



Lab 6: UART, Comunicación entre microcontroladores

Nivel 4

1. Introducción

Este laboratorio hace uso de una de las características relevantes de los microcontroladores: La Comunicación Serial. Esta funcionalidad permite transmitir y recibir distintos tipos de contenidos entre diferentes dispositivos, permitiendo un mayor nivel de control que usando simplemente GPIOs o señales analógicas. En esta oportunidad se trabajará con el protocolo de comunicación UART o *Universal Synchronous and Asynchronous Receiver and Transmitter*, variando entre comunicación de un computador a microcontrolador para finalizar con la comunicación entre microcontroladores.

En sentido, cuando se realiza comunicación entre dos dispositivos, es común esperar que esta sea de carácter bidireccional, por lo que se espera que sea capaz de transmitir como de recibir. Para esto, existen diversos métodos, como lo es consultar constantemente el estado de variables (*polling-base communication*) o de esperar tiempos muertos hasta cumplir una condición (*busy-waiting*). Estos métodos resultan muy poco eficientes, pues mantienen completamente ocupado al microcontrolador en la transmisión o recepción (dentro de *loops* infinitos).

Si el objetivo es eliminar estas limitaciones, aprovechando al máximo las funcionalidades del microcontrolador. Este proceso puede ser realizado mediante el uso de interrupciones, las que corresponden a rutinas que pueden gatillarse cuando ocurre algún evento. En el laboratorio anterior se realizaron interrupciones de pines y de timers, por lo que es momento de incorporar la recepción de datos en comunicación serial.

Finalmente, un protocolo de comunicación no es suficiente si solo se envían datos sin sentido, por lo que es importante identificar el mensaje recibido y así, poder tomar decisiones acerca de algún proceso o de información a enviar.

El objetivo de este laboratorio, es lograr la implementación de un pequeño sistema de comunicaciones, en que un dispositivo es el encargado de tomar datos y enviando estos datos a un dispositivo distinto. Este otro dispositivo solo tendrá la capacidad de mostrar la información recibida y consultarla al dispositivo que adquiere los datos. Para lograr esto, la actividad se dividirá en 3 partes:

- Task 1: Adquisición de datos en un AVR y comunicación con un computador.
- Task 2: Mostrar datos en un MSP que fueron recibidos mediante un computador.
- Task 3: Sistema que comunicará el AVR y MSP.



2. Evaluación

Este laboratorio, al completar los requerimientos será evaluado con nota 1.0 a 6.0. Existirá un apartado de bonus para poder lograr el 7.0. La actividad se separa en 3 actividades complementarias, que ayudarán a guiar el trabajo a desarrollar. Para la evaluación online solo se revisará el Task 3, en caso de que algún alumno no logre terminar el Task 3, podrá presentar los task que logró y su nota se calculará en base a eso.

3. Actividad a realizar

La actividad consistirá en la implementación de un sistema de comunicaciones. Por una parte, se tendrá el ATmega328P que medirá la variación de un fotorresistor y un potenciómetro, mediante ADC y enviará este valor por UART. Por otra parte, se tendrá un MSP430F5529, que recibirá la información UART y la desplegará en un display LCD.

3.1. Task 1: Comunicación AVR - PC

En esta primera parte, deberá transmitir el valor medido por el ADC al computador de un fotorresistor y de un potenciómetro digital. Para esto, se deberán implementar dos modos de envío:

3.1.1. Task 1.1: Modo manual (por defecto)

En el modo manual, el computador deberá enviar un pequeño mensaje (definido por usted), que puede ser de 1 o más caracteres en el que se indique qué dispositivo se quiere leer. Dependiendo del que se solicite, el microcontrolador deberá enviar el dato medido por el ADC (10 bits). Este deberá ser mostrado como número decimal en un terminal serial.¹

Requerimientos y recomendaciones:

- Puede implementar un pequeño menú para decidir qué solicitar.
- Se debe utilizar un baudrate igual a 57600, 8 bits de mensaje, 1 bit de stop y sin paridad. Configuración conocida como 57600 8N1.

¹Para llevar a cabo esta labor, necesitará de un programa terminal como minicom, gtkterm, CuteCom o **RealTerm** (los ayudantes recomiendan este). Es de su responsabilidad la elección, instalación y familiarización con el programa que usted escoja.



- Si se solicita el fotorresistor o potenciómetro, se debe enviar el mensaje: LDR: xxxx o POT: xxxx respectivamente. Donde x representa el valor en decimal. Queda a su decisión enviar el valor con ceros o sin ceros para números menores a 1000. Ej. Se acepta LDR: 0052 o LDR: 52.
- Debe completar la función `void USART_Transmit.String(char* string)`
- Se debe ocupar el ADC con resolución de 10 bits. Deben recordar que solo se pueden enviar paquetes de tamaño 8 bits. Por lo que deberá solucionar este inconveniente.
- La recepción se debe realizar mediante el uso de interrupciones de recepción.
- El buffer de recepción queda a criterio del estudiante.

3.1.2. Task 1.2: Modo automático

En el modo automático, el microcontrolador envía constantemente los valores medidos del potenciómetro y del fotorresistor de forma alternada con un delay entre ambos. Para activar o desactivar este modo, se debe presionar el botón de la placa y se procederá a enviar constantemente los valores.

Requerimientos y recomendaciones:

- El botón debe ser reconocido mediante interrupciones de pin.
- Se debe mantener la misma estructura del mensaje del modo manual.

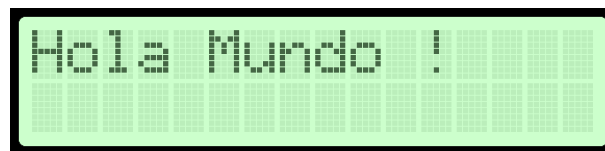


3.2. Task 2: Comunicación MSP - PC

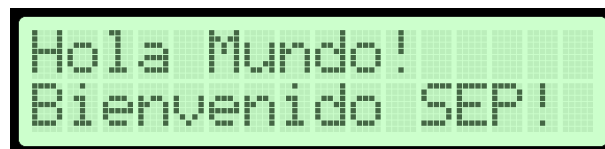
En esta parte, se deberá transmitir un mensaje por UART desde el microcontrolador y ser capaz de mostrarlo en el display LCD. Por otra parte, se deben enviar mensajes del MSP al computador utilizando los botones de la placa.

El display entregado es de tamaño 16 x 2, lo que significa que solo puede mostrar un mensaje de 16 caracteres cada fila. Para determinar qué file escribir, en el computador se deberá mandar un mensaje de la estructura `Y.MENSAJE AMOSTRAR`, en que la parte `Y.` puede ser 1. para la primera fila y 2. para la segunda fila. Además, de algunos comandos sencillos. Para entender el funcionamiento, se tienen los siguientes ejemplos:

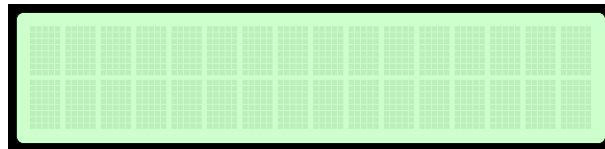
- Se envía el mensaje 1.Hola Mundo ! se mostrará:



- Si luego se envía el mensaje 2.Bienvenido SEP! se mostrará (fijarse que el mensaje anterior se mantiene):



- Si se envía 0.CUALQUIERMENSAJE, se reconocerá la estructura 0. y se limpiará el LCD.



- Si se envía el mensaje 3.No me veré :(, la interfaz no debe mostrar el mensaje, ya que la opción 3. no está asociada a ninguna opción.
- Si se envía el mensaje 2 Tampoco me vere :(la interfaz no debe mostrar el mensaje, ya que la opción 2 no está asociada a ninguna acción (falta el punto .)



Además de lo anterior, se deben enviar mensajes por UART al PC utilizando los botones de placa. Si se presiona el botón izquierdo, se debe enviar el mensaje **BTN IZQ**; si se presiona el botón derecho, se debe enviar el mensaje **BTN DER**.

3.3. Task 3: Comunicación MSP - AVR

En esta etapa, se deben unir lo creado en el Task 1 y Task 2, para crear el sistema de comunicaciones buscado. Deberá ajustar los mensajes y las recepciones para implementar lo siguiente. La separación en sub-Task es solo para sugerirle como abordar el problema, deben ser concurrentes.

3.3.1. Task 3.1 Consulta de valor en modo manual

El MSP enviará un pequeño mensaje al AVR al presionar alguno de los botones. Si se presiona el botón izq, se deberá solicitar al AVR que envíe los datos del fotorresistor. Por otra parte si se presiona el botón derecho, se deberá solicitar al AVR que envíe los datos del potenciómetro. El mensaje que envíe el AVR puede tener la siguiente estructura:

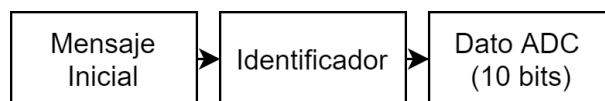


Figura 1: Paquete de ejemplo de mensaje

El identificador, es para que el MSP sepa qué valor se le está enviando. Es importante que se envíe el valor completo (10 bits) del ADC. Con este valor, el MSP calculará el equivalente en Voltaje que será mostrado en conjunto con el valor en 10 bits. A modo de ejemplo, puede ver la siguiente imagen

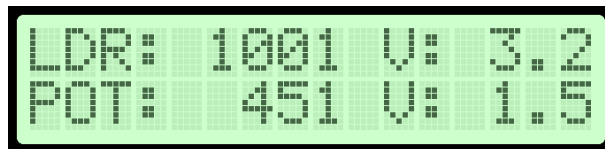


Figura 2: Información en display



3.4. Task 3.2 Cambio a modo manual y modo automático

Para cambiar entre modo automático y modo manual se utilizará el botón del AVR. Con esto el AVR empezará a enviar los datos de forma alternada con un delay entre 500 ms y 2 seg. Es importante considerar este tiempo para que la pantalla del LCD se actualice sin problemas.

3.5. Consideraciones Importantes

- El MSP no tiene forma de saber si el AVR está en modo automático o modo manual. Por lo que debe ser capaz de recibir los datos en cualquier momento y separarlos de forma correcta.
- El AVR no tiene forma de saber si el MSP está conectado o no, por lo que si se está en modo automático enviará los valores sin problemas. Se sugiere utilizar el led de la placa o un externo para saber en qué modo se encuentra.
- **El MSP funciona a 3.3 V, y el AVR funciona a 5 V. Para evitar posibles cortocircuitos, debe implementar un divisor resistivo que cambie la señal de 5 V a 3.3 para poder transmitir del AVR al MSP. No es necesario hacerlo para transmitir del MSP al AVR.** El siguiente diagrama muestra la conexión a realizar.

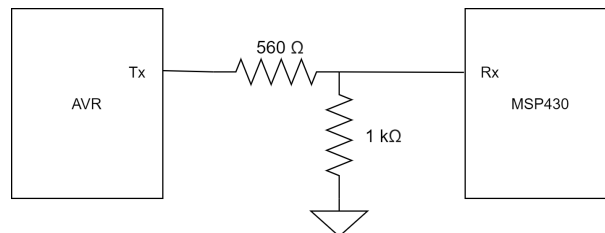


Figura 3: Conexión

- Tanto los botones, como la recepción de datos, debe ser implementada con interrupciones de pin e interrupciones de recepción.



4. Lectura recomendada

- [ATmega328/P Complete Datasheet.](#)
- [MSP430x5xx and MSP430x6xx Family User's Guide](#)
- [MSP430F552x, MSP430F551x Mixed-Signal Microcontrollers datasheet.](#)
- Ayudantías y códigos subidos

5. Rúbrica de Evaluación

5.1. Consideraciones generales

- Debe cumplir con el horario asignado. Tendrá un periodo de 20 min de revisión.
- 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.
- No se reciben trabajos después del módulo de presentación. Trabajos no entregados son calificados con nota mínima.
- Solo dispone del tiempo asignado para ser revisado, atrasarse y no cumplir hará que no sea posible revisarlo de forma correcta.
- Solo se revisará el Task 3. En caso de no lograrlo, puede presentar el Task 1 y Task 2 por separado. En este caso la nota máxima será de 4.0 sin derecho a bonus.
- **IMPORTANTE:** se prohibirá el uso de funciones de Arduino IDE y Energia.nu para la programación de las tarjetas de desarrollo, esto por la simplicidad que involucra, por lo que deberán mostrar que entienden qué están realizando al momento de programar.

5.2. Criterios de aprobación

5.2.1. Task 1: AVR - PC, Task 2: MSP - PC

Sin revisión por parte de los ayudantes.



5.2.2. Task 3: AVR - MSP

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:
 - Programa compilado y ejecutándose
 - Debe detectarse el botón mediante interrupciones de pin.
 - Utilización de ADC para medir el voltaje en el potenciómetro y fotorresistor.
 - Uso de interrupciones de recepción de UART para la comunicación.
 - Uso de la pantalla LCD. Deberá buscar una librería para poder usarla en el MSP.
 - Transmisión y recepción con el protocolo de comunicación 57600 8N1.
 - Código ordenado y estructurado. Se sugiere pensar el laboratorio en máquinas de estados (lo aprendido con los estados en el LAB 04 puede ser útil).
2. Preguntas. Se responde satisfactoriamente preguntas aleatorias al momento de la presentación final, las cuales abarcan los siguientes temas:
 - Preguntas de código.
 - Funcionamiento del sistema del protocolo UART en cada uno de los microcontroladores.

6. Bonus

Al cumplir con todos los puntos indicados en la **Rúbrica de Evaluación** obtendrá la nota máxima equivalente a 6.0. Contará con los siguientes bonus disponibles para lograr la nota máxima (Tope 7.0).

6.1. Bonus 1 (10 décimas). Implementar TMP102 con I2C en el sistema

En este bonus deberá incorporar el sensor de temperatura TMP102 al sistema completo. Para esto, deberá agregar más opciones en la comunicación para poder enviar los valores del TMP102 y mostrar la temperatura en el LCD. Para esto puede basarse en la siguiente imagen.

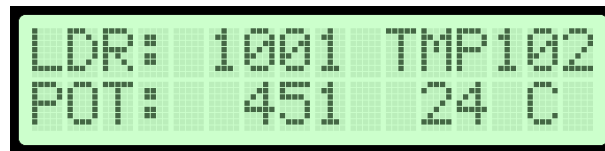


Figura 4: TMP102 incorporado al sistema

6.2. Bonus 2 (5 décimas): Timer para activar modo automático

Si el AVR está en modo manual y han pasado más de 30 segundos sin recibir solicitudes por parte del MSP, se activará automáticamente el modo automático y comenzará a enviar datos.

6.3. Bonus 3 (5 décimas): Sistema robusto

En un sistema de comunicaciones, es bastante normal que ocurran errores inesperados. Por esto mismo, existen mecanismos para llevar un chequeo de errores que permiten informar la situación actual. Para este bonus, usted deberá implementar un checksum al dato enviado.

En terminos simples, este es un mensaje que se añade al paquete transmitido que es la suma de los bytes enviados. En el receptor, se suman los bytes recibidos y se compara con el checksum recibido. Si es que el valor es el mismo, se considera correcto y se muestra el valor en el LCD. En caso contrario, el display mostrará ----.

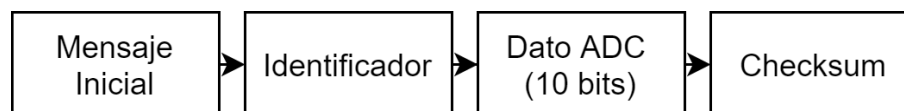


Figura 5: Checksum incorporado al sistema



6.4. Bonus 4 (5 décimas): ACK y Detección de desconexión

El ACK o *acknowledgement* es un paquete que se utiliza para confirmar la recepción de la información por parte del dispositivo que la solicita. Es utilizado en el protocolo **TCP-IP**.

Usted deberá implementar un paquete ACK que envíe el MSP para confirmarle al AVR que recibió el dato correctamente. Cuando el AVR se encuentre en modo manual deberá esperar este paquete, si no es recibido en 1 segundo deberá mandar nuevamente el valor hasta que reciba el ACK. Este proceso solo se puede repetir 3 veces. Si después de 3 intentos no recibe el ACK, el led del AVR deberá comenzar a parpadear para indicar que no hay conexión con el MSP.

Por otra parte, si el MSP envía la solicitud al AVR de un dato y este no es recibido en 3 seg. Deberá comenzar a parpadear un led de la placa para indicar la desconexión del AVR y mostrar en el LCD el mensaje error.

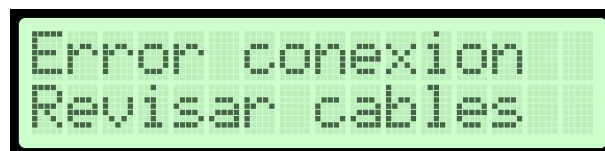


Figura 6: Mensaje de error