FreqEZ Band Decoder Software

A Versatile Band Decoder

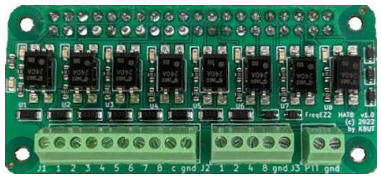
by Larry K8UT

IMPORTANT: There may be a newer version of this Help file on the FreqEZ website. Check this file's date (bottom margin) and compare it to the website's FreqEZhelp.pdf on the website's /Download page.

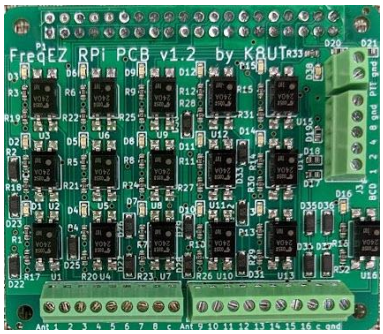
https://freqez.com

FreqEZ is a combination hardware/software project that provides highly configurable Band Decoding and Remote Antenna selection. For amateurs who use N1MM+, DXLab, DXLog, UcxLog, TR4W. Log4OM or Logger32 (and perhaps others) logging software, FreqEZ will leverage those programs' <RadioInfo> UDP broadcasts for antenna switching. For other amateurs, FreqEZ can connect to the BCD band outputs available from most transceivers or from logging programs that support parallel port (LPT) BCD output. Without available UDP packets or BCD band information, FreqEZ can also be used as a sophisticated manual antenna switch.

- **FreqEZ software** is a pair of programs running a Windows *FreqEZ Console* (freqezwin.exe) and a Raspberry Pi *FreqEZ Controller* (freqezrpi). These programs communicate via Ethernet or WiFi using UDP packets. Both programs are freely available to all hams.
- **FreqEZ hardware** uses a Raspberry Pi platform with a controller board to activate remote antenna switches. The Controller hardware can be a pre-assembled and tested Raspberry Pi HAT (Hardware Attached on Top) Controller purchased from K8UT or a Do-It-Yourself construction project from the provided parts list and documentation.



FreqEZ RPi HAT8



FreqEZ RPi HAT16

Do-It-Yourself FreqEZ Controller

Copyright

Disclaimer

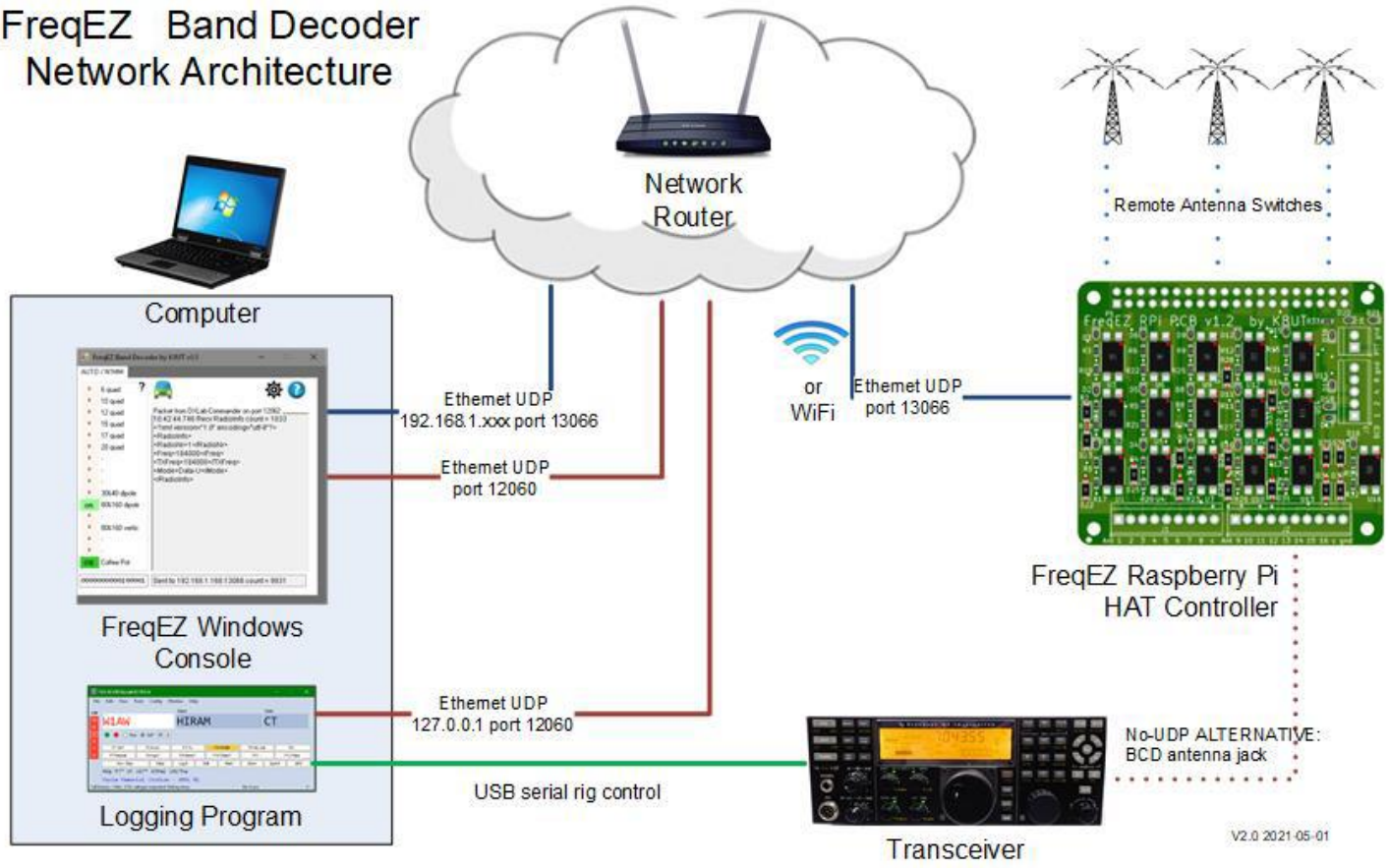
Table of Contents

Overview

The N1MM+, DXLab Commander, DXLog, UcxLog, TR4W and Logger32 (and perhaps other) logging programs send UDP packets containing radio information that the Windows FreqEZ Console receives and interprets according to the FreqEZ configuration settings. The Windows FreqEZ Console creates command strings in XML syntax that are sent in UDP packets to Raspberry Pi FreqEZ Controllers, enabling any combination of up to 64 devices (antenna switches, band filters, receive antennas, amplifiers, transverters…).

Key Features

- Inexpensive, off-the-shelf, readily available hardware
- All configuration is via software settings. No DIP switches, diodes, solder-bridges
- Manual antenna selection with simple mouse clicks in the Windows FreqEZ 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
        - Responds to antenna activation from the N1MM {auxantsel nn} macro
    - In DXLab, Logger32, DXLog, UcxLog, Log4OM, TR4W and N1MM+ (and perhaps other loggers) - use the Tx radio frequency from the UDP RadioInfo packet
    - Auto-magic switching (no re-configuration) from N1MM's antenna numbers to other logging programs' Tx Frequencies. No changes necessary when switching between you contest logging program (N1MM, of course!) and your general purpose logging program
    - Hard-wired method, for logging programs that do not send <RadioInfo> UDP broadcasts
        - Connect to the transceiver's BCD output jack
        - OR… use BCD signals from a PC LPT parallel port (supported by some logging programs) {ugh!}
- SO1V, SO2V or SO2R profiles define different antennas for Radio1 versus Radio2 with N1MM+ and other loggers that adhere to <RadioInfo> UDP packet definitions
- The Windows FreqEZ Console and the Raspberry Pi FreqEZ Controllers communicate via WiFi or wired Ethernet
    - Each Raspberry Pi FreqEZ Controller operates "headless" – can be located anywhere
- FreqEZ supports a maximum of four Raspberry Pi Controllers (64 total outputs)
    - The FreqEZ Controller HAT boards use bi-directional opto-isolated MOSFET solid-state devices which can each handle 30 volts AC/DC at 500 milliamperes
    - The FreqEZ Do-It-Yourself (DIY) Controllers provide "dry contacts" that sink (ground) or source (+12 vdc) in any combination from SainSmart relay boards
- PTT / Tx Inhibit input prevents the FreqEZ Controller from switching antennas while transmitting
- Flexible, software-driven configuration for shack-specific customization
- Network operation for Multi-Multi stations with an enforced *First Come, First Served* assignment protocol



Required Materials

- Raspberry Pi 4 2gb or Pi Zero 2W recommended, but runs on any Raspberry Pi model 3 or 4
- MicroSD card Class 10, at least 8gb, 16gb recommended
- Case for the Raspberry Pi (see text)
- Power Supply or USB cable with connector that matches your model Raspberry Pi
- FreqEZ RPi HAT Controller Board or DIY Controller Assembly (see text)
- Mouse, Keyboard and Monitor for configuring the Raspberry Pi (should only be needed for WiFi access)

Helpful Freeware Applications

Although not required, you may find the following free Windows programs helpful in installing and troubleshooting FreqEZ.

- **Raspberry Pi Imager** for creating the Raspbian operating system image on the microSD card https://www.raspberrypi.com/news/raspberry-pi-imager-imaging-utility/
- **WinSCP** for copying FreqEZ RPI Controller files from Windows to the Raspberry Pi https://winscp.net/eng/index.php
- **Putty** telnet client for connecting from Windows to a Raspberry Pi terminal session https://www.putty.org/

The Raspberry Pi Controller

Preparing the microSD Card

The following can be accomplished entirely from your Windows PC and is the easiest method for preparing your FreqEZ Controller. An alternative "manual" method that requires direct interaction with the Raspberry Pi is described in the Appendix: "Alternative Method microSD Card Preparation."

Download the FreqEZ image file from www.freqez.com >Downloads:
`FreqEZ_RPiController_v220.img.gz`

Download and Install the Raspberry Pi Disk Imager utility program:
https://www.raspberrypi.com/news/raspberry-pi-imager-imaging-utility/

Running the Raspberry Pi Image program:

- Insert a 16gb (or larger) microSD card into the PC's card reader
- Launch the Raspberry Pi Image application
- Select >Operating System, and scroll down the list to >Use Custom. Locate the downloaded `FreqEZ_RPiController` image file
- Select >Choose Storage – and choose the microSD card
- Select Configure (gear) Icon. Unless changes are needed [see **Note**], do not select any Customization Options.
- Click [Save]
- Click [Write]. Burning the image file to disk should take 5 – 10 minutes
- When the disk burning process finishes, remove the microSD card from your PC. Close the Disk Imager utility

NOTE: The downloaded FreqEZ Image file is pre-configured for: hardwired ethernet connection, Hostname "ezcontroller-1", Login ID "ez", Login Password "freqez", Location "New York" and Keyboard "us". If needed, change these in the Customization Options dialog window. For example, if you have more than one RPi HAT board, you should use unique Hostnames for each: ezcontroller-1, -2, -3 and -4.

Booting the Raspberry Pi with the new microSD card:

- Unplug the power cable from the Raspberry Pi
- Attach the FreqEZ HAT interface to the Raspberry Pi. Use spacers as necessary to prevent the HAT from touching any components on the Raspberry Pi circuit board.
- Connect the network cable to the Raspberry Pi RJ-45 jack
- Insert the microSD card in the Raspberry Pi microSD slot
- Boot the Raspberry Pi by connecting the power supply cable
- The Raspberry Pi will boot and auto-launch the `freqezrpi` program. Be patient - the initial boot of a new microSD card will take several minutes
- Done! The `freqezrpi` program will broadcast (on address 255.255.255.255, port 13066) its credentials in <EZRPiBeacon> UDP packets. The Windows FreqEZ Console listens for and displays these packets.

At this point, FreqEZRPi is running on your Raspberry Pi and no further action is necessary. Refer to Windows "FreqEZ Console Setup – RPi Controller Settings" for configuration instructions.

Optional Adjustments to the FreqEZrpi Controller Software

IMPORTANT: At this point, your Raspberry Pi Operating System and FreqEZ Controller installations are complete. You can reboot the Raspberry Pi, and `freqezrpi` will auto-start, broadcast its credentials on the network, and wait for packets from the Windows FreqEZ Console. **You should never need to the following instructions for editing the configuration `freqezconfig.xml` file on the RPi - unless you change the IP port number or need to do some advanced troubleshooting**.

Editing the Raspberry Pi `freqezconfig.xml` Configuration File

Connect to a command line interface (CLI) on the Raspberry from either a Windows telnet client or from a Raspberry Pi terminal session.

- From a Windows telnet client: Download and install the free Putty telnet client program (www.putty.org) and connect to the Raspberry Pi in an SSH session
- OR - From the Raspberry Pi telnet session: Connect a monitor, mouse and keyboard to the Raspberry Pi and launch the terminal app.

enter the following commands:

IMPORTANT – WHEN EDITING THE freqezconfig.xml FILE

Adhere to the UPPER and lower CaSe ChAraCteRs as shown in this documentation's examples. `freqezrpi` will not recognize statements with incorrect XML case.

After logging into the Raspberry Pi command line interface, use a simple text editor to open the config file

```
sudo nano /usr/bin/freqezconfig.xml
```

Make changes as necessary to `freqezconfig.xml`

```
<FreqEZconfig>
      <StationName></StationName>
      <RPiPort>13066</RPiPort>
      <FreqEZverbose>OFF</FreqEZverbose>
</FreqEZconfig>
```

Save your changes and close the editor by typing **<ctrl>X Y <return>**

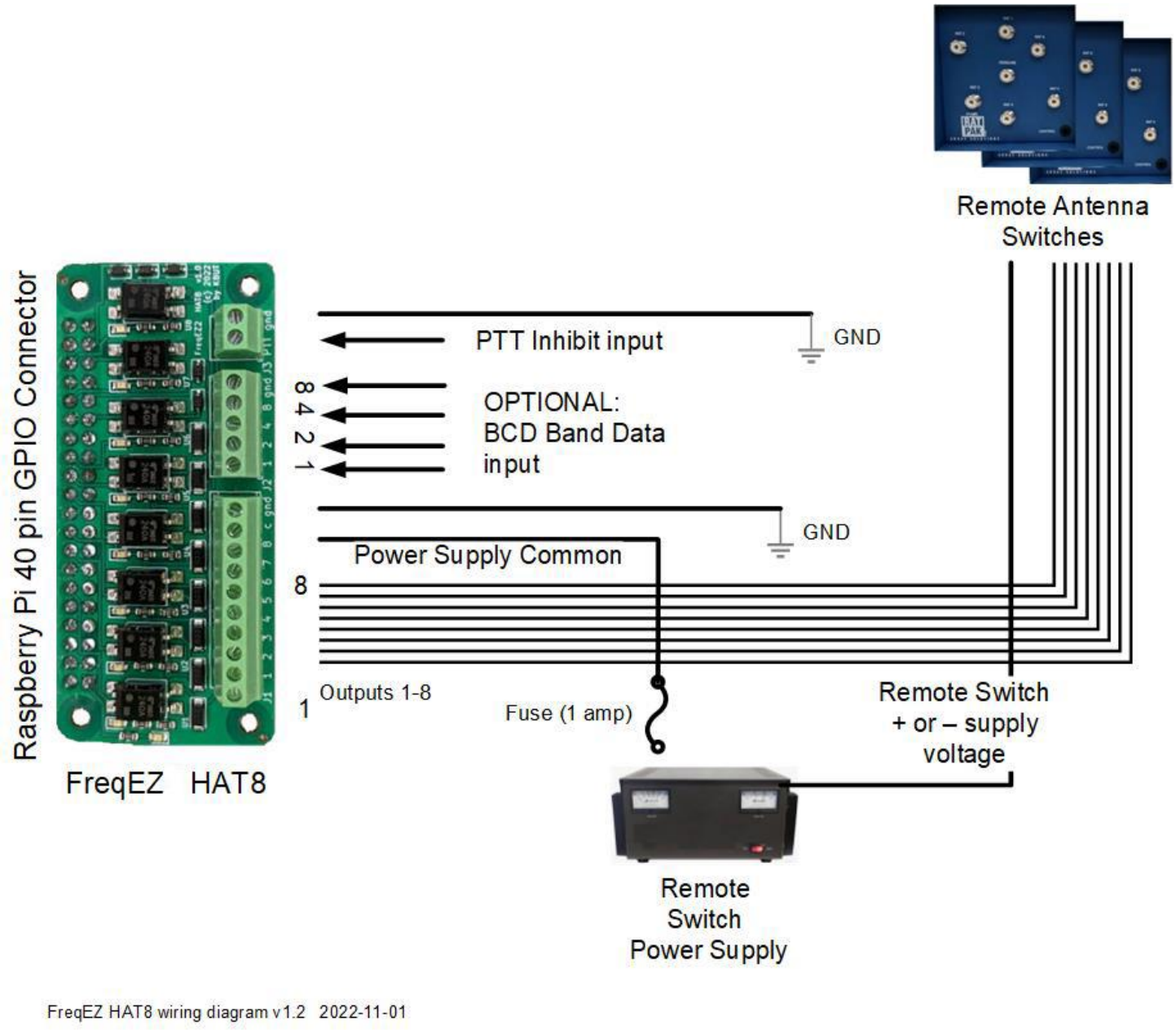
- **StationName is normally empty**. StationName is an *optional* entry that instructs `freqezZrpi` to only process packets from a specific N1MM+ Console, based on its NetBIOS name. By default, this entry is empty and FreqEZ accepts valid packets from any station on the specified port on the network. 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
- **RPiPort is normally 13066**. RPiPort is the port used by the FreqEZwin Windows Console program to communicate with the Raspberry Pi Controller. Do not change RPi`Port` on the Raspberry Pi without also changing the corresponding RPiPort entry in the FreqEZ >Setup for the RPi Controllers
- **FreqEZverbose is normally OFF**. FreqEZverbose is used for troubleshooting packets coming from the Windows FreqEZwin Console computer. With FreqEZ`verbose`=ON, freqezrpi prints the entire packet to the Raspberry Pi's terminal window as each packet is processed. To view these verbose packets, you must be running a terminal session of `freqezrpi` on the Raspberry Pi.

RPi Controller Final Assembly - FreqEZrpi HATs with 8 or 16 Outputs

Refer to the following wiring diagram when attaching your remote antenna switches.

IMPORTANT: Use a low value (1 amp or even smaller?) fuse in the supply line from the power supply. A short circuit in the system may damage the opto-MOSFETs or destroy the circuit board traces!

FreqEZ RPi HAT8 (Hardware Attached on Top) Controller Board



FreqEZ HAT8 wiring diagram v1.2 2022-11-01

Assemble a FreqEZ HAT8 Board, a Raspberry Pi, and a Case

The FreqEZ HAT8 board's dimensions align with the Raspberry Pi Zero. Solder a 40 pin male GPIO connector on the top of the RPi Zero. Insert the FreqEZ HAT8 board with 11mm standoffs onto the RPi Zero.

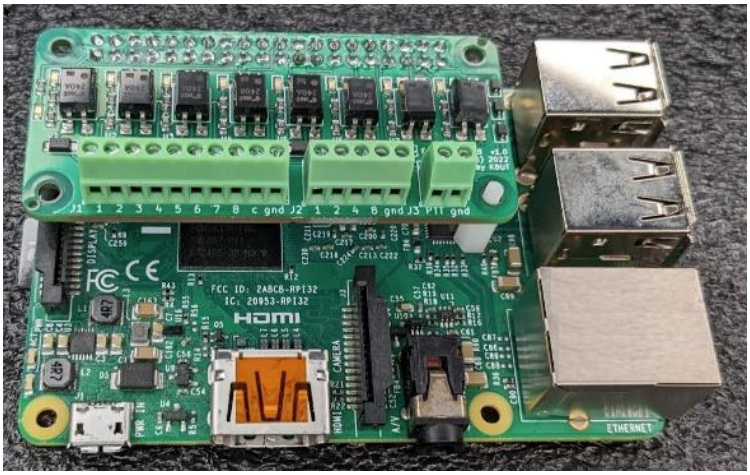
There are not many cases available for the RPi Zero, none of them metal. You will need a case of sufficient height for the FreqEZ HAT8 board and its screw terminal connections. Here are photos of three versions of an inexpensive UniPiCase in which the HAT8 fits:

There is a slot on the rear of the UniPiCase case for flat ribbon cables, or purchase a blank front panel, a panel with round holes or a case with rectanglular openings.

https://www.pishop.us/product/unipicase-zero-tall/

With some modifications, (mostly drilling mounting holes and access ports) you could mount the Raspberry PI Zero in a tall Raspberry Pi 4 case (but not the Argon Neo) listed elsewhere in this manual.

This photo depicts a FreqEZ HAT8 board attached to a Raspberry Pi 3. Note the nylon (non-conductive!) m2.5x11m spacer supporting the front of the HAT.

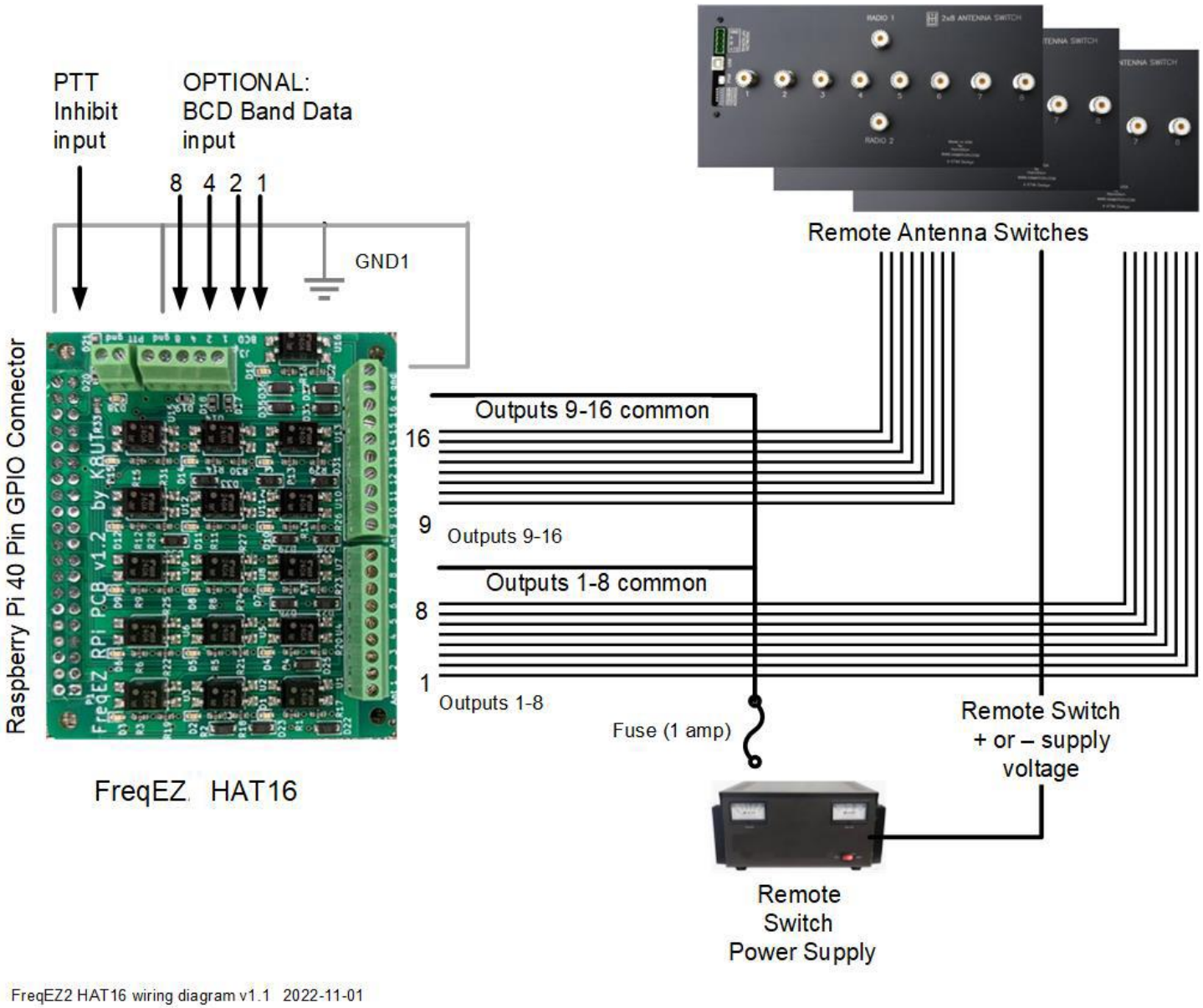


RPi Controller Final Assembly - A FreqEZrpi HAT with 16 Outputs

Refer to the following wiring diagram when attaching your remote antenna switches.

IMPORTANT: Use a low value (1 amp or even smaller?) fuse in the supply line from the power supply. A short circuit in the system may damage the opto-MOSFETs or destroy the circuit board traces!

FreqEZ RPi HAT16 (Hardware Attached on Top) Controller Board



FreqEZ2 HAT16 wiring diagram v1.1 2022-11-01

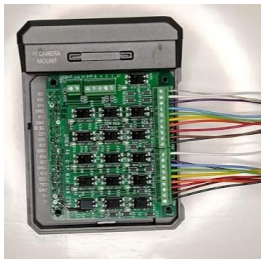
This diagram represents a simple installation in which the RPi HAT Board connects to identical remote switches with similar power supply requirements. Your installation may be more complicated. For example, you could use two dissimilar relays with dissimilar power supply requirements by feeding the Relay 1-8 Common with a different power supply and voltage than Relay 9-16 Common.

NOTE: The independent common leads from each bank of relays (1-8 and 9-16) is intentional - allowing separate sources and polarities for dissimilar Remote Antenna switches. If your remote switches use a common supply source, install a jumper to tie the "1-8 common" to the "9-16 common." (See diagram)

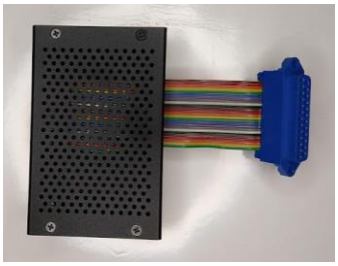
Assemble a FreqEZ HAT16 Board, a Raspberry Pi, and a Case

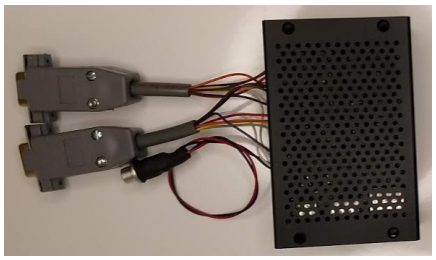
There are dozens of cases available for the RPi-4. You will need a case of sufficient height for the FreqEZ HAT board and its screw terminal connections. Here are few example cases that will fit:

Argon Neo – Inexpensive, small, metal, but plastic bottom for WiFi reception. Requires minor nibbling/filing for wire exit from case

GeekPi (short) - Inexpensive, small, metal. Requires no modification to case for wire or flat cable exit from case





GeekPi (Tall) – Inexpensive, metal. Room on case ends for connectors (2 x DB9, relay power, PTT inhibit)


If you use a different case, take a photo for us!

The Windows Console

Install the FreqEZ Program on the Windows PC

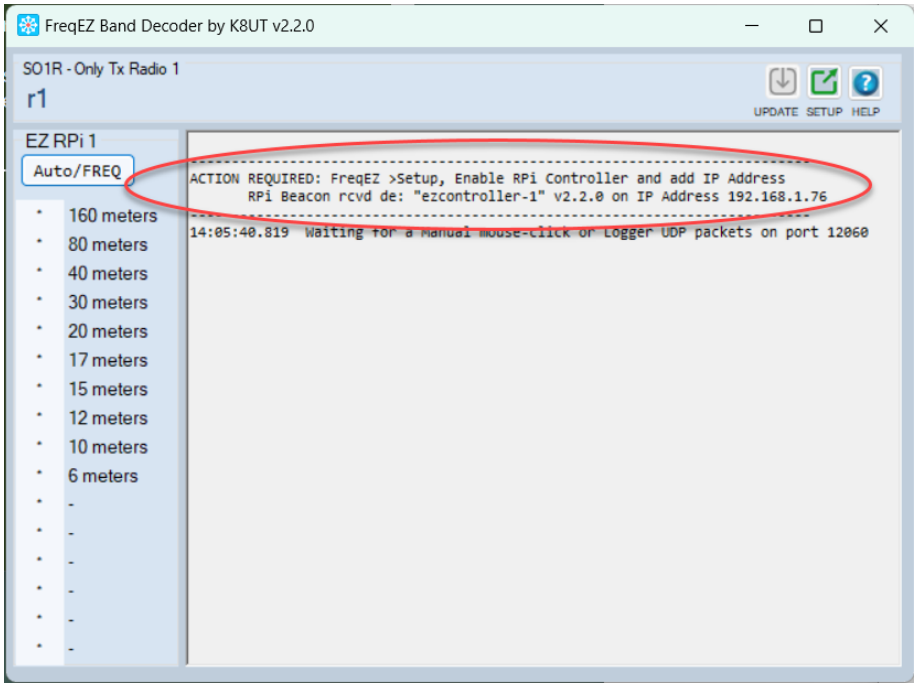
Using your favorite browser, download the current version of `setup`*`FreqEZwin_Console.exe`* from the FreqEZ >Downloads page on the www.freqez.info website.

Execute the *`setupFreqEZwin_Console.exe`* setup program. Choose an install location NOT in the reserved C:\Program Files directory tree.

The setup program will create a FreqEZ directory with the following files: the FreqEZ executable **FreqEZwin.exe**, and this **FreqEZhelp.pdf** file. The setup program can also create a FreqEZ shortcut icon on your Windows desktop.

When you launch the FreqEZwin Console the first time, the program will automatically create the *FreqEZconfig.xml* and *FreqEZsettings.xml* files with some default settings. These files will need some adjusting. Press the Setup icon to open the FreqEZ Setup dialog window.

The FreqEZ Windows Console



The FreqEZ Windows Console listens for broadcast packets from orphaned FreqEZ RPi Controllers, as shown in this screenshot. The Windows Console will display the Controller's credentials until you enable the Controller and configure its IP Address in the Console's >Setup dialog (below).

FreqEZ Windows Console Setup

NOTE: EDITING FreqEZconfig.xml or FreqEZsettings.xml outside of the FreqEZ program:

All of the settings in FreqEZ can be adjusted using its >`Setup` and >`Config` menu options. However, you can edit the configuration files with a standard text editor. When opening these XML file with File Explorer, the *file associations* in Windows may not launch a text editor - rendering a fractured view of its contents. Instead, right-click on the XML filename and select ***>Open with*** to choose your favorite text editor (Notepad++?) or the Windows default text editor Notepad.exe (ugh).

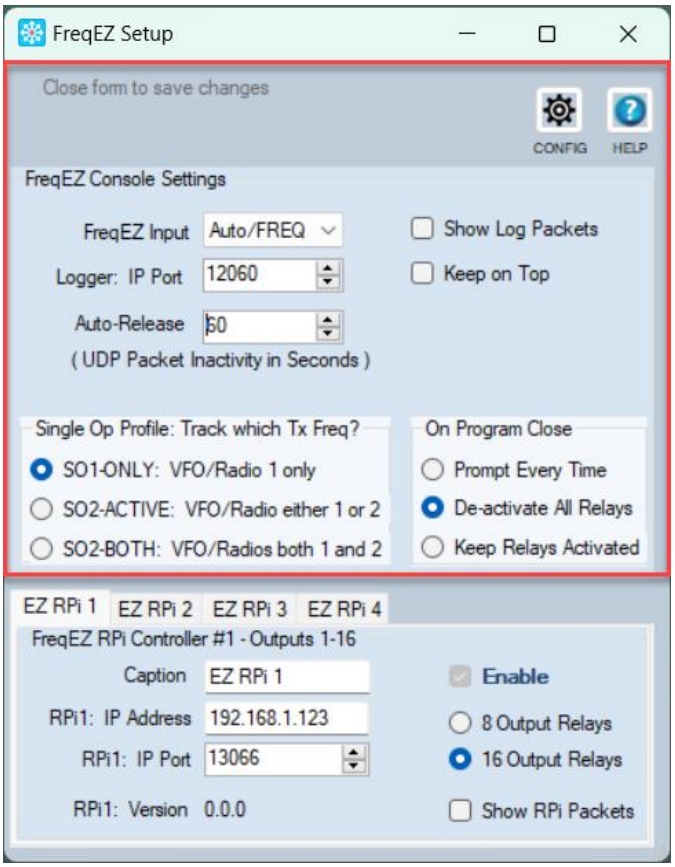
Important: If you use an independent text editor, do not edit these files when FreqEZ is running. Any changes you make will be overwritten by FreqEZ when it closes.

FreqEZ Console Setup Options

All of the settings in FreqEZ are controlled from the FreqEZ `>Setup` and `>Config` dialog windows.

Config Button: Opens the editor for making output assignments (see "Editing the FreqEZConfig.xml file" below)

Help Button: Opens this pdf file, FreqEZhelp.PDF, on the local disk drive.



FreqEZ Console Option Descriptions

- **Band Decoder Input:** Select from the four FreqEZ input choices
  - **Auto/FREQ** activates outputs based on the <TxFreq> tag in the <RadioInfo> UDP packet
  - **Auto/N1MM** activates outputs based on the <Antenna> tag (0-63) in the UDP <RadioInfo> packet
  - **Auto/BCD** activates outputs based on the 4 wire BCD table for your radio or logging program
  - **Manual** activates outputs based on clicking the output buttons in the FreqEZ Console
- **IP Port:** The UDP port for listening to <RadioInfo> packets from the logging program. Do not change this value without also changing the corresponding entry in your logging program
- **Auto-Release:** An inactivity timer for multi-station configurations with two or more networked Windows Consoles controlling FreqEZ outputs. The timer starts when an RPi Controller stops receiving packets from the Console. When the timer stops the RPi Controller releases that Console's assignments so that they can be available to other Consoles on the network. Auto-Release prevents output "hostage" conditions when Consoles or network connections fail. Default release time is 60 seconds. Auto-Release time can be set between 21 and 3600 seconds (an hour).
- **Show Log Packets:** Checkbox to display UDP <RadioInfo> packets from the logging program in the packet window. Useful for troubleshooting
- **Keep on Top:** Checkbox to keep the FreqEZ Console display above other programs in Windows

Single Op Profile Setting

SO1-ONLY: For SO1R (Single Operator One Radio) operation. FreqEZ only responds to packets tagged as <RadioNr>1</RadioNr> (Radio 1) from the logging program. In SO1-Only operation output activations are derived from the <FREQSOURCE>, <N1MMSource>, or <BCDSource> sections of >Setup >Config XML file. Do not set Single-Op Profile to SO1-Only if you plan to operate SO2V or SO2R, because Radio 2 packets will be ignored.

- **SO2-ACTIVE:** For SO2R (Single Operator Two Radios) or SO2V (Single Operator Two VFOs) operation. FreqEZ responds to packets for whichever radio number (R1 or R2) is listed as the <ActiveRadioNr> in packets from the N1MM logging program, or as the <RadioNr> in packets from DXLab Commander (and other general purpose loggers). In N1MM, the ActiveRadioNr can be identified by the Red dot in N1MM's Entry Window, or in DXLab Commander's MultiRadio R1, R2 buttons. Other general-purpose loggers that lack SO2V/R support will always consider Radio 1 as the <ActiveRadioNr>.
- **SO2-BOTH:** For SO2R (Single Operator Two Radios) or (SO2V (Single Operator Two VFOs) operation. FreqEZ responds to packets for both Radio 1 and Radio 2 from the logging program. This profile activates outputs to both VFOs/Radios simultaneously, ignoring which Radio/VFO might be the current Active Radio.

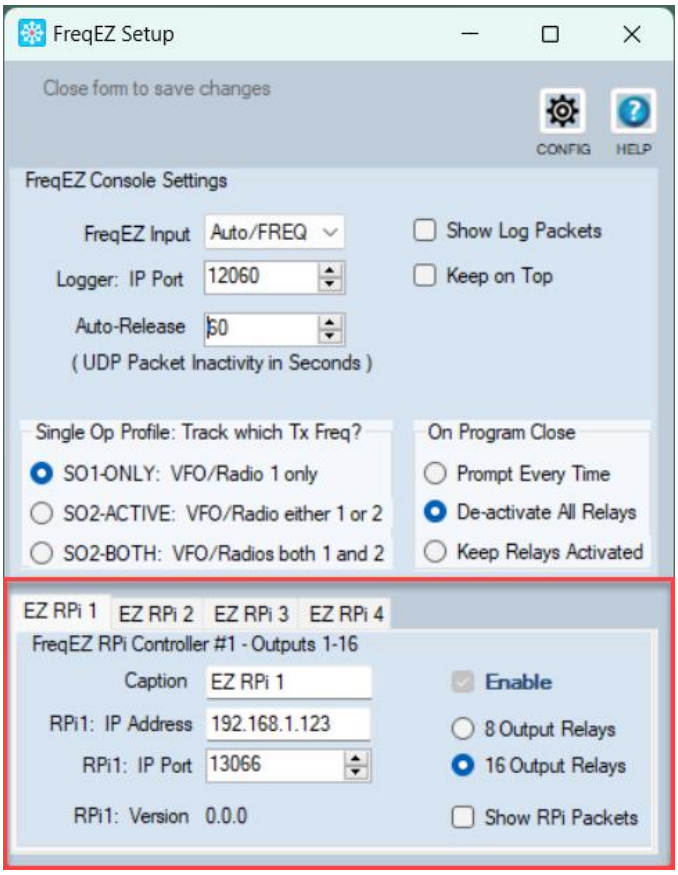
  NOTE: In the SO2-Active and SO2-Both operation, output activations are derived from the derived from the <FREQSOURCE>, <N1MMSource>, or <BCDSource> sections of the >Setup >Config XML file when referring to Radio 1, and to the <FREQ2SOURCE></FREQ2Source>, <N1MM2Source>, or <BCD2Source> sections section when referring to Radio 2. You will receive an error message if the <…Source> and <…2Source> sections do not contain entries for each band/frequency.

On Program Close: Controls the activation / deactivation of this Console's FreqEZ HAT outputs upon program close. If applicable, other network station assignments will be unaffected.

NOTE: Any changes in >Setup will be automatically saved upon closing the FreqEZ Setup window

FreqEZ Console Setup – RPi Controller Settings

The four tabs across the bottom of the Setup window set configuration options for the four Controllers.



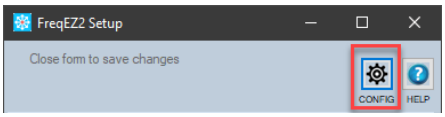
Tabs 1 through 4: Configure the RPi Controllers

- **Caption:** A user-friendly name for each Raspberry Pi Controller
- **IP Address:** This RPi Controller's TCP/IP address. Example 192.168.1.123 The RPi Controller's IP address can be found in the "Action Required" statement on the Console's packet display. See previous section on Windows Console.
- **IP Port:** This Controller's IP port. A number between 1024 and 65535. The default FreqEZ port for Console -to- Controller traffic is 13066. All Controllers use the number assigned to RPi #1. Changing RPi Controller #1 will change the port for the others.
- **RPi Version:** Current software version running on this Controller

- **Enable:** Checkbox to activate packets to this Raspberry Pi Controller. Enabled Controllers will appear in the Windows FreqEZ Console packet window. Disabled Controllers do not receive packets from the Console
- **8 or 16 Output Relays:** Adjust the height of the Console display for HAT 8 or HAT 16 Controller boards
- **Show RPi Packets:** Checkbox to display packets to and from this Raspberry Pi Controller in the Console packet window. Useful for troubleshooting

The FreqEZ Configuration Editor

Pressing the >Config button in the FreqEZ Console's Setup window opens a text editor containing the activation settings for the Raspberry Pi Controllers.



IMPORTANT - FreqEZ Output Numbering: In the FreqEZConfig.XML file, activated outputs are numbered sequentially from 1-64. RPi Controller #1 = 1-16; RPi Controller #2 = 17-32; RPi Controller #3 = 33-48; RPi Controller #4 = 49-64

Auto/FREQ: Activate Outputs Based on Radio's Tx Frequency

FreqEZ activates outputs based on the <TxFreq> tag in the <RadioInfo> packet that it receives from loggers that support <RadioInfo> XML definitions in UDP packets. Upon receipt of a new packet, FreqEZ examines the <FREQSource> section of the FreqEZconfig.xml file and activates outputs based on the first entry that falls within the specified frequencies. FreqEZ supports a maximum of 254 <FREQ> assignments, from <FREQ1> to <FREQ254>.

FREQSource syntax: <SourceNr> **StartFreq[DASH]EndFreq [COMMA] EnableOutputNr(s)** </SourceNr>

Where: SourceNr is a sequential number from 1 – 254, EnableOutputNr is one or more RPi HAT relay output numbers from 1-64, separated by commas.

Auto/FREQ example 1: A simple list of antennas with separate outputs for each band/frequency. From this list, a packet with 2125500 as the <TxFreq> would activate output #4.

```
<FREQSource>
        <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 <FREQ254> entries
</FREQSource>
```

Auto/FREQ example 2: A multi-band beam fed with one coax (e.g. Steppir 20 to 6 meters). From this list, a packet with 2128500 as the <TxFreq> frequency would activate output #1. In this example, output #1 covers all frequencies from 20 to 6 meters.

```
<FREQSource>
        <FREQ1>1400000-5000000,1</FREQ1>
        <FREQ2>1000000-1015000,2</FREQ2>
        <FREQ3>700000-750000,3</FREQ3>
        <FREQ4>350000-400000,4</FREQ4>
        <FREQ5>180000-200000,5</FREQ5>
</FREQSource>
```

Auto/FREQ example 3: A list with two dual-band ("fan") dipoles for 30 & 40, 80 & 160. From this list, a packet with 1012500 as the <TxFreq> frequency would activate output #7. Any frequency on 30 or 40 meters would activate output #7. Any frequency on 80 or 160 meters would activate output #8.

```
<FREQSource>
        <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>
</FREQSource>
```

Auto/FREQ example 4: Activate several relays simultaneously on multiple Raspberry Pi Controllers. Suppose you wanted to activate a remote antenna switch, a receive antenna relay, and a different bandpass filter for every band? From this list, a packet with 715100 as the <TxFreq> frequency would activate outputs 2 and 6 on RPi Controller #1 and outputs 2 (=18) and 16 (=32) on RPi Controller #2.

```
<FREQSource>
        <FREQ1>350000-400000,1,5,17,31</FREQ1>
        <FREQ2>7000000-740000,2,6,18,32 </FREQ2>
        <FREQ3>1010000-1015000,1,8,15</FREQ3>
</FREQSource>
```

Auto/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 this example, a packet for the phone portion with 390300 as the <TxFreq> frequency would activate output #5. QSY'ing to the CW portion of the 80 meter band would activate output #4.

```
<FREQSource>
        <FREQ1>180000-200000,3</FREQ1>
        <FREQ2>350000-370000,4</FREQ2>
        <FREQ3>370000-400000,5</FREQ3>
        <FREQ4>700000-740000,6</FREQ4>
        <FREQ5>1010000-1015000,7</FREQ5>
        . . . up to <FREQ254> entries
</FREQSource>
```

Auto/FREQ example 6: If your remote antenna switch or band filter requires a BCD signal rather than individual wires for each relay, you can use FreqEZ's "multiple relay" feature to impersonate a BCD signal. Refer to the BCD table supplied by your device manufacturer. Here is a sample Band -to- BCD table from a filter manufacturer:

Band, MHz	Pin No.1	Pin No.2	Pin No.3	Pin No.4
1.8	1	0	0	0
3.5	0	1	0	0
7	1	1	0	0
10	0	0	1	0
14	1	0	1	0
18	0	1	1	0
21	1	1	1	0
24	0	0	0	1
28	1	0	0	1

To match those BCD outputs based on frequency, set the XML configuration file as follows:

```
<FREQSource>
        <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>2800000-2850000,1,4</FREQ9>
</FREQSource>
```

Auto/FREQ SO2R: Activate Outputs Based on Radio Frequencies from Radios 1 and 2

A list of antennas for Radio2 when using a Single Op Profile of either SO2-Active or SO2-Both. **This designation requires that you insert <FREQ2Source> definitions for every anticipated frequency – essentially matching every entry in your <FREQSource> stanza**. From the following list, a Console packet from Radio2 with 2125500 as the <TxFreq> would activate output 2 on RPi #1, 18 on RPi #2 and 51 on RPi #4.

```
        <FREQSource>
                <FREQ1>5000000-5400000,1</FREQ1>
                <FREQ2>2800000-3000000,33,49/FREQ2>
                blah blah blah
                . . . up to <FREQ254> entries
        </FREQSource>
        <FREQ2Source>
                <FREQ1>5000000-5400000,2</FREQ1>
                <FREQ2>2800000-3000000,17,50/FREQ2>
                <FREQ3>2400000-2600000,34/FREQ3>
                <FREQ4>2100000-2145000,2,18,51</FREQ4>
                . . . up to <FREQ254> entries
        </FREQ2Source>
```

Refer to the earlier <FREQSource> examples for other descriptions that can be inserted within the <FREQ**2**Source> stanza for SO2R operation.

Auto/N1MM: Select Outputs Based on N1MM+ Antenna Settings

FreqEZ makes 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 <N1MMSource> section of the FreqEZconfig.xml file and activates outputs based on an exact match between antenna number and the N1MMSource tag.

N1MMSource syntax: `<N1MMAntennaNr>` **`EnableOutputNr(s)`** `</N1MMAntennaNr>`

Where: N1MMSourceNr is the number from the N1MM's Configurer's Antenna table from 1 – 64, EnableOutputNr is one or more HAT output numbers from 1-64, separated by commas.

Auto/N1MM example 1: A simple list of 9 antennas with multiple outputs for each N1MM antenna. From this list, a packet with Antenna #5 would activate output 1 (=17) on RPi Controller #2.

```
<N1MMSource>
        <N1MM0>1,12</N1MM0>
        <N1MM1>2,13</N1MM1>
        <N1MM2>3,14</N1MM2>
        <N1MM3>4,15</N1MM3>
        <N1MM4>5,16</N1MM4>
        <N1MM5>17</N1MM5>
        <N1MM6>7,18</N1MM6>
        <N1MM7>8,18</N1MM7>
        <N1MM8>9,19</N1MM8>
        . . . up to <N1MM64> entries (max number in the N1MM+ Configurer)
</N1MMSource>
```

Auto/N1MM example 2: Similar to the FREQ examples, a single antenna number from N1MM can activate multiple relays in FreqEZ. From this list, a packet with Antenna #4 would activate output 4 on RPi Controller #1, output 2 (=18) on RPi Controller #2, and output 3 (=35) on RPi Controller #3.

```
<N1MMSource>

        <N1MM0>1</N1MM0>
        <N1MM1>1,9,11</N1MM1>
        <N1MM2>2,6,12</N1MM2>
        <N1MM3>3,6,13</N1MM3>
        <N1MM4>4,18,35</N1MM4>
        <N1MM5>5,6,15</N1MM5>
        <N1MM6>6</N1MM6>
        <N1MM7>7</N1MM7>
        <N1MM8>8</N1MM8>

</N1MMSource>
```

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

Auto/N1MM SO2R: Select Outputs Based on N1MM+ Antenna Settings for Radios 1 and 2

A list of antennas for Radio2 when using a Single Op Profile of either SO2-Active or SO2-Both. **This designation requires that you insert <N1MM2Source> definitions for every anticipated antenna number – essentially matching every entry in your <N1MMSource> stanza**. From the following list of 9 antennas with multiple outputs for each N1MM antenna. A packet with Antenna #5 would activate outputs 6 on RPi Controller #1 and outputs 1 (=17) on RPi Controller #2.

```
<N1MMSource>
        <N1MM0>1,12</N1MM0>
        <N1MM1>2,13</N1MM1>
        <N1MM2>3,14</N1MM2>
        . . . up to <N1MM64> entries (max number in the N1MM+ Configurer)
</N1MMSource><N1MMSource>
<N1MM2Source>
        <N1MM0>2,12</N1MM0>
        <N1MM1>2,13</N1MM1>
        <N1MM2>3,14</N1MM2>
```

```
        <N1MM4>5,16</N1MM4>
        <N1MM5>6,17</N1MM5>
        <N1MM6>7,18</N1MM6>
        <N1MM7>8,18</N1MM7>
        <N1MM8>9,19</N1MM8>
        . . . up to <N1MM64> entries (max number in the N1MM+ Configurer)
</N1MM2Source>
```

Refer to the earlier <N1MMSource> examples for other descriptions that can be inserted within the <N1MM**2**Source> stanza for SO2R operation.

NOTE for using Band Decoder Input = Auto/N1MM:

If N1MM+ is your contesting program, and your general-purpose logging program, FreqEZ will auto matically adjust between Auto/N1MM and Auto/FREQ modes when you change from contesting to general purpose logging. Select Auto/N1MM and configure tables for both sources (FREQSource and N1MMSource). FreqEZ will activate relays based on N1MM antenna numbers only when the UDP packet <app> = "N1MM." UDP Packets from any other logger will be treated as if your Band Decoder Input were configured for Auto/FREQ (frequency based antenna selection). This feature allows you to switch back-and-forth between your contesting program and your general logging program without reconfiguration.

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.

Auto/BCD: Select Antennas Based on the Radio's Band Output Pins

FreqEZ makes antenna selections based on the four wires from your radio's BCD antenna output to four inputs on the Raspberry Pi Controller's GPIO connector. The Raspberry Pi Controller continuously monitors those four inputs and sends the results to the Windows Console in the UDP packet <BCDValue > tag. The Windows FreqEZ Console examines the <BCDSource> section of the FreqEZconfig.xml file and activates relays based on BCD values (0-15). Refer to your radio's documentation for Band -to- BCD values.

Regardless of the input source being used (Auto/FREQ, Auto/N1MM, or Auto/BCD) each FreqEZ RPi HAT controller constantly monitors the status of the four BCD input pins and reports back to the Windows Console. In >Setup, enable "Show RPi Packets" to see the current BCD value. If there are no connections to the BCD input pins - all four pins floating high - the BCD value will always be 15 (1111). Once connected, changing bands on your radio should instantly (just a few milliseconds) be reflected in a new BCD value being sent from the RPi Controller to the Windows Console.

To troubleshoot BCD issues, enable the RPi Controller's *Show Packets* option to see the BCD value being sent to the Windows Console from the RPi Controller.

BCDSource syntax: `<BCD Value>` **`EnableOutputNr(s)`** `</BCD Value>`

Where: BCD Value is the number from 0 -15 derived from the RPi HAT BCD inputs, EnableOutputNr is one or more HAT output numbers from 1-64, separated by commas.

Auto/BCD Example: A BCD value of 6 from the radio would activate output 5 on RPi #1, 2 on RPi #2 (18==2), and 15 on RPi #4 (63==15).

```
<BCDSource>
        <BCD0>9</BCD0>
```

```
        <BCD1>9</BCD1>
        <BCD2>9</BCD2>
        <BCD3>8</BCD3>
        <BCD4>8</BCD4>
        <BCD5>6</BCD5>
        <BCD6>5,18,63</BCD6>
        <BCD7>4</BCD7>
        <BCD8>3</BCD8>
        <BCD9>2</BCD9>
        <BCD10>10</BCD10>
        <BCD12>1,2,3</BCD12>
</BCDSource>
```

Auto/BCD SO2R: Select Outputs Based on BCD Values from Radios 1 and 2

SO2R from BCD inputs – By default, BCD input is derived from the BCD values sent from RPi Controller #1 to the Windows Console. However, for SO2R operation, FreqEZ can include the BCD values sent from a second BCD source – Radio2. SO2R on BCD requires a second RPi and FreqEZ HAT board be connected to a second radio, but provides a capability not normally available to BCD wired stations. See the SO2R section later in this documentation.

A list of antennas for Radio2 when using a Single Op Profile of either SO2-Active or SO2-Both. This designation will require that you insert <BCD2Source> definitions for every anticipated BCD number – essentially matching every entry in your <BCDSource> stanza.

Auto/BCD SO2R Example: A BCD value of 6 from Radio1 would activate output 5 on RPi #1, 2 on RPi#2 (18==2), and 15 on RPi #4 (63==15).

```
<BCDSource>
        <BCD0>9</BCD0>
        <BCD1>8</BCD1>
        <BCD2>7</BCD2>
        Blah blahblah
        …to max BCD value 15
</BCDSource>
<BCD2Source>
        <BCD0>10</BCD0>
        <BCD1>11</BCD1>
        <BCD6>5,18,63</BCD6>
        <BCD7>4</BCD7>
        <BCD8>3</BCD8>
        <BCD9>2</BCD9>
        …to max BCD value 15
</BCD2Source>
```

Refer to the earlier <BCDSource> examples for other descriptions that can be inserted within the <BCD**2**Source> stanza for SO2R operation.

Manual: Select Outputs by Clicking on Output Buttons in the FreqEZ Console

If you have no direction connection to radio frequency information from other sources, you can place FreqEZ in permanent Manual mode by choosing Manual Band Decoder Input on the FreqEZ Setup screen. Otherwise, you can place individual FreqEZ Controllers in Manual mode by clicking the Auto - Manual button at the top of each Output Panel, toggling that FreqEZ Controller's mode between your configured input (Auto/BCD, Auto/FREQ or Auto/N1MM) and Manual operation. When in Manual operation, special functions like Macros and Constraints will continue working properly, but FreqEZ ignores input from FREQ, N1MM, or BCD sources.

To activate an output relay, click on an Output Button in the Output Panel. The click will: automatically place that Controller in Manual mode; activation that output; and de-activate all other outputs on that Controller. The output will toggle ON/OFF with subsequent mouse clicks. That Controller remains in Manual mode until cancelled by clicking the Auto – Manual button at the top of the Output Panel.

Read more about Manual mode in the section *FreqEZ Console Controls*.

Manual Plus: Activate Multiple Outputs from a Single Output Button Click

Manual mode operation activates only one output, while deactivating any other outputs on that Controller. However, for those instances when you'd like to manually activate more than one output with a single click, use FreqEZ's Manual Plus feature.

Manual Plus's configuration is similar to other designations in the FreqEZConfig.xml file – an XML stanza containing an XML tag with a list of the output numbers to be activated.

MANUALPLUSSource syntax: `<RPiHatOutput>` **`EnableOutputNr(s)`** `</RPiHatOutput`

Where: `RPiHatOutput` is the RPi HAT output number (1-64) corresponding to the RPiNr (1-4) times the clicked band button position (1-16) in the Windows Console Band Button display, EnableOutputNr is one or more RPi HAT output numbers from 1-64, separated by commas.

MANUALPLUS Example:

```
<MANUALPLUSSource>
        <MANUAL16>1,2,18</MANUAL16>
        <MANUAL32>3,19</MANUAL32>

</MANUALPLUSSource>
```

In this example, clicking the last Output Button on RPi Controller#1 (button #16) would activate output numbers 1 and 2 on RPi Controller#1 and output 2 (#18) on RPi Contoller#2. Clicking the last Output Button on RPi Controller#2 (button #16) would activate output number 3 on RPi Controller#1 and output 3 (#19) on RPi Contoller#2.

Important details about Manual Plus:

- The first click of a Manual Plus button will deactivate all outputs on the affected RPi Controller(s) and place the Controller(s) in Manual Mode
- The second click of a Manual Plus button will activate all of the configured output numbers
- Subsequent clicks of a Manual Plus button will toggle the configured outputs ON/OFF
- Clicking on the configured Manual Plus button does not activate that output, unless you intentionally include that output number in your Manual Plus output list
- Canceling Manual Mode requires a Manual/Auto button click on the affected RPi Controller(s)
- The Manual Plus button itself cannot be configured to apply Constraints on other outputs

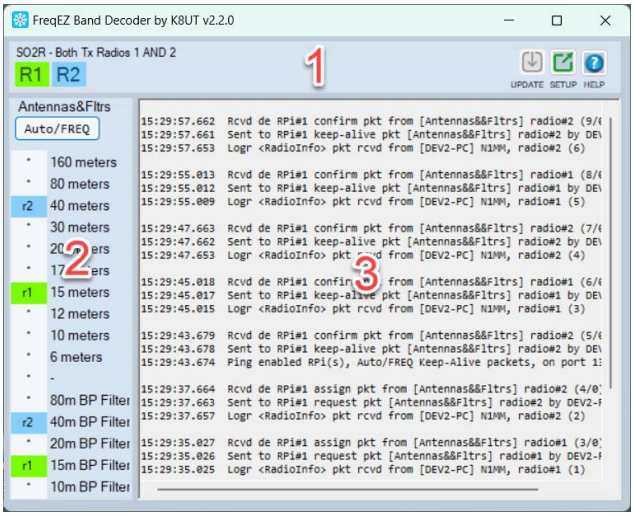
Updating FreqEZ Versions v2.2.0 and later

New Version of the FreqEZ Windows Console: When the FreqEZ console launches, it checks www.the freqez.com for a newer version of the program. If available, a notification window will open and the FreqEZ Console [Update] button will be enabled. Clicking on the [Update] button opens a Setup screen to install the new version.

New Version of the FreqEZ Raspberry Pi Controller: When the FreqEZ Windows Console is running the latest version of the program, the Console checks for a new version of the FreqEZ RPi Controller program. If available, a notification window opens and the FreqEZ Console sends a software update command to the RPi Controller. The RPi Controller installs the new version and reboots. Be patient – the update process and reboot of the RPi might take a couple of minutes to complete.

Operating FreqEZ

The FreqEZ Console Description



1. **The Icon Bar**

   - **Caption** – Current Single-Op Profile
   - **Radio 1 and Radio 2 indicators** – Show current status of each Radio/VFO
   - **Update button** – Download and install new version of FreqEZ Console software
   - **Setup button** – Open Setup dialog window
   - **Help button** – Open the FreqEZ manual if `FreqEZhelp.pdf` Help file is present

2. **Output Panels 1 - 4 (enabled Raspberry Pi Controllers)**

   - **Caption** shows friendly name of this RPi Controller
   - **Auto / Manual toggle button** places this RPi Controller in Manual or Automatic mode
   - **Output Buttons** manually select one output when clicked. For more than one output use *<ctrl>+Click* to force relays =ON or =OFF. *<ctrl>+Click* will activate the selected output without placing FreqEZ in Manual mode
   - **Output Labels** describe the 16 outputs - Click on the text to edit a label. When done, press <tab>. See also Macros later in this document
   - **TX/RX Indicator** shows red TX (not visible in this screenshot) when this RPi Controller's *PTT Inhibit* input is shorted to ground. This FreqEZ RPi Controller will ignore all inbound antenna changes when TX is displayed

   Output Button Expressions

*****	This output is off (not activated)
r1 r2	This output was activated by a band/frequency or BCD setting for radio 1 or 2
{1) {2}	This output was activated by a macro for active radio 1 {IsR1} or 2 {IsR2}
{R}	This output was activated by a receive {IsRx} macro
{T}	This output was activated by a transmit {IsTx} macro

X x	This output has been assigned to another station on the network
{J} {K}	This output was activated by a Lockout macro for Radio 1 {IsR1Lk} or radio 2 {IsR2Lk}
c	This output is disabled due to a Constraint configuration (See Constraints)
M	This output was manually activated by a mouse-click
N	This output was activated by an N1MM {auxantsel} macro
ON	This output was manually activated by a <ctrl>+mouse-click (see Forced ON)
OFF	This output was manually de-activated by a <ctrl>+mouse-click (see Forced OFF)

3. UDP Packet Display

- Shows UDP packet traffic from your logging program and/or from the Raspberry Pi Controllers. Enabled by *Show Packets* checkboxes in the >Setup window

FreqEZ Console Controls

Upon launch, the Windows Console will automatically resume operating as per your settings.

You can adjust FreqEZ's relay outputs with the following mouse-click action in the Output Panel:

The **Auto/<source> – Manual** Button

> Mouse-clicking on the Auto - Manual button toggles FreqEZ operation between your configured Source (BCD, FREQ or N1MM) and Manual operation. In Manual operation, special functions like Macros and Constraints will continue working properly, but FreqEZ will not activate outputs based on FREQ, N1MM, or BCD packets.

Mouse-Click on an output button

> Mouse-clicking on an output button activates that relay output and places the output panel in Manual operation (see Manual operation, above). A subsequent mouse-click will de-activate that relay output, but leave the output panel in Manual operation. Mouse-clicking on a different output button will activate that relay output and de-activate the previous output selection. To cancel Manual operation, click the Auto – Manual button.

<ctrl>+Mouse-Click on an output button, **Forced ON**

> <ctrl>+Mouse-clicking on an output button activates that relay output in Forced ON status without placing the output panel in Manual operation. The Forced ON output remains activated despite other changes in the band panel, and can be used as a method for manually activating multiple relay outputs. Forced ON status can be cancelled with a standard mouse-click

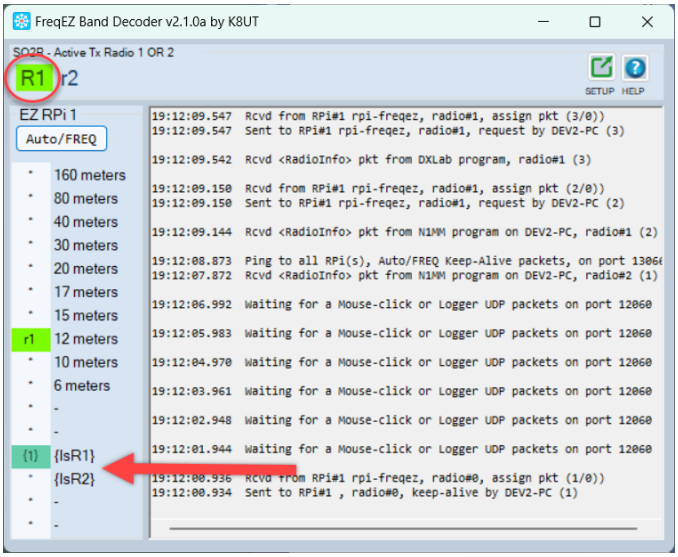
<ctrl>+Mouse-Clicking on an output button, **Forced OFF**

> <ctrl>+Mouse-clicking on an output button that is already Forced ON de-activates that relay output in Forced OFF status without placing the output panel in Manual operation. The Forced OFF output remains de-activated despite other changes in the band panel. Forced OFF status can be cancelled with a standard mouse-click

Macros - Special Output Label Expressions for SO1R and SO2R Single Op Profiles

FreqEZ macros provide automatic output switching with functions that are independent of the radio's band/frequency. Macros are defined by entering the macro as the label on an output (see screenshot). The expression must be entered exactly as shown – matching upper/lower case characters as listed below

Macro example - screenshot: {IsR1} and {IsR2} macro expressions have been entered as labels in output positions 13 and 14. In this example, Radio 1 is the active radio (red dot in the N1MM Entry Window) and the output 13 is activated and 14 is not activated. When Radio 2 is the active radio, output position 13 would not be activated and 14 would be activated.



- **{IsTx}** and **{IsRx}** macros: When inserted as output labels, activates the output when the PTT Inhibit input on the Raspberry Pi Controller is grounded {IsTx} or not {IsRx}. For example, you may want to enable a receive antenna only when Receiving, and ground that antenna when Transmitting
- **{IsR1}** and **{IsR2}** macros for SO2R operation: When inserted as output labels, activates the output based on the logging program's UDP packet <ActiveRadioNr>. This macro is used in SO2V and SO2R operation when Setup's Single Op Profile is defined as SO2-ACTIVE or SO2-BOTH. For example, you might want to operate a relay for Radio1 or Radio2 ONLY WHEN that radio has the program's Tx Focus (the Red Dot in N1mm's Entry Window). To activate an exclusive Tx Focus output, insert the macro {IsR1} or {IsR2}
- **{IsR1Lk}and {IsR2Lk}** macros: Hardware lockout relay that activates when Radio 1 {IsR1Lk} or Radio 2 {IsR2Lk} is the Active Radio and the RPi Controller is in Transmit (PTT input is grounded)

NETWORK NOTE: Macro Expressions are assigned to a station's outputs on a *First-Come, First-Served* basis. The assignment will appear on other network stations' FreqEZ Consoles as being InUse ("X").

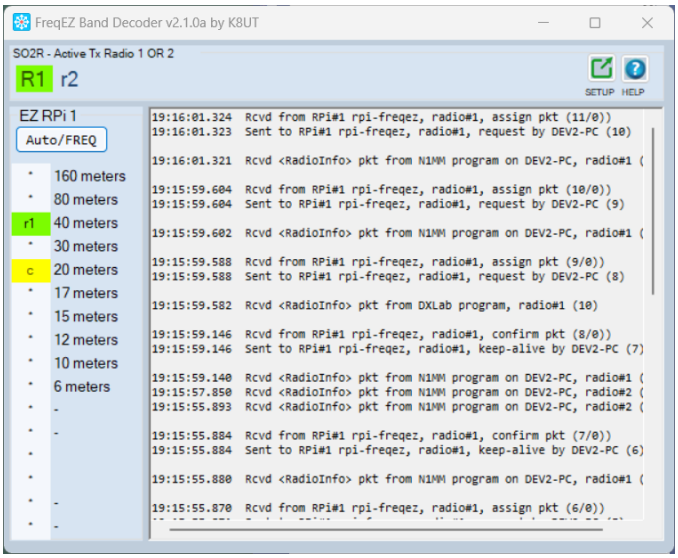
An External Macro - Using N1MM's AUXANTSEL function key macro to Activate FreqEZ Outputs

FreqEZ supports the N1MM *AUXANTSEL* and *AUXANTSELNAME* macro, and will activate the output number defined in the function key message.

To turn off an activated AUXANTSEL output, click on it – similarly to clicking on a Manually operated output.

CAUTION: Using N1MM's AUXANTSEL with networked stations may produce unexpected results when Constraints and Conflicts are involved – be sure to thoroughly test your configuration.

Constraints – Prevent Outputs from being Simultaneously Activated

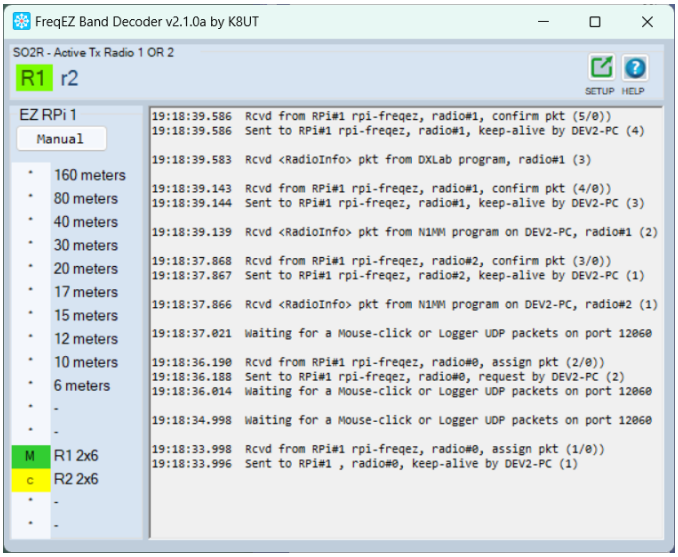


FreqEZ Constraints are configuration options that define preemptive activation rules between outputs. "If output #1 is activated, do not allow output #10 to be activated." For example, if you had 40 meter and 20 meter beams co-located on a single boom, you might want to avoid activation of the 20 meter beam when the 40 meter antenna is being used.

The Constraint definitions are found in the XML <Constraints> section of FreqEZ's >Setup >Config editor. The numbers refer to corresponding FreqEZ outputs – from output #1 in RPi Controller 1 to output #64 on RPi Controller 4. The **Constraining Output** is the XML tag expression <Constraint1> </Constraint1>, and the **Constrained Output** is the inhibited output number within the XML tags.

When a constraint definition is encountered and the Constraining Output will be activated, the Constrained Output will be disabled and displayed in yellow in the FreqEZ output panel. In the example above, a configuration entry with `<Constraint3>5</Constraint3>` activated the 40 meter beam (output #3) and disabled the 20 meter beam (output #5).

For KK1L 2x6 Antenna Switch Owners



Constraints can prevent dangerous combinations of activations that might result in the radios being cross-connected. This example and screenshot show two outputs that prevent each other from activating. If output #13 is activated, output #14 disabled. If output #14 were to be activated, output #13 would be disabled. This configuration entry from the >Setup >Config file would look like this:

```
<Constraint13>14</Constraint13>
<Constraint14>13</Constraint14>
```

Another method for N1MM Stations using 2x6 antenna switch control would configure macros {IsR1} and {IsR2} as Constraining outputs, with defined Constrained outputs to be disabled. When you switch between Radio 1 and Radio2 in N1MM, the R1/R2 macros would prevent the Constrained outputs from activating. (See screenshot in Macros section for IsR1 and IsR2 example)

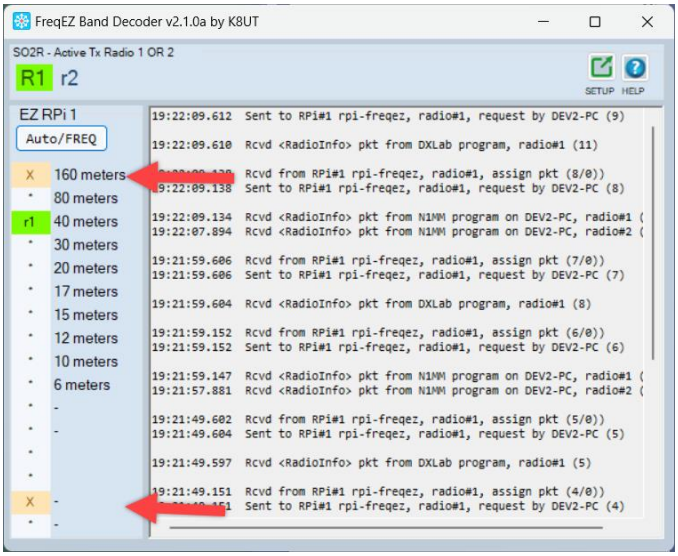
Constraints syntax:

`<Constraining RPiHatOutput>` **`Constrained RPiHatOutput(s)`** `</Constraining RPiHatOutput>`

Where: `Constraining RPiHatOutput` is the RPi HAT output number (1-64) corresponding to the RPiNr (1-4) times the activated band button position (1-16) in the Windows Console Band Button display, `Constraining RPiHatOutput` is one or more RPi HAT output numbers from 1-64 that are prevented from operating., separated by commas.

Conflicts – Network Operation with Multiple PCs and FreqEZ Consoles

When a FreqEZ Windows Console sends output request packets to an RPi Controller, the Controller assigns the request to the requested output on a *First-Come, First-Served* basis. If more than one Windows Console sends FreqEZ packets to an RPi Controller, the Controller notifies the other network FreqEZ Widows Consoles of those assignments, which appear in the Console's Output Panel as an InUse "X" character.



In this screenshot, this station is using the 40 meter antenna, output #3. There are two InUse ("X") indicators. The other station has been assigned the 160 meter antenna (output #1), and has a macro assigned for output #15. This station cannot use those two outputs until the other station has surrendered its assignments.

In this example, the InUse assignments will be surrendered when the other station moves to another band (releasing its use of the 160m dipole), or when the macro for output #15 is deleted.

FreqEZ prevents "hostage" InUse assignments, in which there is no active FreqEZ Console claiming ownership of an output. See the explanation in Console Setup under "Auto-Disconnect" for more information.

APPENDIX:

Alternative Method for Preparing the microSD Card

- If not already loaded with the Raspbian Operating system see Appendix ***Install the Raspbian Operating System on the microSD Card***
- Check to ensure you have the latest version (see Appendix ***Update the Raspbian Operating System on the microSD Card***

Install FreqEZrpi Controller Software on the Raspberry Pi

The latest version of FreqEZrpi can be downloaded from the `www.FreqEZ.com` website. The >Downloads page contains FreqEZ_RPiController_v2xx.zip – a compressed file with the files to be installed on the Raspberry Pi. The files can be installed from a Windows telnet client or from a Raspberry Pi terminal session.

- From a Windows telnet client: Download and install the free Putty telnet client program (`www.putty.org`) and connect to the Raspberry Pi in an SSH session
- From the Raspberry Pi telnet session: Connect a monitor, mouse and keyboard to the Raspberry Pi and launch the terminal app.

After logging into a Raspberry Pi telnet session, enter the following commands:

Ensure that a previous version of freqezrpi is not running

```
sudo killall freqezrpi
```

Change directory to Downloads

```
cd $HOME/Downloads
```

Download the latest version from the FreqEZ.com website

```
wget https://freqez.com/downloads/FreqEZ_RPiController_v2xx.zip
```

Unzip contents of the zip file into the Downloads directory

```
unzip -o FreqEZ_RPiController_v2xx.zip
```

Copy the freqezrpi program into the /usr/bin directory

```
sudo cp freqezrpi /usr/bin/freqezrpi
```

Make the freqezrpi program an executable file

```
sudo chmod +x /usr/bin/freqezrpi
```

Copy the freqezconfig.xml file into the /usr/bin directory

```
sudo cp freqezconfig.xml /usr/bin/freqezconfig.xml
```

Copy the rc.local auto-start file into the /etc directory

```
sudo cp rc.local /etc/rc.local
```

Reboot the FreqEZ RPi Controller

```
sudo reboot
```

Install the Raspbian Operating System on the microSD Card

The Raspberry Pi Foundation provides a utility program for burning the Raspberry Pi OS to a microSD card. This utility is much easier and less error prone than using Win32DiskImager. Download and install Raspberry Pi Imager on your Windows PC to create the microSD card: https://www.raspberrypi.org/downloads/

Launch the Raspberry Pi Imager utility.

1. Insert your new microSD card into the microSD/USB port on your PC. (NOTE: Windows might offer to format the microSD card. Click Cancel. Close any notifications that Windows might present to you about this microSD card. You do not need any help from Windows! This will only waste time and might confuse the RPi Imager software)
2. Press [CHOOSE OS] and select the recommended **Raspberry Pi OS (32 Bit)**
3. Press [CHOOSE SD CARD] and select the microSD card to be loaded with the RPi OS
4. Click the gear (config) icon
    a. *Enable Hostname* and give this RPi Controller a unique name that will serve as the hostname (network) name of this device. Keep the name under 15 characters in length. If you have more than one RPi Controller, assign a different name to each. For example: ezcontroller-1, ezcontroller-2, -3 and -4
    b. *Enable SSH* and set the username and password. The default username had always been "pi," but you can now assign a different name. (Be sure to record the name and password somewhere)
    c. *Enable Local Settings* and indicate your timezone and keyboard layout
    d. If your RPi Controller will be connected via WiFi, Enable *Configure Wireless LAN* and enter the SSID and WiFi password
    e. Click [Save]
5. Click [WRITE]

When the Raspbian Operating System installation is finished (takes about 10 minutes), remove the microSD from your PC. With the power removed from the Raspberry Pi, insert the microSD card in the slot on the underside of the Raspberry Pi circuit board. Connect a monitor, keyboard and mouse to the RPi, and then insert the power cable to reboot the new operating system.

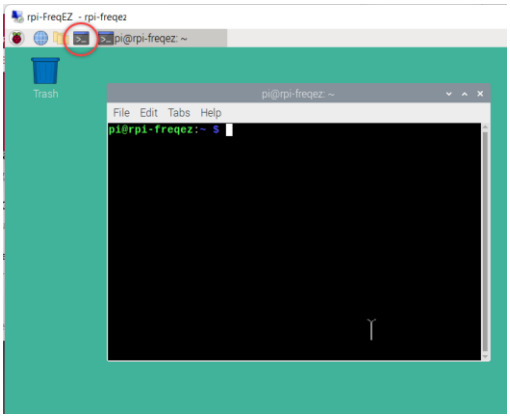
This completes the installation of the Raspberry Pi Operating System and the OS Configuration for FreqEZ operation. Following a fresh install, proceed to Appendix ***Update the Raspbian Operating System on the microSD Card***

Update the Raspbian Operating System on the microSD Card

After installing the Raspbian Operating System, and at periodic intervals thereafter (every 3 months?), you should update the Raspbian Operating System to current revision levels.

Click the Terminal (Command Line interface) icon in the Raspberry Pi GUI. In the terminal dialog widow, type:

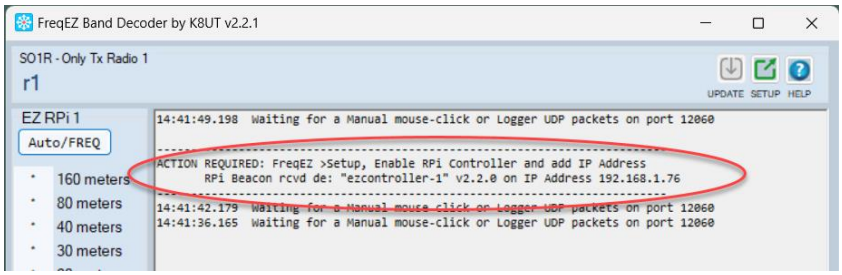
```
sudo apt update <enter>
sudo apt upgrade -y <enter>
sudo apt autoremove -y <enter>
sudo reboot <enter>
```



For FreqEZ version 2.1.0 and later:

When the RPi reboots, the Controller will begin sending broadcast packets announcing its IP address to the Windows Console.

This "Action Required" bulletin will contain the name of the Raspberry Pi and its IP Address. Enter that IP Address in the Consoles Setup dialog window.



For FreqEZ versions prior to 2.2.0:

After the Raspberry Pi reboots, mouse-over the network icon in the upper-left corner of the screen to determine the IP Address assigned by your router to this Raspberry Pi (in might have changed). This IP Address will be entered into the *RPi: IP Address* field of the Windows FreqEZ Console >Setup dialog.

Troubleshoot the FreqEZ Controller

Launching `freqezrpi` Manually

IMPORTANT TROUBLESHOOTING DETAIL: *With FreqEZrpi auto-start enabled, an instance of FreqEZrpi will always be running after the RPi boots. When you login to the RPi command line, you can confirm that FreqEZrpi is running by displaying the current tasks with Linux's command* **`ps -A`** *(*`freqezrpi` *should be found within that long list of tasks). If you were to manually launch* freqezrpi *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 command* **`sudo killall freqezrpi`** *. Confirm that freqez*rpi *is not running with the* **`ps -A`** *command. Then manually launch a new instance of the program with the commands listed above.*

FreqEZ can be started manually as a terminal session from a Windows telnet session or from Raspberry Pi's console (Command Line Interface). To launch ***`freqezrpi`*** from the console:

Close any other instance of freqezrpi:	**`sudo killall freqezrpi <enter>`**
Start the freqezrpi program:	**`freqezrpi <enter>`**

To halt the `freqezrpi` executable, type **<ctrl>+C**

When running, `freqezrpi` prints status information to the console. At a minimum, the program prints a splash screen upon launch and thereafter lines of information with each received and sent packet.

```
16:48:27.115 New <RadioInfo> packet from HAM2-PC at 192.168.1.95 on port 13066 Radio#0 to RPi#1 as Console#2 of 2

16:48:27.115 EZWin ********** 0000000000000000 15 Rx r0 rcvd keep-alive from HAM2-PC on 12060 in Manual (1/0)
16:48:27.115 EZRPi ********** 0000000000000000 15 Rx r0 sent assign to HAM2-PC at 192.168.1.95 on 12060
             Netwk ********** 0000000000000000          sent notify to DEV2-PC at 192.168.1.119 on 12060

16:48:32.777 EZWin ********** 0000000000000000 15 Rx r1 rcvd keep-alive from DEV2-PC on 12060 in Manual (56/0)
16:48:32.777 EZRPi ********** 0000000000000000 15 Rx r1 sent confirm to DEV2-PC at 192.168.1.119 on 12060

16:48:35.154 EZWin ********** 0000000000000000 15 Rx r1 rcvd keep-alive from HAM2-PC on 12060 in Manual (2/0)
16:48:35.154 EZRPi ********** 0000000000000000 15 Rx r1 sent confirm to HAM2-PC at 192.168.1.95 on 12060
```

Red Text: Windows Console Arrival and Departure status information

- Initial packet arrival from a Windows Console, the RPi Controller announces its Console number
- Disconnect notice – when a Windows Console is removed from the Controller and its outputs are released for reassignment

Yellow Text: Request Packets received by the Raspberry Pi Controller from the Windows Console

- Timestamp and "EZWin" label
- Requested output Bytes (16): *=off, 1=Radio1, 2=Radio 2, M=Manual, B=Both, O =Forced ON, F=Forced OFF
- Requested output Bits (16): *=off, 1=on
- BCD Input Value: 0-15, Tx/Rx Status,, Reporting Radio Nr: r1 or r2
- Packet Type: "Request" with new information, "Keep-Alive" with repeat information
- Packet Source and Destination: Computer NETBIOS names
- Packet Count: Total packets / Error count

Green Text: Assignment packets sent from the Raspberry Pi Controller to the Windows Consoles

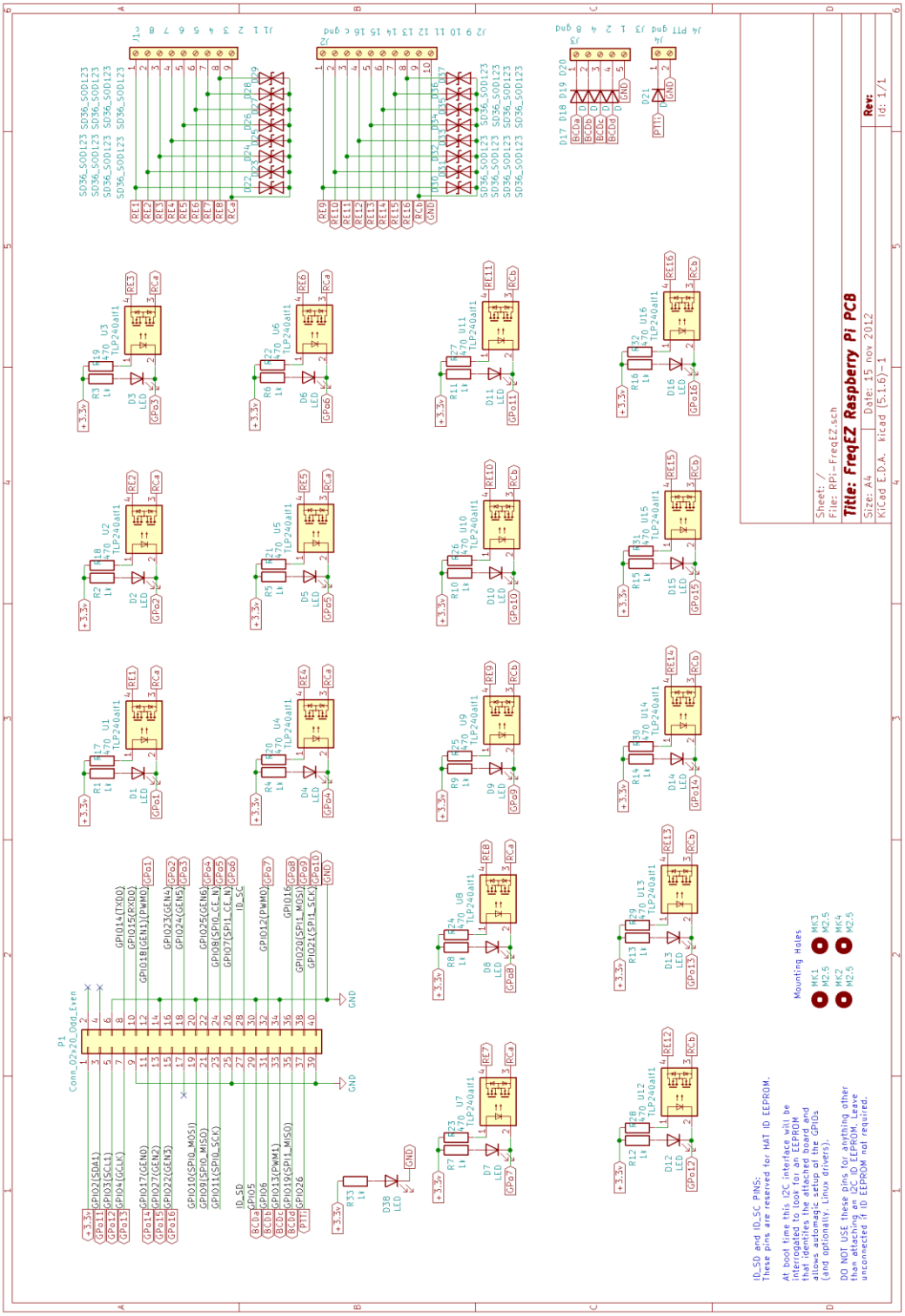
- Timestamp and "EZRPi" or "Netwk" label

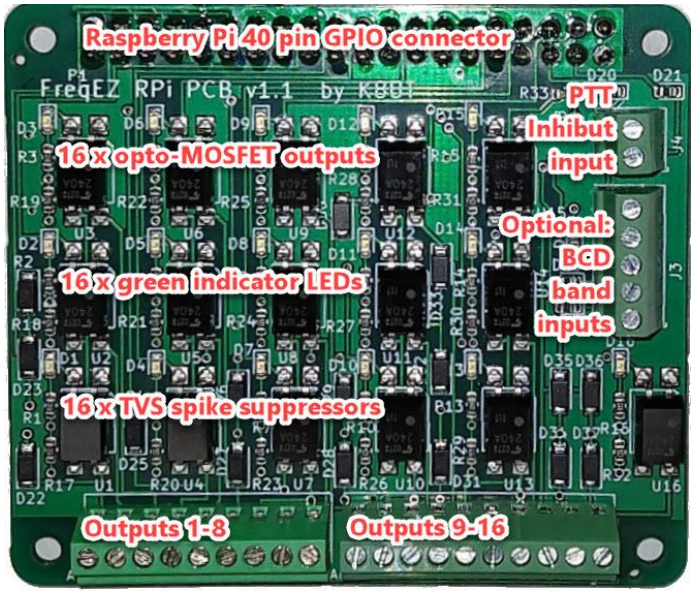
- Assigned output Bytes (16): *=off, 1=Radio1, 2=Radio 2, M=Manual, B=Both, O =Forced ON, F=Forced OFF
- Network Summary Bytes (16): Outputs assigned to other network stations
- BCD Input Value: 0-15, Tx/Rx Status, Reporting Radio Nr: 1 or 2
- Packet Type: “Assign” with new approvals, “Confirm” with repeat information, “Repeat” to network stations
- Packet Source and Destination: Computer NETBIOS names in Console IP address

•

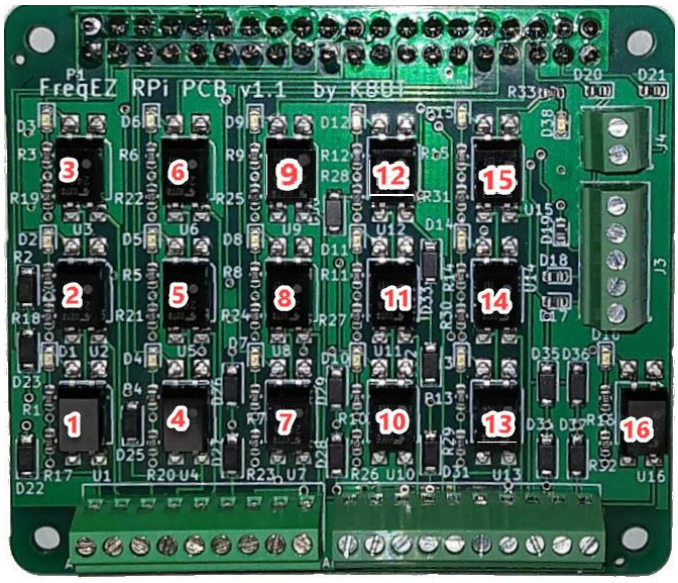
FreqEZ Board Schematics

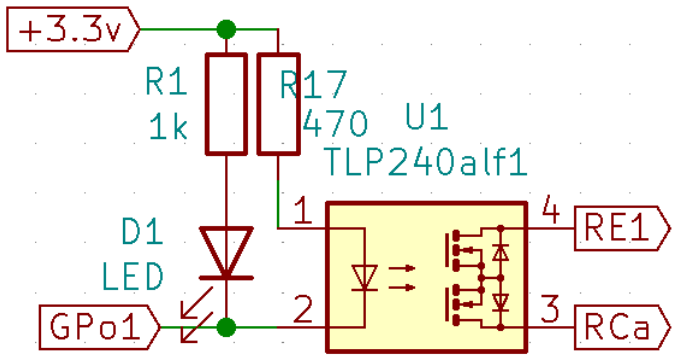
The FreqEZ schematic may look complicated, but the board contains 16 identical, simple circuits.





Parts Layout and Circuit Description





- Pins 1 and 2 are the inputs to the opto-MOSFET. Pin 1 (represented by a painted dot on the device) connects to the RPi regulated 3.3 volt power supply through a 470 ohm current limiting resistor. Pin 2 connects to a dedicated RPi GPIO pin controlled by the FreqEZrpi software

- Pins 3 and 4 are the outputs from the opto-MOSFET. Pin 3 connects to common screw-terminals for each bank (1-8, 9-16) of outputs. Pin 4 is connected to a dedicated screw terminal (1-16) for each circuit
- A green LED with 1k ohm current limiting resistor is connected in parallel with each opto-MOSFET input
- A 20 volt TVS spike suppressor (not shown in the diagram) is connected in parallel with each opto-MOSFET output

Measure Voltage on the opto-MOSFET Input Pins 1 and 2

- When not activated, the voltage to ground on pins 1 and 2 should be 3.3vdc
- When activated, the voltage to ground on pin 1 should be 1.4vdc. The voltage to ground on pin 2 should be 0.2vdc. The voltage between pins 1 and 2 should be the difference of those two - 1.2 vdc

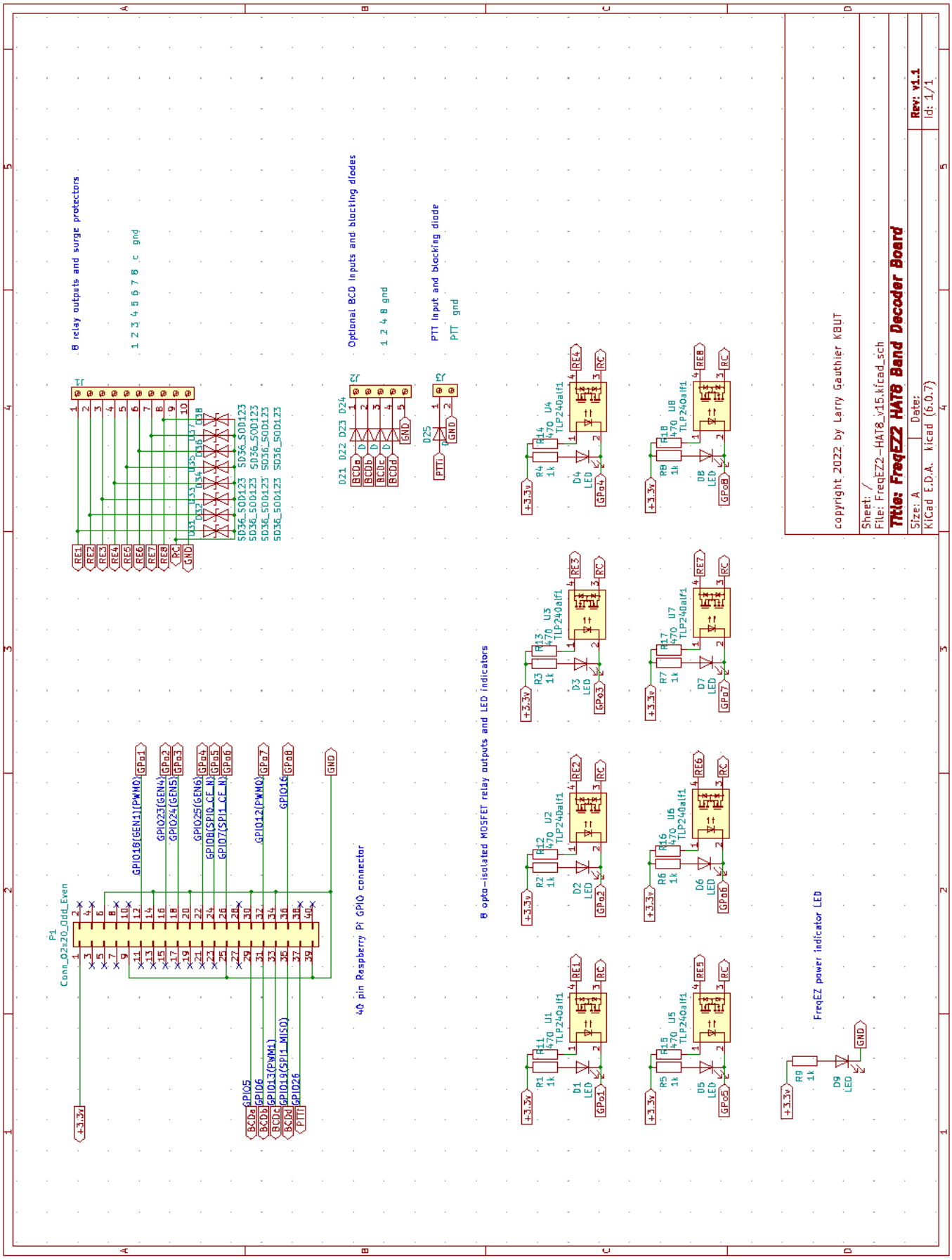
Measure Resistance on the opto-MOSFET output Pins 3 and 4

The output resistance is easily measured on the screw terminal connectors. Connections to remote antenna switches must be removed to make these measurements.

- When not activated, the resistance between each output terminal and the common terminal is infinite (open circuit)
- When activated, the resistance between each output terminal and the common terminal is very low – less than 1 ohm

FreqEZ HAT8 Board Schematic

See HAT16 details for circuit description and troubleshooting information.


8 relay outputs and surge protectors
1 2 3 4 5 6 7 8 c gnd
Optional BCD inputs and blocking diodes
1 2 4 8 gnd
PTT input and blocking diode
PTT gnd
8 opto-isolated MOSFET relay outputs and LED indicators
40 pin Raspberry Pi GPIO connector
FreqEZ power indicator LED
copyright 2022 by Larry Gauthier K8UT
Sheet: /
File: FreqEZ2-HAT8_v15.kicad_sch
Title: FreqEZ2 HAT8 Band Decoder Board
Size: A
Date:
Rev: v1.1
KiCad E.D.A. kicad (6.0.7)
Id: 1/1

Helpful Linux Commands

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

`cd`	change directory
`cd ~ or $HOME`	change directory to the user's (pi?) login home directory
`<ctrl>+c`	when running *freqezrpi* from the CLI, halts the program and return to the terminal prompt
`clear`	clear the screen
`htop`	open the Raspbian performance monitor
`ls`	list the names of files in the current directory
`ls -l`	list the names and attributes of files in the current directory
`lsblk`	list disks and partitions on the Raspberry Pi
`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 <command>`	**S**uper **U**ser **DO** – equivalent to Windows "run as administrator"
`freqezrpi`	launch freqezrpi in the /usr/bin directory (in the Path setting)
`sudo apt-get install <program>`	install the listed program (without the brackets)
`sudo apt-get install xrdp`	install xrdp program for remote GUI access using the Windows Remote Desktop Protocol
`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 freqezrpi`	set the freqezrpi 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 freqezrpi`	halt all instances of freqezrpi (esp. the Auto-Launch instance)
`sudo nano //usr/bin freqezconfig.xml`	launch the *nano* text editor with admin privileges and modify the freqezconfig XML file
`sudo nano /etc/rc.local`	launch the *nano* text editor with admin privileges and modify the rc.local auto-start file
`sudo raspi-config`	launch a text equivalent to the GUI >Preferences >Raspberry Pi Configuration program. Useful for configuring the RPi when access is only available via telnet and a command line interface