

ESCUELA SUPERIOR POLITÉCNICA DEL
LITORAL

FIEC

Laboratorio de Microcontroladores

Proyecto
ENCRİPTADOR DE SEÑALES

Alumno
Ángel Guale

Paralelo: #6

Grupo: #2

Fecha de presentación:
28 de Diciembre del 2015

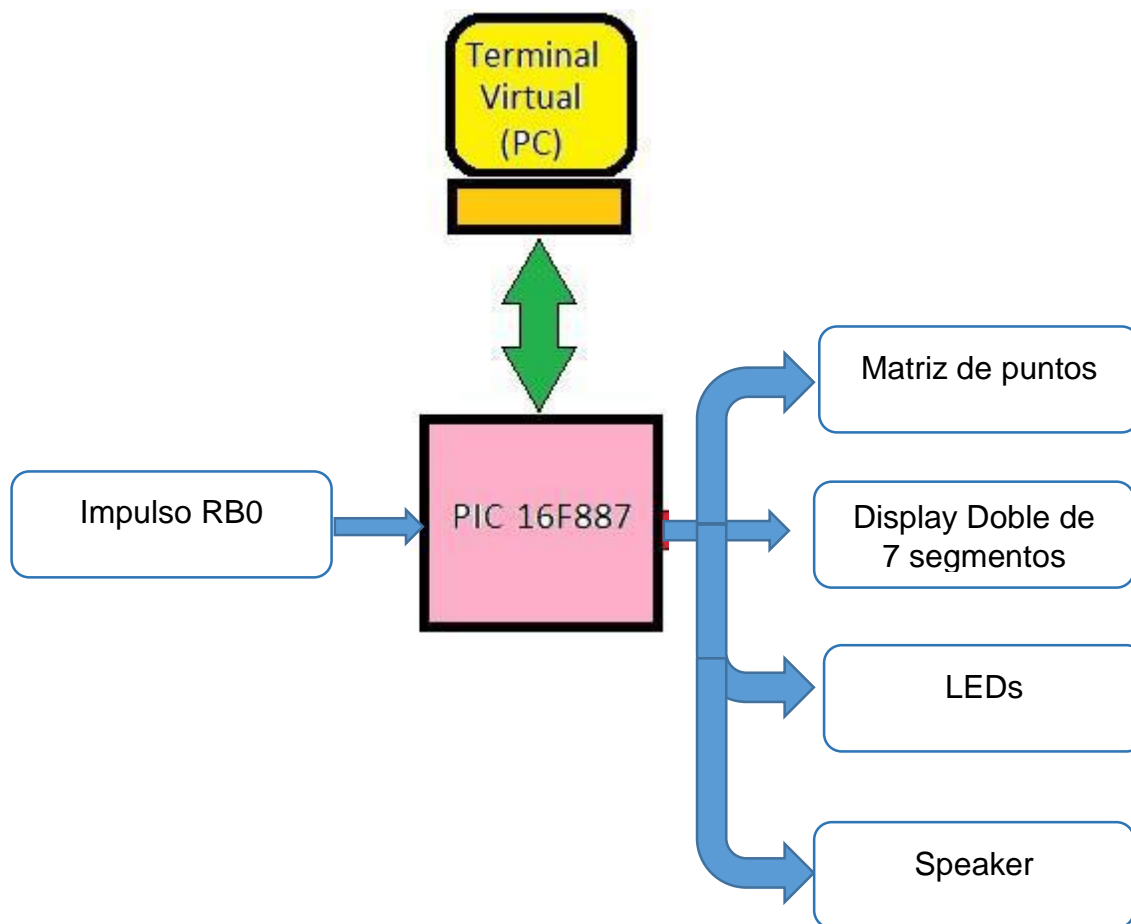
2015 – 2° TÉRMINO

1.- Enunciado del proyecto

Se está recibiendo una señal en formato binario de 8 bits en un puerto entrada del PIC, esta debe ser encriptada y enviada hacia otro punto de operación (puerto salida del PIC) cuando el operador presione Recibir, se debe encriptar la señal original multiplicándola por 16 o desplazada a la derecha 3 espacios o invertida (ejemplo: 10010110 → 01101001), esta selección es aleatoria, cuando la encriptación este lista se debe generar un sonido por 10 segundos (el sonido debe de corresponder a la encriptación) y mostrar el dato en la salida, además se debe contar el número de mensajes que han llegado con un display doble, se debe mostrar en la matriz de puntos el dato que llego antes de su encriptación en formato Hexadecimal de ser necesario desplazándose de derecha a izquierda, hasta que llegue otro dato (no se pueden ingresar datos durante el proceso de encriptación).

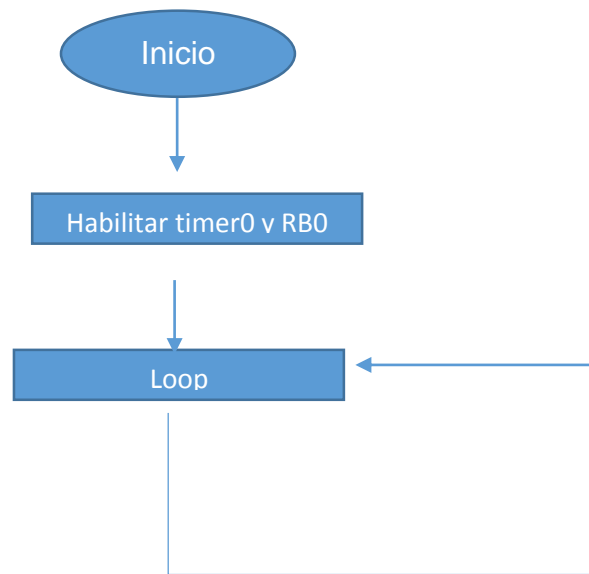
El proceso se reinicia cuando el usuario presione el botón de reset.

2.- Diagrama de Bloques

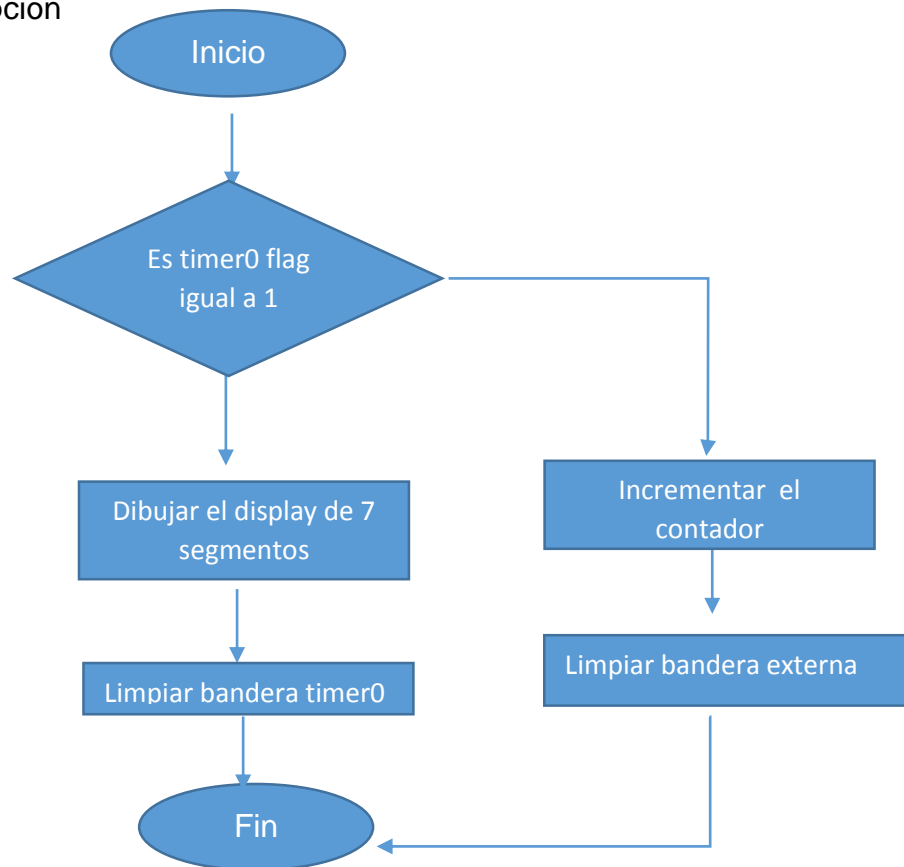


3.- Diagrama de flujo funcional del Programa

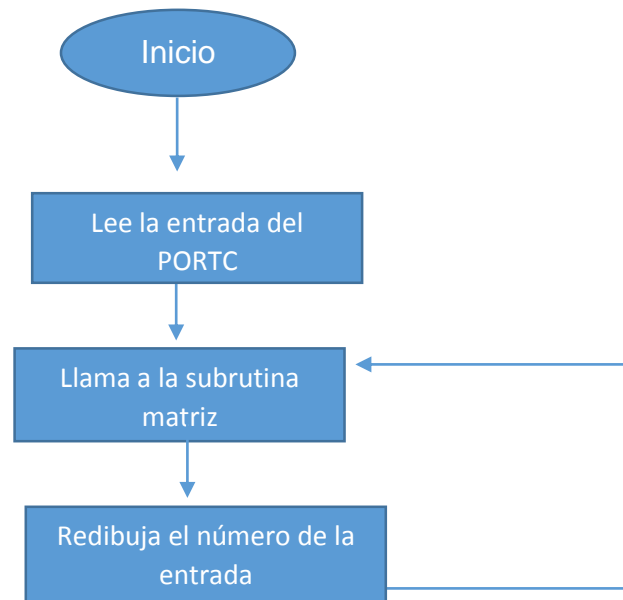
Principal



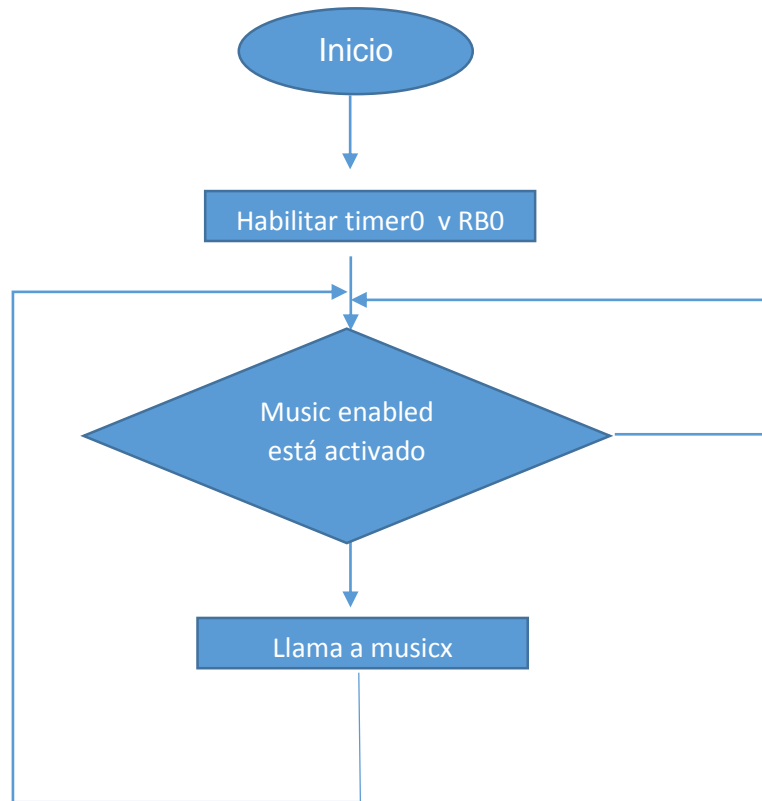
Subrutina: Interrupción



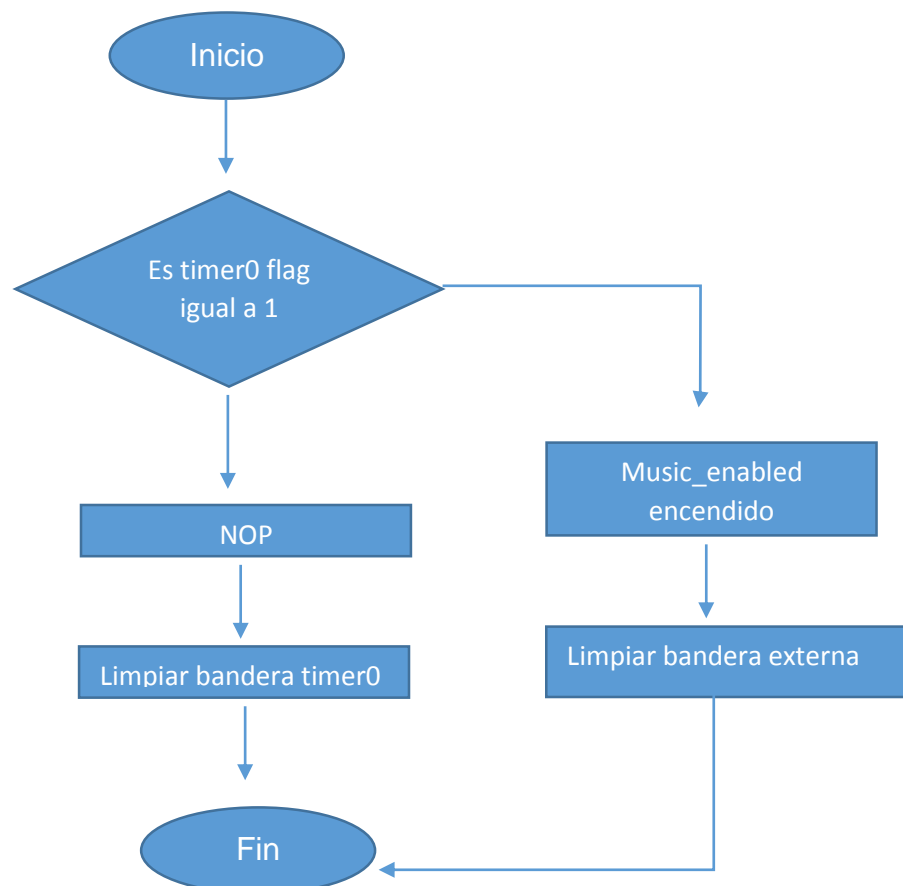
Principal



Principal



Subrutina: Interrupción



4.- Descripción del algoritmo o estrategia utilizada

Algoritmo Principal

1. Se inicializa el PIC con una frecuencia de reloj de 4MHZ.
2. Se habilita la interrupción externa por el pin 0 del puerto B.
3. Si se envía un impulso por RB0 se ejecuta la rutina de interrupción externa
4. Una vez que la subrutina de interrupción es ejecutada, se habilita de nuevo la interrupción externa y se limpia la bandera INTF

Interrupción

1. Si se activa la bandera INTF se suma uno al contador de mensajes del display doble de 7 segmentos.
2. Se obtiene la entrada de la señal ingresada por el usuario y se lo muestra en la matriz de puntos en formato hexadecimal.
3. Con la entrada ingresada por el usuario se aplica la subrutina de encriptar, ya sea multiplicar por 16 o multiplicar por cuatro o invertida.
4. Se ejecuta la reproducción de un sonido al momento de encriptar.

5.- Listado del programa fuente

```
;*****PIC1*****+
;*****HEADER*****
;*****

LIST      p=16F887
INCLUDE   P16F887.INC
_CONFIG _CONFIG1, _CP_OFF&_WDT_OFF&_XT_OSC
errorlevel -302
;*****
; bloque de variables
;*****

cblock 0x20
    counter1
    counter2
    counter3
    num_msj

    unidades
    uni_cod
    decenas
    dec_cod
    sel

endc

;*****START OF PROGRAM *****
; forma de iniciar programa que usa interrupciones
;*****

Org      0x00      ; vector de reset
Goto     main      ; salto a label "main"
Org      0x04      ; vector de interrupción
Goto     inter      ; salto a interrupción
org       0x05      ; continuación de programa

;***** MAIN PROGRAM *****
; inicio de programa principal
;*****
;SETEO DE PUERTOS Y REGISTROS
main
    banksel    ANSEL      ;Bank containing register ANSEL
    clrf       ANSEL      ;Clears registers ANSEL and ANSELH
    clrf       ANSELH     ;All pins are digital
```



```

    banksel      TRISC ;Selects bank containing register TRISB
    movlw b'11111111'
    movwf        TRISC ;All port C pins are configured as outputs
    clrf         TRISD ;All port D pins are configured as outputs
    banksel TRISA
    clrf         TRISA
    bsf          TRISB,0      ;PORTB,0 configurado como entrada
banksel TRISE
    clrf         TRISE

    banksel      OPTION_REG ; Bank containing register OPTION_REG
    movlw        b'00000111' ;carga divisor con 255, se lo aplica a
;TMR0
                                ;PSA =0 (BIT 3); se aplica el divisor
;al TMR0
                                ;TOCS=0 (BIT 5); TMR0 origen de pulsos
;Fosc/4
    movwf        OPTION_REG

    movlw        b'10110000' ;habilita interrupción por Timer 0 y Global
                                ;GIE=1 (BIT 7); habilita interrupciones globales
                                ;TMR0IE=1 (BIT 5); habilita interrupciones por TMR0
                                ;INTE=1 (BIT 4); habilita interrupciones por RB0
    movwf        INTCON
    clrf         counter1; variable de conteo de mensajes
    banksel      PORTC      ; Bank containing register PORTC
    clrf         PORTC      ; Clears Port C
    clrf         PORTD      ; Clears Port D
    clrf         PORTA
    incf         PORTD
    bsf PORTE,0
    clrf num_msj
    clrf unidades
    clrf decenas
    clrf uni_cod
    clrf dec_cod
    clrf sel

;PROGRAMACION DEL TMR0
;banksel      OPTION_REG ;Selecciona el Bank1
;movlw        b'00000111' ;TMR0 como temporizador
;movwf        OPTION_REG ;con preescaler de 256
BANKSEL TMR0      ;Selecciona el Bank0
movlw .217        ;Valor decimal 217
movwf TMR0        ;Carga el TMR0 con 217

```

loop

```
        nop
    goto loop        ; Permanece en el lazo
```

binario_a_s_segmentos ; Tabla para display de 7 segmentos.

```
    addwf PCL,F
    ;'-gfedcba'        segmentos
    retlw b'00111111'        ; El código 7 segmentos para el "0".
    retlw b'00000110'        ; El código 7 segmentos para el "1".
    retlw b'01011011'        ; El código 7 segmentos para el "2".
    retlw b'01001111'        ; El código 7 segmentos para el "3".
    retlw b'01100110'        ; El código 7 segmentos para el "4".
    retlw b'01101101'        ; El código 7 segmentos para el "5".
    retlw b'01111101'        ; El código 7 segmentos para el "6".
    retlw b'00000111'        ; El código 7 segmentos para el "7".
    retlw b'01111111'        ; El código 7 segmentos para el "8".
    retlw b'01100111'        ; El código 7 segmentos para el "9".
    retlw b'01110111'        ; El código 7 segmentos para el "A".
    retlw b'01111100'        ; El código 7 segmentos para el "B".
    retlw b'00111001'        ; El código 7 segmentos para el "C".
    retlw b'01011110'        ; El código 7 segmentos para el "D".
    retlw b'01111001'        ; El código 7 segmentos para el "E".
    retlw b'01110001'        ; El código 7 segmentos para el "F".
```

```
;***** INTERRUPT ROUTINE *****
; Inicio de rutina de interrupción llamado desde org 0x04
;*****
```

inter

```
    ;movlw        .20
    ;movwf        counter3
    btfss        INTCON,TMR0IF
    goto         externo
        ;*****
    movf    sel,w        ;Se mueve a si mismo para afectar bandera
    btfss    STATUS,2    ;sel=0 refresca dig1; sel=1 refresca dig2
    goto    dig2
```

dig1

```
    movf    unidades,w
    call    binario_a_s_segmentos
    movwf    uni_cod
    movf    uni_cod,w
    bsf     PORTA,0
    bsf     PORTA,1
    movwf    PORTD
    bcf     PORTA,0
```

```

        comf    sel,f
        goto    dec

dig2
        movf    decenas,w
        call    binario_a_s_segmentos
        movwf   dec_cod
        movf    dec_cod,w
        bsf     PORTA,0
        bsf     PORTA,1
        movwf   PORTD
        bcf     PORTA,1
        comf    sel,f
        ;;;;

dec

; incf        PORTC        ; Increments register PORTC by 1
bcf          INTCON,TMR0IF    ; Clears interrupt flag TMR0IF
        movlw   ~.39
        movwf   TMR0          ; Repone el TMR0 con ~.39
        goto    Seguir

externo
        movfw   PORTC
        ;aqui falta la funcion de encriptacion
        ;call encripta_x_4
        ;incf num_msj
        ;movfw num_msj
        ;call binario_a_s_segmentos
        ;bsf PORTA,0
        ;bcf PORTA,1

        ;movwf PORTD
        ;;;;incrementa el numero de mensajes;;;;;
        INCF    unidades,f    ;Ahora sí 10x100=1000ms=1seg
        movlw   .10
        subwf   unidades,w
        btfss   STATUS,2
        goto    cont
        clrf    unidades
        incf    decenas
        movlw   .10
        subwf   decenas,w
        btfss   STATUS,2
        goto    cont
        clrf    decenas
        ;;;;incrementa el numero de mensajes

cont

```

bcf

INTCON,INTF

Seguir

retfie

```

; Return from interrupt routine

```

; IMPORTANTE: rutina de interrupción termina en retfie

```

;***** SUBROUTINES *****

```

End

; Fin de programa

;***PIC2*******

LIST p=16F887
INCLUDE P16F887.INC
_CONFIG _CONFIG1, _CP_OFF&_WDT_OFF&_INTOSCIO
errorlevel -302

;*****

CBLOCK 0X020
contador
unidades
uni_cod
decenas
dec_cod
d1
d2
d3
f1
f2
temp1ms
sel
dcont
o
m
n
terminoD
tsupp
tinf
aux
ENDC

;*****

;PROGRAMA

ORG 0x00
GOTO MAIN
ORG 0x04
GOTO Interrupcion

tabla2

clrf aux
movwf aux
movlw .15
subwf aux,w
btfsc STATUS,2
goto quince_F
movlw .14
subwf aux,w
btfsc STATUS,2

```
goto    catorce_E
movlw   .13
subwf   aux,w
btfsc   STATUS,2
goto    trece_D
movlw   .12
subwf   aux,w
btfsc   STATUS,2
goto    doce_C
movlw   .11
subwf   aux,w
btfsc   STATUS,2
goto    once_B
movlw   .10
subwf   aux,w
btfsc   STATUS,2
goto    diez_A
movlw   .9
subwf   aux,w
btfsc   STATUS,2
goto    nueve
movlw   .8
subwf   aux,w
btfsc   STATUS,2
goto    ocho
movlw   .7
subwf   aux,w
btfsc   STATUS,2
goto    siete
movlw   .6
subwf   aux,w
btfsc   STATUS,2
goto    seis
movlw   .5
subwf   aux,w
btfsc   STATUS,2
goto    cinco
movlw   .4
subwf   aux,w
btfsc   STATUS,2
goto    cuatro
movlw   .3
subwf   aux,w
btfsc   STATUS,2
goto    tres
movlw   .2
subwf   aux,w
```

```

    btfsc STATUS,2
    goto dos
    movlw .1
    subwf aux,w
    btfsc STATUS,2
    goto uno
    movf aux,f
    btfsc STATUS,2
    goto cero
    return

```

tabla

```

    ADDWF PCL,F
    RETLW 0x3F
    RETLW0x06
    RETLW0x5B
    RETLW0x4F
    RETLW0x66
    RETLW0x6D
    RETLW0x7D
    RETLW0x07
    RETLW0x7F
    RETLW0x67

```

Interrupcion

```

    movf sel,w
    btfss STATUS,2
    goto dig2

```

dig1

```

    movf unidades,w
    call tabla
    movwf uni_cod
    movf uni_cod,w
    bsf     PORTE,0
    bsf     PORTE,1
    movwf PORTB
    bcf     PORTE,0
    comf sel,f
    goto Seguir

```

dig2

```

    movf decenas,w
    call tabla
    movwf dec_cod
    movf dec_cod,w
    bsf     PORTE,0
    bsf     PORTE,1
    movwf PORTB

```

```
bcf      PORTE,1
comf    sel,f
```

Seguir

```
bcf      INTCON,T0IF
movlw   ~.39
movwf   TMR0
retfie
```

MAIN

;SETEO DE PUERTOS

```
BANKSEL  ANSEL
CLRF     ANSEL
CLRF     ANSELH
BANKSEL  TRISD
movlw    0xFF
movwf    TRISD
CLRF     TRISA
clrf     TRISB

CLRF     TRISC
CLRF     TRISE
```

;INICIALIZACION

```
BANKSEL  PORTD
CLRF     PORTA
CLRF     PORTB
CLRF     PORTC
CLRF     PORTE
clrf     unidades
clrf     decenas
clrf     sel
clrf     terminoD
```

;PROGRAMACION DEL TMR0

```
banksel  OPTION_REG
movlw    b'00000111'
movwf    OPTION_REG
BANKSEL  TMR0
movlw    .217
movwf    TMR0
clrf     contador
movlw    .30
movwf    dcont
movlw    b'10000000'
movwf    INTCON
```


entrada

```
CLRF PORTA  
CLRF PORTC
```

loop

```
call matrix  
;call click  
;call delay250ms  
goto entrada
```

matrix

```
movlw b'11110000'  
;movf PORTD,0  
movwf terminoD  
movwf tsupp  
;movwf tinf  
call termino_superior  
;call termino_inferior  
movf tsupp,0  
;movlw b'00001111'  
;movwf PORTB
```

dibuja

```
movfw PORTD  
call tabla2  
goto dibuja
```

return

termino_superior

```
bcf STATUS,0  
rrf tsupp  
bcf STATUS,0  
rrf tsupp  
bcf STATUS,0  
rrf tsupp  
bcf STATUS,0  
rrf tsupp  
return
```

termino_inferior

```
movlw b'00001111'  
andwf tinf,f  
return
```

click

```
INCF unidades,f  
movlw .10  
subwf unidades,w
```

```

btfss STATUS,2
return
clrf unidades
incf decenas
movlw .10
subwf decenas,w
btfss STATUS,2
return
clrf decenas
return

```

; TABLA DE CONVERSION-----

cero

```

bcf STATUS,2
movlw b'01111101'
movwf PORTC
movlw b'00111000'
movwf PORTA
call simple_delay

movlw b'10000011'
movwf PORTC
movlw b'01000100'
movwf PORTA
call simple_delay

decfsz dcont,1
goto cero

movlw .20
movwf dcont
return

```

uno

```

bcf STATUS,2
movlw b'11111101'
movwf PORTC
movlw b'00001000'
movwf PORTA
call simple_delay

movlw b'11111011'
movwf PORTC
movlw b'00011000'
movwf PORTA
call simple_delay

```

```
movlw b'11110111'  
movwf PORTC  
movlw b'00101000'  
movwf PORTA  
call  simple_delay
```

```
movlw b'10001111'  
movwf PORTC  
movlw b'00001000'  
movwf PORTA  
call  simple_delay
```

```
movlw b'01111111'  
movwf PORTC  
movlw b'000111100'  
movwf PORTA  
call  simple_delay
```

```
decfsz dcont,1  
goto  uno
```

```
movlw .20  
movwf dcont  
return
```

dos

```
bcf      STATUS,2  
movlw b'01101101'  
movwf PORTC  
movlw b'00111100'  
movwf PORTA  
call  simple_delay
```

```
movlw b'11110011'  
movwf PORTC  
movlw b'00000100'  
movwf PORTA  
call  simple_delay
```

```
movlw b'10011111'  
movwf PORTC  
movlw b'00100000'  
movwf PORTA  
call  simple_delay
```

```
decfsz dcont,1  
goto dos
```

```
movlw .20  
movwf dcont  
return
```

tres

```
bcf STATUS,2  
movlw b'01101101'  
movwf PORTC  
movlw b'00111100'  
movwf PORTA  
call simple_delay
```

```
movlw b'10010011'  
movwf PORTC  
movlw b'00000100'  
movwf PORTA  
call simple_delay
```

```
decfsz dcont,1  
goto tres
```

```
movlw .20  
movwf dcont  
return
```

cuatro

```
bcf STATUS,2  
movlw b'11110001'  
movwf PORTC  
movlw b'00100100'  
movwf PORTA  
call simple_delay
```

```
movlw b'11101111'  
movwf PORTC  
movlw b'00111100'  
movwf PORTA  
call simple_delay
```

```
movlw b'00011111'  
movwf PORTC  
movlw b'00000100'
```

```
movwf PORTA
call  simple_delay
```

```
decfsz dcont,1
goto  cuatro
```

```
movlw .20
movwf dcont
return
```

cinco

```
bcf          STATUS,2
movlw b'01101101'
movwf PORTC
movlw b'00111100'
movwf PORTA
call  simple_delay
```

```
movlw b'11110011'
movwf PORTC
movlw b'00100000'
movwf PORTA
call  simple_delay
```

```
movlw b'10011111'
movwf PORTC
movlw b'00000100'
movwf PORTA
call  simple_delay
```

```
decfsz dcont,1
goto  cinco
```

```
movlw .20
movwf dcont
return
```

seis

```
bcf          STATUS,2
movlw b'01101101'
movwf PORTC
movlw b'00111100'
movwf PORTA
call  simple_delay
```

```
movlw b'11110011'
movwf PORTC
```

```
movlw b'00100000'  
movwf PORTA  
call    simple_delay
```

```
movlw b'10011111'  
movwf PORTC  
movlw b'00100100'  
movwf PORTA  
call    simple_delay
```

```
decfsz dcont,1  
goto    seis
```

```
movlw .20  
movwf dcont  
return
```

siete

```
bcf          STATUS,2  
movlw b'11111101'  
movwf PORTC  
movlw b'01111000'  
movwf PORTA  
call    simple_delay
```

```
movlw b'00010011'  
movwf PORTC  
movlw b'00001000'  
movwf PORTA  
call    simple_delay
```

```
movlw b'11101111'  
movwf PORTC  
movlw b'00011100'  
movwf PORTA  
call    simple_delay
```

```
decfsz dcont,1  
goto    siete
```

```
movlw .20  
movwf dcont  
return
```

ocho

```
bcf          STATUS,2  
movlw b'01101101'
```

```
movwf PORTC
movlw b'00111000'
movwf PORTA
call    simple_delay
```

```
movlw b'10010011'
movwf PORTC
movlw b'01000100'
movwf PORTA
call    simple_delay
```

```
decfsz dcont,1
goto   ocho
```

```
movlw .20
movwf dcont
return
```

nueve

```
bcf          STATUS,2
movlw b'01101101'
movwf PORTC
movlw b'00111100'
movwf PORTA
call    simple_delay
```

```
movlw b'11110011'
movwf PORTC
movlw b'00100100'
movwf PORTA
call    simple_delay
```

```
movlw b'00011111'
movwf PORTC
movlw b'00000100'
movwf PORTA
call    simple_delay
```

```
decfsz dcont,1
goto   nueve
```

```
movlw .20
movwf dcont
return
```

diez_A

```
bcf          STATUS,2
movlw b'11111101'
movwf PORTC
movlw b'00111000'
movwf PORTA
call  simple_delay
```

```
movlw b'00010011'
movwf PORTC
movlw b'01000100'
movwf PORTA
call  simple_delay
```

```
movlw b'11101111'
movwf PORTC
movlw b'01111100'
movwf PORTA
call  simple_delay
```

```
decfsz dcont,1
goto  diez_A
```

```
movlw .20
movwf dcont
return
```

once_B

```
bcf          STATUS,2
movlw b'01101101'
movwf PORTC
movlw b'01111000'
movwf PORTA
call  simple_delay
```

```
movlw b'10010011'
movwf PORTC
movlw b'01100100'
movwf PORTA
call  simple_delay
```

```
decfsz dcont,1
goto  once_B
```

```
movlw .20
movwf dcont
return
```


doce_C

```
bcf          STATUS,2  
movlw b'01111101'  
movwf PORTC  
movlw b'00011100'  
movwf PORTA  
call  simple_delay
```

```
movlw b'10111011'  
movwf PORTC  
movlw b'00100000'  
movwf PORTA  
call  simple_delay
```

```
movlw b'11000111'  
movwf PORTC  
movlw b'01000000'  
movwf PORTA  
call  simple_delay
```

```
decfsz dcont,1  
goto  doce_C
```

```
movlw .20  
movwf dcont  
return
```

trece_D

```
bcf          STATUS,2  
movlw b'01111101'  
movwf PORTC  
movlw b'01111000'  
movwf PORTA  
call  simple_delay
```

```
movlw b'10111011'  
movwf PORTC  
movlw b'01100100'  
movwf PORTA  
call  simple_delay
```

```
movlw b'11000111'  
movwf PORTC  
movlw b'01100010'
```

```
movwf PORTA
call    simple_delay
```

```
decfsz dcont,1
goto    trece_D
```

```
movlw .20
movwf dcont
return
```

catorce_E

```
bcf          STATUS,2
movlw b'01111101'
movwf PORTC
movlw b'01111100'
movwf PORTA
call    simple_delay
```

```
movlw b'10000011'
movwf PORTC
movlw b'01100000'
movwf PORTA
call    simple_delay
```

```
movlw b'11101111'
movwf PORTC
movlw b'01111100'
movwf PORTA
call    simple_delay
```

```
decfsz dcont,1
goto    catorce_E
```

```
movlw .20
movwf dcont
return
```

quince_F

```
bcf          STATUS,2
movlw b'11111101'
movwf PORTC
movlw b'01111100'
movwf PORTA
call    simple_delay
```

```
movlw b'00000011'  
movwf PORTC  
movlw b'01100000'  
movwf PORTA  
call simple_delay
```

```
movlw b'11101111'  
movwf PORTC  
movlw b'01111000'  
movwf PORTA  
call simple_delay
```

```
decfsz dcont,1  
goto quince_F
```

```
movlw .20  
movwf dcont  
;goto dibuja  
return
```

delay1seg

```
movlw 0x07  
movwf d1  
movlw 0x2F  
movwf d2  
movlw 0x03  
movwf d3
```

delay_1seg_0

```
decfsz d1, f  
goto $+2  
decfsz d2, f  
goto $+2  
decfsz d3, f  
goto delay_1seg_0  
goto $+1  
goto $+1  
goto $+1  
return
```

delay250ms

```
movlw 0x4E  
movwf f1  
movlw 0xC4  
movwf f2
```

delay250ms_0

```
decfsz f1, f
```

```
goto $+2
decfsz f2, f
goto delay250ms_0
goto $+1
nop
return
```

```
delay1ms
    MOVLW        .100
    MOVWF        temp1ms
sigue
    NOP
    DECFSZ        temp1ms,F
    GOTO          sigue
    RETURN
```

```
simple_delay
    movlw .1
    movwf o
dtres
    movlw .8
    movwf m
ddos
    movlw .250
    movwf n
duno
    nop
    nop
    decfsz n,1
    goto duno
    decfsz m,1
    goto ddos
    decfsz o,1
    goto dtres
    return
```

END ; FINAL

*******PIC3*******

LIST p=16F887 ; Tipo de microcontrolador
INCLUDE P16F887.INC ; Define los SFRs y bits del
; P16F887

__CONFIG __CONFIG1, __CP_OFF&__WDT_OFF&__INTOSCIO
; Ingresa parámetros de
; Configuración

errorlevel -302 ; Deshabilita mensajes de
; Advertencia por cambio
; Bancos

CBLOCK 0X20

OPCION

tempTMR0

temp1ms

temp

FLECHA1

FLECHA2

FLECHA3

LETRA1

LETRA1H

LETRA1L

LETRA2

LETRA2H

LETRA2L

LETRA3

LETRA3H

LETRA3L

LETRA4

LETRA4H

LETRA4L

LETRA5

LETRA5H

LETRA5L

d1 ;para el delay de 1seg

d2

d3

f1 ;para el delay de 250ms

f2

musica_enable

select_music

counter2

ENDC

; INICIO DEL PROGRAMA

```
ORG    0x00                ; Comienzo del programa (Vector de Reset)
GOTO   MAIN
ORG    0x04
GOTO   INTER
```

encripta_x_16

```
movfw PORTD
movwf counter2
bcf STATUS, C
rlf counter2,1
bcf STATUS, C
rlf counter2,1
bcf STATUS, C
rlf counter2,1
movfw counter2
movwf PORTC
return
```

encripta_x_4

```
movfw PORTD
movwf counter2
bcf STATUS, C
rlf counter2,1
bcf STATUS, C
rlf counter2,1
movfw counter2
```

movwf PORTC

return

encripta_x_2

```
movfw PORTD
movwf counter2
bcf STATUS, C
rlf counter2,1
movfw counter2
```

movwf PORTC

return

INTER

```
BTFSK    INTCON, RBIF ;si la bandera del TMR0 se activo, va a la funcion
```

INTTMR0

```
GOTO     INTRB
BTFSK    INTCON, T0IF
GOTO     INTTMR0
BTFSK    INTCON, INTF
GOTO     RBCINTERR
RETFIE
```

RBCINTERR

```
    movlw b'11111111'  
    movwf musica_enable  
    BCF          INTCON, INTF  
    movfw  PORTD  
    ;btfsc  select_music,1  
    ;goto  encripta_x_2  
    ;btfsc  select_music,0  
    ;goto  encripta_x_4  
    ;goto  encripta_x_16
```

fin_int

```
    ;movwf PORTC  
    RETFIE
```

INTRB

```
    BCF          INTCON, RBIF  
    MOVF         PORTB,F  
    BCF          INTCON,0  
    CLRW  
    BTFSS        PORTB,0  
    MOVLW        0X01  
  
    MOVWF        OPCION  
    RETFIE
```

INTTMRO

```
    BCF          INTCON,T0IF  
    BTFSC        PORTE,0  
    GOTO         hacer0  
    BSF          PORTE,0  
    MOVF         tempTMR0,w  
    MOVWF        TMR0  
    RETFIE
```

hacer0

```
    BCF          PORTE,0  
    MOVF         tempTMR0,w  
    MOVWF        TMR0  
    RETFIE
```

; SETEO DE PUERTOS

MAIN

```
    BANKSEL      TRISB          ; selecciona el banco conteniendo TRISB  
    MOVLW        0XB1  
    MOVWF        OPTION_REG ; TMR0 CON PRESCALADOR 1:4 Y PULLUP
```

ACTIVADO

```
    COMF         IOCB,F  
    MOVLW        0XB8
```

```

MOVWF          INTCON          ;  HABILITA  INTERRUPCIONES  X
CAMBIO DE ESTADO PORTB Y TMR0
CLRF           TRISC
CLRF           TRISD
CLRF           TRISE
BANKSEL        ANSEL
CLRF           ANSEL ; configura puertos con entradas digitales
CLRF           ANSELH      ; configura puertos con entradas digitales
BANKSEL        PORTC; selecciona el puerto B como salida
MOVLW          0XFF
MOVWF          PORTB
CLRF           PORTE
CLRF           OPCION

```

```

clrf musica_enable
clrf select_music
clrf PORTC

```

; DESARROLLO DEL PROGRAMA

LOOP

```

CLRF           PORTD
incf select_music
movlw b'11111111'
subwf musica_enable, W
BTFS          STATUS,Z
GOTO          LOOP
btfsc select_music,1
goto MUSIC3
btfsc select_music,0
goto MUSIC2
goto MUSIC1

```

;***SUBROUTINAS*******

delay1ms

```

MOVLW          .100
MOVWF          temp1ms

```

sigue

```

NOP
DECFSZ         temp1ms,F
GOTO          sigue
RETURN

```

delay1seg

;999990 cycles

```

movlw 0x07
movwf d1
movlw 0x2F
movwf d2

```



```

        movlw 0x03
        movwf d3
delay_1seg_0
        decfsz d1, f
        goto  $+2
        decfsz d2, f
        goto  $+2
        decfsz d3, f
        goto  delay_1seg_0
                                ;6 cycles

        goto  $+1
        goto  $+1
        goto  $+1

                                ;4 cycles (including call)
        return

```

```

delay250ms
                                ;249993 cycles

```

```

        movlw 0x4E
        movwf f1
        movlw 0xC4
        movwf f2
delay250ms_0
        decfsz f1, f
        goto  $+2
        decfsz f2, f
        goto  delay250ms_0

                                ;3 cycles

        goto  $+1
        nop

                                ;4 cycles (including call)
        return

```

```

SIN_NOTA
        BCF      PORTE,0
        BANKSEL  OPTION_REG
        BSF      OPTION_REG,T0CS
        MOVLW    0X96      ;aprox 150 repeticiones (150ms)
        MOVWF    temp
        CALL     delay1ms
        DECFSZ   temp,F
        GOTO     $-2
        BCF      OPTION_REG,T0CS
        BANKSEL  PORTA

```

DO	MOVLW	0X11
	MOVWF	TMR0
	MOVWF	tempTMR0
	RETURN	
RE	MOVLW	0X2C
	MOVWF	TMR0
	MOVWF	tempTMR0
	RETURN	
MI	MOVLW	0X42
	MOVWF	TMR0
	MOVWF	tempTMR0
	RETURN	
FA	MOVLW	0X4D
	MOVWF	TMR0
	MOVWF	tempTMR0
	RETURN	
SOL	MOVLW	0X60
	MOVWF	TMR0
	MOVWF	tempTMR0
	RETURN	
LA	MOVLW	0X72
	MOVWF	TMR0
	MOVWF	tempTMR0
	RETURN	
SI	MOVLW	0X82
	MOVWF	TMR0
	MOVWF	tempTMR0
	RETURN	
do	MOVLW	0X88
	MOVWF	TMR0
	MOVWF	tempTMR0
	RETURN	
re	MOVLW	0X96
	MOVWF	TMR0
	MOVWF	tempTMR0
	RETURN	
mi		

	MOVLW	0XA1
	MOVWF	TMR0
	MOVWF	tempTMR0
	RETURN	
fa		
	MOVLW	0XA6
	MOVWF	TMR0
	MOVWF	tempTMR0
	RETURN	
sol		
	MOVLW	0XB0
	MOVWF	TMR0
	MOVWF	tempTMR0
	RETURN	
la		
	MOVLW	0XB9
	MOVWF	TMR0
	MOVWF	tempTMR0
	RETURN	
si		
	MOVLW	0XC0
	MOVWF	TMR0
	MOVWF	tempTMR0
	RETURN	
MUSIC1		
	BANKSEL	OPTION_REG
	BCF	OPTION_REG,T0CS
	BANKSEL	PORTA
	CALL	encripta_x_16
	CALL	DO
	CALL	delay1seg
	CALL	RE
	CALL	delay1seg
	CALL	MI
	CALL	delay1seg
	BANKSEL	OPTION_REG
	BSF	OPTION_REG,T0CS
	BANKSEL	PORTA
	clrf	musica_enable
	GOTO	LOOP

MUSIC2

```
BANKSEL    OPTION_REG
BCF        OPTION_REG,T0CS
BANKSEL    PORTA
```

```
CALL encripta_x_4
CALL      DO
CALL      delay250ms
CALL      delay250ms
CALL      SOL
CALL      delay1seg
CALL      DO
CALL      delay1seg
CALL      delay1seg
```

```
BANKSEL    OPTION_REG
BSF        OPTION_REG,T0CS
BANKSEL    PORTA
CALL      delay1seg
clrf musica_enable
GOTO      LOOP
```

MUSIC3

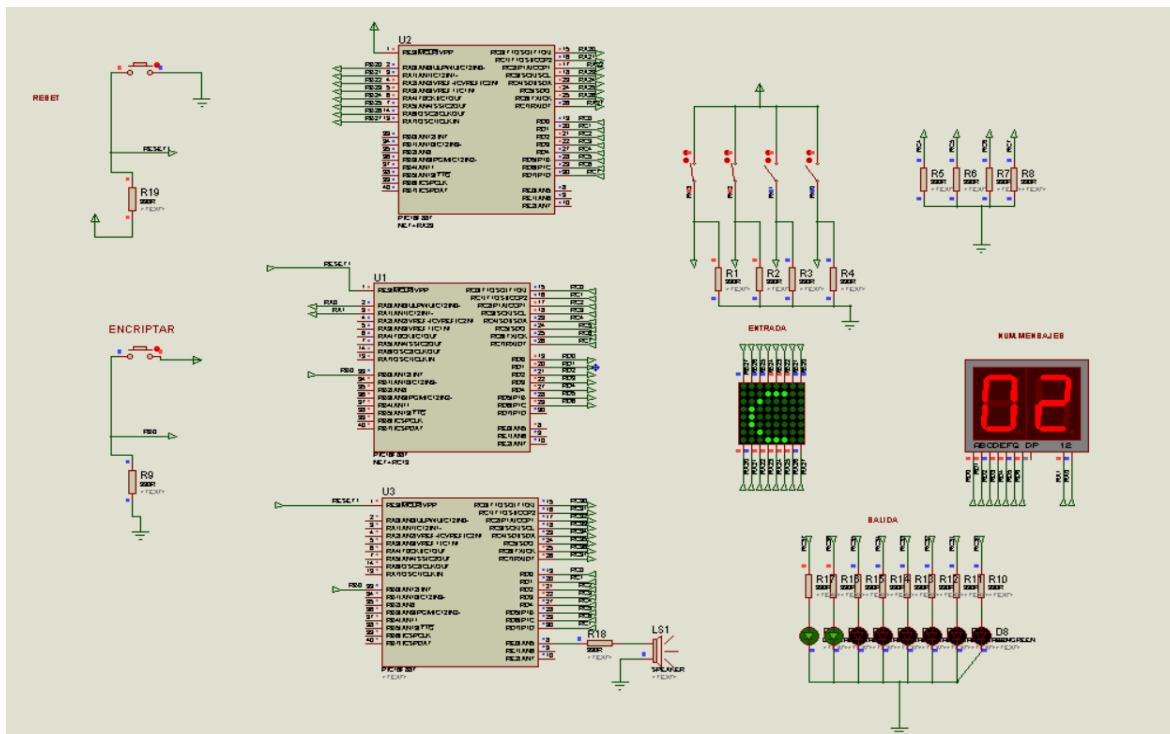
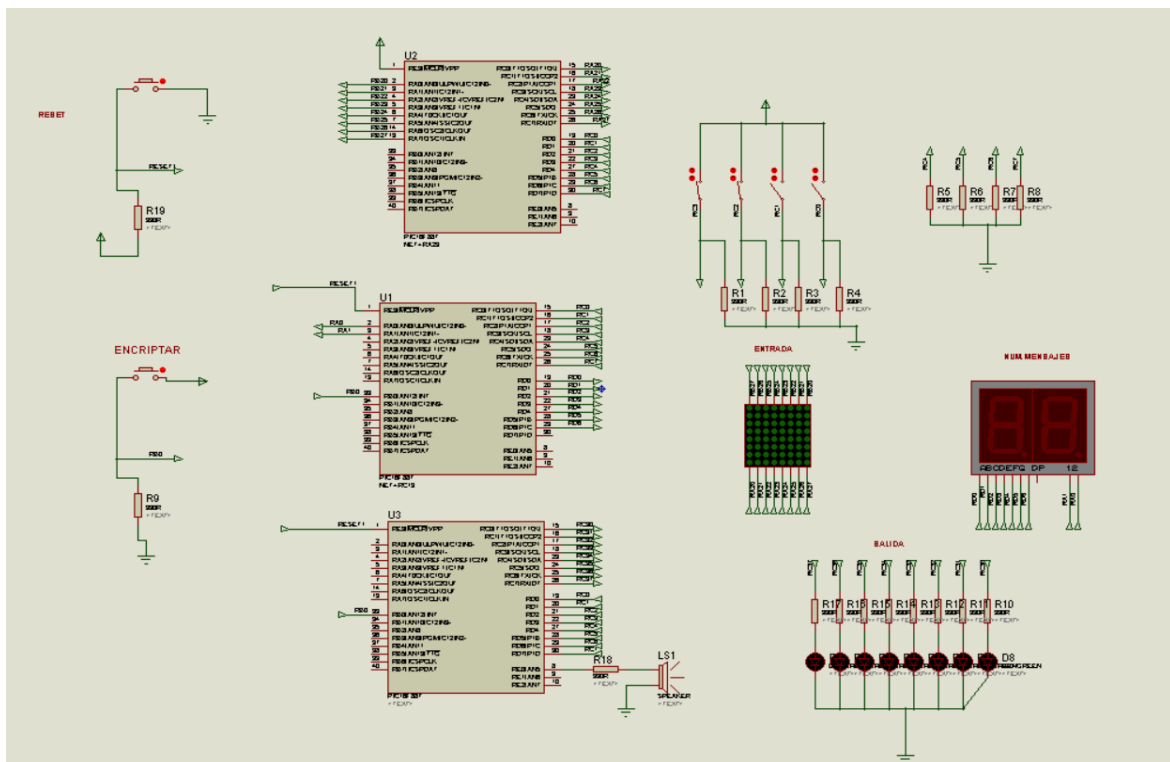
```
BANKSEL    OPTION_REG
BCF        OPTION_REG,T0CS
BANKSEL    PORTA
```

```
CALL encripta_x_2
CALL      SI
CALL      delay250ms
CALL      delay250ms
CALL      SI
CALL      delay1seg
CALL      LA
CALL      delay1seg
```

```
BANKSEL    OPTION_REG
BSF        OPTION_REG,T0CS
BANKSEL    PORTA
CALL      delay1seg
clrf musica_enable
GOTO      LOOP
```

```
END ; fin del programa
```

6.- Copia del circuito en Proteus



7.- Conclusiones

El circuito encripta la señal de cuatro entradas de bits que recibe, y las multiplica por 4. La multiplicación se ha logrado usando una propiedad de los binarios, en la cual al desplazarse a la izquierda dos bits la multiplicación por 4 es efectuada.

El display doble de 7 segmentos es usado para mostrar la cantidad de mensajes que se han encriptado. Por su naturaleza, el display doble no puede encender ambos display al mismo tiempo por lo que se los muestra de manera alternada a una frecuencia en la cual el ojo humano no detecta el cambio, generando así un efecto de encendido al mismo tiempo.

Con la matriz de puntos se muestra el valor hexadecimal de mensaje que se va a encriptar. Una matriz de puntos debe ser controlada por dos puertos en los cuales se controla la multiplexación y los led que se deberán encender.

El speaker muestra un sonido de acuerdo a la clase de encriptación, la cual se selecciona de manera aleatoria. El sonido que sale del speaker se ha construido manipulando la frecuencia de la señal de salida.

8.- Recomendaciones

- Verificar la frecuencia a la que trabaja el simulador ya que por default el valor en el simulador es 4 MHz. Esto puede causar que el retardo en la simulación tenga un comportamiento no deseado ya que los tiempos no coincidirían con la frecuencia establecida en el programa.
- Revisar que el bit GIE del puerto INTCON esté en uno para que las interrupciones puedan ejecutarse.
- Verificar que el registro TRISA esté en cero cuando el puerto A se usará como salida.
- Restablecer los valores de las banderas de las interrupciones ya que de no hacerlo, esto podría causar comportamientos no esperados en el programa.
- Comprobar detenidamente que las etiquetas estén correctamente relacionadas las entradas con las salidas, ya que podríamos estar mostrando un valor erróneo en la salida esperada