

# TALLER DE PROYECTO

## II

### INFORME FINAL

Implementación de sable de luz de Star  
Wars en EDU-CIAA

Carrera: Ingeniería en computación

Barral Adrián, Ercoli Juan Martín, Matto Joaquín,  
Pacheco Nicolás

**13 de febrero de 2024**



FACULTAD DE INGENIERÍA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA



# INDICE

1.	Introducción .....	5
1.1.	Motivación .....	5
1.2.	Características principales del proyecto.....	5
2.	Objetivos del proyecto .....	7
2.1.	Objetivo central.....	7
2.2.	Objetivos primarios.....	8
	Objetivos secundarios .....	9
3.	Análisis de requerimientos.....	11
3.1.	Requerimientos funcionales primarios .....	11
3.2.	Requerimientos funcionales secundarios .....	11
3.3.	Requerimientos no funcionales.....	12
3.4.	Hardware a utilizar.....	12
4.	Diseño de hardware.....	13
4.1.	Introducción .....	13
4.2.	Bloque acelerómetro .....	13
4.3.	Bloque RJ45.....	14
4.4.	Bloque de sensores touch.....	16
4.5.	Bloque de DAC R2R .....	17
4.6.	Bloque de tira de LEDs WS2812B .....	19
4.7.	Botón encoder .....	20
4.8.	Amplificador - PAM8403 .....	21
4.9.	Bloque de sub-sistema de alimentación.....	22
5.	Diseño del firmware, simulación y depuración .....	25
5.1.	Arquitectura del firmware.....	25
5.2.	Pseudocódigo de la arquitectura de firmware.....	26



5.3.	Módulo controlador del botón .....	26
5.4.	Módulo controlador de la tira de LEDs .....	27
5.4.1.	Protocolo de la tira de LEDs .....	27
5.4.2.	Solución para el protocolo de transmisión de datos.....	29
5.4.3.	Efectos de la tira de LEDs .....	30
	C. Implementación del Efecto Onda Sinusoidal .....	31
5.5.	Módulo controlador de sensores touch .....	32
5.6.	Módulo controlador de acelerómetro .....	33
5.7.	Módulo controlador de sonido .....	34
5.7.1.	Generación del archivo fuente de sonido .....	34
5.7.2.	Obtención de los valores hexadecimales en crudo .....	37
5.8.	Estados del sable laser .....	37
5.9.	Resumen del software .....	39
6.	Ensayos y mediciones .....	39
7.	Cronograma preliminar .....	45
8.	División de tareas del grupo .....	46
9.	Bibliografía .....	49
10.	Guía rápida de uso del sable láser.....	50
11.	Conclusiones .....	51
11.1.	Cumplimiento de objetivos .....	51
11.2.	Cumplimiento de requerimientos.....	53
11.3.	Conclusiones generales .....	54
12.	Anexos .....	57
12.1.	Circuito esquemático .....	57
12.2.	Circuito esquemático R2R .....	57
12.3.	PCB R2R.....	59
12.4.	PCB PONCHO.....	62



12.6. Lista de materiales .....	66
---------------------------------	----



## 1. Introducción

En una galaxia muy, muy lejana, llena de aventuras y tecnologías ajenas a nuestro tiempo, existe un objeto emblemático que cautivo a varias generaciones: el sable de luz.

Inspirado por la saga Star Wars, este proyecto tiene como fin traer este objeto a la realidad a través de la implementación de una réplica funcional.



*Figura 1.1 Darth Vader*

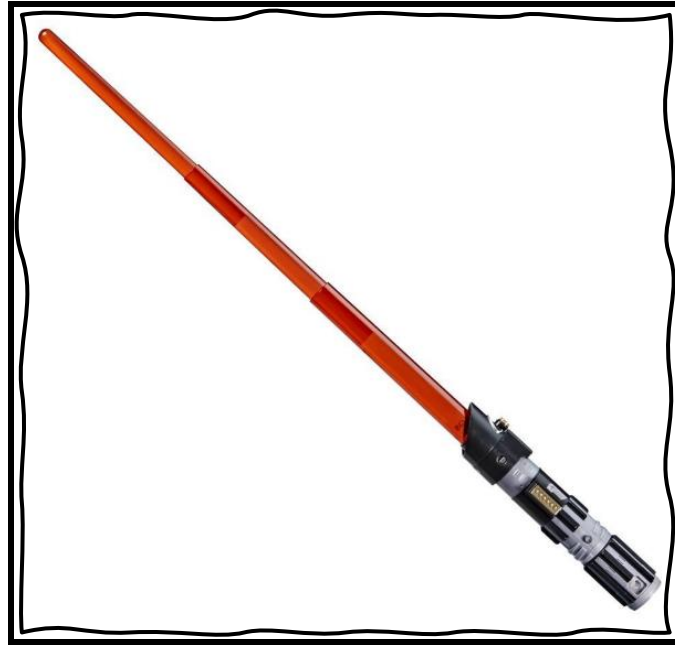
### 1.1. Motivación

La motivación que impulsa a realizar este proyecto es la pasión de los integrantes del grupo por la ciencia ficción, tecnología y creatividad. El sable de luz es un símbolo del avance tecnológico, y su recreación nos desafía a explorar varios aspectos técnicos.

Es un desafío integrador ya que se deben combinar conceptos de la electrotecnia, programación y diseño para lograr una réplica funcional. Pero más allá de la fascinación tecnológica, el proyecto busca ser fuente de inspiración para futuros ingenieros y fanáticos de la ciencia ficción, demostrando que los límites de la imaginación pueden superarse con habilidad y dedicación.

### 1.2. Características principales del proyecto

El proyecto se centra en construir una réplica del sable de luz del personaje Darth Vader de la saga Star Wars. En la Figura 1.2, se puede apreciar un juguete con el modelo que utiliza este personaje.



*Figura 1.2 Réplica del sable de luz de Darth Vader hecho por Hasbro*

Las características principales a tener en cuenta para este proyecto son:

- I. Desarrollo utilizando como placa de desarrollo a la EDU-CIAA-NXP [1]
- II. Diseño del mango de plástico en impresora 3D: la réplica contará con el mango de plástico de Darth Vader
- III. Hoja iluminada: la hoja del sable de luz se compondrá de un tubo de plástico transparente con una tira de LEDs en su interior
- IV. Sensor acelerómetro: se colocará un acelerómetro en la mitad de la hoja para detectar los movimientos y sincronizar efectos visuales.
- V. Elemento de control: se utilizará un pulsador para el manejo del encendido de los LEDs de la hoja del sable y el cambio de colores de estos
- VI. Sonido: se utilizará un sistema de audio para reproducir los efectos de sonidos del sable según el encendido/apagado y el movimiento
- VII. Sensores touch capacitivos: Se integrarán sensores touch capacitivos en la hoja del sable de luz para permitir interacciones adicionales con el dispositivo. Estos permiten simular los efectos visuales de choque de sable

## 2. Objetivos del proyecto

Los objetivos de este proyecto se centran en la creación de una réplica del sable de luz de Darth Vader, no sólo para emular la apariencia del simbólico sable de Star Wars, sino que la idea es ofrecer una experiencia interactiva.

A continuación, se presenta el diagrama en bloques del sistema:

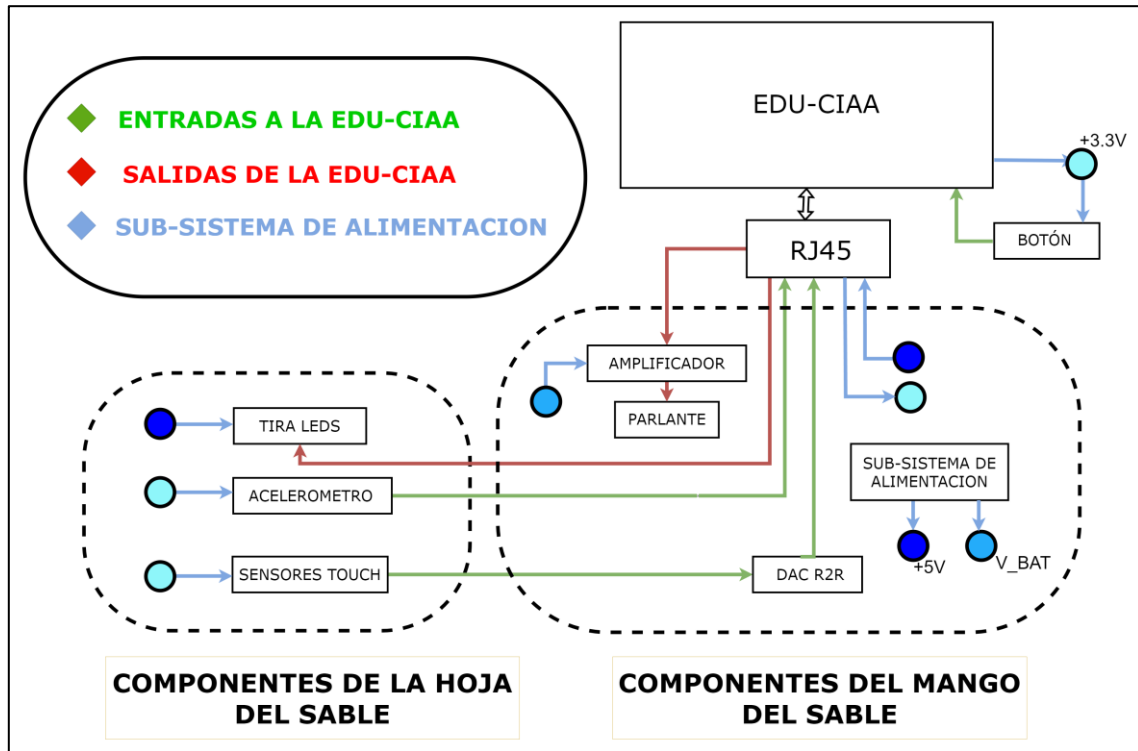


Figura 2.1 Diagrama en bloques del sistema propuesto

### 2.1. Objetivo central

La construcción del sable de luz encuentra su base en un mango de plástico concebido para albergar los elementos esenciales necesarios para recrear la experiencia.

En el mango, los elementos que se albergan son:

- Cables que van a la hoja del sable: conducirán la energía y los datos hacia la hoja del sable, tanto para medir con el acelerómetro como para el funcionamiento de la tira de LEDs
- Botón de encendido/apagado: este botón estará en la cara exterior del mango
- Conectores: para los cables que tienen dirección hacia afuera del mango; para los cables que van a la hoja de luz
- Sistema de audio: para generar los efectos de sonido
- Sub-sistema de alimentación: se encarga de alimentar los componentes de la hoja del sable y el sistema de audio



En cuanto a la hoja del sable, esta está equipada con:

- Tiras de LEDs: para simular el haz de luz
- Acelerómetro: para detectar los movimientos en el sable, y, así poder realizar efectos visuales y sonoros en el mismo
- Sensores touch capacitivos: se encargan de detectar si una persona toca la hoja del sable

Debido a que la implementación se realiza con la EDU-CIAA, las dimensiones [2] de esta no cuadran para encapsularla dentro del mango de plástico del sable [3], por lo que a priori, se decide acomodar la computadora por fuera del sable, dentro de una caja tematizada al estilo del pecho de la armadura de Darth Vader, como se puede ver en la Figura 2.2.



*Figura 2.2 Pecho de la armadura de Darth Vader*

## 2.2.Objetivos primarios

Para lograr caracterizar las funciones principales del sistema de un sable laser se establecen los siguientes objetivos primarios:

- Implementación del Código:
  - Desarrollar un código que permita las siguientes funciones:
    - Lectura del botón: facilitar el encendido y apagado del sable
    - Control del acelerómetro: detectar los movimientos del sable, modulando la señal para la generación de sonido





- Manejo de LEDs: controlar la iluminación del sable, ajustando la intensidad y el brillo según el movimiento
- Sincronizar el funcionamiento: es requerida una sincronización entre los componentes para lograr una buena experiencia. Los componentes deben obedecer un comportamiento coordinado que puede tomar el sistema del sable
- Implementación de un sistema de sonido: el agregado de este componente implica una complejidad extra a la hora de sincronizar con el resto de los componentes. Este objetivo no solo se centra en el agregado de hardware adecuado para emitir el sonido, sino también en el código donde la generación de sonidos debe responder al:
  - Encendido del sable
  - Apagado del sable
  - Movimiento del sable
  - Detectar un cese de movimiento abrupto
- Implementación de sensores táctiles en la hoja del sable: para lograr un efecto táctil en la hoja del sable se propone la incorporación de sensores táctiles en la superficie de la hoja. Estos sensores detectarán el contacto físico y podrán traducirlo en cambios en la iluminación, brindando así una experiencia más inmersiva
- Implementación de una batería recargable: se propone dotar al sable de luz de una batería autónoma que permita su funcionamiento sin necesidad de estar conectado a una fuente de alimentación externa. Además, se contemplará que sea recargable.

### Objetivos secundarios

Además de los objetivos primarios que se enfocan en las funciones esenciales del sable, de ser posible en tiempo, para mejorar la experiencia inmersiva, se contempla implementar los siguientes objetivos secundarios:

- Implementación de un módulo Bluetooth para cambio de colores con aplicación móvil:
  - Integrar un módulo Bluetooth que permita cambiar los colores del sable de luz a través de una aplicación móvil. Esto proporcionará a los usuarios un control intuitivo y conveniente sobre la paleta de colores, permitiendo una personalización instantánea según sus preferencias
- Implementación de un módulo WI-FI para el control de los colores:
  - Se propone introducir la capacidad de controlar los colores del sable a través de una conexión WI-FI y una interfaz web. Esto permitiría a los usuarios



personalizar y cambiar los colores del sable de luz de manera remota, agregando una dimensión más creativa al proyecto

- Implementación de Memoria SD para almacenar efectos sonoros:
  - Incorporar una memoria SD que permita almacenar una variedad de efectos sonoros. Esto ampliará la gama de sonidos que el sable de luz puede reproducir, ofreciendo una experiencia auditiva más rica y dinámica para los usuarios
- Implementación de encoder:
  - Integrar un encoder que posibilite controles adicionales. Este componente ofrecerá una forma precisa y ergonómica de ajustar parámetros, como podría ser el cambio de color de los LEDs



### **3. Análisis de requerimientos**

En esta sección, se definen los requerimientos que darán forma al sable de luz, dividiéndolos en funcionales y no funcionales.

#### **3.1. Requerimientos funcionales primarios**

- I. **Encendido del sable:** la tira de LEDs debe encenderse cuando se oprime el botón por primera vez, indicando así que el sable está activado y listo para su uso
- II. **Apagado del Sable:** la tira de LEDs debe apagarse cuando se mantiene pulsado el botón durante un segundo mientras el sable está encendido, indicando que el sable se ha apagado correctamente.
- III. **Emisión de Sonido y Luz Continua:** Mientras el sable esté encendido, debe emitir un sonido de fondo continuo y mantener los LEDs encendidos
- IV. **Ajuste de Intensidad y Sonido con el Movimiento:** la Intensidad de los LEDs y el sonido aumentan con el movimiento
- V. **Efectos al Encender o Apagar el Sable:** al encender o apagar el sable, se debe producir un sonido distintivo y un efecto visual en los LEDs para indicar claramente el cambio de estado del dispositivo
- VI. **Reacción al Tacto:** el sable de luz reacciona al tacto en determinadas zonas de la hoja, esto produce un aumento en la intensidad de LEDs en dicha zona y genera el efecto de sonido correspondiente

#### **3.2. Requerimientos funcionales secundarios**

- I. **Cambio de color mediante encoder:** implementar la capacidad de cambiar el color de la tira de LEDs al girar un encoder
- II. **Memoria SD para efectos de sonido:** integrar una memoria SD que permita precargar una biblioteca de efectos de sonido. Esto ofrecerá una variedad de opciones sonoras para que los usuarios elijan, mejorando así la experiencia auditiva del sable de luz.
- III. **Control bluetooth a través de aplicación:** establecer la capacidad de conectarse a través de una aplicación Bluetooth dedicada que permita a los usuarios controlar tanto el efecto de colores como el sonido del sable de luz
- IV. **Control por Wi-Fi e interfaz web:** habilitar la conexión a través de Wi-Fi a una interfaz web que permita el control del efecto de color y sonido en tiempo real



### 3.3.Requerimientos no funcionales

- I. Resistente a los movimientos bruscos
- II. Funcionamiento autónomo
- III. PlugAndPlay: el sistema no requiere configuración previa para ser usado
- IV. Entrega en tiempo y forma

### 3.4.Hardware a utilizar

- I. 1xEDU-CIAA con MCU LPC4337
- II. 1xBotón\encoder
- III. 1xTira LED WS2812B 111 LEDS (1.85 metros)
- IV. 1xAcelerómetro MPU6050
- V. 1xPack de baterías de litio
- VI. 1xModulo de carga TP4056
- VII. 2xModulo fuente DC-DC STEP-UP XL6009
- VIII. 1xSwitch de prendido
- IX. 3xSensores touch TTP223
- X. 1xAmplificador PAM8403
- XI. 1xParlante 1W

## 4. Diseño de hardware

### 4.1. Introducción

En esta sección se explicará con mayor detalle el diagrama en bloques presentado en la Figura 3, explicando características e interconexión de los componentes que conforman cada uno. Se agrega en este capítulo el concepto de poncho para la EDU-CIAA.

### 4.2. Bloque acelerómetro

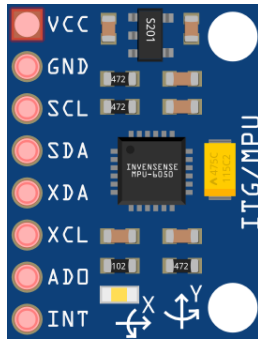


Figura 4.1 Esquema gráfico del MPU6050

Este bloque se compone del sensor acelerómetro MPU6050 (ver Figura 5). Este es un sensor de seis ejes que combina un giroscopio y un acelerómetro en un solo dispositivo.

Este sensor utiliza el protocolo de comunicación I2C que consiste en utilizar un bus bidireccional para la transmisión de datos entre uno o varios maestros (la EDU-CIAA-NXP) y uno o varios esclavos, en este caso el MPU6050. El bus I2C se compone de dos líneas: línea de datos (SDA) y línea de clock. (SCL).

En este protocolo, la transferencia de datos es inicializada siempre por un maestro; el esclavo reacciona. La sincronización y la velocidad de transferencia son controladas por la línea de reloj, mientras que la línea de datos transporta la información entre los dispositivos.

Los cables que se conectan al SDA y SCL del MPU6050 están en el conector RJ45.

Se decide utilizar el MPU6050 debido a su capacidad para medir aceleración y giro, el giro se utiliza para medir el módulo de la aceleración angular en el sable en dos ejes, descartando la lectura de aceleración angular para el eje de rotación que mantiene la simetría del cilindro (o mango).

Respecto a las características eléctricas del MPU6050:

Tabla 1 Características eléctricas del MPU6050

CARACTERISTICA	VALOR
Voltaje de operación (VDD)	2.375V-3.46V
Voltaje de entrada lógica (VIH)	0.7*VDD
Voltaje de entrada lógica (VIL)	0.3*VDD
Consumo de corriente	3.9mA (modo normal) 8μA (modo reposo)
Rango de temperatura	-40°C a +85°C

El MPU6050 [5] se alimenta directamente desde la EDU-CIAA.

Dado que el MPU6050 puede operar en un rango de voltaje que incluye los 3.3V de la EDU-CIAA, es eléctricamente compatible en términos de voltaje de alimentación. El consumo de corriente es coherente respecto de lo que puede entregar como máximo la EDU-CIAA, que unos 300mA.

#### 4.3. Bloque RJ45

Este bloque está compuesto por un conector RJ45, la idea de este es facilitar la conexión entre los pines del poncho de la EDU-CIAA y los componentes que están en el cable.

Estos conectores tienen 8 pines.

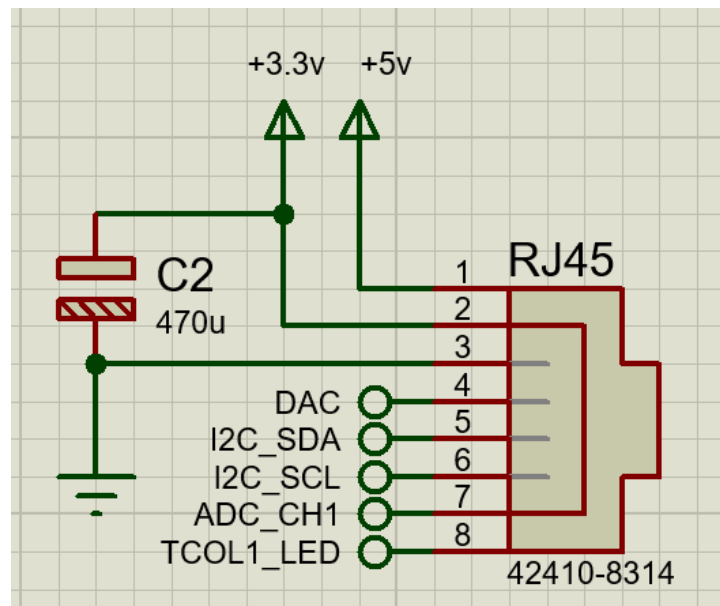


Figura 4.2 Esquema de conector RJ45 en Proteus

#### Consideraciones de la disposición de pines del RJ45

Se diseñó esta disposición de pines por varios motivos, principalmente para disminuir el ruido inducido a la señal de la tira neopixel.

- Los primeros 3 pines están reservados para transporte de energía, los +5v en el pin 1, va a estar trenzado con el cable de +3.3v en el pin 2, el pin 3 tiene la referencia de tierra, es el pin de transporte de energía que está trenzado junto a la salida del DAC del pin 4.
- El pin del DAC, transporta una señal de sonido (aproximadamente con un BW hasta 8kHz) por lo que no le debe interferir el ruido de conmutación inducido en el pin 1 (que para el módulo step-up MT3608 puede ser de 200KHz).
- Además, como el cable de sonido (el DAC del pin 4) va a estar trenzado junto a la referencia de tierra, debería hacerle efecto “apantallamiento” que es beneficioso para limitar los ruidos inducidos en el cable.



- El pin 5 y pin 6 tienen los cables de comunicación I2C ( sda y scl ), se decidió ponerlos juntos para que reciban el mismo nivel de ruido.
- El par de pines 7 y 8 lleva la señal analógica del estado de los touch, para leer con el ADC, y el cable de señal neopixel.
- El pin 8 lleva la información del neopixel, señal que comprobamos que es muy importante que se deforme lo menos posible, y se consideró conectar el pin, lo más lejos posible de los cables de alimentación, y que sea trenzado con el cable de señal que tenga menor ancho de banda, justamente la señal de la salida del circuito R2R solo varia cambia de estado cuando se presiona o suelta algún touch, cambio que sucede a lo sumo algunas veces por segundo.
- El pin 7 lleva la señal analógica de lectura de los touch, no transfiere corriente, ya que es una entrada ADC, que mide el nivel de tensión del circuito R2R.

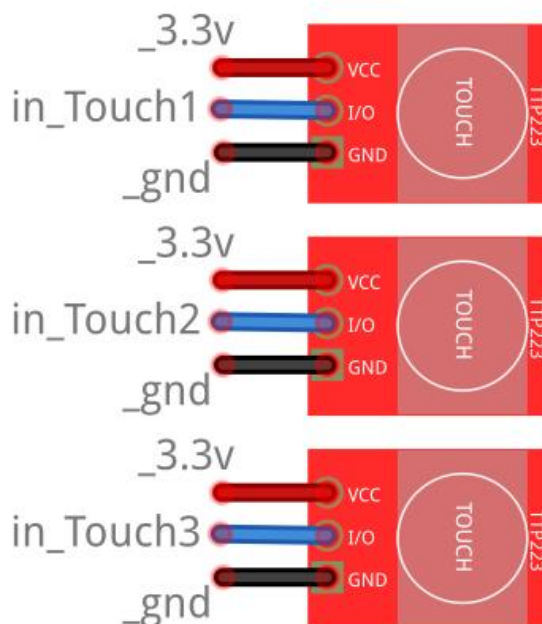
A continuación, en la tabla 2 se describe un resumen de los pines mencionados en orden, y su función.

*Tabla 2 características de los terminales del conector RJ45*

TERMINALES	FUNCIÓN
<b>+5V</b>	Alimentar a la EDU-CIAA a través del sub-sistema de alimentación
<b>+3.3V</b>	Alimentación de sensores touch y acelerómetro
<b>gnd</b>	Referencia de tierra
<b>DAC</b>	Transmite la señal de sonido
<b>I2C_SDA</b>	Extiende la línea de datos del bus I2C
<b>I2C_SCL</b>	Extiende la línea de reloj del bus I2C
<b>ADC_CH1</b>	Lee la información analógica de los sensores touch
<b>TCOL1_LED</b>	Comunicación con la tira de leds



#### 4.4. Bloque de sensores touch



*Figura 4.3 Esquema gráfico de sensores touch*

El bloque de tres sensores touch se compone de sensores capacitivos que detectan la presencia de un objeto conductor (como un dedo humano) sin necesidad de presionar físicamente el sensor. Estos sensores funcionan midiendo los cambios en la capacitancia de un área determinada. Cuando un objeto conductor se acerca al sensor, la capacitancia del área cambia y el sensor puede detectar esta variación.

Respecto a sus características eléctricas:

*Tabla 3 Características eléctricas del TTP223*

CARACTERISTICA	VALOR
<b>Voltaje de operación (VDD)</b>	2.0V-5.5V
<b>Corriente de operación (IDD)</b>	1.5 $\mu$ A-3 $\mu$ A (bajo consumo) 6.5 $\mu$ A-13 $\mu$ A (modo fast)
<b>Tiempo de respuesta</b>	Máximo 60ms (bajo consumo)
<b>Corriente de salida en ALTO</b>	Puede conducir corriente hasta 4mA
<b>Temperatura de operación</b>	-20°C - +70°C

El hecho de que la corriente de salida del pin de datos pueda ser de hasta 4mA no es un inconveniente debido a que el circuito DAC R2R al que se conectan los sensores touch consume menos de 2mA hacia TIERRA.



Cada uno de los sensores se alimenta por medio del terminal que sale del conector RJ45 “3.3v”.

#### 4.5. Bloque de DAC R2R

El propósito para incluir este circuito es optimizar la cantidad de pines de comunicación ya que estamos limitados a los 8 cables del conector RJ45, y ya tenemos todos ocupados, por lo que se tiene que codificar la información de estos bits en una señal para transmitirse. En resumen, usaremos un DAC para codificar el estado de los sensores táctiles y leeremos el valor con un ADC, y posteriormente procesaremos el valor recibido para detectar cuales sensores táctiles fueron activados.

Estos sensores serán alimentados con la salida del regulador de 3.3V de la EDUCIA.

Se describe en la figura 4.4 el diagrama eléctrico del DAC. Y se analiza el caso de mayor consumo eléctrico de la salida del sensor táctiles

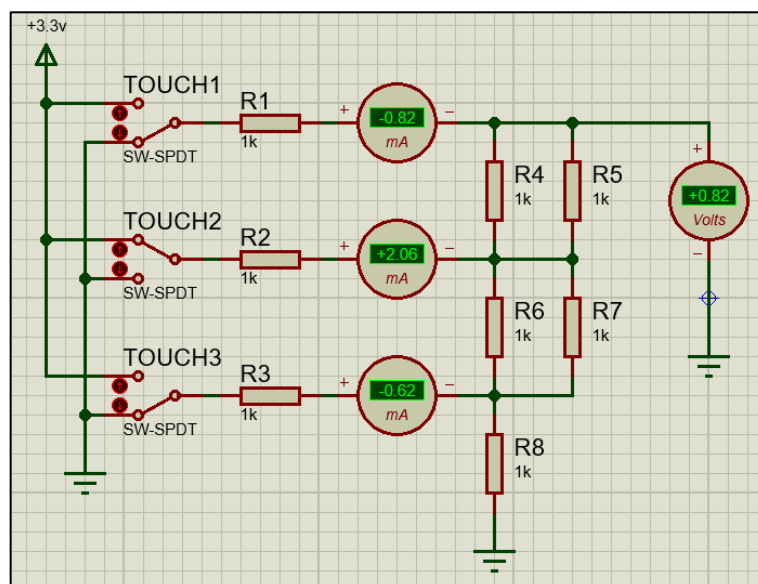


Figura 4.4 Simulación de circuito de DAC R2R conectado a los sensores touch en Proteus

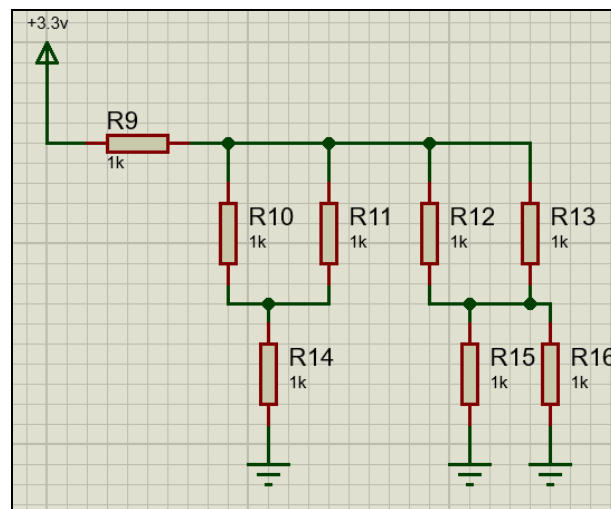
El DAC R2R (Digital a Analógico con resistencias en configuración R y 2R) es un circuito que convierte señales digitales en señales analógicas. Está compuesto por una red de resistencias en una configuración específica que permite obtener diferentes niveles de voltaje analógico a partir de una entrada digital.

La conexión del bloque de sensores touch al DAC R2R tiene el propósito de transformar las señales digitales generadas por los sensores touch en señales analógicas. Este tiene una salida que va a la entrada de alta impedancia de la EDU-CIAA del ADC.

Su funcionamiento obedece los siguientes pasos:



Se utilizan  $R=1\text{k}\Omega$ .



*Figura 4.5 Diagrama del circuito equivalente para el cálculo de la corriente máxima teórica*

A continuación, se calcula la corriente de salida máxima teórica de un sensor táctil, la cual se da cuando los sensores tienen la siguiente configuración de estados

Touch3=OFF;	Touch2=ON;	Touch1=OFF
-------------	------------	------------

se calculan la corriente total en los componentes resistivos con la ley de ohm

$$I = \frac{V}{R}$$

La resistencia máxima equivalente de salida es:

$$1k + \left(\frac{3k}{2} \dots 1k\right) = 1k + \left(\frac{\frac{3}{2}k}{\frac{5}{2}}\right) = 1k + \frac{3}{5}k = 1600\Omega$$

$$I_{Touch2} = \frac{3.3[V]}{1600[\Omega]} = 2.0625 [mA]$$

La corriente máxima no supera los 4mA de IOH. Por lo que garantizamos que los sensores táctiles no se van a quemar al utilizarlos.

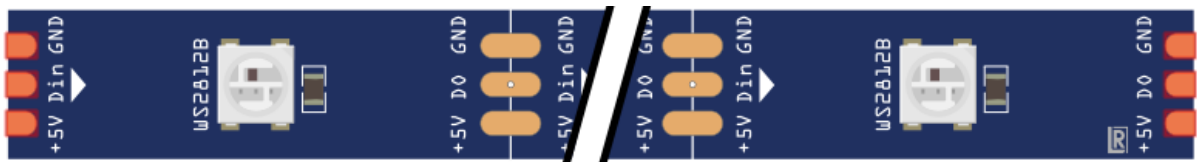


No se calcula la corriente que ingresa la EDU-CIAA, porque el pin de lectura ADC está configurado en alta impedancia, por lo que la corriente absorbida en la medición es despreciable, y al estar conectado el circuito del R2R al regulador de 3.3v de la EDU-CIAA, nos aseguramos que la máxima medición de tensión nunca puede superar este valor (que es el máximo admitido por el pin ADC).

A futuro puede modificarse la cantidad de sensores touch dependiendo del efecto deseado en las pruebas preliminares del sable.

#### 4.6. Bloque de tira de LEDs WS2812B

La tira de LEDs WS2812B es el componente esencial del sable láser. Se seleccionó esta tira de LEDs inteligente debido a su capacidad para controlar el color de cada píxel de manera independiente, lo que permite una amplia gama de efectos visuales.



*Figura 4.6 Esquema gráfico de una porción de la tira de LEDs*

La tira LED cuenta con tres pines: +VCC (alimentación), DATA (datos) y GND (tierra). Estas tiras Led tienen una particular forma de comunicación, se envían los bits de datos en serie, los paquetes de 24bits representan unívocamente cada color. El paquete de color se recibe por el primer píxel, que se apropia del primer paquete y le comunica al siguiente píxel todos los paquetes siguientes. Este proceso se repite a lo largo de la tira, permitiendo refrescar los LEDs más de 60 veces por segundo. En este caso, tenemos 111 LEDs conectados al prototipo de sable. Este componente es el que lleva el mayor consumo eléctrico de todo el sable.

La hoja de datos del LED no proporciona un valor específico para el consumo eléctrico, sino que ofrece una estimación general. Indica un consumo de 20mA por color, lo que se traduce en un total de 60mA por píxel. Esto sugiere un consumo máximo teórico de 6.66A para toda la tira de LEDs, una cifra que puede parecer exagerada a primera vista.

Para obtener una medición más precisa del consumo, se llevaron a cabo pruebas empíricas evaluando la tira completa bajo diferentes efectos luminosos. Estas pruebas, documentadas en un video adjunto [6], se realizaron utilizando una fuente de 5V y un amperímetro en serie. Los resultados mostraron que, dependiendo del efecto de luz seleccionado, el consumo varía entre 700mA y 1700mA.

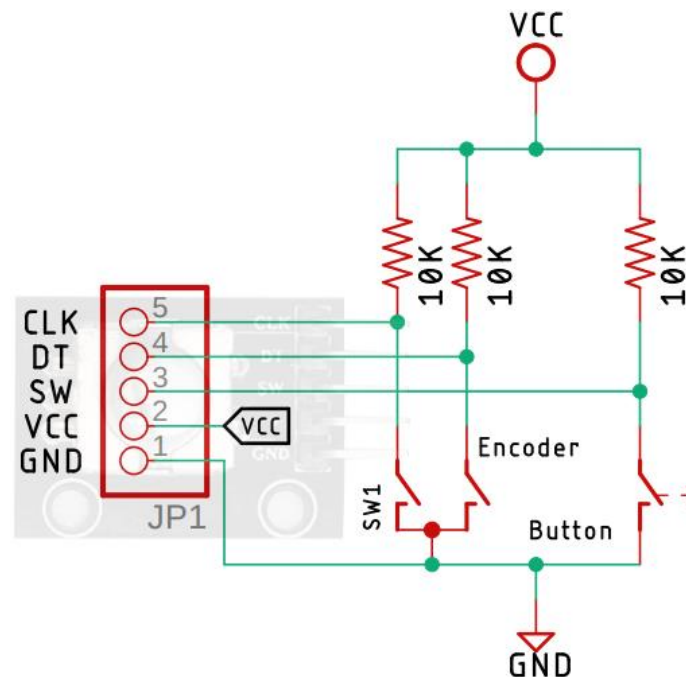
Dado estos hallazgos, el sistema de alimentación ha sido diseñado para proporcionar al menos 2A a la tira de LEDs, garantizando así un margen de seguridad adecuado en todas las situaciones.

*Tabla 4 Características eléctricas de la tira de LEDs*

TERMINALES	FUNCIÓN
VCC	Alimentación: Rango de 3.5v a 5.5v. Consumo: Menor a 2A
DIN	Entrada de datos: Utiliza un protocolo propietario. Consumo: Menor a 1mA
GND	Referencia de tierra
DO	Salida de datos: En esta implementación no se utiliza

#### 4.7. Botón encoder

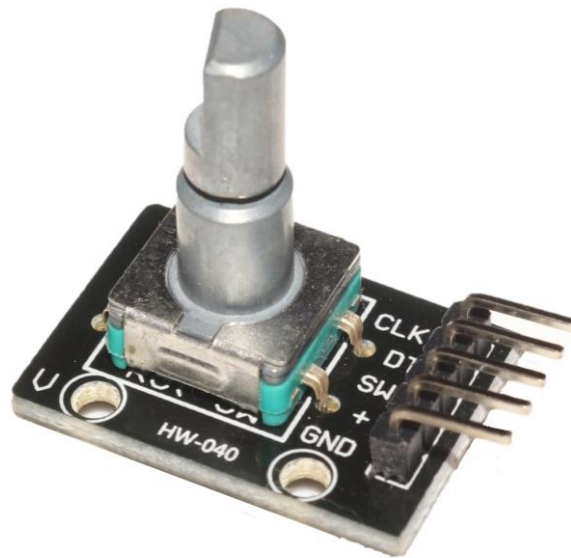
Se eligió que sea un botón encoder para que el proyecto sea expansible y, a futuro, se pueda usar el encoder para cambiar el color del sable, por otro lado, este cuenta con una resistencia de pull-up de 10kΩ. Iría colocado en el poncho.



*Figura 4.7 Diagrama de botón encoder*

Se utiliza la función de botón como interruptor de encendido y apagado, mediante el pin 3 (SW) a un pin de I/O, y el pin 1 conectado a tierra (GND) en el poncho. Este pulsador indica

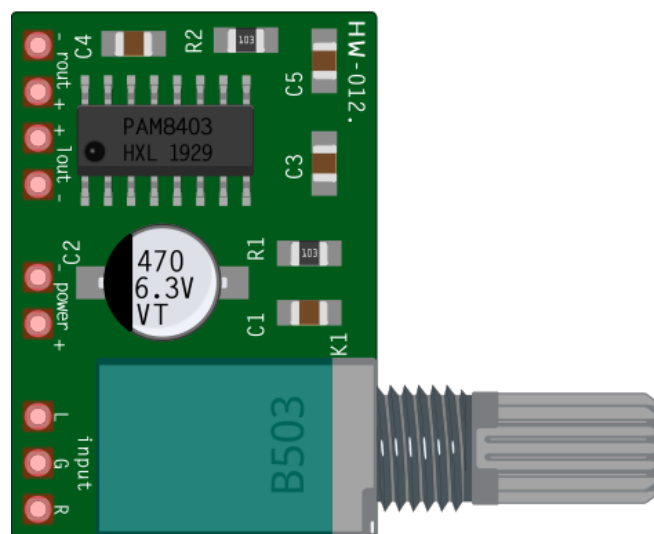
una señal que será aprovechada por el programa del sable para gestionar las transiciones de encendido a apagado y viceversa.




*Figura 4.8 Encoder HW-040*

#### 4.8. Amplificador - PAM8403

Este componente le da la capacidad al circuito de manejar la potencia para mover parlantes dinámicos. Es el intermediario entre las señales generadas por la EDU-CIAA y las señales eléctricas que impulsarán los parlantes. El modelo adquirido cuenta con un potenciómetro que permite ajustar la ganancia interna del amplificador. Tras ser calibrado adecuadamente según su aplicación, este ajuste permanecerá transparente para el usuario, garantizando una experiencia de uso fluida.



*Figura 4.9 Esquema gráfico del PAM8403*



El módulo amplificador PAM8403 es un amplificador tipo-D estéreo que maneja hasta 2 parlantes de 3W RMS en simultaneo. Para los propósitos de este proyecto los sonidos generados son en mono, como se quiere utilizar de la mejor manera los componentes, se conecta la entrada estéreo a la misma fuente de audio. Para esto se cortocircuitan los pines Input\_R e Input\_L, y se conectan a la señal de la EDU-CIAA.

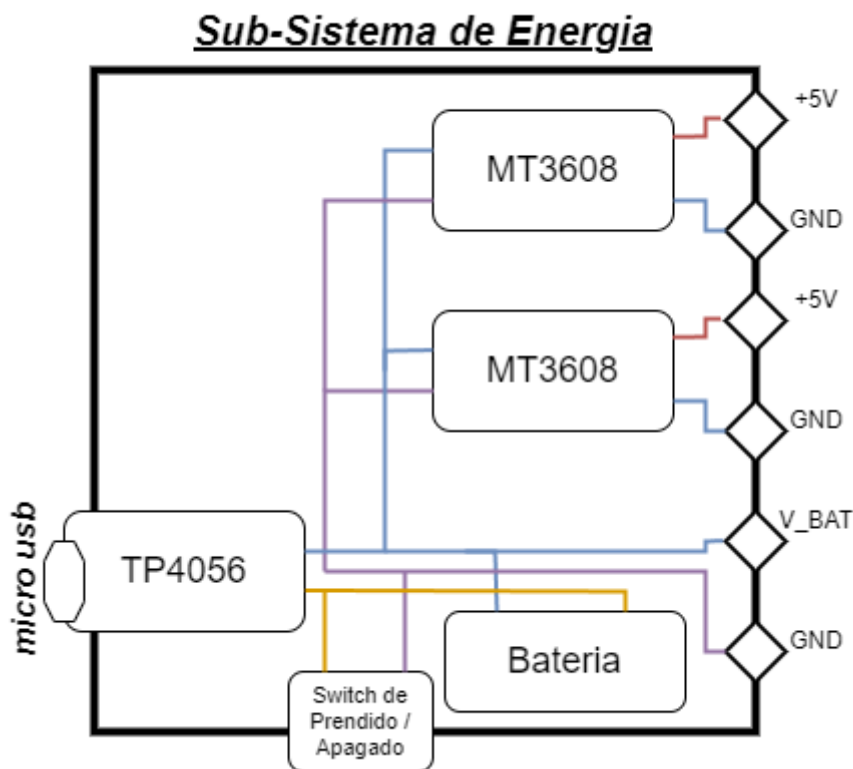
*Tabla 5 características eléctricas del PAM8403*

Pines	FUNCIÓN
<b>VCC</b>	Alimentación directa de la batería del sub-sistema de alimentación; consumo menor a 1A
<b>GND</b>	Referencia de tierra
<b>IN_L + IN_R</b>	Entrada de señal de audio
<b>R_Out</b>	Salida para el parlante derecho
<b>L_Out</b>	Salida para el parlante izquierdo

#### 4.9.Bloque de sub-sistema de alimentación

El sub-sistema es el encargado de suministrar la energía a todo el sistema principal, se opta por utilizar baterías de litio por su buena capacidad y la fácil manera de conseguirlas, estas baterías deben estar acompañadas de su correspondiente circuito de carga para poder usarlas, en este caso decide utilizar el TP4056, que es un módulo de carga de baterías con protección de sobre carga, no supera 1A de corriente de carga, para no recalentar las baterías y no desgastar innecesariamente su capacidad.

También se utiliza el módulo de fuente Step-Up MT3608, que es una fuente conmutada DC-DC elevadora de tensión, que se calibra mediante su potenciómetro multivuelta a 5.1V de salida, y luego se pegó con silicona la perilla para que no cambie de valor.



*Figura 4.10 Diagrama de conexionado del sub-sistema de energía*

### **Limitaciones Eléctricas:**

La salida directa de la batería no tiene protecciones, lo que permite a los sistemas conectados a ella sacar picos de corriente sin impedimentos, esta salida tiene la limitación física de la máxima corriente que puede entregar la batería sin dañarse, para el alcance de este proyecto, estamos en los márgenes razonables del uso directo de la batería.

La salida regulada de 5V está limitada a lo que puede entregar el MT3608, que según su datasheet es de hasta 2A, actualmente estamos cercanos a los límites de su funcionamiento, por lo que tiene en cuenta el uso de un disipador de calor para los usos extendidos de sable. Además, al ser una fuente conmutada, genera ruido de alta frecuencia en su salida, que se decidió filtrar con un capacitor de 1000uF 6.3V en paralelo a la salida de 5V. Esto principalmente para que cuando los leds tomen picos de consumo, la EDU-CIAA no se apague repentinamente.

### **Consumo de componentes:**

Como todos los sensores están alimentados por la EDU-CIAA, a través de la toma de 3.3v, solo se considera el peor caso de consumo de la EDU-CIAA con la tira de leds. También se excluye el consumo del amplificador de audio para este cálculo, ya que está conectado directamente a la batería y no aporta consumo al regulador de 5V.

En Resumen, el consumo general del regulador esta acotado a 2A, por lo que es posible alimentar todo con 1 solo modulo MT3608, por una cuestión de potencia se requirió utilizar 2 módulos porque se el mismo se recalentaba usando solo uno.

*Tabla 6 Tabla de consumo de corriente de componentes*

Componente	Consumo de corriente [ mA ]
Tira led WS2812b con 111 pixeles	1700
EDU-CIAA	300
Amplificador - PAM8403	-Conectado a la bateria-
DAC - R2R	-Conectado a la edu-ciaa-
Acelerómetro - MPU6050	-Conectado a la edu-ciaa-
sensores Touch - ttp223	-Conectado a la edu-ciaa-
Boton - Encoder	-Conectado a la edu-ciaa-

Se puede realizar una estimación de la autonomía del dispositivo considerando ciertos parámetros de consumo. La línea de 5V tiene un consumo limitado a 2A, y se espera que el amplificador no consuma más de 2W, lo que resulta en un consumo total estimado de 2550mAH.

Para alimentar este circuito, se cuenta con un conjunto de cuatro baterías de celular conectadas en paralelo. Cada una de estas baterías tiene una capacidad declarada superior a 1350mAH, lo que proporciona una capacidad total combinada de 5400mAH.

Las baterías están desgastadas por su almacenaje durante años, pero se estima que conservan al menos el 60% de la capacidad original dando al menos 3200mAH de capacidad real. Luego de todas estas suposiciones, se estima que la autonomía puede ser de alrededor de  $3200/2550 = 1:15\text{hs}$ . Que luego se verificará en la práctica.


Hasta acá igual al informe anterior.

#### 4.10. Diseño de la PCB

Nuestro principal objetivo a la hora de diseñar la PCB es la distribución espacial del sistema, tanto de los componentes que se colocan en la placa como el tamaño de la PCB en sí.

Dado que se busca que el producto final sea similar al sable de las películas y por lo tanto pueda ser maniobrable, necesitamos disminuir el tamaño de la PCB lo más posible. Por otra parte, la placa debe estar conectada a la EDU-CIAA coincidiendo las tiras de pines de la PCB con los puertos de la CIAA, haciendo que el ancho mínimo con el que podemos trabajar no pueda ser modificado fácilmente.





Es por esto que optamos por dividir el diseño de la placa en dos placas separadas, una pequeña de 35mm por 42mm aproximadamente, que se colocaría en el mango del sable junto con las baterías, el puerto de carga y el sistema de audio, y una placa mas grande del mismo ancho de la CIAA, la cual seria en el poncho en si de la placa.

Ambas placas estarán conectadas a través de un cable de red , por lo que ambas cuentan con un conector RJ42<sup>45</sup> para la transmisión de datos y energía.

De esta forma distribuimos los componentes liberando un poco la parte del sistema del mango la cual, de lo contrario, estaría demasiado sobrecargada haciendo muy difícil su manipulación, depuración e incluso su propio uso.

Además el poder desconectar los componentes para transportarlos por separado ayuda notoriamente a la hora de trasladarlos considerando que el sable ya es de por si bastante robusto.

En los capítulos 9.3 y 9.4 al final del informe se pueden encontrar los diseños de ambas PCB, su vista previa en 3D e imágenes del resultado final.

Respecto al prototipo final quedó muy bien disimulado el hardware.

## **5. Diseño del firmware, simulación y depuración**

### **5.1. Arquitectura del firmware**

La arquitectura elegida para el firmware del sable láser es una arquitectura híbrida que integra módulos apropiativos controlados por eventos y módulos no apropiativos controlados por polling. Esto significa que el sistema reaccionará a eventos específicos, como entradas del usuario o señales de sensores, como es el caso de la tira LED y del sonido, pero también a los resultados obtenidos del sondeo, como es el caso de los sensores touch y del botón encoder.

Por otro lado, es híbrida además porque los LEDs y el sonido son apropiativas respecto de las tareas de polling, pero no son apropiativas entre sí, ya que se requiere de la actualización constante de los LEDs para generar los efectos de luces o de las muestras para la generación de sonidos. Por lo tanto, la arquitectura combina elementos controlados por eventos con elementos que funcionan de manera “constante”.

Por cada periférico se tiene un módulo controlador, en la Figura 5.1 se puede apreciar un esquema inicial basado en capas de la arquitectura del firmware.

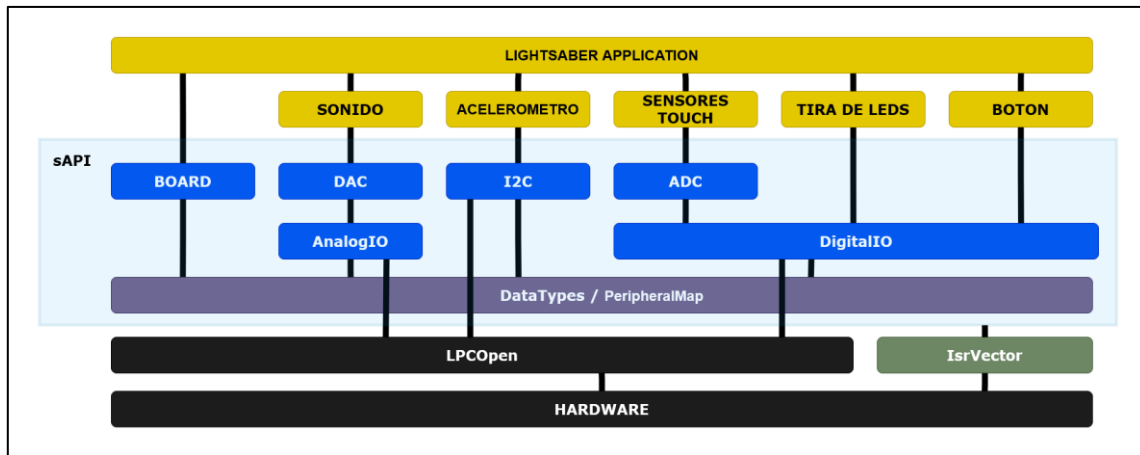


Figura 5.1 Capas de firmware del sable laser (amarillo) basado en sAPI (azul)

## 5.2. Pseudocódigo de la arquitectura de firmware

*Inicializar módulos*

*Configurar módulos*

*Establecer estado inicial del sable (apagado)*

*Mientras el sistema este activo*

*Si el sable está encendido*

*Modular luz y sonido según movimiento y detección de touch*

*Si se pulsa el botón*

*Transicionar al apagado del sable*

*Actualizar LEDs*

*Generar sonido*

*Sino*

*Si se pulsa el botón*

*transicionar al encendido del sable*

*Esperar al siguiente ciclo*

*Pseudocódigo 1 Pseudocódigo inicial de la arquitectura de firmware*

## 5.3. Módulo controlador del botón

Este módulo es responsable de gestionar la entrada del botón. Se encarga de detectar pulsaciones y generar los eventos correspondientes. En principio cuando se presione se encenderá/apagará el sable. Más adelante se plantea darle múltiples usos, debido a su naturaleza de botón encoder.



#### 5.4. Módulo controlador de la tira de LEDs

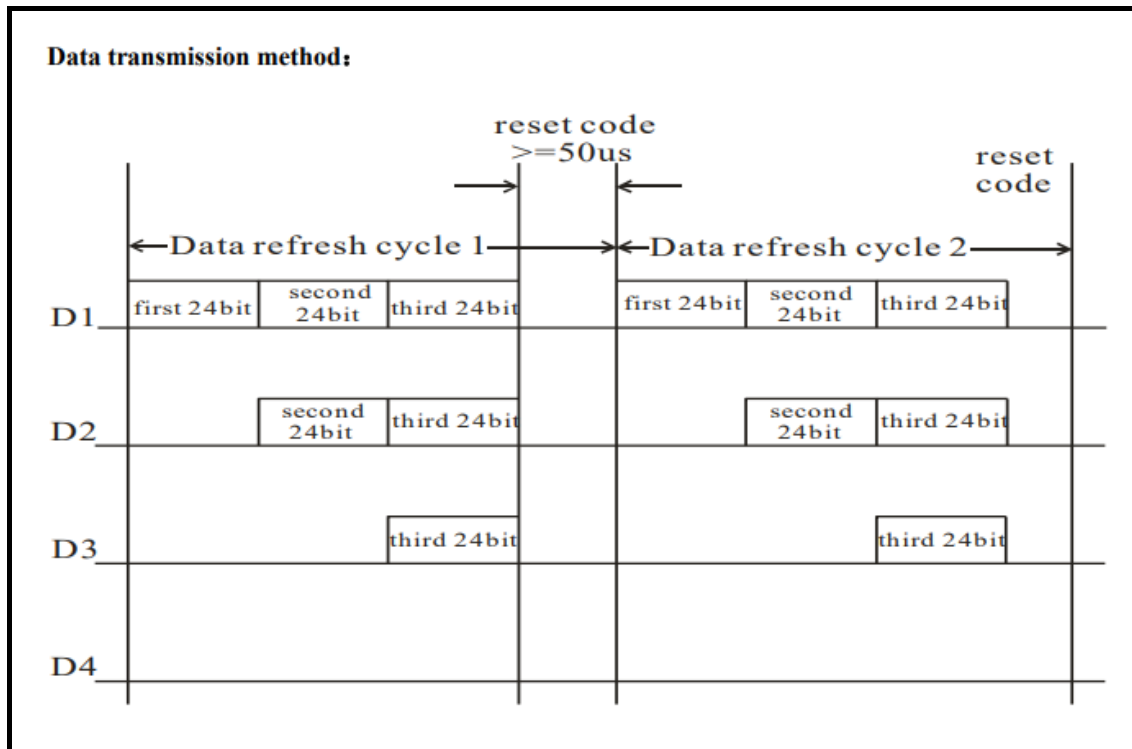
El módulo controlador de la tira de LEDs gestiona la iluminación y los efectos visuales del sable láser. Puede controlar el color, la intensidad y los patrones de iluminación de los LEDs. Además, se plantea utilizar código ensamblador en el siguiente avance, debido a que el protocolo para cargar valores de la tira de LEDs requiere una temporización muy particular que está en el orden de los cientos de nanosegundos.

##### 5.4.1. Protocolo de la tira de LEDs

La carga de valores en la tira de LEDs se realiza de la siguiente forma:

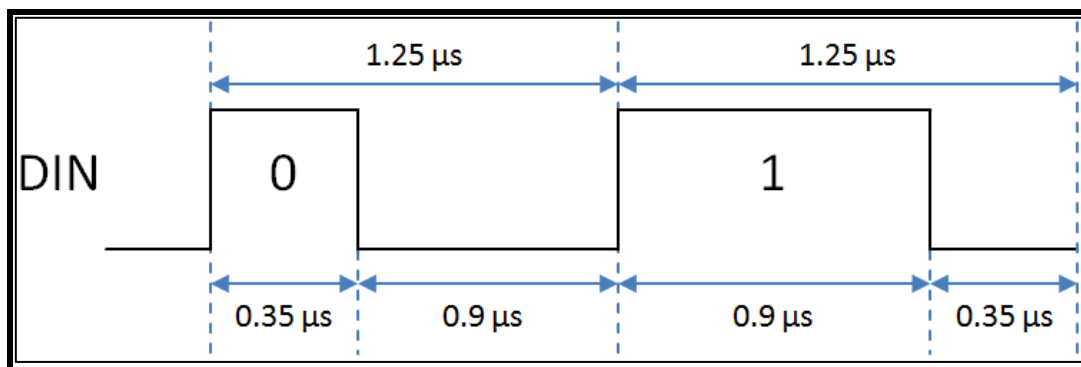
Se envía un paquete binario en orden GRB, con 8 bits para cada componente de color, lo que resulta en un total de 24 bits por paquete de color. El primer paquete se envía al primer LED en el inicio de la tira. Al enviar el siguiente paquete de color, el sistema omite el LED ya activado y se dirige al siguiente. Esta técnica permite configurar un color específico para cada uno de los píxeles usando un solo cable, sin un límite aparente de cantidad de LEDs.

La velocidad de transmisión es de  $1.25 \mu\text{s}$  por bit, lo que se traduce en un retardo de 30 microsegundos por color/LED. Esto permite asignar 33333 bits por segundo, lo que desprecia el tiempo de standby que la información debe permanecer en cada LED para ser visualizada correctamente. Esta característica impone la mayor restricción de la tecnología, limitando el número máximo recomendado de LEDs a un estándar de 1024 por tira, aunque esta cantidad supera ampliamente las necesidades de este proyecto.



*Figura 5.2 Descripción de paquetes de transmisión para la tira de LEDs*

La Figura 5.2 presenta un diagrama temporal que ilustra cuándo cada diodo (D1, D2, D3) recibe un determinado "paquete de color". En este ejemplo, se muestra la transmisión de datos para tres LEDs inteligentes. Durante el "Ciclo de refresco de datos 1", se envían paquetes de 24 bits que determinan la combinación de color única para cada LED. Tras esta primera transmisión, se establece un intervalo para que los LEDs puedan mostrar el color que se les ha asignado. Es esencial que este intervalo dure al menos 50 microsegundos para garantizar que los LEDs reflejen el color de manera adecuada.



*Figura 5.3 Descripción de transmisión de bits para la tira de LEDs*

Cada bit de un paquete se transmite de forma individual, y su codificación se basa en el "ancho de pulso". La transmisión de todos los bits se realiza en un ciclo que comprende dos flancos. La duración de cada bit, que es constante a lo largo de toda la comunicación, está especificada por el fabricante en 1.25 microsegundos. La distancia entre estos flancos determina si se está transmitiendo un 0 lógico o un 1 lógico. Aunque los tiempos de separación establecidos cuentan con una tolerancia de 0.125 microsegundos, para garantizar una detección precisa de cada bit, se adopta el margen que ofrece la mayor distinción entre los bits.

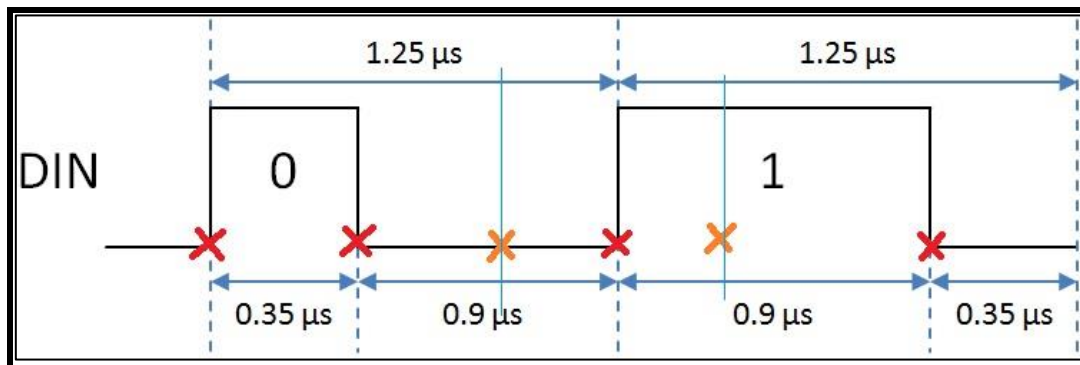


Figura 5.4 Descripción de transmisión de bits para la tira de LEDs

Es importante destacar que las escrituras de salida de datos se distribuyen de manera periódica. El flanco ascendente siempre debe sincronizarse con el flanco anterior cada 1.25us. Por otro lado, el flanco descendente se sitúa a 0.35us o 0.9us del flanco ascendente, dependiendo de si se desea transmitir un '0' o un '1'. Esta estructura facilita la implementación de futuras actualizaciones, centrándose únicamente en estos intervalos específicos, que están señalados con cruces rojas o naranjas en la Figura 5.4.

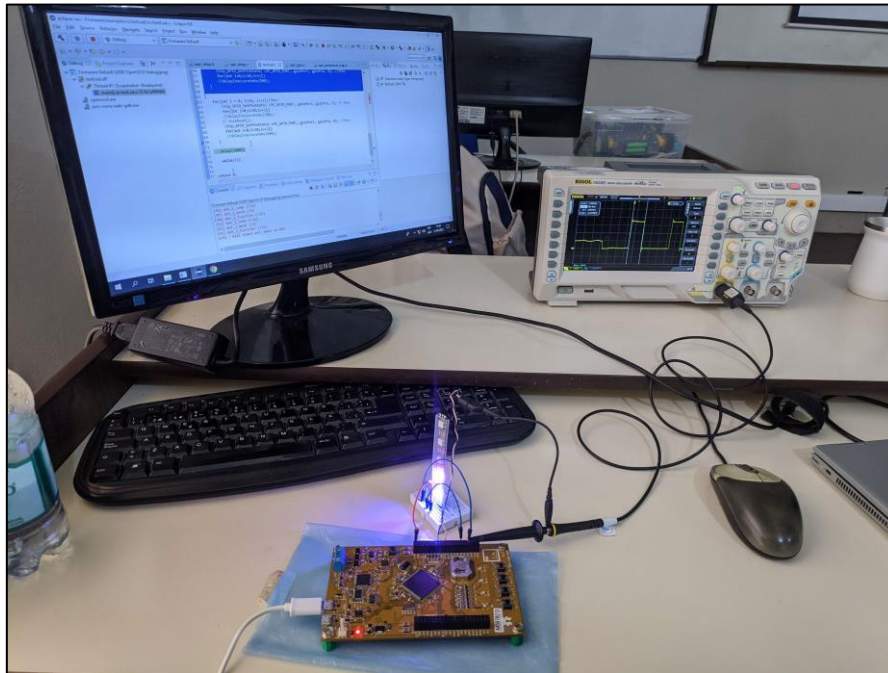
#### 5.4.2. Solución para el protocolo de transmisión de datos

Para abordar este desafío, como se observó, necesitamos gestionar intervalos de tiempo extremadamente precisos, en el rango de fracciones de microsegundos, o, en otras palabras, cientos de nanosegundos. A lo largo de estas semanas, exploramos diversas alternativas para llevar a cabo esta tarea específica.

Finalmente, la implementación específica del protocolo de comunicación Neopixel fue proporcionada por la cátedra y posteriormente optimizada. La base principal de esta implementación se basa en una interrupción periódica del timer0 cada 1200 ns, dedicada a la generación de onda Neopixel. Cuando la variable `Update` es diferente de cero, se comienzan a enviar los bits decodificados uno a uno, generando el flanco ascendente al inicio de la interrupción y el flanco descendente al final. La información del bit transmitido se encuentra en la duración del ciclo alto. Durante esta interrupción, se analiza el bit correspondiente a



transmitir, segmentando a qué byte y número de bit se debe transmitir. Una vez obtenido el byte a analizar, se accede a él y se utiliza un conjunto de máscaras para obtener el estado del bit específico.



*Figura 5.5 Día de pruebas de tira de LEDs*

#### 5.4.3. Efectos de la tira de LEDs

##### **A. Capa Base - Efecto Onda**

Los efectos aplicados a la tira LED Neopixel consisten en una superposición de capas, específicamente una capa base de "efecto onda". En cada iteración, se calcula esta capa base, alterando la intensidad del color base del sable. El color base, actualmente constante para toda la ejecución, es (0,160,160), y se prevé la posibilidad de cambiarlo en futuras versiones. La alteración se realiza mediante operaciones seno desplazadas, dependientes del ciclo de ejecución y del número de píxeles, generando una onda semoidal que se dispersa a través del sable y se desplaza en cada ciclo de ejecución.

##### **B. Capa de Choque**

La segunda capa de efectos corresponde al "efecto choque". La detección de colisiones se realiza a través de la lectura analógica de sensores táctiles TTP, cuyos valores llegan en forma analógica a la CIAA a través del puerto CHAN1. En la función `Detectar_colisiones`, se lee la señal analógica y se decodifican los estados de los 4 táctiles, almacenando el estado de cada sensor en flags globales. Estos flags se verifican periódicamente por la segunda capa de efectos "efecto de choque", que genera un difuminado con un color aleatorio en el lugar específico correspondiente al flag del touch activado. Este efecto de difuminado se realiza mediante una



función de mezcla de colores ponderados, dando más peso al color del choque más cerca del centro.

### **C. Implementación del Efecto Onda Sinusoidal**

La función `efect_sinoidal_breath_c(struct color c)` recibe como parámetro el color base que se utilizará para todos los píxeles y asigna una proporción (entre 0 y 1) de ese color a cada píxel de la tira LED. La proporción asignada del color base a cada píxel se interpreta como la "intensidad" del color base en función del tiempo. A cada píxel se le asigna un color como el producto del color base y la intensidad correspondiente, es decir, `ColorPixel_i = ColorBase * Senoide(alpha)`. La operación "senoide" se trunca de manera lineal para tener un valor de 0 a 1, y "alpha" es un ángulo que se incrementa proporcionalmente al número de píxel y al tiempo de ejecución.

En la Figura 5.6 se presenta un resumen general en un diagrama de flujo de las funciones que intervienen en la generación de los efectos. En la rama izquierda se encuentra el bucle principal que genera el efecto "`efect_sinoidal_breath_c(color_base)`", lee el estado de los sensores táctiles a través del ADC, deposita esta información en banderas globales (similar a un nano despachador), y luego añade el efecto choque al lugar correspondiente si la bandera está habilitada. Una vez que se ha definido el color final que se debe mostrar, se activa `Update=ON`, habilitando el comportamiento de la generación de la señal Neopixel. Hasta que no se termine de transmitir esta señal, el programa principal queda bloqueado en el `while(update==ON)`.

En el lado derecho de la misma figura, se observa cómo, concurrentemente a la ejecución del bucle principal, cada 1200 ns se activa la interrupción periódica que verifica si `Update==ON`. En caso de activarse la señal de actualizar la tira LED, entra en ejecución la lógica de generación de señal Neopixel, con sus flancos ascendentes y descendentes previamente comentados. Al final de la transmisión de cada bit, se corrobora si ya se terminó de transmitir toda la trama completa de bits de la tira LED. En caso de terminar de generar todos los bits, se deshabilita el flag compartido `Update=OFF`.



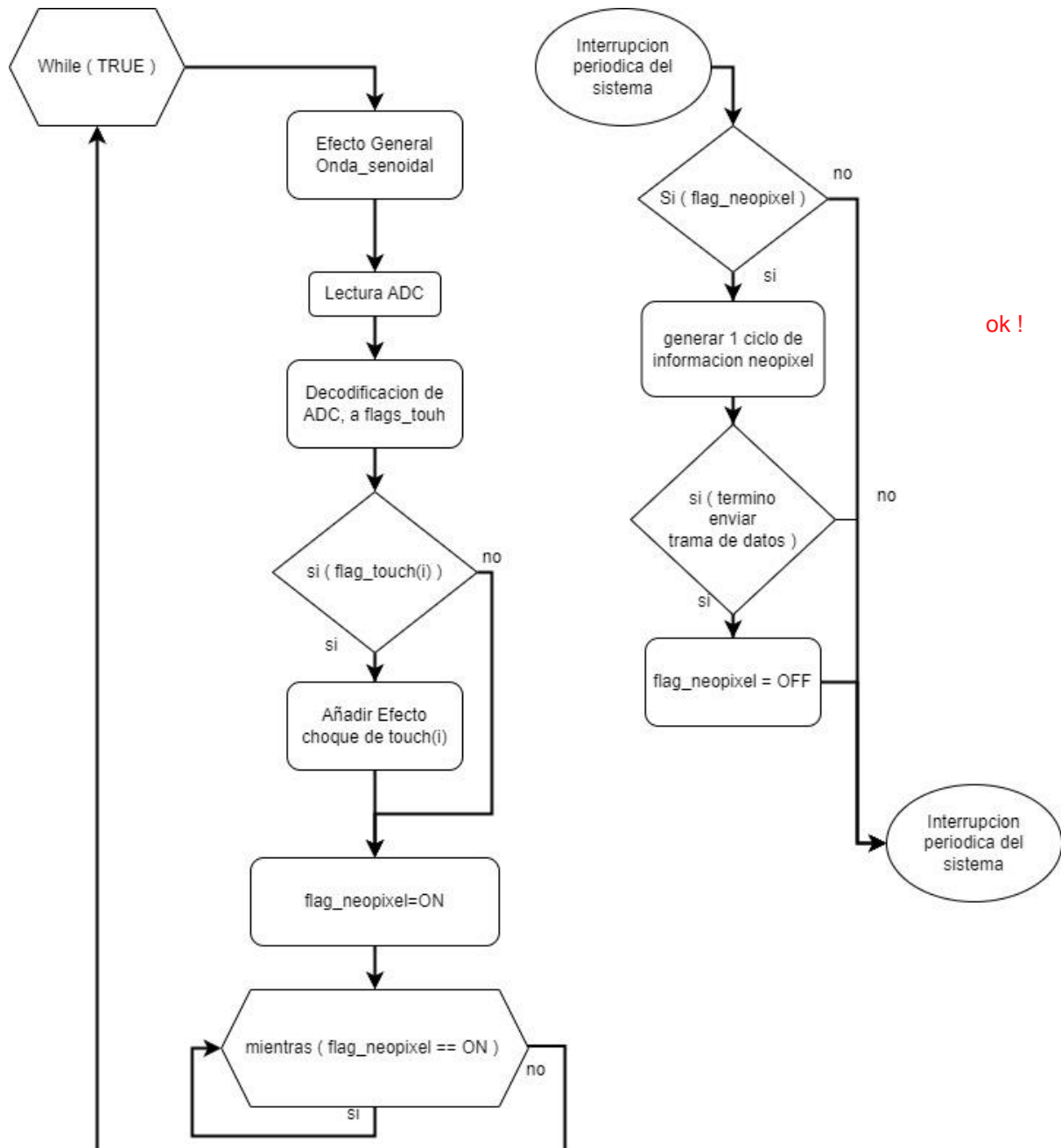


Figura 5.6 Diagrama de flujos de generación de efectos de tira de LEDs

### 5.5. Módulo controlador de sensores touch

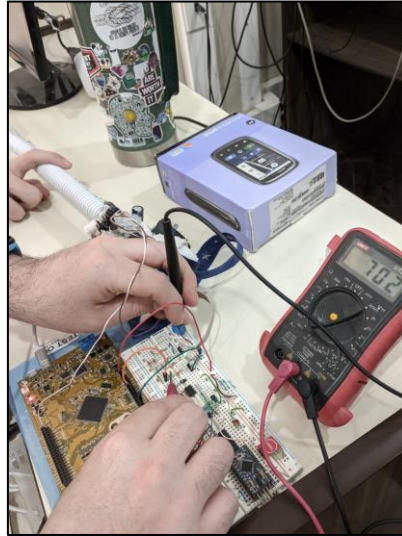
Este módulo se encarga de detectar y procesar las interacciones táctiles del usuario con el sable. Estas interacciones táctiles se usan para realizar variaciones de efectos de luz y sonido. Se utilizará la biblioteca de sAPI para realizar lecturas con el ADC.

Dentro del módulo, se disponen de 3 variables flag de tipo uint8\_t, cada una hace referencia a uno de los sensores touch. Indican si se detectó una lectura.

Se posee también una variable que almacena la lectura del ADC. Tal cómo se mencionó en la sección 4.5, el bloque DAC R2R se encarga de generar la señal que será leída por el ADC. Y,



utilizando 3 variables de trigger value para cada touch se los puede distinguir, poniendo en comparación el valor de trigger con el valor leído por el ADC. Estos valores de trigger se determinaron de forma experimental.



*Figura 5.7 Midiendo los valores de trigger de los sensores touch*

*Variables de 8 bit: flag\_touch\_1, 2 y 3 = 0*

*Variables de 16 bit: lectura\_adc = 0, trigger\_value\_1, 2 y 3*

*Iniciar\_ADC()*

*Mientras (true)*

*Iniciar\_conversion\_ADC()*

*Mientras no se complete la conversión*

*Esperar*

*Leer\_valor\_ADC()*

*Verificar\_flags\_touch()*

*Pseudocódigo 2 Programa para probar los sensores touch*

Con un programa similar al mostrado en el Pseudocódigo 2, se realizaron las pruebas de los sensores touch, en el momento que se toca un sensor (o varios a la vez), se realiza un efecto en la tira de LEDs donde se enciende con efecto de choque, la sección que abarca el sensor touch tocado.

#### 5.6. Módulo controlador de acelerómetro

El módulo controlador de acelerómetro es esencial para detectar los movimientos y orientación del sable láser. Puede identificar acciones como giros, sacudidas o movimientos rápidos. Esta información se utiliza para sincronizar efectos visuales y sonoros, es decir, se utilizará para modular las señales de sonido y generar cambios de intensidad en los LEDs.



Se utilizó la biblioteca de comunicación I2C para la lectura de dos de los ejes del giroscopio que provee el MPU6050 [5].

### 5.7. Módulo controlador de sonido

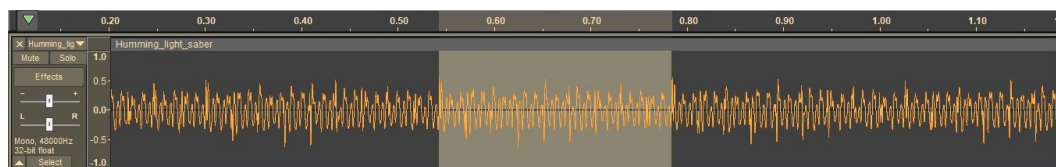
Este módulo gestiona la generación y reproducción de sonidos en el sable láser. Podrá producir sonidos de encendido, apagado, choque, movimiento y otros efectos sonoros característicos de un sable láser. La sincronización precisa con los otros módulos asegura que los sonidos se reproduzcan en el momento adecuado, mejorando la inmersión del usuario. Se utilizará al DAC para la generación y modulación de estas señales de sonido.

#### 5.7.1. Generación del archivo fuente de sonido

Se parte de un archivo de audio descargado de internet, que contiene el sonido de zumbido del sable laser, “Humming\_light\_saber.mp3”. Mediante el software Audacity, se procede a hacer las adecuaciones pertinentes que hacen a la idea del proyecto de sable laser:

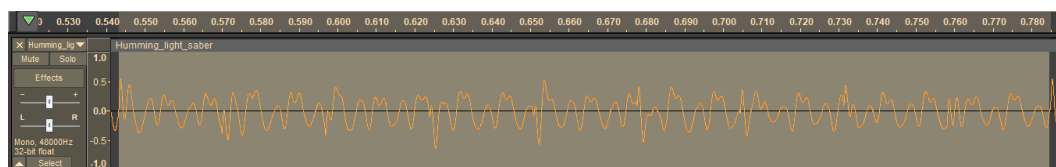
1-Dado que solo se utiliza un canal para reproducir el sonido, puesto que no se utiliza ningún efecto de holofonía, se elimina una de las dos pistas mediante: Tracks >> Mix >> Mix Stereo Down to Mono. De esta forma, se pasa de un sonido estéreo a mono.

2-Basta con elegir una parte de sonido representativa, que permita reproducir en bucle y de la sensación del zumbido del sable laser. Para ello, se toma una porción en la que aproximadamente se observe un período, ver Figura 5.8.



*Figura 5.8 Porción representativa de sonido en Audacity*

En la Figura 5.9 se puede ver que la selección es aproximadamente periódica, con lo cual, podría servir para reproducir el sonido. Esta selección se puede refinar aún más, haciendo zoom y tomando una muestra cercana a 0 desde que se observa el primer pico a la izquierda, hasta otra muestra cercana a 0 cuando se ve nuevamente el pico a la derecha, ver Figura 5.7.

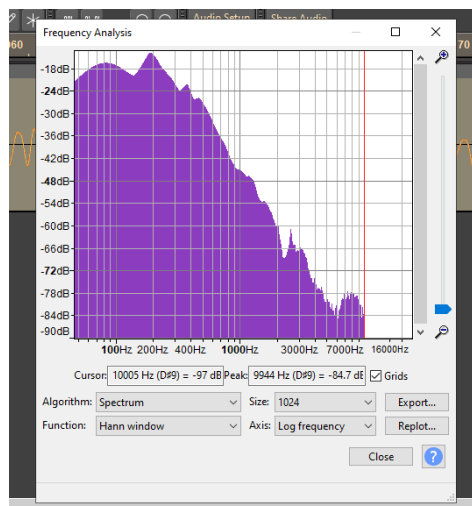


*Figura 5.9 Porción representativa de sonido 2 en Audacity*

Aquí se puede ver como el comienzo desde 0 hacia el primer pico positivo es solidario respecto del final, donde la señal se eleva desde un valor negativo hacia 0. Esto produce una

transición suave y evita que audiblemente se note una diferencia entre el principio y el final, algo de suma importancia teniendo en cuenta que el sonido se reproducirá en bucle.

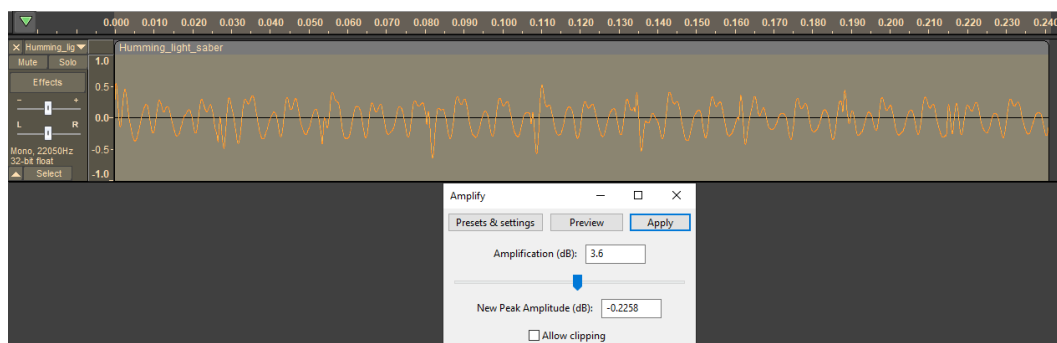
3-Teniendo el recorte representativo ya listo, se puede remuestrear la señal para que el archivo que contiene los datos sea de menor tamaño, algo crucial para sistemas embebidos, ya que se tiene una cantidad reducida de RAM. Para ello, se puede ver el espectro del recorte con la funcionalidad Analyze >> Plot Spectrum:



*Figura 5.10 Espectro del recorte en Audacity*

En la captura se puede ver que la señal llega a los 10kHz, con lo cual, teniendo en cuenta que el teorema del muestreo indica que la frecuencia de muestreo debe ser el doble de la frecuencia fundamental, se puede concluir que con 20kHz alcanza para reconstruir completamente la señal. Ahora bien, mediante la funcionalidad Tracks >> Resample, Audacity permite remuestrear a 22050Hz.

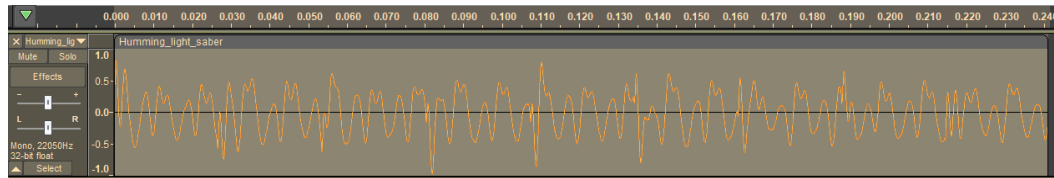
4-En cuanto a la amplitud, se puede observar que los picos de la señal no llegan al máximo del rango de representación. Por lo tanto, para aprovechar mejor la señal, se puede amplificar la misma con Effects >> Volume and Compression >> Amplify:



*Figura 5.11 Amplificando la señal con Audacity*



Un detalle importante al amplificar la señal es evitar el clipping o recorte, lo que implica la pérdida de datos por exceder el límite del rango de representación. Para tal fin, se desplaza el selector de tal forma que New Peak Amplitude sea el valor negativo más cercano a 0, situación en la cual Amplification es el valor máximo de amplificación sin pérdida de datos:



*Figura 5.12 Señal amplificada en Audacity*

Efectivamente se aprecia que la señal se amplificó sin que se recorten los picos, es decir, sin pérdida de datos.


5-Finalmente, se debe exportar el audio editado en un formato apropiado. Primero se debe configurar la tasa de muestreo del proyecto, a través de Edit >> Preferences >> Project Sample Rate, para que coincida con la frecuencia de muestreo elegida (22050Hz).

Luego, se procede con Export >> Export Audio, se elige Other uncompressed files, formato RAW (Header-less), codificación Unsigned 8 bit PCM.

El formato RAW, como su nombre lo indica, no posee cabeceras que contengan información de la señal, son simplemente series de bits, unos tras otros, que representan los tonos del audio; por otro lado, la codificación de 8 bits sin signo quiere decir que cada tono se representa con un byte, cuyo rango está dado de 0 a 255.

Se eligió un formato sin cabeceras ya que los datos serán escritos directamente al DAC, a la misma frecuencia de muestreo, por lo que no se necesita información adicional. En cuanto a la codificación, es de interés poder modular la amplitud de la señal, conforme a los valores del acelerómetro, entonces, con 8 bits es posible cuadruplicar la amplitud sin perder información, considerando el máximo establecido por el DAC de 1023 ( $255 * 4 = 1020$ ).

Adicionalmente, las pruebas en clase tuvieron un buen resultado, y el sonido suena bien, evidentemente una codificación de 8 bits, si bien implica una gran reducción en el rango de representación, sirve para reproducirlo con una calidad apropiada al presente proyecto.



### 5.7.2. Obtención de los valores hexadecimales en crudo

Para extraer las muestras o tonos del archivo .raw generado en el apartado anterior, se utiliza un programa escrito en C con el siguiente pseudocódigo:

*Abre archivo fuente como tipo binario de lectura*

*Crea archivo destino de tipo texto plano de escritura*

*Obtiene la cantidad de bytes del archivo fuente*

*Rebobina el archivo fuente*

*Por cada byte del archivo fuente - 1*

*Lee 1 byte*

*Lo almacena en el archivo destino con un formato legible*

*Agrega el último con la terminación apropiada*

*Cierra ambos archivos*

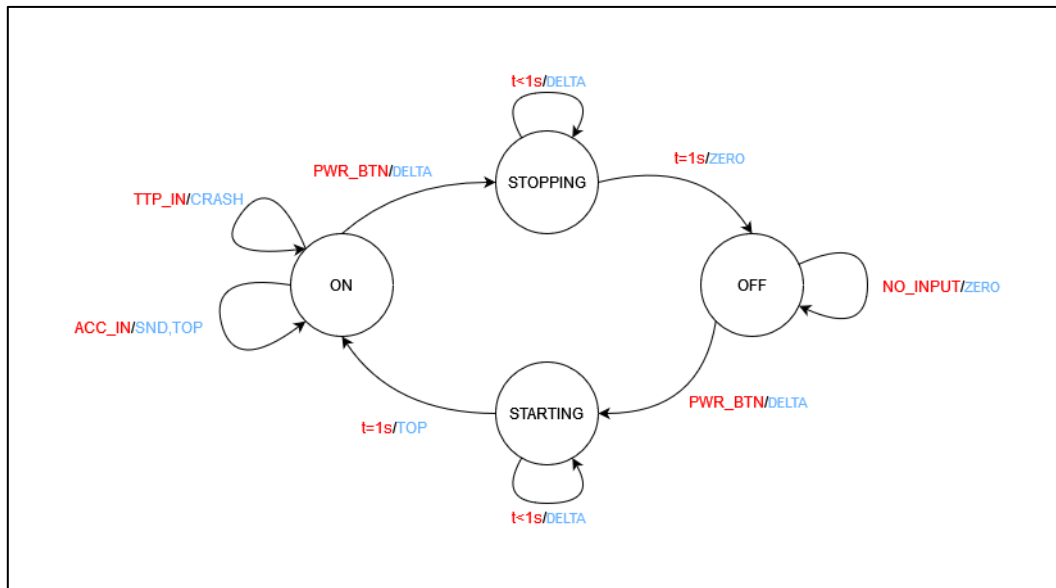
*Pseudocódigo 3 Conversión del raw a valores hexadecimales formateados*

Este programa toma como archivo fuente el .raw colocado en la carpeta del proyecto C, y crea un archivo .txt como destino de valores hexadecimales separados por coma y espacio con el siguiente formato: 0xYZ, siendo Y el nibble alto y Z el nibble bajo.

Los valores hexadecimales se copiarán a un arreglo en memoria para su posterior escritura en el DAC. Esto se tiene como pre-condición, la copia es manual y no se realiza en tiempo de ejecución.

### 5.8. Estados del sable laser

El sable laser tiene 4 estados, la forma de transición es secuencial como se puede observar en la Figura 5.6.



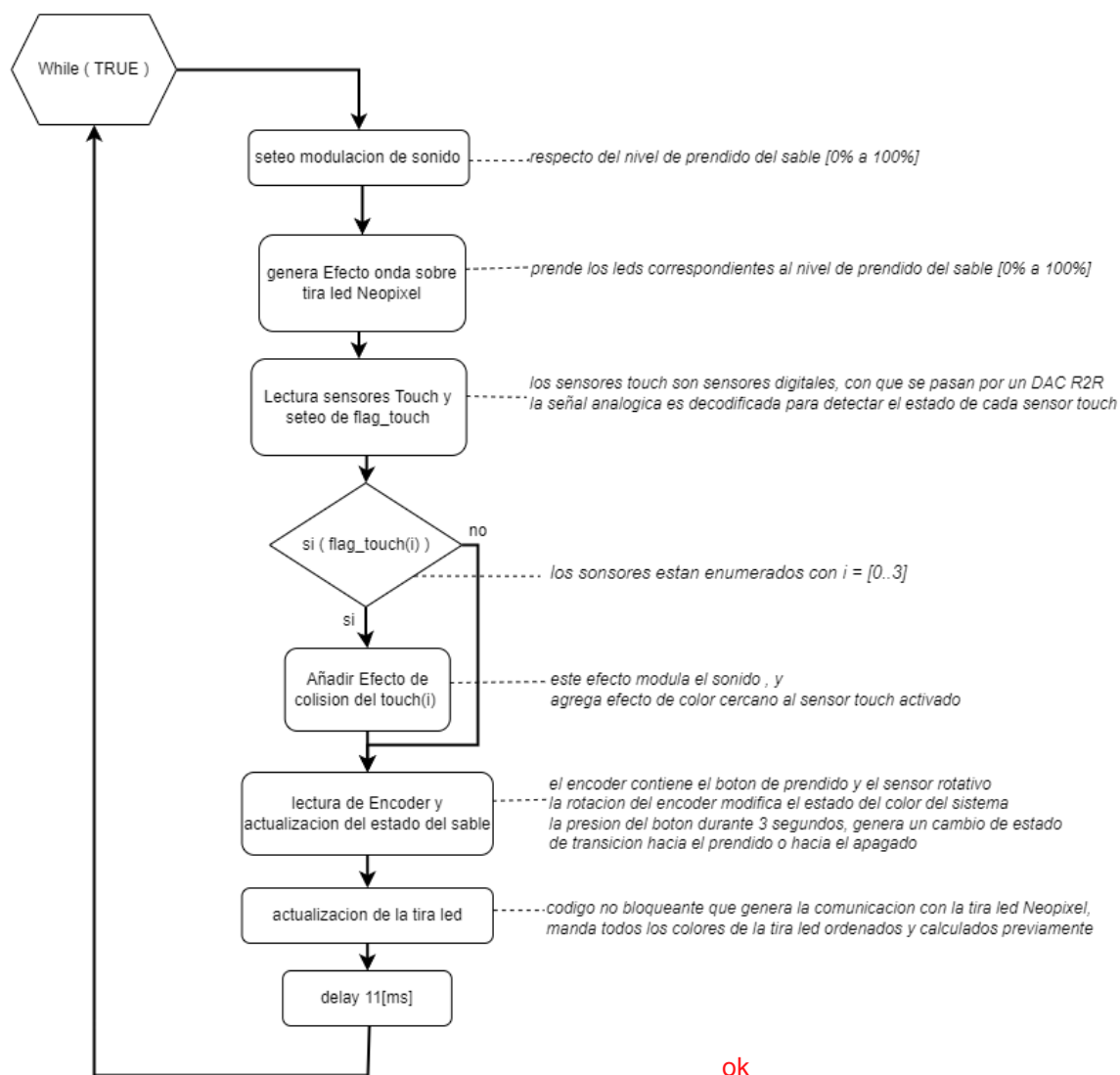
*Figura 5.13 Diagrama de estados del sable laser*

- **OFF:** es el estado por defecto del sistema; estado de apagado del sable
- **STARTING:** estado transitorio entre **OFF** y **ON**; Al presionarse el botón de encendido, se transiciona a este estado y comienza a darse una salida **DELTA**, esta salida se encarga de realizar un encendido incremental en el sentido de que se van encendiendo progresivamente los LEDs y el sonido se va modulando para acompañar de forma acorde y fiel a la película. Al terminar de encenderse (**TOP**) se transiciona al estado **ON**
- **ON:** estado de encendido del sable; en este estado se realizan las lecturas del acelerómetro (**ACC\_IN**) que producen las salidas **SND** (modulación de sonido) con una intensidad máxima de LEDs (**TOP**). Se dan lecturas de los sensores touch (**TTP\_IN**), estos pueden indicar una colisión (**CRASH**), modulan el sonido de forma tal que se escuche como una “colisión” y se enciende con color blanco la sección de LEDs donde se midió con el sensor/sensores touch. Si se presiona el botón se pasa al estado **STOPPING** donde se comienza a realizar un **DELTA** inverso al de encendido
- **STOPPING:** estado transitorio entre **ON** y **OFF**; en este estado se hace un **DELTA** de apagado (disminuye el sonido y se apagan progresivamente los LEDs), una vez apagado (**LOW**) se transiciona al estado **OFF**

hasta acá igual al informe anterior

## 5.9. Resumen del software

A continuación, se muestra gráficamente cómo funciona el flujo del programa final que maneja todo el sistema luego de haber inicializado los respectivos componentes (Tira LED, DAC, etc.):



## 6. Ensayos y mediciones

### 6.1. PCB

Nuestra PCB es un tanto peculiar, ya que debido a la distribución de componentes por la que se optó se tienen en realidad dos PCBs unidas por un cable de red flexible. Por lo que ni bien tuvimos las placas listas y soldadas nos pusimos a probar continuidad, ya



sea en las mismas pistas de la placa como en la conexión de una PCB a la otra, los cuales nos mostraron que efectivamente las conexiones estaban correctas.

## 6.2. Construcción de las PCB

Teniendo los diseños de ambas placas listas con el cobre, se procedió a soldar los componentes.

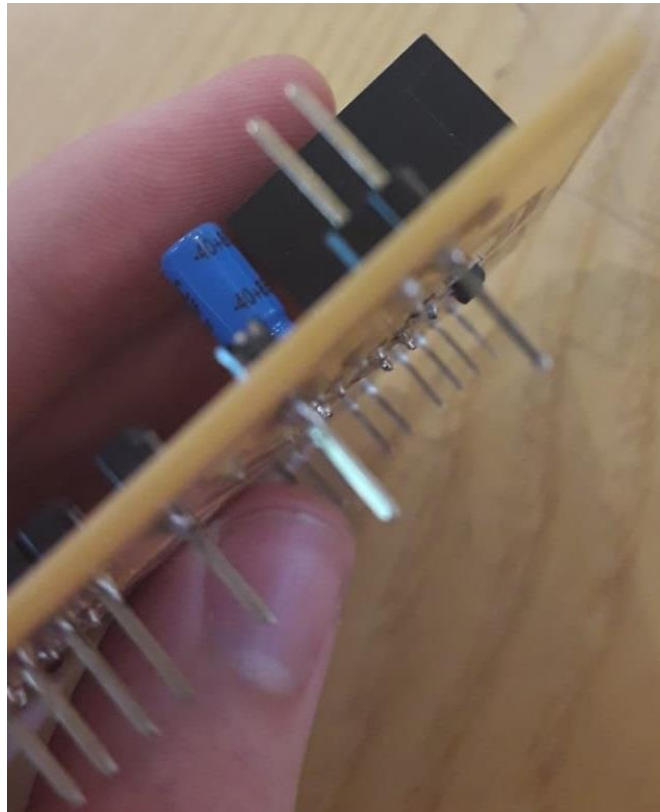
En general no hubo demasiadas complicaciones a la hora de soldar, el momento con mayor dificultad técnica fue cuando tuvimos que soldar las resistencias SMD en la PCB. Si bien se hace complicado debido a su minúsculo tamaño, logramos soldar todas siguiendo los consejos de la cátedra y ayudándonos con una brusela para sostener el componente mientras se soldaba a la placa.

Genial!

El resultado final se puede apreciar en el capítulo 9.3.

En cuanto a la PCB más grande tuvimos un percance con la alimentación del sistema. En un principio pensamos que el poncho podía ser alimentado por el voltaje de los pines de entrada/salida de la CIAA. En realidad, la alimentación del poncho debe hacerse a través de las borneras que tiene la CIAA específicamente para este trabajo. Junto con la ayuda de la cátedra, llegamos a la solución de este problema. Habiendo ya soldado todos los pines en la placa, sacamos los pines que entraban en los puertos de voltaje y tierra y los invertimos como se puede ver en la siguiente imagen para que queden hacia arriba en vez de hacia abajo.

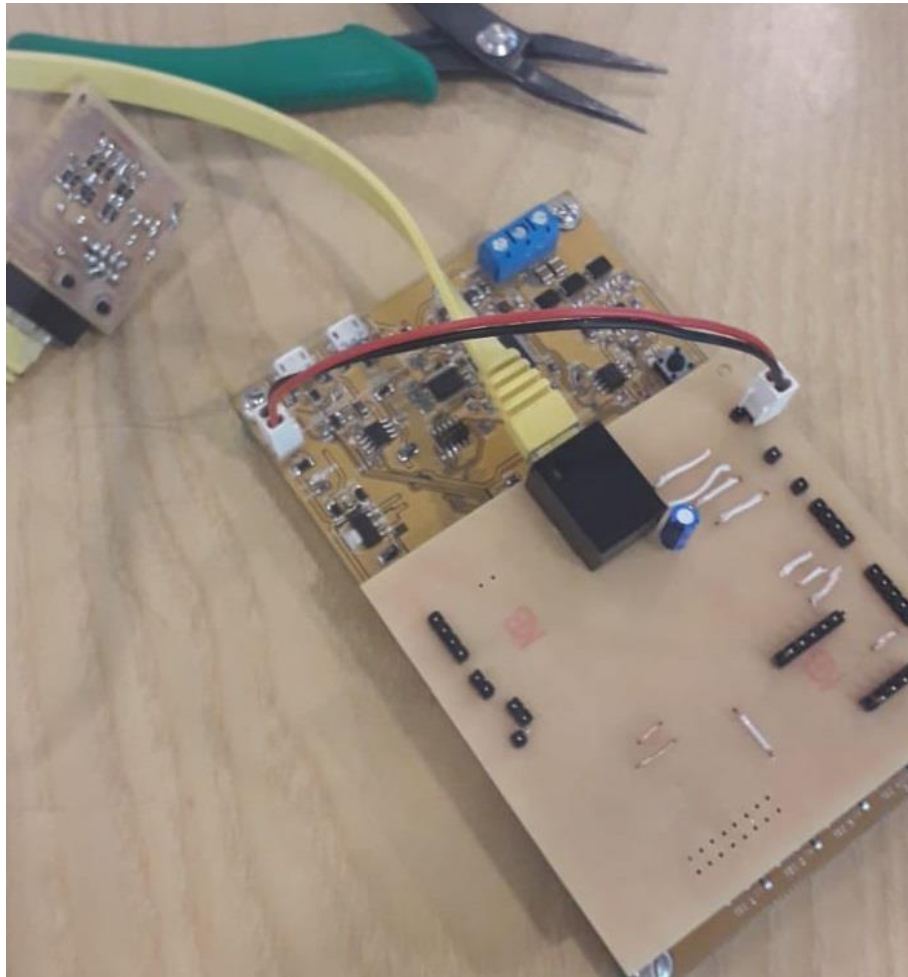




muy bueno

Con esta modificación pudimos simplemente usar dos cables que conectaran directamente a la bornera.

En la siguiente imagen se puede ver el poncho conectado a la EDU-CIAA como así también la segunda PCB más pequeña conectada al poncho con el cable de red.



buen trabajo

Una vez terminadas estas modificaciones se realizaron pruebas sencillas para comprobar la alimentación correcta del sistema a través de las baterías las cuales fueron exitosas.

### 6.3.Integración de periféricos

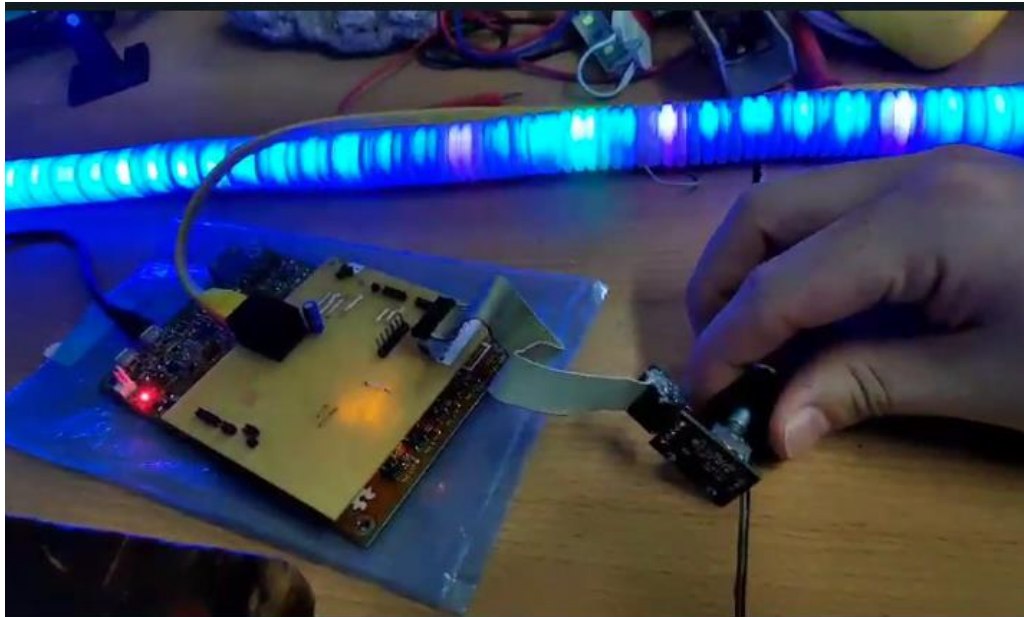
En la recta final del proyecto llegamos con casi todos los componentes del sistema funcionando como queríamos, con la diferencia de que todos funcionaban y habían sido probados por separado exceptuando la tira led y los sensores touch que si funcionaban en conjunto.

A la hora de poner todos los componentes juntos como cabía esperar surgieron problemas.

Al tener unos tiempos de envío de datos muy precisos por la forma en que hay que tratar con la tira LED, una forma muy evidente de saber cuando algo interfiere con la tira de alguna forma es simplemente ver como se prende.



Cuando integramos el sonido al sistema de LED + TOUCH “estable” descubrimos estos errores visibles en numerosas ocasiones.



En la imagen se puede apreciar como la mayoría del sable se encuentra en un tono azul lo cual es correcto, sin embargo, hay puntos específicos donde el color esta un tanto rojo o verde que son los errores antes mencionados.

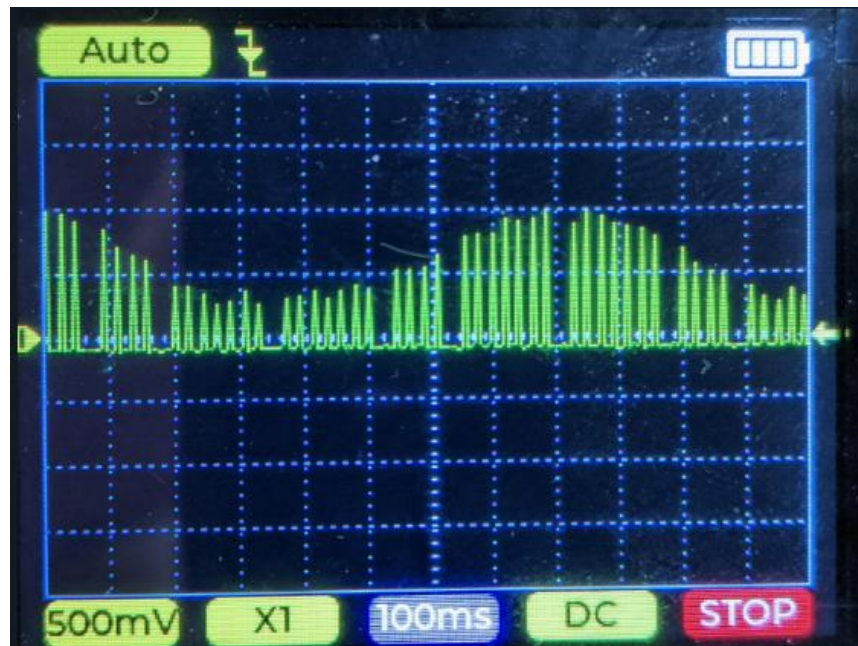
Descubrimos que dichos errores eran ocasionados porque al hacer uso del puerto DAC para el sonido, este a su vez utiliza el timer 0 de la CIAA, el cual nosotros ocupamos exclusivamente para la tira LED.

Si bien la lectura DAC y el refresco de los LEDS ocurren en lugares y tiempos distintos a nivel código, de alguna forma el conversor estaba interfiriendo en la transmisión de datos de la tira.

Nuestra solución constó en activar el DAC solo cuando vaya a reproducir una porción del audio en el ciclo actual e inmediatamente después desactivarlo, jugando con el hecho de que la tira LED tiene una tasa de refresco mucho más alta que la del DAC.

Si bien midiéndolo con un osciloscopio es notorio el cambio, la naturaleza del sonido que reproducimos y la velocidad de los intervalos hace que sea casi imperceptible.

A continuación, se puede ver en la lectura del DAC en un osciloscopio como la suma de muchos picos de valores forman algo parecido a una onda sinusoidal.



Esta muestra sin embargo resulto ser demasiado lenta para nuestras necesidades, de hecho, el sonido generado era 100 veces más lento.

Para agilizar la emisión en cada interrupción se habilita y se llama a emitir al DAC haciendo que la onda se asemeje aún más a un sonido sin interrupciones. En la siguiente captura, y en el apartado de videos en la bibliografía, se puede apreciar la emisión de audio en su versión final:



ok





Por otro lado, contábamos también con la MPU para leer valores y modificar el comportamiento de las luces y el audio respecto a los valores leídos.

Habíamos probado el componente y teníamos escrita una librería que consistía a grandes rasgos en tomar algunos de los valores de la MPU, sumarlos y devolver un porcentaje, el cual iba a ser utilizado como múltiplo para la modulación del sonido. Al momento de integrar librerías y código nos topamos con que la comunicación I2C que utiliza la MPU funciona con interrupciones. Si bien no estamos completamente seguros, debido a que por falta de tiempo no pudimos ahondar en este error en particular, coincidimos en que esto hace que en medio de un refresco de la tira LED se pueda interrumpir lo que hace que los valores actuales de los colores se corrompan provocando un efecto de parpadeo con colores aleatorios similar al que se describía anteriormente con el DAC.

Lamentablemente tuvimos que descartar la integración de la MPU el producto final, luego de algunas pruebas debido a los escasos de tiempo para terminar.

Finalmente, el ultimo problema de integración que tuvimos fue con los sensores touch.

Al tener todo el sistema integrado junto empezamos a notar como los efectos de los sensores se activaban presuntamente sin motivo aparente. Pudimos leer la señal con un osciloscopio y pudimos ver que parte del problema se encuentra en un pico de tensión en el canal cuando la EDU-CIAA lee del puerto <sup>ADC</sup> DAC. Esto aparentemente altera los valores lo suficiente como para que los touch se activen cuando no deberían. Una solución posible seria agregar un capacitor en el mango en la salida de 3.3v, lo cual reduciría dichos picos.

De nuevo, debido a la escasez de tiempo e itinerarios ajetreados de los integrantes del grupo, optamos por concluir el proyecto final con este error, el cual es visible en los videos grabados para la muestra, accesibles en el capitulo 9.

Este proyecto continuará...

## 7. Cronograma <sup>final</sup> preliminar

A continuación, se muestra el cronograma estimado del desarrollo del proyecto.

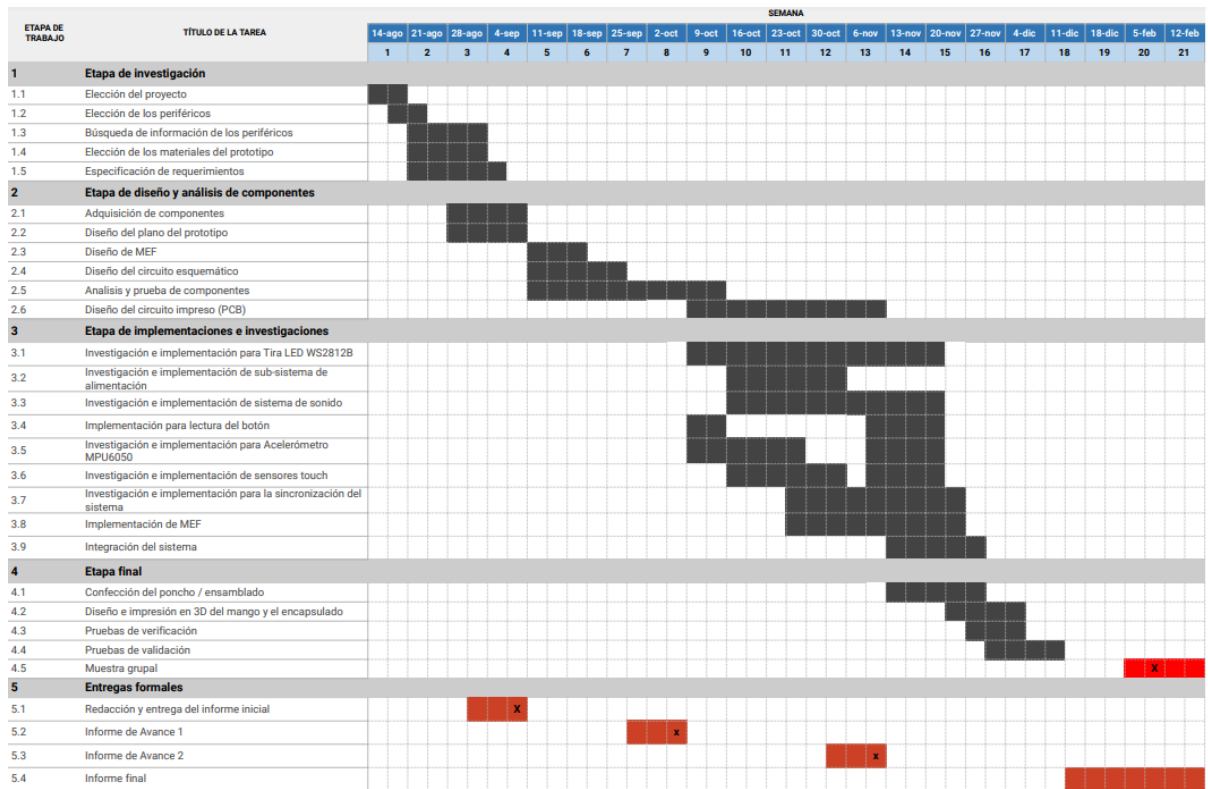


Figura 7.1 Diagrama de Gantt con los tiempos manejados del proyecto

claramente el punto 4 se realizó luego de la fabricación y ensamble de los pcb.

## 8. División de tareas del grupo

Tabla 8.1 División de las tareas del grupo

TAREA	RESPONSABLE	COLABORADOR
Elección del proyecto	Joaquín Matto, Juan Martín Ercoli	-
Elecciones de los periféricos	Nicolas Pacheco, Adrian Barral, Augusto Borromeo,	-



	Joaquín Matto, Juan Martín Ercoli	
<b>Búsqueda de información de los periféricos</b>	Adrian Barral, Nicolas Pacheco, Juan Martín Ercoli	-
<b>Elección de los materiales del prototipo</b>	Joaquín Matto, Adrian Barral	-
<b>Especificación de requerimientos</b>	Juan Martín Ercoli, Joaquín Matto, Nicolas Pacheco, Augusto Borromeo, Adrián Barral	-
<b>Adquisición de componentes</b>	Nicolas Pacheco	Joaquín Matto
<b>Diseño del plano del prototipo</b>	Augusto Borromeo	Nicolas Pacheco
<b>Diseño de MEF</b>	Adrián Barral	Augusto Borromeo, Joaquín Matto, Nicolas Pacheco
<b>Diseño del circuito esquemático</b>	Nicolas Pacheco, Adrián Barral	Juan Martín Ercoli, Augusto Borromeo
<b>Análisis y prueba de componentes</b>	Juan Martín Ercoli, Joaquín Matto, Nicolas Pacheco, Augusto Borromeo, Adrián Barral	-
<b>Diseño del circuito impreso (PCB)</b>	Nicolas Pacheco, Joaquín Matto	-
<b>Implementación para lectura del botón</b>	Joaquín Matto	Adrián Barral



<b>Investigación e implementación para Tira LED WS2812B</b>	Nicolas Pacheco, Joaquín Matto	Augusto Borromeo, Juan Martín Ercoli
<b>Investigación e implementación para Acelerómetro MPU6050</b>	Juan Martín Ercoli, Adrián Barral	Augusto Borromeo
<b>Investigación e implementación de sub-sistema de alimentación</b>	Adrián Barral	Nicolas Pacheco
<b>Investigación e implementación de sistema de sonido</b>	Joaquín Matto	Juan Martín Ercoli, Adrián Barral
<b>Investigación e implementación de sensores touch</b>	Joaquín Matto, Nicolas Pacheco	Adrián Barral
<b>Investigación e implementación para la sincronización del sistema</b>	Juan Martín Ercoli, Joaquín Matto, Nicolas Pacheco	Adrián Barral, Augusto Borromeo
<b>Implementación de MEF</b>	Augusto Borromeo, Adrián Barral	Juan Martín Ercoli, Joaquín Matto
<b>Integración del sistema</b>	Nicolas Pacheco	Juan Martín Ercoli, Joaquín Matto, Adrián Barral
<b>Confección del poncho / ensamblado</b>	Nicolas Pacheco	Augusto Borromeo, Juan Martín Ercoli, Adrián Barral, Joaquín Matto





<b>Diseño y fabricación del encapsulado</b>	Nicolas Pacheco	Augusto Borromeo, Juan Martín Ercoli
<b>Pruebas de verificación y validación</b>	Nicolas Pacheco, Joaquín Matto	Augusto Borromeo, Adrián Barral
<b>Armado de informes de avance</b>	Juan Martín Ercoli, Joaquín Matto, Adrián Barral	Nicolas Pacheco, Augusto Borromeo
<b>Armado de informe final</b>	Joaquín Matto, Nicolas Pacheco, Juan Martín Ercoli, Adrián Barral	-

<b>Integrante</b>	<b>Hs de trabajo promedio por semana</b>	<b>Hs totales de trabajo</b>
<b>Adrián Barral</b>	2.8	49
<b>Augusto Borromeo</b> ausente	2.0	31
<b>Juan Martín Ercoli</b>	3.0	47
<b>Joaquín Matto</b>	2.8	46
<b>Nicolas Pacheco</b>	3.5	53

La cantidad de horas extras invertidas x alumno más las horas presenciales son acordes al alcance del proyecto y la organización de la materia.  
Era pertinente entregar el proyecto hasta pudo completarse.

## 9. Bibliografía

- [1] Pernia, Eric (2019). Especificaciones EDU-CIAA-NXP v1.1. Disponible en: [https://github.com/epernia/firmware\\_v3/blob/master/documentation/CIAA\\_Boards/NXP\\_LPC4337/EDU-CIAA-NXP/EDU-CIAA-NXP%20v1.1%20Board%20-%202019-01-03%20v5r0.pdf](https://github.com/epernia/firmware_v3/blob/master/documentation/CIAA_Boards/NXP_LPC4337/EDU-CIAA-NXP/EDU-CIAA-NXP%20v1.1%20Board%20-%202019-01-03%20v5r0.pdf)
- [2] CIAA (2015). Dimensiones de la placa EDU-CIAA-NXP. Disponible en: <https://github.com/ciaa/Hardware/blob/master/PCB/EDU-NXP/Doc/PCB.pdf>



- [3] therpf.com. Dimensiones del sable de luz de Darth Vader. Disponible en: <https://pinterest.com/pin/449374869069170409/>
- [4] datasheets utilizadas. Disponible en: [https://drive.google.com/drive/folders/1xwCUE12hI3yfk16IMlvPfyeCa87qsa7O?usp=drive\\_link](https://drive.google.com/drive/folders/1xwCUE12hI3yfk16IMlvPfyeCa87qsa7O?usp=drive_link)
- [5] Pruebas sobre el MPU6050. Disponible en: [https://photos.google.com/share/AF1QipPF1EEo8qG7dZZNsB-3JCYZttvLHLYGVdpcVnYyoV95vlSJc1VXNyONoE1OWotu7w?pli=1&key=TDJia\\_kfIRkVFMm9zc0hoZnFyaDFtRmh2ankyBHJB](https://photos.google.com/share/AF1QipPF1EEo8qG7dZZNsB-3JCYZttvLHLYGVdpcVnYyoV95vlSJc1VXNyONoE1OWotu7w?pli=1&key=TDJia_kfIRkVFMm9zc0hoZnFyaDFtRmh2ankyBHJB)
- [6] Pruebas sobre el sub-sistema de alimentación y la tira de LEDs. Disponible en: <https://photos.google.com/share/AF1QipOGf0wvHxbiZt6McGEPJz1BAkk-8fxERXKOJdggzEKuPCF5a3NCshrCH9PJTtvByA?key=b3RoQIVtb1o1WkFqVVR4clJHekpqZmdCMV9JRvv3>
- [7] Videos de las etapas finales del proyecto, integración de componentes y producto final: <https://drive.google.com/drive/folders/1NVWANyicEaEPGVbLjPLstcAzH20hq82?usp=sharing>
- [8] Repositorio de GitHub <https://github.com/augustobor/Taller1-EDU-CIAA>

## **10. Guía rápida de uso del sable láser**

A continuación, se describe paso a paso como usar el sable laser:

- ✓ Una vez grabado el programa en la EDU-CIAA ya se puede cerrar la caja contenedora, asegurándose antes de que el cable de red esté correctamente conectado al poncho.
- ✓ Conecte el otro extremo del cable de red al mango del sable.
- ✓ En la parte inferior del mango se puede encontrar un botón de dos posiciones, al oprimirlo se permite la alimentación desde las baterías a todo el sistema.



- ✓ Mantener oprimido el botón encoder un segundo enciende o apaga las funciones del sable.
- ✓ Mientras el sable este encendido se puede girar la perilla del encoder para cambiar el color de la tira LED entre una selección preestablecida.
- ✓ Con las baterías completamente cargadas, el sistema cuenta con una vida útil de alrededor de una hora. De ser necesario al costado del mango del sable se encuentra el puerto de carga con un conector tipo micro usb.

GENIAL !

## **11. Conclusiones**

### **11.1. Cumplimiento de objetivos**

A continuación, enumeramos los objetivos planteados en el capítulo 2 y agregamos un breve comentario denotando el grado de cumplimiento y nuestra satisfacción al respecto.

#### **Objetivos primarios:**

- Lectura del botón: facilitar el encendido y apagado del sable - (Completado al 100%).
- Control del acelerómetro: detectar los movimientos del sable, modulando la señal para la generación de sonido - (Completado al 50%: el acelerómetro junto con el código para leer los datos y realizar la modulación se implementó, sin embargo, no se llegó a integrar satisfactoriamente al proyecto).
- Manejo de LEDs: controlar la iluminación del sable, ajustando la intensidad y el brillo según el movimiento - (Completado al 100%).
- Sincronizar el funcionamiento: es requerida una sincronización entre los componentes para lograr una buena experiencia. Los componentes deben obedecer un comportamiento coordinado que puede tomar el sistema del sable - (Completado al 100%).




75%: se logró sincronizar la mayoría de los componentes del sistema, con la excepción del acelerómetro y parte de los sensores touch).

- Implementación de un sistema de sonido: el agregado de este componente implica una complejidad extra a la hora de sincronizar con el resto de los componentes. Este objetivo no solo se centra en el agregado de hardware adecuado para emitir el sonido, sino también en el código donde la generación de sonidos debe responder al:
  - Encendido del sable – (Completado al 100%).
  - Apagado del sable – (Completado al 100%).
  - Movimiento del sable – (Completado al 50%, no integrado al sistema final).
  - Detectar un cese de movimiento abrupto – (Completado al 50%, no integrado al sistema final).
- Implementación de sensores táctiles en la hoja del sable: para lograr un efecto táctil en la hoja del sable se propone la incorporación de sensores táctiles en la superficie de la hoja. Estos sensores detectarán el contacto físico y podrán traducirlo en cambios en la iluminación, brindando así una experiencia más inmersiva – (Completado al 90%, integrados y funcionando con errores de lectura).
- Implementación de una batería recargable: se propone dotar al sable de luz de una batería autónoma que permita su funcionamiento sin necesidad de estar conectado a una fuente de alimentación externa. Además, se contemplará que sea recargable. (Completado al 100%).

Objetivos secundarios:    creo que no resta al proyecto el hecho de no haber avanzado con estos objetivos.

- Implementación de un módulo Bluetooth para cambio de colores con aplicación móvil.- (Sin implementar).
- Implementación de un módulo WI-FI para el control de los colores– (Sin implementar).
- Incorporar una memoria SD que permita almacenar una variedad de efectos sonoros. – (Sin implementar).
- Integrar un botón encoder que posibilite controles adicionales. Este componente ofrecerá una forma precisa y ergonómica de ajustar parámetros, como podría ser el cambio de color de los LEDs. – (Completado al 100%, terminamos eligiendo esta opción por sobre las demás).



## 11.2. Cumplimiento de requerimientos

### Requerimientos funcionales primarios

- ✓ **Encendido del sable:** cuando se oprime el botón por primera vez, y luego de mantenerlo presionado 1 segundo, se enciende el sable laser. **Se completó al 100%.**
- ✓ **Apagado del Sable:** cuando se oprime el botón estando encendido el sable laser, y luego de mantenerlo presionado 1 segundo, se apaga el sable. **Se completó al 100%.**
- ✓ **Emisión de Sonido y Luz Continua:** los LEDs se mantienen encendidos con su color correspondiente y el sonido se mantiene en su forma de onda luego de una transición de apagado a encendido y viceversa. **Se completó al 100%.**
- ✓ **Ajuste de Intensidad y Sonido con el Movimiento:** se desarrollaron las librerías de ajuste variable, pero al integrarlas al proyecto no se pudo hacer funcionar dichas funcionalidades debido a 3 problemas para las cuales no se alcanzó a intentar solucionar por cuestiones de tiempos:
  - 1- El MPU es demasiado lento en comparación a los demás periféricos.
  - 2- La interrupción del sonido, por lo tanto, interrumpe el procesamiento de la tira LED, provocando bugs en el color.
  - 3- El MPU es muy susceptible a las desconexiones, algo que se agrava en un sable laser, que su idea se basa en el movimiento.

**Se completó al 60%,** librería implementada, no integrada ni depurada. **seguiremos adelante tratando de solucionarlo**
- ✓ **Efectos al Encender o Apagar el Sable:** al encenderse el sable, se van iluminando progresivamente los LEDs desde el mango hasta la punta, acompañado por un sonido que varía en frecuencia y amplitud de zumbido hasta la forma de onda final del sonido que se obtuvo de Audacity en la entrega de informe anterior, dando la sensación de encendido gradual, al igual que los LEDs. De manera similar al encendido, al apagarse los LEDs dejan de iluminarse progresivamente desde la punta hasta el mango, y el sonido acompaña con una variación inversa de frecuencia y amplitud, dando la sensación de apagado gradual. **Se completó al 100%.**
- ✓ **Reacción al Tacto:** se desarrollaron las librerías, pero al integrarlas al proyecto no se comportaban como era esperado, ya que por la superficie de funcionamiento de los sensores táctiles y por el margen de error que no es el apropiado para este proyecto, los sensores tenían interferencias que perjudicaban su detección, siendo que se activaban cuando no debían hacerlo. **Se completó al 80%.**



### **Requerimientos funcionales secundarios**

- ✓ **Cambio de color mediante encoder:** al girar la perilla del encoder, el sable varía su color según una paleta de 10 colores predefinidos. **Se completó al 100%.**
- ✓ **Memoria SD para efectos de sonido:** mediante un preprocesamiento detallado en la entrega de informe anterior, se consiguió reducir significativamente el espacio en memoria requerido para almacenar el sonido, sumado a que se realiza una modulación a través de la misma CPU de la EDU-CIAA, no se consideró necesario utilizar memoria SD.
- ✓ **Control bluetooth a través de aplicación:** se incorporó al diseño, pero no se implementó por cuestiones de tiempo.
- ✓ **Control por Wi-Fi e interfaz web:** se incorporó al diseño, pero no se implementó por cuestiones de tiempo.

### **Requerimientos no funcionales**

- ✓ **Resistente a los movimientos bruscos:** para ser un prototipo, es resistente. Tal vez con un diseño industrial se lo podría mejorar.
- ✓ **Funcionamiento autónomo:** el sable funciona a batería, dependiendo del volumen de sonido y brillo de los LEDs puede funcionar alrededor de 1h, con las baterías usadas que se le instalaron al prototipo.
- ✓ **PlugAndPlay:** al presionar el botón de encendido, el sable inicia con el color y volumen de sonido establecido o caso contrario el por defecto.
- ✓ **Entrega en tiempo y forma:** la muestra final se realizó el jueves 08/02/2024, estando dentro del tiempo establecido para la entrega.

### 11.3. Conclusiones generales

excelente aporte de cada uno

Aprecié mucho el desafío de lo que pensé que sería un proyecto relativamente sencillo de implementar. Me animó e interesó el hecho de poder usar el laboratorio ATEI para la realización de la PCB y como nuestro diseño en particular nos empujó a trabajar con “limitaciones” de espacio.



En general me gusto mucho poder trabajar en una idea planteada por nosotros mismos y verla materializarse poco a poco fue una experiencia emocionante.

-Joaquin.

Fue un desafío emocionante participar en el proyecto. La primera contribución fue la elección del proyecto, el resto de mis contribuciones se basaron en investigar y hacer las pruebas del MPU6050 y del sistema de sonido. También participe activamente en los informes de avance.

A pesar de que la extensa investigación nos restó bastante tiempo y la falta de una metodología de trabajo grupal no trajo los mejores resultados, reconozco que esta experiencia fortaleció nuestras habilidades técnicas y de colaboración, personalmente quede muy entusiasmado por explorar más este campo a futuro.

- Juan

Al disponer de poca información en internet, el proyecto constó en gran medida en desarrollo de software desde cero, con la consiguiente situación de encontrar dificultades y la necesidad de ajustar los objetivos según las posibilidades.

En cuanto al sonido, no se encontró otro proyecto similar utilizando EDU-CIAA, y se encontró poca información sobre cómo reproducir sonido sin hardware dedicado; luego de investigar, se dio con la idea de obtener las muestras en crudo y reproducirlas vía DAC a la frecuencia de la pista de audio. Una vez estudiado cómo realizar el preprocesamiento del sonido y ya teniendo las muestras listas, se las probó y se escuchaba bien, con lo cual, posteriormente se intentó realizar la reproducción mediante DMA sin éxito. Finalmente se decidió utilizar Timer1 para reproducir, y nuevamente nos encontramos con dificultades para integrar con el giroscopio.

Se tuvo que prescindir de ciertas funcionalidades por falta de tiempo, ya que fue necesario detenerse con dichas problemáticas para poder avanzar, lo que produjo un cambio en el tiempo estimado de desarrollo. De todas formas, se pudo realizar una entrega que cumplía relativamente en gran medida con el plan original.

-Adrián

En cuanto al diseño del proyecto elegido, fue un desafío integrar periféricos tan distintos, analógicos, digitales, botones, filtros, DAC, ADC, entre otros. La mayor dificultad



del diseño de hardware estuvo en la fabricación del propio sable, integrando el sistema de alimentación, los periféricos principales y la PCB pequeña en 1 solo cuerpo. Esta construcción mecánica fue rigurosamente trabajada para que se integre todos los periféricos, corroborando el correcto funcionamiento de cada uno, sin romper los otros. En cuanto al poncho, el diseño y fabricación se llevó sin mucha demora, ya que el poncho está construido casi solo con fichas de conexión. El diseño del software fue un desafío de ingeniería, porque hay componentes muy distintos, a diferentes velocidades, que trabajan en paralelo.

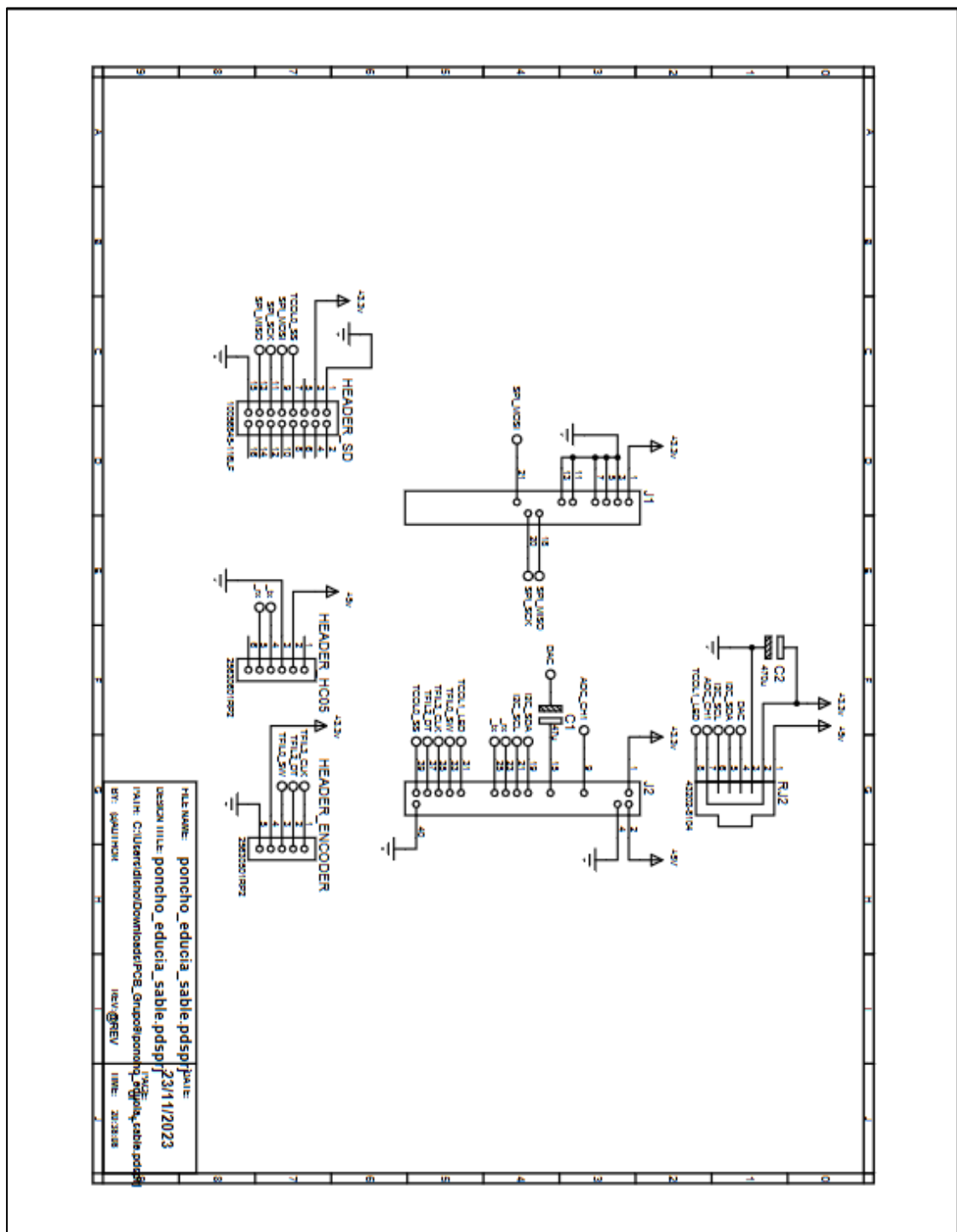
Exceptuando la tira Led, los módulos de software se desarrollaron y probaron independientemente con pocas dificultades. A la hora de hacer la integración de estos módulos, aparecieron diversos errores de incompatibilidad, incluyendo la incompatibilidad con la librería SAPPI, que luego se modificó. El proceso de integración llevo más tiempo de lo esperado, y optamos deshabilitar el MPU6050 para evitar incompatibilidades con el resto del proyecto En cuanto a la gestión del proyecto, pudimos comunicar y repartir las tareas de hardware y software equitativamente, donde cada integrante se hizo responsable de alguna parte del proyecto, las tareas se dividieron habitualmente en sub-grupos de a 2.

-Nicolás

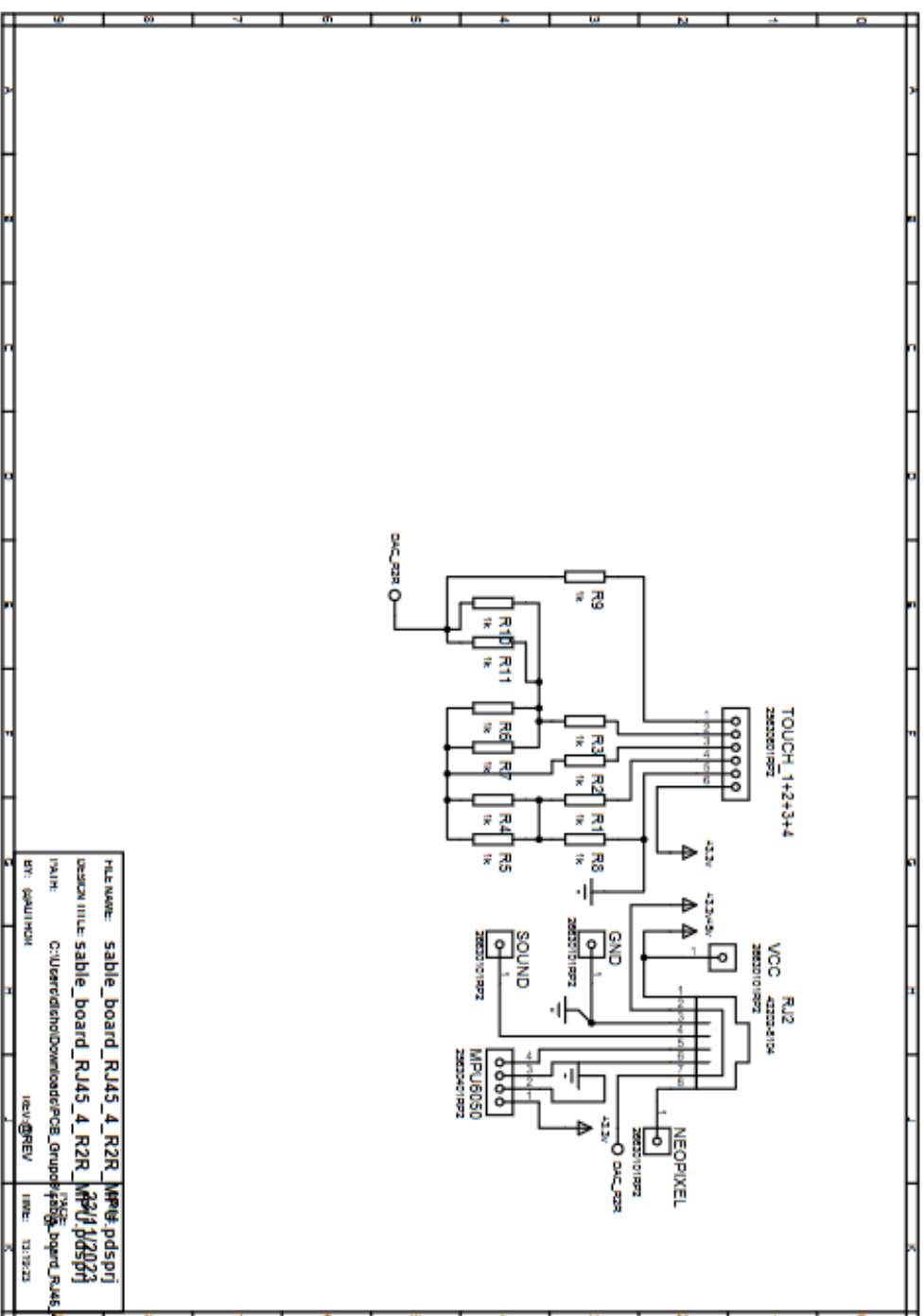


## 12. Anexos

### 12.1. Circuito esquemático



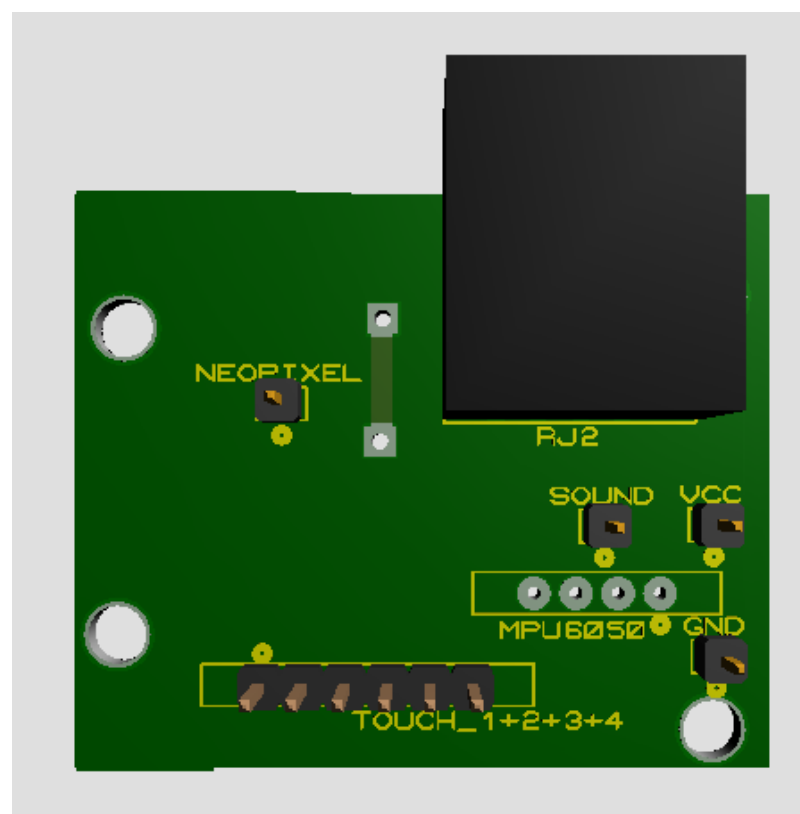
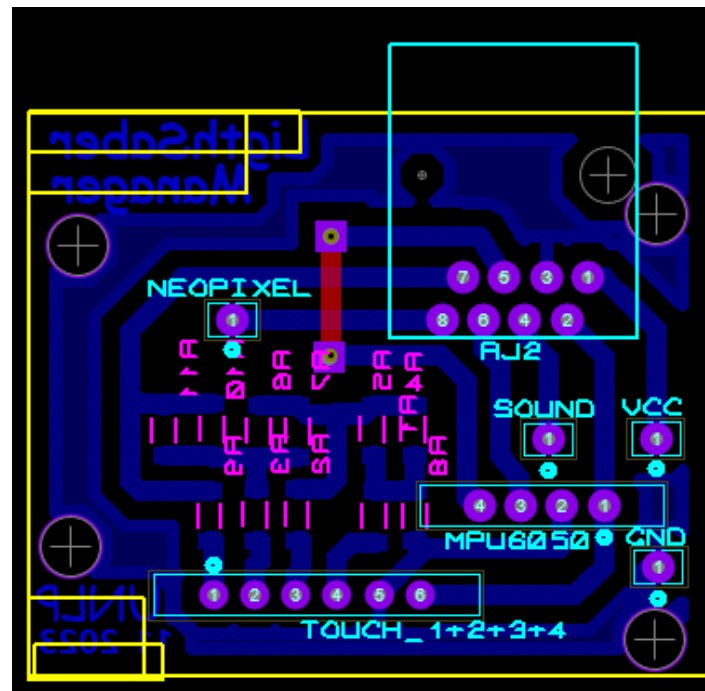
### 12.2. Circuito esquemático R2R

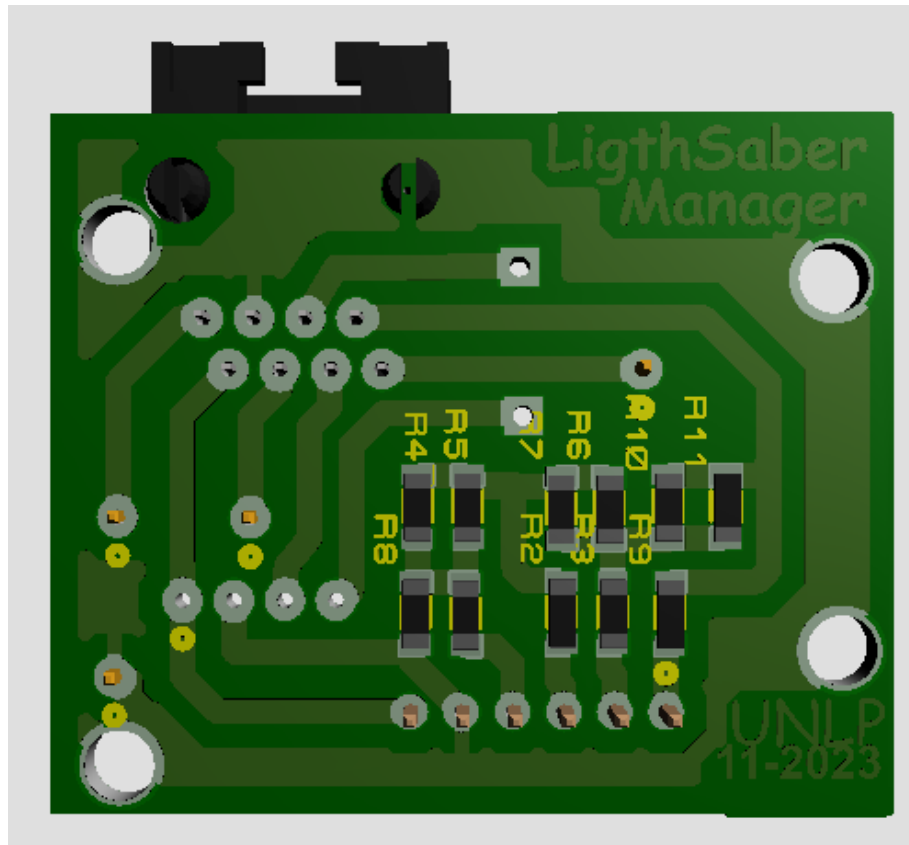


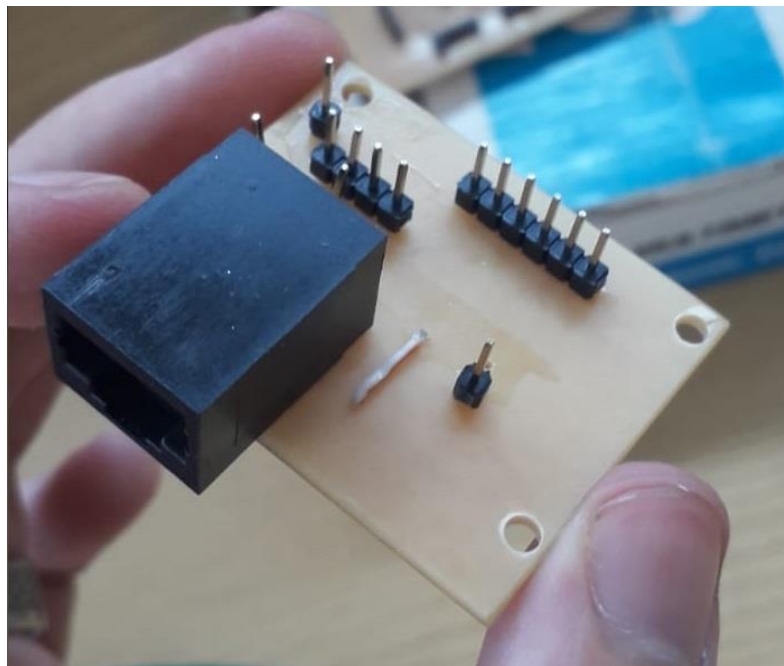
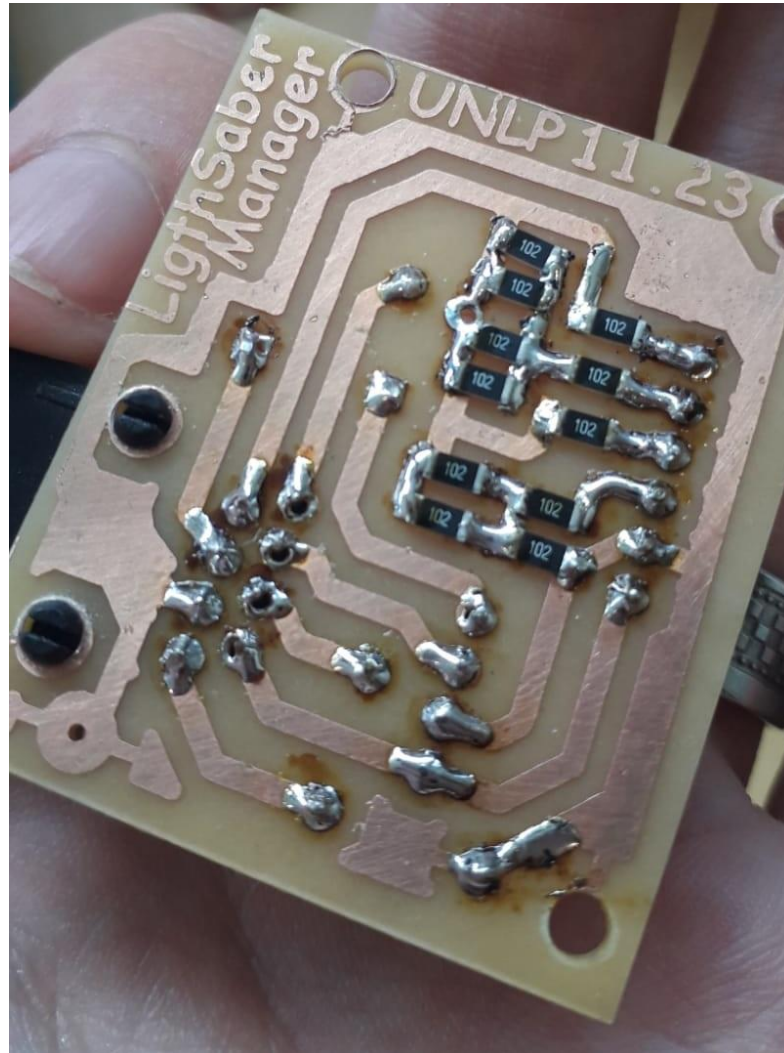
FILE NAME	sable_board_R145_4_R2R_MP01.pdssprj
DESIGN TITLE	sable_board_R145_4_R2R_MP01.pdssprj
PATH	C:\Users\deno\Downloads\PCB_Group01\pcb_group01.pdssprj
REV	1
DATE	12/19/23



12.3. PCB R2R

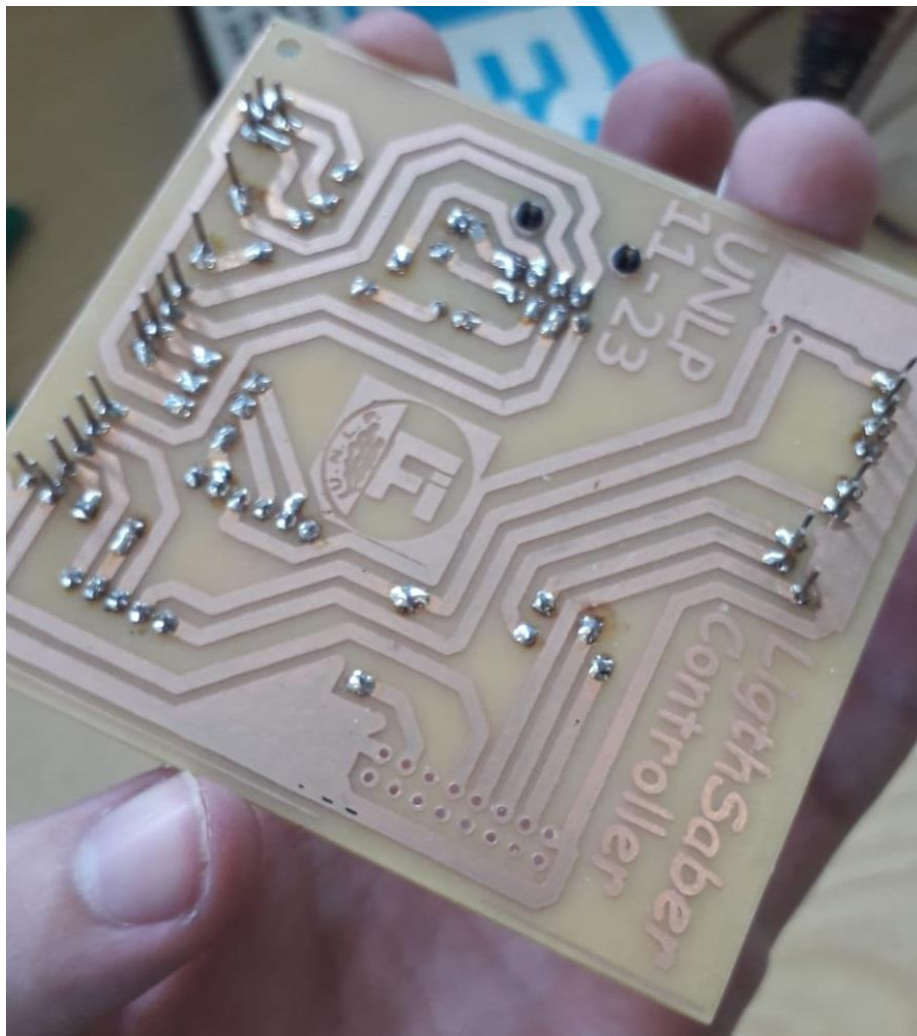
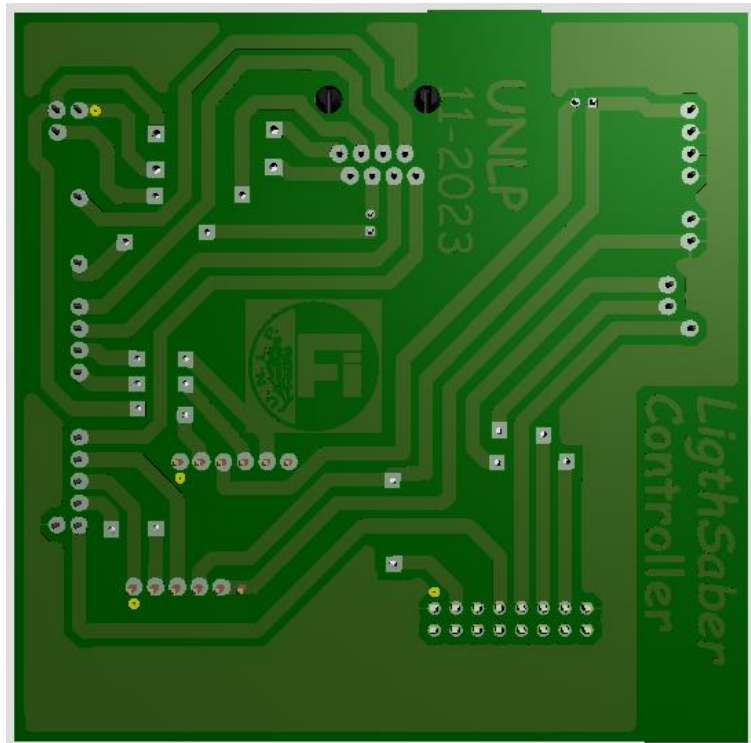


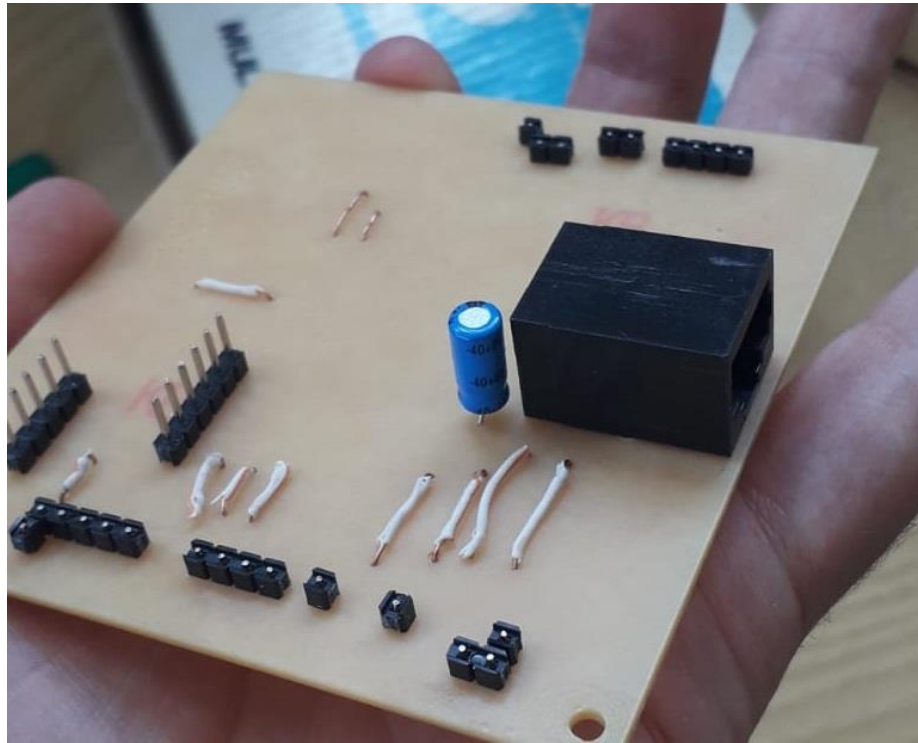








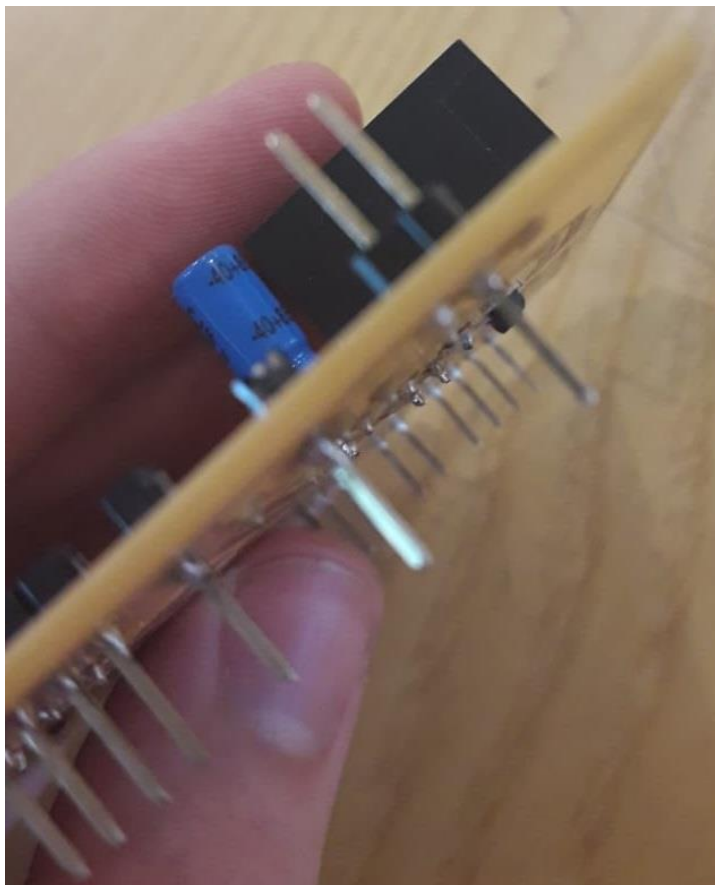
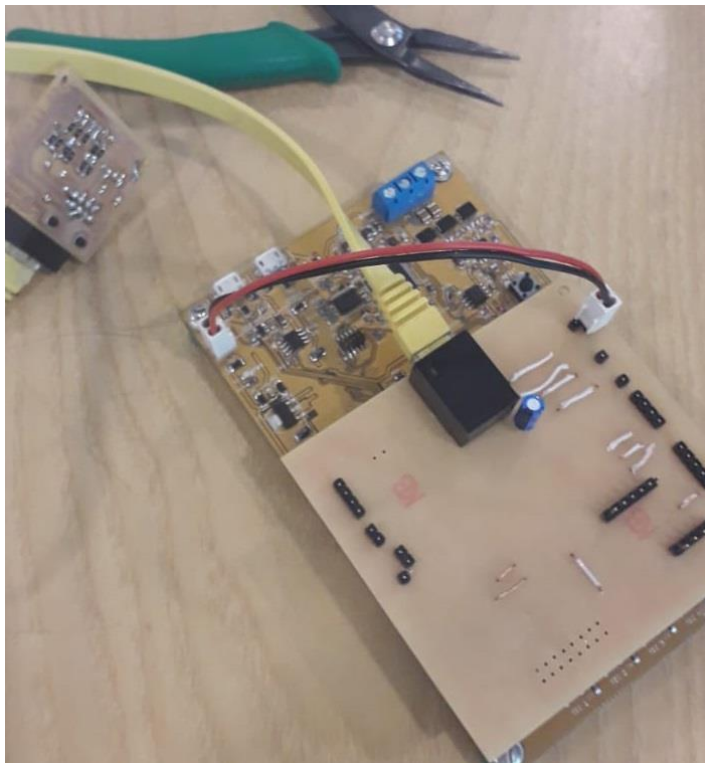








### 12.5. PCB PONCHO Modificada





## 12.6. Lista de materiales

A continuación, se presenta el listado de materiales que se utilizarán en este proyecto, el costo de los materiales es cero debido a que se dispone de ellos.

### PCB chica

#### Bill Of Materials for sable\_board\_RJ45\_4\_R2R\_MPU (2)

**Design Title**           sable\_board\_RJ45\_4\_R2R\_MPU (2)  
**Author**  
**Document Number**  
**Revision**  
**Design Created**       jueves, 23 de noviembre de 2023  
**Design Last Modified** martes, 28 de noviembre de 2023  
**Total Parts In Design** 18

##### 11 Resistors

Quantity	References	Stock Code	Value
11	R1-R11	Resistencias 1K	1k

SMD 1206 ?

Sub-totals:

##### 7 Miscellaneous

Quantity	References	Stock Code	Value
4	GND,NEOPIXEL,SOUND,VCC	Pines Macho	26630101RP2
1	HEADER_MPU6050	Tira 4 Pines Macho	25630401RP2
1	RJ45	Ficha RJ45	43202-8104
1	TOUCH_1+2+3+4	Tira 6 Pines Macho	25630601RP2

Sub-totals:

Totals:

martes, 13 de febrero de 2024 19:16:18



## PCB grande

### Bill Of Materials for poncho\_educia\_sable (2)

**Design Title** poncho\_educia\_sable (2)  
**Author**  
**Document Number**  
**Revision**  
**Design Created** jueves, 23 de noviembre de 2023  
**Design Last Modified** jueves, 30 de noviembre de 2023  
**Total Parts In Design** 8

#### 2 Capacitors

<u>Quantity</u>	<u>References</u>	<u>Value</u>	<u>Stock Code</u>
1	C1	100uF	Capacitor Electrolitico 16V
1	C2	470u	Capacitor Electrolitico 16V

Sub-totals:

#### 6 Miscellaneous

<u>Quantity</u>	<u>References</u>	<u>Value</u>	<u>Stock Code</u>
1	HEADER_ENCODER	25630501RP2	Tira 5 Pines Macho
1	HEADER_HC05	25630601RP2	Tira 6 Pines Macho
1	HEADER_SD	10056845-116LF	Header 2x8 Pines Hembra
2	J1-J2		Tira 2x20 Pines Macho Educiaa
1	RJ2	43202-8104	Ficha RJ45

Sub-totals:

---

Totals:

martes, 13 de febrero de 2024 19:24:25