industriales
etsii

Escuela Técnica Superior de Ingeniería Industrial

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería Industrial

Desarrollo de interfaz para monitorización en instrumentos de percusión a partir de sensores piezoeléctricos

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

Autor: **Jesús Balsalobre Martínez**
Director: Joaquín Roca González

Cartagena, a 3 de Octubre de 2017

Universidad Politécnica de Cartagena

Agradecimientos

Quiero incluir en este apartado a todas aquellas personas que, gracias a su apoyo y cariño, han hecho que llegue a ser quien soy hoy en día y que haya llegado hasta aquí.

A mis hermanos, Paula y David, por ser un pilar fundamental en mi vida y saber que siempre estarán donde los necesite.

A mi madre Santi, por todo lo que ha aguantado, lo que ha trabajado y la ilusión que siempre le ha hecho que llegara a tener unos estudios superiores.

Y en especial a mi padre Jesús, ya que, además de trabajar día y noche para poder darme una educación, es el responsable de que me interesara por la música y la electrónica, pasiones que se reúnen en este proyecto. Te lo prometí Papá.

Por último, muchas gracias a mi tutor Joaquín Roca González por su amplio conocimiento de la materia, por sus consejos y por darme la oportunidad de poder acabar mis estudios de Grado con este trabajo.

Índice

Índice de imágenes

1 Introducción

1.1 Antecedentes

La idea de este proyecto nace de la necesidad de cubrir un vacío existente en la industria musical que permita grabar instrumentos de percusión independientemente del lugar donde estos se encuentren.

Es una situación común que la elección de un sitio o estudio de grabación venga condicionado únicamente por el sonido de la batería, debido a la acústica del lugar, ya que el resto de instrumentos (guitarra, bajo, piano, trompeta, etc.) son inmunes a esta situación.

Además, este proyecto también está orientado a usuarios de baterías acústicas que se encuentren en la situación de no poder tocar por problemas de ruido (vivienda no insonorizada, problemas de vecinos, horarios incompatibles) o como una opción económica a la hora de poder grabar un instrumento de percusión, ya que el gasto en grabación es muy elevado (micrófonos, jirafas, mesa de mezclas, etc).

La motivación para desarrollar este proyecto viene de una combinación de temas que a mí, personalmente, me apasionan. Mi padre me introdujo en la música y en la batería desde pequeño, lo que me ha permitido conocer ampliamente el instrumento, tanto partes, técnicas y configuraciones. La electrónica, por otro lado, siempre me ha interesado y de hecho fue la carrea por la que me decanté.

El proyecto ha estado en mi cabeza desde hace bastante tiempo, pero ahora, finalizando mis estudios universitarios, con la oportunidad del trabajo de fin de grado y el apoyo de mi tutor de TFG (muchas gracias Joaquín) he visto la oportunidad de desarrollarlo ampliamente.

1.2 Objetivos

Los objetivos de este proyecto consisten en la realización de las distintas partes que, una vez unidas en integradas, constituirían el proyecto en su totalidad.

A la hora de estructurar y dividir las distintas partes del trabajo se planteó la siguiente división de tareas que, a su vez, se consideraron como objetivos a cumplir para ir, poco a poco, avanzando en el desarrollo y construcción de nuestro proyecto:

- Adaptación de sensores piezoeléctricos a nuestro propósito: sabemos cómo funcionan de forma general este tipo de sensores, pero aun así debemos adaptarlos a nuestra aplicación particular mediante el uso de algún tipo de acondicionamiento o algún tipo de procesamiento de señal.

- Tipo de montaje físico: estos sensores deben de montarse en las distintas partes de la batería de alguna forma establecida, por lo que este objetivo consistirá en diseñar un sistema de montaje para ellos.
- Procesado de las señales de los sensores piezoeléctricos: aquí se incluirá todo el cálculo y el código necesario que se realizará desde la llegada de las señales de los sensores hasta el envío de los datos MIDI correspondientes a cada golpeo.
- Cableado: aunque parezca un apartado sin importancia, en esta parte habrá que valorar qué tipo de conectores se van a usar, tipo de cable, valorar si la longitud puede afectar a la señal, adaptaciones o acondicionamientos necesarios y otras complicaciones que puedan surgir.
- Decisión del microncontrolador: todo el procesamiento, cálculo y envío de datos debe llevarse a cabo mediante un microcontrolador. La variedad de modelos a elegir es muy extensa, por lo que la elección de uno es un punto vital del proyecto.
- Integración del sistema: hay que añadir la unión de todas las partes en un todo que acabará siendo el proyecto en sí. En esta parte de integración suelen aparecer problemas que no se habían detectado cuando se trabajaban las partes por separado, por lo que es importante contarla como una más.

Una vez identificados los objetivos de nuestro proyecto, podemos hacer uso de otros objetivos secundarios o parciales que nos ayuden a desarrollar los objetivos principales:

- Revisión del estado del arte: este estudio sobre los distintos aspectos del instrumento y del protocolo nos ayudará a la hora de integrar las diferentes funcionalidades en el sistema que se pretende diseñar.
- Diseño y caracterización de los sensores: la realización de ciertas pruebas sobre los sensores en un laboratorio con equipo especializado (osciloscopio por ejemplo) puede ayudarnos a saber las características de estos y, por ende, ayudarnos en la realización de cualquier acondicionamiento.
- Elección del programa musical: aunque el sistema se diseñará para ser una interfaz MIDI (independiente del programa) deberemos elegir uno con el que realizaremos las distintas pruebas y experimentos.
- Pruebas intermedias: es interesante también la realización de pruebas intermedias no realizadas directamente sobre el sistema , sino sobre partes en concreto o técnicas que no conozcamos y queramos utilizar (uso del monitor serie para depurar código, uso de códigos de ejemplo de algunas librerías, etc).

2 Revisión del estado del arte

2.1 Batería

2.1.1 Historia y artistas signifcativos

La batería es un conjunto de instrumentos de percusión, montados de manera tal, que un solo músico sea capaz de tocarlos todos de manera coordinada y simultánea. Estos instrumentos de percusión tal cual los conocemos hoy en día, tuvieron orígenes en distintas partes del mundo. Se sabe que en África, se originaron los tambores, así como también otros que provienen de China; sin embargo, en el caso del bombo, este procede de Europa. Otro elemento muy utilizado en las baterías, son los platillos o platos, que también se les pueden denominar címbalos o cimbales (procedente del término inglés *cymbals*), los cuales nacieron en medio oriente.

Se sabe que entre los años 1890 y 1910, existían orquestas en las cuales había de tres a cuatro percusionistas que tocaban varios elementos por separado, por ejemplo, uno tocaba la caja (también conocida como redoblante o redo en Sudamérica), otro músico se encargaba del bombo, y otros de distintos elementos más, como los platos y panderos. Tras las pérdidas generadas por la primera guerra mundial, se vio la necesidad de unir varios instrumentos de percusión debido a las pérdidas de músicos, que luego de manera básica, constituyeron lo que más tarde conoceríamos como la batería actual, pero para esto pasó por una serie de transformaciones a través de los años.

En el año 1910, fueron comercializados por primera vez, los pedales para bombo, y los soportes de caja, por la compañía Ludwig. Esto permitió el ensamblaje de distintos elementos y así poder ser ejecutados por un solo instrumentista. Cabe destacar que en aquellos tiempos, la batería no era como hay la conocemos, sino que tenían elementos poco usuales como los cajas chinas y un solo plato suspendido, y en cuanto a los parches, estos eran de piel animal.

Para principios del siglo XX, la batería no jugaba ningún papel relevante ni protagónico en las obras musicales, sino que cumplía con el rol de mantener el tempo de las canciones, basándose solo en los rudimentos del tambor clásico y marchas muy parecidas a las militares. Más adelante empiezan a aparecer mejoras en su configuración, surgiendo otros tipos de soportes para ensamblar elementos adicionales al bombo, se empiezan a utilizar los primeros toms con la capacidad de afinarse, y se incorporan los llamados sock cymbals, Low Boy o low hat, que eran unos platillos que se tocaban con la ayuda de un pedal, concepto que más tarde daría lugar al actual Charles o Hi-Hat.

Imagen 1. Batería de 1920

Imagen 2. Low Hat de 1930

Con la llegada de la década de los 30, se hacen necesarios los baterías de ritmo debido al auge de la música para bailar, y la multiplicación de clubes de baile por doquier. Con esto, la fabricación de las baterías mejoró en tecnología y la calidad de los materiales con los cuales eran fabricadas, trayendo consigo la inclusión de otros implementos como el pedal de charles, lo cual transformó la manera de llevar el pulso, cambiando el desplazamiento de los sonidos de la batería a través del ritmo. De esta manera, poco a poco, los bateristas fueron ampliando la variedad de ritmos, incrementando la versatilidad del instrumento, con la implementación del plato ride, y las modificaciones hechas a los toms, en lo que refiere a afinación. Estos términos serán explicados más adelante.

Para el año 1937, con el virtuoso baterista Gene Krupa, la ejecución de la batería gana mayor protagonismo, con esto, los baterías se convirtieron en auténticos solistas de sus agrupaciones y orquestas. Fue así como incluso, la forma de configurar la batería, cambió, y los fabricantes tomando en cuenta las ideas de los bateristas de la época, llegaron a formar el set de instrumentos que conocemos hoy en día.

Es importante destacar, que ya para los años 50 se introducen los parches fabricados con materiales sintéticos, con la marca Remo y el fabricante Rogers, específicamente en el año 1959, crea un sistema de soportes, con los que los toms, quedarían fijados sobre el bombo.

Imagen 3. Batería Rogers con sistema de montaje de tom sobre bombo

Con el boom del rock'n'roll, la llegada de grupos musicales como The Beatles y Rolling Stone, la batería se vuelve un instrumento muy popular entre los jóvenes de la época. Los fabricantes de baterías, comienzan a implementar materiales diferentes llegando a producir baterías con un sonido de alta calidad. La década de los 60, trae consigo la aparición de muchos talentosos bateristas entre los cuales se pueden destacar Keith moon, de la banda The Who y más tarde John Bonham junto a Led Zeppelin e Ian Paice en Deep Purple.

Imagen 4. Ian Paice tocando junto a Deep Purple

Durante los años 70, llegan al mercado las baterías procedentes de fabricantes asiáticos, como la marca japonesa Tama, quienes proporcionaron una fuerte competencia a las baterías americanas y europeas, debido a que estas eran de alta calidad y precios bajos. Otro de los aspectos a destacar en las historia de la batería, es que en esta se inventan las baterías electrónicas, así como también la utilización de sonidos y ritmos sintetizados. De aquí en adelante la influencia de la música rock sobre el jazz, y también del blues, contribuyeron en la evolución de la batería, hasta llegar a la configuración que conocemos actualmente.

2.1.2 Fundamento físico y acústico

La batería, como ya se ha aclaro, se encuadra dentro de los instrumentos de percusión ya que su sonido se produce cuando uno de sus partes es golpeada. La percusión se distingue por la variedad de timbres que es capaz de producir y por su facilidad de adaptación con otros instrumentos musicales. Cabe destacar la gran variedad de sonidos que se pueden conseguir en función de las baquetas que se usen para golpear cada uno de los timbales o platos que componen la batería.

Cualquier instrumento de percusión puede ser usado para crear patrones rítmicos, e incluso algunos para crear melodías con afinaciones estándar (xilófono en este caso, la batería no entraría en este grupo). Se suele acompañar a otros instrumentos con el fin de crear y mantener el ritmo, como se puede ver en cualquier grupo o banda de rock, pop o en orquestas de jazz o blues.

Si tuviéramos que dividir los instrumentos de percusión en dos grupos, estos podrían ser:

- De altura puntual definida: los que producen notas identificables, o lo que es lo mismo, aquellos cuya altura de sonido está determinada (dan una nota determinada). Ejemplos: timbales de orquesta, xilófono, vibráfono.
- De altura puntual no definida: aquellos cuyas notas no son identificables, es decir, producen notas de una altura indeterminada. Entre ellos el cencerro, las claves, la caja o el bombo (incluyendo con esto a la batería).

Si hablamos particularmente de la batería, sabemos que el sonido consiste en las vibraciones del parche (membrana). Las membranas son cuerpos de superficie grande con relación a su espesor; excitadas por percusión o fricción emiten sonidos caracterizados por un complejo grande de parciales discordantes. Las membranas necesitan tensión previa para vibrar. Las membranas circulares (más usadas) no producen series armónicas. Se producen nodos radiales y circulares. Las figuras que identifican los diferentes modos de oscilación fueron estudiadas por Chladni y suelen conocerse con ese nombre: figuras de Chladni.

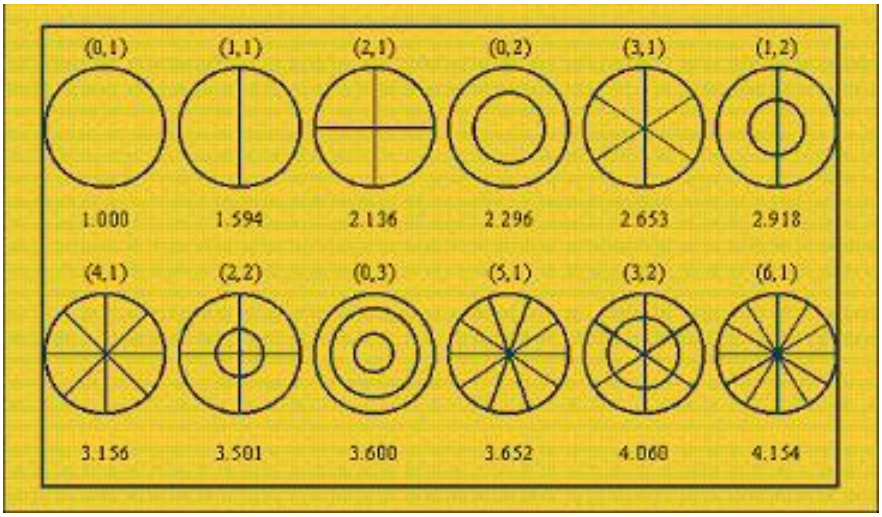


Imagen 5. Doce primeros modos de vibración de una membrana circular ideal

En esta figura se pueden ver representados los doce primeros modos de vibración, en orden creciente. Para denominarlas se utiliza una notación compuesta por dos dígitos: con el primero se indicar el número de nodos diametrales y con el segundo el número de nodos circulares. En el modo fundamental (0,1) toda la membrana se mueve en fase. La frecuencia de vibración se expresa como múltiplo de la del modo fundamental y aparece debajo de cada diagrama particular. La secuencia no forma una serie armónica.

De manera similar a las cuerdas, la frecuencia más grave de la onda de una membrana en oscilación será directamente proporcional a la tensión a la que está sometida e inversamente proporcional a su radio y a densidad de superficie de la misma. Tímbricamente dependerá fundamentalmente del material con el cual está construida la membrana, pero también del punto en el cual sea excitada y el tipo de baqueta que se use para excitarla (en rigor, la superficie de la baqueta que tenga contacto con la membrana).

La vibración de membranas, se basa en los mismos principios que la vibración de cuerdas, ya que son materiales elásticos tensados. La diferencia, es que mientras la cuerda es una línea de puntos vibrando, la membrana es una superficie, y los puntos nodales de la cuerda se transforman en líneas nodales en la membrana; por consiguiente las ondas lineales en la cuerda, son de tipo superficial en la membrana, por lo que las ondas estacionarias son de tipo bidimensional.

Si comparamos la vibración de una cuerda (cuya vibración da una altura determinada y, por ende, una nota) afinada en La (primera octava, 107,8 Hz) con la vibración de un parche de un timbal base de 16 pulgadas podemos ver ciertas diferencias.

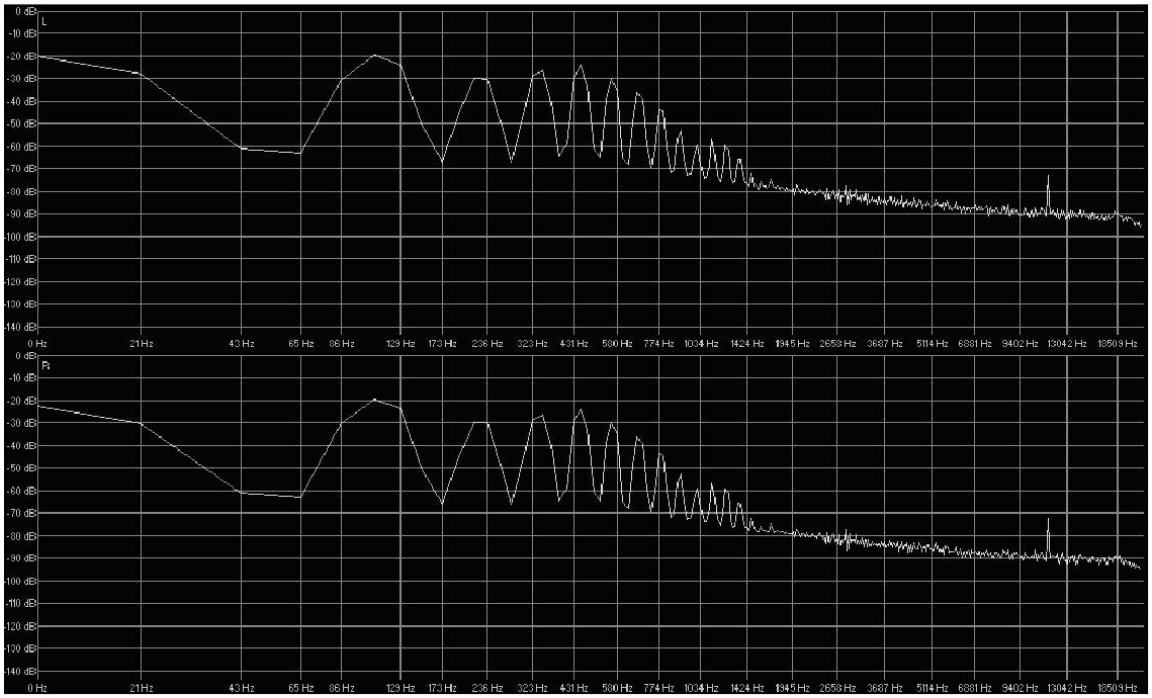


Imagen 6. Gráfica Frecuencia-Decibelios de una cuerda de guitarra afinada en La

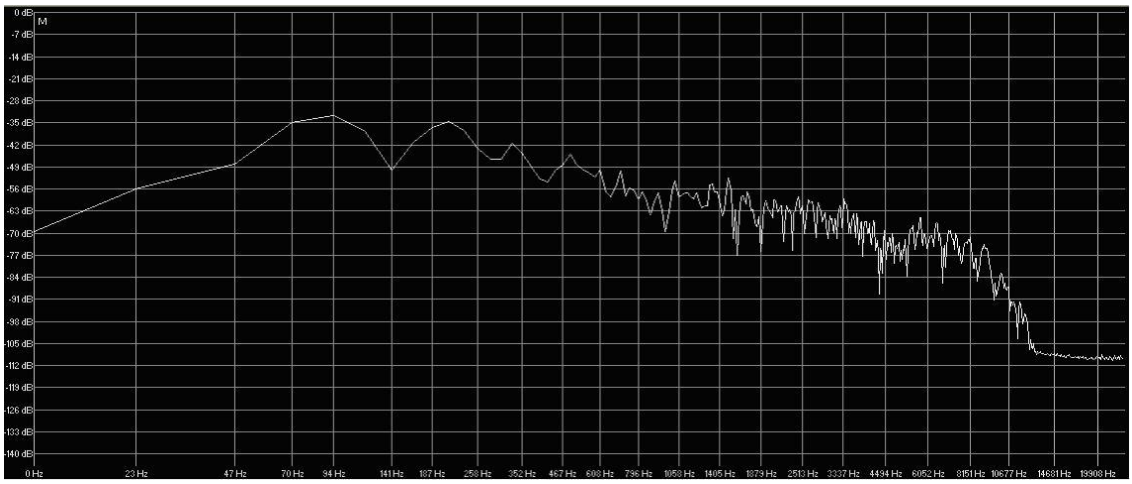


Imagen 7. Gráfica Frecuencia-Decibelios de un timbal base de 16 pulgadas

En la muestra de cuerda fija se observa un caso de espectro armónico, es decir en los cuales las frecuencias presentes eran múltiplos de cierta frecuencia, denominada frecuencia fundamental, dando así un sonido periódico donde puede representarse como la suma de una serie de armónicos. En el caso del sonido por el timbal base vemos que el sonido no es periódico, a pesar de lo cual también pueden representarse gráficamente en un oscilograma, a este sonido se lo denomina sonido parcial. Sin embargo, no podrá identificarse una frecuencia ni un periodo. El espectro correspondiente a estos sonidos se denomina espectro inarmónico.

Después de este ejemplo podemos concluir que la batería no es un instrumento de afinación definida. Esto se acentúa cuando encontramos un timbal o tambor con dos parches porque esto hace que ambas membranas oscilen de manera inarmónica porque la vibración del aire dentro del cuerpo produce alteraciones constantes en el modo de vibración de los parches, sus ondas y el pache, que se mueve de manera irregular sin poder generar una nota definida (fundamental) y sus armónicos. Aunque hay que aclarar que el uso de dos parches no es malo, ya que esto permite que el sonido tenga más sustain (resuena más tiempo). Si en cambio se elimina el parche inferior (parche resonante, el superior se denomina batidor) el sonido del timbal será más seco.

2.1.3 Partes de una batería

Como ya se ha especificado anteriormente, la batería nace de la unión de diversos instrumentos de percusión que se adaptan y montan para poder ser tocados por un único interprete (llamado batería, al igual que el instrumento). Aunque la intención de cualquier batería consiste en crear patrones rítmicos intentando integrar los elementos de la batería de la forma más homogénea posible y consiguiendo también cierta melodía, es cierto que conviene estudiar los elementos por separado ya que no se puede estudiar de la misma forma la construcción de un plato como la de un bombo o un timbal.

Imagen 8. Batería estándar con todos sus elementos identificados

Viendo los elementos aquí en conjunto, a continuación se explicarán y definirán por separado cada elemento que constituye el instrumento.

2.1.3.1 Bombo

El bombo de la batería (en inglés bass drum) es el elemento que produce el sonido más grave del conjunto. Este se coloca apoyado en el suelo y se golpea con un pedal.

El bombo se compone de un casco que consiste en un cilindro de madera contrachapada, en el que se instala un parche a cada lado de este y un conjunto de tensores denominados torres (en inglés lugs) que permiten apretar un aro contra el parche y tensarlo, lo que nos permite conseguir una gama de sonidos, siempre moviéndonos en sonidos graves).

Imagen 9. Bombo de batería y pedal actual

En la imagen se puede ver el parche trasero o batidor, donde el pedal golpea para producir el sonido, junto al mecanismo de torres y aro para poder tensar y afinar el parche. En la parte delantera se usa el mismo mecanismo.

El máximo desarrollo de la batería como instrumento vino cuando William F. Ludwig desarrolló su pedal, rompiendo con los cánones establecidos hasta la fecha. Este invento junto al desarrollo de sistemas para montar los toms en el mismo bombo supuso el comienzo del desarrollo del instrumento hasta llegar a nuestros días.

A partir de la década de 1930 se llevó finalmente a cabo una importante reducción en el tamaño de los bombos, así como la introducción de herrajes más ligeros. Aunque no se habían realizado grandes cambios en el diseño del instrumento exceptuando la llegada de la doble tensión en lugar de la tensión simple o por torres, el diseño personalizado de parches pintados a mano (usualmente con referencias a la marca del instrumento, el grupo o incluso el nombre del artista) había cobrado por otra parte cierta importancia. Esto servía como medio para atraer atención sobre el instrumento, que por entonces cumplía básicamente con la función de marcar y mantener el tiempo con figuras sencillas como negras regulares.

En 1940, la firma Gretsch empezó a construir modelos de baterías con doble bombo, gracias a la idea de Louie Bellson, aunque el rol del segundo bombo era meramente el de dar una imagen más impresionante, ya que en los géneros que se tocaban entonces no se sacaba partido a este elemento, cosa que cambiaría posteriormente con el Hard Rock, Metal, etc. Justo en esa década que empezaba, con la era del Bebop, el rol del bombo en la batería evolucionó para pasar a ser un elemento interactivo, marcando ciertos acentos junto con la caja. Los anteriores bombos de 28" habían pasado a ser sustituidos por modelos más pequeños con diámetros de 20" e incluso 18". Hasta el momento, los cascos se construían de una sola pieza de madera, y Gretsch introdujo otro cambio importante en el diseño de las baterías: el uso de contrachapado.

Más tarde, desde la aparición del Rock a mediados de la década de 1950 empezaron a usarse otra vez bombos más grandes, de 22" y 24", pero siendo esta vez apagados o amortiguados, para sacar máximo rendimiento al ataque. En 1965, Joseph Peters registró una patente de un pedal con el que buscaba poder utilizar dos mazas diferentes en un mismo bombo. Este elemento ha acabado integrándose de forma armónica en el ecosistema de la batería y en ciertos géneros se hace imprescindible. De hecho es más normal ver una batería con doble pedal que con dos bombos debido a su facilidad de transporte, a la necesidad de afinar un único bombo y al menor coste.

Imagen 10. Doble pedal DW 9000

A partir de la década de 1980, si bien los parches seguían estando amortiguados, se comenzó a admitir la resonancia, y se fue afianzando un diámetro estándar de 22". Modernamente también es habitual el uso de modelos de 18" —principalmente en entornos acústicos y jazz—, o de 20".

Imagen 11. Remo Bass Drum Muffling System, producto comercial para apagar el sonido del bombo

2.1.3.2 Caja

La caja (en inglés snare y en partes de Sudamérica también conocida como redoblante, redo o tarola) puede llegar a constituir un instrumento en sí mismo (de hecho se usa en bandas de música y orquestas sin necesidad de ir unido a una batería. Se encuadra en los instrumentos de percusión de sonido indeterminado, dentro de la familia de los membráfonos.

La caja está compuesta por un casco (al igual que el bombo) que puede ser de madera o metal. Esta es una diferencia sustancial ya que no se construyen bombos o toms de metal. Es obvio que cada madera proporcionará un timbre distinto al instrumento, al igual que cada metal. Los materiales más comunes son arce, abedul, roble, latón, bronce y acero.

El sistema de tensión de parches es igual al de un bombo, con la diferencia que los aros usados son de metal y con una forma característica que permite golpear el aro sin dañar la baqueta.

Respecto a los tamaños, aunque existe una gran variedad de tamaños (desde 12 pulgadas a 16 o incluso mayores) el estándar es de 14 pulgadas. Puesto que la afinación de una caja es notablemente más alta es normal que se instalen de 8 a 12 torres en el casco, cuando en toms lo normal suele ser de 6 a 8 dependiendo del tamaño. Es muy común disponer de una caja principal y otra secundaría (normalmente de tamaño más pequeño y por tanto más aguda) para poder tener cierta versatilidad).

Imagen 12. Caja Ludwig Supraphonic, posiblemente la caja más emblemática

La mayor diferencia entre una caja y el resto de toms es la existencia de una bordonera. Esto consiste en un conjunto de hilos (conocidos como bordones) que recorren el casco de lado a lado. Estos vibrarán conjuntamente con el parche inferior al golpear el superior, lo que proporciona al instrumento un timbre muy particular y reconocible con cierto toque de zumbido. Los bordones se suelen fabricar en alambre rizado, cable metálico o nylon. Además existe una gran variedad de bordoneros, con diferentes números de filas, longitud, etc.

Es conveniente mencionar que estos bordones pueden anularse debido a un mecanismo que se incluye en el 99% de las cajas del mercado, por lo que puede conseguirse el sonido natural de caja o el sonido que proporcionaría la caja sin los bordones (parecido al de un tom pero algo más agudo debido a la tensión del parche y a que suelen ser más cortas que un tom). Este mecanismo suele ser una palanca que si está subida mantiene los bordones contra el parche inferior, haciendo que suenen, pero si la palanca se baja estos se destensan y desaparece el timbre.

Imagen 13. Sistema de tensado de los bordones de una caja DW

También existe otro sistema mucho más complejo desarrollado por Ludwig llamado “Supersensitivo”, el cuál fue introducido en el famoso modelo Ludwig Suphrasonic. El sistema de bordonera supersensitiva consiste en un mecanismo que sube y baja la bordonera para apretarla contra el parche superior o no. La diferencia con el sistema común es que, mientras el sistema común solo la destensa, el sistema supersensitivo la aleja completamente del parche, por lo que se pueden ajustar muchos más parámetros como la distancia del bordonero al parche.

Imagen 14. Detalle del sistema de bordonero Supersensitivo desarrollado por Ludwig

Como ya se ha dicho antes, el sonido de la caja viene determinado principalmente por el parche y el diámetro y altura del casco, aunque existen otros factores importantes como la tensión de afinación, el ángulo de los aros y

el casco, la tensión de la bordonera y el material del mismo casco. Es común usar sordinas o materiales atenuantes como cinta adhesiva o los famosos MoonGel para intentar anular ciertos armónicos indeseados que aparecen al afinar el parche. También es muy común que el parche usado en la caja, tanto superior como inferior, sea de diferente tecnología y composición al de los toms o el bombo.

Dentro de la batería, la caja tiene una función fundamental, ya que es la encargada de marcar el ritmo junto con el bombo. Generalmente, en géneros como el rock, blues y los derivados de estos, el bombo suele aparecer en el primer y tercer tiempo del compás, mientras que la caja lo hace en el segundo y el cuarto. Esta norma no escrita cambia totalmente cuando hablamos de jazz o músicas latinas.

Además es la parte de la batería donde más tipo de golpes pueden darse para conseguir diferentes sonidos dentro del mismo instrumento.

2.1.3.3 Tom

El tom o timbal es un instrumento musical de percusión originario de las culturas asiáticas que forma parte de la batería desde comienzos del siglo XX. Es una parte que, aunque a veces se usa en la creación de ritmos, es más común utilizarse en “fills”, que son pequeños fraseos que se usan para separar dos ritmos diferentes y que el cambio de uno a otro no sea tan abrupto, además de aportar más frescura al ritmo y romper la monotonía.

El casco de un tom es muy similar al de una caja o un bombo, cambiando únicamente en las medias, pero manteniéndose el sistema de aros y tensión de parches de una caja. Este armazón suele consistir en 6 u 8 capas de madera, la cuál puede ser de distintas variedades como caoba, abedul, arce o roble. Esto influye significativamente en el sonido de los timbales, afectando principalmente al tono y al sustain.

Respecto a las medidas de los timbales, podemos encontrar una variedad enorme. Aunque las medidas estándar de los toms de batería serían de 12, 13 y 16 pulgadas, podemos encontrar toms desde las 8 a las 20 pulgadas con relativa facilidad en el mercado.

Los sistemas de montaje de los toms varían principalmente dependiendo de su tamaño. Generalmente hasta las 14 o 15 pulgadas de diámetro se suelen montar con soportes especiales que agujerean el casco del tom, y unidos al bombo mediante un brazo metálico, quedando así anclados al bombo.

Imagen 15. Sistema estándar de montaje de toms sobre el bombo

Aunque en los últimos años las marcas han invertido muchos recursos en mejorar los diseños y el sonido de sus cascos. Es por esto que en las baterías de gama alta generalmente se opta por sistemas de montaje que no agujereen ninguno de los cascos, ya que así no se reduce el sustain del tono.

Imagen 16. Batería DW con sistema de sujeción de cascos sin agujeros extras

En los toms que tiene un diámetro superior a 15 pulgadas (a veces estos inclusive) se suelen montar con un sistema de patas, las cuáles se apoyan en el suelo. Esto se puede ver en la imagen anterior (Imagen 16). A estos toms se les suele llamar toms base, timbales base, goliats o chanchas (sobre todo en Sudamérica).

2.1.3.4 Parches

Un parche, en inglés drumhead, es una membrana tensada sobre uno o ambos extremos abiertos de un casco (caja, bombo o tom). Originalmente los parches eran confeccionados con cuero de vacuno, pero era muy difícil mantenerlos afinados, así como muy pobre su resistencia a los efectos climáticos (humedad y temperatura).

En 1956, Chick Evans inventó el parche de plástico, recurriendo a un derivado del poliéster llamado Mylar, creado por DuPont. Las ventajas del parche de plástico fueron evidentes desde un principio: menor costo, tonalidad relativamente estable, menores efectos ambientales y mayor durabilidad.

Luego de la aparición del Mylar, DuPont produjo una nueva fibra sintética derivada de la aramida, el Kevlar, con el que también se comenzaron a elaborar parches. Pese a su gran resistencia e incomparable durabilidad, generaban muchas afecciones en las manos de los bateristas (por rebote excesivo) como tendinitis y síndrome del tunel carpiano; por otro lado, ofrecían muy poca resonancia y un estrecho rango tonal.

Es así que progresivamente fue restringiéndose la utilidad del Kevlar a dos situaciones específicas: la confección de protectores contra impacto (contra la maza del pedal del bombo) y la elaboración de parches para tambores de marcha. Hoy en día el Mylar sigue siendo el material preferido para la confección de parches. Entre los más prominentes fabricantes de parches a nivel mundial se incluyen empresas como Aquarian, Evans y Remo.

La estructura de un parche a día de hoy consiste en un aro de metal en forma de U donde se introduce la membrana del parche. Posteriormente este aro se sella con pegamento para evitar que la membrana se salga al apretar el parche.

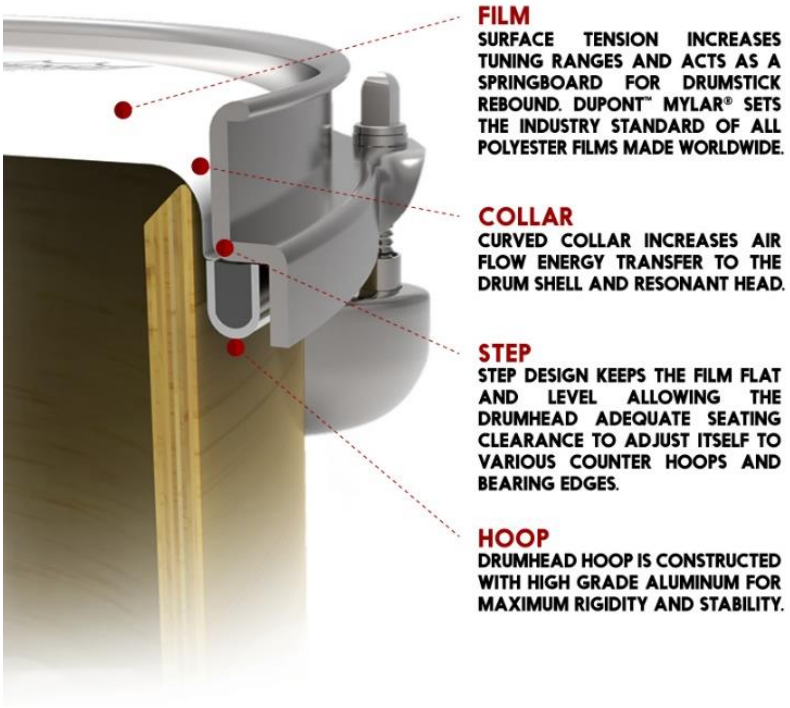


Imagen 17. Anatomía de un parche

En esta imagen se puede ver la sección del parche, la membrana y como el sistema de aro y torres tensa el parche para conseguir el tonto deseado.

En la actualidad existen muchos tipos de parches en relación a su construcción, a continuación se intenta hacer una breve síntesis de los diferentes modelos:

- Parches lisos: Producen el máximo volumen, sustain, y rango armónico, las variables se producen con elementos como sordinas (muffling), orificios de ventilación, sordinas tipo aro, y cualquier otra forma para eliminar armónicos, (sustain y la definición del ataque, son inversamente proporcionales.)

- Parches arenados: La razón más obvia en esta elección es de tocar con escobillas (brushes) aunque esa cualidad de aspereza también aparece con las baquetas, pero en menor medida. Además aparece un sonido muy particular cuando se lo toca cerca del aro que se resalta cuando se usa un micrófono. Las cualidades tonales y su durabilidad, varían de acuerdo a los fabricantes.

- Parches reforzados: Un circulo de Mylar o un compuesto laminado logran dar mayor durabilidad, menos vibración, y acentúa el rango de sonidos medios. La articulación, diámetro y espesor varían de acuerdo al modelo y al fabricante.

- Parches tipo Pinstripe: Contrariamente al reforzado, acentúa el registro y amplia la definición del ataque. Remo fue pionero en esto con su famoso modelo Pinstripe, el cual posee una fina línea de epoxy entre sus dos placas de 7/100 pulgadas de grosor. Posteriormente el resto de marcas sacaron modelos equivalentes. .

- Parches laminados (mylar): Tipo Fiberskyn II (modelo representativo de Remo en esta clase). Sonido cálido, oscuro, predominando las medias y bajas frecuencias. Usados generalmente en Jazz y música latina.

Imagen 18. Parche Remo Fiberskyn II

- Parches con orificios de ventilación: "Evans Exclusive," sonido de tambor y bombo seco, con pequeños orificios de ¾ de pulgada. Elimina las vibraciones, reduce al mínimo los armónicos agudos, al margen de la respuesta física o las características de la batería.

Imagen 19. Parche Evans Genera Dry 1, ejemplo de parche con orificios de ventilación

- Parches Dobles: Consiste en el uso de dos membranas simples en un mismo parche. Se consigue mayor durabilidad y pocos armónicos, su superficie vibra en forma lenta y es por eso que suena grave en comparación con los simples a una misma tensión, el rebote de la baqueta es sutilmente menor.

- Parches Hidráulicos (solamente Evans): Disponen de un aceite entre las dos capas, que inhibe la vibración. Siendo hidráulicos irónicamente dan sonido seco, el tono es sacrificado en pos de un ataque definido, el rebote del palillo es lento, comparado con los otros.

Imagen 20. Evans Hydraulic, parche de aceite o hidráulico

- Parches de máxima durabilidad: Fabricados técnicamente con láminas de Maylar, y Kevlar (fibra de compuesto plástico) que se utiliza en los chalecos antibalas!, reduce los armónicos y el rebote del palillo, máxima durabilidad, estos varían con respecto al grosor y la rigidez de la lámina, no son recomendables para escobillas (brushes).

2.1.3.5 Platos

Los platos, platillos o címbales (del inglés cymbals) son instrumentos de percusión de sonido indeterminado, lo que, como ya se ha explicado, significa que las notas que proporcionan no tienen una altura determinada. Se calsifican en la familia de los idiófonos ya que el sonido se produce entrechocando uno con otro o golpeándolos con baquetas.

Un plato consiste en un disco cóncavo de metal, construidos en distintas aleaciones dependiendo de la calidad y el coste del plato, con un agujero central que, en el caso de la batería, permite montarlo en un soporte para poder estar suspendido. Las aleaciones que se suelen usar son:

- Latón: Combinación de cobre y Zinc
- Bronce: Combinación de cobre y estaño
- Nickelsilver: Combinación de niquel y plata (esta es muy poco común)

El latón es comúnmente usado para producir platos de series bajas, mientras que el bronce se utiliza en series de mayores calidades (intermedias y altas). Cada aleación puede mostrar proporciones variables de sus componentes, pero normalmente es la proporción de cobre la que marca el precio. Es también conveniente aclarar que existen series de gran calidad que no contienen porcentajes tan altos de cobre. Es el caso de la serie 2002 de Paiste, una de las

más emblemáticas y más usadas en la historia del Rock (serie utilizada por grandes baterías como John Bonham de Led Zeppelin e Ian Paice de Deep Purple.

La mayor proporción de cobre produce tonos más graves y profundos a igualdad de masa, mientras que el agregado de estaño, hierro o zinc produce tonos más agudos. Para obtener una estructura más cristalina y producir variantes en la sonoridad las fábricas han introducido otros metales como antimonio o bismuto en pequeñas cantidades.

Dentro de los platos existe una enorme variedad de tipos, tamaños y modelos de cada marca dentro de cada tipo en especial. A continuación se intentará hacer un pequeño resumen:

Imagen 21. Kit completo de una batería con los diferentes platos etiquetados

- Crash (etiquetado en rojo): es un plato mediano cuyo tamaño suele oscilar entre 13 y 18 pulgadas, aunque existen tamaños mayores (Ian Paice usa un crash de 24). Se usa para acentuar ciertos puntos de compases, para finalizar fills o fraseos entre ritmo y ritmo y a veces como sustitutivo del charles en ciertos ritmos para dar mayor énfasis. Existen variantes de estos platos como los O-Zone de Sabian o los Generation X de Meinl, los cuáles están perforados por todo el cuerpo y producen un sonido que abre muy rápido pero a la vez es muy seco y corto.

Imagen 22. Crash Sabian AAX Ozone de 16"

- Ride (etiquetado en lila): es un plato grande que varía entre las 17 y las 26 pulgadas generalmente, aunque pueden existir más grandes. Se usa para llevar el ritmo, especialmente en el Swing y el Jazz, mientras que en el Rock y Blues se usar en combinación con el Charles para el mismo propósito. La diferencia es que en el jazz los ride suelen ser más cálidos, de sonido algo sucio y abren menos, mientras que en el rock suelen ser fuertes, potentes y limpios. Existen también los llamados Crash-Ride que es un punto intermedio entre estos dos tipos, donde se puede llevar el ritmo y a la vez usarlo como crash (un ride normal no se puede usar así ya que genera mucho estruendo y tarda mucho en apagarse). Otra variación común en el jazz es el uso de rides con remaches o tachuelas, lo que ensucia mucho más el sonido.

Imagen 23. Ride Zildjian con tres remaches

- Splash (etiquetado en amarillo): es un plato pequeño cuyo tamaño oscila entre las 6 y las 12 pulgadas. Se usa para pequeños cortes entre ritmos, comúnmente en pasajes de poca intensidad sonora, o también en combinación con otros platos al hacer ciertos fills (pequeña frase que se realiza entre ritmos o para pasar de un ritmo a otro). Son platos de sonido más corto que los anteriores y mucho más brillante, además de mucho más frágiles.

- China (etiquetado en negro): se fabrican en todo tipo de medidas, aunque los que son menores de 13 son conocidos como Splash-China. La forma es totalmente diferente a la del resto de platos al igual que el sonido. Es un sonido que recuerda al de un gong, pero mucho más rápido, estridente y con mucho ataque. Se puede usar igual que un crash, incluso en ciertos géneros como el metal se llega a usar como ride. Se suele montar justo al revés que el resto de platos. Existen también modelos con orificios como en los crash.

Imagen 24. China Paiste PST8

Falta por describir el charles o Hihiat, pero este merece un capítulo aparte que se encuentra a continuación.

Como dato extra, los principales productores de platos en el mercado son Paiste, Sabian, Zildjian y Meinl. Son las marcas más grandes y las que más invierten en avances, producción y patrocinios de artistas.

2.1.3.6 Hihat o charles

El hihat o charles es una de las piezas básicas de la batería. El funcionamiento consiste en dos platillos montados sobre un trípode. Existe un pedal en la base que acciona una varilla que sube o baja de forma gradual en función de si se pulsa el pedal más o menos. Mientras el plato inferior (bottom) descansa libre en el soporte, el superior (top) se encuentra enganchado a la varilla mediante un tornillo, por lo que cuando se pisa el pedal estos dos platos chocan, y cuando se levanta se separan. Tanto la altura del plato superior como la tensión del pedal o la altura del plato inferior son ajustables.

Imagen 26. Charles montado en su soporte específico

El charles es de vital importancia ya que es donde se lleva el ritmo junto al bombo y la caja golpeando en el plato superior que lo compone. Esto se puede ver en la siguiente imagen, que consiste en el ritmo básico de la batería. Las notas con cruces corresponden al charles, las que se encuentran entre la primera y segunda línea corresponden al bombo (fa) y las restantes a la caja (do).

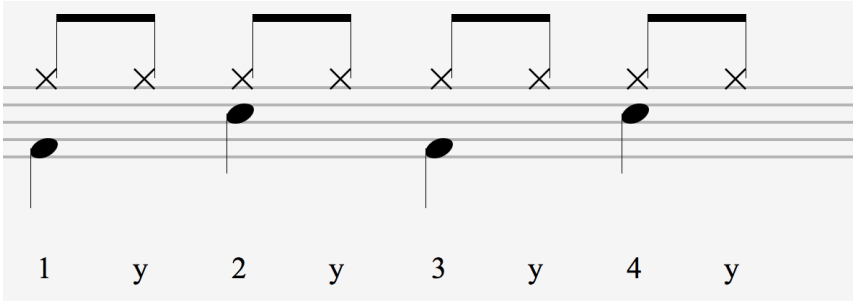


Imagen 25. Partitura del ritmo básico en la batería

Aquí se puede ver porque se dice que el ritmo se marca con estos tres elementos. Aunque el charles ofrece una gran cantidad de sonidos en función de cuanto se abra, si se abre rápido o despacio, si se pisa fuerte o flojo, etc. Por esto mismo es importante destacar su funcionamiento.

Respecto a los platos que se usan, son especiales para charles, normalmente de 14 pulgadas, aunque existen de mayor y menor tamaño como en el resto de platos. El sonido del charles es muy particular, ya que son dos platos a la vez los que suenan a la vez que se rozan. Hay que mencionar que estos platos tienen posición, hay uno superior y otro inferior especialmente determinados. Esto es importante ya que hay modelos en los que el plato inferior es ondulado, por lo que el sonido que se consigue es algo distinto a los normales.

Imagen 27. Ejemplo de charles ondulado, Paiste 2002 Edge 15"

2.1.4 Tipos de sonido y golpes

Aunque en general se piensa que a la hora de tocar un instrumento de percusión como la batería simplemente se trata de golpear la parte que corresponda y que esta suene como sea, esto no es así en absoluto. Existe una amplia variedad de tipos de sonidos y de golpes que se usan normalmente y que conviene conocer antes del desarrollo del proyecto para entender porque se han tomado ciertas decisiones en este.

2.1.4.1 Golpes y sonidos en la caja

Aunque en un instrumento de percusión la cantidad de sonidos puede ser prácticamente infinita, vamos a reducirnos a los sonidos que han llegado a convertirse en estándar y que cualquier batería sabe o debe saber usar.

2.1.4.1.1 Redoble

Un redoble consiste en sustituir los golpes normales, en los que la caja se golpea simplemente, por golpes en los que se deja rebotar la baqueta dos veces en cada golpe. Si estos golpes se dan de forma correcta, suficientemente rápido y con ambas manos se llega a crear lo que se llama redoble, que, en vez de sonar como un montón de golpes amontonados, da una sensación de un zumbido continuo. Esta técnica se puede usar durante el tiempo que se desee, por lo que se suelen tomar los mismos equivalentes que en golpes sencillos, por ejemplo, se puede dar un golpe como una negra, o un redoble como una negra. La diferencia será que en el golpe normal no se dará otro hasta que pase un tiempo, mientras que el redoble será continuo durante un tiempo.

En la siguiente forma de onda, que dura 10 segundos, se puede ver cómo, en lugar de aparecer golpes sueltos, aparece una sucesión de golpes muy seguida, que es la responsable del sonido característico de un redoble.

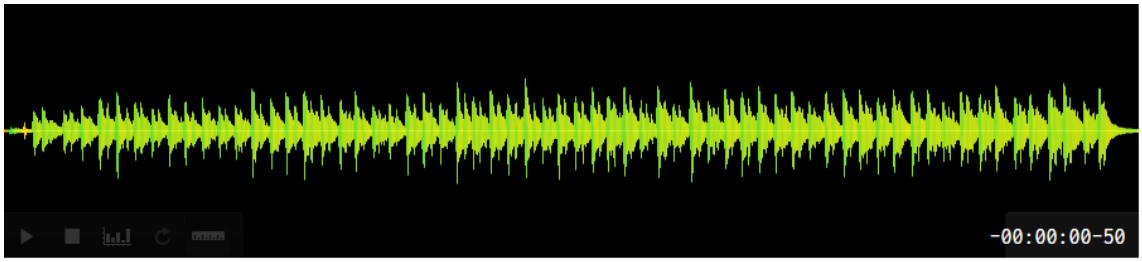


Imagen 28. Forma de onda que resulta de grabar un redoble con un micrófono

2.1.4.1.2 Rimshot o golpe de aro

Un golpe de aro (rimshot en inglés), consiste en golpear a la misma vez el aro de la caja y el parche. Es una técnica muy usada y extendida, y que una vez dominada da mucho juego tanto en ritmos como en fills y redobles. El sonido que se consigue es muy distinto al sonido normal de una caja. De esta forma aparecen muchos más armónicos y cierto sonido más metálico, aunque la caja sea de madera. En la siguiente imagen se puede ser bien el golpe.

Imagen 29. Captura de la realización de un golpe de aro

2.1.4.1.3 Cross

El cross es un golpeo que consiste apoyar una punta de la baqueta en el parche y golpear con la otra en el aro, mientras la otra punta pivota sobre el propio parche. Se usa en sustitución de la caja en partes donde no existe tanta emoción o expresividad, aunque también es cierto que es muy usado en entornos de jazz o música latina. El sonido que se consigue es parecido al de una caja china, un

sonido muy parecido al de golpear una madera, pero con algo más de resonancia y empaque.

Imagen 30. Captura de la realización de un cross

2.1.4.2 Golpes y sonidos en los platos

A la hora de golpear los platos hay que tener cierto control y técnica, ya que aunque parezca un simple golpe, siempre hay que acomodar el volumen del instrumento al lugar donde estemos tocando, por lo que no es tan parecido como parece. Además en los platos se distinguen normalmente tres tipos de golpeo.

- Golpe en el borde o crasheo: se denomina así porque es como se tocan normalmente los crash. El golpe se produce en el borde del plato, ya sea con la punta de la baqueta o con el cuerpo (con el cuerpo es más fácil). Con esto conseguimos que el sonido del plato sea muy abierto, aunque con muy poca definición. Este golpeo se suele usar en crash, spalsh y chinas. No tanto en rides ya que el sonido es muy alborotado (no están pensados para eso).

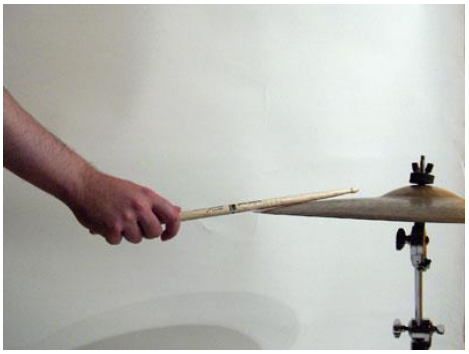


Imagen 31. Captura de un golpe en el borde del plato

- Golpe en el cuerpo del plato: en este caso el plato se golpea en una zona intermedia de su cuerpo con la punta de la baqueta. De esta forma el sonido es más definido y el plato se abre muy poco. Se usa para llevar el ritmo sobre todo en jazz o en ciertas partes del rock. Este golpeo se suele usar en los rides.

Imagen 32. Captura de un golpe en el cuerpo del plato

- Golpe en la campana del plato: en este último caso el golpe se puede dar con la punta o el cuerpo de la baqueta, pero el lugar es la campana del plato. La campana es la parte central del plato, en la cual el radio de curvatura es menor. Al golpear ahí conseguimos el sonido más definido posible, el plato no se abre nada y el tono es muy distinto al del cuerpo. Se usa cuando se quiere enfatizar en algunas notas de un ritmo de ride.

Imagen 33. Captura de un golpe en la campana

Estos son los tres golpes más comunes en los platos, aunque siempre se puede innovar o probar cosas distintas. Aquí se adjunta una imagen que resumen muy buen este tema tratado.

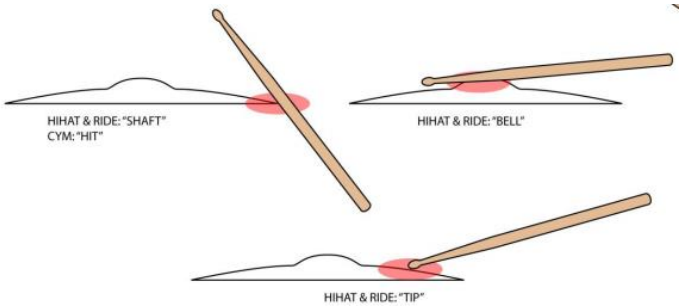


Imagen 34. Resumen de los tipos de golpeo en un plato

2.1.4.3 Golpes y sonidos en el charles

El charles nos ofrece, gracias a la combinación del golpeo al plato y la distinta apertura entre los dos platos que lo componen, una amplísima gama de sonidos. Los tipos de golpeo con las baquetas son exactamente los mismos que los de un plato normal, aunque es cierto que no es muy común golpear la campana de un charles, aunque puede hacerse perfectamente.

El tema que se tratará aquí es el de los sonidos que podemos sacar en función de la apertura y la presión que hagamos en el pedal. Lo normal es tocar el charles cerrado totalmente (con ambos platos tocándose) pero sin ejercer mucha presión en el pie. Así conseguimos el sonido característico corto y agudo.

Si levantamos un poco el pie, hasta el punto en el que los platos sigan en contacto pero vemos que al golpearlos se mueve, este es otro punto muy usado, ya que el sonido es más estridente y aparece el sonido del roce entre los dos platos. Esto se usa mucho en partes de canciones que van a llegar a un punto fuerte de énfasis, justo antes de llegar a algún fill o corte.

Otro tipo sería el abrir el charles en alguna nota determinada del ritmo y cerrarlo a continuación. De esta forma el ritmo no es tan monótono. Aquí podemos jugar con abrirlo un poco y sacar el sonido antes comentado o abrirlo más hasta que los platos dejen de estar en contacto. De este modo conseguiremos que solo suene el plato superior, con su propio sonido.

Lo último sería comentar que los propios baterías marcan el ritmo con el pie en el pedal, abriéndolo y cerrándolo normalmente en cada tiempo del compás. Si se sabe hacer no se producen sonidos que molesten al ritmo, ya que se pisa rápido y fuerte, y además el sonido que se crea es muy breve y sirve al resto de integrantes del grupo para poder llevar el compás.

2.1.5 Interfaces y sensorización

A continuación se tratarán con un propósito general los sensores piezoeléctricos, los cuáles han sido los escogidos para llevar a cabo el registro de golpes en este proyecto, y el protocolo MIDI, el cuál es el usado para el envío de información hacia el ordenador.

2.1.5.1 Sensor piezoeléctrico

Un sensor piezoeléctrico es un dispositivo que utiliza el efecto piezoeléctrico para medir presión, aceleración, tensión o fuerza, transformando las lecturas de estas magnitudes en señales eléctricas.

Estos sensores se consideran herramientas versátiles para la medición de distintos procesos. Aunque el efecto piezoeléctrico fue descubierto por Pierre Curie en 1880, no comenzó a ser implementado en sensores hasta 1950. Desde entonces el uso de este principio de medición se ha incrementado debido a su fácil manejo y su alto nivel de fiabilidad. Tiene aplicaciones en campos como la medicina, la industria aeroespacial y la instrumentación nuclear, así como en pantallas táctiles de teléfonos móviles. En la industria automovilística, los elementos piezoeléctricos se utilizan para monitorear la combustión durante el desarrollo de motores de combustión interna, bien montados directamente en hoyos adicionales en la culata o en las bujías, que están equipadas con un sensor piezoeléctrico en miniatura.

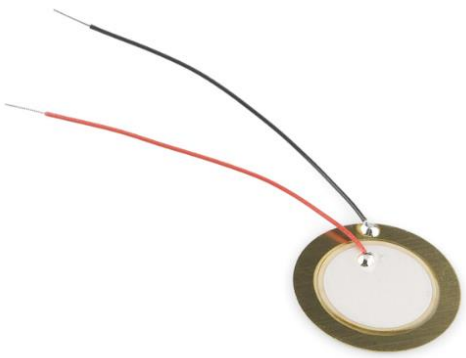


Imagen 35. Ejemplo de sensor piezoeléctrico

A pesar de que los sensores piezoeléctricos son sistemas electromecánicos que reaccionan a la compresión, los elementos sensoriales muestran una deflexión casi nula. A ello se debe la alta precisión de estos sensores, ya tienen una frecuencia natural muy alta y una buena linealidad en amplio rango. Además, la tecnología piezoeléctrica es insensible a campos electromagnéticos

y a la radiación. Algunos materiales usados (como el fosfato de galio o la turmalina), poseen un alto grado de sensibilidad incluso al ser expuestos a altas temperaturas, permitiendo que el sensor sea eficiente a temperaturas del orden de 1000 °C. La turmalina también posee piroelectricidad, por lo que se genera una señal eléctrica cuando la temperatura del cristal es alterada. Este efecto es muy común en materiales piezocerámicos.

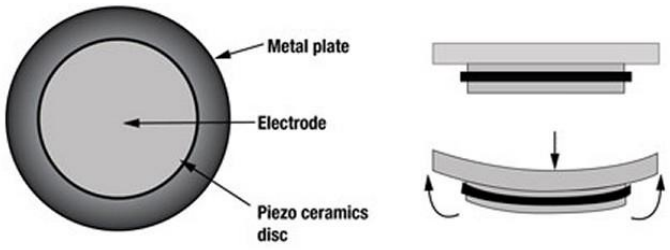


Imagen 36. Partes de un sensor piezoeléctrico y ejemplo de flexión de uno

Una desventaja de los sensores piezoeléctricos es que no se pueden utilizar para mediciones de estática, ya que una fuerza estática resultaría en una cantidad fija de cargas sobre el material piezoeléctrico. Al trabajar con dispositivos de visualización convencionales y materiales aislantes imperfectos, así como por la reducción de la resistencia interna del sensor, resulta poco eficiente debido a la pérdida constante de electrones y el bajo rendimiento de la señal. Además, las temperaturas elevadas causan una falla adicional en la resistencia interna y en la sensibilidad de la medición. La principal consecuencia del efecto piezoeléctrico es que cuando aumenta significativamente la presión y la temperatura la sensibilidad se reduce debido al llamado montaje gemelo (twin-formation).

2.1.5.2 MIDI

La aparición de los sintetizadores digitales a finales de la década de los 70 trajo consigo la problemática de la compatibilidad en la interconexión entre los dispositivos que cada fabricante sacaba al mercado, algo que no era problema con los sintetizadores analógicos anteriores.

En ese momento era evidente que era necesaria una forma que les permitiera a los músicos y usuarios de estos tipos de sistemas poder utilizar todo el potencial de sus aparatos sin restricciones de ningún tipo, y es por ello que nació MIDI, un protocolo común que todos los fabricantes deberían seguir para evitar incompatibilidades y hacer más sencilla la utilización y programación de sus aparatos. En este artículo conoceremos algunos detalles de su historia y alcances técnicos.

MIDI fue desarrollado por Dave Smith, ingeniero electrónico, quien en 1981 presentó el primer borrador de la especificación MIDI, por aquel entonces

llamado “interfaz universal de sintetizadores (USI)”, el cual tras revisiones y mejores se convirtió finalmente en MIDI.

MIDI es la abreviatura de “Musical Instruments Digital Interface”, la cual básicamente es un protocolo de comunicación de datos que les permite a PC, sean Windows, Mac, Linux o cualquier otra plataforma, sintetizadores, controladores y muchos otros dispositivos e instrumentos desarrollados para la creación musical comunicarse sin problemas y en forma estándar.

Antes que nada, es necesario destacar que MIDI no maneja datos analógicos, es decir que no envía ni recibe señales de audio, sino que es un protocolo digital, lo que significa que se envían y reciben datos, los que pueden ser mensajes de control de eventos, los que serán interpretados por el dispositivo que los reciba de la forma en que fue programado.

Para ello se vale de 127 valores numéricos, los que le brindan la suficiente flexibilidad para incluir toda la información necesaria, es decir nota musical, duración y otras características. Todos estos datos serán procesador por el aparato que los recibió para ser convertidos en audio real, el cual además, luego se puede procesar.

Otra de las ventajas del protocolo MIDI es que permite componer, editar y grabar música directamente en el formato, lo que ofrece como resultado archivos increíblemente pequeños para la magnitud de los sonidos que contienen, lo que favorece a todos los aspectos de la creación musical.

Físicamente, un instrumento MIDI se conecta a través de un conector del Tipo DIN de cinco pines, pero en la actualidad grandes fabricantes como Roland o Korg están reemplazando lentamente este tipo de conector por el más simple y popular USB, lo que sin dudas garantiza una estandarización y compatibilidad mucho mayor, y que además permiten conectarlos a una computadora de manera mucho más sencilla.

Imagen 37. Conector DIN de 5 pines

El protocolo MIDI es utilizado en la actualidad en todos los ámbitos de la realización musical, desde la creación, la edición y la grabación, es más, muchos de los sonidos generados gracias a una cadena MIDI son utilizados directamente para la mezcla debido a la perfección sonora y de matices que ha alcanzado el protocolo a través de los años. Tal perfección sonora ha permitido a músicos de

todo el mundo crear piezas musicales como música orquestal, muy difíciles y caras de grabar por medios tradicionales.

2.2 Sensorización de baterías con piezoeléctricos

2.2.1 Funcionamiento básico de los sensores piezoeléctricos

Como ya se ha mencionado, el funcionamiento de estos sensores se basa en el efecto piezoeléctrico, el cual consiste en la aparición de una carga eléctrica cual el cristal se somete a una deformación.

Hay que aclarar que la orientación inicial de los dominios en los materiales usados para la construcción de los sensores es indeterminada (en cualquier dirección por el dominio de Weiss), es decir, su comportamiento inicial ante la deformación es nulo. Sin embargo, después de aplicar un proceso de polarización (aplicando tensiones mayores de 2000 V/mm) se consigue la reordenación de estos dominios y, por consiguiente, la consecución del efecto piezoeléctrico. Después de este procedimiento solo resta colocar el material en medio de dos electrodos de material aislante, capaces de producir carga en él.

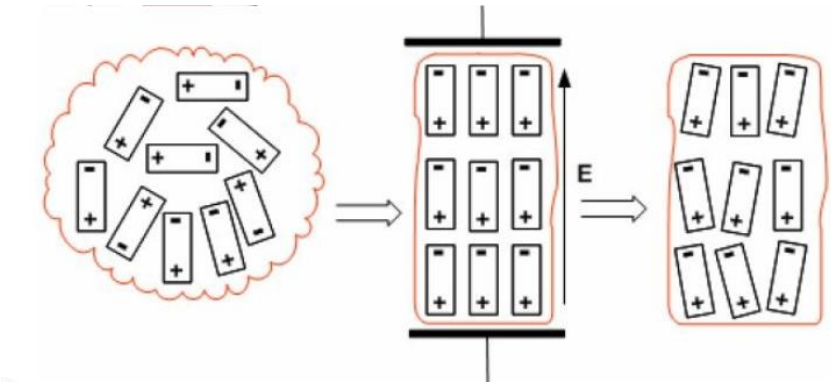


Imagen 38. Gráfico representativo de la polarización de un sensor piezoeléctrico

A la hora de elegir el elegir un sensor piezoeléctrico en concreto se deberá tener en cuenta principalmente la frecuencia de trabajo, ya que esto nos proporciona unos valores máximos y mínimos de frecuencias para el uso de este, lo que determina el tiempo de respuesta en la aplicación que deseemos. Los fabricantes a veces proporcionan el valor de la capacidad piezoeléctrica C_p, la sensibilidad al parámetro que está MIDIendo y la frecuencia de resonancia (ocasionada por la presenta de elementos inductivos o capacitivos en un circuito) o el margen de esta en la que funcionan estos valores.

Existen diferentes tamaños y formas en las que se fabrican los sensores, las cuáles determinan la zona plana sobre la que se debe actuar y el tipo de esfuerzos que va a poder medir.

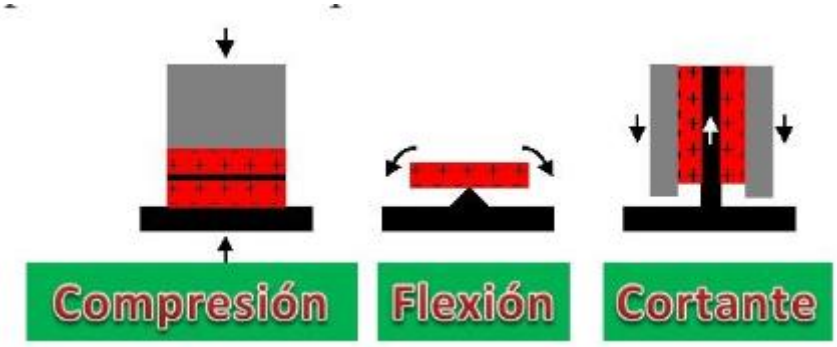


Imagen 39. Composición de un sensor piezoeléctrico en relación al esfuerzo que tiene que medir

La deformación del sensor genera una carga Q cuyo valor es aproximadamente proporcional al acortamiento unitario del espesor del cristal (para deformaciones muy pequeñas), o sea:

$$Q = \frac{K \cdot z}{e}$$

Donde K es una constante que depende del material y de la dirección de la talla, z es el acortamiento unitario y e es el espesor del cristal antes de la deformación.

Un transductor piezoeléctrico tiene una muy alta impedancia de salida de corriente continua y puede ser modelado como una fuente proporcional de voltaje y como una red de filtro.

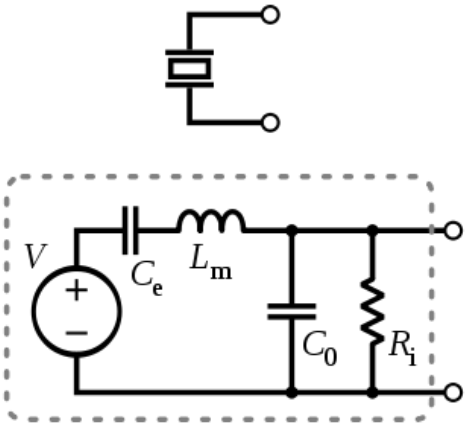


Imagen 40. Modelo equivalente de un sensor piezoeléctrico

El voltaje V de la fuente es directamente proporcional a la fuerza, presión o tensión aplicada. La señal producida está relacionada con esta fuerza mecánica como si hubiera pasado a través del circuito equivalente especificado en la imagen anterior. Un modelo más detallado incluiría los efectos de la construcción mecánica del sensor y otras no ideales. La inductancia Lm es causada gracias a la masa sísmica y la inercia del propio sensor. Ce es inversamente proporcional a la elasticidad mecánica del sensor. C0 representa la capacitancia estática del transductor, la cual es resultado de la inercia de una masa de tamaño infinito. Ri

es la resistencia de la salida del aislamiento del elemento del transductor. Si el sensor está conectado a una resistencia de carga, esto también actúa en paralelo con la resistencia del aislamiento, incrementando la alta frecuencia de corte.

2.2.2 Circuitos de acondicionamiento generales para piezoeléctricos

Puesto que este tipo de sensores se pueden usar para diversas aplicaciones, existen diversos circuitos de acondicionamiento que nos permite adaptar la señal que nos proporciona el sensor para un propósito más específico.

Una de las opciones que podría usarse para adaptar la señal original del sensor a cualquier dispositivo podría ser usar un amplificador de carga.

Un amplificador de carga es un circuito cuya impedancia de entrada es un condensador, ofreciendo así una alta impedancia a baja frecuencia. Su función es ofrecer en la salida, con una impedancia muy baja, una tensión proporcional a la carga de la entrada. Es, por tanto, un convertidor carga-tensión.

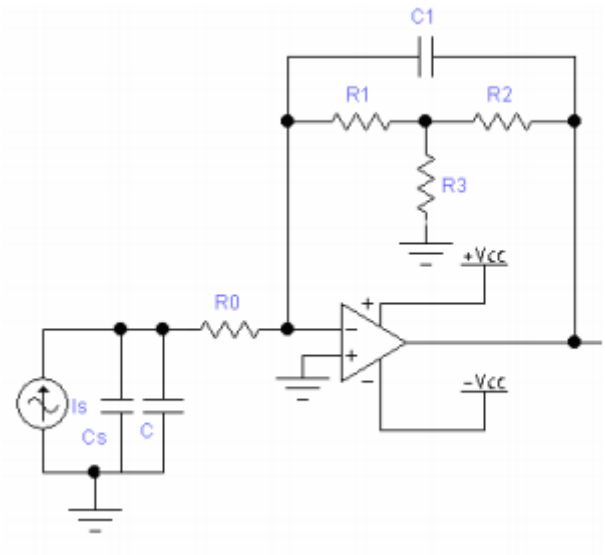


Imagen 41. Ejemplo de ampificador de carga

Considerando A_d la ganancia en lazo abierto del amplificador operacional y C la capacidad del cable, la tensión de salida de esta etapa sería:

$$v_o = \frac{q_s}{C_1 + \frac{C + C_1}{A_d}}$$

Esta expresión se puede simplificar si suponemos que a bajas frecuencias A_d>>1.

$$v_o = \frac{q_s}{C_1}$$

Ahora la sensibilidad no dependerá del cable, aunque esta aproximación será válida solo a bajas frecuencias. Para el resto de frecuencias tendremos una función de transferencia:

$$\frac{V_o(s)}{Q_s(s)} = \frac{-s \cdot R \cdot f \cdot C_1}{1 + s \cdot R \cdot f \cdot C_1}$$

Otra posible opción sería acondicionar la señal con una configuración que use un amplificador electromético. Esta configuración se basa en la medida de corrientes débiles en sensores. La medida de corrientes débiles se puede realizar tomando directamente la caída de tensión en una resistencia de valor elevado (figura (a)) o realizando una conversión corriente-tensión mediante un amplificador de transimpedancia (figura (b)) basado en un AO con características electrométricas.

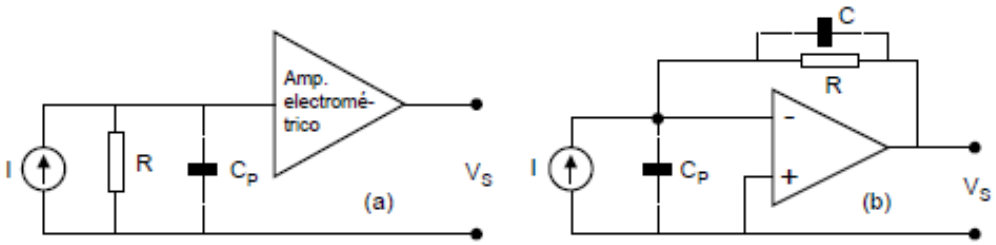


Imagen 42. Figuras a) y b) de ejemplos de amplificadores electrométricos

Con el circuito de la figura (a) no se pueden medir fenómenos dinámicos, pues CP (suma de la capacidad del sensor, la del cable y la de entrada del amplificador) limita la respuesta. Si, por ejemplo, R=1TΩ y Cp=100pF, la frecuencia de corte es fc=1/2πRCP=1.6x10-3 Hz. El tiempo de respuesta es, pues, tr=0.35/fC=220s.

Con el convertidor corriente-tensión de la figura (b), la respuesta es mucho más rápida. La función de transferencia es también paso bajo con fc=1/ 2πRC. Para R= 1TΩ, la capacidad asociada sería más o menos de 1pF. Con estos valores fc=0.16Hz, y tr=0.35/fc=2.2s. El efecto del AO ha sido eliminar prácticamente la capacidad parásita al quedar dividido su valor por A en la función de transferencia.

Si la impedancia de entrada del AO se supone infinita, la impedancia de entrada del circuito (a) es R mientras que la del circuito (b) sería R/A, lo que supone un efecto de carga mucho menor.

Como ejemplo extraído del datasheet de un amplificador operacional (AD745) podría valorarse el siguiente acondicionamiento.

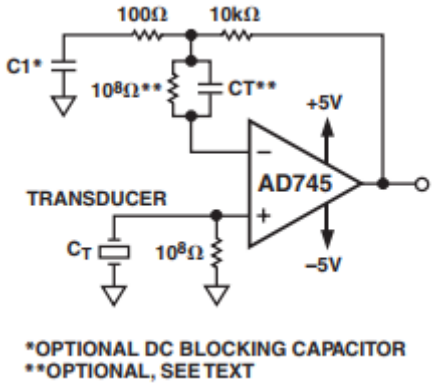


Imagen 43. Ejemplo de acondicionamiento de piezoeléctrico

2.2.3 Acondicionamiento de piezoeléctricos en baterías electrónicas

Antes de hablar del acondicionamiento usado en baterías electrónicas conviene describirlas un poco para poder entrar en situación. Se conoce como batería electrónica al instrumento de percusión en el que el sonido es emitido por un generador de ondas electrónicas o un sampler, que produce las ondas sonoras en las baterías acústicas. Una batería electrónica común está compuesta de tres elementos principales:

- Los pads, en los cuales se encuentran los sensores piezoeléctricos llamados comúnmente triggers.
- Un convertidor trigger-to-MIDI, encargado de transformar la señal creada por los triggers a una señal MIDI.
- Un módulo sampler, que produce un sonido determinado por la señal MIDI recibida.

Al golpear un pad con la baqueta, se crea una diferencia de potencial en los sensores piezoeléctricos. Las señales resultantes son enviadas a un módulo mediante cables, y son transformadas en ondas de sonido, las cuales producen el sonido de batería deseado, atendiendo al pad que origina la señal MIDI, así como la fuerza aplicada a éste y otros datos.

Imagen 44. Roland TD 30KV, ejemplo de batería electrónica

Hay que destacar que los parches y los platos se hacen de materiales que hagan el menor sonido posible al golpearse, ya que estas baterías están pensadas mayormente como alternativas a una batería acústica cuando no se puede hacer ruido, o en caso más minoritarios, poder elegir los sonidos deseados en cualquier pad. Si nos basamos en los acondicionamientos de los sensores piezoeléctricos que encontramos en los fabricantes de baterías electrónicas más conocidos (Roland o Yamaha) descubrimos que no usan ningún circuito para ello.

Imagen 45. Captura de un pad Roland desmontado

Se pueden encontrar numerosas imágenes y videos por la red de desmontajes de estos toms y no existe ningún circuito intermedio entre el sensor piezoeléctrico y la clavija de salida. Sin embargo sí que existe un pequeño cono de gomaespuma entre el parche y el sensor, siendo este el único acondicionamiento (de tipo físico) que se encuentra. Desconocemos si en el módulo de la batería se realizará algún acondicionamiento por software, pero se puede comprobar que no existe un circuito (amplificador, divisor de tensión u otro tipo) entre el piezoeléctrico y el módulo. De hecho los cables que se encuentran a la salida de los pads y platos suelen ser notablemente largos, lo que conlleva capacidades parásitas, pero aun así no se usa acondicionamiento de tipo electrónico.

Imagen 46. Interior de un pad Roland

2.3 Interfaces de comunicación

2.3.1 Protocolo MIDI

2.3.1.1 Generalidades

El protocolo estándar de comunicaciones MIDI (Musical Instrument Digital Interface) se concibió en 1983 para comunicar sintetizadores musicales. La especificación fue creada como convenio entre fabricantes, que constituyeron la MMA (MIDI Manufacturers Association). Mantienen actualizada la especificación y la publican periódicamente.

La especificación consta de un nivel físico en el que describe conectores y cables. En un principio se estableció como estándar el conector DIN de 5 pines, pero con la aparición del USB el conector DIN fue quedando poco a poco en desuso.

Inicialmente comunicaba dos categorías de dispositivos separados físicamente: controladores y módulos de sonido. La conexión MIDI entre ellos es unidireccional: el emisor o controlador era, normalmente, un teclado semejante al de un piano, que era accionado por el músico; el receptor o módulo de sonido era un aparato electrónico capaz de generar sonidos de la altura musical correspondiente a la nota. La comunicación se establece mediante mensajes.

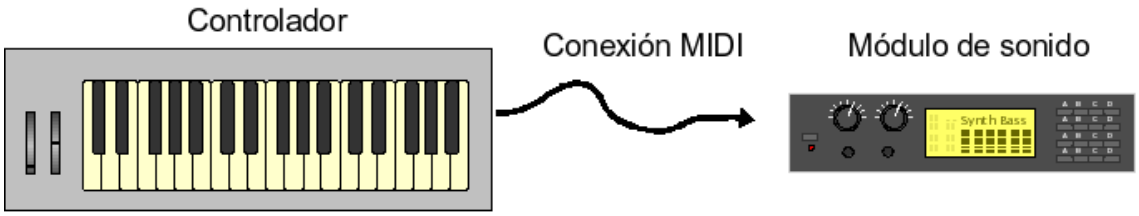


Imagen 47. Diagrama de conexión MIDI en sus inicios

Una vez establecido el estándar, se popularizó rápidamente. El protocolo pasó a utilizarse en otros campos, como el control automatizado de iluminación teatral o en la activación sincronizada de efectos especiales en la producción cinematográfica. En cuanto a la música, la estandarización creó un gran mercado de instrumentos musicales digitales con los que se creó el estilo musical Tecno-pop.

Los principales dispositivos son los siguientes:

- Controladores. Son las fuentes primarias de mensajes MIDI y por eso disponen de una entrada MIDI-OUT. Los más frecuentes son teclados, pero existen símiles electromecánicos de diversos tipos de instrumentos:

flautas, saxofones, guitarras, baterías y otros. Los controladores no producen ningún sonido.

- Módulos de sonido de diversos tipos. Son los receptores de mensajes y por eso disponen de una entrada MIDI-IN. Son los consumidores primarios de mensajes MIDI, y los traducen en señal de audio. La conversión depende del diseño del módulo y de los parámetros que se le hayan fijado. Estos es: en general, cada módulo tendrá sus sonidos, que se podrán modificar en mayor o menor medida mediante algunos parámetros. Los módulos suelen llevar una salida MIDI-THRU que reproduce los mensajes MIDI recibidos por la entrada para poder conectar otros módulos en cascada.
- Sintetizadores completos, que combinaban un controlador de tipo teclado con un módulo de sonido. Pueden funcionar de forma autónoma conectando el controlador con el módulo interno; pero también admiten conexiones de entrada y de salida con otros dispositivos MIDI. Como son productores y receptores de mensajes, sus conexiones son tres: MIDI-IN, MIDI-OUT y MIDI-THRU.
- Cajas de ritmos, que combinan un módulo de sonido (especializado en instrumentos de percusión) con una fuente de mensajes programable. Sus usuarios lo usan para programar esquemas rítmicos propios de una batería. Suelen disponer de las tres conexiones MIDI.
- Secuenciadores: dispositivos que almacenan la información MIDI de manera que pueden repetirla después de manera controlada. Un secuenciador permite el tratamiento de la información MIDI almacenada. También suelen disponer de las tres conexiones MIDI.
- Dispositivos de encaminamiento de mensajes: concentradores, bifurcadores y otros, que permiten combinaciones flexibles de diversos componentes.

Con estos dispositivos, se pueden crear complicadas topologías. Las instalaciones MIDI han dado soporte a la música electrónica durante 30 años. Una instalación como la de la figura muestra una posible configuración. En rojo aparece el flujo de los mensajes MIDI, en negro el audio.

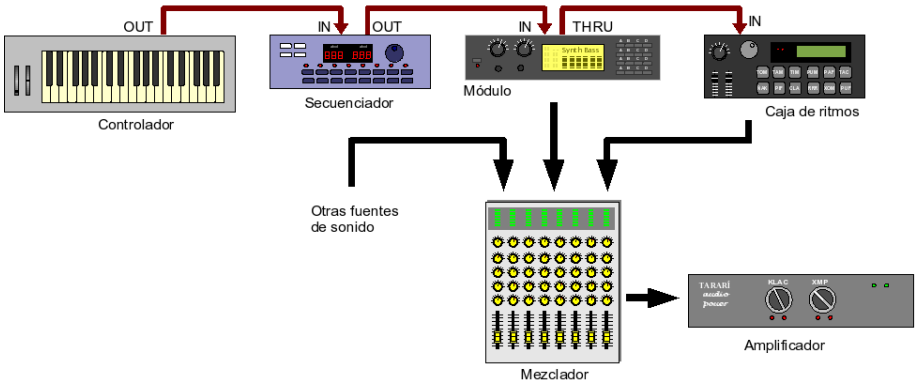


Imagen 48. Diagrama de una configuración MIDI con diversos instrumentos y dispositivos

2.3.1.2 MIDI y el ordenador

Con la entrada del computador en el mundo de la música electrónica, la mayor parte de los dispositivos están en fase de desaparición. Primero desaparecieron los secuenciadores, porque un computador de mediados los años 80, con el adaptador MIDI apropiado, podía ejecutar programas como Notator, Cubase y otros que realizaban las mismas funciones que un secuenciador hardware y otras muchas más.

Muchas de las DAW actuales (2011) son la evolución tecnológica de los secuenciadores MIDI de hace 20 años. Conforme la capacidad de almacenamiento y la velocidad de cálculo han ido creciendo, otros componentes MIDI se han "virtualizado", es decir, se han emulado por programa. Por ese motivo los módulos de sonido y las cajas de ritmos están desapareciendo de las tiendas sustituidos por programas cada vez más baratos. De los antiguos dispositivos, sólo los controladores han sobrevivido en el mercado.

También ha evolucionado la adaptación del MIDI al computador. Al principio, se instalaban en el bus de tarjetas o se conectaban al puerto serie estándar (Roland MPU-401). Posteriormente, las tarjetas de sonido corrientes incorporaron una conexión MIDI a través de un conector para juegos incorporado. El puerto USB cambió el aspecto del adaptador. En la actualidad, los dispositivos MIDI (diversos controladores y módulos) disponen de puertos USB alternativos a la conexión del estándar.

El estándar MIDI incorporó el SMF (Standard MIDI Files) que describe el formato de los archivos de computador donde se almacena información MIDI.

Finalmente, se han creado diversos estándares de sintetizador software, que permiten la inserción de instrumentos virtuales dentro de las DAW en forma de plug-in.

Fabricante	Estándar	Plataforma
Steinberg	*VSTi*	Windows, Linux, Mac
Digidesign	*RTAS instrument*	Windows, Mac
Microsoft	*DXI*	Windows
Apple	*Audio Unit instrument*	Mac
(libre)	*DSSI*	Linux
(libre)	*Nyquist*	Linux

Imagen 49. Resumen de estándares y fabricantes existentes en el mercado

Para integrar el encaminamiento de mensajes internamente, los computadores ofrecen diversas estrategias. En los entornos Windows, MIDI Yoke permite la

interconexión libre entre equipos externos conectados a los puertos MIDI físicos y los secuenciadores y sintetizadores virtuales disponibles. En Mac OS X, el sistema operativo integra esa funcionalidad.

2.3.1.3 Los canales

El protocolo MIDI rige un sistema de comunicaciones serie cuya unidad es el mensaje. Un mensaje está formado por uno o más bytes y cada byte va acompañado de un bit de paridad para detección de errores. Cada mensaje está producido por una acción del intérprete sobre el controlador (pulsar una tecla, soltarla, etc.). El controlador asocia los mensajes a un canal y es recibido por un módulo de sonido, afectando al sonido producido, o por un secuenciador que lo almacena. Emisor y receptor están configurados para comunicarse a través del mismo canal preestablecido.

Un sistema MIDI básico permite mantener 16 canales independientes. Así, a través del mismo conector, pueden transmitirse de forma separada el producto de 16 intérpretes, cada uno de ellos actuando sobre un controlador; cada interpretación será procesada por el receptor asociado al canal apropiado.

Esta organización es útil para la producción musical corriente. Veamos diversas situaciones:

- En el caso más sencillo, un intérprete actúa sobre un controlador que emite mensajes por un único canal. En el otro extremo, un módulo de sonido recibe los mensajes por el mismo canal y genera los sonidos apropiados (un piano, por ejemplo)
- Un intérprete actúa sobre un controlador que emite sobre el canal A mientras una caja de ritmos envía un patrón rítmico de mensajes sobre el canal B según un programa dado. Un módulo de sonido recibe los mensajes del canal A para generar sonidos de piano y otro módulo recibe los mensajes del canal B y genera sonidos de percusión.
- En una fase previa, el intérprete registra una interpretación en un secuenciador a través del canal A. Posteriormente, envía desde el controlador una segunda interpretación a través del canal B mientras el secuenciador reproduce lo grabado. Los módulos de sonidos generan sonidos de piano a partir del canal A y de órgano litúrgico a partir del canal B.
- Por etapas, un único intérprete registra 16 interpretaciones y las almacena de forma separada en los 16 canales de un secuenciador. Cada canal representa un instrumento concreto: guitarra, batería bajo, piano, saxofón, etc. El proceso permite la modificación y corrección de cada interpretación a gusto del intérprete. Finalmente, asocia cada canal a un módulo de sonido para escuchar el resultado.

El sistema MIDI es la herramienta básica de los compositores de música popular desde los años 80. Permiten a un único músico producir algo parecido a un conjunto. Buena parte de la música pop se realiza mediante sistemas MIDI.

2.3.1.4 Clasificación de instrumentos según MIDI

Con el nombre de instrumento MIDI, abstraeremos cualquier generador de sonido físico o virtual. Los instrumentos MIDI pueden ser de diversos tipos.

- Monofónico o polifónico. Un instrumento monofónico sólo puede producir un sonido a la vez. Si el intérprete pulsa dos teclas simultáneamente, o si pulsa una tecla antes de liberar otra, el instrumento seleccionará uno de los sonidos correspondientes. Un instrumento que simula un saxofón, por ejemplo, puede ser monofónico por similitud con el instrumento real. Muchos de los sintetizadores clásicos eran monofónicos. Los instrumentos polifónicos, en cambio, pueden hacer sonar un número limitado de notas. Un instrumento que simule un piano, por ejemplo, deberá tener una polifonía de 32 o más notas si quiere aproximarse a un piano real.
- Monotímbrico o politímbrico. Esta distinción se aplica a los instrumentos que ofrecen diversos timbres seleccionables. Estos instrumentos se denominan módulos y los actuales suelen disponer de un surtido de cientos de timbres: pianos, instrumentos de viento, de cuerda, de percusión, órganos y muchos otros. Un módulo es monotímbrico si sólo permite seleccionar un timbre dado y politímbrico cuando permite la selección concurrente de varios timbres. Un módulo politímbrico equivale a un conjunto de instrumentos.
  Considera un instrumento politímbrico capaz de recibir mensajes por los 16 canales MIDI y que contiene un banco de 128 timbres distintos. La configuración del módulo consistirá en asignar a cada canal de entrada un timbre.
  Los instrumentos politímbricos de propósito general se adhieren al subestándar GM (General MIDI) que codifica los nombres y los identificadores numéricos de 128 timbres.
- Mapeados o no mapeados. En los instrumentos mapeados cada nota tiene asociada timbre. Los sonidos de los instrumentos mapeados no dan sensación de altura musical, y no tiene sentido construir melodías con ellos. Los casos más frecuentes de instrumento mapeado son las percusiones y los efectos especiales. Para las percusiones, el subestándar GM especifica nombres del instrumento y las teclas del controlador a que van asociados. Los no mapeados son los demás instrumentos: todas las notas de un instrumento MIDI con sonido de piano corresponden al mismo timbre (el del piano) y a las posibles alturas musicales

2.3.1.5 Mensajes

Los mensajes MIDI están estructurados en octetos. Cuando se transmiten, a cada octeto se le añade un bit de paridad para detección de errores.

El primer octeto de un mensaje contiene un comando codificado en cuatro bits de la forma 1MMM (en hexadecimal, 0x8 a 0xF) y un identificador de canal de forma CCCC (por eso hay 16 canales, que generalmente se numeran del 1 al 16).

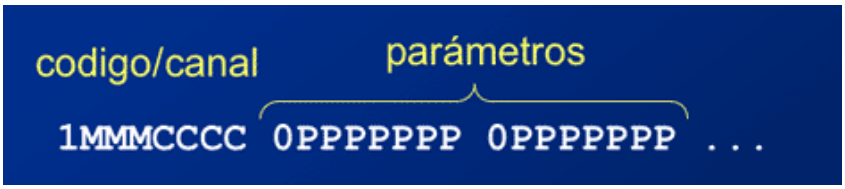


Imagen 50. Estructura de los mensajes MIDI

Los comandos admitidos en esta codificación son:

Mensaje	Cód.	Parámetro 1	Parámetro 2
Note Off	0x8	número de nota	velocidad
Note On	0x9	número de nota	velocidad
Note Aftertouch	0xA	número de nota	presión
Controller	0xB	núm. de controlador	controller value
Program Change	0xC	núm. de programa	—
Channel Pressure	0xD	presión	—
Pitch Bend	0xE	*pitch value (LSB)*	*pitch value (MSB)*

Imagen 51. Tabla resumen de mensajes, códigos y parámetros de un mensaje MIDI

Cada comando va acompañado de sus parámetros característicos. Los parámetros se alojan en octetos cuyo bit más significativo es siempre 0. Los siete bits restantes codifican un valor comprendido entre 0 y 127.

Los comandos Note On y Note Off, por ejemplo, codifican la tecla en un rango suficiente para representar las 88 notas del piano y otras que no existen en los instrumentos analógicos tradicionales.

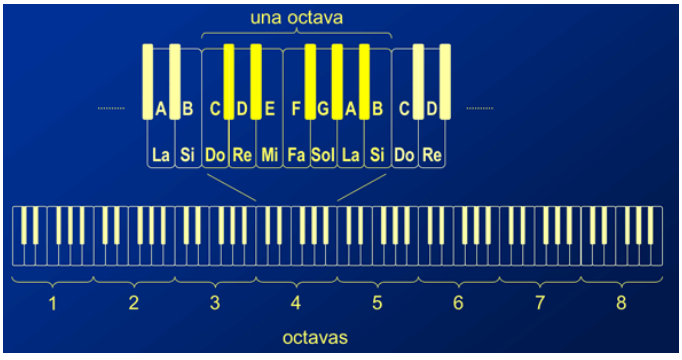


Imagen 52. Diagrama resumen de un teclado

	Note Numbers											
Octava	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
-1	0	1	2	3	4	5	6	7	8	9	10	11
0	12	13	14	15	16	17	18	19	20	21	22	23
1	24	25	26	27	28	29	30	31	32	33	34	35
2	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59
4	60	61	62	63	64	65	66	67	68	69	70	71
5	72	73	74	75	76	77	78	79	80	81	82	83
6	84	85	86	87	88	89	90	91	92	93	94	95
7	96	97	98	99	100	101	102	103	104	105	106	107
8	108	109	110	111	112	113	114	115	116	117	118	119
9	120	121	122	123	124	125	126	127				

Imagen 53. Equivalencia entre un teclado y los mensajes MIDI que envía

La relación entre la frecuencia f del sonido y el código t de tecla es $f = 8.1757989156 \cdot 2^{t/12}$.

Los controladores con sensibilidad en el teclado codifican la velocidad de pulsación y de liberación de cada nota en el rango de 0 a 127. Los módulos de sonido pueden interpretar la velocidad como potencia del sonido generado, pero también como la duración de ataque de la envolvente o cualquier otro parámetro propio del instrumento.

2.3.1.6 Especificaciones técnicas del bus

El protocolo definido en 1983 especifica tanto la conexión física y el interfaz de hardware, como el formato de los datos y el orden y disposición de los mensajes que se pueden transmitir entre dispositivos. El protocolo MIDI es bastante parecido al RS-232, aunque utiliza niveles eléctricos diferentes y ofrece una mayor velocidad de transmisión. Los mensajes se transmiten de forma binaria y en serie, es decir, mediante pulsos (bits) sucesivos. La transmisión se produce de forma asíncrona o, lo que es lo mismo, siempre que un dispositivo decida enviar un mensaje (por ejemplo, porque un músico ha apretado una tecla). Esta asincronía obliga a que cada byte de mensaje vaya rodeado de un bit de comienzo y un bit de final. Estas transmisiones se realizan a una velocidad de 31.250 bits por segundo, por lo que la velocidad máxima de transmisión es de 3125 bytes/sec. El interfaz MIDI de un dispositivo es el responsable de recibir y transmitir estos mensajes. Aunque la mayoría de dispositivos MIDI incluyen como mínimo un receptor y un emisor, también es posible que incluyan tan solo uno de los dos.

- El puerto emisor, denominado MIDI OUT, se encarga de convertir los datos digitales generados por el dispositivo en series de voltajes eléctricos.
- El puerto receptor, denominado MIDI IN, realiza el proceso inverso.
- Puede existir un tercer puerto, denominado MIDI THRU, que simplemente reenvía la información llegada al MIDI IN del interfaz. Todos ellos utilizan

conectores DIN hembras de cinco pines (de los cuales sólo se utilizan en realidad tres).

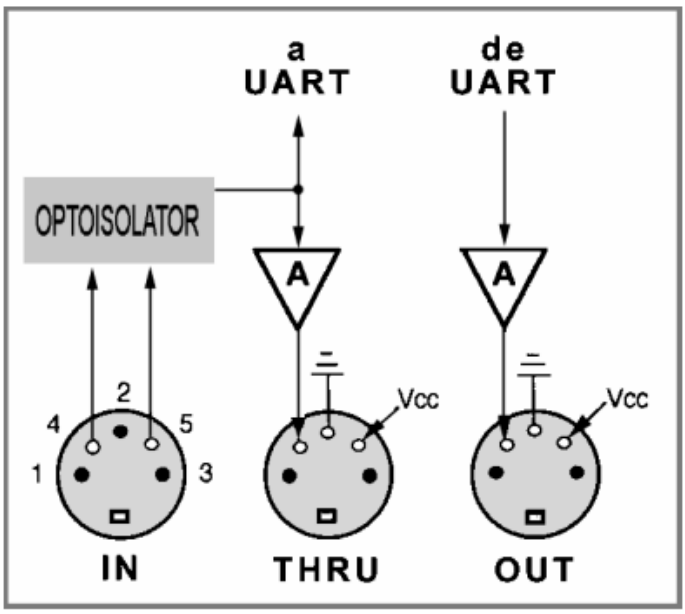


Imagen 54. Esquema simplificado Interfaz MIDI

Los cables MIDI, que utilizan forzosamente conectores DIN machos, conectan el MIDI OUT o el MIDI THRU de un dispositivo con el MIDI IN de otro. Su construcción garantiza la transmisión sin errores en longitudes inferiores a 15 metros.

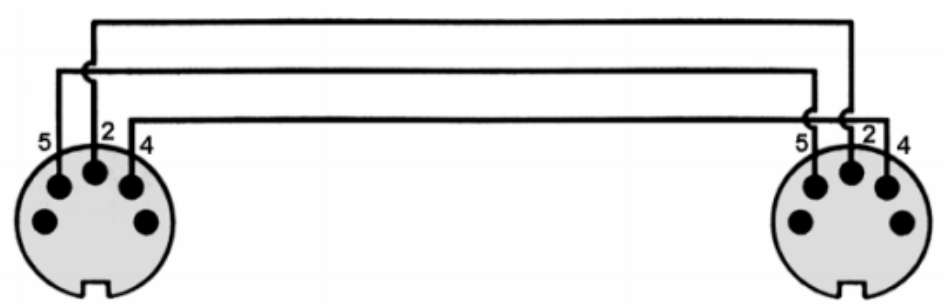


Imagen 55. Esquema conexiones cable MIDI DIN 5 pines

2.3.2 MIDI sobre USB HID

Antes de ver como el protocolo MIDI ha conseguido incluirse dentro un estándar y protocolo más grande y más estandarizado como el USB conviene dar una pequeña descripción de HID.

HID (por sus siglas en inglés Human Interface Device), o dispositivo de interfaz humana, hace referencia a un tipo de interfaces de usuario para computadores que interactúan directamente, tomando entradas proveniente de humanos, y pueden entregar una salida a los humanos. El

término HID comúnmente se refiere a la especificación USB-HID. Éste fue acuñado por Mike Van Flandern de Microsoft cuando propuso al comité USB, crear un grupo de trabajo para la clase Human Input Device. El nombre del grupo de trabajo se cambió a la clase Human Interface Device sugerido por Tom Schmidt de DEC, ya que la norma propuesta soportaba la comunicación bidireccional.

En el protocolo HID existe 2 entidades: el host y el dispositivo. El dispositivo es la entidad que directamente interactúa con el humano, como lo hace un teclado o un ratón. El host se comunica con el dispositivo y recibe datos de entrada del dispositivo en las acciones ejecutadas por el humano. Los datos de salida van del host al dispositivo y luego al humano. El ejemplo más común de un host es un computador, pero algunos celulares y PDA también pueden ser hosts. Cuando se valoró el uso de dispositivos e interfaces MIDI se llegó a la conclusión de que estos cumplían las condiciones y especificaciones de los dispositivos HID, por lo que se optó por introducir estos dispositivos dentro del USB-HID, aprovechando así la estandarización existente y las características que pudieran introducir ventajas en el protocolo MIDI.

El protocolo HID realiza la implementación de los dispositivos de manera sencilla. Los dispositivos definen sus paquetes de datos y luego presentan un descriptor HID al host. El descriptor HID es codificado como un grupo de bytes que describen los paquetes de datos del dispositivo. Esto incluye: cuantos paquetes soporta el dispositivo, el tamaño de los paquetes, y el propósito de cada byte y bit en el paquete. Por ejemplo, un teclado con un botón que ejecuta el programa de calculadora puede decirle al host que el estado de presionar/soltar ese botón, es almacenado en el 2º bit del 6º byte en el paquete de datos número 4 (nota: estas localizaciones solo son ilustrativas y son específicas de cada dispositivo). El dispositivo normalmente almacena el descriptor HID en la ROM y no se necesita intrínsecamente analizar sintácticamente el descriptor HID. Algunos hardware de ratones y teclados en el mercado de hoy son implementados usando solo una CPU de 8 bits.

Se espera del host que sea una entidad más compleja que el dispositivo. El host necesita obtener el descriptor HID del dispositivo y lo analiza antes de que se puede entablar la comunicación con el dispositivo. Analizar el descriptor HID puede ser complicado. Se sabe que algunos sistemas operativos tenían shipped bugs en los controladores de los dispositivos para analizar los descriptores HID, años antes de que los drivers del dispositivo fueran originalmente lanzados al público. Sin embargo, esta complejidad es la razón del porqué fue posible la innovación rápida con dispositivos HID.

El mecanismo anterior describe lo que es conocido como «el modo de reporte» HID. Ya que se conoce que no todos los hosts tendrán la capacidad de analizar los descriptores HID. El HID también define «el modo de arranque». En modo arranque solo dispositivos específicos soportan características específicas, ya que paquetes de datos definidos son usados. El descriptor HID no es usado en

este modo así que la innovación está limitada. Sin embargo, el beneficio es que esta mínima funcionalidad es aún posible en los hosts que no soportan HID.

Visto en funcionamiento del protocolo HID y la necesidad que aparece de aunar todo tipo de conectores se valoró la posibilidad de intentar integrar el protocolo MIDI en el USB, el protocolo por excelencia de conexión y comunicación entre ordenadores, dispositivos y periféricos. Además introduciendo el MIDI en la arquitectura USB podemos conseguir mejoras como:

- Estandarizar los tipos de conectores: hoy en día tenemos distintos tipos de conexiones USB, pero todas están estandarizadas por USB-IF.
- Intercambio en caliente: los dispositivos USB pueden ser conectados y desconectados según se necesite, mientras el ordenador está funcionando. Así, no es necesario reiniciar.
- Plug and Play: los dispositivos USB están divididos en tipos funcionales (audio, imagen, interfaz de usuario, almacenamiento masivo). De este modo, el sistema operativo puede identificar, configurar y cargar el driver apropiado automáticamente, nada más conectar el dispositivo USB.
- Alto rendimiento: USB ofrece tasas de transferencia bajas (1.5 Mbit/s), a toda velocidad (12 Mbit/s) y alta velocidad (hasta 480 Mbit/s). USB 3.0 (SuperSpeed USB) alcanza los 5.0 Gbit/s.
- Capacidad de expansión: en teoría, se pueden conectar hasta 127 dispositivos distintos en un mismo bus USB a la vez.

Es importante tener claro que Universal Serial Bus (USB) es un bus controlado por host. Todas las transferencias de datos se inician y se controlan desde el host, y los periféricos USB son esclavos que responden a los comandos de ese host. Por eso, para utilizar periféricos USB-MIDI siempre necesitarás un ordenador, smartphone o tablet en el sistema, para controlar e iniciar la comunicación.

Los dispositivos USB se definen en clases funcionales específicas, como decíamos antes. Por ejemplo: imagen, interfaz de usuario (teclado, ratón, joystick), almacenamiento masivo y audio. De esa forma, el sistema operativo puede saber para qué ha sido diseñado el dispositivo y cargar automáticamente un driver "class compliant" (es decir, un driver que "cumple" con las especificaciones de la clase de dispositivo que vamos a usar).

En 1999, USB-IF desarrolló la especificación MIDI USB en cooperación con la MIDI Manufacturers Association, incluyendo así la clase de dispositivos de audio. Por eso a veces pasa que, cuando conectas un periférico USB-MIDI, el sistema operativo dice que se ha conectado un dispositivo USB-Audio: en lo que respecta al USB, el MIDI es un dispositivo Audio Class Compliant.

Sin embargo hay una característica típica de los dispositivos USB HID que no beneficia a los dispositivos MIDI. Los drivers class compliant son muy útiles porque no es necesario descargar ningún software externo, pero a menudo, los

drivers específicos de los fabricantes ofrecen funcionalidades añadidas. Por ejemplo, el caso de Yamaha: como la transferencia de datos sobre USB es mucho más rápida que las conexiones DIN de 5 pines, es posible tener múltiples puertos MIDI (un puerto es un grupo de 16 canales MIDI) en un único cable USB; el driver dedicado de Yamaha ofrece ocho puertos USB-MIDI de alta velocidad, incluye los nombres de todos los dispositivos que son compatibles con el driver, y añade algunas capacidades de ruteo. Estas funcionalidades sólo se pueden conseguir descargando el driver de la web de Yamaha.

Muchas interfaces de audio son también interfaces MIDI que envían y reciben datos con un cable USB. Así, si se usa una interfaz MIDI o audio conviene revisar el manual y la web del fabricante para ver si hay un driver USB específico para el producto, que añada funcionalidad adicional.

2.3.3 Interfaces y módulos de batería MIDI comerciales

Antes de hacer un análisis de los productos que existen en el mercado que permiten comunicar vía MIDI una batería electrónica es conveniente aclarar la diferencia entre los dos productos más comunes.

2.3.1 Módulo

El módulo de una batería electrónica es un dispositivo que recibe las señales de los sensores piezoeléctricos que albergan los pads y los platos, y, a partir de estas, es capaz de producir sonidos de forma autónoma, sin necesidad de un ordenador. Normalmente dispone de potenciómetros e incluso pantalla para poder regular la sensibilidad de las señales o seleccionar distintos sonidos que se encuentran dentro de su memoria. Aunque pueden funcionar de forma autónoma, sin necesidad de un ordenador ni uso del protocolo MIDI, incluso los módulos de gama más baja ofrecen conexión MIDI mediante USB o conector DIN de 5 pines (fácilmente convertible a USB mediante un adaptador).

Imagen 56. Conversor DIN de 5 pines a USB para MIDI

La marca más conocida en este campo es Roland, la cual ha llegado a desarrollar equipos que rozan la sensibilidad y el feeling de una batería acústica. Se analizarán los dos módulos de más alta gama de ambos fabricantes para poder ver las características que ambos incorpora.

El buque insignia de la compañía es el Roland TD-50, disponible por 2490 €. Las características a destacar de este producto son:

- Memoria con capacidad para 100 Drum Kits
- Tecnología de modelado Prismatic Sound Modelling
- Reproducción de WAV-Samples desde tarjetas SD
- Salida de audio USB de 10 canales para grabación multipista con solo un cable USB
- 14 entradas Jack estéreo de 6,3mm para pads de un sensor
- 3 entradas para Pads Roland Digital Drum con tecnología Multi Sensor (detección de la zona donde se golpea dentro del mismo parche)
- 2 salidas Main XLR (simétricas) que permite usar el módulo como amplificador de sonido para altavoces o monitores
- 8 salidas Jack de 6,3 mm con elección de los sonidos que salen por cada salida (para poder grabar sin necesidad de ordenador ni MIDI)
- Salida de auriculares estéreo minijack de 3,5 mm
- Entrada estéreo minijack de 3,5 mm para conexión de reproductores de MP3 o CD
- MIDI In & Out con conexión DIN de 5 pines
- Interfaz USB-MIDI para conexión directa con ordenadores Mac o PC
- Entrada estéreo Jack de 6,3 mm para interruptor de pedal (para uso de pedales de charles para batería electrónica)
- Metrónomo con función "Quiet Count"
- Ecualizador de 3 bandas y compresor para cada uno de los Pads
- 3 procesadores multiefectos con 30 algoritmos
- Multicompresor de 2 bandas y ecualizador de 4 bandas para las salidas

Imagen 57. Módulo Roland TD-50

2.3.2 Interfaz

Una interfaz MIDI hace el mismo papel que un módulo, conseguir sonidos a partir de las señales de los sensores piezoeléctricos, pero con la diferencia de que una interfaz depende de un ordenador. De esta forma la señal eléctrica llega a la interfaz. En esta se pueden ajustar ciertos parámetros como sensibilidad, volumen o sonidos a elegir por pad, y a partir de aquí la interfaz envía los respectivos mensajes MIDI al ordenador, el cual, mediante un software como Cubase, Ableton o FL Studio, produce los sonidos que el usuario puede elegir. Está claro que un módulo es superior a una interfaz, pero el menor precio de una interfaz puede hacerla atractiva para según que usos.

Una de las interfaces más conocidas en el mercado es la ATD aD5, la cual se puede encontrar por 756 €. Las especificaciones de este producto son:

- 8 entradas para pads multizona
- 2 entradas para platos trizona (especial para platos Roland y Yamaha)
- Salida para auriculares
- Salida de jack de 3,5 mm para grabar o incluir en un mezclador analógico
- Entrada de 3,5 mm para conectar audio externo
- Interfaz USB para conectar con PC o Mac
- Ranura para Micro SD donde guardar configuraciones

Imagen 58. Interfaz ATD aD5

3 Desarrollo del circuito de acondicionamiento de los piezoeléctricos

3.1 Descripción de los sensores utilizados

Los sensores utilizados para este proyecto son sensores piezoeléctricos estándar. Principalmente se buscaban unos sensores que tuvieran un buen funcionamiento y que su precio no fuera muy excesivo. Se optó por estos en particular debido a las buenas opiniones de los clientes en Ebay y al envío, que tardó una semana al enviarse desde España. Se eligieron dos medidas, 27 y 35 milímetros, debido a que se pensó en usar los pequeños en los platos, ya que el especio en la campana del ride es reducido y al tener menos superficie serían más fáciles de fijar. Perfectamente se podrían haber elegido cualquiera de las dos medidas en todas las partes, las especificaciones son exactamente las mismas.

Imagen 59. Fotografía de los dos tipos de piezoeléctricos usados

3.2 Caracterización en laboratorio

Una vez conseguidos los sensores que se iban a encargar de recoger las vibraciones de los parches de cada cuerpo y de los platos se comprobó cómo funcionaban y que tipo de señal se conseguía de ellos. Para esto se usaron los datos que proporcionó el osciloscopio.

Para poder recoger los valores se utilizó la herramienta OpenChoiceDesktop, disponible en la web de Tektronik, marca que ha fabricado el osciloscopio que se usó durante este proyecto.

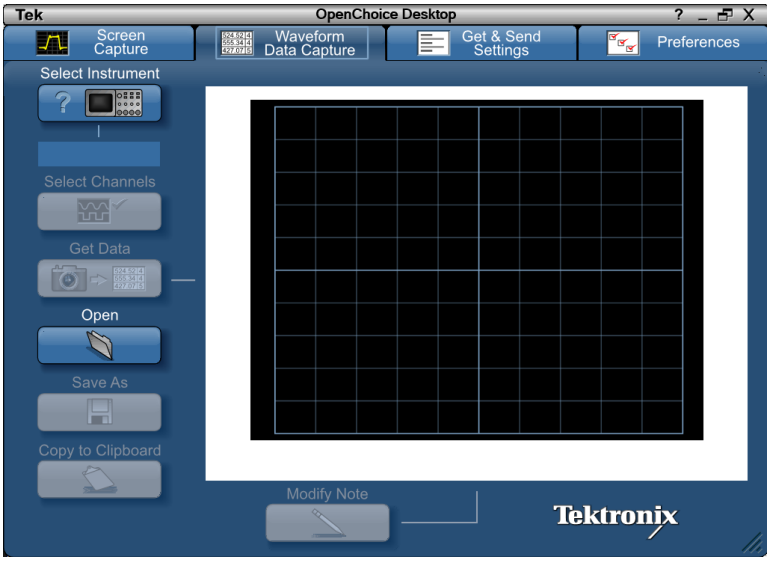
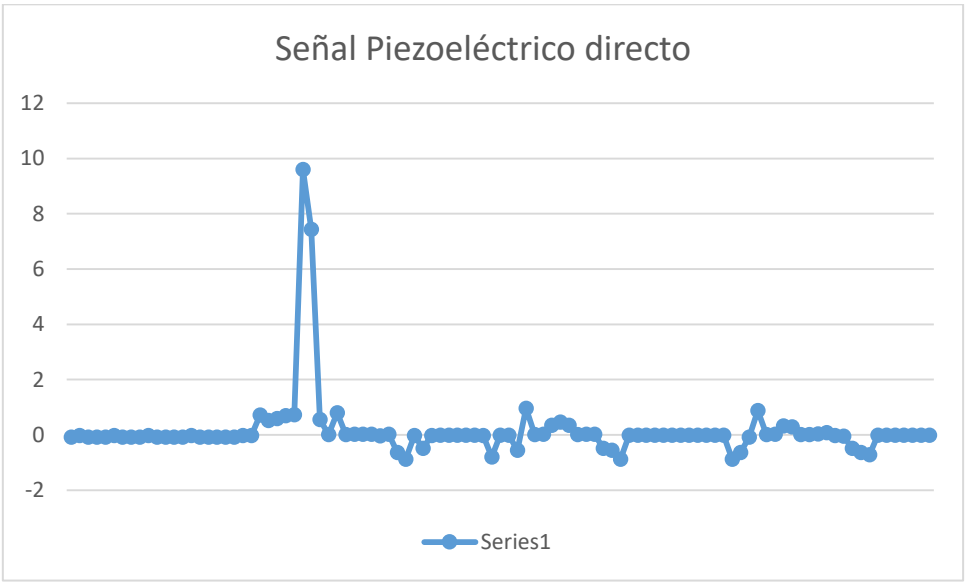


Imagen 60. Captura del software del osciloscopio para captura de datos

Seleccionando el instrumento que aparecerá al conectarlo al USB y seleccionando "Save As" y en formato CSV se consiguen los datos para poder representarlos en Excel, por ejemplo.



3.3 Circuitos de acondicionamiento

Desde el principio del proyecto la idea fue basarse en el sistema usado por marcas como Roland o Yamaha, en el que usan un pequeño cono de gomaespuma sobre los piezoeléctricos para atenuar levemente la señal, consiguiendo así evitar sobrepicos de tensión y creando cierto amortiguamiento que permita que la señal decaiga más rápido que si el sensor se pegara directamente a un parche o plato.

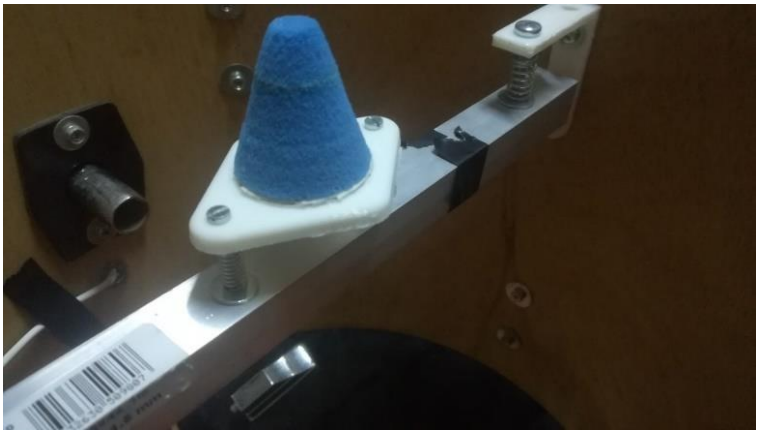


Imagen 61. Estructura donde se montó el sensor piezoeléctrico

De esta forma, y aunque se dedicará un apartado exclusivo para ello, se instaló en el tom que se usó para las pruebas una estructura donde sujetar el piezo con su respectivo cono. Con este montaje hecho, las primeras pruebas consistieron en montar el sensor piezoeléctrico en la base del soporte y pegar a este el cono de gomaespuma. Una vez hecho esto se conectó el sensor a un osciloscopio para poder ver cuál era la salida que se conseguía cuando se golpeaba el parche con una baqueta.

En este caso se decidió recoger los datos del osciloscopio como captura de pantalla del osciloscopio (en lugar de CSV, ya que aparecen muchos problemas a la hora de representar los valores). Esto se consigue de forma similar al caso anterior.

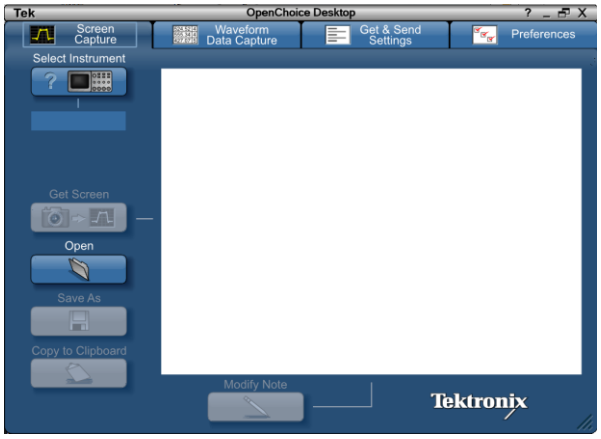


Imagen 62. Captura del software del osciloscopio para captura de pantalla

La única diferencia es seleccionar la pestaña de Screen Capture, seleccionar nuestro dispositivo, pulsar la opción de Get Screen cuando la onda esté en la pantalla del osciloscopio y finalmente se guarda en el formato de imagen que más convenga.

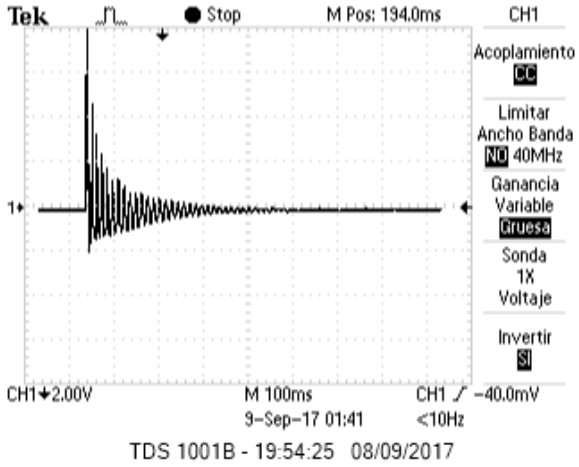


Imagen 63. Onda resultante de golpear el parche sin acondicionamiento

Viendo la imagen anterior se puede ver el resultado esperado. La señal no presenta ruidos indeseados, la gomaespuma hace su efecto de atenuar el voltaje producido (el pico es de casi 6 voltios cuando antes, sin el cono, podíamos llegar a 15 fácilmente) y la caída no se prolonga mucho en el tiempo.

Sin embargo siguen apareciendo dos problemas: el primero es que aparecía cierto voltaje negativo que podría llegar a dañar a el microcontrolador. El segundo problema es que, aunque se había conseguido reducir el tiempo de bajada, todavía seguía bajando de forma demasiado lenta. Debido a estas dos ideas se planteó la idea de realizar un acondicionamiento electrónico (se habla de electrónico porque ya se instaló la gomaespuma como acondicionamiento mecánico).

El esquema en el que se pensó fue el siguiente:

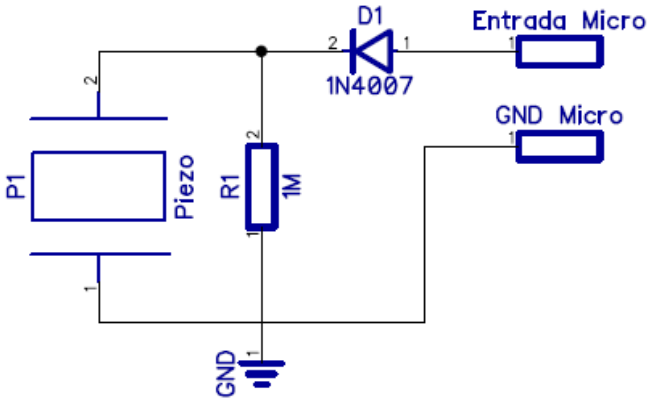


Imagen 64. Esquemático propuesto para el acondicionamiento del sensor

El valor de 1M para la resistencia que se encuentra en paralelo con el sensor piezoeléctrico realiza la misma función que una resistencia en paralelo con un condensador: marca la velocidad de descarga. Se pensó en esta idea ya que en parte un sensor piezoeléctrico también se descarga de forma similar a un

condensador. El valor de 1M fue elegido ya que se habían escogido resistencias de este valor en otros proyectos similares (véase ArduinoMidiDrums de Evan Kale), y aunque se probaron diferentes valores, se vio que este era el más adecuado. Respecto al diodo, sirve simple y únicamente como rectificador de mediaonda que permite eliminar el voltaje inverso que aparecía en la anterior captura. La orientación es la indicada porque los piezoeléctricos están montados al revés (con el círculo blanco hacia arriba), por lo que el voltaje circula al revés de lo supuesto y el diodo se debe colocar así (cátodo con el negativo, cable negro).

Se eligieron los diodos 1N4007 porque son fáciles de encontrar, muy robustos (no van a tener problemas en esta aplicación) y porque viendo que el pico que daba en el sensor en la captura anterior (casi 6 voltios) una caída de 0,7 voltios haría que nuestra señal se encontrara cerca a los 5 voltios, valor ideal para una entrada analógica.

Después de plantear esta idea, se montó en una pequeña placa perforada para poder conectarse fácilmente al sensor del tom de pruebas.

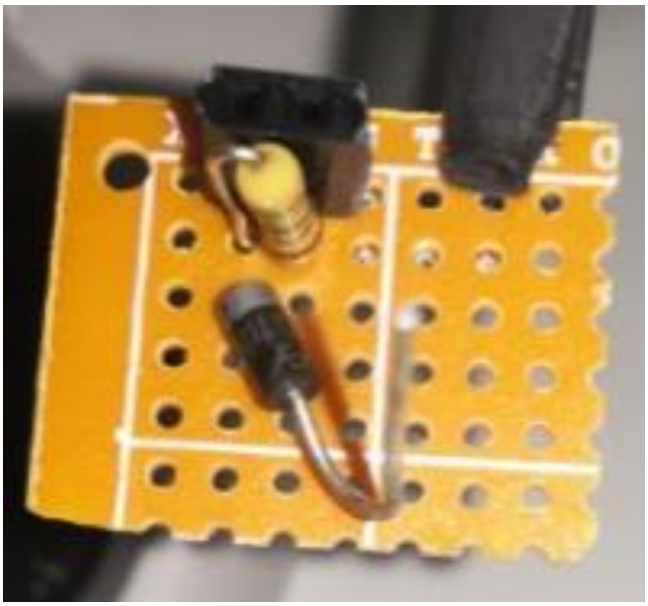


Imagen 65. Acondicionamiento en placa perforada

El siguiente paso fue conectar el sensor a la entrada del circuito (pinera hembra de la imagen) y ver la salida en el osciloscopio (positivo del osciloscopio a patilla del diodo y negativo a una pista de GND que había en la placa). Conectándolo todo correctamente aparece la siguiente salida.

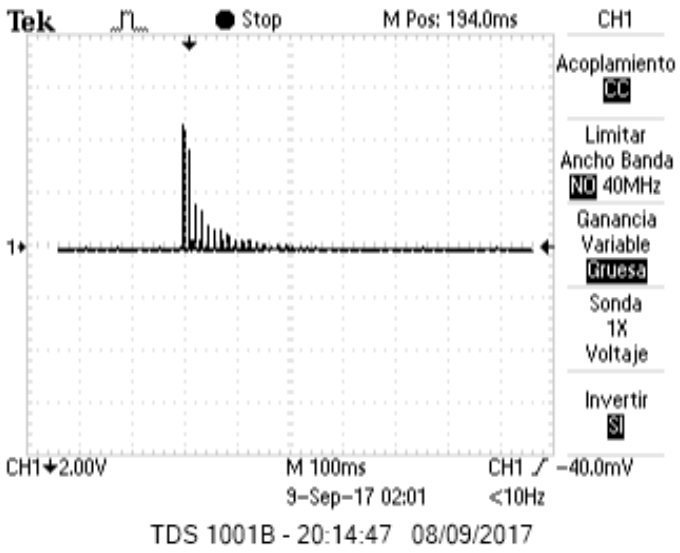


Imagen 66. Onda resultante de golpear el parche con acondicionamiento

Viendo la onda resultante se puede ver que con el acondicionamiento se ha conseguido lo que se buscaba: la onda baja a cero más rápido que antes y el voltaje inverso desaparece completamente. Además, la caída de voltaje que se produce no penaliza a la señal.

3.4 Pruebas realizadas

Después de ver que el acondicionamiento ha funcionado correctamente, el siguiente paso es introducir la señal en el microcontrolador elegido (Arduino en este caso) y ver las lecturas que este proporciona para poder ver si eran válidas. Para esto simplemente se siguió el esquemático mostrado anteriormente, conectando el ánodo del diodo a la entrada A0 de un Arduino Nano y conectando GND de nuestro circuito a GND del microcontrolador.

A continuación se cargó el siguiente código en el Arduino:

```
int sensor=A0;
int value;

void setup(){
Serial.begin(9600);
}

void loop(){
  value=analogRead(sensor);
 Serial.println(value);
  delay(20);
}
```

Este código únicamente lee el puerto A0, donde hemos conectado nuestro piezoeléctrico con su acondicionamiento, e imprime el valor registrado (de 0 a 1023) por el monitor serie del entorno de Arduino. Dichos valores se recogieron y se representan a continuación:

Imagen 67. Representación de los datos recibidos por el ADC

Se puede ver como los valores son convertidos de forma correcta por el conversor analógico-digital. Esto garantiza que se puede operar de manera correcta con estos datos cuando se programe el microcontrolador.

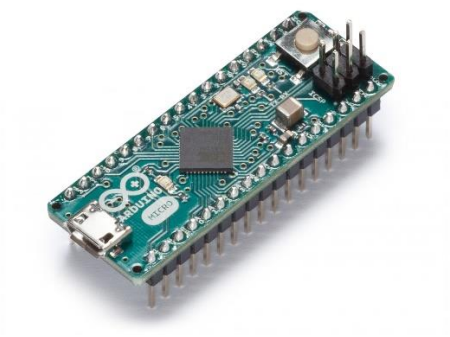
4 Desarrollo e implementación de Interface MIDI USB-HID

4.1 Descripción del microcontrolador usado

La elección del microcontrolador central del sistema viene condicionada principalmente por una de las restricciones del sistema: el sistema debe conectarse y comunicarse vía MIDI usando un puerto USB.

Si se pretende cumplir esta restricción y además basarnos en un sistema abierto (Arduino en este caso) debemos recurrir a las placas que poseen el microprocesador ATmega32u4, ya que este es el único compatible con USB-HID de la familia.

La decisión se basa ahora en elegir entre las placas Arduino Leonardo y Pro Micro.



Imágenes 68 y 69. Arduino Pro Micro y Leonardo

Viendo que las únicas diferencias entre ambas es que la placa Leonardo y la Pro Micro es que la primera incluye clavija para alimentación y es considerablemente más grande se acabó eligiendo la segunda.

Las especificaciones de la placa que se va a usar finalmente (Arduino Pro Micro) son las siguientes:

Microcontroller	ATmega32U4
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	20
PWM Channels	7
Analog Input Channels	12
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega32U4) of which 4 KB used by bootloader
SRAM	2.5 KB (ATmega32U4)
EEPROM	1 KB (ATmega32U4)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	48 mm
Width	18 mm
Weight	13 g

Imagen 70. Especificaciones de la placa Arduino Pro Micro

4.2 Descripción de USB-HID sobre Arduino Pro Micro

Una vez que se ha escogido el microcontrolador que se encargará de comunicarse con el ordenador con la misión de enviar los mensajes MIDI para

poder obtener finalmente los sonidos deseados, ahora se debe investigar cómo enviar los mensajes en sí.

Aunque es conocido que la placa es compatible con USB-HID, enviar mensajes según el protocolo MIDI no es algo trivial aunque se conozca la estructura de estos mensajes.

Viendo otros proyectos basados en Arduino y en entornos musicales (MIDI) se ha podido encontrar un proyecto denominado Arcore. Este proyecto se basa en un fork de las librerías hardware de Arduino con la misión de añadir funcionalidades MIDI a las placas que integren un microprocesador capaz de funcionar como dispositivo USB-HID (en caso de Arduino serían las placas Leonardo y Pro Micro).

Para poder usar este fork en nuestra placa primero se instalaran ciertas funcionalidades en el IDE de Arduino. Se descargará la carpeta donde se incluyen todos los archivos necesarios desde https://github.com/rkistner/arcore.

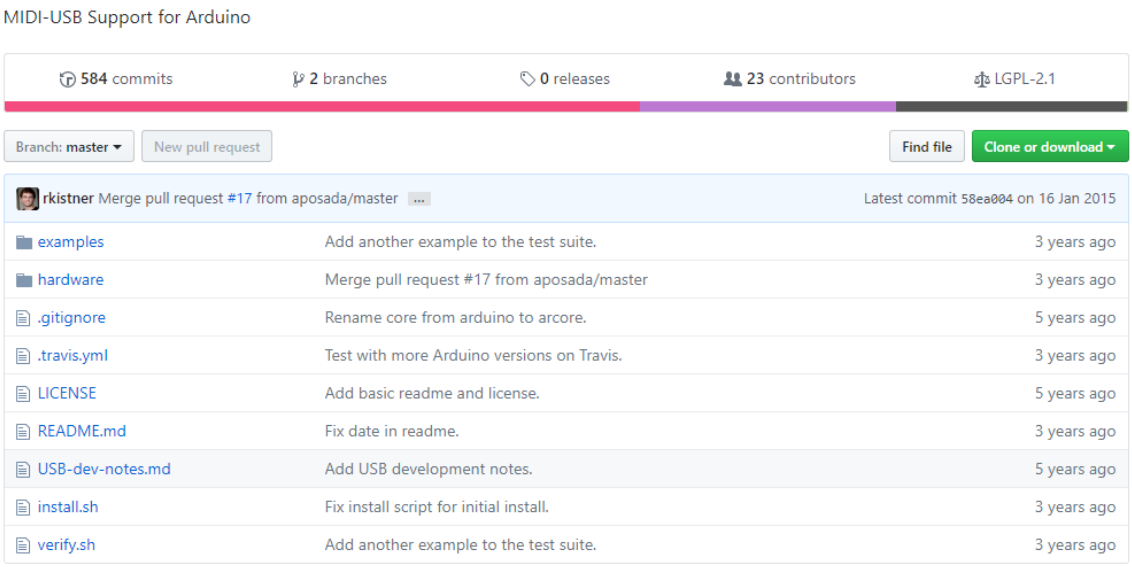


Imagen 71. Captura de la página de descarga de arcore

Una vez descargados los archivos, se incluiran en la carpeta de instalación del IDE de Arduino, aunque hay que mencionar que, según el autor, solo se asegura el funcionamiento de la librería en la versión 1.5.7 del IDE, y no funcionará en versiones anteriores a la 1.5.0.

Otra posibilidad para instalar estas librerías sería usar el Gestor de Tarjetas del entorno de desarrollo. Entrando en “Herramientas” y en “Placas” podremos ver esta opción:

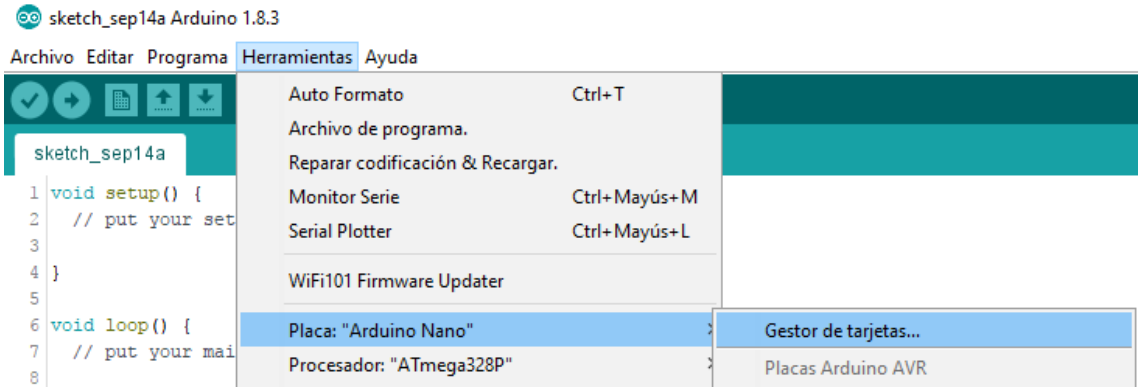


Imagen 72. Captura referente al acceso al Gestor de Tarjetas

Buscando la palabra clave “arcore” aparecerá la librería deseada, e instalarla solo supondrá pulsar sobre esta opción.

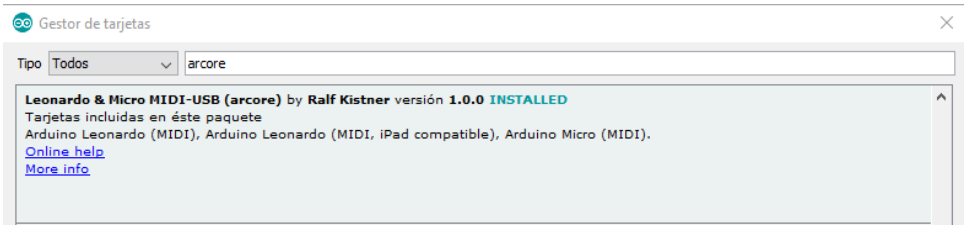


Imagen 73. Captura referente a la búsqueda de la librería arcore

Una vez instalada esta librería, que es una de las partes más importantes del proyecto en sí, podremos ver que en la sección de “Placa” aparecen tres nuevas placas, en las cuáles se permiten usar las funciones propias de la librería que se ha instalado, y que va a poder permitir programar nuestro microprocesador para enviar los mensajes MIDI necesarios.

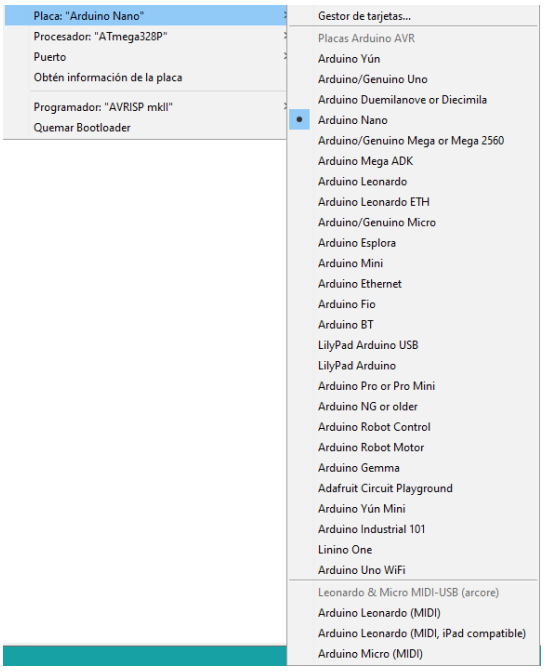


Imagen 74. Captura de las nuevas placas incluidas en el IDE.

4.3 Pruebas realizadas

Ahora que se dispone de las librerías necesarias para poder enviar y recibir mensajes MIDI entre el ordenador y nuestro microcontrolador, se procederá realizar las primeras pruebas para verificar que estos mensajes se envían correctamente y que, por tanto, se pueden generar sonidos a partir de comandos programados en nuestro Arduino.

Entrando a la página de la librería (incluida anteriormente) se puede encontrar un código de ejemplo que, según el autor, enviará un Note-on (Comienzo de nota) y un Note-off (Fin de nota) cada dos segundos:

```
void noteOn(byte channel, byte pitch, byte velocity) {
  MIDIEvent noteOn = {0x09, 0x90 | channel, pitch, velocity};
  MIDIUSB.write(noteOn);
}

void noteOff(byte channel, byte pitch, byte velocity) {
  MIDIEvent noteOff = {0x08, 0x80 | channel, pitch, velocity};
  MIDIUSB.write(noteOff);
}

// First parameter is the event type (0x0B = control change).
// Second parameter is the event type, combined with the channel.
// Third parameter is the control number number (0-119).
// Fourth parameter is the control value (0-127).

void controlChange(byte channel, byte control, byte value) {
  MIDIEvent event = {0x0B, 0xB0 | channel, control, value};
  MIDIUSB.write(event);
}

void loop() {
  noteOn(0, 48, 64);   // Channel 0, middle C, normal velocity
  MIDIUSB.flush();
  delay(500);

  noteOff(0, 48, 64);  // Channel 0, middle C, normal velocity
  MIDIUSB.flush();
  delay(1500);

  // controlChange(0, 10, 65); // Set the value of controller 10 on channel 0
to 65
}

void setup() {

}
```

Cargando este código en nuestra placa (hay que seleccionar la opción “Arduino Micro (MIDI)”) y entrando al software Addictive Drums se puede ver que el programa reconoce a la placa como un dispositivo MIDI y además se produce un golpe de charles cada 2 segundos, tal y como decía el autor de la librería.

Imagen 75. Captura del software Addictive Drums

Viendo que el código funciona lo siguiente es intentar que los sonidos que se producen en el programa sean deseados. Usando el tom que estaba montado para las pruebas que se realizaron en el apartado 3.3 conectamos el sensor piezoeléctrico al Arduino Pro Micro mediante el acondicionamiento que se menciona en el mismo apartado:

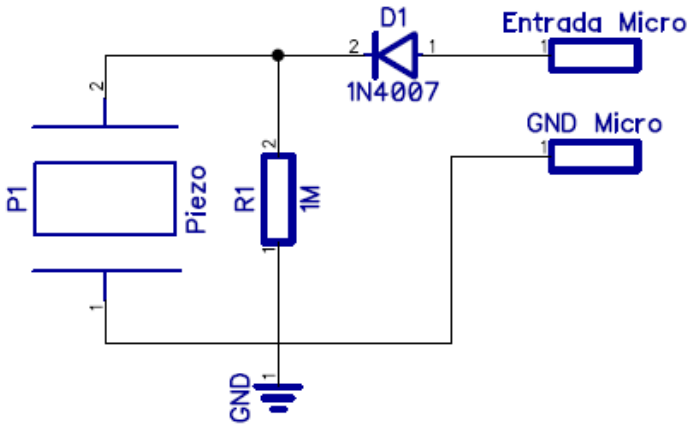


Imagen 64. Esquemático propuesto para el acondicionamiento del sensor

Usando como entrada del microcontrolador la entrada A0 de nuestra placa, se realiza el siguiente montaje en una protoboard:

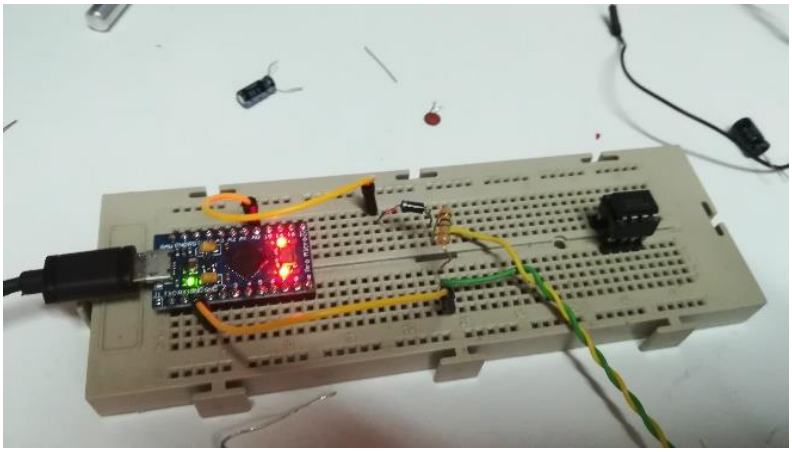


Imagen 76. Montaje en protoboard del acondicionamiento junto al microcontrolador

El código que se usa es una pequeña variación del ejemplo, donde simplemente se introduce la salida del acondicionamiento en la entrada A0 y se activa el NoteOn y seguidamente el NoteOff si el sensor supera un valor de 500. El código es el siguiente:

```
void noteOn(byte channel, byte pitch, byte velocity) {
  MIDIEvent noteOn = {0x09, 0x90 | channel, pitch, velocity};
  MIDIUSB.write(noteOn);
}

void noteOff(byte channel, byte pitch, byte velocity) {
  MIDIEvent noteOff = {0x08, 0x80 | channel, pitch, velocity};
  MIDIUSB.write(noteOff);
}

int sensor=A0;
int value;

void setup(){

}

void loop(){
  value=analogRead(sensor);
 // value=map(value,0,1023,0,255);
if(value>500){
  noteOn(0,48,64);
  MIDIUSB.flush();
  delay(50);
  noteOff(0, 48, 64);  // Channel 0, middle C, normal velocity
  MIDIUSB.flush();
  delay(110);
}
  delay(20);
}
```

Cargando el código vemos que el funcionamiento es correcto, por lo que se ha conseguido producir sonidos reales en el software a partir de golpes en el parche.

A partir de aquí el siguiente objetivo es ver en qué influyen los valores de las funciones noteOn y noteOff del código del ejemplo. Para esto la idea que se planteó fue la de crear un pequeño circuito que se pudiera conectar en una protoboard junto al Arduino y poder variar los valores de las funciones antes mencionadas.

Para hacer esta tarea lo más productiva posible se planteó un prototipo rápido de un sistema en el que se incluyera una pantalla en la que ver los valores de cada variable de las funciones y unos botones que nos permitiera varias estos tres valores. De esta forma no se tendrían que estar cargando diferentes códigos en el microcontrolador ni seleccionando el Arduino Micro en la lista de dispositivos MIDI en Addictive Drums cada vez que se cargara el código.

El diagrama de bloques que se planteó para esta idea fue el siguiente:

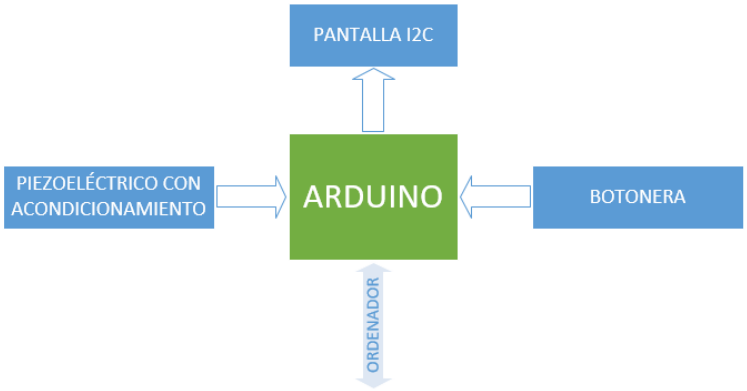


Imagen 77. Diagrama de bloques del sistema para variación de variables de las funciones MIDI

Con la idea de este diagrama de bloques se desarrolló un esquemático que nos permitiera construir la botonera, ya que la pantalla I2C se conectaría por cables hembra-macho y el piezoeléctrico con acondicionamiento ya se encuentra puesto en la protoboard.

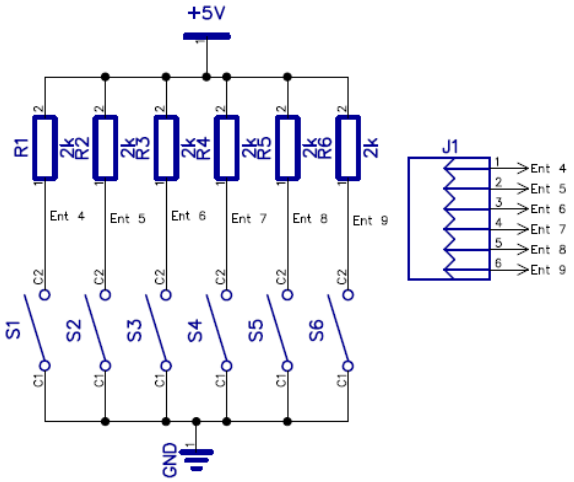


Imagen 78. Esquemático de la botonera

Una vez hecho el esquemático se procedió a realizar el montaje físico, ya que los botones de los que se disponían eran muy cortos para usarlos en la protoboard. Como este montaje solo iba a servir para experimentar con los valores de las funciones NoteOn y NoteOff se hará este montaje en placa perforada. Realizando las conexiones entre los componentes con cable y cordones de soldadura se consiguió este resultado:

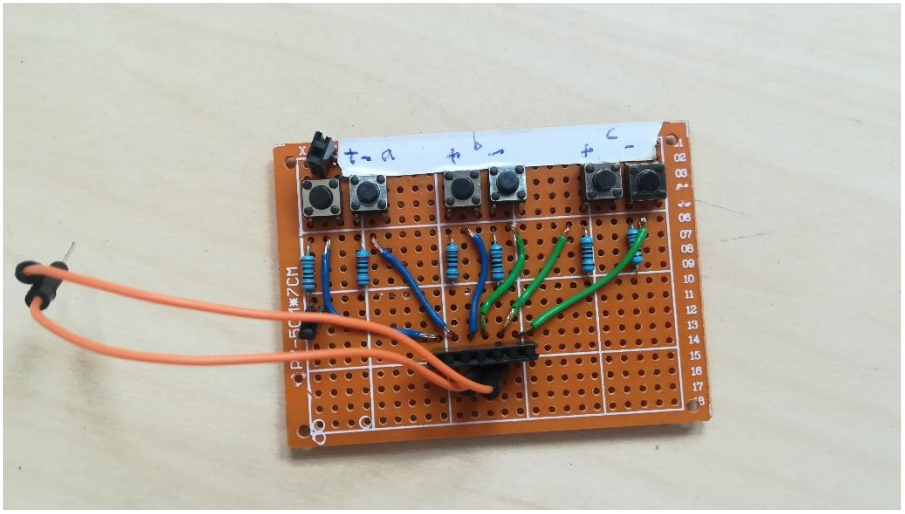


Imagen 79. Resultado final de la botonera

Una vez construida la placa de la botonera y conectando esto a la protoboard falta crear un código que permita variar los valores de las funciones NoteOn y NoteOff para ver cómo afectan estos a los mensajes MIDI y, por ende, a los sonidos. El código que se ha creado para este propósito se encuentra en los anexos, pero se incluye una parte a continuación. El código solo permite aumentar y disminuir los valores de a,b y c con los botones a la vez que los muestra por la pantalla. Así se pueden ver las variaciones de sonidos cada vez que se de un golpe.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

void noteOn(byte channel, byte pitch, byte velocity) {
  MIDIEvent noteOn = {0x09, 0x90 | channel, pitch, velocity};
  MIDIUSB.write(noteOn);
}

void noteOff(byte channel, byte pitch, byte velocity) {
  MIDIEvent noteOff = {0x08, 0x80 | channel, pitch, velocity};
  MIDIUSB.write(noteOff);
}

void controlChange(byte channel, byte control, byte value) {
  MIDIEvent event = {0x0B, 0xB0 | channel, control, value};
  MIDIUSB.write(event);
}
```

```
int sensor=A0;
int value;
int a=0;
int b=40;
int c=64;
bool upa;
bool downa;
bool upb;
.
.
.
```

El montaje completo funcionó como se esperaba. Las conclusiones que se han visto al variar los valores de a, b y c es que solo el valor de b afecta. Cada valor de b da un sonido diferente. Por ejemplo, si b=7 el sonido es el de un charles muy cerrado, si b=8 sigue siendo un charles pero algo más abierto, y si b=9 el sonido corresponde a golpear un charles en la campana. Viendo cómo funcionan NoteOn y NoteOff en función del valor de b se puede realizar una tabla donde se especifique el valor de b, donde se golpea y una breve descripción del golpe. Esto puede ayudar a la hora de codificar y elegir qué valor de b debemos mandar en función de la fuerza con la que se golpee. Se incluye a continuación una parte de la tabla para que se vea el contenido, pero el contenido completo se incluirá en los anexos.

B	LUGAR	DESCRIPCIÓN
4	Ride	Pequeño crasheo (entre ritmo y crasheo)
6	Caja	Golpe normal de caja, algo bajo (quizá no para ritmos)
7	Charles	Golpe normal muy cerrado
8	Charles	Igual pero algo menos cerrado
9	Charles	Cerrado en la campana
36	Bombo	Golpe normal y corriente
37	Caja	Golpe de aro (rimshot)
38	Caja	Golpe normal, algo flojo
39	Caja	Golpe de aro, prácticamente igual al otro
40	Caja	Golpe normal
41	Caja	Golpe solo de aro
42	Caja	Cross
43	Caja	Golpe algo flojo pero menos seco
44	Caja	Golpe al aro pero sin rimshot
…	…	…

5 Desarrollo e implementación de los nodos del sistema

5.1 Descripción de nodo y microcontrolador usado en ellos

Antes de hablar del microcontrolador que se ha usado en esta parte del sistema debemos aclarar el concepto de “nodo” en este sistema.

La primera idea para hacer llegar las señales desde los sensores piezoeléctricos hasta el Arduino que integrara la función USB-HID era cablear el sensor directamente hasta el microcontrolador, pero esto planteaba tres problemas de solución complicada:

- Los sensores piezoeléctricos sufren mucho cuando se usan cables muy largos, ya que una longitud considerable de cable puede llegar a tener un carácter capacitivo, lo que sería un serio enemigo de la señal producida por estos sensores (el voltaje producido por el sensor se perdería y no tendríamos señal a la entrada del ADC del microcontrolador). Para solucionar esto supondría incluir acondicionamiento en cada sensor como seguidores de tensión y amplificadores para aumentar la señal lo que supondría un trabajo importante.
- Esta solución suponía que únicamente un microcontrolador (el Arduino Pro Micro) debería hacer todo el cálculo de todo el sistema (muestreo, análisis de señal, codificación según golpeo y envío por USB). Esto podría llegar a ser un problema debido al muestreo de tantas partes (tom, tom base, la caja, el bombo, el charles y dos platos).
- Ninguna de las dos opciones que actualmente ofrece Arduino compatible con USB-HID ofrece entradas analógicas suficientes para incluir todas las partes que tiene una batería. Una posible solución sería usar multiplexores analógicos, pero esto haría descender bruscamente la velocidad máxima a la que se podría muestrear la señal de cada parte.

Viendo estos problemas se planteó la idea de usar una distribución de unas placas o nodos, como se decidió denominar finalmente, que pudieran comunicarse con el Arduino central (mediante protocolos establecidos como el I2C), que eliminara parte del cálculo que debía realizar en un origen el Arduino central y que, además, se encontraran en cada parte de la batería para que los cables que comunican este nodo con el sensor piezoeléctrico fueran lo más cortos posibles.

El primer prototipo que se ideó para estos nodos fue el uso de un Arduino Nano, el cuál es más barato que el Arduino Pro Micro y usa un conector Mini-USB, por lo que puede programarse de forma cómoda sin necesidad de usar programadores como en el Arduino Mini Pro. Se eligió este microcontrolador porque dispone de las mismas especificaciones que el Arduino Pro Micro, lo que permite muestrear de la misma forma que con este, pero sin disponer de USB-HID aunque esto no es necesario para este uso. Este microcontrolador recogería la señal de los piezoeléctricos que se usaran en cada parte (una caja contiene dos, un plato tres, un tom uno, etc) usando el mismo circuito de acondicionamiento que se ha usado en las pruebas anteriores y mandaría un mensaje por I2C al Arduino Pro Micro diciendo que tipo de golpe se ha dado.

Para esto se planteó el siguiente esquemático:

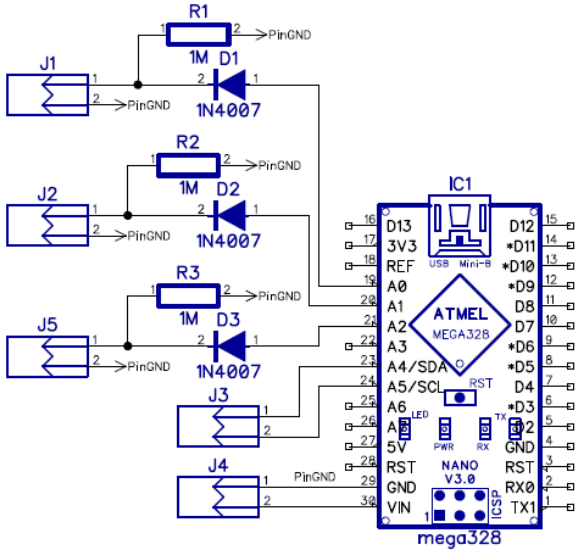
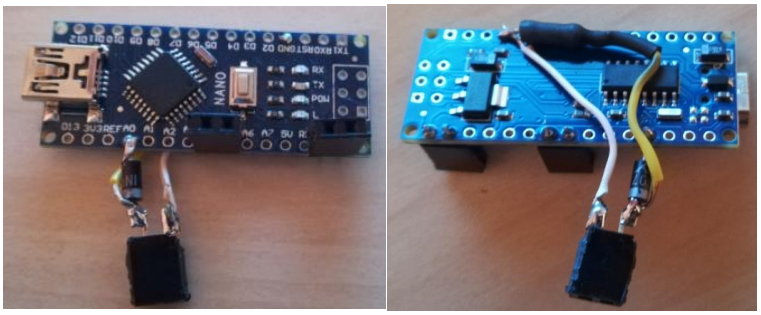


Ilustración 80. Esquemático Prototipo Nodo

En este esquemático se pueden ver tres entradas analógicas del Arduino donde se han conectado un circuito de acondicionamiento como el que se había comentado anteriormente. El resto de conexiones consiste en el uso de los pines A4 y A5 que son los responsables de la transmisión de datos mediante el bus I2C (SDA y SCL) y los pines de alimentación Vin y GND que permitan el funcionamiento del microcontrolador sin necesidad del uso del Mini-USB.

El primer prototipo que se fabricó fue simplemente conectando los componentes sobre el mismo microcontrolador mediante cable y soldadura, sin ningún tipo de PCB. Esto se hizo porque queríamos probar desde el principio el prototipo usando cables cortos en los sensores piezoeléctricos. Se crearon varias placas dependiendo de si la parte necesitaba un, dos o tres sensores, y el resultado de una de estas placas fue el siguiente:



Imágenes 81 y 82. Fotografías del primer prototipo de nodo

5.2 Pruebas de conexión sobre I2C

Una vez que SE decidió usar el protocolo I2C como comunicación en nuestro sistema se realizaron una sería de pruebas para ver cómo implementar esta comunicación en nuestro proyecto. Basándonos en los ejemplos que aporta el IDE de Arduino se realizó el siguiente montaje:

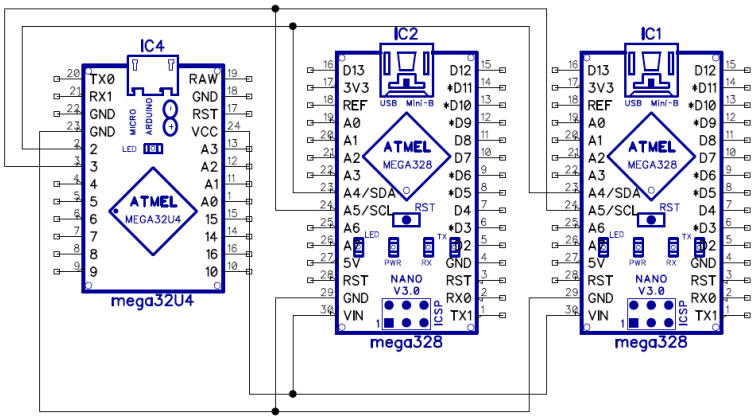


Imagen 83. Esquemático de la prueba de I2C

En este circuito simplemente se conectan los terminales correspondientes a SDA y SCL de cada dispositivo entre sí, al igual que la alimentación y la masa. Los tres circuitos serán alimentados por la tensión que suministra el USB al Arduino Pro Micro, la cual se repartirá entre los tres dispositivos. A continuación se montó todo en una protoboard:

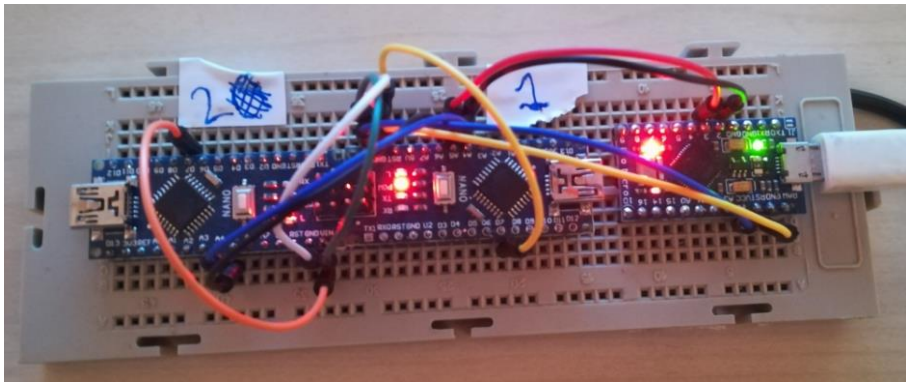


Imagen 84. Fotografía del montaje de prueba de I2C en protoboard

La idea es usar un cable en cada Arduino Nano (esclavos) para llevar la señal del pin digital D7 a masa o a VCC, con el objetivo de que la entrada detecte ON u OFF. Esto se enviará mediante I2C al Arduino Pro Micro (maestro), el cual deberá mostrar por el monitor serie el estado de estas salidas.

Para esto se usaron dos códigos distintos. A continuación se adjunta el código cargado en los esclavos. La única diferencia entre ambos que es uno tiene una dirección de esclavo (1) y el otro tendrá otra totalmente distinta (2 en este caso).

```
#include <Wire.h>

void setup() {
  pinMode(7, INPUT);
  Wire.begin(2);                // join i2c bus with address #8
  Wire.onRequest(requestEvent); // register event
}

void loop() {
```

```
  delay(100);
}

void requestEvent() {
  if(digitalRead(7)==LOW){
    Wire.write("a");
  } else (Wire.write("b"));
}
```

El código para el maestro es el que se adjunta a continuación. En este caso se pide a cada esclavo que se le envíen datos, los cuales corresponden al estado de la salida D7 de cada uno. Finalmente estos se muestran por el monitor serie.

```
#include <Wire.h>

char c;

void setup() {
  Wire.begin();        // join i2c bus (address optional for master)
  Serial.begin(9600);  // start serial for output
}

void loop() {
  for(int i=1;i<3; i++){
  Wire.requestFrom(i, sizeof(char));    // request 6 bytes from slave device
#8
  while (Wire.available()) { // slave may send less than requested
    c = Wire.read();// receive a byte as character
  }
   Serial.print("Ent Ard ");
  Serial.print(i);
  Serial.print(" esta en ");
    Serial.println(c);

 }
  delay(100);
}
```

Con el código una vez cargado y habiendo entendido el funcionamiento de las comunicaciones en Arduino se procedió a comprobar la salida del sistema en el monitor serie:

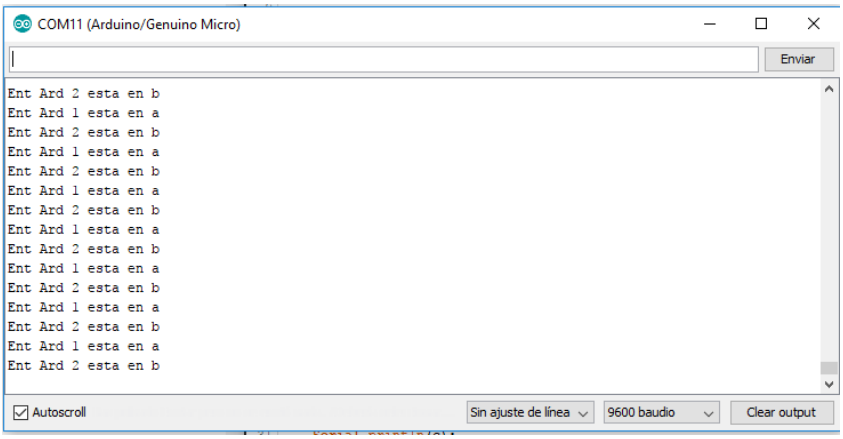


Imagen 85. Salida del monitor serie en las pruebas de comunicación I2C

En el monitor se podía ver como el estado de las salidas cambiaba en función de si llevábamos la entrada D7 a 5v o a GND, tal y cómo nos habíamos propuesto.

6 Integración Hardware/Software del sistema

6.1 Hardware para el montaje de sensores

Aunque hasta ahora solo se ha hablado del funcionamiento de los sensores y de las pruebas que se han realizado para procesar la señal y enviarla es importante señalar también como se han montado sobre cada parte de la batería.

El principal problema era situar en el centro de cada casco el sensor piezoeléctrico de una forma que pudiera sostenerse de forma firme, segura y con cierto ajuste. Además también hay que aclarar la construcción de los conos que se encuentran en la parte superior de cada sensor, cosa que se mencionó en los primeros apartados.

La primera parte que se explicará en detalle será la del acondicionamiento mediante los conos de gomaespuma. Está idea está basada en los productos de gama alta de Roland. Cuando se vio que la señal proporcionada por el piezoeléctrico era demasiado elevada cuando este se posicionaba directamente contra el parche surgió la idea de usar este método para ver su resultado y, efectivamente, funcionó mejor de lo previsto.

Para la construcción de estos conos lo primero fue buscar un material que fuera el adecuado. No se puede elegir una gomaespuma cualquiera, ya que tiene que reunir unas condiciones concretas:

- Dureza media: no puede deformarse muy fácilmente pero tampoco puede crear elevaciones en el parche cuando este se apriete encima del cono
- Facilidad para el moldeo.
- Precio reducido

Consultando en diversos foros se llegó a la conclusión de que una espuma valida podría ser la usada en los tacos de lija de dureza media. Esta se deforma para adaptarse a las curvas de la madera, pero a la vez tiene la consistencia necesaria para eliminar material sin desgranarse. Además el precio es muy bajo (sobre 75 céntimos por bloque).

Una vez que se tenga el bloque solo se tendrá que eliminar la capa de lija (con una pequeña cuchilla y mucho cuidado) y cortar el máximo número de cilindros de 35 mm de diámetro (diámetro de los piezoeléctricos grandes). Normalmente de un bloque suelen salir tres conos ya que la parte superior no es necesario que sea de el mismo diámetro que la inferior.

No hace falta realizar un corte muy preciso, pero sí que es importante que no sea menor del diámetro inferior del cono. Estas piezas luego se pegarán con adhesivo en spray para poder juntar las dos piezas de gomaespuma y tener una única pieza de gomaespuma que podamos moldear con el taladro y la cuchilla.

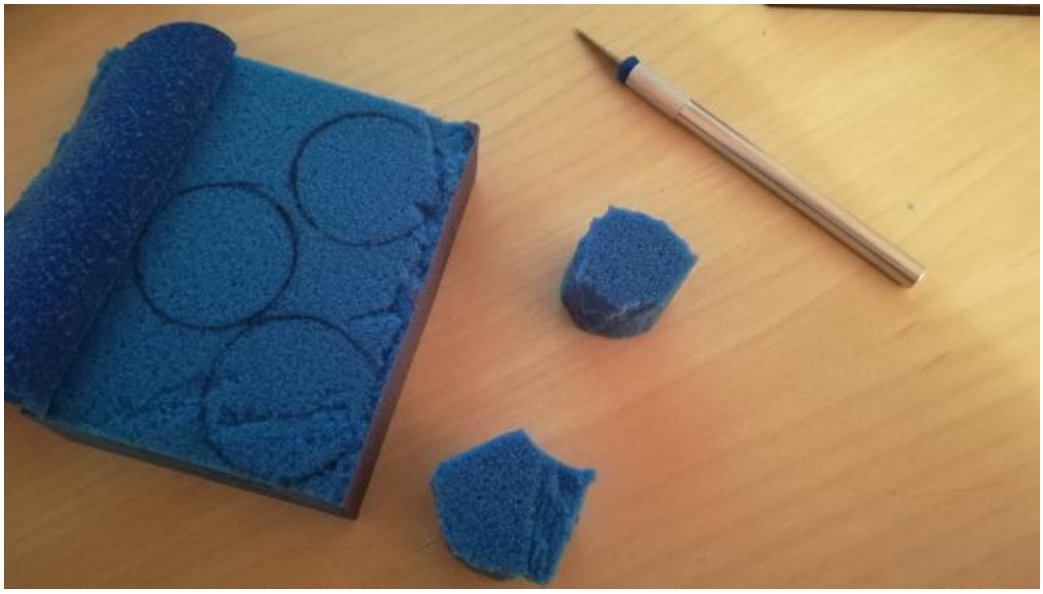


Imagen 86. Muestra de la eliminación de la capa de lija y los cortes que se realizan

A partir de aquí solo deberemos pegar uno de los trozos pequeños sobre uno de los trozos grandes y darles forma. Para pegar esta gomaespuma se ha usado adhesivo en spray especial para estos materiales. Para poder darle forma se ha diseñado una pieza que permite pegar el material a una cara y montarlo a su vez en un taladro para darle forma.

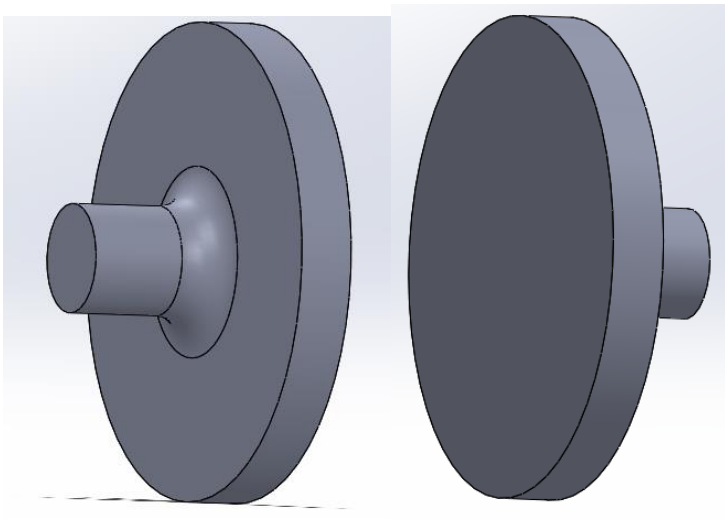


Imagen 87 y 88. Detalles de la pieza diseñada para dar forma a los conos

Pegaremos a esta pieza la que hemos conseguido pegando dos trozos de gomaespuma con cinta de doble cara, y montaremos el eje en un taladro, cuidando que el giro sea lo más concéntrico posible. A partir de aquí solo tendremos que darle forma a la gomaespuma. En este proceso se usó primero una cuchilla a modo de herramienta de torno para eliminar la gomaespuma sobrante y finalmente un papel de lija fino para mejorar el acabado.

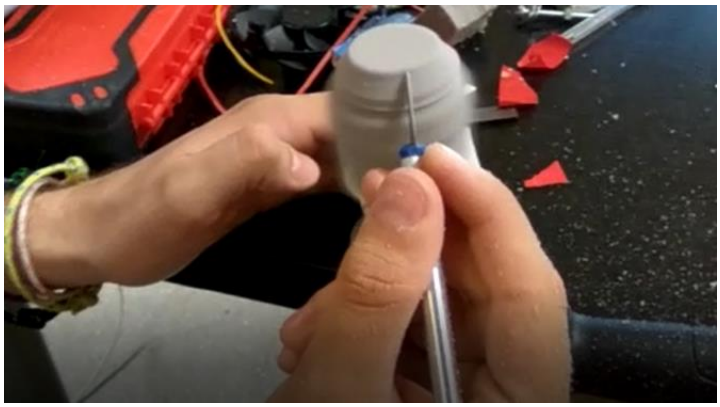


Imagen 89 y 90. Detalle del montaje del taladro y captura de cómo se da forma a la gomaespuma

Las medidas usadas para los conos han sido extraídas de internet y son las mismas que usa Roland en sus modelos. También se ha usado este método para montar los piezoeléctricos en los platos, pero en vez de conos se han hecho cilindros con un diámetro de 27 mm (la medida de los sensores pequeños) y 10 mm de altos.

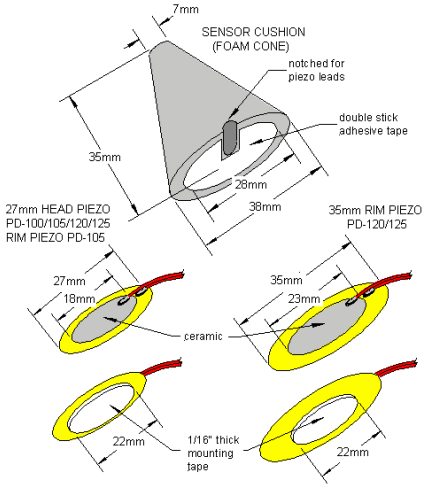
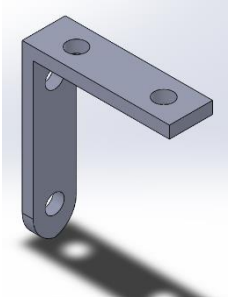


Imagen 91. Detalle de las medidas de los conos

Una vez terminado el cono faltaría adherirlo al sensor piezoeléctrico. Es conveniente hacer una pequeña marca o canal en la parte inferior del cono para que el cono quede bien pegado. Se ha usado cinta de doble cara de cierto grosor y esponjosa que permita trasladar las vibraciones al sensor.

Imágenes 92 y 93. Cinta usada y detalle del cono pegado al sensor

Una vez terminado el sensor con el acondicionamiento mecánico falta ver como posicionarlo en el centro de cada parche. Para esto se ideó un sistema que permitiera fijar un perfil de aluminio que cruzara el casco de cada tom, caja o bombo. Lo primero fue diseñar una serie de piezas que eran necesarias para fijar el perfil al casco o el sensor al mismo perfil:



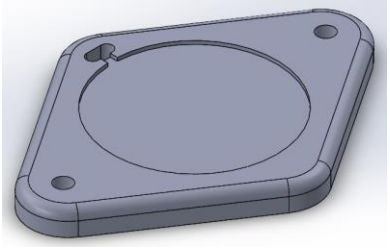


Imagen 94 y 95. Capturas del diseño de las escuadras y los soportes para los sensores

Estas piezas fueron finalmente impresas en 3D. Para poder montar los perfiles de aluminio se aseguraron dos escuadras a los agujeros de dos torres opuestas con tornillos de M4 y una longitud de 20 mm. Una vez esto se cortó un perfil de aproximadamente 2 cm menos que la diagonal del casco y, montándolo sobre las escuadras, se marcaron los agujeros laterales (donde se unen a las escuadras) y dos centrales (donde se montan los soportes de los sensores, por lo tanto la medida de los agujeros y la distancia es la misma). Una vez hecho esto se usaron tornillos de M5 para unir las escuadras a los perfiles (con muelle entre la escuadra y el perfil) y tornillos de M3 para unir el soporte al perfil (con muelle también). El uso de muelles nos permite regular la altura del cono y, por tanto, la presión que este realiza sobre el cono. Por último introducimos los cables del sensor por el agujero existente y pegamos la parte plana y dorada del sensor en la hendidura del soporte con cinta de doble cara. Este proceso se repitió en todos los cascos de la batería. Un ejemplo sería el siguiente:

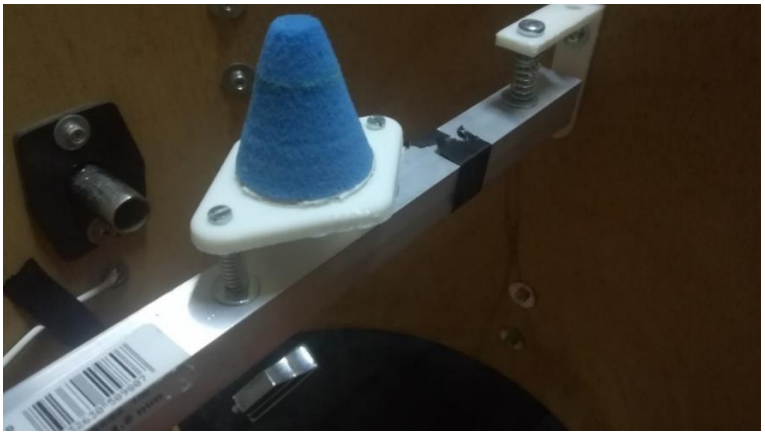


Imagen 96. Ejemplo del montaje del sensor en el casco

Lo siguiente sería montar los sensores en los platos. Para ello simplemente se pegarán los piezoeléctricos a la zona deseada del plato incluyendo entre el sensor y el plato la pieza de gomaespuma descrita anteriormente con cinta de doble cara.

Imagen 97. Montaje de los sensores en los platos (ejemplo ride con 3 sensores)

Por último faltaría montar el sensor del pedal del charles, el cual marcará como está de abierto y así ver que sonido se mandará dentro de las posibilidades de la tabla de sonidos. Para monitorizar la apertura se decidió usar un potenciómetro lineal deslizante, el cuál actuará como divisor de tensión, dando un valor entre 0 y 5 para indicar el grado de apertura del soporte. Para montarlo se hizo un pequeño soporte usando un sobrante de perfil de aluminio y aprovechando los agujeros del mismo soporte. Con este soporte podemos accionar el potenciómetro con el pedal y, por tanto, tener siempre una señal que indica la apertura.

Imágenes 98 y 99. Detalles del soporte y detalles del montaje

6.2 Desarrollo de los nodos finales

Aunque se ha explicado en un apartado anterior como se llegó al desarrollo del prototipo de nodos con el que se han hecho las pruebas de conexión y ciertas pruebas finales se llegó a la conclusión de que, si planteábamos este proyecto como el desarrollo de un producto que tuviera que llegar a un consumidor, la idea en la que habíamos pensado no era la más idónea por las siguientes razones:

- El hardware usado en cada parte es distinto: un nodo de un plato tendrá solo una entrada, uno de caja dos, uno de un ride tres o incluso el del charles usará otra entrada analógica que ninguno de los anteriores necesita. Esto supone que se pierde la intercambiabilidad entre los nodos de distintas partes, lo que es un obstáculo a la hora de conseguir que el uso del dispositivo sea fácil
- En estos prototipos se dependía mucho del software de cada uno. Está claro que el código cargado entre el nodo de un tom y de una caja tiene que ser diferente, pero lo que se pretende es que se dependa lo mínimo posible del software en el nodo
- Por último, en el prototipo anterior el reemplazo de piezas era muy difícil. Si el Arduino se rompía había que desoldar todos los componentes prácticamente, además de que el montaje no se podía realizar de forma sistemática, ya que los componentes se soldaban sobre la propia placa.

Analizando estos tres problemas se llegó a la conclusión de que la solución que se implantara debería solucionar estos tres problemas y además ser genérica. Con genérica nos referimos a una solución que, aunque se varíe el software de una parte a otra (por ejemplo de una caja a un plato), el hardware deberá ser el mismo para todas las partes.

Viendo estas razones se pensó en crear un diseño que intentará solucionar estos problemas pero que a la vez no volviera a recaer en los que se solucionaron con el anterior prototipo. Lo primero era crear una placa que fuera universal para cualquier parte de la batería, por lo que debía integrar como mínimo tres entradas acondicionadas además de otra entrada analógica donde conectar el potenciómetro del charles.

Además la idea era poder seleccionar la dirección I2C del dispositivo mediante la colocación de unos jumpers o interruptores pequeños, pudiendo así tener más posibilidades de elegir la dirección que no sean solo mediante software. Por último, se planteó la idea de que la placa fuera de tipo shield, lo que permitiría la sustitución fácil del microcontrolador si hubiese algún problema con este.

Algo que se pretende mantener en relación al prototipo anterior es el tamaño reducido de la placa. En el primer prototipo los componentes incrementaban el tamaño de la placa de forma ínfima. Sabemos que en este segundo prototipo el tamaño deberá aumentar, pero la intención es mantener el menor tamaño posible para evitar problemas de peso con las conexiones.

Viendo las funcionalidades que deberíamos añadir a los nodos se optó por la realización de un diagrama de bloques para poder ver las funciones añadidas de forma más directa y, además, nos permite organizar las mejoras para poder implementarse de forma ordenada a la hora de diseñar la placa.

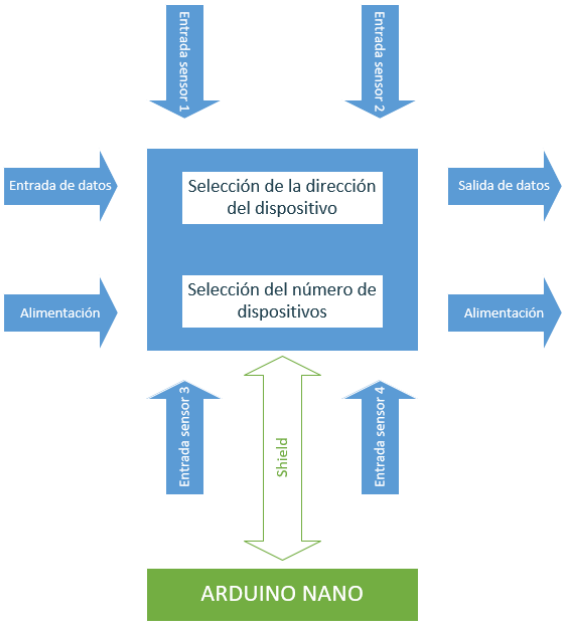


Imagen 100. Diagrama de bloques del segundo prototipo de los nodos

Viendo estas ideas reflejados en el diagrama de bloques se procedió a diseñar la PCB en Diptrace usando la librería de componentes y patterns del profesor del Departamento de Tecnología Electrónica Pedro Díaz Hernández. Lo primero fue la realización del esquemático, donde lo primordial era conectar todos los componentes de forma correcta pero de la manera más clara posible, usando conexiones entre Nets sin cable para evitar el desorden. El resultado final del esquemático fue el siguiente:

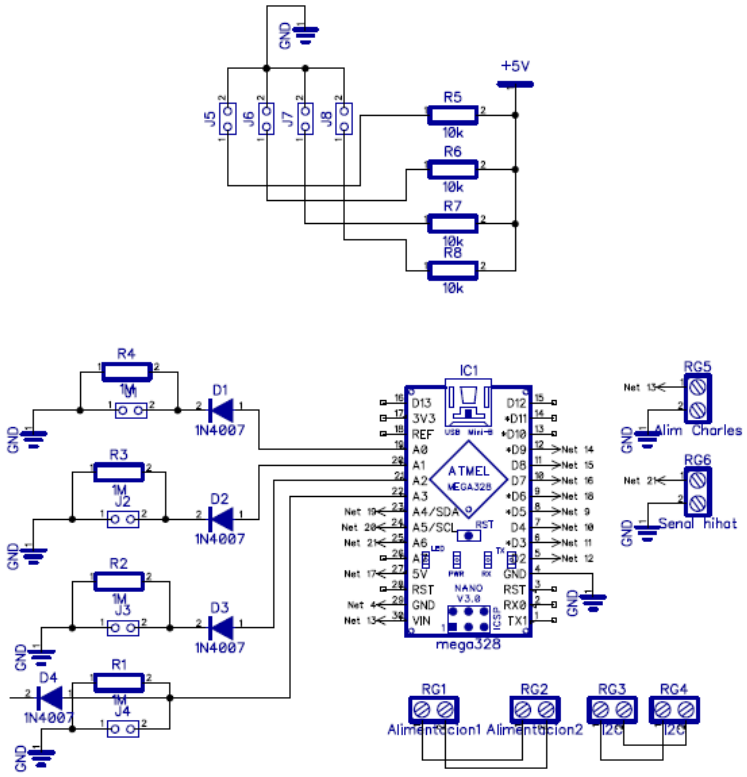


Imagen 101. Esquemático del segundo prototipo de los nodos

El este esquemático se puede ver la sección de jumpers donde seleccionaríamos la dirección del nodo, dos entradas a cada lado para sensores piezoeléctricos, lo que permite conectarlos de forma más cómoda si un lado de la placa es más accesible que el otro y, por último, las entradas de alimentación y de los cables SDA y SCL del bus. Además se ha añadido una entrada dedicada exclusivamente al sensor del charles con la alimentación necesaria para el potenciómetro.

El siguiente paso sería la fabricación física de la placa. Aunque existe la posibilidad de realizar este montaje en placa perforada se pensó que sería una buena idea realizar el diseño de enrutado de pistas por si fuera posible acceder al equipamiento necesario para poder realizar la placa con insoladora y atacado químico. Para esto se hizo uso de la herramienta disponible en Diptrace, la cual nos permite pasar desde el esquemático al diseño de PCB manteniendo los

componentes y las uniones entre ellos. Se optó por diseñar la placa en doble cara para minimizar el espacio, aunque se usarán componentes estándar o Trough Hole (se ha desechado la opción de SMD debido a la dificultad de obtener los componentes y del soldarlos). Una vez realizado el enrutado de las pistas el resultado fue el que se muestra a continuación:

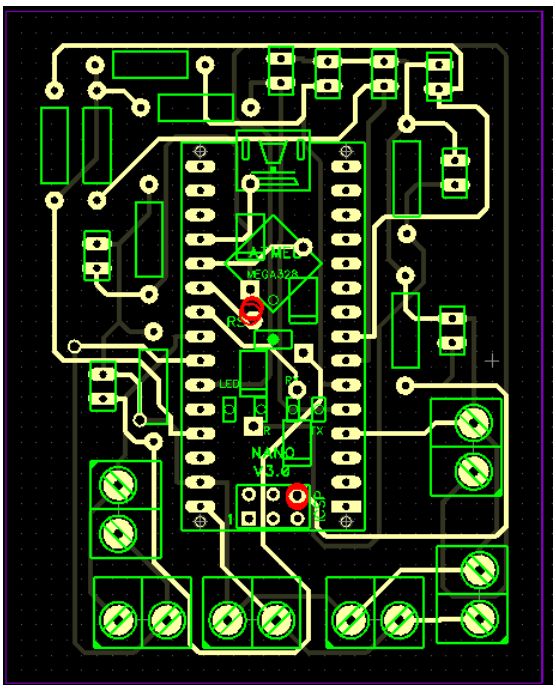


Imagen 102. Diseño de la PCB del segundo prototipo de los nodos

6.3 Cableado del sistema

Una vez establecido el sistema falta la comunicación entre ellos. Es por esto que, aunque en la placa se hayan instalado regletas con terminales, la idea es usar conectores estándar para que la conexión entre nodos sea lo más limpia posible y fácil de realizar. En principio se deben llevar desde un dispositivo a otro la alimentación y el bus I2C. Esto conlleva usar cuatro cables o hilos, ya la masa es compartida entre la alimentación y el bus. La idea que surgió a la hora de cablear cada nodo fue usar una conexión de tipo Daisy Chain. De esta forma no necesitamos que haya un cable desde cada nodo hasta el Arduino Pro Micro. Usando la configuración Daisy Chain solo necesitamos un cable que vaya desde un nodo hasta el siguiente, por lo que la cantidad de cable usada es mucho menor.

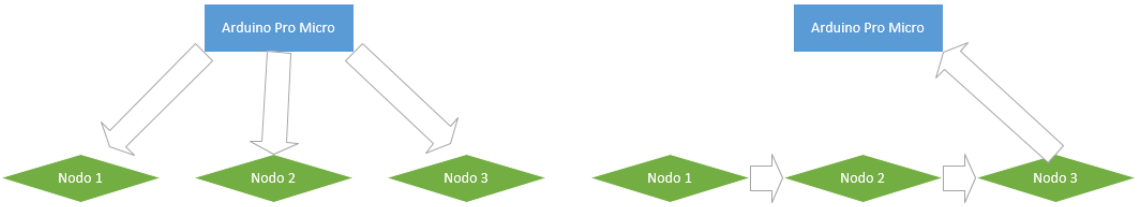


Imagen 103. Distribución estándar frente a distribución Daisy Chain

Una vez decidida la distribución se pensó también en distribuir de esta forma la alimentación, para intentar que se tenga que usar la menor cantidad de cables y la menor longitud posible. A partir de aquí faltaba decidir qué tipo de conector se usaría. Hay que aclarar que aunque se decida instalar un cable con un conector específico los nodos están preparados para instalar cualquier tipo de cable siempre que disponga de los terminales necesarios. Cambiar de uno a otro solo depende de instalarlos en los terminales correspondientes.

La primera idea fue usar conectores de tipo Jack de 3,5 mm con cuatro terminales. Estos conectores suelen usarse en video debido a que permiten enviar por el mismo cable audio estéreo y señal de video. También son muy usados en auriculares ya que permiten enviar audio estéreo y señal de micrófono. Además los terminales no son excesivamente grandes y son fáciles de manipular y soldar.

Imagen 104. Ejemplo de conector Jack 3,5 mm de cuatro terminales

El problema de estos conectores es que su disponibilidad es muy reducida. Solo se han podido encontrar conectores machos de tipo aéreo (nada de hembras, ya sea de chasis o aéreas). La siguiente opción fue pensar en usar conectores XLR, muy extendidos también en el ámbito de la industria musical (la mayoría de conexiones entre micrófonos y mesas de mezclas o altavoces se realizan con estos conectores).

Imagen 105. Ejemplo de conector XLR de 4 pines

El principal problema de estos conectores vuelve a ser la dificultad para conseguirlos, además de que los de cuatro pines no están tan extendidos y el coste es bastante superior a los Jack. En este momento se decidió que la posibilidad de aunar los 4 hilos necesarios en un solo conector era muy difícil si queríamos usar conectores estándar de la industria.

Viendo las dificultades surgidas se consideró la opción de separar el único cable original en dos cables, uno para alimentación y otro para el bus I2C. De esta forma se podrían usar conectores Jack estándar estéreo, fáciles de conseguir y de precio asequible. El único problema es que tendremos en cada nodo cuatro hembras: una entrada de alimentación, una entrada de bus, una salida de alimentación y una salida de bus. Finalmente nos decidimos por esta opción ya que los cables estéreo macho-macho de 3,5 mm se pueden encontrar muy fácilmente, aunque los nodos están preparados para usar otro tipo de conectores en el futuro.

El siguiente paso sería diseñar y construir los cables que irían desde los nodos hasta la salida de cada parte de la batería con sus respectivas hembras. Se ha optado por usar un cable de 4 hilos desde el nodo hasta el exterior del casco, donde se puentean las clavijas de dos en dos. El siguiente esquemático aclara el montaje:

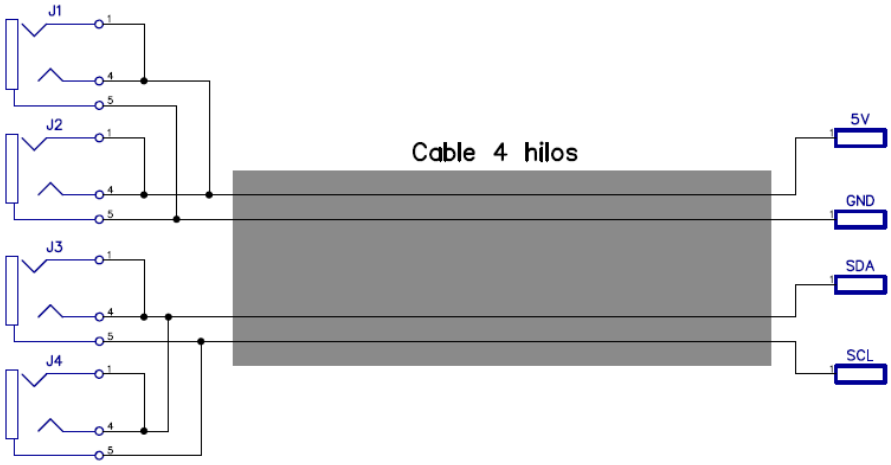


Imagen 106. Esquemático de los cables hembra que entran a los nodos

Se optó por usar conectores estéreo en lugar de mono debido a que al comprar varios se conseguía mejor precio, y con puentear los dos terminales que no van a masa se consigue el mismo funcionamiento que si se instala un conector mono.

Finalmente, el cable de cuatro hilos en un cable normal y corriente, sin apantallamiento ni propiedades especiales. El resultado final sería el siguiente, aunque en la imagen aparecen instalados terminales de pinera macho porque se usaban en el primer prototipo.

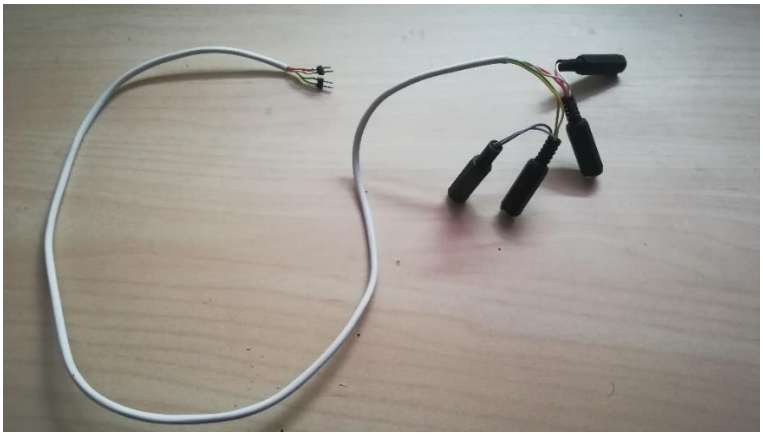


Imagen 107. Resultado final de los cables hembra para los nodos

Una vez diseñados y construidos los terminales hembras para los conectores simplemente haría falta diseñar los cables de conexión macho-macho. Ya que hemos decido usar dos conectores estéreo como entrada (bus y alimentación) y otros dos como salida necesitaremos crear cables que dispongan de dos machos estéreo en cada extremo. Hay que mencionar que se podrían usar dos cables macho-macho normales que se encuentran en cualquier tienda, pero preferimos intentar hacer un diseño que nos permitiera aunar la alimentación y el bus en un solo cable.

El diseño es muy similar al de los cables hembra, ya que necesitamos de la realización de puentes entre los dos terminales que no son masa. El esquemático realizado para estos cables es el siguiente:

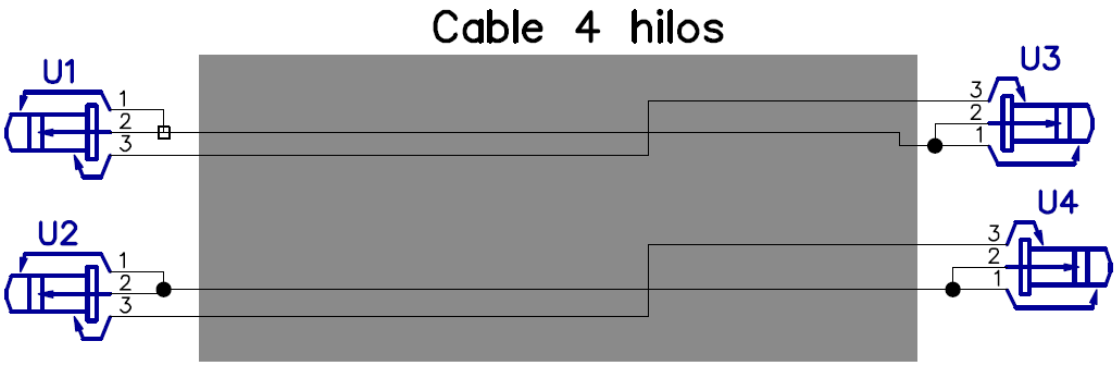


Imagen 108. Esquemático de conexión de los cables macho-macho

Con el esquemático ya claro lo siguiente era fabricar los cables. En este caso se optaron por clavijas macho acodadas para los toms, ya que la conexión es más cómoda en estas situaciones. En los platos se usaron clavijas macho rectas, aunque esto es intercambiable sin ningún tipo de problema, la conexión de los cables sigue siendo la misma.

Imagen 109. Resultado final de un cable macho-macho

6.4 Base y alimentación del Arduino central

Una vez resueltos todos los problemas de comunicación entre los nodos debemos diseñar el circuito donde nuestro Arduino Pro Micro se encontrara asentado. El acondicionamiento que necesita el microcontrolador es mínimo, simplemente las conexiones al bus I2C, sin embargo, en la placa donde lo instalemos se encontrará también la alimentación general de todos los nodos.

Cabe destacar que el primer prototipo se realizó en placa perforada y fue usado para probar el sistema con los primero prototipos de nodos. En este caso el sistema se alimentaba con una fuente de laboratorio de voltaje regulable. El circuito era el siguiente:

Imágenes 110 y 111. Fotografías del prototipo de la base del Arduino Pro Micro

Para el diseño final seguiremos las mismas etapas que en todos los anteriores, por lo que lo primero consistirá en realizar el esquemático de esta parte. En este

prototipo final vamos a incluir la clavija de alimentación ya que el sistema se piensa alimentar con un alimentador de pared y un sistema de jumpers para poder elegir el número de nodos que habrá en el sistema independientemente del software.

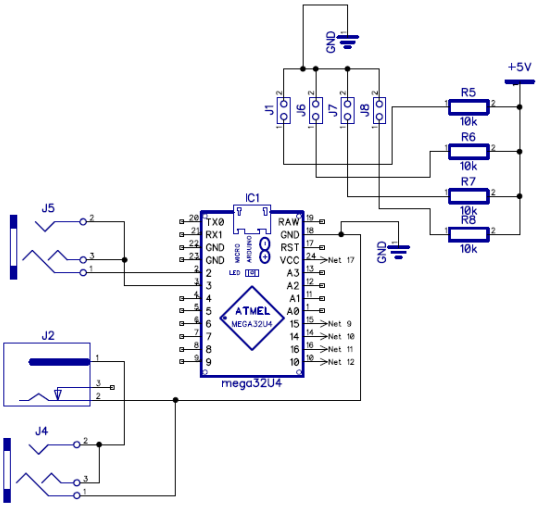


Imagen 112. Esquemático del último prototipo de base

En el esquemático se puede ver la clavija hembra de alimentación que alimentará a todo el sistema de nodos, pero no al propio Arduino Pro Micro. Esto es debido a que cuando se intentó alimentar a todos los microprocesadores con la misma alimentación (la fuente en el prototipo anterior) aparecían errores en el USB, así que se optó por alimentar el Arduino Pro Micro solo con la tensión del USB, aunque se tienen que compartir las tierras para el correcto funcionamiento del bus I2C. La clavija de Jack superior sirve para la conexión del bus con alguno de los nodos y la inferior para la salida de alimentación para dichos nodos. Hecho esto se optó también por hacer una PCB con el enrutado de las pistas por si se pudiera acceder al equipamiento necesario para construirla.

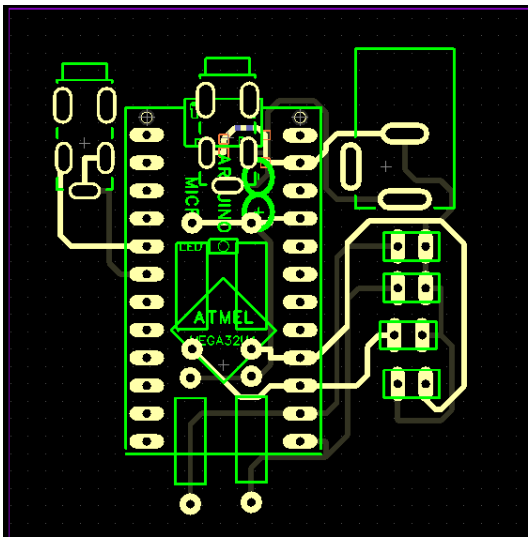


Imagen 113. PCB del prototipo de la base

Se puede ver que el conector utilizado para la alimentación es el típico conector de alimentador de pared, ya que usando uno de estos dispositivos eliminamos el tema de la rectificación y filtrado de la corriente alterna además de los ruidos indeseados en las señales que la corriente alterna puede provocar. Además en las pruebas que se hicieron con la fuente de alimentación regulada se observó una intensidad consumida de 1 A, por lo que se usará una de 2A sobredimensionando.

Imagen 114. Alimentador de pared válido para nuestro proyecto (5V-2A)

6.5 Programación de los nodos

Después de ver el desarrollo de todo el hardware necesario para hacer el sistema funcionar ahora debemos hablar del software desarrollado para que el funcionamiento sea el correcto además de conseguir el mejor funcionamiento posible. Se ha mostrado ya el funcionamiento general del sistema, desde la entrada analógica hasta la salida en forma de sonido por el ordenador. En el siguiente esquema se resume el funcionamiento de forma muy general:

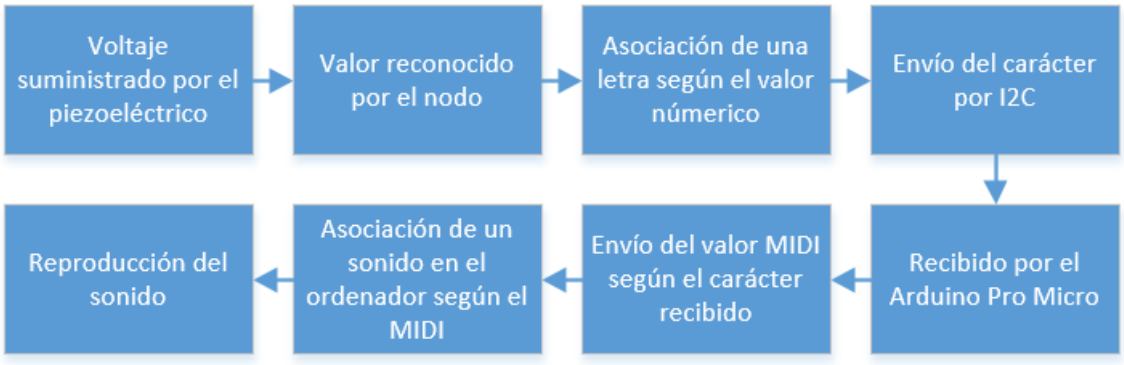


Imagen 115. Esquema del funcionamiento general

Viendo este resumen, en este apartado se hablará en detalle desde que el valor es reconocido por el nodo hasta que la información es enviada al Arduino central. Aunque parezca una tarea trivial la de codificar los valores reconocidos, lo cierto

es que esta parte conlleva un acondicionamiento digital de la señal (aparte del acondicionamiento analógico y mecánico ya hecho) y además la codificación se hace en función de numerosas condiciones diferentes para cada parte de la misma batería.

El software de los nodos se estructurará en tres partes principalmente:

- Reconocimiento de la señal
- Codificación en función del golpe recibido
- Envío

Para estas funciones nos hemos basado en las pruebas realizadas en apartados anteriores de este documento. Aquí se mostrará el código de una de las partes como ejemplo del funcionamiento y desarrollo que se ha llevado a cabo, pero el resto de software se adjuntará en la sección de anexos.

La primera versión de este software fue la que se desarrolló para los primeros prototipos que se realizaron para este sistema. En estos la asignación de la dirección I2C y el número de dispositivos en la red se debían asignar por código. El siguiente ejemplo corresponde al código escrito para la caja, donde se usaban dos sensores (casco y parche):

```
#include <Wire.h>

int ent0=A0;
int ent1=A1;
int valorent0; //sensor del parche
int valorent1;// sensor del casco

void setup() {
  Wire.begin(2); //Dirección 2 corresponde a la caja
  Wire.onRequest(respuesta);
}

void loop() {
 valorent0=analogRead(ent0);
 valorent1=analogRead(ent1);
delay(25);
}
void respuesta(){
  if ((valorent1>valorent0) && (valorent1>80)){
    Wire.write("w"); //mandamos señal cross
  } else if((valorent0>200) && (valorent1>100)){
    Wire.write("e");// rimshot
  } else if(valorent1<100){
    if(valorent0>60){
      Wire.write("r");//golpe flojo
      } else if ((valorent0<150) && (valorent0>60)){
      Wire.write("t");//golpe medio-flojo
    } else if ((valorent0<300) && (valorent0>150)){
      Wire.write("y");//golpe medio
    } else if (valorent0>300){
      Wire.write("u");//golpe fuerte
    }
  } else (Wire.write("m"));
```

```
}
```

Se puede ver el funcionamiento de forma muy fácil: se toman valores de los sensores cada 25 milisegundos, y cada vez que el maestro solicita información se llama a la función respuesta: esta, en función de las comparaciones que hemos establecido, enviará un carácter que luego el Arduino central traducirá en una señal MIDI. Cabe mencionar que los valores con los que se realiza la comparación han sido establecidos mediante experimentos, realizando prueba y error hasta conseguir la mejor respuesta posible, al igual que la frecuencia de muestreo.

Sin embargo con este software han aparecido dos problemas principalmente:

- El primero son los dobles golpes que aparecen dependiendo de la fuerza con la que se toque. Se ha intentado solucionar subiendo el delay o regulando los valores de comparación pero no se ha conseguido solucionar
- El segundo es que se depende del software para poder elegir la dirección en el bus del dispositivo. En este caso se ha seguido un orden preestablecido (1: bombo, 2: caja, 3: tom superior, 4: tom base, 5: charles, 6: crash, 7: ride), pero queremos solucionar este problema de otra forma.

Para solucionar estos problemas se optó por implementar un filtro de media móvil, con el que se intentará eliminar los dobles golpes ajustando los coeficientes del filtro de forma experimental, además del software necesario para elegir la dirección del nodo mediante los jumpers que implementa el segundo prototipo de placa de nodo ya enseñada en apartados anteriores.

A continuación se adjunta el ejemplo de la caja con las nuevas mejoras explicadas anteriormente:

```
#include <Wire.h>

int ent0=A0;
int ent1=A1;
int vecent0[]={0,0,0,0};//sensor del parche
int vecent1[]={0,0,0,0};// sensor del casco
int valorent1=0;
int valorent0=0;
bool leerjumpers[4];
int i;
int dir;

void setup() {
  pinMode(5, INPUT);
  pinMode(4, INPUT);
  pinMode(3, INPUT);
  pinMode(2, INPUT);
    for(i=0; i<4; i++){
    leerjumpers[i]=digitalRead(i+2);
  }
  dir=leerjumpers[0]+2*leerjumpers[1]+4*leerjumpers[2]+8*leerjumpers[3];
```

```
  Wire.begin(dir); //Dirección 2 corresponde a la caja
  Wire.onRequest(respuesta);
}

void loop() {
 for(i=3; i>0; i--){
  vecent0[i]=vecent0[i-1];
  vecent1[i]=vecent1[i-1];
 }
 vecent0[0]=analogRead(ent0);
 vecent1[0]=analogRead(ent1);
 valorent0=0.7*vecent0[0]+0.2*vecent0[1]+0.05*vecent0[2]+0.05*vecent0[3];
 valorent1=0.7*vecent1[0]+0.2*vecent1[1]+0.05*vecent1[2]+0.05*vecent1[3];
delay(25);
}
void respuesta(){
  if ((valorent1>valorent0) && (valorent1>60)){
    Wire.write("w"); //mandamos señal cross
  } else if((valorent0>180) && (valorent1>80)){
    Wire.write("e");// rimshot
  } else if(valorent1<80){
    if(valorent0>40){
      Wire.write("r");//golpe flojo
      } else if ((valorent0<130) && (valorent0>40)){
      Wire.write("t");//golpe medio-flojo
    } else if ((valorent0<280) && (valorent0>130)){
      Wire.write("y");//golpe medio
    } else if (valorent0>280){
      Wire.write("u");//golpe fuerte
    }
  } else (Wire.write("m"));
}
```

Este filtro consiste en guardar los últimos cuatro valores muestreados y ponderarlos en función del valor de cada uno, dando siempre más importancia a los más recientes.

Se puede ver como se han tenido que reducir los valores de comparación para asignar un carácter u otro que se envía por el bus, debido a que, aunque el filtro reduce muy poco el valor de cada golpe, los valores anteriores causaban que los golpes se tuviesen que dar demasiado fuerte.

De esta forma se puede ver el funcionamiento general de los nodos. Aunque aquí solo se muestre como ejemplo el funcionamiento de la caja, en los anexos se adjuntan el resto de los códigos, donde se podrán ver los valores de comparación y el funcionamiento análogo.

6.6 Programación del Arduino Pro Micro

Una vez visto el método de envío de información necesitamos crear un software que pueda transformar los caracteres que enviamos en los valores MIDI que se

encuentran en la tabla de sonidos que se ha mostrado en apartados anteriores. Si volvemos a fijarnos en la Imagen 57, el Arduino central se encargará de la quinta y sexta etapa.

Aun así existen dos funcionalidades que son las más difíciles de implementar:

- Establecer mediante hardware el número de nodos que se encuentran en el sistema.
- Identificar mediante los mensajes recibidos que tipo de nodo es cada uno (caja, tom, bombo, etc).

Para averiguar el número de nodos que existen en el sistema se establecerá un sistema de jumpers similar al de los nodos, donde el usuario solo deberá seleccionar el número de dispositivos que se encuentran en su red. Este sistema se puede ver en el esquemático final de la base (mostrado en los apartados anteriores).

```
#include <Wire.h>

void noteOn(byte channel, byte pitch, byte velocity) {
  MIDIEvent noteOn = {0x09, 0x90 | channel, pitch, velocity};
  MIDIUSB.write(noteOn);
}

void noteOff(byte channel, byte pitch, byte velocity) {
  MIDIEvent noteOff = {0x08, 0x80 | channel, pitch, velocity};
  MIDIUSB.write(noteOff);
}

bool leerjumpers[4];
int entjumpers[]={10,14,15,16}; //10 lsb, 16 msb
int i;
int numdisp;
char c;
char charrecibidos[17];//número máximo de dispositivos según jumpers
                       //usaremos solo las posiciones necesarias
void setup() {
  for(i=0;i<4;i++){
  pinMode(entjumpers[i], INPUT);
  }
    for(i=0; i<4; i++){
    leerjumpers[i]=digitalRead(i);
  }
  numdisp=leerjumpers[0]+2*leerjumpers[1]+4*leerjumpers[2]+8*leerjumpers[3];
  Wire.begin();
}

void loop() {
  for(i=1;i<(numdisp+1);i++){
    Wire.requestFrom(i, sizeof(char));
    while(Wire.available()){
      charrecibidos[i]=Wire.read();
    }
  } //ya tenemos el valor de todos los nodos
//  for(i=1;i<(numdisp+1);i++){
//    intenviarMIDI[i]=correspondencia([charrecibidos[i]);
```

```
//  }
  for(i=1;i<(numdisp+1);i++){
    noteOn(0,correspondencia([charrecibidos[i]),64);
  }
  MIDIUSB.flush();
  delay(30);
}
```

Este ha sido el código desarrollado para el Arduino central. En el podemos ver como al principio y mediante los jumpers seleccionamos el número de elementos en la red y después escaneamos todos los nodos pidiendo información. Esta se guarda en un vector de caracteres que, mediante la función correspondencia (se encuentra en los anexos) convierte cada char en su respectivo número de nota MIDI. Este es enviado como parámetro de la función noteOn para conseguir así el sonido deseado. No se ha añadido la función noteOff porque se han hecho pruebas viendo si esta era necesaria, ya que las notas de una batería tienen una duración determinada (en una guitarra o un teclado sí que tendría sentido usar la función y no se podría eliminar como en este caso).

Con esto queda terminado todo el trabajo de integración hardware-software del proyecto. Lo único que resta es el montaje final y las pruebas en busca de limitaciones del sistema o posibles mejoras.

7 Montaje final y pruebas del sistema

Una vez explicada y desarrollada cada parte que compone el sistema completo llega el momento de montar todos los nodos en su respectivo lugar para poder usar el sistema completo y, de esta forma, poder valorar el funcionamiento final del conjunto.

Algo que no se había mencionado hasta ahora era la colocación de las clavijas de entrada y salida de datos y alimentación a los nodos. Una de las ideas con las que se diseñó el sistema de sustentación de los sensores piezoeléctricos fue la de no alterar en absoluto los cascos de la batería. De la misma forma que se diseñaron las escuadras con la distancia justa de las torres para evitar realizar orificios se optó por usar los agujeros de ventilación para sacar los cables de cada nodo. De esta forma las clavijas quedan accesibles para realizar las conexiones necesarias a la vez que la placa del nodo no queda expuesta (se encuentra protegida en el interior del casco).

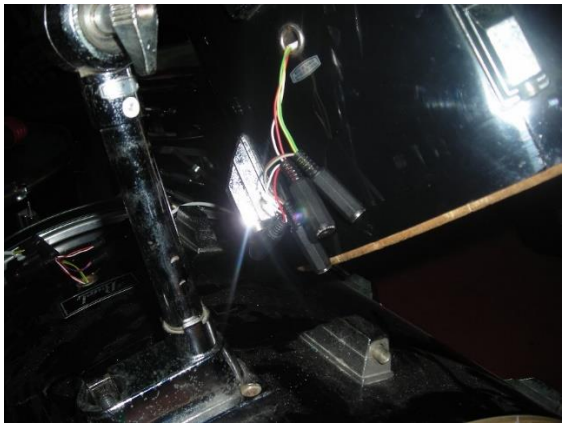


Imagen 116. Muestra de la colocación de las clavijas en los cascos

Cabe destacar que no se planteó un sistema de sujeción de los nodos dentro del casco, ya que las placas son pequeñas y pesan poco, por lo que se solían poner colgando y sujetando con cinta los cables. Una idea que puede arreglar este problema es realizar un par de orificios en los perfiles de aluminio y atornillar los nodos en estos (aislando la placa para evitar cortocircuitos, se podrían usar separadores de nylon por ejemplo).

Otro aspecto a valorar es el ajuste de los conos. Lo ideal es que este sobresalga unos dos milímetros por encima del casco. Para ello usaremos el sistema de ajuste de muelles y tornillos para conseguir la distancia deseada. No es conveniente que el cono sobresalga en exceso ya que puede aplastarse demasiado, deteriorarse y crear bultos en el parche, algo que es muy incómodo a la hora de tocar. Además, hay que procurar tocar siempre lo más próximo al centro del parche, para mejorar así la señal producida por los piezoeléctricos.

Imagen 117. Detalle del sistema de ajuste por muelles

Otro tema a destacar en el montaje es que los platos están sensorizados en cierto sector de su área, por lo que el sonido y la detección óptima se producirán cuando se golpee en esta área.

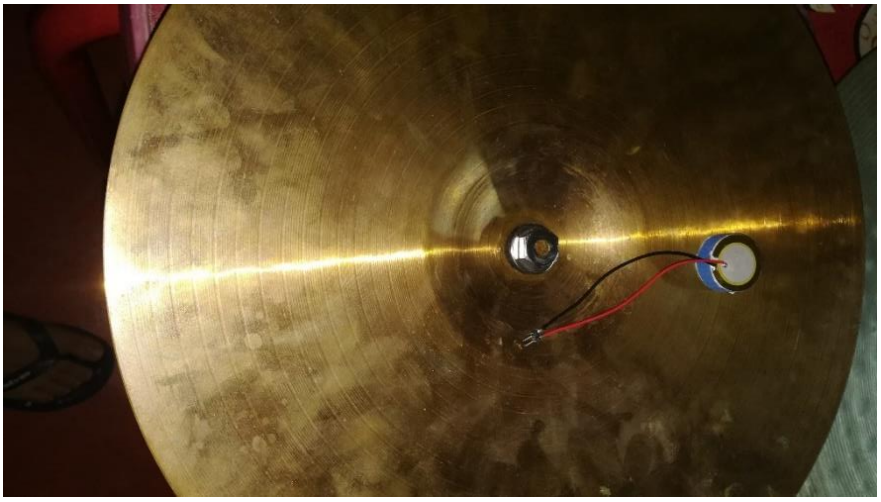


Imagen 118. Detalle del sector donde se coloca el sensor en el plato del charles

Como último detalle del montaje destacar que la tensión en los parches no debe ser ni muy alta ni muy baja (algo normal en toms y base para conseguir ataque y sonido grave). Un parche muy aflojado no vibrará lo suficiente y, por ende, no se generará voltaje en nuestros sensores.

Como ya se ha explicado a lo largo de toda la memoria el funcionamiento y montaje de cada nodo y de la placa central además del montaje de las estructuras y del sensor de charles, el montaje se reduce a montar una batería estándar (sin modificaciones).

Finalmente se procedió a conectar el sistema al ordenador (mediante el puerto Micro-USB del Arduino Pro Micro) y a configurar el dispositivo como entrada MIDI de Addictive Drums (mostrado también en apartados anteriores).

Imagen 119. Muestra del sistema montado y cableado completamente

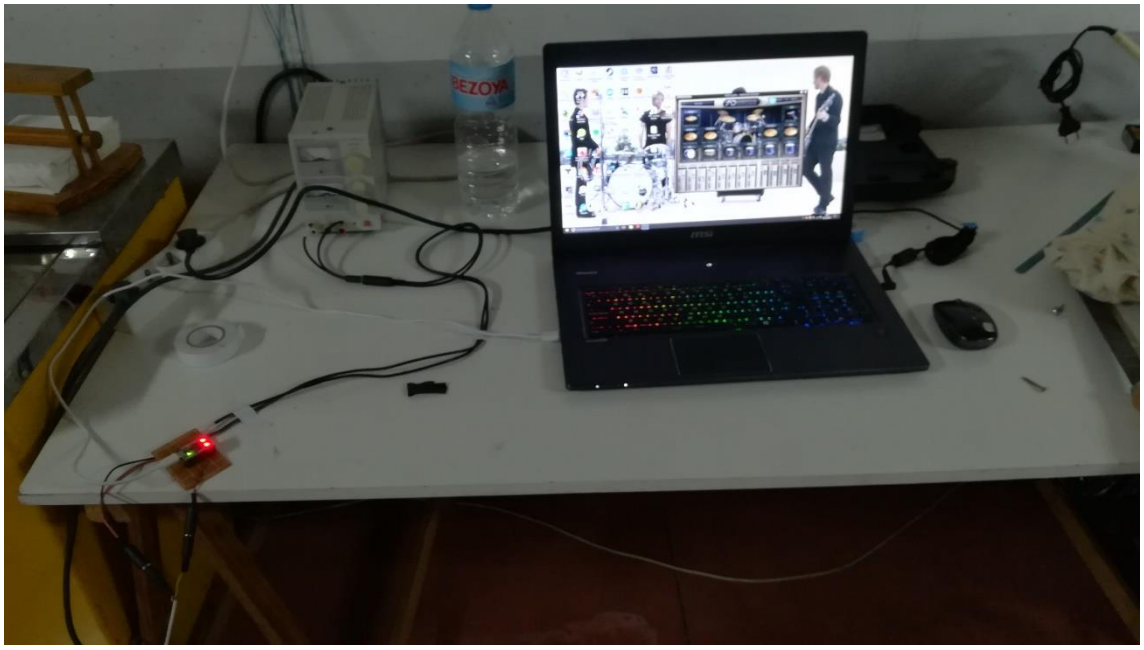


Imagen 120. Muestra del bus conectado a la interfaz y esta al ordenador con Addictive Drums

Las primeras pruebas que se le realizaron al sistema fueron simplemente golpear todos los parches viendo si según variábamos la fuerza del golpeo los sonidos cambiaban.

Lo siguiente fue realizar esto mismo pero en los platos, probando el cambio de sonido según varía la fuerza además de la detección de zonas en el ride y la caja.

Además se probó el sensor del pedal del charles viendo las variaciones según la apertura.

El sistema respondió de forma correcta a todas las pruebas, tal y como se esperaba, aunque el resultado final del proyecto no sea tan bueno como el deseado como se expondrá en el siguiente apartado.

8 Presupuesto

A continuación se ha realizado un presupuesto para valorar el coste total del desarrollo del proyecto. Como primera parte se muestra el precio total de los materiales que se han utilizado.

DESCRIPCIÓN	UNIDADES	PRECIO UNITARIO	PRECIO
Sensores piezoeléctricos de 35 mm	10	0,48	4,8
Sensores piezoeléctricos de 27 mm	10	0,44	4,4
Cable 4 hilos (en metros)	4	0,5	2
Clavijas macho jack	36	0,8	28,8
Clavijas hembra jack	36	0,8	28,8

Pinera macho (tiras)	10	0,4	4
Pinera hembra (tiras)	10	0,4	4
Tornillería varia	1	25	25
Perfiles aluminio (metro)	4	2,5	10
Resistencias varias	1	2	2
Diodos	20	0,1	2
Estaño	1	9,26	9,26
Bloques de lija	5	0,7	3,5
Spray adhesivo	1	8	8
Cable de un hilo rígido (en metros)	3	0,55	1,65
Muelles varios	1	4,5	4,5
Placa preperforada	20	0,25	5
Arduino Pro Micro	7	3	21
Arduino Nano	2,5	10	25
PLA para impresión 3D	1	16,51	16,51
Cinta doble cara	1	8	8
Termoretráctiles varios	1	10	10
Clavija de alimentación	1	0,25	0,25
Alimentador de pared	1	15	15
Placa fotosensible de doble cara	1	20	20
Regletas de dos pines	35	0,2	7
Cinta aislante	1	3,5	3,5
		TOTAL	**273,97**

Una vez visto el coste de los materiales, debemos contemplar el precio de las licencias de software que hemos usado para desarrollar el producto, ya que estas herramientas son imprescindibles para nosotros.

DESCRIPCIÓN	UNIDADES	PRECIO UNIDAD	PRECIO
Addictive Drum	1	145	145
Diptrace Estándar	1	395	395
Office	1	150	150
		TOTAL	**690**

Por último debemos contemplar el coste de las horas de desarrollo de código, de montaje, de pruebas, de redacción, etc. Esto corresponde al trabajo del ingeniero que ha desarrollado el proyecto (el autor de la memoria, yo en este caso).

DESCRIPCIÓN	HORAS	PRECIO POR HORA	PRECIO
Desarrollo de código	20	30	600
Desarrollo de esquemáticos y PCB	15	25	375
Montaje	12	20	240

Pruebas	5	20	100
Redacción y documentación	10	10	100
		TOTAL	**1415**

Una vez valoradas todas las variables del proyecto, podemos concluir que el coste final del desarrollo del proyecto asciende a la suma del coste de cada parte:

DESCRIPCIÓN	PRECIO
Materiales	273,97
Software	690
Trabajo del ingeniero	1415
TOTAL	2378,97

Finalmente podemos ver que el coste de desarrollo asciende a DOS MIL TRESCIENTOS SETENTA Y OCHO EUROS CON 97 CÉNTIMOS.

9 Conclusiones y futuros trabajos

9.1 Conclusión del proyecto y limitaciones de este

Como conclusión principal del proyecto podemos afirmar que el uso de sensores piezoeléctricos es totalmente compatible y válido para la sensorización de golpeo en instrumentos de percusión usando el acondicionamiento correspondiente. Hemos conseguido nuestro propósito usando estos sensores y plataformas de código abierto (Arduino), lo que ha permitido el desarrollo de una interfaz de precio asequible. Sin embargo existen ciertas limitaciones que hacen que, aunque el sistema funciona correctamente, provocan que el sistema no sea del todo amigable.

El principal problema es, sin duda, la falta de realismo o sensibilidad en el sonido conseguido. Puesto que, como se ha podido ver en los códigos, cambiamos de sonido en sonido a partir de un valor límite, lo que no nos permite realizar cambios graduales (crescendos por ejemplo) ya que llegará un momento en el que el sonido cambie bruscamente. Además no se consiguió encontrar un comando MIDI para poder graduar el volumen de la nota emitida, lo que hubiera conseguido cierta graduación.

También hay que mencionar que el sistema requiere de cierto tiempo de adaptación a él ya que el usuario debe adaptar en cierta forma la fuerza de los golpes para conseguir los sonidos que se desean ya que al principio podríamos conseguir sonidos indeseados sin el sistema se toca como una batería estándar sin tener en cuenta estas consideraciones.

Otra limitación del sistema consiste en el establecimiento de una tensión de parche mínima. Esto puede ser contraproducente ya que en baterías normales los timbales suelen estar muy poco tensados para conseguir tonos graves, pero en este sistema esto no se podría hacer, por lo que la sensación de tocar en timbales con parches tensos puede ser extraña al principio.

Aunque esto suene como algo que hace el sistema inútil no es así en absoluto. El sistema es usable para grabar pistas poco exigentes, ritmos básicos o usar el sistema como batería de aprendizaje. Además se puede usar este sistema para mejorar una batería muy básica o barata ya que conseguiríamos mejores sonidos y más variedad.

Sin embargo no se contempla la idea de usar este sistema en grabaciones exigentes, conciertos en directo o en géneros musicales donde se necesite mucha expresividad (jazz por ejemplo).

9.2 Mejoras del proyecto y futuros trabajos

Viendo los problemas y limitaciones que han surgido a lo largo de todo el proyecto se han ido acumulando ciertas ideas que podrían mejorar nuestro sistema.

La primera mejora directa al sistema sería conseguir conectores de cuatro pines, lo que conseguiría reducir los cables que van de nodo a nodo a un solo cable con una clavija. Ya se habló de esto en un capítulo de la memora, por lo que todo se reduce a conseguir conectores hembra y macho de tipo Jack de cuatro pines o XLR de cuatro pines.

La siguiente mejora sería la detección automática de los valores que se usan de comparación para pasar de un sonido a otro. Esto evitaría establecer los valores a base de prueba y error, lo que puede provocar que para algunos usuarios el sistema sea poco sensible (haya que tocar muy fuerte) mientras para otros pase justo lo contrario.

Un método que evitaría estos problemas sería la detección del pico de la onda, sabiendo que este será el valor más alto y, así, disparar en este momento el comando MIDI. Estudiando así los picos y la forma de onda todo se compararía de forma relativa y no se dependería de una comparación con un valor entero definido.

Otra mejora que daría un salto de calidad al proyecto sería la detección de zona de golpeo. Roland incorpora ya esta tecnología en la que se produce un sonido diferente según la zona del parche que se golpee usando un solo sensor piezoeléctrico, lo que permite una gran variedad de tonos y, combinado con el control de volumen, una muy buena expresividad.

La siguiente mejora consistiría en que la tensión del parche fuera independiente de la recepción de la fuerza de golpeo del sensor. Esto permitiría que la sensación de golpeo fuera exactamente igual en una batería normal y en nuestro sistema.

Por último, sería una gran mejora el diseñar un sistema de montaje de los conos en los cascos que no dependiera de los agujeros de las torres. La idea sería crear un sistema de montaje rápido en el casco, lo que aceleraría muchísimo el tiempo de instalación de los sensores del sistema.

Además de las mejoras que se han planteado, este proyecto puede dar lugar a otros trabajos o ideas que partan desde este mismo sistema, creando mejoras o funcionalidades añadidas para el sistema.

Una idea que se planteó como mejora fue la de usar este sistema para enseñar a tocar la batería a gente mediante el uso de LEDs en los timbales y platos. Una vez que nuestro sistema se encuentra conectada vía MIDI al ordenador podríamos mandar señales en forma de LEDs que se iluminarían en cada casco o plato cuando hubiera que tocar este, creando así un método de aprendizaje muy visual y dinámico. Mediante la señal MIDI que enviamos al golpear podríamos saber si se ha golpeado en el momento justo o no y, en función de si el golpeo se ha hecho bien o mal, iluminar el LED de un color u otro.

10 Bibliografía y referencias

- Maria Alfaro. Origen y evolución de la batería, conjunto de instrumentos de percusión (consultado en Julio de 2017)

  http://danielmartin-mallets.com/blog-percusion/es/origen-y-evolucion-de-la-bateria-conjunto-de-instrumentos-de-percusion/

- Mauro Orizi (2012). Principios de funcionamiento acústico musical de una batería (consultado en Julio de 2017)
  https://whatawonderfuldrumworld.wordpress.com/2012/01/17/principios-de-funcionamiento-acustico-musical-de-una-bateria/

- Wikipedia. Bombo de batería (consultado en Julio de 2017)
  https://es.wikipedia.org/wiki/Bombo_de_bater%C3%ADa

- Ralf Kistner. Repositorio en GitHub del fork Arcore MIDI (consultado en Julio de 2017)
  https://github.com/rkistner/arcore

- Tienda Arduino (especificaciones y modelos de microcontroladores) (consultado en Julio de 2017)
  https://store.arduino.cc/

- Fancho. Caja (instrumento musical) (consultado en Agosto de 2017) https://allendemusica.wordpress.com/2013/01/28/corno-ingles/

- Ecured. Caja (instrumento musical) (consultado en Agosto de 2017) https://www.ecured.cu/Caja_(Instrumento_Musical)

- Ricky Santos. Clasificación de parches (consultado en Agosto de 2017) http://www.mundopercusion.com/bateria/mundo-bateria/321-clasificacion-de-los-parches.html

- Pdaldrums. Tipos de parches (consultado en Agosto de 2017) http://www.pdaldrums.com/index.php/escritos-sobre-cajas/tipos-de-parches

- Wikipedia. Platillos (consultado en Agosto de 2017) https://es.wikipedia.org/wiki/Platillos

- Alex González (2015). Aleaciones de platillos (consultado en Agosto de 2017) http://blogdelmusico.com/aleaciones-de-platillos/

- Wikipedia. Sensor piezoeléctrico (consultado en Agosto de 2017) https://es.wikipedia.org/wiki/Sensor_piezoel%C3%A9ctrico

- Tecnología Fácil (2014). ¿Qué es el MIDI? (consultado en Agosto de 2017) https://tecnologia-facil.com/wp-content/cache/all/que-es/que-es-midi//index.html

- Cristian Parra Albarracín (2016). Sensores Piezoeléctricos (consultado en Agosto de 2017) https://prezi.com/bs1wtk-d2d6z/sensores-piezoelectricos/

- Laura Fernández Román y Lara Navarro Morales (2017). Sistema de bajo consumo para la detección del ritmo cardiaco periférico mediante un sensor piezoeléctrico (consultado en Agosto de 2017) https://upcommons.upc.edu/bitstream/handle/2099.1/4153/memoria.pdf

- Wikipedia. Batería Electrónica (consultado en Septiembre de 2017) https://es.wikipedia.org/wiki/Bater%C3%ADa_electr%C3%B3nica

- UPV. Síntesis del sonido (Protocolo y formato MIDI) (consultado en Septiembre de 2017) http://www.disca.upv.es/adomenec/IASPA/tema5/Midi.html

- Wikipedia. HID (consultado en Septiembre de 2017) https://es.wikipedia.org/wiki/HID

- The MIDI Association (2016). USB y MIDI: conceptos básicos (consultado en Septiembre de 2017) https://www.hispasonic.com/blogs/usb-midi-conceptos-basicos/42115

- Sergi Jordá Puig (1997). Audio digital y MIDI (consultado en Septiembre de 2017) http://www.ccapitalia.net/reso/articulos/audiodigital/pdf/07-IntroMIDI.pdf

Anexos

Esquemáticos:

- Acondicionamiento de los sensores
- Botonera para pruebas
- Nodo 1.0
- Conexionado en pruebas de comunicación
- Nodo 1.1 (con detección de dirección)
- Cable hembra para nodo
- Cable macho-macho para interconexión de nodos
- Microcontrolador central (con selección del número de nodos)

PCBs:

- Nodo (con detección de dirección)
- Microcontrolador central (con detección del número de nodos)

Planos:

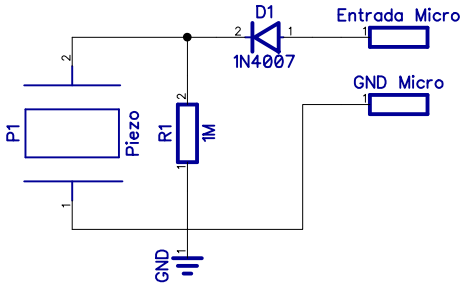
- Escuadra de montaje
- Soporte sensor piezoeléctrico
- Soporte para dar forma a conos

Códigos:

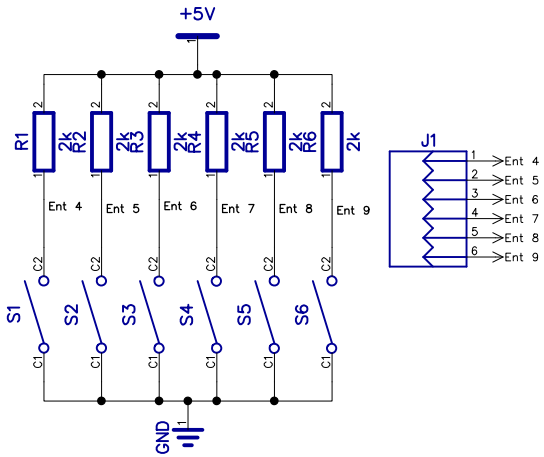
- Valores de salida por serial
- Ejemplo MIDI Arcore
- Prueba MIDI Arcore con sensor
- Ver variaciones con botonera y LCD
- Código esclavos prueba comunicación I2C
- Código maestro prueba comunicación I2C
- Código nodo v 1.0
- Código nodo v 1.2 bombo
- Código nodo v 1.2 caja
- Código nodo v 1.2 tom
- Código nodo v 1.2 tom base
- Código nodo v 1.2 charles
- Código nodo v 1.2 crash
- Código nodo v 1.2 ride
- Código Arduino Central

Otros:

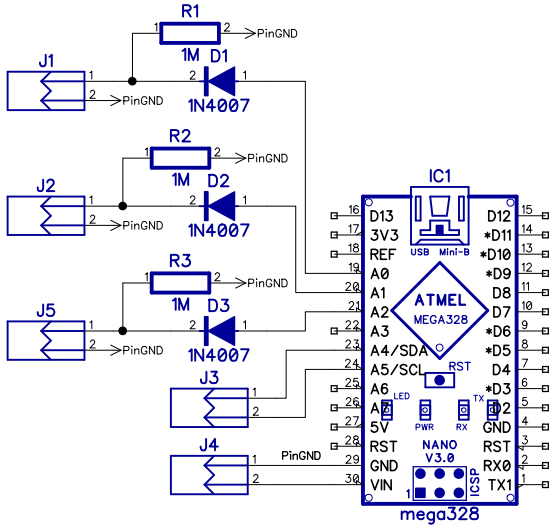
- Tabla de sonidos



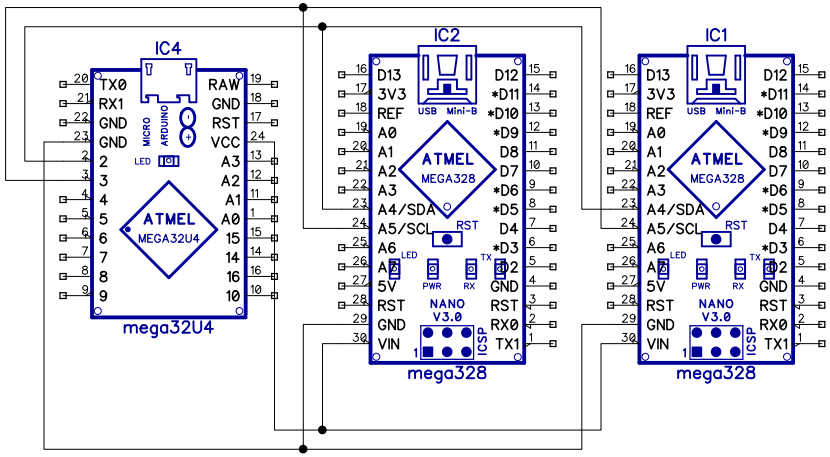
Desarrollo de interfaz para monitorizacion en instrumentos de percusion a partir de sensores piezoelectricos		
N esq 1	Descripcion esquematico Acondicionamiento de los sensores	Rev 1.0
Fecha: 30/09/2017		Sheet 1
Jesus Balsalobre Martinez		TFG



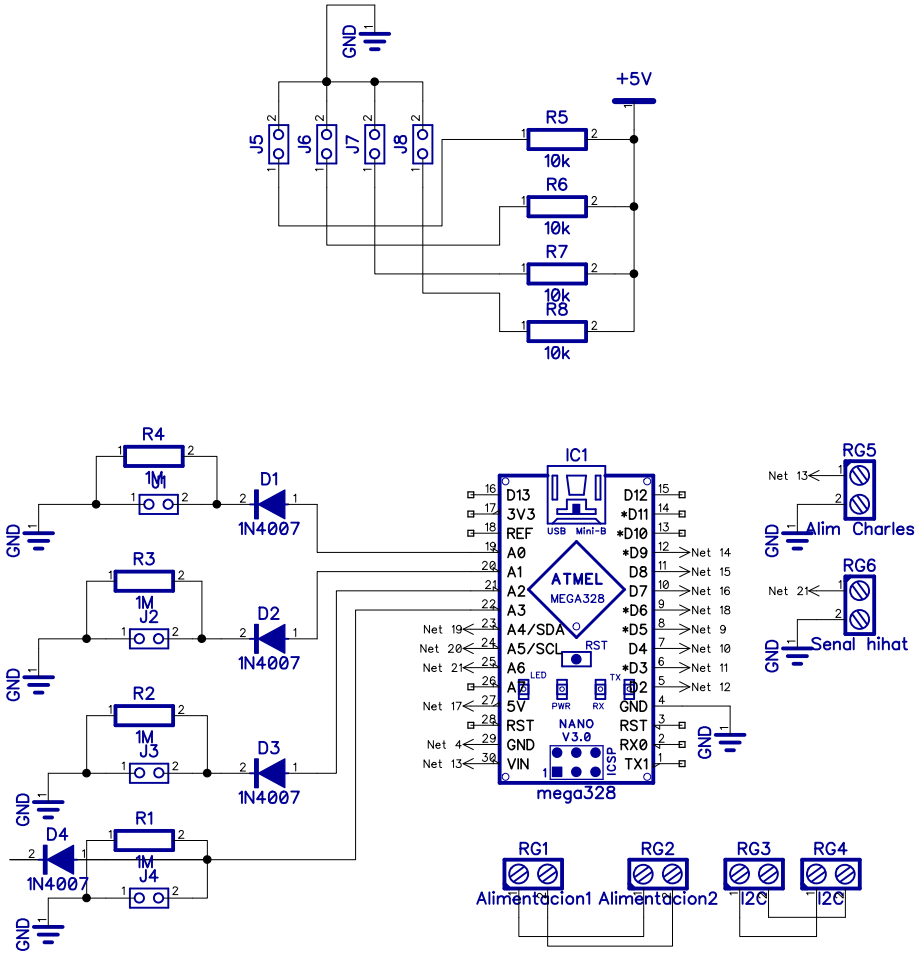
Desarrollo de interfaz para monitorizacion en instrumentos de percusion a partir de sensores piezoelectricos		
N esq 2	Descripcion esquematico Botonera para pruebas	Rev 1.0
Fecha: 30/09/2017		Sheet 1
Jesus Balsalobre Martinez		TFG



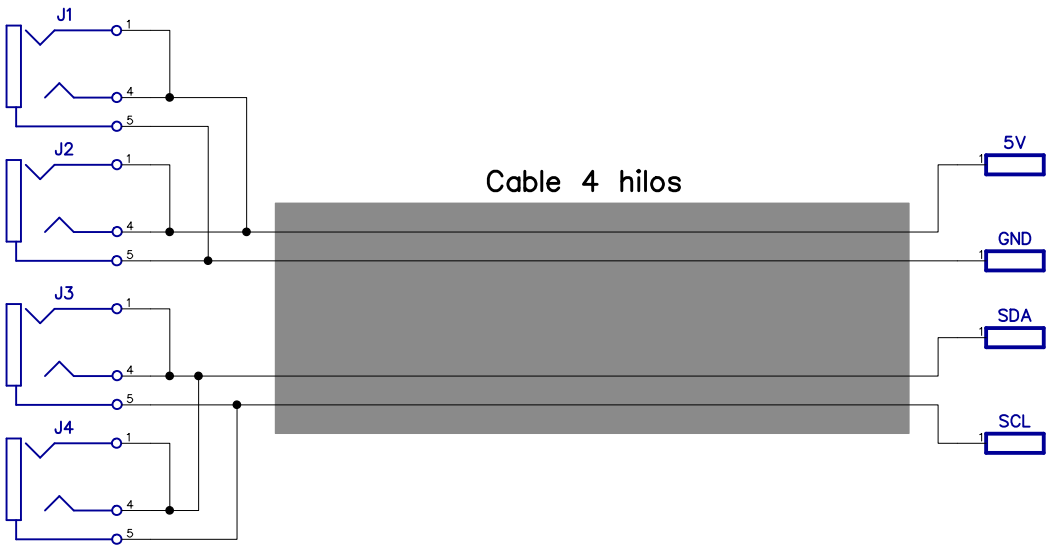
Desarrollo de interfaz para monitorizacion en instrumentos de percusion a partir de sensores piezoelectricos		
N esq 3	Descripcion esquematico Nodo	Rev 1.0
Fecha: 31/09/2017		Sheet 1
Jesus Balsalobre Martinez		TFG



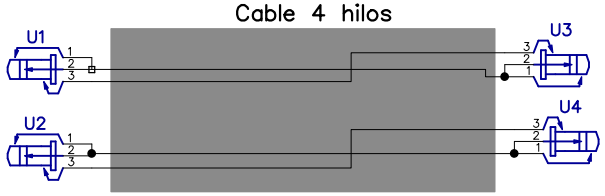
Desarrollo de interfaz para monitorizacion en instrumentos de percusion a partir de sensores piezoelectricos		
N esq 4	Descripcion esquematico Conexionado en pruebas de comunicacion	Rev 1.0
Fecha: 30/09/2017		Sheet 1
Jesus Balsalobre Martinez		TFG



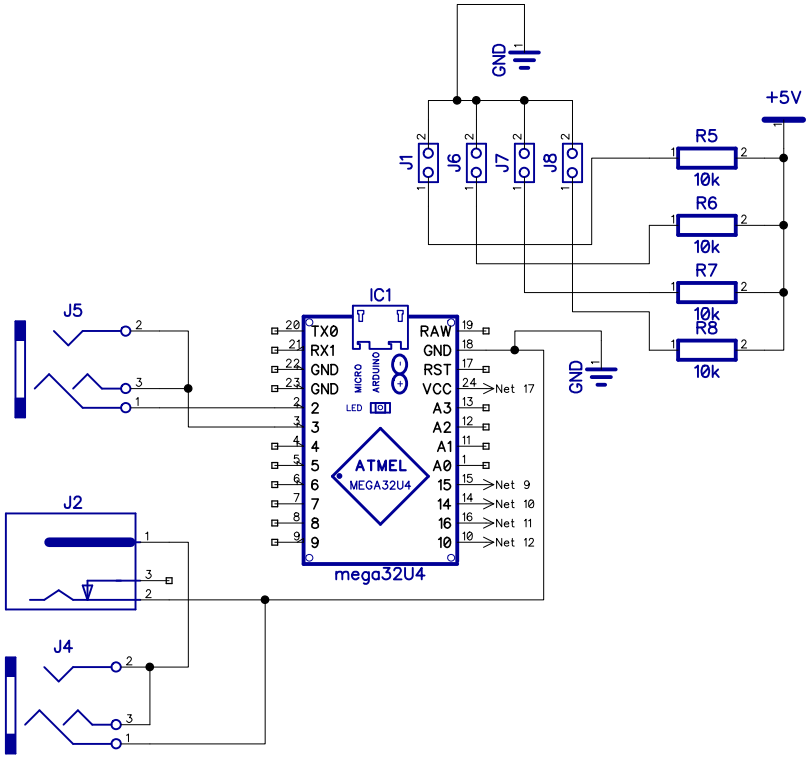
Desarrollo de interfaz para monitorizacion en instrumentos de percusion a partir de sensores piezoelectricos		
N esq 5	Number Nodo (con deteccion de direccion)	Rev 1.1
Fecha: 30/09/2017	Sheet 1	
Jesus Balsalobre Martinez	TFG	



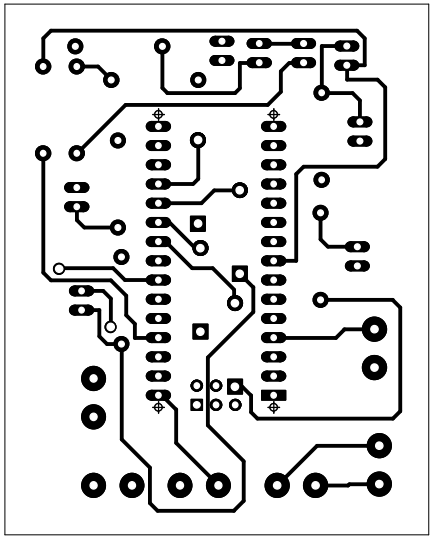
Desarrollo de interfaz para monitorizacion en instrumentos de percusion a partir de sensores piezoelectricos		
N esq 6	Descripcion esquematico Cable hembra para nodo	Rev 1.0
Fecha: 30/09/2017		Sheet 1
Jesus Balsalobre Martinez		TFG



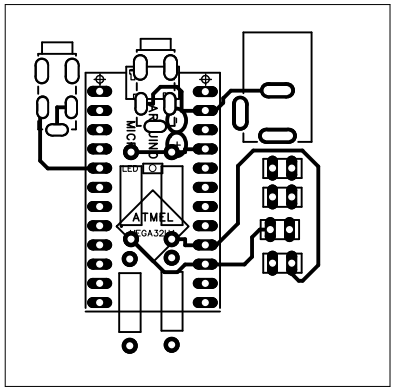
Desarrollo de interfaz para monitorizacion en instrumentos de percusion a partir de sensores piezoelectricos		
N esq 7	Descripcion esquematico Cable macho-macho para interconexion de nodos	Rev 1.0
Fecha: 30/209/2017		Sheet 1
Jesus Balsalobre Martinez		TFG



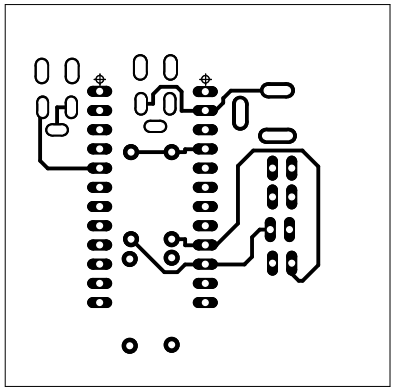
Desarrollo de interfaz para monitorizacion en instrumentos de percusion a partir de sensores piezoelectricos		
N esq 8	Descripcion esquematico Microcontrolador central (con seleccion del numero de nodos)	Rev 1.1
Fecha: 30/09/2017		Sheet 1
Jesus Balsalobre Martinez		TFG



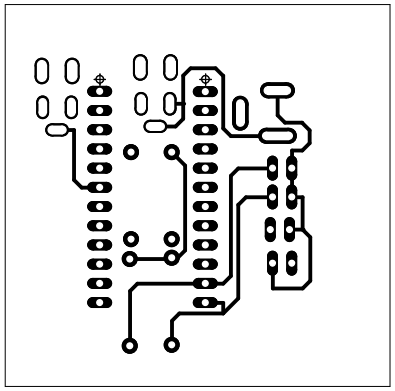
Desarrollo de interfaz para monitorizacion en instrumentos de percusion a partir de sensores piezoelectricos		
Num	Descripcion de la placa	Rev
1	Nodo con deteccion de direccion	1.0
Fecha: 30/09/2017		Capa TOP
Jesus Balsalobre Martinez		TFG



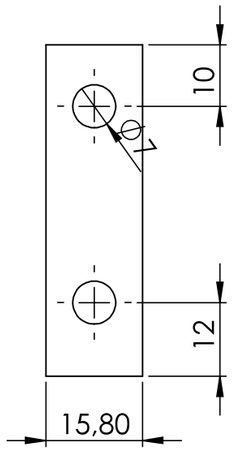
Desarrollo de interfaz para monitorizacion en instrumentos de percusion a partir de sensores piezoelectricos		
Num	Descripcion de la placa	Rev
2	Placa del microncontrolador central (con deteccion del numero de nodos)	1.0
Fecha: 30/09/2017		Capa BOTTOM
Jesus Balsalobre Martinez		TFG

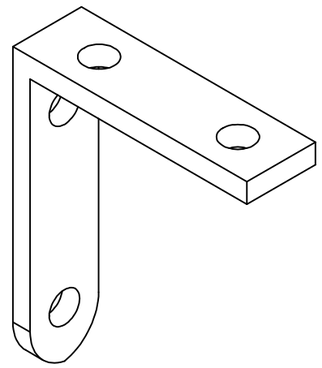


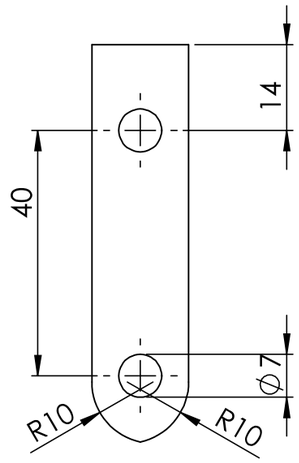
Desarrollo de interfaz para monitorizacion en instrumentos de percusion a partir de sensores piezoelectricos		
Num	Descripcion de la placa	Rev
2	Placa del microncontrolador central (con deteccion del numero de nodos)	1.0
Fecha: 30/09/2017		Capa TOP
Jesus Balsalobre Martinez		TFG

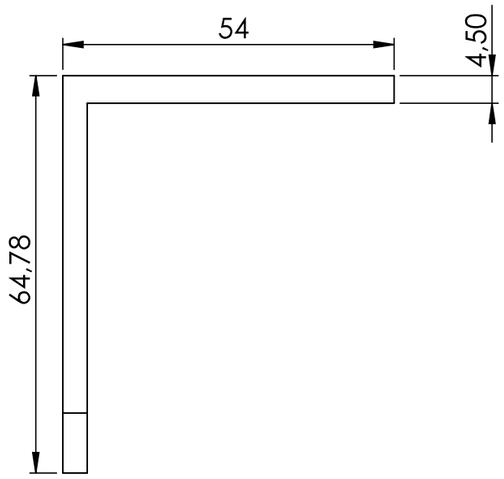


Desarrollo de interfaz para monitorizacion en instrumentos de percusion a partir de sensores piezoelectricos		
Num	Descripcion de la placa	Rev
2	Placa del microncontrolador central (con deteccion del numero de nodos)	1.0
Fecha: 30/09/2017		Capa BOTTOM
Jesus Balsalobre Martinez		TFG

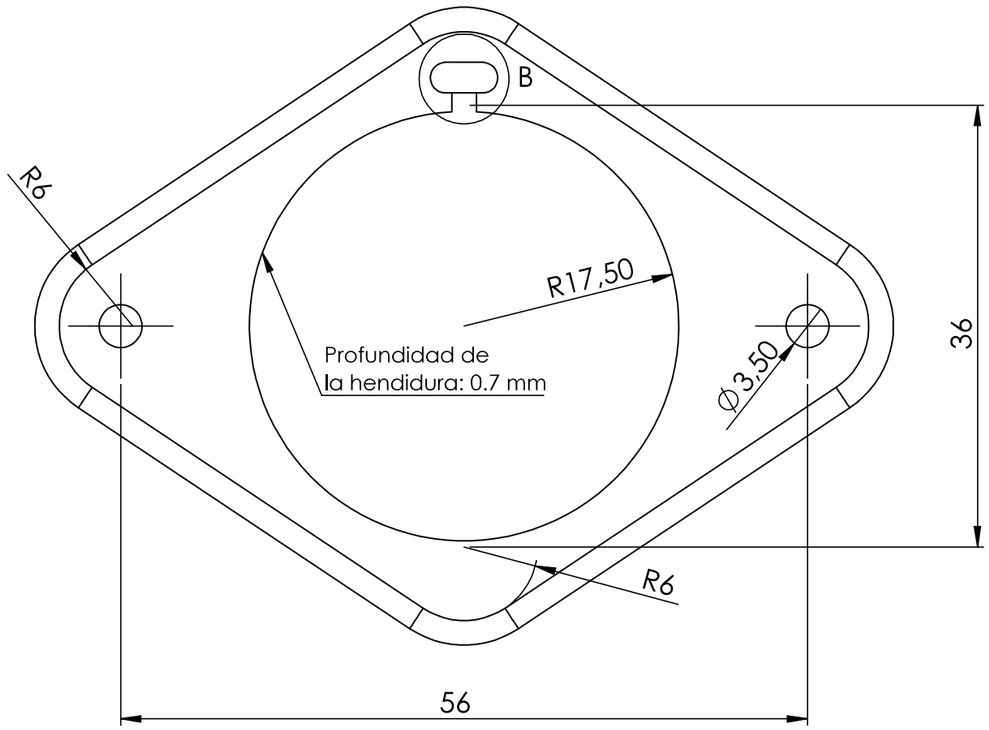


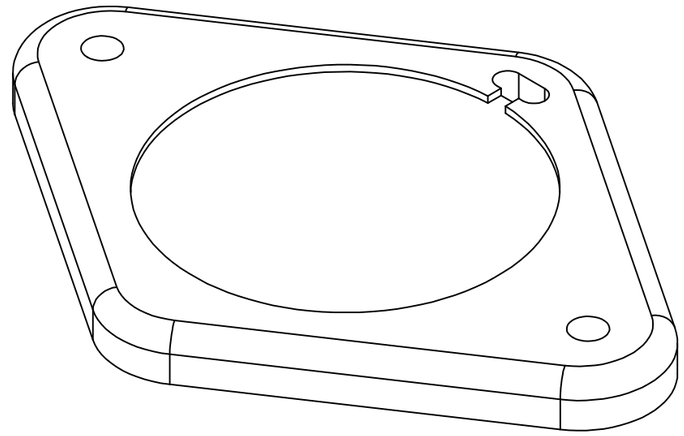


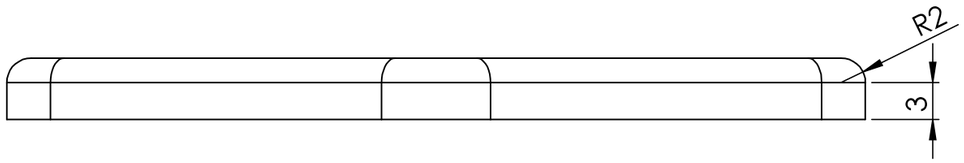


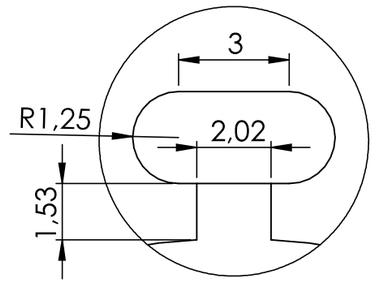


UNIVERSIDAD POLITÉCNICA DE CARTAGENA Departamento de Expresión Gráfica			DISEÑO ASISTIDO POR ORDENADOR		
			Curso 2016/2017	Práctica nº	Hoja / Escala
Titulación	GIEIA	Denominación: Escuadra de montaje		TFG	1 / 1:1
Fecha	30/09/2017			Revisión:	1.0
Apellidos y Nombre:	Jesús Balsalobre Martínez			DNI:	48705932M



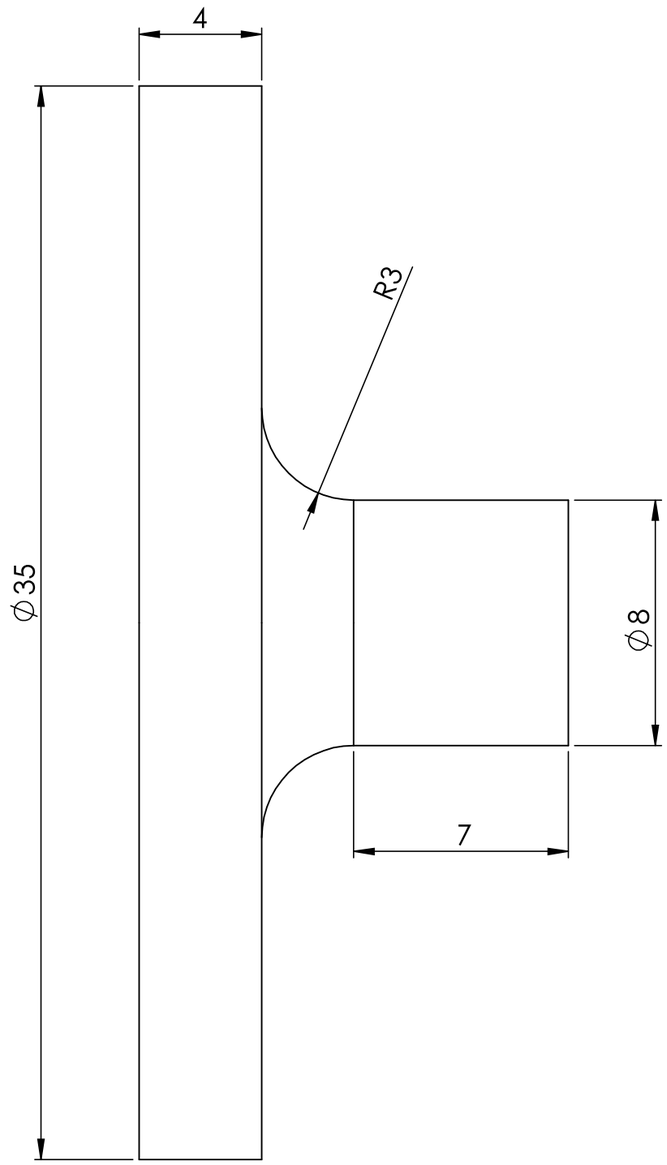


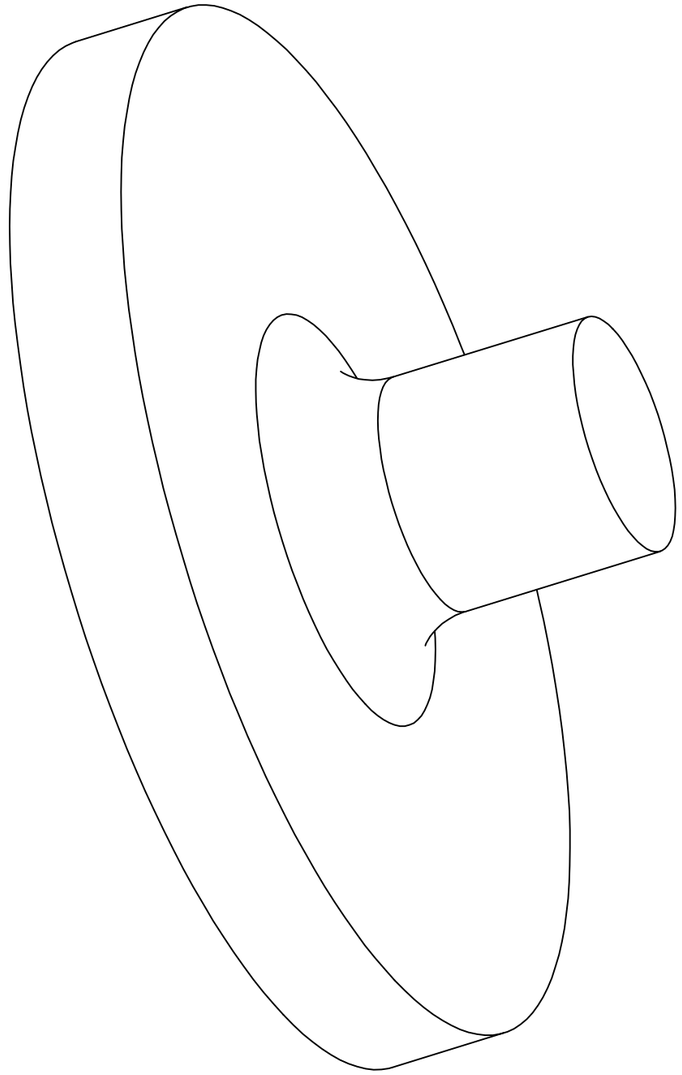




DETALLE B
ESCALA 6 : 1

UNIVERSIDAD POLITÉCNICA DE CARTAGENA Departamento de Expresión Gráfica		DISEÑO ASISTIDO POR ORDENADOR			
		Curso 2016/2017	Práctica nº	Hoja	Escala
Titulación	GIEIA	Denominación: Soporte sensor piezoeléctrico	TFG	1	1:1
Fecha	30/09/2017		Revisión:	1.0	
Apellidos y Nombre: Jesús Balsalobre Martínez			DNI: 48705932M		





UNIVERSIDAD POLITÉCNICA DE CARTAGENA Departamento de Expresión Gráfica		DISEÑO ASISTIDO POR ORDENADOR			
		Curso 2016/2017	Práctica nº	Hoja	Escala
Titulación	GIEIA	Denominación:	TFG	1	1:1
Fecha	30/09/2017	Soporte para dar forma a conos	Revisión:	1.0	
Apellidos y Nombre:	Jesús Balsalobre Martínez		DNI:	48705932M	

```
1   ////////////////////////////////////////
2   /*
3   VALORES DE SALIDA POR SERIAL
4   Jesús Balsalobre Martínez
5   TFG: Desarrollo de interfaz para monitorizacion en instrumentos
6   de percusion a partir de sensores piezoelectricos
7
8   */
9
10  ////////////////////////////////////////
11  int sensor=A0;
12  int value;
13
14  void setup(){
15  Serial.begin(9600);
16  }
17
18  void loop(){
19    value=analogRead(sensor);
20   Serial.println(value);
21    delay(20);
22  }
23
```

```
////////////////////////////////////////
/*
EJEMPLO MIDI ARCORE
Jesús Balsalobre Martínez
TFG: Desarrollo de interfaz para monitorizacion en instrumentos
de percusion a partir de sensores piezoelectricos

*/

////////////////////////////////////////
void noteOn(byte channel, byte pitch, byte velocity) {
  MIDIEvent noteOn = {0x09, 0x90 | channel, pitch, velocity};
  MIDIUSB.write(noteOn);
}

void noteOff(byte channel, byte pitch, byte velocity) {
  MIDIEvent noteOff = {0x08, 0x80 | channel, pitch, velocity};
  MIDIUSB.write(noteOff);
}

// First parameter is the event type (0x0B = control change).
// Second parameter is the event type, combined with the channel.
// Third parameter is the control number number (0-119).
// Fourth parameter is the control value (0-127).

void controlChange(byte channel, byte control, byte value) {
  MIDIEvent event = {0x0B, 0xB0 | channel, control, value};
  MIDIUSB.write(event);
}

void loop() {
  noteOn(0, 48, 64);   // Channel 0, middle C, normal velocity
  MIDIUSB.flush();
  delay(500);

  noteOff(0, 48, 64);  // Channel 0, middle C, normal velocity
  MIDIUSB.flush();
  delay(1500);

  // controlChange(0, 10, 65); // Set the value of controller 10 on channel 0 to 65
}

void setup() {

}
```

```
1   ////////////////////////////////////////
2   /*
3   PRUEBA MIDI ARCORE CON SENSOR
4   Jesús Balsalobre Martínez
5   TFG: Desarrollo de interfaz para monitorizacion en instrumentos
6   de percusion a partir de sensores piezoelectricos
7
8   */
9
10  ////////////////////////////////////////
11  void noteOn(byte channel, byte pitch, byte velocity) {
12    MIDIEvent noteOn = {0x09, 0x90 | channel, pitch, velocity};
13    MIDIUSB.write(noteOn);
14  }
15
16  void noteOff(byte channel, byte pitch, byte velocity) {
17    MIDIEvent noteOff = {0x08, 0x80 | channel, pitch, velocity};
18    MIDIUSB.write(noteOff);
19  }
20
21  int sensor=A0;
22  int value;
23
24  void setup(){
25
26  }
27
28  void loop(){
29    value=analogRead(sensor);
30   // value=map(value,0,1023,0,255);
31  if(value>500){
32    noteOn(0,48,64);
33    MIDIUSB.flush();
34    delay(50);
35    noteOff(0, 48, 64);  // Channel 0, middle C, normal velocity
36    MIDIUSB.flush();
37    delay(110);
38  }
39    delay(20);
40  }
41
```

```
////////////////////////////////////////
/*
VER VARIACIONES EN SONIDOS CON BOTONERA Y LCD
Jesús Balsalobre Martínez
TFG: Desarrollo de interfaz para monitorizacion en instrumentos
de percusion a partir de sensores piezoelectricos

*/

////////////////////////////////////////

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

void noteOn(byte channel, byte pitch, byte velocity) {
  MIDIEvent noteOn = {0x09, 0x90 | channel, pitch, velocity};
  MIDIUSB.write(noteOn);
}

void noteOff(byte channel, byte pitch, byte velocity) {
  MIDIEvent noteOff = {0x08, 0x80 | channel, pitch, velocity};
  MIDIUSB.write(noteOff);
}

// First parameter is the event type (0x0B = control change).
// Second parameter is the event type, combined with the channel.
// Third parameter is the control number number (0-119).
// Fourth parameter is the control value (0-127).

void controlChange(byte channel, byte control, byte value) {
  MIDIEvent event = {0x0B, 0xB0 | channel, control, value};
  MIDIUSB.write(event);
}

int sensor=A0;
int value;
int a=0;
int b=40;
int c=64;
bool upa;
bool downa;
bool upb;
bool downb;
bool upc;
bool downc;

LiquidCrystal_I2C lcd(0x27,20,4);

void setup() {
  lcd.init();
  // Print a message to the LCD.
  lcd.backlight();
 // lcd.setCursor(3,0);

pinMode(4,INPUT);
pinMode(5,INPUT);
pinMode(6,INPUT);
pinMode(7,INPUT);
pinMode(8,INPUT);
pinMode(9,INPUT);
}
void loop() {
  value=analogRead(sensor);
  value = map(value, 0, 1023, 0, 255);
  if(value>100){
  noteOn(a, b, c);   // Channel 0, middle C, normal velocity
  MIDIUSB.flush();
  delay(25);
  noteOff(a, b, c);  // Channel 0, middle C, normal velocity
  MIDIUSB.flush();
  delay(10);
  }
    upa=digitalRead(4);
    downa=digitalRead(5);
    upb=digitalRead(6);
    downb=digitalRead(7);
    upc=digitalRead(8);
    downc=digitalRead(9);
if(upa){
  delay(500);
  a++;
}
if(downa){
  delay(500);
```

```
    a--;
  }
  if(upb){
    delay(500);
    b++;
  }
  if(downb){
    delay(500);
    b--;
  }
  if(upc){
    delay(500);
    c++;
  }
  if(downc){
    delay(500);
    c--;
  }
      lcd.print("a=");
     // lcd.setCursor(2,0);
      lcd.print(a);
      lcd.setCursor(8,0);
      lcd.print("b=");
      lcd.print(b);
      lcd.setCursor(0,1);
      lcd.print("c=");
      lcd.print(c);
      lcd.home();
  }

```

```
////////////////////////////////////////
/*
CÓDIGO ESCLAVOS PRUEBA COMUNICACION I2C
Jesús Balsalobre Martínez
TFG: Desarrollo de interfaz para monitorizacion en instrumentos
de percusion a partir de sensores piezoelectricos

*/

////////////////////////////////////////

#include <Wire.h>

void setup() {
  pinMode(7, INPUT);
  Wire.begin(2);                // join i2c bus with address (la que queramos)
  Wire.onRequest(requestEvent); // register event
}

void loop() {
  delay(100);
}

// function that executes whenever data is requested by master
// this function is registered as an event, see setup()
void requestEvent() {
  if(digitalRead(7)==LOW){
    Wire.write("a");
  } else (Wire.write("b"));
  // as expected by master
}

```

```
////////////////////////////////////////////
/*
CODIGO MAESTRO PRUEBA COMUNICACION I2C
Jesús Balsalobre Martínez
TFG: Desarrollo de interfaz para monitorizacion en instrumentos
de percusion a partir de sensores piezoelectricos

*/

////////////////////////////////////////////

#include <Wire.h>

char c;

void setup() {
  Wire.begin();        // join i2c bus (address optional for master)
  Serial.begin(9600);  // start serial for output
}

void loop() {
  for(int i=1;i<3; i++){
  Wire.requestFrom(i, sizeof(char));    // request 6 bytes from slave device #8
  while (Wire.available()) { // slave may send less than requested
    c = Wire.read();// receive a byte as character
  }
   Serial.print("Ent Ard ");
  Serial.print(i);
  Serial.print(" esta en ");
    Serial.println(c);

 }
  delay(100);
}

```

```
////////////////////////////////////////
/*
CODIGO NODO V 1.0
Jesús Balsalobre Martínez
TFG: Desarrollo de interfaz para monitorizacion en instrumentos
de percusion a partir de sensores piezoelectricos

*/

////////////////////////////////////////

#include <Wire.h>

int ent0=A0;
int ent1=A1;
int valorent0; //sensor del parche
int valorent1;// sensor del casco

void setup() {
  Wire.begin(2); //Dirección 2 corresponde a la caja
  Wire.onRequest(respuesta);
}

void loop() {
 valorent0=analogRead(ent0);
 valorent1=analogRead(ent1);
delay(25);
}
void respuesta(){
  if ((valorent1>valorent0) && (valorent1>80)){
    Wire.write("w"); //mandamos señal cross
  } else if((valorent0>200) && (valorent1>100)){
    Wire.write("e");// rimshot
  } else if(valorent1<100){
    if(valorent0>60){
      Wire.write("r");//golpe flojo
      } else if ((valorent0<150) && (valorent0>60)){
      Wire.write("t");//golpe medio-flojo
    } else if ((valorent0<300) && (valorent0>150)){
      Wire.write("y");//golpe medio
    } else if (valorent0>300){
      Wire.write("u");//golpe fuerte
    }
  } else (Wire.write("m"));
}

```

```
////////////////////////////////////////
/*
CODIGO NODO V 1.2 BOMBO
Jesús Balsalobre Martínez
TFG: Desarrollo de interfaz para monitorizacion en instrumentos
de percusion a partir de sensores piezoelectricos

*/

////////////////////////////////////////

#include <Wire.h>

int ent0=A0;
//int ent1=A1;
int vecent0[]={0,0,0,0};//sensor del parche
//int vecent1[]={0,0,0,0};// sensor del casco
//int valorent1=0;
int valorent0=0;
bool leerjumpers[4];
int i;
int dir;

void setup() {
  pinMode(5, INPUT);
  pinMode(4, INPUT);
  pinMode(3, INPUT);
  pinMode(2, INPUT);
    for(i=0; i<4; i++){
    leerjumpers[i]=digitalRead(i+2);
  }
  dir=leerjumpers[0]+2*leerjumpers[1]+4*leerjumpers[2]+8*leerjumpers[3];
  Wire.begin(dir); //Dirección 2 corresponde a la caja
  Wire.onRequest(respuesta);
}

void loop() {
 for(i=3; i>0; i--){
  vecent0[i]=vecent0[i-1];
  //vecent1[i]=vecent1[i-1];
 }
 vecent0[0]=analogRead(ent0);
 //vecent1[0]=analogRead(ent1);
 valorent0=0.7*vecent0[0]+0.2*vecent0[1]+0.05*vecent0[2]+0.05*vecent0[3];
 //valorent1=0.7*vecent1[0]+0.2*vecent1[1]+0.05*vecent1[2]+0.05*vecent1[3];
delay(25);
}
void respuesta(){
 if(valorent0>375){
    Wire.write("q");// golpe de bombo
  } else (Wire.write("m"));
}

```

```
////////////////////////////////////////
/*
CODIGO NODO V 1.2 CAJA
Jesús Balsalobre Martínez
TFG: Desarrollo de interfaz para monitorizacion en instrumentos
de percusion a partir de sensores piezoelectricos

*/

////////////////////////////////////////

#include <Wire.h>

int ent0=A0;
int ent1=A1;
int vecent0[]={0,0,0,0};//sensor del parche
int vecent1[]={0,0,0,0};// sensor del casco
int valorent1=0;
int valorent0=0;
bool leerjumpers[4];
int i;
int dir;

void setup() {
  pinMode(5, INPUT);
  pinMode(4, INPUT);
  pinMode(3, INPUT);
  pinMode(2, INPUT);
    for(i=0; i<4; i++){
    leerjumpers[i]=digitalRead(i+2);
  }
  dir=leerjumpers[0]+2*leerjumpers[1]+4*leerjumpers[2]+8*leerjumpers[3];
  Wire.begin(dir); //Dirección 2 corresponde a la caja
  Wire.onRequest(respuesta);
}

void loop() {
 for(i=3; i>0; i--){
  vecent0[i]=vecent0[i-1];
  vecent1[i]=vecent1[i-1];
 }
 vecent0[0]=analogRead(ent0);
 vecent1[0]=analogRead(ent1);
 valorent0=0.7*vecent0[0]+0.2*vecent0[1]+0.05*vecent0[2]+0.05*vecent0[3];
 valorent1=0.7*vecent1[0]+0.2*vecent1[1]+0.05*vecent1[2]+0.05*vecent1[3];
delay(25);
}
void respuesta(){
  if ((valorent1>valorent0) && (valorent1>60)){
    Wire.write("w"); //mandamos señal cross
  } else if((valorent0>180) && (valorent1>80)){
    Wire.write("e");// rimshot
  } else if(valorent1<80){
    if(valorent0>40){
      Wire.write("r");//golpe flojo
      } else if ((valorent0<130) && (valorent0>40)){
      Wire.write("t");//golpe medio-flojo
    } else if ((valorent0<280) && (valorent0>130)){
      Wire.write("y");//golpe medio
    } else if (valorent0>280){
      Wire.write("u");//golpe fuerte
    }
  } else (Wire.write("m"));
}

```

```
////////////////////////////////////////
/*
CODIGO NODO V 1.2 TOM
Jesús Balsalobre Martínez
TFG: Desarrollo de interfaz para monitorizacion en instrumentos
de percusion a partir de sensores piezoelectricos

*/

////////////////////////////////////////

#include <Wire.h>

int ent0=A0;
int ent1=A1;
int vecent0[]={0,0,0,0};//sensor del parche
//int vecent1[]={0,0,0,0};// sensor del casco
//int valorent1=0;
int valorent0=0;
bool leerjumpers[4];
int i;
int dir;

void setup() {
  pinMode(5, INPUT);
  pinMode(4, INPUT);
  pinMode(3, INPUT);
  pinMode(2, INPUT);
    for(i=0; i<4; i++){
    leerjumpers[i]=digitalRead(i+2);
  }
  dir=leerjumpers[0]+2*leerjumpers[1]+4*leerjumpers[2]+8*leerjumpers[3];
  Wire.begin(dir); //Dirección 2 corresponde a la caja
  Wire.onRequest(respuesta);
}

void loop() {
 for(i=3; i>0; i--){
  vecent0[i]=vecent0[i-1];
 // vecent1[i]=vecent1[i-1];
 }
 vecent0[0]=analogRead(ent0);
// vecent1[0]=analogRead(ent1);
 valorent0=0.7*vecent0[0]+0.2*vecent0[1]+0.05*vecent0[2]+0.05*vecent0[3];
// valorent1=0.7*vecent1[0]+0.2*vecent1[1]+0.05*vecent1[2]+0.05*vecent1[3];
delay(25);
}
void respuesta(){
  if ((valorent0<230) && (valorent0>70)){
      Wire.write("i");//golpe flojo tom
    } else if (valorent0>230){
      Wire.write("o");//golpe fuerte tom
    } else (Wire.write("m"));
}

```

```
////////////////////////////////////////
/*
CODIGO NODO V 1.2 TOM BASE
Jesús Balsalobre Martínez
TFG: Desarrollo de interfaz para monitorizacion en instrumentos
de percusion a partir de sensores piezoelectricos

*/

////////////////////////////////////////

#include <Wire.h>

int ent0=A0;
//int ent1=A1;
int vecent0={0,0,0,0};//sensor del parche
//int vecent1={0,0,0,0};// sensor del casco
//int valorent1=0;
int valorent0=0;
bool leerjumpers[4];
int i;
int dir;

void setup() {
  pinMode(5, INPUT);
  pinMode(4, INPUT);
  pinMode(3, INPUT);
  pinMode(2, INPUT);
    for(i=0; i<4; i++){
    leerjumpers[i]=digitalRead(i+2);
  }
  dir=leerjumpers[0]+2*leerjumpers[1]+4*leerjumpers[2]+8*leerjumpers[3];
  Wire.begin(dir); //Dirección 2 corresponde a la caja
  Wire.onRequest(respuesta);
}

void loop() {
 for(i=3; i>0; i--){
  vecent0[i]=vecent0[i-1];
  //vecent1[i]=vecent1[i-1];
 }
 vecent0[0]=analogRead(ent0);
 //vecent1[0]=analogRead(ent1);
 valorent0=0.7*vecent0[0]+0.2*vecent0[1]+0.05*vecent0[2]+0.05*vecent0[3];
// valorent1=0.7*vecent1[0]+0.2*vecent1[1]+0.05*vecent1[2]+0.05*vecent1[3];
delay(25);
}
void respuesta(){
  if ((valorent0<295) && (valorent0>70)){
      Wire.write("p");//golpe flojo
    } else if (valorent0>295){
      Wire.write("a");//golpe fuerte
    }
  } else (Wire.write("m"));
}

```

```
////////////////////////////////////////
/*
CODIGO NODO V 1.2 CHARLES
Jesús Balsalobre Martínez
TFG: Desarrollo de interfaz para monitorizacion en instrumentos
de percusion a partir de sensores piezoelectricos

*/

////////////////////////////////////////

#include <Wire.h>

int ent0=A0; //plato charles
int ent6=A6; //sensor pedal
int vecent0[]={0,0,0,0};
int vecent6[]={0,0,0,0};
int valorent6=0;
int valorent0=0;
bool leerjumpers[4];
int i;
int dir;

void setup() {
  pinMode(5, INPUT);
  pinMode(4, INPUT);
  pinMode(3, INPUT);
  pinMode(2, INPUT);
    for(i=0; i<4; i++){
    leerjumpers[i]=digitalRead(i+2);
  }
  dir=leerjumpers[0]+2*leerjumpers[1]+4*leerjumpers[2]+8*leerjumpers[3];
  Wire.begin(dir);
  Wire.onRequest(respuesta);
}

void loop() {
 for(i=3; i>0; i--){
  vecent0[i]=vecent0[i-1];
  vecent6[i]=vecent6[i-1];
 }
 vecent0[0]=analogRead(ent0);
 valorent0=0.7*vecent0[0]+0.2*vecent0[1]+0.05*vecent0[2]+0.05*vecent0[3]; //filtro media
movil
 valorent6=0.25*vecent6[0]+0.25*vecent6[1]+0.25*vecent6[2]+0.25*vecent6[3]; //media de
los valores del pedal
delay(25);
}
void respuesta(){
  if (valorent0>100){
    if(valorent6<128){
      Wire.write("j");
    } else if((valorent6<256)&&(valorent6>128)){
      Wire.write("k");
    } else if((valorent6<384)&&(valorent6>256)){
      Wire.write("l");
    } else if((valorent6<512)&&(valorent6>384)){
      Wire.write("z");
    } else if((valorent6<640)&&(valorent6>512)){
      Wire.write("x");
    } else if((valorent6<768)&&(valorent6>640)){
      Wire.write("c");
    } else if((valorent6<896)&&(valorent6>768)){
      Wire.write("v");
    } else if((valorent6<1024)&&(valorent6>896)){
      Wire.write("b");
    }
  } else (Wire.write("m"));
}

```

```
////////////////////////////////////////
/*
CODIGO NODO V 1.2 CRASH
Jesús Balsalobre Martínez
TFG: Desarrollo de interfaz para monitorizacion en instrumentos
de percusion a partir de sensores piezoelectricos

*/

////////////////////////////////////////

#include <Wire.h>

int ent0=A0; //borde del ride
int ent1=A1; //cuerpo del ride
int ent2=A2; //campana del ride
int vecent0[]={0,0,0,0};
int vecent1[]={0,0,0,0};
int vecent2[]={0,0,0,0};
int valorent2=0;
int valorent1=0;
int valorent0=0;
bool leerjumpers[4];
int i;
int dir;

void setup() {
  pinMode(5, INPUT);
  pinMode(4, INPUT);
  pinMode(3, INPUT);
  pinMode(2, INPUT);
    for(i=0; i<4; i++){
    leerjumpers[i]=digitalRead(i+2);
  }
  dir=leerjumpers[0]+2*leerjumpers[1]+4*leerjumpers[2]+8*leerjumpers[3];
  Wire.begin(dir);
  Wire.onRequest(respuesta);
}

void loop() {
 for(i=3; i>0; i--){
  vecent0[i]=vecent0[i-1];
  vecent1[i]=vecent1[i-1];
  vecent2[i]=vecent1[i-1];
 }
 vecent0[0]=analogRead(ent0);
 vecent1[0]=analogRead(ent1);
 vecent2[0]=analogRead(ent2);
 valorent0=0.7*vecent0[0]+0.2*vecent0[1]+0.05*vecent0[2]+0.05*vecent0[3];
 valorent1=0.7*vecent1[0]+0.2*vecent1[1]+0.05*vecent1[2]+0.05*vecent1[3];
 valorent2=0.7*vecent2[0]+0.2*vecent2[1]+0.05*vecent2[2]+0.05*vecent2[3];
delay(25);
}
void respuesta(){
 if((valorent0>valorent1)&&(valorent0>valorent2)){//golpe en el borde del ride
    if((valorent0>100)&&(valorent0<390)){
      Wire.write("d");
    } else if(valorent0>390){
      Wire.write("f");
    }
  } else if((valorent1>valorent0)&&(valorent1>valorent2)){
    if(valorent1>125){
      Wire.write("g");
    }
  } else if((valorent1>valorent0)&&(valorent1>valorent2)){
    if(valorent2>140){
      Wire.write("h");
    }
  } else(Wire.write("m"));
}

```

```
////////////////////////////////////////
/*
CODIGO NODO V 1.2 RIDE
Jesús Balsalobre Martínez
TFG: Desarrollo de interfaz para monitorizacion en instrumentos
de percusion a partir de sensores piezoelectricos

*/

////////////////////////////////////////

#include <Wire.h>

int ent0=A0; //borde del ride
int ent1=A1; //cuerpo del ride
int ent2=A2; //campana del ride
int vecent0[]={0,0,0,0};
int vecent1[]={0,0,0,0};
int vecent2[]={0,0,0,0};
int valorent2=0;
int valorent1=0;
int valorent0=0;
bool leerjumpers[4];
int i;
int dir;

void setup() {
  pinMode(5, INPUT);
  pinMode(4, INPUT);
  pinMode(3, INPUT);
  pinMode(2, INPUT);
    for(i=0; i<4; i++){
    leerjumpers[i]=digitalRead(i+2);
  }
  dir=leerjumpers[0]+2*leerjumpers[1]+4*leerjumpers[2]+8*leerjumpers[3];
  Wire.begin(dir);
  Wire.onRequest(respuesta);
}

void loop() {
 for(i=3; i>0; i--){
  vecent0[i]=vecent0[i-1];
  vecent1[i]=vecent1[i-1];
  vecent2[i]=vecent1[i-1];
 }
 vecent0[0]=analogRead(ent0);
 vecent1[0]=analogRead(ent1);
 vecent2[0]=analogRead(ent2);
 valorent0=0.7*vecent0[0]+0.2*vecent0[1]+0.05*vecent0[2]+0.05*vecent0[3];
 valorent1=0.7*vecent1[0]+0.2*vecent1[1]+0.05*vecent1[2]+0.05*vecent1[3];
 valorent2=0.7*vecent2[0]+0.2*vecent2[1]+0.05*vecent2[2]+0.05*vecent2[3];
delay(25);
}
void respuesta(){
 if((valorent0>valorent1)&&(valorent0>valorent2)){//golpe en el borde del ride
    if((valorent0>100)&&(valorent0<390)){
      Wire.write("d");
    } else if(valorent0>390){
      Wire.write("f");
    }
  } else if((valorent1>valorent0)&&(valorent1>valorent2)){
    if(valorent1>125){
      Wire.write("g");
    }
  } else if((valorent1>valorent0)&&(valorent1>valorent2)){
    if(valorent2>140){
      Wire.write("h");
    }
  } else(Wire.write("m"));
}

```

```
////////////////////////////////////////
/*
CÓDIGO ARDUINO CENTRAL
Jesús Balsalobre Martínez
TFG: Desarrollo de interfaz para monitorizacion en instrumentos
de percusion a partir de sensores piezoelectricos

*/

////////////////////////////////////////

#include <Wire.h>

void noteOn(byte channel, byte pitch, byte velocity) {
  MIDIEvent noteOn = {0x09, 0x90 | channel, pitch, velocity};
  MIDIUSB.write(noteOn);
}

void noteOff(byte channel, byte pitch, byte velocity) {
  MIDIEvent noteOff = {0x08, 0x80 | channel, pitch, velocity};
  MIDIUSB.write(noteOff);
}

bool leerjumpers[4];
int entjumpers[]={10,14,15,16}; //10 lsb, 16 msb
int i;
int numdisp;
char c;
char charrecibidos[17];//número máximo de dispositivos según jumpers
                       //usaremos solo las posiciones necesarias
void setup() {
  for(i=0;i<4;i++){
  pinMode(entjumpers[i], INPUT);
  }
    for(i=0; i<4; i++){
    leerjumpers[i]=digitalRead(i);
  }
  numdisp=leerjumpers[0]+2*leerjumpers[1]+4*leerjumpers[2]+8*leerjumpers[3];
  Wire.begin();
}

void loop() {
  for(i=1;i<(numdisp+1);i++){
    Wire.requestFrom(i, sizeof(char));
    while(Wire.available()){
      charrecibidos[i]=Wire.read();
    }
  } //ya tenemos el valor de todos los nodos
//  for(i=1;i<(numdisp+1);i++){
//    intenviarMIDI[i]=correspondencia([charrecibidos[i]);
//  }
  for(i=1;i<(numdisp+1);i++){
    noteOn(0,correspondencia([charrecibidos[i]),64);
  }
  MIDIUSB.flush();
  delay(30);
}

```

TABLA DE VALORES Y SONIDOS EN FUNCIÓN MIDI ARCORE

B	LUGAR	DESCRIPCIÓN DEL SONIDO	LETRA DE CODIFICACIÓN
0	Nada		m
1	Nada		m
2	Nada		m
3	Nada		m
4	Ride	Pequeño crasheo (entre ritmo y crasheo)	d
5	Nada		m
6	Snare	Golpe normal de caja, algo bajo (quizá no para ritmos)	t
7	Hihat	Golpe normal muy cerrado	
8	Hihat	Igual pero algo menos cerrado	
9	Hihat	Cerrado en la campana	
10	Nada		m
11	Nada		m
12	Nada		m
13	Nada		m
14	Nada		m
15	Nada		m
16	Nada		m
17	Nada		m
18	Nada		m
19	Nada		m
20	Nada		m
21	Nada		m
22	Nada		m
23	Nada		m
24	Nada		m
25	Nada		m
26	Nada		m
27	Nada		m
28	Nada		m
29	Nada		m
30	Nada		m
31	Nada		m
32	Nada		m
33	Nada		m
34	Nada		m
35	Nada		m
36	Kick	Golpe normal y corriente	q
37	Snare	Golpe de aro (rimshot)	e
38	Snare	Golpe normal, algo flojo	r
39	Snare	Golpe de aro, pracicamente igual al otro	
40	Snare	Golpe normal	u
41	Snare	Golpe solo de aro	
42	Snare	Cross	w

43	Snare	Golpe algo flojo pero menos seco	y
44	Snare	Golpe al aro, nada de rimshot	
45	Ride	Golpe normal	g
46	Cymbal 1	Crasheo flojo	
47	Xtra	Golpe normal	
48	Hihat	Sonido al cerrar el pie	
49	Hihat	Golpe más cerrado	j
50	Hihat	Golpe algo más abierto pero menos que el 48	k
51	Hihat	Golpe más abierto que los anteriores	z
52	Hihat	Golpe menos abierto que el 51 pero más que los otros	l
53	Hihat	Golpe en la campana cerrado	x
54	Hihat	Golpe con el charles entreabierto (más abierto que los anteriores)	c
55	Hihat	Golpe entrabierto (más abierto que el 54)	v
56	Hihat	Casi abierto (toca el plato de abajo por poco) (más abierto que el 55)	b
57	Hihat	Equivalente al 54 o 55	
58	Hihat	Golpe en la campana abierto	
59	Hihat	Pisar de golpe para que suene (como al marcar pero sonando)	
60	Ride	Golpe normal (muy parecido al 45)	
61	Ride	Golpe en campana	h
62	Ride	Crasheando un poco (flojo) (más fuerte que el 4)	f
63	Nada		m
64	Nada		m
65	Tom 4	Golpe normal/flojo	p
66	Tom 4	Golpe normal/fuerte	a
67	Tom 3	Golpe normal/flojo	
68	Tom3	Golpe normal pero atenuado (un plato encima quizás)	
69	Tom 2	Golpe normal/flojo	i
70	Tom 2	Golpe normal/fuerte (plato otra vez?)	o
71	Tom 1	Golpe flojo	
72	Tom 1	Golpe normal	
73	Nada		m
74	Nada		m
75	Snare	Choque de dos baquetas	
76	Nada		m
77	Cymbal 1	Crasheo normal (algo flojo)	s
78	Nada		m
79	Cymbal 2	Crasheo normal (algo flojo)	
80	Nada		m
81	Cymbal 3	Crasheo normal	