

MiDispositivoMIDI v2.0



Guía de montaje y programación

Este documento aborda paso a paso el proceso de montaje y programación del controlador MIDIspositivoMIDI V2, proyecto desarrollado por Pablo de Miguel Morales y Álvaro López Revuelta.

midispositivomidi@gmail.com
facebook.com/midispositivomidi

Asociación de Sonido. AST
[www.facebook.com/sonidoteleco](https://facebook.com/sonidoteleco)

Asociación de electrónica. AETEL
[www.facebook.com/pages/AETEL-IEEE-SB-UPM-Campus-Sur/200916653259710](https://facebook.com/pages/AETEL-IEEE-SB-UPM-Campus-Sur/200916653259710)



Índice

1. Materiales	5
1.1. Material Electrónico	5
1.2. Material Hardware.....	6
2. Montaje MiDispositivoMIDI v2.0	6
2.1. Soldadura de placa PCB Breakout 4x4 <i>PAD</i>	6
<i>Paso 1: Soldadura de diodos 1N4048</i>	<i>6</i>
<i>Paso 2: Soldadura de LEDs RGB</i>	<i>7</i>
<i>Paso 3: Soldadura de pines 1x4</i>	<i>8</i>
2.2. Soldadura de placa PDB Control	9
<i>Paso 1: Soldadura de resistencias de LEDs.....</i>	<i>9</i>
<i>Paso 2: Soldadura Circuitos Integrados y los condensadores</i>	<i>9</i>
<i>Paso 3: Soldadura Switch y Reset externo</i>	<i>10</i>
<i>Paso 4: Soldadura de los zócalos 1x4.....</i>	<i>10</i>
<i>Paso 5: Soldadura de los conectores y jumper MODE.....</i>	<i>11</i>
<i>Paso 6: Soldadura de los pines de conexión a Arduino UNO.....</i>	<i>12</i>
<i>Paso 7: Soldadura de los potenciómetros.....</i>	<i>12</i>
2.3. Montaje mecánico	13
3.1. <i>Ensamblaje mecánico</i>	<i>13</i>
<i>Paso 1: Unión del PAD de silicona, el protector y la placa superior</i>	<i>13</i>
<i>Pase 2: Colocación de las patas</i>	<i>14</i>
3.2. <i>Colocación del embellecedor del potenciómetro</i>	<i>15</i>
3. Programación en Arduino de MidispositivoMIDI	16
3.1 Instalar entorno de desarrollo Arduino	17
<i>MacOS.....</i>	<i>17</i>
<i>Windows</i>	<i>17</i>
3.2 Entorno Arduino	18
3.3 Instalar librerías	19
<i>MacOS.....</i>	<i>19</i>
<i>Windows</i>	<i>19</i>
3.4 Uso de ejemplos	21
3.5 Flashear Arduino.....	21
3.5.1 <i>Windows</i>	<i>22</i>
3.5.1.2 <i>Solucionando problema. Forma 1, descargar los drivers de la página</i>	<i>25</i>
3.5.1.3 <i>Solucionando problema. Forma 2 descargar programa Zadig.....</i>	<i>27</i>
3.5.1.4 <i>Terminar de flashear.....</i>	<i>29</i>
3.5.1.5 <i>Anexo. No reconoce puerto COM.....</i>	<i>30</i>
3.5.2 <i>Mac y Linux</i>	<i>32</i>
3.6 Hazlo sonar. MiDispositivoMidi Sampler v2.0	34
4. Anexos	35
4.2 ¿Qué es MIDI?.....	36
4.3. Esquemático MiDispositivoMIDI.....	37
5. Referencias	39

Índice de figuras

Figura 1. Lista de componentes electrónicos	5
Figura 2. Lista de componentes mecánicos	6
Figura 3. Soldando diodos	6
Figura 4. Esquema diodo. Ánodo/Cátodo	7
Figura 5. Leds RGB soldados	7
Figura 6. Esquema diodo RGB, patillas	7
Figura 7. Conexionado leds RGB. Patilla común	8
Figura 8. Leds soldados, detalle	8
Figura 9. Pines soldados.....	8
Figura 10. Pines hembra 3D	9
Figura 11. Soldando ICs. Izquierda, real. A la derecha modelo 3D	9
Figura 12. Soldando condensadores 3D	10
Figura 13. Soldando switches	10
Figura 14. Placa “bottom” finalizada	11
Figura 15. Soldando pines macho dobles	11
Figura 16. Conexionado de Arduino	12
Figura 17. Soldado de potenciómetros.....	12
Figura 18. Ensamblaje mecánico.	13
Figura 19. Colocando tornillos de ensamblaje.....	14
Figura 20. Colocación de patas	15
Figura 21. Colocación del embellecedor (knob)	15
Figura 22. Carpeta Arduino MAC	17
Figura 23. Sketch Arduino Windows.....	18
Figura 24. Barra Sketch Arduino	18
Figura 25. Contenido carpeta	19
Figura 26. Importando librería en Windows.....	20
Figura 27. Flasheando en Windows. Flip	22

Figura 28. Conexionado del jumper para flashear.....	23
Figura 29. Icono para flasheo.....	23
Figura 30. Selección de dispositivo para flasheo	24
Figura 31. Selección de puerto usb.....	24
Figura 32. Posible error.....	24
Figura 33. Soluciando error.....	25
Figura 34. Solucionando error	26
Figura 35. Solucionando error	26
Figura 36. Solucionando error	27
Figura 37. Solucionando error	28
Figura 38. Solucionando error	28
Figura 39. Solucionando error	29
Figura 41. Problema puerto COM Windows.....	31
Figura 42. Instalando dirver windows.....	31
Figura 43. Instalando drivers	32
Figura 44. Verificando el funcionamiento.....	32
Figura 45. Conexionado jumper.....	33
Figura 46. Terminal, comandos para flasheo.....	33
Figura 47. Configuración midi en Ableton Live	34
Figura 48. Programa Sampler	34
Figura 49. Para soldar (de instructables)	35



1. Materiales

1.1. Material Electrónico

CANTIDAD	COMPONENTE
16	DIODO 1N4048 INSERCIÓN
16	LED RGB INSERCIÓN
05	ZOCALO DIL16
02	CI 74HC595 DIL16
03	CI 74HC4051 DIL16
03	CAPACITOR 100nF INSERCIÓN
02	POTENCIOMETRO 5kΩ INSERCIÓN
02	EMBELLECEDOR POTENCIOMETRO
02	SWITCH INSERCIÓN
14	RESISTENCIA 220Ω INSERCIÓN
02	TIRA 40 PINES MALE
01	TIRA 30 PINES FEMALE
02	JUMPER
16	DIODO 1N4048 INSERCIÓN
16	LED RGB INSERCIÓN
03	ZOCALO DIL16
02	CI 74HC595 DIL16
01	CI 74HC4051 DIL16
01	PCB 4x4 PAD SPARKFUN
01	PCB MIDIspositivoMIDI V2
01	PAD SILICONA
01	PAD PROTECTOR
01	ARDUINO UNO + CABLE

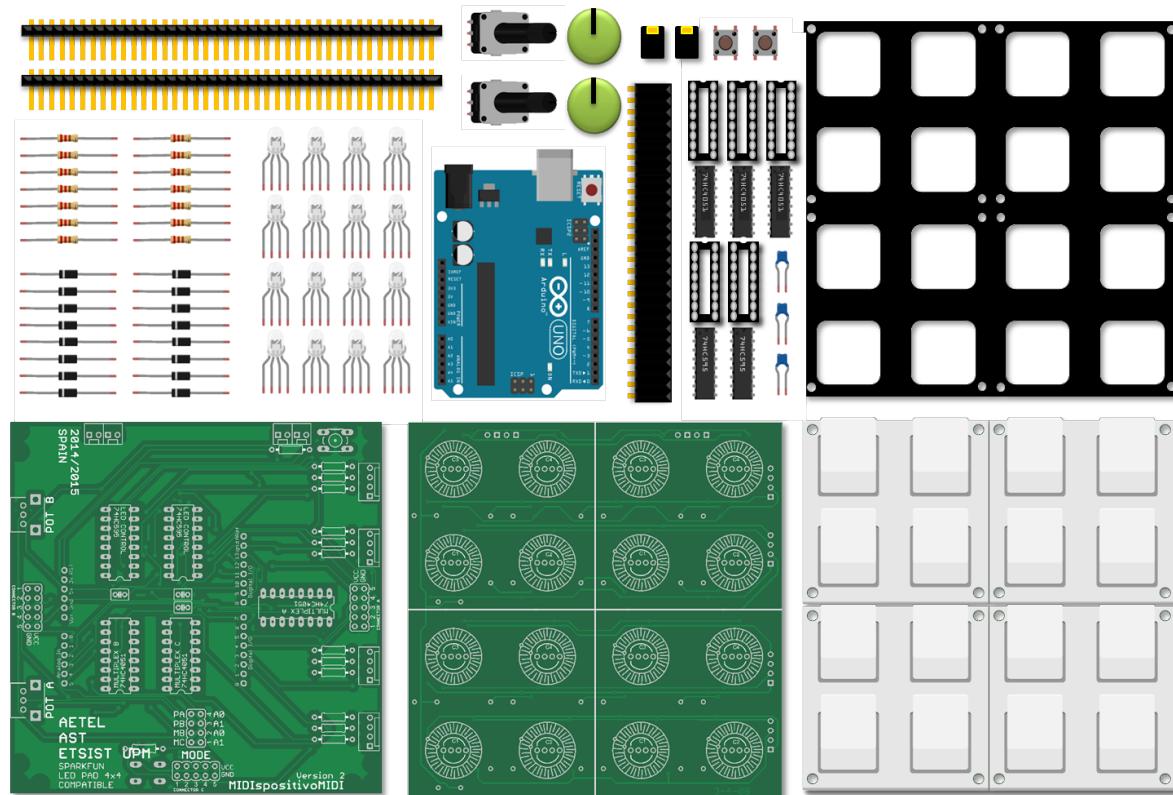


Figura 1. Lista de componentes electrónicos

1.2. Material Hardware

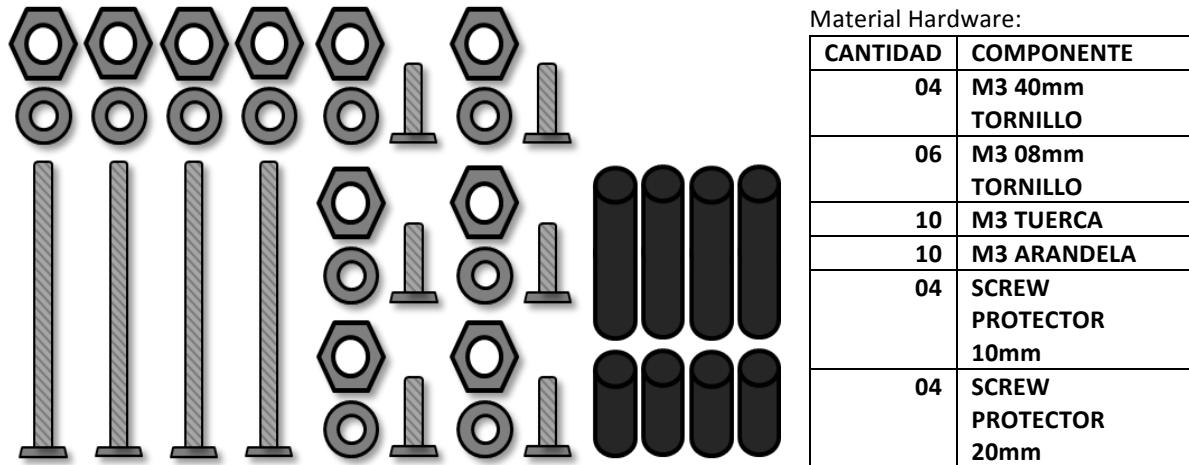


Figura 2. Lista de componentes mecánicos

2. Montaje MiDispositivoMIDI v2.0

En esta sección se aborda la soldadura de las dos placas PCB, tanto la superior como la inferior. Es la sección mas larga del proceso de montaje y su duración es de aproximadamente 40 minutos. Sin embargo, esta duración puede variar en función de la experiencia del usuario a la hora de soldar. Para usuarios que no hayan soldado nunca, se incluye una breve guía del proceso de soldado con el fin de facilitar el proceso. En cualquier caso se recomienda una primera sesión de soldado tutelada para un mejor aprendizaje y resultado.

2.1. Soldadura de placa PCB Breakout 4x4 PAD

Paso 1: Soldadura de diodos 1N4048

Los diodos 1N4048 tienen el objetivo de evitar el efecto de *ghosting* que afecta a las matrices de botones. Su funcionamiento se aborda con mas detalle en el manual de programación del taller. Hay un total de 16 diodos, uno para cada PAD de la matriz.

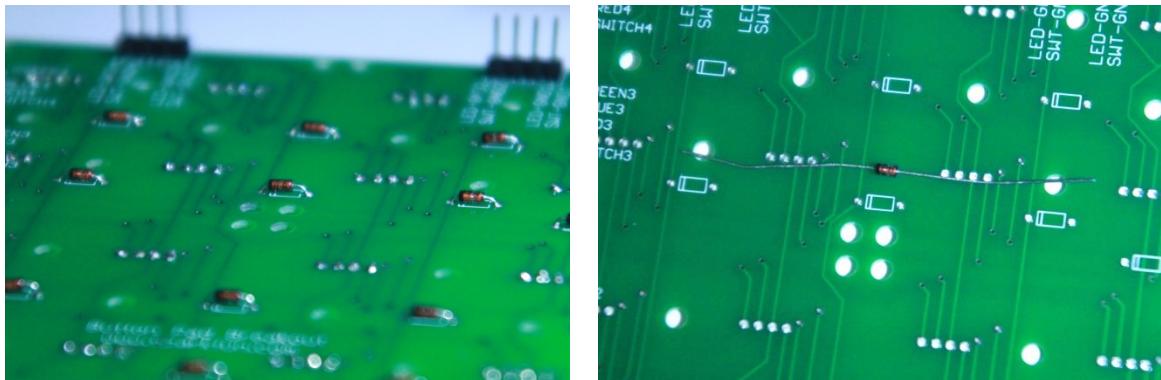


Figura 3. Soldando diodos

Su orientación está marcada en la serigrafía de la PCB, y coincide con el Cátodo marcado en el propio encapsulado del diodo 1N4048. Se deben insertar en la cara posterior de la placa y soldarlos por la cara superior para que no levanten la silicona de los PADs.

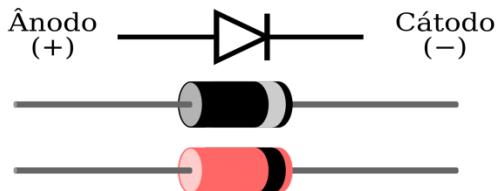


Figura 4. Esquema diodo. Ánodo/Cátodo

En la cara superior se aprecia el corte de las patas del diodo al ras de la PCB, de modo que no suponga obstáculo para los PADs de silicona.

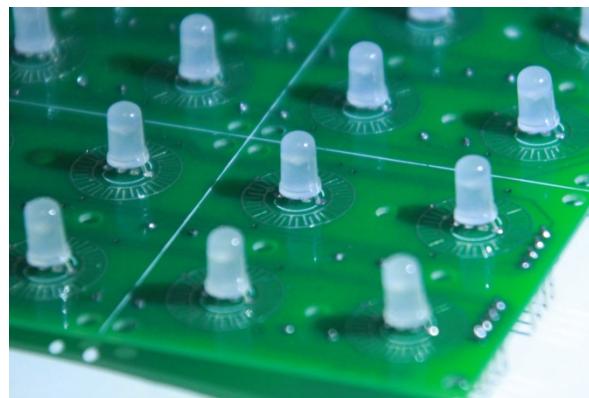


Figura 5. Leds RGB soldados

Una vez hayan sido soldados los 16 diodos 1N4048 se puede continuar con siguiente paso.

Paso 2: Soldadura de LEDs RGB

Los LEDs RGB son posiblemente la característica más atractiva del controlador MIDIspositivoMIDI y por lo tanto su montaje y correcto soldado exigen gran atención. Los LEDs RGB utilizados pueden ser tanto de *ánodo común* como de *cátodo común*, teniendo que ser esto especificado durante la programación. Ambos modelos de LEDs son similares en apariencia pero su funcionamiento es muy distinto:

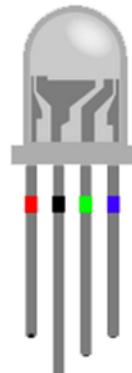
El funcionamiento de un LED RGB de Ánodo Común es similar al de 3 LED RED, BLUE y GREEN cuyo Ánodo está conectado a un mismo nodo.

Para encender un determinado color, por lo tanto, es necesario excitar el LED colocando en el Ánodo una tensión positiva con respecto a cada uno de los Cátodos de los distintos colores.

En la siguiente tabla se muestran por lo tanto las posibles combinaciones que se pueden lograr en función de la tensión que se coloque en los pines:

	OFF	R	G	B	RG	RB	BG	RGB
ANODE	0V	5V						
PIN R	X	0V	5V	5V	0V	0V	5V	0V
PIN G	X	5V	0V	5V	0V	5V	0V	0V
PIN B	X	5V	5V	0V	5V	0V	0V	0V

R G B LED



COMMON ANODE

Figura 6. Esquema diodo RGB, patillas

Debido al funcionamiento de los LEDs, es necesario orientar bien el componente antes de soldarlo. Para esto, primero hay que localizar cual es el Ánodo/Cátodo de los LEDs. El Ánodo/Cátodo de los LEDs se corresponde con su patilla más larga, similar al dibujo.

Una vez localizados los Ánodo/Cátodo, se localizan los PADs (Los agujeros de la placa en los que se sueldan componentes) que corresponde a los Ánodos en la parte inferior de la placa.

En esta imagen se ha marcado en rojo un de las pistas de *Ánodo/Cátodo* común de la columna de *LED RGB*. Lo que diferencia las 4 pistas de *Ánodo/Cátodo* común de la placa (Correspondientes a las 4 columnas de la matriz de *LEDs*) es que la misma pista conecta todos los *PADs*.

Es necesario por lo tanto colocar los *LEDs* de tal modo que el *Ánodo/Cátodo* (pata más larga del *LED*) quede en esos *PADs* que comparten pista. El resto del *LED* solo tiene una colocación natural, que es correcta.

Si existen dudas sobre como colocar los *LEDs RGB* es mejor consultar con alguien que sepa cómo hacerlo antes de arriesgarse a soldar, ya que luego, una vez estén las patillas cortadas, es más complicado localizar el *Ánodo/Cátodo Común*.

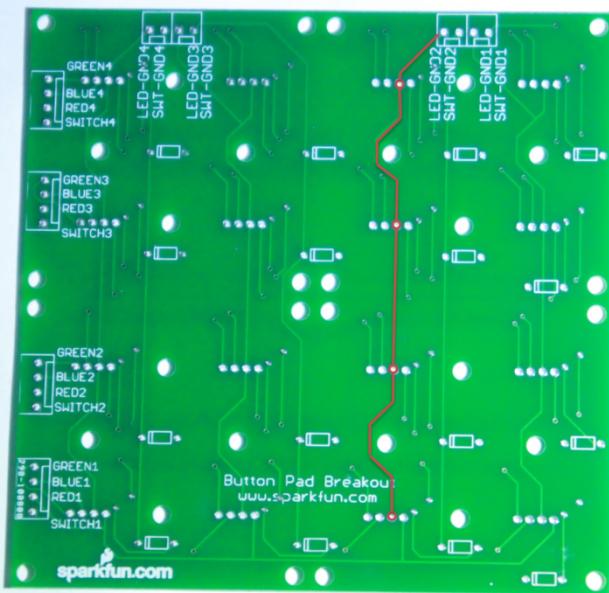


Figura 7. Conexionado leds RGB. Patilla común



Figura 8. Leds soldados, detalle

Es importante que el led quede lo suficientemente metido en los agujeros. Si asoma demasiado, al poner el pad de goma por encima podría llegar a chocar con el led y no llegar a tocar la placa, lo que provocaría que no se detectara el botón pulsado. Es pues importante ajustar los leds a la placa todo lo posible. **Antes de soldar todo, comprobar con un led y el pad que efectivamente no toca con la parte interior del pad.**

Paso 3: Soldadura de pines 1x4

Para la unión de las dos placas se ha optado por usar pines en detrimento de cables (Que no dan más que problemas). Para soldar los pines es necesario en primer lugar cortarlos en 6 tiras de 4 pines. Para este proceso se puede usar un pequeño alicate o incluso hacerlo con las manos, ya que los pines están torneados

Las tiras de pines deben colocarse de modo que la soldadura se aplique por la parte superior de la placa y la parte larga de los pines quede en la parte inferior (Que irá conectada a la placa intermedia)

Una vez haya sido soldados las 6 tiras 1x4 pines se habrá finalizado la soldadura de la placa superior y se debe proceder a soldar la placa intermedia.

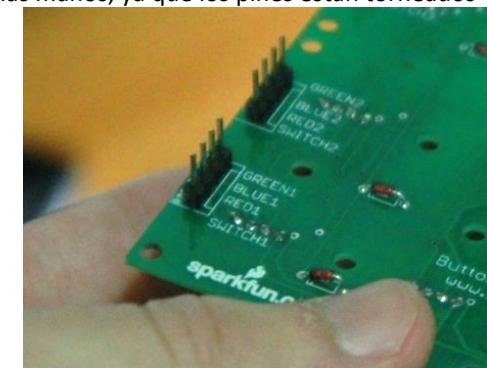


Figura 9. Pines soldados

2.2. Soldadura de placa PDB Control

Paso 1: Soldadura de resistencias de LEDs

Para controlar la corriente máxima que atraviesa el LED y evitar su degradación progresiva o instantánea, es necesario polarizarlo con ayuda de una resistencia. Para este propósito se van a utilizar resistencias de inserción 220Ω 5%TOL.

Es por lo tanto necesario insertar estas resistencias por la cara superior y soldarlas por la cara inferior. Para localizar los PADs correctos, se propone situar la placa de frente de modo que se lea MIDIspositivoMIDI correctamente. Una vez hecho eso, se deben localizar 4 grupos de resistencias en el lado derecho, cada grupo conformado por 3 *Footprints* de resistencias.

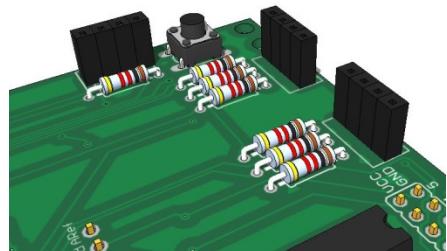


Figura 10. Pines hembra 3D

Una vez hayan sido soldadas las 12 resistencias, se puede continuar al siguiente paso.

Paso 2: Soldadura Circuitos Integrados y los condensadores

La placa de control hace uso de 5 circuitos integrados para el control del dispositivo. Estos son 2 74HC595 (Registro de desplazamiento de 8 bits) y 3 74HC4051 (Multiplexor Analógico 8to1). Su funcionamiento se abordará en la etapa de programación y por ahora no se debe contemplar su problemática.

Para evitar su deterioro durante el soldado, los CI (Circuitos Integrados) se montan sobre zócalos, que no cumplen ninguna función electrónica. Para orientar correctamente los CI, la serigrafía incluye una pestaña de orientación en forma de muesca. Se observa que tanto el CI como el zócalo comparten esta característica. Es necesario, para cada CI, que estas tres señales sean coherentes.

Es mas el planteamiento que es concepto es similar al del año anterior, muy conciso detallando los pasos y con fotos.

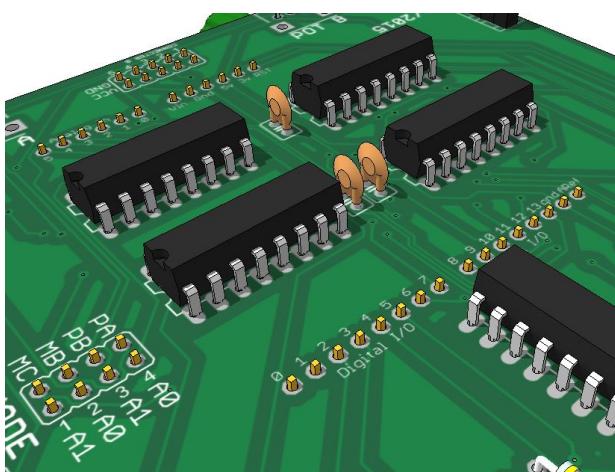
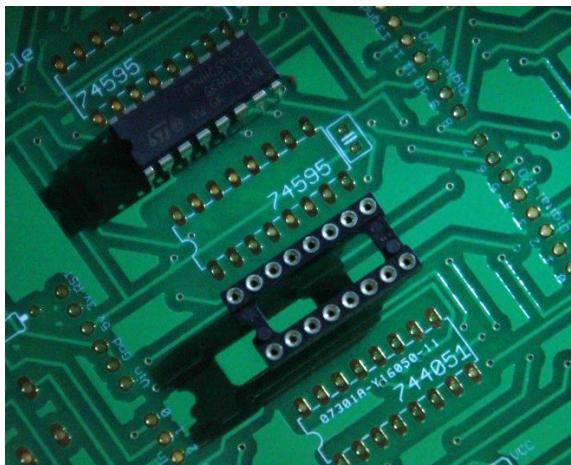


Figura 11. Soldando ICs. Izquierda, real. A la derecha modelo 3D

Los condensadores de $100\mu F$ que se deben soldar junto con los CI cumplen funciones de desacoplamiento, reduciendo el consumo y el ruido de alimentación durante la conmutación de los circuitos *CMOS* y mejorando por lo tanto el rendimiento. Se deben soldar en los *PADs* junto a los CI insertándose por la cara superior y soldándose por la inferior.

Una vez hayan sido soldados los 5 zócalos y los 3 condensadores, y los CI hayan sido encajados en los zócalos, se puede continuar con el siguiente paso.

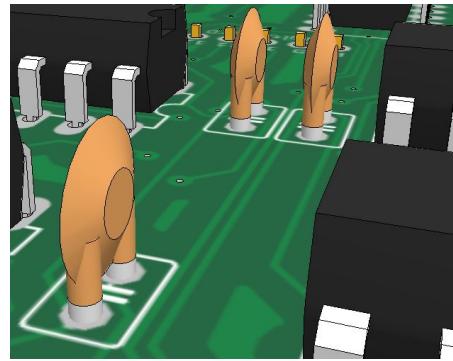


Figura 12. Soldando condensadores 3D

Paso 3: Soldadura Switch y Reset externo

La placa hace uso de 2 *switch* táctil con una resistencia de *Pull-Down* de 220Ω . Uno de los *switch* se usará durante el control del dispositivo mientras que el otro es tan solo un *reset* externo de *Arduino*, ya que el integrado en la propia placa *Arduino UNO* no se encuentra accesible. La resistencia se debe insertar por la cara superior y ser soldada por la cara inferior mientras que el *switch* debe ser insertado por la cara inferior y soldados por la cara superior, siendo el resultado el siguiente:

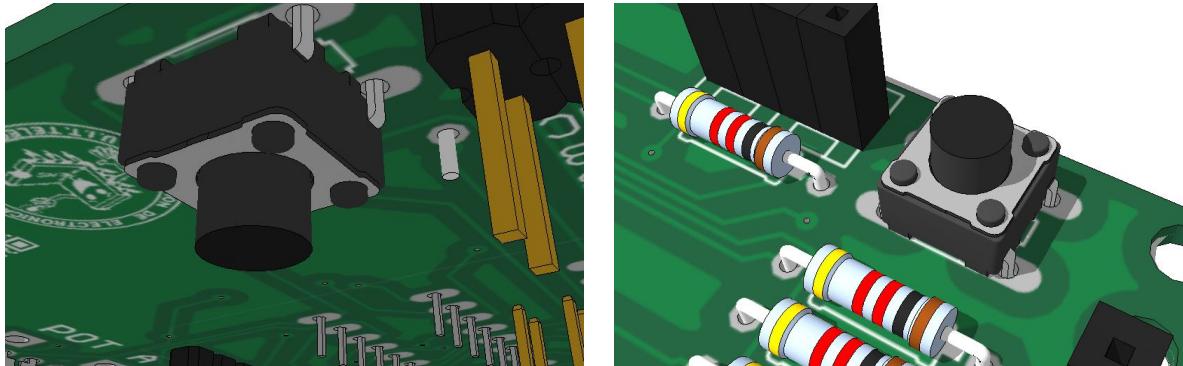


Figura 13. Soldando switches

En el caso del *switch* de *reset*, se debe insertar por la cara superior y soldar por la cara inferior. Se puede soldar para que pueda ser presionado por la cara inferior pero puede dar lugar a *reseteos* accidentales. Una vez hayan sido soldados ambos *switches* y las resistencias de *Pull-Down*, se puede continuar con siguiente paso.

Paso 4: Soldadura de los zócalos 1x4

Para conectar ambas placas sin cables, se ha decidido hacerlo mediante pines. En la placa PCB superior se han soldado ya los pines que deben conectarse a tiras de zócalos en la placa PCB de control.

Se deben cortar tiras de 1x4 zócalos y soldarlas en los 6 *PADs* que delimitan la placa.

El corte de los zócalos no es tan sencillo como el de los pines y tienden a partirse. Para evitarlo se recomienda tratar de cortarlos con una cuchilla en vez de con unas tenazas de corte. En ningún caso se debe tratar de partirlos con las manos.

Para facilitar la soldadura se pueden introducir en los pines ya soldados de la placa superior y apoyarla en la placa inferior.

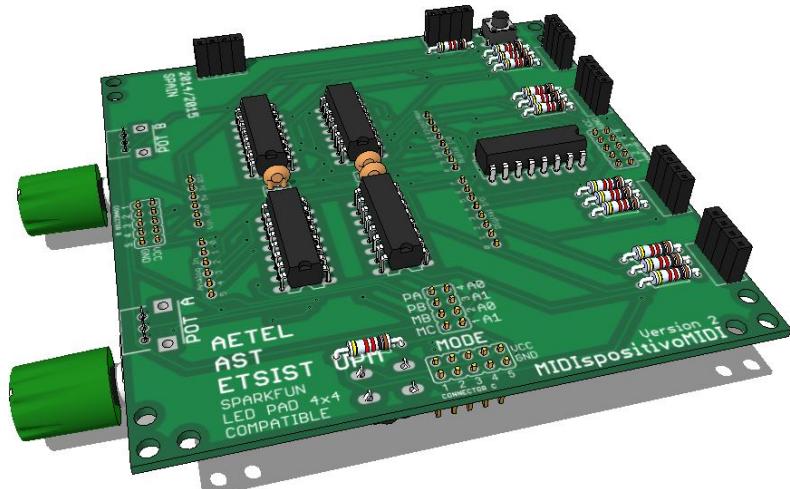


Figura 14. Placa “bottom” finalizada

Una vez hayan sido soldadas las tiras de zócalos 1x4, se puede continuar con el siguiente paso.

Paso 5: Soldadura de los conectores y jumper MODE

Para la conexión de placas de extensión se usan conectores formados por 02x05 pines. Estos pines se deben soldar en los 3 *PADs* situados en tres de los laterales de la *PCB*. Se deben introducir por la cara inferior y soldar por la superior.

Los jumper de *MODE* son necesarios a la hora de usar más de una placa de expansión, teniéndose que desconectar los potenciómetros laterales para asociar los pines *ANALOG* a los multiplexores *MUX B* y *MUX C*. Para esta función se usa una tira formada por 02x04 pines que debe ser introducida por la parte inferior y soldada por la superior para que pueda ser manipulada sin la necesidad de desmontar el dispositivo completo.

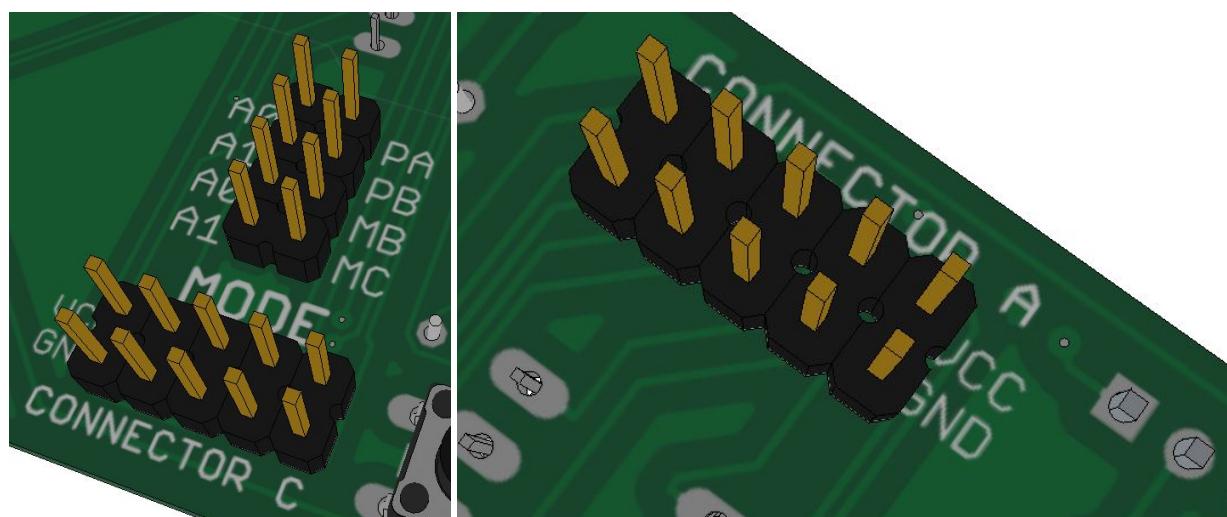


Figura 15. Soldando pins macho dobles

Una vez hayan sido soldados los conectores y los jumper *MODE*, se puede continuar con el siguiente paso.

Paso 6: Soldadura de los pines de conexión a Arduino UNO

La conexión con el *Arduino* se hace a través de pines por la cara inferior. De manera similar al paso 3 de la placa de LEDs, se deben cortar tiras de pines adecuadas para que se ajusten a las tiras de agujeros. El resultado debe ser similar al mostrado.

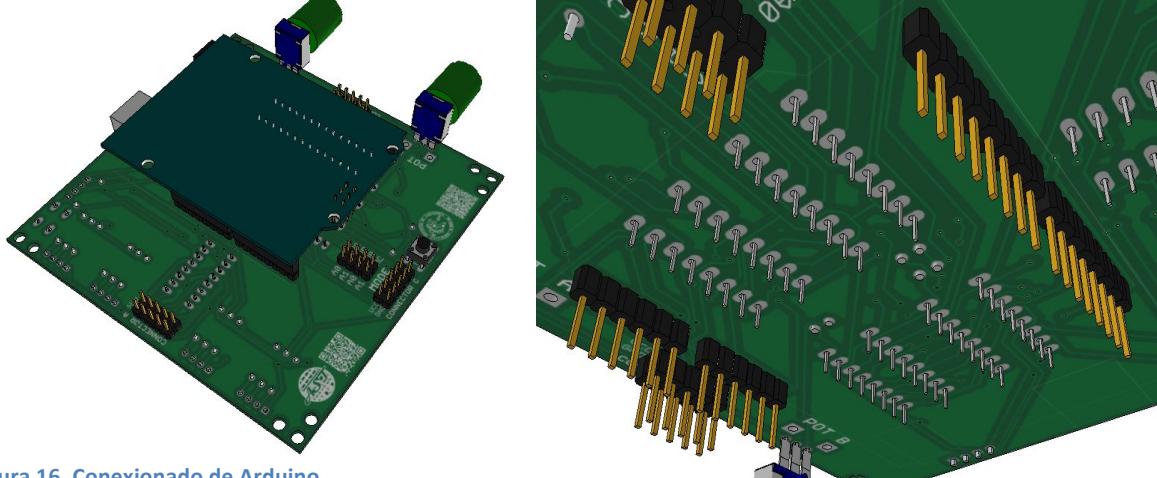


Figura 16. Conexionado de Arduino

Una vez hayan sido soldados todos los pines de conexión con el *Arduino*, se puede continuar al siguiente paso.

Paso 7: Soldadura de los potenciómetros

Se han incluido dos potenciómetros para control de señales *fader* tales como volumen, *pitch* o cualquier señal que el usuario decida manejar a través de ellos. Los potenciómetros se introducen por la cara inferior y se sueldan por la superior, de modo que aparezcan como en la imagen.

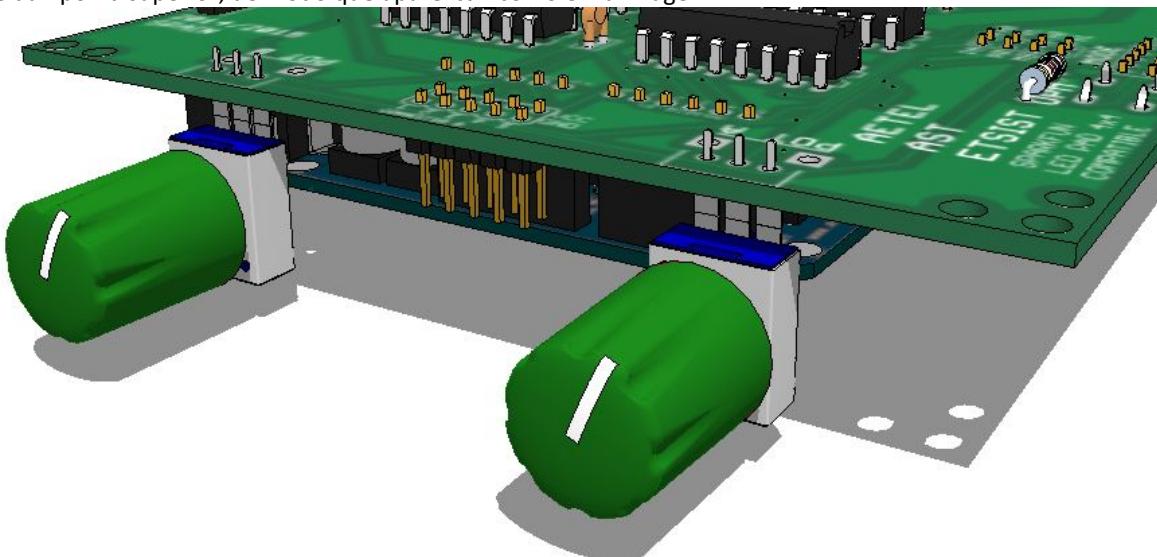


Figura 17. Soldado de potenciómetros

Una vez se hayan soldado los potenciómetros, se habrá finalizado el proceso de soldado de ambas placas.

2.3. Montaje mecánico

3.1. Ensamblaje mecánico

De manera esquemática se puede ver en la siguiente simulación 3D el controlador con todas sus partes ordenadas en la proyección de su posición final:

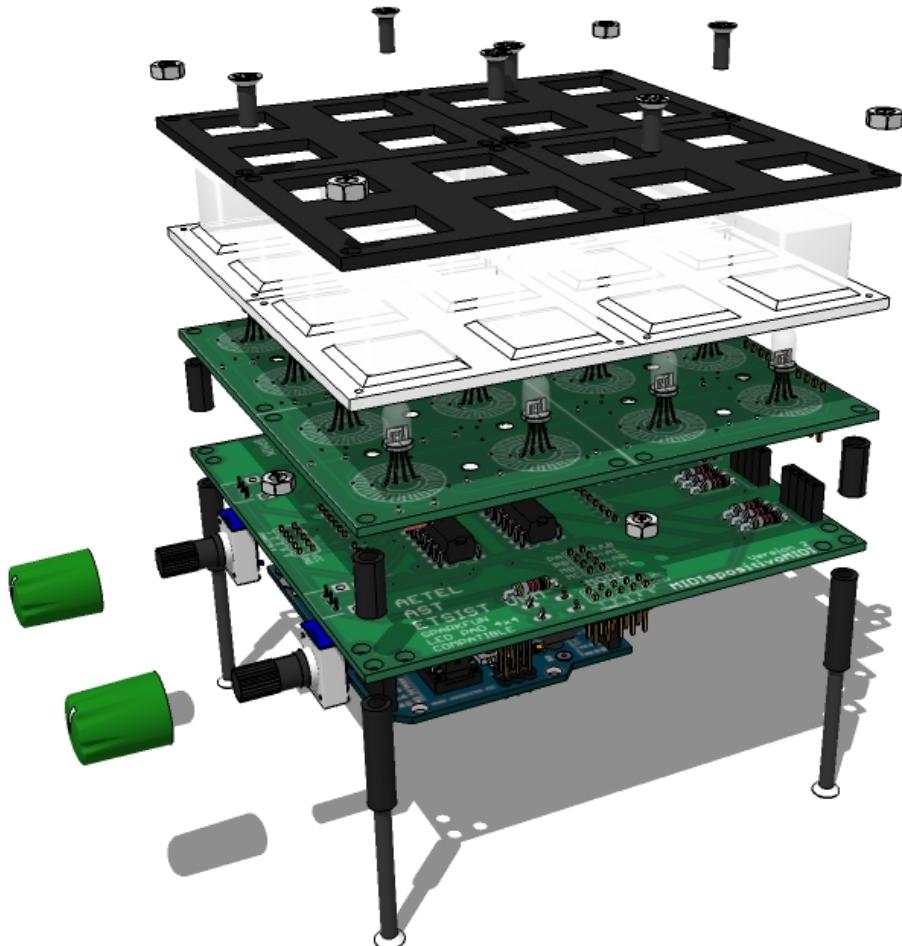


Figura 18. Ensamblaje mecánico.

Paso 1: Unión del PAD de silicona, el protector y la placa superior

Se debe unir el PAD de silicona a la placa y situar encima el protector. El protector, además de su función estética, tiene el objetivo de fijar mejor la botonera haciendo más fácil su uso. Para la unión de estas tres secciones se deben usar 6 tornillos M3 8mm junto con sus tuercas. Tanto el protector como la placa disponen de un total de 12 agujeros para esta unión, pero se considera que 6 son más que suficientes.

Se deben usar los agujeros centrales y los laterales no esquinados, ya que los esquinados serán usados para las patas principales.

El uso de arandelas junto con las tuercas es posible pero para nada necesario.

Una vez se hayan unidos estas tres secciones, se puede continuar con el proceso de ensamblaje mecánico.

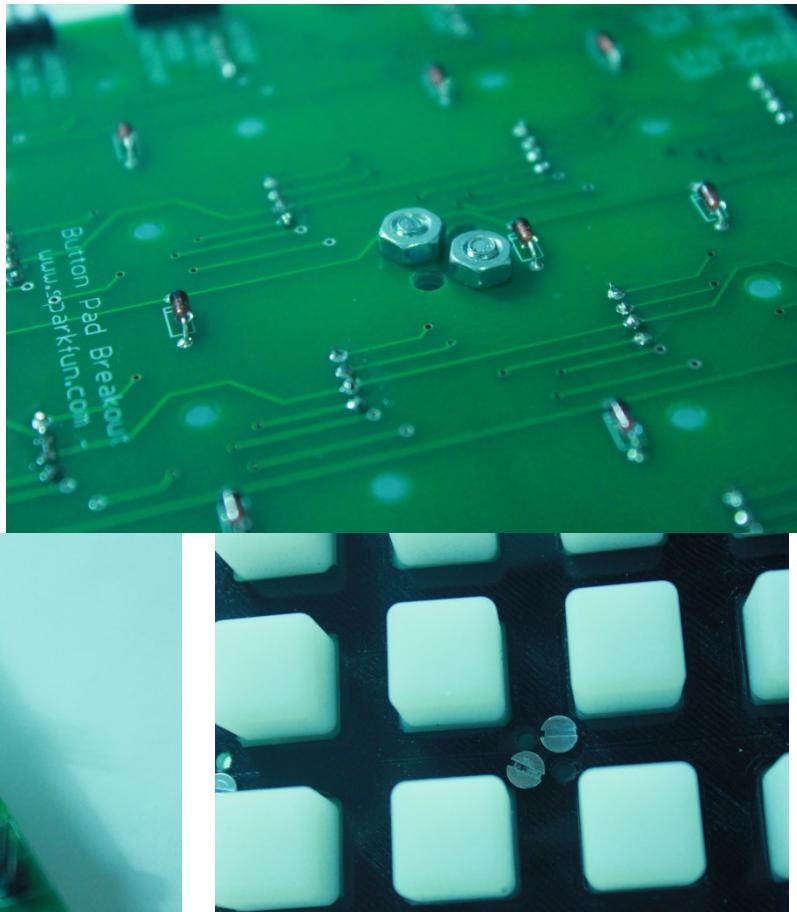


Figura 19. Colocando tornillos de ensamblaje.

Pase 2: Colocación de las patas

Para finalizar el proceso de ensamblaje se añaden las patas. Para hacerlo primero se unen ambas placas a través de los pines de la única manera posible. Una vez hecho, se añade a un tornillo M3 40mm un protector negro 20mm. Este conjunto se introduce por la parte inferior del dispositivo en una de las esquinas mientras se mantiene un protector de 20mm para que quede entre ambas placas. Una vez introducirlo el tornillo con los protectores se atornilla por la cara superior con una tuerca y una arandela para que quede bien sujetado.

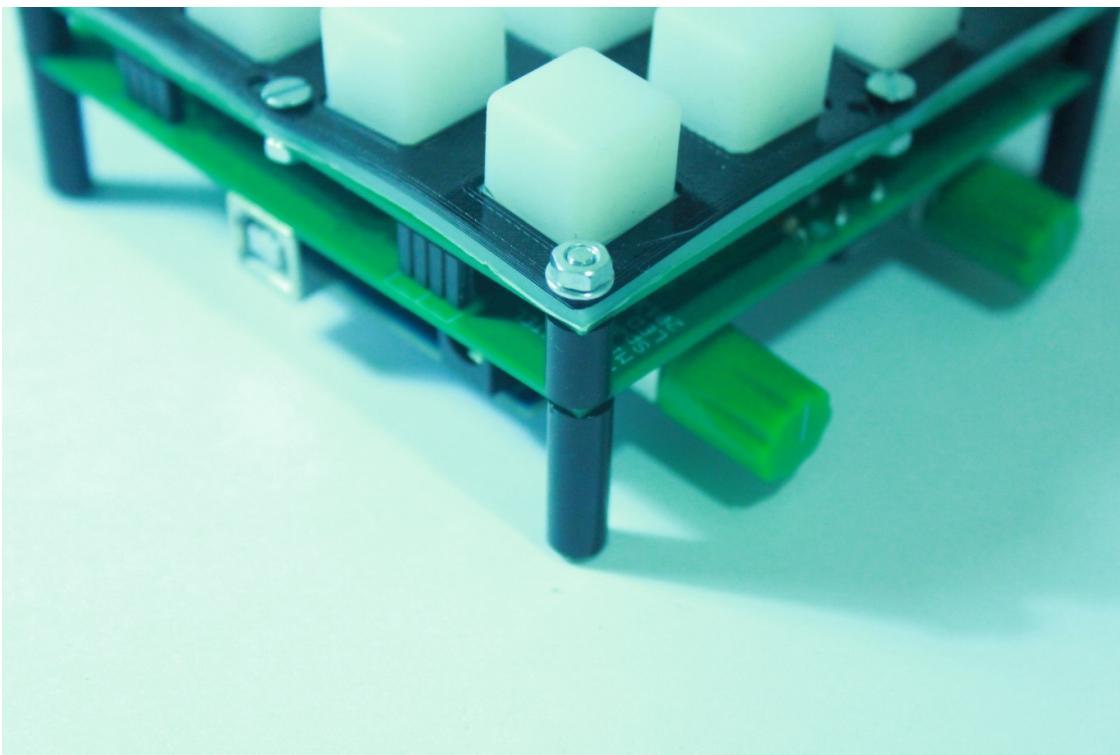


Figura 20. Colocación de patas

3.2. Colocación del embellecedor del potenciómetro

Para colocar el embellecedor se debe ajustar el tornillo una vez colocado en la posición deseada.

Lo ideal es girar el potenciómetro hasta uno de sus extremos y fijar el puntero donde se quiera. Un vez se hayan colocado los embellecedores se ha finalizado todo el proceso de montaje del controlador **MIDIpositivoMIDI V2**.

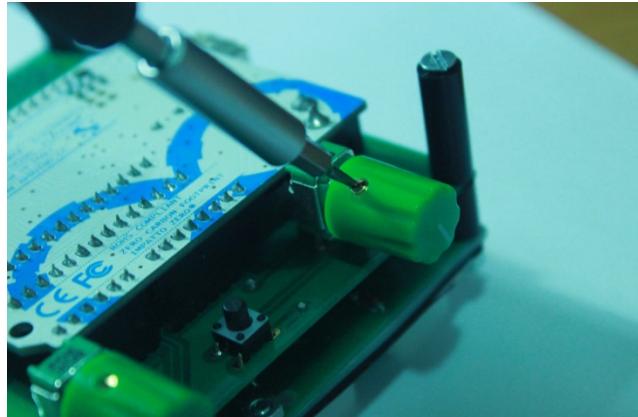


Figura 21. Colocación del embellecedor (knob)



3. Programación en Arduino de MidispositivoMIDI

En este apartado se explicará como realizar la programación de MiDispositivoMIDI. Para ello se ha desarrollado una librería con ejemplos, lo que os facilitará la programación. Las personas interesadas podrán elaborar y modificar su propio código, pero también se proporcionan ejemplos ya realizados, que simplemente hay que cargar al Arduino. Los pasos a seguir serán los siguientes:

- Familiarización con el entorno de Arduino. Como cargar un código.
- Uso de ejemplos proporcionados (librería MiDispositivoMidi v2).
- Flashear el Arduino para que sea reconocido como dispositivo MIDI.
- Hacerlo sonar ☺. Para la gente sin conocimientos en el mundo del audio, se proporciona un programa muy sencillo que permite lanzar sonidos.

En nuestro GitHub podrás encontrar todo lo necesario:

<https://github.com/midispositivomidi>

- Librerías necesarias
- Firmware (original y HIDUINO)
- Programa con sonidos

3.1 Instalar entorno de desarrollo Arduino

MacOS

Desde Google se puede descargar fácilmente si tecleamos “Arduino MAC”, pero vamos a explicarlo brevemente. El primer paso es **descargarse el Sketch de Arduino del siguiente enlace**. Esto es lo que **nos permitirá cargarle el código a nuestro Arduino**.

<http://arduino.cc/download.php?f=/arduino-1.0.6-macosx.zip>

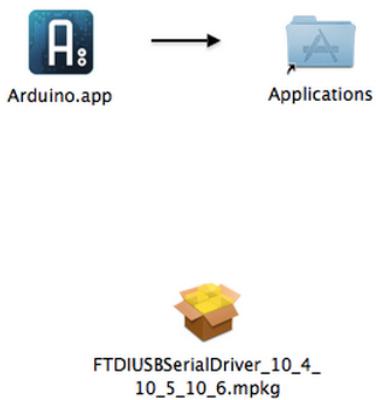


Figura 22. Carpeta Arduino MAC

Arrastra Arduino.app justo a la derecha a la carpeta Aplicaciones, con esto ya quedará copiado y tendréis instalado Arduino. Por otro lado, dependiendo de la versión de Arduino que tengáis podrá ser necesario o no instalar los FTDI Drivers. Si los instaláis, os pedirá reiniciar. Una vez reiniciado ya estáis preparados para pasar al apartado de Ejemplos Librería.

Windows

El entorno de desarrollo de Arduino se puede obtener del siguiente enlace. Este entorno de desarrollo nos permitirá crear y cargar el código que se desee al Arduino.

<http://arduino.cc/download.php?f=/arduino-1.0.6-windows.exe>

Una vez ahí, pinchamos dentro de Windows y se descargará directamente un ZIP llamado Arduino. Para instalarlo primero hay que descomprimir el archivo y ejecutamos el archivo arduino.exe

3.2 Entorno Arduino

Una vez instalado, al abrir el software de Arduino nos sale lo siguiente. Aquí podremos crear todo nuestro código, usar códigos ya realizados, importar librerías,...

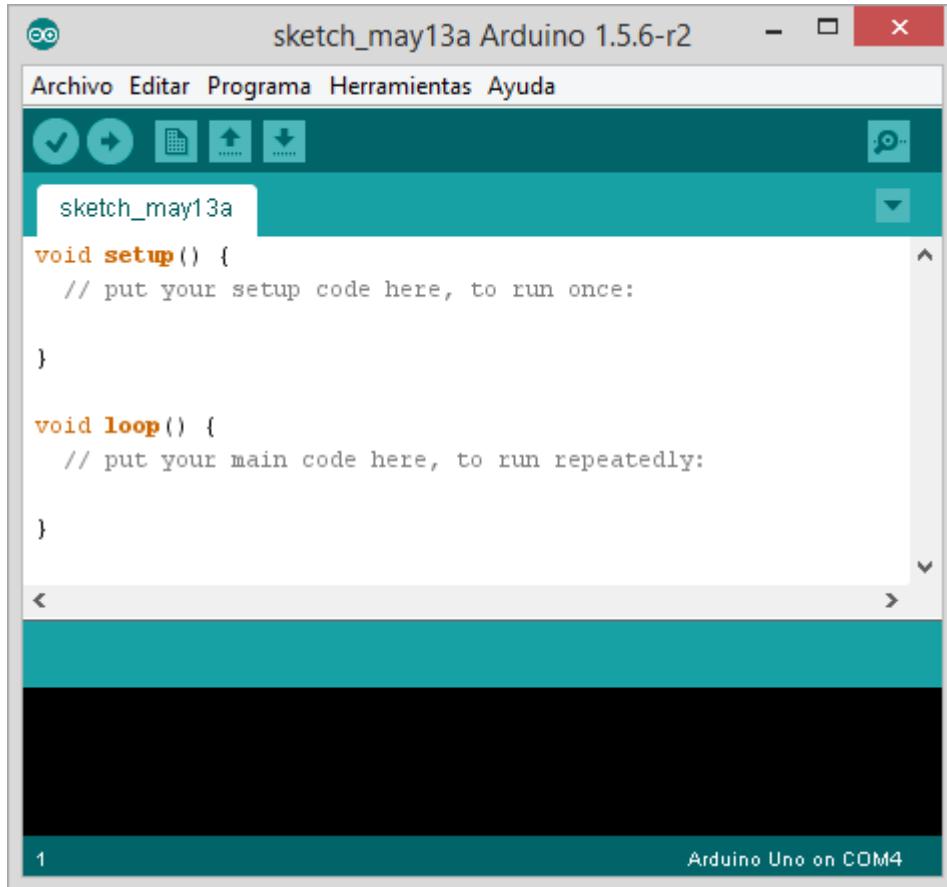


Figura 23. Sketch Arduino Windows

En la barra superior están las herramientas mas importantes, que son “**Verificar**”, que tiene un icono de un tick, y el “**Cargar**” localizado a la derecha del anterior y es el que permite cargar el código al Arduino. A continuación explicaremos que código usar.



Figura 24. Barra Sketch Arduino

3.3 Instalar librerías

Con el fin de facilitar la programación, **se ha realizado una librería con ejemplos** que permite que sea más fácil el uso de MiDispositivoMIDI.

MacOS

Descargarse las librerías del enlace facilitado. Una vez las tengamos, es necesario meterlas en la carpeta de librería de Arduino para que puedan ser usadas. Para meterlas en dicha carpeta, primero hay que descomprimir las librerías. Aquí se pueden ver las librerías que se van a usar, adjuntas en el enlace.

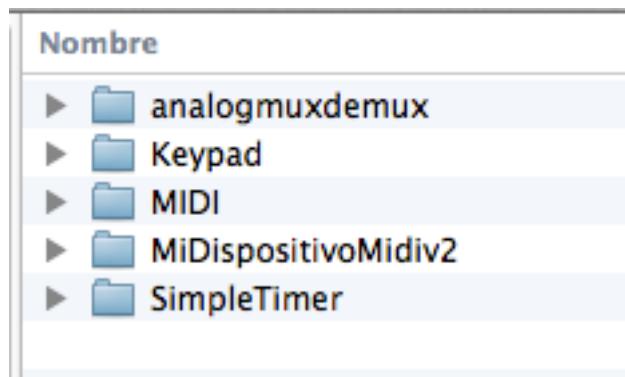


Figura 25. Contenido carpeta

Para que Arduino pueda usarlas, hay que copiarlas en la siguiente carpeta. Aplicaciones → Arduino (Click derecho mostrar contenido del paquete) → Contents → Resources → Java → libraries. Pegamos justo ahí las cinco librerías que se han mostrado.

Una vez realizado esto ya podremos usar la librería de Arduino creada para el curso. Desde el Sketch de Arduino, llendo a Archivo → Ejemplos → MiDispositivoMidiv2.

Windows

Podemos añadir las librerías desde Programa → Importar Librería → Añadir librería. Añadimos **las cinco librerías** que se han facilitado en el enlace. Una vez añadidas, la que interesa es la de MiDispositivoMidiv2. Podemos ver los ejemplos con Archivo → Ejemplos → MiDispositivoMidiv2.

En la siguiente imagen se muestra una aproximación de lo que se ve al importar una librería.

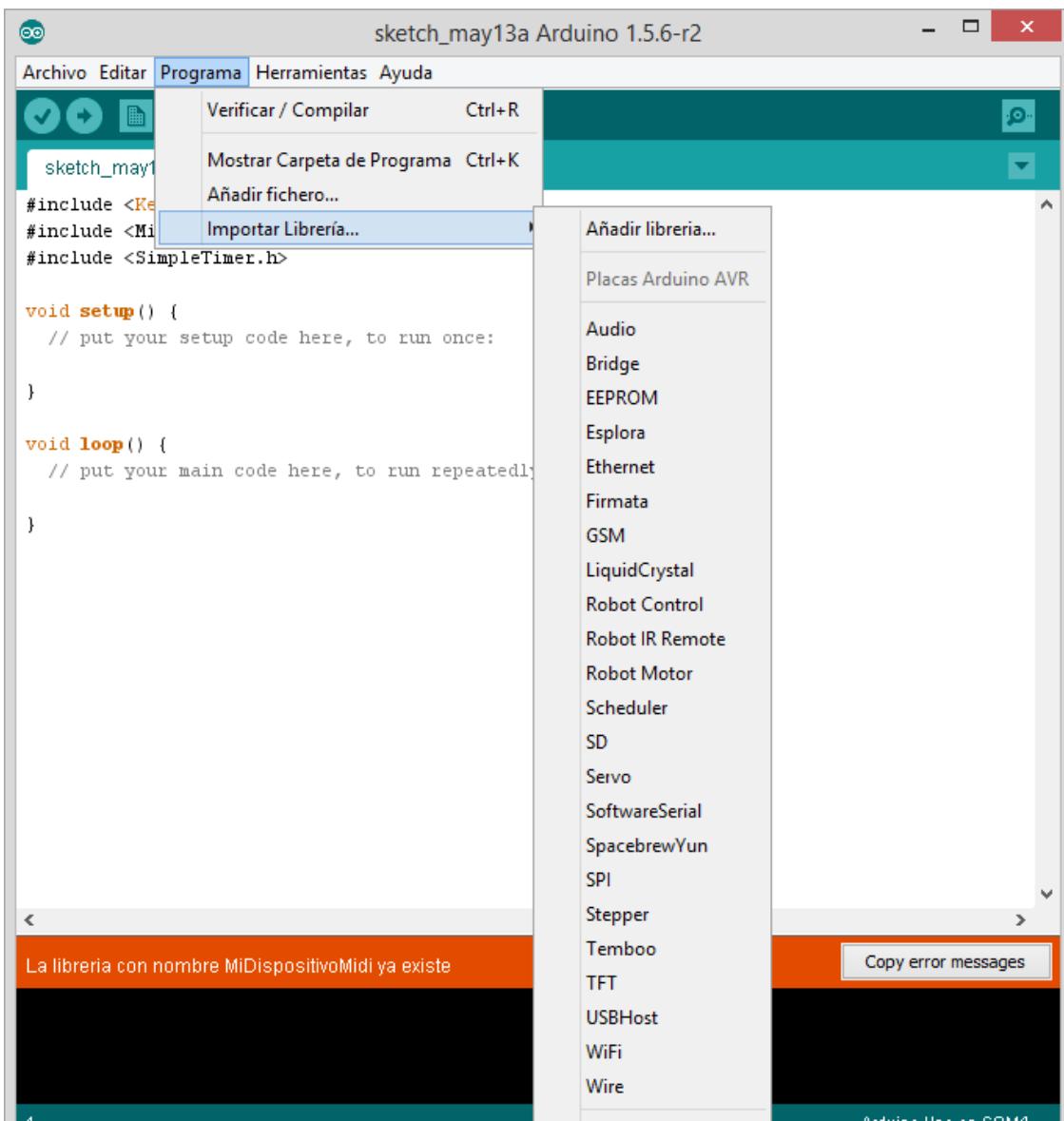


Figura 26. Importando librería en Windows

3.4 Uso de ejemplos

Una vez importadas correctamente las librerías (o pegadas en la carpeta de Arduino), si vamos a Archivo → Ejemplos → midispositivomidiv2, podemos acceder a la librerías de ejemplo que son las siguientes. **Probad a cargar algún ejemplo.** Para ello, clickear en el botón de “Cargar” explicado anteriormente y seleccionar el puerto Serial USB que será algo como **COM1 en Windows o /dev/tty.usbserial-0001 en Mac**. Estos ejemplos sirven para verificar que todo se haya soldado correctamente.

- **1. Led Handling:** Permite familiarizarse con algunas de las funciones desarrolladas para el manejo de los leds rgb. Permite encender elementos de la matriz, columnas, filas o todo del color que se quiera. Es interesante cambiar este código, por ejemplo variando los colores. En este ejemplo aún no se usa MIDI, por lo que no esta pensado para “sonar”.
- **2. Led Array:** Se centra en la función mas importante para el manejo de los leds. Esta función permite iluminar un determinado patrón que se le pasa a la función a través de un array. Este ejemplo tampoco envía MIDI, simplemente es para el manejo de los leds.
- **3. MiDispositivoMIDIv2_easy:** Ejemplo sencillo de envío de MIDI. **Este ejemplo es para ser usado antes de flashear el Arduino (se explica mas adelante).** Básicamente se emplean dos timers para iluminar los leds siguiendo un patrón y se envían diferentes mensajes MIDI de nota activada en función de que botón se pulse. Este ejemplo es para usar con el programa que se explicará mas adelante → **MiDispositivoMidiSampler v2.0**
- **4. MiDispositivoMIDIv2_full:** Se trata de un ejemplo con funcionalidades similares al anterior, pero con una complejidad elevada. Hay una sección de código en la que se puede personalizar que colores se iluminan, cuales están de fondo, que notas se envían y permite la gestión de hasta tres expansiones de potenciómetros. **Este ejemplo esta pensado para flashear al Arduino posteriormente** y usar con algún programa de audio.

3.5 Flashear Arduino

Esta parte es **muy importante**, ya que si no se realizan los pasos adecuados el controlador no funcionará. Para que el Arduino sea reconocido como un dispositivo MIDI comercial, **es necesario reflashearlo**, esto es, cargarle un **nuevo firmware llamado HIDUINO**. Esto permitirá que el ordenador lo reconozca como controlador y pueda ser usado con cualquier programa de edición musical (Nuendo, Traktor, Logic,...). El proceso es el siguiente:

- Cargar el código deseado al Arduino (Los **ejemplos 3 o 4** son los completos). **Ejemplo 3 recomendado para gente sin conocimientos en programas de audio.**
- Reflashear el Arduino con el firmware de HIDUINO.
- Ya esta listo para ser usado, con el programa proporcionado o con programas de audio profesionales.

Es importante decir que **una vez que se reflashee el Arduino, no se podrá volver a cambiar el código**, salvo que se vuelva a flashear con el firmware original (este firmware también se proporciona en el enlace del inicio). No es ningún problema, pero hay que tenerlo en cuenta.

3.5.1 Windows

Existe mucha documentación en internet para la realización del flasheo en Windows, a continuación se detalla el proceso y además se propone una solución para un problema frecuente. Este apartado ha sido extraído del documento “Flasheo con Flip” de Víctor Pastor Rebollo con alguna adaptación.

3.5.1.1 Flashear utilizando Flip

Los usuarios de Windows necesitarán la siguiente herramientas para realizar el flasheo del Arduino:

1. Una aplicación para flashear. En nuestro caso **Flip 3.4.7**
2. Cable USB y Jumper.

El software Flip lo podemos obtener directamente de la página de Atmel (<http://www.atmel.com/tools/FLIP.aspx>). Dentro de la pagina tenemos disponibles varios enlaces de descarga . Mi recomendación es que descarguéis el que tiene Java incluido y vayáis instalando todo lo que os indica hasta finalmente tener el acceso directo a Flip en el escritorio.

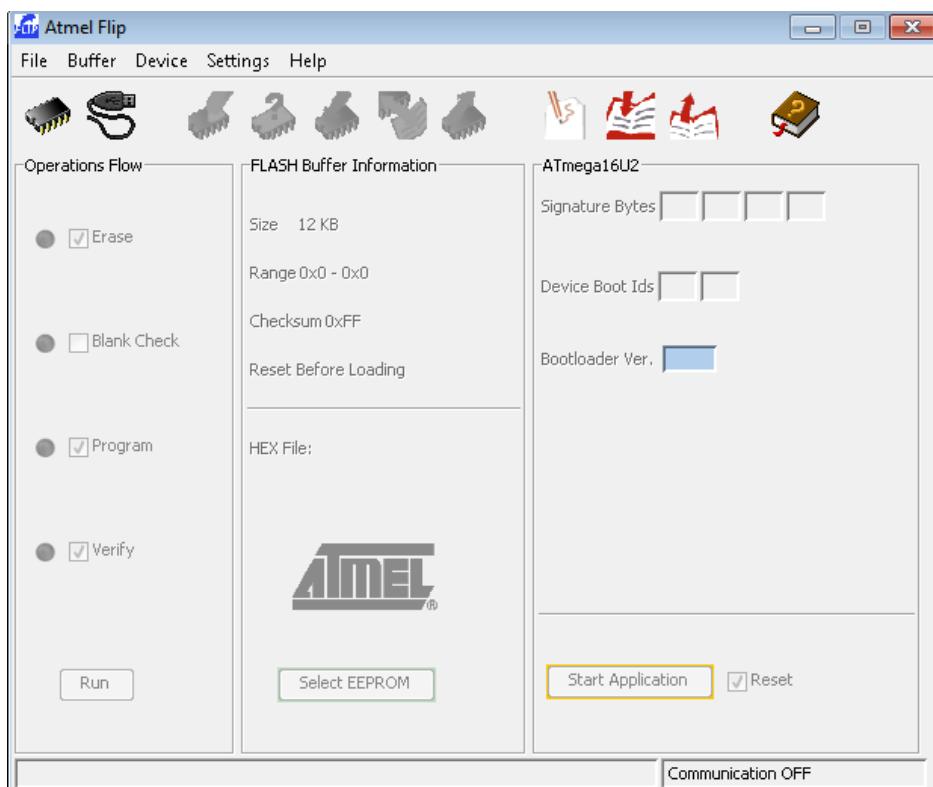


Figura 27. Flasheando en Windows. Flip

Se puede ver en la figura superior el interfaz de Flip. Lo que primero necesitaremos para flashear al Arduino será entrar en modo DFU (Device Firmware Update). Para poner en el modo DFU nuestra placa Arduino lo que tenemos que hacer es cortocircuitar los dos pines que podemos ver en la figura inferior. Se puede hacer usando un jumper, o con cualquier elemento que cortocircuite ambas patillas.

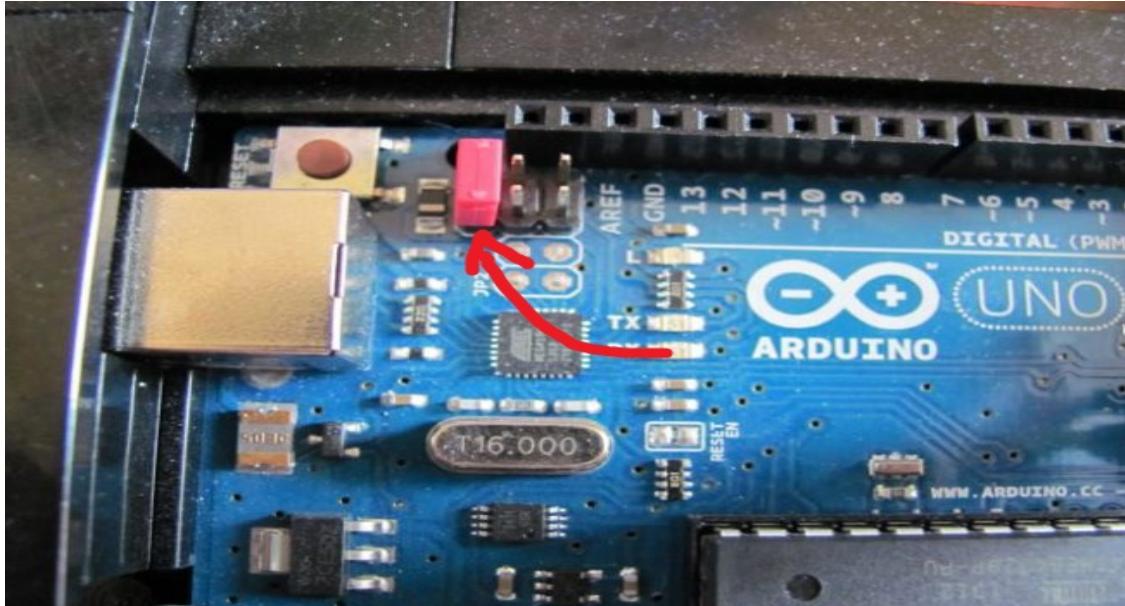


Figura 28. Conexionado del jumper para flashear

Es importante que una vez realizado el cortocircuito, hay que quitar el jumper.

Antes de llegar a este punto, se da por supuesto que ya se ha cargado el código deseado a vuestro Arduino. Se recuerda que una vez cambiado el Firmware no se podrá cargar código (salvo que se vuelva a flashear con el firmware original).

Para continuar con el flasheo, abrimos nuestro programa Flip. Y clickeamos en el siguiente botón (tiene forma de circuito integrado).

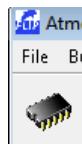


Figura 29. Icono para flasheo

Ahora es necesario seleccionar “ATmega16u2”. Este nombre podrá depender de vuestra versión de Arduino.



Figura 30. Selección de dispositivo para flasheo

Ahora es necesario clickear en el icono con forma de USB, localizado a la derecha del anterior, y seleccionar USB. A algunos usuarios les ha salido el siguiente error, por lo que se explica como resolverlo. Si no te ha salido ningún error, puedes pasar al **apartado 3.5.1.4**.

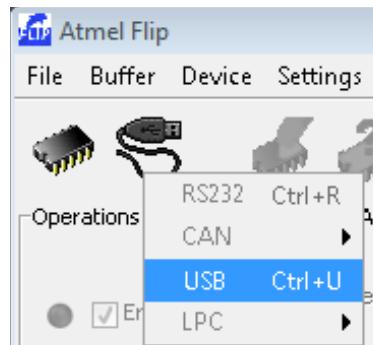


Figura 31. Selección de puerto usb

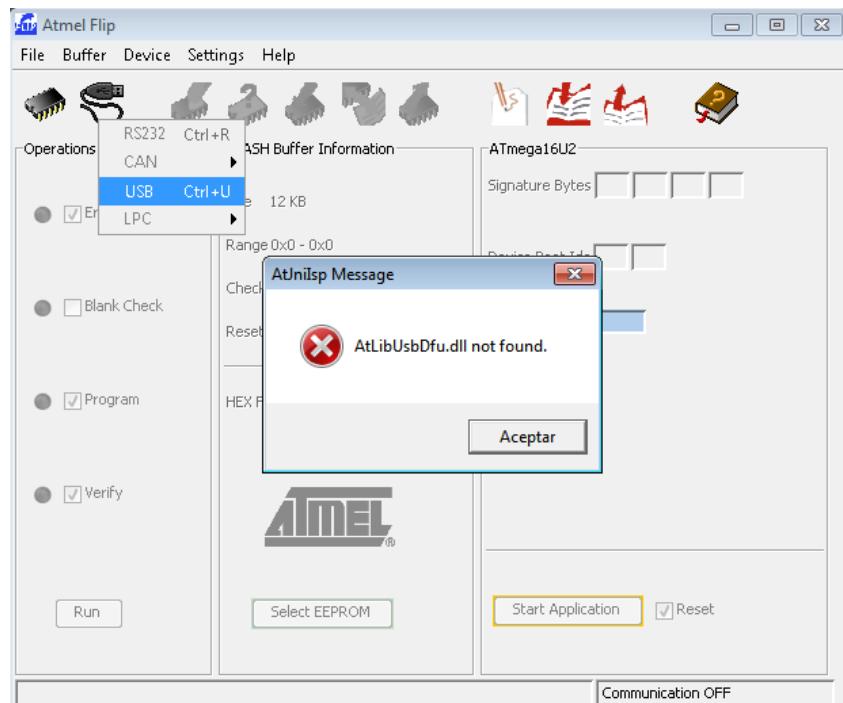


Figura 32. Posible error

La causa es que a partir de la versión 3.4.2 no utiliza los drivers llamados *libusb0.sys* , sino que utiliza otros (*libusb-win32 Windows driver*) , y de alguna manera tampoco reconoce el driver *AtLibUsbDfu.dll*. Aún así, se puede ver que dentro del la carpeta donde se ha instalado el flip , dicho driver (*AtLibUsbDfu.dll*) si que está presente. En la siguiente imagen se puede ver el driver y la ruta.

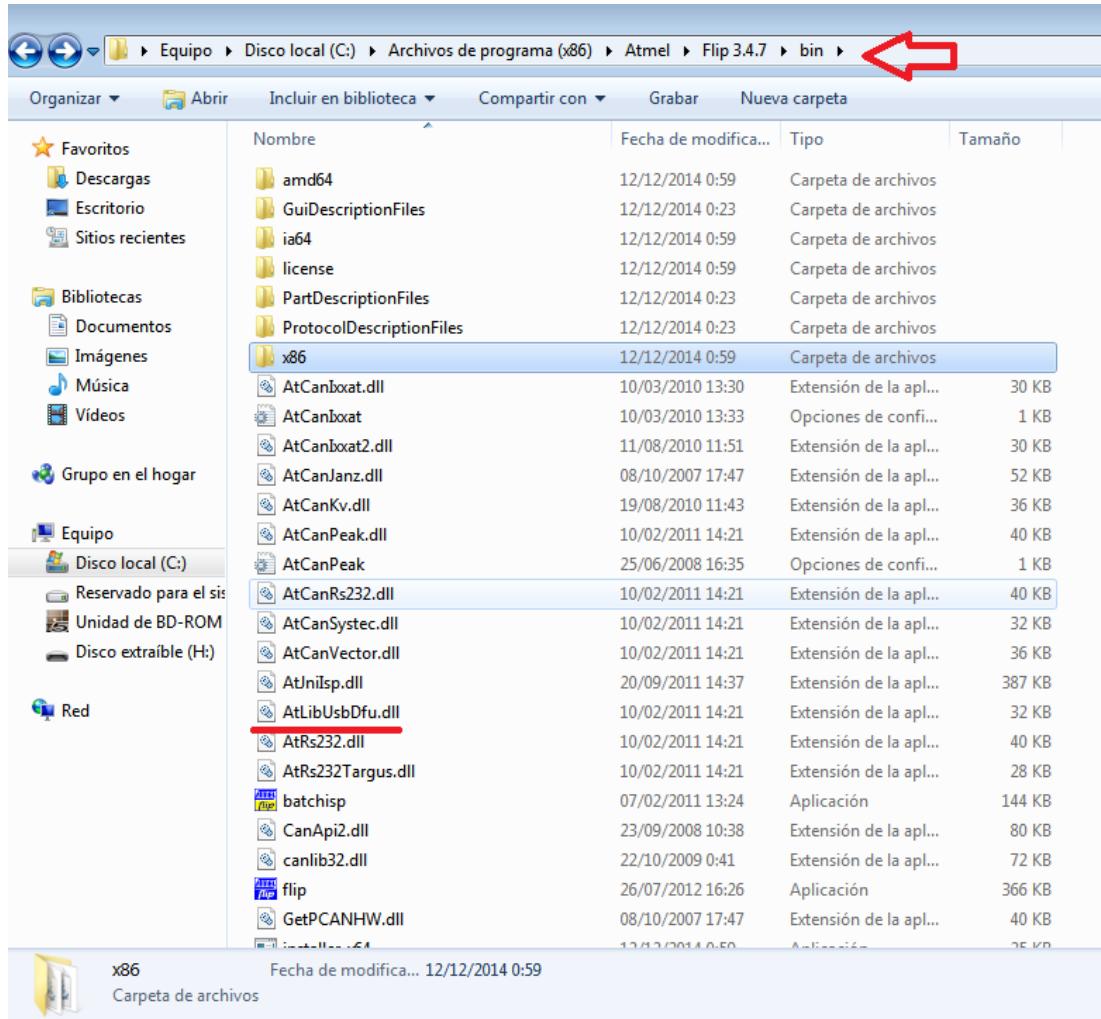


Figura 33. Soluciando error

Resumiendo, hay que instalar los drivers *libusb0.sys*. Para ello existen dos formas que se explican en los **apartados 3.5.1.2 y 3.5.1.3**.

Perfecto, ahora empieza lo bueno , para instalar los drivers que acabo de mencionar existen dos métodos . Insisto en hacer los dos ya que hacer solo uno no garantiza que Flip vaya a funcionar bien

3.5.1.2 Solucionando problema. Forma 1, descargar los drivers de la página

Esta es la manera más rápida de obtener los drivers necesarios para instalarlos en el equipo. Los podemos descargar directamente de esta página. Una vez dentro , seleccionamos la ultima versión que podemos descargar.

<http://sourceforge.net/projects/libusb-win32/files/libusb-win32-releases/>

Name	Modified	Size	Downloads / Week
Parent folder			
1.2.6.0 	2012-01-17	3.539	
1.2.5.0	2011-07-24	133	
1.2.4.0	2011-04-12	83	
1.2.3.0	2011-03-17	25	
1.2.2.0	2010-10-02	55	

Figura 34. Solucionando error

Al hacer click nos lleva al contenido de la carpeta , ahí lo que tenemos que descargar el archivo ZIP [libusb-win32-bin-1.2.6.0.zip](#). (Para guiarnos mejor , es aconsejable ver cuáles son los archivos que mayores descargas tienen, como se puede ver en la parte derecha de la imagen superior).

Una vez que hemos descargado el archivo Zip, lo extraemos donde queramos. Nos metemos en la carpeta donde hemos descomprimido el archivo nos metemos en la carpeta Bin , y ahí debemos **ejecutar como administrador** la aplicación [Inf-Wizard](#) .

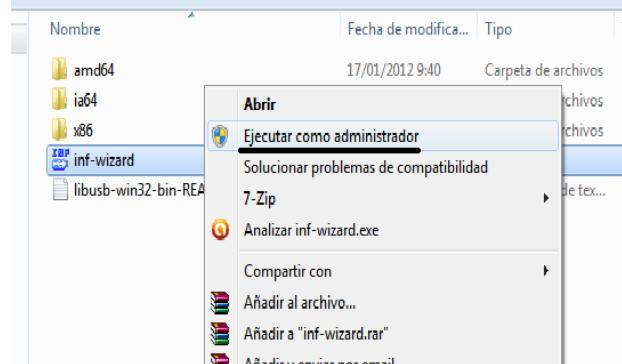


Figura 35. Solucionando error

Una vez que lanzamos la aplicación , nos pide que seleccionemos el dispositivo en el que queremos que se utilicen los drivers que vamos a instalar . Es muy importante , como se indicó anteriormente que ***la placa Arduino se encuentre en el modo dfu***.

En la ventana siguiente , podemos dejar lo que viene por defecto escrito, no va a repercutir en ningún aspecto

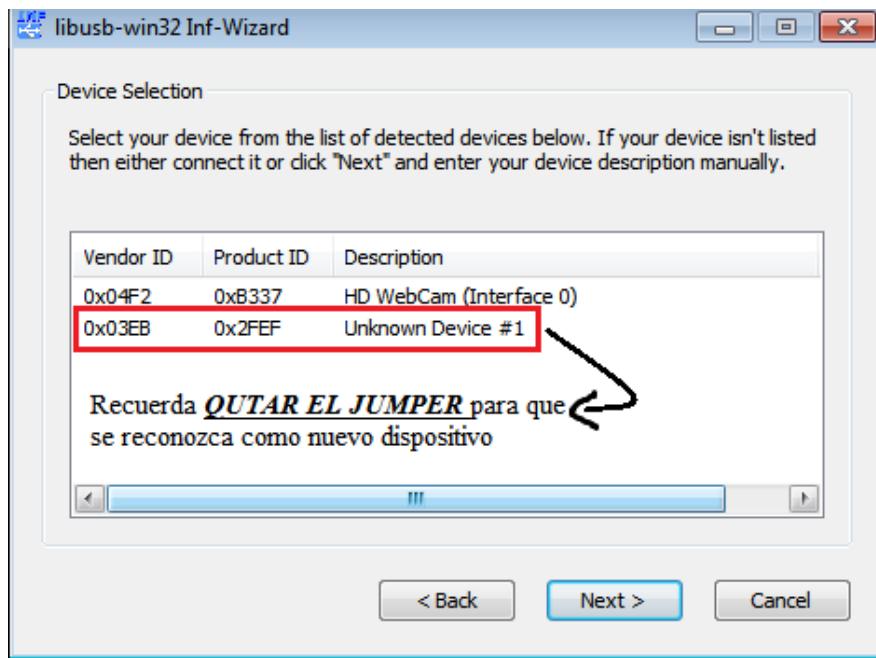


Figura 36. Solucionando error

A continuación , se va a indicar dónde queremos que se guarden los drivers. Para ello buscaremos la carpeta donde hemos instalado Flip , dentro de la carpeta Bin. **Es importante haber ejecutado como administrador.**

3.5.1.3 Solucionando problema. Forma 2 descargar programa Zadig.

En este método tenemos que descargarnos el programa Zadig.exe , que lo podemos encontrar en el siguiente enlace :

<http://code.google.com/p/libusb-winusb-wip/downloads/detail?name=zadig.exe>

Este enlace nos redirecciona a una página , simplemente tenemos que hacer click en el nombre del programa y se nos descargara un archivo que podemos guardar donde queramos.



Figura 37. Solucionando error

Una vez que tenemos descargado el programa lo ejecutamos. Seleccionamos la pestaña options y abrimos *List All Devices* y *Advanced Mode*. (**Recordad tener siempre la placa de Arduino en modo DFU y con el Jumper quitado**).

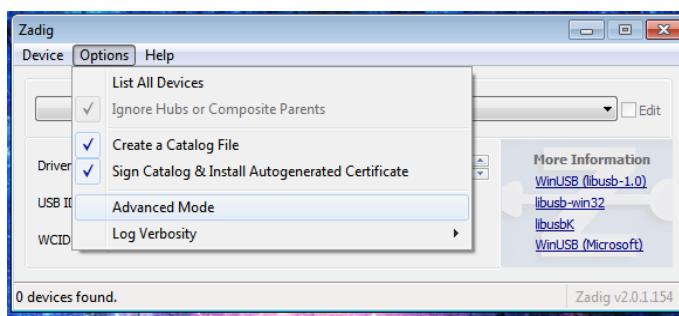


Figura 38. Solucionando error

Una vez que hemos activado estos dos elementos, hacemos click en la barra que hay justo debajo y seleccionamos nuestro dispositivo. Después, en el cuadrado que hay al lado del la flecha verde ,que tendrá seleccionado por defecto WinUSB... debemos cambiarlo a libusb-win32 (v1.2.5.0), y si hacemos click en icono de la carpeta , podemos elegir donde guardamos los drivers que vamos a

instalar.

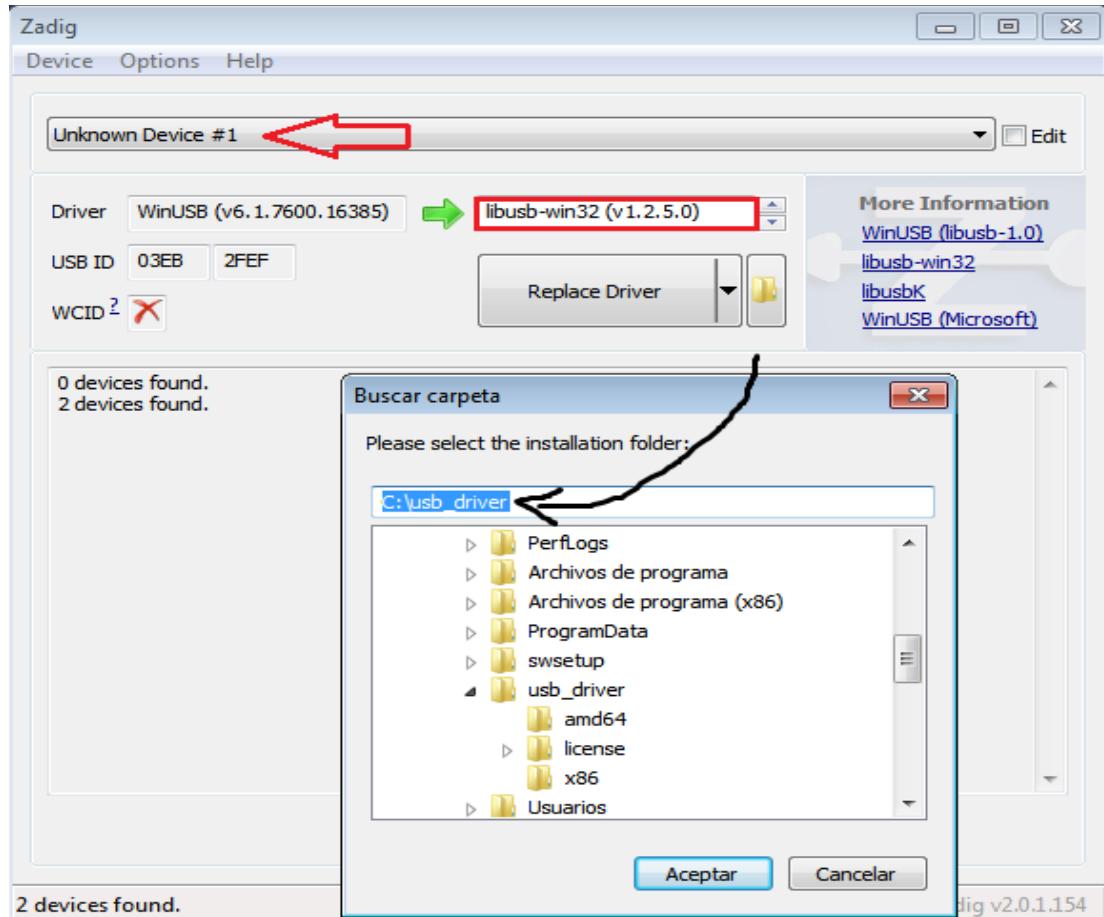


Figura 39. Solucionando error

Una vez que ya lo tenemos todo, le damos click en *Replace Driver*, esperamos un momento y ya estará todo solucionado.

3.5.1.4 Terminar de flashear

Si has llegado a este punto es que , o todo ha ido bien a la primera , o te han surgido alguno de los errores que hemos comentado arriba y los hemos solucionado.

El procedimiento es el mismo que hemos hecho al principio de este documento :

1. Abrimos Flip
2. Pulsamos en el icono del CI y seleccionamos ATmega16u2
3. Hacemos Click en el icono del USB.

A continuación:

4. Le damos a open (vemos como en la placa de Arduino se iluminan dos leds naranjas y que en el programa ahora en la parte de abajo USB OFF ha cambiado a USB ON)
5. Hacemos click en el icono del libro que tiene una flecha señalándole
6. Cargamos nuestro archivo con extensión .HEX (el firmware en cuestión)
7. En la zona que pone *Operations Flow* le damos a run .

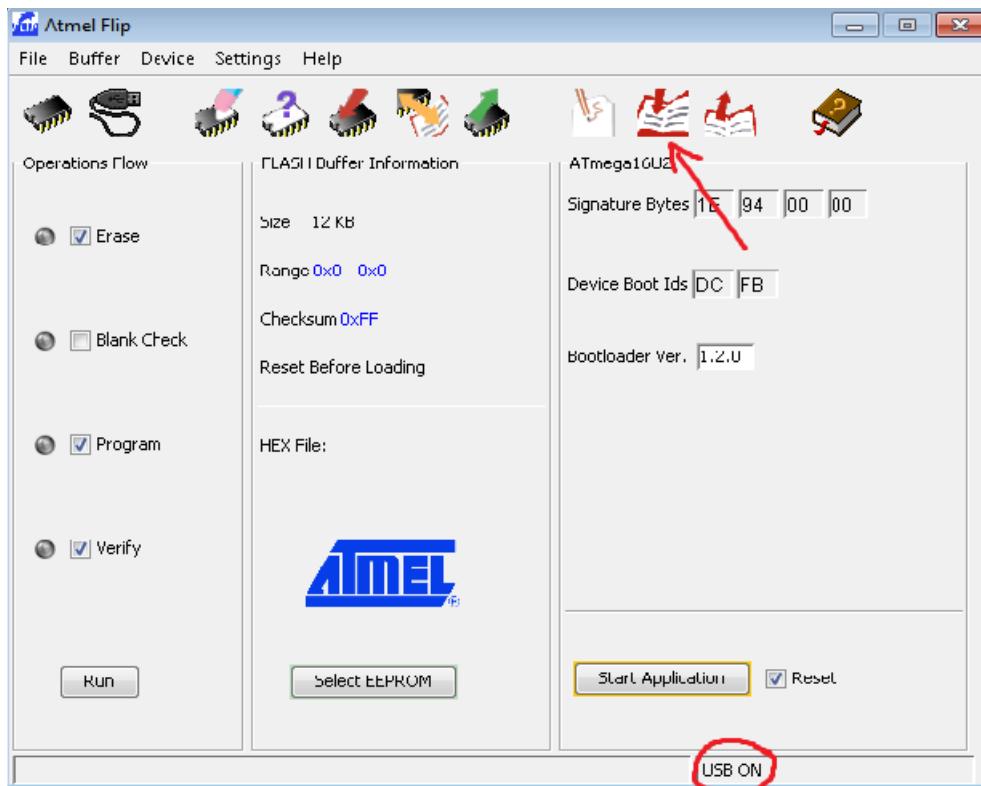


Figura 40. Finalizando flasheo

3.5.1.5 Anexo. No reconoce puerto COM.

En el caso de que no os reconozca el puerto COM como se ve en la siguiente imagen. Uno de los problemas más comunes es que no tengáis los drivers necesarios instalados. Explicamos como instalarlos.

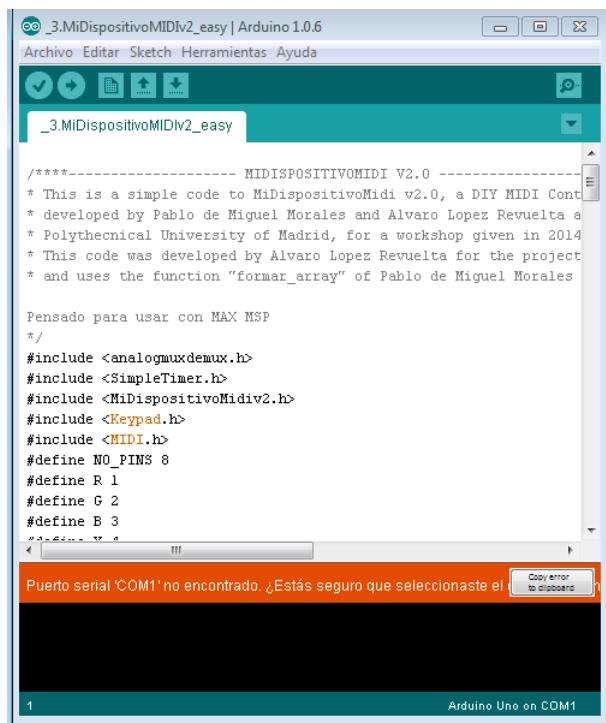


Figura 41. Problema puerto COM Windows

Lo que ocurre es que no tenemos instalados los drivers que hacen que reconozca el puerto serial al que tenemos conectado la placa. Pero la solución es sencilla, simplemente tenemos que dirigirnos a panel de control → Hardware y Sonido → Administrador de Dispositivos.

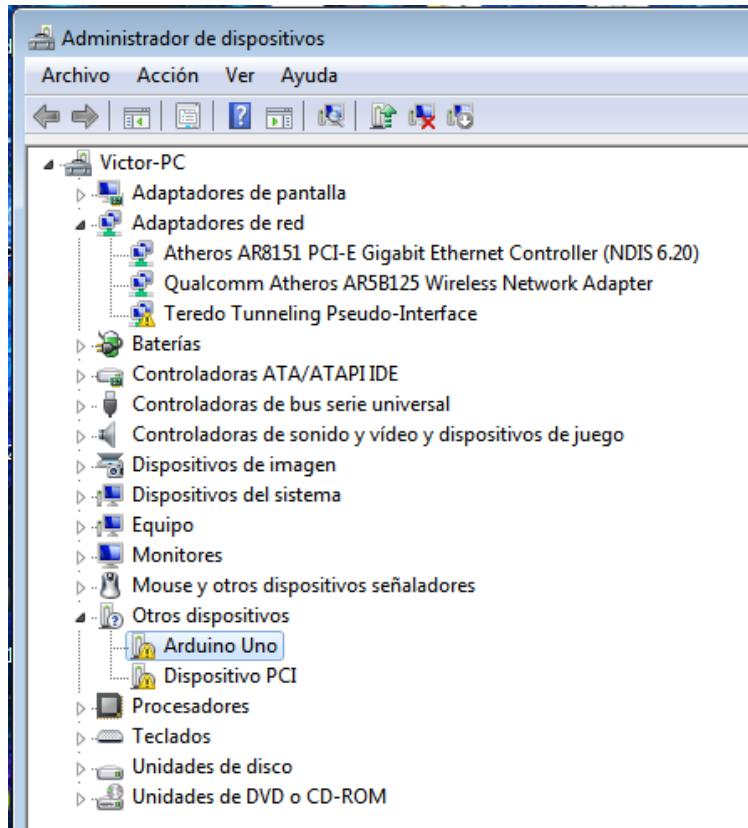


Figura 42. Instalando dirver windows

Lo único que tenemos que hacer es click derecho en el dispositivo de Arduino → Actualizar Software del Controlador → Buscar Software de controlador en el equipo. Coloca la ruta de la carpeta drivers ubicada dentro de la carpeta arduino-1.0.6 , hacemos click en siguiente y listo , ya tenemos nuestro driver instalado .

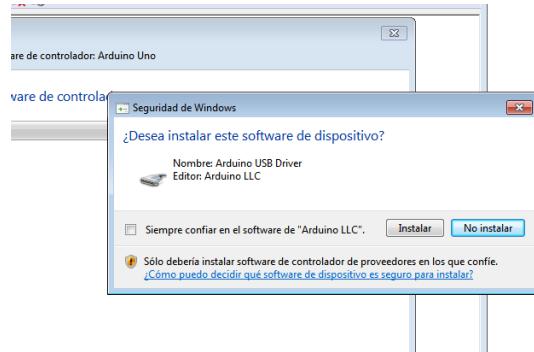


Figura 43. Instalando drivers

Ahora, nos vamos a nuestro programa de Arduino y en herramientas, puerto serial , ya somos capaces de seleccionar el puerto donde está conectado la placa. Todo solucionado.

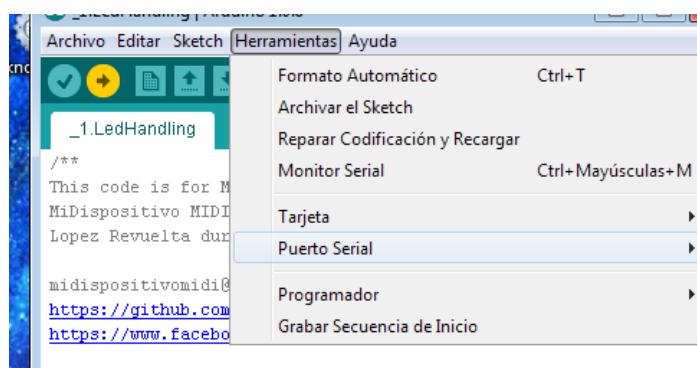


Figura 44. Verificando el funcionamiento

3.5.2 Mac y Linux

Lo primero que deberemos hacer es **descargarnos el firmware de Hiduino**. Este firmware se ha facilitado en la carpeta enlazada en el apartado 3, pero puede ser descargado de aquí.

<https://github.com/ddiakopoulos/hiduino>

Ahora nos tenemos que descargar el “**dfu-programmer**”, que existe para Linux y para MacOS.

En Linux:

```
sudo apt-get install dfu-programmer
```

En MAC: (hay que tener instalado el MacPorts)

```
sudo port install dfu-programmer
```

A continuación necesitaremos que nuestro Arduino **entre en modo DFU** para que se nos permita introducirle el firmware. **Antes de nada es muy importante que hayáis cargado el código que deseáis de los ejemplos.** Una vez que introducimos este nuevo firmware, no podemos volver a meter código en el Arduino, pero ¡tranquilos! que si volvemos a meter el firmware original de Arduino todo volverá a la normalidad (Se hace de manera idéntica a introducir cualquier firmware). Como he dicho hay que entrar en modo DFU con nuestro Arduino, para ello:

Con el Arduino conectado al ordenador, conecta un **jumper entre los terminales** que se ven en la figura. Se verá como parpadea un LED. Ya se está en modo DFU.

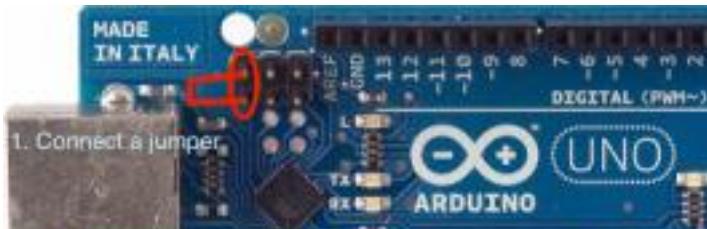


Figura 45. Conexionado jumper

En mi caso, dejé el firmware de Arduino (llamado **HIDUINO_MIDI.hex**) en el escritorio. A continuación se explica como cargar ese firmware al Arduino. (Linux y MAC igualmente válido). El **“atmega16u2”** puede variar dependiendo del modelo de Arduino, pero para la v2.0 de MiDispositivoMIDI habrá que acerlo como se indica:

```
cd \Desktop
sudo dfu-programmer atmega16u2 erase
sudo dfu-programmer atmega16u2 flash HIDUINO_MIDI.hex
sudo dfu-programmer atmega16u2 reset
```

Desconecta y vuelve a conectar el cable del Arduino. Ahora tu Arduino ya se comporta como si fuera un controlador MIDI comercial. Conéctalo al programa de audio que desees (Ableton, Cubase, Nuendo, Sonar, Traktor,...), asígnale los sonidos que desees y ¡A tocar!. Si no sabes usar estos programas, nunca es mal momento para iniciarse. Son bien sencillos y muy divertidos.

```
MacBook-Pro-de-AlvaroLR:Desktop alvarolopezrevuelta$ sudo dfu-programmer atmega16u2 erase
Password:
MacBook-Pro-de-AlvaroLR:Desktop alvarolopezrevuelta$ sudo dfu-programmer atmega16u2 flash HIDUINO_MIDI.hex
Validating...
3320 bytes used (27.02%)
MacBook-Pro-de-AlvaroLR:Desktop alvarolopezrevuelta$ sudo dfu-programmer atmega16u2 reset
MacBook-Pro-de-AlvaroLR:Desktop alvarolopezrevuelta$
```

Figura 46. Terminal, comandos para flasheo

Ya que los pasos anteriores para el flasheo de Arduino puede resultar complicados, en las sesiones presenciales del curso se flaseará el Arduino a quien lo necesite, con el fin de ahorrar tiempo. Aún así, resulta interesante saber hacerlo y entenderlo.

En el caso de que estemos usando un software como Ableton, podemos ver que ya nos lo reconoce sin la necesidad de tener instalado ningún programa.

MIDI Ports	Pista	Sincro	Remote
▷ Input: Driver IAC (Bus 1)	No	No	No
▷ Input: HIDUINO	Sí	No	No
▷ Output: Driver IAC (Bus 1)	No	No	No
▷ Output: HIDUINO	No	No	No

Figura 47. Configuración midi en Ableton Live

3.6 Hazlo sonar. MiDispositivoMidi Sampler v2.0

Los usuarios que sepan manejar programas de audio podrán emplear MiDispositivoMIDI con cualquier programa que deseen. Para los que no tengan ningún programa de audio (DAW), **se proporciona un sencillo programa llamado Sampler**, que permite hacer sonar a tu dispositivo. Para ello:

- Carga el [Ejemplo 3: MiDispositivoMIDIv2_easy](#)
- Flashea el Arduino con firmware HIDUINO_MIDI.hex
- Descárgate el programa (versión para Mac y Win) de MiDispositivoMIDI Sampler v2.0 (viene en el enlace proporcionado).
- A jugar

Para usar el software mencionado simplemente es necesario abrirlo, hacer doble click en “notein” para seleccionar HIDUINO (que es por donde queremos recibir el MIDI) y **clickear en “trigger”** que hará que se carguen unos determinados samples.

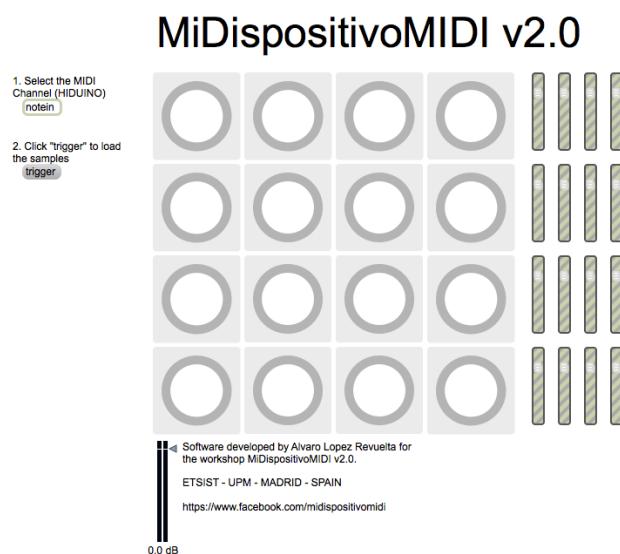


Figura 48. Programa Sampler

*Actualizado. Se ha desarrollado una versión ***MiDispositivoMidiCustomSamplerv2*** para Mac y Win que **permite al usuario cargar sus propios samples** en cada botón. Para eso y mucho más:

<https://github.com/midispositivomidi>

4. Anexos

4.1 ¿ Cómo soldar?

En este enlace podéis encontrar una guía muy completa y bastante buena de cómo se debe soldar. Además, esta página contiene muchos proyectos para que pluriempleis vuestros *Arduinos* una vez sepáis como manejarlos.

<http://www.instructables.com/id/How-To-Soldering/>

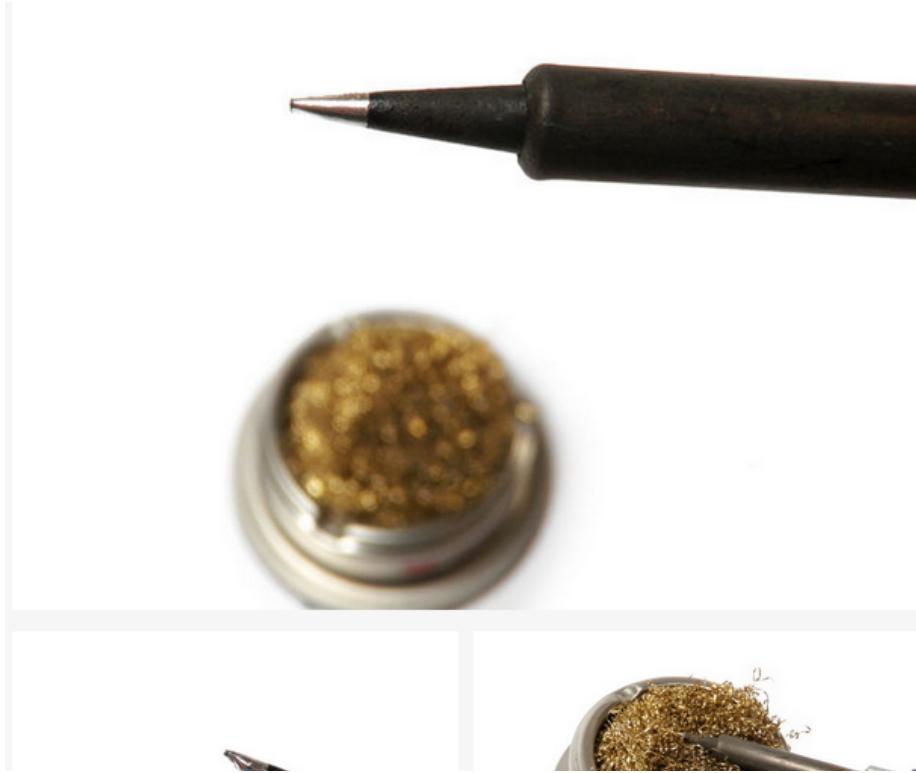


Figura 49. Para soldar (de instructables)

Es importante tomar las debidas precauciones a la hora de soldar. Además de soldar en un lugar amplio, bien iluminado y ventilado, se recomienda protegerse con una mascarilla y unas gafas que protejan de posibles salpicaduras o el vapor de flux.

4.2 ¿Qué es MIDI?

A modo de resumen, MIDI es un protocolo de comunicación muy usado en el mundo de la música, ya que permite interconectar todo tipo de controladores entre sí. Debido a que es un protocolo de comunicación, MIDI no suena por si mismo, por lo que cuando se habla de controladores MIDI, son simplemente elementos que envían información de nota, intensidad y muchos otros parámetros a un dispositivo que reconoce esos mensajes y los asocia a determinados sonidos.

Algo general con ejemplos → <http://alvarorevuelta.net/midi-and-how-does-it-work/>

Interesante → <http://www.hispasonic.com/reportajes/protocolo-midi/13>

Tablas de mensaje MIDI. Bastante técnico pero importante → <http://www.midi.org/techspecs/midimessages.php>

4.3. Esquemático MiDispositivoMIDI

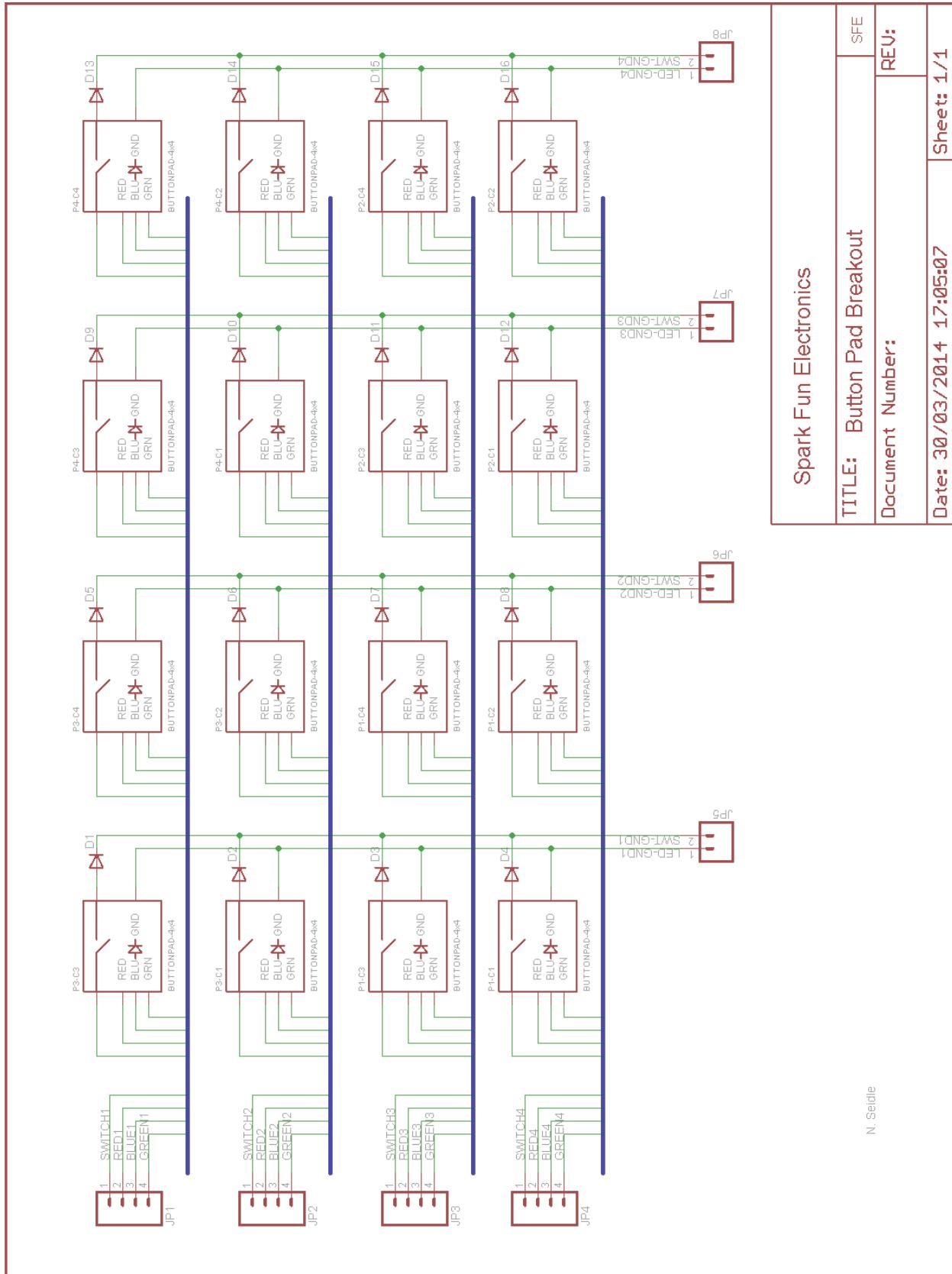


Figura 50. Esquemático placa superior Sparkfun

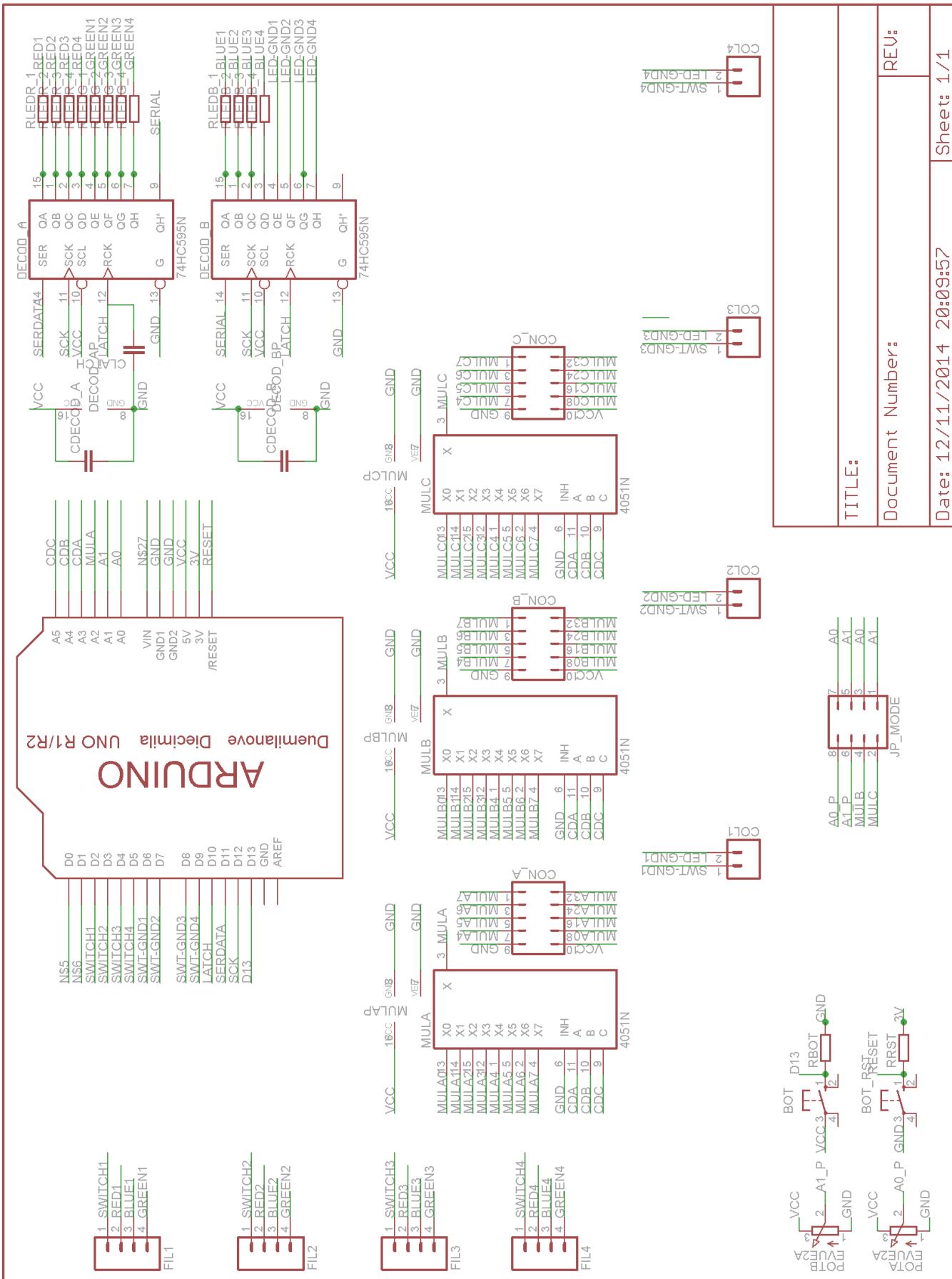


Figura 51. Esquemático placa inferior MiDispositivoMIDI

5. Referencias

- [1] LÓPEZ REVUELTA, Álvaro. “*MIDI and how does it work*”. Spain, August 17th 2014 [Consulted: December 5th 2014]. <http://alvarorevuelta.net/midi-and-how-does-it-work/>
- [2] LÓPEZ REVUELTA, Álvaro. “*Arduino + MIDI: Tu propio piano*”. Crea tu piano controlador y haz que suene. Spain, August 29th 2014 [Consulted: December 5th 2014]. <http://geekytheory.com/arduinomidi-tu-propio-piano/>
- [3] RANDY. “*How to Soldering*”, San Francisco Mars 7th 2006. [Consulted: December 5th 2014]. <http://www.instructables.com/id/How-To-Soldering/>
- [4] HOOD, Nico. “Tutorials & Documentation about evil and nerdy shiy” *Install new Firmwares with DFU and Flipo n Arduino UNO/Mega R3*. June 3rd 2014. [Consulted December 10th 2014]. <http://nicohood.wordpress.com/2014/06/03/install-new-firmwares-with-dfu-and-flip-on-arduino-unomega-r3/>
- [5] Arduino Website. “*Updating the Atmega8U2 and 16U2 on an Uno or Mega2560 using DFU*”. [Consulted: 12th December 2014]. <http://arduino.cc/en/Hacking/DFUProgramming8U2>
- [6] PCB HEAVEN, “*How a Key Matrix Works*” June 27th 2010. [Consulted: 12th December 2014]. http://pcbheaven.com/wikipages/How_Key_Matrices_Works/
- [8] PASTOR REBOLLO, Victor. “*Flasheando Arduino con Flip*”. December 2014