# CURSO DE PROGRAMACIÓN E INTERFAZ DE MICROCONTROLADORES PIC

Video 09 — Resumen y Cierre del Curso de Programación e Interfaz de Microcontroladores PIC · Plataforma Oficial: cursosmecatronica.com

**Módulo:** Resumen Ejecutivo y Cierre del Curso de Microcontroladores PIC
**Enfoque de Ingeniería:** Integración Total Hardware-Software en Sistemas Embebidos PIC y Siguientes Proyectos en cursosmecatronica.com

```
Sistema Embebido Confiable = Hardware Protegido + Firmware Modular Determinista + Depuracion Sistematica
```

## 1. Desarrollo Técnico Diapositiva por Diapositiva

### Diapositiva 01: Síntesis de Arquitectura PIC, PicBasic Pro e Interrupciones

Del Microprocesador y el PIC16F84 al Dominio de Comandos Avanzados y Tiempo Real en cursosmecatronica.com

- **Turno 1 (Orus):** Felicitaciones por completar este recorrido integral por la programación e interfaz de microcontroladores PIC.
- **Turno 2 (Aoede):** Dominamos la arquitectura Harvard y RISC, los bancos de memoria, registros especiales y grabación en circuito del PIC dieciséis efe ochenta y cuatro.
- **Turno 3 (Orus):** Programamos en PicBasic Pro desde variables, estructuras y subrutinas hasta control de servomotores, comunicación serial e interrupciones.
- **Turno 4 (Aoede):** Recuerda que tienes todas las guías y códigos descargables disponibles en cursosmecatronica.com.

### Diapositiva 02: Integración de Periféricos, Método de Ingeniería y Siguientes Proyectos

Teclado, LCD, Etapas de Potencia, Depuración, Fuentes, Baterías y Comunidad en cursosmecatronica.com

- **Turno 1 (Orus):** También diseñamos interfaces robustas con teclados matriciales, pantallas L C D, optoacopladores, transistores, MOSFETs y relés protegidos.
- **Turno 2 (Aoede):** Integramos el método sistemático de ocho pasos junto con protocolos reales de depuración, fuentes reguladas, baterías y diseño en placa impresa.
- **Turno 3 (Orus):** Ahora estás preparado para transformar cualquier requerimiento de automatización en un sistema embebido confiable y profesional.
- **Turno 4 (Aoede):** Sigue explorando más laboratorios y especializaciones en cursosmecatronica.com. ¡Hasta el próximo proyecto!

## 2. Código Fuente en PicBasic Pro (picbasic_pro_video_09.pbp)

```
' ================================================================================
' CURSO: PROGRAMACION E INTERFAZ DE MICROCONTROLADORES PIC (cursosmecatronica.com)
' Video 09 – Resumen y Cierre del Curso de Programación e Interfaz de Microcontroladores PIC
' Enfoque: Integración Total Hardware-Software en Sistemas Embebidos PIC y Siguientes Proyectos en
cursosmecatronica.com
' Ecuaciones Clave: Sistema Embebido Confiable = Hardware Protegido + Firmware Modular Determinista + Depuracion
Sistematica
' 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, ["V09 Ciclo=", #ContadorCiclos, " Est=", #Estado, 13, 10]
    PAUSE 100
    GOTO LazoPrincipal
    END
```

## 3. Referencia en Ensamblador MPASM (ensamblador_mpasm_video_09.asm)

```
; ==========================================================================
; CURSO: PROGRAMACION E INTERFAZ DE MICROCONTROLADORES PIC (cursosmecatronica.com)
; Video 09 – Resumen y Cierre del 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_09.c)

```
/* ============================================================================
 * CURSO: PROGRAMACION E INTERFAZ DE MICROCONTROLADORES PIC (cursosmecatronica.com)
 * Video 09 – Resumen y Cierre del 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);
    }
}
```