PIC MICROCONTROLLER-BASED ELECTRONIC LOCK

■ ANSHUMAN BEZBORAH

EFY TESTED S.V.N. HARISH K.

An electronic lock allows activation of an electric appliance only on entering the correct password. Here we present such an electronic locking system in which a PIC16F877A microcontroller plays the role of the processing unit. The MCU is interfaced with a 4×4 matrix keypad and a 16×2 LCD to form the user interface. Using this circuit, you can make any electrical appliance password-protected. It can also be used as an electronic door lock by interfacing the output of the circuit with an electrically actuated door lock. The system turns on the appliance on entering a four-digit password set by the user.

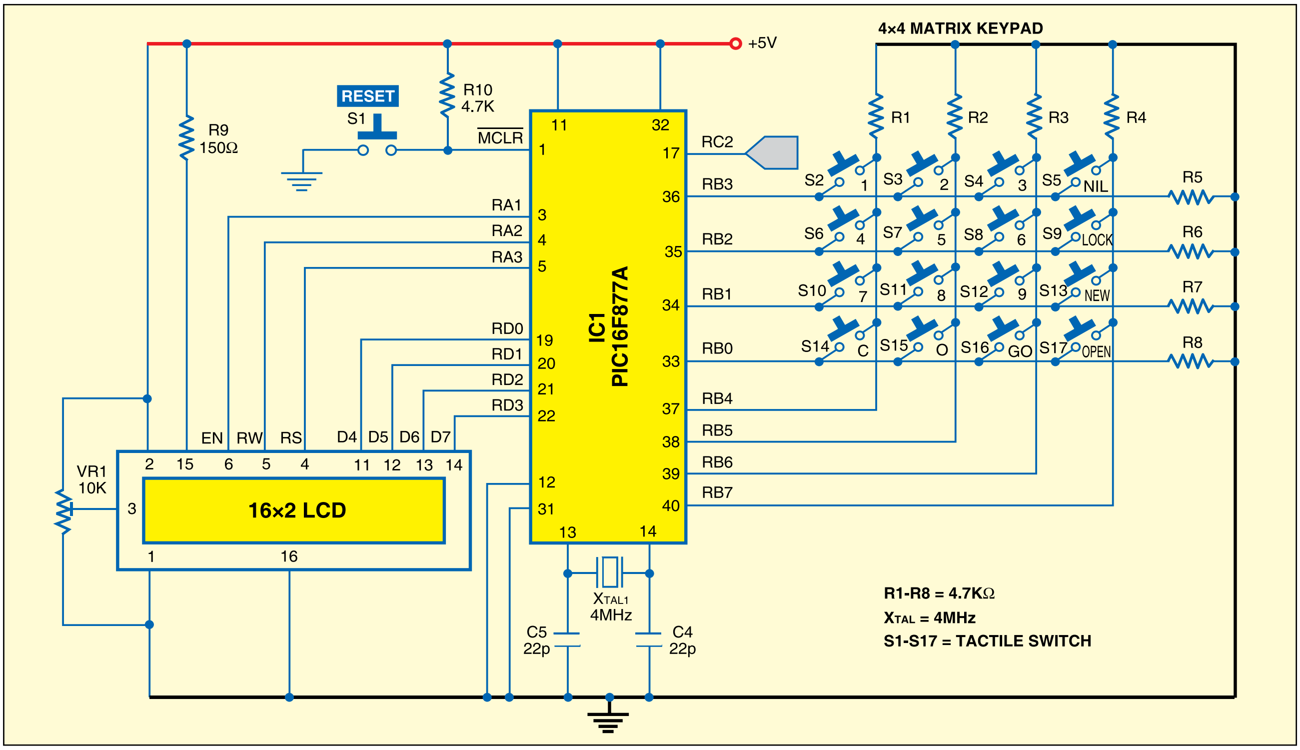

Fig. 1: Circuit of PIC microcontroller-based electronic lock

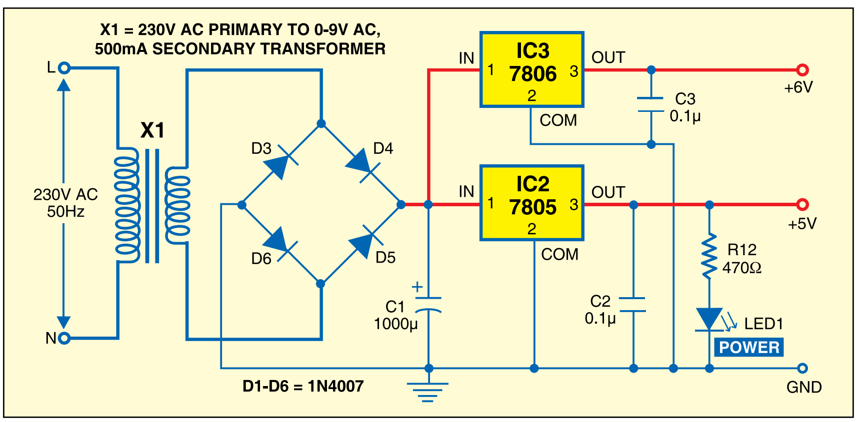

Fig. 2: Power supply circuit

Circuit description

Fig. 1 shows the circuit of the PIC microcontroller-based electronic lock. It can be divided into five sections: input (4×4 matrix keypad), processing unit (PIC16F877A MCU), appliance controller (relay driver), display (16×2 LCD), and power supply.

PIC16F877A MCU. The PIC16F877A is an 8-bit microcontroller based on reduced instruction set computer (RISC) architecture. It has 8k×14-

PARTS LIST

Semiconductors:
IC1 - PIC16F877A microcontroller
IC2 - 7805 voltage regulator
IC3 - 7806 voltage regulator
LED1 - 5mm Light-emitting diode
D1 - 1N4148 diode
D2-D6 - 1N4007 diode
T1 - SL100 transistor

Resistors (all ¼-watt, ±5% carbon unless stated otherwise):
R1-R8, R10 - 4.7-kilo-ohm
R9 - 150-ohm
R11 - 10-kilo-ohm
R12 - 470-ohm
VR1 - 10-kilo-ohm preset

Capacitors:
C1 - 1000µF, 25V electrolytic
C2, C3 - 0.1µF ceramic
C4, C5 - 22pF ceramic

Miscellaneous:
X_{TAL} - 4MHz crystal oscillator
X1 - 230V AC primary to 0-9V, 500mA secondary transformer
- HD44780-based 16×2 LCD
RL1 - 6V, 1C/O relay
S1-S17 - Push-to-on tactile switch

Functions of Various Keys of the Keypad and Their Labeling

S.No.	Row	Column	Label	Operation/digit entry
1	1	1	1	Digit '1'
2	1	2	2	Digit '2'
3	1	3	3	Digit '3'
4	1	4	Nil	No operation allotted
5	2	1	4	Digit '4'
6	2	2	5	Digit '5'
7	2	3	6	Digit '6'
8	2	4	Lock	Lock or turn off the appliance
9	3	1	7	Digit '7'
10	3	2	8	Digit '8'
11	3	3	9	Digit '9'
12	3	4	New	Change the password
13	4	1	C	Clear or backspace
14	4	2	0	Digit '0'
15	4	3	Go	Should be pressed after entering the password
16	4	4	Open	Open the lock (asks for password when pressed)

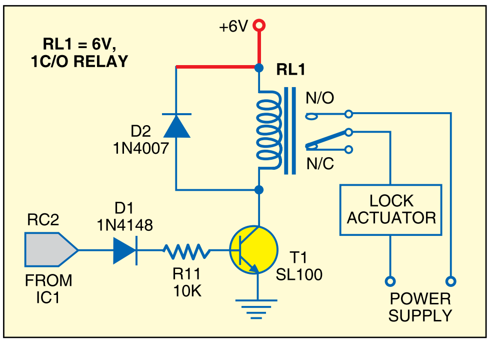

Fig. 3: Relay driver circuit

bit flash program memory, 368 bytes of RAM and many other internal peripherals like analogue-to-digital converter, USART, timers, synchronous serial port, compare captures and pulse-width modulation modules, EEPROM and analogue comparators.

The job of the MCU in this project is to receive signals from the input device (keypad) and take corresponding actions. Whenever any key is pressed on the keypad, the software program in the MCU identifies the pressed key and accordingly turns on or turns off the appliance. Simultaneously, it also displays a message on the LCD screen.

4×4 matrix keypad. A 4×4 matrix keypad is used to give commands and the password to the MCU. It consists of 16 keys (S2-S17) arranged in the form of a square matrix of four rows and four columns. Each key in the matrix is labeled according to the operation assigned to it. The connections from the pin-outs of the keypad to the MCU pins are shown in Fig. 1. Rows 1 through 4 are connected to pins RB3, RB2, RB1 and RB0 of Port B of the MCU, respectively. Columns 1 through 4 are connected to pins RB4 through RB7 of Port B, respectively.

16×2 LCD. A Hitachi HD44780 16×2 LCD is used to display various messages. It also displays an asterisk mark (*) for each digit of the password entered. Control lines EN, RW and RS of the LCD module are connected to pins RA1, RA2 and RA3 of Port A of the MCU, respectively. Commands and the data to be displayed are sent to the LCD module in nibble mode from Port D of the MCU. The higher four data bits of the LCD (D4 through D7) are connected to the lower nibble of Port D (RD0 through RD3) of the MCU.

Relay driver. RC2 pin of Port C of the MCU is interfaced with the relay driver circuit (shown in Fig. 3) to switch on or switch off the AC load (appliance). A relay driver circuit is nothing but a simple electronic circuit that drives an electromechanical relay. In this project, a 6V, single-changeover relay is used for switching the appliance 'on' or 'off.' Transistor SL100 plays the role of the relay driver.

Whenever the user enters the correct password, RC2 pin goes high (RC2=1). Consequently, transistor SL100 is triggered to energise the relay and the appliance turns 'on.' When RC2 is low (RC2=0), the appliance turns 'off.' Free-wheeling diode 1N4007 protects the relay driver circuit from the reverse voltage developed in the relay coil.

You can also use optocoupler MCT2E to isolate the relay driver circuit from the microcontroller circuit. Whenever the user enters the correct password, RC2 pin goes high (RC2=1) and the internal LED of the MCT2E IC glows, which, in turn, triggers the internal transistor of MCT2E.

Power supply. Fig. 2 shows the power supply circuit. The 230V AC mains supply is stepped down to 9V AC using step-down transformer X1. The output from the secondary of the transformer is rectified by a bridge rectifier comprising diodes D3 through D6 and filtered by capacitor C1. The filtered output is regulated by ICs 7805 and 7806 connected in parallel to obtain the required 5V and 6V, respectively.

Software

The software code is written in 'C' language and compiled using Hitech C

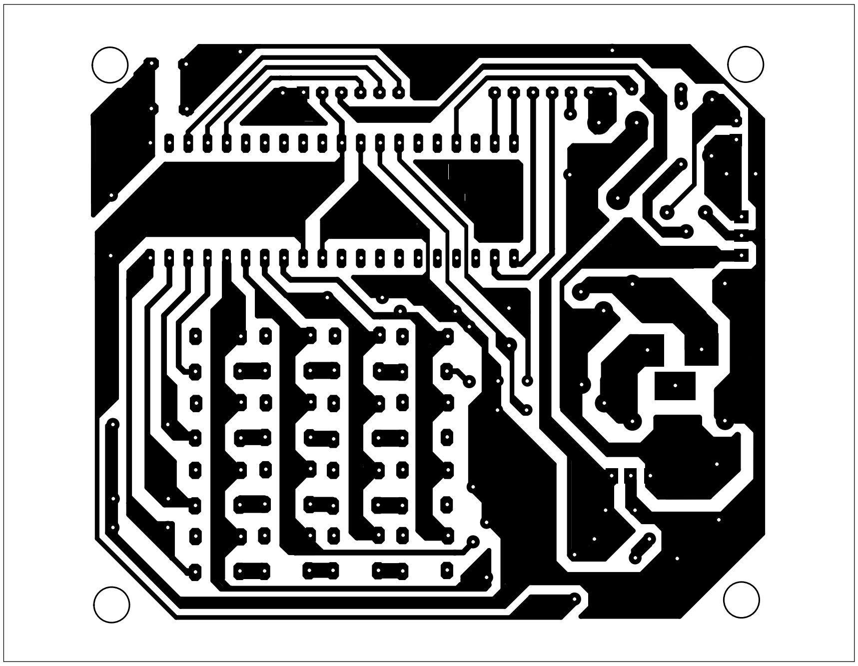

Fig. 4: Actual-size, single-side PCB for the PIC microcontroller-based electronic lock

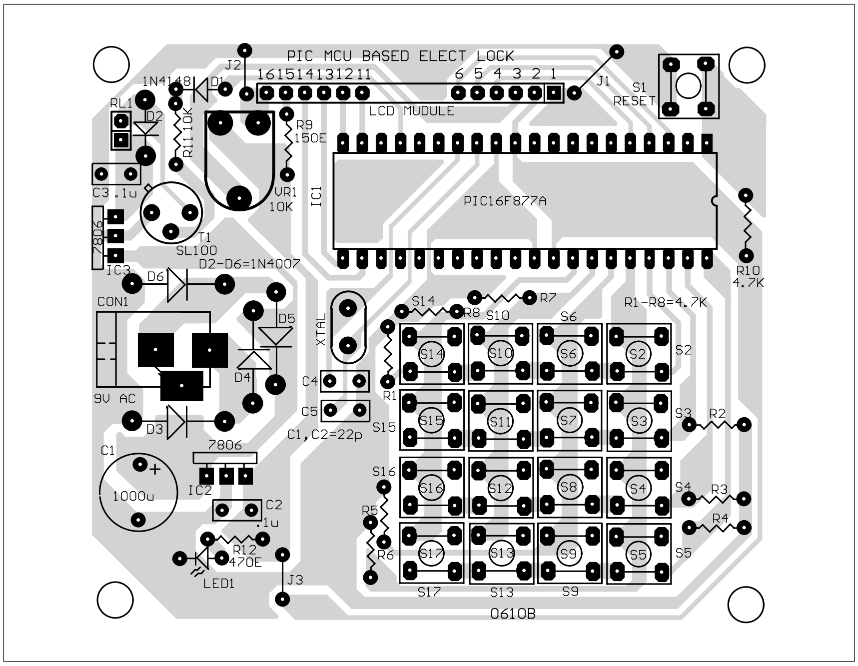

Fig. 5: Component layout for the PCB

compiler in MPLAB IDE. MPLAB IDE is a very powerful software development tool for Microchip's MCUs. It can be downloaded from www.microchip.com free of cost. It consists of tools like text editor, assembler, cross compliers and simulator. Hitech C compiler is meant for Microchip's PIC10/12/16 series of MCUs. Its Lite edition comes for free with newer versions of MPLAB IDE like MPLAB v8.2 or v8.3, and it can also be downloaded for free from www.htsoft.com.

The tasks performed by the software are:

1. Identify the key

2. Take the action allotted to the identified key

The key identification is done by identifying the row and the column to which the key belongs. Fig. 1 shows how the keypad is connected to Port B of the MCU. The lower nibble of Port B is declared as output pins (scan lines) and the upper nibble is declared as input pins (return lines). The number 0Fh is written to Port B so that the lower four bits become high and the upper four bits become low.

Whenever a key is pressed, the upper nibble pin (return line) of Port B, to which the column containing the key is connected, goes high. Thus the column is identified. Column identification is done using a switch-case block in the main program. On identifying the column, the rowfind(int) function is called, which does the job of row identification. To identify the row, scan lines are made low one by one in sequence and status of the return line corresponding to the key is checked. If it becomes low, the key belongs to that scan line or row. The row and column numbers are stored in two global variables 'row' and 'col.' A key debouncing delay of 20 ms is provided in the program by calling the DelayMs(20) function.

After identifying the key, the action( ) function is called in the main program, to perform the action corresponding to the identified key.

The detailed procedure for developing the project using MPLAB IDE, compiling the same using Hitech C compiler and burning the executable hex file to the microcontroller was explained in 'Construction' section of EFY's May issue.

The above description is available in a file named 'lock.c.' Functions lcd_init( ), lcd_goto(int), lcd_clear( ) and lcd_putch(char) are defined in a file named 'lcd.c' and the DelayMs(int) function is defined in the delay.c file. Add all the three 'C' files—lock.c, lcd.c and delay.c—as source files to the MPLAB IDE project. Save the project file as 'Elock.mcp.' Set configuration bits properly before building the project. Select the oscillator as XT and disable all other features like watchdog timer, power-up timer and brownout detection. After successfully building the project, the Elock.hex file is generated. Burn it into the chip using a suitable programmer, e.g., MPLAB ICD2.

Testing

Once the program is burnt into the chip and the hardware setup is ready, the user can test the system. When the power supply is switched on, message "Welcome" is displayed on the LCD screen. The default password set in the program is 1234.

To turn on the appliance, press 'Open' key. The system will ask for the password. Enter the password as 1234 and press 'Go.' The appliance should turn on (RC2=1) and the message "Password Accepted" should be displayed for two seconds followed by the message "Lock Open."

To turn off the appliance, press 'Lock.' The appliance should immediately turn off (RC2=0) and the message "Lock Closed" should be displayed on the LCD screen.

To set a new password, press 'New' key. The system should ask for the current and new passwords. Press 'Go' each time after you enter the four-digit password. The message "Password Saved" should appear for two seconds, followed by the message "Welcome." Now you can turn on the device by pressing 'Open' and then entering the new password that you have set. Key 'C' acts like 'Backspace' key in a PC's keyboard.

The only disadvantage of this system is that the password set by the user is stored in the RAM and hence it is lost when the supply is switched off. When the supply is switched on again, the system is reset with the default password 1234.

Construction

A single-side, solder-side PCB layout for the PIC microcontroller-based electronic lock is shown in Fig. 4 and its component layout in Fig. 5.

Connect the 4×4 matrix keypad and 16×2 LCD to PIC16F877A microcontroller as shown in Fig. 1. Complete the remaining connections also as shown in Fig. 1. Connect the appliance to be controlled to RC2 (pin 17) of Port C through the relay-driver circuit as shown in Fig. 3. Connect 6V power supply from the power supply circuit to the relay driver circuit. The 5V supply required by the microcontroller is obtained from the 7805 regulator output.

EFY note. The complete project folder (named Elock.zip) containing the source code and related files is included in this month's EFY-CD and also available on www.efymag.com website. ●