



**PAC**- Performance-centered Adaptive Curriculum for Employment Needs  
Programa ERASMUS: Acción Multilateral - 517742-LLP-1-2011-1-BG-ERASMUS-ECUE

# **MASTER DEGREE:**

## **Industrial Systems Engineering**

**ASIGNATURA ISE5:**  
**Controladores industriales de diseño de alto nivel**

**MÓDULO 3:**  
**Sistemas Embebidos (ES)**

**TAREA 3-1:**  
**INGENIERÍA DE LOS SISTEMAS EMBEBIDOS (HARDWARE -HW)**



Departamento de Ingeniería Eléctrica, Electrónica y de Control  
<http://www.ieec.uned.es/>



## Contenido

TAREA 3-1: INGENIERÍA DE LOS SISTEMAS EMBEBIDOS (HARDWARE – HR)	3
1. INTRODUCCIÓN Y OBJETIVOS	3
2. CONTENIDO	3
2.1 APARTADO 1: ¿Cuáles son las características básicas de los sistemas embebidos (ES)?	4
2.2 APARTADO 2: Análisis de las características típicas de la CPU y la memoria utilizada en ES	8
2.3 APARTADO 3: Tipos de interfaces de la memoria	12
2.4 APARTADO 4: Comparar las principales características de la MCU de las series PIC, AVR y ARM	15
3. CONCLUSIONES	18
4. BIBLIOGRAFÍA Y/O REFERENCIAS	19
5. ENLACES DE INTERÉS	19

## Índice de figuras

Figura 1: Detalle de MP configuración mínima	9
Figura 2: Conexión de los buses entre el MP, RAM y dispositivos de E/S (I/O)	13
Figura 3: Diagrama de bloque de un AVR	16

## Índice de tablas

## TAREA 3-1: INGENIERÍA DE LOS SISTEMAS EMBEBIDOS (HARDWARE – HR)

### 1. INTRODUCCIÓN Y OBJETIVOS

En esta tarea se trata de analizar qué son los sistemas embebidos, cuáles son sus características básicas, así como las interfaces. La comunicación adquiere gran importancia en los sistemas embebidos. Lo habitual es que el sistema pueda comunicarse mediante interfaces estándar de cable o inalámbricas.

Los componentes de un sistema embebido, son los siguientes, en la parte central se encuentra el microprocesador, microcontrolador, DSP, etc. La CPU o unidad que aporta la capacidad de cómputo del sistema, pudiendo incluir memoria interna o externa, un micro con arquitectura específica según los requisitos.

Se analizará la estructura de un sistema de Microprocesador, los componentes del MP. Memorias, puertos de entrada/salida (I/O), buses de datos, de direcciones y de control, así como la conexión de los buses entre el MP, RAM y E/S (I/O) dispositivos.

Se estudiarán los diferentes tipos de arquitectura (Von Neumann, Harvard, Super Harvard), así como los microcontroladores tipo megaAVR (serie ATmega).

Finalmente, a modo de ejemplo se expondrán algunos sistemas embebidos utilizados en la actualidad en diferentes campos.

### 2. CONTENIDO

A continuación, se realizará una introducción del temario a desarrollar en dicha actividad. En primer lugar, analizaremos la definición de **sistema embebido**:

Se trata de un sistema de computación diseñado para realizar una o algunas funciones dedicadas frecuentemente en un sistema de computación en tiempo real. Al contrario de lo que ocurre con los ordenadores de propósito general (como por ejemplo una computadora personal o PC) que están diseñados para cubrir un amplio rango de necesidades, los sistemas embebidos se diseñan para cubrir necesidades específicas. En un sistema

embebido la mayoría de los componentes se encuentran incluidos en la placa base (la tarjeta de vídeo, audio, módem, etc.) y muchas veces los dispositivos resultantes no tienen el aspecto de lo que se suele asociar a una computadora.

Los sistemas embebidos se pueden programar directamente en el lenguaje ensamblador del microcontrolador o microprocesador incorporado sobre el mismo, o también, utilizando los compiladores específicos, pueden utilizarse lenguajes como C o C++; en algunos casos, cuando el tiempo de respuesta de la aplicación no es un factor crítico, también pueden usarse lenguajes como JAVA

A continuación analizaremos los conceptos relacionados en el documento ppt de la Tarea 3.1-Specific Training, así como las siguientes cuestiones:

- ¿Cuáles son las características básicas de los sistemas embebidos (ES)?
- ¿Cuáles son las características típicas de la CPU y la memoria utilizada en ES?
- Comparar las principales características de la MCU de la serie PIC, AVR y ARM?

## 2.1 APARTADO 1: ¿Cuáles son las características básicas de los sistemas embebidos (ES)?

Las características básicas de los sistemas embebidos son las siguientes:

- Deben ser confiables,
- La confiabilidad, en inglés reliability  $R(t)$ , es la probabilidad de que el sistema trabaje correctamente dado que está funcionando en  $t=0$ .
- La mantenibilidad, en inglés Maintainability  $M(d)$ , es la probabilidad de que el sistema vuelva a trabajar correctamente  $d$  unidades de tiempo después de un fallo.
- La disponibilidad, en inglés Availability  $A(t)$ , es la probabilidad de que el sistema esté funcionando en el tiempo  $t$ .
- La seguridad informática: consiste en disponer de una comunicación confidencial y autenticada.
- La creación de un sistema confiable debe ser considerada desde un comienzo, no como una consideración posterior.
- Deben ser eficientes en cuanto a la energía, al tamaño de código, al peso y al costo.
- Están dedicados a ciertas aplicaciones.
- Interfaces de usuario dedicadas (sin ratón, keyboard y pantalla)

Muchos sistemas embebidos deben cumplir restricciones de tiempo real. Un sistema de tiempo real debe reaccionar a estímulos del objeto controlado (u operador) dentro de un intervalo definido por el ambiente.

Hay que tener en cuenta que respuestas correctas pero tardías son erróneas. Una restricción de tiempo real se dice DURA ó ESTRICTA (hard) si su incumplimiento puede resultar una catástrofe.

Toda otra restricción de tiempo es blanda (soft).

La mayoría de los sistemas embebidos son de tiempo real (Real-Time) y la mayoría de los sistemas de tiempo real son embebidos.

Se encuentran frecuentemente conectados a ambientes físicos a través de sensores y actuadores. Son sistemas híbridos (es decir, poseen partes analógicas + digitales), típicamente son sistemas reactivos, los cuales son “aquellos que están en interacción continua con su entorno y su ejecución es a un ritmo determinado por ese entorno” (Bergé, 1995).

Podemos añadir también que los sistemas embebidos poseen un número limitado de funciones predefinidas para actuar, tienen una fuente de alimentación limitada y una administración de energía efectiva.

Poseen disponibilidad de recursos de reservas para situaciones inesperadas.

Los sistemas embebidos suelen tener en una de sus partes una computadora con características especiales conocida como microcontrolador que viene a ser el cerebro del sistema, el cual incluye interfaces de entrada/salida en el mismo chip. Normalmente estos sistemas poseen un interfaz externo para efectuar un monitoreo del estado y hacer un diagnóstico del sistema.

Además cabe reseñar que el uso de sistemas embebidos en productos complejos implica un desafío de la seguridad en TI para proteger la información contenida en el sistema embebido y también la que es transmitida desde y hacia el dispositivo por redes privadas o Internet. Por tanto cabe incluir funciones criptográficas, diseño de protocolos y consultoría en análisis y verificación así como servicios de pruebas de seguridad, así como evaluaciones específicas para sistemas embebidos.

El diseño de un producto que incorpora sistemas embebidos está orientado a minimizar los costos y maximizar la confiabilidad, pero también es imprescindible incorporar en el diseño consideraciones de seguridad, incluyendo funciones y protocolos criptográficos que protejan la información durante todas las fases. Los sistemas embebidos a menudo operan en un ambiente dedicado con condiciones operacionales y escenarios muy específicos. Es importante que dichas condiciones y amenazas se tengan en cuenta cuando se diseñan las funciones de seguridad.

Tradicionalmente esto ha sido realizado a través de una terminal serie, pero con el tiempo la industria ha observado las ventajas del monitoreo a distancia,

así como también la posibilidad de efectuar pequeños ajustes sin necesidad de estar físicamente en el mismo lugar donde surten efecto dichos cambios.

Existen varias interfaces:

- Las interfaces de operador (Hombre-Máquina-HMI) - monitores, interruptores, botones, indicadores, emisores individuales o grupales de los diferentes tipos de señales, motores eléctricos, solenoides y otros. Se puede aplicar en los trenes. Las características del software son las siguientes: robustez, facilidad de uso, presentación clara de la información, diseño atractivo, flexibilidad de proyecto.
- Las interfaces eléctricas (interfaces con otros componentes y dispositivos): Interno - I2C, SPI, ISA y otros.
- Las interfaces Exteriores - RS232, TTY, Ethernet, Centronics, FlexRay, CAN, LIN, RF y otros.

### **Estructura**

Las principales características de un sistema embebido son el bajo costo y consumo de potencia. Dado que muchos sistemas embebidos son concebidos para ser producidos en miles o millones de unidades, el costo por unidad es un aspecto importante a tener en cuenta en la etapa de diseño. Normalmente, los sistemas embebidos emplean procesadores muy básicos, relativamente lentos y memorias pequeñas para minimizar los costos.

La velocidad no solo está dada por la velocidad del reloj del procesador, sino que la totalidad de la arquitectura se simplifica para reducir costos. Usualmente un ES (sistema embebido) utiliza periféricos controlados por interfaces seriales sincrónicas, las cuales son muchas veces más lentas que los periféricos de un PC.

Un ES debe afrontar fuertes restricciones de recursos, por tanto normalmente deberá hacer uso de sistemas operativos especiales, denominados de tiempo real (RTOS Real time operating system), tal y como se mencionó en el apartado anterior en las características de un ES. Los sistemas embebidos deberán reaccionar a estímulos provenientes del entorno, respondiendo con fuertes restricciones de tiempo en muchos casos, por lo tanto, un sistema se dice que trabaja en tiempo real si la información después de la adquisición y tratamiento es todavía vigente. Es decir, que en el caso de una información que llega de forma periódica, los tiempos de adquisición y tratamiento deben ser inferiores al período de actualización de dicha información.

Los programas en estos sistemas se ejecutan minimizando los tiempos muertos y afrontando fuertes limitaciones de hardware, ya que usualmente no tienen discos duros, ni teclados o monitores, una memoria flash reemplaza los discos

y algunos botones y una pantalla LCD normalmente reemplazan los dispositivos de interfaz.

El software que controla un dispositivo de hardware, se conoce como Firmware. La programación en estos dispositivos se realiza en lenguaje ensamblador o en lenguaje C, actualmente se han desarrollado algunas máquinas virtuales y otros compiladores que permiten el diseño de programas más complejos.

Se puede encontrar depuradores, simuladores, bases de datos entre otras herramientas para el diseño y programación de este tipo de sistemas.

### **Componentes de un sistema embebido**

Un ES estaría formado por un microprocesador y un software que se ejecute sobre éste. Sin embargo este software necesitará sin duda un lugar donde poder guardarse para luego ser ejecutado por el procesador. Esto podría tomar la forma de memoria RAM o ROM. Todo sistema embebido necesitará una cierta cantidad de memoria, la cual puede incluso encontrarse dentro del mismo chip del procesador. También contará con una serie de salidas y entradas necesarias para comunicarse con el mundo exterior.

Debido a que las tareas realizadas por sistemas embebidos son de relativa sencillez, los procesadores comúnmente empleados cuentan con registros de 8 o 16 bits.

En su memoria sólo reside el programa destinado a gobernar una aplicación concreta. Sus líneas de entrada/salida (I/O) soportan el conexionado de los sensores y actuadores del dispositivo a controlar y todos los recursos complementarios disponibles tienen como finalidad atender a sus requerimientos.

Estas son las únicas características que tienen en común los sistemas embebidos, todo lo demás será totalmente diferente para cada sistema embebido en particular debido a la variedad de aplicaciones disponibles.

### **Plataforma de sistemas integrados:**

- El Microprocesador (MP o  $\mu$ P) y los microcontroladores (MCU), que tienen menos poder de cómputo, pero varios periféricos;
- Arquitecturas de base - Von Neumann and Harvard;
- Utilizado  $\mu$ P y MCU - CISC (Complex Instruction Set Computer) y más a menudo RISC (Reduced Instruction Set Computer);
- Popular RISC familias de procesos: ARC (ARC International), ARM (ARM Holdings), AVR (Atmel), PIC (Microchip), MSP430 (TI) y otros;

- CISC CPUs: Intel y Motorola;
- Por lo general en el interior hay una memoria caché y procesamiento canalización de instrucciones;
- Memoria para datos e instrucciones: RAM, PROM - OTP (One-Time Programmable), EEPROM o memoria Flash;
- Periféricos: General Purpose Input / Output - GPIO, temporizadores, ADC, DAC y mucho más.
- **Comunicación:**
  - RS-232, RS-422, RS-485, UART / USART (Receptor Universal Sincrono y Asíncrono / Transmisor);
  - I2C (Circuito Inter-Integrado), SPI (Bus de Interface periférico en serie), SSC y ESSI (Interfaz mejorada serie síncrona), USB (Universal Serial Bus);
  - Protocolos de comunicación de red: Ethernet, CAN (Controlador del área de red), LonWorks etc.
  - Software: Popular OS – QNX4 RIOS, Linux embebido y Linux-base (Android, etc.), iOS, Windows CE, etc.
- **Herramientas para probar y corregir (Depuración)**
  - JTAG (Joint Test Action Group) – una interfaz especializada para la prueba saturada PCB;
  - ISP (In-System Programming) – Programación de Circuito;
  - ICSP (circuito de programación en serie) - un método para la programación directa del microcontrolador, por ejemplo, de la serie PIC and AVR;
  - BDM (Modo de depuración de fondo) – utilizado principalmente en productos de Freescale;
  - IDE (Entorno de desarrollo integrado) – para el desarrollo de programas.

## 2.2 APARTADO 2: Análisis de las características típicas de la CPU y la memoria utilizada en ES

Un MCU es un **microprocesador** de un solo chip, que se utiliza para controlar varios dispositivos.

A diferencia de los microprocesadores de propósito general en los PCs, el MC se orienta a un número limitado de tareas y muchas veces a una sola tarea. El MC típico generalmente contiene toda la memoria y la periferia necesaria para una aplicación dada:

- El procesador central – por lo general pequeño y simplificado;



- Generador de señal de reloj – más a menudo un oscilador, basado en un resonador de cuarzo o un circuito RC;
- RAM para el almacenamiento de datos;
- ROM o variaciones (PROM, EPROM, FLASH) para el programa;
- Interfaces de entrada/salida – puertos en serie y paralelos;
- Otros dispositivos periféricos, tales como temporizadores, contadores, ADC, etc.

Un PC embebido posee una arquitectura semejante a la de un PC. Brevemente describiremos los elementos básicos:

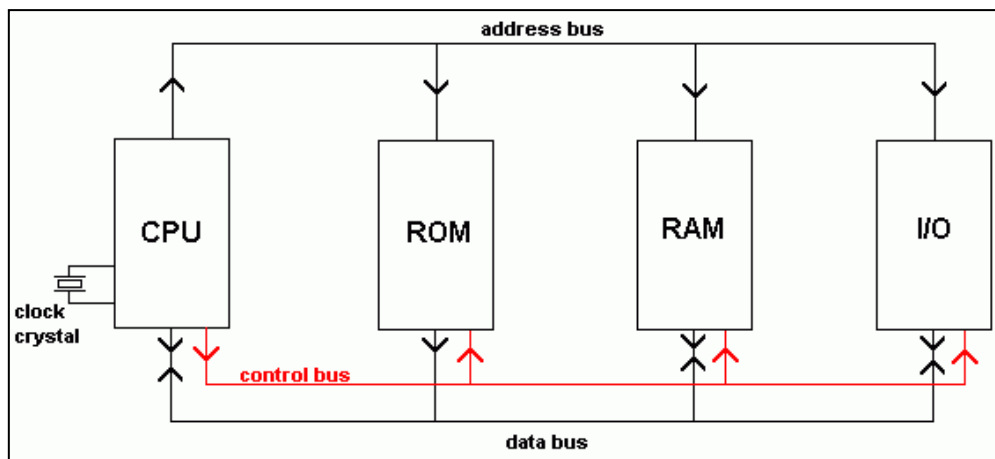


Figura 1: Detalle de MP configuración mínima

## Microprocesador

Es el encargado de realizar las operaciones de cálculo principales del sistema. Ejecuta el código para realizar una determinada tarea y dirige el funcionamiento de los demás elementos que le rodean. Un microprocesador es una implementación en forma de circuito integrado (IC) de la Unidad Central de Proceso CPU de una computadora. Frecuentemente nos referimos a un microprocesador como simplemente “CPU”, y la parte de un sistema que contiene al microprocesador se denomina subsistema de CPU. Los subsistemas de entrada/salida y memoria pueden ser combinados con un subsistema de CPU para formar una computadora o sistema embebido completo. Estos subsistemas se interconectan mediante los buses de sistema (formados a su vez por el bus de control, el bus de direcciones y el bus de datos).

## Memoria

En ella se encuentra almacenado el código de los programas que el sistema puede ejecutar así como los datos. Su característica principal es que debe tener un acceso de lectura y escritura lo más rápido posible para que el

microprocesador no pierda tiempo en tareas que no son meramente de cálculo. Al ser volátil el sistema requiere de un soporte donde se almacenen los datos aun sin disponer de alimentación o energía.

### **Caché**

Es una Memoria más rápida que la principal en la que se almacenan los datos y el código al que se ha accedido últimamente. Dado que el sistema realiza microtareas, muchas veces repetitivas, la caché hace ahorrar tiempo ya que no hará falta ir a memoria principal si el dato o la instrucción ya se encuentra en la caché. Tiene un tamaño muy inferior (8 – 512 KB) con respecto a la principal (8 – 256 MB).

### **Disco duro**

En él la información no es volátil y además puede conseguir capacidades muy elevadas.

A diferencia de la memoria que es de estado sólido éste suele ser magnético. Pero su excesivo tamaño a veces lo hace inviable para PCs embebidos, con lo que se requieren soluciones como discos de estado sólido. Existen en el mercado varias soluciones de esta clase (DiskOnChip, CompactFlash, IDE Flash Drive, etc.) con capacidades suficientes para la mayoría de sistemas embebidos (desde 2 hasta más de 1 GB). El controlador del disco duro de PCs estándar cumple con el estándar IDE y es un chip más de la placa madre.

### **Disco flexible**

Su función es la de un disco duro pero con discos con capacidades mucho más pequeñas y la ventaja de su portabilidad. Siempre se encuentra en un PC estándar pero no así en un PC embebido.

### **BIOS-ROM**

BIOS (Basic Input & Output System, sistema básico de entrada y salida) es el código que es necesario para inicializar el ordenador y para poner en comunicación los distintos elementos de la placa madre. La ROM (Read Only Memory, memoria de sólo lectura no volátil) es un chip donde se encuentra el código BIOS.

### **CMOS-RAM**

Es un chip de memoria de lectura y escritura alimentado con una pila donde se almacena el tipo y ubicación de los dispositivos conectados a la placa madre (disco duro, puertos de entrada y salida, etc.). Además contiene un reloj en permanente funcionamiento que ofrece al sistema la fecha y la hora.

### **Chip Set**

Chip que se encarga de controlar las interrupciones dirigidas al microprocesador, el acceso directo a memoria (DMA) y al bus ISA, además de ofrecer temporizadores, etc.

Es usual encontrar la CMOS-RAM y el reloj de tiempo real en el interior del Chip Set.

### **Puertos de Entrada / Salida (I/O)**

Son puntos (nodos) en los que los dispositivos periféricos se pueden conectar y pueden intercambiar información con la memoria y el procesador central.

Los puertos contienen en sí mismos un número definido de registros, los cuales se utilizan para el almacenamiento temporal de varios tipos de datos. Las direcciones de los registros y sus funciones están definidas con precisión (standard). El subsistema de entrada acepta datos del exterior para ser procesados mientras que el subsistema de salida transfiere los resultados hacia el exterior. Lo más habitual es que haya varios subsistemas de entrada y varios de salida. A estos subsistemas se les reconoce habitualmente como periféricos de E/S.

Tipos:

- puertos serie**, en los que los datos se transfieren bit a bit de forma secuencial (COM1, COM2);
- puertos paralelos**, en los que los datos se transfieren en paralelo (en un byte, una palabra, etc.) (LPT1).
- puertos universales** (USB).

El subsistema de memoria almacena las instrucciones que controlan el funcionamiento del sistema. Estas instrucciones comprenden el programa que ejecuta el sistema. La memoria también almacena varios tipos de datos: datos de entrada que aún no han sido procesados, resultados intermedios del procesado y resultados finales en espera de salida al exterior.

Es importante darse cuenta de que los subsistemas estructuran a un sistema según funcionalidades. La subdivisión física de un sistema, en términos de circuitos integrados o placas de circuito impreso (PCB) puede y es normalmente diferente. Un solo circuito integrado (IC) puede proporcionar múltiples funciones, tales como memoria y entrada/salida.

Un microcontrolador (MCU) es un IC que incluye una CPU, memoria y circuitos de E/S. Entre los subsistemas de E/S que incluyen los microcontroladores se encuentran los temporizadores, los convertidores analógico a digital (ADC) y digital a analógico (DAC) y los canales de comunicaciones serie. Estos

subsistemas de E/S se suelen optimizar para aplicaciones específicas (por ejemplo audio, video, procesos industriales, comunicaciones, etc.).

Un SE consiste en un sistema con microprocesador cuyo hardware y software están específicamente diseñados y optimizados para resolver un problema concreto eficientemente. Normalmente un SE interactúa continuamente con el entorno para vigilar o controlar algún proceso mediante una serie de sensores. Su hardware se diseña normalmente a nivel de chips, o de interconexión de PCB, buscando la mínima circuitería y el menor tamaño para una aplicación particular.

Un ES (sistema embebido) simple contará con un microprocesador, memoria, unos pocos periféricos de E/S y un programa dedicado a una aplicación concreta almacenado permanentemente en la memoria. El término embebido o empotrado hace referencia al hecho de que el microcomputador está encerrado o instalado dentro de un sistema mayor y su existencia como microcomputador puede no ser aparente.

Las microcomputadoras en estos sistemas controlan electrodomésticos tales como: televisores, videos, lavadoras, alarmas, teléfonos inalámbricos, etc. Incluso una PC tiene sistemas embebidos en el monitor, impresora, y periféricos en general, adicionales a la CPU de la propia PC. Un automóvil puede tener hasta un centenar de microprocesadores y microcontroladores que controlan cosas como la ignición, transmisión, dirección asistida, frenos antibloqueo (ABS), etc.

Los sistemas embebidos se caracterizan normalmente por la necesidad de dispositivos de E/S especiales. Cuando se opta por diseñar sistemas embebidos partiendo de una placa con microcomputador también es necesario comprar o diseñar placas de E/S adicionales para cumplir con los requisitos de la aplicación concreta.

### **2.3 APARTADO 3: Tipos de interfaces de la memoria**

A los Buses se les denomina también sistemas de líneas para la conexión interna y externa entre los dispositivos en un Sistema informático.

Dependiendo de los dispositivos que se conectan, se pueden distinguir: un bus de sistema (bus principal), buses internos para la conexión con la memoria RAM principal, la conexión con la memoria Caché, buses de entrada/salida I/O, etc.

Principales tipos de buses son los siguientes:

**1. Bus de direcciones:** está diseñado para enviar las direcciones, preparadas en el microprocesador, con el objetivo de elegir una celda definida de la memoria o un Puerto I/O (Entrada/salida). El bus de direcciones es de un solo sentido: las direcciones siempre son generadas por la MS

**2. Bus de datos:** A lo largo del bus de datos de intercambio de información (instrucciones o datos) se lleva a cabo entre el microprocesador y los dispositivos periféricos – se trata de un intercambio de dos vías. Se trata de operaciones de lectura y escritura.

**3. Bus de control:** es utilizado para el envío y la recepción de señales de control. Las señales de control aseguran la sincronización (control del tiempo) entre el MS y el resto de los componentes del Sistema

A continuación se detalla un esquema de la conexión de los buses entre el MP, RAM y dispositivos de E/S (I/O)

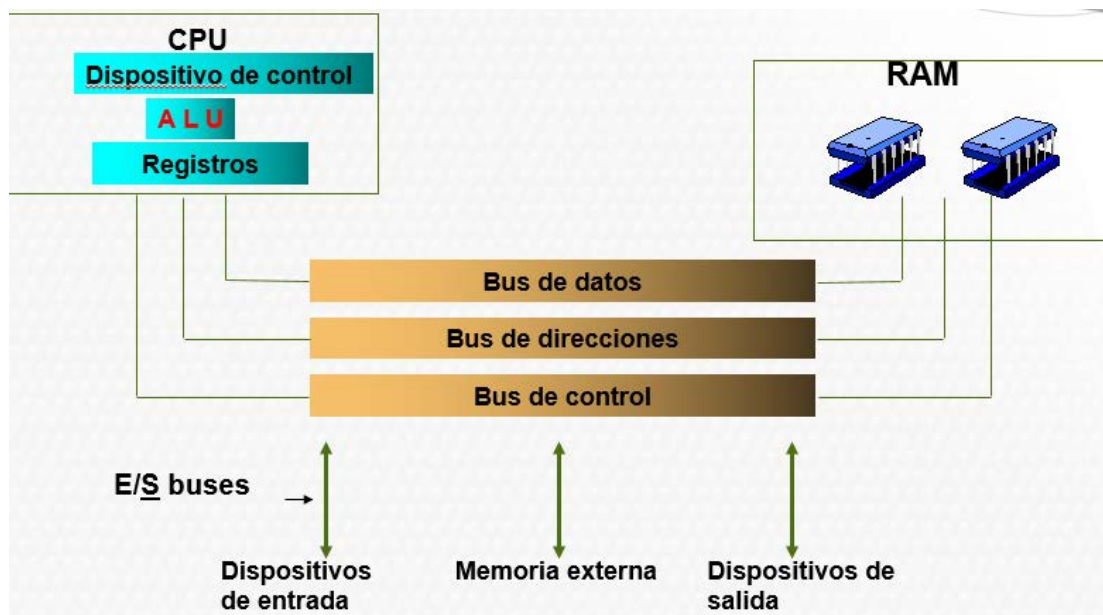


Figura 2: Conexión de los buses entre el MP, RAM y dispositivos de E/S (I/O)

Existen varios tipos de arquitectura:

- **Arquitectura Von Neumann ( Tipo Princeton)** – memoria común, bus de datos e instrucción.
- **Arquitectura Harvard** – memorias separadas, buses de datos e instrucciones
- **Arquitectura Super Harvard** – memorias separate + CACHE para instrucciones + controladores I/O (E/S).

La arquitectura Harvard es típica para microcontroladores y DSP. También se utiliza en la organización de la memoria CACHE en los sistemas MP más complejos.

La arquitectura Von Neumann utiliza el mismo dispositivo de almacenamiento tanto para las instrucciones como para los datos, siendo la que se utiliza en un ordenador personal porque permite ahorrar una buena cantidad de líneas de E/S, que son bastante costosas, sobre todo para aquellos sistemas donde el procesador se monta en algún tipo de zócalo alojado en una placa madre. También esta organización les ahorra a los diseñadores de placas madre una buena cantidad de problemas y reduce el costo de este tipo de sistemas.

En un ordenador personal, cuando se carga un programa en memoria, a éste se le asigna un espacio de direcciones de la memoria que se divide en segmentos, de los cuales típicamente tendremos los siguientes: código (programa), datos y pila. Es por ello que podemos hablar de la memoria como un todo, aunque existan distintos dispositivos físicos en el sistema (disco duro, memoria RAM, memoria flash, unidad de disco óptico...).

En el caso de los microcontroladores, existen dos tipos de memoria bien definidas: memoria de datos (típicamente algún tipo de SRAM) y memoria de programas (ROM, PROM, EEPROM, flash u de otro tipo no volátil). En este caso la organización es distinta a las del ordenador personal, porque hay circuitos distintos para cada memoria y normalmente no se utilizan los registros de segmentos, sino que la memoria está segregada y el acceso a cada tipo de memoria depende de las instrucciones del procesador.

A pesar de que en los sistemas integrados con arquitectura Von Neumann la memoria esté segregada, y existan diferencias con respecto a la definición tradicional de esta arquitectura; los buses para acceder a ambos tipos de memoria son los mismos, del procesador solamente salen el bus de datos, el de direcciones, y el de control. Como conclusión, la arquitectura no ha sido alterada, porque la forma en que se conecta la memoria al procesador sigue el mismo principio definido en la arquitectura básica.

Algunas familias de microcontroladores como la Intel 8051 y la Z80 implementan este tipo de arquitectura, fundamentalmente porque era la utilizada cuando aparecieron los primeros microcontroladores.

La arquitectura Harvard, es la utilizada en supercomputadoras, en los microcontroladores, y sistemas integrados en general. En este caso, además de la memoria, el procesador tiene los buses segregados, de modo que cada tipo de memoria tiene un bus de datos, uno de direcciones y uno de control.

La ventaja fundamental de esta arquitectura es que permite adecuar el tamaño de los buses a las características de cada tipo de memoria; además, el procesador puede acceder a cada una de ellas de forma simultánea, lo que se traduce en un aumento significativo de la velocidad de procesamiento. Típicamente los sistemas con esta arquitectura pueden ser dos veces más rápidos que sistemas similares con arquitectura Von Neumann.

La desventaja está en que consume muchas líneas de E/S del procesador; por lo que en sistemas donde el procesador está ubicado en su propio encapsulado, solo se utiliza en supercomputadoras. Sin embargo, en los microcontroladores y otros sistemas integrados, donde usualmente la memoria de datos y programas comparten el mismo encapsulado que el procesador, este inconveniente deja de ser un problema serio y es por ello que encontramos la arquitectura Harvard en la mayoría de los microcontroladores.

Por eso es importante recordar que un microcontrolador se puede configurar de diferentes maneras, siempre y cuando se respete el tamaño de memoria que este requiera para su correcto funcionamiento.

#### 2.4 APARTADO 4: Comparar las principales características de la MCU de las series PIC, AVR y ARM

A continuación se procederá a comparar las principales características de la MCU de las series PIC, AVR y ARM.

Los microcontroladores megaAVR:

- Ofrecen la más **extensa selección de dispositivos y opciones periféricas** y son adecuados para una gran variedad de aplicaciones. Contienen dispositivos con interfaces para LCD USB, CAN y LIN además de las interfaces standards SPI, UART y I2C.
- Son desarrollados para aplicaciones, requieren un gran código de programación y ofrecen una **memoria para grandes programas y volúmenes de datos**.
- Cuando el proyecto requiere algo más poderoso, **la solución es un MegaAVR**. El rendimiento aquí es de **1MIPS por 1MHz**.
- **Universalidad**: El MC de la familia megaAVR se utiliza para dispositivos de uso general, así como para los dispositivos periféricos especializados, tales como los controladores USB o LCD, CAN, LIN y controladores Power Stage, etc.
- **Bajo consumo**: El consumo de corriente **es inferior a 200 uA a 1.8V y 1MHz** y aproximadamente **100 nA** en modo **power-down**.

- **Alta integración:** Los MCs megaAVR ofrecen volúmenes diferentes de **memoria flash** para los programas, **SRAM**, embebido **EEPROM** para datos, un comparador analógico, un multicanal de 10-bit de ADC, la elección de un generador de señal de reloj precisa interno o externo y común de puertos de E/S. Los puertos de E/S con funciones alternativas funcionan como entradas para el ADC o salidas para PWM. Contienen un temporizador de guarda incorporado y detector Brown Out.
- Los MCs megaAVR contienen también **dispositivos analógicos adicionales embebidos** tales como un medidor de temperatura, comparador analógico rápido y amplificador analógico programable. Un sensor de temperatura interno permite la calibración fiable de los parámetros del Sistema.
- Los **microcontroladores ATmega** difieren principalmente en la cantidad de la memoria de programa y otros tipos de memoria, y el volumen de la memoria de programa pueden ser definidos por los **números después de ATmega**. Hay ciertas, aunque no grandes, diferencias en los módulos periféricos también.

A continuación se detalla el diagrama de bloque de un AVR:

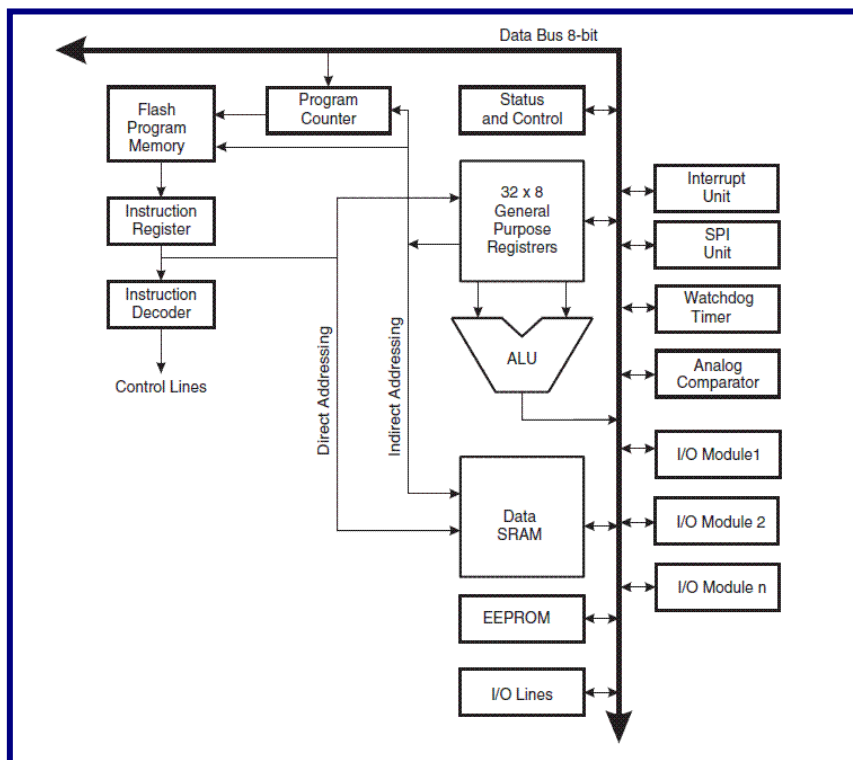


Figura 3: Diagrama de bloque de un AVR



Los Microprocesadores y microcontroladores ARM:

- **ARM** - es la abreviación del inglés **Advanced RISC Machine (Máquina Avanzada RISC)** - una de las más ampliamente utilizadas microprocesadores de 16/32-bit y microcontroladores en el mundo de los dispositivos de telefonía móvil.
- El desarrollo original de los núcleos ARM pertenece a la compañía **Acorn Computers Ltd.**, pero actualmente los microprocesadores y microcontroladores basados en ARM son producidos por muchas compañías: Alcatel, Atmel, NEC, NVIDIA, NXP (anteriormente Philips), Oki, Qualcomm, Samsung, Sharp, ST Microelectronics, Symbios Logic, Texas Instruments, VLSI Technology etc.
- Desde 2009, los procesadores ARM representan casi el **90% de todos los procesadores RISC embebidos de 32-bit** (electrónica de consumo, asistentes digitales personales - PDA, teléfonos móviles, reproductores iPod y otros medios de comunicación digitales y reproductores de audio, consolas de juegos, calculadoras, HDD, routers etc.).
- Familia ARM MP incluye **ARM7, ARM9, ARM11 y Cortex**. Las velocidades de reloj varían ampliamente - de decenas de MHz a 1GHz.
- **ARMs** son una muy buena combinación de la lógica moderna, de alta funcionalidad, bajo consumo (en su mayoría son CMOS), de bajo coste, la arquitectura es simplificada, lo que permite una fácil integración en varios dispositivos. [http://es.wikipedia.org/wiki/Arquitectura\\_ARM](http://es.wikipedia.org/wiki/Arquitectura_ARM).

#### Microcontroladores serie PIC

El PIC utiliza un juego de instrucciones tipo RISC, cuyo número puede variar desde 35 para PICs de gama baja a 70 para los de gama alta. Las instrucciones se clasifican entre las que realizan operaciones entre el acumulador y una constante, entre el acumulador y una posición de memoria, instrucciones de condicionamiento y de salto/retorno, implementación de interrupciones y una para pasar a modo de bajo consumo llamada sleep.

Está caracterizada por las siguientes prestaciones:

- Área de código y de datos separadas (Arquitectura Harvard).
- Un reducido número de instrucciones de longitud fija.
- Implementa segmentación de tal modo que la mayoría de instrucciones duran 1 tiempo de instrucción (o 4 tiempos de reloj). Pueden haber instrucciones de dos tiempos de instrucción (saltos, llamadas y retornos de subrutinas y otras) o inclusive con más tiempo de instrucción en PICs de gama alta. Esto implica que el rendimiento real de instrucciones

por segundo del procesador es de al menos  $1/4$  de la frecuencia del oscilador.

- Un solo acumulador (W), cuyo uso (como operador de origen) es implícito (no está especificado en la instrucción).
- Todas las posiciones de la RAM funcionan como registros de origen y/o de destino de operaciones matemáticas y otras funciones.
- Una pila de hardware para almacenar instrucciones de regreso de funciones.
- Una pequeña cantidad de espacio de datos direccionable (típicamente, 256 bytes), extensible a través de manipulación de bancos de memoria.
- El espacio de datos está relacionado con el CPU, puertos, y los registros de los periféricos.
- El contador de programa está también relacionado dentro del espacio de datos, y es posible escribir en él (permitiendo saltos indirectos).

### 3. CONCLUSIONES

Como conclusión del trabajo podemos hacer referencia a la variedad de ejemplos de sistemas embebidos que se utilizan en la actualidad:

- Máquinas y otros dispositivos de respuesta y contestación automática.
- Sistemas para el control automático de objetos en movimiento.
- Los teléfonos móviles y sistemas GPS
- Dispositivos de una red de ordenadores, tales como routers, hubs, firewalls etc.
- Impresoras para computadoras, copiadoras y dispositivos multifunción.
- Dispositivos de control para HDD y FDD
- Controladores de un motor de automóvil, sistemas de frenos, sistemas de cierre, etc
- Aparatos electrodomésticos tales como refrigeradores, acondicionadores de aire, sistemas de seguridad, hornos de microondas, lavadoras, TV y sistemas de DVD, etc.
- Equipamiento médico
- Los asistentes personales como computadoras portátiles, notebooks, etc.
- Controladores Lógicos Programables (PLC) para aplicación industrial.
- Dispositivos para juegos de video.
- Cámaras digitales y cámaras de video, etc.

## 4. BIBLIOGRAFÍA Y/O REFERENCIAS

En este apartado se hará referencia a toda la bibliografía utilizada.

- [1] Jonathan Valvano. Introducción a los Sistemas De Microcomputadora Embebidos: Simulación De Motorola G811 Y G812. (2003)
- [2] José Daniel Muñoz Frías. Sistemas empotrados en Tiempo Real (2009).

## 5. ENLACES DE INTERÉS

Se detallarán los enlaces o links que puedan ser de interés en relación con el tema de la tarea.

[http://es.wikipedia.org/wiki/Arquitectura\\_ARM](http://es.wikipedia.org/wiki/Arquitectura_ARM)  
[http://es.wikipedia.org/wiki/Sistema\\_embebido](http://es.wikipedia.org/wiki/Sistema_embebido)  
<http://es.wikipedia.org/wiki/AVR>  
<http://webdiis.unizar.es/~joseluis/SE.pdf>  
<http://es.wikipedia.org/wiki/Microcontrolador>