



P8LCD Display (GPIOs)

Microcontroladores



**Tecnológico
de Monterrey**

Eduardo Rafael Silva de la
Peña A00819263



- Primer inciso al correr el programa y muestra un mensaje **BUILD SUCCESSFUL**

```

void portInit( void ) {
    ANSELB = digital;
    ANSELB = digital;
    TRISBbits.TRISB0 = output;
    TRISBbits.TRISB1 = output;
    TRISBbits.TRISB2 = output;
    TRISBbits.TRISB3 = output;
    TRISBbits.TRISB4 = input;
    TRISBbits.TRISB5 = input;
    TRISBbits.TRISB6 = input;
    TRISBbits.TRISB7 = input;
    WPUB = 0b11110000;
    INTCON2bits.RBPV = 0;
    ANSELE = digital;
}

```

CODIGO

En las imágenes podemos analizar el programa completo

```

1 #include "../config_header.h"
2 #define _XTAL_FREQ 1000000
3 #define LCD_DATA_R PORTD
4 #define LCD_DATA_W LATD
5 #define LCD_DATA_DIR TRISD
6 #define LCD_RS LATCbits.LATC2
7 #define LCD_RS_DIR TRISCbits.TRISC2
8 #define LCD_RW LATCbits.LATC1
9 #define LCD_RW_DIR TRISCbits.TRISC1
10 #define LCD_E LATCbits.LATC0
11 #define LCD_E_DIR TRISCbits.TRISC0
12
13 enum port_dir{ output, input };
14 enum port_DCAC{ digital, analog };
15 enum led_state{ led_OFF, led_ON };
16 enum button_state{ pushed, no_pushed };
17
18 enum button_state{ pushed, no_pushed };
19
20 __interrupt(high_priority) void high_isr(void){
21     Nop();
22 }
23
24 __interrupt(low_priority) void low_isr(void){
25     Nop();
26 }
27
28 void delay(void);
29 void portInit(void);
30 void LCD_rdy(void);
31 void LCD_init(void);
32 void LCD_cmd(char cx);
33 void send2LCD(char xy);
34 char keypad(void);

```

```

void main(void) {
    char y;
    char arreglo [3];
    char count = 0;
    char resultado;
    char resultado_d;
    char resultado_u;
    portInit ();
    LCD_init();
    LCD_cmd(0x80);
    while (1){
        y = keypad ();
        _delay_ms(300);
        arreglo [count] = y;
        y = y +48;
        send2LCD(y);

        count ++;
        if (y == 61) {
            arreglo [1] = arreglo [1] +48;
            if (arreglo [1] == 43){

                resultado = arreglo [0] + arreglo [2];

            }
            if (arreglo [1] == 42){

                resultado = arreglo [0] * arreglo [2];

            }

            if (arreglo [1] == 45){

                resultado = arreglo [0] - arreglo [2];

```

```

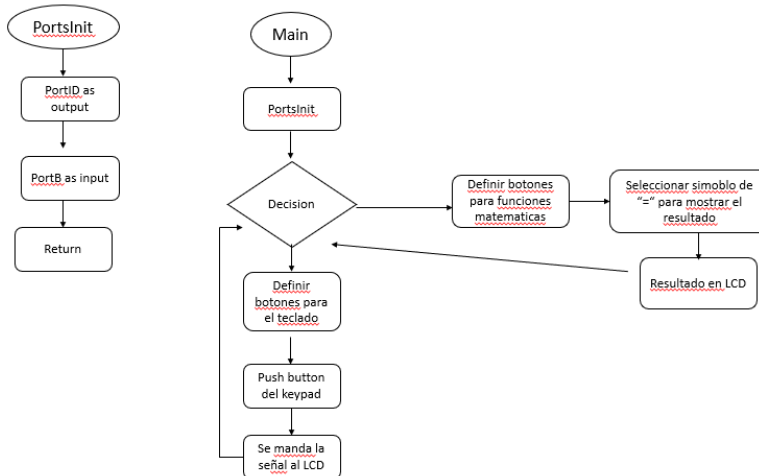
void LCD_cmd( char cx ){
    LCD_rdy();
    LCD_RS=0;
    LCD_RW=0;
    LCD_E=1;
    Nop();
    LCD_DATA_W = cx;
    Nop();
    LCD_E=0;
}

void send2LCD( char xy ){
    LCD_RS=1;
    LCD_RW=0;
    LCD_E=1;
    LCD_DATA_W=xy;
    Nop();
    Nop();
    LCD_E=0;
}

void portInit( void ) {
    ANSEL_D = digital;
    ANSEL_B = digital;
    TRISBbits.TRISB0 = output;
    TRISBbits.TRISB1 = output;
    TRISBbits.TRISB2 = output;
    TRISBbits.TRISB3 = output;
    TRISBbits.TRISB4 = input;
    TRISBbits.TRISB5 = input;
    TRISBbits.TRISB6 = input;
    TRISBbits.TRISB7 = input;
    WPUB = 0b11110000;
    INTCON2bits.RBPV = 0;
    ANSEL_E = digital;
}

```

Flow diagram



Explicación

Este programa tiene como función programar la pantalla LCD reflejando los datos que se implementan, por medio del keypad 4x4. En el que debe cumplir las funciones de división, multiplicación, resta y suma. Una vez planteando la ecuación se le da al botón de "=" mostrándonos el resultado. Básicamente este programa cumple la función de una calculadora básica, en nuestra pantalla LCD.