



# MICROCONTROLADORES

# ARQUITECTURA Y ORGANIZACIÓN DE COMPUTADORAS

EQUIPO 1

INTEGRANTES:

AGUILAR MENDIOLA DAVID

HERNÁNDEZ AYALA OSCAR URIEL

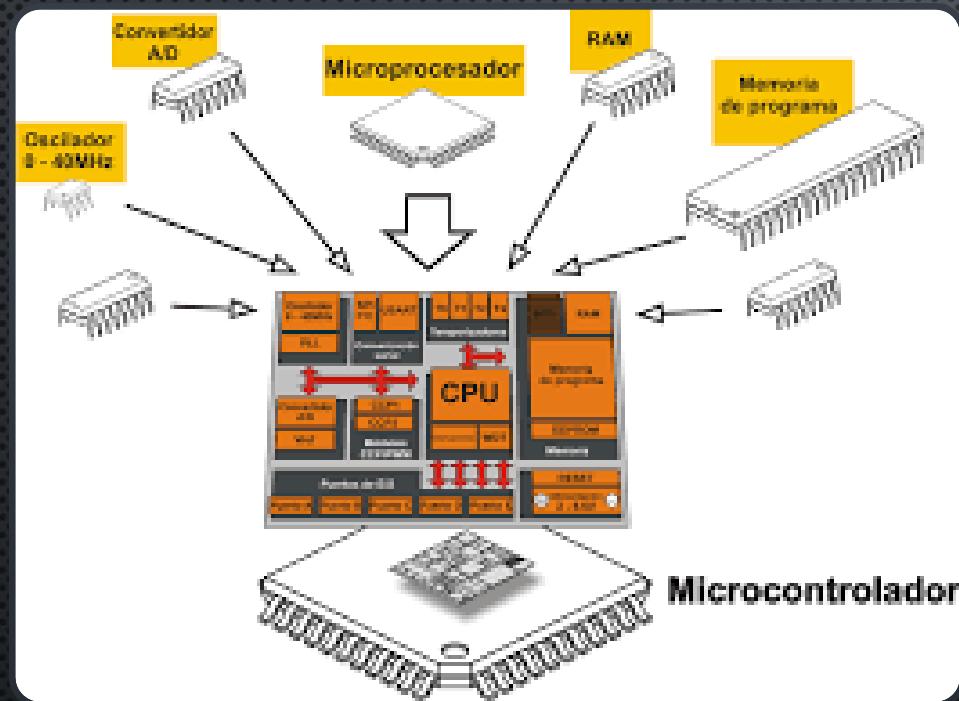
HUERTA TEHUACATL ANGEL DANIEL

TREJO MÉNDEZ EDITH ALEJANDRA

ROMERO CASTILLO CARLOS BRAYAN

GRUPO: 2NV31

# MICROCONTROLADORES



- LOS MICROCONTROLADORES SON CIRCUITOS INTEGRADOS PROGRAMABLES QUE CONTIENEN TODOS LOS ELEMENTOS NECESARIOS PARA DESARROLLAR Y CONTROLAR UNA TAREA DETERMINADA.
- ALGUNOS EJEMPLOS DE MICROCONTROLADORES SON:
  1. MEMORIA RAM.
  2. MEMORIA DE PROGRAMA .
  3. CONVERTIDOR OSCILADOR
  4. PUERTO DE COMUNICACIÓN.
  5. MICROCONTROLADORES PIC

LOS MICRO CONTROLADORES PICMICRO DE MICROCHIP SON  
CONSIDERADOS COMO LOS IDÓNEOS Y MAS ACEPTADOS  
TANTO PARA PRINCIPIANTES AFICIONADOS COMO PARA BUENA  
PARTE DE PROFESIONALES.

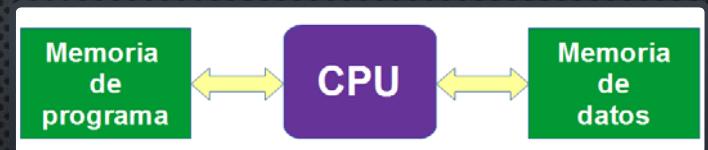
LAS VENTAJAS QUE OFRECEN LOS MICRO CONTROLADORES  
PIC SON:

- EXISTEN GRAN VARIEDAD DE FAMILIAS
- POSEEN HERRAMIENTAS DE DESARROLLO COMUNES
- EXISTE UNA GRAN CANTIDAD DE UNIDADES FUNCIONALES  
EMBEBIDAS
- PRECIOS COMPETITIVOS
- AMPLIO SOPORTE



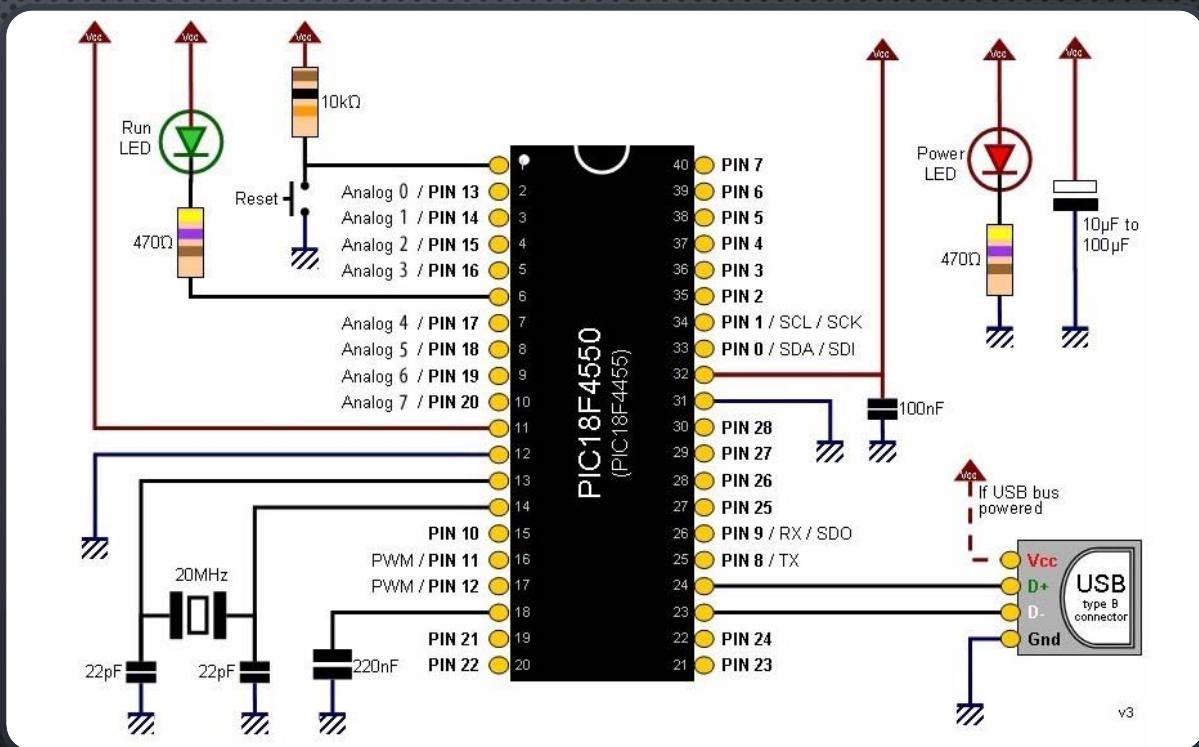
# ARQUITECTURA INTERNA

- LOS MICRO CONTROLADORES PIC UTILIZAN ARQUITECTURA HARVARD, QUE DISPONEN DE DOS MEMORIAS INDEPENDIENTES, UNA PARA EL PROGRAMA Y LA OTRA PARA LOS DATOS, CADA UNA CON SUS RESPECTIVOS BUSES.
- ADEMÁS, LOS MICRO CONTROLADORES PIC CUENTAN CON LA TECNOLOGÍA RISC, POR LO QUE POSEEN UN NÚMERO REDUCIDO DE INSTRUCCIONES Y SOLAMENTE LAS INSTRUCCIONES DE CARGA Y ALMACENAMIENTO TIENEN ACCESO A LA MEMORIA DE DATOS. SU OBJETIVO PRINCIPAL HACER POSIBLE LA SEGMENTACIÓN Y PARALELISMO EN LA EJECUCIÓN DE INSTRUCCIONES Y REDUCIR LOS ACCESOS A MEMORIA.



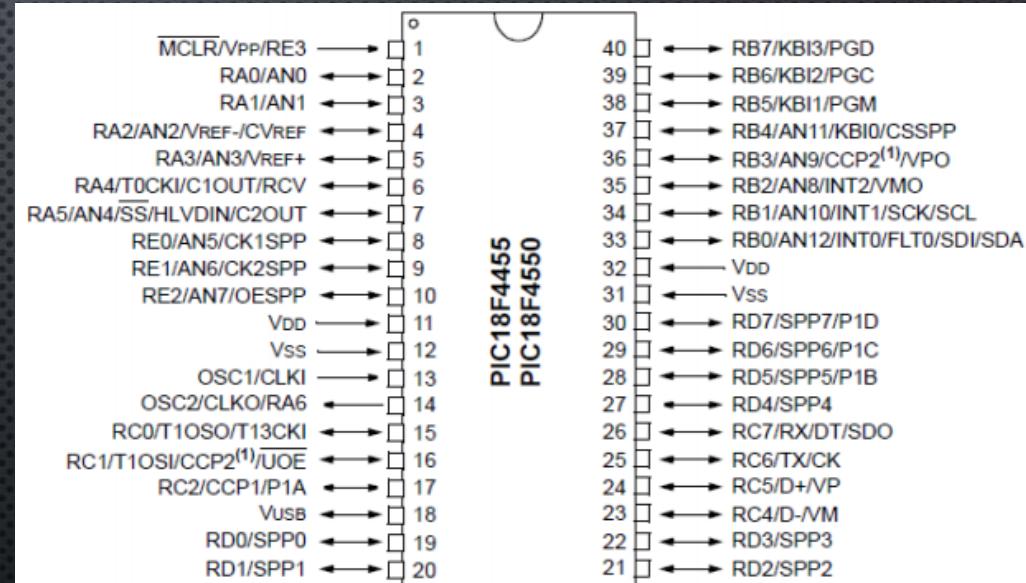
# PIC 18F4550

- LOS MICROCONTROLADORES PIC EXISTEN EN GAMAS DE 8-BIT, 16 BIT, Y 32 BIT.  
DENTRO DE LA GAMA DE 8 BIT SE ENCUENTRA EL PIC 18F4550.  
QUE FUE EL QUE UTILIZAMOS EN ESTE PROYECTO.



# HOJA DE DATOS DEL PIC18F4550

LOS MICRO CONTROLADORES PIC EXISTEN EN GAMAS DE 8-BIT Y 32-BIT. EL PIC18G4550 PERTENECE A LA FAMILIA PIC18 MCU. SUS CARACTERÍSTICAS DE MEMORIA DE PROGRAMA, MEMORIA DE RAM, NÚMERO DE ENTRADAS/SALIDAS, NUMERO DE CANALES ANALÓGICOS Y TIPOS DE PUERTOS DE COMUNICACIÓN, CONVIRTIÉNDOSE EN UNO DE LOS MÁS UTILIZADOS PARA DIVERSAS APLICACIONES.



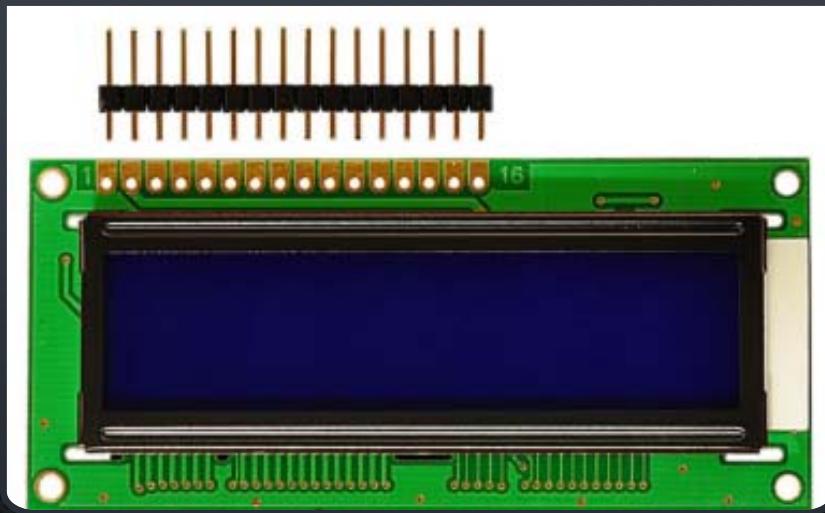
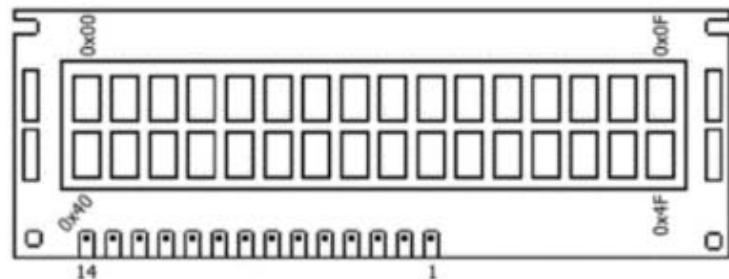
# CARACTERÍSTICAS PIC 18F4550.

Features	PIC18F4550
Operating Frequency	DC – 48 MHz
Program Memory (Bytes)	32768
Program Memory (Instructions)	16384
Data Memory (Bytes)	2048
Data EEPROM Memory (Bytes)	256
Interrupt Sources	20
I/O Ports	Ports A, B, C, D, E
Timers	4
Capture/Compare/PWM Modules	1
Enhanced Capture/ Compare/PWM Modules	1
Serial Communications	MSSP, Enhanced USART
Universal Serial Bus (USB) Module	1
Streaming Parallel Port (SPP)	Yes
10-Bit Analog-to-Digital Module	13 Input Channels
Comparators	2
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT
Programmable Low-Voltage Detect	Yes
Programmable Brown-out Reset	Yes
Instruction Set	75 Instructions; 83 with Extended Instruction Set enabled
Packages	40-Pin PDIP 44-Pin QFN 44-Pin TQFP

# LCD



LCD Display with 2 lines x 16 characters :



## LCD

- EL LCD CUENTA CON DOS LÍNEAS DE TRABAJO Y 16 COLUMNAS PARA REPRESENTAR EL TEXTO QUE SE DESEE, PARA QUE UTILIZAR UN LCD, PARA PODER MONITOREAR RESULTADOS, PARA PRESENTAR UN MENÚ DE TRABAJO, DIGAMOS ES UNA FORMA DE INTERACTUAR CON EL USUARIO.

# LCD

- CUENTA CON 16 PINES PARA LA COMUNICACIÓN DE DONDE LOS DEFINIREMOS EN 3 BLOQUES, PINES DE ALIMENTACIÓN, PINES DE CONTROL Y PINES DE COMUNICACIÓN DE DATOS.

FUNCIÓN	NÚMERO	NOMBRE	ESTADO LÓGICO	DESCRIPCIÓN
Tierra	1	Vss	-	0V
Alimentación	2	Vdd	-	+5V
Contraste	3	Vee	-	0 - Vdd
Control de funcionamiento	4	RS	0 1	D0 – D7 considerados como comandos D0 – D7 considerados como datos
	5	R/W	0 1	Escribir los datos (del microcontrolador al LCD) Leer los datos (del LCD al microcontrolador)
	6	E	0 1 Transición de 1 a 0	Acceso al visualizador LCD deshabilitado Funcionamiento normal Datos/comandos se están transmitiendo al LCD
	7	D0	0/1	Bit 0 LSB

Datos / comandos	8	D1	0/1	Bit 1
	9	D2	0/1	Bit 2
	10	D3	0/1	Bit 3
	11	D4	0/1	Bit 4
	12	D5	0/1	Bit 5
	13	D6	0/1	Bit 6
	14	D7	0/1	Bit 7 MSB

# LCD

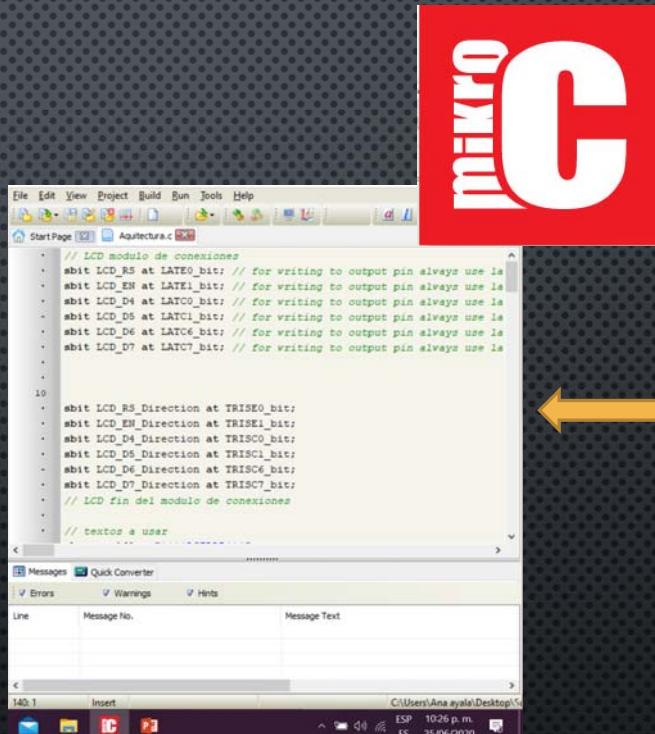
- UNA FUNCIÓN QUE TIENE EL LCD, ES QUE SE PRESENTA UN EFECTO DE MOVIMIENTO COMO SI ENTRARA O SALIERAN LAS LETRAS, EN REALIDAD LO QUE TENEMOS ES UN ARIA DE LA MEMORIA DONDE SE COLOCAN LOS CARACTERES FIJOS Y LO QUE SE MUEVE ES LA PANTALLA, COMO SE VE EN LA FIG. SIGUIENTE. LO DE VERDE MÁS OSCURO ES LO QUE SE DESPLAZA.



PROYECTO

## PROGRAMA

PARA EL PROGRAMA SE OCUPO MIKROC QUE ES UN COMPILADOR DE ENSAMBLADOR BASADO EN C DONDE TE PERMITE SIMPLIFICAR FUNCIONES QUE EN ENSAMBLADOR (ASM) SIGNIFICARÍAN UNA GRAN CANTIDAD DE LÍNEAS DE CÓDIGO.



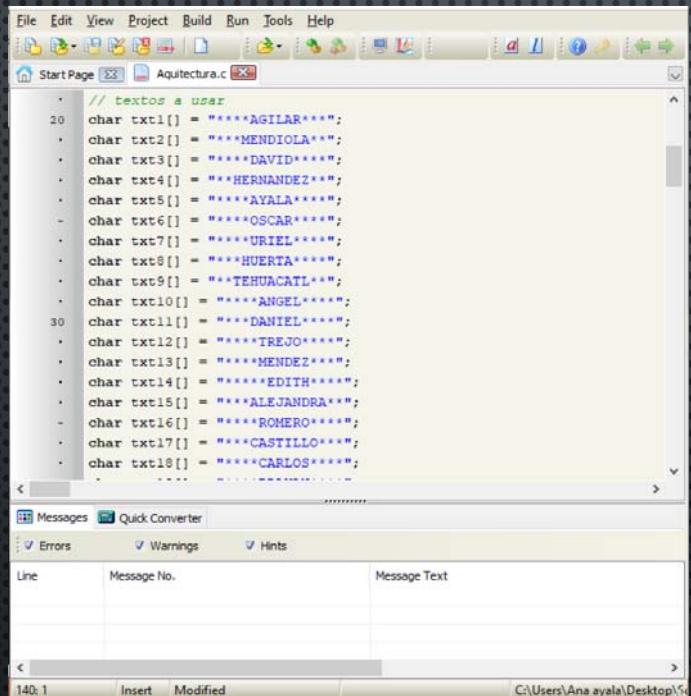
The screenshot shows the MikroC IDE interface with the following code:

```
File Edit View Project Build Run Tools Help
Start Page A Archetura.c
// LCD modulo de conexiones
. sbit LCD_RS at LATC0_bit; // for writing to output pin always use is
. sbit LCD_EN at LATC1_bit; // for writing to output pin always use is
. sbit LCD_D4 at LATC0_bit; // for writing to output pin always use is
. sbit LCD_D5 at LATC1_bit; // for writing to output pin always use is
. sbit LCD_D6 at LATC6_bit; // for writing to output pin always use is
. sbit LCD_D7 at LATC7_bit; // for writing to output pin always use is
.
10 . sbit LCD_RS_Direction at TRISD0_bit;
. sbit LCD_EN_Direction at TRISD1_bit;
. sbit LCD_D4_Direction at TRISD0_bit;
. sbit LCD_D5_Direction at TRISD1_bit;
. sbit LCD_D6_Direction at TRISD6_bit;
. sbit LCD_D7_Direction at TRISD7_bit;
. // LCD fin del modulo de conexiones
.
// textos a usar
```

Below the code editor is a messages window showing no errors or warnings. The status bar at the bottom indicates the file path as C:\Users\Ana ayala\Desktop\S and the date/time as ESP 10:26 p.m. ES 25/06/2020.

EN ESTA PARTE DEL PROGRAMA SE DECLARAN LOS PUERTOS O LOS PINES DEL LCD PARA QUE ESTEN ACTIVOS.

AQUI SE OCUPO UNA FUNCION (CHAR TXT#{}) QUE TINE MIKROC QUE TE PERMITE DECLARAR UNA LINEA COMPLETA DEL DISPLAY SIN TENER QUE DECLARAR LETRA POR LETRA LA PALABRA.



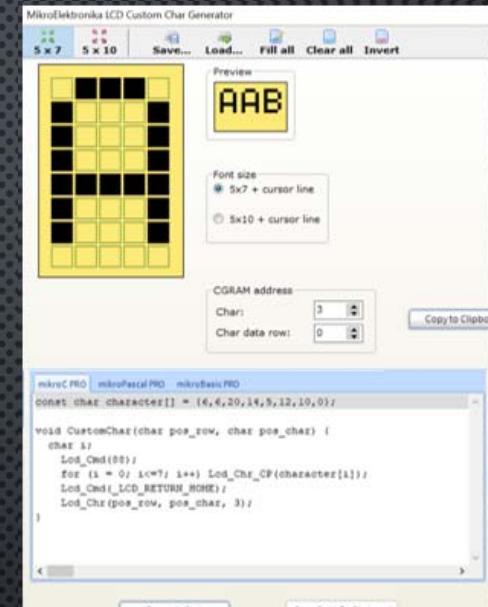
```

File Edit View Project Build Run Tools Help
Start Page Aquitectura.c
// textos a usar
20 char txt1[] = "*****AGILAR*****";
char txt2[] = "*****MENDIOLA*****";
char txt3[] = "*****DAVID*****";
char txt4[] = "*****HERNANDEZ*****";
char txt5[] = "*****AYALA*****";
char txt6[] = "*****OSCAR*****";
char txt7[] = "*****URIEL*****";
char txt8[] = "*****HUERTA*****";
char txt9[] = "*****TEHUACATL*****";
char txt10[] = "*****ANGEL*****";
30 char txt11[] = "*****DANIEL*****";
char txt12[] = "*****TREJO*****";
char txt13[] = "*****MENDEZ*****";
char txt14[] = "*****EDITH*****";
char txt15[] = "*****ALEJANDRA*****";
char txt16[] = "*****ROMERO*****";
char txt17[] = "*****CASTILLO*****";
char txt18[] = "*****CARLOS*****";

```

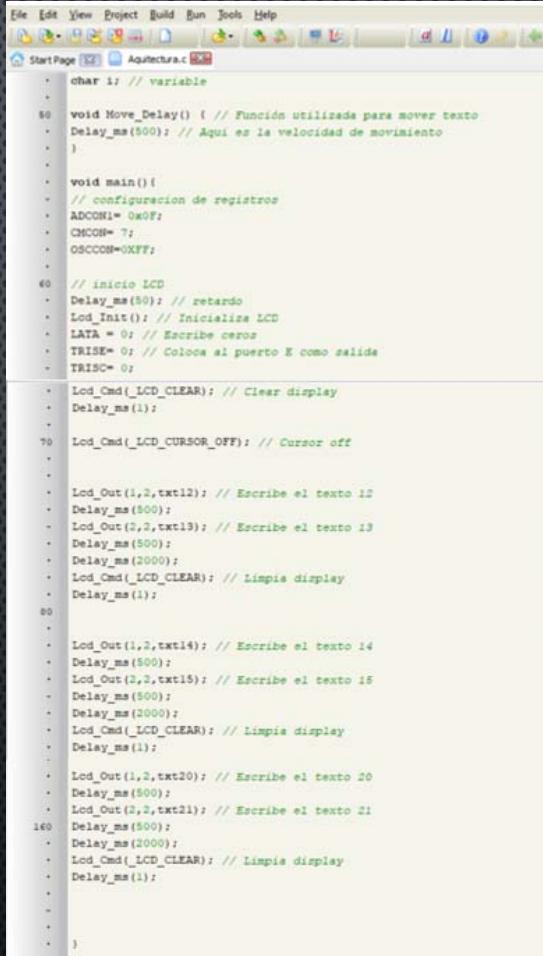
Messages Quick Converter

Line	Message No.	Message Text
140:1	Insert Modified	C:\Users\Ana ayala\Desktop\S



COMO SE TENDRIA QUE DECLARAR PALABRA POR PALABRA.

EN ESTA PARTE YA SE ESCRIBE LO QUE UNO QUIRE QUE REALICE EL MICROCONTROLADOR Y A QUE PINES DEL DISPLAY SE ENVIARA LA SEÑAL PARA QUE SE IMPRIMA EL TEXTO.



```
char i; // variable

void Move_Delay() { // Función utilizada para mover texto
    Delay_ms(500); // Aquí es la velocidad de movimiento
}

void main(){
    // configuración de registros
    ADCON1= 0x0F;
    CMCON= 7;
    OSCCON=0XFF;

    // inicio LCD
    Delay_ms(50); // retardo
    Lcd_Init(); // Inicializa LCD
    LATB = 0; // Escribe ceros
    TRISE= 0; // Coloca al puerto E como salida
    TRISB= 0;

    Lcd_Cmd(_LCD_CLEAR); // Clear display
    Delay_ms(1);

    Lcd_Cmd(_LCD_CURSOR_OFF); // Cursor off

    Lcd_Out(1,2,txt12); // Escribe el texto 12
    Delay_ms(500);
    Lcd_Out(2,2,txt13); // Escribe el texto 13
    Delay_ms(500);
    Delay_ms(2000);
    Lcd_Cmd(_LCD_CLEAR); // Limpia display
    Delay_ms(1);

    Lcd_Out(1,2,txt14); // Escribe el texto 14
    Delay_ms(500);
    Lcd_Out(2,2,txt15); // Escribe el texto 15
    Delay_ms(500);
    Delay_ms(2000);
    Lcd_Cmd(_LCD_CLEAR); // Limpia display
    Delay_ms(1);

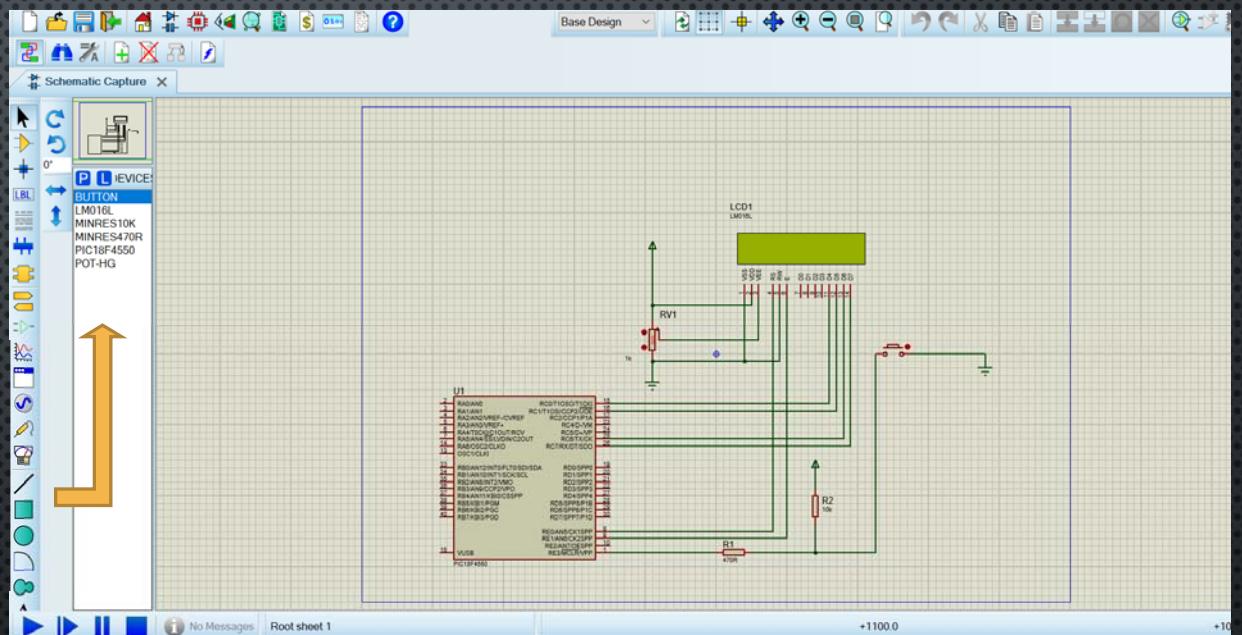
    Lcd_Out(1,2,txt20); // Escribe el texto 20
    Delay_ms(500);
    Lcd_Out(2,2,txt21); // Escribe el texto 21
    Delay_ms(500);
    Delay_ms(2000);
    Lcd_Cmd(_LCD_CLEAR); // Limpia display
    Delay_ms(1);

    ...
}
```

AQUÍ SE EMPLARON FUNCIONES DE RETRASO, MOVIMIENTO DE TEXTO, DECLARACION DE VARIABLES, DECLARACION DE PUERTOS, CONFIGURACION DE REGISTROS Y ELIMINACION DE REGISTROS.

## ARMADO

AQUÍ OCUPAMOS PROTEUS 8 PARA REALIZAR LA SIMULACION. LO QUE SE OCUPÓ PARA ARMAR EL CIRCUITO SE PRESENTA ACONTINUACIÓN.



EJEMPLOS

## ALGUNOS EJEMPLOS EN DÓNDE SE USAN:



PANTALLAS DEL METRO



ESTEREROS



ANUNCIOS



# CONCLUSION

COMO SE REALACIONA LA MATERIA CON EL PROYECTO?

- EN LA MATERIA VEMOS COMO FUNCIONAN VARIOS COMPONENTES DE LA COMPUTADORA Y SU FUNCION, COMO EN EL CASO DEL PROYECTO LOS MICROCONTROLADORE.
- ESTO NOS DA UNA IDEA DE LAS DISTINTAS FORMAS DE COMO LOS ELEMENTOS DE UNA COMPUTADORA PUEDEN SER OCUPADOS PARA REALISAR DISTINTAS FUNCIONES Y NO SOLO COMO UNA UNIDAD LOGICA ARITMETICA, COMO POR EJEMPLO EN EL PROYECTO QUE ACABAMOS DE PRESENTAR. TAMBIE NOS SIRVE PARA COMPRENDER COMO FUNCIONAN LOS MISMOS, YA QUE SI NO TUvierAMOS IDEA DE COMO FUNCIONAN NO SE NOS HUBIESE OCURRIDO RALIZAR UN PROYECTO BASADO EN ALGUN DISPOSITIVO O ELEMENTO DE LA COMPUTADORA.