



Universidad Nacional de La Matanza

**Universidad Nacional de La Matanza
Departamento de Ingeniería e Investigaciones Tecnológicas**

Asignatura: Sistemas Operativos Avanzados

Proyecto: Lightsaber

Ciclo Lectivo: 2do cuatrimestre – 2017

Días de Cursada: Miércoles / Turno: Noche

Aula: 266

Docentes a cargo:

Esteban Carnuccio

Graciela de Luca

Gerardo Garcia

Waldo Valiente

Mariano Volker

Sebastián Barillaro

Integrantes:

Ahmed Nahuel

DNI: 37238669

Artime Martin

DNI: 38066159

Becerra Pablo

DNI: 38428620

Espinola Daniel

DNI: 36697331

Índice

1. Objetivo general del proyecto	3
2. Solución propuesta	3
3. Diagramas	3
4. Hardware utilizado	3
5. Descripción:	4
MPU 6050	4
LDR (Sensor lumínico)	4
KY-021 (Reed Switch)	5
LED RGB	5
Vibrador	5
Parlante	6
Arduino Nano	6
Módulo Bluetooth (HC-05)	7
Módulo SD	8
Powerbank (x2)	8
Resistencias	9
Cajas para electrónica (20x20)	10
Cables varios	10
Plaquetas de prototipado	11
Mango impreso en 3D	12
Tubo de Acrílico	12
6. Diagrama de conexión	13
7. Implementación	13
Sistema embebido	13
8. Aplicación Android	16
9. Problemáticas	16
10. Bibliografía	17

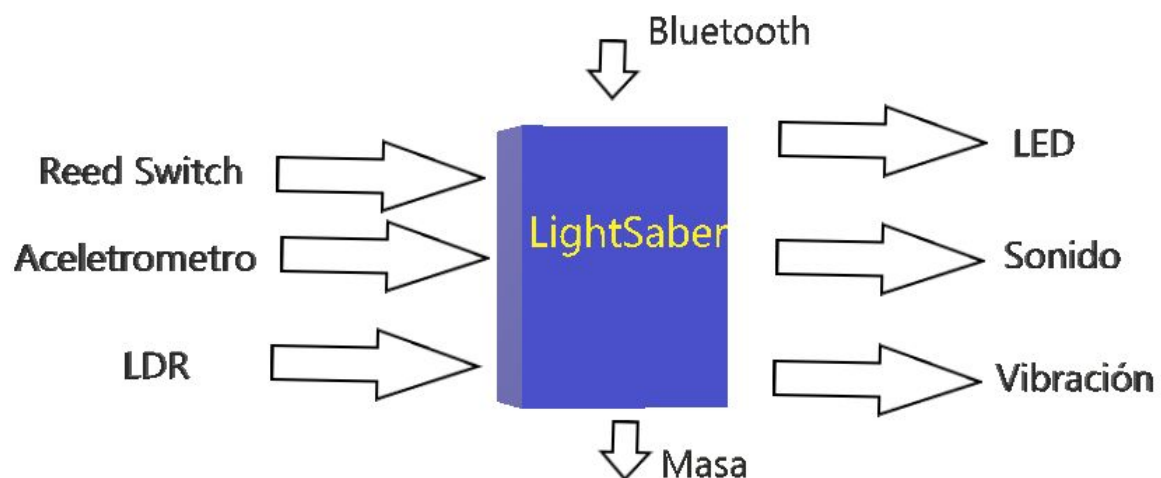
1. Objetivo general del proyecto

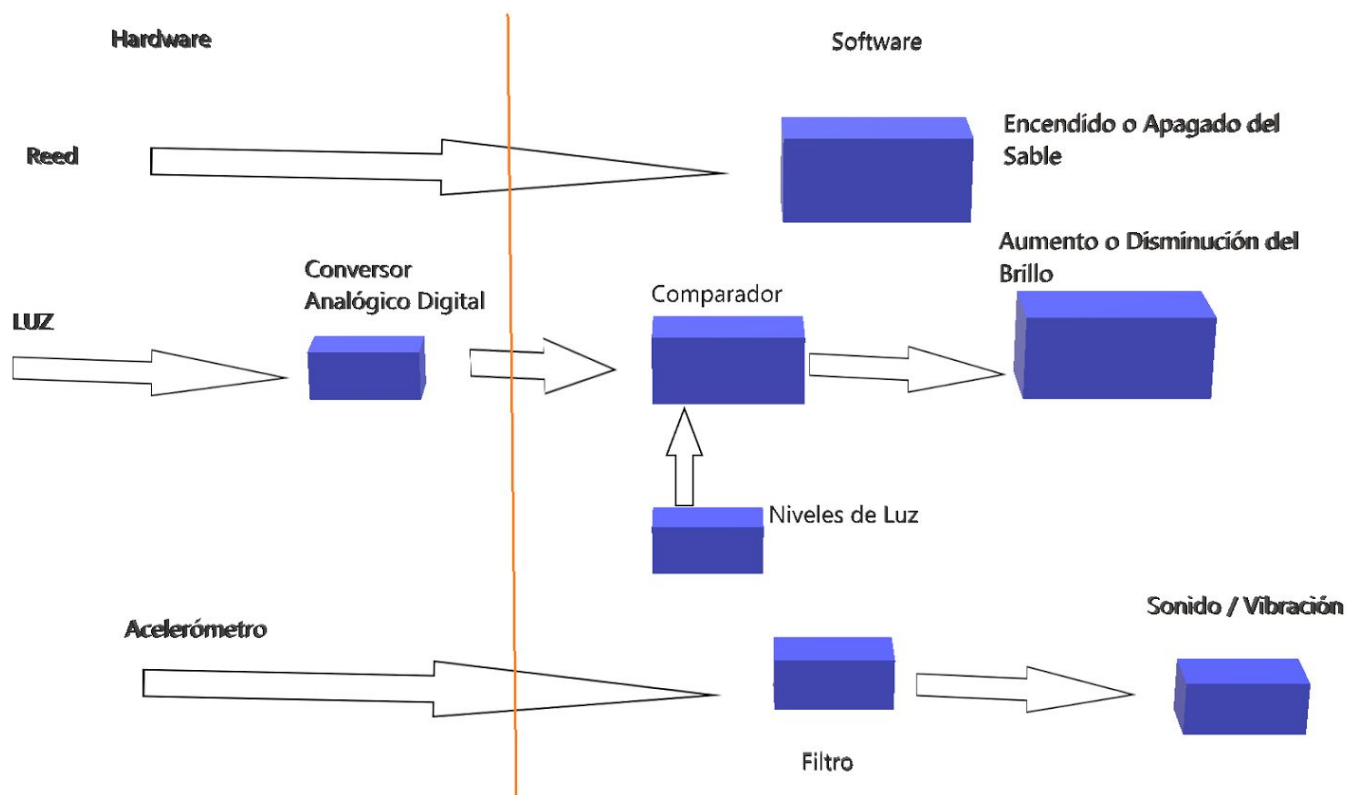
El objetivo del proyecto es crear un prototipo de Sable de Luz que emule las funcionalidades del mítico LightSaber de la saga 'Star Wars'.

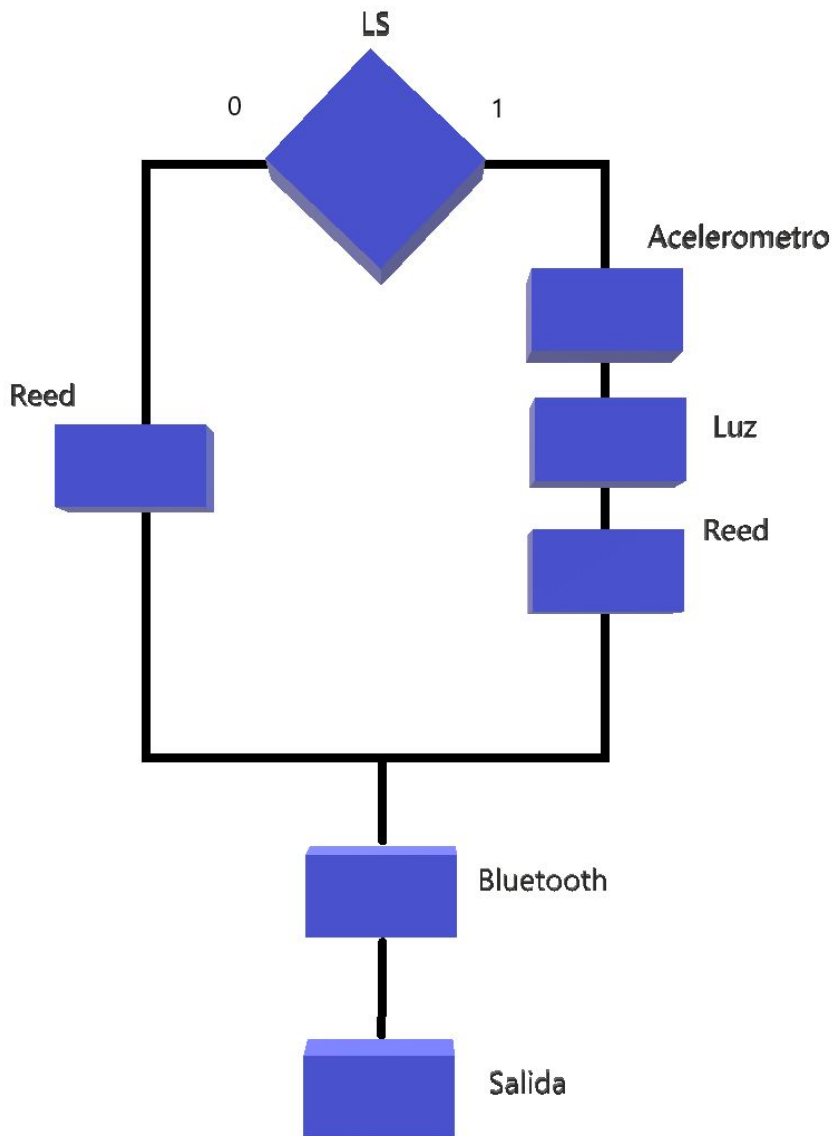
2. Solución propuesta

La solución que proponemos es utilizar un Arduino Nano y diferentes sensores que le dan la funcionalidad deseada. Este sistema embebido se comunica vía bluetooth con un dispositivo Android para activar funcionalidades del Lightsaber.

3. Diagramas







4. Hardware utilizado

Sensores:

- MPU 6050 (Acelerometro / Giroscopo)
- LDR (Sensor lumínico)
- KY-021 (Reed Switch)

Actuadores:

- LED RGB
- Vibrador

- Parlante

Otros componentes:

- Arduino Nano
- Módulo Bluetooth (HC-05)
- Módulo SD
- Powerbank (x2)
- Resistencias
- Plaquetas
- Caja para electrónica (20x20)
- Cables varios
- Plaquetas de prototipado (x2)
- Mango impreso en 3D
- Tubo de Acrílico

5. Descripción:

MPU 6050



El sensor MPU-6050 está compuesto por un acelerómetro y un giroscopio MEMS (sistema microelectromecánico) en su chip. Es muy preciso, ya que contiene hardware de conversión de analógico a digital de 16 bits para cada canal, por lo tanto, captura el canal X, Y y Z al mismo tiempo. El sensor usa el bus I2C para interactuar con el Arduino. En nuestro sistema embebido se utilizó para determinar si se produjo un movimiento o no, si lo hubo se reproducirá un sonido que lo represente.

LDR (Sensor lumínico)



El LDR es un componente electrónico cuya resistencia disminuye con el aumento de intensidad de luz incidente. El valor de resistencia eléctrica de un LDR es bajo cuando hay luz incidiendo en él (puede descender hasta 50 ohms) y muy alto cuando está a oscuras (varios megaohms). Se utilizó para determinar la iluminación y de esta forma aumentar o disminuir el brillo emanado por el LED según corresponda.

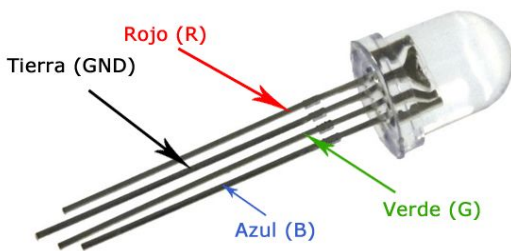
KY-021 (Reed Switch)



Un Reed switch es un sensor magnético que normalmente está abierto y se cierra cuando se expone a un campo magnético.

El módulo incluye una resistencia de 10KOhm, por lo que no se necesita agregar piezas adicionales. Este componente se utilizó para encender el sable de luz, utilizando un imán que activa el funcionamiento del Reed.

LED RGB



Un LED RGB es la unión de tres LEDs de los colores rojo, verde y azul, en un encapsulado común, compartiendo la 'tierra' (cátodo es otro nombre más para el negativo).

En función de la tensión que pongamos en cada pin podemos conseguir la mezcla de color que deseemos con relativa sencillez.

Este componente se utilizó para iluminar un tubo de acrílico.

Vibrador



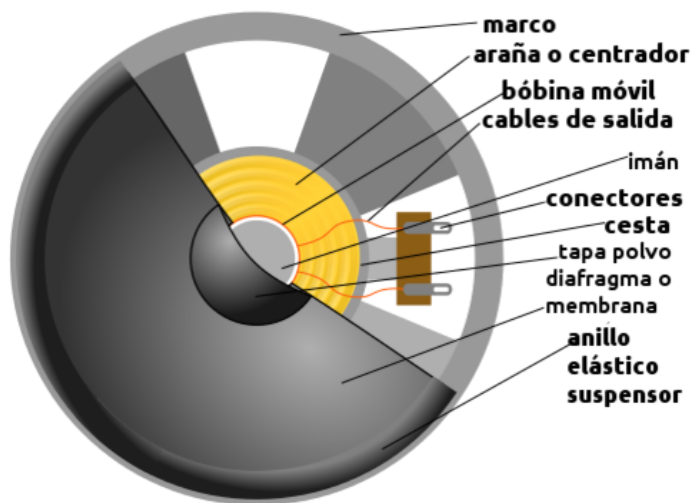
Voltaje de funcionamiento: 1.5-6VDC

Velocidad: 1500 rpm

Eje de vibración Tamaño 18x10x4mm (L * W * T)

Tamaño del cuerpo: 12x25mm (H * D)

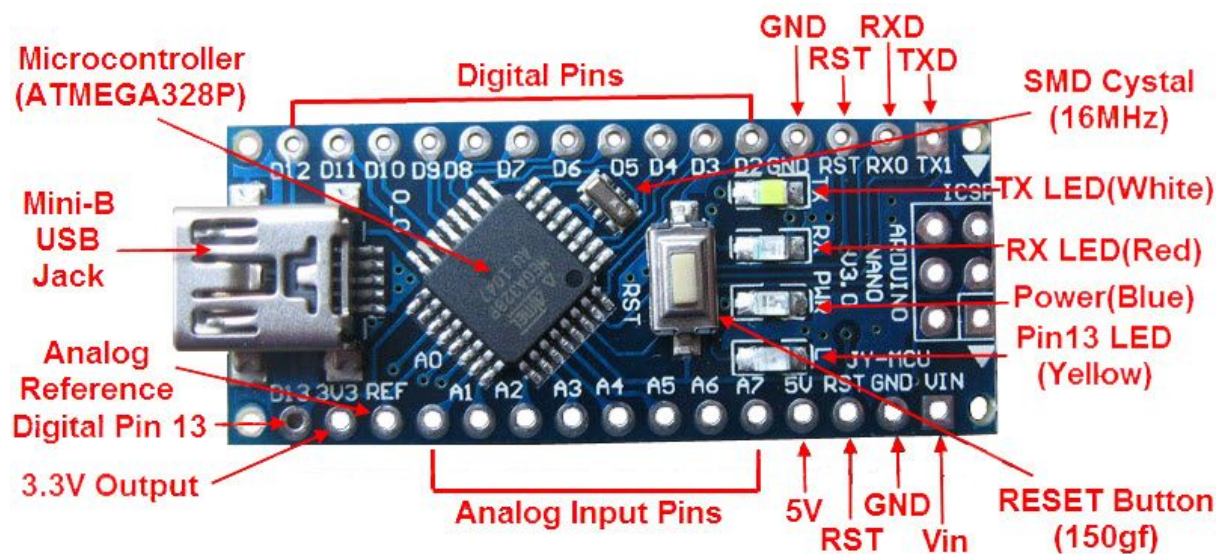
Parlante



Un parlante es un transductor electroacústico utilizado para la reproducción de sonido.

Este elemento se utilizó para reproducir sonidos cuando se enciende, se apaga o se produce algún movimiento al sable de luz.

Arduino Nano



El Arduino Nano es una pequeña y completa placa basada en el ATmega328 que se usa conectándose a una protoboard. Tiene la misma funcionalidad que el Arduino Uno, pero con una presentación diferente. No posee conector para alimentación externa, y funciona con un cable USB Mini-B en vez del cable estándar.

Pines E/S Digitales: 14 (de los cuales 6 proveen de salida PWM)

Entradas Analógicas: 8

Corriente máx por cada PIN de E/S: 40 mA

Módulo Bluetooth (HC-05)



El módulo bluetooth HC-05 viene configurado para trabajar como maestro o esclavo. En el modo maestro puede conectarse con otros módulos bluetooth, mientras que en el modo esclavo queda a la escucha peticiones de conexión.

Características:

TENSIÓN DE ALIMENTACIÓN: 3.3 V.

Alcance 10 mts

1200bps a 1.3Mbps

MEDIDA 28X15X2.35MM

Este componente se utilizó para generar una conexión entre una aplicación desde un celular con Android y el arduino para el envío de “mensajes” y así utilizar los actuadores del arduino.

Módulo SD



Nos permite insertar una memoria Micro SD, el módulo se puede alimentar con 3.3V o 5V usando los pines respectivos:

- 5v
- gnd
- cs (Chip Select)
- mosi (Master Out Slave In)
- sck (Serial Clock)
- miso (Master In Slave Out)

Este componente se utilizó para almacenar los diferentes sonidos para que se reproduzcan a través del parlante.

Powerbank (x2)



Power Bank hace referencia a las pilas y baterías recargables de material de ion de litio o polímero de litio con salidas USB y Micro USB para efectuar cargas de dispositivos electrónicos .

Los power bank o baterías externas tienen diferentes capacidades que se mide en mAh (miliamperio x hora) que es una unidad de medida utilizada para medir la carga eléctrica acumulada en un determinado período de tiempo.

En nuestro proyecto se implementó 2 powerbank por el gran consumo de corriente que teníamos en especial por el módulo bluetooth.

El primer powerbank se conecta al arduino y este alimenta a todo el circuito a través de la salida de 5v, mientras que el segundo powerbank, alimenta solo al bluetooth, para ello se cortó un cable usb, para soldar los cables de alimentación del mismo al módulo bluetooth.

Resistencias



En el proyecto se utilizaron 6 resistencias en total. Para el LED RGB, reed, vibrador y parlante

Las mediciones son las siguientes:

- Para el reed 10k ohm
- Para el led RGB, 2 de 270 ohm
- Para el vibrador 10k ohm
- Para el parlante 10k ohm

Cajas para electrónica (20x20)



Se utilizó para colocar la mayoría de los componentes del circuito (fotoresistor, plaquetas, vibrador, parlante , powerbank,etc) y se coloco debajo del mango impreso conectados entre si.

Este elemento tuvo la necesidad de usarse ya que no se podían colocar todos los elementos en el mango impreso del sable.

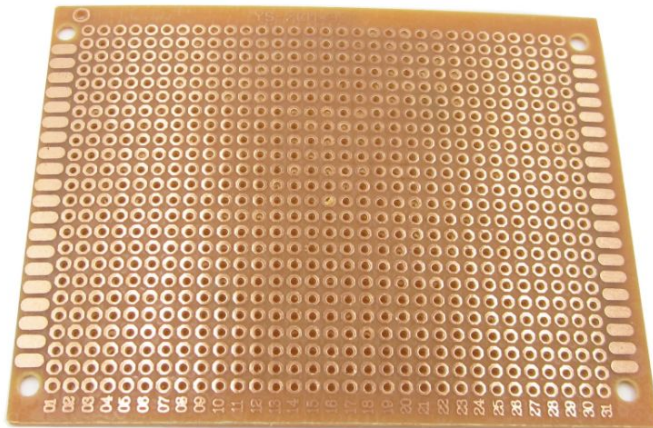
Cables varios



Los cables puente se fijan mediante la inserción de sus extremos en los agujeros previstos a tal efecto en las ranuras de la placa de prototipado, la cual debajo de su superficie tiene unas planchas interiores paralelas que conectan las ranuras en grupos de filas o columnas según la zona. Los conectores se insertan en la placa de

prototipado, en los agujeros que convengan para el conexionado del diseño.

Plaquetas de prototipado



Es un tablero con orificios que se encuentran conectados eléctricamente entre sí de manera interna, habitualmente siguiendo patrones de líneas, en el cual se pueden insertar componentes electrónicos y cables para el armado de circuitos electrónicos. En el proyecto se usaron dos plaquetas para colocar los componentes electrónicos y se las colocó en la caja para electrónica.

Mango impreso en 3D



En este componente se pusieron el LED RGB con el acrílico y el Reed Switch para

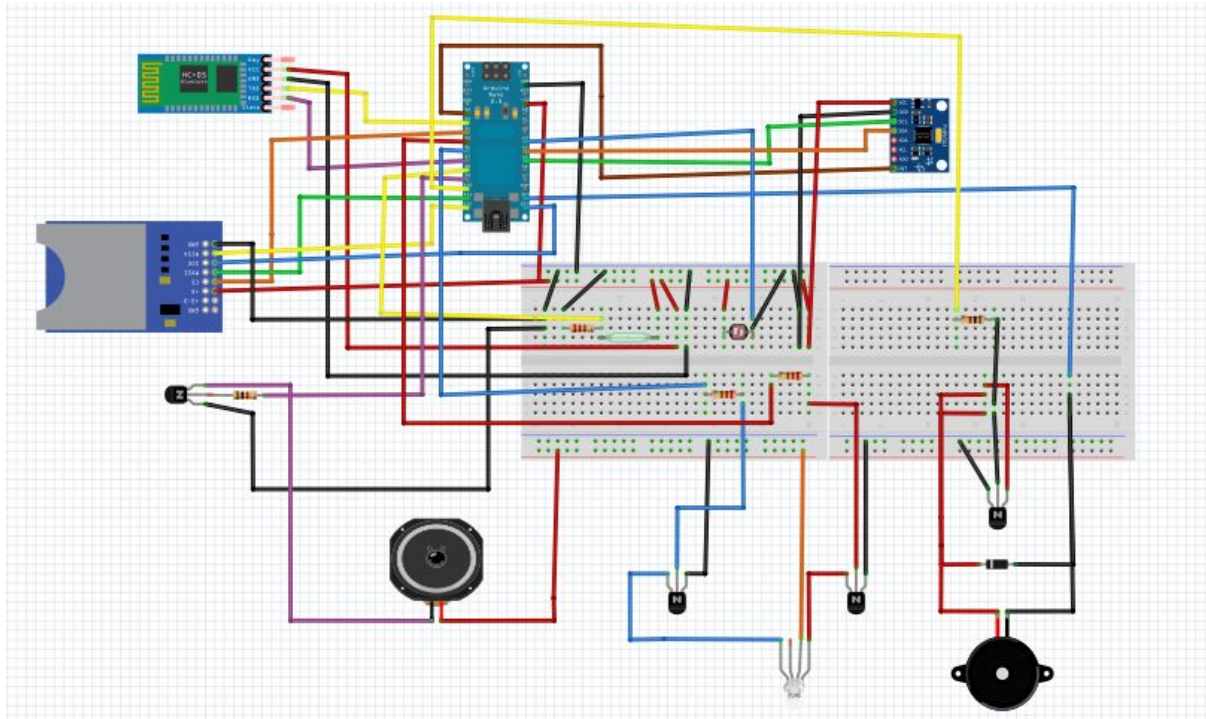
prender el sable utilizando un imán.

Tubo de Acrílico



Este componente se utilizó para reflejar la luminosidad del sable con el LED RGB. Utilizamos un tubo de 60cm de largo con el diámetro perfecto para que entre justo con el diámetro del sable 3D.

6. Diagrama de conexión



7. Implementación

Sistema embebido

Una vez hechas todas las conexiones pertinentes expuestas en la ilustración anterior, se pasa a implementar el código <https://github.com/The-Jedi-Order/Lightsaber> en el SE.

Bibliotecas:

Wire

Se encarga de la comunicación a través del I2C.

“La biblioteca de Wire configura el chip ATmega para que el hardware TWI interno use sus pines, que están codificados en dos pines analógicos Primero prueba el bus I2C para ver si esta libre. El bus está libre cuando ambos, SDA (pin 4) y SCL (pin 5), son altos (porque los pines del dispositivo tienen un estado predeterminado en un estado tri-state o no conectado en el bus)”

Primero debe llamarse a `Wire.begin ()` o `Wire.begin (dirección)` para iniciar cualquier comunicación I2C utilizando la biblioteca Wire. El método `Wire.begin ()` establece los índices y longitudes de almacenamiento interno de lectura y escritura de la biblioteca Wire en 0.

<https://www.arduino.cc/en/Reference/Wire>

<https://playground.arduino.cc/Main/WireLibraryDetailedReference>

iC2dev

“La biblioteca de dispositivos I2C (i2cdevlib) es una colección de clases uniformes y bien documentadas para proporcionar interfaces simples e intuitivas a los dispositivos I2C. Cada dispositivo está diseñado para hacer uso de la clase genérica I2Cdev, que abstrae la comunicación I2C de nivel de bit y de bytes de cada clase de dispositivo específico, lo que facilita la limpieza de la clase del dispositivo y proporciona una forma sencilla de modificar una clase para portar el código de comunicación I2C en diferentes plataformas (Arduino, PIC, MSP430, Jennic, simple bit-banging, etc.). Las clases de dispositivos están diseñadas para proporcionar una cobertura completa de todas las funcionalidades descritas en la documentación de cada dispositivo, además de funciones de conveniencia genéricas que son útiles.”

Debe ser usada con el `wire.h`

<https://github.com/jrowberg/i2cdevlib>

MPU6050

Representa al sensor, dicha biblioteca se encarga de inicializar el dispositivo y tomar los valores que este le da.

<https://github.com/jrowberg/i2cdevlib/blob/master/Arduino/MPU6050/MPU6050.h>

SoftwareSerial

El hardware Arduino tiene soporte incorporado para comunicación serie en los pines 0 y 1 a través de UART.

La biblioteca SoftwareSerial se ha desarrollado para permitir la comunicación serial en otros pines digitales del Arduino, utilizando software para replicar la funcionalidad (de ahí el nombre SoftwareSerial). Es posible tener múltiples puertos seriales de software con velocidades de hasta 115200 bps. Un parámetro habilita la señalización invertida para los dispositivos que requieren ese protocolo. La biblioteca tiene las siguientes limitaciones conocidas:

Si utiliza múltiples puertos serie de software, solo uno puede recibir datos a la vez.

<https://www.arduino.cc/en/Reference/SoftwareSerial>

SD

La biblioteca SD permite leer y escribir en tarjetas SD.

Los nombres de archivos pasados a las funciones de la biblioteca SD pueden incluir rutas separadas por barras diagonales, /, directorio / nombreadarchivo.txt. Como el directorio de trabajo siempre es la raíz de la tarjeta SD, un nombre hace referencia al mismo archivo, incluya o no una barra diagonal (por ejemplo, /archivo.txt es equivalente a archivo.txt). A partir de la versión 1.0, la biblioteca admite la apertura de múltiples archivos.

La comunicación entre el microcontrolador y la tarjeta SD utiliza SPI, que tiene lugar en los pines digitales 11, 12 y 13 (en la mayoría de las placas Arduino) o 50, 51 y 52 (Arduino Mega). Además, se debe usar otro pin para seleccionar la tarjeta SD. Este puede ser el pin SS de hardware - pin 10 (en la mayoría de las placas Arduino) o pin 53 (en la Mega) - u otro pin especificado en la llamada a SD.begin (). Tenga en cuenta que incluso si no utiliza el pin SS de hardware, debe dejarse como salida o la biblioteca SD no funcionará.

<https://www.arduino.cc/en/Reference/SD>

TRCMP

Biblioteca Arduino para reproducción asincrónica de archivos PCM / WAV directamente desde la tarjeta SD Utiliza la biblioteca estándar Arduino SD, tarjeta SD y dispositivo de salida (altavoz,

auriculares, amplificador, etc.) Funciones:

TMRpcm audio;

audio.play(archivo); reproduce un archivo

audio.play(archivo,30); reproduce un archivo empezando a los 30 segundos

audio.speakerPin = 11;

audio.disable(); deshabilita el temporizador en el pin de salida y detiene el sonido.

audio.stopPlayback(); detiene el sonido pero deja el temporizador corriendo.

audio.isPlaying(); retorna 1 si el audio se está reproduciendo y sino 0.

audio.pause(); pausa o continua reproduciendo el audio.

audio.quality(1); Setear 1 para 2x oversampling

audio.volume(0); 1 sube el volumen y 0 lo baja.

audio.setVolume(0); 0 a 7. Setea el volumen.

audio.loop(1); 0 o 1. Puede ser cambiado en la reproducción para un control absoluto del loop.

Limitaciones:

“Esta biblioteca puede ser muy intensiva en el uso del procesador, y la ejecución del código durante la reproducción será más lenta de lo normal La carga de procesamiento se puede reducir utilizando sonidos de menor calidad codificados a una frecuencia de muestreo más baja (8 kHz mínimo) Puede interferir con otras bibliotecas que dependen de las interrupciones. El isPlaying () deshabilita () o no Interrumpe() las funciones se pueden usar para evitar la ejecución paralela de código. El control de

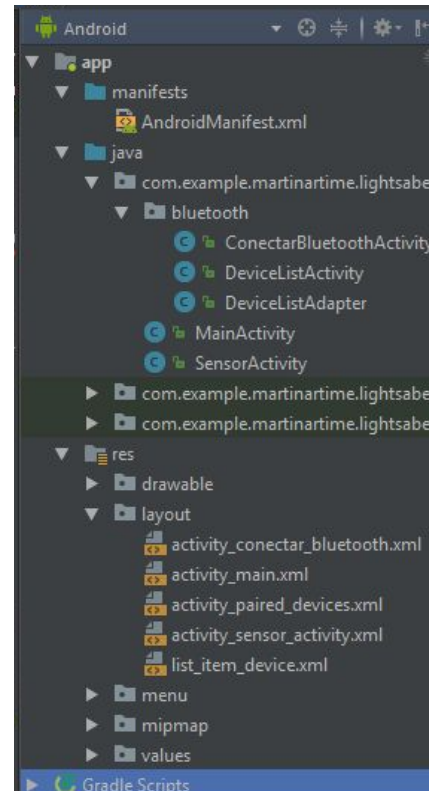
volumen permite un buen rango en el control de volumen, pero se distorsionará si el volumen es demasiado alto”

<https://github.com/TMRh20/TMRpcm/wiki>

8. Aplicación Android

La aplicación de Android consta del siguiente árbol de clases:

- MainActivity: es el menú principal mediante el cual se puede acceder a la conexión bluetooth (ConectarBluetoothActivity) y a la de los sensores (SensorActivity).
- Sensor Activity: en ésta clase se puede ver los valores que están siendo sensados por los sensores del celular en tiempo real y por medio de afectarlos se comunica por bluetooth con el Arduino y el sable actúa acorde. Es obligatorio estar en estado de "bond" con el HC-05 para poder acceder a esta parte de la app.
- ConectarBluetoothActivity: ésta activity permite encender el bluetooth desde la app, realizar la conexión con dispositivos previamente conectados (ir a DeviceListActivity) o buscar nuevos dispositivos (misma activity de destino).
- DeviceListAdapter: esta clase proporciona un formato de como se formará cada registro nuevo encontrado en la clase anterior.
- DeviceListActivity: utiliza el anterior adapter para mostrar los datos de los dispositivos encontrados permitiendo la conexión y bondeo con los mismos. Luego de realizar el apareo se vuelve al menú principal, ahora pudiendo acceder a SensorActivity.



9. Problemáticas

A lo largo del desarrollo surgieron una variedad de problemas que fueron resueltos (o no) eventualmente.

Entre ellos:

Diseño:

- Tomámos la errónea decisión de mandar a imprimir el mango en 3D previo a tener bien diseñada la disposición de los componentes electrónicos por lo que tuvimos que ajustarlo luego agregando una caja abajo (poco estético).
- Al tomar mal la medida del mango también nos vimos afectados en utilizar dos plaquetas de prototipado que si al final hubiéramos usado la caja nos hubiera convenido utilizar 1 sola más grande.

Electrónica:

- Alimentación: debido a que el módulo bluetooth consume mucho más de lo que imaginábamos, tuvimos que utilizar otro powerbank exclusivo para poder manejar todos los componentes sin que se perjudique la performance general.
- Similarmente, inicialmente teníamos un problema donde el parlante para

reproducir sonidos, debido a la falta de potencia en el circuito, sonaba extremadamente bajo, lo que era inaceptable para la presentación del prototipo.

- Ruido eléctrico: actualmente (es algo que no se pudo resolver), debido a cómo están dispuestos los componentes y a que no tenemos ningún tipo de filtro aplicado al circuito, el parlante al reproducir emite un sonido (blanco) muy agudo paralelamente a los sonidos intencionales, algo indeseable pero que está actualmente presente en el prototipo.

Físico:

- Como “parte luminosa” del sable utilizamos un tubo de acrílico de 60cm de largo y 16mm de diámetro, cuando íbamos a poner el tubo en el mango impreso en 3D, nos dimos cuenta que hacía falta limarlo y quitarle diámetro (inicialmente tenía 20mm) para adaptarlo a la forma del mango, esto se hizo y está resuelto actualmente.
- Al tomar la decisión de utilizar una caja conectada al mango por debajo nos vimos envueltos en la problemática de diseñar el tamaño, disposición de los elementos, agujeros, conexión con el mango para formar una sola pieza, agujeros para la salida del sonido y del cabezal del sensor LDR.

10. Bibliografía

Bibliografía que se utilizó para la construcción de la Aplicación Android

- <https://www.tutorialspoint.com/android/>
- https://github.com/codepath/android_guides/wiki/Reducing-View-Boilerplate-with-Butterknife
- https://developer.android.com/guide/topics/sensors/sensors_overview.html
- <https://paruro.pe/aprende/arduino/bluetooth/comunicaci%C3%B3n-arduino-android-por-bluetooth>

Bibliografía que se utilizó para el Arduino

- <https://www.arduino.cc/>
- https://create.arduino.cc/projecthub/chathuranga-liyanage/prank-your-friends-with-talking-darth-vader-972824?ref=search&ref_id=saber&offset=0