



# ELECTRONICA MICROCONTROLADA

*Profesor Gonzalo A. Vera.*

*Estudiante Macarena A. Carballo*

## TP semana 10: Entrenador Básico

### Objetivos

- Familiarizarse con un entorno de desarrollo de bajo nivel.
- Introducir las primeras practicas con micros PIC de 8 bit.
- Introducción a las primeras instrucciones en assembler.
- Realizar desarrollos básicos con el PIC16F648.

### Desarrollo

- 1) Identificar la palabra de configuración y explicar que representa cada fusible.
- 2) Completar el código provisto en el entrenador básico PIC16f648, de manera que el display haga el conteo de 0 a 9. Se permite utilizar como delay la instrucción nop.

Esta práctica se debe desarrollar y presentar en el monorepositorio del alumno establecido en la organización de la materia, como TP10 respetando la estructura ABP:

### TP10

- ☐ A requisitos; acá va este documento
- ☐ B investigación; hoja de datos del pic16f648, otros
- ☐ C prototipo; desarrollo de esta practica
- ☐ D presentación; video presentando la solución (3m)
- ☐ README.md; Readme acorde

# 1

Estas configuraciones son directivas de compilación usadas en los microcontroladores PIC para definir cómo se comportarán ciertos aspectos del hardware del microcontrolador.

`_CONFIG _CP_OFF & _CPD_OFF & _LVP_OFF & _BODEN_ON & _MCLRE_OFF & _PWRTE_ON & _WDTE_OFF & _INTOSC_OSC_NOCLKOUT`

-----CONFIG Options-----

`_INTOSC_OSC_NOCLKOUT EQUH'3FFC'`; INTOSC oscillator: I/O function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN

`_WDTE_OFF EQUH'3FFB'`; WDT disabled

`_PWRTE_ON EQUH'3FF7'`; PWRT enabled

`_MCLRE_OFF EQUH'3FDF'`; RA5/MCLR/VPP pin function is digital input, MCLR internally tied to VDD

`_BODEN_ON EQUH'3FFF'`; BOD enabled

`_LVP_OFF EQUH'3F7F'`; RB4/PGM pin has digital I/O function, HV on MCLR must be used for programming

`_CPD_OFF EQUH'3FFF'`; Data memory code protection off

`_CP_OFF EQUH'3FFF'`; Code protection off

## **`_CP_OFF` (Code Protection Off):**

**Propósito:** Deshabilita la protección de código.

**Función:** Permite que el código almacenado en la memoria del programa sea leído y copiado. Sin esta protección, el código del microcontrolador puede ser copiado, lo que puede ser útil durante el desarrollo, pero inseguro para productos finales.

## **`_CPD_OFF` (Data EEPROM Code Protection Off):**

**Propósito:** Deshabilita la protección de código de la EEPROM de datos.

**Función:** Permite que el contenido de la memoria EEPROM de datos sea leído y copiado. Similar a la protección de código, pero para la memoria de datos.

#### **\_LVP\_OFF (Low Voltage Programming Off):**

Propósito: Deshabilita la programación en baja tensión.

Función: La programación del microcontrolador debe hacerse a un voltaje más alto, lo que es más común en entornos de producción y menos propenso a errores.

#### **\_BODEN\_ON (Brown-Out Detect Enable):**

Propósito: Habilita la detección de caídas de tensión (brown-out).

Función: Resetea el microcontrolador si la tensión de alimentación cae por debajo de un umbral específico, evitando comportamientos erráticos debido a insuficiente energía.

#### **\_MCLR\_OFF (Master Clear Reset Pin Disable):**

Propósito: Deshabilita el pin MCLR como reset.

Función: El pin MCLR puede ser utilizado como un pin de entrada/salida general en lugar de ser usado exclusivamente para reiniciar el microcontrolador.

#### **\_PWRTE\_ON (Power-Up Timer Enable):**

Propósito: Habilita el temporizador de encendido.

Función: Introduce un retraso durante el encendido del microcontrolador, permitiendo que la tensión de alimentación se estabilice antes de comenzar a ejecutar el código.

#### **\_WDTE\_OFF (Watchdog Timer Enable Off):**

Propósito: Deshabilita el temporizador Watchdog.

Función: El temporizador Watchdog no está habilitado, por lo que no reiniciará el microcontrolador automáticamente en caso de que se cuelgue.

#### **\_INTOSC\_OSC\_NOCLKOUT (Internal Oscillator, No Clock Out):**

Propósito: Selecciona el oscilador interno como la fuente de reloj, sin salida de reloj externo.

Función: El microcontrolador usará su oscilador interno para generar el reloj, y no se producirá una señal de reloj en los pines externos.

Estas configuraciones permiten personalizar el comportamiento del microcontrolador para adaptarse a las necesidades específicas de una aplicación.

## 2

```
; Main o Principal ( Desde el Vector de Reset)
;=====
Principal
; Setup
; bcf STATUS, RP0 ; selecciono el banco 0
; CLRF PORTA
; MOVLW 0x07 ; Turn comparators off and
; MOVWF CMCON ; enable pins comparator function

; clrf PORTB
; bsf STATUS, RP0 ; Seleccionar banco 1
; movlw b'00000000' ; Configurar todos los pines del puerto B como entrada
; movwf TRISB
; movlw b'00000000' ; Configurar todos los pines del puerto A como salida
; movwf TRISA
; bcf STATUS, RP0 ; Seleccionar banco 0

; Loop

Loop
; movlw h'BF' ; cargo w con el digito '0'
; movwf PORTB
; call delay
; nop

; movlw h'86' ; cargo w con el digito '1'
; movwf PORTB
; nop

; movlw h'DB' ; cargo w con el digito '2'
; movwf PORTB
; nop

; movlw h'CF' ; cargo w con el digito '3'
; movwf PORTB
```

```
movlw h'E6'           ; cargo w con el digito '4'
movwf PORTB
nop

movlw h'ED'           ; cargo w con el digito '5'
movwf PORTB
nop

movlw h'FD'           ; cargo w con el digito '6'
movwf PORTB
nop

movlw h'87'           ; cargo w con el digito '7'
movwf PORTB
nop

movlw h'FF'           ; cargo w con el digito '8'
movwf PORTB
nop

movlw h'EF'           ; cargo w con el digito '9'
movwf PORTB
nop

;movf PORTB, W        ; Leer el valor del puerto B (estado de Los switches DIP)
;movwf PORTA          ; Escribir el valor en el puerto A (control de Los LEDs)
;goto $              ; Repetir el Loop
;
; Librerias .INC
;=====
END
;=====
```