



*Making Communications Fun Again!*



# DMX-40 REFERENCE MANUAL

*By Eric C Anderson  
PreppComm  
Version 3.1, December 2020*

Copyright © 2020 PreppComm. All rights reserved.

# Table of Contents

Title Page	1
Table of Contents	2
1. <a href="#">Overview</a>	5
2. <a href="#">Warranty and Cautions</a>	7
1. <a href="#">Important Notice</a>	7
2. <a href="#">The Elephant in the Room</a>	8
3. <a href="#">Getting Started... Quickly</a>	8
3. <a href="#">Getting Acquainted</a>	10
4. <a href="#">Initial Setup</a>	13
5. <a href="#">Setup with External Transceiver</a>	14
6. <a href="#">Power On Walk-Thru</a>	15
7. <a href="#">Menus</a>	18
1. <a href="#">Local Station Setup</a>	19
2. <a href="#">VFO Adjust</a>	21
3. <a href="#">Typing Speed</a>	22
4. <a href="#">Microprogramming</a>	24
5. <a href="#">MHz Memories</a>	25
8. <a href="#">Keyboard Controls</a>	27
1. <a href="#">Receive Mode Keyboard Controls</a>	27
2. <a href="#">Decoder Mode Control Key</a>	28
3. <a href="#">Gain Control Key</a>	29
4. <a href="#">Editing Keys</a>	30
5. <a href="#">Frequency Control and Tuning</a>	30
6. <a href="#">RIT Tuning</a>	31
7. <a href="#">Frequency Memories &amp; Frequency Memory Editor</a>	32
8. <a href="#">Function Keys</a>	34
9. <a href="#">Transmit Keyboard Controls</a>	34
9. <a href="#">Decoder Operation</a>	36
1. <a href="#">HCW Mode</a>	36
2. <a href="#">CTX Mode</a>	37
3. <a href="#">Switching Modes</a>	38
4. <a href="#">Tuning a Station</a>	38
5. <a href="#">Two Frequencies?</a>	40
6. <a href="#">The Tuning Rule</a>	41
7. <a href="#">Switching Bands (Modes)</a>	41
10. <a href="#">Tone Output Control and KEY IN</a>	43

11. <a href="#">CALLING</a>	44
1. <a href="#">Before We Begin...</a>	44
2. <a href="#">General Calling</a>	44
3. <a href="#">Setting Transmit Speed</a>	45
4. <a href="#">Setting a New Default Transmit Speed</a>	46
5. <a href="#">The CALL Sequence</a>	47
6. <a href="#">Setting Up a Directed Call</a>	47
7. <a href="#">Three Ways for Establishing a CCS</a>	49
8. <a href="#">What Happens After You CALL?</a>	50
12. <a href="#">ANSWERING</a>	51
1. <a href="#">The ANSwer Validate and ANSwer Microprogram</a>	52
2. <a href="#">The ANSwer Validate Component</a>	53
3. <a href="#">The ANSwer Start Component</a>	53
4. <a href="#">The Pop-Up Type-Ahead Buffer</a>	54
5. <a href="#">Sending Your Message</a>	56
6. <a href="#">The ANSwer Ending Component</a>	56
7. <a href="#">The INFO Microprogram</a>	57
8. <a href="#">What Happens After an ANSWER?</a>	57
13. <a href="#">Receiving Using RIT Tuning</a>	59
1. <a href="#">Activating RIT</a>	59
2. <a href="#">Deactivating RIT</a>	61
14. <a href="#">Appendix 1: Pro-Signs and No-Signs</a>	62
15. <a href="#">Appendix II: Special Characters</a>	63
16. <a href="#">Appendix III: Calibrating your VFO</a>	64
17. <a href="#">Appendix IV: Terminology</a>	66
18. <a href="#">Appendix V: RPL and User Microprograms</a>	68
1. <a href="#">Introduction</a>	68
2. <a href="#">Built-In Microprograms</a>	69
3. <a href="#">Microprogram Types</a>	70
4. <a href="#">RPL Control Characters</a>	70
5. <a href="#">Built-In Microprogram Review</a>	73
6. <a href="#">Inserting into a Running CALL or ANS Microprogram</a>	75
7. <a href="#">The Microprogram Editor</a>	76
8. <a href="#">Microprogram Help</a>	78
9. <a href="#">Some Examples</a>	79
19. <a href="#">Appendix VI: Keyboard and RPL Layouts</a>	81
1. <a href="#">Receive Mode Keyboard Layout</a>	82
2. <a href="#">Transmit Mode Keyboard Layout</a>	83
3. <a href="#">RPL Command Table Page 1</a>	84
4. <a href="#">RPL Command Table Page 2</a>	85

20. <a href="#">Appendix VII: Common CW Abbreviations</a>	87
21. <a href="#">Appendix VIII: The Tuning Rule, Graphically Explained</a>	89
22. <a href="#">Appendix IX: Alternate USB Keyboards</a>	91
23. <a href="#">Appendix X: Wiring for the 3.5 mm Jacks</a>	92

## 1. Overview

Congratulations for purchasing the new **DMX-40 Morse Code Decoder & Converter Transceiver!** The **DMX-40** was created to make it possible and easy for non-technical people to utilize the HF (High Frequency) Amateur Radio bands for battery-powered emergency communications. This is done by providing a texting interface to the user, making communications via Morse code similar to texting on a modern smartphone: simple, easy, and fun! It also provides a fantastic platform for learning Morse Code, should the operator have the interest in doing so.

The **DMX-40** transceivers are easy to operate, due to its graphical user interface, presented on a 3.5" color touchscreen display. In addition, an included USB/PS2 keyboard is utilized to provide text input as well as control functions.

While the transceiver can be seen as a simple tool for non-technical operators, it also includes many advanced features often not found in much more expensive equipment:

1. Automatic code speed setting: transmitter speed set from received code speed
2. Hyper-active tone extractor and powerful digital filter to extract text in high-noise situations, with extreme noise immunity
3. Super selective decoder (120 Hz wide sensitivity) without audio filtering - you hear full 100 - 10,000 Hz audio, but the decoder only hears a tiny slice around the set tone frequency.
4. Auto limits on transmit frequency based on license levels - impossible to transmit on unlicensed frequencies
5. Automatic capture of remote station call letters
6. Built-in typing speed calibration to limit transmit speed to your comfortable speed, to prevent starving the transmitter while transmitting
7. 76 character type-ahead pop-up buffer for keeping ahead of the transmitter and correcting mistakes before they go out
8. Built-in microprograms for calling, answering, and sending information
9. 12 User definable microprograms for customizing operational modes and messages, using a powerful microprogramming language
10. Full function microprogram editor
11. Word wrap display for easy reading of incoming and outgoing text.

Thus, for the more advanced user, the **DMX-40** transceiver can provide an excellent choice for portable operation, due to its very small size and light weight. It can also easily be placed into an EMP protective container for the purpose of having it available in case of an EMP event, as well. This is one of the features of our **Professional Station GO bag**.

### **Using the Reference Manual**

The manual is written sequentially for you to go through and learn how to use your transceiver. It is also designed as a reference manual, by including all relevant information under any given topic.

Several Appendices are provided at the end of the manual with keyboard layout diagrams and other tables that can be copied or printed out to use to support the learning process. Also a terminology appendix is included in case you need help with any terms you don't understand.

### **Under the Hood**

For those of you with technical interest, the **DMX-40** is based on a triple-processor real-time multi-tasking system architecture. A complex dance of real-time interaction between three interrupt-driven real-time processors makes the magic of the **DMX-40** possible. The receiver is a Direct Conversion Synchrodyne design, with a balanced mixer front-end.

### **Cross-linked Reference**

This manual is provided as a PDF document with many cross-links. Specifically, the Table of Contents links to each section, and there is a link at the top of every page to return to the Table of Contents. Likewise, whenever a reference is made to another section for more information, a link is provided. This makes finding information much easier. Of course the PDF file may also be printed, but then you lose the cross-linking feature. We will work on inventing a way for links to work on paper documents... :)

## 2. Warranty and Cautions

The warranty for this unit is against defects in material and workmanship for a period of one year. **Properly ground the DMX transceiver before transmitting**, especially when the humidity is low and you are getting static shocks from doorknobs, etc. Use either the ground of the SMA antenna connector or a corner screw on the top panel as your ground point.

---

### 2.1 Important Notice: Things Not Covered by the Warranty

1. **Operating your DMX-40 transceiver without a dummy load or properly designed and installed 40 meter antenna** can cause failure of your transmitter PA (Power Amplifier), and **void your warranty**. We recommend an SWR of 3.0 or less.
2. **Operating your DMX-40 transceiver at more than 16 volts** also voids the warranty. This can cause failure of the transmitter Power Amplifier.
3. **Carefully align and plug connectors - do not force**. The side mounted USB and audio connectors are mounted on the rear surface of the front panel printed circuit board. They can be broken off from the circuit board if excessive force is used when plugging in a cable. This type of damage is not covered by the warranty.
4. **Do not scratch or gouge the touch screen with a sharp object**. The touch screen is not like the capacitive touch screen on your cell phone, but rather a less expensive resistive touch panel. This means it requires a little more force, and that there is a thin plastic layer with invisible wires over the glass. It can be cut with sharp objects, which will disable all or part of the touch function. Also, too much pressure on the screen can crack the glass, which can also cause touch failure and ultimately LCD failure. Use a rounded tip or eraser end of a pencil if you want to avoid fingerprints. Use the included microfiber cloth to clean off fingerprints. You can also create microprograms to replace pressing some of the buttons on the Main Screen if you wish so they can be accessed directly from your keyboard. Finally, use the lid provided to protect the front panel and LCD when not in use. Physical damage to the front panel, such as damaged LCD, or damaged LED's is not covered under the warranty.
5. **Take care how you run the keyboard cable**. The keyboard cable can act as an antenna and pick up RF from the transmitter, especially if you have RF in the shack. A low SWR antenna with a good ground or a balanced antenna with a 1:1 Balun will eliminate most RF in the shack problems. Keep the keyboard cable away from the rear of the **DMX-40**, and coil it up at the rear of the keyboard to shorten it (use a tight coil) to reduce RF issues. There is an RF filter inside, but you can overpower it and cause your **DMX** to lose it's mind when you transmit.

**TROUBLESHOOTING TIP:** If your transceiver starts acting weird, the first thing to check is the keyboard wire. If RF gets into the computer "brains" it can cause some very strange behavior! Fix the problem by locating the wire away from the RF, and reboot the transceiver.

---

## 2.2 The Elephant(s) In The Room

**The Big Elephant:** that would be **the antenna**. Yes, you need an antenna, and no, you can't use a "rubber duck" or vertical whip antenna on virtually any of the HF bands except maybe 10 meters and expect good results. For example, for the DMX-40, a 40 meter band unit, this means you need a 40 meter capable antenna. Since you will be running QRP (low power on transmit), having an efficient antenna is pretty important. While we are not going to go into great detail here (this is not the place for an antenna tutorial), please get some help from other amateur radio operators in selecting and installing a good antenna (if you don't know how to do it yourself). Antenna kits are available and range from inexpensive to very expensive, but selecting the right one for your situation that will work well for QRP (there it is again: low power operation) and getting it installed properly will save you loads of headaches - and grant you loads of contacts! For portable operation, the **PreppComm Professional Station GO Bag** includes an easy to install 40 meter end-fed antenna, but it is not designed for permanent installations.

No, this is not a walkie-talkie, but then, you can communicate worldwide directly with the **DMX** transceiver, no repeaters, no internet, no nothing, just you and your transceiver and your antenna. That is what a good antenna will get you. So don't ignore the elephant in the room: get yourself a good antenna, and if that means GET HELP!

**The "Other" Elephant** is ... experience. Once you have it, you can quickly tune around the band. But there is some expertise you will gain over time on adjusting the decoder gain, and the frequency, determining which of the two sides of the signal you are on, and selecting the correct one. And being aware of band conditions, which are related to time of day, sunspots, and other factors. Do not be disappointed if you don't make any contact on your first try! You are not running a 1000 watt station with a 150 foot tower! Patience and practice are very important.

If you have a buddy across town, or across the valley that has a similar rig, that is a great way to start. That way, you can learn the ropes with someone you can talk to over the phone if things are not working the way you expected. So, initially, finding a local ham operator with Morse Code experience, or someone with an **DMX-40** of their own to get some on-the-air experience with is an excellent way to gain confidence and experience.

## 2.3. Getting Started... Quickly

We have provided a **Quick Start Guide** (separate file) to enable you to get started quickly with your new transceiver. We recommend using that as a starting point, and then using this Reference Manual to fill in the gaps in your knowledge of the various functions and features that are not covered in the Guide. We have included many cross links within the document, as well as Table of Contents links and a back link at the top of every page to get you back to the Table of Contents with a single click. This greatly enhances exploring the contents of the manual.

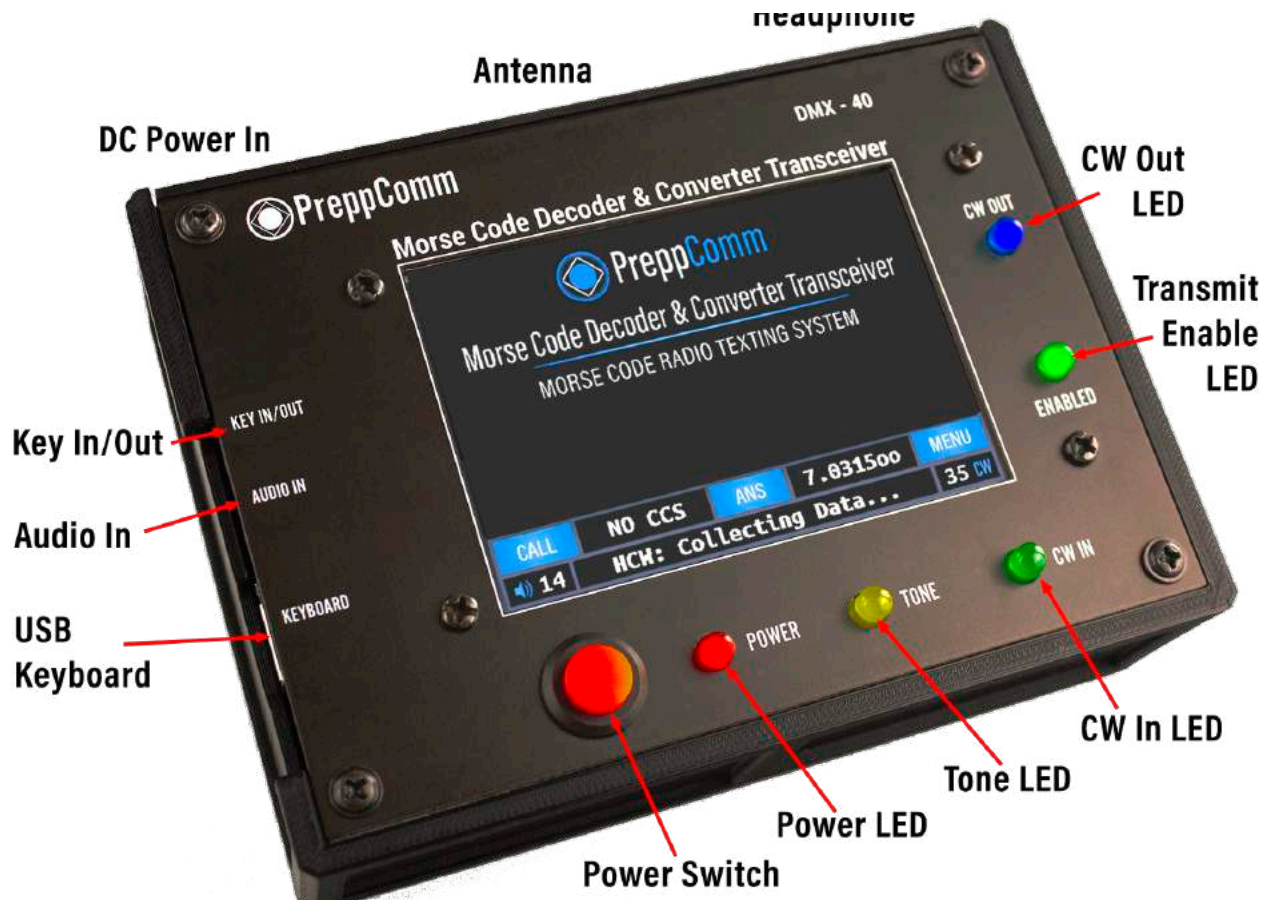
As mentioned before, there are a number of appendices that you should check out early on. We recommend you print out the keyboard command charts, for example, to have next to you as you are learning how to use the various features of your new transceiver. At first, that might seem to be a daunting task, but practice a little, and it will become second nature.

As always, we love to hear from our customers, so if you have questions, comments, or suggestions, please use the Contact Us page on our website:

<https://PreppComm.com>

### 3. Getting Acquainted

Figure 1 is a labeled diagram of the **DMX-40** transceiver:



**FIGURE 1. DMX-40 TRANSCEIVER**

#### **3.5" Color LCD Display**

At the center is the 3.5" color LCD display, which is also a touch screen input device.

#### **Key In/Out Jack**

This is a stereo 3.5mm jack that has one channel dedicated to Key Out, when used with an external transceiver (External Mode). The other channel is dedicated to support a Morse Code key for manual code entry for training and on-air use, if desired. See **Appendix X** for a wiring diagram.

#### **Audio In Jack**

The Audio In jack is used to connect the audio output from an external rig to the internal decoder (External Mode).

### **USB Keyboard**

This is a USB-A female connector for connecting the included USB/PS2 keyboard to the **DMX-40**.

### **CW Out LED**

This blue LED displays the outgoing Morse Code when the transceiver is in transmit mode (either with or without an internal transceiver).

### **Enabled LED**

This green LED indicates when you have set the frequency to one you are allowed to transmit on, based on the license level you have specified in local station setup. If the unit is in External Mode, the LED is always lit, and it is up to you to manage your external rig.

### **CW In LED**

This green LED indicates what the computer is seeing as Morse Code input. It represents the considered opinion of a complex set of analog and digital filters, and is a very accurate representation of the incoming Morse Code signal. This LED is used to assist tuning the receiver to center on the frequency of the sender.

### **Tone LED**

This yellow LED indicates the output of just the analog portion of the filtering system. This LED is normally flickering from noise, as well as lighting brightly during a strong signal. This LED is helpful for adjusting the decoder gain correctly. Decoder gain is controlled from the keyboard.

### **Power LED**

This LED is lit when the power is applied to the internal circuitry.

### **Power Switch**

This is a push-on/push-off power switch for controlling the power to the unit. Normally,

### **Headphone Jack**

The headphone 3.5mm jack is located on the rear panel of the transceiver. Headphone volume is adjusted from the keyboard.

### **Antenna Connector**

The antenna connector is a female SMA connector, and is located on the rear panel of the transceiver. And yes, an antenna for the band you will be operating on, or a dummy load is essential. For best results, do not use a car antenna as they are extremely inefficient (less than 1% efficiency, typically). Rather, use a dipole, end-fed, or NVIS antenna designed for 40 meters.

### **12V Power Connector (Not Shown), Power Consumption and Output Power**

The 12V power connector is located on the rear panel of the transceiver. A 12V adaptor plug with screw terminals (as shown, right) is supplied with your unit. This allows you to wire a battery to plug into the 12V connector. The power connector can be connected to a 12V ½ amp or greater power supply that has a 2.1 x 5.5mm plug on it, or to a 12 volt battery capable of supplying up to ½ amps of power (peak load) via the adaptor. Typical receiver current is 80-120 ma, and for transmit peak is from 380 to 430 ma at 12V power supply voltage. Average power consumption (receive & transmit) is about 2 watts. RF power output is about 2.7W at 12V. Note: Some 12V wall plug power supplies generate a LOT of RF noise. Others are poorly filtered, so you will hear 120 Hz hum. Working off a battery is the best way to eliminate these problems.



The unit can operate at 13.8V as well, where power output rises to 3.5W. If you have a regulated power supply, 16V will give you 4.2W RF output. Maximum recommended voltage is 16 volts, but only with a well-regulated power supply. Factory tests have gone up to 18 volts, but if damage occurs because you have exceeded 16 Volts, it is not covered on the warranty.

### **Ground Connector**

(Not Shown). Any radio transceiver or transmitter requires a good ground to operate properly, and to prevent RF (Radio Frequency) energy from backing up to the unit via the antenna cable, which can cause minor shocks or burns to the operator. Be sure to properly ground your transceiver before using it with anything other than a dummy load. A good antenna with a low SWR (Standing Wave Ratio) will generally eliminate any RF in the shack issues. See more on grounding below.

**WARNING: Operating the transmitter without a properly installed antenna or dummy load may damage your transmitter, and will void your warranty.**

## 4. Initial Setup

The standard setup for the **DMX-40** is shown below. A dummy load is an optional alternative to a 40 meter antenna, or using External Mode, but allows you to fully operate the transceiver without causing any band interference, and without an antenna. A 5W dummy load can be purchased on Amazon for about \$7. Note that an antenna/rig ground wire is not shown.



**FIGURE 2: STANDARD TX/RX SETUP**

An alternative to the headphones is a powered speaker. These can be purchased for as little as \$16 on Amazon. If you purchased the Professional Station GO Bag, it comes with a great set of earphones.

## 5. Setup with an External Transceiver

If you plan to use the DMX-40 as a decoder and encoder for a different transceiver, typically a multi-band 100 watt transceiver, for example, use Figure 3 below.



FIGURE 3: EXTERNAL MODE SETUP FOR DMX-40

Again, here you can replace the earphones with a powered speaker. A 3.5mm splitter cable, available for a few dollars at Amazon, can provide the connection between your external transceiver headphone output, your headphones or powered speaker, and the audio input to the DMX-40 audio in jack. Key Out from the KEY IN/OUT jack requires correct wiring. See Appendix X for wiring diagram. This connects to your external transceiver KEY IN jack, and can handle up to 12V.

**IMPORTANT NOTE:** when operating the external rig **DO NOT USE THE CW FILTER!** Use the SSB filter. Otherwise you will find the decoder not able to operate. The decoder is looking for a signal at around 1300 Hz, which is well within the 300 - 2700 Hz typical filter range of an SSB audio filter, but NOT within the 300 Hz bandwidth of a typical CW filter.

## 6. Power On Walk Through

Once you have set up your transceiver, press the Power switch, you will immediately see the splash screen appear, as shown in Figure 4:



FIGURE 4: SPLASH SCREEN

This screen will be present for a few seconds, and will then be followed by the Main Screen, shown in Figure 5. Each item on this screen is described below.

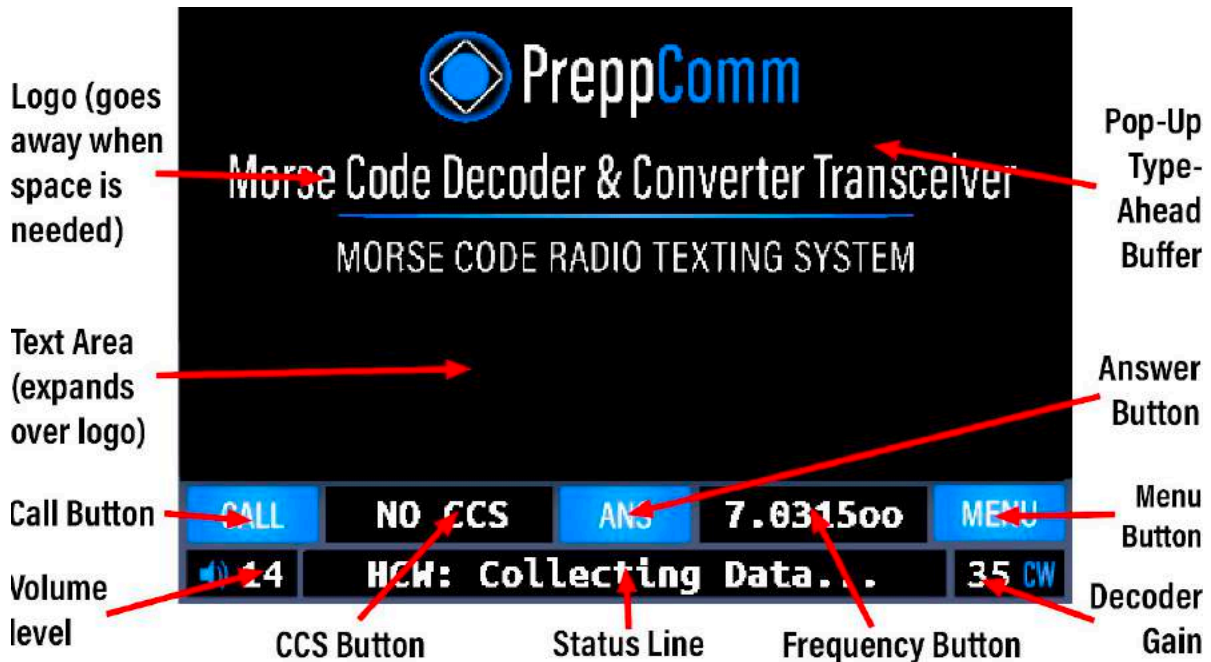


FIGURE 5: MAIN SCREEN

### Text Area

Most of the screen is dedicated for text - both for received text from other stations, and transmitted text from you. The initial screen shows the PreppComm logo, but this will

disappear as soon as the space is needed. Text you transmit is shown in blue. Text you receive is shown in gray. The text area is 37 characters wide and 10 lines high, and is fully word-wrapped - i.e., automatic wrapping to the next line always occurs on word boundaries for easy reading. You cannot directly type into this space: only the encoder (transmitter) and the decoder (receiver) can place text here. Of course you do type most outgoing encoder text for the transmitter, but that first appears on the type-ahead buffer line. It only shows up in the text area as it is transmitted.

### **Status Line**

The Status Line of text is shown below the button line, and is always white text. In Figure 5. The Status Line shows "**HCW: Collecting Data...**" This indicates that the decoder is in HCW mode, and is searching for data to determine the speed of the sender, and analyzing five key characteristics of the sender's technique. Note: When you tune outside of the 40 meter band (7.0 - 7.3 MHz) the Status Line will display "**SW Listening**" for Short Wave Listening. You can tune in many AM short wave stations, including WWV at 10 MHz and 5 MHz, depending on your antenna and band conditions (receiver range is 4.9 - 10.1 MHz). You can also tune in SSB signals using the **DMX-40**.

### **Decoder Gain**

The decoder gain is set to 34 in Figure 5. The range is from 0 to 40. Keyboard commands < and > are used to control this value.

### **Volume Level**

The headphone or powered speaker volume level is set to 14 in Figure 5. This is controlled via keyboard keys , and . (Comma and period). The volume level also goes from 0 to 40

### **CALL Button**

This button controls the sequence of events required for you to make a Directed Call (to a specific station) or General Call to anyone. More on the operation of this control in [Section 11](#). A CALL in progress disables all other buttons on the main screen.

### **CCS Button**

CCS stands for Captured Call Sign. The decoder has the ability to automatically capture the call sign of another station you are listening to. Details on that function are described later in [Section 11](#) and [Section 12](#). NO CCS means just that. This display area is also a button, which allows manual control over the CCS value,

### **ANS Button**

ANS is an abbreviation for ANSWER. This is the other transmit mode. This button provides two separate functions. The first is what we call the answer-validate function, to see if the other station can hear you and thus establish a 2-way conversation. In addition, it allows you to answer back to the other station you are in a conversation

with. Like the CCS Button, it has some advanced functionality we will discuss in [Section 12](#). An ANS in progress disables all other buttons on the main screen.

### **Frequency Button**

This button displays the current transmitter frequency, or a difference or incremental frequency as described in [Section 13](#). As a button, it allows you to select between normal VFO Mode (Variable Frequency Oscillator Mode - where you can receive and transmit based on a frequency you select) and External Mode, where the internal Tx/Rx board is disabled, and you are using an external transceiver for receiving and transmitting, and the DMX-40 is being used for decoding and encoding. In receiver incremental tuning (RIT) mode, pressing this button will revert to the standard frequency mode.

The frequency range of the DMX-40 receiver is 4.9 MHz to 10.1 MHz. The 40 meter band is 7.0 MHz to 7.3 MHz. Depending on your license, the range of frequencies you can transmit on may be more limited Than the 40 meter band. This is the Enabled LED, which lights up on frequencies you can transmit on.

### **Menu Button**

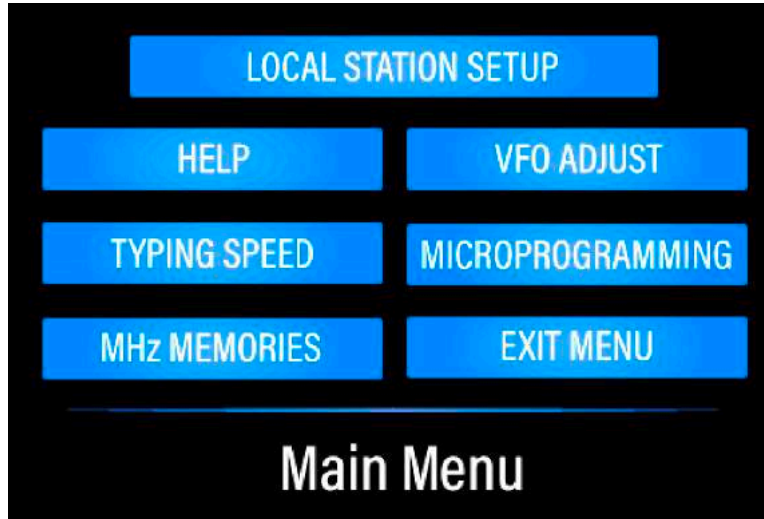
This button brings up the menu, which will be described in [Section 7](#). Also note that while the receiver is still active when you enter the menu and its related functions, any decoder output received during the menu function will be lost. Thus, don't use menu functions while receiving text from another station.

### **Type-Ahead Buffer Pop-Up**

A pop-up "window" for typing your message at the same time the transmitter is transmitting will appear near the top of the text area on the screen. This is a two-line buffer containing up to 76 characters. This pop-up is described in more detail in [Section 12.4](#).

## 7. Menus

The menu system is fairly simple on the **DMX-40**. Figure 6 is the screen displayed when the **MENU** Button is pressed.



**FIGURE 6: MAIN MENU SCREEN**

There are only 6 options on the menu screen, besides **EXIT MENU**, which returns you to the Main Screen. They are:

### **LOCAL STATION SETUP**

This menu item is primarily used the first time you power up, and infrequently used thereafter. More details for station setup are described in [Section 7.1](#).

### **HELP**

This button brings you to a second menu screen, where you can select between **RECEIVE**, **TRANSMIT**, and **MICROPROGRAMMING** or of course **EXIT MENU**. The first two get you to the keyboard command list when in receive or transmit mode, and the third gets you to the list of commands in the microprogramming language that is used in the 12 microprograms you can create to optimize the operation of the **DMX-40** to suit your preferences.

### **VFO ADJUST**

This button accesses the VFO calibration screen. Your transceiver has been factory calibrated at high precision, and you can see the calibration value on this screen. It is not a bad idea to visit the screen, and write down the calibration value for your records. Then press the **FINISH** button to return to the main screen. More information on how to adjust the VFO correction factor is provided in [Section 16, Appendix III](#).

### **TYPING SPEED**

The default typing speed is set to the maximum DMX-40 transmit speed, or 60 WPM. This is not likely to be your comfortable typing speed, so you should select this option, take the speed test, and get the maximum speed reduced to your comfortable typing speed. This feature prevents starving the transmitter from your text input, which would generate poorly timed Morse Code output. More details in [Section 7.3](#).

### **MICROPROGRAMMING**

This button accesses the Microprogram List pages. These will be discussed in detail in [Section 7.4](#).

### **MHz MEMORIES**

This button accesses the frequency memory list. This list and frequency memories in general will be covered in [Section 7.5](#).

### **EXIT MENU**

This button returns you to the Main Screen.

---

## 7.1 Local Station Setup

This menu item is designed for you to provide necessary information to the transceiver about your local station, and leads to a series of display pages to collect from you a set of information to be stored in internal memory. The first screen is shown in Figure 7 below, to collect your call letters:



**FIGURE 7: STORING YOUR CALL LETTERS**

As you can see, the screen text prompts you to enter your call letters. Note that the computer always capitalizes your alphabetic characters, so you don't have to use the

shift key. You can use either the **Delete** and **Backspace** key to delete a character for editing purposes. Simply type in your call letters, and press the **Next** button to enter your call letters into memory.

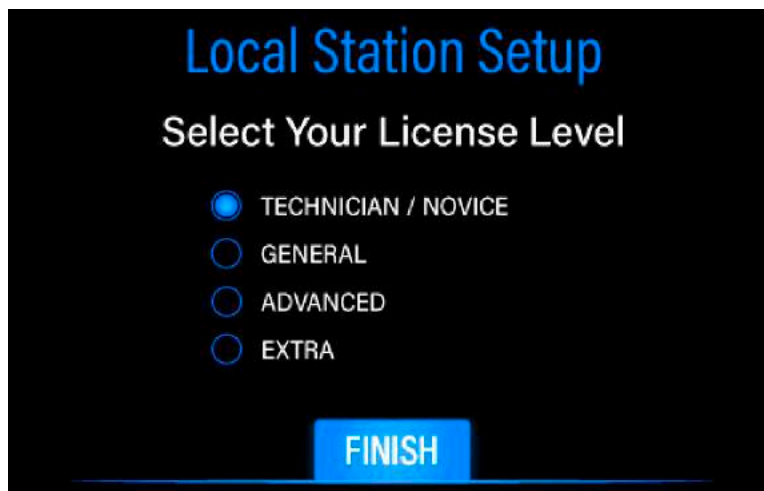
Normally, amateur radio call signs are a minimum of 4 characters, and a maximum of 6 characters. We allow from 3 to 9, so you can add extra characters at the end of your call sign, such as W2OL/RM. for W2OL Remote. Also, there are some uses of the DMX-40 where military units will use them, and they can have 3 character call signs.

Pressing the **NEXT** Button will bring you a very similar screen, but in this case, it is asking for your handle, or name. The progression of entry screens in Local Station Setup is as follows:

1. Call Letters
2. Handle or Name
3. QTH or Location
4. Rig Type (PREPPCOMM DMX-40)
5. Antenna Type
6. License Level

Your location can simply be the name of your town and state, such as **CHICAGO, IL**. Your rig is a **PREPPCOMM DMX-40** (for example). If you don't have an antenna yet, just put **NO ANTENNA** for now. If you have a 40 meter dipole, put **40M DIPOLE**.

Pressing the final **Next** Button on the Antenna screen will display the License Select Screen, which uses radio buttons rather than text input, as shown in Figure 8 below:



**FIGURE 8: LICENSE LEVEL ENTRY SCREEN**

For entering your license level, you can press on the license name of your choice. Select your current license level, and press the **FINISH** Button. You will be immediately returned to the Main Screen. Note that your license level is used to determine which

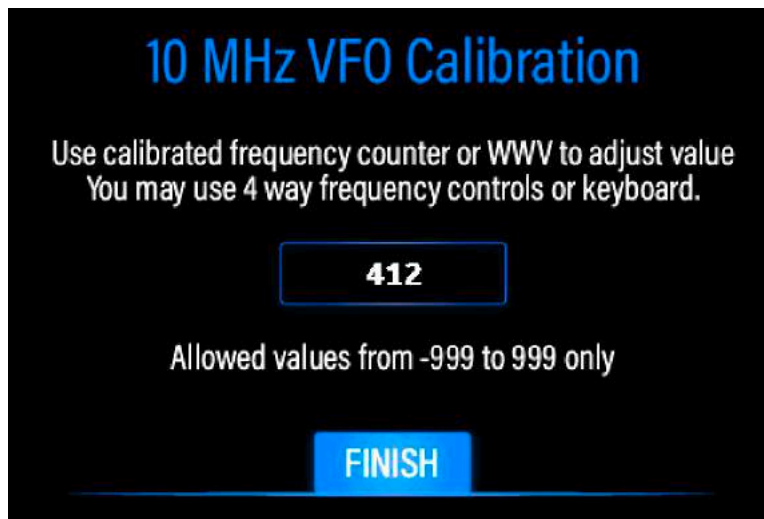
parts of the band you are licensed to operate on. As you move the frequency up and down the band, the Enabled LED will be lit in areas of the band you are licensed to operate in, and will be off for areas you are not licensed to operate in. The **CALL** and **ANS** buttons will only operate when the Enabled LED is lit.

---

## 7.2 VFO Adjust

The VFO Setting screens is shown below in Figure 9. However, if the **DMX-40** is in External Mode (See Figure 5, Frequency Button) when you press the **VFO ADJUST** Button, you will not see this screen. Rather, you will be returned to the Main Screen. This is because - in the External Mode - the VFO is turned off. You must be in Frequency (VFO) Mode to get to the VFO Calibration screen.

Figure 9 shows a value of 412, which means the crystal frequency has been adjusted so at 10 MHz, the frequency synthesizer it has adjusted 412 Hz to correctly generate 10 MHz when it is set to 10 MHz. Your **DMX-40** will have a different number.



**FIGURE 9: ADJUSTING VFO ACCURACY**

It is a good idea to write this number down somewhere in case it gets changed accidentally. Note that when you are in this screen, the VFO is set to 10 MHz, and if you are in a location and time of day when WWV can be heard, you can zero-beat WWV to get within 1 Hz of an atomic-clock controlled frequency. Do not try this if you don't know how to zero-beat a signal, and certainly without writing down the factory number first.

Adjusting the number can be done by a standard editing from the keyboard (backspace to delete, typing in to add), or you can adjust via the left-right arrow keys. The step size is set to 10 Hz steps. You can go to 1 Hz steps using the down arrow, or 100 Hz steps using the up arrow. If you end up not being able to get a zero beat, then

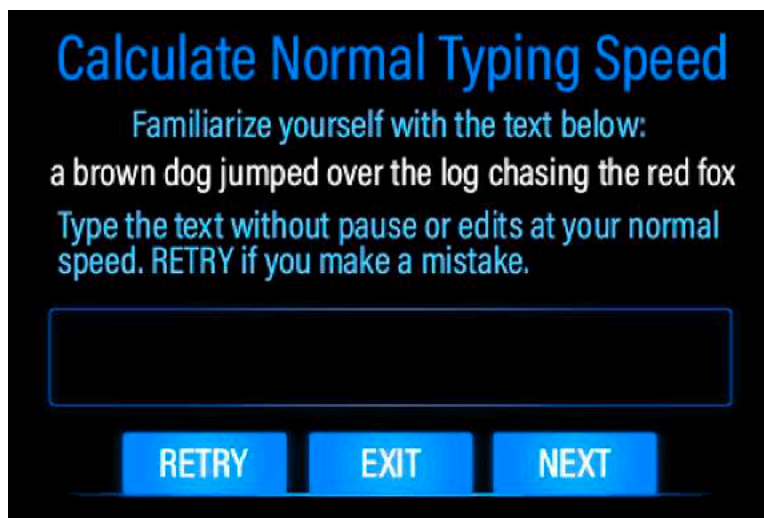
just re-enter the factory number until you learn how to zero-beat WWV. During periods when there is only a click, you cannot zero-beat. There must be a tone being transmitted.

To exit this screen and apply the displayed correction value, press **FINISH**. You will return to the Main Screen.

---

## 7.3 Typing Speed

This function will bring up the following screen, shown in Figure 10a below:



**FIGURE 10A: TYPING SPEED TEST**

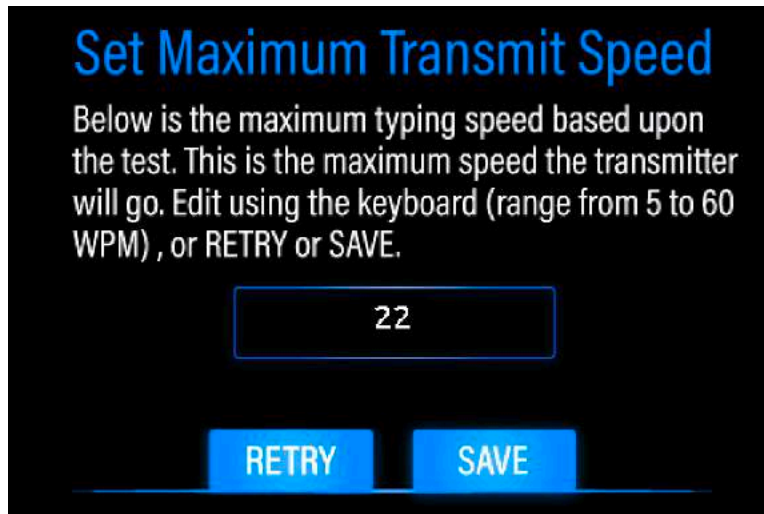
The purpose of establishing your comfortable or "normal" typing speed is to prevent the transmitter from getting away from you, into a situation we call "starving the transmitter." In this situation, you are unable to keep up with the transmitter. This causes two separate problems:

1. The transmitted code falls into a timing situation where each letter is sent as a word, rather than a single letter as part of a word. This is very unhelpful for the person at the other end trying to decode what you are sending.
2. You, the operator, realizing that the transmitter is starved, tend to panic, and start making typing mistakes, adding even more trouble for the person decoding at the other end.

Avoid the by taking this simple test to determine your typing speed.

The reality is, this IS a test, and you will try to beat the clock. It's simply human. Show the computer how fast you can type the line! Well, that is counter-productive, so the

number suggested by the computer on the very next screen takes that into account, shown below in Figure 10b. This screen pops up automatically when you finish the sentence.



**FIGURE 10B: SETTING THE MAXIMUM TRANSMITTER SPEED**

Here, we see the computer measured a typing speed, and assigned its best guess: 22 WPM. Of course you can edit this screen to change the value via the keyboard. It starts out at 60 WPM on first boot, and you should either take the test, or click on the **NEXT** button on the "Calculate Normal Typing Speed" screen, and enter any value you choose without taking the test.

The computer's best guess, for me, was still a little high, even though it is substantially lower than my actual speed measured during the test. That is because - even though told to relax and type at a comfortable rate, I typed at a "speed to the death" rate. There is something about how our minds work: YIKES! This is a TEST! Gotta type it fast!

I had to go back and reduce the number. I ended up around 22. You can simply edit the number on the Set screen from the keyboard. Click **RETRY** to go back to the test, and click **SAVE** to save the value and return to the Main Screen.

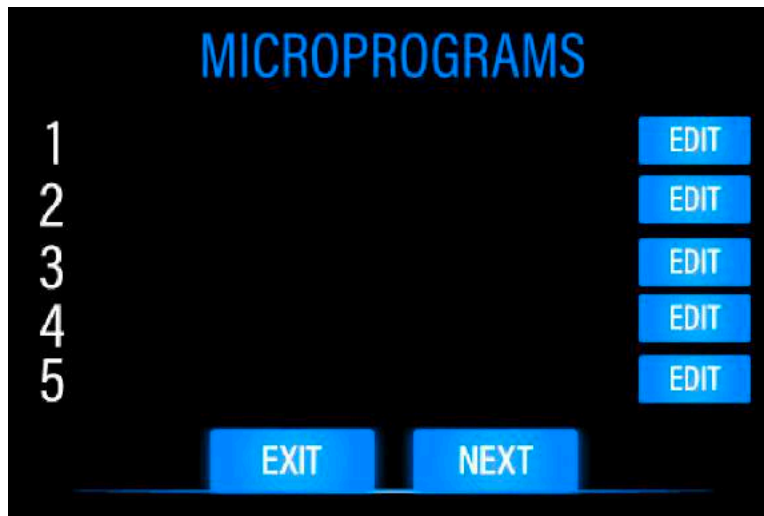
How do you know if you need to adjust this number?

1. **Adjust UP** if you often fill the entire type-ahead buffer when transmitter is running at the maximum rate (check the rate - it could be slower due to slow sender)
2. **Adjust DOWN** if you often find yourself in a situation where the buffer goes empty while you are typing, again, when transmitting at the maximum rate.

## 7.4 Microprogramming

Another term for Microprogramming that is often used is Microprograms. We use the term microprograms rather than micrograms because the programming language is so much more powerful than any amateur radio microprogram language we have ever seen, including allowing total control over the transmitter, subroutine call-like capabilities, repeat loops, delay and timer loops, decision blocks, and much more. With microprograms, you can take full control of how the DMX-40 operates, and control that from your keyboard F1 to F10 keys. F11 and F12 are not functional on the included keyboards, but they can be used as text or commands to be called from the main 10 functions. They can also be chained for longer text messages. Much more about how to use microprograms will be covered in [Section 18 Appendix V](#).

When pressing the **MICROPROGRAMMING** Button in the Main Menu, the first of 3 screens showing quick views of the 12 microprograms is displayed, as shown in Figure 11 below:



**FIGURE 11: PAGE 1 OF MICROPROGRAMS LIST**

On this list, you see only 5 microprograms, number 1 thru 5, and you only see the first part of the microprogram that fits. If the microprogram is too long to fit, a "..." Will appear at the end, just before the **EDIT** button for that microprogram.

**Pressing on the actual text of an existing microprogram will execute it.**

Pressing **EDIT** gets you to the Microprogram Editor screen, which is described in [Section 18.7](#). Pressing **NEXT** will get you to microprograms 6-10, and pressing again will get you to microprogram 11 and 12. **PREVIOUS** buttons will appear when you are on the second or third page, and the last page has no **NEXT** button, of course. Thus,

you can easily navigate through the three screens of 12 microprograms. Each screen also provides an **EXIT** Button to return you to the Main Screen.

The purpose of this list is for you to quickly find the microprogram you are looking for. It is very easy to forget which microprogram is assigned to which function key. In addition, it gives you an easy access to the Microprogram Editor via the **EDIT** Buttons, and finally, allows you to directly execute a microprogram by simply touching it.

In addition, several keyboard shortcuts are provided:

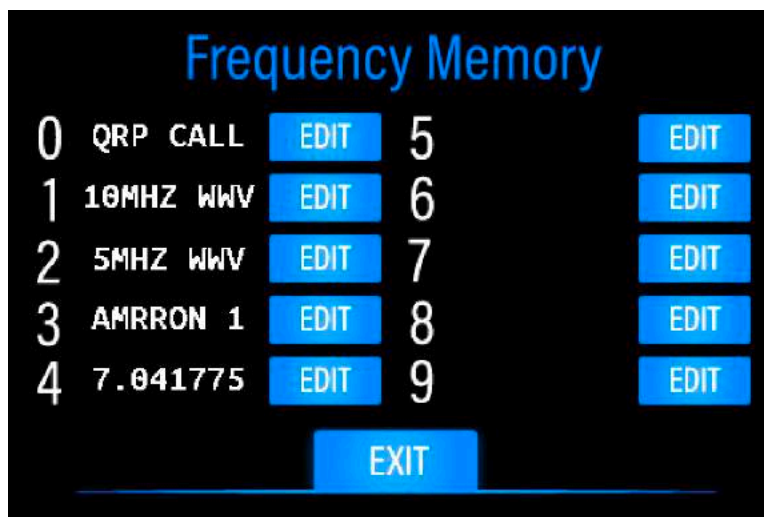
1. **M Key:** The M key takes you directly from the Main Screen to the Microprogram List 1-5. Pressing M again and again gives you a loop, from 1-5, 6-10, 11-12, Main Screen.
2. **F1-F10 Keys:** These keys execute the specified microprogram
3. **Shift F1-F10 Keys:** These keys take you directly to the Microprogram Editor for the specified microprogram.

More information on the Microprogram Editor is provided in [Section 18.7](#).

---

## 7.5 MHz Memories

The DMX-40 provides 10 Frequency Memories for your use. Pressing the MHz MEMORIES Button on the Main Menu brings you to the Frequency Memory List in Figure 12a, or to the External Mode Notification Screen in Figure 12b:



**FIGURE 12A: FREQUENCY MEMORY LIST**

In this example list, you see 5 frequency memory slots define, and five empty. You can also see that text labels can be assigned, or in the case of frequency memory 4, no label is required. The Frequency Memory Editor will be described in more detail in

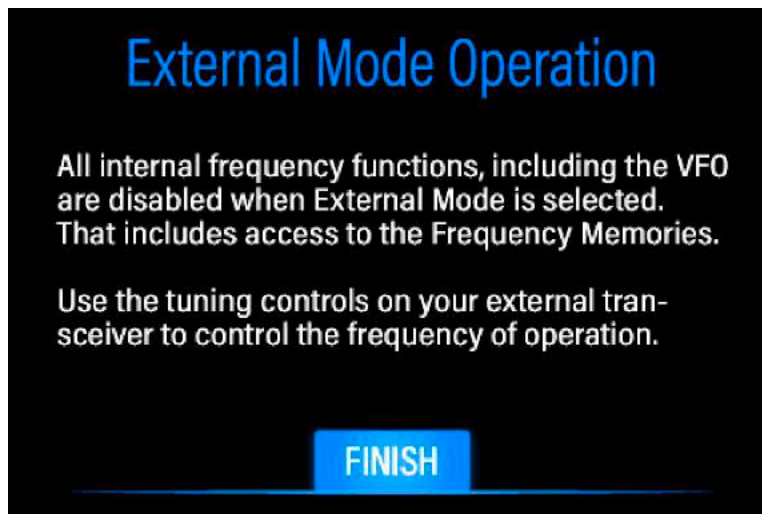
[Section 8.7](#), and is accessed from the keyboard or by pressing one of the **EDIT** buttons on this screen. **EXIT** returns to the Main Screen.

**Pressing on the actual text of an frequency memory, either label or actual frequency, will load it into the VFO and return you to the Main Screen.**

Like the Microprograms List, the Frequency Memory List has keyboard shortcuts:

1. **F Key:** This key brings up this list, and also returns to the Main Screen.
2. **0-9 Keys:** These keys instantly load the frequency stored in the specified memory. These will work from the Main Screen or from the Frequency Memory List
3. **Shift 0-9 Keys;** These keys instantly load the Frequency Memory Editor for the specified memory. These will work from the Main Screen or from the Frequency Memory List

If you have attempted to access the Frequency subsystem, such as the list or editor when you are in External Mode, you will see the screen in Figure 12b instead of the frequency list:



**FIGURE 12B: EXTERNAL MODE NOTIFICATION**

This screen is self explanatory. Press **FINISH** to return to the Main Screen.

## 8. Keyboard Controls

There are two keyboard "modes" that are very different and distinct:

1. **Transmit Mode:** In this mode, most of the keyboard is dedicated to creating your outgoing message. Thus, the standard "typing" keys - A-Z, 0-9, and common punctuation keys transmit exactly what you type. There are several keys that perform other functions, described in [Section 8.9](#) Below.
2. **Receive Mode:** In this mode, in general most keys are used for special functions. However, there is a submode of Receive Mode where the keyboard temporarily reverts to a typing mode similar to Transmit Mode: **Edit Mode**. This mode is entered when you access screens from the Main Menu or from special keyboard keys that require text entry.

### 8.1 Receive Mode Keyboard Commands

The Receive Mode keyboard controls can be seen below in Figure 13:



**FIGURE 13: RECEIVE MODE KEYBOARD COMMAND LAYOUT**

Also, note the keyboard layouts available in [Section 19 Appendix VI](#) for printing out and laying alongside your keyboard while you are learning how to operate the **DMX-40**.

Keep in mind that all alphabetic characters entered are automatically capitalized by the transceiver's computer. If you use the shift key with an alphabetic character out of habit, no harm done. The computer will then not have to capitalize it.

Below is a complete table, if you find that a more useful way to see the Receive Mode keyboard commands. This list is similar to the keyboard help screens accessible from the **HELP** menus.

---

<b>ESC</b>	Switch between HCW and CTX modes ( <a href="#">Section 9</a> )
<b>, (comma)</b>	Reduces headphone volume
<b>. (period)</b>	Increases headphone volume
<b>&lt; (Shift-)</b>	Reduces decoder gain (min = 0)
<b>&gt; (Shift-.)</b>	Increases decoder gain (max = 40)
<b>space bar</b>	Special cases use (described below)
<b>delete</b>	Delete a previously entered character
<b>G</b>	Save current gain settings to memory for next boot
<b>K</b>	Toggles On/Off the Tone Output
<b>M</b>	Shortcut to the Microprogram Lists
<b>F</b>	Shortcut to the Frequency Memory List
<b>backspace</b>	Delete a previously entered character
<b>left arrow</b>	Decreases frequency by step amount
<b>right arrow</b>	Increases frequency by step amount
<b>up arrow</b>	Increases frequency step size 10x (max is 100,000 Hz) in RIT mode, max is 100 Hz
<b>down arrow</b>	Decreases frequency step size 10x (min is 1 Hz)
<b>R</b>	Clears RIT mode, goes to receiver frequency
<b>T</b>	Clears RIT mode, goes to transmit frequency
<b>U, D</b>	U = Move up, D = Move down 2600 Hz (tuning assist)
<b>0-9</b>	Load a stored frequency from a frequency memory
<b>shift-0 - shift-9</b>	Store to or clear a frequency memory
<b>F1 - F12</b>	Execute a stored user-defined microprogram
<b>shift-F1 - shift-F12</b>	Enter the microprogram editor for microprogram 1 to 12

---

**TABLE 1: RECEIVE MODE KEYBOARD COMMANDS**

---

## 8.2 Decoder Mode Control Keys

**SPACE: Decoder Restart:** This key is used to restart data collection for any new signal being received. This allows clean restart with the new signal, rather than combining datasets from multiple signals and thus getting incorrect results **Always restart the decoder for a new signal!**

**ESC: Decoder Mode Control:** The decoder operates in two modes, referred to as HCW mode (human-generated CW) and CTX mode (computer-generated CW). The **ESC**ape key is used to switch between these modes. The initial (default) starting mode is HCW mode. The decoder operation will be described in more detail in [Section 9](#).

Pressing ESC will shift the status line text. In HCW mode, the status line will display:

### **HCW: Collecting Data...**

Once the HCW decoder has determined the speed of incoming Morse code, it will display:

### **DECODING HCW @ NN WPM**

Where NN indicate the current decode speed of NN words per minute. The displayed speed will update over time if it changes, or if the decoder refines its estimate. When in CTX mode, the status line will show

### **DECODING CTX @ 30 WPM**

as no data collection is required. See [Section 9](#) for more details on the decoder.

ESC also does some new things. If the screen goes dark (after 3 minutes of no activity to save power), press ESC to turn on the backlight. If the backlight is on, and you are not on the Main Screen, ESC will return you to the Main Screen. If you are at the Main Screen, then ESC will do as described above.

---

## 8.3 Gain Control Keys

### **, . (Comma, Period) keys: Headphone Volume Control**

The unshifted , (comma) and . (period) keys control the headphone volume. To be clear, it is the same two keys for both decoder gain and headphone volume, but the **shift** key must be used for decoder gain. The initial default headphone gain is set to 10 on startup, but you can both increase and decrease the volume from its initial value. The values range from 0 to 40, and are stored in memory with the **G** key if you wish to change the starting values.

### **< > keys: Decoder Gain Control**

The decoder gain control allows you to adjust the signal level getting to the decoder. The decoder works extremely well when the Tone LED is blinking rapidly. Leave the setting at 30 unless:

1. The CW IN LED is not turning off correctly with the incoming tone. This indicates the signal is too strong, so decrease the gain until you see a faithful following of the incoming tone by the CW IN LED.
2. The signal is too weak, so the CW IN LED is not lighting for all parts of the incoming signal. Increase the gain to 40.t See [Section 9.4](#) for more information on using the Tone LED for decoder gain setting.

Holding the < or > key down (**shift** , and **shift** .) will start up the auto-repeat on most keyboards, for a much more rapid change in volume setting. You should see the gain number in the status display rapidly decreasing or increasing when auto-repeat is active. Note that the decoder gain goes from 0 to 40, and is initially set to 30. You can change the initial value by adjusting the gain and pressing **G** to store the value.

---

## 8.4 Editing Keys

### **Space Bar key: Default Speed Select, Decoder Restart**

When starting the **CALL** and **ANS** functions, the **space bar** can provide special default functions. These are covered in the [Section 11](#) and [Section 12](#). It also allows the user to reset the decoder for a new signal. This use is covered in [Section 8.2](#) on the decoder operation.

### **Delete and Backspace keys**

When entering text in various Edit Mode screens, Station Setup or VFO Settings, or when entering other information into the Status Line, or when typing text to be transmitted, either of these keys allow you to delete a character you typed by mistake (Note that when transmitting, this is ONLY TRUE if the mistake is still visible on the type ahead buffer).

---

## 8.5 Frequency Control and Tuning

**Left/Right Arrow Frequency Control:** These two keys allow you to change frequency up or down. The auto-repeat key function of your keyboard is useful with 100 or 1000 Hz steps for scanning the band. **Note that the frequency display can have both large and small 0's (zeros):**

<u>Displayed</u>	<u>Frequency</u>	<u>Step Size</u>
7.100000	7,100000 MHz	100,000 Hz
7.100000	7.100000 MHz	10,000 Hz
7.100000	7.100000 MHz	1,000 Hz
7.100000	7.100000 MHz	100 Hz
7.100000	7.100000 MHz	10 Hz

7.100000                      7.100000 MHz                      1 Hz                      Initial Step Size

**TABLE 2: FREQUENCY DISPLAY SHOWING STEP SIZE**

Thus, besides telling you the frequency, the display also tells you the step size. The step size is how much of a jump up or down in frequency you will get when pressing either frequency controls. When you first turn on your **DMX** transceiver, the step size is set to 1 Hz, thus all large zeros will be displayed.

For example, if you see **7.030000** and you press the **right arrow** key, you will see **7.030001**. Holding the key down will initiate the auto-repeat function, and the frequency will quickly rise at 1 Hz steps. if you see **7.100000** and press the **right arrow**, it will show **7.200000**. If you see **7.100000** and press the **left arrow**, you will see **7.099000**. The frequency can change as fast as you can press the key, and will correctly respond to the "auto repeat key" function of your keyboard for band scanning. Band scanning is best done with 100 Hz steps, reduced to 10 Hz steps when you are zeroing in on a signal.

### **Up/Down Arrow Step Size Control**

The **up** and **down arrow** allows you to increase or decrease the size of the step by a factor of 10, between 1 Hz and 100,000 Hz. When you are increasing the step size, you are also forcing the frequency to have zeros where the small o's are shown. In other words, if you see **7.352000**, which is 7.352000 MHz, and you increase step size by 10 by pressing the **up arrow** once, you will see **7.350000**, which is 7.350000 MHz. The "2" was dropped off, and thus, your frequency changed by 2KHz when you increased the step size - from 7,352,000 to 7,350,000 Hz. Keep this in mind. If you accidentally increase step size you will probably be changing your frequency, unless there was already a zero at that position in the frequency.

### **U and D keys (Tuning Assist)**

These two keys are used to rapidly move up and down the band. However, the specific purpose and best use of the D key is to drop down 2600 Hz if you are tuned to a station on the "wrong side" of their signal. This is explained in more detail in [Section 8.5](#) and [Section 21 Appendix VIII](#). Note that these two functions will not work in two cases:

1. RIT Tuning is active: since you have already transmitted on a specific frequency, any changes you make should be small. If you need to use the 2600 Hz keys, press the **R** key to turn off RIT tuning, with the receiver frequency becoming the new frequency. Then **U** and **D** keys will operate.
2. If you are closer than 2600 Hz to the top or bottom of the band, the key that would take you beyond the band edge will not work. This prevents accidental excursions outside of the tuning range of the current band definition.

## 8.6 RIT Tuning

After transmitting, the receiver goes into **RIT** (Receiver Incremental Tuning) mode, allowing you to adjust the receiver frequency without changing the transmitter frequency. This is displayed in the Frequency/Band button as **RIT±n**, where **n** is the offset frequency, and the sign tells you which direction the offset is from the starting frequency. If you started at **7.100000 MHz**, and you now see **RIT+300** your receive frequency is **7.100300**. The value **n** does not indicate step size by using small o's as shown in Table 1 above.

Exiting **RIT** mode is possible using the **R** or **T** keys or by touching (pressing) the **Frequency/Band** Button when it displays RIT. **T** will return you to the transmitter frequency (your receiver frequency will change - same as if you set RIT+0). **R** will keep you at the receive frequency, but your transmitter frequency will change to match. Touching the button is the same as typing **T**. The frequency display will return to display the full frequency. More information is available in [Section 13](#).

In **RIT mode**, the initial step size is 1, and you can adjust that to 10 or 100 using the **up arrow** key, or return to a smaller value with the **down arrow** key.

---

## 8.7 Frequency Memories & Frequency Memory Editor

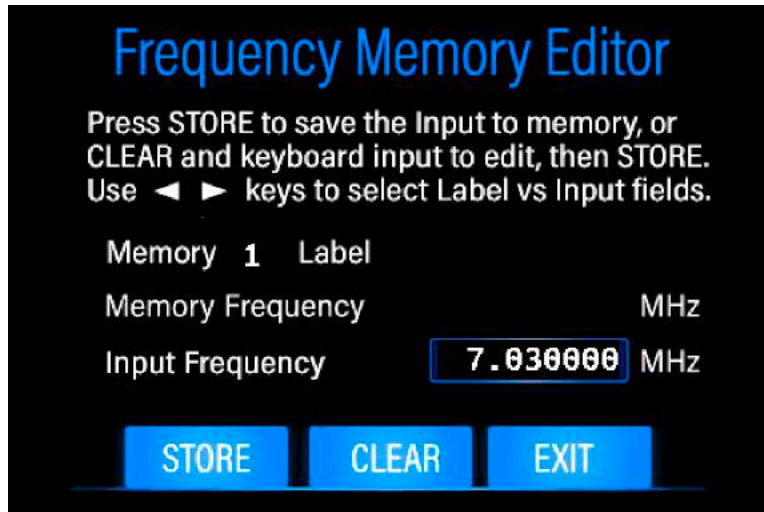
### 0-9 keys

These keys will load a frequency into the receiver from the selected frequency memory, memory number 0 to 9. If no frequency has been stored in that memory, nothing will happen.

### Shift-0 to Shift-9

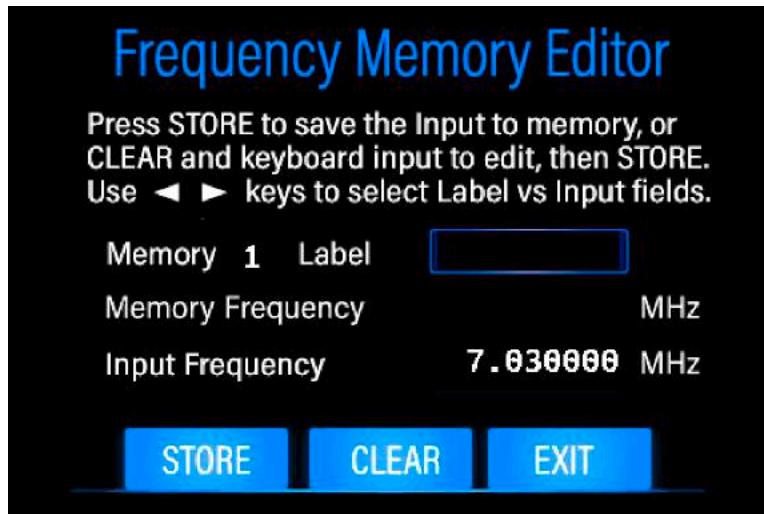
These key combinations are used to access the Frequency Memory Editor for the specified frequency memory. The Frequency Memory Editor can also be accessed by using the **Edit** Buttons on the Frequency Memory List (see Figure 12a). The Frequency Memory Editor allows you to save the current VFO frequency to memory, to clear and enter any frequency (within the bounds of the receiver), and to set a label on the specified frequency. This means you can label your favorite nets or special frequencies, such as WWV, for example. If a frequency has no label, the frequency will be displayed in the list. If there is both a frequency and a label defined, the label will be defined, as shown in Figure 12a.

The Frequency Memory Editor is shown on the next page in Figure 14a.



**FIGURE 14A: FREQUENCY MEMORY EDITOR**

Note the blue rectangle around the Current Frequency value: This indicates any edit action will apply to this value, such as pressing **CLEAR**, and typing in some value you wish to put into the memory. To get to the Memory Label field, press either the left or right arrow, and the blue box will move to that area, indicating you can edit that value. This is shown in Figure 14b below:



**FIGURE 14B: EDITING THE LABEL FIELD**

Type in a label value, up to 9 characters. The delete/backspace key can be used to correct errors. Finally, press the **STORE** Button to save the value, or **EXIT** to return without changing the value already in the memory. Note: If there was already a value in the memory, it would be shown in the Memory Frequency value area. Storing a new value will instantly change the value shown in Memory Frequency. Likewise, if you press **CLEAR** and then **STORE**, you will clear the memory.

---

## 8.8 Function keys

The function keys F1 to F12 and Shift-F1 to Shift-F12 are used to run user-defined microprograms for station control and automatic text output. This includes streamlined operation modes, such as for contests, or net operations. Refer to [Section 18 Appendix V](#) for information on microprogramming and the function keys.

---

## 8.9 Transmit Keyboard Controls

There are fewer control keys available during transmit mode, as most keys are required to send your messages. These control keys are shown in Table 3 below:

---

<b>ESC</b>	Abort Transmit Mode
~ (tilde)	Insert the INFO microprogram
F1 - F12	Insert User microprogram 1 - 12
Morse Character	Sent to Type-Ahead Buffer
<b>SET CALL WPM = ? or SET ANS WPM = ? (Status Line)</b>	
0-9	Enter speed from 5 to 60 WPM
Space	Accept entered speed
S	Accept and Save as default speed
<b>ANS WHO? (Status Line)</b>	
A-Z, 0-9	Enter call sign
Space	Set entered CCS and start ANS mode transmission

---

**TABLE 3: KEYBOARD CONTROLS IN TRANSMIT MODE**

### **ESC key**

Normally this key should not be used, except in an emergency, when you can't wait for the normal exit from CALL or ANSWER mode, or if your microprogram running amok. ESC will force exit from instant transmit mode. Note that a forced exit from transmit mode will not activate RIT tuning, and will terminate any running microprogram.

### **~ (Tilde) key**

This key inserts the INFO microprogram into your outgoing message. This is described in detail in [Section 12.7](#).

### **F1 - F12 key**

These keys are used to insert a microprogram into your outgoing message, in a similar way to the INFO microprogram. More information on microprogramming is provided in [Section 18 Appendix V](#).

### **SET CALL or ANS WPM**

These prompts will appear in the Status Line display when the encoder needs to be given a transmit speed. This is described in detail in [Section 11.3](#) and [Section 11.4](#). Normally, ANS speed is set by the station you are answering (that is, by the speed they were using), unless capped by your typing speed. CALL will use any predefined speed, but if none is set, it will request you set one. Pressing **SPACE** will set the default speed of 20 WPM. Note: if you have set a maximum transmit speed via the Typing Speed mechanism, then no matter what you enter, the speed will not exceed the maximum typing speed you have set. Nor will it respond to a fast station at that sender's speed, but will limit your speed to your comfortable typing speed.

### **ANS WHO?**

This prompt will appear in the Status Line display when a call sign is required when processing an **ANS** Button press. Refer to [Section 12](#) for more information. When you see this prompt, you must enter the station call sign you are answering.

## 9. Decoder Operation

The decoder built into the **DMX-40** is quite complex. It includes a very sophisticated and sensitive analog and digital filtering system that is capable of extracting CW from an extremely noisy environment with excellent results. We have seen it do this when you can barely tell there is a CW signal under the noise. The decoder is dynamically adaptable to changes in the incoming signal. Provisions are made to handle major changes in the incoming signal, such as when tuning to a completely different signal. This is done by pressing the **SPACE** bar, which restarts the decoder. The HCW decoder also has the ability to recognize when it is way off the speed, and automatically restart. However, for fastest results, always press the **SPACE** bar when tuning into a new signal.

The decoder operates in two modes: HCW (human-generated CW) and CTX (computer-generated CW). The mode you are in is controlled by the **ESC** key when in Receive Mode. The default mode is HCW when you power up.

---

### 9.1 HCW Mode

The advantage of the HCW decoder is its adaptability to speeds from 5 WPM to well over 40 WPM (it is capable of 60 WPM in the lab, but we have not extensively tested it on-air, and can't type that fast anyway), and its ability to handle sloppy code generated by humans. In HCW mode, the decoder will collect data for a short period of time before attempting to decode it. During this time, the Status Line will display:

#### **HCW: Collecting Data..**

When enough data is available, it will calculate and estimate the sender's speed in words per minute (WPM), using a complex algorithm looking at dit times, dah times, and each of three different space times (spaces within characters, space between characters, and space between words). These seven values and associated demarkation values are used to determine the sender's speed, and to decode the data collected, which is then quickly spit out to the LCD display. Thus, you do not actually lose any of the code that was received during the collection process - it is just delayed, and only displayed when the decoder has calculated the speed. This is a much better solution than showing gibberish until the decoder locks onto the sender's speed, which is typical of most other decoders on the market.

Once calculated, the determined speed in WPM is displayed. The status line will display something like this:

#### **DECODING HCW @ 18 WPM**

The displayed speed (18 WPM in this example) will be updated from time to time, as the decoder refines its database. Once the speed and statistics of the sender have been calculated, the decoder will do "real time" decoding. This means that once a character has been received along with a following inter-character or inter-word space, the character will appear on the LCD. If the station stops sending, the last character will appear approximately one second later. The decoder will continuously adjust to the sender dynamically within a reasonable range (changing from 20 WPM to 5 WPM would not be considered reasonable).

Note that once you see the speed of decoding has been set, your transmit speed has also been set. Your transmit speed will always be set to the speed of the other station, and will override any default speed you may have specified earlier. HOWEVER, if the speed is faster than your pre-defined maximum transmitter speed in the Typing Speed function, then this will be the upper limit, and if the sender is sending faster than that, you will be sending slower than he or she is. But, you will be able to keep ahead of the transmitter and not starve the transmitter, and be able to think clearly rather than being rushed and panicked.

When tuning to another signal, it is a good idea to reset and restart the HCW decoder. This is done simply by pressing the **SPACE bar**. This will return the status line to **HCW: Collecting Data...** until the new signal is analyzed. This is always useful when a noticeable different speed is heard. A single character may be missed due to the brief reset and restart time, but everything else will be decoded and displayed once the decoder has completed its data gathering and determined the sender's speed.

At one point, it was planned to automatically reset and restart the decoder when the receiver frequency was changed more than some set number of Hz. After some consideration, this feature was dropped. We believe it is better for the user to tune in a new signal, and then press the **SPACE bar**, rather than depending on a not necessarily well timed reset and restart that is automated.

Finally, the HCW decoder can also automatically reset and restart the decoder if it finds a series of T's and E's and spaces. This indicates the timing is way off, usually due to a new signal being received.

---

## 9.2 CTX Mode

CTX mode is very useful between two stations that have CTX capability. This can be discovered easily if each station uses the INFO microprogram, which includes the rig type. This will be discussed later in [Section 12](#).

The CTX decoder is set to 30 WPM, and while it can adjust slightly to that speed from band fluctuations, it is quite stable. The advantage of this is a substantially lower error

rate when listening to another CTX station using the same mode. The speed 30 was selected to make the type-ahead buffer useful. At 60 WPM, for example, few if any can type faster than the transmitter can send, and having a type-ahead buffer is a total waste. Transmitting at that speed is typically way above most operator's typing speed, and generates angst due to the inability to correct errors before they are transmitted. Of course, 30 WPM is also above many, so this mode is seldom used.

Another difference between HCW and CTX is that in CTX mode, you already know the speed, you already know the statistics of the sender, so you can instantly decode. There is no need to collect data. When the CTX decoder is active, the status line looks something like this:

### **DECODING CTX @ 30 WPM**

Thus, you can always tell your decoder mode from the status line.

---

## 9.3 Switching Modes

As described earlier, the **ESC** key allows you to switch between the HCW and CTX decoder modes. Normally you are going to be using HCW, but if you make a contact with another CTX user, you can both agree to switch to CTX for more accurate results.

Besides switching between modes, the **ESC** key does a few other important things. One is to force a reset and restart for the decoder. Another is to clear a user-entered transmit speed, but it only does this when changing from CTX mode to HCW mode. The purpose of this special function is described in the [Section 9.3](#).

---

## 9.4 Tuning a Station

To better understand how best to tune in a station, a diagram should help. Figure 15 on the next page is a simplified diagram of the main elements of the decoder signal processing chain.

1. Input filtering and gain feeds a somewhat reshaped version of the signal to the next stage
2. Hyper-Active Tone Digitizer converts the incoming reshaped signal into a very noisy digital signal
3. Digital Filter separates the noise from the signal, like your ear does. They have very different characteristics

- The output is fed to the HCW decoder pre-processor, which figures out the speed of the CW as well as 5 major parameters. This is followed by the actual CT to text conversion step.

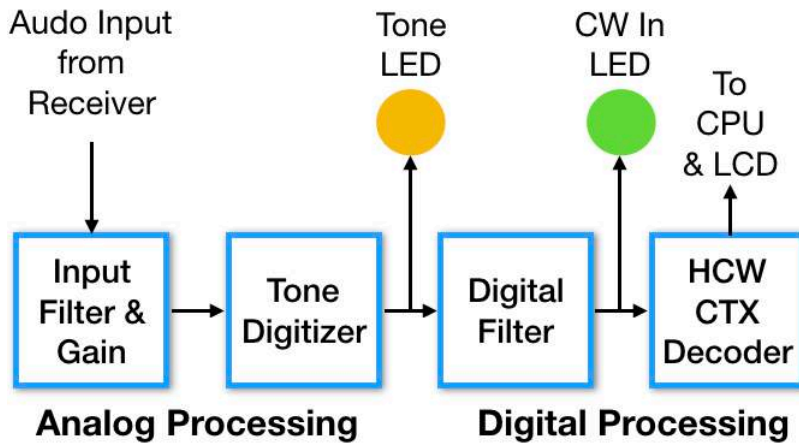


FIGURE 15: THE DMX DECODER BLOCK DIAGRAM

The Hyper-Active approach in the design creates a lot of noise on the **Tone LED**, causing significant flickering when a signal is being received. This is normal.

It is important to note that the system works around the center frequency of approximately 1300 Hz. This is selected to maximize the speed of the decoder, as well as reliability. This is the same frequency as the TONE OUT of the system, so you will get used to the frequency fairly quickly. This is about twice as high a frequency as most decoders use.

Note that the DMX-40 decoder system is extremely selective: it will only respond to signals that are within 50 to 60 Hz from its center frequency, around 1300 Hz. The frequency selectivity is higher at lower signal volumes. Thus, it is important for best reception to quickly center between the "start" and "end" frequencies where the current signal shows up on the **CW IN LED**. It is not critical that the centering be exact, but approximate. You just don't want to be decoding at the edge of the signal decode band, as this can cause more errors to occur, especially if the signal is fading in and out.

Tuning is quick and simple:

- Scan the band for an appropriate signal using the **left** and **right arrow** keys. A great way to do this is by setting the "step" size to 100 and hold down the **left** or **right arrow** key. Stop when you hear a signal.
- If the signal is weak, increase the decoder gain to 40. Otherwise, leave it at 30. Only reduce this if the signal is strong, and it is causing the **CW IN LED** to not go out between dits and dahs.

3. Quickly tune up/down with 10 Hz steps and center (roughly) using the **CW IN** LED as an indicator. This step is somewhat optional.
4. Press the **SPACE bar** to restart the decoder.

Within a few seconds, the decoded characters will be displayed, along with the senders speed. Of course if you are in CTX mode, characters will be displayed immediately. Step 4 can be eliminated in CTX mode.

While this may seem a little complex at first, you will quickly learn how to do this without thinking.

---

## 9.5 The Tuning Rule

You hear tones with CW because the receiver is inserting a frequency that is "beating" against the incoming intermittent carrier wave of the signal you are decoding. Technically, the audio comes from the A+B and A-B rule for mixing signals. This, if you mix a 7 MHz signal with a 7.001,300 MHz signal, you will get the following:

1.  $A+B = 14.001,300$  MHz
2.  $A-B = 1,300$  Hz

It is the difference frequency, A-B, that we are interested in. Note that if A-B does not give a real frequency (such as -1300), then B-A will work just fine. It turns out that -1300 Hz is turned into 1300 Hz anyway (I used to know how that happened, but then, college electrical engineering was a LONG LONG TIME ago....

Anyway, what this means is that if you are 1300 Hz below OR 1300 Hz above a carrier wave or CW, you will hear 1300 Hz tones. This can make things difficult. If you don't know which one you are on, how can you be sure to answer a station? If you are wrong, you will be 2600 Hz off, and he/she will not hear you.

For the **DMX-40**, the frequency you see in the **Frequency** Button is the transmitter frequency, NOT the receiver frequency. The receiver is tuned 1300 Hz lower, so any station transmitting on the displayed frequency will be heard by you and the decoder as a 1300 Hz tone. Unfortunately, the signal could also be 2600 Hz lower than the displayed frequency.

This is most easily explained using diagrams, which are shown in Section [21 Appendix VIII](#). However, there is a simple way to deal with this problem. We call it "The Tuning Rule." This rule is described in the next section.

## 9.6 The Tuning Rule

The tuning rule is based on a two simple observations, as explained below:

1. If you made the call and someone answered, skip this. You are automatically on the same frequency as the other station.
2. If you answered a call and it validated (see [Section 12.2](#) on the Answer-Validate step), test the tone with a small change in frequency. If increasing your frequency one step makes the received tone go DOWN in frequency, skip this. You are already on the correct frequency.
3. If the tone went up, Press the **D** key to drop down 2600 Hz, and adjust your frequency slightly to receive the other station accurately (i.e., make sure you are within the decoding frequency bounds, and restart the decoder).

Experiment with this process by tuning in stations, then applying the Tuning Rule.

---

## 9.7 Switching Bands (Modes)

The **DMX-40** contains a transceiver board that operates on a single frequency. Future versions of the product may include dual or even try-band transceiver boards. For single-band units, such as the DMX-40, you have the option of using it on the selected band or turning off the Tx/Rx board. Pressing the **Frequency/Band** button on the display will switch you between the current band, and **External** mode. Pressing again returns to the band, but at the starting frequency (7.03 MHz for 40 Meters).

**External** mode is simply the mode where the transceiver board and its VFO are ignored (VFO actually turned off). This is the best mode to use when playing with the CALL and ANS functions, until you are confident in their operation. Alternately, a dummy load can be attached to the antenna connector, which allows you to experiment with CALL and ANS including using the frequency controls and RIT tuning.

For future DMX products that support multiple bands, pressing the **Frequency/Band** Button on the LCD screen will switch between the supported bands, from lowest frequency to highest, and **External**. This is the mechanism for band selection when a multi-band product is in use. The transceiver board will be notified as to which band

you have selected, so that its circuitry can adjust appropriately. Note that each band has 10 frequency memories - you don't have to "share" a measly 10 memories with all of the supported bands.

A quick way to return to 7.03 MHz, therefore, is a quick double-tap on the **Frequency Button**.

## 10. Tone Output Control and KEY IN for Training

Before we get into transmitter functions, getting familiar with the Tone Output and Key Input function is important.

Your DMX-40 can generate a 1300 Hz tone for both computer generated output as well as if you utilized a Morse Code mechanical key plugged into the **KEY IN/OUT** Jack. Refer to [Section 23 Appendix X](#) For information on wiring a mechanical key to this jack.

The tone generator is turned on and off by either a keyboard letter K or by holding down the Morse Key for 3 seconds. Whatever state you leave it in will be saved in the memory for the next time the DMX is started up.

Tone Output is at a fixed volume which cannot be changed. It uses the same 1300 Hz frequency desired by the decoder, so it gets you used to looking for the correct frequency.

With the Tone Output turned ON, you can listen to both incoming code from the receiver, but also the outgoing code from the transmitter. The transmitter code will include some level of clicking, due to high current flow nearby.

If you have a Morse Key plugged in, you can use it in both transmit and receive modes:

1. **Transmit Mode:** Obviously, the key allows you to send text via key rather than by keyboard. Note that the key does not operate if the transmitter is busy, such as when it is sending the initial and final station hand-offs. Of course, using microprograms, you can eliminate those if you want total control of transmit operation.
2. **Receive Mode: WORLD CLASS CODE TRAINING!** Yet another unique feature, your DMX-40 is not merely a "code practice oscillator" but a very smart Morse Code decoder! This means it knows the rules of the road, and how far outside the rules you can swim and still be understood by most listeners. Thus, it provides world-class training for correct Morse Code sending! Once you have sent enough to establish your speed, you will get immediate feedback on your characters and spacing. Spacing correctly is very important for easy to "read" code. Many coders run words together, or send words as separate characters, confusing the listener, who may be already stressing out trying to understand your message. Give your listeners a break - learn to transmit clean code using our code trainer!

To establish your speed, send CQ or even just C over and over again until it spits them out and shows your speed. As long as you continue at the same speed, you now have real-time decoding.

## 11. CALLING

There are four primary ways to establish a connection to communicate via radio. These four methods are as follows:

1. **General Calling:** calling for anyone to answer, known as "calling CQ"
2. **Directed Calling:** calling for a specific station to answer, usually on a schedule
3. **Validate-Answer:** Validating Two-Way communication connection when answering a call
4. **Answering a Directed Call:** talking back and forth with another station, known as having a QSO or conversation. This is also known as "handing it off" or "handing it back" to the other station.

The first two are covered in this section. Validate-Answer and Answering a call is discussed in [Section 12](#).

---

### 11.1 Before We Begin...

**IMPORTANT!** Please press the Frequency/Band button on the display until you see **External** displayed. This will ensure your transmitter is turned off. If you wish to use the transceiver to experiment with the CALL function, please connect a dummy load to the antenna connector first. Once you are confident in the CALL and ANS functions, you can proceed to use it with an antenna.

**WARNING: Operating the transmitter without a properly installed antenna or dummy load may damage your transmitter, and will void your warranty.**

---

### 11.2 General Calling

In amateur radio terminology, this is known as "Calling CQ." CQ is a term that originally meant "change of quarters" but is now simply a code used when doing a general call on a radio. This term is widely used in amateur radio by morse code as well as SSB and FM voice users.

Your transceiver has a built-in general calling microprogram. Here is the format, where [YOU] is your call sign:

**CQ CQ CQ CQ DE [YOU] [YOU] K**

Note: if you have not entered your call sign yet, you will get a warning message to do local station setup. Here is what the sequence looks like with W1AW as the call sign:

**CQ CQ CQ CQ DE W1AW W1AW K**

This is a single repeat of the general calling sequence. However, the actual calling sequence repeats automatically (without the ending K) until you tell it to stop. Once you tell it to stop, it completes whatever repeat cycle it is on, adds in the K and exits the call mode. Instructions on how to stop the repeating CALL are described in [Section 11.5](#) below (essentially, pressing the **CALL** button a second time). FYI, the "DE" stands for "from" - so General Call from W1AW.

The K is a character that tells a listener that you are looking for anyone to come back or respond to your general call (K = Come Back to Me). It will be automatically appended at the end of the last repeat of the calling sequence.

As you can see, entering your information in **Local Station Setup** has now come in handy. The transceiver computer knows what your call letters are, so it automatically inserts them into the calling sequence for you. Thus, to call, no typing is required.

---

## 11.3 Setting Transmit Speed

Before we go any farther, we need to address the issue of transmit speed. This will come up for various reasons because the encoder has no idea what speed you want it to send at. Here are the cases where you will need to supply a transmit speed:

1. The decoder has not set a decode speed, or it has been reset and restarted
2. No transmit speed has yet been set by you (typical first time after power on)
3. You are not in CTX mode, where the speed is always 30 WPM

Note here that if you are listening to another station, once the decoder sets a speed, then the transceiver computer knows and displays the correct sending speed: the decoder speed for that particular sender. Of course the speed may be limited to your typing speed (see [Section 7.3](#)). If you are in CTX mode, it always knows the transmit speed: 30 WPM. But what if you are in HCW mode, and either you have not been receiving anyone, or you have cleared the speed by pressing **ESC** twice. In this case, what speed should be used to CALL?

Naturally, you will probably want to use a preferred speed. But **when** do you need to define the speed? Never fear, the transceiver computer knows when it needs a speed and does not have one. If you start a CALL and no speed is known, you will be asked for a speed on the Status Line.

## SET CALL WPM = ?

Simply enter any number from 5 to 60 (5 to 60 WPM). Best rates are under 40 - you will need to experiment with faster rates to see if they will work well for you and whoever you are talking with - if you can type that fast! Note: if you have already set your preferred (comfortable) typing speed, if you enter a higher number here, it will be capped at your typing speed. Thus, your typing speed cannot be exceeded, unless you change it. After typing in a speed, press **SPACE** to accept the speed you entered, and the CALL sequence will automatically save that speed as the current speed, and execute the CALL, as described. The **delete** or **backspace** keys can be used to correct any mistakes, and the **SPACE bar** will not do anything if the value entered is not from 5 to 60.

Alternatively, you can accept the default transmit speed set at the factory. The factory default speed is 20 WPM. To enter this speed, simply press the **SPACE bar** rather than entering a speed followed by the **SPACE bar**. 20 WPM will then be saved as the current speed, unless your typing speed is lower. You will not need to re-enter this speed unless you clear it by pressing ESC twice (remember, returning to HCW mode from CTX mode clears the transmit speed).

Keep in mind that the decoder speed will ALWAYS override the transmit speed if it is defined. In other words, if you have set the transmit speed to 15 WPM, and someone answers you at 18 WPM, when you answer back, it will be at 18 WPM, again, assuming your typing speed is higher than 18 WPM. If for whatever reason the decoder is reset, you will revert to 15 WPM until and unless the decoder establishes the sender speed again.

Note that the decoder speed is cleared by resetting the decoder, but that does not clear any transmit speed you have set. This is why when you reset the decode in the example above the speed drops to 15 WPM: the decoder speed was cleared, but you have set 15 WPM as the current transmit speed. When encoding, the speed from the decoder is first checked. If it is not set, the transmit speed is checked. If it is set, it uses it. If not, you get the request for a speed.

Sound complicated? No, it's not. Just use it for a while and it will become clear and easy.

---

### 11.4 Setting a New Default Transmit Speed

There is a provision for you to change the default factory speed of 20 WPM to your preferred speed. You do this by pressing the **S** key rather than the **SPACE bar** after entering a speed on request. This will store your speed, which will then replace the factory default. Now, every time a transmit speed is requested, press the **SPACE bar**

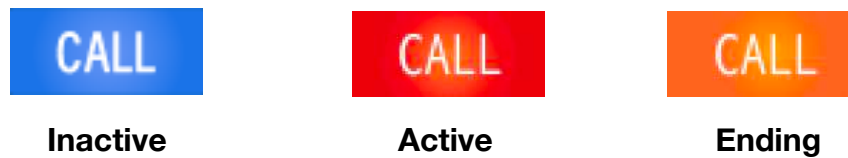
to set your default speed, which will now be whatever speed you selected when you pressed **S**. Of course **S** will not work unless the speed entered is from 5 to 60. The **backspace** or **delete** keys can be used to correct entry errors, It's a great idea to set the default speed to your typing speed, unless you want to talk with some slow operators.

For more on setting a new default transmit speed, refer to [Section 11.4](#) below.

---

## 11.5 The CALL Sequence

How do you activate the calling sequence? Simply press the **CALL** Button. There are actually 3 states for the **CALL** Button, as shown in Figure 16:



**FIGURE 16: CALL BUTTON STATES**

The first **CALL** Button image on the left represents the inactive **CALL** Button state, waiting for you to press it and activate the CALL sequence (and Transmit Mode). Touching the **CALL** Button brings up the second **CALL** Button state indicates that the CALL sequence is actively repeating the CALL microprogram over and over, i.e., that you are in transmit mode (**RED**) and the computer has responded to you pressing the **CALL** Button. As soon as you press the **CALL** Button a second time, it will change to the third **CALL** Button state, indicating that the CALL Ending sequence has been activated. This third state is **AMBER**, indicating the CALL sequence is not complete, and Transmit Mode is still active.

Once the transceiver computer has completed the current CALL sequence, and appended the **K**, the **CALL** Button will return to the first inactive **CALL** Button state, and the Receive Mode will be restored. However, you will find yourself in **RIT Mode** (Receiver Incremental Tuning Mode) at that point. More on that in [Section 13](#).

---

## 11.6 Setting Up a Directed Call

As mentioned earlier, a **Directed Call** is a call to a specific station. For example, say you decided to meet another station for a chat at 11:00 AM at 7.115 MHz. Typically, you have an agreement for one of you to initiate the call. If it is you, press the NO CCS

Button and enter your friend's call sign. You will see the screen shown below in Figure 17 below:



**FIGURE 17: CAPTURING A CALL SIGN MANUALLY**

Enter your friend's call sign, and press **FINISH**, or the **SPACE** bar. The NO CCS Button will change to show your friend's call sign.

In our example, At 11:00 AM, you would tune to 7.115 MHz and start calling using the **CALL** Button. Since you have entered a call sign for another station, the **DMX-40** will execute a directed call, such as shown below, where W1AW represents their call letters, and K2AAI represents your call letters:

**W1AW W1AW W1AW W1AW DE K2AAI K2AAI KN**

Like the general call, this is a repeating sequence, where **KN** is appended at the end. **KN** is used to indicate you want only the called station to come back to you. The **CALL** Button works exactly the same way, so what is different? Simple. In the General Call example, the computer does not know a station to call. For a Directed Call it does. Of course if you don't get a response after 5 minutes of calling, pausing every 30 seconds or so to listen, that could be the band, or your friend is not available.

---

## 11.7 Three Ways For Establishing a CCS

A CCS or Captured Call Sign is a call sign that has been automatically captured by the decoder by recognizing a standard format of letters in a station identification sequence, or has been captured from the keyboard manually, entered by you, as shown above. This feature of the decoder is the primary source of acquiring another station's call sign when you are listening to stations, and find one you want to call or answer. This

capture is automatic, and overrides any previous call sign that might have been entered by you or captured previously by the decoder. Any call sign captured or entered by you is displayed in the **CCS** Button. If there is no defined CCS, the button will display **NO CCS**.

Since a Directed Call requires TWO different sets of call letters - yours and someone else's - the place the second call letters are found is in the CCS memory. However, it is quite possible that no CCS has yet been captured, or you cleared it to make a general call, and now you want to make a Directed Call.

There are 3 ways a CCS can be established:

1. **By the decoder**, when it sees the required series of characters during a station identification
2. **By the user**, via the **CCS** Button on the Main Screen.
3. **By the user**, when prompted when using the ANSwer function (more on that in [Section 12](#)).

1) Decoder CCS Capture: The decoder requires a specific format of received characters to capture a call sign, as shown below, where the underscore character "\_" is used to show where spaces are required:

### \_DE\_CALLSIGN\_

Three spaces, with DE and a call sign in between are required. The call sign must not have any spaces within it, and must be at least 3 characters and no more than 9 characters in length. Thus, " DE W1AW " would generate W1AW as the CCS. But "DEW1 AW " would not. Captured call signs appear automatically in the **CCS** Button, and replaces the **NO CCS** text or any previous CCS. Note that no other validity checks are made on the call sign other than no spaces within and a length of 3 to 9. It is up to you to ensure any CCS you use is a real call sign. Thus, call signs can include the "/" character, for example.

Note that some operators do not use the standard format shown above. In those cases you will have to enter their call signs manually.

2) To manually enter a call sign - the CCS - simply press the **CCS** Button, when it is displaying **NO CCS** or when you want to manually enter a new call signs shown above in [Section 11.6](#).

If you decide you don't want the CCS set, you can clear the CCS by pressing the **CCS** Button when it displays a defined call sign. This will force it back to the **NO CCS** state.

Note that the CALL function will never request a call sign. Rather, if none is defined, it will simply default to a General Call. Thus, manually entering a CCS is required for a Directed Call.

---

## 11.8 What Happens After You CALL?

There is a very good reason for the repeating nature of the CALL function. And that is, it often takes time for another station to happen by your frequency and hear you calling. Depending on your style, the congestion on the band, and the state of the ionosphere in bouncing your signal around the earth, you may want to call only a single repeat (press **CALL**, then press it again), or you may want to do several repeats each time. If you get no response, more repeats may be required. Don't be alarmed if you get no response, especially if you hear little activity on the band. There are days when the band is as dead as a rock: it all depends on solar activity (sunspots), and time of day. For 40 meters, the band usually opens in the afternoon and early evening.

After the CALL sequence has ended, listen for a minute or two for an answer. The answer may be off frequency enough that the **CW IN** LED does not light (but the **STONE** LED does). The receiver will automatically be in **RIT Mode**, with 1 Hz steps. Press the up arrow to switch to 10 Hz steps. Adjust the frequency slightly until you see the **CW IN** LED blinking. However, by using the Auto Repeat key function of your keyboard, 10 Hz or 1 Hz steps may be good enough. More on **RIT Mode** is explained in [Section 13](#).

Be patient. Sometimes the band condition is not good for your location, and no one answers you. Other times, you might get several people to answer. That is where the fun starts: Unfortunately neither you nor the **DMX** computer can decode two CW signals right on top of each other. Try to extract one call sign to answer by careful adjustment of the RIT offset.

If you only get a part of a call sign, and it is not automatically detected because of the dual competing signals, manually enter the piece you did get, such as W1?? for example. When you answer using W1?? the W1 station will come back, the other will not, and you will get the correct call sign, typically automatically loaded by the decoder, for your next transmission.

## 12. ANSWERING

Before you can answer, you have either first called via a General or Directed Call and got a response, or you have heard another station either doing a general call or a directed call for you.

There are two types of answering. One is called "validate-answer" and the other is just a normal answer. What is the difference?

**Validate-Answer:** the purpose of this function is to establish if a two-way conversation is possible and practical. Thus, you are validating that the other station can hear you. Naturally, this is not required if they have answered your general or directed call, as that implies that yes, they can hear you. It does apply when you are answering someone else's general or directed call to you. You certainly can hear them, but can they hear you? Important to find out before you start sharing your life!

**Answer:** the answer function is simply the mechanism for responding to the other station's most recent transmission, allowing you to respond to any questions, send them information, or ask them questions.

There are two things needed for the ANSWER function, just as there was for the Directed Call: a transmit speed and another station's call letters.

If a speed has already been established, either by you setting it as described in the CALLing section, by:

1. entering s speed when asked, or
2. using the space bar to use the default speed, or
3. the decoder has set the speed, or
4. you are in CTX mode

then no additional action will be required. However, if the transmit speed has not yet been set, or has been cleared with a double ESC, and there is no decoder speed set, you will be asked to specify a speed. This works exactly the same as described in [Section 11.3](#) above: enter a value from 5 to 60 followed by pressing **SPACE**, or simply press **SPACE** for the default speed.

If the other station's call letters have been set, either by:

1. manual entry by touching the **CCS** Button and typing it, or
2. the decoder recovering a call sign from the decoded stream

then no additional action will be required. However, if there is no defined CCS, you will be asked to enter one on the Status Line:

## ANSWER WHO?

Type in the call sign, followed by the **SPACE bar**. The **backspace** or **delete** key can be used to correct mistakes. Your entry will immediately show up to replace the **NO CCS** label on the **CCS** Button.

If you see a call sign in the **CCS** Button, but it is NOT the one you want, press the **CCS** Button to clear it before pressing the **ANS** Button.

Once the speed and call letters have been defined, the ANSwer sequence will immediately begin.

---

### 12.1 The ANSwer Validate and ANSwer Microprogram

Like the **CALL** Button, the **ANS** Button has three states, as shown in Figure 18:



**FIGURE 18: ANSWER BUTTON STATES**

As with the **CALL** Button, the 3 states are Inactive, Active, and Ending. However, there are many differences between CALL and ANSwer modes. The microprogram that is executed can be thought of as three separate components

1. The **Validate-ANSwer Microprogram Component**: This is a short microprogram to test to see if the other station can hear you. It is activated by pressing the ANS Button twice, quickly. By quickly, we mean before the microprogram completes the sending of the CCS, as we will describe in more detail below. This can be referred to as either the ANSwer Validate or Validate-Answer, whichever you prefer.
2. the **ANSwer Start Component**, which handles the station identification step for beginning the transmission.
3. The **ANSwer End Component**, which handles the station identification step for handing the conversation back to the other station.

We refer to these three as components, because they are actually all part of a single, fairly complex microprogram. We will address these differences below.

## 12.2 The ANSwer Validate Component

This component of the microprogram should be executed when you have tuned into a station calling CQ, and you want to find out if they can hear you, since you already know you can hear them. It is not uncommon to have a situation where you can hear a station very clearly and even loudly, but they cannot hear you. This can be due to differences in the amount of transmitted power or antenna configuration, as well as the ionosphere status at your two locations. For example, they may be in a location after dark, and you might still be in daytime. They have a much better ionosphere for long bounce than you do. Thus, before you start attempting to talk to them, first validate that two-way communication is possible using the Answer-Validate function.

This component of the ANS microprogram is activated by pressing the **ANS** Button twice. The second press must be completed at the latest prior to the completion of the transmission of the first CCS and the following space. This typically is a few seconds, so it is fairly easy to accomplish. If you have pressed the **ANS** Button prior to that, the status line will show:

### **ANSWER-VALIDATE**

Note that the ANS function ALWAYS starts with sending the CCS call sign at least once. The resulting Answer-Validate transmission looks like this:

**[THEM] [THEM] DE [YOU] [YOU] KN**

Using W1AW for the other station, and W2OL for you, it looks like this:

**W1AW W1AW DE W2OL W2OL KN**

This sequence simply calls the station, and provides your call letters. The KN also says you are specifically calling one station to respond. If that station hears you and responds back, you have established that you have a two-way channel, and can proceed with a two-way conversation.

---

## 12.3 The ANSwer Start Component

During a two-way communication, pressing the **ANS** Button only once causes the ANSwer Start component to begin. The Start portion of the ANS microprogram is transmitted. This is the normal station identification sequence the is normally sent at the beginning of a transmission, and is automated by the **DMX** computer for your convenience. This component transmission is as follows;

## [THEM] DE [YOU]

Using W1AW for the other station, and W2OL for you, it looks like this:

### W1AW DE W2OL

Once this component has completed, nothing happens until you either press the **ANS** Button again or you type a message to the other station on the keyboard. Note: A space is automatically included at the end of the ANS Start component, so you don't need to remember to include one at the beginning of your message.

---

## 12.4 The Pop-Up Type-Ahead Buffer

As soon as you press the **ANS** Button, a two-line type-ahead buffer pop-up appears at the top of the screen, as shown in Figure 19a:



FIGURE 19A: POP-UP TYPE-AHEAD BUFFER

Figure 19a shows a point in time where the Start component has been sent, but the user has not typed anything. At this point, nothing is being transmitted, and nothing is happening. The user either must type a message, or press the **ANS** Button to end the transmission. We only show a single calling sequence by W1AW calling W2OL (ah, should it be that easy!), and where the local operator (you) is W2OL, and you, the user, have correctly pressed the **ANS** button, and have gone to sleep on the switch. A much better scenario is shown in Figure 19b, where you, the operator, has started typing as soon as you have pressed the **ANS** Button, so you are way ahead of the transmitter. In Figure 19b, we have you typing at a blinding speed, probably faster than you can actually type... but this is for the purpose of showing you a different state in the answering process.

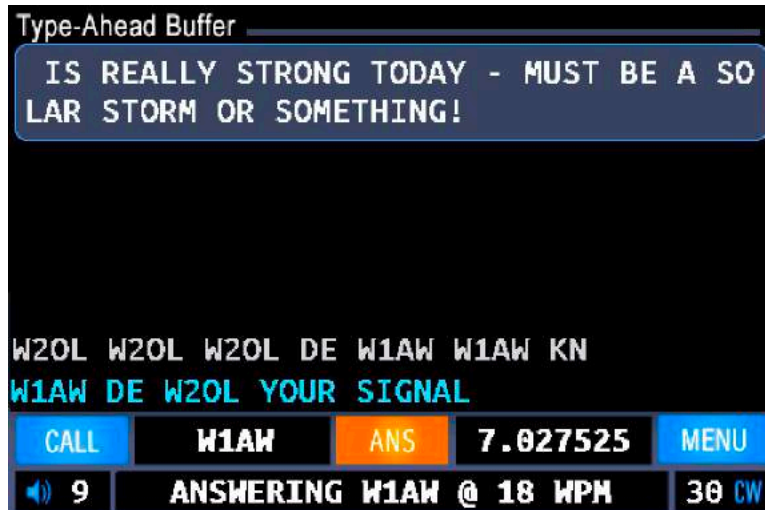


FIGURE 19B: TYPING DONE, TRANSMITTING AND THEN ENDING

Specifically, an outgoing message has been typed, the transmitter has already transmitted the first two words "YOUR SIGNAL" and is working on emptying the buffer one character at a time. The **ANS** Button has been pressed the second time, as indicated by the **AMBER** color. This tells the **DMX** computer that you have finished typing, so when it empties the buffer, it can transition immediately into the ending component can be transmitted. Also note that you always know what is going on by reading the Status Line, here indicating that you are answering W1AW at 18 WPM.

NOTE: It is easy to forget to press the **ANS** Button the second time after finishing typing. This is awkward - once you wake up and press it, you have left a silence period that may make the other station think you have handed it back to them without a station ID. Consider pressing **ESC** to get out of transmit mode in this case if the delay is too long.

The pop-up buffer will disappear as soon as the last character (a ! Character in this example) is read out of the buffer and its transmission begins. There are 76 characters in the type-ahead buffer. Naturally, you cannot type anything if it is full, so wait until there is some space to continue typing. Finally, the End component transmits the handoff station ID to the other station.

Of course you can change the standard operation of the **DMX-40** by writing your own microprograms to replace the built-in ones.

This approach to texting over the air is very simple and easy to understand, and also easy to read!

## 12.5 Sending Your Message

Depending on the transmit speed you are working at, there is actually quite a bit of time between when the transmitter starts sending the first character in the Start component and when the Start component completes. This is your time! Don't waste it! The keyboard is active, and you can start typing your message immediately. There is great advantage in doing so, as you now have the opportunity to get ahead of the transmitter so you can see errors in your typing and correct them before the transmitter sends them out. If you have properly set your typing speed (see [Section 7.3](#)), then you will be able to stay ahead of the transmitter, and the code output will be perfectly formed, i.e., there will be no extra space between characters, causing confusion on the other end.

### Thoughts on Default Transmit Speed

**NOTE:** It is a good practice to set your default speed to one that is aligned with your typing speed. Once you determine your typing speed using the built-in typing speed system, you can then enter it as the default using the **S** key when asked for a speed. Just press **ESC** twice to clear the transmitter speed, and press **CALL**. Type in your typing speed, press **S** to save it as the new default, then **ESC** to terminate the CALL.

---

## 12.6 The ANSwer Ending Component

Once your message is completely typed, press the **ANS** Button again. It's graphic state will change to the Ending state (**AMBER** button). This tells the computer that you are done typing. Once it completes sending what you typed, it will automatically append the ANSwer Ending Component. The Ending Component also includes a space at the beginning, so you don't need to remember to add a space at the end of your message. The Ending component looks like this, where [THEM] is their call sign, and [YOU] is your call sign:

**[THEM] DE [YOU] KN**

Once this sequence completes, the **ANS** Button returns to its normal inactive **BLUE** state. Note that you can tell instantly if you are in transmit mode by checking the screen: if either **ANS** or **CALL** Buttons are **RED** or **AMBER**, you are in transmit mode. Of course the transmitter is only actually transmitting when the **CW Out** LED is lit.

The ANSwer microprogram ends with KN, indicating that you only want the station you are talking to to "come back" to you.

Once you are returned to receive mode, you will automatically enter **RIT** (Receiver Incremental Tuning) Mode. This is described in [Section 13](#).

---

## 12.7 The INFO Microprogram

A common practice in amateur radio is to share basic information with the other station. This information typically includes you handle or name, your QTH or location, your rig, and your antenna. Rather than having to retype this information over and over and over, it has been automated for you by the built-in **INFO** microprogram. This microprogram can be typed at any point during the message part of your answer session. Simply type a **tilde** (~) character. Note that this is a shifted character - press and hold the **shift** key, and then the ` key. This will be replaced by the following microprogram:

**" NAME IS [NAME], QTH IS [QTH], USING [RIG] WITH [ANT]. "**

Of course the items in brackets are retrieved from memory, from what you entered during Local Station Setup. Note that this is a conceptual representation - the actual code is simpler (You can learn more about how to create your own microprograms in [Section 18 Appendix V](#)):

**" NAME IS ^1, QTH IS ^2, USING ^3 with ^4. "**

Note that there are spaces at the beginning and end, to avoid run-on with your text. Just type stuff, press the tilde key, and type some more. No additional spaces are required around the **tilde**. The tilde character will show up in the Type-Ahead Buffer, but when it disappears to the left, the encoder will execute the **INFO** microprogram before "eating" any more of your typed characters. This is another opportunity to type additional message text, while the encoder is busy with the **INFO** microprogram. Said another way, this microprogram only takes up a single character in the type ahead buffer, even though it actually results in lots of transmitted characters. The actual text sent might look like this:

**NAME IS JOE, QTH IS CHICAGO, IL, USING DMX-40 WITH 40M DIPOLE**

---

## 12.8 What Happens After an ANSWer?

The other station will (hopefully) begin transmitting when you are done. Of course that is assuming you did not fade out completely on their end. You might hear them trying to call you in that case. And, be prepared to adjust your **RIT** offset frequency, as they may have adjusted their transmit frequency when you were transmitting. More on RIT Mode is described in the next section.

[Return to Table of Contents](#)

After a few contacts with other stations, this will become easier and easier to manage. Keep in mind that different stations have different ways of operating, some of which may be very confusing. There are also many abbreviations used, which at first can be daunting. Finally, there are "pro-signs" that advanced operators use. The decoder knows most or all of these, and will display a useful indicator for you when they are decoded. A list of such pro-signs is provided in Section [Section 14 Appendix I](#).

## 13. Receiving Using RIT Tuning

At various places in this manual, we have described receiver operation. The RIT function was briefly described in [Section 8.6](#). However, to make it easy to find, this section covers the use of the **RIT** function in more detail.

The purpose of RIT is to maintain a stable frequency for your QSO (conversation). When a station answers a CQ, they are typically not on the exact same frequency as the calling station. This could cause the calling station to adjust their frequency to "match" the other station's frequency, but then the other station does likewise, and they drift together up or down the band. This is due to inaccuracies of their VFO's and what frequency they like to "hear" you at. This problem of the early days of ham radio has been eliminated with the RIT function.

By allowing you to adjust the receiver independently via an offset, this problem no longer occurs. If you don't move your transmit frequency, the other station will probably also not move their transmit frequency, making for a much easier QSO.

Essentially, you may have to adjust the receiver offset once at the beginning, assuming the other station is stable. After that, you should no longer need to adjust, and can even touch the **Frequency** Button to display the current transmitter frequency. However, every time you transmit, **RIT** Mode will re-assert.

---

### 13.1 Activating RIT

Essentially, the RIT mode is activated automatically after any transmit session - either CALL or ANSWER. You can "tune around" nearby looking for a response using the standard frequency controls (**left/right/up/down arrows**). The **Frequency** Button then looks like this:

**RIT ±N**

where **N** is the offset in Hz. A plus sign means you are offset above the transmitter frequency, and a minus sign indicates you are below the transmit frequency. For example:

**RIT +20**

means you are 20 Hz above the transmitter frequency. In Figure 20 below, a simple example of a completed CQ from you, W2OL, is shown, along with what the **Frequency** Button displays. Note that rather than showing the frequency you are transmitting on, the **Frequency** Button now shows you the offset from the transmitter

frequency. Initially, this value is +0, which means you are on the same frequency as your transmitter was.



FIGURE 20: RIT MODE EXAMPLE

If you get an answer to your CQ, it will probably not be exactly on your frequency, due to VFO errors and other factors. Press the Up Arrow to set 10 Hz steps and adjust for best decoder performance, i.e., the tone is around 1300 Hz and the **CW IN** LED is happy. Adjustment may not be required. Figure 21 below shows a possible adjustment of -30 Hz.

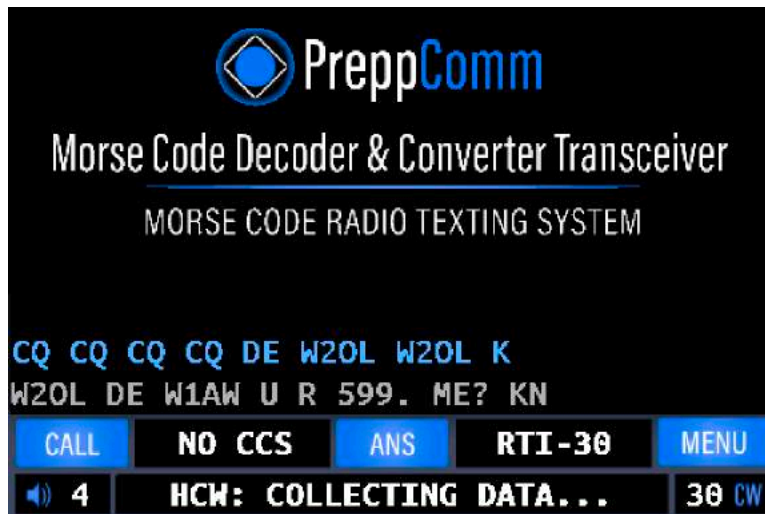


FIGURE 21: ADJUSTED RIT AFTER A RESPONSE

Figure 21 shows a response coming back from W1AW to your call with a signal report and a request for a signal report. During their transmission, the local receiver frequency was adjusted -30 Hz for best decoding. This means you are tuning to 30 Hz below for best decoder operation. Figure 21 is a great example as well of how the color coding of receiver vs. transmitter makes it very easy to see what is happening.

Note that when RIT is active, the **U** and **D** key functions are disabled.

---

## 13.2 Deactivating RIT

When you decide you are done with that station and that frequency, you can turn off RIT mode using a control key on the keyboard, or by touching the **Frequency** Button. There are two different ways you can end RIT mode:

1. Turn RIT off, without changing the transmitter frequency. This WILL change your receiver frequency to the transmitter frequency, unless the **Frequency** Button is showing RIT +0. This function is activated by pressing the **T** key (T for Transmitter Frequency) OR by touching the **Frequency** Button when RIT is displayed.
2. Turn RIT off, without changing the receiver frequency. This WILL change your transmitter frequency to the receiver frequency, unless the Frequency/Band button is showing RIT +0. This function is activated by pressing the **R** key (R for Receiver Frequency).

Of course these two control keys - **R** and **T** - are only used as control keys in Receive Mode. Also, when you enter **RIT** Mode, you may have set the step size to a value higher than the starting value of 1. When RIT is activated, the step size is reset to 1. Also, when clearing RIT mode, the step size is again reset to 1. This is done to preserve the frequency as it is currently set, rather than lopping off several digits because of the step size being set to a higher value.

## 14. Appendix I: Pro-Signs and No-Signs

The table below shows the pro-signs that the **DMX-40** recognizes, their meaning, and what is displayed on the screen if they are decoded. Note that there is no provision for sending Pro-Signs via the keyboard.

Code	As in (run together)	Meaning	Displayed
.-.-.	AR	End of Message	[EM]
.-...	AS	Wait, Stand By	[SB]
-...-	BK	Break	[BK]
-...-	BT	New Paragraph	[P]
-...---	CL	Clear (going off the air)	[CL]
-...-	CT	Start Copying	[SC]
-...---	DO	Change to Wabum	[WB]
-...-	KN	Invite specific station	[KN]
...-.-	SK	End of Transmission	[ET]
...-.	SN	Understood	[OK]

**TABLE 2: PRO-SIGNS RECOGNIZED BY THE DMX DECODER**

### No-Signs [?]

If a series of dit's and dah's are decoded that cannot be recognized, the [?] symbol is displayed on the screen. We refer to this as a "No-Sign." This can be caused by band noise, fading on the band, or bad spacing by the sender.

## 15. Appendix II: Special Characters

There were a number of strange issues we had to deal with when building the encoder/decoder system. Here are a few:

### Equal Sign and New Paragraph

The code for the equal sign = and New Paragraph is the same: -...- So how do you resolve this? We accidentally discovered this when listening to one station that was using a LOT of equal signs. But they did not make sense.

Naturally, I assumed the decoder was not working right. After much analysis of the output from my logic analyzer, it was clear that it was working just fine. That is when I discovered that there was a single code for two things!

I have never actually seen an equal sign used as an equal sign in a CW session, so I decided to accept the code -....- as new paragraph, which is displayed as **[P]**. I then assigned a new set of dit's and dah's to the = character. Note that we did not start a new line when we detect **[P]** as display space is limited.

What, you say? You assigned a new set of dit's and dah's to the equal sign? Ah, wait. There is much more...

### AA/New Line and the A with two dots...

Yes, there are also a set of special international characters - such as letters with circles, dots, double dots, and other accent marks over them. We decode all of them. We do not display the accent marks, however. So Å is displayed as A. At least we get the characters correct! For Ch, we just show C. There are about 8 of these the decoder recognizes. But AA, a pro-sign for New Line, has the same code as A with two dots. So, we decided to decode that as an A. Well, who knows? Maybe that was the wrong decision. :) Besides, we already have **[P]**...

### 13 Other Special Characters

Now for the fun and crazy part: non-standard special characters! I already mentioned above that the equal sign on the **DMX-40** has a unique assigned code. We went hog wild and assigned codes to other special characters. Table 3 below shows the character and the code sequence we have assigned. Naturally, if you send these characters, no one except another **DMX** owner will know how to decode them! But who knows? They might just take on a life of their own someday!

Character	Code	Character	Code
=	.---..-	)	-.----.
\$	-.-...-	%	.-.-..-
&	..-.-.	;	..-.-.
<	-.-....	>	.-.-...-
_	....--	*	-.-.-.
#	-.-.-.	!	.-.-..-
+	..-.-.		

**TABLE 3: SPECIAL CHARACTER CODE SEQUENCE ASSIGNMENTS**

Of course we have endeavored to ensure there are no code overlaps. Hopefully, you will not find one!

Note that we have a code for ). This is because there is a standard Morse code for (, but is it (I guess) assumed to be for both a ( and a ), and you are supposed to figure out which. But the decoder is not smart enough to say "oh, there was a ( back there, so probably this is a ). It COULD do that, and maybe sometimes it would do so correctly. But you would have to put in all kinds of checks, to make sure there was no break in the decoding for example. Not worth the effort. So, if you see something like:

I live in Sandpoint (Idaho( with my dog.

You will know it was supposed to be

I live in Sandpoint (Idaho) with my dog.

Of course when communicating with another **DMX-40**, you can actually use both ( and ) and they will see it correctly. :)

If you disapprove of these added characters, simply don't use them.

## 16. Appendix III: Calibrating your VFO

The VFO "chip" or integrated circuit that is included on every transceiver board utilizes a 27 MHz crystal. All frequencies are digitally generated from this crystal frequency. This crystal is quite accurate all on its own (about 0.004%), but your transmitter can be off by enough that if you are transmitting at the edge of your allowed band, your actual frequency might be out of your band range. That can get the FCC and other amateur operators pretty upset. Of course you can avoid trouble by staying at least 500 Hz from the edges of the band, but it is even better to know your frequency is accurate.

At the factory, we have calibrated our frequency meter against WWV, which is atomic clock accurate, and then calibrated the VFO against the calibrated frequency meter. The result? Your frequency is accurate to within a few Hz - about 0.00005% error. Thus, we do not recommend you change it unless you have access to equipment that can match that accuracy, or WWV can be received in your area. It operates on 10.0 MHz as well as 5.0 MHz, but the calibration function in the VFO chip requires we use the 10 MHz station. You also have to know how to "zero beat" an AM station (yes, WWV is an AM station, with a continuous carrier frequency). It is best to do it when they are transmitting a tone. Then, you can hear the very low frequency beat as a wavering in amplitude of the tone being transmitted. It is easy to get close from the loud tone that gets lower and lower (and then higher and higher once you pass the frequency), but once you are very close, you cannot hear the very low frequencies. But when WWV is transmitting a tone with one-second clicks, you can easily move the frequency in single Hz steps so the wavering gets slower and slower and then goes away - right at the frequency of 10 MHz.

Do this from the VFO Adjust screen using the 4-way frequency controls - the arrow keys, in the same way you tune in a station. It is best to just go to 10 MHz in Receive Mode, and try to zero-beat at the receiver. It should be very close from factory settings, so try going away 10 or 100 or so Hz and, without looking, see if you can get back to zero beat. Do this until you are very comfortable getting it right before you actually try it in the VFO Adjust screen.

## 17. Appendix IV: Terminology

There are a few terms we will use in this manual that are used frequently that should be understood:

- ANS: ANS is the abbreviation for ANSWER, one of the two transmit modes. in this mode, you are answering someone - either a General Call they are making, a Directed Call to you they are making, or during a back-and-forth conversation between you and another station. They "talk" and you "answer." **Details in [Section 12](#).**
- CALL: CALL is the second transmit mode for the **DMX-40**. Here you are transmitting a General Call, or you are calling a specific station (Directed Call). Details in [Section 11](#).
- CCS: This abbreviation stands for Captured Call Sign. On the display, there is an area where NO CCS is displayed, which means no call sign has been captured by the decoder. Details in [Section 11.6-11.7](#).
- CTX Computer-Assisted Morse Code Texting System. This is also used to indicate perfect Morse Code, with perfect timing.
- CW Continuous Wave. This term really refers to use of Morse Code to turn the transmitter on and off, usually by a manual "key" or by a computer. It was originally named this because there is no other modulation of the transmitter carrier wave other than being turned on and off rapidly using Morse Code.
- Dit, Dah: Morse code is made up of a series of short and long signals. These are called DITS (short) and DAHS (long). A dah is three dit's long. There are other values required for the space between dit's and dah's, space between characters, and space between words. A single character can take from one to six dit's and dah's to encode. Encoding is done simply by turning the transmitter on and off.
- Handle: This is the term used by Amateur Radio operators for the operator name.
- HCW HCW stands for Human-created CW, or Morse Code. This tends to be sloppy, and not conform to the definition of proper timing for either the dit's and dah's, but also for spaces between. It varies considerably from one operator to another. It is notoriously difficult to decode without some fairly advanced algorithms by a computer.
- HF HF stands for High Frequency. This designation was created many many years ago, and now, really stands for fairly low frequencies, starting just

above the AM broadcast band. The 80 meter band includes from 3.5 MHz to 4.0 MHz, for example.

- Micro-program: A microprogram is a pre-set series of characters that have been built into the DMX computer, or created by you to automate the transmit function of the transceiver. The built-in's include the CALL microprogram, the ANS microprogram, and the INFO microprogram. Each are described in detail in this manual. The purpose of a microprogram is to reduce the amount of typing you have to do to communicate by automating repetitive tasks. Up to 12 user microprograms can be defined using RPL (Rig Programming Language), defined in [Section 18 Appendix V](#).
- MHz: MHz stands for Megahertz, or what used to be called Megacycles per second. Some time ago, cycles per second was renamed to Hertz in honor of one of the greats in electric history. AM radio is from 550 KHz (Kilohertz) to 1600 KHz or 1.6 MHz. The HF bands extend from 80 Meters (3.5 MHz) all the way to 30 MHz.
- QTH: This is the term used by Amateur Radio operators for their location
- Rig: This is the term used by Amateur Radio operators for the transceiver they are using.
- RPL Rig Programming Language: For advanced users, RPL allows the creation of up to 12 user-defined microprograms. This allows the user to configure the **DMX-40** for different modes of interaction, such as DX, contests, field day, and for setting up automation for special cases, such as contacting buddies.
- Technician: A Technician Class license is the lowest available FCC license for use in Amateur Radio service. However, it comes with serious restrictions, which are some of the reasons for the existence of the **DMX-40**.
- VFO: Variable Frequency Oscillator. Transceivers use a VFO to control what frequency the receiver is tuned to, as well as the transmit frequency. For the **DMX** transceiver, the VFO is controlled by the computer controller. However, the actual VFO chip resides on the Tx/Rx board. Thus, if there is no Tx/Rx board installed, there also is no radio frequency generated.
- WWV: WWV is a US standards radio station at 10 MHz. Its frequency is controlled extremely accurately by atomic clock, and can be used as a reference for correcting frequency errors in any unit capable of tuning it in. It is also a very accurate time standard.

## 18. Appendix V: RPL and User Microprograms

This section is designed for advanced, technically-minded users. However, as far as we know, you can't break anything by creating bad microprograms, and you always have the ESC key to get you back to receive mode if your microprogram fails, so experimenting is a great way to learn! Probably best not to do that on the air, however.

---

### 18.1 Introduction

One of the more exciting advanced features of the **DMX-40** is the technology underlying the CALL, ANS, and INFO functions. This technology is called **RPL**, for **Rig Programming Language**. This language consists of text, with special command characters and normal text. The entire function of the CALL, ANS, and INFO functions are completely defined by three character strings in memory, processed by RPL in real time.

These three microprograms, what we refer to as the DEFAULT microprograms, are always available to any user by pressing the **CALL** or **ANS** Button, or by typing the tilde (~) character within an outgoing message. However, there are many other ways one might want to operate besides the specific ways we constructed these three default microprograms. Of course, we built them the way we did for casual users that only use the transceiver for emergency and occasional communications. However, for more advanced users that want to operate in a more streamlined manner in different situations, RPL provides an excellent way for them to achieve the exact operating methodology they desire.

We strongly recommend not experimenting with RPL with an antenna attached. It is easy to make mistakes and incorrect assumptions until you get enough experience writing RPL code that it would be better to try out your creations using a dummy load or in External mode. It is possible to create never-ending sequences, and for that, we have added the ESC function to abort transmit mode, allowing you to escape your mistake.

As described in [Section 7.4](#), access to the Microprogram Editor is via Shift-Fn, where n is from 1 to 12, or via the **M** key, or via the Main Menu.. Thus, there are 12 user-defined microprograms. Executing a microprogram is by typing Fn. Control microprograms may be started from Receive Mode, and text microprograms can be inserted in transmit mode, even while another microprogram is running. An example of that is typing a tilde character in a message, which is during the execution of the default ANS microprogram. That inserts the default INFO microprogram into the outgoing text stream. Note that only text microprograms should be executed from Transmit Mode. Only control microprograms (which can of course include text) can be executed from Receive Mode.

---

## 18.2 The Built-In Microprograms

So what do the Built-In microprograms look like? The Built-in CALL Microprogram looks like this:

```
?<[@ @ @ @ DE * * ]KN%[CQ CQ CQ CQ DE * * ]K>
```

As you can see, there are a number of punctuation characters as well as text. All of the punctuation characters in this example are actually RPL commands. The Built-in ANS Microprogram looks like this:

```
!(@ <@ DE *%DE * ) BACK 2U @ DE> * KN
```

You might guess that "?" enters the CALL mode, and "!" enters the ANS mode, and you would be correct! Here is what the INFO Microprogram looks like:

```
NAME IS ^1, LOCATION IS ^2, USING ^3 WITH ^4.
```

Here we don't start either CALL or ANS, but just add some text. As you can probably figure out, ^n inserts data from the memory. In particular, ^1 inserts your name, etc.

It is important to understand the difference between the default CALL and ANS Microprograms, as executed from memory as shown above, and a CALL or ANS Microprogram you might create. Both of them operate from "someone" or "something" pushing the **CALL** or **ANS** Button on your LCD. For the normal case, it is your very own finger. In the user microprogram case, it is ... your microprogram! That's correct: your microprogram reaches out across the wires into the LCD and "pushes" the CALL or ANS Button! The button reacts just as if you pushed it! But you just pushed a function key to make that happen.

As you recall, both Built-in CALL and ANS Microprograms require a second "push" to the LCD button to end the repetition (in the CALL case) or to specify an Answer-Verify or end the message entry (in the ANS case), allowing the microprogram to complete and end. This is usually but not necessarily true of any user CALL or ANS microprogram, but again, the pushing of the LCD buttons is by your microprogram, over the wires, to the LCD. So you will see the buttons on the LCD behave as if you have pushed them both times. How do you push them the second time? You have two ways: you can actually push the LCD button, or you can press the same function key that started the currently running CALL or ANS microprogram a second time.

For example, if you created a CALL microprogram for F1, to start it, press F1. To cause it so complete its current repeat and finish, press F1 a second time For more complex

microprograms, it is possible for them to require more than 2 function key presses, base on your design, of course.

---

## 18.3 Microprogram Types

In our presentation above of the Built-in microprograms, each represents a "type" of microprogram. There are three types of microprograms:

1. **CALL Microprogram:** this is the method of entering transmit CALL Mode, and usually does not require any user typing. These microprograms must include a ? character.
2. **ANS Microprogram:** this is the method of entering transmit ANSwer mode, and usually requires some user typing - the message. These microprograms must include a ! character.
3. **Text Microprogram:** the INFO microprogram is a text microprogram. It types commonly desired information for you into the message stream, saving you time and effort.

Using the RPL control characters, some imagination, and some trial and error, you will be able to customize your **DMX-40** transceiver operation to your hearts desire, and enable you to operate exactly the way you want. You will be able to optimize it for contests, for calling friends at their favorite CW speeds, and much more.

---

## 18.4 RPL Control Characters

Below is a table of the RPL control characters. A print-out version is available in [Section 19 Appendix VI](#) in a form for copying and using next to your keyboard when writing microprograms.

Character	Description of Operation
*	Insert User Call Sign
^1	Insert User Name
^2	Insert User Location
^3	Insert User Rig
^4	Insert User Antenna

Character	Description of Operation
`c	The ` key is the escape character. It allows you to use any of the RPL control characters as text, such as ?, !, etc. "c" here represents the character you are using that is also a control character.
@	Insert CCS call sign. If none is defined, insert a `? (this is escape-? to avoid the ? from being misunderstood as a CALL command)
?	Enter CALL mode (by pressing the CALL Button). A second press of the same function key will press the CALL Button again. Subsequent presses will do likewise but have no effect on the CALL Button. This command also turns off the keyboard input to the type-ahead buffer as well as output from the type-ahead buffer. This command can be preceded by a CCS command and/or a speed command or other commands. If the transmit speed has not been set by the microprogram itself, the decoder, or the user prior to the execution of the microprogram, the transmit speed will automatically be requested from the user. CALL mode ends when the microprogram completes execution. This command is ignored if either a CALL or ANS mode is already in progress.
!	Enter ANS mode (by pressing the ANS Button). A second press of the same function key will press the ANS button again. Subsequent presses will do likewise but have no effect on the ANS Button. Keyboard input to the type-ahead buffer is enabled by this command, but output from the type-ahead buffer is disabled. This command can be preceded by a CCS command and/or a speed command, or other commands. If the transmit speed has not been set by the microprogram itself, the decoder, or the user prior to the execution of the microprogram, the transmit speed will automatically be requested from the user. If the CCS has not been defined, either by the microprogram itself, by the decoder or the user, it will be requested automatically from the user. ANS mode ends when the microprogram completes execution. This command is ignored if either a CALL or ANS mode is already in progress.
	Wait for condition, and enable the keyboard type ahead buffer output. The condition waited for is an empty keyboard type ahead buffer, an empty encoder, and a second press of the same function key. Note that keyboard input to the type-ahead buffer may or may not be enabled during the wait (depending on the previous commands). Function key presses are remembered and saved via a counter, so a microprogram may be designed to require more than one additional press of the function key.
(	Enables keyboard to type-ahead buffer.

Character	Description of Operation
)	Disables keyboard to type-ahead buffer, enables type-ahead buffer output, and waits until the buffer and encoder is empty before continuing to the next character.
[...]	Repeat the microprogram characters between the brackets until a second (additional) press of the function key is received. At that time, the current repeat is completed, and the microprogram continues with the character immediately following the ] character. Only one level of repeats is allowed, i.e., no repeats within repeats. However, multiple repeats may be used in a single microprogram. Thus, if a repeat is in process, a [ character is ignored, and the second ] character is also ignored. The repeat will run between the first [ and the first ].
#nn	<p>This command has 4 different functions, depending on what character follows:</p> <p>#C: This utilizes or executes the built-in CALL microprogram, so you don't have to type in the long string of characters of the CALL microprogram. An easy way to put CALL on a function key!</p> <p>#A: This utilizes or executes the built-in ANSwer microprogram, for the same purpose as #C.</p> <p>#Tnn: This function allows you to specify a delay or a repeating timer. The value of nn is 01 to 99, 00 for 1 to 99, 100 minutes. If used outside of a repeat loop, it is a delay. If used inside of a repeat loop, it starts a timer and continues the loop first time. Second time, it stops until timer runs out, then restarts the timer and continues through the loop the second time, etc.</p> <p>#nn: This is a jump to microprogram nn. Nothing from the current microprogram remains.</p>
&nn	Insert microprogram nn, where nn is from 01 to 12. This is typically used to expand a text microprogram, where the text microprogram replaces the &nn characters. It works more like a subroutine call - you do not have to worry about running out of space. Thus, you can have multiple in a row, such as &02&03&04, etc., for long text.
~	Inserts the built-in INFO microprogram.
{CCS}	Define the CCS. Use BEFORE ? or ! command.

Character	Description of Operation
<b>=nn</b>	Set transmit speed to nn, where nn is from 05 to 60, in words per minute (WPM). Use BEFORE ? or ! command.
<b>&lt;a%b&gt;</b>	<p>This microprogram IF construct operates in two different ways:</p> <p>In CALL MODE: If CCS is defined, use sequence a, otherwise use sequence b. When this function executes, what remains is just a or b: the &lt;, &gt; and % characters are removed along with the sequence not to be used.</p> <p>In ANS MODE: If ANS Button is pressed a 2nd time (a quick double press), then use sequence a, otherwise use sequence b. When this function executes, what remains is just a or b: the &lt;, &gt; and % characters are removed along with the sequence not to be used.</p> <p>Note: The test for 2nd press on the ANS button (or function key that started an ANS transmission) is checked when the &lt; character is executed in the microprogram. In the same way, the &lt; character is where the CCS is evaluated for a CALL microprogram. Thus, the &lt; is the IF, and the &gt; is the END IF.</p>

**TABLE 4: RPL CONTROL COMMANDS**

---

## 18.5 Built-in Microprogram Review

Let's go back and look at the Built-in microprograms, and analyze them, now that we have revealed the functions of the various control characters. Let's start with the CALL microprogram:

**?<[ @ @ @ @ DE \* \* ]KN%[CQ CQ CQ CQ DE \* \* ]K>**

The execution sequence for this microprogram, assuming the CCS has not been defined, is shown in Table 5 below:

Character	Execution
<b>?</b>	Press CALL Button, enters transmit CALL mode
<b>&lt;</b>	Test if CCS is defined. Assume Not for the example
	Delete <[ @ @ @ @ DE * * ]KN% and >

Character	Execution
	What remains is [CQ CQ CQ CQ DE * * ]K
[	Start the repeat
<b>CQ.. DE</b>	Send CQ CQ CQ CQ DE to encoder
* *	Send user call sign to encoder twice (including spaces)
]	Repeat from [ to ] until second function key press or CALL button press
<b>K</b>	Send K to encoder
<b>(end)</b>	Terminate CALL mode, return to receive mode, engage RIT tuning

**TABLE 5: BREAKDOWN OF BUILT-IN CALL MICROPROGRAM**

Let's look at the ANS microprogram, shown again below:

**!(@ <@ DE \*%DE \* |) BACK 2U @ DE> \* KN**

Table 6 describes the execution sequence of this microprogram:

Character	Execution
<b>!</b>	Press ANS Button, enter transmit ANS mode
<b>(</b>	Enable keyboard input to type-ahead buffer
<b>@</b>	Insert CCS (must be defined, because if not, it will ask ANSWER WHO? Sends the CCS and a single space to the encoder
<b>&lt;</b>	Test if second press of ANS button has occurred In this case, assume that it has NOT been pressed. Thus, <@ DE *% and the > character has been removed.
	What remains is DE *  ) BACK 2U @ DE * KN
<b>DE</b>	Send DE and spaces to encoder
<b>*</b>	Insert user call sign with surrounding spaces
<b> </b>	Enable the type-ahead buffer output, wait for empty encoder, buffer and a second function key or ANS Button press.
<b>)</b>	Disables keyboard input to buffer, waits until buffer and encoder is empty (short wait because preceded by a   character)
<b>BACK 2U</b>	Send BACK 2U and spaces to encoder
<b>@</b>	Insert CCS

Character	Execution
<b>DE</b>	Send DE to encoder with spaces
*	Insert user call sign with spaces
<b>KN</b>	Send KN to encoder
<b>(end)</b>	Terminate ANS mode, return to receive mode, engage RIT tuning

**TABLE 6: BREAKDOWN OF BUILT-IN ANS MICROPROGRAM**

For the INFO microprogram, it is simply inserted into the microprogram as it stands, and the insertion of name, etc. occurs when the RPL microprogram processor reaches the ^ command characters.

---

## 18.6 Inserting Into a Running CALL or ANS Microprogram

Note that while an ANS or CALL microprogram is running, other function keys may be pressed. However, they will only have an effect if the keyboard to type-ahead buffer is enabled. The microprogram will be inserted into the keyboard type-ahead buffer, and displayed as a "f" in the type-ahead buffer display on the LCD. This is to conserve buffer space - the actual function number is not displayed, and only one special character is placed into the actual buffer. When the encoder receives the function key character, the microprogram is inserted at that moment into the running microprogram. This works very much like the ~ INFO microprogram, where you can type, press ~, and type some more. For example, if you typed

HELLO, THANKS FOR COMING BACK TO ME!~IT'S SUNNY HERE, HOW ABOUT THERE?

In this case, the INFO microprogram is not inserted until the ~ character is reached. Since it includes a starting and ending space, it is as if you typed:

HELLO, THANKS FOR COMING BACK TO ME! NAME IS JOHN, LOCATION IS CHICAGO, IL, USING DMX-40 WITH 40M DIPOLE. IT'S SUNNY HERE, HOW ABOUT THERE?

Your info (text) microprograms can include whatever info you like to offer your contacts, and can likewise be inserted into your message stream at the touch of a function button. **Remember to include a starting and ending space!**

## 18.7 The Microprogram Editor

The microprogram editor is designed to make it easy to enter and edit your microprograms. The Microprogram Editor screen is shown in Figure 22a below:



**FIGURE 22A: MICROPROGRAM EDITOR WITH NO MICROPROGRAM**

The function key number is displayed, along with any microprogram definition previously stored. In Figure 22a, there was no previous definition for F6, so the cursor will be placed at the beginning of the first line. As you type, the cursor will move ahead of the characters as they are being added to the microprogram. Sorry, the display does NOT support blinking cursors... (sigh).

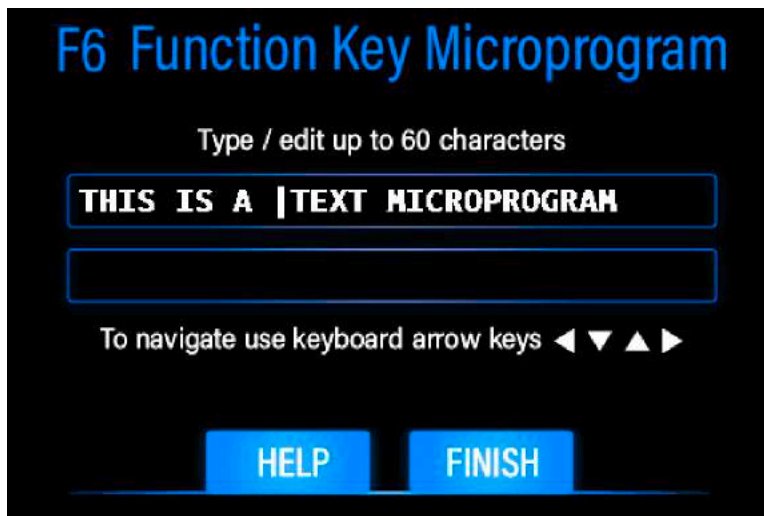
You can position the cursor anywhere in the microprogram using the four **arrow** keys, the **up-arrow** and **down-arrow** keys allow movement quickly from the bottom line to the top line, or vice versa. They will only allow movement within the microprogram, however, so if there is only one line of text, they will do nothing. The **left-arrow** and **right-arrow** Keys move the cursor left and right. Since the two lines actually represent a single long string of characters, moving to the right from the end of line one will get you to the beginning of line two. Likewise, backing up at the beginning of line 2 will get you to the end of line one. Essentially, the **arrow** keys operate as a four-way movement control for the cursor.

To demonstrate how to enter and edit a text microprogram, Figure 22b on the next page displays what the screen looks like after you type in the text "THIS IS A TEXT MICROPROGRAM". Note that the cursor is positioned at the end of the line, ready to accept another character, to be inserted **BEHIND** the cursor as it moves forward one step. Likewise, deleting a character deletes the character **BEHIND** the cursor.



**FIGURE 22B: TEXT EDITOR AFTER TYPING IN CHARACTERS**

Figure 22 shows what the screen looks like after we have moved the cursor to the left, using the **left arrow** key, to do an insert:



**FIGURE 22C: READY FOR AN INSERT**

Since the display does not have a "cursor" function, the DMX-40 simulates it using the | character. Unfortunately, this takes up a character space, so it does feel a little weird moving the cursor and seeing it move the text around to insert itself between characters. However, you will get used to it. Typing in "VERY LONG " above in Figure 22c would then result in the microprogram "THIS IS A VERY LONG TEXT MICROPROGRAM"

Naturally, this will not fit on the first line, so it will spill over to the second line. Just like you would expect. You would not believe the programming required to make this all work on a display that does not support cursors! But yes, this is programmed into the actual display as part of the **DMX-40** programming.

Once you have entered and/or edited your microprogram, press the **FINISH** Button to save it to memory. The **FINISH** Button is also the way out if you have not changed anything, i.e., it is also the **EXIT** Button.

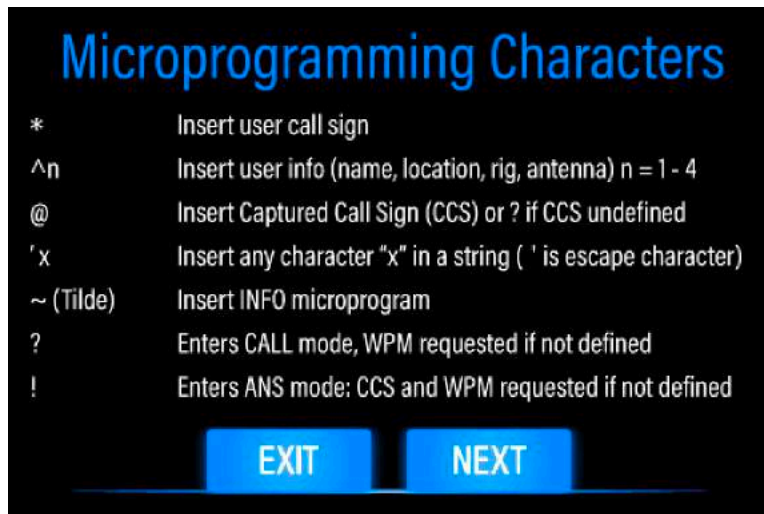
The **HELP** Button takes you to the same Microprogramming Help Screens that are accessible from the **HELP** Menus, covered in more detail below.

A good technique to keep in mind when entering a text microprograms is to place a **SPACE** before and after the text - in other words, first and last character. This conforms to the standard of not having to enter a space before and after pressing the function key to send your microprogram. It does add two characters, but prevents run-on words in our outgoing transmission. NOTE: we did NOT do that in the example above.

---

## 18.8 Microprogram Help

There are three Help Screens available (in case the manual is not handy), which are shown in Figures 23a, 23b and 23c below. You will always have these to refer to in case you mislaid your manual, or are off on a field trip and want to create a new microprogram. Note that the EXIT Buttons on these Help Screens do not return to the Main Screen, like they do if you access them from the Help System. Rather, they return you to the Microprogram Editor - just like you need!



**FIGURE 23A: PAGE 1 OF RPL CONTROL COMMAND HELP**

This is the first microprogram help screen. You should recognize the commands - they are the same ones we have already shown in the table and examples above. In Figure 23b and 23c below, the rest of the commands are displayed:

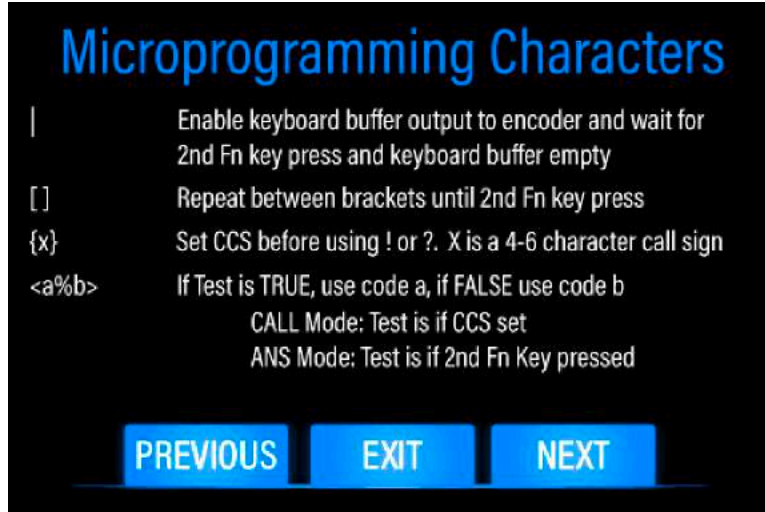


FIGURE 23B: PAGE 2 OF RPL CONTROL COMMAND HELP

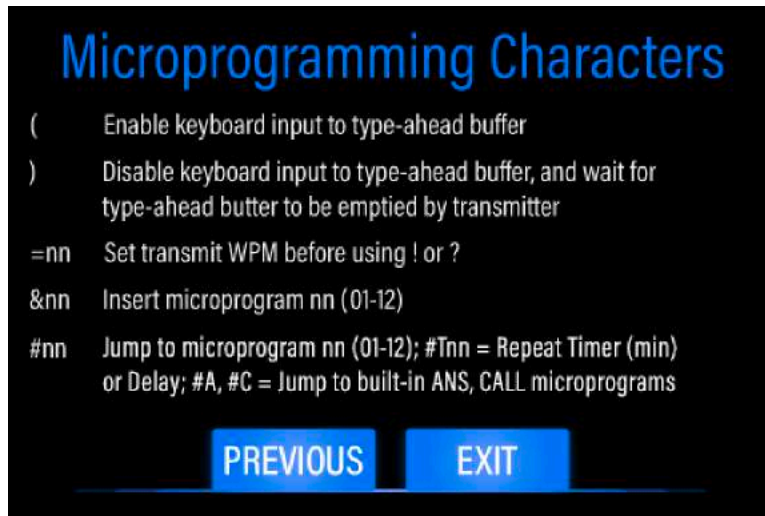


FIGURE 23B: PAGE 3 OF RPL CONTROL COMMAND HELP

---

## 18.9 Some Examples

1. F1 = "?@ DE \*" This is a very short station ID, for those in a hurry. Note that this microprogram will terminate immediately after sending your call letters - no second function key press is required.
2. F2 = "?ID: \* WITH @" This microprogram does a quick station ID with your call letters and the other station's call letters. Nice for a long QSO.

3. F3 = "!()" Just lemme transmit microprogram. It sends nothing but what you type. Press the function key again to end. Great for long QSO's along with #2 for ID every 10 minutes.
4. F4 = "=18{K2AAI}?@ @ DE \* \* KN" and F5 = "=12{W1FAZ}?@ @ DE \* \* KN"  
These two are examples of setting up to call your buddies where their call letters and their speed is preset by the microprogram. Since no repeat is included, they will run once. Of course you can add a repeat as follows: F4 = "=15{K2AAI}?[ @ @ DE \* \* ]KN" and F5 = "=12{W1FAZ}?[ @ @ DE \* \* ]KN" In that case, a second function key press will be required to end the repeat.
5. F1 = "17#T05?[#T30DE W2OL BULLETIN: #02#03 MSG REP 30 MIN.]" This is a message that is delayed 17 minutes (probably because it is 13 minutes past the hour or 17 minutes before the hour when it was started) which will repeat every 30 minutes, on the hour and half hour. The message begins with DE W2OL BULLETIN as station ID and identification of the purpose of the transmission, followed by up to 120 character message stored in microprogram 2 and 3.

These few examples should suffice to get you thinking of things you might want your computer automated **DMX** transceiver to do! Have fun, and if you create a killer microprogram, send it to us! Or, if you think of a killer microprogram you can't program with the existing RPL language, send us a request for additional features!

## 19. Appendix VI: Keyboard and RPL Command Layouts

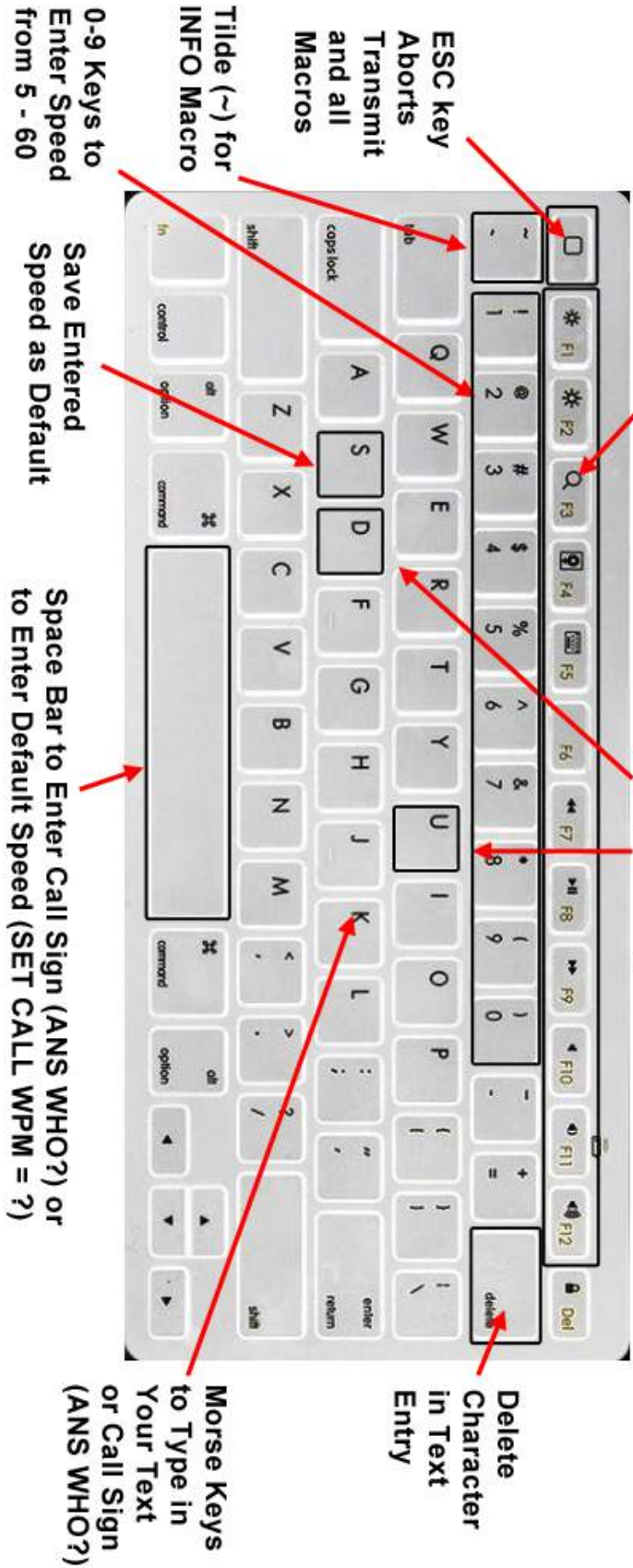
On the next four pages are keyboard layout diagrams and an RPL command table diagram for your use. Print or copy them, and ideally laminate them. They were designed for you to keep next to your station when learning to use the **DMX** transceiver. The first two keyboard layouts summarize the control and function keys available to you in transmit and receive mode. The RPL Command charts are provided to give you a quick access to the commands when you are constructing your microprograms.

### Receive Mode Keyboard Controls

<p>F1 - F12 to run 12 user Macros</p> <p>Shift F1 - F12 to program Macros</p> <p>ESC key to switch between CTX and HCW Decoder Modes</p> <p>Drop Freq 2600 Hz</p> <p>Frequency Memories</p> <p>Store Gain Settings</p> <p>Microprograms</p> <p>Gain Controls &lt; and &gt; for Decoder Gain, , and . for Headphone Gain</p> <p>Receiver Frequency Control Up/Down controls Step (1 to 10K) Left/Right controls frequency</p>		<p>R = Exit RIT to Receiver Freq</p> <p>T = Exit RIT to Transmitter Freq</p> <p>Raise Freq Up 2600 Hz</p> <p>0 - 9 keys load from Frequency Memories shift 0-9 to loads to Frequency Memories</p> <p>Delete</p> <p>Character in Text Entry</p> <p>Key Tone ON/OFF</p>
--	--	---

## Transmit Mode Keyboard Controls

F1 - F12 to Insert 12 user Macros  
D (Down) and U (Up) key to shift down or up the tone frequency instantly



## Table of RPL Commands

Character	Description
*	Insert User Call Sign
^1	Insert User Name
^2	Insert User Location
^3	Insert User Rig
^4	Insert User Antenna
`c	The ` key is the escape character. It allows you to use any of the RPL control characters as text, such as ?, !, etc. "c" here represents the character you are using that is also a control character.
@	Insert CCS call sign. If none is defined, insert a `? (this is escape-? to avoid the ? from being misunderstood as a CALL command)
?	Enter CALL mode (by pressing the CALL Button). A second press of the same function key will press the CALL Button again. Subsequent presses will do likewise but have no effect on the CALL Button. This command also turns off the keyboard input to the type-ahead buffer as well as output from the type-ahead buffer. This command can be preceded by a CCS command and/or a speed command. If the transmit speed has not been set by the macro itself, the decoder, or the user prior to the execution of the macro, the transmit speed will automatically be requested from the user. CALL mode ends when the macro completes execution. This command is ignored if either a CALL or ANS mode is already in progress.
!	Enter ANS mode (by pressing the ANS Button). A second press of the same function key will press the ANS button again. Subsequent presses will do likewise but have no effect on the ANS Button. Keyboard input to the type-ahead buffer is enabled by this command, but output from the type-ahead buffer is disabled. This command can be preceded by a CCS command and/or a speed command. If the transmit speed has not been set by the macro itself, the decoder, or the user prior to the execution of the macro, the transmit speed will automatically be requested from the user. If the CCS has not been defined, either by the macro itself, by the decoder or the user, it will be requested automatically from the user. ANS mode ends when the macro completes execution. This command is ignored if either a CALL or ANS mode is already in progress.

[Return to Table of Contents](#)

## Table of RPL Commands

Character	Description
	Wait for condition, and enable the keyboard type ahead buffer output. The condition waited for is an empty keyboard type ahead buffer, an empty encoder, and a second press of the same function key. Note that keyboard input to the type-ahead buffer may or may not be enabled during the wait (depending on the previous commands). Function key presses are remembered and saved via a counter, so a macro may be designed to require more than one additional press of the function key.
(	Enables keyboard to type-ahead buffer.
)	Disables keyboard to type-ahead buffer, enables type-ahead buffer output, and waits until the buffer and encoder is empty before continuing to the next character.
[...]	Repeat the macro characters between the brackets until a second (additional) press of the function key is received. At that time, the current repeat is completed, and the macro continues with the character immediately following the ] character. Only one level of repeats is allowed, i.e., no repeats within repeats. However, multiple repeats may be used in a single macro. Thus, if a repeat is in process, a [ character is ignored, and the second ] character is also ignored. The repeat will run between the first [ and the first ].
#nn	This command has 4 different functions, depending on what character follows: #C: This utilizes or executes the built-in CALL microprogram, so you don't have to type in the long string of characters of the CALL microprogram. An easy way to put CALL on a function key! #A: This utilizes or executes the built-in ANSWER microprogram, for the same purpose as #C. #Tnn: This function allows you to specify a delay or a repeating timer. The value of nn is 01 to 99, 00 for 1 to 99, 100 minutes. If used outside of a repeat loop, it is a delay. If used inside of a repeat loop, it starts a timer and continues the loop first time. Second time, it stops until timer runs out, then restarts the timer and continues through the loop the second time, etc. #nn: This is a jump to microprogram nn. Nothing from the current microprogram remains.

[Return to Table of Contents](#)

## Table of RPL Commands

Character	Description
<b>&amp;nn</b>	Insert microprogram nn, where nn is from 01 to 12. This is typically used to expand a text microprogram, where the text microprogram replaces the &nn characters. It works more like a subroutine call - you do not have to worry about running out of space. Thus, you can have multiple in a row, such as &02&03&04, etc., for long text.
<b>~</b>	Insert the INFO macro. This does the same thing as &nn, but with the special default INFO macro.
<b>{CCS}</b>	Define the CCS
<b>=nn</b>	Set transmit speed to nn, where nn is from 05 to 60, in words per minute (WPM).
<b>&lt;a%b&gt;</b>	This microprogram IF construct operates in two different ways:  In CALL MODE: If CCS is defined, use sequence a, otherwise use sequence b. When this function executes, what remains is just a or b: the <, > and % characters are removed along with the sequence not to be used.  In ANS MODE: If ANS Button is pressed a 2nd time (a quick double press), then use sequence a, otherwise use sequence b. When this function executes, what remains is just a or b: the <, > and % characters are removed along with the sequence not to be used.  Note: The test for 2nd press on the ANS button (or function key that started an ANS transmission) is checked when the < character is executed in the microprogram. In the same way, the < character is where the CCS is evaluated for a CALL microprogram. Thus, the < is the IF, and the > is the END IF.

## 20. Appendix VII: Common CW Abbreviations

Below is a list of common abbreviations used in CW communications.

73--Best Regards  
88--Hugs and Kisses  
ABT--About  
AGN--Again  
ANT--Antenna  
ARND--Around  
B4--Before  
BCNU--Be Seeing You  
BK--Break, Back ("I'm back now" or "Back to you")  
BN--Been  
BTR--Better  
BUX--Dollars  
CLDY--Cloudy  
CLR--Clear (usually a description of the weather)  
CNTCT--Contact  
CONDX--Conditions  
CPY,CPI--Copy  
CU--See you  
CUD--Could  
CUL--See You Later  
CUZ--Because  
DR--Dear (often used by foreign/DX operators in front of the other operator's name)  
ES--And  
FB--Fine Business (good, excellent, OK)  
FER--For  
FT--Feet  
GA--Good Afternoon, Go Ahead  
GB--Good Bye, God Bless  
GE--Good Evening  
GL--Good Luck  
GM--Good Morning  
GUD--Good  
HI--Telegraphic Laughing  
HPE--Hope  
HR--Here, Hear  
HV--Have  
HW--How (or "How copy?")  
LIL--Little  
MNI--Many  
NR--Near, Number  
NW--Now

[Return to Table of Contents](#)

O 0--(usually used when stating transmitter output of 100 Watts. If using a straight key it is also OK to use a very long dah to represent 0.)

OM--Old Man

OP--Operator (Sometimes used insted of "name" when introducing oneself.)

PSE--Please

PWR--Power

R--are

RCVR--Receiver

SIGS--Signals

SN--Soon

SRI--Sorry

TMW--Tomorrow

TNX, TKS--Thanks

TU--Thank You (usually used at end of contact, meaning "Thanks for the contact".)

TX--Transmit

U--You

UR--Your, You Are

URS--Yours

VY--Very

WID--With

WL--Well, will, we'll

WTS--Watts

WUD--Would

WX--Weather

XCVR--Transceiver

XMTR--Transmitter

XYL--Wife

YL--Girlfriend ("Young Lady")

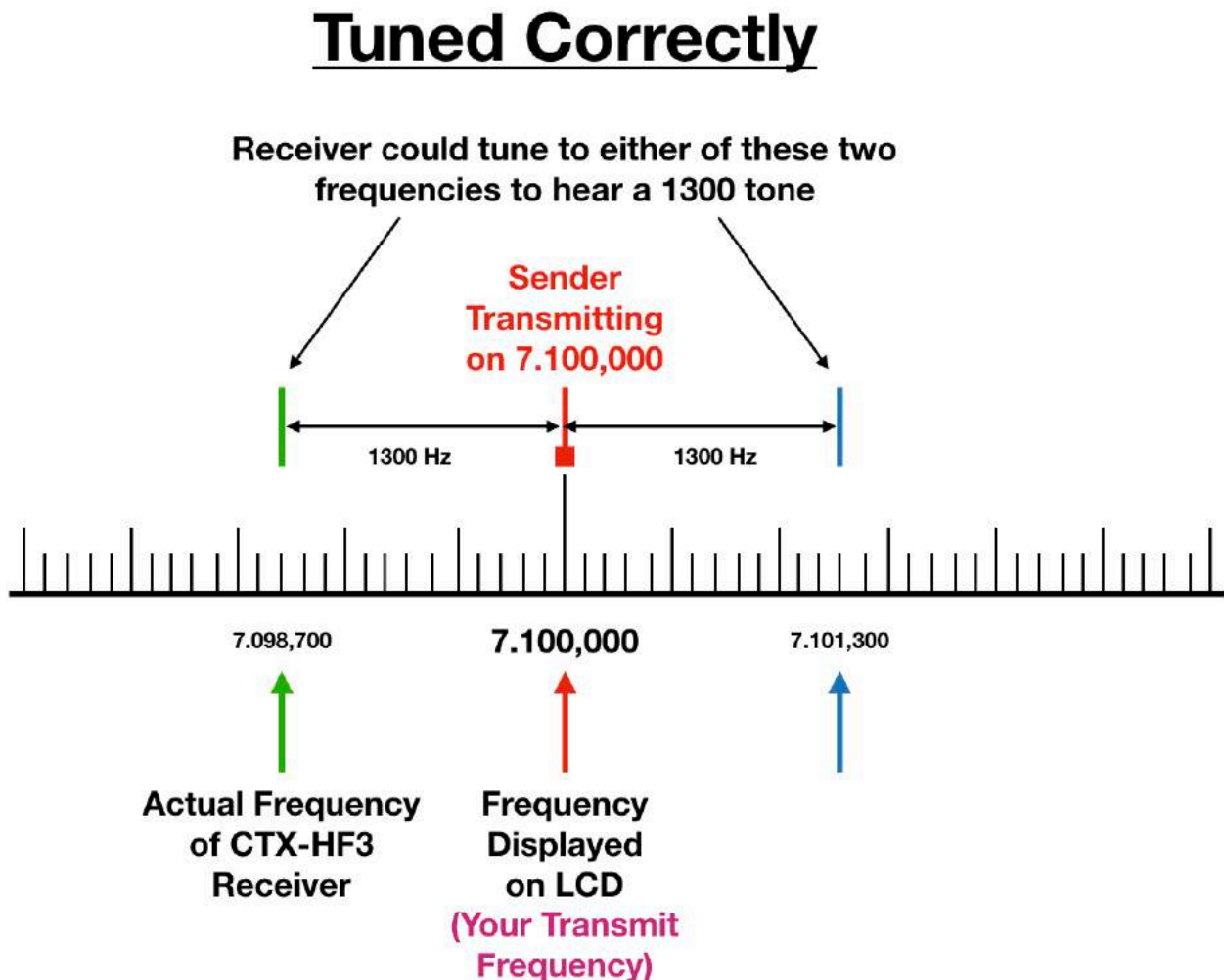
YRS--(Years)

## 21 Appendix VIII: The Tuning Rule Graphically Explained

For any Morse Code signal you receive, there are actually two frequencies where a correct tone will be generated and thus activate your decoder. One is 1300 Hz below the signal, and one is 1300 Hz above.

We handle this by setting the receiver frequency 1300 Hz below what the frequency display says. Of course the frequency of the transmitter is exactly what the frequency display says. Thus, if the frequency on the LCD says 7,100,000 Hz, or 7.1 MHz, the receiver is listening at 7,098,700 Hz and the transmitter will transmit on 7,100,000 Hz.

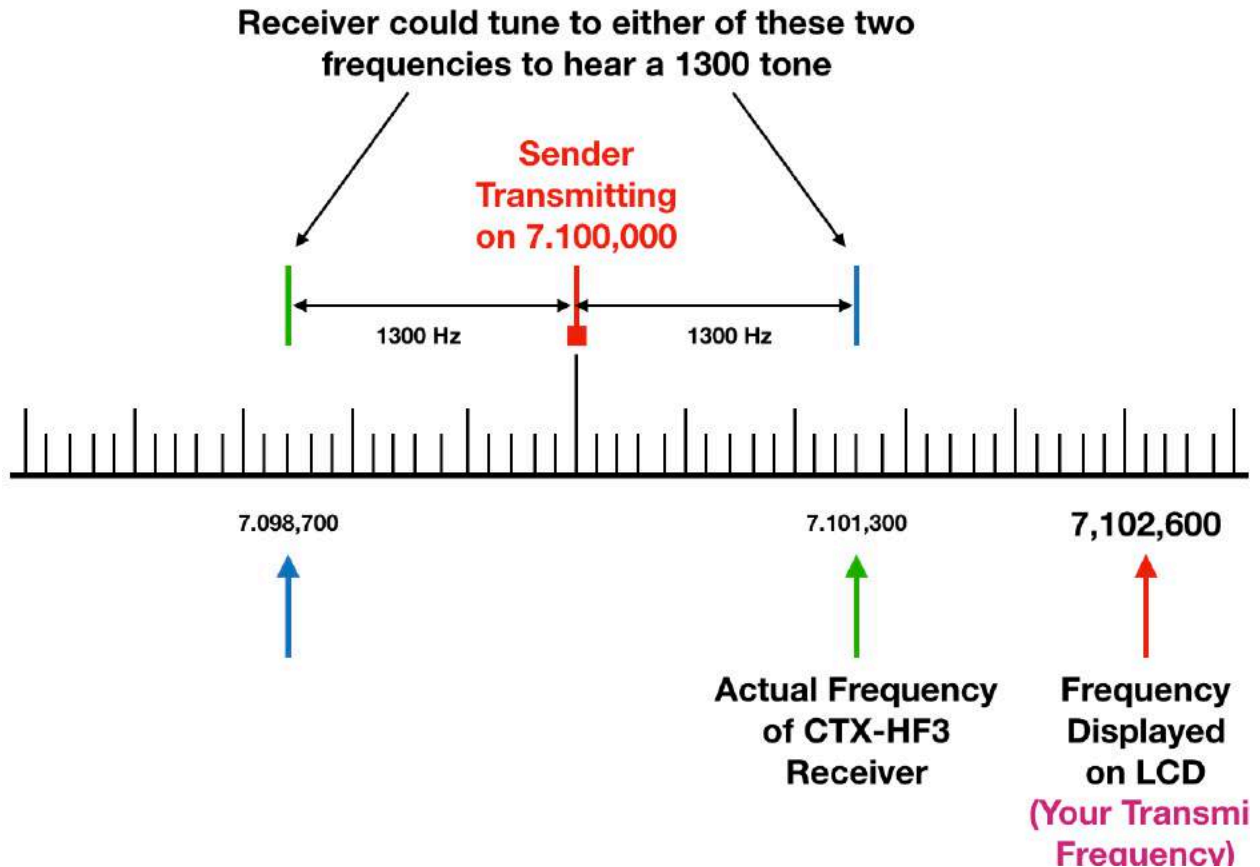
Below is a diagram of a properly tuned signal.



As you can see, the frequency displayed matches the sender's frequency, and we are listening at the lower of the two frequencies where a 1300 Hz tone can be heard. Note that as you move the frequency of the transceiver upward, the tone will decrease until

you get to 7,101,300, where it will disappear at a very low frequency. As you continue up in frequency, the tone will rise up again, and will reach 1300 Hz when you tune to 7,102,600 Hz as shown below:

## Tuned Incorrectly



Here, you are tuned to the upper frequency where a 1300 Hz tone can be heard, but since the receiver is set assuming you are tuned to the lower frequency, you are actually tuned 2600 Hz too high. Your transmitter frequency is now WAY OFF from the sender - so far off, they will probably not hear you (their tone filter will filter you out).

**So, the Tuning Rule for finding a signal while tuning around is simple:**

**Step up the receiver frequency 10 Hz. If the tone goes lower, you are set. If not, drop down 2600 Hz by pressing the D key.**

## 22. Appendix IX: Alternate USB Keyboards

While it may seem reasonable, you cannot plug most USB keyboards into the USB connector on the DMX-40 transceiver. The reason for this is pretty simple. Most USB keyboards today are designed for working with USB 3.0 or higher. This means they have a USB HID chip inside them, so they can send keyboard characters at the full bus speed of USB.

When USB 1.0 was first implemented, it was to replace PS/2, a serial keyboard and mouse system created by the initial IBM PC. Thus, it was possible to send the same slow serial signal from the keyboard to the computer, essentially by shutting down the USB high-speed clocked communication system and allowing a much slower device to communicate. Since keyboard inputs are not all the frequent (in high-speed space), USB 1.0 was made backward compatible. Probably also increased the acceptance of USB 1.0. That same approach continued in USB 2.0. But of course if you were going to connect disks and other high data rate devices to USB, as is common in USB 3.0 and 4.0, then being interrupted by a slug of a signal from an old keyboard was not a great solution. So bye bye for the old times, bring in the new keyboards!

The new keyboards then are significantly more expensive, and require a USB HUB chipset and a fairly substantially large software stack, and thus a fast processor, and lots of memory. Or, one could say, a modern OS.

The DMX-40 has no OS. It talks directly to the "metal" and is a true real-time multi-tasking software program. Memory space is limited, and to keep power consumption down, size down, and required memory space and processor speed down for lowering cost, we choose to continue to use the old-style PS2 USB keyboards. There are lots of sources for them, including Amazon, AliExpress, Alibaba, etc. Even flexible keyboards you can roll up. Just to avoid confusion and consternation, we included a very nice keyboard with your DMX-40, but feel free to look around for a different one if your heart so desires. Let us know if you find one you think is better!

## 23. Wiring the 3.5 mm Jacks

The **AUDIO IN** Jack and **HEADPHONE** Jack are standard stereo 3.5 mm audio jack. Both channels (Left and Right) are connected together (shorted) to provide a signal to the decoder from your transceiver, or to your headphones from the internal receiver.

**AUDIO IN:** Since it is assumed that a headphone signal is being used, 200 mv (millivolts) is sufficient for a full decoder response. Decoder response will begin around 50 mv, but frequency selectivity will be narrow, perhaps as narrow as  $\pm 50$  Hz around 1300 Hz. As the amplitude increases, selectivity decreases to perhaps  $\pm 75$  Hz, and maxes out at  $\pm 90$  Hz to 100 Hz at 200 mv or higher. While it may seem advisable to use a lower level to achieve more selectivity, keep in mind band conditions. If the signal is fading in and out, a higher volume is required so the lowest volume in the fade cycle is still decodable. Note the too high a volume will cause the decoder to turn off too slowly, and miss spaces between dits and dahs. Thus, reduce decoder gain, or volume on your external rig to eliminate overdriving the decoder.

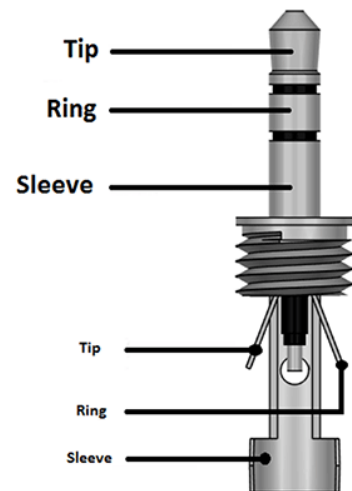
**KEY IN/OUT:** This jack is also a 3.5 mm audio jack, and requires a 3.5mm stereo plug as depicted in the image to the right. The **DMX-40** cable wiring is as follows:

FOR **KEY OUT:**

USE **Sleeve** for ground shield  
USE **Tip** for KEY OUT

FOR **KEY IN:**

USE **Sleeve** for ground shield  
USE **Ring** for KEY IN



**DO NOT ATTEMPT TO USE THE SAME CABLE FOR BOTH FUNCTIONS!**

**PLEASE NOTE that SHIELDED WIRE MUST BE USED to prevent RF from getting into the internal circuitry. This may cause your DMX-40 to lose it's mind, so to speak. Avoid problems by using shielded wire!**