

RaspberryPi 用に WSPR-X をコンパイル ©2015 JA5FP

本稿では、RaspberryPi 2 model B を用いて WSPR-X を動作させるための、追加ハードウェアとソフトウェアのインストールについて述べます。

1 RaspberryPi のハードウェア資源

低消費電力でかつ静穏な小型のカード PC である RaspberryPi によって、WSPR の長時間運用に最適な環境を作ることができます。

Raspberry Pi の最新モデルである Raspberry Pi 2 B は LAN や USB など一通りの IO を備えています。WSPR に必須である AF 入力デバイスがありません。その理由は十分な S/N を確保できる基板スペースがないための設計と思われる。

そこで対応策として、USB 外付けオーディオカードを用いて、AF 信号入出力を行います。市場には多品種の USB オーディオが出回っていますが、安価な物では内部雑音が高いかもしれませんので微弱な信号を対象とする WSPR 用には高品質の品を使う方がよいでしょう。とりあえず、Creative 社の SB1140 と Onkyo 社の SE-U55 オーディオプロセッサを接続しました。

表 1: Raspberry Pi 2B のハードウェアと主要性能

SoC	Broadcom BCM2836
CPU	900 MHz / ARM Cortex-A7 4 コア
GPU	250 MHz / Broadcom VideoCore IV
メモリ	1 GB / SDRAM
USB 2.0 ポート	(LAN9512 内蔵ハブ) 4
映像入力	15 ピン MIPI カメラ (CSI)
映像出力	RCA (PAL/NTSC), HDMI 1.3/1.4
音声出力	3.5mm ジャック, HDMI, I2S
ストレージ	microSD カード
ネットワーク (RJ45)	LAN9514(10/100Mbps イーサネット)
低レベル周辺機器	GPIO ヘッダーピン 40 ピン
電源	800mA (4.0W)
電源ソース	5V / microUSB または GPIO

2 WSPR-X のインストール

Joe Taylor, K1JT による WSPR-X ソフトウェアは OS ごとに、Windows 版および Mac OS X 版は実行ファイルとして提供されていますので PC へのインストールは簡単です。それらは次の URL からダウンロードできます。

<http://physics.princeton.edu/pulser/K1JT/wspr.html>

一方 Raspberry Pi のように OS が Linux の場合では、現段階ではユーザが自身でコンパイルする必要があります。さらにプログラム開発ツールも進化過程にあり、バージョンごとにコンパイル

方法が異なり、少しトリッキーな手法を使います。それがために、普及が進んでいないのでしょう。

そこで筆者の実体験を基に、RaspberryPi への WSPR-X の現実的なインストールの道筋を紹介します。この手順は、George Smart,M1GEO の方法を参考にしています。

出典:http://www.george-smart.co.uk/wiki/Compiling_WSPR-X

まず最初に、ダウンロードとコンパイルに必要なツールとライブラリーのパッケージを Debian サイトから入手します。次の囲み内はコマンドライン画面から入力する内容を示し、以下の全ての段階でも同じです。

```
sudo apt-get install subversion python2.7-dev python-numpy python-imaging-tk python-pmw
libportaudio2 portaudio19-dev libsamplerate0-dev gfortran cl-fft3
python-dev qt4-qmake libqt4-dev-bin libqt4-dev libqwt-dev libhamlib-utils
```

インターネットに接続した状態ならば、`apt-get install` コマンドにより自動的にインストールできます。

次に WSPR-X コード本体のダウンロードです。George は Berlios サイトからのダウンロードを例示していますが、そのサイトが現在は閉鎖されているようです。代替というより本来の開発サイトである SourceForge から、次のようにサブバージョンを入手します。

```
svn co svn://svn.code.sf.net/p/wsjt/wsjt/branches/wsprx@3621
```

ここで@3621 の部分は WSPR-X v0.9 r3621 を指しており、省略すると別バージョンがインストールされてしまいます。ここでは r3621 を明示しておきます。

svn によって、ホームディレクトリー下に `wsprx` ディレクトリー以下が自動的に構成されますので、その `lib` ディレクトリーに移動します。

```
cd ~/wsprx/lib
```

Linux 用の `make` ファイルを指定して `make` します。

```
make -f Makefile.linux
```

一度 `wsprx` ディレクトリーに上ります。

```
cd ..
```

`wsprx.pro` の内容を修正するために、スーパーユーザ権限で `nano` により `wsprx.pro` ファイルを開きます。

```
sudo nano wsprx.pro
```

編集は、その 57 行目にある `qwt-qt4` を `qwt` に変更します。また 59 行目にある `libqwt-qt4.so` を `libqwt.so` とし、上書き保存します。

次の手順に進みます。

```
qmake
```

ビルドを行います。

```
make
```

ここで次のエラーメッセージが出ることでしょう。

```
mkdir: cannot create directory './wsprx.install/': File exists
```

```
make: ***[./wsprx.install/wsprx] Error 1
```

そこで、次のコマンドでファイルを削除します。

```
rm -R ../wsprx_install
```

再度 make します。

```
make
```

今度はエラーなしにビルドができたでしょう。
再度 lib ディレクトリーに移り、次の手順を執ります。

```
cd lib/  
make clean  
make -f Makefile.linux
```

実行ファイルなどが格納されているディレクトリーと中味を確認します。

```
cd ../../wsprx_install  
ls
```

WSPRcode wsprd wsprx の 3 ファイルが生成されているはずですが、wsprx が実行ファイルです。
ここでデータを収納する save フォルダを作成しておきます。

```
mkdir save
```

以上で WSPR-X のインストールと動作準備が整いました。直ちに WSPR-X の起動を行うには、次のとおりです。

```
./wsprx
```

なお、次回からの起動も ~ /wsprx_install ディレクトリーから行います。

3 WSPR-X の設定

WSPR-X が起動され、受信信号オーディオ信号を正常にデコードすると次の画面になります。

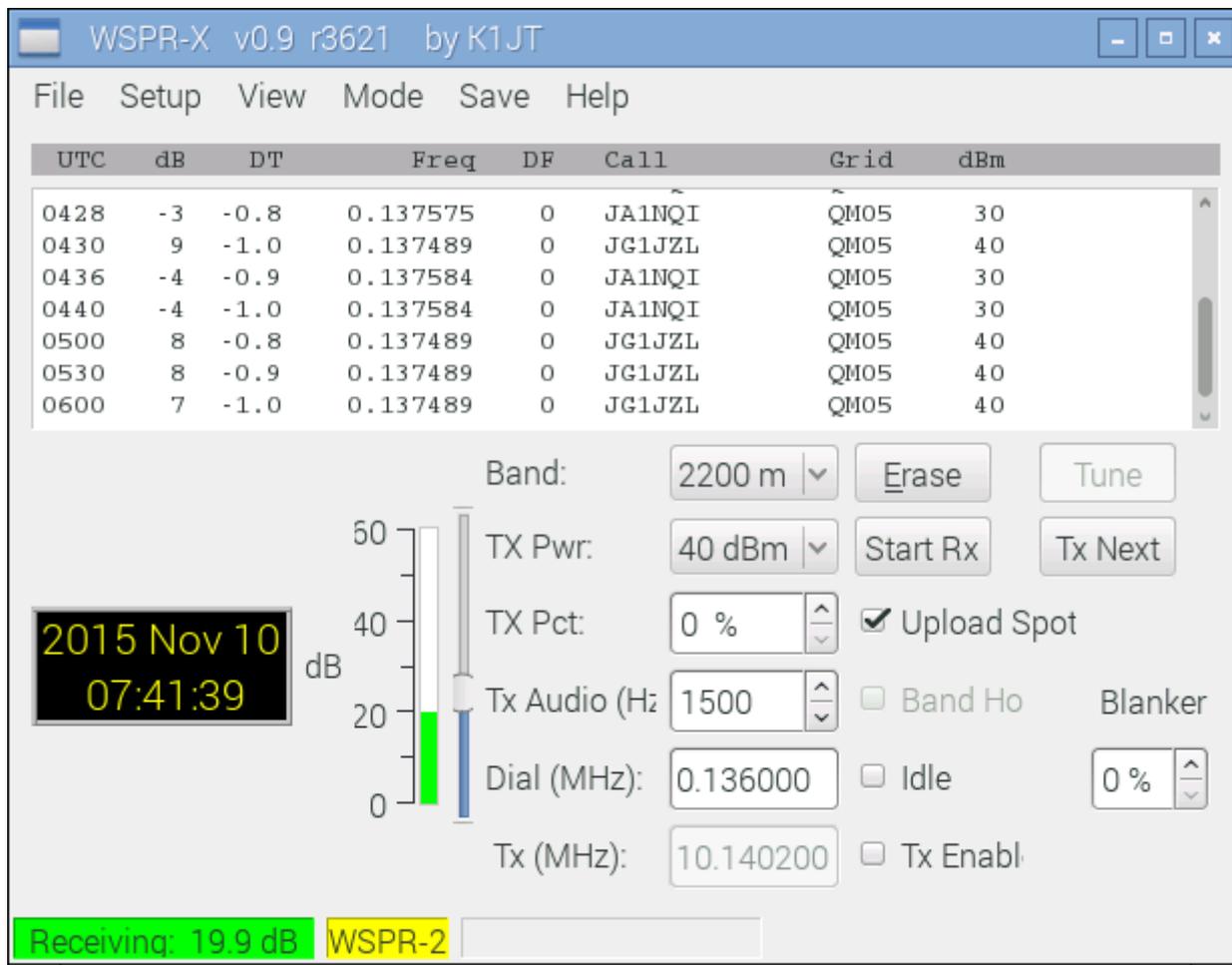


図 1: WSPR-X のメイン画面

コールサインや GL などの各項目について適当な設定を行う必要があります。その詳細な内容は次の URL の説明を参照してください。

http://www.physics.princeton.edu/pulsar/K1JT/WSPR-X_Users_Guide.pdf

RaspberryPi に特有なオーディオ入出力デバイスについては、筆者の環境では次のとおりとしました。

SB1140 では Setup > Options F2 > Audio In : 3 128 sysdefault

Audio Out : 0 0 USB Device0x41e:0x30d3:Audio(hw(0,0))

SE-U55 では Setup > Options F2 > Audio In : 3 128 sysdefault

Audio Out : 1 0 bcm2835 ALSA:-hw(0,0)

(オーディオ出力は RaspberryPi の 3.5mm ジャック)

WSPR-X を最初に立ち上げた際は”sysdefault” とし、自動検索されたオーディオデバイス候補の中から適応する USB デバイスを選択するのが良いようです。ただし、サンプリングレートなどの

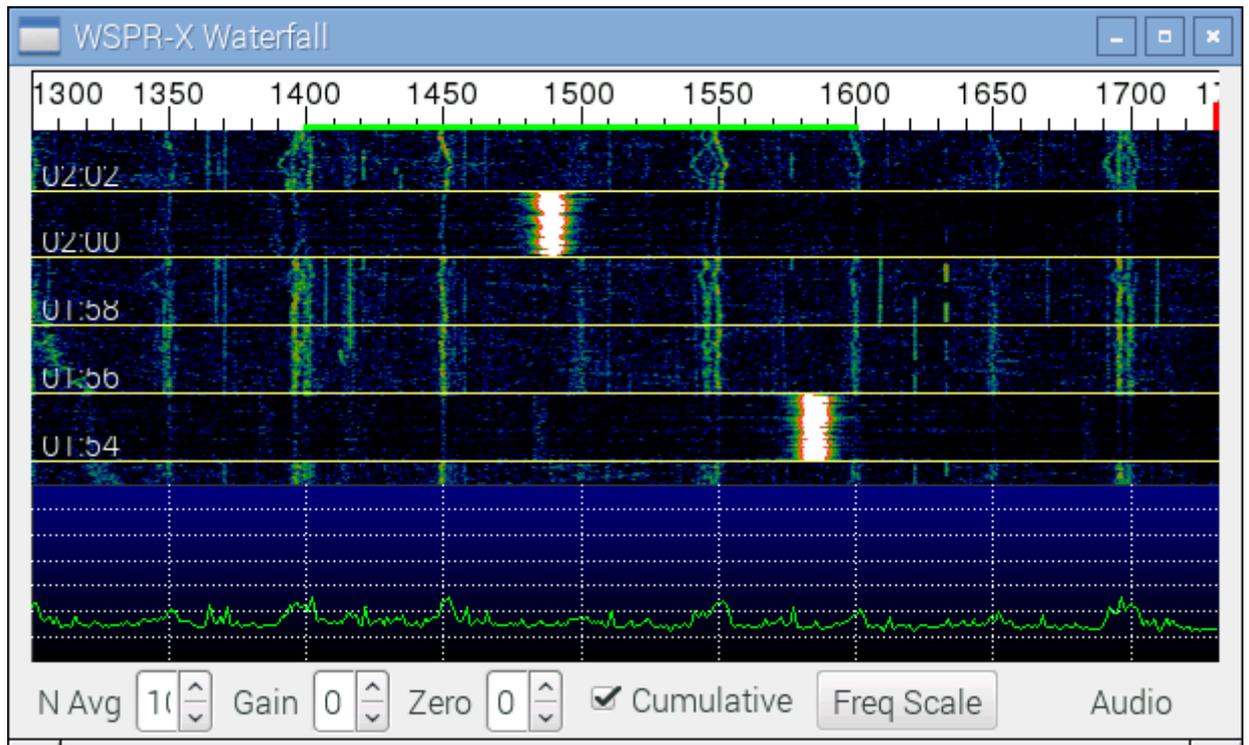


図 2: WSPR-X のウォーターフォール

条件によりオーディオ出力周波数が所定の 1/2 になる場合がありますので、実際の確認をします。

4 送信その他の動作検証

これまでに受信動作は、Windows 版の WSPR-X v0.8 r3058 と同様であることを確認しました。一般の PC を用いた WSPR では適宜の方法で PC 時刻を u.t.c に対して $\pm 2s$ 以内に修正する必要がありますが、RaspberryPi にはインターネット接続により NPT を取得する機能が備わっていますので、心配無用です。

2015/11/12 記
2015/11/16 追記