Carrera de Especialización en Sistemas Embebidos
Sistemas Operativos en Tiempo Real 2.

Trabajo Práctico 1:

El objetivo de esta práctica es aprender a trabajar con algoritmos de asignación de memoria en un contexto de ejecución de tiempo real.

Estos algoritmos están orientados a tener un tiempo de ejecución corto y determinista.

No se pretende que el alumno escriba un asignador de memoria dado que hay numerosas implementaciones disponibles en la red. Se espera que el alumno justifique su elección del algoritmo de asignación de memoria.

Descripción conceptual de la práctica:

Basándose en el driver de USART de sAPI, se deberá implementar un driver de mayor nivel, que permita separar paquetes entrantes.

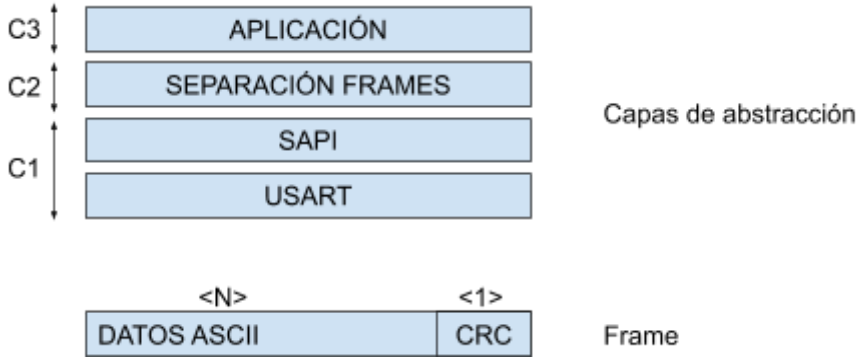
El criterio de separación será por tiempo; Es decir que cuando se reciba un byte y luego pase un cierto tiempo, ese paquete deberá "cerrarse".

El protocolo de esta capa de separación de frames, deberá validar el paquete entrante, siendo el último byte recibido un código de verificación que utilice el algoritmo crc8 sobre el contenido del paquete recibido (en caso de que no coincida, descartar)

El procesamiento de estos paquetes deberá ocurrir en contexto de ISR.

Al recibir un paquete, la capa de separación informará a la aplicación de la llegada de un paquete nuevo.

La aplicación deberá tener la única responsabilidad de recibir paquetes, procesarlos y enviar una respuesta.



La aplicación deberá validar el contenido del paquete. Los paquetes deberán contener solo texto y deberá procesarlo de manera que transforme todas las letras minúsculas en mayúsculas. El texto en mayúscula deberá posteriormente ser enviado a la capa de framing y está, en el sentido contrario será la responsable de agregarle el campo de verificación.

En caso de que algún caracter recibido no sea válido, la aplicación deberá responder un mensaje de error.

Carrera de Especialización en Sistemas Embebidos
Sistemas Operativos en Tiempo Real 2.

Requerimientos:

Del TP:

- Todas las justificaciones deberan estar plasmadas en un archivo readme.md (markdown). Deberá estar versionado con el código fuente.
- El alumno deberá justificar la elección de las arquitecturas que utilice.
- El alumno deberá justificar la elección del esquema de memoria dinámica utilizada.

Generales:

- Cada capa de abstracción deberá estar diseñada de forma tal que se pueda instanciar (y eventualmente se pueda reutilizar varias instancias dentro de un mismo Firmware, por ej, para utilizarla con varias USARTs).

Capa separación de frames (C2).

- Deberá procesar en contexto de ISR todos los bytes entrantes y los salientes.
- Los bytes entrantes, deberán almacenarse en un bloque de memoria dinámica.
- Deberá tener control sobre la máxima cantidad de bytes recibidos.
- Al elevar un paquete recibido a la capa de aplicación (C3), la C2 deberá poder seguir recibiendo otro frame en otro bloque de memoria dinámica, distinto al anterior.
- En caso de no haber memoria, la recepción de datos del driver de la C1 deberá anularse.
- Cada frame de datos estará separado por un tiempo T=3ms.
- El timeout deberá implementarse con un timer de FreeRTOS
- En caso de que el código de verificación no sea validado, el paquete deberá descartarse.
- Al recibir un mensaje correcto, se deberá señalizar a la aplicación de su ocurrencia, para ser procesada.
- Cuando la aplicación (C3) desee enviar un mensaje por el canal de comunicación, C2 deberá agregarle el codigo de comprobacion.
- La transmisión de las respuestas al canal, deban cumplir la misma premisa que en la recepción: Entre paquetes debe existir un tiempo muerto de T=3ms:
- Al final la transmisión, se deberá liberar la memoria dinámica utilizada para la transacción.

Capa de aplicación (C3)

- Deberá procesar solo paquetes de texto a-z y A-Z
- Si hay paquetes con caracteres que no cumplan, deberán descartarse, y enviar a la capa C2 el texto ERROR
- El texto entrante deberá pasarlo a mayúscula.

Opcionales:

- En ocasiones, los protocolos que separan frames por tiempo, definen timeouts de frame en función del baudrate (ej MODBUS RTU). En algunos casos, el ms no es una unidad válida porque el TO resulta ser más chico. Implementar el ejercicio SIN timers de FreeRTOS, utilizando un timer de hardware.

Carrera de Especialización en Sistemas Embebidos
Sistemas Operativos en Tiempo Real 2.

Sugerencias:

- Deténgase a pensar la arquitectura papel, modularizado y asignando responsabilidades a cada módulo.
- Implementar los requisitos de a uno y verificar el funcionamiento de los mismos antes de pasar al siguiente.
- Implementar el procesamiento de datos con una FSM.

Historial de cambios

Rev	Fecha	Autor	Detalle
0	2019/09/06	Franco Bucafusco	Creación del documento