



**Ministerio de Cultura y Educación**  
**Universidad Nacional de San Luis**  
**Facultad de Ciencias Físico Matemáticas y Naturales**  
**Departamento: Física**  
**Area: Area V: Electronica y Microprocesadores**

**(Programa del año 2016)**

**I - Oferta Académica**

<b>Materia</b>	<b>Carrera</b>	<b>Plan</b>	<b>Año</b>	<b>Período</b>
PROCESADORES II	ING.ELECT.O.S.D	010/0 5	2016	1° cuatrimestre

**II - Equipo Docente**

<b>Docente</b>	<b>Función</b>	<b>Cargo</b>	<b>Dedicación</b>
MURDOCCA, ROBERTO MARTIN	Prof. Responsable	P.Adj Exc	40 Hs
HERNANDEZ VELAZQUEZ, SERGIO FE	Responsable de Práctico	A.1ra Exc	40 Hs
ANDRADA TIVANI, ASTRI EDITH	Auxiliar de Práctico	A.2da Simp	10 Hs

**III - Características del Curso**

<b>Credito Horario Semanal</b>				
<b>Teórico/Práctico</b>	<b>Teóricas</b>	<b>Prácticas de Aula</b>	<b>Práct. de lab/ camp/ Resid/ PIP, etc.</b>	<b>Total</b>
Hs	3 Hs	Hs	2 Hs	5 Hs

<b>Tipificación</b>	<b>Periodo</b>
B - Teoria con prácticas de aula y laboratorio	1° Cuatrimestre

<b>Duración</b>			
<b>Desde</b>	<b>Hasta</b>	<b>Cantidad de Semanas</b>	<b>Cantidad de Horas</b>
14/03/2016	24/06/2016	15	75

**IV - Fundamentación**

En la industria, automóvil, comunicaciones, entretenimiento, laboratorios y en general en todos los equipos electrónicos actuales se tiene un microcontrolador embebido encargado de controlar el funcionamiento de dicho aparato o instrumento. El Ingeniero Electrónico debe enfrentar el mercado con capacidades de diseño de sistemas digitales cada vez más complejos, de especificaciones variables, de menor costo y reducidos tiempos de desarrollo. Conocer por lo tanto estas tecnologías como así también su programación y herramientas de desarrollo es importante para el diseño, implementación y mantenimiento de estos equipos electrónicos.

**V - Objetivos / Resultados de Aprendizaje**

Estudiar la arquitectura y programación de microprocesadores y microcontroladores actuales de 8, 16 y 32 bits.  
 Estudiar una familia de microcontroladores y utilizar herramientas de desarrollo para realizar la programación de los mismos en lenguajes de alto nivel.  
 Utilizar herramientas de modelado de software para desarrollar los modelos de programación de Sistemas Embebidos.  
 Desarrollar los conocimientos y habilidades necesarios para diseñar un sistema embebido dedicado a una aplicación específica descrita mediante requisitos de diseño, tanto a nivel de conexionado eléctrico (Hardware) como de su programación (Software).  
 Mejorar la habilidad para escribir documentos relacionados con un desarrollo de ingeniería.

## VI - Contenidos

### **Unidad 01: Tecnologías y Arquitecturas.**

Tecnologías de Sistemas Digitales. Lógica cableada, programable, hardware reconfigurable. Evolución de los sistemas de cómputo. Lógica programable: el Microprocesador y el Microcontrolador. Clasificación por aplicación o propósito: GPP, Microcontrolador, DSP, ASIC, ASIP, GPU. Clasificación por arquitectura: RISC, CISC, Superescalar, VLIW, Reconfigurable, System on chip, Network on chip. Clasificación de Flynn: SISD, SIMD, MISD, MIMD. Clasificación por conjunto de instrucciones (ISA). Microcontroladores. Características generales. Aplicaciones. Mercado actual de Microcontroladores. Sistemas Embebidos. Definición. Áreas de aplicación de sistemas embebidos.

### **Unidad 02: Arquitecturas de Microcontroladores de 8 bits.**

Estudio de un Microcontrolador RISC de 8 bits. Características Generales. La Unidad Aritmético y Lógica (ALU). El registro W. Ciclo de máquina y ejecución de instrucciones. El Contador de Programa (PC). Segmentado (pipeline) en la ejecución de instrucciones. Osciladores. Reset. Conexiones básicas del Microcontrolador. Alimentación. Familias de microcontroladores PIC: gama baja, media y alta. Descripción de la Arquitectura general de la gama media y alta. Características. Multiplicador HW. Descripción de placa de desarrollo para microcontrolador de gama alta.

### **Unidad 03: Organización de la Memoria.**

La memoria en los microcontroladores. Conceptos básicos. Organización lógica de la memoria. Repaso de memorias usadas en Microcontroladores: RAM, ROM, EPROM y OTP, EEPROM y FLASH. La memoria en los microcontroladores PIC de gama alta. La memoria de programa. El contador de programa (PC). La pila (Stack). La memoria de datos. Mapa de memoria y bancos en la memoria de datos. Los registro de funciones especiales (SFR). Registro STATUS. Registro OPTION. La memoria EEPROM de datos.

### **Unidad 04: Programación y herramientas de desarrollo.**

Repaso de código máquina y lenguaje ensamblador. Estructura de las instrucciones. Modos de direccionar los datos. Ortogonalidad. Repertorio de instrucciones de los PIC de gama media y alta. Instrucciones de transferencia de datos, aritméticas - lógicas, transferencia, de control, de salto condicional e instrucciones que operan con bits. Directivas del lenguaje ensamblador. Subrutinas. Manejo de tablas en memoria de programa. Implementación de estructuras de programación en lenguaje ensamblador. Herramientas de desarrollo. Entorno de desarrollo integrado MPLABX IDE. Simuladores. Emuladores. Programadores. Programación ICSP. Técnicas de depurado (Debugging).

### **Unidad 05: Periféricos Internos del Microcontrolador.**

Estructura de los puertos de los microcontroladores PIC. Registros de configuración de puertos. Características eléctricas. Los temporizadores y contadores. El módulo Timer 0. Modos de funcionamiento y programación. El Módulo Timer1. El módulo Timer 2. Módulo CCP: Modo captura, comparación y PWM. Comparadores analógicos. Conversores Analógico-Digital. Comunicación serie: la USART. Interrupciones. Tipos de interrupciones. Las interrupciones en los microcontroladores PIC. Interrupciones desde eventos externos e internos. El Perro guardián (WDT).

### **Unidad 06: Programación en lenguaje de alto nivel.**

Conceptos básicos sobre compilado en alto nivel. Compilador C. Pre-procesador. Librerías. Elementos del lenguaje C. Estructura de un programa en C. Cross compilador C para microcontroladores. Aspectos particulares del lenguaje C en los microcontroladores. Control de periféricos e interrupciones en lenguaje C. Técnicas de programación de sistemas embebidos. Sistemas reactivos, interactivos y transformacionales. Programación mediante interrupciones (background / foreground). Máquinas de estados finitos (FSM). Tablas y diagramas de estados. Implementación de FSM en C. Herramientas de modelado para máquinas de estado.

### **Unidad 07: Interface con Dispositivos Externos.**

Tecnologías lógicas. Características. Interfaces. Dispositivos de E/S. Conexión de dispositivos externos al puerto del microcontrolador. Características eléctricas de los puertos del microcontrolador. Interfaces con dispositivos de entrada. Conexión de Interruptores y pulsadores. Rebote de contactos. Tratamiento por software. Interface con teclados matriciales. Técnicas de exploración de teclados. Adaptación de señales de entrada. Opto aislación. Interfaces con dispositivos de salida. Interface con LEDs. Visualizadores o Displays. Distintos tipos de display. Conexión a Display 7 segmentos. Conexión a Display multiplexado. Display LCD inteligente. Displays gráficos. Interfaces de potencia. Interface con circuitos de corriente alterna. Interfaces con reles. Interface con otros periféricos externos.

### **Unidad 08: Arquitecturas de Microcontroladores de 32 bits.**

Arquitecturas ARM. Breve historia de ARM. Características generales. Mercado actual de Microcontroladores ARM. Características. Aspectos fundamentales de la arquitectura. Registros de propósitos generales y especiales. La Unidad Aritmético y Lógica (ALU). Modos de operación. Excepciones y controlador de interrupciones. Características del Timer SysTick. Mapa de memoria. Arquitectura de buses. Pipeline. Alimentación. Capacidades de cómputo. Set de instrucciones. Diferencias entre las variantes de Cortex-M. Estudio de un Procesador comercial Cortex-M. Características. Interrupciones utilizando SysTick. Descripción de placa de desarrollo para Cortex-M.

### **Unidad 09: Herramientas de Programación y Modelado de Software.**

Programación en C para ARM. Entorno de desarrollo integrado (IDE). Programación eficiente en C. Uso de modelos en la programación de microcontroladores. Control de periféricos e interrupciones en lenguaje C. Máquina de Estado y Diagrama de Estado. Codificación en C de modelos. Ejercitación: salidas, entradas, temporización, drivers y aplicaciones.

### **Unidad 10: Librerías de periféricos y Drivers.**

Librerías de abstracción de hardware. Introducción a las librerías CMSIS. Otras librerías. Reglas de codificación y convenciones. Gestión de entradas y salidas y soporte a interfaces de comunicación en CMSIS. Drivers para control de LEDs, Display LCD. Uso de la UART. Interface I2C y SPI. Uso del Conversor AD. Ejemplos de aplicaciones típicas.

### **Unidad 11: Otras Arquitecturas actuales.**

Otros microcontroladores de 8, 16 y 32 bits. Plataformas comerciales. Arduino. Características. IDE. Aplicaciones. Shields. Plataforma ChipKit UNO32. Arquitecturas Intel. Plataforma Intel Galileo. Características. Herramientas de desarrollo. Compatibilidad con Arduino. Aplicaciones con Intel Galileo. Desarrollo de aplicaciones.

## **VII - Plan de Trabajos Prácticos**

### **PRÁCTICAS DE LABORATORIO**

#### Laboratorio 01

Herramientas de Desarrollo para Microcontroladores.

#### Laboratorio 02

Tecnologías Lógicas e interfaces con dispositivos externos.

#### Laboratorio 03

Introducción a la programación embebida en C y uso de la placa de desarrollo.

#### Laboratorio 04

Programación Avanzada en C.

#### Laboratorio 05

Herramientas de Desarrollo para ARM Cortex-M3.

#### Laboratorio 06

Manejo de Entradas y Salidas con Cortex-M3. Uso de Periféricos.

#### Laboratorio 07

Uso de la plataforma Intel Galileo.

### **PRÁCTICAS DE RESOLUCION DE PROBLEMAS**

Guía 01 - Tecnologías y Arquitecturas.

Guía 02 - Interfaces de Entrada – Salida (I).

Guía 03 - Microcontroladores PIC.

Guía 04 - Programación en Ensamblador.

Guía 05 - Periféricos del Microcontrolador.

Guía 06 - Programación Embebida en C.

Guía 07 - Interfaces de Entrada – Salida (II).

Guía 08 - Microcontroladores Avanzados. Arquitecturas ARM.

## VIII - Regimen de Aprobación

### A) Promoción y Examen Libre

Esta materia no se aprueba por régimen de promoción.

Esta materia no se puede rendir en la modalidad de examen alumno libre (examen libre).

### B) Regularización

Para regularizar la materia los alumnos deberán cumplir con los siguientes requisitos:

- Guías de laboratorio: aprobar la totalidad de las guías de laboratorio con su respectiva carpeta de informes. La cantidad de laboratorio es 7 y habrá un total de 3 recuperaciones. Antes de ingresar a cada laboratorio se tomará un cuestionario con tres preguntas de tipo múltiple opción. Para aprobar el cuestionario se requieren dos preguntas correctas como mínimo. El alumno que no apruebe el cuestionario deberá recuperar ese laboratorio, perdiendo con esto una de las tres recuperaciones posibles.

- Parciales: Durante el cuatrimestre se tomarán dos parciales. Los parciales incluirán temas teóricos y temas desarrollados en las guías de problemas. Cada parcial tiene dos recuperaciones. La aprobación de los parciales se obtiene con una nota igual o superior a 7 (siete).

Se debe rendir un examen final para aprobar la materia. Este examen puede ser oral o escrito.

## IX - Bibliografía Básica

- [1] Valdez Fernando E., Areny Ramón P., Microcontroladores: Fundamentos y aplicaciones con PIC, España, Marcombo Ediciones Técnicas, 2007, 344 p.
- [2] M. Rafiqzaman, Ph.D., Microcontroller Theory and Applications with the PIC18F, Hamilton Printing, 2011, 498p.
- [3] Dogan Ibrahim, Advanced PIC Microcontroller Projects in C: From USB to RTOS with the PIC18F Series, ELSEVIER, 2008, 560p.
- [4] Galeano Gustavo, Programación de Sistemas Embebidos en C, México, Alfaomega Grupo Editor, 2009, 544p.
- [5] Breijo Eduardo García, Compilador C CCS y Simulador Proteus para Microcontroladores PIC, España, Alfaomega Grupo Editor, 2008, 260p.
- [6] Bates Martin, Interfacing PIC Microcontrollers: Embedded Design by Interactive Simulation, Gran Bretaña, Elsevier, 2006, 298p.
- [7] Saravia Andrés R. Bruno, Coria Ariel, Arquitectura y Programación de Microcontroladores PIC – Buenos Aires, MC Electronics, 2010, 312p.
- [8] Tim Wilmshurst, Designing Embedded Systems with PIC Microcontrollers, Principles and Applications, ELSEVIER, 2010, 651p.
- [9] The Definitive Guide to the ARM Cortex-M3 - Joseph Yiu, 2ª Edición, Newnes Elsevier Inc, 2010.
- [10] Cortex™-M3, Revision r2p0, Technical Reference Manual – ARM
- [11] ARM@v7-M Architecture, Reference Manual – ARM
- [12] UM10360 LPC176x/5x User Manual & LPC1769/68/67/66/65/64/63 Product data sheet – NXP
- [13] Reference Guide & User Guide of visualState – IAR
- [14] El Lenguaje Unificado de Modelado, G. Booch, J. Rumbaugh, I. Jacobson, 2ª Edición, Addison-Wesley, 2006
- [15] Intel Galileo Board User Guide – March 2014.
- [16] Intel Galileo Getting Started Guide – 16 October 2013.
- [17] Getting Started with Intel Galileo- Matt Richardson Maker Media, Inc. - 2014 - 194p.

## X - Bibliografía Complementaria

- [1] Ganssle Jack, The Firmware Handbook – EEUU, ELSEVIER, 2004, 365p.
- [2] Di Jasio Lucio, Wilmshurst Tim, Ibrahim Dogan, Morton John, Bates Martin, Smith Jack, Smith D.W., and Hellebuyck Chuck, PIC Microcontrollers: Know It All, EEUU, ELSEVIER, 2008, 928p.
- [3] Zurawski Richard, Embedded Systems HandBook, Taylor & Francis Group, 2006, 1089p.
- [4] Katzen Sid, The Essential PIC18 Microcontroller, Springer, 2010, 612p.
- [5] Timothy D. Green, Embedded Systems Programing with the PIC16F877, 2da Edicion, 2008, 196p.
- [6] RobertT B. Reese, Microprocessors: From Assembly Language to C Using the PIC18Fxx2, Da Vinci Engineering Press, 2005, 687p.
- [7] Keith E. Curtis, Embedded multitasking with Small Microcontrollers, , ELSEVIER, 2006, 417p.

[8] Ken Arnold, Embedded Controller Hardware Design, , LLH Technology Publishing, 2000, 244p.

[9] Tammy Noergaard, Embedded Systems Architecture A Comprehensive Guide for Engineers and Programmers, , ELSEVIER, 2005, 657p.

## **XI - Resumen de Objetivos**

Estudiar la arquitectura y programación de microprocesadores y microcontroladores de 8, 16 y 32 bits actuales.

Utilizar herramientas de desarrollo para realizar la programación de una familia de microcontroladores.

Dotar al alumno de los conocimientos necesarios para desarrollar un sistema embebido.

## **XII - Resumen del Programa**

Unidad 01 - Tecnologías y Arquitecturas.

Unidad 02 - Arquitecturas de Microcontroladores de 8 bits.

Unidad 03 - Organización de la Memoria.

Unidad 04 - Programación y herramientas de desarrollo.

Unidad 05 - Periféricos Internos del Microcontrolador.

Unidad 06 - Programación en Lenguaje de alto nivel.

Unidad 07 - Interface con Dispositivos Externos.

Unidad 08 - Arquitecturas de Microcontroladores de 32 bits.

Unidad 09 - Herramientas de Programación y Modelado de Software.

Unidad 10 - Librerías de periféricos y Drivers.

Unidad 11- Otras Arquitecturas actuales.

## **XIII - Imprevistos**

Cambio tecnológico: actualización por parte de fabricantes de microcontroladores.

Incompatibilidades y bugs de nuevas versiones de software.

Disponibilidad en el mercado local de partes y componentes necesarios para las prácticas.

## **XIV - Otros**