MMX ZERO
Morse Code
Decoder/Encoder
REFERENCE MANUAL

By Eric C Anderson
PreppComm
Version 1.2, July 2023

Morse Made Easy™

MMX ZERO
Morse Code Decoder/Encoder
REFERENCE MANUAL

By Eric C Anderson
PreppComm
Version 1.2, July 2023

Copyright © 2023 PreppComm. All rights reserved.

Table of Contents

1. Overview

Congratulations for purchasing the new **MMX ZERO Morse Code Decoder/Encoder!** The **MMX ZERO** was created to provide a texting interface to the user for operating on CW, 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. And by the way, learning it even if you don't want to. The learning happens when you have QSO's automatically.

The **ZERO** is 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 device 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. The decoder can copy fading signals you can't hear under the noise!
3. Super selective decoder (160 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, approximately 1220-1380 Hz.
4. Automatic capture of remote station call letters. I love this one!
5. Built-in typing speed calibration to limit transmit speed to your comfortable speed, to prevent starving the transmitter while transmitting because the other station can send Morse faster than you can type Morse
6. 76 character type-ahead pop-up buffer for keeping ahead of the transmitter and correcting mistakes before they go out, including an automatic station ID to give you time to get ahead of the transmitter at the beginning
7. Built-in microprograms for calling, answering, and sending information (CALL, ANS, and INFO)
8. 12 User definable microprograms for customizing operational modes and messages, using a powerful microprogramming language
9. Full function microprogram editor
10. Word wrap display for easy reading of incoming and outgoing text.

Thus, for the more advanced user, the **MMX ZERO** can provide an excellent choice for enabling or enhancing their CW communication. 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. We offer the **Transceiver EMP Bundle** for this purpose**.**

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 **MMX** 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 **MMX** possible.

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 one year warranty for this unit is against defects in material and workmanship for a period of one year. The software is provided "as is." Occasional software updates will be provided, but due to the triple processor architecture, the MMX must be returned to the factory for software updates.

2.1 Important Notice: Things Not Covered by the Warranty

2. **Carefully align and plug connectors - do not force**. The side mounted USB and audio connectors are mounted on the surface of the front 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. Our upgraded cabinet holds the connectors tightly, and should prevent any accidental damage.

3. **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.**

4. **Take care how you run the keyboard cable**. The keyboard cable can act as an antenna and pick up RF from your 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.

2.2 The Elephant(s) In The Room

The Big Elephant: 1300 Hz. Yup, you heard that right. Or maybe you didn't hear it if you have hearing problems. The ZERO does NOT decode your favorite 600 or 650 Hz tone on CW. It operates at about twice the usual frequency This can be hard to get used to. In addition, setting up your transceiver for 1300 Hz operation is different than

you expect. We have a blog on how to do it for an IC-7300 on the website. Locate the Blog menu item along the top of the home page, and click. You will see several blogs. Scroll until you see the IC-7300 image. That's the one. Here is the link if you are reading this on PDF:

https://www.preppcomm.com/blogs/news/setting-up-an-icom-7300-for-1300-hz-tones

Follow this example for your own transceiver. And order our **External Transceiver Cable Assembly** if you have not done so already. It will make connecting the **ZERO** to your rig easy, and look great in your shack. A **MMX/DMX Stand** is also a good choice as it holds the **ZERO** at 45° so it is easier to read.

The Second Elephant in the room is ... experience. While we may Morse easy, we don't make learning CW easy. CW is a lot of things besides just the code. CW is about frequencies, methods, operating styles, abbreviations, and much more. Once you learn it, you can quickly tune around the band, and know when you are receiving a fellow amateur or hearing a digital transmission that is decoded as gibberish. Also there is some expertise you will gain over time on adjusting the decoder gain, and the receiver 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! Patience and practice are very important.

One other thing: this one messed me up when I started. There are stations on there that are actually transmitting what appears to be random junk. It is not random junk, it is data, sent via Morse Code. Best way to start is to call CQ or listen for someone calling CQ. Don't be fooled into thinking the decoder is not working by a data station!

If you have a buddy across town, or across the valley that does CW, 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 **MMX, ZERO** or **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

You can also get help from the PreppComm Community at:

http://PreppComm.mn.co

3. Getting Acquainted

Figure 1 is a labeled diagram of the **MMX ZERO**:
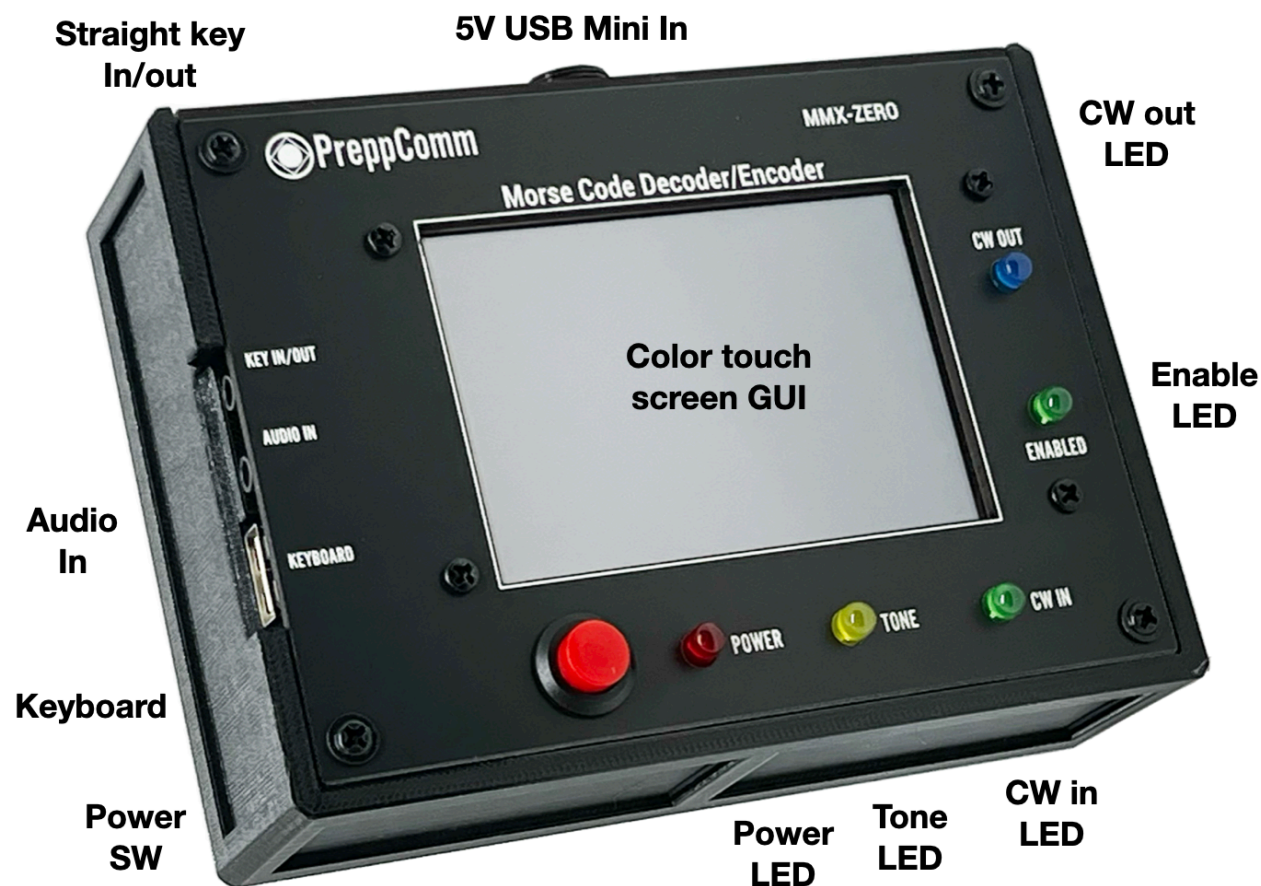
FIGURE 1. MMX ZERO

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 **MMX**.

CW Out LED
This blue LED displays the outgoing Morse code when the transceiver is in transmit mode (either with or without the internal transceiver engaged).

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 frequency setting when transmitting..

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 signal processors, 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 signal processing system. This LED is normally flickering from noise, as well as lighting brightly during a strong Morse 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.

5V USB Mini Power Connector
The ZERO is powered by 5V via an included USB cable. The USB A plug can be powered from a laptop, a computer, an outlet strip with a USB output, or a wall plug with USB outputs. Total current draw is about 0.25A.

5. Setup with your External Transceiver

Connecting up to your base station or portable transceiver is quite simple, especially if you use our External Transceiver Cable Assembly, shown in Figure 2 below
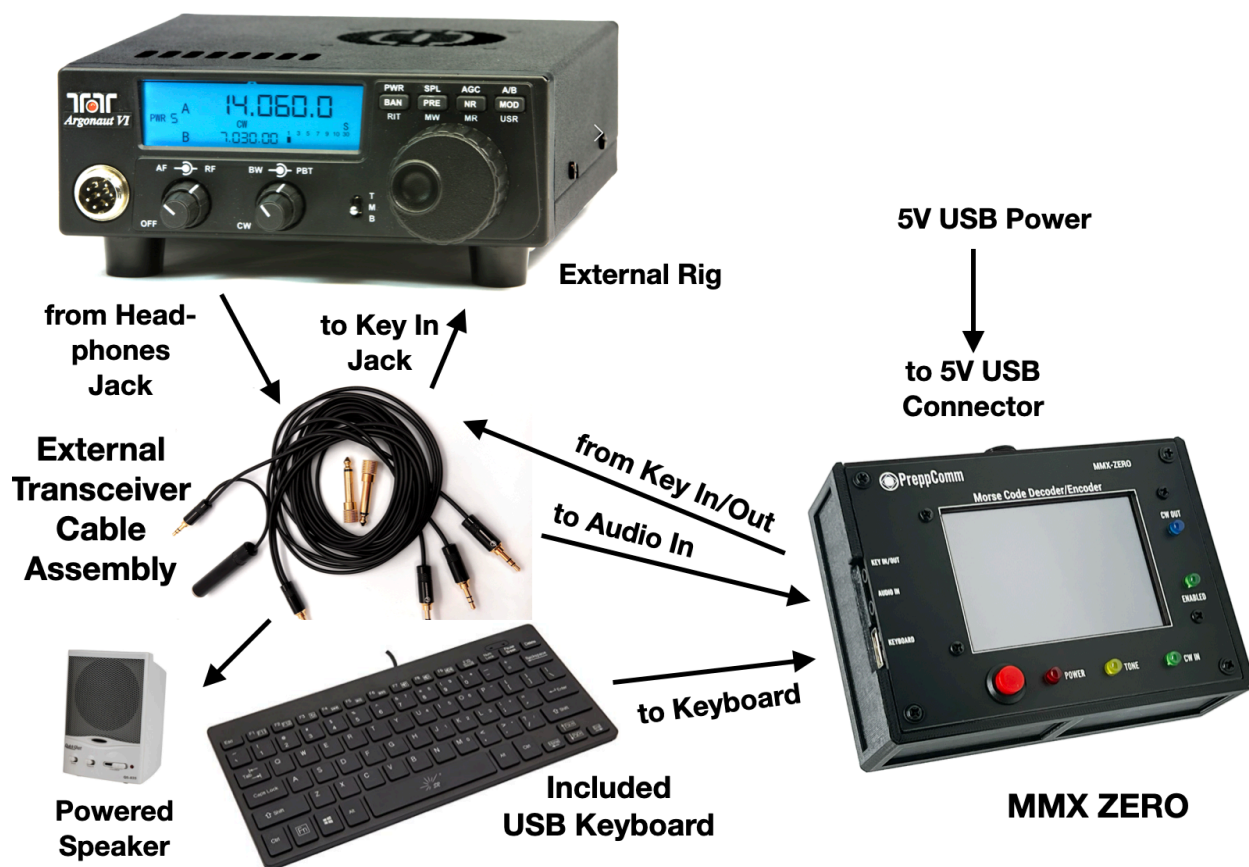
FIGURE 2: EXTERNAL MODE SETUP FOR MMX

In the diagram above, we use the **External Transceiver Cable Assembly** accessory, available on our website. This cable set has everything you need to connect the two 3.5mm jacks (KEY IN/OUT, AUDIO IN) on the MMX to your external rig (KEY IN, HEADPHONE OUT) as well as a plug for a powered speaker and an external straight key, if you so desire. If you wish to use paddles or an iambic paddle, an external keyer that converts from those dual-contact formats into pure Morse is required.

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 3:

FIGURE 3: SPLASH SCREEN

Note that this is the same screen as the multi-band MMX, as it is really the same product, with the transceiver removed. This screen will be present for a few seconds, and will then be followed by the Main Screen, shown in Figure 4. Each item on this screen is described below.
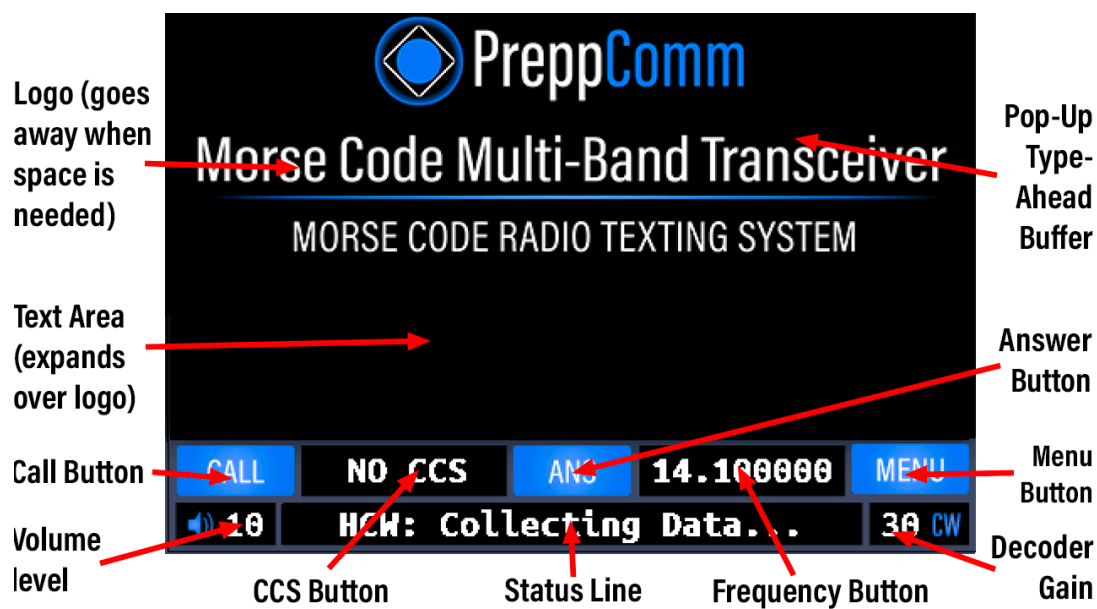
FIGURE 4: MAIN SCREEN

At some point, perhaps we will replace the screen graphics on the ZERO so it is different from the multi-band MMX. But for now, they have the same graphic..

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 **light 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 lines. 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. Finally, the Status Line is also a button. Pressing it turns the black background blue (while being pressed) as visual feedback, and restarts the decoder if you are in Receive Mode. This allows decoding with no keyboard, as decoder restarts can be done from the touch screen.

Decoder Gain

The decoder gain is set to 30 in Figure 5. The range is from 0 to 40. Keyboard commands < and > are used to control this value.

Volume Level

Since there is no transceiver inside, there is also no audio output, and thus, this control is not functional on the **ZERO**.

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 (Calling CQ). 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,

Page 16

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 will always display **External**, due to the lack of a built-in transceiver. Because of that, pressing it does nothing, so in effect, it is not functional in the **ZERO**.

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 (not shown)

A pop-up "window" for typing your message ahead of the transmitter 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 **MMX**. Figure 5 is the screen displayed when the **MENU** Button is pressed. Graphics and firmware versions are shown at bottom left and right corners (not shown here):
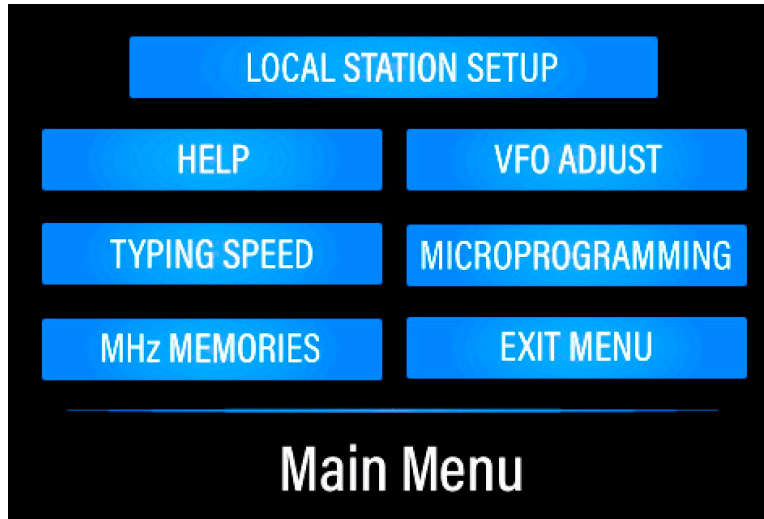
FIGURE 5: MAIN MENU SCREEN

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 **MMX** to suit your preferences.

VFO ADJUST
As there is no VFO in the **ZERO**, this button will give you a "External Mode" warning, rather than the VFO setting screen.

TYPING SPEED
The default typing speed is set to the maximum MMX 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 badly 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
Frequency memories are not useful or functional without a VFO to send them to, so this button is also not functional.

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 6 below, to collect your call letters:
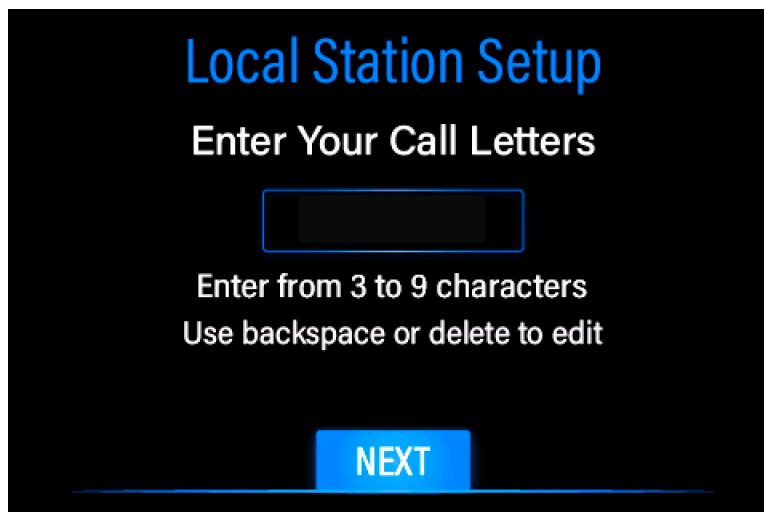
FIGURE 6: STORING YOUR CALL LETTERS

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 to capitalize. 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 MMX 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
5. Antenna Type
6. License Level

This information is used to create the INFO microprogram. 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 7 below:
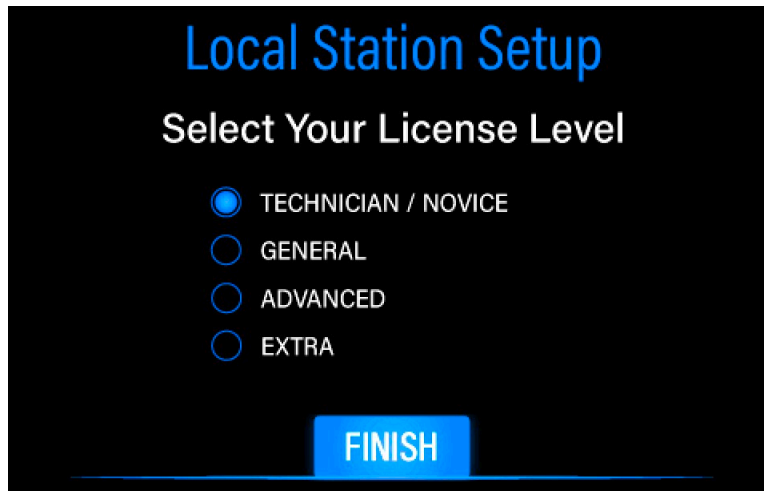
FIGURE 7 LICENSE LEVEL ENTRY SCREEN

Since you are using an external transceiver, this function cannot limit the frequency that you set on that rig, so it is not functional. Any value will do. Press **FINISH** to return to the main screen.

7.2 Typing Speed

This function will bring up the following screen, shown in Figure 8 below:
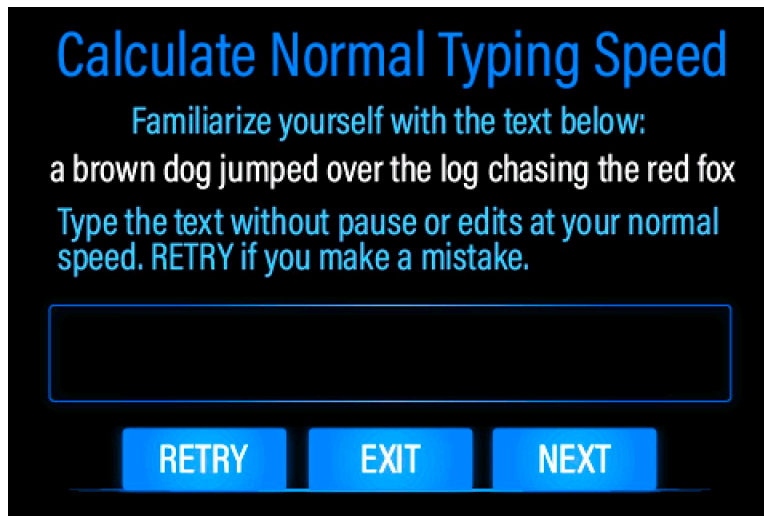
FIGURE 8: TYPING SPEED TEST

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 9 on the next page. This screen pops up automatically when you finish the sentence.
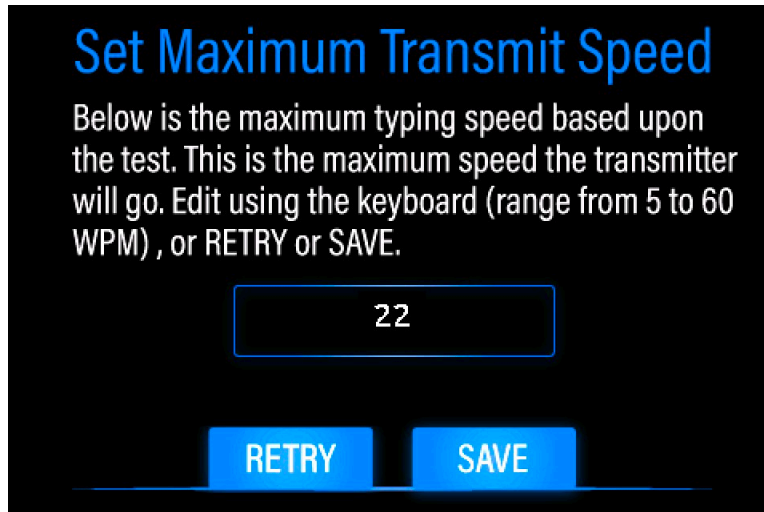
FIGURE 9: SETTING THE MAXIMUM TRANSMITTER SPEED

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, using BACKSPACE to delete 60 first.

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!

Two other factors: this is typing off of a screen, not thinking about what to say, and not editing mistakes. Both of these are taken into consideration, in addition to the "typing fast" issue above, when calculating the suggested maximum transmitter speed.

I had to go back and reduce my number. I ended up around 22. You can simply edit the number on the Set screen from the keyboard. 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.

7.3 Microprogramming

Another term for Microprograms that is often used is macros. We use the term microprograms rather than macros because the programming language is so much more powerful than any amateur radio macro 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 MMX 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 or jumped to from the main 10 functions. Microprograms 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:
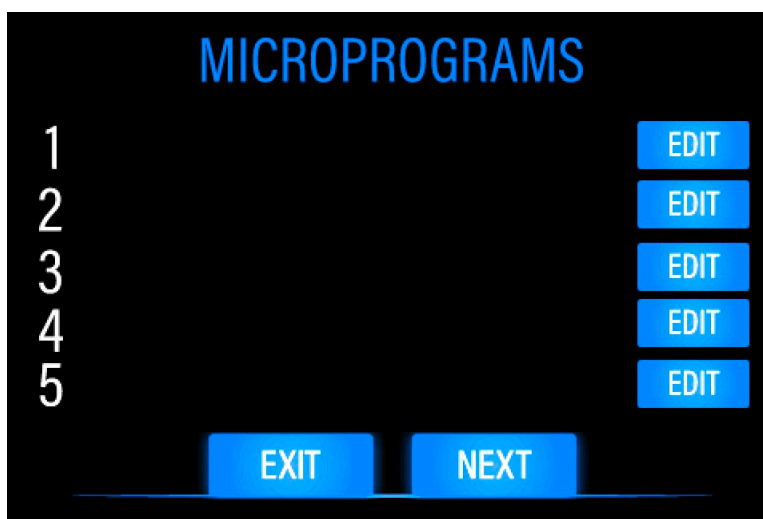
FIGURE 11: PAGE 1 OF MICROPROGRAMS LIST

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 .

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 from Receive Mode ≠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:
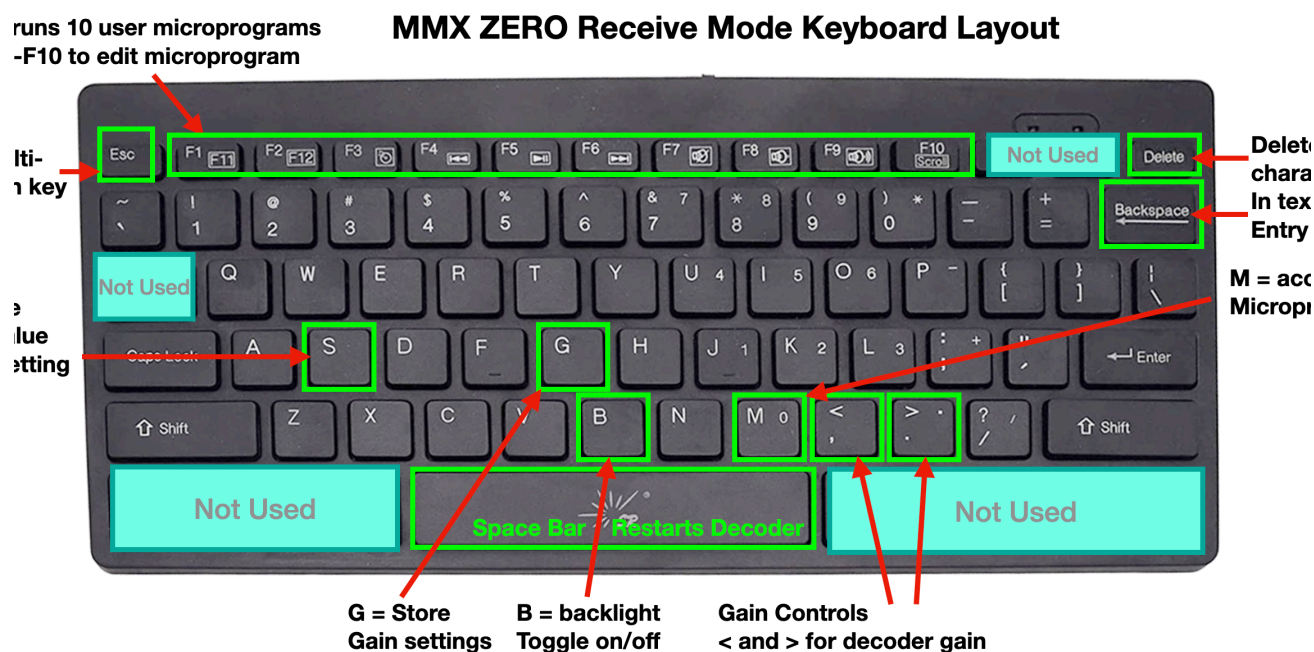
FIGURE 13: RECEIVE MODE KEYBOARD COMMAND LAYOUT

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 **MMX.**

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. However, in future enhancements, a shift-letter may be detected for control purposes.

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.

ESC	Multi-Function Key (Section 9)
< (Shift-,)	Reduces decoder gain (min = 0)
> (Shift-.)	Increases decoder gain (max = 40)
space bar	Special cases use (described below)
delete	Delete a previously entered character
G	Save current gain settings to memory for next boot
M	Shortcut to the Microprogram Lists
S	Saves WPM in CALL
B	Backlight on/off toggle
backspace	Delete a previously entered character
F1 - F12	Execute a stored user-defined microprogram
shift-F1 - shift-F12	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 clear the decoder cache, and activate the **speed and hand algorithms**. This requires a short period of capture, followed by an analysis of the data. The resulting speed (WPM) and 5 hand variables are then applied, and the captured data is decoded and spit out to the screen. This all happens during the "**HCW: Collecting Data...**" Period. The status then changes to "**DECODING HCW @ NN WPM**", where NN represents the speed algorithm result. Decoding then continues real time using the **tracking algorithm**. It should be obvious from this description that restarting the decoder on every NEW CW stream is necessary for good decoder performance. **Always restart the decoder for a new signal!**

ESC: Multi-Functoin key: This key has many functions.. One of them is to switch between the two decoder modes, referred to as HCW mode (human-generated CW) and CTX mode (computer texting mode). 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 as 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), ESC will 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. It is best to use the B key to turn on the backlight, as the ESC key will also change to decoder modes when waking up the display.

8.3 Gain Control Keys

< > 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. 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, for a much more rapid change in volume setting. You will 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 revised starting value in memory..

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, Typing Speed Setting, 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).

8.8 Function keys

The function keys F1 to F10 and Shift-F1 to Shift-F10 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. Like with the number keys, F1 to F10 execute a microprogram, and shift-F1 to Shift-F10 open the microprogram editor for the specific function key. Note that F11 and F12 are not available on our standard keyboard (they are functional but unavailable with this keyboard). However, using the jump command, they can be appended for long text strings to any of the first 10 functions.

8.9 Miscellaneous Receive Mode Controls

There are six additional single-key functions shown on the keyboard map. Each is described in the list below in some detail:

1. **G (Gain) Key**: Pressing the **G** key saves the headphone gain setting and the decoder setting currently shown at the bottom left and right of the main screen to memory. This sets them as the power-up default. You can change this at any time by changing the values and pressing G again.

2. **S (Save) Key**: This key function only is active as you are typing in a WPM speed, either in CALL or ANS. Normally, pressing SPACE uses the default speed. If you want a different speed, you type two digits in, from 5 to 60 and press space to enter. However, pressing the **S** key rather than space will save the entered speed as your new speed default, as well as loading the speed for use. The default will survive power cycles.

3. **M: (Microprograms) Key**: Pressing the **M** key gives you immediate access to the microprogramming list, starting with the first five microprograms. Thus, this is a shortcut key. Note that you can directly access the editor for a specific function key by pressing shift-Fn.

4. **B: (Backlight) Key**: A relatively new key, the **B** key is the backlight toggle, It is the best way to exit the power saving "backlight off" state, as it does not cause any other changes, as pressing ESC does. The 3-minute no-activity auto-off function is still operational, and will again turn off the screen 3 minutes after you turn it back on, if no activity on the screen occurs. Activity includes change values, such as gains, decoder inputs, or change of status. The key is a toggle, so it can also be used to turn off the screen backlight at any time

8.10 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:

ESC	Abort Transmit Mode
~ (tilde)	Insert the INFO microprogram
F1 - F12	Insert User microprogram 1 - 12
Morse Character	Sent to Type-Ahead Buffer
SET CALL WPM = ? or SET ANS WPM = ? (Status Line)	
0-9	Enter speed from 5 to 60 WPM
Space	Accept entered speed
S	Accept and Save as default speed
ANS WHO? (Status Line)	
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 is seldom used when transmitting, except in an emergency, when you can't wait for the normal exit from **CALL** or **ANS**wer mode, or if your microprogram

running amok. **ESC** will force instant exit from 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 - F10 key
These keys are used to run a microprogram. This could be a transmit/receive toggle, such as the four-character program "!(|)" for example, a station ID, such as " @ DE * ", or some other function, such as "#C" or "#A", which run the **CALL** or **ANS** microprograms without having to touch the screen. 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 speed the station you are answering is using, unless limited by your typing speed. **CALL** will use the speed previously used, but if none is set (first power up, or after a double **ESC**), it will request you set one. Pressing **SPACE** will set the factory default speed of 20 WPM, unless you have changed it using the S key described in **Section 8.9** above.

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. This can fool you sometimes if you forgot the limit you set, and you are trying to set a higher speed. No, it's not broken. It's applying the limit you set.

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 **MMX** has two modes: **HCW** (Human CW) and **CTX** (Computer Texting). HCW is used for most situations, unless you are communicating with other hams that have an **MMX or DMS** transceiver, or are using a computer to send code. CTX will almost always work better in those cases, however it has only one speed - 30 WPM, so the type-ahead buffer is of limited use for slow typist. HCW has two sub-modes:

1. **Speed and Hand Analysis**: This is the first stage of decode for a given CW stream. It is activated by pressing the SPACE bar when you are starting to decode a new station. If you do not do this, your decoder will fare poorly, as it will remain in the Speed Tracking mode unless things are bad enough to trigger an automatic restart. Pressing the SPACE bar will always clear the memory and cache for the Speed and Hand Analysis system, and begin collecting data for a new analysis. During the collection phase, there is no output from the decoder. **Note that the speed and hand algorithm run once** after enough data is collected This is at the point where the status display changes to show the decoder speed. Restart will create the data collection sequence followed by the algorithm run each time.
2. **Speed Tracking**: This is the second stage of decode, and begins when the decoder analyzes the collected data, sets the statistics values for decode, and outputs the subsequent decode of the collected data using the statistics values for hand and speed. Subsequently, the decoder operates real time, and makes minor adjustments to the speed and to hand values to track slow and minor changes over time.

It is **CRITICALLY IMPORTANT** to utilize the SPACE bar to restart the decoder each time you locate a new CW stream to wish to decode. Not doing that ensures a decoder performance similar to other decoders on the market, which do not analyze for hand, and do a poor job with speed as well.

The decode mode you are in is controlled by the **ESC** key when in Receive Mode. The default mode is HCW when you power up. Note that switching between modes will clear the transmit WPM speed, so you will have to reset it on the next transmission, unless set by the decoder. Also note that using ESC to turn on a dark screen also switches to CTX mode, so it is better to use the B key.

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 tested it on-air at those speeds: I 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 analysis for a

short period of time before processing it and 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 senders hand, 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 and hand parameters. 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 in Speed Tracking mode. Once the speed and statistics of the senders hand 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 ESSENTIAL to restart the HCW decoder. This is done simply by pressing the **SPACE bar**. If you are using External Mode, you can also press the status line as an alternative, allowing keyboard free use for decoding only.

Pressing the SPACE BAR will return the status line to **HCW: Collecting Data...** until the new signal is analyzed. 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 and hand.

At one point, it was planned to automatically 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 restart that is automated.

Finally, the HCW decoder can also automatically 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. The advantage of this is a substantially lower error rate when listening to another CTX station using the same mode. CTX operates at a single speed which is fairly high for many typists, and thus may limit the value of the type-ahead buffer.

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 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. The ESC key has other functions, as well, described elsewhere in this document.

9.4 Tuning a Station

To better understand how best to tune in a station, a diagram should help. Figure 15 below 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
4. The output is fed to the HCW decoder pre-processor, which figures out the speed of the CW as well as 5 major parameters for the senders hand. This is followed by the actual CW to text conversion step.
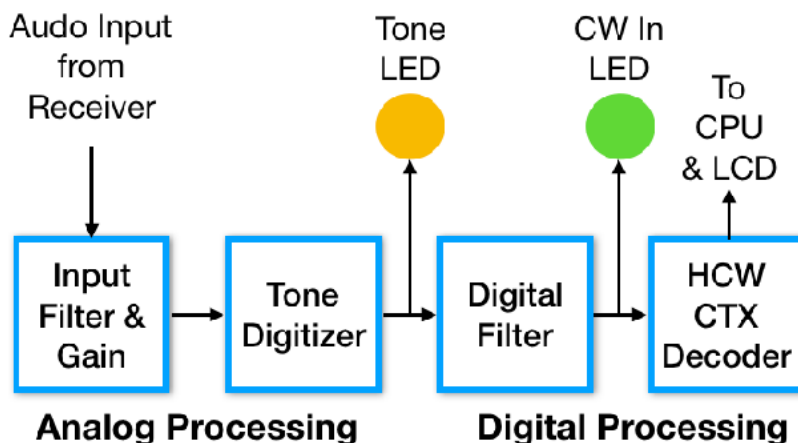
FIGURE 15: THE MMX DECODER BLOCK DIAGRAM

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 MMX 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:

1. Scan the band for an appropriate signal. It is at the right frequency if the CW IN LED is blinking. If you hear the CW clearly, but no CW IN LED, it is because you are tuning with your ear, rather than with the CW IN LED.
2. If the signal is weak, adjust your RF gain. If strong, turn down the gain
3. Tune back and forth to get a feel for the center frequency where the LED lights, and set it there.
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.

9.5 Setting up your transceiver

There has been so much confusion on how to set up an external transceiver so the **ZERO**, or the **DMX-40** or **MMX** multi-band in External Mode can receive the proper 1300 Hz signals, and still have the transmitter transmit on the correct frequency - the same frequency of the station that is being decoded.

I don't have a wall full of transceivers, as some folks do, I have only my trusty old Icom 7300. I say old, because I bought it when it first came out at $1500. I recognized even before it was on sale from the information made available that it was a breakthrough product, and I was right. What an amazing day in amateur radio history! Below is a step-by-step method for setting up your transceiver. The process will need to be adjusted for your specific case, as terminology and arrangement of functions are different between transceiver companies, but the functions referred to should be available on your transceiver, unless it is a very old unit.

Anyway, let's start out by first setting the CW PITCH or sidetone as high as it will go. To do this for the IC-7300, go to CW mode, and then, using the Multi knob/button, press once for the display shown on the right. Touch CW PITCH to select. Turn the Multi

knob until the value reaches 900 Hz. You should see something like the view on the right. Press Multi to save the value.

Now let's consider what this setting is doing. For many, it is a mystery. Basically, to turn a carrier wave - which has no sound in itself - into a tone, we do this using a method called creating a beat frequency. A beat frequency is the difference frequency between two signals. So, if you have your dial set at 7.030.000, or 7,030,000 Hz, or 7.03 MHz, to get a 600 Hz beat frequency audio tone, you must use another frequency at 7.030.600. Note that 7.029.400, which is 600 Hz lower, will also generate a 600 Hz beat frequency tone. So in effect, this 600 Hz higher frequency is your VFO or receive frequency, and the displayed frequency is your transmitter frequency. Thought of in another way, a zero beat will occur 600 Hz higher on the dial, indicating that your receiver is, in effect, operating at 600 Hz higher in frequency than your transmitter, so you can hear the other station.
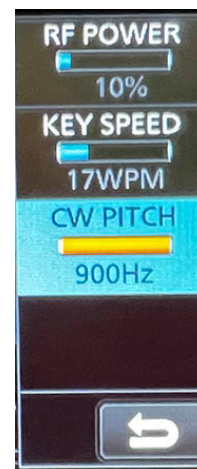
Thus, the CW PITCH control is an offset between your transmitter frequency and a VFO frequency generated within the transceiver. You can tune up or down, and this tone will go up and down depending on the distance between an incoming carrier and the VFO, which is offset from the displayed frequency by this setting, now 900 Hz. The displayed frequency is both your transmitter frequency AND the station's frequency you are listening to,

The VFO frequency is the receive frequency, and it is the frequency we "insert" into the RF mixer to generate an audio signal. To receive 1300 Hz tones for a given frequency, we actually set the receiver frequency (VFO) 1300 Hz above the displayed frequency, and turn off the 1300 Hz offset when transmitting. We have only achieved 900 Hz so far, and we will add the extra 400 Hz later,

Step 2: Selecting a Filter

There are 3 built-in filters, and the one you are currently using is shown to the right of the blue CW button on the 7300 LCD screen, at the top of the screen (see Figure 16 on the next page). Here, we see a gray button labeled FIL1, which is Filter 1. Pressing briefly switches between filters, but pressing and holding brings up the settings for the filter. In the figure, you see the effect of holding down FIL1 at the bottom of the screen, an area called FILTER (CW). Note that you can use any of the 3 filters for this purpose.
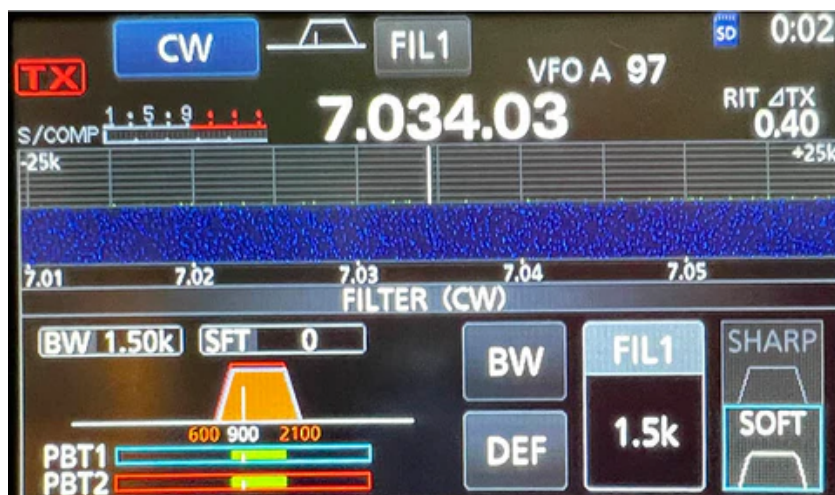
FIGURE 16: 7300 FILTER SETTING

Note in the bottom left, the filter "center" is 900 Hz, as you have set in the CW PITCH, and the bandwidth in this example happens to be set to 1.5K, shown below FIL1 between the DEF button and SOFT button. That is 1500 Hz, and the filter is from 600 to 2100 Hz as shown on the left. SOFT is a good setting - SHARP will introduce too much ringing and other artifacts into the signal, interfering with the signal processing in the **ZERO**.

Step 3. Selecting Bandwidth Edit

Press the BW button to the left of FIL1, as shown in Figure 17 below:

FIGURE 17: SELECTING BW

The BW button and the box below FIL1 will light up, and the label FIL1 will turn darker blue. This indicates that the filter bandwidth is ready to be adjusted.

Step 4: Adjust the Bandwidth of the Selected Filter

Using the main tuning knob, adjust the bandwidth so that the upper number is at least 1500 Hz, and better yet, 1700 Hz to give plenty of unfiltered content for the **ZERO** signal processor. Having a filter too tight can cause issues and reduce quality of the decode output.

In Figure 18 below, you can see we adjusted the bandwidth to 1.1K or 1100 Hz, resulting in a filter range of 600 to 1700 Hz. This should work nicely with the DMX-40 or MMX decoder.

FIGURE 18: FILTER SET

Step 5: Adding 400 Hz to 900 Hz

You may have noticed that RIT was already set to 0.40, but the photo here shows you the setting again, top right on the waterfall display. Use the Multi knob to adjust this to 0.40, which is 0.4 KHz or 400 Hz.

Now what is actually happening when you set RIT to 400 Hz? It moves the receiving frequency up an additional 400 Hz from the transmitter and displayed frequency. Since the offset is already 900 Hz (generated by whatever means is used in the 7300), the tone for a tuned in signal that was at 900 Hz will rise to 1300 Hz - without touching the tuning knob! But the transmitting frequency, and the display frequency will stay the same. So no changes are needed between transmit and receive!

For PreppComm products, more than 900 Hz is needed, so we add in 400 more with RIT.

If you could turn off CW PITCH, or make it zero, then RIT would need to be 1.3, or 1300 Hz.

You don't need to move the dial between receive and transmit. If you need to adjust for a station that is not exactly on your frequency - i.e., the tone is not quite right - adjust with your RIT control. It has steps of 10 Hz, and you can do fine adjustments.

In fact, your DMX-40 or MMX decoder is only APPROXIMATELY 1300 Hz. It should be very close, but it can be 20-30 Hz off in either direction, and can drift as the temperature changes inside the decoder. So a little experimenting over time, you may discover that the correct setting for your PreppComm decoder is actually 0.42 or 0.39, for example. However, each station you are communicating with, depending on how close they are transmitting to your frequency, may require you to make a fine adjustment for best results. This should not be more than 50-60 Hz, for a sloppy operator. Always return to the default best setting between QSO's.

Other Transceivers

Other transceiver owners, you will have to translate these instructions for your unit. Older transceivers may not have an RIT control, or even no adjustment of tone, so there may be situations where your transceiver cannot operate with 1300 Hz tones. Filter adjustment is also a potential issue. The Icom 7300 and other similar modern transceivers all have these kinds of settings, often called by a different name.

11. CALLING

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

5. **General Calling:** calling for anyone to answer, known as "calling CQ"
6. **Directed Calling**: calling for a specific station to answer, usually on a schedule
7. **Validate-Answer**: Validating Two-Way communication connection when answering a call
8. **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.

11.2 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. 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 u̲nder the decoder speed will ALWAYS override the transmit speed̲ if it is defined after running the speed algorithm. 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 restarted, you will revert to 15 WPM until and unless the decoder establishes the sender speed again.

Note that the decoder speed is cleared by restarting the decoder, but that d̲oes not clear any transmit speed you have set̲. This is why when you restart 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 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 on the next page:

Inactive   **Active**   **Ending**
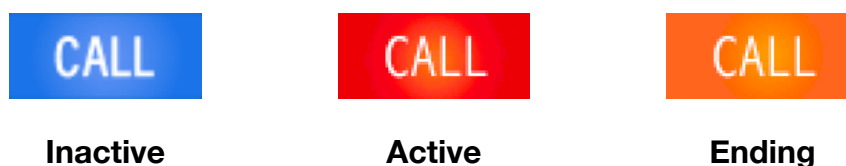
FIGURE 16: CALL BUTTON STATES

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. The "#C" microprogram can be used to press the **CALL** button.

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:
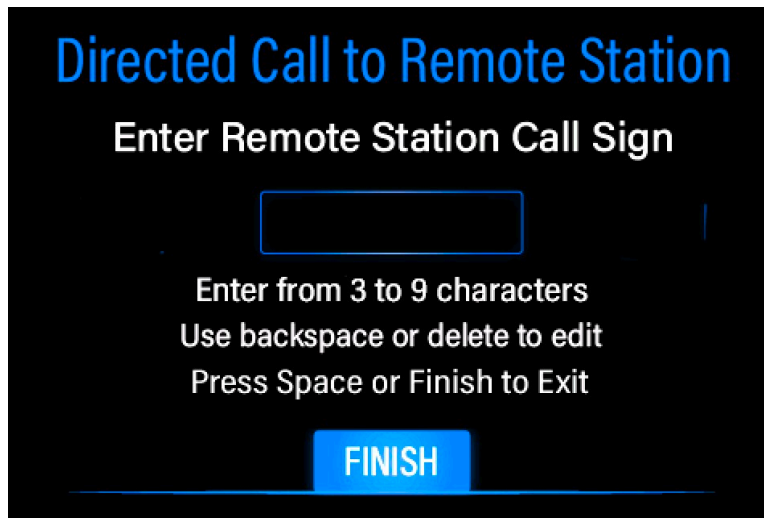
FIGURE 17: CAPTURING A CALL SIGN MANUALLY

Enter your friend's call sign, and press **FINISH**, or the **SPACE bar**. The NO CCS Button with 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 **MMX** 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.

The fourth Way
We said there were three ways to set CCS, but in fact there is another way, the {callsign} method in a microprogram.

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**, them 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 **TONE** 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 **MMX** 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:

ANS	ANS	ANS
Inactive	**Active**	**Ending**

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 <u>when you have tuned into a station calling CQ</u>, 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 **MMX** 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:
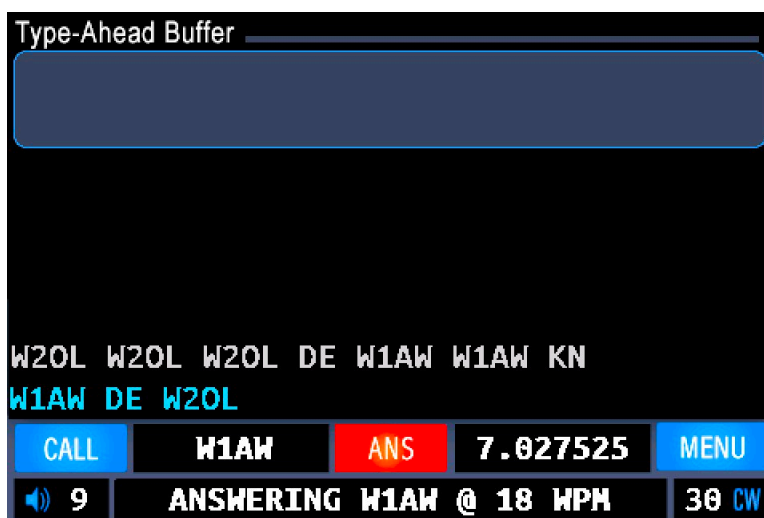
FIGURE 19A: POP-UP TYPE-AHEAD BUFFER

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, 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 on the next page, we have you typing at a blinding speed, probably faster than you can actually
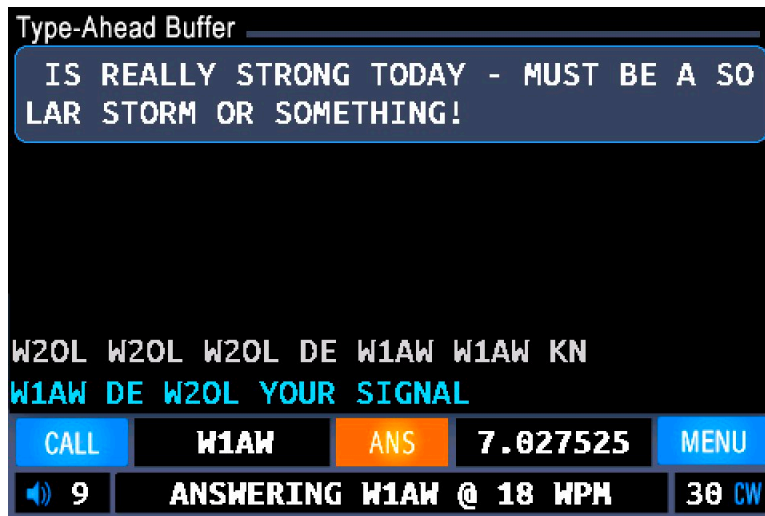
FIGURE 19B: TYPING DONE, TRANSMITTING AND THEN ENDING

type... but this is for the purpose of showing you a different state in the answering process.

Here, the 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 **MMX** computer that you have finished typing, so when it empties the buffer, it can transition immediately into the ending component, which then can transmit the station ID. 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 **MMX** 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 if you see errors in your typing, you can 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 MMX 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.

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.

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

The table below shows the pro-signs that the **MMX** 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 MMX 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 a s "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 **MMX** 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 **MMX** 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 **MMX**, you can actually use both **(** and **)** and they will see it correctly. :)

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

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 **MMX**. 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 MMX 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 **MMX** 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 **MMX**.

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 **MMX** 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 **MMX** 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, by using the #C or #A microprograms, 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 or Transmit 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.

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 **MMX** 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
0^	Sets counting variable to zero
^+	Increments the counting variable
^-	Decrements the counting variable
^?	Insert the counting variable
`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.

Character	Description of Operation
@	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.
)	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.

Character	Description of Operation
[...]	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	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.
&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.
=nn	Set transmit speed to nn, where nn is from 05 to 60, in words per minute (WPM). Use BEFORE ? or ! command.

Character	Description of Operation
<a%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.

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
?	Press CALL Button, enters transmit CALL mode
<	Test if CCS is defined. Assume Not for the example
	Delete <[@ @ @ @ DE * *]KN% and >
	What remains is [CQ CQ CQ CQ DE * *]K
[	Start the repeat

Character	Execution
CQ.. DE	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
K	Send K to encoder
(end)	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
!	Press ANS Button, enter transmit ANS mode
(	Enable keyboard input to type-ahead buffer
@	Insert CCS (must be defined, because if not, it will ask ANSWER WHO? Sends the CCS and a single space to the encoder
<	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
DE	Send DE and spaces to encoder
*	Insert user call sign with surrounding spaces
|	Enable the type-ahead buffer output, wait for empty encoder, buffer and a second function key or ANS Button press.
)	Disables keyboard input to buffer, waits until buffer and encoder is empty (short wait because preceded by a | character)
BACK 2U	Send BACK 2U and spaces to encoder
@	Insert CCS
DE	Send DE to encoder with spaces

Character	Execution
*	Insert user call sign with spaces
KN	Send KN to encoder
(end)	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 MMX 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:
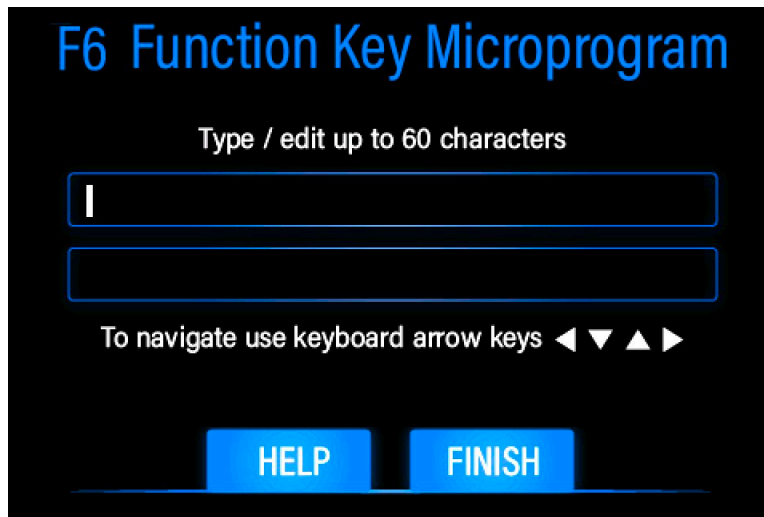

FIGURE 22A: MICROPROGRAM EDITOR WITH NO MICROPROGRAM

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 vise versa. They will only allow movement within the microprogram, however, so if there is only one line of text, up arrow and down arrow 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.
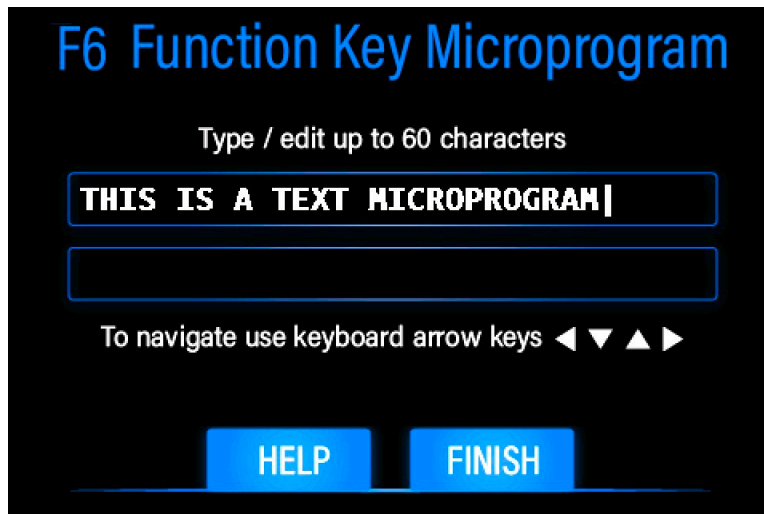
FIGURE 22B: TEXT EDITOR AFTER TYPING IN CHARACTERS

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:
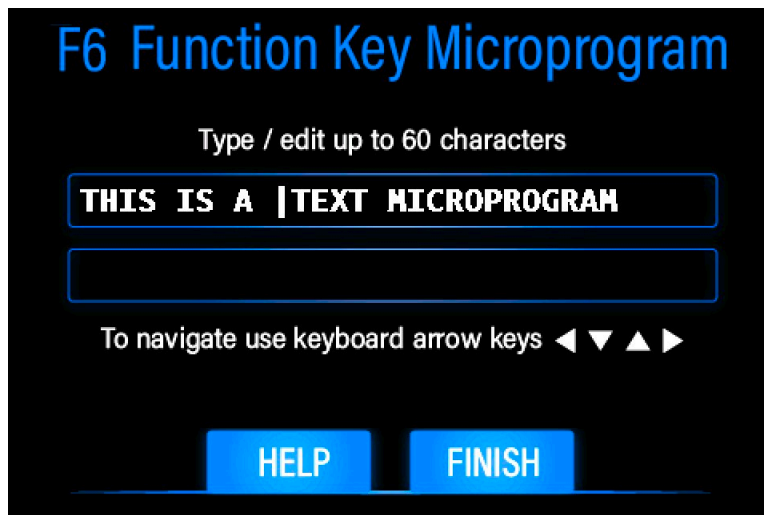
FIGURE 22C: READY FOR AN INSERT

Since the display does not have a "cursor" function, the MMX 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 **MMX** 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!
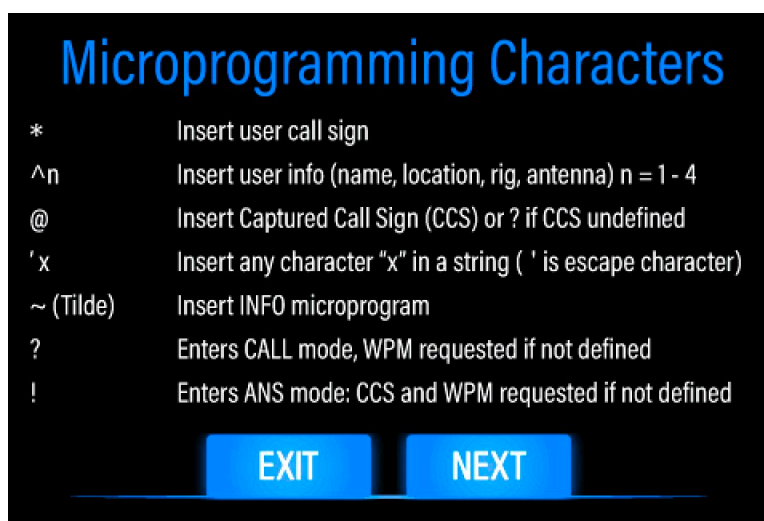
FIGURE 23A: PAGE 1 OF RPL CONTROL COMMAND HELP

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 23C: PAGE 3 OF RPL CONTROL COMMAND HELP

The new command additions for the counting variable are not shown in the microprogramming help screens.

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.

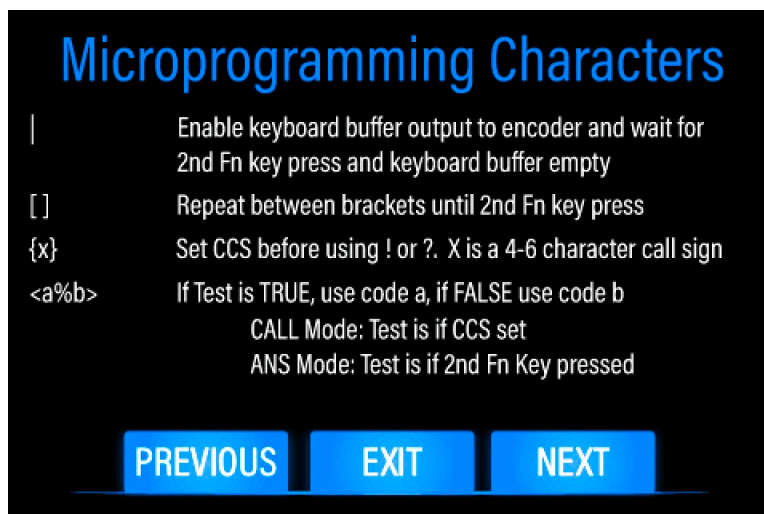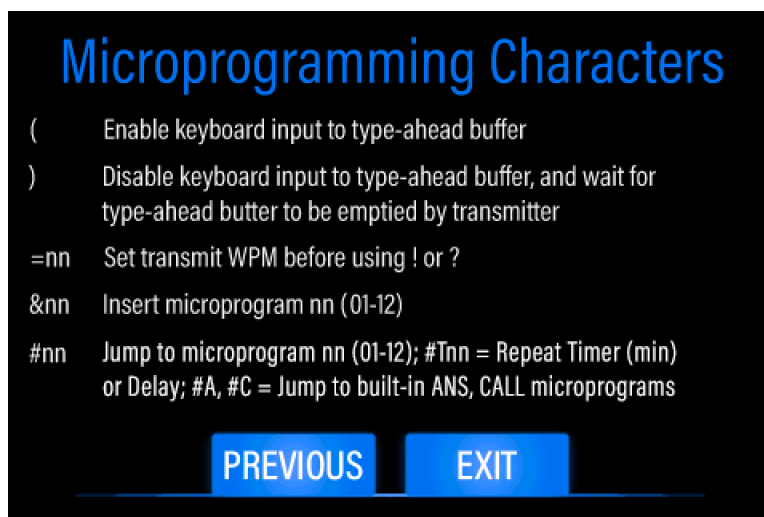2. F2 = "**?ID: * WITH @**" This microprogram does a quick station ID with your call letters and the other station's call letters. Nice for a long QSO.

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 **MMX** 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!

18.10 The Starter Set

There are four microprograms we recommend you start with. Here they are:

1. F1: #C. This assigns the CALL function to F1
2. F2: #A: This assigns the ANSwer function to F2

3. F3: !(|): This is a transmitter on/off function. You press F3, type away (into the type-ahead buffer), press F3 again. It will turn off the transmitter after everything has been sent out of the type-ahead buffer
4. F4: _@ DE *_ The _ are showing were spaces are included, or " @ DE * ". Pressing F4 sends a station ID during a long transmission.

Those four are great starters and #3 is very useful for breaking back and forth quickly w/o station ID's.

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 **MMX** 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.

Unfortunately, the printable summary of the RPL commands do not include the four counting variable commands, FYI..

MMX ZERO Receive Mode Keyboard Layout

F1-F10 runs 10 user microprograms
Shift F1-F10 to edit microprogram

ESC multi-
Function key

S = Save
WPM value
When setting
In CALL

G = Store
Gain settings

B = backlight
Toggle on/off

Gain Controls
< and > for decoder gain

M = access
Microprograms

Delete
character
In text
Entry

Not Used

Not Used

Not Used

Not Used

Space Bar / Restarts Decoder

Page 74
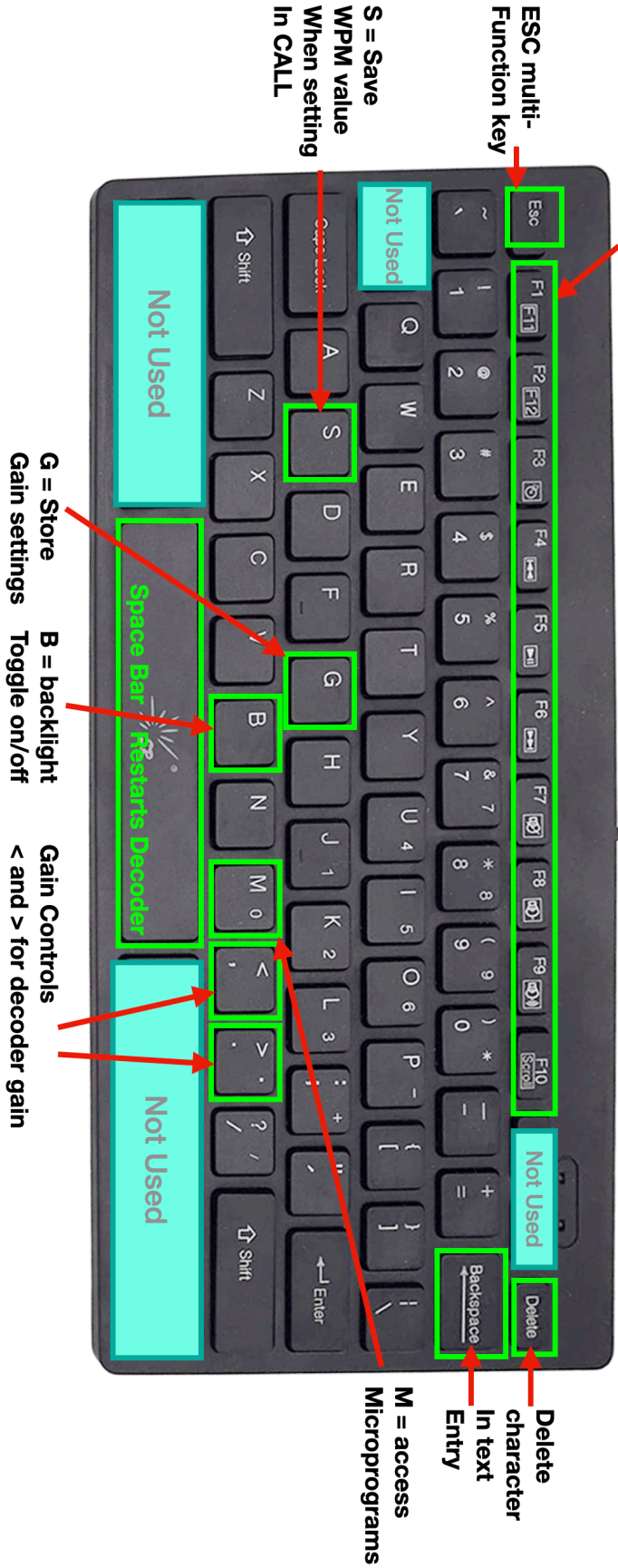
Transmit Mode Keyboard Layout

F1-F10 runs 10 user microprograms

Abort Transmit

Send INFO (tilde ~ key)

Save entered Speed as Default

Not Used

Not Used

Space Bar enters default WPM when requested

Not Used

Not Used

0-9 keys enter WPM when requested

Delete character In text Entry

Backspace In text Entry

Green keys are reserved control keys. Yellow keys are control in special circumstances. All other keys are text entry.

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.

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:

#A: This utilizes or executes the built-in ANSwer microprogram, for the same purpose as #C.

#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!

#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. |

Page 2

Table of RPL Commands

Page 3

Character	Description
&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.
˷	Insert the INFO macro. This does the same thing as &nn, but with the special default INFO macro.
{CCS}	Define the CCS
=nn	Set transmit speed to nn, where nn is from 05 to 60, in words per minute (WPM).
<a%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.

19.1. 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

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
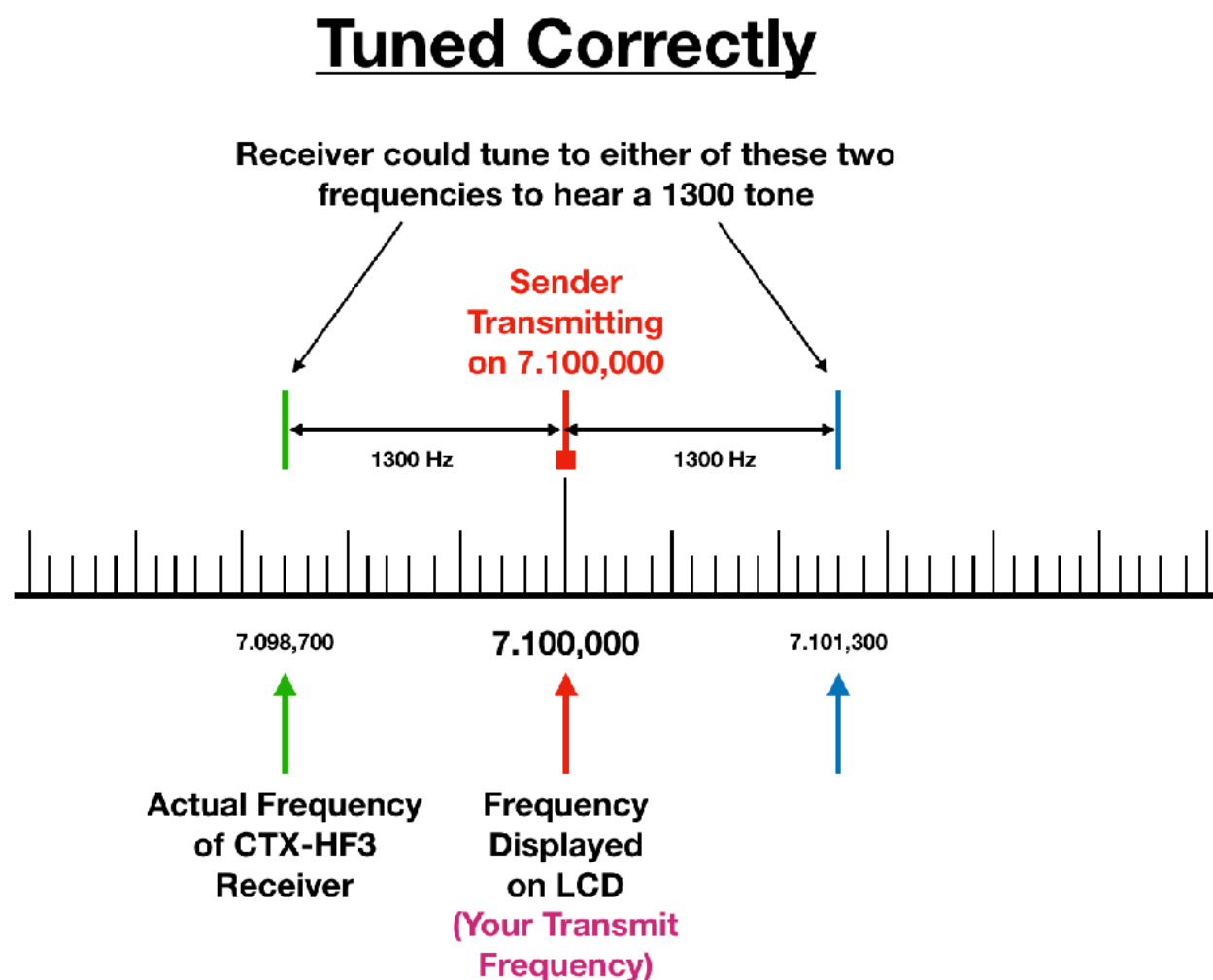XMTR--Transmitter
XYL--Wife
YL--Girlfriend ("Young Lady")
YRS--(Years)

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.
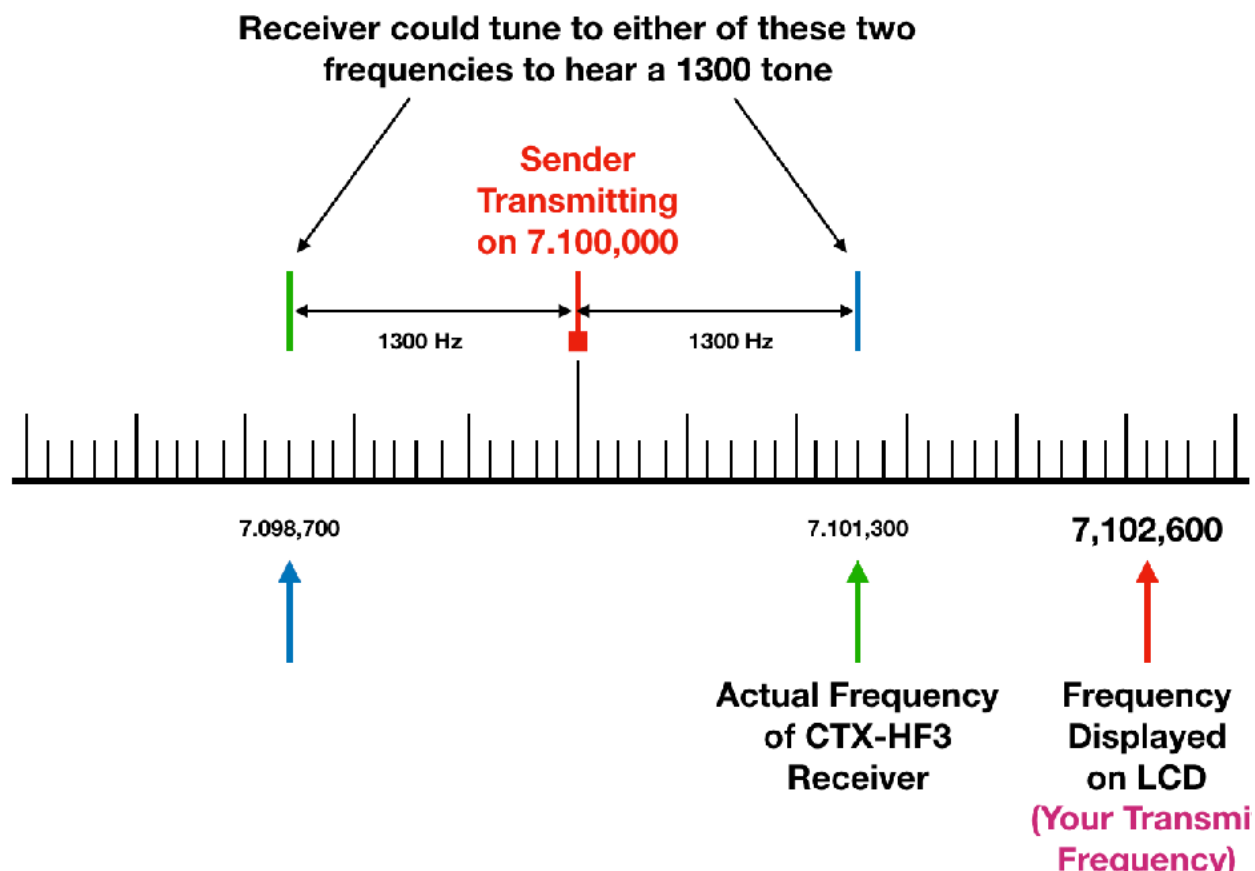
Below is a diagram of a properly tuned signal.

Tuned Correctly

Receiver could tune to either of these two frequencies to hear a 1300 tone

Sender Transmitting on 7.100,000

1300 Hz   1300 Hz

7.098,700   **7.100,000**   7.101,300

Actual Frequency of CTX-HF3 Receiver

Frequency Displayed on LCD
(Your Transmit Frequency)

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

Receiver could tune to either of these two frequencies to hear a 1300 tone

Sender Transmitting on 7.100,000

1300 Hz   1300 Hz

7.098,700   7.101,300   **7,102,600**

Actual Frequency of CTX-HF3 Receiver

Frequency Displayed on LCD
(Your Transmi
Frequency)

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.**

PreppComm
Amateur Radio

130 McGhee Road, Ste 220
Sandpoint, ID 83864

(208)-215-2585
preppcomm.com

Cover Image: iStockPhoto.com

22. Appendix IX: Alternate USB Keyboards

While it may seem reasonable, you cannot plug most USB keyboards into the USB connector on the MMX 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-sped 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 MMX 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 MMX, 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 ±50 Hz around 1300 Hz. As the amplitude increases, selectivity decreases to perhaps ±75 Hz, and maxes out at ±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.
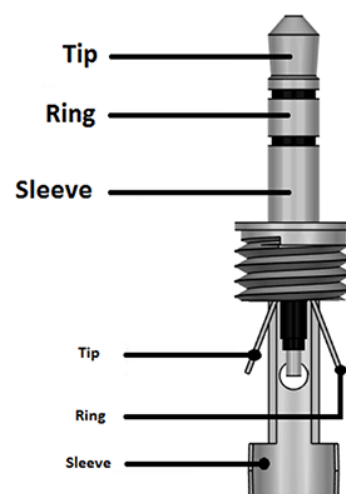
KEY IN/OUT: This jack is also a 3.5 mm audio jack, and requires a 3.5mm stereo plug as depicted in the image to the right. The **MMX** cable wiring is as follows:

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 MMX to lose it's mind, so to speak. Avoid problems by using shielded wire!