FreqEZ Band Decoder

A Versatile Band Decoder

FreqEZ is a Do-It-Yourself hardware/software project that provides highly configurable Band Decoding and Remote Antenna selection. For amateurs who use N1MM+ and DXLab logging software, FreqEZ will leverage those programs' TCP/UDP broadcasts for antenna switching. For other amateurs, FreqEZ can connect to the BCD band outputs available from most transceivers. The FreqEZ software consists of a pair of programs that run on the Windows and Raspberry Pi Raspbian operating systems, and is freely available to all hams.

Assembled FreqEZ enclosure prior to cable installation

Table of Contents

Overview

Hardware

The FreqEZ Parts List consists of components that were intentionally chosen selected as being inexpensive and globally available. Using popular sources like Amazon, eBay, Mouser and DigiKey, hams should be able to purchase all of the FreqEZ components for approximately $100 (US).

Software

FreqEZ is a pair of companion software applications that runs on a Windows PC and a Raspberry Pi (RPi) single-board computer. The user interface and software configuration run on Windows; the remote antenna switching runs on a Raspberry Pi. These two companion programs communicate via Ethernet or WiFi using TCP.

The N1MM+ and DXLab Commander logging programs send UDP broadcast packets containing radio information that the FreqEZ console receives and interprets according to the FreqEZ configuration settings. The console creates command strings that are sent to the Raspberry Pi controller, enabling any combination of up to 16 devices (i.e. antenna switches, band filters, receive antennas, transverters…). From DXLab, that packet information is the transceiver frequency in the <Freq> field. From N1MM+, that packet information could be the transceiver frequency, or it could be the packet's <Antenna> number field entered derived from the N1MM+ Antenna configuration table.

Key Features

- Inexpensive, off-the-shelf, readily available hardware. No custom interface required
- No DIP switches, diodes, solder-bridges. All configuration is via software settings
- Manual antenna selection with simple mouse clicks in the Windows console
- Automatic antenna selection from network-based UDP packets or hard-wired BCD inputs
    - In N1MM+ - use the antenna numbers from the Configurer >Antenna table
        - Supports the N1MM+ Keyboard Shortcut for multiple antennas per band by toggling <alt>+F9
    - In DXLab or N1MM+ - use the Rx radio frequency from the UDP RadioInfo packet
    - Auto-magic switching (no re-configuration) from N1MM's antenna numbers to DXLab's Frequencies
    - Hard-wired method, for logging programs that do not support UDP broadcasts
        - Connect to a transceiver's BCD output jack
        - OR… use BCD signals from a PC LPT parallel port (supported by some logging programs) {ugh!}
- The Windows console and the Raspberry Pi controller communicate via WiFi or wired Ethernet connections
    - The Raspberry Pi controller operates "headless" – can be located anywhere
- FreqEZ controls 16 discrete single-pole double-throw (SPDT) outputs
    - 16 SPDT relays provide "dry contacts" that sink (ground) or source (+12 vdc) in any combination
    - The 16 relays could be configured as four BCD encoded lines for compatible remote antenna switches
- PTT / Tx Inhibit line prevents FreqEZ from switching antennas while transmitting
- Flexible, software-driven configuration for shack-specific customization

Required Materials

- Windows PC Console
  - Windows 7 or later
  - 8 GB of RAM minimum
  - 10 MB of disk space
  - Ethernet/WiFi connection
- Raspberry Pi Controller
  - Model 3b or newer recommended. Must contain 40 GPIO pins
  - Raspberry Pi micro-USB power cable with power supply or PC as a power source
  - 8GB (or larger) microSD card with Raspbian operating system. Fast card (class 10) recommended
- Band Decoder switches
  - Two SainSmart relay modules (or equivalent) with 8 relays each – 5 volt DC relay model
  - Dupont rainbow female-to-female jumper cables for connecting the 40 pin RPi GPIO connector to the SainSmart relay modules. Recommend 30 total conductors, approximately 8 inches in length
- Optional
  - Enclosure – size to accommodate an RPi and two SainSmart 8 relay boards. Search Amazon or eBay for an aluminum enclosure with 203mm x 144mm x 68mm dimensions – many sources and colors available
  - Mounting tray inside the enclosure – rather than connect the hardware devices directly to the case, cut a piece of aluminum or per board to slide along the guides within the enclosure (see photos)
  - 12 volt to 5 volt regulator – for feeding the enclosure with only one DC power source. Useful when operating the console remotely from the shack. Delivers 5 volts DC for the Raspberry Pi and SainSmart relays.
  - Connectors and wire – as appropriate for connecting the enclosure to external remote antenna switches. Suggest D-sub 9 pin female for antennas, a D-sub 9 male for BCD inputs, RCA jack for PTT/Tx.

Step 1. Preparing the Raspberry Pi

Installing the Raspbian Operating System

There are many excellent guides to assist in preparing the Raspbian operating system on your Raspberry Pi. Some recommendations:

Books:
Raspberry Pi in Easy Steps by Mike McGrath
Raspberry Pi User Guide by Gareth Halfacree
Websites:
Setting Up Your Raspberry Pi `http://www.raspberrypi.org/help/quick-start-guide/`
Setting Up a Raspberry Pi with NOOBS https://learn.adafruit.com/setting-up-a-raspberry-pi-with-noobs

Use the guides to install the most recent Raspbian operating system on your Raspberry Pi. After configuring the Raspbian operating system on your RPi, use the Command Line Interface (CLI) console to enter the commands:

```
sudo apt-get update
sudo apt-get upgrade
```

Those commands ensure that you have the latest version of Raspbian and its installed applications.
[LINUX STUFF: the ***sudo*** *command (acronym – "**S**uper **U**ser **DO**") is the Linux equivalent of Windows' "Run As Administrator". It is used when administrative permissions are required for system tasks. See the Helpful Linux Command glossary at the end of this document.]*

From the Raspberry Pi command line, issue the command **`ifconfig t`**o determine the Raspberry Pi controller's IP address. The controller IP address is required for configuring the Windows console.

Strongly recommended: enable SSH terminal access in the >Raspi-Config >Interfaces menu to allow remote configuration, monitoring, and troubleshooting.

Step 2. Downloading and Installing the Software

Downloading and Installing the program

The latest version of FreqEZ can be downloaded from the K8UT.COM website. Look in the >Files >Programs section of the website. The single ZIP file contains the programs for both the Windows and Raspberry Pi installations.

In Windows, create a new directory for the FreqEZ WinBandDecoder program. Unzip all of the contents of the Windows ZIP into this new directory. The FreqEZ Windows package includes four files: this documentation **`FEZhelp.pdf`**, the FreqEZ executable **WinBandDecoder.exe**, the **FEZsettings.xml** file, and the **FEZconfig.xml** file. In addition, an optional **SampleConfig** subdirectory may contain examples of different configurations.

In Raspbian, copy the files from the FreqEZ RPiBandDecoder ZIP into the **/home/pi** directory. The FreqEZ Raspberry Pi package includes three files: the binary executable **rpibanddecoder,** the **fezconfig.xml** file, and a sample **rc.local** autostart file. (NOTE: all lower case filenames in the Linux world).

You must set the **`fezbanddecoder`** binary file's attribute to *Execute*, which can be accomplished several ways:

- Within the FileZilla or WinSCP programs, right-click on the file and set >Properties
- or, Login to the RPi GUI, start the file manager, and right-click on the file and set >Properties

- or, Login to the RPi console, and from the CLI set its properties with: s**udo chmod +x rpibanddecoder**

Adding Boot Auto-Start to FreqEZ

Perhaps you remember using AUTOEXEC.BAT files to automatically start MS-DOS programs on an IBM PC? The Raspbian operating system has a similar feature. You can configure your Raspberry Pi to automatically launch FreqEZ when the RPi reboots. Raspberry Pi's boot commands are located in a file called **rc.local** in the **/etc** directory. Editing this file requires root user (sudo) privileges. From the Raspberry Pi CLI, type the command **sudo nano /etc/rc.local** . When ***nano*** (a text editor) opens, use the down-arrow key to move near the bottom of the file and insert the following text on a line immediately above the **exit 0** statement:

```
cd /home/pi
sudo ./rpibanddecoder
```

After adding those lines, press **<crtl>+X** to exit, **Y** to save the changes, then **<Enter>**. From now on, FreqEZ will automatically start whenever the Raspberry Pi boots.

IMPORTANT TROUBLESHOOTING DETAIL: With FreqEZ auto-start enabled, an instance of FreqEZ will always be running after the RPi boots. When you connect to the RPi console, you can confirm that FreqEZ is running by displaying the current tasks with Linux's command **ps -A** *(*rpibanddecoder *should be found within that long list of tasks) If you were to manually launch* rpibanddecoder *again from the CLI, you would then be running two instances of the program - with very strange results. To avoid the collision of two instances, halt the auto-started instance from the CLI by typing the Linux command* **sudo killall rpibanddecoder** *. Confirm that* rpibanddecoder *is not running with the* **ps -A** *command. Then launch a fresh instance of the program with the command* **sudo ./rpibanddecoder .**

Step 3. Configuring the Software

Raspberry Pi fezconfig.xml Configuration File

FreqEZ on the Raspberry Pi is controlled by statements in the **fezconfig.xml** configuration file. On the Raspberry Pi, this file contains a single parent section called FEZconfig. That parent section contains several child statements, which begin with an opening tag <childexample> and end with an XML closing tag </childexample>.

Using the Raspberry Pi's text editor

The standard applications on the Raspberry Pi include two editors: a GUI-based editor called ***LeafPad***, and a command line editor called ***nano.*** If you have a mouse, keyboard, and monitor connected to the RPi with a GUI display, then it is probably easiest to use the RPi file explorer and *Leafpad* to edit the configuration file. Otherwise, you can connect to your RPi via an SSH/ telnet session and use *nano*. To launch nano and edit the configuration file, open a terminal session (command line interface = CLI) using a telnet client like PuTTY, and enter the command ***nano fezconfig.xml***. (NOTE: all lower case letters. Linux is CaSe SeNsItIvE.) In the nano editor you must navigate using the up/down arrays (the mouse will not work here). To exit the editor, type ctrl+X. If you have made changes *nano* will ask whether you want to save those changes. Press Y and then <enter> to save with the same filename and return to the CLI. After savings changes to fezconfig.xml, the program will automatically update with the new settings.

IMPORTANT: When editing the configuration file, adhere to the UPPER and lower CaSe ChAraCteRs as shown in this documentation's examples. The program will not recognize statements with incorrect XML case.

Modifying fezconfig.xml on the Raspberry Pi

```
<FEZconfig>
```

```
        <StationName></StationName>
        <ParentName>RadioInfo</ParentName>
        <RPiRecvPort>13066</RPiRecvPort>
        <FEZverbose>OFF</FEZverbose>
</FEZconfig>
```

- **StationName is normally empty**. StationName is an *optional* entry that instructs FreqEZ to only process packets from a specific N1MM+ station. By default, this entry is empty and FreqEZ accepts valid packets from any station on the network. However, on multi-station networks it may be necessary to filter by the sending computer's NetBIOS name (as seen in the N1MM+ network table) to ensure that this instance of FreqEZ only responds to antenna commands sent from a specific station
- **ParentName is normally AntennaInfo**. AntennaInfo is the title of the parent tag within packets sent to the Raspberry Pi from the Windows console. This parent title must match the parent title used by the WinBandDecoder console program. Do not change this entry unless you are running a custom version of FreqEZ
- **RPiRecvPort is normally 13062**. RPiRecvPort is the port used by the WinBandDecoder console program to communicate with the Raspberry Pi. Do not change RPiRecvPort without also changing the corresponding RPiPort entry in the FEZconfig.xml file on the Windows computer
- **FEZverbose is normally OFF**. FEZverbose is used for troubleshooting packets coming from the Windows FreqEZ computer. With FEZverbose=ON, rpibanddecoder prints the entire packet to the Linux console as each packet is processed

Windows FEZconfig.xml Configuration File

FreqEZ on Windows is controlled by statements in the **fezconfig.xml** configuration file. The file contains one mandatory parent section, FEZconfig, and at least one or more other parent sections. Each parent section begins with an XML opening tag <parentexample> and ends with an XML closing tag </parentexample>. Each parent contains many child data statements, which begin with a similar opening tag <childexample> and ends with an XML closing tag </childexample>.

The mandatory parent section is called FEZconfig. FEZconfig contains six child statements, which begin with an opening tag <childexample> and end with an XML closing tag </childexample>.

```
<FEZconfig>
        <StationName></StationName>
        <ParentName>RadioInfo</ParentName>
        <RecvPort>12062</RecvPort>
        <RPiAddress>192.168.1.149</RPiAddress>
        <RPiPort>13066</RPiPort>
        <BandSource>FREQ</BandSource>
</FEZconfig>
```

- **StationName is normally empty**. StationName is an *optional* entry that instructs FreqEZ to only process packets from a specific N1MM+ station based on the <StationName> tag in the packet. By default, this entry is empty; however, on multi-station networks it may be necessary to enter the sending computer's NetBIOS name (as seen in the N1MM+ network table) to ensure that this instance of FreqEZ only responds to antenna commands sent from a specific station

- **ParentName is normally RadioInfo**. RadioInfo is the title of the parent tag within packets sent from N1MM+ and DXLab Commander. Do not change this entry unless you are running a custom version of FreqEZ
- **RecvPort is normally 12062**. RecvPort is the port used to receive packets from N1MM+ and DXLab. Do not change RecvPort without also changing the corresponding entry in the N1MM+ Configurer or DXLab Commander.
- **RPiPort is normally 13066**. RPiPort is the port used to communicate with the Raspberry Pi's rpibanddecoder control program. Do not change RPiPort without also changing the corresponding entry in the fezconfig.xml file on the Raspberry Pi
- **RPiAddress has no default value**. RPiAddress is the IP address of the Raspberry Pi running rpibanddecoder. Upon installing your Raspberry Pi you must determine its IP address by issuing an `ifconfig` command from its console. Enter that IP address into the RPiAddress tag in the Windows FEZconfig.xml file
- **BANDsource has three valid values: FREQ, N1MM,** and **BCD**
    - **BANDsource=FREQ**, FreqEZ selects antennas based on the <Freq> tag in the RadioInfo packet
    - **BANDsource=N1MM**, FreqEZ selects antennas based on the <Antenna> tag in the RadioInfo packet
    - **BANDsource=BCD**, FreqEZ selects antennas based on the 4 wire BCD table for your radio

BANDsource=FREQ: Explanation and Examples

BANDsource=FREQ instructs FreqEZ to make antenna selections based on the <Freq> tag in the RadioInfo packet. Upon receipt of a new packet, FreqEZ examines the <FREQradio1> section of the config file and activates relays based on the first entry that falls within the upper and lower limits is an entry.

FREQ example 1: A simple list of antennas with separate relays for each antenna. From the list below, a packet with 2125500 as the frequency would activate relay #4

```
<FREQradio1>
        <FREQ1>5000000-5400000,1</FREQ1>
        <FREQ2>2800000-3000000,2</FREQ2>
        <FREQ3>2400000-2600000,3</FREQ3>
        <FREQ4>2100000-2145000,4</FREQ4>
        <FREQ5>1800000-1820000,5</FREQ5>
        <FREQ6>1400000-1440000,6</FREQ6>
        <FREQ7>1000000-1015000,7</FREQ7>
        <FREQ8>700000-750000,8</FREQ8>
        <FREQ9>350000-400000,9</FREQ9>
        <FREQ10>180000-200000,10</FREQ10>
        . . . up to <FREQ50> entries
</FREQradio1>
```

FREQ example 2: A multi-band beam that is fed with one coax (e.g. Steppir 20 to 6 meters). From the list below, a packet with 1428500 as the frequency would activate relay #1. In this example, relay #1 covers all frequencies from 6 to 20 meters

```
<FREQradio1>
        <FREQ1>5000000-1440000,1</FREQ1>
        <FREQ2>1000000-1015000,3</FREQ2>
        <FREQ3>700000-750000,5</FREQ3>
        <FREQ4>350000-400000,7</FREQ4>
        <FREQ5>180000-200000,9</FREQ5>
        . . . up to <FREQ50> entries
```

```
</FREQradio1>
```

FREQ example 3: A list with two dual-band (“fan”) dipoles for 30 & 40, 80 & 160. From the list below, a packet with 1012500 as the frequency would activate relay #7. Any frequency on 30 or 40 meters would activate relay #7

```
<FREQradio1>
        <FREQ1>5000000-5400000,1</FREQ1>
        <FREQ2>2800000-3000000,2</FREQ2>
        <FREQ3>2400000-2600000,3</FREQ3>
        <FREQ4>2100000-2145000,4</FREQ4>
        <FREQ5>1800000-1820000,5</FREQ5>
        <FREQ6>1400000-1440000,6</FREQ6>
        <FREQ7>1000000-1015000,7</FREQ7>
        <FREQ8>700000-750000,7</FREQ8>
        <FREQ9>350000-400000,8</FREQ9>
        <FREQ10>180000-200000,8</FREQ10>
        . . . up to <FREQ50> entries
</FREQradio1>
```

FREQ example 4: Activate several relays simultaneously. Suppose you wanted to activate a remote antenna switch, a receive antenna relay, and a different bandpass filter for every band? From the list below, a packet with 715100 as the frequency would activate relays #2, 5, and 14.

```
<FREQradio1>
        <FREQ1>350000-400000,1,5,13</FREQ1>
        <FREQ2>7000000-740000,2,5,14</FREQ2>
        <FREQ3>1010000-1015000,3,5,15</FREQ3>
        . . . up to <FREQ50> entries
</FREQradio1>
```

FREQ example 5: Divide bands into frequency segments with different antennas for each segment. Suppose you wanted to activate a different dipole on 80 meters than on 75 meters? From the list below, a packet with 380300 as the frequency would activate relay #3.

```
<FREQradio1>
        <FREQ1>180000-200000,1</FREQ1>
        <FREQ2>350000-370000,2</FREQ2>
        <FREQ3>370000-400000,3</FREQ3>
        <FREQ4>700000-740000,4</FREQ4>
        <FREQ5>1010000-1015000,5</FREQ5>
        . . . up to <FREQ50> entries
</FREQradio1>
```

FREQ example 6: If your remote antenna switch requires a BCD signal rather than individual wires for each relay, you can use FreeqEZ's "multiple relay" feature to impersonate a BCD signal. Refer to the BCD table supplied by your transceiver manufacturer. This example contains the BCD signals supplied from an Elecraft transceiver.

```
<FREQradio1>
        <FREQ1>180000-200000,1</FREQ1>
        <FREQ2>350000-4000000,2</FREQ2>
        <FREQ3>700000-740000,1,2</FREQ3>
        <FREQ4>1010000-10150000,3</FREQ4>
        <FREQ5>1400000-1450000,1,3</FREQ5>
        <FREQ6>1806800-1816800,2,3</FREQ6>
        <FREQ7>2100000-2150000,1,2,3</FREQ7>
        <FREQ8>2489000-2499000,4</FREQ8>
        <FREQ9>2100000-2145000,1,4</FREQ9>
        <FREQ10>500000-540000,2,4</FREQ10>
</FREQradio1>
```

BANDsource=N1MM: Explanation and Examples

BANDsource=N1MM instructs FreqEZ to make antenna selections based on the <Antenna> tag in a RadioInfo packet from the N1MM+ logging program. Upon receipt of a new packet, FreqEZ examines the <N1MMradio1> section of the config file and activates relays based on an exact match between antenna number and the N1MMradio1 tag.

N1MM example 1: A simple list of 8 antennas with separate relays for each antenna. From the list below, a packet with Antenna #5 as the antenna would activate relay #5

```
<N1MMradio1>
```

```
        <N1MM1>1</N1MM1>
        <N1MM2>2</N1MM2>
        <N1MM3>3</N1MM3>
        <N1MM4>4</N1MM4>
        <N1MM5>5</N1MM5>
        <N1MM6>6</N1MM6>
        <N1MM7>7</N1MM7>
        <N1MM8>8</N1MM8>
        . . . up to <N1MM16> entries (max number in the N1MM+ Configurer)
</N1MMradio1>
```

N1MM example 2: Similar to the FREQ examples, a single antenna number from N1MM can activate multiple relays in FreqEZ. From the list below, a packet with Antenna #4 as the antenna would activate relays #4, 6, and 14

```
<N1MMradio1>
        <N1MM1>1,9,11</N1MM1>
        <N1MM2>2,6,12</N1MM2>
        <N1MM3>3,6,13</N1MM3>
        <N1MM4>4,6,14</N1MM4>
        <N1MM5>5,6,15</N1MM5>
        <N1MM6>6,16</N1MM6>
        <N1MM7>7</N1MM7>
        <N1MM8>8</N1MM8>
        . . . up to <N1MM16> entries (max number in the N1MM+ Configurer)
</N1MMradio1>
```

N1MM example 3: See *FREQ example 6* for a method by which N1MM+ antenna numbers could drive a BCD-requiring remote antenna switch.

NOTES for using BANDsource=N1MM:

If N1MM+ is your contesting program, and DXLab is your general-purpose logging program, and you want to use BANDsource=N1MM when operating N1MM but BANDsource=FREQ while operating DXLab: Declare BANDsrource=N1MM and configure tables for both sources. FreqEZ will activate relays based on the N1MM antenna numbers only when the received packets include antenna numbers. If the packets do not include antenna numbers (as would be the case for a packet from DXLab), FreqEZ will automatically shift gears and activate relays based on BANDsource=FREQ. Thus you can switch back-and-forth between your contesting program and your general logging program without recjonfiguration.

FreqEZ supports the N1MM+ antenna toggle feature activated by <alt>+F9. Define your antennas per the Antenna Tab Field instructions in the N1MM+ documentation and FreqEZ will cycle through your antennas-per-band with each press of <alt>+F9.

BANDsource=BCD: Explanation and Examples

BANDsource=BCD

Future deliverable

Step 4. Hardware - Connecting the Raspberry Pi to the SainSmart Relay Modules

Refer to the manufacturer's instructions for your remote antenna switches. From those instructions decide how to wire your SainSmart relays. In most cases, you will want to wire a common ground or a common + 12 volts DC supply to the Normally Open (N/O) contacts of the 8 relays.

The switched contacts of the SainSmart relays will connect to external devices such as remote antenna switched and band filters. The following diagram describes pn the SainSmart modules.that is connected to your remote antenna switch when the selected relays are activated.

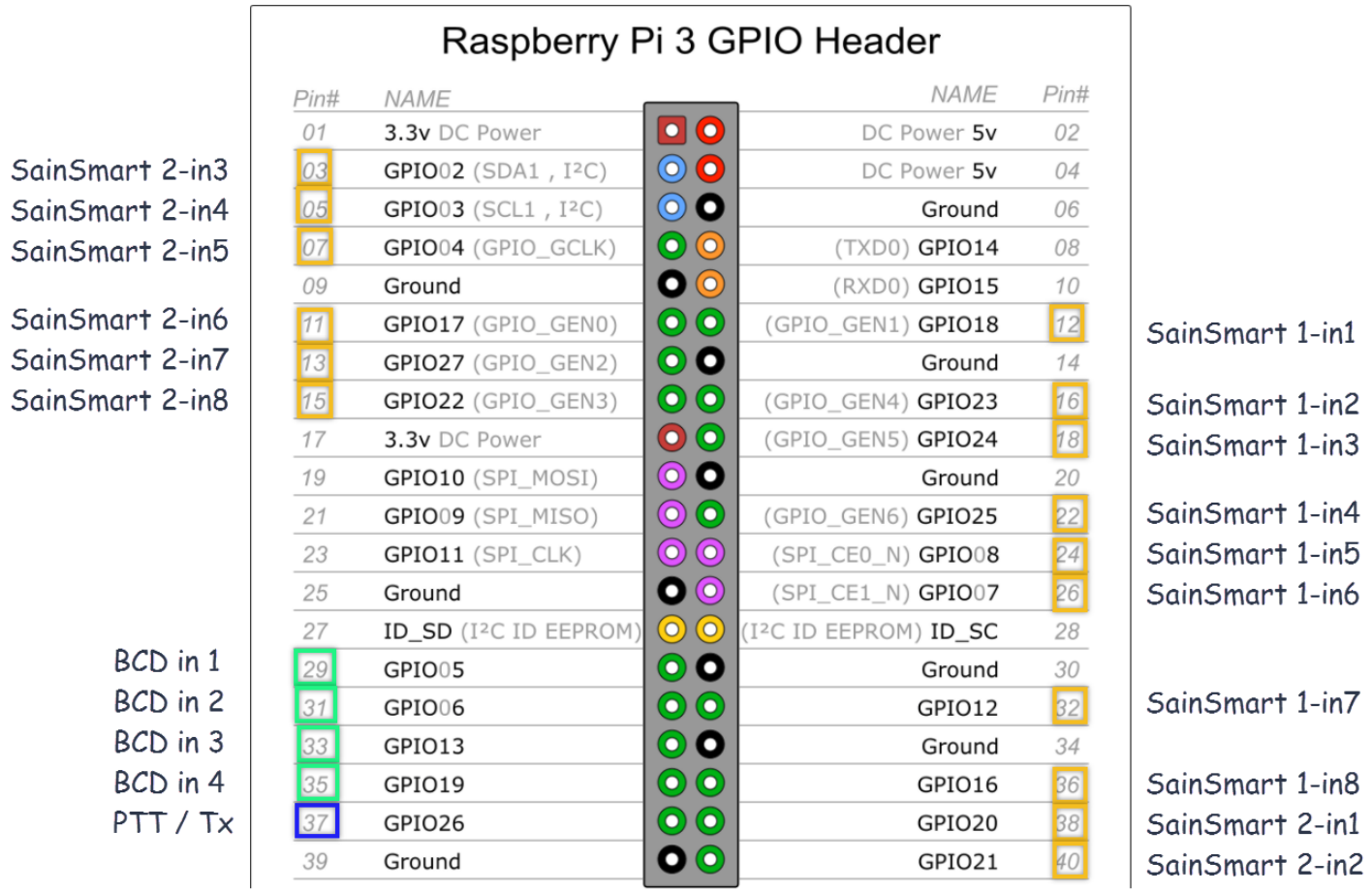
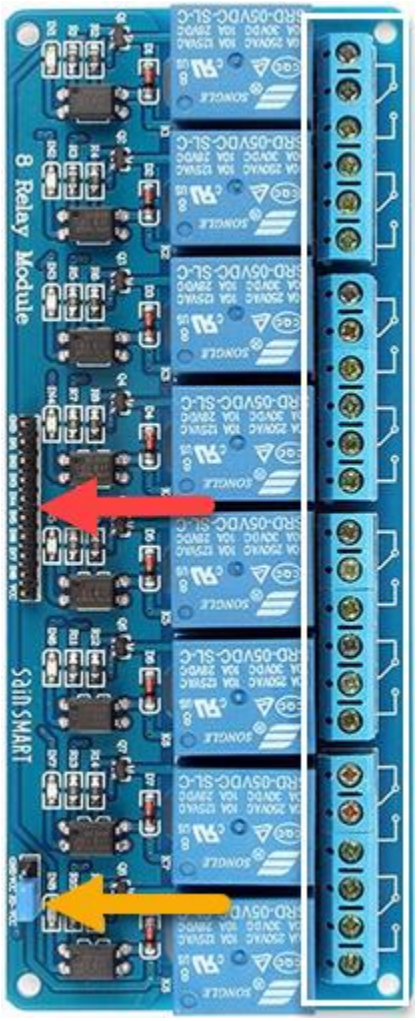


Figure 1: The Raspberry Pi 40 Pin GPIO Connector (courtesy: www.ele4ment14.com)

Figure 1 shows the allocation of FreqEZ pins for Band Decoder switching (16 orange rectangles), for optional BCD control (4 green rectangles), and optional PTT sense/Tx inhibit (1 blue rectangle).

Connecting to the Raspberry Pi for Band Decoder Operation

Refer to the red arrow in adjacent photo. This table lists the connection scheme between the 8 relay board and the Raspberry Pi GPIO pins for Band Decoder operation.



10 pin connector on 8 relay board. For relays 1-8

```
GND (ground) to GPIO pin 6
In1 (relay 1) to GPIO pin 12
In2 (relay 2) to GPIO pin 16
In3 (relay 3) to GPIO pin 18
In4 (relay 4) to GPIO pin 22
In5 (relay 5) to GPIO pin 24
In6 (relay 6) to GPIO pin 26
In7 (relay 7) to GPIO pin 32
In8 (relay 8) to GPIO pin 36
VCC (+5 vdc) to GPIO pin 4
```

10 pin connector on a second 8 relay board. For relays 9-16

```
GND (ground) to GPIO pin 34
In1 (relay 9) to GPIO pin 38
In2 (relay 10) to GPIO pin 40
In3 (relay 11) to GPIO pin 3
In4 (relay 12) to GPIO pin 5
In5 (relay 13) to GPIO pin 7
In6 (relay 14) to GPIO pin 11
In7 (relay 15) to GPIO pin 13
In8 (relay 16) to GPIO pin 15
VCC (+5 vdc) to GPIO pin 2
```

Step 5. Hardware - Connecting SainSmart Relay Modules to Remote Devices

Refer to the manufacturer's instructions for your remote antenna switches. From those instructions decide how to wire the output contacts (white rectangle below) of your SainSmart relays. In most cases, you will want to wire a common ground or a common + 12 volts DC supply to the Normally Open (N/O) contacts of the 8 relays.

All outputs from the relay modules are identical: one single-pole double-throw (SPDT) "dry" set of contacts for each of 8 external connections. The switched contacts of the SainSmart relays connect to external devices such as remote antenna switched and band filters. In the following photo, the relay contacts begin on the left side with relay 1, through to relay 8 on the right side.

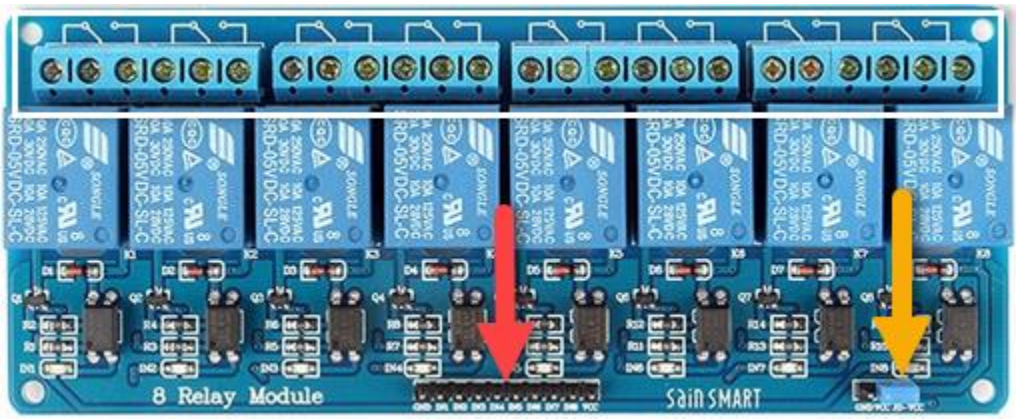


Photo 1: 8 Relay model showing connection for GPIO (red) rainbow cable and the jumper for 5 VDC operation (orange)

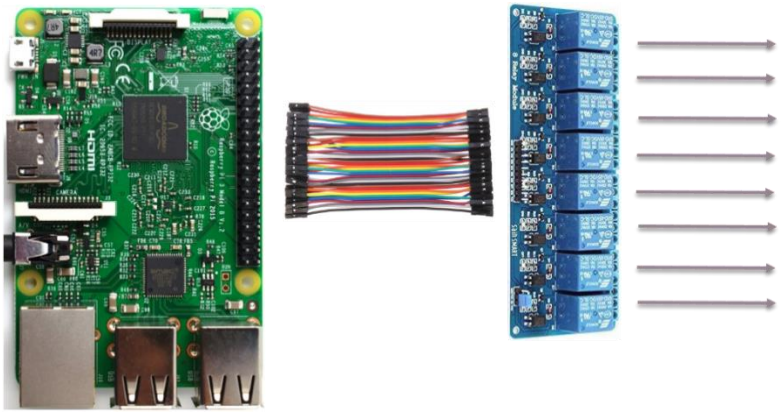


Figure 2: Raspberry Pi, Dupont rainbow jumpers, SainSmart relay module, outputs to external devices

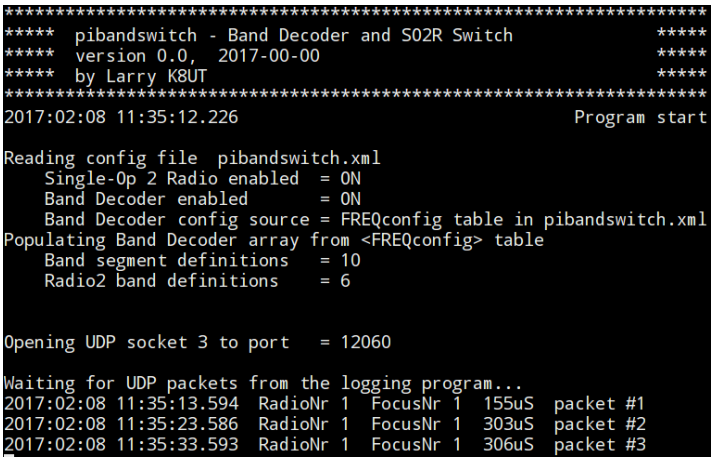
Step 4. Operating FreqEZ

Launching the program

FreqEZ can be started from the console, or configured to start automatically when the RPi is rebooted (see Software configuration section). To launch FreqEZ from the console:

navigate to the Pi user's home directory:	`cd ~` *or* `cd \home\pi`
start the program:	`./FreqEZ`

When running, FreqEZ prints status information to the console. At a minimum, the program prints a splash screen upon launch and thereafter one line of information with each received packet.



FreqEZ screenshot with UDPverbose=OFF, BANDverbose=OFF and SO2Rverbose=OFF

This minimal console shows the launch Splash Screen followed by a one-line summary for each received packet.

```
<?xml version="1.0" encoding="utf-8"?>
<RadioInfo>
        <StationName>K8UT-PC</StationName>
        <RadioNr>1</RadioNr>
        <Freq>1807460</Freq>
        <TXFreq>1807460</TXFreq>
        <Mode>CW</Mode>
        <OpCall>K8UT</OpCall>
        <IsRunning>False</IsRunning>
        <FocusEntry>2694038</FocusEntry>
        <Antenna>6</Antenna>
        <Rotors>YaesuCom15</Rotors>
        <FocusRadioNr>1</FocusRadioNr>
        <IsStereo>False</IsStereo>
</RadioInfo>
```

FreqEZ screenshot with UDPverbose=ON and a UDP <RadioInfo> packet from N1MM+.

The UDPverbose console shows the contents of the most recent UDP packet. The key FreqEZ fields in an N1MM packet are RadioNr, FocusRadioNr, Freq, Antenna and IsStereo.

NOTE: The difference between an N1MM UDP packet and a DXLab UDP packet.

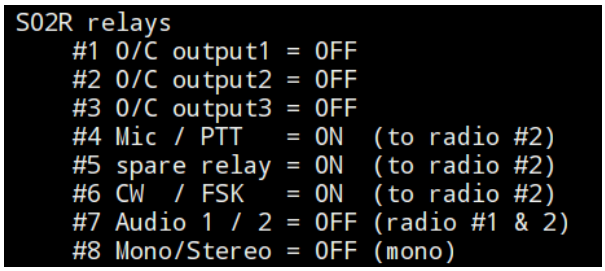
```
<?xml version="1.0" encoding="utf-8"?>
<RadioInfo>
<RadioNr>1</RadioNr>
<Freq>1807460</Freq>
<TXFreq>1807460</TXFreq>
<Mode>CW</Mode>
</RadioInfo>
```

FreqEZ screenshot with UDPverbose=ON and a UDP <RadioInfo> packet from DXLab Commander. The key FreqEZ fields in a DXLab packet are RadioNr and Freq.

```
Band Decoder relays
    #1 = OFF
    #2 = OFF
    #3 = OFF
    #4 = OFF
    #5 = OFF
    #6 = ON
    #7 = OFF
    #8 = ON
    #9 = OFF
    #10 = ON
    #11 = OFF
    #12 = ON
    #13 = OFF
    #14 = OFF
    #15 = OFF
    #16 = OFF
```

PuTTY screenshot with BANDverbose=ON

The BANDverbose console shows the current status of FreqEZ's sixteen Band Decoder outputs. In this example outputs 6, 8, 10 and 12 are activated.



PuTTY screenshot with SO2Rverbose=ON

The SO2Rverbose console shows the current status of FreqEZ's eight SO2R outputs. In this example FocusRadioNr = 2 and the headphones are in monaural mode. The Mic, PTT, CW and FSK signals are connected to Radio 2.

Troubleshooting

Appendix

Schematic and parts layouts

Pertinent Linux Commands

Unlike DOS and Windows, everything in Linux is case sensitive. Typing cd is not the same as CD; the filename FreqEZ.ini is not the same as FreqEZ.INI. The following commands - all related to installing, configuring, and operating FreqEZ - are entirely lower case.

```
cd                          change directory
cd ~                        change directory to the pi login home directory
cd /home/pi                 change directory to the pi login home directory
ctrl-c                      <ctrl>-c – halt FreqEZ when in the CLI
clear                       clear the screen
ls                          list the files in the current directory
lsusb                       list the devices connected to the USB ports
ls /dev/tty*                list all active O/S connections (including the USB ports)
ls /dev/ttyUSB*             list only the active USB port O/S connections
ps –A                       list the current processes running on the RPi
sudo                        Super User DO – equivalent to Windows "run as administrator"
sudo ./FreqEZ               launch FreqEZ (you must be in the  pi  home directory)
sudo apt-get install <program>   install the listed program (without the brackets)
sudo apt-get remove <program>    uninstall the listed program (without the brackets)
```

```
sudo apt-get update      update the list of available RPi packages and their versions, but does not
                         install or upgrade any packages
sudo apt-get upgrade     actually install newer versions of RPi packages you have
sudo chmod +x FreqEZ     set the FreqEZ binary as an executable program
sudo halt                properly close all O/S processes and prepare for removal of power. wait at
                         least 20 seconds before unplugging
sudo killall FreqEZ      halt all instances of FreqEZ (esp. the Linux auto-launch instance)
sudo nano ~/FreqEZ.ini   launch the nano editor and modify the FreqEZ INI file
sudo nano /etc/rc.local  launch the nano editor and modify the rc.local auto-start file
```

A longer list of Linux commands can be found on the Raspberry Pi Foundation's website:
http://www.raspberrypi.org/documentation/linux/usage/commands.md

References

pi@rpi-Dev-v3c:~ $ clear all

pi@rpi-Dev-v3c:~ $ ./rpibanddecoder

****

***** RPiBandDecoder - Band Decoder switch *****

***** version 0.0, 2018-00-00 *****

***** by Larry K8UT *****

****

2018:01:17 15:34:54.212 Program start

Reading config file fezconfig.xml

Band Decoder enabled =

Band Decoder config source =

Opening program socket 3 to port = 13062

Waiting for UDP packets from WinBandDecoder...

2018:01:17 15:34:58.704 0000010000000000 (1/0)

<?xml version="1.0" encoding="utf-8"?>

<AntennaInfo>

```
<StationName>

</StationName>

    <BandSource>FREQ</BandSource>

    <RelayBits>0000010000000000</RelayBits>

    <CheckSum>32</CheckSum>

</AntennaInfo>
```

```
<N1MM7>9,10,11</N1MM7>
```

Band Decoder translation: *When the UDP packet* <Antenna> *field contains the number 7, activate output relays 9, 10 and 11.*

- By activating a list of output relays, you can do more than just choose an antenna. You can also set a band filter, switch a receive antenna, bypass an amplifier…
- By using the antenna numbers within N1MM+, you can use N1MM's <alt>+F9 Keyboard Shortcut to quickly jump between antenna definitions for the current band. For example, instantly switch between north-east pointing tri-band beam on tower 1, to the southwest pointing quad on tower 2.

Optional – using different antennas for Radio 2 than for Radio1: the <RADIO2config> section

Normally FreqEZ activates the same Band Decoder selections for Radio 1 and Radio 2. When operating on a specific band, some SO2R stations may want to activate different relays for Radio 2 than for Radio 1. The optional parent tag <RADIO2config> </RADIO2config> provides a mechanism for distinct Radio 2 Band Decoder definitions for <FREQconfig> and <N1MMconfig> settings.

RADIO2config example using the FREQ *source for the Band Decoder {DXLab or N1MM]*

```
        <FREQn>list</FREQn>

<RADIO2config>
        <FREQ1>3,4,5,6</FREQ1>
        <FREQ3>5,6,7,8</FREQ3>
        <FREQ6>15</FREQ6>
</RADIO2config>
```

RADIO2config example using the N1MM *source for the Band Decoder [N1MM only]*

```
        <N1MMn>list</N1MMn>

<RADIO2config>
        <N1MM1>3,4,5,6</N1MM1>
        <N1MM3>5,6,7,8</N1MM3>
        <N1MM6>13,14</N1MM6>
</RADIO2config>
```

After configuring your standard band decoder settings in the <FREQconfig> section of the XML file, add a new section titled <RADIO2config>. Insert FREQn and/or N1MMn fields as necessary to identify those bands on which you want Radio 2 to select different relays than Radio 1. Your RADIO2config section can include a mixture of both FREQn and N1MMn entries – the software will figure out which ones to use.

IMPORTANT 1: the number n must correspond to the same n used in your <FREQconfig> and <N1MMconfig> settings!
IMPORTANT 2: use the exact UPPER-lower case expressions shown in the example above!

Define any exceptions to the <FREQconfig> and <N1MMconfig> sections for Radio 2 by adding FREQn or N1MMn statements as seen in these examples. It is not necessary to re-list frequencies where Radio 2 will be using the same settings as Radio 1 – only cases where Radio 2's settings are different than Radio 1.