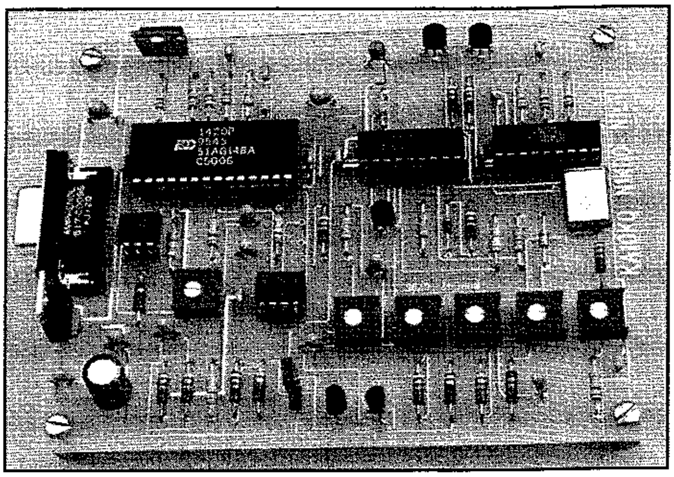
By Jeff Otterson, N1KDO, Peter Gailunas, KA1OKQ and Rich Cox, N1LTL

Build a $60 Talking Repeater Controller



This inexpensive programmable controller features stored voice, simplex or duplex repeater control and low power consumption.

Recent repeater projects (portable repeaters for disaster sites and a repeater system with voted remote receivers) inspired us to build a low-cost, easy-to-build repeater controller. The projects require a controller that can be built for less than $60, with enough features to control a duplex repeater, a link or a simplex repeater. Other required features include remote control and programming via DTMF, a hang timer, time-out and ID timers, CW ID and stored voice messages for all applications except link control. This article describes the product of our work. Partial kits and assembled controllers are available (while they last).[1]

Repeater Controller Mode

The controller has an identification (ID) timer that can trigger one of two stored voice IDs, or the CW ID. These IDs are controlled by a user-programmable ID timer with intervals from 10 to 2550 seconds. The controller plays the *initial ID* immediately after keying if there has been no activity for an entire ID-timer interval. "Welcome to N1KDO repeater" is a typical initial ID. The *normal* ID, something like "N1KDO repeater," plays after each ID-timer interval with repeater activity.

The controller tries to be polite in several ways. First, it IDs between user transmissions if possible. That is, it IDs immediately when a user's carrier drops with 60 seconds or less remaining before the end of the ID interval. If a user transmits during a voice ID, the controller cancels the voice ID and plays the CW ID instead. Also, if a user keys the repeater and the controller plays the initial ID, the normal ID interval begins. When the interval ends, a normal ID plays only if there was additional activity after the initial ID. This prevents unnecessary long-delayed IDs after "kerchunks."

The controller provides a hang timer and a courtesy tone. (A hang timer keeps the repeater's transmitter on for a short time after a user unkeys.) This reduces repeater-transmitter cycling and eliminates some extraneous squelch crashes for users. You can set the hang-timer for intervals of 0.1 to 25.5 seconds. The courtesy tone sounds after a user's transmission has ended *and the time-out timer has been reset.*

You can set the user-programmable time-out timer for periods of 1 to 255 seconds. The time-out timer prevents damage to the repeater's transmitter in the event a user sits on his microphone before starting a long ride, or the repeater's receiver becomes unsquelched for some reason. The time-out message plays twice: once when the time-out timer expires and again when the time-out condition ends. Thus, listeners know about a time-out condition immediately, and the offending operator knows when the time-out condition ends.

A tail message can play after a programmed number of hang-timer cycles have elapsed. The tail message might advertise a net or club meeting, warn of inclement weather, etc.

Link Controller Mode

The controller can control link radios for remote receivers or split-site repeaters. In link-controller mode, the controller does not use any voice messages—only CW IDs. You can build link controllers without the ISD1420 and associated support circuitry, which lowers the cost of link support. In most link-controller modes, the hang time is zero.

Simplex Repeater Controller Mode

The controller can also run a "simplex repeater." A simplex repeater records up to 20 seconds of received audio, then transmits the recorded audio. In this mode, the controller IDs in CW at the end of each ID interval. The simplex mode requires a control operator on duty; FCC rules do not permit automatic control of voice simplex repeaters.

Power Consumption

This controller is ideal for remote solar and battery powered applications. In standby mode, the circuit draws less than 10 mA. The greatest current consumption occurs while recording voice messages; that's under 30 mA. Normal repeater operation requires less than 20 mA.

Circuit Description

The controller consists of a Microchip PIC16C84 microcontroller IC, a Mitel/Teltone M8870 DTMF decoder IC, an ISD1420 voice record/playback IC and an op-amp IC. Refer to the digital-control schematic in Figure 1.

The heart of the controller is the Microchip PIC16C84 microcontroller (U1). This IC provides all the timers, CW generation, DTMF validation, and other digital I/O requirements of the controller. It uses the 3.58-MHz clock generated by the DTMF decoder.

The M8870 (U3) decodes DTMF input by filtering the audio signal into its high and

[1] Notes appear on page 40.

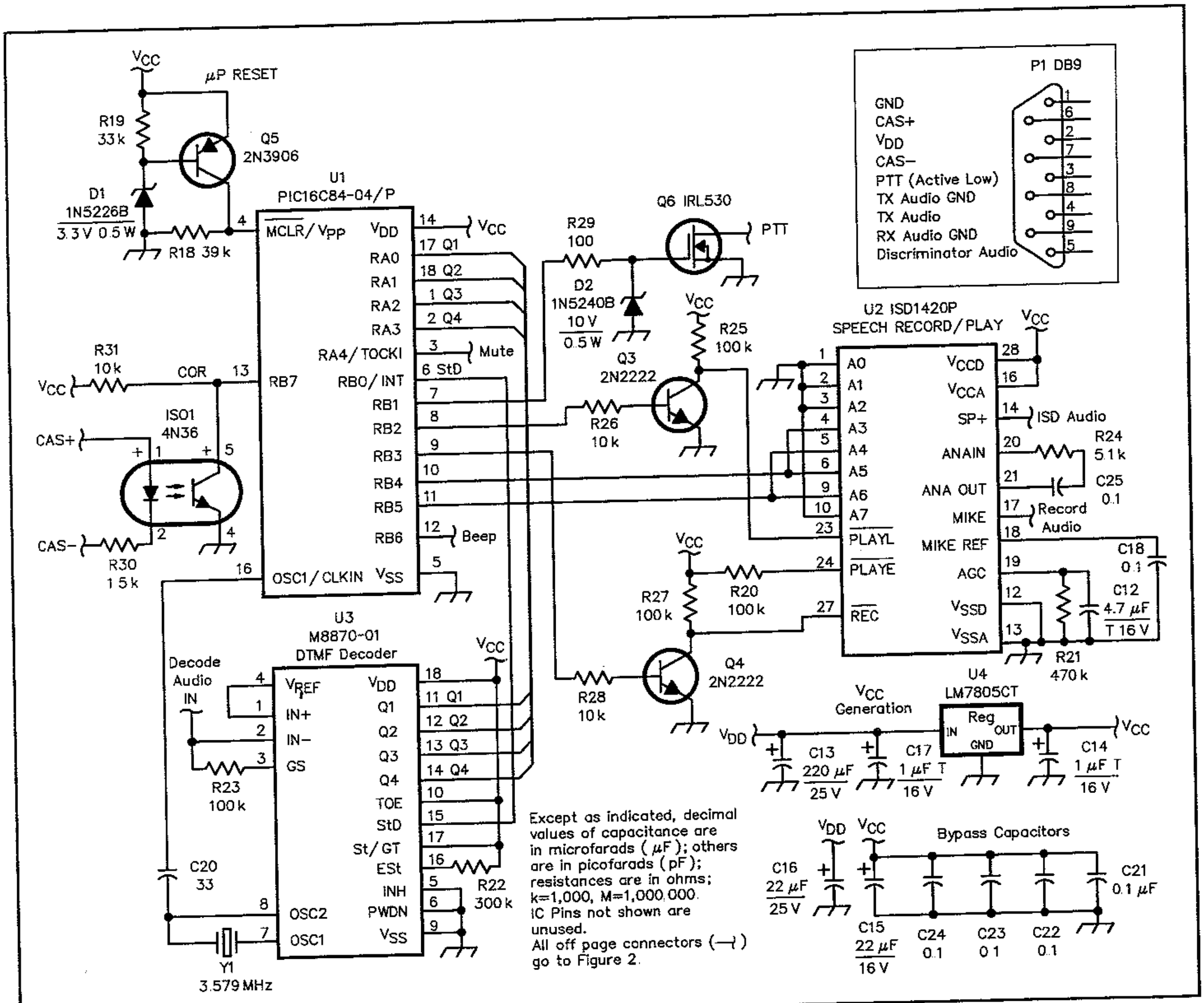


Figure 1—Schematic of digital circuitry. All resistors are ¼-W, 5%-tolerance components unless labeled otherwise. All polarized capacitors are 16-V electrolytics, all nonpolàrized are 50-V ceramic discs, unless labled otherwise. See Table 1 for additional parts information.

low components and counting the frequency of each component. This approach makes it much less likely to detect voice as a DTMF digit (false decode). When it decodes a valid digit, U3 raises the StD (delayed steering) lead voltage, to inform the 16C84

The ISD1420 (U2) stores speech messages by recording analog levels into flash EEPROM cells. The controller application implements four messages of about five seconds each. The device's address-line configuration allows messages to start at the 0, 5, 10 and 15-second addresses. The 16C84 controls the message-address, playback and record functions.

Audio processing (Figure 2) uses an optional deemphasis circuit that applies a –6dB/octave slope to the receiver audio This allows the controller to work with the receiver's discriminator output. If you prefer to use a flat audio source, such as a line or speaker output, omit C3 and make R3 and R4 100 kΩ each.

An FET (Q1) mutes the transmit audio when the receiver squelch is active or DTMF tones are present. A simple audio mixer combines audio from the receiver, U2 and beep-tone audio for the transmitter audio input.

Software

The 16C84 does absolutely nothing without software. We can loosely describe the software operation as a polled loop, with interrupt-based timing.

The controller's software was assembled with Microchip's MPASM. The source and object code are available for unlimited non-commercial use by amateurs worldwide. It can be downloaded from the Internet.[2] Development tools for the PIC microcontroller (MPASM and MPSIM) are also available on the Internet.[3] Several sources on the Internet describe the construction of a programmer for the PIC 16C84 (See the References.)

Radio Interfacing

The controller uses a female DB9 connector for all signals. It requires receiver audio and a signal-present indication (CAS) from the receiver, 13.8-V dc power and sends transmit audio and PTT signals to the transmitter. Be very careful when wiring dc power to the controller; reverse polarity will destroy the ICs.

The hot side of the squelch control is usually a good source of receiver audio. This is emphasized audio that requires processing through the controller's optional deemphasis circuit. Optionally, you can take audio from later in the receiver's audio chain, where it is already flat. Take care that the radio's volume control does not affect this audio level. Receiver audio that still carries emphasis has a very "tinny," unnatural sound to it.

The receiver must provide a signal-present indication (sometimes called COR, RUS, CAS) to the controller. Because of the varieties of polarity and state that this signal can take, we have implemented the controller's signal-present input with an optoisolator (ISO1). The anode and cathode f the LED in the optoisolator are exposed through current-limiting resistor R30. This allows easy interfacing to active-high, active-low and combinations of both to indicate the presence of a received signal to the controller. With clever wiring, builders can create CTCSS-and-COR, CTCSS-or-COR or similar configurations.

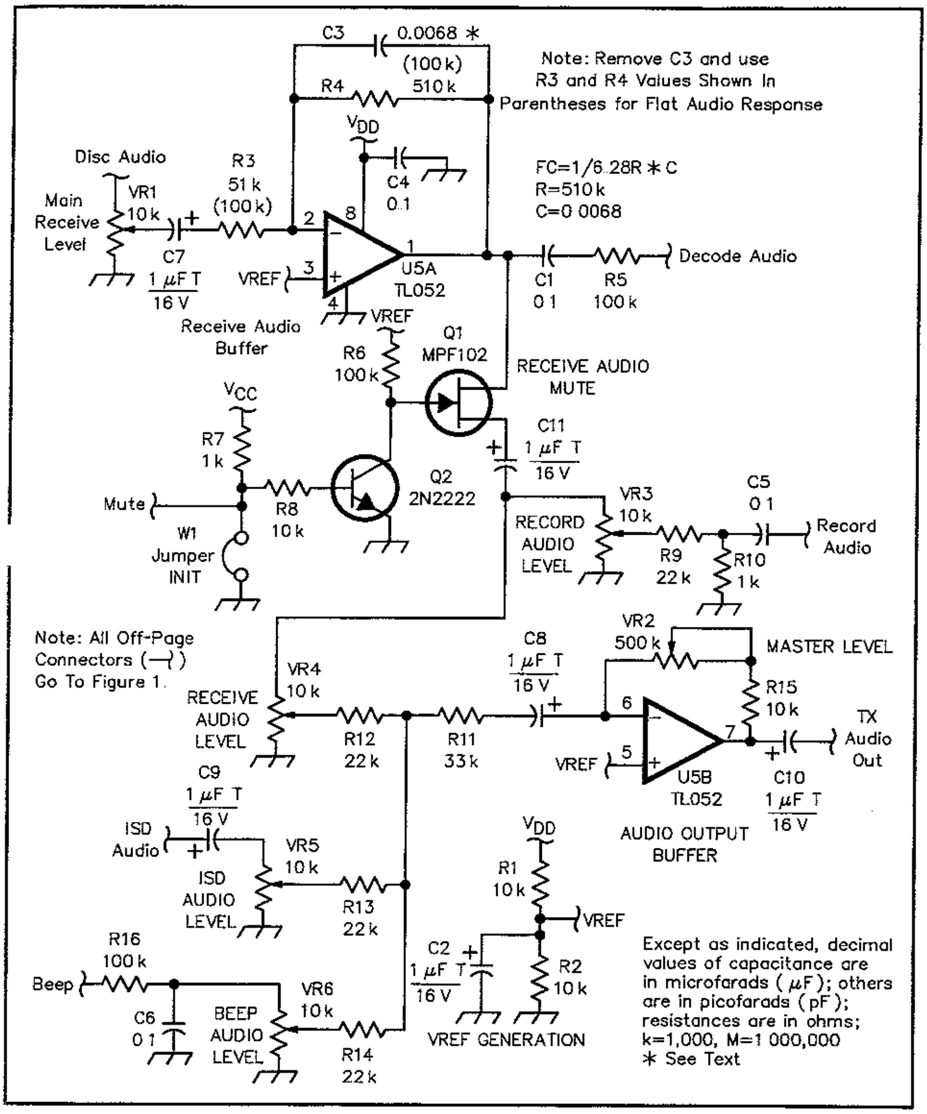


Figure 2—Schematic of audio circuitry. All resistors are 1/4-W, 5%-tolerance components unless labled otherwise. All polarized capacitors are 16-V electrolytics, all nonpolarized caps are 50-V ceramic discs, unless labeled otherwise. See Table 1 for additional parts information.

Table 1
Parts List

This article uses part designations that deviate from *QST* style to be consistent with the authors' materials. Parts that are adequately labled in the schematic have been omitted to save space. (See Note 4.)

C1, C4-C6, C18, C19, C21-C25—0.1 µF, 63 V, Z5U ceramic
C3—0.0068 µF, 100 V X7R ceramic
C20—33-pF, 100 V C0G ceramic
U1—PIC16C84-04/P 8-bit RISC microcontroller
U2—ISD1420P single-chip voice record/playback
U3—M-8870-01 DTMF decoder
U4—LM7805CT 5-V, 1.5-A voltage regulator
U5—TL052 or LM358 dual low-power Op-Amp
VR1, VR3-VR6—10-kΩ, 3/8″-square single-turn Cermet pot
VR2—500-kΩ, 3/8″-square single-turn Cermet pot
W1—2-pin 0.100″ center header
Y1—3.579545-MHz Color-burst crystal
8-pin DIP socket for U1
18-pin DIP sockets for U2, U3
28-pin DIP socket for U4

Feed the transmitter audio directly into the transmitter microphone input. The **MASTER LEVEL** control, VR2, sets the audio level sent to the transmitter. Adjust the transmit-audio level for proper deviation with the aid of a service monitor or deviation meter.

A power MOSFET (Q6) essentially provides a closure to ground for the PTT line. This circuit can key many transmitters directly. For other transmitters, Q6 can drive a small relay to key the radio. Although Q6 can handle several amperes safely, the PC board traces are very thin. We recommend that no more than 100 mA flow through them.

Adjusting the Audio Levels

Preset all potentiometers to midrange. Connect a scope probe or DVM to pin 15 of U3. (Use the power-supply ground for the scope's ground or the DVM's return.) Key a radio on the input frequency, send DTMF tones and adjust VR1 (**RECEIVE LEVEL**) until a 5-V level on pin 15 indicates reliable DTMF decoding. Disconnect the scope or DVM. Adjust VR2 (**MASTER LEVEL**) for proper transmitter deviation, as measured with a deviation meter or service monitor. Adjust VR6 (**BEEP LEVEL**) to set the courtesy- and CW-tone level.

The easiest way to adjust U2's input and output levels is to select the simplex repeater mode and record messages until the audio sounds right. VR3 adjusts U2's record level. Adjust this control for the best-sounding recorded audio. VR5 sets the playback level. Adjust this control for best acceptable transmitter deviation on message playback. VR4 sets the received-audio level relative to the other sources in the transmit audio; its setting may not vary from midpoint.

Initializing the Controller

To initially program your secret code into the controller, you must apply power to the controller with the pins on the **INIT** jumper, (W1) shorted, putting the controller into the initialize mode. Remove the jumper a few seconds after you apply power. Initialization resets all of the values stored in the EEPROM to their defaults, and readies the controller to accept a new four-digit secret access code. This resets the CW ID to the default value "DE NHRC/2" as well. When the controller is in the initialize mode the courtesy tone is 1/2 second long, instead of the usual 1/5 second. Key up and enter your four-digit access code. The controller should respond with the normal (1/5-second) courtesy beep. The secret access code is stored in nonvolatile memory in U1. Use this code as the prefix for all commands you send to the controller.

Programming the Controller

The data in each storage location is loaded with a sequence of eight DTMF digits: the four-digit access code plus the address (two hex digits) and the new data (two hex digits). The DTMF * digit represents hex E, and the #, hex F For example, let's program the hang timer (address 04 hex) with five seconds (50 decimal = 32 hex) Assuming that your access code is 1234, key your radio; press 1, 2, 3, 4, 0, 4, 3, 2; then unkey. If the OK message has been programmed, the controller responds with the CW message, **OK**. If the NG message has been programmed, and you enter an invalid address, the controller responds with the CW message **NG**. The range of valid addresses is 00 through 3F hex. The controller uses 40h and 41h for message play and record commands.

The controller timers have three different resolutions. All timers have 8-bit locations that can hold any value from 0 to 255. The hang timer is set in 0.1-second increments To program a hang time of 5 seconds, convert 50 (decimal) to 32 (hex) and store it in the hang-timer preset location. The command sent to the controller would be "cccc0232" where "cccc" is your secret access code. The time-out timer is in 1-second increments (0 to 255 seconds). The ID timer is in 10-second increments (0 to 2550 seconds).

Messages

There are four messages each, for voice and CW. (Detailed information is shown in Table 3 of the Otterson template package [4]) You can store voice messages up to 4 8 seconds each The controller removes about 100 ms from the end of each voice message to cut off any recorded squelch crash.

To record a voice message, enter the access code, then 410*x*, where *x* is the number of the message you want to program. Unkey after the DTMF sequence, then key up again, speak your message and unkey

You can play a voice message with the command 401*x*, where *x* is the number of the voice message you want to play (Don't forget the access code!) To play a CW message, use 400*x*

Record the tail message like any other message, but it will not play until you program the tail-message counter (address 05) to a nonzero value. Programming the tail-message counter to 0 disables the tail message.

You may want to have a family member or someone of the opposite sex record your voice ID messages The recorded audio sounds so natural that users have actually tried to call the ID message!

CW messages reside in the controller's nonvolatile memory. Their programming procedure is similar to that used for the timers. Each message has a fixed base address and a maximum number of characters. CW symbols are stored in a binary-encoded form. They're read from right to left, with a 1 representing a dash, and a 0 representing a dot. The leftmost 1 indicates the width of the symbol. To program the first letter of the ID message (D), you would enter your access code, the address (0E) and the encoded letter D (09): "cccc0E09" (where cccc represents your secret access code). The ID message can be up to 39 characters long and must end with the End-Of-Message character, hex FF CW messages play back at 19 2 wpm.

Selecting Controller Modes

The control operator sets controller mode via the configuration flags (address 01). You can simultaneously select multiple modes by adding their values together to set multiple bits in the configuration byte. Let's call the location CONFIG. To select the normal (full-duplex) controller mode, program CONFIG with 00. To select the link-controller mode (no ISD1420, CW messages only), program CONFIG with 01, and optionally program the hang timer (address 02) to 00. When in link-controller mode, you may want to have the controller pass DTMF tones to a downstream controller. Program CONFIG with 21 hex in this case. To select the simplex-repeater mode, program CONFIG with 02 In normal- or link-control mode, you can suppress the courtesy tone by adding hex 10 to the CONFIG value For instance, to use link-controller mode with no courtesy tone, program CONFIG with 11. Note that in either normal- or link-control mode, setting the hang time to zero also suppresses the courtesy tone. You can have the tail message (address 03) replace the courtesy beep by adding hex 40 to the CONFIG value

RFI

RFI is particularly troublesome at a repeater site. This controller, like any logic circuitry, can generate significant RFI. It is important to install the controller in a grounded RF-tight box.

Conclusion

We found that the low cost, variety of features, and low power consumption of this controller make it a winner for several of our repeater projects We are happy to share the results with our fellow amateurs and hope that many of you will find this controller useful and functional in your own repeater projects.

Acknowledgments

The authors' thanks go to Mike Martin at Prototype America in Manchester, New Hampshire, for his extra-speedy work in the rapid turnaround of our prototype boards.

Jeff Otterson, N1KDO, is a professional software developer and a student at Southern Polytechnic State University. He is a Technician licensee, first licensed in 1991. Jeff received his first electric shock at four years of age, "smoked up" some wire at five and has been hooked on electronics ever since. You can contact Jeff at 3543 Tritt Springs Way, Marietta, GA 30062; e-mail **otterson@ mindspring.com.**

Peter Gailunas, KA1OKQ, is a 1991 graduate of New Hampshire Technical College and is an Associate of Science in Industrial Electronics. He has been employed as an engineering technician since 1981. He presently works for DEKA Research and Development in Manchester, New Hampshire. Peter was first licensed in 1986 He entered Amateur Radio because he enjoys building electronic projects. Repeaters are his main interest. He presently owns five repeaters in New Hampshire and all of them are mostly home built Peter has been designing and building microprocessor-based repeater controllers since around 1987. Readers can reach Peter at 444 Micol Rd, Pembroke, NH 03275; e-mail **PGAILUNAS@ worldnet.att.net.**

Rich Cox, N1LTL, is an Electronic Engineering Technician, also with DEKA. He has a Technician license and has been active in Amateur Radio for four years. Rich has been developing ham-related electronics with Jeff and Peter for the past three years. You can reach Rich at 452 Brown Rd, Candia, NH 03034; e-mail **rcox@mv.mv.com**

Notes

[1] While supplies last, partial kits containing a programmed PIC16C84, ISD1420, M8870 and the PC board (those parts unavailable from Digi-Key) are available for $30 plus $5 for shipping. Assembled and tested controllers are $129 postpaid. Contact author Peter Gailunas, KA1OKQ, 444 Micol Rd, Pembroke, NH 03275; e-mail **PGAILUNAS@worldnet.att.net.**

[2] The source code and manual for the repeater controller is available on the Web **http://www.mindspring.com/~otterson/nhrc2** or (source code only) from the ARRL "Hiram" BBS (tel 860-594-0306), or the ARRL Internet ftp site: **oak.oakland.edu** (in the **pub/hamradio/arrl/qst-binaries** directory). Look for the file OTTERSON.ZIP. The source code is also available from N1KDO, send a blank PC-formatted diskette (5¼, 360 KB to 1.2 MB; 3½, 720 KB or 1.44 MB) and a stamped, self-addressed diskette mailer.

[3] PIC16C84 programmer information is available on the Web at many locations. Here are two of them:
http://digiserve.com/takdesign/pic-faq/hardware.html
http://www.paranoia.com/~filipg/HTML/LINK/ELE/F_PIC_faq.html

[4] A complete controller manual is available at the Web address in Note 2. The manual contains several useful tables and a detailed Parts List complete with manufacturers' names and Digi-Key part numbers. These tables are too lengthy to include in this article, but they are available as a template package. Contact the Technical Department Secretary at ARRL Headquarters (by any means described on page 10 of this issue) and ask for the February 1997 *QST* Otterson Template. ARRL members send $2 (nonmembers $4) for shipping and handling

References

PIC 16/17 Microcontroller Data Book Microchip Corporation, Chandler Arizona; Web **http://www.microchip.com**

ISD Data Book, "Voice Record and Playback ICs" (San Jose: Information Storage Devices, 1995); Web **http://www.isd.com**

"Telecom Design Solutions," *Component Data Book,* (Bothell, WA: Teltone Corp); or see Mitel's Web site **http://www.semicon.mitel.com/Products.html.**

Application Note MSAN-108, "Applications of the MT8870 Integrated DTMF Receiver," Mitel Semiconductor, Kanata, ON, Canada; Web **http://www.semicon.mitel.com.**

Linear Circuits Data Book, Volume 1, Operational Amplifiers, (Dallas: Texas Instruments, 1992); Texas Instruments doesn't have information on the LM358 on their Web site, but SGS Thompson does, at **http://www.st.com**