



Lab 04, ADC y UART

Laboratorio Nivel 3

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 USART o *Universal Synchronous and Asynchronous Receiver and Transmitter* para realizar la comunicación con un computador.

Cabe indicar, que cuando se realiza la 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 lo que 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, impidiendo poder utilizar ese tiempo en otras actividades.

Este tiempo en que no hay comunicación, puede ser utilizado para la adquisición de información. En laboratorios anteriores se trabajó con inputs mediante GPIOs usando o no interrupciones. Para este laboratorio además, se utilizará el Conversor Analógico Digital (ADC) incorporado en el microcontrolador para adquirir información y a partir de esta, ejecutar acciones.

Esta información, junto con la capacidad de comunicación, son insuficientes si solo se envían datos sin sentido. Por lo que entra en juego la capacidad de contar con un Protocolo de Comunicación que permita identificar un mensaje recibido o crear uno para ser enviado, con el objetivo de tomar las decisiones pertinentes acerca de algún proceso o de información a enviar.

Junto con lo anterior, se incorpora el concepto de interrupciones para la utilización de Timers y para la detección de entradas externas mediante puertos digitales.

El objetivo de este laboratorio, es lograr la implementación de un pequeño sistema, que a partir de instrucciones y datos leídos, sea capaz de controlar un motor servo.



2. Actividad

Esta actividad consistirá en la creación de un controlador y almacenamiento de posiciones de un motor servo. Este sistema tendrá 2 métodos de control, seleccionables mediante un terminal serial. Cada método de control deberá cumplir con las siguientes características:

2.1. Método 1: Controlar la posición del servo mediante Joystick integrado en Booster Pack

Para esta actividad, deberá configurar el joystick del Booster Pack en su eje horizontal para controlar la posición del motor servo entregado. La posición de más a la izquierda llevará el servo a la posición correspondiente a 0° y la posición de más a la derecha, llevará el servo a la posición correspondiente a 180° .

En conjunto a lo anterior, el botón derecho del MSP430 actuará como "SAVE", almacenando la posición en la que se encuentre el servo en un momento determinado. Se deberán guardar 5 posiciones, las que además deberán ser enviadas mediante comunicación serial al computador, para ser visualizadas en un terminal serial. Aquí se desplegará la opción de repetir la secuencia, en que el servo se moverá a automáticamente a las posiciones indicadas, en caso contrario, se podrá volver a un menú principal.

2.1.1. Requisitos

El control debe cumplir con los siguientes requisitos:

- Funcionar para un *baudrate* de al menos 9600, 19200, 57600 ó 115200, seleccionables mediante código. Para esto, deberá completar la estructura entregada del comunicación serial.
- Para ayudar a la selección de los ángulos, deberán ser enviados a medida que se mueve el servo mediante comunicación serial.
- El botón debe ser configurado mediante interrupciones. Puede agregar un LED que parpadee indicando que la posición fue guardada con éxito.
- Los ángulos son guardados con 3 dígitos, rellenando con ceros a la izquierda en caso de ser necesario.
- El formato de visualización en el terminal serial, es el que se muestra en la siguiente figura:



```
1 Menu:
2 1. Control con Joystick.
3 2. Control mediante comunicación serial.
4
5
6 -----
7 1. Control con Joystick
8 -----
9
10 Se guardo la siguiente secuencia:
11
12 034-120-060-076-170
13
14 ¿Repetir secuencia?
15 1. Si
16 2. No
17
```

Figura 1: Ejemplo de menú seleccionando la opción 1.

- En cuanto al método de captura del ADC, se deberá justificar la elección de dicha forma en el README utilizando razones concretas.

2.2. Método 2: Controlar la posición del servo mediante comunicación serial

Para esta opción, deberá ser capaz de entregarle una serie de posiciones mediante un texto enviado por comunicación serial. El microcontrolador debe ser capaz de almacenar y reproducir estos ángulos en el motor servo en el orden entregado. Luego de reproducirlos, se debe entregar la opción de volver a realizarlo o de volver al menú principal.

2.2.1. Requisitos

El control debe cumplir con los siguientes requisitos:

- Funcionar para un *baudrate* de al menos 9600, 19200, 57600 ó 115200, seleccionables mediante código. Para esto, deberá completar la estructura entregada del comunicación serial.
- Los ángulos son entregados con 3 dígitos, rellenando con ceros a la izquierda en caso de ser necesario.
- Puede asumir que el formato de la secuencia será siempre igual, terminando con un salto de línea (`\n`), pero que pueden entregarse ángulos fuera de rango. Por lo que el microcontrolador deberá responder con un mensaje indicando el error.
- El formato de visualización en el terminar serial, es el que se muestra en la siguiente figura (puede modificar los textos si lo considera necesario):



```
20 Menu:
21 1. Control con Joystick.
22 2. Control mediante comunicación serial.
23
24
25 -----
26 2. Control mediante comunicación serial
27 -----
28
29 Ingrese la secuencia a reproducir:
30
31 Se ingresó la siguiente secuencia:
32
33 004-300-060-999-170
34
35 Existen errores, ingrese nuevamente:
36
37 Se ingresó la siguiente secuencia:
38
39 004-170-060-076-170
40
41 Secuencia aceptada, reproduciendo.
42
43 ¿Repetir secuencia?
44 1. Si
45 2. No
```

Figura 2: Ejemplo de menú seleccionando la opción 1.

2.3. Consideraciones generales

- La elección de la forma de utilizar el botón, queda a su criterio. Debe justificar el uso o no uso de interrupciones. Debe considerar el uso de debouncing.
- Tanto la recepción como la transmisión de UART pueden ser implementadas con técnicas de *polling*. El uso de interrupciones será opcional.
- El motor servo puede ser implementado con interrupciones de timers, sin embargo, queda a su criterio la elección.
- **No sobrecargue de instrucciones a las interrupciones.** Las interrupciones deben ser acciones pequeñas que cambien estados del programa, no se espera que todo su programa esté en las interrupciones.
- El modo *debug* de Code Composer Studio es muy útil para monitorear cada línea del código. Se pueden seleccionar breakpoints para monitorear las interrupciones y llegar a esas líneas cuando se ejecuta una acción.
- Revisar y estudiar el Datasheet. Contiene información relevante para la realización del laboratorio.
- Utilizar máquinas de estado. Al igual que el laboratorio anterior, será muy necesario.



3. Lectura recomendada

- Páginas iniciales del Datasheet del MSP430F5529
- Capitulo 28: ADC12_A del [MSP430x5xx and MSP430x6xx Family User's Guide](#).
- Capitulo 36: Universal Serial Communication Interface – UART Mode del [MSP430x5xx and MSP430x6xx Family User's Guide](#).
- *Buzzer* [https://www.cuidevices.com/product/audio/buzzers/audio-transducers/cem-1203\(42\)](https://www.cuidevices.com/product/audio/buzzers/audio-transducers/cem-1203(42))
- [MSP430x5xx and MSP430x6xx Family User's Guide](#).
- [MSP430F552x, MSP430F551x Mixed-Signal Microcontrollers datasheet](#).
- BOOSTXL-EDUMKI User Guide <https://www.ti.com/lit/ug/slau599a/slau599a.pdf>
- [Datasheet Servo SG90](#)



4. Sesión de preguntas

4.1. Consideraciones generales

- Debe cumplir con el horario asignado y solamente tendrá aproximadamente 10 min para realizar la presentación. Se pide encarecidamente respetar esta regla.
- 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.
- Puede utilizar cualquier recurso que tenga disponible. Ya sea un block de notas, mostrar brevemente alguna parte de su código, tener una presentación de apoyo, el Datasheet, etc. Queda a criterio del ayudante corrector la solicitud de mostrar este apoyo.
- **IMPORTANTE:** se prohibirá el uso de funciones de 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.

4.2. Preguntas

Responde satisfactoriamente preguntas aleatorias que abordan los siguientes temas:

- Especificaciones generales del MSP430F5529 en lo que respecta al uso del modulo de ADC.
- Especificaciones generales del MSP430F5529 en lo que respecta a *timers*. Tales como frecuencias admitidas, elección de modos, cantidad de contadores, cantidad de *timers*, etc.
- Funcionamiento del sistema del protocolo UART en el MSP, tanto a nivel general como a nivel específico.
- Comprender **cada línea** de su programa.
- Entendimiento de *Headers Files* o archivos de extensión .h. ¿Cuál es el beneficio que tiene la creación de bibliotecas y cómo ayuda a la estructuración de código ?
- ¿Qué significan las directivas `#IFDEF` `#DEFINE` `#ENDIF` ?



5. Fecha y forma de de entrega

La entrega se debe realizar el día jueves 10 de junio las 23:59 hrs. El día miércoles 9 de junio, en horario de laboratorio se realizará el control de salida.

Deberá subir su **código completo y ordenado** a su repositorio de GitHub junto con un archivo Readme.md que indique las fuentes utilizadas y una lista de lo que no fue implementado en su código. De forma que el ayudante corrector lo tenga claro. Además debe indicar sistema operativo usado y/o si utilizó un compilador online.

El código entregado debe cumplir con lo siguiente, el no cumplimiento de estos puntos podría incurrir en un descuento:

- Programa compilable y sin errores. Recomendamos subir el proyecto completo generado en Code Composer Studio.
- Código ordenado y debidamente comentado. No se exigirá ningún estándar en particular, pero se debe mantener una estructura que permita una lectura fácil. ¹²
- Recursos utilizados se deben citar en el README.md de su repositorio.
- Su código debe tener un encabezado como el mostrado en el ejemplo de “Hello World!” del LAB 01.

¹En <https://developer.gnome.org/programming-guidelines/stable/c-coding-style.html.en> pueden encontrar ejemplos de cómo mantener un buen formato de código.

²En <https://codebeautify.org/c-formatter-beautifier> pueden reformatear un código.