

;muestra en LCD -MENSAJES en interfase de 4 bits- nombre del archivo lcd2011--version 19-9-11
;se usara 4 bits MSB del puertob para envio de datos a 4bits - RB4 RB5 RB6 RB7 respectivamente
conectados a D4 D5 D6 D7 y RB0 RB1 a RS y E respectivamente- RW va a masa.

;RECORDAR CAMBIAR DIRECCIONES DE REGISTROS DE PROPOSITO GENERAL A PARTIR DE
20h SI UTILIZA OTRO PIC

```
list    p=16f84
#include p16f84.inc
```

---Listado de registros especiales utilizado en este programa--

```
portb  equ    06h ;el puerto b esta en la posicion 06h de la ram
porta  equ    05h;el puerto b esta en la posicion 05h de la ram
trisa  equ    85h  ;regis de config de puerto b esta en la direcc 85h de la ram
trisb  equ    86h  ;regis de config de puerto b esta en la direcc 86h de la ram
status equ    03h ;registro de estado esta en la direcc 03h de la ram
```

---Fin de listado de registros especiales utilizado en este programa--

;Comienza listado de registros de proposito general empleados en este programa

-----registros de subrutina retardo-----
temp5 equ 20h

-----Fin registros de subrutina retardo-----

-----registros de subrutina INSTRUC/ESCRIB-----

```
aux      equ    21h
```

-----Fin registros de subrutina INSTRUC/ESCRIB-----

----- registros de subrutina espera y largaespera---(largaespera no es usada en este programa)----

```
tempo1   equ    26h
tempo2   equ    28h
tempo3   equ    2Ah
```

----- Fin registros de subrutina espera y largaespera-----

-----registro bandera para control reconocimieto positivo----(solo usamos el bit0)----

```
bandera  equ    2Bh
```

-----registro bandera para control reconocimieto positivo----(solo usamos el bit0)----

```
copiacodigo equ 2Ch
```

-----Fin del listado de registros de proposito general empleados en este programa-----

;-----INICIO programa principal-----
;////////////////////////////////////

inicio

bsf status,5 ;se posiciona en banco1
clrf trisb ;define el portb como salida
movlw b'00011111'
movwftriba;define al porta como entrada

bcf status,5; se posiciona en banco 0

call lcd; se llama subrutina de inicializacion del LCD (fundamental para configurarlo)

;-----

bcf bandera,0; hace 0 el bit0 de bandera para preparar para mensaje ACCESO DENEGADO si el reconocimiento ; es negativo

;-----

esperacodigo

btfsc porta,4; si se esta presionando ACCEDER entonces sigue analizando codigo (o sea si es 0 salta siguiente ;instruccion (no la hace)- No sigue haciendo analisis

goto analisis

call ingresecodigo; escribe IngresarCodigo en 2 lineas

;el programa espera un 1 en el pin RA4 para escribir mensaje1 o mensaje2 o mensaje3 ; (ud prodra agregar mas mensajes si lo desea)

btfsc porta,4; si en el pin 4 del porta hay un 0 entonces salta goto analisis (no lo ejecuta) ;y sigue esperando codigo-si hay un 1 entonces hace analisis de codigo (ejecuta ;la instruccion goto analisis).

goto analisis

goto esperacodigo

;-----Aqui comienza el analisis del codigo introducido-----

analisis

movf porta,0; mueve contenido de porta a W
andlw b'00001111'; hace AND entre el literal y W- Almacena resultado en W- Es una mascara ;para hacer cero todo lo que no es codigo a analizar.(importan solo los

pins (RA3-RA2-RA1-RA0)

movwf copiacodigo; mueve contenido de W a registro copiacodigo

;-----

;SE VERA SI SE TIENE QUE ESCRIBIR EL mensaje1 ASOCIADO AL CODIGO1 ("0001" en este ejemplo)

;RECUERDE QUE EL CODIGO ESTA EN ESTE MOMENTO GUARDADO EN copiacodigo

pruebacod1

movf copiacodigo,0; se mueve el contenido del copiacodigo a W

xorlw b'00000001'; prueba si el contenido de W coincide con "00000001", o sea el codigo "0001"

btfs status,2; prueba si el bit2 de status es 0 (bit Z de status) lo que implica que no hubo
;coincidencia (NO muestra mensaje1)- Pero si el bit 2 de status es 1 implica
;que si hubo coincidencia y MUESTRA mensaje1.

call mensaje1

;

;SE VERA SI SE TIENE QUE ESCRIBIR EL mensaje2 ASOCIADO AL CODIGO2 ("0010" en este ejemplo)
;RECUERDE QUE EL CODIGO ESTA EN ESTE MOMENTO GUARDADO EN copiacodigo

pruebacod2

movf copiacodigo,0; se mueve el contenido del copiacodigo a W

xorlw b'00000010'; prueba si el contenido de W coincide con "00000010", o sea el codigo "0010"

btfs status,2; prueba si el bit2 de status es 0 (bit Z de status) lo que implica que no hubo
;coincidencia (NO muestra mensaje2)- Pero si el bit 2 de status es 1 implica
;que si hubo coincidencia y MUESTRA mensaje2.

call mensaje2

;

;SE VERA SI SE TIENE QUE ESCRIBIR EL mensaje3 ASOCIADO AL CODIGO ("0011" en este ejemplo)
;RECUERDE QUE EL CODIGO ESTA EN ESTE MOMENTO GUARDADO EN copiacodigo

pruebacod3

movf copiacodigo,0; se mueve el contenido del copiacodigo a W (importan solo los pines (RA3-RA2-RA1-RA0))

xorlw b'00000011'; prueba si el contenido de W coincide con "00000011", o sea el codigo "0011"

btfs status,2; prueba si el bit2 de status es 0 (bit Z de status) lo que implica que no hubo
;coincidencia (NO muestra mensaje3)- Pero si el bit 2 de status es 1 implica
;que si hubo coincidencia y MUESTRA mensaje3.

call mensaje3

;

btfs bandera,0; prueba si es 1 el bit0 de bandera- si es 1 es porque se mostro mensajes positivos
;si es 0 el bit0 de bandera es porque el reconocimiento fue negativo

call denegado; este mensaje es de ACCESO DENEGADO O CODIGO ERRONEO

bcf bandera,0; hace 0 el bit0 de bandera para preparar para mensaje ACCESO DENEGADO si el reconocimiento
; es negativo

goto esperacodigo; reinicia la espera de codigo.

-----fin programa principal-----

////////////////////////////////////

;-----SUBRUTINAS-----

;-----Subrutina LCD -----

;esta subrutina inicia la pantalla LCD

;esta subrutina no utiliza registros de proposito general

lcd

movlw 02h; carga w con literal 02h RB1=1 o sea E=1 habilito LCD

call instruc; enviaremos este valor al portb y por lo tanto al LCD

;-----otra instruccion-----

movlw 28h; carga w con literal 28h

;la intrucc asociada dice: interfase de datos de 4 bits y se va a usar 2 lineas - corresponde a la intruccion ACTIVAR FUNCION

call instruc; enviaremos este valor al portb y por lo tanto al LCD

;-----otra instruccion-----

movlw 06h; carga w con literal 06h

;la intrucc asociada dice: dato fijo en pantalla- corresponde a la intruccion SELECCIONAR MODO

call instruc; enviaremos este valor al portb y por lo tanto al LCD

;-----otra instruccion-----

movlw 0Ch; carga w con literal 0Ch

;la intrucc asociada dice: encienda la pantalla y desactive el cursor- corresponde a la intruccion ENCENDER O APAGAR PANTALLA

call instruc; enviaremos este valor al portb y por lo tanto al LCD

;-----otra instruccion-----

movlw 01h; carga w con literal 01h

;la intrucc asociada dice: BORRAR PANTALLA

call instruc; enviaremos este valor al portb y por lo tanto al LCD

return

;esta subrutina inicio la pantalla LCD

;-----Fin Subrutina LCD -----

;-----Subrutina INSTRC/ESCRIB-----

;Subrutina que manda una instruccion o dato al LCD

;Esta subrutina utiliza el registro de proposito general aux

instruc

bcf portb,0; RB0=0 o sea RS=0 es una instruccion para LCD

goto dato2

escrib

bsf portb,0; RB0=1 o sea RS=1 es una dato para LCD

dato2

;lo que se va hacer es para no alterar el contenido de los 4 bit menos significativos dell portb

movwf aux; guarda contenido de w en aux

movlw 0fh; carga 00001111 en w

andwf portb,1; AND entre w=00001111 y portb y guarda resultado en portb

;lo que se termino de hacer es para no alterar el contenido de los 4 bit menos significativos dell portb

movf aux,0; carga aux en w

andlw b'11110000'; AND entre literal 11110000 y w en el cual esta copiado aux (guarda el resultado en w)

iorwf portb,1; OR entre w y portb - guarda resultado en portb

bsf portb,1; RB1=1 o sea E=1 habilito LCD

call retardo

bcf portb,1; RB1=E=0 deshabilita LCD

call retardo

;---terminó envio 4 bit MSB-----

;---comienza el envio de 4 bit LSB---

movlw 0fh; carga 00001111 en w

andwf portb,1; AND entre w=00001111 y portb - guarda resultado en portb

swapf aux,0; intercambia NIBLES de aux y lo guarda en w

andlw b'11110000'; AND entre literal 11110000 y w en el cual esta copiado aux con los NIBLES intercambiados (guarda el resultado en w)

iorwf portb,1; OR entre w y portb - guarda resultado en portb

;lo que se termino de hacer es para no alterar el contenido de los 4 bit menos significativos dell portb

bsf portb,1; RB1=1 o sea E=1 habilito LCD

call retardo

call retardo

bcf portb,1; RB1=E=0 deshabilita LCD

call retardo

;---termino el envio de lo 4bit LSB---

return

;-----Fin subrutina que manda instruccion o dato al LCD-----

;-----Fin Subrutina INSTRC/ESCRIB-----

;-----Subrutina retardo-----

;Esta subrutina utiliza el registro de proposito general temp5

retardo

movlw 0ffh; carga literal 0ffh en w

movwf temp5; mueve contenido de w a temp5

decr

decfsz temp5,1; decrementa temp5 y guarda resultado en temp5- salta la intruccion siguiente si temp5 es igual a cero

goto decr

movlw 0ffh; carga literal 0ffh en w

movwf temp5; mueve contenido de w a temp5

decr2

decfsz temp5,1; decrementa temp5 y guarda resultado en temp5- salta la intruccion siguiente si temp5 es igual a cero

goto decr2

return

;----- Fin Subrutina retardo-----

;-----Subrutina largaespera - -----

largaespera

;Escribe una E en display para indicar larga espera

movlw 0C7h;

call instruc

movlw "E";

call escrib

;Termino de escribir la E

```

retardo4
    incf    tempo1,1
retardo5
    incf    tempo2,1
retardo6
    incf    tempo3,1
    btfss  tempo3,7
    goto   retardo6
    clrf   tempo3
    btfss  tempo2,7
    goto   retardo5
    clrf   tempo2
    btfss  tempo1,7
    goto   retardo4
    clrf   tempo1

```

;Llama reiteradas veces a subrutina espera -para incrementar espera
;Tanto como subrutina espera y largaespera utilizan los mismos registros de proposito general

```

    call espera
    call espera
; Termino de llamar reiteradas veces a subrutina espera

```

```

;Limpia la E en display que escribio al principio
    movlw 0C7h;
    call instruc
    movlw " ";
    call  escrib
;Termino de limpiar la E

```

```

    return
;-----Fin subrutina largaespera-----

```

;-----Subrutina espera -(Retardo general)-----
;Esta subrutina utiliza los registros de proposito general- temp1-temp2-temp3
;Son los mismos que utiliza la subrutina largaespera

espera nop

```

retardo1
    incf    tempo1,1
retardo2
    incf    tempo2,1
retardo3
    incf    tempo3,1
    btfss  tempo3,6
    goto   retardo3
    clrf   tempo3
    btfss  tempo2,5
    goto   retardo2
    clrf   tempo2
    btfss  tempo1,4
    goto   retardo1
    clrf   tempo1
    return

```

;-----Fin subrutina espera-----

;---Subrutina ingresecodigo-----IngresarCodigo-----

ingresecodigo

```

;-----
    bcf    portb,3; hace 0 pin RB3 de portb para apagar led verde- encender led rojo y encender motor
;-----
    movlw 01h

```

```
call   instruc; borrar pantalla
```

```
movlw 80h; se indica posicion de memoria del LCD en que quiere escribir el dato
```

```
call   instruc  
movlw "I";  
call   escrib
```

```
movlw 81h;  
call   instruc  
movlw "N";  
call   escrib
```

```
;repito con el resto de los caracteres
```

```
movlw 82h;  
call   instruc  
movlw "G";  
call   escrib
```

```
movlw 83h;  
call   instruc  
movlw "R";  
call   escrib
```

```
movlw 84h;  
call   instruc  
movlw "E";  
call   escrib
```

```
movlw 85h;  
call   instruc  
movlw "S";  
call   escrib
```

```
movlw 86h;  
call   instruc  
movlw "E";  
call   escrib
```

```
movlw 87h;  
call   instruc  
movlw " ";  
call   escrib
```

```
;---FIN DE PRIMERA LINEA--- PALABRA INGRESE---
```

```
bcf    portb,2;hace 0 pin rb2 de portb - Apaga el BUZER
```

```
;---ESCRIBO SEGUNDA LINEA--- PALABRA CODIGO---
```

```
movlw 0C0h; el cero adelante de debe colocar para que no de error el compilador
```

```
call   instruc  
movlw "C";  
call   escrib
```

```
movlw 0C1h;  
call   instruc  
movlw "O";  
call   escrib
```

```
movlw 0C2h;  
call   instruc  
movlw "D";
```

```
call  escrib
```

```
movlw 0C3h;  
call instruc  
movlw "I";  
call  escrib
```

```
movlw 0C4h;  
call instruc  
movlw "G";  
call  escrib
```

```
movlw 0C5h;  
call instruc  
movlw "O";  
call  escrib
```

```
movlw 0C6h;  
call instruc  
movlw ".";  
call  escrib
```

```
;---FIN DE SEGUNDA LINEA--- PALABRA CODIGO---
```

```
call  espera  
call  espera  
call  espera  
call  espera
```

```
return
```

```
;---Fin subrutina ingresecodigo-----
```

```
;---Subrutina mensaje1-----PEREZ-----
```

```
mensaje1
```

```
movlw 01h  
call  instruc; borrar pantalla
```

```
movlw 80h; se indica posicion de memoria del LCD en que quiere escribir el dato
```

```
call instruc
```

```
movlw "P"; este es el primer caracter del mensaje - al estar entre comillas se le indica al compilador  
que el dato requerido s el valo ASCII del caracter
```

```
call  escrib
```

```
;repito con el resto de los caracteres
```

```
movlw 81h;  
call instruc  
movlw "E";  
call  escrib
```

```
movlw 82h;  
call instruc  
movlw "R";  
call  escrib
```

```
movlw 83h;  
call instruc  
movlw "E";  
call  escrib
```



```
movlw 84h;
call instruc
movlw "Z";
call  escrib
```

```
movlw 85h;
call instruc
movlw ".";
call  escrib
```

```
movlw 86h;
call instruc
movlw ".";
call  escrib
```

```
movlw 87h;
call instruc
movlw ".";
call  escrib
```

```
movlw 88h;
call instruc
movlw ".";
call  escrib
```

```
movlw 89h;
call instruc
movlw ".";
call  escrib
```

```
movlw 8Ah;
call instruc
movlw ".";
call  escrib
```

```
movlw 8Bh;
call instruc
movlw ".";
call  escrib
```

```
bsf  bandera,0; hace 1 bit0 bandera de para informar que el reconocimiento fue positivo
```

```
call  espera
call  espera
call  espera
call  espera
```

```
call OKPASE
```

```
return
```

```
;---Fin subrutina mensaje1-----
```

```
;---Subrutina mensaje2-----MORALES----
```

```
mensaje2
movlw 01h
call  instruc; borrar pantalla
```

movlw 80h; se indica posicion de memoria del LCD en que quiere escribir el dato

call instruc

movlw "M"; este es el primer caracter del mensaje - al estar entre comillas se le indica al compilador que el dato requerido es el valor ASCII del caracter

call escrib

;repite con el resto de los caracteres

movlw 81h;
call instruc
movlw "O";
call escrib

movlw 82h;
call instruc
movlw "R";
call escrib

movlw 83h;
call instruc
movlw "A";
call escrib

movlw 84h;
call instruc
movlw "L";
call escrib

movlw 85h;
call instruc
movlw "E";
call escrib

movlw 86h;
call instruc
movlw "S";
call escrib

movlw 87h;
call instruc
movlw ".";
call escrib

movlw 88h;
call instruc
movlw ".";
call escrib

movlw 89h;
call instruc
movlw ".";
call escrib

movlw 8Ah;
call instruc
movlw ".";
call escrib

movlw 8Bh;
call instruc
movlw ".";

```
call  escrib
```

```
bsf   bandera,0; hace 1 bit0 bandera de para informar que el reconocimiento fue positivo
```

```
call  espera
```

```
call  espera
```

```
call  espera
```

```
call  espera
```

```
call OKPASE
```

```
return
```

```
;---Fin subrutina mensaje2-----
```

```
;---Subrutina mensaje3-----LOPEZ-----
```

```
mensaje3
```

```
movlw 01h
```

```
call  instruc; borrar pantalla
```

```
movlw 80h; se indica posicion de memoria del LCD en que quiere escribir el dato
```

```
call instruc
```

```
movlw "L"; este es el primer caracter del mensaje - al estar entre comillas se le indica al compilador que el dato requerido s el valo ASCII del caracter
```

```
call  escrib
```

```
;repito con el resto de los caracteres
```

```
movlw 81h;
```

```
call instruc
```

```
movlw "O";
```

```
call  escrib
```

```
movlw 82h;
```

```
call instruc
```

```
movlw "P";
```

```
call  escrib
```

```
movlw 83h;
```

```
call instruc
```

```
movlw "E";
```

```
call  escrib
```

```
movlw 84h;
```

```
call instruc
```

```
movlw "S";
```

```
call  escrib
```

```
movlw 85h;
```

```
call instruc
```

```
movlw ".";
```

```
call  escrib
```

```
movlw 86h;
```

```
call instruc
```

```
movlw ".";
```

```
call  escrib
```

```
movlw 87h;
```

```
call instruc
```

```
movlw ".";
call  escrib
```

```
movlw 88h;
call instruc
movlw ".";
call  escrib
```

```
movlw 89h;
call instruc
movlw ".";
call  escrib
```

```
movlw 8Ah;
call instruc
movlw ".";
call  escrib
```

```
movlw 8Bh;
call instruc
movlw ".";
call  escrib
```

```
bsf  bandera,0; hace 1 bit0 bandera de para informar que el reconocimiento fue positivo
```

```
call  espera
call  espera
call  espera
call  espera
```

```
call OKPASE
```

```
return
```

```
;---Fin subrutina mensaje3-----
```

```
;---Subrutina denegado-----ACCESO DENEGADO-----
```

```
denegado
```

```
;-----
```

```
bsf  portb,2;hace 1 pin rb2 de portb para encender BUZER
```

```
;-----
```

```
movlw 01h
call  instruc; borrar pantalla
movlw 80h; se indica posicion de memoria del LCD en que quiere escribir el dato
call instruc
movlw "A";
call  escrib
```

```
movlw 81h;
call instruc
movlw "C";
call  escrib
```

```
;repite con el resto de los caracteres
```

```
movlw 82h;
call instruc
movlw "C";
call  escrib
```

```
movlw 83h;
call instruc
movlw "E";
```

```
call  escrib
```

```
movlw 84h;  
call instruc  
movlw "S";  
call  escrib
```

```
movlw 85h;  
call instruc  
movlw "O";  
call  escrib
```

```
movlw 86h;  
call instruc  
movlw " ";  
call  escrib
```

```
movlw 87h;  
call instruc  
movlw " ";  
call  escrib
```

;---FIN DE PRIMERA LINEA--- PALABRA ACCESO---

;---ESCRIBO SEGUNDA LINEA--- PALABRA DENEGADO---

```
movlw 0C0h; el cero adelante de debe colocar para que no de error el compilador  
call instruc  
movlw "N";  
call  escrib
```

```
movlw 0C1h;  
call instruc  
movlw "E";  
call  escrib
```

```
movlw 0C2h;  
call instruc  
movlw "G";  
call  escrib
```

```
movlw 0C3h;  
call instruc  
movlw "A";  
call  escrib
```

```
movlw 0C4h;  
call instruc  
movlw "D";  
call  escrib
```

```
movlw 0C5h;  
call instruc  
movlw "O";  
call  escrib
```

```
movlw 0C6h;  
call instruc  
movlw ".";  
call  escrib
```

;---FIN DE SEGUNDA LINEA--- PALABRA DENEGADO---

```
bsf  bandera,0; hace 1 bit0 bandera de para informar que el reconocimiento fue positivo
```

```
call  espera
call  espera
call  espera
call  espera
```

```
return
```

```
;---Fin subrutina denegado-----
```

```
;---Subrutina OKPASE-----
```

```
OKPASE
```

```
;-----
```

```
    bsf    portb,3; hace 1 pin RB3 de portb para encender led verde- apagar led rojo y apagar motor
```

```
;-----
```

```
;---ESCRIBO EN SEGUNDA LINEA-----
```

```
    movlw 0C0h; el cero adelante de debe colocar para que no de error el compilador
```

```
    call  instruc
    movlw "O";
    call  escrib
```

```
    movlw 0C1h;
    call  instruc
    movlw "K";
    call  escrib
```

```
    movlw 0C2h;
    call  instruc
    movlw "-";
    call  escrib
```

```
    movlw 0C3h;
    call  instruc
    movlw "P";
    call  escrib
```

```
    movlw 0C4h;
    call  instruc
    movlw "A";
    call  escrib
```

```
    movlw 0C5h;
    call  instruc
    movlw "S";
    call  escrib
```

```
    movlw 0C6h;
    call  instruc
    movlw "E";
    call  escrib
```

```
call  espera
call  espera
call  espera
call  espera
```

```
return
```

```
;---Fin subrutina ingresecodigo-----
```

```
;-----FIN SUBRUTINAS-----
```

```
end
```