Microprocessor Repeater Controller, Part I

Add versatility to your repeater system.

by John Bednar WB3ESS

Not long after publishing an article on my Link Controller in the December 1989 issue of *QST,* I realized just how many repeater owners needed a repeater controller they could home-brew on a modest budget. My first single-chip microprocessor repeater controller had been in operation for almost 10 years. Before offering it to others, however, I decided to completely rewrite the software to add some new features. I knew that if the design were economical, it would bring those repeater owners with diode matrix IDers and intermittent touchtone control into the 21st century.

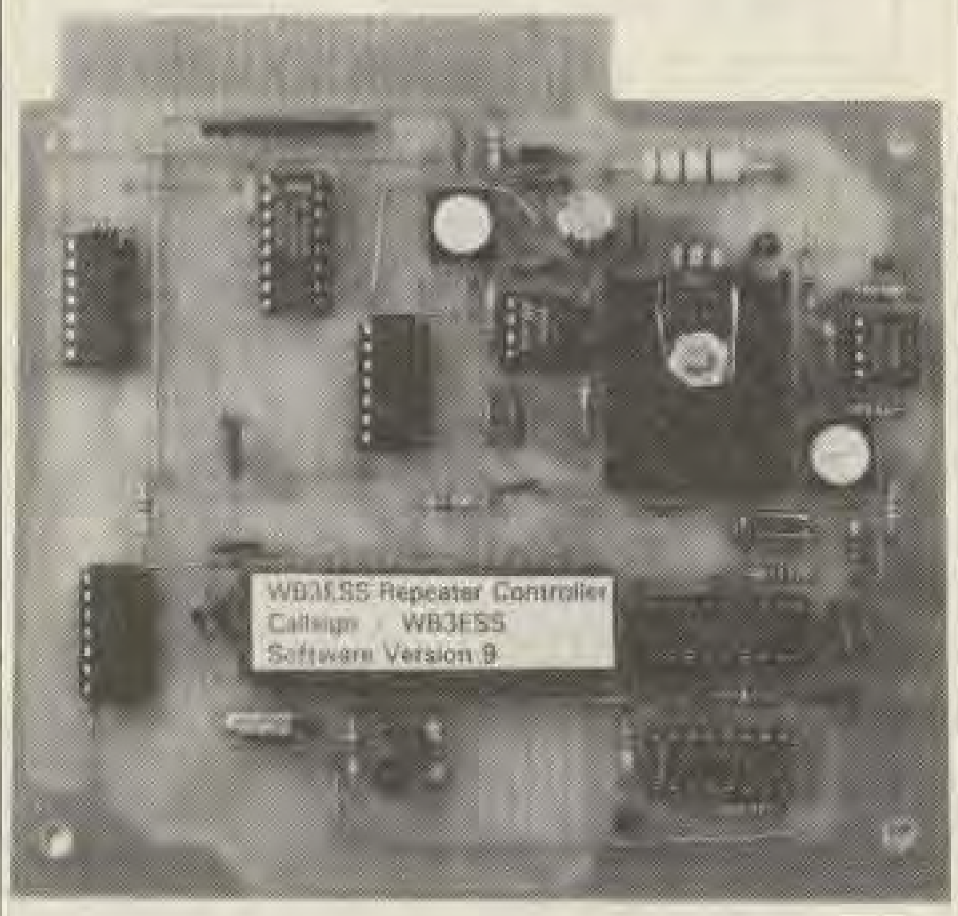

Photo A. The repeater controller consists of a computer board and an audio board. Shown above is the microprocessor board.

Selected Features

The heart of this controller is an 8749H single-chip microprocessor. To date, I am still amazed at how many features I was able to pack into it! Selecting the features required prudence, since the memory of the 8749H is limited to 2K bytes of EPROM. You could add external memory, but that would increase the size and cost of the finished product. It would also produce RFI, and you would have to mount the controller in a shielded box with feed-through capacitors to bypass the wires.

See Table 1 for a list of the features I selected. Macro capability, voice messages, reverse auto patch, and measurement of signal strength would be nice extras, but you'd spend seven to ten times more. The project in this article is for those who want an economical, easy-to-construct repeater controller with a wealth of useful features.

Overview of the Controller

The complete repeater controller consists of two circuit boards, one with the microprocessor circuits and the other with the audio and phone interfaces. With a modest junk box, you can build both of them for approximately $130. (I will be making the boards and programmed microprocessors available.) The microprocessor board (see Photo A) has nine outputs and two inputs for control and monitoring. All of these outputs are reserved for the user; they are not dedicated to any specific task.

Additional controller outputs are provided for autopatch, audio muting, repeater PTT, link PTT, and two outputs for the Link Controller Host PTT and Busy inputs. All user outputs are open collector type, able to withstand 30 volts, and sink 40 mA of current. The repeater controller has inputs for repeater CAS, link CAS, superuser, and link monitor, plus two reserved inputs for users to monitor things. All repeater controller inputs are CMOS, and offer a wide input voltage range to make interfacing easy.

Command Structure

The repeater controller has two priorities in the DTMF command structure: the "user" and "superuser" levels. None of the superuser commands can be executed when the controller is in the user mode, but all of the commands can be executed in the superuser mode. What's even nicer is that the owner can assign these priority levels to all of the 39 commands!

In most cases, you don't have to use the commands to change the CW speed, key-up delay, and hang-time available to all users. By restricting these and similar commands to superuser priority, they can not be executed unless the repeater controller is in the superuser mode.

When entering a command with superuser priority, the superuser pin must be low. If this pin input is high, only the lower priority user commands can be executed. This input pin can be connected to many different sources, the simplest being a controller output. Since output #1 is adjacent to the superuser pin, a solder ball across these two pins on the card edge connector will make the connection.

Another interesting source for the superuser input is a PL tone decoder output. With this type of connection the control operator would turn on a subaudible tone to enter superuser commands. With this type of external control of the command priority, the owner can adapt the repeater controller to whatever level of security is necessary for the environment.

Courtesy Beeps

When the repeater controller is in the non-link mode, you can choose a very short single beep, a short two-tone beep, or a no-courtesy beep. The decision to use the short single beep or the two-tone beep is based on whether output #9 is programmed high or low. I use this output to signal repeater users on whether a repeater function is on or off. By using this output to control some

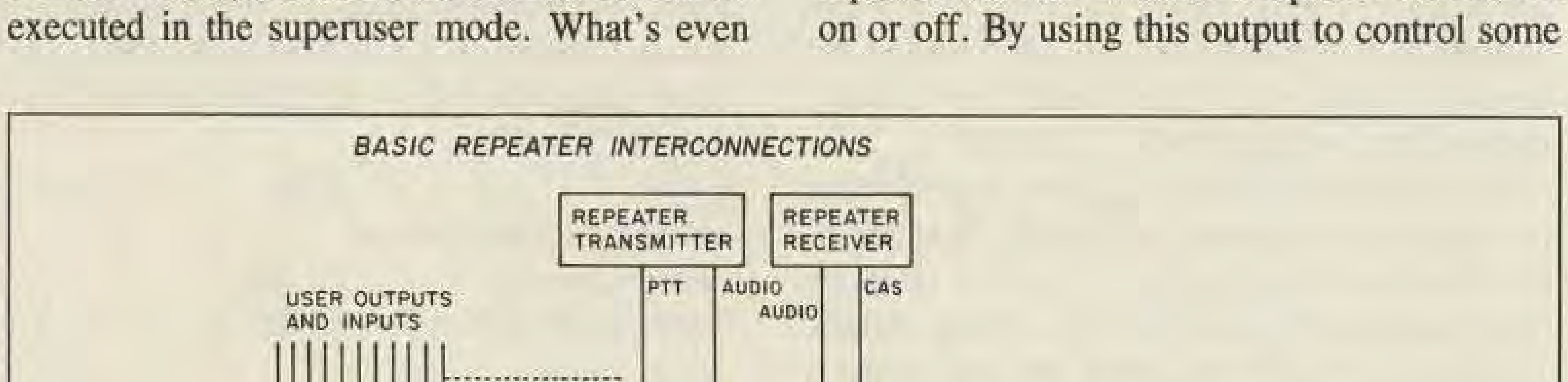

Figure 1. Block diagram of a basic repeater system using the repeater controller.

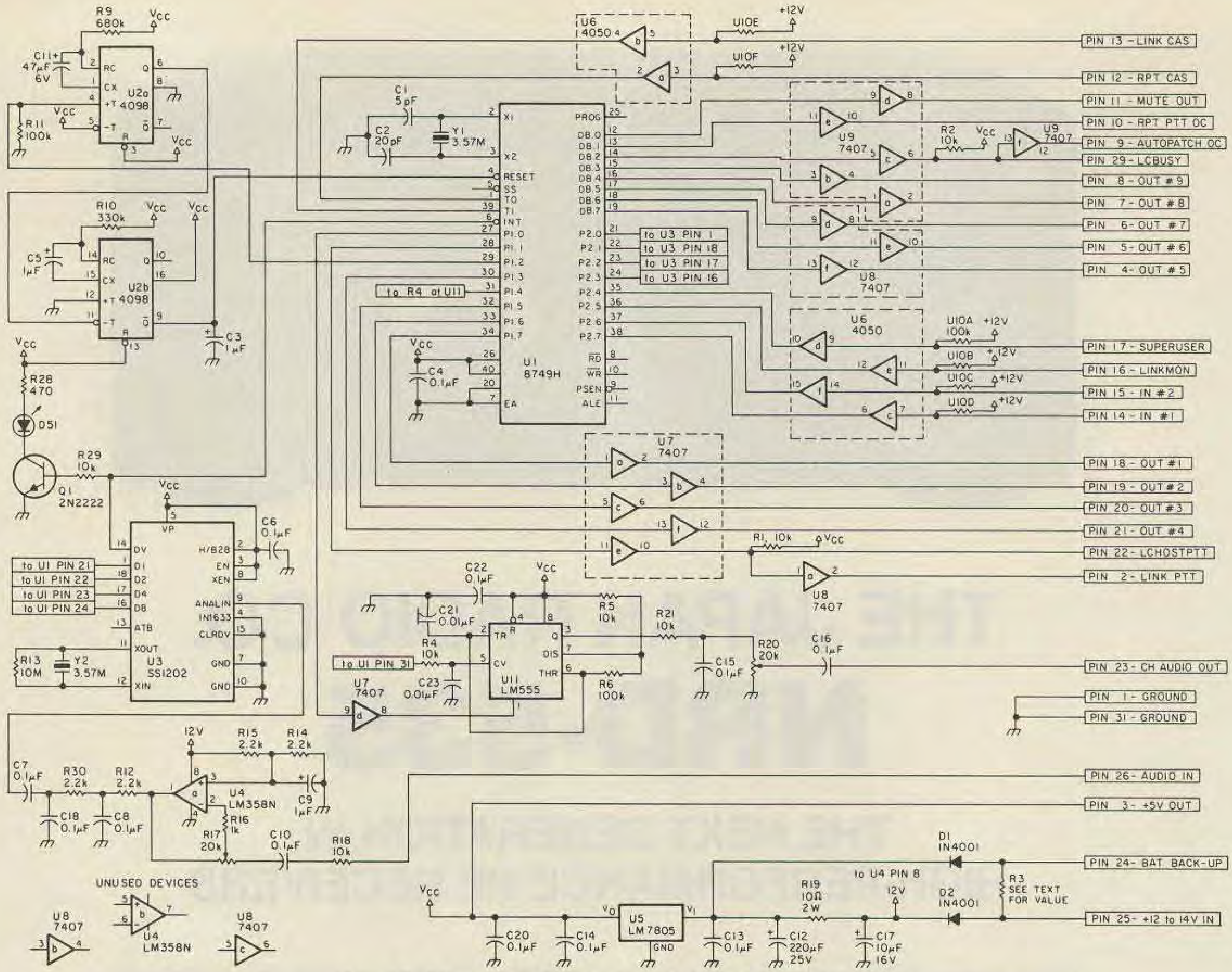

Figure 2. Schematic for the repeater controller computer board.

function, you can let your users know the on/off status of the function.

For example, you could use output #9 to turn the PL on and off. If you don't like courtesy beeps, you can simply program the courtesy beep delay to zero, and it won't sound during non-link operation! The link monitor input puts the repeater controller in the link mode via local or external control. (The repeater controller and hardware in this article is optimized for a repeater with a single link. You can add multiple links, but you have to build more hardware. I will include this information with kit orders.)

With this link monitor input, you can have an external device like a Link Controller turn the link on when a remote user wants to link into your repeater from another frequency. In link mode (linkmon input low), the computer uses an alternate set of courtesy beeps to let your users know that a link is enabled. If a user unkeys from the repeater frequency, a short double beep is sent; however, if a user unkeys from the link frequency, a dash-type beep is heard on the repeater. This simple selection of the courtesy beeps during linking operation instantly informs the users where the signals are coming from.

For additional flexibility, the pitch of the courtesy tones can be varied by changing components in the tone oscillator circuit (U11). Finally, if a user keys the repeater during the courtesy beep, the computer stops the tone so it won't interfere with conversation.

DTMF Features

Continuing with more features, the repeater controller can accept DTMF commands anytime, *even when sending CW.* When entering a command, the first digit must be valid for at least 200 ms. This is done to reduce the possibility of the controller being "triggered" by normal speech. Because of this delay, a short burst of the initial DTMF tone will be heard on the repeater, but all remaining DTMF tones will be muted.

Like the Link Controller, the repeater controller DTMF commands can be executed immediately by placing a "#" at the end of a command string. This feature can be used to control the repeater in the presence of other signals, or to string commands together. Normally, all DTMF tones are muted on the repeater and the link; however, by beginning a DTMF string with a "*" all remaining DTMF tones are transmitted over the repeater and link frequencies until the user unkeys.

This is useful for sending DTMF tones to a remote Link Controller board or some other external device. No need to worry about the initial burst of the "*" digit mixing up a remote Link Controller. Every Link Controller is already programmed to ignore invalid leading digits! Finally, an internal timer clears the DTMF digit buffer if the user doesn't unkey within three seconds of the last digit entered. This timer will aid the control operator if errors are made when commanding the repeater controller in the presence of other signals. If an error is made, the control operator simply waits three seconds and then re-enters the command.

Table 1. Repeater Controller Features

- Station ID, time-out timer, DTMF touchtone muting
- Nine outputs and two readable inputs for the user
- Programmable CW speed, hang-time, key-up delay, and courtesy beep delay
- Four-digit commands with programmable prefixes
- Programmable CW on/off read-back messages
- Programmable dual-priority level command structure
- Disable/enable repeater transmitters, link transmitter, time-out timer, and DTMF decoder.
- Autopatch and linking features
- Direct connect outputs for the Link Controller (uses commercial circuit boards and common parts)
- Multiple audio inputs and outputs with audio gating
- Phone interface, PL gating, and local speaker output
- All software, I/O, and timers are contained in one chip!

ID & Timeout

The repeater controller has a fixed ID interval of seven minutes, and the repeater will ID only when nobody is talking—unless the time-out timer ID is disabled. When disabled, the repeater will ID whenever the interval timer reaches zero (while users are talking).

To save valuable memory and eliminate extra transmissions by the repeater controller, I did not program the controller to do an "end ID." This is the type of ID routine where, a few minutes after the QSO has ended, the controller sends the station callsign and sometimes an extra message. I personally like a repeater controller without lots of chatter, and that weighted my decision.

To help reduce repeater key-ups caused by intermod bursts and dialing kerchunkers, I added programmable key-up delay to the software. When the repeater is being used, the key-up delay is unnoticeable. But after 30 seconds of no activity, the controller switches to the programmable key-up delay value, which is adjustable from 0 to 2.6 seconds.

The repeater controller time-out timer is fixed at the legal maximum of three minutes. Before the repeater times out, the controller sends a message to the users with a station ID. If the user is still talking, the transmitter, link transmitter, and phone patch are turned off. The transmitters stay off until the offender unkeys and realizes his mistake and transmits again.

There is no post time-out harassment from the repeater controller; that task is left up to the repeater users. Of course, if the repeater is timed out, a control operator can enter the command to disable the time-out timer, and the repeater transmitter will come on again. To be successful, the control operator must be able to capture the repeater receiver. The "#" feature must be used.

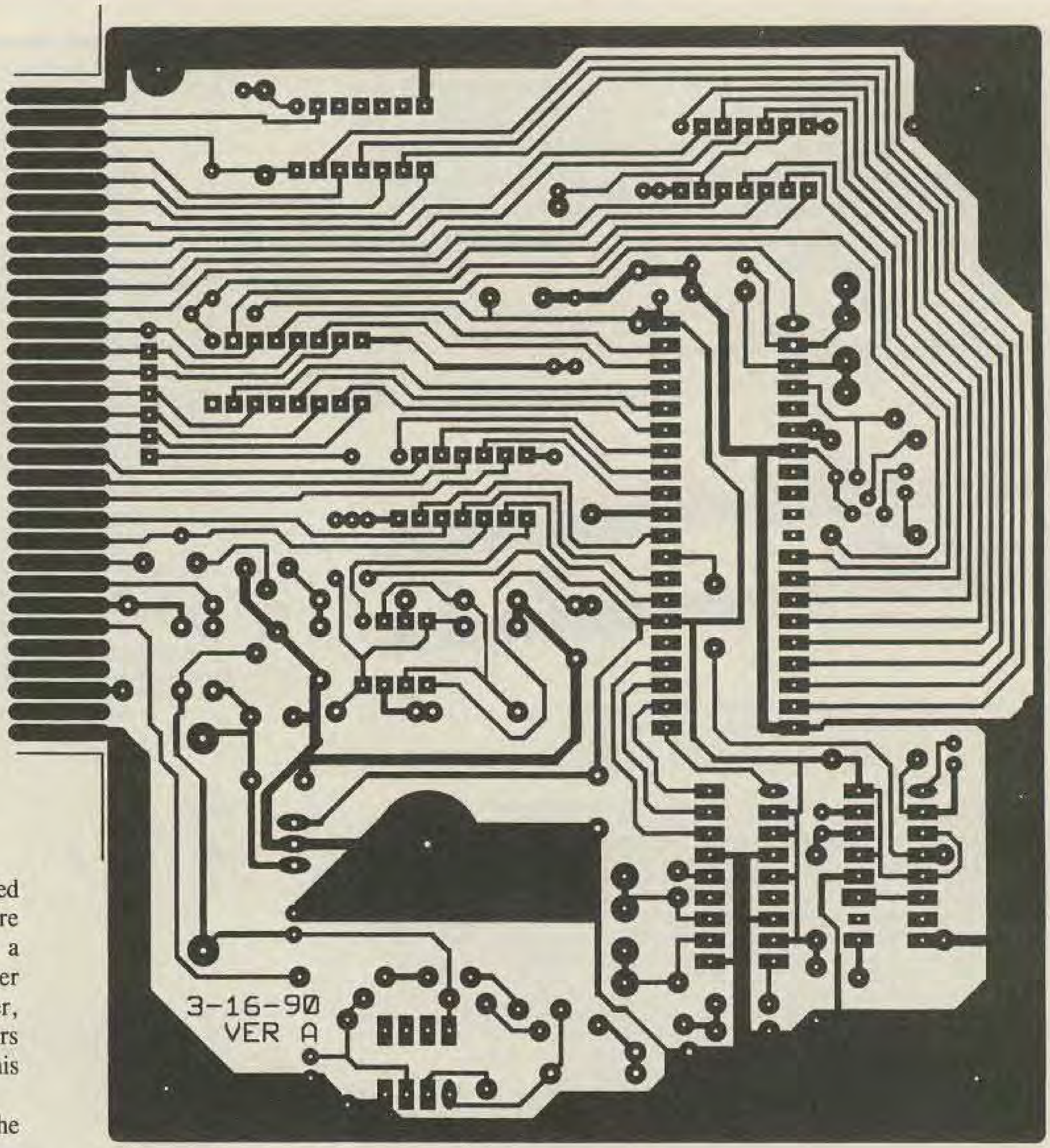

Figure 3. PC foil pattern for the computer board.

Table 2. Commands

Command Function	Powerup Priority
Output # 1 low	User
Output # 1 high	User
Output # 2 low	User
Output # 2 high	User
Output # 3 low	User
Output # 3 high	User
Output # 4 low	User
Output # 4 high	User
Output # 5 low or pulsed	Superuser
Output # 5 high	Superuser
Output # 6 low or pulsed	Superuser
Output # 6 high	Superuser
Output # 7 low or pulsed	Superuser
Output # 7 high	Superuser
Output # 8 low or pulsed	Superuser
Output # 8 high	Superuser
Output # 9 low, two tone beep	Superuser
Output # 9 high, single beep	Superuser
Read input # 1	Superuser
Read input # 2	Superuser
Autopatch on	Superuser
Autopatch off	Superuser
Increase keyup delay	Superuser
Decrease keyup delay	Superuser
Increase CW speed	Superuser
Decrease CW speed	Superuser
Increase courtesy beep delay	Superuser
Decrease courtesy beep delay	Superuser
Increase hang time	Superuser
Decrease hang time	Superuser
Disable time out timer	Superuser
Enable time out timer	Superuser
Disposable repeater	Superuser
Enable repeater	Superuser
Disable DTMF decoder	Superuser
Enable DTMF decoder	User
Disable link transmitter	Superuser
Enable link transmitter	Superuser
Change command prefix	Superuser

Autopatch

The repeater controller phone patch support is basic but novel. There really isn't enough memory to implement long distance lock-out, reverse patch, auto dial, or control from the phone. Despite this, the controller has several nice autopatch features.

For instance, when dialing the phone number, all digits are muted so that repeater listeners are unable to hear it. Additionally, there is an input on the computer board that allows owners to customize the autopatch with long distance lock-out or a patch limit timer if needed.

During autopatch calls, the microprocessor monitors input #2. If this input is pulled low, the autopatch will be terminated as if the OFF code had been entered from the touchtone pad. Since this input is scanned only when the microprocessor is not sending Morse code, the external signal will have to be latched until the patch goes off for correct operation.

Due to some clever software, this input functions identically to input #1 during non-autopatch periods. Therefore, input #2 may be *multiplexed* for both functions. The above features, plus being able to lock out users with the superuser priority, should aid control operators.

DTMF Commands

Initially, every repeater controller powers up with the same set of default commands. All commands are fixed at four digits long, except the editing commands, which are eight digits long. Since the leading two digits of each command can be edited, unique command sets can be created.

Because the repeater controller has two command priority levels, it's not absolutely necessary to change the prefix of all 39 commands. By making the access to the superuser function unique, 30 of the commands are instantly protected from outside parties (30 of the commands power up with superuser priority).

All commands are listed in Table 2, along with the power-up priority of the command. Most of these commands are self-explanatory. The four pairs of increase/decrease commands simply change timing values in the software. The owner can use a touchtone pad to increase or decrease timer values in fixed increments. The command can be repeated to make larger changes.

I chose this method for two reasons—it keeps the operation simple and it conserves precious memory. Although it's not as glorious as programming delays in milliseconds, the result is identical. Also, my first repeater controller used this method of changing delay parameters, and it has worked well to date.

Command Editing

All of the controller commands have CW read-back to confirm the action, except for the autopatch ON command where it wouldn't make sense. To make the interfacing easier to the user inputs and outputs, I thought it would be convenient if I could control the *sense* of the CW read-back message. The power-up standard is output/input—high reads back as OFF and low reads back as ON.

If you like, you can reverse these messages when you're programming the prefix codes. Once again a simple but effective method was chosen to do this. To keep the standard read-back messages, program a 1, 3, 5, 7, 9, or A as the first digit of the prefix. To reverse the read-back messages, program a 2, 4, 6, 8, 0, #, or B as the first digit of the prefix. To eliminate the read-back message entirely, program a C or D as the first digit.

Since the "*" is reserved to pass touch tones, it can not be used as a valid first digit. If an "*" is programmed as a leading digit by mistake, simply program a new prefix with a different leading digit to correct the error.

You can program outputs #5 through 8 for pulsed low operation or a static (no pulse) function. Because of memory limitations, pulsed operation could only be added to the *output low* commands of these outputs. If a leading prefix digit of 1,2, or D is programmed, the output will pulse low for 150 ms and then return to a high state. Also, these three digits allow the owner to select one of the three possible read-back messages. If any other leading prefix digit is programmed, the output will behave like the other outputs (no pulses).

I am sure this flexibility in read-back messages is welcomed, as you don't have to invert signals in hardware to make the CW read-back message correct.

Since the above method worked so well, I decided to use the same scheme to program the command priority. To designate the command as user priority, simply select a 1, 3, 5, 7, 9, *, A, or C as the second digit of the command prefix; and as a superuser priority, select a 2, 4, 6, 8, 0, #, B, or D as the second digit.

To program a new prefix code into the controller, simply touchtone the following eight-digit sequence without unkeying: the four-digit "change command" code, and the two-digit "code number," and "new two-digit prefix." The software counts the number of digits entered, then checks the code number range. If no errors are detected, an "R" will be heard when unkeying, to confirm the change of prefix. Remember, the controller will clear the command buffer if you pause for more than three seconds between digits. If editing becomes necessary in the presence of other signals, just use the force feature "#" at the end of the eight-digit sequence.

All modifications to the power-up state of the controller are saved in the computer's RAM. Since the 8749H power-down feature wasn't usable in this design, I decided to provide battery backup power to the entire board. Everything needed for this is on the computer board, including the diode switch and the charging resistor (R3) for the battery pack. The computer board requires approximately 225 mA, so a pack of seven AA NiCd batteries will keep the board alive for close to three hours. If longer periods of backup are required, you can substitute a backup battery with greater capacity. To allow the charging circuit to function properly, the full charged terminal voltage of the battery must be at least 1 volt less than the power supply voltage of pin 25 on the card edge connector. If you notice that your computer doesn't retain programming changes after power loss, measure the voltage across R3 to see if charging current is flowing into the battery under normal conditions. For those owners who have a 6 volt battery pack lying around, a high efficiency regulator can be substituted for U5 (LM2940CT-5.0). With this regulator, the terminal voltage of the backup battery can be as low as 6.1 volts. Resistor R3 should be selected according to the battery backup scheme you use. If you power the computer board with a 13–14 volt supply, R3 should be 390 ohms (you can use either voltage regulator) when using a seven-cell AA NiCd pack; R3 should be 470–510 ohms if you use a six-cell AA NiCd pack (use the optional regulator). See the parts list for a good backup battery source.

Computer Board Operation

The heart of the computer board is the microprocessor U1. It controls the entire repeater controller. The inputs to the microprocessor are buffered by a 4050 IC (U6) and the outputs are buffered by 7407 open collector buffers (U7, U8, and U9). Pull-up resistors in a SIP resistor pack (U10) pull all the inputs to an idle state if the pins aren't connected. The board uses a 555

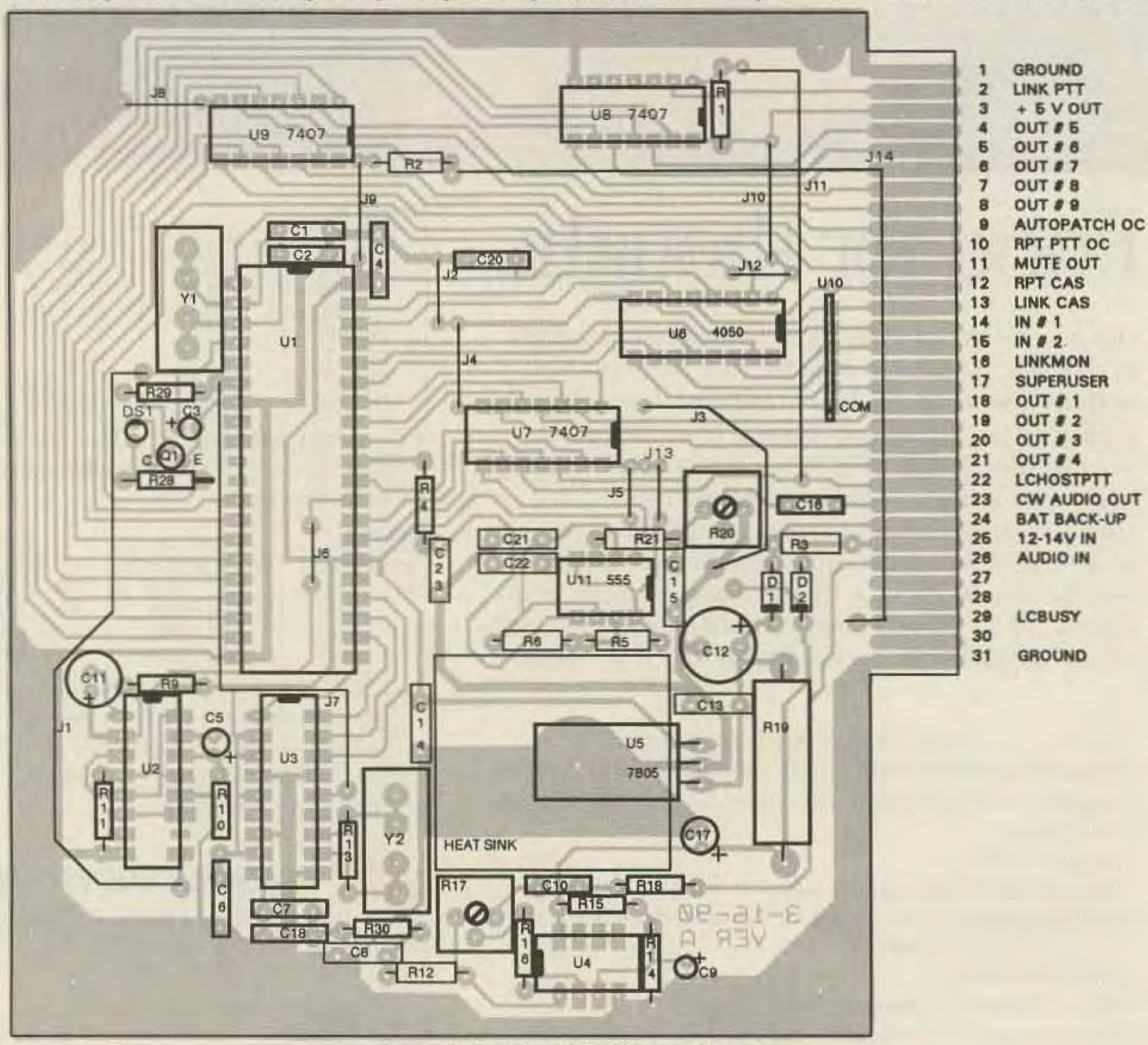

Figure 4. Computer board parts placement.

oscillator circuit (U11) to generate the CW tones for all IDs and messages. The resistors and capacitors in this circuit can be changed to modify the waveform and frequency. To change the pitch of the tone for some IDs and courtesy beeps, the microprocessor pulls pin 31 low. Resistor R21 and capacitor C15 form an RC filter for the tone oscillator, and pot R20 is used to adjust the CW ID level.

The computer board uses a DTMF decoder (U3) and an associated audio buffer (U4) to decode the DTMF tones. The gain of the circuit is controlled by pot R17. A filter after the op amp (R30, R12, C8, and C18) rolls off discriminator audio before passing it to the decoder chip. When the decoder chip thinks it detects a valid DTMF tone, pin 14 of U3 goes HIGH, signaling the microprocessor and lighting the LED DS1. Occasionally this LED may flash briefly during normal speech. If it occurs at a high rate, the filter after U4 may have to be modified.

The watchdog circuit (U2) monitors the microprocessor and resets the computer if the program stops execution. Pin 4 of U2 is continually pulsed by the microprocessor. If the program stops execution, these pulses will disappear, and after 8–15 seconds of delay, the watchdog should reset the microprocessor. When this happens, a power-up message will be sent on the repeater, and all default commands and parameters will be reloaded into memory.

It's important to connect a DC backup source to the battery backup pin to protect the microprocessor from being reset and loading the default parameters. Without a backup battery, power supply glitches may occasionally scramble both the microprocessor and the watchdog circuit. These cases are rare and they seem highly dependent on the transient suppression of the main DC supply.

The remaining circuits on the board provide regulated 5 volts to the board. If the supply voltage drops, the backup battery will provide power to the board through diode D1. Diodes D1 and D2 form a DC switch, and resistor R3 charges the external backup battery. If a non-chargeable battery is used, this resistor must be removed.

Table 3. Computer Board Parts List

Part	Description	Source
C1	5 pF ceramic capacitor	Mouser 21FK005
C2	20 pF ceramic capacitor	Mouser 21FL020
C3,5,9	1.0 µF tantalum	Mouser 540-1.0M35
C4,6,7,8,10,13, 14, 15,16,18,20,22	0.1 µF ceramic	Mouser 140-CD12R6-104Z
C11	47 µF, 6V tantalum	Mouser 540-47M06
C12	220 µF, 25V electrolytic	Mouser 140-XR35V220
C17	10 µF, 16V tantalum	Mouser 540-10M16
C21,23	0.01 µF ceramic	140-CD50Q6-103Z
D1,2	1N4001 diode	Mouser 333-1N4001
DS1	LED, any color	Mouser 35BL501
Q1	2N2222 NPN transistor	Mouser 511-2N2222
R1,2,4,5,18,21,29	10k, 1/4 W	Mouser 29SJ250-10k
R3	see text	see text
R6,11	100k, 1/4 W	29SJ250-100k
R9	680k, 1/4 W	Mouser 29SJ250-680k
R10	330k, 1/4 W	Mouser 29SJ250-330k
R12,14,15,30	2.2k, 1/4 W	Mouser 29SJ250-2.2k
R13	10MEG, 1/4 W	Mouser 29SJ250-10M
R16	1k, 1/4 W	Mouser 29SJ250-1k
R17,20	20k potentiometer, single turn	ME323-4255P-20k
R19	10 ohm, 1 or 2 W	Radio Shack 271-151
R28	470 ohm, 1/4 W	Mouser 29SJ250-470
U1	8749H microcontroller	WB3ESS; see note below
U2	4098 IC	Mouser 511-4098
U3	SSI202 Touchtone decoder IC	Radio Shack 276-1303
U4	LM358 IC	Mouser 511-LM358N
U5	7805 voltage regulator (see text)	Mouser 511-L7805ACV
U6	4050 IC	Mouser 511-4050
U7,8,9	7407 IC	526-NTE7407
U10	100k, 10-pin SIP	Mouser 266-100k
U11	555 timer IC	Mouser 511-NE555N
Y1,2	3.57 MHz crystal	Radio Shack 272-1310
1	PC board	WB3ESS RCCB3-16-90-A
1	40-pin IC socket	Mouser ME151-8040
2	15-pin IC sockets	Mouser 15IC016
1	18-pin IC socket	Mouser ME151-8018
3	14-pin IC sockets	Mouser 15IC014
2	8-pin IC socket	Mouser 15IC008
1	TO-220 heat sink	Radio Shack 276-1363
1	Card edge connector 31/62 (mounting holes)	Digi-Key S1312
	Alternate edge connector (no mounting holes)	Radio Shack 276-1453

Parts are available from: **Digi-Key Corporation**, 701 Brooks Ave. South, P.O. Box 677, Thief River Falls MN 56701-0677. Phone: (800) 344–4539; and **Mouser Electronics**, 12 Emery Avenue, Randolph NJ 07869. Phone: (800) 346–6873.

The computer and audio blank PC boards and a programmed 8749H microcontroller chip are available for $19 each from John Bednar WB3ESS, 548 Cherryville Road, Northampton PA 18067. When ordering the programmed microprocessor, please include the repeater call as you want it sent, including the prefix (de) and suffix (/rpt) along with all spaces clearly marked. SSI 202 touchtone decoder chips are available in limited supply for $7. Please add $4 shipping for all orders. Foreign orders should include additional postage.

If you want to program your own controller IC, the source code is available in DOS format from the author at the above address. Send $10 and a formatted floppy (any size, any density). If you write the author requesting information, please enclose an SASE.

For battery backup, an assembled 7-cell NiCd AA battery pack is available from Cunard Associates. Phone: (814) 623–7000.

Assembly and Test Instructions

Before you begin assembly, here are some pitfalls to watch out for:

1. Don't attempt to assemble this board with a high temperature soldering iron or gun.

2. If the LED DS1 is installed backwards, it will give you the impression that the touchtone decoder isn't working.

3. Make sure resistor pack U10 is positioned correctly, and of the correct type (one common pin and the resistors internally tied to this pin).

4. Be aware that not all the ICs are oriented in the same direction.

Begin assembly by installing the 14 jumpers. Don't miss jumper J6; it's located under U1, an IC. Follow this by adding all IC sockets. It's important to install a socket for U1 so that the chip can be removed without damage and reprogrammed if necessary. Next, install the voltage regulator and heat sink and put a little heat sink compound on the regulator tab to aid in the heat transfer. Bolt the regulator and heat sink firmly to the PC board. Finish the board assembly by adding all remaining components.

Perform the initial testing with the ICs removed. Apply 12–14 volts to the +12V IN and GROUND pins of the board, and measure the regulated +5 volts at pin 40 of U1. If the supply voltage isn't within 0.25 volts of +5 volts, measure the voltage drop across R19. If this voltage drop is greater than 4 volts, look for a shorted trace somewhere on the board. Once the voltages are correct, remove the power and insert all ICs. Reapply power and re-measure the supply voltage. With all ICs installed, it should still be within 0.25 volts of +5 volts.

The computer board sends a power-up ID whenever the computer is powered up or reset. This power-up ID can be used to check the initial operation of the computer board. Two test methods will be given; the first requires an oscilloscope and the second a voltmeter. Connect a scope probe to pin 23 of the board and apply power. While monitoring the CW AUDIO OUT pin, the CW power-up message should be visible on the scope. If no tone is observed, it's possible that the microprocessor isn't running, or the 555 tone circuit is nonfunctional. The second test method checks to see if the microprocessor is running. First connect a resistor (anything between 1k and 10k) between pins 3 and 10 of the computer board, and attach a voltmeter between pin 10 and ground. When the board is powered up, the voltage on pin 10 should drop to near zero volts. After 6–8 seconds, this voltage should rise to near 5 volts. If this doesn't happen, the microprocessor isn't running or U9 is faulty.

When the above tests are successful, the watchdog operation should be checked. Using either test configuration from above, place a 1k resistor across the crystal terminals Y1 when the computer is sending the power-up message. The resistor stops the microprocessor oscillator and crashes the program. Within 8–15 seconds, the watchdog circuit U2 should restart the microprocessor and the power-up message should be sent. If using the voltmeter technique, the voltage on pin 10 should go back to 5 volts 6–8 seconds after the computer is reset.

Next month in Part II we will discuss the audio board and operation of the whole controller. 73