

MICROPROCESSOR BASED SYSTEM DESIGN LAB

LAB 10



Spring 2021

CSE307L MBSD Lab

Submitted by: **Shah Raza**

Registration No. : **18PWCSE1658**

Class Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: _____

Submitted to:

Engr. Amaad Khalil

Tuesday, August 10, 2021

Department of Computer Systems Engineering
University of Engineering and Technology, Peshawar

Task 1:

Write a code for calculator using 8051 MCU with 16x2 LCD and Keypad.

Code:

```
#include <reg51.h>
#include <stdio.h>

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

//*****
//Pin description
/*
P2 is data bus
P3.7 is RS
P3.6 is E
P1.0 to P1.3 are keypad row outputs
P1.4 to P1.6 are keypad column inputs
*/
//*****
// Define Pins
//*****
sbit RowA = P1^0; //RowA
sbit RowB = P1^1; //RowB
sbit RowC = P1^2; //RowC
sbit RowD = P1^3; //RowD

sbit C1 = P1^4; //Column1
sbit C2 = P1^5; //Column2
sbit C3 = P1^6; //Column3
sbit C4 = P1^7; //Column4

sbit E = P3^6; //E pin for LCD
sbit RS = P3^7; //RS pin for LCD

unsigned int key_count = 0, result_int;
char array[3], result_char;
```

```

// *****
// Main program
//
int main(void)
{
    char key;          // key char for keeping record of pressed key

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

    writecmd(0x95);
        writedata('T');          //write
        writedata('i');          //write
        writedata('m');          //write
        writedata('e');          //write
        writedata(' ');          //write
        writedata('2');          //write
        writedata('3');          //write
        writedata(':');          //write
        writedata('5');
        writedata('9');
        writedata(':');
        writedata('2');
        writedata('7');
    writecmd(0xd5);
        writedata('D');          //write
        writedata('a');          //write
        writedata('t');          //write
        writedata('e');          //write
        writedata(' ');          //write
        writedata('3');          //write
        writedata('1');          //write
        writedata('/');          //write
        writedata('1');
        writedata('2');
        writedata('/');
        writedata('2');
        writedata('0');
        writedata('2');
    writedata('1');              //write

    writecmd(0x80);
    while(1)
    {
        key = get_key();        // Get pressed key
        if (key == 'C')

```

```

        {
            writcmd(0x01);    // Clear screen
            array[0]= array[1]= array[2]='n';
            result_int = 0; key_count = 0;
        }
    else if(key == '=')
    {
        writedata(key);    // Echo the key pressed to LCD
        if(array[1] == 'x')
            result_int = (array[0] - '0') * (array[2] - '0');
        else if(array[1] == '/')
            result_int = (array[0] - '0') / (array[2] - '0');
        else if(array[1] == '+')
            result_int = (array[0] - '0') + (array[2] - '0');
        else if(array[1] == '-')
            result_int = (array[0] - '0') - (array[2] - '0');
        if(result_int > 9)    //if the result is greater than 9 then we need to store
each digit in an array
        {
            int result[2];    //Delare an array
            result[0] = result_int/10; //Store the digit at ten's place at index 0
            result[1] = result_int%10; //Store the digit at unit's place at index 1
            result_char = (char)result[0] + '0'; //Convert the integer at index 0 to
character

            writedata(result_char );    // Echo the key pressed to LCD
            result_char = (char)result[1] + '0'; //Convert the integer at index 1 to
character

            writedata(result_char );    // Echo the key pressed to LCD
        }
        else
        {
            result_char = (char)result_int + '0';
            writedata(result_char );    // Echo the key pressed to LCD
        }
    }
    else
        writedata(key);    // Echo the key pressed to LCD
}
}

void cct_init(void)
{
    P0 = 0x00; //not used
    P1 = 0xf0; //used for generating outputs and taking inputs from Keypad
    P2 = 0x00; //used as data port for LCD

```

```

        P3 = 0x00; //used for RS and E
    }

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

void writedata(char t)
{
    RS = 1;          // This is data
    P2 = t;          //Data transfer
    E = 1;           // => E = 1
    delay(150);
    E = 0;           // => E = 0
    delay(150);
}

void writecmd(int z)
{
    RS = 0;          // This is command
    P2 = z;          //Data transfer
    E = 1;           // => E = 1
    delay(150);
    E = 0;           // => E = 0
    delay(150);
}

void lcdinit(void)
{
    //////////// Reset process from datasheet ////////////
    delay(15000);
    writecmd(0x30);
    delay(4500);
    writecmd(0x30);
    delay(300);
    writecmd(0x30);
    delay(650);
    //////////////////////////////////////
    writecmd(0x38); //function set
    writecmd(0x0c); //display on,cursor off,blink off
    writecmd(0x01); //clear display
    writecmd(0x06); //entry mode, set increment
}

```

```

void Return(void)    //Return to 0 location on LCD
{
    writecmd(0x02);
    delay(1500);
}

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'; }
    if (C4 == 0) { delay(10000); while (C4==0); return '/'; }

    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'; }
    if (C4 == 0) { delay(10000); while (C4==0); return 'x'; }

    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'; }
    if (C4 == 0) { delay(10000); while (C4==0); return '-'; }

    RowA = 1; RowB = 1; RowC = 1; RowD = 0;    //Test Row D

    if (C1 == 0) { delay(10000); while (C1==0); return 'C'; }
    if (C2 == 0) { delay(10000); while (C2==0); return '0'; }
    if (C3 == 0) { delay(10000); while (C3==0); return '='; }
    if (C4 == 0) { delay(10000); while (C4==0); return '+'; }
    return 'n';    // Means no key has been pressed
}

char get_key(void)    //get key from user
{
    char key = 'n';    //assume no key pressed

    while(key=='n')    //wait untill a key is pressed
        key = READ_SWITCHES(); //scan the keys again and again
    array[key_count++]=key;
}

```

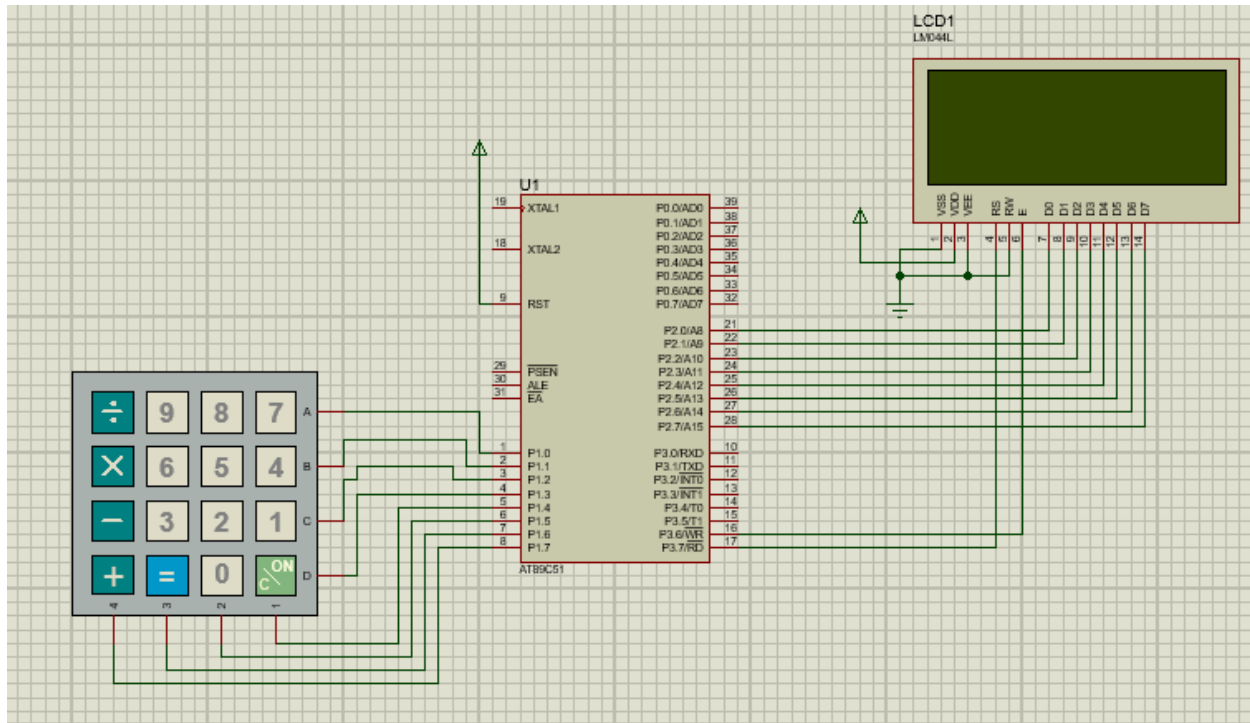
```

    return key;           //when key pressed then return its value
}

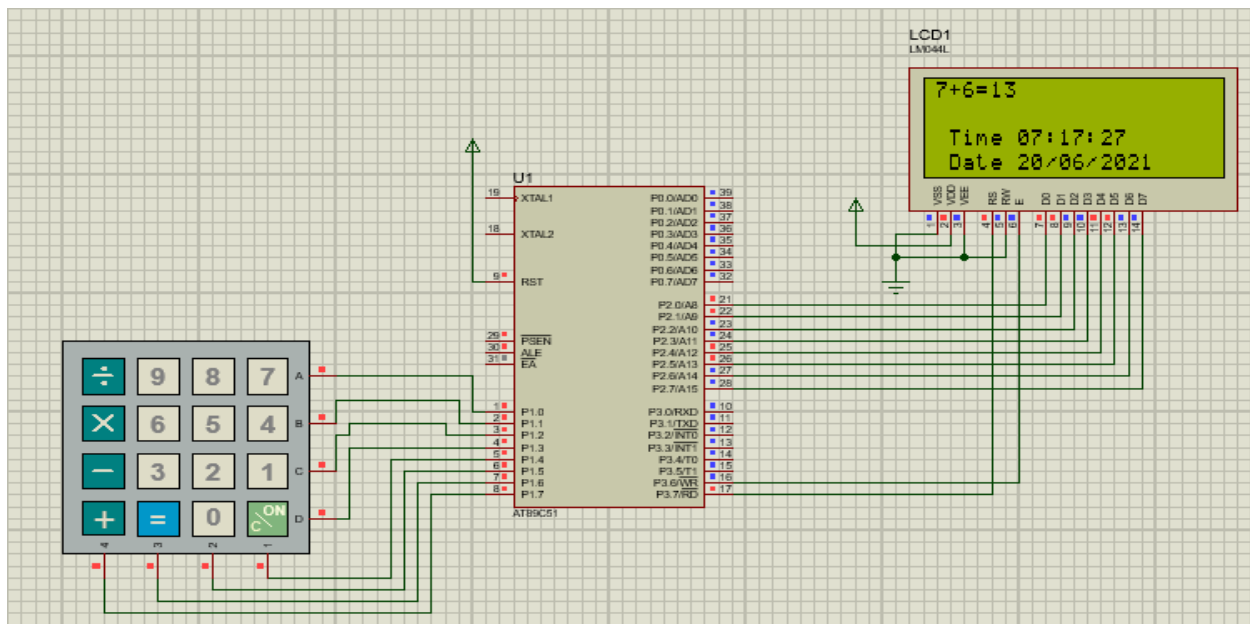
```

Output / Graphs / Plots / Results:

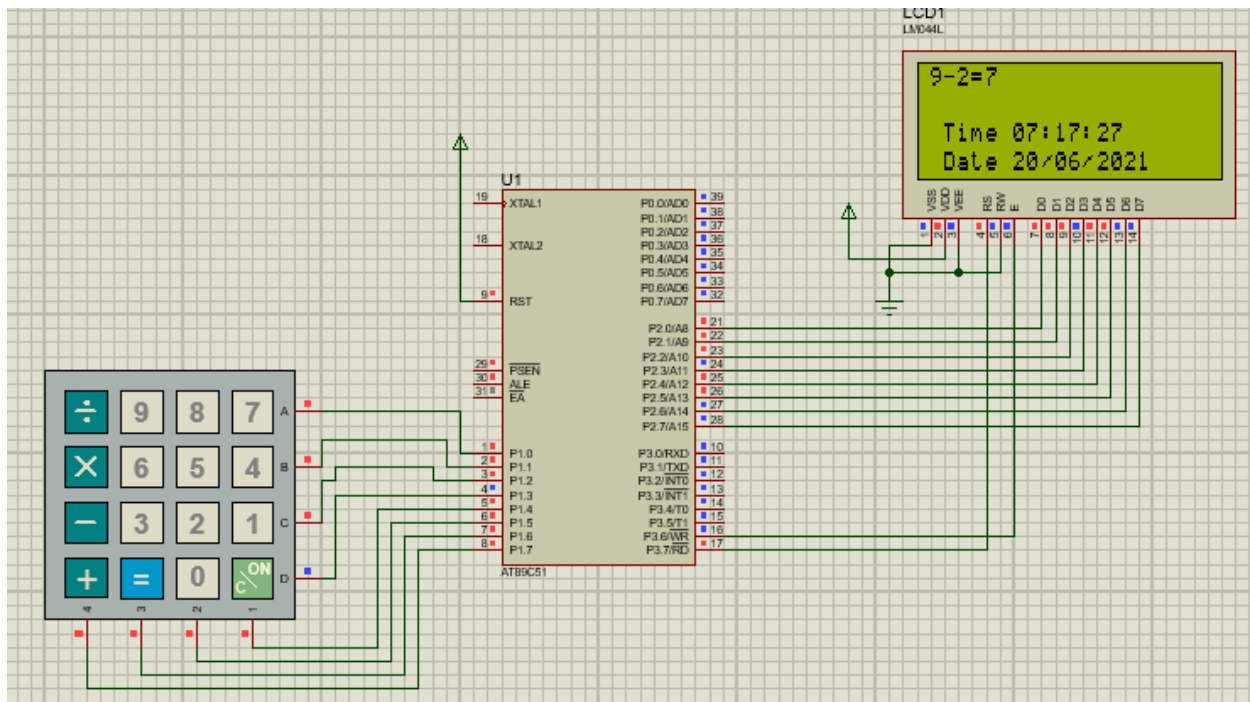
Schematic:



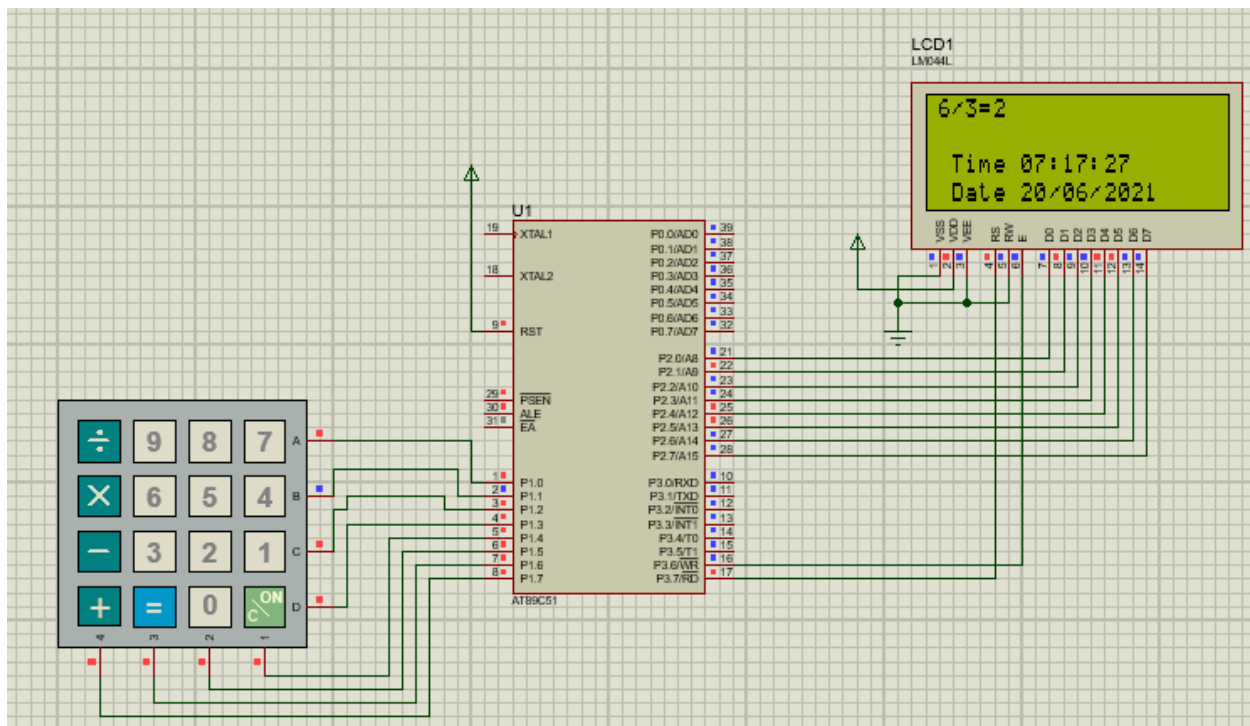
Addition:



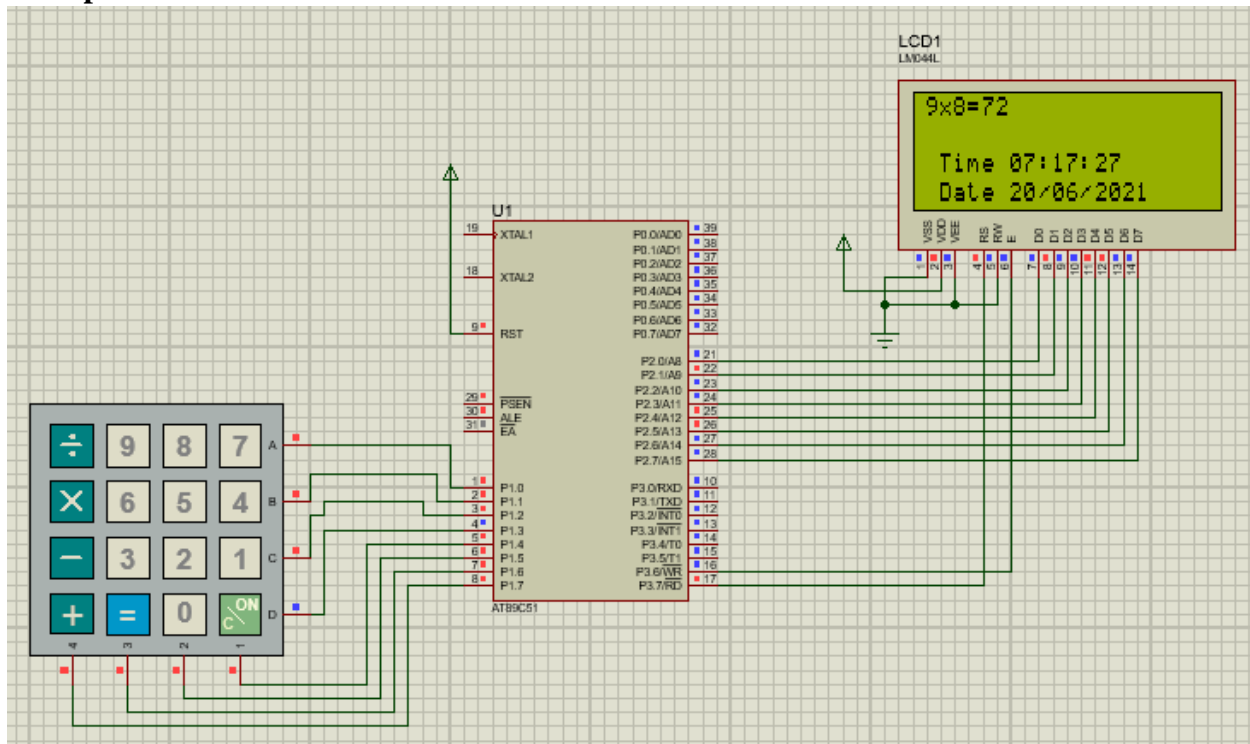
Subtraction:



Division:



Multiplication:



Task 2:

Write a code custom characters on 16x2 LCD using 8051 microcontroller.

Code:

```
#include <reg51.h>
#include <string.h>
/* Function declarations */
void cct_init(void);
void delay(int);
void lcdinit(void);
void writecmd(int);
void writedata(char);
void writeline(char[]);
void ReturnHome(void);
void LCDBuildChar(unsigned char, unsigned char*);

// ----- Define Custom Characters ----- //
unsigned char Character1[8] = { 0x1c,0x16,0x1d,0x01,0x1d,0x16,0x1c,0x00 }; // Phone Up 1
unsigned char Character2[8] = { 0x07,0x0d,0x17,0x10,0x17,0x0d,0x07,0x00 }; // Phone Up 2
unsigned char Character3[8] = { 0x1f,0x1f,0x12,0x04,0x09,0x10,0x1f,0x00 }; // Phone Down 1
unsigned char Character4[8] = { 0x1f,0x1f,0x09,0x04,0x12,0x01,0x1f,0x00 }; // Phone Down 2
unsigned char Character5[8] = { 0x04,0x04,0x04,0x04,0x15,0x0e,0x04,0x00 }; // Downward Arrow
unsigned char Character6[8] = { 0x04,0x0e,0x15,0x04,0x04,0x04,0x04,0x00 }; // Upward Arrow
```

```

unsigned char Character7[8] = { 0x01,0x02,0x04,0x08,0x10,0x11,0x1f,0x00 }; // Curvy Object
unsigned char Character8[8] = { 0x00,0x1f,0x11,0x11,0x11,0x1f,0x00,0x00 }; // Square Box
/*-----*/
/* Main program */
int main(void)
{
    cct_init();           /* Make all ports zero */
    lcdinit();           /* Initilize LCD */
    writedata(0x00);      /* Write Custom Character 1 */
    writedata(0x01);      /* Write Custom Character 2 */
    writedata(' ');      // Space
    writedata(0x02);      /* Write Custom Character 3 */
    writedata(0x03);      /* Write Custom Character 4 */
    writedata(' ');      // Space
    writedata(0x04);      /* Write Custom Character 5 */
    writedata(0x05);      /* Write Custom Character 6 */
    writedata(' ');      // Space
    writedata(0x06);      /* Write Custom Character 7 */
    writedata(0x07);      /* Write Custom Character 8 */
    while(1)
    {;}
}

void cct_init(void)
{
    P0 = 0x00; /* not used */
    P1 = 0x00; /* not used */
    P2 = 0x00; /* used as data port */
    P3 = 0x00; /* used for generating E and RS */
}

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

void writedata(char t)
{
    P3 |= 0x01; /* => RS = 1 */
    P2 = t; /* Data transfer */
    P3 |= 0x02; /* => E = 1 */
    delay(150);
    P3 &= 0xfd; /* => E = 0 */
    delay(150);
}

void writecmd(int z)
{
    P3 &= 0xfe; /* => RS = 1 */
    P2 = z; /* Data transfer */
    P3 |= 0x02; /* => E = 1 */
}

```

```

    delay(150);
    P3 &= 0xfd;    /* => E = 0    */
    delay(150);
}
void lcdinit(void)
{
    /* ----- Reset process from datasheet ----- */
    delay(15000);
    writecmd(0x30);
    delay(4500);
    writecmd(0x30);
    delay(300);
    writecmd(0x30);
    delay(650);
    /* ----- */
    writecmd(0x38); /* function set */
    writecmd(0x0c); /* display on,cursor off,blink off */
    writecmd(0x01); /* clear display */
    writecmd(0x06); /* entry mode, set increment */
    /* ----- Build Custom Characters ----- */
    LCDBuildChar(0, Character1); /* Build Character1 at position 0 */
    LCDBuildChar(1, Character2); /* Build Character2 at position 1 */
    LCDBuildChar(2, Character3); /* Build Character3 at position 2 */
    LCDBuildChar(3, Character4); /* Build Character4 at position 3 */
    LCDBuildChar(4, Character5); /* Build Character5 at position 4 */
    LCDBuildChar(5, Character6); /* Build Character6 at position 5 */
    LCDBuildChar(6, Character7); /* Build Character6 at position 6 */
    LCDBuildChar(7, Character8); /* Build Character6 at position 7 */
}
void ReturnHome(void) /* Return to 0 cursor location */
{
    writecmd(0x02);
    delay(1500);
}
void writeline(char Line[])
{
    int i;
    for(i=0;i<strlen(Line);i++)
    {
        writedata(Line[i]); /* Write Character */
    }
    ReturnHome(); /* Return to 0 cursor position */
}

void LCDBuildChar(unsigned char loc, unsigned char *p)
{
    unsigned char i;
    if(loc<8) /*If valid address

```

```

    {
        writcmd(0x40+(loc*8));          //Write to CGRAM
        for(i=0;i<8;i++)
            writedata(p[i]);            //Write the character pattern to CGRAM
    }

    writcmd(0x80);                      //shift back to DDRAM location 0
}

```

Output / Graphs / Plots / Results:

