

INDIAN INSTITUTE OF TECHNOLOGY PATNA

EC3101: MICROCONTROLLER AND
EMBEDDED SYSTEM LAB



Experiment No: 02

Submitted by :

Name: Shaurya Aggarwal

Roll No: 2301EE56

Group No: 02(Thursday)

Part1: Implementation of 7-Segment LED Display using PIC16F84A Microcontroller

Aim:

To interface a 7-segment display with the PIC16F84A microcontroller and program it to display decimal digits from 0 to 9 sequentially..

Apparatus :

1. MikroC Pro for PIC → for writing and compiling the embedded C code.
2. Proteus Design Suite → for circuit design and simulation.

THEORY :

A 7-segment display is an electronic display device used to represent decimal numbers. It consists of seven LEDs (segments a–g) arranged in a figure-8 pattern. By turning ON specific segments, digits from 0 to 9 can be displayed.

There are two types of 7-segment displays:

Common Anode and Common Cathode. In this experiment, we assume a common cathode display, where logic HIGH at a segment pin turns it ON. The PIC16F84A controls the display through PORTB. By writing predefined binary values to PORTB, the microcontroller turns ON the correct combination of LEDs to form each digit. For example:

- 0b00111111 → displays digit 0 (segments a, b, c, d, e, f ON).
- 0b00000110 → displays digit 1 (segments b, c ON).

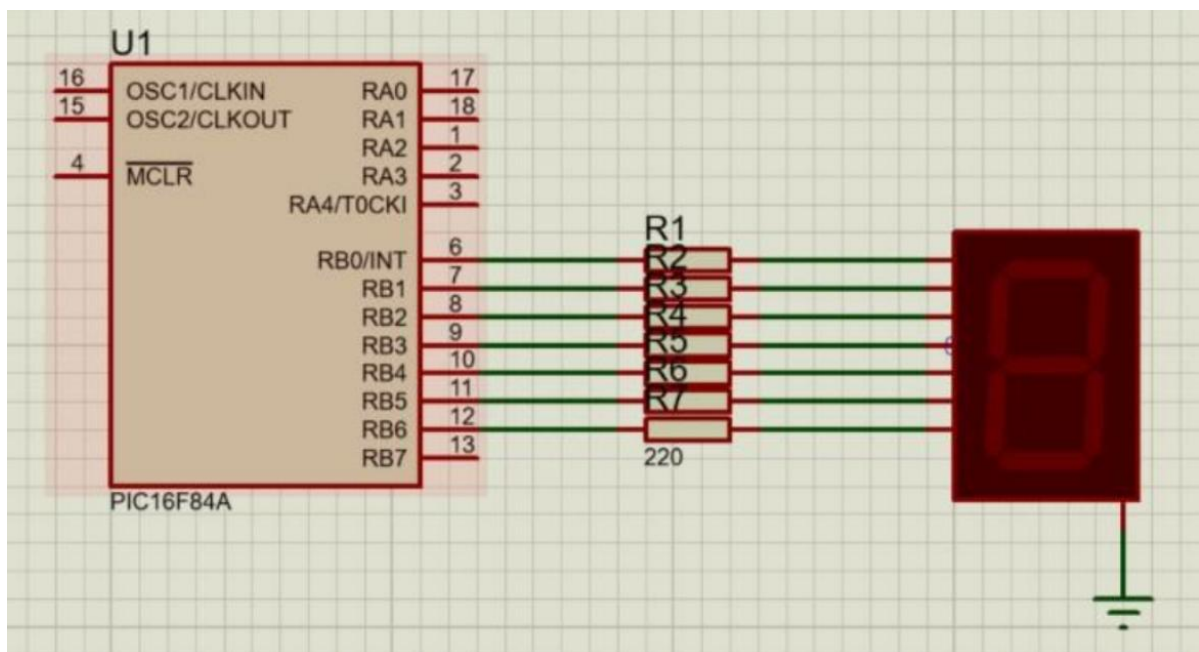
- up to 0b01101111 → displays digit 9.

A delay of 250ms between updates allows each digit to be visible before changing to the next. The process repeats in a loop, cycling through all digits 0–9. Applications include digital *clocks, counters, calculators, and display panels.

Working:

The 7-segment display is connected to PORTB of PIC16F84A.

- Each segment (a–g) corresponds to one PORTB pin.
- Binary patterns are written to PORTB to glow the correct combination of segments for each digit.



Code B:

```
void main() {  
    TRISB = 0x00; // Configure PORTB as output  
    while(1) {  
        PORTB = 0b00111111; // Display 0  
        Delay_ms(250);  
        PORTB = 0b00000110; // Display 1  
        Delay_ms(250);  
        PORTB = 0b01011011; // Display 2  
        Delay_ms(250);  
        PORTB = 0b01001111; // Display 3  
        Delay_ms(250);  
        PORTB = 0b01100110; // Display 4  
        Delay_ms(250);  
        PORTB = 0b01101101; // Display 5  
        Delay_ms(250);  
        PORTB = 0b01111101; // Display 6  
  
        Delay_ms(250);  
        PORTB = 0b00000111; // Display 7  
        Delay_ms(250);  
        PORTB = 0b01111111; // Display 8  
        Delay_ms(250);  
        PORTB = 0b01101111; // Display 9  
        Delay_ms(250);  
    }  
}
```

Observation:

The 7-segment display showed digits from 0 to 9 sequentially with a 250ms delay. • After displaying 9, the sequence restarted from 0.

- The simulation confirmed the correct working of segment-to-pin mapping.

Results:

The 7-segment display was successfully interfaced with PIC16F84A. All digits from 0 to 9 were displayed sequentially in the Proteus simulation, verifying correct code execution and hardware mapping.

Part2: Displaying a Name on 16×2 LCD using PIC16F877A

microcontroller

Aim:

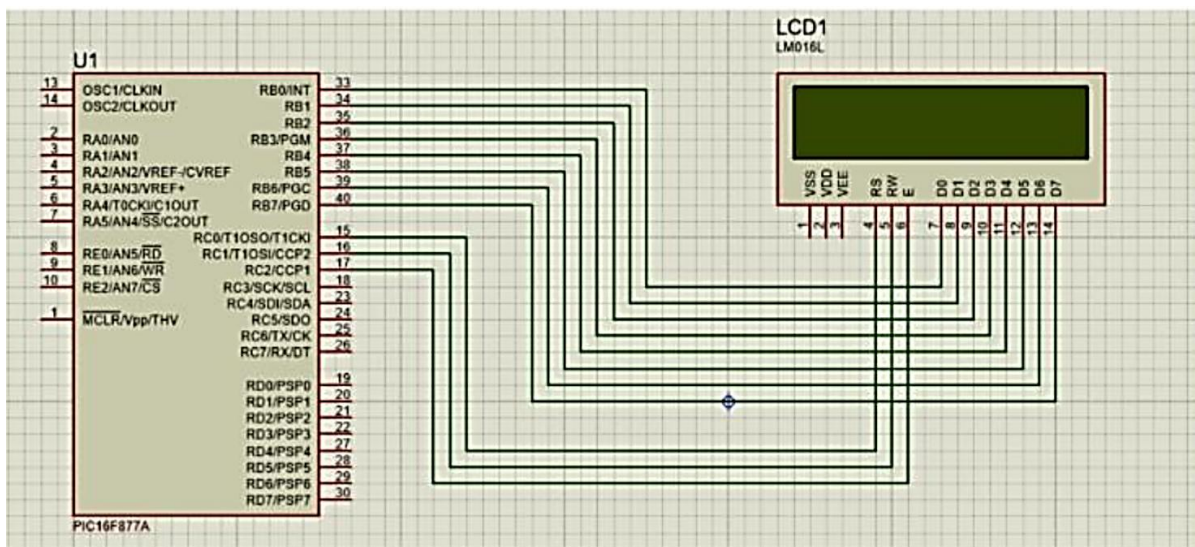
To interface a 16×2 alphanumeric LCD with PIC16F877A microcontroller and display a name with continuous left-shifting using MikroC PRO for PIC and simulated in Proteus.

Software Used:

MikroC Pro for PIC → for writing and compiling the embedded code.

Proteus Design Suite → for circuit design and simulation.

- The LCD data pins (D0–D7) are connected to PORTB of the PIC.
- The control pins RS, RW, EN are connected to RC0, RC1, RC2 respectively.
- 16×2 LCD operates in 8-bit mode.



Theory:

A Liquid Crystal Display (LCD) is a widely used output device in embedded systems for showing alphanumeric characters, symbols, or simple graphics. The 16×2 LCD has two rows, each capable of displaying 16 characters. Each character is generated from a 5×8 dot matrix inside the LCD controller (HD44780 standard).

The PIC16F877A communicates with the LCD through its data lines (D0–D7) and control lines:

- RS (Register Select): Selects command register (RS=0) or data register(RS=1).
- RW (Read/Write): Chooses between write (RW=0) and read (RW=1).
- EN (Enable): Latches data on the high-to-low transition.

In this experiment, PORTB of PIC16F877A is configured as output to send data/commands to the LCD, while RC0, RC1, RC2 control RS, RW, and EN respectively. Initialization commands set the LCD in 8-bit, 2-line, 5×8 font mode, turn ON the display, and set entry mode. The program displays the string “CHITRA IYER” and continuously shifts the text left using the command 0x18. This creates a scrolling display effect. Applications of such interfacing include message displays, notice boards, industrial control panels, and embedded GUIs.

CODE:

```
// Define LCD Control Pins on PORTC
#define rs PORTC.F0
#define rw PORTC.F1
#define en PORTC.F2

unsigned int j;
unsigned int i;

void delay_custom();
void lcd_init();
void cmd(unsigned char a);
void dat(unsigned char b);
void show(unsigned char *s);

void main() {

    // Set directions
    TRISB = 0x00;    // PORTB as output (LCD data)
    TRISC.F0 = 0;    // RS
    TRISC.F1 = 0;    // RW
    TRISC.F2 = 0;    // EN

    lcd_init();
    cmd(0x8A); // Force cursor to 0x8A
    show("SHAURYA AND ABHAY");
```

```

while(1) {
    Delay_ms(250); // Delay approx 250ms
    cmd(0x18);    // Shift display left
    Delay_ms(250);
}
}

void delay_custom() {
    for(j = 0; j < 1000; j++);
}

void lcd_init() {
    cmd(0x38); // Function set: 8-bit, 2 line, 5x8 font
    cmd(0x0C); // Display ON, cursor OFF
    cmd(0x06); // Entry mode set: increment
    cmd(0x80); // Set DDRAM to 0x00 (home position)
}

void cmd(unsigned char a) {
    PORTB = a;
    rs = 0;
    rw = 0;
    en = 1;
    delay_custom();
    en = 0;
}

void dat(unsigned char b) {
    PORTB = b;
    rs = 1;
    rw = 0;
    en = 1;
    delay_custom();
    en = 0;
}

void show(unsigned char *s) {
    while(*s) {
        dat(*s++);
    }
}

```


Observation:

- The LCD initializes properly and displays the text "SHAURYA AND ABHAY".
- The string scrolls continuously to the left across the LCD screen.

RESULT:

The name was successfully displayed on a 16×2 LCD using PIC16F877A, and a scrolling effect was achieved by repeatedly sending the shift-left command (0x18).