

**UNIVERSIDAD POLITÉCNICA DE MADRID**

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA  
Y SISTEMAS DE TELECOMUNICACIÓN**



**PROYECTO FIN DE CARRERA**

**DISPARADOR MIDI PARA BATERÍA**

Autor

Manuel Antonio Ballesteros Carballo

Tutor

Eduardo Nogueira Díaz

JULIO 2014



*A mi novia Aida, por su cariño y comprensión.*

*A mi familia, por permitirme estudiar lo que siempre he deseado.*

*A mis amigos y compañeros, por compartir conmigo esta senda, que es mi vida, y sus momentos inolvidables.*

*“Sin música, la vida no tendría sentido”* Friedrich Nietzsche

*“La ciencia será siempre una búsqueda, jamás un descubrimiento real. Es un viaje, nunca una llegada”* Karl Popper

# RESUMEN

---

La batería acústica es un instrumento de percusión muy complejo que requiere altas dosis de coordinación, interdependencia y musicalidad.

Los músicos necesitan nuevas formas de desarrollarse y expresarse. Buscando caminos que mejoren su comprensión. Así también los aprendices requieren herramientas para poder disfrutar del instrumento musical.

Construir un disparador MIDI mejora esas prestaciones. Aumenta la cantidad de sonidos, es adaptable a la forma de tocar y simplifica los conceptos evitando problemas a los principiantes.

Actualmente existen escasos disparadores MIDI en el mercado. En su lugar se presentan módulos electrónicos con altas características, complejos de utilizar y con alto coste.

El proyecto a implementar consiste en un disparador MIDI, un equipo cuyo fin es captar los impactos que efectúa un músico; un percusionista.

En el proyecto se estudian y analizan las problemáticas a las que se enfrenta este tipo de dispositivo, se propondrá distintas soluciones de concepto y, finalmente se construirá dicho sistema que será probado en un ordenador personal. Con un ideal en mente, emular una batería acústica y ser un proyecto abierto en el sentido de ser modulable y ampliable.

# ABSTRACT

---

Drum set is a very complex percussion instrument that requires a lot of coordination, interdependence and musicality.

Musicians need new ways to develop and express themselves, as well as learners require tools to enjoy the musical instrument.

Build a MIDI drum trigger improves these benefits. It increases the amount of sounds, is customizable and simplifies problems for drummer beginners.

There are currently a few MIDI triggers on the market. Instead, electronic modules with high features, complex to use and they are sold at high cost.

The aim of the project is to implement a MIDI drum trigger, a kit to capture the impact that percussionist makes. The project studies and analyzes the problems which this type of device. Different solutions concept will be proposed and finally the system will be built. With an ideal in mind, emulate an acoustic drum.

# INDICE

---

|            |  |           |
|------------|--|-----------|
| <b>1</b>   | <b>INTRODUCCIÓN.....</b>                             | <b>17</b> |
| <b>1.1</b> | <b>Propósito de la memoria .....</b>                 | <b>19</b> |
| 1.1.1      | Consideraciones del autor .....                      | 20        |
| 1.1.2      | Objetivos .....                                      | 21        |
| 1.1.3      | Organización del libro .....                         | 22        |
| 1.1.4      | Herramientas utilizadas .....                        | 23        |
| <b>1.2</b> | <b>Introducción al mundo de la batería.....</b>      | <b>24</b> |
| 1.2.1      | Un principio; el sonido .....                        | 24        |
| 1.2.2      | La batería acústica .....                            | 27        |
| 1.2.2.1    | Sonido y componentes .....                           | 28        |
| 1.2.2.2    | ¿Cómo se toca la batería?.....                       | 30        |
| 1.2.3      | Las Baterías Electrónicas .....                      | 32        |
| 1.2.3.1    | Funcionamiento y componentes .....                   | 33        |
| 1.2.3.2    | Ventajas.....  | 35        |
| 1.2.3.3    | Limitaciones .....                                   | 35        |
| 1.2.3.4    | Estudio de mercado.....                              | 37        |
| 1.2.4      | Sintetizador.....                                    | 42        |
| 1.2.4.1    | Particularidades.....                                | 42        |
| 1.2.4.2    | Modalidades de síntesis .....                        | 43        |
| 1.2.4.3    | Software de batería basado en tabla de ondas .....   | 44        |
| <b>2</b>   | <b>ANÁLISIS TEÓRICO.....</b>                         | <b>51</b> |
| <b>2.1</b> | <b>Protocolo comunicaciones.....</b>                 | <b>53</b> |
| 2.1.1      | Introducción a MIDI .....                            | 53        |
| 2.1.1.1    | ¿Por qué nació MIDI? Ventajas y breve historia ..... | 53        |
| 2.1.2      | Interfaz física .....                                | 55        |
| 2.1.2.1    | Conexionado .....                                    | 55        |
| 2.1.2.2    | Aparatos .....                                       | 57        |
| 2.1.2.3    | Interconexión entre aparatos.....                    | 57        |
| 2.1.3      | Formato de los mensajes MIDI.....                    | 59        |
| 2.1.3.1    | Canales MIDI.....                                    | 59        |
| 2.1.3.2    | Modos MIDI .....                                     | 60        |
| 2.1.3.3    | Mensajes MIDI .....                                  | 61        |
| 2.1.4      | General MIDI .....                                   | 64        |
| 2.1.5      | MIDI over USB .....                                  | 66        |

|            |  |            |
|------------|--|------------|
| 2.1.5.1    | Fundamentos USB .....  | 66         |
| 2.1.5.2    | Conversor USB-MIDI .....   | 66         |
| 2.1.6      | Ejemplo práctico MIDI .....                                      | 69         |
| <b>2.2</b> | <b>Sensores.....</b>   | <b>72</b>  |
| 2.2.1      | Sensores de presión .....  | 72         |
| 2.2.2      | Sensores piezoelectricos .....                                   | 73         |
| 2.2.2.1    | Ventajas .....   | 73         |
| 2.2.2.2    | Limitaciones .....   | 74         |
| 2.2.2.3    | Comportamiento eléctrico de los materiales piezoelectricos. .... | 74         |
| 2.2.2.4    | Elección del piezoelectrico .....                                | 79         |
| <b>2.3</b> | <b>El microcontrolador.....</b>                                  | <b>81</b>  |
| 2.3.1      | Plataformas de Desarrollo .....                                  | 81         |
| 2.3.2      | El Atmega328p.....   | 85         |
| 2.3.2.1    | Nucleo AVR de Atmel .....  | 85         |
| 2.3.2.2    | Puertos de entrada/salida .....                                  | 87         |
| 2.3.2.3    | Periféricos usados .....   | 87         |
| <b>2.4</b> | <b>El lenguaje Arduino.....</b>                                  | <b>90</b>  |
| 2.4.1      | Lenguaje C .....   | 90         |
| 2.4.1.1    | Ventajas de lenguaje C sobre Ensamblador .....                   | 91         |
| 2.4.2      | Bootloader .....   | 92         |
| 2.4.3      | Instalación entorno Arduino .....                                | 93         |
| 2.4.4      | Librerías utilizadas de Arduino .....                            | 94         |
| <b>3</b>   | <b>DESARROLLO DEL PROYECTO.....</b>                              | <b>97</b>  |
| <b>3.1</b> | <b>Disparador MIDI .....</b>                                     | <b>99</b>  |
| 3.1.1      | Descripción.....   | 99         |
| 3.1.2      | Especificaciones .....   | 102        |
| <b>3.2</b> | <b>Estructura Mecánica .....</b>                                 | <b>103</b> |
| 3.2.1      | Definición de pad .....  | 103        |
| 3.2.2      | Análisis de Pads Comerciales .....                               | 104        |
| 3.2.2.1    | Pads Yamaha ep-75.....   | 104        |
| 3.2.2.2    | Pad Yamaha PCY65 .....   | 106        |
| 3.2.3      | Diseño y construcción de Pads .....                              | 109        |
| 3.2.3.1    | Compra de materiales.....  | 109        |
| 3.2.3.2    | Modificación pad de prácticas.....                               | 110        |
| 3.2.3.3    | Construcción de pad con Goma EVA.....                            | 112        |
| 3.2.3.4    | Construcción de pad con contrachapado .....                      | 113        |

|            |  |            |
|------------|--|------------|
| 3.2.3.5    | Construcción de pad de plato .....                                       | 115        |
| 3.2.4      | Análisis del equipamiento completo .....                                 | 117        |
| <b>3.3</b> | <b>    Análisis de la respuesta de los sensores del kit DD-508 .....</b> | <b>119</b> |
| 3.3.1      | Caracterizar piezoelectrónico .....                                      | 119        |
| 3.3.1.1    | Análisis osciloscopio.....   | 120        |
| 3.3.2      | Pad de caja .....  | 122        |
| 3.3.3      | Pad de plato.....  | 124        |
| 3.3.4      | Pad de bombo .....   | 126        |
| <b>3.4</b> | <b>    Diseño Acondicionador .....</b>                                   | <b>129</b> |
| 3.4.1      | Solución 1: Acondicionador Pasivo.....                                   | 129        |
| 3.4.1.1    | Introducción .....   | 129        |
| 3.4.1.2    | Estructura .....   | 130        |
| 3.4.1.3    | Simulación .....   | 131        |
| 3.4.2      | Solución 2: Acondicionador Activo.....                                   | 133        |
| 3.4.2.1    | Introducción .....   | 133        |
| 3.4.2.2    | Estructura .....   | 133        |
| 3.4.2.3    | Elección del Amplificador Operacional.....                               | 134        |
| 3.4.2.4    | Simulación .....   | 136        |
| 3.4.3      | Calculo de $\tau$ .....  | 137        |
| 3.4.4      | Construcción PCB .....   | 138        |
| 3.4.4.1    | EAGLE.....   | 138        |
| 3.4.4.2    | Modelo 3D.....   | 140        |
| 3.4.4.3    | Envío de PCB a la fábrica .....  | 141        |
| <b>3.5</b> | <b>    Software y algoritmia .....</b>                                   | <b>143</b> |
| 3.5.1      | Código y diferentes versiones .....                                      | 143        |
| 3.5.1.1    | Versión 0.1 .....  | 143        |
| 3.5.1.2    | Versión 0.5.....   | 143        |
| 3.5.1.3    | Versión 1.0 .....  | 143        |
| 3.5.1.4    | Versión 1.1 .....  | 144        |
| 3.5.1.5    | Version 1.2.....   | 145        |
| 3.5.1.6    | Versión 1.3.....   | 145        |
| 3.5.1.7    | Versión 1.4 .....  | 145        |
| 3.5.1.8    | Versión 2.0.....   | 145        |
| 3.5.2      | Problemática sensibilidad .....  | 146        |
| 3.5.2.1    | Solución 1: Truncamiento con desplazamiento. ....                        | 146        |
| 3.5.2.2    | Solución 2: Aproximar el valor por la pendiente. ....                    | 147        |
| 3.5.2.3    | Conclusión .....   | 148        |

|            |  |            |
|------------|--|------------|
| 3.5.3      | Parámetros Roland/DD508 .....                          | 150        |
| 3.5.3.1    | Tipos de parámetros .....                              | 150        |
| 3.5.4      | El problema de la latencia .....                       | 153        |
| 3.5.4.1    | Latencia mínima tolerable por el oído humano .....     | 153        |
| 3.5.4.2    | ASIO.....  | 153        |
| <b>3.6</b> | <b>Interfaz gráfica, controles de teclado .....</b>    | <b>156</b> |
| 3.6.1      | Interfaz gráfica .....                                 | 156        |
| 3.6.1.1    | Hitachi HD44780 .....                                  | 157        |
| 3.6.1.2    | Conexión.....  | 158        |
| 3.6.2      | Controladores de teclado .....                         | 159        |
| 3.6.2.1    | Encoder .....  | 159        |
| 3.6.2.2    | Programación mediante interrupción .....               | 161        |
| 3.6.3      | Menú de usuario .....                                  | 163        |
| 3.6.4      | Memoria de parámetros .....                            | 164        |
| 3.6.5      | Carcasa protectora .....                               | 165        |
| <b>3.7</b> | <b>Controlador del charles.....</b>                    | <b>167</b> |
| 3.7.1      | Solución 1: Pedal DD508 .....                          | 168        |
| 3.7.1.1    | Proceso de análisis .....                              | 168        |
| 3.7.1.2    | Esquema eléctrico .....                                | 170        |
| 3.7.1.3    | Algoritmo.....   | 170        |
| 3.7.2      | Solución 2: Pulsador .....                             | 172        |
| 3.7.2.1    | Tiempo entre transiciones .....                        | 172        |
| 3.7.2.2    | Algoritmo.....   | 173        |
| 3.7.3      | Solución 3: Sensor FSR.....                            | 174        |
| 3.7.3.1    | Introducción .....                                     | 174        |
| 3.7.3.2    | Espuma resistiva .....                                 | 175        |
| 3.7.3.3    | Funcionamiento .....                                   | 176        |
| 3.7.4      | Conclusión .....                                       | 177        |
| <b>4</b>   | <b>ELABORACION DE ENSAYOS Y RESULTADOS .....</b>       | <b>179</b> |
| <b>4.1</b> | <b>Resultado final .....</b>                           | <b>181</b> |
| 4.1.1      | Manual de usuario .....                                | 181        |
| <b>4.2</b> | <b>Testeo del funcionamiento del prototipo .....</b>   | <b>183</b> |
| 4.2.1      | Test eléctrico y lógico .....                          | 183        |
| 4.2.1.1    | Entrada y salida del acondicionador.....               | 183        |
| 4.2.1.2    | Salida del transmisor .....                            | 184        |
| 4.2.1.3    | Salida del acondicionador y salida del transmisor..... | 184        |

|            |  |            |
|------------|--|------------|
| 4.2.1.4    | USBPcap y Wireshark .....                              | 185        |
| 4.2.2      | Verificación con sampleador.....                       | 187        |
| 4.2.2.1    | Material Utilizado.....                                | 187        |
| 4.2.2.2    | Configurar Kontakt 5 Player y DrumMic'a .....          | 188        |
| 4.2.2.3    | Configurar BFD2 .....                                  | 190        |
| 4.2.3      | DTXmanía .....   | 192        |
| 4.2.3.1    | Procedimiento de configuración.....                    | 192        |
| 4.2.3.2    | Tocar una canción .....                                | 193        |
| 4.2.4      | Difusión .....   | 194        |
| 4.2.4.1    | Participación en jornadas y congresos .....            | 194        |
| <b>5</b>   | <b>CONCLUSIONES Y FUTUROS DESARROLLOS.....</b>         | <b>197</b> |
| <b>5.1</b> | <b>Conclusiones finales.....</b>                       | <b>199</b> |
| <b>5.2</b> | <b>Futuros desarrollos .....</b>                       | <b>200</b> |
| <b>6</b>   | <b>REFERENCIAS.....</b>                                | <b>201</b> |
| <b>6.1</b> | <b>Referencias bibliográficas .....</b>                | <b>203</b> |
| <b>6.2</b> | <b>Referencias digitales .....</b>                     | <b>206</b> |
| <b>7</b>   | <b>ANEXOS.....</b>                                     | <b>209</b> |
| Anexo A:   | Pruebas ejecutadas con Arduino .....                   | 211        |
| Anexo B:   | Hojas de características .....                         | 221        |
| Anexo C:   | Acreditación Conferencia.....                          | 226        |
| Anexo D:   | Código de programa .....                               | 227        |
| Anexo E:   | Preguntas frecuentes -Manual del usuario [Drola] ..... | 236        |

# INDICE DE FIGURAS

---

|  |    |
|--|----|
| Figura 1: Armónicos y fundamentales de diferentes instrumentos [Peto01] .....          | 26 |
| Figura 2: Diferencia de Timbres en ondas acústicas [Peto01] .....                      | 26 |
| Figura 3: Batería acústica sonorizada para directo.....                                | 27 |
| Figura 4: Partes de la Batería Acústica[Bla05].....                                    | 28 |
| Figura 5: Frecuencias fundamentales de la batería[Cuco03].....                         | 29 |
| Figura 6: Agarre de baquetas[Stao05].....  | 31 |
| Figura 7: Patrón básico en 4 por 4 de pop/rock [Pino01].....                           | 31 |
| Figura 8: Similitudes entre batería acústica y electrónica [Dthom] .....               | 32 |
| Figura 9: Pads de plato, tom, bombo y hi-hat [Drola] .....                             | 34 |
| Figura 10: Módulo de la batería modelo TD-30 [Drola].....                              | 34 |
| Figura 11: Forma óptima de golpear los platos[Dbata].....                              | 37 |
| Figura 12: Diagrama bloques proceso de síntesis .....                                  | 43 |
| Figura 13: Grafica ADSR [Zor98].....   | 44 |
| Figura 14: Panel de Control Addictive Drums [Dxlna] .....                              | 45 |
| Figura 15: Panel de control BFD2 [Dfxpa] .....   | 46 |
| Figura 16: Panel de Control Ezdrummer [Dtoon].....                                     | 47 |
| Figura 17: Panel de Control Drummic`a [Dsenn] .....                                    | 48 |
| Figura 18: Panel de Control Hydrogen [Dhydr] .....                                     | 49 |
| Figura 19: Forma de onda de sonido de bombo .....                                      | 49 |
| Figura 20: ¿Cómo conectarías todos estos aparatos sin el protocolo MIDI? [Dmidi] ..... | 54 |
| Figura 21: Conectores MIDI .....   | 55 |
| Figura 22: Circuito entrada MIDI típico [Aso95] .....                                  | 56 |
| Figura 23: Circuito de salida MIDI [Aso95] .....                                       | 56 |
| Figura 24: Conexión entre diferentes dispositivos [Dmidi] .....                        | 58 |
| Figura 25: Ejemplo utilizando los canales .....  | 59 |
| Figura 26: Conexión MIDI-USB [Dedru] y Diagrama de bloques [Du2m] .....                | 67 |
| Figura 27: Vista del dispositivo y de la cara de componentes del conversor USB .....   | 67 |
| Figura 28: Esquema eléctrico conversor MIDI-USB [Dmfmo] .....                          | 68 |
| Figura 29: Piano Casio CTK-431:[Dedru].....  | 69 |
| Figura 30: Panel de MIDI-OX con ejemplo de tecla pulsada.....                          | 70 |
| Figura 31: Conexión entre salida de piano y osciloscopio .....                         | 70 |
| Figura 32: Análisis de la trama leída del osciloscopio .....                           | 71 |
| Figura 33: Esquema sensores de presión [Pal03].....                                    | 72 |
| Figura 34: Efecto piezoelectrónico[Cupoo].....   | 73 |
| Figura 35: Comportamiento eléctrico piezoelectrónico [Mon10] .....                     | 74 |
| Figura 36: Analogía mecánico-eléctrica [Mon10] .....                                   | 76 |
| Figura 37: Circuito cerrado sensor piezoelectrónico [Mon10] .....                      | 77 |

|   |     |
|---|-----|
| Figura 38: Respuesta en frecuencia sensor piezoeléctrico [Dwiki] .....                          | 78  |
| Figura 39: Sensor cerámico piezoeléctrico.....  | 79  |
| Figura 40: Pad Yamaha modelo XP100D y vista interior piezoeléctrico .....                       | 80  |
| Figura 41: Plataformas de desarrollo analizadas .....   | 82  |
| Figura 42: Núcleo de Atmega 328P [Atmo9].....   | 85  |
| Figura 43: Ejecución en paralelo de instrucciones[Atmo9].....                                   | 87  |
| Figura 44: Esquema eléctrico entrada ADC.....   | 88  |
| Figura 45: Código en el que se accede a registro a nivel de bit .....                           | 91  |
| Figura 46: El uso del bootloader [DNeot] .....  | 92  |
| Figura 47: Placa de desarrollo Arduino modelo UNO [Dardu].....                                  | 93  |
| Figura 48: Panel de control del entorno de desarrollo de Arduino .....                          | 93  |
| Figura 49: Proyecto disparador MIDI .....   | 99  |
| Figura 50: Diagrama de bloques disparador MIDI .....  | 100 |
| Figura 51: Interior del pad comercial modelo EP75 .....   | 105 |
| Figura 52: Disposición de los piezoeléctricos del EP75.....                                     | 105 |
| Figura 53: Esquema de conexiones modelo EP75 .....  | 106 |
| Figura 54: Forma de onda osciloscopio modelo EP75.....  | 106 |
| Figura 55: Pad de plato PCY65 .....   | 107 |
| Figura 56: Interior del pad de plato PCY65 .....  | 107 |
| Figura 57: Esquema de conexiones plato PCY65 .....  | 108 |
| Figura 58: Forma onda osciloscopio .....  | 108 |
| Figura 59: Pentagrama del rudimento Single Ratamacue [Stao5] .....                              | 110 |
| Figura 60: Despiece y modificación pad de prácticas remo [Dhell] .....                          | 111 |
| Figura 61: Resultado final pad remo .....   | 111 |
| Figura 62: Construcción de Pad con goma EVA .....   | 112 |
| Figura 63: Despiece y construcción de pad completo [Tomo3].....                                 | 114 |
| Figura 64: Fotografía interior pad contrachapado .....  | 114 |
| Figura 65: Fotografía resultado construcción pad de plato .....                                 | 115 |
| Figura 66: Dibujo batería electrónica modelo DD508 [Med12] .....                                | 117 |
| Figura 67: Dato en formato .csv.....  | 120 |
| Figura 68: Vista frontal y lateral pad de caja estéreo .....                                    | 122 |
| Figura 69: Vista interior pad caja .....  | 122 |
| Figura 70: Esquema eléctrico pad de caja .....  | 122 |
| Figura 71: Señal eléctrica producida por un golpe de fuerte intensidad en pad de caja.....      | 123 |
| Figura 72: Señal eléctrica producida por un golpe de intensidad débil en pad de caja .....      | 123 |
| Figura 73: Señal producida en el conector estéreo .....   | 124 |
| Figura 74: Vista frontal y lateral pad plato.....   | 124 |
| Figura 75: Vista interna pad de caja .....  | 125 |
| Figura 76: Señal eléctrica producida por un golpe de intensidad débil en pad de plato .....     | 125 |
| Figura 77: Señal eléctrica producida por un golpe de fuerte intensidad en el pad de plato ..... | 125 |

|  |     |
|--|-----|
| Figura 78: Esquema eléctrico pad de plato .....  | 126 |
| Figura 79: Vista frontal y lateral pad bombo .....   | 126 |
| Figura 80: Vista interior pad bombo .....  | 127 |
| Figura 81: Esquema eléctrico Vista interior pad bombo .....                                  | 127 |
| Figura 82: Señal eléctrica producida por un golpe de intensidad débil en pad debombo .....   | 127 |
| Figura 83: Señal eléctrica producida por un golpe de fuerte intensidad en pad de bombo ..... | 128 |
| Figura 84: Acondicionador de la solución 1 [Tomo3] .....                                     | 130 |
| Figura 85: Limitador con diodos Schottky .....   | 131 |
| Figura 86: Simulación acondicionador 1 .....   | 132 |
| Figura 87: Construcción PCB del acondicionador 1 .....                                       | 132 |
| Figura 88: Acondicionador 2 .....  | 134 |
| Figura 89: Simulación acondicionador 2 .....   | 136 |
| Figura 90: Distintos tamaños de encapsulado .....  | 139 |
| Figura 91: Layout de circuito acondicionador .....   | 140 |
| Figura 92: Circuito acondicionador en modelado 3D .....                                      | 140 |
| Figura 93: Revisión del resultado de los gerber con Viewplot .....                           | 142 |
| Figura 94: Resultado final PCB acondicionador .....  | 142 |
| Figura 95: Diagrama de flujo programa principal .....  | 144 |
| Figura 96: Correspondencia de los matices en MIDI .....                                      | 146 |
| Figura 97: Grafica y fórmulas de conversión .....  | 147 |
| Figura 98: Parámetro velocity curve [Rolo4] .....  | 151 |
| Figura 99: Parámetro Scantime [Rolo4] .....  | 151 |
| Figura 100: Parámetro Treshold [Rolo4] .....   | 151 |
| Figura 101: Parámetro Mask Time [Rolo4] .....  | 152 |
| Figura 102: Parámetro Retrigger [Rolo4] .....  | 152 |
| Figura 103: Panel de control de ASIO4ALL .....   | 155 |
| Figura 104: Pines del módulo LCD Hitachi HD44780 [Dardu] .....                               | 158 |
| Figura 105: Encoder [Dneot] .....  | 160 |
| Figura 106: Problemática de rebotes en encoder .....   | 161 |
| Figura 107: Circuito anti-rrebotes para encoder .....  | 161 |
| Figura 108: Diagrama de estados del menú .....   | 163 |
| Figura 109: Prototipo completo en caja de metal .....  | 165 |
| Figura 110: Impresora 3D modelo Rep Rap prusa i2 .....                                       | 166 |
| Figura 111: Herraje en batería acústica y su imitación en la electrónica .....               | 167 |
| Figura 112: Fotografia interior del pedal HH65 .....   | 169 |
| Figura 113: Esquema eléctrico con Multisim de pedal comercial .....                          | 170 |
| Figura 114: Diagrama de flujo algoritmo HH65 .....   | 171 |
| Figura 115: Diagrama de la conexión de los contactos del pedal de HH .....                   | 172 |
| Figura 116: Algoritmo pedal solución 2 .....   | 173 |
| Figura 117: Gráfico y símbolo de sensor FSR [Dstan] .....                                    | 174 |

|   |     |
|---|-----|
| Figura 118: Sensor FSR de construcción artesanal .....                                | 175 |
| Figura 119: Medida de la variación del sensor FSR .....                               | 176 |
| Figura 120: Circuito eléctrico FSR.....   | 176 |
| Figura 121: Vista interna del prototipo de disparador MIDI .....                      | 181 |
| Figura 122: Puntos de test de las diferentes medidas.....                             | 183 |
| Figura 123: Prueba de entrada y salida del acondicionador .....                       | 183 |
| Figura 124: Comando recibido para prueba de entrada y salida del acondicionador ..... | 183 |
| Figura 125: Prueba de UART .....  | 184 |
| Figura 126: Comando recibido para prueba de UART .....                                | 184 |
| Figura 127: Latencia del disparador MIDID .....                                       | 185 |
| Figura 128: Trama USB enviada por el disparador .....                                 | 186 |
| Figura 129: Panel de Control Kontakt 5 Player.....                                    | 189 |
| Figura 130: Menu Configuración del audio en Kontakt 5 Player.....                     | 189 |
| Figura 131: BFD2 configuración completa .....   | 191 |
| Figura 132: Configuracion de DTXmania.....  | 193 |
| Figura 133: Captura de pantalla del testeо de prototipo en DTXmania .....             | 193 |
| Figura 134: Horario del domingo en OSHWdem .....                                      | 195 |
| Figura 135: Presentación del proyecto en OSHWdem .....                                | 196 |
| Figura 136: Señal sinusoidal .....  | 217 |
| Figura 137: Espectro unilateral señal sinusoidal .....                                | 217 |
| Figura 138: Señal recogida con el ADC del pad .....                                   | 218 |
| Figura 139: Espectro unilateral del pad .....   | 218 |
| Figura 140: Señal recogida con el ADC del pad construido .....                        | 219 |
| Figura 141: Espectro unilateral pad .....   | 219 |
| Figura 142: Señal recogida con el ADC del pad caja .....                              | 219 |
| Figura 143: Espectro pad caja .....   | 220 |

# INDICE DE TABLAS

---

|  |     |
|--|-----|
| Tabla 1: Cualidades del sonido .....   | 25  |
| Tabla 2: Comparación de los modelos de batería eléctrica más representativos .....             | 41  |
| Tabla 3: Modos disponibles en protocolo MIDI.....  | 60  |
| Tabla 4: Todos los mensajes de estado posibles [Leh93] .....                                   | 61  |
| Tabla 5: Correspondencia valor numérico y nota musical en protocolo MIDI [Tomio3] .....        | 62  |
| Tabla 6: Sonidos de batería definidos por General MIDI[Dmidi] .....                            | 65  |
| Tabla 7: Comparación de características de los distintos microcontroladores .....              | 83  |
| Tabla 8: Comparación plataformas de desarrollo .....   | 84  |
| Tabla 9: Lista de materiales y lugar de su compra .....  | 109 |
| Tabla 10: Costes y materiales utilizados en pad de prácticas Remo .....                        | 112 |
| Tabla 11: Costes y materiales utilizados en pad de goma EVA .....                              | 113 |
| Tabla 12: Costes y materiales utilizados en pad de contrachapado .....                         | 115 |
| Tabla 13: Costes y materiales utilizados en el pad de plato .....                              | 116 |
| Tabla 14: Medidas pad de caja.....   | 123 |
| Tabla 15: Medidas pad plato.....   | 126 |
| Tabla 16: Medidas pad bombo .....  | 128 |
| Tabla 17: Comparación de empresas dedicadas a la construcción de prototipos electrónicos ..... | 141 |
| Tabla 18: Comparación de diferentes interfaces táctiles .....                                  | 159 |
| Tabla 19: Parámetros recomendados en el pad de caja conectando el prototipo.....               | 164 |
| Tabla 20: Tensiones y códigos del pedal de HH a diferentes pulsaciones.....                    | 169 |

# 1 INTRODUCCIÓN

---



## **1.1 Propósito de la memoria**

Esta memoria del proyecto fin de carrera de Manuel Antonio Ballesteros Carballo anhe-  
la ser una recogida de información de todos los problemas acontecidos en el mismo y su  
correspondiente resolución. Así como la descripción de la parte teórica utilizada en el pro-  
ceso.

La ordenación ha intentado ser minuciosa pero con sus pequeños errores frutos del  
tiempo y, de las limitaciones del humilde autor. Se ha buscado la sencillez antes que la  
complejidad primando la calidad frente a la cantidad. Es un proyecto sin aditivos.

El lenguaje utilizado es claro sin llegar a ser banal. Una lectura cómoda. Se ha fijado  
como un ideal allanar el camino a aquella persona que se aventure en la futura mejora de  
este sistema. Puesto que existe mucha información, pero a veces es dispersa e incluso  
errónea. Y sin olvidar a aquella persona que desea únicamente construirlo, para disfrutar-  
lo en su casa.

Se trata de un texto que demuestra que se han adquirido los conocimientos y aptitudes  
propias de un Ingeniero Técnico de Telecomunicación poseyendo una amplitud de miras  
tanto teórica como experimental.

### **1.1.1 Consideraciones del autor**

Es necesario aclarar que es un proyecto planteado por iniciativa del alumno, tal y como recoge la normativa de la Universidad Politécnica de Madrid, en el que con la supervisión y los múltiples consejos de mi tutor Don Eduardo Nogueira, se ha llegado a buen término. Es difícil que un profesor colabore en este tipo de proyectos por lo que le estaré siempre agradecido.

Porque en él hay incluidas muchas motivaciones personales:

- Soy aspirante a Ingeniero en la especialidad de Sistemas Electrónicos. Esta especialidad ha sido desde siempre mi pasión y he intentado que se muestre en esta memoria. Recordando el momento en que de niño quise descubrir la magia de la electricidad a disfrutar, en estos momentos, de la construcción de un diseño personal elaborado a medida.
- Mis vinculaciones con la música son varias. Soy baterista amateur y fundador de la Asociación Juvenil de Música “Notas y Ruido”.

El camino ha sido duro lleno de baches pero tras concluir la etapa nos queda la satisfacción personal.

### **1.1.2 Objetivos**

Los fines de este proyecto y recogidos en la memoria son:

- Asimilación de conceptos imprescindibles: piezoeléctricos, MIDI, microcontroladores, sintetizadores y batería acústica. Así como, estudio teórico de casos particulares tales como respuesta impulsiva en piezoeléctrico, optimización de recursos en un sistema digital, acondicionamiento de señales...
- Adaptación y síntesis de características de productos comerciales existentes en batería electrónica.
- Realización de un prototipo de complejidad:
  - Elaboración de unas especificaciones analizado un problema.
  - Diseño del dispositivo electrónico.
  - Envío de PCB a fábrica especializada y adquisición de componentes.
  - Construcción.
  - Codificación y puesta en funcionamiento.
  - Análisis de resultados y mejoras constantes.
- Utilización de programas y recursos que son los más característicos y mejor optimizados en sus áreas. Utilizados por muchos ingenieros en sus labores diarias.

### **1.1.3 Organización del libro**

La memoria se ha estructurado de la siguiente forma: 14 capítulos sobre distinta temática englobados en 5 secciones que se relacionan en las partes en las que se abordó este proyecto. Un primer enfoque con “Introducción”, estudio en profundidad de conceptos con “Análisis Teórico”. Tras ellas la parte más técnica con “Desarrollo del Proyecto” para una vez terminado, realizar “Elaboración de ensayos y resultados” y finalizar con “Conclusiones y futuros desarrollos”. También se encuentra la parte “Referencias” y “Anexos”.

Explicando la temática de los capítulos se dispone:

En la primera sección es una toma de contacto sobre el universo de la batería como instrumento musical. Partiendo desde la concepción de sonido en él se planteará un escenario como es la conversión de este instrumento a uno electrónico.

La sección 2 se divide en cuatro capítulos. El capítulo 2.1 centrado en el protocolo de comunicaciones para instrumentos digitales por excelencia (MIDI). Se hablará brevemente de su nacimiento así como de lo fundamental para trabajar con él. Se incluye una pequeña demostración y proponer algo original a la documentación consultada. El capítulo 2.2 estudia los sensores disponibles para captar la interpretación de un percusionista y se centrará, en los sensores más utilizados. El capítulo 2.3 versa sobre los microcontroladores del mercado y en función de sus peculiaridades (memoria, periféricos, precio) se elegirá uno para trabajar y se explicarán sus características. El capítulo 2.4 sobre el lenguaje de programación y sobre aspectos nuevos que introduce Arduino.

La sección 3 está formada por 7 capítulos. En 3.1 se definirán conceptos y explicaciones sobre el disparador MIDI. En 3.2 versará sobre la estructura mecánica y, las versiones y construcciones elaboradas a la par. En el capítulo 3.3 se realizará un estudio de la respuesta obtenida al impactar unas baquetas sobre las distintas superficies. En el capítulo 3.4 se diseñará el acondicionador de señal. En el capítulo 3.5 el software y en el capítulo 3.6 el resto de los bloques de digitalización para finalizar sobre el controlador de pedal de la batería.

La sección 4 desentrañará los datos y pruebas realizadas para verificar el funcionamiento. Se utilizarán programas para reproducción de audio en los que se explicará su configuración.

En las últimas secciones se ofrecerán una serie de conclusiones y líneas de trabajo futura, se expondrá la documentación consultada y los anexos con información adicional.

#### **1.1.4 Herramientas utilizadas**

En la realización de este proyecto se ha contado con múltiples herramientas que se podrían englobar en:

- Utensilios: Una serie de prototipos han sido construidos por lo que se han utilizado tanto productos de bricolaje (sierra, martillo, adhesivo, etc.) como aquellas más especializadas en un técnico de electrónica (multímetro, osciloscopio, estación de soldadura).
- Software: Diversos programas correspondientes a distintas áreas. Se hablará de ellos en sus apartados correspondientes pero por citarlos; Multisim (Análisis de esquemas eléctricos), Eagle (Diseño de circuito impreso), MIDI-OX (monitorear secuencias MIDI), Matlab (cálculos matemáticos), SketchUp (modelado 3D) y los DAW y sampler de batería Kontakt 5, BFD2 e Hydrogen.

Por último y con un tono más distendido la ilusión de un niño y la paciencia de un monje budista.

## 1.2 Introducción al mundo de la batería

El ser humano se ha acostumbrado a escuchar música. La disfruta y le emociona. Desde melodías pegadizas hasta sinfonías complejas. Es capaz de transportar. Parece complejo pero el compositor utiliza unas piezas elementales (7 notas) y el tiempo. Estos dos los introduce con la ayuda de los silencios, en una sucesión creando magia. Pues desde una base conocida se puede llegar muy lejos.

Dado que se quiere elaborar un desarrollo en profundidad será necesario cimentar sobre unos principios. Se empezará definiendo el sonido y alguna de sus características. A continuación se explicarán aspectos relacionados de la batería acústica como instrumento musical y se darán unas pinceladas a un miembro de su familia: la batería electrónica. Se concluirá explicando la reproducción electrónica del sonido y más concretamente de los sonidos de percusión.

### 1.2.1 Un principio; el sonido

El sonido, según la Acústica, son un conjunto de vibraciones que producen sensaciones a través del órgano del oído y a las que se pueden atribuir cualidades fisiológicas. [Recoo].

El sonido, según la Física, se genera por el movimiento vibratorio de un cuerpo y se propaga en forma de onda a través de medios elásticos. La onda sonora es mecánica, longitudinal y esférica [Sero8].

Podemos comprobar estos fenómenos pues al tocar un tambor con la palma de la mano que había sido percutido (en el apartado 1.2.4 se verá que tiene largo sostenimiento) las vibraciones y el sonido cesan. Así también en una sala de vacío no existe sonido.

Por otro lado en la música, como arte que es, califica el sonido en categorías audibles. No miden, por tanto, parámetros cuantitativos sino parámetros de percepción sonora. Una forma de clasificación es mediante las propiedades básicas: altura, duración, timbre e intensidad. La tabla 1 muestra estas propiedades, sus diferencias en la forma de onda y las palabras utilizadas para expresar el rango.

La altura indica el orden y distancia frente a una referencia entre los sonidos. En muchas ocasiones la función de referencia corresponde al diapasón y se usa para afinar, pues aporta una frecuencia constante de 440Hz denominado “la”. Se considera que son 7 notas (do, re, mi, fa, sol, la, si) ordenadas en diferentes escalas musicales y representadas en el pentagrama. Este parámetro se relaciona con la frecuencia de onda. [Peto1]

| Cualidad            | Característica       | Rango  |
|---------------------|----------------------|--|
| <b>Altura/ tono</b> | Frecuencia de onda   | Agudo o grave  |
| <b>Intensidad</b>   | Amplitud de onda     | Fuerte o débil   |
| <b>Timbre</b>       | Armónicos de la onda | Entran factores cualitativos e incluso personales (metálico, dulce, áspero, aterciopelado, etc.) |
| <b>Duración</b>     | Tiempo de vibración  | Largo o corto  |

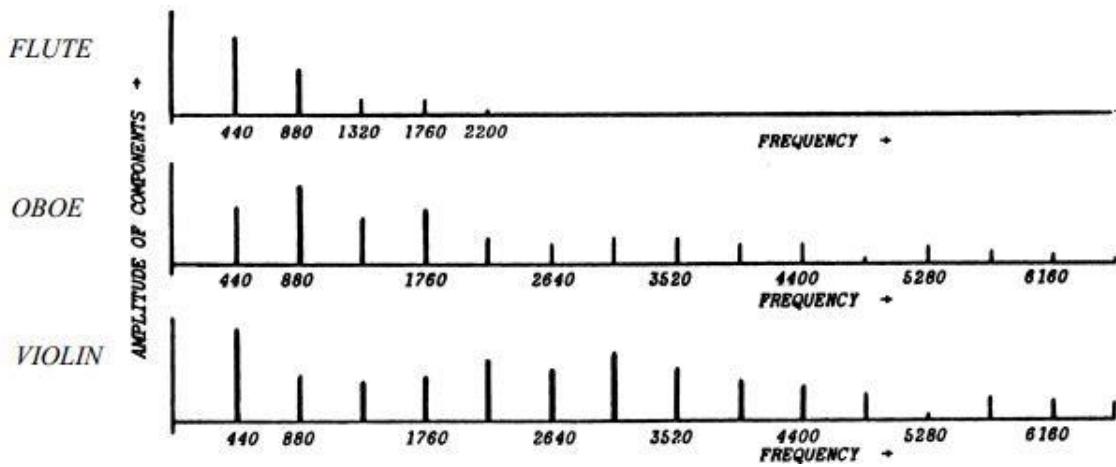
*Tabla 1: Cualidades del sonido*

La intensidad indica la fuerza con la que se produce el sonido. En la partitura se escribe p (piano en italiano) para poca intensidad y, fff (fortíssimo) para máxima intensidad.

La duración corresponde al tiempo de vibración. Se relaciona con las figuras (blanca, corchea...) y el tempo de la partitura. El tempo antiguamente se media en grados (lento, andante, presto) pero, actualmente también aparece en bpm (beats per minute) y para su medida se utiliza el metrónomo.

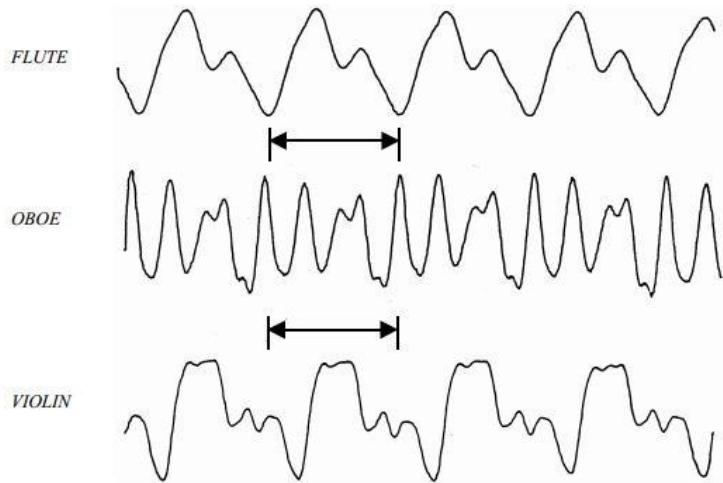
El timbre es la característica que permite diferenciar acústicamente elementos. Por ejemplo un violín frente a una flauta o, a un oboe. Aun cuando toquen las mismas notas con intensidades similares son perfectamente distinguibles por esta cualidad [Peto1]. Esta diferenciación es muy extrema pero la conclusión extraída es aplicable a tambores o instrumento de percusión que aunque golpeados con la misma fuerza son distintos en composición (madera, tamaño, parche...) y distintos, por tanto, en timbre. El timbre de esa nota musical depende de su frecuencia fundamental y de los armónicos que contiene. Además de sus amplitudes relativas. En la figura 1 se observa que en función de cuál sea la amplitud de estos armónicos y sus frecuencias (en qué lugar del espectro se encuentren), variará el timbre. A través de funciones de Fourier, cualquier sonido es reducible a una combinación de ondas senoidales de distinta frecuencia y amplitud. Este fundamento es utilizado por los sintetizadores musicales (apartado 1.2.4). [Zor98]

En la figura 2 aparecen las diferentes ondas del violín y la flauta lo que les aporta sonidos característicos para una misma nota (LA=440Hz). Así también diferenciaremos más tarde a los toms o tambores aunque sean golpeados con la misma fuerza



**Figura 1: Armónicos y fundamentales de diferentes instrumentos [Peto1]**

La batería se basa en la oscilación de diafragmas al ser golpeados. Vibrarán a una frecuencia fundamental y por su geometría a unos armónicos. [Cuco3]. La onda que emite un instrumento de percusión al ser golpeado en un punto y con la misma intensidad será siempre la misma. Las únicas diferencias serán la forma del golpe (material con el que se golpee, punto de aplicación) y la intensidad que se le aplique.



**Figura 2: Diferencia de Timbres en ondas acústicas [Peto1]**

### 1.2.2 La batería acústica



**Figura 3: Batería acústica sonorizada para directo**

La batería es el instrumento de percusión por antonomasia y un emocionante mundo en el que siempre hay algo que aprender. La batería es ritmo, tempo y vitalidad. La parte más visceral de la música moderna.

¿Quién no ha movido los pies o incluso la cabeza ante un ritmo pegadizo? Hay veces que irremediablemente nuestro cuerpo lo exige.

Y aun así siendo la batería tan instintiva, ésta es capaz de acompañar a los instrumentos más tenues y melodiosos del panorama clásico. Ciento es que no encontramos una batería como tal en las formaciones de las orquestas sinfónicas aunque encontramos instrumentos de percusión tales como timbales, platillos, triángulo...

La batería es un instrumento de percusión y por tanto familiar a todo este elenco. En consecuencia se hablará de los distintos instrumentos de percusión al hablar de la batería y se entrelazarán términos de esta índole tales como Groove, tempo, ritmo...

Por otro lado la batería dispone de ciertas ventajas, desde la especial aura que rodea a este instrumento que atrae a todo el mundo, a ser uno de los instrumentos en los que más se ha avanzado tanto en construcción, como en proliferación de métodos de enseñanza. Por contra, cuenta con ciertas inconvenientes y desventajas frente a otros instrumentos en materia de transporte, ergonomía y montaje como se entiende en la figura 3. Esa figura corresponde a una batería Ddrum modelo Dios con platos Turkish, herrajes Gibraltar prestada por Madrid Musical para las jornadas Sphera de la EUIT de Telecomunicación.

Es el instrumento que más cuesta montar, preparar y sacar sonido. En la figura 3 cada una de las partes de percusión incorpora un micrófono lo que da lugar a un total de ocho más sus respectivos cables, la mesa de mezclas y el equipo de PA. Es el equipo básico para una audición en directo. A esto hay que añadir un monitor si lo desea el músico. Aparte es el instrumento que más se procesa y trata su sonido en un estudio de grabación.

#### **1.2.2.1      Sonido y componentes**

La primera vez que se observa a un músico sentado en la batería introduce en la persona una sensación de admiración y cierto grado de misterio. No se concibe la manera en la que tantos elementos distintos pueden funcionar y sobre todo de manera coordinada. Cada parte de la batería es un instrumento en sí y tiene sus características. En la figura 4 se examina un dibujo señalando dichas partes y que se declaran a continuación:



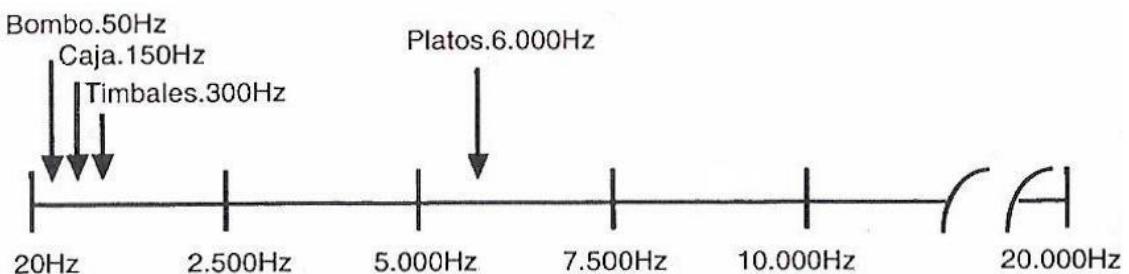
**Figura 4: Partes de la Batería Acústica[Bla05]**

- Caja (Snare en inglés): Parte principal. Posee unas láminas de metal en el parche resonante (parche inferior) denominada bordonera. Sonido agudo, corto y crepitante. [Bla05]
- Bombo (Kick): Elemento más grande. Sonido seco y grave. Se toca con un pedal que acciona una maza.

- Platos: Con un sonido brillante y largo sostenimiento. Fabricados con diferentes aleaciones (latón, B20...) son martilleados para sacarles el sonido. Distinguimos 3 modelos en función de su uso: Charles, Hi-hat en inglés, (tamaño 13" o 14") dos platos que se accionan mediante un pedal. Sonido distinto en cada posición (abierto, cerrado). Ride (tamaño 20" o 22") Sonido muy definido y utilizado para estribillos. Crash(tamaño 16" o 18") gran ataque, utilizado para cambiar cualquier cambio de ritmo o acentuación.
- Toms. Tambores genéricos. Pueden estar flotantes, sobre un rack o sobre patas. Sonido gordo y compacto. Utilizado para marcar ritmos ligeros denominados fills. Distintos tamaños (12",13" ó 16") y distintos grosorres para conseguir distintas afinaciones. A menor tamaño más agudo y seco.
- Soportes: Son las partes metálicas encargadas de la sujeción de tambores y platos. Construidos en su mayoría de acero cromado, minimizando el contacto con cada parte pues afectan a la calidad del sonido.

Respecto al sonido, tal y como se presentó en el apartado 1.2.1:

- Altura: Depende de la afinación de cada elemento. Aportando un sonido que ordenaríamos de grave a agudo según la frecuencia fundamental de la figura 5.



*Figura 5: Frecuencias fundamentales de la batería[Cuco3]*

- Intensidad: Es posible tocar a pequeño volumen pero al ser utilizado por estilos musicales con gran energía es tocado con gran volumen.
- Duración: Sonido breve en comparación con otros instrumentos. No es posible alargar la nota como si sucede en los instrumentos de viento.
- Timbre. Entre distintos bombos podemos encontrarnos infinidad de sonidos. La riqueza de armónicos marca la diferencia y el gusto en el sonido. Por ejemplo un plato Crash de principiante estaría alrededor de 50€ y uno de profesional, con las mismas medidas, 450€ (9 veces el valor).

En la batería, como aspecto caracterizador del sonido existen algunas condiciones [Cuc03]

- Articulación: Considerada como la forma y el lugar del impacto.
- Proyección: Relación entre el volumen y la distancia
- Interacción con el mundo real: Interna entre la baqueta, el parche, la madera, el borde del aro... Externa entre los elementos entre sí, la microfonía (aunque se intente aislar), la acústica del local.

Podríamos hablar de infinidad de temas que abordan el sonido de una batería acústica. En tambores podríamos hablar de su construcción, de las maderas utilizadas, de los distintos tamaños y grosores, así como de los parches. En platos de las distintas formas, aleaciones y martilleados. Y si nos saliéramos del tema de sonido: de herrajes, de baquetas, de hardcase , de moongel.... Digamos que es sólo una introducción.

#### **1.2.2.2      *¿Cómo se toca la batería?***

Este instrumento posee una curva de aprendizaje particular. Al principio la curva es muy rápida pues las primeras habilidades son muy sencillas de reproducir y es posible tocar un sencillo patrón de rock en unas semanas. Aunque a medida que pasa el tiempo y se indaga en el instrumento, mejorar o aprender algo nuevo se convierte en trabajo de muchos meses.

Cuando se procede a tocar, lo primero es lograr una buena posición adecuada que facilite tocar evitando malas posturas, vicios y sus correspondientes lesiones. Se procederá colocando el sillín a la altura de las rodillas y sentarse en él con la espalda recta. El apoyo con el sillín deberá ser el mínimo posible pues debe existir total libertad en el movimiento de las piernas.

Las piernas deberán apoyarse sobre los dos pedales y formar un ángulo de  $120^{\circ}$  con la vertical. Los brazos deberán colocarse lo más pegado al cuerpo manteniendo los codos en ángulo recto. [Stao5]

Respecto al agarre (grip) deberá realizarse siguiendo dos pautas:

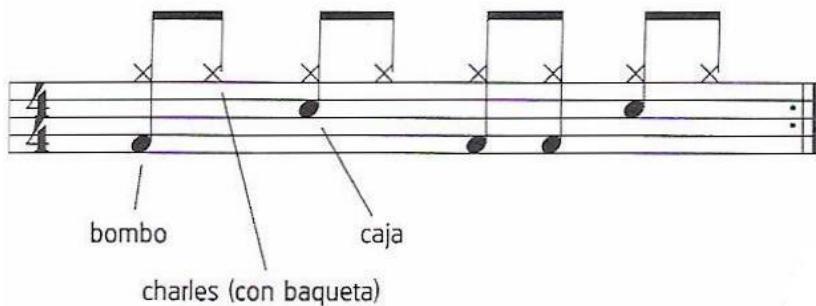
- Mínima fricción posible por lo que se cogerá con dos dedos (pulgar e índice) y se abrazará con el resto, tal y como se muestra en la figura 6.

- Punto de máximo rebote, llamado fulcrum. Es la zona de la baqueta con la cual con el mínimo esfuerzo se consigue la mayor eficacia y gestión del movimiento. Generalmente se encuentra a 2/3 del mango aunque depende de cada baqueta el punto exacto.



*Figura 6: Agarre de baquetas[Stao5]*

Una vez colocados en la batería de manera ergonómica y sabiendo coger las baquetas se dispondrá a tocar el ritmo más típico y utilizado en el pop/rock. Tarareado como tu-pa-tutu-pa, se escribe en el siguiente pentagrama de la figura 7.



*Figura 7: Patrón básico en 4 por 4 de pop/rock [Pino1]*

Es la forma en la que se escribe en la batería. Las notas en el pentagrama no indican su altura, sino el elemento al que se debe golpear. Por tanto a un ritmo lento se cuenta 4 e introducir cada sonido en su posición. Si resulta complicado comenzar manteniendo un ritmo continuo de charles e ir introduciendo el resto poco a poco.

El objetivo es mantener un tempo constante y una fuerza adecuada.

### 1.2.3 Las Baterías Electrónicas

Las baterías electrónicas (electronic drum o e-drum en inglés) son instrumentos de percusión en el que el sonido es generado electrónicamente; en lugar de, como ocurre en las baterías acústicas, la vibración que produce el parche y la carcasa de madera. Además son instrumentos musicales relativamente jóvenes pues nacieron en los años 80 y en sus primeros años de vida se caracterizaban por una baja gama dinámica (poca sensibilidad), y unos sonidos artificiales. Unos problemas que no le hicieron justicia entre los músicos.

Aunque en esta época y con la revolución tecnológica, es diferente. Aparecen modelos de alta gama (más de 3000€) capaces de modelar sonidos, lanzar samples de calidad profesional, con capacidades MIDI avanzadas (apartado 2.3), varias zonas de golpeo, capacidad de reproducir música y grabar... características que hacen dudar de la definición antes expuesta.

Por último, antes de adentrarse levemente en el funcionamiento y en los componentes que lo forman, es necesario destacar una cosa. La imitación. Desde que se inventó siempre se ha buscado la semejanza o, incluso la identidad con su prima hermana, la batería acústica, figura 8. Por eso muchas de las afirmaciones que se den deberán extrapolarse al mundo de la percusión.



*Figura 8: Similitudes entre batería acústica y electrónica [Dthom]*

### **1.2.3.1      Funcionamiento y componentes**

La batería electrónica transforma una fuerza externa que presiona sobre una superficie, en un instante de tiempo, a un impulso eléctrico que es captado por un dispositivo capaz de convertirlo en un sonido.

Dicha presión o golpe se produce en una zona donde hay colocado un sensor de fuerza que emite una señal que recoge un sistema; el módulo. Éste analiza la señal, la procesa y reproduce el sonido acústico que el usuario desee (mediante unos cascos o altavoces). Por tanto este tipo de baterías se basan en la interrelación entre dispositivos electrónicos para captar la interpretación del baterista.

Sobre componentes, a grandes rasgos podríamos decir que una batería electrónica está formada por 3 componentes: Pads, disparador MIDI y sintetizador

#### **1.2.3.1.1    Pads**

La zona de golpeo previamente dicha. En estos elementos se contienen los disparadores (triggers en inglés) formados a su vez por unos sensores, transductores piezoeléctricos, (apartado 2.2.4). Los pads existen de diversos tamaños, formas y texturas puesto que representan la superficie de contacto con el baterista y son muy personales (apartado 3.2).

Se diferencian atendiendo a su función dentro del kit (figura 9):

- Pads de plato
- Pads de tom/caja
- Pads de bombo: Similares a los pads de caja pero con un soporte para su colocación en el suelo. Suelen disponer de una única fuente de señal (mono) que es la maza al golpear la superficie.
- Pads de Hi-hat: Similares a los pads de plato. En alta gama son diferentes pues disponen de mecanismos que incorporan la gama de sonidos del charles tradicional (apertura, cierre, movimiento del pedal...).



*Figura 9: Pads de plato, tom, bombo y hi-hat [Drola]*

#### 1.2.3.1.2 Disparador MIDI:

El verdadero cerebro de la batería electrónica pues es él quien recibe la información, toma las decisiones y, envía un mensaje con un cierto formato. Es, por tanto, el elemento central al que todos los pads van unidos [apartado 3.1].

#### 1.2.3.1.3 Sintetizador

El sintetizador toma los mensajes digitales y los convierte en un sonido a través de unos altavoces. Sobre como realiza esta acción se comentará en el capítulo 1.4 porque existen distintos modos. Por ejemplo una tarjeta de sonido, como la de un ordenador, podría realizar esta función.

Atendiendo a esta clasificación los módulos comerciales como el módulo TD-30 de la figura 10 son la suma de un disparador MIDI y un sintetizador. Recogen las señales de los distintos pads y reproducen el sonido mediante auriculares. Además suelen integrar otras funciones como ecualización o configuración de parámetros personales.



*Figura 10: Módulo de la batería modelo TD-30 [Drola]*

### **1.2.3.2      Ventajas**

Como se deduce de esta lectura las diferencias entre las baterías acústicas y las electrónicas son palpables. En este apartado se listarán las ventajas que aporta el concepto electrónico: [Dbata]

- Control de volumen: El sonido máximo es regulable en las e-drum lo que posibilita tocar con playalongs o con reproductores. En la batería acústica solo se puede bajar el volumen tocando con más suavidad.
- Gran rango de instrumentos a elegir: En función del módulo. Esto sobre todo dispara las posibilidades de comprensión de sonidos. Primero por la posibilidad de cambiar de instrumento según las necesidades y sus respectivas combinaciones que deben coordinarse.
- Grabación: Al ser un sistema digital permite almacenar composiciones sin necesidad de tener que colocar micrófonos o buscar un sitio con condiciones acústicas suficientes (figura 3). Además simplifica el procesamiento de señales y la facilidad de compartir archivos.
- Afinación: No es necesario afinar, el sonido es de calidad. No es necesario invertir tiempo en ello.
- Costes de mantenimiento menor (baquetas, parches...). Las baquetas de punta de madera suelen astillarse contra los platos de metal y en el momento en que esto sucede se debe cambiar puesto que las baquetas astilladas rompen parches con más facilidad. Las baquetas utilizadas en electrónica son de punta de nylon y los pads de goma lo que aumenta su durabilidad. Cada par de baquetas tienen un coste aproximado de 10€ y un parche de tom alrededor de 20€
- Insonorización: El ruido estará minimizado por lo que es posible convivir con vecinos y familiares sin convertir el estudio en su suplicio.

### **1.2.3.3      Limitaciones**

Muchos baterías consideran a las baterías electrónicas como un instrumento diferente por ser utilizado para otras tareas distintas como son la composición o practicar con comodidad. En este apartado se van desglosar sus limitaciones. [Dbata]

#### **1.2.3.3.1    Distribución del set:**

La electrónica en general tiene toms y cajas más pequeños y, el bombo, no es el centro principal de la distribución. Por lo que si el usuario de batería electrónica se sienta delante

de una acústica tendrá una configuración muy diferente. Por esta razón las baterías de alta gama son más grandes o bien, los músicos prefieren colocar triggers específicos (ddrum redshot) y un módulo, sobre una acústica convirtiéndola en una electroacústica.

El charles de las electrónicas. Lo veremos en el apartado 3.7. El pad PCY65 de Yamaha más el controlador HH65 es una imitación con poca sensibilidad y articulaciones. Además se permite situar el plato y el pedal donde el usuario se sienta más cómodo lo cual es una mala práctica. En una batería acústica todo se coloca en un mismo herraje con pedal.

#### 1.2.3.3.2 El sonido

Los sonidos de cada una de las partes aplicadas al músico no son tan amplios como el rango que se puede obtener de una acústica. Se generará el sonido según la característica físicas del tambor, la intensidad del golpe y el punto exacto o zona donde se ejecuta. También habrá que contar con la incorporación de bordonera y si se juega con su tensión, número de bordones, etc.

Además todas estas variables consumen una gran cantidad de recursos de memoria y de procesamiento por lo que suelen omitirse. La actual solución, aparte de módulos de gama alta con tablas de memoria amplísimas y sensibilidad posicional es conectar la electrónica a un sampler de PC para usar un programa como BFD2 o Hydrogen.

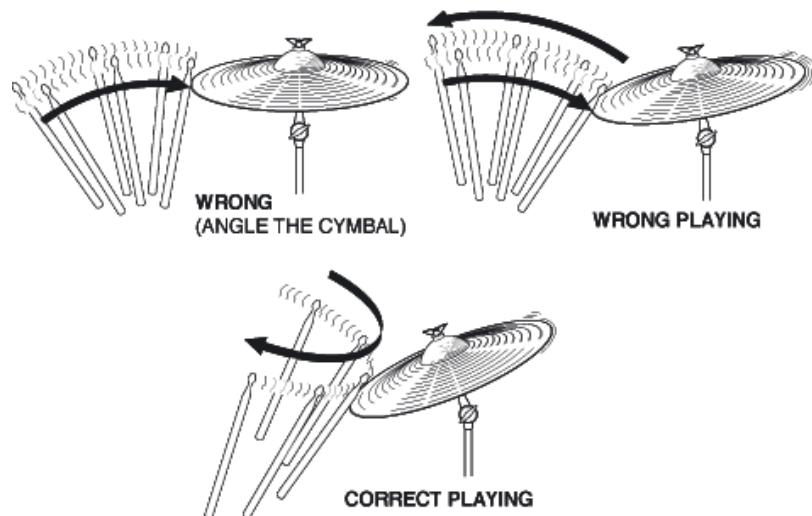
Tampoco hay una respuesta en todos esos toques experimentales que proporciona un conocimiento extensivo de la percusión. Además pensando en un batería innovador como Benny Greb que somete a la batería acústica como un terreno amplio de conocimiento e investigación. Este prolífico músico logra sonidos y patrones rozando platos, tocando con las manos o, golpeando un parche que estaba siendo presionado por la otra baqueta.

Por último un baterista que usa acústica debe experimentar con la afinación de los tipos de parches, conocer los materiales que componen su equipo, colocarlos adecuadamente... Todo ello para buscar su sonido, sintiendo la relación entre lo que toca y lo que se está oyendo, etc. En una eléctrica no hay que afinar nada y los sonidos son los que están en la memoria del sistema. Por tanto, bajo la opinión del autor, pierde una parte gratificante del mundo de la batería.

#### 1.2.3.3.3 Forma de golpeo (técnica):

Los platos de una eléctrica producen un sonido aceptable aunque se golpee con las manos. En una acústica se les saca el mejor sonido golpeando con la cabeza y parte del

cuerpo de la baqueta en el ángulo observado en la figura 11. Además se produce un movimiento oscilatorio natural en su cuerpo al golpearlos. Hecho que no se produce en el plástico de una e-drum.



*Figura 11: Forma óptima de golpear los platos[Dbata]*

La técnica de manos (agarre de baquetas) es esencial como se recalcó en el apartado 1.2.2.2 para evitar problemas de salud. Además en una acústica, hay que golpear correctamente para liberar el sonido y que suene bien. En la eléctrica, al igual que con los platos, se produce un sonido agradable. Puesto que sintetiza un mismo sonido.

En cuanto a los matices, es muy común que en la electrónica se modifiquen los parámetros por comodidad. Pero esto provoca vicios ya que el cuerpo utilizará descompasadamente más fuerza en los miembros superiores que en los inferiores.

#### **1.2.3.4      Estudio de mercado**

En las especificaciones de las baterías electrónicas no se mencionan elementos acerca de la detección del impacto (disparador MIDI) sino sobre la cantidad de entradas (número de pads y zonas de impacto), la síntesis (cantidad y calidad de sonidos) y el software (parámetros, configuraciones y entretenimiento).

En la tabla 2 se vislumbra esta conclusión sobre una serie de baterías electrónicas comerciales. Se clasifican según el rango de precios y las características comentadas que identifican gama baja, media y alta. Esta clasificación en gamas se estructura en el siguiente subapartado.

En la tabla 2 se integran y comparan las características que ofrecen (módulos, conectores...). Para realizar el trabajo se buscaron descripciones, fotografías y se visitaron tiendas.

Explicando las columnas de la tabla 2. El módulo identifica las especificaciones, escuetas del fabricante, sobre el aparato, los sonidos que es capaz de reproducir, la organización sistemática de ellos (drumkits) y patrones de secuenciación.

Los conectores se obtienen con un vistazo a su carcasa, al igual que los controles. Permite examinar el grado de complejidad de este tipo de sistemas.

Las columnas de pads (número de zonas, material, forma de ellos) y experiencia, se identifican con las necesidades particulares. Es distinto un bombo formado por un interruptor/switch que uno formado por un pad completo de malla con sensibilidad posicional.

Además se fija el rango de precios, a mayo de 2014, recogida en una distribuidora. [Dthom]

En el estudio se presentan baterías electrónicas de marcas como Roland y Yamaha, que son los líderes en el mercado y también conocidas en otros ámbitos. Otra de las marcas que se comenta es la sueca 2Box que ofrece la reproducción de muestras de alta calidad elegidas por el usuario además de un diseño en los pads muy atractiva. Alesis es otra de las marcas importantes que empuja la gama media.

Se incorporan los enlaces a las páginas de los fabricantes sobre los nombres de los modelos. Con ello se pretende contrastar la fuente de información primaria.

Medelli es una OEM (Original Equipment Manufacturer) que fabrica productos que luego distribuye a otra empresa que vende al por menor bajo su marca. Por tanto Medelli fabrica multitud de baterías electrónicas y coloca los logotipos de muchas marcas como DDrum, Thomman o incluso Alesis ya que se ha especializado. [Dvdru]

#### 1.2.3.4.1 Gama baja

- Enfocada al aficionado sin experiencia.
- Baja respuesta sonora. Pocos sonidos y de baja calidad.
- No suele detectar otros modelos de pads a los que vienen originariamente e incluso limita su incorporación. Por ejemplo el HD-3 de Roland con su conector DB25.

- Los pedales suelen estar formados por conmutadores que son más económicos y ocupan menos espacio.
- Dimensiones reducidas, para los pisos pequeños.

#### 1.2.3.4.2 Gama media

- Aumenta la calidad de los sonidos reproducidos. Posibilidad de introducirlos externamente por memoria USB o tarjeta SD.
- Mayor cantidad de pads aumentando zonas de golpeo y su superficie. Aparece el pad de bombo como elemento de la batería que debe golpearse, como en la acústica, con un pedal que sostiene una maza. Conteniendo un solo disparador.
- La sensibilidad de los pads es más cuidada. Se mejora el tacto con la baqueta. Las marcas sacan sus respectivas patentes Roland (malla), Yamaha (silicona) o Alesis (parche de Mylar).
- Salidas de audio de dos canales estéreo.

#### 1.2.3.4.3 Gama alta

- Un precio prohibitivo para la mayoría de las personas. Enfocado a aquel músico profesional que lo utiliza tanto en estudio como directo.
- Sonidos de altísima calidad y posibilidad de expansión.
- Centenares de efectos, secuenciadores y parámetros de control que mejoran la experiencia del usuario.
- Pads similares a acústica en tamaño, estabilidad, disposición.
- Salidas de audio profesional de 6 canales para conseguir sonido 5.1 (sonido envolvente).
- Charles integrado en herraje y con las articulaciones especiales totalmente mejoradas.

|                                       | Módulo                                      | Conectores  | Controles   | Pads   | Apariencia  | Precio | Fotografía  |
|---------------------------------------|---|---|---|--|---|--------|---|
| <b>Gama Baja</b>                      |   |   |   |  |   |        |   |
| <a href="#"><u>Roland HD-3</u></a>    | 20 drum kits<br>20 Patrones                 | MIDI Out<br>MiniJack<br>In/Out/Phones<br>DB25 con triggers  | On/off<br>14 botones<br>Knob Volumen<br>Pantalla LCD de 2 líneas                        | 4 Pads(Caja Estéreo)<br>3 Platos (Ride Estéreo)<br>HH switch 3 zonas<br>Bombo switch | Pedales integrados en rack<br>Parche de malla caja.<br>No permite intercambiar<br>Permite posición a zurdos | 699    |    |
| <a href="#"><u>Medelli DD502</u></a>  | 307 sonidos<br>70 drum kits<br>50 Patrones  | 9V, USB<br>MIDI IN/OUT<br>Jack Out LR/In/phones<br>Jack 8 pads  | On/off<br>22 botones<br>Knob Volumen<br>LCD 2 líneas<br>Leds pulsación                  | 4 pads estéreos<br>3 platos mono con Choke<br>HH 2 zonas<br>Bombo piezo              | Pads de goma<br>Pedal de bombo  | 366    |    |
| <a href="#"><u>Yamaha DTX400K</u></a> | 169 sonidos<br>10 drum kits<br>10 canciones | 12V, USB<br>Mini Jack IN<br>Jack Phones/Out<br>Jack 7 Pads  | On/Off<br>20 botones<br>Leds de estado  | 4 toms mono<br>3Platos<br>HH 2 zonas switch<br>Bombo switch                          | Mal rebote<br>Pads pequeños<br>Rack de acero<br>Modo practicar  | 409    |    |
| <b>Gama Media</b>                     |   |   |   |  |   |        |   |
| <a href="#"><u>Roland TD-11KV</u></a> | 190 sonidos<br>50 drum kits<br>20 Patrones  | 9V, USB, USB host<br>DB25 para trigger<br>Extra trigger TRS<br>MIDI Out<br>MiniJack In/<br>Jack Out LR/Phones | 14 botones<br>1 Knob Volumen<br>1 Encoder<br>Pantalla GLCD                              | 4 Pads estéreo<br>3 Platos estéreo (Ride 3 zonas)<br>HH 3 zonas                      | Pedales en el rack<br>Parche de malla en toms.<br>No permite intercambiar sonidos                           | 1399   |   |
| <a href="#"><u>Medelli DD508</u></a>  | 665 sonidos<br>99 drum kits<br>240 Patrones | 9V, USB<br>MIDI IN/OUT<br>Jack Out LR/In/phones<br>Jack 10 pads<br>SD   | On/off<br>30 botones<br>3 Knob Volumen<br>1 Encoder<br>Pantalla LCD 2 líneas y Leds con | 4 pads estéreos<br>4 platos estéreo con Choke (ride 3 zonas)<br>HH 3 zonas           | Pads de goma<br>Pedal de bombo<br>Secuenciador<br>Metrónomo   | 555    |  |

|                                   |  |  |   |   |   |      |   |
|-----------------------------------|--|--|---|---|---|------|---|
|                                   |  |  | pulsación   |   |   |      |   |
| <a href="#">Yamaha DTX542K</a>    | 449 sonidos<br>70 drum kits<br>60 canciones  | 12V, USB<br>Jack Phones/Out LR<br>Jack 8 Pads                                  | On/Off<br>Encoder<br>10 botones<br>LCD 2 líneas<br>3X 7 segmentos                           | 4 toms estéreo<br>3 Platos de 3 zonas y función choke<br>HH 3 zonas     | Muy buen rebote<br>Pads grandes   | 1329 |    |
| <a href="#">Alesis DM10X kit</a>  | 1,047 sonidos con muestras de 16-bit sin compresión<br>100 drum kits<br>100 patrones | AC, USB, MIDI IN/OUT<br>TRS 12 trigger inputs<br>Jack 4 OUT + Phones +2 IN     | 26 botones<br>2 Knob Volumen<br>1 Encoders<br>6 Slicer<br>5 leds<br>Pantalla GLCD           | 5 toms<br>4 platos estéreos (ride 3 zonas)<br>Pedal de control continuo | Superficies muy grandes<br>Pie de caja<br>Secuenciador  | 985  |    |
| <b>Gama Alta</b>                  |  |  |   |   |   |      |   |
| <a href="#">2 Box DrumIt-Five</a> | Sonidos (24 bits) almacenados en memoria SD 4GB<br>100 Kits                          | 18Vdc, USB, MIDI IN/OUT<br>Jack 10 pads<br>Jack 6 OUT + Phones + IN            | 9 botones<br>Knob Volumen<br>3 Encoders<br>Pantalla LCD de 4 líneas                         | 4 Toms estéreos<br>1 Bombo<br>3 Platos<br>HH multi-zona                 | Pads de malla con sistema flotante<br>Herraje separado de la caja y del HH                            | 2275 |    |
| <a href="#">Roland TD-30</a>      | 1100 sonidos<br>Multiefectos<br>Secuenciador   | AC, USB, USB memory<br>MIDI IN/OUT/TRU<br>Jack OutLR/In/phones<br>Jack 15 pads | On/off<br>22 botones<br>Knob Volumen<br>GLCD 80 x 256                                       | 4 pads estéreos<br>3 platos de 3 zonas con Choke<br>HH multi-zona       | Parches de malla con alta sensibilidad<br>Modelado de sonidos<br>Herraje separado de la caja y del HH | 3799 |   |
| <a href="#">Yamaha DTX900K</a>    | 1326 sonidos<br>100 drum kits<br>Multiefectos<br>Secuenciador                        | 16V, USB<br>Mini Jack IN<br>Jack Phones/Out<br>Jack 15 Pads                    | On/off<br>Sliders<br>22 botones<br>Knob Volumen<br>GLCD 240 x 64<br>7-segment LED (3-digit) | 4 toms estéreo<br>4 Platos de 3 zonas<br>HH multi-zona                  | Pads silicona<br>Herraje separado de la caja y del HH   | 4188 |  |

*Tabla 2: Comparación de los modelos de batería eléctrica más representativos*

#### **1.2.4 Sintetizador**

Recapitulando del apartado 1.2.1, para reproducir el sonido de un instrumento, tan sólo tendríamos que usar un generador de sonido, conectado a un altavoz, que reprodujera las frecuencias y los armónicos del instrumento. Así también se controlaría la amplitud y el amortiguamiento o tiempo que tarda en dejar de vibrar. Y es esto lo que hacen los sintetizadores, tienen almacenados algunos patrones de la respuesta de los instrumentos que pueden reproducir y generan las formas de onda según qué instrumento sea, qué valor de velocidad (que representa el golpe aplicado) y otros parámetros que se configuran en el mismo sintetizador.

Un sintetizador, por definición, es un instrumento musical electrónico; es capaz de generar sonidos artificialmente. Su nombre proviene por el proceso de síntesis que sufre la onda. [Zor98]

Así mientras los instrumentos tradicionales producen el sonido mediante sus características físicas (formas, dimensiones, fabricación...) el sintetizador los produce electrónicamente.

Esto proporciona una increíble versatilidad y flexibilidad puesto que se transforman los sonidos con cada nueva programación. Y no solo esto sino que al principio, desde la perspectiva musical, se crearon sonidos nuevos y artificiales. Ahora en nuestros días se tiende más a buscar la perfección en los sonidos reales.

##### **1.2.4.1 Particularidades**

Existen diferencias entre los sintetizadores pudiendo distinguir entre:

- Reproducen una nota en cada momento (monofónicos) o son capaces de reproducir varios sonidos a la vez (polifónicos). A día de hoy, lo normal, es que tengan más de 128 voces de polifonía.
- Reproducen los sonidos de un único instrumento (monotímbrico) o de un abanico de posibilidades (multitimbrico).
- Según los componentes internos; podemos diferenciar entre analógicos y digitales. En los analógicos son componentes discretos con imperfecciones. Su componente principal es un VCO (Voltage Controlled Oscillator). En cambio los digitales son más precisos y baratos lo que produce frialdad y delgadez. Su componente principal pasa a denominarse DCO (Digitally Controlled Oscillators) y son imperantes en la actualidad. [Pero3]

#### 1.2.4.2 Modalidades de síntesis

- Substractiva: el más utilizado. Se basa en eliminar armónicos del generador. Utilizando distintas formas de onda (senoidal, triangular, ruido). En la figura 12 podemos ver el diagrama de bloques de este proceso. El LFO (Low Frequency Oscillator) influye en la frecuencia del oscilador principal, en el filtraje de las componentes en frecuencia y de su respectiva amplificación. [Nabo1]

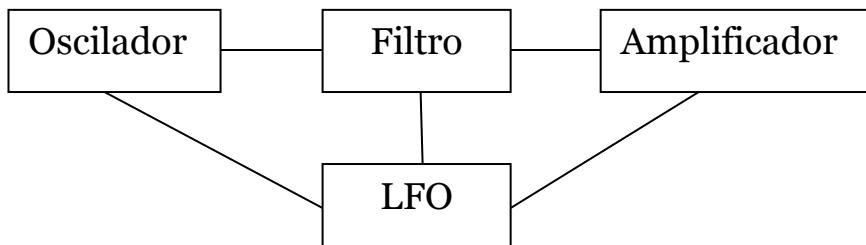
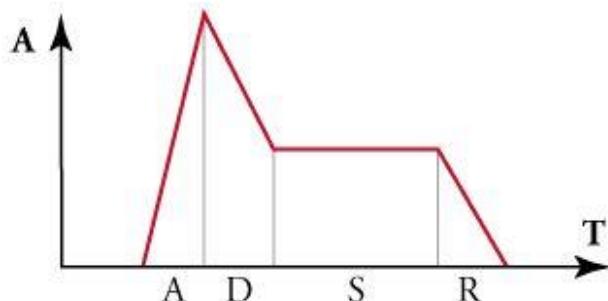


Figura 12: Diagrama bloques proceso de síntesis

- Aditiva: Sumando armónicos de distinta amplitud.
- FM. LFO es la modulada y el oscilador es la portadora. Escasamente utilizada.
- Tabla de ondas: Es la más utilizada en los sistemas PC. Graba una muestra musical y la reproduce a distintas velocidades produciendo distintas frecuencias. Consumo gran cantidad de memoria y potencia de cálculo aunque es más precisa. Hay veces que resulta indispensable grabarlo a distintas velocidades.
- Modelado físico: Representación matemática del instrumento.

Otro concepto que se debe conocer, es la manipulación de la envolvente ADSR, evolución temporal de cualquier sonido y por tanto del componente dinámico de los instrumentos. Consta de las fases de Ataque, Decaimiento, Sostenimiento y Relajación (figura 13).

En un tambor, la dinámica comprendería solo la porción de ataque y de relajación (muy rápidos los dos).



*Figura 13: Grafica ADSR [Zor98]*

#### 1.2.4.3 Software de batería basado en tabla de ondas

Se ha hablado de cómo se reproducen electrónicamente las ondas, de los distintos sintetizadores pero... ¿es necesario comprarse un dispositivo específico, un sintetizador?

La respuesta es no; pues al vivir en la era informática existe la alternativa del PC pero esta decisión traerá complicaciones (ver apartado 3.5.2)

La alternativa es un software específico para instrumentos musicales. Se tratan de unos programas que integran unas muestras de batería grabadas y, reproducen la onda acústica mediante síntesis por tabla de ondas.

Son grabados utilizando instrumentos musicales de alta gama en estudios de grabación míticos y con profesionales de alto nivel (ingenieros y músicos). El resultado es un producto de gran valor añadido.

Incluso son utilizados por los productores de música para sustituir al artista de sesión por distintas razones, como por ejemplo, abaratar costes. Se puede citar que en el disco “Justicia” del grupo Habeas Corpus aparece en la parte posterior de la carátula “Batería programada por Chifly (guitarra)” y ser difícil apreciar la diferencia.

Algunos de ellos se denominan VST (Virtual Studio Technology) pues mediante esta interfaz propietaria se introduce un complemento sobre un software particular dedicado al audio (DAW o Digital Audio Workstation). La tecnología VST simula por software un instrumento, tienen la capacidad de procesar y generar audio, como también interactuar con interfaces MIDI. [Dstei]

Aquí se propone una lista de software que se han encontrado para trabajar. Las características que se contemplan son:

- Presets/ajustes Adopta configuraciones personalizadas para facilitar el uso. Objetivo sentarse y tocar.
- Numero de Kits y de articulaciones: Diferentes formas de tocar el dispositivo. Por ejemplo el Hi-Hat puede tener hasta 11 articulaciones (Borde, campana, pedal...)
- Velocity layers: Pues cada articulación ha sido grabada a distintas velocidades para responder a distintas intensidades.
- Grabación en multi canal: Por lo que es posible controlar más parámetros (delay, mute, pan, overdrive...).
- Patrones MIDI: Ficheros en formato MIDI con patrones rítmicos para utilizar en composición de canciones y realizar loops.
- Precio.
- Una imagen que defina la interfaz gráfica y también la de control.

Por último todos los productos comerciales ofrecen versiones de prueba o limitadas que han permitido, en la mayoría, compararlos y probarlos.

#### 1.2.4.3.1 Addictive Drums- 179,95€



*Figura 14: Panel de Control Addictive Drums [Dxlna]*

Addictive Drums es un software creado por XLNAudio. Sus características incluyen:  
[Dxlna]

- 3 kits de batería para obtener diferentes sonidos. Marcas de baterías de gama alta como DW, Sonor y Tama.
- Efectos Múltiples para los tambores.
- Más de 100 presets y 3.000 patrones MIDI.
- Muestras grabadas a 96khz y con una resolución de 24 bits. Procesadas sin compresión y reproducidas con bit variable y a 44'1khz

#### 1.2.4.3.2 BFD2- 199€



*Figura 15: Panel de control BFD2 [Dfxpa]*

BFD2 es creado por FXpansión. Explican en su sitio web que han utilizado diversos micrófonos en distintas posiciones [Dfxpa]. Las siguientes características son técnicas:

- 10 Kits con 96 velocity layers entre ellos las baterías de Ringo Starr (Ludwig Oyster Black Pearl) y John Bonham (Ludwig Vistalite).
- 5000 patrones MIDI.
- Efectos de humanización más realista. Evita lanzar la misma muestra en varias ocasiones. Incorpora retardos, filtrado, amplificación, distorsión.
- Mapas MIDI de los principales módulos de batería.
- Sonidos grabados a 44'1khz y 24 bits. Sin compresión ni procesamiento. Permite importar sonidos propios y utilizarlos

#### 1.2.4.3.3 EZdrummer- 119€



*Figura 16: Panel de Control Ezdrummer [Dtoon]*

EZdrummer desarrollado por Toontrack . Se trata de una versión simplificada de su predecesor, Superior Drummer. [Dtoon]

Toontrack ha lanzado doce paquetes de expansión para Ezdrummer. Los más destacados son: Drumkit from Hell (grabado por Tomas Haake de Messugah) y Jazz (grabado por Roy Wooten). [Dtoon]

Sus características incluyen:

- 7,000 muestras de sonidos a 16-bit / 44.1 kHz,
- 8,000 patrones MIDI

#### 1.2.4.3.4 Drummic'a

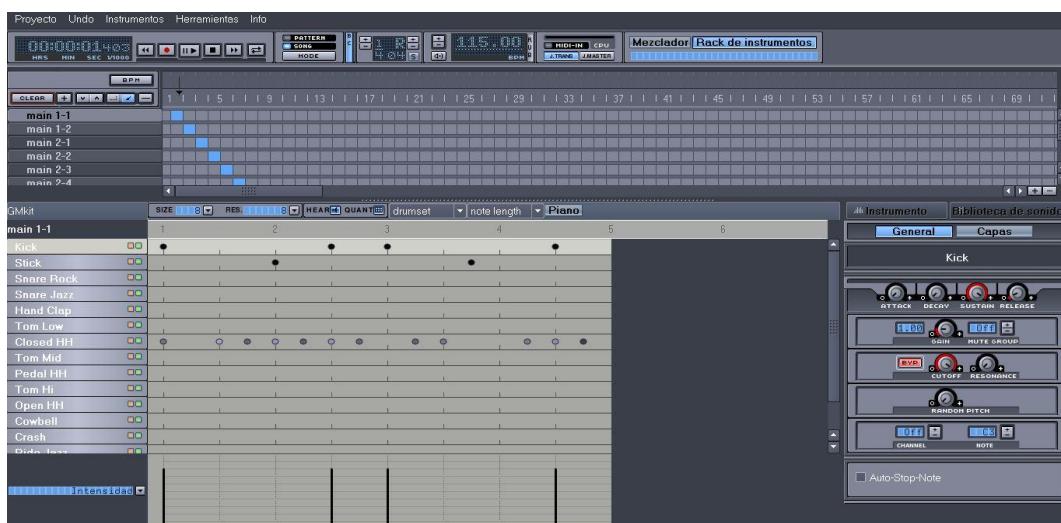


**Figura 17: Panel de Control Drummic'a [Dsenn]**

Es una alternativa de software de baterías sin coste. El programa es proporcionado por el fabricante de micrófonos Sennheiser para que se evalúen sus productos. Grabado con 15 micros Sennheiser y Neuman en 9 partes de la batería.[Dsenn]

- La batería utilizada es de la marca Wahan un set acrílico y platos Sabian.
- 13000 muestras de sonido. Sonido sin procesar ni comprimir con una resolución lineal de 24 bits. Con un total de 25 velocity layers
- Mezclador, Ecualizador 4 bandas y selección de micrófonos.

#### 1.2.4.3.5 Hydrogen



*Figura 18: Panel de Control Hydrogen [Dhydr]*

Hydrogen es un programa de emulación de patrones avanzados para Linux. Su principal objetivo es la secuenciación de ritmos de una manera sencilla, intuitiva pero sobre todo profesional. [Dhydr]

Es distinto a los comentados anteriormente por que no es exclusivo de batería sino que su función principal es secuenciar. Pero, al igual que dichos programas, se utilizará para la reproducción de sonidos en directo que se ejecutara través del prototipo MIDI al efectuar las percusiones correspondientes.

La última versión actualizada es la 0.9.5 que se puede descargar en su página web existiendo instaladores para todos los sistemas operativos.

Especificaciones: [Dhydr]

- Importa sonidos en formato .wav .aiff y .flac.
- Sonidos multicapa.
- Posee un amplísimo banco de sonidos, muchos de ellos configurados en kits de batería de amplísima calidad y por supuesto de libre distribución. Los instrumentos soportan multicapa (hasta 16 muestras para cada instrumento) pero lo destacable es la posibilidad de ver las muestras e incluso editarlas (figura 19).
- Utiliza un mapa MIDI fijo. Sin embargo, se puede cambiar el mapeo con sólo arrastrar el instrumento del kit de batería hacia una posición en la lista. La posición del instrumento en la lista determina la nota MIDI que este sonido se asigna a: Posición 1 = nota 36, pos2 = 37



*Figura 19: Forma de onda de sonido de bombo*



## 2 ANÁLISIS TEÓRICO

---



## 2.1 Protocolo comunicaciones

La pretensión es llegar a comprender para luego diseñar un objeto similar a la batería electrónica, un instrumento musical digital. Este apartado versara sobre el protocolo exclusivo de estos sistemas.

### 2.1.1 Introducción a MIDI

MIDI (Musical Instrument Digital Interface) es una norma de la industria de la música que establece la forma en la que los instrumentos musicales de corte electrónico deben comunicarse. MIDI es, por tanto, un protocolo capaz de permitir que un instrumento musical pueda comunicarse con otro.

En la norma se especifica tanto, la interfaz física a cumplir por los equipos, como los datos intercambiables entre ambos, como el formato de almacenamiento de los ficheros. [Aso95]

Porque cuando el músico presiona una tecla, un botón, una palanca o activa un sensor se genera cierta información que se denomina mensaje.

Un mensaje consta de un comando (una orden) y de unos datos asociados (que pueden ser uno, varios o ninguno, en función de la acción realizada). Son secuencias de bits enviados de forma serie y asíncrona. En ellos se recoge multitud de información como la nota pulsada, cambio de instrumento, volumen de la ejecución, etc. Indica qué sonidos deben tocarse (~melodía) y en qué momento del tiempo (ritmo) o incluso informa con qué intensidad (volumen) deben ejecutarse.

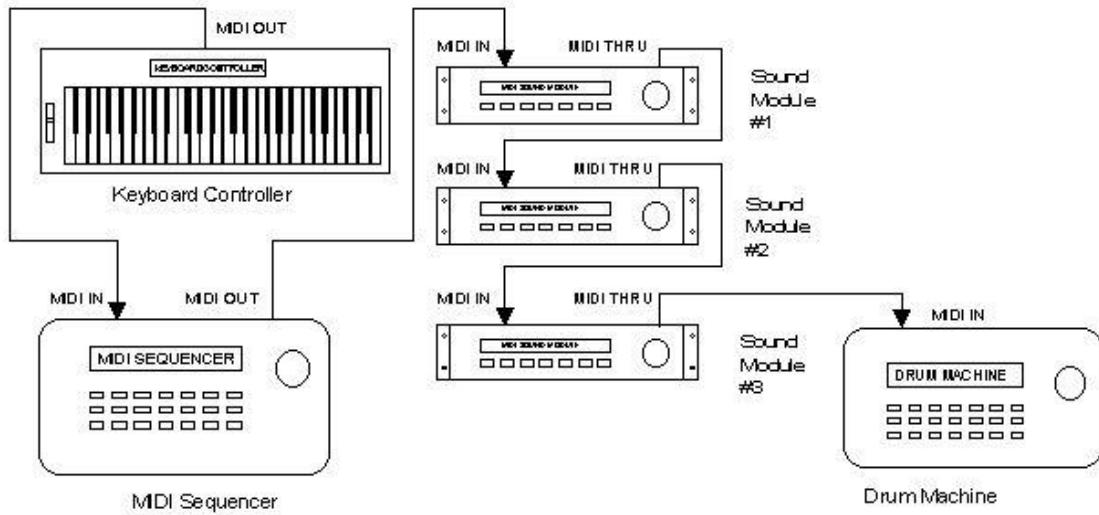
En este proyecto particular como existen diferentes acciones que se pueden realizar sobre una batería MIDI, existirán diferentes tipos de mensaje para describir dichas acciones.

#### 2.1.1.1 ¿Por qué nació MIDI? Ventajas y breve historia

En la década de los 80 se produce una auténtica revolución musical. Es un periodo de experimentación, de cambio y de ruptura con lo anterior. Además se popularizaron instrumentos como los sintetizadores digitales que, como novedad, cautivaron tanto a los grupos como Pink Floyd o Depeche Mode como a muchos aficionados.

Atraídos por un mercado emergente y una producción más económica aparecieron muchos fabricantes de instrumentos electrónicos. Como desventaja decir que cada uno diseñaba su propio interfaz para comunicar entre sí las máquinas que fabricaba.

Con el nacimiento de MIDI en 1982 (con la construcción del Prophet 600) se cambia esta mentalidad por completo siendo un protocolo capaz de intercambiar información entre diferentes tipos y marcas de dispositivos musicales, tal y como se entiende en la figura 20.



*Figura 20: ¿Cómo conectarías todos estos aparatos sin el protocolo MIDI? [Dmidi]*

MIDI es un sistema abierto: muchos mensajes MIDI están todavía sin definir por lo que la norma se ha ido enriqueciendo, con nuevas características de los dispositivos musicales digitales. [Dedru] La última ha sido General MIDI (GM). Esta ampliación determina una clasificación por timbres en los dispositivos musicales que manufatura cada fabricante.

Además se ha generado un sinfín de software de edición MIDI siendo cada uno más intuitivo que el anterior. La esencia; ser utilizado por gente inexperta en el mundo de la informática y de las telecomunicaciones tales como músicos, artistas y aficionados. Algunos ejemplos a citar de este tipo de software son el Guitar Pro (especial para guitarristas) o editores de partituras como el Finale.

Por último el formato de archivo (.mid) en torno al software mencionado permite reproducir y componer música. Se caracteriza por la ligereza de los archivos ya que permite almacenar multitud de melodías complejas como las de música clásica, en la que intervienen multitud de instrumentos, en muy poca memoria.

### **2.1.2 Interfaz física**

MIDI se fundamenta en una interfaz serie que opera a velocidad de transmisión/recepción de 31.250 baudios (con una tolerancia del 1%). [Aso95]

Además es asíncrona, lo que significa que no hay ninguna línea adicional con una señal de reloj o algún medio similar de sincronización. Para llevar a cabo la función de sincronización se utiliza por tanto un bit de inicio (Start) y otro de parada (Stop) por cada 8 bits de datos enviados. La transmisión envía los bits de menos a más significativos (LSB primero). Recordando que un baudio es un bit que se transmite en un segundo por lo que en MIDI la trama de 10 bits se entregará en 312,5 microsegundos.

Una peculiaridad que diferencia de muchos otros sistemas serie es que se basa en la activación y desactivación de una corriente, en lugar de tensión, para representar los niveles lógicos. La corriente de activación es de 5 miliamperios y representa el cero lógico mientras que el uno lógico está representado por una corriente casi cero (menor de un 1 microamperio). Con ello se consigue mejorar las atenuaciones del cable. [Tomo3]

#### **2.1.2.1 Conexionado**

Generalmente, existen tres puertos en los instrumentos musicales MIDI nominados IN, OUT y TRU. Los puertos MIDI, figura 22, tienen conexiones DIN de cinco patillas (180 grados) de las cuales sólo se utilizan la 2, 4 y 5, quedando siempre la patillas 1 y 3 desconectadas. Los paneles de las cajas son los DIN hembras. [Aso95]

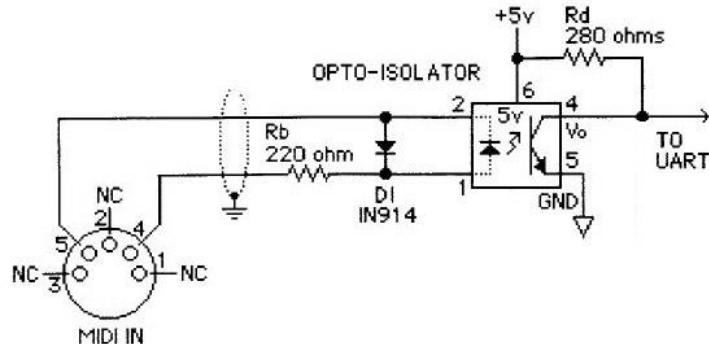
La norma específica que los cables deben ser de pares trenzados y apantallados y de no más de 15 metros de longitud. Además la malla debe de estar conectado al pin 2 del conector en ambos extremos que debe conectarse a la masa de todos los circuitos.



*Figura 21: Conectores MIDI*

Se ilustrará más en detalle el esquema eléctrico de cada puerto:

El conector MIDI IN recibe la información que proviene de un conector OUT de otro instrumento.

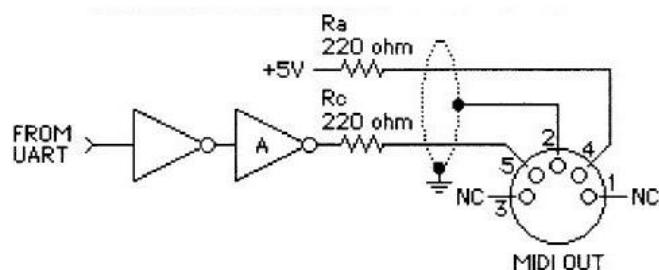


**Figura 22: Circuito entrada MIDI típico [Aso95]**

En el circuito de la figura 22 se observa que R<sub>b</sub> proporciona parte de la limitación de corriente para conseguir el bucle de corriente de 5 miliamperios. D<sub>1</sub> se incluye para proteger al optoaislador de inversiones de polaridad. R<sub>d</sub> es una resistencia de carga del fototransistor del optoaislador. Todas las entradas deben estar aisladas por medio de un optoaislador, el cual necesita ser de alta velocidad con tiempos de elevación y decaimiento de 2 nanosegundos o menos. Un ejemplo válido de optoaislador es el 6N138. [Tomo3]

El conector MIDI OUT envía la información generada en los controles del instrumento hacia el IN de otro instrumento.

Un circuito de salida completo (figura 23) consiste simplemente en un inversor y un transistor o IC de salida (A) con resistencias limitadoras de corriente. La utilización del inversor depende la polaridad con la que transmita la UART.



**Figura 23: Circuito de salida MIDI [Aso95]**

El conector TRU tiene como misión repetir, sin modificaciones, la información que llega al puerto IN.

A simple vista, parece que este puerto no tiene mucha utilidad, sin embargo es imprescindible para el conexionado en cadena, configuración en la que varios dispositivos están conectados entre sí. Cada equipo recibe y reenvía por este puerto toda la información del sistema, de forma que cada equipo sólo procesa la información que va en su canal pero retransmite toda para que los demás dispositivos puedan hacer lo mismo. Para ver su conexionado recomendamos dirigirse a la bibliografía.

Como se visualiza en los circuitos del conexionado (figura 23 y 24), buena parte de los aparatos MIDI son capaces de enviar y recibir información, pero desempeñan un papel diferente dependiendo de si están recibiendo o enviando información; también dependen de la configuración del programa o programas que pueden usar dicho dispositivo. Como regla general utilizan una configuración maestro- esclavo. Esta configuración proporciona que el lenguaje sea simple y muy bien gestionado.

#### **2.1.2.2 Aparatos**

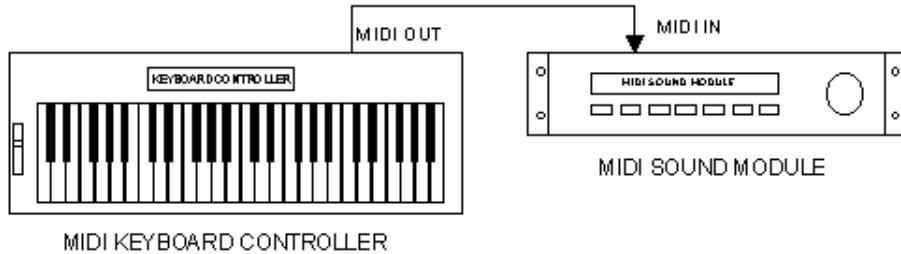
Como hemos visto con los puertos podríamos clasificar los aparatos MIDI en función del tratamiento que realizan con la información: [LEh93]

- **Controladores:** generan los mensajes MIDI. El controlador más habitual tiene aspecto de teclado de piano, al ser este instrumento el más utilizado a la hora de componer e interpretar; sin embargo, hoy día se han construido todo tipo de instrumentos con capacidad de transmisión MIDI: percusión, guitarras o incluso, flautas MIDI.
- **Unidades generadoras de sonido:** Sintetizadores o samplers. Reciben los mensajes MIDI y los transforman en señales sonoras.
- **Secuenciadores:** destinados a grabar, reproducir o editar mensajes MIDI. Pueden desarrollarse bien en formato de hardware, o incorporados en un sintetizador como software.

#### **2.1.2.3 Interconexión entre aparatos**

En la figura 24 el instrumento denominado como maestro envía el mensaje producido en los controles del instrumento al exterior a través del puerto OUT del aparato para ser dirigido a través de los cables hacia otros instrumentos externos MIDI.

Con este esquema estaríamos consiguiendo que los eventos producidos en el instrumento maestro al interpretar una canción también sonasen en el esclavo (debido a que los dos tienen generadores de sonido)



**Figura 24: Conexión entre diferentes dispositivos [Dmidi]**

Las notas serán reproducidas simultáneamente en los dos dispositivos, maestro y esclavo. Aunque el mensaje después de ser procesado por el maestro tiene que viajar a través de un cable y ser gestionado por el interfaz y los generadores de sonido del esclavo. Por consiguiente las notas no suenan simultáneamente, pero la diferencia de tiempo entre las dos emisiones es tan pequeña que el oído humano no la puede percibir y las reúne en una sola sensación. Este tiempo es denominado latencia, y es tratado en el capítulo 3.5.2.

Las restricciones que pone MIDI respecto a este tema: [Aso95]

- Para un coro (polifonía) de 10 notas se permite un retardo máximo de 10mSg entre la primera y la última de ellas.
- Desde la generación del evento (pulsación o suelta) hasta la transmisión al exterior se permite un retardo máximo de 7 milisegundos.

En el caso de que los instrumentos fueran más numerosos (16 como máximo) habría de conectarse en forma de cadena (Daisy chain en inglés) a través de sus puertos IN y TRU. El maestro enviaría un mensaje que entraría por el puerto MIDI IN del dispositivo que está conectado directamente y este primer esclavo lo repetiría por el puerto TRU al esclavo 2 que haría indistintamente esas mismas acciones hasta que el aparato al que va destinado se identifique realizando su tarea.

### **2.1.3 Formato de los mensajes MIDI**

Este apartado resumirá todo lo concerniente a la estructura de mensaje y la organización de los bytes.

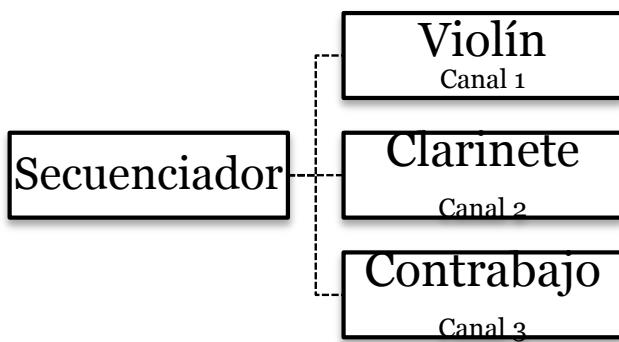
Antes de explicar más detalladamente las características de algunos de los mensajes, conviene conocer dos importantes características de MIDI: los canales y los modos.

#### **2.1.3.1 Canales MIDI**

MIDI está pensado para comunicar un único controlador con varias unidades generadoras de sonido todo por un mismo medio de transmisión. Todos los aparatos conectados a la cadena MIDI reciben todos los mensajes generados desde el controlador. Ello hace necesario un método para diferenciar cada uno de los instrumentos. Este método es el uso del canal.

MIDI puede direccionar hasta 16 canales (también llamados voces, o instrumentos); por ello, al instalar el sistema MIDI será necesario asignar un número de canal para cada dispositivo.

Pongamos un ejemplo para entender este concepto. Si tuviéramos un grupo de instrumentos MIDI compuesto por un clarinete, un violín y un bajo, y quisieramos que se comunicaran para hacer música y grabarlo, lo más apropiado sería asignarles números de canal diferentes, como el 1, 2 o 3. De esta forma estarían correctamente configurados y el sistema podría ejecutar una canción.



*Figura 25: Ejemplo utilizando los canales*

En la figura 25, el secuenciador al tempo establecido enviaría las tramas a los tres instrumentos y estos recibiendo los tres mensajes solo responderían ante su mensaje de canal, reproduciendo el sonido.

Añadir que existen los llamados mensajes de sistema (System Exclusive) que se utilizan para añadir funcionalidades especiales y son ejecutados por todos los dispositivos conectados.

### **2.1.3.2 Modos MIDI**

Dentro del sistema MIDI, se decidió crear una serie de diferentes modos de funcionamiento, según los instrumentos que se conectasen, cada uno con ciertas características. De ello se ha hablado someramente en el apartado 1.2.4.1. La diferencia entre instrumento monofónico y polifónico.

Monofónico: un instrumento monofónico sólo puede reproducir una nota simultáneamente. Es decir, para reproducir una nueva nota debe primero dejar de sonar la anterior. Por ejemplo, los instrumentos de viento son monofónicos, ya que sólo reproducen un único sonido cada vez.

Polifónico: un instrumento polifónico puede reproducir varias notas simultáneamente. Un ejemplo es un piano, que puede formar acordes por medio de hacer sonar dos o más notas a la vez.

Una vez aclarado este aspecto, se resume los modos MIDI en la tabla 3:

| <b>Modo</b> | <b>Nombre</b> |
|-------------|---------------|
| 1           | Omni on/poly  |
| 2           | Omni on/mono  |
| 3           | Omni off/poly |
| 4           | Omni off/mono |

*Tabla 3: Modos disponibles en protocolo MIDI*

Los dos primeros modos se denominan "Omni on". Esto se debe a que en esos modos la información de canal está desactivada. Esas configuraciones se reservan para configuraciones donde sólo se utilice un instrumento. Los otros dos modos, "Omni off", sí admiten la información de canal. Poly o mono indican si el funcionamiento va a ser monofónico o polifónico.

### 2.1.3.3 Mensajes MIDI

Existen dos tipos de mensajes: De estado -*status byte*- y de información -*data byte*-. Se diferencian por el primer bit: si es un 1, es un byte de estado, y si es un 0, es un byte de datos. Al generar un mensaje MIDI, por norma general, siempre enviamos un byte de estado, que puede estar seguido de cierta cantidad de bytes de datos. Por ejemplo, enviar un primer mensaje de estado "activar nota", seguido de un byte de datos informando qué nota es la que se activa.

A su vez, los mensajes de estado se dividen en dos grupos: mensajes de canal y mensajes de sistema. Los mensajes de canal se envían a un dispositivo específico, mientras que los mensajes de sistema son recibidos por todos los equipos.

| Byte estado | Descripción                  |
|-------------|------------------------------|
| 1000cccc    | Desactivación de nota        |
| 1001cccc    | Activación de nota           |
| 1010cccc    | Postpulsación polifónica     |
| 1011cccc    | Cambio de control            |
| 1100cccc    | Cambio de programa           |
| 1101cccc    | Postpulsación monofónica     |
| 1110cccc    | Pitch                        |
| 11110000    | Mensaje exclusivo fabricante |
| 11110001    | Mensaje de trama temporal    |
| 11110010    | Puntero de posición de canal |
| 11110011    | Selección de canción         |
| 11110100    | No definido                  |
| 11110101    | No definido                  |
| 11110110    | Entonación                   |
| 11110111    | Fin de mensaje exclusivo     |
| 11111000    | Reloj para sincronizar       |
| 11111001    | No definido                  |
| 11111010    | Inicio                       |
| 11111011    | Continuación                 |
| 11111100    | Parada                       |
| 11111101    | No definido                  |
| 11111110    | Espera activa                |
| 11111111    | Reseteo del sistema          |

Tabla 4: Todos los mensajes de estado posibles [Leh93]

MIDI es más extenso que lo explicado en estas páginas (Mensajes exclusivos de sistema, sincronización, cambio de pitch...) pero para este proyecto se ha utilizado esta pequeña parte de potencial. En la tabla 4 hay una lista con todos los mensajes de estado disponibles.

Son comandos para ordenar secuenciaciones, cambio de altura, vibratos de instrumento... por tanto acciones que no aparecerán de manera habitual como opciones de una batería electrónica.

Como se explicó en el apartado 1.2.3 el objetivo es enviar el pad sobre el que se ejercita la acción, una nota. Los bytes de estado que se utilizarán son los llamados “desactivación de nota” y “activación de nota”.

En la tabla 3, las primeras casillas, cuyos últimos cuatro bits están marcados como "cccc", se refieren a mensajes a un canal concreto; el resto de bytes son mensajes de sistema y proporcionan órdenes a todos los dispositivos conectados.

Recapitulando en el status-byte se indica el tipo de mensaje y el número de canal. A estos les siguen dos mensajes de datos. En el primer byte de datos (Valor de nota) lleva el número siguiendo el criterio que presenta en la tabla 4. En ella aparecen las 7 notas básicas y sus cinco alteraciones (en notación europea e inglesa) y las diferentes octavas posibles.

| Octava | Números de Nota   |     |    |     |    |    |     |     |      |    |     |    |  |  |
|--------|---|-----|----|-----|----|----|-----|-----|------|----|-----|----|--|--|
|        | DO  | DO# | RE | RE# | MI | FA | FA# | SOL | SOL# | LA | LA# | SI |  |  |
|        | 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                         |     |    |     |    |    |     |     |      |    |     |    |  |  |

**Tabla 5: Correspondencia valor numérico y nota musical en protocolo MIDI [Tomio3]**

Si, por ejemplo, el músico ha pulsado un MI medio el instrumento enviará el número 64 en binario. El rango de datos (0 a 127), junto con la resolución de un semitono da una

cobertura de más de diez de octavas, lo que resulta bastante amplia para los instrumentos conocidos.

De hecho son poco los instrumentos MIDI con un rango tan amplio pero esto no es problema ya que si un instrumento recibe una nota fuera de su rango puede o ignorarlo o trasladarla una o más octavas para que quede dentro de su rango.

En los mensajes de desactivación puede parecer que no tiene sentido el valor de la nota, pero es absolutamente esencial porque en el caso de que se trabaje en uno de los modos que permiten polifonía en un canal necesitamos determinar que nota es la que tiene que acabarse. Como excepción a esto, para instrumentos de batería no es necesario el envío de desactivación de nota. La razón reside en que, por ejemplo, para un teclado influye el tiempo que se mantiene pulsadas las teclas para sonidos que se mantengan durante la presión, mientras que en la percusión sólo influye el impacto transmitido (velocidad) y no tiene sentido hablar de continuidad del sonido. Puesto que si algún batería requiere de alargar la nota de su tambor golpeará repetidamente, con la misma intensidad y a gran velocidad creando el famoso redoble. Los sintetizadores son capaces de ocuparse correctamente de esta diferencia entre instrumentos, ya que se especifica en la norma General MIDI que los instrumentos de batería transmitan por el canal 10.

El segundo byte de datos (Valor de velocidad) lleva el valor de una medida de la fuerza con que se pulsan las teclas o golpean los sensores del instrumento. Este parámetro es importante, ya que es lo que deben captar los sensores de la batería.

Cuando se golpea un instrumento de batería, aparecerá una vibración en la membrana o superficie del mismo, con amplitud máxima en instantes cercanos al impacto y con una disminución de la misma en forma exponencial a medida que pasa el tiempo. El parámetro de velocidad que se debe enviar para la batería es ese valor de amplitud máxima, en los momentos iniciales.

El hecho de llamar a este parámetro “velocidad” se debe a que los primeros instrumentos MIDI eran teclados, y el cálculo de la fuerza con que se pulsaba la tecla se hacía mediante dos sensores colocados en las teclas(uno se activaba al inicio de la pulsación, y el segundo al llegar la tecla al final de su recorrido). La fuerza se hallaba en función del tiempo que se tardara desde que se activaba un sensor hasta que se activaba el otro, es decir, la velocidad que adquiría la tecla en bajar. Para este caso, la velocidad es la fuerza con que se golpean los sensores.

El rango de este 2º byte de datos puede oscilar entre 1 (más débil) y 127 (más fuerte).

#### **2.1.4 General MIDI**

General MIDI (GM) define características específicas de un sintetizador musical que responde a mensajes MIDI. No es por tanto ni un protocolo, ni un formato de ficheros, ni conectores como si es MIDI 1.0. Antes que existiera esta extensión un instrumento sonaba totalmente distinto según quien lo reprodujera. Por consiguiente GM conduce a una compatibilidad de los generadores de sonido (samplers, teclados, tarjetas de sonido, software...). A día de hoy se encuentra actualizada a GM 2 desde el año 1999.

Si el fabricante de un determinado sintetizador desea colocarle el logotipo de General MIDI deberá cumplir una serie de requisitos. [Aso98]

- 128 programas/voces definidas: Por ejemplo el 0 es un piano grande y el 68 es un oboe. Para realizar un cambio de programa se envía una trama de estado en el que se especifica el cambio y a partir de ello se envían las notas musicales con normalidad.
- Sensible al parámetro de velocidad
- Permite 24 voces simultáneas bajo 16 canales distintos.
- Soporte para polifonía.

En lo que respecta al instrumento de la batería electrónica GM establece que se transmítirá por el canal reservado 10. GM2 añade opcionalmente el canal 11. Cada nota MIDI diferente corresponde a un sonido diferente de batería tal y como se observa en la tabla 6. Si se da el caso de juntar instrumentos MIDI que utilicen este canal reservado deberá de ser seleccionado un cambio de programa seleccionando un kit de batería adecuado. Así existirán diferencias apreciables en los sonidos. Por ejemplo en un dispositivo se utilizaría el kit estándar y en otro se programaría el kit 118 (“Sinth Drum”).

Por último comentar que los programas VST del apartado 1.2.4.3 son totalmente compatibles con GM, por lo que identificaran cada uno de los números de la tabla 6.

| <b>Nota</b> | <b>Sonido Batería</b> | <b>Nota</b> | <b>Sonido Batería</b> |
|-------------|-----------------------|-------------|-----------------------|
| 35          | Bombo acústico        | 59          | Ride 2                |
| 36          | Bombo                 | 60          | Bongó agudo           |
| 37          | Golpe de baquetas     | 61          | Bongó grave           |
| 38          | Caja acústica         | 62          | Conga aguda muteada   |
| 39          | Palmas                | 63          | Conga aguda           |
| 40          | Caja electrónica      | 64          | Conga grave           |
| 41          | Goliat                | 65          | Timbal agudo          |
| 42          | Charles Cerrado       | 66          | Timbal grave          |
| 43          | Goliat agudo          | 67          | Agogó agudo           |
| 44          | Pedal charles         | 68          | Agogó grave           |
| 45          | Tom grave             | 69          | Cabasa                |
| 46          | Charles abierto       | 70          | Maracas               |
| 47          | Tom medio grave       | 71          | Silvato corto         |
| 48          | Tom medio agudo       | 72          | Silvato largo         |
| 49          | Crash 1               | 73          | Guiro corto           |
| 50          | Tom agudo             | 74          | Guiro largo           |
| 51          | Ride 1                | 75          | Clave                 |
| 52          | China                 | 76          | Caja china aguda      |
| 53          | Campana del Ride      | 77          | Caja china grave      |
| 54          | Tambourine            | 78          | Cuica apagada         |
| 55          | Splash                | 79          | Cuica                 |
| 56          | Cencerro              | 90          | Apagar triángulo      |
| 57          | Crash 2               | 81          | Triángulo             |
| 58          | Vibraslap             |             |                       |

*Tabla 6: Sonidos de batería definidos por General MIDI[Dmidi]*

## **2.1.5 MIDI over USB**

Actualmente resulta laborioso conseguir un ordenador con conectores MIDI. Las tarjetas de sonido convencionales no disponen de ellos. La solución que se ha planteado desde el sector es utilizar un conversor MIDI-USB (figura 26) que por un módico precio de 16€ posibilitan utilizar instrumentos digitales auxiliados por un ordenador conectado por el conocido conector USB.

### **2.1.5.1 Fundamentos USB**

USB (Universal Serial Bus) es un estándar de protocolo de comunicaciones serie. Usado masivamente desde hace años en los ordenadores personales para la comunicación con dispositivos.[Usboo]

A nivel físico utiliza la codificación NRZI que cambia de nivel cuando aparece un o lógico y se han introducido diversos conectores cada vez más pequeños. Aunque siguen teniendo en común la disposición de los contactos. 2 de alimentación VCC y GND y dos de transmisión D+ y D-.

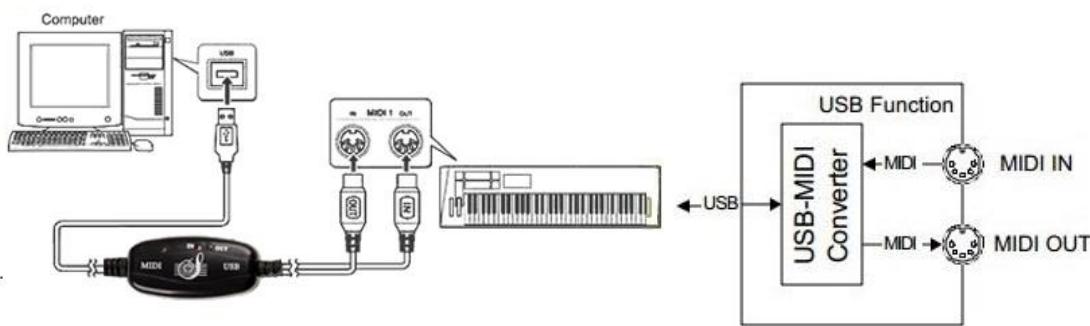
El protocolo comunicación se basa en que el Host (ordenador) inicia la transferencia y el dispositivo (device) responde. Cada dispositivo está identificado por una única dirección lógica y es una comunicación fiable y orientada a conexión. La topología de bus que lo identifica es en árbol puesto que la conexión se expande gracias a dispositivos Hub.

La comunicación es full speed. Hay cuatro tipo de transferencias: Bulk (transferencia masiva), Isochronous (divide el ancho de la trama entre los diferentes dispositivos para obtener tiempo real), Interrupt (transferencia secuencial cada x ms), y Control (utilizadas en bulk para controlar el flujo de información).

### **2.1.5.2 Conversor USB-MIDI**

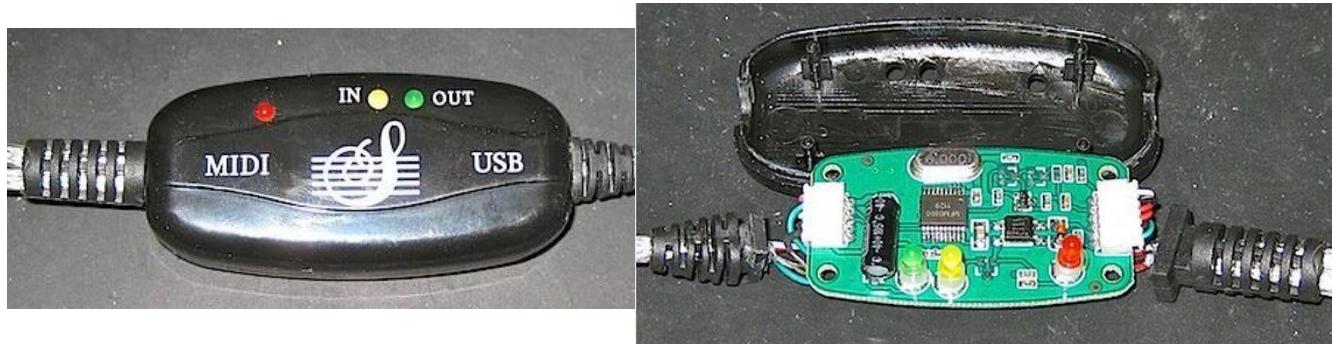
Características técnicas: [Du2m]

- Interfaz de alta velocidad, compatible con USB 2.0.
- Cumple estándar MIDI. Conectores y conexiones eléctricas. Además de la disposición a utilizar los 16 canales lógicos.
- Totalmente Plug&Play. No es necesario instalar controladores en Windows XP/7; se puede utilizar directamente. Cumple la directiva de diseño de productos Universal Serial Bus Device Class Definition for MIDI Devices. [USB99]



*Figura 26: Conexión MIDI-USB [Dedru] y Diagrama de bloques [Du2m]*

Como las especificaciones del distribuidor son muy ambiguas. [Du2m] El dispositivo fue abierto para su análisis (figura 27).



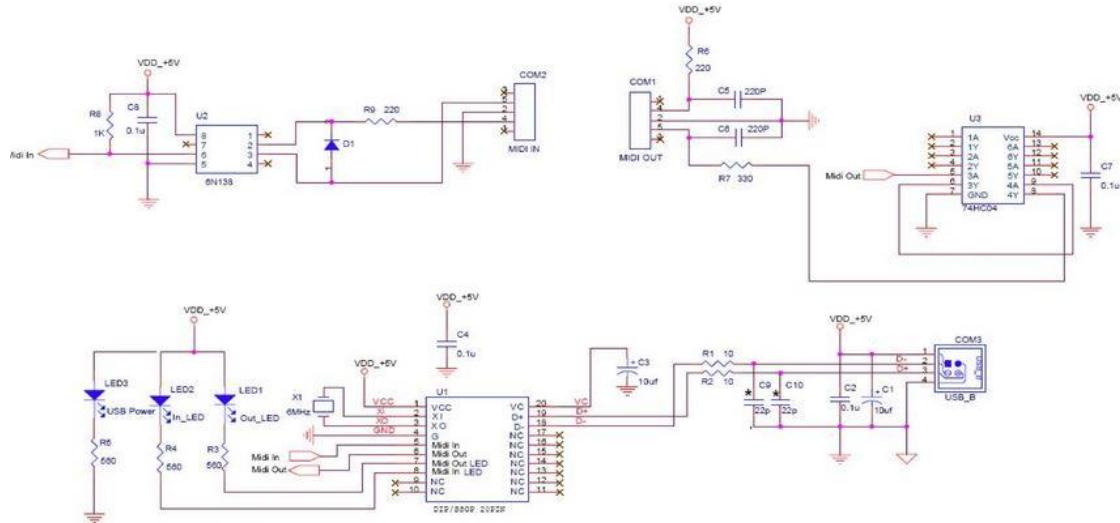
*Figura 27: Vista del dispositivo y de la cara de componentes del conversor USB*

El integrado principal se denomina MFM0860. Es un componente de propósito específico que adapta tramas MIDI a USB. Solo es necesario alimentarlo, (VDD y GND) colocarle un cristal en las patillas XI y XO y un condensador electrolítico en VC. [Dmfmo] Con esta configuración tan simple gestiona tramas MIDI a través de USB y viceversa.

En el datasheet del fabricante se encuentra un esquemático completo que se puede asemejar al dispuesto en la PCB de la figura 27. [Dmfmo] Se parte a identificar bloques funcionales entre la figura 27 y el esquema eléctrico proporcionado por el fabricante en la figura 28.

El dispositivo identifica los comandos señalizando a nivel bajo las líneas para LEDS (MIDI OUT LED y MIDI IN LED) y construye una nueva trama siguiendo las especificaciones de USB 2.0. En el esquema se aprecian los Leds indicativos. El led rojo señala cuando se conecta la alimentación. El Led amarillo cuando se recibe una trama desde el conector MIDI IN para transmitirla al ordenador y el led verde cuando se recibe un mensaje del ordenador al instrumento MIDI.

También se aprecian los conectores MIDI y USB, así como los componentes asociados a estas interfaces (optoacoplador, condensadores). El sistema digital es el componente principal y encargado de recoger la trama y tras el procesamiento enviarla en otro protocolo. Como se ha dicho el sistema sea alimenta a los 5V de USB y el integrado va a una velocidad marcada por el cristal de 6Mhz.



**Figura 28: Esquema eléctrico conversor MIDI-USB [Dmfmo]**

Por último, al comprender el mecanismo se asegura un incremento de latencia, al introducir en el sistema de comunicación MIDI; formado por instrumentos y ordenador, la unidad de procesamiento del conversor.

## 2.1.6 Ejemplo práctico MIDI

En todo este capítulo se ha analizado el protocolo de comunicaciones MIDI las claves a utilizar. El propósito de este apartado será un acercamiento desde un producto ya elaborado y comercializado. Se analizarán tramas MIDI. Estos mensajes podrían ser utilizados para grabar una melodía o reproducir con una mejor calidad o simplemente crear una partitura.

Los materiales que se utilizaron son los siguientes:

- Piano Casio CTK-431: Un sintetizador con forma de teclado. Dispone de 49 teclas, 100 melodías del banco de canciones (con un controlador para ensayar que te va guiando), 100 instrumentos y 12 notas polifónicas.



*Figura 29: Piano Casio CTK-431:[Dedru]*

- Conversor MIDI- USB: Figura 26.
- MIDI-OX: Programa multifunción de libre distribución, todo relacionado al MIDI. Permite monitorizar los eventos MIDI en pantalla (este será su uso aquí), capturar sysex y enviarlos, aplicar filtros para descartar información. [Dmiox]
- Osciloscopio Agilent Technologies modelo DSO 5014 propiedad de la EUITT: Configurado correctamente (colocado en modo disparo, con una sensibilidad adecuada)
- Resistencia de 1k y protoboard: Resistencia de 1000 ohmios unida al puerto MIDI OUT transformará la corriente de salida en tensión que medir en el osciloscopio. El cero lógico será una corriente de 5mA que transformará en una tensión de 5V, tal y como se ve en la figura 32. El uno lógico al ser una corriente inapreciable será una tensión de 0V.

En primer lugar, configurar el programa MIDI-OX. Para ello de manera muy resumida se selecciona en el panel de control: 1º) Options-> Pass SisEx 2º) Options-> MIDI Devices-> Seleccionar “Dispositivo de audio USB” y pulsar OK 3º) Seleccionar en el panel de control el icono del semáforo para ponerlo “verde” 4º) Comenzar a enviar datos MIDI.

Más en detalle, monitorizar el instrumento, en el que se observan diferentes campos. Ordenados temporalmente, se ha pulsado SOL-LA-SI-DO (figura 30) que se indican con notación americana. Primero se envía la trama de nota on seguido de la trama de nota off cuando se ha levantado de pulsar la tecla.

| TIMESTAMP | IN PORT | STATUS | DATA1 | DATA2 | CHAN | NOTE | EVENT        |
|-----------|---------|--------|-------|-------|------|------|--------------|
| 205881    | 1       | --     | 144   | 79    | 100  | 1    | G 5 Note On  |
| 205961    | 1       | --     | 144   | 79    | 0    | 1    | G 5 Note Off |
| 206831    | 1       | --     | 144   | 81    | 100  | 1    | A 5 Note On  |
| 206951    | 1       | --     | 144   | 81    | 0    | 1    | A 5 Note Off |
| 207721    | 1       | --     | 144   | 83    | 100  | 1    | B 5 Note On  |
| 207841    | 1       | --     | 144   | 83    | 0    | 1    | B 5 Note Off |
| 208511    | 1       | --     | 144   | 84    | 100  | 1    | C 6 Note On  |
| 208591    | 1       | --     | 144   | 84    | 0    | 1    | C 6 Note Off |

*Figura 30: Panel de MIDI-OX con ejemplo de tecla pulsada*

El bit de estado envía el comando seguido del canal sobre el que se envía, en este caso canal 1. En el tercer bit de datos se observa que no hay detección de velocidad puesto que la fuerza ejercida ha sido distinta en cuatro ocasiones y en cambio, el dato enviado ha sido el mismo, el número 100.

Para visualizar la trama físicamente el proceso de medida es el siguiente: En la proto-board de la figura 31 se coloca una resistencia de 1k y se conectan dos cables que irán a las patillas 2 y 4 del MIDI OUT.



*Figura 31: Conexión entre salida de piano y osciloscopio*

A continuación, se pulsa una nota conocida, por ejemplo SI y además de reproducir el sonido en el entorno, envía la trama MIDI que recoge el osciloscopio en la figura 32.

En la figura se han dibujado la secuencia de bits que se observa. Indicando el bit de inicio y el bit de parada de la transmisión asíncrona; así como los bits más y menos significativos. En la parte inferior de la figura 34 también se observa la decodificación de la trama. Se envía primero el byte de estado de nota on, luego el primer byte de datos que es la nota SI, y por último el segundo byte de datos que es la velocidad. Al no ser sensible el teclado envia una nota estándar como es el 75 de 127 posibles.

Comunicación teórica al pulsar la nota SI medio de un teclado

- Byte de estado: 1001xxxx (note on); Las x será el canal de la transmisión que se puede modificar. En este caso 0.
- Primer byte de datos: 01010011 (valor 83 que corresponde a la nota Si de la quinta octava. En notación americana "B5")
- Segundo byte de datos: 01001011 (velocidad 75).

Pero al soltarla se puede omitir el byte status y apagarla por volumen enviando segundo byte de datos: 0oooooooo (velocidad cero). Otra posibilidad es que usase el 1000xxxx (note off) para apagarla.

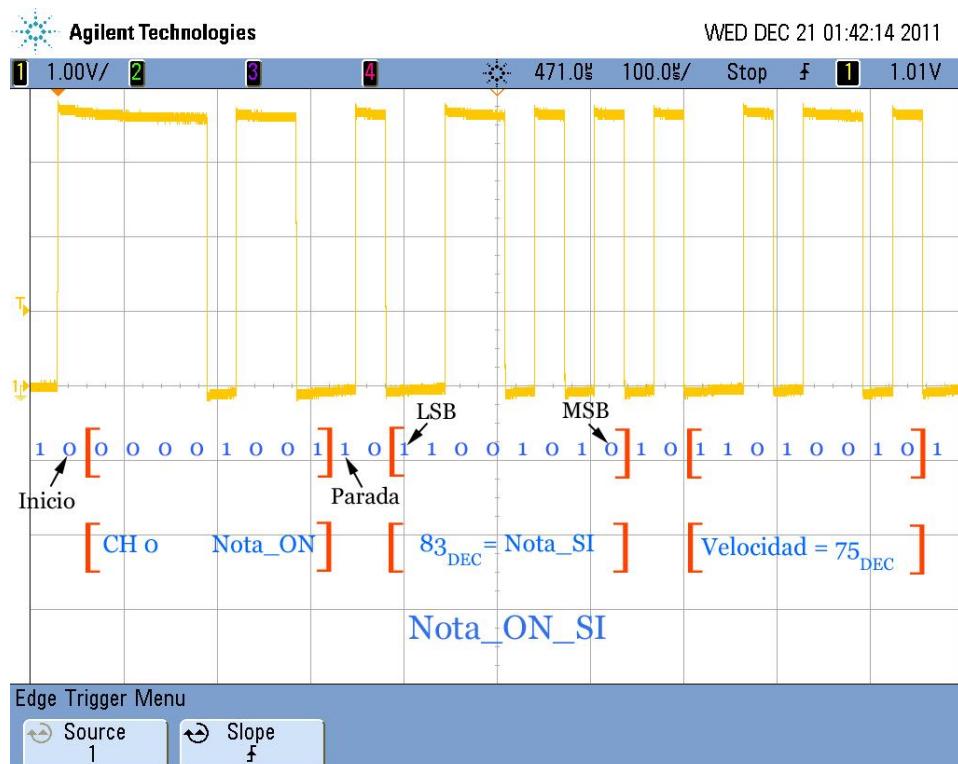


Figura 32: Análisis de la trama leída del osciloscopio

## 2.2 Sensores

Los sensores son los dispositivos de entrada capaces de obtener información del exterior de una magnitud física concreta. [Palo3] El dispositivo necesario para medir el gesto producido por el músico al levantar la baqueta y golpearla sobre la superficie. Aquél capaz de captar la energía mecánica (movimiento de la baqueta) y transformarla en energía eléctrica para ser procesada.

Se utilizarán transductores electromecánicos que convierten la energía de entrada producida por el golpe en energía eléctrica. Se realizará un estudio de los efectos físico-eléctricos y se analizarán las diversas opciones que nos plantea la documentación, buscando la que mejor se adapte.

### 2.2.1 Sensores de presión

La presión es una magnitud física vectorial que mide la fuerza en dirección perpendicular a una superficie. [Tipo6] Es la magnitud que es necesario medir en el sistema de percusión.

En la figura 33 se tienen los sensores de presión disponibles [Palo3] Siguiendo los siguientes criterios se concluye que se requieren sensores piezoelectrinos:

- La deformación es global y deben ser ligeros para su colocación
- La transferencia debe ser rápida en los cambios de variaciones de presión.
- La información que se requiere es dinámica, no estática.

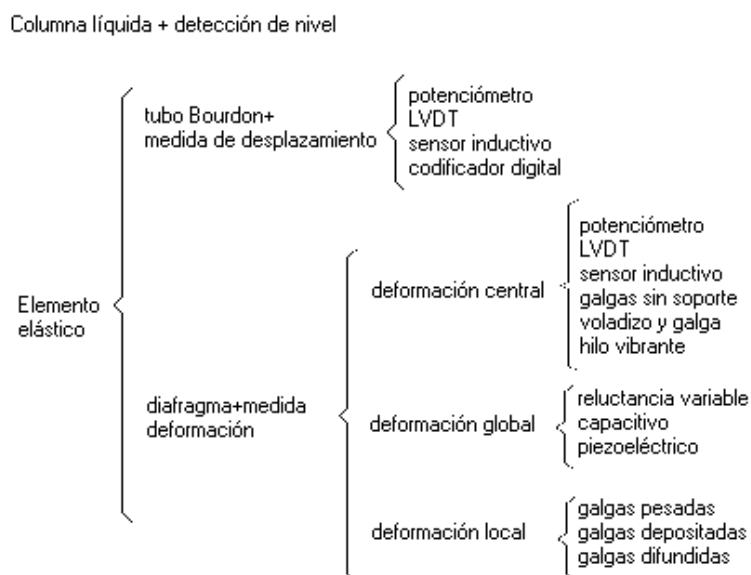


Figura 33: Esquema sensores de presión [Palo3]

## 2.2.2 Sensores piezoelectricos

La palabra “piezo” se deriva de una palabra griega cuyo significado es apretar u oprimir. Fue en 1880 cuando el matrimonio Curie descubrió que al aplicar presión a un cristal de cuarzo se establecían cargas eléctricas como en el dibujo de la figura 34. Este es el principio de lo que más tarde se llamó efecto piezoelectrico.

Algo más tarde se descubrió que era un efecto reversible, es decir, que al aplicar un campo eléctrico al cristal se produce una deformación. [Cupoo] Por consiguiente se descubrió que estos materiales son transductores y pueden ser utilizados para convertir energía mecánica en eléctrica y viceversa.



Figura 34: Efecto piezoelectrico[Cupoo]

Las aplicaciones ante éste fenómeno tardaron en llegar siendo el fundamento de detectores ultrasónicos en submarinos en la 1º Guerra Mundial. A día de hoy su uso está más extendido. Algunos ejemplos: Encendedores en mecheros, altavoces en despertadores.

En las últimas décadas se han elaborado materiales sintéticos, con propiedades mejoradas. Esto es debido a que el efecto piezoelectrico mostrado por materiales naturales tales como el cuarzo, la sal de Rochelle y la turmalina es muy pequeño siendo problemáticos para los sistemas de instrumentación. Los materiales sintéticos como titanatos-circonatos de plomo (PZT), fluoruro de polivinilideno (PVDF) son muestras de ello.

Existen un número muy grande de materiales con propiedades piezoelectricas. Si quiéramos clasificarlos podríamos hacerlo en cristales de cuarzo hidrosolubles, monocrystales, semiconductores piezoelectricos, cerámicas piezoelectricas, polímeros y compuestos piezoelectricos.[Pal03]

### 2.2.2.1 Ventajas

Entre las propiedades de los sensores piezoelectricos se puede destacar:

- Alta sensibilidad.
- Bajo coste.
- Alta rigidez mecánica debido a que las deformaciones sufridas, dada su alta sensibilidad son inferiores a 1um. Son capaces de aguantar impactos.
- Tamaño pequeño.

#### **2.2.2.2 Limitaciones**

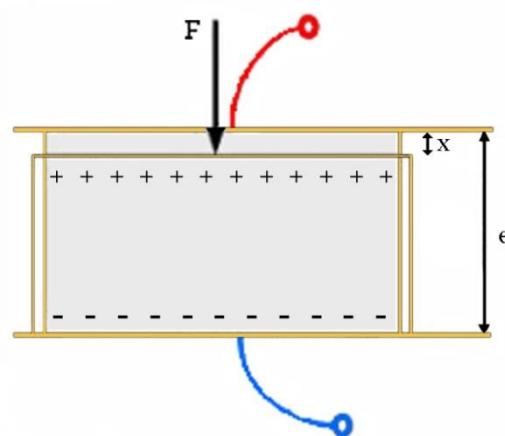
Es muy importante tener conocimiento de las limitaciones inherentes de estos materiales para la realización de un análisis preventivo, evitando así, recalcular para solucionar problemas previsibles.

Las principales limitaciones de los materiales piezoelectricos son las siguientes: [Ert11]

- Envejecimiento natural (y acelerado por las condiciones de uso).
- Inestabilidad de las propiedades en función de variaciones de temperatura. Temperatura de Curie.
- Límites de tracción mecánicos. También máxima tensión antes de la rotura.
- Límites eléctricos.

#### **2.2.2.3 Comportamiento eléctrico de los materiales piezoelectricos.**

Profundizando en el estudio de estos dispositivos mediante la observación de la figura 35. En ella aparece un material piezoelectrico con un espesor  $e$  al que se le aplica una fuerza  $F$ , provocando la siguiente deformación  $x$  en forma de compresión. [Mon10] Se muestran sendas placas metálicas a las que se solapan dos cables de color rojo y azul.



**Figura 35: Comportamiento eléctrico piezoelectrico [Mon10]**

Se ha simplificado el estudio con el material sólido omitiendo deformaciones, compresiones y cizalladuras en el resto de posiciones de los ejes.

Continuando sobre la figura 35 la suma de los comportamientos inercial, elástico y viscoso del material (modelo no ideal) provoca la aparición de fuerzas que contrarrestan el efecto de la fuerza  $F$  resultante. Comprendiendo 3 componentes de fuerza. Por un lado la inercia del movimiento definida por la 1º Ley de Newton. Por otro la elasticidad del material por la Ley de Hooke. Y en tercer lugar, la fricción o fuerza de rozamiento siempre dirigida en sentido opuesto de la velocidad. [Sero8] Expresado en las siguientes ecuaciones siendo  $m$  la masa del cristal,  $r$  el coeficiente de rozamiento,  $s$  el coeficiente elástico y  $x$  la deformación total producida.

$$m \cdot \frac{d^2x}{dt^2} = -r \cdot \frac{dx}{dt} - s \cdot x$$

Trasladando los componentes:

$$F(t) = m \cdot \vec{a} + r \cdot \vec{v} + s \cdot x = m \cdot \frac{d^2x}{dt^2} + r \cdot \frac{dx}{dt} + s \cdot x$$

La relación entre entrada y salida satisface una ecuación diferencial de coeficientes constantes. Este tipo de ecuaciones caracteriza un sistema LTI continuo [Opp98]. Exactamente se trata de un sistema de segundo orden; un sistema subamortiguado.

Las condiciones iniciales serán nulas porque se estudiará como sistema causal.

Ahora hay que tener en cuenta el efecto piezoelectrónico que provoca la aparición de una carga  $q$ , directamente proporcional a la deformación producida  $x$  e, inversamente proporcional al espesor  $e$  del cristal:

$$q = \frac{k'}{e} \cdot x$$

Si los terminales del dispositivo se cierran sobre un circuito, aparece una corriente eléctrica  $i$  según la ecuación:

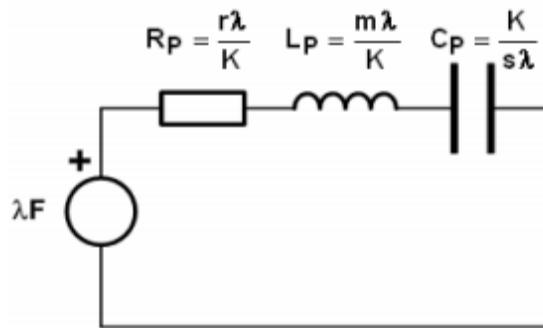
$$i = \frac{dq}{dt} = \frac{k'}{e} * \frac{dx}{dt} = K * \frac{dx}{dt}$$

Esto significa que la corriente que aparecería sería proporcional a la velocidad de variación de la deformación y que, con una fuerza estática aplicada, la deformación sería constante y la corriente tendería a anularse.

Considerando las expresiones anteriores podemos escribir:

$$F = \frac{m}{K} \cdot \frac{di}{dt} + \frac{r}{K} \cdot i + \frac{s}{K} \cdot \int i \cdot dt$$

Esta expresión tiene la misma forma que la ecuación que rige la evolución de las variables eléctricas de un circuito RLC en serie lo que permite establecer una analogía con ese circuito [Mon10] y se representa en la figura 36:



*Figura 36: Analogía mecánico-eléctrica [Mon10]*

Un circuito RLC serie que se excita con una fuente de tensión de valor V está representado por:

$$V = L \cdot \frac{di}{dt} + R \cdot i + \frac{1}{C} \cdot \int i \cdot dt$$

La ecuación que representa al circuito será pues:

$$V = \lambda F = R_p \cdot i + L_p \cdot \frac{di}{dt} + \frac{1}{C_p} \cdot \int i \cdot dt$$

Donde lambda es un factor de conversión dimensional.

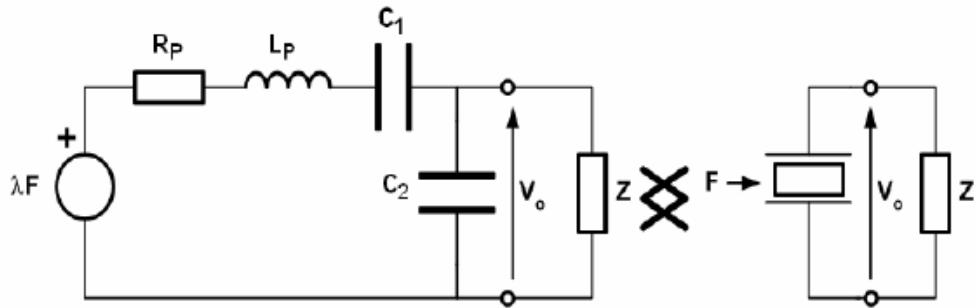
Este circuito representa perfectamente el comportamiento del dispositivo piezoelectrónico, pero no se ha tenido en cuenta ni la conexión, (los cables rojo y azul de la figura 35 que se sueldan sobre los electrodos) ni el circuito a conectar. Todo ello afecta al comportamiento.

Por un lado el par de placas de electrodos y los cables soldados a ellas, separadas por el material piezoelectrónico, actúan como un capacitor C en paralelo con el cristal. Asumiendo que tiene una superficie A, que los electrodos están separados a la distancia e y que la permitividad dieléctrica absoluta del cuarzo es epsilon, se obtiene:

$$C = \varepsilon \cdot \frac{A}{e}$$

En la figura 37 se incluye este condensador. Con los valores típicos de la mayoría de los cristales se puede considerar sin demasiado error que  $C_2 >> C_1$ . [Palo3]

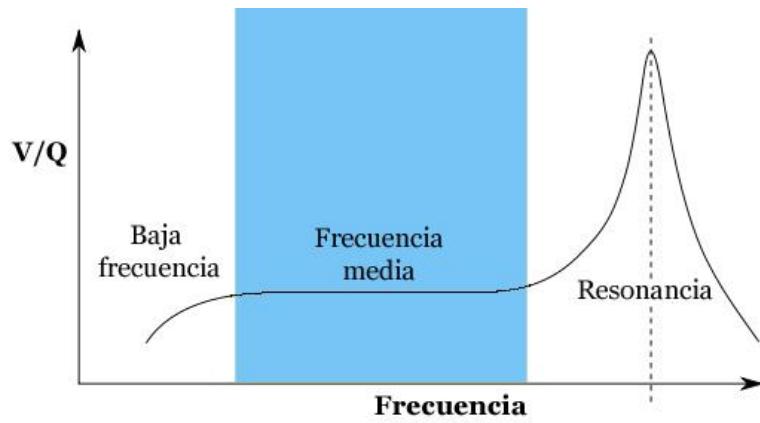
Otro efecto para conocer el comportamiento de estos sensores es al conectar el sensor a una carga (figura 37). [Mon10]



*Figura 37: Circuito cerrado sensor piezoelectrónico [Mon10]*

Si la carga tiene una componente resistiva y la fuerza es estática, la presencia del condensador  $C_1$  provocará que la señal de tensión en la salida vaya cayendo de valor hasta llegar a cero. En la figura 38 con la respuesta en frecuencia del dispositivo piezoelectrónico, aparece una frecuencia de corte inferior. En el fondo, lo que está ocurriendo es que se trata de un sensor generador de señal y sólo se introduce una cantidad limitada de energía por lo que no puede mantenerse indefinidamente en la salida una tensión sobre la resistencia.

Con cargas dinámicas (del tipo de vibraciones) se podrá desarrollar una tensión de salida de forma permanente ya que se estará aportando energía constantemente. Sin embargo, en este caso, habrá que tener en cuenta el tipo de carga ya que la presencia de componentes reactivas modificará el comportamiento del sistema (su frecuencia de resonancia y su ganancia). Como se observa en la figura 38, la respuesta en frecuencia presenta un pico de resonancia muy fuerte.



**Figura 38: Respuesta en frecuencia sensor piezoelectrónico [Dwiki]**

La banda con respuesta plana es lo que interesa pues se comporta como un sistema LTI que depende de la magnitud sensorial y de la sensibilidad. A la vista de lo que presenta la curva de la respuesta en frecuencia de la figura 40 y siempre que se trabaje en la zona plana se desprecia los efectos capacitivo e inductivo. [Palo3]

En general, los fabricantes proporcionan el valor del condensador de salida, la sensibilidad al parámetro y la frecuencia de resonancia. Esta poca información se observa en el anexo de esta memoria.

Además esto provocará que el soporte mecánico (donde quede fijado el sensor) influya principalmente en el amortiguamiento y en la frecuencia a la que se quede vibrando. La relación entre la fuerza aplicada y la correspondiente respuesta del piezoelectrónico depende de 3 factores:

- Dimensiones y geometría del dispositivo.
- Propiedades piezoelectricas del material.
- Dirección de la excitación, ya sea mecánica o eléctrica.

También hay que tener en cuenta que la impedancia de salida que presentan los materiales piezoelectricos es alta (condensador muy pequeño con alta resistencia de fugas), por lo que para medir la señal generada hay que tener cuidado con los sistemas de medición y acondicionadores para no perder nivel en la señal. [Palo3]

También es interesante recalcar la sujeción del cable. Como precaución adicional y debida a que se gestionan vibraciones el cable debe apoyarse firmemente. En caso contrario aparecerá el efecto triboeléctrico fenómeno de generación de electricidad por frotamiento.

#### 2.2.2.4 Elección del piezoelectrónico

Ante todo, en la elección del sensor, ha de primar la sencillez y economía. Comercialmente, hay disponibles sensores de presión de gran precisión y con gran exactitud en sus parámetros, pero son demasiado caros, y ello los hace inviables para este proyecto.

Así que, como alternativa, se han usado discos cerámicos piezoelectrónicos que son muy utilizados en otras aplicaciones, como zumbadores, y que son muy baratos y asequibles. [Tomo3]

#### Transductor piezoelectrónico

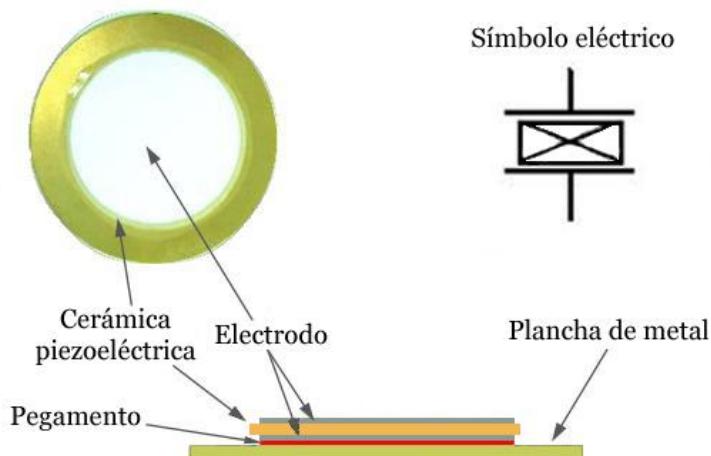


Figura 39: Sensor cerámico piezoelectrónico

En la figura 39 aparece la cerámica piezoelectrónica y el símbolo eléctrico para identificarlo que ya se ha usado en la figura 37. El transductor está compuesto por la cerámica piezoelectrónica, sensible a la fuerza; el electrodo, polarizado eléctricamente y sobre el que se suelda el cable; la plancha de metal que transmite las vibraciones y el pegamento que une ambos. La cerámica piezoelectrónica de tipo PZT (Titanato Zirconato de Plomo) son las más utilizadas.

Por su geometría se pueden fijar fácilmente a superficies planas con un buen pegamento adhesivo, y mantiene sus características piezoelectrísticas, esto es, cuando se deforme la superficie a la que va unido, el cristal vibrará coherentemente y producirá un voltaje proporcional entre sus caras metálicas. Así se aprovechará su funcionalidad generadora.

Como inconveniente queda decir que no se disponen de hojas de características (ver anexo) que permitan realizar cálculos convenientes como sensor de presión. En contrapuesta se realizarán pruebas y testeos para elegir la mejor configuración.

Un hecho que marcó la elección fue descubrir que las cerámicas piezoeléctricas se han utilizado siempre en los productos comerciales de batería electrónica. Son económicos y fiables.

Un dato que corrobora esta afirmación es el nuevo pad de Yamaha modelo XP100D y que podemos observar en la figura 40. Es una pieza que ronda el precio de 450€ y está revolucionando a los amantes de las baterías electrónicos, pues posee un tacto mejorado. El secreto es un gel de polímero de silicona en la zona de golpeo. Pero aun siendo la novedad lleva internamente el mismo sensor cerámico piezoeléctrico. [Dyama]



**Figura 40:** Pad Yamaha modelo XP100D y vista interior piezoeléctrico

El piezoeléctrico del interior de la figura 40 está pegado sobre una superficie metálica unida a dos cables. Se encuentra colocado en el centro para recoger todas las vibraciones producidas por el golpe y perfectamente protegido.

## 2.3 El microcontrolador

Los microcontroladores son circuitos integrados capaces de ejecutar órdenes almacenadas en memoria y con posibilidad de reprogramación. Disponen además de periféricos para funciones dedicadas (temporizadores, bloques de lógica...) y comunicación.

Se realizará un estudio de mercado de distintas plataformas. Sumergiéndose en una de ellas. Se comentará en este apartado aspectos no tan conocidos como Hardvard, RISC y pipeline. En el último subapartado se explicarán los periféricos utilizados.

### 2.3.1 Plataformas de Desarrollo

Las plataformas de desarrollo son una manera sencilla y de relativo bajo coste para empezar al amparo de una tecnología. En numerosas ocasiones son creadas por los propios fabricantes para dar a conocer sus productos.

Estas plataformas suelen poseer un HW muy simplificado dando prioridad al elemento central (FPGA, CPLD...) y un IDE (Integrated Development Environment) en el que se elaboran las aplicaciones. Generalmente gratuito aunque con limitaciones de uso.

Además se nutre de la idea de la plataforma de prototipado rápido. Un hardware modular que permite por un lado, una fácil expansión y por otro lado, la reutilización del hardware ya que se omitirían fases de pruebas y cálculos. En definitiva es una mejora del ciclo de diseño. [Whe11][Dtexas]

En el mercado de los microcontroladores, que es el buscado, existen opciones muy variadas dependiendo de los propósitos que se quieren derivar. La decisión debería ser tomada basándose en criterios de capacidad, prestaciones y coste.

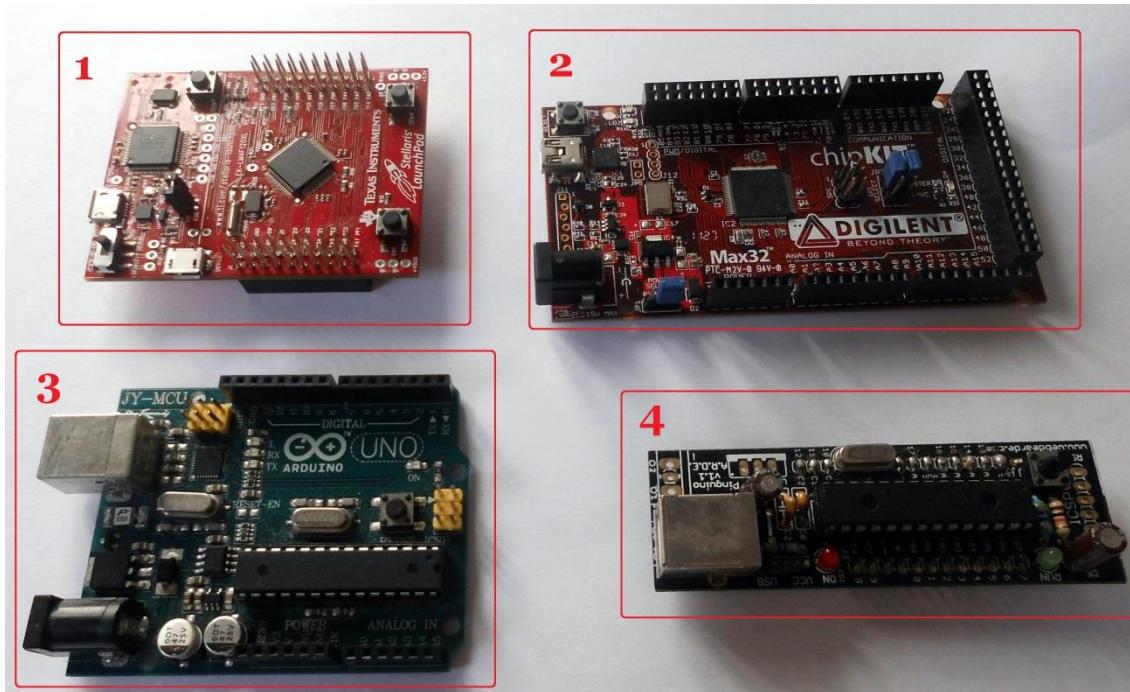
En el caso que atañe a esta memoria, la primera elección se ha llevado a cabo teniendo como criterios lo explicado en las especificaciones como parte esencial; apartado 3.1. Por ejemplo que posean ADC. También la fortaleza de la empresa en el mercado de los microcontroladores y la estabilidad de la plataforma basada en la eficacia en otros usos.

En la figura 41 se presenta dicha elección. (1) Stellaris Launchpad, (2) ChipKIT, (3) Arduino, (4) Pingüino.

Todas ellas disponen de comunicación con PC vía USB, pudiéndose programar y probar sin necesidad de dispositivo externo.

Además los microcontroladores de todos ellos están perfectamente documentados con Datasheets o Application Notes; siendo este apartado un pequeño bosquejo de características que se consideran esenciales para continuar. Para cualquier aclaración o consulta dirigirse a la documentación del fabricante.

En el caso que atañe a esta memoria, la primera elección se ha llevado a cabo teniendo como criterios lo explicado en las especificaciones (apartado 3.1) como parte esencial. Por ejemplo que posean ADC.



*Figura 41: Plataformas de desarrollo analizadas*

Detallando cada plataforma:

- Stellaris: Placa basada en un ARM de 32 bits creada por Texas Instruments. Extremadamente potente el núcleo es un ARM Cortex M4 (100DMIPS @ 80Mhz) con una unidad especial de coma flotante y la multiplicación en un solo ciclo. Muy barato (12€ con envío incluido). El microcontrolador posee periféricos avanzados. La placa se puede colocar en una protoboard y viene con debugger integrado.
- Pinguino: Placa basada en un Pic16F2550. Es la respuesta a una pequeña parte de la comunidad de Arduino que exigía una simplicidad y mayor liberalización del HW. Se alimenta de la cantidad de proyectos elaborados para PIC en la de

cada de los 90 y 2000. Destaca por poseer USB nativo y la velocidad de procesamiento de su arquitectura Harvard.

- Chipkit: Placa basada en un PIC32MX795F512 realizada por Digilent. Es la placa más potente de las mostradas (105DMIPS @ 80Mhz). El núcleo es de la familia MIPS.
- Arduino: Placa basada en un Atmega328P. A día de hoy es la plataforma de referencia. Simplificando el software y el hardware ha conseguido que se hagan multitud de proyectos, desde domótica hasta telemedicina.

En la tabla 7 se ha realizado una comparación entre las características de los microcontroladores integrados en las plataformas de desarrollo de la figura 41. El resto de elementos que acompañan y la hacen funcionar (alimentación, conexión USB, tiras de pines, reloj de cuarzo...) son muy similares entre ellas.

| <b>Microcontrolador</b> | <b>Memoria</b>               | <b>Perifericos</b>                         | <b>Comunicación</b>  | <b>Velocidad de procesamiento</b> |
|-------------------------|------------------------------|--|--|-----------------------------------|
| <b>TM4C123GH6PM</b>     | F-256KB,<br>R-32K,<br>E-2K   | 12 Ch ADC, 24 timers, 16 CCP, 3 Comparador | 43 GPIO, 4 SPI, 4 I2C, 8 UART, 1USB, 2 CAN                     | 100 DMIPS @80Mhz                  |
| <b>Pic16F2550</b>       | F-32KB,<br>R-2048,<br>E-256  | 10 Ch.ADC, 4 timers, 2CCP;                 | 28 GPIO, 1 USART, 1 SPI, 1 I2C, 1 USB                          | 12 MIPS @48 Mhz                   |
| <b>PIC32MX795F512</b>   | F-512K,<br>R-128K            | 16 Ch ADC, 5 timers, 5 CCP, 2 Comparadores | 85GPIO, ,5USART/SPI/I2C, 1 USB USB 2.0, 10/100 Ethernet, 2 CAN | 105DMIPS @80Mhz                   |
| <b>Atmega328P</b>       | F-32KB,<br>R-2048,<br>E-1024 | 8 Ch ADC, 3 timers, 6 CCP, 1 Comparador    | 23 GPIO, ,1USART, 2SPI, 1 I2C                                  | 16 MIPS @ 16 Mhz                  |

*Tabla 7: Comparación de características de los distintos microcontroladores*

Por ejemplo, siguiendo la tabla 7, en la columna de memoria existen claras diferencias en las cantidades suministradas. Se utiliza memoria flash para el código de programa, RAM para la memoria de datos volátil y EEPROM para datos que se necesitan almacenar de manera duradera. La plataforma Chipkit es la única que no incluye memoria EEPROM pero es la que contiene más memoria de otros tipos.

Respecto a periféricos todos disponen de ADC, timers, módulos de captura, comparación, PWM y algún comparador analógico. La diferencia radica en el número de periféri-

cos y de las características internas de estos. Por ejemplo el ADC de Stellaris posee 12 canales de 12 bits con un rango de conversión de 0 a 3,3V y una fadc de 1 Mhz. En cambio el Pinguino posee 10 canales con una resolución de 10 bits, un rango de 5V y una frecuencia de conversión máxima de 10Khz.

La comunicación en estas plataformas resulta esencial dada la filosofía modular explícada. Aquí se distinguen pines de entrada salida (GPIO), comunicaciones serie síncronas (SPI, I2C) o asíncrona (UART) o incluso si posee módulos dedicados para comunicaciones avanzadas (Ethernet, CAN, USB).

Por último la velocidad de procesamiento; la cantidad de instrucciones que es capaz de medir por unidad de tiempo. Existen 2 reglas los MIPS que indican los millones de instrucciones por segundo que es capaz de ejecutar y los DMIPS (Dhrystone MIPS) que es un benchmarking más avanzado que indica cómo se comporta sobre una referencia. DMIPS es más estándar y, se evalúan sobre un lenguaje de alto nivel, su compilador y sus bibliotecas al resolver un algoritmo. De esta forma la velocidad de ejecución de instrucciones en ensamblador es menos importante.

Tras probar los entornos de desarrollo de todas ellas y viendo las características en periféricos y memoria vemos que todas pueden adaptarse a la solución. Ahora bien con la tabla 8 emergen otro tipo de opciones que barajar. Se toma como referencia los tiempos de desarrollo y aprendizaje de un uso extensivo con las librerías. Además de la amplia documentación así como la solución que te pueda aportar la comunidad (personas registradas a marzo de 2014).

| <b>Plataforma</b> | <b>Microcontrolador</b> | <b>Coste (€)</b> | <b>Comunidad(1*)</b> | <b>Documentación</b> |
|-------------------|-------------------------|------------------|----------------------|----------------------|
| <b>Stellaris</b>  | ARM 32bits              | 12€              | 2.100                | Abundante            |
| <b>Pingüino</b>   | PIC de 16 bits          | 16 €             | 1.700                | Amplia pero cerrada  |
| <b>Chipkit</b>    | PIC de 32 bits          | 50 €             | 2.900                | Poca                 |
| <b>Arduino</b>    | Atmega 8 bits           | 25 €             | 183.767              | Inmensa y muy simple |

*Tabla 8: Comparación plataformas de desarrollo*

Por lo que la opción más lógica ha sido tomar al líder de estas plataformas, Arduino. En el supuesto caso de que las características sean un problema existe la opción de migrar fácilmente a una plataforma de la familia más potente como MEGA o también DUE. [Dar-dú]

## 2.3.2 El Atmega328p

El Atmega328P es un microcontrolador CMOS de 8 bits a baja potencia basado en arquitectura RISC de AVR. Ejecutando las instrucciones en un solo ciclo de reloj, el Atmega32 alcanza un desempeño de 1 MIPS por MHz permitiendo al diseñador optimizar consumos de potencia contra la velocidad de procesamiento. [Whe11]

### 2.3.2.1 Nucleo AVR de Atmel

La principal función del núcleo del CPU es asegurar la correcta ejecución de un programa. El CPU debe ser capaz de acceder a la memoria, ejecutar cálculos, controlar los periféricos y manejar las interrupciones. La figura 42 muestra el diagrama de bloques de la CPU.

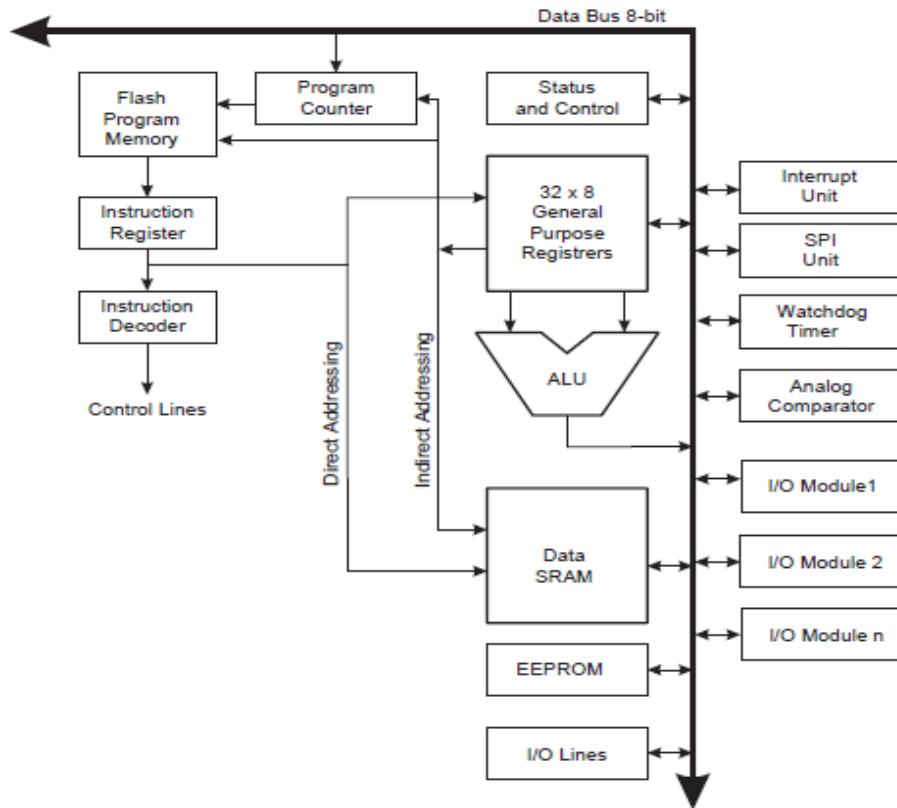


Figura 42: Núcleo de Atmega 328P [Atmo9]

Los conceptos generales de núcleo del procesador Atmega32 que hay que conocer son:

- Arquitectura Harvard.
- Procesador RISC.
- Procesador segmentado.

#### 2.3.2.1.1 Arquitectura Harvard

En la arquitectura HARVARD existen dos espacios de memoria independientes: uno para datos y otro para instrucciones y cada uno de ellos dispone de sus propios buses (figura 42). Se diferencia, por tanto, de la arquitectura clásica de VON NEUMMAN. [Usao6]

La existencia de buses independientes en la arquitectura HARVARD incrementa notablemente el rendimiento del procesador por dos razones:

1º. Se puede acceder simultáneamente al área de datos y a la de instrucciones.

2º. La anchura de las posiciones de memoria del programa se ajusta al formato de las instrucciones y la de la memoria de datos a la de los datos.

#### 2.3.2.1.2 Procesador Risc

La filosofía RISC (Reduced Instruction Set Computer) se orienta a ejecutar las instrucciones en un solo ciclo máquina. En los Atmega, todas las instrucciones se realizan en un ciclo, excepto las de salto y las llamadas a subrutinas que necesitan dos. Las operaciones aritmética-lógicas las realiza la ALU en base a dos operandos, que se ubican en un banco de registros; en el área de la memoria de datos.

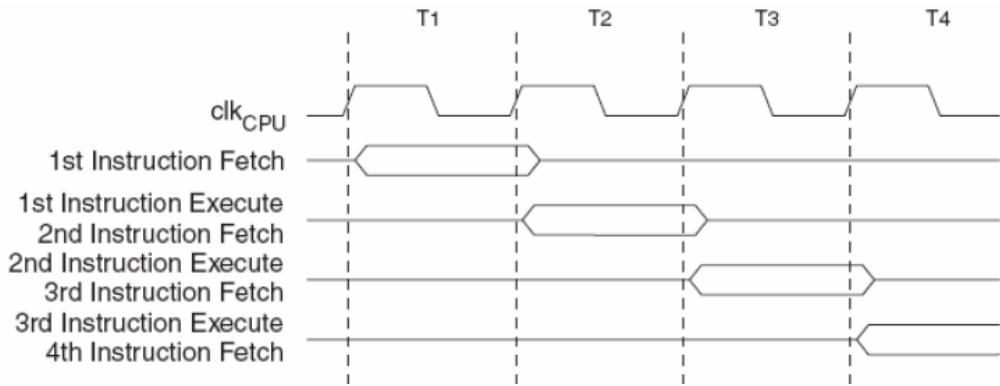
#### 2.3.2.1.3 Procesador segmentado

La segmentación (pipeline) consiste en descomponer la ejecución de cada instrucción en varias etapas para poder empezar a procesar una instrucción diferente en cada una de ellas y trabajar con varias a la vez.

En el procesador Atmega, con una segmentación simple, se ejecutan las siguientes etapas en una instrucción:

- IF (Instruction Fetch): búsqueda
- ID (Instruction Decode): decodificación
- EX (Execution): ejecución

En la figura 43 se advierte como aumenta el rendimiento, ejecutando en un ciclo tres etapas de distintas instrucciones.



**Figura 43: Ejecución en paralelo de instrucciones[Atm09]**

### 2.3.2.2 Puerto de entrada/salida

Todos los puertos del Atmega 328P son GPIO y pueden leerse y escribirse sin modificar al resto de los pines.

Son tres ubicaciones de memoria las que se asignan por cada puerto, una para el Registro de Datos – PORTx, otra para el Registro de Dirección de Datos – DDRx, y otra para los Pines del Puerto de Entrada – PINx. [Atm09]

En el Registro DDRx se selecciona la dirección de ese pin. Si se le escribe un uno lógico, se configura como un pin de salida y se le escribe un cero lógico, se configura como entrada. Si además se le escribe a PORTxn un uno lógico cuando el pin está configurado como un pin de entrada, el resistor pull-up se activa. También se permite configurar el pin en alta impedancia.

### 2.3.2.3 Periféricos usados

#### 2.3.2.3.1 USART

USART (Universal Synchronous Asynchronous Receiver Transmitter) es el periférico que incorporan muchos microcontroladores para comunicarse con dispositivos que soportan el estándar RS-232. De su nombre se deduce que puede trabajar en modo Síncrono o Asíncrono. Esta presentación se enfocará exclusivamente al modo asíncrono como lo demanda el protocolo MIDI.

Las características comunes del USART en los AVR son: [Atm09]

- Full-dúplex; transmitir y recibir datos, al mismo tiempo.

- Permite configurar interrupciones. Tanto al recibir dato, como después de enviarlo. Así las transferencias de datos se lleven a cabo mientras la CPU realiza otras tareas.
- Baud Rate configurable.
- Los datos pueden ser de 5, 6, 7, 8 ó 9 bits. Puede trabajar con uno o dos bits de Stop. Comprobación del bit de paridad.

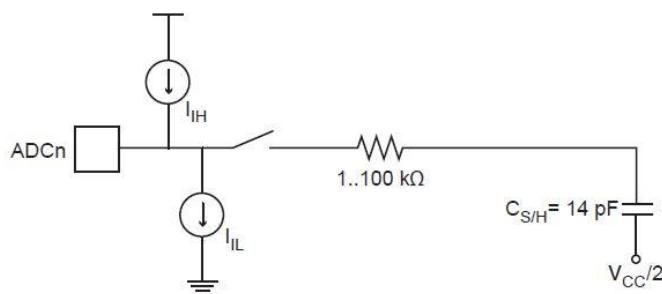
La USART tiene seis registros donde se configura el periférico y se obtiene el resultado de la conversión.

- UDRn: Registro de datos para transmisión y recepción.
- UCSRnA, UCSRnB, UCSRnC: Registros de control y estado.
- UBRRnL y UBRRnH Configuración del Baudrate.

#### 2.3.2.3.2 Módulo convertidor A/D

La mayoría de los fenómenos físicos toman cantidades continuas pudiendo tomar cualquier valor en un rango. Por ejemplo, en el apartado 2.3 se vio que los sensores piezoelectrinos varían su magnitud eléctrica de forma proporcional al valor de presión ejercida; una señal analógica.

Para poder procesar esa variable se necesita discretizarla con un ADC (Analog to Digital Converter) que transforma una tensión de entrada en un código binario.



*Figura 44: Esquema eléctrico entrada ADC*

El ADC integrado en el Atmega328P es un convertidor de aproximaciones sucesivas y tiene una resolución de 10 bit, luego tiene un rango de 0-1023 ( $2^{10} = 1024$ ). [Atmo9] El ADC puede utilizarse en 2 modos conversión: única y libre. En el modo de conversión única, la utilizada en el disparador MIDI, se ordena una conversión, se realiza y el periférico se mantiene a la espera. En el modo libre se recogen muestras de manera sistemática.

El ADC necesita un impulso de reloj para realizar su conversión. Esta fuente de reloj puede ser generada por el reloj del sistema a través de un divisor de frecuencia denominado pre-escalador. Además posee 8 canales cuyo esquema de entrada puede verse en la figura 44.

El ADC tiene cuatro registros donde se configura el periférico y se obtiene el resultado de la conversión.

- ADMUX: Selección de canal de entrada y la tensión de referencia.
- ADCSRA: Control del ADC y registro de estado.
- ADCL y ADCH: Registro de datos; el resultado final de la conversión.

#### 2.3.2.3.3 Temporizadores

El tiempo es una magnitud física fundamental cuya unidad en el SI es el segundo. Gestionar el tiempo en un sistema es esencial para tomar determinadas decisiones.

Utilizar el oscilador externo (reloj del sistema) proporciona una aproximación del tiempo absoluto ya que funcionan de manera continua. Los temporizadores con su característica de arranque, parada y recarga proporcionan medidas temporales relativas entre dos eventos. El reloj de Arduino funciona a 16Mhz aportando una precisión de 63,5nseg.

El oscilador generará eventos periódicos en los flancos de subida de la onda cuadrada. Estos eventos serán acumulados por el temporizador y mediante software los aplicaremos en las unidades de tiempo que se utilicen (milisegundos, minutos...).

El Atmega328P proporciona 3 temporizadores, dos de 8 bits (Timero y Timer2) y 1 de 16 bits (Timer1). Disponen de funciones especiales como captura y comparación de valor además de generación de señales PWM.

Poseen los siguientes registros para su utilización. Siendo n el número del timer. Obviamente las funciones especiales:

- TCCRnA y TCCRnB: Control del Timer/Counter
- TCNTn: Registro Timer/Counter Register
- TIMSKn – Configuración de interrupción
- TIFRn – Borrar la bandera de overflow de la interrupción

## 2.4 El lenguaje Arduino

Arduino como hemos visto en el apartado 2.3.1 es una plataforma de desarrollo. Esos fueron sus comienzos puesto que ahora posee un entorno de desarrollo sólido e incluso una API de programación que muchas personas consideran un lenguaje como tal.

Arduino ha simplificado conceptos. Bajo el lema: “Aprende, comparte y modifica” ha enganchado a muchas personas que de otra manera mirarían con recelo la electrónica digital.

Los diseñadores de Arduino han conseguido abstraer la capa hardware para ser resolutivo con problemas y aplicaciones. Su público objetivo han sido artistas y diseñadores pero como observamos en estas páginas estamos ante un hardware muy preciso y con un gran potencial.

Las librerías están escritas en C++ con una estructura orientada a objetos aunque los programas que se escriben en su IDE lo hacen bajo programación estructurada y en C.

### 2.4.1 Lenguaje C

El lenguaje de programación de propósito general C fue creado en 1972 por Dennis M. Ritchie en los laboratorios Bell. Es uno de los lenguajes de programación más importantes y el que más ha influenciado a los existentes actualmente.

C Es un lenguaje orientado a la implementación de sistemas [Ker91]. Es Utilizado para escribir compiladores y sistemas operativos, aunque también se utiliza para crear aplicaciones. De hecho el primer sistema operativo escrito en un lenguaje distinto a ensamblador fue UNIX que se escribió en este lenguaje. C es apreciado por la eficiencia del código que produce ya que uno de los objetivos de diseño que se planteó en su origen es que sólo fueran necesarias unas pocas instrucciones en lenguaje máquina para traducir cada elemento del lenguaje.

C es un lenguaje fuertemente tipificado de alto nivel, con estructuras típicas de dichos lenguajes (if/else/while...) pero, a su vez, posee elementos que permiten un control a muy bajo nivel de abstracción, como el acceso a memoria mediante el uso de punteros.

### **2.4.1.1      Ventajas de lenguaje C sobre Ensamblador**

Únicamente la programación cuidadosa en lenguaje ensamblador produce un código más rápido. Es necesario poner en una balanza el control total sobre la máquina a costa de un precioso coste en tiempo de desarrollo y un coste de mantenimiento difícil de calcular en algoritmos complejos. El coste de mantenimiento es alto porque el código ensamblador debe estar completamente documentado y asimilado por el servicio técnico. Además en ensamblador el código es totalmente dependiente del hardware y su disponibilidad por el fabricante. También la programación en ensamblador es exclusivamente estructurada.

C permite un desarrollo rápido y, un mantenimiento y mejora, menos costoso. Además de acceso a recursos hardware. Los inconvenientes respecto a ensamblador son despreciables a día de hoy. La diferencia en la optimización del código es insignificante con los avances en los compiladores y la escasez de recursos hardware (memoria, velocidad) es desdeñable con la absoluta mejora de los microprocesadores.

#### **2.4.1.1.1    Aplicaciones embebidas**

Los sistemas embebidos son sistemas de computación de función específica, frecuentemente en tiempo real. El disparador MIDI para batería es un sistema empotrado y C es el lenguaje común para programar estos sistemas.[Jaso8].

Por ejemplo en cualquier microcontrolador donde es necesario una manipulación a nivel de bit para controlar los periféricos (apartado 2.3.2.3). Los sistemas contienen registros mapeados en memoria a través de los cuales se configuran estos periféricos. Estos registros poseen varias configuraciones en la misma dirección de memoria, aunque en bits distintos. Este tipo de manipulación es imposible en otros lenguajes de alto nivel. Por ejemplo en el código de la figura 45.

```
int *pad; /* puntero al registro que queremos modificar */
pad = 0x40008ABC; /* dirección de memoria del registro */
pad |= 1<<7; /* pone a 1 el bit 8 sin modificar los demás */
* pad &= ~(1<<12) /* pone a 0 el bit 13 sin modificar los demás */
```

**Figura 45: Código en el que se accede a registro a nivel de bit**

## 2.4.2 Bootloader

El bootloader es un programa sencillo que está diseñado para preparar todo lo que necesita el programa principal para funcionar.

Para que se entienda mejor, un bootloader es un pequeño conjunto de instrucciones que se graban en un microcontrolador, para permitir un posterior manejo y actualización de sus programas internos (firmware) sin necesidad de utilizar programadores (hardware) específicos como en la figura 46. Es decir, utilizar un programador de microcontroladores una única vez para cargar el mencionado bootloader al microcontrolador en la memoria flash y luego bastará con una sencilla aplicación de ordenador para cambiar el funcionamiento del sistema. Todo mediante una típica conexión al puerto USB.

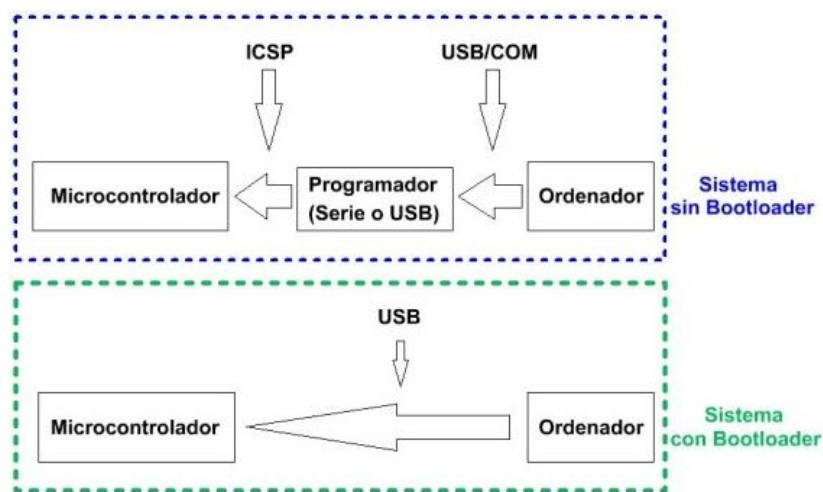


Figura 46: El uso del bootloader [DNeot]

Un ejemplo de las miles de ventajas que puedes tener: Edrumlowcost.com ha vendido a un usuario una aplicación y luego pasado un tiempo de su uso envía un correo expresando algunas deficiencias del producto, y comentando algunas mejoras para lograr un funcionamiento óptimo y acorde a sus necesidades. Ante esa situación: A) El usuario debería enviar el equipo a la empresa. Caro y costoso. B) El programador viaja al lugar del usuario, cambia dos líneas del programa, conecta el programador y resuelve el problema. Gastos de desplazamiento y mal soporte. C) Enviar al usuario un pequeño archivo y unas instrucciones por correo electrónico. Es la opción idónea en la que el dispositivo posee el bootloader y el usuario mismo puede actualizar la versión de firmware bajo su propia demanda.

Así trabajan Arduino, Pinguino, Chipkit y Stellaris. Así de sencillo, en el apartado 2.3.1

### 2.4.3 Instalación entorno Arduino

Comenzar en las placas de la familia de microcontrolador de Arduino; por ejemplo en el modelo UNO de la figura 47. Todo lo necesario para disponer del entorno de desarrollo, los drivers de comunicación y de los ejemplos proporcionados. [Dardu]



Figura 47: Placa de desarrollo Arduino modelo UNO [Dardu]

Seguir estas sencillas reglas.

1. Obtener un Arduino y un cable USB de terminación tipo A hacia tipo B.
2. Descargar la última versión del IDE (mayo del 2014 versión 1.0.5). [Dardu]
3. Conectar el Arduino al ordenador. El led verde de alimentación debe encenderse
4. Instalar los controladores.

Ir a Panel de control>Seguridad y sistema>Sistema>Administrador de dispositivos. Allí en Puertos (COM y LPT) debe aparecer una exclamación con "Arduino UNO (COMxx)" Dar encima con el botón secundario y acceder a "actualizar software de controlador". Explorar hasta encontrar en la carpeta drivers de Arduino el archivo "ArduinoUNO.inf" y esperar su correcta instalación.

5. Iniciar la aplicación de Arduino y abrir el programa de ejemplo blink. File > Examples > 1.Basics > Blink
6. Selecciona la placa en Tools > Board y seleccionar el puerto serie en Tools>Serial Port. Siempre por encima del COM3.
7. Seleccionar en el panel de control de la figura 48 "Upload de program". Observar que se enciende el led azul del pin 13 por lo que todo es correcto y se puede empezar a usar.



Figura 48: Panel de control del entorno de desarrollo de Arduino

#### **2.4.4 Librerías utilizadas de Arduino**

Se enseñan aquí las funciones de las librerías de la plataforma Arduino utilizadas en el código de la aplicación dispuesta en el anexo. Se indica su funcionalidad y los parámetros necesarios para su uso. [Dardu]

- `pinMode(pin, mode)`

Configurar un pin digital como entrada o salida.

pin es el número de pin a configurar [0..13].

mode es INPUT, OUTPUT.

- `val=digitalRead(pin)`

Lectura digital de un pin especificado.

pin es el número de pin digital a leer.

Devuelve: val es HIGH (1), LOW (0).

- `digitalWrite(pin,val)`

Escritura digital de un pin especificado.

pin es el número de pin digital a escribir.

val es HIGH (1), LOW (0).

- `val=analogRead(pin)`

Lectura analógica de un pin especificado.

pin es el número de pin analógico a leer [0—6].

Devuelve: val es un número entero (int) entre 0 y 1023.

Produce una resolución entre las lecturas de: 5 voltios/1024 unidades=4,9 mV por unidad.

El rango de entrada y la resolución se puede cambiar con `analogReference()`.

Se tarda aproximadamente 100 microsegundos en ejecutar esta función..

- `Val=micros()`

Obtener el tiempo en microsegundos que transcurre desde que Arduino comenzó a ejecutar el programa actual.

Devuelve: val es el número de microsegundos(unsigned long).

- `delay(milisec)`

Detiene la ejecución del programa durante un tiempo especificado en milisegundos.

Milisec es el número de milisegundos (unsigned long).

- Serial.begin(speed)

Inicializa la placa para enviar y recibir datos por el puerto serie.

speed es la velocidad de transferencia (típicamente 31250).

- Serial.write(data)

Enviar datos binarios a través del puerto serie.

data es el dato a enviar.



# 3 DESARROLLO DEL PROYECTO

---



## 3.1 Disparador MIDI

### 3.1.1 Descripción

El objetivo consiste en la implementación de un instrumento electrónico al que llamaremos disparador MIDI. Este sistema es capaz de gestionar las percusiones efectuadas en unos sensores que emulan instrumentos de batería. El equipo se encarga de procesar y transmitir dichas señales.

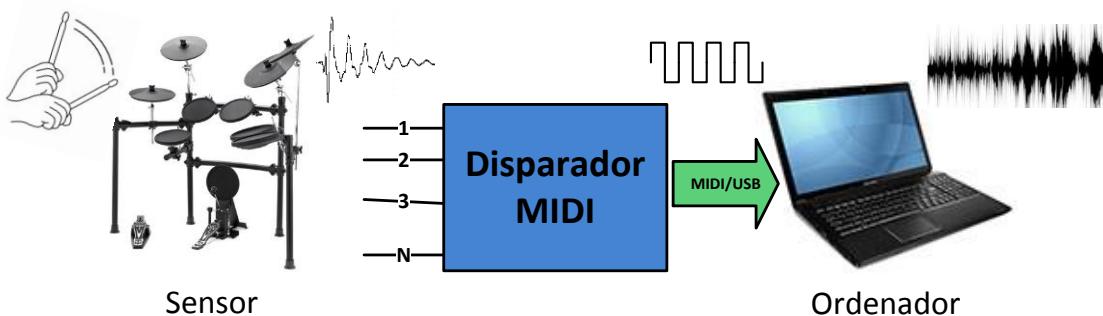


Figura 49: Proyecto disparador MIDI

El sistema de la figura 49 recogerá una serie de señales diferentes generadas por sensores de percusión y, tras efectuar una serie de tareas se comunicará con otros dispositivos. Por ejemplo un sintetizador encargado de la reproducción de sonidos.

Los sensores de percusión consisten en una serie transductores piezoelectrinos (apartado 2.2) colocados adecuadamente en un chasis. Mediante el efecto piezoelectrónico transforman las variaciones de presión aplicadas en ellos en impulsos eléctricos. La amplitud está en función de la fuerza ejercida y la respuesta al impulso será una señal subamortiguada.

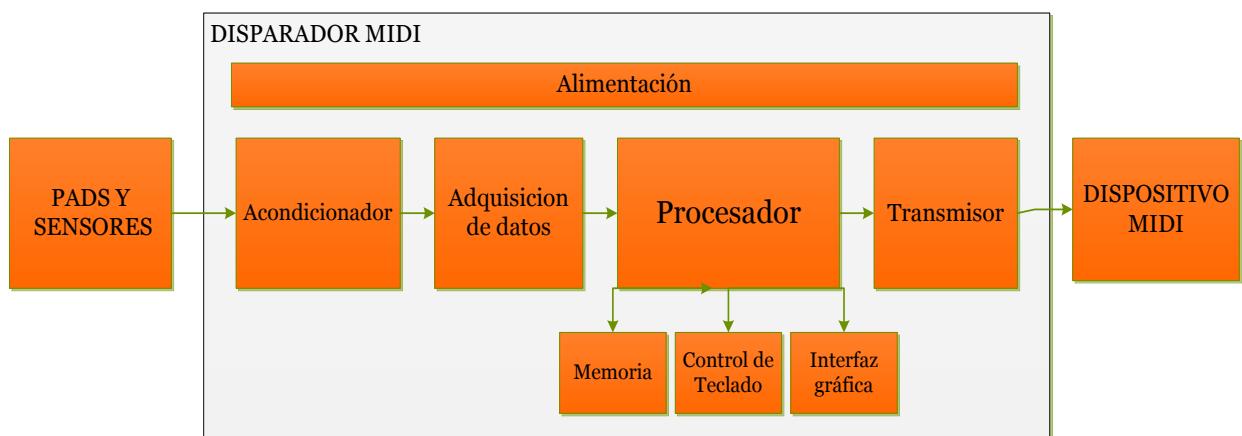
El protocolo de comunicaciones es MIDI (apartado 2.1) que sigue plenamente vigente y que permite conexión directa con infinidad de instrumentos musicales específicos aunque no con el PC; el utilizado por muchos usuarios. Para solucionar el que un ordenador no incorpore conector MIDI-IN será necesario conectar un conversor MIDI-USB ideado para este fin. Este conversor permite una comunicación transparente entre el disparador MIDI y el software de PC.

Finalmente y mediante un software específico, interpreta correctamente los mensajes y, reproduce los sonidos en la propia tarjeta de sonido. Este software de batería basado en

muestreo musical (apartado 1.2.4) además de reproducir el sonido permite al usuario dar un toque personal.

De manera que el proceso de creación de sonidos de percusión consiste en que el controlador ha de detectar la cantidad de golpe aplicada, enviarla y, será el sintetizador el que genere la forma de onda con esta información y los parámetros que tenga almacenados para dicho instrumento.

Por tanto, el disparador MIDI es un sistema de medida (capta los cambios y variaciones de los sensores) que se compone por cuatro bloques principales: bloque de acondicionamiento de señal, bloque de adquisición de datos, bloque de procesamiento de la información y un bloque transmisor. Realizando otras funciones encontramos un bloque de alimentación que suministra la energía necesaria y un bloque de interfaz de usuario formado por tres bloques más pequeños: Memoria, control de teclado e interfaz gráfica. En forma de diagrama quedaría expresado en la figura 50.



*Figura 50: Diagrama de bloques disparador MIDI*

El bloque acondicionador formado por un conjunto de componentes electrónicos discretos se encarga de entregar a su salida la envolvente de la señal que dan los sensores; un valor más estabilizado así como limitar su nivel de tensión. Permitiendo aprovechar mejor el rango de trabajo del siguiente bloque (conversor A/D) y también la cantidad de muestras necesarias. Además suprimirá componentes frecuenciales innecesarias mediante un filtrado.

El bloque convertidor A/D transforma las señales analógicas en digitales para poder ser tratadas por el bloque procesador. La señal es muestreada, cuantificada y al final codificada.

El bloque de tratamiento de la información, o bloque microprocesador, gestiona las órdenes de conversión A/D, los códigos recibidos, el número de canal del sensor que corresponda. En función de dichos parámetros y otros introducidos por el usuario genera bytes de mensajes MIDI correspondientes a dichos eventos. Aparte realiza las esperas necesarias y, controla teclado y pantalla.

El último bloque principal es el transmisor serie asíncrono cuya función es enviar la información a la velocidad y formato indicada en la especificación.

Estas tres últimas etapas comentadas (bloque A/D, microprocesador y transmisor serie) están implementadas en un solo circuito integrado comercial, el microcontrolador, que permite compactar y simplificar el circuito final. El microcontrolador está dispuesto, junto a todos los componentes que facilitan su funcionamiento, en una placa de desarrollo (apartado 2.3.1).

Por otro lado se habilita la gestión de la interfaz del usuario. Se dispone de un control de teclado donde se pueden seleccionar una serie de opciones y parámetros aparecidos en una pantalla (interfaz gráfica), pudiendo modificar y adaptar sus preferencias puesto que se almacenan en memoria no volátil.

Por último el prototipo construido, con sus componentes electrónicos, se monta sobre placa de circuito impreso y se ubica en una carcasa. En esta carcasa se colocan los conectores y sus respectivas conexiones así como los módulos aquí descritos.

### **3.1.2 Especificaciones**

Las especificaciones que cumplirá el prototipo son:

- Disparador MIDI para batería. Comunicación estándar MIDI e integrable con programas de síntesis de baterías.
- Conectores Jack de entrada para los sensores y conector DIN-5 de salida.
- Recoger 6 señales de 6 sensores piezoeléctricos. Todos ellos sensibles, captando la energía de un percusionista. Ampliable a 12 sensores. Ser capaz de detectar y cubrir cualquier pad de cualquier marca. Una mejora respecto a sus competidores comerciales (Roland, Yamaha...). Por ejemplo la marca Roland en sus módulos solo detecta los pads de Roland.
- Permitir sin pérdida de información, una interpretación en notas fusa de un ritmo de negra igual a 60 bpm. Además de una polifonía de los 4 miembros del cuerpo.
- Combinar 5 tipos de parámetros que mejoren la experiencia del percusionista.
- Interfaz de usuario original y con opciones fácilmente articulables. Panel de control fácil e intuitivo pudiéndose almacenar las preferencias del usuario en una memoria no volátil.
- Alimentación a través de la red eléctrica y también a través de USB.
- Programa simple y eficiente. Número de sentencias reducidas y con un modelo de ejecución concurrente.
- Controlador de pedal. Disponer de movimiento de apertura y cierre. Además del cambio de sonido producido, por ese movimiento, en otro pad.

## 3.2 Estructura Mecánica

### 3.2.1 Definición de pad

Los pads son las zonas de golpeo (aparatado 1.2.3.1). En ellos se deben cumplir una serie de requisitos para su construcción.

- Proporcionar sostén y protección a la electrónica que llevan dentro; el disco piezoeléctrico.
- Ser cómodos y agradables al tacto del músico. En este caso la tendencia de las marcas que desarrollan este tipo de productos ha sido la de cuidar el rebote asemejándolas a una acústica. También deben estar al alcance del ejecutante para que puedan ser golpeados sin dificultad.
- Tamaño y forma. Son pequeños, con materiales ligeros y estéticamente de forma circular (Ver figura 40).
- Evitar vibraciones y ruidos acústicos. Aíslan vibraciones que se transmiten por la estructura y disminuyen ruidos acústicos que desvirtúen los generados en el módulo.

Como dato, al hablar de parches y medidas en la batería se usan las pulgadas. Por ejemplo los pads suelen ser de 10" o 12", unos 15 o 18 cm. Es uno de los resultados de la gran influencia de Estados Unidos en la difusión de la música moderna.

### **3.2.2 Análisis de Pads Comerciales**

El mercado de segunda mano en los instrumentos musicales es muy grande. Comprar instrumentos nuevos no es económico, se rompen o incluso las personas que los tocan los pueden cambiar por razones personales.

En sitios como mundoanuncio.com convergen multitud de ofertas variadas y en un sitio especializado como batacas.com se ofrece todo lo que se necesita del mundo de la batería.

En dichos lugares se adquirieron los siguientes productos que se analizan a continuación:

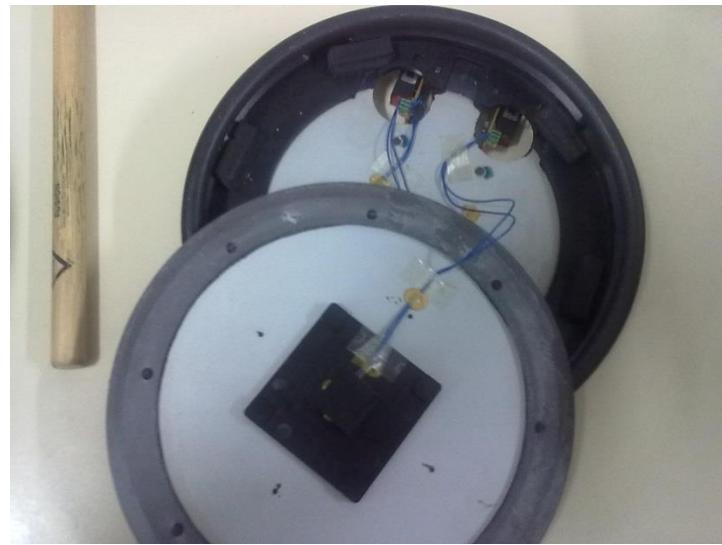
- 2 pad Yamaha ep-75.
- 1 pad de plato Yamaha modelo PCY65.

Los dos han sido adquiridos de la marca Yamaha debido a recomendaciones de expertos que alegaban la innovación y durabilidad de esta empresa japonesa.

#### **3.2.2.1 Pads Yamaha ep-75**

El pad Ep-75 es un pad de caja estéreo compatible con los módulos Yamaha. Se coloca en un rack y se ubica suspendido en el aire.

Los pads son de goma de 10 pulgadas de diámetro. La zona del aro viene reforzada con un plástico más consistente. El piezoeléctrico se sitúa tal y como viene en la imagen 51 resguardada por una gomaespuma. Los cables son pequeños de sección de 0,75mm<sup>2</sup>. Estos se encuentran sujetos al chasis tanto por pasta como por cinta adhesiva. Los jack son de 1/4 y están colocados en una PCB junto a componentes pasivos (resistencias). El sistema en su conjunto es compacto y amortigua las vibraciones exteriores.



*Figura 51: Interior del pad comercial modelo EP75*

Atendiendo a la figura 52 posee 3 piezoelectrinos de 35mm de diámetro. El principal ubicado en el centro está adherido a una plancha metálica para recoger las vibraciones. Y luego los otros dos en los bordes del aro para recoger el golpe lateral. Las señales van por dos caminos diferentes a través de dos conectores Jack mono marcados.



*Figura 52: Disposición de los piezoelectrinos del EP75*

#### 3.2.2.1.1 Análisis eléctrico

Con el pad desmontado (figura 51) se recoge el esquema eléctrico proveniente de los cables y la PCB. Para este propósito se utiliza un multímetro en modalidad testeador.

La figura 53 recoge los componentes (resistencias y piezoelectrónico) así como el valor de estos. Como se ha comentado son dos circuitos distintos. El primero corresponde al circuito del centro del pad y el segundo al del aro.



**Figura 53:** Esquema de conexiones modelo EP75

Ahora se ha cogido dicho pad y se golpea recogiendo la señal proveniente del piezoelectrónico central con un osciloscopio (modelo Agilent Technologies DSO 5014). La respuesta ha sido la de la figura 57 para un golpe de mediana intensidad producido en el centro del pad. Una señal subamortiguada, con muchas componentes frecuenciales con un máximo en 4V.



**Figura 54:** Forma de onda osciloscopio modelo EP75

### 3.2.2.2 Pad Yamaha PCY65

El PCY65 (figura 55) es un pad de plato mono compatible con módulos de Yamaha. Puede montarse en cualquier soporte de plato convencional. Es de  $\frac{1}{4}$  de sección de una pieza circular de 10”.

Está construido en plástico y reforzado en la superficie de golpeo con goma.



*Figura 55: Pad de plato PCY65*

El modelo PCY65S tiene función Choke añadiendo al PCY65 un switch formado por dos piezas metálicas separadas. [Dyama] Esta aseveración fue comprobada al observar que el circuito impreso viene preparado para esta opción pero que no era utilizada.

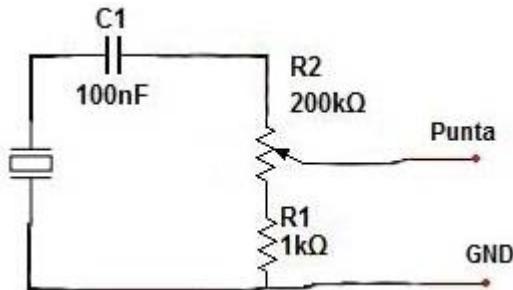
Se denomina Choke a la parada de plato y, se utiliza para silenciar el sonido de largo sostenimiento de un plato. Se aprieta con la mano, en la superficie del plato, emulando apagar las vibraciones.



*Figura 56: Interior del pad de plato PCY65*

En la figura 56 aparece el piezoeléctrico. Es de un tamaño de 27mm de diámetro y posee una resina y un trozo de cinta aislante negro que lo sujeta y proporciona amortiguación. El pegamento con el que está adherido es de base de silicona.

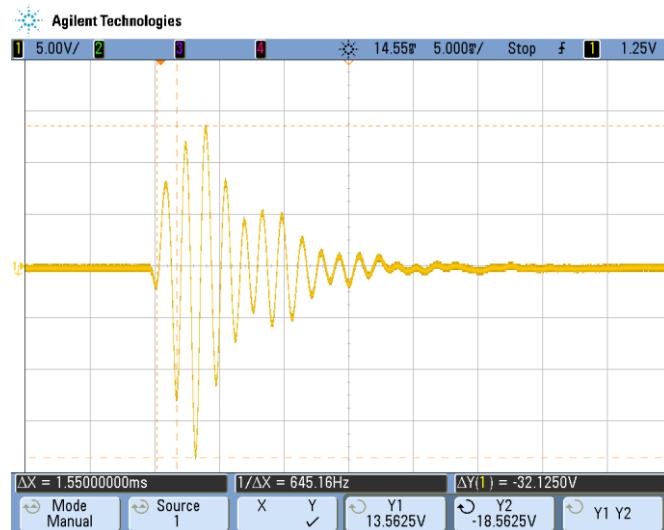
### 3.2.2.2.1 Análisis eléctrico



*Figura 57: Esquema de conexiones plato PCY65*

En el esquema eléctrico de la figura 57 aparece un condensador que actuará de filtro paso alto y un potenciómetro para regular el nivel de tensión.

En la figura 58 se ve una señal sinusoidal subamortiguada de frecuencia 645Hz y un valor máximo de 13,6V producida por un golpe de fuerte intensidad.



*Figura 58: Forma onda osciloscopio*

### 3.2.3 Diseño y construcción de Pads

#### 3.2.3.1 Compra de materiales

Atendiendo a los requisitos del apartado 3.2.1 los materiales pueden ser diversos (plásticos, maderas, metales...) por lo que es necesario ser más exhaustivos. Para ello se realizó un estudio cualitativo de materiales.

Pero la primera cuestión que una persona no habitual al bricolaje debe responder es el lugar donde se adquieren todos esos diversos materiales

En primera instancia parece una pregunta simple e incluso tonta pues desde la niñez la tienes asimilada. Metales en ferretería, madera en carpintería.... Aunque esto no es así cuando necesitas un nivel de especialización óptimo. Ej.) Plancha de acero ionizado de 0.5mm.

A esto se añade que los buscadores como Google o Yahoo no están optimizados para estas búsquedas y saldrán tiendas. En las bibliotecas aún es más difícil al tratarse de conocimientos basados en experiencias particulares. La única alternativa son familiares, amigos y conocidos, reduciendo las posibilidades al boca a boca y a la experiencia que tuvieran estas personas en su momento. Por tanto la búsqueda de información se basa en la persecución de un rumor que te lleve a otro y esperar a encontrar algo.

Añadida la tabla donde se puede observar donde se produjo la compra de materiales, el nombre de la tienda y la ubicación.

| Tienda                | Descripción            | Compra  | Emplazamiento         |
|-----------------------|------------------------|---|-----------------------|
| <b>Madrid Musical</b> | Instrumentos musicales | Pad de prácticas y pad sordo de plato         | C/ Málaga 8           |
| <b>Leroy Merlin</b>   | Bricolaje              | Acero, aluminio , contrachapado , tornillería | Parque Oeste Alcorcón |
| <b>Resopal</b>        | Plásticos              | Duralar                                       | C/ Cronos, 14         |
| <b>Conectrol</b>      | Electrónica            | Componentes electrónicos                      | C/ Jorge Juan 57      |
| <b>RS</b>             | Electrónica            | Sensores piezoeléctricos                      | Avenida de Europa, 19 |
| <b>Bellas Artes</b>   | Manualidades           | Goma Eva                                      | C/ Pelayo, 38         |
| <b>Jeco</b>           |                        |   |                       |
| <b>Corte Espuma</b>   | Goma Espuma            | Goma a medida                                 | C/ Carlos Arniches 20 |

Tabla 9: Lista de materiales y lugar de su compra

### **3.2.3.2 Modificación pad de prácticas**

En primer lugar se pensó en colocar el sensor piezoelectrónico en una zona utilizada habitualmente por el baterista acústico; que éste se sintiera cómodo y así comentar sus sensaciones y opiniones.

Pero colocar los sensores analizados en el apartado 2.2 (sensores cerámicos piezoelectrónico) es tedioso porque se deben adherir al elemento de la batería, colocar unos conos de gomaespuma que rocen el parche, etc. Significa, en gran medida, modificar la batería acústica con taladros y pegamento para sujetar.

La solución parcial reside en comprar una pieza y modificarla; el pad de prácticas de la marca Remo (figura 61).

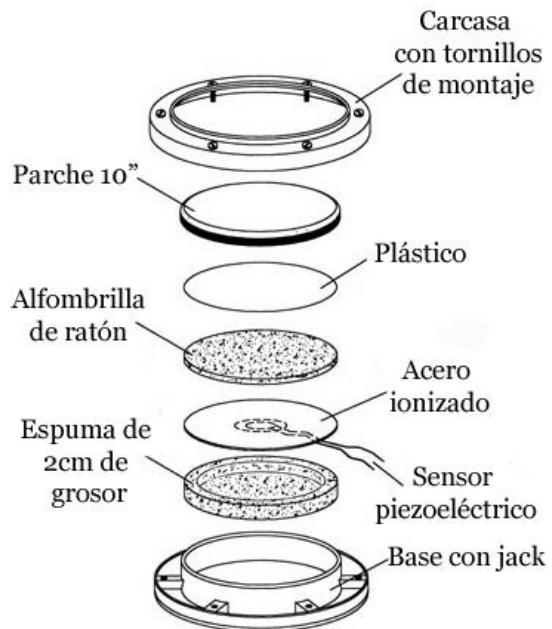
El pad de prácticas se utiliza para ejercitarse las manos con las baquetas y estudiar rudimentos. Los rudimentos son pequeños ritmos y combinaciones que el baterista practica para mejorar. Son 40 y van desde golpes simples alternando manos hasta el Ratamacue del compás de la figura 59 que posee tresillos, drags y golpes dobles.



*Figura 59: Pentagrama del rudimento Single Ratamacue [Stao5]*

Volviendo al tema de la colocación del sensor en el pad. Este se desmonta y se observa el interior. Un parche de Mylar de 10" (15 cm), gomaespuma de 2 cm de espesor y la caja que hace de receptáculo. Parte de ello se puede ver en el dibujo de la figura 60.

Por otro lado se prepara el sensor. Se corta la plancha de acero ionizado a la misma medida que la espuma con unas tijeras de hojalatero. Se pega el disco cerámico en el centro con un pegamento especial de vibraciones, como el adhesivo Ceys Triaction. Existían otras opciones al acero inoxidable pero se decantó por el factor del precio. El acero inoxidable era más barato que el acero bruto o el aluminio.



*Figura 60: Despiece y modificación pad de prácticas remo [Dhell]*

Se sueldan unos cables a los que ya dispone el sensor y se pasan por un agujero hecho en el plástico de la carcasa. Por último la ubicación de las diferentes capas es la de la figura 60; colocando encima del sensor una alfombrilla de ratón que atenúa los impactos.

El resultado es el de la figura 61, que está colocado sobre su soporte.



*Figura 61: Resultado final pad remo*

Respecto a los materiales, en la tabla 10. El coste ha sido excesivo, alrededor de 40 €. Respecto al trabajo, este pad contiene un componente manual moderado.

| Materiales                            | Costes          |
|---------------------------------------|-----------------|
| Pad de prácticas remo                 | 22 €            |
| Acero 50X25 de espesor 1mm            | 15,55 €         |
| Plástico de 48X42                     | 1,50 €          |
| Conector Jack                         | 0,35 €          |
| Cable de cobre de 0.75mm <sup>2</sup> | 1 €             |
| Sensor piezoeléctrico                 | 1,50 €          |
| Alfombrilla de ratón                  | Reciclada -- 0€ |
| Trabajo                               | Horas           |
| Taladrar/Cortar Plancha y alfombrilla | 2 h             |
| Colocación del sensor y cables        | 2 h             |

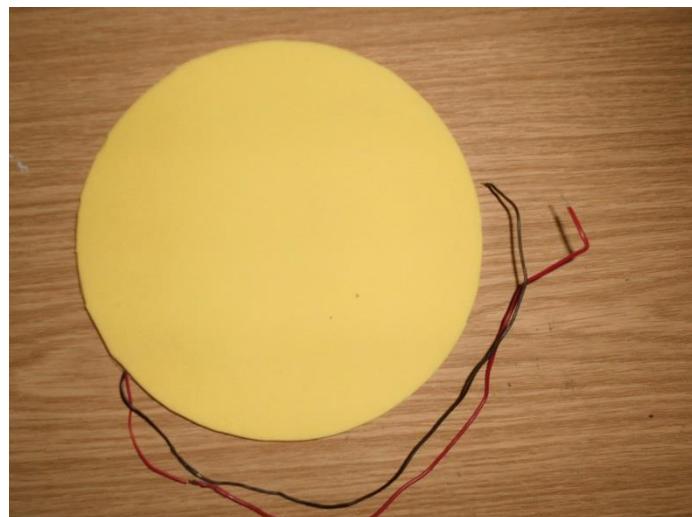
*Tabla 10: Costes y materiales utilizados en pad de prácticas Remo*

### 3.2.3.3 Construcción de pad con Goma EVA

En segundo lugar se pensó en algo fácil de construir, duplicar para hacer en cantidad y que estéticamente fuera agradable. Además el coste de materiales tenía que ser bajo. Los materiales de manualidades fueron una buena opción y se eligió:

Goma EVA, fácil de cortar y con un tacto agradable. También un plástico similar al Mylar utilizado en los parches de una batería acústica que se llama Duralar, se pensó que de esta manera debería transmitir mejor las vibraciones.

El proceso es simple. Se pega el sensor en el centro del Duralar y se envuelve en goma EVA. Los materiales fueron recortados en forma de círculos como puede verse en la figura 62.



*Figura 62: Construcción de Pad con goma EVA*

Fue un absoluto fracaso, dado que el plástico no es un material que transmita la vibración, como todos los metales, sino que atenúa al colocarlo sobre una superficie como una mesa. La conclusión es que si no se presionaba directamente en el sensor (en el centro) no generaba señal. El desgaste y su rotura eran cuestión de tiempo.

Si se encontrara una superficie vibrante, de no demasiado peso y grosor, podría utilizarse para el disfrute de los niños puesto que la goma EVA es blanda y podría elaborarse en distintos colores.

Por último los cables de conexión no eran adecuados. Eran unifilares de sección 0.25mm<sup>2</sup>, utilizados en conexiones de protoboard, lo que les confiaba poca flexibilidad.

| <b>Materiales</b>          | <b>Costes</b> |
|----------------------------|---------------|
| Goma Eva Din-A3            | 4 €           |
| Duralar                    | 2 €           |
| Sensor piezoelectrónico    | 1,50 €        |
| <b>Trabajo</b>             | <b>Horas</b>  |
| Cortar plástico y goma eva | 0.5 h         |
| Armar las piezas           | 0.5 h         |

*Tabla 11: Costes y materiales utilizados en pad de goma EVA*

### 3.2.3.4 Construcción de pad con contrachapado

Esta incursión en el mundo de construcción de pads fue un híbrido entre los dos experimentos anteriores. Por un lado asemejarse mucho a la idea de la batería acústica y su rebote. Por otro fabricar con materiales fáciles de encontrar y trabajar.

Para realizar la estructura se utiliza contrachapado, un material ligero, barato y fácil de cortar. Con una sierra de calar se realizan las formas circulares y con un taladro los agujeros.

Entre medias de las dos tablas se introduce la espuma, la plancha metálica con el sensor y el parche. El parche es un Remo Encore con tacto rugoso como el utilizado habitualmente en las cajas acústicas. Todo ello se cierra con tuercas (figura 63).

Las arandelas se utilizan para no estropear la estructura al apretar las tuercas y los tornillos con la intensidad necesaria para que el parche tenga consistencia.

Para que se vea el interior se realizó una foto (figura 64). El resultado es aceptable, pues el tacto y el rebote de la baqueta están conseguidos. Pero el coste de materiales es mode-

rado y el trabajo a dedicar es alto si se necesitan hacer en cantidad. Pensar por ejemplo en seis pads.

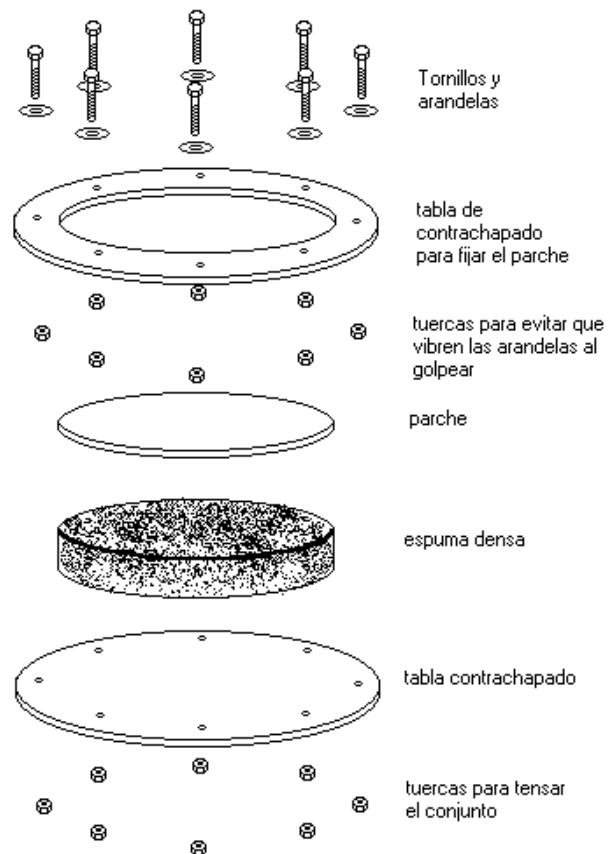


Figura 63: Despiece y construcción de pad completo [Tomo3]



Figura 64: Fotografía interior pad contrachapado

| Materiales                   | Costes |
|------------------------------|--------|
| Parche Encore                | 11 €   |
| Plancha de metal             | 10 €   |
| Contrachapado                | 12 €   |
| Gomaespuma                   | 7 €    |
| Tornillos y arandelas        | 3 €    |
| Trabajo                      | Horas  |
| Serrar/Taladrar              | 3 h    |
| Cortar plancha y gomaespuma  | 2 h    |
| Colocacion del sensor/Cables | 1 h    |

*Tabla 12: Costes y materiales utilizados en pad de contrachapado*

### 3.2.3.5 Construcción de pad de plato

La última aventura en el mundo del bricolaje fue la elaboración de un pad de plato mono. Para ello se compró lo que se denomina un plato sordo en una tienda de instrumentos. Es de plástico rígido con estrías y de color negro ofreciendo una sensación estética muy similar a un plato de bronce.

Una vez dispuesto se pega el piezoelectrónico en la parte superior de la zona que va a ser golpeada y para protegerlo se solapa una goma gruesa como la utilizada en algunas almohadillas de ratón.

Con anterioridad se realizó un taladro para pasar los cables del sensor y se suelda un conector Jack hembra de chasis. Este conector también se adhiere con cola de contacto para finalizar con el acabado de la figura 65.



*Figura 65: Fotografía resultado construcción pad de plato*

El coste de materiales y el trabajo realizado queda explicado en la tabla 13.

| Materiales                    | Costes          |
|-------------------------------|-----------------|
| Plato sordo                   | 9 €             |
| Piezoeléctrico                | 1,5€            |
| Jack hembra aéreo             | 0,40 €          |
| Alfombrilla de ratón          | Reciclada-- o € |
| Trabajo                       | Horas           |
| Taladrar y cortar alfombrilla | 0.5 h           |
| Colocacion del sensor/Cables  | 2 h             |

*Tabla 13: Costes y materiales utilizados en el pad de plato*

### 3.2.4 Análisis del equipamiento completo

En las anteriores construcciones del apartado 3.2.3 se acompaña una tabla con el coste de materiales y de trabajo. Los recursos que serían necesarios aportar son muy altos al tener que construir gran cantidad de pads por lo que se concluye que no es viable construir un kit completo a partir de medios artesanales. También se descarta la opción de comprar un set de pads en segunda mano pues los dos pads EP65 se comprueba que son totalmente distintos en sensibilidad.

También y, según el apartado 2.2 el tiempo degrada el plástico aunque estuvieran bien tratados pues eran pads de 10 años.[Dyama].

En el apartado 3.1 se requería una serie de pads y demostrar la potencia del disparador MIDI. Es necesario adquirir mediante compra un kit completo para estudiarlo. La elección se orientó hacia la opción más económica, con una buena estructura, con numerosos pads y diversas zonas de golpeo. Pero sobre todo la experiencia de un foro profesional como batacas.com. [Dbat] Por ello la adquisición, fue la batería DD508 de la tienda de instrumentos alemana Thoman con un coste de 530€ (figura 66). [Dthom] La elección más clara para una persona con una afición estable del instrumento.

En el estudio de mercado realizado en el apartado 1.2.3.4 y su correspondiente tabla (tabla 2) se compara con otros modelos, quedando por encima de sus competidores.

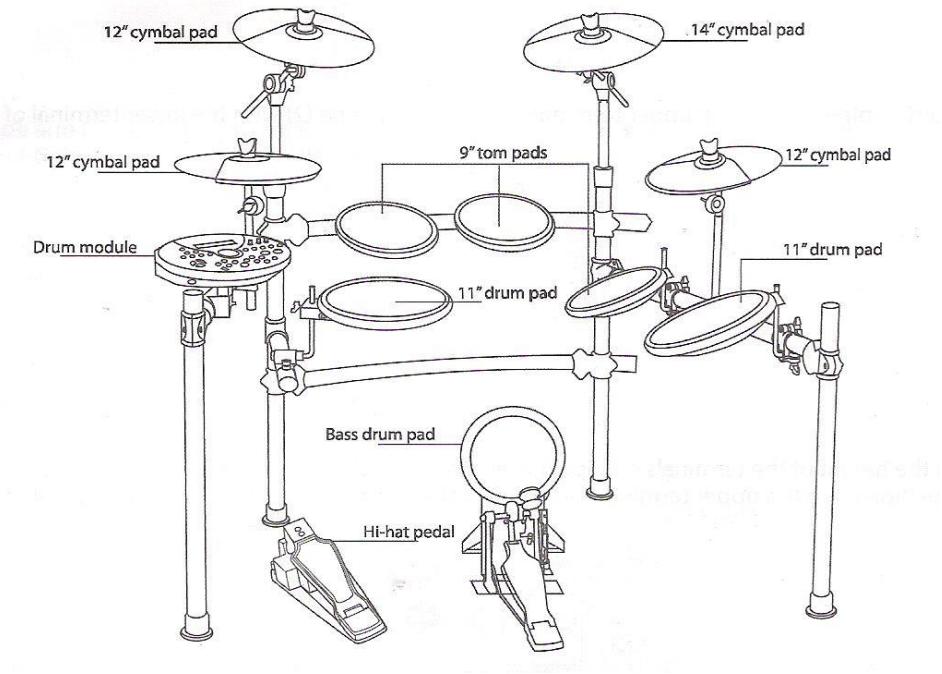


Figura 66: Dibujo batería electrónica modelo DD508 [Med12]

A partir de este momento este será el kit a controlar asimilando muchas ideas de su diseño. Resumiendo sus características: [Dthom]

- 1 módulo. Será el aparato que se tome de referencia para sustituirlo.
- 2x 11" de caja estéreo. Superficie de goma con sensación agradable al toque. Es estéreo porque envía dos señales a través del cable. Una señal para un golpe en el centro y otra para el golpe que se produzca en el borde.
- 3x 9" Pads Tom también estéreo.
- 1x 9" Pad de bombo. Con una estructura que posibilita colocar el pedal de bombo. Es lo suficientemente amplio para que entre un doble pedal.
- 3x 12" Pads de plato estéreo. Tienen 2 zonas sensibles (superficie y borde) y con función de atenuación (Choke).
- 1x 14" Pad de plato estéreo.
- Incluye el controlador Hi-Hat con posibilidad de golpear a media abertura, por lo tanto dispone de 3 zonas (apartado 3.7).
- Drumrack completo de un material consistente, estable y ligero. Similar a la fibra de carbono.
- Una manguera de cables con terminación Jack que facilitan su uso. Los cables son de diferentes longitudes de acuerdo a la situación de cada elemento.
- Herrajes y soportes para su enganche, movilidad y ajuste dentro del espacio.

### **3.3 Análisis de la respuesta de los sensores del kit DD-508**

En el capítulo 2.2 se ha realizado un estudio exhaustivo de conceptos relacionados con los sensores piezoelectríficos como sensores que proporcionan información sobre magnitudes relacionadas con la fuerza. También se concluyó la utilización de cerámicas y que dichos elementos no proporcionan unas características particulares ya que no disponen de datasheet.

Resulta un requisito indispensable analizar las respuestas que posteriormente vamos a procesar.

#### **3.3.1 Caracterizar piezoelectrífico**

Se ha comentado en el capítulo 2.2 que el hecho de colocar el piezoelectrífico en distintas superficies modifica la respuesta que obtenemos de dicho sensor. Como consecuencia el diseño de pads, a este nivel, se hace impensable así como dificulta un acondicionamiento universal.

Por consiguiente se hace necesario un estudio pormenorizado de estos para poder controlarlos. Las herramientas que se deberían de utilizar pasan por un sistema automático de medida (dispositivo específico) que aplicará en las mismas condiciones un estímulo de fuerza así como digitalizaría sus resultados. El pad se convertiría en un DUT (Device Under Test). Como dicho estudio sale de los propósitos de este proyecto se optará por una breve previsión dado las herramientas con las que se cuenta.

El estudio se realizará sobre casos extremos que podrían aparecer. Un golpe de una gran fuerza, utilizando una técnica de gran intensidad para la caja (técnica Moeller). Y otro de intensidad débil como por ejemplo los utilizados en los golpes de relleno (ghost note).

Se utilizará un osciloscopio Rigol DS1052E para captar la respuesta subamortiguada como respuesta a un golpe ofrecida por una baqueta de tamaño 5A y punta de nylon. Los pads corresponderán a partes específicas del sistema a controlar, el kit DD508 que se ha adquirido para este proyecto.

En el análisis se procederá a realizar los experimentos sobre pad de caja de 9", pad de plato de 12" y el bombo de 9". Para poseer un modelo más real el bombo será percutido con un pedal con la maza de plástico.

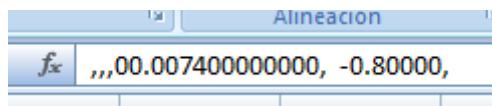
Para que no exista una excesiva ambigüedad y abstracción en el documento se fotografió cada pad estudiado con una vista frontal y otra lateral; mejorando la visión espacial.

A parte se desmontaron como en “Análisis de pads comerciales” (apartado 3.2.2) fotografiando su interior para visualizar las posiciones relativas de los sensores, su composición, etc. También desentrañando los componentes eléctricos integrados y colocando su esquema.

Entonces para caracterizar los sensores piezoeléctricos se tomaran dos posibles ensayos de cada pad. Una respuesta débil y otra fuerte.

Esta respuesta gráfica del osciloscopio se almacenará mediante USB en una memoria flash en dos formatos: .jpg para visualizarla y .csv para procesarla en la CPU pues son los datos temporales y de amplitud. Nótese que .csv (comma-separated values) es un formato administrado por Excel del paquete Office y del Bloc de Notas de Windows.

El procesamiento correrá a cargo de Matlab y Multisim no sin antes modificarle el formato. En Matlab se analizará la forma de onda y su respuesta en frecuencia. Con Multisim se introducirá dicho fichero en las pruebas como estímulo de los acondicionadores propuestos. Esto último corresponde al capítulo 3.4. El procesamiento de Matlab de algunos pad se encuentra en anexos.



*Figura 67: Dato en formato .csv*

En la figura 70 hay un dato obtenido del osciloscopio. Informa de que la onda en el instante de 7,4ms (eje de abscisas) poseía una tensión de -0,8V (eje de ordenadas).

Para cambiar el formato se abre el fichero con el Bloc de notas de Microsoft. Con el comando remplazar se quitarán todas las tabulaciones del principio y el espacio limitador entre el dato x, y el dato y, será el carácter espacio, para ser leído por Matlab. En Multisim sustituir los puntos del decimal por comas.

### **3.3.1.1      Análisis osciloscopio**

En este apartado se explicará la forma de onda que producirá el piezoeléctrico al ser golpeado el pad. Se mostrará una imagen del osciloscopio así como los datos más importantes a tener en cuenta. Todo partiendo del experimento del anterior apartado.

Las medidas que se han considerado características de la señal y que se mostraran en tablas son:

- Valor máximo: Tensión del máximo pico de la señal
- Valor mínimo
- Valor pico a pico
- Valor medio: Valor de continua. Aproximadamente 0V
- Periodo y frecuencia de la señal sinusoidal
- Tiempo de pico. Desde que se da el golpe hasta que se alcanza el máximo
- Tiempo de subida: Tiempo desde su paso por 0V hasta que alcanza el máximo.
- Tiempo de bajada Es el tiempo de estabilización. El tiempo que transcurre hasta que alcanza el 5% del valor final.

Del análisis de la respuesta de los sensores del apartado anterior se concluyó que nuestra entrada era un impulso provocada por el golpe de la baqueta. Y que los sensores piezoelectríficos modulaban ese impulso con las oscilaciones del piezoelectrífico realizando la señal eléctrica.

Se producirá una superposición de señales complejas. Pero es distinguible una sinusoidal y una exponencial. Por consiguiente en la respuesta temporal, aparte de muchas otras señales superpuestas aparecen dos principales:

Señal de carga. Se requiere la tensión máxima y la frecuencia de la señal oscilante.

$$V(t) = Vm * \sin(2 * \pi * f * t)$$

Señal de descarga: Se requiere la tensión máxima y la constante de descarga.

$$V(t) = Vm * e^{\frac{-tmax}{\tau}}$$

### 3.3.2 Pad de caja



Figura 68: Vista frontal y lateral pad de caja estéreo



Figura 69: Vista interior pad caja

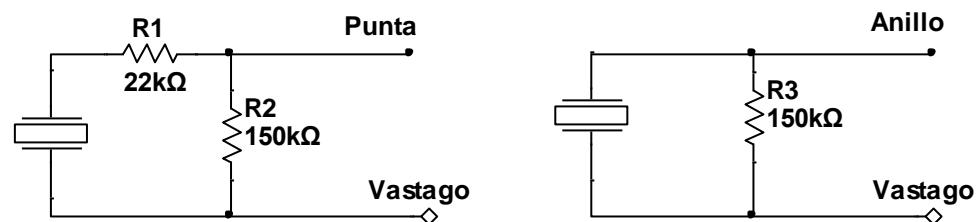
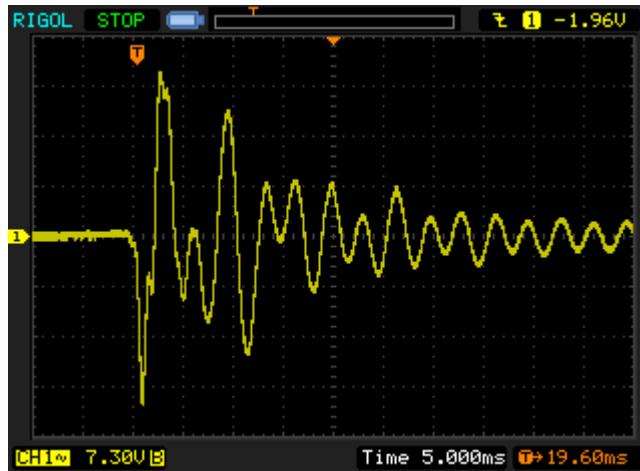
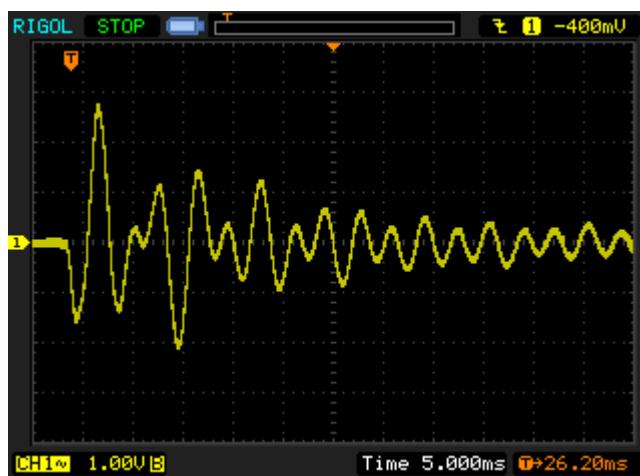


Figura 70: Esquema eléctrico pad de caja



*Figura 71: Señal eléctrica producida por un golpe de fuerte intensidad en pad de caja*

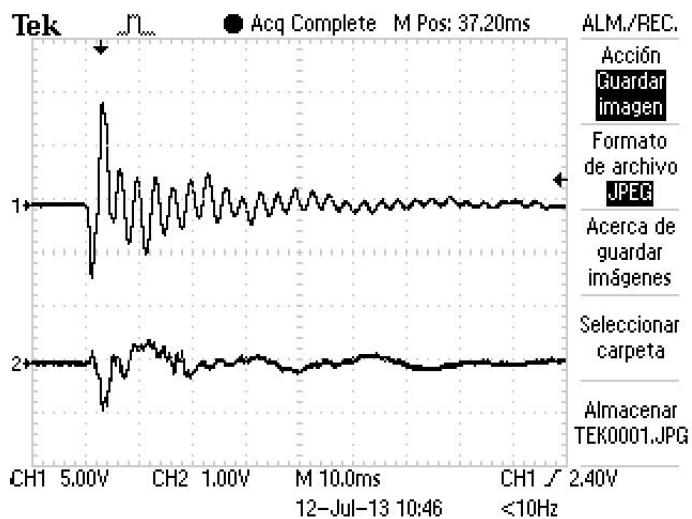


*Figura 72: Señal eléctrica producida por un golpe de intensidad débil en pad de caja*

|                                 | Fuerte      | Débil       |
|---------------------------------|-------------|-------------|
| <b>Valor máximo</b>             | 23,9V       | 2,76V       |
| <b>Valor medio</b>              | 56mV        | -8mV        |
| <b>Valor mínimo</b>             | -24,5V      | 2,12V       |
| <b>Valor pico a pico</b>        | 48,5V       | 4,88V       |
| <b>Periodo/Frecuencia</b>       | 6,6ms/152Hz | 6,6ms/152Hz |
| <b>Tpico</b>                    | 2,6ms       | 3,2ms       |
| <b>Tiempo de subida</b>         | 0,7ms       | 1ms         |
| <b>Tiempo de estabilización</b> | 62ms        | 53ms        |

*Tabla 14: Medidas pad de caja*

A parte del experimento explicado, con el osciloscopio se analizó si existía diafonía. Para ello utilizando los dos canales del osciloscopio se ejecuta un golpe fuerte en el centro pad (figura 73).



*Figura 73: Señal producida en el conector estéreo*

La única diferencia, a parte del nivel de señal, es el tiempo en el que aparece la vibración. Se produce un retardo, de unos 2ms, desde que se produce la señal en el sensor principal hasta que es captado por el sensor de aro.

Como conclusión de la gráfica es que no se utilizan pads estéreos. Si se introducen estas señales en el algoritmo de procesamiento, no se descarta y se envían 2 señales cuando solo se ha realizado una.

### 3.3.3 Pad de plato



*Figura 74: Vista frontal y lateral pad plato*



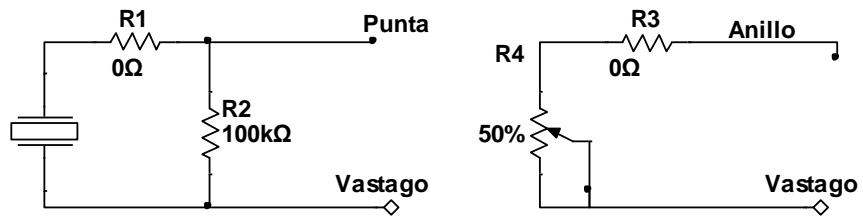
Figura 75: Vista interna pad de caja



Figura 76: Señal eléctrica producida por un golpe de intensidad débil en pad de plato



Figura 77: Señal eléctrica producida por un golpe de fuerte intensidad en el pad de plato



*Figura 78: Esquema eléctrico pad de plato*

|                                 | Fuerte    | Débil     |
|---------------------------------|-----------|-----------|
| <b>Valor máximo</b>             | 3,1V      | 650mV     |
| <b>Valor medio</b>              | 15mV      | 8mV       |
| <b>Valor mínimo</b>             | -4,3V     | -700V     |
| <b>Valor pico a pico</b>        | 7,4V      | 1,35V     |
| <b>Periodo/Frecuencia</b>       | 4ms/225Hz | 5ms/200Hz |
| <b>Típico</b>                   | 1ms       | 3ms       |
| <b>Tiempo de subida</b>         | 2ms       | 5ms       |
| <b>Tiempo de estabilización</b> | 40ms      | 35ms      |

*Tabla 15: Medidas pad plato*

### 3.3.4 Pad de bombo



*Figura 79: Vista frontal y lateral pad bombo*

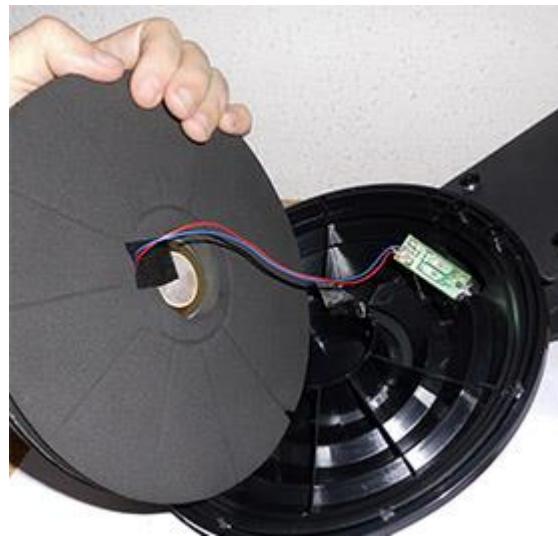


Figura 80: Vista interior pad bombo

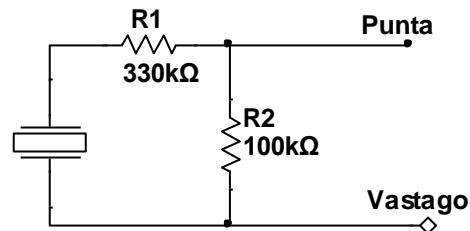


Figura 81: Esquema eléctrico Vista interior pad bombo

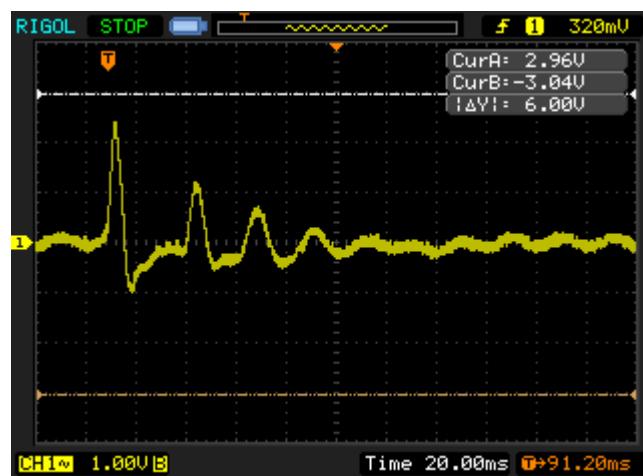
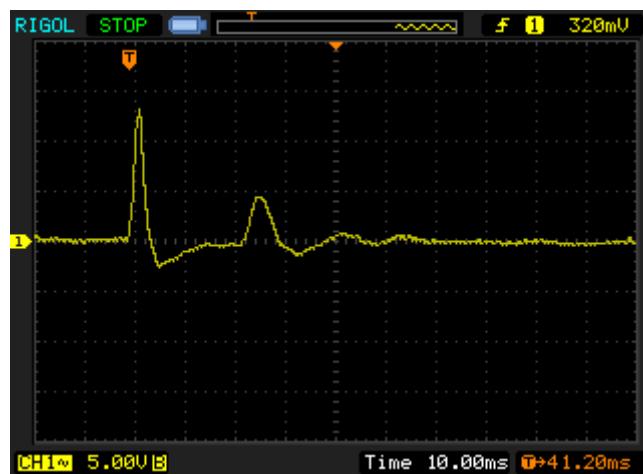


Figura 82: Señal eléctrica producida por un golpe de intensidad débil en pad de bombo



*Figura 83: Señal eléctrica producida por un golpe de fuerte intensidad en pad de bombo*

|                                 | <b>Fuerte</b> | <b>Débil</b> |
|---------------------------------|---------------|--------------|
| <b>Valor máximo</b>             | 13,2V         | 2,4V         |
| <b>Valor medio</b>              | 200mV         | 10mV         |
| <b>Valor mínimo</b>             | -2,60V        | -1V          |
| <b>Valor pico a pico</b>        | 15,8V         | 3,4V         |
| <b>Periodo/Frecuencia</b>       | 24ms/41Hz     | 32ms/31Hz    |
| <b>Tpico</b>                    | 2ms           | 4ms          |
| <b>Tiempo de subida</b>         | 2ms           | 4ms          |
| <b>Tiempo de estabilización</b> | 54ms          | 100ms        |

*Tabla 16: Medidas pad bombo*

## 3.4 Diseño Acondicionador

Visto el estudio realizado en la sección 3.3 se observa la necesidad de interponer un circuito que amolde las señales del sensor. Si no fuera así el microcontrolador procesaría señales oscilantes que aportan poca información y generan mucho gasto de procesamiento. Además el software podría no ser capaz de decidir. Es imprescindible por tanto utilizar un circuito, denominado acondicionador, diseñarlo y simularlo.

Multisim es un potente programa para diseñar circuitos electrónicos y realizar simulaciones. Dispone de una base de datos actualizada de componentes electrónicos así también permite trabajar con modelos Spice , VHDL o un VI de Labview como pueda ser un multímetro. [Dnati]

Ventajas frente a Orcad que es el enseñado en la EUIT de Telecomunicación:[Dvisi]

- Base de datos actualizada
- Simulación más precisa
- Simplicidad en los comandos y máxima potencia de calculo

Multisim es software propietario por lo que para este trabajo se utilizará la versión de estudiante. El límite de esta versión es una duración de un año.

El proceso que se utiliza para la simulación es sencillo, una vez se controla la herramienta Multisim, se dispone del circuito del esquemático, correctamente configurado y, sin fallos eléctricos. Se recoge el fichero “.txt” proveniente del osciloscopio y del análisis del apartado 3.3 y se coloca en la simulación del circuito acondicionador. Para ello se dispondrá del objeto Vpwl que permite seleccionar un fichero como fuente de entrada.

A partir de aquí se seleccionan los parámetros adecuados del análisis transitorio de corriente alterna y se simula obteniendo las gráficas.

### 3.4.1 Solución 1: Acondicionador Pasivo

#### 3.4.1.1 Introducción

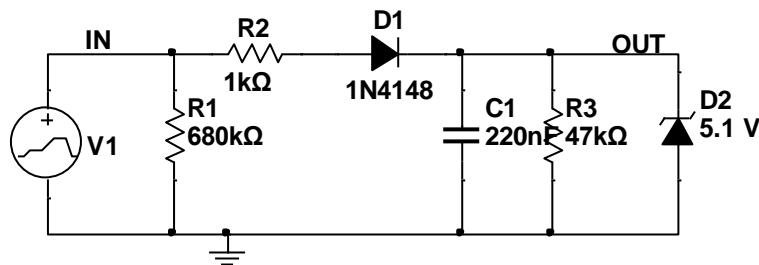
El primer acondicionador que se propone es un circuito pasivo (figura 84) caracterizado por un rectificador de onda positiva, una etapa de filtrado y un limitador. [Tomo3]

Utilizaremos componentes simples, en su mayoría pasivos porque:

-No necesitan alimentación y por tanto el consumo es menor. Los sensores piezoelectrinos son sensores generadores que suministran energía eléctrica.

-Su coste de adquisición es menor dado que son mucho más comunes y económicos. Además son sólo 6 componentes los que acompañan a cada piezoelectrino.

### 3.4.1.2 Estructura



*Figura 84: Acondicionador de la solución 1 [Tomo3]*

Desmontamos el primer circuito en varios subsistemas:

- Resistencia de carga R<sub>2</sub>: La resistencia R<sub>2</sub> cargará la tensión proveniente del piezoelectrino. Deberá ser de un valor alto pero no extremadamente pues empezarían a aparecer irregularidades tales como ruido electrónico o tolerancias (5% de tolerancia en 1Meg son 500K).
- Rectificador de media onda: El diodo de silicio 1N4148 (0,7V de tensión de umbral) y su resistencia limitadora de corriente R<sub>1</sub> permiten el paso de valores positivos de tensión, protegiendo la integridad del ADC.
- Filtro Paso Bajo: El condensador C<sub>1</sub> y la resistencia R<sub>3</sub> forman un filtro paso bajo que eliminará las señales de alta frecuencia. Su objetivo es almacenar el valor máximo y suavizar las abruptas curvas que derivan del piezoelectrino.

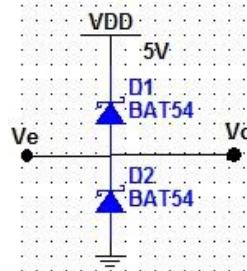
$$f_c = \frac{1}{2 * \pi * C_1 * R_3} = 15\text{Hz}$$

- Limitador: El diodo Zener actúa como limitador de 5'1V. La función de este elemento es protección. Si detecta una tensión superior a 5,1V cortocircuita sus terminales. Un sensor cerámico piezoelectrino puede dar hasta 30V lo que afectaría seriamente al sistema.

Es poco común utilizar el diodo Zener en esta configuración. Lo habitual es ver al Zener en la zona de ruptura actuando como un regulador, manteniendo la tensión en la carga con mínimas fluctuaciones. Por ejemplo eliminando la tensión de rizado en las fuentes de

alimentación. Se recuerda que las diferentes zonas que posee un diodo son: Directa, inversa y la de ruptura.

Entrando más en el tema de limitadores se pensó en utilizar una estructura ampliamente usada. Dispuesta en la figura 85. Pero se descartó por necesitar más componentes y una tensión adicional. [Mal91]



*Figura 85: Limitador con diodos Schottky*

La idea era utilizar diodos Schottky porque tienen una tensión umbral inferior a la de los diodos de silicio, alrededor de 0,2V. Haciendo más preciso el rango de funcionamiento. Estos diodos son utilizados cuando se necesita mucha velocidad y pocas perdidas. [Mal91]

$$\text{Si } V_e > V_{cc} + 0,2 \rightarrow V_o = 5V; \quad D1 \text{ CONDUCE}$$

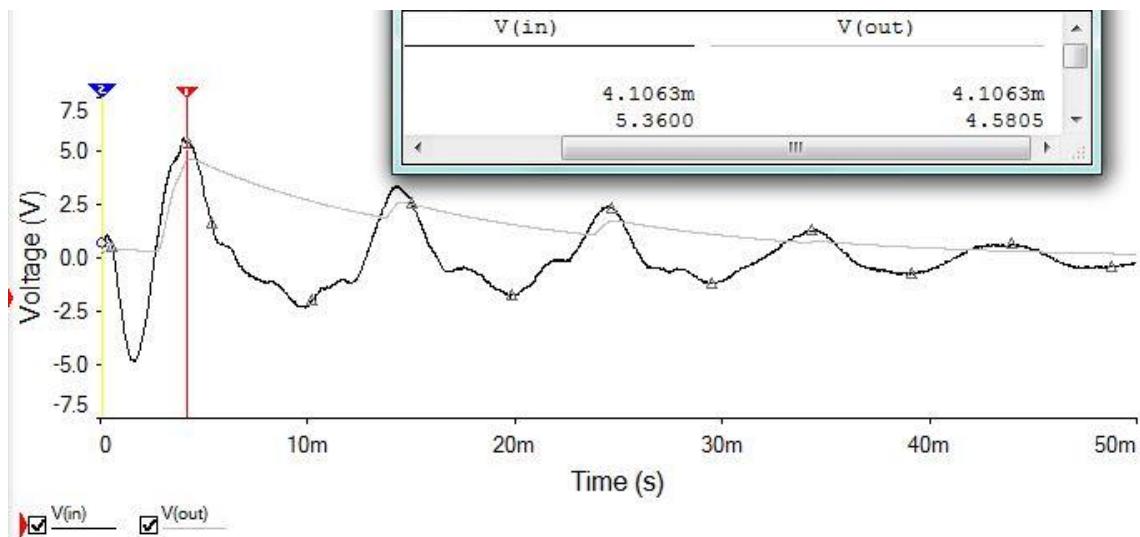
$$\text{Si } V_e < GND - 0,2 \rightarrow V_o = 0V; \quad D2 \text{ CONDUCE}$$

### 3.4.1.3 Simulación

En la simulación de la figura 86 se indica la caída de tensión producida por el diodo de silicio (-0.7V).

Además, el problema principal, el sensor tiene una impedancia altísima, visto en el apartado 2.2. El acondicionador no está adaptado y se producen pérdidas de información. La señal de entrada dependería en gran medida del pad conectado, del piezoelectrónico que disponga, de las resistencias internas de dicho circuito... Una gran cantidad de factores que no son controlables y perjudicarían la ejecución del algoritmo.

Además de que si la entrada supera los 5,1V no habría sensibilidad puesto que se saturaría. Esto se ha comprobado en la figura 71 del pad de caja donde se ha llegado a 23V con un golpe fuerte. Es necesario atenuar este valor de entrada a un valor conocido utilizando resistencias lo que cambiaría aún más la impedancia de este acondicionador.

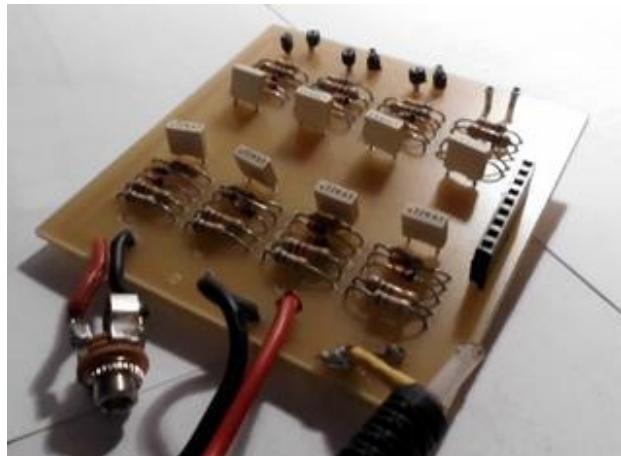


*Figura 86: Simulación acondicionador 1*

El cálculo de tau no está optimizado. Se producen falsos picos. Consultar el apartado 3.4.3.

Aún con esas dificultades y perdiendo precisión puede ser utilizado si se calculan todas esas posibles circunstancias. Se construyó en PCB y se utilizó en las primeras pruebas.

En la fotografía de la figura 87 se observa la disposición de 8 acondicionadores pasivos dispuestos con conectores Jack mono.



*Figura 87: Construcción PCB del acondicionador 1*

### **3.4.2 Solución 2: Acondicionador Activo**

#### **3.4.2.1 Introducción**

Utilizando los conocimientos de la carrera se pretende diseñar un acondicionador detector de pico que supla las carencias del anterior.

Se selecciona el detector de pico de precisión formado por dos amplificadores operacionales realimentados negativamente y se adapta al problema del piezoelectrónico.

Así se parte de las ventajas realimentación negativa. [Fra05]

- Desensibilización de la ganancia
- Menor distorsión lineal
- Menor ruido
- Control de las impedancias terminales. Debemos adaptar impedancias. El pad está conectado al circuito a través de un cable largo coaxial con sus capacidades parásitas. Por consiguiente a parte de la alta impedancia del piezoelectrónico debemos contar con las atenuaciones y distorsiones del cable e impedancia de entrada ADC. Aparte recordar que los pads fueron diseñados por sus fabricantes con resistencias en paralelo de un valor alto 150k (Figura 81).
- Mejora del ancho de banda

Inconvenientes

- Perdida de ganancia de lazo abierto
- Posible oscilación

#### **3.4.2.2 Estructura**

Además de rectificar con precisión una señal, los diodos y amplificadores operacionales pueden interconectarse para construir un detector de pico [Fra05]. Este circuito sigue los picos de voltaje de una señal y almacena los valores más altos en un condensador. El voltaje de pico más alto se almacena hasta que el condensador se descarga por medio de un interruptor mecánico o electrónico.

En lugar de interruptor de restablecimiento se ocupa una resistencia de gran valor. En este caso nos desvinculamos del libro de Sergio Franco pues no nos interesa colocar un MOSFET y utilizar una salida del microcontrolador para controlarlo. Serían necesarios un total de 16 MOSFET y sus 16 puertos de micro. En su lugar colocamos una resistencia de

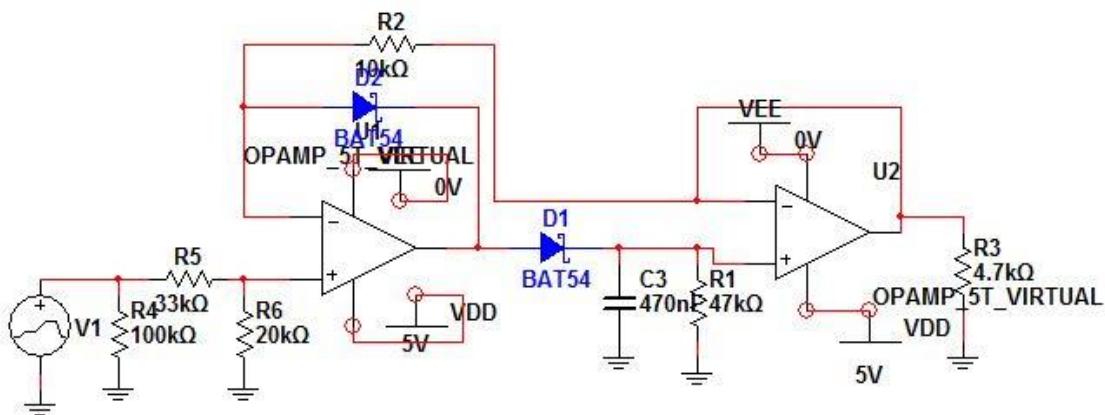
descarga en paralelo con C1 que calculamos en el apartado 10.4. Con ello se consigue una descarga suave y controlada.

Además se han elegido diodos shottky y amplificadores rail to rail para conseguir el margen de salida más amplio.

En el circuito seguidor y retenedor de pico de la figura 100 se identifican 4 bloques:

- Una memoria analógica que retiene el valor del pico más reciente; es el condensador C3
- El diodo D2, un componente de corriente unidireccional que carga a C3.
- Un rastreador de voltaje formado por el operacional U1
- Resistencia de descarga R1.
- U2 separa el voltaje del condensador impidiendo que se descargue en R.

Las resistencias R4,R5 y R6 se utilizan para atenuar la señal 2/5. De esta forma el voltaje máximo de 12V se convierte en 4,8V.



*Figura 88: Acondicionador 2*

### 3.4.2.3 Elección del Amplificador Operacional

Para la elección del Amplificador Operacional se deben cumplir una serie de criterios. El conocimiento del amplificador uA741 en profundidad [Fra05] será el punto de partida y de referencia para la comparación. Adelantar que no se puede utilizar cualquier A.O. sin que cumpla los siguientes requisitos:

- Una única alimentación de 5V (Single Supply). Por tanto se descarta de partida a un A.O. con alimentación simétrica nominal de +/−15V como el 741. Así se re-

duce la complejidad del diseño, pues en caso contrario será necesario incorporar estructuras que produzcan dichas tensiones y esto son más componentes. La elección de 0 a 5 V radica en ser el rango de conversión del ADC y la alimentación del microcontrolador.

- Salida Rail to Rail: Característica de algunos A.O. que permite a la salida alcanzar voltajes muy cercanos a los de alimentación. Se requeriere todo el margen de tensiones de 0 hasta 5V. Así con esta característica la salida es +/-100mV de las fuentes de alimentación, es decir que el rango real del amplificador teórico sería, en el peor de los casos, de 100mV a 4,9V. En contraprestación al archiconocido 741 con +/-2V de su rango de tensiones.
- No linealidad: Dado que el circuito de la figura 88 no funciona como amplificador aspectos tales como ganancia en lazo abierto y el ancho de banda no parecen indispensables. Aun así como se requieren las propiedades de la realimentación negativa, y dado que el 741 tiene más de 20 años de existencia, seguro que son más óptimas.
- Mejora de las características esenciales: Mayor impedancia de entrada, menor impedancia de salida y más ganancia.
- Slew Rate: Efecto no lineal en los amplificadores, se define como la máxima tasa de cambio en el voltaje de salida cuando cambia el voltaje a la entrada. Habrá situaciones en las que el A.O. deberá responder a salidas de 0 a 5V en menos de 1 ms por lo que será un parámetro a tener en cuenta. Un valor bajo nos daría distorsiones. El 741 tiene un Slew Rate de 0.5V/us.
- Encapsulado THD y SMD, dispositivo comercial y no excesivamente caro. De nada sirve tener un A.O. excelente sino se puede conseguir repuestos. El 741 es un buen ejemplo; barato y fácil de encontrar.
- Bajas tensiones de offset, corrientes de offset, derivas y ruido. Provocan errores que es mejor reducir. Son también llamados amplificadores de precisión pues efectúan una corrección interna de estos errores reales.

Con todas estas premisas se buscó orientación en un fabricante líder de circuitos integrados como Texas Instruments. En su plataforma online de comparación se dispusieron varias alternativas. En un principio en su catálogo hay disponibles 1381 dispositivos. Se marcó la opción “Precision Amplifier” y dentro de esta categoría se seleccionó la alimentación única y de valor máximo 5 V. Como se utilizaran en cantidad 2 A.O, por entrada y alrededor de unas 6 entradas se seleccionó la máxima integración de amplificadores (4 en cada chip). También se seleccionó “rail to rail output” y que se encontrara en un encapsu-

lado habitual como PDIP (Inserción) y SOIC (Superficial). Por último un precio máximo por unidad de 1,5\$

Sobre esta selección aparecieron 3 identificaciones: OPA4342, TLV2774 y LMC6484.

Con estos 3 amplificadores operacionales se pasó a un análisis con las hojas de características y las notas de aplicación [Texaso]. Con estos documentos de la mano se dedujo la utilización del TLV2774.

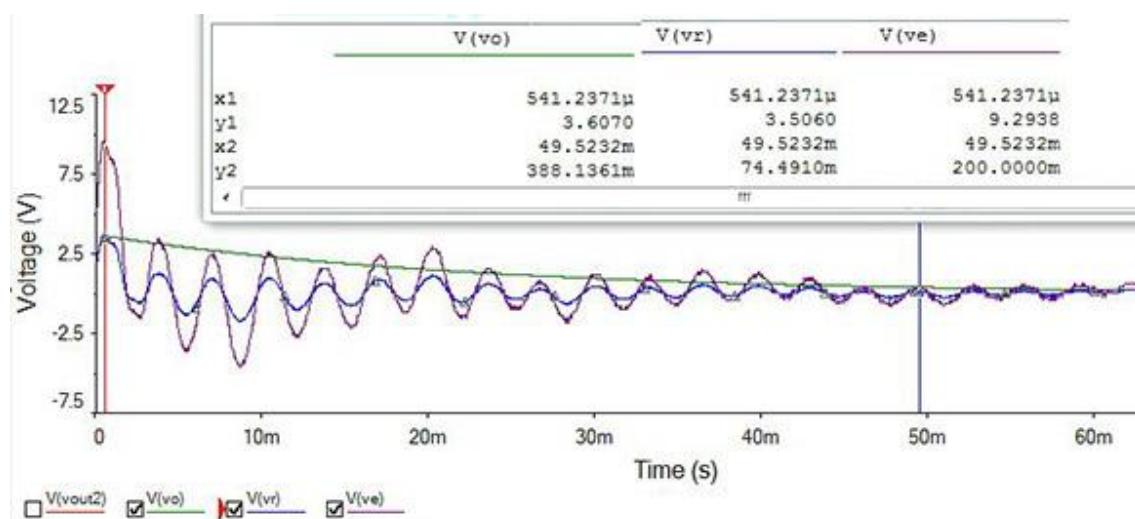
### **3.4.2.4      *Simulación***

Si introducimos un nuevo pico positivo por V1, la salida del A.O. U1 será positiva (D2 estará en inversa y D1 estará en directa y conducirá cargando C3) U1 estará realimentado negativamente a través del lazo D1-U2-R2. En R2 no fluye corriente puesto que  $V_o = V_1$  y U1 compensa la caída en D2.

Cuando el pico decrece,  $U_1$  también decrece y como el seguidor almacena la carga de  $C_1$  ( $D_2$  conduce y  $D_1$  no).  $U_1$  en realimentación negativa a través de  $D_1$ , cortocircuito virtual y  $V_o$  será la descarga del condensador.

Si la tensión de entrada es mayor que VCC satura a Vcc-02

En la figura 89 se observa que  $V_{(ve)}$  es la señal de uno de los pads de caja,  $V_{(vr)}$  es esa misma señal atenuada  $2/5$  y  $V_{(vo)}$  es la salida del detector de pico. La descarga de esa señal está en función del valor del condensador  $C_3$  y la resistencia  $R_1$  del esquema de la figura 88. No son casuales y se explican en el apartado 3.4.3.



**Figura 89:** Simulación acondicionador 2

### 3.4.3 Calculo de $\tau$

Se ha contemplado en las figuras 86 y 89 la descarga del condensador. Esta descarga es importante pues durante ese tiempo nuestro procesador no necesita captar información porque ya ha captado la sensibilidad. Abstrayendo, sería el tiempo entre golpe y golpe para que el microcontrolador no esté hiperactivo.

Este parámetro será adaptativo pero para tener una referencia se busca el máximo número de golpes dados por una persona. Se recurre al Guiness de los records, donde es difícil encontrar información y por ello se busca otras fuentes. Como datos fiables se tienen los 1203 beats por minuto (20 golpes por segundo) hechos por Mike Mangini en single stroke roll (1 golpe alternativamente con cada mano) o los 1744 dados por Eric Okamoto en la modalidad de doublé stroke roll. [Dfast]

Ahora con estos datos se elige un parámetro de diseño. Los máximos golpes por segundo que es capaz de captar el sistema. Utilizar una cantidad redonda y musical como 16 golpes por segundo. Una cantidad muy alta debido que en un bpm=60 debe de dar golpes en semifusa y es equivalente a que en un tempo de 240bpm debe de dar golpes en semicorcheas o lo que es lo mismo buzz o ruido.

Ahora con la ecuación de descarga, suponemos que la tensión inicial  $V_0=5.1V$ , que el tiempo requerido para los propósitos es de  $t_{max}=63ms$  y que la tensión final es muy pequeña, despreciable e inferior a  $V=100mV$ . Con ello conseguiremos que el ADC obtenga datos que sean significativos hasta entonces. Nuestra única incógnita es la tau que como sabemos es RXC.

$$V = V_0 \cdot e^{\frac{-t}{\tau}}$$

Ahora se eligen los componentes según la serie de valores comerciales E-24 y se escogen  $R=47k$  y  $C=470nF$  con lo que se tiene una constante de tiempo de 22,1 ms. Esto implica que la tensión del condensador (utilizando la ecuación 1) en 63 ms es 90mV. Tensión prácticamente 0.

### **3.4.4 Construcción PCB**

#### **3.4.4.1 EAGLE**

EAGLE (Easily Applicable Graphical Layout Editor) es un editor de esquemas y un editor de PCB (con autoruteo incluido) tanto para gente experto como para estudiantes. Como aval de su potencial argumentar que es la herramienta empleada por los ingenieros de Sparkfun y Digilent.

Entre los puntos fuertes de dicho programa es su sencillez de edición y que dispone de prácticamente todos los dispositivos existentes en el mercado. Además su potencia de cálculo basada en scripts y en programas definidos por usuario (ULP) es muy superior a otras plataformas. Otro de los beneficios es que dispone de una versión libre de la página oficial de CadSoft Inc. (la versión actual es la 6.6). [Dcads]

Las limitaciones de dicha versión freeware son:

- El área de la placa está restringida a 100 x 80 mm. Fuera de esta área no es posible situar encapsulados ni dibujar señales.
- Sólo se pueden usar dos capas (no hay capas internas).
- Un esquema sólo puede tener una hoja.

Concluiremos que estos requerimientos son suficientes para nuestra aplicación.

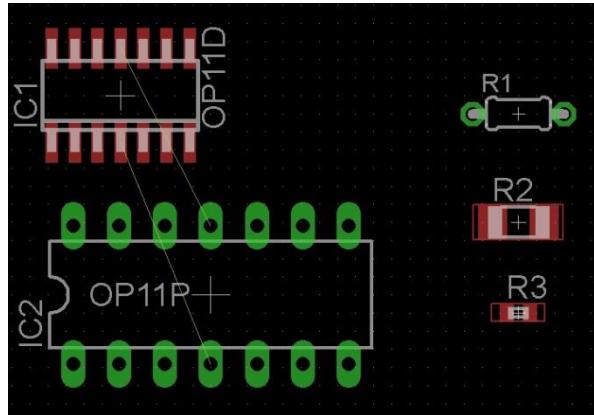
##### **3.4.4.1.1 Archivos de EAGLE**

En la siguiente tabla se muestra un listado con los tipos de archivo más importantes que pueden ser editados con EAGLE y con los cuales hemos trabajado:

| <b>Tipo</b>                  | <b>Ventana</b>               | <b>Nombre</b> |
|------------------------------|------------------------------|---------------|
| Placa                        | Editor de Líneas de conexión | *.brd         |
| Esquema                      | Editor de Esquemas           | *.sch         |
| Librería                     | Editor de Librerías          | *.lbr         |
| Fichero Script               | Editor de Textos             | *.scr         |
| Programa en Lenguaje Usuario | Editor de Textos             | *.ulp         |
| Cualquier archivo de texto   | Editor de Textos             | *.*           |

Dependiendo de la tecnología de montaje (inserción o superficial), de las herramientas que se dispongan y de la habilidad para hacer nuestra placa más evolucionada, más pe-

queña y manejable, pero sobre todo fiable. Para muestra la diferencia de encapsulados (figura 90) entre un componente con 4 amplificadores en DIL14 y SO14. También las resistencias de inserción y smd de 0204 y 1208.



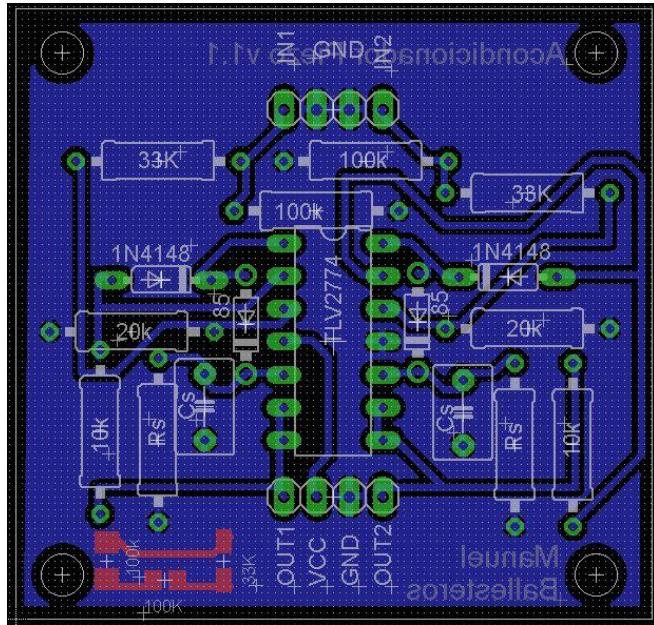
*Figura 90: Distintos tamaños de encapsulado*

#### 3.4.4.1.2 Elaboración de PCB

En la herramienta de CAD el proceso simplificado que seguimos es el siguiente:

- Abrir el esquemático (.sch). Introducir mediante el uso de librerías y el buscador todos los componentes necesarios, fijándose en su correspondiente huella. Muchos de ellos están nominados por su referencia comercial lo que facilita la búsqueda.
- Unir los componentes por cables de manera que el resultado final sea el deseado. También con el fin de hacer más claro el esquemático utilizar etiquetas entre pines de los dispositivos.
- Comprobar con el ERC(Electric Rule Check) que no existen fallos en el esquemático tales como cortocircuitos o pines desconectados.
- Pasar el esquemático a PCB (.brd) y en ella se colocan los componentes de la manera más adecuada.
- Ejecutar el programa de usuario (.ulp) “brd\_to\_dsn.ulp” el cual nos permite trasponer el diseño en un formato .dsn que será el cual procese el freerouting.
- Utilizar el freerouting (disponible en <http://www.freerouting.net/>) programado en lenguaje java que permite tener un ruteado de mayor calidad y en menor espacio. Eligiendo unas reglas de diseño claras (ancho de pista, distancia entre pads...) realiza una optimización del diseño durante un largo periodo de tiempo.

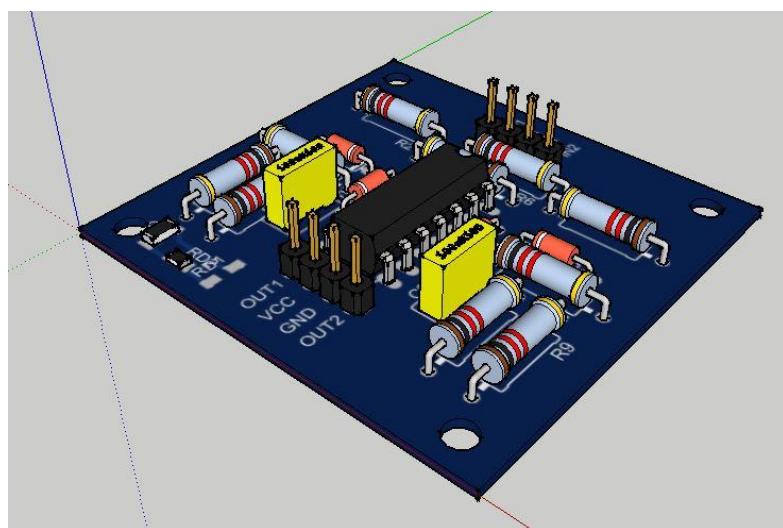
- Ejecutar el fichero que da el freerouting como un script (.scr), volcándolo en la placa. Por último en la placa le añadimos los últimos toques posibles (plano de masa, texto aclaratorio). Y el resultado es la figura 91.



*Figura 91: Layout de circuito acondicionador*

#### 3.4.4.2      *Modelo 3D*

Antes de mandarla a producción es aconsejable hacer pruebas por ejemplo viendo cómo quedaría ensamblada. Así se observan deficiencias en encapsulado o en serigrafía. Un complemento del Eagle junto con la herramienta de diseño gráfico Sketch'Up nos convierte un circuito plano en un objeto físico distingible como la figura 92.



*Figura 92: Circuito acondicionador en modelado 3D*

### **3.4.4.3 Envío de PCB a la fábrica**

En la actualidad decir que los costes de logística y producción han disminuido parece un tópico pero resulta que en la construcción de una PCB profesional así es. Se puede obtener un diseño propio en 10 PCB de 2 caras por menos de 20 € con el envío incluido.

Las características son una placa de 2 caras de 5x5 cm construida en FR-4 (fibra de vidrio de espesor 1mm) con capa anti soldadura y de serigrafía. El acabado es HASL (hot air solder).

Realizando una prospección de mercado en distintos medios (Dangerous Prototypes y Hack a day) tenemos 3 empresas profesionales. En la tabla 17 sus conclusiones: [Dyour]

|                   | <b>Proceso</b> | <b>Tiempo de espera</b> | <b>Calidad</b> | <b>Testeo Eléctrico</b> | <b>Precio(\$)*</b> |
|-------------------|----------------|-------------------------|----------------|-------------------------|--------------------|
| <b>Itead</b>      | Normal         | 1 mes                   | Normal         | Si                      | 9,9                |
| <b>SeedStudio</b> | Normal         | 20 días                 | Normal         | 50%                     | 9,9                |
| <b>OSHspark</b>   | Normal         | 2 meses                 | Alta           | No                      | 20                 |

*Tabla 17: Comparación de empresas dedicadas a la construcción de prototipos electrónicos*

Tras la elección se describe el proceso.

- Se diseña la PCB en el programa CAD (Eagle).
- Se generan los ficheros gerber con la herramienta CAM processor. Opcionalmente se pueden examinar las capas de esos ficheros con el programa Viewplot (figura 93).
- Rellenar el formulario de compra, pagar y enviar el fichero con los gerber. En el tiempo estimado las placas llegan al domicilio.

Los ficheros Gerber que requiere la empresa son:

- Top layer: pcbname.GTL
- Bottom layer: pcbname.GBL
- Solder Stop Mask top: pcbname.GTS
- Solder Stop Mask Bottom: pcbname.GBS
- Silk Top: pcbname.GTO
- Silk Bottom: pcbname.GBO
- NC Drill: pcbname.TXT

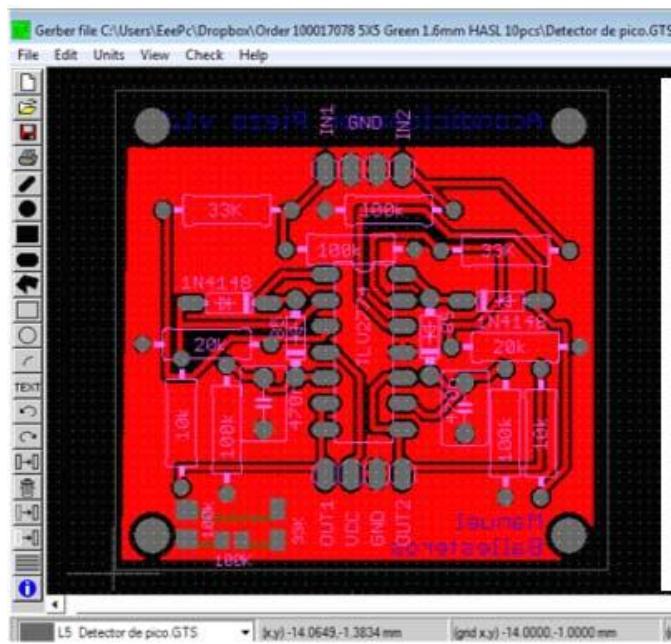


Figura 93: Revisión del resultado de los gerber con Viewplot

El resultado final de la placa en cara de componentes y el prototipo ensamblado en la figura 94.

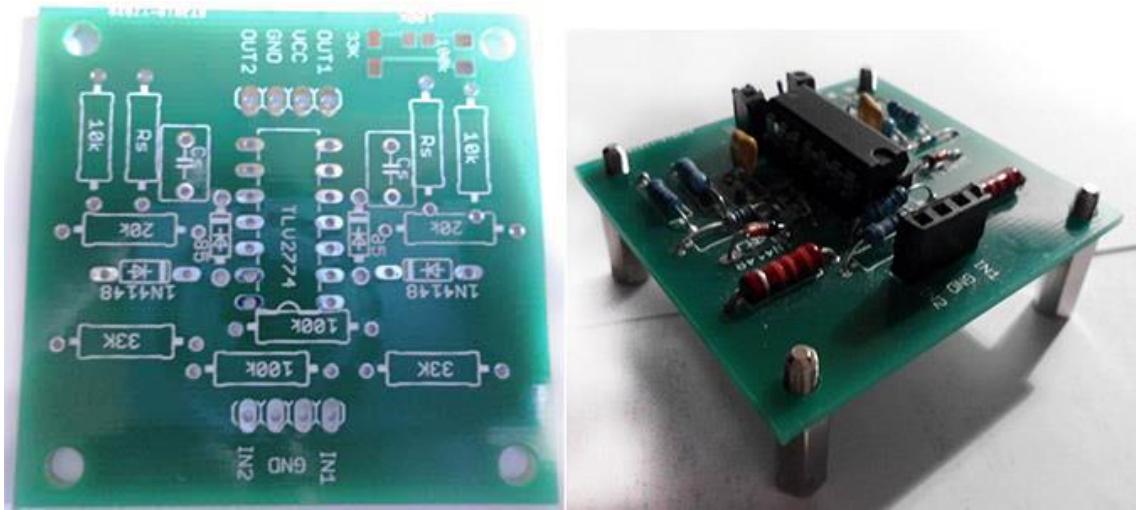


Figura 94: Resultado final PCB acondicionador

## 3.5 Software y algoritmia

### 3.5.1 Código y diferentes versiones

Hasta obtener el código final que puede consultarse en el anexo, hubo un ciclo de vida del firmware obteniendo diferentes resultados. En este apartado se enumeran las versiones.

#### 3.5.1.1 Versión 0.1

Al tratarse de una toma de contacto con los dispositivos y con el entorno de desarrollo el objetivo principal se marcó como sencillo: Enviar un comando MIDI si se detectaba la vibración del piezoelectrónico. Para conseguirlo el canal ADC se encontraba en modo “polling” muestreando exclusivamente. Si el valor recogido superaba el código referido a la tensión de 1 Voltio, la UART enviaba el valor recogido más una cadena de caracteres “Envío:”

#### 3.5.1.2 Versión 0.5

Uno de los problemas de la anterior versión era la cantidad de información enviada al producir un evento. Era necesario realizar una selección.

Siguiendo con el escaneo exclusivo de un solo canal, se realizó el algoritmo para recoger el máximo valor el cual era enviado por el puerto serie.

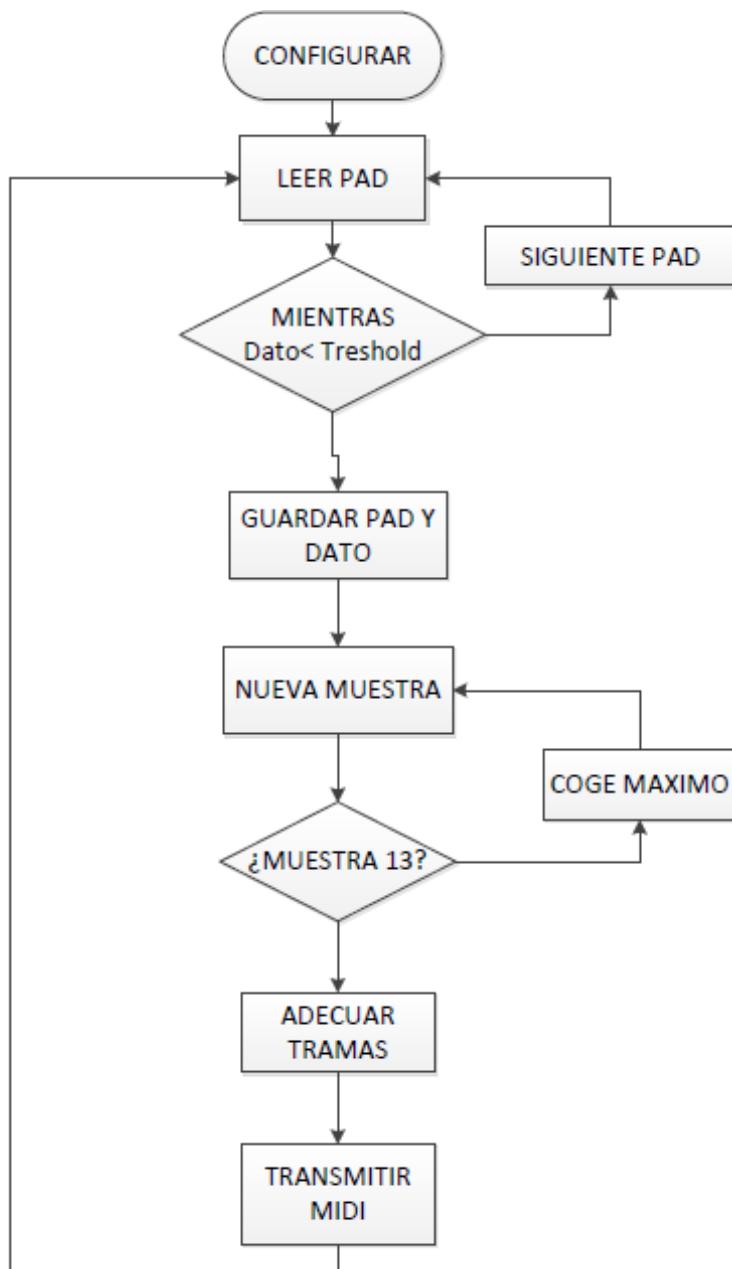
Además en esta ocasión se utilizaban las especificaciones de formato y hardware propias del protocolo MIDI. Se comprobó satisfactoriamente con el programa de visualización de eventos MIDI-OX.

#### 3.5.1.3 Versión 1.0

Se incluyen 6 canales y se sigue utilizando el algoritmo de máximo aplicado a 6 pads.

En el diagrama de flujo (figura 95) se denota que tras la configuración (inicialización de variables, puesta en marcha de periféricos) se lee con el ADC cada uno de los pads secuencialmente.

Si se supera el umbral se ha producido una pulsación. Se obtienen 13 muestras y se obtiene el máximo. Con esta información se componen las tramas y se transmite el mensaje MIDI.



*Figura 95: Diagrama de flujo programa principal*

Este sencillo diagrama de flujo es el fundamento del resto de versiones. Se añaden más funciones y características abriendo el abanico de sucesos pero el núcleo del algoritmo es el mismo. Se observan los pads y, si alguno supera el umbral, se captura su máximo, se procesa y se transmite la trama.

#### 3.5.1.4 Versión 1.1

Se incluyen 6 canales, así como una serie de parámetros indispensables (Gain, threshold, Masktime, Note, Curve...).

Utilización de un Timer que gestiona el tiempo transcurrido desde que se encendió el dispositivo. Este se emplea para que no se efectúen falsos disparos junto al parámetro Mask Time.

#### **3.5.1.5      *Version 1.2***

Se modifica el algoritmo creando un parámetro similar a Scan Time; un tiempo en el que se toman muestras de un único pad.

En anteriores versiones la búsqueda del máximo requería un tiempo fijo de 1,3ms, tomando 13 muestras. Con este parámetro se adapta a las necesidades de los pads. Por ejemplo, la caja tendrá un tiempo de 1ms y el bombo de 3 ms de acuerdo a las gráficas del apartado 3.3.1.

#### **3.5.1.6      *Versión 1.3***

Se introduce el código necesario para ejecutar el panel de control. Se integra tanto la interfaz de teclado como la pantalla LCD. (Ver apartado 3.6)

Aparece el menú de interacción intuitivo y robusto.

#### **3.5.1.7      *Versión 1.4***

Se introduce el algoritmo de control del pedal del HH con sus temporizaciones correspondientes.

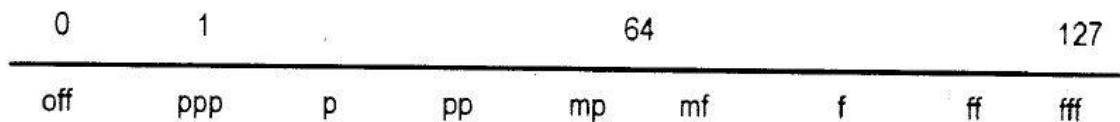
#### **3.5.1.8      *Versión 2.0***

Optimización y solución de errores tras pruebas de estrés y funcionamiento. Puede consultarse en el anexo.

### **3.5.2 Problemática sensibilidad**

Evidentemente es importante controlar la dinámica de un instrumento, ya que es la única forma en la que el músico puede poner su marca personal. A los artistas dedicados a la percusión se les hace particularmente importante este aspecto siendo la única forma de canalizar la emotividad al interpretar su pieza musical.

La norma MIDI procesa un campo de datos con esta información. Diferenciando con 128 valores todas las distintas graduaciones musicales que son 8 (figura de matices número 96). Para MIDI 127 es un matiz fortíssissimo (golpe máximo), o sería un sonido apagado y 1 pianississimo (golpe mínimo).

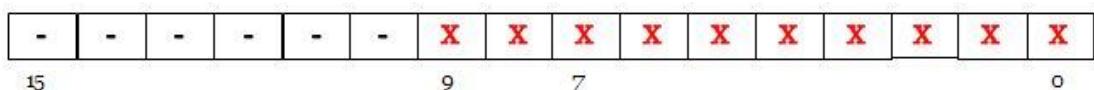


**Figura 96: Correspondencia de los matices en MIDI**

Además, incidir que en el algoritmo, gracias al periférico conversor analógico digital, se dispone de un rango mayor de 1024 valores de intensidad de señal. Producíendose el dilema.

La función analogRead() de Arduino lee los 10 bits del ADC y para enviar la trama sólo se necesitan  $2^7$  bits=128. El equilibrio de fuerzas será balanceándose en la precisión a un determinado coste. Por tanto gráficamente y, mostrado en color rojo con X, en el registro del ADC:

## LECTURA DEL ADC



### **3.5.2.1 Solución 1: Truncamiento con desplazamiento.**

El contenido devuelto por el convertidor analógico digital será desplazado en 3 posiciones y almacenado en una variable de 7 bits. Se trata de una forma de dividir entre 8 el valor inicial.

Se supone, por ejemplo, que el batería marcar un ritmo con un golpe de intensidad  $mf$  (mezzoforte) que correspondería a la mitad del rango; aparecería en el registro de datos del ADC:

|          |   |   |   |   |   |   |   |   |   |
|----------|---|---|---|---|---|---|---|---|---|
| <b>1</b> | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|----------|---|---|---|---|---|---|---|---|---|

Obteniendo un valor en decimal de 512. Si se aplica esta solución, se desplazaría el contenido 3 posiciones, obteniendo un valor 64 que entraría en el tipo char de C/C++.

|   |          |   |   |   |   |   |   |   |   |
|---|----------|---|---|---|---|---|---|---|---|
| - | <b>1</b> | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|----------|---|---|---|---|---|---|---|---|

Otro ejemplo sería dar un golpe piano de valor 128. Eso daría una velocidad, siguiendo esta solución, de 16.

### 3.5.2.2 Solución 2: Aproximar el valor por la pendiente.

En el siguiente diagrama cartesiano se dibuja una recta. El eje x es la muestra tomada por el ADC e y, la velocidad que se va a enviar. Para lo cual se realiza el cálculo de la pendiente matemáticamente por sus puntos extremos.

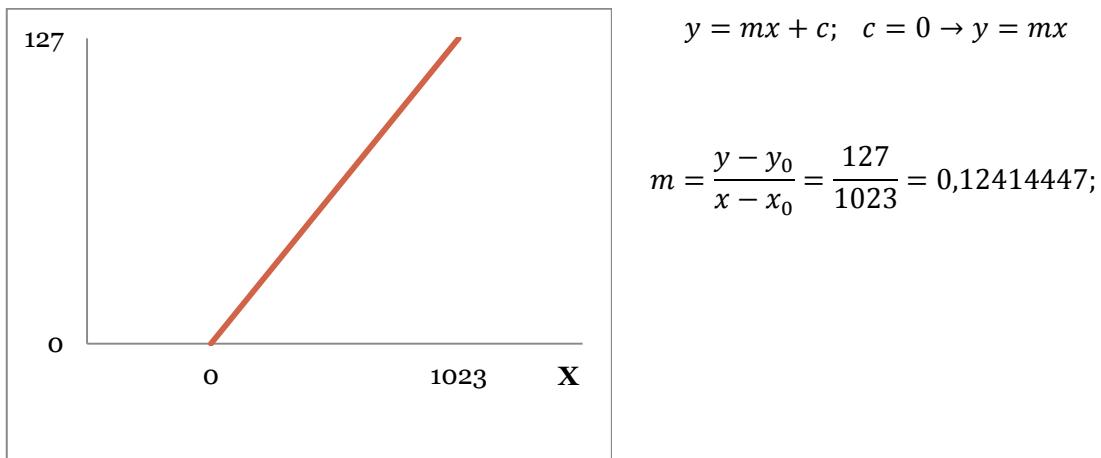


Figura 97: Grafica y fórmulas de conversión

Si ahora se supone el mismo ejemplo, el gople  $mf$  sería:

$$y = m \cdot x = 0,12414447 * 512 = 63,562086$$

Una vez realizada esa multiplicación en binario debe codificarse el resultado decimal 63,562086 usando el metodo de la IEEE 754 (estandar para la coma flotante), que es el estandar que se utiliza en los lenguajes maquina para representar los números reales (float en C). Para ello tenemos que obtener el signo, el exponente y la fracción. El procedimiento es el siguiente: [Jaso8]

- Dado que es un número positivo, el bit de signo es "0".
- Escribir el número usando notación binaria. 1º) Se transforma la parte entera a binario. 2º) Se sigue con la parte fraccionaria, multiplicando cada número por 2. Si el resultado es mayor o igual a 1 se anota como un uno 1 binario. Si es menor se anota como un 0 binario. El resultado, en este caso, es 111111,1000111101001001.
- Normalizar el número en coma flotante. Mover la coma decimal a la izquierda, dejando sólo un 1.  $111111,1000111101001001 = 1,11111000111101001001 \cdot 2^5$
- El significante es la parte a la derecha de la coma decimal, rellenada con ceros a la derecha si fuera el caso hasta que se tengan los 23 bits. Es decir 1,11111000111101001001.
- El exponente es 5, pero se desplaza con un numero 127, así es que  $5 + 127 = 132$ . En binario, esto se escribe como 10000100.

Poniendo todo junto en una dirección de memoria float. En verde el signo, en azul el exponente y en rojo la parte significante:

|    |   |   |   |   |   |   |   |   |    |   |   |   |   |   |   |    |
|----|---|---|---|---|---|---|---|---|----|---|---|---|---|---|---|----|
| 0  | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1  | 1 | 1 | 1 | 1 | 1 | 1 | 1  |
| 31 |   |   |   |   |   |   |   |   | 23 |   |   |   |   |   |   | 16 |
| 0  | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1  | 0 | 0 | 1 | 0 | 0 | 1 |    |
| 15 |   |   |   |   |   |   |   | 9 | 7  |   |   |   |   |   |   | 0  |

### 3.5.2.3 Conclusión

Intentar linealizar algo que no es en esencia lineal. Como una conversión analógica-digital, acarrea estos problemas:

- Necesidad de memoria volátil de mayor tamaño, un float de 16 bits que posteriormente no vas a necesitar.

- Aumento de la potencia de cálculo. En un microprocesador el desplazamiento en un registro tarda un ciclo máquina. En cambio multiplicar dos valores en coma flotante hablaríamos con seguridad de 8 ciclos máquina.
- El intérprete podría no necesitar dicha linealidad, una proporcionalidad en sus golpes porque requiere que todos los goples suenen igual.

Por consiguiente se decidió utilizar la primera opción por su sencillez de aplicación y porque su error máximo en el peor escenario posible es de 7 bits (40mV) con relación a la señal original proveniente del acondicionador. Además a la hora de reproducir el mensaje MIDI es despreciable porque en su escala de 128 se desplazaría en torno a un +/-1 LSB.

$$E_{relativo}(\%) = \frac{E_{absoluto}}{Valor\ exacto} * 100 = \frac{1}{127} * 100 = 0.79\%$$

### **3.5.3 Parámetros Roland/DD508**

Recoger el manual de usuario de un modelo comercial para observar los parámetros que son modificables por un usuario vía software. Qué información es útil para dar una buena experiencia al ejecutante y caracterizar cualquier pad.[Med12]

Se desglosarán las variables que afectan a los pads formados por piezoelectrinos denominados en estos documentos triggers (disparadores). En otra opción aparece el caso de que sea switch (piezas metálicas) en el que sólo se utilizaría un parámetro de ganancia dada su simpleza.

Se explica formalmente el fenómeno sobre el que actúan. Se incide en los problemas ocasionados por su mal uso entre los cuales está que el dispositivo no funciona por una pésima configuración. Aparecen gráficas que ayudan a la comprensión de estos conceptos.

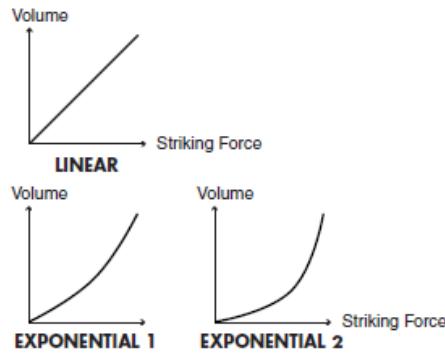
Los niveles de señal, formas de onda y duración son distintos según el pad que se utilice. Esto es una conclusión vista en el apartado 3.4 que dificulta la asignación de valores estándar y algoritmos; datos clave para el control de este sistema.

#### **3.5.3.1 Tipos de parámetros**

En conjunto hay 2 grandes grupos de parámetros:

1) Aquellos que cambian la sensibilidad del trigger: la relación entre fuerza de golpeo y intensidad de sonido producido por la misma.

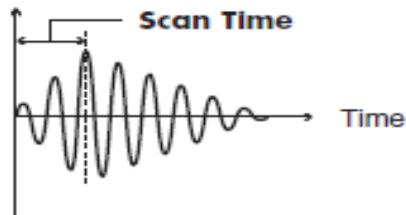
- Sensivity: Ganancia aplicable al pad. Un mismo golpe, al incrementar este parámetro provoca, si está conectado a un reproductor, un sonido más fuerte. PROBLEMA: al aumentarlo en exceso se pierden dinámicas, es decir, que a partir de una determinada fuerza de golpeo todos los golpes suenan a volumen máximo, es por eso que es recomendable graduarlo de forma correcta (tenerlo al mínimo que ofrezca una buena sensibilidad).
- Velocity curve: Permite escoger entre diferentes graficas que relacionan la fuerza utilizada para golpear el pad y el volumen producido por el módulo. De esta manera se modificar la respuesta del módulo, consiguiendo relaciones no lineales entre fuerza y volumen. Es posible elegir diferentes tipos de curva (lineal, exponencial, logarítmica...) de la figura 98. PROBLEMA: Esto nos sirve para adaptar nuestra forma de tocar la batería a los pads que tengamos. Ej) Batería de metal extremo.



*Figura 98: Parámetro velocity curve [Rolo4]*

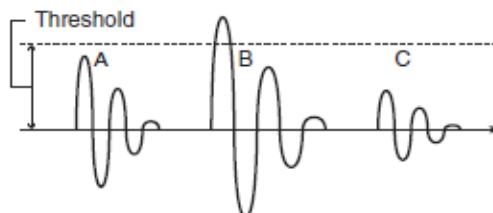
2) Eliminan o disminuyen los fenómenos de doble golpe, diafonías, y señales inintencionadas.

- Scan Time: Tiempo en que el controlador detecta el máximo. PROBLEMA: si golpes de idéntica fuerza producen distintos sonidos. RANGO: 0-4ms en pasos de 0.1ms. 40 posibles escaneos.



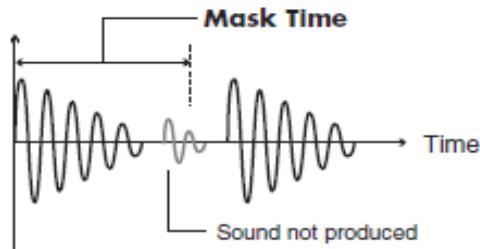
*Figura 99: Parámetro Scantime [Rolo4]*

- Threshold: Es el nivel mínimo de señal recibida por el módulo que lanza un sonido; por debajo de este nivel no se produce ningún sonido. PROBLEMA: Si es un valor muy alto no detectará pulsaciones.



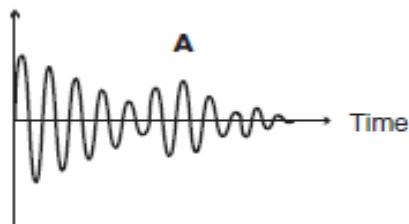
*Figura 100: Parámetro Threshold [Rolo4]*

- Mask Time. Tiempo en el que un mismo sensor no produce ninguna actividad. PROBLEMA: Se producen dobles disparos del mismo pad sin intención.



**Figura 101: Parámetro Mask Time [Rolo4]**

- Retrigger: Parámetro utilizado en formas de onda alteradas, que podrían provocar un falso disparo. PROBLEMA: triggers de percusión acústica. No utilizado.



**Figura 102: Parámetro Retrigger [Rolo4]**

- Crosstalk: Diafonía. En telecomunicaciones el concepto diafonía es el efecto que se produce cuando una señal se introduce en un canal distinto. PROBLEMA: Es el fenómeno por el cual se lanza el sonido de un pad sin que se haya producido el golpeo del mismo (figura 73). Se suele producir cuando, al golpear un pad, la vibración producida desencadena el sonido de otro pad inintencionadamente. Es habitual que se produzca en pads instalados en un mismo soporte.

### **3.5.4 El problema de la latencia**

La latencia es el tiempo que transcurre entre un estímulo y su respuesta. En el caso tratado en este proyecto será el tiempo desde que el percusionista golpee un pad hasta que el sonido se reproduce en un altavoz.

Este término es especialmente delicado en el ejemplo de un percusionista. El músico es a la vez emisor y receptor y si se produce un retardo apreciable entre el momento en que éste emite una nota y el que la escucha, tendrá problemas de interpretación.

La latencia no es apreciable cuando se interpreta un instrumento acústico pero sí cuando procesamos el sonido a través de un sistema de audio digital como un ordenador de propósito general.

#### **3.5.4.1 Latencia mínima tolerable por el oído humano**

Un tema que se ha intentado responder con un dato fiable. El oído y cerebro tienen una tolerancia relativamente alta a la latencia y varían dependiendo de la persona, de las características del sonido, de las condiciones del espacio.... Por ejemplo si es un sonido repetitivo con ataque súbito como el golpe de un tambor la tolerancia será menor que si se trata de otro de ataque lento.

El sonido se propaga por el aire a una velocidad de 340 metros/segundo. Un baterista acústico tiene su instrumento a unos 50 cm de los oídos, por tanto escucha lo que toca con un retardo de 1,5 ms después de tocarlo. Por tanto trabaja con latencias controladas.

Se ha comprobado experimentalmente con ayuda de un metrónomo que en tempo “allegretto” (120 bpm), 2 semifusas ( $16^*T$ ) son indistinguibles. Estas notas se encuentran a una distancia de 31,2 ms. Ciertamente resulta extraño escribir 2 semifusas a ese tempo, sería demasiado rápido, pero si podría corresponderse por ejemplo con un redoble de caja y probablemente representa la distancia más corta entre dos notas en un ámbito musical.

Suponemos como límite de reproducción 20 milisegundos. Desde que se pulsa el pad hasta que se envía la trama transcurren, según la figura 127 unos 3,6 ms a esto hay que anadir el procesamiento del audio en el sintetizador y la reproducción.

#### **3.5.4.2 ASIO**

Audio Stream Input/Output (ASIO) es un protocolo de tarjetas de sonido de ordenador para audio digital. Provee una baja latencia, omite capas del software, y una interfaz de

alta fidelidad, al no truncar muestras, entre la aplicación software y el hardware conectado (instrumentos electrónicos).

Funciona sobre distintos S.O. (Sistemas Operativos). Funciona sobre Windows contraponiéndose al servicio estándar DirectSound. Este servicio se utiliza para que las aplicaciones reproduzcan archivos de audio. [Dasio]

Steinberg es su propietario y desarrolló ASIO como un protocolo abierto y por tanto pudo ser adoptado por fabricantes de tarjetas y de programas de audio, por lo que a día de hoy es utilizado por todos los fabricantes de tarjetas de audio profesionales del entorno Windows.

En este proyecto se utilizará una tarjeta de sonido estándar integrada en un ordenador portátil. Por lo que esta decisión simple y fácil de obtener, debe estudiar aspectos con el tema de la latencia en el audio.

El ordenador opera con paquetes de muestras de sonido. Estos paquetes se guardan provisionalmente en buffers de memoria intermedia y el número de paquetes no se puede cambiar. Los profesionales del audio utilizan DSP (Digital Signal Processing) que trabajan con buffer muy pequeños.

Por otro lado el ordenador conecta diversos dispositivos (ratón, disco duro, tarjeta gráfica, memoria USB...) y todos deben ser atendidos por el sistema operativo. Es quien en definitiva arbitra y administra las tareas y las prioridades y el que permite a los programas de audio trabajar y comunicarse con la tarjeta de sonido y demás periféricos.

#### 3.5.4.2.1 ASIO4ALL

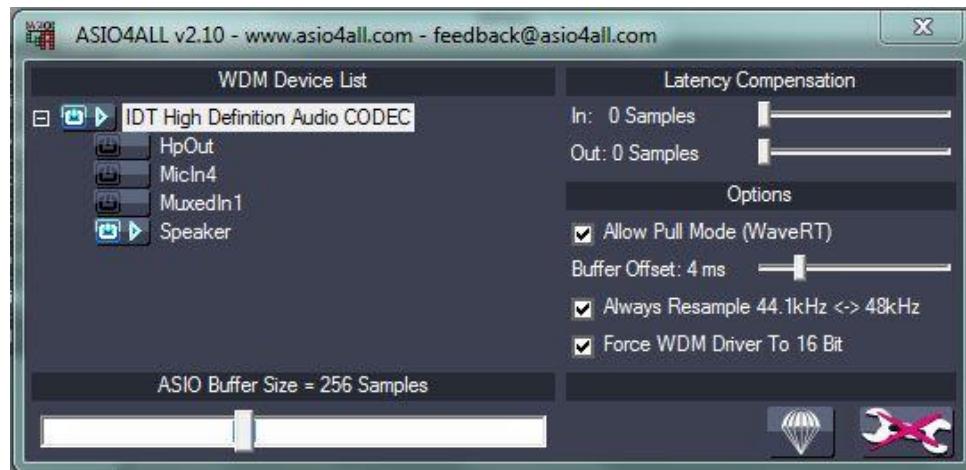
Dado que ASIO es software propietario utilizaremos una alternativa libre: ASIO4ALL, driver de baja latencia para Windows 7 con una interfaz fiable en el hardware.

#### 3.5.4.2.2 Configuración ASIO4ALL

Tras la instalación en su página web se describen las variables más importantes para configurar. El panel de selección es el de la figura 103. [Dasio]

1. Pulsar el botón de opciones avanzadas.
2. Elegir el dispositivo utilizado. En este caso “IDT High Definition Audio CODEC” y las interfaces, el resto desactivarlas para que no consuman recursos. En la figura 103 se ha marcado como interfaz los altavoces “Speaker”.

3. Tamaño del buffer, el parámetro más decisivo de optimizar. Un valor muy bajo (64 o 128) nos da menor latencia, pero a costa de mayor consumo de CPU. Hay que alcanzar un compromiso entre latencia/rendimiento. Si cuando tengamos todo configurado al tocar la batería tenemos retardo hay que bajar este valor. Si se percibe que los sonidos se cortan bruscamente y se pierden sonidos es que el valor es demasiado bajo.
4. Compensación para la latencia. Sino conocemos las especificaciones hardware de nuestra tarjeta el parámetro a 0.
5. Opciones extra de la tarjeta. Habilitar el modo evento; buffer offset menor de 10ms mejora estabilidad. Y submuestrear a 44,1Khz y forzar a una profundidad de bit de 20 a 16 mejora el procesamiento de audio para tarjetas con codecs AC97 (la inmensa mayoría de los procesadores de Intel).



*Figura 103: Panel de control de ASIO4ALL*

## 3.6 Interfaz gráfica, controles de teclado

Los objetos que diseñan los ingenieros deben ser sensibles a los seres humanos. Deben poder verse, oírse, tocarse...

Por esa simple premisa no pueden ser estáticos y el disparador MIDI deberá poseer unos parámetros modificables por el usuario. A esto anterior habrá que añadirles, por tanto, una pantalla y una serie de dispositivos táctiles y manejables.

Se continúa con el diagrama de bloques de la figura 50. En esta ocasión centrándose en los bloques “Interfaz gráfica, control de teclado y memoria”. Se creará un menú en una pantalla con distintas opciones y será el usuario el que interaccione con ellas. En un principio se utilizará para modificar los parámetros del capítulo 3.5.3 pero está abierto a ampliaciones.

Además será desgranada la utilización de la memoria de parámetros personalizados del usuario y el diseño de la carcasa que soporta todos los dispositivos electrónicos.

### 3.6.1 Interfaz gráfica

Existen distintos tipos de visualizadores en los que elegir. Basta un vistazo de los productos de cualquier distribuidor de electrónica (Digikey, Mouser, Farnell) para saber que los hay disponibles de todas las características y tamaños. Y es que el mundo de la optoelectrónica ha avanzado a pasos agigantados con la introducción de la tecnología LED en todas sus vertientes (OLED, AMOLED...), al igual que el cristal líquido (LCD, TFT...). Lejos quedan esas primeras bombillas detrás de carteles con números que seleccionaban las plantas por las que el ascensor iba pasando.

Todas las pantallas tienen en común que son dispositivos de salida y recibirán la señal de video por un bus de datos, pero ahí terminan las coincidencias, pues según los fabricantes nos podemos encontrar distintas conexiones y distinto funcionamiento. Empezando porque la señal puede ser tanto analógica como digital.

Entonces hay que valorar una serie de criterios de decisión:

- Caracteres o gráficos: En un primer pensamiento sería positivo la posibilidad de introducir en las pantallas imágenes que contextualicen la acción que se realice. Un dibujo de una batería, un mapa de bits de la acción de golpear un tambor o, simplemente iconos fácilmente identificables por el usuario. Pero

ello encarece el desarrollo. Se decide visualizar exclusivamente mensajes con caracteres cortos y que no den pie a confusión.

- Tensión de alimentación continua de valores a elegir entre 5V y 3,3V. Estas tensiones son reguladas por la plataforma de desarrollo.
- Bajo coste de adquisición y fácil reemplazo: Existen pantallas con una calidad de imagen muy precisa pero su coste se dispara en cientos de euros. Además debe ser fácil de adquirir ya que no se debe comprar un prototipo sino un producto comercial con cierto bagaje.

La solución que se adapta, que se encuentra dentro del equilibrio expuesto, es el módulo HITACHI HD44780 con bajo coste, muy utilizado en este tipo de desarrollos y robusto.

### **3.6.1.1 Hitachi HD44780**

El dispositivo Hitachi HD44780 es un controlador para una pantalla LCD de 16X2 (16 columnas por dos filas). La principal ventaja en el uso de estos displays es que poseen un microprocesador interno que realiza la tarea del refresco de los caracteres en pantalla, tarea que asume tiempo de ejecución de programa. Resulta fácil de conseguir y su precio oscila los 6€. Un punto excelente a su favor es que su conexión y codificación se ha convertido en norma para el resto de módulos LCD.

Las características son: [Whe11]

- Alimentación variable de 2.7V a 7V. Consumo muy reducido, del orden de 7.5mW
- Pantalla de caracteres ASCII, además de otros caracteres (japoneses, griegos y matemáticos). Permite que el usuario pueda programar 8 caracteres.
- Desplazamiento de los caracteres hacia la izquierda o a la derecha
- Memoria de 40 caracteres por línea de pantalla, visualizándose 16 caracteres por línea
- Movimiento del cursor y cambio de su aspecto
- Pueden ser gobernados de 2 formas principales, conexión con bus de 4 bits o con bus de 8 bits

### 3.6.1.2 Conexión



Figura 104: Pines del módulo LCD Hitachi HD44780 [Dardu]

Observando la figura 104 se conectarán las alimentaciones y la retroalimentación a +5V y a masa. La retroalimentación (backlight) mejora el contraste. Se conectarán 4 pines del microcontrolador entre Data 4 y Data 7, simplifica la conexión por bus de 4 bits. RS (selección de control de registro de datos y control) y E (habilitación del LCD) irán conectados a sendos pines también para su control. R/W irá conectado a masa al ser dispositivo de salida.

Por último Ve saldrá de un tercer terminal de un potenciómetro multivuelta entre 5 V y masa para realizar el contraste. El valor será de 10K y se ajustará manualmente con un destornillador.

### 3.6.2 Controladores de teclado

Se elegirá un teclado en el que el usuario se sienta cómodo y pueda seleccionar los parámetros según sus preferencias.

Las posibilidades estudiadas son variopintas. Un simple pulsador que ataque a un pin digital, un encoder rotatorio para incrementar y disminuir, un teclado hexadecimal y un teclado táctil capacitivo.

En la tabla 18 se realiza una evaluación general de varios de ellos. Destacando aspectos esenciales como si necesitan algún circuito adicional (ejemplo circuito anti-rebotes), el precio, su facilidad de uso por parte del usuario (Usabilidad) y su codificación por parte del desarrollador (facilidad de integración). Por último si hace falta añadir algo más se colocará en observaciones.

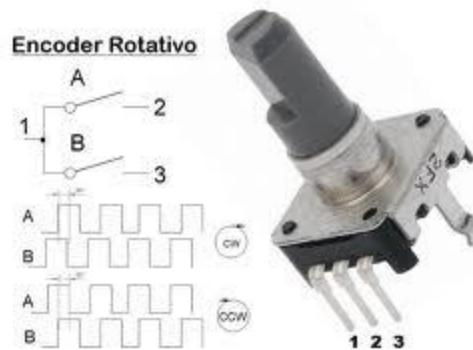
|                    | Circuito Auxiliar | Usabilidad | Codificación | Interrup. | Precio | Observ.          |
|--------------------|-------------------|------------|--------------|-----------|--------|------------------|
| <b>Pulsador</b>    | SI                | Baja       | Baja         | 1         | Bajo   |                  |
| <b>Encoder</b>     | SI                | Alta       | Media        | 2         | Bajo   |                  |
| <b>Hexadecimal</b> | SI                | Muy baja   | Media        | 4         | Medio  |                  |
| <b>Capacitivo</b>  | NO                | Media      | Alta         | 0         | Alto   | Requiere espacio |

*Tabla 18: Comparación de diferentes interfaces táctiles*

Los criterios de decisión primordiales son la usabilidad y el que ocupen poco espacio. El término usabilidad, en inglés sería similar a user-friendly , se refiere a la facilidad con la que el ser humano podría utilizar la herramienta, simplificando todas las posibilidades y centrándolas en las que de verdad se necesitan. También se incorpora en la tabla 18 el que la codificación no sea muy tediosa. Por ello la elección ha sido el Encoder.

#### 3.6.2.1 Encoder

El codificador rotatorio (rotary encoder) es un dispositivo electromecánico. En él se producen una serie de cambios de estado al girar. En la figura 105 se aprecia que es visualmente parecido a un potenciómetro pero su funcionamiento es totalmente distinto, en primer lugar porque no tiene fin de recorrido.



**Figura 105: Encoder [Dneot]**

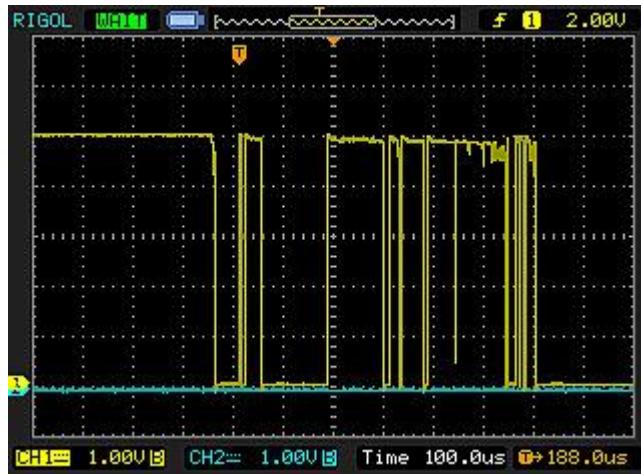
En la figura 105 también se observa el esquema eléctrico y las formas de onda que se producen en su movimiento. Está formado por contactos y se usa para valores numéricos que se incrementan y reducen al ser distinguible los diferentes estados que transita tanto para movimientos en sentido horario como anti horario.

Analizando y observando la forma de onda en binario de la secuencia AB(figura 105), siendo B el LSB tenemos:

- En sentido horario pasa por la secuencia 00->01->11->10. En decimal sería 0, 1, 3, 2.
- En sentido anti horario la secuencia de estados es inversa 00->10->11->01. En decimal tenemos 0, 2, 3, 1.

Además se encuentra una variedad de este dispositivo que incorpora un conmutador mecánico que se activa presionando la perilla y es una magnífica manera de seleccionar.

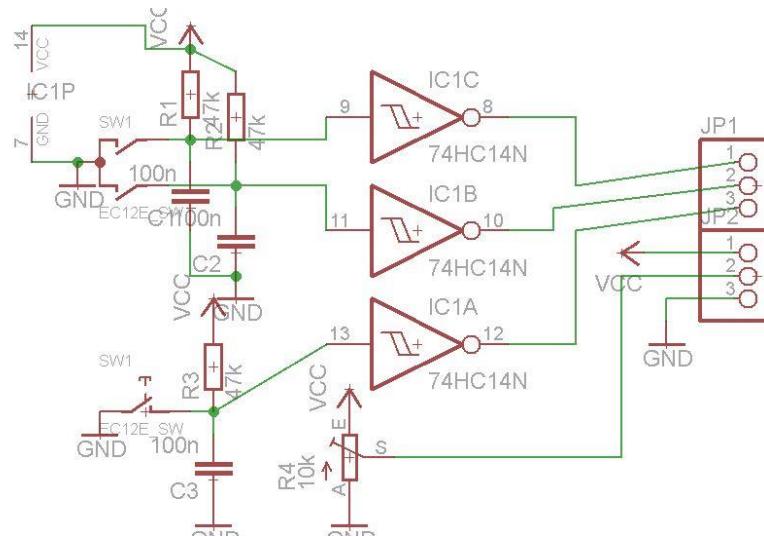
Ahora bien existe un problema de estas señales digitales y son la cantidad de meta-estados que provoca incertidumbre. En la figura 106 no se observan los cambios con claridad y eso provocaría errores en la decisión. Hay demasiadas fluctuaciones.



**Figura 106: Problemática de rebotes en encoder**

La solución es incorporar el circuito antirrebotes de la figura 107.

El circuito para cada fuente problemática incorpora una resistencia de pull-up, un condensador y un inversor Schmitt Trigger. Las fuentes problemáticas son el SWA y SWB del encoder y el pulsador incorporado también en el encoder.



**Figura 107: Circuito anti-rrebotes para encoder**

### 3.6.2.2 Programación mediante interrupción

Se ha visualizado con la figura 105 que el encoder transita por una serie de etapas que deben ser reconocidos por el procesador. La primera técnica que se plantea es que el propio procesador se encargara de sondar los dispositivos cada cierto tiempo para averiguar si tenía pendiente alguna comunicación para él (polling). Este método presenta el incon-

veniente de ser muy ineficiente, ya que el procesador consumía tiempo en realizar esa tarea que no es prioritaria.

La otra opción es por interrupción. La interrupción es una señal recibida por el procesador, indicando que debe parar el curso de ejecución actual y pasando un código específico para tratar esta situación, la rutina de atención a la interrupción o ISR.

Las interrupciones permiten al procesador desentenderse y delegar en el periférico la responsabilidad de comunicarse con el procesador cuando lo necesite activando la señal de interrupción.

En el microcontrolador Atmega existen 23 fuentes de interrupción. [Atm09] de las cuales para esta función se utilizan 3 que son activadas por flanco en los pines de conexión.

### 3.6.3 Menú de usuario

El código codificado se puede abstraer en una serie de estados (figura 108).

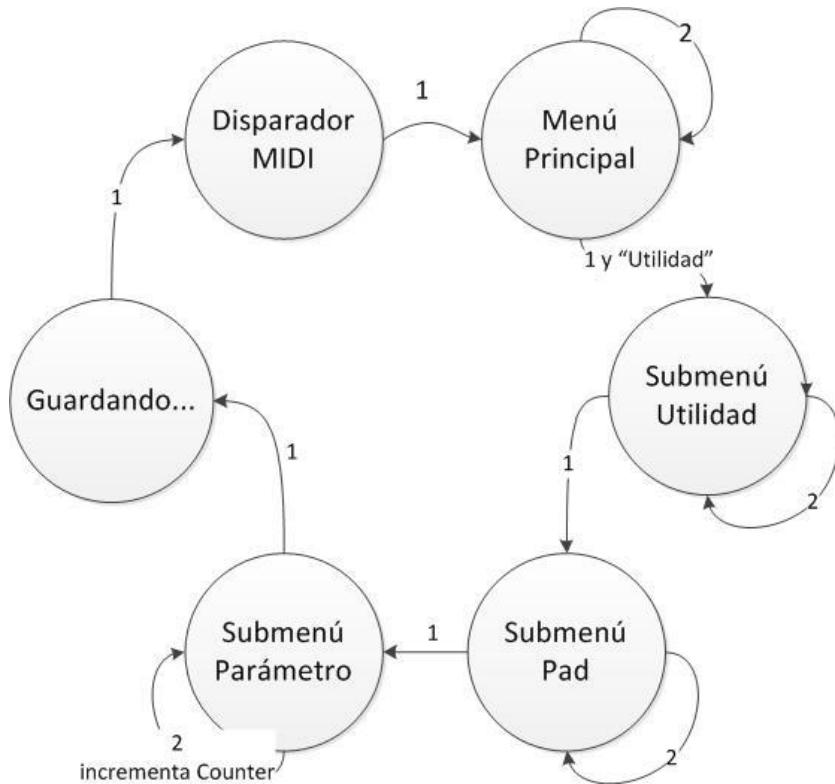


Figura 108: Diagrama de estados del menú

El objetivo es que el usuario seleccione una utilidad, un pad y un parámetro específico. A los diferentes estados se accede accionando el pulsador del encoder, señal 1 en el diagrama de estados de la figura 108. La señal 2 se refiere a elegir entre diferentes opciones, moviéndose a izquierda y derecha con el encoder.

Tras la inicialización se parte del estado inicial de “Disparador MIDI”. Tras la pulsación se accede a la interrupción y se cambia de estado a “Menú Principal”. En este estado es posible seleccionar entre las opciones “Utilidad”, “Créditos” y “Salir” con el encoder. Elegido “Utilidad” se pulsa y se introduce el “Menú Utilidad”.

En el estado “Submenú Parámetro” el programa deberá leer de la eeprom y en “Guardando...” escribirá en esta memoria. Ver el siguiente apartado.

Son opciones muy sencillas con pocas posibilidades y con una estructura de estados en forma de circulo cerrado que no permite errores.

### 3.6.4 Memoria de parámetros

En el menú se actualiza una tabla que se encuentra en la memoria EEPROM. Esta es una memoria de datos no volátil muy útil en estos casos pues al iniciar el sistema tendremos la configuración personal en lugar de programa base que se guarda en la flash.

Los parámetros han sido analizados en el apartado 3.5.1. Como ejemplo los parámetros óptimos en el pad de goma de la caja se muestran en la tabla 19 así como los efectos que se consiguen colocando los parámetros al máximo de su rango; 255.

| Parámetro        | Rango | Normal en Caja | Maximo efecto con ese rango           |
|------------------|-------|----------------|---------------------------------------|
| <b>Threshold</b> | 0-255 | 20             | No detecta valores inferiores a 1,3 V |
| <b>Gain</b>      | 0-255 | 2              | No respuesta dinámica. Todo 127       |
| <b>ScanTime</b>  | 0-255 | 13             | Fallo en golpes rápidos               |
| <b>MaskTime</b>  | 0-255 | 64             | Fallo en golpes rápidos               |

*Tabla 19: Parámetros recomendados en el pad de caja conectando el prototipo*

### 3.6.5 Carcasa protectora

Durante buena parte de la duración del proyecto el sistema se trasladaba de un lado a otro en una caja de cartón amplia en el que se colocaba fácilmente los sistemas y la proto-board.

La fiabilidad dejaba mucho que desear. Eran muchos los momentos en los que se revisaban conexiones. Para evitar esto se pensó en instalarlo en una caja de metal. El resultado fue el de la figura 109. No sin problemas de preparación por la dificultad de tratar el metal. Se agujereo con un taladro de mesa, se lijó manualmente y se pulió con una máquina Dremel. Se finalizó con una capa de pintura metálica roja.

La caja de metal era reciclada de otro proyecto y de un tamaño de 20X12 (cm). Esta limitación provocó que el espacio se amolde a 6 entradas Jack mono y tuviera que ser recortada por un extremo.



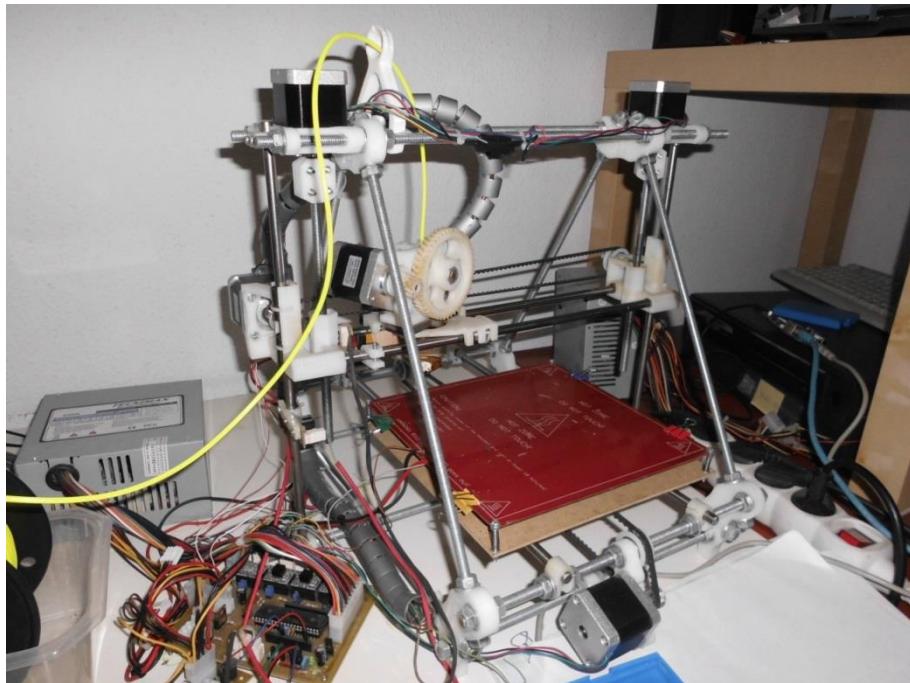
*Figura 109: Prototipo completo en caja de metal*

Tras este prototipo se pensó que fuera posible ampliar el sistema y por tanto la carcasa. Pero dado que las cajas son de tamaños específicos y lo engorroso de trabajar con metal se pensó en modelar con el fin de utilizar una impresora 3D y sus numerosas ventajas. A partir de aquí es una descripción teórica de los pasos a seguir.

Las impresoras 3D son máquinas capaces de reproducir cualquier objeto del entorno. [Est12]

En este proceso se utilizaría el modelo Prusa I2 que se puede encontrar en la Asociación de Electrónica de Telecomunicaciones de la UPM (AETEL) para uso y disfrute de sus socios. (figura 110).

El mecanismo de funcionamiento de la prusa es superponer distintas capas del material plástico ABS hasta alcanzar la forma deseada. Desde un ordenador se le mandan los comandos adecuados y se mueve el cabezal hot-end hacia la posición deseada donde se aplica el plástico.



**Figura 110: Impresora 3D modelo Rep Rap prusa i2**

Se partiría de un fichero software que simula un objeto (la nueva carcasa). Se crea a partir de programas de modelaje entre los que se elegiría el google Sketch Up por ser una alternativa libre muy completa.

El SketchUp es un programa de diseño y modelado gráfico en 3D. Con una barra de herramientas muy intuitiva a la par que potente. En SketchUp se pueden seleccionar distintas formas, orbitar en los 3 ejes, cambiar tamaños... Es compatible con multitud de formatos de ficheros como el autoCAD.

Una vez realizado el diseño se necesitaría un plugin adicional denominado “export to STL”. Obteniendo el formato vectorial requerido .stl para el siguiente paso.

El siguiente paso sería utilizar el Slicer convirtiendo de nuevo el fichero a otro que si entiende la impresora; el formato Gcode. Se introduciría ese fichero en el programa Pronterface que controla la comunicación con la impresora y los motores. Con este último paso la impresora se dispondría a construir la carcasa. [Fer12]

### 3.7 Controlador del charles

Se llama controlador del charles (Hi-Hat controller en inglés) al componente de una batería electrónica encargada de la gestión de una serie de acciones sobre el pad denominado charles. Se encuentra en forma de pedal para el pie izquierdo y emula el herraje y parte del plato de una batería acústica.

En la figura 111 se vislumbra el herraje que sujeta los dos platillos de una batería acústica. El herraje es un trípode de acero que permite accionando el pedal controlar el mecanismo de chocar los platillos y distintos grados de apertura. El baterista utilizará ese pedal para crear gran variedad de sutilezas. [Stao5] Además se observa la palomilla que ajusta la apertura máxima de los platillos.



*Figura 111: Herraje en batería acústica y su imitación en la electrónica*

A la derecha del herraje de la figura 111, el controlador del charles adoptado por fabricantes de baterías electrónicas. La evolución sigue teniendo forma para el pie pero la construcción es plástico y en el interior dispone de un circuito electrónico.

Analizando las necesidades de un batería se necesita un dispositivo que marque una serie de señales a cada una de las siguientes acciones:

- Si no se presiona, el pedal estaría levantado por lo que si se golpea el charles además de enviar intensidad del golpe el disparador debería ser consciente de

que la nota en MIDI a enviar sería la de un charles abierto Nota=42. Si por el contrario el pedal está presionado el disparador debe enviar, cada vez que se golpee el pad de charles la Nota=46. Los módulos de gama alta envían notas entre medias. Por consiguiente una señal a tratar será la posición del pedal en ciertos instantes.

- El cierre del hihat al presionar. Independientemente de que se haya golpeado el pad. Lo que comúnmente se denomina “chic”. También debería tener sensibilidad a la presión. Por consiguiente se requiere una señal que varíe con la presión ejercida.

Tras la inspección de multitud de baterías y sus correspondientes pedales se recae en que existen multitud de formas para controlar esa apertura: Potenciómetro, LDR, Efecto Hall, interruptores... La precisión con la que se ejecuten determinará su complejidad.

### **3.7.1 Solución 1: Pedal DD508**

Se utiliza el pedal comercial de la batería DD508 que aparece en la figura 111 para evitar la dificultad en la construcción mecánica de un pedal similar. Este pedal es análogo al utilizado por otras marcas (Yamaha HH65 y Roland FD8). [Dhell]

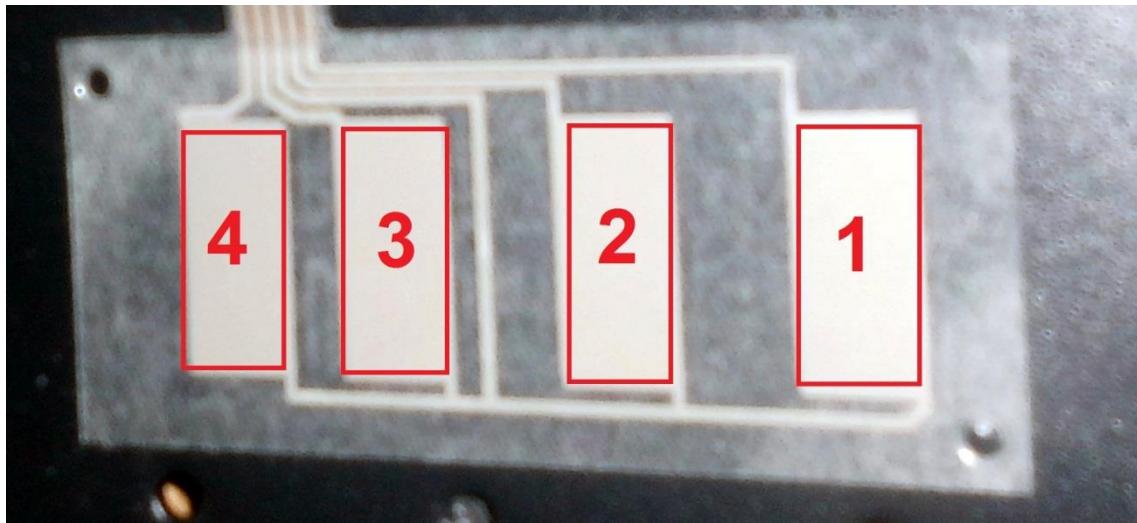
#### **3.7.1.1 Proceso de análisis**

Se parte de la suposición de que las salidas del pedal conmutan al accionarlas, funcionando como un pulsador. Suposición incorrecta comprobada con un multímetro.

Se procede al desmontaje del pedal y se descubre una serie de contactos eléctricos tal y como se señala en el esquema de la figura 113. Con el multímetro se identifican las pistas y los dispositivos.

El circuito está formado por commutadores impresos recubiertos por un plástico de protección. En la figura 112 aparecen y son numerados para identificarlos tanto aquí como en el esquema eléctrico. Si la goma que está adherida al pedal presiona cada uno de ellos cierra el circuito de la figura 113. Si por ejemplo se presionase fuertemente se cerrará el commutador S4 y la tensión de salida será 3.3V.

Se analiza el circuito y simulación en Multisim con análisis DC. Son divisores de tensión regulados por la presión del pedal en 4 posiciones. Ver tabla 20.



*Figura 112: Fotografía interior del pedal HH65*

Se utiliza el convertidor ADC puesto que la tensión toma 5 valores a lo largo del tiempo. Diseño del programa para identificar los códigos del ADC para cada transición. Ver tabla 20 y la fórmula siguiente:

$$LSB = V_o * \frac{V_{ADC}}{2^n} = V_o * \frac{5}{1024}$$

Sabiendo que señales envían se utiliza un pedal y el pad de charles que viene con ese módulo. Obtención del SW/algoritmo. El algoritmo en el uso de este pedal no es inmediato por lo que se efectúan una serie de pruebas con el módulo del fabricante.

Elaboración de un programa que realice el algoritmo del módulo (figura 114).

Testeo con el modulo corroborando que posee sensibilidad en el chic puesto que utiliza la transición anterior para delimitar si el recorrido efectuado es amplio (nota enviada más fuerte) o no.

| Estado | Análisis en Multisim (VDC) | Recogido en microcontrolador (LSB) | Recogido en microcontrolador (VDC) |
|--------|----------------------------|------------------------------------|------------------------------------|
| NP     | 0                          | 0                                  | 0                                  |
| 1      | 1,39                       | 290                                | 1,42                               |
| 2      | 2,39                       | 500                                | 2,45                               |
| 3      | 3,09                       | 646                                | 3,16                               |
| 4      | 3,3                        | 688                                | 3,4                                |

*Tabla 20: Tensiones y códigos del pedal de HH a diferentes pulsaciones*

### 3.7.1.2 Esquema eléctrico

Las diferentes opciones de pulsación se recogen en la tabla 20.

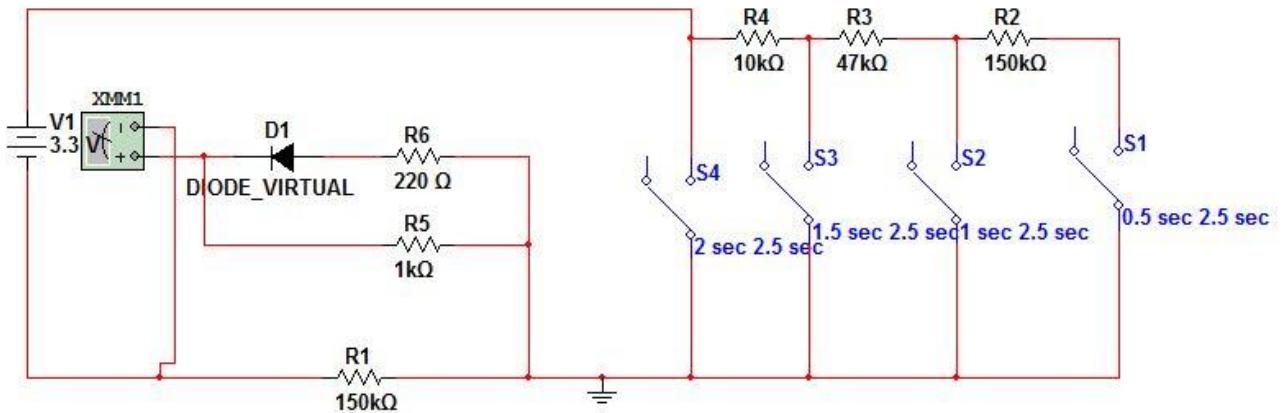


Figura 113: Esquema eléctrico con Multisim de pedal comercial

### 3.7.1.3 Algoritmo

El algoritmo, figura 114 actuaría en 2 partes que se ejecutarían en concurrencia.

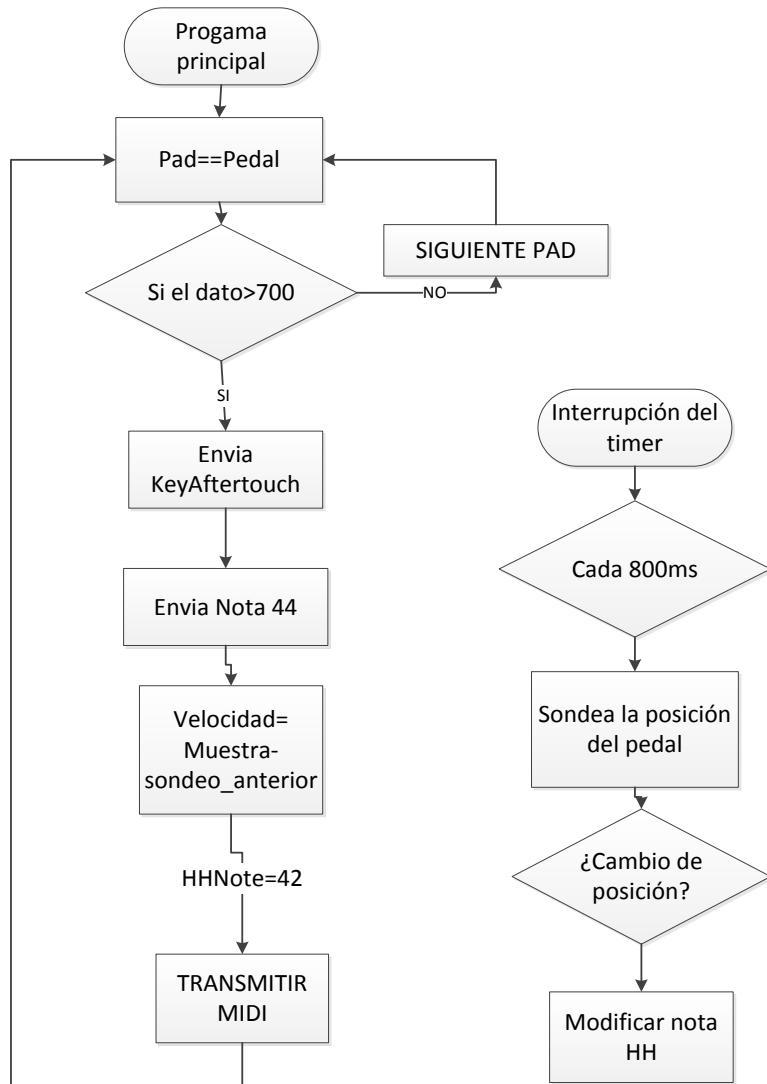
En el programa principal, tras sondear el resto de pads, se mandaría al ADC la acción de que sondeara también el pedal.

Si el dato recogido en el pedal es superior al número binario 640, aparecerá el estado 3 de la tabla 20 y por tanto ha habido una pulsación fuerte. En ese momento se enviará vía MIDI la nota 44 correspondiente al pedal y la intensidad. A partir de este momento la nota enviada y transformada por el pad de plato asociada será el de HHcerrado que es el número decimal 42.

Por otro lado estará el timer 2, configurado su interrupción por overflow cada 64 ms. Dentro del ISR, cada 800 ms sondeará la posición del pedal almacenando si hay un cambio de estado.

Como hemos dicho esto es la suposición basada en un razonamiento y la experiencia de haber trabajado con el módulo original. También podría darse el caso de que estuviera capturando los valores iniciales y finales para calcular su pendiente con el tiempo transcurrido.

$$\frac{V_f - V_i}{T_f - T_i} = \text{Parámetro velocidad}$$



*Figura 114: Diagrama de flujo algoritmo HH65*

### 3.7.2 Solución 2: Pulsador

La primera propuesta es recurrir a la distancia entre 2 contactos que denominaremos C1 y C2 en un tiempo dado, midiendo así la velocidad que transcurre entre ellos. La transición entre que se pulsan los dos debe ser tenido en cuenta por el sistema para decidir la presión que se ha ejercido tal y como sería en una batería acústica la apertura y cierre del pedal.

Estos contactos atacarían a sendos puertos digitales del microcontrolador que serían leídos para constatar la posición del pedal. La circuitería que acompaña al pulsador en la figura 115 es un clásico circuito anti-rebotes. Si no se pulsa el sistema debe entender que las notas producidas al presionar el Hihat debe ser la nota abierto pues se mantiene la conexión C1. Y se enviará la nota cerrado cuando se accione la línea asociada a C2.

Además nos permite la posibilidad de expandir el sistema a un tercer estado que sería el hi-hat semiabierto que es muy utilizado, al poseer un sonido característico. Tan solo con añadir otro pulsador.

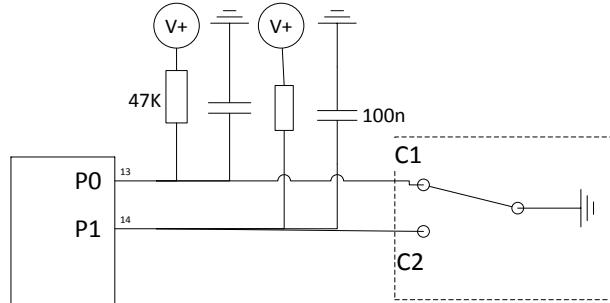


Figura 115: Diagrama de la conexión de los contactos del pedal de HH

Utilizar un pulsador SPDT (Single Pole Dual Throw), mostrado en la figura 115 con los contactos C1 y C2, provoca irregularidades mecánicas se producen rebotes también llamados meta estados. Estos meta estados producen inestabilidades que deben ser eliminadas para un funcionamiento algorítmico completo.

#### 3.7.2.1 Tiempo entre transiciones

La dinámica en la pulsación del pedal de batería vendrá dependiente del tiempo que transcurra entre contactos. Los tiempos dispondrán al sistema el valor (entre 0 y 127) que debe tomar.

La forma en la que se manejan estos valores de velocidad no está especificada en la norma puesto que no se asignan los valores de tiempo que corresponden al rango dinámico. Es pues que se dispone de total libertad.

Por otro lado utilizar un timer del microcontrolador para esta función. Lo configuraremos para que se produzca desbordamiento cada 16 ms (máximo preescalador) y utilizando su interrupción para elaborar la estrategia.

En un principio se marcaron valores estables para tiempos conocidos con el timer. Estableciendo unos rangos de velocidad para blanca=60bpm y para ese mismo tempo, negras corcheas y semicorcheas. Esto producía una sensibilidad no lineal.

Tras analizar que con un valor máximo conocido lo mejor era disminuir el valor acor dado dado el tiempo y realizar una función lineal. Esto es el algoritmo que finalmente se eligió.

### 3.7.2.2 Algoritmo

El primer algoritmo queda perfectamente explicado con esta captura de pantalla.

```
pulsador1=digitalRead(12);
if (pulsador1==0) { //Si se pulsa el pedal viniendo de reposo
    delay(20);
    pulsador2=digitalRead(11);
    if ((pulsador2==0) && (pulsado==1)){//Con un metrónomo bpm 60
        delay(20);
        Note[2]=42;//Nota charles cerrado
        if(ticks>64)                                //Blanca
            velocidad=60;
        if ((ticks<64)&&(ticks>32)) //negras
            velocidad=90;
        if((ticks<32)&&(ticks>16)) //Corcheas
            velocidad=100;
        if((ticks<16)&&(ticks>1)) //Semicorcheas
            velocidad=127;
        pulsado=0;
        ticks=0;
        notaOn(9, 44, velocidad);
```

*Figura 116: Algoritmo pedal solución 2*

En el segundo modificamos los tiempos haciéndolo lineal con esta tabla en lugar de ser 4 y totalmente estáticos.

### 3.7.3 Solución 3: Sensor FSR

#### 3.7.3.1 Introducción

En la solución 1 no se terminaba de conseguir un funcionamiento completo, ofrecer sensibilidad al presionar el pedal. En la solución 2 se comprendía que existía un problema de materializar el concepto puesto que la mecánica y la inexperiencia iban en contra.

Esta solución intenta aunar los problemas de la solución 2: Facilidad de construcción y sensibilidad. Además de una gran precisión.

Además permite integrarse en un auténtico herraje de batería acústica. Mejora las sensaciones y la estabilidad de la batería.

Se utiliza un sensor resistivo de fuerza o FSR (Force Sensing Resistor). Utiliza la propiedad eléctrica de la resistencia para actuar como sensor de fuerza. La película resistiva va debajo de una cinta conductora que al presionar entra en contacto aumentando la conductividad entre los extremos y disminuyendo la resistencia. Como podemos ver en la figura 117.

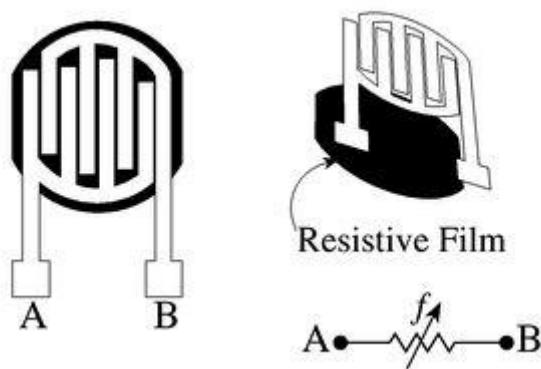


Figura 117: Gráfico y símbolo de sensor FSR [Dstan]

No es adecuado para mediciones precisas de fuerza pues posee un error del 15% [Palo3]. Es la solución utilizada en los módulos de alta gama para realizar la función de parada de platos. Además de servir para producir nuevos sonidos. Cuando se golpea con la baqueta, envía una señal que combinada con la del piezoelectrónico, informa de que tiene que reproducir un nuevo sonido.

### **3.7.3.2 Espuma resistiva**

Los sensores FSR no son baratos. Puedes encontrar películas resitivas por el precio de 15€ en algunas tiendas de electrónica.

Siguiendo con la mecánica de este proyecto se intentó ver alternativas. Se encontró su construcción con algo tan común entre los electrónicos como PCB sin atacar, pegamento, espuma conductora y herramientas fáciles de encontrar.

La espuma conductora viene con frecuencia en el embalaje donde se ubican los micro-controladores. Es un descubrimiento utilizar una propiedad especial de algo que parecía únicamente para proteger. [Heno6] Es especial y tiene un granulado característico por lo que no todas las espumas valen. Además de que las características difieren entre unas y otras. Por tanto dificulta su estudio.

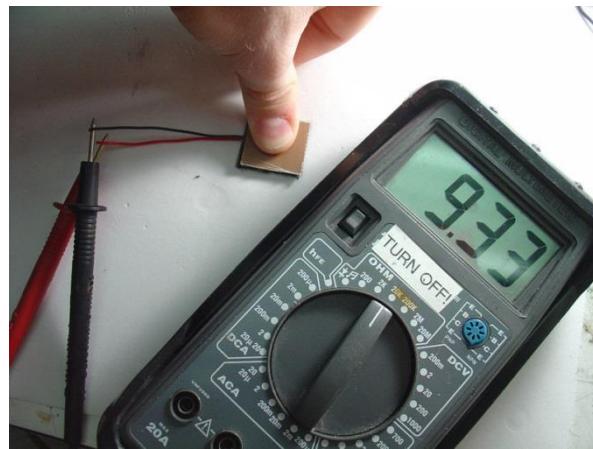
Construcción del sensor y colocación del herraje:

- 1) Cortar con la cizalla dos cuadrados de la PCB de por ejemplo de 2 cm de lado. Con un cúter cortar la espuma a esa misma anchura.
- 2) Soldar un cable a cada uno de los cuadrados de cobre.
- 3) Pegar las 3 piezas con el pegamento. Dado que el pegamento es aislante utilizar la mínima cantidad de ello pegando solo un lateral. Este paso se visualiza en la figura 118.



*Figura 118: Sensor FSR de construcción artesanal*

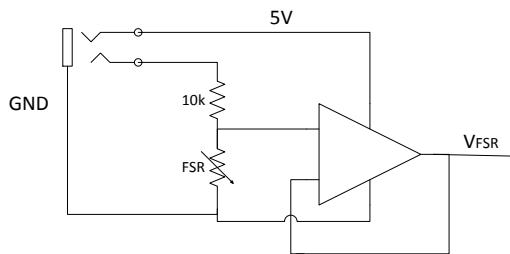
- 4) Realizar las pruebas con un polímetro de la resistencia máxima y mínima ante una presión. En este caso son 2,6k y 10k. Si la espuma fuera más grande los valores serían más altos y viceversa. Ver figura 119.



*Figura 119: Medida de la variación del sensor FSR*

### 3.7.3.3 Funcionamiento

Se elige una resistencia limitadora de 10K y se construye un divisor resistivo. Añadir un amplificador operacional en modalidad seguidor de tensión alimentado a 5V para adaptar impedancias al ser conectado con el microcontrolador y la impedancia de entrada del convertidor analógico-digital. El esquema se visualiza en la figura número 120.



*Figura 120: Circuito eléctrico FSR*

Se obtienen los distintos valores de prueba:

- Sin pulsar  $30M\Omega \approx \infty$  lo que equivale con divisor resistivo de valor 10K y alimentado a 5V a una tensión próxima a 5V.
- Tocando suavemente con el pie vararía entre 10KΩ y 25kΩ. La salida del seguidor de tensión rondaría entre 2 y 4 V
- Presionando con fuerza una variación de 2,6kΩ hasta 8kΩ lo que equivaldría de 0 a 2V.

### **3.7.4 Conclusión**

Por la sencillez del planteamiento se optó por la solución 2 y con un pequeño pulsador se hicieron las pruebas obteniendo los resultados esperados y la solución al controlador del HH.

Tras las pruebas con el pulsador se planteó realizar la construcción de un pedal en el que colocar los contactos eléctricos.

Una construcción sencilla sería reutilizando un pedal de una batería acústica y se soldarían 2 tiras de metal de forma cuadrada en el lateral. Un cable conectado a masa sería desplazado entre los contactos por el movimiento del pedal.

Se omitió la construcción de la propuesta al estudiar los problemas devenidos por la mala conexión entre el cable y las tiras. Además del rozamiento entre dichas piezas que impediría un manejo cómodo. En definitiva problemas mecánicos.



# 4 ELABORACION DE ENSAYOS Y RESULTADOS

---



## 4.1 Resultado final

En la figura 121 aparecen los componentes que se encuentran dentro del interior de la carcasa. Con el número 1 el circuito impreso que hace funcionar el encoder, elaborada con una placa fotosensible y siguiendo el esquema de la figura 107. Con el número 2 los conectores del módulo LCD. Con el número 3 el conector MIDI OUT. Con el número 4 las 3 placas de circuito acondicionador colocadas en forma de pila. Con el número 5 la placa de desarrollo del Arduino MEGA y concluyendo, con número 6, los 6 conectores Jack que se conectan a la batería electrónica DD508.

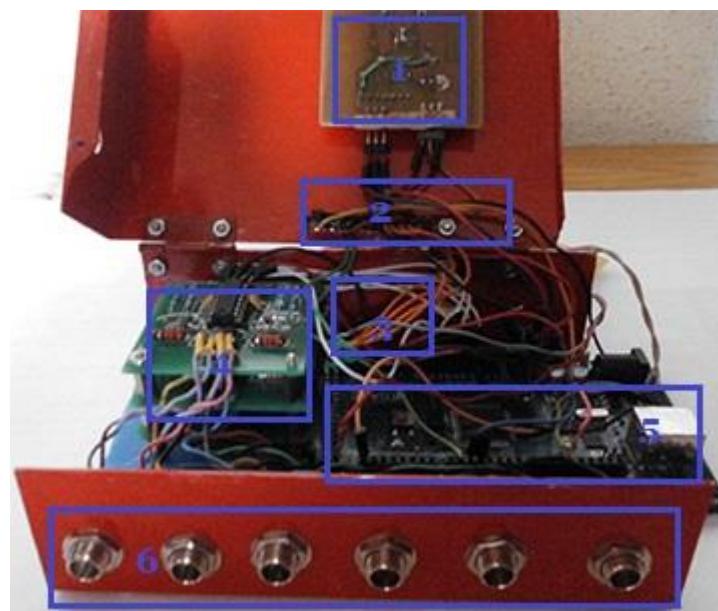


Figura 121: Vista interna del prototipo de disparador MIDI

### 4.1.1 Manual de usuario

En primer lugar se supone que el prototipo de disparador (figura 121) se encuentra con la última actualización firmware que se puede consultar en el anexo. A continuación se siguen las instrucciones:

1. Encienda el ordenador.
2. Conecte todos los pads con los conectores Jack de la parte delantera del dispositivo.
3. Conecte una de las alimentaciones. Puede elegirse por un lado 9V por el conector DC o bien conectando a un USB activo que provea alimentación, por ejemplo el ordenador encendido. Debe aparecer en la pantalla el mensaje “Disparador MIDI”.

4. Conecte la interfaz USB-MIDI en los dos extremos. Por un lado el conector marcado como MIDI IN al DIN-5 del disparador y por otro el conector USB al puerto USB del ordenador.
5. Efectúe el primer test. Golpee algún pad y observe como se enciende el LED amarillo de envío de datos. Si no ocurre esta acción vuelva a iniciar los pasos o vaya al apartado de preguntas frecuentes.
6. Ejecutar un programa de muestras de batería.
7. Seleccionar las muestras a reproducir
8. Seleccionar el mapeado-General MIDI
9. Todo correcto, empiece a tocar. En caso contrario reinicie el ordenador y comience de nuevo estas instrucciones.

## 4.2 Testeo del funcionamiento del prototipo

### 4.2.1 Test eléctrico y lógico

Se muestra el camino de las señales con el osciloscopio en las distintas partes señaladas en rojo. El test lógico se realiza con MIDI-OX observando la señal resultante.

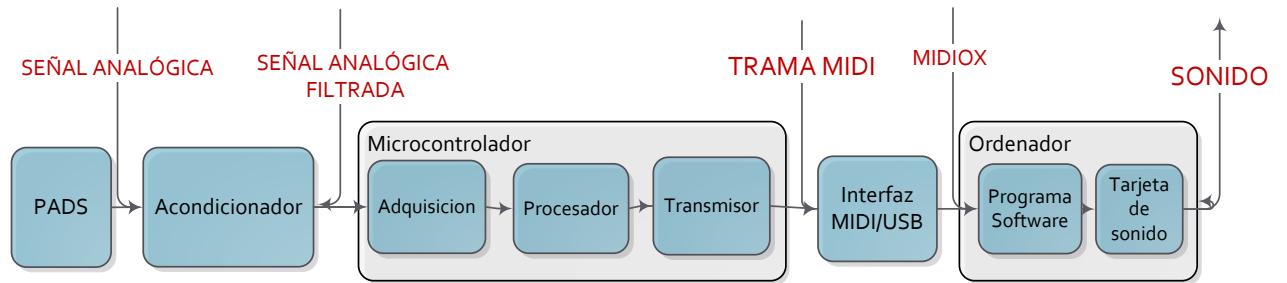


Figura 122: Puntos de test de las diferentes medidas

#### 4.2.1.1 Entrada y salida del acondicionador

Para un golpe de tom de baja intensidad se observa, con el osciloscopio, la salida del pad correspondiente y la salida del acondicionador. Orientarse con la figura 123.

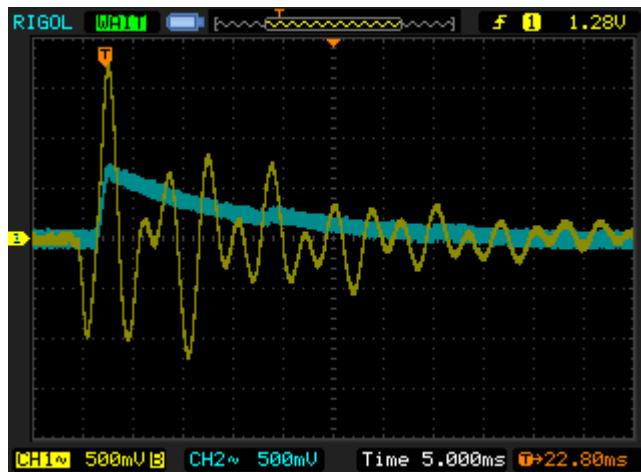


Figura 123: Prueba de entrada y salida del acondicionador



Figura 124: Comando recibido para prueba de entrada y salida del acondicionador

Tras la captura se observa satisfactoriamente que el prototipo atenúa la señal al nivel indicado y captura el máximo de la señal manteniéndolo durante el tiempo de escaneo (unos 2ms). A partir de ese instante cayendo exponencialmente.

Para ese valor máximo de unos 640mV el ADC lo transforma en el código 128. Pasa por el procesador y el algoritmo de los anexos. Primero el truncamiento con desplazamiento del apartado 3.2.3 y luego multiplicar por la ganancia de 4 elegida por el usuario. El dato de velocidad enviado por el protocolo MIDI es 64 como se observa en la trama MIDI capturada. En esa trama también aparece el número en decimal 153 correspondiente al comando Note\_ON y el primer byte de datos correspondiente a la nota pulsada que es 41 y según la tabla corresponde a un Goliat.

#### 4.2.1.2 Salida del transmisor

El siguiente paso es corroborar el buen funcionamiento de la UART. Se golpea el pad levemente. Para su decodificación seguir el ejemplo del apartado 2.1.6 y la figura 33. El resultado puede verse a nivel lógico en la figura 126.

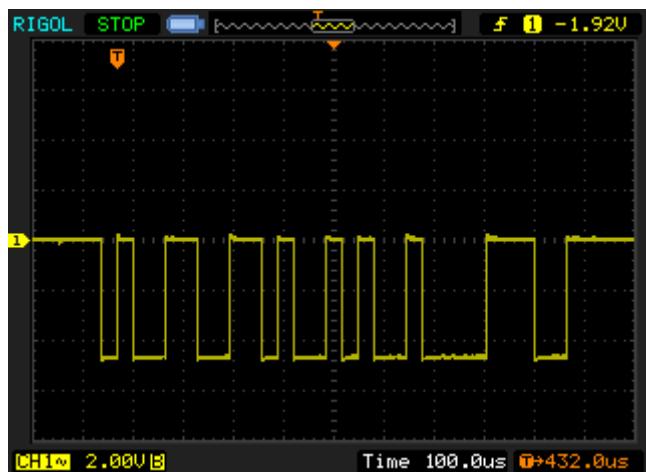


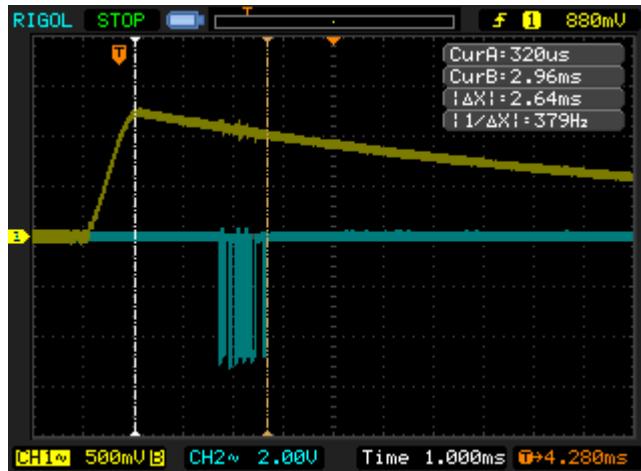
Figura 125: Prueba de UART

| TIMESTAMP | IN | PORT | STATUS | DATA1 | DATA2 | CHAN | NOTE | EVENT       |
|-----------|----|------|--------|-------|-------|------|------|-------------|
| 550987    | 1  | --   |        | 153   | 41    | 56   | 10   | F 2 Note On |

Figura 126: Comando recibido para prueba de UART

#### 4.2.1.3 Salida del acondicionador y salida del transmisor

El sistema digital necesita un tiempo en tomar muestras, procesarlas y mandar la trama completa. En la figura 127 se muestra que esta latencia es de unos 3ms.



*Figura 127: Latencia del disparador MIDID*

#### 4.2.1.4 USBPcap y Wireshark

Wireshark es un analizador de protocolos para redes de comunicaciones. Utilizado para el régimen didáctico y también en el entorno profesional para solucionar problemas. Es un programa de software libre.

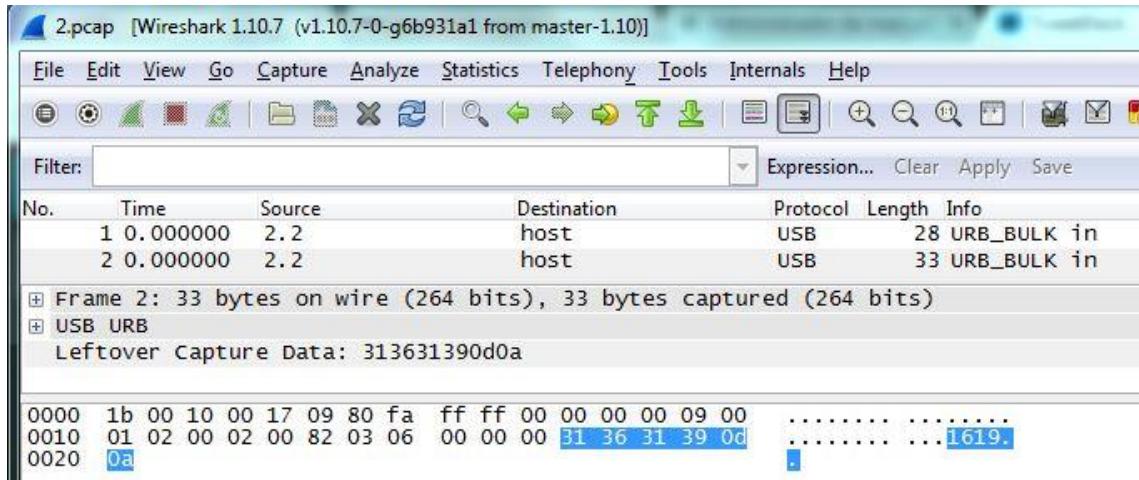
USBPcap es un plugin adicional del wireshark utilizado para capturar paquetes USB en el entorno Windows. [Ddeso]

Con estos dos programas se estudiará la comunicación producida entre los instrumentos musicales a través del conversor MIDI/USB visto en el apartado 2.1.5.

Las instrucciones que se siguen. [Ddeso]

1. Conectar USB y ejecutar el programa USBpcapCMD.exe.
2. Identificar la ruta para realizar la captura dependiendo de la conexión interna o si dispone de Hub. Indica que dispositivo se encuentra conectado ahí por lo que facilita el reconocimiento. Nominar la captura con “2.pcap”.
3. Capturar los paquetes haciendo las pruebas oportunas. Golpear una vez el pad y con ayuda del acondicionador, el ADC y el microcontrolador se captura el valor adecuado de la gráfica y lo envía por USB. Además se introduce en el algoritmo un número adicional de control y visualización; el 5.
4. Analizar los datos del fichero abriendo el programa Wireshark. En la figura 128 se observa la encapsulación que se produce en el protocolo USB con la información de cabecera y datos de control.

Se capturan dos tramas de 33 bytes (264 bits). En la primera se envía el valor 5. En la segunda se manda el máximo obtenido que es el número 1619 que ocupa 6 bytes y se encuentra marcado en la figura 128.



**Figura 128: Trama USB enviada por el disparador**

#### **4.2.2 Verificación con sampleador**

Una vez construido el disparador MIDI sabemos que todos los golpes sobre la superficie de los pads se convierten en tramas. Ahora, tal y como se explicó en el apartado de teoría, se requiere un sampler que las transforme en sonido.

El propósito será explicar la configuración de todos los programas y dispositivos para poder usar el prototipo de batería electrónica con el fin de disparar los sonidos incluidos en un PC. Utilizando programas instalables en cualquier ordenador de prestaciones medias.

Se han elegido 2 programas de muestras de batería, explicados en el apartado 1.2.4.3 por ser muy identificativos. Por un lado BFD2, un programa propietario mundialmente conocido y por otro DrumMic'a, un programa de licencia freeware. El software con licencia freeware permite el uso completo sin coste pero no permite ninguna modificación, es decir solo necesita un registro en su base de datos.

Se listará el material (hardware y software) que hace falta para conectar el disparador al PC. Luego se explicará cómo configurar los programas y samplers de baterías acústicas. Pero por supuesto existen otras opciones, otras configuraciones y lo que se explica aquí sirve para el resto de programas.

No se puede documentar de manera rigurosa las sensaciones auditivas por lo que se posterga a la presentación ante el tribunal.

##### **4.2.2.1 Material Utilizado**

1. Salida MIDI OUT del prototipo de disparador MIDI.
2. Interfaz MIDI-USB. Figura 27
3. Ordenador con al menos 2 gigas de RAM y 60 gigas de espacio libre en disco.  
Dar importancia a la tarjeta de sonido que será la que reproduzca fielmente el sonido. El ordenador utilizado en las pruebas es un portátil HP Probook 4540 con las siguientes características:[Dhpc0]
  - Windows 7 de 64 bits.
  - Procesador Intel Core i5.
  - SDRAM DDR3 de 4 GB.
  - SRS Premium Sound PRO.
  - 500 GB SATA.

4. Un sampler de batería. Existen muchos ejemplos como los analizados apartado 1.2.4.3
5. Un programa host para VST (Kontakt Cubase, FL Studio, ...). El BFD tiene opción de ejecutarse por él mismo por lo que sino interesa grabar, sólo tocar en directo, no es necesario otro programa adicional.
6. Drivers ASIO para la tarjeta de sonido. Si no tiene unos propios, que es lo más frecuente, hay que usar los drivers genéricos ASIO4ALL (Instalar estos drivers antes de instalar los programas) (Ver apartado 3.5.2.2)

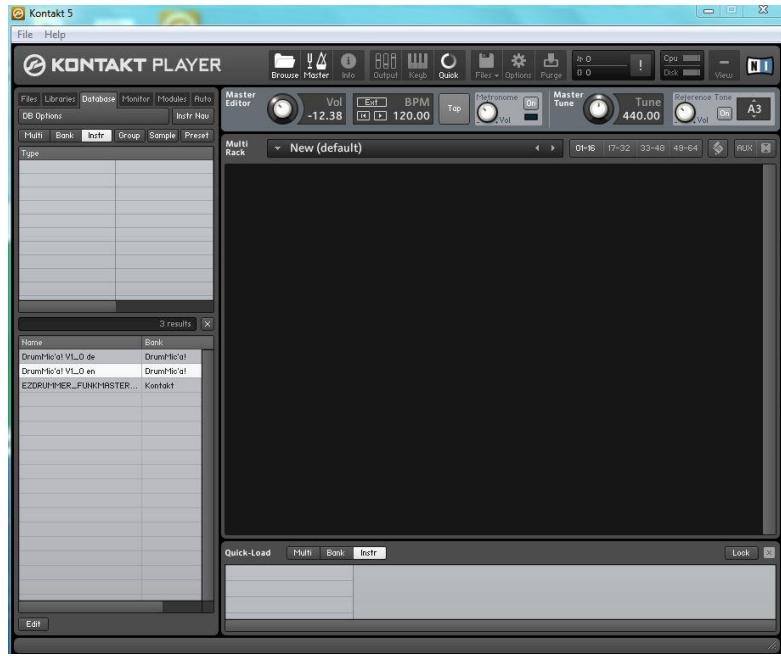
#### **4.2.2.2     *Configurar Kontakt 5 Player y DrumMic'a***

Kontakt 5 Player de Native Instrument es un programa de reproducción de samples de licencia freeware que no dispone de todas las prestaciones de su homólogo. [Dkont] Permite gestionar diferentes instrumentos virtuales. Existen otros ejemplos de pago (en torno a 200€) como el Cubase o el Ableton Live. Estos programas además de cargar instrumentos pueden hacer mezclado de canales de audio, secuenciar pistas para hacer canciones, añadir efectos (reverb, distorsión, amplificación...) a las canciones.

Tras instalar y ejecutar el programa tenemos el siguiente panel de control de la figura 129 con barras de herramientas propias del entorno Windows y otras más referidas al sonido.

Se procede con las opciones de audio. Acceder a File>Options>Audio. Seleccionar el ASIO driver de la tarjeta. El botón "ASIO config" permite configurar los drivers de sonido. Se eligen los drivers libres ASIO4ALL V2 que en el capítulo 3.5.2.2.1 se explica cómo configurarlos

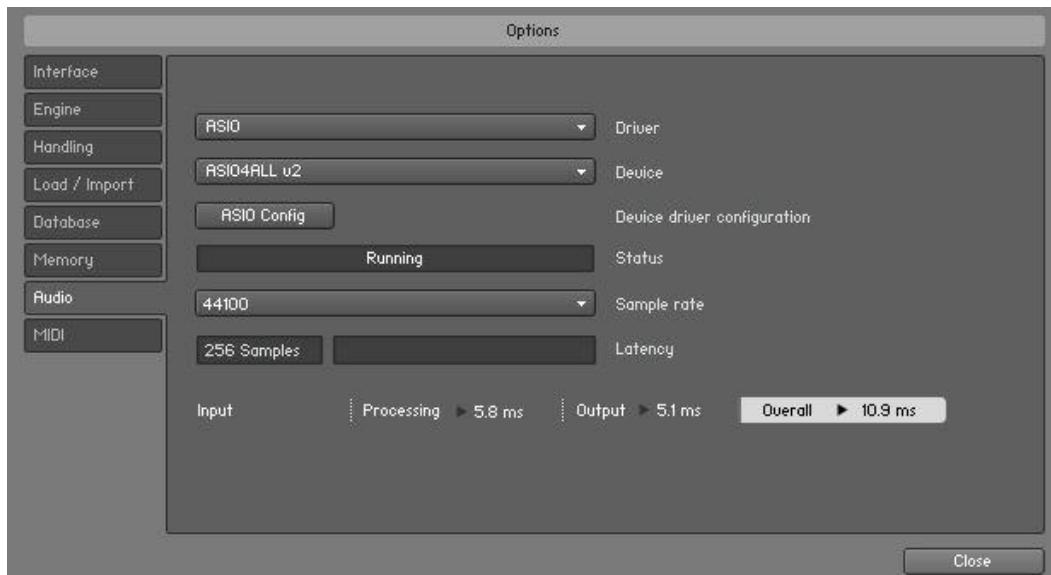
La configuración en Kontakt 5 debería quedar como en la figura 130. Queda configurado con una frecuencia de muestreo de 44,1khz y un bloque de 256 muestras. En ella se informa del tiempo utilizado en el procesamiento 5,8 ms así como el tiempo que tarda la tarjeta de sonido en reproducirlo a la salida 5,1ms. Después de configurar los drivers pulsar "Apply".



**Figura 129: Panel de Control Kontakt 5 Player**

Los plugins VST de este programa se instalan por defecto en C:\Usuarios\DocumentosPublicos\Kontakt5. Es importante recordarlo porque a la hora de instalar cualquier otro VST debe dirigirse a esta carpeta. Sino añadir otros directorios mediante File>Options>Database>Add [Dsenn]

Tas haber instalado DrumMic'a correctamente saldrá en la pestaña Database >Instruments para elegir. Cualquier instrumento virtual aparece en esa pestaña. Seleccionar y se dispone a cargar durante 3 ó 5 segundos y debería parecerse a la figura 17.



**Figura 130: Menú Configuración del audio en Kontakt 5 Player**

El programa habilita modificar cualquier micrófono de cualquier elemento de la batería pulsando sobre su imagen con ello conseguimos distinto comportamiento en frecuencia.

Además en el apartado option se dispone del parámetro “velocity” del sampleador que permite gestionar las capas de muestras de las que está compuesto el programa. En esa ventana se gestiona “MIDI mapping” que se encuentra seleccionado por defecto en GM.

Con todas estas configuraciones se probó satisfactoriamente el prototipo.

#### **4.2.2.3      Configurar BFD2**

La instalación de este programa es mucho más sencilla que la Drummic'a, ya que es guiada con un instalador. Una vez instaladas todas las muestras ejecutamos el programa. En unos segundos vemos la pantalla con un entorno gráfico amigable como la figura 131.

Comprobar que están seleccionados los drivers ASIO4all en I/O settings>Audio Devices>ASIO. La configuración de estos drivers la gestionamos con lo explicado en el apartado 3.4.2.2.

Con el conversor MIDI-USB conectado lo seleccionamos en I/O Settings>MIDI Port>USB.

Ahora se cargan una serie de muestras, en torno a un kit completo. Es posible elegir entre los que vienen preconfigurados o bien crear uno propio. En este caso se llama “Manuel's drum”. Se describe en detalle el kit, el porqué de su elección.

Los platos son una mezcla de platos brillantes (Crash 14" Sabian AAX y Crash 16" Zildjian Avedis) y otro con presencia como el HH Bosphorus Traditional. Como batería es una mezcla entre Ludwig Vistalite y DW con una caja Brady de madera de arce. En la figura 131 se pueden ver sus iconos. Un kit compacto a elección del usuario.

Tras la elección del Kit esperar a que cargue en memoria (ver las barras que se mueven indicándolo). En este mismo panel ("Kit-Piece") se puede cambiar el tono de cada pieza (Tune), lo que equivaldría a afinarlos. También se configuran los micrófonos de ambiente y el volumen general de todo.



*Figura 131: BFD2 configuración completa*

Por último la configuración del mapeado MIDI, asignar notas a los elementos de la batería. Pero al igual que con el DrumMic® existe la configuración por defecto de GM.

También es posible asignar nota MIDI a cada pad por separado. Basta con ir pulsando Learn y golpear el pad correspondiente para completar la asignación. Cuando se acabe guardar la configuración como preset para otras ocasiones.

Una herramienta es el "Log" de notas MIDI. Es una ventana en la que aparece cada golpe, cada pulsación de pedal y comando MIDI. Esto es una gran fuente de información a la hora de saber qué notas envía la batería para cada articulación. Se anotan y se cambian manualmente en la pestaña "Keymapping/Automation Page".

### **4.2.3 DTXmanía**

DTXmania es un programa de simulación de baterías electrónicas para PC. Es de licencia abierta con un código fuente accesible e incluso participar en sus mejoras. [Ddtx]

El simulador está configurado como un juego y su propósito es aprender a tocar el instrumento. Es posible crear canciones bajo soporte MIDI y se establece una amplia biblioteca online con canciones muy conocidas que ha realizado la comunidad.

Para aquellos que no han tenido la oportunidad de experimentar con uno de estos juegos musicales, el juego es muy simple de definir. El objetivo es golpear el botón correcto (en este caso pads de e-drum, platos, y los pedales) en el instante en que se muestra en la pantalla. Por supuesto todo al ritmo de la música. Aunque parece de naturaleza simple es una de las experiencias de videojuego más adictiva.

Está basado en Drummania el primer videojuego de esta clase realizado por konami en 1999. Actualmente se denomina DTX puesto que la marca Yamaha es quien soporta la mayoría de las actualizaciones. Existen otros ejemplos de este tipo de juegos para las consolas más populares como son los archiconocidos Rock Band. La casa Roland tiene otro programa software de aprendizaje denominado Drum Tutor pero no llega a la potencia y optimización del DTXmania.

Su uso es muy intuitivo si se examina la figura 133.

#### **4.2.3.1 Procedimiento de configuración**

Utilizar el mismo material físico presentado en el apartado 4.2.2.1: pads de batería DD508, el disparador MIDI, el interfaz USB y el ordenador.

En el juego, para configurar los pads, hay una opción del menú principal llamada "Configuración" y dentro de él, el apartado "Drum". Luego buscar el apartado "Drum Keys" y, por ejemplo, configurar la caja pulsando "Snare". Aparecen muchos eventos asignados, la mayoría asignados a MIDI, En la figura 132 se pulsa el número 15 y golpear el pad. Queda guardado el valor. Una vez hecho esto, configura el resto de la e-drum por partes.



Figura 132: Configuración de DTXmania

#### 4.2.3.2 Tocar una canción

Otra de las pruebas que se realizó es sobre una canción. En la figura 133 podemos ver una captura de pantalla del momento en que se tocaba la canción de Joan Jett "I love Rock'n Roll". En ella se puede observar cómo se estaban detectando golpes. El número 381 indica ese número, numero de golpes sin pérdida de ritmo.



Figura 133: Captura de pantalla del testeo de prototipo en DTXmania

#### **4.2.4 Difusión**

Algunos de los progresos de este proyecto han sido difundidos en el blog personal creado para tal fin [edrumlowcost](#). [Dedru]

Con 3000 visitas a lo largo del año se considera un éxito dada la temática tan específica de las baterías electrónicas así como el enfoque que se le asigna de conocimientos de software y hardware.

En este blog, cimentado bajo el gestor de contenido blogger, propiedad de Google, se han diseminado conocimientos en forma de artículo divulgativo. Se mantuvo comunicación con expertos que aportaron claridad a la investigación a través de comentarios.

##### **4.2.4.1 *Participación en jornadas y congresos***

Este proyecto ha auspiciado grandes expectativas y comentarios. Se han recibido muchas preguntas y ayudas y ha habido muchas personas interesadas en su puesta en marcha.

Con el paso del tiempo y, la superación de hitos, se vio la necesidad de compartir esa experiencia y recibir críticas constructivas para un mayor crecimiento. Se colaboró activamente en la realización de conferencias en distintos lugares y fechas.

Sphera'12 Bajo el título experimentos sonoros. Se enseñaron a los alumnos recién egresados aspectos prácticos como halofonía (sonido envolvente), MIDI con Arduino (Hardware libre) además del conocido "I'm sitting in a room". Fue el germen de las motivaciones que desembocaron en este proyecto.

Fecha: 22 al 26 de Abril de 2013

Lugar: EUITT UPM

Más Información: [sphera](#)

Hackculthu: La temática de las jornadas de Hackculthu es sobre software y creaciones artísticas aportando productos con alguna innovación. Celebrando 3 ediciones. Es, en este punto, donde se ambienta la charla. Las personas con inquietudes podrían construirse su propio instrumento electrónico de bajo coste.

Fecha: 17 al 19 de Mayo de 2013  
Lugar: C/ Tembleque 136 Madrid  
Más Información: [hackculthu.org](http://hackculthu.org)

OSHWDem 2013: La convención de hardware libre más importante de España. Carta de acreditación en el anexo firmada por los organizadores agradeciendo la participación. El horario del domingo, que incluye la presentación, se encuentra en la figura 134.

Fueron dos días de exposición del prototipo enseñando cualidades y defectos a más de 400 asistentes. La fotografía de la figura 135 inmortaliza esos instantes. También se impartió en el segundo día una presentación, en pantalla gigante, de las claves del proyecto.

Fecha: 30 noviembre y 1 diciembre de 2013  
Lugar: Centro Cultural Ágora, La Coruña  
Más Información: [oshwdem.org](http://oshwdem.org)

#### DOMINGO

- 10:00 Apertura de puertas y recepción de asistentes.
- 10:30 – 14:00 Charlas, espacios y sorteos.
  - Charla Aeromedia: octocóptero y demostración
  - Charla Escáner 3D y graffiti laser
  - Charla Estación de sensores
  - Charla modulos de comunicacion y cuadricóptero
  - Charla interfaces por control mental y gestos
  - Charla E-drum low cost
  - Charla Raspberry pi monitorizando jaula de animales
  - Charla semáforo de ruido
- 14:00 – 16:00 Comida (las puertas permanecerán cerradas)
- 16:00 – 20:00 Concursos, talleres y espacios.
  - 16:00 – 17:00 Concurso de preguntas y respuestas.
  - 17:00 – 19:00 Talleres (simultáneos)
    - Taller de soldadura
    - Taller de software libre para diseño electrónico
- 20:00 Depedida y cierre.

*Figura 134: Horario del domingo en OSHWdem*

En todas las charlas se informó de que se trataba de un trabajo englobado en el Proyecto Fin de Carrera de un alumno de la Universidad Politécnica de Madrid.



*Figura 135: Presentación del proyecto en OSHWdem*

# 5 CONCLUSIONES Y FUTUROS DESARROLLOS

---



## 5.1 Conclusiones finales

Se ha realizado un prototipo de Disparador MIDI para batería que cumple con las especificaciones propuestas en el apartado 3.1.2. Los análisis funcionales del apartado 4.2 verifican su puesta en marcha y la utilización del dispositivo junto a sampleadores, le posicionan como un producto en vías de ser similar a uno comercial. Aparte también cumple con las características que definen un sistema embebido: [Villo5]

- Conurrencia: Los componentes del sistema controlado son atendidos generando acciones de control y de visualización. Un computador ejecuta acciones de manera secuencial pero con tanta rapidez que aparenta simultaneidad. En el diseño se ha tenido un programa principal se ejecuta indefinidamente y todos los sucesos independientes se tratan por interrupciones y se gestionaran rápidamente. Por ejemplo los cambios de parámetros o las percusiones.
- Fiabilidad: El número de errores es limitado. El sistema es sencillo y funcional. Asegurar que si el sistema de control falla quede en un estado seguro.
- Eficiencia: El sistema responde con rapidez ante los cambios. Si se produce una percusión el sistema responde en menos de 3ms.
- Interacción con dispositivos físicos: Son los intermediarios del entorno y el microprocesador debe comunicarse con ellos. El sistema está compuesto por 6 pads es mono, una pedalera para el Hi-hat, un encoder y un LCD.
- Robustez: No siempre trabajan en las mejores condiciones. Son transportados y reciben impactos y vibraciones. Por ello se requiere el factor de protección adicional que proporciona la carcasa de metal.
- Bajo consumo: Se alimenta por conexión USB 5V@500mA de un ordenador portátil a batería.
- Bajo peso: Aparato de pequeñas dimensiones y poco peso.
- Bajo precio: Electrónica de consumo basada en módulos hardware intercambiables.

## 5.2 Futuros desarrollos

Tras realizar este intenso trabajo creemos en que puede dar pie a otras profusas búsquedas, mejoras adicionales y construcciones de prototipos.

- Integrar una memoria en el dispositivo y utilizarla como sampleador
- Estudio de charles con sensor óptico. Ventajas: Más transiciones y más estables.
- Integrar en otras plataformas con más potencia como Stellaris. También realizar el sistema digital a medida incorporando un microcontrolador más potente por ejemplo un PIC32
- Montaje SMD, más profesional y menor tasa de errores.
- Programar un driver MIDI USB para la comunicación de este dispositivo con un entorno como Windows. Eliminar el dispositivo intermedio, conversor USB.
- Licencia Open Source Hardware.
- Parada de plato.
- Carcasa de plástico con impresora 3D.
- Detector de pads estéreos.
- Mayores posibilidades de personalización del menú.
- Incorporar un Sistema Operativo para microcontroladores que posibilite una mayor abstracción y ejecución en concurrencia.

# 6 REFERENCIAS

---



## 6.1 Referencias bibliográficas

- [Recoo] RECUERO LÓPEZ, M. Ingeniería Acústica.2º ed. España: Paraninfo, 2000.
- [SEo8] SERWAY RAYMOND, A. Física para ciencias e ingeniería.7º ed. Cengage Learning, 2008.
- [Peto1] PETHERSEN, M. Mathematical Harmonies. Universidad de Carolina del Sur. 2001
- [Blao5] BLADES, J and DEAN, J. Cómo tocar la batería y otros instrumentos de percusión.7ª ed. Madrid: EDAF, 2005.
- [Cuc03] CUCCIARDI, F. La Batería acústica.2ª ed. Valencia: Rivera Editores, 2003.
- [Pino1] PINKSTERBOER, H. Batería Guía Mundimúsica. España: Mundimúsica Ediciones, 2001.
- [Stao5] STARR, E. Manual para tocar la batería.1ª ed. Barcelona: Manontropo, 2005.
- [Aso95] ASOCIATION MIDI MANUFACTURERS. MIDI 1.0 Detailed Specification.Vol. Version 4.2. 1995.
- [Aso98] ASOCIATION MIDI MANUFACTURERS. General MIDI System Level 1. Version 2. 1998.
- [Leh93] LEHRMAN, PD and TULLY, T. MIDI for the Professional. Amsco Publications, 1993.
- [Pen96] PENFOLD, RA. Advanced MIDI user's guide. PC Publishing, 1996.
- [Est12] ESTEBAN ILLESCAS, M. Construcción de una impresora 3D Open Source edited by A Valero Gómez, J( González Gómez, Universidad Carlos III de Madrid. Departamento de Ingeniería de Sistemas y Automática. 2012.
- [Villo5] VILLARROEL, J. Sistemas Empotrados. Universidad de Zaragoza 2005
- [Atm12] ATMEL CORPORATION Datasheet Atmega 2560 2012
- [Tex01] TEXAS INSTRUMENTS, Signal Conditioning Piezoelectric Sensor, Application Report, Septiembre 2001
- [Atmo9] ATMEL C., Datasheet Atmega 328P . Rev D, 2009
- [Fra05] FRANCO, S. Diseño con amplificadores operacionales y circuitos integrados analógicos.3ª ed. México D.F.: McGraw-Hill, 2005.
- [Mal91] MALVINO, AP et al. Principios de electrónica.Vol. 2. McGraw-Hill, 1991.

[Ker91] KERNIGHAN, B and RITCHIE, D. El lenguaje de programacion C.2<sup>a</sup> ed. Mexico: Prentice-Hall Hispanoamericana, 1991.

[Palo3] PALLÁS ARENY, R. Sensores y acondicionadores de señal.4<sup>o</sup> ed. España: Marcombo, 2003.

[Tipo6] TIPLER, P and MOSCA, G. Oscilaciones y Ondas. Fisica para la ciencia y la tecnología. 5<sup>o</sup> Edición ed. Barcelona: Reverté, 2006.

[Cupoo] CUPICH P. ELIZONDO F. Actuadores Piezoelectricos, 2000

[Mur11] MURATA C. Piezoelectric Ceramic Sensors (Piezotite) 2011

[Tomo3] TOMICO, P. Disparador MIDI. Universidad Politécnica de Madrid. Departamento de Electrónica Física. 2003

[Whe11] WHEAT, D. Arduino Internals.New Edition ed. Apress L. P., 2011. doi: pmid: ISBN 978-1-4302-3882-9.

[Med12] MEDELLI C. DD-508 Owner's Manual, 2012

[Rolo4] ROLAND C. TD.3 Owner's Manual, 2004

[Opp98] OPPENHEIM, AV, WILLSKY, AS and NAWAB, SH. Señales y sistemas. Pearson Educación, 1998.

[Mon10] MONJE CENTENO, Daniel. Conceptos electrónicos en la medida de la aceleración y la vibración, 2010.

[Zor98] ZORITA, DS. Análisis y Síntesis de señales musicales. revised by Escuela Universitaria de Ingeniería Técnica de Telecomunicación, 1998.

[Nab01] NABATOV, VU. Acústica. revised by Escuela Universitaria de Ingeniería Técnica de Telecomunicaciones, Departamento de Publicaciones, 2001.

[Pero3] PÉREZ, MS et al. Electrónica de comunicaciones. Prentice Hall, 2003.

[Fer12] FERNÁNDEZ M. Estudio e integración de sistemas de bajo coste para el Diseño Digital y el prototipado rápido Universidad Politécnica de Valencia 2012

[Usb99] USB ORG. Universal Serial Bus Device Class Definition for MIDI Devices Noviembre 1999

[Usboo] USB ORG. USB 2.0 Detailed Specification Revisión 2, Abril 2000

[Usao6]USATEGUI, Angulo. JM i Angulo Martínez, I. Microcontroladores PIC: Diseño práctico de aplicaciones. 2006

[Jaso8] JASIO L., Programming PIC32 in C, Elsevier, 2008

[Ert11] ERTURK A., INMAN J., Piezoelectric Energy Harvesting, Willey 2011

[Heno6] HENRY T. The electronic drum cookbook Rev 4 2006

[Fxp07] FXPANSION C. BFD2 manual Rev D 2007

## 6.2 Referencias digitales

- [Dbata] Portal de aficionados a la batería Batacas.com. <http://www.batacas.com/>
- [Drola] Roland Iberia. <http://www.rolandiberia.com/es/produtos/bateria-e-percusion/>
- [Dyama] Yamaha <http://es.yamaha.com/es/products/musical-instruments/drums/>
- [Dales] Alesis <http://alesis.com/en/index.php>
- [D2box] 2box Drumitfive <http://www.2box.se/World/index.php>
- [Dthom] Thomann distribuidora de instrumentos <http://www.thomann.de/es/index.html>
- [Dvdru] Foro de baterías electrónicas realizadas a mano <http://vdrums.com/forum/>
- [Dsenn] Sennheiser fabricante de microfonía. <http://en-de.sennheiser.com/>
- [Dxlna] Addictive Drums [http://www.xlnaudio.com/products-addictive\\_drums-addictive\\_drums](http://www.xlnaudio.com/products-addictive_drums-addictive_drums)
- [Dfxpa] BFD2 <http://www.fxpansion.com/index.php?page=53>
- [Dtoon] Ezdrummer <http://www.toontrack.com/ezdrummerline.asp>
- [Dmfmo] Datasheet IC MFM 0860 <http://practicalusage.com/pu/wp-content/uploads/2011/05/MFM0860-USB-MIDI.pdf>
- [Du2m] Hoja de características interfaz USB a MIDI  
[http://www.cablesdirect.co.uk/product-catalogue/adapters/Adapters\\_USB/USB-MIDI-Adaptor/USB-MIDI2](http://www.cablesdirect.co.uk/product-catalogue/adapters/Adapters_USB/USB-MIDI-Adaptor/USB-MIDI2)
- [Dhydr] Hydrogen <http://www.hydrogen-music.org>
- [Dhell] Blog de baterías electrónicas <http://www.hellfiredrums.com/>
- [Dardu] Portal de Arduino. <http://www.Arduino.cc/>
- [Dedru] Edrum Low Cost <http://edrumlowcost.blogspot.com.es/>
- [Dmidi] MIDI.org <http://MIDI.org/>
- [Dmiox] MIDI-OX<http://www.midiox.com/>
- [Dyour] Comparación de fabricantes de prototipos en PCB<http://www.yourtronics.com/seeed-studio-vs-itead-studio-vs-osh-park/>
- [Dinst] Instructables <http://www.instructables.com/>
- [Dstei] Desarrollador de programas de audio digital <http://www.steinberg.net/>
- [Ddtx] DTXmania <http://es.sourceforge.jp/projects/dtxmania/>
- [Dmega] Megadrum<http://www.megadrum.info/>

[Dmicr] Microdrum <http://microdrum.altervista.org/>

[Dkont] KONTAKT 5 Player <http://www.native-instruments.com/es/products/komplete/samplers/kontakt-5/>

[Dhpc0] HP 4540s [Link](#)

[Dwiki] Wikipedia <http://es.wikipedia.org/>

[Dstan] Center for Computer Research in Music and Acoustics, University of Stanford  
<https://ccrma.stanford.edu/CCRMA/Courses/252/sensors/node8.html>

[Dnati] National Instruments <http://spain.ni.com/>

[Dvisi] Comparación de simuladores electrónicos  
<http://visionics.blogspot.com.es/2012/04/comparison-between-edwinxp-pspice.html>

[Dfast]  
[http://en.wikipedia.org/wiki/WFD\\_World's\\_Fastest\\_Drummer\\_Extreme\\_Sport\\_Drumming](http://en.wikipedia.org/wiki/WFD_World's_Fastest_Drummer_Extreme_Sport_Drumming)

[Dcads] Eagle CAD <http://www.cadsoftusa.com/>

[Ddeso] USBPcap <http://desowin.org/usbpmpcap/index.html>

[Dwire] Wireshark <http://www.wireshark.org/>



# 7 ANEXOS

---



## Anexo A: Pruebas ejecutadas con Arduino

Se realizaron y documentaron algunos ejercicios para aprender el lenguaje y funcionalidades de Arduino. Se en anexos por ser parte del proceso pero no poseer una relación clara con el temario.

### Rendimiento ADC

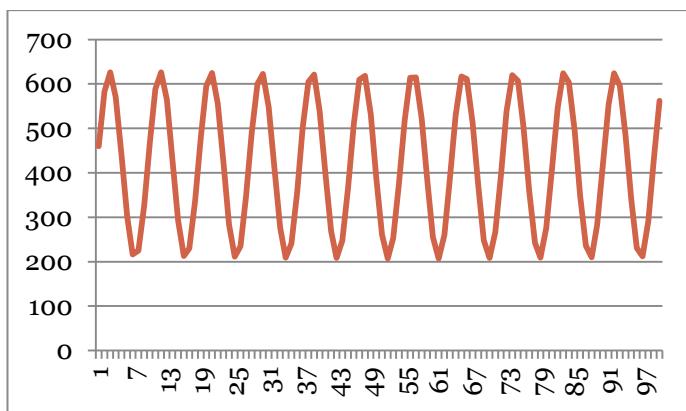
El interés de este ejercicio radica en observar si se cumple lo propuesto: convertir cualquier valor de la entrada analógica en un espacio de tiempo acordado en un código decimal entre 0 y 1023.

El microcontrolador Atmel convierte cualquier señal en un periodo de 100us ( $f_{ADC}=10\text{khz}$ ) con la función analogRead (). Por tanto, según el teorema de Nyquist la frecuencia máxima de muestreo para reconstruir cualquier señal es de 5khz [Opp98].

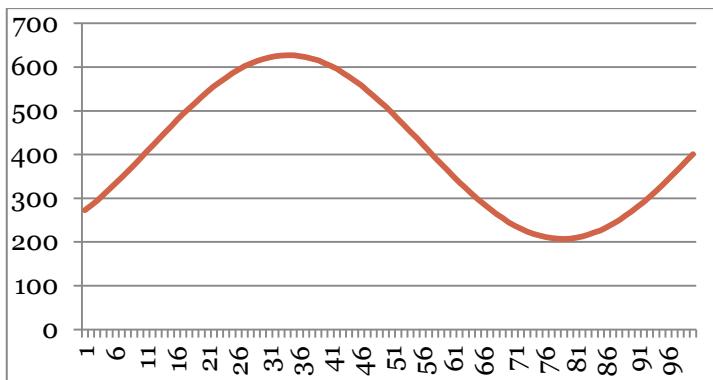
Se utilizaron sinusoidales, con el generador de funciones del laboratorio, de frecuencia X, de tensión 1Vpp y de offset 2V. El programa utilizado en Arduinorecoge datos analógicos y los envía por el puerto serie hacia el hiperterminal del ordenador. Se reconocen los valores en una tabla Excel y se visualiza en una gráfica con el comando Insertar>línea.

### Resultados

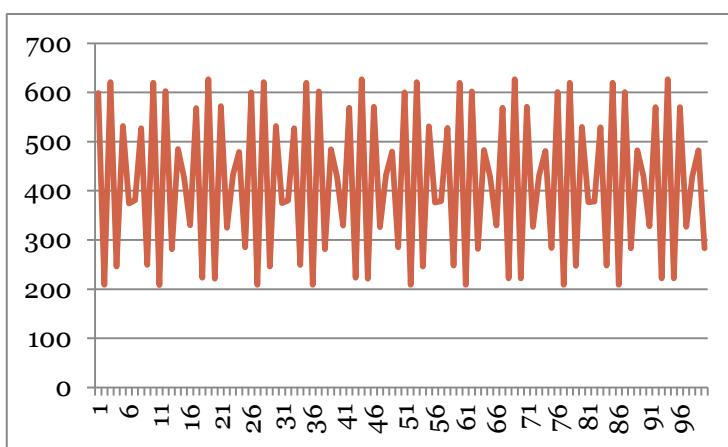
100 muestras a 1khz. El código digital del valor medio es 408 que equivale haciendo las conversiones a 2V. El valor pico a pico es 204 que corresponde a 1V.



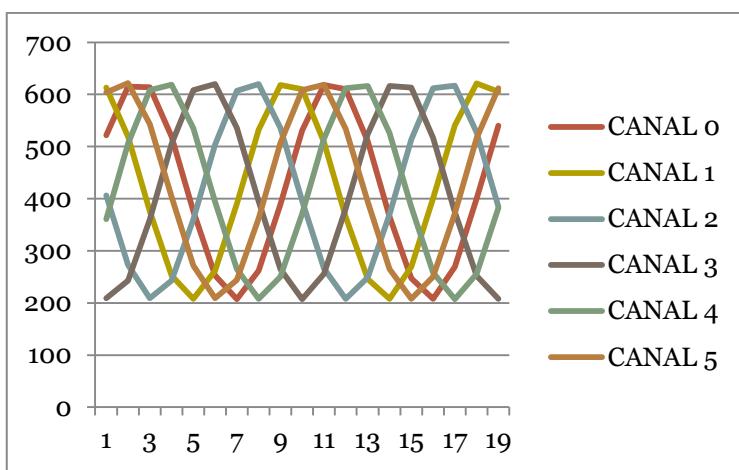
100 Hz 100 muestras. Reconstrucción perfecta de la señal



5khz. Límite de Nyquist. Con dificultad se podría estudiar esta señal.



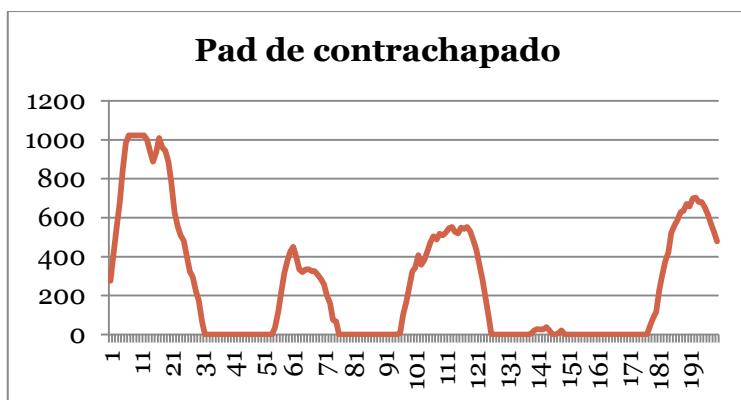
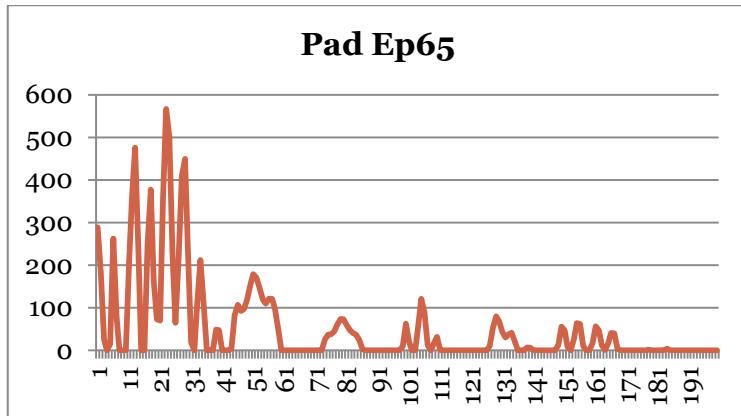
Hacemos lo mismo pero para 6 canales. 20 muestras a 1khz con programa tabla\_sinusoidales.



Utilizando otro programa en Arduino se examina cómo se comportan los siguientes pads. Si supera un umbral de 100mV recoge 200 muestras y las envía por el puerto serie. Con ello se comprende el fenómeno mecánico-eléctrico.

Los momentos de código o son por la conducción del diodo de protección por lo que se concluye que estamos ante tensión negativa.

Se observa en las dos gráficas, la cantidad de oscilaciones que se producen, siendo fluctuaciones más rápidas y pronunciadas en el pad comercial. El pad descrito en el apartado 3.2.3.4 tiene una amortiguación más lenta.



## Análisis de tiempo

En estas páginas intentaremos resolver la siguiente pregunta: ¿Cuánto demora en ejecutarse una sentencia en el lenguaje Arduino?

El lenguaje de programación Arduino se basa en C/C++, por lo tanto, se trata de un lenguaje de alto nivel. Un lenguaje de alto nivel es altamente independiente del procesador y por tanto no existe una clara correspondencia entre las instrucciones(o sentencias) del lenguaje Arduino y el lenguaje de nemáticos propio del Atmega328p (lenguaje de máquina o de bajo nivel).

Revisando la hoja de datos del Atmega328p observaremos el número de ciclos de reloj correspondientes a cada instrucción en lenguaje máquina. Por lo que conociendo

la frecuencia del reloj utilizado es posible saber el tiempo que demora cada instrucción en ejecutarse. ".

Arduino UNO funciona a 16Mhz. El ciclo de reloj (inverso a la frecuencia) es de 62,5 nseg. Y según el datasheet el Atmega328 utiliza un ciclo en la mayoría de las instrucciones (salvo saltos y multiplicaciones). Por consiguiente cada instrucción tarda 62,5nseg.

| Mnemonics                         | Operands | Description                  | #Clocks |
|-----------------------------------|----------|------------------------------|---------|
| ARITHMETIC AND LOGIC INSTRUCTIONS |          |                              |         |
| ADD                               | Rd, Rr   | Add two Registers            | 1       |
| ADC                               | Rd, Rr   | Add with Carry two Registers | 1       |

Existe absoluta claridad en cuanto a los tiempos de ejecución, pero esto, siempre y cuando sea utilizando las "instrucciones del lenguaje de máquina

Esto es la teoría pero en la práctica, y lo observamos en las bibliotecas de Arduino, nos encontramos que a una función de alto nivel le corresponden un conjunto definido de sentencias en lenguaje C que a su vez se componen de tantas llamadas a funciones como se puede ver a continuación con la función digitalWrite().

```
sketch_oct20c $  
void digitalWrite(uint8_t pin, uint8_t val)  
{  
    uint8_t timer = digitalPinToTimer(pin);  
    uint8_t bit = digitalPinToBitMask(pin);  
    uint8_t port = digitalPinToPort(pin);  
    volatile uint8_t *out;  
  
    if (port == NOT_A_PIN) return;  
  
    // If the pin that support PWM output, we need to turn it off  
    // before doing a digital write.  
    if (timer != NOT_ON_TIMER) turnOffPWM(timer);  
  
    out = portOutputRegister(port);  
  
    uint8_t oldSREG = SREG;  
    cli();
```

En definitiva el cálculo teórico es tedioso.

### Possibles Soluciones para calcular tiempo

Las soluciones que se manejan sería recoger funciones cuyo valor se conoce y utilizarlas como regla para medir otras tantas.

Bastaría por tanto recoger los tiempos de antes y después de ejecutar una zona crítica.

**Solución 1** utilizar la función micros() que devuelve el número de microsegundos desde que la placa Arduino empezó a correr el programa actual. Su resolución es de 4useg en Arduino UNO funcionando a 16Mhz.

Problema: La función micros genera incertidumbre.

Resultado al medir la función analogread(). El programa se denomina tiempo\_micros



**Solución 2** Utilizar un osciloscopio.

Análisis de tiempo>tiempo puerto

Coger ese programa de prueba e introducir la zona crítica entre las dos sentencias de los puertos.

PORTB= xxxxxxx; es una INSTRUCCIÓN. Las instrucciones, no todas son de un ciclo, en este caso sí lo es. Esta vendría a equivaler en ASM; ldi PORTB, xxxxxxxx

Para calcular pondremos el osciloscopio en modo trigger y a la salida del puerto B. Detectará una subida y una bajada de señal producida por los puertos antes mencionados y la distancia entre ellos será el tiempo que tarda en ejecutarse nuestra sentencia en C.

Resultado al medir analogRead().El programa se denomina tiempo\_puerto

## Análisis con Matlab del pad de batería

Una vez se tienen las muestras en un fichero de texto se ejecuta el siguiente programa escrito en Matlab.

```
17 - load k.txt
18 - kv=k*0.0049; %Cada LSB 4.9mV
19 - freq_muestreo=10000;
20 - n_muestras=length(kv);
21 - t1=0:1/freq_muestreo:(n_muestras-1)/freq_muestreo;
22 - grid;
23 - figure(1)
24 - plot (t1,kv)
25 - NFFT= 2^nextpow2(n_muestras);
26 - m= fft(kv,NFFT)/n_muestras;
27 - f=freq_muestreo/2*linspace(0,1,NFFT/2+1);|
28 - figure(2)
29 - plot(f,2*abs(m(1:NFFT/2+1)))
30
```

Código utilizado en Matlab para adquirir la FFT

Lee las muestras del fichero de texto y las carga en una variable. Utilizando parámetros como la resolución del ADC o la frecuencia de muestreo representan en una gráfica la señal y en otra gráfica, el espectro. Para el análisis también se ha utilizado la función de Matlab fvtool.

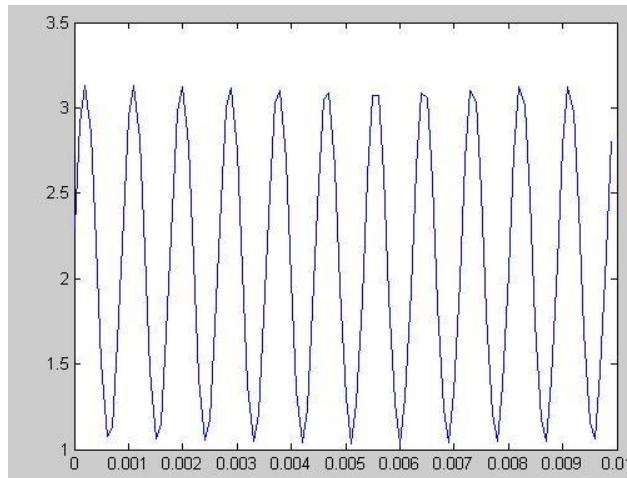
Por otro lado para las señales discretas necesitamos realizar el algoritmo de la transformada rápida de Fourier (FFT) [Opp98]. Por lo que, como se ha comentado, al final del programa de la figura se muestra el espectro unilateral de la señal estudiada.

Recordatorio para entender las gráficas del espectro en frecuencia.

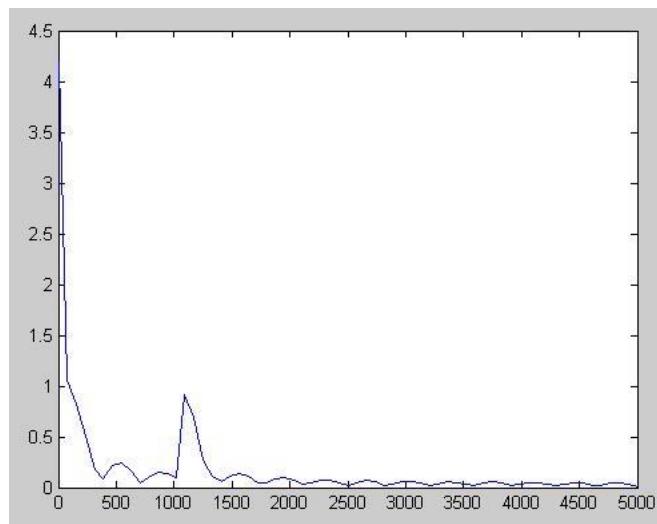
- Para una señal no periódica el espectro consiste en un continuo de frecuencias.
- El ADC al ser un circuito no lineal provoca armónicos y distorsiones en la respuesta en frecuencia.
- La relación de Parseval para señales periódicas continúas es la suma de las componentes frecuenciales:

$$P = \frac{1}{T} * \int_{-T/2}^{T/2} |x(t)|^2 dt = \sum_{k=-\infty}^{\infty} |a_k|^2$$

Primero se prueba con una onda conocida para ver si los cálculos son correctos. Señal sinusoidal digital de  $f_s=10\text{khz}$  2V de offset y 1V de amplitud



**Figura 136: Señal sinusoidal**



**Figura 137: Espectro unilateral señal sinusoidal**

El espectro unilateral de la figura es 2 veces uno bilateral. En este caso al ser unilateral:

$$P=2A_0=4\text{W} \text{ en continua}$$

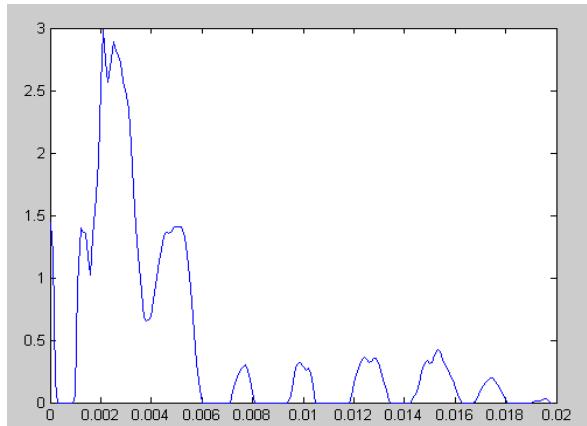
$$P_{\text{sinusoidal}} = V_{\text{sin}}^2 = A_0 = 1\text{W}$$

La potencia total es de 5W

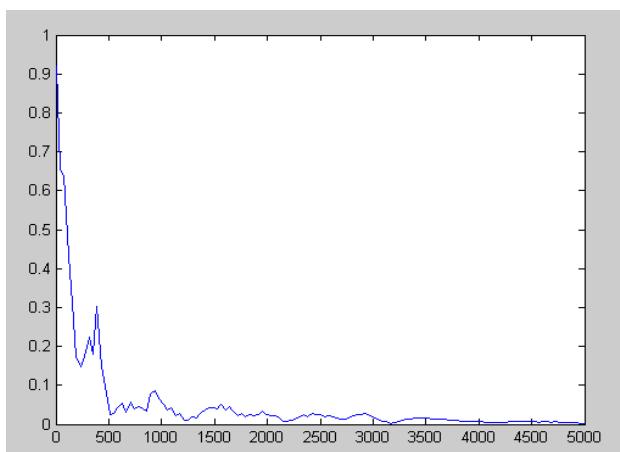
Coinciendo con la figura del espectro.

Ahora se prueba con la señal capturada del ADC del Arduino proveniente de un PAD Yamaha Ep65. La frecuencia de muestreo es de 10khz.

Las señales son recogidas con un ADC que tiene protección en las entradas ante voltajes negativos lo que provoca los cortes abruptos.



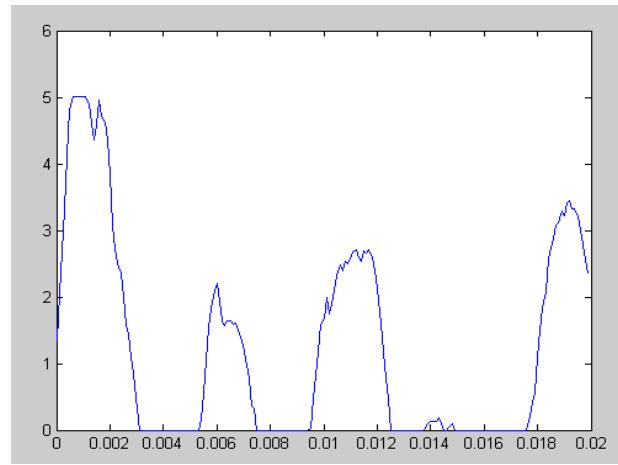
*Figura 138: Señal recogida con el ADC del pad*



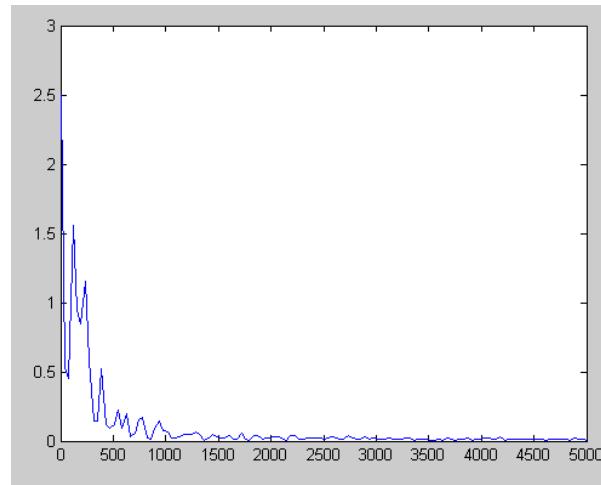
*Figura 139: Espectro unilateral del pad*

Señal capturada del ADC del pad de contrachapado del apartado 3.2.4.3. La frecuencia de muestreo es la misma, 10khz.

Se observa que prácticamente toda la información está contenida en frecuencias inferiores a 1Khz.

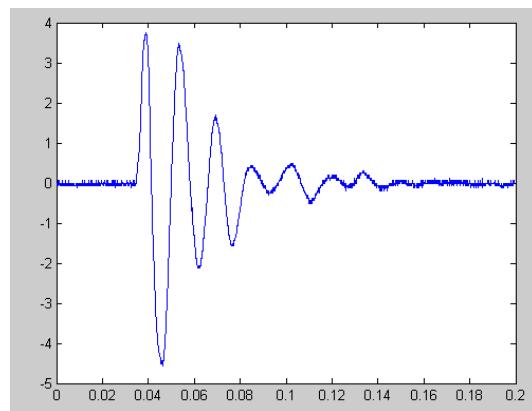


*Figura 140: Señal recogida con el ADC del pad construido*



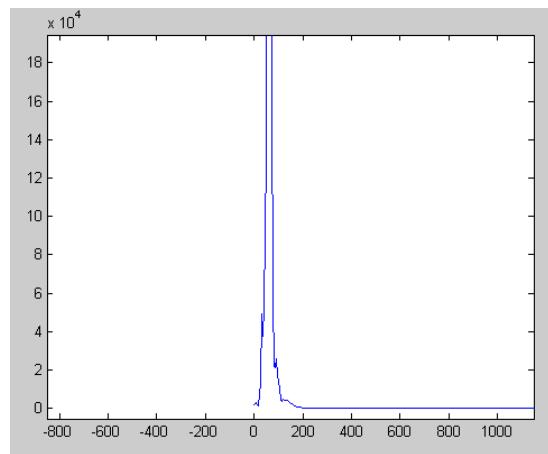
*Figura 141: Espectro unilateral pad*

Señal capturada del osciloscopio TP65 Yamaha DTX. Fs=40khz



*Figura 142: Señal recogida con el ADC del pad caja*

En este caso se ha elegido un espectro bilateral recogida con el ADC del pad. Es una señal con un contenido espectral muy bajo (inferior a 100Hz) y que se asemeja a una delta.



*Figura 143: Espectro pad caja*

## Anexo B: Hojas de características

### Piezoelectric Sound Components

**muRata**

#### Piezoelectric Diaphragms

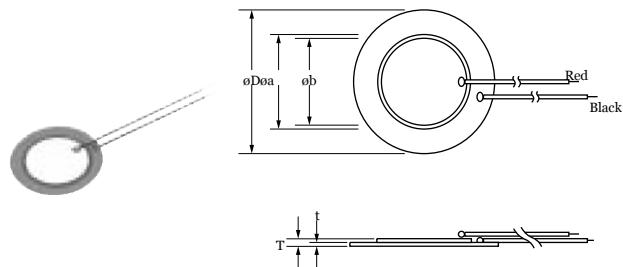
##### Features

1. Clear sound
2. Ultra thin and lightweight
3. No contacts: therefore, no noise and highly reliable
4. Low power consumption for voltage type

##### Applications

Clocks/Calculators/Digital camera/Various alarms

(Burglar alarms, etc.)



#### External Drive Type

| Part Number | Resonant Frequency (kHz) | Resonant Impedance (ohm) | Capacitance (nF) | Plate Size dia. D (mm) | Element Size dia. a (mm) | Electrode dia. b (mm) | Thickness T (mm) | Plate Thickness t (mm) | Plate Material |
|-------------|--------------------------|--------------------------|------------------|------------------------|--------------------------|-----------------------|------------------|------------------------|----------------|
| 7BB-12-9    | 9.0 ±1.0kHz              | 1000 max.                | 8.0 ±30% [1kHz]  | 12.0                   | 9.0                      | 8.0                   | 0.22             | 0.10                   | Brass          |
| 7BB-15-6    | 6.0 ±1.0kHz              | 800 max.                 | 10.0 ±30% [1kHz] | 15.0                   | 10.0                     | 9.0                   | 0.22             | 0.10                   | Brass          |
| 7BB-20-3    | 3.6 ±0.6kHz              | 500 max.                 | 20.0 ±30% [1kHz] | 20.0                   | 14.0                     | 12.8                  | 0.22             | 0.10                   | Brass          |
| 7BB-20-6    | 6.3 ±0.6kHz              | 350 max.                 | 10.0 ±30%        | 20.0                   | 14.0                     | 12.8                  | 0.42             | 0.20                   | Brass          |

|                   |                         |           |                            |      |      |      |      |      |   |
|-------------------|-------------------------|-----------|----------------------------|------|------|------|------|------|---|
|                   |                         |           | [1kHz]                     |      |      |      |      |      |   |
| <b>7BB-20-6Lo</b> | $6.3 \pm 0.6\text{kHz}$ | 1000 max. | $10.0 \pm 30\%$<br>[1kHz]  | 20.0 | 14.0 | 12.8 | 0.42 | 0.20 | Brass (with<br>Lead Wire:<br>AWG32 Length 50mm) |
| <b>7BB-27-4</b>   | $4.6 \pm 0.5\text{kHz}$ | 200 max.  | $20.0 \pm 30\%$<br>[1kHz]  | 27.0 | 19.7 | 18.2 | 0.54 | 0.30 | Brass   |
| <b>7BB-27-4Lo</b> | $4.6 \pm 0.5\text{kHz}$ | 300 max.  | $20.0 \pm 30\%$<br>[1kHz]  | 27.0 | 19.7 | 18.2 | 0.54 | 0.30 | Brass (with<br>Lead Wire:<br>AWG32 Length 50mm) |
| <b>7BB-35-3</b>   | $2.8 \pm 0.5\text{kHz}$ | 200 max.  | $30.0 \pm 30\%$<br>[1kHz]  | 35.0 | 25.0 | 23.0 | 0.53 | 0.30 | Brass   |
| <b>7BB-35-3Lo</b> | $2.8 \pm 0.5\text{kHz}$ | 200 max.  | $30.0 \pm 30\%$<br>[1kHz]  | 35.0 | 25.0 | 23.0 | 0.53 | 0.30 | Brass (with<br>Lead Wire:<br>AWG32 Length 50mm) |
| <b>7BB-41-2</b>   | $2.2 \pm 0.3\text{kHz}$ | 250 max.  | $30.0 \pm 30\%$<br>[1kHz]  | 41.0 | 25.0 | 23.0 | 0.63 | 0.40 | Brass   |
| <b>7BB-41-2Lo</b> | $2.2 \pm 0.3\text{kHz}$ | 300 max.  | $30.0 \pm 30\%$<br>[1kHz]  | 41.0 | 25.0 | 23.0 | 0.63 | 0.40 | Brass (with<br>Lead Wire:<br>AWG32 Length 50mm) |
| <b>7NB-31R2-1</b> | $1.3 \pm 0.5\text{kHz}$ | 300 max.  | $40.0 \pm 30\%$<br>[120Hz] | 31.2 | 19.7 | 18.2 | 0.22 | 0.10 | Nickel Alloy                                    |

TLV277x, TLV277xA  
**FAMILY OF 2.7-V HIGH-SLEW-RATE, RAIL-TO-RAIL OUTPUT  
OPERATIONAL AMPLIFIERS**

SLOS209G – JANUARY 1998 – REVISED FEBRUARY 2004

- High Slew Rate . . . 10.5 V/ $\mu$ s Typ
- High-Gain Bandwidth . . . 5.1 MHz Typ
- Supply Voltage Range 2.5 V to 5.5 V
- Rail-to-Rail Output
- 360  $\mu$ V Input Offset Voltage
- Low Distortion Driving 600- $\Omega$  0.005% THD+N
- mA Supply Current (Per Channel)
- 7nV/ $\sqrt{\text{Hz}}$  Input Noise Voltage
- 2 pA Input Bias Current
- Characterized From TA = -55°C to 125°C
- Available in MSOP and SOT-23 Packages
- Micropower Shutdown Mode . . . IDD < 1  $\mu$ A
- Available in Q-Temp Automotive  
High Reliability Automotive Applications  
Configuration Control / Print Support  
Qualification to Automotive Standards

### description

The TLV277x CMOS operational amplifier family combines high slew rate and bandwidth, rail-to-rail output swing, high output drive, and excellent dc precision. The device provides 10.5 V/ $\mu$ s of slew rate and 5.1 MHz of bandwidth while only consuming 1 mA of supply current per channel. This ac performance is much higher than current competitive CMOS amplifiers. The rail-to-rail output swing and high output drive make these devices a good choice for driving the analog input or reference of analog-to-digital converters. These devices also have low distortion while driving a 600- $\Omega$  load for use in telecom systems.

These amplifiers have a 360- $\mu$ V input offset voltage, a 17 nV/ $\sqrt{\text{Hz}}$  input noise voltage, and a 2-pA input bias current for measurement, medical, and industrial applications. The TLV277x family is also specified across an extended temperature range (-40°C to 125°C), making it useful for automotive systems, and the military temperature range (-55°C to 125°C), for military systems.

These devices operate from a 2.5-V to 5.5-V single supply voltage and are characterized at 2.7 V and 5 V. The single-supply operation and low power consumption make these devices a good solution for portable applications. The following table lists the packages available.

**FAMILY PACKAGE TABLE**

| DEVICE  | NUMBER OF CHANNELS | PACKAGE TYPES |      |      |        |       |      |      |      | SHUTDOWN | UNIVERSAL EVM BOARD                             |
|---------|--------------------|---------------|------|------|--------|-------|------|------|------|----------|---|
|         |                    | PDIP          | CDIP | SOIC | SOT-23 | TSSOP | MSOP | LCCC | CPAK |          |   |
| TLV2770 | 1                  | 8             | —    | 8    | —      | —     | 8    | —    | —    | Yes      | Refer to the EVM Selection Guide (Lit# SLOU060) |
| TLV2771 | 1                  | —             | —    | 8    | 5      | —     | —    | —    | —    | —        |   |
| TLV2772 | 2                  | 8             | 8    | 8    | —      | 8     | 8    | 20   | 10   | —        |   |
| TLV2773 | 2                  | 14            | —    | 14   | —      | —     | 10   | —    | —    | Yes      |   |
| TLV2774 | 4                  | 14            | —    | 14   | —      | 14    | —    | —    | —    | —        |   |
| TLV2775 | 4                  | 16            | —    | 16   | —      | 16    | —    | —    | —    | Yes      |   |

A SELECTION OF SINGLE-SUPPLY OPERATIONAL AMPLIFIER PRODUCTS†

| DEVICE  | VDD (V)   | BW (MHz) | SLEW RATE (V/ $\mu$ s) | IDD (per channel) ( $\mu$ A) | RAIL-TO-RAIL |
|---------|-----------|----------|------------------------|------------------------------|--------------|
| TLV277X | 2.5 – 6.0 | 5.1      | 10.5                   | 1000                         | O            |
| TLV247X | 2.7 – 6.0 | 2.8      | 1.5                    | 600                          | I/O          |
| TLV245X | 2.7 – 6.0 | 0.22     | 0.11                   | 23                           | I/O          |
| TLV246X | 2.7 – 6.0 | 6.4      | 1.6                    | 550                          | I/O          |

†All specifications measured at 5 V

TLV277x, TLV277xA  
FAMILY OF 2.7-V HIGH-SLEW-RATE, RAIL-TO-RAIL OUTPUT  
OPERATIONAL AMPLIFIERS

**Recommended operating conditions = 2.7 V (unless otherwise noted)**

| PARAMETER      | TEST CONDITIONS   | TA†  | TLV277xC      |      |      | UNIT                         |  |
|----------------|---|--|---------------|------|------|------------------------------|--|
|                |   |  | MIN           | TYP  | MAX  |                              |  |
| VIO            | TLV2770/1/2<br>TLV2773/4/5  | VO = 0,<br>VIC = 0,<br>VDD = $\pm 1.35$ V,<br>$R_S = 50 \Omega$ ,<br>No load | 25°C          | 0.48 | 2.5  | mV                           |  |
|                |   |  | Full range    | 0.53 | 2.7  |                              |  |
|                |   |  | 25°C          | 0.8  | 2.7  |                              |  |
|                |   |  | Full range    | 0.86 | 2.9  |                              |  |
| $\alpha_{VIO}$ | Temperature coefficient of input offset voltage                       | VIC = 0,<br>$R_S = 50 \Omega$  | 25°C to 125°C | 2    |      | $\mu\text{V}/^\circ\text{C}$ |  |
| IIO            | Input offset current  |  | 25°C          | 1    | 60   | pA                           |  |
| IIB            | Input bias current  |  | Full range    | 2    | 100  |                              |  |
|                |   |  | 25°C          | 2    | 60   |                              |  |
|                |   |  | Full range    | 6    | 100  |                              |  |
|                |   |  | 25°C          | 2.6  |      | V                            |  |
| VOH            | High-level output voltage   |  | Full range    | 2.5  |      |                              |  |
|                | IOH = -0.675 mA<br>IOH = -2.2 mA                                      | 25°C   | 2.4           |      |      |                              |  |
|                |   | Full range   | 2.1           |      |      |                              |  |
|                |   | 25°C   | 0.1           |      | V    |                              |  |
| VOL            | Low-level output voltage  | VIC = 1.35 V, IOL = 0.675 mA<br>VIC = 1.35 V, IOL = 2.2 mA                   | Full range    | 0.2  |      |                              |  |
|                |   |  | 25°C          | 0.21 |      |                              |  |
|                |   |  | Full range    | 0.6  |      |                              |  |
|                |   |  | 25°C          | 20   | 380  | V/mV                         |  |
| AVD            | Large-signal differential voltage amplification                       | VIC = 1.35 V, $R_L = 10 \text{ k}\Omega$ ,<br>VO = 0.6 V to 2.1 V            | Full range    | 13   |      |                              |  |
| ri(d)          | Differential input resistance   |  | 25°C          | 1012 |      | $\Omega$                     |  |
| ci(c)          | Common-mode input capacitance   | f = 10 kHz   | 25°C          | 8    |      | pF                           |  |
| Zo             | Closed-loop output impedance  | f = 100 kHz, AV = 10   | 25°C          | 25   |      | $\Omega$                     |  |
| CMRR           | Common-mode rejection ratio   | VIC = 0 to 1.5 V, VO = VDD/2,<br>$R_S = 50 \Omega$                           | 25°C          | 60   | 84   | dB                           |  |
|                |   |  | Full range    | 60   | 82   |                              |  |
| kSVR           | Supply voltage rejection ratio<br>( $\Delta V_{DD} / \Delta V_{IO}$ ) | VDD = 2.7 V to 5 V, VIC = VDD/2,<br>No load                                  | 25°C          | 70   | 89   | dB                           |  |
|                |   |  | Full range    | 70   | 84   |                              |  |
| IDD            | Supply current (per channel)  | VO = VDD/2, No load  | 25°C          | 1    | 2    | mA                           |  |
|                |   |  | Full range    |      | 2    |                              |  |
| IDD(SHDN)      | Supply current in shutdown (per channel)                              |  | 25°C          | 0.8  | 1.5  | $\mu\text{A}$                |  |
|                |   |  | Full range    | 1.3  | 2    |                              |  |
| V(ON)          | Turnon voltage level  | TLV2770<br>TLV2773<br>TLV2775  | AV = 5        | 25°C | 1.47 | V                            |  |
|                |   |  |               |      | 1.43 |                              |  |
|                |   |  |               |      | 1.40 |                              |  |
| V(OFF)         | Turnoff voltage level   | TLV2770<br>TLV2773<br>TLV2775  | AV = 5        | 25°C | 1.27 | V                            |  |
|                |   |  |               |      | 1.21 |                              |  |
|                |   |  |               |      | 1.20 |                              |  |

TLV277x, TLV277xA  
FAMILY OF 2.7-V HIGH-SLEW-RATE, RAIL-TO-RAIL OUTPUT  
OPERATIONAL AMPLIFIERS

---

**operating characteristics at specified free-air temperature, VDD = 2.7 V (unless otherwise noted)**

| PARAMETER   | TEST CONDITIONS                                       | TAT†        | TLV277xC |         |        | UNIT   |
|---|---|-------------|----------|---------|--------|--------|
|   |   |             | MIN      | TYP     | MAX    |        |
| SR<br>Slew rate at unity gain                         | VO(PP) = 0.8 V, CL = 100 pF,<br>RL = 10 kΩ            | 25°C        | 5        | 9       |        | V/μs   |
|   |   | Full range  | 4.7      | 6       |        |        |
| Vn<br>Equivalent input noise voltage                  | f = 1 kHz   | 25°C        |          | 21      |        | nV/√Hz |
|   | f = 10 kHz  | 25°C        |          | 17      |        |        |
| VN(PP)<br>Peak-to-peak equivalent input noise voltage | f = 0.1 Hz to 1 Hz                                    |             |          | 0.33    |        | μV     |
|   | f = 0.1 Hz to 10 Hz                                   |             | 25°C     |         | 0.86   |        |
| In<br>Equivalent input noise current                  | f = 100 Hz  | 25°C        |          | 0.6     |        | fA/√Hz |
| THD + N<br>Total harmonic distortion plus noise       | RL = 600 Ω,<br>f = 1 kHz                              | AV = 1      |          | 0.0085% |        |        |
|   |   | AV = 10     | 25°C     |         | 0.025% |        |
|   |   | AV = 100    |          |         | 0.12%  |        |
| Gain-bandwidth product                                | f = 10 kHz,<br>CL = 100 pF                            | RL = 600 Ω, | 25°C     |         | 4.8    | MHz    |
| Ts<br>Settling time                                   | AV = -1,<br>Step = 1 V,<br>RL = 600 Ω,<br>CL = 100 pF | 0.1%        | 25°C     |         | 0.186  | μs     |
|   |   | 0.01%       | 25°C     |         | 0.3    |        |
| Φm<br>Phase margin at unity gain                      | RL = 600 Ω,   | CL = 100 pF | 25°C     |         | 46°    | dB     |
|   |   |             | 25°C     |         | 12     |        |

† Full range is 0°C to 70°C.

## Anexo C: Acreditación Conferencia

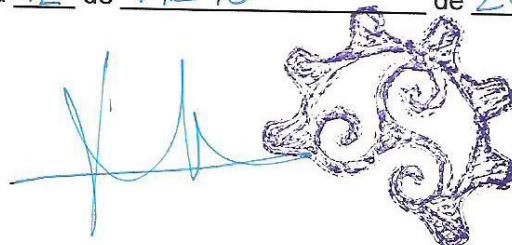


### OSHWDem 2013

«E-drum Low Cost»

CERTIFICA que **Manuel Ballesteros Carballo** ha presentado el proyecto «E-drum Low Cost» en OSHWDem 2013, celebrada los días 30 de noviembre y 1 de diciembre de 2013 en el Centro Sociocultural Ágora de A Coruña.

A Coruña, a 12 de MAYO de 2014



Fdo. XOÁN SAMPRANO VILAGRA, presidente de Bricolabs

## Anexo D: Código de programa

```

#define NPADS 6
#define CANALMIDI 9
#define NumeroDeMenus 6
#define NumeroMaximoDeSubmenus 4
#include <avr/interrupt.h>
#include <stdlib.h>      /* atoi */
#include <EEPROM.h>
#include <LiquidCrystal.h>

LiquidCrystal lcd(49, 48, 53, 52, 51, 50); //interfaz pins(RS,
E, D4, D5, D6, D7)
static char tituloMenu[NumeroDeMenus] [17] = {
    "Menu Principal ",
    "_submenu Utilidad",
    "_submenu Pad     ",
    "Parametro       ",
    "Guardando...   ",
    "Disparador MIDI ", //16 caracteres visibles más `\\0'
};

char tituloSubMenu[NumeroDeMenus] [NumeroMaximoDeSubmenus] [17] =
{
    "                         ,,"                         ,,"                         ,,"                         ,,"",
    ",," Utilidad      ,," Creditos      ,," Salir      ,," "",,
    "Pad de Bombo     ,," Pad de Caja     ,," Pad de HH     ,," Pad de
Crash      ,," Umbral          ,," Ganancia     ,," Mascara
    ,," Muestras      ,,
};

char borrador [17]= { "                         "}; //Utilizada para itoa

int encoderPin1 = 2; // 2,3 y 4 son pines de interrupción
int encoderPin2 = 3;

int encoderSwitchPin = 21; //switch interrupcion pin 21
static const byte stat_seq[]      ={3,2,0,1,3,2,0,1,3}; //Estados
encoder
static const byte invstat_seq[]={0,1,3,2,0,1,3,2,0}; //Estados
encoder al colocar inversor 74hc14

```

```

char stat_seq_ptr=4;
int i=0,j=0,k=0;
int counter = 0;
volatile int dir; //Guarda el pad de la eeprom- filas
int param; //Guarda el param de la eeprom- columnas
    //PIN 00,01,02,03,04,05
byte Note[]           = {36,38,46,48,41,50};
    //Kick, SD, HHClosed, Crash1, Tom1, Ride-
bell
unsigned int Threshold[] = {20,10,20,40,40,40};
byte Gain[]           = {10,4,5,4,4,4};
unsigned long MaskTime[] = =
{64000,64000,60000,64000,64000,68000};
unsigned long Muestras[] ={20,20,40,30,20,30};
byte FormOfCurve[] = {0,0,0,0,0,0};
unsigned long timer[NPADS];
int dato, maximo;
byte x, y, pad, canal, nota, velocidad;
byte pulsador1, pulsador2;
int ticks;
boolean pulsado;
//-----
// Rutina de Interrupcion [TIMER2_OVF_vect es la rutina que se
ejecuta cuando el Timer2 ->Overflow]
// Arduino tiene un cristal de 16 Mhz
// Por lo que tenemos una interrupcion cada => 1/ ((16000000 /
PRESCALER) / 256)
ISR(TIMER2_OVF_vect)
{
    ticks--;
}

void setup() {
    lcd.begin(16, 2);
    lcd.print("Disparador MIDI");
    pinMode(encoderPin1, INPUT);
    pinMode(encoderPin2, INPUT);
}

```

```

pinMode(encoderSwitchPin, INPUT);
//Llamadas a funciones de interrupcionn
attachInterrupt(0, updateEncoder, CHANGE);
attachInterrupt(1, updateEncoder, CHANGE);
attachInterrupt(2, updateSwitch, FALLING);

Serial.begin(31250);
// NORMAL MODE
TCCR2A = (1<<COM2A1) | (0<<COM2A0) | (0<<COM2B1) |
(0<<COM2B0) | (0<<3) | (0<<2) | (0<<WGM21) | (0<<WGM20);
//Preescaler 1024 cada 16ms OVF
TCCR2B = (0<<FOC2A) | (0<<FOC2B) | (0<<5) | (0<<4) |
(0<<WGM22) | (1<<CS22) | (1<<CS21) | (1<<CS20);
// Activo Timer2 Overflow Interrupt
TIMSK2 =(0<<7) | (0<<6) | (0<<5) | (0<<4) | (0<<3) |
(0<<OCIE2B) | (0<<OCIE2A) | (1<<TOIE2);
// Oscilador interno
ASSR = (0<<7) | (0<<EXCLK) | (0<<AS2) | (0<<TCN2UB) |
(0<<OCR2AUB) | (0<<OCR2BUB) | (0<<TCR2AUB) | (0<<TCR2BUB) ;
//Activo interrupciones globales.
sei();
pinMode(12,INPUT);
pinMode(11,INPUT);
pulsado=1;

ticks=127;
canal=CANALMIDI;
dato=0;
pad=0;
for (y=0;y<NPADS;y++)
    timer[y]=0;

//ADCSRA=ADCSRA&0xF8; //FADC=FCPU/32=500khz Ts=29us
//ADCSRA=ADCSRA|0x05; //Preescaler de 32
}

void loop() {
    do {

```

```

dato = analogRead(x);
pad=x;
x++;
if (x==NPADS) { //Miramos el HH
    x=0;
    pulsador1=digitalRead(12);
    if (pulsador1!=0) { //Si se pulsa el pedal viniendo
de reposo
        delay(2);
        pulsador2=digitalRead(11);
        if ((pulsador2==0) && (pulsado==1)) { //Con un me-
trónomo bpm 60
            delay(2);
            Note[2]=42; //Nota charles cerrado
            velocidad=ticks;
            if ((velocidad<-1) || (velocidad>127))
                velocidad=1;
            pulsado=0;
            ticks=127;
            notaOn(9, 44, velocidad);
        }
        else if (pulsador2==1) {
            pulsado=1;
            Note[2]=46; //Charles abierto
        }
    }
    else{
        ticks=0;
        Note[2]=46;
    }
}
if (micros()-timer[pad]>MaskTime[pad]) //Masktime
{
    timer[pad]=0;
}
}
while (dato<Threshold[pad]);

```

```

if(timer[pad]==0) //Scan Time
{
    for (y=0;y<Muestras[pad];y++) { //Recoger x muestras
        maximo= analogRead(pad); // y saca el maximo
        if (maximo>dato)
            dato=maximo;
    }
    //dato=dato-400; offset
    //dato=dato*(Gain[pad]/8);
    velocidad=byte (dato>>3); //Equivalente a dividir entre 8
    velocidad=velocidad*Gain[pad];
    if(velocidad>127)
        velocidad=126;
    nota=Note[pad];
    notaOn(canal,nota,velocidad);
    timer[pad]=micros();
}
}

void notaOn(byte channel, byte note, byte velocity) {
    MIDIMsg( (0x90 | channel), note, velocity);
}

void notaOff(byte channel, byte note, byte velocity) {
    MIDIMsg( (0x80 | channel), note, velocity);
}

void MIDIMsg(byte cmd, byte data1, byte data2) {
    Serial.write(cmd);
    Serial.write(data1);
    Serial.write(data2);
}

void updateEncoder(){//Bloquear sino se utiliza //CAMBIAR A BYT
    int MSB = digitalRead(encoderPin1); //MSB = most significant
    bit
    int LSB = digitalRead(encoderPin2); //LSB = least significant
    bit
    int encoded = (MSB << 1) | LSB; //converting the 2 pin value to
    single number
    if (encoded==invstat_seq[stat_seq_ptr+1]){ //Incremento

```

```

        stat_seq_ptr++;           //Movimiento a derecha
        if (stat_seq_ptr==8) {
            stat_seq_ptr=4;
            i++;
            counter++;
            if(i>3)
                i=3;
            if (counter>255)
                counter=255;
            lcd.setCursor(0, 1);
            lcd.print(tituloSubMenu[j][i]);
        }
    }

else if (encoded==invstat_seq[stat_seq_ptr-1]) { //Decremento
    stat_seq_ptr--;           //Movimiento a izquierda
    if (stat_seq_ptr==0) {
        stat_seq_ptr=4;
        i--;
        counter--;
        if(i<0)
            i=0;
        if (counter<0)
            counter=0;
        lcd.setCursor(0, 1);
        lcd.print(tituloSubMenu[j][i]);
    }
}

if(j==4) {
    lcd.setCursor(3, 1);
    lcd.print(counter);
}

void updateSwitch() {
    switch (j) {
        case 0:      //Menu Principal
        i=0;        //Utilidad
        lcd.setCursor(0, 0);
}

```

```

lcd.print(tituloMenu[j]);
lcd.setCursor(0, 1);
lcd.print(tituloSubMenu[1][0]);
j++;
break;
case 1: {
    switch (i) {
        case 0:
            lcd.setCursor(0, 0);
            lcd.print(tituloMenu[j]); //Submenu Utilidad
            lcd.setCursor(0, 1);      //Pad de bombo
            lcd.print(tituloSubMenu[2][0]);
            i=0;
            j++;
            break;
        case 1:
            lcd.setCursor(0, 0);
            lcd.print("Elaborado por:");
            lcd.setCursor(0, 0);
            lcd.print("      Manuel      ");
            lcd.setCursor(0, 1);
            lcd.print(" Ballesteros   ");
            break;
        case 2:
            lcd.clear();
            lcd.setCursor(0, 1);
            lcd.print("Opcion no valida");
            break;
    }
}
break;
case 2:
dir=i;
for(i=0;i<4;i++) {
    if(dir==0)
        k=i;
    else

```

```

        k=((4*dir)+i);
        counter= EEPROM.read(k); //Recoger todos los paráme-
ters
        itoa(counter, borrador, 10); //Convierte un entero
        tituloSubMenu[3][i][9]=borrador[0];
        tituloSubMenu[3][i][10]=borrador[1];
        if(counter>100)
            tituloSubMenu[3][i][11]=borrador[2];
        else
            tituloSubMenu[3][i][11]=' ';
    }
    i=0;
    lcd.setCursor(0, 0);
    lcd.print(tituloMenu[j]); //Submenu Pad
    lcd.setCursor(0, 1);      //Ganancia
    lcd.print(tituloSubMenu[3][0]);
    j++;
    break;
case 3:
    param=i;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(tituloMenu[j]); //Parametro
    if(dir==0)
        k=param;
    else
        k=(4*dir)+param;
    counter = EEPROM.read(k);
    lcd.setCursor(3, 1);
    lcd.print(counter);
    j++;
    break;
case 4:
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(tituloMenu[j]); //Guardando...
    EEPROM.write(k,counter);

```

```

switch(param) {
    case 0:
        Threshold[dir]=counter;//Threshold
        break;
    case 1:
        Gain[dir]=counter;//Gain
        break;
    case 2:
        MaskTime[dir]=counter;//Mask Time
        MaskTime[dir]=MaskTime[dir]*1000;
        break;
    case 3:
        Muestras[dir]=counter;//Muestras
        break;
}
j++;
break;
case 5:
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(tituloMenu[j]); //Disparador MIDI
    j=0;
    break;
}
}

```

## Anexo E: Preguntas frecuentes -Manual del usuario [Drola]

| <b>Problema</b>  | <b>Elemento que comprobar</b>  | <b>Acción</b>   |
|--|--|---|
| <b>La unidad no se enciende</b>                                  | ¿Está bien conectado el cable de alimentación?   | Conecte el adaptador de CA correctamente.   |
| <b>No se oye sonido</b>  | ¿Puede haberse soltado un cable interno?<br>¿Están los cables de todos los pads conectados correctamente?  | Revise las conexiones<br>Asegúrese de que las conexiones de los pads se han realizado correctamente y de que cada pad está conectado a su conector correspondiente. |
|  | ¿El software de batería está bien inicializado?  | Reinicie el ordenador y comience de nuevo las instrucciones.  |
| <b>Se produce un retraso en el sonido</b>                        | ¿Valor adecuado del parámetro Scan Time?<br>¿Valor adecuado en los driver ASIO?  | Compruebe los ajustes de sensibilidad tabla 17.<br>Vaya al panel de control y coloque un valor de 64 en muestras por paquete.                                       |
| <b>El sonido es demasiado bajo/ alto durante la reproducción</b> | ¿Podría estar el mando [VOLUME] totalmente girado a la izquierda?<br>¿Podría estar la sensibilidad del pad ajustada demasiado baja o demasiado alta? | Gire el mando [VOLUME] hacia la derecha.<br>Compruebe los ajustes de sensibilidad tabla 17.   |
| <b>Se produce un disparo doble</b>                               | ¿Podría un cable estar tocando un pad?<br>¿Valor adecuado del parámetro Mask Time?   | Aleje los cables<br>Compruebe los ajustes de sensibilidad tabla 17.   |