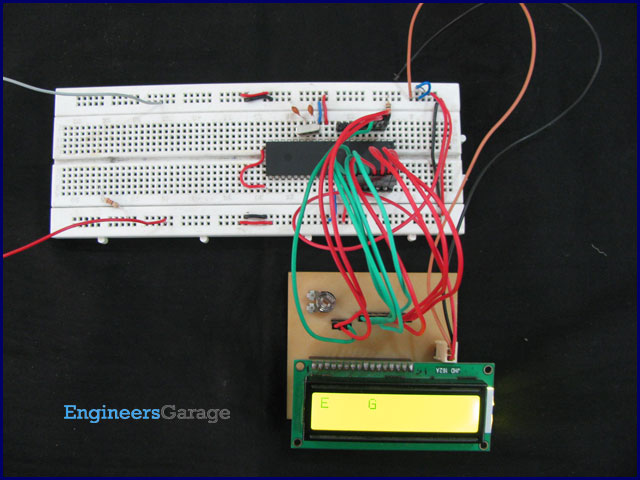
**[How to interface LCD with PIC18F4550 Microcontroller](http://www.engineersgarage.com/embedded/pic-microcontroller-projects/interface-lcd-circuit" \o "How to interface LCD with PIC18F4550 Microcontroller)**

**Summary**



The [character LCDs](http://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet) are the most commonly used display modules. These LCDs are used to display text using alphanumeric and special characters of font 5x7/5x10. For basic working and operations of a character LCD, refer [LCD interfacing with 8051](http://www.engineersgarage.com/microcontroller/8051projects/interface-lcd-at89c51-circuit). Here [PIC18F4550](http://www.engineersgarage.com/components/pic18f4550-microcontroller) has been used to display a single character on a 16x2 character LCD.

For basic details and operations of [character LCD](http://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet), refer [LCD interfacing with 8051](http://www.engineersgarage.com/microcontroller/8051projects/interface-lcd-at89c51-circuit). Here LCD has been interfaced in 8-bit mode\* with data pins (D0-D7) connected to PortB of PIC18F4550. The LCD control pins RS, R/W and EN are connected to PortA pins RA0, RA1 and RA2 respectively.

\*Character LCD can also be interfaced by using only 4 data lines. Refer [LCD interfacing in 4-bit mode](http://www.engineersgarage.com/embedded/pic-microcontroller-projects/lcd-interfacing-4bit-mode-circuit).

**Programming Steps:**

Before displaying anything on LCD, it needs to be configured with proper instructions. The following programming steps explain the procedure of configuring the LCD and display a character on it.

**Step 1**: Initialize the LCD.

The LCD must be initialized the by following pre-defined commands of character LCD.

·         0x38, to configure the LCD for 2-line, 5x7 font and 8-bit operation mode

·         0x0C, for Display On and Cursor Off

·         0x01, to Clear Display screen

·         0x06, to increment cursor

·         0x80, to set cursor position at first block of the first line of LCD.

The above set of commands is written in lcd\_ini() function of the adjoining code.

**Step 2**: Send the commands to LCD.

·         Send the command byte to the port connected to LCD data pins

·         RS=0, to select command register of LCD

·         RW=0, to set the LCD in writing mode

·         EN=1, a high to low pulse to latch command instruction

·         Delay of 1ms

·         EN=0

The above set of commands is written in lcdcmd(unsigned char) function.

**Step 3**: Send data to LCD.

·         Send data at the port which connected to LCD data pins

·         RS=1, register select to select data register of LCD

·         RW=0, this set the LCD in writing mode

·         EN=1, a high to low pulse to latch data

·         Delay of 1ms

·         EN=0

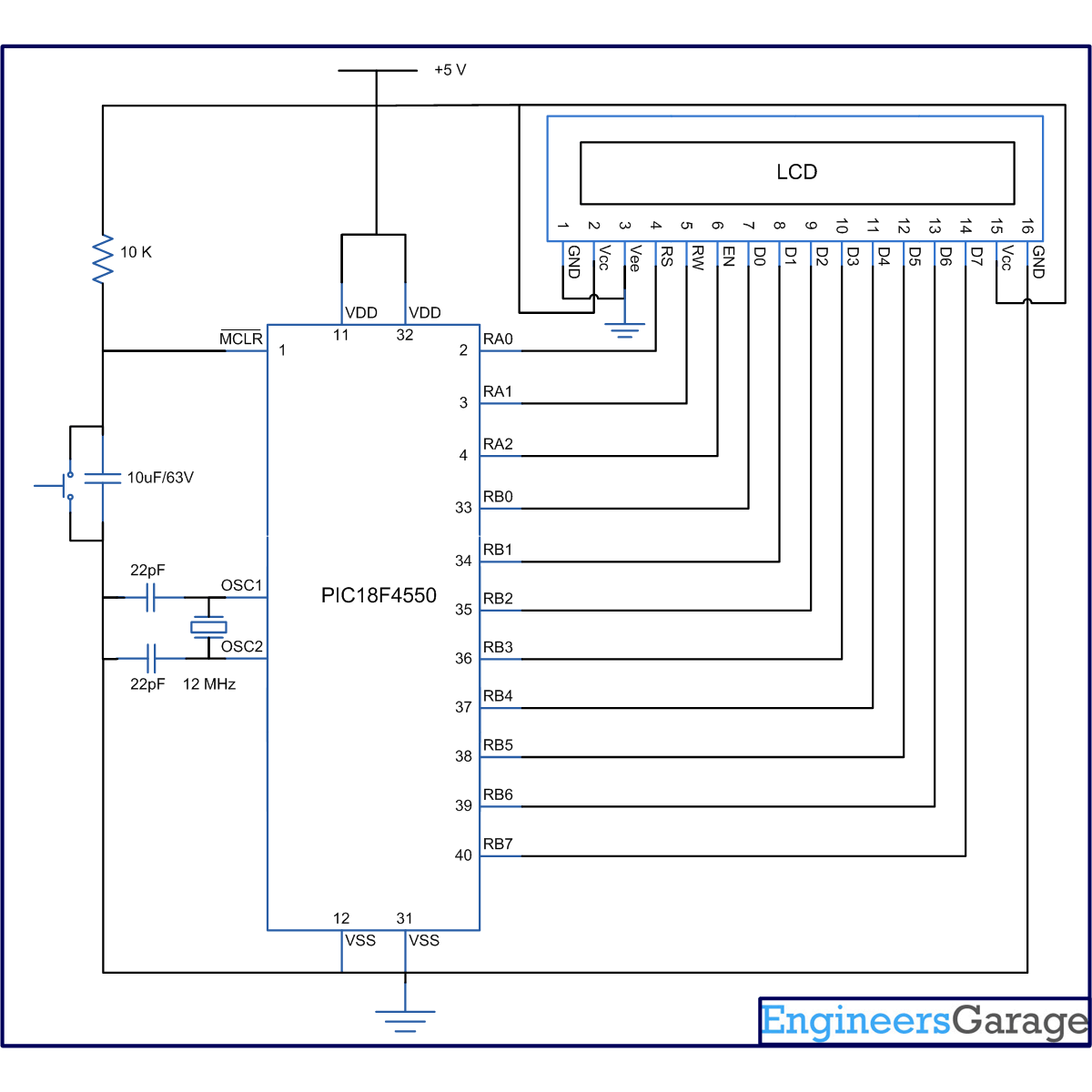
The lcddata(unsigned char) function has the above set of instructions.

**Step 4**: Display character on LCD.

The functions lcdcmd() and lcddata() are user-defined functions. They are used to send a character (E in this case) to be displayed on LCD.

lcdcmd(0x38);             // send command 0x38 to LCD

lcddata(‘E’);                // send character E to LCD



**// Program to interface 16x2 LCD and display single character using PIC18F4550 Microcontroller**  
  
// Configuration bits  
/\* \_CPUDIV\_OSC1\_PLL2\_1L, // Divide clock by 2  
 \_FOSC\_HS\_1H, // Select High Speed (HS) oscillator  
 \_WDT\_OFF\_2H, // Watchdog Timer off  
 MCLRE\_ON\_3H // Master Clear on  
\*/  
  
//LCD Control pins  
#define rs LATA.F0  
#define rw LATA.F1  
#define en LATA.F2  
  
//LCD Data pins  
#define lcdport LATB  
  
void lcd\_ini();  
void lcdcmd(unsigned char);  
void lcddata(unsigned char);  
unsigned int i=0;  
  
void main(void)  
{  
 TRISA=0; // Configure Port A as output port  
 LATA=0;  
 TRISB=0; // Configure Port B as output port  
 LATB=0;  
 lcd\_ini(); // LCD initialization  
 lcddata('E'); // Print 'E'  
 Delay\_ms(1000);  
 lcdcmd(0x85); // Position 1st Line, 6th Column  
 lcddata('G'); // Print 'G'  
  
}  
void lcd\_ini()  
{  
 lcdcmd(0x38); // Configure the LCD in 8-bit mode, 2 line and 5x7 font  
 lcdcmd(0x0C); // Display On and Cursor Off  
 lcdcmd(0x01); // Clear display screen  
 lcdcmd(0x06); // Increment cursor  
 lcdcmd(0x80); // Set cursor position to 1st line, 1st column  
}  
  
void lcdcmd(unsigned char cmdout)  
{  
 lcdport=cmdout; //Send command to lcdport=PORTB  
 rs=0;   
 rw=0;  
 en=1;  
 Delay\_ms(10);  
 en=0;  
}  
  
void lcddata(unsigned char dataout)  
{  
 lcdport=dataout; //Send data to lcdport=PORTB  
 rs=1;  
 rw=0;  
 en=1;  
 Delay\_ms(10);  
 en=0;  
}