CMOS SUPER KEYER III

designed by:

KCØQ and NØII

marketed by:

Idiom Press
PO Box 1985
Grants Pass, OR 97528

TUTORIAL

Please Note: This tutorial presumes you wired the six memory buttons as per the construction article and in a row of six, with button #1 to the left end going to #6 at the right end. It also presumes that the monitor speaker and speed control are both installed.

GETTING STARTED

When you apply power to your new keyer for the first time, if everything is working properly the keyer will send a crisp "OK" in perfect Morse, telling you that it has run a built-in diagnostic routine, and has found everything to be in order.

However, one thing it does not know is the position of the speed control. To allow it to calibrate itself, turn the speed knob all the way clockwise and send a few dots, then turn it counterclockwise and send some more dots. That sets it up. The initial speed range of the keyer is set for 5-40 words per minute. Later we will learn how to change that if we want.

Now, send a little code from your paddle. Oops! The dot and dash sides are reversed from where you want them? (Most right handed operators prefer the left paddle for dots and the right paddle for dashes.) NO problem! Simply press buttons #2 and #5 down at the same time, and release. The keyer will send back in Morse a message - "RV", as in ReVerse. Now you will find that what was the dot side is now the dash side, and vice-versa. And it will stay that way until you reverse it again by pushing those two buttons again. Neat, huh? Now, adjust the speed to a comfortable setting.

You say you are used to a Curtis type "A" timing, or to a keyer with no dot or dash memories, and already you can't send code comfortably? Not to worry! We can fix that right now by using one of the emulations. (If you like the feel of the keyer already, as most do, skip the rest of this paragraph and the next.) To do so, first look at the "V" function table in the Operating Manual, in the EMULATION OPTIONS section. Select the "V" setting you want to try. Now, follow these directions exactly. Later we will explain them, but right now let's just do it. Let's suppose you want "V6", the Curtis emulation.

Press buttons #1 and #2 down simultaneously, then release them. The keyer will respond by sending a "F". Now, using your paddle, send "V6". There. You just did it, and now the keyer should feel comfortable. Easy, wasn't it? If you made a keying mistake, the keyer sent a raucous "raspberry." No problem - just start over again.

If you have used keyers before, you will instantly notice the fluid smooth timing as the CW rolls off. Now, decrease the speed of the keyer. See how the speed control is linear? OK, the keyer is hooked up and working, the dot side is where we want it. Now let's learn how to load a message into memory.

Your keyer is capable of storing long messages in each of six active memories. If you use only the six memories, you can store messages up to 255 characters in each memory. that is a lot of message! And if you choose to configure the memory into 18 messages (of which more later) you still have storage good for 85 characters mer message available. Now, let's store a message in memory. Press the far left button (#1) down, and hold it several seconds until you hear a tone. Then release the button. The keyer will send a "C". (This stands for Character mode. We'll get into that later, too.)

Begin sending your message, a word at a time. Let's load a message, "the quick brown fox". the first word is "the", so simply send "the". release the paddle. There. The keyer just sent a high- pitched Morse "I" to you. that means it has accepted the word, and has injected a word space. Now, send "quick" through the paddle, then stop again. The keyer will send another Morse "I". At this point, the keyer is prepared to wait as long as necessary for you to program in the next word. If you want, you can take time to go get a cup of coffee and return to the shack. It will still be waiting for the next word.

Let's go on and send "brown fax". Whoops. We wanted "brown FOX" didn't we? Do we have to start all over again? Heck no. Instead, send the international "I goofed" symbol - a stream of seven or more dots, and release the paddle. The keyer will send back to you "brown", telling you it has erased the incorrectly sent word (in this case "fax"), and backed up to the previous word, "brown". It sent the word "brown" to remind you where you are in the message, and it is ready for you to resume loading the message you want. So now send "fox".

OK, we've loaded the message. To close the message, simply press button #1 down momentarily and release it. Now, to play the message, press button #1 again briefly, and listen to the keyer send the message you programmed, complete with the correction.

Let's load another message, "jumped over the lazy dogs back" into the second memory, using button #2. Load the message in just like the first message. OK? Play it back to make sure you got it right.

Press button #1, release it, and immediately press button #2 and release. The two messages will play, one after the other. you could load your call in message #1, "AB1CD", and in message #2 "DE AB1CD". In message #3, you could put in "AR K". Then, by pressing buttons #2, #1, and #3 in that order, you would have chained or "queued" together a message "DE AB1CD AB1CD AR K". (You also have the option of NOT having multiple button presses give queued messages, but rather stop one message and begin another. We will get into that later.) You can stop a message being sent at any time by simply touching your paddle, which instantly kills the memory transmission. You can also kill a message by pressing any two of the memory buttons and releasing, which will stop an extra dot or dash from going out over the air.

To erase a message you have already loaded, press that memory button and hold it until you hear the tone. The message is now erased. release the button, then either enter a new message, or simply press the button once again to close the empty memory.

Suppose we want to shut off the monitor when we go on the air, so we can use the transceiver sidetone instead. Simple. Press the left two buttons (#1 & #2) down together, then release them. The keyer will
send back to you the letter "F". (This stands for "Function".) Now, send the letter "M" (for "Monitor"). That's it. The monitor will be off during normal transmissions, and you instead rely on the transceiver's sidetone.

Now, let's turn the monitor back on. Press the left two buttons again. You will hear the speaker send an "F", even though the monitor is disabled for normal sending. Now, send another "M", which you will hear though the monitor as you send it. That's it. The monitor is on again. The monitor control function is a toggle command, and you just learned how to switch it on and off. Note that the "F" the keyer sent did not go out over the air, nor did the "M" you sent to toggle the monitor function. In pressing the two function buttons, you took the keyer "off line" and off the air until you had completed your command. Suppose you screwed up and sent an "O" instead of an "M". Since the "O" is meaningless to the keyer, it will send a raspy signal, a Bronx Cheer. That means you goofed. To recover, simply press the two left buttons again, and send the "M" again. Or, if the command had been a valid one, but not what you had intended to send, you could have sent a string of 7 or more dots, and the keyer would have given the error message and then returned to normal mode.

Need to close the key to tune up the rig? Press buttons #2 and #3 together, then release them. The keyer will send an "H", as in "Hand- Key". Now, any time you press either the dot or dash paddle, instead of sending dots or dashes, you will get continuous output. This lets you hold the "key" down while you tune up that big rig (into the dummy load, of course) in easy stages. When finished, press any button, and the keyer will be back to normal.

OK, that's enough for now. You've learned how to program messages. It's time to hook the keyer up to the rig, get on the air and make a few QSO's, and enjoy how fluid and clean CW can be. And when you are ready, move on to the next section if this tutorial, and we'll go into some of the fancier options. They are easy to learn too, but right now let's use what we have learned.

SECOND SESSION

OK. You've been using your CMOS Super Keyer and you've discovered how smooth and well behaved it is. Likely your transmitting speed has improved as well. Let's start exploring the next level of features your keyer has to offer. Let's turn the rig off, and explore more of the commends available to you.

First, let's try the "Inquiry" mode. Push the right two buttons (#5 & #6) together, then release them, which always puts the keyer in the Inquiry mode. The keyer responds with a "?" in CW. Now, simply send the letter "S" through the paddles. The keyer sends back a number - the speed in WPM the keyer is presently set for. Press buttons #5 & #6 again. (The keyer automatically leaves the Inquiry mode after each question, so for each new inquiry, you must re-enter the Inquiry mode.) Now send a "Q" through your paddles. The keyer will respond by sending in Morse either "ON" or "OFF". Now, press the two right buttons, and send the "Q" again. You will get the same answer. The point here is that the Inquiry mode only tells you what the keyer is set for, and down NOT affect the setting. The Function mode, on the other hand, would have toggled (reversed) the setting. A little later we will actually discuss the "Q" command.

Press the Inquiry buttons again, and this time respond to the Morse "?" by sending the number "1" through your paddles. The keyer will play back to you the message presently stored in message memory #1. You can read out the other messages by calling out their number, which corresponds to their button number. If there is no message stored, there will be no response.

What is the advantage of entering Inquiry mode to read out a message memory? Isn't it easier to just push the number 1 button and have it read out? Yes, except that in doing so, the stored message will go out over the air if the transmitter is active. Reading the memory contents through the Inquiry mode will play the message back over the monitor speaker, even if the monitor is toggled off, and NOT out over the air.

Now let's work with the "Function" mode, always activated by pressing the two left buttons (#1 & #2) down. You will recall using the function mode to toggle the monitor speaker on and off earlier. The function mode is a very powerful tool, and is used to control a number of other keyer functions, some of which we are now going to explore.

Let's reset the speed range. This will give you a good example of how the keyer is programmed at the same time. You say you never send slower then 10 WPM, and rarely go above 45 WPM? OK, let's set up a range of say 8-50 WPM. (And remember, we can always change it later if we want.) Look in the Operating Manual under "FUNCTION COMMANDS" and find "R" for Range. The instruction says that we can program inside a range of 5-60 WPM, so we are OK. The command is "R dd ee". The "dd" stands for the low range limit, the "ee" stands for the high speed limit. SO the command we want is "RØ85Ø" (or R Ø8 5Ø". That is, I want a range of 8 to 50 word per minute. Now let's enter that command. Activate the Function mode by pressing the left two buttons, #1 & #2. The keyer responds by sending "F" in Morse. Now, send the "RØ85Ø" command ("R Ø8 5Ø" is OK, too). The keyer now knows what speed we want, but it is not sure where the speed knob is set, so we should turn the speed control fully clockwise and send a few dots and then fully counterclockwise and send a few more dots. That gives the keyer the calibration information it needed. Now we can set the speed knob anywhere we want and start sending, with the range tailored exactly as we want it. (Be sure to note: To enter a speed below 10 WPM, you must enter a leading zero, such as 07 for 7 WPM, when sending the "Range" command.)

But let's say you really screwed up, and somehow programmed in a range of 50-60 WPM. And you absolutely top out at 45 WPM, and now you can't reset the range to a more manageable number (because you can't send the numbers properly at the higher speeds). Not to worry - there is an escape hatch programmed in. Simply press buttons #1 and #6, and hold them down for 2 seconds and release. Bang! The range has just been reset to 5-40 WPM, and you are back in control. But do remember to immediately do the speed calibration exercise with the knob.

While we're discussing speed, let's look at the keyer speed while you are in the function (or inquiry) mode. Normally, the code speed in the function mode is identical to the regular keyer speed. But, if you like, you can set the function speed to a fixed value, with a speed range available of 5-30 WPM. This feature could be valuable to you if you like sending at high speeds, for example, but want a slower, more deliberate speed to access anything through the function or inquiry modes. To set the function mode speed at a fixed value, enter the Function mode (always by pressing buttons #1 & #2, then releasing) and send "F 1Ø". This will set your function mode speed at 10 WPM. Immediately after you have finished sending the "10" the keyer returns automatically to normal mode. If the keyer was at 20 WPM before, it returns to 20 WPM. But now enter the Function mode again. The "F" prompt will come back to you at 10

WPM. And now, while you are entering function commands, the keyer is set at 10 WPM. As soon as you are finished with the Function mode, the keyer will return to the normal speed you were at, in this case 20 WPM.

If you decide you prefer the function speed to track the normal keyer speed, which is the default setting for the keyer, you may return to it by entering the Function mode, then sending "F ØØ". Now the Function mode speed will always be set the same as the normal mode speed.

In the first tutorial session we briefly discussed the "Q" setting, which determines whether messages can be "queued" by pressing message buttons in sequence. The default setting is "ON", which most operators prefer. To turn it off, enter the Function mode and answer the "F" prompt by sending a "Q". The keyer will respond by sending "OFF", indicating that is has turned the queueing function off. Now, press button #1, release it, pause a moment, then press button #2 while message #1 is still playing. The instant you press button #2, message #1 will terminate, and the message stored in memory #2 will begin transmission. To restore queueing, again enter Function mode and again send the "Q" command. The keyer will respond with "ON", indicating that queueing has been restored.

Let's try the weighting function. Your keyer has an extremely accurate weighting capability because it digitally processes the lengths of dots and dashes separately. Default weighting is 50%, supposedly ideal. And for many users it is. But the weighting can be easily and precisely modified. To do so, enter the Function mode. Answer the "F" prompt by sending "W 3Ø" from the paddles. Then send your callsign. Sure sounds different, doesn't it? That's 30% weighting. Now, enter the Function mode again, and this time send the command "W 7Ø". Send your callsign again. Yup, it sure did change... 25% through 75% is the maximum range of weighting available, and of course, in normal operation, these extremes would never be used. However, at higher speeds of sending, some operators prefer heavier weighting, using perhaps 55% or even 60%. Again, operators at slower speeds, particularly in the 6-10 WPM range, may prefer a weighting of perhaps 45% to 40% for a more pleasant sound to their ears. If you are inexperienced in such matters, the best advice is to restore the weighting to 50% and leave it there until you have a specific reason to change it. You might want to get a friend who is a good CW operator to listen to your signal over the air and advise you.

The weighting setting the keyer is using can also be asked for through the inquiry mode. To check, simply press the right two buttons (#5 & #6) to enter the Inquiry mode, and answer the "?" prompt by sending a "W" through your paddles. The keyer will send the spacing percentage it is presently set for.

Some transmitters, unfortunately, do not perfectly reflect the keying supplied to them. The usual problem is that they tend to shorten the length of all dots and dashes from the keyer. Such delays allow QSK circuitry time to function, and the keying errors are the same at all speeds. A recent major transceiver, for example, subtracts 18 milliseconds from all dots and sashes. This gives the listener to your rig an impression of light weighting, especially at higher speeds. Such induced errors can, theoretically, be cured by adjustment of the weighting towards a heavier value. However, such correction would be correct at only one speed setting. Your CMOS Super Keyer offers a specific correction feature for this problem, the "K" function. To zero out an 18 millisecond error, enter the Function mode, then answer the "F" prompt with a "K 18" command sent through your paddles. The "18" adds 18 milliseconds of transmit time to every transmitted dot and dash, thereby canceling out the keying error.

But what correction is appropriate for your rig? There are several ways to find out. One is to ask stations on the air about how your weighting sounds, particularly if it is set at 50%. If other operators tell you it is light, try adding say 5 milliseconds (K 5) and ask for further reports, preferable from the same station. remember, it takes only a few seconds to change the setting for an experiment. Another way is to check out magazine product reviews. Some reviews, using photographs of a dual trace oscilloscope, show the actual keying against the transmitter output, thus displaying the keying error. Again, is some cases, the manufacturer actually specifies the variation in the manual.

The "K" function setting can be read from the Inquiry mode, if desired, by responding to the "?" prompt with "K" sent through your paddles.

Another function mode control is to switch auto-spacing on or off. Auto-spacing has always been controversial among CW operators; some operators prefer it while other abhor it. What auto-spacing does is force the operator to leave at least three space elements between transmitted letters, so that with auto-spacing letters are not run too close together. When you send a letter and then pause before sending the next letter, the keyer senses that more than one space element has gone be in the timing, and will not begin transmission of the next letter until a full three space elements have passed. Without auto- spacing, the operator alone is responsible for the timing of transmitted letters.

The reason some operators dislike it is that if they try to force letters through too quickly, the keyer "stutters" and won't start sending the next letter until the mandated three element spaced have gone by. This causes a feeling of loss of control in the mind of some operators. Other operators relish the extremely precise letter spacing that auto-spacing allows. The CMOS Super Keyer default is to have auto-spacing off. To turn it on, enter Function mode, then answer the "F" prompt by sending "A". Since auto-spacing is a toggle function, this will reverse the existing state, and auto-spacing will be turned on. Once it is on, send a little CW for practice, and see what you think. Auto-spacing is much more noticeable at slower speeds, especially for operators going slower than their normal operating speed. To turn auto-spacing off, again enter the Function mode and send the letter "A" through your paddles.

The on and off state of the auto-spacing switch can also be checked by using the inquiry mode. Respond to the "?" prompt by sending "A", and the keyer will answer with "ON" or "OFF".

Please, one last comment about auto-spacing on your keyer. If you disliked auto-spacing on other keyers, try it again on the CMOS Super Keyer. You will almost surely find the implementation friendlier than that of any other keyer you have ever used, and you might just change your mind. Give it a chance.

While we're playing around, let's try resetting the monitor tone. the default setting is 700 hertz. Enter the Function mode, and enter "T9Ø" (or "T 9Ø"). Suddenly, you will find a monitor tone of 900 Hz. The command is "Tdd" where "dd" represent the first two digits of the monitor tone frequency. If you don't like 900 Hz, a "T7Ø" will restore the monitor to the default. The available tone range is 500 Hz to 990 Hz (T5Ø through T99).

Another available function command offers a choice between "Character Mode" and Real-Time Mode" for loading messages. You already know that when you hold down the memory button to initiate the loading of a message, the keyer responds with a Morse "C", telling you the keyer is in "Character" mode. When we discussed this earlier, we were more interested in getting you up and running, rather than covering all the fine points at once. We stated that we would come back to that "C" for Character mode later - and now is the time.

You already know that to program a message you send a word, and then the keyer sends a high-pitched "I" to tell you the word has been added, as well as a word space increment, and then accepts the next word. You also learned that you could remove a mis-sent word by sending a string of dots, which the keyer would recognize and would then erase the last word loaded. these techniques allow maximum utilization of the memory, allowing the longest possible messages to be loaded. They also are more convenient for most operators.

There is an alternate load mode available, however, called "Real- time". In Real-time mode, what you load is mirrored back to you, warts and all. This mode can be useful if you have a special characteristic way of sending something, usually your callsign. For example, if your call is WY9EI, you might prefer a little extra spacing between the "I" and the "E" for emphasis, so the E won't get lost in the shuffle. In Character mode message loading, you could add a complete word space between the "I" and the "E", but that would be excessive. In real-time mode, you get the exact spacing you want.

Real-time mode has several disadvantages. One is that it is wasteful of memory. Real-time mode uses a relatively inefficient storage pattern that wastes memory space. Also, since Real-time mode is a mirror of what you send to the memory, you can no longer correct loading mistakes with a string of dots.

The default load mode is Character mode. The command switching the load modes is a toggle command. To switch, enter Function mode, then send "L", which will toggle you into the opposite load mode, and indicate the new mode with a "C" or a "R" as appropriate. Whenever you load a message by pressing the memory channel button several seconds, you will be reminded which load mode the keyer is in; the keyer will send either a "C" or a "R". Note that some memories can be stored in Character mode, while others are stored in Real-time mode. the CMOS Super Keyer will accommodate both modes at the some time.

You can simply ask the keyer which mode it is in through the inquiry mode. Simply enter Inquiry mode and send "L" following the "?" prompt. The keyer will respond with "C" or "R".

Now let's discuss the Bank command and how the keyer uses memory. You will remember that in the default mode, the keyer has six memories, each capable of storing messages up to 255 characters in length. However, the keyer also offers the option of dividing the memory into 18 messages, each 85 characters in length. (Still very long!) The catch is that you cannot use all the messages at any given time.

If you are in the default state, which is "BØ", the memory is divided into 6 banks of 255 characters each. However, if you change the banking command to "B1", "B2", or "B3" via the Function mode, the memory is automatically set to store 18 messages. "B1" allows you access to six messages in bank #1, and all the other commands work as stated. If you desire to set up another bank of messages, enter the Function mode and send "B2". You will now have available six new empty messages. Going to bank 3 by the Function command "B3" gives you the last six message memories.

Messages stored in each bank can only be called and used when that bank is active. So, the active contest operator could store one set of messages for the CQ WW (CQ Worldwide DX) contest in Bank 1, the ARRL DX contest in Bank 2, and Sweepstakes in Bank 3. Or a family of hams could each have their own bank, callable at will.

OK, that's enough for this session. Of course, many of the commands and features we have explored will be used lightly or never. Which is exactly as it should be. The defaults are set to reflect the desires of most CW operators. But by now you surely have a greater understanding about some of the features of the keyer, and have doubtless customized several of the defaults for your own satisfaction. In the final lesson, we will learn how to set up the message memories to do some really neat and useful tricks, list automatic serial number generation, automatic speed changes in mid-message, closed loop messages, etc. But take some time off before you get into that. Get on the air and make some contacts with your keyer, and play with what you have already learned.

THIRD LESSON

Although the features you have learned thus far already put your CMOS Super Keyer far beyond most keyers, there is one more layer of commands and tools that will really make you and your keyer shine. These features include several more function mode commands, and another type of command, the embedded command. Embedded commands are inserted into programmed messages, and allow pauses, closed loops, calling one message from within another, contest serial numbers, speed changes inside the message, and more.

What is an embedded command? It is a special command inserted (through the paddle) into a message being loaded into memory. Embedded commands can only be used in a message loaded in Character mode, rather then in Real-time mode. Embedded commands always follow a word space ("I"), and start with a "/" followed immediately by the command (all sent as one word).

For purposes of this tutorial only a "_" in the following test is to be considered a WORD SPACE - not a transmitted character. .
The underline dash is easier to recognize than a simple space. OK? Good.

How do you program your portable callsign if the "/" is used for entering an embedded command, you ask? Simple. As long as the slash bar is NOT preceded by a word space, the keyer will recognize it as a slash bar. So, if you are signing WY9EI/KH7, no problem at all. Just key it into the memory. (Now, on the other hand, if you want to program in WY9EI_/KH7 with a word space pause, that can be done with the Gap command, which we will learn shortly.)

So, Let's try out a message with an embedded command. First, let's call one message from inside another. To do so, first load into message #1 your callsign - "DE WY9EI K". Now, load the following into message memory #2: "CQ_CQ_CQ_/1". (Don't forget that the "_" means a word space in your message load, indicated by the keyer's high-pitched "I".) Got it stored? OK. Now, simply, press the #2 button to play out its stored message. Hah! See that? It called and played message #1, running it out with a perfect word space between the two messages. OK, now load a different message into message #1: "CQ_CQ_CQ_DE_ WY9IE_WY9IE_/1". OK, now play it back. See how it keeps playing over and over again? You have created a closed loop, with the message continuously calling itself. Of course, to stop the message, all you have to do is to touch your paddles, or press (and release) any two buttons at the same time.

Now, let's examine the embedded Pause command, called by "/Pdd", where "dd" stands for an operator-selected number between 00 and 99, and where each digit stands for one tenth of a second. So, load this message into memory #1: "CQ_CQ_CQ_DE_WY9EI_WY9EI_/P5Ø_/1", then play it back. Now you should have the CQ message play, sign your call, and then after a 5-second pause, repeat again. And again. And again, etc. The "/P5Ø" is your embedded command into the message to pause five seconds, then continue, where continuation in this case calls for a repeat of message #1. And note that

you embedded two commands in a row, with each doing its thing. This closed loop message with a pause for listening is break for beacons, contest CQing on a relatively dead band, meteor shower skeds, etc. Try "ABC_DEF_GHI_/P5Ø_JKL_MNO". See how the pause allows internal breaks in message transmission" But remember, if you touch the paddle during the pause, the message will be killed, and you are restored to normal mode.

Let's make a beacon message out of it, with a key down tone of 5 seconds. Load this into message #3: "TEST TEST DE WY9IE/B WY9EI/B /X /P5Ø E /3". The "/X" of course puts the keyer in the "key-down" mode, and the "/P5Ø" command causes it to go for 5 seconds. You ask what the "E" after the "/P5Ø" command is for? It is silent - it breaks the "key-down" mode. Load it and try it. To escape the looped beacon mode, simple touch the paddle or press any two buttons.

Now, suppose you want a message that would include hand-sent text, like the other station's callsign in a contest exchange. For that, embed a "/B" command, "B" as in [B]reak. Load this message: "UR_RST_ /B_DE WY9IE" and play it. Notice that the message playback gives you the "UR RST", then quits. Now, manually enter the RST report - say, "579", then release the paddle. Almost immediately after you release the paddle, the "DE WY9IE" will play. the "/B" command opens a window for paddle entry. When the message reaches the "/B", it stops and waits for your paddle entry. While you are sending manually, the keyer monitors your sending, and as soon as it detects a space greater than a word space, it picks up where it left off and finishes the message. Note that there is one possible problem in using this embedded command: you will find, especially at high speeds, that any delay during your manual sending will cause the keyer to try to resume BEFORE you are finished. Then, when it senses your continued paddle closures, the keyer will think you have terminated the message and will stop, leaving you wondering what happened, and maybe feeling a little foolish. (This problem may be lessened in auto-space mode, because you can send just a bit "ahead" for the timing and still have perfectly-spaced CW, while not exceeding the word space timing window.)

A similar command, "R", as in [R]esume, allows more timing tolerance for hand-sent entry. Try putting a "/R" command in a message where you want to insert hand entry. When the message plays to that point, it stops, and you may now enter any hand-sent material you desire. The keyer will not finish the rest of the message until you press the SAME memory button again, at which time the message will resume. Both multiple "/R" and multiple "/B" commands may be entered in a message.

Another embedded command is the "/S" command which lets the operator set the speed in a message. Try entering this message, then play it back: "/SØ6_CQ_CQ_/S12_CQ_CQ_/S24_CQ_CQ/S48_DE_WY9IE_K". Note that for speeds under 10 WPM, you must use a leading zero, as in "Ø6". When the stored message is complete, the keyer speed will revert back to the knob setting. These commands will be especially useful in contest exchanges with "canned" responses, like "59904" or "599KW". Load this message: "/SU15_599Ø4_73_/SD15", then set your keyer to perhaps 10 WPM and send through the paddle: "ZA1DX DE WY9IE", then press the message button. When you return to the paddle, you will find that you are back at 10 WPM or whatever speed you started at. As you can see, this ability to speed up an expected message can give you more QSO's per hour in a contest situation, and thus can be extremely valuable. When embedding such commands, remember, any time you use a "/SU" command, you will almost certainly want to also use the complimentary "/SD" command to restore the keyer to the speed it started at. And don't forget that the speeds must be given in the command in two digits. So an increase of 8 WPM would be "/SUØ8". Another embedded command can be useful for a number of functions - the"/Gd", or [G]ap command. Remember when we talked about sending "WY9IE" and trying to put extra space between the "I" and the "E"? In a simplistic Character mode message, this is impossible; the only option being a whole word space. The other simplistic option is storing a message in the Real-time mode.

But with the "/Gd" command, (the "d" as in "digit") it is possible to set an exact "gap" between the "I" and the "E". Here's how. A normal space between two characters is 3 "bits" long. (A bit being the length of a single dot.) The "/Gd" embedded command adds as many bit spaces as you want (up to 9) to extend a normal letter space. remember, the minimum space is three bits long, and the Gd command adds to that basic space. A "/G1" embedded command lengthens a space between two letters only a small amount. A "/G4" command will open up the space from a letter space to a word space (7 bits total). Try programming: "WY9I_/G2_E" and see how it sounds. Experiment with different values for "Gd". Once you understand it, you will almost surely use it for one special situation or another.

It's time to leave the subject of embedded commands and examine serial number generation, primarily useful for contest operation. The CMOS Super Keyer serial number controls are unmatched by

any other keyer design available. First, you must decide what number format you want to use. Do you send zeros in contest serial numbers using five dashes, or a single dash? Do you use leading zeros? What about nines? Do you spell out a proper nine, with four dashes and a dot, or do you follow the now common format of using an "N", as in "5NN"? The CMOS Super Keyer "Zd" function command lets you specify your choice. Examine the Z Option table in the OPERATING MANUAL, and choose your serial number format.

Let's suppose you decide on option #6, with both leading zeros and other zeros sent as a "T". Now, enter the Function mode of the keyer and respond to the prompt with the command "Z6". This will cause generated serial numbers to reflect that format.

Now that that has been set, you also have the option of presetting a serial number. If you do not do so, the first serial number sent will be "TT1" (for 001). If you do want to preset a number, say 1066, enter the Function mode again, and enter "N 1Ø66" at the prompt. Note that you must enter a five-dah ZERO, as in "1066" even though the keyer will send it as a "T" under the "Z6" option. Now, enter the inquiry mode (buttons 5 & 6), and enter "N" at the prompt. The keyer will send back "1T66", CW shorthand for 1066. If you want to reset the serial number at a low number such as 23, you must use leading five-dash zeros to enter a four-digit number (e.g. "0023").

So how do we use the serial number in a contest message? Right - with an embedded command. Let's enter: "NR_/N_A WY9IE 8Ø IA K". This will give you "NR TT23 A WY9IE 8Ø IA K". "But", you say, "There is a word space between the serial number and the 'A'. And that is wasted time in a contest where time is of the essence." Right. So let's get rid of that unwanted word space with the embedded "/Gd" command. Reprogram the message thus: "NR_/N_/GØ_A WY9IE 8Ø IA K". Now we have eliminated the word space between the serial number and the "A". It will not take long for the contest operator to appreciate the flexibility these commands offer.

"But", you say, "What do I do when some clown calls me for a duplicate QSO and I send the serial number before I realize he's a dupe? Do I have to reset the serial number with the "N" function command? That would be a pain!" And indeed it would be. But you don't have to reset the number. All you do is press (and release) the middle two buttons (#3 & #4) and the keyer sends a "D" and automatically decrements the serial number by one! That's all there is to it. And each time you press buttons #3 & #4, you decrement another number in the stored serial number.

There is another button function we have not covered - another tune-up approach. Pressing and releasing buttons #4 & #5 together causes the keyer to send an "X" over the monitor and closes the key contacts to the transmitter for tune-up purposes. The rig stays keyed until you touch either side of the paddle, which releases the rig.

A special embedded command is the "U" parameter for Ultra-Speed transmission, which allows programmed messages to be sent at speeds from 70-990(!) WPM. This mode is used almost exclusively by meteor- scatter enthusiasts, who record the messages on high-speed tape recorders, then play back the tapes at slower speed for decoding. To generate an ultra-speed message, start the message with an "/Udd" embedded command, where "dd" are the first two digits of the desired speed. For example, "/UØ7" will send the message at 70 WPM, while "/U9Ø" will send it at 900 WPM. Note that weighting, compensation (K), and tone settings are suspended during the ultra-speed transmission. Ultra-speed may be used as part of a message that also contains regular speed text. Read the Operating Manual description of Ultra-Speed in the EMBEDDED FUNCTIONS section of the manual.

There are two different ways to examine the contents of a message stored in memory without going on the air. Load message #1 with embedded functions, such as "/SUØ9" and "/SDØ9", in the message. Now, enter the Inquiry mode and answer the prompt by pressing button #1. Notice that the message plays back exactly as it will go out over the air. Now, again enter Inquiry mode and this time answer the prompt sending a "1" through your paddles. Note this time that the playback of the stored message is exactly as you loaded it, showing the
slash bar and embedded commands, exactly as you entered them. You will find each way of examining a stored message valuable at different times.

If you decide you want to change an already stored message, the "Edit" function can help avoid completely erasing the message. Editing works by deleting "words" off the back of the message until you reach to point where you wish to retain the balance of the message. You can then add to the message or keep the shortened message. To edit a message, enter the Function mode, and send the letter "E" followed

by the number of the message you wish to edit. The keyer will now send the last word in the message. Remember, the 'word' could be an embedded function, such as a looping command like "/2". If you want to delete the "/2", simply send a string of seven or more dots. The "word" will be erased, and the keyer will send the next prior word for you to decide to keep or erase. At any point, you may close the message and the editing session by pressing the message button, or you may add text by keying it in. Remember that editing works only for messages stored in Character mode.

If you examine the regular operation manual, you will note that some of the commands discussed above can be entered in several ways. For example, the decrement command can be sent through the function mode. But pressing buttons #3 & #4 is often so much easier. For this reason, the function mode method of decrementing was not covered in the tutorial. (Till now, anyhow!) Nonetheless, you will find it and similar commands in the Operating Manual.

(Note that any INQUIRY command may also be issued from within the FUNCTION MODE if it is prefaced by a "?". For example, entering the Function mode and sending "?W" will cause the keyer to respond with its current Weight setting. This option is a carry-over from the CMOS Super Keyer II.)

So. That concludes the tutorial of your CMOS Super Keyer. We know you will be delighted by this outstanding design. Enjoy it, and be sure and tell your friends about it as well. 73!