



SISTEMAS CON MICROPROCESADORES Y MICROCONTROLADORES

1. OBJETIVOS

- 1) Entender funcionalmente cómo trabaja un sistema de computadora.
- 2) Escribir programas en lenguaje ensamblador para ARM M4 y ejecutarlos en un simulador de un microcontrolador NXP LPC4337.
- 3) Describir los diversos módulos internos y el mapa de Memoria del NXP LPC4337.
- 4) Ampliar la capacidad limitada de conexión de E/S de un microcontrolador mediante diversos métodos.
- 5) Emplear el Controlador de Prioridades de Interrupciones Anidadas (NVIC) de la arquitectura ARM para configurar el sistema de Interrupciones.
- 6) Emplear interrupciones externas y los puertos de E/S paralelos (GPIO) para diversas funciones.
- 7) Describir el funcionamiento de un sistema de Acceso Directo a Memoria para transferir datos hacia y desde un periférico.
- 8) Emplear el módulo de Temporizadores y sus canales (TIMER 0/1/2/3) en sus modos de Captura de Entrada y Comparación de Salidas.
- 9) Emplear los conceptos aprendidos en el curso para diseñar un sistema embebido.

2. CONTENIDOS

- 1) INTRODUCCIÓN A ARQUITECTURA DE COMPUTADORAS, PROGRAMACIÓN DEL ARM M4.
 - a) **Introducción a la Estructura de Computadoras.** Configuración básica. Programa e Instrucciones. Unidad Central de Procesamiento (CPU). Camino de Datos y Control. Bus: implementación mediante conexión con “tercer estado” y “colector abierto”. Interacción CPU-Memoria. Formato de la Instrucción. Ciclo de la instrucción. Tipos de Instrucciones. Arquitectura del Set de Instrucciones. Lenguaje de Transferencia de Registros (RTN) abstracto y concreto. Ejemplo de funcionamiento de un CPU. Interrupciones y Excepciones. Lenguajes: de máquina, ensambladores, superiores. Arquitectura Von Newman y Harvard. Pipelining. Concepto de Sistema Operativo, ensamblador, compilador e intérprete. Sistemas de Propósito General. Sistemas Embebidos y Generales. Concepto de Pipelining.
 - b) **Diseño con Microcontroladores.** Pasos de un proyecto: introducción, objetivos, desarrollo, conclusiones. Conceptos de trabajo en grupo. Entrada con llaves y teclas mecánicas. Ejemplo de un proyecto: diseño de un sistema de alarmas domiciliario.
 - c) **ISA del Microprocesador (ARM Cortex-M4).** Descripción General. Registros. Tipo de Datos. Camino de Datos. Duración de las instrucciones. Clases de Instrucciones. Modos de direccionamiento: implícito, inmediato, de registros, de registro indirecto, indexado, relativo. Registro de Código de Condición: Flags. Tipos de Instrucciones. Set de Instrucciones. Ejemplos.
 - d) **Conceptos de Programación.** Lazos. Pilas. Direccionamiento de Pila. Subrutinas. Variables Globales y Locales. Pasaje de Parámetros. Estándares. Subrutinas Anidadas y Recursivas. Transparencia. Ejemplos.
- 2) CONCEPTOS DE HARDWARE Y SU APLICACIÓN AL MC68HC08.
 - a) **Sistema de Microcomputador.** Mapas de memoria con y sin redundancia. Conexión CPU – M – E/S. Señales. Conexión de dispositivos externos: registros, contadores, buffers,



- drivers/receivers. Dispositivos Externos configurables mediante Registro de Configuración y Status. Ejemplos. Bit-banding en ARM. E/S mapeada en memoria y E/S separada.
- b) **Sistema de Interrupciones.** El concepto de la interrupción. Interrupciones múltiples: conexiones, prioridades, método de encuestas, múltiples líneas, controlador de prioridades de interrupción (PIC). Interrupciones anidadas. Excepciones. Inicialización del sistema: Reset. Modos Supervisor/Usuario. Supervisor Call. Sistemas de interrupciones en ARM-M4: registros e instrucciones especiales. Ejemplo de cambio de contexto. NVIC: Visión lógica y física, prioridades, habilitación de interrupciones, interrupciones pendientes y activas, reset. SysTick Timer. Descripción: registros, inicialización, ejemplos con y sin interrupción. Ejemplo de un reloj en tiempo real.
 - c) **Descripción General del Microcontrolador NXP LPC4337.** El concepto de Familia de Microcontroladores. introducción a los módulos: Núcleos, Buses, Bancos de Memoria, Memoria, Reloj de tiempo real (RTC), Oscilador Interno, Reset y sus causas, SIM (Módulo de Integración del Sistema), Puertos de E/S, Conversor A/D y D/A, Módulos de bajo consumo, Watchdogs (perro guardián), Temporizadores. Introducción a periféricos: GPIO, SGPIO. Interrupciones: internas de los módulos, externas desde GPIO.
 - d) **Dispositivos de E/S: GPIO en LPC4337.** Modos de Trabajo y Configuración: pines, pull-up/down, default, registros. Esquema circuital. Ejemplos de aplicación: conexión de Hw Externo. Expansión de conexiones externas: decodificador y buffers. Interrupciones en el GPIO: grupales e individuales. Configuración: nivel/flancos, máscara, estado. Ejemplo: polling e interrupción.
 - e) **Sistema de Entrada-Salida (I/O).** Arquitectura de un sistema embebido: control, adquisición de datos, automatización. Interfaces. Comparación con Sistemas Analógicos y Digitales de lógica cableada. Confiabilidad y Disponibilidad. Entrada / Salida de Datos. Protocolos de comunicación sincrónicos y asincrónicos. Transferencia de datos entre periféricos y memoria: por programa, por interrupción, por acceso directo a memoria (DMA). Velocidad de transferencia. Comparación. Acceso directo a memoria (DMA): modo de bloques y robo de ciclo. Ejemplos. Limitaciones con dispositivos lentos. Limitaciones con dispositivos rápidos. Efecto de E/S de datos por interrupciones y DMA en la performance del CPU. Introducción a GPDMA en LPC4337.
 - f) **Temporizadores.** Descripción General. Esquema Circuital. Configuración: prescaler, registros de comparación, canales, interrupciones. Eventos de Entrada; Input Capture. Eventos de Salida: Output Capture. Ejemplos: esperar un tiempo determinado, medición de frecuencia de una señal externa, cálculo del ancho de un pulso, generación de un pulso, de una señal periódica.
- 3) **PROYECTOS DE DISEÑO**
Diseño de diferentes sistemas realizados por grupos de estudiantes. Seminario de Exposición y Defensa. Informe Final.



3. BIBLIOGRAFÍA

- Heuring, Vincent & Jordan, Harry F, Computer Systems Design and Architecture, Addison-Wesley, 1997.
- Yifeng Zhu, Embedded Systems with ARM Cortex-M Microcontrollers in Assembly Language and C, E-Man Press LLC, 2nd Edition. New York, 2015.
- Joseph Yiu, The Definitive Guide to ARM® Cortex®-M3 and Cortex®-M4 Processors, 3ed Edition. Newnes. Boston, 2013.
- Jonathan Valvano, Introduction to ARM Cortex-M Microcontrollers, Volume 1, 4th Edition, 2013.
- Jonathan Valvano, Embedded Systems: Real-Time Interfacing to ARM Cortex-M Microcontrollers, Volume 2, fourth edition, 2014.
- Tocci Ronald J., Widmer Neal S. & Moss Gregory L., Sistemas Digitales, 10ma Edición, Pearson Education, México, 2007.

4. METODOLOGÍA DE ENSEÑANZA

- Clases teóricas: 4 horas semanales, con exposición basado en transparencias (brindadas previamente al alumno mediante nuestro sitio web) y uso de pizarra.
- Trabajos prácticos de laboratorios: se desarrollan 4 trabajos prácticos de laboratorio sobre la temática vertida en clase, aproximadamente un trabajo por cada unidad temática. La duración de cada clase es de 2 horas.
 - Los trabajos de laboratorio se realizan empleando placas EDU-CIAA, sobre la cual se ha montado una placa “poncho” diseñada por la Cátedra. Al comenzar el período lectivo se entrega una placa por estudiante, la que se devuelve al terminar el cursado. El estudiante puede realizar los trabajos de laboratorio en forma individual o grupal en el lugar y el horario que mejor le convenga – rompiendo así la estructura rígida del laboratorio convencional. Cada trabajo de laboratorio se evalúa en máquina para todos los estudiantes juntos, mediante nuevos problemas de similar complejidad que deben resolver y ejecutar en las placas individuales en clase práctica.
- Trabajos prácticos de resolución de problemas: 5 trabajos de resolución de problemas de ingeniería sobre la temática vertida en clase.
- Diseño de un proyecto con Microcontrolador. Los estudiantes trabajan en grupo para el diseño de un Sistema y su posterior exposición en Seminario frente a la clase, con la presentación de un informe escrito final. Los grupos tienen un docente tutor a quien recurren para consultas. En el tema 2 de la asignatura se dicta un proyecto completo y los estudiantes eligen tema. El objetivo es que durante el cursado de la asignatura vayan incorporando los nuevos conocimientos al diseño del proyecto.

5. EVALUACIÓN

Método de evaluación para regularizar la asignatura, se deben cumplir los siguientes requisitos:

- 1) Aprobación de todos los trabajos de laboratorio.
- 2) Promedio superior o igual a 40/100 entre los dos exámenes parciales del curso.



- 3) Promedio ponderado de 40/100 entre evaluaciones de prácticos, laboratorios, parciales, seminarios de proyectos y cuestionarios de clase.
- 4) Asistencia superior o igual al 80% de las evaluaciones de prácticos.
- 5) Aprobación del Seminario de Proyecto y presentación del informe por escrito.

Método de evaluación para aprobar la materia (en caso que sea regular). La evaluación es del tipo distribuida contemplando la calificación de todas las calificaciones de prácticas, trabajos de laboratorio, diseño y seminario de proyecto, cuestionarios en clase y parciales.

6. CARGA HORARIA

- Teoría: 4 horas semanales distribuidas en dos clases de 2 horas cada una.
- Resolución de problemas de Ingeniería (Práctica): clases de 2 horas semanales. La frecuencia de las clases prácticas depende del avance de los conceptos teóricos vertidos en la teoría y de las clases de práctica de laboratorio: cuando hay clases de laboratorio no hay clases de resolución de problemas y viceversa.
- Formación Experimental: Laboratorios a implementar en la placas que se reparten a los estudiantes a principio de año, alternadas con las horas de trabajos prácticos de resolución de problemas de ingeniería. El Módulo III solo contiene trabajo de grupo para el diseño de un Sistema con Microcontrolador.
- **Carga Horaria Total:** 6 hs. semanales – 96 hs. por cuatrimestre.

7. OTRA INFORMACIÓN

Articulación: El aprendizaje de sistemas embebidos se encara de manera vertical. Concretamente, los conceptos básicos se aprenden en profundidad a partir de un microcontrolador de 32 bits. El estudiante contará con una base para encarar el aprendizaje y aplicación a otros procesadores y sistemas embebidos. Sistemas de tiempo real se aprenden en la asignatura Sistemas Operativos. Arquitectura de procesadores embebidos se estudian comparativamente con procesadores de propósito general en la asignatura Arquitectura de Computadoras. El lenguaje que se enseña es assembler, ya que C se enseña en materia previa de programación.

La materia se dicta en forma modular: comprenderá tres módulos. La evaluación de los dos primeros módulos se realizará mediante los correspondientes trabajos prácticos, de laboratorio y examen parcial. La evaluación del tercer módulo se llevará a cabo mediante seminarios de diseño de diversos temas, a cargo de grupos de estudiantes. Una vez aprobado el seminario, cada grupo presentará un informe final

Formación Experimental: Los trabajos de laboratorio se realizan empleando placas EDU-CIAA, sobre la cual se ha montado una placa “poncho” diseñada por la Cátedra. Al comenzar el período lectivo se entrega una placa por estudiante, la que se devuelve al terminar el cursado. El estudiante puede realizar los trabajos de laboratorio en forma individual o grupal en el lugar y el horario que mejor le convenga – rompiendo así la estructura rígida del laboratorio convencional. Cada trabajo de laboratorio se evalúa en máquina para todos los estudiantes juntos, mediante nuevos problemas de similar complejidad que deben resolver y ejecutar en las placas individuales en clase práctica.

Formación Práctica: Se emplea una metodología inductiva. Problemas de ejercitación para prácticos se entregan una semana previa a su evaluación, con guía de algunos problemas resueltos. Los estudiantes



intentan por su cuenta resolverlos contando con posibilidad de consultas al cuerpo docente de la cátedra. La evaluación del práctico se realiza con problemas que plantean situaciones nuevas empleando los conceptos con que se ejercitaron y aprendieron en clases teóricas. Al terminar la evaluación, el Docente a cargo de la misma muestra la resolución de un problema del tema con complejidad superior.

Seminarios de Proyecto: Se emplea una metodología inductiva. El Profesor enseña cómo se resuelve en general un problema de diseño al comienzo del curso. Los estudiantes proceden a elegir su tema de proyecto. A medida que transcurre el curso, los estudiantes van obteniendo las herramientas para diseñar el proyecto.

Manuales de Fabricantes disponibles para descargar:

Sitio de la cátedra: <http://microprocesadores.unt.edu.ar/procesadores>

- Manuales de ARM Cortex M4 – (sitio Web de la Cátedra).
- Manual del Microcontrolador LPC4337: UM10503 NXP LPC4337 (Sitio Web de la Cátedra).