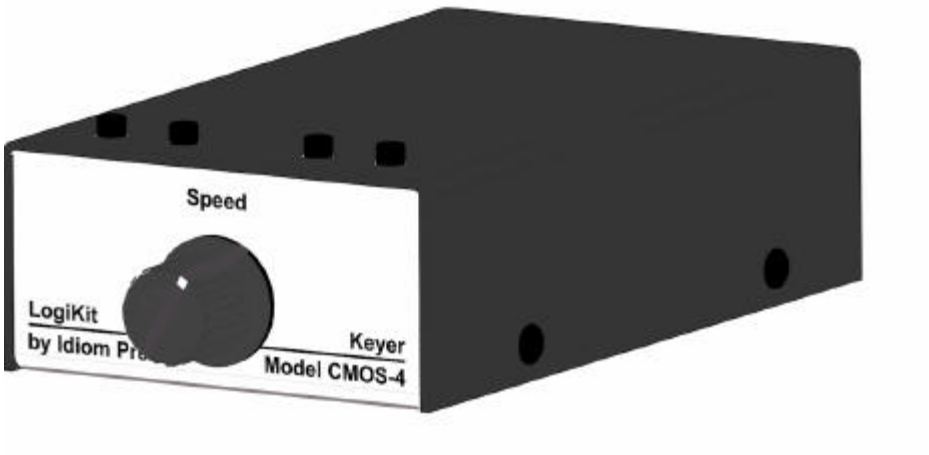


Logikit CMOS-4 Keyer

Operating Manual 2
Tutorial 12
Troubleshooting 22
Circuit Diagram 23
Warranty 24

Getting Started: You will need the following with your Logikit CMOS-4 keyer
1 3.5mm stereo plug with two wire shielded cable for paddle
1 2.5mm DC power plug, with wire to power source
1 RCA plug terminated cable to transmitter, shielded center conductor
1 Paddle
12 Volts DC - **Do NOT** use typical 12 volt wall style transformers
3 – AAA batteries
See additional notes on Page 12 of this manual.

Idiom Press
P.O. Box 1985
Grants Pass, OR 97528
www.idiompress.com

OPERATING MANUAL

Logikit CMOS-4 KEYER

The Logikit CMOS-4 keyer is a compact, full-featured memory keyer combining a CMOS microprocessor and a non-volatile RAM chip for a full featured, low cost, high reliability design. Commands are simply sent to the keyer in Morse code using your paddles! Some of its features:

1. Iambic keyer with dot and dash memories.
2. Four active messages plus 8 "banked" messages, 1020 characters total.
3. Messages may 'call' others and contain programmed functions.
4. Input queue to store multiple message activations.
5. Contest serial number - 001 to 9999.
6. Digital and linear analog speed control - 5 to 60 WPM.
7. Adjustable weight on code elements - 25% to 75%.
8. Built-in adjustable frequency sidetone monitor, Adjustable monitor level
9. Tune function for transmitter adjustment.
10. Selectable automatic character spacing.
11. Timed pauses within messages.
12. Message loop capability for continuous replay.
13. Messages can allow break-in for paddle-inserted text.
14. Emulation available for other keyers, including Curtis "A" timing.
15. Ultra Speed mode allows messages at speeds to 990 WPM!
16. Full beacon capability.
17. Message editing capability.
18. Messages and keyer configuration saved when power is lost.
19. Keyer can compensate transmitter character shortening.

START-UP

After power is applied, the keyer responds with "OK" and is ready for operation. It is initialized as follows:

Speed range:	5-40 WPM	Load mode:	character
Weight:	50%	Input queue:	on
Monitor:	On	Serial number:	001
Auto-space:	Off	Emulation:	Logikey K1, K3
Function speed:	Equals Paddle speed	Monitor Tone:	700 Hertz

Note: At start-up the first time, the speed control knob should be turned fully clockwise and a few dots sent. This “calibrates” the speed range. The monitor volume control is the trimmer resistor accessible through the bottom hole. Adjust the control with a small Phillips screwdriver, while sending a string of dots.

RESET

The reset command erases all settings and messages from the EEPROM memory. If the keyer is not acting properly, even on initial start up, execute the RESET command. The command is implemented by pressing buttons 1,3, an d 4 down simultaneously and release. When the reset command is used, all stored messages and settings will be lost.

KEYING POLARITY

Keying polarity is set by the internal jumper on the header strip. For grid-block keyed rigs, the jumper should be on the outside two pins of the header; for solid state rigs on the inside.

FUNCTION COMMANDS

Commands to the keyer are entered in Morse code using your paddle. To alert the keyer that a function is desired, momentarily press buttons 1 and 2 simultaneously. The keyer enables the monitor, disables output keying, and acknowledges your request by sending "F".

After hearing the "F", simply paddle in the desired function character(s) explained below. When the sending of the function is complete, the monitor automatically reverts to its previous state, output keying is enabled, and the previous operating speed is restored.

Input command strings and the function performed are as follows:

A — Auto-space - toggles the automatic character spacing feature on or off. The keyer confirms the new state by sending "ON" or "OFF". When on, auto-spacing aids the operator in forming properly spaced characters within a word. Character crowding is prevented by forcing at least 3 elements of space whenever more than 1 space has been detected. This effect is very noticeable at low speeds and can be felt by high-speed operators as well. Without auto-spacing, character spaces are determined by the operator.

B *d* — Bank - Message banking is provided as an option to support multiple operators or preloaded messages for different contest exchanges. In the default mode, there are four messages of 255 characters each, with a single “bank”. As an option, the available memory can be split into three distinct banks each having 4 messages of 85 characters. Bank 1 is activated by the "B1 " command. Similarly, commands "B2" and "B3" activate banks 2 and 3 respectively. Banking is disabled by the "BØ" command, and the currently active bank can be queried with the "?B" command when in the Function Mode.

D — Decrement - decrements the serial number by one, effectively canceling the automatic increment applied when last played from a message. The decrement function accommodates resending the last serial number, as might be needed when a repeat of a contest exchange is requested, or canceling the exchange with a station that proved to be a "dupe."

E — Edit - allows the operator to append onto or edit an existing message by entering the "E" command in function mode, followed by the number of the desired message. The keyer will find and play the last word in that message. Then paddle in more text, or use the error symbol (seven or more dots) to erase existing words, just like the delete procedure when loading a message.

F *dd* — Function speed - sets the speed used for function entry to *dd* WPM, where *dd* are two digits in the range 06 to 30. This speed is employed for entering commands and loading messages. It is independent of the operating speed and is unaffected by the analog speed control. Alternatively, the function speed can be made to follow the operating speed by using *dd* = 00.

H — Hand-key - The keyer enters hand-key mode. Keying output follows closures of the dot or dash paddle levers, allowing hand-sent code. Normal iambic keyer operation is regained by any button closure.

K *dd* Keying Compensation - increases keying on-time and decreases keying off-time by *dd* milliseconds, where *dd* are two digits in the range 00 to 25. Although similar to increasing weight, the adjustment is independent of speed. This setting is used primarily to correct keying distortion by certain transceivers, usually in QSK mode. Note that the adjustment is not heard on the monitor output, except when playing back the message from the "Inquire" mode.

L Load mode - toggles the load mode between character and real-time. The keyer confirms the new mode by sending "C" or "R" as appropriate.

M Monitor - toggles the audio monitor on or off. The keyer is usually operated with the monitor off in favor of the rig's sidetone.

N *dddd* Number - initializes the contest serial number to *dddd*, where *dddd* are four digits in the range 0000 to 9999. Note that 4 digits must be entered, with leading zeros if needed. Also note that in transmission of a serial number that a 4th place leading zero is never sent.

Q Queue - toggles the input queue on or off. The keyer confirms the new mode by sending "ON" or "OFF" as appropriate. When off, message button pushes are acted upon immediately, canceling any message in progress. When on, up to 8 button presses are remembered in order and acted upon in succession as each message completes.

R *ddee* Range - programs the speed range covered by the pot, with a range of 5 - 60 words per minute. Sets the current operating speed to *dd* WPM, where *"dd"* is the low setting, and *"ee"* the high setting. A command of RØ545 would set a range of 5 to 45 words per minute, and the present position of the pot would determine the keyer speed within that range. Control via the knob is linear and increases speed clockwise.

T *dd* Tone frequency - Available range is 500 - 990 hertz, where *dd* is the first two digits of the desired monitor frequency, i.e. “T70” = 700 hz.

V *d* Emulation - allows the operator to select emulation of the timing characteristics of other keyers. See EMULATION later in the manual.

W *dd* Weight - sets to code weight to *dd* percent, where *dd* are two digits in the range 25 to 75. Weight is the duty cycle of a continuous string of dots, which is 50% for perfect code. A higher weight produces a heavier sound, and a lower weight causes characters to sound lighter. Once set, weight remains constant and independent of speed.

X Xmit (tune) - Continuously keys the output for purposes of transmitter and amplifier adjustment. Tuning is stopped by simply tapping either the dot or dash paddle lever.

Z *d* Zeros and Nines: controls the way that zeros and nines are sent in a contest style serial number. See SERIAL NUMBER OPTIONS.

INQUIRY FUNCTIONS

Inquiry functions allow the current state of the keyer to be determined. State information is played to the operator in Morse code with the monitor automatically enabled and keying output disabled. Inquiries operate just like command functions: to enter an inquiry, momentarily press the right two buttons (#3 & #4) simultaneously. After receiving the "?" reply, enter the desired inquiry command as follows:

A Inquire Auto-space - the keyer responds by sending "ON" or "OFF" as appropriate.

B Inquire Bank - setting the keyer sends the current operational bank number, BØ, B1, B2, or B3.

F Inquire Function Speed - the keyer sends the current function speed setting in WPM as two digits. A “0” inidicates function speed follows the pot speed.

K Inquire Keying Compensation - the keyer sends the current compensation in milliseconds in one or two digits.

L Inquire Load mode - the keyer responds by sending "C" if in character mode or "R" if in real-time mode.

N Inquire Number - the keyer plays the current contest serial number (but does not increment it).

Q Inquire Queue - the keyer responds with "ON" or "OFF" as appropriate.

R Inquire Range - the keyer sends the current operating speed range in WPM as four digits with a pause between the slow limit and the fast limit.

S Inquire Speed - the keyer sends the current speed setting in wpm.

T Inquire Tone - setting for monitor - the keyer sends the current monitor tone setting.

V Inquire Emulation Setting - the keyer sends the current emulation. See EMULATION

W Inquire Weight - the current weight percentage is sent by the keyer as two digits.

Z Inquire Zeros and Nines: the keyer responds with the option number (0-9) currently in effect.

1 Inquire Message 1 (or 2, 3 or 4) - message 1 (or 2, 3 or 4) is played exactly as it would go over the air, but with the output disabled.

Note: You can also play back a message sounding the embedded function commands. (See below) To do so, momentarily press the right two buttons (#3 & #4) simultaneously. After receiving the "?" reply, press the message memory button you wish to review. The message will be played back with any embedded commands.

EMBEDDED FUNCTIONS

Certain functions can be embedded within character messages. To distinguish them from normal text, the command strings are prefixed by a "/" and are entered as a separate word. When encountered during a a message play, the functions are executed. Note that if "/" is part of a single word, as in W9KNI/ZA2, it is sent as expected and is not interpreted as a command prefix. Embedded command strings and their use are explained below:

/B Break - message play is suspended to allow insertion of paddle text. The operator may then insert code using the paddle. Once paddle input has begun, the Break function is canceled when inactivity exceeding a word space is detected. The interrupted message then resumes. A Break may also be aborted by pressing a button, which will cause the corresponding message to resume play immediately.

/D Decrement - decrements contest serial number by one.

/Gd Gap - the normal 7-element interword space is modified to 3+d, where d is a digit from 0 - 9. It is used to exaggerate inter-character or inter-word spacing. For example, a call like WØEJ can be entered with slight lengthening of the space between the "E" and the "J" for emphasis, making it easier to copy. Note that /GØ yields a normal character space, while /G4 yields a normal word space. Also, note that you need to wait until after the high pitched "dit-dit" before sending the next letter. Multiple "/G*d*"s can be stacked for greater length.

/N Number - the current value of the serial number is played, then automatically increased by one. Also see SERIAL NUMBER OPTIONS.

/P*dd* Pause - a speed-independent pause of *d.d* seconds is inserted, where dd are two digits in the range 00 to 99. For example, /P35 will result in a delay of 3.5 seconds. Pauses longer than 9.9 seconds are obtained by using consecutive commands that total the value desired.

/R Resume - Stops message play to allow hand-sent entry. When manual keying is completed, press the message button and the message will resume from that point. Multiple "/R" commands are permitted.

/S*dd* Speed - Sets the operating speed to *dd* WPM in a message, where *dd* are two digits in the range 6 to 60. This is used when you want a message to be sent at a specific speed rather than the pot speed. If not changed, subesequent messages called from within the message will also be played at the specified speed. To restore message speed to the pot speed, embed the command "/SØØ".

/SU*dd* Speed Up - increases the operating speed by *dd* WPM, where *dd* is a number in the range 01 and up.

/SD*dd* Slow Down - decreases the operating speed by *dd* WPM, where *dd* is a number in the range 01 and up.

/U*dd* Ultra-speed - sets the ultra-speed mode for a message, used primarily for meteor scatter work. Range is 70 - 990 words per minute, where *dd* are the two digits representing the first two numbers of the speed setting desired. For example, "Ø7" is 70 WPM, "77" is 770 WPM. Weighting (W), Compensation (K) and Tone (T) settings are disabled during Ultraspeed transmission. Messages can be created using both

regular and ultra speeds. To exit Ultraspeed mode in a message, a work-around" is used. At the end of the desired ultraspeed text, send /SUØ1 /SDØ1 . This trick tells the keyer that ultraspeed is ended and returns the keyer to the pot speed. Alternately, the message speed can be set to a normal speed by use of the /S command, such as "/S2Ø" but this will send text following the ultraspeed text at 20 WPM in this case, rather than the speed set by the panel speed control.

/X Close key - allows the keyer to send beacon messages that include extended key down periods. Once started, the output keying will remain on until either the paddle is closed on either side or it is timed-out by a P*dd* embedded command. (see above) The P*dd* command controls the length of the key closure. The P*dd* command should be followed by a letter "e" which will not be transmitted but instead is used to "break" the key down mode. Regular text may then follow.

/1 Message 1 - message 1 is played in its entirety followed by resumption of the current message. To create a continuous loop, end the message with the number of the message as an embedded command, such as "/1", where the "1" is the message being programmed or played. Also, other messages can be appended. For example, message #2's contents can be appended to the end of message #1 by ending message #1 with a "/2" embedded command. And a loop could then be created by ending message #2 with a "/1" embedded command.

MULTIPLE-BUTTON FUNCTIONS

Single-button closures are reserved for activating messages. As already discussed, the 1-2 combination alerts the keyer for paddle-entry functions. Certain functions are duplicated, wholly or in part, by other button combinations:

2-3 Decrement - the keyer acknowledges with "D" and then decrements the serial number.

3-4 Inquiry - the keyer acknowledges with "?" and then waits for paddle entry of the desired option.

1-3 Hand-key - the keyer acknowledges with "H", then enters hand-key mode until another button closure occurs.

2-4 Tune - the keyer acknowledges with "X" and then keys the output continuously until a paddle closure occurs.

1-4 Reverse - the keyer acknowledges with "RV" and then reverses the paddles.

1-3-4 Complete Reset - All stored setting, memories etc. are erased and the keyer returns to original default settings.

Note: To kill a message already transmitting without sending a "dit" over the air, press any two buttons and release.

LOADING CHARACTER MESSAGES

In character mode, each Morse character uses one byte of message memory. Precise 3-element intercharacter and 7-element interword spaces are employed when the message is played (unless modified using the /G*d* function). To load a message, first confirm that character-mode loading is in effect by using the "L" Inquiry function. Then press and hold the desired message button. After 2 seconds, a tone is emitted and the button may be released. The keyer then sends "C" to confirm character mode and waits for input.

Morse text and embedded functions can then be entered via the paddles. When each word is complete, simply stop sending. The keyer will detect and insert a word space, then prompt you for the next word by sending a high-pitched "I". There is no time limit between words, so there is no need to hurry your sending.

The keyer includes an edit feature for immediate correction of errors during message loading. If a mistake occurs, simply send an error indication of 7 or more dots. The keyer will erase the last word sent. It will then play the 'new' last word (if any) so that the position in the message is known exactly. As many words as needed can be erased this way. When the desired position is achieved, continue to enter the remainder of the message.

After the keyer responds with a high pitched "I" following the final word, the message is closed by a momentary press of the button. To completely erase a message from memory, initiate the start of a message, but rather than keying in text, simply press the message button again immediately after the "C" is sent. In the event that message capacity is exhausted during the load, the keyer will send the raspy "error" message and the message will terminate at that point. The operator might wish to then delete the last word with the error string, then have the balance of the message continue in new message channel, and use the embedded command number command (i.e. "/2") to tie the messages together.

LOADING REAL-TIME MESSAGES

Some operators prefer messages containing stretched or compressed spacing rather than perfect timing. Real-time mode stores and replays messages exactly as entered. Along with somewhat reduced message capacity, the primary disadvantage of real-time messages is that they cannot contain embedded functions. Use the Function Mode "L" command to switch to the real-time load mode, then press and hold the desired message button. After 2 seconds, a continuous tone will emit, at which time loading mode is enabled and the button may be released. The keyer then responds by sending "R" to confirm real- time mode and waits for paddle input.

Since the keyer waits until the first paddle closure, there is no need to rush the first entry. Once entry begins, however, the keyer loads continuously - any pauses are stored as spaces in the message. All intercharacter and interword spacing is strictly up to the operator. To end the message, simply press the message button momentarily. Note that the elapsed time from the end of the last character to message termination is stored as space at the end of the message. If room is exhausted during the load, the message is terminated automatically, and the raspy error signal is sent by the keyer, indicating that the loading has terminated. The operator will also notice that messages are not as easy to enter in this mode, since the keyer clock is free-running and thus not resynched with paddle closures. However the stored message will play back cleanly.

PLAYING MESSAGES

It couldn't be simpler: just tap the desired message button. Both character-mode and real-time messages can be played regardless of the current load mode.

If the input queue is enabled, multiple message button closures will be remembered. Each message will be played in succession as the previous completes. As a simple example, suppose that message 1 contains "CQ" and message 4 contains "DE WB8ZRL". Then pressing button 1 three times and button 4 once, in quick succession, will cause "CQ CQ CQ DE WB8ZRL" to be played. As many as 8 button activations will be remembered in this mode. With the input queue disabled, however, a button closure immediately cancels any current message and starts the commanded one.

When a paddle closure is detected, messages are immediately aborted and the input queue flushed. The only exception is during execution of an embedded /B break or a /R resume instruction, when paddle input is expected. Otherwise, the paddles always take priority over message playback.

EMULATION OPTIONS

This parameter allows the keyer timing to mimic that of other keyers, making the "feel" of the keyer more comfortable for operators used to different timing patterns. The default setting is VØ, a timing pattern which has proven to be the most user-friendly for many operators. Other values are as follows:

VØ	Logikey K1, K3 timing w/dot and dash memory
V1	Logikey K1, K3 timing w/dot memory only
V2	Logikey K1, K3 timing w/dash memory only
V3	Accukeyer timing w/dot and dash memory
V4	Accukeyer timing w/dot memory only
V5	Accukeyer timing w/dash memory only
V6	Curtis "A" timing w/dot and dash memory
V7	Curtis "A" timing w/dot memory only
V8	Curtis "A" timing w/dash memory only
V9	Iambic timing w/no dot or dash memory

Note that this makes a full featured memory keyer finally available to those used to the Curtis "A" timing! Tell your Curtis equipped friends! To implement a setting different from VØ, enter the Function Mode and send 'V*d*', where "*d*" is the desired setting. To determine the present setting of the V parameter in your keyer, enter the Inquiry Mode and respond to the "?" prompt with a "V". The keyer will annunciate the "V" setting.

ERROR INDICATION

When an erroneous input or exceptional condition is detected, the operator is notified by a distinctive raucous tone burst. Examples include nonexistent functions, invalid numeric parameters, and exhaustion of message capacity during a load.

SERIAL NUMBER OPTIONS

The "Z*d*" command selects one of 10 options for sending zeros and nines in the contest serial number. Zeros may be replaced by "O" or "T", nines replaced by "N", and leading zeros suppressed. The options available are:

Option (d)	Leading Zeros	Other Zeros	Nines
Ø	Ø	Ø	9
1	-	Ø	9
2	O	Ø	9
3	O	O	9
4	-	O	9
5	T	Ø	9
6	T	T	9
7	-	T	9
8	T	T	N
9	-	T	N

Note: For numbers less than 1000, the first zero is always suppressed.

EMBEDDED FUNCTION EXAMPLES

Perhaps the most powerful feature of the keyer is its ability to store functions within messages. The stored commands are executed as encountered when the message is played. A consecutive serial number, for example, is needed in several contest exchanges. An ARRL Sweepstakes exchange, for an answering station, could be programmed as exemplified by: "NR /N/GØ A KCØQ 80 IA BK". A second message might contain a serial number repeat message. This makes use of the decrement function: "/D NR /N BK".

Speed changes within messages are also permitted, allowing parts of a message to be played at differing speeds. Using relative changes as in "/SU15 QRZ DE WØSR/7O UP 5 /SD 15" plays the message faster but then returns to the previous operating speed.

Messages may call other messages. Suppose message 4 contains "WA9CNS/KH7". Then the message "CQ CQ CQ DE /4 /4 K" will, when played, yield "CQ CQ CQ DE WA9CNS/KH7 WA9CNS/KH7 K". Call nesting may be as deep as desired. Thus, message #2 can call message #1 which calls message #4 which calls message #3! In fact, continuous loops can be programmed. A loop will result if a message calls itself (directly or via some other message).

Loops can be very useful. A CQ loop in message 1, using a pause, is a good example: "CQ DX CQ DX DE WØWP WØWP K /P35 /1". The message will play continuously, with 3.5 seconds in between. When an answering station is heard, simply tap either paddle to cancel the loop.

If you are fortunate enough to be able to "run 'em" in a contest, the break function and looping can be a real advantage. Message 1 might contain: "QRZ TEST DE WØWP /B 599 IA BK /B /1 ". Here, activating message 1 first causes "QRZ TEST DE WØWP" to be sent. The /B breaks the message and allows the responding station's call sign to be copied. The call is then sent via the paddles. The message then automatically continues and sends "599 IA BK" followed by another break. The other station's report is then copied while the keyer waits. A simple "TU" or "R" is then sent via the paddles to acknowledge the exchange. This completes the break and causes an automatic loop back, restarting the whole sequence.

If no one responds to the QRZ, simply tap button 1 again to resend the QRZ (remembering that a /B is canceled by a button closure). If the responding station's exchange is missed during the second break, use message 2 to ask for a repeat: "AGN? BK /B /1". After getting the repeat, acknowledge with "TU" or "R" as before, and the QRZ loop is automatically resumed.

Setting up a beacon message can be useful for certain functions. Here would be a typical beacon message attached to message #11: "/S2Ø TEST TEST DE KØHGB KØHGB /X /P5Ø E /1". This would send the message "TEST" twice at a speed of 20 WPM, then sign the call, then send a 5 second carrier. The "E" at the end will not actually go out over the air, but will serve to "break" the key-down condition, then the message will loop back to the start, and recycle ad infinitum.

User Note: If the keyer gets excessive RF on the input or output leads it is possible that the keyer will hang up, and operate improperly or not at all. In such cases, the user should first try removing power from the keyer for 30 seconds, then reapply. If that does not cure the fault, do a complete reset by pressing buttons 1-3-4. If this fails to correct the keyer's problem there is some other fault.

FUNCTION, INQUIRY AND DEFAULT SUMMARY

Function Command	Operation	Default Setting	Inquiry Mode	Range
A	Autospace - toggle Autospace on or off	A	A	
B	Bank - select message bank d or disable	Ø	B	-
D	Decrement - subtract one from serial #	-	-	-
Ed	Edit - begin editing existing message d	-	-	-
Fdd	Function Speed - use dd wpm for entry	ØØ	F	05-30
H	Hand Key - Output follows paddle	-	-	-
Kdd	Keying Compensation	ØØ	K	0-25
L	Load Mode - Toggle character/real time	char	L	C/R
M	Monitor - toggle monitor on or off	on	-	on/off
Ndddd	Number - set serial # to dddd	ØØ1	N	9999
Q	Queue - toggle input queue on or off	on	Q	on/off
Rddee	Range - limit speed from dd to ee wpm	05-40	R	05-60
Tdd	Tone - Set monitor frequency to ddØ Hz.	70	T	50-99
Vd	Variant - select keying emulation d	Ø	V	0-9
Wdd	Weight - set weight to dd percent	50	W	25-75
X	Xmit - continuous key down	off	-	-
Zd	Zeros & nines - use serial # option d	Ø	Z	0-9
?	Inquire - current setting or msg. content	-	-	-

TUTORIAL

Logikit CMOS-4
by
Idiom Press
Box 1025
Geyserville, CA 95441-1025

Getting Started

To use your Logikit, you will need several cables. The keyer paddle cable and the transmitter cable should be shielded, with the shield used as ground. The keyer requires either 12 Volts DC from a 2.5 mm power plug, or can run off the internal battery pack. If you have the batteries installed and also hook up an external power source, the batteries will not be used, as long as external power is supplied. Be sure the external plug is 2.5 mm; 2.1 mm plugs that look identical will not work! Many modern transceivers have a 12 volt DC accessory jack which is ideal. Before hooking up the keyer, measure the voltage to be sure it is less than 14.5 volts, and is DC! **DO NOT** use hobby store wall mounted transformers.

DO NOT use your Logikit CMOS-4 keyer with cathode-keyed rigs without an isolating relay. (This normally applies only to rigs made before about 1970)

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. If the keyer does not thus respond, and everything is hooked up and powered properly, press buttons 1,3, and 4 down simultaneously, then release. This resets the keyer EEPROM. The keyer should now respond with the "OK", and you can proceed.

One thing the keyer does not know at start-up is the position of the speed control knob. Turn the speed knob all the way clockwise and send a few dots. This calibrates the speed range. 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.

Reversing Dots and Dashes: Send some code from your paddle, and adjust the speed to your liking. Are the dot and dash reversed from what you want? (Most right handed operators prefer the left paddle for dots and the right paddle for dashes.) No problem! Simply press the outside two buttons, buttons, #1 and #4, down at the same time, and release. The keyer will respond in Morse with an "R", as in Reverse. You have just electronically reversed the keying leads. And it will stay that way until you reverse it again by pushing those two buttons again.

Emulations: You say you are used to a Curtis type "A" timing, or to a keyer with no dot or dash memories, and already you find you can't send code comfortably? Not to worry! Let's fix that now by changing emulations. (if you like the feel of the keyer already, as most do, skip the next two paragraphs.) To do so, first look at the "V" function table printed in the Operating Manual. 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 "A" emulation. Press buttons #1 and #2 down simultaneously, then release them. The keyer will respond by sending an "F". Now, using your paddle, send "V6". There. 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, by pressing buttons #1 and #2 again, then sending the "V6" after the "F" prompt.

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.

Memories: Your keyer can store long messages in each of four active memories. If you use only the four memories, you can store messages of up to 255 characters in each memory. That is a lot of message! And if you choose to configure the memory into 12 messages (of which more later) you still have storage good for 85 characters per message available. Now, let's store a message in memory. Press the far left button down, (button #1), 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 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 “di-di” (a 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. 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 go get a cup of coffee and return to the shack. It will still be waiting for you to send the next word for the message.

Let's go on and send "brown fax". Whoops. We wanted "brown FOX" didn't we? Do we have to start all over again? Nope. Instead, send the international "I goofed" symbol - a stream of seven or more dots. The keyer will send back to you "brown". This tells you it has erased the incorrectly sent word (In this case "fax") and backed up to the word before, "brown". It sent the word "brown" to remind you where you are in the message, and is ready for you to resume loading the message you want. So now send "fox".

OK, that’s all the message. Close it by simply pressing button #1 down momentarily, then release. To play the message, press button #1 again briefly, and listen to the keyer send the message you programmed, complete with 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 on message #1, "AB1CD", and on message #2 "DE AB1CD". On 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'll pick that up 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 several seconds until you hear the tone. The message is now erased. Release the button, then either enter a new message, or press the button again to close the empty memory.

Monitor: Suppose you want to shut off the monitor when you go on the air, so you can use the transceiver sidetone instead? Simple. Press the left two buttons (one and two) down together, then release them. The keyer will send back to you the letter "F". (This stands for "Function".) Now, send the letter "M". 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 through 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 or 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 not a valid command, it will send a raspy signal, a Bronx Cheer. That means you goofed. Simply press the two left buttons again, and resend the "M". If the command sent in error had been a valid one, but not what you had intended, 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.

Tune-up: Need to close the key to tune up the rig? Press buttons #1 and #3 together, then release them. The keyer will send an "H", as in "Hand-key." Now, any time you press either side of the 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 returned 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 of 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 so far.

SECOND SESSION

OK. You've been using your Logikit CMOS-4 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 commands available to you.

Inquiry Mode: Let's try the "Inquiry" mode. Push the right two buttons (#3 & #4) 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 paddle. The keyer sends back a number - the speed in WPM the keyer is presently set for. Press buttons #3 & #4 again. (The keyer automatically leaves the inquiry mode after each question, so for each new inquiry you must re-enter the inquiry mode, by pressing buttons #3 and #4.) Now send a "Q" through your paddle. 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 does not affect the setting, The Function mode, on the other hand, would have 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". The keyer will play back the message (if any) presently stored in memory number one. You can read 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 the 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.

Function Mode: Now let's work with the "Function" mode, always activated by pressing the left two 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, used to control many other keyer functions, some of which we are now going to explore.

Speed Range: Let's reset the speed range. This will give you a good example of how the keyer is programmed at the same time. You never send slower than 10 wpm,

and rarely go above 45 wpm. So 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 *ddee*". The "*dd*" stands for the low range limit, the "*ee*" for the high speed limit. So the command we want is "RØ85Ø". That is, I want a range of 8 to 50 words per minute. Now let's enter that command.

Enter the Function mode by pressing the left two buttons, #1 & #2. The keyer responds by sending "F" in Morse. Now, send the "RØ85Ø" command. The keyer now knows what speed we want, but is not sure where the speed knob is set. Turn the knob fully clockwise and send a few dots. That allows the keyer to calibrate itself. Now set the knob anywhere you want and start sending, with the range tailored exactly as desired. (Be sure to note: To enter a speed below 10 WPM, you must enter a leading zero such as Ø7 for 7 WPM when sending the "Range" command.)

But suppose you really screwed up, and programmed in a range of 50 - 60 words per minute. But your top speed is 35 wpm, and now you can't reset the range to a more manageable number? Not to worry - there is an escape hatch, Simply press buttons #1, 3 and 4 down together, then release. This causes a complete reset. You lose all stored settings and memories, but you are back in control.

Function Speed: 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 Function mode is identical to the regular keyer speed. But, if you like, the function speed can be set at a fixed value, with an available range of 5 - 30 WPM. This feature could be valuable if you like sending at high speeds, for example, but want a more deliberate speed for the function or inquiry modes. To set the function mode speed at a fixed value, enter the function mode (always by pressing down buttons #1 and #2, then releasing) and send "F1Ø". This will set your function mode speed at 10 WPM. Immediately after you have finished sending the "1Ø" 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, 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 again be the same as regular 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 it has turned the queuing function off. Now, press button #1, release, pause a moment, then press button #2. Memory #1 will start playing, but the instant you push button #2 message #1 will terminate, and the message stored in memory #2 will begin transmission. To restore queuing, again enter the function mode and again send the "Q" command. The keyer will respond with "ON" indicating that queuing has been restored.

Weighting: 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%, normally ideal. And for most users it is. But the weighting can be easily and precisely modified. To do so, enter the Function mode. Answer the keyer's "F" response by sending "W 3Ø" from your paddle. Then send your call sign. 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 call again. Yup, it sure did change... Now you will likely prefer to return the weighting to 50%.

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, some prefer heavier weighting, using perhaps 55% or 60%. Again, operators at slower speeds, particularly in the 6 - 10 WPM range, may prefer a weighting of perhaps 45% or 40% for a more pleasant sound. If you are new to such matters, the best advice is to restore the weighting to 50% and leave it until you have a specific reason to change it.

The weighting percentage the keyer is set for can also be queried through the Inquiry mode. To check, simply press the right two buttons (#3 and #4) to enter the Inquiry mode, and answer the "?" prompt by sending a "W" through your paddle. The keyer will send the spacing percentage it is presently set for.

QSK Correction: 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. One major transceiver, for example, subtracts 18 milliseconds from all dots and dashes. This gives the listener to your rig an impression of light weighting, especially at higher speeds.

Such induced errors can be cured by setting weighting to a heavier value. But such a correction is correct at only one speed setting. Your Logikit Keyer offers a specific correction for this problem, the "K" function. To correct an 18 millisecond error, enter the Function mode, then send the command "K 18" through your paddle. The "K 18" adds 18 milliseconds of transmit time to every transmitted dot and dash, thereby canceling out the keying error at any speed setting!

But what correction is appropriate for your rig? Ask stations on the air how your weighting sounds, particularly if it is set at 50%. If other operators tell you it is light, try adding say 4 milliseconds (K Ø4) and ask for further reports, preferably from the same station. Remember, it takes only a few seconds to change the setting for an experiment. Note too that you may need to use a different setting when you switch your linear on or off.

The "K" function setting can be read from the "Inquiry" mode if desired by responding to the "?" prompt by sending the letter "K" through your paddle.

Also, during normal playback through the monitor, the effect of the "K" factor is masked by the keyer. However, when a stored message is played back through the Inquiry mode, the extra compensation may become noticeable.

Auto Spacing: 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 others abhor it. What auto spacing does is force the operator to leave at least three space elements between transmitted letters, so that with auto character 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 by 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 spaces 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 Logikit CMOS4 default is to have auto spacing off. To turn it on, enter the function mode, then answer the "F" prompt by sending "A". Since auto-spacing is a toggled 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 answer the "F" prompt by again sending the letter "A" through your paddle.

The on or off state of the auto-spacing switch can also be checked by using the Inquiry mode. Respond to the "?" prompt with the letter "A". The keyer will answer with either "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 your Logikit CMOS4 Keyer. You will almost surely find the implementation far friendlier than that of any other keyer you have ever used, and you might just change your mind. Give it a chance.

Monitor Tone: While we're playing around, let's try resetting the monitor tone. The default setting is 700 hertz. Enter the Function Made, and enter "T9Ø". Suddenly, you will find a monitor note of 900 hertz. The command is "T*dd*" where "*dd*" represent the first two digits of the monitor frequency. If you don't like 900 Hertz, a "T7Ø" will restore the monitor to the default. The available tone range is 500 hertz to 990 hertz.

Message Load Mode: Another available function command offers a choice between "Character Mode" or "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 that 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 missent 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 message to be loaded. They also are more convenient.

There is an alternate load mode available, however, called "Real-Time". In Real-Time mode, what you load is mirror-imaged back to you, warts and all. This mode can be useful if you have a special characteristic way of sending something, usually your call sign. For example, if your call is WY9IE, 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 image of what you send to the memory, you can no longer correct loading mistakes with a string of dits.

The default load mode is Character Mode. The command switching the load modes is a toggle command. To switch, enter the function mode, then answer the "F" prompt by sending "L", which will toggle you into the opposite load mode, and indicate the new mode with a "C" or an "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" for Character mode, or an "R" for Real-Time mode. Note that some messages can be stored in Character mode while others are stored in Real-Time mode – your keyer will store both at the same time.

You can also ask the keyer which Load mode it is in through the Inquiry mode. Simply enter the inquiry mode by pressing down the two right buttons, and answer the "?" prompt by sending "L" with your paddle. The keyer will respond with either a "C" or an "R".

Note that when loading Real-Time messages the keyer will feel as though a crude form of auto-spacing is on. What happens is that the keyer clock does not resynch with each paddle closure in this load mode. This is necessary to maximize memory usage.

There is one other point about loading messages in character mode. Using the embedded "G" command, the operator can tailor the space between two letters in stored memories. But we will examine that later.

Message Banks: Now let's discuss the Bank command and how the keyer uses memory. You will remember that in the default mode the keyer has four memories, each capable of storing messages up to 255 characters in length. However, the keyer also offers the option of dividing the memory into 12 messages, each capable of storing 85 characters each. (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 4 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 12 messages. "B1" allows you access to four 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 4 new empty messages. Going to bank 3 by the Function command "B3" gives you the last four.

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 CQWW 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, like 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 SESSION

Embedded Commands: Although the features you have learned thus far already make your Logikit CMOS4 Keyer very powerful, 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.

An embedded command 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 than Real-Time mode. Embedded commands always follow a word space "I", and start with a "/", followed immediately by the command.

For purposes of this tutorial only, a "_" in the following text 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 call sign in if the "/" is used for entering an embedded command? Simple. As long as the slash bar is NOT preceded by a word space, the keyer recognizes it as a slash bar. So if you are signing WY9IE/KH7, no problem. Just key it into the memory. (On the other hand, if you want to program in WY9IE_/KH7 with a word space pause that can be done with the Gap command, which we will learn shortly.)

Calling a message from another message: Let's try a message with an embedded command. First, let's call one message from inside another. First load into memory #1 your call sign "DE WY9IE AR". Now, load the following into message #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.

Loops: Load a new 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. To stop the message, all you have to do is touch your paddle, or press any two buttons and release.

Insert one message into another: Load into message #2: "12345". Then, load this into message #1: "ABC_DEF_GHI_/2_JKL_MNO". Now play message #1 back. See how message #1 picked up the embedded call to message #2, then, when message #2 had run its course, message #1 continued?

Embedded Pauses: Now, let's examine the embedded Pause command, called by "/P*dd*", where *dd* stands for an operator selected number between ØØ and 99, and where each digit stands for one tenth of a second. So, load this message into memory #1: "CQ_CQ_DE_WY9IE_WY9IE_AR_/P5Ø_/1", then play it back. Now you should have the CQ message play, sign your call, and then after a five second pause, repeat again. And again. And again etc. The /P5Ø is your embedded command into the message to pause 5 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, each doing its thing. This closed loop message with a pause for listening is great for beacons, CQing on a relatively dead band, Meteor Shower skeds etc . Try "ABC_DEF_GHI_/P50_JKL". 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 back in normal keyer mode.

Beacon Messages: 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 WY9IE/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" is 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 message, touch the paddle or press any two buttons.

Inserting Hand-Sent Text in a Message: Now, let's program a message that would include hand-sent text, like the other station's call in a contest exchange? For that, embed a "/B" command, "B" as in "Break". Load this message: "UR_RST_/B_DE WY9IE" and play it. Notice that the message play back 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 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 dumb.

A similar command, "R", as in "Resume", allows more timing tolerance for hand sent entry. Try putting an "/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 memory button again, at which point the message will resume. Both multiple "/R" and

multiple "/B" commands may be entered in a single message.

Embedded Speed Control: The embedded “/S” command lets you set the speed in a message. Try: "/SØ6_CQ_CQ_/S12_CQ_CQ_/S24_CQ_CQ_/S48_DE WY9IE_AR", and play back. For speeds under 10 WPM, you must use a leading zero, as in "Ø6." When message playback is complete, the keyer speed will revert back to the knob setting. If you wish to revert to the knob setting in the course of a programmed message that already includes a speed command, use a command "/SØØ". This causes the message following the command to play at the knob speed.

A pair of embedded speed controls will probably be very useful for many contest-oriented and DXpedition operators, the /SU*dd* and /SD*dd* commands. These commands increase and decrease the speed of the keyer from its current setting. These commands will be especially useful in exchanges with "canned" responses, like "59904" or "599KW".

Load this message: "/SU15_59904_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 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 an "/SU" command you will almost certainly want to also use the complimentary "/SD" 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".

GAP Control Another embedded command can be useful for a number of functions - the "/G*d*", or "Gap" 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, that 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 "/G*d*" 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 as a single dot.) The "/G*d*" 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 G*d* 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. Try programming "WY9I_/G2_E" and see how it sounds. Experiment with different values for /G*d*. Once you understand it, you will almost surely use it for one special situation or another.

Serial Numbers: It's time to leave the subject of embedded commands and examine serial number generation, primarily useful for contest operation. First, you must decide what number format you want to use. Do you send zeros in contest serial numbers using five dashes, three 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 "Z*d*" function command lets you specify your choice. Examine the Z Option table in the accompanying instruction 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 "F" 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". If you do want to preset a number, say 1066, enter the function mode again, and answer the "F" prompt with the command, "N 1Ø66". Note that you must use a five-dash zero, as in "1Ø66" even though the keyer will send it as a "T" under the "Z6" function. Now, enter the inquiry mode by pressing buttons 3 & 4, and respond to the "?" prompt with an "N". 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 "ØØ23".

So how do we use the serial number in a contest message? Right - with an embedded command. Let's enter: "NR /N A WY9IE 80 IA BK". This will give you "NR 0023 A WY9IE 80 IA BK". "But," you say, "There is a word space between the serial number and the "A". Which is wasted time in a contest where time is of the essence." Right. So let's eliminate that with an embedded /G*d* command. Reprogram the message thus: "NR_/N_/GØ_A_WY9IE_80_IA_BK". Now we have eliminated the word space between the serial number and the "A".

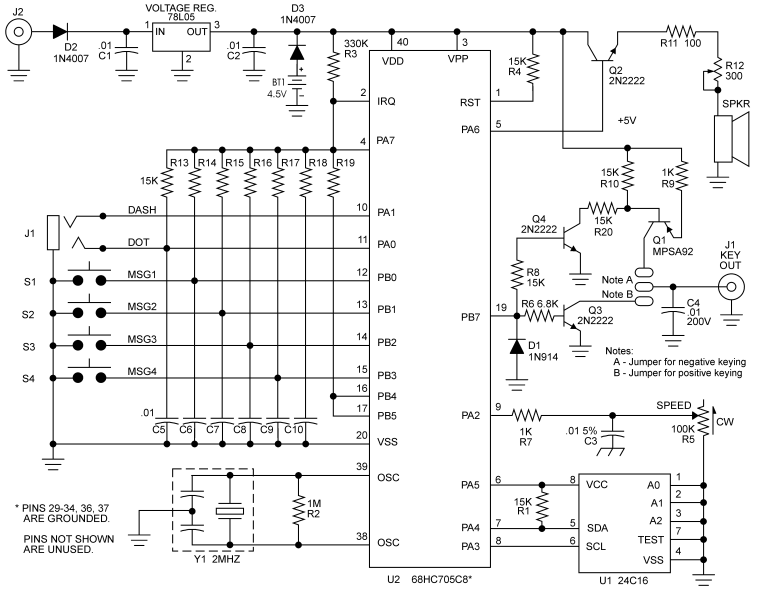
Decrement Serial Numbers: But, you say, "What do I do when some clown calls me for a duplicate QSO and I send the serial number before I realize he is a dupe? Do I have to reset the serial number with the 'N' function command? That would be a pain!" And indeed it would be. But you don't have to reset the number. All you do is press the middle two buttons, (2 & 3) and the keyer sends a "D" and will automatically decrement (subtract) one number! That's all there is to it. And each time you press buttons #2 & #3, you decrement another number in the stored serial number.

Tune-Up: There is another button function we have not covered - another tune-up approach. Pushing buttons 2 & 4 together and releasing them 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 your paddle, which releases the rig.

Ultra-Speed: 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 received 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 "/U*dd*" 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 the message at 900 WPM. Note that weighting, compensation (K) and tone settings are suspended during ultra-speed transmission. Ultra-speed may be used as a 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 for exact details.

Examining Stored Messages : There are two different ways to examine the contents of a message stored in memory without going on the air. Load a #1 message with embedded functions, such as an "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 the Inquiry Mode and this time answer the "?" prompt by sending a "1" through the paddle. Note this time that the playback of the stored message is exactly as you loaded it, showing you the slashbars and embedded commands, exactly as you entered them. You will find each way of examining a stored message valuable at different times.

Editing Messages: 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 end of the message until you reach the 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 then send the next prior word for you to decide to keep or erase. At any point you may close the message and the editing



Note: For negative keyer rigs, the internal jumper goes to the outside two pins of the header, for solid state rigs use the two inside pins of the three pin header.

Much thanks goes to Tom Hammond, NØSS, and to Howard Nurse, W6HN, for their considerable assistance in beta testing the Logikit keyer, in the preparation of the manual and this schematic.

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 operating 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 the two center buttons is so much easier it would be a rare operator that prefers the function mode entry for this command. 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.

So. That concludes the tutorial for your Logikit CMOS4 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!

Troubleshooting

Your CMOS4 Keyer has been carefully engineered to avoid problems.

Most problems stem from one of two problems - excess power supply voltage, and excess exposure to RF fields.

If a voltage higher than 14.5 volts is applied, the keyer may or may not survive. Also, the warranty is voided when excessive voltage is applied. Note that many wall mounted transformer power suplies that claim to be 12 volts DC put out as much as 20 volts, often with a very high AC component. Such a supply can damage your keyer, and will void the warranty! If there is any doubt at all, measure the output of your proposed power source with an accurate voltmeter. If you get in excess of 14.5 volts, do not use that supply!

If the keyer stops working and you suspect that the maximum voltage has been exceeded, first remove the power from the keyer. Make whatever changes are necessary to insure a proper DC voltage is supplied, then re-apply power to the keyer. If the keyer does not respond with its "OK", press buttons #1, 3, and 4 down simultaneously and release. If the keyer still fails to respond, one or more of the chips has probably been destroyed. Contact Idiom Press for instructions and repair.

Should excessive RF get into the keyer, it is possible that the CPU will "lock-up" or "crash", even thought the design uses CMOS circuitry which is well known for RF immunity. Such crashes are easily corrected by pressing buttons #1, 3, and 4 down simultaneously and releasing them. However, all settings in memory will be lost, including stored messages, the speed range setting and the like.

If the problem regularly repeats itself when transmitting on the air, there is a grounding problem somewhere. Make sure all cables are shielded, and that the integrity of the shielding is OK at both ends. Cables should be as short as possible if you are having RF problems. High SWR's often contribute to excessive RF in the shack, so addressing an antenna problem may cure the problem. Until the problem is resolved, operating at reduced power levels may be a temporary "fix." Although they should not be necessary, the user can put .1 uF capacitors across the paddle base between the dot post and ground, and the dash post and ground.

For any other problems, contact Idiom Press, P.O. Box 1025, Geyserville, CA 95441. Include a stamped addressed return envelope and a careful description of the problem. Or e-mail to info@idiompress.com You will receive a prompt response. Also, Idiom Press will repair non-functioning keyers for a reasonable fixed charge which will be quoted in response to your inquiry. The fixed charge repair will not apply to keyers which have had over-voltage applied or have been damaged by lightning strikes near the station. In these cases, quotations for repairs will be made after receipt of the keyer and examination.

Logikit CMOS-4 Keyer Limited Warranty

This Logikit Keyer is warranted against manufacturing defects in material and workmanship for 90 days from the date of purchase from Idiom Press or from an authorized Idiom Press dealer.

This warranty does not cover damage or failure caused by or attributable to Acts of God (such as nearby lightning strikes), abuse, misuse, attempts to plug in wrong connector plugs, improper or abnormal usage, faulty construction, use of acid core or water soluble resin solders in construction, faulty installation, improper maintenance, application of excessive voltage, or improper repairs other than those provided by Idiom Press.

Further, this warranty does not cover damages caused by switching or keying circuits where the current to be switched or keyed exceeds 20 milliamperes.

Idiom Press is not responsible or liable for indirect, special or consequential damages arising out of or in connection with the use or performance of the product or other damages with respect to loss of property, loss of revenues or profit, or costs of removal, installation or reinstallation.

Except as provided herein, Idiom Press makes no express warranties, and any implied warranty of merchantability or fitness for a particular purpose is limited in its duration to the duration of the written limited warranties as set forth herein.

Always contact Idiom Press at info@idiompress.com before returning a keyer. Always include a letter carefully describing the problem, and a return shipping label. Ship the keyer to:

Idiom Press
P.O. Box 1025
Geyserville, CA 95441