



Lab 2: Output e Input Básicos (GPIOs)

Laboratorio Nivel 2

1. Objetivo

El presente laboratorio tiene por propósito el introducirlos al funcionamiento básico de sus microcontroladores, a través del manejo de entradas y salidas básicas (*General Purpose Inputs and Outputs* o GPIOs). A modo de concretar este objetivo, deberán comprender el cómo operan estos puertos a través de una lectura a conciencia de los respectivos *datasheets* de cada microcontrolador. Una vez entendido su funcionamiento, deberán hacer uso de los conocimientos adquiridos para generar un programa capaz de controlar un display de 7 segmentos y botones de distinto tipo.

Esta experiencia estará separada en 2 partes, **cada una de ellas con un microcontrolador distinto**. En primer lugar, deberán hacer uso de uno de sus microcontroladores para manipular un *display* de 7 segmentos como un contador. Luego, usando el segundo microcontrolador, deberán utilizar el botón incluido en la tarjeta de desarrollo y un botón externo, para controlar la dirección de la cuenta del *display*¹.

2. Descripción de la actividad

Task 1: Contador 7 segmentos

Para la siguiente actividad deberá hacer uso del *display* 7 segmentos que se le ha entregado en su kit de SEP. La actividad consiste en la programación de un contador que repita de forma cíclica, debe contar desde 00 hasta 99 (avanzando de uno en uno) a una velocidad determinada por usted y que permita ver "los dos dígitos encendidos." al mismo tiempo, tal y como se muestra en la Figura 1. Junto con lo anterior, elija resistencias que permitan que la intensidad de cada segmento sea la misma (vea el *datasheet*).

¹Como pueden percatarse, evidentemente se verán obligados a realizar la primera actividad en ambos microcontroladores, de modo que les recomendamos conocer a fondo el cómo operan sus 7 segmentos



Figura 1: Patrón *Task 1*.

Tenga en consideración que existen *displays* 7 segmentos tanto de ánodo común como de cátodo común. Depende completamente de usted identificar de qué electrodo común es su componente, por medio de la lectura del *datasheet* del mismo. Si lleva un buen rato probando su código y ve que el 7 segmentos no enciende, considere que:

- Puede que esté conectando el electrodo incorrecto.
- Puede que esté haciendo un gestión incorrecta del GPIO de su microcontrolador.
- Puede que haya quemado el 7 segmentos (improbable pero no imposible).

Task 2: Contador con botones

En esta sección se busca introducir al alumno a la lectura de entradas digitales y cómo procesar estas para realizar alguna acción. Esto se realizará mediante la configuración de dos botones, el de la placa y uno armado por ustedes mediante dos contactos de aluminio, los cuales serán las entradas, y un *display* de 7 segmentos, que representa la salida del sistema.

Para el caso del botón de aluminio experimentará el efecto de *bounce*, por lo cual deberá diseñar mediante *software* una rutina que se encargue del *debouncing* de dicho botón. En el caso del botón de la placa, dependerá del microcontrolador de su elección, porque existen algunos que tienen un sistema de debouncing por *hardware*, así que deberá averiguar si es que es así o no antes de programar.

Lo que se pide en esta actividad es que primero configure el botón de la placa para que, cuando sea presionado, aumente el valor del contador implementado mediante el 7 segmentos en una unidad. En el caso de encontrarse en el máximo valor (9), debe pasar de este al número 0 para continuar el ciclo.

Como contraparte de este botón, se deberá configurar el botón de aluminio para que, cuando tenga una lectura de que fue presionado, disminuya el valor del contador. En caso de estar en el valor 0 y presionar el botón, deberá pasar a 9 para continuar el ciclo.

Su programa debe contemplar que mientras alguno de los dos botones se encuentre presionado, presionar el otro no tenga efecto.

IMPORTANTE. El cambio del número solo debe ocurrir en el instante en que el botón es presionado, NO cuando es liberado ni tampoco cuando se mantiene presionado.

Para implementar el botón de aluminio, simplemente conecte dos cables caimán a un par de trozos de aluminio, tal como se muestra en la Figura 2. Luego, conecte uno de los caimanes a tierra y otro a algún pin de la placa.

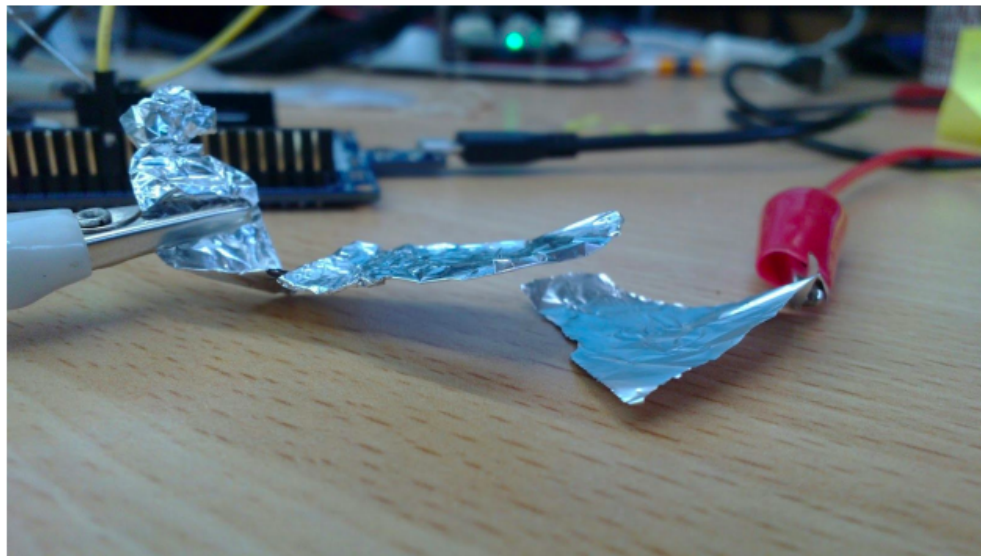


Figura 2: Ejemplo botón aluminio

Por como esta armado este "botón", se tiene que el efecto del *bouncing* será significativo, por lo que debe tener esto en consideración en caso de utilizar delays en su programa, para definirlos adecuadamente. En este *task* tendrá dos opciones de microcontrolador, ATmega328P o MSP430F5529, sin embargo, **la que utilice acá deberá ser distinta de la que usó en el *task* anterior.**

Es importante notar que al estar configurando pines como entradas, es necesario utilizar resistencias de *pull-up*. Estas vienen en general dentro de la tarjeta o se pueden utilizar resistencias externas, pero para este laboratorio es un requerimiento que usen las internas.



3. Lectura recomendada

- [ATmega328/P Complete Datasheet](#).
- [MSP430x5xx and MSP430x6xx Family User's Guide](#).
- [MSP430F552x, MSP430F551x Mixed-Signal Microcontrollers datasheet](#).
- Datasheet del *display* 7 segmentos [HDSP-431G/433G](#).

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 10:30 hrs.
- La nota **Suficiente** 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 **Suficiente**, una mitad del puntaje queda asignada a Aprobación, el restante a Reprobación.

A modo de ejemplo, este Laboratorio es nivel 2, si un alumno tiene **Suficiente**, 10 pts se acumularán a Aprobación y 10 pts será para Reprobación.

- Cualquier consulta sobre el enunciado o 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.



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:

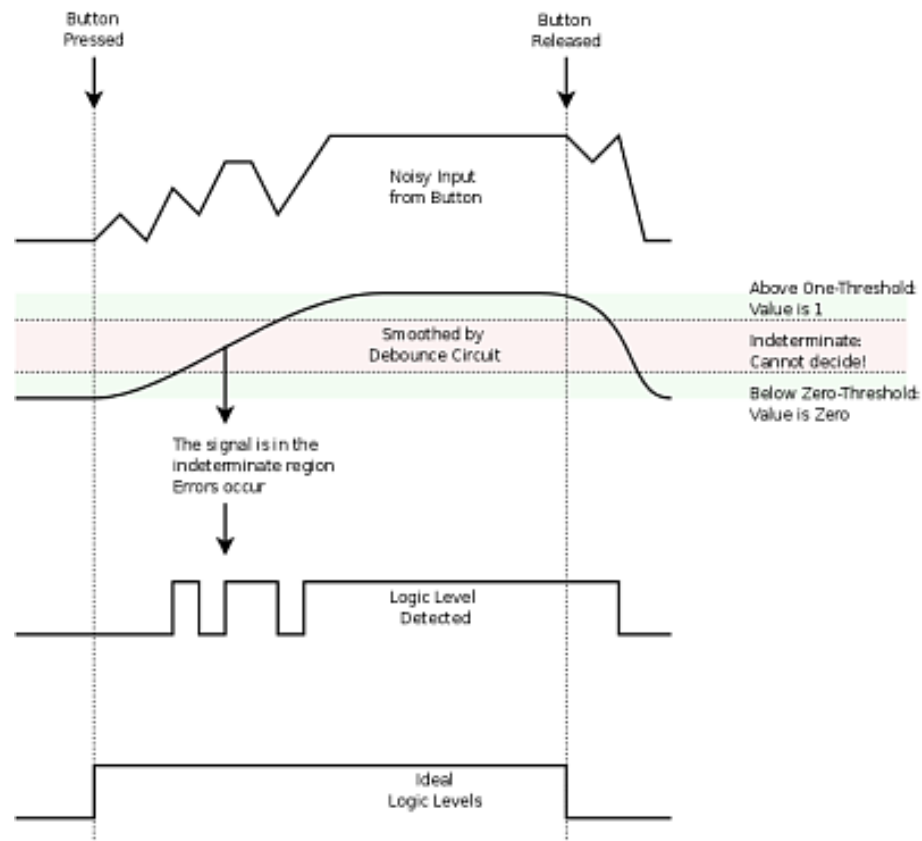
Requerimientos Task 1:

- Programa compilable y cargable.
- Se ejecuta la cuenta en el 7 segmentos de forma concurrente y cíclica, de 0 hasta 9.
- El programa y cableado están ordenados.
- Todo código debe estar subido a GitHub.

Requerimientos Task 2:

- **Utiliza un microcontrolador distinto al Task 1**
- Programa compilable y cargable.
- Se implementa de forma adecuada un método de *debounce*. Este recibe una señal de un contacto sujeto a perturbaciones eléctricas y/o mecánicas y genera una salida **limpia**. El *debouncer* aceptará un primer cambio en la señal, pero rechazará todos los cambios subsecuentes hasta que no haya transcurrido una determinada cantidad de tiempo, establecida por el programador.

Tome como referencia la siguiente imagen, donde el primer gráfico es el comportamiento de una señal con perturbaciones y la última es el comportamiento deseado para esta actividad:



Fuente: HBFS.

- Se implementa exitosamente el *debouncer* en el botón del microcontrolador usado.
- Se implementa exitosamente el *debouncer* en el contacto de dos papeles de aluminio, tal como se muestra en la imagen del enunciado.
- Se produce de forma efectiva un aumento de 1 en el contador al presionar el botón y un decremento de 1 al juntar los aluminios. Cabe destacar que el cambio en el contador debe producirse al momento de presionar el botón o juntar los aluminios, **no después de un “pequeño delay”**.
- Si se mantienen presionados los botones o juntos los aluminios, no se deben producir cambios.
- Debe usarse un botón del microcontrolador y otro hecho con los aluminios, **la ausencia de cualquiera de estos será motivo de no cumplimiento de esta task**.



- Todo código debe estar subido a GitHub.

Preguntas:

Se responde satisfactoriamente a 2 de 3 preguntas aleatorias al momento de la presentación final, las cuales abarcan los siguientes temas:

- Comprender *cada línea* de su programa.
- Características eléctricas de los puertos GPIO de sus microcontroladores: voltaje máximo y corriente máxima que soportan los pines, presencia o ausencia de protecciones.
- Características de su *display* 7 segmentos: cátodo o ánodo común, funcionamiento general de los pines.
- Fenómeno de *bounce*: ¿Qué es? ¿Cómo solucionarlo? Ventajas y desventajas de su algoritmo de *debounce* en comparación a otros métodos.
- Resistencias de *pull-up*: propósito de uso, ¿cómo habilitarlas y deshabilitarlas?

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 análisis, sobre este punto, es importante notar que el mostrar un distinguido implica que el alumno comprende cada línea de su código y los conceptos que utilizó para ello. Estos son analizados en función de lo que presenten sus compañeros. Los distinguidos pueden caer (no exclusivamente) en algunas de las siguientes líneas generales:

- Funcionalidades adicionales sobresalientes, en la línea de uso de GPIOs.
- Métodos de *debounce* mediante *timers* (y su comprensión demostrada por el alumno)
- Uso de ADC o PWM.
- Funcionalidades creativas :D



- **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 código²**.

Para optar a distinguido en este laboratorio solamente deben hacer una idea en un microcontrolador a elección.

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.

²Hint