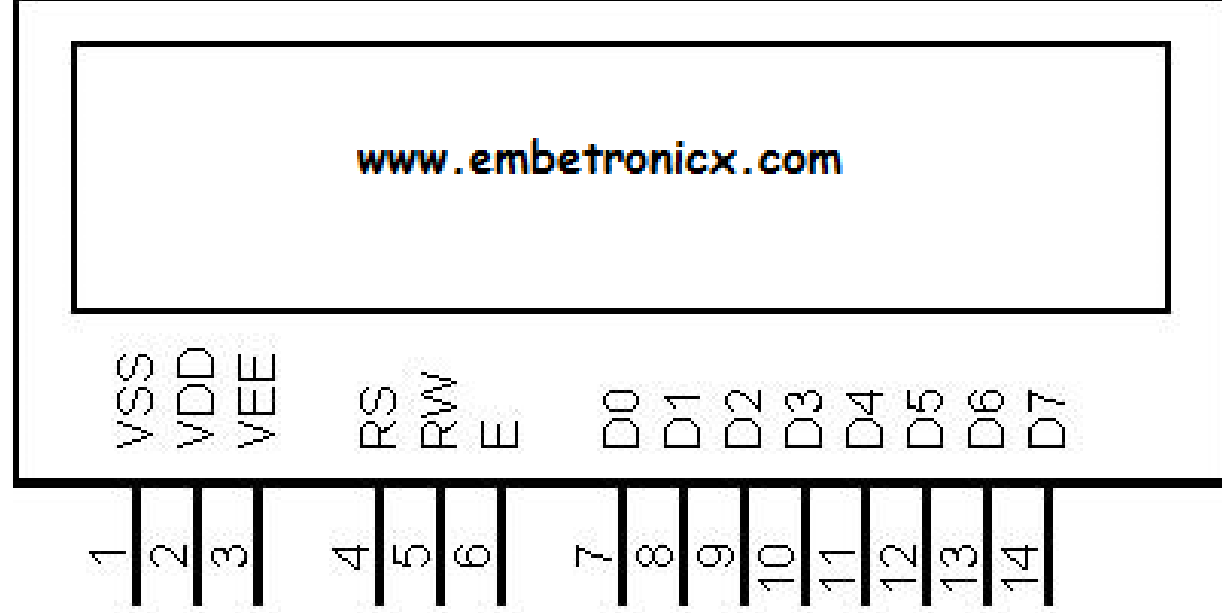


# Interfacing an LCD to the 8051 Microcontroller:

- Engr.Shahzada Fahim Jan

- LCDs are economical
- easily programmable
- have no limitation of displaying special & even custom characters (unlike in seven segments), animations, and so on.
- A 16×2 LCD means it can display 16 characters per line and there are 2 such lines
- In this LCD each character is displayed in a 5×7 pixel matrix.
- This LCD has two registers, namely, Command and Data.

- The command register stores the command instructions given to the LCD.
- A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling the display, etc.
- The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.



<b>Pin</b>	<b>Symbol</b>	<b>Input/output</b>	<b>Details</b>
1	Vss	-	GND
2	Vcc	-	+5V
3	Vee	-	Contrast Control
4	RS	Input	RS=0 Command Register RS=1 Data Register
5	R/W	Input	R/W=0 for Write R/W=1 for Read
6	EN	Input/output	Enable
7	DB0	Input/output	8 bit data bus
8	DB1	Input/output	8 bit data bus
9	DB2	Input/output	8 bit data bus
10	DB3	Input/output	8 bit data bus
11	DB4	Input/output	8 bit data bus
12	DB5	Input/output	8 bit data bus
13	DB6	Input/output	8 bit data bus
14	DB7	Input/output	8 bit data bus
15	LED+	-	Backlight of LCD to VCC
16	LED-	-	Backlight of LCD to GND

- **2) RS (Register Select):**

- 

- The RS pin is used to select command code register or data register. If RS=0 the command code register is selected which allows us to send the instructions to LCD. If RS=1 the data register is selected which allows us to send data to be displayed on LCD.

- **3) RW(Read/Write):**

- 

- R/W input allows the user to write information to the LCD or read info from it.
- R/W =1 when reading
- R/W =0 when writing

- **4) EN (Enable):**

- when data or command is sent to LCD a high-to-low pulse must applied to the PIN. So Enable pin is toggled by sending first 1 and then 0.

## LCD Command Codes

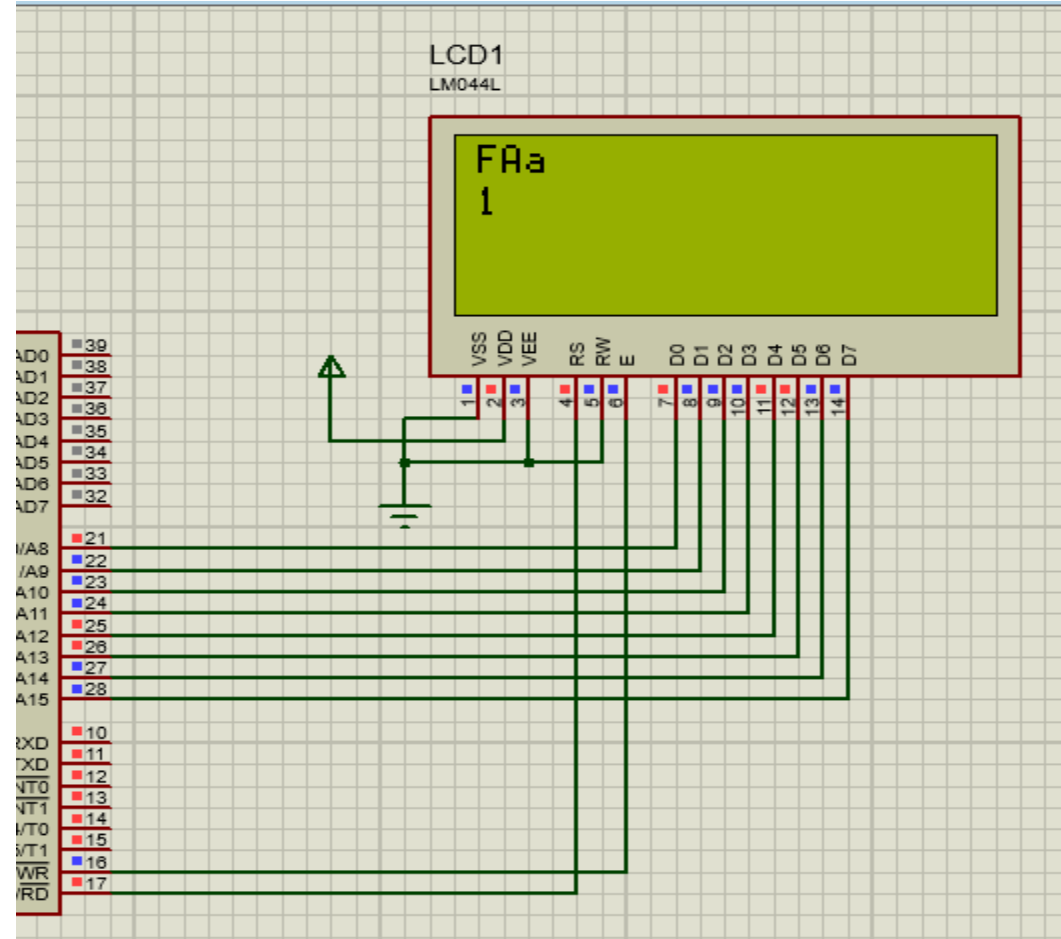
Code (Hex)	Command to LCD Instruction Register
1	Clear display screen
2	Return home
4	Decrement cursor (shift cursor to left)
6	Increment cursor (shift cursor to right)
5	Shift display right
7	Shift display left
8	Display off, cursor off
A	Display off, cursor on
C	Display on, cursor off
E	Display on, cursor blinking
F	Display on, cursor blinking
10	Shift cursor position to left
14	Shift cursor position to right
18	Shift the entire display to the left
1C	Shift the entire display to the right
80	Force cursor to beginning to 1st line
C0	Force cursor to beginning to 2nd line
38	2 lines and 5x7 matrix

```
void writedata(char t) {  
    RS = 1;    // This is data  
    P2 = t;    // Data transfer  
    E = 1;    // => E = 1  
    delay(100);  
    E = 0;    // => E = 0  
    delay(100);  
}  
  
void writecmd(int z) {  
    RS = 0;    // This is command  
    P2 = z;    // Data transfer  
    E = 1;    // => E = 1  
    delay(100);  
    E = 0;    // => E = 0  
    delay(100);  
}  
  
void lcdinit(void) {  
    delay(15000); // Wait for LCD to power up  
    writecmd(0x38); // Function set  
    writecmd(0x0C); // Display on, cursor off, blink off  
}
```



```
#include <reg51.h>
#include <string.h>
// Function declarations
void delay(int);
void lcdinit(void);
void writecmd(int);
void writedata(char);
sbit E  = P3^6; //E pin for LCD
sbit RS = P3^7; //RS pin for LCD
int main(void) {
    //lcdinit(); // Initialize LCD
    writecmd(0x0C);
    writedata('F');
    writedata('A');
    //writecmd(0x01); // clear display
    writedata('a');
    writecmd(0xC0);
    writecmd(0x38);
    writedata('1');
    while (1) {
    }
}

void delay(int a) {
    int i;
    for (i = 0; i < a; i++); // Null statement
}
```



# Reading Key

```
char READ_SWITCHES(void) {  
    RowA = 0; RowB = 1; RowC = 1; RowD = 1; // Test Row A  
  
    if (C1 == 0) { delay(10000); while (C1 == 0); return '7'; }  
    if (C2 == 0) { delay(10000); while (C2 == 0); return '8'; }  
    if (C3 == 0) { delay(10000); while (C3 == 0); return '9'; }  
  
    RowA = 1; RowB = 0; RowC = 1; RowD = 1; // Test Row B  
  
    if (C1 == 0) { delay(10000); while (C1 == 0); return '4'; }  
    if (C2 == 0) { delay(10000); while (C2 == 0); return '5'; }  
    if (C3 == 0) { delay(10000); while (C3 == 0); return '6'; }  
  
    RowA = 1; RowB = 1; RowC = 0; RowD = 1; // Test Row C  
  
    if (C1 == 0) { delay(10000); while (C1 == 0); return '1'; }  
    if (C2 == 0) { delay(10000); while (C2 == 0); return '2'; }  
    if (C3 == 0) { delay(10000); while (C3 == 0); return '3'; }  
  
    RowA = 1; RowB = 1; RowC = 1; RowD = 0; // Test Row D  
  
    if (C1 == 0) { delay(10000); while (C1 == 0); return 'C'; } // Clear screen  
  
    return 'n'; // Means no key has been pressed or an invalid key  
}
```

```
void writedata(char t) {  
    RS = 1;           // This is data  
    P2 = t;           // Data transfer  
    E = 1;            // => E = 1  
    delay(100);  
    E = 0;            // => E = 0  
    delay(100);       //  
}  
  
void writecmd(int z) {  
    RS = 0;           // This is command  
    P2 = z;           // Data transfer  
    E = 1;            // => E = 1  
    delay(100);  
    E = 0;            // => E = 0  
    delay(100);  
}
```

```

// Function declarations
void cct_init(void);
void delay(int);
void lcdinit(void);
void writecmd(int);
void writedata(char);
char READ_SWITCHES(void);
char get_key(void);
int get_num(char);
void disp_num(int);

```

```

void delay(int a) {
    int i;
    for (i = 0; i < a; i++);
}

```

```

int main(void) {
    char key;      // key char for keeping a record of pressed key
    int num = 0;   // Number entered by the user

    cct_init();    // Make input and output pins as required
    lcdinit();     // Initialize LCD

    while (1) {
        key = get_key(); // Get key from the user
        num = get_num(key); // Get the number from the key (if valid)

        if (num != Error) { // If a valid number is entered
            disp_num(num); // Display the number on the LCD
        }
    }

    void cct_init(void) {
        P0 = 0x00;
        P1 = 0xF0; // Used for generating outputs and taking inputs from Keypad
        P2 = 0x00; // Used as data port for LCD
        P3 = 0x00;
    }
}

```

```

] char get_key(void) {
    char key = 'n'; // Assume no key pressed

]   while (key == 'n') {
        key = READ_SWITCHES(); // Scan the keys until a key is pressed
    }

    return key; // Return the pressed key
}

] int get_num(char ch) {
]   switch (ch) {
        case '1': return 1; break;
        case '2': return 2; break;
        case '3': return 3; break;
        case '4': return 4; break;
        case '5': return 5; break;
        case '6': return 6; break;
        case '7': return 7; break;
        case '8': return 8; break;
        case '9': return 9; break;
        default: return Error; break; // Invalid input
    }
}

] void disp_num(int num) {
    writedata(num + '0'); // Display the number on the LCD
}

```

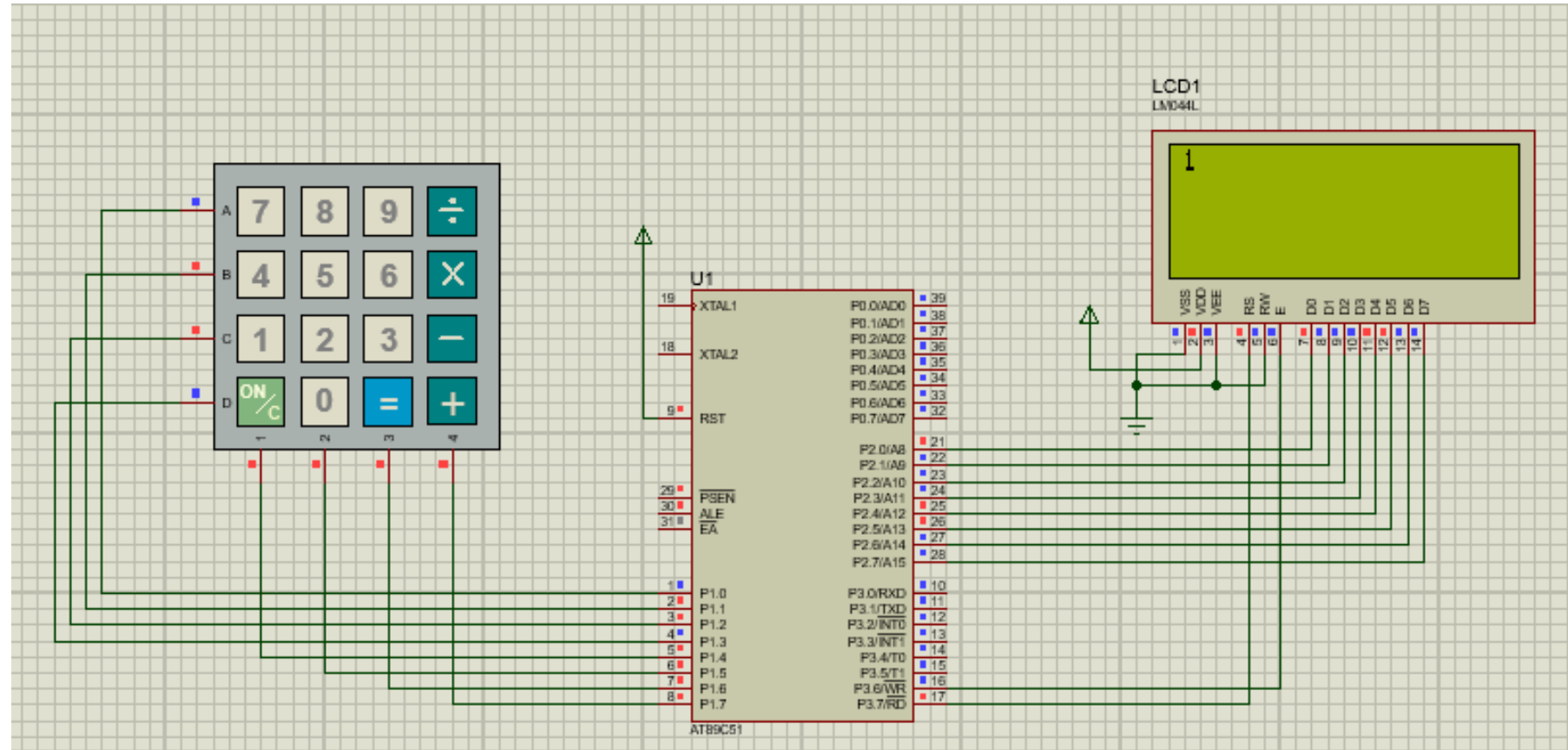
```

] void lcdinit(void) {
    delay(15000); // Wait for LCD to power up
    writecmd(0x38); // Function set
    writecmd(0x0C); // Display on, cursor off, blink off
    writecmd(0x01); // Clear display
    writecmd(0x06); // Entry mode set: increment cursor, no shift
}

```

# Clear display if C/On key is pressed

```
int main(void) {  
    char key;           // key char for keeping a record of pressed key  
    int num = 0;        // Number entered by the user  
  
    cct_init();         // Make input and output pins as required  
    lcdinit();         // Initialize LCD  
  
    while (1) {  
        key = get_key(); // Get key from the user  
        /* num = get_num(key); // Get the number from the key (if valid)  
  
        if (num != Error) { // If a valid number is entered  
            disp_num(num); // Display the number on the LCD  
  
        }*/  
        if (key == 'C') { // If C/On button is pressed  
            writecmd(0x01); // Clear display  
            delay(1000); // Delay for a second after clearing  
        } else {  
            num = get_num(key); // Get the number from the key (if valid)  
  
            if (num != Error) { // If a valid number is entered  
                disp_num(num); // Display the number on the LCD  
            }  
        }  
    }  
}
```





# TASKS:

- 1) Write Your Name on First Line and registration Number on second line of LCD.
- 2) Input numbers from keypad from 1 to 9 and display on the LCD
- 3) Make a calculator using keypad and LCD which perform (+,-,\*,/)

$$6 + 6 = 12$$

$$6 * 6 = 36$$

$$6 / 6 = 1$$

$$6 - 3 = 3$$