NØXAS   Super PicoKeyer

Low Power Iambic Keyer

The PicoKeyer is a single chip, automatic iambic Morse code memory keyer. Its small size, low power requirements and robust set of features make it perfect for portable or QRP operation or for integrating into transmitters or transceivers.

Features of the Super PicoKeyer Kit:

- Dual MOSFET keying circuit will key transmitters up to 60 V, positive or negative
- Simple one-button "menu" interface
- Four 60-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
- Setup and message entry using your paddle
- Auto straight key detect, all message memories available with straight key
- "Bug" mode allows automatic dots with manually formed dashes
- Speed adjustable from 5 to 60 WPM via menu OR optional speed potentiometer
- Dual-Set Speed allows quick QRS/QRQ and return to favorite speed
- Adjustable weight
- Variable pitch audio sidetone
- Curtis "A" or "B" or Ultimatic modes
- Variable letter/word spacing (Farnsworth)

- Tune mode with on/off carrier or automatic string if dits for easy tuneup
- Beacon mode with adjustable 0 – 99 second repeat delay and optional power-on auto start
- MCW mode for sending audio Morse over voice radios
- Memory "pause" command 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
- Variable transmitter QSK delay compensation
- Memory and parameter settings retained with power off
- Low voltage - from as low as 2.5 to 5.5V
- External DC power supply up to 24V can be with optional on-board voltage regulator
- Low current - typical sleep current .005 µA, under 1 mA when keying.
- Keyer kit PCB has pads for all connections (paddles, audio, power & keyline) for easy integration into other equipment

Assembling the PicoKeyer Kit

Your 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 fine 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.
- Fine electronic solder. Use a good quality, small diameter rosin core solder intended for electronic assembly. DO NOT use acid core solder!
- Small needle-nose pliers and a pair of small diagonal 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.
- If you want to use a speed control potentiometer, you will need a 10K to 50K Ohm, linear taper pot. Mouser PN 311-1902-50K or 313-1210-50K are good choices. You can order a 50K linear taper pot from HamGadgets.com.

Once you have all of your parts, 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 resistors 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:

- 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.
- 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 a 45-degree angle to hold the part in place.
- 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 lead and the hole at the same time. Wait a couple of seconds for the lead to heat up, then touch the solder to the resistor lead, NOT the soldering iron tip. The solder should flow into the joint. Remove the solder and iron and don't move the PCB until the solder has cooled. The joint should be 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.
- 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!
- 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.

Step-By-Step Assembly Instructions

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

2. ☐ Find resistor R1, a 10K Ohm (Brn-Blk-Orn), 1/4 Watt resistor. Insert the resistor in the location marked R1 but *do not solder it yet*. The resistor is not polarity sensitive, meaning don't worry – you can't install it backwards!

3. ☐ Find the .1 µF capacitor. This will be small rectangular part with two parallel leads, marked 104. The capacitors used in your kit are not polarity sensitive. Install the .1 µF capacitor in the location marked C1. Save one of the cut-off leads for later use as a jumper wire if you do not plan to use a speed control potentiometer.

4. ☐ Find the two .01 µF capacitors. These look similar to C1, but are marked 103. Install the two .01 µF capacitors in the locations marked C2 and C3.

5. ☐ Install one of the 2N7000 MOSFET transistors in location Q1. Make sure the flat side of the transistor is facing the direction indicated by the silkscreen printed outline. Leave about 1/8" between the circuit board and the bottom of the transistor to avoid putting too much stress on the leads.

6. ☐ Install the second 2N7000 MOSFET in location Q2, just like you did the first one.

7. ☐ Now find the 8-pin IC socket. Orient the socket so that the notched end is at the end indicated by the silkscreen markings. Insert the pins into the PCB. 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 eight pins in place. Be careful not to use too much heat or too much solder.

8. ☐ Find pushbutton switch S1. You may need to straighten the leads with a pair of needle-nose pliers before inserting them into the PCB. Install the switch in the location marked S1.

9. ☐ *If you are building your PicoKeyer with a voltage regulator for external power, skip to the next step*. If you are building your PicoKeyer with the on-board battery, find and install the battery holder. You may need to use a piece of adhesive tape to hold it in place while you solder. If you have a black battery holder, you will need to leave about 1/8" of space between R1 and the board and bend its leads to move R1 out of the way of the battery holder. Solder the leads for R1 after the battery holder is in place.

10. ☐ *If you wish to use an external power source*, install a 78L05 voltage regulator and 0.33 µF capacitor (not supplied) in the locations marked VR1 and C4. **Power should be supplied at JP1-1 and JP1-2** (7 to 20 Volts DC). Remember to solder the leads for resistor R1 if you have not already done so.

11. ☐ Install the two stereo jacks in the locations indicated on the component side of the PCB. Be sure to press them far enough in to get them fully seated. There are three plastic "bumps" on the bottom that will fit into holes in the PCB when they are pushed all the way in. Install the two threaded, knurled collars on the two stereo jacks. This keeps them from getting lost! You can use them to mount your PicoKeyer in a box or tin.

12. ☐ Find the speaker and install it in the location marked SPKR. The speaker may be 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. Don't spend too much time soldering the speaker or it may be damaged.

13. ☐ If you wish to use an analog speed control potentiometer, connect it to points H, L and C on the circuit board as shown on the schematic. The center pin of the potentiometer connects to point C, the high speed end to point H, and the low speed end to point L. **Important:** *If you do not wish to use a potentiometer, solder a wire jumper between points C and L.*

14. ☐ We're almost there! Find the PicoKeyer chip and remove it from its protective anti-static package (foam or tube). Locate the Pin 1 end – this end will be marked with a molded notch and/or a dot (the dot may be painted). 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.

15. ☐ If you built your PicoKeyer for battery power, 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. Insert 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.

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/kit-hints.html.

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. 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.

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 buck or so to cover the postage. If it was bad when you got it, just say so. I'll send a replacement. I'm pretty easy to get along with.

Q: The sidetone audio is too low, I can't hear it!
A: Try covering the small hole in the top of the speaker with a piece of tape. You can also adjust the sidetone audio frequency if you have difficulty hearing high or low frequencies. If that doesn't do enough for you, you may want to use a small audio amplifier and larger speaker.

Q: I'm having problems entering messages into memory.
A: Try setting the speed a few WPM slower than you normally send while you enter the message. More detail is at http://www.hamgadgets.com/kit-hints.html.

Q: I can't enter a message into memory while using a bug or straight key, or when in bug mode.
A: Correct. You must be in iambic mode (A or B), 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: I removed and replaced the battery, and now the keyer doesn't seem to work right.
A: The bypass capacitors on the board can store enough energy to power the 12F683 chip for several seconds. Remove the battery, then press the setup button to discharge the filter caps. Now re-install the battery and you should be OK.

Note: Leaving a straight key plugged in will consume roughly 150 to 250 uA additional current, which can drain the on-board lithium coin cell in a few months. *If you use a straight key, unplug the key when not in use. Leaving a straight key plugged in can reduce battery life.*

Operating the PicoKeyer:

When power is applied, the keyer sends "73" via the sidetone signal. This lets you know the keyer is awake and operating normally, and all settings have been loaded from nonvolatile memory. At this point the keyer will operate as a normal iambic keyer, sending dots and dashes with automatic spacing and timing according to paddle inputs and stored settings. If you have a speed control potentiometer attached, you can vary the speed from 5 to 60 WPM by simply turning the pot.

One momentary pushbutton switch input is used for memory messages and setup. Pressing and immediately releasing the button once (a "tap" of the button) will send the message stored in Memory 1 (M1), if there is one stored.

Pressing and holding in the pushbutton more than about half a second will allow you to send one of the other three memory messages. If you wish to send the contents of message memory 2 (M2), release the button when you hear two dits, sent via the sidetone only. Similarly, to send message 3 or 4 (M3 or M4), release the button when you hear three or four dits. To enter setup mode, continue to hold the button down after you hear the four dits for M4.

This is the method used to access all features other than M1. In general, the way to access message memories M2 through M4 or any setup menu item is the same. Simply hold the setup button down until you hear the keyer send the character corresponding to the item you want, then immediately release the button.

In setup mode the keyer will use the sidetone only and not key the transmitter. As you hold in the button the keyer will cycle through the available menu choices, described later in this document. You can simply release the button when you hear the item you want to check or change. Once you finish with that menu item, tap the button and listen for "SK" to exit the menu, or hold the button down to advance to the next item.

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 non-volatile memory, and is used when the speed pot is turned to its minimum position, OR if the speed potentiometer is not installed. At any time you may use the speed setting pot (if installed) to set any speed from 5 to 63 WPM. Setting the speed with the pot does not affect the stored speed, which will always be available by simply turning the pot to its minimum setting. 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. You can omit the speed pot and only use the stored speed, changing speed via the paddles and menu. Finally, you can simply ignore the stored speed and always use the pot for speed control – it's up to you.

Master Reset:

You can return your PicoKeyer to its default settings by holding both paddles while turning power on. The keyer will send "73"; immediately release the paddles. At this point the keyer will send a question mark (?); tap either paddle to perform a complete reset. Note that you may need to try this a few times, depending on how quickly you let go of the paddles while the keyer "wakes up".

Straight Key Mode:

The PicoKeyer can automatically detect and use a properly wired straight key if one is plugged in when power is applied. During its power-on program, the PicoKeyer checks to see if either paddle input is grounded. If one is, 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 menu will still function, many parameters will not be adjustable (and would not apply to a straight key anyway). You will not be able to record messages while using a straight key.

Powering the PicoKeyer:

Power can be supplied by the on-board CR2032 lithium battery or an external DC power source. Any voltage from 2.0 to 5.5 V should work fine without needing to install the regulator VR1. If you are keying your rig with a MOSFET such as the 2N7000 supplied with the kit, keep the supply voltage at 3 V or higher to get enough gate voltage to turn the MOSFET on. For normal use as a standalone keyer, a single CR2032 lithium coin cell will last a long time. If you plan to build your PicoKeyer into a rig, don't worry about battery backup - all memory and parameter settings are retained in non-volatile EEPROM memory and will be automatically recalled when power is applied.

If you wish to use an external power source the PicoKeyer kit may be built without the battery holder, instead installing a 78L05 voltage regulator and .33 uF ceramic capacitor in the locations marked VR1 and C4.

Sidetone:

The PicoKeyer's sidetone is a square wave audio signal. Sidetone can be turned on or off and the audio frequency can be changed from the setup menu. If you are building the PicoKeyer into an existing rig with sidetone, you can simply couple the sidetone signal from JP1-3 to your rig's audio stage through a capacitor. Using a passive low-pass filter will give you a much cleaner signal that will be easier on the ears during extended use.

Keying the transmitter:

The PicoKeyer will key any solid state, tube or hybrid transmitter or transceivers that uses a keying voltage of 60V or less, positive or negative. If you are using the keyer with a negative keyed transmitter, you will need to make sure the KEY and RIG jacks are isolated from each other – not mounted to a common metal box or panel. The easiest way is to use a plastic enclosure, or insulate the RIG jack from the panel or box.

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 either more robust output transistors such as VN2410L (available from Mouser) or a separate keying adapter. The Universal Keying Adapter 2 available from NØXAS at www.hamgadgets.com is optically isolated and will handle solid-state, grid-block or cathode keyed transmitters at up to 400 V.

If you are integrating a PicoKeyer into a transmitter or transceiver, the keying signal is available on JP1-5. This is a positive-going logic signal; do NOT use this to directly key a transmitter! Use it to drive an external keying circuit such as a 2N7000, optoisolator, or a common NPN such as a 2N2222A with a series resistor sufficient to limit base current to less than 20 mA. Avoid any voltage on JP1-5 that exceeds the supply voltage Vdd, and limit current to less than 20 mA.

Support:

Should you need support, have questions, have feature requests or bug/problem reports, please feel free to contact me via email at **n0xas@botkin.org** or **n0xas@arrl.net**. I will make every effort to respond as quickly as possible.

Warranty:

All parts in your kit are tested and are guaranteed against defects for 90 days from date of purchase. This warranty does not cover damage due to incorrect assembly, improper soldering or wiring, overvoltage, static damage or other misuse or abuse. If you have problems, please contact me via email to arrange for an exchange or replacement part. If you accidentally damage a part, don't panic – just contact me, replacements are not expensive. Unfortunately I cannot offer assembly or repair service, but you should be able to find someone local to help out in the event of problems with assembly.

Keyer 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.

PicoKeyer Setup Menu Commands

U	**Tune** mode: Tune mode is used to send either a steady carrier or a series of dits for adjusting or testing your station equipment. While in tune mode, each paddle acts as an on/off toggle switch. Tap the dash paddle to turn a steady carrier on or off. Tap the dot paddle to start or stop a continuous stream of dits. This is gives you a 50% duty cycle signal that is preferred by some operators as a way to tune up with less stress on final PA, tuner and antenna components.
S	**Speed**: The keyer will always announce the *current speed* in WPM. If the *current* speed is not the same as the *stored* speed, the keyer will announce the current speed followed by a slash and the stored speed. For example: Assume the stored speed is 13 WPM, any you have the optional speed pot installed and set for 20 WPM. The keyer will announce "20/13". If you have the pot set to its minimum (or if you have the jumper installed on the PCB and are not using a speed pot), you will be using the stored speed and the keyer will only announce the stored speed, since it is the same as the current speed. It sounds more complicated than it is; play with it a little and you'll get the hang of it. You can adjust the *stored speed* from the menu. Tapping the dot paddle will decrease the speed by one WPM, or the dash paddle will increase it. Holding either paddle will continuously increase or decrease the speed, with a dot or dash sent at the new speed for each step. When the paddle is released, the keyer will again announce the current speed setting and the stored speed, if it is different from the current speed. Speed may be set from 5 to 60 WPM.
M	**Messages**: Four message memories are available, numbered 1 through 4. When you enter Message mode, message 1 will be the default selection. You have a choice of actions available to you when in memory mode, selected by sending a single character from your paddle: • Send the number 1, 2, 3 or 4 to select a message memory. The keyer will respond by sending *1*, *2*, *3* or *4* to confirm. • Send **P** (**P**lay) to listen to the contents of the currently selected memory. The keyer will play the message, followed by the Morse prosign *AR* and the message number. • Send **R** (**R**ecord) to record a new message. If a message already exists it will be erased and replaced. The keyer will respond with *K* to let you know it is in record mode. Enter your message, with exaggerated word space but normal spacing between characters. If you make a mistake when recording the message, just send 8 dits and the keyer will backspace one word. You will hear a single dit to confirm this (two dits means you are at the beginning of the message). Tap the setup button once when you are finished recording. The keyer will send *R* and the message number to indicate the end of the message. You can then **P**lay the message back. • Send **C** (**C**ontinue) to add to or edit the message. The keyer will play the current message, then enter Record mode. You can backspace over existing words if needed. You can use the **R, P** and **C** commands to listen and change your message until you're satisfied. Each memory can hold up to 60 characters. If you send something other than 1 - 4, P, R, or C the keyer will respond with "**?**" and let you try it again. 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 //.** • **/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/R**DE (callsign) **/R/R**K". This can save a lot of memory space.

- **/1**, **/2**, **/3** and **/4** can be used to chain the message memories. The indicated message will be played immediately when one of these commands is encountered. You can chain messages in any order.
- **/P** will insert a pause in the message. This will cause the message to pause while you manually send information such as manually entered QSO number, RST, etc. Tapping the setup button while paused will resume the message. *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.
- 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 below) and re-send the message. You can terminate beacon operation by tapping either paddle or the 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 PicoKeyer, store the **/A** command as the first two characters in message #1. This is useful for autostarting a keyer used as part of a beacon. 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 to change speed by more than one WPM – for example, **/SU/SU** will increase your speed by 2 WPM.
- To decrease keyer speed by one WPM, send **/SD** (Speed Down).
- To alter the letter spacing, send **/F***n* 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 **/F2**N0XAS **/R/R** K **/F0**/B". Remember to set the letter spacing back to your normal setting.
- To insert an extra word space in your message, use the special prosign character "*1M*" (.------).
- To insert a steady carrier, send more than 9 dashes. The length of the carrier will be equal to the number of dashes, with no spaces in between.

Q	**QSO Number:** The keyer will send the current QSO number. You can use the paddles to set the QSO number anywhere from 1 to 255. Note that when the QSO number is auto-incrementing, it can go from 1 to 65535.
R	**Reset QSO number:** Keyer sends "?" and waits. Tapping either paddle will reset the QSO number to 1.
L	**Auto letter spacing:** The keyer will announce the current setting and wait for input. Automatic letter spacing takes effect for messages sent from memory as well as code sent manually with the paddle. • 0 turns automatic letter spacing off. In this mode you control the spacing between letters. This is the default setting, and is the same as most keyers. • 1 turns on auto letter spacing at the set speed. 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. • Settings from 2 to 9 will set automatic letter spacing with longer delays. For example, selecting 2 will insert one extra "dit" length spacing between characters. Selecting 3 will insert an extra 2 "dit" lengths, and so on. • Note that auto letter spacing is inactive while in setup mode.
X	**Cut numbers:** The keyer will send **Ø** and **9** using the current setting and wait. You can cycle between no cut numbers (**Ø 9**) cut zeros (**T 9**), cut nines (**Ø N**), or both (**T N**). Note that this *only* 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.

Z	**Leading Zeros:** The keyer will send "Y" or "N" and wait. As with other menu settings, either paddle may be used to switch between the two settings. **Y** will 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** sends numbers without leading zeros. Like the X setting, this affects only QSO numbers and not other numbers stored in messages or sent manually. The default setting is N.
B	**Beacon delay:** The keyer sends the current beacon delay in seconds. Use the paddles to increase or decrease the delay between beacon transmissions from 0 to 99 seconds.
W	**Weight**: The keyer announces the current weight and waits for input. The dot paddle may be used to decrease the weight or the dash paddle to increase it. Weight can be set anywhere from 1 (50% "light") to 5 (normal) to 9 (50% "heavy"). The default setting is 5.
T	**Sidetone**: The keyer announces the current sidetone setting and waits for input. You can use the dot and dash paddles 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.
K	**Key Mode:** 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 key or paddle 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.
P	**Paddle Selection**: 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!
A	**Audio Tone:** The keyer will send a dash at the selected sidetone audio frequency each time a paddle is hit. Use the paddles to increase or decrease the audio frequency as desired. The default setting is approximately 700 Hz.
D	**Transmit delay compensation:** 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. The default setting is zero.
V	**Firmware version:** Keyer sends the version number of its program.
F	**Factory Reset:** Keyer sends "?" and waits. If you tap either paddle the keyer will be completely reset to its original settings All message memories are deleted and the QSO number is reset to 1. You can also perform a factory reset at power-on. If both paddle inputs are grounded when power is first applied, the keyer will enter the "F" menu function and send "?" in Morse code. Tap either paddle to perform the reset, or press the button to exit without resetting.
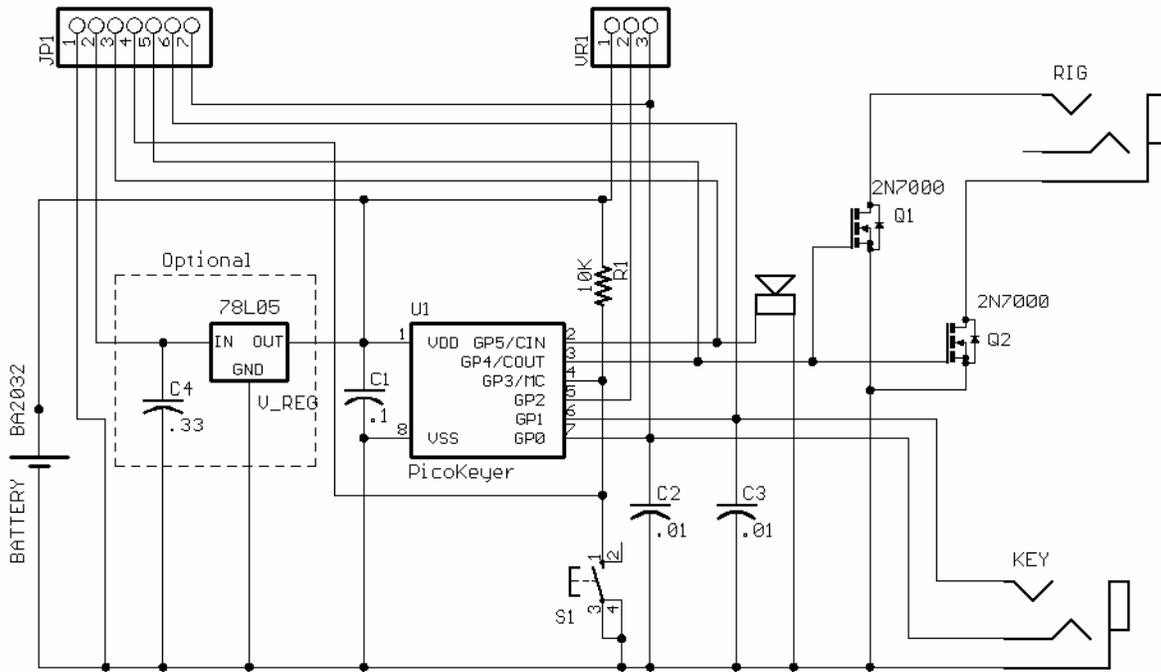
PicoKeyer Kit Schematic

Connector pads JP1

Connector JP1 is found in the same location as the battery holder. The pinout can be found in Table 2 below. You can use the pads of this header to more easily integrate your PicoKeyer into your own equipment. Note that the keying output on JP1-5 is a positive-going logic signal – **do NOT** use this to directly key a transmitter! Either use the RIG output jack pads or add your own external keying circuit.

PicoKeyer Menu Quick Reference			
2 dits (I)	Send M**2**	Z	Leading **Z**eros on/of
3 dits (S)	Send M**3**	B	**B**eacon delay
4 dits (H)	Send M**4**	W	**W**eight
U	t**U**ne mode	T	sidetone
S	**S**peed	K	**K**eyer Mode (**A**, **B**, **U**matic, bu**G**, **S**traight)
M	**M**emory	P	**P**addle reverse
Q	**Q**SO# Set	A	**A**udio tone
R	**R**eset QSO#	D	Xmit precomp. **D**elay
L	**L**etter spacing 0-9	V	Firmware **V**ersion
X	Cut numbers 0 / 9	F	**F**actory Reset

Table 1

Pin	Function
Pin 1	GROUND
Pin 2	+V to regulator input
Pin 3	Sidetone audio out
Pin 4	Pushbutton In
Pin 5	Key signal out (logic)
Pin 6	DOT paddle input
Pin 7	DASH paddle input

Table 2 (JP1 Pins)