CURSO DE PROGRAMACIÓN E INTERFAZ DE MICROCONTROLADORES PIC

Video 01 — Parte 1: Microprocesadores, Microcomputadoras y Microcontroladores · Plataforma Oficial: cursosmecatronica.com

Módulo: Parte 1 — Microprocesadores, Microcomputadoras y Microcontroladores
Enfoque de Ingeniería: CPU/ALU/Registros, Buses de Direcciones-Datos-Control, Ciclo Fetch-Decode-Execute, Von Neumann vs Harvard y CISC vs RISC

```
N_dir = 2^B_addr | T_ciclo = 4 / f_osc | CPI_RISC = 1 (2 en saltos) | P_din = C_par · V_DD^2 · f_osc
```

1. Desarrollo Técnico Diapositiva por Diapositiva

Diapositiva 01: El Microprocesador: Unidad Central de Procesamiento en Silicio

Unidad Aritmética Lógica ALU, Unidad de Control, Registros de Trabajo y Banderas de Estado

- **Turno 1 (Orus):** Para dominar la programación de sistemas embebidos debemos distinguir primero qué es un microprocesador: se trata de una unidad central de procesamiento integrada en un solo circuito de silicio cuya misión es ejecutar instrucciones secuenciales.
- **Turno 2 (Aoede):** En su interior residen tres bloques esenciales: la unidad aritmética y lógica, encargada de sumar, restar y operar bits; el banco de registros temporales de alta velocidad; y la unidad de control, que decodifica cada instrucción.
- **Turno 3 (Orus):** Junto al acumulador o registro de trabajo existe un registro de estado cuyas banderas binarias indican si la última operación produjo un resultado igual a cero, un acarreo aritmético o un desbordamiento.
- **Turno 4 (Aoede):** Sin embargo, un microprocesador aislado no puede funcionar por sí mismo, pues carece de memoria interna de programa y de puertos físicos para leer sensores o accionar cargas externas.

Diapositiva 02: De Microprocesador a Microcomputadora: Arquitectura de Tres Buses

Bus de Direcciones Unidireccional, Bus de Datos Bidireccional, Bus de Control, Memoria y Puertos E/S

- **Turno 1 (Orus):** Cuando rodeamos a un microprocesador con chips externos de memoria de solo lectura, memoria volátil de acceso aleatorio y circuitos integrados de entrada y salida sobre una placa impresa, conformamos una microcomputadora completa.
- **Turno 2 (Aoede):** Todos estos bloques se comunican a través de tres autopistas eléctricas paralelas llamadas buses: el bus de direcciones, el bus de datos y el bus de control.
- **Turno 3 (Orus):** El bus de direcciones es unidireccional desde el procesador y determina cuántas localidades de memoria puede alcanzar: un bus de dieciséis líneas permite direccionar exactamente dos elevado a dieciséis, es decir, sesenta y cinco mil quinientas treinta y seis posiciones.
- **Turno 4 (Aoede):** Por su parte, el bus de datos es bidireccional para transportar palabras binarias hacia o desde la memoria, mientras que el bus de control sincroniza las señales de lectura, escritura, reloj e interrupción.

Diapositiva 03: El Ciclo de Ejecución: Búsqueda, Decodificación y Ejecución

Contador de Programa PC, Registro de Instrucción IR, Oscilador Maestro y Determinismo Temporal

- **Turno 1 (Orus):** Toda computadora digital opera bajo el ritmo implacable de un oscilador de reloj, repitiendo millones de veces por segundo el ciclo fundamental de búsqueda, decodificación y ejecución de instrucciones.
- **Turno 2 (Aoede):** Durante la fase de búsqueda, el contador de programa coloca en el bus la dirección de la siguiente instrucción almacenada en memoria, la copia en el registro de instrucción e incrementa automáticamente su valor.
- **Turno 3 (Orus):** En la fase de decodificación, los circuitos lógicos de la unidad de control interpretan el código binario de operación para determinar qué rutas internas de datos y qué operación aritmética deben activarse.
- **Turno 4 (Aoede):** Finalmente, en la fase de ejecución se leen los operandos necesarios, la unidad aritmética calcula el resultado, se actualizan las banderas de estado y se almacena el dato en el registro o puerto destino.

Diapositiva 04: Arquitecturas de Memoria: Von Neumann frente a Harvard

Bus Único Compartido con Cuello de Botella frente a Buses Separados de Programa y Datos

- **Turno 1 (Orus):** Existen dos filosofías clásicas para conectar el procesador con sus memorias: la arquitectura Von Neumann, típica de las computadoras personales históricas, y la arquitectura Harvard, predilecta en los microcontroladores modernos.
- **Turno 2 (Aoede):** En el modelo Von Neumann, las instrucciones del programa y las variables de datos comparten el mismo espacio de memoria y el mismo bus único, por lo que el procesador no puede leer una instrucción y un dato simultáneamente.
- **Turno 3 (Orus):** En contraste, la arquitectura Harvard separa físicamente la memoria no volátil de programa y la memoria volátil de datos, dotando a cada una de sus propios buses independientes de direcciones y de datos.
- **Turno 4 (Aoede):** Esta separación no solo duplica el ancho de banda efectivo permitiendo buscar la próxima instrucción mientras se ejecuta la actual, sino que permite que el ancho de palabra del programa sea distinto al ancho de ocho bits de los datos.

Diapositiva 05: Filosofías de Conjunto de Instrucciones: CISC frente a RISC

Instrucciones Complejas de Múltiples Ciclos frente a Repertorio Reducido de Ciclo Único y Segmentación

- **Turno 1 (Orus):** El diseño de la unidad de control también divide a los procesadores en dos familias: computadoras con conjunto de instrucciones complejo, o C I S C, y computadoras con conjunto de instrucciones reducido, conocidas como R I S C.
- **Turno 2 (Aoede):** Un procesador de conjunto complejo posee cientos de instrucciones especializadas de longitud variable que tardan un número distinto de ciclos de reloj en completarse, encareciendo el área de silicio y dificultando predecir tiempos exactos.
- **Turno 3 (Orus):** Por el contrario, un procesador de conjunto reducido como la familia PIC de gama media emplea apenas treinta y cinco instrucciones ortogonales de longitud fija, optimizadas en hardware para ejecutarse en un solo ciclo de instrucción.
- **Turno 4 (Aoede):** Al combinar el repertorio reducido con la arquitectura Harvard se logra una segmentación de dos etapas: mientras se ejecuta la instrucción presente, ya se está extrayendo la siguiente, salvo en saltos que toman dos ciclos.

Diapositiva 06: ¿Qué es un Microcontrolador? Una Microcomputadora en un Solo Chip

Integración Monolítica de CPU, Memoria Flash, RAM, EEPROM, Temporizadores y Puertos de Entrada/Salida

- **Turno 1 (Orus):** Un microcontrolador es una microcomputadora completa integrada dentro de una única pastilla monolítica de silicio, diseñada específicamente para gobernar máquinas, instrumentos y procesos físicos en lazo cerrado.
- **Turno 2 (Aoede):** Dentro del mismo encapsulado conviven la unidad central de procesamiento, la memoria no volátil de programa, la memoria de datos, temporizadores por hardware, lógica de interrupciones y puertos digitales directos hacia el exterior.
- **Turno 3 (Orus):** A diferencia de una computadora de escritorio pensada para ejecutar múltiples aplicaciones ofimáticas con gigabytes de memoria externa, el microcontrolador ejecuta un único programa de control dedicado de forma autónoma e indefinida.
- **Turno 4 (Aoede):** Gracias a esta altísima integración desaparecen los buses externos en la tarjeta electrónica, reduciendo drásticamente el tamaño físico, el consumo eléctrico, la sensibilidad al ruido electromagnético y el costo total de fabricación.

Diapositiva 07: Comparativa de Ingeniería: Microprocesador frente a Microcontrolador

Capacidad de Cómputo Masivo y Sistemas Operativos frente a Control Determinista de Pines y Bajo Consumo

- **Turno 1 (Orus):** Comparemos con rigor técnico ambos dispositivos: un microprocesador opera a frecuencias de gigahercios con buses de sesenta y cuatro bits, disipa decenas de vatios y requiere sistemas operativos complejos que no garantizan tiempos microsegundos exactos.
- **Turno 2 (Aoede):** Un microcontrolador clásico trabaja a unos pocos megahercios con registros de ocho o dieciséis bits, consume apenas unos pocos miliamperios a cinco voltios y arranca instantáneamente al aplicar energía eléctrica.
- **Turno 3 (Orus):** Además, el conjunto de instrucciones de un microcontrolador está orientado al manejo directo de bits individuales, permitiendo encender un relé, leer un interruptor o limpiar una bandera con una sola línea de código atómica.
- **Turno 4 (Aoede):** En sistemas mecatrónicos avanzados ambos mundos colaboran: un microprocesador superior puede procesar visión artificial mientras varios microcontroladores dedicados aseguran el control determinista de motores y sensores.

Diapositiva 08: Ecosistema de Microcontroladores, Encapsulados y Tecnologías de Memoria

Evolución de OTP y EPROM Ultravioleta a Memoria Flash Eléctricamente Reprogramable y Encapsulados DIP y SMD

- **Turno 1 (Orus):** Históricamente los primeros microcontroladores almacenaban su programa en memorias programables una sola vez o en memorias borrables con lámpara de luz ultravioleta a través de una ventana de cuarzo, lo que ralentizaba enormemente el desarrollo.
- **Turno 2 (Aoede):** La revolución llegó con la tecnología de memoria Flash eléctricamente borrable y reprogramable, que permite grabar, corregir y volver a grabar el código miles de veces en cuestión de segundos sin retirar el chip del circuito.
- **Turno 3 (Orus):** Para el aprendizaje y el prototipado en matriz de contactos utilizamos el encapsulado de doble hilera en línea, o D I P, cuyos pines separados por dos coma cincuenta y cuatro milímetros encajan perfectamente en la protoboard.
- **Turno 4 (Aoede):** Una vez validado el producto para fabricación industrial en serie, el mismo núcleo lógico se migra sin cambiar el código hacia encapsulados compactos de montaje superficial que ahorran valioso espacio en la placa impresa.

Diapositiva 09: Anatomía de la Familia PIC de Gama Media y División Cuádruple del Reloj

Palabra de Instrucción de 14 Bits, Bus de Datos de 8 Bits y Subciclos Internos Q1, Q2, Q3 y Q4

- **Turno 1 (Orus):** Dentro del universo de microcontroladores, la familia PIC de gama media fabricada por Microchip destaca por su robustez eléctrica, facilidad de aprendizaje y arquitectura Harvard con palabras de instrucción de catorce bits.
- **Turno 2 (Aoede):** Un detalle fundamental para calcular tiempos precisos es que el oscilador externo del PIC se divide internamente entre cuatro fases no traslapadas denominadas cu uno, cu dos, cu tres y cu cuatro para formar un ciclo de instrucción.
- **Turno 3 (Orus):** Por consiguiente, si conectamos un cristal de cuarzo muy común de cuatro megahercios, la frecuencia efectiva de ciclo de instrucción es cuatro megahercios dividido entre cuatro, equivalente a un megahercio exacto.
- **Turno 4 (Aoede):** Esto significa que cada ciclo de instrucción dura exactamente un microsegundo, convirtiendo el cálculo de temporizaciones, anchos de pulso y retardos de software en una aritmética limpia y totalmente predecible.

Diapositiva 10: Criterios Prácticos para Seleccionar un Microcontrolador en un Proyecto

Conteo de Pines, Periféricos Integrados, Memoria, Velocidad, Consumo, Herramientas y Disponibilidad

- **Turno 1 (Orus):** Al iniciar un diseño mecatrónico nunca elegimos el microcontrolador al azar ni sobredimensionamos sin motivo: evaluamos primero cuántos pines digitales o analógicos exigen nuestros sensores, indicadores y actuadores más un margen de reserva.
- **Turno 2 (Aoede):** En segundo lugar verificamos los requerimientos de velocidad de reloj, tamaño de memoria Flash para alojar el algoritmo, memoria de datos volátil y necesidad de memoria no volátil para guardar parámetros de calibración ante cortes de energía.
- **Turno 3 (Orus):** En tercer lugar ponderamos el consumo en reposo si el equipo operará con baterías, la corriente directa que pueden entregar sus puertos y la disponibilidad de compiladores maduros como PicBasic Pro y programadores accesibles.
- **Turno 4 (Aoede):** Con estas bases arquitectónicas firmes, en la Parte dos estudiaremos en detalle cada terminal, banco de memoria, registro especial y proceso de grabación física del célebre microcontrolador PIC dieciséis efe ochenta y cuatro.

2. Código Fuente en PicBasic Pro (picbasic_pro_video_01.pbp)

```
' ================================================================================
' CURSO: PROGRAMACION E INTERFAZ DE MICROCONTROLADORES PIC (cursosmecatronica.com)
' Video 01 – Parte 1: Microprocesadores, Microcomputadoras y Microcontroladores
' Enfoque: CPU/ALU/Registros, Buses de Direcciones-Datos-Control, Ciclo Fetch-Decode-Execute, Von Neumann vs
Harvard y CISC vs RISC
' Ecuaciones Clave: N_dir = 2^B_addr  |  T_ciclo = 4 / f_osc  |  CPI_RISC = 1 (2 en saltos)  |  P_din = C_par ·
V_DD^2 · f_osc
' 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, ["V01 Ciclo=", #ContadorCiclos, " Est=", #Estado, 13, 10]
    PAUSE 100
    GOTO LazoPrincipal
    END
```

3. Referencia en Ensamblador MPASM (ensamblador_mpasm_video_01.asm)

```
; ==========================================================================
; CURSO: PROGRAMACION E INTERFAZ DE MICROCONTROLADORES PIC (cursosmecatronica.com)
; Video 01 – Parte 1: Microprocesadores, Microcomputadoras y Microcontroladores – 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_01.c)

```
/* ============================================================================
 * CURSO: PROGRAMACION E INTERFAZ DE MICROCONTROLADORES PIC (cursosmecatronica.com)
 * Video 01 – Parte 1: Microprocesadores, Microcomputadoras y Microcontroladores – 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);
    }
}
```