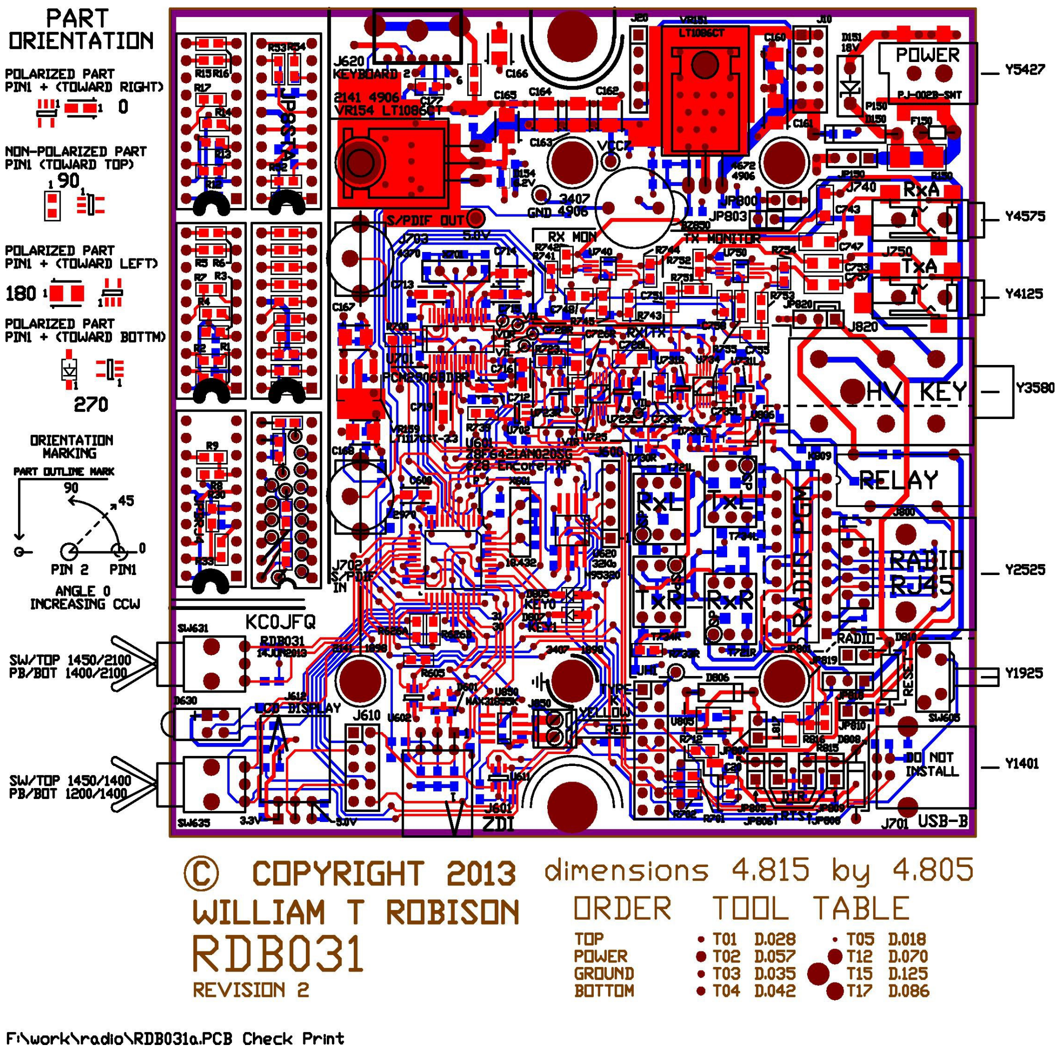


DIGITAL RADIO INTERFACE

System

http://n952.dyndns.ws/radio/

ICARC DIGITAL RADIO INTERFACE

User Guide and Reference Manual

http://n952.dyndns.ws/radio/

1 Table of Contents

2 Quick Start

IMPORTANT!

Do NOT start plugging things into the computer yet!

First download the FTDI drivers for the FT4232! Prior to first plugging the unit into your computer the drivers may need to be installed. There are O/S specific instructions that must be followed for a successful install of the serial port drivers for the FT4242 USB to serial adapter. The USB hub and the USB CODEC will install automatically on both Windows and Linux without assistance.

2.1 Radio configuration jumper

Select and install the appropriate jumper card for your radio. The jumper card holds an attenuation pad that must be populated for your radio. Switching radios may require switching the jumper cards to correctly interface to the radio.

2.2 Working without an LCD

There is code in the eZ8 to allow use of the interface without the LCD. This feature can be enabled by default by sending the following command on the serial control port.

Command(s): SAVE,DS=INIT,1,1,1,100

This saves a record in the flash file system to issue the ECHO command as part of the INIT processing. Information that is sent to the LCD is then sent to the host through the serial port. This was developed on a system using *Hyperterminal* as that is what is usually available on bare Windows systems. It assumes that the terminal emulator will correctly process VT100 escape sequences as it uses screen positioning commands to paint the screen image. *Hyperterminal* has been observed to misbehave and switch into VT52 mode (screen positioning commands are incompatible) which will not display correctly. If *Hyperterminal* appears to print nonsense, restart it or switch it back to VT100 emulation.

The LCD display is primarily used to see and adjust the current gain levels. The data can be displayed on the host system using the following commands to enable and disable the display.

Command(s): ECHO,*duplex*, 1

Command(s): ECHO,*duplex*, 0

Command(s): DSPY

Windows may also treat the serial traffic as if a serial mouse were connected. Correcting this requires a register hack to the following key

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\sermouse/start = 4

2.3 Gain and VOX Controls

Adjust the gain controls for clean operation and save the settings in the flash file system. If using multiple radios, the setup for each radio may be saved individually and retrieved as needed from the front panel.

The VOX control setting should be about right, it is not dependent on the radio, rather the VOX control code looks at the transmit level out of the CODEC.

I have found that on the IC7200 that was is used for software development that getting the programming header attenuation level is critical to easy operation. With the transmit gain level set correctly the computer volume control may be set to full and the gain setting in the interface set to 45 that the IC7200 is just starting to drive the ALC. This seems to be the best way to operate PSK31 at around 30 watts. The radio power level can then be increased and the volume in the interface slightly decreased to achieve higher output power. During the HI JUNO event I was able to reduce audio levels from the interface front panel and raise the power output of the IC7200.

2.3.1 Transmit gain level

The audio CODEC (T.I. PCM2906) has pins that control the volume on Windows machines. The CODEC controls consist of a volume up, volume down and mute lines. The volume up and mute lines are controlled by pins on the the eZ8, as such they are under software control. When the eZ8 is reset is pulses the volume up line long enough to cause the volume control to move to 0dB attenuation on the Windows system. The volume setting is driven to full volume anytime the eZ8 is reset.

The mute line is controlled by the right front panel button when the control toggle switches are both in the center position. Witch the switches in the center position the audio to the radio (and to VOX control) can quickly be muted.

The analog gain levels in the interface are set through the front panel switches when they are set to the following configuration:

Left Switch	Right Switch
Center	Left

The buttons are then used to adjust the setting. The number is **not** in dB, rather it is best to think of it as an arbitrary number. Since this is changing a digital potentiometer, the setting are repeatable, so 45 is always the same gain level.

Left Button	Right Button
Up 10	Down 3

A value of 50 is approximately unity gain. A setting of 100 is about +6dB. A value of 25 is about -6dB.

2.3.2 Receive gain level

Set the front panel switched to the following configuration:

Left Switch	Right Switch
Center	Right

The buttons are then used to adjust the setting. The number is **not** in dB, rather it is best to think of it as an arbitrary number. Since this is changing a digital potentiometer, the setting are repeatable, so 45 is always the same gain level.

Left Button	Right Button
Up 10	Down 3

As with the transmit levels, 50 is unity gain and 100 is about +6dB.

2.4 Personal Identification

If your callsign or name are incorrect, this can also be loaded into the flash file system. Once in the flash file system, this information is non-volatile, so you need not re-enter following a power cycle or reset event.

Command(s): SAVE,CF=<call>,*new-callsign*

Command(s): SAVE,CF=<name>,*new-name*

Command(s): SAVE,CF=<zitu>,*new-ITU-zone*

Command(s): SAVE,CF=<z_cw>,*new-CW-zone*

Command(s): SAVE,CF=<name>,*new-maidenhead-locator*

Code that is generated in the interface will have your name and callsign substituted into the message before it is sent out. This allows generic messages to be stored in the permanent area of the flash file system that require callsign or name.

2.5 RF in the shack

The USB bus can be sensitive to RF in the shack, particularly 20M and below. The USB HUB device is a full speed hub device that operates at 12MHz on the USB bus (although the FT4232 is a high speed device, the PCM2906 and the TUSB2036 are full speed devices and the TUSB2036 effectively limits the FT4232 to a bus rate of 12MHz). This is not too far from 20M frequency of 14MHz and excessive RF in the shack has the potential to cause data corruption on the USB bus which leads to device disconnection.

This was encountered during development and was cured by providing adequate shielding and grounding of the system. Grounding the host computer chassis to the interface ground alleviated USB problems.

2.6 FLDigi

Fldigi usually needs to be reconfigured to operate with left channel controls. Check the following temps in the *Configure → Soundcard* menu area.

Configure → Soundcard → Devices

Make sure you have the correct USB CODEC selected. If you have other USB sound devices (Signalink, for example) make sure you are looking at the correct sound device.

Configure → Soundcard → Settings

I have the sample rate set at *48,000 samples/second* and the *Medium Sync Interpreter* selected. The PCM2906B device has sample rate problems corrected, so you should be able to run it an any rate above ***2•Fny*** That FLDigi will accept.

Configure → Soundcard → Right Channel

Here we must force the use of the left channel or configure the programming header to work the left channel. Note that the programming headers all have provisions to connect the right channel so having software that deals only with the right channel is not a show stopper (although it may require populating thw audio transformers).

I have mono audio output checked. There is also a reverse left/right selection that may help.

2.7 Windows COM Ports

The FT4232 appears to the windows system as a group of 4 consecutive COM ports, 3 of which actually show up. Port B of the FT4232 is configured as a bit-bang interface, so it doesn't appear as a COM port although the port number that would have been assigned to it can not be seen or used) .

The following commands may be used to start the device manager so that all the COM ports in the registry may be viewed (and removed if necessary).

```
set devmgr_show_nonpresent_devices=1
start devmgmt.msc
```

2.8 SWR/Power Detector

And addition to the system for 2014 is an SWR/Power meter. The SWR/Power meter connects through the CI-V bus or through an RS485 bus. It usilizes the same *eZ8 Encore XP!* processor used in the radio interface along with many of the same peripheral blocks used in the radio interface to provide a real-time power and SWR meter function. Adding a haywire to the radio interface allows the CI-V traffic to be monitored and displayed on the radio I/F front panel.

3 Heritage

What influenced the design.

There is a broad spectrum of digital radio interfaces on the market today. None seem to have quite all the features one might want in a radio interface and we want to avoid having a proliferation of boxes occurring in the shack.

By taking the best of existing devices we hope to arrive at a design that satisfies a broad number of applications for digital radio interfaces.

Some of the demands we place on the interface are.

- Reasonable size and cost
- Good analog performance
- Interface cleanly with as many existing radios as possible

3.1 Size

Size of the project is set by the box it is housed in, a Hammond 1598B. This enclosure was chosen to leverage existing knowledge gained from previous projects using this enclosure to produce mechanically working boards with as little effort as necessary. Panel mechanical layout is already in existence and is reused.

3.2 Analog Performance

Set by the PCM2906 and the transformer, a Triad Magnetics SP-70.

The PCM2906 samples at rates up to 48KHz giving a bandwidth in excess of 20KHz. The SP-70 transformer is a broadband audio transformer that has a frequency range of 300HZ to 100KHz. The transformer performance allows the full bandwidth of the PCM2906 to be used for SDR applications.

Data gathered from the prototype indicate a noise floor below -100dB. The power supplies are well isolated and well bypassed giving excellent analog performance.

3.3 Interface capability

Audio, of course, using a stable USB based audio CODEC. Since the CODEC is an older design, it is well supported in current operating systems. Audio is transformer isolated and fully buffered.

Transmit keying control may be controlled by the host software or through the use of the onboard eZ8 Encore processor monitoring the right transmit channel.

Isolated RS232 interface for radios using a DE9 type connector. The solution used here is relatively inexpensive at the cost of being minimally isolated, only enough to keep the hum out of things as long as the radio equipment is operating near ground potential.

Well isolated TTL serial port. This is easier to achieve with opto-isolators that incorporate logic level gates. We get better isolation.

SPI/JTAG capability provides a means of programming the eZ8 and other equipment that have JTAG or serial debugging interfaces.

3.4 SWR/Power Meter

This accessory subsystem makes use of many of the logic blocks that were developed for the radio interface. The CI-V interface, EEPROM interface, buzzer interface, programming interface and reset control are taken directly from the radio interface.

The control interface is similar to the radio interface with and evolutionary step in managing setup parameters that may migrate back into the radio interface in the future. Rather than having different real-time and stored setup strings, this unit saves configuration strings as a set of commands that are run during system initialization.

Descriptions that are for the SWR/Power Meter as shown in light blue throughout the manual.

4 Features

List of device features.

This manual is probably out-of-date. If you think it is current, we will change it, just to prove you wrong (although more likely just to add more information so you can get at it).

Look on the website for the current revision:

http://n952.dyndns.ws/radio/index2.html

4.1 Radio Interface

The Radio Interface and SWR/Power meters are combined in one manual here, so the radio interface features are listed here first.

4.1.1 USB Hub

Multiple USB device using a single USB port on the host system.

Provides a single downstream port (i.e. USB-A) for one additional USB device.

4.1.2 FT4232 legacy serial interface

Support for the device directly from *www.ftdichip.com*. Linux drivers already present in later kernels.

4 ports from a single chip

External EEPROM to store USB descriptors. Manufacturer, Product and Serial descriptors are specific to our project.

4.1.3 USB Downstream port

Take advantage of the spare port on the USB hub. Has full power control and monitoring as well as a means of bypassing the USB current limit for things like the Raspberry-PI.

Should you choose to override the current limit, verify that the TO220 regulators do not overheat. It may be necessary to reduce the input voltage to 8V and retrim the first stage regulator.

4.1.4 Isolated RS232

This is a 2/data 2/control isolated RS232 port presented on a DE9P, pinout matching the PC-AT serial port.

4.1.5 Isolated CI-V

The CI-V port, for controlling ICOM radios, is optically isolated and may be configured for full or half duplex operation. Although the 3.5mm connector is a tip/ring/sleeve configuration (i.e. stereo), jumpers allow it to be used with the tip/sleeve configuration seen on ICOM radios.

For applications that require full duplex, a tip/ring/sleeve type cable is required to keep transmit data and receive data separate.

The CI-V interface is 5V logic levels that are inverted with respect to RS232 polarity. In other words the line idles at +V and the start bit is 0V. The pull-up on the transmit

4.1.6 SPI/JTAG

The JTAG interface is an unbuffered port from the FT4232 device. Using the bit-bang modes this channel may be used to program SPI devices and/or JTAG devices (such as the Linksys WRT54 routers)

4.1.7 Display

The display is not absolutely required, it can be eliminated to reduce cost slightly. Software in the eZ8 can direct status information to the host through the host serial port on the eZ8 as well as the display.

4.1.8 Control Switches

These two switches can be inspected by the host computer and by the eZ8. Since they are center off, they may used to select one of nine settings or modes.

4.1.9 Status LEDs

Two of these are controlled by the host system and two are controlled by the eZ8. When using typical ham software.

4.1.10 Zilog eZ8 Processor

Large code space (64K). Looks like we needed it as the current code is about 50K bytes.

Vendor supplied development tools at no cost. C based development environment can be downloaded form web.

4.1.11 TI (Burr Brown) PCM2906B CODEC

USB 1.1

Revision B silicon fixes some issues with odd sampling rates that were present in earlier revisions.

The revision 1 artwork has an added feature to allow the eZ8 to drive the volume control to full on to eliminate the need to worry about this in the Windows environment.

4.1.12 Multi Layer Circuit Board

Both PCBs are 4-layer boards to make routing much easier and to provide an electrically quiet environment. The daughterboard splits the ground plane to keep circulating currents separate and to isolate the radio from the computer. The power plane is also split to keep coupling to a minimum.

4.1.13 Digital Gain Control

Keeps low level audio signals localized.

Computer (eZ8) required to set, but can be reset accurately.

4.1.14 Buffered Audio Monitor

Audio Monitor plug/unplug doesn't affect audio levels.

4.1.15 LogicLevel Keying

Opto-isolator with Darlington output. This is a sinple open-collector driver with voltage limits imposed by the optical isolator chip.

4.1.16 Relay controller Keying

Reed relay using separate control path. This reed relay provides 100)V isolation for use with tube radios. The relay contacts and circuit board geometry limit the maximum voltage (i.e. this is not a solid state switch).

This connection to the outside world may also be used to drive a solid state relay to control a reflow oven. The jack is in a TIP-RING-SLEEVE configuration with the SLEEVE connected to circuit common. TIP-RING are used to control a radio as they are completely isolated from circuit common.

4.1.17 Keyboard Connector

Potential connection point for an iambic paddle or a PC keyboard. The connector has 2 logic levels pins connected directly to the eZ8 along with power and ground.

This could also be used to connect a paddle if the appropriate software were added to the eZ8.

4.1.18 S/PDIF connectivity

Chip supports it, so we provision it. It consists of a pair of vertical mount RCA jacks (although they aren't physically installed).

4.1.19 Thermocouple amplifier

The RDB board has a MAX31855 thermocouple amplifier and A/D in a small package. It need only be installed if it is to be used although the system will operate just fine with the device installed.

Perhaps useful as a temperature probe elsewhere?

4.2 SWR/Power Monitor

The SWR/Power Monitor features.

4.2.1 Isolated CI-V

The CI-V port is optically isolated and may be configured for full or half duplex operation. Although the 3.5mm connector is a tip/ring/sleeve configuration (i.e. stereo), jumpers allow it to be used with the tip/sleeve configuration seen on ICOM radios. The 1K pullup resistor is connected through a jumper to limit the drive current required on the CI-V bus.

For applications that require full duplex, a tip/ring/sleeve type cable is required to keep transmit data and receive data separate.

The CI-V interface is 5V logic levels that are inverted with respect to RS232 polarity. In other words the line idles at +V and the start bit is 0V. The pull-up on the transmit

4.2.2 Display

The display interface is a 4-bit parallell interface and is provided only as a diagnsotic aid. When connected it will be updated but is not required for operation.

4.2.3 Zilog eZ8 Processor

Large code space (64K). Looks like we needed it as the current code is about 50K bytes.

Vendor supplied development tools at no cost. C based development environment can be downloaded form web.

4.2.4 Multi Layer Circuit Board

Both PCBs are 4-layer boards to make routing much easier and to provide an electrically quiet environment. The daughterboard splits the ground plane to keep circulating currents separate and to isolate the radio from the computer. The power plane is also split to keep coupling to a minimum.

4.2.5 Digital Slope/Gain Control

Optional parts to trim the AD8307.

Computer (eZ8) required to set, but can be reset accurately.

4.2.6 Audio Alarm

The same sonalert type buzzer is present in the SWR/Power Meter. As it is software controlled, the software will determine it's function.

5 Summary of Functional Blocks

Yak Yak

5.1 Radio Interface

The .

5.1.1 USB Hub

The USB hub is based on a Texas Instruments TUSB2036 full speed USB hub controller chip. The controller chip implements the complete USB 1.1 HUB function providing power control for the single downstream port. The downstream port is provisioned with a power switch and monitor to fully comply with USB 1.1 specifications. If there is a need to draw more power through the port, a jumper is provided on the motherboard to override the power control from the hub chip to the power switch.

The second downstream port us ised to connect to the daughterboard where the PCM2906 is located. No power control is needed as there are no provisions to switch the PCM2906 on and off other than through removing the power supply for the system.

The first downstream port connected to the local FT4343 quad serial port chip. As with the PCM2906, the FT4232 is continuously powered. The power control output from the TUSB2036 is inverted and delivered to the FT4232 as a reset control signal. The host can effect a reset of the FT4232 through the USB enumeration protocol.

5.1.2 FT4232 Serial Interface

This device provides 4 serial ports to the host system. Three are presented on the back panel for use externally and one is routed to the daughterboard and to the LCD connector.

Although the FT4232 is a high speed USB device (i.e. capable of operating at 480Mb/S rate on the USB bus), the TUSB2036 is a USB1.1 hub device that effectively limits the FT4232 to operating at the 12Mb/S rate of a full speed USB device. This is not anticipated to be a problem for the applications this interface is used in,

5.1.2.1 Serial channel 0, ICOM CI-V

The first serial channel of the FT4232 is presented on the rear panel as a 3.5mm stereo jack for use with the ICOM CI_V system. The CI-V interface is optically isolated and may be jumpered for half duplex operation.

The TIP carries the receive signal, the RING carries the transmit signal and ground is carried on the SLEEVE of the phone jack.

Two jumpers (JP923 and J924) are provided on the board to configure the mode and allow a bit of flexibility in connecting to an external CI-V interface. JP923 connects transmit and receive together forming a half duplex system that is compatible with the CI-V interface. JP924 can be removed to isolate the RING on the 3.5mm jack to allow a stereo patch cord to be used with a CI-V radio. When this jumper is in place a mono style patch cable must be used to connect to an ICOM radio.

The voltage high level presented on the interface is around 5V. The transmit signal is buffered through an open drain driver with a 10K pullup.

The receive signal is buffered through a schmitt trigger buffer. The receive line also has a 100K pullup.

The DTR and/or RTS modem control signals on this channel may be jumpered on the RDB board to drive either/both of the keying circuits.

5.1.2.2 Serial channel 2, SPI/JTAG

The second serial channel of the FT4232 is routed directly to a 20 pin header to be used as a JTAG or SPI programmer. The FT4232 has built in bit-banging modes for this application.

5.1.2.3 Serial channel 3, Command/Display

The third serial channel of the FT4232 is routed to a connector that delivers this channel to the daughterboard. OPN the daughterboard the serial connection is routed to the on-board eZ8 processor.

5.1.2.4 Serial channel 4, RS232

The fourth serial channel of the FT4232 is presented on the rear panel as a DE9P connector with pin-outs that match the serial ports found on older PCs. The board is typically populated with a MAX3250 isolated level shifter. This level shifter provides some degree of isolation between the radio and the host computer. When configured with this chip only TXD, RXD RTS and CTS are presented to the outside world. DCD and DSR are tied active and RI is tied inactive.

If all signals are required and isolation is not an issue, the MAX3250 may be replace with a MAX3241. With the MAX3241 all signals on the DE9P are present. This is achieved at the cost of eliminating the isolation feature.

The DTR and/or RTS modem control signals on this channel may be jumpered on the RDB board to drive either/both of the keying circuits.

5.1.3 Power Subsystem

Power to system is applied through a 5.5mm coaxial power connector. There is overvoltage and overcurrent and reverse voltage protection on the power input. Following the input protection the raw input voltage is routed to the other board. Part of the input overcurrnet/overvoltage protection circuit will drop the input voltage by 500 to 800 millivolts.

Voltage regulation is provided using three terminal regulators in three stages. The first stage regulator drops the input voltage to 7V. A second stage regulator drops the input voltage to 5V which is supplied to circuitry that required this voltage (such as the CI-V power converter and the PCM2906 and audio processing circuitry). A third stage regulator drops the 5V to 3.3V for use by the digital logic (serial chip, hub chip, eZ8)

Power dissipation in the 3.3V regulator is negligible, current consumption is small and voltage drop is only 1.7V.

The 5V regulator dissipation is also relatively low, although it is larger than the 3,3V regulator as it carries all the 3.3V current load in addition to all the 5V load. The relatively small 2V drop keeps power dissipation in check on this device.

The first stage regulator, the 7V output device, carries the primary heat load as it sees a large voltage drop when using a 12V supply for the system. If there is a need to provide substantial current to the downstream USB port on the motherboard changing from a 12V supply to a 9V supply reduces the heat load substantially.

5.1.4 Display Connector

Display connectors are provided both on the daughterboard and the motherboard. The inter-board connector allows the display to connect to either connector (although the power pins are slightly different). Selction of the appropriate connector depends on how the USB downstream port will be used.

If the downstream port is connected to a low power device, the display should be connected to the motherboard to reduce power dissipation in the 7V regulator on the daughterboard.

If there is a need for more power on the downstream USB port the display can be connected to the daughterboard to balance the power dissipation on the 7V regulators.

5.1.5 LED Indicators

There are 4 LED indicators on the front panel and 2 on the daughterboard.

The top 2 LED indicators on the front panel are mounted on the daughterboard and are controlled by the eZ8 processor. The bottom 2 LED indicators are mounted on the motherboard and are controlled by the control pins on the 3rd. serial port of the FT4232.

There are also 2 surface mount LEDs on the daughterboard that are located in the control signal to the keying circuit. The illuminate when the eZ8 drives the keying circuit.

5.1.6 Switches

There are 2 3-position toggle switches on the front panel that are mounted on the daughterboard. The are connected to the eZ8. The eZ8 can detect all of the 3 positions.

5.1.7 Buttons

There are 2 push buttons on the front panel that are mounted to the motherboard. These two pushbuttons are debounced and delivered to the eZ8 on the daughterboard. In addition, the buttons are latched and presented to the same serial port that controls the LEDs on the motherboard.

There is also a reset switch located on the back panel that is routed to the eZ8. This reset is exclusive to the eZ8 (i.e. it is not used elsewhere on the daughterboard nor is it routed to the motherboard).

5.1.8 EZ8 Processor

The processor used on the board is a ZiLog *eZ8 Encore XP*. The processor is in a 44 pin package and has 64KB of program flash ROM and 4KB of RAM. It has a generous complement of peripherals that allow control of the audio processing section of the system.

5.1.9 PCM2906

This is the USB audio CODEDC (Coder/Decoder) that provides the soundcard interface.

The PCM2906 provides 3 volume control input lines; *volume-up*, *volume-down*, and *mute*. The eZ8 controls the *volume-up* and *mute*, while the *volume-down*, is unconnected. Pulsing the *volume-up* line tells the USB audio handler to increase the volume, eliminating the need to deal with this in the Windows environemnt.

5.1.10 Transmit/Receive Audio Path

The transmit and receive audio path are very similar. There are 2 channels in each direction that consists of an opamp gain stage between the CODEC and the isolation transformer. Gain is controlled by a programmable resistor in the feedback path of the opamp. The programmable resistor is controlled by the eZ8. The audio signals are confined to the path between the CODEC and the isolation transformers, that is to say since we don't have analog controls on the front panel, the low-level audio signal does not have to leave the immediate area of the opamp so there is less chance to pick up noise.

5.1.11 Monitor Audio Path

The audio monitors consist of a simple unity-gain opamp placed between the CODEC and the audio monitor jack on the back panel. This provides isolation to keep external noise out of the audio path and to prevent loading the audio signals in and out of the CODEC (so adding or removing an audio monitor from the back panel jacks will not change signal levels).

No gain control is provided for the audio monitor, all we see on the jack is a 1 volt line level output. Gain control could have been added, but is only adds parts and complicates the board layout. It appears to be more effective to simply make use of the volume control on the external speaker.

5.1.12 Audio Isolation

The audio signals to and from the radio are routed through 1:1 audio transformers to isolate the radio from the host. The transformers chosen seem to be the most expensive transformers I could find. Looking at the noise floor and the audio bandwidth seems to make the investment worthwhile.

Note that for basic operation one would only have to populate the right channel parts. Both right and left channel are provisioned on the artwork to allow the interface to be used with a quadrature system (i.e. Software Defined Radio)

5.1.13 Radio Keying

Two independent keying circuits are employed in the design. Most modern radios are compatible with a simple open collector design. For these cases KEY0 is a photo-darlington. For the radio that has other requirements KEY1 is a relay switch colsure.

The eZ8 can control the keys independently, so it is possible to control these two independently (this becomes a software issue).

Either RTS or DTR from the motherboard may also be employed to key either of the keying circuits. The RTS and DTR signal from the motherboardf may come from either the CI-V interface or from the RS232 interface.

5.1.14 Pinout Configuration

The radio connection is through an RJ45 modular socket. The pin-out is determined by a 20 pin header that is pin compatible with the programming headers us on the Tigertronics Signalink. The left channels are supplied on the additional pins.

5.1.15 Annunciator

A Sonalert may be added to the board that is controlled by the eZ8. It might be useful for code work and as a feedback for button presses.

5.1.16 Thermocouple Monitor

A Maxim-IC *Cold-Junction Compensated*

Thermocouple-to-Digital Converter is used to measure board temperature and to monitor the temperature of an external thermocouple. This monitoring capability combined with the keying circuit is used to implement a reflow oven controller.

5.2 SWR/Power Meter

The SW/Power meter uses the same *eZ8 Encore XP!* As the processor as is used in the radio interface. One of the two serial channels of the eZ8 are used to communicate between SWR meters and the other is used to implement an ICOM CI-V interface. The A/D subsystem is used to read the power level from the directional coupler.

5.2.1 Directional Coupler

The circuit board is configured to allow implementing either a Bruene bridge or a tandem match bridge. Half of the circuit board area is used for implementing the bridge.

5.2.2 Log Detector

A log detector is employed to convert the RF signals from the directional coupler into what amounts to a dB/count scale. Using log scaled system makes better use of the available bits in the A/D. The AD8307 log detector is also responsible for rectification of the RF signal from the coupler.

An input attenuation network is provisioned on the board to allow some scaling of the RF signal so that is fits within the range of the AD8307.

The circuit board also has pads for a MAX5388 (digital potentiometer) to allow the MAX5388 to be trimmed should that be necessary. Note that the software in the eZ8 employs a 3rd order polynomial fit so the MAX5388 should not be necessary.

5.2.3 Detector Filtering

The output of each of the the log compressors is passed through a buffer amp (TS321) to isolate the AD8307 from the eZ8 analog subsystem. A small amount of gain can be added through the TS321 but that should not be necessary as the AD8307 levels match well with the input range for the A/D subsystem.

Following the buffer amp, the analog signal is delivered directly to eZ8. IN addition there are two filters the signal is passed through before being routed to the eZ8 on two additional A/D channels. One filter is a simple RC filter and the other is a peak detector. The peak detector is a simple diode/capacitor that captures the peak voltage. The voltage from the AD8307 with no power through the coupler is high enough to account for the forward voltage drop of the diode. As the count-to-power conversion is done using a line fit, the voltage drop of the diode as accounted for.

5.2.4 Power

The power supply is taken from the radio interface with minimal changes. As the SWR meter is intended to be powered from the radio interface, some of the input protection has been eliminated although there is a reverse biased diode across the input that will conduct should the power be connected backwards.

The power is dropped to around 7 volts in the first regulation stage using a TO220 device. Power consumption is quite low, so using a 12V input should not present a problem as the TO220 is mounted to a heatsink.

The second stage regulator is fixed 5V regulator. The 5V power is individually filtered before being passed by the AD8307. The TS321 is powered directly from the 5V regualtor.

A third stage regulator provides the 3.3V used by the eZ8 and the remaining logic on the board. This is also a simple 3-terminal fixed regulator.

5.2.5 eZ8 programming and reset

The reset circuit and the in-circuit programming circuit are takes directly from the radio interface. A reset button is located on the rear panel and the programming header is a naked 6-pin header.

5.2.6 Display Interface

The power meter is provisioned with a parallel LCD interface that is intended to be used to calibrate and debug the unit (when in use, the data from the unit is delivered through the CI-V interface and may then be monitored and displayed by the eZ8 in the radio interface).

The LCD interface is a 4-bit parallel interface that is compatible the the more-or-less industry standard HD44780 chip. There are three control signals: *register select*, *read/write*, and *chip enable*; all controlled by simple GPIO bits on the eZ8. Two PWM channels in the eZ8 are used to generate the contrast control voltage and the backlight drive. The LCD device is assumed to be a 5V device as that is what is supplied on the power pin.

The contrast voltage is generated by a simple RC filter on the T2OUT line from the eZ8. The backlight circuit is a little more sophisticated, using an inductor and a switching transistor to regulate the current through the LED backlight while reducing spurious emissions.

5.2.7 EZ8 connections

The eZ8 GPIO pins are connected up in much the same manner as the radio interface. The control pins to the MAX5388 are buffered with 5V level shifters. The same external EEPROM is provisioned to allow calibration data to be updated without having to make use of the eZ8 flash memory.

The A/D subsystem employs a precision reference in the MAX6102 rather than using the internal reference in the eZ8 A/D controller. The 2.5V value matches the available voltage switch from the AD8307.

5.2.8 Serial Port Connections

The two serial ports of the eZ8 provide and RS485 bus, an ICOM CI-V bus, and a debugging interface.

The RS485 and CI-V will be covered in the following sections so no additional details will be discussed here.

The serial pins on the eZ8 are routed through resistors with the CV-I pins normally having a low value resistor (100 ohms), The RS485 pins are also routed through resistors but make use of large value resistors (3.3K to 4.7K) to allow the debug feature to function.

The RS485 signals are routed to a 4-pin header. The conenction is electrically between the eZ8 and the series resistor in the data lines to the RS485 driver. The series resistor allows the debug circuit to driver the eZ8 RxD pin so all that is required to used this feature is to connect a logiv level serial devince into the header. The SPI/JTAG port on the radio interface is suitable for this task although in a window environment it may be necessary to update the flash configuration in the FT4232 to switch the B channel back to a simple serial device.

5.2.9 RS485

The RS485 interface is implemented using a MAX3075 to form a half-duplex communications link to other meters. Four pins on the eZ* are employed to control the RS485 bus. The transmit and receiver data, of course, are used as expected, to send and receive serial data. To control the transmit function in the MAX3075 the serial controller in the eZ8 provides a transmitter enable pin (called "DE1")that is connected to the transmit enable pin on the MAX3075. When the eZ8 transmits data, it asserts the "DE1" pin one bit before the start bit and continues to drive it one bit time after the stop bit.

The receiver enable on the MAX3075 is connected to the transmit enable pin through a 4.7K resistor and to a GPIO pin on the eZ8 (called "CTS1"). To suppress receiver during transmit, the eZ8 leaves the the "CTS1" pin configured as an input and allows the "DE1" pin to switch the MAX3075 between transmit and receive (the transmit and receive enables pins on the MAX3075 are opposite polarity).

If the eZ8 needs to listen to the RS485 bus during transmit, the "CTS1" pin may be programmed as an output to directly control the receive enable.

Although the MAX3075 doesn't have an internal RS485 bus termination (that is provided by a resistor and jumper on the circuit board), it does handle the bus correctly when not driven. When the bus is idle, the receiver path presents the correct idle polarity to the eZ8.

5.2.10 CI-V

The CI-V circuit is almost identical to that used in the radio interface. The only change is the addition of a jumper to allow the 1K pullup on the CI-V data pin to be taken out of the circuit (the radio interface provides that pullup so there is no need for another).

5.2.11 Annunciator

Same sonalert buzzer as used in the radio interface. As on the radio interface, the buzzer is controlled by a transistor connected to a GPIO pin. Obviously this function is controlled by software.

6 Assembly Notes

Text Body

The SWR/Power Meter can be built using a magnifier and a high quality soldering station. The only fine pitch part is the MAX5388 digital potentiometer which is not currently installed in the first boards (so the software does not make use of it at this time). If there is interest in producing these boards, more can be fabricated although assembly is left to the user.

6.1 Reflow

The first boards were NOT reflowed.

6.2 Hand Assembly

The first boards were hand assembled.

6.3 Uninstalled Parts

Many parts are provisioned on the board that will not be installed. These are features not needed or implemented. If room is available on the board, it is easier to provide pads now rather than trying to add things later.

Some features are not provisioned, such as the oven temperature monitor. The feature can be *activated*, if you will, by installing the parts. As of V1.49 there is enough room in the ROM to provide support for all of the features present so all that is required is installing parts if an unimplemented feature is desired.

6.3.1 USB Connector

The USB connector on the daughterboard is not installed (it should never be isntalled as the USB connection is through the hub on the motherboard). It was provided to test the daugherboard standalone. R701 and R702 serve to isolate the traces to this connector and are also left uninstalled.

6.3.2 Keyboard Connector

No software planned for this feature at this point in time. Requires modifications to the plastic housing to provide access to this connector.

6.3.3 S/PDIF Connectors

No radio equipment has S/PDIF I/O, these are superfluous.

6.3.4 MAX51855 (thermocouple interface)

Required only if using the reflow oven controller. J850 may also be left off.

6.3.5 LCD Display Connector, J612

LCD will be connected to the motherboard to push the power load from the display away from the daughterboard. Power pins on the two connectors are slightly different. If the downstream USB port is to be used to provide much power, it may be desirable to move the LCD to the daughterboard to spread heat dissipation between the two boards.

6.3.6 Patch boards

RDB015 artwork has some errors that are corrected with a patchboard. The patchboard corrects a wiring error in the audio monitor circuit. The parts of the audio monitor are removed and new parts installed on the patchboard. The patchboard is then installed onto the RDB015 using wires is several vias and several haywires.

RMB015 has two issues that are addressed with patchboards. The reset control to the FT4232 is incorrect and requires an inverter that is located on a patchboard. The patchboard is populated and attached to the RNB board with hot melt glue after applying several haywires. The power converter for the CI-V interface is a little anemic and an updated circuit is supplied on a patchboard to replace the existing circuit. A few parts are removed from the RMB and the patchboard is used by glueing a transformer with the patchboard attached to the RMB board.

7 Connections

All connectgions to the outside world occur on the back panel as there isn't any room on the front panel with the LCD , LEDs, buttons and switches.

7.1 Radio Interface

Conenctions for the radio interface

7.1.1 Power

Power to the interface may come through either the motherbaord or the daughterboard. The power connector may be through either a 5.5mmx2.5mm or a 5.5mmx2.1mm coaxial power connector. Both power connectors are provisioned with reverse voltage protection, surge suppression, and over current protection. In addition we may populate the boards with differing power connectors to allow either connector size to be used (this would make power supply selection less painful as either size is accommodated.

Although a downstream power user may be plugged into the unused connector to send power to an additional device, use care so as not to overload the input protection network and cause the polysilicone fuse to open.

Although a 12V supply will work, there is a preference for a 9V supply to reduce power dissipation in the regulators.

7.1.2 USB

The USB connection to the host system is through the USB connector on the lower board (i.e. the motherboard or RMB board). The USB connector on the top board would not normally be installed unless there was a reason to operate the analog board by itself. Once the boards are integrated the series resistors in the USB data line would be removed from the top board rendering the top USB connector inactive.

The USB connector is a standard size B connector, the use of mini-USB or micro-USB has been avoided to keep the mechanical design as robust as possible.

7.1.3 RS232

The RS232 channel is presented on a DE9P connector with pinouts to match the connector traditionally used on PCs. Normally the board would have been populated with an isolated driver that only provides for the data lines and the RTS/CTS lines. The other lines are not connected in this configuration and the serial device has the unused pins terminated to their proper active states (i.e. the RI signal is inactive)

RAD015 board have a male connector with the pinouts that would be appropriate for a female connector. In this cas a straight through male to mal connector would be indicated.

7.1.4 ICOM CI-V

Determine if this interface will be used as a CI_V interface with an ICOM or similar radio, or as a full duplex interface for another application. There is a jumper on the RMB that needs to be installed to force half duplex operation.

With the jumper in place there is quite a bit of flexibility in cable selection as either a mono 3.5mm or a stereo 3.5mm patch cord may be used to connect to the CI_V interface on an ICOM radio. If a custom cable is to be fabricated, with a stereo jack on one end and a mono jack on the other, the half duplex bonding may occur in the cable allowing the jumper on the RMB to be left out. In this case the interface can be switched from half duplex to full duplex operation with the connecting cable.

The CI-V being an open collector configuration requires a pull-up. The latest parts selections has a 1K resistor value for the pull-up which appears to allow reasonably reliable communications with the radio at 19,200 bits/second.

7.1.5 SPI/JTAG

The 2nd channel of the FT4232 is presented as raw pins on this connector. There are several utilities available on the web to allow the interface to be used to access the JTAG ports on various devices. (such as de-brixking Linksys wireless routers)

7.1.6 Reset

Not a connector, but it is on the back panel. This is the reset button for the eZ8. Poke it if you're frustrated, but once the software is stable it should not be needed.

The SWR/Power Meter has the same reset button on its back panel.

7.1.7 Radio RJ45

The RJ45 connector is the primary interface to your radio. This will be the only connector required to access your radio if not using one of the serial connectors. A jumper block is installed just inside of the connector to alter the pinout for your specific radio.

Cables for the signalink interfaces may be used to eliminate the need to fabricate cables. Although the programming headers from Tigertronics are pin compatible, they are not mechanically compatble as the socket connectors are intended for 0.025” square (wirewrap) posts. These connectors are more rugged that a DIP socket and wire wrap posts may be directly installed in the sockets allowing the use of Kynar wire (wire-wrap wire).

7.1.8 Radio KEY

The 1/4” phone jack adjacent to the RJ45 connector allow connecting to the keying circuit of a tube radio. This keying connector makes use of a relay contact closure that is insensitive to polarity. Jumpers on the RDM must be correctly installed prior to using this connector.

This is also the connection used for controlling a reflow oven.

7.1.9 Tx Audio Monitor

The pair of 3.5mm stereo jacks on the RDB board are the audio monitors. The jack closest to the ¼” phone jack is the transmit audio monitor. Any common stereo speakers may be plugged into this jack to listen to the audio signal being sent to the radio.

This monitor is taken directly from the CODEC, before the gain control circuit so changing gain does not affect its level, it is expected that the volume control knob on the monitor speaker is the best place to control the volume. This monitor is also buffered so as not to present a changing load to the CODEC and gain circuit when the attached speakers are switched on and off or removed and reinstalled. Also note that the monitor is is buffered with a unit gain opamp so the levels presented are essentially what the CODEC generated which will be about 1V RMS..

7.1.10 Rx Audio Monitor

The second 3.5mm jack is the receive audio monitor, As with the transmit audio monitor, this is attached between the gain stage and the audio CODEC. As this is the receive path, the gain control stage does affect the signal level at the monitor jack. As with the transmit audio monitor, the receive audio monitor is buffered so changes in the load impedance should not affect signal levels at the CODEC.

Also keep in mind that both the receive monitor and the transmit monitor are unity gain. This means that to receive a signal that is the same level as the transmit signal, the volume from the receive monitor should be about the same as the transmit monitor. Also note that the unity gain means that an oscilloscope can be used on the monitor ports to inspect the audio signals.

7.2 SWR/Power Meter

Conenctions for the radio interface

7.2.1 J1 and J2, Signal

J1 (SO239) is the connection towards the transmitter.

J2 (SO239) is the connection towards the antenna.

7.2.2 P100, Power

Conenctions for the radio interface

7.2.3 J641/J642

RS485 bus.

The power bus from P100 is also connected to both J641 and J642 to provide power to a remote SW Meter.

7.2.4 J920/J921, CI-V Bus

These connectors are the CI-V bus. Two jacks are provided to allow a bus to easily be formed.

8 Host Interface

The 3rd serial interface on the FT4232 connects to the eZ8. This serial channel is used to communicate with the eZ8 that is primarily used to implement the VOX function and to control the audio gain to and from the radio.

In addition to the off-line control functions, the host interface is also used to configure the radio interface operation. Gain levels, trigger levels, oven programs and CW activities are all accessible through the host control interface.

8.1 Radio Interface Commands

Commands to the eZ8 are generally limited to 32 characters plus a terminator. Either a carriage return (0x0D) or a linefeed (0x0A) may be used to terminate a command buffer with the carriage return being preferred. The eZ8 will answer back with some text to indicate the success or failure of the command.

When the eZ8 receives any command, the host status display flag is cleared before processing the command. This will suppress the status display to the host if this feature is enabled to allow the command to be viewed.

8.1.1 Command: DUMP

No arguments.

Dump the flash file system. This produces one line for each active record in the file system. The internal flash area is dumped first followed by the external flash system. The internal/external flag follows the colon in the record dump.

Exammple:

```
dump
STS16,000:I LB=INIT,V1.1 Radio Setup
STS16,001:I RT=INIT,22,50,50,15
STS16,002:I RS=INIT,45,45,45,45,50
STS16,003:I CS=INIT,25,1,3,7
STS16,004:I CK=INIT,K1,K0,L0,BZ
STS16,005:I DS=INIT,1,0,1,100
STS16,017:I OS=LEAD000,1,10,PHONE+HEAT
STS16,018:I OS=LEAD001,1,10,BEEP,pre heat
STS16,019:I OS=LEAD002,1,10
. . .
STS16,105:I OS=LEAD088,5,105,, door open
STS16,106:I OS=LEAD089,5,100,, door open
STS16,108:I CF=START,KC0JFQ Radio Interface
STS16,109:I CF=SWDATE,Feb 10 2014
. . .
STS16,126:I CF=END,Configuration area end
STS17,000:E CW=<call> <call> 599 K
STS17,001:E CW=<call> 599 TU
STS17,004:E LB=JT65,JT65 and JT9
STS17,005:E RT=JT65,22,150,150,150
STS17,006:E RS=JT65,30,30,50,50,10
STS17,007:E LB=PSK31, FLdigi
```

STS17,008:E RT=PSK31,22,50,50,50

STS17,009:E RS=PSK31,45,45,45,45,10

STS15,117:

8.1.2 Command: READ

String argument.

Read a named record or group of records from the flash file system.

Simple string matching is used to select records from the flash file system. The string may occur anywhere in the record and is case sensitive.

The following command reads all the INIT records and any other record that happen to contain the “INIT” string:

```
read,INIT
STS12,000:I LB=INIT,V1.1 Radio Setup
STS12,001:I RT=INIT,22,50,50,15
STS12,002:I RS=INIT,45,45,45,45,50
STS12,003:I CS=INIT,25,1,3,7
STS12,004:I CK=INIT,K1,K0,L0,BZ
STS12,005:I DS=INIT,1,0,1,100
STS12,002:E LB=DUMMY, INIT
STS12,7:
```

For this example, a dummy label record was added to the external file system to illustrate the simple string matching that is used in the read command.

8.1.3 Command: SAVE

String argument.

Save a setup record into the external flash file system

8.1.4 Command: ERAS

String argument.

Erase a setup record in the external flash file system.

8.1.5 Command: CLRN

Numeric argument.

Clear the external flash file system

8.1.6 Command: PING

No arguments.

Similar to the ping used to perform a quick connectivity test, this command simply echos back a status to indicate the system is alive and operating.

8.1.7 Command: HANG

Three numeric arguments.

Sets the transmit hang timer. The hang timer is used to prevent the VOX code from dropping the transmitter in the middle of a transmission. The A/D sampling is not terribly sophisticated, the 16 most recent samples are saved in order to calculate the mean value of right transmit channel. A minimum and maximum are extracted from these samples and a deviation is calculated. When this deviation is above the VOX trip point (see the VOXL command), the transmitter keying timers are set to the specified values. Every 10 mS the transmitter keying timers are decremented until reacheing zero. At this zero point, the transmitter key is deasserted. The A/D polling loop and the keying timer are not synchronized, so this hang timer allows the sampling loop to run at a slower pace than the timer. In addition, this allows the A/D sampling loop to encounter a string of mid-range values (such as when a PSK phase change occurs) without unkeying the transmitter. Also note that there are three distinct timers, one of which is currently unused. The first two timers control the optically isolated darlington (first argument) and the reed relay (second argument) which may be set independantly.

This value must be kept large enough to avoid unintended keying farts, but small enough that the transmitter doesn't remain on for too long of a period after the transmission ends.

The idle level on the transmit channel is continuously recalculated so the software driving the CODE is not required to set the last DAC value to any particular value. If the DAC is left set to a value far away from mid-scale, the software will adjust after a maximum of 16 samples have been collected.

8.1.8 Command: TXVL

Two numeric arguments.

Sets the transmit channel gain. The transmit channel gain is controlled by a digital pot in the feedback loop of the buffer amp circuit. The transmit level can be set independently for right and left channels should that be necessary. If only one value is specified in the command it is applied to both channels. When two values are supplied they are for the right and left channels respectively.

8.1.9 Command: RXVL

Two numeric arguments.

Sets the receive channel gain. This works in exactly the same manner as the transmit channel gain. Right and left are independent although they are typically treated as a single entity. When two values are supplied they are for the right and left channels respectively.

8.1.10 Command: VOXL

Numeric argument.

Sets the audio deviation required to key the transmitter. Se the HANG command for a discussion of VOX management.

8.1.11 Command: CWPM

Numeric argument.

Sets the code rate in words per minute. This effectively sets the length of a dot with the dash being 3 times as long as a dot.

8.1.12 Command: CTIM

Three numeric arguments.

This command sets the code generation timing parameters. There are threee parameters; an inter bit timing, and inter character timing, and an inter word timing. All are expressed in terms of dot length.

8.1.13 Command: CALL

String argument.

This command changes the callsign until the next reset from the read panel. Affects the callsign used in stored code buffers.

8.1.14 Command: CODE

String argument.

This command sends code. The text body of the message is sent using the current code generation settings.

This size of the message is limited by the input buffer. Since the eZ8 has less than 4KB of storage space available, the maximum length is set to a value well below 100 bytes. If a command is sent that exceeds the buffer size, all of the text at the end of the message is discarded.

8.1.15 Command: TEXT

String argument.

This command sends one line of text to the LCD. The LCD has only 16 bytes of display area per line, so messages should not exceed about 21 bytes.

8.1.16 Command: DISP

Numeric and string arguments.

This is similar to the TEXT command, but requires the X and Y position be sent before the message. As with the TEXT command above, send only 16 bytes of message text in this command.

8.1.17 Command: TIME

Numeric argument.

This command load a value into the internal seconds counter. This is intended to allow a timetagged list of oven temperatures. There is no other requirement that the time be set.

The value sent in this command is saved in the seconds field that is updated every second. No attempt is made to perform precision synchronization, it is intended to be accurate to a few seconds.

The eZ8 uses a 10mS timer to update the seconds clock. The accuracy is, therefore, dependent on the clock stability in the eZ8 and on the eZ8 not missing interrupts.

8.1.18 Command: DSPY

No arguments.

This command simply toggles the *host flag* and is intended to be used as an interactive means of changing the control flag.

8.1.19 Command: ECHO <echo>, <host_flag>, <lcd_flag>

Numeric arguments.

This command sets or clears the echo control flag and the display flags. All three arguments are required for unambiguous execution of the command, any missing arguments are replaced with the last argument supplied. This means that *ECHO, 1* will turn on all 3 flags.

8.1.19.1 <echo>

The *echo* flag should typically be set for interactive use to allow the eZ8 to echo keystrokes back to the host. When software is commanding the eZ8 it can be left off to eliminate superfluous traffic that may interfere with decoding status messages.

8.1.19.2 <host_flag>

When the *host flag* is set, the status display is routed to the host. This would be useful is the unit is built without a front panel display.

8.1.19.3 <lcd_flag>

When the *LCD flag* is set, traffic destined for the LCD is actually sent to the LCD. Clearing this flag will speed up the eZ8 slightly.

8.1.20 Command: KEYN, <key>, <key>, <key>

String arguments.

This command may be used to grossly key the radio. Since there is a long interaction path between the host and the keying circuit through the eZ8 logic, this can not be used to send code at any normal rates (the CODE command would be used to send code traffic through the eZ8 and normal CW speeds). This was added to the eZ8 software to support the *HI JUNO* program in October of 2013. Any combination of the *key* values may be taken from the following table.

<key>	**Function**
<blank>	Opens the keying circuits and silences the Sonalert
KEY0 Key0 key0	Closes the darlington keyer through J800
KEY1 Key1 key1	Closes the relay keyer through J820
BEEP Beep beep	Sounds the Sonalert

Note that the strings are case sensitive. *KEy1*, for example will not be recgonized.

8.1.21 Command: DSTS

No arguments.

This command displays the system settings.

Example:

```
dsts
DSTS Version RADV1.43 Z8F6421AN020SG
DSTS Built Oct 22 2013 19:17:39
DSTS Time   0x00000434
DSTS Call KD0JHW
DSTS Name Bill
DSTS Host status_flag=0
DSTS LCD enable_flag=1  intensity=100
DSTS I/F Setup "command"
DSTS Tx Hang  [K0]=50 [K1]=50 [2]=15
DSTS Tx Host   VOL-idle
DSTS Tx Gain  L=45 R=45
DSTS Rx Gain  L=45 R=45
DSTS VOX Dev=22
DSTS CW WPM=25 Gaps: Bit=1 Chr=3 Wrd=7
DSTS CW key mask  L0 K1 K0 BZ
STS31,0
```

8.1.22 Command: CIVD

No arguments.

This dumps the CIV buffer.

The dump routine understands the ICOM CI-V protocol. It looks for the 0xFD character at the end of the command to terminate the buffer (it will not work as a general line monitoring routine).

8.1.23 Command: CIVM

This continuously dumps the CI-V buffer. Use a <CR> to exit the monitor mode.

While the monitor is running, the eZ8 is dedicated to this activity so the VOX stops operating.

The dump record consists of a three digit sequence number, a timestamp that cycles at 100 seconds, and a delta between packets. This is followed by a hex dump of the CI-V buffer. Each CIV buffer has two header bytes (0xFE 0xFE), a destination byte, a source byte, the CIV buffer and a termination byte of 0xF0 through 0xFD).

Sample dump:

```
civm
    1  1.180 +0.040 FE FE 76 E0 04 FD
    2  1.190 +0.010 FE FE E0 76 04 01 02 FD
    3  1.240 +0.050 FE FE 76 E0 1A 04 FD
    4  1.260 +0.020 FE FE E0 76 1A 04 00 00 FD
    5  1.310 +0.050 FE FE 76 E0 1A 02 FD
    6  1.330 +0.010 FE FE E0 76 1A 02 40 FD
    7  1.380 +0.060 FE FE 76 E0 15 02 FD
    8  1.390 +0.010 FE FE E0 76 15 02 02 04 FD
    9  1.450 +0.050 FE FE 76 E0 14 01 FD
   10  1.460 +0.020 FE FE E0 76 14 01 01 70 FD
```

The first column is simply a monotonically increasing sequence number. It rolls back to zero after 999, The second column is a second field that rolls back to zero at 99 seconds. The timer that is used to produce the sub-seconds field is a 10mS tick, so the unit millisecond is always zero. The third column is the delta-T between packets. As with the time tag, the unit millisecond field is always zero (in other words, it hold no significance as the eZ8 does not keep track of time to any finer resolution. The CI-V packet hex dump follows, 2 hexadecimal digits per column.

8.1.24 Command: SAVE

String argument.

Save a setup record into the external flash file system

8.1.25 Command: setup

Noarguments.

Setups can be loaded from flash by simply entering the label name. As there is a default INIT setup in the file system, there is always an INIT command present. Loading additional setups into the external flash effectively extends the command set.

Here are two example setups:

```
LB=INIT,V1.1 Radio Setup
RT=INIT,22,50,50,15
RS=INIT,45,45,45,45,50
CS=INIT,25,1,3,7
CK=INIT,K1,K0,L0,BZ
DS=INIT,1,0,1,100

LB=JT65,JT65 and JT9
RT=JT65,22,100,100,100
RS=JT65,30,30,50,50,10
```

The INIT set is used when the system initially starts, i.e. it forms the default configuration. The audio gain is set to near unity gain and the programming header provides the correct attenuation for this to operate correctly with PSK31 using Fldigi. When using JT65/JT9 the levels presented to the radio are too high for clean operation. Reducing the gain to 30 provides the correct drive for that mode/software combination. Simply entering “JT65” or “INIT” will switch the setting to that mode. Keep in mind that this can also be accomplished using the front panel.

8.1.26 Command: HELP

This command lists all the currently implemented commands along with a short clause indicating what they do.

Example:

```
help
HELP DUMP dump flash records
HELP SAVE save 1 record (record image)
HELP READ read flash records (1 text)
HELP ERAS erase flash device
HELP CLRN clear flash record (1 text)
HELP PING aliveness echo
HELP HANG VOX hang timer (1 numeric)
HELP TXVL transmit level R,L(1 or 2 numeric)
HELP RXVL receive  level R,L(1 or 2 numeric)
HELP VOXL audio deviation (1 numeric)
HELP CWPM CW parameters wpm, (1 numeric)
HELP CTIM CW parameters inter-bit. inter-char, inter-word, (3 numeric)
HELP CODE CW buffer to send (text buffer)
```

```
HELP TEXT text to LCD (text buffer)
HELP DISP text to LCD (2 numeric, text buffer)
HELP TIME set time (1 numeric)
HELP ECHO set echo flag (1 numeric)
HELP DSTS dump audio status
HELP HELP help text
STS15,0
```

8.2 SWR/Power Meter Commands

This is the command list for the SWR/Power Meter

8.2.1 Command: ALRM

This command configures the annunciator.

Example:

8.2.2 Command: CIV0

This command sets the message traffic that is delivered over the CI-V bus.

Example:

8.2.3 Command: CIV1

This command set the host and radio address bytes.

Example:

8.2.4 Command: CIV2

This command sets the CI-V message forwarding to allow multiple SWR/Power Meters to communicate over the RS485 bus. One meter may then be configured to pass the combined traffic over the CI-V bus.

Example:

8.2.5 Command: CIV3

This command sets the trip point and message timing. The trip point is the level below which CI-V traffic is not generated.

Example:

8.2.6 Command: CIVM

This is the CIV Message forwarding command. It consists of the command key followed by a HEX-ASCII CI-V message.

Example:

CIVM FD FD E0 97 12 34 56 FE

8.2.7 Command: CIVR

This command is part of the CIV Message forwarding scheme. It requests that a dump of the current set of CI-V messages be dumped back to the RS-485 port on which thew command arrived. This command facilitates polling.

Example:

CIVR

8.2.8 Command: CLRN

This command clears a numbered record in the external flash file system. It works the same way as the radio interface command of the same name although the SWR/Power Meter uses 64 byte records in the flash file system to accommodate loading the coefficients tables.

Example:

8.2.9 Command: COEF

This command dumps one of the coefficients tables.

Example:

8.2.10 Command: DSPY

This command is similar to the radio interface command. It controls what is displayed through the RS485 port.

Example:

8.2.11 Command: DSTS

This command is similar to the radio interface command.

Example:

8.2.12 Command: DUMP

This command is similar to the radio interface command. The file system records are 64 byte long, so there are half as many available as there are on the radio interface.

Example:

8.2.13 Command: HELP

This command is similar to the radio interface command.

Example:

8.2.14 Command: INFO

This command dumps the “CF=INFO” records in the file system.

Example:

8.2.15 Command: PING

This command is similar to the radio interface command.

Example:

8.2.16 Command: POLY

This command loads a single entry in the coefficients table. Following the command key are from for to six numeric values. The first two numbers identify the table and the channel to load. The remaining two to four numbers are the polynomial coefficients for the specified channel. The numbers are laid out with the 0^{th} order coefficient appearing first followed by the first order coefficient, second order coefficient, and finally the third order coefficient. The coefficients are floating points and should be expressed using c-style exponential notation. Note that the line length is limited to a maximum of 63 characters and that this command will probably be embedded as part of the initialization stored in the external file system.

See the save command for additional restrictions that apply to forming this command

Example:

8.2.17 Command: READ

This command is similar to the radio interface command.

Example:

8.2.18 Command: SAVE

This command is similar to the radio interface command.

Example:

SAVE SC=POLY,0,0,0.00,1.00,0.00,0.00

8.3 Setups

The external flash device connected to the eZ8 is used to store radio setup strings. The largest available device with a 32 byte page is 32Kb. This page size determines the size of each setup record that is held in the flash device. Each setup record effectively has a record type and a file name to allow implementing access routines that can get at the information in the falsh device in a regualr manner. Details of the file system are in another section.

The setups are grouped into several types as listed below. All records in the flash file system consist have type key delimited with an equal sign(=) as the beginning of the record. Other than a code message records also have name delimited with a comma the immediately follows the type key. This is, followed by body of the record which usually consists of several setup parameters.

8.3.1 Type LB, Label Record LB=<name>,<text>

The **Label** record is used to select a group of identically named setup commands to be decoded and used to configure the interface. A complete setup consists of a label record followed by the RS, RT, CS, and CK records. All of the records in the group have the same name. Not all of the records need be present but do keep in mind that only those settings that are specified are changed.

The <text> field is displayed on the LCD. It may contain anything that the user finds convenient.

8.3.2 Type DS, Display Setup

DS=<echo>,<host_status>,<lcd_status>,<lcd_intensity>

The Display Setup record is used to configure the manner in which the LCD is used. This also allows status to be displayed on the host serial port should the LCD not be available.

8.3.2.1 <echo>

Host port character echo.

8.3.2.2 <host_status>

Enables status traffic on the host serial port.

8.3.2.3 <lcd_status>

Enqbles status traffic on the LCD.

8.3.2.4 <lcd_intensity>

Tghis is an 8-bit unsigned number representing the intensity level for the LCD backlight. A value of 255 is full brightness for the backlight. Smaller values reduce power dissipation in the regulators (and slightly reduce the heat produced by the regulators).

8.3.3 Type RJ, Programming Jumper Record

RJ=<name>,<card>,<TxAtten>,<RxAtten>

The **Radio Programming Jumper** record is a s documentation aid that has the jumper module configuration associated with a particular radio. The *<card>* field identifies the jumper module used with this particular radio and the *<TxAtten>* and *<RxAtten>* indicate the resistor values used in the attenuator pads on the jumper module.

8.3.4 Type RS, Label Record RS=<name>, <TxL-left>, <TxL-right>, <RxL-left>, <RxL-right>, <USB-vol>

The **Radio Setup** record holds gain control settings. Although most radios will make use of only the right channel, the right and left channels may be set independently. Typically the right and left values will match and there is no shorthand to reduce the number of parameters.

8.3.4.1 <TxL> and <RxL>

Enqbles status traffic on the LCD.The values for the volume controls are nominally about 50 for unity gain through the buffer amps. The range of values is an 8 bit unsigned integer so 0 to 255. Peak levels that may be obtained are limited by the 5V supply rail and the offset of the VCOM level out of the PCM2906. The PCM2906 VCOM level is typically a little below 2.5V so the maximum peak-peak swing is a bit less than 4V.

8.3.4.2 <USB-vol>

Enqbles status traffic on the LCD.The radio interface also controls the volume-up button on the PCM2906. Setting this field to a value greater than 8 will cause the radio interface to pulse the volume-up control following a reset. Larger numbers will extend the length of time that the volume up control is pulsed. This is intended to drive the volume control to full without having to go into the Windows setup menu. It appears to function as intended in the Linux environment as well. The LEDs on the RDB board alternate as the volume is driven to full following reset.

8.3.5 Type RT, Label Record

RT=<name>,<trip>,<hang0>,<hang1>,<hang2>

The **Radio Trip** record sets the audio deviation required to cause the eZ8 to key the radio and the length of time after the audio goes quiet that the radio will remain keyed.

The audio deviation is in counts from the 10 bit A/D system and may range from 0 to about 500. Note that the audio CODEC outputs about a 1VRMS (about 1.4V Pk-Pk) signal and the A/D range is 3.3V so the maximum deviation that will be seen is much lower than 500 counts.

The eZ8 uses an average of the last 16 samples from the A/D to set the zero level, so if the CODEC stops generating audio, the radio will stop transmitting. In other words, the idle level is continuously calculated so the DAC in the CODEC does not need to be stopped at a mid-scale value.

The hang timers are expressed in timer ticks. The timer used to control the VOX hang function is programmed for a 10mS tick, so these are units of 1/100 of a second. The *hang0* timer controls the optically isolated darlington which will be used to key the radio in most cases. The *hang1* timer controls the relay, which is typically not used. The *hang2* timer is currently uncommitted, so the associated time value is effectively useless so it should be set to a small number or to the same value as the othere hang timers.

8.3.6 Type CS, Label Record

CS=<name>,<wpm>,<bit>,<char>,<word>

The **Code Setup** record sets the timing parameters for the code generator. The <wpm> is the word per minute rate the code generator will run at. It can range from 5 to about 50. This number sets the timing of the dots. Dashes are fixed at 3 dot times. The inter-bit, inter-character, and inter-word timings are set by the remaining 3 parameters and are expressed I terms of dots.

8.3.7 Type CK, Label Record CK=<name>,<mne>.<mne>...

The **Code Keying** record controls which of the keying controls are to be used to send code. Mnemonics from the following table are comma separated to specify keying control.

mnemonic	Control
A1	Audio w/keying channel 1 Key-1 is on continuously and an audio tone is placed on the right audio out channel
A0	Audio w/keying channel 0 Key-0 is on continuously and an audio tone is placed on the right audio out channel
L1	Top LED Function LED is continuously on while transmitting a code message
L0	Bottom LED Function LED is continuously on while transmitting a code message
BZ	Sonalert on-board buzzer annunciates the code message
K1	Keying channel 1 This is the reed relay Slower than the darlington but higher voltage
K0	Keying channel 0 This is the optically isolated darlington. Faster on and off than the relay

8.3.8 Type CW, Label Record CW=<code text>

The **Code** (Continuous Wave) record hold a short message that can be selected and transmitted. Since these records are displayed as part of the selection processing, there is no need for a name so no name space is wasted in the record.

8.3.9 Type CF, Configuration Record CF=<field>,<value>

The **Configuration** records hold setup information for the system such as serial numbers and owner. Some of the fields may be effectively changed by writing new records in the external flash, and some of the records are only retrieved from internal falsh so attempts to change the values are ineffective.

field	INT	EXT	description
START	Y		This record identifies the device. This string is displayed on the LCD and sent through the serial channel to the host when the system starts.
SWDATE	Y		This string is the date when the software was built.
SWTIME	Y		This string is the local time when the internal file system was compiled.
SWVERS	Y		This string holds the software version.
HWDATE	Y		This string holds the date when the device was initially programmed with its serial number and owner strings.
<ownr>	Y		This string is the owners name.
<call>	Y	Y	This string is the owners callsign.
<name>	Y	Y	This string is the owners first name.
<zitu>	Y	Y	This string is the ITU zone you are transmitting from.
<z_cw>	Y	Y	This string is the CW zone you are transmitting from.
<locn>	Y	Y	This string is the maidenhead locator you are transmitting from.

<bser>	Y		This string is the board serial number. It should match the serial number in the FT4232 device.

8.3.10 Type SC, Setup Codmmand SC=<command>

The SWR/Power Meter scans the flash file system at startup looking for SC records and feeds the text field to the command processor as if the commands arrived through the RS485/debug port. This allows the meter to be configured using the debug connection so it will behave as desired at the next reset.

8.3.11 Type OS, Oven Setup Record OS=<name><seq>,<time>,<temp>,<control>

The **Oven Setup** record holds a step in the oven operating profile. The <name> will be either “LEAD” or “L20C” and the <seq> is monotonically increasing decimal number that runs from 000 up to less than 127 (as there are only 127 locations in the external file system.

The <time> field is the number of seconds the specified temperature will be held, and the <temp> is the target temperature. The <control> field appears only on the first record to specify the channel that will be used to enable the ovens heater.

Released software versions do not include the oven control program in the internal flash, so it must be loaded into the external flash to make use of the oven control feature.

8.4 Oven Programs

These are the steps required to configure the reflow oven control loop. The specify time and temperature as well as select the control that will be used to control the oven (i.e. the relay or the darlington).

The prototype oven has a solid state relay the r3quires 5 volts to activate the heater.

9 Front Panel Controls

The front panel consists of an LCD display with an LED backlight two buttons and two center-off toggle switches. The toggle switches with center-off provide for 9 unique settings that are broken into 3 groups; one operating position with both switches off, one setup group where one of the two switches is moved off center, and a group of 4 positions where bot switched are off center that has special setups and operating modes.

9.1 LCD Display

The LCD display is available for use by host software through the command structure. Two commands are available to place text on the display. The DISP command has positioning parameters to allow the text to be placed anywhere on the screen and the TEXT command simply feeds the test from the host directly to the display.

Without the eZ8 software the LCD display cannot be accessed from the host.

9.2 LED Indicators

The bottom 2 LEDs are controlled by the modem control line on the 3rd serial channel of the FT4232. This is the channel used to communicate with the eZ8.

9.3 Buttons

The front panel buttons are accessible to the host through the 3rd serial channel of the FT4232 on 4 of the modem status lines. This is the channel used to communicate with the eZ8.

Two of the modem status lines are directly connected to the button closure and reflect the real-0time status of the buttons. If the host is not able to poll the serial chip fast enough button transitions can easily be missed. To eliminate the real-time requirement, each button press is latched and made available on the remaining two modem status lines. The button status remains latched until there is activity on the serial data to the eZ8 (poll before talking to the eZ8)

9.4 Switches

The switches on the RDB board are connected to the modem status lines on the first channel of the serial controller. Since the switch position is controlled by fumbly fingers there is no need to latch them, they are simply the real time status of the switches.

Left Switch	Right Switch	Select

center	center	Normal operation This allows the software to operate normally. VOX control uses the right transmit channel to determine when to key the radio.
center	left	Transmit Gain Adjust This allows the software to operate normally. VOX control uses the right transmit channel to determine when to key the radio. The buttons are used to adjust the transmit channel gain.
center	right	Receiver Gain Adjust This allows the software to operate normally. VOX control uses the right transmit channel to determine when to key the radio. The buttons are used to adjust the reveive channel gain.
left	center	VOX Trip Point Adjust This allows the software to operate normally. VOX control uses the right transmit channel to determine when to key the radio. The buttons are used to adjust the VOX trip point.

right	center	CW Operation (no VOX) This software will transmit internally store code messages. VOX control is disabled. Left button scrolls through available messages and the right button transmits them.
left	left	(unused) This allows the software to operate normally.
left	right	Oven Control (no VOX) The software that controls a reflow oven is active. VOX control is disabled. Right button selects a 235C lead profile and the left button selects a 265C lead-free profile.
right	left	Setup Select (no VOX) This allows the software to operate normally. VOX control is disabled. Left button scrolls through available setups and the right button activates the selection.

right	right	Receive Level Monitor (no VOX) The software switches the A/D channel to monitor the right receive channel. Limitations in the A/D architecture make it difficult to monitor more than one A/D channel on a continuous basis, so the transmit channel isn't monitored in this mode (hence the lack of VOX control).

Each of these modes is enumerated in detail below.

9.5 Mode 0: Normal Operations center-center

Left Switch Position	Right Switch Position
◁ ■ ▷	◁ ■ ▷
Left Button Function	**Right Button Function**
Key Radio	Pulse MUTE on CODEC

For digital modes, both switches are left in the center position. The interface is sensitive to host commands. The eZ8 monitors the transmit level to implement the VOX function. Host may initiate CW messages.

9.6 Mode 1a: Receive Gain Adjust center-right

Left Switch Position	Right Switch Position
◁ □ ▷	◁ □ ▶
Left Button Function	Right Button Function
Increase Gain 10 steps	Decrease Gain 3 steps

Buttons change the receive gain level. Right and Left channels are set to the same levels. Display shows the audio deviation (VOX) value and the transmit and receive gain settings. The eZ8 monitors the transmit level to implement the VOX function. Host may initiate CW messages.

9.7 Mode 1b: Transmit Gain Adjust center-left

Left Switch Position	Right Switch Position
◁ ■ ▷	◀ □ ▷
Left Button Function	Right Button Function
Increase Gain 10 steps	Decrease Gain 3 steps

Buttons change the transmit gain level. Right and Left channels are set to the same levels. Display shows the audio deviation (VOX) value and the transmit and receive gain settings on the bottom line in the same format as for the previsous switch configuration. The top display line shows the transmit deviation numbers. The eZ8 monitors the transmit level to implement the VOX function. Host may initiate CW messages.

The deviation display is a real-time monitor of the modulation sent from the host, or in other words, and indication of what the volume control slider is set to, if deviation is down, it is because the host computer has its levels reduced.

9.8 Mode 1c: VOX trip Adust left-center

Left Switch Position	Right Switch Position
◀ □ ▷	◁ ■ ▷
Left Button Function	Right Button Function
Increase Level 10 steps	Decrease Level 3 steps

Again, the buttons change the audio deviation setting that controls the VOS function. The zero point is continuously recalcualted, so there is typically no need to alter this setting, when the CODEC is idle the deviation falls to a small value (typically just a few counts) and when transmitting the deviation is relative quite large (well over 100) so the default value of 20 appears to work well. The top display line shows the transmit deviation numbers The eZ8 monitors the transmit level to implement the VOX function. Host may initiate CW messages.

9.9 Mode 1d: front panel CW right-center

Left Switch Position	Right Switch Position
◁ □ ▶	◁ ■ ▷
Left Button Function	Right Button Function
Scroll through messages	Send Message

This switch position **disables the VOX function** and places the interface in a mode that allows the code generator to run, The eZ8 remains sensitive to commands from the host.

The left button scrolls through the *CW=* records in the flash file system.

The right button begins code generation for the currently displayed message.

Message length is limited to 29 characters (the *CW=* string having been removed from the 32 byte record).

9.10 Mode 2a: Receive Level Monitor right-right

Left Switch Position	Right Switch Position
◁ □ ▶	◁ □ ▶
Left Button Function	Right Button Function
Key Radio	

This switch position switches to monitoring the receive signal level. We see the same type of deviation numbers that were seen on the transmit gain adjust. Due to limitations in the eZ8 A/D controller, this mode inhibits the VOX circuit.

This mode may be used to adjust the receive levels without having the host computer operating the CODEC.

Also note that the left switch can be moved to the center position to adjust the receive gain level and back to the right to resume level monitoring.

9.11 Mode 2b: Radio Setup right-left

Left Switch Position	Right Switch Position
◁ ☐ ▶	◀ ☐ ▷
Left Button Function	Right Button Function
Scroll through setups	Select Setup

This switch position scrolls through stored radio setups with the left button. When the desired setup is seen it can be activated with the right button.

The setups may have as much or as little setup information as required. That is to say a setup need not contain all of the setup records.

The audio and VOX settings are displayed on the front panel.

9.12 Mode 2c: unused left-left

Left Switch Position	Right Switch Position
◀ ☐ ▷	◀ ☐ ▷
Left Button Function	Right Button Function

This switch position is currently not used

9.13 Mode 2d: Reflow Oven Operation left-right

Left Switch Position	Right Switch Position
◀ ☐ ▷	◁ ☐ ▶
Left Button Function	Right Button Function
	Start LEAD Reflow

This switch position is used to control a reflow oven. A thermocouple is connected to an internal connector on the RDB board. The thermocouple monitors the internal temperature of the reflow oven and modulated the control circuit that drives the heater to implement the temperature profile set by the oven setup.

The left button runs the lead-free profile (which must be loaded into the external flash file system to be used).

The right button runs the lead based profile (which is loacated in the internal file system).

.

10 Oven Control

The radio interface is equipped with an SPI thermocouple interface that allows the eZ8 to read the temperature of a thermocouple. The type of thermocouple is driven by the selection of the specific variant of the MAX31855 that is installed on the RDB board. Initial testing was performed using a type-K thermocouple.

10.1 Preparations

The oven selected for prototyping is a Black & Decker model ??? that has been refitted. The thermostat supplied with the devices is replaced with a solid-state relay that is, in turn, connected to the radio interface through the 1/4” phono jack keying connector. The RDB board must be configured to supply a switched 5V keying signal to the solid state relay.

Apply the following jumper configuration to allows the RDB board to be used to control the oven.

.

10.2 Reflow Profile

The reflow profile is stored in the internal file system in a file called “LEAD”. And additional profile may be downloaded under the name “FREE”

10.3 Thermal issues

The heating elements in the oven have considerable heat storage capability that must be managed by altering the reflow profile, the control software is not terribly sophisticated as it is limited in size and does not manage the oven characterics internally.

10.4 Reflow Operation

The reflow oven must be below 50C and the controller must be at room temperature or the software will not initiate a reflow cycle. The left button initiated a temperature profile for lead solder and the right button initiated a profile for lead-free solder. Once the cycle begins it may be intgerrupted by changing the switches on the front panel or if a fault condition occurs in the MAX31855 (this would be an issue with the thermocouple probe).

10.4.1 Display

The display shows the current reflow program step, a status indicator and the heat control state on the first line and the oven and room temperatures on the second line. As the temperature profile steps through the display is updated to indicate the step number of the program that is currently being processed.

The eZ8 is also sending regular updates to the USB serial chip on the motherboard. Port-C traffic can be saved to a file as a record of the reflow operation. If the system time has been set on the radio controller the timetags in the file will be reasonably accurate.

10.4.2 Status Reporting

When switched for reflow oven operation, the host serial port receives status updates on a regular basis. Once per second a status report is sent in what is essentially a ***.csv*** file that can be loaded into a spreadsheet or analyzed externally. The fields in the file are comma delimited and defined as follows.

Field	Contents
Version	This is the software version string. Use this to determine the format of the data in each line.
Time	Timetag expressed as a decimal number. This is time from last reset unless the TIME command has been used to set the system clock. Use this field to calculate relative times.

Dwell	Dwell timer This is the dwell time from the current step in the oven program. It counts down to zero before the oven control program will move on to the next program step.
Program	Program Name This is the name of the running profile. It will be “idle” LEAD or L20C
Condition	Program Condition taken from the condition table that follows
Switch	Key state Indicates if the relay is ON or OFF
oven	Current Oven Temperature
room	Current Room Temperature
target	Target Temperature
fault	MAX31855 fault bits
beep	Buzzer Flag Set to “BZ” when the sonalert is active
comments	Comments from oven program step

10.4.2.1 Condition Mnemonics

The display has an indication of the current condition of the oven controller with values taken from the following table:

CODE	Meaning
AOK	(idle) Oven Controller Ready to run This code appears once when the front panel switches are set to operate the reflow oven.
MAX	(idle) MAX31855 Fault Condition (NOT READY) This code appears when the MAX31855 detects a problem with the thermocouple. If the probe has become disconnected or shorted to something this is what will indicate that there is an issue that needs to be dealt with.
HOT	(idle) Hot Condition (NOT READY) This code indicates the oven or the controller are too hot to operate. The oven must be down to below 50C before a reflow operation can be initiated. In addition if the controller is above 30C the reflow operation will not be started.
CLD	(idle) Cold Condition (NOT READY) This code inidcates the oven or the controller is below 10C. A reflow operation will not be stated at this low temperaturea.

FLT	(idle) Fault Condition during reflow This code is used to flag any issues encountered once a reflow operation has started. Promarily this is an indication that the MAX31855 probe monitor indicates a problem.
RDY	(idle) Oven Ready for operation This code indicates that a reflow operation may be initiated. The system is sensitive to the buttons at this point.
SET	(idle) Over Set to run This code appears once to mark the transition into a relow operation. This is the last code to occur without a program name indicator.
BEG	(program) Beginning of a reflow cycle This code marks the beginning of a reflow operation. It occur only once and may be used to find the start of the status lines that may be used to plot the temperature profile. Temperature control is **not yet active** when this code is displayed.
STP	(program) Program Step This code indicates the handler has read the next record in the reflow program file. Temperature control is active when this code is displayed.

DWL	(program) Programmed Dwell This code indicates the handler is counting down a programmed dwell time. Temperature control is active when this code is displayed.
RUN	(program) Waiting for target temperature This code indicates the handler is heating the oven to bring it to the temperature specified in the program step. The handler waits at this code until the oven reaches the specified temperature. Temperature control is active when this code is displayed.
BRK	(program) Program Break (end of program) This code occurs once at the end of a program. Temperature control is **inactive** when this code is displayed.
FLT	(program) MAX 31855 fault This code indicates a problem has occurred with the thermocouple probe. Temperature control is **inactive** when this code is displayed.
END	(program) Program End This code occurs once at the end of a program. Temperature control is **inactive** when this code is displayed.

10.4.2.2 MAX31855 Fault Bits

The MAX31855 device detects several fault conditions as indicated in the following table:

CODE	Meaning
K OK	Thermocouple fault detector indicates no faults condition
K Opn	Thermocouple fault detector indicates thermocouple open
G Sht	Thermocouple fault detector indicates thermocouple shorted to ground
V Sht	Thermocouple fault detector indicates thermocouple shorted to VCC

10.5 Jumper Configuration.

The RDB board jumpers must be configured to supply switch 5V to the oven. The jumper configuration is slightly different for the two boards due to updates in the board topology to better accommodate the keying circuit. The ¼” phone jack on the RDB015 should not be used to key a radio if a rear panel is installed as the SLEEVE will be connected to ground through the panel. The RDB031 board changes the ¼” phone jack to TIP-RING-SLEEVE where TIP-RING carry the isolated contact closure (SLEEVE is connected to logic ground on the PC board and through the panel).

10.5.1 Oven Control Cable

The oven control circuit is through the ¼“ jack. For both the RDB015 and the RDB031 boards the connection is TIP-SLEEVE with tip being positive and the sleeve being ground.

10.5.2 RDB015 Jumpers

The interface is configured with the ¼“ jack and the radio interface connected to logic ground. Connecting to a radio with the jumpers in this configuration bypasses the isolation circuits in the interface.

10.5.3 RDB031 Jumpers

The RDB031 board always has the SLEEVE of the ¼" jack connected to logic ground. The relay will be configured to supply 5V through the TIP and the radio interface remains isolated.

10.6 Oven Control Profiles

The LEAD profile is laoded into the internal flash file system and the L20C can be obtained from the web page.

10.6.1 Lead Profile “LEAD”

```
OS=LEAD000,1,10,PHONE+HEAT
OS=LEAD001,1,10,BEEP,pre heat
OS=LEAD002,1,10
OS=LEAD003,7,15
OS=LEAD004,7,20
OS=LEAD005,7,25
OS=LEAD006,7,30
OS=LEAD007,7,35
OS=LEAD008,7,40
OS=LEAD009,7,45
OS=LEAD010,7,50
OS=LEAD011,7,55
OS=LEAD012,7,60
OS=LEAD013,7,65
OS=LEAD014,7,70
OS=LEAD015,7,75
OS=LEAD016,7,80
OS=LEAD017,7,85
OS=LEAD018,7,90
OS=LEAD019,7,95
OS=LEAD020,7,100
OS=LEAD021,7,105
OS=LEAD022,7,110
OS=LEAD023,7,115
OS=LEAD024,7,120
```

```
OS=LEAD025,7,125
OS=LEAD026,7,130
OS=LEAD027,7,135
OS=LEAD028,7,140
OS=LEAD029,7,145
OS=LEAD030,7,150,,Ts MIN
OS=LEAD031,7,155
OS=LEAD032,7,160
OS=LEAD033,7,165
OS=LEAD034,7,168
OS=LEAD035,7,170
OS=LEAD036,7,172
OS=LEAD037,7,174
OS=LEAD038,7,176
OS=LEAD039,7,178
OS=LEAD040,7,180
OS=LEAD041,7,182
OS=LEAD042,30,183,,liquidus
OS=LEAD043,7,185
OS=LEAD044,7,190
OS=LEAD045,7,195
OS=LEAD046,7,200,,Ts MAX
OS=LEAD047,7,205
OS=LEAD048,7,210
OS=LEAD049,7,215
OS=LEAD050,7,217,,Reflow
OS=LEAD051,2,218
OS=LEAD052,2,219
OS=LEAD053,2,220
OS=LEAD054,2,221
OS=LEAD055,2,222
OS=LEAD056,2,223
OS=LEAD057,2,224
OS=LEAD058,2,225
OS=LEAD059,2,226
```

OS=LEAD060,2,227
OS=LEAD061,2,228
OS=LEAD062,2,229
OS=LEAD063,20,230,,Peak
OS=LEAD064,7,225,COOL,door 1
OS=LEAD065,7,220,BEEP,door 1
OS=LEAD066,5,215,,door 1
OS=LEAD067,5,210,,door 1
OS=LEAD068,5,205,,door 1
OS=LEAD069,5,200,,door 1
OS=LEAD070,5,195,,door 1
OS=LEAD071,5,190,,door 1
OS=LEAD072,5,185,,door 2
OS=LEAD073,5,180,,door 2
OS=LEAD074,5,175,,door 2
OS=LEAD075,5,170,,door 2
OS=LEAD076,5,165,,door 2
OS=LEAD077,5,160,,door 3
OS=LEAD078,5,155,,door 3
OS=LEAD079,5,150,,door 3
OS=LEAD080,5,145,,door 3
OS=LEAD081,5,140,,door 3
OS=LEAD082,5,135,,door 4
OS=LEAD083,5,130,,door 4
OS=LEAD084,5,125,,door 4
OS=LEAD085,5,120,,door 4
OS=LEAD086,5,115,,door open
OS=LEAD087,5,110,,door open
OS=LEAD088,5,105,,door open
OS=LEAD089,5,100,,door open

10.6.2 Lead Profile “L20C”

OS=L20C000,1,10,PHONE+HEAT

OS=L20C001,1,10,BEEP,pre heat
OS=L20C002,1,10
OS=L20C003,7,15
OS=L20C004,7,20
OS=L20C005,7,25
OS=L20C006,7,30
OS=L20C007,7,35
OS=L20C008,7,40
OS=L20C009,7,45
OS=L20C010,7,50
OS=L20C011,7,55
OS=L20C012,7,60
OS=L20C013,7,65
OS=L20C014,7,70
OS=L20C015,7,75
OS=L20C016,7,80
OS=L20C017,7,85
OS=L20C018,7,90
OS=L20C019,7,95
OS=L20C020,7,100
OS=L20C021,7,105
OS=L20C022,7,110
OS=L20C023,7,115
OS=L20C024,7,120
OS=L20C025,7,125
OS=L20C026,7,130
OS=L20C027,7,135
OS=L20C028,7,140
OS=L20C029,7,145
OS=L20C030,7,150,,Ts MIN
OS=L20C031,7,155
OS=L20C032,7,160
OS=L20C033,7,165
OS=L20C034,7,168
OS=L20C035,7,170

OS=L20C036,7,172
OS=L20C037,7,174
OS=L20C038,7,176
OS=L20C039,7,178
OS=L20C040,7,180
OS=L20C041,7,182
OS=L20C042,30,183,,liquidus
OS=L20C043,7,185
OS=L20C044,7,190
OS=L20C045,7,195
OS=L20C046,7,200,,Ts MAX
OS=L20C047,30,200,,Reflow
OS=L20C048,7,200,COOL,door 1
OS=L20C049,7,200,BEEP,door 1
OS=L20C050,5,195,,door 1
OS=L20C051,5,190,,door 1
OS=L20C052,5,185,,door 2
OS=L20C053,5,180,,door 2
OS=L20C054,5,175,,door 2
OS=L20C055,5,170,,door 2
OS=L20C056,5,165,,door 2
OS=L20C057,5,160,,door 3
OS=L20C058,5,155,,door 3
OS=L20C059,5,150,,door 3
OS=L20C060,5,145,,door 3
OS=L20C061,5,140,,door 3
OS=L20C062,5,135,,door 4
OS=L20C063,5,130,,door 4
OS=L20C064,5,125,,door 4
OS=L20C065,5,120,,door 4
OS=L20C066,5,115,,door open
OS=L20C067,5,110,,door open
OS=L20C068,5,105,,door open
OS=L20C069,5,100,,door open

11 eZ8 Programming

The eZ8 is programmed using the ZDI debug header (J601) located next to the display connector. ZiLog sells several variates of a debuger and the RDB015 board had a debug adapter that could be plugged into the SPI/JTAG connector on the RMB to reflash the eZ8.

A portion of the eZ8 program memory (currently 4KB) is used to sotre the internal flash file system. Default radio setups are held here along with the reflow oven profiles. Changing the internal flash file system currently requires reflashing the eZ8. The eZ8 does have the capability to erase and reprogram the flash but that capability is currently not implemented in the eZ8 software.

The pagesize of program flash in the eZ8 is 512 bytes which requires a more sophisticated handler than the external flash where the pagesize matches the file system record size.

12 Radio Interface

The interface to the radio is through connectors on the back panel. The primary audio interface is through the RJ45 jack. Audio and radio keying are routed through this connector. To accommodate older radios, a relay controlled jack provides higher voltage isolation than the low voltage keying circuit. Also on the back panel are two separate serial ports for programming the radio.

The back panel also has jacks to connect power, audio monitors, and the USB connection to the host and a downstream USB port that can be used to connect additional equipment.

12.1 RJ45

Similar to the connections used on the *Tigertronics Signalink* ® the RH45 carries audio and a keying signal. It is possible to route both left and right channel audio through this connector

12.2 ¼" Phone Jack

The phone jack is provided to accommodate radios that are incompatible with the open-collector circuit that is used with newer radios. The relay closure can be connected through the RJ45 connector is required.

12.3 3.5mm CI-V serial interface

The CI-V interface is presented on a stereo jack and can be configured as a half duplex connection for use with the CI_V interface, or as a full duplex logic level interface. Jumpers allow the ring to be isolated from the circuit to allow use of a common stereo jumper cable with ICOM radios.

12.4 DE9 serial interface

RDB015 boards have a female connector and RDB031 boards have a male connector. Pinout for both connectors is identical and matches the normal signals that appear on an older PC.

12.5 PTT & Relay Configuration

Two of the serial ports may be utilized to control radio keying if the VOX circuit is, in some way, inadequate or imappropriate. An example might be when using the host computer to directly generate CW. In most cases, however, the PTT/SEND control signal to the radio is typically managed by the eZ8 using the VOX feature.

The control circuits are isolated so it is perfectly acceptable to jumper the system to allow control from multiple sources. That is to say, the interface hardware will not be damaged if multiple keying controls are enabled.

Schematic RAD012 describes the routing of serial channel 1 or 4 on the motherboard through JH1 to the daughterboard. JP122 (JH1 pin 11, RTS signal) and JP124 (JH1 pin 7, DTR signal) on the motherboard are used to select the signal to be routed to the daughterboard. Jumpering pins 1 and 2 selects channel 1 and jumpering pins 2 to 3 selects channel 4. The mating connector on the daughterboard, shown in schematic RAD010, also carries the reference designator JH1 with pins 7 and 11 delivered to schematic RAD011 where they pass through an inverting buffer (which may be swapped out for a non-inverting buffer if polarity is an issue). The two buffered signals are then passed along to RAD801 where the keying circuit is located. On schematic RAD810 a set of 4 jumpers provide a means of connecting either of the DTR/RTS signals to either of the keying circuits. JP805 and JP806 provide access to the photo coupled darlington that feeds the RJ45 while JP808 and JP809 provide connections to the relay coil. The jumpers on the daughterboard are easily accessible by removing only the cover. Changing the channel jumpers requires that the boards be unstacked.

12.6 Configuration Header

The configuration header is located immediately behind the RJ45 connector and is similar in function to the header used in the the *Tigertronics Signalink* ®. The configuration header is 20 pins in our case to allow for the addition of the left channel audio signals. The 16 pins at the pin-1 end have identical pinouts to the the *Signalink* ®. Although the programming header for the signalink could be used, the socket provided mates with a 0.025" square post rather than the small machined pins used on the *Signalink* ®.

The programming header provided with the unit has pads for an attenuation network (L-PAD) on both the transmit and receive audio. Changing headers for different radios also changes the attenuation. The header not only programs the pinout for the RJ45, but also programs the levels that the radio and interface see. The intent is to allow the CODEC to operate at or near full volume to reduce quantization noise in the transmit audio and to allow the gain control stage to have a reasonable adjustment range.

Many of the headers provide a path through a resistor from a power source in the radio to the radio interface. This may be used to power the opto-isolator in the RLSD signal is to be used to detect then the squelch is open on the radio. This signal is routed to the eZ8.

12.6.1 Configuration Header JP8PD

The configuration header for 8-pin DIN connector.

PIN	SIGNAL	Description
1	RTTY	
2	GND	Ground
3	SEND	Radio Keying
4	MOD IN	Tx Audio
5	AF OUT	Rx Audio
6	SQU	
7	+13V	
8	ALC	

12.6.2 Configuration Header JP6PM

The configuration header for a 6-pin DIN connector.

This is the more-or-less standard data connector supplied on most new radios.

PIN	SIGNAL	Description
1	PKD	Tx Audio
2	DE	Ground
3	PKS	Radio Keying
4	PR9	Rx Audio
5	PR1	(Rx Audio)
6	SQC	
7		
8		

12.6.3 Configuration Header JP8STA

The configuration header for and SDR type radio. Both left and right channels are routed out to the RJ45 connector with attenuation networks on all 4 audio circuits.

PIN	SIGNAL	Description
1	Rx R	Rx Audio
2	GND	GTround
3	Tx R	Tx Audio
4	KEY	Radio Keying
5	RLSD	
6	Rx L	Rx Audio
7	GND	Ground
8	Tx L	Tx Audio

12.6.4 Configuration Header JPBRJ4

The configuration header for an 8-pin modular (RJ45) microphone jack.

PIN	**SIGNAL**	**Description**
1		
2		
3		
4	GND	Ground
5	MIC	Tx Audio
6	PTT	Radio Keying
7	SPKR	Rx Audio
8		

12.6.5 Configuration Header JP13I

The configuration header for the 13 pin accessory data port on many ICOM radios.

PIN	SIGNAL	Description
1	AF OUT	Rx Audio
2	AF IN	Tx Audio
3	MSEND HF	Radio Keying
4	MSEND VHF	
5	8V	
6	SQLS	
7		
8	GND	Ground

This header will have a 10dB pad installed for use with most ICOM radios.

R41 (connects pins 2 to pin 19) is 2.2K

R46 (bottom of board, connects to pin 19) is 1.0K

12.6.6 Configuration Header JP10K

The configuration header has attenuation networks on the right channel transmit and receive audio lines and pads to allow jumpers to be installed to accommodate any application. This replaces installing jumpers directly into the configuration header socket to allow a convenient change from one radio to another.

PIN	SIGNAL	Description
1		
2		
3		
4		
5		
6		
7		
8		

13 VOX Function

The VOX function is implement by simply polling the A/D at a regular interval to determine when there is a signal present on the right audio channel of the PCM2906. When activity is detected, the hang timer counts are loaded and the corresponding key is active until the counters expire. The counters are decremented every 10 milliseconds and the key is allowed to remain active until the timers expire.

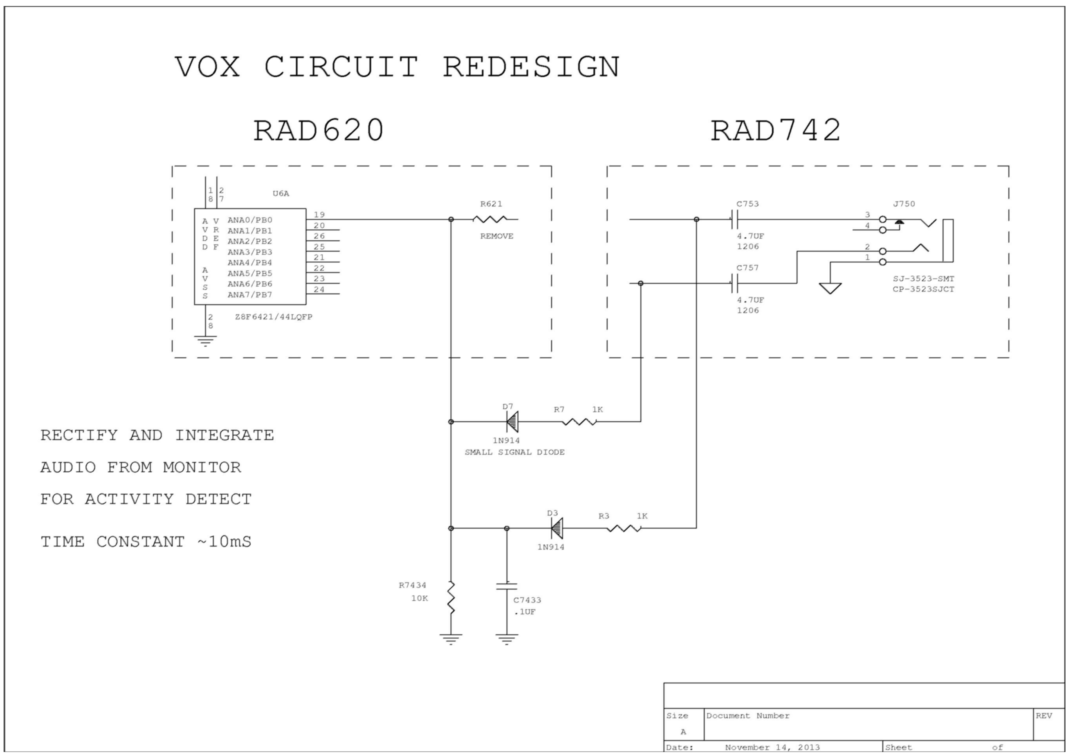
If the audio signal is synchronous with the A/D sampling rate, it is possible for the VOX routine to see a flat line from the PCM2906 and allow the transmitter to be unkeyed. Two steps can be taken to avoid this situation without resorting to hardware changes.

First, simply move the transmit audio up or down a little to avoid multiples of 1KHz (the sampling rate of the A/D), 2KHz and 3KHz.

Second, the VOX timers may be lengthened to allow a longer idle period to occur. While this may be effective it does increase the turn time between transmit and receive. For most digital communications this would not be a problem. It can lead to chopping off the beginning of a reply.

13.1 VOX integrator

Another potential solution is to make use of a simple integrator to provide a DC signal to the A/D in the eZ8 and change the sampling methodology. The audio monitor circuit provides a buffered source for the transmit audio signal to avoid distortion of the signal provided to the radio.



Implementing this solution would require some changed to the software in the eZ8, but there is currently enough room in the ROM to support this. It should be possible to route the.

14 ICOM CI-V Interface

The CI-V interface hardware may be jumpered for full duplex or half duplex operation. The ICOM CI-V bus operates as a half-duplex link between host and multiple radios. Although there is a single connector provisioned on the circuit board, the hardware is capable of controlling more than one ICOM radio.

Also note that there is some flexibility in how J920 is configured. For use with the 3.5mm connector used by ICOM radios, install the JP923 jumper and remove the JP924 jumper. This allows the use of either a mono cable or a stereo style cable. The ring on the mini phone jack/plug is not connected as there is no need for it. This also allows the use of the stereo style connector in the radio if convenient.

The CI-V bus is driven using an open drain / open collector driver. The pullup in the interface is 1K to 5V requiring a maximum of 5mA drive current. The increased drive current allows operation at 19.200 bits/second.

14.1 YAESU Cloning Interface

The levels presented on the CI-V interface are correct for many of the YAESU radios that clone through the microphone jack. Polarities are consistent with those used with the ICOM radios

15 Jumpers

There are many jumpers spread across the two boards to allow configuration flexibility.

15.1 RMB Jumpers

There are 11 jumpers block on the RMB artwork, eight of which must be installed for basic operation.

15.1.1 JP3

NOT Installed.

This jumper controls the downstream USB power switch. In default configuration, the USB hub device monitors and controls the USB power to the downstream port to meet the USB 1.1 power specification. If an overload occurs power is removed from the downstream port. The 7V and 5V regulators are at their thermal limits when supplying the full 500mA to the downstream port.

15.1.1.1 1-2 Installed

Were this jumper to be installed the power limiting function for the downstream USB port is inhibited. The downstream USB port may then draw in excess of the 500mA allowed by the USB 1.1 specification.

15.1.2 JP10

This jumper controls how the USB HUB is reset.

15.1.2.1 1-2 Installed

The USB HUB reset is generated only when the radio interface is first powered on.

15.1.2.2 2-3 Installed (default)

The USB HUB is reset when USB power is applied to the radio interface. This may occur when a hot-plug event occurs of when the host system enables the downstream power to the USB port to which the radio interface is attached.

15.1.3 JP11

NOT Installed.

This jumper grounds the BUSPWR pin on the TUSB2036 telling the host that power is being obtained from the upstream USB port. This is invalid as the radio interface is powered exclusively from external power and never through the upstream USB port.

15.1.4 JP14

NOT Installed.

This is the EXTMEM pin on the TUSB2036 device. IN this design we intend to make use of the default Vendor and Device ID numbers so the jumper pins are not installed.

15.1.5 JP15

NOT Installed.

This is the PWRSW pin on the TUSB2036 device. The nominal function is to report back to the host about the power switching capability of the hub.

15.1.6 JP122

This routed one of two RTS signals from the FT4232 to the daughterboard.

15.1.6.1 1-2 Installed (default)

This selects the RTS signal from channel A of the FT4232. This is the CI-V port.

15.1.6.2 2-3 Installed

This selects the RTS signal from channel D of the FT4232.

15.1.7 JP124

This routed one of two DTR signals from the FT4232 to the daughterboard.

15.1.7.1 1-2 Installed (default)

This selects the DTR signal from channel A of the FT4232. This is the CI-V port.

15.1.7.2 2-3 Installed

This selects the DTR signal from channel D of the FT4232.

15.1.8 JP205

This jumper configures J205-10 of the SPI.JTAG port.

15.1.8.1 1-2 Installed

This setting supplies 3.3V on pin 10 of the J205 connector. This is intended to supply power for an external interface.

15.1.8.2 2-3 Installed (default)

This setting grounds pin 10 of the J205 connector which is more-or-less the way is it set for using it as a JTAG port.

15.1.9 JP430

This is the LCD power selection.

15.1.9.1 1-2 Installed (MOS-AL162A)

Supplies 5V to the LCD. This is the position the jumper must be in when making use of the MOS-AL162A display as it requires 5 volts.

15.1.9.2 2-3 Installed

Supplies 3.3V to the LCD.

15.1.10 JP923

This is the CI-V interface duplex selection.

15.1.10.1 1-2 Installed (default)

This connects transmit and receive data together for half duplex operation. Install when connecting to an ICOM radio.

15.1.11 JP924

This is the ring connection on J920.

15.1.11.1 1-2 Installed

Connects the ring of J920 to the data out of the CI-V interface on the board. This jumper should be left out when using the CI-V interface to control an ICOM radio.

15.2 RMB Resistor Configurations

Several resistor pairs are present that may be installed to change the boards function slightly. They are labeled with a single reference designator with the tow parts having an “A” and ”B” suffix swap

15.2.1 R925

This resistor pair may be used to swap transmit and receive line on the CI-V interface when it is used in a full duplex configuration. Either orientation is acceptable for use with the ICOM radios as the CI-V is half duplex (i.e. running bidirectional data ove a single wire).

15.3 RMB Uninstalled parts

Several parts on the board are unpopulated.

15.3.1 J16/U15

Not installed.

U15 is a serial EEPROM for caltering the USB identifications fields for the USB hub device. J16 provides access to the pins of the serial EEPROM for programming. This is provided for experimentation, the default values provided by the USB hub device are appropriate for Linux and Windows environments.

15.4 RDB Jumpers

These are the jumpers for the daughterboard.

15.4.1 JP150

NOT Installed.

Text Body

15.4.1.1 1-2 Installed

Text Body

15.4.2 JP800

NOT Installed.

This jumper connects the logic ground into the isolated ground of the radio connector.

15.4.2.1 1-2 Installed

Connects grounds together.

15.4.3 JP803

NOT Installed.

This jumper is used to connect logic ground to the relay for controlling a reflow oven.

15.4.3.1 1-2 Installed

Connects grounds together.

15.4.4 JP805

RTS connection from motherboard.

15.4.4.1 1-2 Installed

Connects RTS from motherboard to the optically isolated darlington to key the radio.

15.4.5 JP806

DTR connection from motherboard.

15.4.5.1 1-2 Installed

Connects DTR from motherboard to the optically isolated darlington to key the radio.

15.4.6 JP807

NOT Installed.

Keying control for handheld radio. For this function to be used, R815 needs to be installed. R816 is also provisioned in the artwork to drive the left channel should the need arise. Installing both R815 and R816 will cause crosstalk in the left and right channels, only one should be physically installed if needed.

15.4.6.1 1-2 Installed

Keying function is present on the transmit audio (i.e. a DC path to ground).

15.4.7 JP808

DTR connection from motherboard.

15.4.7.1 1-2 Installed

Connects DTR from motherboard to the relay to key the radio.

15.4.8 JP809

RTS connection from motherboard.

15.4.8.1 1-2 Installed

Connects RTS from motherboard to the relay to key the radio.

15.4.9 JP810

There is a diode installed on the motherboard across the relay contacts to allow switching an inductive load. J820 tip should be connected to the more negative switching point on the radio.

15.4.9.1 1-2 Installed (default)

Diode is connected.

15.4.10 JP818

Keying pin of the J800 radio connector (RJ45). This is used to select either the optically isolated darlington or the relay for keying the radio.

15.4.10.1 1-2 Installed

Relay is used for radio keying.

15.4.10.2 2-3 Installed (default)

Optically isolated darlington is used for radio keying.

15.4.11 JP819

This provides a connection between the relay and the radio ground.

15.4.11.1 1-2 Installed

Radio and Relay are bonded together.

15.4.12 JP820

Relay power for reflow oven.

15.4.12.1 1-2 Installed (default)

The realy is operating as a radio transmit control.

15.4.12.2 2-3 Installed

5V is supplied to the solid state switch in the reflow oven when the relay is closed.

15.5 RDB Resistor Configurations

Several resistor pairs are present that may be installed to change the boards function slightly. They are labeled with a single reference designator with the tow parts having an “A” and ”B” suffix swap

15.5.1 R625

This resistor pair may be used to swap transmit and receiver between the eZ8 and the LCD. The default locations are , however, correct.

15.5.2 R626

This resistor pair may be used to swap transmit and receiver between the eZ8 and the host. The default locations are , however, correct.

15.5.3 R721

This reference designator refers to two resistors that may be used to provide a load inboard of the receive audio transformer if needed. There is one part for each channel (R721L and R721R)

15.5.4 R731

This reference designator refers to two resistors that may be used to provide a load outboard of the transmit audio transformer if needed. There is one part for each channel (R731L and R731R). Together with R733R/R733L an attenuator is formed.

15.5.5 R802/R803

This resistors may be changed out to sum the right and left channels. Sent to the radio.

15.5.6 R806/R809/R810/R811

These resistors are provided to allow the RLSD optical isolator to be routed to the eZ8. By default, the board is configured to take power from the radio and expects RLSD signal to be driven to ground. The optical isolator has a 330 ohm current limit resistor in series with the emitter. If this would allow excessive current to flow, the programming header that matches the radio to the interface must be provisioned with the correct resistor. Consult the software release notes to see if this function is supported.

15.6 RDB Uninstalled parts

Several parts on the board are unpopulated.

15.6.1 J600

Not installed.

The J600 pins provide access to the serial EEPROM that holds the external file system. The eZ8 has software routines that provide full access to the device to the host, so there is no real need for this connector to be installed.

15.6.2 J612

Not installed.

The J612 pins provide another means of adding an LCD display. The pinout of this LCD connector is almost identical to the one that is populated on the motherboard. The 5V connection has been moved so the interconnect to the LCD must be made to match either RDB or RMB.

If the downstream USB port is being used to provide power and the regulators on te RBM get hot, the LCD can be moved to the RDB by installing the connector and changing the location of the power pin. Plugging the LCD connector into the wrong LCD connecter should not damage the display, but it would not function plugged into the wrong port as it would not be powered.

15.6.3 U19

This part **IS** installed.

This is a dual inverter that is used to control the polarity of the keying lines that come from RDB. The output of these gates drive Q808 and Q809 which are then routed through JP805 – JP809 to allow the host to key the radio through two of the channels on the UARTS on RMB.

15.6.4 U850/J850

This is the thermocouple amplifier/A/D chip and associated connector. It is not normally installed on the board although the oven control software is typically present in the eZ8 (this assumes that the software image hasn't grown out of control and no longer fits into the flash available in the eZ8.

16 eZ8 Software

This section describes the eZ8 software organization.

Keep in mind that the eZ8 is a microcontroller, not a full up microprocessor so resources are limited to a degree. The eZ8 used in the interface provides a large complement of interface blocks, so the interface to the radio logic is rather simple, being connected to device pins with a minimum of interface laogic. In addition the program flash in the device selected is large (for a microcontroller). The primary limiting resource in the eZ8 is the size of the RAM (only 3808 bytes).

16.1 Radio Interface

The first section wcovers the code in the eZ8 used in the radio interface.

16.1.1 Logic flow

The eZ8 control program is built as a foreground/background system where the foreground activities are in the form of interrupt service routines and the single background activity is a polling loop that manages the user interface.

16.1.1.1 Timer 0

The interrupt routine associated with this hardware timer manages simple timers. Keying control is handled in this interrupt routine. Reading the A/D for the VOX function is also managed in this interrup t routine.

16.1.1.2 Timer 1

The CW function makes use of this hardware timer with the code timing implemented in the interrupt routine. The period of this interrupt is set by the code chipping rate. That is to say the interrupt occurs at the *dit* rate. Changing the code speed changes the count register that controls the code word rate.

16.1.1.3 Timer 2

Timer 2 is used to generate a square wave when using audio keying, such as when controlling an FM radio. Note, however, that using a square wave will generate odd harminics.

16.1.1.4 UART0 input

Text received from the host system is buffered using the hardware interrupt from UART channel 0. Traffic is accumulated at the interrupt level until the buffer is filled or a <CR> character is received.

Keep in mind that RAM resources are extremely limited on the eZ8 so the host is required to manage buffer sizes sent to the eZ8 to avoid data loss.

16.1.2 Module functions

16.1.2.1 analog_local.c

This is the low-level driver for the A/D hardware.

16.1.2.2 code.c

Morse code generator.

16.1.2.3 command.c

Host command processor.

16.1.2.4 flash_file.c

Default file system in program flash.

16.1.2.5 flash_local.c

Flash file system.

16.1.2.6 gpio_local.c

GPIO and SPI handler.

16.1.2.7 oven.c

Oven Controller.

16.1.2.8 owner.c

Template for the fixed information block that is located in the top 512 bytes of program memory.

16.1.2.9 setup_local.c

This code uses the setup information in the flash file system to configure the interface for oepration.

16.1.2.10 test1.c

Main Control Loop.

16.1.2.11 timer_local.c

Timer handlers.

16.1.2.12 type_local.c

Handler to extract the eZ8 part number.

16.1.2.13 uart_local.c

UART handler.

16.2 SWR/Power Meter

The second section covers the code in the eZ8 used in the SWR/Power Meter. The code modules from the radio interface were reused as much as possible, but there has been no attempt to use common modules as the regression testing becomes extensive.

16.2.1 Logic flow

The eZ8 control program is built as a foreground/background system where the foreground activities are in the form of interrupt service routines and the single background activity is a polling loop that regularly formats and forwards data..

16.2.1.1 Timer 0

The interrupt routine associated with this hardware timer manages simple timers. A simple system time is managed here to allow the outer loops to mark time with a reasonable degree of accuracy.

16.2.1.2 Timer 1

This time channel is dedicated to generating a PWM signal to drive the LCD backlight. Although it may be altered by command, it is essentially a set & forget portion of the hardware.

16.2.1.3 Timer 2

This time channel is dedicated to generating a PWM signal to drive the LCD contrast pin. Although it may be altered by command, it is also a set & forget portion of the hardware.

16.2.1.4 UART0

This is the CI_V channel for the SWR/Power meter. Although the hardware may be configured to operate full duplex, the CI-V bus is half duplex and that is how the hardware is configured.

Both transmit and receive channels are interrupt driven. Using interrupts to manage the transmitter allow for a small degree of processing overlap. The very limited RAM of the eZ8 effectively limits the overlap thaat can be achieved.

16.2.1.5 UART1

This is the RS485 channel for the SWR/Power meter. The hardware only allows for half duplex operation. The bus driver chip has provisions to disable the receive channel when transmitting which is used to reduce the channel overhead.

As with UART0, both transmit and receive channels are interrupt driven. Using interrupts to manage the transmitter allow for a small degree of processing overlap. The very limited RAM of the eZ8 effectively limits the overlap that can be achieved. When sending single messages (i.e. when the meter is configured to deliver all data in a single message), the use of interrupts is most effective.

16.2.2 Module Functions

Descriptions of the various code modules used in the SWR/Power Meter.

16.2.2.1 analog_local.c

This is the low-level driver for the A/D hardware. It configures the A/D controller in the eZ8 to make use of the DMA assist so as to collect 8 A/D measurements without requiring the eZ8 to expend CPU cycles (scanning 8 channels slows the A/D cycle rate down slightly).

In addition to the active A/D DMA buffer, the interrupt routine saves data into a set of 16 buffers so that the A/D data can be averaged before being analyzed.

16.2.2.2 civ_gen.c

ICOM CI-V message generator. This module is responsible for formatting and sending CI-V traffic.

16.2.2.3 command.c

Host command processor. Similar to the module of the same name in the radio interface. Similar to the module of the same name in the radio interface.In addition to the active A/D DMA buffer, the interrupt routine saves data into a set of 16 buffers so that the A/D data can be averaged before being analyzed.

16.2.2.4 eu_cvt.c

Engineering Units conversions. This module has routines to perform conversions from raw counts to several engineering units. The conversion tables are laoded using the “POLY” command listed elsewhere.

16.2.2.5 flash_file.c

Default file system in program flash. This is where the “CF=” records are stored.

16.2.2.6 flash_local.c

Flash file system. Slightly stripped down flash file system taken from the radio interface. The record size of the flash file system is increased to 64 bytes in order to accommodate the coefficients tables that are loaded into the external file system as part of the calibration process.

16.2.2.7 gpio_local.c

GPIO and SPI handler. Basically taken from the radio interface with modifications to bit assignments taken into account.

16.2.2.8 test2.c

Main Control Loop.

16.2.2.9 timer_local.c

Timer handlers.

16.2.2.10 type_local.c

Handler to extract the eZ8 part number.

16.2.2.11 uart_local.c

UART handler. Updated to add interrupt to the transmit path. As the SWR/Power Meter will be constantly sending data messages when transmitting, this update improves overlap slightly.

16.2.3 CI_V Message Traffic

The message format is based on the ICOM CI-V messages seen in recent ICOM amateur radio systems. Addressing and message identifications are configurable to allow the system to avoid conflict with the radio. To coexist with applications such as Flrig that poll the radio for message traffic, the CI-V traffic generated in the SWR/Power interface attempts to synchronize with other traffic on the CI-V bus.

Messages over the CI-V bus would normally be binary, as documented in radio manuals and elsewhere on the web. The message traffic on the CI-V bus may be reconfigured to dump in ASCII to allow debugging.

Messages over the RS485 bus are sent using the CIVM and CIVR commands. The messages are formatted as HEX-ASCII so as to be compatible with the interactive command processor. The CIVM command passes a CI-V command from the remote system to the local CI-V bus. The CIVR command causes CIVM commands to be generated. The software can be configured to send CIVM messages on schedule or the message traffic may be polled.

16.2.3.1 CIV-V message format

The CI-V messages processed by the system are somewhat configurable. The host and local addresses may be configured through the command interface (and, therefore using the SC= setup interface). Also the message command pattern may be set in the same manner.

A message on the CI-V bus will be binary of clear text as set through the command interface (i.e. set using the "SC=" in the flash file system). Either way the data fields are as follows:

FD FD HA LA MC dd ... dd FE

Where there is a 2 byte synchronization/start pattern, a host address byte, a local address byte, a message command byte, some number of data bytes, and finally a end-of-message byte. When the message is sent as text, there will be a field prior to the message to make it easier for us to read. In the case of message traffic over the RS485 bus, the HEX-ASCII buffer starts with the CIVM command so the other end of the connection can decode the message appropriately.

16.2.3.2 CIV-V message forwarding

CI_V message traffic that comes across the RS485 bus is automatically forwarded on the CI_V bus as part of the command decoding processing. The system may be als configured to forward CI-V message traffic from CI-V bus to the RS485 bus.

When a CIVM message arrives in the system, it is decoded as any command would be and acted upon. To disable forwarding, simply disable generation of the automatic CIVM traffic and depend on one system polling the other.

Traffic on the CI-V bus that is aimed at the local system can be processed in much the same manner as commands are processed, using a command processor for CI-V traffic. Traffic aimed at the remote system can simply be forwarded based on its address by allocating a group of addresses based on the address of the unit connected to the CI-V bus.

17 eZ8 flash file system

The flash file system in the eZ8 holds configuration setting for the VOX control, the CW generator, and the oven control routine. The file system is broken into two regions, one in the eZ8 program flash and one in an external byte programmable flash device. Only the external flash file system may be altered, the internal flash file system is fixed when the application is burned into the eZ8 by the programmer.

The internal flash file system is further subdivided into an area that is loaded when the device is reprogrammed and an area that is loaded once when the device is initially programmed. The fixed area holds the device serial number, the original users callsign and the time when the device was put into service. Updates to the operating software does nto contain records to update this fixed area so the identity of the device is not lost.

17.1 Internal Flash

The internal flash is located in program flash in the eZ8. Physically it resides in the top 4K bytes of program flash. Since the flash file system does not deal with writes to internal flash, this filesystem is built into the software load and is altered by reprogramming the entire program flash area. Since the eZ8 does not depend on a flash loader and the programmer is relatively simple to implement, this should not pose a significant problem.

The program flash on the eZ8 is erased in 512 byte pages. To take advantage of this, the top 512 bytes of the internal flash file system contain identification information that is not changed or overwritten when the device is reprogrammed. The device serial number and original owner are stored here.

17.2 External Flash

The external flash file system resides in the MN95320 device located at U620 on the daughterboard. The MN95320 is a byte writeable device so there is no need to deal with page erase management. The flash file system record size of 32 bytes does match the page write size of the device, again making page management quite simple.

18 Radio Setup Notes

This section contains setup notes gathered for various radios.

In particular, audio levels are set on the programming headers to allow the gain stage to be set close to unity gain if at all possible and to allow the CODEC to be operated using as large a signal swing as possible.

Operating the CODEC near full swing takes advantage of as much of the dynamic range of the device as possible. Reducing the volume using the volume control slider in the host computer limits the dynamic range that the CODEC can present to the radio.

The following pages contain the notes gathered in setting up with the various radios.

18.1 Radio: ICOM IC-7200

This radio has a built in USB connection providing most of the connectivity of the interface, but lacks an effective VOX control (although the radios does implement VOX) and a convenient means of changing configurations (you have to dig through menus a bit).

This radio has a relatively wide AF bandwidth. Response is from about 300Hz to 3300HZ. There is a slight peak at the bottom of the AF band (a few dB). The bandpass and notch filters are effective.

18.1.1 Jumper JP13I

The following parts are installed for use with the IC-7200.

Receive audio has a 6dB pad

Transmit audio has approximately a 12dB pad to allow close to unity gain without overdriving the amplifier.

Ref Des	Value	Function
R36	1K	Receive Audio Attenuator
R47	1K	
R41	4.7K	Transmit Audio Attenuator
R46	1K	
R44	0	SEND / PTT

18.1.2 Radio Settings

LINE	MENU	Item	Value
1	SET(long)	DATA	oF
2	SET(long) SET(long)	Mod	A
3	SET(long) SET(long)	CI-V BAUd	192
4	SET(long) SET(long)	CIV ADR	76
5	SET(long) SET(long)	CIV TRN	on
6	SET(long) SET(long)	RF/SQL	rS
7	FILTER(long)	WIDE	3600
8	AGC(long)		AGC-OFF
9	NB		OFF
10	NR		OFF
11	ANF		OFF
12	MNF		OFF until needed
13	Twin BPF		CENTERED until needed

We are not using the internal CODEC and UART, so we connect the accessory port on the back. With a microphone that isn't muted, you may want to leave it unconnected (Line 2 is **not** set to *M A*).

Set the CI-V up to 19,200 bits/second, which is as fast at the 7200 will operate if there is not a speed conflict with other devices on the bus. The interface provides a stiff pull-up to allow the bus to operate at this higher speed.

18.1.3 Interface Records

The following records were added to the external file system to provide quick setup of the interface for working 20M. Note the change in the hang timer for the JT65/JT9 modes.

```
Save,LB=JT65,JT65 and JT9
Save,RT=JT65,22,150,150,150
Save,RS=JT65,30,30,50,50,10

Save,LB=PSK31, FLdigi
Save,RT=PSK31,22,50,50,50
Save,RS=PSK31,45,45,45,45,10
```

18.2 Radio: ICOM IC756

The IC756 uses an 8 pin DIN connector for the data connection.

18.2.1 Jumper JP8PD

The following parts are installed for use with the IC-756.

Receive audio has a 6dB pad

Transmit audio has approximately a 12dB pad

Ref Des	Value	Function
R14	1K	Receive Audio Attenuator
R29	1K	
R13	4.7K	Transmit Audio Attenuator
R28	1K	
R12	0	SEND / PTT
R15	0	Receive Audio Echo

R15 may be installed to address issues with software demanding input appear on the left channel. T721L would need to be installed as well to pass the audio signal on to the CODEC.

18.3 Radio: YAESU FT1500

The FT-1500 is a simple VHF radio that makes use of the more-or-less standard 6-pin mini-DIN connector for accessing the audio circuitry prior to the volume control.

18.3.1 Jumper JP6PM

Text Body

Ref Des	Value	Function
R4	100	Receive Audio Attenuator
R27	1K	
R1	4.7K	Transmit Audio Attenuator
R26	1K	
R2	0	SEND / PTT

18.4 Radio: YAESU FT-897

The FT-897 suffers from a rather narrow SSB filter.

18.4.1 Jumper JP6PM

The jumper is configured to deliver receive audio unattenuated to the interface and the transmit audio is attenuated by 20dB.

Ref Des	**Value**	**Function**
R4	10	Receive Audio Attenuator
R27	1K	
R1	10K	Transmit Audio Attenuator
R26	1K	
R2	0	SEND / PTT

18.4.2 Radio Settings

Menu item 037, DIG GAIN needs to be set to a lower value to eliminate overdriving the radio. Try an initial value of 10.

18.5 Radio: YAESU FT-1000MP

This radio has some minor issues that need to be addressed in order to operate some digital modes. In particular, when set for packet operation the radio wants to operate LSB rather than USB. This shortcoming needs to be address by configuring a user data mode with appropriate shifts to force the radio into USB operation.

Traffic on the web seems to indicate that early models, where you ca not set the transmit carrier below 456.300, cannot actually use USB. It looks like all you really manage is to artificially shift the lower sideband up. Although this would work for PSK31, other modes like JT65 will not decode as the tone positions are inverted.

18.5.1 Jumper JP10K

These are the PAD values for the audio attenuators.

Ref Des	Value	Function
R56	1K	Receive Audio Attenuator
R66	1K	
R57	4.7K?	Transmit Audio Attenuator
R67	1K	
	0	SEND / PTT

18.5.2 Radio Settings

Settings specific to this radio, in particular the USER mode setups for running digital modes on the upper sideband.

18.5.2.1 USER Setup

- Enter the "Menu" mode by pushing and holding in the [FAST] key and then pushing the [ENT] key on the keypad momentarily.
- Rotate the [MEM/VFO CH] knob to select Menu Item 8-6.
- Now set up the parameters in each category shown in one of the tables below. These parameters will shift the various mixers, while in the "PKT/LSB" mode, so as to place them at the frequencies they would be in if there were a dedicated "PKT/USB" mode. In effect, you are creating such a mode by using the "USER" key setup capability.
- Push the [ENT] key momentarily to save your new settings and exit to normal operation. The "User" mode is now configured for pseudo-USB operation in the AFSK (Packet) operating mode.

Table 1 Valid ranges (from YAESU manual)

Menu Item 8-6	**Range**	**Default**
USER SETTING		
Mode	LSB USB CW-L CW-U RTTY-L RTTY-U **PKT**	LSB
Display Offset	+/- 5.000 KHz	0.000
Receive PLL	+/- 5.000 KHz	-1.450
Receive Carrier	450.000KHz - 460.000KHz	456.450
Transmit PLL	+/- 5.000 KHz	-1.500
Transmit Carrier	LSB: 456.300 - 460.000 kHz USB: 450.000 - 453.700 kHz **PKT: 450.000 - 453.700 kHz** **or 456.300 - 460.000 kHz** all others: 450.000 - 460.000	456.500
RTTY Offset	+/- 5.000 KHz	0.00
Easy Set	**OFF**/SSTV/FAX/PSK-31	OFF

Note the 455KHz center on both the transmit carrier and receive carrier.

So why don't we simply shift the carrier values to the opposite side of 455KHZ and invert the sign on the PLL setting. In this analysis we are ignoring the early production models as they appear not to deal with USB correctly.

Table 2 Initial Suggested Settings

Menu Item 8-6	**Range**	**Default**	**New Value**
USER SETTING			
Mode	LSB USB CW-L CW-U RTTY-L RTTY-U **PKT**	LSB	**PKT**
Display Offset	+/- 5.000 KHz	0.000	
Receive PLL	+/- 5.000 KHz	-1.450	**+1.450**
Receive Carrier	450.000KHz - 460.000KHz	456.450	**453.550**
Transmit PLL	+/- 5.000 KHz	-1.500	**+1.500**
Transmit Carrier	LSB: 456.300 - 460.000 kHz USB: 450.000 - 453.700 kHz **PKT: 450.000 - 453.700 kHz** or 456.300 - 460.000 kHz all others: 450.000 - 460.000	456.500	**453.500**
RTTY Offset	+/- 5.000 KHz	0.00	**0.00**
Easy Set	**OFF**/SSTV/FAX/PSK-31	OFF	**OFF**

Will the display offset need to be changed to account for the 3KHz shift in the transmit carrier or the 2.9KHz shift in the receive carrier?

Table 3 (Marty Wayne) W6NEV

Menu Item 8-6	Setting	
	Late	Early
USER SETTING		
Mode	PKT	
Display Offset	+2.125kHz	
Receive PLL	+2.210kHz	
Receive Carrier	452.790kHz	
Transmit PLL	+2.210kHz	
Transmit Carrier	452.790kHz	456.300
RTTY Offset		
Easy Set		

Table 4 (Tigertronics, W5ERX)

Menu Item 8-6	**Setting**	
	Late	**Early**
USER SETTING		
Mode	PKT	PKT
Display Offset	Adjust if needed	+2,125
Receive PLL	1.450	+2,210
Receive Carrier	453.550	452.790
Transmit PLL	1.500	+2,210
Transmit Carrier	453.300	452.790
RTTY Offset	0.000	
Easy Set	OFF	

18.6 Radio:

Text Body

18.6.1 Jumper JP13I

Text Body

18.6.2 Radio Settings

Text Body

19 Software Revision and Release Notes

This section lists changes to the software and notes about upgrading the software.

19.1 Upgrading the eZ8 software

As mentioned earlier in the manual, the internal file system is stored in program flash and will be updated with software changes. Since this part of the overall file system can not be written by the user, there will be no user setups stored in this part of the file system for upgrades to overwrite.

The upgrade image contains, at most, 65,020 bytes of memory image data covering from 0x0000 through 0xFDFF. This leaves the top 512 bytes from 0xFE00 to 0xFFFF unaltered. The program flash is arranged into 512 byte blocks so the top 512 bytes of program memory store the following records:

CF=HWDATE,*<build date>*

CF=<ownr>,*<owner>*

CF=<call>,*<callsign>*

CF=<name>,*<name>*

CF=<zitu>,*<ITU zone>*

CF=<z_cw>,*<CW zone>*

CF=<locn>,*<maidenhead locator>*

CF=<bser>,*<serial number>*

CF=END,Configuration area end

The italicized parameters will have values assigned when the unit is put into service. The build date (HWDATE) and board serial (<bser>) are always read from the internal file system when generating status reports to the host as they should not be altered. The other fields can be superseded by adding records to the external flash file system. This allows the fields used in the CW strings to be altered if needed.

Since the upgrade image does not contain image data for the top 512 bytes of memory, the owner and board serial are preserved through a software upgrade. In addition, the external flash files system is left unaltered by a software upgrade. Configuration and Messages remain intact. If the handling of the setup strings is altered, it may be necessary to update the external file system to reflect the changes.

ZiLOG makes several flash programming devices for the eZ80 and eZ8 family. The least expensive being the USB flash programmer. The ZiLOG part number is ZUSBSC00100ZACG which DigiKey stocks under their number 269-4539-ND for about $20.00. Perhaps the programming adapter for the JTAG port will be useable for this as well.

19.2 Revision Notes

New additions...

19.2.1 RADV1.??

New additions...

19.2.2 RADV1.56

Fix some issues with the READ command. Format for the read and dump buffers chage slightly to reduce the code footprint that generates the dumps.

19.2.3 RADV1.51

Added CI-V traffic monitor haywire to the S/N 2078-0-0026 unit to continue with the CI-V traffic monitor. Added timing information to the dump and changed the host bit rate to 57,600 bits/second. The increased bit rate to the host allows CI-V traffic dumps to occur correctly.

19.2.4 RADV1.50

It appears that we can add a CI-V traffic monitor to the hardware by removing the RXD line from the LCD and grabbing the data directly from the opto-isolator circuit on the motherboard at R917 and connecting it to the eZ8 UART1 RX data line at J401 pin 7. The LCD connections are passed between boards through J401-J610.

The V1.50 software update enables the receive interrupt for the UART1 RxD and adds two commands for monitoring traffic. Note that this requires that the LCD and the CI-V bus use the same bit rate. As of this version it is set to 9600 bits/seconds, but we may add something to the configuration records to allow this to be changed.

19.2.5 RADV1.49

Some small forgotten improvement.

19.2.6 RADV1.48

Removed the *INIT* command and replaced it with a quicker system. Now you can simply type in the name of the setup and it will be loaded. Looks rather like a command extension.

19.2.7 RADV1.47

Fixed the setup selection code. Now you can select setups from the front panel and they display the setup on the LCD. This setup select switch position is now sensitive to audio coming out of the CODEC (so you can leave it set there should you deisre to switch back and forth between modes quickly).

19.2.8 RADV1.45/1.46

Added <zitu>, <z_cw>, and <locn> to the CF= records to allow code substitution.

19.2.9 RADV1.44

Added the INIT command to allow the radio setup to be selected through the host in much the same manner as it is selected from the front panel.

Also removed all of the test setups in the internal flash file system and added the lead oven profile back into the internal file system. Since we can't rewrite the internal file system, the over reflow profile might just as well be there in case it's needed. Right button activates lead profile, matching the right button-to-activate of other functions. Ther Left button does the L20C profile which would need to be loaded into external flash.

Radio setups and CW messages can easily be added in the external flash file system.

19.2.10 RADV1.43

Any command from the host stops the status display. Switching to the oven control position on the front panel switches also stops the status display so as not to obscure the over status reports. These are intended to make the system a little more flexible/useable when the LCD is not present

19.2.11 RADV1.42

Updated KEYN command to add BEEP control.

19.2.12 RADV1.41

Added DSPY command.

19.2.13 RADV1.40

The 5th. Argument to the RS= setup record was added. This change uses this 5th. Parameter to enable the USB volume control. When non-zero USB volume is driven up when the eZ8 is reset with the length of time set by the value of the parameter.

19.2.14 RADV1.38

The DS= setup key was added to the setup decode. This configures the LCD operation and allows host status reports to be enabled at reset.

19.2.15 RADV1.37

The <name> argument substitution capability added.

19.2.16 RADV1.36

The first stable release version.

20 Impedance Bridge/Power Monitor accessory

This is the documentation for the Impedance Bridge / Power Monitor that may be attached to the radio interface through the CI-V port.

20.1 Circuit Descriptions

Circuit Descriptions

20.1.1 Bridge Circuit

The impedance bridge may be built up using either a 4 port bridge (sometimes referred to as a *tandem match* bridge) or a Bruene Bridge. The radio and antenna ports are terminated in the SO239 connectors on the front panel of the chassis. The forward and reflected points are terminated in 50 ohms internally and have an attenuation that is set by the turns ration of the transformer(s), typically about 40dB.

The tandem match consists of two transformers, one connected as a current monitor and the other as a voltage monitor.

The Bruene bridge consists of a current transformer with two secondary windings and two voltage taps. The forward power and reflected power are taken independent of each other, hence the need for two I/V measurements.

20.1.2 Signal Conditioning

The resulting voltages from the bridge circuit is fed into a pair of RF log detectors. Using the log detector effectively eliminates the need for scaling circuits to manage power levels from one watt to on kilowatt.

20.1.3 Trim

Trim components for the log detector are provided by programmable resistors, in much the same manner as used to control the audio gain in the radio interface board.

20.1.4 Processor

The processor is the same eZ8 used in the radio interface. The processor device has 8 analog input channels, six of which are used to measure the voltages produced by the log detectors.

20.1.5 CI-V interface

The CI-V interface matches that used on the radio interface. The CI-V interface is optically isolated. Unlike the radio interface, the eZ8 is able to read and write to this interface.

As on the radio interface, the CI-V connection uses a 3.5 mm stereo jack and may be configured for half duplex or full duplex operation.

20.1.6 RS-485 interface

The RS485 interface is not isolated (although the coupler is transformer based so there is isolation between the RF ground and the interface ground).

20.1.7 LCD interface

Unlike the radio interface, the LCD interface would normally not be populated as this unit is intended to be located away from the station. The LCD interface provided on the board is able to drive a parallel LCD in 4-bit mode or using the SPI interface. LCD matching the mechanical size of that used in the radio interface are available.

20.2 Topic 5 .

Text Body

21 More Topic .

Text Body

21.1 Topic 2 .

Text Body

21.1.1 Topic 3 .

Text Body

21.1.1.1 Topic 4 .

Text Body

21.1.1.1.1 Topic 5 .

Text Body