

PRÁCTICA 3: DISEÑO DE UNA PLATAFORMA INDEPENDIENTE

INTRODUCCIÓN

Con esta práctica pretendemos que completéis el trabajo final, acercándonos al máximo a un juego completo autónomo y funcionando correctamente en el procesador LPC2105.

OBJETIVOS:

El objetivo principal de esta práctica es finalizar el proyecto en el que habéis estado trabajando para conseguir un sistema empotrado. Particularmente, nos fijaremos los siguientes objetivos de aprendizaje:

- Diseñar e implementar una gestión completa, portable y robusta de periféricos.
- Componer una arquitectura de sistema que aproveche los modos del procesador.
- Implementar y usar llamadas al sistema para interactuar con un dispositivo, un temporizador, y para activar y desactivar las interrupciones.
- Concebir e implementar un protocolo de comunicación robusto para el puerto serie.
- Identificar condiciones de carrera y su causa, así como solucionarlas.
- Realizar un juego completo haciendo uso de la estructura modular.

REQUISITOS FUNCIONALES Y NO FUNCIONALES

RF1: El juego deberá iniciar y terminar de manera ordenada.

RF2: No habrá en el juego ninguna espera activa salvo las condiciones de overflow.

RF3: Se gestionarán como llamadas al sistema la activación y desactivación de las rutinas de servicio de periféricos y la lectura del tiempo transcurrido.

RF4: La entrada de la jugada se realizará a través de línea serie. El programa enviará al usuario/a el tablero para visualizar los movimientos por pantalla.

RF5: Se dará opción de confirmar y cancelar mediante los botones.

RF6: Al final de la partida, se presentarán métricas de rendimiento y calidad de servicio a través de la línea serie.

RF7: En caso de mal funcionamiento el sistema se reseteará de forma autónoma.

RNF1: Se puede depurar con nivel de optimización O0, pero la versión final debe funcionar correctamente en modo *release* con **O3**.

CONECTA K

Si hemos realizado correctamente todos los módulos del entorno, nos debería costar muy poco realizar sobre él cualquier aplicación o juego.

En esta última práctica vamos a incluir toda la lógica para jugar al Conecta K. Debéis incluir el código necesario rescatándolo de la primera práctica. Las funciones para comprobar la jugada deben ser aquellas que en la P1 os daban el mayor rendimiento.

Es preciso que diseñéis la máquina de estados y la entrada salida del juego que cumpla todos los siguientes requisitos:

- Al inicio de la partida, el programa mostrará por línea serie información sobre el juego y cómo jugar mientras se espera la pulsación de cualquier botón o llega un nuevo comando de nueva partida.

Los comandos aceptables codificados en ASCII son:

- Acabar la partida: \$END!
- Nueva partida: \$NEW!
- Jugada: \$#-#! donde #-# será el número de fila-columna codificado con dos caracteres (1..7, 1..7). Ejemplo: el comando \$2-3! Indica la fila 2 y la columna 3.

- Las entradas y salidas de las prácticas anteriores se mantienen salvo que sean modificadas en las siguientes especificaciones.
- A continuación, aparecerá el **tablero** por línea serie en la consola, el formato lo dejamos libre, pero deberéis intentar que se vea de la mejor forma posible para facilitar el jugar. En la primera partida, y para poder depurar y presentar el proyecto, se empezará con el tablero de test de P1 y turno del jugador uno.
- Se indicará, en cada momento a que jugador le corresponde jugar.
- Vamos a incluir la llegada de comandos por línea serie y mantener las opciones de aceptar o cancelar por botones desarrollados en la práctica 2. La fila y columna se recibirá por línea serie. Si el comando recibido o la fila-columna **no es válida** se ignorará, se escribirá en la pantalla el mensaje "Columna no valida" o "Comando erróneo" y se encenderá el led GPI029 (siguiendo el patrón de reservas y comunicación al módulo). En caso contrario, Se mostrará el tablero con un carácter especial para la nueva ficha y el mensaje "Pulsa botón 1 para cancelar". El jugador dispondrá de 3 segundos para cancelarlo pulsando el **botón 1** (antes se usaba con otro significado). Si se cancela se escribirá en la pantalla el mensaje "Movimiento cancelado" y se volverá a restaurar el tablero anterior.
- Si se realiza el movimiento, se enviará de nuevo el tablero por la línea serie y se cambiará de turno de jugador.
- La recepción del comando de final de partida o la pulsación del botón 2 indicará que el jugador al que le toca el turno se rinde y terminaremos la partida actual.
- Al terminar la partida, se mostrará la siguiente información:
 - o Causa por la que se termina (botón de reiniciar, victoria de uno de los jugadores, ...)
 - o Tiempo total de uso de procesador en esta partida (sin power-down).
 - o Total y media de tiempo de computo de `conecta_K_hay_linea`.
 - o Total y media de tiempo que al humano le cuesta pensar jugada.

- Total de eventos encolados en esta partida e histograma por tipo de evento.
- Además, se mostrará la leyenda de volver a jugar. Si el usuario decide volver a jugar se empezará una nueva partida (con el tablero inicial vacío y alternando el turno del jugador inicial).
- Si el planificador pasa más de un segundo sin procesar ningún mensaje o evento se reseteará el sistema. Teniendo en cuenta que estamos generando interrupciones y mensajes periódicos, esto sólo ocurrirá si el sistema se ha quedado colgado.
- Queremos seguir manteniendo las indicaciones de desbordamiento de la pila y las alarmas y el hello_world (modificando ioreserva.h)

Debéis presentar el diagrama de la máquina de estados, tanto para la corrección, como para la memoria.

Respetad la modularidad, p.e. el planificador no sabe a qué está jugando, no conoce el tablero ni la columna. Se mantienen las alarmas, el hello_word, las políticas de bajo consumo, los overflow, etc. La funcionalidad del demostrador de la cuenta de la P2b no es necesario mantenerla.

Reevaluar el rendimiento y determinar cuáles son ahora las funciones más costosas.

APARTADOS OPCIONALES:

- Podéis añadir Toolboxes conectadas al GPIO que emulen botones

MATERIAL DISPONIBLE:

En la página de la asignatura en Moodle podéis encontrar códigos de ejemplo. Os recordamos, podrán aparecer indicaciones a lo largo de la práctica, se recomienda consultar Moodle con frecuencia y en particular la información que vaya apareciendo en la wiki.

EVALUACIÓN DE LA PRÁCTICA

Esta práctica se presentará durante la última semana de clase (semana 15 EINA), se publicarán los detalles en Moodle. La memoria se presentará el 8 de enero.

La evaluación de esta práctica y la presentación podrá ser individual. Cada miembro del grupo debe ser capaz de explicar cualquier parte del diseño. La memoria incluirá el trabajo realizado en la práctica 2 y la práctica3.

PRESENTACIÓN EN DICIEMBRE

Debéis entregar previa a la corrección en el laboratorio del proyecto completo comprimido en .zip a través de Moodle. En un fichero comprimido Correccion_P3_NIP-Apellidos_Alumno1_NIP-Apellidos_Alumno2.zip. Los detalles de los horarios de corrección se publicarán en Moodle.

ENTREGA FINAL

Debéis entregar dos ficheros, la memoria y el proyecto.

La memoria en formato pdf El archivo que contiene la memoria debe llamarse

p2-3_memoria_NIP-Apellidos_Alumno1_NIP-Apellidos_Alumno2.pdf

El proyecto Keil incluiría todos los ficheros y se entregará comprimido en un fichero zip. El fichero debe tener el siguiente nombre

p2-3_NIP-Apellidos_Alumno1_NIP-Apellidos_Alumno2.zip

Por ejemplo: p2-3_memoria_345456-Gracia_Esteban_45632-Arribas_Murillo.pdf y p2-3_345456-Gracia_Esteban_45632-Arribas_Murillo.zip

ANEXO 1: REALIZACIÓN DE LA MEMORIA FINAL

Esta memoria incluirá el trabajo realizado en las prácticas 2 y 3. La memoria de las prácticas tiene diseño libre, pero debe seguir la estructura de una memoria técnica (recomendable revisar el documento Redacción de una Memoria Técnica, disponible en la web de la asignatura). Es obligatorio que incluya al menos los siguientes contenidos (**esto no es la estructura** que debéis seguir **sino una lista de requisitos**):

1. Resumen ejecutivo (una cara como máximo). El resumen ejecutivo es una página independiente al inicio de la memoria que describe brevemente qué habéis hecho, por qué lo habéis hecho, qué resultados obtenéis y cuáles son vuestras conclusiones. **Escribidlo al acabar la memoria como si fuese un documento independiente.** No lo uséis como introducción.
2. Estructura de : Introducción, objetivos, metodología, resultados y conclusiones
3. Código fuente comentado (en un **anexo**). Sólo hay que incluir el que habéis desarrollado nuevo y sea **relevante** de las prácticas 2 y 3. Como siempre, cada función debe incluir una cabecera en la que se explique qué hace, qué parámetros recibe....
4. Descripción de los problemas encontrados en la realización de la práctica y sus soluciones.
5. Resultado final: incluyendo test de funcionamiento realizados y medidas de rendimiento final.
6. **Conclusiones y autoevaluación** de vuestro proyecto final. ¿Qué nota os pondríais cada uno de vosotros y por qué? También agradecemos, aunque no es obligatorio, recibir retroalimentación y cualquier crítica constructiva de la asignatura (a través de las encuestas, Moodle, correo electrónico o en persona).
7. Si las hubiera, las correcciones solicitadas en las entregas previas.

Se valorará que el texto sea **claro y conciso, se recomienda una extensión menor de 15 páginas, no se podrán entregar memorias extensión superior a 25 páginas**, sin contar portada, índice y los anexos, con el correspondiente rigor técnico. Podéis incluir en la memoria partes relevantes de vuestro código que sean ilustrativas.