

; INICIALIZACIÓN DE LOS PUERTOS Y DE LA BANDERA.

INICIALIZACION

```

NOP
BCF      STATUS, RP1
BCF      STATUS, RP0      ; CAMBIA AL BANCO 0
CLRF    PORTA             ; INICIALIZA EL PUERTO A
CLRF    PORTC             ; INICIALIZA EL PUERTO C
BSF     STATUS, RP0      ; SELECCIONA EL BANCO 1
MOVLW   B'11111111'
MOVWF   TRISA             ; CONFIGURA EL PUERTO A COMO ENTRADAS
MOVLW   B'11110000'
MOVWF   TRISC            ; CONFIGURA EL PUERTO C COMO ENTRADAS
                           ; Y SALIDAS
MOVLW   B'00001111'
MOVWF   ADCON1           ; JUSTIFICA A LA IZQUIERDA, AN0 ANALÓGICA,
                           ; AN<7:1> COMO DIGITAL, RA3 COMO VREF+ Y
                           ; RA2 COMO VREF-
BCF     STATUS, RP0      ; BANCO 0
MOVLW   B'01000001'
MOVWF   ADCON0           ; FOSC/8, RA0 CANAL DE ENTRADA ANALÓGICO,
                           ; A/D ENCENDIDO
MOVLW   B'00000000'
MOVWF   BANDERA         ; INICIALIZACIÓN DE LA BANDERA
    
```

MAIN

```

CALL    CONVERSION      ; LLAMA A LA SUBROUTINA DE CONVERSION
MOVLW   B'11111111'
MOVWF   TIEMPO
    
```

WAIT

```

MOVLW   B'11111111'    ; ENTRA EN UN CICLO DE ESPERA INTEGRADO
MOVWF   RETRASO        ; POR WAIT, ESPERA Y CICLO; CON EL FIN DE
DECFSZ  TIEMPO, 1     ; ASEGURAR QUE EL PIC NO RECIBA DOS VECES
GOTO    ESPERA        ; LA MISMA SEÑAL DEL RECEPTOR.
    
```

```

GOTO    MAIN
    
```

ESPERA

```

MOVLW   B'00001111'
MOVWF   DELAY
DECFSZ  RETRASO, 1
GOTO    CICLO
    
```

```

GOTO    WAIT
    
```

CICLO

```

DECFSZ  DELAY, 1
    
```

```

GOTO      CICLO

GOTO      ESPERA

CONVERSION

NOP
NOP
BSF      ADCON0, GO      ; ESPERA EL TIEMPO MÍNIMO DE ADQUISICIÓN
                        ; EMPIEZA LA CONVERSIÓN A/D

CHECA_CONV

BTFSC    ADCON0, GO
GOTO     CHECA_CONV      ; ESPERA QUE SE COMPLETE LA CONVERSIÓN

BCF      STATUS, RP0      ; BANCO 0
MOVWF   ADRESH, W        ; PONE EN W EL RESULTADO DE CONVERSIÓN
BSF     STATUS, C
SUBLW   B'01110110'      ; HACE LA RESTA DE VALOR - W
BTFSC   STATUS, C        ; VERIFICA LA BANDERA DE CARRY PARA
                        ; ASEGURAR QUE LA SEÑAL NO SEA RUIDO

GOTO     CHECA

BTFSS   BANDERA, 0      ; CHECA SI ESTÁ EN EL AREA DE CONTROL
GOTO    CONTROL        ; SI ES 0, ESTÁ ENTRANDO Y VA A CONTROL
GOTO    BAND           ; SI ES 1, ESTÁ SALIENDO Y VA A BAND

CHECA      ; ESTA SUBROUTINA VERIFICA SI EL VEHÍCULO
           ; ESTÁ DENTRO O FUERA DEL ÁREA DE
           ; DE CONTROL.
BTFSC    BANDERA, 0
GOTO     CONTROL        ; DE ESTAR DENTRO DEL AREA VA A CONTROL
GOTO     VELOCIDAD     ; SI ESTA FUERA DEL AREA VA A VELOCIDAD

BAND

MOVLW   B'00000000'      ; ESTA SUBROUTINA INDICA QUE EL VEHÍCULO
MOVWF   BANDERA        ; ESTÁ SALIENDO DEL ÁREA DE CONTROL, POR
GOTO    CHECA          ; LO QUE HACE BANDERA = 0 PARA INDICARLO

VELOCIDAD

BCF      STATUS, RP0      ; BANCO 0
BTFSS   PORTC, 4        ; VERIFICA SI ESTA EN PRIMERA VELOCIDAD
BTFSS   PORTC, 5        ; VERIFICA SI ESTA EN SEGUNDA VELOCIDAD
GOTO    PWM1            ; LLAMA A LA PRIMER VELOCIDAD
GOTO    PWM2            ; LLAMA A LA SEGUNDA VELOCIDAD

CONTROL

MOVLW   B'00000001'      ; SE MODIFICA LA BANDERA PARA INDICAR
MOVWF   BANDERA        ; QUE SE ESTÁ DENTRO DEL ÁREA DE CONTROL
BCF     STATUS, RP0      ; BANCO 0

```

```

BTFFS      PORTC, 4      ; VERIFICA SI ESTA EN PRIMERA VELOCIDAD
BTFFS      PORTC, 5      ; VERIFICA SI ESTA EN SEGUNDA VELOCIDAD
GOTO       PWM1          ; LLAMA A LA PRIMER VELOCIDAD
GOTO       PWM3          ; LLAMA A LA VELOCIDAD DE CONTROL
    
```

```

*****
;* PWM1:
;*   PWM PERIOD = [PR2+1]*4*TOSC*TMR2 PRESCALE VALUE
;*
;*   SI:   PWM PERIOD = 10 Us
;*         FOSC = 4 MHz
;*         TOSC = 250 nS
;*         TMR2 PRESCALE VALUE = 1
;*
;*   DESPEJANDO LA ECUACION ANTERIOR
;*     PR2 = (PWM PERIOD/(4*TOSC*TMR2 PRESCALE VALUE))-1
;*     PR2 = 9
;*
;*   CICLO DE TRABAJO:
;*     PWM DUTY CYCLE=(CCPR1L:CCP1CON<5:4>)*TOSC*TMR2 PRESCALE
;*
;*   USANDO LOS SIGUIENTES VALORES:
;*     PWM DUTY CYCLE = 43% DEL PERIODO = 4.3 uS
;*     TOSC = 250 nS
;*     TMR2 PRESCALE VALUE = 1
;*   DESPEJANDO (CCPR1L:CCP1CON<5:4>) DE LA ECUACION
;*     (CCPR1L:CCP1CON<5:4>) = 01
;*     (CCPR1L:CCP1CON<5:4>) = 00000100
*****
    
```

PWM1

```

BCF        STATUS, RP0      ; BANCO 0
MOVLW     B'00000000'
MOVWF     CCP1CON           ; DESHABILITAMOS EL PWM ANTERIOR
MOVWF     T2CON             ; INICIALIZA EL TMR2.
CLRF      PORTB             ; INICIALIZA EL PUERTO B.
BSF       STATUS, RP0      ; BANCO 1
MOVLW     B'00001001'
MOVWF     PR2               ; CUANDO EL TMR2 = 9 TERMINA EL PERIODO
BCF       STATUS, RP0      ; BANCO 0
MOVLW     B'00000100'
MOVWF     CCPR1L           ; ASIGNA LOS 8 BITS MÁS SIGNIFICATIVOS DEL
                           ; CICLO DE TRABAJO AL CCPR1L

BCF       CCP1CON, CCP1X
BSF       CCP1CON, CCP1Y   ; ASIGNA LOS 2 BITS MENOS SIGNIFICATIVOS
                           ; DEL CICLO DE TRABAJO AL CCP1X Y CCP1Y
    
```

; DEFINIENDO LA SALIDA DEL PWM

```

BSF       STATUS, RP0      ; BANCO 1
MOVLW     B'00000000'     ; EL PIN 2 DEL TRIS C DEFINIDO COMO SALIDA
ANDWF     TRISC
BCF       STATUS, RP0      ; BANCO 0
    
```

```

MOVLW    B'00000100'
MOVWF    T2CON           ; PRESCALADOR TMR2 = 1, ENCIENDE EL TMR2
MOVWF    CCP1CON, W
ANDLW    B'00110000'     ; MÁSCARA PARA NO ALTERAR EL VALOR DEL
                           ; CICLO DE TRABAJO

IORLW    B'00001111'
MOVWF    CCP1CON         ; SE HABILITA EL MODO PWM
RETURN
    
```

```

;*****
;* PWM2:
;*
;*
;*     CICLO DE TRABAJO:
;*     PWM DUTY CYCLE=(CCPR1L:CCP1CON<5:4>)*TOSC*TMR2 PRESCALE
;*
;*
;*     USANDO LOS SIGUIENTES VALORES:
;*     PWM DUTY CYCLE = 60% DEL PERIODO = 6 uS
;*     TOSC = 250 nS
;*     TMR2 PRESCALE VALUE = 1
;*     DESPEJANDO (CCPR1L:CCP1CON<5:4>) DE LA ECUACION
;*     (CCPR1L:CCP1CON<5:4>) = 36
;*     (CCPR1L:CCP1CON<5:4>) = 0000100100
;*****
    
```

PWM2

```

BCF      STATUS, RP0     ; BANCO 0
MOVLW    B'00000000'
MOVWF    CCP1CON         ; DESHABILITAMOS EL PWM ANTERIOR
MOVWF    T2CON           ; INICIALIZAMOS EL TMR2.
BSF      STATUS, RP0     ; BANCO 1
MOVLW    B'00001001'
MOVWF    PR2             ; CUANDO EL TMR2 = 9 TERMINA EL PERIODO
BCF      STATUS, RP0     ; BANCO 0
MOVLW    B'00000110'
MOVWF    CCP1L           ; ASIGNA LOS 8 BITS MÁS SIGNIFICATIVOS DEL
                           ; CICLO DE TRABAJO AL CCP1L

BCF      CCP1CON, CCP1X
BCF      CCP1CON, CCP1Y  ; ASIGNA LOS 2 BITS MENOS SIGNIFICATIVOS
                           ; DEL CICLO DE TRABAJO AL CCP1X Y CCP1Y
    
```

; DEFINIENDO LA SALIDA DEL PWM

```

BSF      STATUS, RP0     ; BANCO 1
MOVLW    B'00000000'     ; EL TRIS C QUEDA DEFINIDO COMO SALIDA
ANDWF    TRISC
BCF      STATUS, RP0     ; BANCO 0
MOVLW    B'00000100'
MOVWF    T2CON           ; PRESCALADOR TMR2 = 1, ENCIENDE EL TMR2
MOVWF    CCP1CON, W
ANDLW    B'00110000'     ; MÁSCARA PARA NO ALTERAR EL VALOR DEL
                           ; CICLO DE TRABAJO

IORLW    B'00001111'
MOVWF    CCP1CON         ; SE HABILITA EL MODO PWM
    
```

RETURN

```

;*****
;* PWM3:
;*
;*     CICLO DE TRABAJO:
;*     PWM DUTY CYCLE=(CCPR1L:CCP1CON<5:4>)*TOSC*TMR2 PRESCALE
;*
;*     USANDO LOS SIGUIENTES VALORES:
;*     PWM DUTY CYCLE = 48% DEL PERIODO = 4.8 uS
;*     TOSC = 250 nS
;*     TMR2 PRESCALE VALUE = 1
;*     DESPEJANDO (CCPR1L:CCP1CON<5:4>) DE LA ECUACION
;*     (CCPR1L:CCP1CON<5:4>) = 10
;*     (CCPR1L:CCP1CON<5:4>) = 0000001010
;*****

```

PWM3

```

BSF     STATUS, RP0      ; BANCO 0
MOVLW  B'00000000'
MOVWF  CCP1CON          ; DESHABILITAMOS EL PWM ANTERIOR
MOVWF  T2CON            ; INICIALIZAMOS EL TMR2.
BSF     STATUS, RP0      ; BANCO 1
MOVLW  B'00001001'
MOVWF  PR2              ; CUANDO EL TMR2 = 9 TERMINA EL PERIODO
BSF     STATUS, RP0      ; BANCO 0
MOVLW  B'00000100'
MOVWF  CCP1L            ; ASIGNA LOS 8 BITS MÁS SIGNIFICATIVOS DEL
                        ; CICLO DE TRABAJO AL CCP1L

BSF     CCP1CON, CCP1X
BSF     CCP1CON, CCP1Y  ; ASIGNA LOS 2 BITS MENOS SIGNIFICATIVOS
                        ; DEL CICLO DE TRABAJO AL CCP1X Y CCP1Y

```

; DEFINIENDO LA SALIDA DEL PWM

```

BSF     STATUS, RP0      ; BANCO 1
MOVLW  B'00000000'      ; EL TRIS C QUEDA DEFINIDO COMO SALIDA
ANDWF  TRISC
BSF     STATUS, RP0      ; BANCO 0
MOVLW  B'00000100'
MOVWF  T2CON            ; PRESCALADOR TMR2 = 1, ENCIENDE EL TMR2
MOVF   CCP1CON, W
ANDLW  B'00110000'      ; MÁSCARA PARA NO ALTERAR EL VALOR DEL
                        ; CICLO DE TRABAJO

IORLW  B'00001111'
MOVWF  CCP1CON          ; SE HABILITA EL MODO PWM
RETURN

```

END



PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

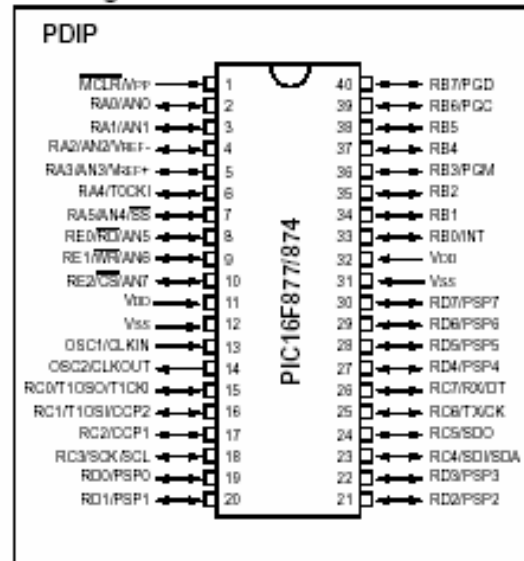
Devices Included in this Data Sheet:

- PIC16F873
- PIC16F876
- PIC16F874
- PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
Up to 368 x 8 bytes of Data Memory (RAM)
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and
Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC
oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM
technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two
pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature
ranges
- Low-power consumption:
 - < 0.6 mA typical @ 3V, 4 MHz
 - 20 µA typical @ 3V, 32 kHz
 - < 1 µA typical standby current

Pin Diagram



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,
can be incremented during SLEEP via external
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period
register, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master
mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver
Transmitter (USART/SCI) with 9-bit address
detection
- Parallel Slave Port (PSP) 8-bits wide, with
external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for
Brown-out Reset (BOR)