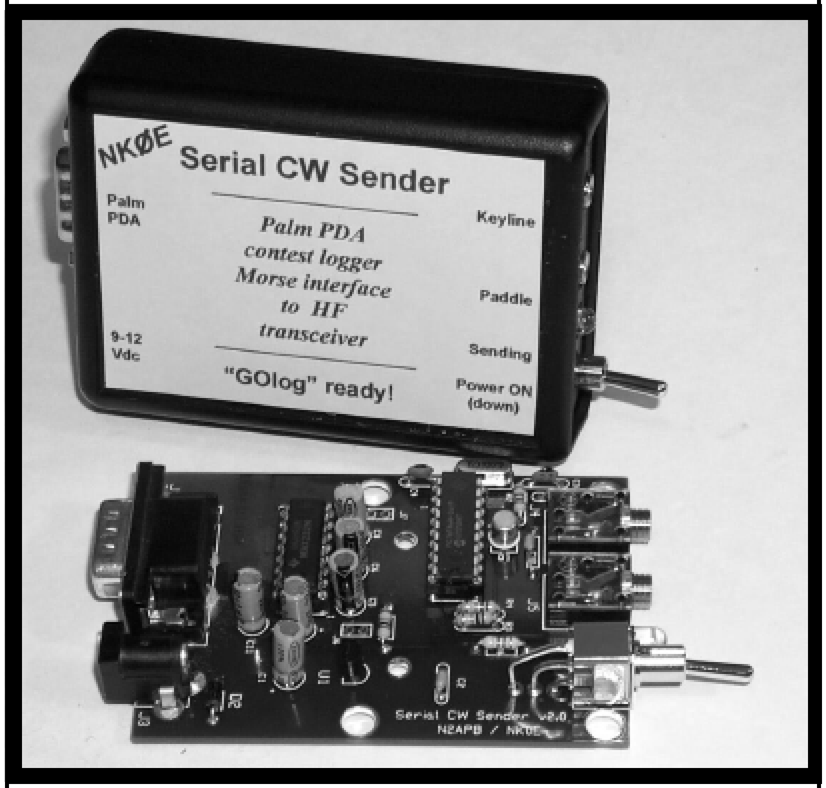
NK0E

Serial CW Sender



Designed by: **David Ek, NK0E**

Kitted by: **the NJQRP Club**

NK0E Serial CW Sender

by David Ek, NK0E *nk0e@earthlink.net*

Here's a fun and useful accessory to take along on your next Field Day operating adventure. The Serial CW Sender connects between your Palm PDA and your transceiver to provide automated CW keying while using the GOLog contest logging software on your handheld. Additionally, by plugging a standard paddle into the Serial CW Sender, you can manually key your transmitter to help nab those elusive stations!

Introduction

The NKØE Serial CW Sender was designed specifically for use with the GOLog ham radio contest logging software for handheld computers running the Palm OS®[1] operating system. The Serial CW Sender allows GOLog users to automate contest CW keying while contesting from the field. It was developed primarily with the QRP contester in mind. The GOLog software is available for free from

http://home.earthlink.net/~golog

The GOLog user's guide covers the operation of GOLog, and that information won't be repeated here. The user's guide also describes how to use the Serial CW Sender with GOLog.

Besides using it for handling keying for GOLog, you can plug a paddle into the Serial CW Sender and handle the manual keying that you must inevitably do during most contests. Whenever you use the paddle to send, any keying being done from GOLog is interrupted. It's not really practical to use a paddle with the Serial CW Sender unless you're controlling it from GOLog, though, since GOLog is used to control the keying speed, even for the paddle.

Requirements

To use the Serial CW Sender with your handheld, you'll need a serial cable connection between your handheld computer and the Serial CW Sender circuit. Palm™ sells serial HotSync® cables for most if not all of its handheld computers and these should work for connecting the Sender to your Palm handheld. *USB HotSync cables will not work.*

Users of Handspring™ Visor™ handheld computers must take extra steps to use the Serial CW Sender with their handhelds. Handspring sells both USB and serial HotSync cradles for their handhelds. The Sender cannot be used with a USB HotSync cable or cradle, even with a USB-to-serial adapter. Furthermore, Handspring serial HotSync cradles are not compatible with the interface. The reason is that the Visor itself outputs TTL signal levels instead of RS-232. The Handspring HotSync cradle contains circuitry to convert the TTL levels to RS-232 for the computer. However, the Handspring HotSync cradle expects to draw its supply voltage from the device to which it's connected (normally, a PC). The Serial CW Sender does not provide this voltage, though, so the cradle won't work with it. However, there are third-party cables available that contain the necessary level-conversion circuitry and also are wired to draw their power from the Visor rather than from the external device, and these cables should allow Visor users to connect successfully to the Sender. Try Mark/Space (http://www.markspace.com/) or Purple Data Cables (http://www.pcables.com/) for the necessary cable.

Circuit Description

The schematic diagram for the Serial CW Sender is shown in Appendix A. Appendix B gives the list of parts for the Sender. In the circuit, U1, C1, and C2 make up a regulated +5V supply for the rest of the circuit, and any voltage greater than about 8V may be supplied as input to the regulator at J3. D2 is a polarity-protection diode to protect the Sender from damage if the voltage supply is connected backwards. S1 is the on/off switch.

U2 is a programmed PIC16F84 microcontroller. The PIC chip is clocked by crystal X1 and requires C3 and C4 to be present (the values of C3 and C4 need not be exactly 22 pF—any value around 30pF or so should work). The frequency of X1 is 4.0 MHz and other frequencies should not be used in its place due to timing considerations for the serial communications. The pins labeled RESET (J7) can be shorted temporarily to provide a soft reset for the PIC chip by grounding U2 pin 4, which is normally held high by R1. If desired, you can connect a normally open momentary SPST switch here to serve as a reset button.

Pins 6 and 18 of U2 serve as serial input and output, respectively. They are connected to U3, a MAX232 chip that converts the TTL voltage levels from the PIC to the RS-232 levels (-12V and +12V) expected by the attached computer. C11 serves to additionally filter the voltage supply for U3. Capacitors C5 through C8 are used by U3 to create the necessary RS-232 voltages given the +5V supply. J1 is a standard male DB9 connector and connects directly to the Palm handheld using its HotSync cable. Pins 1, 4, and 6 on J1are tied together, as are pins 7 and 8, to simulate any handshaking signals that the handheld might expect.

Pin 1 of U2 keys the CW on the attached transceiver. Q1 is used to close the circuit on the keyline just like would be done with a straight key. R2 limits the current flowing through the base of Q1. C9 and C10 help to filter out any RF generated by U2's 1 MHz internal clock. Pin 2 of U2 is used to drive LED D1, which will flash the CW being sent by the Sender. D1 and R6 can be omitted to reduce current consumption, if desired (although the savings will likely be small).

Pins 8 and 9 are connected to the dit and dah levers on a paddle for manual sending. Pressing either the dit or dah paddle grounds the associated pin on U2 and causes dits and dahs to be sent. R3 and R4 hold the paddle pins high when the levers are not pressed. Pin 7 on U2 is grounded (by shorting the MODE pins at J6) to select Curtis mode A keying, or held at +5V using a 10K resistor (R5) for mode B.

Firmware

The firmware programmed into the PIC chip handles three main functions. First, it handles the serial communications with the handheld computer. All serial communications take place at 9600 baud, eight bits, one stop bit, no parity, and no flow control. The commands sent by the handheld to the Sender are given in Appendix C.

Second, the Sender firmware handles keying the rig by grounding the keyline to form dits and dahs. The Sender translates the ASCII code for each character that was received from the handheld into the correct sequence of dits and dahs, and keys the rig accordingly. Note that GOLog sends a string of characters in a single command, and then the Sender can key those characters without further interaction with GOLog. This frees GOLog (and the user) to do other things (like finishing log entries) while the Sender is keying.

Finally, the Sender firmware also handles the manual keying done when the user connects a paddle and uses it to send. Any time the user depresses a paddle, any sending currently being done is interrupted. There is no provision for adjusting the speed of the manual keyer other than by setting the speed from within GOLog.

Although the PIC chip in this kit has already been programmed, the source code and hex code for the firmware are also available on the GOLog web site. The PIC chip can be reprogrammed if NKØE releases an updated version of the firmware. You'll need a PIC programming circuit and software to do so. There are many such circuits and programs available on the web, or they can be purchased commercially.

Construction

Construction of the Serial CW Sender is straightforward. First, you should inventory the parts you received using the parts list in Appendix B. Make sure you can identify each of the parts to ensure you get it into its proper place on the PC board. You can then begin construction. Experienced builders can go in any order they please, but the following procedure is recommended:

□ Install the IC sockets for U2 and U3, making sure that you insert them so that the notch in one end of the socket lines up with the notch in the parts outline on the PC board. This will help to make sure you don't install the chips backwards later. When installing a socket, solder only the two opposite corner pins first. Then turn the board over and make sure the sockets are correctly aligned and seated flush on the board. If necessary, reheat each of the two soldered pins so you can seat the socket firmly. Once the socket is seated correctly, solder the remaining pins. Do not install the ICs yet.

□ Install all the resistors. R1, R3, R4 and R5 are 10K (brown-black-orange). R2 is 1K (brown-black-red), and R6 is 470 ohms (yellow-violet-brown).

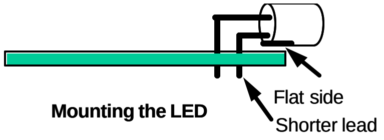
□ Install the electrolytic capacitors. C1 is 10uF, and the remaining electrolytic capacitors (C5 through C8) are 1uF. Electrolytic capacitors are polarized. The positive lead is the longer of the two, and the negative lead is marked on the body of the capacitor with a light stripe and minus signs. Make sure you insert the positive lead into the hole marked as positive on the board.

□ Install the remaining capacitors. They are not polarized, so either lead may go in either hole. C2, C9, and C10 are 0.1uF ("104") and C3 and C4 are 22 pF ("22").

□ Install crystal X1. Space it up by about 1/32" to ensure its case doesn't touch the two pads. Try not to use too much heat on the crystal—no more than a few seconds on each lead with the soldering iron.

□ Install Q1, the PN2222 resistor, and U1, the 78L05 voltage regulator. Make sure you install them in the orientation indicated by their outline on the PC board. (Q1 may be supplied as a tabbed-case device, so just orient its 3 leads in the same pattern as the pads on the board.) As with the crystal, try not to overheat these semiconductor devices—only a few seconds on each lead with the iron.

□ Install D1, the LED, so that its lighted end is toward the edge of the board. Carefully bend the leads as shown below and mount with the flat side of the plastic body against the pc board.



Mounting the LED

□ Install D2, the 1N4001 diode. Note that there is a silver band on one end of the diode body. Insert the diode so that the banded end corresponds to the band on the diode's symbol on the board.

□ Install the DB9M connector J1. Space it up about 1/32" to best fit in the case later. (Try using a scrap piece of thin cardboard as a temporary spacer while soldering J1 in place, then remove the cardboard.) J1 pins are closely spaced, so make sure you don't create any solder bridges. After installing J1, install the jacks for the keyline, the paddle, and the power.

□ Turn the board over and examine the solder side for any unsoldered pins, solder bridges, or poor solder joints. Use a magnifying glass to inspect each joint. Solder joints should be smooth and shiny. If any joints look bumpy, uneven or dull, retouch them with the soldering iron. Usually, a bit more heat will make them look smooth and shiny.

□ Before installing U2 and U3 into their sockets, apply power to the circuit and use a voltmeter to test for +5V at pin 14 of U2 and pin 16 of U3. If both measure at 5V, disconnect power go to the next step. If you didn't measure 5V, disconnect power and see the troubleshooting section.

□ Insert U2 and U3 into their sockets. Make sure that the end of each chip with the notch or dimple is oriented with the notch in the corresponding socket. Before inserting the chip in the socket, you may need to bend the pins inward slightly to align the pins with the sockets. Once inserted, make sure that all pins made it into their sockets, and none were bent underneath the chip.

Enclosure

The pc board for the Serial CW Sender was specially designed to fit in the Hammond 1593P plastic enclosure supplied in the kit. Since all components are board mounted, you just need to cut appropriate holes in the front/back end panels of the enclosure, as indicated in the photos.

Testing

Probably the easiest way to test your Sender is to connect it to your handheld and run GOLog. First, with power disconnected, connect the Sender to your handheld using your serial HotSync cable. Next, using a 1/8" stereo or mono phono cable, connect the key jack from the rig to the keyline in the Sender. Then connect your rig to a dummy load. Finally, plug your paddles into the sender using a 1/8" stereo phono cable. Then apply power to the Sender and attempt to send with your paddles. If that works, run GOLog on your handheld. Consult the GOLog users guide for instructions on how to create a new log and use the Serial CW Sender, and give it a try. If you can send with GOLog and with your paddles, you're all set.

You can also test the Sender by hooking it up to a PC using a null modem cable and attempting to communicate with it using a communications program like Hyperterminal. Set up the terminal program to communicate directly through the serial port to which you've connected the circuit (rather than through a modem), and set the communications parameters to 9600 baud, 8 bits, 1 stop bit, no parity, and no flow control. Make sure the circuit is powered up, and then type "^" (shift-6, or ASCII 94) in the terminal program. If everything is working correctly, you should see something like "Serial CW v1.17 © 2002 by NKØE" appear in the terminal window. If not, see the Troubleshooting section. (Note that the Sender doesn't echo the commands you type—you'll have to turn on the "local echo" feature in your terminal program to see what you type, if you feel the need.)

Troubleshooting

This is a fairly simple circuit, so problems usually aren't hard to find. One note: the source of the problem is very rarely a faulty chip. It's much more likely to be a bad solder joint. Here are some things to check:

□ Did you connect the Sender to the Palm and the rig correctly? Make sure you didn't mix up the connections.

□ Did you apply power to the Sender? If so, are you getting +5V at U2 pin 14 and U3 pin 16? If not, you need to check the installation of U1, C1, and C2, and also check the polarity of the power connection.

□ Make sure all the solder joints are good—no partially-soldered or unsoldered leads, and all the joints are smooth and shiny. Use the soldering iron to touch up any suspicious-looking joints. Bad solder joints are by far the most frequent reason for circuit problems.

□ Is Q1 installed correctly (as indicated by its outline on the board)?

□ Are U2 and U3 installed in the proper orientation?

Battery life

When you're in the wilderness, battery life is a definite consideration. I've powered the Serial CW Sender with a 9V transistor battery and received several hours of operation. For Field Day in 2001 I carried two such batteries for the interface and probably could have gotten by with only one.

The battery life of your handheld is also a consideration. Some models, like the Palm m100, get many hours of use from two AAA alkaline batteries, while more expensive models with color displays use rechargeable batteries but have shorter lives because of their increased current consumption. The serial ports on these handhelds also consume power when in use, so the GOLog software was coded to keep the serial port closed except when actually in use.

Contacting Us

If you discover any missing parts from your kit, please contact the NJQRP Club at:

Dave Porter, AA3UR
647 Middle Holland Road
Holland, PA 18966
email: njqrp-kits@comcast.net

If you have technical questions, please contact the project designer:

Dave Ek, NK0E
5605 Oro Grande Drive
Colorado Springs, CO 80918
email: ekdave@earthlink.net

Please be sure to check the NJQRP Club website pages for the Serial CW Sender project. We often post usage tips, construction techniques, kit errata, application notes, etc. for our various kits.

www.njqrp.org/serialsender

References

Palm is a trademark of Palm, Inc. Palm OS and HotSync are registered trademarks of Palm, Inc. Handspring and Visor are trademarks of Handspring, Inc.

APPENDIX B: Parts List

QTY	Part	Description	Label
1	C1	10 uF 35V electrolytic capacitor	“10 uF 35V”
3	C2, C9, C10	0.1uF monolithic or ceramic disc capacitor	“104”
2	C3, C4	22 pF monolithic or ceramic disc capacitor	“22”
5	C5-C8, C11	1 uF 50V electrolytic capacitor	“1 uF 50V”
1	D1	LED	
1	D2	1N4001 diode	1N4001
1	J1	male DB9 connector	
1	J3	5.5x2.1mm coaxial power jack	
2	J4, J5	1/8” stereo phone jack	
1	Q1	PN2222 transistor	“PN2222”
4	R1, R3-R5	10K ¼ watt resistor	Brown-black-orange
1	R6	470-ohm 1/4 watt resistor	Yellow-violet-brown
1	R2	1K ¼ watt resistor	Brown-black-red
1	S1	SPDT power switch	
1	U1	78L05 voltage regulator	“78L05”
1	U2	PIC16F84-04/P microcontroller	“PIC16F84-04/P”
1	U3	MAX232CPE level converter	“MAX232”
1	X1	4 MHz crystal	“4.000”
1		Hammond Enclosure	1593P
1		18 pin IC socket	
1		16 pin IC socket	

APPENDIX C: Communications Protocol

Command Character sent to the device	Data sent to the device after the command	Reply from the device	Description
< (ASCII 60)	up to 54 characters, followed by a carriage return (ASCII 13)	r (ASCII 114), after sending is complete	causes the device to key the characters as CW on the attached transceiver. The following characters are supported: A through Z (upper or lower case) 0 through 9 period (.) comma (,) slash (/) BT prosign (=) AR prosign (+) question mark (?) SK prosign (*) KN prosign (:) AS prosign (-) Other characters will be ignored. Characters beyond the 54 currently supported will be ignored until the terminating carriage return is received.
> (ASCII 62)	two characters	r (ASCII 114), immediately after receiving the two characters	sets the inter-dot and inter-character spacing. To get the appropriate values, divide 1300 by the desired words per minute. The characters sent are those whose ASCII values yield the desired speeds. For example, “A” has an ASCII value of 65 and will give a speed of approximately 20 wpm (1300/65). The first character sets the speed at which the dits and dahs are sent for a single CW character. The second character sets the spacing between CW characters (like in Farnsworth spacing). Note that speeds are approximate and not precise.
^ (ASCII 94)	none	a string of characters terminated by a carriage return (ASCII 13)	returns a string describing the firmware version.
Any other character	none	r (ASCII 114), after sending of the current CW character is complete	interrupts CW keying. The CW character currently being sent is finished, and the remaining characters are forgotten. If no sending is in progress, this command is ignored.