

Apéndice C: Listado del Programa desarrollado para el PIC16F874 (Etapa de Transmisión)

Este es el programa desarrollado para el PIC16F874 en la Etapa de Transmisión:

```

;*****
; This file is a basic code template for assembly code generation      *
; on the PICmicro PIC16F874. This file contains the basic code        *
; building blocks to build upon.                                       *
;                                                                       *
; If interrupts are not used all code presented between the ORG       *
; 0x004 directive and the label main can be removed. In addition     *
; the variable assignments for 'w_temp' and 'status_temp' can        *
; be removed.                                                          *
;                                                                       *
; Refer to the MPASM User's Guide for additional information on       *
; features of the assembler (Document DS33014).                       *
;                                                                       *
; Refer to the respective PICmicro data sheet for additional         *
; information on the instruction set.                                  *
;                                                                       *
; Template file assembled with MPLAB V4.00 and MPASM V2.20.00.      *
;                                                                       *
;*****
;                               *
; Filename:   transv32.asm      *
; Date:      15 de Mayo 2005   *
; File Version: 000           *
;                               *
; Author:    Otto Joel Paz Luna *
; Company:   UDLA              *
;                               *
;                               *
;                               *

```

```

;*****
;
;
; Files required:
;
;
;
;
;*****
;
;
; Notes:
;
;
;
;
;
;*****

list    p=16f874    ; list directive to define processor

#include <p16f874.inc>    ; processor specific variable definitions

__CONFIG _CP_OFF & _WDT_OFF & _BODEN_OFF & _PWRTE_OFF & _XT_OSC &
_WRT_ENABLE_OFF & _LVP_OFF & _DEBUG_OFF & _CPD_OFF

; '__CONFIG' directive is used to embed configuration data within .asm file.
; The labels following the directive are located in the respective .inc file.
; See respective data sheet for additional information on configuration word.

;**** VARIABLE DEFINITIONS

contador1    equ    0x20    ; variable used for context saving
contador2    equ    0x21    ; variable used for context saving
tiempo       equ    0x22
loop         equ    0x23
iniciobff    equ    0x24

;**** MACROS PARA CAMBIAR DE BANCO

banco0 MACRO

    bcf    status,rp0

    bcf    status,rp1

ENDM

```

banco1 MACRO

```

    bsf    status,rp0
    bcf    status,rp1
    ENDM

```

banco2 MACRO

```

    bcf    status,rp0
    bsf    status,rp1
    ENDM

```

banco3 MACRO

```

    bsf    status,rp0
    bsf    status,rp1
    ENDM

```

;*********

```

    ORG    0x0000    ; processor reset vector
    clrf   PCLATH    ; ensure page bits are cleared
    goto   principal ; go to beginning of program

```

principal

```

    banco0    ; se asegura iniciar en banco 0
    nop
    banco1    ; ir a banco 1
    movlw    .192
    movwf    trisc    ;se ajustan pines a 11000000
    bsf     trisd,0    ; se ajusta pin 0 puerto como entrada.
    movlw    .51
    movwf    SPBRG    ; se ajusta baud rate con BRGH=1 y 9600Kbps a 8 MHz
    movlw    .36
    movwf    TXSTA    ; se ajusta reg txsta a 00100100
    movlw    .64
    movwf    SSPSTAT    ; se configura reg sspstat a 01000000

    banco0    ; ir a banco 0
    movlw    .144

```

```

movwf RCSTA      ; se ajusta reg rcsta a 10001000

movlw .33
movwf SSPCON     ; se ajusta reg sspcon a 00100001 para Clock Fosc/64

apuntador
movlw 0x24
movwf fsr        ; se determina comienzo de búfer
movlw .91
movwf contador2 ; se determina longitud de búfer
clrf  contador1

repcion
inst si activa  btfss  pir1,rcif  ; se verifica bandera de recepcion de ocho bits, se brinca
                goto   $-1
                goto   almacenar ; ir a subrutina
                goto   transmisión ; ir a subrutina

almacenar
btfsc RCSTA,OERR ; se verifica si hay overrun error
goto  ErrOverrun ; ir a subrutina si hay overrun error
btfsc RCSTA,FERR ; se verifica si hay error de frame
goto  ErrFrame   ; ir a subrutina si hay error de frame
movf  rreg,w     ; el contenido del rreg se mueve al acumulador
movwf INDF      ; se guarda el dato en la memoria indicada por fsr
incf  fsr,1     ; se aumenta el apuntador en 1

incf  contador1,1 ; se incrementa el contador1 en 1

cero
decsz contador2,1 ; se decrementa en 1 el contador2 y verifica cuando es

inst si activa  goto  espera      ; ir a subrutina
                btfss  pir1,rcif  ; se verifica bandera de recepcion de ocho bits, se brinca

```

```

goto    $-1
bcf     RCSTA,4      ; se desactiva recepción
movf    rcreg,w      ; el contenido del rcreg se mueve al acumulador
movwf   INDF         ; se guarda el dato en la memoria indicada por fsr
incf    contador1,1 ; se aumenta el apuntador en 1
goto    transmision ; sale de esta rutina

```

espera

```

movlw   .17
movwf   loop

```

subrut

```

call    verifica    ; se llama subrutina
decfsz  loop,1
goto    subrut
bcf     RCSTA,4      ; se desactiva recepción
goto    transmision

```

verifica

```

movlw   .255
movwf   tiempo

```

checa

```

btfsc   pir1,rcif   ; se verifica bandera de recepción de ocho bits
goto    almacenar
decfsz  tiempo,1
goto    checa
return  ; fin de subrutina

```

transmision

```

movlw   0x24
movwf   fsr          ; ir al primer banco de memoria del búfer

```

sincronia

```

movlw .219          ; se escribe 11011011 en el registro w
instruccion si activo
btfss portd,0      ; se verifica que el pin 0 de este puerto, se brinca
goto $-1

nop
nop
nop
nop
nop
nop
nop
nop

seleccionada
movwf sspbuf      ; se transmite la informacion sobre la linea de video
banco1
btfss sspstat,bf
goto $-1          ; se espera a leer el dato que llega
banco0
movf sspbuf,w     ; se mueve el dato que llego y se desecha

movf INDF,w       ; se saca del buffer un dato y se coloca en el registro w
nop

movwf sspbuf      ; se envia el dato por el SPI

brica instruccion
btfss PIR1,TXIF   ; se verifica si el transmisor esta ocupado de lo contrario
goto $-1

transmite
movwf txreg       ; se mueve el contenido del acumulador al txreg y se

incf fsr,1        ; se incrementa el apuntador en 1

```

```

    banco1
    btfss  sspstat,bf
    goto  $-1          ; se espera a leer el dato que llega

    banco0
    movf  sspbuf,w     ; se mueve el dato que llego y se desecha
    btfsc portd,0     ; se verifica que el pin 0 de este puerto, se brinca
instrucion si activo
    goto  $-1
    decfsz contador1,1 ; se decrementa el contador 1 y se verifica si llega a cero
    goto  sincronia
    bsf   RCSTA,CREN

    goto  apuntador

ErrOverrun:  bcf   RCSTA,CREN ; se pone en 0 el bit CREN
             bsf   RCSTA,CREN ; se pone en 1 el bit CREN para rehabilitar la recepcion
             return

ErrFrame:   movf  RCREG,W          ; se lee el registro RCREG y se va a stinfo
             goto  almacenar

             nop
             nop

    END          ; directive 'end of program'

```