

```

;GENERADOR DE UNA ONDA SINUSOIDAL
;Los datos de entrada son de 8 bits y se obtienen de una tabla en RAM de 128
;muestras para un periodo de la onda sinusoidal.
;Se utiliza un modulador Sigma-Delta de primer orden de 7 bits de entrada y
;4 bits de salida.
;La resolución efectiva es de 7 bits
;Rango=> 2 Vpp, 1 Vdc, Vmin=-60 mV, Vmax=2.06 V
;Frecuencia señal generada=> 100 Hz
;Modulación Sigma-Delta:
;Frecuencia del modulador => 250 KHZ
;DAC=> Se utilizan los cuatro bits del generador de referencia
;Terminal analógico de salida=> PIN 21/RA2
;Base de tiempos=> Frecuencia CPU= 64 MHz
;Timer 0 genera la interrupción del modulador
;Timer 1 genera la frecuencia de muestreo del seno
;Probado con PICKit 3
;Camilo Quintáns Graña, Universidad de Vigo. Agosto 2016

```

```

LIST      P=PIC18F45K20          ; Define el Microcontrolador
#include <P18F45K20.INC>         ; Define simbolos del PIC18F45K20

```

```

CONFIG FOSC = INTIO67    ;selección de oscilador interno
CONFIG PBadEN = OFF      ;Programa los bits PB<4:0> como E/S digital
CONFIG LVP = OFF         ;Deshabilitado ICSP
CONFIG WDTEN = OFF       ;Deshabilita el watchdog

```

```
#DEFINE long_tabla 0x80
```

```
#DEFINE tabla 0
```

```
CBLOCK 0x00
```

```
dx
```

```
dv
```

```
dy
```

```
dh
```

```
dato
```

```
valor_FVR
```

```
;registros para salvar temporalmente los valores de los registros
```

```
;especiales de la CPU
```

```
W_TEMP1
```

```
STATUS_TEMP1
```

```
BSR_TEMP1
```

```
W_TEMP2
```

```
STATUS_TEMP2
```

```
BSR_TEMP2
```

```
ENDC
```

```
ORG 0x000
```

```
GOTO main
```

```
ORG 0x08
```

```
GOTO rutina_int_prioridad_alta
```

```
ORG 0x20
```

```
#INCLUDE tabla_ram_seno_128_puntos.inc
```

```
ini_puertos
```

```
;inicialización del puerto conectado a los LEDs
```

```
CLRF TRISD,ACCESS ; Puerto D como salida
```

```
SETF LATD
```

```
;se configura el pin 21 (RA2) para manejar señales analógicas
```

```
BSF TRISA,2,ACCESS ; Deshabilita el buffer de salida digital
```

```
BSF ANSEL,ANS2,ACCESS ; Deshabilita la entrada digital
```

```
RETURN
```

```
;***Configura el módulo FVR para generar una señal de salida en el pin 21*****
```

```
ini_FVR
```

```
CLRF CVRCON,ACCESS ;pongo a cero todos los bits del registro de control
```

```
BCF CVRCON,CVRSS,ACCESS ;selecciona VDD y GND como niveles de referencia
```

```
BSF CVRCON,CVRR,ACCESS ;selecciona el rango alto de salida
```

```
BSF CVRCON2,FVREN,ACCESS;activa el módulo
```

```
BSF CVRCON,CVREN,ACCESS ;activa el módulo
```

```
BSF CVRCON,CVROE,ACCESS ;activa el pin de salida del módulo
```

```
MOVFF CVRCON,valor_FVR ;salvo en el dato los bits de control, así escribo el
;byte entero cuando envío un nuevo dato (4 bits bajos)
```

```
RETURN
```

```
;*****
```

```

;*****subrutina de inicialización del timer 0*****
ini_timer0
    BCF INTCON,TMR0IE,ACCESS    ;deshabilita las interrupciones
    BCF INTCON,TMR0IF,ACCESS    ;borra el flag de interrupción
    BCF T0CON,TMR0ON,ACCESS    ;se apaga el timer 0
    BSF T0CON,T08BIT,ACCESS    ;se configura para 8 bits
    BCF T0CON,T0CS,ACCESS      ;se selecciona el reloj interno Fosc/4
    BSF T0CON,PSA,ACCESS       ;sin preescalado
    ;(64 MHz/4) * (1/64)=250 kHz
    MOVLW 0xFF-D'64'
    MOVWF TMR0L,ACCESS
    BSF T0CON,TMR0ON,ACCESS    ;Se activa el temporizador
    BSF INTCON,TMR0IE,ACCESS    ;se permiten las interrupciones del timer 0
RETURN
;*****

ini_timer1
    BCF PIE1,TMR1IE,ACCESS    ;deshabilita las interrupciones
    BCF PIR1,TMR1IF,ACCESS    ;borra el flag de interrupción
    BCF T1CON,TMR1ON,ACCESS    ;se apaga el timer 0

    ;se carga un 0xFFFF-0x04E2(1250)=0xFB1D
    MOVLW 0xFB
    MOVWF TMR1H,ACCESS
    MOVLW 0x1D
    MOVWF TMR1L,ACCESS

    BSF T1CON,TMR1ON,ACCESS
    BSF PIE1,TMR1IE,ACCESS

RETURN

;*****subrutina de interrupción*****
rutina_int_prioridad_alta
    ;SAVING STATUS, WREG AND BSR REGISTERS IN RAM
    MOVWF W_TEMP1 ; W_TEMP is in virtual bank
    MOVFF STATUS, STATUS_TEMP1 ; STATUS_TEMP located anywhere
    MOVFF BSR, BSR_TEMP1 ; BSR_TMEP located anywhere

    BCF INTCON,GIE,ACCESS

    ;***** TIMER 0 *****
    ;***** Base de tiempos para el modulador *****

    BTFSS INTCON,TMR0IF, ACCESS
    BRA no_es_TMR0

    ;MOVLW 0xFF-D'48' ;resultan unos 250 kHz
    MOVLW 0xFF-D'112' ;resultan unos 125 kHz

    MOVWF TMR0L,ACCESS
    BCF INTCON,TMR0IF,ACCESS

    ;***** Procesado del modulador Sigma-Delta *****
    ;***** Se Suma de la muestra actual dx con el residuo anterior dh *****
    MOVF dh,0,ACCESS
    ADDWF dx,0,ACCESS
    MOVWF dv,ACCESS
    ;***** Se actualiza el residuo con la parte baja de la salida que no ****
    ;***** se envía al DAC *****
    MOVLW 0x0F
    ANDWF dv,0,ACCESS
    MOVWF dh,ACCESS

    ;***** Enviar dato de salida de 4-bits al DAC *****
    ;***** Se envían los 4 MSB de la Suma a los 4 LSB del DAC (FVR) *****
    MOVLW 0xF0 ;máscara para poner a cero la parte baja
    ANDWF valor_FVR,1,ACCESS ;se ponen a cero los cuatro bits bajos

    BCF STATUS,C,ACCESS
    RRCF dv,1,ACCESS
    RRCF dv,1,ACCESS
    RRCF dv,1,ACCESS
    ;RRCF dv,1,ACCESS

```

```

RRCF    dv,0,ACCESS ;con la última rotación ya se deja en el acumulador

;MOVWF    dv,0,ACCESS ;se carga el dato en el registro de trabajo
IORWF   valor_FVR,1,ACCESS ;se actualizan los cuatro bits de VREF
MOVFF   valor_FVR,CVRCON ;se actualiza el registro de control
;*****

no_es_TMR0
;***** TIMER 1 *****
;***** código que carga una nueva muestra de la tabla *****
;***** Frecuencia de muestreo FS= 12800 Hz
BTFSS   PIR1,TMR1IF, ACCESS
BRA     no_es_TMR1

;se carga un 0xFFFF-0x04E2(1250)=0xFB1D
MOVLW   0xFB
MOVWF   TMR1H,ACCESS
MOVLW   0x1D
MOVWF   TMR1L,ACCESS

BCF     PIR1,TMR1IF,ACCESS

BTG     LATD,0,ACCESS

;se carga un dato nuevo de la tabla
MOVLW   long_tabla
CPFSLT  FSR0L,ACCESS
CLRF    FSR0L,ACCESS ;si se pasó de long_tabla-1 se pone a cero
MOVF    POSTINC0,0 ;se carga el dato de la tabla en el registro
           ;de trabajo y se incrementa el puntero
;MOVWF    dato,ACCESS ;se guarda el dato
MOVWF   dx,ACCESS ;se guarda el dato

no_es_TMR1

BSF     INTCON,GIE,ACCESS

MOVFF   BSR_TEMP1, BSR ; Restore BSR
MOVFF   STATUS_TEMP1, STATUS ; Restore STATUS
MOVF    W_TEMP1, W ; Restore WREG
RETFFIE
;*****

main
BSF     OSCCON,IRCF2,ACCESS ;se selecciona el reloj de 16 MHz
BSF     OSCTUNE,PLLEN,ACCESS ;se activa el PLL=> reloj a 64 MHz

CALL    ini_puertos
CALL    ini_tabla_128
CALL    ini_FVR
CALL    ini_timer1
CALL    ini_timer0
BSF     INTCON,GIE,ACCESS ;habilitación de las interrupciones
BSF     INTCON,PEIE,ACCESS
BRA     $
END

```