APÉNDICE A

```
;*********************************************************************
;                                                                    *
;                       TESIS                                        *
;           ADQUISICION DE LA POSICION DE UN VEHÍCULO                *
;                                                                    *
;*********************************************************************
;                                                                    *
;     Programa que recibe datos de un GPS y los transmite            *
;                                                                    *
;     1. Recibe datos del GPS                                        *
;     2. Filtra la informacion                                       *
;     3. Espera un tiempo (X) para empezar a transmitir              *
;        los datos recibidos a la Creatalink                         *
;                                                                     *
;*********************************************************************
;                                                                    *
;    Filename:        filtro.asm                                      *
;    Date:          26/Nov/03                                        *
;    File Version:                                                   *
;                                                                    *
;    Author: Cesar Martinez Mozo                                     *
;    Company: UDLA                                                   *
;                                                                    *
;                                                                    *
;*********************************************************************



      list      p=16f874            ; list directive to define processor
      #include <p16f874.inc>        ; processor specific variable
definitions

__CONFIG _CP_OFF & _WDT_ON & _BODEN_ON & _PWRTE_ON & _RC_OSC &
_WRT_ENABLE_ON & _LVP_ON & _CPD_OFF

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

;=====================================================================
;                 DEFINICION DE VARIABLES
;=====================================================================

w_temp        EQU     0x70        ; variable used for context saving
status_temp   EQU     0x71        ; variable used for context saving
BYTE_COUNTER  EQU     0X72      ;CONTADOR DE BYTES
BYTE_ARROBA   EQU     0X40        ;40=CODIGO ASCII DE @
BYTE_E        EQU     0X45      ;45=CODIGO ASCII DE E
BYTE_Q        EQU     0X71      ;71=CODIGO ASCII DE q
DATOS_PTR0    EQU     0X20      ;INICIO DE MEMORIA RAM LIBRE BANCO 0
DATOS_PTR1    EQU     0XA0      ;INICIO DE MEMORIA RAM LIBRE BANC0 1
```

```
LIM_BANK0     EQU     0X32      ;LIMITE DE MEMORIA RAM DEL BANCO 0
                          ;32h=50 LOCALIDADES DE MEMORIA
BRG_VALUE     EQU     0x81
BYTE_PARO     EQU     0X0A      ;2A=CODIGO ASCII DE *

NO_SEG        EQU     0X01        ;NUMERO DE SEGUNDOS EN HEXADECIMAL
SEC_COUNTER   EQU     0X73        ;CONTADOR DE SEGUNDOS
;*****************************************************************
      ORG     0x000             ; processor reset vector
      clrf    PCLATH            ; ensure page bits are cleared
      goto    LOAD              ; go to beginning of program


      ORG     0x004             ; interrupt vector location
      movwf   w_temp            ; save off current W register contents
      movf  STATUS,w          ; move status register into W register
      bcf     STATUS,RP0        ; ensure file register bank set to 0
      movwf status_temp       ; save off contents of STATUS register


; isr code can go here or be located as a call subroutine elsewhere


      bcf     STATUS,RP0        ; ensure file register bank set to 0
      movf    status_temp,w     ; retrieve copy of STATUS register
      movwf STATUS            ; restore pre-isr STATUS register contents
      swapf   w_temp,f
      swapf   w_temp,w          ; restore pre-isr W register contents
      retfie                    ; return from interrupt


;=====================================================================
;           TIEMPO DE ESPERA ANTES DE INICIAR RECEPCION
;=====================================================================
LOAD                          ;ESPERA TIEMPO PARA QUE LA CREATALINK
      CLRF  STATUS                  ;ENCUENTRE LA RED DE SKYTEL
      CLRF  PORTB
      BSF   STATUS,RP0
      CLRF  TRISB
      BCF   STATUS, RP0
      BSF   PORTB, 0X07
      MOVLW 0X3C              ;3CH=60D
      MOVWF SEC_COUNTER       ;CARGA EL CONTADOR CON 60 SEG
      CALL  TIMER
      BCF   PORTB, 0X07

;=====================================================================
;                       INICIALIZACION
;=====================================================================

MAIN
      CLRF  STATUS
      CLRF  BYTE_COUNTER            ;INICIALIZA EL CONTADOR
      MOVLW DATOS_PTR0
      MOVWF FSR
      CLRF  PORTB             ;CONFIGURACION DEL PUERTO B
      BSF   STATUS, RP0       ;PARA LA DETECCION DE ERRORES
```

```
      CLRF  TRISB
      CLRF  TRISC
      BCF   TRISC, 0X06       ;CONFIGURA TX = SALIDA
      BSF   TRISC, 0X07       ;CONFIGURA RX = ENTRADA
      MOVLW       BRG_VALUE         ;CONFIGURA EL BAUD RATE
      MOVWF SPBRG
      MOVLW 0X26              ;RX DE 8 BITS, BRG ALTA
      MOVWF TXSTA             ;HABILITA TX
      BCF   STATUS, RP0
      MOVLW 0X90              ;HABILITA EL PUERTO SERIAL ASINCRONO
      MOVWF RCSTA             ;RECEPCION CONTINUA



;=====================================================================
;                 SELECION DE INFORMACION
;=====================================================================

SEL_INFO
      BCF   STATUS, RP0
      BTFSS PIR1, RCIF
      GOTO  SEL_INFO

      BCF   PIR1, RCIF
      BTFSC RCSTA, FERR
      GOTO  ERR_FERR
      BTFSC RCSTA, OERR
      GOTO  ERR_OERR
      MOVF  RCREG, W          ;COPIA EL DATO RECIBIDO A W
      XORLW BYTE_ARROBA
      BTFSS STATUS, Z
      GOTO  SEL_INFO

CHK_ARROBA
      BCF   STATUS, RP0
      BTFSS PIR1, RCIF
      GOTO  CHK_ARROBA

      bcf   pir1, rcif
      BTFSC RCSTA, FERR
      GOTO  ERR_FERR
      BTFSC RCSTA, OERR
      GOTO  ERR_OERR
      MOVF  RCREG, W          ;COPIA EL DATO RECIBIDO A W
      xorlw byte_arroba
      btfss status, z
      goto  sel_info

chk_e
      bcf   status, rp0
      btfss pir1, rcif
      goto  chk_e

      bcf   pir1, rcif
      BTFSC RCSTA, FERR
      GOTO  ERR_FERR
```

```
      BTFSC RCSTA, OERR
      GOTO  ERR_OERR
      MOVF  RCREG, W          ;COPIA EL DATO RECIBIDO A W
      xorlw byte_e
      btfss status, z
      goto  sel_info

chk_q
      bcf   status, rp0
      btfss pir1, rcif
      goto  chk_q

      bcf   pir1, rcif
      BTFSC RCSTA, FERR
      GOTO  ERR_FERR
      BTFSC RCSTA, OERR
      GOTO  ERR_OERR
      MOVF  RCREG, W          ;COPIA EL DATO RECIBIDO A W
      xorlw byte_q
      btfss status, z
      goto  sel_info


;=====================================================================
;                       RECEPCION
;=====================================================================

RX_DATOS

      BCF   STATUS, RP0
      BTFSS PIR1, RCIF        ;ESPERA TERMINO DE TRANSMISION
                              ;POLEO DE RCIF
      GOTO  RX_DATOS          ;RCIF=1==>RECEPCION COMPLETA
                              ;RCIF=0==>RSR ES LEIDO Y ESTA VACIO
      BCF   PIR1, RCIF
      BTFSC RCSTA, FERR
      GOTO  ERR_FERR
      BTFSC RCSTA, OERR
      GOTO  ERR_OERR

      MOVF  BYTE_COUNTER, W
      XORLW LIM_BANK0
      BTFSC STATUS, Z         ;Z=0 DIFERENTES=>BANCO0
      CALL  CAMB_BANK1        ;Z=1 IGUALES=>BANCO1

      MOVF  RCREG, W          ;COPIA EL DATO RECIBIDO A W
      CALL  COMP_RX
      MOVWF INDF              ;COPIA W A UNA LOCALIDAD
      INCF  FSR               ;INCREMENTA EL FSR
      incf  byte_counter            ;incrementa el contador
      GOTO  RX_DATOS


;=====================================================================
;                        TANSMISION
;=====================================================================
```

```
TX_DATOS
      BCF   STATUS, RP0
      bcf   rcsta, cren       ;deshabilita la recepcion
      MOVLW BYTE_PARO         ;GUARDA COMO DATO FINAL AL BYTE DE PARO
      MOVWF INDF
      clrf  byte_counter            ;inicializa el contador
      MOVLW DATOS_PTR0
      MOVWF FSR
TX_AUN
      movf  byte_counter, w
      xorlw lim_bank0
      btfsc status, z         ;Z=0 DIFERENTES=>BANCO0
      call  camb_bank1        ;Z=1 IGUALES=>BANCO1

      MOVF  INDF, W           ;EL DATO APUNDADO POR FSR, SE COPIA A W
      CALL  COMP_TX
      INCF  FSR
      incf  byte_counter
      MOVWF TXREG
ESPERA                              ;ESPERA TERMINO DE TRANSMISION
      BCF   STATUS, RP0
      BTFSS PIR1, TXIF
      GOTO  ESPERA

      BCF   PIR1, TXIF
      GOTO  TX_AUN


;=====================================================================
;                          SUBRUTINA TIMER
;=====================================================================
TIMER
      movwf   w_temp                ; save off current W register contents
      movf  STATUS,w                ; move status register into W register
      movwf status_temp             ; save off contents of STATUS register

      CLRF  STATUS                  ;Borra el registro STATUS
      ;CLRF PORTB
      ;MOVLW      NO_SEG
      ;MOVWF      SEG_COUNTER
      CLRF  T1CON             ;Borra el registro T1CON
      MOVLW 0x0A              ;Preescaldor 1:1, Clk externo,
      MOVWF T1CON             ;sincroniza Clk, stop Timer1
      BCF   PIR1, TMR1IF            ;Desactiva TMR1IF


      BSF   STATUS, RP0
      ;CLRF TRISB
      BSF   PIE1,TMR1IE       ;Habilita el TMR1IE

AGAIN
      BCF   STATUS, RP0
      BCF   PIR1, TMR1IF
      MOVLW 0x00
      MOVWF TMR1L             ;Carga el TMR1L con 0x00
      MOVLW 0x80
      MOVWF TMR1H             ;Carga el TMR1H con 0x80
```

```
      BSF   T1CON, TMR1ON           ;Start Timer1

WAIT_SEC                      ;ESPERA 1 SEGUNDO
      BTFSS PIR1, TMR1IF
      GOTO  WAIT_SEC

      DECF  SEC_COUNTER       ;DECREMENTA EL CONTADOR DE SEGUNDOS
      BTFSS STATUS, Z
      GOTO  AGAIN


      movf    status_temp,w    ; retrieve copy of STATUS register
      movwf STATUS             ; restore pre-isr STATUS register contents
      MOVF  W_TEMP, W          ; restore pre-isr W register contents
      RETURN


;=====================================================================
;                       SUBRUTINA CAMB_BANK1
;=====================================================================
CAMB_BANK1
      movwf   w_temp               ; save off current W register contents
      movf  STATUS,w               ; move status register into W register
      movwf status_temp            ; save off contents of STATUS register

      MOVLW DATOS_PTR1
      MOVWF FSR

      movf    status_temp,w    ; retrieve copy of STATUS register
      movwf STATUS             ; restore pre-isr STATUS register contents
      MOVF  W_TEMP, W          ; restore pre-isr W register contents
      RETURN



;=====================================================================
;                       SUBRUTINA SEL_INFO
;=====================================================================
SEL_INFO1
      movwf   w_temp               ; save off current W register contents
      movf  STATUS,w               ; move status register into W register
      movwf status_temp            ; save off contents of STATUS register



      movf    status_temp,w   ; retrieve copy of STATUS register
      movwf STATUS            ; restore pre-isr STATUS register contents
      MOVF  W_TEMP, W         ; restore pre-isr W register contents
      RETURN

;=====================================================================
;                       SUBRUTINA COMP_RX
;=====================================================================

COMP_RX                             ;SUBRUTINA QUE COMPARA W CON ASTERISCO
      movwf   w_temp               ; save off current W register contents
      movf  STATUS,w               ; move status register into W register
```

```
      movwf status_temp            ; save off contents of STATUS register

      MOVF  W_TEMP, W
      XORLW BYTE_PARO         ;COMPRUBA SI W = BYTE_PARO
      BTFSC STATUS, Z         ;Z=0 => DIFERENTES; Z=1 => IGUALES
      GOTO  TX_DATOS          ;INICIA LA TRANSMISION

      movf    status_temp,w   ; retrieve copy of STATUS register
      movwf STATUS            ; restore pre-isr STATUS register contents
      MOVF  W_TEMP, W         ; restore pre-isr W register contents
      RETURN

;=====================================================================
;                      SUBRUTINA COMP_TX
;=====================================================================

COMP_TX                             ;SUBRUTINA QUE COMPARA W CON ASTERISCO
      movwf   w_temp                ; save off current W register contents
      movf  STATUS,w                ; move status register into W register
      movwf status_temp             ; save off contents of STATUS register

      MOVF  W_TEMP, W
      XORLW BYTE_PARO         ;COMPRUBA SI W = BYTE_PARO
      BTFSC STATUS, Z         ;Z=0 => DIFERENTES; Z=1 => IGUALES
      GOTO  REST_BP                 ;RESTAURA EL BYTE DE PARO

      movf    status_temp,w   ; retrieve copy of STATUS register
      movwf STATUS            ; restore pre-isr STATUS register contents
      MOVF  W_TEMP, W         ; restore pre-isr W register contents
      RETURN


;=====================================================================
;                    RESTAURA EL BYTE DE PARO
;=====================================================================

REST_BP

      movf    status_temp,w   ; retrieve copy of STATUS register
      movwf STATUS            ; restore pre-isr STATUS register contents
      MOVF  W_TEMP, W         ; restore pre-isr W register contents

      INCF  FSR
      incf  byte_counter
      MOVWF TXREG
ESPERA1                             ;ESPERA TERMINO DE TRANSMISION
      BCF   STATUS, RP0
      BTFSS PIR1, TXIF
      GOTO  ESPERA1
      goto  load
;=====================================================================
;                       ERRORES
;=====================================================================

err_ferr
      bcf   status, rp0
      movlw 0x01
```

```
      movwf portb
      goto  error_rx

err_oerr
      bcf   status, rp0
      movlw 0x02
      movwf portb
      goto  error_rx

error_rx
      goto  error_rx


END                                 ;directive 'end of program'
```