

ANEXO I – Res.C.D.Nº022/15

**UNIVERSIDAD NACIONAL DE
RÍO CUARTO
FACULTAD DE INGENIERÍA**



**DIPLOMATURA SUPERIOR
– SISTEMAS EMBEBIDOS –
Dpto. de Telecomunicaciones**

Índice

- 1.0 Relevancia del Proyecto**
- 2.0 Correspondencia con los fines y objetivos de la Universidad Nacional de Río Cuarto y la Facultad de Ingeniería**
- 3.0 Justificación de la Diplomatura**
- 4.0 Objetivos de la Diplomatura**
- 5.0 Características de la Diplomatura**
- 6.0 A quienes va dirigido**
- 7.0 Perfil del egresado**
- 8.0 Diseño Curricular y carga horaria**
- 9.0 Financiamiento del Proyecto**
- 10 Coordinación del Proyecto**

1. Relevancia del Proyecto

El nivel de desarrollo y la calidad de vida de nuestra sociedad dependen, sustancialmente, de la interacción de factores y agentes tales como estado y sociedad; producción y empleo; oferta y demanda; crecimiento económico y desarrollo social; economía y medio ambiente; desarrollo científico tecnológico y su transferencia e impacto social; así como de la capacidad de reorganización y transformación continua de las estructuras para aprovechar ventajas y oportunidades, minimizando efectos negativos.

Las necesidades sociales no deben ser ajenas a ninguna institución dedicada a la formación de conocimiento. Es por ello que nuestra institución ha asumido un fuerte compromiso con el objetivo de lograr una profunda integración y compromiso social en la región central del país.

Las características de la demanda de personal capacitado evolucionan continuamente en la región de influencia de la FI-UNRC. Por lo tanto la oferta académica de la FI contempla los siguientes aspectos principales:

- mejoramiento y adecuación continua de las ofertas de grado y posgrado, articuladas con el desarrollo científico-tecnológico y las necesidades de nuestra sociedad;
- generación de nuevos conocimientos orientados a potenciar la utilización racional de los recursos naturales, acorde a las necesidades propias del ámbito geográfico, económico y social;
- formación de profesionales capacitados para generar, interpretar, desarrollar y aplicar nuevos conocimientos y tecnologías, organizar nuevas formas asociativas y de vinculación producción-sociedad;
- vinculación y articulación de la investigación con el medio socio-productivo, de forma tal que las demandas sean rápidamente interpretadas, atendidas y resueltas

En este marco, y atendiendo a que es una función básica de las Instituciones Universitarias promover y desarrollar la investigación científica y tecnológica (Ley de Educación Superior N° 24.521; Título IV, Capítulo 1, Artículo 28, inciso b), así como la de extender su acción y sus servicios a la comunidad, con el fin de contribuir a su desarrollo y transformación, estudiando en particular los problemas nacionales y regionales, prestando asistencia técnica al Estado y a la comunidad (Ley de Educación Superior N° 24.521; Título IV, Capítulo 1, Artículo 28, inciso e), es que se realizó la presente propuesta.

2. Correspondencia con los fines y objetivos de la Universidad Nacional de Río Cuarto y la Facultad de Ingeniería

Esta Diplomatura Superior, correspondiéndose con los objetivos de la Universidad Nacional de Río Cuarto, (UNRC), y de la Facultad de Ingeniería, (FI), apunta fundamentalmente a ofrecer una propuesta educativa de cuarto nivel que se inserte en el Programa de Posgrado de la FI y que esté dirigida a satisfacer las demandas dentro de

un área específica del conocimiento científico y/o de extensión o articulación con el medio social.

La UNRC considera, tal como queda plasmado en la Res. C.S 270/11 que *“la creación de las diplomaturas superiores se fundamenta en el interés que se suscita en las propuestas educativas de cuarto nivel, que ofrecen formación intensiva en períodos relativamente breves desarrollados en cursos y/o seminarios que se articulen en torno a una temática específica”*.

Por su parte la FI en el Plan Trienal de Desarrollo Institucional 2014-2017 (aprobado por resolución de Consejo Directivo 225/14) prevé en sus lineamientos consolidar a la FI-UNRC como centro de excelencia de formación continua y de Posgrado, considerando a dicha formación como una actividad de fundamental importancia para el perfeccionamiento y formación de graduados y docentes. Establece además, como objetivo específico en el área de posgrado la *“implementación de carreras cortas de formación específica direccionadas a profesionales de carreras afines a las desarrolladas en la Facultad”*.

Con el mismo espíritu en su política de Extensión y Vinculación con el Medio y Cooperación Internacional, propone como una de sus líneas de acción *“promover el dictado de cursos de capacitación y/o actualización, dirigido a profesionales y técnicos que requieren la formación y/o actualización en conocimientos específicos asociados a las actividades que desempeñan en empresas privadas y/u organismos públicos.”*, como así también en las políticas para Graduados, en donde establece como una de las líneas de acción *“Implementación de programas integrales de formación continua para graduados: esto mediante la implementación de Carreras cortas con competencia en áreas específicas y estratégicas”*.

En el mismo sentido el Dpto. de Telecomunicaciones presentó en el año 2010 un Plan de Acciones Integral, aprobado por Resolución CD N° 149/10, el cual está estructurado en programas. Cada uno de estos tiene como objetivo organizar de manera clara las actividades a realizar de forma tal de aportar al fortalecimiento estructural de la carrera. En el Programa de “Vinculación con el Medio” se plantea como uno de los objetivos el llevar adelante todo tipo de acción que promueva la formación continua de los graduados, entendiendo que las mismas tienden a sostener y/o profundizar las dimensiones definidas en la Resolución ME N° 1456/06 (Anexo IV inciso I.8 y IV.7).

De todo lo expuesto anteriormente se desprende que la creación de una Diplomatura Superior dirigida a promover una formación de cuarto nivel en un área específica del conocimiento científico-tecnológico, como es el Diseño de Sistemas Embebidos, se corresponde en un todo con los objetivos institucionales tanto de la UNRC, como de la FI y su departamento de Telecomunicaciones

3. Justificación de la Diplomatura Superior en Sistemas Embebidos

En la región aledaña a la Universidad Nacional de Río Cuarto, la industria y la oferta de servicios con desarrollo nacional tiene un importante auge a partir de las nuevas políticas Nacionales (Plan Estratégico Industrial 2020 y el Plan Estratégico Agroalimentario y Agroindustrial.). En este contexto, la ingeniería es una disciplina fundamental para lograr consolidar el desarrollo industrial, relacionar conocimiento con

innovación productiva, y disminuir los niveles de dependencia tecnológica. Es por este motivo que el área de conocimiento de los Sistemas Embebidos se constituye, en una pieza fundamental para este tipo de políticas.

La Facultad de Ingeniería, más precisamente el Departamento de Telecomunicaciones a través de sus grupos de investigación: Grupo de Investigación y Desarrollo Aplicado a las Telecomunicaciones (GIDAT) y Grupo de Sistemas de Tiempo Real (GSTR), se está consolidando como un lugar fuerte para el diseño y desarrollo de sistemas embebidos.

Los Sistemas Embebidos son un aspecto esencial de nuestra vida cotidiana, en todos los ámbitos, desde los sistemas de transporte (aeronáutica, espacio, carretera, ferrocarril y marítimo), a los sectores de energía y también para los sistemas de comunicación (por ejemplo, *smartphones*).

Sistemas Embebidos es el nombre genérico que reciben los equipos electrónicos que realizan el procesamiento de datos, pero que a diferencia de una computadora personal, están diseñados para satisfacer una función específica, como en el caso de un teléfono celular, el sistema de control de un automóvil, de un satélite o de una maquina agrícola. Es un sistema electrónico que esta contenido ("embebido") dentro de un equipo completo que incluye por ejemplo, partes mecánicas y electromecánicas.

Actualmente hay un fuerte interés por emprendedores y empresas de la región en incorporar tecnología a sus procesos, productos y/o servicios a través de Sistemas Embebidos.

4. Objetivos del Proyecto

El desarrollo de sistemas embebidos requiere de un amplio conocimiento en las tecnologías disponibles en la actualidad, así como también criterios de diseño que permitan abordar la solución en tiempos acordes a los ciclos del mercado actual, manteniendo una alta calidad y prestación demandada por los usuarios actuales.

Objetivo general:

- Formar al estudiante en las tareas de programar, diseñar y desarrollar sistemas embebidos.

Objetivos específicos

- Aprender en detalle sobre los componentes que conforman un sistema embebido.
- Seleccionar, configurar y programar a través de herramientas específicas, un sistema embebido.

5. Características del proyecto

- A. Nivel: Diplomatura Superior
- B. Modalidad: Presencial

Quienes completen el cursado y aprobación recibirán el título de “Diplomado Superior en Diseño de Sistemas Embebidos” emitido por la Universidad Nacional de Río Cuarto.

6. A quienes va dirigido

La Diplomatura está dirigida a un espectro amplio de profesionales, no sólo ingenieros de cualquier especialidad, sino también a licenciados, afines a la electrónica y la programación. Tal como se contempla en la Res. CS 270/11 también podrán aspirar al cursado de esta propuesta de posgrado los egresados de nivel superior no universitario cuya formación resulte compatible con las exigencias de la carrera propuesta.

La Diplomatura complementa, en general, la formación de grado que reciben los estudiantes de disciplinas afines, ya que difícilmente se reúne en una sola orientación de Ingeniería la formación transversal que requiere el desarrollo de Sistemas Embebidos.

7. Perfil del egresado

El egresado tendrá la capacidad de identificar nuevas necesidades de desarrollo, así como también, el mejoramiento de sistemas embebidos. Podrá seleccionar la tecnología más conveniente y aplicar criterios de diseño electrónico para abordar la solución, logrando resultados de muy alto nivel tecnológico.

Los criterios de diseño electrónico abarcan desde la elección de los componentes más convenientes, hasta las herramientas de software que permiten desarrollar los programas que se ejecutan en el sistema embebido.

Además, el egresado incorporará conocimientos y competencias que le permitirán llevar adelante la solución en el marco de un emprendimiento propio, potenciando así el desarrollo de futuras nuevas empresas en la región.

8. Estructura curricular y carga horaria

Los cursos incluidos en la currícula son todos de régimen mensual, pensados para ser desarrollados en un mínimo de cuatro encuentros de cinco horas cada uno, donde se deberían incluir todas las evaluaciones necesarias para que el estudiante apruebe los cursos. La modalidad de los mismos son presenciales y de contenido Teórico/Práctico.

Los encuentros se organizarán los viernes por la tarde de 16 a 21 hs o sábado por la mañana de 9 a 14 hs según disponibilidad del grupo de estudiantes. El programa académico de la Diplomatura contempla un total de 180 horas de cursos, talleres y/o seminarios, distribuidas en 3 módulos curriculares:

Modulo I. Software

1. Programación embebida: C.
2. Programación embebida: Python.
3. Sistemas operativos de tiempo real (RTOS).
4. Linux Embebido.

5. Desarrollo de Aplicaciones para Android.
6. Aplicaciones de red en sistemas embebidos.

Modulo II. Hardware

7. Plataformas de Desarrollo: Arduino I
8. Plataformas de Desarrollo: Arduino II
9. Plataformas de Desarrollo: Intel Galileo
10. Plataformas de Desarrollo: EduCIAA
11. Interfaces y Perifericos

Módulo III. Seminarios y Talleres de aplicaciones y actualización

12. Seminario de aplicación/actualización I
13. Seminario de aplicación/actualización II
14. Taller de Trabajo Final Integrador

Para obtener el título de Diplomado Superior deberán completarse 180 horas de cursado, a elección del alumno, distribuidos de manera tal que se acrediten 60 horas mínimas de cursado en cada módulo.

9. Financiamiento del Proyecto

Para la puesta en marcha de la Diplomatura Superior es necesario poder contar con el hardware en número suficiente para la realización adecuada de las actividades prácticas previstas en cada uno de los cursos planteados. Por tal motivo la apertura de inscripción a una cohorte de estudiantes, queda supeditada a la realización de posibles convenios de financiamiento con organismos públicos, organismos privados y/o cualquier otra institución que garantice al menos el recurso de hardware para diez (10) estudiantes. Para los potenciales estudiantes que no estén contemplados en los mencionados convenios de financiamiento, la diplomatura tendrá un costo total de \$ 9.000⁰⁰. En caso de querer realizar solo algunos cursos, estos tendrán un costo por curso de \$ 1.000⁰⁰.

10. Coordinación

De acuerdo con la reglamentación vigente en la UNRC, la diplomatura considera la siguiente estructura de coordinación:

Coordinador General: Mg. Gustavo Rodríguez

Coordinador Adjunto: Mg. Sebastián Tosco

Coordinador Logístico: Mg. Fernando Corteggiano

Personal de apoyo administrativo: a designar.

**UNIVERSIDAD NACIONAL DE
RÍO CUARTO
FACULTAD DE INGENIERÍA**

**Anexo I
Contenidos mínimos Cursos, Seminarios
y Talleres**

CODIGO CURSO: CDIP.SE 01

DENOMINACIÓN: Programación embebida: C.

RESPONSABLE: Mg. Manuel AMOR

CONOCIMIENTOS O REQUERIMIENTOS PREVIOS: Dominio básico de Linux.

OBJETIVOS:

- Adquirir la capacidad de programar aplicaciones embebidas en lenguaje C.
- Conocer las posibilidades implementación de comunicación a través de diferentes interfaces en aplicaciones escritas en C.
- Entender la importancia de las buenas prácticas de programación.
- Llevar a cabo ejercicios prácticos que permitan la aprehensión de los conceptos teóricos.

JUSTIFICACIÓN:

Hoy en día está muy difundido el uso del lenguaje C en aplicaciones de todo tipo. Se ha extendido profundamente en el en el ámbito de la Ingeniería en general y en la de los sistemas embebidos en particular. Los sistemas embebidos han tomado un gran protagonismo en la actualidad por los avances tecnológicos que se han ido sucediendo. Aplicaciones de agricultura, de medicina, de aeronáutica, del espacio, etc, tienen lugar en los sistemas embebidos. Es por ello que se considera fundamental que el ingeniero a cargo tanto del diseño como de la enseñanza de sistemas embebidos incorpore el dominio del lenguaje C como herramienta necesaria y fundamental para el éxito de sus proyectos.

DURACIÓN Y ORGANIZACIÓN:

20 hs. Cuatro clases de 5 hs, 1 clase por semana

METODOLOGÍA DE DICTADO:

La metodología de dictado será TeóricoPráctica. Se desarrollarán los temas con el apoyo de presentaciones multimedia, ejemplos funcionales que se demostrarán en el momento y ejercicios a realizar por los asistentes.

METODOLOGÍA DE EVALUACIÓN Y APROBACIÓN:

Para la aprobación del curso se requiere la asistencia al 80% de las clases y la realización de los ejercicios de evaluación.

MATERIAL QUE SE OFRECE AL ASISTENTE:

Material didáctico y códigos usados para el dictado del curso.

CANTIDAD MÍNIMA Y MÁXIMA DE ASISTENTES:

La cantidad máxima de asistentes será de 20 (veinte) participantes.

NECESIDADES Y DISPONIBILIDAD DE INFRAESTRUCTURA:

Se requiere de un aula con capacidad para 20 personas, 20 computadoras (preferentemente conectadas en red) y 1 proyector.

Software Necesario:

- Linux (preferentemente debian o ubuntu)
- GNU Toolchain

Hardware Necesario:

- PCs estándar

CONTENIDOS:

UNIDAD 1: Tipos y/o Funciones Básicas del lenguaje C ANSI. Funciones, procedimientos, pasajes de parámetros. Aritmética de punteros (básico). Memoria Estática, Memoria Dinámica. Ciclos, bloques de programación, alcance (scope) y tiempo de vida de datos. Expresiones aritméticas. Estructuras de datos: Pilas, Colas, Listas

UNIDAD 2: Conceptos generales de Entrada/Salida. Entrada/Salida por archivos. Entrada/Salida por comunicación de datos básica: RS232, TCP/IP y UDP. Pipes.

UNIDAD 3: Concurrencia. Concepto de Threads: Biblioteca Poxi Threads (No ANSI). Creación de procesos con llamada de sistema fork(). Mecanismos para sincronización de datos y exclusión mutua: Semáforos, Mutex. Problemas de DeadLock, y Race Condition.

UNIDAD 4: Librerías. Creación de librerías estáticas y dinámicas. Interfaces de librerías, ocultamiento de información.

BIBLIOGRAFÍA:

G.L.U.P. Guía de Linux Para el Usuario. Larry Greenfield 1997.

Linux Installation and Getting Started. MattWelsh et al. 2002.

Running Linux MattWelsh et al. O'Reilly Associates, Inc. 1999.

El lenguaje de programación C. Kernighan, Brian, Ritchie, Dennis. Prentice Hall 1988

Topics in C Programming. Stephen Kochan, Patrick Wood. John Wiley & Sons. 1991.

CODIGO CURSO: CDIP.SE 02

DENOMINACIÓN: Introducción al Lenguaje de Programación Python para la ingeniería

RESPONSABLE: Mg. Sebastián TOSCO

DOCENTES PARTICIPANTES: Mg. Fernando CORTEGGIANO

CONOCIMIENTOS O REQUERIMIENTOS PREVIOS: Se deberá tener conocimientos básicos previos sobre conceptos generales de programación.

OBJETIVOS:

- Brindar las herramientas y conocimientos básicos para que los asistentes puedan desempeñarse e interiorizarse en el lenguaje de programación Python, en particular en lo que respecta a la programación orientada a aplicaciones de ingeniería.

JUSTIFICACIÓN:

Python es un lenguaje de sintaxis simple, clara y sencilla. La disponibilidad de librerías para múltiples aplicaciones científicas y técnicas, hace que sea una herramienta importante para su implementación en el ámbito de la ingeniería. No se requiere la compra de licencia para la utilización de este lenguaje.

DURACIÓN Y ORGANIZACIÓN:

20 hs. Cuatro clases de 5 hs, 1 clase por semana

METODOLOGÍA DE DICTADO:

La metodología de dictado será Teórico-Práctica. Se desarrollarán los temas con el apoyo de presentaciones multimedia, ejemplos funcionales que se demostrarán en el momento y ejercicios a realizar por los asistentes.

METODOLOGÍA DE EVALUACIÓN Y APROBACIÓN:

Para la aprobación del curso se requiere la asistencia al 80% de los módulos y la realización de un trabajo final con defensa en seminario.

MATERIAL QUE SE OFRECE AL ASISTENTE:

Material didáctico y códigos usados para el dictado del curso.

CANTIDAD MÍNIMA Y MÁXIMA DE ASISTENTES:

La cantidad máxima de asistentes será de 20 (veinte) participantes.

NECESIDADES Y DISPONIBILIDAD DE INFRAESTRUCTURA:

Se requiere de un aula con capacidad para 20 personas, 20 computadoras (preferentemente conectadas en red) y 1 proyector.

Software Necesario:

- Linux (preferentemente debian o ubuntu)

- Sistema Operativo: Windows ó Linux (preferentemente Linux)
- Python 2.7
- Numpy
- Scipy
- pylab
- matplotlib

Hardware Necesario:

- PCs estándar

CONTENIDOS:

UNIDAD 1: Python. Características Principales. Módulos. Uso del Intérprete. Uso de dir() y help()

UNIDAD 2: Tipos de datos y sus propiedades. Enteros, de punto flotante, Complejos. Cadenas de texto. Tuplas y listas. Conjuntos. Diccionarios.

UNIDAD 3: Control de flujo. Instrucciones if, for, while, try – except.

UNIDAD 4: Funciones. Definición de una función. Utilización. Pasaje de parámetros.

UNIDAD 5: Clases. Definición. Utilización de clases en Python.

UNIDAD 6: Manejo de archivos en Python. Objeto indentificador de archivo. Sentencias de apertura y cierre de archivos. Ejemplos de uso.

UNIDAD 7: Utilización de herramientas para graficación y cálculo. Uso de matplotlib y pylab. Ejemplos de uso.

UNIDAD 8: Sockets. Definición. Tipificación. Usos comunes. Ejemplos.

UNIDAD 9: Interfaz gráfica (GUI). Justificación de uso. Librerías disponibles. Ejemplos.

UNIDAD 10: Trabajo final. Organización de un proyecto. Desarrollo del trabajo final. Seminario de exposición

BIBLIOGRAFÍA:

Tutorial de Python – <http://python.org.ar/tutorial>

Documentación Oficial de Python – <http://docs.python.org/>

Matplotlib <http://matplotlib.sourceforge.net/>

PyQT <https://wiki.python.org/moin/PyQt/Tutorials>

WxGlade <http://wxglade.sourceforge.net/>

CODIGO CURSO: CDIP.SE 03

DENOMINACIÓN: Sistemas Operativos de Tiempo Real

RESPONSABLE: Mg. Gustavo RODRIGUEZ

CONOCIMIENTOS O REQUERIMIENTOS PREVIOS: Conocimientos básicos de programación y sistemas operativos

OBJETIVOS:

- Incorporar elementos de programación utilizando RTOS, como extensión superadora de la programación bare-metal(sin RTOS).
- Discernir la diferencia entre RTOS dinámicos y estáticos, y en qué casos es conveniente la utilización de cada tipo.
- Entender los motivos de las diferentes políticas de scheduling que implementan los RTOS a diferencia de los Sistemas Operativos de propósito general.
- Llevar a cabo ejercicios prácticos que permitan la aprehensión de los conceptos teóricos.

JUSTIFICACIÓN:

Hoy en día existen muy pocas aplicaciones que empleen sistemas embebidos basados en microcontroladores de 32-bits que no utilicen un Sistema Operativo de Tiempo Real (RTOS, por sus siglas en inglés). Un RTOS puede verse básicamente como una biblioteca de software que es aprovechada por el código del usuario a fin de sumar ciertas características al diseño de su aplicación, como ser:

- Multitarea cooperativa y/o expropiativa.
- Política de scheduling de procesos orientada a satisfacer requerimientos de respuesta en tiempo real.
- Gestión dinámica de memoria.
- Mecanismos de comunicación entre procesos.

Dichas características, entre otras, dotarán a la aplicación embebida de una confiabilidad superior, además de aumentar la portabilidad de su código a otras arquitecturas, dado que muchos RTOS disponen de Interfaces de Programación de Aplicaciones (API) basadas en estándares como POSIX. Es por ello que se considera fundamental que el ingeniero a cargo tanto del diseño como de la enseñanza de sistemas embebidos incorpore los conceptos fundamentales que serán explicados en el presente curso.

DURACIÓN Y ORGANIZACIÓN:

20 hs. Cuatro clases de 5 hs c/u. 1 clase por semana.

METODOLOGÍA DE DICTADO:

La metodología de dictado será Teórico Práctica. Se desarrollarán los temas con el apoyo de presentaciones multimedia, ejemplos funcionales que se demostrarán en el momento y ejercicios a realizar por los asistentes.

METODOLOGÍA DE EVALUACIÓN Y APROBACIÓN:

Para la aprobación del curso se requiere la asistencia al 80% de las clases y la realización de los ejercicios de evaluación.

MATERIAL QUE SE OFRECE AL ASISTENTE:

Material didáctico y códigos usados para el dictado del curso.

CANTIDAD MÍNIMA Y MÁXIMA DE ASISTENTES:

La cantidad máxima de asistentes será de 20 (veinte) participantes.

NECESIDADES Y DISPONIBILIDAD DE INFRAESTRUCTURA:

Se requiere de un aula con capacidad para 20 personas, 20 computadoras (preferentemente conectadas en red) y 1 proyector.

Software Necesario:

- Linux (preferentemente debian o ubuntu)
- GNU Toolchain
- qemu emulator

Hardware Necesario:

- NXP LPC1769

CONTENIDOS:

UNIDAD 1: Arquitecturas de hardware. Características generales. Modelo del programador. Modos de ejecución privilegiado (kernel) y no privilegiado (user). Modelo de Excepciones e Interrupciones. Controlador NVIC. Unidad de Protección de Memoria (MPU).

UNIDAD 2: Conceptos generales de Sistemas Operativos de Tiempo Real. Cambio de contexto y políticas de scheduling. Gestión de memoria. Gestión de interrupciones de hardware y excepciones del procesador. Comunicación entre procesos: Semáforos, Message Queues, Mutexs.

UNIDAD 3: Sistemas Operativos de Tiempo Real estáticos. Caso de estudio: OSEK-OS. Caso de estudio: POK.

UNIDAD 4: Sistemas Operativos de Tiempo Real dinámicos. Ventajas y desventajas respecto a los RTOS estáticos. Caso de estudio I: FreeRTOS. Caso de estudio II: RTEMS.

BIBLIOGRAFÍA:

ARMv7-M Architecture Reference Manual, <http://infocenter.arm.com/>

The Definitive Guide to ARM® Cortex®- M3 and Cortex®-M4 Processors, 3rd Edition. J. Yiu, Newnes (2013).

Using the FreeRTOS Real-Time Kernel”, NXP LPC17xx edition. R. Barry.

Implementación de un Kernel de Tiempo Real para Arquitectura ARMv7- M. P. Ridolfi et al. Congreso Argentino de Sistemas Embebidos (2013).

OSEK-OS 2.2.3 standard. <http://www.osek-vdx.org/>

CODIGO CURSO: CDIP.SE 04

DENOMINACIÓN: Linux Embebido

RESPONSABLE: Mg. Gustavo RODRIGUEZ

CONOCIMIENTOS O REQUERIMIENTOS PREVIOS: Conocimientos básicos de programación en C y Linux.

OBJETIVOS:

- Introducir a los alumnos en la arquitectura de un sistema embebido basado en Linux.
- Presentar las herramientas de software necesarias en el proceso de desarrollo.
- Presentar alternativas de plataformas de hardware.

JUSTIFICACIÓN:

Hoy en día abundan los sistemas operativos para un sistema embebido, tanto de código abierto como propietario. Linux es una de estas opciones. No importa lo que se utiliza para el entorno de desarrollo (Linux o Windows), es necesario aprender a programar utilizando el sistema operativo de destino. En este sentido, un Linux Embebido no es muy diferente de usar a un Sistema Operativo de Tiempo Real u otro sistema operativo. Es necesario comprender cómo el sistema operativo está diseñado, cómo configurarlo, y cómo programar usando su interfaz de programación de aplicaciones (API).

Es por ello que se considera fundamental que el ingeniero a cargo tanto del diseño como de la enseñanza de sistemas embebidos incorpore los conceptos fundamentales que serán explicados en el presente curso.

DURACIÓN Y ORGANIZACIÓN:

20 hs. Cuatro clases de 5 hs c/u. 1 clase por semana.

METODOLOGÍA DE DICTADO:

La metodología de dictado será Teórico Práctica. Se desarrollarán los temas con el apoyo de presentaciones multimedia, ejemplos funcionales que se demostrarán en el momento y ejercicios a realizar por los asistentes.

METODOLOGÍA DE EVALUACIÓN Y APROBACIÓN:

Para la aprobación del curso se requiere la asistencia al 80% de las clases y la realización de los ejercicios de evaluación.

MATERIAL QUE SE OFRECE AL ASISTENTE:

Material didáctico y códigos usados para el dictado del curso.

CANTIDAD MÍNIMA Y MÁXIMA DE ASISTENTES:

La cantidad máxima de asistentes será de 20 (veinte) participantes.

NECESIDADES Y DISPONIBILIDAD DE INFRAESTRUCTURA:

Se requiere de un aula con capacidad para 20 personas, 20 computadoras (preferentemente conectadas en red) y 1 proyector.

Software Necesario:

- Linux (preferentemente debian o ubuntu)
- GNU Toolchain

CONTENIDOS:

UNIDAD 1: Introducción a Linux Embebido. Ventajas. Hardware para Linux Embebido. Ejemplos de placas comerciales. Arquitectura básica de un sistema Linux.

UNIDAD 2: Entorno de desarrollo para Linux Embebido: Toolchain. Definición de toolchain. Formas de armar un toolchain. Componentes. Bibliotecas de C. Armandando un toolchain: construcción manual, toolchains precompilados, uso de herramientas de ayuda en el armado.

UNIDAD 3: Kernel de Linux. Breve historia. Características principales. Esquema de versiones. Fuentes del kernel y parches. Configuración del kernel: xconfig, gconfig, nconfig, menuconfig. Opciones principales de configuración. Cross-compilación e instalación del kernel.

UNIDAD 4: Bootloaders. Definición. Secuencia de boot en x86. Secuencia de boot en embebidos.

UNIDAD 5: Root filesystem. Definición. Montaje de un sistema de archivos. Network Filesystem. Organización de un filesystem. Archivos de dispositivos. Sistemas de archivos virtuales. Sistemas de archivos de bloque. Sistema de archivos Flash.

BIBLIOGRAFÍA:

Embedded Linux Primer: A Practical Real-World Approach, Christopher Hallinan, 2010, Prentice Hall.

Building Embedded Linux Systems, Karim Yaghmour, Jon Masters, Gilad Ben-Yossef y Philippe Gerum, 2008, O'Reilly.

CODIGO CURSO: CDIP.SE 05

DENOMINACIÓN: Desarrollo de Aplicaciones para Android

RESPONSABLE: Mg. Gustavo RODRIGUEZ

CONOCIMIENTOS O REQUERIMIENTOS PREVIOS: Conocimientos básicos de programación y sistemas operativos.

OBJETIVOS:

Al término del curso los participantes serán capaces de:

- Introducir a los alumnos en la plataforma de aplicaciones Android.
- Presentar las herramientas de software necesarias en el proceso de desarrollo.
- Presentar alternativas de hardware para distintas aplicaciones.

JUSTIFICACIÓN:

Hasta la fecha abundan diferentes Sistemas Operativos, pero ninguno tan sobresaliente como lo ha sido Android, que si bien es parte de la familia de Linux, es sobresaliente debido a que se desarrolló para dispositivos móviles con pantallas táctiles, como teléfonos inteligentes o tablets; y también para relojes inteligentes, televisores y automóviles.

Tiene una gran comunidad de desarrolladores creando aplicaciones para extender la funcionalidad de los dispositivos. A la fecha, se ha llegado ya al 1.000.000 de aplicaciones de las cuales, dos tercios son gratuitas.

Hoy en día abundan diferentes plataformas de desarrollo que trabajan bajo diferentes arquitecturas. Android es una de estas opciones. Ofrece un entorno de desarrollo de software de código abierto, complementa con hardware con diferentes aplicaciones y usos para ofrecer funciones informáticas mas avanzadas a quienes ya están familiarizados con otras herramientas.

Es por ello que se considera fundamental que el ingeniero a cargo de desarrollo de sistemas embebidos se instruya la Plataforma de Desarrollo Android e incorpore los conceptos fundamentales que serán explicados en el presente curso.

DURACIÓN Y ORGANIZACIÓN:

20 hs. Cuatro clases de 5 hs c/u. 1 clase por semana.

METODOLOGÍA DE DICTADO:

La metodología de dictado será Teórico Práctica. Se desarrollarán los temas con el apoyo de presentaciones multimedia, ejemplos funcionales que se demostrarán en el momento y ejercicios a realizar por los asistentes.

METODOLOGÍA DE EVALUACIÓN Y APROBACIÓN:

Para la aprobación del curso se requiere la asistencia al 80% de las clases y la realización de los ejercicios de evaluación.

MATERIAL QUE SE OFRECE AL ASISTENTE:

Material didáctico y códigos usados para el dictado del curso.

CANTIDAD MÍNIMA Y MÁXIMA DE ASISTENTES:

La cantidad máxima de asistentes será de 20 (veinte) participantes.

NECESIDADES Y DISPONIBILIDAD DE INFRAESTRUCTURA:

Se requiere de un aula con capacidad para 20 personas, 20 computadoras (preferentemente conectadas en red) y 1 proyector.

Software Necesario:

- Linux (preferentemente debian o ubuntu)
- Eclipse
- Emulador Android

CONTENIDOS:

UNIDAD 1: Introduccion Android. Fundamentos de Android y Conceptos de Programación. Tipos de Datos y Pruebas Logicas Bucles y Estructuras de Control.

UNIDAD 2: Interfaz Usuario. Componentes de desarrollo. Software de Instalación. Layouts en Android.

UNIDAD 3: Comunicación con Sensores. Sensores de Movimiento. Sensores Ambientales. Sensores de Posicion (Mapas y GPS).

UNIDAD 4: Productividad. Publicación de aplicaciones realizadas. Creación de Base de Datos.

BIBLIOGRAFÍA:

<https://www.android.com/>

<http://www.aprendeandroid.com/menu.htm>

Curso Android - Desarrollo de aplicaciones moviles - Versión 1 / junio 2011.

www.maestrosdelweb.com/editorial/curso-android.

Curso Programacion Android - Salvador Gomez Oliver - www.sgoliver.net.

Curso de programacion en android para principiantes. desarrollado por Robert P. para FaqAndroid.com del Grupo bemoob.

CODIGO CURSO: CDIP.SE 06

DENOMINACIÓN: Aplicaciones de red en sistemas embebidos

RESPONSABLE: Mg. Fernando CORTEGGIANO

CONOCIMIENTOS O REQUERIMIENTOS PREVIOS: Dominio básico de Linux y conceptos generales de programación en Python y C/C++.

OBJETIVOS:

Se pretende brindar las herramientas y conocimientos básicos para que los asistentes puedan entender y aplicar esquemas de comunicación en el contexto de los sistemas embebidos.

JUSTIFICACIÓN:

En sistemas embebidos que soportan sistemas operativos es posible utilizar herramientas que facilitan las comunicaciones entre dichos sistemas, permitiendo el desarrollo de aplicaciones de red y comunicaciones eficientes de fácil implementación.

DURACIÓN Y ORGANIZACIÓN:

20 hs. Cuatro clases de 5 hs c/u. 1 clase por semana.

METODOLOGÍA DE DICTADO:

La metodología de dictado será Teórico Práctica. Se desarrollarán los temas con el apoyo de presentaciones multimedia, ejemplos funcionales que se demostrarán en el momento y ejercicios a realizar por los asistentes.

METODOLOGÍA DE EVALUACIÓN Y APROBACIÓN:

Para la aprobación del curso se requiere la asistencia al 80% de las clases y la realización de los ejercicios de evaluación.

MATERIAL QUE SE OFRECE AL ASISTENTE:

Material didáctico y códigos usados para el dictado del curso.

CANTIDAD MÍNIMA Y MÁXIMA DE ASISTENTES:

La cantidad máxima de asistentes será de 20 (veinte) participantes.

NECESIDADES Y DISPONIBILIDAD DE INFRAESTRUCTURA:

Se requiere de un aula con capacidad para 20 personas, 20 computadoras (preferentemente conectadas en red) y 1 proyector.

Software Necesario:

- Sistema Operativo ARM: Linux (preferentemente Ubuntu)
- Python 2.7
- Compilador C/C++
- Librerías básicas
- Editor de Texto / IDEs. Ejemplos:
- Gedit
- Geany

- QtCreator / CodeBlocks

Hardware Necesario:

- PCs estándar
- Placa con hardware embebido con capacidad para sistema operativo linux ubuntu/debian (tipo Beaglebone – Pcdduino)
- Dongle para añadir capacidad WiFi + usb-hub
- Conectores, fuente y cables varios

CONTENIDOS:

UNIDAD 1: Conceptos generales de un sistema linux embebido. Manejo básico. Comandos de interés. Comparación Linux desktop vs. Embebido vs. Linux embebido RTO.

UNIDAD 2: Conceptos sobre sistemas distribuidos. Sistemas basados en mensajes (MOM). Arquitectura Cliente – Servidor. Arquitectura Publicador – Subscriptor.

UNIDAD 3: Uso de tecnologías middleware para comunicaciones. Definiciones. Ejemplos de distintos middleware para comunicaciones. Bindings a distintos lenguajes. Instalación de middleware. Configuración de middleware en distintos escenarios. Pruebas y mediciones de funcionamiento. Ejercitación.

UNIDAD 4: Trabajo final. Organización de un proyecto. Desarrollo del trabajo final. Seminario de exposición.

BIBLIOGRAFÍA:

Tutorial de Python – <http://python.org.ar/tutorial>

Documentación Oficial de ROS – <http://www.ros.org/>

Sacha Krakowiak, S., "Middleware Architecture with Patterns and Frameworks". Under CC v.3, access from <http://proton.inrialpes.fr/~krakowia/MW-Book/main-onebib.pdf> (last access 24/02/2015)

Página oficial de RabbitMQ – <http://proton.inrialpes.fr/~krakowia/MW-Book/main-onebib.pdf>

Página oficial de ZeroMQ – <http://zeromq.org/>

Sitio de interés – <http://elinux.org/BeagleBoardUbuntu>

Material del curso.

CODIGO CURSO: CDIP.SE 07

DENOMINACIÓN: Plataformas de Desarrollo: Arduino I

RESPONSABLE: Mg. Sebastian TOSCO

CONOCIMIENTOS O REQUERIMIENTOS PREVIOS: Conocimientos básicos de programación y sistemas operativos.

OBJETIVOS:

Al término del curso los participantes serán capaces de:

- Conocer los diversos protocolos de comunicación soportado por Arduino.
- Realizar lecturas de sensores digitales con protocolo 1-Wire e I2C.
- Comprender el funcionamiento de librerías y su utilización.
- Manejar datos en memorias no volátiles.
- Manejar conceptos de comunicaciones inalámbricas.
- Establecer comunicación entre dispositivos mediante bluetooth

JUSTIFICACIÓN:

El Arduino es una plataforma libre, que permite el prototipado de proyectos que requieren un microcontrolador para basar su funcionamiento. Realmente, la aplicación del Arduino puede ir más allá de un simple proyecto, ya que puedes emplearlo en muchos ámbitos de la vida cotidiana, como modificación de tus aparatos eléctricos, creación de nuevos aparatos, elaboración de proyectos para la Universidad o más aún, brindar una solución a una problemática de una empresa.

Una de las principales ventajas del Arduino es que realmente es una tarjeta entrenadora, lo que permite que sólo conectes periféricos o módulos auxiliares a la misma. Esta placa entrenadora se basa en tecnología Atmel, específicamente con microcontroladores como del ATMega328.

DURACIÓN Y ORGANIZACIÓN:

20 hs. Cuatro clases de 5 hs c/u. 1 clase por semana.

METODOLOGÍA DE DICTADO:

La metodología de dictado será Teórico Práctica. Se desarrollarán los temas con el apoyo de presentaciones multimedia, ejemplos funcionales que se demostrarán en el momento y ejercicios a realizar por los asistentes.

METODOLOGÍA DE EVALUACIÓN Y APROBACIÓN:

Para la aprobación del curso se requiere la asistencia al 80% de las clases y la realización de los ejercicios de evaluación.

MATERIAL QUE SE OFRECE AL ASISTENTE:

Material didáctico y códigos usados para el dictado del curso.

CANTIDAD MÍNIMA Y MÁXIMA DE ASISTENTES:

La cantidad máxima de asistentes será de 20 (veinte) participantes.

NECESIDADES Y DISPONIBILIDAD DE INFRAESTRUCTURA:

Se requiere de un aula con capacidad para 20 personas, 20 computadoras (preferentemente conectadas en red) y 1 proyector.

Software Necesario:

- Linux (preferentemente debian o ubuntu)
- IDE Arduino

Hardware Necesario:

- Placa Arduino Nano o Arduino UNO.
- Placa entrenadora multipropósito compatible con Arduino.
- Componentes electrónicos Básicos (Resistencias, Led, etc).
- Kit de sensores varios compatibles con Arduino.

CONTENIDOS:

UNIDAD 1: Introducción a Arduino. Definición. Historia. Introducción al movimiento Open Hardware y Cultura libre. Ejemplos de aplicación. Tipos de placas Arduino. Descripción de sus características. Descarga del software e instalación.

UNIDAD 2: Reconocimiento del IDE de Arduino. Partes y funciones del IDE. Partes de un sketch. Compilación de un sketch. Carga de un sketch a la tarjeta Arduino. Sintaxis de programación. Estructuras. Variables. Funciones. Bucles de Control. Librerías. Funciones y usos de Librerías.

UNIDAD 3: Conceptos de Electronica Aplicado a Arduino. Conexión de protoboards con arduinos. Tipos de señales (análogo y digital). Salidas (digital y PWM). Uso de actuadores. Entradas (digital y análogas). Uso de sensores. Conexión e instalación de los componentes electrónicos más usados.

BIBLIOGRAFÍA:

<http://www.arduino.cc/>

<http://www.arduino.cc/es/>

<http://arduino.cc/es/Secundaria/Secundaria>

<http://arduino-ubuntu.blogspot.com/>

<http://visualp5.net/visualp5net-taller-arduino.html>

<https://sites.google.com/a/divinechildhighschool.org/electronics/Home/Arduino-Lessons>

CODIGO CURSO: CDIP.SE 08

DENOMINACIÓN: Plataformas de Desarrollo: Arduino II

RESPONSABLE: Mg. Sebastian TOSCO

CONOCIMIENTOS O REQUERIMIENTOS PREVIOS: Conocimientos básicos de programación y sistemas operativos.

OBJETIVOS:

Al término del curso los participantes serán capaces de:

- Identificar y aplicar los conceptos fundamentales de electricidad, electrónica y programación aplicados a Arduino.
- Conocer los componentes de la tarjeta Arduino y la utilización de ella.
- Conocer las distintas formas de programación y su realización.
- Diseñar, montar y programar circuitos con sensores y actuadores.
- Montar circuitos en protoboard y simularlos.

JUSTIFICACIÓN:

El Arduino es una plataforma libre, que permite el prototipado de proyectos que requieren un microcontrolador para basar su funcionamiento. Realmente, la aplicación del Arduino puede ir más allá de un simple proyecto, ya que puedes emplearlo en muchos ámbitos de la vida cotidiana, como modificación de tus aparatos eléctricos, creación de nuevos aparatos, elaboración de proyectos para la Universidad o más aún, brindar una solución a una problemática de una empresa.

Una de las principales ventajas del Arduino es que realmente es una tarjeta entrenadora, lo que permite que sólo conectes periféricos o módulos auxiliares a la misma. Esta placa entrenadora se basa en tecnología Atmel, específicamente con microcontroladores como del ATMega328.

DURACIÓN Y ORGANIZACIÓN:

20 hs. Cuatro clases de 5 hs c/u. 1 clase por semana.

METODOLOGÍA DE DICTADO:

La metodología de dictado será Teórico Práctica. Se desarrollarán los temas con el apoyo de presentaciones multimedia, ejemplos funcionales que se demostrarán en el momento y ejercicios a realizar por los asistentes.

METODOLOGÍA DE EVALUACIÓN Y APROBACIÓN:

Para la aprobación del curso se requiere la asistencia al 80% de las clases y la realización de los ejercicios de evaluación.

MATERIAL QUE SE OFRECE AL ASISTENTE:

Material didáctico y códigos usados para el dictado del curso.

CANTIDAD MÍNIMA Y MÁXIMA DE ASISTENTES:

La cantidad máxima de asistentes será de 20 (veinte) participantes.

NECESIDADES Y DISPONIBILIDAD DE INFRAESTRUCTURA:

Se requiere de un aula con capacidad para 20 personas, 20 computadoras (preferentemente conectadas en red) y 1 proyector.

Software Necesario:

- Linux (preferentemente debian o ubuntu)
- IDE Arduino

Hardware Necesario:

- Placa Arduino Nano o Arduino UNO.
- Placa entrenadora multipropósito compatible con Arduino.
- Componentes electrónicos Básicos (Resistencias, Led, etc).
- Kit de sensores varios compatibles con Arduino.

CONTENIDOS:

UNIDAD 1: Comunicación con Arduino. Comunicación serial por hardware y software. Comunicación serial PC-Arduino y Arduino-Arduino. Sensores I2C. Fundamentos de comunicación Ethernet.

UNIDAD 2: Almacenamiento de datos. Lectura y escritura de EEPROM interna. Lectura y escritura de tarjetas SD. Utilización de reloj de tiempo real. Ejercicio práctico: Construcción de un datalogger.

BIBLIOGRAFÍA:

<http://www.arduino.cc/>

<http://www.arduino.cc/es/>

<http://arduino.cc/es/Secundaria/Secundaria>

<http://arduino-ubuntu.blogspot.com/>

<http://visualp5.net/visualp5net-taller-arduino.html>

<https://sites.google.com/a/divinechildhighschool.org/electronics/Home/Arduino-Lessons>

CODIGO CURSO: CDIP.SE 09

DENOMINACIÓN: Plataformas de Desarrollo: Intel Galileo

RESPONSABLE: Mg. Fernando CORTEGGIANO

CONOCIMIENTOS O REQUERIMIENTOS PREVIOS: Conocimientos básicos de programación en C y Linux.

OBJETIVOS:

- Introducir a los alumnos en la plataforma de desarrollo de Intel Galileo.
- Presentar las herramientas de software necesarias en el proceso de desarrollo.
- Presentar alternativas de aplicaciones de hardware.

JUSTIFICACIÓN:

Hoy en día abundan diferentes placas de desarrollo que trabajan bajo diferentes arquitecturas. Intel Galileo es una de estas opciones. Se trata de una familia de placas de desarrollo compatibles con Arduino que trabaja bajo arquitectura Intel.

Ofrece un entorno de desarrollo de hardware y software de código abierto, complementa y amplía la línea de productos Arduino para ofrecer funciones informáticas más avanzadas a quienes ya están familiarizados con las herramientas de prototipos Arduino. Es por ello que se considera fundamental que el ingeniero a cargo del diseño de Sistemas Embebidos incorpore los conceptos fundamentales que serán explicados en el presente curso.

DURACIÓN Y ORGANIZACIÓN:

20 hs. Cuatro clases de 5 hs c/u. 1 clase por semana.

METODOLOGÍA DE DICTADO:

La metodología de dictado será Teórico Práctica. Se desarrollarán los temas con el apoyo de presentaciones multimedia, ejemplos funcionales que se demostrarán en el momento y ejercicios a realizar por los asistentes.

METODOLOGÍA DE EVALUACIÓN Y APROBACIÓN:

Para la aprobación del curso se requiere la asistencia al 80% de las clases y la realización de los ejercicios de evaluación.

MATERIAL QUE SE OFRECE AL ASISTENTE:

Material didáctico y códigos usados para el dictado del curso.

CANTIDAD MÍNIMA Y MÁXIMA DE ASISTENTES:

La cantidad máxima de asistentes será de 20 (veinte) participantes.

NECESIDADES Y DISPONIBILIDAD DE INFRAESTRUCTURA:

Se requiere de un aula con capacidad para 20 personas, 20 computadoras (preferentemente conectadas en red) y 1 proyector.

Software Necesario:

- Linux (preferentemente debian o ubuntu) o Windows 7.

Hardware Necesario:

- Placa Intel Galileo

CONTENIDOS:

UNIDAD 1: Introducción a Intel Galileo. Definición. Historia. Introducción al movimiento Open Hardware y Cultura libre. Ejemplos de aplicación.

UNIDAD 2: Placas Intel Galileo. Descripción de sus características. Componentes de desarrollo. Software e instalación.

UNIDAD 3: Comunicación con Galileo. Comunicación serial por hardware y software. Sensores I2C. Fundamentos de comunicación Ethernet.

UNIDAD 4: Almacenamiento de datos. Lectura y escritura de EEPROM interna. Lectura y escritura de tarjetas SD. Utilización de reloj de tiempo real.

BIBLIOGRAFÍA:

<https://communities.intel.com/docs/DOC-22886>

<https://communities.intel.com/docs/DOC-22475>

<https://communities.intel.com/community/makers/galileo/documentation>

CODIGO CURSO: CDIP.SE 10

DENOMINACIÓN: Plataformas de desarrollo: EduCIAA

RESPONSABLE: Mg. Manuel AMOR

CONOCIMIENTOS O REQUERIMIENTOS PREVIOS: Conocimientos básicos de programación en C y arquitectura de procesadores ARM.

OBJETIVOS:

- Introducir a los alumnos en la arquitectura de una plataforma de desarrollo moderna y económica basada en la CIAA.
- Presentar las herramientas de software necesarias en el proceso de desarrollo.
- Brindar una base para diseñar sistemas capaces de considerar cuestiones de disponibilidad, confiabilidad, verificación, validación y seguridad intrínseca.

JUSTIFICACIÓN:

La Computadora Industrial Abierta Argentina (CIAA) es una plataforma electrónica de hardware libre que, además, presenta ciertas características que la hacen cualitativamente diferente respecto de las demás, al punto que configura una experiencia única a nivel global.

Una de las principales diferencias es su concepción, en la que se priorizó generar un producto robusto para que soporte las condiciones hostiles de los ambientes industriales los que abundan ruidos, vibraciones, temperaturas extremas, picos de tensión e interferencias electromagnéticas, pero, a la vez, diseñarla de modo tal que pueda ser fabricada en Argentina.

Es por ello que se considera fundamental que el ingeniero a cargo tanto del diseño como de la enseñanza de sistemas embebidos incorpore los conceptos fundamentales que serán explicados en el presente curso.

DURACIÓN Y ORGANIZACIÓN:

20 hs. Cuatro clases de 5 hs c/u. 1 clase por semana.

METODOLOGÍA DE DICTADO:

La metodología de dictado será Teórico Práctica. Se desarrollarán los temas con el apoyo de presentaciones multimedia, ejemplos funcionales que se demostrarán en el momento y ejercicios a realizar por los asistentes.

METODOLOGÍA DE EVALUACIÓN Y APROBACIÓN:

Para la aprobación del curso se requiere la asistencia al 80% de las clases y la realización de los ejercicios de evaluación.

MATERIAL QUE SE OFRECE AL ASISTENTE:

Material didáctico y códigos usados para el dictado del curso.

CANTIDAD MÍNIMA Y MÁXIMA DE ASISTENTES:

La cantidad máxima de asistentes será de 20 (veinte) participantes.

NECESIDADES Y DISPONIBILIDAD DE INFRAESTRUCTURA:

Se requiere de un aula con capacidad para 20 personas, 20 computadoras (preferentemente conectadas en red) y 1 proyector.

Software Necesario:

- Linux (preferentemente debian o ubuntu)
- GNU Toolchain

Hardware Necesario:

- EDU-CIAA
- Kit de sensores varios

CONTENIDOS:

UNIDAD 1: Introducción a la CIAA. Descripción. Ventajas. Hardware de la CIAA (versiones multiprocesadores). Firmware de la CIAA. Software de la CIAA.

UNIDAD 2: Entorno de desarrollo para CIAA: Toolchain. Definición de toolchain. Formas de armar un toolchain. Componentes. Bibliotecas de C. Armandando un toolchain: construcción manual, toolchains precompilados, uso de herramientas de ayuda en el armado.

UNIDAD 3: freeOSEK. Breve historia. Características principales. Esquema de versiones. Fuentes del kernel y parches. Opciones principales de configuración. Cross-compilación e instalación del kernel.

UNIDAD 4: Depuración. Definición. Depuración en x86.

UNIDAD 6: Testing y Certificación. Definición. Testing. Configuración del entorno. Organización del testing. Certificación bajo IEC 61508

BIBLIOGRAFÍA:

<http://proyecto-ciaa.com.ar/>

CODIGO CURSO: CDIP.SE 11

DENOMINACIÓN: Interfaces y Periféricos

RESPONSABLE: Dr. Guillermo MAGALLAN

CONOCIMIENTOS O REQUERIMIENTOS PREVIOS: Conocimientos básicos de programación en C y arquitectura de procesadores ARM.

OBJETIVOS:

El curso tiene por objetivo dar al alumno el conocimiento de las principales interfaces y periféricos comúnmente utilizados en sistemas electrónicos digitales.

Se pretende también que el alumno logre identificar las características principales de las interfaces y la elección más conveniente para una aplicación puntual: especificaciones de velocidad, interconexión con otros dispositivos, inmunidad al ruido. Capacitar al alumno en el uso de periféricos para interacción con el mundo analógico e interfaces de usuario.

JUSTIFICACIÓN:

Interfaces y Periféricos es un curso que pretende que el alumno adquiera conocimientos de las diferentes arquitecturas basadas en microprocesadores y microcontroladores y en los diferentes periféricos que conectamos de forma habitual a través de interfaces bien definidas.

Un Ingeniero a cargo de desarrollos de sistemas embebidos debe poder enfrentarse a las diferentes tareas relacionadas con el uso e implementación de interfaces y sus periféricos asociados, así como la programación de controladores para su correcto funcionamiento, es por ello que se considera fundamental que se capacite e incorpore los conceptos fundamentales que serán explicados en el presente curso.

DURACIÓN Y ORGANIZACIÓN:

20 hs. Cuatro clases de 5 hs c/u. 1 clase por semana.

METODOLOGÍA DE DICTADO:

La metodología de dictado será Teórico Práctica. Se desarrollarán los temas con el apoyo de presentaciones multimedia, ejemplos funcionales que se demostrarán en el momento y ejercicios a realizar por los asistentes.

METODOLOGÍA DE EVALUACIÓN Y APROBACIÓN:

Para la aprobación del curso se requiere la asistencia al 80% de las clases y la realización de los ejercicios de evaluación.

MATERIAL QUE SE OFRECE AL ASISTENTE:

Material didáctico y códigos usados para el dictado del curso.

CANTIDAD MÍNIMA Y MÁXIMA DE ASISTENTES:

La cantidad máxima de asistentes será de 20 (veinte) participantes.

NECESIDADES Y DISPONIBILIDAD DE INFRAESTRUCTURA:

Se requiere de un aula con capacidad para 20 personas, 20 computadoras (preferentemente conectadas en red) y 1 proyector.

Software Necesario:

- Linux (preferentemente debian o ubuntu)
- Eclipse
- Emulador Android

CONTENIDOS:

UNIDAD 1: Interfaces de conexión. Tipos y niveles de interfaces y periféricos. Sistemas basados en microprocesadores, microcontroladores, PCs. Compatibilidad de tecnología. Velocidades, cantidad de información.

UNIDAD 2: Entradas y salidas. Tipos de entradas y salidas digitales. Acondicionamiento de señales digitales. drivers. Aislación de entradas y salidas. Optoacopladores. Entradas y salidas analógicas, acondicionamiento, aislación.

UNIDAD 3: Interfaces seriales y paralelas estándares. Puerto paralelo del PC (IEEE1284). Modos ECP, EPP. Aplicaciones de comunicaciones paralelas. Norma RS232. Norma RS485. Drivers. Bus I2C. Bus SPI. USB. Ethernet. Aplicaciones.

UNIDAD 4: Periféricos. Conversores A/D y D/A, externos e internos. Aplicaciones con periféricos.

BIBLIOGRAFÍA:

Halsall, Fred. (1998). Comunicación de datos, redes de computadores y sistemas abiertos. México: Addison Wesley. (ISBN : 968-444-331-5).

Paret, Dominique. (1995). El bus I2C. De la teoría a la práctica. Madrid: Paraninfo. (ISBN: 84-283-2189-2).

Martínez Durá, Rafael J. et al. Estructura de computadores y periféricos. Madrid: RAMA. (ISBN: 84-7897-447-4).

Neil Willis, “Fundamentos de arquitectura de ordenadores y comunicaciones de datos”. Ed. Anaya Multimedia (1990).

CODIGO SEMINARIO: SDIP.SE 12

DENOMINACIÓN: Seminario de aplicación/actualización I

RESPONSABLE: Mg. Héctor MAGNAGO

CONOCIMIENTOS O REQUERIMIENTOS PREVIOS:

OBJETIVOS:

- Presentar problemas de aplicación concreta de los sistemas embebidos en diferentes ámbitos tales como agropecuarios, industriales, de seguridad, etc.

JUSTIFICACIÓN:

El planteamiento de necesidades concretas y actuales por parte de especialistas o usuarios de otras disciplinas, permite al alumno visualizar un panorama de aplicación más amplio de los contenidos estudiados en esta diplomatura a la vez que plantearán distintos desafíos cuyo ataque y resolución puede, no solamente realizar un aporte directo a la culminación de la diplomatura sino que brinda la oportunidad de dar solución a dichas necesidades.

DURACIÓN Y ORGANIZACIÓN:

20 hs. Cinco reuniones de 4 hs c/u.

METODOLOGÍA DE DICTADO:

En cada una de las tres primeras reuniones un invitado especial, presentará la problemática a resolver. Los alumnos utilizarán estas reuniones para recabar la mayor información posible y seleccionarán, individualmente o por grupos, al menos uno de los problemas planteados. En las dos últimas reuniones los alumnos plantearán un esbozo de solución que será discutida entre todos los actores presentes.

METODOLOGÍA DE EVALUACIÓN Y APROBACIÓN:

Para la aprobación del curso se requiere la asistencia al 80% de las reuniones de planteo de problemas y la participación en al menos una de las reuniones de propuesta de solución.

MATERIAL QUE SE OFRECE AL ASISTENTE:

No corresponde.

CANTIDAD MÍNIMA Y MÁXIMA DE ASISTENTES:

La cantidad máxima de asistentes será de 20 (veinte) participantes.

NECESIDADES Y DISPONIBILIDAD DE INFRAESTRUCTURA:

Se requiere de un aula con capacidad para 20 personas y 1 proyector.

TEMÁTICAS DE LAS REUNIONES:

Aplicaciones de los Sistemas Embebidos en:

- Actividades agropecuarias.
- Sistemas de observación y monitoreo.
- Sistemas de alarma contra incendios forestales
- Sistemas de alarma por crecidas de ríos y/o arroyos

CODIGO SEMINARIO: SDIP.SE 13

DENOMINACIÓN: Seminario de aplicación/actualización II

RESPONSABLE: Mg. José Luis HERNANDEZ

CONOCIMIENTOS O REQUERIMIENTOS PREVIOS:

OBJETIVOS:

- Presentar problemas de aplicación concreta de los sistemas embebidos en diferentes ámbitos tales como agropecuarios, industriales, de seguridad, etc.

JUSTIFICACIÓN:

El planteamiento de necesidades concretas y actuales por parte de especialistas o usuarios de otras disciplinas, permite al alumno visualizar un panorama de aplicación más amplio de los contenidos estudiados en esta diplomatura a la vez que plantearán distintos desafíos cuyo ataque y resolución puede, no solamente realizar un aporte directo a la culminación de la diplomatura sino que brinda la oportunidad de dar solución a dichas necesidades.

DURACIÓN Y ORGANIZACIÓN:

20 hs. Cinco reuniones de 4 hs c/u.

METODOLOGÍA DE DICTADO:

En cada una de las tres primeras reuniones un invitado especial, presentará la problemática a resolver. Los alumnos utilizarán estas reuniones para recabar la mayor información posible y seleccionarán, individualmente o por grupos, al menos uno de los problemas planteados. En las dos últimas reuniones los alumnos plantearán un esbozo de solución que será discutida entre todos los actores presentes.

METODOLOGÍA DE EVALUACIÓN Y APROBACIÓN:

Para la aprobación del curso se requiere la asistencia al 80% de las reuniones de planteo de problemas y la participación en al menos una de las reuniones de propuesta de solución.

MATERIAL QUE SE OFRECE AL ASISTENTE:

No corresponde.

CANTIDAD MÍNIMA Y MÁXIMA DE ASISTENTES:

La cantidad máxima de asistentes será de 20 (veinte) participantes.

NECESIDADES Y DISPONIBILIDAD DE INFRAESTRUCTURA:

Se requiere de un aula con capacidad para 20 personas y 1 proyector.

TEMÁTICAS DE LAS REUNIONES:

Aplicaciones de los Sistemas Embebidos en:

- Captación y recuperación de datos en estaciones meteorológicas remotas.

- Domótica. Comandos remotos aplicados a la seguridad. Simulaciones de presencia. Monitoreo de hogares, oficinas o plantas.
- Agroindustria. Instrumentación industrial

CODIGO TALLER: TDIP.SE 1

DENOMINACIÓN: Taller de Trabajo final integrador

CONOCIMIENTOS O REQUERIMIENTOS PREVIOS:

OBJETIVOS:

- Que el alumno concrete un trabajo final integrador sobre un tema a elección con el cual finalice sus estudios y obtenga el título de Diplomado Superior

JUSTIFICACIÓN:

La integración de los conocimientos tomados en el desarrollo de los cursos que conforman el proyecto permite al alumno reconocer sus capacidades y aplicar su aprendizaje a la resolución de un caso concreto. La guía de los profesores y el establecimiento de los plazos a cumplir, pretende garantizar la finalización de los estudios de la mayor parte de los alumnos de una cohorte.

DURACIÓN Y ORGANIZACIÓN:

20 hs. Ocho reuniones de 2.5 hs c/u.

METODOLOGÍA DE DICTADO:

En la primera reunión se hará una presentación por parte de los profesores guía sobre los temas en los cuales podrán asesorar y apoyar a los alumnos. Asimismo se dará a conocer una lista de problemas, no excluyente, para que puedan elegirse para su resolución. Cada alumno podrá, a su vez, proponer otra problemática y consensuarla con su profesor guía. Las posteriores reuniones servirán para desarrollar el proyecto y la última de ellas se utilizará para la exposición de la solución.

METODOLOGÍA DE EVALUACIÓN Y APROBACIÓN:

Para la aprobación del curso se requiere la asistencia al 80% de las reuniones y la exposición de la solución al problema elegido.

MATERIAL QUE SE OFRECE AL ASISTENTE:

No corresponde.

CANTIDAD MÍNIMA Y MÁXIMA DE ASISTENTES:

La cantidad máxima de asistentes será de 20 (veinte) participantes.

NECESIDADES Y DISPONIBILIDAD DE INFRAESTRUCTURA:

Se requiere de un aula con capacidad para 20 personas y 1 proyector.