Sistemas Embebidos para Tiempo Real
Reglamento del Curso

21 de febrero de 2025

1. Inscripción

Los estudiantes interesados en cursar la asignatura deben:

- Inscribirse en bedelía. El plazo de inscripción en bedelía es corto (finaliza el día previo al inicio de clases) y NO tiene período de desistimiento, por favor no dejar para último momento la inscripción.
- Matricularse en EVA.

2. Metodología del curso

El curso está organizado en: clases teóricas, laboratorios, y un proyecto. En la modalidad de curso de grado, los laboratorios y el proyecto se realizan en equipos de tres estudiantes (excepcionalmente pueden ser de dos estudiantes). Como curso de posgrado esto puede variar.

3. Aprobación del curso

La asistencia a las defensas de laboratorio y presentaciones de proyecto es obligatoria. Cada estudiante puede tener hasta **una** inasistencia debidamente **justificada** al laboratorio. En ese caso, los docentes podrán proponer una evaluación complementaria de ese laboratorio en fecha y hora a coordinar.

El curso no tiene examen y para su aprobación se deberá aprobar cada una de las partes del curso (laboratorio y proyecto). La calificación final será ponderada de la siguiente manera:

- Laboratorio: 40 %, considerando los cuestionarios individuales, y el desempeño durante la práctica.
- Proyecto: 60 %, considerando la ejecución del proyecto y los resultados obtenidos, las entregas intermedias, las presentaciones, el video de la demostración (demo), y la entrega. En el Apéndice B se presenta el listado de los puntos deseables para cada una de las instancias evaluables del proyecto.

4. Teórico

Se dictarán 15 clases aproximadamente de 90 minutos cada clase, dos clases por semana.

En caso de inasistencia, se recomienda que durante la semana siguiente se realicen los "ejercicios de teórico". Para ello el estudiante debe determinar el tema dado en la clase que faltó y fijarse en la lista de "Ejercicios de teórico" en el EVA lo que debe realizar. Aquellos estudiantes que deseen que esta entrega sea evaluada deben entregarla dentro de la semana siguiente.

5. Laboratorio

Se realizarán 4 prácticas de laboratorio que deberán ser completadas en equipo por grupos de dos o tres estudiantes. Si bien se recomienda el intercambio de ideas entre grupos, es obligatorio que cada grupo realice los trabajos en forma individual en su computadora. Se recomienda que cada integrante del grupo trabaje en su PC, para asegurarse de adquirir las habilidades prácticas.

Se realizarán 4 defensas de laboratorio de hasta 4 horas cada una. Los estudiantes deberán asistir a las defensas habiendo preparado las mismas de acuerdo a lo solicitado en la letra de laboratorio correspondiente. Las prácticas de laboratorio tendrán las siguientes etapas:

- Publicación de la letra: a más tardar una semana antes de la defensa se publicará la letra correspondiente.
- Clase de consulta: en los días previos habrá al menos una clase de consulta.
- Defensas de laboratorio: constarán de dos partes.
    1. Se realizará un **cuestionario** individual a todos los estudiantes consistente en preguntas/ejercicios de resolución rápida.
    2. Cada grupo realizará una **demostración/defensa** del correcto funcionamiento de todo lo solicitado en la letra del laboratorio. **Los estudiantes serán evaluados individualmente y grupalmente** considerando:
        - si todas las tareas indicadas en la letra del laboratorio se completaron,
        - si las soluciones implementadas para las tareas son realizadas correctamente, y
        - si los comentarios en los programas siguen las recomendaciones provistas en los ejemplos que se encuentran en la sección *Laboratorios* de la web del curso (ver *Ejemplo comentario módulo (Doxygen)* o archivo *temperatura.h*). En particular, en el caso de la interfaz de los módulos (archivo ".h") se verificará sí se detalla la forma de uso de cada una de las funciones y la descripción de cada uno de los parámetros.

5.1. Cuestionario del laboratorio

El control de conocimientos (cuestionario) tiene como objetivo verificar que el estudiante ha preparado la práctica razonablemente y que cuenta con los conceptos requeridos para su aprovechamiento.

- Los cuestionarios constan de 4 preguntas y se aprueba con un mínimo de 2 preguntas correctas (nota mínima = 0, nota máxima = 4, nota de aprobación >= 2).
- El cuestionario tiene una duración de 15 minutos y se realizará durante la defensa de laboratorio.
- Si un estudiante llega luego de empezado el cuestionario, pero antes de su finalización, puede realizarlo, pero deberá entregar al mismo tiempo que el resto de los estudiantes.
- Si un estudiante llega luego de finalizado el cuestionario, tiene nota 0.

Los cuestionarios son estrictamente individuales. Cualquier violación a este punto será tratado con la severidad correspondiente y de acuerdo a las normas vigentes de la Facultad.

5.2. Calificación del laboratorio

El puntaje de la práctica y la aprobación del laboratorio se determina de la siguiente forma:

- La nota final del laboratorio (sobre el total de 40 puntos) es individual y se computa de acuerdo al cuestionario individual, y al desempeño grupal e individual durante la defensa del laboratorio.
- Cada laboratorio se puntúa de 0 a 3, donde:
  - 0 = Deficiente (actividad durante la práctica con grandes carencias que demuestran un trabajo y/o una dedicación francamente insuficiente).
  - 1 = Regular
  - 2 = Bueno
  - 3 = Muy Bueno o Excelente
- Un 0 en la nota de la práctica, afectará de forma muy negativa en el nota final del laboratorio.
- Si un estudiante obtiene nota 0 en dos o más de las cuatro prácticas, reprueba el laboratorio y por lo tanto pierde el curso.
- Sobre el cuestionario:

- Si un estudiante obtiene nota insuficiente (cero o uno) en tres o más de los cuatro cuestionarios, reprueba el laboratorio y por lo tanto pierde el curso.
- Si un estudiante cumple con las condiciones para aprobar el laboratorio y obtiene nota insuficiente (cero o uno) en dos de los cuatro cuestionarios, pierde **todos** los puntos del laboratorio (es decir, tendrá 0 de los 40 puntos del laboratorio, independientemente de la nota en las prácticas restantes).

6. Proyecto

Al final del curso cada grupo debe defender su proyecto. Las fechas de las diferentes entregas se publicarán oportunamente y se realizará mediante una tarea del EVA. Eventualmente se podrá solicitar la entrega de una versión impresa.

Atención: todos los documentos deben ser entregados en formato PDF.

6.1. Primera entrega de la especificación del proyecto (versión breve)

6.2. Entrega de la especificación del proyectos (versión detallada)

Se entregará un documento (en formato PDF) que deberá abordar el siguiente contenido:

- Nombre del proyecto: corto y descriptivo
- Nombre de los integrantes y tutor
- Descripción del problema a ser resuelto: qué problema se va a abordar.
- Antecedentes: proyectos anteriores, artículos, libros, soluciones disponibles, etc.
- Objetivos del proyecto: qué se va a lograr, qué se va a entregar.
- Alcance del proyecto: definir claramente qué incluye y qué no.
- Descripción del sistema:
  - Descripción funcional (lo que debe hacer)
  - Diagrama de bloques (conceptual): división jerárquica/bloques constitutivos.
- Requerimientos y restricciones del sistema:
  - Procesamiento y memoria: estimación preliminar.

    - Tiempos de respuesta: estimación de los tiempos de respuesta máximos requeridos.

- Diseño preliminar:
    - Plataforma de hardware: descripción de las partes, especificar si se necesita diseñar/construir hardware adicional y detallar qué.
    - Arquitectura de software: describir la arquitectura justificando su selección.

- Planificación:
    - Actividades/tareas (nombre, descripción, duración)
    - Describir las pruebas a realizar.
    - Hito intermedio: describir entregables (buscar que coincidan con la fecha de la presentación)
    - Cronograma.

Se sugiere mirar el template de la memoria del proyecto (disponible en la página web del curso) antes de empezar la redacción de este documento.

6.3. Presentaciones

- Consejos generales:
    - Seleccionar el contenido de las presentaciones adecuadamente, si se improvisa se degrada mucho la calidad de la presentación y aumenta el tiempo innecesariamente.
    - Ensayar la presentación tantas veces como sea necesario y controlar el tiempo para realizar posibles recortes o ajustes para no superar el tiempo disponible.

- Organización:
    - Las presentaciones comenzarán a la hora fijada de manera puntual.
    - Luego de la presentación habrá un espacio para preguntas y para la devolución de la presentación.

- Evaluación (ítems considerados):
    - Contenido: selección de los temas, alcance, abordaje de puntos esenciales (en función de cuál presentación se trate).
    - Claridad: orden, precisión en los términos, etc.
    - Profundidad técnica: especialmente en la presentación del problema y la solución.

- Calidad de las transparencias: poco uso de textos (orientado a conceptos fundamentales), buen soporte gráfico (imágenes, figuras, gráficas, esquemas, diagramas, etc.), y en una escala adecuada (visibles).
- Exposición: correcto uso del lenguaje, adecuada actitud y postura corporal (mirar al auditorio, etc.), equilibrio en el reparto del tiempo entre los integrantes, cumplimiento **estricto** en las restricciones de tiempo. La presentación debe ser dinámica y fácil de seguir. Lo que se dice oralmente no es exactamente lo que está escrito en las diapositivas. Se evidencia que todos los integrantes dominan el 100 % del proyecto.

6.3.1. Presentación de avance (hito intermedio)

- Objetivo: promover la culminación del proyecto en el plazo estipulado.
- Método: cada grupo toma conciencia del grado de avance que ha logrado en el proyecto respecto a la planificación inicial. Además de la propia evaluación, los docentes les podrán señalar posibles complicaciones y sugerencias de caminos a seguir.
- Contenido: la presentación debe permitir entender la esencia del proyecto y el grado de avance, sin entrar en detalles técnicos que escapan al alcance posible dentro del tiempo estipulado. Se recomienda no utilizar más de 10 diapositivas (una por minuto), utilizando una organización similar a esta:
  - Carátula (1 slide)
  - Repasar brevemente y de manera concreta el problema que van a resolver y delinear la solución propuesta (aprox 3 slides).
  - Contar en lo que han trabajado hasta la fecha y qué dificultades se han tenido (aprox 3 slides).
  - Opcionalmente se puede mostrar un video de 30 segundos mostrando lo alcanzado.
  - Evaluar el avance, comparando la planificación con la ejecución real. Presentar lo que se había prometido como entregable para este hito y qué porcentaje de avance se tiene. Comentar si corresponde, los ajustes que hayan tenido que hacer y los necesarios a futuro (aprox 3 slides).
- **Tiempo disponible: cada grupo tendrá 10 minutos.**

6.3.2. Defensa

La defensa consta de tres etapas: presentación oral final, demostración (demo) y respuestas a preguntas.

- Organización:

- Los estudiantes que defienden deberán estar presentes en el aula (presencial o virtual) al menos 10 minutos antes del inicio de la hora asignada, y deberán tener todo pronto (presentación, demo, video, quién va compartir pantalla, etc.).
- Tener en cuenta que podría haber público externo al curso (otros docentes, estudiantes, etc.) por lo que se ruega puntualidad. Por tanto, pueden estar presentes en las presentaciones de sus compañeros (aunque no es obligatorio).
- La defensa consistirá en una presentación oral de 20 minutos, una demo del sistema desarrollado funcionando (estimado en 5 minutos), y un bloque de preguntas y respuestas.
- Respetar el tiempo asignado es muy importante. Se recomienda fuertemente ensayar y controlar el tiempo durante los ensayos. A los 15 minutos se les avisa a los estudiantes para que terminen a los 20 minutos, y al llegar a 25 minutos las presentaciones son interrumpidos.

- Presentación oral final:
  - La presentación permite entender la esencia del proyecto y el grado de completitud de los objetivos.
  - Tomar en cuenta las sugerencias/críticas de las presentaciones anteriores.
  - Deben presentar en forma clara: la descripción del problema, los objetivos del proyecto, el alcance, la solución desarrollada y las conclusiones. También deben comentar, al menos brevemente, el desarrollo del plan de trabajo. La presentación puede basarse en la estructura sugerida en la memoria del proyecto (ver más adelante): introducción, diseño, solución, y conclusiones.
  - La mayoría del tiempo debería dedicarse a describir la solución, esto es, el diseño y su implementación. Si tuviera desarrollo de hardware debería señalarse y describirlo, pero brevemente. El énfasis estará en la solución de software y en cómo se aplicaron los conceptos desarrollados durante el curso.

- Demostración (demo):
  - Deberán presentar la demo grabada en un video de duración máxima 3 minutos. Asimismo, deberán ser capaces de mostrar el funcionamiento "en vivo" si algún docente lo solicita.
  - La demo debe permitir observar que el proyecto funciona correctamente, y no levantar suspicacias sobre bugs escondidos.
  - Mostrar el caso de uso "normal", y si da el tiempo otros casos de uso.
  - Mostrar el setup, y la interacción con el usuario o con otros equipos cuando la haya.

- Mostrar la configuración del sistema, y si hay interfaz mostrarla (ya sea PC, display, leds, botones o teclados).
- El video debe tener un perfil técnico (no promocional/marketinero) y debe ser dinámico. Se debe mostrar el sistema en funcionamiento, y además apreciar los detalles más relevantes. Una voz en off o texto sobre-impreso deben orientar al espectador dónde mirar y qué se espera ver.

- Luego de la presentación oral y la demo, habrá una instancia de preguntas de los docentes. La defensa es la instancia más importante de evaluación del curso, por tanto las preguntas pueden ser tanto del proyecto, como de conceptos vistos en el curso.

6.4. Entrega del proyecto

La entrega del proyecto incluye los siguientes ítems:

1. Memoria del proyecto en formato PDF según template disponible en la página web.
2. Archivos fuentes (en un anexo de la memoria se deben listar y describir brevemente los archivos).
3. Documentación generada en Doxygen en formato HTML (si además queda integrada en una anexo a la memoria mejor, pueden generar un archivo PDF o Latex para ello).
4. Fecha de entrega: ver cronograma disponible en página web del curso.

6.5. Re-entrega final del proyecto

Se solicitará una re-entrega final posterior a la defensa. La re-entrega deberá contemplar las eventuales correcciones recibidas, tanto al código como a la memoria. Las correcciones serán enviadas por escrito, y/o oralmente durante la presentación. La re-entrega final del proyecto incluye los siguientes ítems:

1. Memoria del proyecto en formato PDF según template disponible en la página web.
2. Archivos fuentes (en un anexo de la memoria se deben listar y describir brevemente los archivos).
3. Documentación generada en Doxygen en formato HTML (si además queda integrada en una anexo a la memoria mejor, pueden generar un pdf o latex para ello).
4. Archivo en formato TXT con un link al video de la demo (puede ser en youtube, dropbox, drive, etc.). Los videos (o fragmentos de ellos) se podrán usar para difusión. Si alguien tiene inconvenientes con esto por favor debe avisarle al responsable de la asignatura.

5. Fecha de entrega: ver cronograma disponible en página web del curso.

A. Apéndice: temas clave

A.1. modularización / encapsulamiento

- visibilidad de variables (local vs global)
- almacenamiento de variables (auto vs static)
- evitar uso de variables globales / declarar static
- Utilizar funciones de acceso a variables privadas

A.2. lenguaje c

- tipos de datos, uso de uint8_t / uint16_t
- tipos de datos definidos por usuario (typedef struc) manipulación de bits, manejo de máscaras (escritura/lectura de registros del uC)
- punteros
- colas de datos
- test: uso de assert en uC
- macros vs funciones

A.3. interrupciones

- entender como funcionan: qué es una IRQ, qué es una ISR
- vinculación IRQ - ISR (tabla de vectores de interrupción)
- como sé implementan en uC: manejo por registros, diferencia entre el bit de flag (Señaliza IRQ) y el bit de enable (habilita interrupción)
- como sé implementan en C: #pragma

A.4. round robin con interrupciones / planificación por encolado de funciones

- qué se hace en el main
- qué se hace en la interrupción
- donde se va dormir, cómo se va dormir
- problema de datos compartidos

A.5. tiempos de respuesta

- tiempo de respuesta según diferentes arquitecturas
- tiempo de respuesta de una acción realizada en una interrupción
- tiempo de respuesta de una acción en el main
- problemas asociados a tareas con interrupciones inhabilitadas
- problemas asociados a tareas bloqueantes

A.6. Periféricos

- funcionamiento básico Basic Clock Module+: diferencias entre fuentes de reloj, y reloj, opciones disponibles,
- funcionamiento básico del timer
- funcionamiento básico de la usci/uart. Interrupción de registro vacío, nociones básicas del protocolo uart, representación y manipulación de datos (ASCII, Hexadecimal, Decimal)
- funcionamiento básico del adc

A.7. Otras herramientas

- manejo básico de git: clonar repo, aportar cambios al repo, ramas, manejar conflictos simples
- manejo básico de doxygen: documentar el .h, poder compilar y generar un PDF o HTML
- manejo básico de AD2: analizador lógico, osciloscopio, trigger, generador de señales
- manejo básico de Enery Trace
- manejo básico de una terminal

A.8. Habilidades transversales

- Lectura de documentación técnica
- Saber identificar dónde buscar la información (diferencia entre cartillas del launchpad, microcontrolador y familia)
- Trabajar en equipo
- Redactar documentación técnica
- Presentación de trabajos/resultados técnicos

B. Apéndice: puntos deseables para cada una de las instancias evaluables del proyecto

Con el objetivo de guiar a los estudiantes en el desarrollo de un proyecto "excelente", se presenta la siguiente lista de "puntos deseables" para cada una de las instancias evaluables del proyecto. Vale la pena mencionar que la totalidad de puntos de esta lista corresponde a un proyecto más que excelente, y que es perfectamente posible sacar una calificación de 12 sin cubrir necesariamente todos los puntos de manera minuciosa. Esta lista se facilita solamente a modo de referencia. Los componentes evaluables de un proyecto son:

- Documentación (incluye entregas intermedias).
- Base de código.
- Presentación oral primer hito.
- Presentación oral final.
- Demo.
- Respuestas a preguntas durante la defensa.
- Ejecución de proyecto.

B.1. Documentación

- Sigue las pautas establecidas en el template, en la Sección 6.1 y 6.2.
- La documentación incluye una introducción clara al problema, suficiente sin un contexto anterior. Se describen antecedentes suficientes.
- Los objetivos del proyecto están explicados claramente, y coinciden con los objetivos expresados en la especificación detallada. En caso de existir diferencias, están debidamente justificadas.
- El alcance del proyecto queda claro, sin ambigüedades respecto a lo que cubre y a lo que no cubre.
- El sistema está descrito de forma detallada:
    - Descripción de software: arquitectura general del software, descripción de módulos, y arquitectura individual de los módulos más notables. Idealmente se debe hacer uso de diagramas de bloques. En caso de utilizarse máquinas de estados se usan diagramas de transición de estados. En caso de existir una interacción compleja entre los actores, se usan diagramas que representen las interacciones.
    - Descripción de hardware: diagrama de bloques, datos sobre los componentes más notables, etc.

- Las pruebas y mediciones deben estar debidamente reportadas:
  - Las pruebas deben estar orientadas a validar los objetivos del proyecto. Todos los objetivos deben estar acompañados de un test que lo evalúe.
  - El funcionamiento general del sistema debe estar debidamente reportado.
  - No puede haber dudas respecto a la integridad de los datos reportados.
- Conclusiones
  - Evitar conclusiones genéricas. En lugar de"funcionamos bien como grupo" o "profundizamos sobre los conceptos de la asignatura", detallar qué aspectos se destacan del funcionamiento o qué conceptos se profundizaron.
  - Análisis independiente y detallado en relación a los objetivos del proyecto. Valoración explícita del grado de éxito del proyecto. Si corresponde, detallar principales lecciones aprendidas.
  - Es deseable contar con un análisis que compare la dedicación planificada con la realizada.
- Otros elementos de documentación: documento bien estructurado, buenas imágenes en un estilo armonioso, ausencia de faltas de ortografía, anexos conceptualmente correctos (no deben adjuntar secciones importantes a los anexos, ni viceversa), carátula correcta (contiene todos los elementos solicitados).

B.2. Base de código

- El repositorio incluye un README.md que guía al lector en sus primeros pasos con el proyecto.
- El proyecto debidamente estructurado (carpetas /src, /include, etc.).
- El directorio del proyecto incluye solamente archivos fuente. No incluye archivos auxilares de ningún tipo: documentos (".pdf", ".doc", etc.), imágenes, binarios, u otros (".o", ".elf", etc.).
- Código correctamente modularizado: funcionalidades independientes contenidas en archivos independientes. Correcto definición y uso de variables (global, local, static, etc.). No hay constantes sueltas en el código (todo valor "duro" se define mediante un "define").
- La documentación generada Doxygen es razonablemente seguible.
- Indentación correcta.

B.3. Presentaciones, demo y respuestas

- Se siguen las pautas establecidas en la sección 6.3.
- Presentación de avance: permite entender la esencia del proyecto y el grado de avance.
- Presentación oral final: permite entender la esencia del proyecto y el grado de completitud de los objetivos.
- Demo: permite observar que el sistema desarrollado funciona correctamente.
- Respuestas a preguntas durante la defensa: las preguntas se responden con solvencia, utilizando lenguaje técnico apropiado. Todos los integrantes del grupo responden preguntas.

B.4. Ejecución de proyecto

- Los tutores tienen constancia de que el proyecto se ejecutó correctamente, manteniendo una buena comunicación.
- Hubo trabajo sostenido a lo largo del semestre
- Gestión de versiones: Se evidencia participación de todos los integrantes en la generación de código, utilizando mensajes de commit adecuados.