CURSO DE PROGRAMACIÓN E INTERFAZ DE MICROCONTROLADORES PIC

Video 00 — Introducción al Curso de Programación e Interfaz de Microcontroladores PIC · Plataforma Oficial: cursosmecatronica.com

Módulo: Introducción General al Curso de Programación e Interfaz de Microcontroladores PIC
Enfoque de Ingeniería: Panorama Integral: Microprocesadores, PIC16F84A, PicBasic Pro, Interrupciones, Teclado/LCD, E/S Digitales, Diseño y Depuración

```
f_ciclo = f_osc / 4 | T_inst(4MHz) = 1 us | I_pin(max) = 25 mA | R_LED = (V_DD - V_f) / I_f
```

1. Desarrollo Técnico Diapositiva por Diapositiva

Diapositiva 01: Bienvenidos al Curso de Programación e Interfaz de Microcontroladores PIC

Microprocesadores, Microcomputadoras, Arquitectura del PIC16F84 y Programación en cursosmecatronica.com

- **Turno 1 (Orus):** Bienvenidos al curso completo de Programación e Interfaz de Microcontroladores PIC en cursosmecatronica.com.
- **Turno 2 (Aoede):** Estudiaremos desde la arquitectura de microprocesadores y microcomputadoras hasta la estructura interna y programación del PIC dieciséis efe ochenta y cuatro.
- **Turno 3 (Orus):** Dominaremos los fundamentos y ejemplos prácticos de PicBasic Pro junto con la atención precisa de interrupciones en tiempo real.
- **Turno 4 (Aoede):** Todo explicado paso a paso con esquemas claros, ecuaciones de ingeniería y código fuente listo para implementar.

Diapositiva 02: Interfaz con Periféricos, Método de Diseño e Ingeniería Práctica

Teclado Matricial, Pantalla LCD, Entradas y Salidas Digitales, Fuentes, Baterías y Depuración en cursosmecatronica.com

- **Turno 1 (Orus):** Conectaremos teclados numéricos matriciales, pantallas L C D alfanuméricas y etapas seguras de entrada y salida digital de potencia.
- **Turno 2 (Aoede):** Además, aplicaremos un método sistemático de diseño de sistemas con microcontrolador desde los requerimientos hasta la validación.
- **Turno 3 (Orus):** Cerraremos con procedimientos de depuración, fuentes reguladas, características de baterías y buenas prácticas de prototipado electrónico.
- **Turno 4 (Aoede):** Encuentra todos los recursos en cursosmecatronica.com y comencemos con la Parte uno sobre microprocesadores y microcontroladores.

2. Código Fuente en PicBasic Pro (picbasic_pro_video_00.pbp)

```
' ================================================================================
' CURSO: PROGRAMACION E INTERFAZ DE MICROCONTROLADORES PIC (cursosmecatronica.com)
' Video 00 – Introducción al Curso de Programación e Interfaz de Microcontroladores PIC
' Enfoque: Panorama Integral: Microprocesadores, PIC16F84A, PicBasic Pro, Interrupciones, Teclado/LCD, E/S
Digitales, Diseño y Depuración
' Ecuaciones Clave: f_ciclo = f_osc / 4  |  T_inst(4MHz) = 1 us  |  I_pin(max) = 25 mA  |  R_LED = (V_DD - V_f)
/ I_f
' Microcontrolador Objetivo: PIC16F84A @ 4.000 MHz (1 ciclo = 1 us)
' ================================================================================
@ DEVICE pic16F84A, XT_OSC, WDT_OFF, PWRT_ON, PROTECT_OFF
Include "modedefs.bas"
DEFINE OSC 4

' --- Declaracion de Pines y Variables ---
EntradaSensor  VAR PORTA.0
SalidaRele     VAR PORTB.0
LedHeartbeat   VAR PORTB.7
TxDiagnostico  VAR PORTA.1
ContadorCiclos VAR WORD
Estado         VAR BYTE

' --- Inicializacion Segura de Puertos (Estado antes que Direccion TRIS) ---
PORTA = %00000000
PORTB = %00000000
OPTION_REG.7 = 0            ' Pull-ups internos activos en PORTB
TRISA = %00000001           ' RA0 entrada, resto salidas
TRISB = %00000000           ' PORTB como salidas digitales
ContadorCiclos = 0
Estado = 0

' --- Lazo Principal Determinista ---
LazoPrincipal:
    TOGGLE LedHeartbeat
    ContadorCiclos = ContadorCiclos + 1
    IF EntradaSensor = 0 THEN
        PAUSE 20            ' Filtro antirrebote 20 ms
        IF EntradaSensor = 0 THEN
            HIGH SalidaRele
            Estado = 1
        ENDIF
    ELSE
        LOW SalidaRele
        Estado = 0
    ENDIF
    SEROUT TxDiagnostico, T2400, ["V00 Ciclo=", #ContadorCiclos, " Est=", #Estado, 13, 10]
    PAUSE 100
    GOTO LazoPrincipal
    END
```

3. Referencia en Ensamblador MPASM (ensamblador_mpasm_video_00.asm)

```
; ==========================================================================
; CURSO: PROGRAMACION E INTERFAZ DE MICROCONTROLADORES PIC (cursosmecatronica.com)
; Video 00 – Introducción al Curso de Programación e Interfaz de Microcontroladores PIC – Referencia de Bajo Nivel MPASM (PIC16F84A)
; ==========================================================================
    LIST        P=16F84A
    #INCLUDE    <P16F84A.INC>
    __CONFIG    _XT_OSC & _WDT_OFF & _PWRTE_ON & _CP_OFF

    CBLOCK  0x0C
        W_TEMP
        STATUS_TEMP
        CONTADOR
    ENDC

    ORG     0x0000
    GOTO    INICIO

    ORG     0x0004              ; Vector de Interrupcion Hardware
    MOVWF   W_TEMP              ; Salvado de contexto
    SWAPF   STATUS, W
    MOVWF   STATUS_TEMP
    BCF     INTCON, INTF        ; Limpiar bandera RB0/INT
    SWAPF   STATUS_TEMP, W
    MOVWF   STATUS
    SWAPF   W_TEMP, F
    SWAPF   W_TEMP, W
    RETFIE

INICIO:
    BCF     STATUS, RP0         ; Banco 0
    CLRF    PORTA
    CLRF    PORTB
    BSF     STATUS, RP0         ; Banco 1
    MOVLW   B'00000001'
    MOVWF   TRISA               ; RA0 entrada
    CLRF    TRISB               ; PORTB salidas
    BCF     STATUS, RP0         ; Regresar a Banco 0

BUCLE:
    BTFSS   PORTA, 0
    BSF     PORTB, 0
    BTFSC   PORTA, 0
    BCF     PORTB, 0
    GOTO    BUCLE
    END
```

4. Equivalencia en Lenguaje C XC8 (firmware_xc8_video_00.c)

```
/* ============================================================================
 * CURSO: PROGRAMACION E INTERFAZ DE MICROCONTROLADORES PIC (cursosmecatronica.com)
 * Video 00 – Introducción al Curso de Programación e Interfaz de Microcontroladores PIC – Equivalencia en C
Embebido (XC8)
 * ============================================================================ */
#include <xc.h>
#define _XTAL_FREQ 4000000UL

#pragma config FOSC = XT, WDTE = OFF, PWRTE = ON, CP = OFF

void main(void) {
    PORTA = 0x00;
    PORTB = 0x00;
    OPTION_REGbits.nRBPU = 0;   /* Pull-ups internos activos en PORTB */
    TRISA = 0b00000001;         /* RA0 como entrada */
    TRISB = 0b00000000;         /* RB0..RB7 como salidas */

    while (1) {
        PORTBbits.RB7 ^= 1;     /* LED Heartbeat */
        if (PORTAbits.RA0 == 0) {
            __delay_ms(20);
            if (PORTAbits.RA0 == 0) {
                PORTBbits.RB0 = 1;
            }
        } else {
            PORTBbits.RB0 = 0;
        }
        __delay_ms(100);
    }
}
```