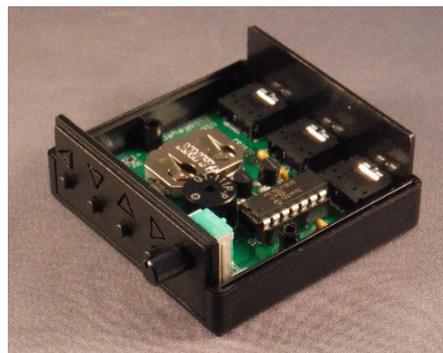


# HamGadgets

## Ultra PicoKeyer

### Morse Code Memory Keyer

- Dual MOSFET keying circuit will key transmitters up to 60 V, positive or negative
- Simple, convenient pushbutton interface with Morse code prompts
- Eight 127-character message memories can be chained together for longer messages
- Works with any dual lever iambic paddle, single lever keyer paddle or straight key
- Dot and dash memories, automatic timing and element spacing
- Easy message entry using your paddle
- Simple feature setup with no paddle needed
- Auto straight key detect, all message memories available with straight key
- “Bug” mode allows automatic dots with manually formed dashes
- Speed adjustable from 5 WPM to 60 WPM via menu OR speed control potentiometer
- Speed control can be set to your preferred speed range
- Dual-Set Speed allows quick QRS/QRQ and return to favorite speed
- Adjustable weight
- Variable pitch audio sidetone
- QRSS mode with 1-255 second dots
- Curtis “A” or “B”, “bug” and Ultimatic keying modes
- Variable letter spacing
- Tune mode with 50% or 100% duty cycle for easy tuneup
- Beacon mode with repeat delay up to 255 seconds and optional power-on auto start
- MCW mode for sending audio Morse over voice radios
- Memory “pause” command with automatic resume allows manual insertion of RST etc. into message
- Auto-incrementing QSO / serial number can be embedded in memory messages, with or without leading zeros
- Paddle switching - select left or right handed operation
- Transmitter QSK delay compensation
- Memory and parameter settings retained with power off
- Low voltage - from as low as 2.5 V to 3.5 V
- Low current - typically under 1 mA when keying, with automatic extreme low current sleep mode
- All controls and connectors on board
- Two MYCALL memories for storing your callsign or other information



## Operating the Ultra PicoKeyer

Once your Ultra PicoKeyer is assembled, mounted in the cabinet and has the battery installed, it's time to connect your paddles and transmitter or transceiver and take a little time to read this manual to learn how to use your PicoKeyer's features.



This is the **connections** end of your PicoKeyer. The center jack is for your paddles (input); the right hand jack is the on/off output for keying your transmitter or transceiver (output). The left hand jack is a low power audio output that can be used for ear buds or headphones.



This is the **controls** end of your PicoKeyer. The speed control knob is on the right, the memory/setup pushbuttons are on the left. The triangle symbols ◀, ▼, ▲ and ▶ are used with the setup menu system. The buttons are not marked with numbers, but are numbered from 1 through 4 from left to right for message memories.

### The Speed Control

This is the control you will probably use the most often. The code speed can be adjusted from 5 words per minute (WPM) to 30 WPM by simply turning the speed control knob. The high and low limits can be set using the setup function described later in this manual. The default high and low range settings are 5 WPM and 30 WPM, but you can set these anywhere between 5 WPM and 60 WPM.

You may notice that the speed will decrease to 5 WPM as you rotate the knob counter-clockwise, then suddenly jump to 13 WPM. This is a special feature known as “Dual-Set Speed”.

Two Morse code speed settings are available for your use. The *current speed* is the speed at which the keyer is operating, regardless of whether that speed is derived from the speed pot or the menu. The *stored speed* is saved in the chip's internal non-volatile memory, and is used when the speed pot is turned to its minimum position. At any time you may use the speed adjustment to set any speed between the low and the high range setting. Setting the speed with the pot does not affect the stored speed, which will always be available by simply turning the pot to its full counter-clockwise position. This feature can be very useful; for example, you can keep your speed set at a constant, favorite speed, but have the option of rapid QRS/QRQ to answer faster or slower stations. Or, you can simply ignore the stored speed and always use the pot for speed control – it's up to you.

Two things to note: First, you can set the stored speed anywhere from 5 WPM to 60 WPM – and it *doesn't* have to be within your speed control range. You can also set the range of the speed control to whatever is best for you. Second, if you set the stored speed and low range the same, it effectively disables the dual-set speed feature – you'll have a constant range of speed adjustment. See the Range (RL & RH) commands in the “Setup Menu Commands” section.

## The Pushbuttons

**Special Note:** If you try to send something with your paddle and get an unexpected result, you may still be in setup mode. To exit setup, push buttons 3 and 4 (▲ and ►). If you hear “dit dit” you’re back in normal operating mode.

The Ultra PicoKeyer has four memories, controlled by the four pushbuttons on the front panel. Operating these buttons is very simple and straightforward, and their use is as you might expect from a modern electronic device.

### Playing a Stored Message

Pressing and immediately releasing any of the four buttons once (a “tap” of a button) will send the message stored with that button, if there is one stored. If you want to stop a message while it’s being played, just tap either paddle OR tap the same message button again.

Messages can be “stacked” or queued. If a message is playing, pressing any other message button will queue that message to be played as soon as the current message is finished. You can stack up to four messages. Tapping either paddle will always terminate all message playback.

### Using Keyer Features

There are several functions that can be used by pressing multiple buttons at the same time.

- **To enter Tuneup Mode**, press buttons 1 & 3 (◀ and ▲) at the same time. In Tuneup Mode, you can tap the dash paddle to send a steady carrier or tap the dot paddle to send a string of dits. Tap either paddle to stop sending. Tap any button to exit Tuneup Mode. While in Tuneup mode, the sidetone will always be ON regardless of the sidetone setting as a reminder that you’re transmitting.
- **To decrease the QSO counter by one**, press buttons 1 & 2 (◀ and ▼) at the same time. The keyer will send two “dits” using the sidetone only to let you know it decremented the QSO counter.
- **To check the current operating speed**, press buttons 2 & 4 (▼ and ►) at the same time. The keyer will announce the currently selected speed in Morse code, using the sidetone only.
- **To send the contents of the MYCALL memory**, press the outer two buttons (◀ and ►) at the same time.
- **To check for a straight key** and adjust for straight key or paddle use, press buttons 2 & 3 (▼ and ▲) at the same time. The keyer will check to see if either paddle input is grounded, and if so it will treat the other input as a straight key. If neither input is grounded it will assume a paddle is in use. The keyer will announce “K” to let you know it’s ready for normal operation.
- **To enter Setup Mode**, press buttons 3 & 4 (▲ and ►) at the same time. See the “Setup Mode” section for information about how to change settings. When you’re finished in setup mode, just press buttons 3 & 4 together again to exit. The keyer will automatically exit setup mode after roughly 30 to 35 seconds of inactivity.

### Recording, Reviewing and Saving Messages

Don’t let the following description fool you – recording, reviewing and saving messages is very simple and intuitive! It would take less time to demonstrate than it will to read these instructions. The buttons on your Ultra PicoKeyer are powerful little things and very easy to use.

To record a message: To record a message in the currently selected message bank, press and hold one of the pushbuttons to record the message associated with that button. Simply hold the button until you hear the Morse code “K” prompt (about two seconds). You are now in **recording mode**. Record your new message using the paddle as you normally would. When you’re done recording, press and hold the button again to save the new message. The keyer will announce “R” to let you know it recorded the message.

To exit without making changes: If you decide you've made a mistake and want to start over, or if you were just reviewing an existing message, or if you accidentally entered recording mode, simply tap any button *except the one you just pressed* to stop recording and return to normal operation without saving the message. The keyer will announce "NR" to let you know it *did not* record the message.

To review and add to a message: If you just want to hear what you have stored in any of the message slots, just enter recording mode (hold the button down until you hear "K", then release the button. Now *tap* the same button to play back your message. This works either before you start to record a new message (to review one already stored), or after you record a message to make sure you've recorded what you want before saving. Once you review the message, the "pointer" in memory is not set to zero – in other words, if you review the message and start recording, what you send will be appended to the end of the message.

To erase an existing message: If you want to erase a message you have stored in one of the memory locations, it's easy. Enter recording mode (hold down the button until you hear "K"), then release. Now press and hold the button down again until you hear "R". This will erase anything stored in that message location.

Let's walk through a couple of typical uses of the memory buttons. First, let's assume you want to *record a new message* in memory location #1.

1. Hold down button 1 until you hear "K", then release the button.
2. Send your message using the paddle. You can pause as long as you want between words.
3. If you want to review what you've sent so far, tap button #1.
4. Once you're satisfied with the message, press and hold button #1 again until you hear "R".

Now let's say you think you have a message stored in M1, but you're not sure what it is. You want to *review* the message and possibly change it, but you don't want to key the transmitter. Here's what you would do:

1. Hold down button 1 until you hear "K", then release the button.
2. Tap button 1. The keyer will play the stored message (if there is one) without keying the transmitter.
3. If you want to append something to the existing message, you can do it just by sending with the paddles before you exit.
4. If you want to leave that message as it is, just tap button 2, 3 or 4. The keyer will respond with "NR" (think: "Not Recorded"). If you changed the message and want to save it, hold the button until you hear "R".

Now let's say you want to *replace* this message with a new one:

1. Press and hold button 1 as before until you hear "K".
2. If you want to record a new message, just use the paddle to send the new message.
3. To review what you've just sent, tap button 1 again. If you want to add to it, you can.
4. Once you're satisfied with the message, press and hold button 1 until you hear "R". Now you're done and out of recording mode.

So let's say you want to just *delete* message #1.

1. Press and hold button 1 as before until you hear "K".
2. Press and hold the button again until you hear "R". The message is now erased; in other words, you just recorded an empty message.

The **MYCALL** messages are a special case; they must be recorded using the setup menu, and they can each hold up to 63 characters instead of 127. Other than that, they work just like the other four messages

### **Special Message Commands**

Several special embedded commands may be used in messages. All commands start with a slash followed by one or two characters. When playing back a message in setup mode you will hear the command itself, not its effect – message chaining, QSO numbers, pause and beacon mode are inactive while in setup mode. For example, you will

hear **/R** instead of the word to be repeated. **To store a slash character in a message, save it as //**. For example, if I want the message to send N0XAS/B I'd store it in memory as N0XAS//B.

- **/R** will repeat the last word, including the word space after it. This can save a lot of memory space, since each **/R** takes up only two character positions in memory. For example, to send a 3x3 CQ, you can simply store "CQ **/R/RDE** (call sign) **/R/RK**". This can save a lot of memory space.
- **/1, /2, /3 ... /8** can be used to "call", or insert another stored message as part of the one you're recording. The indicated message will be played immediately when one of these commands is encountered, then the current message will resume.
- **/P** will pause the message. This will cause the keyer to wait while you manually send information such as a signal report or other comment. The message will automatically resume after a full word space has passed with no paddle input. *Hint:* If you use **/P**, store it immediately following the preceding characters without a word space. In other words, store "UR RST**/P** ..." instead of "UR RST **/P** ...". This prevents you starting to send before the word space completes, which will terminate memory playback completely. If you want to terminate a message while it's paused so it won't resume, simply tap any message button.
- To have your message automatically repeat at timed intervals, insert the command **/B** (BEACON) at the end of your message. This will cause the keyer to delay for the number of seconds set with the **B** parameter (see the setup menu instructions) and re-send the message. You can terminate beacon operation by tapping either paddle or any button. This can be especially useful for calling CQ, or to use your PicoKeyer to control a propagation beacon or "fox" transmitter.
- To have Message #1 automatically start whenever power is applied to the keyer, store the **/A** command as the first two characters in message #1. This is useful for automatically starting a keyer used as part of a beacon station. Remember that you will still need to use **/B** at the end of the message if you want it to repeat.
- To send the QSO number and increment it by one, send **/QI** (QSO & Increment).
- To send the QSO number and NOT increment it, send **/QN** (QSO & No increment).
- To the last (previous) QSO number, send **/QR** (QSO Repeat). This is useful during contests if you need to send a "fill".
- To temporarily **increase** the keyer speed by one WPM, send **/SU** (Speed Up). Note that this and the **/SD** command will take effect immediately and will remain in effect only until the message is finished. You can store multiple **/SU** or **/SD** commands to change speed by more than one WPM – for example, **/SU/SU** will increase your speed by 2 WPM.
- To temporarily **decrease** keyer speed by one WPM, send **/SD** (Speed Down). This command works exactly the same as **/SU**. **/S** or **/S0** will resume normal speed.
- To temporarily **set a specific speed**, send **/Snn** where **nn** is the speed you want. **/S** or **/S0** will resume the normal speed. For example, to send a signal report at 30 WPM and return to the normal speed you would use **/S30 599/S0**.
- To set **QRSS mode**, send **/SQnn** where **nn** is the number of seconds per dot. **/S** or **/S0** will cancel QRSS mode. Example: **/SQ3DE N0XAS/S0** will send "DE N0XAS" with 3-second dots.
- To alter the **letter spacing**, send **/Fn** where **n** is one digit, 0 through 9. This will act the same as setting the letter spacing in the menu. For example, say you want to add a little extra space between letters in your call sign in a CW message. You could store, "CQ **/R/R DE /F2N0XAS /R/R K /F0/B**".
- To insert an extra **word space** in your message, use the special prosign character "**IM**" (.------).
- To insert a steady **carrier**, use the **/Cn** command, where **n** is the number of seconds (from 1 to 9) that you wish to send the carrier.
- To insert the contents of a **MYCALL** message, insert the command **/Y, /Y1 or /Y2**. **/Y** or **/Y1** will send MYCALL #1; **/Y2** will send MYCALL #2.

## Battery and External Power

Note that your Ultra PicoKeyer does not have a power switch! The microprocessor “brain” of the keyer will go into a low-current sleep mode after about a second or so with no inputs from either paddle or any of the buttons. In sleep mode, the keyer chip draws so little current – a few microamperes -- that the internal battery can last for years. As soon as you touch a paddle, the chip instantly “wakes up” and continues its normal operation. You won’t be able to tell the difference when the keyer goes to sleep; the transition from power-saving sleep to operating is instantaneous and requires no action from you.

The battery in your Ultra PicoKeyer should last for a long time – anywhere from several months to several years. Battery life depends on a few factors:

- Frequency of use. Obviously, the more you use the keyer, the shorter the battery life will be. Even with regular use, you shouldn’t have to replace the battery more than once a year or so.
- Sidetone. The sidetone speaker draws more power than the rest of the keyer functions. Using the sidetone speaker will reduce battery life – but you should still only have to replace it once in a great while.
- Beacon mode. The keyer does not sleep between messages while in beacon mode. If you use your PicoKeyer to control a beacon full time, you should use an external power source. This can be as simple as a pair of series AA or AAA alkaline cells. Remember to remove the coin cell before connecting an external power source, and never allow the external source to exceed 3.3 V – and of course, double-check the polarity before applying power!

Battery replacement is straightforward – just remove the two screws from the cabinet bottom, remove the keyer circuit board from the cabinet, slide the coin cell battery out of its holder and install the new battery. All of your settings and messages will be retained even with no battery power.

If you should need to replace the battery in your keyer, you will need a CR2032 or equivalent lithium cell. These are very common and are used in many devices from garage door openers to car remotes to thermometers and medical monitors. You can order high quality CR2032 cells from [www.HamGadgets.com](http://www.HamGadgets.com). Don’t use any battery or power source that exceeds 3.3 volts.

## Straight Key Mode

The Ultra PicoKeyer can detect and use a properly wired straight key. During its power-on program, the PicoKeyer checks to see if either paddle input is grounded. If one input is shorted, the other input is assumed to be a straight key. This way you can plug in a straight key wired to a mono plug and use it without any changes or adjustments. Be aware that while the setup menu will still function and you can send messages, you will not be able to record messages while using a straight key.

Of course it’s not always convenient to pull out the battery when you switch between a paddle and a straight key. To detect paddles or a straight key, just press buttons 2 and 3 (the middle two) at the same time. It will automatically re-scan the paddle inputs and determine whether or not a straight key is plugged in (you will hear “P” if a paddle or no key is detected, or “K” if a straight key is detected). If you use a straight key with a stereo plug, you may find it more convenient to switch between paddles and a straight key using the **K** menu setting.

## Sidetone

The PicoKeyer’s sidetone is a square wave audio signal generated by the microprocessor. Sidetone can be turned on or off, and the audio frequency can be changed from the setup menu. There is a very noticeable peak in the response of the on-board speaker at roughly 2 kHz; if your PicoKeyer is installed in the plastic cabinet you’ll

probably want to find this peak so you can hear it with the case assembled. The default setting should be at this audio peak.

## Using headphones for sidetone

If you prefer, you can use a 3.5 mm stereo plug to connect an external device for sidetone. This will disconnect the on-board speaker. The PIC processor chip used in the Ultra PicoKeyer can drive small ear buds or headphones with 32 ohms or greater impedance. Low impedance phones will result in very low or no sidetone volume. For larger speakers or lower impedance headphones, you'll need to use an external amplifier. An inexpensive amplified PC speaker may work well, or you can use this as an opportunity to build a low power audio amplifier. An LM386 amplifier chip will work well for this, as well as many other types.

You will need to use a stereo plug for an external audio device. If you need to use a mono plug, you can cut the PCB trace on the bottom of the board under the PHONES jack. It's the trace that runs between the two outer-most pads of the jack. This will work fine with mono headphones, and will give you sidetone in one ear with stereo 'phones.

USE CAUTION when trying ear buds or headphones! Some ear buds or headphones may be uncomfortably loud, or even loud enough to damage your hearing. NEVER use a headset or earbuds directly on or in your ears until you know how loud the sidetone will be. If it's too loud, you can try a smaller value capacitor in place of C3. You can also connect an external variable resistor to use as a volume control.

## Keying the transmitter

The PicoKeyer will key any solid state, tube or hybrid transmitter or transceivers that uses a keying voltage of **60 V or less**, either positive or negative.

Your transmitter or transceiver may need either a mono or stereo cable. It depends on the rig; check your owner's manual for details for connecting a straight key to your transmitter. The PicoKeyer's output acts like a straight key, so you'll use the method for connecting a straight key. Remember to disable the rig's internal keyer, if it has one.

If you intend to use the PicoKeyer with a rig requiring grid-block or cathode keying voltages over 60 V, you will need to use a separate high voltage keying adapter. The Universal Keying Adapter 3 available from NØXAS at [www.hamgadgets.com](http://www.hamgadgets.com) is optically isolated and will handle solid-state, grid-block or cathode keyed transmitters at up to 400 V. Of course, you can always build your own!

## Keying Modes (What's Mode A, Mode B and Ultimatic??)

There have been a couple of different operating modes for iambic keying that have evolved over the years. Modes A & B are simply a matter of when the keyer checks for input from the paddles. In iambic mode A, the keyer only checks for paddle inputs after the end of each dot or dash. In iambic mode B, on the other hand, the keyer will check for paddle input during each dot or dash.

In practice, this can mean that you get "extra" or "dropped" dots or dashes at the end of a character, depending on how you send. If you find that the keyer often drops the last dot or dash in a character, or you often get an extra dot or dash at the end of a character, try switching between modes A & B and see which one best suits you. I find that Mode B worked best for me when using a single-lever paddle, while Mode A works best with a dual-lever paddle.

Ultimatic mode is a different way of handling iambic keying. In modes A & B, if the keyer sees both paddles closed it will alternate sending dots and dashes. Ultimatic, on the other hand, will send dots or dashes *according to the last paddle to be pressed*. For example, to send the letter P in mode A or B, you would close the dot paddle,

then close the dash paddle and release the dot paddle for the two dashes, then release the dash paddle and close the dot paddle for the last dot. In Ultimatic mode, you would close the dot paddle and hold it closed, close the dash paddle for the two dashes, then release it for the last dot. Some letters are easier to send and require less effort using Ultimatic mode. It's a little bit of an adjustment from regular iambic keying; it took me a couple of hours of practice to get used to it.

Selecting a keying mode (A, B or Ultimatic) is largely a matter of personal preference. There is no one "right" way that works for everyone; find which works best for you.

Two additional keying modes are available with the Ultra PicoKeyer. "Bug" mode (mode G in the setup menu) offers automatic dots and manual dashes, emulating a mechanical semi-automatic key. Straight key mode (mode S in the setup menu) allows completely manual sending, treating each paddle as a straight key input. This can also be used in the same way you would a "cootie" key – like two straight keys back to back.

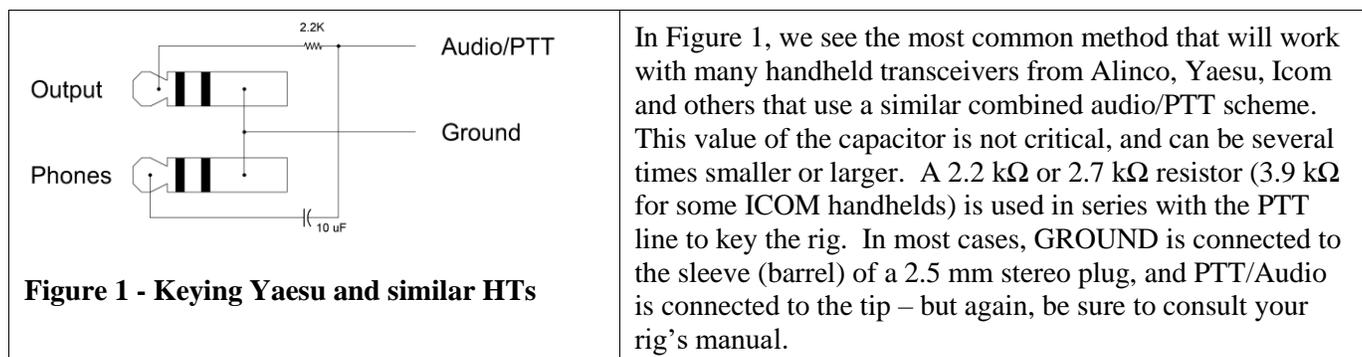
Keep in mind that you can always send stored messages, but you will not be able to record messages while in bug or straight key modes.

## Using Your Keyer with a Handheld or FM Rig ("MCW" Mode)

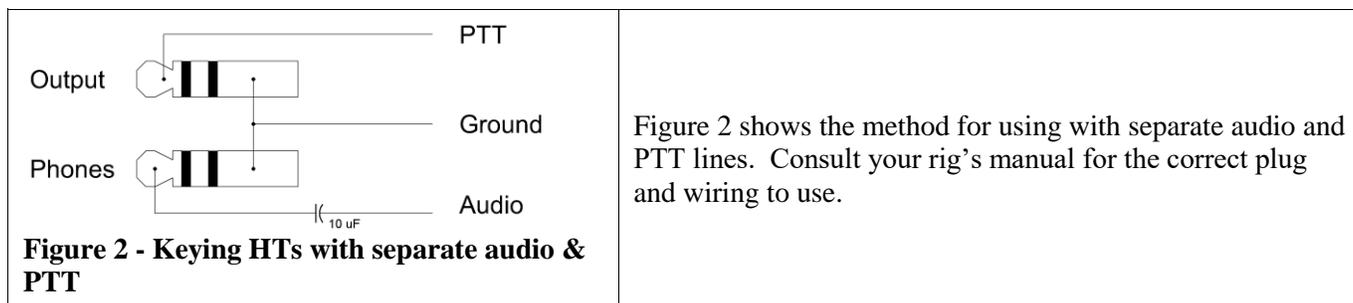
One of the unique features of your Ultra PicoKeyer is MCW mode (menu option "T" for sidetone, select "M" for MCW). In this mode, the keying output can be used to trigger the PTT input of a VHF or UHF handie-talkie or FM mobile rig, allowing you to use your FM rig for Morse code. This is not CW (Continuous Wave) operation, nor is it true MCW (Modulated CW) mode – it's really just an FM transmission, with audio Morse code tones sent instead of voice. It is a good way, though, to have a CW practice net using a local repeater – with the blessing of the repeater operator, of course – or a simplex frequency. Those participating don't need HF privileges or HF equipment to learn and practice Morse code on the air.

It is not possible to give detailed instructions for attaching your keyer to every rig on the market. Different transceivers, even from the same manufacturer, may require different setups to work well. You will need to take a look at your rig's manual to determine how the audio and PTT signals from the keyer need to be connected. Look for instructions for connecting a packet radio TNC for a good start. Doing it wrong could possibly damage your rig and/or your keyer! Refer to the operator's manual for your rig for specific requirements for PTT and audio.

Using MCW mode will probably require making a custom cable to connect the keying and sidetone outputs to your FM transceiver. Depending on your radio you may also need to install a shorting jumper across capacitor C3, and/or you may need to use an external variable resistor to control the audio level supplied to your transceiver.



**Figure 1 - Keying Yaesu and similar HTs**



## About Setup Mode

**Just remember:** Enter and exit setup mode with ▲ and ►. Use the ◀ and ▶ buttons to navigate through the menu; use the ▼ and ▲ buttons to change settings. “?” means you’re in setup mode, and “did dit” means you’re back in normal operating mode.

Using the Ultra PicoKeyer’s setup mode is very simple and intuitive, with only a few very simple things to remember.

- To *enter or exit setup mode*, press the ▲ and ► buttons (3 & 4) at the same time. The keyer will send “?” to let you know you’re in setup mode, or “dit dit” to let you know you’ve exited setup mode.
- The ◀ and ▶ buttons (1 and 4) will move forward and backward through the menu items. The menu “wraps around”, meaning you can navigate through the menu in either direction. The order of menu items is shown in the table below. You can move through the menu choices in either direction using the left and right buttons.
- The ▲ and ▼ buttons (2 and 3) will change the selected item’s settings. The selections also wrap around, with a couple of exceptions as noted below.
- *Tap* any button to go to the next item or selection and stop.
- *Hold* any button to scroll through items until you release the button. Holding ◀ or ▶ will announce each menu item in turn; holding button ▼ or ▲ will send a dot or dash for each decrement or increment of the selected item’s setting.
- To *reset* the item to its original “factory default” setting, press and release the middle two buttons (▼ and ▲) at the same time.
- To send the contents of MYCALL #1, if it’s set, press the outer two buttons (◀ and ▶) at the same time.

Many operators will find the default settings of the Ultra PicoKeyer satisfactory for normal operation. However, some people like things set up a little differently. There are numerous settings you can change to suit your operating speed, style or equipment. Probably the most common changes are the speed control range settings and the default speed.

Most people won’t use setup mode often, and it can be a chore to try to remember all of the commands. Others change settings often, and may want the menu operation to be quick and efficient. For that reason, you have a choice between two menu modes – we’ll call them “Long” and “Short”. In “Long” mode, each menu selection is spelled out with a short word like SPEED, WEIGHT or CUT NUM. In “Short” mode, the menu selections are sent as one or two letters only – such as S, W or X. To quickly change between the two modes, you can use the ◀ button to quickly select MENU (▶ will get you there just as easily, but not as quickly). The MENU selection will always be spelled out regardless of the menu mode selected, just in case you’ve selected short mode but have not

used the menu for a while. Once you have MENU selected, tap ▼ or ▲ to switch between “L” (long” and “S” (short) modes.

As an example, let’s look at the S menu item (SPEED setting). Once in setup mode, tap button 4 to advance to the “SPEED” menu item. The keyer will announce the currently set speed. You can now tap button 2 to decrease the speed by one WPM, or button 3 to increase it. Holding either button will continuously increase or decrease the speed, with a dot or dash sent at the new speed for each step. When the button is released, the keyer will again announce the current speed setting. Speed may be set from 5 WPM to 60 WPM. If you want to return to the default 13 WPM setting, simply press and release buttons 2 and 3 at the same time.

## PicoKeyer Setup Menu Commands

Long Version	Short Version	Menu item description
<b>SPEED</b> (Default: 13 WPM)	<b>S</b>	<b>Speed:</b> The keyer will announce the <i>stored speed</i> in WPM. Use buttons 2 & 3 (▲ and ▼) as Down/Up or -/+ buttons to adjust the speed, as described above.
<b>QSO NUM</b> (Default: 1)	<b>Q</b>	<b>QSO Number:</b> The keyer will send the current QSO number. You can use the ▲ and ▼ buttons to set the QSO number anywhere from 1 to 65535. To reset the QSO number to 1, simply press both buttons at the same time.
<b>RANGE L</b> (Default: 5 WPM)	<b>RL</b>	<b>Range Low:</b> This sets the low end of the speed control pot range. The setting is adjusted the same way you would set the stored speed.
<b>RANGE H</b> (Default: 30 WPM)	<b>RH</b>	<b>Range High:</b> This sets the high end of the speed control pot range. It is set the same way as the stored speed and low range setting.
<b>CUT NUM</b> (Default: 0/9)	<b>X</b>	<p><b>Cut numbers:</b> “Cut” numbers are often used in contests to send contact serial numbers and signal reports. The most commonly used cut numbers are 0 and 9.</p> <p>For this item, button 2 will toggle the <i>zero</i> setting, and button 3 will toggle the <i>nine</i> setting. You can select no cut numbers (Ø 9); cut zeros (T 9); cut nines (Ø N); or both (T N). Note that this <i>only</i> affects the way QSO numbers are sent. Other numbers in stored messages, or numbers sent by hand, are not affected. The default setting is no cut numbers.</p>
<b>LTR SPC</b> (Default: 0)	<b>L</b>	<p><b>Auto letter spacing:</b> The keyer will announce the current setting. Automatic letter spacing takes effect for messages sent from memory as well as code sent manually with the paddle.</p> <ul style="list-style-type: none"> <li>• 0 turns automatic letter spacing off. In this mode you control the spacing between letters. This is the default setting; it is the same way most keyers operate and most operators are accustomed to.</li> <li>• 1 turns on automatic letter spacing. A letter space is automatically inserted if the keyer detects no input from either paddle at the end of the space after a dot or dash.</li> <li>• Settings from 2 to 9 will turn on automatic letter spacing with longer delays. For example, selecting 2 will insert one extra “dot” length spacing between characters. Selecting 3 will insert an extra 2 “dot” lengths, and so on.</li> <li>• Note that auto letter spacing is ignored while in setup mode.</li> </ul>
<b>LDG Ø</b> (Default: N)	<b>Z</b>	<b>Leading Zeros:</b> As with several other menu settings, the middle buttons (▲ and ▼) may be used to switch between two settings. Select “Y” to send numbers in stored messages with up to two leading zeros (1 is sent as 001, 99 is sent as 099, 123 is sent as 123, 1000 is sent as 1000). “N” will send numbers without leading zeros. Like the CUT NUM setting, this affects only QSO numbers and not other numbers stored in messages or sent manually. The default setting is N.
<b>BCN</b> (Default: 15 s)	<b>B</b>	<b>Beacon delay:</b> The keyer sends the current default beacon delay in seconds. Use the ▲ and ▼ buttons to set the delay between beacon transmissions from 0 to 255 seconds.

<b>WEIGHT</b> (Default: 5)	<b>W</b>	<b>Weight:</b> The keyer announces the current weight. The ▲ and ▼ buttons may be used to decrease or increase the weight setting. Weight can be set anywhere from 1 (50% "light") to 5 (normal) to 9 (50% "heavy"). The default setting is 5.
<b>SIDE TN</b> (Default: Y)	<b>T</b>	<b>Sidetone:</b> The keyer announces the current sidetone setting. You can use the ▲ and ▼ buttons to switch between “N” (sidetone OFF), “Y” (sidetone ON) and “M” (MCW mode). In MCW mode, the keying output is active any time code is being sent and for two word spaces after the key is released. This can be used to control the PTT line of an FM transmitter. Regardless of the sidetone setting, the sidetone is always used while in setup mode.
<b>KEY</b> (Default: Mode A)	<b>K</b>	<b>Key Mode:</b> The keyer will send the current keying mode: "A" or "B" for iambic A or B timing modes, “U” for Ultimatic, “G” for bug or “S” for straight key. You can use the ▲ and ▼ buttons to switch between modes. In “Bug” mode, dots are made automatically with the correct spacing and length with one paddle input, while dashes are made manually with the other. If straight key mode is selected while using a paddle, either paddle input will key the transmitter.
<b>PADL</b>	<b>P</b>	<b>Paddle Selection:</b> This will allow you to reverse the paddles for mis-wired paddles or left handed operators. Simply hit whichever paddle you want to use for DITs. No need to rewire your paddle!  After selecting this menu item the keyer will announce “DIT”. Tap the paddle you wish to use as the dot paddle; this setting will take effect as soon as you exit the setup menu.
<b>AUDIO</b> (Default: Around 2 kHz)	<b>A</b>	<b>Audio Tone:</b> The keyer will send a dash at the selected sidetone audio frequency each time ▲ or ▼ is pressed. The default setting is approximately 2000 Hz, which gives the loudest audio from the on-board speaker.
<b>DELAY</b> (Default: 0)	<b>D</b>	<b>Transmit delay compensation:</b> Some transmitters tend to shorten Morse code elements when used in QSK mode. This setting can be used to lengthen Morse elements and shorten spaces to compensate. The setting can be from 0 to 50 milliseconds. This is similar to weighting, except that it is independent of speed. Weighting shortens or lengthens elements by a percentage; delay lengthens elements by a specified number of milliseconds. Be aware that this can cause problems when using large delays and fast speeds.
<b>MYCALL</b>	<b>Y</b>	<b>“MYCALL” messages:</b> When in this menu item, the two center buttons (▼ and ▲) will record and play back the two 63-character MYCALL messages. To record the MYCALL #1, press and hold the ▼ button; to hear the message, tap and release. The ▲ button works the same way for MYCALL #2.
<b>BANK</b>	<b>BK</b>	<b>Bank Select:</b> Selects message memory Bank 1 (messages 1-4) or Bank 2 (messages 5-8). The selected bank will be assigned to the four buttons.
<b>MENU</b>	<b>MENU</b>	<b>Menu mode:</b> Long mode sends words for each menu selection. Short mode sends one or two letters only. MENU is always sent in long mode regardless.
<b>VERS</b>	<b>V</b>	<b>Firmware version:</b> The keyer sends the version number of its internal firmware program. This is informational and cannot be changed.

# Assembly Instructions

## Before You Start

Your Ultra PicoKeyer kit was designed with the beginning kit builder in mind. With just a little care and practice, even a first time kit builder can complete the project in a relatively short time. You will need to gather a few tools and supplies together before beginning to assemble your kit. Here's what you will need:

- A clean, level, static-free work area with good lighting. Wooden workbenches are fine. If you are working on a kitchen table, be sure to spread out some newspaper or something else to keep solder splatters and sharp wire ends from damaging the table top.
- A soldering iron. A small, low-wattage (25-35 watt) pencil type iron is ideal. Avoid larger, pistol-grip types. You can find inexpensive irons at your local Radio Shack. You will need a small pointed or "screwdriver" type tip intended for electronics. Be sure to use an iron rest or holder to keep the iron from damaging your work surface. If you plan to assemble more kits, I recommend investing in a good quality, temperature controlled soldering station such as the Weller WES or WLC series. You'll be glad you did! Follow the iron manufacturer's instructions for tinning the tip, and keep a damp sponge handy to keep the tip clean.
- Solder suitable for electronics work. Use a good quality, small diameter rosin core solder intended for electronic assembly. 63/37 tin/lead solder is great for electronics work, but lead-free will work also. DO NOT use acid core solder!
- Small needle-nose pliers and a pair of small diagonal or flush wire cutters. The smaller you have, the better off you will be. Again, you can find hand tools intended for electronics work at Radio Shack and other suppliers such as Techni-Tool, Jensen, Mouser and Sears.
- A clamp or small vise to hold the work is a good idea. I use a PanaVise, but you can also construct a board holder out of scrap wood and rubber bands. If you use a regular bench vise, use gentle pressure and something to cushion the vise jaws.
- A pencil to check off each step as you finish it.

Once you have all of your tools and supplies gathered together, you're ready to get started. Warm up the iron while you remove the parts from the bag and lay them out on the work surface. We'll start with the small parts and work our way through each component, checking them off on the list as we go.

To install a component such as a resistor or capacitor, follow these steps:

1. Hold or gently clamp the PCB with the component side up. The side with the white printing is called the "*component side*" or top; the side with no white lettering is called the "*solder side*" and is the bottom.
2. Bend the component wire leads, if necessary, to fit the spacing of the holes in the PCB. Insert the leads through the holes in the PCB. From the bottom side of the PCB, bend the leads out at about a 45-degree angle to hold the part in place.
3. Working from the bottom of the PCB, solder the leads in place. Remember to place the tip of the iron at the point where the lead comes through the PCB hole, so you heat the wire and the hole at the same time. Wait a couple of seconds for the lead to heat up, then touch the solder to the lead and pad, NOT the soldering iron tip. The solder should flow into the joint. Remove the solder and iron and don't move the PCB for a couple of seconds until the solder has cooled. If you're using tin/lead solder the joint should look smooth and shiny. If it looks dull or rough, touch the tip of the iron to the joint to re-melt the solder. If there is a blob of solder, use some solder wick or a solder sucker to clean it up; re-solder the joint if needed. Lead-free solder joints won't look shiny, but should be smooth and free of cracks.

Be careful not to leave the iron on the joint too long, and don't use too much solder. Electronic

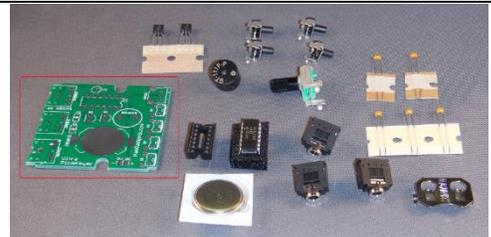
components and circuit boards can be damaged by too much heat for too long. If you have never soldered electronic components before, it would be a good idea to find some scrap parts and PCB and practice on them first. Also, it's a great idea to find someone more experienced to help you learn this skill!

4. Once the part is in place and the solder has cooled, use a pair of fine pointed cutters to trim the excess component leads close to the PCB.

During normal use, the included battery should last at least a year or two (perhaps longer). If you wish to use external power (really only needed for beacon duty), there are a pair of solder pads near the corner of the board marked + and - for a DC power supply not to exceed 3.3 volts. Do *not* under any circumstances connect an external power source without first removing the battery! Connecting external power while a battery is installed can result in component damage and could even cause the coin cell battery to heat up and burn or explode.

## Step-By-Step Assembly Instructions

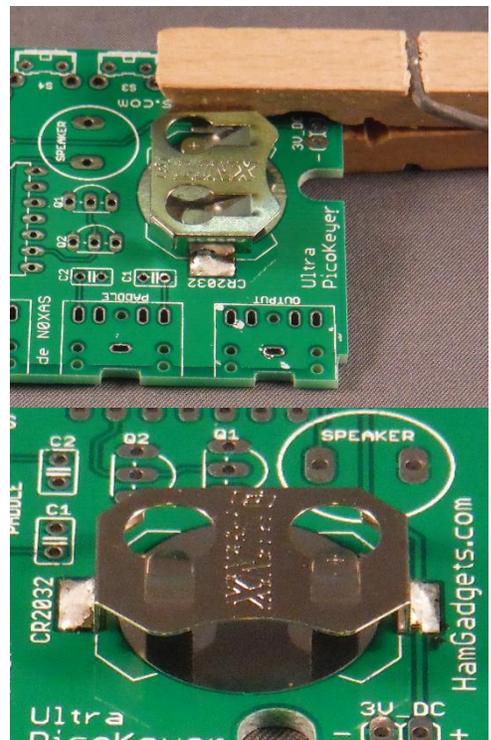
Locate the printed circuit board (PCB). Orient the PCB with the component side (the side with the white markings) on top and the lettering right-side up as you look at it.



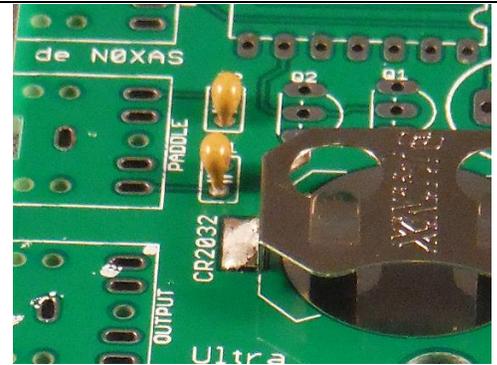
Locate and install the battery clip. Make sure the open side of the battery clip faces the edge of the board. You won't be able to install the battery if you solder it on backwards.

The battery clip supplied with your kit is a through-hole part. You can just solder it in place like any other part. You may want to use a small clamp or clothespin to hold the part in place while you solder. The battery clip will get HOT while you solder, so don't try to hold it in place with your fingers! Tape or rubber bands are probably not a good idea either as they may melt or smoke.

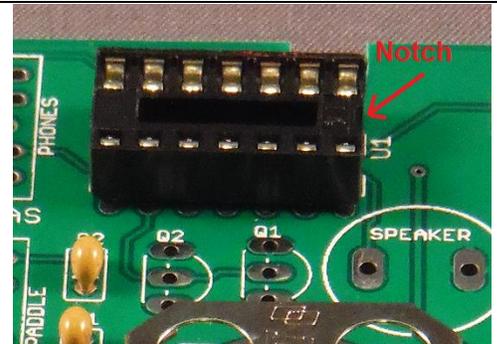
It is not necessary to apply solder to the top side of the board. The pictures on the right show a surface-mount version of the battery clip, which was supplied with only a few of the very early kits. Just soldering the pins on the bottom of the board like any other through-hole part is enough.



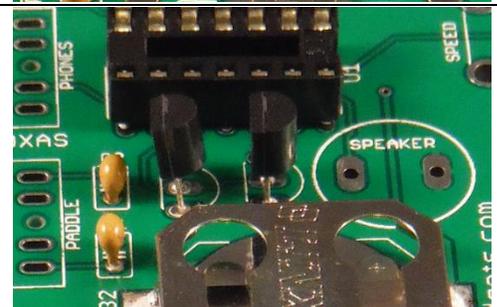
Locate the paper strip with three identical ceramic capacitors in your kit. These will be small rectangular parts with two parallel leads. These three are .01  $\mu\text{F}$  (marked 103). Install two capacitors in the locations shown for C1 and C2. The ceramic capacitors used in your kit are not polarity sensitive (meaning, don't worry – you can't install them backwards).



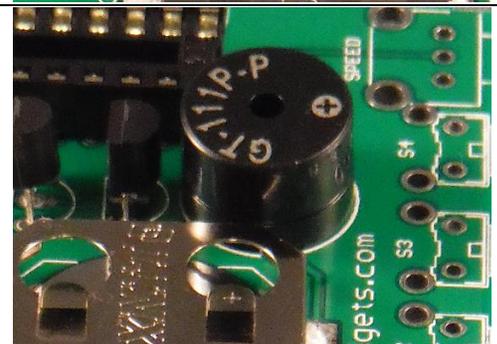
Now find the 14-pin IC socket. One end of the socket has a notch to indicate that it is the "Pin 1" end. Orient the socket so that the notched end matches the silkscreen markings. Line up and insert the pins into the PCB. Make sure all the pins go into the holes in the circuit board. You may need to bend the pins at any two diagonally opposite corners flat against the bottom of the PCB to hold the socket in place while you solder. Solder all fourteen pins in place. Be careful not to use too much heat or too much solder.



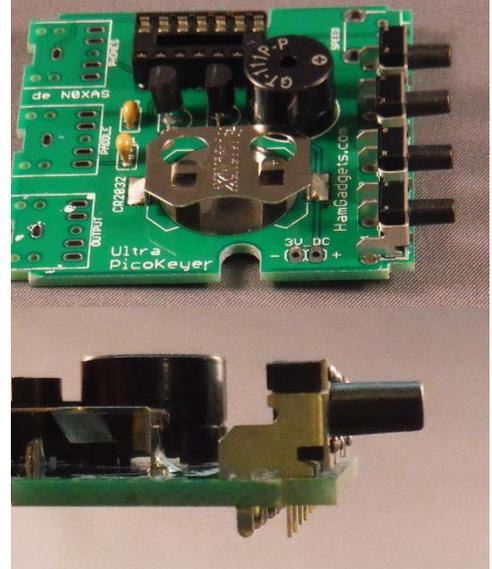
Install the two 2N7000 MOSFET transistors in locations Q1 and Q2. Make sure the flat side of each transistor is facing the direction indicated by the silkscreen printed outline. Don't try to push them in too far. Leave about 1/8" or so between the circuit board and the bottom of the transistor to avoid putting too much stress on the leads.



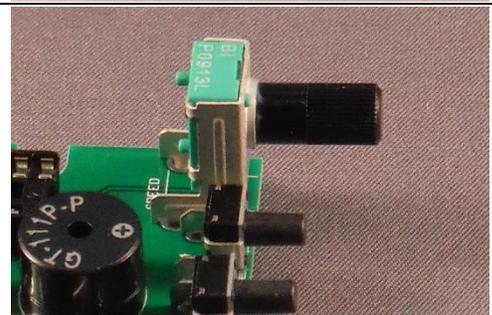
Find the speaker and install it in the location marked SPKR. The speaker is marked with a + on one side. Don't worry about polarity as it can be installed either way. Make sure the speaker is inserted fully into the holes before soldering. Don't bend the leads of the speaker; you may want to use a bit of adhesive tape to hold it in place while you solder it. Don't spend too much time soldering the speaker or it may be damaged.



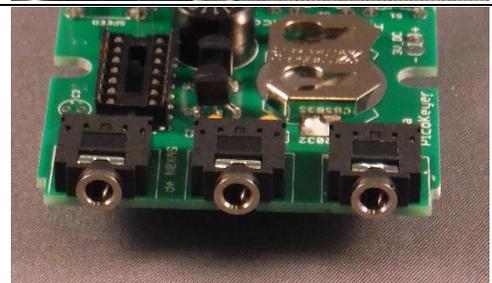
Locate pushbutton switches S1 through S4. Install the switches, one at a time, in the marked locations. Make certain each switch is fully seated and that the metal base of each switch is straight and flush with the board. It's important to get these lined up right so the pushbuttons will fit properly in the holes of the cabinet end panel.



Install the speed control potentiometer in the location marked SPEED. Like the pushbutton switches, this part must be inserted **fully** into the PCB so that the bottom of the metal bracket sits firmly flush against the PCB. Even a slight mis-alignment here can cause fit problems with the cabinet later, so double check your work. After you solder it in place, turn the adjusting shaft fully counter-clockwise.

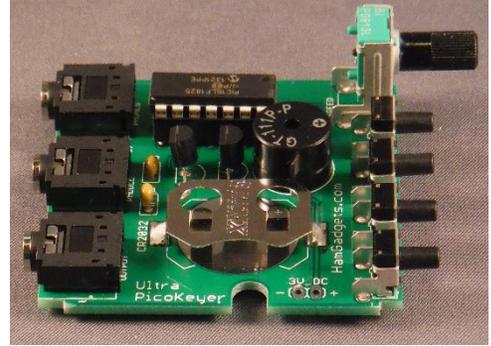


Install the three stereo jacks in the locations indicated on the component side of the PCB. Be sure to get them fully seated and flush against the board. There are plastic pins on the bottom that will fit into holes in the PCB when they are pushed all the way in. Again, it's important to make sure these are properly mounted and straight. If they're not lined up right, you will have trouble installing the board into the plastic cabinet.

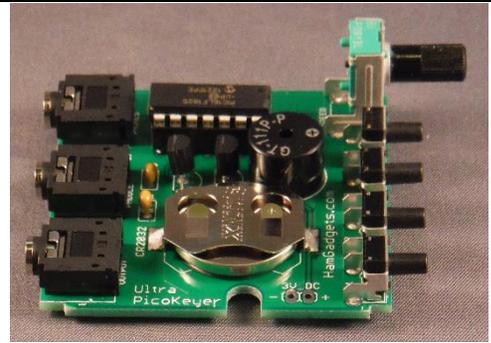


C3 is used to limit the amount of audio supplied to the headphone jack. Install one of the remaining capacitors in location C3. You may want to leave this unsoldered until you have had a chance to test it with your headphones or earphones. The **.047  $\mu$ F capacitor** (marked 473) gives good results with many earphones. If you need more audio, try the **0.1  $\mu$ F capacitor** (marked 104). If it's too loud even with the .047  $\mu$ F capacitor, you may need to try a smaller value such as 0.01  $\mu$ F (marked 103). **Be careful** while testing this! **Too much audio can damage your hearing. If you intend to use earphones or headphones, try it with the earphones or headphones OUT of your ears first.**

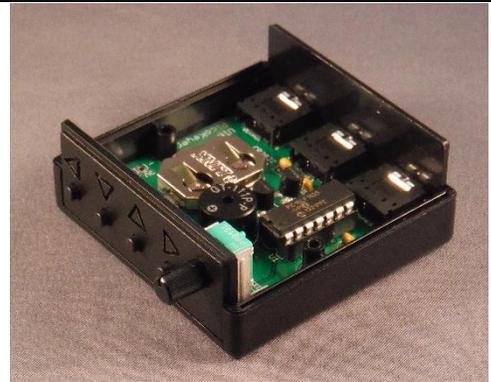
We're almost there! Find the PicoKeyer chip and remove it from its protective anti-static foam. Locate the Pin 1 end – this end will be marked with a molded notch and/or a dot. If you hold the chip so you can read the markings on top, Pin 1 is toward your left. Orientation is important here! Insert the chip into the socket so that the notch and/or dot on the chip are on the same end as the notch in the socket and the notch outline on the PCB – the end closest to the SPEED potentiometer.



Now locate the battery and remove it from its protective packaging. Note that the flat side is marked with a plus sign (+). This side will be **up** when the battery is inserted into the battery holder. Slide the battery into the battery holder. If you have done everything right, you should hear the keyer send "73" in Morse code through the speaker. Congratulations! Your kit is complete.



Now is a good time to mount your keyer in the cabinet. The board is supported in the cabinet by the jacks and speed control shaft. You may need to trim the pins sticking through the bottom of the board to get things to fit right. Take your time and pay attention to getting everything lined up properly before closing up the box. You may need to trim the cut-off component leads a little closer to the board to get everything to fit right.



# Support Information

## Warranty & Support

Your Ultra PicoKeyer Kit is guaranteed against defects for one year from date of purchase. This warranty does not cover damage due to improper modification, improper soldering or wiring, overvoltage, static damage or other misuse or abuse. That said, if you have problems please contact me via email to arrange for an exchange or replacement part. If you accidentally damage your keyer, don't panic! Replacement parts are not expensive. Send an email and let me know what you need.

Should you need support, have questions, have feature requests or bug/problem reports, please feel free to contact me via email at [n0xas@HamGadgets.com](mailto:n0xas@HamGadgets.com). I will make every effort to respond as quickly as possible.

## Troubleshooting

Having problems during or after assembly of your kit? Don't worry... it's fixable! Here are some common problems and what to do about them. More hints can be found at <http://www.hamgadgets.com>.

**Q:** *Everything is done, but I get no "73" when I install the battery!*

**A:** Almost all of these so far have turned out to be soldering mistakes. Remove the battery and check the voltage – it should be just above 3 volts. Make sure you had the battery in right side up (+ marking on top). Now carefully go over the solder joints with a magnifying glass. Re-melt any that look suspicious, and use de-soldering braid or a solder sucker to clean up any blobs you have left. Next, check the resistance across the speaker leads. A good speaker will show somewhere around 50 ohms.

**Q:** *I broke (or melted) a part! (Or, I got a bad part!)*

**A:** Don't panic. Email me, or just mail the bad part back to me with a note. Include your address! If you broke it, it would be nice to include a couple of bucks to cover the postage and packaging. If it was bad when you got it, just let me know and I'll send a replacement at no charge, including postage. I'm pretty easy to get along with.

**Q:** *The sidetone audio is too low, I can't hear it!*

**A:** You can adjust the sidetone audio frequency (menu setting "A"); it will get quite a bit louder near the speaker's resonant frequency around 2 kHz. If that doesn't do enough for you, you may want to use headphones or a small audio amplifier and larger speaker.

**Q:** *I'm having problems entering messages into memory.*

**A:** The keyer will insert a word space if it sees more than about 1 "dit" time between characters. I allowed a little play, but not a lot. I recommend that when storing the message, don't even try to worry about character spacing. Let the keyer do it. Enter the characters as you normally would, but when you finish one character just start the next immediately. You only need an instant after the last dit or dah when you're not touching the paddles.

Many people find it helpful to slow the keyer down when entering messages - if you normally work at 12 WPM, slow it down to 10 or even 8, but enter the characters with almost no space between them. Let the keyer do the character spacing. Just pause a second or two between words and let the keyer do the word spacing -- just like you can't get the character spacing too close, you can't get the word spacing too large.

**Q:** *I can't enter a message into memory while using a bug or straight key.*

**A:** Correct. You must be in iambic (A or B) or Ultimatic mode, and you need to use a paddle of some sort. Single or dual lever is OK, but it must have separate dot and dash contacts.

*Q: When I plug in a straight key, I just get dashes! How do I make it work with a straight key?*

A: Straight key auto-detection is done during the power-on sequence, or when you push buttons 2 & 3 at the same time. So... push buttons 2 & 3 (▼ & ▲) at the same time.

*Q: When I plug the keyer into my radio, I get constant dashes (or dots) and weird results when I try to use the paddle!*

A: Make sure your rig's internal keyer is turned off, and the cable from the keyer to the rig is wired for operation with a straight key. Consult your radio's owner manual for the correct wiring for this.

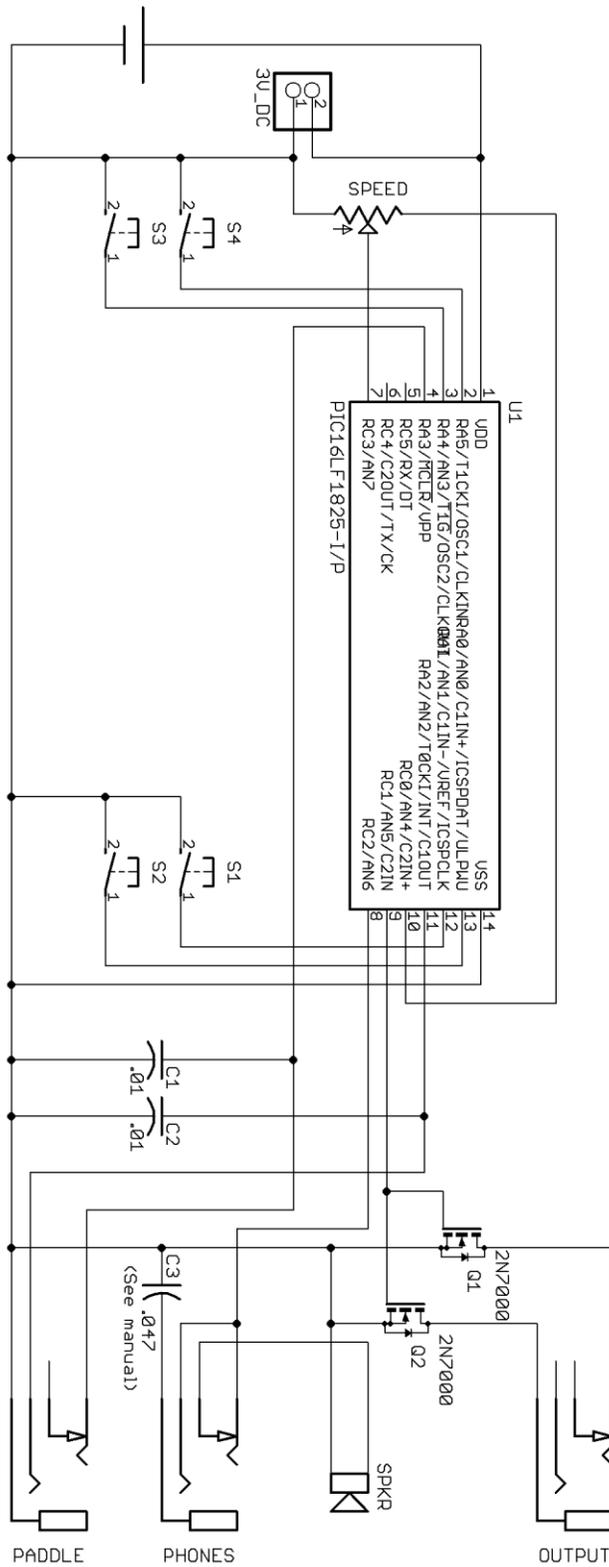
*Q: My radio keys up as soon as I touch the paddle, and stays keyed until a second or two after I stop sending.*

A: You have probably selected "MCW" mode. In this mode the keying output is used for PTT and audio is fed to an FM transmitter for on-air code practice with FM radios. Check the sidetone or "T" menu setting.

*Q: When I try to use the keyer, it doesn't key the radio. Then after a while I hear "R" or "R K".*

A: You probably have a stuck button. The keyer is in "Record" mode, and the "R" means you have filled that memory slot. Check to see if you need to adjust the fit of your keyer in the cabinet. It's usually either a button or the seed pot not installed correctly, or leaving the cut-off leads too long on the bottom of the board.

# Schematic Diagram



## Firmware version history

Vers.	Date	Remarks
1.0	2014/09/23	Initial release
1.1	2014/10/23	Fixed message record with sidetone OFF Fixed letter spacing (L) = 1 Changed multi-press button scan timing to make it more reliable Fixed auto letter spacing and multi-button interaction
1.2	2014/11/09	Fixed sidetone during speed announcement, QSO decrement and tune mode
1.3	2014/11/28	Improved paddle response during message transmission Fixed message save issue that was truncating messages at 64 bytes, followed by a compiler bug that was corrupting longer messages.
1.4	2015/01/09	Disabled BOR for power reduction, and to fix problems with chips resetting when buttons or paddles are pressed. This happens only with certain chips (silicon rev?)
1.5	2015/01/20	Fixed message record backspace (8 dits) to leave the space after the previous word Fixed use of multiple /Snn commands in a message
1.6	2015/03/04	Fixed beacon delays > 25 seconds
2.0	2015/05/15	Added MYCALL setting. MYCALL consists of two message slots, each of which can be any string up to 63 characters. Added QRSS mode, with dot lengths from 1 to 255 seconds. Added support for separate PTT output on RC5 (pin 5) "Relaxed" the paddle timing when recording messages, to allow a little more time before determining end-of-character.
2.1	2015/11/23	Fixed short menu mode for MENU and MYCALL. Fixed paddle/straight key detect in all cases. Improved memory stacking button response. Messages can be stacked up to 4 deep. Pressing the same memory button while a message is playing will terminate the message. Added ~ 32 second timeout to setup menu.
2.2	2017/01/27	Paddle re-scan using middle buttons will now announce "P" for PADDLE or "K" for KEY. Added BANK 2 with four more memory slots, and BANK/BK menu.