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 AVR para manipular un *display* de 7 segmentos como un cronómetro, que tenga opción de inicio y stop con un botón. 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: Cronómetro 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 cronómetro que al **presionar el botón** de la placa comience a contar debe contar desde 00 hasta 99 (avanzando de uno en uno) a una velocidad determinada por usted y que permita ver "los dos digitos encedidos.[a]l mismo tiempo, tal y como se muestra en la Figura 1. **Para detener el contador se debe volver a presionar el botón**. Para iniciar el contador nuevamente puede agregar un estado intermedio que retome el valor 00 y espere a volver a presionar el botón o al presionar inicia la cuenta inmediatamente. Junto con lo anterior, elija resistencias que permitan que la intensidad de cada segmento sea la misma (vea el datasheet).

[1] Como 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* [2] 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).
- **No olvide colocar resistencias en serie para limitar la corriente.**

Dado que aún no tienen en su poder el kit de SEP, se deberá emular este circuito con un TinkerCad, tal como se muestra en la figura 2. Es importante notar que deben seleccionar **Anodo Cómun** en cada uno de los display de 7 segmentos para que sea exactamente igual a cómo es en el kit de SEP. Además, dado que el microcontrolador no tiene un botón de placa, deberá uilizar un pulsador externo con su respectiva resistencia de pull-up o pull down según estime conveniente.

[2] Datasheet del *display* 7 segmentos HDSP-431G/433G.

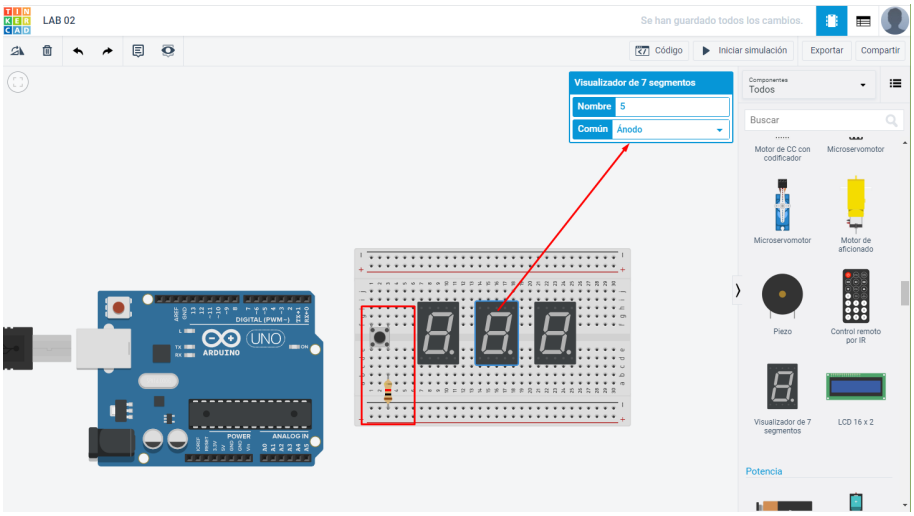


Figura 2: Caption

Task 2: Contador con botones [Pendiente hasta nuevo aviso]

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 *hadware*, 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 3. Luego, conecte uno de los caimanes a tierra y otro a algún pin de la placa.

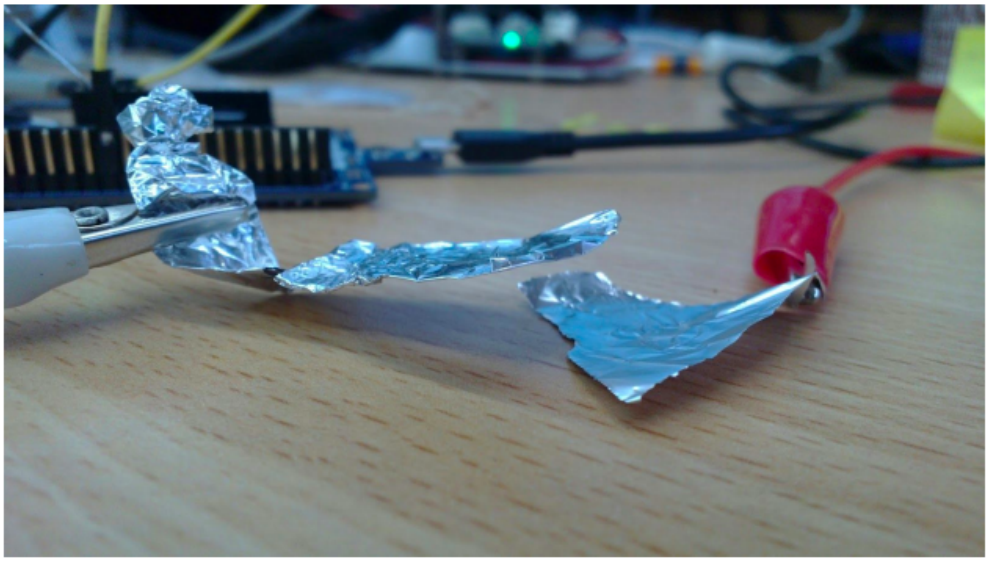


Figura 3: Ejemplo botón aluminio

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* se utilizará el MSP430F5529.

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

- Capitulo 18: I/O-Ports del ATmega328/P Complete Datasheet.
- Capitulo 12: Digital I/O Module del MSP430x5xx and MSP430x6xx Family User's Guide.
- Secciones 5 y 6 del MSP430F552x, MSP430F551x Mixed-Signal Microcontrollers datasheet.
- 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.
- Conexión botón pulsador Conexión

4. Rúbrica de Evaluación

4.1. Consideraciones generales

- Debe cumplir con el horario asignado y solamente tendrá aproximadamente 6 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.
- No se reciben trabajos después del módulo de presentación. Trabajos no entregados son calificados con nota mínima.
- 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.

4.2. Criterios de aprobación

4.2.1. Task 1: ATmega328P

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 compilable y cargable.
   - Se ejecuta la cuenta en el 7 segmentos de forma concurrente y cíclica, de 00 hasta 99.
   - El botón inicia y detiene el contador.
   - El cambio entre los digitos no es visible o es levemente visible.
   - El programa y cableado están ordenados.
   - Todo código debe estar subido a GitHub.
2. Preguntas. Se responde satisfactoriamente preguntas aleatorias al momento de la presentación final, las cuales abarcan los siguientes temas:

- Especificaciones generales del ATmega328P en lo que respecta a inputs y outpus. Tales como existencia de resistencias internas, voltajes y corrientes máximas admitivas.
- Conceptos iniciales del tiempo del procesador y el significado de delay().
- Configuración de pull-ups y pull-down si aplica.
- Comprender **cada línea** de su programa.

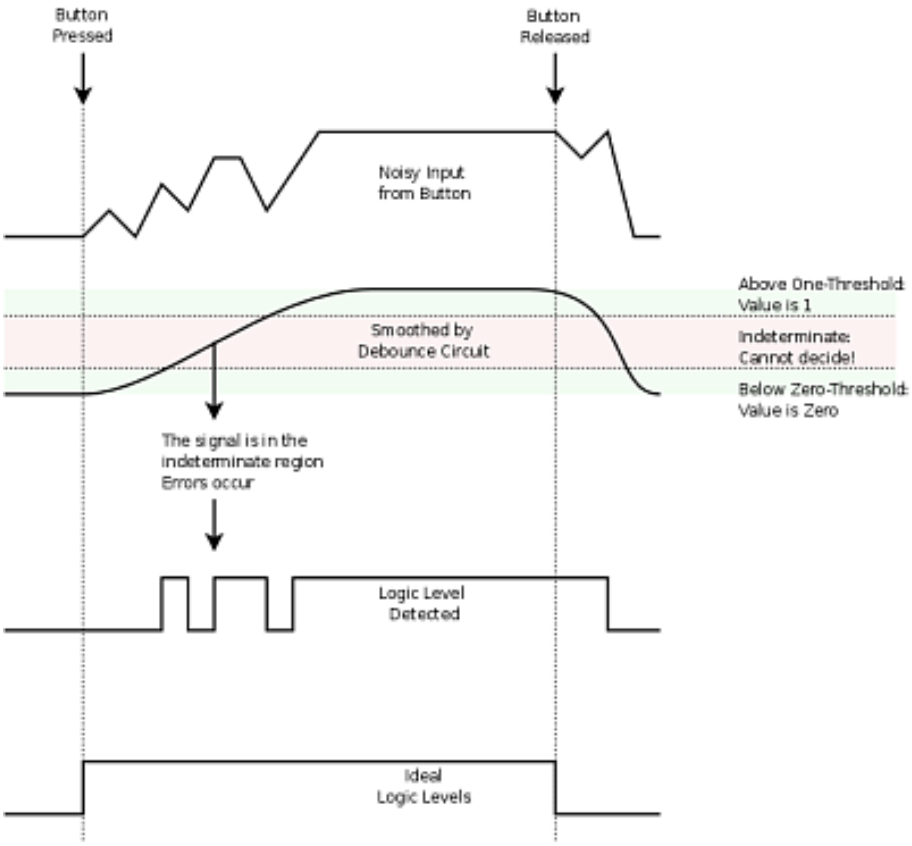
Solo se dispone de una oportunidad para responder estas preguntas y únicamente será posible si es que cumple con el mínimo informado en la rúbrica. Fallar en este requisito se traduce en en que no se considerarán los puntos obtenidos en la parte **Funcionamiento de requerimientos**.

4.3. Task 2: MSP430F5529

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:

   - **Utilización de MSP430F5529**
   - 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.

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

   - Especificaciones generales del MSP430F5529 en lo que respecta a inputs y outpus. Tales como existencia de resistencias internas, voltajes y corrientes máximas admitivas.
   - Conceptos iniciales del tiempo del procesador y el significado o generación de delays().
   - Configuración de pull-ups y pull-down si aplica.
   - Comprender **cada línea** de su programa.
   - Entendimiento dl concepto de debouncing por software y su diferencia por hardware.

   Solo se dispone de una oportunidad para responder estas preguntas y únicamente será posible si es que cumple con el mínimo informado en la rúbrica. Fallar en este requisito se traduce en en que no se considerarán los puntos obtenidos en la parte **Funcionamiento de requerimientos**.

4.4. Bonus

Al cumplir con todos los puntos indicados en la **Rúbrica de Evaluación** obtendrá la nota máxima equivalente a 6.0. Para poder aumentar esta nota existirán una serie de bonus que estarán indicados en la rúbrica. La asignación de estos puntos estarán sujetos a cumplir un mínimo en la presentación y condicionados a preguntas que se le harán al momento de presentarlo, ya que debe mostrar un dominio en el tema indicado.