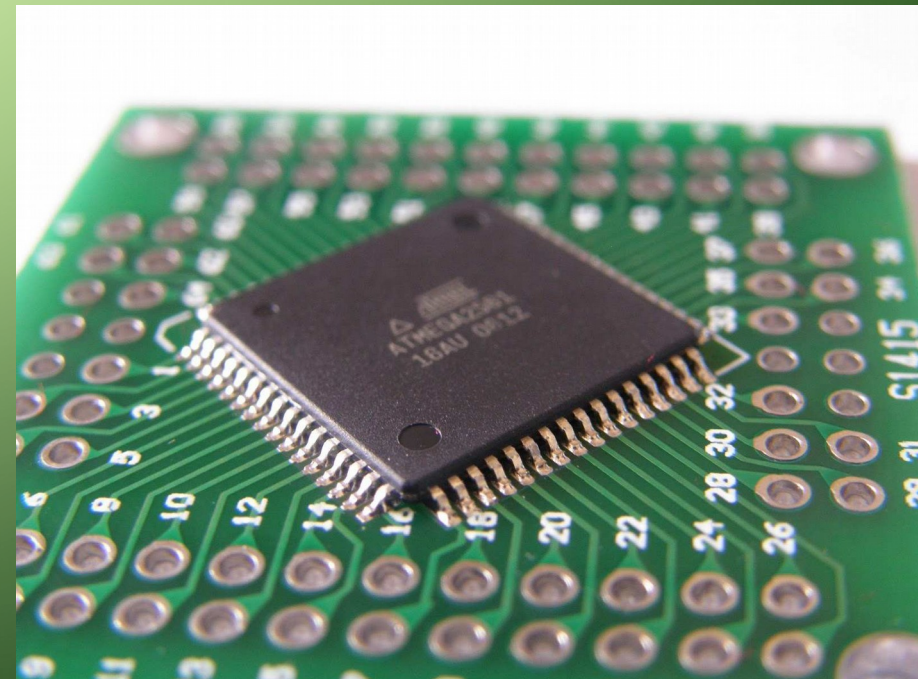


# INTRODUCCIÓN A PROGRAMACION DE MICROCONTROLADORES

PROGRAMACIÓN EN C - INFORMÁTICA 2



# ¿QUÉ ES UN MICROCONTROLADOR?

- Un microcontrolador es un circuito integrado que incorpora en su interior un procesador, memorias y una serie de periféricos que lo hacen apto para desempeñar distintas funciones. Su funcionamiento depende del programa que tenga cargado en su memoria.

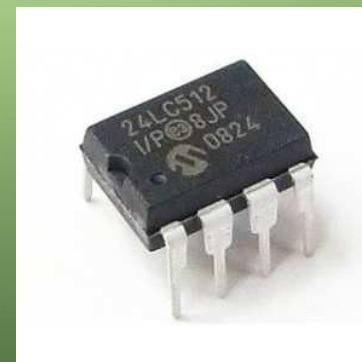
# ¿QUÉ ES UN MICROPROCESADOR?

- Un microprocesador es un circuito integrado complejo que se encarga de ejecutar instrucciones de bajo nivel.
- Esta compuesto por:
  - Unidad de control
  - Conjunto de registros
  - Unidad aritmético y lógica
  - Buses de conexión
  - Unidad de cálculo de coma flotante



# MEMORIAS

- EPROM\*
- EEPROM
- FLASH
- RAM
- NVRAM\*



# PERIFÉRICOS EN MICROCONTROLADORES

Tipos:

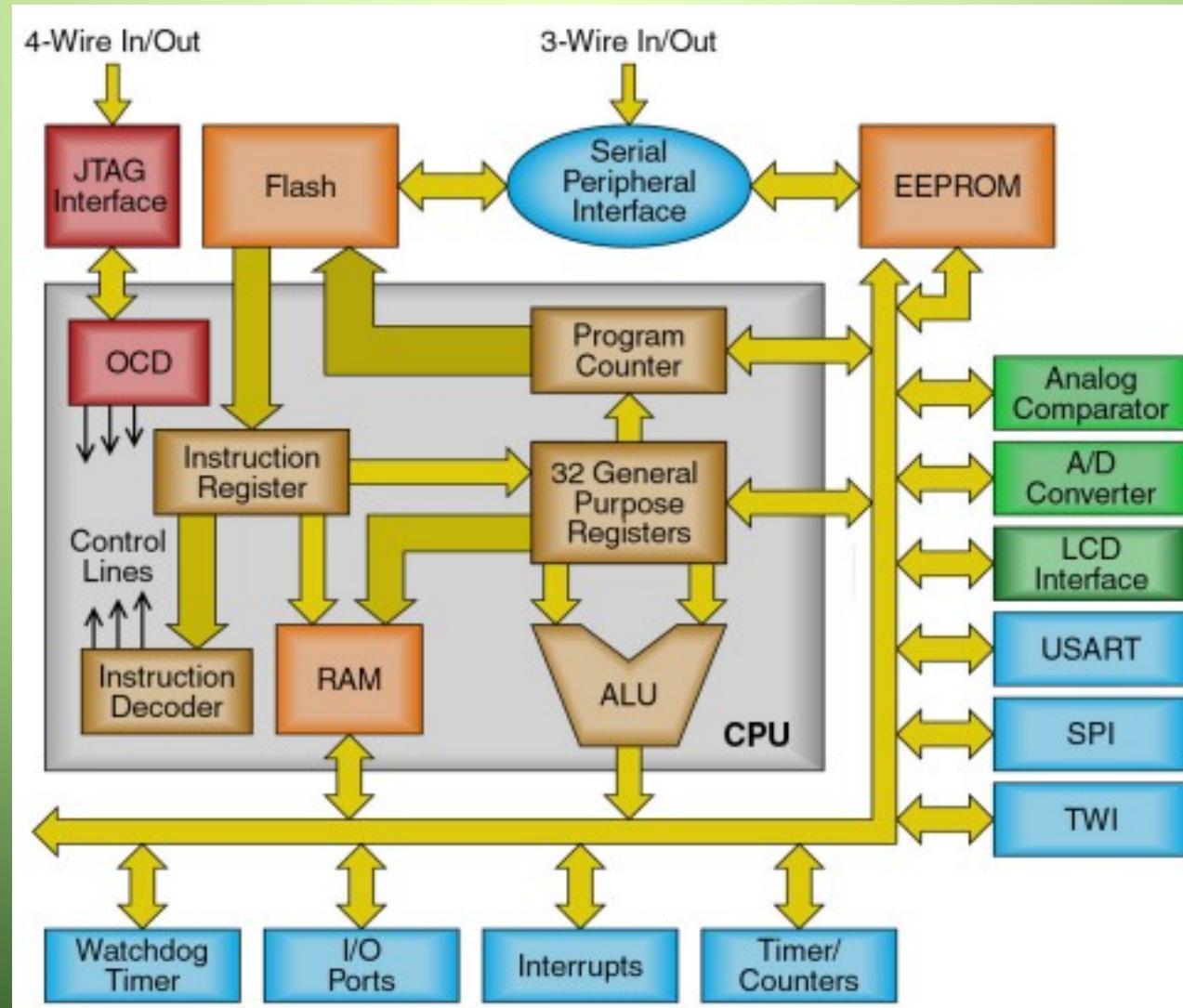
- Puertos de Entradas / Salidas
- Timers/Contadores
- Generador de interrupciones
- Interfaces de comunicación
- Conversor Analógico / Digital
- Conversor Digital / Analógico

# MEMORIA Y PERIFÉRICOS

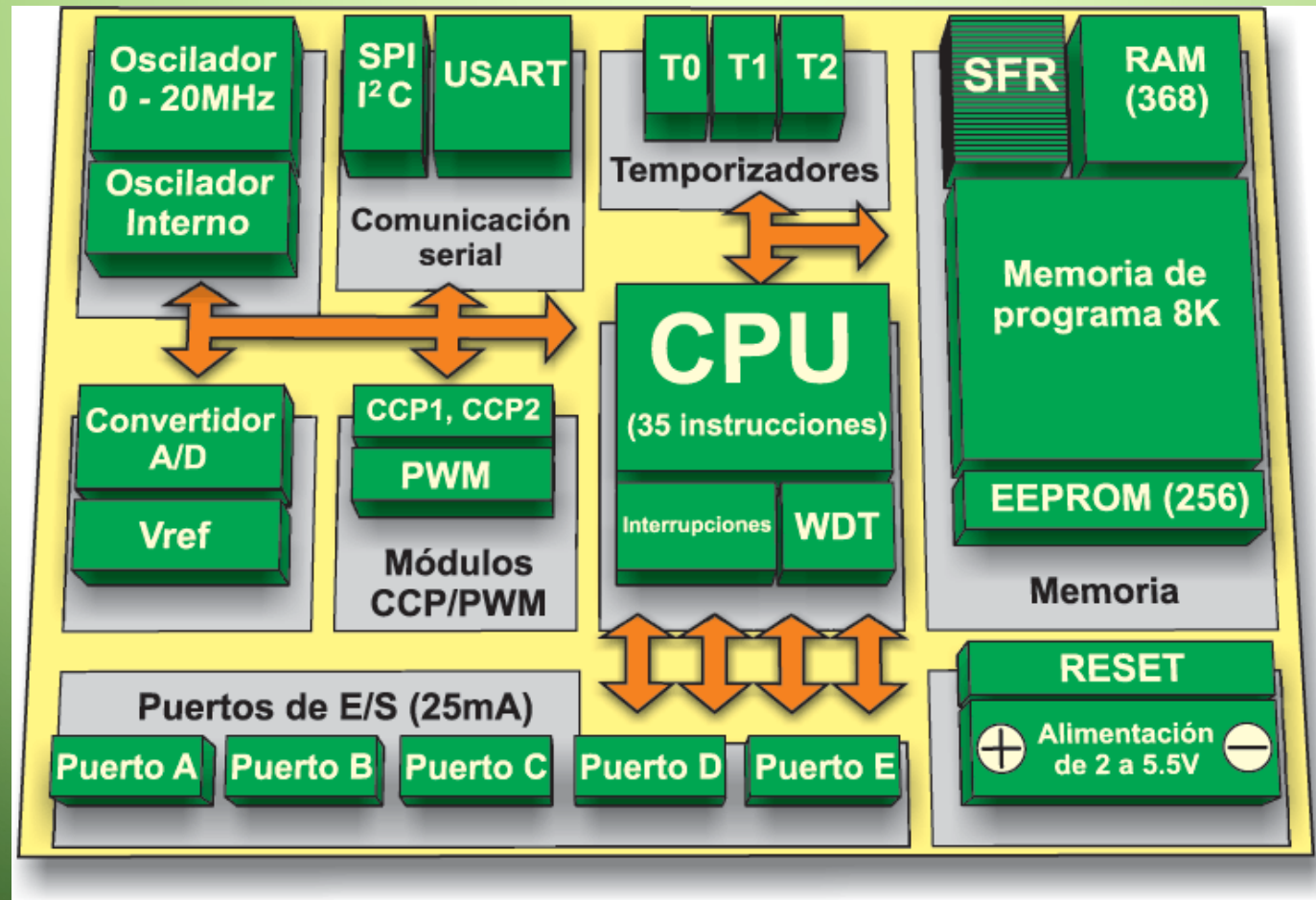
- El tipo y cantidad de periféricos, va a depender de cada fabricante. Lo mismo el tamaño y tipo de las memorias. Se debe elegir al microcontrolador de acuerdo a las necesidades del proyecto a implementar. Los fabricantes ofrecen cuadros comparativos entre sus distintas familias, los cuales permiten ver de forma detallada las distintas características de cada producto



# ARQUITECTURA AVR

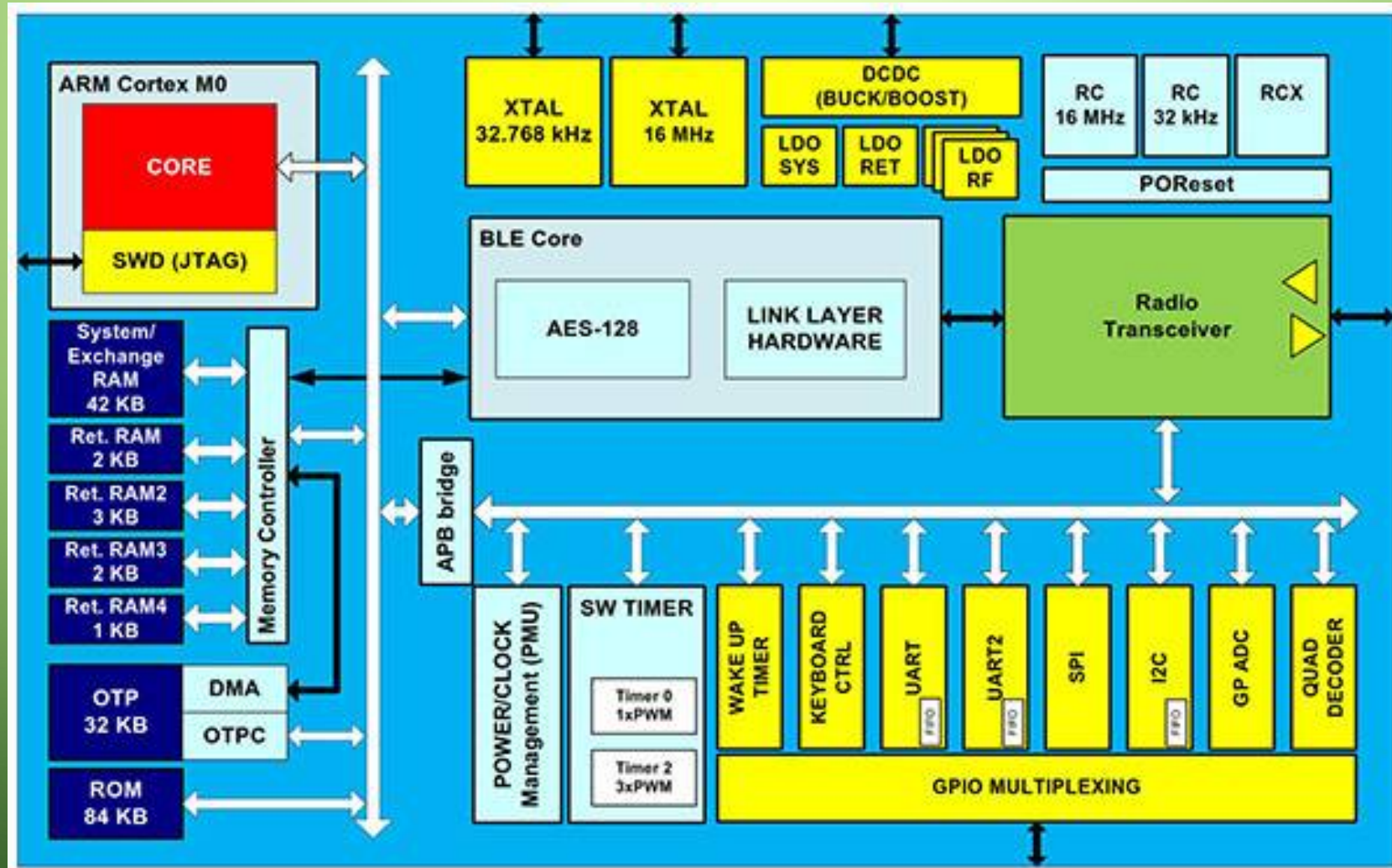


# ARQUITECTURA PIC





# ARQUITECTURA ARM



# REDUCED COMPLEXITY INSTRUCTION SET COMPUTER (RISC)

- Reduced COMPLEXITY Instruction Set Computer. No significa un número de instrucciones reducido como a menudo se piensa. Significa que se reduce la complejidad de los circuitos encargados de la decodificación de las instrucciones , haciéndolos más simples y eficientes.

# RISC

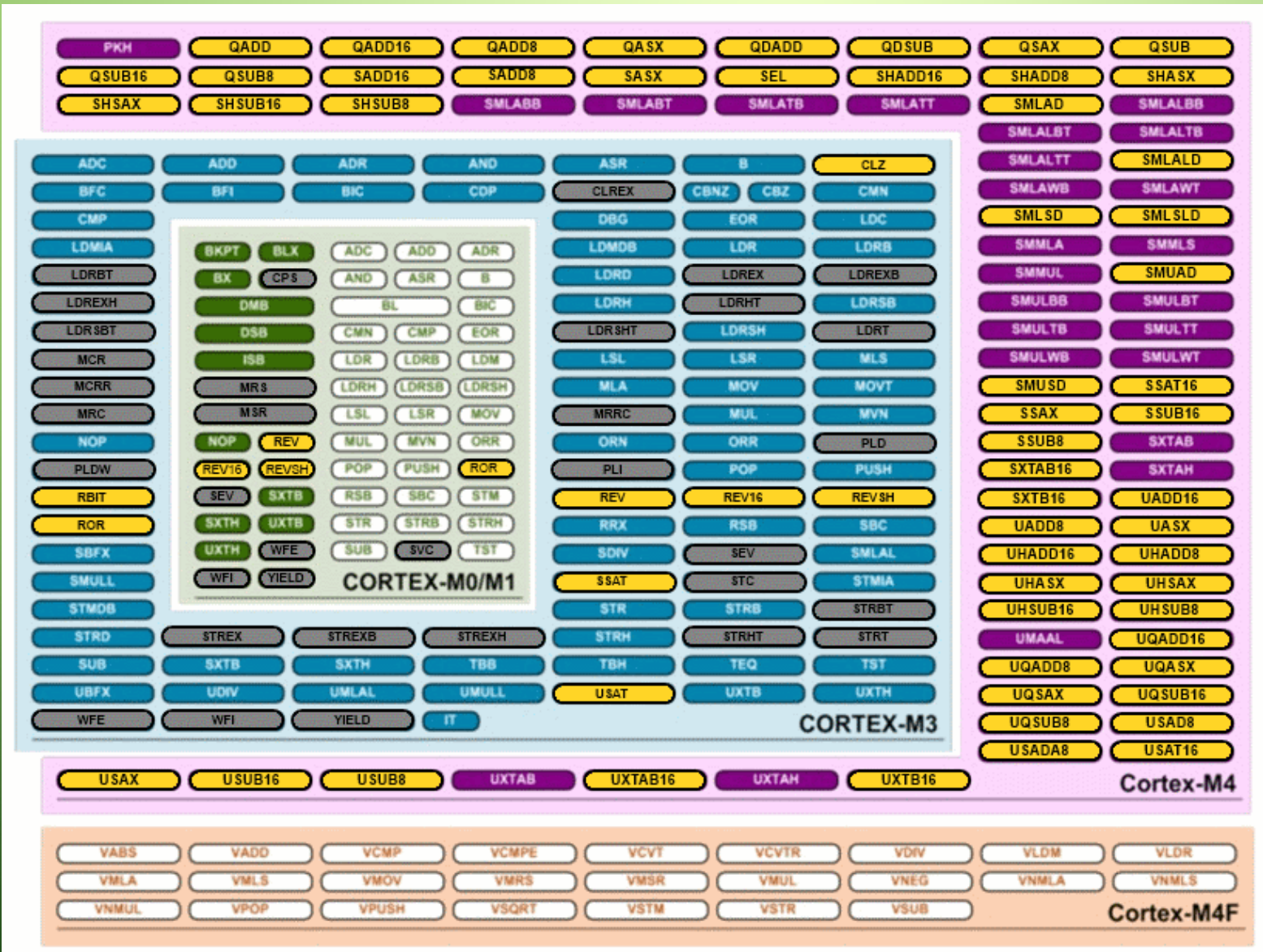
- La gran mayoría de los Microcontroladores son de arquitectura RISC. Los microcontroladores PIC poseen solo el set de instrucciones mas reducido del mercado. Por esa razón se convirtieron, desde sus inicios, en los mas fáciles y por ende en los de mayor difusión.

Con la implementación de compiladores en C para microcontroladores, esto dejo de ser una ventaja para convertirse en perjudicial.

Microcontroladores, como los AVR tienen un set de instrucciones mas amplio y compatible con el control de flujo de C por lo que se logran códigos mas reducidos y por lo tanto mas rápidos y eficientes.



# SET DE INSTRUCCIONES ARM



# ¿QUÉ MÁS DEFINE LA ARQUITECTURA?

La arquitectura de un microcontrolador no solo está definida por su set de instrucciones, sino también por la arquitectura del bus de comunicaciones.

- Arquitectura de Von Newmann:

Los buses de acceso a la memoria de programa y a la memoria de datos son los mismos, resultando así un único mapa de memoria

- Arquitectura Harvard:

Se utilizan buses separados para la memoria de programa y la de datos.



# ¿QUE TENEMOS QUE TENER EN CUENTA AL PROGRAMAR UN MICROCONTROLADOR?

- La memoria no es infinita.

Los microcontroladores tienen una cantidad de Flash y RAM limitada, sobre todo la RAM. Por lo que es importante elegir con cuidado los tipos de variables de nuestro código para optimizar su uso.

- Cada Función guarda su contexto en RAM.

Si hacemos muchos llamados a funciones en forma recursiva, llamado dentro de llamados, corremos el riesgo de llenar la memoria y colgar al microcontrolador. Las funciones recursivas no son aconsejables en los microcontroladores.



# ¿MACROS O FUNCIONES?

- Las funciones guardan contexto, y por lo tanto su llamado en tiempo de ejecución consume memoria RAM.
- Las Macros son sentencias de preprocesador que en realidad se convierten en bloques de código insertados en el lugar de su llamado. Por eso, una macro no consume RAM adicional al llamarse pero si consumirá mas ROM.

# ¿POR QUÉ HACER EL CÓDIGO REUTILIZABLE?

- Como la tecnología avanza rápidamente y los microcontroladores cambian, se dejan de fabricar, o se dejan de conseguir, es imprescindible hacer que nuestro código sea lo mas PORTABLE posible.

# ¿COMO LO HAGO?

- Usando definiciones para todo lo relacionado al hardware propietario del micro.
- Separando en funciones simples y genéricas.
- Separando el código en diferentes archivos C y H para hacer el trabajo mas simple.

Resumiendo: Programando con las buenas practicas de C.



# COMO PROGRAMO EN C UN MICROCONTROLADOR

- Hoy en día los fabricantes proveen definiciones y, en algunos casos, funciones básicas para manejar cada uno de los periféricos. Por lo que esa será nuestra librería a incluir
- A diferencia de una PC, no tenemos Sistema operativo a cual regresar al finalizar el programa. Por lo que nuestro código deberá estar dentro de un ciclo infinito.

```
for(;;){      while(1){  
    }        }
```

The background is a green gradient. In the corners, there are decorative circuit-like patterns made of thin green lines and small circles, resembling a printed circuit board (PCB) layout.

# VEAMOS ALGUNOS EJEMPLOS...