

Evaluación Globalizadora A

Sistemas Operativos en Tiempo Real

CARLASSARA Fabrizio
Auxiliar
Técnicas Digitales III - UTN FRA
fabrizio.carlassara@gmail.com

Ing. MARÍN Ariel
Profesor Adjunto
Técnicas Digitales III - UTN FRA
td3utnfra@gmail.com

VAN ROEY Julián
Auxiliar
Técnicas Digitales III - UTN FRA
julianvroey@gmail.com

I. INTRODUCCIÓN

En este documento se desarrollan las condiciones y la consigna de trabajo para la Evaluación Globalizadora A (en adelante EGA). Este

II. CONDICIONES DEL TRABAJO

A continuación se describen algunas formalidades en cuanto a fechas, entrega y condiciones de aprobación.

A. Contenidos analizados

Entre los contenidos que abarca esta evaluación, tenemos:

- Capítulo 1: Manejo de Tareas
- Capítulo 2: Manejo de Colas
- Capítulo 3: Gestión de Interrupciones / Semáforos
- Capítulo 4: Administración de Recursos / Exclusión
- Capítulo 5: Uso de Memoria

B. Condiciones de aprobación

Condición de Aprobación (Nota: 4 o 5): Sujeta al funcionamiento general del equipo con base en su funcionamiento mínimo y exposición oral.

C. Condiciones de promoción

Condición de Promoción (Nota: 6 - 10): Tener el 100% de la consigna desarrollada con informe y presentación oral. Todos los integrantes del grupo DEBEN estar presentes en todas las instancias de entrega del Parcial.

D. Fechas importantes

- Curso 511 - Miércoles
 - Entrega de consignas: 25/06/2025
 - Primera evaluación de avance: 02/07/2025
 - Segunda evaluación de avance: 16/07/2025
 - Evaluación de Proyecto: 13/08/2025
 - 1er recuperatorio: 27/08/2025
 - 2do recuperatorio 11/02/2026
- Curso 512 - Lunes
 - Entrega de consignas: 23/06/2025
 - 1ra evaluación de avance: 30/06/2025
 - 2da evaluación de avance: 14/07/2025
 - Evaluación de Proyecto: 11/08/2025
 - 1er recuperatorio: 25/08/2025
 - 2do recuperatorio: 09/02/2026

III. CUESTIONARIO TEÓRICO

Las siguientes preguntas deben contestarse y entregarse en un informe aparte.

1) Sistemas Operativos - Generalidades

- a) ¿Quién es el encargado de administrar el tiempo de la CPU y cuál es su función principal dentro del sistema?
- b) Explique a que se denomina “Contexto de Ejecución”. ¿Cuáles son las variables intervinientes?
- c) ¿A qué se denomina Sistema Operativo de Tiempo Real? ¿Por qué usar un Sistema Operativo de Tiempo Real?

2) Kernel de FreeRTOS

- a) Explique y desarrolle las características de una “Tarea” en FreeRTOS. Proponga un ejemplo de una tarea manteniendo la sintaxis correcta.
- b) Explique y desarrolle los “Estados de una Tarea” y sus transiciones. Indique las funciones que permiten las transiciones.
- c) ¿A qué nos referimos con el concepto de “tareas de procesamiento continuo”? ¿y a qué nos referimos con el concepto de “tareas manejadas por eventos”?
- d) ¿Cuáles son los tipos de eventos por los que una tarea puede estar esperando al entrar en el estado bloqueado? Explicar.
- e) ¿Cuál es la función de la IDLE TASK? ¿Puede el procesador no estar ejecutando ninguna tarea en algún momento? Explicar.

3) Desarrolle cómo es el almacenamiento de datos en el recurso “Cola” proporcionado por el Kernel.

4) Explique utilizando un ejemplo con código el porqué es común que el recurso Cola posea múltiples tareas escritoras y una sola tarea lectora. Explique el proceso de bloqueo por escritura y bloqueo por lectura en el recurso Cola.

5) A la hora de enviar datos compuestos: explique cómo se lleva a cabo dicho proceso y por qué no genera conflictos

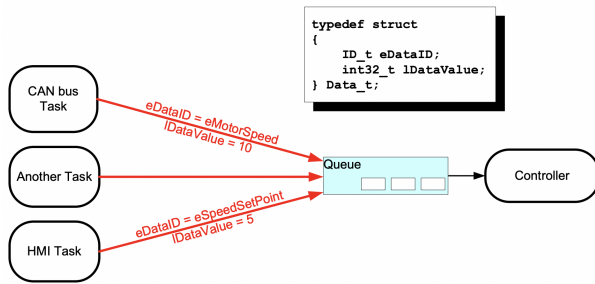


Fig. 1. Recibir datos de múltiples fuentes

con la definición de la macro del recurso. ¿Cuál sería la implementación si los datos fueran “grandes”, lo que ocasionaría un problema con la memoria RAM?

- 6) Explicar el funcionamiento del esquema de la fig 1:
- 7) Desarrollar un programa ejemplo que permita comprender el concepto y uso de “Semáforos Mutex”.
- 8) Desarrollar un programa ejemplo que permita comprender el concepto y uso de “Inversión de Prioridad”. Como mínimo se deben utilizar dos tareas y una cola gestionando un recurso del microcontrolador.
- 9) Explicar el concepto de “Herencia de Prioridad” utilizando diagramas gráficos que permitan comprenderlo.
- 10) Desarrollar un programa ejemplo que permita comprender el concepto de “Punto Muerto”. Como mínimo se deben utilizar dos tareas y una cola gestionando un recurso del microcontrolador.

IV. DESARROLLO PRÁCTICO

Como se mencionó antes, la evaluación tendrá una componente fuertemente práctica. En esta sección, se definen las condiciones, objetivos y alcances esperados de esta parte de la EGA.

A. Condiciones

La etapa práctica de la evaluación deberá completarse en su totalidad para poder acceder a la defensa del proyecto. El proyecto debe haber sido desarrollado en su totalidad como especifica la consigna asignada para ser considerado aprobado.

El proyecto debe:

- Integrar conceptos de FreeRTOS vistos en la cursada y cuidar las buenas prácticas y usos de trabajar con un RTOS.
- El prototipo electrónico debe ser desarrollado en un PCB (no placa experimental) con test points apropiados.
- Debe dejarse libre una línea de I2C y UART con GND disponibles para la EGB en la segunda mitad del año.
- El PCB debe tener un membrete que diga “TD3 / AÑO / APELLIDO”.

B. Consignas

A continuación, se definen las posibles consignas para desarrollar la parte práctica de la EGA. Estas se asignan a cada grupo por los docentes de la cátedra. En todos los casos, se debe implementar un control PID.

1) *Control de temperatura:* Diseñar un sistema de control de temperatura en donde se pueda regular el setpoint con algún conjunto de teclas, botones o potenciómetro. El sistema deberá controlar la temperatura en un ambiente cerrado con un error acotado, teniendo la posibilidad de calentar y refrigerar el ambiente.

2) *Control de posición de motor DC:* Diseñar un sistema de control en el que se pueda regular la posición de un motor de DC. Se debe proveer la posibilidad de ingresar o cambiar el ángulo deseado del motor. Además, se debe planear la posibilidad de visualizar y contrastar el ángulo actual del motor.

3) *Control de luz ambiente:* Diseñar un sistema de control en el que se regule la intensidad lumínica de un ambiente cerrado. La luz será sensada y contrastada con la que se ingrese a través de algún teclado o conjunto de botones. Debe poder visualizarse la intensidad lumínica actual.

4) *Carga electrónica programable:* Diseñar un sistema de control en el que se pueda programar por teclado la resistencia que va a presentar la carga electrónica frente a un circuito, independientemente de la tensión aplicada. El sistema debería tener una forma de ver la tensión aplicada y la resistencia que presenta.

5) *Control de velocidad de motor DC:* Diseñar un sistema de control en el que se pueda regular la velocidad de un motor de DC con un potenciómetro o teclado. Debe además poder controlarse la dirección de giro, teniendo algún indicador de esta.

6) *Control de nivel programable:* Diseñar un sistema de control en el que se pueda regular el ángulo de una superficie y que pueda mantenerse constante frente a perturbaciones. El ángulo debe poder programarse por una entrada de teclado o botones y debe poder verse el ángulo medido actualmente de forma constante.

7) *Control de fuente elevadora de tensión:* Diseñar un sistema de control en el que se pueda regular, a partir de 5V, la tensión de salida de una fuente switching elevadora de 12V a 24V. La tensión de salida debe ser programable y debe poder verse la tensión de salida actual en todo momento.

8) *Péndulo invertido:* Diseñar un sistema de control estilo péndulo invertido o equilibrista. El sistema deberá buscar mantener un péndulo en posición vertical o un balancín de pie. Debe poder programarse la tolerancia aceptable para el sistema.

9) *Levitador neumático:* Diseñar un sistema de control que permita sostener en el aire un elemento liviano como una pelota de telgopor. El sistema debe poder tener la posibilidad de regular a través de un potenciómetro o teclado la altura que se desea mantener y regular la intensidad de un cooler para contenerlo.

Importante: En todos los casos se debe proveer una forma de almacenar la información de la consigna (valores principales) en forma de datalogger. La forma y los parámetros son a elección del grupo.

V. DEFENSA DEL PROYECTO

El grupo entero que desarrolló el prototipo presenta y hace una defensa oral individual de lo desarrollado. En esta defensa, se busca validar los conceptos, que en forma integral, fueron aplicados en la consigna.

En la Evaluación de Proyecto se debe hacer entrega de toda la documentación y desarrollo en GitHub como se describe en la sección siguiente. Como parte de la defensa, se debe desarrollar y entregar un informe en IEEE describiendo el funcionamiento del proyecto.

A. Características del informe

El informe debe ser desarrollado en formato IEEE y debe contener por lo menos los siguientes ítems:

- Índice.
- Abstract.
- Fundamentación teórica de hardware y software.
- Alcance propuesto como solución.
- Diagrama en bloques (indicando todos los pines de interconexión entre módulos y microcontrolador).
- Sobre el hardware. Debe incluir especificaciones técnicas, valores esperados, rangos de medición y precisión obtenida de cada módulo, sensor o dispositivo utilizado. Detallar el funcionamiento de cada uno de los dispositivos. Incluir esquemáticos de los circuitos desarrollados y justificación de la elección de componentes donde corresponda.
- Sobre el software. Se debe incluir una guía de código especificando la relación entre funciones, tareas y recursos del RTOS. Debe ser desarrollada en una carilla individual y puede ser en formato de máquina de estado o diagrama de flujo.
- Diseño de PCB. Debe incluir la lista de test points.
- Conclusiones que representen el nivel del alcance efectivamente comprobado. Indicar posibles mejoras y situaciones no previstas.
- Bibliografía, librerías, referencias, link al video funcionando.
- Manual de Usuario o Manual de Funcionamiento Técnico según corresponda. Puede ser desarrollado fuera del formato IEEE.

VI. ENTREGA

A. Entregas parciales

A lo largo de la EGA, se realizarán entregas parciales en las fechas pautadas. Cada una de esas entregas tendrá una serie de requisitos que se describen a continuación.

1) *Primer entrega:* Esta entrega, denominada v1 para la entrega por GitHub, requerirá que se entregue:

- Diagrama en bloques completo con interconexión de pines, con la propuesta para resolver la parte práctica.
- Avance en el firmware que puede incluir: pruebas individuales con sensores/actuadores baremetal o con FreeRTOS.

2) *Segunda entrega:* Denominada v2 para la entrega por GitHub, requerirá:

- Esquemático y PCB diseñado (PDF y proyecto del software usado).
- Tarea Guardiana de cada dispositivo hardware en versión ensayo.
- Guía de código tentativa con mapa de tareas propuesto del código principal.
- Cualquier corrección o reversión de la etapa anterior.

3) *Evaluación de Proyecto:* Denominada `evaluacion_proyecto` para la entrega por GitHub, será la entrega final que deberá incluir:

- Cualquier corrección o ajuste a lo entregado en las etapas anteriores.
- Resolución del cuestionario teórico en su documento separado (versión en PDF).
- Informe final que respete los requisitos establecidos anteriormente.
- Versión final del firmware implementado.
- Cualquier anexo, referencia, diagrama mecánico, 3D o archivos adicionales que formen parte del desarrollo del proyecto.
- Link al video donde se demuestre el funcionamiento del prototipo.

B. Convenciones de Git

La entrega deberá ser realizada por GitHub dentro del repositorio forkeado de la cátedra. Dentro del repositorio personal, se deberá trabajar en una rama con el nombre **ega/vx** donde x es el número de versión que se está entregando, siendo 1 para la primera entrega, 2 para la segunda y así sucesivamente. Ejemplo:

```
git checkout -b ega/v1
```

La rama deberá luego ser pusheada al repositorio personal de forma individual y, eventualmente, realizar un pull request al repositorio de la cátedra.

```
git push origin ega/v1
```

El pull request debe hacerse desde la rama **ega/vx** a la rama personal dentro del repositorio de la cátedra. El nombre de la rama a la que solicitar el pull request tiene el nombre **5xx_apellido_nombre** donde xx es el curso correspondiente. **No hacer pull request a la rama main.**

El título del pull request debe decir "EGA vx" reemplazando x por la versión que se esté entregando. Debe proveerse una descripción en la pull request de qué se está entregando con la pull request.

Serán rechazadas todas las entregas que no cumplan con los requisitos mencionados.