

2 - Sistemas embebidos

Sistemas embebidos

CPU, Memorias, dispositivos de E/S y buses

Sistema microprocesador ejemplo: Arduino



Sistemas embebidos

■ ¿Qué es un sistema embebido/empotrado?

Es un sistema electrónico de **propósito específico** donde el procesador/computador está totalmente dedicado al dispositivo que controla y encapsulado en él. Al contrario de un computador de propósito general, como un ordenador personal, un sistema empotrado lleva a cabo **tareas predefinidas** con requisitos muy específicos (*no valdrían para otra cosa*).



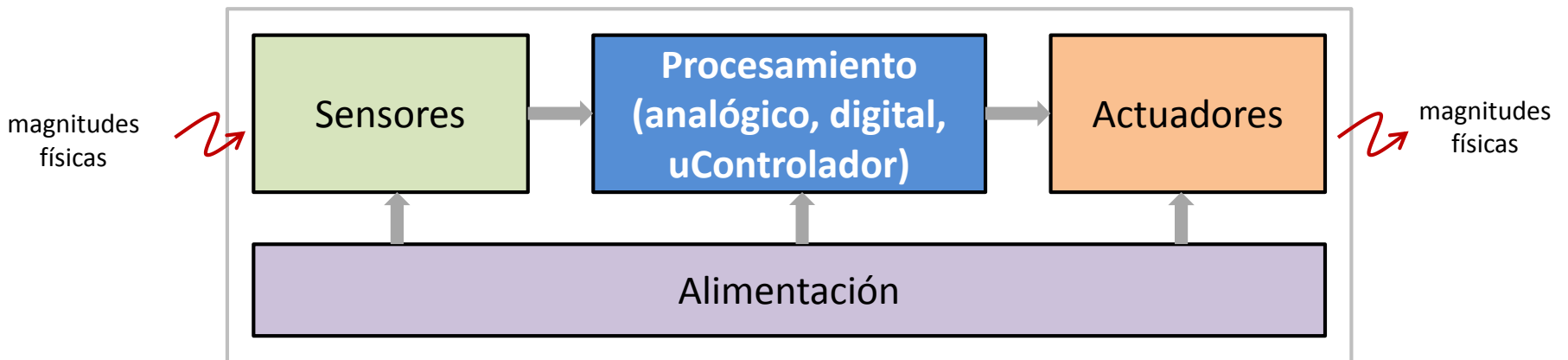
■ ¿Por qué un sistema embebido?

Al estar dedicados a una tarea específica, es posible **optimizarlo**, reduciendo el tamaño, coste y/o consumo. Los sistemas embebidos generalmente se producen en masa, así que los ahorros pueden ser multiplicados por millones.



Sistemas embebidos: elementos

- Como todo sistema electrónico dispone de...



- En un sistema embebido el procesamiento (al menos en parte), es realizado por un microcontrolador que ejecuta un programa de control para cumplir con un unas determinadas tareas o funcionalidades.

Sistemas embebidos: ejemplos



Sensores

Actuadores

Funcionalidades

Sistemas embebidos: ejemplos



Sensores

Botones
Selectores
Temperatura
Corriente
Humedad
Velocidad
Vibración
Nivel
Presión
Escapes
...

Actuadores

Motores
Pantalla
Indicadores
Válvulas
Cerrojos
Calefactor
Bombas
Ventiladores
...
...
...

Funcionalidades

Control lavado
Control secado
Interfaz usuario
Monitorización
Mantenimiento
Seguridad
Ecología
...
...
...

Sistemas embebidos: ejemplos



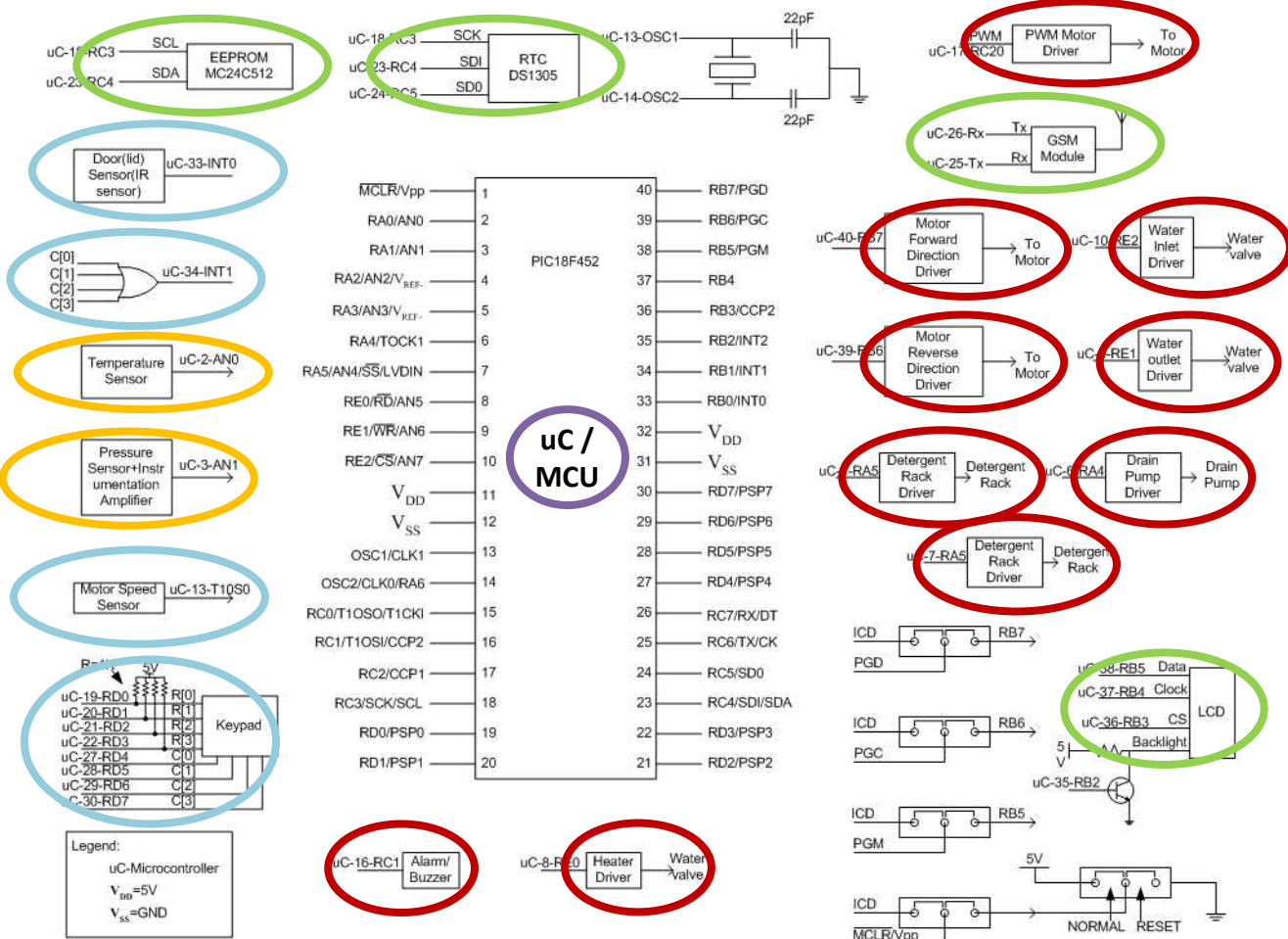
Entradas digitales

Entradas analógica

Salidas digitales

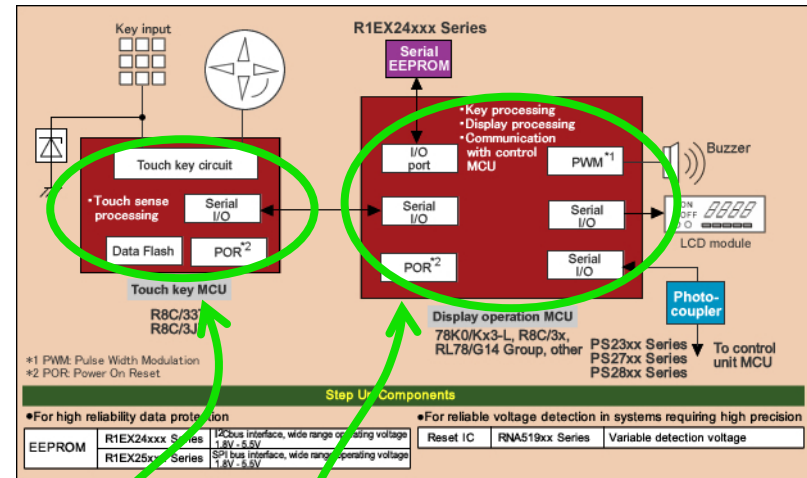
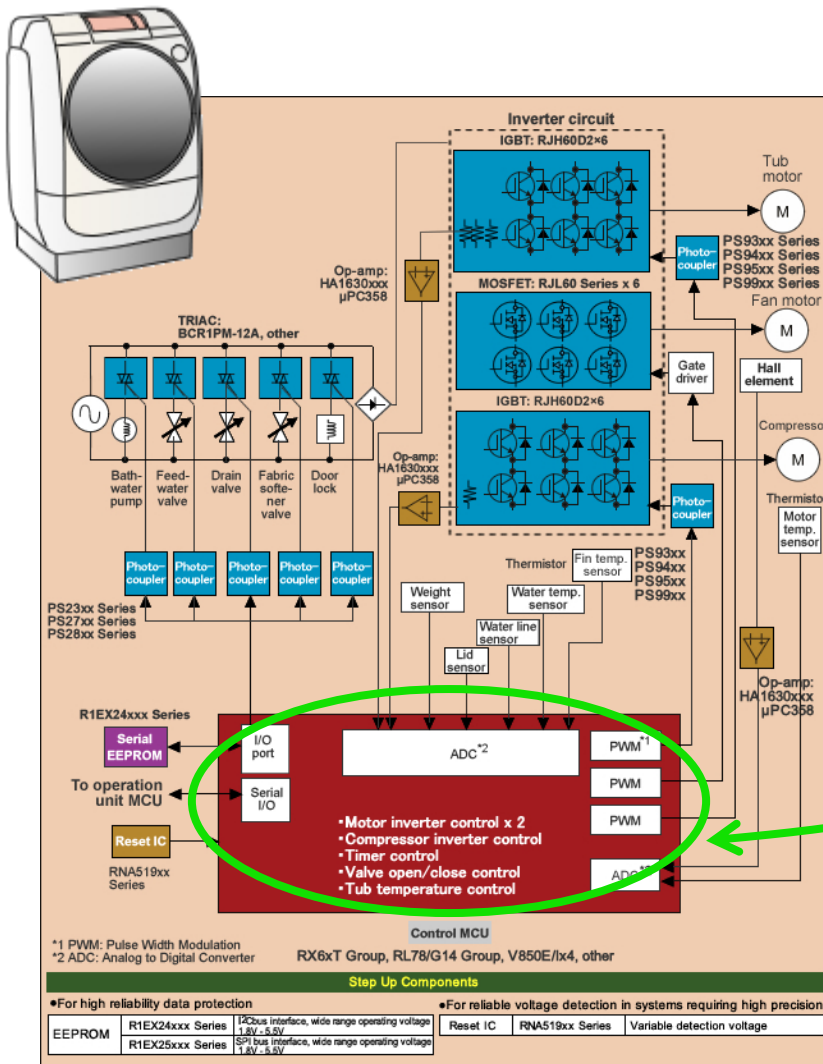
Salidas analógicas

Comunicaciones bajo protocolo



<http://asic-soc.blogspot.com.es/2007/12/embedded-system-for-automatic-washing.html>

Sistemas embebidos: ejemplos



Microcontroladores

En vez de hacer un sistema centralizado, donde una sola MCU controla todo, se tiende a hacer **sistemas distribuidos**, donde cada MCU controla una parte y se comunican entre ellas para coordinarse (al final, la idea es remplazar la lógica de control mediante micros fácilmente programables. Si usásemos una sola MCU, el control se complicaría)

http://am.renesas.com/applications/appliances/washing_machine_high/index.jsp

Sistema embebido: ¿qué necesitamos?

Por parte del microcontrolador

Sensores y actuadores

- Electrónica de acondicionamiento para leer los sensores (ej: amplificadores...)
- Electrónica de acondicionamiento para operar los actuadores (ej: drivers...)

Dispositivos E/S

- Leer el estado de señales digitales/discretas (ej: teclados, sensores...)
- Leer el valor de señales analógicas/continuas (ej: sensores...)
- Actuar sobre señales digitales (ej: luces, calefactores, válvulas...)
- Actuar sobre elementos “analógicos” (ej: motores...)
- Comunicarnos con otros dispositivos (ej: pantallas, relojes/calendarios...)
- Medir el paso del tiempo y actuar en consecuencia (ej: temporizadores...)
- Atender a eventos críticos en tiempo predecible (ej: interrupciones...)

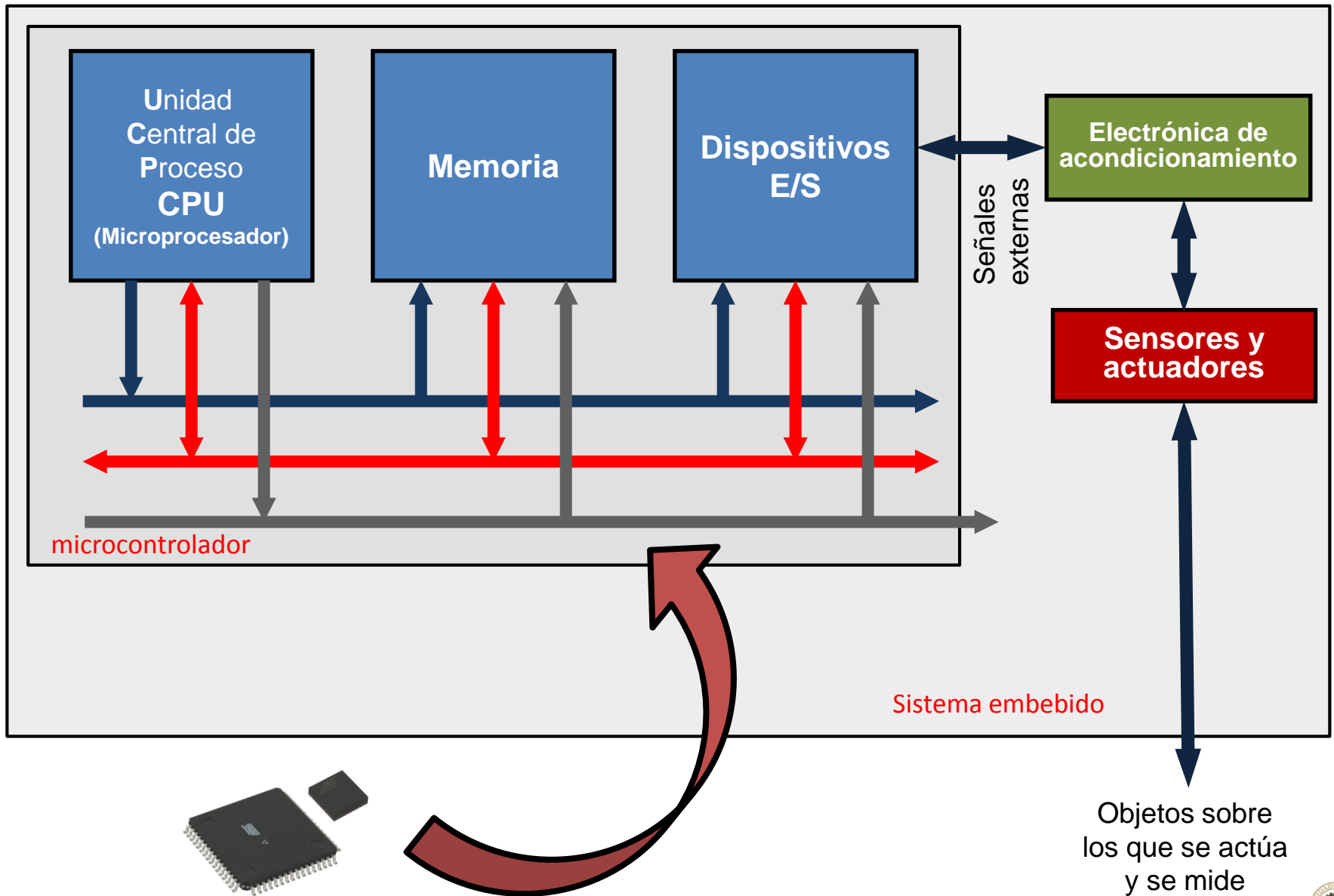
Memoria

- Almacenar datos de operación (RAM) y “settings” (FLASH/EEPROM)
- Almacenar programas (RAM ó ROM/FLASH)

Microprocesador

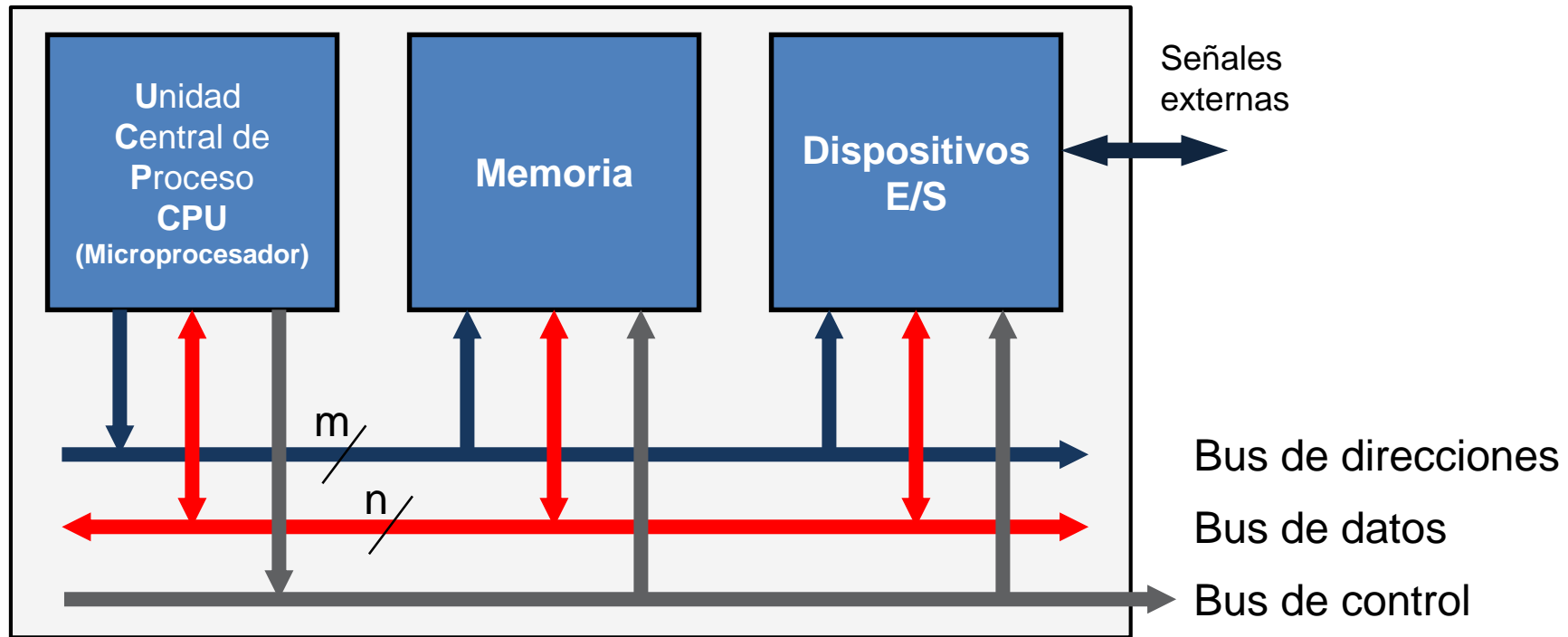
- Ejecutar los programas almacenados en memoria
- Operar con los datos (operaciones aritméticas y lógicas)
- Controlar (leer y escribir sobre) los dispositivos de E/S para que actúen

Sistema embebido: elementos



microcontrolador

Microcontrolador y buses



m bits **direcciones** : 2^m posiciones/direcciones de memoria distintas } Capacidad : $2^m \times n$ bits
 n bits de **datos** : n bits transmitidos en paralelo
 varias señales de **control** (*reloj, reset, read/write, interrupción, etc.*)

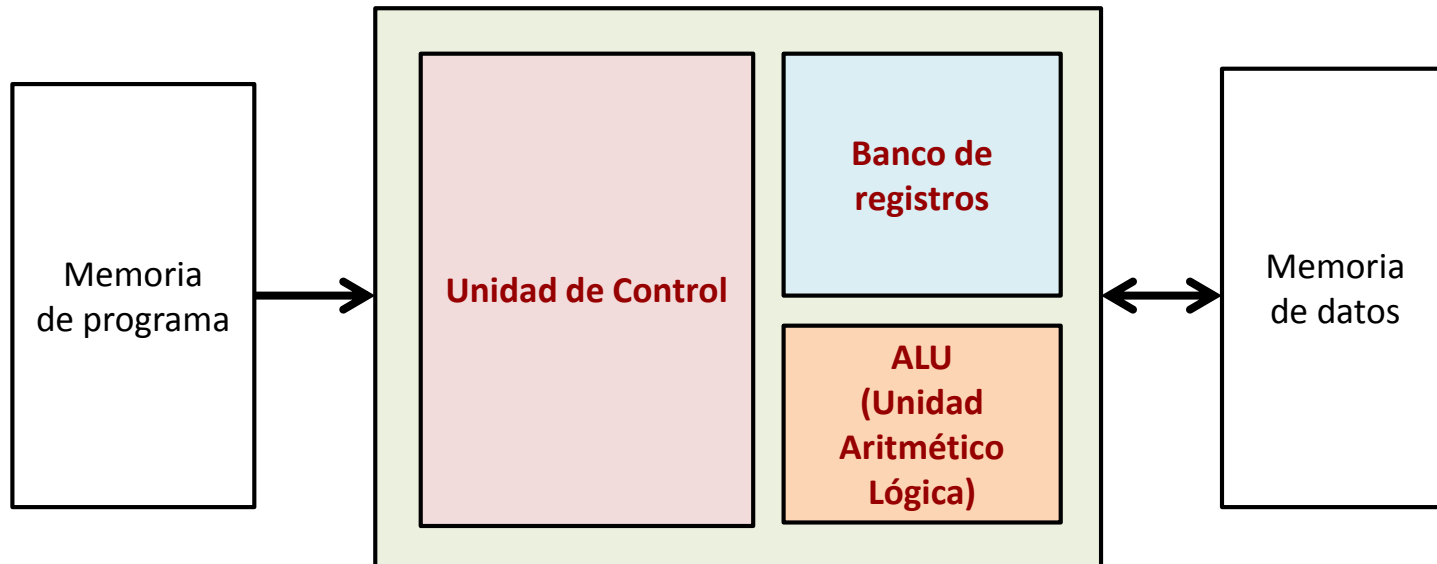
Transferencias de datos: *Escritura*: uP → Memoria *Entrada*: E/S → uP
 Lectura : Memoria → uP *Salida*: uP → E/S

(todas con el micro)

- El tamaño del bus de datos coincide normalmente con el tamaño de la palabra con la que el uP puede trabajar y hacer operaciones. Así, hablamos de micros de 8, 16 o 32 bits. Es la unidad mínima de información que se intercambia entre el uP, la memoria y/o los dispositivos de E/S.

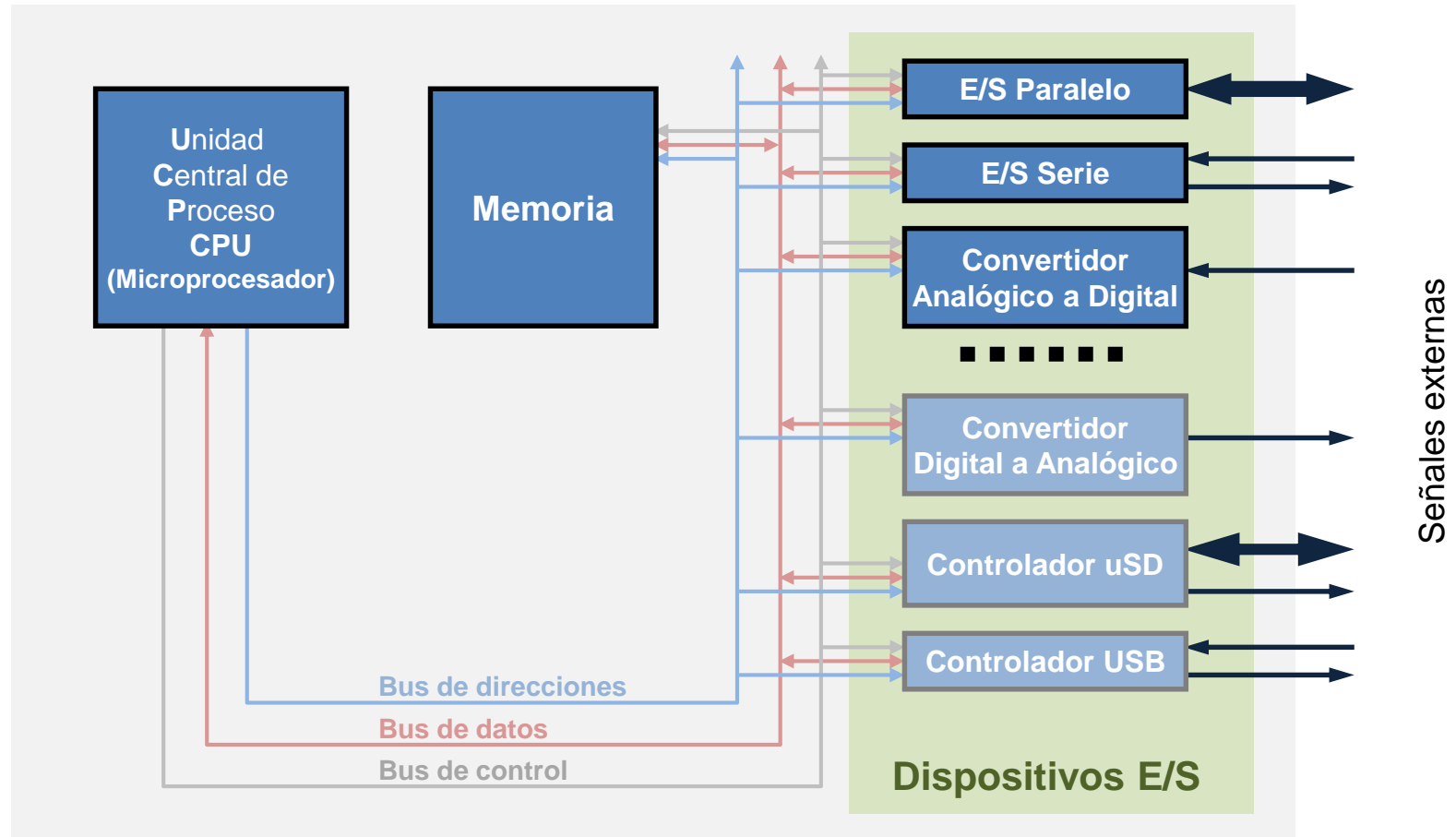
Unidad Central de Proceso (CPU)

- Encargada de la ejecución de todas las operaciones y movimiento de datos entre el microprocesador, la memoria y los dispositivos de E/S.
Típicamente se divide en:
 - *Banco de registros*: almacena datos y resultados de operaciones con la ALU
 - *Unidad Aritmético Lógica (ALU)*: Realiza operaciones entre registros (+, -, and, or...)
 - *Unidad de control*: Secuencia las acciones en función de la instrucción a ejecutar



Dispositivos de E/S

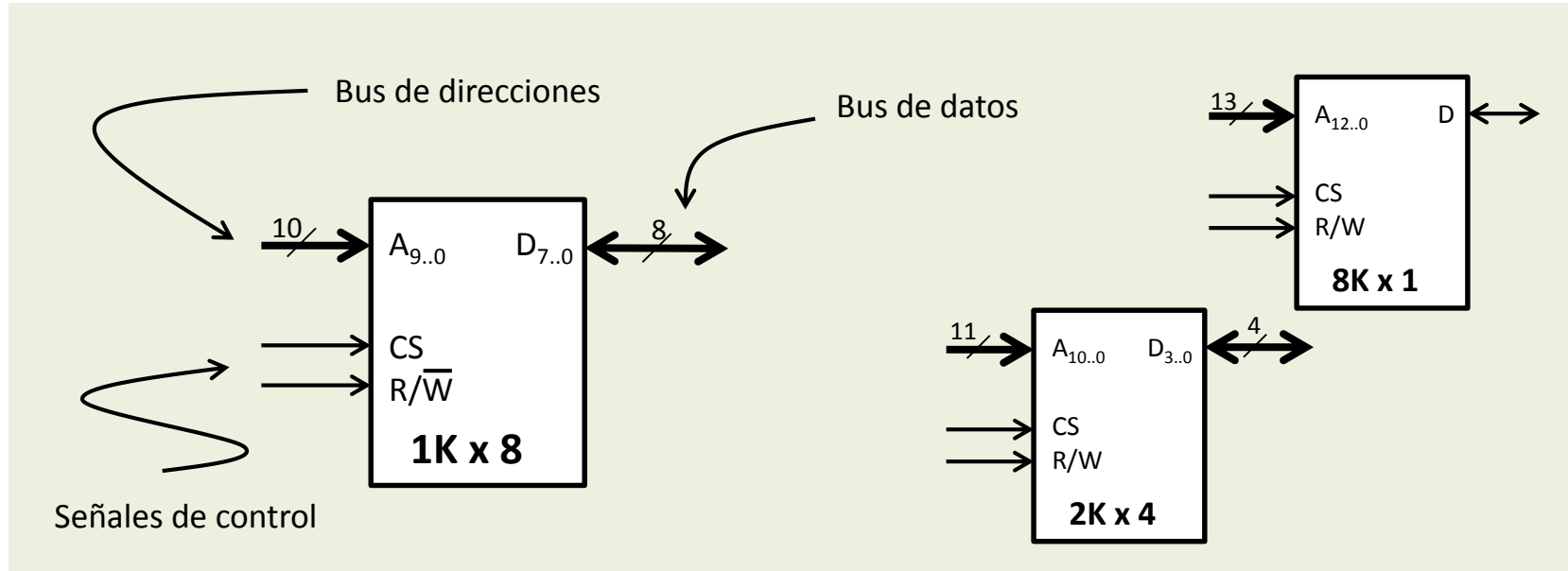
- Circuitería que permite la interacción entre el microprocesador y las señales externas, bien sea para leer el valor de estas señales, para modificarlas o para que operen de una determinada manera.



Señales externas

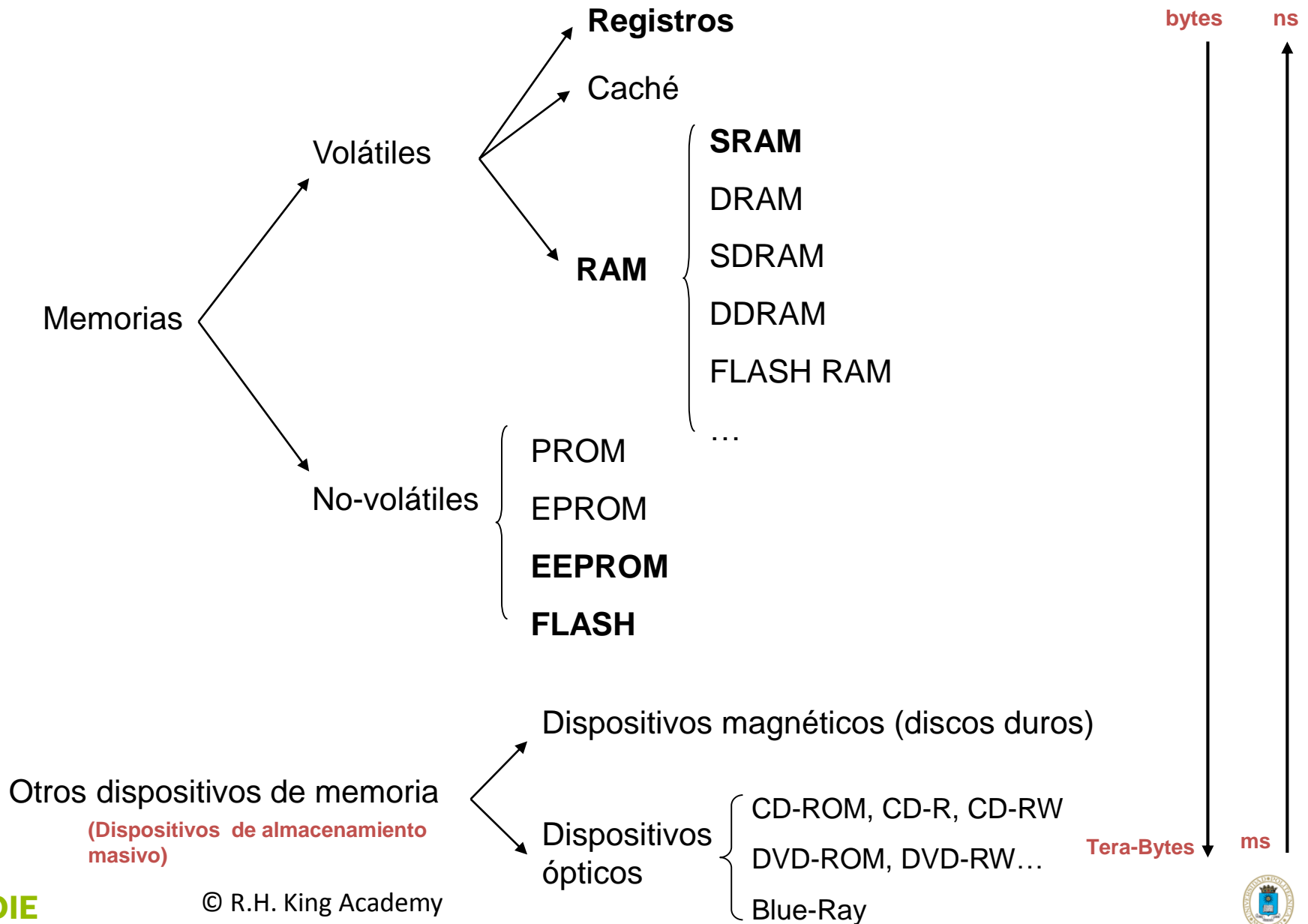
Memorias

- Son circuitos integrados de almacenamiento



- Se estructuran normalmente en 2^m (m = ancho del bus de direcciones) posiciones de memoria de n (n = ancho del bus de datos) bits cada una
- Diseño interno optimizado para almacenar muchos bits a bajo coste (los elementos de memoria no son biestables), por eso son bastante lentas comparadas con otros circuitos digitales. La estructura interna de cada celda de memoria depende del tipo de memoria y de su tecnología de fabricación.

Memorias y dispositivos de almacenamiento



Firmware

- Denominamos *firmware* al programa que está cargado en la MCU y que se encarga del control del sistema.

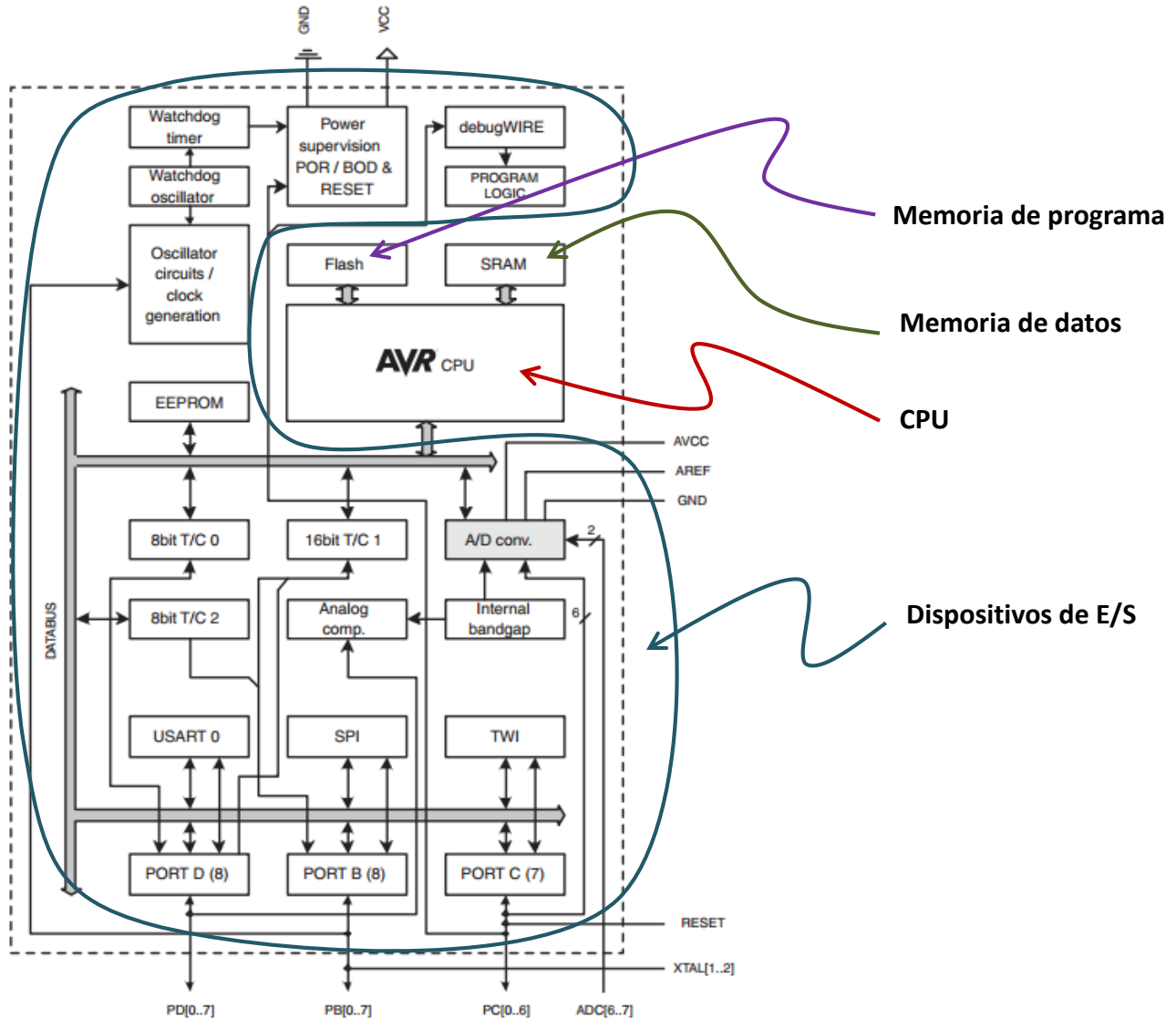
```
int main() {  
    while( true ) {  
        char key = read_keyboard();  
  
        if (key = 'C') {  
            do_cold_washig();  
        }  
  
        if (key = 'T') {  
            char temp = read_temp_encoder();  
            do_hot_washig( temp );  
        }  
    }  
}
```

```
int do_cold_washing() {  
    // 100 washing cycles  
    for( int i = 0 ; i < 100 ; i++ ) {  
        do_spin_right(10); // 10 seconds  
        do_wait(10);  
        do_spin_left(10);  
    }  
  
    // Drying (fast spin) during 3 minutes  
    initialize_time_lapse();  
    do_fast_spin();  
    while ( time_lapsed() < 180 ) { } ;  
    stop_motors();  
}
```

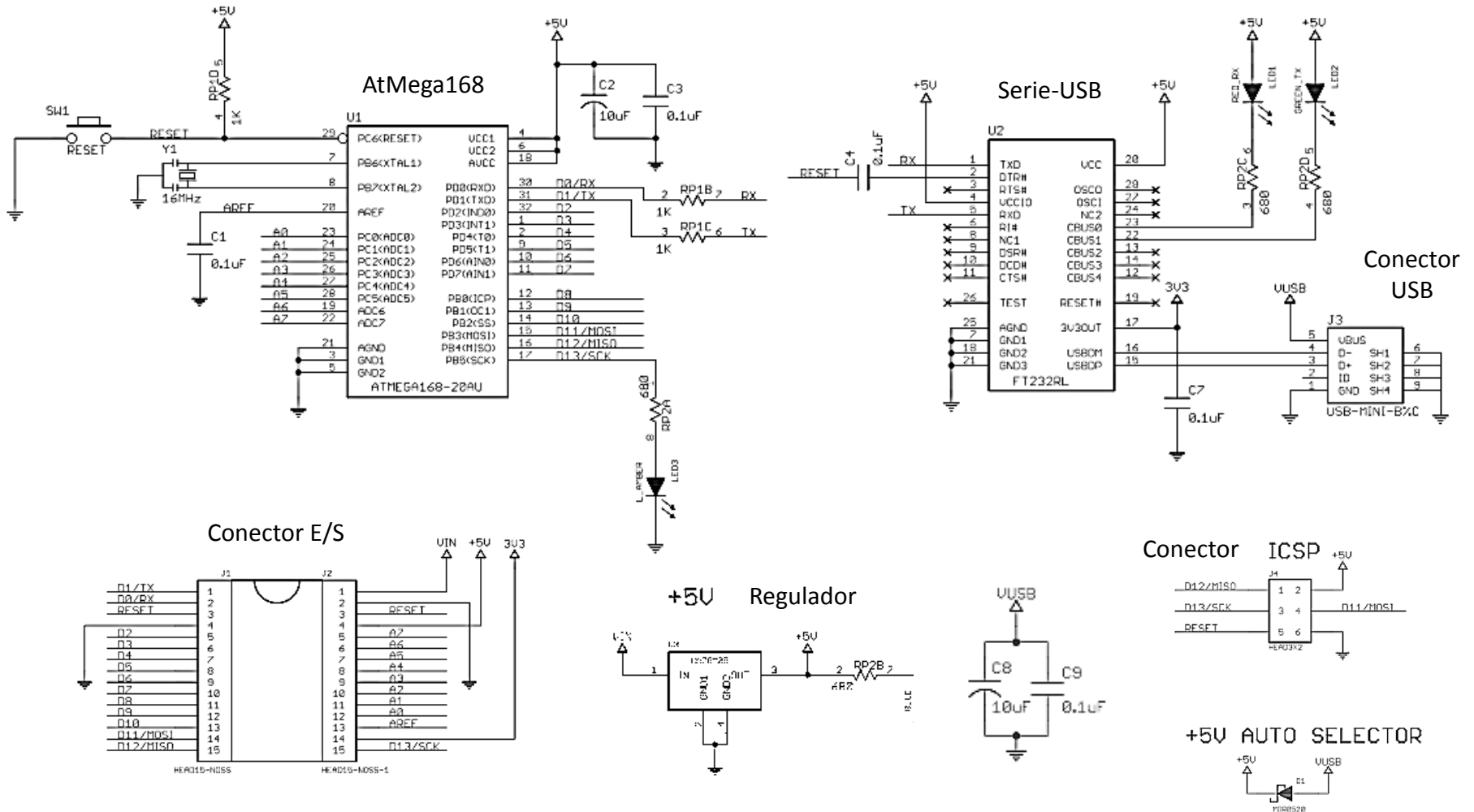
```
int stop_motors() {  
    // Motor driver is in bit 3 of PORTB  
    clrBit(PORTB, 3); // set pin to 0  
}
```

- En un sistema embebido, el programa está siempre activo, ya que el sistema debe estar preparado para responder a su entorno en cualquier momento y sin intervención (sin tener que cargar un programa, por ejemplo)
- En la mayoría de los sistemas empotrados la medición del tiempo juega un papel fundamental, ya que el sistema suele interactuar con elementos que necesitan tiempo para reaccionar y funcionalidades que dependen del tiempo.

Ejemplo de microcontrolador: el AVR AtMega168/328



Ejemplo de sistema microprocesador: Arduino (I)



Ejemplo de sistema microprocesador: Arduino (II)

- La versión utilizada para las prácticas y trabajos

