



**UNCUYO**  
UNIVERSIDAD  
NACIONAL DE CUYO



**FACULTAD  
DE INGENIERÍA**

**Licenciatura en Ciencias de la  
Computación**

# Sistemas Embebidos

## Unidad 1

### **Introducción a los sistemas embebidos**

Definición de sistema embebido, muchas...

*“Dispositivo que contiene componentes de hardware y software fuertemente acoplados para realizar una única función: formar parte de grandes sistemas. No está destinado a ser programado de forma independiente por los usuarios. Se espera que trabaje con mínima o sin interacción humana. Dos características adicionales: operación reactiva y fuertemente restringido”. (John Davies, "MSP430 Microcontroller Basics")*

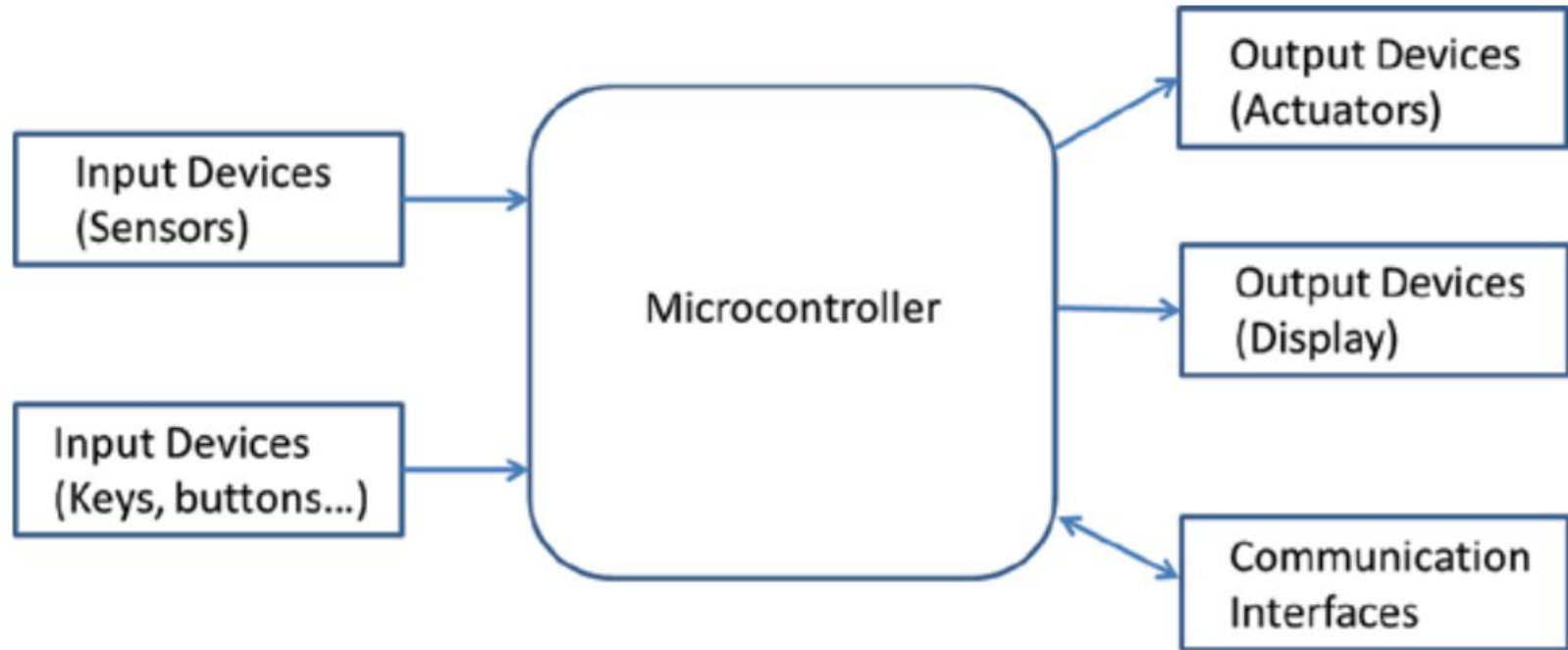
*“Sistemas de procesamiento embebidos en productos del entorno de las personas o ambiente (autos, trenes, aviones, equipos de comunicaciones o industriales, juguetes, electrodomésticos, etc.)” (Marwedel)*

*“Computadora de pequeño tamaño que es parte de una máquina o un gran sistema eléctrico o mecánico. Es llamado embebido porque está embebido o incrustado dentro de otro aparato” (Perry Xiao).*

## **Conceptos relacionados - características de los sistemas embebidos**

- “Cyber-Physical Systems (CPS): integración de procesos de computación y físicos”. (Lee)
- Computación ubicua: Acceso a recursos de computación y/o información en cualquier lugar y tiempo.
- Pervasive computing o computación penetrante: Integración de la tecnología informática con el usuario en cualquier lugar y momento, pudiendo el usuario ser o no consciente de este hecho.
- Computación invisible: Computación integrada en dispositivos de la vida diaria sin que los usuarios sean conscientes de ello.

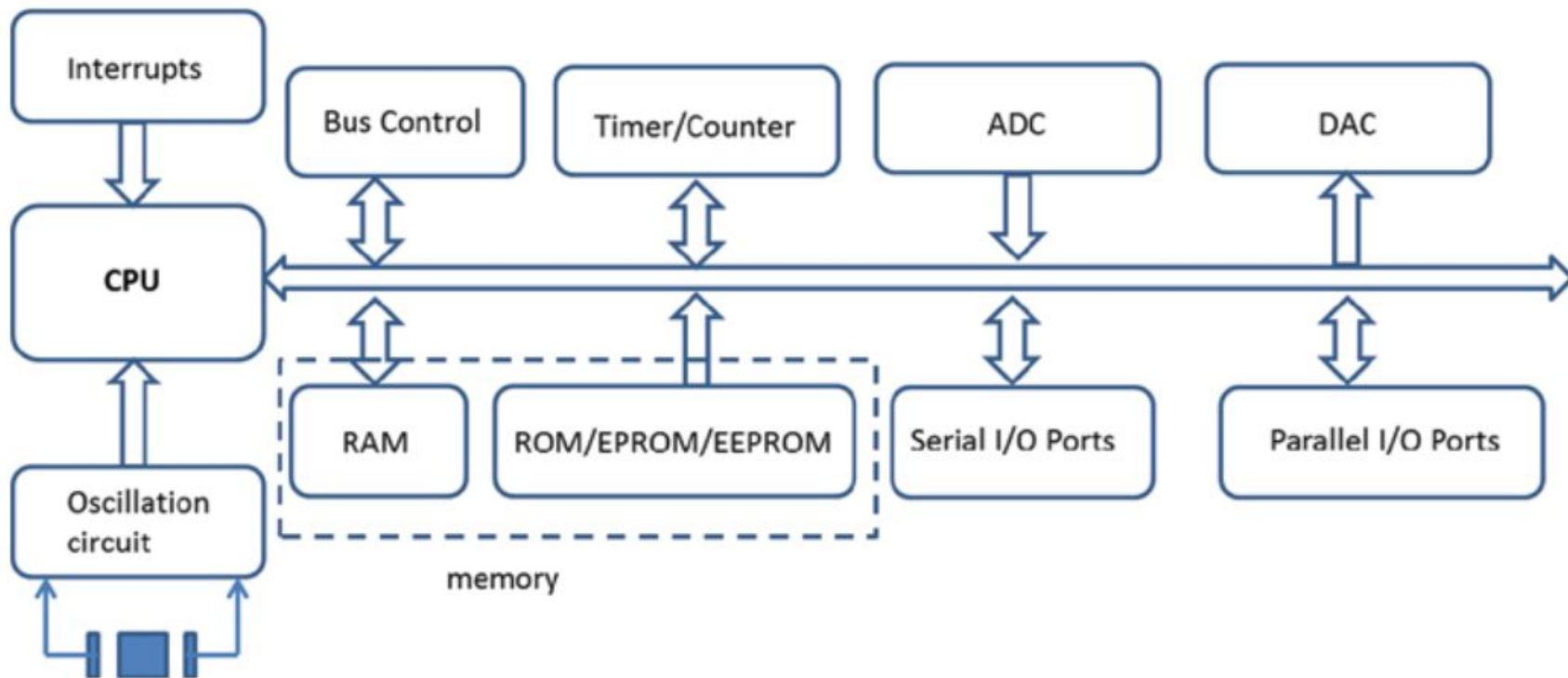
## Diagrama en bloques general de un sistema embebido



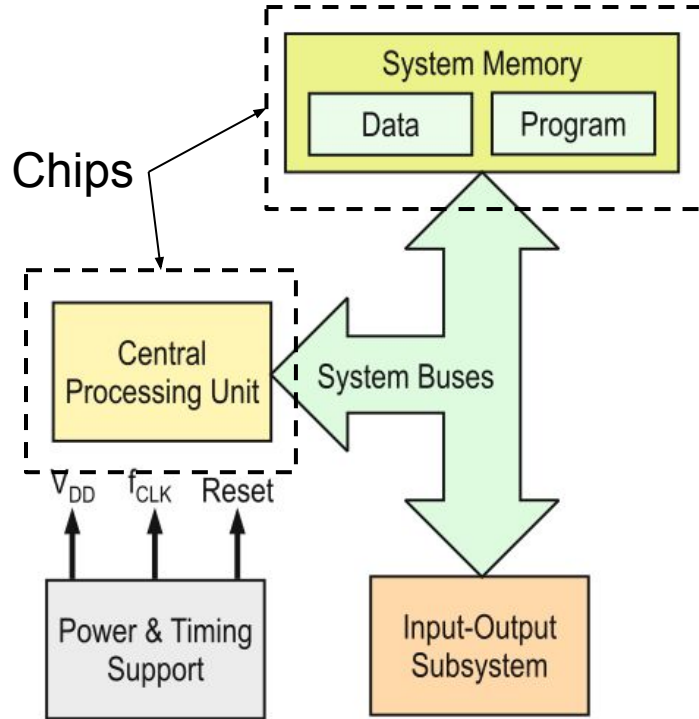
Algunos componentes típicos de los microcontroladores:

- **Timers** (temporizadores)
  - Medición de intervalos de tiempo entre eventos.
  - Temporización de eventos.
  - Modulación de anchos de pulso.
  - Generar pulsos para controlar frecuencia de eventos (despertar periféricos).
  - Watchdog timer.
- **Interfaces de comunicaciones específicas** (bajo costo).
- **Memoria no volátil** (EEPROM o Flash).
- **Conversores A/D y D/A.**
- Módulos **monitores**, **debuggers** y **emuladores**.

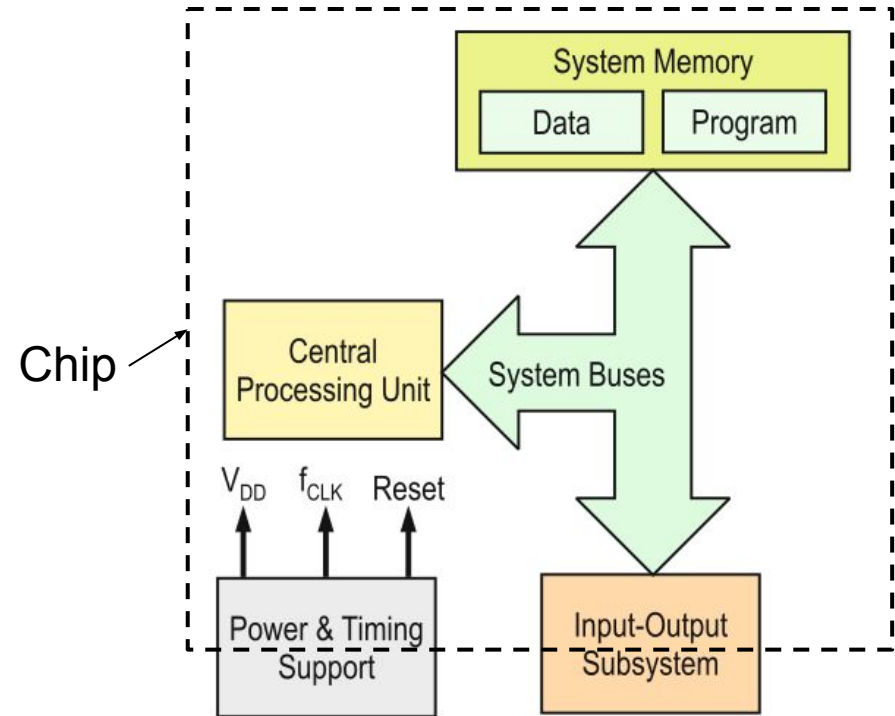
## Diagrama en bloques general de un sistema embebido



### Definición de microcontrolador



microprocesador



microcontrolador

## Diferencias entre procesadores y microcontroladores:

	Microcontroladores	Computadoras
Integración	Todo en un chip (procesador, memoria, periféricos, etc.)	Componentes separados
Evolución centrada en	Reducción consumo, Diferentes familias con diferente poder	Poder de procesamiento
SO	Usado en algunas aplicaciones (basadas en tiempo real), pero otras no	Obligatorio
Tamaño memorias	De pocos Kbytes a MBytes	Lo mayor posible (GBytes)



## **Sistemas embebidos e IoT**

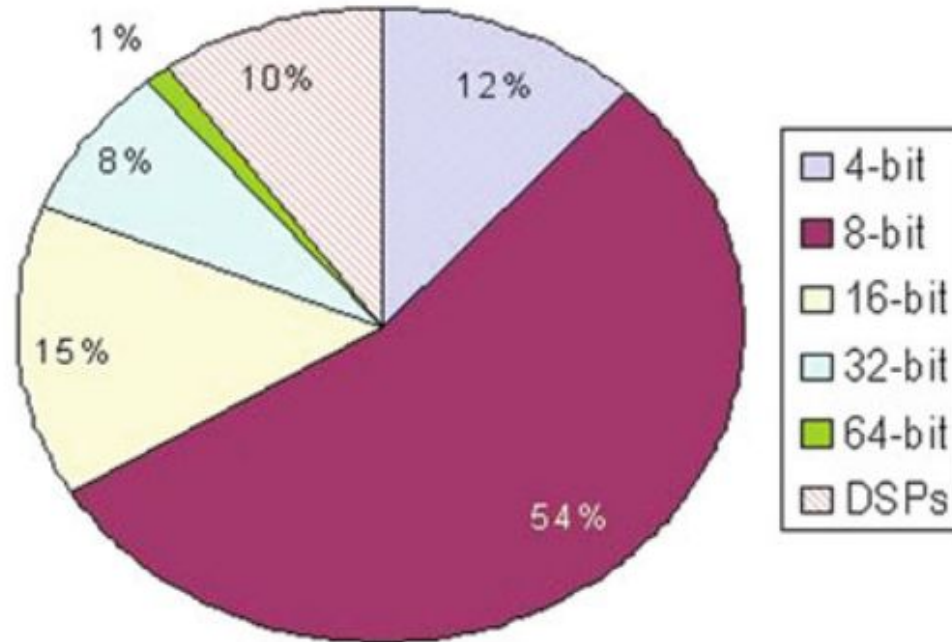
- **IoT o Internet de las cosas:** *“Presencia penetrante de variados dispositivos que contienen sensores y actuadores con un único esquema de direccionamiento, que son capaces de interactuar y cooperar unos con otros para lograr objetivos comunes”*
- Ejemplos: pequeños sistemas embebidos a teléfonos celulares, integrados o no con sistemas de computación como servidores o clouds.
- Requisitos de dispositivos IoT:
  - Cada dispositivo debe tener una dirección única: solución IPv6 y 6LoWPAN.
  - Cada dispositivo debe poder comunicarse (Wifi, 6LoWPAN, NFC, RFID, ZigBee, etc.)
  - Cada dispositivo debe tener sensores y actuadores.
  - Cada dispositivo necesita un microcontrolador.
  - Plataformas de visualización, análisis y almacenamiento de datos.

} SE

## Sistemas embebidos e IoT



## Distribución de procesadores según bits en años recientes



## Clasificación según número de bits:

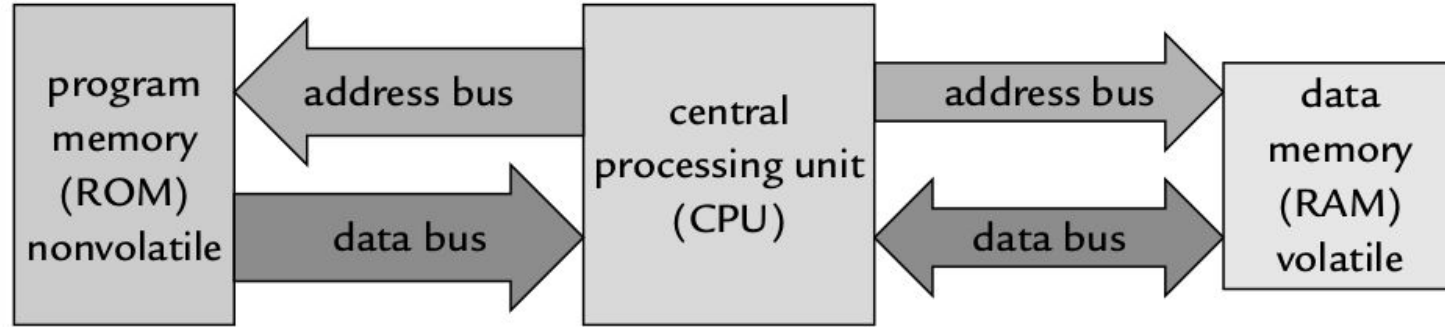
- Gama baja: 8 bits, 4 bits:
  - Periféricos específicos (sensores y motores). Conectados a pines.
  - Comunicación mediante protocolos específicos de bajo costo.
  - Aparatos muy simples, tostadora, lavarropas simples, máquinas de hornear pan, etc.
- Gama media: 16 bits:
  - Electrodomésticos con más funcionalidades (incluyen WiFi).
- Gama alta: 32 o más bits:
  - No muy diferentes a una computadora de propósito general + periféricos específicos de sistemas embebidos.
  - Comunicación a través de TCP/IP, Ethernet, Wifi.
  - Routers, equipos de red, SmartTVs.

## Clasificación según arquitectura.

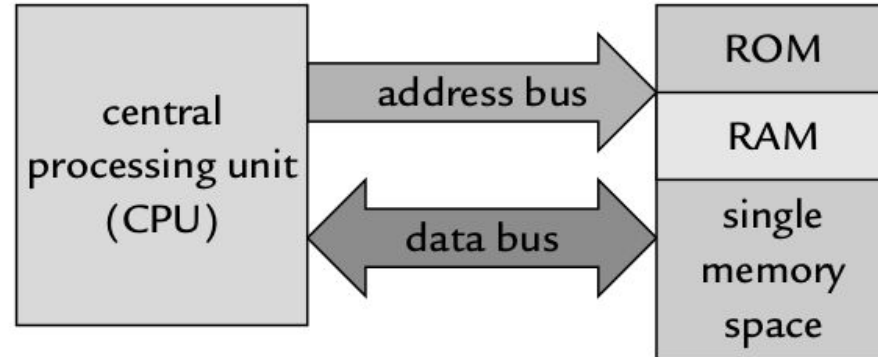
- Pequeños sistemas embebidos.
  - Único chip microcontrolador con casi todos los componentes. Solo algunos componentes externos al chip (componentes análogos, algunos sensores, etc.)
  - Juguetes, controladores de horno microondas, tostadoras, etc.
- Sistemas embebidos distribuidos.
  - Un CPU central y varios componentes y co-procesadores en otras ubicaciones.
  - Usualmente usan RTOS.
  - WSN, videojuegos, procesadores de redes, etc.
- Sistemas embebidos de alta performance.
  - Alta velocidad, robustez, alta tolerancia a fallas.
  - Incluyen DSPs y RTOS
  - CADC (procesador de datos de vuelo militar)



(a) Harvard architecture



(b) von Neumann architecture



**Microcontrolador  
estructura típica**

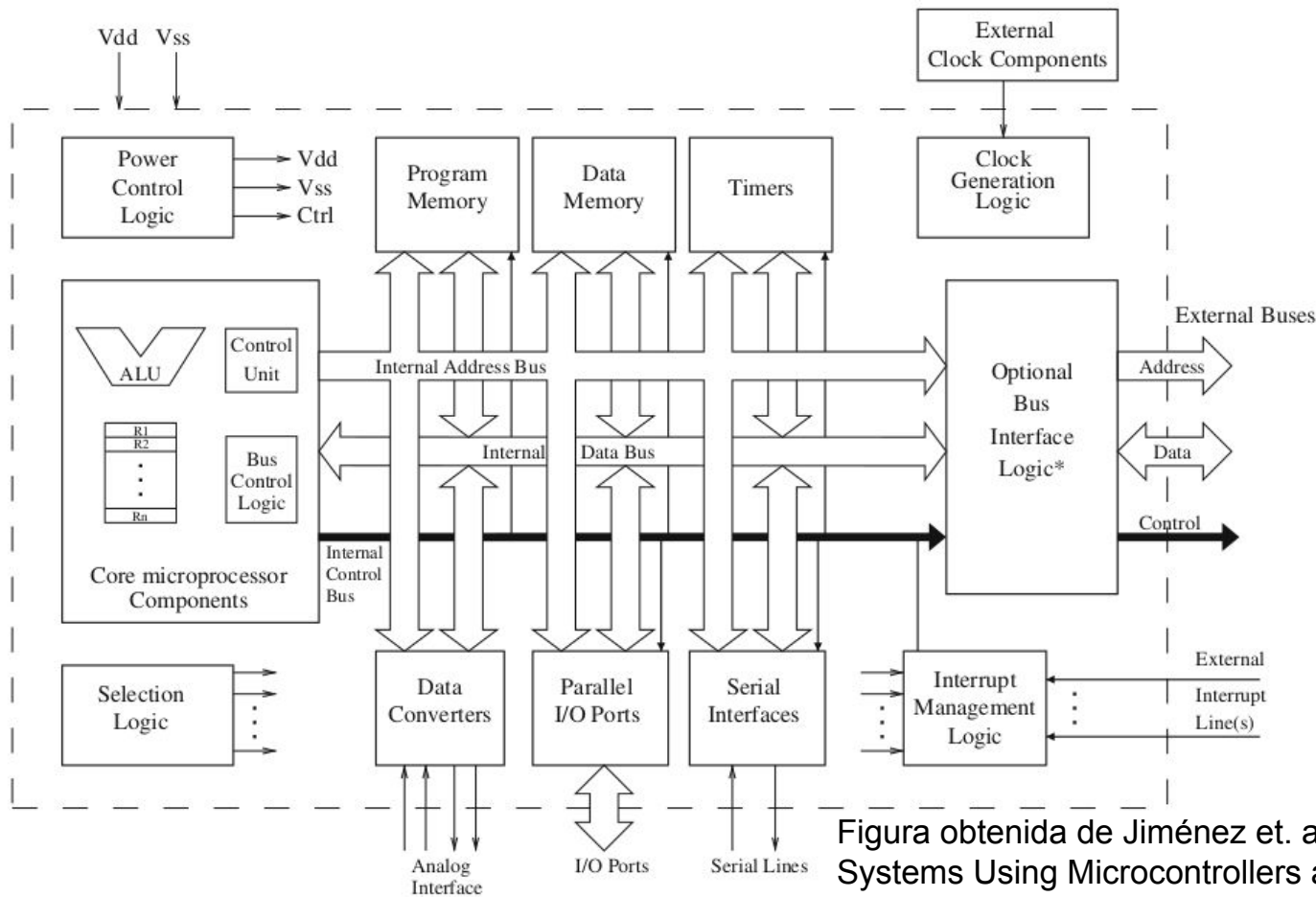
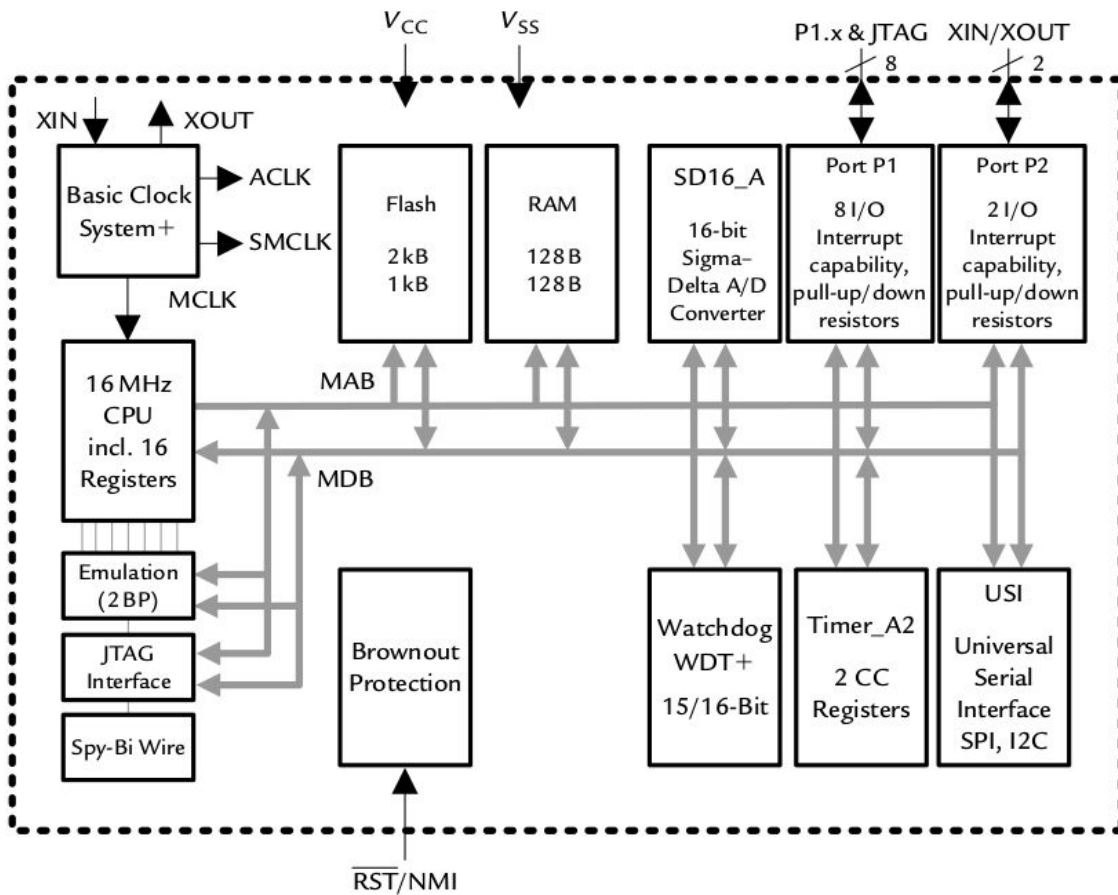


Figura obtenida de Jiménez et. al., "Introduction to Embedded Systems Using Microcontrollers and the MSP430". pag 84.



Microcontrolador MSP430  
Figura obtenida de: John  
Davies, "MSP430  
Microcontroller Basics", pag.  
24





## Aplicaciones más importantes de los sistemas embebidos

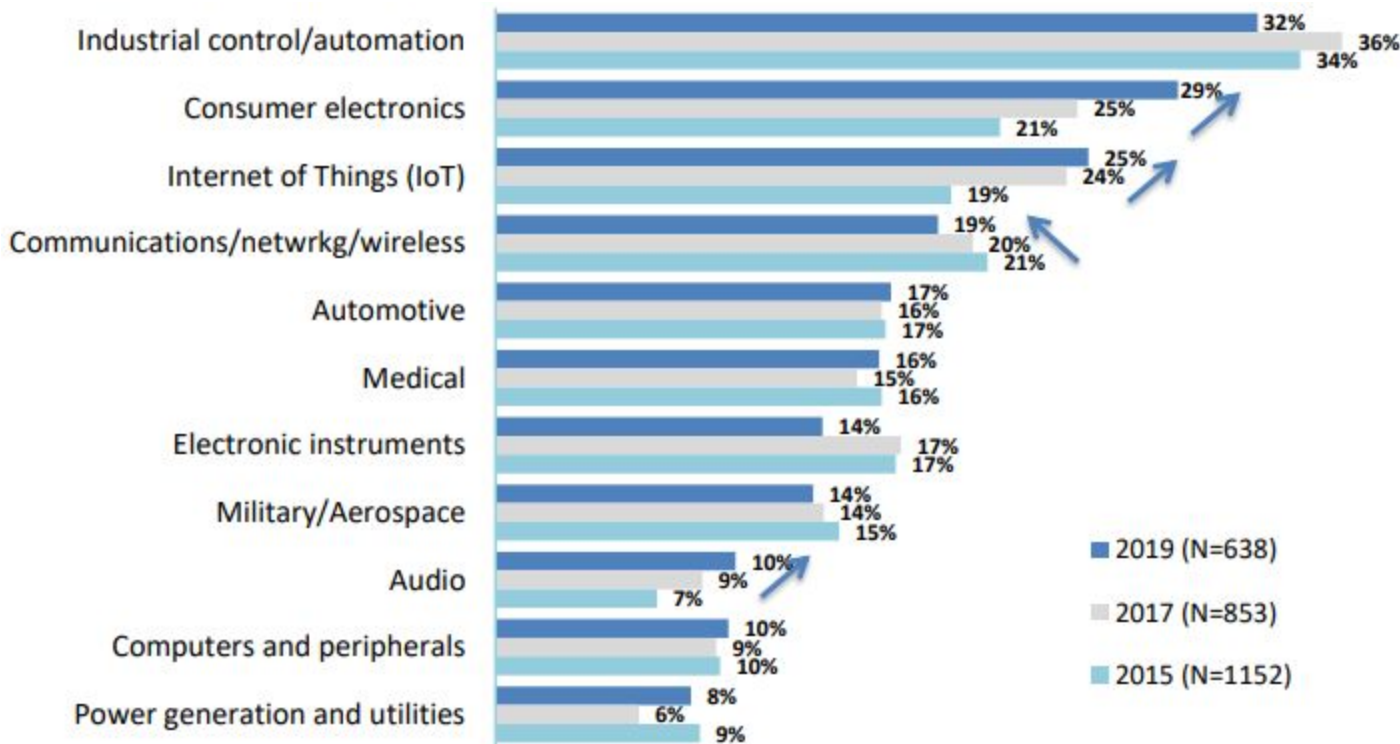


Figura obtenida de EETimes, “2019 Embedded Markets Study”.



## Aplicaciones más importantes de los sistemas embebidos

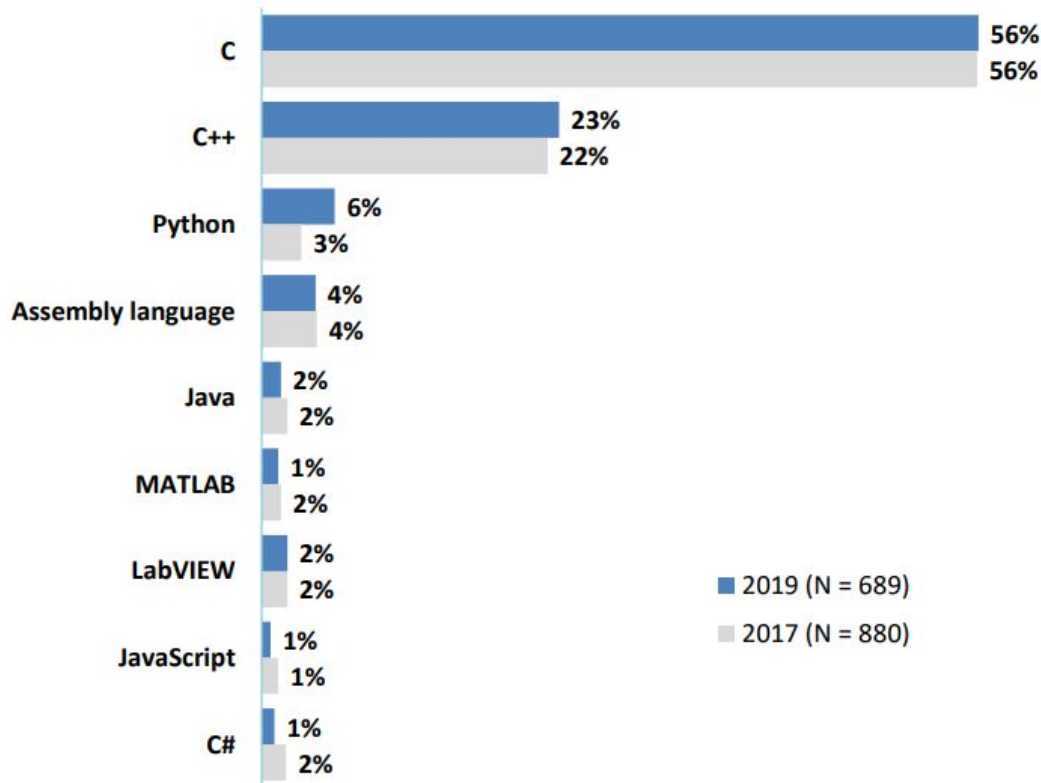


Figura obtenida de EETimes, “2019 Embedded Markets Study”.

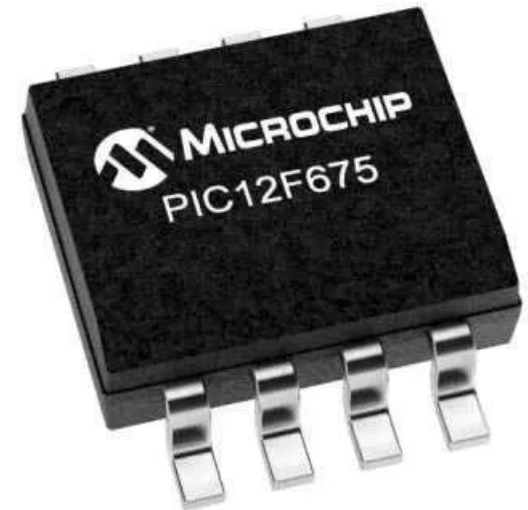
[https://www.embedded.com/wp-content/uploads/2019/11/EETimes\\_Embedded\\_2019\\_Embedded\\_Markets\\_Study.pdf](https://www.embedded.com/wp-content/uploads/2019/11/EETimes_Embedded_2019_Embedded_Markets_Study.pdf)

## Familia PIC

- Creados por Microchip Technology Inc.
- Arquitectura Harvard.
- Programable en lenguaje ensamblador.
  - Interfaces de desarrollo en lenguaje C (MPLAB).

8 bits (PIC 12F675).

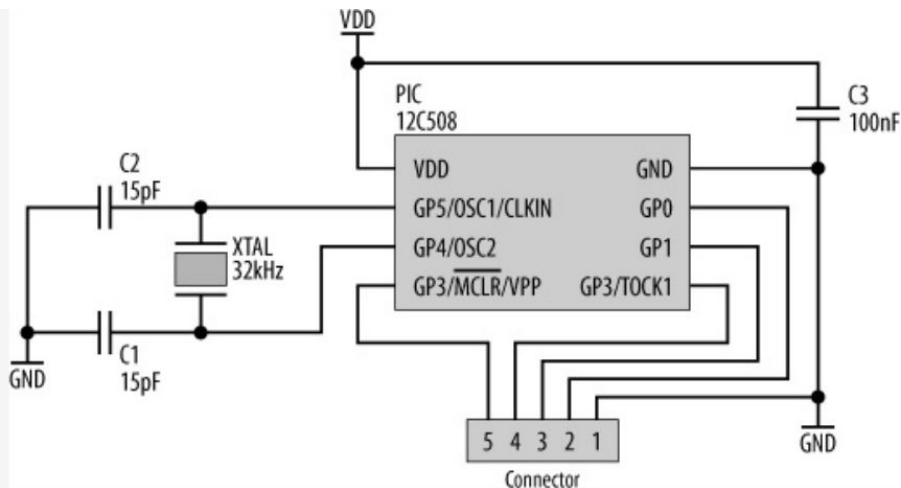
- Memoria de programa de 512 palabras.
- RAM de 25 bytes.
- Comunicación: I2C, SPI.
- Salidas: digitales y PWM.
- Frecuencias de operación:
  - 4 MHz
  - 32 KHz (bajo consumo)



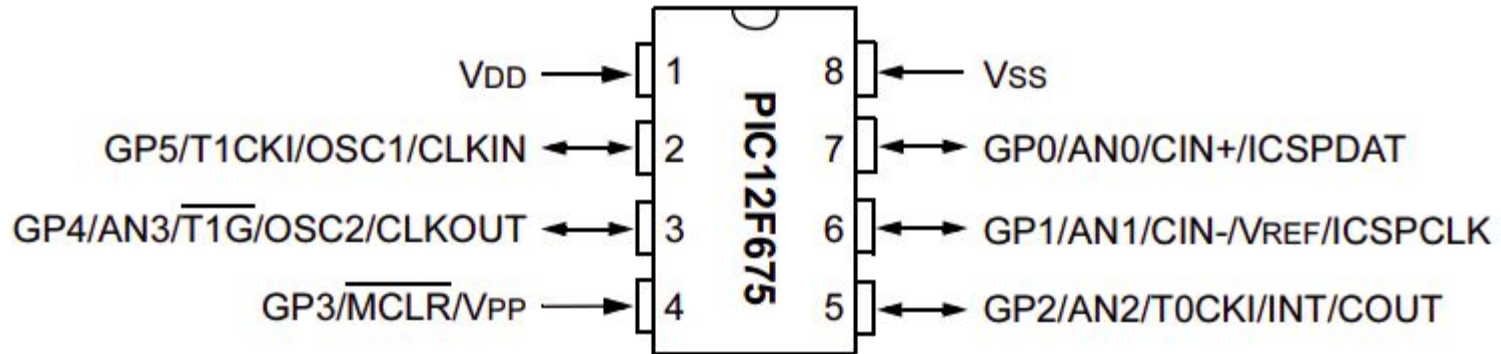
## Familia PIC

8 bits (PIC 12F675), continuación.

- ALU con registro acumulador (W).
- 35 instrucciones.
- Consumo:
  - Normal: 100 $\mu$ A (625 días con pilas AA).
  - Bajo consumo: 1 nA (171mil años con pilas AA).
- Herramientas desarrollo en C o ensamblador.
- Precio: 1.60 USD (Ebay)



**Familia PIC:8 bits (PIC 12F675).**



GP0: Entrada salida digital  
T1CKI: Timer 1 clock input  
AN0: Entrada analógica 0  
CIN+ y CIN-: Comparadores analógicos  
COU: Salida comparador  
ICSPDAT: Comunicación serial

Figura obtenida de Microchip, PIC12F629/675  
Data Sheet.

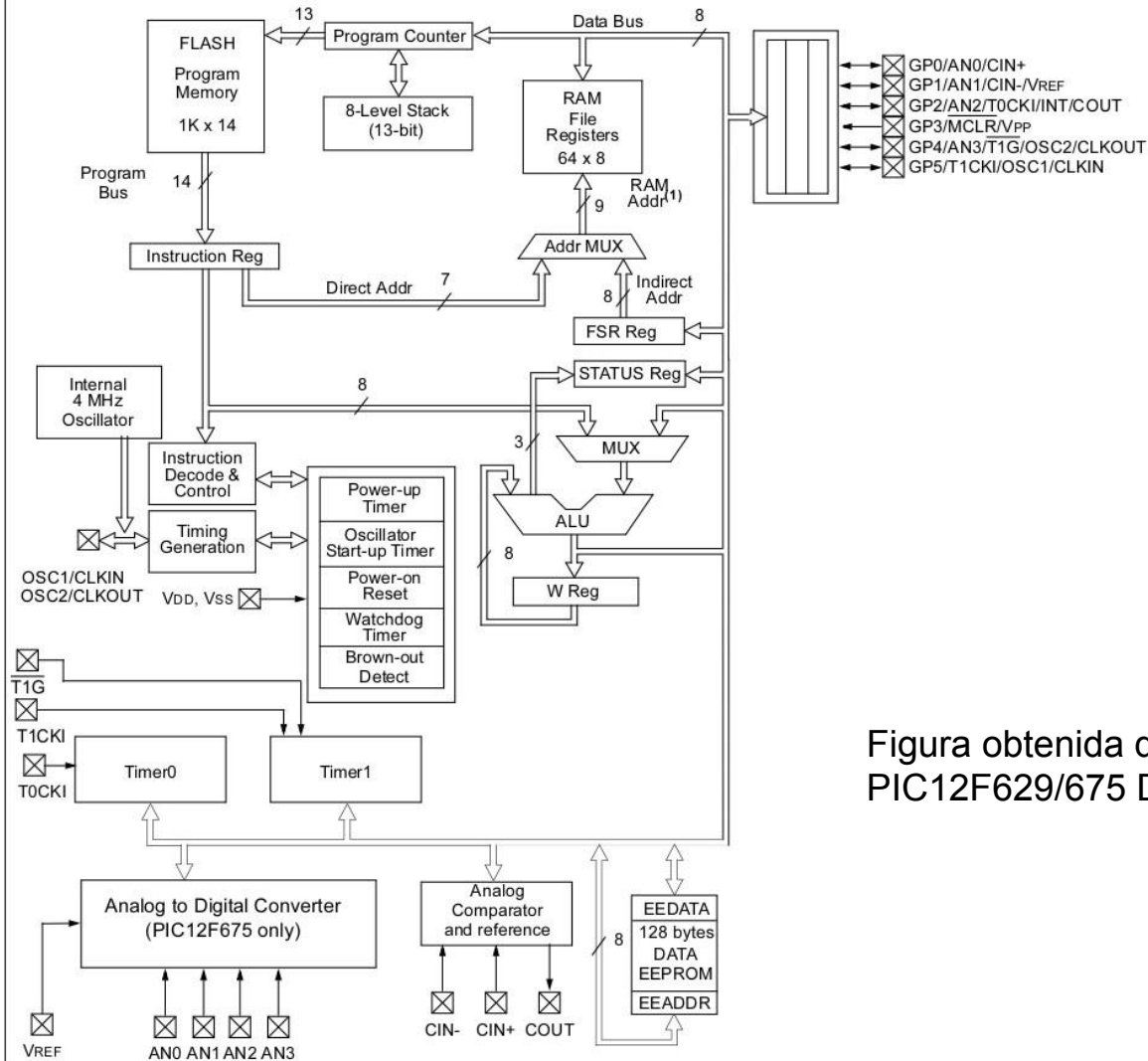


Figura obtenida de Microchip,  
PIC12F629/675 Data Sheet.



## Memoria RAM y mapeo en memoria

File Address	File Address
Indirect addr.(1)	Indirect addr.(1)
00h	00h
TMR0	TMR0
01h	01h
PCL	PCL
02h	02h
STATUS	STATUS
03h	03h
FSR	FSR
04h	04h
GPIO	GPIO
05h	05h
06h	06h
07h	07h
08h	08h
09h	09h
PCLATH	PCLATH
0Ah	0Ah
INTCON	INTCON
0Bh	0Bh
PIR1	PIR1
0Ch	0Ch
0Dh	0Dh
TMR1L	TMR1L
0Eh	0Eh
TMR1H	TMR1H
0Fh	0Fh
T1CON	T1CON
10h	10h
11h	11h
12h	12h
13h	13h
14h	14h
15h	15h
16h	16h
17h	17h
18h	18h
CMCON	CMCON
19h	19h
1Ah	1Ah
1Bh	1Bh
1Ch	1Ch
1Dh	1Dh
ADRESH(2)	ADRESH(2)
1Eh	1Eh
ADCON0(2)	ADCON0(2)
1Fh	1Fh
20h	20h
20h-5Fh	General Purpose Registers 64 Bytes
5Fh	5Fh
60h	60h
60h-DFh	accesses 20h-5Fh
DFh	DFh
E0h	E0h
E0h-FFh	Bank 0
FFh	7Fh
	Bank 1

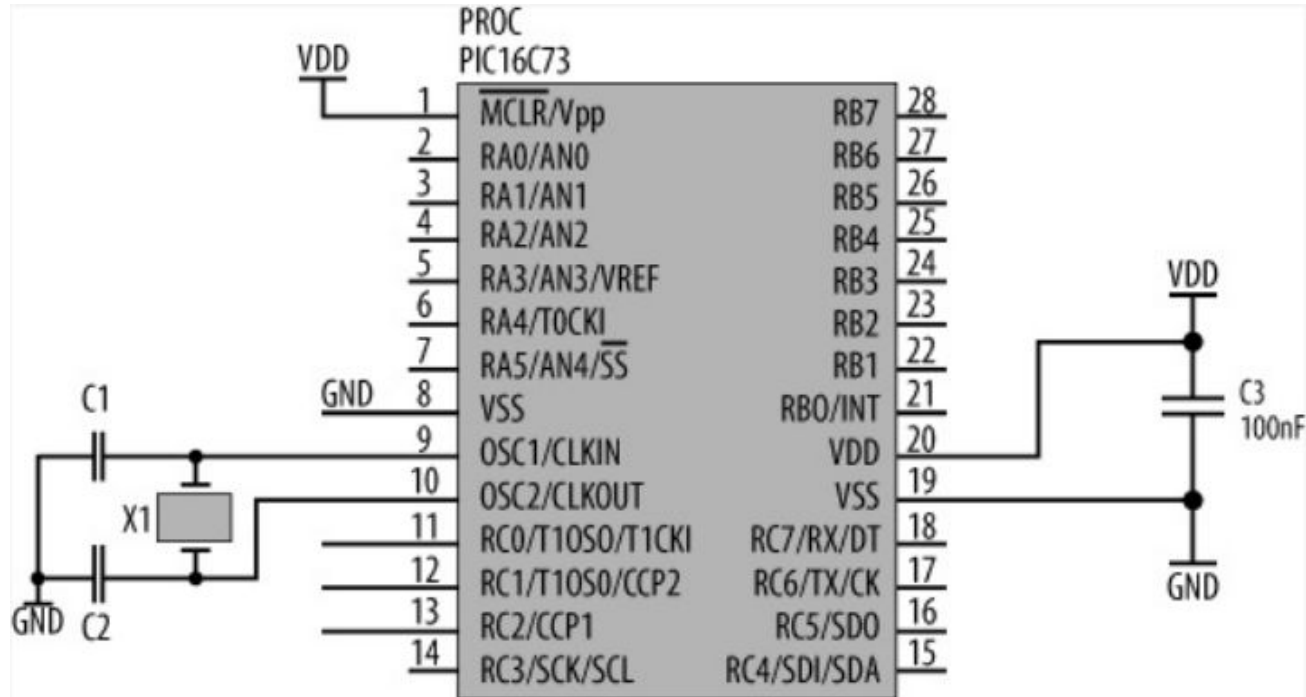
 Unimplemented data memory locations, read as '0'.  
 1: Not a physical register.  
 2: PIC12F675 only.

Figura obtenida de Microchip, PIC12F629/675 Data Sheet.

## Familia PIC

8 bits (PIC16C73).

- Memoria programa: 4K palabras de 14 bits.
- RAM: 192 bytes.
- Puertos: digitales y conversores AD y DA.
- Comunicación: I2C, UART y SPI.
- Frecuencia: 20 MHz (32 KHz bajo consumo).
- Memoria no volátil datos FLASH.
- Precio: 10 USD (Ebay)

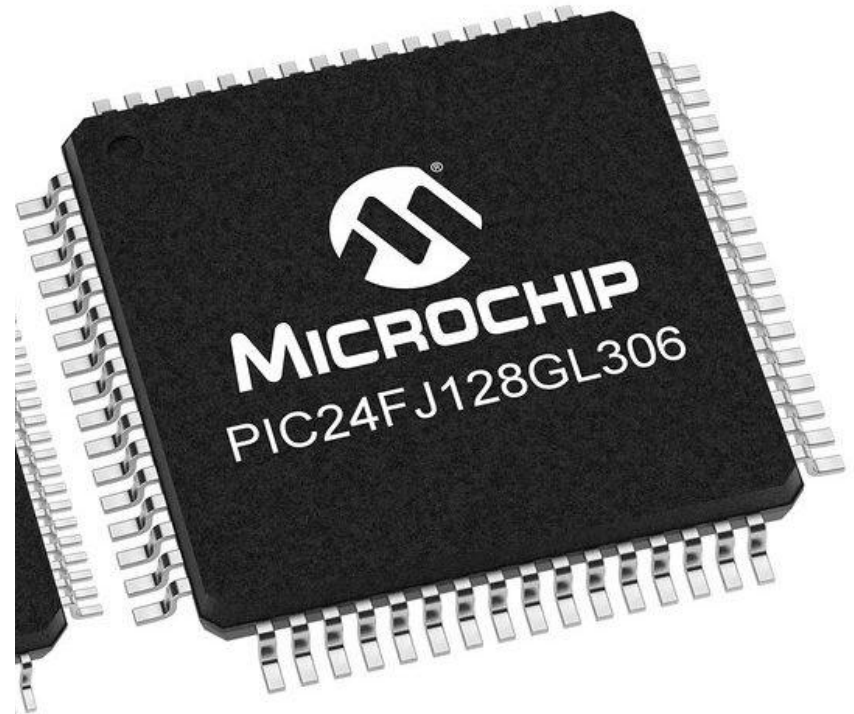


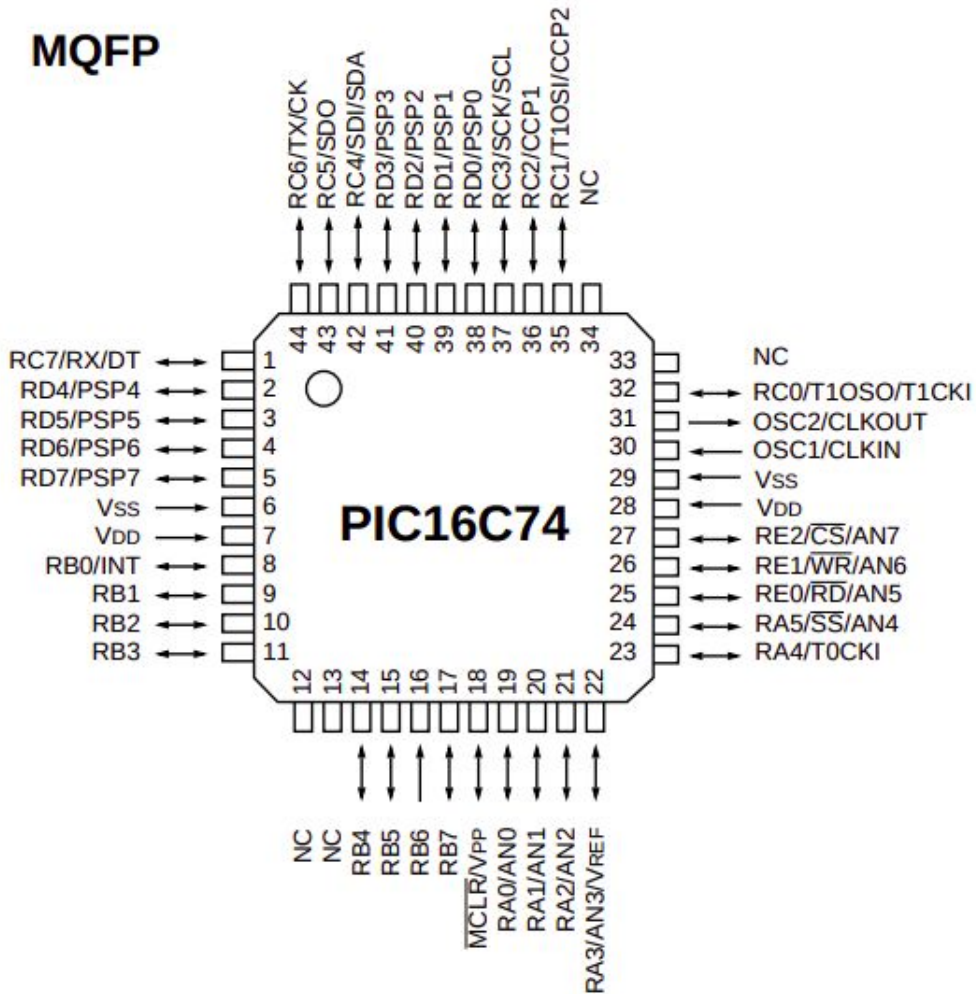


## Familia PIC

16 bits (PIC24F).

- Memoria programa: 256 K
- Memoria datos: 16 K
- Idem periféricos a los anteriores.
- Encriptación AES y DES.
- Precio: 10 USD (Ebay)



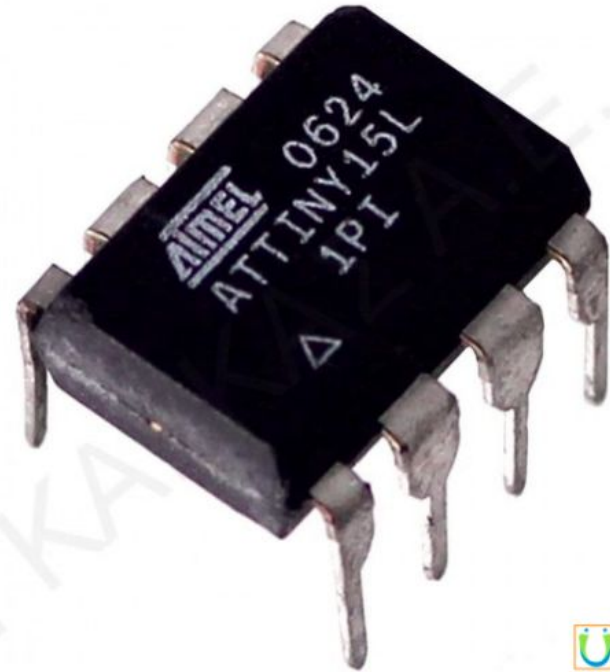


RAx-RBx...: Puertos de  
entrada salida.  
PSP: Parallel Slave Port

## Familia AVR

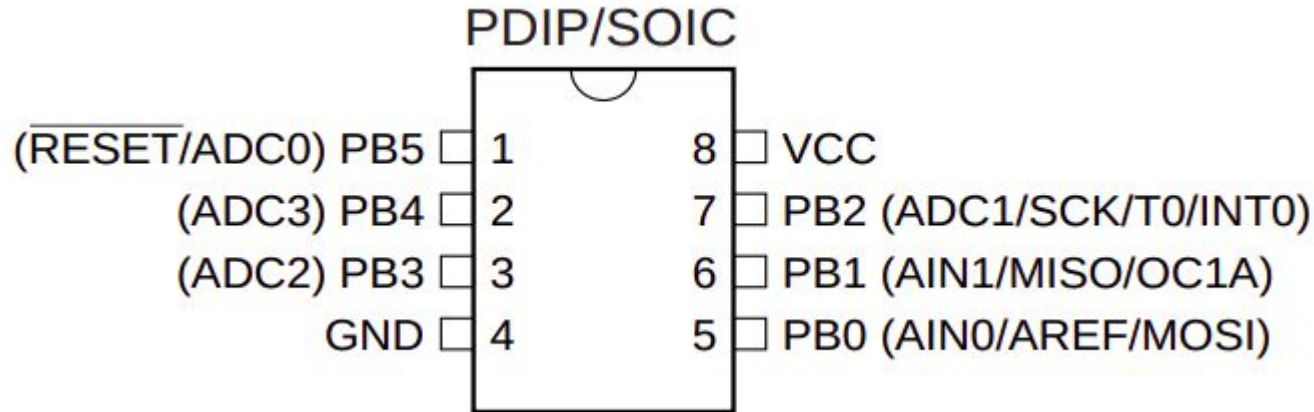
### Procesador ATtiny15

- Memoria programa: 512 palabras
- 90 Instrucciones
- Memoria RAM datos: No posee
- 32 registros.
- EEPROM: 64 bytes
- 5 pines de I/O.
- Conversores A/D
- Precio: 1.60 USD (Ebay)

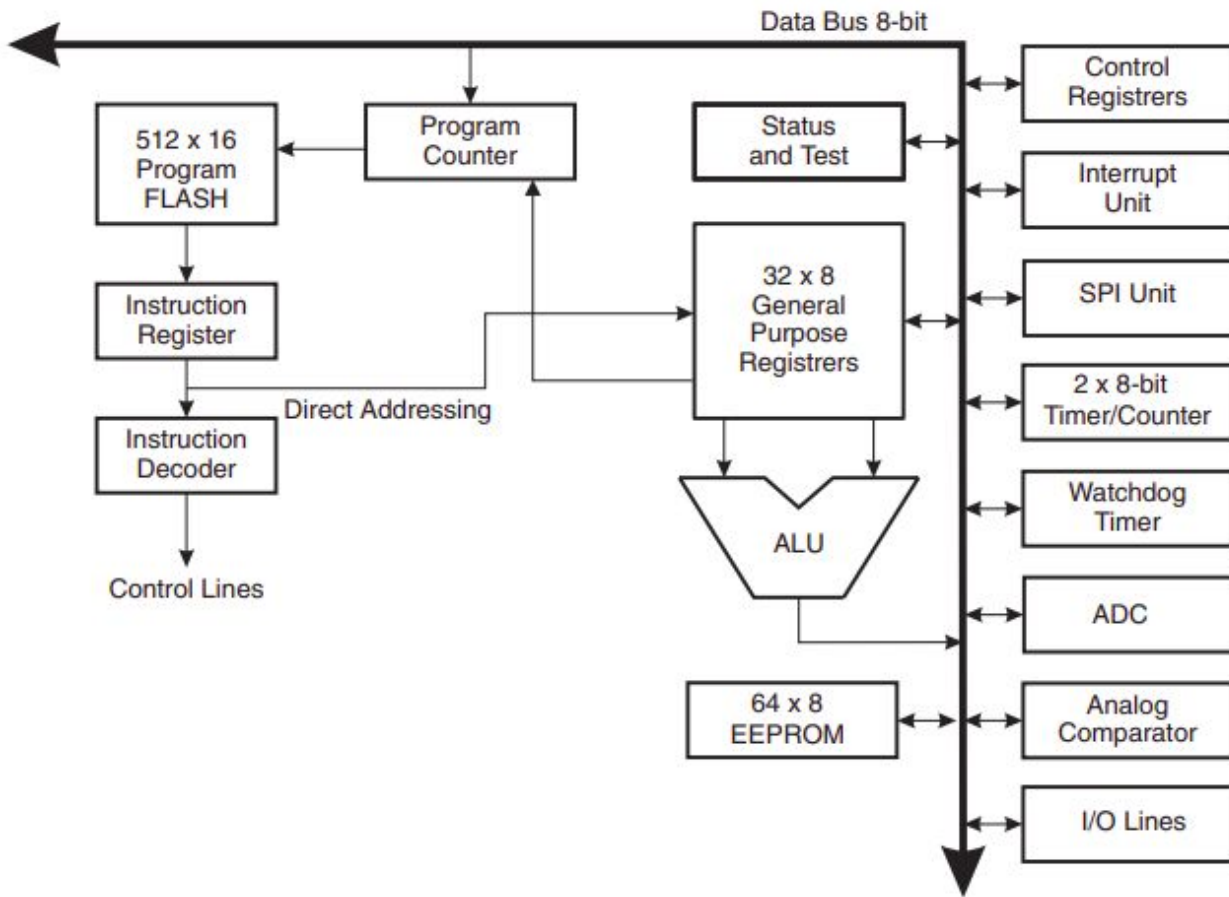


## Familia AVR

### Procesador ATtiny15



INT0: Interrupción externa



## **Familia AVR**

### **Procesador ATmega328**

- Memoria programa: 32KB
- 131 interrupciones
- Memoria RAM datos: 2 KB
- Memoria EEPROM: 1 KB
- frecuencia: 16 MHz
- Conversores AD y DA, SPI, I2C, USAR.
- Consumo:
  - Normal: 1.5 mA (41 días con pilas AA)
  - Bajo consumo: 1  $\mu$ A (171 años).
- Precio: 5.0 USD (Ebay)



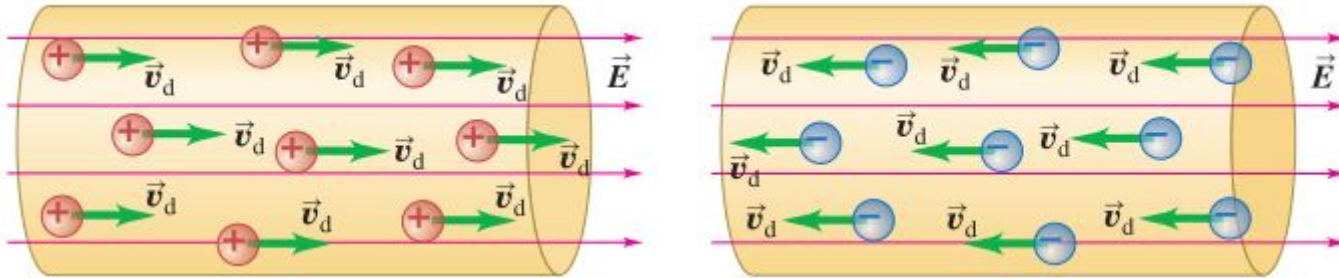


## **JTAG**

- Joint Test Action Group: Norma IEEE 1149.1
- Provee acceso al procesador y sus periféricos.
- Puerto que permite:
  - Escribir las memorias de programa.
  - In-circuit debug.
  - Examinar y escribir registros.
- Puerto de 4 pines.

## Breve introducción a la electrónica, sensores y actuadores

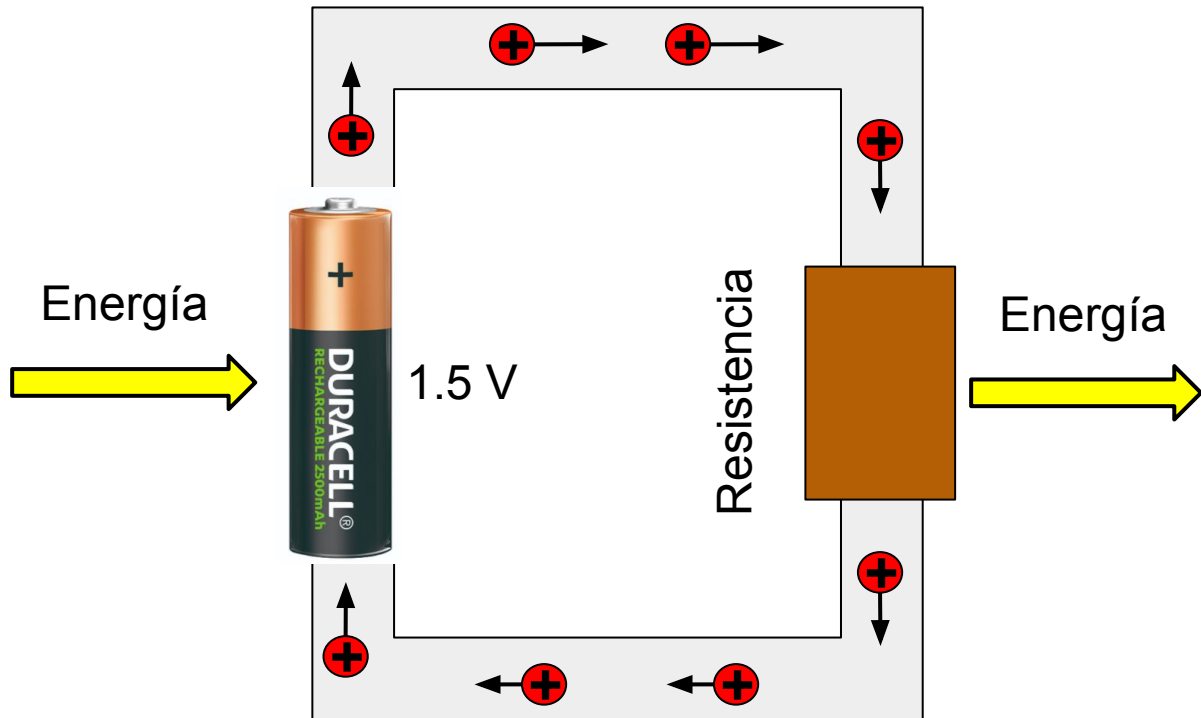
- Corriente eléctrica: Flujo de carga eléctrica por unidad de tiempo.  
 $I = dQ/dt$
- Análogo al caudal de agua a través de una tubería.
- Unidad: Amperes (A).
- Se mide siempre interponiendo el aparato de medición en el circuito.





## Breve introducción a la electrónica, sensores y actuadores

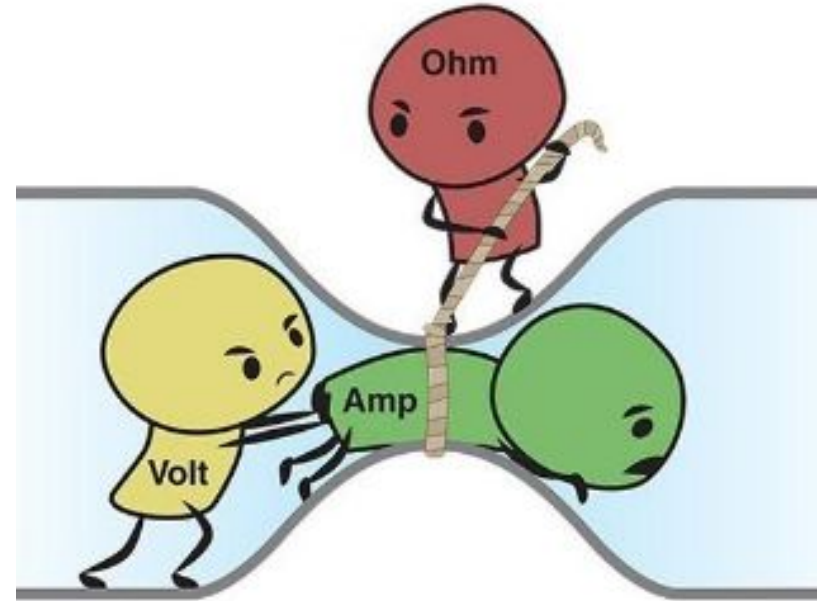
- Voltaje: Energía por unidad de carga.
  - Una fuente de alimentación suma esa energía.
  - Una resistencia quita esa energía.
- Unidad: voltaje (V).
- Se mide entre dos terminales.





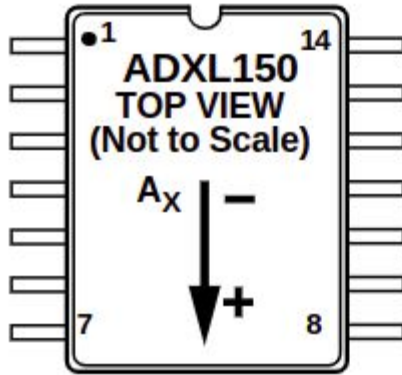
## Breve introducción a la electrónica, sensores y actuadores

- Resistencia:  $R=V/I$  (Ley de Ohm).
- Unidad: Se mide en Ohm ( $\Omega$ )

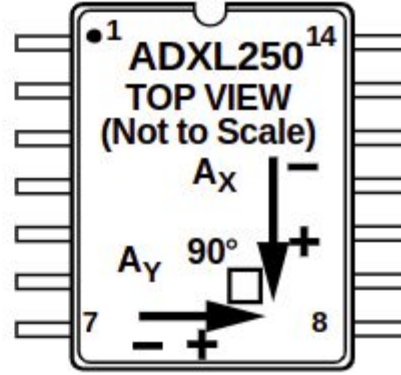




## Acelerómetro



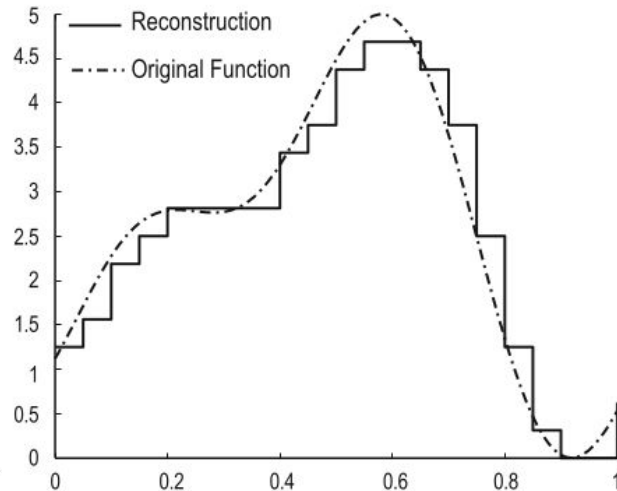
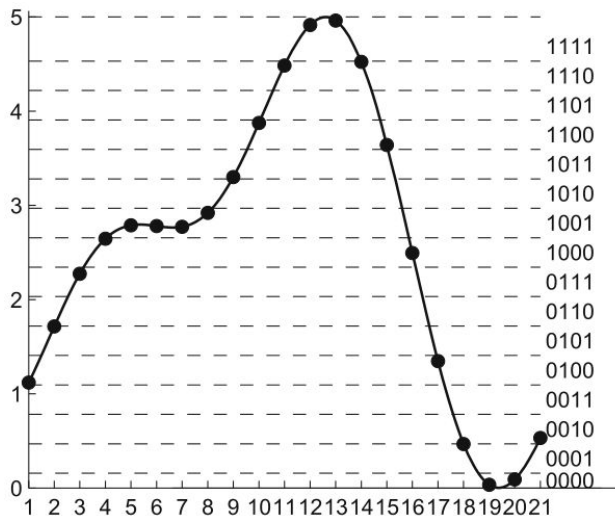
POSITIVE A = POSITIVE  $V_{OUT}$



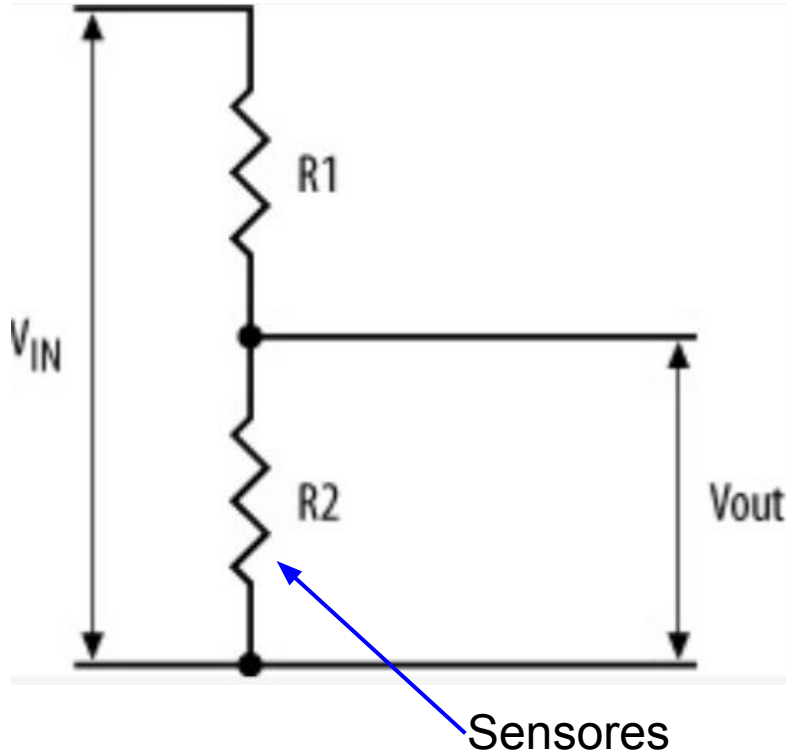
POSITIVE A = POSITIVE  $V_{OUT}$

$$V_{OUT} = V_S/2 - (\text{sensitivity} * V_S/5 * \text{acceleration})$$

**Micrófono (R variable)**



## Divisor de tensión



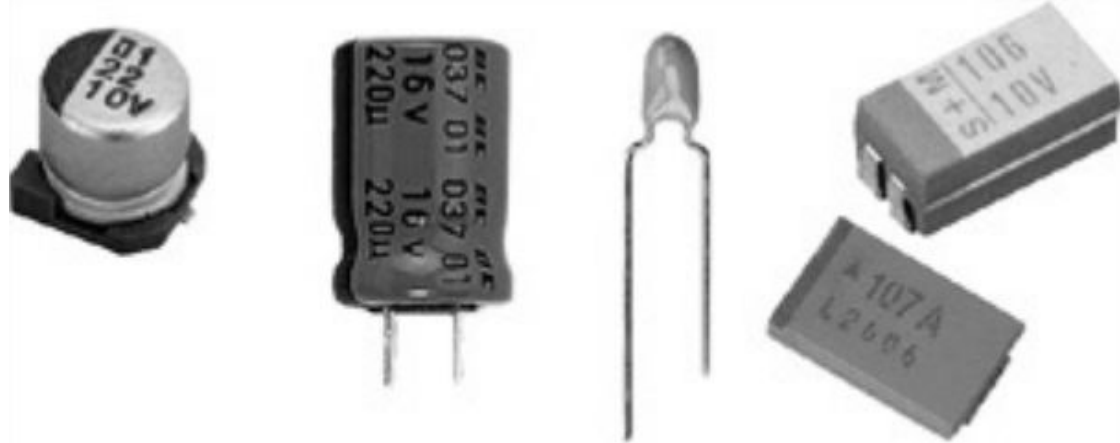
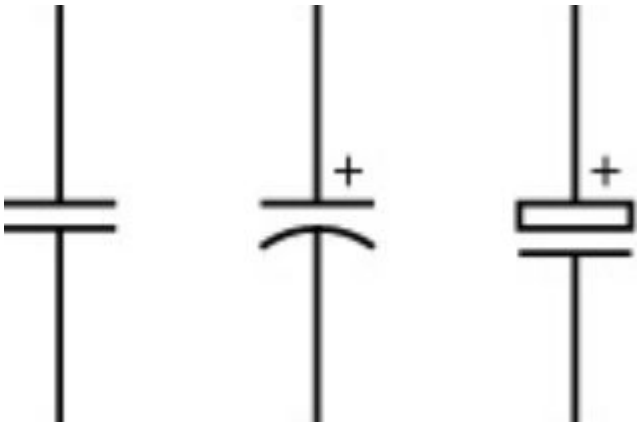
$$V_{OUT} = V_{IN} * R2 / (R1 + R2)$$

Ejemplo:

$$\begin{aligned} V_{OUT} &= 5V \quad 1k / (1k + 1k) \\ &= 5V \quad 1k / 2k \\ &= 5V * 0.5 \\ &= 2.5V \end{aligned}$$

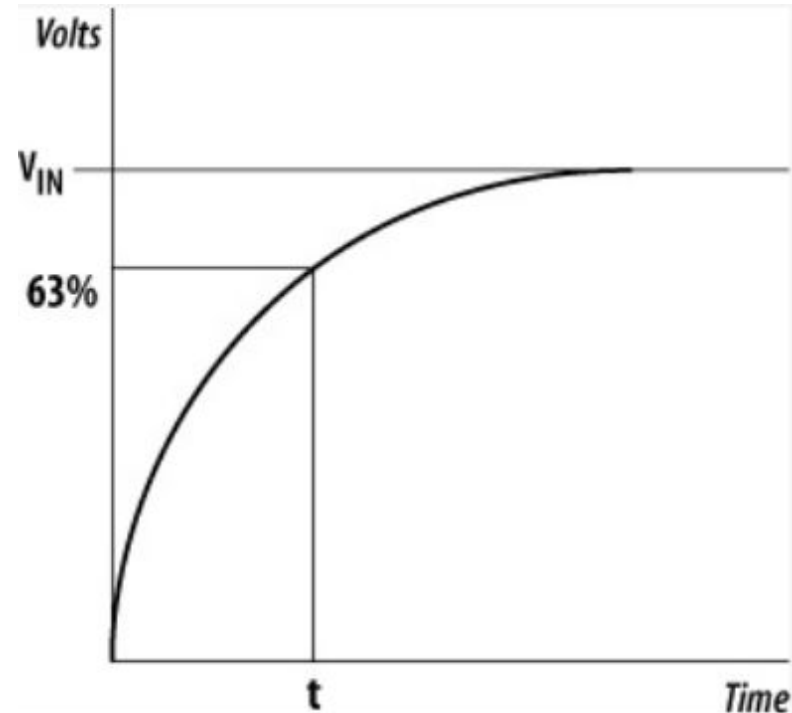
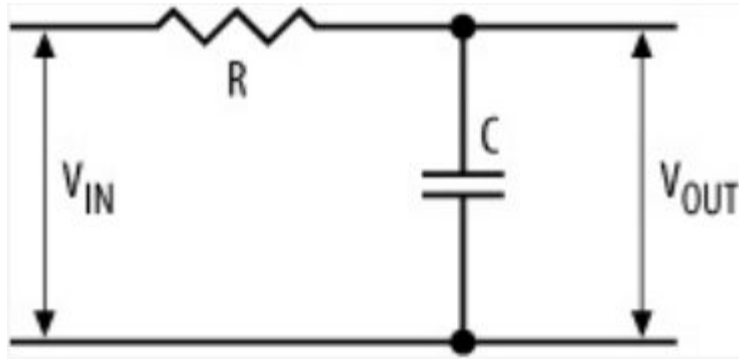
## Capacitor

- Elemento capaz de almacenar energía.
- Análogo a un tanque de agua.
- Unidades: Faradio





## Capacitor







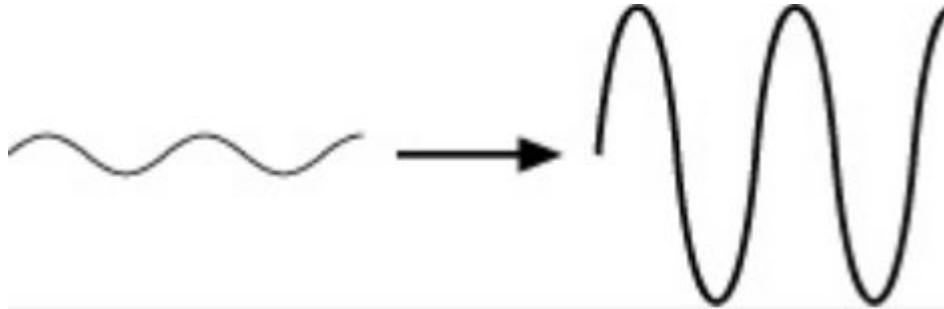
**UNCUYO**  
UNIVERSIDAD  
NACIONAL DE CUYO

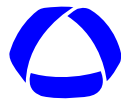


**FACULTAD  
DE INGENIERÍA**

**Licenciatura en Ciencias de la  
Computación**

**Amplificador**





## **Programación de sistemas embebidos**

- **Ensamblador**
  - Referencia a registros y posiciones de memoria.
  - Uso de instrucciones de máquina.
  - Acceso completo al hardware.
- **Lenguaje de alto nivel.**
  - C, C++, Python, Java, NodeJs, etc.
  - Referencias a registros y posiciones de memoria:
    - Microcontroladores pequeños si.
    - Microcontroladores potentes no (se trabaja con variables).
  - Pueden ocultar detalles y componentes de hardware.

## Bibliografía:

- John Davies, "MSP430 Microcontroller Basics". Elsevier. 2008.
- Manuel Jiménez, Rogelio Palomera, Isidoro Couvertier, "Introduction to Embedded Systems Using Microcontrollers and the MSP430". Springer. 2014.
- Peter Marwedel, "Embedded System Design. Embedded Systems, Foundations of Cyber-Physical Systems, and the Internet of Things". Springer. 2018.
- Perry Xiao , "Designing Embedded Systems and the Internet of Things (IoT) with the ARM Mbed". Wiley. 2018.
- Microchip, PIC12F629/675 Data Sheet