JUMA TRX-1 DSB/CW QRP TRANSCEIVER

The Juma TRX-1 is a DSB/CW low-power transceiver that can cover the 160 – 80 – 40 metre amateur bands. This firmware, version 7.1a, has a number of enhanced features:

- Significantly improved frequency synthesiser accuracy
- Availability of both RIT and VFO-A/B selection
- Rapid Band-Switch
- Out-Of-Band alarm
- User settable Out-Of-Band Alarm enable/disable
- Input Voltage low/high alarm
- Selectable Auto or Manual reset of voltage alarm
- Software adjustable voltmeter calibration
- VFO-A/B frequency copy
- Optional LED alarm indicator
- Fast RIT offset cancellation
- Fully prompted System Setup Menu selections
- Voltage averaging for a more stable voltmeter display

Transceiver Controls

VFO Knob/Button

This is used primarily to adjust the transceiver's operating frequency. However, the VFO button is also used to select various operating modes, as well as to step through the menus of the System Calibration and operating mode settings.

Gain – On/Off

This is used to turn the transceiver on or off, and to alter the audio volume.

Keyer Speed

This knob controls the CW speed setting of the optional internal keyer.

CW/DSB

This switch selects the mode, CW or DSB.

RIT
Receiver Incremental Tune On/Off. The display will change to show the RIT offset. A short push of the VFO button cancels the current offset, otherwise it is saved until the RIT is next enabled. Use the VFO knob to alter the offset, the limits are ±9,990Hz. The step rate is fixed at 10Hz.

Note: When RIT is enabled, the only other mode that can be selected is Mode 2 to display the battery voltage and the current step rate. The VFO A/B and Rapid Band-Switch functions are inhibited.

Transceiver Operating Modes
The transceiver has 4 modes of operation selected by means of the VFO push button.

Mode 0 – Normal Operating Mode
In this mode, the left-hand side of the display will be show the graphic S-meter in receive, or the graphic forward/reverse power in transmit.

The right-hand side of the display will show the current transceiver's operating frequency, or, if the RIT switch is on, the RIT offset frequency.

The transceiver will respond to the VFO knob to change frequency, the PTT/key input to switch between transmit and receive, the DSB/CW switch, and the RIT switch, as well as the VFO button.

Mode 1 – VFO A/B Selection
This is selected by means of a short push of the VFO button. If the contact is closed for longer than 50mS but less then 500mS then this mode is selected, and a beep will be given. The left-hand side of the display will show the currently selected VFO, and the right-hand side its frequency.

Rotate the VFO knob clockwise to select VFO-B or anticlockwise to select VFO-A. Note the difference in the decimal points. The normal decimal points at the bottom of the display indicates VFO-A is selected; the larger decimal points nearer the centre of the character indicate VFO-B is selected.

A brief press of the VFO button will select the chosen VFO, and return you to Mode 0.

A medium press of the VFO button, longer than 500mS but shorter than 1.5S, will allow you to copy the frequency from the <u>inactive</u> VFO to the <u>active</u>. The display will update to reflect this change.

A long press of the VFO button, longer than 1.5S, will save the current transceiver's state to the EEPROM, including the frequencies of VFO-A and VFO-B, the currently active VFO, and the current VFO step rate.

Mode 2 – VFO Step Rate Select
This mode is selected with a medium push of the VFO button. A medium push is greater than 500mS and less than 1.5S. The left-hand side of the display will show the current DC input voltage, the right-hand side of the display will show the current step rate, 10Hz, 100Hz, 1kHz, or 5kHz. Rotate the VFO knob clockwise to increase the step rate, anticlockwise to decrease.

The 100kHz step rate is no longer available. The 5kHz rate was chosen as this corresponds to the standard channel spacing for SW broadcast stations.

A brief press of the VFO button will return you to Mode 0 with the chosen step rate active.

Note: If RIT is on, and this mode is selected, then it will only display the battery voltage and the current step rate.

Mode 3 – Rapid Band-Switch
This mode is selected with a long push of the VFO button. A long push is greater than 1.5S. The left-hand side of the display will show the nearest amateur band to the currently active VFO's frequency, as 160m, 80m, or 40m. Rotate the VFO knob select the desired band, the preset frequency will be displayed.

A brief press of the VFO button will return you to Mode 0 with the currently selected amateur band's frequency in the active VFO. As with the other modes, a long push will save the current transceiver's state to the EEPROM.

The preset frequencies for the amateur bands can be edited in the System Setup menu.

Note: Although the 160m band can be selected, note that the output filters are only optimised for the 80m and 40m bands. When operating on the 160m band there will be additional harmonic energy, and an antenna tuning unit or external low-pass filter is recommended.

System Calibration
With the transceiver off, press and hold the VFO button, and turn the transceiver on. The first display will show:

```
Juma TRX-1 v7.1a
```

After a short delay:

```
OH2NLT-OH7SV
```

Finally, after another short delay the message:

```
System Setup
```

Release the button, and the first menu item will be displayed. To select the other items, briefly press the VFO button.

Region Selection
This can be set to either IARU region 1, or the USA region. Rotate the VFO knob clockwise to select USA, or anti-clockwise to select IARU. The band edge frequencies are:

BAND	IARU Region 1	USA
160m Lower	1,810kHz	1,800kHz
160m Upper	2,000kHz	2,000kHz
80m Lower	3,500kHz	3,500kHz
80m Upper	3,800kHz	4,000kHz
40m Lower	7,000kHz	7,000kHz
40m Upper	7,200kHz	7,300kHz

The band limit frequencies are automatically updated to the appropriate region code setting.

Beep Tone On/Off
The annunciator beep can be enabled or disabled in this menu. Rotate the knob clockwise to enable the beeps, anticlockwise to disable.

Alarm On/Off
The out-of-band alarm can be enabled or disabled in this menu. Rotate the knob clockwise to enable the alarm, anticlockwise to disable.

Note: The Low/High input voltage alarm is permanently enabled.

Auto-Manual Alarm Reset
This setting adjusts the behaviour of the input voltage alarm. Rotate the VFO knob to select either On or Off. If set to On, then the alarm will automatically reset if the input voltage returns to within the nominal tolerance limits. If set to Off, then once the alarm has been triggered, it will continue to display the current voltage and blink the message and optional LED until manually reset by means of a brief press of the VFO button.

If the out-of-tolerance condition still exists, the alarm will be re-triggered in either setting. The alarm will be triggered if the input voltage is lower than 10.80V or higher than 15.20V. In order to avoid both false triggering as well as display jitter, once the alarm has been triggered its threshold is altered by 20mV. Hence, in order for the alarm to be cancelled either automatically or manually, the input voltage has to be greater than 10.82V or lower than 15.18V.

The lower threshold was chosen with portable operation using a sealed lead-acid battery as the power source. These batteries should not be discharged to a voltage lower than 10.50V as an absolute minimum. In addition, the low drop-out 10V regulator on the main board has a worst case input voltage limit of 10.60V, thus a conservative limit of 10.80V was chosen to ensure that the transceiver would continue to operate and that the battery would be protected against excessive discharge.

160m/80m/40m Preset Frequencies
The next three items allow you to preset the frequencies used for the rapid band-switch feature. Use the VFO knob to adjust as desired.

Note: The Rapid Band-Switch preset frequencies can only be adjusted in the System Setup Menu. The VFO-A/B frequencies can be set and adjusted from the normal operating mode, and saved.

Note: Since the VFO frequency step selection is not available in the System Setup menu, the frequency step is set to 1kHz.

CW Shift Offset
Adjust this to your preference. The standard default value is 700Hz, the limits are from 300Hz – 1000Hz.

Note: You should also alter the hardware side-tone injection frequency on the main board with R53 so that it corresponds.

Voltmeter Calibration
The output of the A-D convertor will be displayed showing the input voltage. Measure the actual voltage at the input power connector, and use the VFO knob to adjust the display to agree as closely as possible.

Note: The A-D convertor is a 10-bit unit and has a resolution of 20mV. Each step of the VFO knob alters the reading by 10mV. It may not be possible to obtain an exact match. The voltage measurement is averaged over 32 samples, to reduce the effect of noise on the supply line. This was particularly evident leading to display jitter when powered by some small switch-mode power supplies.

Briefly press the VFO button to step back to the first menu item.

To exit the System Setup menu, press and hold the VFO button for at least 1.5S, and release. The message:

```
Settings Saved
```

will be displayed, and the transceiver will drop into its normal operating mode.

Alarm Warnings
If the input voltage is out of tolerance, then Mode 2 is invoked and the current input voltage is displayed, whilst the right-hand side of the display blinks and shows the message:

```
HI/LO V!
```

and, if fitted, the optional alarm indicator LED blinks.

With this version, the user can select whether the voltage alarm will automatically cancel, or whether it will latch and have to be manually reset.

If the current frequency is outside a recognised amateur band, and the out-of-band alarm is enabled, then the decimal points of the normal frequency display will blink, and the optional alarm indicator LED will also blink. When the frequency returns to within an amateur band, the alarm automatically cancels.

Note: Only the carrier frequency is checked. When operating DSB take note that in order to comply with licencing requirements, the sideband energy must also be within the amateur band limits. To ensure you comply, do not operate closer than 3kHz to either the upper or lower band edges in DSB mode, and no lower than the CW offset frequency plus the lower band edge in the CW mode. As there is no direct hardware sense input for the CW/DSB switch it is not possible to automatically ensure the alarm is given for both CW and DSB. There will be an alarm if you close the key in the CW mode and the carrier frequency is outside the band edge.

Annex 1

IMPROVING THE SYNTHESISER ACCURACY

The original firmware calculated the Direct Digital Synthesiser's control word directly in the DDS domain, and the frequency increments were thus approximations to the decimal frequencies they represented. Consequently, as the frequency of the VFO was incremented or decremented, after a sufficient number of steps had occurred there would be an increasing quantisation error.

As an example, the DDS control word is calculated as follows:

DDS = Frequency * 2^28 / 20,000,000

Thus, for example, the DDS control word for 10Hz is:

DDS = 10 * 2,684,354,556 / 20,000,000

= 134.2177280

Since the word can only be an integer, this is truncated to 134.

The output frequency of the synthesiser is calculated as follows:

F = DDS * 20,000,000 / 2^28

and the actual output frequency for a DDS control word of 134 is:

F = 134 * 20,000,000 / 2,684,354,556

= 9.9837780Hz

This gives an incremental error of −0.0162220Hz.

With each step of the VFO knob this error will accumulate and after 308 steps the accumulated frequency error will be: −4.996376Hz.

In other words, with every 3kHz of frequency change at the nominal 10Hz step rate, there will be approximately −5Hz error with respect to the displayed frequency.

Whilst in practice this error is quite insignificant, and well within the normal limits for commercial amateur radio equipment, nonetheless since the basic frequency resolution of the synthesiser is approximately 0.0745Hz, it is an interesting challenge to see if the frequency error can be reduced.

The problem arises as a result of the inability of the compiler to calculate to 64-bit accuracy. The expression to calculate the DDS word requires the multiplication of a 32-bit number by a

32-bit number which leads to a 64-bit product. This has then to be divided by another 32-bit number to give the required DDS word.

The microprocessor chosen is only an 8-bit machine, and whilst in principle the requisite software could be written to handle 64-bit arithmetic, it would be quite tedious to implement, would require a large amount of memory for the various registers, and would also be fairly slow.

The compiler used does not offer any 64-bit data objects, the largest object is a 32-bit long integer or a 32-bit floating-point data type. Whilst the calculation could be performed using 32-bit floating point arithmetic and then converting the result to a 32-bit long integer, this would inevitably lead to various approximations, and would also require the floating point math package to be included.

During the development stage this approach was considered, but with the requirement to include the other features there was insufficient memory available. Nevertheless, there is a way!

The classic way of overcoming this problem and avoiding the need for either 64-bit integer or 32-bit floating point calculations, especially with limited microprocessors, is to use a look-up table.

Given sufficient memory, the obvious way would be to create a table with every frequency from 10Hz – 8MHz in 10Hz increments, and use the decimal frequency as the index into this table to retrieve the requisite DDS word. There would thus be 800,000 entries, each with 3 bytes for the 24-bit DDS word, and thus would only require 2.4Mb of memory.

If this was for a Windows application, judging by what I have observed, some Windows programmers seem to think that the target machine has an infinite amount of memory and disc space, and I can well imagine a sloppy programmer actually implementing this approach!

However, suppose instead we create an array for the MHz decade, another for the 100kHz decade, another for the 10kHz decade and so on down to the 10Hz decade, the amount of memory then required for the 24-bit DDS words would be 6 decades x 10 entries x 3 bytes = 180 bytes, now that is an improvement!

However, if you think about it for a moment you will realise that we do not need all these decades, because the values for the 100kHz decade are already contained within the MHz decade, and, for the 10kHz decade within the 100kHz decade and so on. Thus we only actually need an array with the values for the MHz decade, and we can easily obtain the lower order decades from it.

Now the array size reduces to only 40 bytes – well within the capabilities of this microprocessor since the array, being a constant, will be stored in the program flash memory, not the limited RAM. (40 bytes – each entry is an unsigned long integer, occupying 4 bytes.)

To illustrate, consider the MHz entry for 5MHz. The following table shows the calculated decimal values of the DDS control word for each lower decade:

FREQUENCY, Hz	DDS CONTROL WORD
5,000,000	67,108,864
500,000	6,710,886
50,000	671,088
5,000	67,108
500	6,710
50	671
5	67

It can be easily seen that each lower decade is merely the higher decade divided by 10. Thus by separating out each decade's digit from the frequency, the composite DDS control word can be calculated by adding together all the individual decade control words.

The question then arises, what sort of accuracy would result? Using the table above, suppose the desired output frequency was 5,555,555Hz. The actual control word if it was calculated exactly would be:

DDS = 5,555,555 * 2^28 / 20,000,000

= 74,565,396.9880

Even if this were rounded to the nearest integer it would then be: 74,565,397. If this is used then the actual output frequency would be:

F = 74,565,397 * 20,000,000 / 2^28

= 5,555,555.0009Hz

If instead we were to sum the individual words from each decade the result would be:

67,108,864 + 6,710,886 + 671,088 + 67,108 + 6,710 + 671 + 67

= 74,565,394

This would give an output frequency of:

74,565,394 * 20,000,000 / 2^28

5,555,554.77738Hz,

representing an error of only −0.22352Hz compared to the result obtained from direct calculation.

In fact, if we rounded each decade as it was summed this would give a DDS word of: 74,565,397, exactly that obtained by direct calculation.

An analysis was done using every frequency from 10Hz – 8MHz, comparing the output frequencies obtained by means of both the direct calculated method and the look-up table method, and the worst case error was only 0.7Hz, for the vast majority of frequencies, the errors were far less.

This is the synthesis method employed in this version of the firmware, and the display calculations are performed in decimal, so that the approximation errors are absent, only the minor approximations of the look-up table. Even so, with all the additional features incorporated, only 9 bytes of flash memory remain.

If you intend to do some more modifications, bear in mind that this executable file was obtained from a highly optimising compiler – a straight compiler will almost certainly be unable to accommodate all of the functions and features within the 4,096 byte memory space. Indeed, at one point in the development when even the optimising compiler was giving out-of-memory errors, I considered writing the code in assembler.

The code requires a careful assessment of the use of the most compact data types for the required function, as well as planning the use of global and local variables.

This microprocessor implements only an 8-level stack for functions and interrupts. The data stack is implemented in a separate area of memory. Thus, the larger the type of object you wish to pass to a function the more contiguous memory is required.

Although there are 386 bytes of RAM available, not all of it is contiguous, and much of it is already used by the various registers of the machine. Thus placing data objects in global memory is a way to economise on the stack space required, since many functions can then be implemented with the void parameter so that no objects are passed on the stack. Equally, if the return specifier is also void, then no stack space is required for the return values.

If temporary objects are required, then they are best implemented inside a function or a block of code. As these objects are not permanent, being created only as required, the compiler can then re-use small areas of RAM. Only if the objects are specified as static do problems arise.

A static specifier means that this object persists, and its value must remain from one invocation of a function to another. Thus, the memory used for this object cannot be re-used, and this then subtracts from the available RAM.

It is therefore a matter of careful planning to establish just how many variables can be safely re-used, and which ones need to be global, and others which require to be static.

As a result, the source code is not necessarily an example of ‘good’ ‘C’ programming since a number of measures were taken simply to reduce the amount of memory rather than to achieve ‘best’ programming practice. As I ran short of memory I experimented with a variety of differing loops and other constructions and compared the size of the compiled code to select the most memory efficient techniques.

The overall accuracy is of course dependant upon the accuracy of the crystal oscillator. The crystal chosen has an intrinsic tolerance of ±30ppm giving rise to a possible error of ±600Hz at 20MHz. The adjustable capacitor, C20, has sufficient range so that this can be fairly easily trimmed out.

The easiest way is to measure the DDS output frequency with an accurate frequency counter, but it is also possible to use a local broadcast station as a frequency reference. For most purposes simply switch to CW and set the displayed frequency to that of the carrier. Then carefully adjust C20 for the best zero beat.

Alternatively, if your frequency counter is not particularly accurate, then tune the receiver to a known carrier, and offset it by an exact amount, say, 1kHz, and measure the beat frequency. Now adjust C20 for exactly 1kHz. Even if the counter is out by, for example, +100ppm, this only amounts to an error of +0.1Hz at 1kHz.

Note that since the receiver is a DSB receiver, if you offset your receiver on the high side, and your crystal is high, then the beat frequency will also be higher than 1kHz. Conversely, if you offset 1kHz low, and your crystal is high, then the beat frequency will be lower than 1kHz. This allows you a method of determining whether your crystal oscillator is high or low.

Nevertheless, by adjusting C20 to correct the beat frequency you will correct the oscillator.

Annex B

IMPROVING THE CALIBRATION OF THE VOLTMETER

The original firmware did not contain any method of calibrating the voltmeter. The designers suggested instead that you selected suitable resistors for the potential divider to achieve the best calibration. Whilst this method certainly works, I dislike having to solder and unsolder these components from the board for fear of eventually damaging it – especially as the calibration can be adjusted in software.

The supply voltage is measured by sampling the voltage obtained from a divider chain of resistors R23 (910R), R17 (15K), R1 (15K), and R22 (10K). The voltage measured at the junction of R1 and R22 thus scales the input voltage so the full-scale of the A-D converter represents an input voltage of 20.48V using the +5V supply as the reference voltage.

Since the A-D convertor is a 10-bit unit, its full-scale value is 1024, and thus for a full-scale value of 20.48V the resolution is 20mV.

And, as in all good university lectures, it will be left for the student to elucidate the simple proof!

The calibration is adjusted in the System Setup menu. Press and hold the VFO button until the System Setup message is displayed, and then release.

Step through the menu items using a short push of the VFO button to select the voltmeter calibration. Measure the input voltage with an accurate 4-1/2 digit multimeter and rotate the VFO knob to achieve the best match.

The actual calculation being performed is:

A-D * 2048 / Calibration Factor

Thus, with an input voltage of 13.80V, and for a nominal calibration factor of 1024, if we assume that the resistor chain is precisely accurate, and that the reference voltage is exactly 5.000V, then the input voltage to the A-D convertor is:

$$V_{in} = 13.80 * 10 / (0.91 + 15 + 15 + 10)$$

$$= 13.80 * 10 / 40.91$$

$$= 3.373V$$

This will give an A-D count of:

$$1024 * 3.373 / 5.000$$

$$= 690$$

Using the default calibration factor of 1024, this results in a value of:

$$V = (690 * 2048) / 1024$$

$$= 1380$$

This is displayed as: 13.80V

The values chosen allow the calibration to be made in 1/2LSB increments, i.e, 10mV steps, so it should be possible to achieve a close match as the calibration accounts for both the tolerance of the +5V regulator as well as the resistor chain.

The limit values for the calibration factor for a 5% resistor tolerance, and a ±100mV reference voltage tolerance are 930 – 1128. The actual values used are 930 – 1130.

The displayed battery voltage is the final value after averaging over 32 cycles of the main loop. When displaying the voltage, the main loop cycles in approximately 8mS, thus the sampling period is about 250mS, i.e, about 4 samples/Sec.

During the development phase a considerable amount of display jitter could be observed at certain input voltages since the original firmware sampled the voltage on every cycle of the main loop and displayed it each time through. When using a simple switch-mode power supply the somewhat spiky noise these supplies tend to produce would cause the A-D convertor value to rapidly change between two values leading to display jitter.

To overcome this, as well as reduce the incidence of false alarm triggering when the supply voltage was close to the switching threshold, the value from the A-D convertor was averaged. The initial averaging was 64 samples, but this introduced a somewhat lengthy response time. 32 samples proved to be more suitable.

In addition it was found that when changing the VFO step rate, the voltmeter display would also ‘glitch’. This is because the tone output uses the same pin as the A-D convertor input, and the decoupling capacitor C14, and the coupling capacitor C36 have to recharge to the supply voltage. The ‘glitch’ is simply the result of these capacitors re-charging. Nevertheless, it is unsightly and thus the software inhibits the A-D convertor for a short period of time after any tone output has occurred, thus minimising the glitch.

Annex 2

JUMA FIRMWARE

The downloaded archive contains not only the C source code for the firmware, but also two HEX image files. The file: `trx1_rit.hex` contains the firmware, but with no EEPROM contents.

The file: `Juma TRX-1 v7.1a + Std EEPROM.hex` contains the same firmware, but in addition it also contains the standard default EEPROM contents for the battery calibration, CW shift, 160m/80m/40m rapid band-switch frequencies, IARU/USA and Beep options.

STANDARD DEFAULT VALUES

Region	IARU Region 1
Beep	On
Out-Of-Band Alarm	On
Auto Reset	On
160m Preset Frequency	1,900kHz
80m Preset Frequency	3,600kHz
40m Preset Frequency	7,100kHz
CW Transmit Offset	700Hz
Voltmeter Calibration Factor	1024
Selected VFO	VFO-A
Tuning Rate	10Hz

To improve the look of the S-meter, the scale was changed, and to avoid the right-hand character encroaching on the frequency display, the scale length was shortened. To preserve the overall dynamic range of the S-meter, the input number is scaled by a factor 7/8.

Unfortunately the 16F88 microprocessor chip does not contain a boot loader, so the HEX image file has to be loaded via a device programmer. The easiest way to do this is to use the MicroChip PICkit-3 USB programmer which is readily available from the MicroChip website.

This programmer can also use the MicroChip stand-alone programming software package, it is not necessary to load the full IDE. The file is: `PICkit_3_Programmer_1_0_Setup.zip` If you have difficulty in locating this package, send me an email and I will send it to you.

This is the package I used during the development program, and it has been found to work quite well. Although the manufacturers do not recommend using it as a stand-alone programmer, in my experience it actually works better this way than using an auxiliary 5V power supply for the chip. The relatively modest drive current of about 30mA is easily within the capabilities of a standard USB port.

The beauty of this programmer is that you can simply use a small logic breadboard, readily available from Radio Shack, plus a set of their standard pin plug jumper leads, and connect the programmer to the 16F88 as follows:

PICkit-3 Pins	Signal Name	16F88 Pins
1	Vpp	4
2	V_{DD} (Target)	14
3	V_{SS} (Ground)	5
4	PGD (ICS Data)	13
5	PGC (ICS Clock)	12

Launch the PICkit-3 application, and first select the device, 16F88. Next, from the file drop-down menu select Import Hex, and navigate to the folder containing the HEX load file. Double click on this file to load it into the application, and click on the Write button. The device will be erased, then the programming operation will commence and within a few seconds the device contents will be verified, completing the operation. Simply replace the chip, with the new firmware.

If you load the file: `trx1_rit.hex` then you will have to manually enter the default settings. If you fail to do this the software will either not work, or will operate in a strange way.

To enter the default values, press and hold the VFO button, and turn the power on. After the splash screen you will be presented with the System Setup message, release the button at this point.

The first item is the Region selection. Rotate the VFO knob clockwise to select the USA region or anti-clockwise to select IARU Region 1.

The next item is the Beep On/Off selection. Use the VFO knob to select your preference.

The next item is the out-of-band alarm. Use the VFO knob to enable/disable this alarm.

The next item is the auto alarm reset. Use the VFO knob to select on or off.

The next 3 items are the Rapid Band-Switch frequency settings. The lower band edge frequencies were already selected when the region setting was displayed. Use the VFO knob to set the frequency to your favourite part of the band.

The first band is 160m, followed by 80m, and finally 40m.

Note: The upper and lower band edge frequency limits are automatically set when setting the Region setting.

The next item is the CW Offset frequency. The default is 700Hz, adjust to your preference, and do not forget to adjust the side-tone oscillator on the main board to agree.

The final item is the voltmeter calibration. Measure the input voltage with an accurate multimeter and adjust with the VFO knob to obtain the same reading.

Note: Because as already stated, the voltmeter's resolution is 20mV, it may not be possible to get the least significant digit to exactly match.

Press and hold the VFO button for at least 1.5 seconds, then release and the settings will be saved. The remainder of the boot-up sequence will occur, and you will finally be presented with the Mode 0 operating display.

The selected VFO frequency will be displayed as out-of-band and 0.000.00 – select a suitable amateur band frequency.

Briefly press the VFO button and select VFO-B. In all probability its frequency will display in a bizarre manner, usually as: Σ.967.29 because of the blank EEPROM's contents. Using a medium press of the VFO button, simply copy the inactive VFO's contents. Exit by means of a brief push of the VFO button.

Using a medium push, select Mode 2 and adjust the step rate to 10Hz. Exit from this mode with a long press of the VFO button to save the current transceiver's state, thus updating the EEPROM.

If you wish, you can simply load the second file: `Juma TRX-1 v7.1a + Std EEPROM`. This HEX file was downloaded from my microprocessor after the default values had been installed, and will obviate the need for you to insert them. Simply power up the transceiver.

If necessary you can re-enter the System Setup menu to calibrate the voltmeter and adjust the band switch preset frequencies to your preferences as well as alter the region and beep settings.

Note: You MUST update the EEPROM with at least the standard default settings! Failure to do this and then attempting to use the transceiver with a blank EEPROM will produce quite erroneous and bizarre operation! I strongly suggest loading the full package including my standard default EEPROM settings; you can easily customise them to your preferences.

Note: When saving the settings from the System Setup menu the existing VFO selection, VFO A/B frequencies, and VFO step rate are preserved, only the Region, Beep, Rapid-Bandswitch frequencies, and Voltmeter calibration factor are changed.

Annex 3

SYSTEM WIRING CHANGES & OPTIONAL LED

RIT Switch

For the firmware to operate as designed, it is important to ensure that your RIT switch is wired correctly. One side of the switch is connected to J9 on the DDS/Display module, the other side of the switch must be connected to the TX pad on the main board. This ensures that the RIT offset is cancelled in transmit.

If you had previously selected the RIT switch function to be RIT On/Off, then you will have made this type of connection already. If on the other hand you had it configured as the VFO A/B select switch, then you will need to re-wire it as described above.

Alarm LED

The spare output from pin J6 of the DDS/Display module is now programmed as the alarm indicator output. If you wish to use this feature you will need to obtain a suitable high-efficiency high-intensity LED and panel mount or holder.

You can choose whether to have the LED normally on to act as a power indicator, or normally off. In either case it will blink if either the input voltage is outside nominal tolerance, or if the current VFO frequency is outside a recognised amateur band.

In my case I replaced the LCD with a similar unit equipped with a backlight, and thus there was no need for the panel LED to be illuminated. If your LCD is the original non-backlit version, then it is useful to wire the auxiliary LED as a power indicator.

On a standard LED, the anode is the longer lead.

To use the LED as a power indicator, connect a resistor in series with the LED and wire it to the anode of the LED. Connect the cathode of the LED to the J6 pad, and the resistor to the V_{cc} pad on the module, a suitable point is J1-5, the +5V drive voltage for the optional keyer.

If you wish the LED to be normally off, then connect the resistor to the anode of the LED, and connect the cathode to ground. Connect the resistor to pad J6.

The 74HC74 chip is rated to source or sink a maximum of 25mA, but by using a high-efficiency high-intensity LED the drive current can be substantially less than this maximum. In my case 0.5mA was more than sufficient for an easily visible indication. In which case the series resistor is 3.9K. Any value between 1K and 4K7 is suitable.

Annex 4

OPTIONAL KEYER

An option with the kit is the keyer module. This module allows for automatic dots and dashes to be sent using either a single paddle key or an Iambic key.

The wiring is normally such that the dot contact is connected to the tip, the dash contact to the ring, and the screen is ground.

Power is provided from the regulated +5V available from J1-5 on the DDS-Display module, and the ground return to J1-6.

During power up, the default mode is Iambic, however, if the dot paddle is held closed and the transceiver powered up, then the keyer operates in the Dot Priority mode.

If the dash paddle is held closed and the transceiver powered up, the keyer operates in the straight mode, simply repeating the keying on the dot paddle.

Annex 5

QUICK REFERENCE CHART

MODE	SHORT PUSH	MEDIUM PUSH	LONG PUSH
System Setup	Select Next Item	------------------	Save to EEPROM
Mode 0 RIT Off	Select Mode 1 VFO A/B Select	Select Mode 2 Tuning Step Select	Select Mode 3 Rapid Band-Switch
Mode 0 RIT On	Cancel RIT Offset	Select Mode 2 (Display Only)	------------------
Mode 1 VFO A/B Select	Exit to Mode 0	Copy Inactive VFO	Save to EEPROM
Mode 2 Tuning Step Select	Exit to Mode 0	Exit to Mode 0	Save to EEPROM
Mode 3 Rapid Band-Switch	Exit to Mode 0	Exit to Mode 0	Save to EEPROM