



Lab 5: UART - Buffers y recepción de caracteres

1. Objetivo

En general, cuando se realiza comunicación entre dos dispositivos, es común esperar que esta sea de carácter bidireccional. En este sentido, la “bidireccionalidad” de la comunicación apunta a que los dispositivos sean capaces tanto de transmitir como de recibir información. Siguiendo la línea de protocolos de comunicación del laboratorio anterior, la presente experiencia tiene por objetivo el introducir al alumno a la recepción de datos mediante el protocolo USART, mediante el uso de estructuras complejas basadas herramientas como la gestión de memoria (punteros) y *buffers*.

Al final de la experiencia, se espera que los estudiantes sean capaces de manejar las funciones de Transmisión y Recepción del protocolo USART, a la vez que también sean capaces de generar estructuras complejas (concretamente, *linked lists*), capaces de gestionar la información a transmitir o recibir.¹

2. Descripción de la actividad

Para realizar esta actividad se le ha proporcionado el archivo `USART_implement_me.c`, el cual deberá completar con el código faltante. El archivo en cuestión contiene dos nuevas funciones para su librería de comunicación USART, las cuales, en caso de ser completadas correctamente, le permitirán la recepción de un carácter o *char* (función 1) y la recepción de un *string* completo (función 2).

Una vez que complete estas funciones, deberá implementar listas enlazadas o *linked lists*, las cuales deberá utilizar como *buffer* de recepción y transmisión para recibir una cantidad considerablemente grande de texto y luego desplegarlo en la pantalla del terminal. Las *linked lists* son estructuras de suma utilidad para almacenar cantidades grandes e indefinidas de información y, como son comúnmente utilizadas, existe una gran variedad de información sobre estas en Internet. **Es fundamental que estudie y comprenda estas estructuras, pues en caso de no hacerlo la actividad puede volverse bastante complicada**². Es esencial que entienda cuales podrían ser las razones para utilizar estas estructuras y cuales

¹En la vida laboral, los protocolos de comunicaciones entre fabricantes funcionan o no funcionan. No hay términos medios

²Ponga especial atención a los algoritmos de *push* y *pop*, esenciales para manejar estas estructuras



son las ventajas en comparación con otros métodos como *buffers* lineales y *ring buffers*.

Para esta actividad, **la elección del microcontrolador es libre, y solo es necesario realizarla en uno de ellos para cumplir con la aprobación**. No obstante, es altamente recomendable que lo implemente en ambos microcontroladores, teniendo en consideración los futuros laboratorios. En cuanto a los parámetros de comunicación serial, tales como el *baud rate*, *data bits*, *parity*, etc., estos quedan a libre elección, siempre y cuando le permitan observar un correcto funcionamiento de la tarea.

NOTA: Es importante notar que a la función de `USART_init()`, se le debe agregar una rutina de inicialización para la recepción de caracteres en USART. Esto asumiendo que no implementó esto antes.

3. Lectura recomendada

- [ATmega328/P Complete Datasheet](#).
- [MSP430x5xx and MSP430x6xx Family User's Guide](#).
- [MSP430F552x, MSP430F551x Mixed-Signal Microcontrollers datasheet](#).
- [Tutorial de C](#): material de utilidad para iniciarse en el mundo de la programación en C.



4. Pauta de Evaluación

4.1. Consideraciones generales

- El laboratorio será evaluado exclusivamente con nota 1.0 (**R**eprobado), 5.5 (**A**probado) y 7.0 (**D**istinguido). En ningún caso habrán notas intermedias.
- No se reciben trabajos después del módulo de presentación. Trabajos no entregados son calificados con nota 1.0 y son considerados dentro del criterio de aprobación del curso.
- Cualquier consulta sobre los criterios de evaluación de cada laboratorio debe ser realizada en las **issues**, donde estará disponible para que sea revisada por todos los alumnos.

4.2. Criterios de Aprobación

Se requiere cumplir con todos los puntos mencionados a continuación para poder aprobar. No existen casos excepcionales.

1. Funcionamiento de los requerimientos. El alumno realiza una presentación de su trabajo y se responsabiliza de exponer que su trabajo satisfaga todos los requerimientos mínimos solicitados en la *Descripción de la actividad*, los cuales incluyen en este laboratorio:
 - Implementación adecuada de las librerías de acuerdo a las especificaciones.
 - Debe idear cómo enfrentar cuando el mensaje a recibir supera al tamaño del *buffer* de recepción, existen muchísimas soluciones para eso, en esta ocasión deberán implementar una *linked list*
 - **NO se aceptan soluciones del estilo de cortar el mensaje recibido o realizar un *ring buffer*.**
 - Se controlará este requisito también en el Laboratorio 6, razón por la cual puede resultar doblemente reprobatorio fallar en la implementación de este. Más aún, lograr este requisito en este Laboratorio trivializa significativamente el próximo.
2. Preguntas. Se responde satisfactoriamente a 2 de 3 preguntas aleatorias al momento de la presentación final, las cuales abarcan los siguientes temas:
 - Qué representa cada línea de código y en qué se traducen en el funcionamiento del programa.



- ¿Qué es una *linked list*?, ¿Cómo se implementa?
- ¿Qué hacen las funciones *malloc* y *free*?

Solo se dispone de una oportunidad para responder estas preguntas. Fallar en este requisito se traduce en la reprobación inmediata de la experiencia de forma inapelable.

4.3. Criterios de Distinción

La distinción representa un trabajo adicional que sobresale a los requerimientos mínimos para la aprobación. Agregados adicionales no constituyen por sí mismo una distinción si no representan un verdadero trabajo adicional de comprensión y/o análisis.

Los trabajos distinguidos pueden caer (no exclusivamente) en algunas de las siguientes líneas generales:

- Funcionalidades creativas :D
- Trabajos sobresalientes en la línea de *timers*, I2C, SPI u otras funcionalidades no vistas.
- Comunicación entre microcontroladores: en este caso tenga especial cuidado con las salidas de alto voltaje de ATmega328P, pues son de 5V y si los conecta directamente a MSP430F5529 **PODRÍA QUEMAR LA TARJETA**. (En caso de comunicar, debe demostrar de alguna forma visual que la comunicación se está realizando).
- Interrupciones: en este caso, las rutinas de interrupción deben ser hechas a conciencia y conociendo a cabalidad su funcionamiento. (Además, podría adelantar trabajo para laboratorios futuros). El uso de interrupciones sin comprensión no calificará como distinguido.
- **Portabilidad de código:** Un código es portable si el código fuente en C del laboratorio puede ser compilado y cargado en cualquiera de los microcontroladores del curso de forma indistinta, **sin hacer ninguna modificación a dicho código**³.

Las Distinciones son discutidas caso a caso por la totalidad del equipo de ayudantes al finalizar la corrección del laboratorio. Serán notificadas públicamente después del módulo de evaluación.

³Hint