Lab 6: UART - Interrupciones

1. Objetivo

En las experiencias anteriores usted implementó métodos de comunicación basados en consultar constantemente el estado de variables (*polling-based communication*) o de esperar tiempos muertos hasta cumplir una condición (*busy-waiting*). Estos métodos resultan ser muy poco eficientes, pues mantienen completamente ocupado al MCU en la transmisión o recepción (dentro de *loops* infinitos).

El siguiente paso lógico es romper con estas limitaciones, a modo de poder aprovechar al máximo las funcionalidades del microcontrolador, utilizando los tiempos entre cada comunicación para llevar a cabo otra tarea. Este proceso puede ser realizado mediante el uso de interrupciones, las cuales corresponden a rutinas que pueden ser configuradas para gatillarse cuando algún evento de interés ha ocurrido (por ejemplo, presionar un botón o recibir un *char* desde el terminal, o estar listo para enviar uno). De este modo, podemos **reaccionar** ante eventos, en lugar de esperar a que estos sucedan.

El presente laboratorio tiene por objetivo que usted se relacione con el manejo y configuración de interrupciones, haciendo uso de estas en las rutinas de transmisión y recepción de datos implementadas en sus anteriores laboratorios de comunicación serial.

2. Descripción de la actividad

2.1. *Task 1:*

La actividad consistirá en recibir y transmitir datos a partir de la interacción entre una interfaz serial y su microcontrolador, por medio de comunicación UART implementada con interrupciones, además de detectar por medio de interrupción de cambio de pin, si el botón de la placa fue apretado o soltado, enviando a terminal el mensaje que corresponda según el caso. Es importante recalcar que este mensaje también debe ser enviado por interrupciones, no dentro de un ciclo while recorriendo el string como se realizaba en laboratorios anteriores.

En otras palabras, la tarea a realizar consiste en la unión de dos sub-tareas:

2.1.1. *Task 1.1:* Interrupción de cambio en un pin

Para llevar a cabo la siguiente *task* será conveniente que reutilice las librerías USART que ha implementado en sus laboratorios anteriores. Configure una interrupción de cambio de estado de pin (PCINT) para el botón de la placa, de modo que el microcontrolador envíe un mensaje mediante interrupciones al terminal serial al detectar un cambio en el estado del botón. El mensaje en cuestión debe indicar que se ha detectado un cambio en el pin, adicionalmente, debe **indicar la dirección del cambio, es decir, si el botón fue presionado o soltado**.

NOTA: Para que las interrupciones funcionen deberá activarlas de manera global. Es altamente probable que su comunicación UART repentinamente comience a comportarse extraño. Si esto pasa, le recomendamos poner atención a su función de inicialización USART_Init(), ¿podría ser que accidentalmente configuró interrupciones USART en los laboratorios anteriores?. De ser así, en ese entonces no tuvieron efecto, pero ahora podrían ser un problema ya que las interrupciones fueron globalmente activadas.

2.1.2. *Task 1.2:* Comunicación UART por interrupciones

Para esta actividad deberá realizar modificaciones a la librería USART desarrollada en sus laboratorios anteriores. En primer lugar modifique su función de recepción de caracteres individuales para que esta opere mediante interrupciones. Para lograr esto, deberá gatillar una interrupción cada vez que se envíe un byte al microcontrolador desde el terminal serial. Dado que el byte en cuestión puede llegar de imprevisto, su código no podrá procesarlo inmediatamente, por lo que deberá almacenarlo en un *buffer* hasta que llegue el momento de utilizarlo. Una vez que logre esto, adapte el código para que también pueda recibir *strings* completos. Una vez que consiga implementar las interrupciones de recepción, repita el mismo procedimiento con las de transmisión. Con lo anterior, deberá enviar un string únicamente al momento de ser recibido completamente, para así realizar un echoback del mensaje.

Para revisar que su *Task 1* está siendo realizada adecuadamente en interrupciones, su ciclo while(1); deberá estar vacío.

3. Lectura recomendada

- ATmega328/P Complete Datasheet.
- MSP430x5xx and MSP430x6xx Family User's Guide.
- MSP430F552x, MSP430F551x Mixed-Signal Microcontrollers datasheet.

4. Pauta de Evaluación

4.1. Consideraciones generales

- El laboratorio será evaluado exclusivamente con nota 1.0 (**R**eprobado), 4.0 (**S**uficiente), 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. La hora límite para inscribir a revisión es a las 10:10 hrs, posterior a esto se asignará una posición aleatoria.
- La nota **S**uficiente se otorgará en el caso de falla de una de las tareas de este laboratorio, quedando a criterio del ayudante. En caso de que un alumno haya decidido solamente hacer un 50 % del trabajo, se evaluará con un 1.0.
- Respecto a los puntos de aprobación acumulados, si un alumno obtiene una nota **S**uficiente, una mitad del puntaje queda asignada a Aprobación, el restante a Reprobación.

  A modo de ejemplo, este Laboratorio es nivel 3, si un alumno tiene Suficiente, 15 pts se acumularán a Aprobación y 15 pts será para Reprobación.
- 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:

   - Funcionamiento de interrupción en un botón.
   - Implementación adecuada de las librerías de acuerdo a las especificaciones.
   - Debe utilizar la estructura de *linked list* para los *buffers* de Tx y Rx.
   - Ambos Tx y Rx deben ser implementados mediante ISR.
   - Mostrar el funcionamiento en otra configuración (realizadas en los laboratorios anteriores de USART), a determinar por el ayudante. Esté preparado para recompilar su código rápidamente de ser necesario
   - Las tareas deben ejecutarse de manera concurrente. Es decir, ambas deben estar contenidas en un solo archivo que se carga al microcontrolador. No se aceptan las tareas por separado.

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.
   - Funcionamiento del sistema de interrupciones del microcontrolador especificado. Teoría de operación, registros utilizados, etc
   - 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
- Funcionalidades adicionales, en la línea de I2C, SPI, EEPROM, entre otros.
- 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**. (si realizó esto en el laboratorio anterior, debe utilizar una idea distinta).
- **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**[1].

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.

[1] Hint