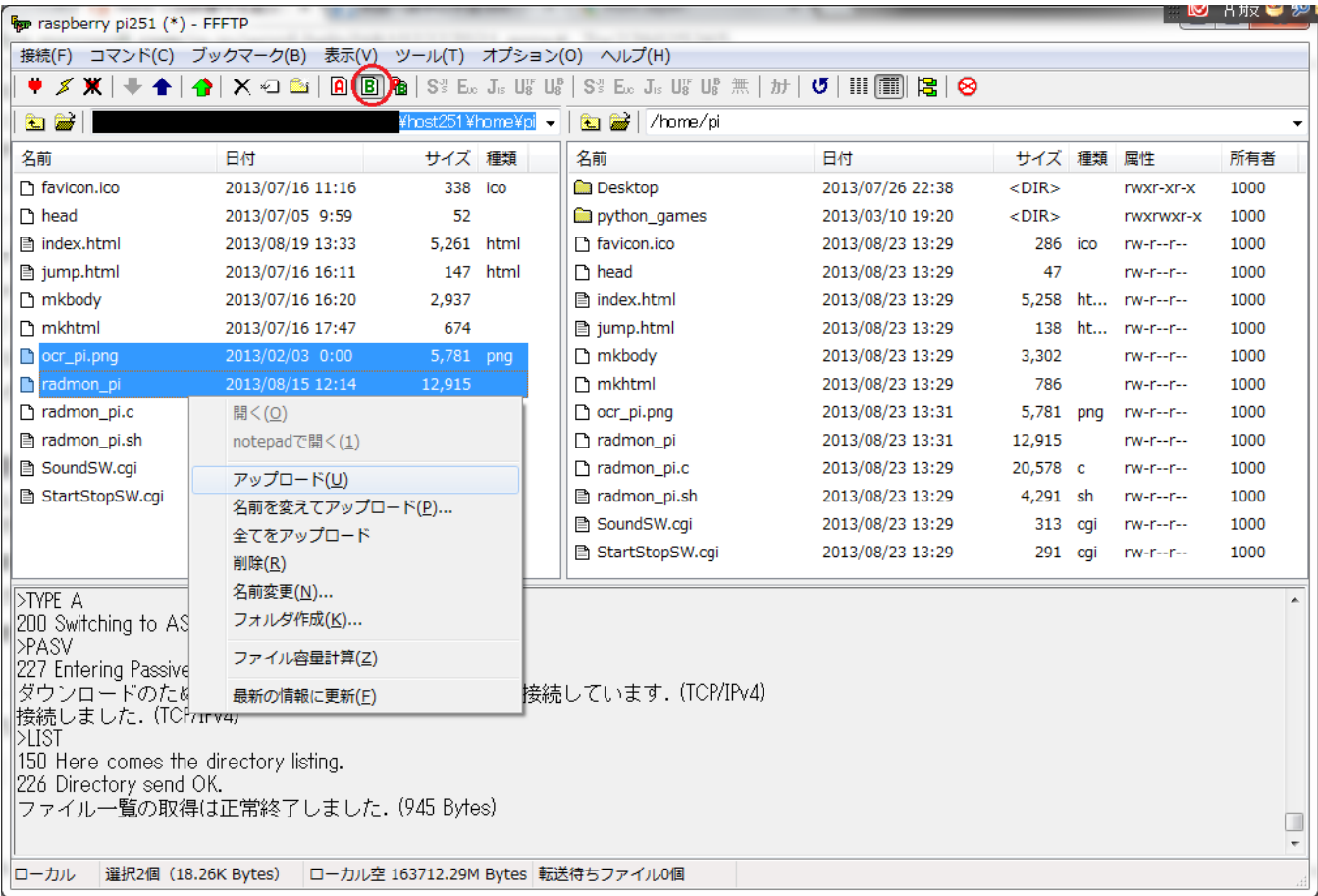


図 50

10. 検知放射線量をデータとして格納

検知した放射線量を時刻データとともにファイルに格納するプログラムが、“radmon_pi.c”にあり、ftp で PC から Raspberry Pi に転送し、コンパイルする。このプログラムの主な仕様は

1. 検知データを時刻とともに“/var/www/radmon.data/”フォルダに一日ごとに CSV 形式で格納する。
2. “/var/www/radmon.data/cmd/”フォルダに“OFF.cmd”というファイル名がある場合は、検知音を OFF し、そのファイル名が無い場合は検知音を ON とする。
ブラウザ表示から CGI にて ON・OFF を制御する
3. “/var/www/radmon.data/cmd/”フォルダに“STOP.cmd”というファイル名がある場合は、データの格納を一旦停止する。“STOP.cmd”というファイル名が無い場合は、データの格納を再開する。
ブラウザ表示から CGI にて ON・OFF を制御する
4. “/var/www/radmon.data/cmd/”フォルダに head というファイルを設けておき、出力ファイルのヘッダーファイルとして使用する。
既定ファイルを当該フォルダに予め格納しておく。
5. 3 の一旦停止からの再開、電源再投入、shutdown, reboot. など検知プログラムが一旦停止し、再開した場合は、再開の日時から新規のファイルを作成して CSV データを格納する。
6. サンプルしたデータのグラフ表示は、製品添付の Windows 版 RADMON を利用できるようにする。

となっている。

10.1. Raspberry Pi と CPI-UR001 の通信仕様とユーザーズマニュアル

CPI-UR001 の通信仕様は

http://www.cpi-tec.jp/cpi-ur001/dl/com_spec.pdf

からダウンロードできる。

また、ユーザーズマニュアルは

http://cpi-tec.jp/cpi-ur001/dl/CPI-UR001_MAN_100.pdf

からダウンロードできる。

10.2. 検出音 ON/OFF 制御方法

“/var/www/radmon.data/cmd/”に”OFF.cmd”ファイルの有無で制御する	
“OFF.cmd”が有る	検出音を OFF
“OFF.cmd”が無い	検出音を ON

10.3. 測定停止・再開制御方法

“/var/www/radmon.data/cmd/”に”STOP.cmd”ファイルの有無で制御する	
“STOP.cmd”が有る	測定記録動作を一時停止
“STOP.cmd”が無い	測定記録動作を再開

10.4. 検出音の ON/OFF と測定停止・再開について

検出音の ON/OFF と測定停止・再開は WEB 表示している状態でスイッチでトグルするように作成するものとする。

10.5. CSV ヘッダーファイルのコピー

検知データが格納される CSV ファイルには、図 48 で転送した”head”を
“/var/www/radmon.data/”にコピーする。 コピーは図 51 に示す手順で行う。

```
pi@raspberrypi ~ $ ls -l /var/www ←“/var/www/”に“index.html”だけがあることを確認
合計 4
-rw-r--r-- 1 root root 177  9月 13 17:22 index.html
pi@raspberrypi ~ $ sudo mkdir /var/www/radmon.data ←radmon.dataフォルダの作成
pi@raspberrypi ~ $ sudo mkdir /var/www/radmon.data/cmd ←cmdフォルダの作成
pi@raspberrypi ~ $ ls -l /var/www ←radmon.dataフォルダが作成できているかを確認
合計 8
-rw-r--r-- 1 root root 177  9月 13 17:22 index.html
drwxr-xr-x 3 root root 4096  9月 19 16:35 radmon.data
pi@raspberrypi ~ $ ls -l /var/www/radmon.data/ ←cmdフォルダが作成できているかを確認
合計 4
drwxr-xr-x 2 root root 4096  9月 19 16:35 cmd
pi@raspberrypi ~ $ ls -l /var/www/radmon.data/cmd/ ←cmdフォルダに何も無いことを確認
合計 0
pi@raspberrypi ~ $ sudo cp ~/head /var/www/radmon.data/cmd/ ←headをcmdフォルダにコピー
pi@raspberrypi ~ $ ls -l /var/www/radmon.data/cmd/ ←headがコピーされていることを確認
合計 4
-rw-r--r-- 1 root root 47  9月 19 16:38 head
pi@raspberrypi ~ $
```

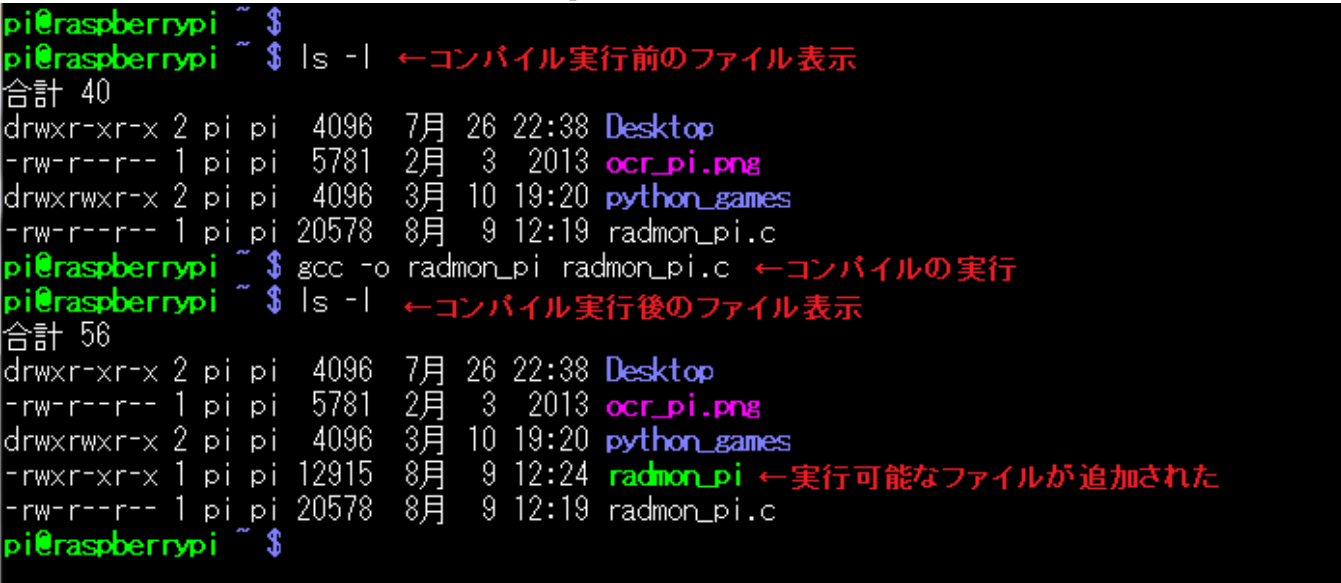
図 51

11. データ格納用ソースプログラム

ソースプログラムは別途添付のとおりなので直接入力するか図 48に示すコード変換付き ftp にて転送する。

11.1. コンパイル

図 52 に示すようにコンパイルし、"radmon_pi"という実行可能なファイルが追加されたことを確認する。

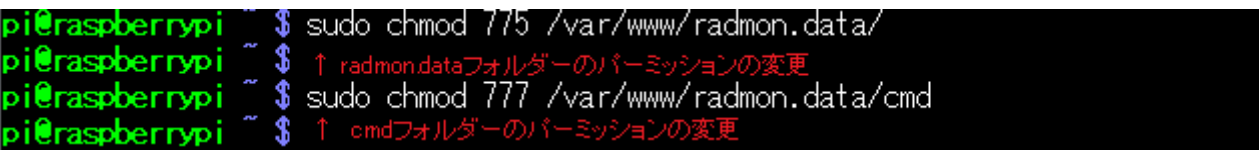


```
pi@raspberrypi ~$  
pi@raspberrypi ~$ ls -l ←コンパイル実行前のファイル表示  
合計 40  
drwxr-xr-x 2 pi pi 4096 7月 26 22:38 Desktop  
-rw-r--r-- 1 pi pi 5781 2月 3 2013 ocr_pi.png  
drwxrwxr-x 2 pi pi 4096 3月 10 19:20 python_games  
-rw-r--r-- 1 pi pi 20578 8月 9 12:19 radmon_pi.c  
pi@raspberrypi ~$ gcc -o radmon_pi radmon_pi.c ←コンパイルの実行  
pi@raspberrypi ~$ ls -l ←コンパイル実行後のファイル表示  
合計 56  
drwxr-xr-x 2 pi pi 4096 7月 26 22:38 Desktop  
-rw-r--r-- 1 pi pi 5781 2月 3 2013 ocr_pi.png  
drwxrwxr-x 2 pi pi 4096 3月 10 19:20 python_games  
-rwxr-xr-x 1 pi pi 12915 8月 9 12:24 radmon_pi ←実行可能なファイルが追加された  
-rw-r--r-- 1 pi pi 20578 8月 9 12:19 radmon_pi.c  
pi@raspberrypi ~$
```

図 52

11.2. 実行前の準備

実行前に必要なディレクトリ・パーミッションの設定を行います。



```
pi@raspberrypi ~$ sudo chmod 775 /var/www/radmon.data/  
pi@raspberrypi ~$ ↑ radmon.dataフォルダーのパーミッションの変更  
pi@raspberrypi ~$ sudo chmod 777 /var/www/radmon.data/cmd  
pi@raspberrypi ~$ ↑ cmdフォルダーのパーミッションの変更
```

図 53

11.3. 試験実行

実行できるかどうかを”sudo ./radmon_pi<Enter>”の入力で確認する。

実行後、”/var/www/radmon.data’を作成できません： ファイルが存在します”や”/var/www/radmon.data/cmd’を作成できません： ファイルが存在します”と表示されるのはエラーではない。

```
pi@raspberrypi ~ $ ls -l /var/www/radmon.data/ ←データ格納フォルダが空であることを確認
合計 4
drwxrwxrwx 2 root root 4096  9月 19 16:38
pi@raspberrypi ~ $ sudo ./radmon_pi ←手動で起動
mkdir: ディレクトリ '/var/www/radmon.data' を作成できません: ファイルが存在します ←エラーではなく、既に存在するという情報
mkdir: ディレクトリ '/var/www/radmon.data/cmd' を作成できません: ファイルが存在します ←エラーではなく、既に存在するという情報
Sampling Start ←サンプリングの開始
Confirmed Sampling Start acknowledge ←機器側からの応答を確認したというメッセージ
^Cpi@raspberrypi ~ $ ls -l /var/www/radmon.data/ ←しばらく、この行でカーソルが点滅しているのでCtrl+Cで実行を停止しデータ格納フォルダに
合計 8                                新しいファイルが作成されたことを確認する
-rw-r--r-- 1 root root  47  9月 19 17:15 130919-171548.csv ←新しいファイルの作成が確認された。
drwxrwxrwx 2 root root 4096  9月 19 16:38
pi@raspberrypi ~ $ sudo rm /var/www/radmon.data/130919-171548.csv ←試験実行で作成されたファイルなので削除する
pi@raspberrypi ~ $ ls -l /var/www/radmon.data/ ←削除されたことを確認する
合計 4
drwxrwxrwx 2 root root 4096  9月 19 16:38
pi@raspberrypi ~ $
```

図 54

<Ctl>+”C”キーの入力で途中終了できる。

このプログラムが実行されると/var/www/radmon.data/に csv ファイルが作成され、1 分ごとにその間のデータが書き込まれる。

(注)

図 54 において、”sudo ./radmon_pi”の手動実行後の「・・・ファイルが存在します」は、OS が日本語化されていない OS では、英文表示となる場合がある。

12. ブラウザでデータをダウンロードできるようにする

Apache のホームディレクトリの”/var/www/”に、ファイル名 index.html を定期的に生成する仕掛けの作成。

```
pi@raspberrypi ~ $ ls -l /var/www ←ホームディレクトリに何も無いことを確認
合計 4
drwxrwxr-x 3 root root 4096  8月 19 14:31 radmon.data
pi@raspberrypi ~ $ chmod 755 mkbody ←パーミッションの変更
pi@raspberrypi ~ $ chmod 755 mkhtml ←パーミッションの変更
pi@raspberrypi ~ $ sudo ./mkhtml ←mkhtmlを実行してみる
pi@raspberrypi ~ $ ls -l /var/www ←ホームディレクトリを再度確認
合計 8
-rw-r--r-- 1 root root 937  8月 19 14:52 index.html ←index.htmlが増えている
drwxrwxr-x 3 root root 4096  8月 19 14:31 radmon.data
pi@raspberrypi ~ $
```

図 55

図 55 の手順で、mkbody, mkhtml のパーミッションを変更し、プログラムが実行できるかを確認。

```
pi@raspberrypi ~ $ sudo touch /var/www/radmon.data/cmd/dir.txt
pi@raspberrypi ~ $ sudo chmod 666 /var/www/radmon.data/cmd/dir.txt
```

図 56

Mkhtml はデータを格納しているファイル名を”/var/www/radmon.data/cmd/dir.txt”に格納するため図 56 に示す手順でファイル名を格納用ファイルを設定する。

/var/www/inde.html が作成されており、ブラウザにて「<http://192.168.11.251/>」（アドレスは、Raspberry Pi に割り当てられた I P アドレス）を入力し図 57 の表示が確認できれば良い。表示される C S V ファイルは 1 つ以上あれば良い。

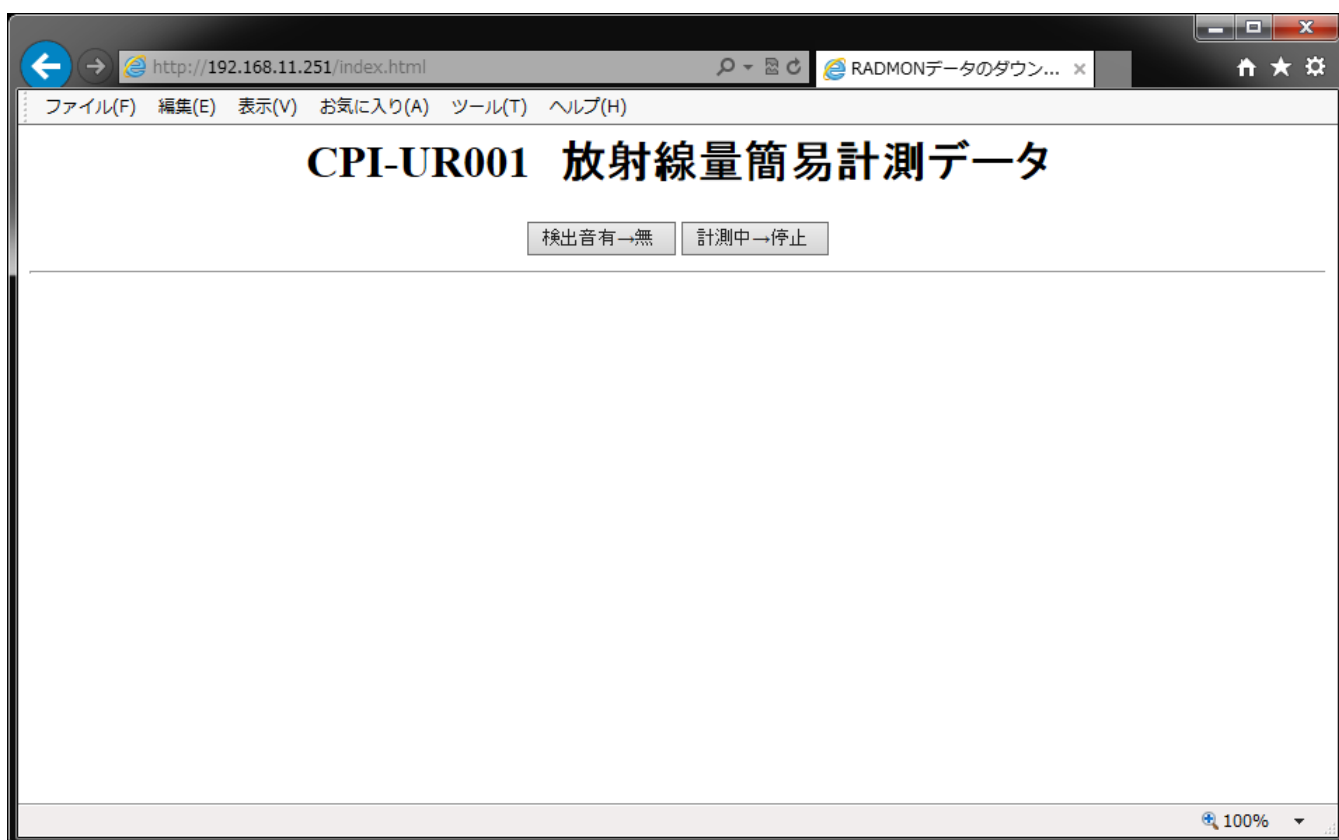


図 57

12.1. Mkhtml の Crontab への登録

図 58 に示すように index.html のパーミッションを変更する。

```
pi@raspberrypi ~ $ ls -l /var/www/index.html
-rw-r--r-- 1 root root 954 10月  4 19:11 /var/www/index.html
pi@raspberrypi ~ $ sudo chmod 666 /var/www/index.html
pi@raspberrypi ~ $ ls -l /var/www/index.html
-rw-rw-rw- 1 root root 954 10月  4 19:17 /var/www/index.html
```

図 58

“mkhtml”は、crontab に登録し、1 分ごとに index.html を更新する設定を行う。
まず、CRONTAB の編集を行う。

```
pi@raspberrypi ~ $ sudo crontab -e
```

と入力すると図 59 の赤枠を除いた表示が出るので、赤枠を追加する。

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
* * * * * /home/pi/mkhtml
```

図 59

“crontab”の編集モードから戻った時の画面を図 60 に示す。

```
pi@raspberrypi ~ $ sudo crontab -e ←crontabを編集する
no crontab for root - using an empty one
crontab: installing new crontab ←編集を終了すると左のメッセージが出力される
pi@raspberrypi ~ $
```

図 60

12.2. CGI の組み込み

「12.1 Mkhtml の Crontab への登録」にて登録した mkhtml は毎分起動され、ホームディレクトリである”/var/www/”に index.html を生成する。この html を「http://192.168.11.251」とアドレスを入力してブラウザで表示すると、上部に「検出音有→無」と「計測中→停止」が表示される。このボタンは、CPI-UR001 の検知音のオン・オフや、計測を一時停止・再開する制御用のボタンである。以下に、このボタンは CGI で動作するようになっており、以下にその組み込み手順を示す。

```
pi@raspberrypi ~ $ ls -l *.cgi ←ftpで転送したファイルタイプcgiを表示
-rw-r--r-- 1 pi pi 313  8月 20 15:29 SoundSW.cgi
-rw-r--r-- 1 pi pi 291  8月 20 15:29 StartStopSW.cgi
pi@raspberrypi ~ $ ls -l /usr/lib/cgi-bin/ ←コピー先に何も無いことを確認
合計 0
pi@raspberrypi ~ $ sudo cp *.cgi /usr/lib/cgi-bin/ ←/usr/lib/cgi-bin/にファイルタイプcgiをコピー
pi@raspberrypi ~ $ ls -l /usr/lib/cgi-bin/
合計 8
-rw-r--r-- 1 root root 313  8月 20 16:04 SoundSW.cgi
-rw-r--r-- 1 root root 291  8月 20 16:04 StartStopSW.cgi
pi@raspberrypi ~ $ cd /usr/lib/cgi-bin/ ←コピー先にディレクトリを変更
pi@raspberrypi /usr/lib/cgi-bin $ sudo chmod 755 SoundSW.cgi ←パーミッションを変更
pi@raspberrypi /usr/lib/cgi-bin $ sudo chmod 755 StartStopSW.cgi ←パーミッションを変更
pi@raspberrypi /usr/lib/cgi-bin $ ls -l /usr/lib/cgi-bin/ パーミッションが変更されていることを確認
合計 8
-rwxr-xr-x 1 root root 313  8月 20 16:04 SoundSW.cgi
-rwxr-xr-x 1 root root 291  8月 20 16:04 StartStopSW.cgi
pi@raspberrypi /usr/lib/cgi-bin $
```

図 61

図 61 のように、CGI プログラムを指定のディレクトリにコピーしパーミッションを変更する。終了後は”cd ~”を実行しておく 1

```
pi@raspberrypi ~ $ ls -l jump.html ←jump.htmlがあることを確認
-rw-r--r-- 1 pi pi 138  8月 19 13:38 jump.html
pi@raspberrypi ~ $ sudo cp jump.html /var/www/ ←/var/www/にjump.htmlをコピー
pi@raspberrypi ~ $ ls -la /var/www/ ←コピーされたことを確認
合計 20
drwxr-xr-x 3 root root 4096  8月 20 16:46 .
drwxr-xr-x 12 root root 4096  8月  9 11:10 ..
-rw-r--r-- 1 root root 1048  8月 20 16:47 index.html
-rw-r--r-- 1 root root 138  8月 20 16:46 jump.html
drwxrwxr-x 3 root root 4096  8月 19 17:58 radmon.data
pi@raspberrypi ~ $
```

図 62

図 62 は CGI から直接 locate で index.html を表示させるとブラウザのアドレス欄に CGI の URL が残る。本来は元の URL となるべきなので、一旦ダミーの jump.html を表示させ、その html ファイルの中には、index.html を表示するというタグを記述することで、解消した。

12.3. ボタン表示更新のタイミング

CGI で操作することのできる動作は、検出音の ON/OFF と計測停止/再開で、ブラウザ上の表示は以下のようにになっています。

	ブラウザでの表示	動作
検知音 ON/OFF	検出音有→無	検出音が鳴らない状態にする
	検出音無→有	検出音が鳴る状態にする
計測一時停止・再開	計測中→停止	計測を一時停止する
	計測停止中→開始	計測を再開する

13. Radmon_pi を/etc/initab へ登録する

作成された”radmon_pi”がシステムの立ち上がりと同時に実行できるよう、/etc/initab に登録する。ホームディレクトリの”radmon_pi.sh”を登録する形にする。このファイルは、/etc/init.d/skelton をホームディレクトリに”radmon_pi.sh”なる名前でコピーし、radmon_pi 用に変更したものである。用意したシェルを以下の手順で/etc/init.d/に設定する。

```
pi@raspberrypi ~ $ ls -l /etc/init.d/r* ←コピー先の”r”から始まるファイルのリスト
-rwxr-xr-x 1 root root 6120 10月 16 2012 /etc/init.d/rc
-rwxr-xr-x 1 root root 782 10月 16 2012 /etc/init.d/rc.local
-rwxr-xr-x 1 root root 117 10月 16 2012 /etc/init.d/rcS
-rwxr-xr-x 1 root root 639 10月 16 2012 /etc/init.d/reboot
-rwxr-xr-x 1 root root 1074 12月 11 2012 /etc/init.d/rmnologin
-rwxr-xr-x 1 root root 2344 6月 15 2012 /etc/init.d/rpcbind
-rwxr-xr-x 1 root root 4395 12月 13 2012 /etc/init.d/rsync
-rwxr-xr-x 1 root root 3054 3月 6 07:07 /etc/init.d/rsyslog
pi@raspberrypi ~ $ sudo cp radmon_pi.sh /etc/init.d/radmon_pi ←ホームディレクトリのシェルをコピー
pi@raspberrypi ~ $ sudo chmod 755 /etc/init.d/radmon_pi ←パーミッションを設定
pi@raspberrypi ~ $ ls -l /etc/init.d/r* ←増えたファイルの確認
-rwxr-xr-x 1 root root 4291 8月 15 16:36 /etc/init.d/radmon_pi ←増えたファイルとパーミッションを確認
-rwxr-xr-x 1 root root 6120 10月 16 2012 /etc/init.d/rc
-rwxr-xr-x 1 root root 782 10月 16 2012 /etc/init.d/rc.local
-rwxr-xr-x 1 root root 117 10月 16 2012 /etc/init.d/rcS
-rwxr-xr-x 1 root root 639 10月 16 2012 /etc/init.d/reboot
-rwxr-xr-x 1 root root 1074 12月 11 2012 /etc/init.d/rmnologin
-rwxr-xr-x 1 root root 2344 6月 15 2012 /etc/init.d/rpcbind
-rwxr-xr-x 1 root root 4395 12月 13 2012 /etc/init.d/rsync
-rwxr-xr-x 1 root root 3054 3月 6 07:07 /etc/init.d/rsyslog
pi@raspberrypi ~ $
```

簀戸

図 63

Radmon_pi を/usr/sbin にコピーする。(図 64)

```
pi@raspberrypi ~ $ ls -la /usr/sbin/radmon*
ls: /usr/sbin/radmon* にアクセスできません: そのようなファイルやディレクトリはありません
pi@raspberrypi ~ $ sudo cp radmon_pi /usr/sbin/radmon_pi
pi@raspberrypi ~ $ ls -la /usr/sbin/radmon*
-rwxr-xr-x 1 root root 12915 8月 19 10:34 /usr/sbin/radmon_pi
pi@raspberrypi ~ $
```

図 64

Insserv コマンドで起動時のスクリプトとして登録する。(図 65)

```
pi@raspberrypi ~ $ sudo insserv radmon_pi ←起動時のスクリプトとしてinsservコマンドで登録
pi@raspberrypi ~ $ sudo find /etc/ -name "*radmon_pi" ←登録されたことを確認
/etc/rc6.d/K01radmon_pi
/etc/rc4.d/S03radmon_pi
/etc/rc3.d/S03radmon_pi
/etc/rc5.d/S03radmon_pi
/etc/rc1.d/K01radmon_pi
/etc/rc2.d/S03radmon_pi
/etc/rc0.d/K01radmon_pi
/etc/init.d/radmon_pi
pi@raspberrypi ~ $
```

図 65

再起動で radmon_pi が自動的に起動する

```
pi@raspberrypi ~ $ sudo shutdown -r now
```

14. おわりに

以上で、Raspberry Pi と CPI-UR001 を接続してネット経由でデータを取得する最小限の設定で紹介したが、実際に利用する際はセキュリティ関連の設定をお勧めする。

また、取得したデータは、CPI-UR001 に添付されている Windows 版 RADMON にて表示できる。但し、注意点としては Raspberry Pi は扱っている文字コードが UTF-8 で、日本語 Windows とそのアプリケーションは SJIS で作成されているので、ヘッダーに日本語を使用する際は、コードが SJIS になっていることに注意が必要である。