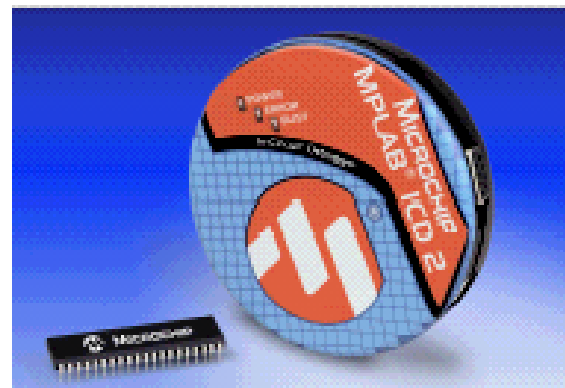
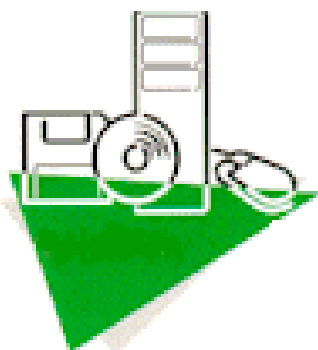


Introducción a los Microcontroladores



Repasando...

- ¿Qué son los Microcontroladores?

Son circuitos integrados que incorporan todos los bloques funcionales de un Sistema Microprocesador en un único encapsulado

- ¿Qué necesitan para funcionar?

Sólo una tensión continua estable (5V, 3.3V, 2.5V, 1.5V...) y un oscilador

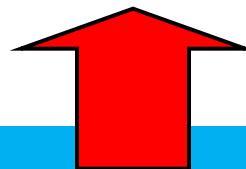
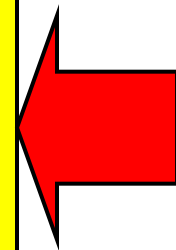
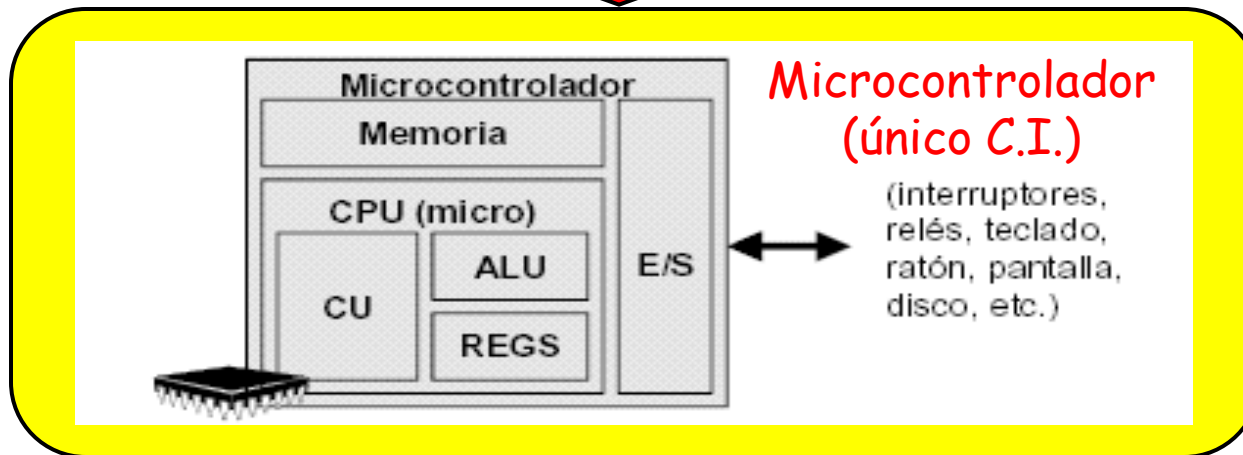
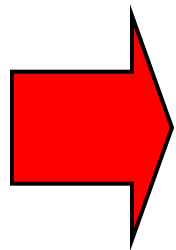
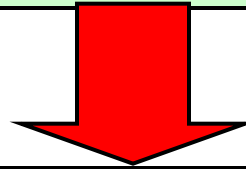
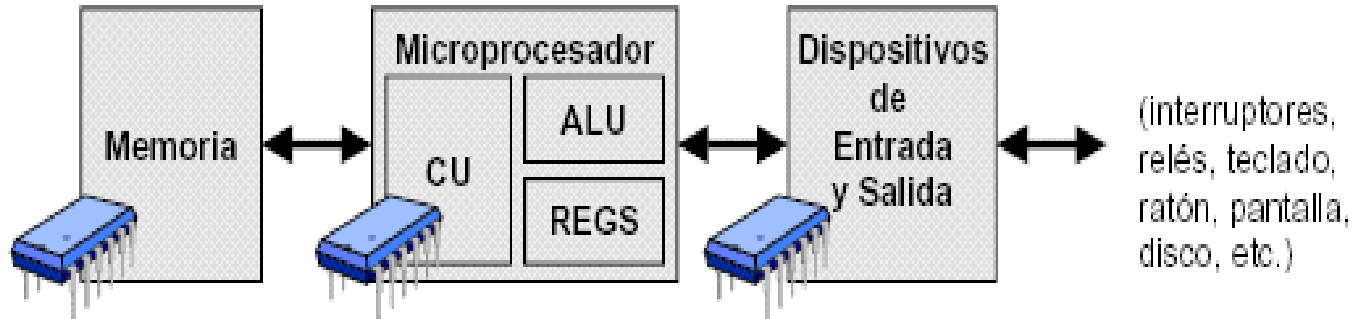
- ¿Qué hacen?

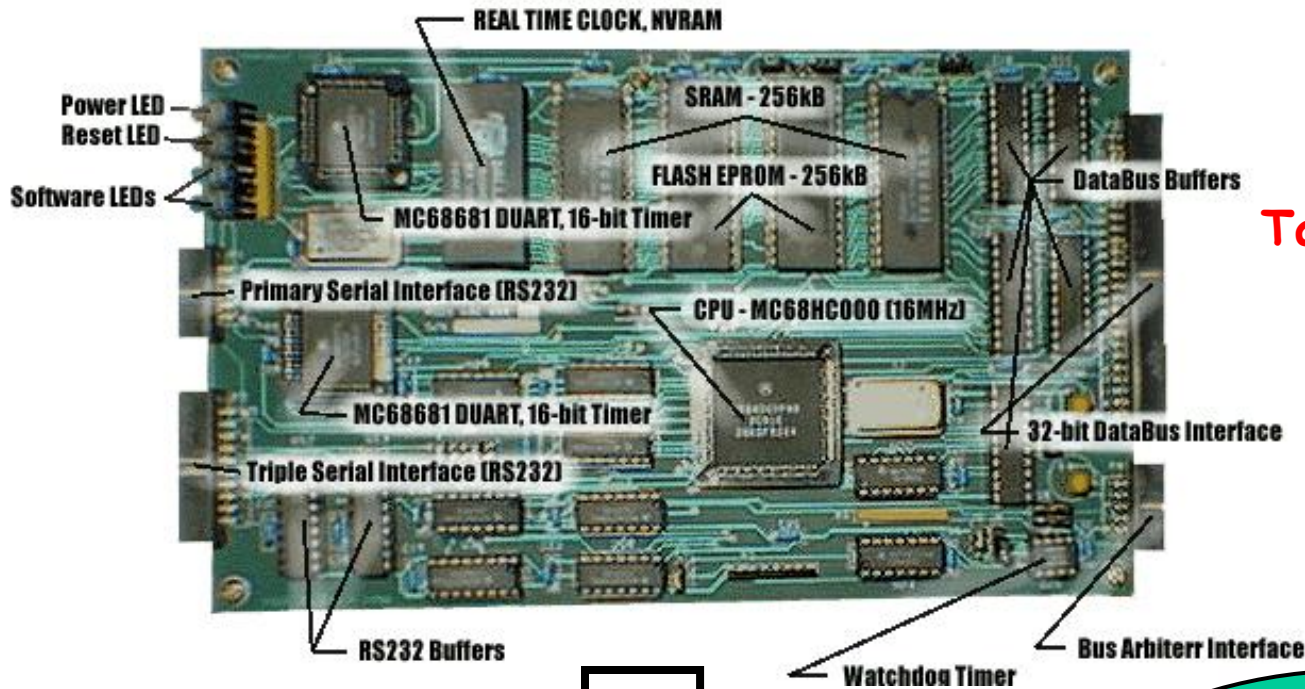
Interpretan (decodifican) combinaciones de bits (instrucciones) y generan señales digitales internas y/o externas

- ¿Para qué?

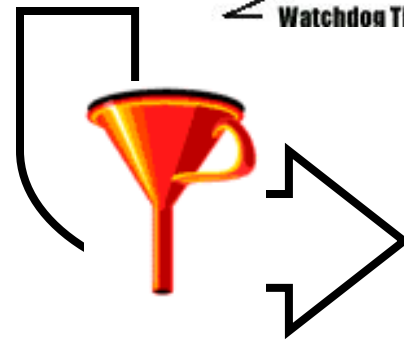
Para "ejecutar" de manera continua una secuencia de instrucciones (programa) que permita controlar un sistema o subsistema electrónico

Sistema Microprocesador (varios C.I. en una PCB)

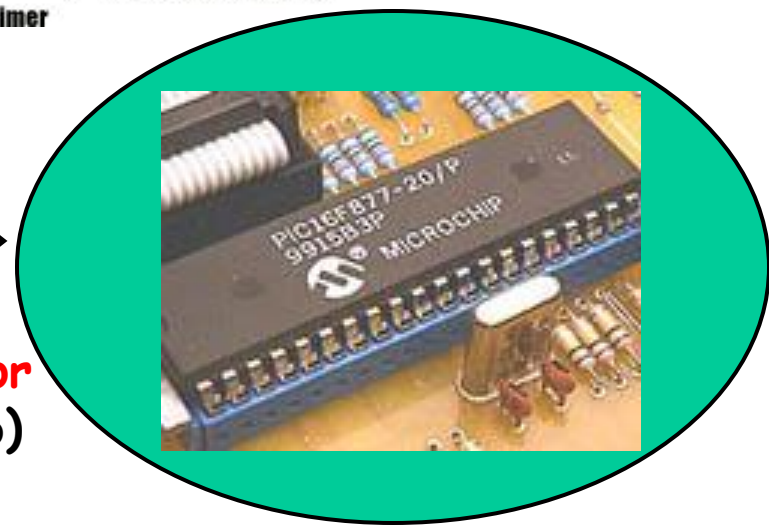




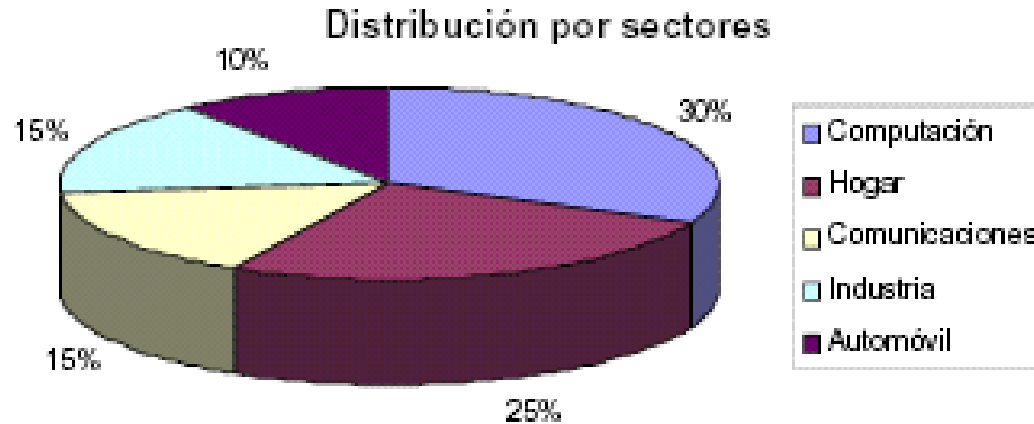
**Tarjeta de Sistema
Microprocesador
para control**



C.I. Microcontrolador
("casi todo" ahí dentro)



¿Dónde tenemos microprocesadores y microcontroladores?



Evolución Microprocesadores: Computadores



75 Millones Microprocesadores/año

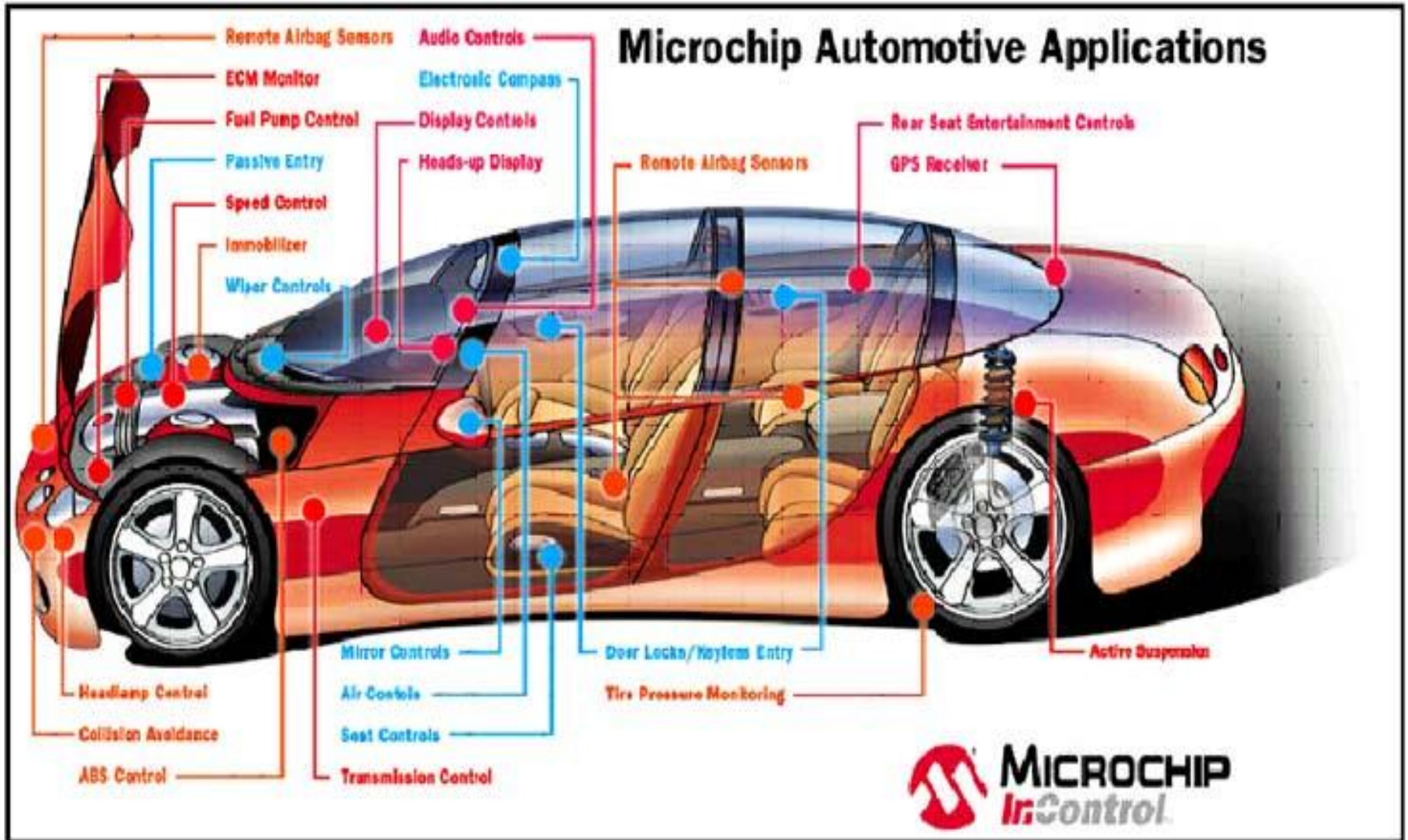
Evolución Microcontroladores: Sistemas Empotrados



"SISTEMAS EMPOTRADOS" (*Embedded systems*):

- Sistemas que incorporan microcontroladores (o microprocesadores) para una tarea específica pero que no son "visibles" ni "programables" directamente por el usuario. "Empotrado" también quiere decir oculto o escondido. Cuando se usa un PC, uno es consciente de que dentro está un microprocesador. ¿Y cuando usamos un teléfono móvil, un reloj, una calculadora, una lavadora, un cargador de baterías, un mando a distancia, un secador de pelo, un lavaplatos, un equipo de música, ...?
- Los microcontroladores de 8 bits dominan en la mayoría de las aplicaciones
- El microcontrolador es el núcleo del sistema electrónico versátil de bajo coste y reducido tamaño que es capaz de detectar las señales de entrada y generar las salidas de un equipo, sistema o instrumento
- Por su reducido tamaño y coste permiten la fácil implantación de sistemas de "inteligencia" distribuida a lo largo de sistemas más complejos
- Los microcontroladores son los semiconductores más abundantes de todos en la actualidad.

¡ Puede haber del orden de 50 microcontroladores en un automóvil !



Los Fabricantes y sus Modelos de Microcontroladores

•INTEL	→	8048,8051,80C196,80386
•MOTOROLA	→	6805,68HC11,68HC12
•HITACHI	→	HD64180
•PHILIPS	→	8051
•SGS-THOMSON	→	ST-62XX
•NATIONAL SEMICONDUCTOR	→	COP400,COP800
•ZILOG	→	Z8,Z86XX
•TEXAS INSTRUMENTS	→	TMS370
•TOSHIBA	→	68HC11
•MICROCHIP	→	PIC

1990

- 1.- Motorola
- 2.- Mitsubishi
- 3.- NEC
- 4.- Intel
- 5.- Hitachi
- 6.- Philips
- 7.- Matsushita
- 8.- National
- 9.- Siemens
- 10.- TI
- 11.- Sharp
- 12.- Oki
- 13.- Toshiba
- 14.- SGS-Thomson
- 15.- Zilog
- 16.- Matra
- 17.- SONY
- 18.- Fujitsu
- 19.- AMD
- 20.- **Microchip**

1996

- 1.- Motorola
- 2.- Mitsubishi
- 3.- SGS-Thomson
- 4.- NEC
- 5.- **Microchip**
- 6.- Philips
- 7.- Zilog

2002

- 1.- **Microchip**
- 2.- Motorola
- 3.- ST-Micro
- 4.- NEC
- 5.- Atmel
- 6.- Sunplus
- 7.- Hitachi
- 8.- Fujitsu
- 9.- Philips
- 10.- Toshiba
- 11.- Mitsubishi
- 12.- Samsung
- 13.- Elan
- 14.- Winbond
- 15.- Zilog
- 16.- Sanyo
- 17.- Matsushita
- 18.- Infineon
- 19.- Holtek
- 20.- National

SEGÚN
UNIDADES
VENDIDAS
DE MICROS
DE 8 BITS

Microcontroladores PIC

- **Arquitectura Harvard**: buses internos separados para memoria de datos (**8 bits**) y de programa (12, 14 ó 16 bits depende de la familia)
- Microprocesador **RISC**: juego de instrucciones reducido
- Estructura **pipe-line**: durante la ejecución de una instrucción, se está accediendo a la memoria de programa para traer la siguiente instrucción a ejecutar. En cuanto se acaba una instrucción, ya se dispone de la siguiente para ejecutar (salvo que se trate de un salto o llamada a subpr.)
- Todas las instrucciones **ocupan una posición de memoria** de programa
- Todas las instrucciones se ejecutan en **un ciclo de instrucción** = 4 ciclos de reloj (salvo las instrucciones de salto)
- **Ortogonalidad** de los registros: se opera entre el registro de trabajo *W* y cualquier otro registro, el resultado puede almacenarse en el citado registro o en *W*

¿Por qué los Microcontroladores PIC de Microchip?

- **Eficiencia** del código: permiten una gran compactación de los programas
- **Rapidez** de ejecución: a frecuencia de 20MHz → 5 millones de instr./seg.
- **Seguridad** en acceso por la separación de memoria de datos y de programa
- **Juego reducido de instrucciones** y de fácil aprendizaje
- **Compatibilidad de pines** y código entre dispositivos de la misma familia o incluso de familias distintas
- Gran **variedad de versiones** en distintos encapsulados (desde 8 hasta 84 pines) sin reducción de las prestaciones internas (muy versátiles)
- Posibilidad de **protección del código** muy fiable
- Herramientas de desarrollo software y hardware **abundantes** y de **bajo coste**

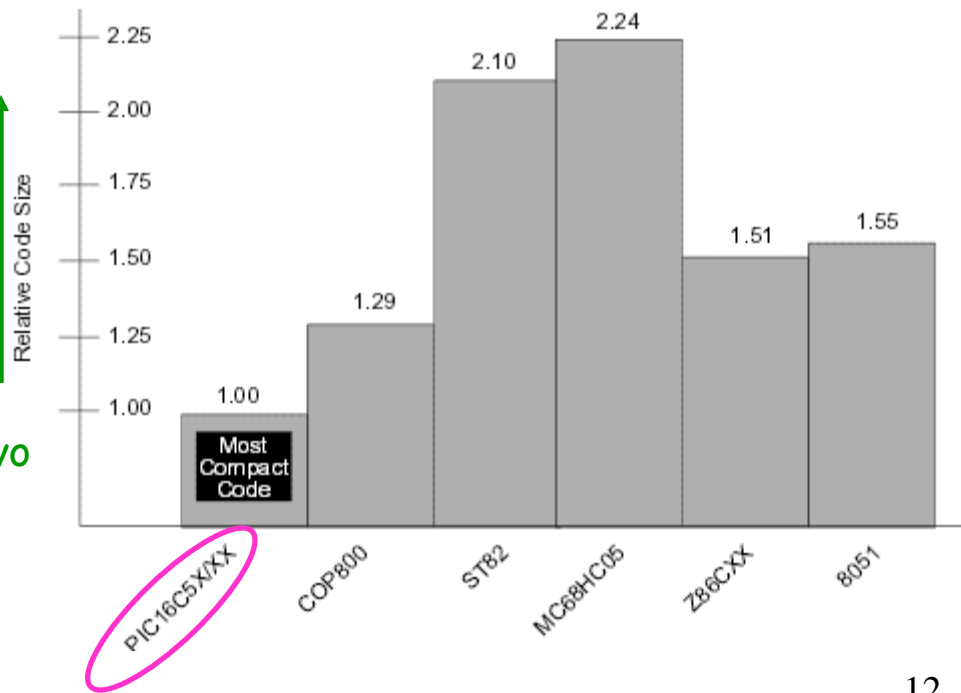
Device	Packing BCD	Loop Control	Bit Test & Branch	8-Bit Sync Transmission	10 ms Software Timer	Overall
COP800	4 2.00	2 1.00	2 1.00	16 1.46	8 1.00	1.29
ST62	10 5.00	2 1.00	3 1.50	19 1.73	10 1.25	2.10
MC68HC05	10 5.00	3 1.50	3 1.50	20 1.82	11 1.38	2.24
Z86CXX	4 2.00	2 1.00	3 1.50	21 1.91	9 1.125	1.51
8051	4 2.00	2 1.00	4 2.00	15 1.36	11 1.375	1.547
PIC16CXX/XX	2	2	2	11	8	1.00

Comparando el tamaño del código con los 16Cxx

Nº posiciones de memoria

$\frac{\text{Nº posiciones de memoria}}{\text{Nº posiciones para PIC16Cxx}}$

Tamaño relativo del código



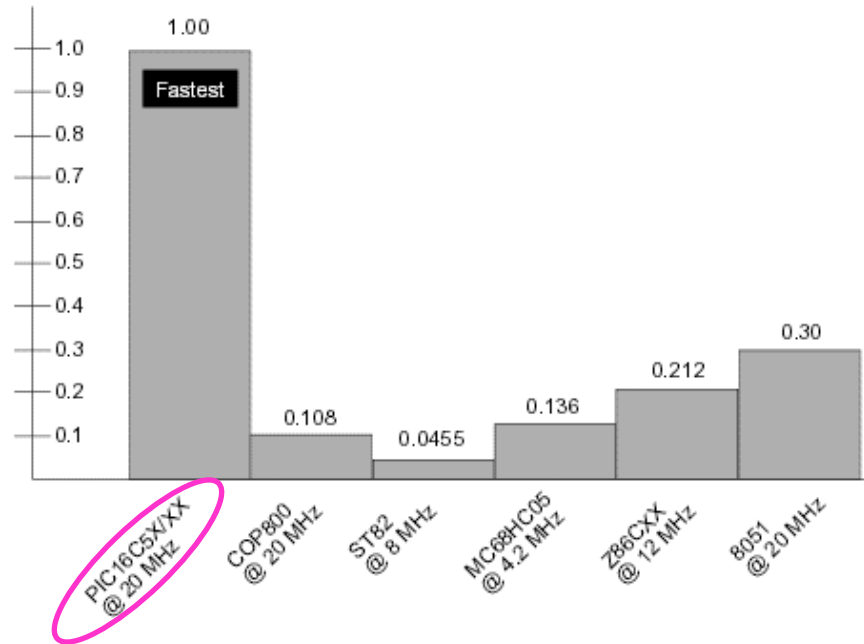
Device	Packing BCD	Loop Control	Bit Test & Branch	8-Bit Sync Transmission	10 ms Software Timer	Overall
COP800 @ 20 MHz	5 μ s 0.08	6 μ s 0.0832	4 μ s 0.1252	105 μ s 0.1408	—	0.108
ST62 @ 8 MHz	45.5 μ s 0.0088	9.75 μ s 0.0615	8.125 μ s 0.0738	390 μ s 0.0329	—	0.0455
MC68HC05 @ 4.2 MHz	10.05 μ s 0.038	2.86 μ s 0.1748	2.38 μ s 0.21	126.7 μ s 0.1168	—	0.136
Z86CXX @ 12 MHz	2.33 μ s 0.172	1.835 μ s 0.272	2.835 μ s 0.176	68.67 μ s 0.224	—	0.212
8051 @ 20 MHz	2.4 μ s 0.1666	1.2 μ s 0.4166	1.8 μ s 0.277	44.4 μ s 0.33	—	0.30
PIC16C5X/XX @ 20 MHz	0.4 μ s	0.6/0.4 μ s	0.6/0.4 μ s	14.8 μ s	—	1.00

Comparando la velocidad de ejecución con los 16Cxx

tiempo que necesita PIC16Cxx
tiempo que necesita el otro "micro"

Velocidad relativa de ejecución

Relative Execution Speed



Familias de Microcontroladores PIC

- **Familia PIC10F20x**

Encapsulados de 6 pines (SOT). Oscilador interno 4MHz
Memoria de programa de 12 bits y datos de 8 bits
Juego de 33 instrucciones

- **Familia PIC12CXXX/12FXXX**

Encapsulados de 8 pines (DIP ó SOIC)
Instrucciones de 12 ó 14 bits en Memoria de Programa
Juego de 33 ó 35 instrucciones
Disponibles con EEPROM de datos
Modelos con módulos de conversión A/D
Permiten alimentación a baja tensión de hasta 2,5V

- **Familia PIC16C5X**

Encapsulados de 14, 18, 20 y 28 pines
Instrucciones de 12 bits
Juego de 33 instrucciones
Es la familia base de partida de los PIC

Familias de Microcontroladores PIC (II)

• Familia PIC16CXXX/16FXXX (+PIC14000)

Encapsulados desde 18 hasta 68 pines (DIP, SSOP, PLCC, QFP)
Instrucciones de 14 bits en Memoria de Programa
Juego de 35 instrucciones
Gran variedad de módulos integrados

• Familia PIC17CXXX

SE ESTÁN DEJANDO DE FABRICAR

Encapsulados de 40 a 80 pines
Memoria de Programa de 16 bits
Juego de 58 instrucciones (oper. de multiplicación por hardware)
Posibilidad de direccionar memoria externa

• Familia PIC18CXXX/18FXXX

Encapsulados de 18 a 80 pines
Memoria de Programa de 16 bits
Juego de 77 instrucciones (multiplicación)

1.- De la familia PIC10F20x

Pin Diagrams

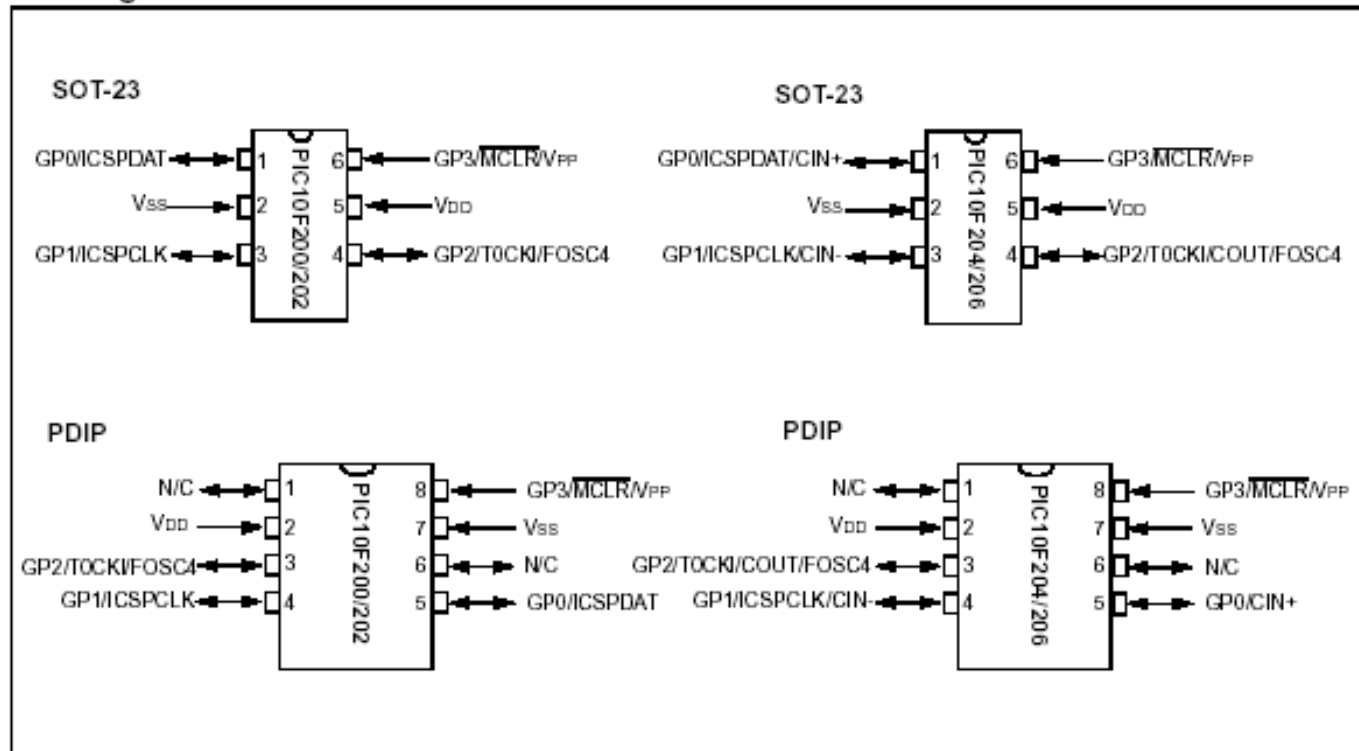


TABLE 1-1: PIC10F2XX MEMORY AND FEATURES

Device	Program Memory	Data Memory	I/O	Timers 8-bit	Comparator
	Flash (words)	SRAM (bytes)			
PIC10F200	256	16	4	1	0
PIC10F202	512	24	4	1	0
PIC10F204	256	16	4	1	1
PIC10F206	512	24	4	1	1

2.- De la familia PIC12CXXX/12FXXX

Devices Included in this Data Sheet:

- PIC12C671
- PIC12C672
- PIC12CE673
- PIC12CE674

Note: Throughout this data sheet PIC12C67X refers to the PIC12C671, PIC12C672, PIC12CE673 and PIC12CE674. PIC12CE67X refers to PIC12CE673 and PIC12CE674.

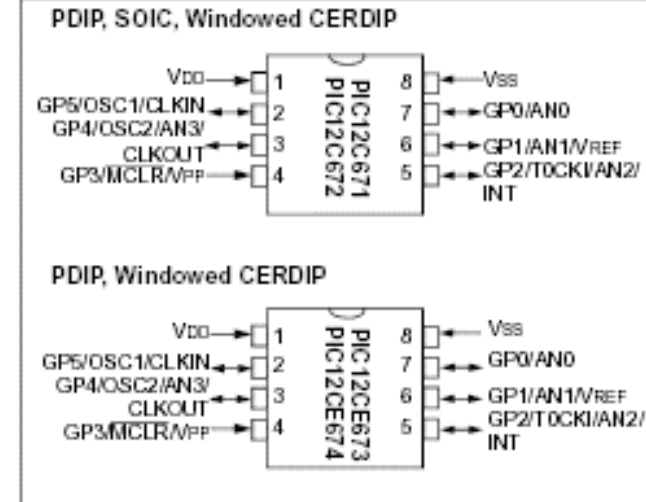
High-Performance RISC CPU:

- Only 35 single word instructions to learn
- All instructions are single cycle (400 ns) except for program branches which are two-cycle
- Operating speed: DC - 10 MHz clock input
DC - 400 ns instruction cycle

Device	Memory		
	Program	Data RAM	Data EEPROM
PIC12C671	1024 x 14	128 x 8	—
PIC12C672	2048 x 14	128 x 8	—
PIC12CE673	1024 x 14	128 x 8	16 x 8
PIC12CE674	2048 x 14	128 x 8	16 x 8

- 14-bit wide instructions
- 8-bit wide data path
- Interrupt capability
- Special function hardware registers
- 8-level deep hardware stack
- Direct, indirect and relative addressing modes for data and instructions

Pin Diagrams:



Special Microcontroller Features:

- In-Circuit Serial Programming (ICSP™)
- Internal 4 MHz oscillator with programmable calibration
- Selectable clockout
- 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
- Interrupt-on-pin change (GP0, GP1, GP3)
- Internal pull-ups on I/O pins (GP0, GP1, GP3)
- Internal pull-up on MCLR pin
- Selectable oscillator options:
 - INTRC: Precision internal 4 MHz oscillator
 - EXTRC: External low cost RC oscillator

3.- De la familia PIC16C5X

Devices Included in this Data Sheet:

- PIC16C54
- PIC16CR54
- PIC16C55
- PIC16C56
- PIC16CR56
- PIC16C57
- PIC16CR57
- PIC16C58
- PIC16CR58

Note: 16C5X refers to all revisions of the part (i.e., 16C54 refers to 16C54, 16C54A, and 16C54C), unless specifically called out otherwise.

High-Performance RISC CPU:

- Only 33 single word instructions to learn
- All instructions are single cycle (200 ns) except for program branches which are two-cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle

Device	Pins	I/O	EPROM/ ROM	RAM
PIC16C54	18	12	512	25
PIC16C54A	18	12	512	25
PIC16C54C	18	12	512	25
PIC16CR54A	18	12	512	25
PIC16CR54C	18	12	512	25
PIC16C55	28	20	512	24
PIC16C55A	28	20	512	24
PIC16C56	18	12	1K	25
PIC16C56A	18	12	1K	25
PIC16CR56A	18	12	1K	25
PIC16C57	28	20	2K	72
PIC16C57C	28	20	2K	72
PIC16CR57C	28	20	2K	72
PIC16C58B	18	12	2K	73
PIC16CR58B	18	12	2K	73

- 12-bit wide instructions
- 8-bit wide data path

Peripheral Features:

- 8-bit real time clock/counter (TMR0) with 8-bit programmable prescaler
- Power-on Reset (POR)
- Device Reset Timer (DRT)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable Code Protection
- Power saving SLEEP mode
- Selectable oscillator options:
 - RC: Low-cost RC oscillator
 - XT: Standard crystal/resonator
 - HS: High-speed crystal/resonator
 - LP: Power saving, low-frequency crystal

CMOS Technology:

- Low-power, high-speed CMOS EPROM/ROM technology
- Fully static design
- Wide-operating voltage and temperature range:
 - EPROM Commercial/Industrial 2.0V to 6.25V
 - ROM Commercial/Industrial 2.0V to 6.25V
 - EPROM Extended 2.5V to 6.0V
 - ROM Extended 2.5V to 6.0V
- Low-power consumption
 - < 2 mA typical @ 5V, 4 MHz
 - 15 μ A typical @ 3V, 32 kHz
 - < 0.6 μ A typical standby current (with WDT disabled) @ 3V, 0°C to 70°C

Note: In this document, figure and table titles refer to all varieties of the part number indicated, (i.e., The title "Figure 14-1: Load Conditions - PIC16C54A", also refers to PIC16LC54A and PIC16LV54A parts) unless specifically called out otherwise.

4.- De la familia PIC16XXX

Devices included in this data sheet:

- PIC16C63A • PIC16C73B
- PIC16C65B • PIC16C74B

PIC16CXX Microcontroller Core Features:

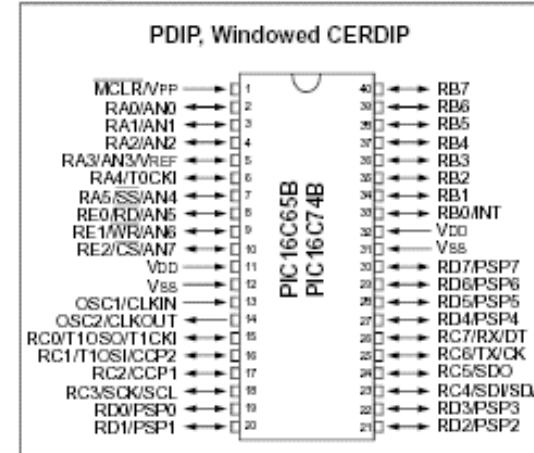
- High-performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- 4K x 14 words of Program Memory,
192 x 8 bytes of Data Memory (RAM)
- Interrupt capability
- 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 EPROM technology
- Wide operating voltage range: 2.5V to 5.5V
- High Sink/Source Current 25/25 mA
- Commercial, Industrial and Automotive temperature ranges
- Low-power consumption:
 - < 5 mA @ 5V, 4 MHz
 - 23 μ A typical @ 3V, 32 kHz
 - < 3 μ A typical standby current

Devices	I/O Pins	A/D Chan.	PSP	Interrupts
PIC16C63A	22	-	No	10
PIC16C65B	33	-	Yes	11
PIC16C73B	22	5	No	11
PIC16C74B	33	8	Yes	12

PIC16C7X 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
- Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 200 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 8-bit multichannel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ and I²C™
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI)
- Parallel Slave Port (PSP), 8-bits wide with external RD, WR and CS controls
- Brown-out detection circuitry for Brown-out Reset (BOR)

Pin Diagram:



5.- De la familia PIC17XXX

Devices included in this data sheet:

- PIC17CR42
- PIC17C42A
- PIC17C43
- PIC17CR43
- PIC17C44
- PIC17C42†

Microcontroller Core Features:

- Only 58 single word instructions to learn
- All single cycle instructions (121 ns) except for program branches and table reads/writes which are two-cycle
- Operating speed:
 - DC - 33 MHz clock input
 - DC - 121 ns instruction cycle

Device	Program Memory		Data Memory
	EPROM	ROM	
PIC17CR42	-	2K	
PIC17C42A	2K		
PIC17C43	4K		454
PIC17CR43			454
PIC17C44		-	454
PIC17C42†	2K	-	232

EN VÍAS DE EXTINCIÓN POR SIMILITUD CON LA FAMILIA PIC18

Multiplier

(available on the PIC17C42)

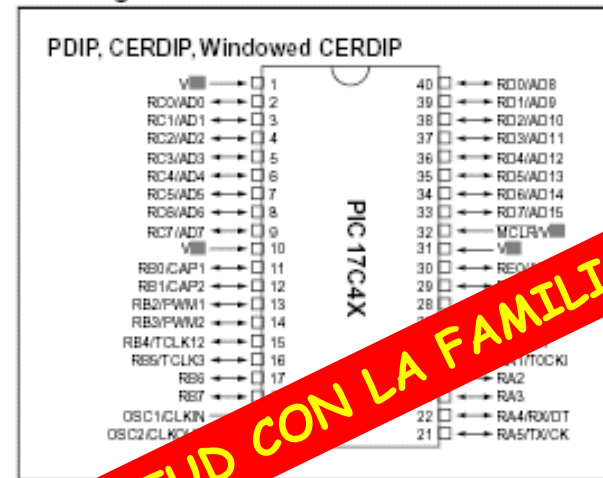
Interrupt capability

- 16 levels deep hardware stack
- Direct, indirect and relative addressing modes
- Internal/External program memory execution
- 64K x 16 addressable program memory space

Peripheral Features:

- 33 I/O pins with individual direction control
- High current sink/source for direct LED drive
 - RA2 and RA3 are open drain, high voltage (12V), high current (60 mA), I/O

Pin Diagram



• Timer/counter

16-bit timer/counter

Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI)

Special Microcontroller Features:

- 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
- Code-protection
- Power saving SLEEP mode
- Selectable oscillator options

CMOS Technology:

- Low-power, high-speed CMOS EPROM/ROM technology
- Fully static design
- Wide operating voltage range (2.5V to 6.0V)
- Commercial and Industrial Temperature Range
- Low-power consumption
 - < 5 mA @ 5V, 4 MHz
 - 100 µA typical @ 4.5V, 32 kHz
 - < 1 µA typical standby current @ 5V

6.- De la familia PIC18XXX

High Performance RISC CPU:

- C-compiler optimized architecture/instruction set
 - Source code compatible with the PIC16CXX instruction set
- * • Linear program memory addressing to 2M bytes
- * • Linear data memory addressing to 4K bytes

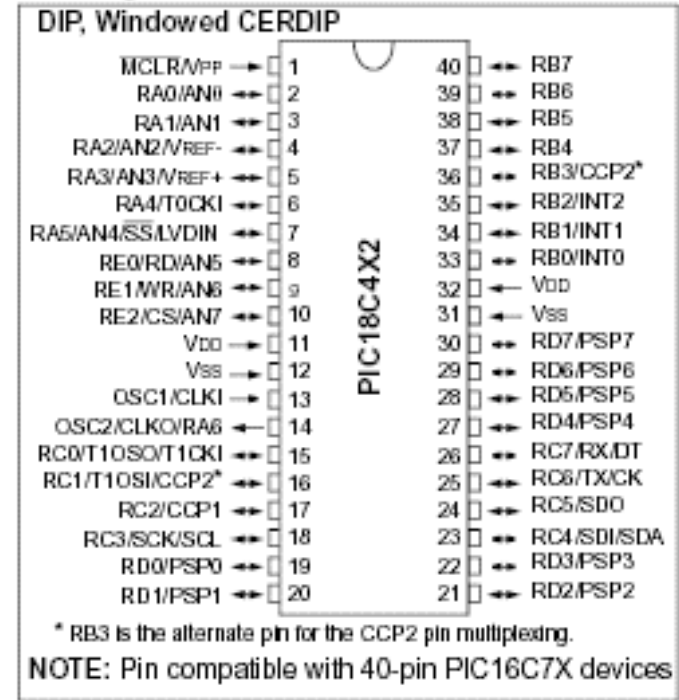
Device	On-Chip Program Memory		On-Chip RAM (bytes)
	EPROM (bytes)	# Single Word Instructions	
PIC18C242	16K	8192	512
PIC18C252	32K	16384	1536
PIC18C442	16K	8192	512
PIC18C452	32K	16384	1536

- * • Up to 10 MIPS operation:
 - DC - 40 MHz osc./clock input
 - 4 MHz - 10 MHz osc./clock input with PLL active
- 16-bit wide instructions, 8-bit wide data path
- Priority levels for interrupts
- * • 8 x 8 Single Cycle Hardware Multiplier

Peripheral Features:

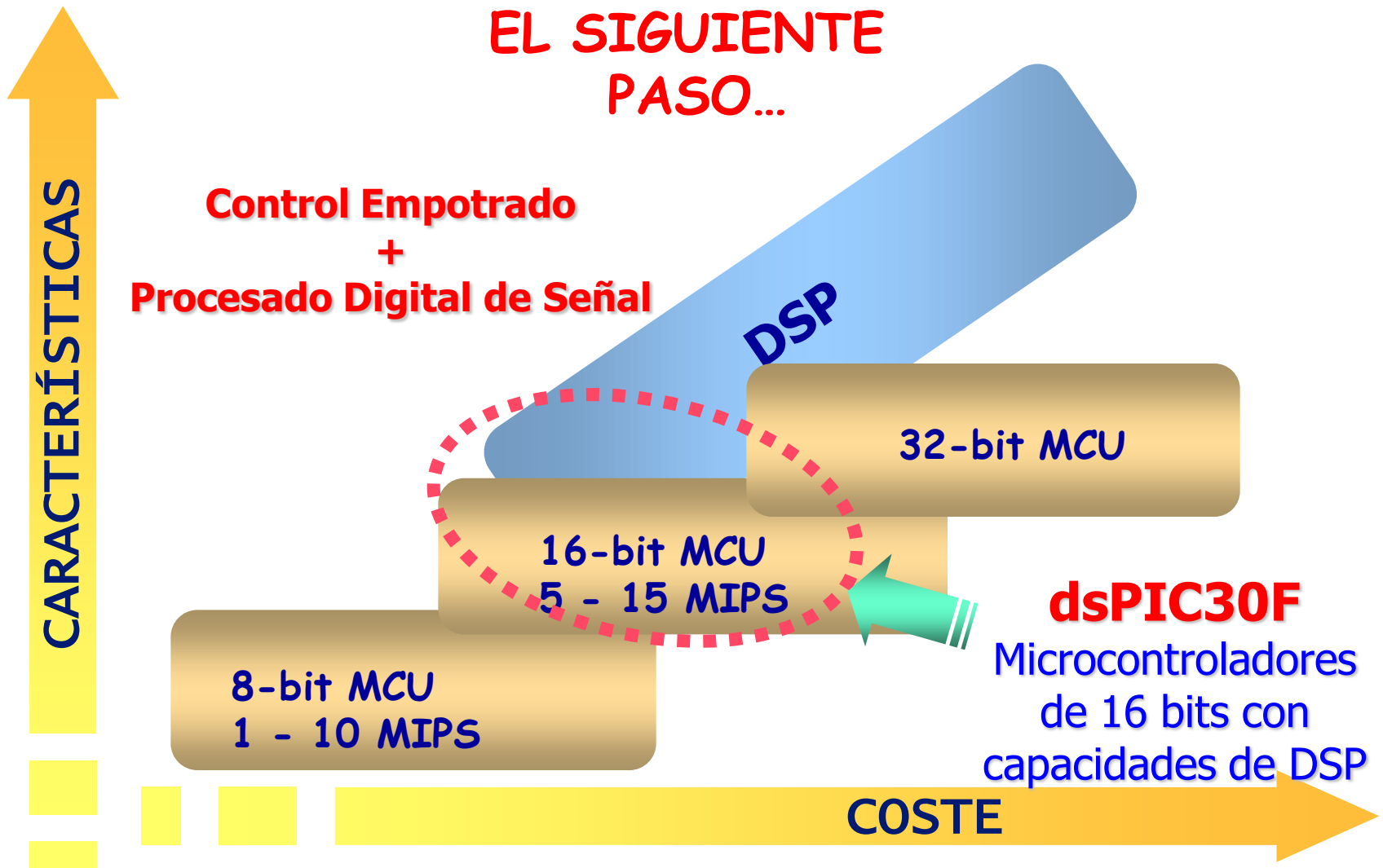
- High current sink/source 25 mA/25 mA
- Three external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter with 8-bit period register (time-base for PWM)
- * • Timer3 module: 16-bit timer/counter

Pin Diagrams



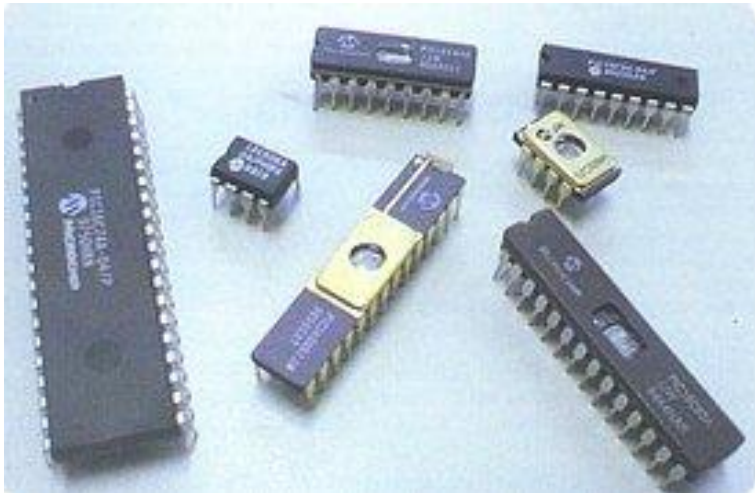
Analog Features:

- 10-bit Analog-to-Digital Converter module (A/D) with:
 - Fast sampling rate
 - Conversion available during sleep
 - DNL = ± 1 LSB, INL = ± 1 LSB
- Programmable Low-Voltage Detection (LVD) module
 - Supports interrupt on low voltage detection
- Programmable Brown-out Reset (BOR)



El catálogo actual de Familias de Microcontroladores PIC (2004-05)

- Familia PIC10F20x 4 Dispositivos
 - Familia PIC12CXXX/12FXXX (12/14 bits) 8 Dispositivos
 - Familia PIC16C5X (12 bits) 9 Dispositivos
 - • Familia PIC16CXXX/16FXXX (14 bits) 74 Dispositivos
 - Familia PIC17CXXX (16 bits) 7 Dispositivos
 - Familia PIC18CXXX/18FXXX (16 bits) 82 Dispositivos
-



TOTAL:

¡ 177 Dispositivos !

Versiones de Memoria de Programa:
OTP, EPROM, EEPROM y FLASH

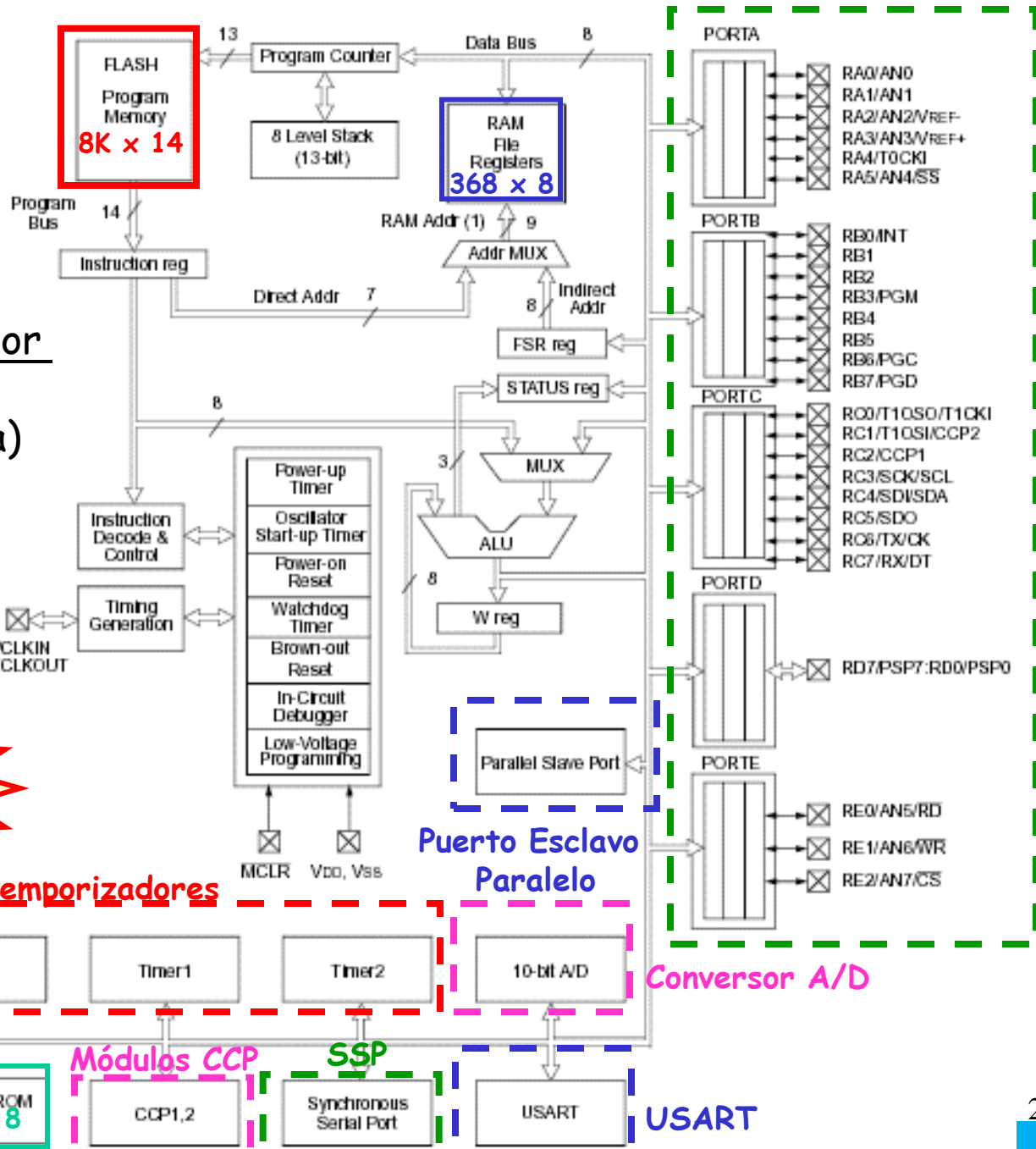
Módulos Internos Disponibles en la Familia Media (PIC16CXXX/16FXXX)

- Puertos de Entrada/Salida
- Puerto Esclavo Paralelo (PSP)
- Temporizadores/contadores (TMRO, TMR1, TMR2)
- Captura / Comparación / PWM (CCP1 y CCP2)
- Conversión Analógica / Digital (A/D)
- Transmisor Receptor Asíncrono Síncrono Universal (USART ó SCI)
- Puerto Serie Síncrono Básico ó Maestro (BSSP ó MSSP)
- Memoria EEPROM de datos
- FLASH EEPROM de programa modificable desde el código

Microcontrolador PIC16F877 (familia media)

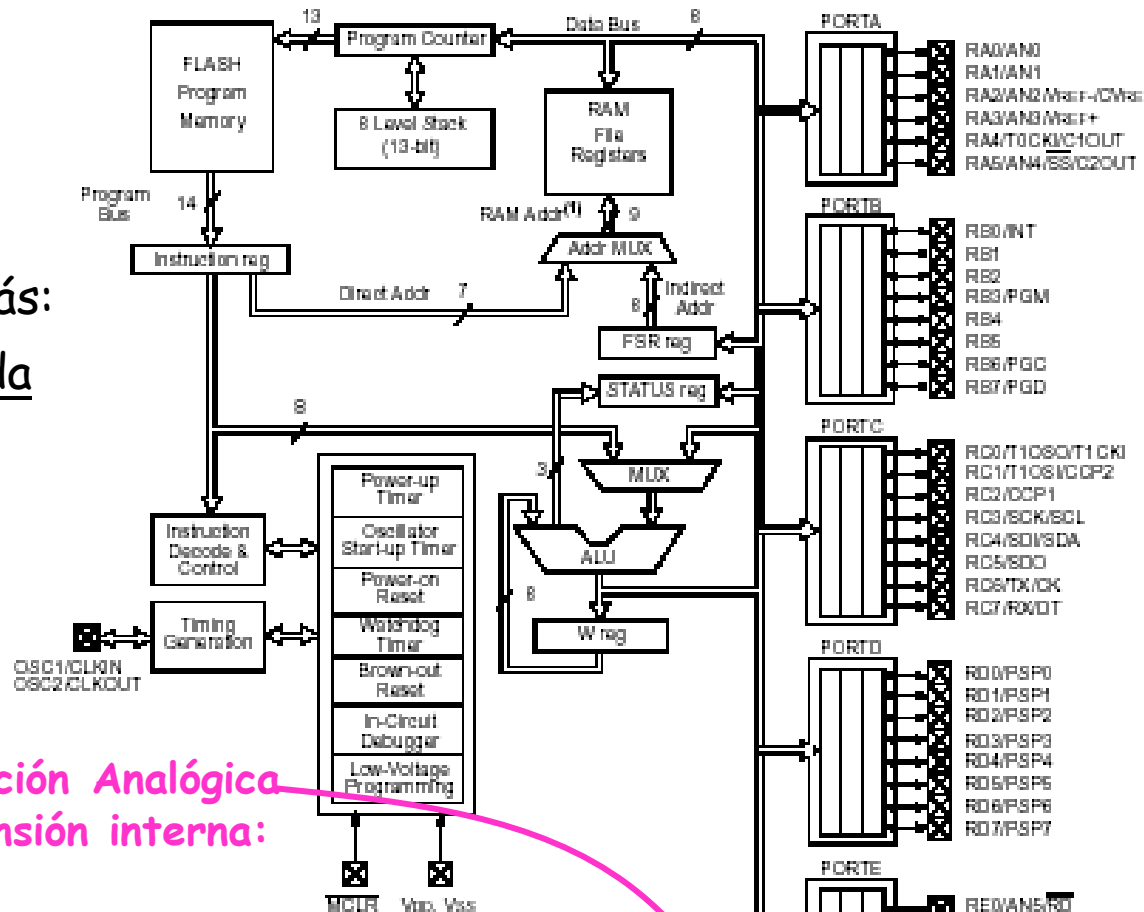


¡ Tiene de todo !



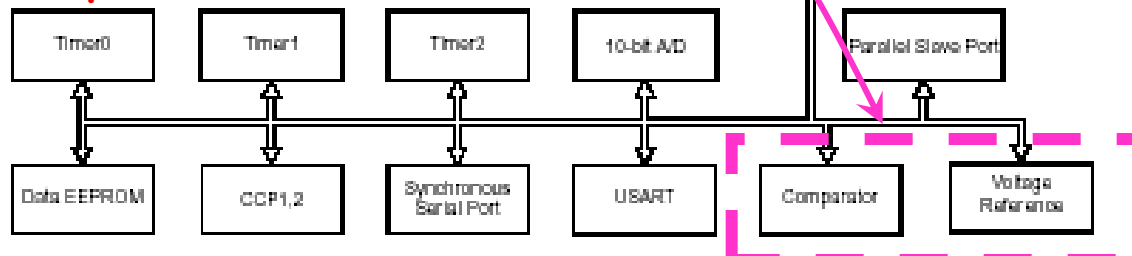
Puertos de E/S

Y todavía hay más:
Versión avanzada
PIC16F877A



Módulo de Comparación Analógica
 + Referencia de tensión interna:

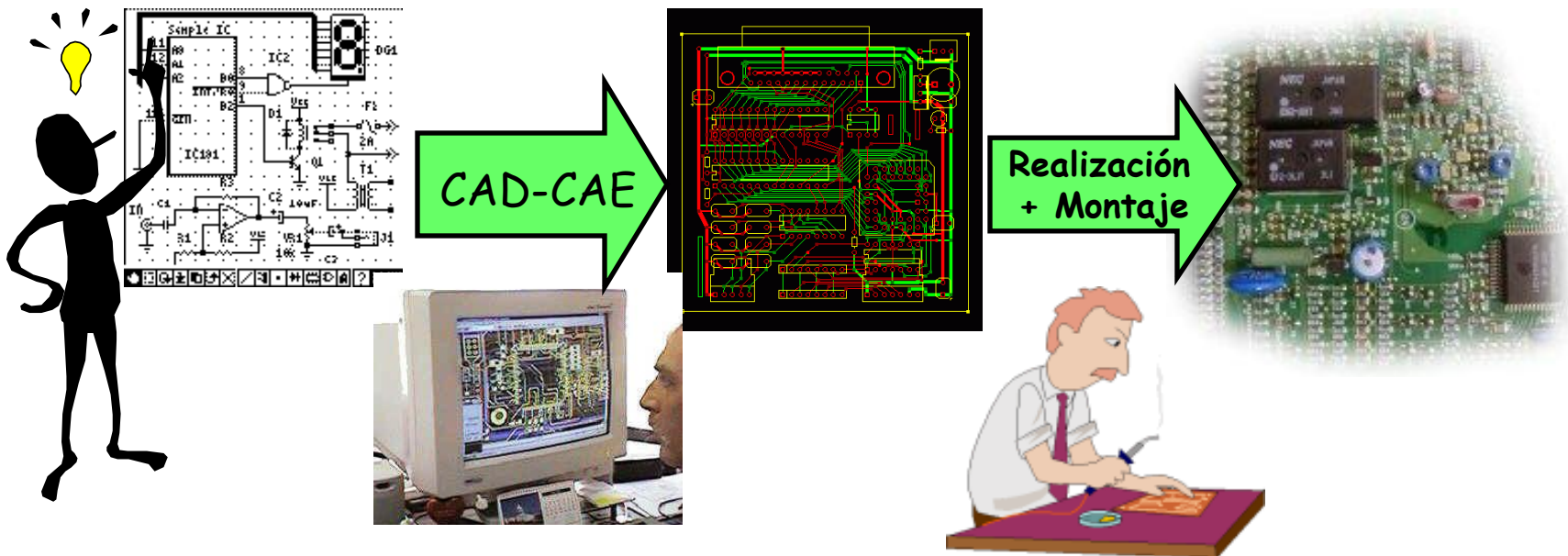
- Dos comparadores
- Referencia de tensión interna programable
- Salidas de comparadores accesibles



El Procedimiento de Diseño con Microcontroladores

- **Diseño hardware:** conectaremos al micro todo lo que necesita para funcionar y lo que le permite interactuar con el mundo exterior

- ⊖ Definición de las señales de Entrada, Salida y Alimentaciones
- ⊖ Etapas de Adaptación/Acondicionamiento Eléctrico de Señales
- ⊖ Selección de los módulos del microcontrolador a utilizar y pines
- ⊖ Diseño de la Placa de Circuito Impreso (PCB)
- ⊖ Montaje y ensamblado



El Procedimiento de Diseño con Microcontroladores (II)

• **Diseño software**: el micro sin un programa cargado en su memoria no hace absolutamente nada, cuando el micro esté grabado con un programa eficiente el conjunto empezará a funcionar

- ⊖ Diseño del programa y escritura del código fuente en ensamblador o en C
- ⊖ Pruebas, verificación y modificación del código:
 - * Simulación del Programa (puro software)
 - * Emulación dentro del Circuito de Aplicación (software y hardware)
- ⊖ Grabación del código máquina en la memoria interna del microcontrolador



Las herramientas para los microcontroladores PIC

- Entorno de Desarrollo: **MPLAB IDE** (*Integrated Development Environment*)
- Herramientas para generar código máquina:

De Microchip:

- **MPASM**: Ensamblador para generar código máquina absoluto o reubicable
- **MPLINK**: Montador de Enlaces para ensamblador y MPLAB-C17
- **MPLIB**: Biblioteca de códigos pre-compilados para utilizar con MPLINK
- **MPLAB-C17**: Compilador de C para la familia PIC17Cxx
- **MPLAB-C18**: Compilador de C para la familia PIC18Cxx

De Otras compañías:

- **CCS PIC C**: Compilador de C
- **HI-TECH PIC C**: Compilador de C
- **IAR PIC C**: Compilador de C
- **PBASIC**: Intérprete de Basic de Parallax

Las herramientas para los microcontroladores PIC (II)

• Simuladores: **MPLAB-SIM** (incluido en MPLAB-IDE)

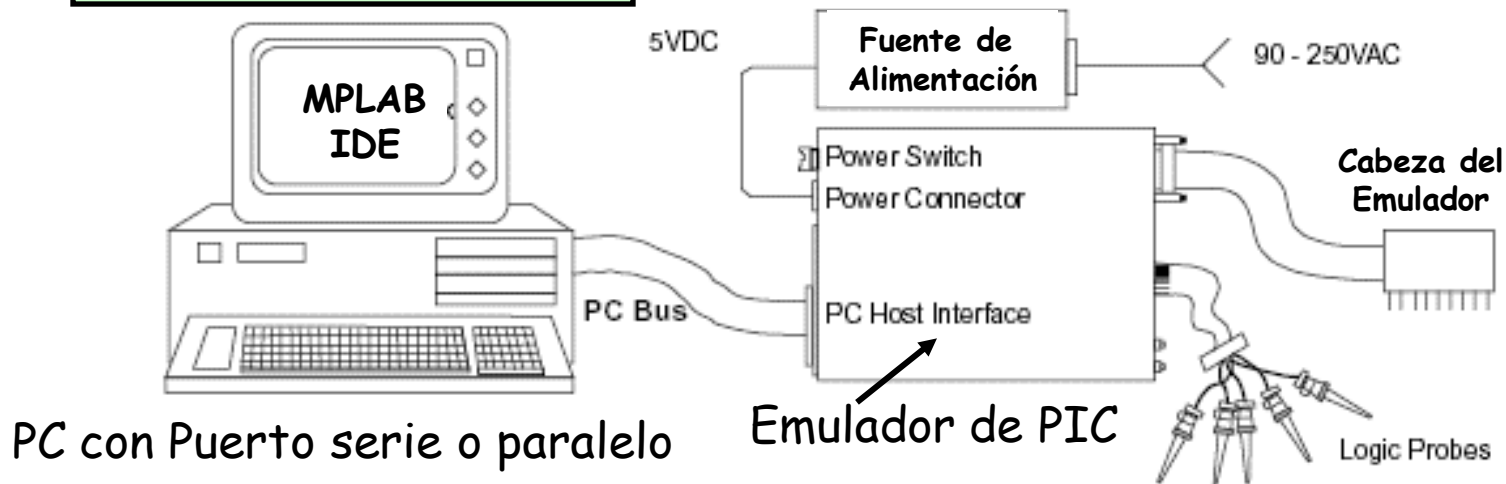
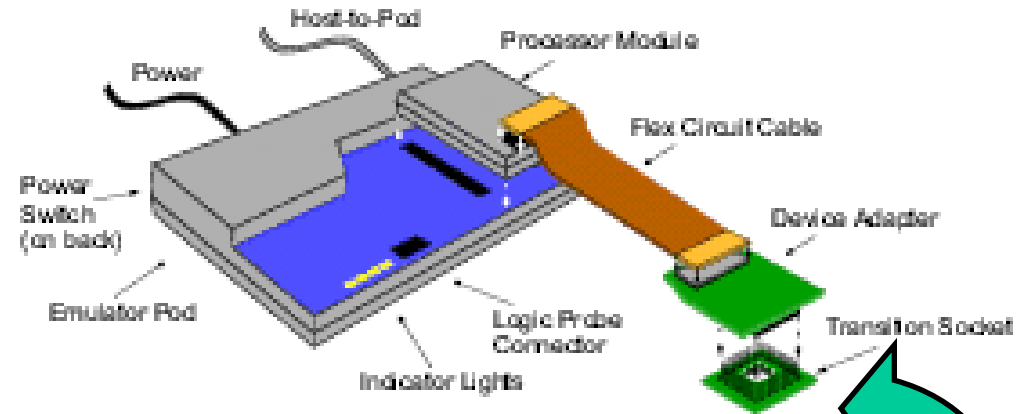
• Emuladores en el Circuito:

Microchip

- MPLAB-ICE 2000
- MPLAB-ICE 4000
- PICMASTER (en desuso)

Otros

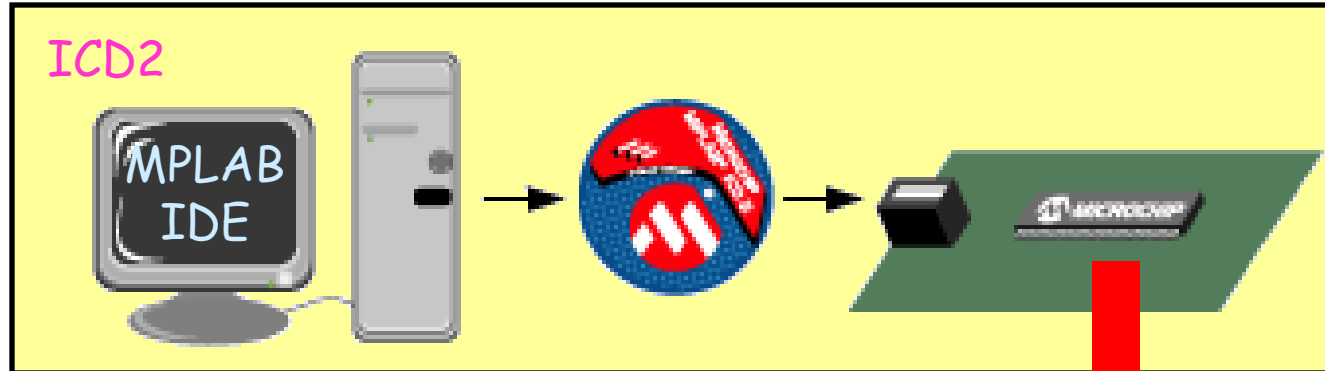
- ICEPIC (RF Solutions)
-



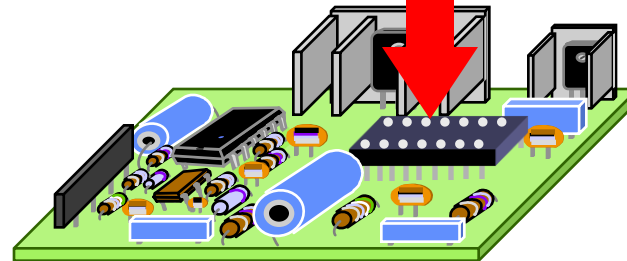
Las herramientas para los microcontroladores PIC (III)

• Depuradores en el Circuito (*In-Circuit Debuggers*):

- MPLAB-ICD
- **MPLAB-ICD2** (también puede funcionar como programador/grabador)



Tarjeta de Aplicación



Las herramientas para los microcontroladores PIC (IV)

• Programadores/Grabadores de la memoria de los PIC

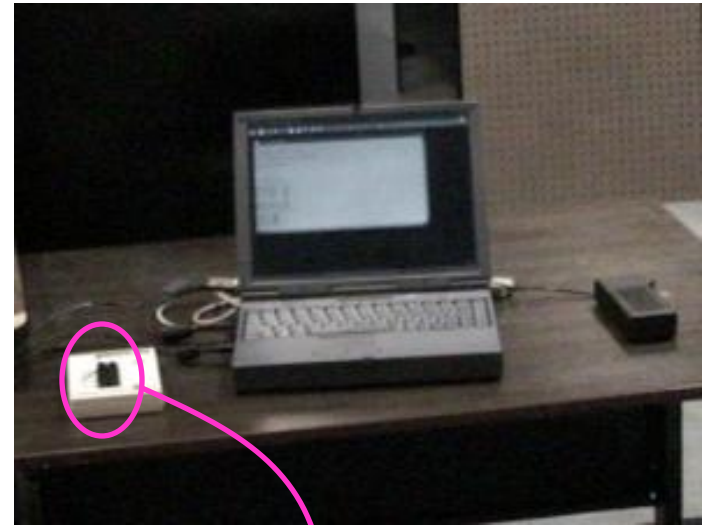
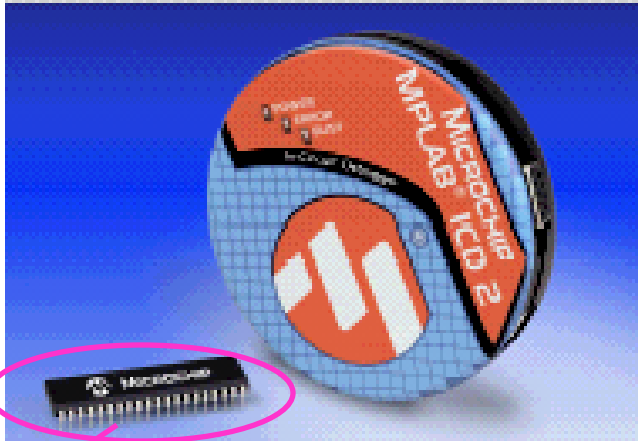
* PICSTART PLUS



* PROMATE II

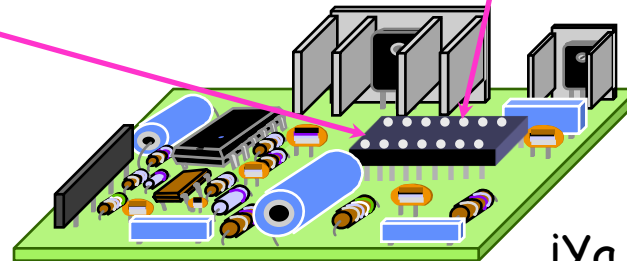


* MPLAB-ICD2 también puede funcionar como programador



Paso final: Grabar la memoria con el programa
e insertar el microcontrolador

Grabaremos con el ICD2
(PICSTART PLUS
y más programadores de
bajo coste)



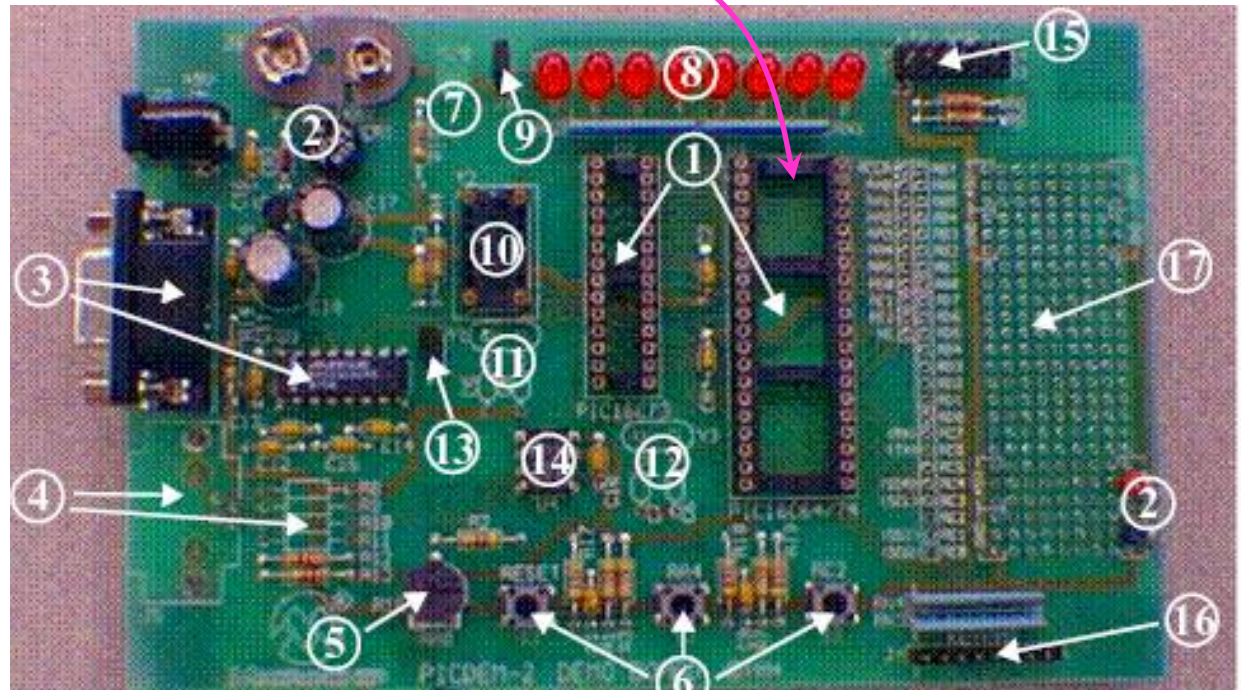
¡Ya es autónomo!

Hardware de Prácticas

• Tarjetas de Evaluación y Demostración

- PICDEM 1
- **PICDEM 2**
- PICDEM 3
- PICDEM 4
- PICDEM 14A
- PICDEM 17
- PICDEM 18R
-

Para entrenamiento
sin necesidad de desarrollar
hardware



Hardware de Prácticas

PICDEM 2 PLUS junto con MPLAB ICD2



El Programador y los microcontroladores

- El **programador para software de PC**, parte de un "lujoso" hardware ya cableado, instalado y en funcionamiento: microprocesador, memoria y dispositivos de E/S como el teclado, el ratón y el monitor. El desarrollo de la aplicación obliga "únicamente" a **tener que escribir el programa** y depurar los errores
- El **programador para microcontroladores** tiene que "**diseñar y crear**" todo el **interface de E/S** con el mundo exterior antes de iniciar la escritura del programa. La mayoría de los programas están hechos a medida de una determinada aplicación y no son demasiado portables. Pero **como todavía no se hacen PCs del tamaño y coste de los microcontroladores** (incluyendo el teclado y el monitor), será necesario seguir utilizando los microcontroladores en los sistemas empotrados
- Las herramientas de desarrollo para diseños basados en micros serán básicamente las mismas que para PC si sustituimos las **librerías gráficas** por otras destinadas al **manejo de los módulos del microcontrolador**.



¿Programar en lenguaje C o en ensamblador?

- La ventaja del C frente al ensamblador es la **rapidez en el desarrollo de las aplicaciones y la comodidad** a la hora de utilizar las funciones de manejo de los módulos internos (la diferencia entre **2 ó 3 días** y **1 ó 2 semanas** con aplicaciones para el manejo de un LCD por ejemplo).
- Las ventajas del **ensamblador** sobre el C residen en **la eficiencia y lo compacto que resulta el código** (entorno a un 80% menor en tamaño). En el ensamblador de los microcontroladores PIC, una instrucción ocupa una única posición de la memoria de programa. Una simple instrucción en C que nos ocupa una única línea de nuestro código fuente puede traducirse en varias posiciones de memoria de programa (como ejemplo se pueden probar unas cuantas condiciones en un "if...").
- Además, cuando uno utiliza la programación en ensamblador, se tiene un **control total sobre el tiempo de ejecución de las instrucciones**, lo que puede resultar especialmente importante en ciertas aplicaciones en tiempo real.

Punto de partida: el Ensamblador