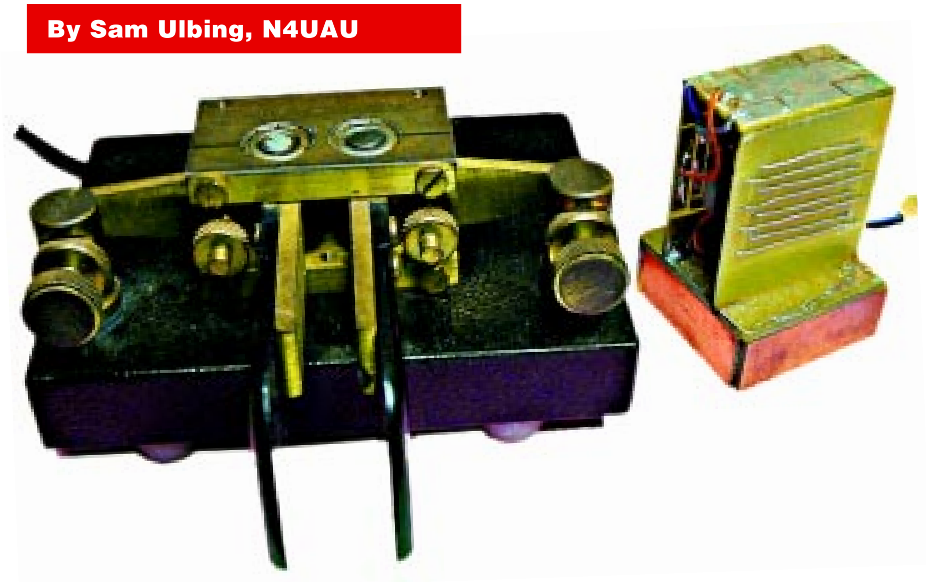
By Sam Ulbing, N4UAU



Uncle Albert's Touch Pad Keyer

Are conventional "compact" keyers still too big? Want to build a small, durable keyer with no moving parts that runs on a single lithium coin battery? Build the touch pad paddles and use them with your present keyer, or build the tiny keyer circuit to accompany your favorite paddles. This project is an experimenter's delight!

Above: Uncle Albert's Touch Pad Keyer, with touch pads, battery and circuit, can be made as small as the finger pads of a standard paddle!

One of the fun things about experimenting with ham radio electronics is that the final result isn't always what you had in mind at the start. That's definitely the case with this project. I wanted to build the world's smallest "surface-mount keyer" because I was impressed with the performance of my "World's Smallest Code Practice Oscillator."[1] But very early in the design I realized that the size of a keyer isn't limited by the electronic parts, but by the size of the power supply, paddles and the user's fingers. I could easily build a keyer on a PC board that's less than one inch square, but if I needed four AA batteries and I used my regular paddles, I would still have a large keyer (see Figure 1). Instead of looking for smaller ICs, I wound up looking for low-voltage/low-power ICs and a way to reduce the size of truly functional paddles.

Touch pads solved the paddle dilemma and, thanks to modern MOSFETs, they are easy to build. As you can see from the title photo, my touch pad keyer (paddles, batteries and circuitry) is barely larger than the finger pads on the paddles. Some of my other versions are even smaller!

I wanted to use a state-of-the-art 8-pin CMOS microprocessor (such as a PIC chip) because of its small size, minimal current drain and low-voltage functionality. Although I have programmed microprocessors for two previous keyer projects, I didn't have the equipment, nor did I want to spend the time to learn the PIC programming language. Fortunately, Steve Elliott, K1EL, had a nice PIC-based keyer IC and, when I asked him if he would be willing to cook up a surface-mount version for my "record attempt," he enthusiastically agreed. You don't have to use this preprogrammed PIC; there are other choices available that can be

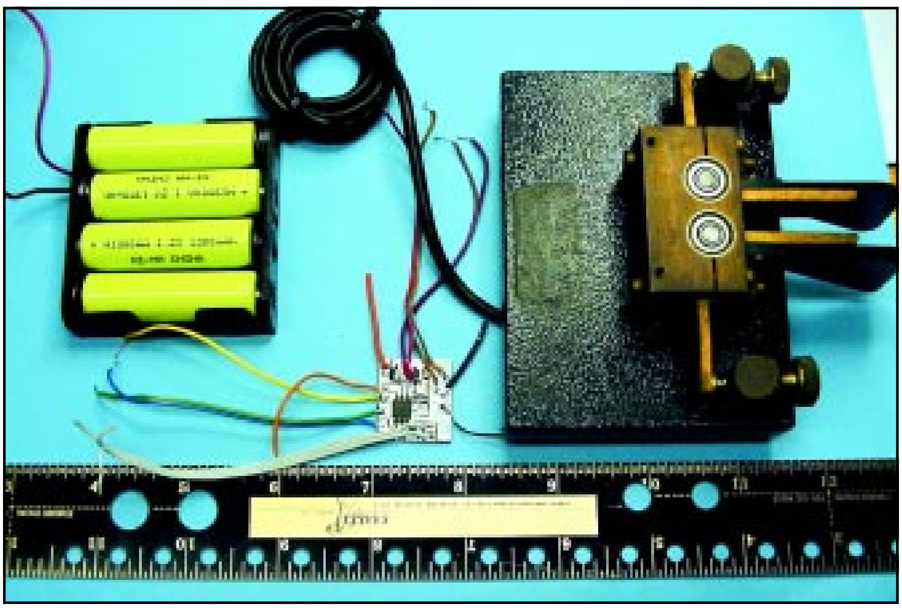


Figure 1—A small circuit with large paddles and batteries is *not* a small keyer!

adapted to this keyer. See the sidebar, "Selecting a Keyer IC."

The Circuit

Figure 2 shows the schematic for my keyer—the K10 keyer circuit by K1EL, adapted for this project. Changes include using the low-current version (PIC12LC672) rather than the standard version (PIC12C672) of the microprocessor, the addition of a bulk capacitor and a voltage-based MOSFET that replaces the current-based output transistor. The low-current microprocessor operates at a lower voltage and uses half of the current of the standard version. The bulk capacitor absorbs current pulses, extending the life of the high-impedance battery. It is important that this capacitor be a low-leakage type because it is always connected across the battery. Using an output MOSFET instead of a transistor reduced current demand by 30%. In addition, Steve changed the code to allow a quicker transition to sleep mode to further reduce power consumption, and he adjusted the sidetone frequencies to include some higher pitches to match the frequency response curve of the Piezo transducer.

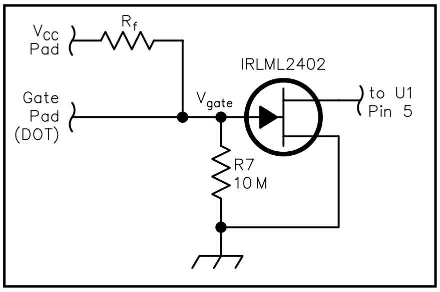


Figure 3—Finger resistance equivalent circuit.

Pins 5 and 6 are the dot/dash inputs. They are activated when these pins go *low*. Although pin 6 has an internal pull-up to keep it normally high, pin 5 needs an external pull-up, R1. Pin 7 is the input for an A/D converter that is built into the IC. This converter allows programming flexibility because it can measure the input voltage and perform actions based on that voltage. Steve chose to use it for three additional message memories (for a total of 4) and as a driver for the speed-control potentiometer. My keyer design doesn't use the speed pot because of its size and there are other ways to vary speed. The voltage at pin 7 is set by the voltage divider comprised by R2 and R4, R5 or R6. The pin voltage is normally held at V_{cc} by R2, but when one of the memory pads is pushed, the voltage divider causes the voltage to drop, which is detected by the microprocessor.

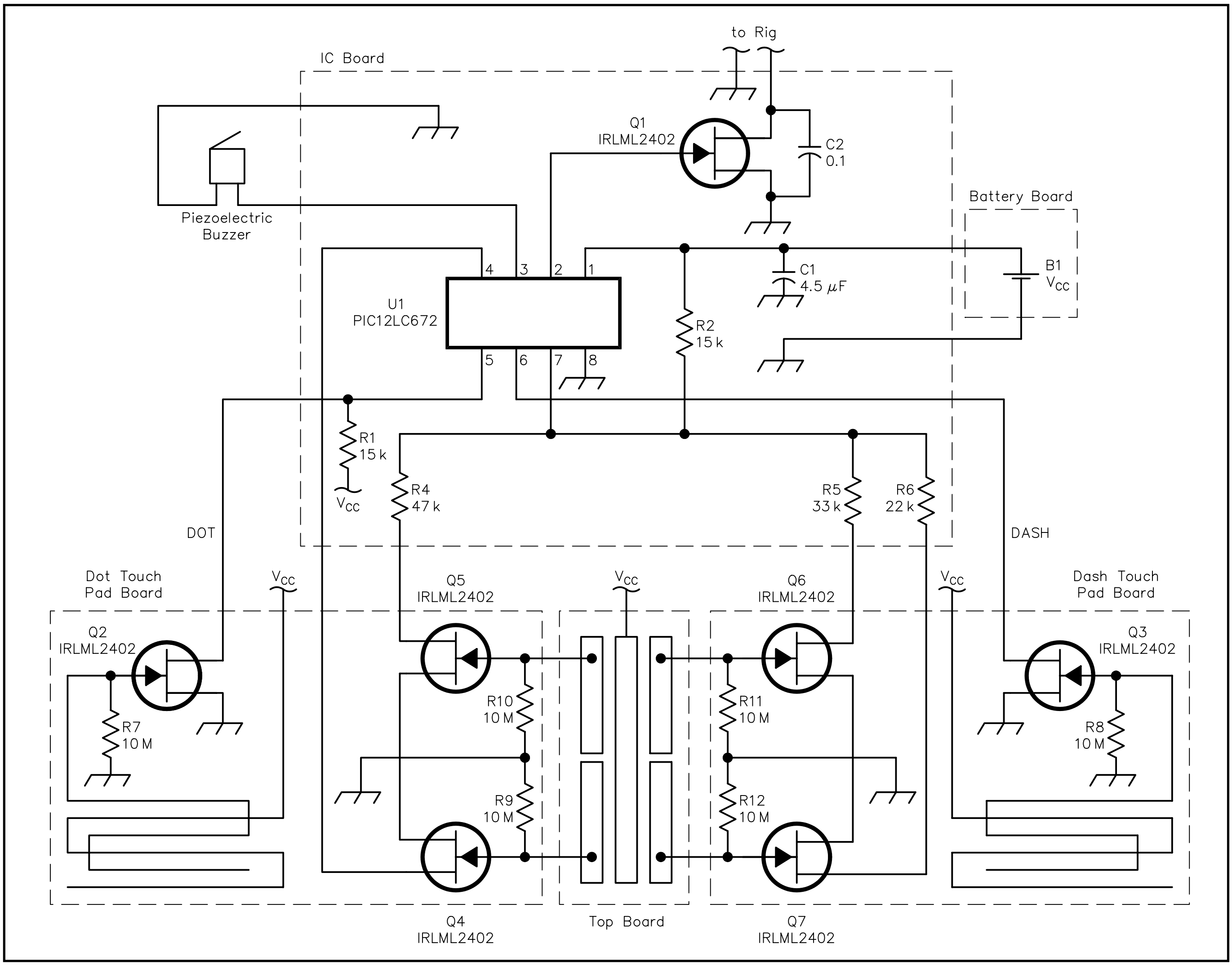


Figure 2—The keyer schematic showing the location of the parts on the various boards. DigiKey part numbers are shown in parentheses. Contact DigiKey at 800-344-4539, or on the Web at www.digikey.com. R3 is not shown; it is not needed when Q1 is a MOSFET.

B1—Lithium coin-cell battery.
C1—4.7 µF ceramic (PCC1842CT).
C2—0.1 µF ceramic (311-1142-1-ND).
Q1-Q7—IRLML2402 (RLML2402CT-ND).
R1, R2—15 kΩ (P15KACT).
R4—47 kΩ (P47KACT).
R5—33 kΩ (P33KACT).
R6—22 kΩ (P22KACT).
R7-12—10 MΩ (311-10MECT).
U1—PIC12LC672 (preprogrammed from K1EL. See text.).
Piezoelectric buzzer, low voltage.

Selecting a Keyer IC

Having built two keyers from scratch, I know that developing code for the microprocessors is very time consuming and often frustrating. I determined early that I would only attempt this project if I could find a "commercial" CW keyer IC to use.

One that I considered was my very own Remote Keyer described in the July/August 2000 *QEX*. I did not use it because the IC could not meet the requirements I had set out for the project. It was too large at 20 pins and it needed 5 V.

I selected the K10 keyer chip by K1EL because it was PIC based, was small and because Steve, K1EL, was willing to modify it to meet the voltage and power requirements I wanted. I had first built this project using one of his original keyers with two lithium batteries and found that it worked well and used relatively little power. The article discusses the changes Steve made to optimize his chip for this project.

Another possible choice would be the TiCK-series chips available from Embedded Research at **www.frontiernet.net/~embres/tick-emb.htm**. You'll need to adapt my circuit for the TiCK pin configuration according to the information provided from Embedded Research.

The granddaddy of keyer chips was the Curtis 8044ABM, but this IC is no longer manufactured. Even so, they still pop up at flea markets and elsewhere. Some Amateur Radio parts vendors may even have a few tucked away. If you find one, see the ARRL Technical Information Service page on the Web at **www.arrl.org/files/infoserv/tech/curtis.txt** for pin-out information. A current replacement is the Curtis 8045ABM, available from MFJ (**www.mfjenterprises.com**).

Pin 4 is the "Command" input and action is initiated when this pin is taken low. Pin 4 actually performs a number of functions, including memory storage and speed changes, as well as command inputs. (See the sidebar, "Main Features of the Keyer IC," for command functions.) Pin 3 outputs a tone to a piezoelectric buzzer and is normally used to signal the IC's response to input commands but can also be used as a sidetone if you want to use the training features of the chip. Q1 is the output MOSFET that keys a solid-state rig.[2]

Q2 and Q3, which perform key roles, are high-impedance MOSFET buffers for the touch pad inputs. IC1's dot/dash pins have impedances around 20 kΩ, which is much too low to permit direct touch keying because of the relatively high resistance of human skin. Q2 and Q3 are held low (normally off) by 10 MΩ resistors R7 and R8. Pressing the dot touch pad inserts the finger resistance between V_{cc} and the gate of Q2, which provides a voltage determined by Equation 1 (Figure 3 shows the equivalent circuit). Q2 and Q3 are those "neat MOSFETs" I discovered when doing my series on surface-mount devices.[3] They conduct when a voltage greater than 1.6 V is applied to the gate. This means that a finger resistance, R_f, of less than 8 MΩ will turn on the MOSFET and trigger the keyer. Most people's finger resistance is less than 2 MΩ[4] so only a light touch is needed.

$$V_{gate} = V_{cc}(R7/(R7+R_f)) \qquad \text{Eq 1}$$

The Power Supply

The keyer operates from 2.5 to 5.5 V and typically draws less than a milliamp when running and 2 microamps when sleeping. Because of these low power demands, the keyer runs on a single lithium coin battery.[5] I chose to use a size 2032 battery, which is relatively large but has a 200 mAh capacity. If the keyer is used for hour-long QSOs each day (transmitting half of the time), the battery should last for more than a year.

The Touch Pads

Touch pads offer several advantages over paddles. They're smaller, much more robust and have no moving parts. In addition, they're easy to home-brew (no machine shop required) and much less expensive. You can drop this keyer on the floor and it will still work. I don't recommend doing this routinely, but if you do that just once with a set of paddles you'll almost certainly be shopping for new ones.

The design was influenced by several factors. First, the pads had to make good electrical contact with the fingers when lightly touched. A light touch is critical to keying smoothly and swiftly. Second, because of the high impedances involved, the PC board resistance between the V_{cc} pad and the MOSFET gate pad had to be very large so it wouldn't cause the keyer to trigger because of dirt, humidity, etc. I made the touch pads on single-sided PC boards, drilled a series of holes in the board and ran #24 solid solder-coated wire to form a grid. One wire goes to V_{cc} and the other goes to the gate of the buffer MOSFET (see Figure 4). When my finger contacts the wires, the resistance between them is reduced.

Two design features help to ensure a low resistance. First, the wires are above

Main Features of the Keyer IC

- Speeds from 5 to 59 WPM.
- 95 stored characters can be divided among four memories in any configuration.
- Keying modes include straight key and Iambic A and B.
- Variable element weighting in 32 steps.
- Paddle-swapping command for multi-user convenience.
- Adjustable sidetone.
- Two code practice modes.
- Low power consumption—no on/off switch required.
- Small size.

Changing Keyer Conditions

- Press and hold the Command Pad until the keyer responds with an "R," then input a command.
- Press and hold the Command Pad, touch a dot or dash paddle before the keyer sends "R" to "fast change" the speed up or down in increments of 2 WPM.
- Press a memory pad after keyer responds with "R" to "fast load" that memory.
- Press and quickly release the Command Pad to send the contents of memory 1.
- Press and quickly release M2, M3 or M4 to send the contents of that memory.
- If you enter an illegal command the keyer responds with "?."

Keyer Command List

This partial list of the commands gives an idea of the capabilities of the keyer. A full data sheet is available at the author's Web site at **n4uautoo.home.sprynet.com**, or the K1EL Web site at **www.k1el.com**.

A—Turn sidetone on/off.
C—Set speed of command input and output.
K—Toggle straight key/Iambic keyer functions.
L—Load memory while transmitting.
M—Load memory off line.
P—Start receive code practice mode.
Q—Query current speed settings.
R—Review a message without transmitting.
S—Set sending speed from 5 to 59 WPM.
T—Sends a constant tone for tuning.
U—Sets auto spacing between letters (very useful when storing messages).
V—Start transmit code practice mode.
W—Change element weighting.
Z—Change sidetone frequency.

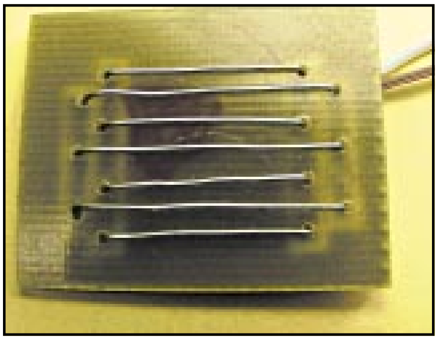


Figure 4—A touch pad as seen from the touch side.

the surface of the PC board so they tend to press into the skin more than if I had used traces etched into the copper clad. Second, the fingers make contact with the wires in several places at the same time, causing a parallel resistance condition that reduces the overall resistance. This design also minimizes false triggering because of surface conductivity. The wires make poor contact with the PC surface and the wires are small but the spacing is large. (Other touch pad designs are described in the sidebar, "Experimenting with the Keyer," if you want to experiment.)

Experimentation showed that finger orientation was important to me, and that my fingers don't always make contact at exactly the same spot every time I touch the pad. The first pads I built were on horizontal pieces of PC board because they were easy to build and sturdy (but sent poorly). Clearly, my "finger-brain coordination" preferred vertical paddles spaced about 3/4-inch apart and positioned about 1/2-inch above the table. This suggested the configuration shown in Figure 5; the IC board and piezo are between the vertical touch pads and the base section contains the battery.

An interesting design requirement arose in the process of testing the touch pads. They worked fine most of the time, but would occasionally stop sending. Because this happened when I was operating off line, I knew it wasn't RFI. Investigation showed that the problem was caused by body resistance and capacitance when my non-sending hand was touching the base section, which was connected to ground via solder joins. To eliminate this problem, I isolated the base from ground by making cuts in the foil on the top of the battery board where the touch pads connect. (See the sidebar, "Effects of the Body for the Intellectually Curious," for an analysis of this effect.)

Layout and Construction

Construction of the keyer is novel because the entire assembly is made from PC board material. There are no boxes, screws or fasteners. Instead, all of the boards are soldered together. Soldering PC boards together might seem like a new approach, but I first learned about it a number of years ago when reading an article by Doug DeMaw[6] on making RFI-tight enclosures. I made all my prototypes using a hacksaw, my Dremel method[7] of cutting traces in the foil and the Dremel drill accessory to drill the holes. A kit of parts is available.[8]

The keyer is made from seven pieces of PC board (see Figure 6). The layout and parts-mounting details I used are available as JPG files on *ARRLWeb* (**www.arrl.org/files/qst-binaries/layout.zip**). The two boards for the touch pads are single sided and the rest are double sided. There are three sections: the IC circuit board, the touch pad section and the base section. The base section

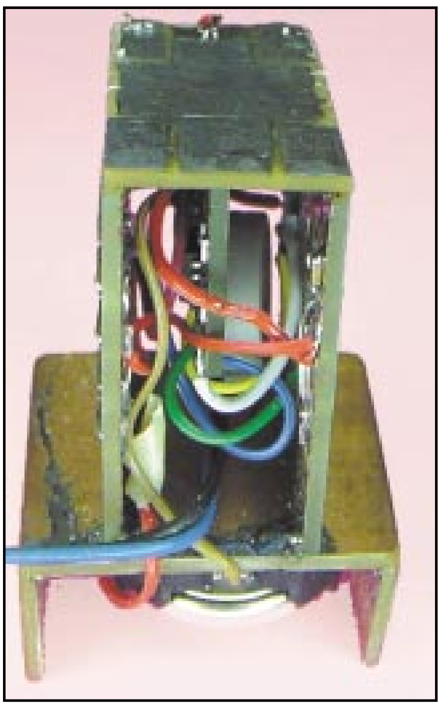


Figure 5—Shown here is the U-shaped touch pad section mounted on the top of the base section. All parts are in place. Note that the IC board and piezo buzzer are located between the touch pad boards.

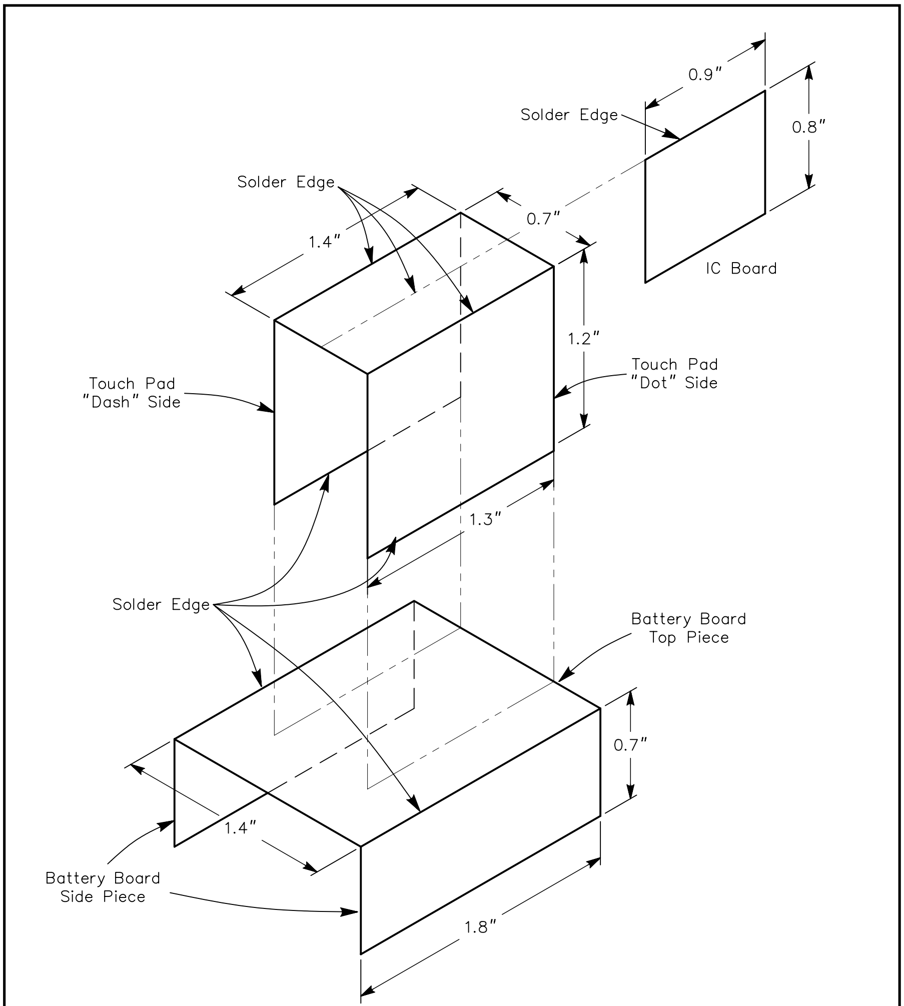


Figure 6—Assembling the seven boards to form the keyers.

Experimenting with the Keyer

Standalone Paddles

The keyer's touch pads will work with any modern keyer. A friend, ND7K, built a version using two AA batteries to power the MOSFETs (and add weight to the paddles). The batteries ought to last a lifetime as the MOSFETs draw no power when off and less than a microamp when the paddles are touched.

Cutting Touch Pads with a Dremel Tool

I built a set of touch pads by cutting them into the foil with a Dremel tool. The pads worked fine but exhibited occasional problems. At first I thought the problems were RFI related, but further experimentation showed that the problem was probably related to moisture or dirt between the fingers of the pads. Increasing the space reduced the problems substantially. This is an area for more investigation.

A Low-Impedance Touch Pad Keyer

To key the microprocessor directly from its dot and dash inputs, the resistance between the inputs and ground must be 5 kΩ or less. A bare finger will not key the circuit. I was puzzled by how to make the touch pad work directly with my finger when my wife, KD4DZX, found the solution. She wanted me to wrap some food in aluminum foil. Aluminum foil has a very low resistance and I found that I could wrap it around my finger and, when I touched the pads, the keyer worked! I had good tactile feedback because the foil was so thin and flexible. Further, because of the low resistance involved, I could simply use my Dremel tool to cut the touch pads and the memory pushbuttons directly into the copper foil.

The low-impedance version affords a more compact layout. If you hold this keyer in your left hand, bare fingers will not key the keyer so it doesn't matter if they contact the touch pads as it does with the high-impedance version. Additionally, body effects will be nil because of the low resistance involved.

If you don't mind a raised eyebrow or two, there are other ways to reduce finger resistance; here are a few I tried (not recommended for group Field Day settings): Aluminum duct tape (used for heating vents) cut into small pieces and stuck to my fingers worked well; also I tried using guitar picks to hold the tape but found them to be too rigid. A wire wrapped around my finger didn't work very well because it tended to be below the skin level. A bandage made conductive with aluminum tape, or a latex glove coated with shellac and graphite powder seemed to offer some promise. There are many more possibilities to try. Figure A shows a low-impedance keyer in action with "aluminum foil resistance reducers."

Making the Keyer even Smaller

It would be easy to mount parts on both sides of the PC board to greatly reduce the size of the IC board. If you substitute a surface-mount LED and resistor for the Piezo, you can make a keyer circuit that's about the size of a postage stamp. Combining that with a smaller battery and bare wires for the touch pads could produce an extremely small keyer. (I experimented using a surface-mount LED in place of the Piezo and found it easy to recognize the microprocessor's responses when the Command speed was set to 15 WPM or less.)

Using the Keyer with Standard Paddles

Because of its small size and power demands, the keyer is worthwhile even if you don't want to use the touch pads. If you simply want to use the keyer circuit, it's possible to mount all of the parts except the memory buttons underneath a standard paddle. See Figure B.

RFI Thoughts

RFI hasn't been a problem for me, but if your variant has RFI trouble, you could add a small capacitor between the MOSFET gates and ground. Keep in mind that the capacitor will need to be around 1000 pF or less because of the long time constant involved with the 10 MΩ resistor.

Ideas for Field Day Use

It should be possible to power the keyer from a small solar panel—a great alternate power use for Field Day. The touch pads can probably be used for a long time when powered by a charged capacitor (if you use one with low leakage). Would a small loop antenna, a diode and a storage capacitor extract enough power from the Field Day ether to provide the few milliwatts needed to run the keyer?

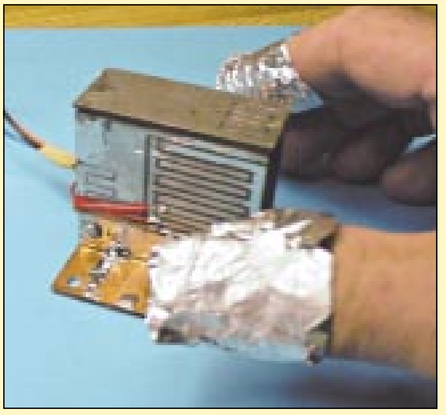


Figure A—Low-impedance keyer with aluminum foil "impedance reducers."

Figure B—The keyer mounted on a Kent Paddle. Build the circuit on a U-shaped set of boards to fit your paddle. The circuit board and battery board will fit under the paddle and only the pushbuttons and piezo buzzer will be visible on top.

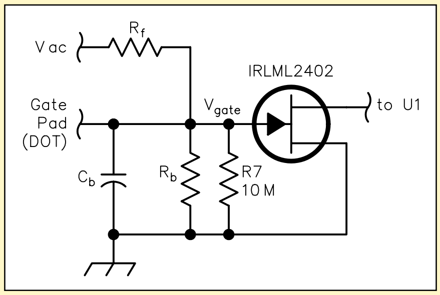
holds the battery and provides a stable base. Because the keyer is small and lightweight, you must hold it firmly in your left hand so it doesn't move as you touch the paddles. I used two pieces of PC board for the sides of the base plate, but you could use standoffs if you prefer.

I built the electronic circuit on a separate board (IC circuit board) instead of combining it with one of the touch pad boards. This allows for a flexible layout (helpful when experimenting) and offers some insurance for home-brewing the PC board. If I made an error cutting the board I would only have to redo it and not the touch pads as well. I mounted all the parts on the same side of the board because it's easier to integrate the circuit board with the rest of the keyer. Building the IC board requires straightforward surface-mount construction techniques and is fairly easy to do with my Dremel method.[9]

The touch pad section consists of three boards. The sense wires are run through holes in the single-sided boards and soldered in place on the foil side. Exact placement of the wires isn't critical, but it's important that they lie flat on the board, as any high spots will tend to make your keying less precise. The high-impedance buffer parts are mounted on the foil side. These boards get soldered to the top board, which has the command and

Effects of the Body for the Intellectually Curious

By using a simplified circuit, it's possible to analyze the effects of the body on the touch pads. If you don't touch ground with your non-sending hand, the key circuit looks like that shown in Figure 3; Equation 1 gives the voltage at the MOSFET gate. If you touch ground, your body resistance is inserted between the MOSFET gate and ground, in parallel with the 10 MΩ resistance. In addition, the body acts as a capacitor. This forms an equivalent circuit as shown in the drawing.



Body effect equivalent circuit (simplified).

You can easily see and calculate the resistance and capacitance effects of your body with a DVM and a stopwatch. Connect the DVM between ground and the MOSFET gate pad. Lightly touch your finger to the pads. If you're not also touching ground, the voltage on the gate should jump instantly to around 3 V. If you are touching ground you will see the observed voltage slowly increase to a value less than 3 V. Press harder and the voltage rises more quickly and to a higher value than if you touch it lightly. As I said, this project has room for a lot of experimenting!

Not touching ground when keying:
- Battery voltage 3.25 V
- Gate voltage when finger touches pads = 3.0 V
- R7=10 MΩ

From Eq 1: R_f (finger resistance) = 833 kΩ

Touching ground when keying:
- Battery voltage = 3.25 V
- Gate voltage (maximum after several seconds) = 1.90 V
- Time for voltage to start keyer working (at 1.66 V) = 2 seconds
- R_f (as determined from above) = 833 kΩ
- R_b (body resistance) = 1170 kΩ (use Eq 1 and ignore effect of the 10 MΩ resistor because R_b is much less)
- The keyer turns on at 1.66 V, which is 87% of the final voltage of 1.9 V.
- Eighty-seven percent is approximately two time constants (from the *ARRL Handbook*). Because a time constant is defined as: $T = R_a C_a$, we calculate C_b (body capacitance) = 5 µF

This means is that the voltage at the gate will be lower when touching ground because of the low body resistance. The body capacitance delays the MOSFET "turn on time"—so the circuit works, but not fast enough to send code. R_f, R_b and C_b all depend on the pressure applied to the pads. The effect is more noticeable for a light touch and for lower values of V_{cc}.

Figure 7—Use a poor ham's 90° jig when soldering the boards.

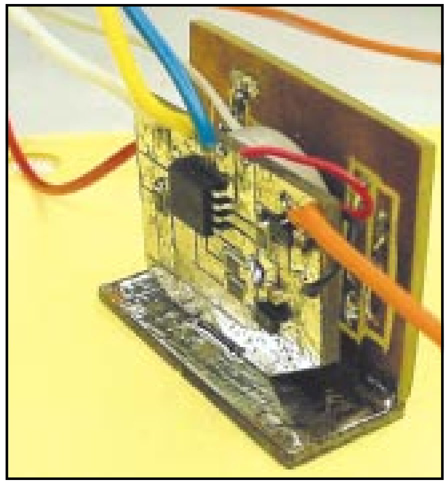


Figure 8—Soldering the IC board to the top board between the touch pad boards.

memory push pads, to form a U shaped, three-sided box. The push pads work in the same fashion as the touch pads, but because a light touch isn't important for them, they are simply rectangular pads cut in the foil. It's important that the keyer sit squarely on the table, which means that the parts need to be cut and soldered squarely. To help square things up I placed mine next to a board while soldering (see Figure 7).

The battery section is made from three pieces of double-sided board—a top and two sides. The battery holder is on the underside of the base top. I mounted the side pieces just slightly in from the edge of the board and soldered them to the battery board on both sides with quite a bit of solder for strength. Be sure to orient the battery holder so the battery can slide in and out of the open side! If you drill a small hole in the battery board behind the battery, you can route the wires through it, which will make the layout neater and provide strain relief for the wires to the rig.

To solder the board edges together, clean the PC board and lightly solder coat the edges.[10] The idea is to get enough solder to tack the two boards together, but not so much that they don't sit flush. After tinning, position the board, touch the iron to the joint and you should get a tack point. Do this at both edges and then go back and add a bit more solder to strengthen the joints. If you make a mistake and need to unsolder a pair of boards, a single-edge razor blade works well to separate the parts. Work it between the parts as you heat the solder and hold it there until the solder cools. I strongly recommend using just a little solder at the corners of the boards to hold them in place until you are certain that the keyer is working correctly.

Soldering the touch boards and the IC board together has to be done in the proper order. Some wires need to be connected to each board before assembling the boards because of the tight clearances involved (see Figure 8). I suggest you do a "dry run" before you actually solder the parts together so you can see which wires will be impossible to reach once the parts are assembled. Handle this section gently until it's soldered to the base.

Solder the battery holder to the underside of the base top and attach the sides. Solder the touch pad subassembly to the base (you only need to tack solder the four corners) and connect the power wires to the battery.

Operating the Keyer

The keyer operates in a fashion similar to most modern keyers. (Some of its main features are shown in the sidebar, “Main Features of the Keyer IC.”[11]) Because of its low current drain, no ON/OFF switch is needed. When you stop sending, the keyer goes into a sleep mode that draws only 2 µA. Touch a touch pad or push pad and the keyer instantly wakes up. Because power is always supplied to the keyer, it retains all of its settings and memories. You simply have to sit down and start sending. To change the keyer’s operating conditions, use the Command push pad as described in the sidebar.

You can use the keyer as a straight key by issuing the “K” command and then turning the keyer on its side so you can tap the dash pad, which acts as the straight key input. To me, the feel is much like using a straight key.

The keyer allows a total of 95 Morse code characters to be stored in its four memories. The partitioning is flexible, so each memory “message” can be from 0 to 95 characters, as long as the total isn’t greater than 95. Thus, if you choose to use only two memories you still have a total of 95 characters to work with.

In addition, the keyer offers two practice modes to help you with your Morse code proficiency. In receive practice mode the keyer will send five-letter character groups for you to copy. In transmit practice mode it will send a letter, pause for you to send the letter yourself, and indicate whether you sent it correctly.[12]

If you already have a keyer that you enjoy, you can build just the touch pads and use them to operate it. I hope you try this project and make several versions of it—like I did! Experimenting with new technology is fun and educational! Isn’t that what ham radio is all about? I would love to see the innovative designs you come up with.

Notes

[1]“World’s Smallest Code Practice Oscillator,” *QST*, Feb 2001.

[2]Modifications necessary to use the circuit for tube based rigs are shown in the K1EL data sheet for the K10 at his Web site (**www.k1el.com**).

[3]I like to use surface-mount parts because they incorporate state-of-the-art technology. These simple MOSFETs are a good example. Since I found them and used them in Part 4 of my surface-mount project (see footnote 7, I have used them in two other projects because of their unique characteristics! Why are they so “neat”? Most MOSFETs need 5 V or more to turn on, so they wouldn’t work in this project. It’s also possible to use a comparator for this job, but comparators draw current and will cause the battery to drain much more quickly. The MOSFETs draw essentially *no* current.

[4]You can check this out by holding the two probes of an ohmmeter between your fingers and reading the resistance. Note that because of its high input impedance, the meter may take a while to settle on a final reading. If you have dry skin the reading may be quite high. Using a little hand cream will usually substantially reduce skin resistance.

[5]You can also build a touch pad keyer using the standard 12C672 IC that K1EL sells, but you will need to make sure the voltage stays above 3 V and expect your batteries to run down faster.

[6]“Experimenting for the Beginner,” *QST*, Sep 1981, p 11.

[7]“Surface Mount Technology: You Can Work With It,” *QST*, Apr-May-June-July 1999.

[8]A kit of parts is available from the author. For the keyer section, it includes the keyer board (IC Board #1) and all the parts in the parts list for that board plus the piezo, small gauge hook up wire and fine solder. For the touch pad section it includes a single etched and drilled PC board containing the boards and side pieces. In addition it includes solid tinned wire for the touch pads, four MOSFETs (Q2, Q3, Q4, Q5), four 10-MΩ pull-down resistors and a lithium 20-mm size battery holder. The board is supplied as a single board about 1.2 inches wide and 6 inches long and requires a hacksaw to separate the pieces. The cuts are marked and only involve cutting the width of the board. Use of a mitre box insures a square cut. See photos on my Web page. Price for the kit is $30. If you wish to build just the touchpads, the cost is $16. Prices include postage in the US and Canada. Shipping overseas add $1.50. Florida residents please add 7% sales tax. Payments should be made by a US or international money order, a check payable by a US bank, or Western Union money transfer. Credit cards are not accepted. Send to Sam Ulbing, 5200 NW 43rd St, Ste 102-177, Gainesville, FL 32606.

[9]I recommend first doing some projects with bigger parts if you have never worked with surface-mount parts before.

[10]If you are working with a non-solder-coated board, lightly sand and flux the edges so the solder will adhere.

[11]See Steve’s Web page or mine for a full data sheet.

[12]Because the buzzer audio output is rather low, an audio amplifier like my *SMALL* audio amplifier (Jun 1996 *QST*, p 40) could be a useful addition especially if you are using the learning feature with a group of hams. Another way to increase the volume of the piezo is to make a Hemholz resonator for it as described in Part 4 of my surface mount article (see Note 7).

Sam Ulbing, N4UAU, holds a BEE and an MBA from Cornell University. Soon after he became a ham in 1988, Sam started contributing project articles to QST, QEX and 73 Amateur Radio Today, many of them focusing on Morse code, his favorite mode of communication. Since his first project using surface-mount parts in June of 1996, he has become fascinated with using these parts for Amateur Radio projects. This project is his first cooperative effort with another ham. He hopes to have the opportunity to do similar articles in the future because he believes that the benefits of combining the skills and knowledge of several project builders will make it possible to develop more sophisticated Amateur Radio projects. You can contact Sam at 5200 NW 43rd St, Suite 102-177, Gainesville, FL 32606; **n4uau@arrl.net**; **n4uautoo.home.sprynet.com**.

NEW PRODUCTS

SYMETRIX DEBUTS AUDIO PROCESSING WEB SITE FOR HAMS

◊ In response to a growing interest in communications audio processing, Symetrix, a veteran manufacturer of audio processors for the broadcast market, has created an area on its web site dedicated to helping hams improve their sound and increase their overall modulation. The site focuses on Symetrix voice processors (such as the Model 528E, shown here), which are well-suited for use in RF environments. Reached from **www.symetrixaudio.com**, the ham radio section features product descriptions, operator testimonials, photos, and links to broadcast dealers that will extend their services to the Amateur Radio market.

DSP BLASTER 2.5 RELEASED

◊ *DSP Blaster* provides real-time DSP filtering and spectral analysis using your PC and sound card. The program uses digital signal processing to provide highpass, lowpass, bandpass, peaking, automatic-notch, noise-reduction, and coherent CW filters, as well as automatic gain control. *DSP Blaster* uses 100% assembly language to implement optimized, floating-point algorithms. The program requires a 486 PC or better, math coprocessor, VGA, mouse, 16-bit non-PCI Creative Labs sound card, and *DOS* 3.0 or later. *DSP Blaster* will run as a *DOS* application under *Windows*. To order, send $75, check or money order, to Brian Beezley, K6STI, 3532 Linda Vista Dr, San Marcos, CA 92069.