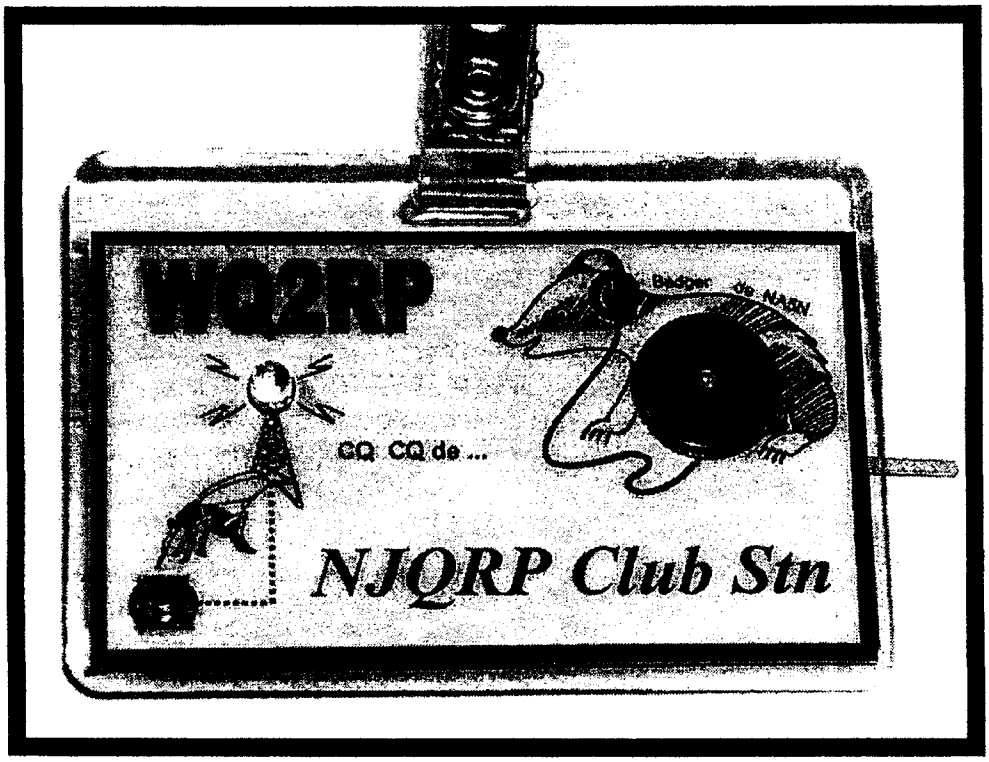
QRP
New Jersey

"The Badger"

A Morse-Identifying SmartBadge Kit



Here's a unique callsign "smartbadge" kit that you can build, wear and have fun with at ham radio events. We've designed a programmable pc board containing a PIC-like microcontroller, a piezo annunciator, an LED, a microswitch and a battery that all fit neatly under a custom laminated plastic callsign badge. At the touch of a button the Badger sounds the owner's callsign using Morse code through a piezo device and simultaneously blinks the code using a SuperBright LED. The smartbadge also has a beacon mode that repeatedly sounds the callsign, and an iambic keyer mode that can be used to key a transmitter. An expansion port provides for some interesting add-on accessories.

Brought to you by the New Jersey QRP Club

"The Badger"

A Morse-Identifying SmartBadge Kit

Thanks for purchasing the Badger SmartBadge kit. It won't take long to assemble the Badger and you'll have years of fun wearing and using this functional badge.

INTRODUCTION

The Badger is a unique and programmable callsign badge that sounds and blinks the owner's pre-programmed callsign in Morse code when triggered.

It consists of a piezo sounding device and a SuperBrite LED on the front of the pc board, and a small pushbutton that is used to start the beeping and blinking of the owner's callsign.

A Ubicom SX-18 microcontroller and several other components also reside on the board which is overlayed with a custom-designed laminated callsign badge.

A 3V lithium battery is provided on the pc board, allowing the smartbadge to be operated free of any encumbering heavier battery.

Add a common badge clip and the Badger can be worn at club meetings and other ham radio events to the amazement of everyone!

FEATURES

- **Badge size** - pcb is 2" x 3.5", laminated overlay is 2.5" x 3.5";
- **Piezo** - approximately 4 kHz, adjustable for peaking to individual piezo device;
- **LED** - a red SuperBrite LED
- **Pushbutton** - micro-size, initiates annunciating sequence;
- **Annunciation** - piezo and LED are modulated using Morse code at given speed to announce callsign and other features;
- **Key/Paddle** - pads on pcb are provided for builder to add a miniature, on-board homebrew key or paddle, and signals provided off-board via 3-position pin header connection;
- **Keyline output** - buffered output line delivers Morse to external transmitter via 2-position pin header connection (user-supplied);
- **Expansion connector** - provides access to special functions, accessible by 7-position pin header (user supplied);
- **Programming connector** - 4-position pin header connector allows user to connect the SX-Key or SX-Blitz programming tool for custom programming of the SX microcontroller chip;
- **Operating modes** - annunciate callsign via piezo and/or LED, repetitive "beacon mode", iambic keyer, keyer speed adjust, piezo tone adjust and straight key tone oscillator;
- **3V lithium battery** - standard watch battery provides circuit power. SX chip "sleeps" during quiet times, providing long battery lifetime.

CIRCUIT DESCRIPTION

Refer to the schematic shown in **Figure 7** for the following discussion.

The SX-18 microcontroller contains the software program that provides all the features of this project. Clocking for this chip is provided by a built-in, on-chip RC oscillator. Timing accuracy of the tone frequency and the Morse code speed is not overly critical so we can live with slight chip-to-chip variations in clocking rates when using the built-in oscillator mode - this is a nice price to pay

for reduced parts count!

Power is normally provided by an onboard 3V lithium battery when Jumper J1 is in place as shown, or by an external 5V source when J1 is in the pin 2-pin 3 position on pin header P3. External 5V is required only when a new program is being burned into the SX chip. This "in-circuit programming" capability of the SX device makes it very easy to re-use in projects.

When the Badger smartbadge is quiet, the SX controller is "asleep". In this state, the SX clock oscillator has stopped and the chip only draws several microamps of current from the 3V battery.

However when the smartbadge is triggered by pushbutton PB-1 being actuated, the SX chip wakes up and begins executing its internal software program.

Turning the smartbadge on by means of a quick tap is reminiscent of the activation scheme for the communicator worn on the shirt in the old Star Trek series ... "Beam me up Scottie!".

Once triggered, the program in the SX chip wiggles the output bit RB1 at about a 4 kHz rate, causing piezo device X1 to emit a reasonably loud tone. The software program modulates the tone with Morse code at about 15 wpm rate to sound the owner's callsign.

The RB2 output pin is also turned on and off at the same Morse code rate and buffered by transistor Q1 to drive the KEY input of a transmitter. In this way the Badger can key a transmitter, which will be seen as a more useful feature when some other Badger features are described in a moment.

LED D1 is also turned on and off according to the Morse code being sounded, giving visual indication of the callsign being announced. The blinking SuperBrite LED is very useful in helping to one copy the audio Morse code. The specs for the piezo state that the sound pressure generated (i.e., audio output level) is at least 80 dB, which is pretty easy to hear in a typical environment. However when the room is filled with talking people and other noise, the beeping can easily be drowned out. Thus the LED blinking in time with the sounding of the piezo greatly helps by correlating visual and audible inputs to the person trying to copy the callsign.

Paddle inputs are available for connecting standard iambic paddles, enabling the SX chip to perform as a keyer. When the smart badge is placed into "keyer mode", grounding of the dit or dah input lines (SX ports RB4 or RB5, respectively) activates the piezo and LED for the corresponding dit or dah time periods. Another related mode, Straight Key mode, instructs the SX program to recognize grounding of the dah input pin as a straight key closure, which in turn sounds the piezo and turns on the LED.

The Badger pc board was layed out to allow the builder to construct a miniature homebrew paddle right on the back of the smart badge. As shown sticking out on the left of Figure 2, a thin strip of scrap pcb material can be soldered in between two grounding stubs such that when the strip is moved from side-to-side by the operator's hand, the Badger sounds off just like a code practice oscillator. Additionally, when pin header P5 connected to a transmitter, the smart badge acts as a full-fledged iambic keyer. Captain Kirk never had it this good!

The pc board was designed to use a surface mount version of the SX-18 microcontroller. This package was selected over a DIP package because of its lower cost, lower profile and lighter weight. Even the pc board was fabricated using .031" material, providing for a lighter weight badge hanging on one's shirt pocket.

Because a surface mounted SX chip is provided (pre-soldered on the pc board in the kit), we needed to provide the means to initially program the chip after it was soldered in place, and for reprogramming the chip when a different or improved program is ultimately available later on. The 4-position pin header P1 is provided in order to allow the SX-Key or the SX-Blitz programmer to connect to the board and feed new software to the SX chip. This connector and in-circuit programming function will not normally be used by most people - only when a new program is desired for the chip.

SOFTWARE DESCRIPTION

In order to follow along with this software part of the discussion, you could refer to the Badger assembly source code listing provided at the Badger website: www.njqrp.org/badger/. You may also write to request a copy of the source code.

The software design is based around an interrupt that happens every 148us when the RTCC counter (which counts clock cycles) rolls around past zero. This is a common way for PICs and SX chips to generate a constant stream of interrupts and is determined by a setting of the RTCC reload value at the end of each interrupt.

Thus operations performed during the interrupt service routine (ISR) happen at that 148us periodic rate. As it turns out, generating the frequency for the piezo sounding device is the highest priority for the program and we toggle the bit connected to the piezo every time we pass through the ISR. Since a "cycle" of wiggling this bit takes two passes through the ISR - bit set on one pass, bit reset on the next - the basic default frequency of piezo operation is 2 x 148us = 296us, which sounds the piezo at 3.37 kHz.

If we programmatically modify the RTCC reload value, the piezo tone will vary around that default frequency and the operator could "calibrate" the piezo output for its natural peak resonance. That is actually the operation done in the Calibrate mode of the software. Once in that mode, when the operator taps the pushbutton two times, the software reduces the RTCC reload value a bit and the resultant ISR rate (and the piezo frequency) is increased. When the operator taps the pushbutton once, the RTCC value is increased, which lengthens the interrupt rate and lowers the piezo tone.

The rest of the Badger's program is really quite elementary, as done by many of the PIC keyer chips/programs available today. We first determine the ASCII character of each letter in the pre-programmed callsign, use a look-up table (LUT) to turn it into an equivalent pattern of 1's and 0's that represent the character's dits and dahs, and then control the length of the piezo "beeps" and the spaces in between those beeps based on the dit and dah sequence.

A default 10ms element of time is used as the Morse unit - a dah is 3 units with the piezo tone on, a dit is one unit, a character space is one unit with the tone off, and a word space is 3 units. Put all these unit tones and no-tones together according to the Morse-equivalent of the ASCII character and you have Morse code beeping and blinking at you at about 12 wpm!

Timing of the Morse output is easy to adjust, via the "Morse Speed" mode settings. When this mode is entered, the basic default unit length is increased or decreased slightly, effecting a corresponding decrease or increase of the Morse output speed.

Once the desired tone frequency and Morse speed selections are made, the software saves the settings in its onboard RAM memory. These settings are retained even during the sleep modes of the processor, and as long as the battery is not removed the settings will remain intact. If, however, the battery is removed or replaced, it will be necessary to perform the simple tone peaking calibration steps again.

OPERATION

The Badger smartbadge has a number of modes to allow it to be calibrated and operated in the prescribed ways.

When the battery is initially installed, the smartbadge goes into "calibrate mode", sounding a constant tone to enable the user to adjust the piezo for the optimal/loudest pitch. See Mode C for further instructions for this once-only condition.

The Badger operates in one of several modes when triggered by the pushbutton. Normally the smartbadge is in Mode A whereby the owner's callsign is sounded once with each actuation of the pushbutton. At the end of the callsign annunciation, the Badger goes to sleep to conserve battery power.

The operating mode may be changed by pressing and holding the pushbutton at the end of the callsign annunciation sequence. It may take some practice to get this timing

right for pressing of the pushbutton, but you'll get it every time when you press the button right at the end of the last callsign letter sounding. When this pushbutton press (and hold) is detected, a series of Morse letters are annunciated in sequence, sort of like an audible menu list, with each letter signifying a specific mode as indicated below. When the operator hears the desired mode letter, the pushbutton is released and that mode of operation is entered. The modes are described below according to their assigned letter.

Mode A – Normal
The callsign is played once whenever the pushbutton is tapped and then the SX chip goes to sleep. Tapping the pushbutton at any time during callsign play will silence the Badger and put it to sleep.

Mode B – Beacon
The callsign is played repeatedly, separated by a short pause in between iterations. Holding down the pushbutton for a second anytime during this mode will silence the Badger, return it to Normal mode (A), and put it to sleep.

Mode C – Calibrate
A steady tone is played by the piezo device, enabling the user to peak the frequency for that specific device and battery voltage. A single quick tap on the pushbutton will raise the frequency of the steady tone. Two quick taps will lower the frequency. The operator should adjust the piezo tone for maximum output. Holding the pushbutton for one second stores the tone setting, returns the Badger to Normal mode and puts it to sleep.

Mode D – Iambic Keyer
The paddle inputs to the SX chip are activated. Grounding the dit input sounds a short tone, and grounding the dah input sounds a longer tone. Iambic operation is achieved by dit/dah insertion, as appropriate. Actuating the pushbutton for one second returns the Badger to Normal mode and puts it to sleep.

Mode E – Morse Speed Adjust
A constant series of Morse dits is played. Tapping the pushbutton once increases the Morse speed and tapping the pushbutton twice lowers the speed. Actuating the pushbutton for one second stores the speed setting, returns the Badger to Normal mode and puts it to sleep.

Mode F – Straight Key
The piezo and LED are sounded whenever the dah input line is grounded. Connecting a straight key to the dah line will allow the Badger to perform as a classic code practice oscillator. Actuating the pushbutton for one second returns the Badger to Normal mode and puts it to sleep.

Mode G – Piezo Only
Only the piezo will be sounded for callsign and keyer annunciation. The Badger is returned to Normal mode and put to sleep.

Mode H – LED Only
Only the LED will be blinked (i.e., no piezo sounding) for callsign and keyer annunciation. The Badger is returned to Normal mode and put to sleep.

Mode I – Both Piezo and LED
Both the LED and piezo will be used for callsign and keyer annunciation. The Badger is returned to Normal mode and put to sleep.

PARTS LIST

1 - pc board with pre-attached components: SX chip, pinheaders P1, P3 and P4, and resistor R1.
1- piezo sounding device X1
1- SuperBrite LED D1
1 - resistor R2, 4.7K, 1/4W (yel-vio-red)
1 - resistor R3, 470-ohm, 1/4W (yel-vio-brn)
1 - transistor Q1, 2N2222A
1 - capacitor C1, 0.1uF ("104")
1 - pushbutton PB-1
1 - pinheader jumper J1 (comes mounted on pinheader P4)
1 - 3V lithium battery BATT-1
1 - battery holder
1 - laminated callsign badge
1 - plastic badge clip
1 - clear acetate for pc board protection

CONSTRUCTION

The Badger smartbadge can easily be constructed within an hour by using the instructions, graphics and photos on the following pages.

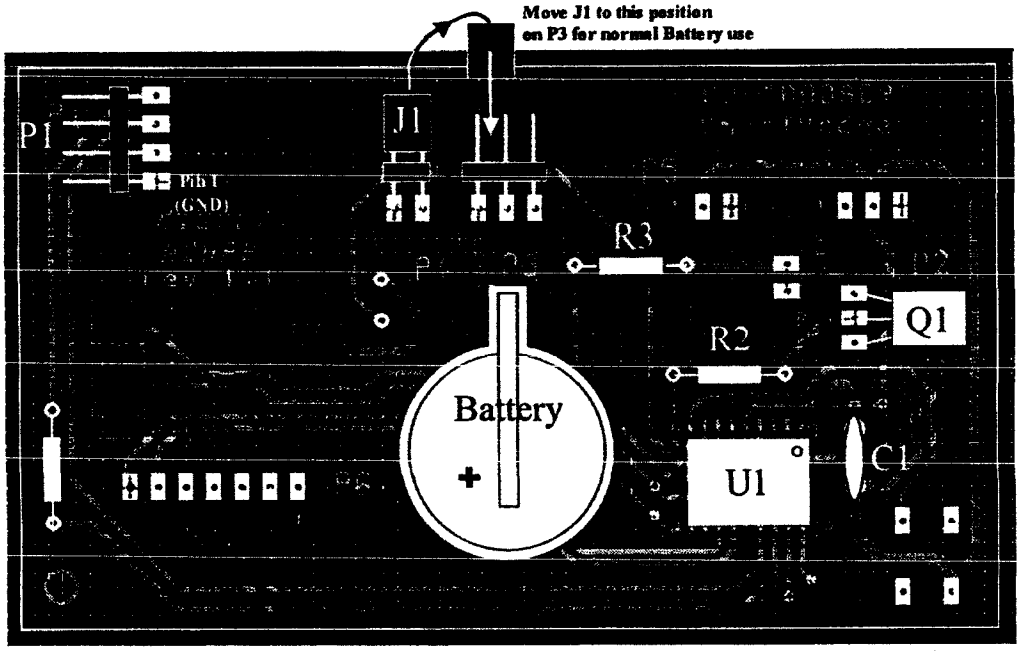


Figure 1 - Assemble the components on the backside of the pc board according to the diagram shown above. Insert and solder R2, R3, C1 and the battery holder. Orient Q1 so the flat side is down against the pc board when its three leads are inserted and soldered in the pads as shown. Move shorting jumper J1 over to the leftmost two pins (1 and 2) on P3 when you are ready for normal battery operation. (Note: when the Badger is being stored and not used for some time, it is advisable to move jumper J1 back to the original position to disconnect the battery.) Once all components are soldered in place, cut a rectangular hole in the CLEAR ACETATE sheet so the plastic fits down flat on the components and around the battery. Glue in place using several daubs of silicon adhesive (like RTV) to hold the acetate down.

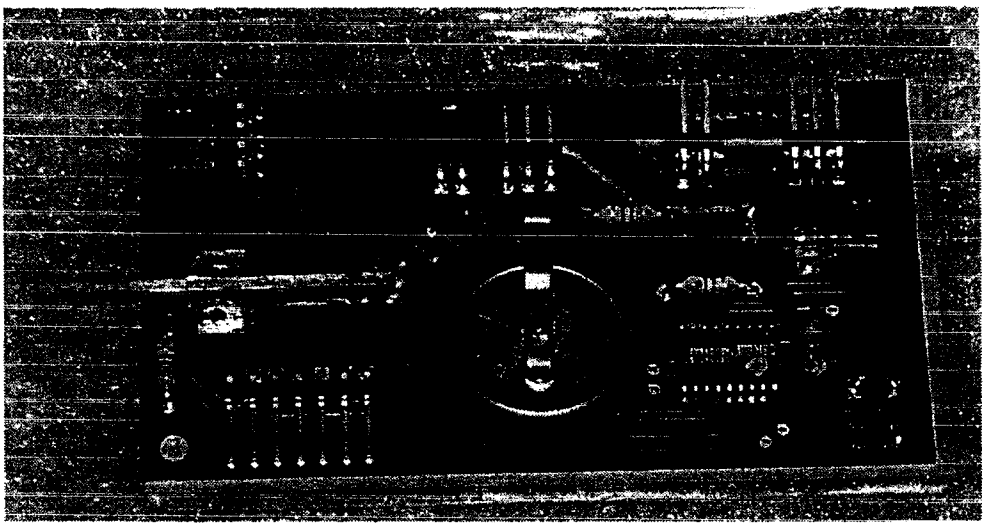


Figure 2 - The back side of the Badger. Note that the various pin header "options" have all been populated, and that a miniature paddle was fabricated on the left side of the smart badge. The 3V lithium battery is held in place via a clipped socket in the center of the board, and the surface mount SX microcontroller is just to the right of it. It's hard to see it in this photo but there is a piece of clear acetate glued over top the components on this back side of the Badger, protecting it from possible shorts when worn.

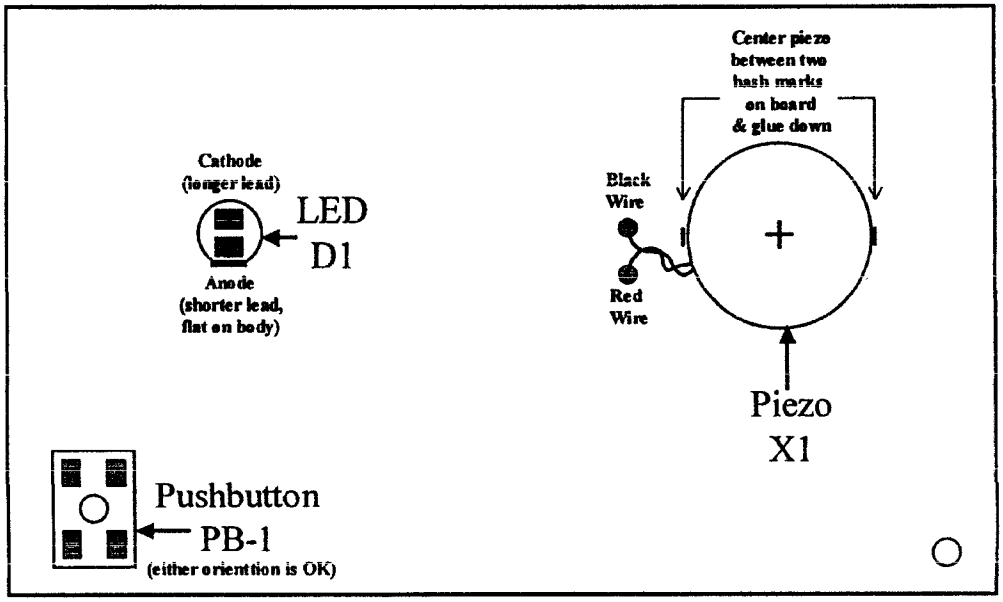


Figure 3 - This figure represents the front side of the Badger pc board with the laminated badge removed.
a) Center the PIEZO as shown over the plus sign and centered between the two hash marks on the pc board. Using SuperGlue (or equiv.) glue the piezo onto the surface of the pc board. Strip and solder its two wires and solder them into the pads as shown above.
b) Insert the LED into the two holes as shown above, placing the longer lead of the device into the top pad, and the shorter lead (with the flat on the side of the LED body) into the bottom pad. Solder and trim the wires.
c) Insert the PUSHBUTTON switch into the four holes in the corner of the pc board. (The switch is symmetrical and the orientation doesn't matter.) Solder all four leads and trim them off close on the back side of the board.

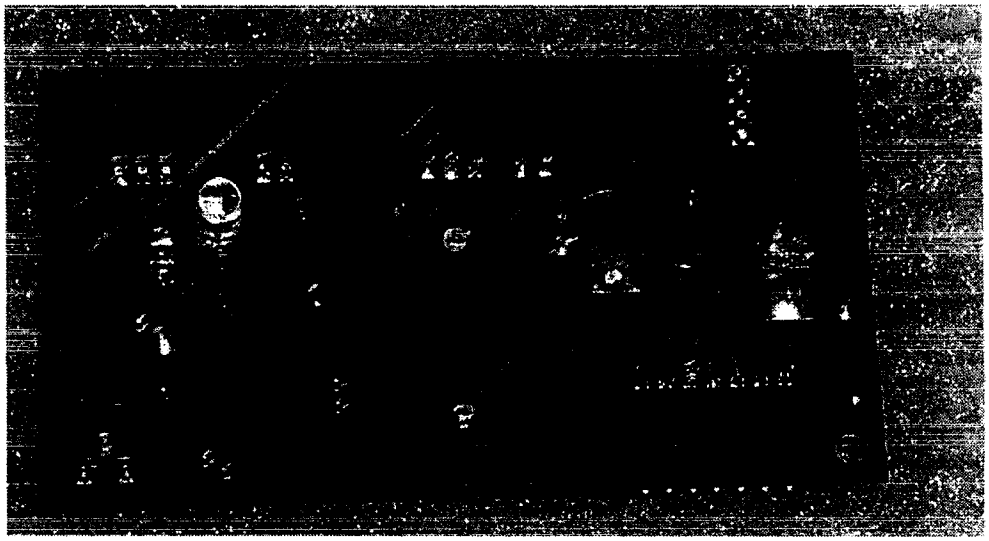


Figure 4 - The front side of the Badger pc board, with laminated badge removed. Only three components are mounted on this side: the pushbutton (left), LED (middle) and the piezo (right). The laminated badge with corresponding holes slips over the components and is glued to the surface of the board.

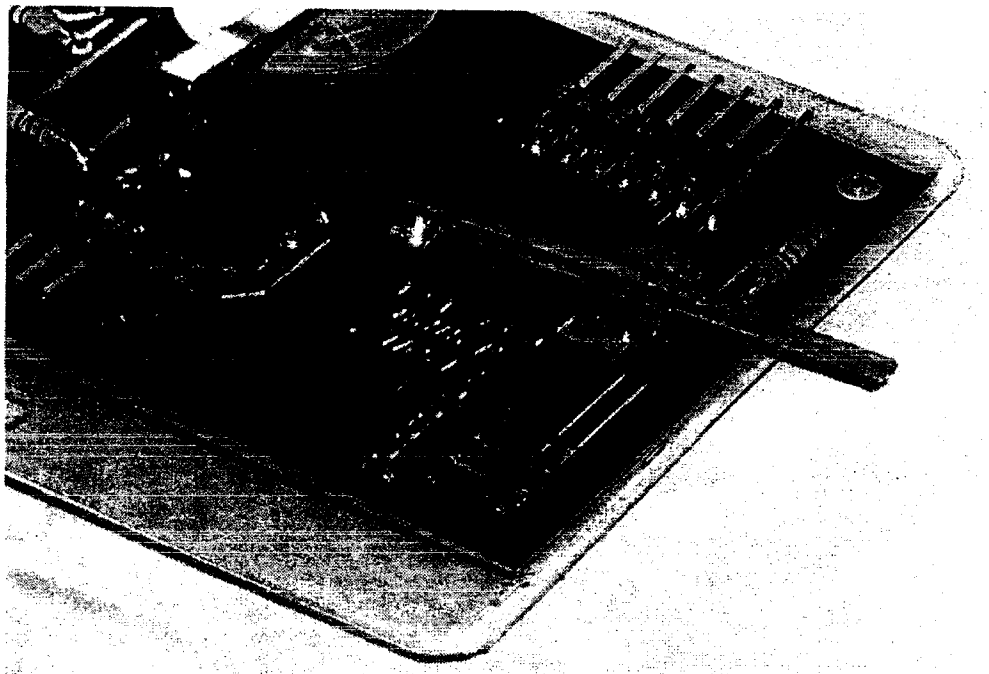


Figure 5 - A homebrew paddle may be constructed from pieces of scrap pcb material and soldered to the larger pads as shown above. This paddle will enable the smartbadge owner to send Morse code using the iambic keyer mode of the badge.

Figure 6 - This is how the laminated badge comes in the kit. The builder must enlarge the hole in the "Badger's belly" using a pair of sharp scissors. Small cuticle scissors work best for this. Cut to the outside of the circle and the badge will fit over top the piezo for a nice snug fit. Some very slight enlarging of the pushbutton hole may be necessary. If the badge doesn't fit on the components snug enough to hold it in place, use several daubs of silicon adhesive to hold the laminated badge on the board.

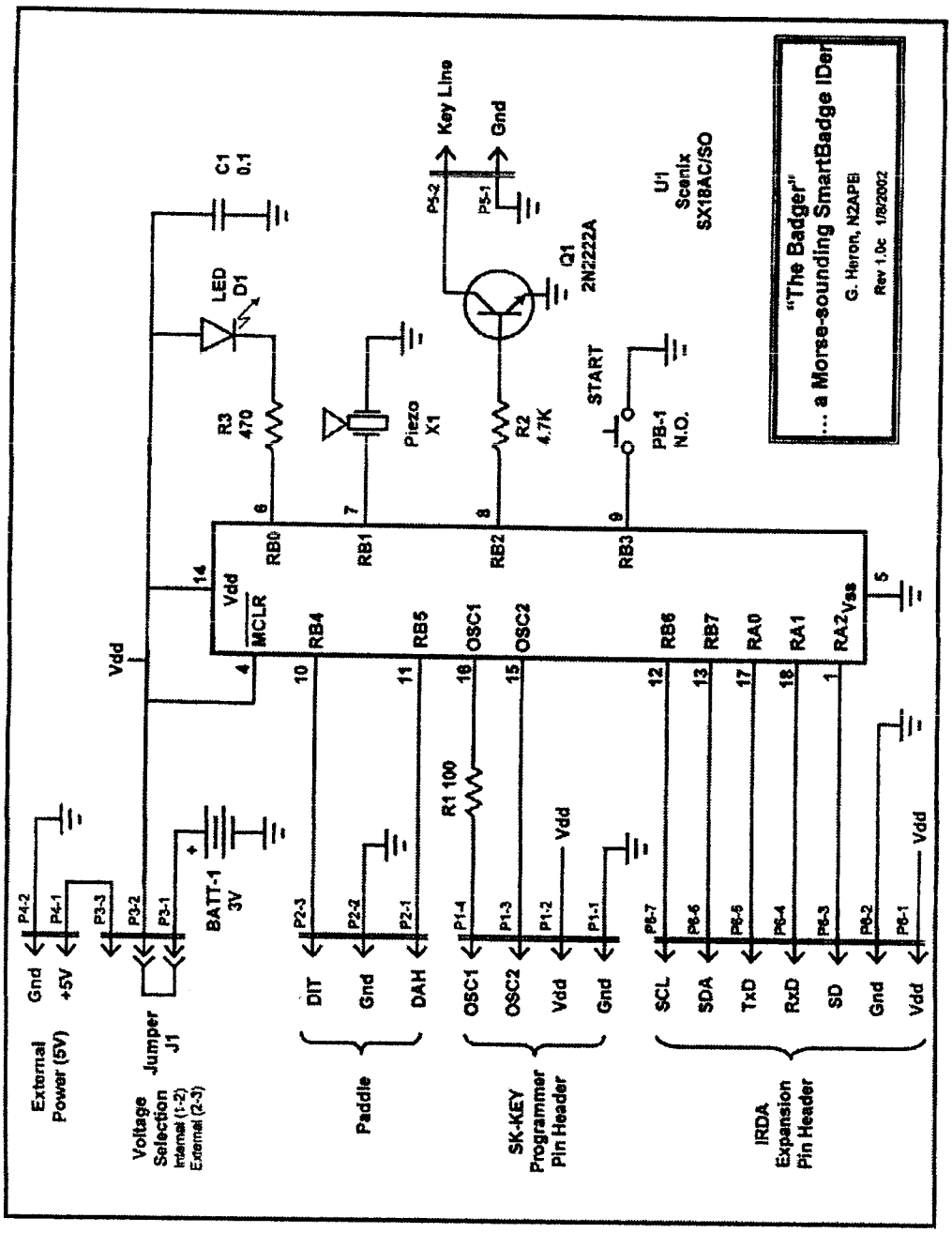


Figure 7 - Schematic of the Badger SmartBadge.

SUMMARY

No matter how you use the Badger - as a regular callsign badge, as a fun novelty topic at a radio club event, or as a keyer for your portable QRP rig - the kit is an exercise in understanding how simple microcontroller devices can be programmed to create a real fun and useful ham project.

If you have problems or questions in building/using this smartbadge kit, please contact me and I'll try to help.

Please be sure to visit the online Badger web pages too. The latest building information, tips, corrections, source code listings, and newest software versions are always maintained there. Point your web broswer to: www.njqrp.org/badger/

Thanks go to the club's good friends for helping in the creation of this kit: Dave Benson K1SWL for designing the pc board; Paul Harden NA5N for designing the graphics, and Joe Everhart N2CX for ideas, test and encouragement along the way.

73, George N2APB

George Heron, N2APB
2419 Feather Mae Court
Forest Hill, MD 21050
email: n2apb@amsat.org