

UBA - Facultad de Ingeniería

Departamento de Electrónica

Laboratorio (66.02)

Proyecto integrador SIMON con Arduino

Año 2019 - 2do cuatrimestre

Curso: 01

Docentes de la teórica: ROSA, Adrián

Docentes de la práctica: MARINO BELCAGUY, Pablo
SOLER, Francisco
CERVETTO, Marcos

GRUPO 2	
Alumno	Padrón
LÓPEZ SAUBIDET, Joaquín	99252
MARTINEZ SASTRE, Gonzalo Gabriel	102321
RIEDEL, Nicolás Agustín	102130

Fecha de aprobación:
Calificación:
Firma de aprobación:

Observaciones:

Índice

1. Resumen	2
2. Introducción	3
2.1. Consideraciones y propuestas analizadas	3
2.2. Versión final del proyecto	3
3. Especificaciones del sistema	4
3.1. Diagrama Esquemático	4
3.2. Diagrama de cableado de la placa	5
3.3. Componentes necesarios	5
4. Proyección y desarrollo del trabajo	6
4.1. Diagrama de Gantt	6
4.2. Inversión realizada	6
4.3. Desarrollo del proyecto	7
5. Mediciones	8
5.1. Instrumentos utilizados	8
5.2. Tiempo de crecimiento (t_c) de la señal que enciende un LED	8
5.3. Tiempo de decrecimiento (t_d) al pulsar un botón	10
5.4. Efecto rebote de un botón	11
5.5. Caída de tensión en el buzzer en función de la frecuencia	12
6. Resultados	13
6.1. Tiempo de crecimiento (t_c) de la señal que enciende un LED	13
6.2. Tiempo de decrecimiento (t_d) al pulsar un botón	13
6.3. Efecto rebote de un botón	13
6.4. Caída de tensión en el buzzer en función de la frecuencia	14
7. Conclusiones	15
8. Anexo	16
8.1. Cálculo del valor de las resistencias requeridas	16
8.2. Tiempo de crecimiento (t_c) e incertidumbre	16
8.3. Tiempo de decrecimiento (t_d) e incertidumbre	16
8.4. Código Arduino	17

1. Resumen

Nuestro Trabajo Práctico se basa en un circuito electrónico controlado por una computadora central programable llamada Arduino, que emula el juego SIMON.

Teníamos como objetivo utilizar los conocimientos adquiridos en la materia y puestos en práctica con las fichas de laboratorio para llevar a cabo el proyecto.

Nuestra intención era lograr mediante la realización del trabajo familiarizarnos con el funcionamiento y desarrollo de un sistema que use un microcontrolador programable por su amplia versatilidad y aplicabilidad.

2. Introducción

El objetivo de este informe es exponer el proyecto del juego SIMON realizado por el grupo 2 de la materia Laboratorio(66.02). Se detallarán los procesos llevados a cabo para su diseño y construcción, además de las mediciones realizadas que se encuentran relacionadas con los contenidos teóricos relevantes a este curso.



Figura 1: El juego SIMON.

2.1. Consideraciones y propuestas analizadas

Inicialmente se tenía planeado, además de hacer el SIMON, construir también una fuente de alimentación. Finalmente se decidió con ayuda del JTP y ayudantes del Labi que era una complicación innecesaria y simplemente se siguió adelante con el proyecto del juego sin la fuente. Para alimentarlo, se utiliza el puerto usb de una computadora, o en su defecto, una batería de 9V. Por ultimo, se consideró, utilizando el mismo hardware, agregarle otros juegos con un selector al inicio pero al observar que no agregaría nada de interés relevante a los temas estudiados se optó por no hacerlo.

2.2. Versión final del proyecto

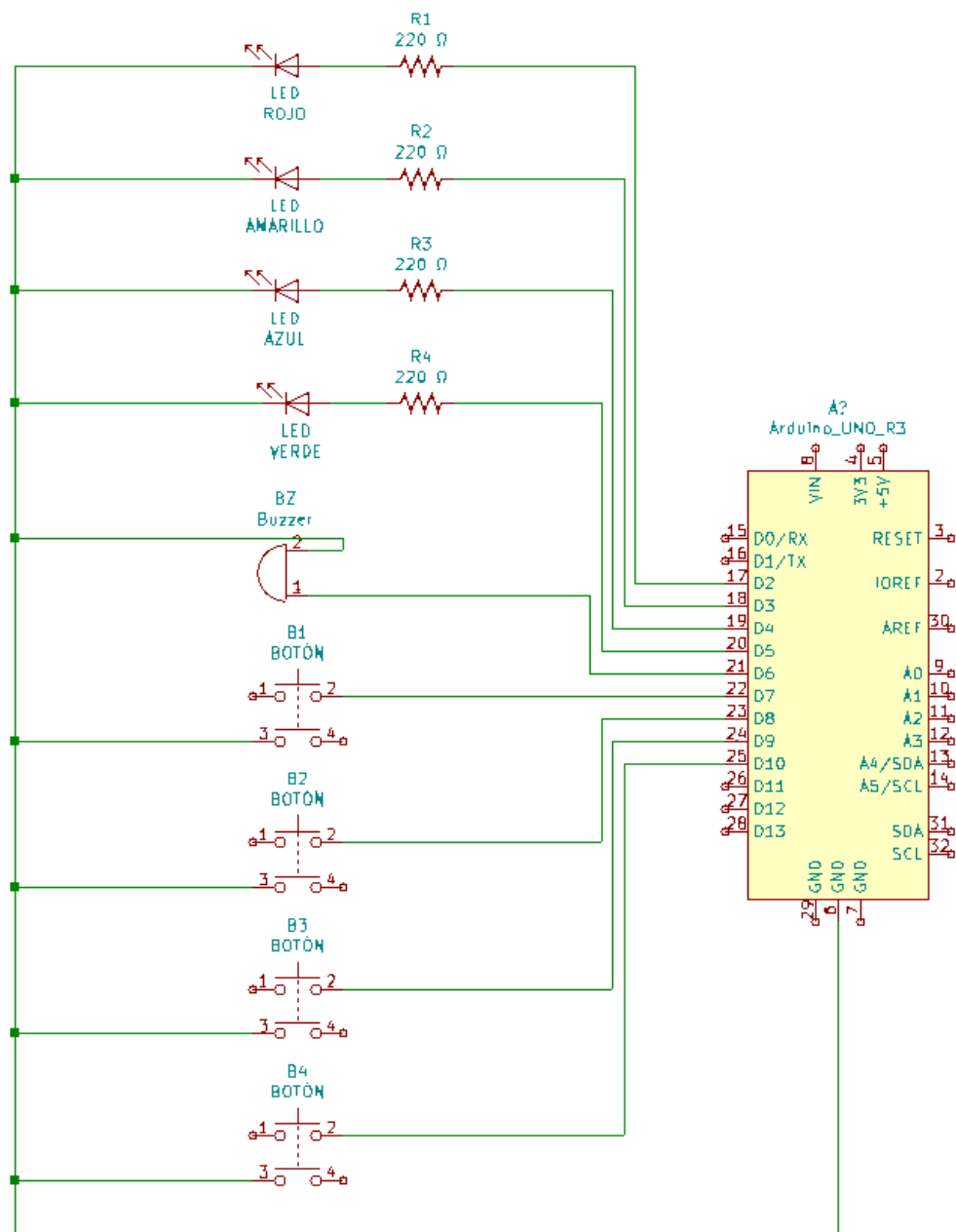
Finalmente, una vez determinado cómo realizar el proyecto, se diagramó aproximadamente cómo estarían ubicados los componentes y las líneas de soldadura. Se diseñaron programas de prueba con el software de Arduino para verificar el correcto funcionamiento de los componentes, y cuando todo estaba funcionando correctamente, se procedió a soldar todo a la placa.

3. Especificaciones del sistema

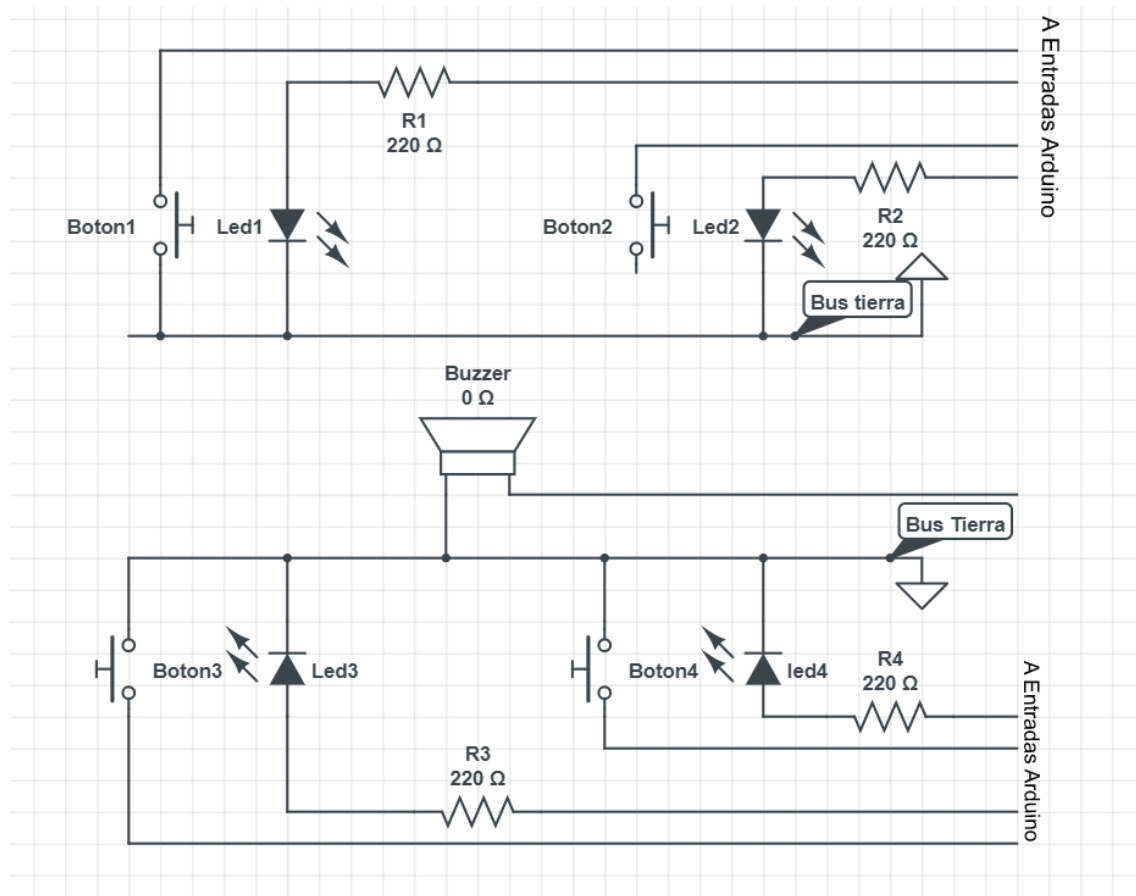
A continuación se describirán los componentes necesarios para llevar a cabo el proyecto así como su distribución dentro del circuito.

3.1. Diagrama Esquemático

El circuito a desarrollar se puede sintetizar en el siguiente esquema.



3.2. Diagrama de cableado de la placa



3.3. Componentes necesarios

ID	Componente	Cantidad
1	Resistencia 220 Ω ¹	5
2	Botón/pulsador	4
3	Placa Universal	1
4	Buzzer Piezoeléctrico 5V	1
5	LED Rojo 5mm	1
6	LED Azul 5mm	1
7	LED Verde 5mm	1
8	LED Amarillo 5mm	1
9	Conector pin hembra	10
10	Conector pin macho	10
11	Batería 9V	1
12	Adaptador Batería-Arduino	1
13	Arduino UNO	1
14	Conectores macho/macho para Arduino	10

¹El cálculo que indica cómo se llegó a este valor fue incluido en el anexo.

4. Proyección y desarrollo del trabajo

Con el fin de unir los conceptos de la materia con el hecho de llevar a cabo un proyecto que resulte interesante al grupo, surgió la idea de montar un objeto con fines lúdicos. Entre las opciones consideradas, la más destacable fue la del SIMON. Tras recibir la confirmación de la viabilidad de la misma, se decidió armar y respetar el siguiente cronograma.

4.1. Diagrama de Gantt

Cronograma de trabajo																
Semana	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Armado de grupo																
Realización de Fichas																
Propuesta de Ideas																
Presentación de Propuesta																
Decisión final																
Compra de materiales																
construcción del proyecto																
mediciones y presentación																
Desarrollo del informe																
Presentación del informe																

4.2. Inversión realizada

ID	Componente	Cantidad	Precio
1	Resistencia 220 Ω	5	\$ 5
2	Botón/pulsador	4	\$ 60
3	Placa Universal	1	\$ 100
4	Buzzer Piezoeléctrico 5V	1	\$ 60
5	LED Rojo 5mm	1	\$ 3.5
6	LED Azul 5mm	1	\$ 3.5
7	LED Verde 5mm	1	\$ 3.5
8	LED Amarillo 5mm	1	\$ 3.5
9	Conector pin hembra	1 tira	\$ 50
10	Conector pin macho	1 tira	\$ 30
11	Batería 9V	1	\$ 160
12	Adaptador Batería-Arduino	1	\$ 35
13	Arduino UNO	1	-
14	Conectores macho/macho para Arduino	10	-
TOTAL			\$514

Tanto el Arduino como sus conectores se encontraban en posesión de los integrantes del grupo desde un principio.

4.3. Desarrollo del proyecto

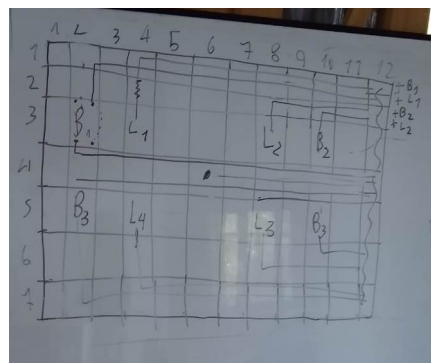
La elaboración del SIMON significaba para los integrantes del grupo la primer experiencia en armado de un circuito electrónico e, incluso, en trabajo con un Arduino. El proyecto se desarrolló como se describe a continuación.

En primer lugar, se conectó una resistencia a una luz LED y se desarrolló un programa para el microcontrolador que se encargara de prender y apagar la luz intermitentemente. Una vez realizado con éxito, se añadió a este circuito un botón y se modificó el software para que la luz se encendiera cada vez que se presionara este.

Una vez familiarizados con los elementos con los que se iba a trabajar, se diagramó en una pizarra el circuito a soldar en la placa y se procedió con el soldado. Para llevar a cabo esto, se hizo uso de los soldadores punta lápiz y estaño que poseían dos de los integrantes del grupo.

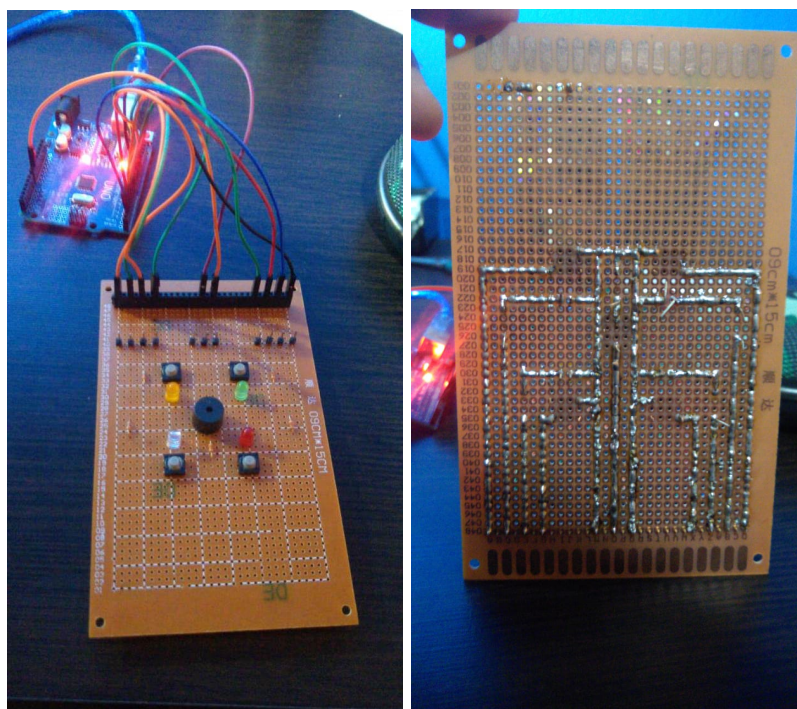
Durante el diseño, se presentó el problema de que faltaban las resistencias para los pulsadores. Sin embargo, tras investigar en distintas fuentes llegamos a que esto no era necesario debido a que se podía usar el pull up interno provisto por el arduino.

Con la placa ya soldada y todos los componentes funcionando se procedió al desarrollo del código, el mismo se encuentra en un repositorio de GitHub. Como relevante se puede mencionar que se usó la biblioteca Easy Button la cual trae, entre otras cosas, una solución al problema del doble pulsado.



<https://es.overleaf.com/project/5dab32cd11bfc90001721d0>

Figura 2: Diagrama en pizarra



(a) SIMON visto desde arriba

(b) Soldaduras realizadas

Figura 3: Proyecto terminado

5. Mediciones

5.1. Instrumentos utilizados

Componente	Descripción
Multímetro Digital True RMS Pro'sKit MT-1707	6000 cuentas. DCV: 600mV/6V/60V/600V/1000V \pm (0.5 % +3d). ACV TrueRMS: 6V/60V \pm (0.8 % +3d) Respuesta en frecuencia: 40Hz - 1kHz
Osciloscopio Digital Tektronix TDS 1002	Ancho de banda: 60MHz. Sensibilidad vertical: 2mV a 5V/div Rango de la base de tiempos: 50ns a 50seg/div.
Puntas de Osciloscopio	R=10M Ω y C=20pF para x10

5.2. Tiempo de crecimiento (t_c) de la señal que enciende un LED

El tiempo de crecimiento se define como el lapso temporal que demora una señal en cruzar un umbral de voltaje inferior específico seguido de un umbral de voltaje superior específico. En este caso, el criterio utilizado para medir fue el intervalo de tiempo entre el 10 % y el 90 % de la amplitud.

Con el objeto de realizar esta medición se decidió hacer uso de un osciloscopio digital puesto que el pulso del cual se busca medir dicho atributo es instantáneo y, por ende, muy difícil de visualizar en un instrumento de tipo analógico.

En primer lugar, se configuró la utilidad Run/Stop para que realice una captura (disparo) cuando detecte un pulso de pendiente positiva que supere cierto valor de tensión. De esta manera, cuando se encendió el led se obtuvo la captura que se muestra a continuación.

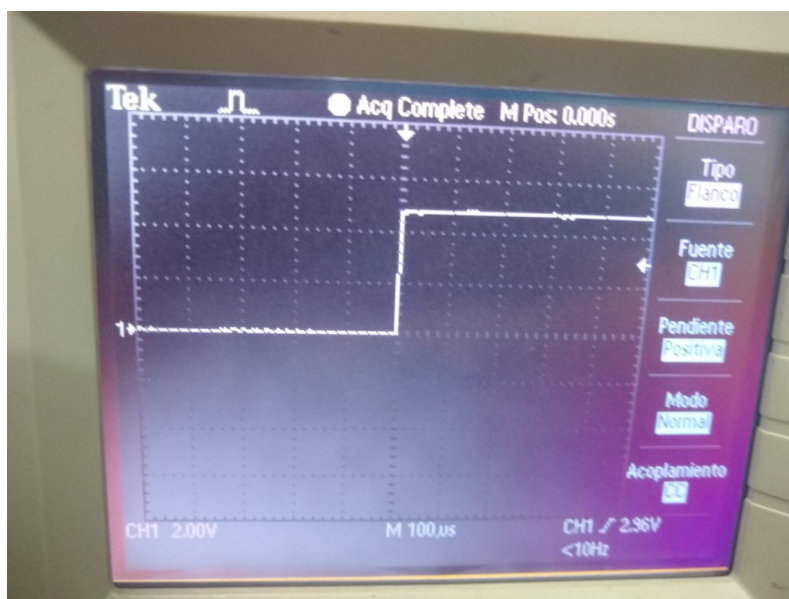


Figura 4: Señal medida en el LED.

Luego, se aumentó la escala de V/div de la captura y se la reposicionó de tal manera que resulte sencillo marcar el 10 % y el 90 % de la señal para, posteriormente, poder llevar a cabo los cálculos correspondientes.



Figura 5: Señal medida modificada.



Figura 6: Método utilizado para medir tiempo de crecimiento (t_c).

5.3. Tiempo de decrecimiento (t_d) al pulsar un botón

Así como se buscó el tiempo de crecimiento de un led, se decidió medir el tiempo de decrecimiento de la señal que recibe el Arduino cuando se presiona un botón. El procedimiento de medición es análogo al anterior.

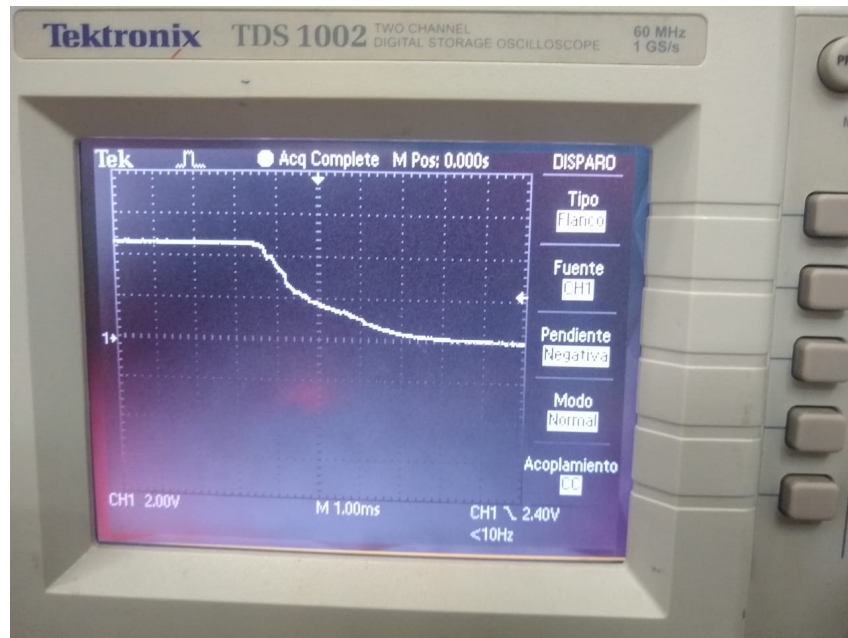


Figura 7: Señal medida.

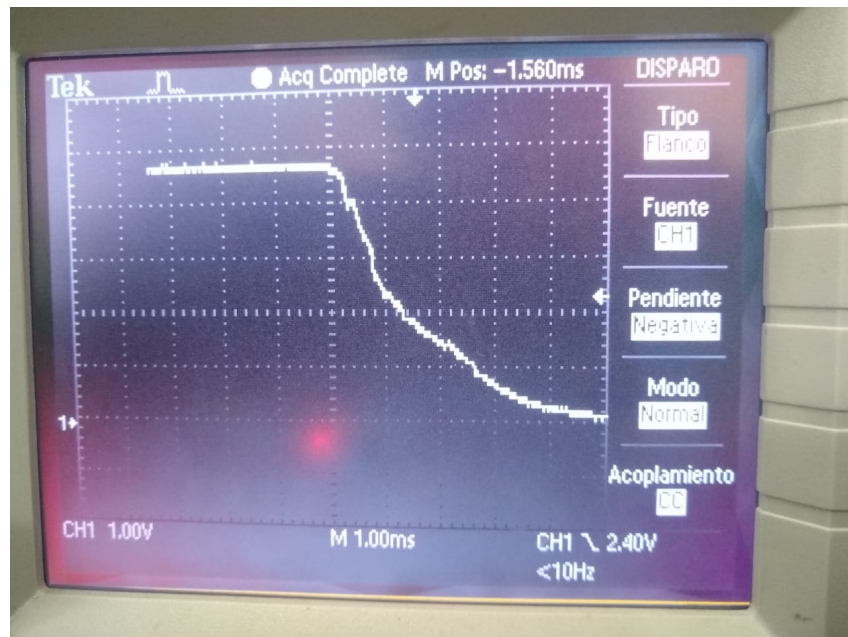


Figura 8: Señal medida modificada.

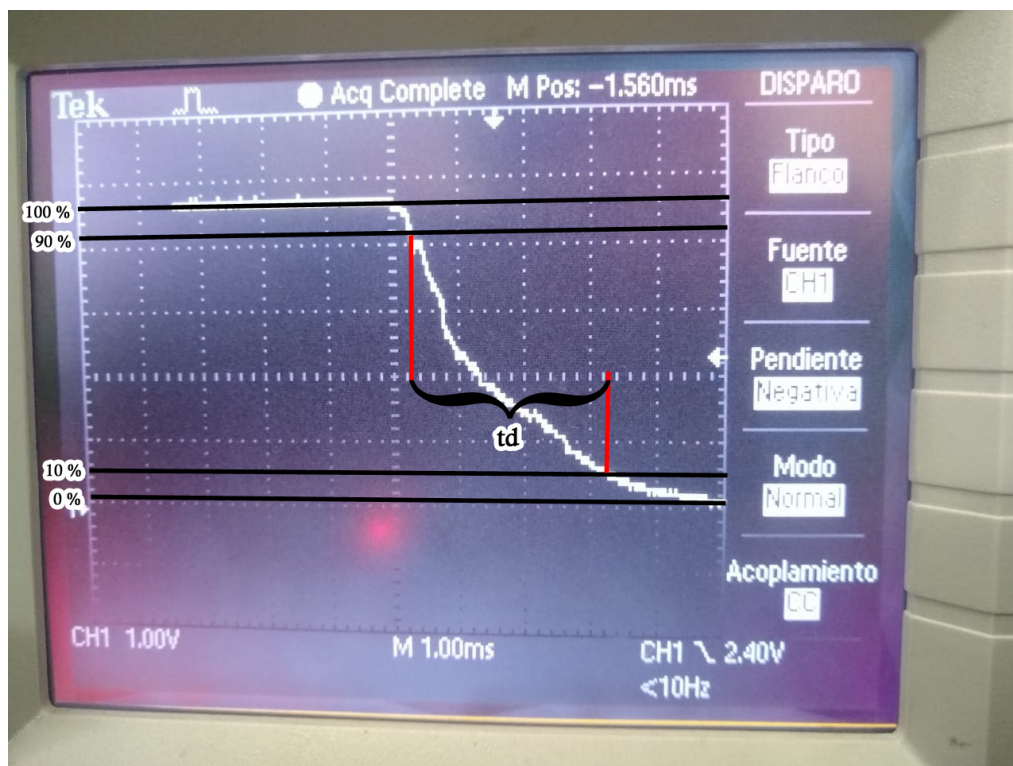


Figura 9: Método utilizado para medir tiempo de decrecimiento (t_d).

5.4. Efecto rebote de un botón

Una cualidad que aparece repetidas veces cuando se refiere al uso de botones/pulsadores es lo que se conoce como efecto rebote. Este consiste en que cuando se pulsa un botón o interruptor, aunque no se pueda ver a simple vista, lo que sucede es que este está rebotando entre en contacto y no en contacto. *"Cuando se cierra el interruptor, los dos contactos se separan y se vuelven a conectar, generalmente de 10 a 100 veces durante un período de aproximadamente 1 ms."* ("The Art of electronics", Horowitz Hill). El resultado de esto se puede observar en la siguiente imagen.

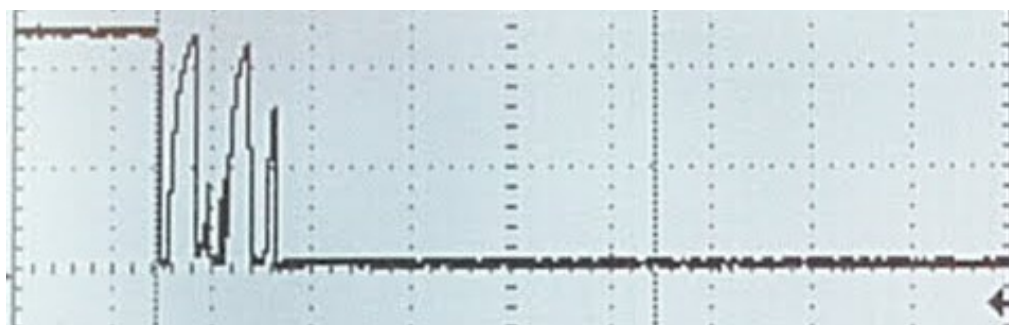


Figura 10: Ejemplo de efecto rebote.

Sin embargo, como se puede observar en la medición anterior, en nuestro caso los botones no cuentan con este efecto, sino que la caída es más suave. La justificación de este fenómeno se llevará a cabo en la sección de resultados.

5.5. Caída de tensión en el buzzer en función de la frecuencia

En un momento, se logró observar que al aumentar la frecuencia que se envía al buzzer, la caída de tensión en el mismo disminuía. Es por esto que se modificó el código fuente del Arduino para que se reproduzca de manera constante una frecuencia determinada y se repitió el siguiente proceso: se medía la tensión en el piezoeléctrico con el voltímetro y luego se aumentaba la frecuencia.

6. Resultados

6.1. Tiempo de crecimiento (t_c) de la señal que enciende un LED

$$t_c = (3,0 \pm 0,5)\mu s \quad (1)$$

En este caso, el error relativo de la medición resultó ser tan alto ($e_r = 16,7\%$) puesto que la cantidad de divisiones medidas era lo suficientemente pequeña como para verse afectada por la incertidumbre de apreciación ($\frac{1}{2}$ división).

6.2. Tiempo de decrecimiento (t_d) al pulsar un botón

$$t_d = (3,0 \pm 0,1)ms \quad (2)$$

En comparación con la medición anterior el error relativo de esta es mucho menor ($e_r = 3,3\%$) puesto que aquí la cantidad de divisiones medidas sí era más grande que la asociada a la incertidumbre de apreciación.

6.3. Efecto rebote de un botón

Como se puede observar en la imagen a continuación, es claro que no hubo efecto rebote.

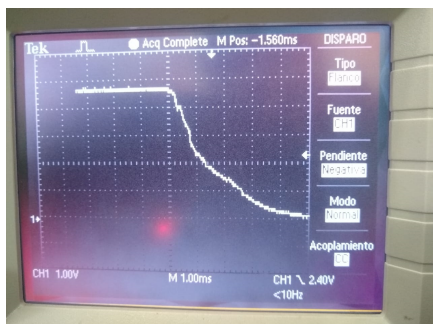


Figura 11: Señal de botón siendo pulsado.

Hay varias maneras de eliminar dicho efecto. En este caso al ver la medición de la señal emitida por el botón podemos intuir que se trata de un debouncer por un circuito RC. Aquí, el circuito funciona como un pasabajos filtrando las altas frecuencias producidas por un efecto rebote.

Como se puede observar en la imagen (Figura 12), al producirse las altas frecuencias gracias al rebote del interruptor, el capacitor se comportará como un cable y la corriente irá a tierra.

En el gráfico de la señal del botón se puede observar la descarga del capacitor hasta llegar a 0.

Ahora bien, ¿dónde se encuentra este circuito dentro del proyecto? La clave de que no se dé este efecto reside en que los botones fueron conectados en los pines del Arduino que poseen la configuración INPUT_PULLUP. Estos pines tienen la cualidad de estar conectados internamente a un circuito de este estilo.

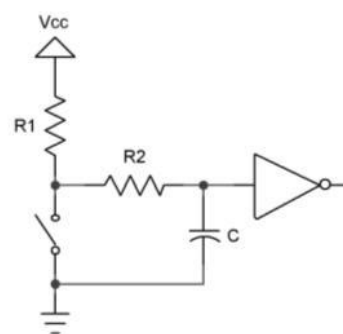


Figura 12: Circuito RC pasa bajos.

6.4. Caída de tensión en el buzzer en función de la frecuencia

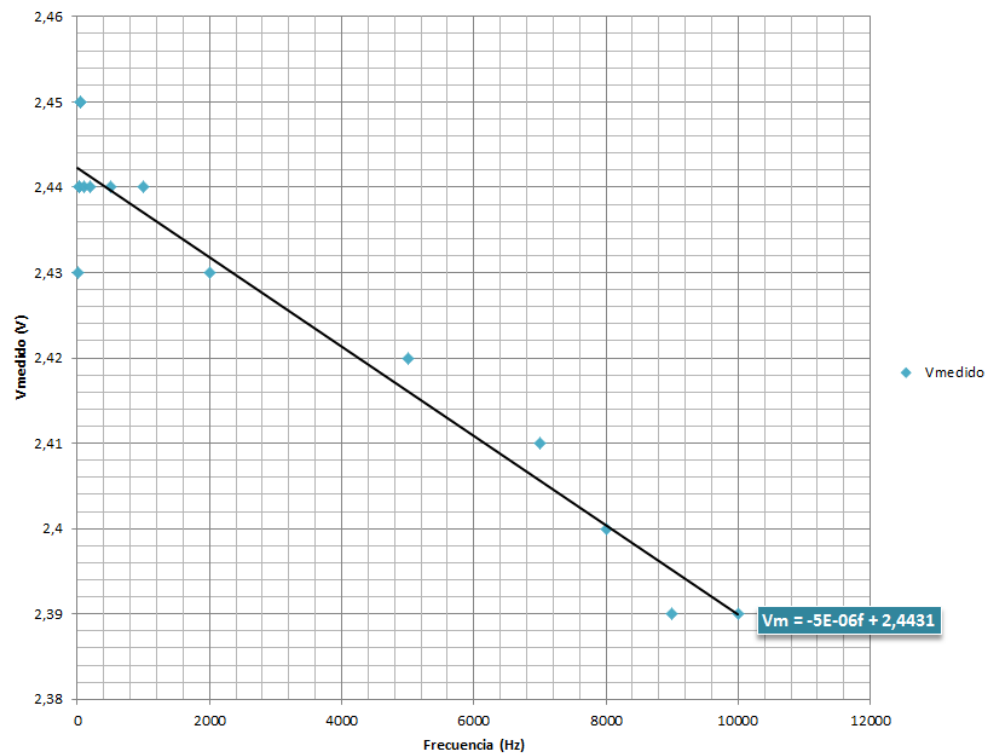


Figura 13: Tensión medida en función de frecuencia.

Teniendo en cuenta que el buzzer es un circuito RL si tomamos el voltaje como $V = (I * Z)$ y el módulo de la impedancia como $|Z| = |R + jX|$, siendo la reactancia inductiva $X = \omega L$, con $\omega = 2\pi f$. Vemos que para valores pequeños de R se espera una relación lineal entre la frecuencia y la tensión.

7. Conclusiones

El desarrollo del trabajo pudo ser llevado a cabo sin mayores complicaciones. Se logró, gracias a pruebas unitarias de hardware previas al ensamblaje definitivo del trabajo, verificar el correcto funcionamiento de los componentes y que su desempeño sea el esperado para nuestros requerimientos.

Si bien, en un comienzo, los integrantes del grupo no poseían experiencia alguna en la realización de proyectos de esta índole, los conocimientos adquiridos a la largo de la materia hicieron que resultara ameno poder llevar adelante el trabajo.

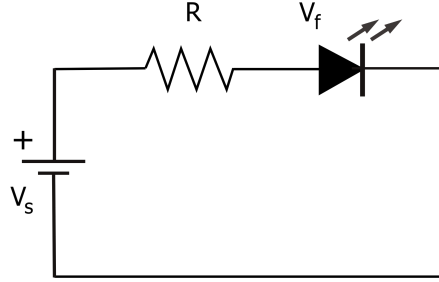
Como resultado de la realización de esta tarea, se adquirió el entendimiento necesario para poder volver confeccionar proyectos que involucren el uso de Arduino o la soldadura de componentes. De igual manera, se obtuvieron enseñanzas que servirán para optimizar los resultados en un futuro, tales como ajustar de una mejor manera la escala de los instrumentos para realizar las mediciones o tener en cuenta que se dispongan de todos los componentes necesarios.

En conclusión, se logró familiarizarse con el diseño e implementación de un proyecto electrónico y en particular con uno basado en un microprocesador como el elegido. Además, mediante las mediciones realizadas se pudo poner en práctica los conocimientos presentados por el curso y, gracias a estos, corroborar la certeza y el sentido lógico de las mismas.

8. Anexo

8.1. Cálculo del valor de las resistencias requeridas

A raíz de lo conversado con los docentes en clase y tras complementarlo con lo investigados por los integrantes del grupo, se llegó a que la caída de tensión promedio en un led ronda los 1.6-2.0V, mientras que la corriente entregada por el Arduino se encuentra aproximadamente en el rango de los 15-20mA. Tomando el siguiente circuito como un esquema equivalente a la situación real de la actividad, podemos calcular el valor estimado que deben tener la resistencias necesitadas.



Tomando $V_s = 5V$ (tensión entregada por Arduino), $V_f = 1.8V$ y $I = 17.5mA$. El cálculo para hallar el valor de R es el siguiente:

$$R = \frac{V_s - V_f}{I} \Rightarrow R \approx 182,86\Omega \quad (3)$$

8.2. Tiempo de crecimiento (t_c) e incertidumbre

En este caso en la imagen se puede observar que la escala de tiempo es de $2.50 \mu s$ y que la cantidad de divisiones que respectivas al t_c son 1.2. Por ende, se puede afirmar que:

$$t_c = 2,50 \frac{\mu s}{div} * 1,20 div = 3\mu s \quad (4)$$

Mientras que la incertidumbre se puede obtener como:

$$\Delta t_c = 3\%t_c(exactitud) + 3\%t_c(alinealidad) + \frac{1}{2}div(apreciación)$$

$$\Delta t_c = 90ns + 90ns + 0,25\mu s$$

$$\Delta t_c = 0,43\mu s$$

Finalmente, se llega a:

$$t_c = (3,0 \pm 0,5)\mu s \quad (5)$$

8.3. Tiempo de decrecimiento (t_d) e incertidumbre

Análogamente al caso anterior, se obtiene que:

$$t_d = 1,0 \frac{ms}{div} * 3div = 3ms \quad (6)$$

$$\Delta t_d = 90\mu s + 90\mu s + 0,1ms$$

$$\Delta t_d \approx 0,1ms$$

Siendo así:

$$t_d = (3,0 \pm 0,1)ms \quad (7)$$

8.4. Código Arduino

- <https://github.com/RiedelNicolas/SimonForArduino>
- <https://github.com/evert-arias/EasyButton>