

Principios de Mecatrónica – SDI-11561  
Instituto Tecnológico Autónomo de México

José Guadalupe Romero Velázquez  
Departamento Académico de Sistemas Digitales

Primavera 2019

**Clases** ■ Grupos 1 y 2, Martes y Jueves 11:30 a 13:30  
en Río Hondo

**Laboratorio** ■ Grupos x,y: – en Laboratorios de Ingeniería

**Instructor** Prof. José Guadalupe Romero (Grupos 1 y 2)

**Oficina** Río Hondo. Departamento Académico de Sistemas Digitales

**Correo-e:** jose.romerovelazquez[at]itam.mx

**Atención en oficina.** Lunes a Jueves de 14:30 a 15:30.  
Disponible en otros horarios previa cita.

**Documentos del curso.** Se utilizará Dropbox para obtenerlos.

- Libros.** ● Introduction to Mechatronics and measurement systems, David. G. Alciatore et al;.
- Sensor and Signal Conditions, Ramon Pallas Areny et al;.
  - Control Systems Engineering, Norman S. Nise.

## 1 Objetivos

Al término del curso, el alumno:

- Comprenderá la arquitectura de un micro-controlador.
- Comprenderá el funcionamiento de las interfaces entre sistemas de cómputo y sensores y actuadores.
- Aplicará técnicas de modelado para sistemas físicos
- Programará un sistema digital con capacidades básicas de percepción y acción.

## 2 Temas

### 1. Introducción

- Sistemas de cómputo de aplicación específica (embedded computers)
- Sistemas mecatrónicos

### 2. Arquitectura de un micro-controlador

- Arquitectura RISC y CISC
- Organización del micro-controlador AVR
- Arquitectura del Set de Instrucciones del micro-controlador AVR
- Programación del micro-controlador AVR

Los temas 1 y 2 serán cubiertos entre 7 y 8 sesiones, la siguiente sesión, examen.

### 3. Percepción y Acción en Sistemas Mecatrónicos

- Percepción: Sensores y Algoritmos
- Acción: Actuadores, Estructuras y Algoritmos
- Análisis, síntesis y toma de decisiones: Computadoras y Algoritmos
- Comunicación: Tecnología y Algoritmos para Wireless Sensor Networks

El tema 3 será cubierto entre 10 y 11 sesiones, la siguiente sesión, examen.

### 4. Modelado de Sistemas Mecatrónicos

- Sistemas y Elaboración de Modelos
- Análisis y diseño de sistemas dinámicos
- Sistemas Eléctricos
- Sistemas Mecánicos
- Sistemas electromecánicos

- Sistemas Térmicos

El tema 4 será cubierto entre 10 y 11 sesiones.

### 3 Habilidades a desarrollar durante el curso

Al termino satisfactorio del curso, el alumno tendrá las siguientes habilidades:

- Comprenderá la arquitectura de y programará un micro-controlador
- Diseñará elementos de percepción en un sistema mecatrónico
- Diseñará elementos de acción en un sistema mecatrónico
- Modelará sistemas eléctricos y mecánicos
- Calculará la respuesta de sistemas eléctricos y mecánicos a entradas conocidas

### 4 Laboratorio

### 5 Reglas de evaluación

**Exámenes.** Dos exámenes parciales (30 %) y un examen final (40 %).

**Requisitos para calificación aprobatoria.** Aprobar el examen final y el laboratorio.

**Calificación final.** = exámenes + laboratorio (30 %)

**Puntos extra.** Durante la duración del curso, puntos extras por participaciones en clase o soluciones presentadas a problemas propuestos se contará como 2 o 3 décimas sumadas al examen final correspondiente.

## 6 Comportamiento Ético

Todo comportamiento fraudulento será tratado conforme establece el Reglamento de Alumnos.

### Embedded Computers

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. De esta forma, se pueden definir como todo sistema que NO es una PC de escritorio, ni un servidor, ni una workstation, ni una supercomputadora, ni un cluster distribuido, en fin, NO es un sistema programable de proposito general.

Unos ejemplos de sistemas embebidos podrían ser:

- Embebidos en industria automotriz : navegador GPS, de la aviación: piloto automatico, control de aterrizaje
- Embebidos en telecomunicaciones: routers, modems
- Embebidos en comunicaciones: teléfonos celulares
- Embebidos en el hogar: control de heladeras, microondas, robots que aspiran, cortan pasto.

**Los sistemas embebidos se pueden programar directamente en el lenguaje ensamblador del microcontrolador o en el microprocesador incorporado sobre el mismo**, o también, utilizando los compiladores específicos, pueden utilizarse lenguajes como C o C++; en algunos casos.

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

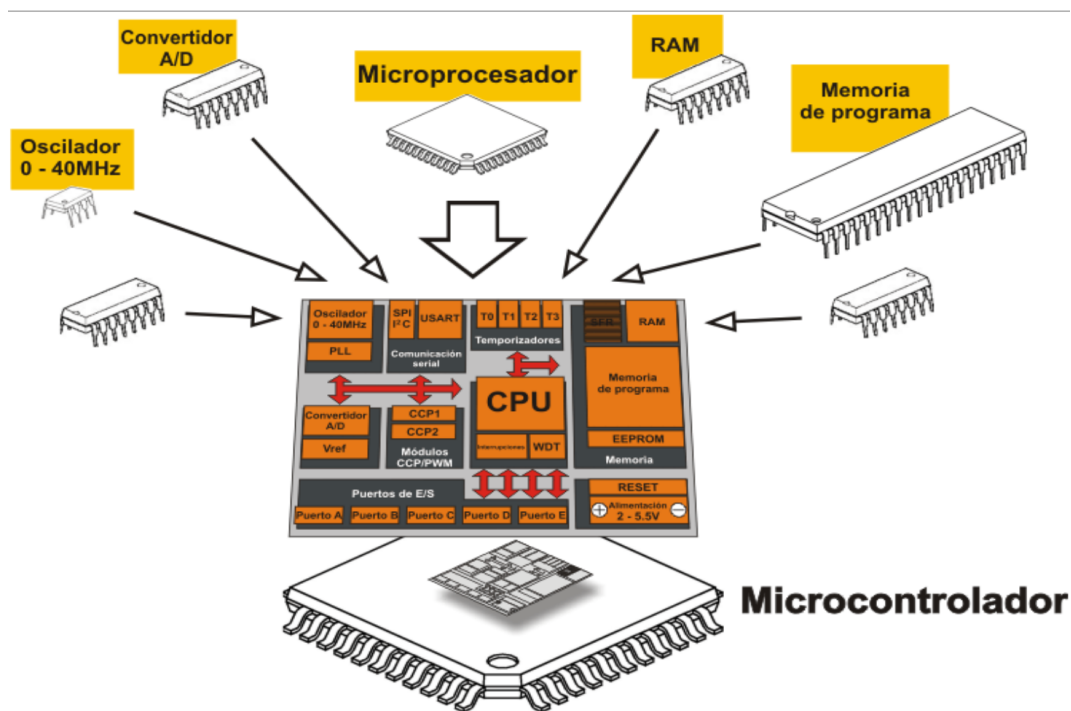
**Se encuentran frecuentemente conectados a ambientes físicos a través de sensores y actua-**

dores. 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).

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.

**Un sistema embebido estará formando por un microprocesador y un software que se ejecute sobre éste.** Sin embargo este debe poder guardarse para luego ser ejecutado. 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 la conexión 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.



**Sistema mecatrónico:** son el resultado de la integración física de las ramas que comprenden el estudio de



sistemas mecánicos, máquinas eléctricas (en algunos casos sistemas de potencia), sensores, actuadores, sistemas de comunicación (telecomunicación) y cómputo. Todo, con la finalidad de obtener una integración funcional, el cual procesa la información disponible en el sistema a través de herramientas de software resultando en acciones automáticas de control ejecutadas al sistema físico.

### **Características de un microcontrolador**

Por ejemplo, un microcontrolador de alto desempeño Atmega328P de 8 bits, con arquitectura **RISC** es utilizado en un Arduino Uno.

Posee características como: 32 KB de memoria flash, 1 KB de memoria EEPROM<sup>1</sup>, 2 KB de SRAM<sup>2</sup>, 23 líneas de E/S de propósito general, 32 registros de proceso general, tres temporizadores flexibles/contadores con modo de comparación, interrupciones internas y externas, una interface serial orientada a byte de 2 cables, 6 canales 10-bits Conversor A/D, “watchdog timer” programable con oscilador interno, y cinco modos de ahorro de energía seleccionables por software. El dispositivo opera entre 1.8 y 5.5 voltios. Su arquitectura le permite ejecutar instruc-

---

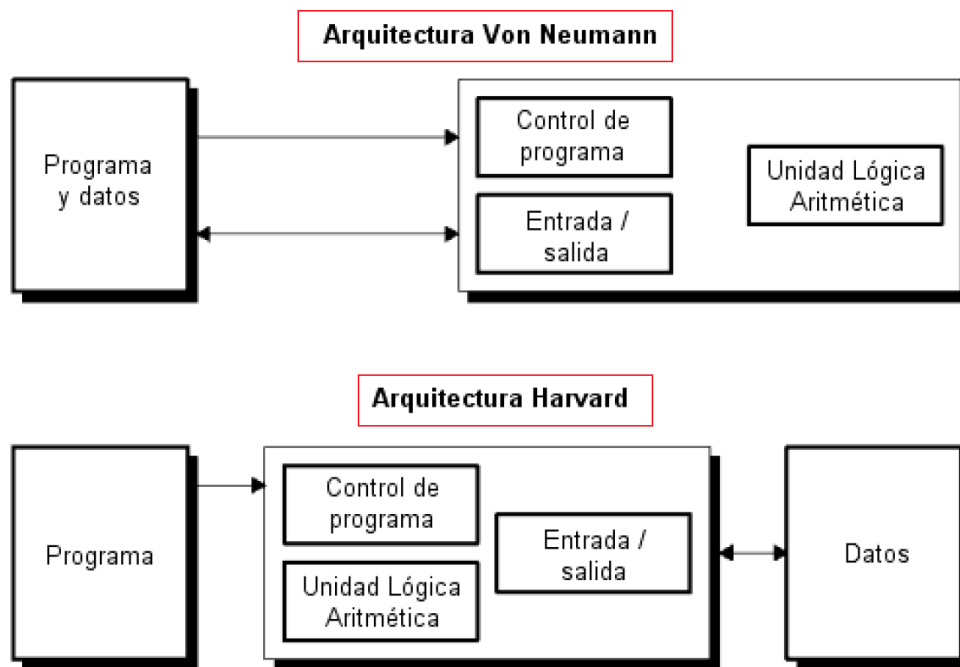
<sup>1</sup>Es un tipo de memoria ROM que puede ser programada, borrada y reprogramada eléctricamente, a diferencia de la EPROM que ha de borrarse mediante un aparato que emite rayos ultravioleta. Son memorias no volátiles.

<sup>2</sup>es una memoria RAM estática de acceso aleatorio, se puede denominar como un tipo de tecnología de memoria RAM basada en semiconductores, capaz de mantener los datos, mientras siga alimentada.

ciones en un solo ciclo de reloj.

La arquitectura de los sistemas embebidos puede ser de dos tipos: Arquitectura Harvard o Arquitectura Von Neumann

- **Von Neumann:** Es una familia de arquitectura de computadora que utilizan **el mismo dispositivo de almacenamiento tanto para las instrucciones de programa como para los datos**. La mayoría de las computadoras modernas están basadas en esta arquitectura, aunque pueden incluir otros dispositivos adicionales (por ejemplo para gestionar las interrupciones de dispositivos externos). **De esta forma las operaciones de lectura y escritura no pueden realizarse simultáneamente.**
- **Harvard:** Esta arquitectura utiliza **dispositivos de almacenamiento físicamente separados para las instrucciones de programa y para los datos**. El inconveniente radica en que debido a que esta divide la información para que funcione lo mejor posible, es necesario que la frecuencia de lectura de instrucciones y de datos sea aproximadamente la misma.



Generalmente, la arquitectura de los sistemas embebidos es de tipo **Harvard**, ya que existe una memoria RAM que almacena los datos y otra de tipo EPROM o EEPROM que almacena el programa. Al usar esta arquitectura **se tiene la ventaja de poder usar simultáneamente los dos tipos de memoria.**

Por otro lado, el **conjunto de instrucciones** define las operaciones básicas que puede realizar el procesador, que conjugadas y organizadas forman lo que conocemos como software. El conjunto de instrucciones vienen siendo como las letras del alfabeto, el elemento básico del lenguaje, que organizadas adecuadamente permiten escribir

palabras, oraciones y cuanto programa se le ocurra.

Existen dos tipos de set de instrucciones: **CISC** y **RISC**. Las instrucciones **CISC** (Complex Instruction Set Computer, Computadora de Conjunto de Instrucciones Complejo) se caracterizan por tener un repertorio amplio de instrucciones, que pueden hacer cálculos muy complejos. Pero también se complica su uso. Además las instrucciones muy complejas no son utilizadas con mucha frecuencia y en muchos casos eran responsables del retardo del desempeño de todo el sistema. Esto en particular debido a que la Unidad de Control debe ser más compleja y **si es microprogramada provoca que todas las instrucciones demoren mayor tiempo en su ejecución.**

El tipo **RISC** (Reduced Instruction Set Computer, Computadora con Conjunto de Instrucciones Reducido) se distingue por sus pocas instrucciones, y un tamaño fijo por instrucción además permite el paralelismo en la ejecución.

Las principales características de estas arquitecturas son las siguientes:

CISC	RISC
Instrucción multiciclo	Instrucción de un único ciclo
Carga/almacenamiento en una instrucción	Carga/almacenamiento son instrucciones separadas
Arquitectura memoria-memoria	Arquitectura registro-registro
Utiliza memoria de microprograma localizado en el circuito integrado	Implementa las instrucciones directamente en hardware
Reduce la dificultad de implementar compiladores	Compiladores complejos
Versatilidad del repertorio de instrucciones	Se añaden nuevas instrucciones sólo si son de uso frecuente
Facilita la depuración de errores	
	Instrucciones de tamaño fijo y presentadas en un reducido número de formatos.

Una de las mayores ventajas de RISC respecto a CISC es que **utiliza un sistema de direcciones no destructivas en RAM**. Eso significa que a diferencia de CISC, RISC conserva después de realizar sus operaciones en memoria los dos operandos y su resultado, reduciendo la ejecución de nuevas operaciones.