```
;                       Programa que controla un sintonizador de TV
;    maneja el MC14499,  dos pulsadores y un control remoto Philips en modo Experimental
;                  El último canal utilizado queda grabado en la EEprom
;                              Realizado por: Camilo Monetta
;                                   cmprod@adinet.com.uy
;                                       Salto,Uruguay
;                                     22/10/01  17:16

;XT=4MHZ
;WDT=OFF
;PWRTE=ON


                list        p=16c84
                #include<p16c84.inc>
                __config _XT_OSC &_WDT_OFF &_PWRTE_ON

estado          equ         0x03
pcl             equ         0x02
pb              equ         0x06        ;puerto B
pa              equ         0x05        ;puerto A
dat_con1        equ         0x08        ;dato lectura/escritura de eeprom
adr_con2        equ         0x09        ;direcciona acceder a la eeprom
d1              equ         0x10        ;digito a mostrar 1
d2              equ         0x11        ;digito a mostrar 2
d3              equ         0x12        ;digito a mostrar 3
aux             equ         0x13        ;contador auxiliar
aux1            equ         0x14        ;contador auxiliar 1
dato            equ         0x15        ;dato a mandar
rota            equ         0x16        ;numero de veces a rotar
canal           equ         0x17        ;canal
banda           equ         0x18        ;banda a sintonizar
nbit            equ         0x19        ;numero de bit a enviar
datomc          equ         0x20        ;dato para el mc14499
dividendo       equ         0x21
divisor         equ         0x22
resto           equ         0x23
cociente        equ         0x24
ir_dir          equ         0x25        ;direccion de identificación del control remoto
                                        ;Philips (7 Experimental)
ir_dat          equ         0x26        ;comando enviado por el control remoto
n               equ         0x27        ;variable
eeif            equ         4           ;indica estado de la escritura
wrer            equ         3           ;señalizado de error de escritura
wren            equ         2           ;activación de escritura
wr              equ         1           ;control de escritura
rd              equ         0           ;control de lectura

;*****************************************************************
;========================= Variables del MC14499P ==========================

mc_dat          equ         0           ;linea de datos
mc_en           equ         1           ;enable del MC14499
mc_ck           equ         2           ;pin de la señal de clock
;*****************************************************************

;====================== Variables del Sintonizador TUGH9-A04M ======================
s_en            equ         3           ;enable del sintonizador
s_ck            equ         4           ;clock del sintonizador
s_dat           equ         5           ;dato del sintonizador

;*****************************************************************
                org         0
                bsf         estado,5    ;pone bit 5 de status = 1 . Ir al Banco 1
                clrf        pb          ;selecciona el puerto B como salida
                movlw       0xff
                movwf       pa          ;puerto A como entrada
                bcf         estado,5    ;pone bit 5 de status = 0 . Ir al banco 0
                clrf        pb
                bsf         pb,mc_en    ;enable=1 del MC
                clrf        d3          ;d3=0
                clrf        d2          ;d2=0
                clrf        d1          ;d1=0
;*****************************************************************
                call        eelect      ;lee la eeprom y trae el ultimo canal utilizado
                movlw       .255        ;w=255
                xorwf       canal,w     ;w= canal xor w
                btfss       estado,2    ;el resultado =0?
                goto        chay        ;hay grabado un canal
                movlw       .2          ;w=2
                movwf       canal       ;canal=2
chay            call        display     ;va a la rutina para mostrar el canal
                call        sintonia    ;sintoniza el canal

inicio          btfss       pa,2        ;compara si el bit 2 del pa es uno
                call        pmas        ;si es cero va a pmas
                btfss       pa,3        ;compara si el bit 3 del pa es uno
                call        rx3         ;va a la recepción del infrarrojo
                btfss       pa,4        ;compara si el bit 4 del pa es uno
                call        pmenos      ;si es cero va a pmenos
                goto        inicio      ;no se presiono ningún pulsador

;*****************************************************************

;====================Secuencia a seguir para sintonizar un canal ==================
sintonia
                bsf         pb,s_en     ;pone en alto enabled del sintonizador
                nop
                call        enbanda     ;va a la subrutina de envio de banda
                call        endivh      ;va a la subrutina de envio de division alta
                call        endivl      ;va a subrutina de envio de division baja
                bcf         pb,s_en     ;pone en bajo enabled del sintonizador
                return                  ;retorna a rutina principal
;*****************************************************************

;========================= Secuencia para enviar bit ==========================
enviabit
                btfss       dato,7      ;pregunta si el bit 7 de dato es 1
                bcf         pb,s_dat    ;dato=0
                btfsc       dato,7      ;pregunto si dato es cero
                bsf         pb,s_dat    ;si es cero
                bsf         pb,s_ck     ;clock =1
                nop
                bcf         pb,s_ck     ;clock=0
                return
;*****************************************************************
;============================ Envia la banda ============================
enbanda
                movlw       HIGH divl
                movwf       PCLATH
                movf        canal,w     ;w=canal
                call        divl        ;Busca el valor de la banda
                movwf       dato
                clrf        PCLATH
                movlw       b'00001111' ;w=00001111
                andwf       dato,wc     ;w= dato and 00001111
                                        ;elimino los 4 bit de mas peso
                movwf       dato        ;dato=w
                rlf         dato,1
                rlf         dato,1
                rlf         dato,1
                rlf         dato,1
                movlw       .4          ;w=4
                movwf       rotar       ;rotar=w=4
cban            call        enviabit
                rlf         dato,1
                decfsz      rotar,1
                goto        cban
                return
;*****************************************************************
;====================== Envia la parte alta de la division ==========================
endivh
                movlw       HIGH  divh
                movwf       PCLATH
                movf        canal,w     ;w=canal
                call        divh
                movwf       dato        ;dato=valor traido de la tabla
                clrf        PCLATH
                rlf         dato,1
                rlf         dato,1      ;rote 2 veces el dato de la tabla divh
                movlw       .6
                movwf       rotar       ;rotar=w=7
                movwf       nbit
cdivh           call        enviabit
                rlf         dato,1
                decfsz      rotar,1
                goto        cdivh
                return

;====================== Envia la parte baja de la division ==========================
endivl
                movlw       HIGH  divl
                movwf       PCLATH
                movf        canal,w     ;w=canal
                call        divl
                movwf       dato        ;valor traido de la tabla
                clrf        PCLATH
                movlw       b'11110000' ;w=11110000 b
                andwf       dato,w      ;w= dato and 1111000
                                        ;elimino los 4 bit de menos peso
                movwf       dato        ;dato=w
                movlw       .8
                movwf       rotar       ;rotar=w=8
cdivl           call        enviabit
                rlf         dato,1
                decfsz      rotar,1
                goto        cdivl
                return


;========================== RUTINAS PARA EL MC ==========================
;*****************************************************************
;========================== SE  PULSO LA  TECLA  -  ======================

pmenos          decf        canal,1     ;canal=canal-1
                movlw        .1         ;w=1
                xorwf       canal,w
                btfss       estado,2    ;canal=1?
                goto        no1         ;canal>1
                movlw        .125
                movwf       canal
no1             call        pp
                return

;========================== SE  PULSO LA TECLA +  ========================

pmas            incf        canal,1     ;canal=canal+1
                movlw       .126        ;w=126
                xorwf       canal,w
```

Control Remoto del Sintonizador

```
                btfss   estado,2        ;canal=126?
                goto    no126
                movlw   .2              ;w=2
                movwf   canal
no126           call    pp
                return


;============================ Rutina comun a p+ y p- ========================
pp              clrf    cociente
                call    display         ;muestra el nuevo canal
                call    sintonia        ;sintoniza el canal
                call    eegrab          ;graba el canal en la EEprom
                call    retardo
                return


;================ Separa de canal cada digito a mostar  D3 D2 D1 ==================
d3d2d1
                movf    canal,w         ;w=canal
                movwf   dividendo       ;dividendo=canal
                movlw   .100            ;w=100
                movwf   divisor         ;divisor=100
                call    divide          ;divide canal/100
                movf    cociente,w      ;w=cociente
                movwf   d3              ;d3=cociente
                movf    resto,w         ;w=resto
                movwf   dividendo       ;dividendo=resto
                movlw   .10             ;w=10
                movwf   divisor         ;divisor=10
                cal     divide
                movf    cociente,w      ;w=cociente
                movwf   d2              ;d2=cociente
                movf    resto,w         ;w=resto
                movwf   d1              ;d1=resto
                return


;=========================  DISPLAY  ======================================
display
                call    d3d2d1          ;separa cada digito
                bcf     pb,mc_en        ;enable=0 MC14499 habilitado para recibir datos
                clrf    datomc          ;dato=0=punto
                call    mc_envia
                movf    d1,w            ;w=d1
                movwf   datomc          ;dato=d1
                call    mc_envia
                movf    d2,w            ;w=d2
                movwf   datomc          ;dato=d2
                call    mc_envia
                movf    d3,w            ;w=d3
                movwf   datomc          ;dato=d3
                call    mc_envia
                clrf    datomc          ;dato=0=4º digito no usado en este caso
                call    mc_envia
                bsf     pb,mc_en        ;enable=1 MC14499 no recibe mas datos
                bsf     pb,mc_ck
                return


;========================= Secuencia para enviar bit ========================
mc_envia
                movlw   .4              ;w=4
                movwf   rotar           ;rotar=4
sigue           call    mc_enbit
                rlf     datomc,1        ;roto el dato 1 lugar a la izquierda
                decfsz  rotar,1         ;rotar=rotar-1
                goto    sigue
                return

;=====================================================================
mc_enbit
                btfss   datomc,3        ;pregunta si el bit 3 de dato es 1
                bcf     pb,mc_dat       ;dato=0
                btfsc   datomc,3        ;pregunto si dato es cero
                bsf     pb,mc_dat       ;si es cero
                bsf     pb,mc_ck        ;clock =1
                nop
                bcf     pb,mc_ck        ;clock=0
                return

;============================== Retardo ==================================
retardo         movlw   .155            ;w=155
                movwf   aux             ;aux=255
p44             movlw   .255            ;w=255
                movwf   aux1            ;aux1=255
redo            nop                     ;1 µSEG
                nop                     ;1 µSEG
                nop                     ;1 µSEG
                nop                     ;1 µSEG
                nop                     ;1 µSEG
                nop                     ;1 µSEG
                nop                     ;1 µSEG
                decfsz  aux1,1          ;decrementa aux1 y salta
                                        ;la siguiente instruccion si es cero
                goto    redo            ;si aux1 es distinto de cero va a redo
                decfsz  aux,1           ;decrementa aux y salta
                                        ;la siguiente instruccion si es cero
                goto    p44             ;si aux es distinto de cero va a p44
                return                  ;retorna a la rutina que lo llamo


;================= Rutina que divide IN(dividendo/divisor )  OUT(cociente,resto) =========
divide:
                clrf    resto           ;borra el resto
                movlw   8               ;Para dividir
                movwf   aux             ;8 bits

divloop:        rlf     dividendo,1     ;Correr el dividendo
                rlf     resto,1
                movf    divisor,w
                subwf   resto,w


CHECKLESS:      BNC     NOSUB           ;Si la porcion corrida del dividendo fu,
                movf    divisor,W       ;De otro modo,
                                        ;REMAINDER = REMAINDER - DIVISOR.
                subwf   resto

NOSUB:          rlf     cociente,1
                decfsz  aux,1
                goto    divloop
                return


;====================== Graba el canal en la EEprom ========================
eegrab
                clrf    adr_con2        ;eeadr=0 direcciona escribir
                movf    canal,w         ;w=canal
                movwf   dat_con1        ;eedata=canal valor a grabar

                bsf     estado,5        ;paso al banco 1
                clrf    dat_con1        ;borra el dato
                bsf     dat_con1,wren   ;Permite escrituras
                movlw   0x55            ;w=55H
                movwf   adr_con2
                movlw   0xAA            ;w=AAH
                movwf   adr_con2
                bsf     dat_con1,wr     ;Inicia un ciclo de escritura
ll              btfsc   dat_con1,wr     ;pregunta si termino la escritura
                goto    ll              ;no termino
                bcf     dat_con1,wren   ;No permite mas escrituras
                bcf     dat_con1,eeif   ;Borra el bit indicador de escritura terminada
                bcf     estado,5        ;Paso al banco 0
                return

;==================== Lectura del canal grabado en la EEprom ====================
eelect
                clrf    adr_con2        ;eeadr=0 direccion a leer
                bsf     estado,5        ;paso al banco 1
                bsf     dat_con1,rd     ;Comienzo de lectura
                bcf     estado,5        ;paso al banco 0
                movf    dat_con1,w      ;w=valor leido
                movwf   canal           ;canal=w
                return



;=====================================================================
;==================== RUTINAS PARA EL CONTROL REMOTO ==================
;=====================================================================
;
;                       FORMATO DE LA SEÑAL RECIBIDA
;
;_____1__ __2__ __3__ __4__ __5__ __6__ __7__ __8__ __9__ __10__ __11__ __12__ __13__ __14__
;     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
;     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
;     |__|  |__|  |__|  |__|  |__|  |__|  |__|  |__|  |__|  |__|  |__|  |__|  |__|  |__|  |__|
;   agc  agc  chk
;        |_________direccion_________|_________ Comando_________________|
;
;
;Direccion= 0 para TV
;Comandos
;0 al 9  numeros
;20H canal +
;21H canal -
;10H volumen +
;11H volumen -

;================================ RX3 ================================
rx3             call    rx              ;va a leer el primer dato
                movf    ir_dat,w
                movwf   d3
                movlw   .255            ;w=255
                xorwf   d3,w            ;w= d3 xor w
                btfss   estado,2        ;el resultado=0?
                goto    p2              ;sigue preguntando
                clrf    d3
                return                  ;Tengo error en la recepcion
p2              movlw   .38             ;w=38  SLEEP (1--)
                xorwf   d3,w            ;w=d3 xor w
                btfss   estado,2        ;el resultado=0?
                goto    chd             ;Se pregunta por ch+ ch- o d2 y d2
                movlw   .1
                movwf   d3              ;d3=1 el canal esta comprendido entre el 100 al 125
                goto    ddd

chd             movlw   .32             ;CHANEL +
```

```
                xorwf   d3,w        ;w= d3 xor d3
                btfss   estado,2    ;Se presiono Chanel +
                goto    chm         ;NO se presiono
                incf    canal,1     ;canal=canal+1
                movlw   .126        ;w=126
                xorwf   canal,w
                btfss   estado,2    ;el resultado=0?
                goto    oich        ;canal<126
                movlw   .2          ;w=2
                movwf   canal       ;canal=2
                goto    oich        ;muestra el canal
chm             movlw   .33         ;w=33  CHANEL -
                xorwf   d3,w        ;w= d3 xor w
                btfss   estado,2    ;Se presiono Chanel -
                goto    d1d2        ;NO se presiono
                decf    canal,1     ;canal=canal-1
                movlw    .1         ;w=1
                xorwf   canal,w
                btfss   estado,2    ;el resultado=0?
                goto    oich        ;canal>1
                movlw   .125        ;canal=1
                movwf   canal       ;canal=125
                goto    oich

d1d2            movf    d3,w
                movwf   d2          ;d2=d3
                clrf    d3
                goto    ud1
ddd
                call    retardo
q               btfss   pa,3
                goto    t
                goto    q

t               call rx
                movf    ir_dat,w
                movwf   d2          ;d2 = ok
ud1
                call    retardo
r               btfss   pa,3
                goto    ds
                goto    r

ds              call    rx
                movf    ir_dat,w
                movwf   d1          ;d1 = ok

;============================= Obtengo el Canal  ============================
oi              clrf    canal
                movlw   .0          ;w=0
                xorwf    d3,w       ;w= d3 xor w
                btfss   estado,2    ;El resultado es =0?
                call     mul100     ;d3>0
                movlw   .0
                xorwf   d2,w        ;w= d2 xor w
                btfss   estado,2    ;el resultado=0?
                call    mul10       ;d2>0
                movf    d1,w        ;w=d1
                addwf   canal,1     ;canal=canal+d1

oich            cal     pp          ;rtuina que muestra ,sintoniza
                                    ;y graba el canal
                return

;============================= MUL100 ================================
mul100
                movlw   .100
                movwf   aux
mul1            incf    canal,1     ;canal=canal+1
                decfsz  aux,1       ;aux=aux-1
                goto mul1
                return


;============================= MUL10 ================================
mul10           movf    d2,w        ;w=d2
                movwf   aux1
mul2            movlw   .10
                movwf   aux
mul             incf    canal,1     ;canal=canal+1
                decfsz  aux,1       ;aux=aux-1
                goto    mul
                decfsz  aux1,1      ;aux1=aux1-1
                goto mul2
                return


;============================= RX ================================
;Espera a que pasen los 3 primeros bit que son de AGC CHECK

rx              clrf    ir_dat
                clrf    ir_dir
                call    ret4_7      ;descarto los 2,75 bit de inicio

;Descarto los 5 bit de direccion
                movlw   .5          ;w=5
                movwf   n           ;n=5
nudir           rlf      ir_dir,1
                btfss   pa,3        ;pregunta si el pin RA3 es 1
                goto    dircer      ;RA=0
                bsf     ir_dir,0
dircer          call    ret_17      ;retardo 1,778 milisegundos
                decfsz  n,1         ;n=n-1
                goto    nudir       ;n<>0


;Recepcion de COMANDO  6 BIT
                movlw   .6          ;w=6
                movwf   n           ;n=6
nucom           rlf     ir_dat,1
                btfss   pa,3        ;pregunta si el pin RA3 es 1
                goto    cero        ;RA=0
                bsf     ir_dat,0    ;pone en 1 el bit 0
cero            call    ret_17      ;retardo 1,778 milisegundos
                decfsz   n,1        ;n=n-1
                goto    nucom       ;n<>0


;Compara si la direccion es 7
                movlw   .7
                xorwf   ir_dir,w
                btfss   estado,2
                goto    err         ;No es igual
                return  ;Es igual
err              movlw  .255
                movwf    ir_dat
                return

;======================== Retardo de 1,778 milisegundos ======================
ret_17          movlw   .254        ;w=42
                movwf   aux1        ;aux1=82
redir           nop
                nop
                nop
                nop
                decfsz  aux1,1      ;decrementa aux1 y salta la siguiente
                                    ;instruccion si es cero
                goto    redir       ;si aux1 es distinto de cero va a redo
                return  ;retorna a la rutina que lo llamo


;=================== Retardo de (2,75* 1,778)=4,889 milisegundos ====================
ret4_7          movlw   .4          ;w=7
                movwf   aux         ;aux=7
p444            movlw   .244        ;w=42
                movwf   aux1        ;aux1=82
redo4           nop
                nop
                decfsz  aux1,1      ;decrementa aux1 y salta la
                                    ;siguiente instruccion si es cero
                goto    redo4       ;si aux1 es distinto de cero va a redo
                decfsz  aux,1       ;decrementa aux y salta la siguiente
                                    ;instruccion si es cero
                goto    p444        ;si aux es distinto de cero va a p44
                return              ;retorna a la rutina que lo llamo



                                                    org h'300'
;==================== Tabla con la parte alta de la division ==================
divh
                addwf   pcl,f
                DT H'00',H'07',H'06',H'06',H'07',H'07',H'07',H'08',H'0D',H'0E',H'0E'
                DT H'0E',H'0F',H'0F',H'10',H'0A',H'0A',H'0B',H'0B',H'0B',H'0C'
                DT H'0C',H'0D',H'0D',H'10',H'10',H'11',H'11',H'11',H'12',H'12'
                DT H'13',H'13',H'13',H'14',H'14',H'14',H'15',H'15',H'16',H'16'
                DT H'16',H'17',H'17',H'17',H'18',H'18',H'19',H'19',H'19',H'1A'
                DT H'1A',H'1A',H'1B',H'1B',H'1C',H'1C',H'1C',H'1D',H'1D',H'1D'
                DT H'1E',H'1E',H'1F',H'1F',H'1F',H'20',H'20',H'20',H'21',H'21'
                DT H'22',H'22',H'22',H'23',H'23',H'23',H'24',H'24',H'25',H'25'
                DT H'25',H'26',H'26',H'26',H'27',H'27',H'28',H'28',H'28',H'29'
                Dt H'29',H'29',H'2A',H'2A',H'2B',H'08',H'08',H'09',H'09',H'0A'
                DT H'2B',H'2B',H'2C',H'2C',H'2C',H'2D',H'2D',H'2D',H'2E',H'2E',H'2E'
                DT H'2F',H'2F',H'2F',H'30',H'30',H'31',H'31',H'31',H'32',H'32'
                Dt H'32',H'33',H'33',H'34',H'34',H'34'
;==============================================================================

;==== Tabla con la parte baja de la division y en los 4 bit menos significativos la banda ====

divl
                addwf   pcl,f
                DT H'00',H'71',H'51',H'B1',H'11',H'B1',H'11',H'D2',H'32',H'92'
                DT H'F2',H'52',H'B2',H'12',H'72',H'D2',H'32',H'92',H'F2',H'52'
                DT H'B2',H'12',H'72',H'72',H'D2',H'32',H'92',H'F2',H'52',H'B2'
                DT H'12',H'72',H'D2',H'32',H'92',H'F2',H'52',H'B2',H'12',H'72'
                DT H'D2',H'32',H'92',H'F2',H'52',H'B8',H'18',H'78',H'D8',H'38'
                DT H'98',H'F8',H'58',H'B8',H'18',H'78',H'D8',H'38',H'98',H'F8'
                DT H'58',H'B8',H'18',H'78',H'D8',H'38',H'98',H'F8',H'58',H'B8'
                DT H'18',H'78',H'D8',H'38',H'98',H'F8',H'58',H'B8',H'18',H'78'
                DT H'D8',H'38',H'98',H'F8',H'58',H'B8',H'18',H'78',H'D8',H'38'
                DT H'98',H'F8',H'58',H'B8',H'18',H'91',H'F1',H'51',H'B1',H'11'
                DT H'78',H'D8',H'38',H'98',H'F8',H'58',H'B8',H'18',H'78',H'D8'
                DT H'38',H'98',H'F8',H'58',H'B8',H'18',H'78',H'D8',H'38',H'98'
                DT H'F8',H'58',H'B8',H'18',H'78',H'D8'
;==============================================================================

                end                 ;fin del  programa
```