



Workshops hands-on del SASE 2012

Introducción a los Sist. Embebidos

Workshop mbed (turno mañana), Club de Robótica (FIUBA) (en colaboración con Rama Estudiantil IEEE FIUBA)

Miércoles 15, jueves 16 y viernes 17 de agosto de 9:00hs a 12:20hs, Aula L4, 1er piso

El objetivo es familiarizarse, de una forma entretenida, con la programación de microcontroladores de 32 bits en lenguaje C/C++ para sistemas embebidos, aprovechando la facilidad de uso de los kits de prototipado rápido. Para ello se desarrollará un proyecto relacionado con la navegación de un robot en modalidad autónomo y/o radio-controlado.

Durante tres días los participantes trabajarán en forma intensiva con el kit mbed LPC1768.

- Día 1: Introducción al mbed y a las herramientas de desarrollo, manejo de puertos y periféricos. Entrega de los vehículos y explicación de las bibliotecas de control.

- Día 2: Programación del control del vehículo y primeras pruebas.

- Día 3: Finalización de la programación de los robots y competencia entre todos los grupos participantes.

Requisitos: conocimientos de programación en lenguaje C; nociones de programación orientada a objetos.

Para más información ver: <http://www.clubderobotica.com.ar/>

LPCXpresso, Juan Cruz et al (UTN-FRBA/FIUBA)

Miércoles 15, jueves 16 y viernes 17 de agosto de 9:00hs a 12:20hs, Aula Egriet, 3er piso

Se presentará una introducción práctica a los microcontroladores LPC17XX del fabricante NXP basados en la tecnología Cortex-M3 de ARM. Se utilizará hardware el "target LPC1768/69" (que incluye un "JTAG debugger" que permite depurar los programas en tiempo real) y el "LPC1769 LPCXpresso board" (un entrenador que incluye periféricos como acelerómetros, sensores de luz, USB, ethernet, etc.). Se utilizará el IDE gratuito de "Code Red" basado en una versión del IDE "Eclipse" y se explorarán diferentes características del microcontrolador LPC17XX como manejo de GPIOs, manejo de módulos UART así como también la temática de arquitectura de software aplicada a los microcontroladores.

Se requieren conocimientos previos básicos sobre programación en el lenguaje C y microcontroladores.

Introducción a la familia Cortex M de ST, Ingeniero de ST (Arrow Argentina)

Miércoles 15 de agosto 9:00hs a 17.10hs, Aula L3, 1er piso

En el tutorial se hará un recorrido por las familias ARM Cortex M3 y M4 que posee ST Microelectronics, en especial la familia M4 con y su entorno integrado de trabajo y las herramientas disponibles para esta familia. Se harán aplicaciones prácticas sobre el kit STM32VLDISCOVERY orientadas al control industrial y otras aplicaciones.

Freescall: Hands on Kinetics Cortex M4, (Electrocomponentes)

Miércoles 15 de agosto 9:00hs a 12:20hs, Aula L14, 1er piso

Arquitectura Cortex M4. Porfolio de productos Freescall. Conceptos de sistema operativo. Sistema operativo MQX.

Aplicaciones. Herramientas de programación y entornos de desarrollo. Desarrollo de aplicaciones de manejo de puertos de entrada y salida, manejo de interrupciones, puerto serie y otros.

Se requieren conocimientos básicos de microcontroladores y de programación en lenguaje C.

Workshop mbed (turno tarde), Club de Robótica (FIUBA) (en colaboración con la Rama Estudiantil IEEE FIUBA)

Miércoles 15, jueves 16 y viernes 17 de agosto de 13:50hs a 17:10hs, Aula L4, 1er piso

Descripción: Idem Workshop mbed (turno mañana)

Programación en C para sistemas embebidos sobre MSP430, Guillermo Friedrich (UTN-FRBB)

Jueves 16 de agosto 9:00 a 17:10hs, Aula L11, 1er piso

Se presentará una breve introducción a la arquitectura de los microcontroladores MSP430 de Texas Instruments, en especial de aquellos elementos sobre los que se va a trabajar durante el workshop. La práctica se realizará sobre el kit Launchpad (que por su sencillez -se usa el puerto USB para alimentación, grabación y "debugging"- y bajo precio -US\$ 7-, es una excelente alternativa para iniciarse en el mundo de los sistemas embebidos). Se usará el IDE Code Composer Studio v5, que se puede descargar de la página de Texas Instruments, y que para los modelos de micros que soporta el kit Launchpad no tiene limitaciones en el tamaño del código. Se verán particularidades y buenas prácticas para la programación en C de un microcontrolador. Se explorarán distintos recursos del MSP430 (reloj, puertos de entrada/salida, temporizadores, interrupciones, puertos para comunicación serie, conversor A/D y controlador de flash). Se experimentará con la escritura en memoria flash, orientado a la posibilidad de actualizar el firmware (grabando, por ejemplo, una imagen descargada desde algún sitio externo). Se presentará en forma práctica la instalación y configuración de un "toolchain" GNU para MSP430, combinado con Code::Blocks (para Win32) como IDE. Se requieren conocimientos previos básicos sobre programación en lenguaje C y microcontroladores.



Introducción a los Sist. Embebidos (continuación)

Hands-on mbed crash-course - A platform for rapid prototyping, Joe Bungo (ARM, USA)

Jueves 16 de agosto de 9:00hs a 12:20hs, Aula L3, 1er piso

With the falling costs and increasing complexities of processors, microcontrollers are becoming cheaper, more powerful and interactive. They are now truly solutions looking for problems, where anyone could conceive a microcontroller application. The problem until now has been turning the idea into a prototype quickly and experimenting with the technology. ARM has changed this with Mbed, a rapid prototyping platform designed to simplify getting started with microcontrollers. Using a web based compiler and a very simple drag-and-drop interface, applications are developed without the need for expensive tools, and a new user can write and execute a "hello world" program in about sixty seconds.

Atmel: Hands on Cortex M3, (Electrocomponentes)

Jueves 16 de agosto 9:00hs a 12:20hs, Aula L14, 1er piso

Comparativa de arquitecturas ARM. Porfolio de productos ARM de ATMEL. Herramientas de desarrollo. AVR Studio 6 / Framework. Instalación del entorno. Generación de un proyecto. Compilación y debugging de una aplicación. Desarrollo de aplicaciones de manejo de puertos de entrada y salida, manejo de interrupciones, puerto serie y otros. Se requieren conocimientos básicos de microcontroladores y de programación en lenguaje C.

Freescale: Hands on Kinetics Cortex M4, (Electrocomponentes)

Viernes 17 de agosto 9:00hs a 12:20hs, Aula L14, 1er piso

Arquitectura Cortex M4. Porfolio de productos Freescale. Conceptos de sistema operativo. Sistema operativo MQX. Aplicaciones. Herramientas de programación y entornos de desarrollo. Desarrollo de aplicaciones de manejo de puertos de entrada y salida, manejo de interrupciones, puerto serie y otros. Se requieren conocimientos básicos de microcontroladores y de programación en lenguaje C.

Hands-on NXP Cortex-M0 LPC1114 en package DIP28, Sergio Scaglia (NXP/FIUBA)

Jueves 16 de agosto de 13:50hs a 17:10hs, Aula L3, 1er piso

Se realizara una presentación del ARM University Program, seguido por una introducción de la familia de microcontroladores Cortex-Mx, focalizando el Cortex-M0, su diversidad de packaging ofrecidos, y sus areas de aplicación. Se presentaran las características fundamentales que los diferencian sobre los de 8/16 bits, y se introducirá los nuevos dispositivos de Low-pin packages. Se continuara con el Hands-on "Diseño con Cortex-M0 en DIP28 package", donde se utilizara una breadboard para conectar el microcontrolador con otros elementos, y se creara una aplicación en LPCXpresso para verificar el funcionamiento del mismo.

Librerías Gráficas con PIC32, Mauricio Jancic (Artimar) / Cesar Fuoco (Elemon)

Viernes 17 de agosto 9:00 a 17:10hs, Aula L11, 1er piso

Introducción al uso de las librerías gráficas de Microchip orientado a la programación en C de GUIs (Graphical user interface) y a la implementación de hardware en displays gráficos touchscreen. Correremos los programas en una placa controladora basada en PIC32 + Controlador de Display externo SSD1963. IDE MPLABX compilador XC32. Todos los materiales se encontraran en los puestos de trabajo. Agenda: Graphics Overview Graphics Library Primitive Layer Lab 1 –GUI Splash Screen Drawing Widgets. Lab 2 –Creating a Simple Icon Menu. Interfacing the User (Message Interface). Lab 3 –User Interface using Message Callback. Advanced Features (Drawing Callback). Lab 4 –User Interface using Draw Callback. Putting it All Together (Multiple Screens). Lab 5 –Full Application

FPGAs y HDLs

Workshop FPGA (turno mañana), Cristian Sisterna (UNSJ/C7T) y Esteban Pelaez et al (UNSL)

Miércoles 15, jueves 16 y viernes 17 de agosto de 9:00hs a 12:20hs, Aula L9, 1er piso

Descripción: Introducción a VHDL. Descripción de circuitos digitales usando VHDL Introducción a FPGA. Características más importantes. Uso de software Quartus II (Altera) para Compilar, Sintetizar y Configurar un sistema digital en un FPGA. Uso de un board con FPGA para desarrollo de Laboratorios y Proyecto final.

Workshop FPGA (turno tarde), Cristian Sisterna (UNSJ/C7T) y Esteban Pelaez et al (UNSL)

Miércoles 15, jueves 16 y viernes 17 de agosto de 13:50s a 17:10hs, Aula L9, 1er piso

Descripción: Idem turno mañana



Linux Embebido

Linux en Beagleboard, Lucas Chiesa et al (FIUBA)

Miércoles 15, jueves 16 y viernes 17 de agosto de 13:50hs a 17:10hs, Aula Egriet, 3er piso

La Beagleboard es una plataforma de alta performance y bajo consumo basada en plataforma OMAP3 de Texas Instruments. El soporte y la comunidad que se generó detrás de ella la hace ideal para experimentar con Linux embebido. Durante el taller se presentarán los fundamentos de la placa y se introducirán los conceptos básicos necesarios para armar distribuciones de Linux para dispositivos embebidos, para poder finalmente subir una imagen y bootear Linux en el OMAP3. Requisitos: Manejo básico de cualquier distribución de linux.

Protocolos y comunicaciones

Telit: Programación sobre módulos celulares, (Electrocomponentes)

Miércoles 15 de Agosto 13:50hs a 17:10hs, Aula L14, 1er piso

Breve introducción a las tecnologías 2G y 3G. Presentación de los módulos celulares y los distintos entornos de programación (C, Python, Java). Aplicaciones. Herramientas de programación y entornos de desarrollo. Desarrollo de aplicaciones en "C" sobre módulos Telit.

Se requieren conocimientos básicos de programación en C y de comunicaciones GSM/GPRS.

Comunicación de aplicaciones mediante ZigBee, Sergio Caprile (CIKA)

Jueves 16 de agosto 13:50hs a 17:10hs, Aula L14, 1er piso

Características principales del stack de protocolos ZigBee y forma de utilización de los módulos XBee ZB programables para comunicación de aplicaciones entre sistemas dedicados. Se requieren conocimientos de programación en C y sistemas de comunicaciones

Comunicación de aplicaciones TCP/IP mediante Wi-Fi con Rabbit, Sergio Caprile (CIKA)

Viernes 17 de agosto 13:50hs a 17:10hs, Aula L14, 1er piso

Arquitectura de módulos Rabbit RCM5600W y utilización en aplicaciones típicas en sistemas dedicados con conectividad TCP/IP. Se requieren conocimientos de programación en C y stack TCP/IP

Software Embebido

RKH: Framework para statecharts, Leandro Francucci (UNMDP)

Miércoles 15 de agosto 13:50hs a 17:10hs, Aula L11, 1er piso

Este workshop utiliza ejercicios prácticos e interactivos combinados con instrucción para mostrar los conceptos subyacentes del framework RKH y como se aplican en el desarrollo de software para embedded systems. Se pretende ayudar a los participantes al cambio del paradigma de la programación secuencial tradicional a la programación dirigida por eventos y el desarrollo de software basado en un framework. Ofreciendo una introducción a la modelización de software mediante diagramas de secuencias y Statecharts, como así también, al modelo computacional RTC, eventos asíncronos, cola de eventos, patrones de diseño y el rol de las máquinas de estados. Para obtener los mejores resultados los participantes deben tener los conocimientos de haber trabajado en C, también como los conocimientos básicos en la programación de embedded systems.

Arquitectura de microprocesadores

Arquitectura MIPS, base de los microprocesadores modernos, Fabian Vargas (PUCRS, Brasil)

Viernes 17 de agosto de 9:00hs a 12:20hs, Aula L3, 1er piso

MIPS (originally an acronym for Microprocessor without Interlocked Pipeline Stages) is a reduced instruction set (RISC) computer architecture developed by MIPS Technologies. MIPS implementations are primarily used in embedded systems such as Windows CE devices, routers, residential gateways, and video game consoles such as the Sony PlayStation 2 and PlayStation Portable. MIPS implementations were also used by Digital Equipment Corporation (NEC), Pyramid Technology, Siemens Nixdorf, Tandem Computers and others during the late 1980s and 1990s. In the mid to late 1990s, it was estimated that one in three RISC microprocessors produced was a MIPS implementation. Considering such strong success experienced by the MIPS architecture in the market in the last decades, this course deals with presenting an overview about the MIPS architecture, briefly describing the main features involved in the design of its internal functional blocks, namely the data and control paths, the pipeline and the memory system.



Robótica

Robótica e Inteligencia Artificial, Claudio Verrastro et al (CNEA/UTN-FRBA)

Viernes 17 de agosto de 13:50hs a 17:10hs, Aula L3, 1er piso

Primera parte: "Implementación FPGA para Manipuladores robóticos y compiladores GNU de programas". Introducción a la Cinemática y Dinámica de los manipuladores robóticos. Implementación en FPGA del control del robot. Introducción y desarrollo de compiladores GNU para robots de N grados de libertad configurables.

Segunda parte: "Sistemas Operativos en Tiempo Real para Robots". Introducción a los Sistemas Operativos para Robots. Ejemplo de Implementación de FreeRTOS en ARM7 para MiniPi. Implementación de un sistema de control Difuso para Sumo de Robots.

Conocimientos previos : Conceptos básicos de ARM, FPGA, Análisis Matemático y Física. Programación C y Matlab.

Implementación de Sistemas Embebidos

Desarrollando proyectos electrónicos con Proteus, Israel Rodrigues (EDASIM, Brasil)

Miércoles 15 de agosto 9:00hs a 12:20hs, Aula L11, 1er piso

Descripción: ¿Qué es un Proyecto Electrónico? Fases de un Proyecto Electrónico con Proteus. Overview Proteus.

Desarrollo de Un proyecto con Proteus - Fase I Requisitos del Proyecto - Fase II Esquemático - Fase III PCB (Layout) - Fase IV Co-simulación con uC - Fase V Gerber