
# BUILD THIS TELEPHONE CALL RESTRICTOR

***Restrict access to those expensive "900" and "976" numbers with this inexpensive call restrictor.***

**TERRY WEEDER**

OVER THE PAST FEW YEARS, THE "900," "976," and "540" telephone exchanges have become an easy way for people to spend a lot of money—too easy, perhaps. Some of those numbers, which provide access to a wide variety of services from sports scores to sex chat lines, can cost several dollars per minute. You won't even see the damages—a huge phone bill—until the end of the month. What makes it worse is that anyone who has access to your phone can call these numbers, forcing you to pay the bill. Most people would not leave their credit cards lying around the house for anyone to use, but in a sense, that is what you are doing if you allow others to have dialing access to all phone numbers from your phone.

The Telephone Call Restrictor described in this article can block access to any particular telephone number, or any group of numbers beginning with a certain prefix. The numbers to be blocked are entered into the call restrictor's memory from a Touch-Tone phone. The restrictor can also be programmed to block all telephone numbers *except* those which you have entered into memory. That feature is ideal for your business if you want to allow incoming calls, but only outgoing emergency calls.

The restrictor can be plugged into any phone jack in your home or office to control all the phones on that line. The restrictor can be disabled on a single-

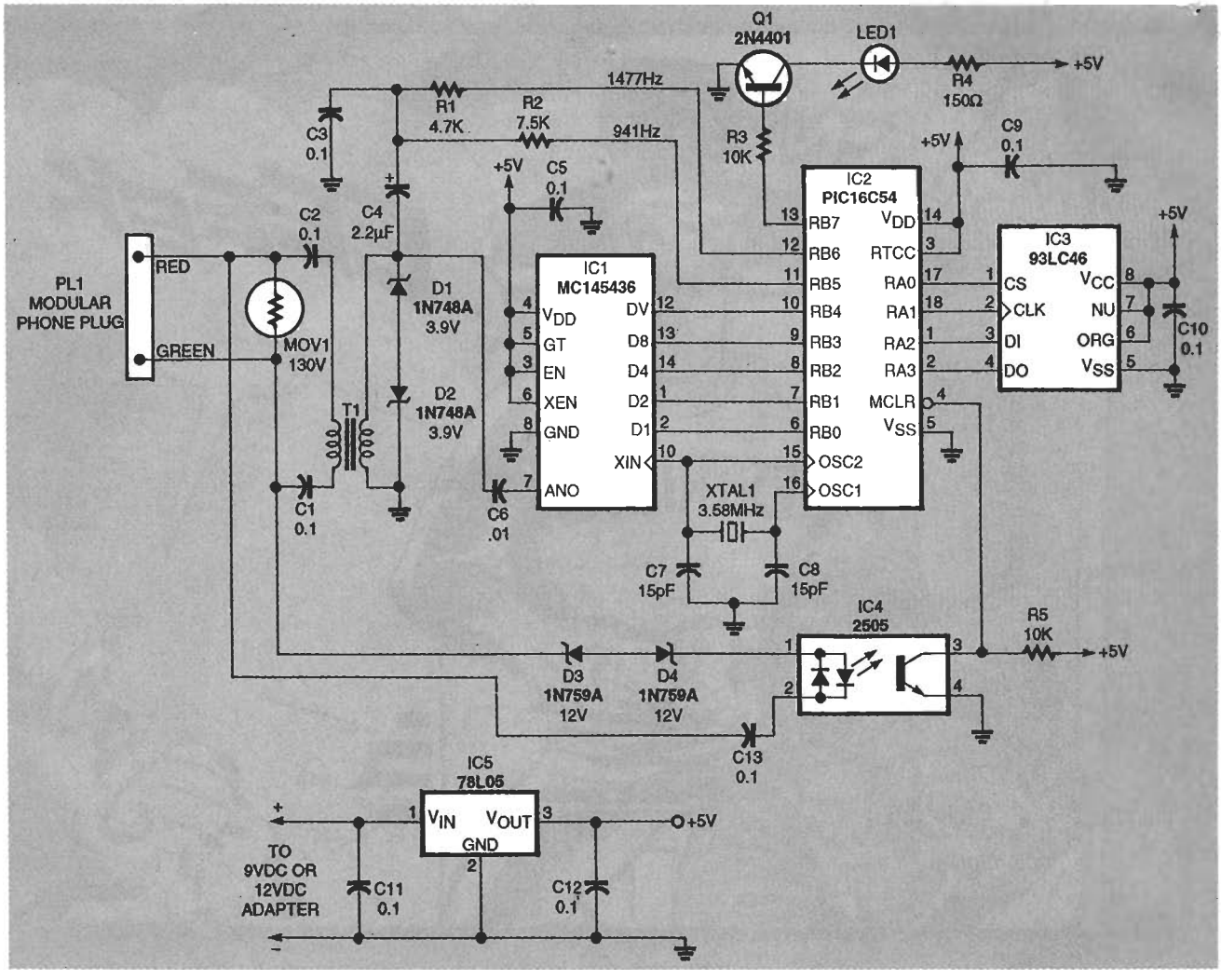



**FIG. 1—THE MICROCONTROLLER (IC2) COMPARES the decoded DTMF tones from IC1 with the telephone numbers stored in the EEPROM (IC3).**

call basis with a four-digit password. The password also prevents others from reprogramming the unit.

The restrictor is programmed from a Touch-Tone phone. An EEPROM provides non-volatile storage of the programmed information. Therefore, the restrictor will not lose its data because of a power failure or relocation of the unit.

## Circuit theory

A schematic diagram of the call restrictor is shown in Fig. 1. Capacitors C1 and C2 and transformer T1 isolate the circuit from the phone line. Those components also couple the DTMF (dual-tone multiple frequency) tones from the phone to pin 7 of IC1, an MC145436 DTMF receiver. A metal-oxide varistor (MOV1) protects the circuit from high-voltage spikes on the line, and D1 and D2 provide protection from ringer voltages. The DTMF receiver converts the Touch Tones into 4-bit TTL-level data. The PIC16C54 microcontroller (IC2) processes the data from IC1, stores the user-entered telephone numbers in EEPROM (IC3), and generates the tones necessary to disable the telephone.

Optoisolator IC4 generates a reset pulse for the microcontroller every time the handset is picked up or put back on hook. The change in voltage from tip to ring (when the line goes from on-hook to off-hook or vice versa) causes current to flow through the optoisolator as C13 charges or discharges to the new voltage level. The current flow forward biases the transistor output of IC4, which pulls IC2's reset pin low. Diodes D3 and D4 provide a threshold voltage that is high enough so that voice or DTMF tones will not trigger a reset.

A low-current voltage regulator (IC5) drops the voltage from a 9- or 12-volt DC adapter to 5-volts DC; C11 and C12 stabilize IC5's output.

The disable tones generated by IC2 (discussed later) are attenuated by R1 and R2, smoothed out by filter capacitor C3, and then coupled to T1 by capacitor C4. Transistor Q1 turns LED1 on when pin 13 of IC2 goes high.

## Microcontroller and EEPROM

The PIC16C54 is an 8-bit CMOS microcontroller manufactured by Microchip Technology, Inc. This microcontroller has one eight-bit I/O port, one four-bit I/O port, 512 × 12 bits of on-chip EPROM, and 32 × 8 bits of data RAM.

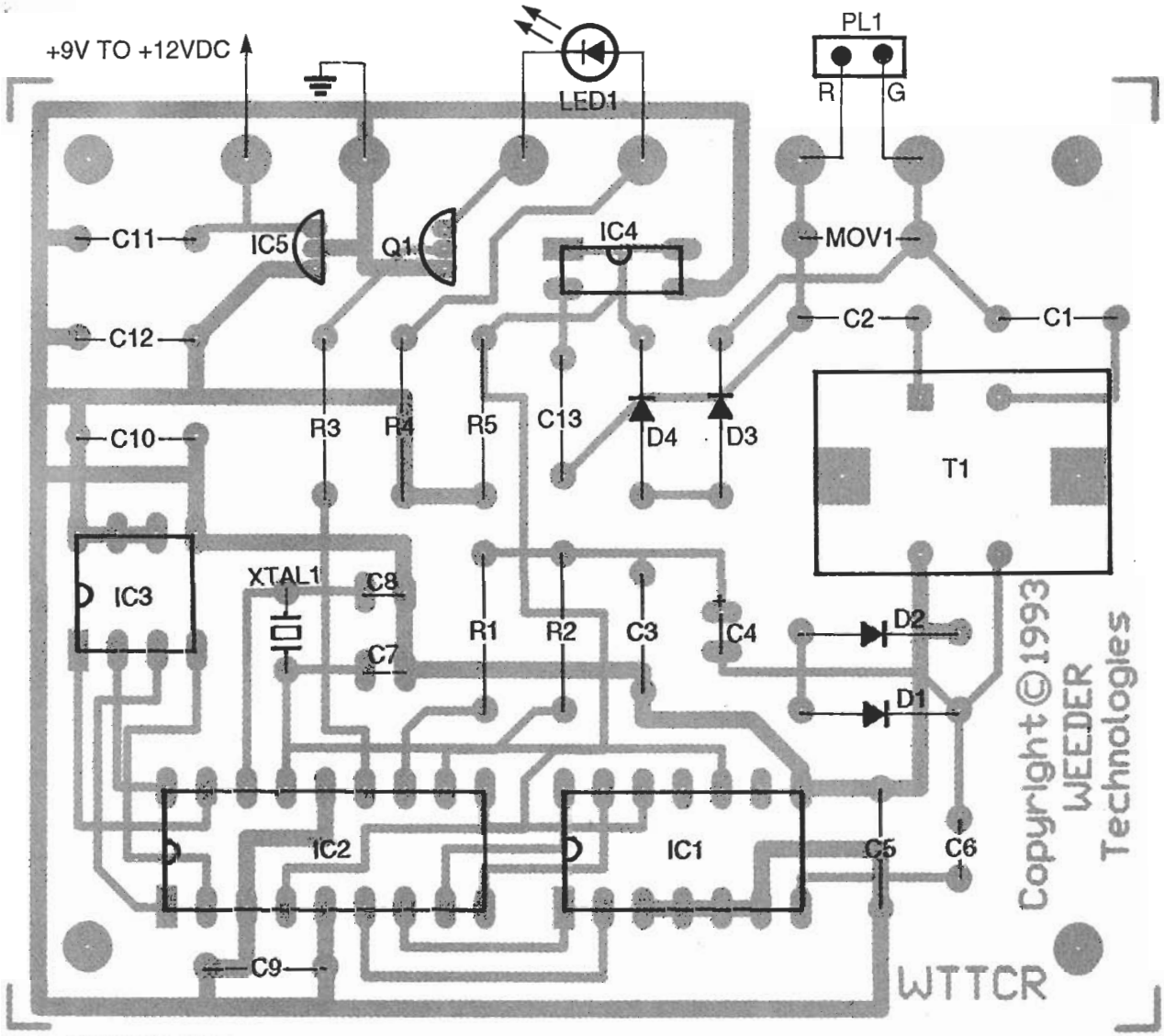



**FIG. 3—PARTS PLACEMENT DIAGRAM. Mount components as shown here. Watch for solder bridges on Q1 and IC5.**

The PIC16C54 contains a two-level stack, an eight-bit wide arithmetic logic unit (ALU), and a real-time clock/counter register with a prescaler. The instruction set consists of 33 single-word operations which require one cycle (four clock pulses) for execution. Instructions that force a program branch require two cycles. Each pin of the I/O ports can be configured individually as either an input or output through software. The PIC16C54 also has its own built-in "watch dog" timer (WDT) and "sleep" mode, but neither are used in this circuit because the chip is reset every time the handset is picked up. Also, the circuit is powered by an AC adapter so low-power operation is not essential.

A preprogrammed PIC16C54 is available from the source given in the Parts List. The source and object code files are available on the *Electronics Now BBS* (516-293-2283, V.32, V.42bis) for those who wish to program their own microcontrollers. A programmer for the microcontroller was described in the January 1994 issue of *Electronics Now.*

The internal RAM of the PIC16C54 functions as working registers for the operating program. All user-entered data is stored in IC3, a 93LC46, 1K serial EEPROM, also manufactured by Microchip. The 93LC46 is connected to Port A of IC2 through four input lines: CHIP SELECT, CLOCK, DATA IN, and DATA OUT. After a high is detected on CHIP SELECT, data is then transferred to and from the 93LC46 on the positive transition of the clock signal.

Each transfer of data consists of one start bit, a two-bit opcode that identifies the function to be performed, then a 6-bit address, followed by the 16 bits of data which is being read from or written to that address. Immediately preceding and following all write operations, the microcontroller sends instructions to the 93LC46 that enable or disable the write function, thereby protecting the data.

The data in the 93LC46 is stored in 16-bit blocks, while the data output from the DTMF receiver is in 4-bits. To make

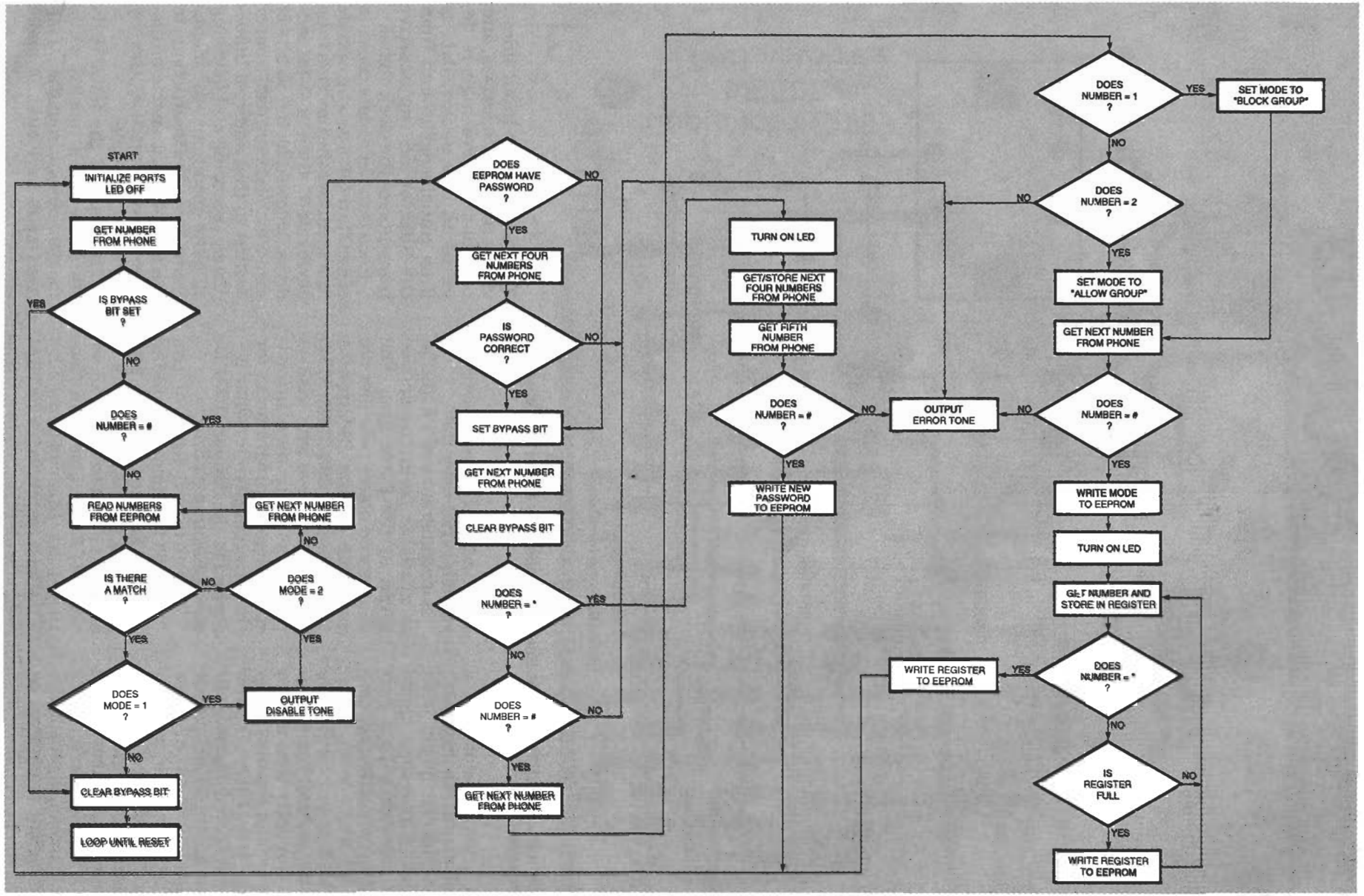
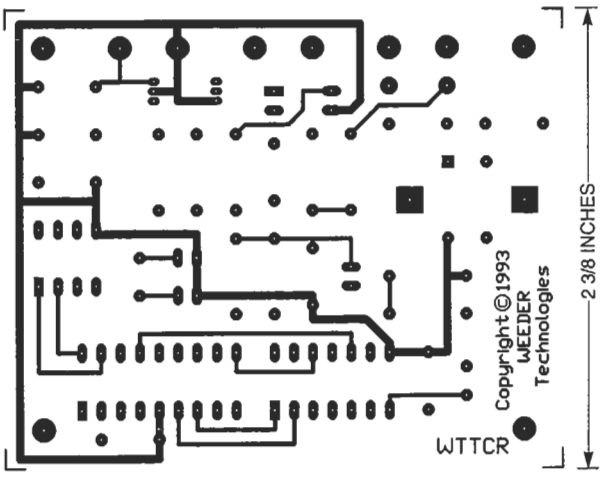



**FIG. 2—FLOW CHART OF THE OPERATING PROGRAM. The source and object code files are available on the *Electronics Now BBS* for programming your own microcontroller, or you can purchase a pre-programmed chip from the source given in the Parts List.**

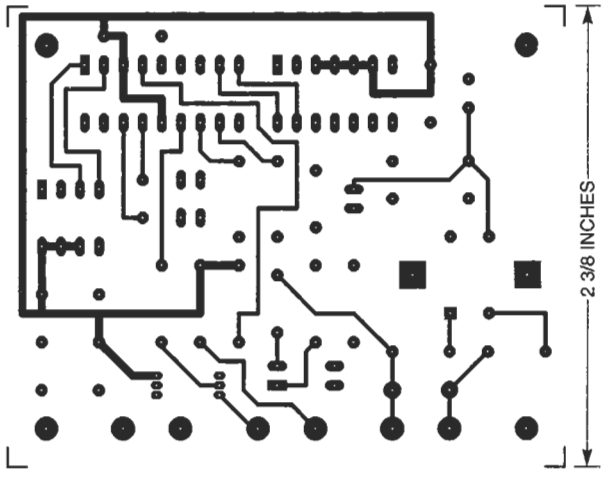
## PARTS LIST

**All resistors are ¼-watt, 10%, unless noted otherwise.**
R1—4700 ohms
R2—7500 ohms
R3, R5—10,000 ohms
R4—150 ohms
**Capacitors**
C1–C3, C5, C9–C13—0.1 μF, Mylar
C4—2.2 μF, 16 volts, tantalum
C6—0.01 μF, Mylar
C7, C8—15 pF, ceramic disc
**Semiconductors**
IC1—MC145436 DTMF receiver (Motorola)
IC2—PIC16C54-XT/P microcontroller (Microchip)
IC3—93LC46 serial EEPROM (Microchip)
IC4—2505-1 optoisolator (NEC or equivalent)
IC5—78L05 low-power 5-volt regulator
D1, D2—1N748A 3.9-volt Zener diode
D3, D4—1N759A 12-volt Zener diode
LED1—light-emitting diode, any color
Q1—2N4401 NPN transistor
**Other components**
MOV1—130 VRMS metal-oxide varistor
T1—600-ohm primary, 600-ohm secondary, audio transformer
XTAL1—3.58 MHz TV colorburst crystal
**Miscellaneous:** Enclosure, PC board, IC sockets, wall adapter (9- or 12-volt DC), telephone cord with modular plug, hook-up wire, solder, hardware
**Note: The following items are available from Weeder Technologies, P.O. Box 421, Batavia, Ohio 45103:**
- **Double-sided PC board (WT-TCR-B)—$9.50**
- **Kit of all board mounted components including pre-programmed PIC16C54 (WT-TCR-C)—$25.50**
- **Pre-programmed PIC16C54 only (PIC-TCR)—$16.00**

**All orders must include $3.50 for shipping and handling. U.S. and Canadian orders only. Ohio residents must add 6% sales tax.**




**COMPONENT SIDE of the call restrictor.**




**SOLDER SIDE of the call restrictor.**

use of all the memory available in the 93LC46, the microcontroller's software is written so that it will stack four numbers at the same address locations when storing the numbers in EEPROM, then extract the numbers in the correct order when reading back from the EEPROM.

### Software

A flow chart of the operating program is shown in Fig. 2. After IC2 resets (which occurs when the phone is taken off-hook), the chip looks for a number entered from the telephone. This microcontroller waits for a high on the DV (DATA VALID) pin of IC1, reads the 4 data lines, and then waits for a low on the DV pin. The microcontroller then checks to see if the pound (#) key is pressed. It indicates a request to program the EEPROM.

If a # is detected, IC2 looks for a password in EEPROM. Initially, a non-programmed 93LC46 EEPROM contains all 1's in its registers. Therefore, the microcontroller can determine if a password has been entered. If a password is found, IC2 reads the next four numbers from IC1 and determines if they match the password stored in EEPROM. If a valid password has been entered, the caller can then program the EEPROM. If an incorrect password is entered, or if a mistake is made in the programming sequence, an error tone is transmitted on the phone line. It continues until the phone is replaced on hook (causing a reset of IC2).

If the first number entered from the phone is not the # key (indicating that a call is being attempted), IC2 stores the number in one of its registers and then looks for a match in EEPROM. IC2 disables the telephone if there is a match, and then places the next number

entered in an adjacent register and looks for a match making use of both numbers stored in its registers. The process is repeated, adding each new number to the string of numbers stored in IC2's registers and comparing this string with numbers stored in EEPROM until either a disable condition is met or the string exceeds 11 characters.

If a disable condition is met, IC2 outputs two tones on port B; 1477 hertz is output on pin 12 and 941 hertz is output on pin 11. The two tones are the same as those generated by pressing the # key on the phone. Upon detection of this tone pair by the central office, a busy signal is automatically issued so that the call cannot be completed. (The same tone pair is also indicates an error as mentioned earlier.)

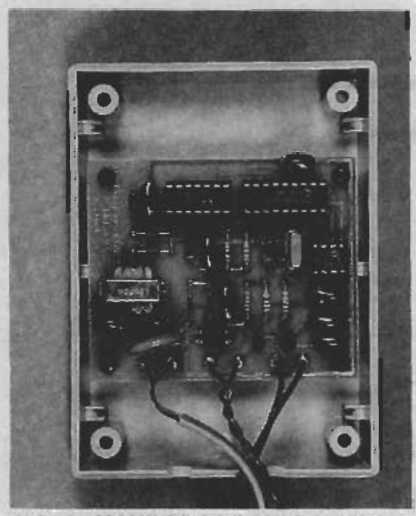


**FIG. 4—MOUNT THE BOARD in the enclosure and cut two slots in the seam of the plastic case for the power cord and the phone cord.**

In areas where the central office does not generate a busy signal when the # key is pressed, the tone generated by the restrictor will interfere with any conversation.

Notice from the flow chart that when programming the EEPROM, the microcontroller sets the bypass bit immediately after verifying that a correct password has been entered. It then clears that bit after receiving another number from the phone. If the caller enters the correct password and then hangs up the phone without entering any additional numbers, the bypass bit will be set. Therefore, when the handset is picked up to place the next call, the set bypass bit will be detected, and the program will branch to an endless loop that keeps the blocking action of the circuit disabled until a reset occurs. Placing the handset back on-hook resets the circuit.

**TABLE 1—PROGRAMMING**

**To Change / Enter Password:**
# – Password – * – New Password – #

**To Disable for Next Call:**
# – Password (hang up)

**To Program "Block Group" Mode:**
# – Password – # – 1 – #
Number – # –
Number – # –
Number – # –
" (up to 248 characters)
Number – # – *

**To Porgram "Allow Group" Mode:**
# – Password – # – 2 – #
Number – # –
Number – # –
Number – # –
" (up to 248 characters)
Number – # – *

Note: "Number" can be any telephone number (1 to 11 digits long).

**Construction**

The circuit fits on a double-sided, 2⅞- × 2⅜-inch printed circuit board. Artwork is provided here for those who wish to make their own boards. Manufactured boards can be purchased from the source given in the Parts List. Refer to the parts-placement diagram in Fig. 3 and start by inserting and soldering IC sockets for IC1 through IC3. Mount IC4 directly to the board and then solder Q1 and IC5 to the board, carefully avoiding solder bridges between the closely spaced pads.

Next, mount the resistors, capacitors and diodes. When soldering the crystal (XTAL1), leave a small space between the bottom of the crystal and the PC board. Caution: The metal case of the crystal could short the two solder pads together if it is pushed flush against the board when soldering. Finish assembly by mounting the transformer (T1) and the varistor (MOV1).

After all components have been soldered to the board, double check for solder bridges on both the top and bottom side of the board, and re-solder them if necessary. Carefully plug IC1, IC2, and IC3 into their sockets.

The board will mount directly in a plastic enclosure available from Digi-Key (Part No. SR131G-ND), but it is not essential that you use this case. Because this unit can be operated remotely with any Touch-Tone phone on the phone line, you might want to build the circuit without a case.

LED1 can be mounted to the top of the enclosure or soldered directly to the PC board. Use a phone cord with a modular jack on one end and solder the red and green wires to the correct locations on the board—you can cut off the black and yellow wires. After determining their polarity, solder the AC adapter's leads to the points labeled POS and NEG.

Mount the board in the enclosure and cut two slots in the seam of the plastic case for the power cord and the phone cord. Figure 4 shows the inside of the

*continued on page 74*