

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA ELECTRÓNICA Y
AUTOMÁTICA INDUSTRIAL



"SENSORIZACIÓN TÁCTIL GESTUAL POR
BARRIDO DE FRECUENCIAS"

TRABAJO FIN DE GRADO

Septiembre
2023

AUTOR: Jaime Ortín Ruiz

DIRECTOR: Rafael Maestre Ferriz

Índice

RESUMEN	4
1.INTRODUCCIÓN	5
1.1. PROBLEMA O NECESIDAD IDENTIFICADA.....	5
1.2 OBJETIVOS.....	6
1.2.1. OBJETIVOS GENERALES	6
1.2.2. OBJETIVOS ESPECÍFICOS	7
1.3 BENEFICIOS DERIVADOS DEL DESARROLLO DEL PROYECTO.....	8
1.4 ESTRUCTURA DEL PROYECTO	9
2.ESTADO DE LA TÉCNICA	10
2.1. HISTORIA Y EVOLUCIÓN DE LOS SENSORES TÁCTILES	10
2.2 FUNDAMENTOS DE LA TECNOLOGÍA MEDIANTE BARRIDO EN FRECUENCIA (TECNOLOGÍA TOUCHÉ)	16
2.3. TECNOLOGÍAS Y SOLUCIONES EXISTENTES.....	17
2.4. LIMITACIONES Y DESAFÍOS DE LAS SOLUCIONES ACTUALES	19
3.METODOLOGÍA	20
3.1 MODELO TEÓRICO.....	20
3.1.1. TOUCHÉ CAPACITIVO.....	20
3.1.2. TOUCHÉ INDUCTIVO.....	23
3.2. HERRAMIENTAS, SOFTWARE Y HARDWARE EMPLEADOS	26
3.2.1. HERRAMIENTAS HARDWARE.....	26
3.2.2. HERRAMIENTAS SOFTWARE	29
3.3. PROCEDIMIENTOS Y TÉCNICAS DE RECOLECCIÓN DE DATOS.....	32
4.DESARROLLO Y RESULTADOS	34
4.1. DETALLES DE LA SOLUCIÓN PROPUESTA	34
4.2 PROCESO DE DISEÑO.	36
4.2.1 COMPONENTES DEL CIRCUITO SEGÚN TOUCHÉ.	36
4.2.2. EVOLUCIÓN Y FASES DEL DISEÑO DEL CIRCUITO TOUCHÉ.....	45
4.2.2.1. FASE 1: OFFSET Y RATIO.....	45
4.2.2.1. FASE 2: CONDENSADORES DE DESACOPLO	48
4.2.2.3. FASE 3: SELECCIÓN DE LA BOBINA Y PRIMERA PCB.....	50
4.2.2.4. FASE 4: SIMPLIFICACIÓN DEL CIRCUITO.....	54
4.2.2.5. FASE 5: DETECCIÓN DE VALLES	57
4.2.2.6. FASE 6: MEJORA DE LA SEÑAL E IMPLEMENTACIÓN DE LA PCB FINAL	59
4.3 PRUEBAS Y VALIDACIONES REALIZADAS	65

4.3.1. CIRCUITO TOUCHÉ CAPACITIVO	66
4.3.1.1. PRUEBA DE VELOCIDAD DE MUESTREO	66
4.3.1.2. PRUEBA DE FUERZA	67
4.3.1.3. PRUEBA DE AISLANTE.....	68
4.3.1.4. PRUEBA DE MATERIALES.....	70
4.3.1.5. PRUEBA CON DIFERENTES PERSONAS	71
4.3.1.6. PRUEBA EN ASIENTO	73
4.3.1.7. PRUEBAS EN UN COLCHÓN.....	76
4.3.2. CIRCUITO TOUCHÉ INDUCTIVO	78
4.3.2.1. PRUEBA CON UNA BOBINA DE 50 ESPIRAS Y UNA CAPACIDAD DE 3PF	78
4.3.2.2. PRUEBA CON UNA BOBINA DE 50 ESPIRAS Y UNA CAPACIDAD DE 220PF	80
4.3.2.3. PRUEBA DE LONGITUD DE BOBINAS	81
4.3.3. DESARROLLO DE LA PROGRAMACIÓN.....	82
5.DISCUSIÓN	85
5.1. RESUMEN DE LOS PUNTOS CLAVE	85
5.2. COMPARACIÓN CON OTROS TRABAJOS O SOLUCIONES EXISTENTES	87
6.CONCLUSIONES	91
6.1. REFLEXIONES FINALES SOBRE LO APRENDIDO Y LOGRADO	91
6.2. LIMITACIONES DEL ESTUDIO Y DE LA SOLUCIÓN PROPUESTA	93
6.3. SUGERENCIAS PARA FUTUROS TRABAJOS	95
7.BIBLIOGRAFÍA	98
ANEXOS	100
ANEXO I: CÁLCULOS.....	101
ANEXO 2: CÓDIGO	103

RESUMEN

En este Trabajo de Fin de Grado se ha desarrollado e implementado un circuito basado en la Tecnología Touché de Disney, la cual utiliza la detección capacitiva por barrido de frecuencias. Con lo que demostramos, que el circuito será capaz de generar una señal senoidal de una frecuencia variable hasta los 3,5MHz, con pasos de 17,5KHz, por lo que tendremos un total de 200 pasos.

El barrido se utiliza para la distinción adecuada de gestos en un objeto sensorizado, por medio de la medida de la respuesta en frecuencia, la cual irá creando diferentes perfiles dependiendo de la forma en la que nosotros interactuemos con el objeto sensorizado.

Una vez definido y descrito tanto el funcionamiento a nivel general como el objetivo de dicha tecnología, se presenta el diseño del circuito, y la implementación de programas en C++ que permiten generar dicha señal, recoger la variación de la señal tras interactuar con el objeto sensorizado y analizar los datos para poder muestrearlos en una gráfica.

Tras una primera implementación se describirán las pruebas realizadas en tres objetos o entornos conocidos: una bola (semejante al pomo de una puerta), una silla y una cama. La finalidad de estas pruebas es refinar Touché y tener un catálogo de actividades con sus respectivas frecuencias para poder identificar y asociar una funcionalidad a cada gesto o interacción, y así estimar las posibilidades y potencial del sistema.

Tras las pruebas realizadas se habilitarán dos programas. Uno cuya función sea distinguir la interacción que estemos realizando en un objeto dado, y otro, que consistirá en un sistema de mensajería capaz de enviar un correo electrónico a una hora que nosotros escojamos si se detecta la actividad que nosotros definamos en el programa.

1.INTRODUCCIÓN

1.1. PROBLEMA O NECESIDAD IDENTIFICADA

Actualmente en la sociedad en la que vivimos, estamos rodeados de tecnología, la cual hace más fácil nuestro día a día, a la par que más cómoda. Esta tecnología no es inmutable, de hecho, está en continuo desarrollo y perfeccionamiento.

Gran parte de esta tecnología se usa para mejorar la comodidad en la vivienda que es lo que habitualmente denominamos como “Domótica”. Ésta busca implementar tecnología en el hogar con la finalidad de hacer múltiples tareas más cómodas. La domótica presenta desafíos en términos de instalación y costos, además de la complejidad que implica tomar decisiones debido a la gran cantidad de alternativas de sistema domótico a implementar en el hogar. Entre otras opciones, por ejemplo, hay tecnología intrusiva que afecta directamente a la privacidad de la persona, como es la videovigilancia por medio de cámaras, o la escucha mediante micrófonos. Es evidente que se necesitan soluciones tecnológicas para estos problemas que abaraten costes, que no requieran tanta sensorización y que se instalen de una manera sencilla y transparente en el domicilio. La finalidad de todo esto sería facilitar las actividades de la vida cotidiana de manera más cómoda y sin apenas percibir que la tecnología está interviniendo para ayudarnos.

Por otro lado, un tema relacionado con esto es el cuidado de personas mayores. La tecnología es capaz de abarcar las necesidades que este sector de la sociedad tiene para poder facilitarles la vida, para poder prevenir riesgos o para poder actuar de manera rápida y efectiva en caso de que se haya producido algún accidente en el hogar. La tecnología domótica puede darles estas soluciones y, además, poder prolongar la independencia de estas personas, sin embargo esto es complejo, porque como hemos mencionado antes, mucha de esta independencia se basa en la colocación de elementos intrusivos en el hogar con el fin de vigilar a este colectivo, ya sean los familiares o cuidadores, por lo que la necesidad implícita de las personas mayores es poder hacer una vida normal sin apenas percatarse de que la tecnología es la que les está ayudando día a día.

1.2 OBJETIVOS

Con el fin de afrontar las necesidades definidas en el apartado anterior, este TFG plantea contribuir a la sensorización de distintas zonas del hogar, de la manera más eficiente y económica posible, y que nos permita conocer las distintas actividades que una persona pueda realizar en su propio domicilio. Para ello se ha elegido una tecnología con un gran potencial y a la vez no lo suficientemente conocida y utilizada, como es la tecnología de Touché. Touché es un tipo de circuito de detección capacitiva por barridos de frecuencia (SFCS), es decir, es una tecnología capaz de detectar variaciones en una señal mediante la interacción humana en objetos o entornos que lleven incorporado partes metálicas o electrodos.

1.2.1. OBJETIVOS GENERALES

El propósito principal de este proyecto consiste en implementar, investigar y demostrar el funcionamiento del circuito Touché, así como las posibles mejoras que puedan llevarse a cabo en el mismo. Con ello se pretende desarrollar una tecnología sensora de contacto que además pueda utilizarse como interfaz de usuario gestual.

Esta tecnología será capaz de sensorizar objetos de uso cotidiano, como, por ejemplo, el pomo de una puerta (para este proyecto se ha empleado una bola metálica similar al citado pomo). El circuito deberá ser capaz de distinguir cinco actividades distintas relacionadas con dicho pomo, las cuales serán detalladas más adelante (Figura 1). Asimismo, se sensorizará una silla y una cama, cada una con sus respectivas actividades específicas.

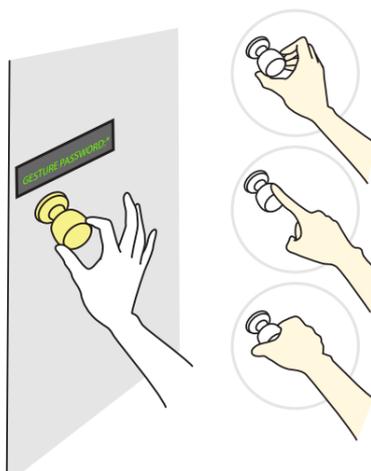


Figura 1: Ejemplos de gestos en el pomo de una puerta

En cuanto a los objetivos, más allá de lo que se hará en términos técnicos, la finalidad de este proyecto es poder brindar comodidad. Esto se traduce en la capacidad de realizar acciones diversas, como bloquear o desbloquear una puerta, o incluso impedir que se abra durante un período determinado, todo mediante gestos en el pomo correspondiente. Aunque pueda parecer un detalle trivial, este enfoque contribuirá gradualmente a hacer los hogares más confortables, sensorizando diversos objetos domésticos para un uso en específico.

No obstante, el objetivo de mayor relevancia persigue la mejora de la calidad de vida de las personas mayores. Este proyecto busca abordar el uso de la tecnología Touché de manera no intrusiva al sensorizar elementos como sillas o camas, que los usuarios apenas percibirán. El enfoque principal es la prevención de riesgos y, si fuese necesario, la generación de alertas en caso de situaciones inusuales en el hogar. Uno de los objetivos clave es enviar notificaciones a un usuario si no se ha registrado cierta actividad en el asiento o la silla (u otro elemento sensorizado) en un período de tiempo específico. Esto permite prevenir accidentes o, en caso de que ya hayan ocurrido, brindar asistencia de manera eficaz y rápida.

Para ejemplificarlo un poco, podemos considerar el caso de una persona mayor que normalmente se levanta entre las 8:00 y las 9:00 de la mañana. Si Touché detecta que la persona sigue acostada más allá de este horario, enviará un mensaje de alerta al familiar o a la persona que elijamos. De esta manera, se puede monitorear las actividades cotidianas de una persona mayor de manera discreta y efectiva, contribuyendo así a su independencia y seguridad.

1.2.2. OBJETIVOS ESPECÍFICOS

Para alcanzar los objetivos generales de este proyecto definidos anteriormente, nos marcaremos una serie de objetivos específicos destinados a mejorar la tecnología Touché:

1. Experimentar con la tecnología Touché para investigar y demostrar sus ventajas, conocer los límites de esta tecnología, y evaluar su potencial.
2. Optimización de la tecnología básica Touché: tras conseguir el objetivo básico de predecir actividades, el siguiente paso será optimizar variables como el tiempo de barrido, para que la interacción de la persona responda lo más rápido posible, esto se podría lograr por medio de barridos de frecuencia selectivos, es decir en donde

no se realiza un barrido completo, sino centrados a frecuencias con mayor probabilidad de detección. Otro punto por mejorar, del cual trataremos en el proyecto, es el de poder detectar no solo los movimientos o gestos que hacemos, sino qué persona lo está haciendo, esto nos interesa, sobre todo, por ejemplo, si vamos a implementar Touché en un hogar donde viven varias personas y queremos poder identificar qué persona está sentada en una silla, o tumbada. Esto no solo nos permitiría tener una interfaz de usuario táctil, sino que podríamos tener una interfaz táctil válida para muchos usuarios.

3. Plantear e investigar alternativas al circuito básico Touché: Hasta ahora hemos presentado a Touché como un circuito capacitivo, que opera en un rango de frecuencias. Hay que tener en cuenta de que la capacidad en cuestión se forma al acercar el dedo o mano al objeto conductor conectado al circuito. Debido a que la impedancia de un circuito capacitivo depende en gran parte de la frecuencia, este barrido puede darte mucha información. Sin embargo, también se puede trabajar con inductancia, cuya impedancia varía con la frecuencia, aunque de forma muy diferente. Uno de los objetivos específicos será investigar más acerca de los posibles usos de la inductancia, e investigar maneras de variar la misma para hacer Touché sensible a cambios en un objeto de una manera inductiva y no capacitiva.
4. Definir los posibles casos de uso o aplicaciones de la tecnología resultante: Hasta el momento, hemos enfocado los objetivos de Touché a la comodidad y a mejorar la calidad de vida de las personas mayores, pero este diseño puede dar pie a muchas otras aplicaciones. Por ejemplo, para controlar dispositivos remotos por medio de gestos.

1.3 BENEFICIOS DERIVADOS DEL DESARROLLO DEL PROYECTO

La población global está experimentando un aumento significativo en la proporción de personas mayores, y este cambio demográfico plantea desafíos importantes en términos de atención médica, seguridad y bienestar en el hogar. En este contexto, la tecnología Touché, que se basa en la detección de gestos y toques en superficies, se presenta como una solución innovadora y altamente prometedora para mejorar la calidad de vida de las personas mayores en sus hogares. A continuación, se enumeran varias razones fundamentales que respaldan la implementación de esta tecnología:

- Mejora de la autonomía y la seguridad: La tecnología Touché permite a las personas mayores realizar diversas tareas cotidianas, como encender y apagar electrodomésticos, ajustar la iluminación y controlar la temperatura, mediante gestos simples. Esto reduce la dependencia de la asistencia constante y promueve la independencia, al tiempo que disminuye el riesgo de accidentes en el hogar.
- Facilita el monitoreo de la salud: La capacidad de Touché para rastrear movimientos y comportamientos en el hogar puede utilizarse para supervisar la actividad física y el bienestar de las personas mayores. Esto permite a los familiares y cuidadores estar al tanto de cambios significativos en la rutina diaria y responder de manera proactiva a problemas de salud emergentes.
- Fomenta el uso de una tecnología no intrusiva: La tecnología Touché se puede integrar en todo tipo de inmobiliarios u objetos cotidianos de una manera en la cual apenas se perciba la sensorización de estos medios, de esta forma hace que el hogar se sienta seguro, fomentando la privacidad del usuario.
- Adaptabilidad y personalización: Touché permite la personalización de la tecnología según las necesidades y preferencias individuales, lo que es esencial en el cuidado de personas mayores, ya que las capacidades y limitaciones pueden variar significativamente de un individuo a otro.
- Reducción de costos en atención médica: Al facilitar el envejecimiento en el hogar y proporcionar una mejor supervisión de la salud, Touché puede ayudar a reducir la carga financiera en sistemas de atención médica y permitir que las personas mayores disfruten de una vida más plena en su entorno familiar.

1.4 ESTRUCTURA DEL PROYECTO

En esta sección realizaremos un breve resumen de los apartados de este proyecto:

- Estado de la técnica: Este capítulo enfatizará en el contexto histórico, así como en la evolución de los sensores táctiles. Profundizará acerca de los fundamentos de Touché, y ofrecerá una explicación detallada de la detección capacitiva con barridos de frecuencia (SFCS¹). También expondrá las tecnologías que se usan en la actualidad y qué soluciones pueden dar a la necesidad o problemática planteada.

¹ SFCS: Swept Frequency Capacitive Sensing

Por último, hablaremos de que limitaciones tiene la tecnología que se usa hoy en día.

- Metodología: Este capítulo se centrará en los fundamentos teóricos que sustentan el proyecto; las herramientas, tanto a nivel software, como hardware empleados, y el procedimiento y técnica de recolección de datos.
- Desarrollo o diseño: presentará en detalle la solución propuesta, profundizando más acerca del diseño e implementación de la tecnología Touché. Por último, expondrá las pruebas y validaciones realizadas, las cuales presentarán una serie de resultados, que se irán analizando durante el apartado.
- Discusión: Este capítulo expondrá los puntos clave identificados durante el desarrollo de este proyecto, que se compararán con otras soluciones existentes parecidas a Touché.
- Conclusiones: Como parte final del proyecto, se hará una serie de reflexiones generales sobre lo aprendido y logrado. Además, se presentará un estudio sobre las limitaciones de esta tecnología y de las soluciones encontradas. Por último, se expondrá un apartado de recomendaciones o sugerencias para futuros trabajos.
- Anexos: Incluyen información adicional relevante del proyecto, así como todos los programas realizados y cálculos teóricos de las diferentes etapas del circuito.
- Bibliografía: Enunciaremos todo el material de referencia que nos haya permitido obtener información para la realización del proyecto (artículos, libros, fuentes biográficas...)

2.ESTADO DE LA TÉCNICA

2.1. HISTORIA Y EVOLUCIÓN DE LOS SENSORES TÁCTILES

Esta sección pretende contextualizar la historia de la tecnología táctil, centrándose en particular en las pantallas y sensores táctiles.

La idea de las interfaces táctiles surgió de la necesidad de controlar el tráfico aéreo, en el centro de investigación 'Royal Radar Establishment' de Malvern, en Reino Unido. Con los radares necesitaban una comunicación fluida, para poder marcar puntos de manera precisa y reaccionar rápidamente ante las amenazas de los aviones enemigos.

En 1965 Eric Johnson, uno de los ingenieros del centro, llegó a la conclusión que una pantalla que pudiera tocarse podía mejorar la capacidad de respuesta de los trabajadores. Es aquí donde nace el concepto de pantalla táctil. (ver Línea Temporal 1)

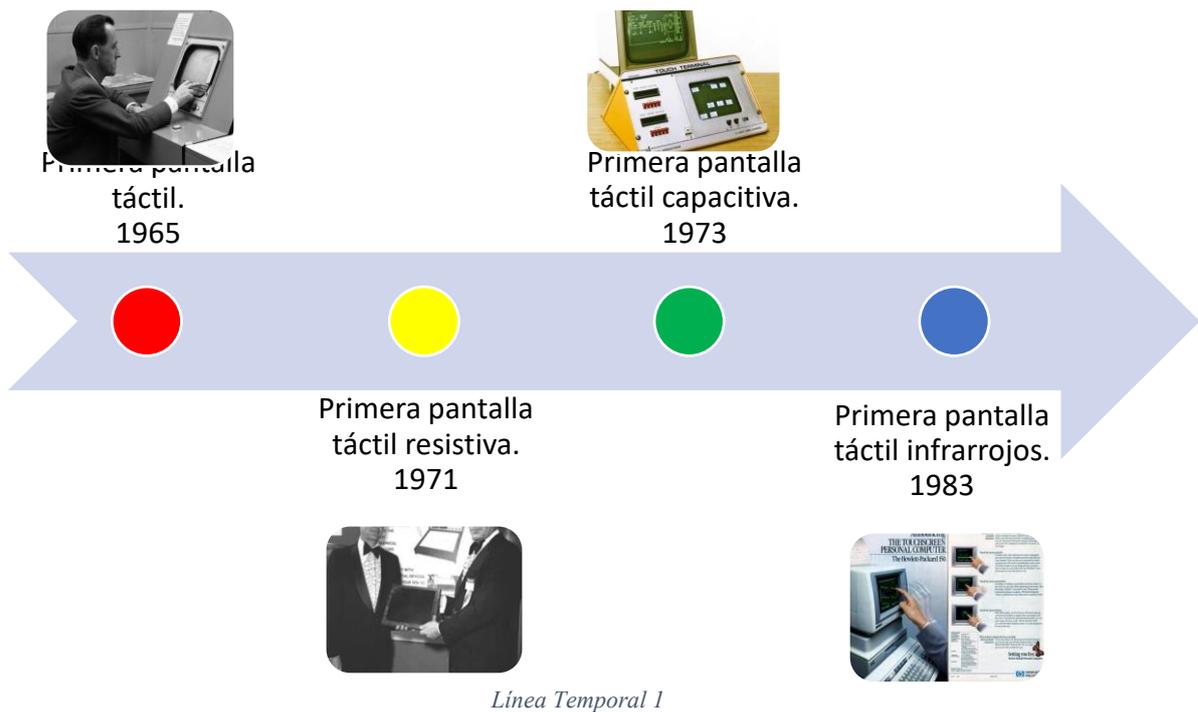
En 1971 el físico George Samuel Hurst descubrió el primer sensor táctil. Hurst creó el “Elograph”, una especie de sensor capaz de leer la posición de los ejes X e Y. Al presionar sobre la cubierta se permitía que el voltaje fluyera entre los cables de cada eje y poder medirse para indicar la coordenada exacta, es aquí donde nace la primera pantalla resistiva. (ver Línea Temporal 1)

A partir de ese punto, esta tecnología experimentó una evolución constante, hasta llegar al tipo de sensor táctil en el cual se fundamenta este proyecto. En 1973 dos ingenieros del CERN, Franck Beck y Bent Stumpe, desarrollaron la primera pantalla táctil capacitiva (Figura 2). La pantalla táctil disponía de un número fijo de botones programables. Cada zona estaba enlazada a un condensador, de forma que, al colocar el dedo encima, la capacidad aumentaría de manera significativa y el sistema lo detectaría (se añadió un recubrimiento de laca para evitar que los dedos tocasen directamente el condensador). Con la llegada de los ordenadores modernos, estas pantallas pudieron expandirse a más dispositivos hasta finalmente llegar a los smartphones (ver Línea Temporal 1)



Figura 2: Primera pantalla táctil capacitiva

Como etapa final de este contexto histórico, en el año 1983, Hewlett-Packard lanzó un PC con pantalla táctil para uso comercial, esta pantalla detectaba la posición mediante infrarrojos, el problema es que era capaz de detectar cualquier objeto que no fuera transparente, por lo que provocó múltiples problemas de usabilidad. (Línea Temporal 1)



Si observamos nuestro entorno, es evidente que prácticamente todo se gestiona a través de sensores táctiles: teléfonos móviles, sistemas de navegación GPS en vehículos, dispositivos PDA utilizados para registrar pedidos en bares o restaurantes, entre otros ejemplos. Tanto es así, que este uso cotidiano de la tecnología táctil lo usamos desde que apagamos el despertador del móvil con un dedo por la mañana, o encendemos la tele antes de dormir. La tecnología táctil, no es más que la entrada de datos de un sistema a otro a través de un medio externo como es el pulsar una pantalla, o un botón táctil [1]. Debido a esta necesidad de interacción con un entorno como puede ser una pantalla táctil, la evolución continua y exponencial en este campo ha permitido desarrollar y afinar la tecnología táctil hasta llegar a lo que tenemos hoy en día [2]:

- Sensores táctiles resistivos: Estos sensores llevan usándose más tiempo que los sensores capacitivos. Son verdaderamente útiles puesto que no depende de la propiedad eléctrica de la capacitancia. Estos sensores lo que hacen es detectar la presión que se está ejerciendo en la superficie. (Figura 3: Esquema de un sensor táctil resistivo)

Un sensor táctil resistivo consta de dos capas conductoras separadas por pequeños espaciadores. La capa inferior suele ser de vidrio o película, mientras que la capa

superior está hecha de película transparente. Se aplica un voltaje a través de la capa conductora. Cuando se aplica presión sobre la película superior con una sonda como un dedo o un lápiz óptico, esta se flexiona hacia adentro y entra en contacto con la capa inferior, lo que produce una caída de tensión. El punto de contacto crea una red de división de tensión en las direcciones X-Y. Un controlador detecta este voltaje y los cambios en el voltaje para calcular la posición del contacto basándose en las coordenadas X-Y del toque.

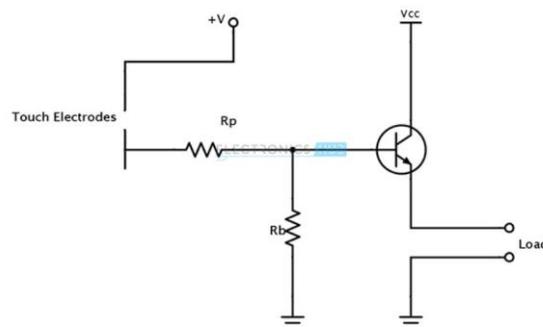


Figura 3: Esquema de un sensor táctil resistivo

- **Sensores táctiles ópticos:** Este tipo de sensores responden a cambios en la intensidad de la luz, estos cambios pueden ser medidos debido a la presión, fuerza o el tacto. El funcionamiento de este circuito consiste en que una fuente de luz envía continuamente luz a un fotodetector, al ejercer cierta fuerza o presión sobre una superficie se produce un cambio en la intensidad de esta luz, este cambio de luz recibida por el fotodetector se muestra o representa virtualmente como una imagen utilizando un circuito especial. (Figura 4)

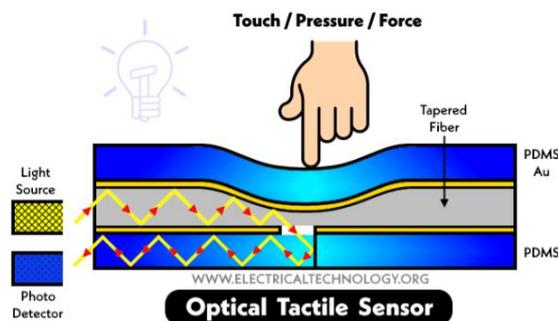


Figura 4: Funcionamiento de un Sensor Táctil Óptico

- **Sensores táctiles capacitivos:** Son ampliamente reconocidos como uno de los tipos de sensores más predominantes en la actualidad. Fundamentalmente, este tipo de sensores se rige por los mismos principios subyacentes que se emplean en el

proyecto. Su funcionamiento se sustenta en la capacidad de discernir variaciones en la capacitancia, resultantes del contacto físico, la aplicación de presión o fuerza externa. Dichas acciones conllevan a la alteración de la distancia entre las placas de un condensador asociado al sensor. Cuando se detecta la presencia de contacto, presión o fuerza, se genera una representación virtual de tales acciones como resultado del procesamiento de los datos derivados de dichos eventos. (Figura 5)

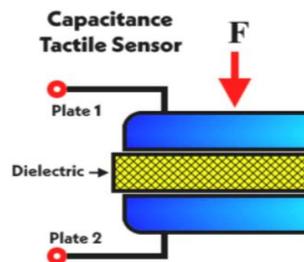


Figura 5: Funcionamiento de un sensor capacitivo

- Sensores táctiles biométricos: Para poder poner en contexto, estos sensores son usados para los lectores de huellas dactilares principalmente (Figura 6). Se usan sobre todo en ámbitos que requieran protección y seguridad ya que son hipersensibles a cambios en su entorno de contacto, pudiendo diferenciar rasgos únicos de una persona como puede ser la biometría de un dedo. Esto podría ser una solución factible a nuestro proyecto, pero tal nivel de precisión hace muy costoso esta implementación en comparación a un sensor capacitivo, el cual no tiene tanta exactitud, pero se abarata mucho más y es capaz de distinguir gestos y acciones del contacto humano. Estos sensores tienen una gran variedad de modelos, cada uno capaz de medir mediante distintas técnicas la biometría de una persona, aquí muestro los tipos:

-Óptico reflexivos (Figura 7: Sensor de huellas dactilares Modelo de un sensor óptico reflexivo)

-Óptico transmisivos

-Capacitivos

-Mecánicos

-Térmicos

-De salida dinámica (Estos permiten identificar aparte de la biometría, si un dedo o cierta parte del cuerpo se desplaza a lo largo del sensor)



Figura 6: Sensor de huellas dactilares

An optical sensor.

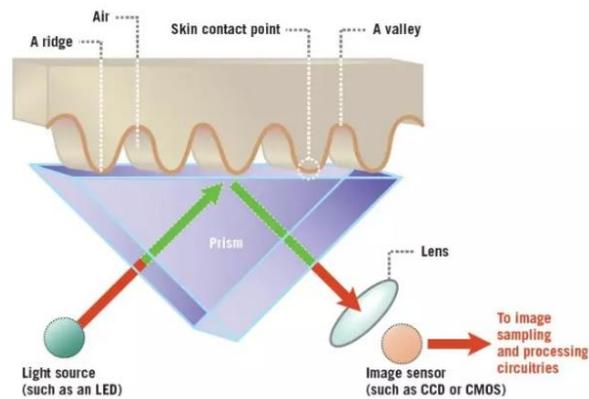


Figura 7: Modelo de un sensor óptico reflexivo

Con el paso del tiempo y el continuo desarrollo de la tecnología táctil, se ha generado la necesidad de mejoras sustanciales en esta área. En este contexto de expansión de la tecnología táctil, se ha dado origen a la tecnología Touché a través de investigaciones y mejoras en los sensores táctiles capacitivos, convirtiéndose en una solución innovadora. Si bien la tecnología táctil convencional ofrece ventajas notables, enfrenta un desafío significativo: su capacidad se limita a la detección del contacto o la no detección de este. En contraste, la tecnología Touché va más allá al no solo detectar la presencia del contacto, sino también la forma en que se establece dicho contacto, e incluso los gestos específicos realizados durante la interacción.

Para ilustrar mejor esta idea, consideremos un ejemplo práctico. En muchas situaciones cotidianas, como poner el teléfono móvil en silencio o bloquear una puerta, normalmente implican una secuencia de pulsaciones en un dispositivo o la manipulación de un objeto físico. Por ejemplo, para silenciar un teléfono, generalmente se requiere encender el dispositivo, introducir una contraseña y luego activar el modo de silencio, todo mediante una serie de pulsaciones. ¿Qué sucedería si pudiéramos simplificar todo este proceso mediante un gesto de la mano, simplemente acercando el dedo índice de nuestra mano a la comisura de los labios? Del mismo modo, ¿qué pasaría si pudiéramos bloquear una puerta simplemente tocando su pomo de una manera específica o abrir la puerta con un toque ligero, todo sin necesidad de utilizar botones? La tecnología Touché hace posible estas acciones al interpretar el contacto de la piel con el objeto en cuestión, ofreciendo una experiencia más intuitiva y eficiente.

2.2 FUNDAMENTOS DE LA TECNOLOGÍA MEDIANTE BARRIDO EN FRECUENCIA (TECNOLOGÍA TOUCHÉ)

El sistema Touché representa un tipo de sensor táctil capacitivo que se basa en una modalidad de detección llamada Swept Frequency Capacitive Sensing (SFCS), traducida al español como "Detección Capacitiva por Barrido de Frecuencias". En este enfoque, similar a los sensores táctiles capacitivos convencionales, en los cuales se aplica una señal eléctrica para excitar un objeto conductor y se supervisa la señal de retorno, la cual experimentará cambios cuando el objeto conductor entre en contacto o se aproxime a la piel humana. Estas variaciones se deben a cambios en la capacitancia del circuito. No obstante, la característica distintiva reside en que los sensores táctiles capacitivos convencionales generan una señal a una frecuencia determinada, esto limita mucho la tecnología puesto que únicamente reconocerá si estás tocando o no algún objeto sensorizado. Sin embargo, el circuito de Touché emplea un rango de frecuencias, es decir, durante un periodo de tiempo, el cual se llama tiempo de barrido, se van haciendo pasos de frecuencia hasta completar un determinado intervalo de frecuencias preestablecido. Debido a esto Touché aporta una cantidad sustancialmente mayor de información, puesto que no se restringe únicamente a los datos proporcionados por una frecuencia específica, sino que posibilita la obtención de datos de todas las frecuencias del barrido. De este modo, se logra una capacidad ampliada para interpretar la manera en que se toca un objeto o el gesto que se está realizando [3].

A través de un sistema de clasificación, se puede anticipar en función de nuestros movimientos la naturaleza de nuestra interacción con un objeto; por ejemplo, si lo estamos sujetando con la mano cerrada o simplemente estamos tocando con un dedo. Este método supera a los sensores táctiles capacitivos convencionales debido a la capacidad de obtener información detallada sobre la forma en que se interactúa con un objeto.

Si lo queremos comparar con los tipos de sensores táctiles que existen actualmente, Touché presenta varias ventajas respecto a cada uno:

- Sensores Resistivos: Touché ofrece un tiempo de respuesta más rápido, y además permite identificar qué tipo de gesto o interacción está realizando sobre el objeto sensorizado.

- Sensores Biométricos: Si bien es capaz de diferenciar gestos con alta precisión, su costo es considerablemente más elevado, lo cual hace muy difícil extender su uso a entornos domésticos. Por el contrario, la intención de Touché es que sea un método de sensorización asequible para cualquier hogar.
- Sensores Ópticos: Podrían representar una alternativa viable para competir con Touché. Estos sensores se basan en la variación de un haz de luz debido a la deformación causada por el contacto físico con una superficie. Teóricamente, podrían medir cómo se toca un objeto. Sin embargo, existen desafíos significativos. Los sensores ópticos requieren supervisión continua, ya que la acumulación de suciedad en la lente puede afectar negativamente los resultados. Además, su consumo de energía es superior; mientras que un sensor óptico promedio consume alrededor de 200 mW, un sensor capacitivo convencional consume aproximadamente 20 mW, y no nos olvidemos que Touché se fundamenta en principios capacitivos, y en última instancia, Touché no solo es sensible al contacto con el objeto excitado por una corriente, sino también a los gestos cercanos al objeto. Por el contrario, el sensor óptico requiere un contacto directo con el objeto para detectar la variación en la intensidad de la luz causada por la deformación de la superficie.

Todas estas comparativas nos demuestran una vez más que la tecnología Touché es mejor para la implementación en un hogar. No sólo es barato y nos permite conocer mucha información de las acciones realizadas en un entorno, sino que añadir sensorización, al tratarse de principios capacitivos, puede ser tan simple como colocar un cable en una silla, o una cama, ya que el acercamiento de la piel con el cable hará que varíe la capacitancia del circuito. Esto sin duda hace que el costo únicamente resida en la parte electrónica del circuito, no como los otros tipos de sensores táctiles, que aparte de esto requerirá costes en los propios sensores.

2.3. TECNOLOGÍAS Y SOLUCIONES EXISTENTES

Ya hemos descrito a términos generales en qué se basa y permite la tecnología de Touché, pero no es la única tecnología capaz de poder identificar interacción con algún medio a través de una interfaz táctil, existen varias tecnologías bastante avanzadas las cuales pueden dar pie a competir con Touché:

- Pantalla táctil capacitiva proyectada: Se basa en la capacidad de los humanos para actuar como un conductor eléctrico. Estas pantallas tienen una capa de capacitancia que puede almacenar electricidad. Cuando un dedo se acerca a la pantalla, cambia la cantidad de carga almacenada en esa capa de capacitancia, lo que a su vez altera la corriente eléctrica del dispositivo. Los cambios en esta corriente son registrados por el dispositivo y se interpretan como una entrada de usuario. Es un tipo de tecnología multitáctil, por lo que pueden soportar múltiples puntos de contacto a la vez, lo que permite gestos más complejos. (Figura 8) [4]



Figura 8: Pantalla táctil capacitiva proyectada

- Cámaras 3D y sensores de profundidad: Es un tipo de tecnología muy reciente, la cual se conoce más comúnmente como Cámaras TOF 3D. El acrónimo TOF, quiere decir “Time of flight”, es decir, Tiempo de Vuelo. Este tipo de cámaras funciona con una tecnología parecida a la del sonar o radar de un submarino. Se utilizan para medir con precisión la distancia a la que se encuentra un objeto proyectando hacia él un haz de luz infrarroja. Cuando la luz incide sobre la superficie del objeto, una parte de ella se refleja en sentido contrario al que se ha propagado inicialmente y regresa hacia la cámara. El tiempo con el que se emite el pulso de luz y regresa se puede medir, al conocer el tiempo y la velocidad (ya que la velocidad de la luz es alrededor de 300.000Km/s), podemos calcular la distancia a la que se encuentra el objeto con precisión. La cámara posee una serie de fotorreceptores, los cuales mediante esta información es capaz de generar una matriz de puntos, los cuales se pueden escalar a poder mapear rangos de profundidad y reconstruir una imagen tridimensional del objeto. Esta tecnología puede ser capaz de detectar diferentes modos de gesticulación, o interacción con un entorno. (Figura 9) [5]



Figura 9: Cámara TOF 3D

2.4. LIMITACIONES Y DESAFÍOS DE LAS SOLUCIONES ACTUALES

Pese a las dos soluciones presentadas en el apartado anterior cabe destacar que hemos optado por utilizar Touché ya que para la finalidad en la que lo queremos emplear es mucho más efectivo y asequible. Aquí presentaremos las limitaciones que tiene cada tecnología respecto con la tecnología Touché:

- Limitación de la pantalla táctil capacitiva proyectada: Esta tecnología es muy similar a Touché en cuanto a que se trata de un tipo de sensor que usa principios capacitivos para captar un gesto. Estos sensores son multitáctiles, es decir, puede recoger información de varios toques en la pantalla o dispositivo asociado a este sensor, recoger los datos e interpretarlos de una forma u otra. Bien, hasta aquí parece que podría detectar gestos. La limitación aparece con las interacciones que incluyan alejarse del objeto sensorizado, ya que para que este tipo de pantalla detecte el tipo de gesto realizado, debe haber un contacto directo, lo cual Touché permitiría detectar un movimiento que no esté en contacto con el objeto, pero si cerca de este. Otro inconveniente que viene un poco de la mano del anterior es que es una tecnología que sólo permite operar con guantes finos (como el látex). Con guantes gruesos o no conductores la detección táctil se hace muy difícil.
- Limitación de cámaras 3D y sensores de profundidad: Esta tecnología se usa principalmente en smartphones de gama alta, esto hace que los costes sean grandes, puesto que es una tecnología novedosa, que tiene una alta precisión y que requiere un sistema capaz de modelar todos los puntos de la matriz en un espacio tridimensional. Esto para un hogar es muy difícil de implementar por dos razones:

- Tecnología cara e intrusiva: Lo que se pretende con este proyecto es encontrar una tecnología domótica capaz de poder ser instalados en domicilios, para mejorar la calidad de vida de las personas mayores, de forma que apenas perciban la sensorización de los objetos del hogar y que sea lo más asequible posible.
- Tecnología aparatosa: Cuando consideramos la necesidad de sensorizar múltiples objetos, como un sofá, una puerta, una cama o un frigorífico, es importante abordar la eficiencia y la economía de la solución. La implementación de una cámara con capacidad para recrear movimientos tridimensionales en cada uno de estos objetos resultaría costosa y poco práctica en términos de comodidad. Sin embargo, la tecnología Touché, que detecta cambios en la capacitancia, ofrece una alternativa más eficiente. En este enfoque, solo se requiere una placa Touché conectada a un cable para sensorizar el elemento del hogar deseado. Esto reduce significativamente los costos, ya que el sensor en sí mismo es muy barato, y proporciona una solución más práctica, cómoda y asequible.

3. METODOLOGÍA

En este capítulo se describe el modelo teórico, las herramientas software y hardware utilizadas, así como las técnicas de recolección de datos.

3.1 MODELO TEÓRICO

3.1.1. TOUCHÉ CAPACITIVO

Hemos planteado previamente que Touché se fundamenta en la detección de los cambios de capacitancia y su respuesta en frecuencia a través de un barrido de frecuencia. Para comprender en profundidad el funcionamiento de Touché, es esencial familiarizarse con los principios teóricos que modelan su comportamiento.

La operación de Touché se basa en la generación de una señal senoidal a una frecuencia fija durante un intervalo predefinido para permitir la estabilización y medida de la respuesta del circuito. Concretamente se mide la envolvente positiva de la salida del circuito. Este proceso de medida se repite para las siguientes frecuencias del barrido hasta finalizarlo. La señal senoidal se conecta en paralelo a un circuito LC, que representa un circuito resonante. En este contexto, "L" denota el valor fijo de inductancia, determinado

por una bobina, mientras que "C" representa el condensador establecido entre el circuito y la proximidad de la piel del usuario. Esta capacitancia varía en función de la forma en que interactuamos con el objeto sensorizado, como, por ejemplo, al modificar la distancia de nuestra mano u otra parte del cuerpo. (Figura 10)

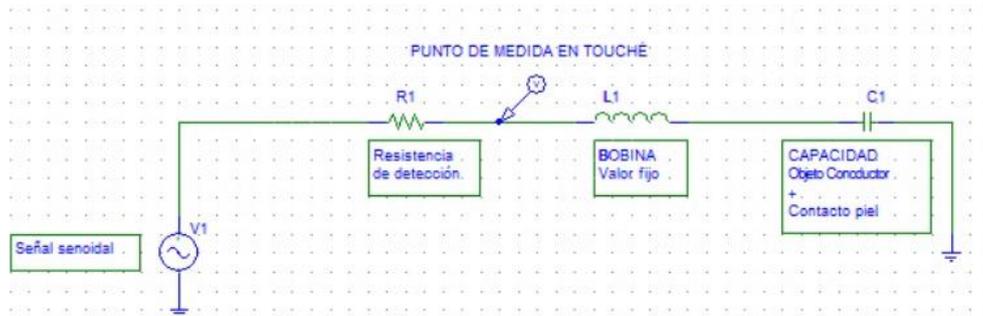


Figura 10: Simplificación de la parte RLC donde se detectan gestos en Touché.

La capacidad se calcula como:

$$C = \epsilon_0 * \epsilon_r * \frac{A}{d}$$

- ϵ_0 : Es la permeabilidad del vacío. ($8,85 * 10^{-12}$)
- ϵ_r : Es la permeabilidad del material dieléctrico. (aislante)
- A: Es el área efectiva del objeto conductor, es decir, la zona de interacción del contacto humano con el objeto conductor.
- d: Es la distancia entre el objeto conductor y la piel humana.

Donde podemos observar que a medida que toquemos un conductor de diferentes formas, irá cambiando el área efectiva y por lo tanto la capacidad de este. También detectará la distancia a la que se hagan esas interacciones con el objeto, por lo que no será necesario, ni siquiera, un contacto directo con el mismo, aunque sí una cercanía importante.

Diferentes formas de contacto conducen a diferentes valores de C, y por tanto el circuito LC entrará en resonancia a diferentes frecuencias específicas, es decir, en ciertos puntos, la impedancia del subcircuito formado por la inductancia y la capacidad se anula, lo que resulta en la formación de valles en la respuesta en frecuencia. Como podemos observar en el circuito visto anteriormente (ver Figura 10), en el punto donde indicamos como punto de medida es donde se generará los valles una vez el circuito LC entre en resonancia. Estos valles se desplazan a lo largo del rango de frecuencias a medida que la capacitancia cambia con los gestos y las interacciones.

Visto esto, profundizaremos en porqué se forma el valle en la señal. Un circuito LC, se fundamenta en lo siguiente. Tenemos dos impedancias Z y Z_C :

- $z_L = j\omega L$ (j: Parte imaginaria, w: Frecuencia angular, L: Inductancia)
- $z_C = \frac{1}{j\omega C}$ (j: Parte imaginaria, w: Frecuencia angular, C: Capacidad)

Teniendo en cuenta esto, tendremos que el circuito LC será fuertemente capacitivo a frecuencias bajas y fuertemente inductivo a frecuencias altas (Figura 11)

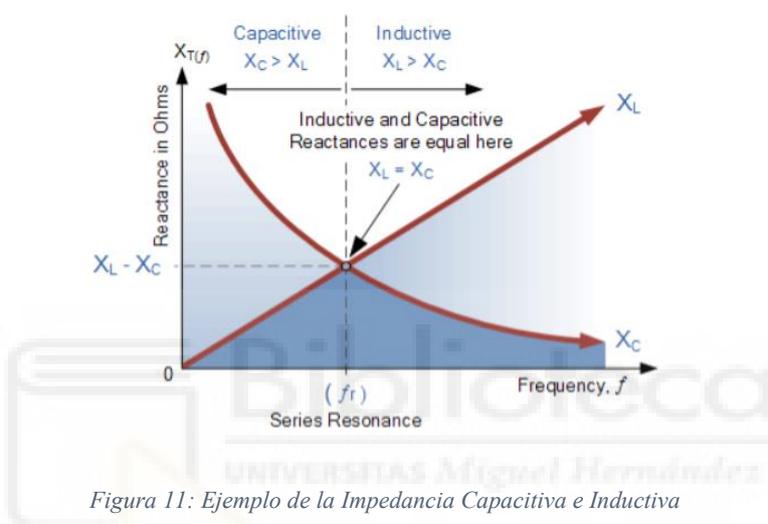


Figura 11: Ejemplo de la Impedancia Capacitiva e Inductiva

Mirando la imagen anterior (Figura 11), podemos observar que en cierta parte de la frecuencia se produce un valle en la reactancia, es decir, habrá una zona en la que las reactancias se anulen, lo que quiere decir que la bobina y el condensador se comportarán como un cortocircuito. Este valle se puede calcular, ya que conocemos que la impedancia del circuito LC es:

$$Z_{LC} = j \cdot \omega \cdot L + \frac{1}{j \cdot \omega \cdot C}$$

En resonancia:

$$Z_{LC}(\omega_0) = 0 \Rightarrow j \cdot \omega_0 \cdot L = -\frac{1}{j \cdot \omega_0 \cdot C} \Rightarrow \omega_0 = \sqrt{\frac{1}{L \cdot C}}$$

Por lo que la frecuencia de resonancia queda:

$$f_0 = \frac{1}{2 \cdot \pi \cdot \sqrt{L \cdot C}}$$

Con esta fórmula final de la frecuencia de resonancia, podemos ver que, si varía la capacidad, dependiendo de la interacción de la piel humana con el objeto conductor, el valle se desplazará a la izquierda o derecha de la gráfica.

Cabe destacar que entre el generador de la señal senoidal original y el circuito LC se coloca una pequeña resistencia de detección para formar un divisor de tensión con el circuito LC. Esto tiene como efecto mantener la tensión alta hasta que se alcance la frecuencia de resonancia, momento en el cual se atenuará la señal hasta formar un mínimo. Teóricamente, se espera que la señal disminuya a cero en ese punto debido a que la impedancia del circuito LC se vuelve nula. No obstante, en la práctica, siempre existirá cierta resistencia mínima, ya sea en la bobina o en el condensador. Esta característica es beneficiosa, ya que el objetivo de este proyecto no es reducir la señal a cero, dado que esto resultaría en una región donde la señal es nula en ciertos rangos de frecuencia. El propósito fundamental es detectar con precisión la frecuencia en la que se produce el mínimo. Para evitar la anulación total de la señal, se utiliza una resistencia de detección baja, lo que permite elevar suficientemente la señal para visualizar la respuesta en frecuencia completa en una gráfica. [6]

3.1.2. TOUCHÉ INDUCTIVO UNIVERSITAT Miguel Hernández

Hemos visto un modelo teórico de cómo funciona Touché a nivel capacitivo, es decir cuando detecta la variación de la capacidad. Si repasamos, uno de los objetivos específicos de este proyecto, era investigar cómo hacer Touché sensible a cambios en la inductancia, así vamos a repasar teóricamente si esto es posible.

Nos vamos a sustentar en exactamente lo mismo que hemos mencionado con anterioridad, la frecuencia de resonancia. Como hemos visto en la fórmula del anterior apartado, la frecuencia de resonancia depende de dos variables, la capacidad y la inductancia. Hemos visto también que la capacidad dependía de distintas variables, de las cuales, la que nosotros podíamos cambiar era el área efectiva tocando de una forma u otra, inclusive la distancia a la que hacemos la interacción con el objeto. Ahora bien, sabemos de dónde viene la capacidad, pero a nivel teórico, ¿de dónde viene la inductancia?

La inductancia es la oposición a un cambio de corriente de un inductor o bobina que almacena energía en presencia de un campo magnético. El valor de la inductancia viene determinado exclusivamente por las características geométricas de la bobina y por la

permeabilidad magnética del espacio donde se encuentra de acuerdo con las ecuaciones de Maxwell [7]. La inductancia viene determinada por:

$$L = \frac{\mu N^2 A}{l}$$

- μ =Permeabilidad magnética del núcleo, en este proyecto se usará el vacío como núcleo, es decir se deja la bobina solo con la forma del solenoide, sin núcleo en el interior. ($4\pi * 10^{-7} H/m$)
- N: Número de espiras de la bobina/solenoide.
- A: Área de la sección de la bobina. (m^2)
- L: Longitud de la bobina. (m)

Si queremos llevar estos términos a niveles prácticos debemos analizar qué variables se pueden modificar con sencillez.

En primera instancia se puede valorar la modificación de la permeabilidad magnética, y se puede pensar que al acercar o alejar la mano o cualquier parte del cuerpo estaríamos cambiando esta permeabilidad, por lo que afectaría al campo electromagnético de la bobina, pero nada más lejos de la realidad. Lo cierto es que la permeabilidad del cuerpo humano es parecida a la del aire ($4\pi * 10^{-7} H/m$), esto significa que los seres humanos no perturban significativamente el campo electromagnético de la bobina. Otra variable que entra en juego es el número de espiras, pero esto es extremadamente difícil de variar si queremos hacer cambios en la inductancia a través de la interacción humana con la bobina, por lo que se descartaría por completo, de hecho, esta magnitud será una variable fija la cual nosotros modificaremos en base a las características del diseño LC.

Ahora restan dos variables a considerar para evaluar la posibilidad de modificar alguna de ellas a fin de ajustar la inductancia. La primera de estas variables es el área de la bobina. Sin embargo, al igual que el número de espiras, resulta inviable su modificación, ya que se asume como un parámetro fijo en las especificaciones de diseño del circuito LC. Alterar este factor requeriría la capacidad de expandir o contraer físicamente la bobina en su ancho, lo cual resulta impracticable mediante acciones manuales.

Por último, nos queda la posibilidad de variar la longitud de la bobina. Desde la perspectiva de hacer que la bobina sea sensible a las interacciones con diferentes partes

del cuerpo, este enfoque también se torna complicado de modificar. No obstante, existe una posibilidad que podría cumplir un propósito específico. Como mencionamos anteriormente, uno de los objetivos específicos de Touché es la capacidad de distinguir si una persona se encuentra sentada o tumbada respecto de otra en un entorno doméstico particular.

Si consideramos esta situación desde una perspectiva capacitiva, resulta difícil distinguir si una persona está sentada en una silla o si se trata de otra persona diferente, dado que la superficie de contacto entre el cuerpo y los sensores (en nuestro caso, los cables) será prácticamente la misma (lo cual abordaremos en secciones posteriores). Sin embargo, existe otra alternativa para diferenciar a una persona de otra, que consiste en considerar el peso de la persona. Si existe una diferencia significativa de peso entre las personas, esta podría ser una forma de distinguir las.

Es importante recordar que una bobina no es más que un hilo de metal enrollado, similar al proceso de enrollamiento de un muelle. En este contexto, podemos aplicar la Ley de Hooke para los muelles en compresión [8]. Esta ley establece:

$$F = -k \cdot \Delta x$$

- F : La fuerza ejercida en el muelle al comprimirlo
- k : Constante elástica del muelle (N/m)
- Δx : La deformación del muelle ($l_f - l_i$)

Podremos calcular la longitud de la bobina en estado inicial y final, es decir, antes y después de comprimirlo, teniendo en cuenta que la fuerza depende de la masa (el peso de la persona). Para los cálculos se quita el signo negativo ya que si estamos hablando de compresión en la parte de ($l_f - l_i$), la deformación sería negativa puesto que la longitud final siempre va a ser más pequeña que la inicial así que, el signo del negativo de la constante elástica con el signo negativo de la deformación se anulan.

- $F = k \cdot \Delta x \rightarrow m \cdot a = k \cdot \Delta x$ (aceleración es $9,8\text{m/s}^2$)
- $L = \frac{\mu \cdot N^2 \cdot A}{l} \rightarrow L = \frac{\mu \cdot N^2 \cdot A}{\Delta x} \rightarrow L = \frac{k \cdot \mu \cdot N^2 \cdot A}{m \cdot a}$

En base al desarrollo de estas ecuaciones, podemos ver que, a diferentes masas, el muelle se comprimirá más o menos, por lo que variará la inductancia, esto hará que podamos distinguir qué persona está sentada o tumbada en base al peso de esta, siempre y cuando haya cierta diferencia de peso.

En resumen, la única manera viable de poder hacer Touché sensible a cambios inductivos es poder variar la longitud de la bobina. Una posibilidad es transformando dicha bobina en un muelle, por el cual habrá que tener en cuenta algún metal que una vez se comprima pueda volver más tarde a su posición inicial. El único beneficio aparente de hacer Touché inductivo es el de poder diferenciar si varía significativamente el peso, si está sentada o tumbada una u otra persona.

3.2. HERRAMIENTAS, SOFTWARE Y HARDWARE EMPLEADOS

3.2.1. HERRAMIENTAS HARDWARE

Para el diseño del circuito, así como su implementación hemos usado una serie de herramientas hardware que nos han permitido hacer pruebas con el circuito Touché. Haremos una lista de las herramientas hardware, así como una descripción y funcionalidad en el proyecto de cada uno de ellos:

- Fuente de alimentación DC ajustable: Se trata de un generador de tensión en corriente continua cuyo principal cometido será alimentar a los amplificadores operacionales del circuito a su tensión de funcionamiento de $\pm 6V$ (Figura 12). Dispone de varios potenciómetros los cuales controlan la corriente la cual será la mínima de funcionamiento, y la tensión. Dispone de dos canales de alimentación por lo que para poder generar tensiones de alimentación negativas conectaremos la tierra de un canal a la alimentación del otro con el voltaje pertinente.

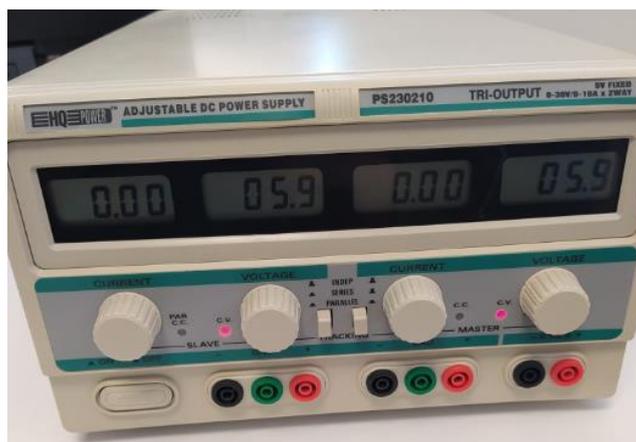


Figura 12: Fuente de alimentación DC ajustable

- Osciloscopio: Es un instrumento de visualización electrónico para la representación gráfica de señales eléctricas que pueden variar en el tiempo (Figura 13). Este aparato ha resultado útil sobre todo para hacer pruebas, poder medir la señal en el circuito RLC y ver si variaba o no la señal a medida que interactuábamos con el objeto sensorizado. Como el osciloscopio mide una amplitud determinada en un rango de tiempo, y no de frecuencias no se puede ver el valle cuando se produce la resonancia, pero si se puede ver como la onda senoidal aumenta o disminuye su amplitud al tocar de una determinada forma el objeto.

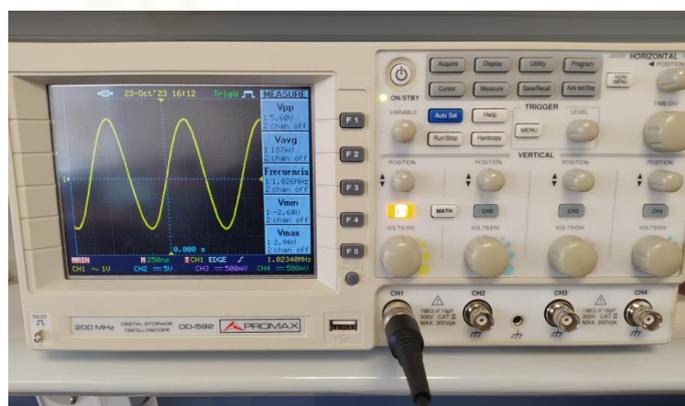


Figura 13: Osciloscopio

- Generador de señales: Es un dispositivo electrónico que permite generar señales de diferentes formas (senoidales, triangulares, cuadradas...) e introducirlas al circuito electrónico en cuestión (Figura 14). Esta herramienta ha resultado ser bastante útil en etapas de pruebas del circuito

ya que nos permitía generar una señal senoidal a la amplitud que quisiéramos (no superaba nunca los 6,6voltios), y a una frecuencia determinada. El inconveniente es que no permite realizar un barrido de frecuencias, pero podemos manualmente ir variando la frecuencia para saber si se producía un valle o no. Cabe destacar que el generador usado solo nos permitía generar señales de hasta 2MHz, así que no podríamos cubrir todo el intervalo de barrido, pero me ha resultado útil en etapas tempranas del proyecto, más tarde programamos y usamos un generador de señales variable para poder muestrear todo el barrido en pocos segundos.



Figura 14: Generador de señales

Describimos brevemente las herramientas físicas externas al propio circuito que he usado para medir, generar y variar señales. Ahora bien, este circuito electrónico requerirá de un diseño en una placa de circuito impreso o PCB. Para este diseño hemos usado la herramienta EAGLE.

- Eagle: Eagle es una herramienta de diseño de PCB que proporciona un entorno de software para crear esquemas electrónicos y diseñar placas de circuito impreso de manera eficiente y precisa. Es ampliamente utilizado por ingenieros, diseñadores y entusiastas de la electrónica para convertir conceptos de circuitos en prototipos funcionales y productos terminados. Eagle permite construir bibliotecas de los componentes, así como sus símbolos y footprints (Huella que tiene el componente en la placa), diseñar el esquemático con el símbolo de todos los componentes necesarios para finalmente, generar un archivo cuyo entorno permita poder colocar y distribuir todos esos componentes a nuestro gusto, así como el

enrutamiento de las pistas. Es importante tener en cuenta que para la fabricación de la placa de circuito impreso se deberá tener en cuenta una serie de parámetros de diseño DRC según indique el fabricante de esta [9]. El DRC indica un estándar de medidas en las pistas, enrutamientos y disposición de las capas y vías, ya que cada fabricante tendrá su propio estándar, este es el utilizado en el proyecto:

- Clearance: 10 mil.
- Distance: copper/Dimension(15 mil) y Drill/Hole(0.8mm)
- Sizes: Minimum Width(10 mil) y Minimum Drill (0.8mm)
- Annular Ring: Pads (Todo con Min:5mil, %25 y Max:10mil)

3.2.2. HERRAMIENTAS SOFTWARE

Ya hemos hablado de las herramientas hardware, tanto físicas como de diseño del propio circuito, pero esto no es todo. Para este proyecto hemos necesitado de varios programas, los cuales nos permitían programar un microprocesador que, a su vez tenía varias funcionalidades: poder programar en base al microprocesador, un generador de ondas variable que nos permita muestrear todo el barrido de frecuencias, poder implementar y adquirir datos del conversor analógico/digital, y por último en etapas finales del proyecto, poder realizar un programa de mensajería que nos permita enviarnos un correo electrónico al móvil si pasa cierta actividad en el circuito Touché a una determinada hora, la cual nosotros pudiéramos fijar.

Todo esto se ha realizado mediante un editor de código fuente llamado Visual Studio Code (Figura 15). Esta herramienta es un editor de código fuente desarrollado por Microsoft para Windows, Mac y Linux, en mi caso Windows. Es un editor de código fuente rápido y liviano que se puede usar para ver, editar, ejecutar y depurar código fuente para aplicaciones.

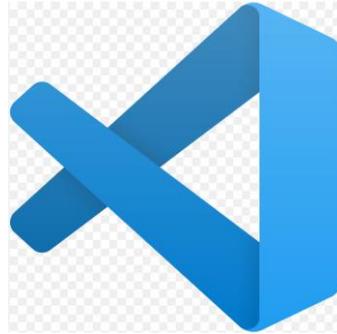


Figura 15: Símbolo de Visual Studio Code.

Visual Studio Code posee múltiples características y ventajas como:

- Soporte para múltiples lenguajes de programación: Visual Studio Code cuenta con soporte integrado en varios lenguajes. Esto hace que pueda detectar fácilmente si hay alguna falla o referencia en otros lenguajes.
- Intelli-Sense: puede detectar si algún fragmento de código queda incompleto. Además, las sintaxis de variables comunes y las declaraciones de variables se realizan automáticamente.
- Compatibilidad multiplataforma: Como hemos mencionado antes, puede funcionar para Windows, Mac y Linux, esto presenta una gran ventaja puesto que tradicionalmente, los editores solo podían funcionar en una de estas tres plataformas por lo que los códigos de software abiertos solo funcionaban en su plataforma en específico y no en las tres.
- Extensiones y soporte: Admite todos los lenguajes de programación, pero si el lenguaje de programación no es compatible, puede descargar la extensión y utilizarla.
- Repositorio: Esta conectado con Git, es decir, permite el almacenamiento seguro y oportuno para extraer o guardar las instancias.
- Proyectos múltiples: Se pueden abrir simultáneamente varios proyectos que contienen varios archivos/ carpetas, los cuales pueden estar relacionados o no entre sí.
- Comentarios: una característica común, pero algunos lenguajes no la admiten. Comentar el código ayuda al usuario a recordar o realizar un seguimiento de acuerdo con la secuencia que desee.

Estas son las características más importantes que nos ofrece Visual Studio Code (ver [10] para una descripción más completa).

Dentro de este editor de código, se tiene que elegir un lenguaje de programación, ya sea usando uno solo o compatibilizándolo con varios a la vez, en nuestro caso, la programación está realizada en C++.

C++ es un lenguaje de programación orientado a objetos que permite la creación de clases y objetos, lo que facilita la encapsulación, la herencia y el polimorfismo. Este tipo de lenguaje presenta las siguientes características:

- Extensión de C: C++ es una extensión de C, lo que significa que hereda todas las características de C y agrega características adicionales, incluida la programación orientada a objetos.
- Reutilización de Código: La programación orientada a objetos en C++ fomenta la reutilización de código a través de la herencia y la composición.
- Eficacia y Eficiencia: C++ se enfoca en el rendimiento y la eficiencia, lo que lo hace adecuado para aplicaciones que requieren control de bajo nivel y manipulación de hardware.
- Grupos de Bibliotecas de Terceros: Existen muchas bibliotecas de terceros disponibles para C++ que pueden acelerar el desarrollo de aplicaciones y ampliar sus capacidades.
- Seguridad de Tipo: C++ proporciona verificación de tipos en tiempo de compilación para ayudar a evitar errores de programación.

Estas son algunas de sus características, he optado por definir las que a mi juicio me parecen más importantes en este proyecto y por lo que he optado por este lenguaje de programación. Aparte de esto, es el entorno en el que más conocimientos tengo, por lo que me resulta más fácil programar en un ámbito en el que tengo nociones de ello.

Aparte de este programa informático, para poder hacer simulaciones de los distintos circuitos implementados, así como modificar los valores de los componentes y poder ir haciendo pruebas, he optado por usar la herramienta de simulación PSIM. (Figura 16)

PSIM es un simulador de circuitos eléctricos cuyas características son:

- Biblioteca de componentes: PSIM ofrece una amplia gama de componentes predefinidos que los usuarios pueden arrastrar y soltar en el lienzo de diseño, lo que simplifica la creación de circuitos y sistemas de control.
- Análisis en el dominio del tiempo y la frecuencia: Permite realizar análisis de simulación en el dominio del tiempo y la frecuencia para evaluar el comportamiento del circuito en diferentes condiciones.
- Generación de informes y resultados gráficos: Ofrece herramientas para generar informes detallados y visualizar los resultados de la simulación a través de gráficos, lo que facilita la interpretación de los resultados.
- Optimización y análisis de parámetros: Permite ajustar parámetros de los componentes y sistemas para la optimización del rendimiento.
- Soporte para diferentes dominios: Además de la simulación en el dominio del tiempo, PSIM también es capaz de manejar dominios como el dominio de la frecuencia, el dominio de la transformada de Laplace y otros.

Estas son algunas de las principales características, pero PSIM consta de muchas otras aparte de ser un software gratuito.



Figura 16: Símbolo de PSIM

3.3. PROCEDIMIENTOS Y TÉCNICAS DE RECOLECCIÓN DE DATOS

Ya hemos hablado previamente que para la salida del circuito Touché se utiliza un conversor analógico digital, pero aún no hemos definido el por qué.

Un conversor analógico digital, es un dispositivo capaz de convertir una señal analógica, ya sea de tensión o de corriente, en una señal digital mediante un cuantificador y codificándose en muchos casos en un código binario particular.

Esto es utilizado en el proyecto para poder recoger los datos de la salida del circuito para posteriormente analizarlos y representarlos. Se usa el conversor básicamente porque el

microprocesador sólo puede procesar señales digitales, y las que le llegan de la salida del circuito son analógicas. En este proyecto se utilizará el conversor del propio microprocesador, ya que el número de bits que ofrece para la resolución es más que suficiente, pero cabe destacar que existen conversores analógicos/digitales capaces de conectarse externamente al microprocesador con mejor resolución.

El microprocesador que se usará será una ESP32 cuyo número de bits será 12, es decir tendrá una resolución máxima de $2^n - 1$, o lo que es lo mismo, si n es 12, la resolución máxima es de 4095, esto quiere decir que el valor máximo de tensión tendrá este valor, por lo que por una sencilla regla de tres podemos pasar este valor, a un valor de tensión si sabemos que su valor máximo será alrededor de 3V. De esta forma tendremos los datos analógicos de la señal convertidos a digitales, los cuales podremos recoger, analizar e interpretar.

Hemos hablado previamente del editor de Visual Studio Code, en el cual programamos en C++, pues bien, esto enfocado al conversor lo único que hace es muestrearnos los datos recogidos en el barrido de frecuencias, para poder analizar estos datos, así como sacar la gráfica de este y ver donde se producen los valles, hemos optado por usar la herramienta Julia, dentro del mismo Visual Studio Code (Figura 17).

Julia es un lenguaje de programación de código abierto diseñado para ser rápido, flexible y fácil de usar en aplicaciones científicas y de análisis de datos. Este lenguaje de programación presenta múltiples características que son las siguientes:

- Rendimiento excepcional: Julia se destaca por su alto rendimiento, lo que lo hace especialmente adecuado para aplicaciones de análisis de datos y cálculos numéricos intensivos. Utiliza un sistema de compilación Just-In-Time (JIT) que optimiza el código en tiempo de ejecución.
- Sintaxis amigable: Julia cuenta con una sintaxis similar a la de otros lenguajes de programación populares, como Python y MATLAB, lo que facilita la transición para quienes ya están familiarizados con estos lenguajes.
- Amplia comunidad y bibliotecas: Julia cuenta con una creciente comunidad de usuarios y un ecosistema de bibliotecas especializadas para análisis de datos y cálculos científicos, como DataFrames.jl y Plots.jl.

- Interoperabilidad: Julia tiene una sólida capacidad de interoperabilidad con otros lenguajes, lo que permite utilizar bibliotecas escritas en C, Python, y R, entre otros. Esto facilita la incorporación de código existente en proyectos de Julia.
- Paralelismo y concurrencia: Julia está diseñado para facilitar el paralelismo y la concurrencia, lo que permite aprovechar eficazmente múltiples núcleos de CPU y aceleradores, como GPU, para acelerar cálculos.
- Licencia de código abierto: Julia es de código abierto, lo que significa que es gratuito para su uso y su desarrollo es colaborativo y transparente.
- Facilidad de desarrollo y prototipado: Julia permite un desarrollo rápido y eficiente, lo que lo hace útil para la prototipación y la experimentación en aplicaciones científicas y de análisis de datos.



Figura 17: Símbolo de Julia

4. DESARROLLO Y RESULTADOS

4.1. DETALLES DE LA SOLUCIÓN PROPUESTA

En el marco de este proyecto de investigación, hemos diseñado y desarrollado un circuito electrónico multifuncional llamado Touché, dividido en varias etapas, cada una con un propósito específico. El objetivo principal del circuito es la generación de una señal cuya frecuencia varíe de manera continua. Haremos un barrido de pasos de 17,5KHz hasta llegar a los 3,5MHz, por lo que tendremos 200 datos en la señal. Posteriormente, someteremos esta señal a un proceso de filtrado (filtro de paso bajo y filtro de paso alto) con el fin de eliminar posibles interferencias ambientales y eliminar el ruido que la señal pueda contener. Dado que la señal de entrada proviene de un generador de señales, su amplitud es inicialmente muy baja, lo que requiere una etapa de amplificación.

La salida del amplificador se conecta a una resistencia de salida en serie con una bobina y un objeto conductor cuya capacidad podrá variar debido a la interacción de este al contacto humano. Ambos elementos forman lo que hemos mencionado en anteriores apartados como el circuito LC. El objeto elegido permitirá que la señal se vea afectada en función de los gestos realizados con la mano u otras partes del cuerpo, lo que resulta fundamental para el propósito de este proyecto.

Para garantizar la integridad de la señal durante su procesamiento, implementaremos un seguidor de tensión que aísla eficazmente la señal. A continuación, la señal pasará por otro filtro de paso bajo (para filtrar cualquier ruido residual de la etapa de amplificación) y acto seguido por un detector de envolventes, donde la señal alterna se convertirá en continua. Finalmente, el microprocesador la podrá muestrear utilizando su propio conversor analógico a digital, completando así la secuencia de procesamiento necesaria. (Figura 18)

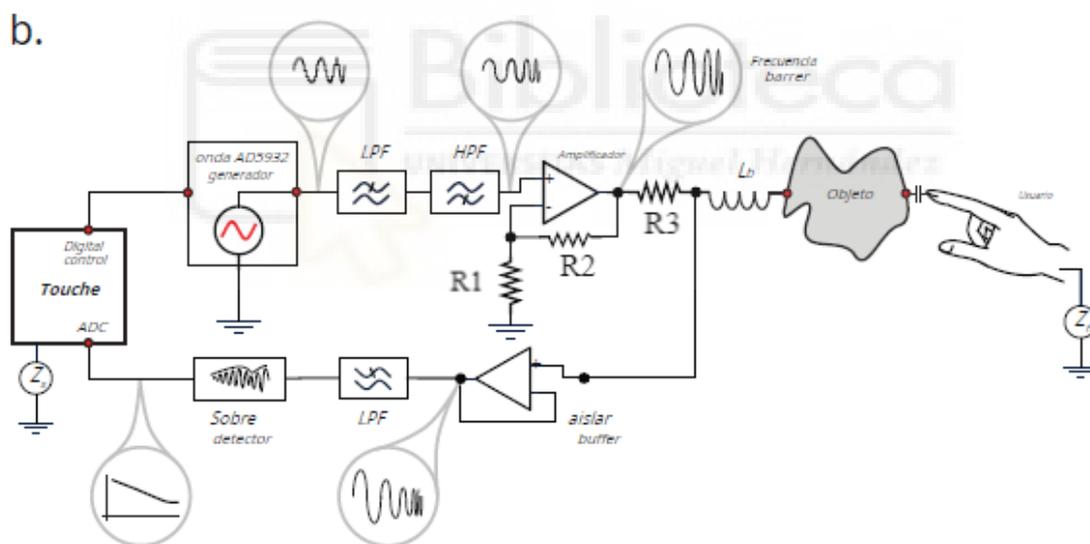


Figura 18: Esquemático del artículo Touché

En la figura anterior se muestra el esquema que nos proporciona el artículo del cual habla Touché, por lo que se ve es bastante abstracto ya que divide el circuito en bloques, por lo que mediante pruebas en los cuales ha habido ensayos y errores, hemos conseguido perfeccionar Touché.

4.2 PROCESO DE DISEÑO.

4.2.1 COMPONENTES DEL CIRCUITO SEGÚN TOUCHÉ.

Para dar inicio al proceso de diseño, nuestro enfoque inicial ha sido la exhaustiva evaluación de los componentes principales del circuito. Este procedimiento implica la recopilación de información detallada sobre cada uno de estos componentes y la determinación de cuál de ellos se adapta de manera óptima a las necesidades específicas requeridas por el circuito en cuestión. A continuación, presentamos los componentes clave del circuito junto con una descripción de cada uno, respaldada por el fundamento detrás de su elección, cabe destacar que todos los componentes mencionados son escogidos en base a un diseño inicial precedido por la idea del esquemático visto en el anterior apartado, más tarde ese diseño se irá mejorando y simplificando. Es esencial resaltar que la selección de estos componentes ha sido el resultado de un prolongado proceso de investigación, que involucró la comparación de hojas de datos (datasheets) y un análisis minucioso destinado a identificar la opción más adecuada para garantizar el rendimiento y funcionalidad óptimos del circuito:

- Microprocesador ESP32: Es el corazón de la placa Touché, dispone de un doble núcleo que funciona a 240MHz, y el voltaje a la que esta placa opera es de 3.3V. Este microprocesador será el encargado de controlar al generador de ondas programable, para que, aparte de hacer que genere una señal, sea capaz de ir variando la frecuencia para poder completar el barrido. También será la misma ESP32 la que reciba la señal de retorno mediante la habilitación de su propio conversor analógico digital (Figura 19). Con esto seremos capaces de muestrear tantos datos como se precisen, asignarle un retardo entre dato y dato e implementar todos los programas que se requieran para la realización del proyecto (como por ejemplo la implementación de un programa de aviso por la realización de cierta actividad). [11]

Para poder enviarle la información de que haga el barrido al generador de ondas, es necesario conectar ciertos pines de la ESP32 a los pines del generador, aparte claro está del propio programa (Figura 20). Estos pines son:

-Pin MOSI: Es el pin encargado de transmitir datos desde el microcontrolador hacia otros dispositivos de manera síncrona. Este pin irá conectado al SDATA del generador de ondas.

-Pin SCK: Es el pin que proporciona el reloj de sincronización para la transferencia de datos entre la ESP32 y el generador de ondas. Este pin marca la velocidad en la que se emiten y reciben los datos. Este pin irá conectado al SCLK del generador de ondas.

-Pin A5: Aunque se puede usar cualquier otro pin analógico. Será el encargado de inicializar y controlar el generador de ondas, para que este pueda emitir la señal, la frecuencia y el tipo de onda que queramos generar. Este pin irá conectado al FSYNC del generador de ondas.

-Pin A2: Este pin ya no es utilizado para manejar el generador de ondas, sino es usado para habilitarlo como conversor analógico digital, es decir, es el encargado de recoger los datos analógicos del circuito, y convertirlos en una señal digital, la cual la ESP32 pueda entender y procesar. Este pin es usado después del detector de envolvertes.

-Pin GND: Es el pin a tierra de la ESP32. Es vital para el funcionamiento de cualquier circuito electrónico, ya que nos proporciona un punto de referencia eléctrico común, elimina ruido y garantiza la seguridad eléctrica. Este pin no solo va conectado al pin AGND y DGND del generador de ondas, sino que está conectado a todas las tierras del circuito.

-Pin 3V3: Es la fuente de alimentación de la placa. Este pin aparte de alimentar y limitar la tensión de la propia ESP32 a 3,3 voltios, permite alimentar a otros componentes para su correcto funcionamiento, como por ejemplo el generador de ondas. Este pin va conectado al pin VCC del generador de ondas.

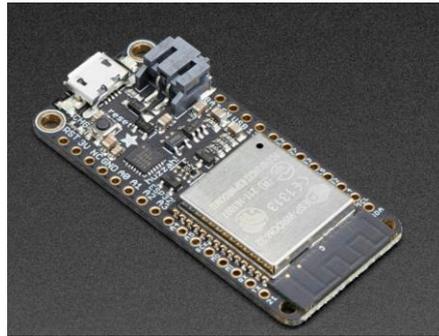


Figura 19: Microprocesador ESP32

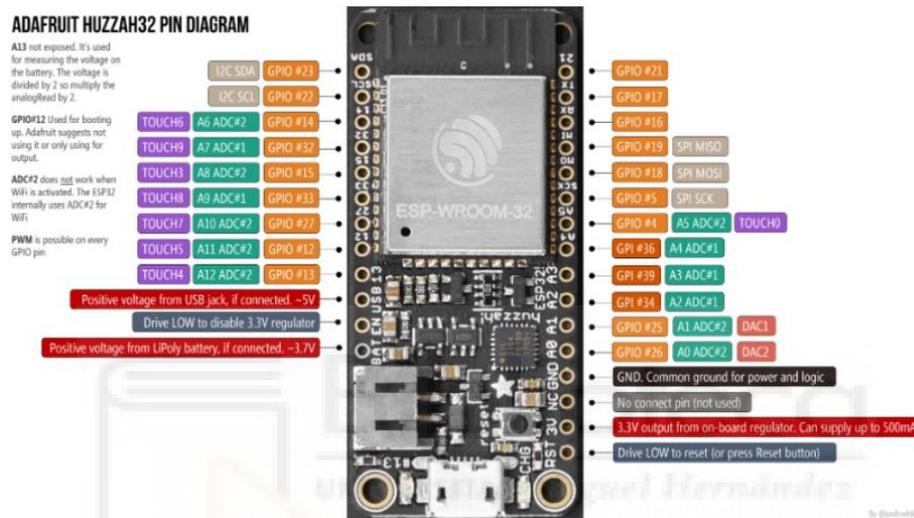


Figura 20: Pinout de la ESP32

- Generador de ondas AD9833:** Es un componente electrónico, el cual puede ser alimentado de +2,3V a +5,5V, por lo tanto, la ESP32 es capaz de mantenerlo con sus 3,3V de alimentación. Es capaz de generar ondas de 0 a 12,5MHz, y también es capaz de emitir varios tipos de ondas (senoidal, triangular, cuadrada) [12]. El funcionamiento es el siguiente: la ESP32 estará programada para decirle al generador de ondas que produzca una señal senoidal, esta señal tendrá una frecuencia variable entre 0 y 3,5MHz, con pasos de 17,5KHz, es decir, tendremos 200 pasos en cada barrido. Con esto podremos saber cómo se comporta la onda senoidal a distintas frecuencias, por lo que nos dará mucha información. El problema es que la onda tiene una amplitud de 0,6 Vpp, por lo que necesitaremos una etapa amplificadora, para amplificar la señal, poder excitar el objeto conductor y así poder detectar con más precisión los cambios en la capacidad

(Figura 21). Estos serán los pines que utilizaremos, y que hemos mencionado anteriormente:

- Pin VCC: Es el pin el cual se encarga de alimentar al generador. Recibe la alimentación directamente de la ESP32 a 3,3V.
- Pin DGND y AGND: Son pines de tierra tanto digital como analógico respectivamente. Al tratarse de un circuito analógico, y no tener ninguna parte digital dentro del propio circuito, los dos tipos de tierra se referenciarán a la tierra global del circuito Touché.
- Pin SDATA: Es un pin de entrada de datos, el cual podrá procesar datos de hasta 16 bits de entrada serie.
- Pin SCLK: Es la entrada del reloj, que está conectada al reloj de la ESP32. Los datos se registrarán en cada flanco descendente del reloj.
- Pin FSYNC: Es el pin de control del generador de ondas.
- Pin OUT: Es el pin de salida del generador de ondas, el cual nos sacará una determinada onda, con una determinada frecuencia y una amplitud de 0,6 Vpp.

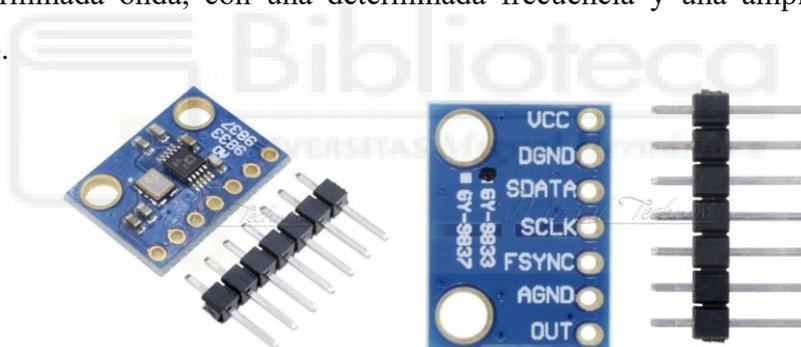


Figura 21: Generador de Ondas AD9833

- Etapa de filtrado: Es una etapa indispensable en el funcionamiento de Touché, principalmente porque la señal que genera el generador de ondas no está del todo limpia, con esto conseguiremos disminuir mucho el ruido. Inicialmente esta etapa de filtrado constaba de dos filtros, el primer filtro es un paso bajo (Figura 22), diseñado con una resistencia en serie del generador de unos 330ohm y un condensador a tierra de 100pF, el cual tenía una frecuencia de corte de aproximadamente 4,8MHz, que es más que suficiente para abarcar todo el barrido (3,5MHz de máxima), este filtro hará que atenúe la señal a frecuencias mayores que la de corte. Este filtro estaba unido a otro filtro de paso alto (Figura 23), que

estaba conectado en serie con un condensador cerámico de 22 nF y una resistencia a tierra de 10kohm, esto resultaría en una frecuencia de corte de alrededor de 720Hz, lo cual haría que atenuase toda la señal hasta que llegase a esa frecuencia de corte, esto resultó ser bastante beneficioso, puesto que las frecuencias más bajas provocaban mucho ruido.

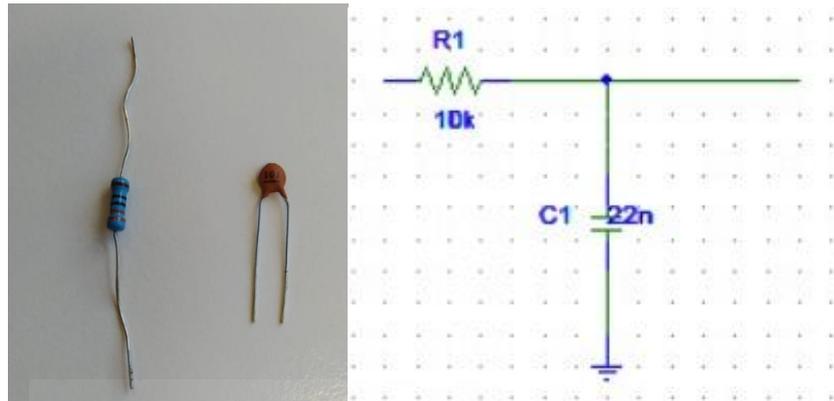


Figura 22: Componentes y esquemático de filtro de paso bajo

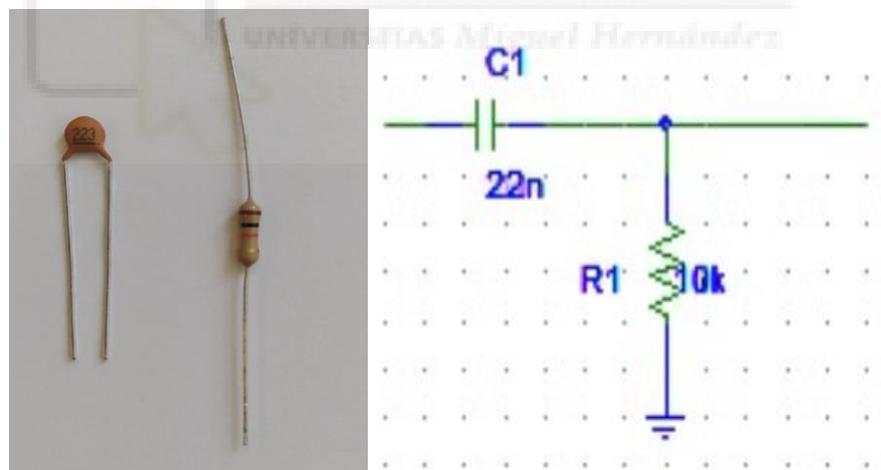


Figura 23: Componentes y esquemático de filtro de paso alto

- **Etapa Amplificadora:** Como previamente hemos mencionado, la señal de entrada tiene una amplitud de 0.6 Vpp, la cual es insuficiente para detectar variaciones significativas en la capacidad. Por lo tanto, después de filtrar la señal, se requiere amplificarla a aproximadamente 6.6 Vpp para excitar el objeto conductor. Para lograr esto, es necesario emplear un amplificador capaz de amplificar la señal

hasta al menos 3.5 MHz (la frecuencia máxima de barrido) con una ganancia de 11. (Figura 24)

Después de una búsqueda exhaustiva y pruebas de diversos amplificadores, identificamos el amplificador AD8032A como una solución adecuada. Este amplificador no solo es económicamente viable, sino que también posee un ancho de banda de 80 MHz en su configuración de ganancia 1. Al dividir ese ancho de banda por la ganancia requerida (11), se obtiene un ancho de banda teórico de 7.27 MHz, lo que indica que el amplificador debería ser capaz de amplificar la señal de manera adecuada. [13]

Para lograr la ganancia de 11 se han utilizado los siguientes componentes:

- 1 Resistencias de 100Kohm: Conectada entre la entrada inversora y la salida del amplificador.
- 1 Resistencia de 10Kohm: Conectada entre la entrada inversora y tierra del amplificador.
- 1 Chip AD8032: El cual es un chip integrado con 2 amplificadores operacionales. (Figura 25) El chip en cuestión se puede alimentar hasta $\pm 6V$, que es la tensión que se le ha suministrado para este proyecto.

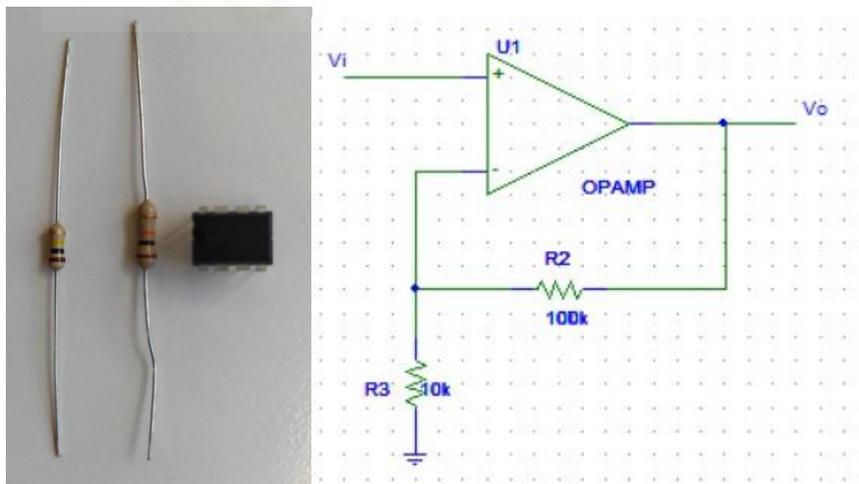


Figura 24: Componentes y esquemático del amplificador operacional

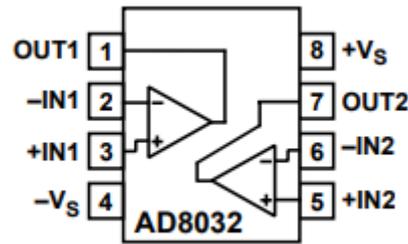


Figura 25: Pinout del chip AD8032A

- Resistencia de detección: A la salida del amplificador operacional, es indispensable colocar una resistencia de detección, cuya función es convertir la señal alterna de voltaje en una señal de corriente. Se usa esta resistencia debido a que nosotros queremos detectar caídas de tensión, inicialmente colocamos una resistencia de 10Kohm, pero como veremos en futuras pruebas esta resistencia lo que hacía era bajar mucho la señal por lo que finalmente se optamos por poner una de 470ohm.
- Circuito LC: Como hemos explicado anteriormente en los fundamentos teóricos, Touché se basa en principalmente en la detección de valles causadas por la frecuencia de resonancia, la cual variará debido a las distintas formas de interacción de la piel humana con un objeto conductor. Según el artículo Touché la bobina tiene un valor fijo de 100mH, por lo que inicialmente usamos ese valor, pero más tarde la hemos cambiado hasta perfeccionar la detección de los valles con nuestro circuito, el cual no es idéntico al del artículo (Figura 26) (Figura 27)

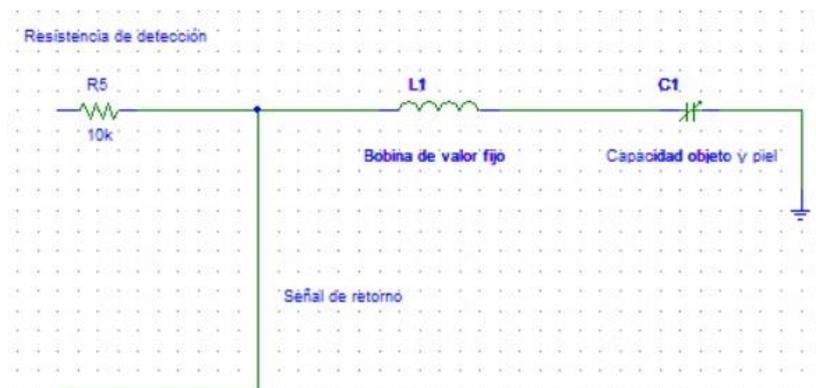


Figura 26: Esquemático del circuito LC

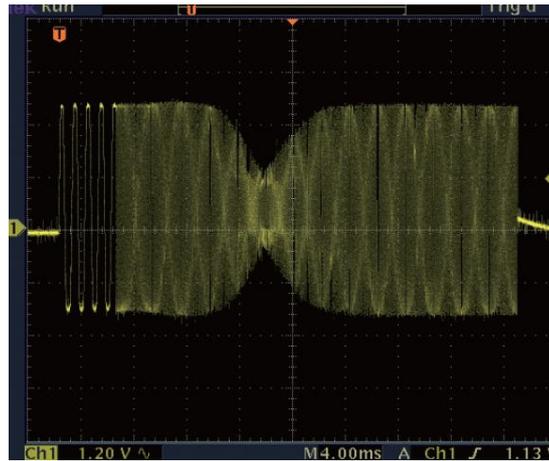


Figura 27: Señal con el valle según el artículo de Touché.

- Seguidor de Tensión: Para medir la señal de salida del circuito se usa un seguidor de tensión, es decir, un circuito con un amplificador AD8032A, que lo que hará es que a la salida exista la misma tensión en la entrada, pero con una impedancia de entrada muy alta, por lo que no supone una carga al circuito, así que aislará la señal que queremos utilizar como referencia. Cabe destacar que tiene una impedancia de salida muy baja, por lo que funciona como una fuente de tensión que sigue a la entrada sin suponer una carga para la misma (Figura 28). Esta etapa del circuito ha variado enormemente, puesto que en las pruebas de más adelante expondremos, se obtienen resultados que no son lo esperado.

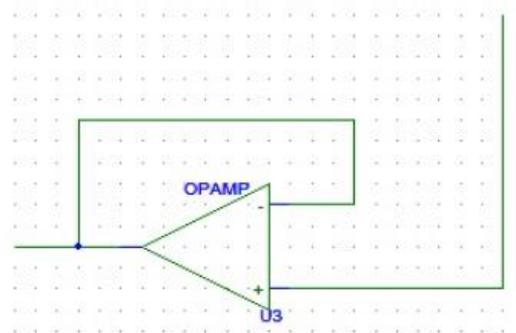


Figura 28: Esquemático de seguidor de tensión

- Detector de envolventes o rectificador de media onda: Para la parte final del circuito antes de llevarlo al conversor analógico digital, hemos diseñado un detector de envolventes, que convertirá la señal de corriente alterna en una señal de corriente continua variable en el tiempo. Para ello utilizaremos un rectificador de media onda con filtro por condensador. Un rectificador de media onda con filtro por condensador es un circuito que permite detectar solo la parte positiva de

la onda que es la que nos interesa, aparte de esto la funcionalidad del condensador es que la onda cuando baje, no vaya a cero sino que cuando esté en el valor máximo de la amplitud de la onda se descargue y continúe hacia el siguiente valor máximo, de esta forma tendremos una forma de onda continua únicamente detectando los valores máximos que es lo que nos interesa (Figura 29) [14]

Cabe destacar que para un rectificador de media onda común se usa un diodo para permitir el paso de la corriente en una sola dirección, pero en este proyecto se ha usado un transistor NPN, que tiene exactamente la misma funcionalidad que el diodo, pero aparte la señal tendrá una respuesta en frecuencia mucho más rápida, y además, también nos permitirá controlar el flujo de la corriente que circula por la base.

Esto mejorará este rectificador ya que no solo actuará como el diodo, sino que tiene una respuesta en frecuencia mucho más rápido, aparte de poder controlar la dirección de la corriente durante la media onda positiva (Figura 30).

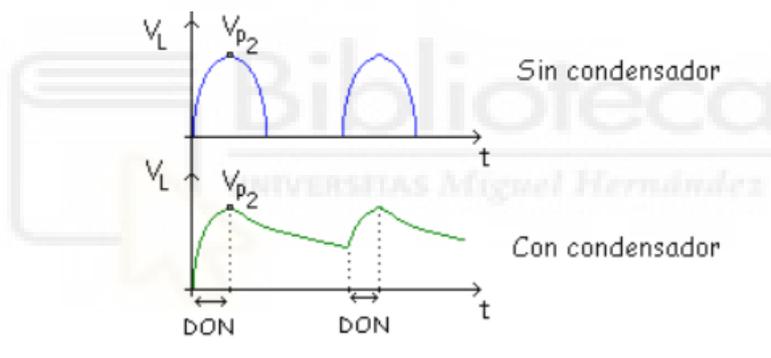


Figura 29: Descarga de un condensador en un circuito rectificador de media onda

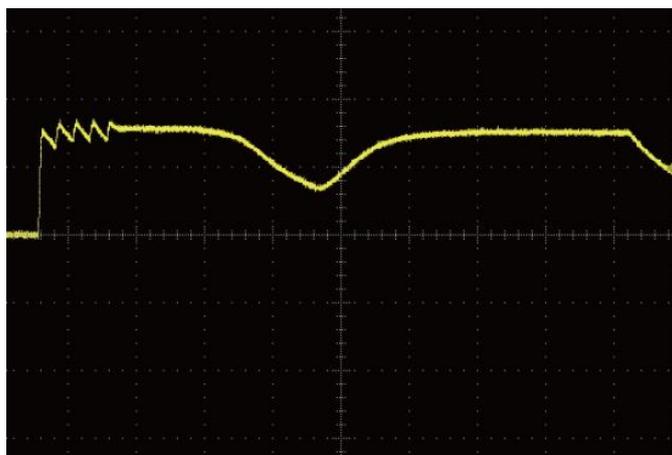


Figura 30: Señal de corriente continua al pasar por el rectificador de media onda según el artículo Tocuhé

Estos son los componentes usados en el rectificador de media onda (Figura 31):

-Transistor NPN BC546B [15]

- Condensador de 1uF
- Resistencia de 10Kohm

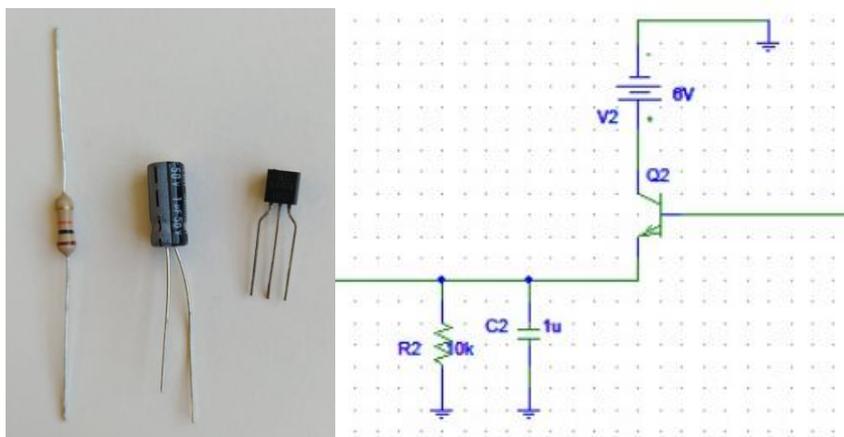


Figura 31: Componentes y esquemático del rectificador de media onda

Lo que hemos descrito anteriormente, es el diseño inicial que se ha realizado en este TFG para cada uno de los componentes de Touché según el artículo en el cual nos hemos basado. Es el primer diseño, el cual se ha visto modificado tras las pruebas que se irán describiendo en el resto de esta memoria. De hecho, el circuito descrito no funcionaba inicialmente, puesto que nos daba una señal muy pequeña, la cual no detectaba valles independientemente de la interacción que hacíamos con el objeto sensorizado, por lo que en el siguiente apartado describiremos el proceso completo del diseño de este circuito, describiendo con detalle la evolución que ha tenido dividiendo el proceso en fases.

4.2.2. EVOLUCIÓN Y FASES DEL DISEÑO DEL CIRCUITO TOUCHÉ

Este apartado lo desglosaremos en fases, las cuales explicarán el desarrollo detallado de la implementación final de Touché, y por qué hemos ido añadiendo, modificando o quitando componentes. Se explicarán los distintos diseños de PCB que se han hecho mediante la herramienta Eagle y todo esto para llegar al diseño final.

Cabe resaltar que es de vital importancia incluir esta sección sobre el proceso de ensayo y error, ya que ahí verdaderamente se ve la esencia de la investigación, del esfuerzo y de ser conocedores que, aunque se tenga una idea de qué hay que hacer, nada sale a la primera.

4.2.2.1. FASE 1: OFFSET Y RATIO

Como hemos visto en el anterior apartado, partimos de una idea original sustentada por el artículo que habla de la tecnología Touché. Desglosamos cada bloque calculando qué

componentes van para cada frecuencia de corte, qué resistencias (R1 y R2) colocar en los amplificadores para tener ganancia 11, así como ver qué resistencia y condensador ponemos en el detector de envolventes. (ver Figura 18)

En primera instancia diseñamos un programa en C++, que nos permitía generar una señal senoidal, en un barrido de 3,5 MHz con pasos de 17,5KHz. Este barrido duraba un tiempo fijo, en el cual se iba muestreando la señal para cada frecuencia tras un periodo de estabilización, a través del conversor analógico digital. Como la resolución de este es de $2^{12} - 1$, un valor de 4095 correspondería al valor máximo de resolución con una tensión de 3,3V. Cualquier valor intermedio tendría una tensión proporcional.

Para probar el circuito usamos de objeto conductor una plancha metálica de 12,5 x 9,5cm, sobre la cual se probaría distintas formas de tocar el objeto (mano, dedo, no tocar...). Al interactuar de diferentes formas con la plancha, ocurrió que gran parte de la señal tenía amplitud 0, por lo que era imposible detectar nada. Para intentar solventar este problema se cambió el seguidor de tensión, por un amplificador inversor de ganancia 1 con un offset de 1 V. Esto lo que haría es elevar la señal alrededor de 1V para que se pudiera ver todos los cambios relevantes en la señal (Figura 32).

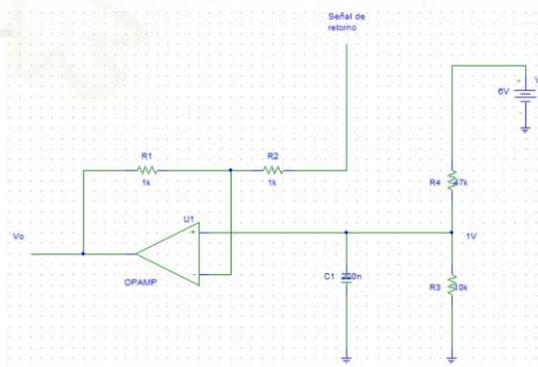


Figura 32: Circuito amplificador de ganancia 1 con offset de 1 V

Implementado esto, cabe destacar que, en el apartado anterior vimos que a la salida del amplificador operacional se conectaba una resistencia de detección y acto seguido va la señal de retorno en paralelo con el circuito LC. Idealmente la señal en la salida del amplificador no debería de cambiar conforme aumenta la frecuencia ya que tiene ancho de banda suficiente, pero en la realidad esto no era así, por lo que decidimos conectarle en la salida del amplificador operacional otro circuito de retorno, exactamente igual al de

la salida de la resistencia de detección, y llevándolo al pin A3 que es una habilitación nueva para el conversor analógico digital, para poder aplicarle el ratio. El ratio consiste en la relación de la señal de salida (pin A2) respecto la señal de entrada (pin A3). (Ver Figura 33)

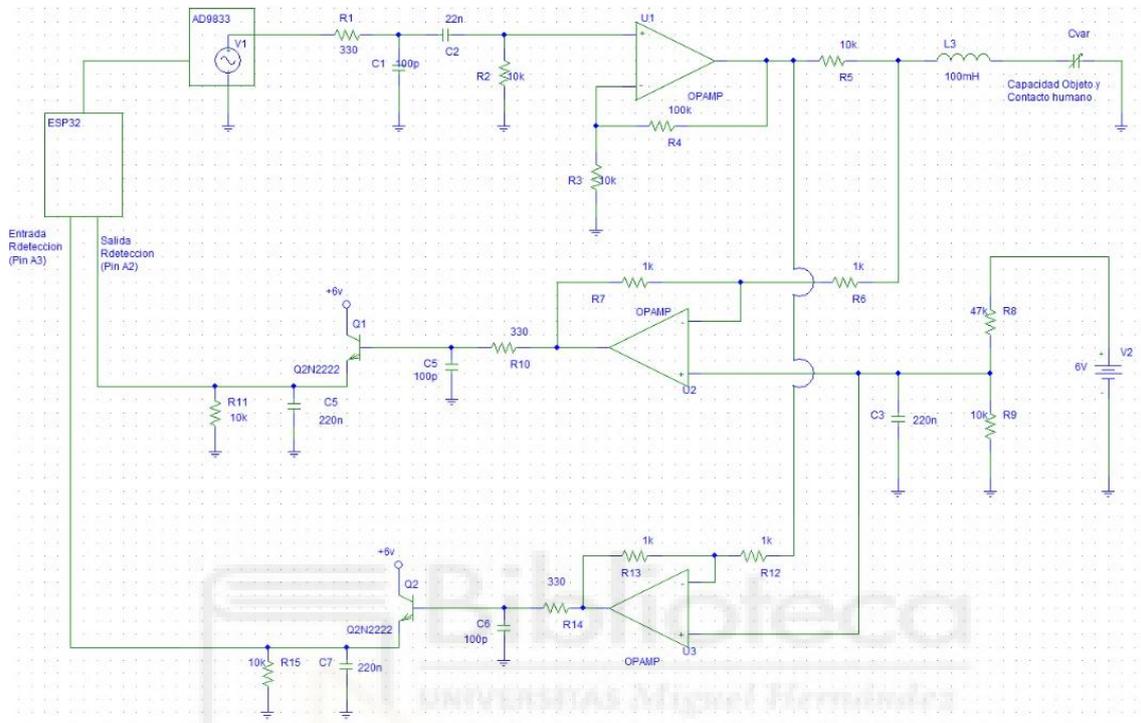


Figura 33: Circuito Touché con habilitación de circuito extra para aplicar el Ratio

De esta forma lo que intentaríamos corregir sería lo siguiente:

- Atenuación a frecuencias altas: Se pensó que era debido a las capacidades parásitas lo que hacía que a frecuencias altas se atenuase demasiado la señal, así que el ratio intentaría corregir esto haciendo un equilibrio entre la señal de salida de la resistencia de detección y la de entrada.
- Caída de la señal: Una caída en la señal de entrada contribuye directamente a una caída en la señal de salida, por lo que si usamos el ratio se intentaría corregir esta caída.
- Eliminar la dependencia de la variación en la señal de entrada: Debido a que la señal de entrada variaba, y en un principio no tendría que ser así, se decidió aplicar el ratio para considerar únicamente la respuesta en frecuencia del circuito.

Con estas implementaciones volvíamos a realizar pruebas interactuando de distintas formas, y lo que logramos fue lo que se ve en las siguientes figuras (Figura 34):

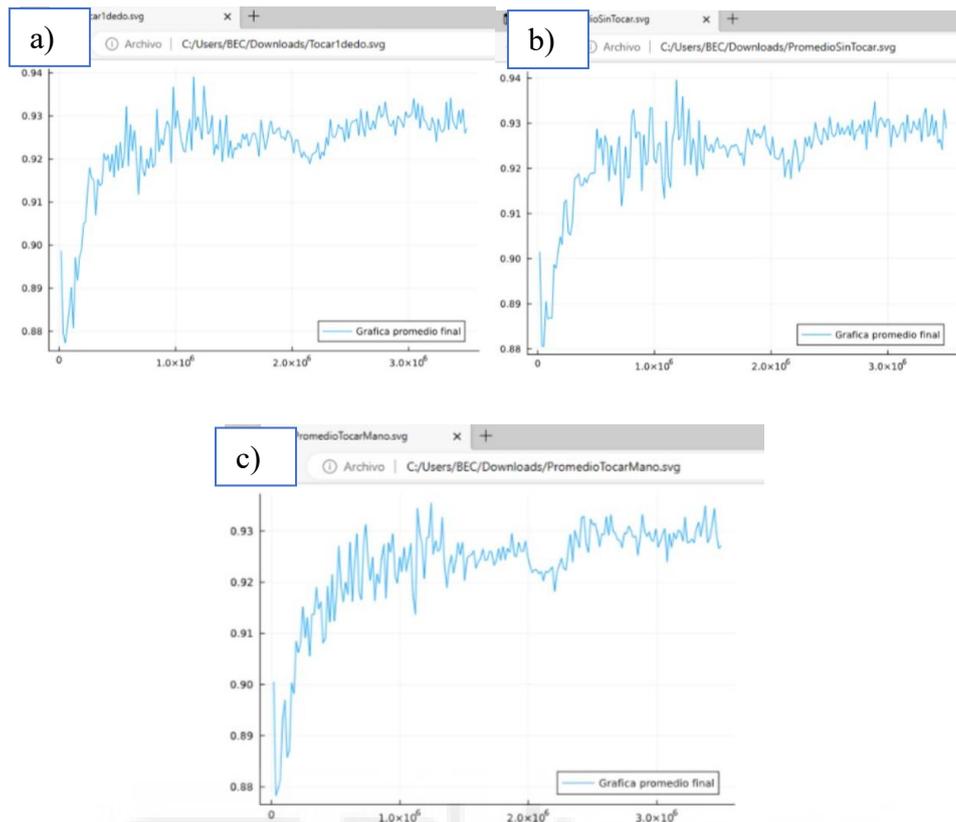


Figura 34: Gráficas mostrando el ratio salida/entrada para los gestos: a) Tocar con un dedo, b) Sin Tocar y c) Tocar con la mano

En esta primera fase logramos hacer que la señal fuera más o menos estable pero muy ruidosa. Esto provoca que si hubiera valles en la señal no se detectasen, por lo que ahora nos centraremos en cómo reducir este ruido.

4.2.2.1. FASE 2: CONDENSADORES DE DESACOPLO

En esta fase del desarrollo del circuito, abordaremos el tema del ruido de manera más concreta. Para comprender mejor este problema, es crucial enfocarnos en la etapa de filtrado. Inicialmente, esta etapa se diseñó principalmente con el propósito de reducir el ruido, y, de hecho, las mediciones en la salida de esta etapa demostraron una señal bastante limpia.

Sin embargo, el problema surge en la etapa posterior de amplificación, específicamente a la salida de los amplificadores operacionales, donde la señal comienza a ser afectada por ruido. Una posible fuente de ruido proviene de la alimentación de corriente continua (DC), y este ruido se transfiere a la señal de salida. Para reducirlo, se tomó la decisión de implementar condensadores de desacoplo en la entrada de alimentación de todos los amplificadores operacionales [16].

Estos condensadores se conectan en paralelo entre la fuente de alimentación y el pin de alimentación de todos los amplificadores utilizados. Los condensadores usados son electrolíticos y tienen una capacidad de 100 microfaradios (uF). Su función radica en filtrar el ruido presente en la señal de alimentación, eliminándolo del circuito y permitiendo que la señal permanezca continua (Figura 35).

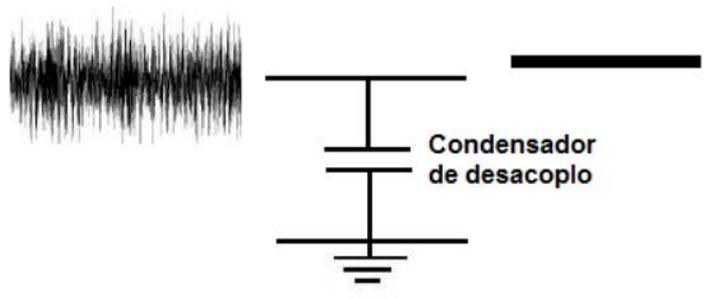


Figura 35: Efecto de un condensador de desacoplo

Por lo que ahora la señal se mantendría mucho más limpia como se puede ver en la siguiente figura (Figura 36):

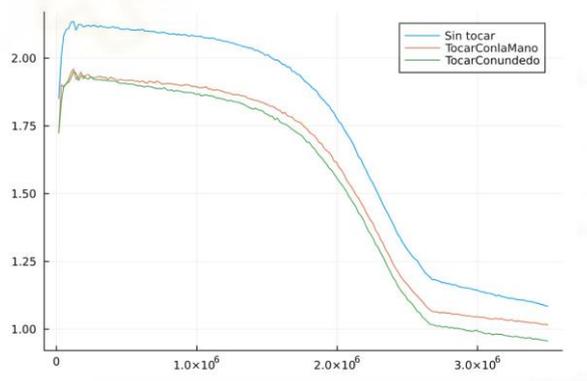


Figura 36: Gráfica del ratio salida/entrada con condensadores de desacoplo

Como vemos, la señal prácticamente no presenta ruido y se puede identificar las distintas actividades puesto que tienen amplitudes distintas. Sin embargo, no se puede apreciar la aparición de los valles relacionados con la frecuencia de resonancia y, por tanto, no se considera la variación de la capacidad al realizar el contacto. En realidad, tras varias pruebas se identificó que la presión que se está ejerciendo al objeto y la superficie de

contacto es la que determinaba la altura de cada curva, es decir, estaríamos midiendo la resistencia de contacto piel-objeto.

4.2.2.3. FASE 3: SELECCIÓN DE LA BOBINA Y PRIMERA PCB

En esta parte del proyecto nos enfocamos en investigar por qué no aparecen los valles esperados durante las distintas formas de interacción con el objeto sensorizado.

Primero, según vimos en la figura anterior (Figura 36), pensamos que sólo aparecía la primera parte del valle que no llegaba a recuperarse. En la figura se puede apreciar que alrededor de 2,5MHz, la señal se mantiene estable a nivel bajo. Para entender lo que estaba ocurriendo, realizamos diferentes experimentos: cambiamos los valores de los componentes del detector de envoltentes por otros, se probaron otros amplificadores para mejorar el ancho de banda; investigamos artículos y foros en los que se hablaba de la detección capacitiva para ver que podría estar ocurriendo.

Tras estas pruebas y análisis llegamos a la conclusión de que esa caída no era debida a la resonancia, sino a otros efectos como las capacidades parásitas del circuito. Estas podían ser elevadas puesto que se estaba utilizando una Protoboard, con cables largos y cruzándose en muchas ocasiones, y con componentes colocados de manera asimétrica. Todo esto inducía a capacidades parásitas que habría que intentar reducir en el diseño final, pero no explicaba que el valle no apareciera, por lo que continuamos investigando cómo conseguirlo.

Otra posibilidad era que el problema residiera en la parte LC, ya que un cuerpo humano es muy complejo de modelar y se desconoce su valor capacitivo, puesto que depende de muchos factores como la anatomía, sudoración, etc. Si la frecuencia de resonancia fuera muy baja, podría estar demasiado a la izquierda como para poder verlo, así que para “mover” el valle hacia la derecha disminuimos el valor de la bobina a 10mH del valor inicial de 100mH (ya que la frecuencia de resonancia es inversamente proporcional a la inductancia). Con este cambio comenzaron a detectarse valles en la señal de salida (Figura 37).

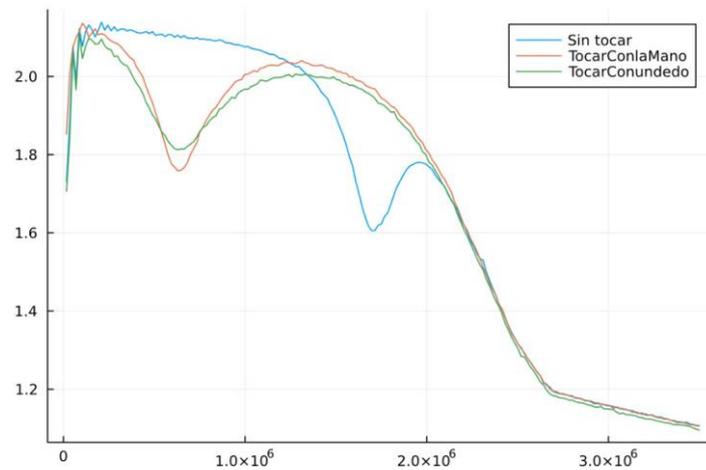


Figura 37: Gráfica del ratio salida/entrada con los valles en las distintas actividades

Como podemos observar, ya podemos ver los valles de las distintas formas de interactuar con la plancha metálica, pero esto es solo una victoria a medias, debido a que lo que queremos es poder distinguir perfectamente el desplazamiento de los valles, y solo vemos que se desplaza la actividad “Sin Tocar”, pero no las otras dos. Así pues, hicimos otra prueba con las mismas actividades, pero tocando con diferente fuerza la plancha metálica a ver si podíamos sacar algún tipo de conclusión y este fue el resultado (Figura 38):

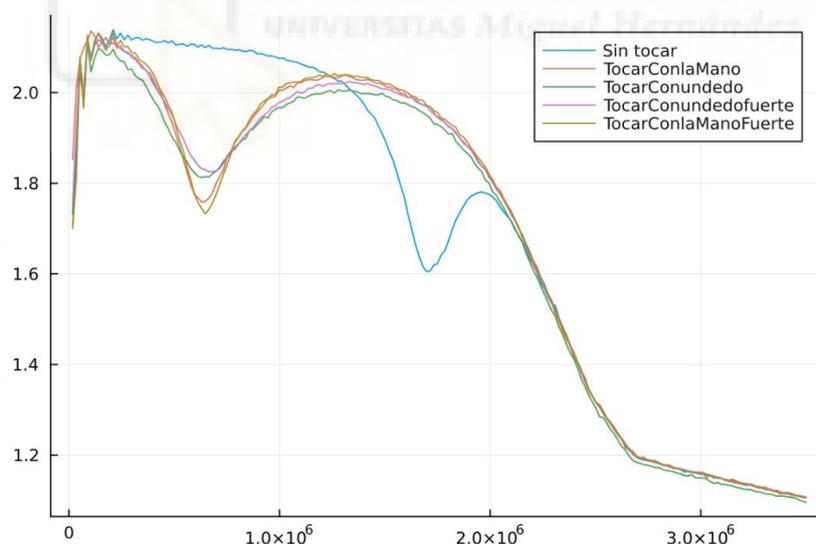


Figura 38: Gráficas del ratio salida/entrada aplicando distintas fuerzas en las actividades

Podemos observar que no se mueven los valles simplemente suben y bajan de amplitud, por lo que llegamos a un punto muerto del proyecto.

En este preciso momento hicimos innumerables pruebas para ver cómo podía hacer que se moviesen los valles, probamos cambiar componentes, añadir nuevos componentes, probar hacer el barrido a más frecuencia, pero todo esto fue en vano. Así que retomando

el concepto que previamente habíamos mencionado le dimos prioridad al tema de poder disminuir las capacidades parásitas. La idea de esto parte de lo siguiente, si nuestro circuito que está montado en una Protoboard tiene muchas capacidades parásitas, es decir, capacidades no deseadas que inducen los cables muy largos o cruzados, componentes alejados entre sí..., lo más probable es que estas capacidades parásitas sean tan grandes que, al intentar interactuar con el objeto, la capacidad de nuestra mano o dedo sea tan pequeña en comparación que hace que no varíe el valle, así que a raíz de esta idea, realizamos el diseño de una PCB para minimizar estas capacidades parásitas, teniendo en cuenta a la hora del diseño los siguientes parámetros:

- Acortar pistas el máximo posible: Esto produce que se reduzca el área superficial de los conductores, lo que disminuye la capacidad parásita.
- Evitar paralelismos entre pistas que transporten la señal: La capacidad eléctrica se fundamenta principalmente en que hay dos láminas en paralelo, separado por un dieléctrico, esto también se relaciona con las pistas, si ponemos dos pistas en paralelo, estaremos provocando que aumente la superficie y por lo tanto la capacidad. Este criterio es principalmente utilizado para aquellas pistas que transporten la señal, las pistas de alimentación o tierra daría igual, debido a que son señales continuas las cuales no afectan estas capacidades.
- Evitar ángulos rectos en las pistas: Debemos poner siempre que se pueda las pistas con cierto ángulo (normalmente 45°), cuando vaya a haber un cambio de sentido, nunca en ángulo recto porque puede dar lugar a concentración de campos electromagnéticos, ya que tienden a concentrarse en las esquinas y esto puede dar pie a interferencias.
- Evitar el uso de vías siempre que sea posible: Si se puede diseñar la PCB con pistas reducidas y no usar ninguna vía es preferible, ya que las vías inducen capacidades parásitas (alrededor de 0,5pF por vía).
- Separar al máximo las pistas entre sí: Esto provocará que haya una gran separación entre pistas por lo que la distancia entre ellas aumenta y reduce considerablemente las capacidades parásitas.

Teniendo en cuenta esto, diseñamos las librerías con los componentes, muchos estaban ya en la herramienta de diseño Eagle y otros tuvimos que añadirlos mediante

librerías externas como el componente de la ESP32 o el generador de ondas. Al final nuestro esquemático fue el siguiente (Figura 39):

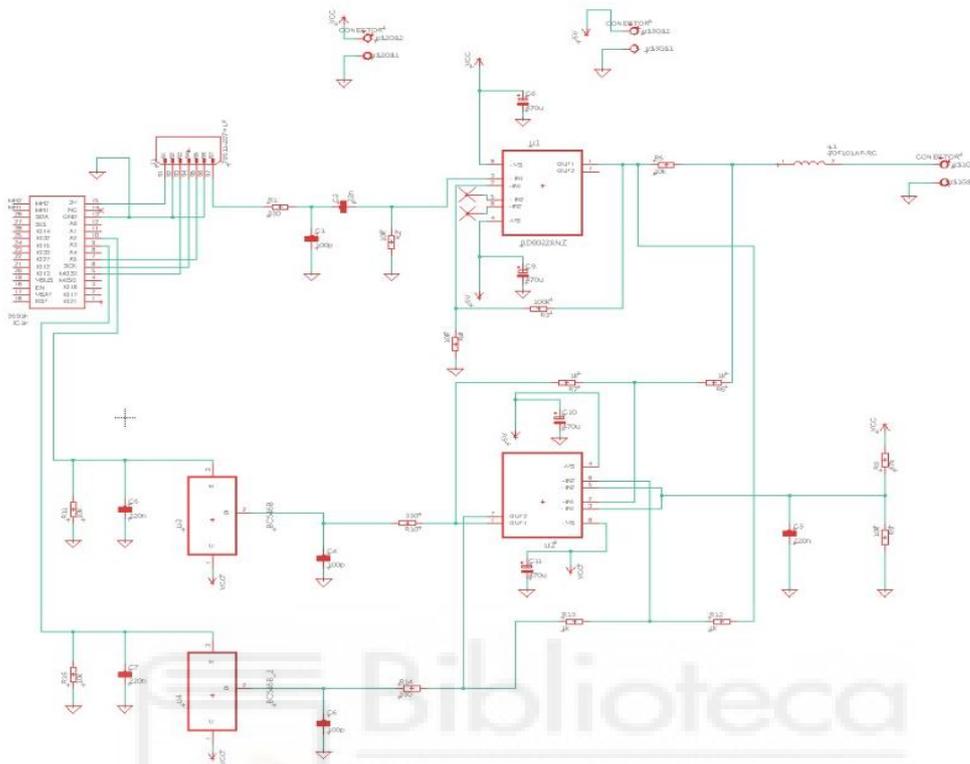


Figura 39: Esquemático del primer diseño de la PCB

Con este esquemático hicimos el diseño y colocación de los componentes en una placa de 80 x 60mm, así como su enrutamiento, los criterios de diseño vistos anteriormente con el DRC visto en la sección de Herramientas Hardware, y este fue el resultado final (Figura 40).

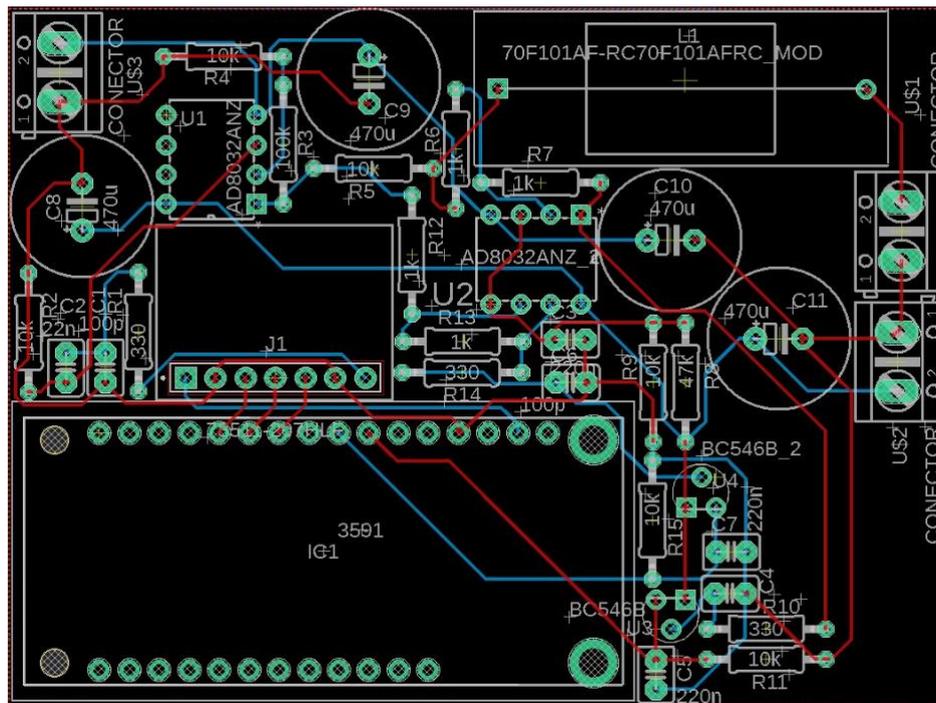


Figura 40: Diseño y colocación de componentes de la primera PCB

4.2.2.4. FASE 4: SIMPLIFICACIÓN DEL CIRCUITO

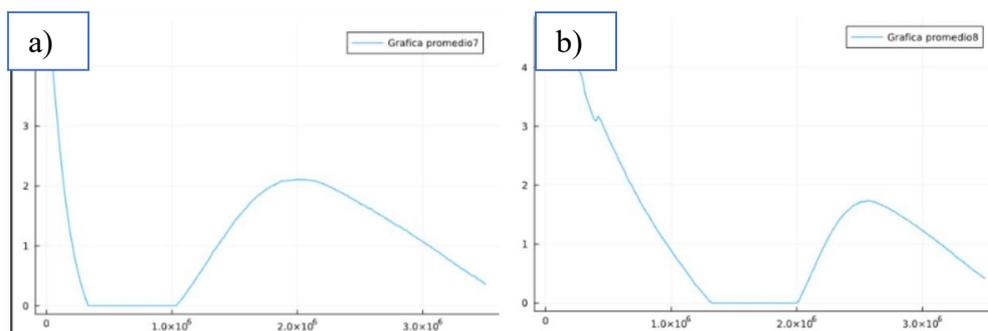
Una vez obtenida la placa de circuito impreso (PCB), procedimos a llevar a cabo la soldadura de todos los componentes, ya que se trataba de componentes con orificios pasantes (THD). Luego, realizamos nuevamente las pruebas, aunque los resultados no fueron alentadores, ya que los valles no mostraban variación significativa. No obstante, hubo una ligera mejora en la caída final de la señal, que ahora descendía a una frecuencia más elevada. Por lo tanto, continuamos realizando pruebas y experimentos.

Una de nuestras prioridades, dado que los valles no presentaban desplazamiento en frecuencia, era mejorar la visualización de la señal y asegurarnos de que los valles se distinguieran con claridad. Esto era esencial para poder diferenciar las actividades, ya que incluso las diferencias mínimas entre los valles podrían ser significativas. Nuestro primer enfoque fue revisar minuciosamente todos los componentes del circuito, incluyendo todas sus etapas, ya que estábamos convencidos de que aún existían capacidades parásitas. En este punto del proyecto, dejamos de seguir estrictamente las pautas del artículo original de Touché.

Comenzamos por eliminar las etapas que no eran necesarias, particularmente los dos filtros de paso bajo, ya que dado que nuestro barrido tenía pasos de 17.5 kHz, el filtro con una frecuencia de corte de 700 Hz tenía poco sentido. En resumen, el único filtro necesario era el de paso alto, que servía para eliminar el ruido y cumplía con ese propósito de manera efectiva.

A continuación, eliminamos el concepto de "ratio" entre salida/entrada, ya que implicaba la inclusión de una gran cantidad de componentes innecesarios. Optamos por conservar únicamente la señal de salida con sus respectivos componentes a la salida de la resistencia de detección. Aunque el "ratio" podría tener ciertos beneficios en algunos aspectos, estábamos convencidos de que el problema residía en otro lugar. También eliminamos el offset del amplificador y las resistencias que proporcionaban una ganancia de 1, dejando el circuito únicamente como un seguidor de tensión, tal como se implementó originalmente. Todas estas modificaciones tenían como objetivo simplificar al máximo el circuito y reducir las capacidades parásitas asociadas.

Para ampliar la anchura del valle y mejorar su visibilidad, redujimos la inductancia de la bobina de 10 mH a 470 uH. Realizamos varias simulaciones en PSIM para evaluar el efecto de variar la inductancia, y efectivamente, esta modificación hizo que el valle se ampliara. Con todas estas modificaciones, obtuvimos las siguientes gráficas para las diferentes interacciones (tocar con la mano, sin tocar y tocar con un dedo) (Figura 41).



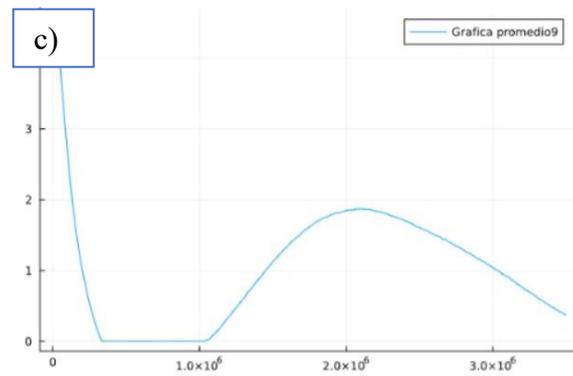


Figura 41: Gráfica de la salida para el circuito simplificado: a) tocar con un dedo, b) sin tocar y c) tocar con la mano

Como se puede apreciar en las gráficas, se ha logrado obtener una señal limpia con valles claramente definidos. No obstante, surgen dos problemas que se requieren abordar. El primero es una persistente dificultad para detectar las interacciones al tocar con un dedo o con la mano, un desafío que persiste en la implementación. El segundo problema radica en la existencia de una región donde la señal alcanza valores nulos, además de que la señal se inicializa en un nivel de tensión considerablemente elevado.

Debido a las cuestiones mencionadas, modelizamos un circuito diseñado para abordar estos problemas. Este circuito aplica un offset a la señal para subirla y, al mismo tiempo, la comprime para que la curva no comience a frecuencias bajas en 2,5 voltios, una elección deliberada para que cayera por debajo de la alimentación del procesador, y de paso que la señal se asemejara más a las gráficas presentadas en el artículo de Touché. Por lo tanto, hemos desarrollado este circuito que se conecta en serie con el divisor de tensión, con el fin de llevar la señal directamente al detector de envolventes (Figura 42).

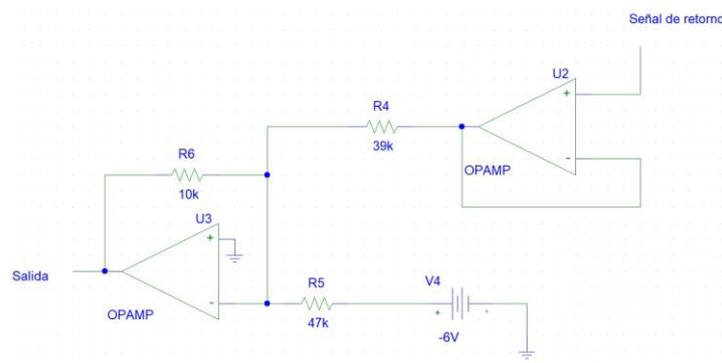


Figura 42: Circuito de offset que aparte recorta la señal

Tras implementar estos cambios se realizaron experimentos, probando a reemplazar la plancha metálica por una bola metálica que se asemejase al pomo de una puerta. Este es

un objeto interesante sobre el que detectar diferentes formas de interactuar. Se hicieron 5 actividades en dicha bola (Agarrar, Círculo, Pellizco, Tocar con un dedo y sin tocar) (Figura 43)



Figura 43: Actividades con la bola metálica

Estas son las actividades que nos gustaría detectar en forma de valles en la señal, así que hicimos el barrido para cada una de estas actividades y este fue el resultado. (Figura 44)

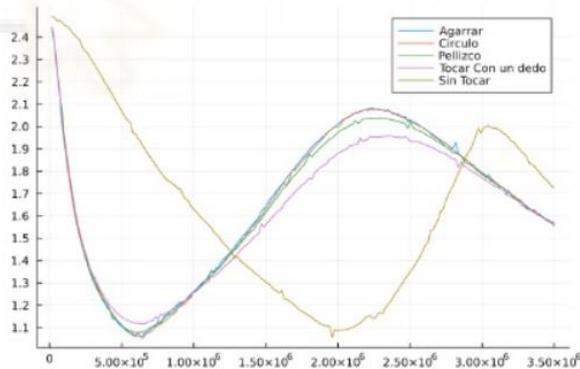


Figura 44: Gráficas de actividades en la bola metálica

Como podemos observar con todo lo implementado anteriormente, se puede ver las señales por encima de 0 voltios y con una amplitud inicial de alrededor de 2,5V. Sin embargo, seguimos teniendo el problema de que los valles no se desplazan en frecuencia caracterizando el tipo de actividad realizada.

4.2.2.5. FASE 5: DETECCIÓN DE VALLES

Esta fase resultó ser de vital importancia en la implementación de Touché. Después de semanas de exhaustiva investigación, nos encontramos con el desafío de comprender por

qué el sistema no detectaba los valles. A nivel de simulación en PSIM, se lograba la detección cuando modelaba la mano como un condensador conectado en serie a una resistencia, pero en la práctica esto no era así.

Fue en ese momento cuando, tras numerosas pruebas, utilizamos un osciloscopio para medir la señal y descubrimos un aspecto de suma relevancia. Notamos que al acercarse o alejarse la mano de la bola, la señal variaba, pero al tocar directamente el metal, la señal se volvía fija, independientemente de cómo tocáramos el metal. En ese momento, recordamos la definición de un condensador, que consiste en dos láminas conductoras separadas por un dieléctrico, que es simplemente un material aislante.

Fue entonces cuando identificamos la clave: la piel humana actúa como un conductor. Al tocar directamente la bola, no estábamos actuando como un condensador, sino únicamente como una resistencia. Por lo tanto, lo único que variaba en la señal era su amplitud, ya que dependía de la forma en que tocara el material conductor. Para abordar esta cuestión, decidimos colocar un aislante en la esfera. Inicialmente, utilizamos una bola de papel para llevar a cabo una prueba rápida de dos formas de tocar la esfera, y este fue el resultado (Figura 45):

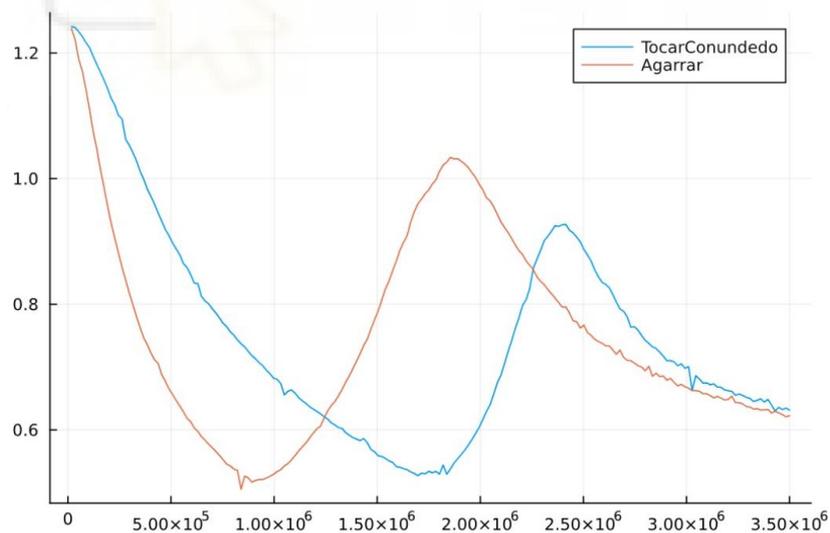


Figura 45: Gráfica con distinción de los valles en dos actividades distintas

Tras este hallazgo, hemos logrado alcanzar el objetivo deseado, que consiste en la implementación exitosa de Touché y la detección de diversas formas de interacción en el objeto conductor. Esto se hace evidente al observar claramente los dos valles desplazados.

Este descubrimiento también nos llevó a realizar un análisis más detenido del circuito LC. En realidad, no es necesario el circuito a la salida del seguidor de tensión. Para que la señal no se aproxime a cero, simplemente se debe reducir la resistencia de detección. El circuito RLC funciona como un divisor de tensión, con R conectado a la señal y con LC a tierra. Si resistencia de R es pequeña frente a la impedancia de LC, la salida se aproxima cero. En nuestro caso, tenía una resistencia demasiado elevada, por lo que optamos por eliminar el circuito de offset (siguiendo el principio de minimizar componentes innecesarios) y reducir la resistencia hasta que los valles fueran perfectamente visibles. Como resultado final, establecimos una resistencia de 470 ohmios. Luego, realizamos una prueba con todas las actividades y estos fueron los resultados (Figura 46):

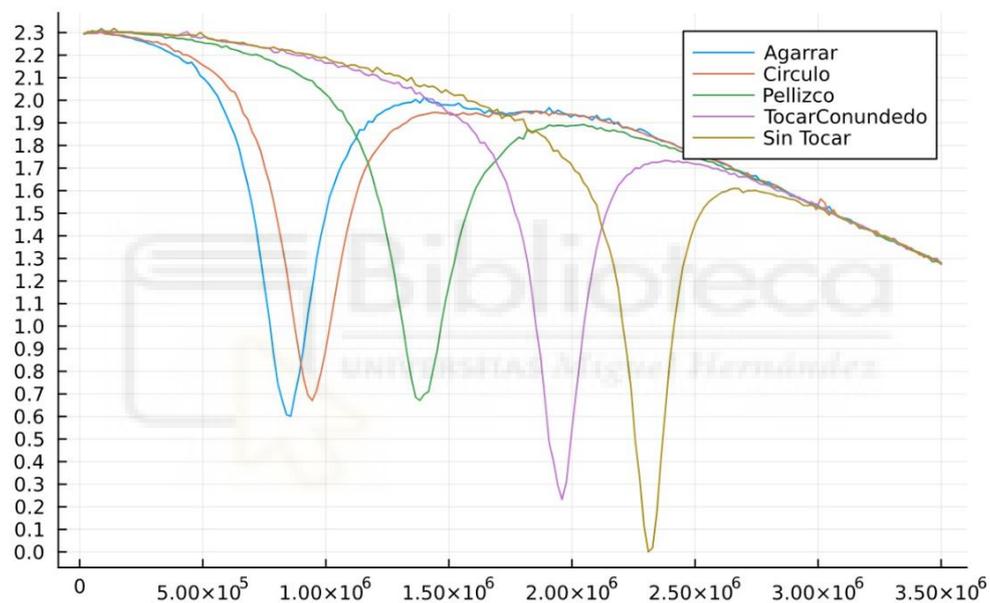


Figura 46: Gráfica de las 5 actividades en la bola

Ahora si tenemos implementado una versión funcional de Touché.

4.2.2.6. FASE 6: MEJORA DE LA SEÑAL E IMPLEMENTACIÓN DE LA PCB FINAL

Ahora que ya tenemos implementado de manera efectiva una versión inicial del circuito Touché, hemos cumplido nuestro principal objetivo, que era demostrar e implementar el funcionamiento de este. Ahora nuestro siguiente paso es optimizarlo. Comenzamos experimentando con diferentes valores de inductancia y estos fueron los resultados (Figura 47):

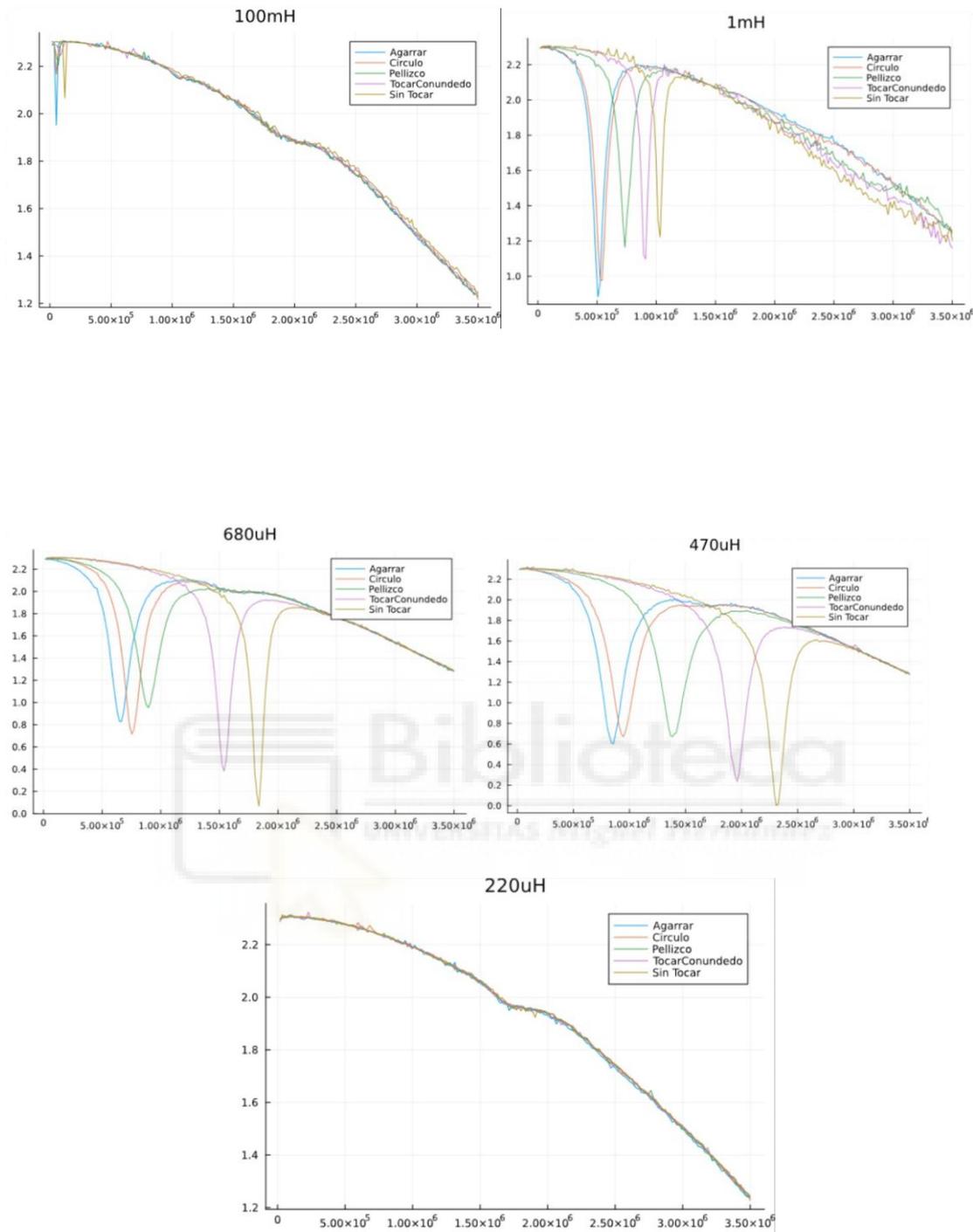


Figura 47: Gráficas de las 5 actividades en la bola con distintas bobinas

Claramente, el valor de 470uH es el mejor y la que se utilizó en el resto del trabajo. Tras encontrar el valor óptimo para la bobina, otra mejora necesaria era eliminar la caída de la señal con frecuencias crecientes. Si nos fijamos en la gráfica de las actividades para una bobina de 470uH, la señal parece que tenga una forma decreciente (independientemente de los valles).

Analizamos el circuito y nos dimos cuenta de que esa bajada pudiera ser debido a dos posibles factores. Uno podría ser las capacidades parásitas, las cuales sabemos que siempre habrá en el circuito. Otro pudiera ser la respuesta en frecuencia del amplificador, poniéndonos a investigar más acerca del mismo.

Debemos tener en cuenta que en el datasheet del AD8032A especificaba que a ganancia uno tiene un ancho de banda de 80MHz, así que, si dividimos esos 80MHz entre la ganancia que tenemos, que es de 11, tendríamos un ancho de banda de 7,2MHz. Ahora bien, esto es una primera aproximación, ya que dentro del ancho de banda existe una caída de tensión que llega hasta el 70% del valor máximo. Por ello realizamos una prueba con el amplificador en solitario montándolo en una protoboard con una ganancia de 11. A la entrada lo conectamos al generador de funciones y fuimos variando poco a poco la frecuencia y esto fue lo que obtuvimos (Figura 48):

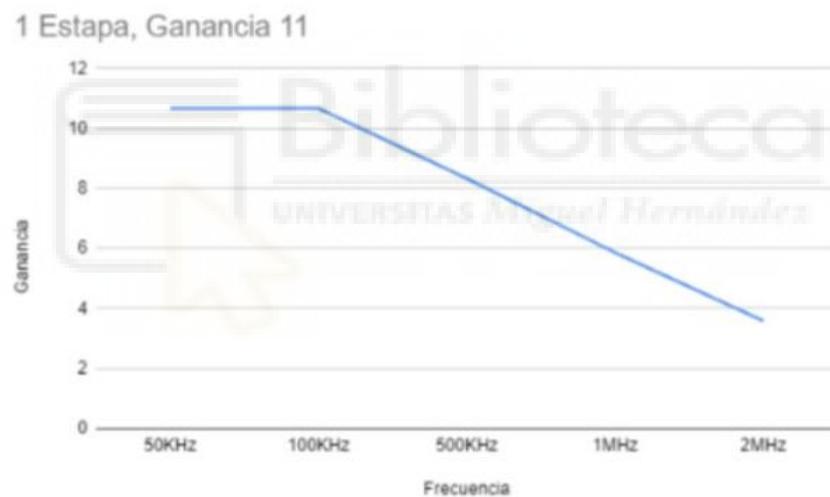


Figura 48: Gráfica de la linealidad del amplificador AD8032A con 1 etapa

Claramente se observa una tendencia descendente inducida por el amplificador elegido. Para mejorar la respuesta, se tomó la decisión de implementar una configuración de amplificación en dos etapas utilizando una amplificación general de ganancia 11. La principal ventaja de esta elección radica en la mejora de la linealidad de la amplificación. Para llevar a cabo la amplificación en dos etapas, se aprovechó la disponibilidad de dos amplificadores operacionales en el chip AD8032A. Cada uno de estos amplificadores se configuró para tener una ganancia de 3,3. Al conectarlos en serie, se obtiene una ganancia total de 11 ($3,3 * 3,3$). Para lograr esta configuración, se emplearon los siguientes componentes (ver Figura 49):

- 2 resistencias de 11Kohm: Conectadas entre la entrada inversora y la salida de cada uno de los amplificadores.
- 2 Resistencia de 4,7Kohm: Conectado entre la entrada inversora y tierra de cada uno de los amplificadores.
- 1 Chip AD8032: El cual es un chip integrado con 2 amplificadores operacionales. El chip en cuestión se puede alimentar hasta $\pm 6V$, que es la tensión que se le ha suministrado para este proyecto.

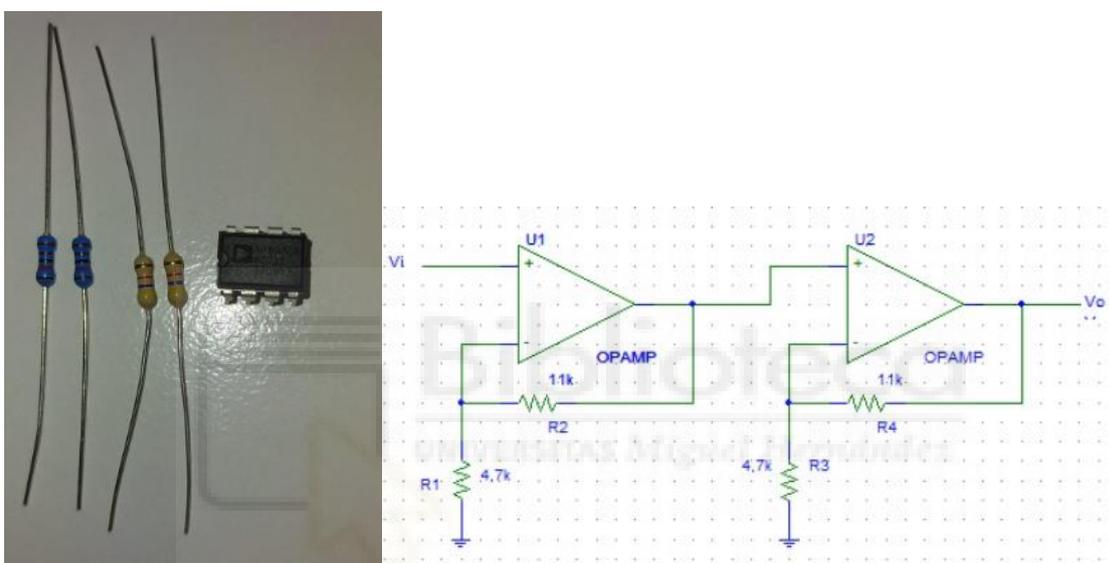


Figura 49: Componentes y esquemático del amplificador AD8032A con 2 etapas

Y el resultado fue el siguiente una respuesta mucho más plana (Figura 50):

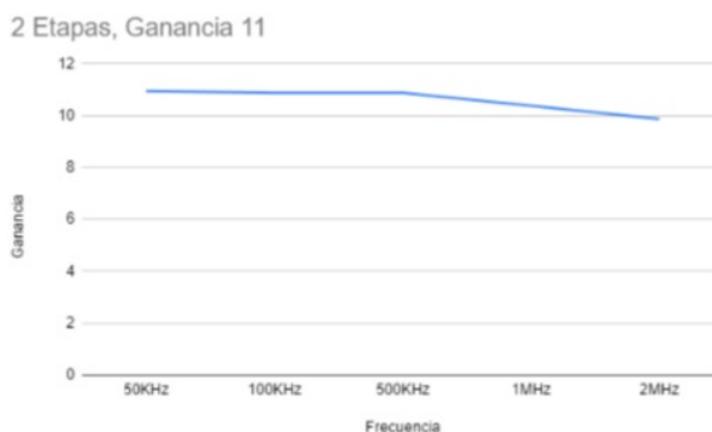


Figura 50: Gráfica de la linealidad del amplificador AD8032A con 2 etapas

Como podemos observar la señal no se atenúa tanto, que es lo que se busca, así que implementamos el amplificador de 2 etapas en nuestro circuito quedando como se muestra en la Figura 51: Esquemático del circuito final en PSIM:

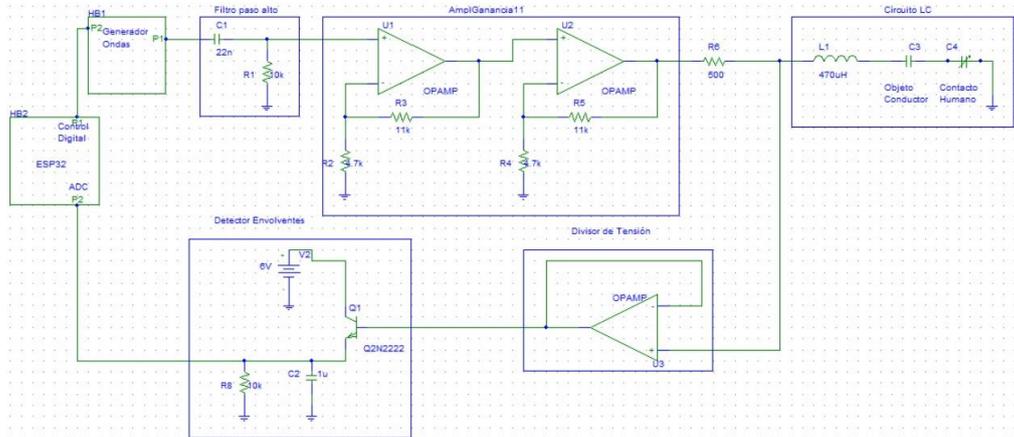


Figura 51: Esquemático del circuito final en PSIM

Este circuito es más sencillo y funciona mucho mejor que las versiones iniciales. Concretamente, se han eliminado varias etapas innecesarias, se han fijado valores, y se ha mejorado la respuesta en frecuencia.

Las pruebas anteriores se realizaron sobre un protoboard, por lo que, claramente, el siguiente paso es realizar el diseño final de la PCB optimizado y comprobar su rendimiento final.

Para ello, realizamos su respectivo esquemático (Figura 52: Esquemático del circuito final en EAGLE):

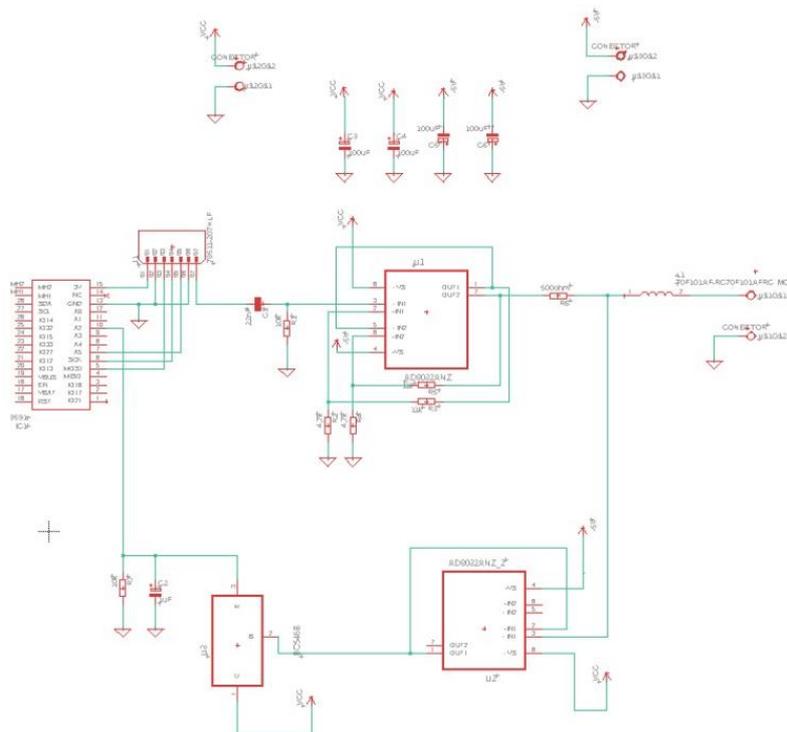


Figura 52: Esquemático del circuito final en EAGLE

Y su respectivo diseño, cuyas medidas siguen siendo la de la anterior placa, 80 x 60mm. Para este diseño hemos tenido mucho más en cuenta maximizar la separación de pistas y evitar líneas paralelas. Por otro lado, hemos hecho todo el enrutamiento de la señal en la cara Bottom, mientras que el enrutamiento de lo demás (conexión a tierra y tensiones) queda en la cara TOP. De esta forma, la señal se separa lo máximo posible del resto intentando reducir posibles capacidades parásitas asociadas (Figura 53):

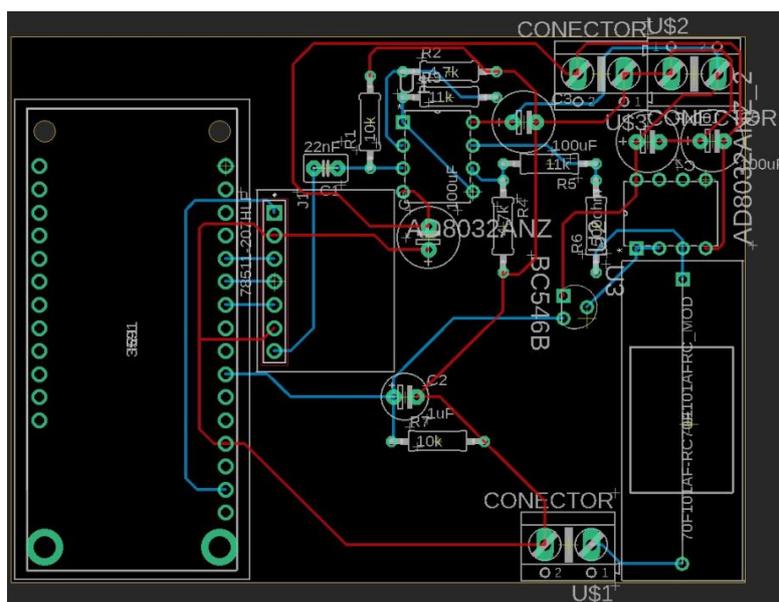


Figura 53: Diseño y colocación de los componentes para la fabricación de la PCB

Como resultado final, se consiguió la placa de nuestro diseño completo de Touché (Figura 54):

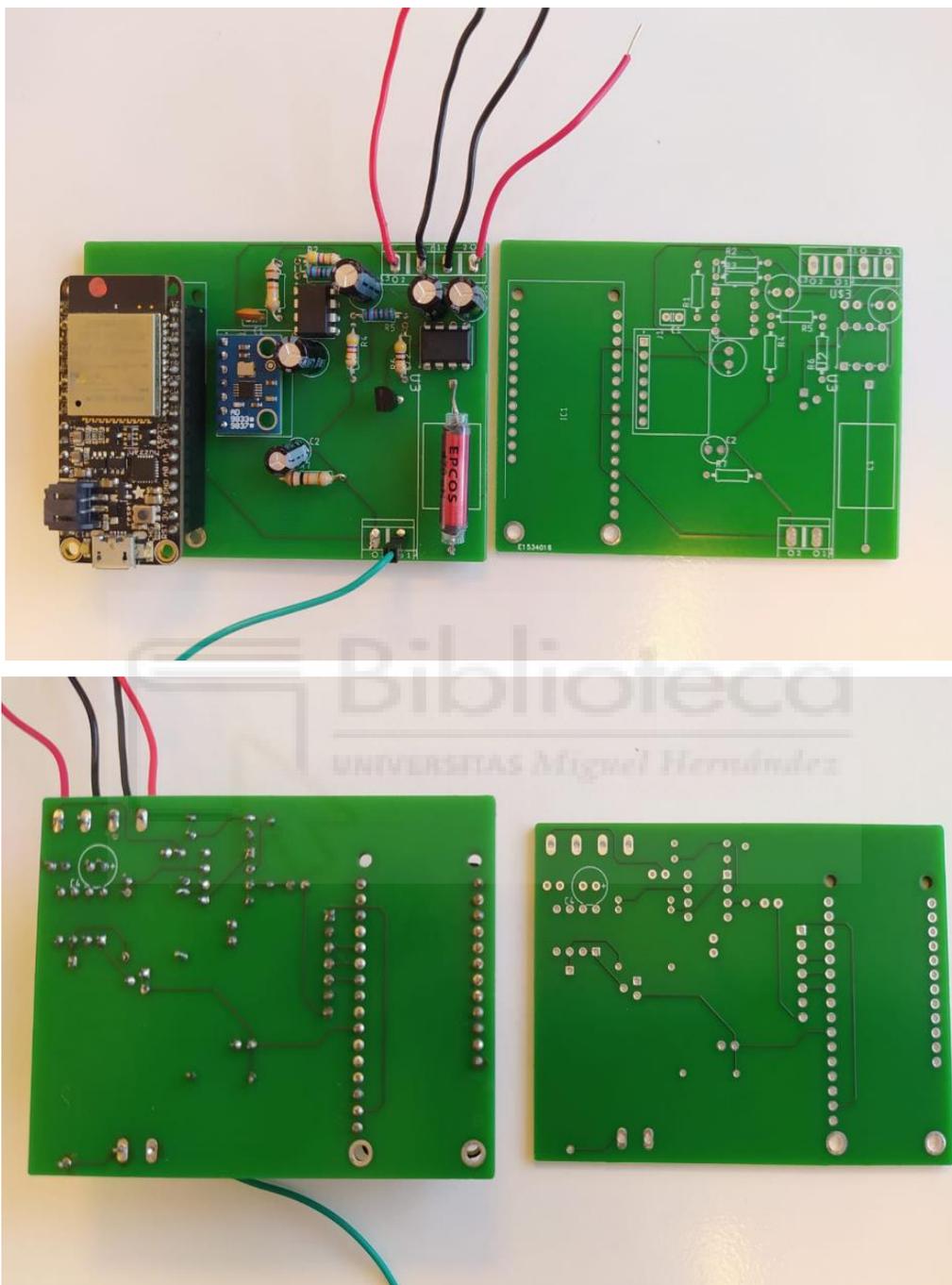


Figura 54: PCB con el circuito final de Touché

4.3 PRUEBAS Y VALIDACIONES REALIZADAS

Tras la fabricación y montaje del circuito Touché se realizaron las pertinentes pruebas de validación para optimizar sus parámetros y funcionamiento.

En una primera fase, se llevarán a cabo pruebas siguiendo el enfoque propuesto en el artículo, es decir, se explorarán las variaciones en la capacidad al interactuar con el objeto conductor. Posteriormente, se considerará la posibilidad de reemplazar la bobina de 470uH por un condensador, y sustituir la sección del objeto sensorizado por una bobina, con el propósito de que Touché sea sensible a cambios en la inductancia.

4.3.1. CIRCUITO TOUCHÉ CAPACITIVO

4.3.1.1. PRUEBA DE VELOCIDAD DE MUESTREO

Como primera optimización decidimos experimentar con la reducción del tiempo de barrido sin que afectase al rendimiento del circuito.

Inicialmente el barrido completo tardaba aproximadamente 1 minuto y 25 segundos en realizarse. El programa toma 15 muestras por frecuencia, para calcular la media y reducir su variabilidad. Por lo que tenía implementado dos delays. El primer delay es el tiempo para estabilizar la salida inicialmente de 50ms, y el segundo es el tiempo entre muestras que inicialmente era de 25ms.

Para elegir los valores óptimos, se realizaron 6 pruebas reduciendo esos delays a la mitad para dos actividades distintas: “Agarrar” y “Sin Tocar”, y estos fueron los resultados (Figura 55):

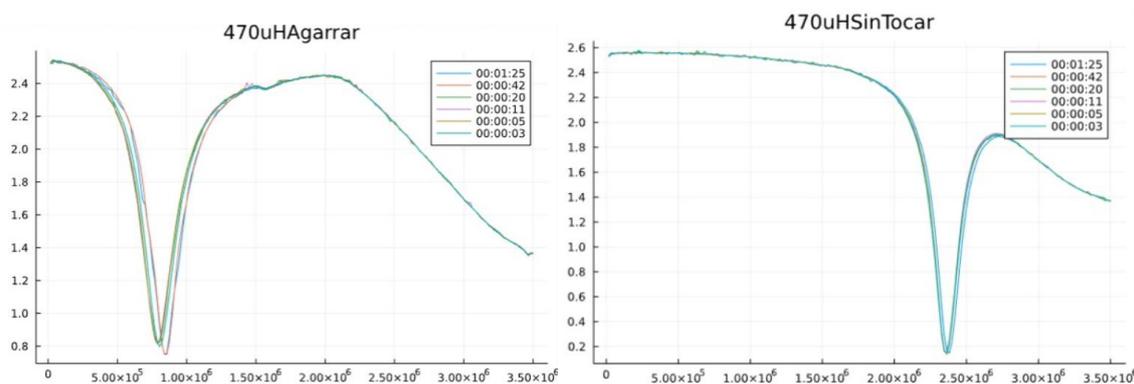


Figura 55: Graficas de velocidad de muestreo.

Como podemos observar la señal prácticamente se mantiene invariante entre pruebas, pero se consigue una reducción del tiempo de barrido para 200 frecuencias distintas de 1 minuto y 25 segundos, a sólo 3 segundos. Ambos delays se pueden reducir a 1 ms cada uno sin apreciar una degradación en los resultados obtenidos.

Para reducir más esa velocidad sin perder calidad en los resultados, se podría hacer de varias formas:

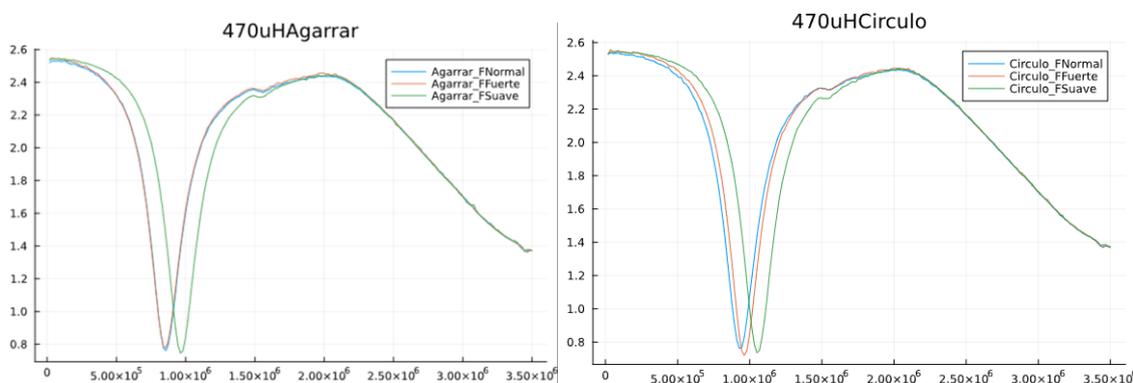
1. Reduciendo el número de muestras por frecuencia para promediar de 15 a un valor menor.
2. Reduciendo los puntos de barrido a menos de 200 frecuencias.
3. Desarrollando un algoritmo inteligente de búsqueda que elija el orden de las frecuencias a muestrear y reduzca el número de frecuencias necesarias.

Finalmente, no se redujo el tiempo de muestreo, ya que, en primera instancia, tres segundos era un tiempo adecuado.

4.3.1.2. PRUEBA DE FUERZA

Lo siguiente que decidimos analizar es ver como varía la señal, si para una misma actividad o gestualidad en el objeto conductor se le aplica distintas fuerzas. Esto viene a raíz de que cuando nosotros hacemos una acción en un medio como por ejemplo el de agarrar el pomo de una puerta no lo hacemos siempre con la misma fuerza y de hecho también varía esto de una persona a otra.

Si sucede que los valles se mueven mucho debido a la fuerza, podría pasar que, a la hora de querer detectar un movimiento o gesto, el programa que detecta actividades mezcle las gráficas, y por ejemplo podría confundir la actividad “Agarrar” con la de “Círculo”. Así que quisimos ver los efectos que tiene el hacer un gesto aplicándole una fuerza suave, una fuerza normal y una fuerte a ver que sucedía (Figura 56):



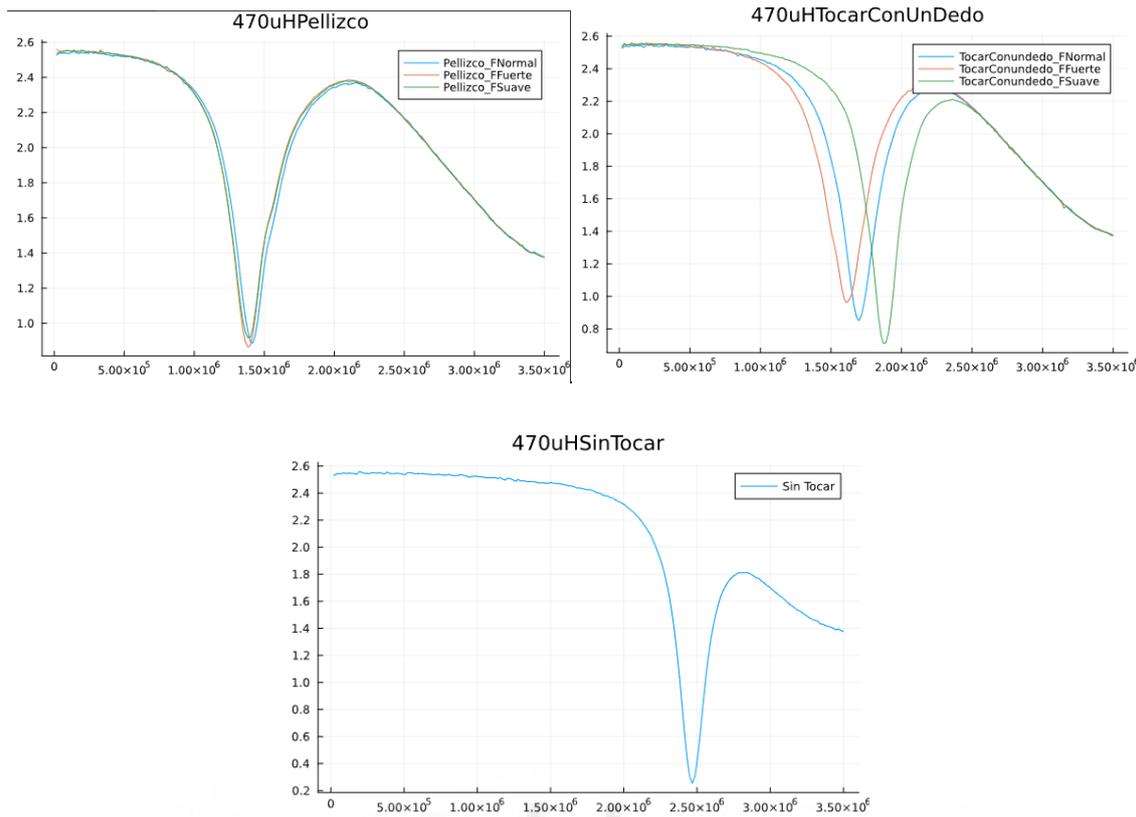


Figura 56 : Gráficas de actividades con diferentes fuerzas

Como podemos observar la fuerza que se le aplique sí que les afecta a los valles, no de manera significativa, pero le afecta, y como hemos comentado anteriormente podría confundir el “Agarrar” con el “Círculo”, lo demás sí que está suficientemente separados para identificarlos, pero de este problema hablaremos más adelante.

4.3.1.3. PRUEBA DE AISLANTE

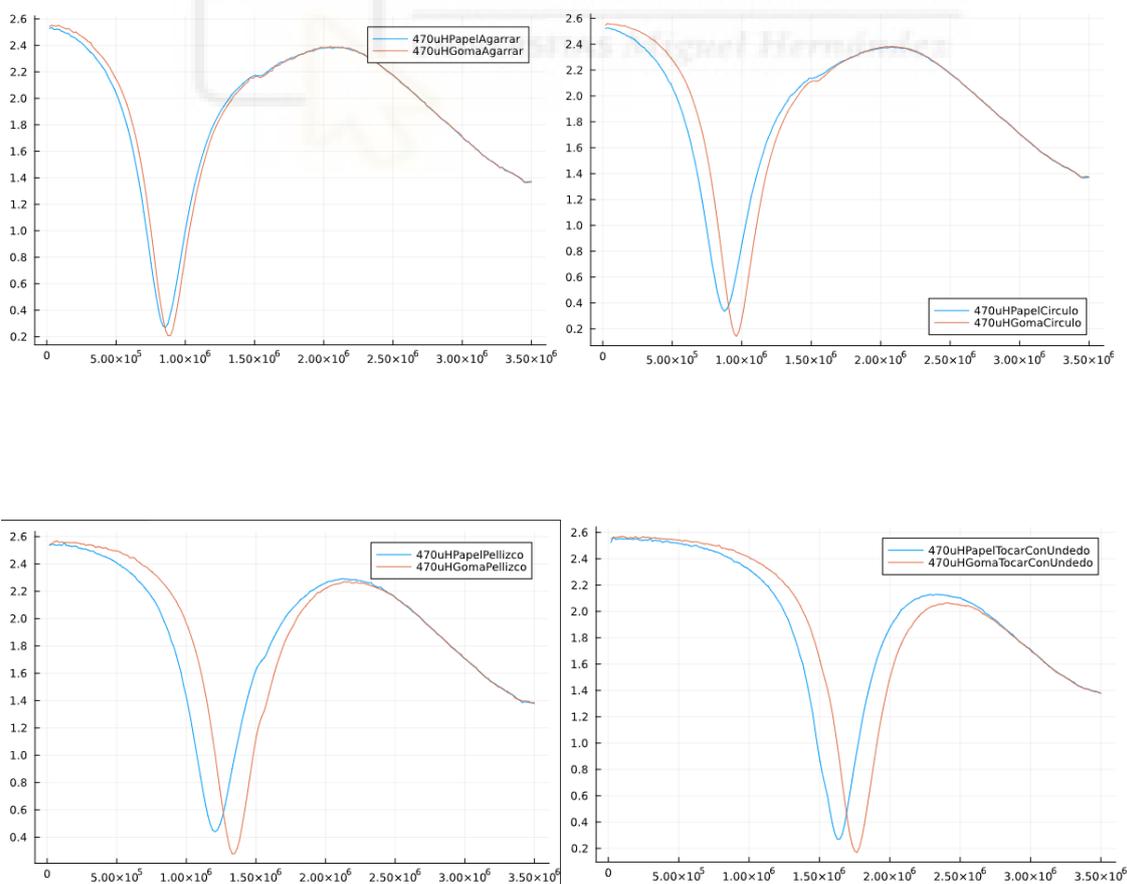
Otra prueba realizada fue cambiar el aislante y ver el efecto en el funcionamiento del circuito. Inicialmente se usó un trozo de papel recubriendo la bola, pero este aislante no era el más indicado, debido a que el papel tenía arrugas que podría hacer que variase la capacidad mínimamente. Tampoco era preciso ya que no se adaptaba bien a la superficie de la bola, por lo que decidimos hacer la comparativa entre el papel y un aislante que fuese barato y que se amoldase por completo a la bola. Por ello, probé con un simple globo, ya que es un material asequible, su elasticidad permite que se amolde por completo al objeto y deja la superficie de este completamente lisa, sin ningún tipo de arruga (Figura 57).



Figura 57: Dos materiales aislantes

Esto es una prueba que se ha hecho con un material que se puede comprar en cualquier tienda, pero obviamente puede haber materiales mejores, se puede hacer un aislante en una impresora 3D con otro material o para darle un toque estético, e incluso otra solución podría ser recubrir la bola de una especie de laca o esmalte que sea aislante, si no queremos variar mucho la estética de la bola.

Estas son las pruebas con esos dos aislantes (Figura 58):



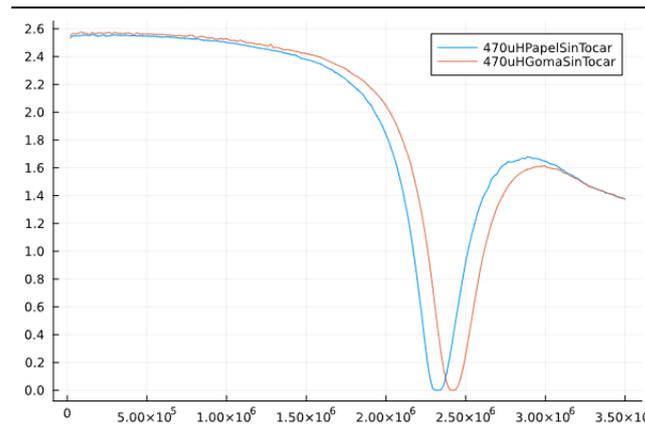


Figura 58: Actividades con distintos aislantes

Podemos observar que el aislante de goma aísla más la bola, lo cual hace que disminuya la capacidad y aumente la frecuencia de resonancia. Por sí mismo, esto no supone ningún problema ni ventaja, puesto que los valles se desplazan relativamente lo mismo. Por otro lado, la amplitud del valle es siempre la misma para el aislante de goma, mientras que, para el de la bola de papel no lo es. Esta repetibilidad en la amplitud del valle permite una mayor fiabilidad de detección.

Todas las futuras pruebas que se hagan con este aislante, ya que hemos visto que da mejores resultados.

4.3.1.4. PRUEBA DE MATERIALES

Analizando las propiedades de la bola cuando no se toca, se puede ver que la propia bola tiene una capacidad intrínseca. Esta capacidad irá directamente asociado a la superficie del objeto utilizado. A más área, más capacidad y más se desplazará el valle hacia la izquierda. El área de diferentes objetos con los que se realizaron pruebas es:

- Malla: $12,5 \times 9,5\text{cm} = 118,75\text{cm}^2$
- Plancha: $12,5 \times 9,5\text{cm} = 118,75\text{cm}^2$
- Bola (Radio=2,5cm): $4\pi \cdot 2,5^2 = 78,53\text{cm}^2$
- Cable (Radio=1mm): $\pi \cdot 0,1^2 = 0,314\text{cm}^2$

La Figura 59 muestra como afectaba estas capacidades intrínsecas probando distintos materiales haciendo la actividad de “Sin Tocar” en cada una de ellas.

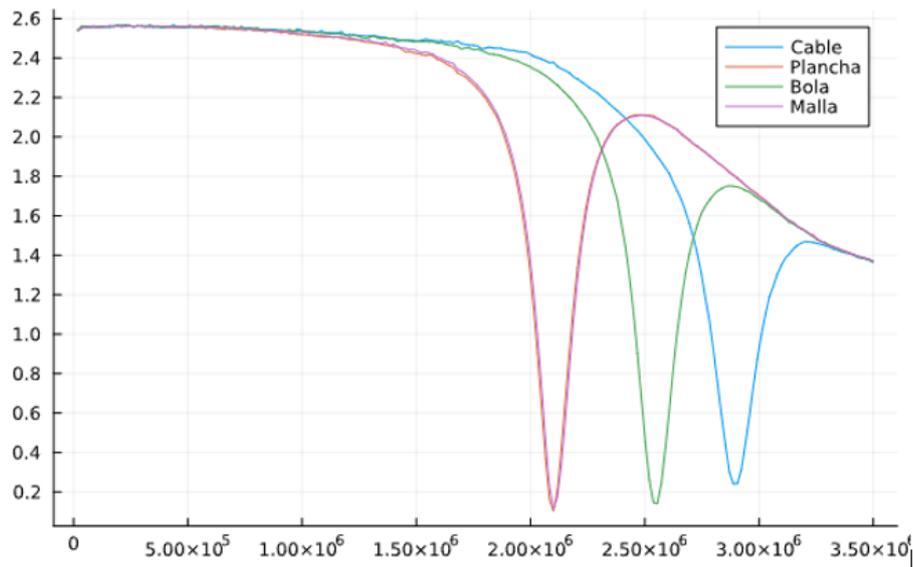


Figura 59: Prueba con distintos materiales.

Como hemos dicho antes, el área del material es directamente proporcional al aumento en la frecuencia de resonancia, por lo que sin duda el mejor material sería el cable ya que es el que presenta menor área y por lo tanto menor capacidad. Así, al acercar la mano al mismo, la capacidad varía de forma más significativa, y se podría decir que sería más sensible al contacto.

De momento se seguirá usando la bola para los experimentos ya que es un material mucho más versátil para hacer interacciones, pero para la silla o la cama se cambiará el objeto a un cable.

4.3.1.5. PRUEBA CON DIFERENTES PERSONAS

Después de todos los ensayos realizados para poder mejorar Touché, así como ver los mejores materiales para el objeto metálico, el siguiente paso es estudiar, cómo varía la gráfica si los gestos en el objeto conductor los hacen diferentes personas. Para ello, hemos realizado las típicas pruebas en la bola con nueve personas distintas. Si recordamos uno de los objetivos de este proyecto era poder hacer una interfaz de usuario táctil global para todas las personas, ya que no es lo mismo que nuestro programa detecte los gestos que he realizado, a que si esos mismos gestos los realiza otra persona, que no sea capaz de detectarlo, por ello el resultado de este ensayo fue el siguiente (Figura 60):

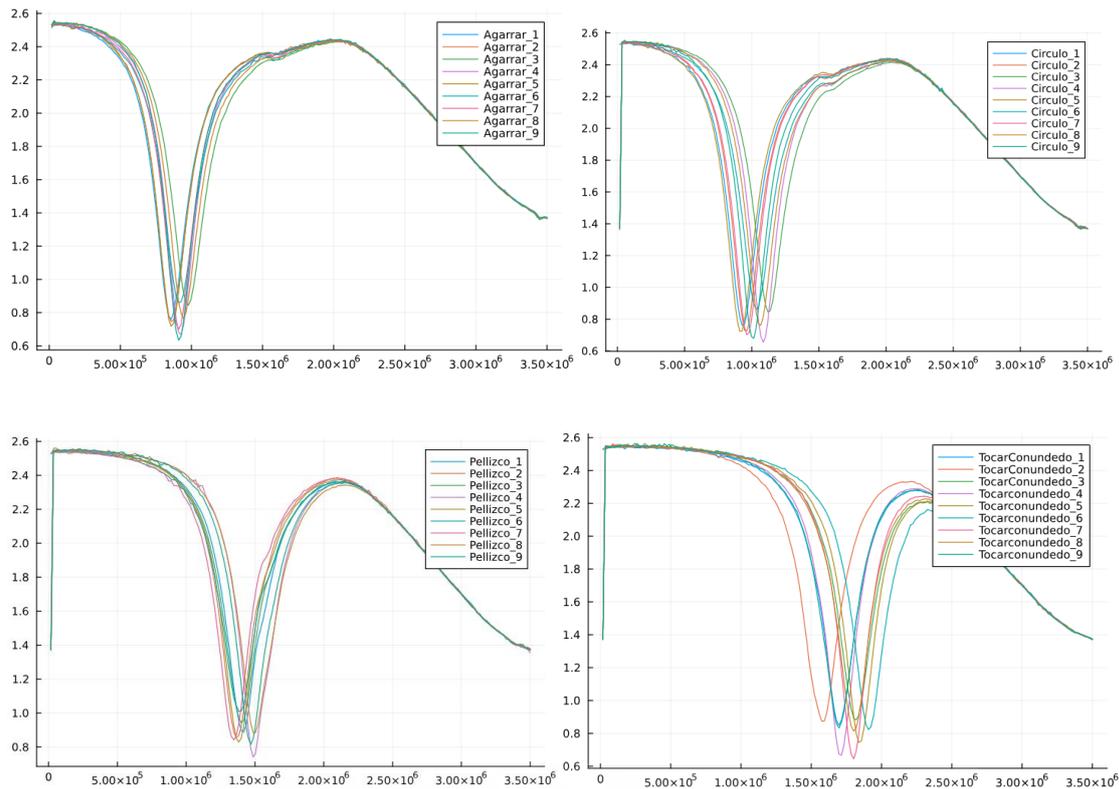


Figura 60: Pruebas con actividades en la bola con distintas personas

Como resultados muestra lo que ya nos esperábamos, y es que no introduce la misma capacidad una persona que otra, puesto que la capacidad propia de una persona depende de múltiples factores únicos. Las gráficas muestran que se puede identificar las actividades sin saber qué persona los realiza, a excepción de dos: “Agarrar” y “Círculo”. Para estas actividades, las frecuencias de resonancias están tan cerca, que hay a veces que se solapan para diferentes personas y actividades.

A raíz de aquí se abren dos posibilidades para poder seguir con el proyecto:

- Reducción de actividades gestuales: Reconocer que Touché, por sí solo, no es capaz de detectar todos los gestos imaginables. Si buscamos que sea capaz de reconocer actividades independientemente de la persona que las realiza, es necesario considerar la eliminación de actividades, como "Círculo". De esta manera, se facilita la implementación de Touché para la detección de cuatro actividades en lugar de cinco, que serían "Agarrar", "Pellizco", "Tocar con un dedo" y "Sin Tocar".
- Identificador de personas: En observaciones previas, hemos constatado que Touché es capaz de identificar todas las actividades cuando una persona realiza

gestos con una fuerza normal en un objeto sensorizado específico. Sin embargo, esta capacidad se limita a una persona en particular. Para abordar esta limitación, se podría considerar la implementación de un identificador de personas. Esto implicaría la recopilación de datos relacionados con las diversas actividades a realizar en el objeto sensorizado para cada persona que pretenda utilizar dicho medio u objeto. Luego, al implementarlo en un entorno de uso, se especificaría qué persona realizará los gestos, permitiendo así una identificación más precisa y personalizada de las actividades.

Ambas soluciones son factibles y se pueden implementar, pero la que nosotros utilizaremos es la primera opción, puesto que cuatro gestos son más que suficientes en el pomo de una puerta para poder programar ciertas funcionalidades en dicho entorno a nuestro gusto.

4.3.1.6. PRUEBA EN ASIENTO

Para esta sección nos iremos a otro entorno, en el cual sensorizaremos una silla. Para esta sensorización, hemos usado el elemento que menos capacidad intrínseca introduce (cable). Hasta el momento solo hemos realizado actividades con la mano, con una capacidad mucho menor que todo el cuerpo. Ahora los valles se desplazarán bastante hacia la izquierda en las gráficas. En primera instancia colocamos un cable a lo largo del asiento, y se realizaron tres actividades (Figura 61):

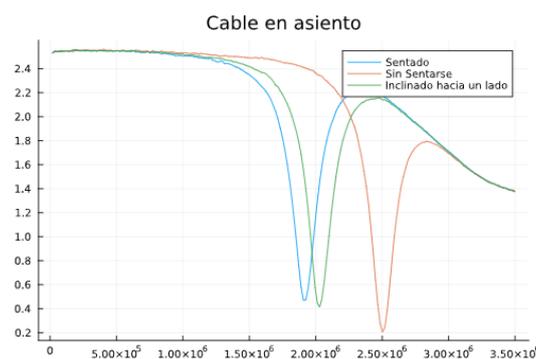


Figura 61: Actividades en el asiento de una silla

Como podemos observar se distinguen a la perfección las 3 actividades, pero el problema es que únicamente se sensorizó el asiento. Para un mayor número de actividades es esencial sensorizar también el respaldo, quedando de la siguiente manera (Figura 62):



Figura 62: Silla con un cable en el respaldo y asiento

Así que ahora que tenemos el asiento sensorizado en toda su longitud lo conectamos al circuito Touché y realizamos las mismas pruebas que antes (Figura 63):

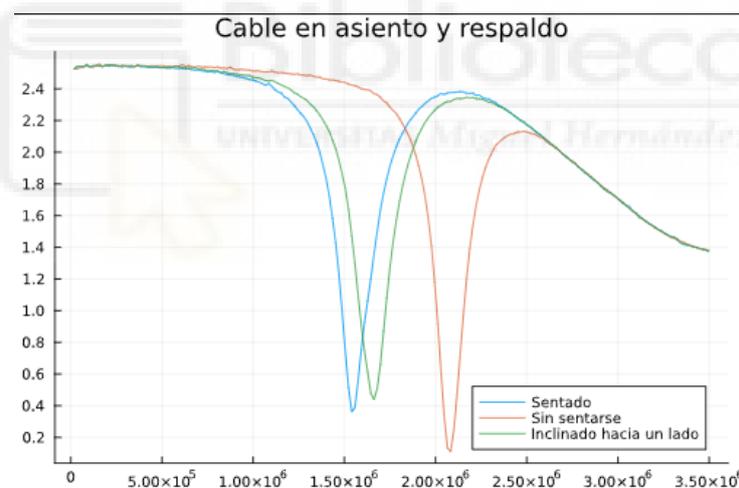


Figura 63: Gráfica de las 3 actividades realizadas con la silla enteramente sensorizada.

Como podemos observar podemos identificar perfectamente las 3 actividades realizadas en el entorno, también podemos apreciar una cosa y es que en cuanto más cable le pongamos mayor capacidad vamos a introducirle al circuito Touché y la frecuencia de resonancia será menor.

El siguiente paso es optimizar la sensorización de la silla, para recortar lo máximo posible el cable. Para ello, nos dispusimos a realizar diversas pruebas para ver cuál era la forma óptima de sensorizar una silla y que, además, abarque puntos críticos de contacto con el

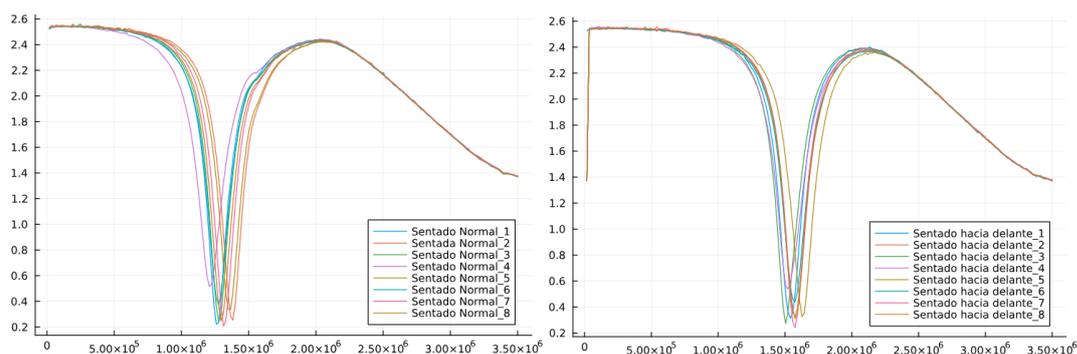
cuerpo humano cuando nos sentemos. Este fue la disposición final del cable para la silla (Figura 64):



Figura 64: Forma más óptima de sensorizar la silla

El cable se dispone de manera que detecta los puntos de mayor contacto con la silla. En la zona del asiento, se ha recortado el cable y enfocado hacia donde hay más contacto al sentarse. Esto se hizo después de notar que la altura de las personas afectaba la medición, ya que una silla baja hacía que los valles de una misma actividad variasen significativamente si la diferencia de altura entre personas era grande. Colocando el cable de esta manera, se da importancia solo a la zona del glúteo, que es consistente para todas las personas.

Tras esta aclaración nos fijamos un poco más en que actividades queríamos detectar y nos dimos cuenta de que las actividades que normalmente hace una persona en una silla son: “Sentado con la espalda hacia delante, Sentado con la espalda apoyada y levantado”, así que implementamos estas tres actividades ya que las otras que teníamos no parecían tan relevantes, e hicimos pruebas esta vez con más personas para ver cuáles eran los resultados (Figura 65):



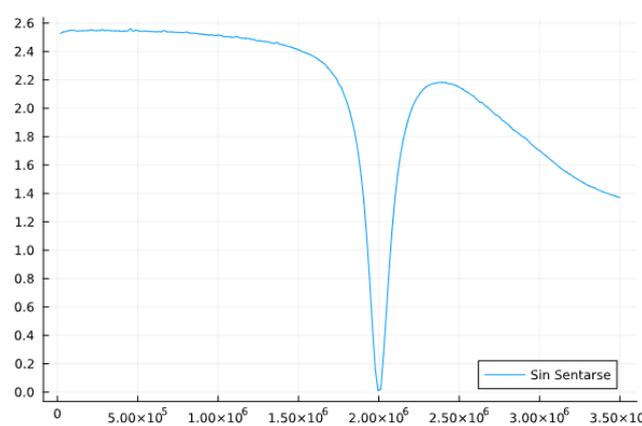


Figura 65: Gráficas de distintas personas en las actividades realizadas en la silla

Podemos observar que para estas tres actividades somos capaces de identificarlas independientemente de la persona que se siente.

4.3.1.7. PRUEBAS EN UN COLCHÓN

Una de las metas de este proyecto consiste en la sensorización de un colchón utilizando un cable. Este desafío es de naturaleza compleja, ya que, como hemos observado previamente, a medida que se incrementa la longitud del cable, aumenta la capacidad intrínseca del mismo. Para un asiento, esta implementación es viable, pero en el caso de un colchón, se requerirá una longitud bastante mayor de cable. En un esfuerzo por abordar esta complejidad, se consideraron las actividades típicas que se realizan cotidianamente en un colchón. Entre estas actividades, se encuentran estar tumbado (ya sea boca arriba o boca abajo, ya que ambas introducen una capacitancia similar), estar tumbado de lado, estar sentado en el colchón o estar levantado. Como resultado, se implementaron cuatro actividades para su detección.

Posteriormente, se exploró la disposición del cable en el colchón. La simple colocación de un cable recto en el centro del colchón no resultaría eficaz, ya que no permitiría una distinción clara entre las distintas posturas, como estar tumbado o tumbado de lado. Además, si el colchón es lo suficientemente ancho, podría darse el caso de que una persona esté tumbada de lado fuera del alcance del cable, lo que sería detectado como si estuviera levantada. Por lo tanto, se idearon dos configuraciones alternativas: una con el cable dispuesto en zigzag y otra con una forma en "W" (Figura 66).



Figura 66: Disposiciones de Zigzag y W para el cableado en el colchón

Los resultados obtenidos de estas dos configuraciones se presentan a continuación (Figura 67):

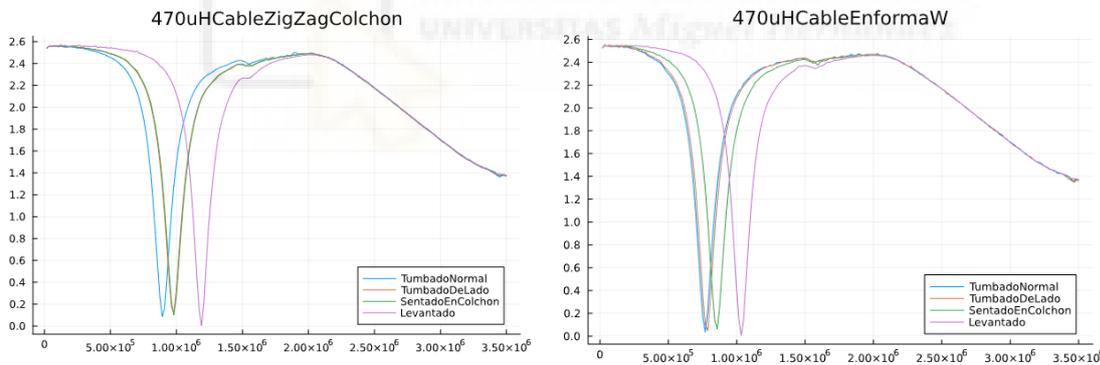


Figura 67: Actividades en las dos configuraciones del colchón

Es evidente que los valles en los resultados se encuentran más cercanos y desplazados hacia la izquierda debido a la significativa capacidad del cable. Sin embargo, enfrentamos una problemática similar a la experimentada anteriormente con la bola, en la que logramos distinguir todas las actividades, excepto la de estar tumbado de lado.

Para abordar esta cuestión, en lugar de eliminar la actividad de estar tumbado de lado, consideramos la posibilidad de incorporarla como "Tumbado Normal", mientras adoptamos la configuración en forma de "W". Esta modificación permitiría diferenciar con precisión entre la posición de estar tumbado, sin importar la postura exacta, estar sentado y estar levantado. Mantener estas tres actividades resulta particularmente útil, ya

que con ellas se podría determinar si una persona se ha despertado a su hora habitual, puesto que de normal al levantarse de la cama en primera instancia se está tumbado, luego nos solemos sentar y finalmente levantarnos. En caso contrario, se podría identificar un patrón de despertar anómalo.

4.3.2. CIRCUITO TOUCHÉ INDUCTIVO

Ya hemos visto como implementar Touché desde un punto de vista capacitivo, es decir, tienes una bobina de un valor fijo y mediante algún objeto metálico le induces capacidad con alguna parte del cuerpo humano. Esto tiene un problema, la capacidad no es fija, aunque hagamos la misma actividad en un cable o en la bola, el valle se puede mover. En principio, no debería de moverse mucho, pero para este proyecto aparte de predecir actividades de cierta persona, queremos detectar que persona está haciendo la actividad. Por ejemplo, si en una casa vive un matrimonio, estaría bien saber si en la silla sensorizada por Touché, está uno u otro miembro de la pareja. Esto mediante capacidades es muy difícil saberlo, ya que el valle siempre va a oscilar en un rango de frecuencias, aunque sea pequeño. Otra forma de analizar los valles de las gráficas es mediante un circuito que detecte cambios en la inductancia. Esta parte requerirá sustituir la bobina que teníamos antes por una capacidad fija, y el objeto por una bobina, la cual ha sido fabricada mediante espiras de hilos de cobre.

4.3.2.1. PRUEBA CON UNA BOBINA DE 50 ESPIRAS Y UNA CAPACIDAD DE 3PF

Ya vimos en anteriores apartados los fundamentos teóricos de las bobinas, así que probamos a fabricar nuestra propia bobina únicamente enrollando hilos de cobre sobre un tubo de cartón, la longitud es de 13,5cm, el radio de la bobina es de 2,5cm y cuenta con 50 espiras enrolladas entre sí, así que teniendo en cuenta la formula vista sobre los inductores en apartados anteriores, tendríamos una bobina de aproximadamente $42\mu\text{H}$ (Figura 68). Una vez hecha la bobina la conectamos al circuito y añadimos un condensador fijo de 3pF, el objetivo era ver si mi mano era capaz de provocar variaciones en el campo electromagnético que provoca la bobina. Así que conectamos los componentes de este modo (Figura 69):

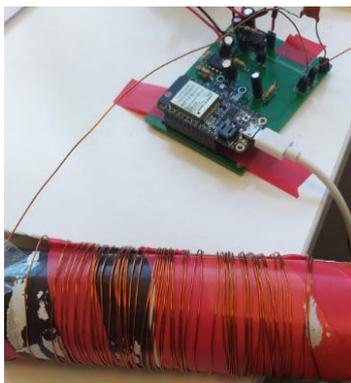


Figura 68: Bobina enrollada sobre un tubo de cartón y conectada a Touché

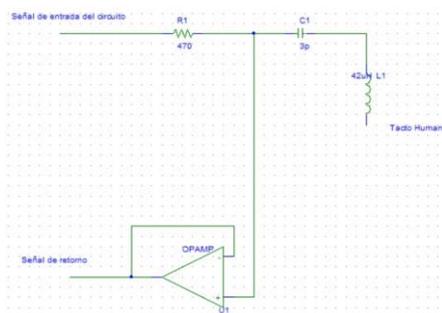


Figura 69: Esquema de la colocación de los componentes.

Tuvimos que aumentarle el rango de frecuencias para poder detectar interacciones con el objeto sensorizado con éxito a 10MHz (200 pasos de 50KHz cada uno) (Figura 70):

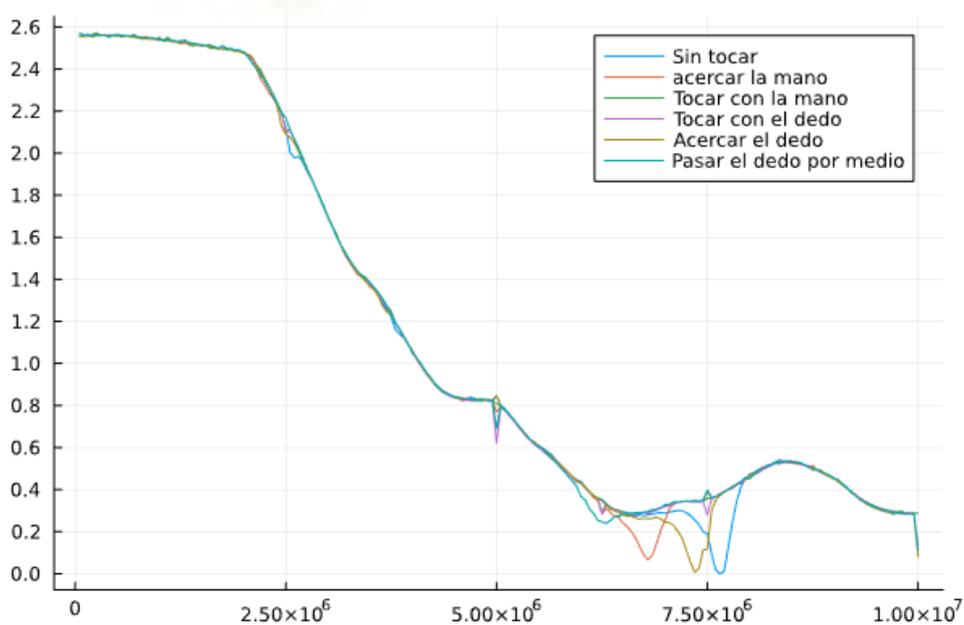


Figura 70: Gráfica en un rango de frecuencias de hasta 10MHz

Como podemos observar detectan los valles, pero solo de algunas actividades y a una muy alta frecuencia. Esto fue una de las primeras pruebas, con una bobina realizada al azar, pero obviamente necesitamos trabajar en la zona de frecuencias que teníamos previamente (hasta 3,5MHz), ya que el filtro paso alto está diseñado para atenuar la señal a unos 4,8MHz y aparte las 2 etapas del amplificador operacional hemos visto que iban a la baja conforme aumentaba la frecuencia de la señal.

4.3.2.2. PRUEBA CON UNA BOBINA DE 50 ESPIRAS Y UNA CAPACIDAD DE 220PF

Optamos por únicamente variar la capacidad del circuito y cambiar el condensador que tenía de 3pF por uno de 220pF con la misma bobina, de esta forma la frecuencia de resonancia estaría sobre los 1,6MHz aproximadamente. Así que hicimos pruebas con distintos gestos realizados en la bobina y este fue el resultado (Figura 71):

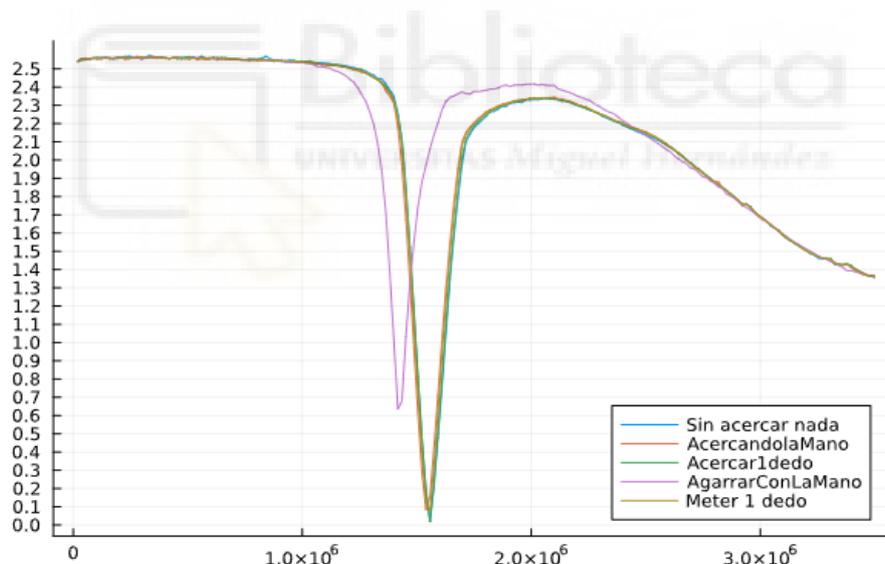


Figura 71: Grafica con el condensador de 220pF

Observamos que la frecuencia de resonancia se sitúa en torno a 1,6 MHz, y las actividades realizadas se encuentran dentro del rango previsto. Sin embargo, surge la problemática de que solo se detecta de manera adecuada la actividad de "Agarrar con la mano". Tras analizar los fundamentos teóricos del proyecto, llegamos a la conclusión de que no estamos modificando el campo electromagnético de la bobina, más bien, la bobina está operando como un condensador, además de su función como inductor, al aproximar la mano. Para ilustrar este concepto, la bobina se comporta como el objeto sensorizado en

el cual, mediante gestos, alteramos la capacidad inducida en él. Únicamente se detecta un gesto debido a que es aquel que introduce mayor capacidad. Los demás gestos implican acercar una parte del cuerpo sin establecer contacto directo con la bobina, y la capacidad de estos gestos, al ponerse en serie con el condensador de 220 pF, puede dar lugar a una variación prácticamente nula en la capacidad.

4.3.2.3. PRUEBA DE LONGITUD DE BOBINAS

Como vimos en el anterior apartado, todos nuestros experimentos al final resultaron ser enfocados nuevamente en principios capacitivos, así que ahora volviendo a lo que vimos en el apartado del modelo teórico de Touché inductivo, nos enfocaremos en la fórmula del inductor ($L = \frac{\mu * N^2 * A}{l}$), la cual dedujimos que lo único factible variar en esa fórmula es la longitud, transformando así la bobina a una especie de muelle.

Por lo tanto, si podemos diferenciar los valles a una longitud y a otra dentro del rango de frecuencias que nosotros queremos, ahora sí que estaríamos haciendo Touché sensible a cambios inductivos.

Realizamos una prueba con dos longitudes distintas, una de 13,5cm y otra de 5cm. La bobina es de 50 espiras, su radio es de 2,5cm y está montado encima de un tubo de cartón, cuya capacidad en serie junto a la resistencia de detección son de 220pF y 33ohm respectivamente. Los resultados de esta implementación son (Figura 72):

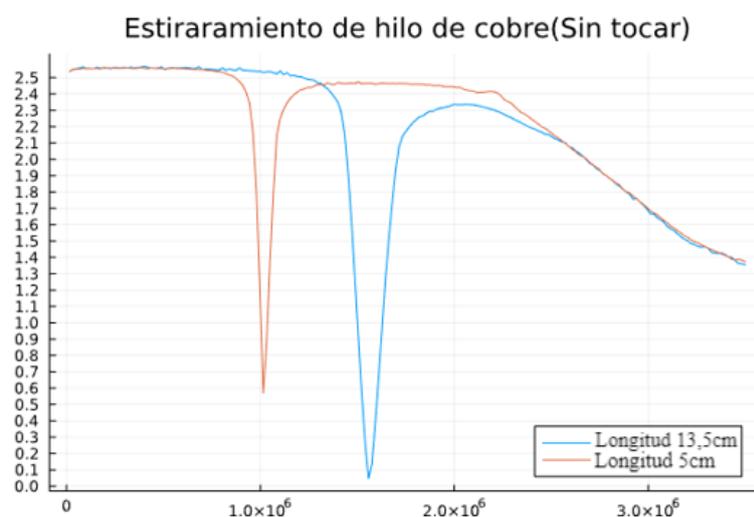


Figura 72: Gráfico estirando la bobina con dos longitudes distintas

Vemos que efectivamente se distingue a la perfección dos longitudes distintas, así que teóricamente se podría identificar dependiendo del muelle que se ponga si se sienta una persona u otra, si existe una diferencia de peso considerable, ya que la deformación del muelle será distinta y por consiguiente la longitud de este.

Muchas camas son de muelle así que se podría usar ese mismo elemento como bobina si queremos sensorizar la cama, sino habría que implementarlo en el interior de dicho elemento, o si queremos sensorizar una silla exactamente lo mismo.

4.3.3. DESARROLLO DE LA PROGRAMACIÓN

Como hemos visto, ya hemos hablado de todas las pruebas realizadas a nivel hardware en el propio circuito, tanto capacitiva como inductivamente. Si recordamos uno de los objetivos principales era poder realizar un programa capaz de enviar mensajes a un determinado usuario, así que antes de eso hablemos de los programas que ya tenemos implementado previo a ese para contextualizar un poco todo lo que llevamos a nivel software.

Tenemos, a nivel programación implementado un programa que es capaz de realizar el barrido de frecuencias de 0 a 3,5MHz, en pasos de 17,5KHz en nada menos que 3 segundos aproximadamente. Además, el programa nos muestrea 15 datos de la tensión en el convertidor analógico/digital por cada frecuencia. Aparte de esto, tenemos que en Julia podemos introducir todos estos datos, los cuales serán introducidos como una matriz de [200 x 15], en los cuales el primer término corresponde a las frecuencias que son 200 en total y el segundo, al número de datos por frecuencia, de esta manera tendremos en total de 3000 datos por barrido. El programa en Julia será capaz de calcularme la media de los 15 datos por fila y poder muestrearme esas 200 medias calculadas en una gráfica para poder visualizar los valles.

Lo que hemos mencionado antes son programas que ya tenía implementados, a raíz de esto he ido evolucionando el programa en C++, ósea el programa del barrido para hacer múltiples funcionalidades más:

1. Detección de gestos en el propio programa: La primera de todas fue en base a un bucle el cual se repetía el barrido de los 200 pasos de frecuencia, poder muestrear por pantalla el gesto realizado, ya sea en la bola, en la silla o en la cama. Para ello

lo que hicimos fue analizar todas las gráficas y cada uno de los valles de todos los gestos y ver donde se encontraba el mínimo, a raíz de ahí, programamos la funcionalidad que tenía en Julia que era la de poder conseguir la media de cada frecuencia, ya que nuestro objetivo era que el propio programa comparase la media de la frecuencia actual, con la media de la frecuencia anterior. De normal la media de la frecuencia anterior debe ser mayor que la media de la frecuencia actual (cuando hablamos de media de la frecuencia quiere decir la media de la tensión en el conversor a esa frecuencia), esto es así hasta un momento, el mínimo o valle. En el momento en el que la media de la frecuencia anterior sea menor que la media de la frecuencia actual significará que hemos detectado un valle. Así que asignamos un intervalo de frecuencias a cada actividad, por ejemplo, la actividad “agarrar”, tiene una condición para saber si se está dando esta actividad y es la siguiente (Figura 73):

```
if(contfrecuencia >=800000 && contfrecuencia<=945000 && (ValorMedioPresente > ValorMedioAnterior) && ValorMedioPresente <=0.9 && aux==0){
    CondicionAgarrar = 1;
    aux=1;
    log_i("HA ENTRADO EN EL IF DE CONDICIONAGARRAR");
}
```

Figura 73: Fragmento del programa donde se cumple que estamos agarrando la bola

Como podemos ver, hemos añadido dentro del if, que estaremos en el gesto “agarrar”, cuando la frecuencia esté en un intervalo dado (es el intervalo donde está el valle), que aparte el valor medio de la frecuencia actual sea mayor que el valor medio anterior y se ha restringido un poco más la condición, ya que incluso hemos puesto que la amplitud del valle sea menor a un determinado valor, todo esto para darle más restricciones y que no falle a la hora de decir que actividad se cumple, y por último una auxiliar, esto meramente hace que durante el análisis en el barrido solo nos imprima una vez por pantalla la actividad y no todo el rato hasta salir del intervalo de frecuencia.

Visto lo anterior, implementamos esa condición de if para cada gestualidad, con sus propios intervalos, y su propia amplitud. Y para saber en qué entorno estamos realizando las interacciones añadimos otro if, el cual lo que hacía era distinguir si los gestos los iba a hacer en la bola, en la silla, o en la cama (valor de 1, 2, 3 respectivamente), ya que cada objeto en el entorno doméstico llevaría su propio circuito Touché con su propia configuración para cada gesto (Figura 74).

```
[ 56316][E][main.cpp:202] loop(): TOCARCONDEDO
[ 56723][E][main.cpp:199] loop(): PELLIZCO
[ 57130][E][main.cpp:199] loop(): PELLIZCO
[ 57537][E][main.cpp:199] loop(): PELLIZCO

[ 58351][E][main.cpp:199] loop(): PELLIZCO
[ 58758][E][main.cpp:199] loop(): PELLIZCO
[ 59165][E][main.cpp:196] loop(): AGARRAR
[ 59572][E][main.cpp:196] loop(): AGARRAR
[ 59979][E][main.cpp:196] loop(): AGARRAR
[ 60386][E][main.cpp:196] loop(): AGARRAR
[ 60793][E][main.cpp:196] loop(): AGARRAR
[ 61200][E][main.cpp:202] loop(): TOCARCONDEDO
[ 61607][E][main.cpp:202] loop(): TOCARCONDEDO
[ 62014][E][main.cpp:202] loop(): TOCARCONDEDO
```

Figura 74: Gestos mostrados por pantalla dependiendo de la forma de interacción

2. Envío de alerta por correo: Después de poder identificar qué tipo de actividad se está realizando al igual de poder identificarlo para los objetos sensorizados (bola, silla, cama), nos propusimos el poder alertar a una persona si se ha cumplido cierta gestualidad a una determinada hora. Esto a simple vista parece una nimiedad, pero vamos a poner un ejemplo el cual ya mencionamos al inicio de este trabajo. Si una persona mayor tiene por costumbre levantarse de 8:00 a 9:00 de la mañana, y vemos que el sensor de Touché ha detectado, y sigue detectando el gesto de “TumbadoNormal”, y en ningún momento detecta “Levantado”, puede ser que esté ocurriendo algo fuera de lo normal o también puede que no, pero un simple mensaje por correo puede alertar y reaccionar a ello en caso de que este pasando algo. Para ello hemos implementado un programa el cual hay que incluir la red wifi a la que quieres conectar Touché, así como su contraseña, también habrá que añadirle el correo al cual quieres enviar el mensaje de alerta, su contraseña y el mensaje que quieres enviar. Este programa es un añadido al que hemos mencionado anteriormente, y se encargará también de ir proporcionando la hora, así que, si en el barrido detectamos cierta actividad lo podemos meter a la condición if junto a la hora a la que queremos que envíe ese mensaje, como, por ejemplo, lo mostrado en la siguiente figura (Figura 75):

```

if (getLocalTime(&timeinfo)) {
    int currentHour = timeinfo.tm_hour;
    int currentMinute = timeinfo.tm_min;
    int currentSecond = timeinfo.tm_sec;
    //Cambiar la hora y la condición para enviar un correo
    if ((currentHour >=8 &&currentMinute)=00 && currentSecond>=00)&&(currentHour<=9 && currentMinute<=00 && currentSecond<=00) && LevantadoColchonCorreo!=1) {

```

Figura 75: Fragmento de código el cual nos dice si no se ha detectado la actividad “LevantadoColchón” de 8:00 a 9:00

Ya hemos visto la evolución de un simple programa que únicamente su función era implementar un barrido de frecuencias y sacar por el conversor analógico/digital el valor de su tensión, a un programa más compacto el cual te dice la hora, el gesto que estás haciendo, en que medio u objeto lo estas realizando y aparte envía un correo electrónico a la persona que nosotros queramos si se cumple o no alguna gestualidad y a cierta hora del día (Figura 76):

<input type="checkbox"/> ☆ yo	ALERTA ACTIVIDAD INUSUAL - Se esta produciendo una anomalia en el horario habitual	10:36
<input type="checkbox"/> ☆ yo	ALERTA ACTIVIDAD INUSUAL - Se esta produciendo una anomalia en el horario habitual	10:36
<input type="checkbox"/> ☆ yo	ALERTA ACTIVIDAD INUSUAL - Se esta produciendo una anomalia en el horario habitual	10:36
<input type="checkbox"/> ☆ yo	ALERTA ACTIVIDAD INUSUAL - Se esta produciendo una anomalia en el horario habitual	10:36
<input type="checkbox"/> ☆ yo	ALERTA ACTIVIDAD INUSUAL - Se esta produciendo una anomalia en el horario habitual	10:36

Figura 76: Mensaje que llega al correo si ocurre lo que nosotros pongamos dentro del if

Cabe destacar una cosa, y es que hemos ido dando la explicación sobre los programas, así como lo que aparece por pantalla o por el correo, y se ha ido adjuntando fragmentos de estos programas para poder sustentar la explicación, pero, estos programas se podrán ver en su totalidad en la sección de ANEXOS, la cual estará incluida más adelante.

5. DISCUSIÓN

5.1. RESUMEN DE LOS PUNTOS CLAVE

Como sección previa a sacar conclusiones acerca de este trabajo de fin de grado, haremos un breve resumen de lo visto anteriormente, haciendo énfasis sobre todo en los puntos clave:

- En primera instancia se realizó un circuito siguiendo lo descrito en un artículo [3] del departamento de investigación de Disney (Disney Research) llamado Touché, el cual consiste en la detección capacitiva mediante barrido de frecuencias de

distintos gestos en un medio u objeto sensorizado. Para ello, se tuvo que concretar el diseño de algunos de los módulos como los filtros y etapas amplificadoras.

- Tras el primer prototipo montado en una Protoboard, se hicieron pruebas, y se observó que no funcionaba correctamente, requiriendo un rediseño profundo del circuito. Entre los funcionamientos indeseados del circuito se encontraron:
 - Intervalos de frecuencia donde la señal se limita a cero ocultando los valles (señal recortada a cero).
 - Ruido elevado.
 - Capacidades parásitas.
 - No se formaban valles en la señal.
 - No aparecía desplazamiento de los valles dentro del rango de frecuencias al realizar diversos gestos en una bola (similar al pomo de una puerta).
 - Mala respuesta en frecuencia, es decir, la señal tendía a decrecer en vez de a estabilizarse.
- Mejoras en el diseño del circuito...
 - Pistas de la PCB final más separadas para evitar capacidades parásitas.
 - Dos etapas de amplificación, con el fin de poder obtener una señal amplificada más estable.
 - Circuito más simplificado que el diseño original del artículo de Disney.
 - Una placa donde las capacidades parásitas son mínimas en comparación con la versión inicial en protoboard.
 - Señal de salida estable con valles claramente diferenciados dependiendo de las distintas formas de interactuar con un objeto.
 - Mejor respuesta en frecuencia del circuito, es decir la señal de salida se atenúa menos a frecuencias altas.
 - Optimización de los valores de los componentes pasivos, puesto que a base de ensayo y error se ha ido perfilando las etapas del diseño, así como sus componentes.
 - Mejora en la velocidad de muestreo del barrido.
 - Mejora y comprobación tanto de buscar el mejor aislante como el propio objeto conductor.

- Distinción de gestos/interacciones:
 - Las mismas actividades para distintos usuarios.
 - En entornos domésticos tanto en una silla como en una cama
 - Distinción de tres iteraciones distintas en los medios mencionados (silla, cama).
- Comprobación de cómo realizar Touché inductivo y que funcionalidad tendría.
- Implementación final de un programa que pueda, mediante la interacción de gestos en un objeto sensorizado, decir que gestos estamos realizando y aparte, indicar si se cumple o no cierto gesto a una determinada hora del día para enviar un mensaje de alerta por correo electrónico.

5.2. COMPARACIÓN CON OTROS TRABAJOS O SOLUCIONES EXISTENTES

En primera instancia, compararemos este proyecto con el artículo en el cual nos hemos sustentado, que es el artículo que habla de la tecnología Touché desarrollado por el departamento de investigación de Disney llamado Disney Research, con la finalidad de observar alguna mejora respecto a un trabajo ya hecho por una gran empresa, o ver si hemos conseguido por lo menos igualar los resultados. Hay que resaltar que el artículo empieza con las mismas interacciones en el pomo de una puerta, entonces compararemos sus resultados con los nuestros, pero después realiza otros experimentos los cuales no hemos realizado (pruebas en el agua, o en pantallas táctiles...), puesto que enfoca el proyecto más a entretenimiento y visualización (ya que es tecnología que se implementa en muchos parques de atracciones), que a la domótica y mejora de la calidad de vida en sí misma. Los resultados en el pomo fueron estos en el artículo de Touché (Figura 77):



Figura 77: Gestos y gráfica según artículo Touché

Y los de nuestro circuito es el siguiente (Figura 78):

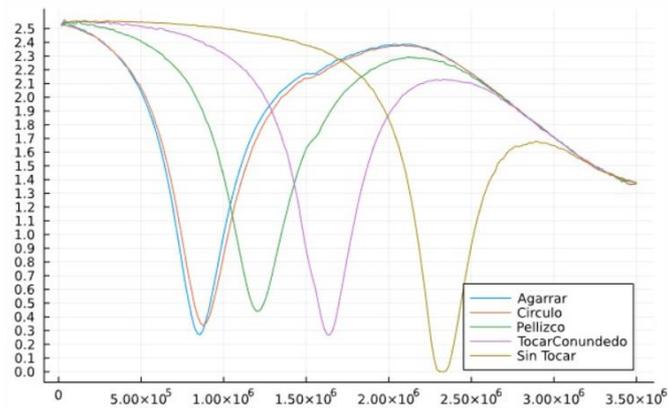


Figura 78: Resultados según nuestro circuito

Podemos ver una diferencia bastante sustancial, y es que nuestro circuito distingue a la perfección la actividad “Pellizco”, de la actividad “TocarConundado”, mientras que la gráfica del artículo Touché presenta esas gráficas tan juntas que puede inducir a error, y más aún, si queremos diferenciar ese gesto entre personas distintas.

Pero también debemos de ser justos y es que el circuito del artículo Touché si ha sido capaz de mantener la señal sin atenuación en todo el rango de frecuencias, mientras que nuestra señal alrededor de los 2,2MHz tiende a descender. Esto debido tanto a capacidades parásitas como a que el amplificador en dos etapas no es completamente lineal a altas frecuencias, aun así, es más que suficiente para la detección de iteraciones en el objeto conductor.

También debemos hacer énfasis al propio circuito, y es que el artículo Touché da muy poca información acerca del mismo sólo desglosa el circuito en bloques y nos da información concreta de únicamente dos componentes, el microprocesador y el generador de ondas (ARM NXP LPC1759, AD5932 Wave Generator, respectivamente). Además, usa muchos componentes SMD que hemos evitado por simplificar la fabricación. El circuito es el siguiente (Figura 79), en comparación con nuestra placa (Figura 80):

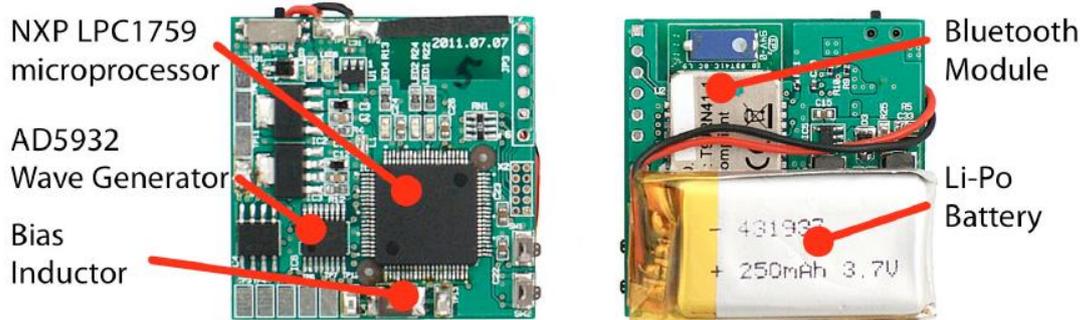


Figura 79:PCB del artículo Touché

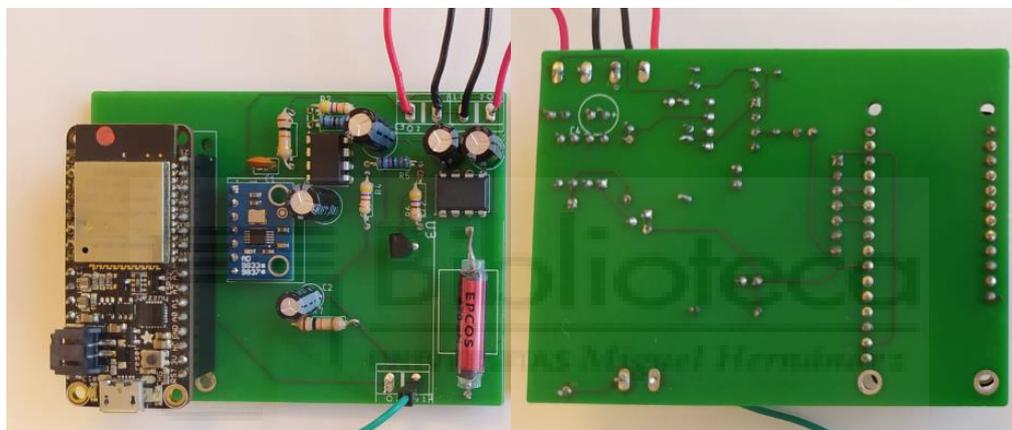


Figura 80:Nuestra PCB

Podemos ver que a pesar de que nuestros componentes sean de agujero pasante (THD), tenemos colocados muchos menos y con una disposición mucho más clara de los mismos. Cabe destacar que tenemos menos componentes, pero la placa del artículo Touché es de 36x36mm mientras que la nuestra es de 80x60mm.

Si nos fijamos en las soluciones parecidas que hay en el segundo apartado de este proyecto, podemos hacer un breve resumen de los beneficios de nuestro circuito respecto a las demás tecnologías, si recordamos teníamos dos tecnologías que podían llegar a competir con Touché, que eran las pantallas capacitivas proyectadas y las cámaras TOF 3D. Recordemos que respecto a la primera, teníamos como principal diferencia que las pantallas capacitivas proyectadas era una tecnología multitáctil, es decir reconoce varias interacciones simultáneamente del contacto humano, pero dificulta el poder detectar

movimientos o incluso gestos fuera del contacto con esta pantalla, y en segundo lugar respecto a las TOF 3D, teníamos que esta tecnología en comparación era muy costosa y aparatosa de implementar a nivel domótico. Teniendo en cuenta este resumen y habiendo profundizado más en la tecnología Touché, podemos hacer una comparativa más exhaustiva de la diferencia entre nuestra tecnología y las otras dos mencionadas:

- Pantallas capacitivas proyectadas: Se limita únicamente su uso a una pantalla táctil, así como interacciones en la misma superficie. Touché es capaz de poder sensorizar no solo una pantalla sino cualquier objeto no conductor en el cual pueda incrustarse un electrodo para medir la capacitancia e identificar gestos interactuando con la propia superficie del objeto o interactuando con el entorno alrededor de dicho objeto. Aparte de esto, las pantallas capacitivas proyectadas, requieren de más capas en el medio sensorizado, por ejemplo, Touché únicamente necesita 2 capas, las cuales son: Objeto metálico, y una capa de aislante que lo recubre. Las pantallas táctiles capacitivas requieren de 4 capas (Figura 81):
 - Capa de vidrio: Esta es la capa más externa de la pantalla y es la que interactúa directamente con el usuario.
 - Capa conductiva: Justo debajo de la capa de vidrio, esta capa es la que permite la conductividad eléctrica.
 - Capa de aislamiento: Esta capa impide que la corriente fluya libremente, forzándola a pasar a través de los electrodos.
 - Electrodo: Estos son los que detectan la variación en la capacitancia cuando un objeto conductor, como un dedo, se acerca a la pantalla.

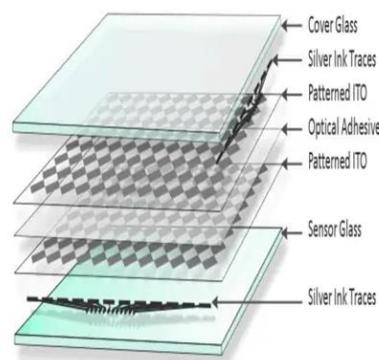


Figura 81: Capas de las pantallas táctiles capacitivas proyectadas.

- Cámaras TOF 3D: En comparativa con la finalidad de Touché es similar, ambas tecnologías reconocen gestos, uno lo hace en un entorno tridimensional a través de cámaras especializadas en escaneo y mapeo del entorno, y otro lo hace midiendo la capacidad que se hace en un objeto sensorizado o cerca de este. Ahora bien, si nos centramos en detalles más técnicos, las cámaras TOF requieren muchas implementaciones para su funcionamiento, sobre todo a nivel software, ya que a nivel hardware estaría parejo con Touché, para implementar una cámara TOF se requeriría de lo siguiente:
 - Hardware:
 - Óptica y Fotónica: Necesitaríamos tecnología y conocimientos en formación de imágenes y propagación de la luz, esto para diseñar sistemas de captura de imágenes.
 - Sensores y Fotodiodos: Implementación de sensores de luz, fotodiodos, y dispositivos fotosensibles para la captura de imágenes.
 - Diseño de circuito electrónico: Estas cámaras también requerirán de diseños electrónicos para la integración con otros componentes.
 - Procesamiento de señales Analógicas y Digitales: Implementación de convertidores para procesar los datos de manera correcta.
 - Software:
 - Programación en algún lenguaje.
 - Procesamiento de imágenes y visión por computador
 - Algoritmos de tiempo de vuelo.
 - Interfaz de usuario y Desarrollo de Aplicaciones.

A parte de esto deberemos tener en cuenta muchos factores físicos y matemáticos.

6.CONCLUSIONES

6.1. REFLEXIONES FINALES SOBRE LO APRENDIDO Y LOGRADO

Como ultima sección del proyecto hablaremos un poco acerca de las conclusiones de todo este trabajo de investigación y desarrollo. En primer lugar, ya hemos destacado y mencionado las cosas que hemos podido desarrollar e implementar de este proyecto. Para

este trabajo he involucrado multitud de aptitudes propias sobre todo de la Ingeniería Electrónica, las cuales, partiendo de una base desarrollada en el transcurso de la carrera, he podido profundizar más en ciertas disciplinas, así como aprender y mejorar en múltiples ámbitos de esta rama de la Ingeniería, las cuales paso a destacar:

- Electrónica Analógica: He adquirido más conocimientos en este ámbito, puesto que he requerido de realizar cálculos de filtros, de ganancia de amplificadores, así como analizar la respuesta en frecuencia de un transistor NPN.
- Electrónica de Potencia: Conocimientos en la implementación de rectificadores de media onda, así como la implementación de un condensador para únicamente ver el máximo de la señal.
- Teoría de circuitos: Investigación y profundización en los circuitos RLC, conocimientos de los fundamentos teóricos de la frecuencia de resonancia, así como los fundamentos teóricos de la capacidad y la inductancia.
- Sistemas Informáticos: He podido también ampliar y practicar la programación enfocada sobre todo en C y C++, pero también he realizado un programa de análisis de datos en un entorno similar a Matlab, llamado Julia.
- Tecnología de circuitos impresos: He puesto en marcha los conocimientos adquiridos en esta asignatura y he profundizado más a la hora de diseñar una PCB, de qué factores son importantes para reducir capacidades parásitas, de la importancia del DRC a la hora de mandar el diseño a fabricar, también he practicado la soldadura en componentes de agujero pasante.

Ya hemos hablado de las aptitudes que he podido aprender y mejorar relacionado con la Ingeniería Electrónica, ahora como conclusión repasaremos un poco lo que hemos logrado con este proyecto.

- Partimos de una idea, que era comprobar cómo mejorar la calidad de vida de las personas mayores a través de la domótica.
- Introducimos un proyecto de un artículo de Disney que podría cubrir esto mismo, ya que propone la sensorización de objetos cotidianos, en los cuales pueden crear una interfaz de usuario táctil, que pueda identificar distintas interacciones en dicho objeto

- Implementamos el circuito, y en base a la investigación, prueba y error logramos obtener un circuito que cumpliera lo mencionado en el apartado anterior. Un circuito que utiliza sensorización capacitiva por barrido de frecuencias para identificar gestualidad en un objeto sensorizado, cuya detección está sustentada por un circuito RLC, donde detectamos valles provocados por la frecuencia de resonancia.
- Logramos distinguir gestos realizados en una bola metálica (semejante al pomo de una puerta), una silla y una cama.
- Conseguimos demostrar cómo hacer el circuito sensible a cambios de inductancia, así como ver cómo podría ser útil a la hora de detectar que usuario es el que está interactuando con un entorno sensorizado.
- Conseguimos también optimizar el barrido de frecuencias, puesto que, en vez de hacer un barrido continuo, aplicamos el barrido a las zonas donde sabemos que se produce una actividad o gesto, de esta manera conseguimos reducir el número de pasos (en un inicio 200)
- También logramos implementar un programa que reconociese los gestos realizados en un entorno dado (bola, silla, cama), así como poder enviar un correo electrónico a una persona en particular si detecta cierta actividad en una franja horaria determinada.

6.2. LIMITACIONES DEL ESTUDIO Y DE LA SOLUCIÓN PROPUESTA

Una de las limitaciones del proyecto es algo que mencionamos en anteriores apartados, y es que no podemos visualizar todos los gestos que queramos en un objeto sensorizado, sino que está limitado, puesto que para dos gestos distintos puede ser que nuestro tacto esté induciendo la misma capacidad, por lo que no los distinguiremos.

Otra limitación que podría considerarse, aunque de manera parcial, es la dificultad que presenta Touché al implementarse en modo capacitivo para identificar la persona que interactúa con un entorno. Esto se debe a que la capacidad inducida por un cuerpo en relación con otro varía dentro de una franja estrecha de frecuencias. En otras palabras, al utilizar Touché en un entorno capacitivo, resulta bastante complicado discernir qué persona está interactuando. Por ejemplo, al sentarse en una silla sensorizada, levantarse y volver a sentarse, la capacidad inducida no será exactamente la misma en cada instancia,

ya que el cuerpo humano es un entorno complejo y experimenta cambios constantes a nivel interno. Esta complejidad hace que la detección precisa de un cuerpo en comparación con otro sea un desafío.

Y un inconveniente añadido es que todo lo metálico que sensoricemos tiene que estar recubierto por un aislante, si queremos hacer gestos mediante la piel humana directamente sobre un entorno metálico, el circuito no se comportará como debe. Esto no supone mucho problema, ya que podemos usar como aislante algún esmalte que el cual haga que no se perciba que lleva puesto dicho esmalte si queremos mantener el tono metálico del objeto.

Desde el punto de vista económico no será como tal una limitación, pero es algo para tener en cuenta si se desea implementar este circuito a muchos objetos o entornos, puesto que se necesitará un circuito específico por cada uno. A continuación, presentaremos una tabla de todos los componentes usados y con sus respectivos precios (Tabla 1:Tabla 1:Precio):

ESP32	19,95€
Generador de ondas AD9833	2,40€
2 chips de amplificador operacional AD8032A	13€
Resistencias (7 en total)	1€ (Valor aproximado puesto que siempre se piden en paquetes de muchas unidades)
Condensadores (5 electrolíticos y 1 cerámico)	1€ (Valor aproximado puesto que siempre se piden en paquetes de muchas unidades)
Transistor	0,15€
Bobina (470uH)	1€
PCB	17,30€
TOTAL	55,80€

Tabla 1: Precio del Circuito

Hay que tener en cuenta que estos precios son en base al diseño final, pero ni de lejos ha sido el presupuesto que ha costado este proyecto, puesto que ha habido etapas en las que se han probado otros componentes que se tuvieron que pedir (como diferentes bobinas, resistencias o condensadores), hicimos y enviamos a fabricar un diseño inicial de la PCB que al final no funcionó. El presupuesto que hemos mencionado antes está hecho en base a componentes únicos, pero esto a nivel real no es así, puesto que los pedidos se hacen al por mayor, es decir, de la PCB no solo llegó una, sino que llegaron 5 PCBs y así con todos los componentes.

6.3. SUGERENCIAS PARA FUTUROS TRABAJOS

Ya estamos llegando a la recta final del proyecto, un proyecto que tuvo una duración de meses hasta conseguir el producto final. Ahora bien, este proyecto además de todas las posibilidades que ofrece y que hemos planteado, tiene la opción de que se le puede implementar muchas otras variantes, investigando más sobre el uso de la frecuencia de resonancia, intentando estabilizar más las señales o realizando el barrido de frecuencias en una franja mayor... A continuación, realizaré un listado de lo que a nivel personal pienso que se podría incidir y mejorar de cara a un futuro:

- Hacer Touché más sensible con más cantidad de gestos: Siempre hemos abordado Touché como un circuito único capaz de poder sensorizar un objeto o medio para distinguir gestos o posturas, pero hay actividades que teníamos que descartar por el mero hecho de que falseaban los resultados, puesto que las capacidades entre una actividad y otra eran muy cercanas. Pero, ahora bien, ¿porqué no empleamos dos circuitos en vez de uno? Vimos con anterioridad que pudimos sensorizar una silla con una determinada colocación de cable y eso nos permitía distinguir tres actividades distintas, pero, ¿y si colocamos dos circuitos?, uno que sensorice por ejemplo el respaldo y otro el asiento, y que una de estas dos placas se comporte como esclavo y envíe los datos a la otra que sea el maestro, esto nos daría lugar a tener un circuito maestro que lo que haría sería analizar dos señales independientes, para formar una única señal con más información, esto sin duda podría hacer que podamos detectar más gestos e incluso quien sabe, poder identificar qué persona está realizando los gestos de manera meramente capacitiva.

- Aumentar las funcionalidades de Touché: En este proyecto hemos enfocado Touché para poder sensorizar tres medios distintos: una bola, una silla y una cama. El objetivo principal siempre fue mejorar la calidad de vida de las personas mayores, esto se hizo, puesto que implementamos un sistema de mensajería el cual nos avisaba de la actividad que fuera dentro de una franja horaria. Ahora bien, se podrían aplicar muchas más funcionalidades, puesto que a raíz de tener el circuito e identificar gestos se nos abre un mundo de posibilidades, y podemos por ejemplo programar un LED que nos ponga una luz roja y emita un pitido cuando estamos en una posición incorrecta en un asiento, podemos introducir un LCD al circuito que ponga que “Vuelvo en 5 minutos” si hago la actividad” “Pellizco” en el pomo de una puerta, podemos incluso hacer que se apaguen automáticamente las luces de la habitación si nos tumbamos en la cama para irnos a dormir, o que se abra la cortina o la persiana de la habitación si a cierta hora del día pasamos de la actividad “tumbado” a “levantado”. Más funcionalidades, por ejemplo, podría ser que, si realizamos la actividad “SentadoNormal”, en una silla o sillón se encienda automáticamente la televisión y que, si pasamos a “SentadoHaciadelante”, se apague. Como vemos hay un sinfín de funcionalidades que se pueden implementar para domotizar y sobre todo mejorar la calidad de vida ya no solo de las personas mayores, sino de las personas en general.
- Sensorizar otros entornos: Esta tecnología se puede emplear no solo para sensorizar una casa y convertirla en un entorno domótico, sino que se puede implementar prácticamente en todos los sectores y para todo tipo de aplicaciones.
 - Seguridad: Se puede implementar por ejemplo en el volante de un coche y que este emita ruido si no tenemos al menos una de las dos manos en dicho volante.
 - Salud: Se puede implementar en un hospital para controlar de manera sencilla todas las camas y saber que pacientes están levantados en sus respectivas habitaciones y cuáles no.
 - Entretenimiento: Se podrían crear juegos basados en esta tecnología, como, por ejemplo, una idea podría ser tener un panel sensorizado y que mediante gestos puedas manejar un coche de juguete, para que vaya a la izquierda, para la derecha, hacia delante...

- Otros medios: Este sistema sería un poco más complejo, pero teóricamente se podría crear unas pulseras atadas a Touché que nos permitiera conectar el circuito a un teléfono móvil, y mediante la interacción de la pulsera variar las acciones del móvil, por ejemplo. También podríamos cambiar una canción únicamente tocando con el dedo índice y corazón la pulsera, o llevando el dedo índice a la comisura de los labios para poner el móvil en silencio. Esto requeriría de la implementación en Touché de un módulo Bluetooth que permita la conexión del circuito con el móvil.
- Personas con problemas de movilidad: Esta tecnología podría usarse también para poder ayudar a personas que tengan problemas de motricidad para adaptarse mejor al entorno que le rodea.



7. BIBLIOGRAFÍA

[1] Contexto Histórico de la tecnología táctil

https://es.wikipedia.org/wiki/Pantalla_t%C3%A1ctil

[2] Tipos de sensores táctiles y sus características

<https://www.electricaltechnology.org/2022/08/tactile-sensor.html>

[3] Munehiko Sato, I. P. (2012). *TOUCHÉ: ENHANCING TOUCH INTERACTION ON HUMANS, SCREENS, LIQUIDS, AND EVERYDAY OBJECTS*. Austin, Texas: Disney Research

[4] Funcionamiento de las pantallas capacitivas proyectadas

<https://www.electricity-magnetism.org/es/pantalla-tactil-capacitiva-proyectada/>

[5] Funcionamiento de las cámaras TOF 3D

<https://www.xataka.com/moviles/camaras-tof-3d-como-funciona-que-atractiva-ultima-gran-innovacion-smartphones-fotograficos>

[6] Universidad Nacional de Rosario. (2016). *RESPUESTA EN FRECUENCIA*. Rosario.

[7] Fundamento teórico de las Ecuaciones de Maxwell

<https://es.wikipedia.org/wiki/Inductancia>

[8] Fundamentos teóricos de la Ley de Hooke

https://es.wikipedia.org/wiki/Ley_de_elasticidad_de_Hooke

[9] Reglas de diseño de una PCB (DRC)

<https://www.geekfactory.mx/tutoriales-eagle/reglas-de-diseno-drc-en-eagle/>

[10] Información y características de Visual Studio Code

<https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>

[11] Datasheet del microprocesador ESP32 Huzzah Feather

<https://cdn-learn.adafruit.com/downloads/pdf/adafruit-huzzah32-esp32-feather.pdf>

[12] Datasheet del Generador de ondas AD9833

<https://pdf1.alldatasheet.com/datasheet-pdf/view/48600/AD/AD9833.html>

[13] Datasheet del Amplificador AD8032A

<https://pdf1.alldatasheet.com/datasheet-pdf/view/48386/AD/AD8032.html>

[14] Fundamentos de Rectificador de media onda con condensador

http://www.sc.ehu.es/sbweb/electronica/elec_basica/tema4/Paginas/Pagina7.htm

[15] Datasheet del Transistor NPN BC546B

<https://pdf1.alldatasheet.es/datasheet-pdf/view/99373/ONSEMI/BC546B.html>

[16] Principios del uso de condensadores de desacoplo

<https://www.learningaboutelectronics.com/Articulos/Condensador-de-desacoplo.php>



ANEXOS

ANEXO I: CÁLCULOS

ANEXO II: CÓDIGO



ANEXO I: CÁLCULOS

FRECUENCIAS DE CORTE DE LOS FILTROS:

1: Filtro paso bajo:

- $f_c = \frac{1}{2\pi RC}$
- $f_c = \frac{1}{2\pi \cdot 330 \cdot 100 \cdot 10^{-12}}$
- $f_c = 4822877,06 \text{ Hz}$

2: Filtro paso alto

- $f_c = \frac{1}{2\pi RC}$
- $f_c = \frac{1}{2\pi \cdot 10000 \cdot 22 \cdot 10^{-9}}$
- $f_c = 723,43 \text{ Hz}$

CÁLCULO DE LOS COMPONENTES DE LOS AMPLIFICADORES OPERACIONALES

1: Amplificador de 1 etapa (Figura 82):

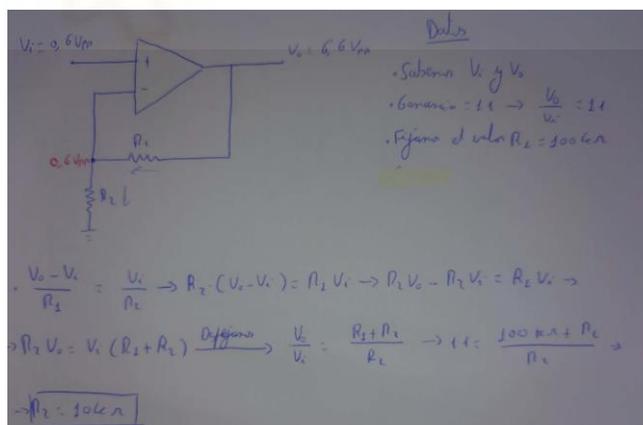


Figura 82: Cálculos de amplificador de 1 etapa

2: Amplificador de 2 etapas (Figura 83):

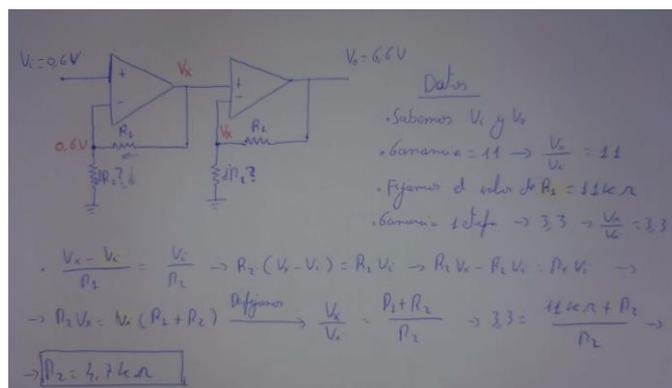


Figura 83: Cálculos del amplificador de 2 etapas

CÁLCULO DE LAS CAPACIDADES PARÁSITAS DE LA PCB Y EL OBJETO CONDUCTOR MEDIANTE LA FRECUENCIA DE RESONANCIA.

- Frecuencia de resonancia de la actividad “Sin tocar” en el cable, según el apartado 4.1.3.4: ~2,8 MHz

- $f_0 = \frac{1}{2\pi\sqrt{LC}}$

- $2,8 \cdot 10^6 = \frac{1}{2\pi\sqrt{470 \cdot 10^{-6} \cdot C}}$

- $C_{\text{parásita}} = 6,87pF$

ANEXO 2: CÓDIGO

CÓDIGO PARA OBTENER LA SEÑAL EN EL CONVERTOR A UNA FRECUENCIA FIJA

```

1 #include <AD9833.h> // Include the library
2 #include <SPI.h>
3 #define FNC_PIN 4 // Can be any digital IO pin
4
5 const int pinEntradaAnalogica = 34;
6 int Frecuencia = 300000; // Change the frequency to the one you want
7
8 //----- Create an AD9833 object -----
9 // Note, SCK and MOSI must be connected to CLK and DAT pins on the AD9833 for SPI
10 AD9833 gen(FNC_PIN); // Defaults to 25MHz internal reference frequency
11
12 void setup()
13 {
14     // This MUST be the first command after declaring the AD9833 object
15     gen.Begin();
16     // Apply a 1000 Hz sine wave using REG0 (register set 0). There are two register
17     sets,
18     // REG0 and REG1.
19     // Each one can be programmed for:
20     //   Signal type - SINE_WAVE, TRIANGLE_WAVE, SQUARE_WAVE, and HALF_SQUARE_WAVE
21     //   Frequency - 0 to 12.5 MHz
22     //   Phase - 0 to 360 degrees (this is only useful if it is 'relative' to some other
23     signal
24     //   such as the phase difference between REG0 and REG1).
25     // In ApplySignal, if Phase is not given, it defaults to 0.
26     gen.ApplySignal(SINE_WAVE, REG1, Frecuencia); // Aplicamos una onda senoidal en el
27     registro Reg0 o Reg1 y le damos la frecuencia en hz que queramos(0 a 12,5Mhz)
28     gen.EnableOutput(true); // Turn ON the output - it defaults to
29     OFF
30     // There should be a 1000 Hz sine wave on the output of the AD9833
31     Serial.begin(115200); // Velocidad del puerto serieD
32     while (!Serial)
33         ;
34     pinMode(pinEntradaAnalogica, INPUT); // Indicate the pin that we use
35 }
36
37 void loop()
38 {
39     float ValorAnalogico = 0;
40     float ValorEscalado = 0;
41     ValorAnalogico = analogRead(pinEntradaAnalogica);
42     ValorEscalado = (ValorAnalogico * 3.3) / 4095;
43     Serial.println(ValorEscalado);
44     delay(100);
45 }

```

CÓDIGO PARA OBTENER LA SEÑAL EN EL CONVERTOR EN EL BARRIDO DE FRECUENCIAS (17,5KHz-3,5MHz) CON PASOS DE 17,5KHz (200 PASOS)

```

1 #include <AD9833.h> // Include the library
2 #include <SPI.h>
3 #include "BluetoothSerial.h"
4 #define FNC_PIN 4 // Can be any digital IO pin
5
6 const int pinEntradaAnalogica = 34;
7 int ValorFrecuencia = 17500;
8 int Frecuencia = ValorFrecuencia;
9 int barridoFrecuencia = Frecuencia;
10
11 //----- Create an AD9833 object -----
12 // Note, SCK and MOSI must be connected to CLK and DAT pins on the AD9833 for SPI
13 AD9833 gen(FNC_PIN); // Defaults to 25MHz internal reference frequency
14
15 void setup()
16 {
17     // This MUST be the first command after declaring the AD9833 object
18     gen.Begin();
19
20     // Apply a 1000 Hz sine wave using REG0 (register set 0). There are two register sets,
21     // REG0 and REG1.
22     // Each one can be programmed for:
23     //   Signal type - SINE_WAVE, TRIANGLE_WAVE, SQUARE_WAVE, and HALF_SQUARE_WAVE
24     //   Frequency - 0 to 12.5 MHz
25     //   Phase - 0 to 360 degrees (this is only useful if it is 'relative' to some other signal
26     //           such as the phase difference between REG0 and REG1).
27     // In ApplySignal, if Phase is not given, it defaults to 0.
28
29     gen.ApplySignal(SINE_WAVE, REG0, Frecuencia); // We apply a sine wave in the Reg0 or Reg1
30     register and give it the frequency in hz we want (0 to 12.5Mhz).
31
32     gen.EnableOutput(true); // Turn ON the output - it defaults to OFF
33     // There should be a 1000 Hz sine wave on the output of the AD9833
34     Serial.begin(115200); // Serial Port speed
35     while (!Serial);
36     pinMode(pinEntradaAnalogica, INPUT); /// Indicate that the pin we use is an input
37 }
38 void loop()
39 {
40     int numLecturasdato = 200; // Number of sweep steps
41     int numValoresporvector = 15; // Number of samples per frequency
42     char iteracion = 'a'; // Iteration number from a(1) to j(10)(change the letter from
43     there to a, b, c... depending on which iteration we are doing)
44
45     int i = 0, j = 0;
46     float ValorEscalado = 0; // Scaling from 0 to 4095 to 0 to 3.3V of ESP32
47     Serial.print("x="); // We set the x-axis which are the frequencies with the following for
48     for (int i = 0; i < numLecturasdato; i++)
49     {
50         Serial.printf("%i", barridoFrecuencia);
51         barridoFrecuencia = barridoFrecuencia + ValorFrecuencia;
52         if (i < numLecturasdato - 1)
53         {
54             Serial.print(",");
55         }
56     }
57     Serial.print("\n");
58     Serial.println("Amplitud");
59     Serial.printf("%c=", iteracion);

```

```

60 // We get the scaled data from the Analogue to Digital Converter with the following for
61 for (int i = 0; i < numLecturasdato;)
62 {
63     int ValorAnalogico = analogRead(pinEntradaAnalogica);
64     ValorEscalado = (ValorAnalogico * 3.2) / 4095;
65     Serial.print(ValorEscalado, 4);
66     j++;
67     if (j < numValoresporvector)
68     {
69         Serial.print(" ");
70     }
71     delay(1); // Delay from data to data within the same frequency
72     if (j > numValoresporvector - 1)
73     {
74         if (i < numLecturasdato - 1)
75         {
76             Serial.print("\n");
77         }
78         j = 0;
79         Frecuencia = Frecuencia + ValorFrecuencia;
80         if (i < numLecturasdato - 1)
81         {
82             gen.ApplySignal(SINE_WAVE, REG0, Frecuencia);
83         }
84         i++;
85         delay(1); // Delay between frequencies
86     }
87 }
88 Serial.print("");
89 Serial.print("\n");
90
91 while (true)
92 { // Stop the program
93     delay(1000);
94 }
95 }

```



-Visualización por pantalla:

```

x=[17500,35000,52500,70000,87500,105000,122500,140000,157500,175000,192500,210000,227500,245000,262500,280000,297500,315000,332500,
350000,367500,385000,402500,420000,437500,455000,472500,490000,507500,525000,542500,560000,577500,595000,612500,630000,647500,665000,
682500,700000,717500,735000,752500,770000,787500,805000,822500,840000,857500,875000,892500,910000,927500,945000,962500,980000,997500,
1015000,1032500,1050000,1067500,1085000,1102500,1120000,1137500,1155000,1172500,1190000,1207500,1225000,1242500,1260000,1277500,
1295000,1312500,1330000,1347500,1365000,1382500,1400000,1417500,1435000,1452500,1470000,1487500,1505000,1522500,1540000,1557500,
1575000,1592500,1610000,1627500,1645000,1662500,1680000,1697500,1715000,1732500,1750000,1767500,1785000,1802500,1820000,1837500,
1855000,1872500,1890000,1907500,1925000,1942500,1960000,1977500,1995000,2012500,2030000,2047500,2065000,2082500,2100000,2117500,
2135000,2152500,2170000,2187500,2205000,2222500,2240000,2257500,2275000,2292500,2310000,2327500,2345000,2362500,2380000,2397500,
2415000,2432500,2450000,2467500,2485000,2502500,2520000,2537500,2555000,2572500,2590000,2607500,2625000,2642500,2660000,2677500,
2695000,2712500,2730000,2747500,2765000,2782500,2800000,2817500,2835000,2852500,2870000,2887500,2905000,2922500,2940000,2957500,
2975000,2992500,3010000,3027500,3045000,3062500,3080000,3097500,3115000,3132500,3150000,3167500,3185000,3202500,3220000,3237500,
3255000,3272500,3290000,3307500,3325000,3342500,3360000,3377500,3395000,3412500,3430000,3447500,3465000,3482500,3500000]

Amplitud
a=[2.5131 2.5131 2.5092 2.5131 2.5061 2.5186 2.4998 2.5139 2.5030 2.5131 2.5061 2.5170 2.5014 2.5061 2.4936;
2.5303 2.5217 2.5178 2.5314 2.5186 2.5178 2.5209 2.5295 2.5295 2.5155 2.4928 2.5217 2.5170 2.4490 2.5217;
2.5209 2.5186 2.5131 2.5241 2.5186 2.5280 2.5201 2.5241 2.5170 2.5217 2.5147 2.5201 2.5248 2.5248 2.5092;
2.5225 2.5170 2.5241 2.5147 2.5381 2.5201 2.5303 2.5256 2.5201 2.5264 2.5326 2.4897 2.5241 2.5131 2.5131;
2.5248 2.5287 2.5178 2.5201 2.5225 2.5116 2.5201 2.5217 2.5147 2.5264 2.5217 2.5256 2.5248 2.5186 2.5147;
2.5201 2.5350 2.5256 2.4436 2.5186 2.5264 2.5209 2.5295 2.5217 2.5233 2.5186 2.5147 2.5170 2.5264 2.5209;
2.5170 2.5178 2.5217 2.5241 2.5256 2.5256 2.5248 2.5170 2.5264 2.5233 2.5155 2.5272 2.5233 2.5186 2.5233;
2.5233 2.5123 2.5139 2.5241 2.5241 2.5162 2.5209 2.5155 2.5139 2.5209 2.5162 2.5147 2.5201 2.5170 2.5178;
2.5178 2.5139 2.5178 2.5217 2.5131 2.5131 2.5186 2.5131 2.5147 2.5217 2.5170 2.5248 2.5186 2.5975 2.5209;
2.5178 2.5170 2.5131 2.5147 2.5131 2.5131 2.5131 2.5147 2.5155 2.5155 2.5084 2.5139 2.5131 2.5116 2.5100;
2.5147 2.5131 2.5186 2.5092 2.5155 2.5201 2.5233 2.5209 2.5131 2.5170 2.5155 2.5178 2.5147 2.5139 2.5170;
2.5155 2.5162 2.5209 2.5131 2.5131 2.5186 2.5170 2.5170 2.5217 2.5186 2.5241 2.5147 2.5155 2.5155 2.5241;
2.5108 2.5209 2.5123 2.5131 2.5131 2.5155 2.5061 2.5061 2.5131 2.5069 2.5186 2.5092 2.5084 2.5186 2.5147;
2.5053 2.5053 2.4662 2.5116 2.5108 2.5061 2.5139 2.5061 2.5131 2.5061 2.5123 2.5030 2.5123 2.5022 2.5061;
2.5108 2.5061 2.5084 2.5061 2.5092 2.5022 2.5053 2.5014 2.5131 2.5084 2.5014 2.5116 2.5116 2.5006 2.5084;
2.4975 2.4889 2.5014 2.5022 2.5045 2.5006 2.5014 2.5092 2.5030 2.5053 2.4998 2.4983 2.5014 2.4936 2.5061;
2.4905 2.4959 2.4936 2.5373 2.5045 2.4225 2.5014 2.4998 2.4905 2.5030 2.4936 2.4990 2.4983 2.4998 2.4881;
2.4990 2.4373 2.5030 2.4936 2.5014 2.4865 2.4897 2.4951 2.4748 2.4990 2.5014 2.5006 2.4905 2.4920 2.4967;
2.4881 2.4795 2.5030 2.4897 2.4873 2.4858 2.4959 2.4858 2.4936 2.4920 2.4905 2.4811 2.4936 2.4873 2.4920;
2.4826 2.4881 2.4811 2.4795 2.4764 2.4803 2.4865 2.4748 2.4740 2.4811 2.4803 2.4779 2.4834 2.4709 2.4772;
2.4686 2.3842 2.4764 2.4647 2.4631 2.4717 2.4670 2.4709 2.4670 2.4725 2.4811 2.4740 2.4678 2.4608 2.4654;
2.4631 2.4654 2.4608 2.4686 2.4639 2.4639 2.5631 2.4631 2.4670 2.4615 2.4123 2.4639 2.4639 2.4584 2.4631;
2.4561 2.4561 2.4568 2.4584 2.4529 2.4506 2.4506 2.4483 2.4553 2.4483 2.4623 2.4436 2.4529 2.4561 2.4561;
2.4475 2.4381 2.4373 2.4436 2.4295 2.4420 2.4350 2.4404 2.4365 2.4334 2.4358 2.4358 2.3787 2.4412 2.4232;
2.4264 2.4225 2.4279 2.4303 2.4248 2.4248 2.4279 2.4873 2.4303 2.4232 2.4232 2.4131 2.4217 2.4217 2.4201;
2.4131 2.4092 2.4139 2.4154 2.4154 2.4139 2.4123 2.4131 2.4131 2.4264 2.4217 2.4115 2.4131 2.4178 2.4147;
2.4061 2.3998 2.4858 2.3998 2.4053 2.3936 2.3998 2.4053 2.3951 2.4358 2.3951 2.3951 2.3967 2.3873 2.3920;
2.3842 2.3771 2.3748 2.3826 2.3834 2.3881 2.3263 2.3834 2.3873 2.3803 2.4787 2.3865 2.3811 2.3857 2.3857;
2.3670 2.3795 2.3654 2.3678 2.3662 2.3639 2.3631 2.4131 2.3631 2.3615 2.3756 2.3639 2.3685 2.3631 2.3639;
2.3474 2.3459 2.3521 2.3514 2.3420 2.3506 2.3545 2.3451 2.3381 2.3451 2.3459 2.3498 2.3467 2.3482 2.3498;
2.3381 2.3178 2.3349 2.3303 2.3342 2.3303 2.3295 2.3279 2.3232 2.3310 2.3271 2.3193 2.3310 2.3326 2.3256;
2.3037 2.3060 2.3217 2.2998 2.3045 2.3123 2.3099 2.3131 2.3107 2.3131 2.3013 2.3053 2.3115 2.3084 2.3006;
2.2756 2.2857 2.2802 2.2873 2.2834 2.2756 2.2873 2.2740 2.2740 2.2756 2.2834 2.2810 2.2865 2.2779 2.2795;
2.2459 2.2521 2.2498 2.2490 2.2513 2.2529 2.2631 2.2466 2.2537 2.2482 2.2466 2.2498 2.2545 2.2466 2.2623;
2.2146 2.2123 2.2130 2.2091 2.2099 2.2130 2.2115 2.2177 2.2130 2.2052 2.2099 2.2130 2.2050 2.2201 2.2130;
2.1685 2.1779 2.1880 2.1794 2.1505 2.1732 2.1497 2.1662 2.1716 2.1740 2.1716 2.1747 2.1709 2.1794 2.1747;
2.1404 2.1263 2.1294 2.1365 2.1302 2.1302 2.1247 2.1310 2.1333 2.1357 2.1255 2.1279 2.1302 2.1349 2.0810;

```

2.0755	2.0810	2.0872	2.0880	2.0864	2.0825	2.0833	2.0818	2.0849	2.0747	2.0810	2.0857	2.0216	2.0857	2.0880;
2.0294	2.0216	2.0232	2.0232	2.0216	2.0286	2.0255	2.0357	2.0247	2.0294	2.0239	2.0333	1.9520	2.0302	2.0325;
1.9544	1.9544	1.9575	1.9575	1.9497	1.9536	1.9559	1.9559	1.9442	1.9567	1.9513	1.9505	1.9059	1.9505	1.9520;
1.8762	1.8762	1.8762	1.8778	1.8700	1.8747	1.8762	1.8778	1.8739	1.8692	1.8770	1.8747	1.8739	1.8747	1.8762;
1.7840	1.7801	1.7856	1.7809	1.7872	1.7879	1.7848	1.7801	1.7832	1.7879	1.7801	1.7879	1.7793	1.7786	1.7879;
1.6676	1.6723	1.6754	1.6770	1.6653	1.6746	1.6715	1.6793	1.6754	1.6699	1.6770	1.6699	1.6684	1.6738	1.6668;
1.5543	1.5457	1.5394	1.5488	1.5418	1.5496	1.5473	1.5496	1.5426	1.5418	1.5379	1.5504	1.5496	1.5457	1.5426;
1.4128	1.4041	1.3989	1.4003	1.4027	1.4089	1.4050	1.3988	1.3964	1.3996	1.4011	1.4089	1.3989	1.4003	1.4027;
1.2753	1.2464	1.2448	1.2503	1.2503	1.2495	1.2495	1.2472	1.2558	1.2456	1.2487	1.2409	1.2503	1.2526	1.2401;
1.1159	1.0776	1.0792	1.0854	1.0807	1.0768	1.0800	1.0768	1.0878	1.0729	1.0815	1.0784	1.0760	1.0823	1.0776;
0.9745	0.9284	0.9260	0.9260	0.9244	0.9276	0.9244	0.9229	0.9237	0.9213	0.9213	0.9244	0.9229	0.9252	0.9213;
0.8275	0.7877	0.7955	0.7924	0.7877	0.7955	0.7986	0.7979	0.7877	0.7861	0.7877	0.7924	0.7979	0.7885	0.7916;
0.7275	0.7111	0.7103	0.7088	0.7072	0.7080	0.7072	0.6955	0.7142	0.7127	0.7127	0.7080	0.7080	0.7119	0.7135;
0.6877	0.6916	0.6892	0.6892	0.6877	0.6877	0.6869	0.6900	0.6877	0.6830	0.6877	0.6853	0.6838	0.6916	0.6877;
0.6736	0.7228	0.7244	0.7306	0.7236	0.7275	0.7361	0.7306	0.7252	0.7127	0.7306	0.7236	0.7228	0.7267	0.7275;
0.8127	0.8189	0.8252	0.8197	0.8275	0.8205	0.8166	0.8150	0.8182	0.8213	0.8268	0.8150	0.8275	0.8463	0.8244;
0.9393	0.9385	0.9401	0.9377	0.8877	0.9393	0.9416	0.9479	0.9377	0.9393	0.9377	0.9424	0.9377	0.9455	0.9432;
1.0776	1.0753	1.0768	1.0823	1.0760	1.0737	1.0776	1.0768	1.0768	1.0768	1.0745	1.0745	1.0815	1.0854	1.0800;
1.2167	1.2128	1.2198	1.2159	1.2198	1.2167	1.2112	1.2269	1.2159	1.2214	1.2128	1.2144	1.2151	1.2237	1.2237;
1.3511	1.3519	1.3503	1.3511	1.3480	1.3425	1.3527	1.3542	1.3558	1.3527	1.3480	1.3550	1.3495	1.3511	1.3503;
1.4629	1.4707	1.4675	1.4675	1.4683	1.4707	1.4636	1.4738	1.4738	1.4707	1.4699	1.4715	1.4707	1.4715	1.4644;
1.5777	1.5762	1.5730	1.5855	1.5668	1.5793	1.5832	1.5746	1.5754	1.5746	1.5762	1.5769	1.5793	1.5730	1.5769;
1.6684	1.6731	1.6738	1.6715	1.6715	1.6723	1.6645	1.6699	1.6707	1.6707	1.6238	1.6738	1.6723	1.6793	1.7145;
1.7536	1.7582	1.7614	1.7809	1.7590	1.7543	1.7582	1.7582	1.7543	1.7551	1.7614	1.7606	1.7590	1.7614	1.7590;
1.8309	1.8278	1.8239	1.8254	1.8309	1.8254	1.8372	1.8637	1.8333	1.8301	1.8340	1.8356	1.8270	1.8247	1.8239;
1.8887	1.8880	1.8895	1.8950	1.8887	1.8895	1.8856	1.8903	1.8903	1.8966	1.8864	1.8903	1.8942	1.8895	1.8989;
1.9450	1.9419	1.9466	1.9434	1.9450	1.9427	1.9559	1.9481	1.9497	1.9403	1.9481	1.9450	1.9395	1.9434	1.9380;
1.9919	1.9935	1.9903	1.9958	1.9880	1.9903	1.9864	1.9903	1.9864	1.9981	1.9880	1.9888	1.9950	1.9950	1.9849;
2.0372	2.0364	2.0380	2.0419	2.0849	2.0341	2.0357	2.0380	2.0372	2.0341	2.0380	2.0357	2.0341	2.0364	2.0380;
2.0708	2.0646	2.0747	2.0716	2.0739	2.0724	2.0755	2.0653	2.0716	2.0716	2.0700	2.0630	2.0747	2.0739	2.0263;
2.0966	2.1052	2.0997	2.0974	2.1294	2.1005	2.1044	2.0904	2.0904	2.0521	2.1160	2.1095	2.1013	2.1044	2.0997;
2.1294	2.1294	2.1185	2.1255	2.1255	2.1326	2.1302	2.1302	2.1240	2.1255	2.1232	2.1286	2.1271	2.1279	2.1255;
2.1122	2.1482	2.1529	2.1505	2.1529	2.1474	2.1427	2.1466	2.1451	2.1583	2.1732	2.1638	2.1521	2.1544	2.1685;
2.1708	2.1701	2.1755	2.1787	2.1755	2.1646	2.1708	2.1732	2.2005	2.1701	2.1701	2.1742	2.1747	2.1701	2.1755;
2.1919	2.2107	2.1810	2.1958	2.1927	2.1966	2.1919	2.1935	2.1505	2.1404	2.1919	2.1974	2.1935	2.1958	2.1880;
2.2154	2.2146	2.2138	2.2146	2.2076	2.2084	2.2091	2.2123	2.2146	2.2130	2.2130	2.2154	2.2201	2.2091	2.2146;
2.2334	2.2302	2.2263	2.2255	2.2295	2.2263	2.2287	2.2287	2.2310	2.2263	2.2310	2.2310	2.2279	2.2295	2.2318;
2.2607	2.2505	2.2537	2.2474	2.2451	2.2505	2.2396	2.2380	2.2435	2.2490	2.2560	2.2357	2.2420	2.2513	2.2552;
2.2545	2.2631	2.2607	2.2623	2.2646	2.2623	2.2576	2.2560	2.2607	2.2615	2.2599	2.2654	2.2599	2.2631	2.2631;
2.2685	2.2849	2.2732	2.2716	2.2654	2.2685	2.2756	2.2732	2.2795	2.2795	2.2740	2.2857	2.2646	2.2716	2.2810;
2.2842	2.2779	2.2810	2.2857	2.2849	2.2873	2.2857	2.2834	2.2842	2.2857	2.2834	2.2857	2.2865	2.2771	2.2826;
2.2927	2.2990	2.2927	2.2888	2.2967	2.2950	2.2959	2.2891	2.2927	2.2951	2.3263	2.2927	2.2959	2.2896	2.2999;
2.3029	2.2982	2.3029	2.3020	2.2998	2.3045	2.3037	2.2967	2.3045	2.3006	2.3076	2.3092	2.3045	2.3060	2.2982;
2.3084	2.3123	2.3115	2.3154	2.3138	2.3178	2.3146	2.3131	2.3115	2.3029	2.3060	2.3154	2.3131	2.3076	2.3013;
2.3154	2.3146	2.3185	2.3240	2.3185	2.3201	2.3263	2.3178	2.3170	2.3217	2.2638	2.3131	2.3154	2.3185	2.3131;
2.3248	2.3232	2.3217	2.2881	2.3240	2.2763	2.3224	2.3209	2.3303	2.3099	2.3271	2.3217	2.3279	2.3263	2.3263;
2.3373	2.3271	2.3310	2.3381	2.4264	2.3310	2.3357	2.3357	2.3318	2.3303	2.3381	2.3217	2.3006	2.3373	2.3295;
2.3373	2.3412	2.3373	2.3357	2.3435	2.2942	2.3349	2.3381	2.3373	2.3381	2.3209	2.3428	2.3342	2.3303	2.3498;
2.3420	2.3295	2.3490	2.3506	2.3498	2.3498	2.3435	2.3420	2.3443	2.3435	2.3428	2.3428	2.3404	2.3451	2.3404;
2.3514	2.3435	2.3451	2.3560	2.3521	2.3545	2.3498	2.3521	2.3498	2.3560	2.3592	2.3764	2.3514	2.3521	2.3498;
2.3498	2.3592	2.3631	2.3631	2.3498	2.3553	2.3584	2.3631	2.3607	2.3506	2.3670	2.3573	2.3584	2.3560	2.3600;
2.3670	2.3560	2.3584	2.3631	2.3592	2.3615	2.3607	2.3639	2.3545	2.3631	2.3615	2.3654	2.3592	2.3639	2.3592;
2.3646	2.3701	2.3646	2.3654	2.3646	2.3654	2.3631	2.3654	2.3685	2.3670	2.3701	2.3639	2.3654	2.3748	2.3623;
2.3685	2.3732	2.3701	2.3764	2.3701	2.3740	2.3678	2.3779	2.3740	2.3662	2.3639	2.3639	2.3654	2.3779	2.3779;
2.3842	2.3709	2.3732	2.3764	2.3701	2.3770	2.3740	2.3756	2.3811	2.3717	2.3764	2.3756	2.3771	2.3748	2.3709;
2.3342	2.3748	2.3857	2.3982	2.3725	2.3803	2.3850	2.3826	2.3826	2.3787	2.3764	2.3803	2.3795	2.3779	2.3717;
2.2935	2.3764	2.3709	2.3787	2.3842	2.3803	2.3857	2.3881	2.3834	2.3803	2.3834	2.3834	2.3881	2.3881	2.3881;
2.4303	2.3942	2.4170	2.3795	2.3881	2.3842	2.3896	2.3873	2.3842	2.3217	2.3826	2.3896	2.3842	2.3889	2.3850;
2.3881	2.3881	2.3873	2.3896	2.3873	2.3865	2.3881	2.3873	2.3842	2.4365	2.3928	2.3834	2.3811	2.3811	2.3881;
2.3881	2.3881	2.3889	2.3365	2.3390	2.3904	2.3904	2.3990	2.3951	2.3912	2.3881	2.3889	2.3920	2.3920	2.4412;
2.3967	2.3928	2.3928	2.3881	2.3936	2.3959	2.3975	2.3881	2.3896	2.3928	2.3904	2.3936	2.3928	2.4061	2.3920;
2.3967	2.3936	2.4131	2.3998	2.4115	2.3967	2.3912	2.3936	2.3998	2.3920	2.3889	2.3889	2.3936	2.4061	2.4061;
2.3982	2.3998	2.3967	2.4045	2.4061	2.4053	2.3998	2.4076	2.3998	2.4076	2.3998	2.4053	2.4053	2.3912	2.3998;
2.3990	2.4123	2.4053	2.3982	2.4084	2.3990	2.4061	2.3998	2.3967	2.3998	2.4061	2.4092	2.3990	2.3967	2.3982;
2.4084	2.4123	2.4170	2.4147	2.4147	2.4131	2.4123	2.4092	2.4061	2.4061	2.4107	2.4053	2.4092	2.4084	2.4107;
2.4061	2.4115	2.3998	2.4131	2.4107	2.4178	2.3998	2.4061	2.4123	2.4131	2.4131	2.4139	2.4131	2.4186	2.4061;
2.4147	2.3959	2.4084	2.4131	2.4162	2.4131	2.4061	2.4092	2.4131	2.4270	2.4131	2.4131	2.4131	2.3998	2.4147;
2.4147	2.3451	2.4131	2.4240	2.4147	2.4084	2.4115	2.4170	2.4107	2.4123	2.4217	2.4139	2.4131	2.4178	2.4131;
2.4092	2.4154	2.4131	2.4186	2.4131	2.4209	2.4178	2.4131	2.4139	2.4131	2.4076	2.4131	2.4154	2.4178	2.4154;
2.4139	2.4303	2.4217	2.4147	2.4170	2.4170	2.4154	2.4264	2.4139	2.4115	2.4053	2.4139	2.4147	2.4209	2.4131;
2.4232	2.4139	2.4186	2.4154	2.4170	2.4170	2.4186	2.4217	2.4217	2.4147	2.4131	2.4248	2.4217	2.4225	2.4272;
2.4147	2.4201	2.4147	2.4334	2.4131	2.4225	2.4240	2.4193	2.4107	2.4709	2.4162	2.4225	2.4217	2.4209	2.4248;
2.4272	2.4334	2.4201	2.4295											

2.2849	2.2685	2.2732	2.2709	2.2631	2.2716	2.2709	2.2709	2.2709	2.2724	2.2748	2.2716	2.2724	2.2701	2.2685;
2.2545	2.2724	2.2631	2.2529	2.2357	2.2552	2.2420	2.2513	2.2318	2.2552	2.2529	2.2607	2.2881	2.2576	2.2529;
2.2443	2.2005	2.2466	2.2716	2.2474	2.2404	2.2490	2.2380	2.2482	2.2388	2.2435	2.2498	2.2380	2.2123	2.2388;
2.2295	2.2357	2.2310	2.2334	2.2279	2.2357	2.2310	2.1810	2.2310	2.2349	2.2334	2.2271	2.2216	2.2248	2.2295;
2.2123	2.2240	2.2107	2.2138	2.2216	2.2146	2.2138	2.2115	2.2130	2.2209	2.2177	2.2091	2.2138	2.2169	2.2169;
2.2021	2.2021	2.2044	2.2052	2.1982	2.2044	2.1958	2.1998	2.1998	2.2091	2.1998	2.1998	2.1998	2.2021	2.2021;
2.1990	2.1888	2.1810	2.1880	2.1880	2.1896	2.1880	2.1826	2.1841	2.1880	2.1857	2.1919	2.1880	2.1927	2.1865;
2.1732	2.1747	2.1701	2.1747	2.1732	2.1724	2.1732	2.1716	2.1755	2.1794	2.1779	2.1724	2.1896	2.1747	2.1927;
2.1591	2.1396	2.1599	2.1615	2.1630	2.1591	2.1630	2.1615	2.1622	2.1630	2.1583	2.1591	2.1591	2.1654	2.1622;
2.1411	2.1380	2.1474	2.1388	2.1279	2.1404	2.1466	2.1490	2.1396	2.1419	2.1458	2.1404	2.1396	2.1380	2.1427;
2.1357	2.1255	2.1302	2.1247	2.1294	2.1271	2.1279	2.1279	2.1263	2.1271	2.1247	2.1341	2.1279	2.1279	2.1302;
2.1146	2.1122	2.1146	2.1099	2.1138	2.1091	2.1146	2.1130	2.1107	2.1138	2.1130	2.1122	2.1107	2.1154	2.1099;
2.0982	2.1005	2.0919	2.1021	2.0982	2.1021	2.1013	2.0982	2.0950	2.1021	2.0935	2.0904	2.0982	2.0958	2.0904;
2.0763	2.0771	2.0810	2.0771	2.0864	2.0716	2.0857	2.0786	2.0794	2.0818	2.0771	2.0771	2.0880	2.0864	2.0786;
2.0653	2.0614	2.0630	2.0653	2.0700	2.0614	2.0651	2.0622	2.0630	2.0638	2.0630	2.0622	2.0630	2.0763	2.0646;
2.0521	2.0528	2.0513	2.0521	2.0505	2.0497	2.0544	2.0489	2.0505	2.0513	2.0497	2.0505	2.0630	2.0403	2.0544;
2.0380	2.0364	2.0349	2.0364	2.0341	2.0239	2.0294	2.0310	2.0349	2.0349	2.0294	2.0466	2.0294	2.0341	2.0372;
2.0177	2.0263	2.0146	2.0185	2.0130	2.0208	2.0177	2.0146	2.0200	2.0185	2.0185	2.0138	2.0185	2.0146	1.9997;
1.9966	1.9997	1.9958	1.9935	1.9981	1.9997	1.9981	1.9989	1.9966	1.9997	1.9958	2.0075	2.0075	1.9966	1.9958;
1.9841	1.9856	1.9802	1.9825	1.9802	1.9833	1.9770	1.9872	1.9950	1.9817	1.9778	1.9880	1.9872	1.9841	1.9810;
1.9739	1.9700	1.9638	1.9630	1.9731	1.9677	1.9716	1.9739	1.9638	1.9700	1.9708	1.9677	1.9653	1.9684	1.9653;
1.9489	1.9544	1.9505	1.9513	1.9544	1.9520	1.9528	1.9528	1.9466	1.9544	1.9528	1.9528	1.9544	1.9505	1.9505;
1.9309	1.9333	1.9380	1.9356	1.9380	1.9434	1.9372	1.9380	1.9372	1.9364	1.9341	1.9294	1.9645	1.9388	1.9294;
1.9161	1.9200	1.9137	1.9380	1.9153	1.9263	1.9177	1.9231	1.9200	1.9130	1.9161	1.9208	1.9208	1.9169	1.9169;
1.8950	1.9044	1.8997	1.9020	1.9005	1.9020	1.8966	1.9036	1.9012	1.9020	1.9044	1.9044	1.9059	1.9005	1.9005;
1.8887	1.8856	1.8887	1.8841	1.8887	1.8856	1.8903	1.8880	1.8833	1.8872	1.8880	1.8880	1.8833	1.8872	1.8903;
1.8669	1.8630	1.8731	1.8716	1.8669	1.8755	1.8731	1.8739	1.8708	1.8630	1.8637	1.8684	1.8590	1.8872	1.8770;
1.8544	1.8497	1.8497	1.8506	1.8544	1.8595	1.8590	1.8544	1.8544	1.8507	1.8536	1.8505	1.8536	1.8528	1.8544;
1.8403	1.8333	1.8395	1.8379	1.8348	1.8403	1.8333	1.8379	1.8372	1.8325	1.8356	1.8458	1.8387	1.8270	1.8379;
1.8004	1.8176	1.8223	1.8239	1.8231	1.8215	1.8208	1.8184	1.8129	1.8184	1.8129	1.8208	1.8105	1.7762	1.8184;
1.8043	1.8098	1.8106	1.8020	1.8059	1.8254	1.8184	1.8083	1.8106	1.8028	1.8106	1.8059	1.8114	1.8036	1.8122;
1.7911	1.7895	1.7942	1.7965	1.7934	1.7887	1.7895	1.7903	1.7918	1.7973	1.7879	1.7879	1.7926	1.7879	1.7958;
1.7770	1.7747	1.7762	1.7770	1.7707	1.7747	1.7700	1.7770	1.7793	1.7707	1.7754	1.7770	1.7723	1.7700	1.7731;
1.7536	1.7629	1.7582	1.7551	1.7559	1.7528	1.7606	1.7551	1.7606	1.7598	1.7528	1.7637	1.7590	1.7629	1.7629;
1.7465	1.7465	1.7379	1.7418	1.7457	1.7426	1.7403	1.7418	1.7364	1.7426	1.7379	1.7418	1.7457	1.7371	1.7403;
1.7356	1.7270	1.7293	1.7231	1.7239	1.7254	1.7223	1.7215	1.7364	1.7246	1.7262	1.7223	1.7301	1.7207	1.7200;
1.7043	1.7082	1.7059	1.7074	1.7059	1.7090	1.7051	1.7106	1.7121	1.7051	1.7012	1.7082	1.7020	1.7082	1.7004;
1.6863	1.6926	1.6879	1.6879	1.6856	1.6887	1.6879	1.6887	1.6879	1.6856	1.7450	1.6887	1.6879	1.6879	1.6879;
1.6731	1.6731	1.6793	1.6738	1.6770	1.6715	1.6746	1.6738	1.6731	1.6684	1.6707	1.6762	1.6653	1.6637	1.6731;
1.6629	1.6629	1.6629	1.6613	1.6606	1.6590	1.6629	1.6613	1.6574	1.6629	1.6629	1.6629	1.6645	1.6606	1.6231;
1.6457	1.6457	1.6496	1.6455	1.6434	1.6387	1.6402	1.6481	1.6457	1.6488	1.6449	1.6496	1.6559	1.6395	1.6504;
1.6356	1.6285	1.6270	1.6340	1.6340	1.6316	1.6348	1.6262	1.6356	1.6332	1.6285	1.6231	1.6270	1.6340	1.6348;
1.6105	1.6129	1.6184	1.6207	1.6129	1.5965	1.6121	1.6168	1.6137	1.6176	1.6137	1.6145	1.6105	1.6168	1.6137;
1.5980	1.5926	1.5934	1.5996	1.5934	1.5965	1.5996	1.5949	1.5980	1.5965	1.5934	1.6059	1.5965	1.5965	1.5965;
1.5832	1.5801	1.5832	1.5793	1.5801	1.5746	1.5738	1.5777	1.5785	1.5824	1.5863	1.5754	1.5934	1.5715	1.5809;
1.5644	1.5637	1.5652	1.5668	1.5652	1.5652	1.5652	1.5652	1.5629	1.5629	1.5676	1.5715	1.5644	1.5707	1.5668;
1.5527	1.5551	1.5535	1.5605	1.5574	1.5496	1.5527	1.5582	1.5527	1.5574	1.5582	1.5551	1.5558	1.5504	1.5504;
1.5379	1.5394	1.5418	1.5379	1.5379	1.5363	1.5387	1.5394	1.5418	1.5379	1.5387	1.5387	1.5379	1.5371	1.5418;
1.5269	1.5269	1.5246	1.5199	1.5238	1.5269	1.5176	1.5277	1.5246	1.5277	1.5277	1.5246	1.5285	1.5394	1.5215;
1.5137	1.5129	1.5176	1.5129	1.5121	1.5097	1.5183	1.5129	1.5137	1.5129	1.5129	1.5137	1.5152	1.5129	1.5129;
1.5105	1.5043	1.5019	1.5074	1.5058	1.5027	1.5004	1.4965	1.5011	1.5011	1.5090	1.5051	1.5058	1.5090	1.5004;
1.4902	1.4918	1.4902	1.4886	1.4902	1.4980	1.4926	1.4886	1.4871	1.4918	1.4800	1.4840	1.4886	1.4855	1.4879;
1.4847	1.4808	1.4754	1.4793	1.4707	1.4769	1.4777	1.4847	1.4840	1.4840	1.4754	1.4824	1.4808	1.4769	1.4793;
1.4636	1.4707	1.4683	1.4715	1.4660	1.4675	1.4699	1.4261	1.4629	1.4707	1.4629	1.4660	1.4675	1.4707	1.4715;
1.4589	1.4621	1.4629	1.4543	1.4629	1.4574	1.4589	1.4582	1.4652	1.4605	1.4550	1.4589	1.4629	1.4574	1.4629;
1.4550	1.4496	1.4472	1.4550	1.4496	1.4480	1.4488	1.4543	1.4472	1.4480	1.4480	1.4511	1.4527	1.4504	1.4519;
1.3878	1.4379	1.4355	1.4425	1.4418	1.4410	1.4480	1.4433	1.4394	1.4394	1.4386	1.4363	1.4386	1.4371	1.4386;
1.4308	1.4347	1.4363	1.4339	1.4246	1.4324	1.4363	1.4339	1.4308	1.4394	1.4308	1.4347	1.4339	1.4277	1.4300;
1.4246	1.4207	1.4246	1.4207	1.4246	1.4230	1.4230	1.4285	1.4230	1.4308	1.4269	1.4207	1.4238	1.4207	1.4285;
1.4199	1.4121	1.4128	1.4136	1.4152	1.4105	1.4136	1.4136	1.4136	1.4136	1.4160	1.4128	1.4121	1.4128	1.4144;
1.4128	1.4050	1.3988	1.4003	1.3902	1.4050	1.3933	1.4035	1.4097	1.4074	1.4003	1.4003	1.4058	1.4042	1.4003;
1.3871	1.3878	1.3863	1.3839	1.3847	1.3878	1.3886	1.3855	1.3878	1.3902	1.3871	1.3871	1.3871	1.3839	1.3988;
1.3667	1.3628	1.3667	1.3628	1.3628	1.3628	1.3675	1.3628	1.3644	1.3667	1.3628	1.3683	1.3652	1.3652	1.3699;
1.3722	1.3699	1.3738	1.3699	1.3667	1.3714	1.3683	1.3683	1.3691	1.3644	1.3402	1.3683	1.3699	1.3706	1.3925;
1.3753	1.3753	1.3675	1.3980	1.3706	1.3683	1.3753	1.3386	1.3284	1.3746	1.3964	1.3769	1.3777	1.3761	1.3777;
1.3769	1.3753	1.3581	1.3746	1.3761	1.4050	1.3667	1.3761	1.3206	1.3761	1.3683	1.3746	1.3675	1.3738	1.3683]

PROGRAMA EN JULIA PARA ANALISIS DE DATOS Y MUESTREO DE LA SEÑAL EN UNA GRÁFICA

```

1 using Plots
2 using Statistics
3
4 function graf()
5 #Eje X son las frecuencias y Eje Y son los datos digitales del circuito(0-4095) escalado a la tension maxima que puede sacar la
  ESP32 (0-3.3)
6 Ejex=
  [17500,35000,52500,70000,87500,105000,122500,140000,157500,175000,192500,210000,227500,245000,262500,280000,297500,315000,332500,
  350000,367500,385000,402500,420000,437500,455000,472500,490000,507500,525000,542500,560000,577500,595000,612500,630000,647500,
  665000,682500,700000,717500,735000,752500,770000,787500,805000,822500,840000,857500,875000,892500,910000,927500,945000,962500,
  980000,997500,1015000,1032500,1050000,1067500,1085000,1102500,1120000,1137500,1155000,1172500,1190000,1207500,1225000,1242500,
  1260000,1277500,1295000,1312500,1330000,1347500,1365000,1382500,1400000,1417500,1435000,1452500,1470000,1487500,1505000,1522500,
  1540000,1557500,1575000,1592500,1610000,1627500,1645000,1662500,1680000,1697500,1715000,1732500,1750000,1767500,1785000,1802500,
  1820000,1837500,1855000,1872500,1890000,1907500,1925000,1942500,1960000,1977500,1995000,2012500,2030000,2047500,2065000,2082500,
  2100000,2117500,2135000,2152500,2170000,2187500,2205000,2222500,2240000,2257500,2275000,2292500,2310000,2327500,2345000,2362500,
  2380000,2397500,2415000,2432500,2450000,2467500,2485000,2502500,2520000,2537500,2555000,2572500,2590000,2607500,2625000,2642500,
  2660000,2677500,2695000,2712500,2730000,2747500,2765000,2782500,2800000,2817500,2835000,2852500,2870000,2887500,2905000,2922500,
  2940000,2957500,2975000,2992500,3010000,3027500,3045000,3062500,3080000,3097500,3115000,3132500,3150000,3167500,3185000,3202500,
  3220000,3237500,3255000,3272500,3290000,3307500,3325000,3342500,3360000,3377500,3395000,3412500,3430000,3447500,3465000,3482500,
  3500000]
7
8 #EJEMPLO DE MATRIZ QUE DEFINE CIERTA ACTIVIDAD
9
10 a=[2.5491 2.5326 2.4428 2.5342 2.5186 2.5358 2.5217 2.5366 2.5272 2.5381 2.5139 2.5358 2.5280 2.5342 2.5147,
11 2.5366 2.5389 2.5436 2.5658 2.5436 2.5467 2.5420 2.5420 2.5389 2.5436 2.5405 2.5397 2.5452 2.5436 2.5428 2.5483 2.5506 2.5342,
12 2.5397 2.5467 2.5483 2.5366 2.5389 2.5389 2.5397 2.5405 2.5366 2.5420 2.5366 2.5420 2.5373 2.5506 2.5342,
13 2.5436 2.5381 2.5467 2.5405 2.5452 2.5358 2.5420 2.5498 2.5459 2.5489 2.5389 2.5459 2.5389 2.4775 2.5498,
14 2.5428 2.5514 2.5381 2.5397 2.5381 2.5381 2.5420 2.5459 2.5428 2.5498 2.5205 2.5334 2.5467 2.5381 2.5381,
15 2.5397 2.5373 2.5373 2.5373 2.5366 2.5452 2.5467 2.5584 2.5428 2.5397 2.5452 2.5483 2.5397 2.5350 2.5483 2.5381,
16 2.5420 2.5452 2.5498 2.5389 2.5514 2.5397 2.5491 2.5483 2.5381 2.5459 2.5420 2.5436 2.5389 2.5498 2.5389,
17 2.5452 2.5436 2.5420 2.5811 2.5483 2.5358 2.5381 2.5467 2.5420 2.5436 2.5397 2.5491 2.5420 2.5405 2.5452,
18 2.5373 2.5491 2.5381 2.5405 2.5545 2.5491 2.5405 2.5514 2.5389 2.5467 2.5452 2.5381 2.5514 2.5452 2.5498,
19 2.5412 2.5452 2.5467 2.5405 2.5536 2.5397 2.5436 2.5483 2.5561 2.5498 2.5452 2.5397 2.5459 2.5459 2.5397,
20 2.5405 2.5389 2.5405 2.5397 2.5412 2.5420 2.5436 2.5381 2.5397 2.5381 2.5467 2.5483 2.5467 2.5397 2.5467 2.5498,
21 2.5428 2.5405 2.5412 2.5397 2.5545 2.5483 2.5452 2.5373 2.5467 2.5498 2.5536 2.5436 2.5428 2.5514 2.5342,
22 2.5366 2.5407 2.5381 2.5483 2.5514 2.5608 2.5506 2.5459 2.5459 2.5608 2.5428 2.5639 2.5483 2.5483 2.5412,
23 2.5491 2.5389 2.5467 2.5405 2.5405 2.5514 2.5498 2.5467 2.5514 2.5381 2.5389 2.5467 2.5522 2.5436 2.5428,
24 2.5483 2.5436 2.5506 2.5483 2.5436 2.5498 2.5428 2.5459 2.5428 2.5483 2.5428 2.5506 2.5475 2.5350 2.5381,
25 2.5498 2.5420 2.5491 2.5467 2.5452 2.5358 2.5405 2.5389 2.5436 2.5475 2.5420 2.5381 2.5405 2.5428 2.5498,
26 2.5491 2.5381 2.5483 2.5381 2.5921 2.5436 2.5452 2.5420 2.5428 2.5389 2.5405 2.5467 2.5389 2.5436 2.5412,
27 2.5405 2.5506 2.5381 2.5381 2.5405 2.5405 2.5483 2.5358 2.5452 2.5405 2.5420 2.5420 2.5420 2.5412,
28 2.5405 2.5389 2.5405 2.5397 2.5412 2.5420 2.5436 2.5381 2.5397 2.5381 2.5467 2.5483 2.5467 2.5397 2.5467 2.5498,
29 2.5428 2.5428 2.5420 2.5381 2.5420 2.5381 2.5405 2.5381 2.5342 2.5342 2.5467 2.5444 2.5381 2.5405 2.5436,
30 2.5342 2.5420 2.5452 2.5428 2.5436 2.5389 2.5491 2.5342 2.5545 2.5420 2.5475 2.5381 2.5491 2.5389 2.5545,
31 2.5428 2.5405 2.5381 2.5373 2.5373 2.5405 2.5459 2.5467 2.5381 2.5436 2.5381 2.5452 2.5405 2.5381 2.5420,
32 2.5381 2.5326 2.4733 2.5436 2.5436 2.5405 2.5381 2.5436 2.5389 2.5420 2.5467 2.5381 2.5381 2.5303 2.5381,
33 2.5334 2.5366 2.5381 2.5428 2.5342 2.5405 2.5459 2.5334 2.5405 2.5358 2.5381 2.5405 2.5389 2.5389 2.5405,
34 2.5389 2.5397 2.5287 2.5381 2.5676 2.5420 2.5373 2.5436 2.5397 2.5334 2.5381 2.5264 2.5350 2.5428 2.5420,
35 2.5436 2.5428 2.536 2.5366 2.5366 2.5342 2.5389 2.5381 2.5381 2.5342 2.5397 2.5358 2.5350 2.5405 2.5381 2.5342,
36 2.5381 2.5436 2.5366 2.5366 2.5366 2.5366 2.5389 2.5381 2.5381 2.5342 2.5397 2.5358 2.5350 2.5405 2.5381 2.5342,
37 2.5366 2.5412 2.5342 2.5381 2.5311 2.5326 2.5373 2.5389 2.5381 2.5389 2.5311 2.5381 2.5358 2.5307 2.5358,
38 2.5389 2.5272 2.5311 2.5358 2.5342 2.5381 2.5366 2.5373 2.5366 2.5381 2.5373 2.5280 2.5358 2.5381 2.5312 2.5373,
39 2.5350 2.5389 2.5311 2.5381 2.5389 2.5342 2.5428 2.5303 2.5366 2.5264 2.5358 2.5358 2.5373 2.5311 2.5311,
40 2.5358 2.5272 2.5280 2.5248 2.5264 2.5358 2.5303 2.5303 2.5358 2.5342 2.5342 2.5358 2.5350 2.5342 2.5217,
41 2.5326 2.5436 2.5311 2.5303 2.5155 2.5358 2.5311 2.5326 2.5295 2.5326 2.5342 2.5280 2.5295 2.5342 2.5280,
42 2.5342 2.5311 2.5311 2.5287 2.5389 2.5264 2.5280 2.5256 2.5272 2.5295 2.5201 2.5303 2.5326 2.5303 2.5412,
43 2.5311 2.5334 2.5311 2.5293 2.5280 2.5280 2.5283 2.5326 2.5350 2.5395 2.5248 2.5280 2.5326 2.5326 2.5326,
44 2.5283 2.5248 2.5186 2.5326 2.5248 2.5248 2.5256 2.5217 2.5209 2.5272 2.5303 2.5287 2.5209 2.5280 2.5366,
45 2.5248 2.5303 2.5350 2.5233 2.5256 2.5264 2.5256 2.5389 2.5295 2.5631 2.5514 2.5248 2.5287 2.5303 2.5248,
46 2.5209 2.5217 2.5178 2.5155 2.5194 2.5217 2.5280 2.5389 2.5280 2.5233 2.5264 2.5248 2.5303 2.5342 2.5256,
47 2.5186 2.5280 2.5295 2.5334 2.5256 2.5217 2.5217 2.5162 2.5248 2.5241 2.5264 2.5264 2.5553 2.5241 2.5155 2.5272,
48 2.5178 2.5209 2.5131 2.5162 2.5186 2.5241 2.5264 2.5241 2.5131 2.5264 2.5170 2.5256 2.5147 2.5209 2.5272,
49 2.5241 2.5147 2.5178 2.5170 2.5217 2.5280 2.5319 2.5248 2.5334 2.5241 2.5248 2.5170 2.5131 2.5162 2.5201,
50 2.5225 2.5155 2.5209 2.5201 2.5220 2.5170 2.5155 2.5123 2.5139 2.5186 2.5381 2.5131 2.5170 2.5201 2.5162,
51 2.5199 2.5147 2.5186 2.5194 2.5162 2.5280 2.5170 2.5162 2.5201 2.5147 2.5190 2.5170 2.5131 2.5194 2.5147,
52 2.5139 2.5116 2.5084 2.5108 2.5498 2.5147 2.5131 2.5241 2.5430 2.5147 2.5061 2.5186 2.5131 2.5131 2.5014,
53 2.5116 2.5108 2.5155 2.5139 2.4936 2.5100 2.5053 2.5131 2.5123 2.5123 2.5030 2.5436 2.5123 2.5084 2.5170,
54 2.5092 2.5045 2.5061 2.5084 2.5053 2.5092 2.5053 2.5061 2.4905 2.5053 2.5108 2.5123 2.5053 2.5076 2.5045,
55 2.4983 2.5061 2.5006 2.5061 2.5061 2.4951 2.5108 2.5022 2.5022 2.4967 2.5053 2.5178 2.5061 2.4908 2.5053,
56 2.5076 2.4975 2.5030 2.5022 2.5030 2.4997 2.5030 2.4998 2.5053 2.5014 2.4951 2.5006 2.5014 2.5006 2.4928,
57 2.4967 2.5241 2.4983 2.4983 2.4920 2.4967 2.4905 2.5061 2.5030 2.4983 2.4162 2.4936 2.4999 2.5014 2.4912,
58 2.4920 2.4881 2.4899 2.4983 2.4883 2.4899 2.4985 2.4957 2.4959 2.4912 2.4951 2.4881 2.4920 2.4881 2.5006,
59 2.4989 2.4897 2.4811 2.4928 2.4881 2.4881 2.4881 2.4881 2.4881 2.4881 2.4881 2.4881 2.4881 2.4881 2.4881 2.4881,
60 2.4240 2.4905 2.4881 2.4873 2.4881 2.4881 2.4881 2.4881 2.4881 2.4881 2.4881 2.4881 2.4881 2.4881 2.4881,
61 2.4865 2.4873 2.4881 2.4842 2.4826 2.4905 2.4154 2.4928 2.4740 2.4865 2.4381 2.4881 2.4873 2.4873 2.4881,
62 2.4131 2.4873 2.4795 2.4834 2.4826 2.4779 2.4881 2.4873 2.4865 2.4764 2.4826 2.4811 2.4967 2.4858 2.4826,
63 2.4764 2.4139 2.4772 2.4772 2.4725 2.4772 2.4756 2.4787 2.4865 2.4787 2.4686 2.4779 2.4826 2.4903 2.4779,
64 2.5241 2.4639 2.4795 2.4842 2.4858 2.4764 2.4795 2.4779 2.4803 2.4779 2.4858 2.4764 2.4740 2.4725 2.4748,
65 2.4131 2.4834 2.4725 2.4747 2.4748 2.4701 2.4678 2.4733 2.4725 2.4686 2.4717 2.4748 2.4756 2.4631 2.4748,
66 2.4170 2.4639 2.4670 2.4670 2.4686 2.4717 2.4740 2.4657 2.4701 2.4761 2.4686 2.4631 2.4725 2.4686 2.4779,
67 2.4576 2.4647 2.4639 2.4584 2.4631 2.4631 2.4631 2.4631 2.4623 2.4631 2.4576 2.4631 2.4639 2.4688 2.4545 2.4576,
68 2.4583 2.4545 2.4686 2.4561 2.4529 2.4553 2.4600 2.4529 2.4561 2.4631 2.4529 2.4608 2.4576 2.4631 2.4584,
69 2.4506 2.4490 2.4498 2.4475 2.4506 2.4498 2.4506 2.4459 2.4561 2.4522 2.4529 2.4529 2.4561 2.4522 2.4529,
70 2.4404 2.4483 2.4483 2.4420 2.4467 2.4483 2.4459 2.4459 2.4498 2.4483 2.4451 2.4490 2.4506 2.4404 2.4420,
71 2.4381 2.4436 2.4342 2.4381 2.4381 2.4436 2.4412 2.4311 2.4381 2.4373 2.4318 2.4381 2.4428 2.4404 2.4397,
72 2.4303 2.4311 2.4365 2.4381 2.4217 2.4279 2.4358 2.4256 2.4295 2.4279 2.4264 2.4217 2.4279 2.4279 2.4295,
73 2.4217 2.4186 2.4217 2.4295 2.4295 2.4217 2.4248 2.4240 2.4350 2.4272 2.4186 2.4256 2.4256 2.4217 2.4248,
74 2.3982 2.4178 2.4178 2.4123 2.4154 2.4170 2.4170 2.4170 2.4170 2.4170 2.4147 2.4217 2.4248 2.4139 2.4115 2.4147,
75 2.4115 2.3998 2.3936 2.4100 2.4115 2.3998 2.3998 2.4154 2.4107 2.4092 2.4131 2.3943 2.4115 2.3998 2.3967,
76 2.3904 2.3881 2.3950 2.4045 2.3936 2.4107 2.3826 2.3881 2.3928 2.3896 2.3881 2.3951 2.3928 2.4123 2.3982,
77 2.3779 2.3764 2.3771 2.3842 2.3826 2.3779 2.3756 2.3811 2.3748 2.4389 2.3795 2.3842 2.3748 2.3756 2.3865,
78 2.3623 2.3287 2.3646 2.3631 2.3646 2.3678 2.3639 2.3740 2.3717 2.3631 2.3748 2.3685 2.3764 2.3685 2.3631,
79 2.3467 2.3584 2.3607 2.3529 2.3490 2.3576 2.3529 2.3482 2.3482 2.3498 2.3514 2.3498 2.3553 2.3584 2.3514,
80 2.3381 2.3435 2.3396 2.3381 2.3381 2.3396 2.3435 2.3381 2.3396 2.3381 2.3404 2.3412 2.3381 2.3428 2.3420,
81 2.3240 2.3178 2.3263 2.3248 2.3170 2.3185 2.3474 2.2888 2.3178 2.3263 2.3256 2.3217 2.3154 2.3263 2.3256,
82 2.2998 2.3092 2.3045 2.3107 2.2998 2.3021 2.3090 2.3107 2.3090 2.3107 2.3090 2.3084 2.3084 2.3084 2.3084 2.3084,
83 2.2904 2.2896 2.2904 2.3670 2.2881 2.2787 2.2412 2.2896 2.2795 2.2881 2.2873 2.2216 2.2881 2.2896 2.2873,
84 2.2677 2.2631 2.2646 2.2623 2.2631 2.2756 2.2638 2.2685 2.2631 2.2646 2.2638 2.2670 2.2607 2.2724 2.2638,
85 2.2490 2.2498 2.2388 2.2380 2.2466 2.2341 2.2427 2.2459 2.2482 2.2380 2.2420 2.1826 2.2404 2.2380 2.2373,
86 2.2169 2.2130 2.2130 2.2185 2.2154 2.2185 2.2138 2.2130 2.2169 2.2162 2.2154 2.2154 2.2138 2.2154,
87 2.1810 2.1810 2.1880 2.1857 2.1857 2.1826 2.1888 2.1810 2.1857 2.1904 2.1826 2.1857 2.1833 2.1841 2.1873,
88 2.1505 2.1513 2.1513 2.1451 2.1505 2.1466 2.1521 2.1537 2.1490 2.1451 2.1497 2.1451 2.1576 2.1497 2.1615,
89 2.1130 2.1075 2.1044 2.1107 2.1075 2.1107 2.1115 2.1154 2.1107 2.1075 2.1060 2.1083 2.1115 2.1115 2.1115,
90 2.0646 2.0646 2.0622 2.0630 2.0614 2.0630 2.0614 2.0630 2.0614 2.0630 2.0614 2.0630 2.0614 2.0630 2.0614 2.0630,
91 2.0130 2.0091 2.9997 2.0130 2.0096 2.0114 2.0106 2.0083 2.0114 2.0114 2.0075 2.0122 2.0130 2.9997 2.0114,
92 1.9356 1.9388 1.9389 1.9403 1.9431 1.9395 1.9380 1.9403 1.9372 1.9356 1.9356 1.9388 1.9372 1.9380 1.9372,
93 1.8583 1.8653 1.8614 1.8598 1.8575 1.8614 1.8630 1.8604 1.8637 1.8630 1.8630 1.8630 1.8614 1.8606 1.8622,
94 1.7653 1.7692 1.7661 1.7684 1.7598 1.7637 1.7629 1.7715 1.7653 1.7668 1.7645 1.7715 1.7715 1.7621 1.7653,
95 1.6582 1.6559 1.6504 1.6527 1.6567 1.6590 1.6512 1.6504 1.6504 1.6504 1.6520 1.6504 1.6527 1.6606 1.6551,
96 1.6340 1.6160 2.6240 2.6230 1.6183 1.6168 1.6199 1.6183 1.6308 1.6246 1.6277 1.6230 1.6238 1.6162 1.6246,
97 1.4919 1.4940 1.3777 1.3917 1.3650 1.3690 1.3693 1.3777 1.3753 1.3769 1.3769 1.3769 1.3693 1.3693 1.3777,
98 2.464 2.128 2.123 2.123 2.123 2.123 2.123 2.123 2.123 2.123 2.123 2.123 2.123 2.123 2.123 2.123 2.123 2.123 2.123,
99 0.8878 0.8292 0.8182 0.8182 0.8174 0.8135 0.8237 0.8096 0.8198 0.8167 0.8229 0.8167 0.8182 0.8182 0.8120,
100 0.8119 0.8307 0.7971 0.7994 0.7986 0.7994 0.8080 0.7963 0.7947 0.7993 0.7916 0.7877 0.7986 0.7986 0.8080,

```

101	0.7127	0.6330	0.5994	0.5712	0.5634	0.5681	0.5744	0.5744	0.5744	0.5626	0.5962	0.5830	0.5759	0.5790	0.5626;
102	0.5079	0.4470	0.3985	0.3767	0.3845	0.3806	0.3837	0.3876	0.3923	0.3853	0.3798	0.3806	0.3829	0.3868	0.3821;
103	0.3384	0.3118	0.3016	0.3016	0.2962	0.2993	0.2962	0.2899	0.2977	0.3024	0.2633	0.2977	0.3001	0.2962	0.2993;
104	0.3485	0.3532	0.3602	0.3626	0.3602	0.3626	0.3626	0.3626	0.3571	0.3579	0.3501	0.3556	0.3634	0.3602	0.3618;
105	0.5329	0.5400	0.5408	0.5448	0.5415	0.5353	0.5400	0.5376	0.5353	0.5376	0.4876	0.5431	0.5353	0.5353	0.5392;
106	0.7517	0.7666	0.7627	0.7596	0.7650	0.7619	0.7619	0.7603	0.7642	0.7603	0.7572	0.7619	0.7627	0.7619	0.7635;
107	0.9877	0.9963	0.9970	0.9948	0.9932	0.9877	0.9893	0.9916	0.9901	0.9948	0.9901	0.9901	0.9987	0.9956	0.9885;
108	1.1964	1.1987	1.1987	1.1917	1.1964	1.2112	1.1995	1.2058	1.1972	1.1979	1.1917	1.1886	1.1987	1.2089	1.1979;
109	1.3761	1.4660	1.4214	1.3839	1.3871	1.3769	1.3863	1.3714	1.3730	1.3792	1.3871	1.3785	1.3730	1.3808	1.3855;
110	1.5340	1.5238	1.5324	1.5293	1.5285	1.5254	1.5246	1.5230	1.5293	1.5262	1.5277	1.5254	1.5269	1.5246	1.5301;
111	1.6543	1.6527	1.6551	1.6543	1.6559	1.6582	1.6559	1.6559	1.6590	1.6551	1.6574	1.6590	1.6567	1.6574	1.6527;
112	1.7629	1.7629	1.7559	1.7629	1.7629	1.7621	1.7629	1.7536	1.7629	1.7645	1.7598	1.7582	1.7606	1.7590	1.7621;
113	1.8505	1.8465	1.8505	1.8505	1.8481	1.8497	1.8489	1.8465	1.8505	1.8458	1.8458	1.8505	1.8520	1.8520	1.8497;
114	1.9208	1.9247	1.9223	1.9255	1.9278	1.9247	1.9184	1.9372	1.9286	1.9239	1.9255	1.9208	1.9231	1.9263	1.9169;
115	1.9966	1.9903	1.9927	1.9880	1.9919	1.9880	1.9864	1.9927	1.9888	1.9880	1.9958	1.9880	1.9927	1.9880	1.9895;
116	2.0513	2.0396	2.0482	2.0482	2.0505	2.0513	2.0505	2.0513	2.0505	2.0489	2.0505	2.0482	2.0505	2.0505	2.0482;
117	2.0935	2.0904	2.0904	2.0919	2.0909	2.0904	2.0958	2.1005	2.0999	2.0958	2.0958	2.1005	2.1294	2.0997	2.0935;
118	2.1484	2.1380	2.1419	2.1372	2.1341	2.1380	2.1419	2.1513	2.1372	2.1388	2.1419	2.1404	2.1388	2.1484	2.1263;
119	2.1763	2.1716	2.1771	2.1788	2.1755	2.1771	2.1794	2.1763	2.1716	2.1740	2.1771	2.1654	2.1802	2.1771	2.1740;
120	2.2076	2.2084	2.2044	2.2044	2.2060	2.2060	2.2115	2.2060	2.2091	2.2084	2.2021	2.1841	2.2044	2.2005	2.2107;
121	2.2357	2.2357	2.2310	2.2380	2.2365	2.2373	2.2373	2.2334	2.2310	2.2326	2.2373	2.2341	2.2295	2.2341	2.2310;
122	2.2584	2.2584	2.2537	2.2607	2.2591	2.2521	2.2560	2.2584	2.2505	2.2607	2.2701	2.2654	2.2529	2.2607	2.2607;
123	2.2761	2.2795	2.2732	2.2763	2.2716	2.2763	2.2779	2.2795	2.2774	2.2716	2.2779	2.2779	2.2779	2.2779	2.2756;
124	2.2904	2.2935	2.2935	2.2912	2.2920	2.2881	2.2951	2.2982	2.2896	2.2896	2.2904	2.3774	2.2912	2.2842	2.2920;
125	2.3053	2.3060	2.3021	2.3013	2.3099	2.2974	2.3123	2.3053	2.3084	2.3006	2.3045	2.3107	2.2998	2.3021	2.3021;
126	2.3162	2.3198	2.3131	2.3131	2.3131	2.3201	2.3178	2.3162	2.2991	2.3131	2.3131	2.3185	2.3240	2.3170	2.3131;
127	2.3170	2.3263	2.3248	2.3185	2.3217	2.3201	2.3232	2.3404	2.3185	2.3185	2.3256	2.3232	2.3256	2.3295	2.3256;
128	2.3357	2.3334	2.3420	2.3404	2.3310	2.3349	2.3506	2.3373	2.3365	2.3279	2.3029	2.3389	2.3310	2.3357	2.3209;
129	2.3326	2.3404	2.3396	2.3365	2.3435	2.3381	2.3387	2.3342	2.3381	2.3428	2.3357	2.3381	2.3381	2.3389	2.3381;
130	2.3459	2.3467	2.3435	2.3428	2.3381	2.3428	2.3459	2.3381	2.3428	2.3482	2.3389	2.3435	2.3389	2.3404	2.3545;
131	2.3435	2.3435	2.3459	2.3498	2.3451	2.3404	2.3459	2.3428	2.3435	2.3404	2.3467	2.3467	2.3467	2.3381	2.3404;
132	2.3498	2.3529	2.3498	2.3498	2.3467	2.3521	2.3553	2.3467	2.3451	2.3396	2.3404	2.3459	2.3498	2.3428	2.3506;
133	2.3435	2.3467	2.3490	2.3467	2.3514	2.3428	2.3482	2.3365	2.3435	2.3498	2.3459	2.3623	2.3451	2.3482	2.3435;
134	2.3474	2.3506	2.3459	2.3474	2.3498	2.3506	2.3498	2.3467	2.3381	2.3412	2.3521	2.3498	2.3459	2.3506	2.4709;
135	2.3498	2.3537	2.3490	2.3459	2.3482	2.3389	2.3396	2.3490	2.3420	2.3490	2.3451	2.4725	2.3451	2.3428	2.3498;
136	2.3420	2.3381	2.3381	2.3451	2.3412	2.3404	2.3435	2.2763	2.3381	2.3389	2.3381	2.3428	2.3342	2.3381	2.3584;
137	2.3381	2.3349	2.3631	2.3373	2.3685	2.3357	2.3381	2.3381	2.3365	2.3607	2.3381	2.3326	2.3381	2.3420	2.3381;
138	2.3381	2.3256	2.3263	2.3381	2.3373	2.3342	2.3217	2.3256	2.3248	2.3263	2.3271	2.3381	2.3396	2.3326	2.3373;
139	2.3295	2.3185	2.3310	2.3310	2.3232	2.3178	2.3185	2.3248	2.3256	2.3248	2.3248	2.3146	2.3185	2.3271	2.3271;
140	2.3178	2.3154	2.3146	2.3146	2.3115	2.3170	2.3154	2.3138	2.3170	2.3146	2.2842	2.3138	2.3123	2.3131	2.3138;
141	2.3084	2.3123	2.3107	2.3045	2.3060	2.3099	2.3131	2.3107	2.3053	2.3013	2.3107	2.3029	2.3107	2.3053	2.3013;
142	2.2990	2.2912	2.3021	2.2951	2.2904	2.2959	2.2959	2.2935	2.3006	2.2959	2.2982	2.2951	2.3021	2.2927	2.2959;
143	2.2888	2.2802	2.2920	2.2810	2.2881	2.2888	2.2779	2.2873	2.2873	2.2834	2.2857	2.2881	2.2865	2.2810	2.2904;
144	2.2748	2.2748	2.2748	2.2756	2.2748	2.2716	2.2740	2.2732	2.2763	2.2756	2.2709	2.2748	2.2724	2.2763	2.2748;
145	2.2654	2.2545	2.2591	2.2623	2.2576	2.2631	2.2631	2.2646	2.2591	2.2638	2.2631	2.2631	2.2623	2.2591	2.2607;
146	2.2513	2.2498	2.2513	2.2529	2.2482	2.2591	2.2412	2.2435	2.2505	2.2513	2.2498	2.2466	2.2334	2.2466	2.2459;
147	2.2341	2.2357	2.2341	2.2466	2.2380	2.2365	2.2396	2.2334	2.2380	2.2357	2.2341	2.2373	2.2357	2.2388	2.2341;
148	2.2380	2.2255	2.2209	2.2201	2.2380	2.2201	2.2201	2.2279	2.2255	2.2216	2.2216	2.2209	2.2248	2.2162	2.2248;
149	2.2107	2.2099	2.2044	2.2099	2.2052	2.2130	2.2107	2.2044	2.2060	2.2091	2.2060	2.2130	2.2084	2.2060	2.2060;
150	2.2005	2.1826	2.1966	2.1951	2.1919	2.1951	2.1984	2.1927	2.1935	2.1998	2.1998	2.2248	2.1998	2.1888	2.1935;
151	2.1833	2.1873	2.1810	2.1810	2.1763	2.1802	2.1794	2.1857	2.1794	2.1787	2.1826	2.2365	2.1818	2.1880	2.1774;
152	2.1687	2.1630	2.1638	2.1630	2.1630	2.1622	2.1654	2.1622	2.1646	2.1482	2.1638	2.1747	2.1646	2.1654	2.1622;
153	2.1474	2.1529	2.1466	2.1521	2.1497	2.1529	2.1451	2.1482	2.1482	2.1497	2.1513	2.1497	2.1490	2.1583	2.1521;
154	2.1372	2.1365	2.1271	2.1341	2.1333	2.1380	2.1247	2.1380	2.1310	2.1380	2.1372	2.1310	2.1326	2.1333	2.1458;
155	2.1185	2.1216	2.1138	2.1224	2.1200	2.1185	2.1130	2.1185	2.1200	2.1177	2.1138	2.1177	2.1310	2.1130	2.1161;
156	2.1021	2.0974	2.1005	2.0950	2.1052	2.0997	2.1044	2.1005	2.1013	2.0989	2.1036	2.0966	2.1005	2.0974	2.1029;
157	2.0857	2.0825	2.0857	2.0904	2.0880	2.0864	2.0880	2.0810	2.0810	2.0864	2.0849	2.0872	2.0904	2.0888	2.0708;
158	2.0763	2.0716	2.0716	2.0661	2.0716	2.0716	2.0700	2.0716	2.0659	2.0677	2.0716	2.0544	2.0732	2.0716	2.0677;
159	2.0560	2.0599	2.0497	2.0528	2.0513	2.0544	2.0552	2.0544	2.0560	2.0505	1.9856	2.0528	2.0560	2.0528	2.0552;
160	2.0372	2.0357	2.0396	2.0396	2.0419	2.0364	2.0403	2.0380	2.0380	1.9927	2.0419	2.0450	2.1294	2.0419	2.0388;
161	2.0232	2.0232	1.9544	2.0263	2.0224	2.0216	2.0161	2.0247	2.0177	2.0341	2.0239	1.9997	2.0239	2.0177	2.0216;
162	2.0075	2.0106	2.0091	1.9966	2.0057	1.9997	2.0075	1.9919	1.9966	1.9997	2.0060	2.0106	1.9997	2.0083	1.9958;
163	1.9849	1.9872	1.9802	1.9817	1.9864	1.9888	1.9841	1.9856	1.9864	1.9841	1.9864	1.9833	1.9840	1.9888	1.9825;
164	1.9638	1.9653	1.9630	1.9653	1.9661	1.9739	1.9747	1.9653	1.9677	1.9669	1.9661	1.9684	1.9661	1.9645	1.9645;
165	1.9489	1.9520	1.9474	1.9528	1.9505	2.0169	1.9450	1.9583	1.9630	1.9497	1.9591	1.9380	1.9536	1.9497	1.9489;
166	1.9380	1.9341	1.9380	1.9177	1.9341	1.9012	1.9364	1.9380	1.9364	1.9341	1.9372	1.9325	1.9278	1.9380	1.9356;
167	1.9231	1.9200	1.9184	1.9239	1.9184	1.9184	1.9169	1.9231	1.9208	1.9208	1.9161	1.9200	1.9130	1.9153	1.9130;
168	1.8997	1.9005	1.9020	1.9012	1.9005	1.9020	1.9044	1.9028	1.9044	1.8997	1.9005	1.8966	1.9091	1.8997	1.8989;
169	1.8872	1.8880	1.8880	1.8872	1.8841	1.8880	1.8856	1.8880	1.8809	1.8981	1.8887	1.8895	1.8872	1.8872	

```

201 1.4168 1.4207 1.4238 1.4199 1.4214 1.4168 1.4183 1.4183 1.4246 1.4168 1.4207 1.4160 1.4128 1.4246 1.4199;
202 1.4214 1.4136 1.4128 1.4128 1.4214 1.4152 1.4113 1.4128 1.4152 1.4136 1.4199 1.4128 1.4136 1.4050 1.4128;
203 1.4105 1.4121 1.4089 1.4105 1.4246 1.4050 1.4089 1.4074 1.4105 1.4050 1.4121 1.4128 1.4121 1.4082 1.4121;
204 1.4019 1.4019 1.4050 1.4003 1.4058 1.4003 1.4042 1.4121 1.4019 1.4058 1.4019 1.4050 1.4027 1.4074 1.4042;
205 1.4003 1.3996 1.3964 1.3949 1.3871 1.3996 1.3957 1.3964 1.3957 1.4011 1.4011 1.4074 1.3964 1.3933 1.3957;
206 1.3878 1.3925 1.3902 1.3933 1.3917 1.3878 1.3925 1.3863 1.3933 1.3839 1.3878 1.3878 1.3847 1.3910 1.3878;
207 1.3894 1.3878 1.3769 1.3847 1.3878 1.3855 1.3824 1.3878 1.3894 1.3800 1.3832 1.3839 1.3777 1.3878 1.3863;
208 1.3800 1.3753 1.3808 1.3792 1.3800 1.3816 1.3839 1.3722 1.3761 1.3777 1.3730 1.3808 1.3777 1.3808 1.3824;
209 1.3777 1.3714 1.3675 1.3714 1.3714 1.3706 1.3753 1.3714 1.3761 1.3706 1.3706 1.3714 1.3738 1.3777 1.3683]
210 «Si tenemos más actividades vamos pegando del programa en C a este las matrices con distintas letras para cada actividad
    (a,b,c...)
211
212
213 «Calculamos la media de cada vector para cada frecuencia
214 Ma=mean(a, dims=2)
215
216 «Si queremos poner más matrices para definir más actividades, únicamente ponemos lo mismo que antes cambiando la letra de la
    matriz que sea
217
218 papro=plot(Ejex, Ma, label="Ejemplo de programa en Julia") «Esta línea pondrá la gráfica con la señal de la actividad
    correspondiente con el eje x de 0 a 3,5MHz
219
220 xticks!(0:500_000:3_500_000) «Definimos los valores que queremos en el eje x y eje y
221 yticks!(0:0.2:3)
222
223 «Si quisieramos poner dos señales en la misma gráfica, únicamente habría que descomentar la siguiente línea:
224 «pbpro=plot!(Ejex, Mb, label="Ejemplo de segunda señal en Julia") «Es lo mismo pero en el plot se coloca una "!"
225 «Si quisieramos más señales pues vamos poniendo lo mismo con sus respectivas letras
226
227 display(papro) «Este comando se usa para sacar la gráfica por pantalla
228
229 end
230 graf()

```



CODIGO PARA IDENTIFICAR LOS GESTOS/ACTIVIDADES EN UN ENTORNO DEFINIDO PREVIAMENTE (BOLA, SILLA, CAMA). EL BARRIDO SE REALIZA UNICAMENTE EN LAS FRANJAS DE FRECUENCIA DONDE SABEMOS QUE EXISTE UN GESTO/ACTIVIDAD

```

1 #include <AD9833.h> // Include the library
2 #include <SPI.h>
3 #define FNC_PIN 4 // Can be any digital IO pin
4
5 const int pinEntradaAnalogica = 34;
6 int ValorFrecuencia = 17500;
7 int Frecuencia1bola = 735000, Frecuencia1silla = 1015000, Frecuencia1cama = 717500;
8 int Frecuencia2bola = 1225000, Frecuencia2silla = 1487500, Frecuencia2cama = 805000;
9 int Frecuencia3bola = 1505000, Frecuencia3silla = 1890000, Frecuencia3cama = 997500;
10 int Frecuencia4bola = 2362500;
11 int FrecuenciaGlobalbola = Frecuencia1bola, FrecuenciaGlobalbola = Frecuencia1silla, FrecuenciaGlobalcama = Frecuencia1cama;
12 int barridoFrecuencia = ValorFrecuencia;
13
14 //----- Create an AD9833 object -----
15 // Note, SCK and MOSI must be connected to CLK and DAT pins on the AD9833 for SPI
16 AD9833 gen(FNC_PIN); // Defaults to 25MHz internal reference frequency
17
18 //-----
19 int IdentificacionObjeto = 3; // 1-->If the object where we want to make the frequency sweep is the ball
20 // 2-->If the object where we want to make the frequency sweep is the chair
21 // 3-->If the object where we want to make the frequency sweep is the bed
22 //-----
23
24 void setup()
25 {
26     // This MUST be the first command after declaring the AD9833 object
27     gen.Begin();
28     // Apply a 1000 Hz sine wave using REG0 (register set 0). There are two register sets,
29     // REG0 and REG1.
30     // Each one can be programmed for:
31     // Signal type - SINE_WAVE, TRIANGLE_WAVE, SQUARE_WAVE, and HALF_SQUARE_WAVE
32     // Frequency - 0 to 12.5 MHz
33     // Phase - 0 to 360 degrees (this is only useful if it is 'relative' to some other signal
34     // such as the phase difference between REG0 and REG1).
35     // In ApplySignal, if Phase is not given, it defaults to 0.
36     if (IdentificacionObjeto == 1)
37     {
38         gen.ApplySignal(SINE_WAVE, REG0, Frecuencia1bola);
39     }
40     if (IdentificacionObjeto == 2)
41     {
42         gen.ApplySignal(SINE_WAVE, REG0, Frecuencia1silla);
43     }
44     if (IdentificacionObjeto == 3)
45     {
46         gen.ApplySignal(SINE_WAVE, REG0, Frecuencia1cama);
47     }
48
49     gen.EnableOutput(true); // Turn ON the output - it defaults to OFF
50     // There should be a 1000 Hz sine wave on the output of the AD9833
51     Serial.begin(115200); // Serial Port speed
52     while (!Serial);
53     pinMode(pinEntradaAnalogica, INPUT); //Indicate that the pin that we use is for input
54 }
55
56 void loop()
57 {
58     int numLecturasdatobola = 73, numLecturasdatosilla = 58, numLecturasdatocama = 48; // Total number of readings to be taken
59     int numValoresporvector = 10; // Normally I will sample 10 data for each frequency
60     char iteracion = 'a'; // Number of iteration from a(1) to j(10)(change the
61     // letter from there to a, b, c.... depending on which iteration we are doing)
62     int i = 0, j = 0;
63
64     int CondicionAgarrar = 0, CondicionPellizeo = 0, CondicionTocarConDedo = 0, CondicionSinTocar = 0;
65     int CondicionSentadoNormal = 0, CondicionSentadohaciaDelante = 0, CondicionLevantado = 0;
66     int CondicionTumbado = 0, CondicionSentadoCama = 0, CondicionLevantadoCama = 0;
67
68     float ValorEscalado = 0, sumavaloresescalados = 0, mediavaloresescalados = 0, aux = 0;
69     while (1)
70     {
71         if (IdentificacionObjeto == 1)
72         {
73             for (int i = 0; i < numLecturasdatobola;)
74             {
75                 int ValorAnalogico = analogRead(pinEntradaAnalogica);
76                 ValorEscalado = (ValorAnalogico * 3.2) / 4095;
77                 sumavaloresescalados = sumavaloresescalados + ValorEscalado;
78                 j++;
79                 if (j < numValoresporvector)
80                 {
81                     delay(1);
82                 }
83                 if (j > numValoresporvector - 1)
84                 {
85                     if (i < numLecturasdatobola - 1)
86                     {
87                         mediavaloresescalados = sumavaloresescalados / 10;
88                         j = 0;
89                         if (FrecuenciaGlobalbola >= 735000 && FrecuenciaGlobalbola <= 1015000 && mediavaloresescalados <= 1 && aux == 0)
90                         {
91                             // ...
92                         }
93                     }
94                 }
95             }
96         }
97     }
98 }

```

```

92         CondicionAgarrar = 1;
93         aux = 1;
94     }
95     if (FrecuenciaGlobalbola >= 1225000 && FrecuenciaGlobalbola < 1505000 && mediavaloresescalados <= 1.2 && aux == 0)
96     {
97         CondicionPellizco = 1;
98         aux = 1;
99     }
100    if (FrecuenciaGlobalbola >= 1505000 && FrecuenciaGlobalbola < 2030000 && mediavaloresescalados <= 1 && aux == 0)
101    {
102        CondicionTocarConDedo = 1;
103        aux = 1;
104    }
105    if (FrecuenciaGlobalbola >= 2362500 && FrecuenciaGlobalbola < 2502500 && mediavaloresescalados <= 0.6 && aux == 0)
106    {
107        CondicionSinTocar = 1;
108        aux = 1;
109    }
110    if (Frecuencia1bola <= 1015000)
111    {
112        Frecuencia1bola = Frecuencia1bola + ValorFrecuencia;
113        FrecuenciaGlobalbola = Frecuencia1bola;
114        gen.ApplySignal(SINE_WAVE, REQ0, Frecuencia1bola);
115        i++;
116        sumavaloresescalados = 0;
117        mediavaloresescalados = 0;
118        delay(1);
119    }
120    else if (Frecuencia2bola <= 1505000)
121    {
122        gen.ApplySignal(SINE_WAVE, REQ0, Frecuencia2bola);
123        Frecuencia2bola = Frecuencia2bola + ValorFrecuencia;
124        FrecuenciaGlobalbola = Frecuencia2bola;
125        sumavaloresescalados = 0;
126        mediavaloresescalados = 0;
127        i++;
128        delay(1);
129    }
130    else if (Frecuencia3bola <= 2012500)
131    {
132        gen.ApplySignal(SINE_WAVE, REQ0, Frecuencia3bola);
133        Frecuencia3bola = Frecuencia3bola + ValorFrecuencia;
134        FrecuenciaGlobalbola = Frecuencia3bola;
135        sumavaloresescalados = 0;
136        mediavaloresescalados = 0;
137        i++;
138        delay(1);
139    }
140    else if (Frecuencia4bola <= 2502500)
141    {
142        gen.ApplySignal(SINE_WAVE, REQ0, Frecuencia4bola);
143        Frecuencia4bola = Frecuencia4bola + ValorFrecuencia;
144        FrecuenciaGlobalbola = Frecuencia4bola;
145        i++;
146        sumavaloresescalados = 0;
147        mediavaloresescalados = 0;
148        delay(1);
149    }
150    }
151    Serial.print("\n");
152    if (CondicionAgarrar == 1)
153    {
154        log_e("AGARRAR");
155    }
156    if (CondicionPellizco == 1)
157    {
158        log_e("PELLIZCO");
159    }
160    if (CondicionTocarConDedo == 1)
161    {
162        log_e("TOCARCONDEDO");
163    }
164    if (CondicionSinTocar == 1)
165    {
166        log_e("SINTOCAR");
167    }
168    Frecuencia1bola = 735000;
169    Frecuencia2bola = 1225000;
170    Frecuencia3bola = 1505000;
171    Frecuencia4bola = 2362500;
172    CondicionAgarrar = 0;
173    CondicionPellizco = 0;
174    CondicionTocarConDedo = 0;
175    CondicionSinTocar = 0;
176    aux = 0;
177    }
178    if (IdentificacionObjeto == 2)
179    {
180        for (int i = 0; i < numLecturasdatosilla;)
181        {
182            int ValorAnalogico = analogRead(pinEntradaAnalogica);
183            ValorEscalado = (ValorAnalogico * 3.2) / 4095;
184            sumavaloresescalados = sumavaloresescalados + ValorEscalado;
185            J++;
186            if (j < numValoresporvector)
187            {
188                }
189            }

```

```

190     delay(1);
191     if (j > numValoresporvector - 1)
192     {
193         if (i < numLecturasdatosilla - 1)
194         {
195             }
196         mediavaloresescalados = sumavaloresescalados / 10;
197         j = 0;
198         if (FrecuenciaGlobal silla >= 1015000 && FrecuenciaGlobal silla <= 1487500 && mediavaloresescalados <= 0.8 && aux == 0)
199         {
200             CondicionSentadoNormal = 1;
201             aux = 1;
202         }
203         if (FrecuenciaGlobal silla > 1487500 && FrecuenciaGlobal silla <= 1750000 && mediavaloresescalados <= 0.8 && aux == 0)
204         {
205             CondicionSentadohaciaadelante = 1;
206             aux = 1;
207         }
208         if (FrecuenciaGlobal silla >= 1890000 && FrecuenciaGlobal silla <= 2117500 && mediavaloresescalados <= 0.2 && aux == 0)
209         {
210             CondicionLevantado = 1;
211             aux = 1;
212         }
213         if (Frecuencia3silla <= 1487500)
214         {
215             Frecuencia1silla = Frecuencia1silla + ValorFrecuencia;
216             FrecuenciaGlobal silla = Frecuencia1silla;
217             gen.ApplySignal(SINE_WAVE, REG0, Frecuencia1silla);
218             i++;
219             sumavaloresescalados = 0;
220             mediavaloresescalados = 0;
221             delay(1);
222         }
223         else if (Frecuencia2silla <= 1750000)
224         {
225             gen.ApplySignal(SINE_WAVE, REG0, Frecuencia2silla);
226             Frecuencia2silla = Frecuencia2silla + ValorFrecuencia;
227             FrecuenciaGlobal silla = Frecuencia2silla;
228             sumavaloresescalados = 0;
229             mediavaloresescalados = 0;
230             i++;
231             delay(1);
232         }
233         else if (Frecuencia3silla <= 2117500)
234         {
235             gen.ApplySignal(SINE_WAVE, REG0, Frecuencia3silla);
236             Frecuencia3silla = Frecuencia3silla + ValorFrecuencia;
237             FrecuenciaGlobal silla = Frecuencia3silla;
238             sumavaloresescalados = 0;
239             mediavaloresescalados = 0;
240             i++;
241             delay(1);
242         }
243     }
244     Serial.print("\n");
245     if (CondicionSentadoNormal == 1)
246     {
247         log_e("SENTADO CON ESPALDA APOYADA");
248     }
249     if (CondicionSentadohaciaadelante == 1)
250     {
251         log_e("SENTADO CON ESPALDA HACIA DELANTE");
252     }
253     if (CondicionLevantado == 1)
254     {
255         log_e("LEVANTADO");
256     }
257     Frecuencia1silla = 1015000;
258     Frecuencia2silla = 1487500;
259     Frecuencia3silla = 1890000;
260     CondicionSentadoNormal = 0;
261     CondicionSentadohaciaadelante = 0;
262     CondicionLevantado = 0;
263     aux = 0;
264 }
265
266 if (IdentificacionObjeto == 3)
267 {
268     for (int i = 0; i < numLecturasdatocama;)
269     {
270         int ValorAnalogico = analogRead(pinEntradaAnalogica);
271         ValorEscalado = (ValorAnalogico * 3.2) / 4095;
272         sumavaloresescalados = sumavaloresescalados + ValorEscalado;
273         j++;
274         if (j < numValoresporvector)
275         {
276             }
277         delay(1);
278         if (j > numValoresporvector - 1)
279         {
280             if (i < numLecturasdatocama - 1)
281             {
282                 }
283             mediavaloresescalados = sumavaloresescalados / 10;
284             j = 0;
285             if (FrecuenciaGlobal cama >= 717500 && FrecuenciaGlobal cama < 805000 && mediavaloresescalados <= 0.3 && aux == 0)
286             {
287                 CondicionTumbado = 1;
288                 aux = 1;
289             }
290             if (FrecuenciaGlobal cama >= 805000 && FrecuenciaGlobal cama < 997500 && mediavaloresescalados <= 0.3 && aux == 0)

```

```

291     {
292         CondicionSentadoCama = 1;
293         aux = 1;
294     }
295     if (FrecuenciaGlobalcama >= 997500 && FrecuenciaGlobalcama <= 1505000 && mediavaloresescalados <= 0.3 && aux == 0)
296     {
297         CondicionLevantadoCama = 1;
298         aux = 1;
299     }
300     if (Frecuencia1cama <= 805000)
301     {
302         Frecuencia1cama = Frecuencia1cama + ValorFrecuencia;
303         FrecuenciaGlobalcama = Frecuencia1cama;
304         gen.ApplySignal(SINE_WAVE, REG0, Frecuencia1cama);
305         i++;
306         sumavaloresescalados = 0;
307         mediavaloresescalados = 0;
308         delay(1);
309     }
310     else if (Frecuencia2cama <= 997500)
311     {
312         gen.ApplySignal(SINE_WAVE, REG0, Frecuencia2cama);
313         Frecuencia2cama = Frecuencia2cama + ValorFrecuencia;
314         FrecuenciaGlobalcama = Frecuencia2cama;
315         sumavaloresescalados = 0;
316         mediavaloresescalados = 0;
317         i++;
318         delay(1);
319     }
320     else if (Frecuencia3cama <= 1505000)
321     {
322         gen.ApplySignal(SINE_WAVE, REG0, Frecuencia3cama);
323         Frecuencia3cama = Frecuencia3cama + ValorFrecuencia;
324         FrecuenciaGlobalcama = Frecuencia3cama;
325         sumavaloresescalados = 0;
326         mediavaloresescalados = 0;
327         i++;
328         delay(1);
329     }
330 }
331 }
332 Serial.print("\n");
333 if (CondicionTumbado == 1)
334 {
335     log_e("TUMBADO");
336 }
337 if (CondicionSentadoCama == 1)
338 {
339     log_e("SENTADOENCAMA");
340 }
341 if (CondicionLevantadoCama == 1)
342 {
343     log_e("LEVANTADOCAMA");
344 }
345
346 Frecuencia1cama = 717500;
347 Frecuencia2cama = 805000;
348 Frecuencia3cama = 997500;
349 CondicionTumbado = 0;
350 CondicionSentadoCama = 0;
351 CondicionLevantadoCama = 0;
352 aux = 0;
353 }
354 }
355 }

```

-Visualización por pantalla (Se visualizarán las interacciones con la bola que hemos confeccionado como modelo real)

```
[ 38694][E][main.cpp:167] loop(): SINTOCAR
[ 39497][E][main.cpp:167] loop(): SINTOCAR
[ 40300][E][main.cpp:167] loop(): SINTOCAR
[ 41103][E][main.cpp:167] loop(): SINTOCAR
[ 41906][E][main.cpp:167] loop(): SINTOCAR
[ 42709][E][main.cpp:159] loop(): PELLIZCO
[ 43512][E][main.cpp:159] loop(): PELLIZCO
[ 44315][E][main.cpp:159] loop(): PELLIZCO
[ 45118][E][main.cpp:159] loop(): PELLIZCO
[ 45921][E][main.cpp:155] loop(): AGARRAR
[ 46724][E][main.cpp:155] loop(): AGARRAR
[ 47527][E][main.cpp:155] loop(): AGARRAR
[ 48330][E][main.cpp:155] loop(): AGARRAR
[ 49133][E][main.cpp:163] loop(): TOCARCONDEDO
[ 49936][E][main.cpp:163] loop(): TOCARCONDEDO
[ 50739][E][main.cpp:163] loop(): TOCARCONDEDO
[ 51542][E][main.cpp:167] loop(): SINTOCAR
[ 52345][E][main.cpp:167] loop(): SINTOCAR
[ 53148][E][main.cpp:167] loop(): SINTOCAR
[ 53951][E][main.cpp:167] loop(): SINTOCAR
```



CODIGO FINAL: CODIGO QUE PERMITE MUESTREAR QUE ACCION ESTAMOS REALIZANDO Y ENVIAR UN CORREO ELECTRONICO SI SUCEDE CIERTA ACTIVIDAD A UNA DETERMINADA FRANJA HORARIA

```

1 #include <AD9833.h> // Include the library
2 #include <SPI.h>
3 #include <Arduino.h>
4 #if defined(ESP32)
5 #include <WiFi.h>
6 #elif defined(ESP8266)
7 #include <ESP8266WiFi.h>
8 #endif
9 #include <ESP_Mail_Client.h>
10 #include <iostream>
11 #include <chrono>
12 #include <ctime>
13 #include <time.h>
14 #define FNC_PIN 4 // Can be any digital IO pin
15
16 const int pinEntradaAnalogica = 34;
17 int ValorFrecuencia = 17500;
18 int Frecuencia = ValorFrecuencia;
19 int barridoFrecuencia = ValorFrecuencia;
20
21 #define WIFI_SSID "CETEM_nave"
22 #define WIFI_PASSWORD "cetem_nave_ele"
23
24 /** The smtp host name e.g. smtp.gmail.com for GMail or smtp.office365.com for Outlook or smtp.mail.yahoo.com */
25 #define SMTP_HOST "smtp.gmail.com"
26 #define SMTP_PORT 465
27
28 /** The sign in credentials */
29 #define AUTHOR_EMAIL "pruebatouche@gmail.com"
30 #define AUTHOR_PASSWORD "dorm lsyj yruu wgqz"
31
32 /** Recipient's email*/
33 #define RECIPIENT_EMAIL "pruebatouche@gmail.com"
34
35 /** Declare the global used SMTPSession object for SMTP transport */
36 SMTPSession smtp;
37
38 /** Callback function to get the Email sending status */
39 void smtpCallback(SMTP_Status status);
40
41 const char *ntpServer = "pool.ntp.org";
42 const long gmtOffset_sec = 3600;
43 const int daylightOffset_sec = 3600;
44
45 void printLocalTime()
46 {
47     struct tm timeinfo;
48     if (!getLocalTime(&timeinfo))
49     {
50         Serial.println("Failed to obtain time");
51         return;
52     }
53     Serial.println(&timeinfo, "%A, %B %d %Y %H:%M:%S");
54 }
55
56 //----- Create an AD9833 object -----
57 // Note, SCK and MOSI must be connected to CLK and DAT pins on the AD9833 for SPI
58 AD9833 gen(FNC_PIN); // Defaults to 25MHz internal reference frequency
59
60 int AgarrarCorreo = 0, PellizcoCorreo = 0, TocarUnDedoCorreo = 0, SinTocarCorreo = 0;
61 int SentadoNormalCorreo=0, SentadoHaciaDelanteCorreo=0, LevantadoSillaCorreo=0;
62 int TumbadoColchonCorreo=0, SentadoColchonCorreo=0, LevantadoColchonCorreo = 0;
63 void setup()
64 {
65     // This MUST be the first command after declaring the AD9833 object
66     gen.Begin();
67     // Apply a 1000 Hz sine wave using REG0 (register set 0). There are two register sets,
68     // REG0 and REG1.
69     // Each one can be programmed for:
70     // Signal type - SINE_WAVE, TRIANGLE_WAVE, SQUARE_WAVE, and HALF_SQUARE_WAVE
71     // Frequency - 0 to 12.5 MHz
72     // Phase - 0 to 360 degrees (this is only useful if it is 'relative' to some other signal
73     // such as the phase difference between REG0 and REG1).
74     // In ApplySignal, if Phase is not given, it defaults to 0.
75     gen.ApplySignal(SINE_WAVE, REG0, Frecuencia); // Aplicamos una onda senoidal en el registro Reg0 o Reg1 y le
76     damos la frecuencia en hz que queremos(0 a 12,5Mhz)
77
78     gen.EnableOutput(true); // Turn ON the output - it defaults to OFF
79     // There should be a 1000 Hz sine wave on the output of the AD9833
80     Serial.begin(115200); // Serial port speed
81     WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

```

```

81 Serial.print("Connecting to Wi-Fi");
82 while (WiFi.status() != WL_CONNECTED)
83 {
84     Serial.print(".");
85     delay(300);
86 }
87 Serial.println();
88 Serial.print("Connected with IP: ");
89 Serial.println(WiFi.localIP());
90 Serial.println();
91 while (!Serial);
92 pinMode(pinEntradaAnalogica, INPUT); // Indicate the pin that we use
93 MailClient.networkReconnect(true);
94 /** Enable the debug via Serial port
95  * 0 for no debugging
96  * 1 for basic level debugging
97  *
98  * Debug port can be changed via ESP_MAIL_DEFAULT_DEBUG_PORT in ESP_Mail_FS.h
99  */
100 smtp.debug(1);
101 /* Set the callback function to get the sending results */
102 smtp.callback(smtpCallback);
103 /* Declare the Session_Config for user defined session credentials */
104 Session_Config config;
105 /* Set the session config */
106 config.server.host_name = SMTP_HOST;
107 config.server.port = SMTP_PORT;
108 config.login.email = AUTHOR_EMAIL;
109 config.login.password = AUTHOR_PASSWORD;
110 config.login.user_domain = "";
111 /*
112 Set the NTP config time
113 For times east of the Prime Meridian use 0-12
114 For times west of the Prime Meridian add 12 to the offset.
115 Ex. American/Denver GMT would be -6. 6 + 12 = 18
116 See https://en.wikipedia.org/wiki/Time_zone for a list of the GMT/UTC timezone offsets
117 */
118 config.time.ntp_server = F("pool.ntp.org,time.nist.gov");
119 config.time.gmt_offset = 3;
120 config.time.day_light_offset = 0;
121 /* Declare the message class */
122 SMTP_Message message;
123 /* Set the message headers */
124 message.sender.name = F("Jaime");
125 message.sender.email = AUTHOR_EMAIL;
126 message.subject = F("ALERTA ACTIVIDAD INUSUAL");
127 message.addRecipient(F("Jaime"), RECIPIENT_EMAIL);
128
129 /*Send HTML message*/
130 /*String htmlMsg = "<div style='color:#2f4468;'><h1>Hello World!</h1><p>- Sent from ESP board</p></div>";
131 message.html.content = htmlMsg.c_str();
132 message.html.content = htmlMsg.c_str();
133 message.text.charset = "us-ascii";
134 message.html.transfer_encoding = Content_Transfer_Encoding::enc_7bit;*/
135
136 // Send raw text message
137 String textMsg = "Se esta produciendo una anomalia en el horario habitual";
138 message.text.content = textMsg.c_str();
139 message.text.charset = "us-ascii";
140 message.text.transfer_encoding = Content_Transfer_Encoding::enc_7bit;
141
142 message.priority = esp_mail_smtp_priority::esp_mail_smtp_priority_low;
143 message.response.notify = esp_mail_smtp_notify_success | esp_mail_smtp_notify_failure |
esp_mail_smtp_notify_delay;
144 /* Connect to the server */
145 if (!smtp.connect(&config))
146 {
147     ESP_MAIL_PRINTF("Connection error, Status Code: %d, Error Code: %d, Reason: %s", smtp.statusCode(),
smtp.errorCode(), smtp.errorReason().c_str());
148     return;
149 }
150 if (!smtp.isLoggedIn())
151 {
152     Serial.println("\nNot yet logged in.");
153 }
154 else
155 {
156     if (smtp.isAuthenticated())
157         Serial.println("\nSuccessfully logged in.");
158     else
159         Serial.println("\nConnected with no Auth.");
160 }
161 /* Start sending Email and close the session */
162 // if (!MailClient.sendMail(&smtp, &message))
163 // ESP_MAIL_PRINTF("Error, Status Code: %d, Error Code: %d, Reason: %s", smtp.statusCode(), smtp.errorCode(),
smtp.errorReason().c_str());
164 Serial.println(" CONNECTED");
165
166 // init and get the time
167 configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
168 printLocalTime();
169 // disconnect WiFi as it's no longer needed
170 // WiFi.disconnect(true);

```

```

171 // WiFi.mode(WIFI_OFF);
172 }
173 int aux2=0;
174 void loop()
175 {
176   int numLecturasdato = 200; // Total number of readings to be taken
177   int numValoresporvector = 15; // Normally I will sample 10 data for each frequency collected.
178   char iteracion = 'a'; // Number of iteration from a(1) to j(10)(change the letter from there to a, b,
   c.... depending on which iteration we are doing)
179
180   int IdentificacionDeObjeto = 1; // 1-->If the object where we want to make the frequency sweep is the ball
181   // 2-->If the object where we want to make the frequency sweep is the chair
182   // 3-->If the object where we want to make the frequency sweep is the bed
183
184   int i = 0, j = 0, contfrecuencia = 0;
185   float ValorEscalado = 0;
186   float SumaValorEscalado = 0, ValorMedioPresente = 0, ValorMedioAnterior = 0;
187   int hour1=11, hour2=12, minute1=00, minute2=00, second1=00, second2=00; //Define the time interval
188   int CondicionAgarrar = 0, CondicionPellizco = 0, CondicionTocarUnDedo = 0, CondicionSinTocar = 0, aux = 0,
   aux2=0;
189   int CondicionSentado = 0, CondicionLevantado = 0, CondicionSentadohaciadelante = 0;
190   int CondicionTumbado = 0, CondicionSentadoColchon = 0, CondicionLevantadoColchon = 0;
191   while (1)
192   {
193     delay(1000);
194     printLocalTime();
195     struct tm timeinfo;
196     if (getLocalTime(&timeinfo))
197     {
198       int currentHour = timeinfo.tm_hour;
199       int currentMinute = timeinfo.tm_min;
200       int currentSecond = timeinfo.tm_sec;
201
202       //-----//
203       //CHANGE THE TIME AND CONDITION FOR SENDING A MAILING
204       if (currentHour >= hour1 && currentHour <= hour2 && aux2==0 && AgarrarCorreo == 1)
205       {
206         //-----//
207
208         Serial.println("¡Es la hora deseada!");
209         /* Declare the Session_Config for user defined session credentials */
210         Session_Config config;
211         /* Set the session config */
212         config.server_host_name = SMTP_HOST;
213         config.server_port = SMTP_PORT;
214         config.login.email = AUTHOR_EMAIL;
215         config.login.password = AUTHOR_PASSWORD;
216         config.login.user_domain = "";
217         config.time.ntp_server = F("pool.ntp.org,time.nist.gov");
218         config.time.gmt_offset = 3;
219         config.time.day_light_offset = 0;
220         /* Connect to the server */
221         if (!smtp.connect(&config))
222         {
223           ESP_MAIL_PRINTF("Connection error, Status Code: %d, Error Code: %d, Reason: %s", smtp.statusCode(),
   smtp.errorCode(), smtp.errorReason().c_str());
224           return;
225         }
226         if (!smtp.isLoggedIn())
227         {
228           Serial.println("\nNot yet logged in.");
229         }
230         else
231         {
232           if (smtp.isAuthenticated())
233             Serial.println("\nSuccessfully logged in.");
234           else
235             Serial.println("\nConnected with no Auth.");
236         }
237         /* Declare the message class */
238         SMTP_Message message;
239         /* Set the message headers */
240         message.sender.name = F("Jaime");
241         message.sender.email = AUTHOR_EMAIL;
242         message.subject = F("ALERTA ACTIVIDAD INUSUAL");
243         message.addRecipient(F("Jaime"), RECIPIENT_EMAIL);
244
245         /*Send HTML message*/
246         /*String htmlMsg = "<div style=\"color:#2f4468;\"><h1>Hello World!</h1><p>- Sent from ESP board</p>
   </div>";
247         message.html.content = htmlMsg.c_str();
248         message.html.content = htmlMsg.c_str();
249         message.text.charset = "us-ascii";
250         message.html.transfer_encoding = Content_Transfer_Encoding::enc_7bit;*/
251
252         // Send raw text message
253         String textMsg = "Se esta produciendo una anomalia en el horario habitual";
254         message.text.content = textMsg.c_str();
255         message.text.charset = "us-ascii";
256         message.text.transfer_encoding = Content_Transfer_Encoding::enc_7bit;
257
258         message.priority = esp_mail_smtp_priority::esp_mail_smtp_priority_low;
259         message.response.notify = esp_mail_smtp_notify_success | esp_mail_smtp_notify_failure |
   esp_mail_smtp_notify_delay;
260         /* Start sending Email and close the session */

```

```

261     if (!MailClient.sendMail(&smtp, &message))
262         ESP_MAIL_PRINTF("Error, Status Code: %d, Error Code: %d, Reason: %s", smtp.statusCode(),
smtp.errorCode(), smtp.errorReason().c_str());
263     // Close the connection after sending mail
264     aux2=1;
265     }
266     else if(currentHour > hour2){ //Condition to send me only one mail per day and not infinite mails if a
certain activity happens.
267         aux2=0
268     }
269     AgarrarCorreo = 0;
270     PellizcoCorreo = 0;
271     TocarUnDedoCorreo = 0;
272     SinTocarCorreo = 0;
273     SentadoNormalCorreo = 0;
274     SentadoHaciaDelanteCorreo = 0;
275     LevantadoSillaCorreo = 0;
276     TumbadoColchonCorreo = 0;
277     SentadoColchonCorreo = 0;
278     LevantadoColchonCorreo = 0;
279     }
280     barridoFrecuencia = barridoFrecuencia + ValorFrecuencia;
281     for (int i = 0; i < numLecturasdato;)
282     { // Para cambiar el numero de frecuencias pero que para cada frecuencia se tomen 15 valores, se cambia
numLecturasdato a 200(o las frecuencias que queramos en el barrido) y la condicion de if(i<15) lo ponemos a if(i<
numero que queramos de datos de frecuencias)
283         int ValorAnalogico = analogRead(pinEntradaAnalogica);
284         ValorEscalado = (ValorAnalogico * 3.2) / 4095;
285         j++;
286         SumaValorEscalado = ValorEscalado + SumaValorEscalado;
287         delay(1);
288         if (j > numValoresporvector - 1)
289         {
290             if (i < numLecturasdato - 1)
291             {
292                 confrecuencia = confrecuencia + 17500;
293                 if (ValorMedioAnterior != 0)
294                 {
295                     ValorMedioAnterior = ValorMedioPresente;
296                     ValorMedioPresente = SumaValorEscalado / 15;
297                 }
298
299                 if (ValorMedioPresente == 0)
300                 {
301                     ValorMedioAnterior = SumaValorEscalado / 15;
302                     ValorMedioPresente = SumaValorEscalado / 15;
303                 }
304                 if (IdentificacionDeObjeto == 1)
305                 {
306                     if (confrecuencia >= 735000 && confrecuencia <= 1225000 && (ValorMedioPresente > ValorMedioAnterior)
&& ValorMedioPresente <= 0.9 && aux == 0)
307                     {
308                         CondicionAgarrar = 1;
309                         aux = 1;
310                     }
311                     if (confrecuencia > 1225000 && confrecuencia < 1505000 && (ValorMedioPresente > ValorMedioAnterior)
&& ValorMedioPresente <= 1 && aux == 0)
312                     {
313                         CondicionPellizco = 1;
314                         aux = 1;
315                     }
316                     if (confrecuencia >= 1505000 && confrecuencia < 2030000 && (ValorMedioPresente > ValorMedioAnterior)
&& ValorMedioPresente <= 1 && aux == 0)
317                     {
318                         CondicionTocarUnDedo = 1;
319                         aux = 1;
320                     }
321                     if (confrecuencia >= 2030000 && confrecuencia < 2502500 && (ValorMedioPresente > ValorMedioAnterior)
&& ValorMedioPresente <= 1 && aux == 0)
322                     {
323                         CondicionSinTocar = 1;
324                         aux = 1;
325                     }
326                 }
327                 if (IdentificacionDeObjeto == 2)
328                 {
329                     if (confrecuencia >= 1015000 && confrecuencia <= 1487500 && (ValorMedioPresente >
ValorMedioAnterior) && ValorMedioPresente <= 0.8 && aux == 0)
330                     {
331                         CondicionSentado = 1;
332                         aux = 1;
333                     }
334                     if (confrecuencia > 1487500 && confrecuencia <= 1750000 && (ValorMedioPresente > ValorMedioAnterior)
&& ValorMedioPresente <= 0.8 && aux == 0)
335                     {
336                         CondicionSentadohaciadelante = 1;
337                         aux = 1;
338                     }
339                     if (confrecuencia >= 1890000 && confrecuencia <= 2117500 && (ValorMedioPresente >
ValorMedioAnterior) && ValorMedioPresente <= 0.8 && aux == 0)
340                     {
341                         CondicionLevantado = 1;
342                         aux = 1;
343                     }
344                 }
345                 if (IdentificacionDeObjeto == 3)
346                 {
347                     if (confrecuencia >= 717500 && confrecuencia <= 805000 && (ValorMedioPresente > ValorMedioAnterior)
&& ValorMedioPresente <= 0.3 && aux == 0)
348                     {
349                         CondicionTumbado = 1;
350                         aux = 1;

```

```

351     }
352     if (contfrecuencia > 805000 && contfrecuencia <= 997500 && (ValorMedioPresente > ValorMedioAnterior)
&& ValorMedioPresente <= 0.3 && aux == 0)
353     {
354         CondicionSentadoColchon = 1;
355         aux = 1;
356     }
357     if (contfrecuencia > 997500 && contfrecuencia <= 1505000 && (ValorMedioPresente > ValorMedioAnterior)
&& ValorMedioPresente <= 0.3 && aux == 0)
358     {
359         CondicionLevantadoColchon = 1;
360         aux = 1;
361     }
362     }
363     SumaValorEscalado = 0;
364     }
365     j = 0;
366     Frecuencia = Frecuencia + ValorFrecuencia;
367     if (i < numLecturasdato - 1)
368     {
369         gen.ApplySignal(SINE_WAVE, REG0, Frecuencia);
370     }
371     i++;
372     delay(1);
373     }
374     }
375     i = 0;
376     contfrecuencia = 0;
377     Frecuencia = ValorFrecuencia;
378     ValorMedioAnterior = 0;
379     ValorMedioPresente = 0;
380     if (CondicionAgarrar == 1)
381     {
382         log_i("\nAGARRAR");
383         AgarrarCorreo = 1;
384     }
385     if (CondicionPellizco == 1)
386     {
387         log_i("\nPELLIZCO");
388         PellizcoCorreo = 1;
389     }
390     if (CondicionTocarUnDedo == 1)
391     {
392         log_i("\nTOCARCONDEDO");
393         TocarUnDedoCorreo = 1;
394     }
395     if (CondicionSinTocar == 1)
396     {
397         log_i("\nSINTOCAR");
398         SinTocarCorreo = 1;
399     }
400     if (CondicionSentado == 1)
401     {
402         log_i("\nSENTADONORMAL");
403         SentadoNormalCorreo = 1;
404     }
405     if (CondicionSentadohaciaadelante == 1)
406     {
407         log_i("\nSENTADOHACIADELANTE");
408         SentadoHaciaDelanteCorreo = 1;
409     }
410     if (CondicionLevantado == 1)
411     {
412         log_i("\nLEVANTADO");
413         LevantadoSillaCorreo = 1;
414     }
415     if (CondicionTumbado == 1)
416     {
417         log_i("\nTUMBADO");
418         TumbadoColchonCorreo = 1;
419     }
420     if (CondicionSentadoColchon == 1)
421     {
422         log_i("\nSENTADOCOLCHON");
423         SentadoColchonCorreo = 1;
424     }
425     if (CondicionLevantadoColchon == 1)
426     {
427         log_i("\nLEVANTADOCOLCHON");
428         LevantadoColchonCorreo = 1;
429     }
430     CondicionAgarrar = 0;
431     CondicionPellizco = 0;
432     CondicionTocarUnDedo = 0;
433     CondicionSinTocar = 0;
434     CondicionSentado = 0;
435     CondicionLevantado = 0;
436     CondicionSentadohaciaadelante = 0;
437     CondicionTumbado = 0;
438     CondicionSentadoColchon = 0;
439     CondicionLevantadoColchon = 0;
440     aux = 0;

```

```

441     Serial.print("\n");
442     delay(1);
443 }
444 }
445
446 /* Callback function to get the Email sending status */
447 void smtpCallback(SMTP_Status status)
448 {
449     /* Print the current status */
450     Serial.println(status.info());
451
452     /* Print the sending result */
453     if (status.success())
454     {
455         // ESP_MAIL_PRINTF used in the examples is for format printing via debug Serial port
456         // that works for all supported Arduino platform SDKs e.g. AVR, SAMD, ESP32 and ESP8266.
457         // In ESP8266 and ESP32, you can use Serial.printf directly.
458
459         Serial.println("-----");
460         ESP_MAIL_PRINTF("Message sent success: %d\n", status.completedCount());
461         ESP_MAIL_PRINTF("Message sent failed: %d\n", status.failedCount());
462         Serial.println("-----\n");
463
464         for (size_t i = 0; i < smtp.sendingResult.size(); i++)
465         {
466
467             /* Get the result item */
468             SMTP_Result result = smtp.sendingResult.getItem(i);
469
470             // In case, ESP32, ESP8266 and SAMD device, the timestamp get from result.timestamp should be valid if
471             // your device time was synched with NTP server.
472             // Other devices may show invalid timestamp as the device time was not set i.e. it will show Jan 1, 1970.
473             // You can call smtp.setSystemTime(xxx) to set device time manually. Where xxx is timestamp (seconds since
474             // Jan 1, 1970)
475             ESP_MAIL_PRINTF("Message No: %d\n", i + 1);
476             ESP_MAIL_PRINTF("Status: %s\n", result.completed ? "success" : "failed");
477             ESP_MAIL_PRINTF("Date/Time: %s\n", MailClient.Time.getDateTimeString(result.timestamp, "%B %d, %Y
%H:%M:%S").c_str());
478             ESP_MAIL_PRINTF("Recipient: %s\n", result.recipients.c_str());
479             ESP_MAIL_PRINTF("Subject: %s\n", result.subject.c_str());
480         }
481         Serial.println("-----\n");
482
483         // You need to clear sending result as the memory usage will grow up.
484         smtp.sendingResult.clear();
485     }
486 }

```

-Visualización por pantalla y en el correo electrónico

```

[ 7][D][esp32-hal-cpu.c:244] setCpuFrequencyMhz(): PLL: 480 / 2 = 240 Mhz, APB: 80000000 Hz
[ 175][D][WiFiGeneric.cpp:1035] _eventCallback(): Arduino Event: 0 - WIFI_READY
[ 265][V][WiFiGeneric.cpp:340] _arduino_event_cb(): STA Started
[ 268][D][WiFiGeneric.cpp:1035] _eventCallback(): Arduino Event: 2 - STA_START
[ 278][V][WiFiGeneric.cpp:97] set_esp_interface_ip(): Configuring Station static IP: 0.0.0.0, MASK: 0.0.0.0, GW: 0.0.0.0
Connecting to Wi-Fi...[ 949][V][WiFiGeneric.cpp:355] _arduino_event_cb(): STA Connected: SSID: CETEM_nave, BSSID: 58:6d:8f:2c:c6:14, Channel: 10, Auth: WPA2_PSK
[ 951][D][WiFiGeneric.cpp:1035] _eventCallback(): Arduino Event: 4 - STA_CONNECTED
[ 1471][V][WiFiGeneric.cpp:369] _arduino_event_cb(): STA Got New IP:10.12.54.213
[ 1471][D][WiFiGeneric.cpp:1035] _eventCallback(): Arduino Event: 7 - STA_GOT_IP
[ 1476][D][WiFiGeneric.cpp:1098] _eventCallback(): STA IP: 10.12.54.213, MASK: 255.255.255.0, GW: 10.12.54.200

Connected with IP: 10.12.54.213

#### Connecting to SMTP server...
> C: ESP Mail Client v3.4.9
> C: Reading time from NTP server.....
> C: connecting to SMTP server
> C: Host > smtp.gmail.com
> C: Port > 465
[ 25750][V][WiFiClientImpl.h:317] tcpConnect(): Starting socket

#### SMTP server connected
> C: SMTP server connected, wait for greeting...
< S: 220 smtp.gmail.com ESMTP z16-20020a05600c221000b0040472ad9a3dsm2538095wml.14 - gsmt

#### Sending greeting response...
> C: send SMTP command, EHLO
< S: 250-smtp.gmail.com at your service, [193.31.36.73]
< S: 250-SIZE 35882577
< S: 250-8BITMIME
< S: 250-AUTH LOGIN PLAIN XOAUTH2 PLAIN-CLIENTTOKEN OAUTHBEARER XOAUTH
< S: 250-ENHANCEDSTATUSCODES
< S: 250-PIPELINING
< S: 250-CHUNKING
< S: 250-SMTPUTF8

#### Logging in...
> C: send SMTP command, AUTH PLAIN
> C: pruebatouche@gmail.com
> C: *****
< S: 235 2.7.0 Accepted

```



```

Successfully logged in.
CONNECTED
Friday, November 17 2023 13:35:04
Friday, November 17 2023 13:35:05
[ 31056][I][main.cpp:397] loop():
SINTOCAR

Friday, November 17 2023 13:35:09
[ 35263][I][main.cpp:397] loop():
SINTOCAR

Friday, November 17 2023 13:35:14
[ 39468][I][main.cpp:392] loop():
TOCARCONDEDO

Friday, November 17 2023 13:35:18
[ 43669][I][main.cpp:387] loop():
PELLIZCO

Friday, November 17 2023 13:35:22
[ 47870][I][main.cpp:382] loop():
AGARRAR

Friday, November 17 2023 13:35:26
¡Es la hora deseada!
#### Connecting to SMTP server...
> C: ESP Mail Client v3.4.9
> C: connecting to SMTP server
> C: Host > smtp.gmail.com
> C: Port > 465
[ 48886][V][WiFiClientImpl.h:317] tcpConnect(): Starting socket

#### SMTP server connected
> C: SMTP server connected, wait for greeting...
< S: 220 smtp.gmail.com ESMTP m5-20020a056000180500b003316a32cb25sm1746892wrh.86 - gsmt

#### Sending greeting response...
> C: send SMTP command, EHLO
< S: 250-smtp.gmail.com at your service, [193.31.36.73]
< S: 250-SIZE 35882577
< S: 250-8BITMIME
< S: 250-AUTH LOGIN PLAIN XOAUTH2 PLAIN-CLIENTTOKEN OAUTHBEARER XOAUTH
< S: 250-ENHANCEDSTATUSCODES
< S: 250-PIPELINING
< S: 250-CHUNKING
< S: 250-SMTPUTF8

#### Logging in...
> C: send SMTP command, AUTH PLAIN
> C: pruebatouche@gmail.com
> C: *****
< S: 235 2.7.0 Accepted

Successfully logged in.

#### Sending Email...
> C: send Email

#### Sending message header...
> C: send message header
< S: 250 2.1.0 OK m5-20020a056000180500b003316a32cb25sm1746892wrh.86 - gsmt
< S: 250 2.1.5 OK m5-20020a056000180500b003316a32cb25sm1746892wrh.86 - gsmt

#### Sending message body...
> C: send message body
< S: 354 Go ahead m5-20020a056000180500b003316a32cb25sm1746892wrh.86 - gsmt

#### Finishing the message sending...
> C: finishing the message sending
< S: 250 2.0.0 OK 1700224530 m5-20020a056000180500b003316a32cb25sm1746892wrh.86 - gsmt

#### Closing the session...
> C: terminate the SMTP session
< S: 221 2.0.0 closing connection m5-20020a056000180500b003316a32cb25sm1746892wrh.86 - gsmt

#### Message sent successfully
> C: message sent successfully

-----
Message sent success: 1
Message sent failed: 0
-----

Message No: 1
Status: success
Date/Time: November 17, 2023 13:35:29
Recipient: pruebatouche@gmail.com
Subject: ALERTA ACTIVIDAD INUSUAL

-----
[ 56119][I][main.cpp:397] loop():
SINTOCAR

```

