

# Hiduino - Proceso para convertir un Arduino en un instrumento musical Midi

## [Repo oficial](#)

El proceso creativo para desarrollar este ejercicio se lleva a cabo en varios momentos, primero se diseña y se realiza la parte de Hardware, luego de software y por ultimo las respectivas programaciones y composiciones musicales o para lo que se este desarrollando.

En este caso automatizaremos un xilofono para que actue como un instrumento musical esclavo.

1. Elaboracion y diseño del instrumento Robotico.
2. Programación y composición en el Tracker de la Gameboy
3. Programación y composición en el Tracker de la Gameboy.
4. Sincronización midi Gameboy como MidiOut y controlador USB (Hiduino) como esclavo.
5. Programación y composición en el computador con la música construida en la Gameboy.
6. Composición de la música final añadiendo las partes y melodías del instrumento robótico.

## **Elaboracion y diseño del instrumento robotico**

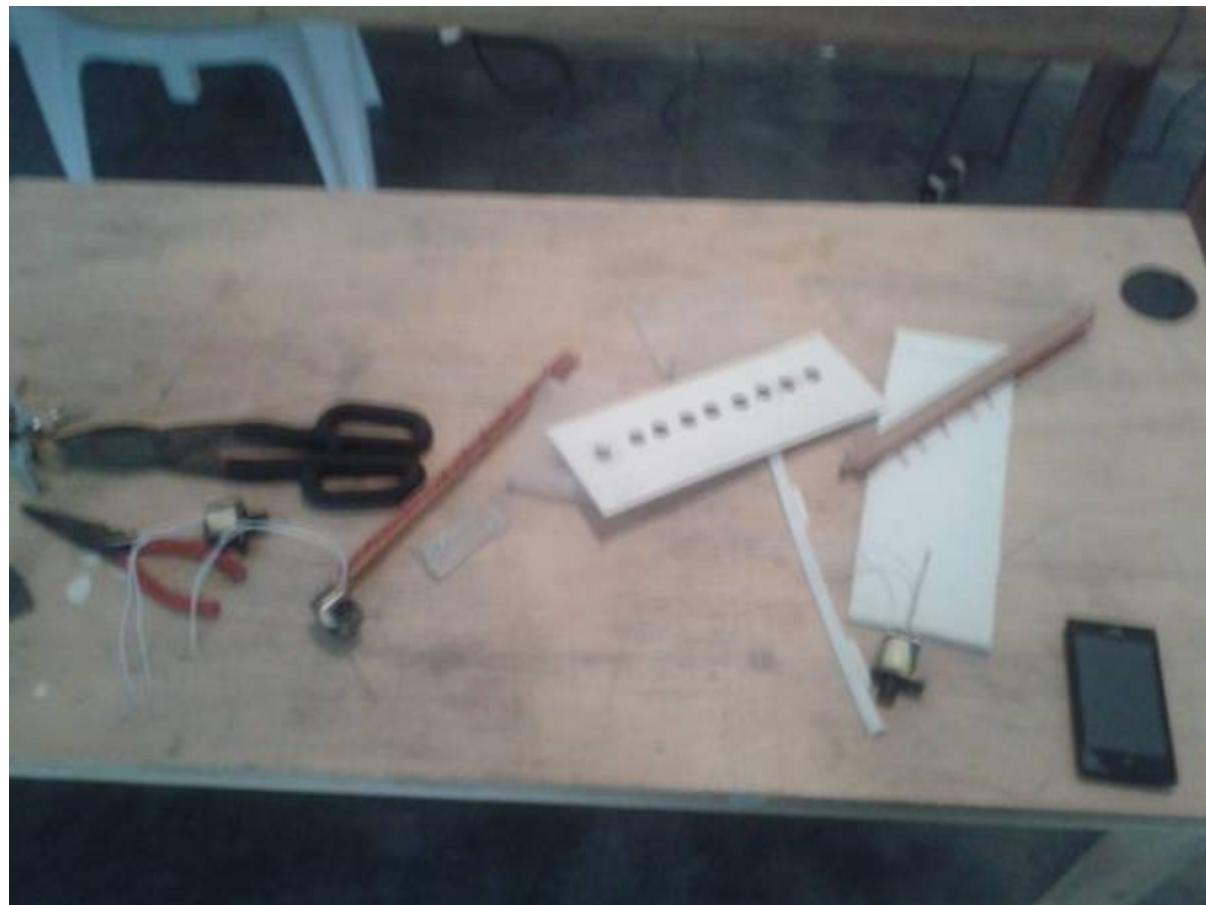
El desarrollo del instrumento robótico se lleva a cabo en diferentes partes, tanto hardware como software:

### **Construcción del instrumento**

El instrumento musical es un rediseño de un viejo xilófono, adaptado para ejecutar un golpe con un voltaje definido y realizado por los solenoides y que además no distorsione el sonido del instrumento.





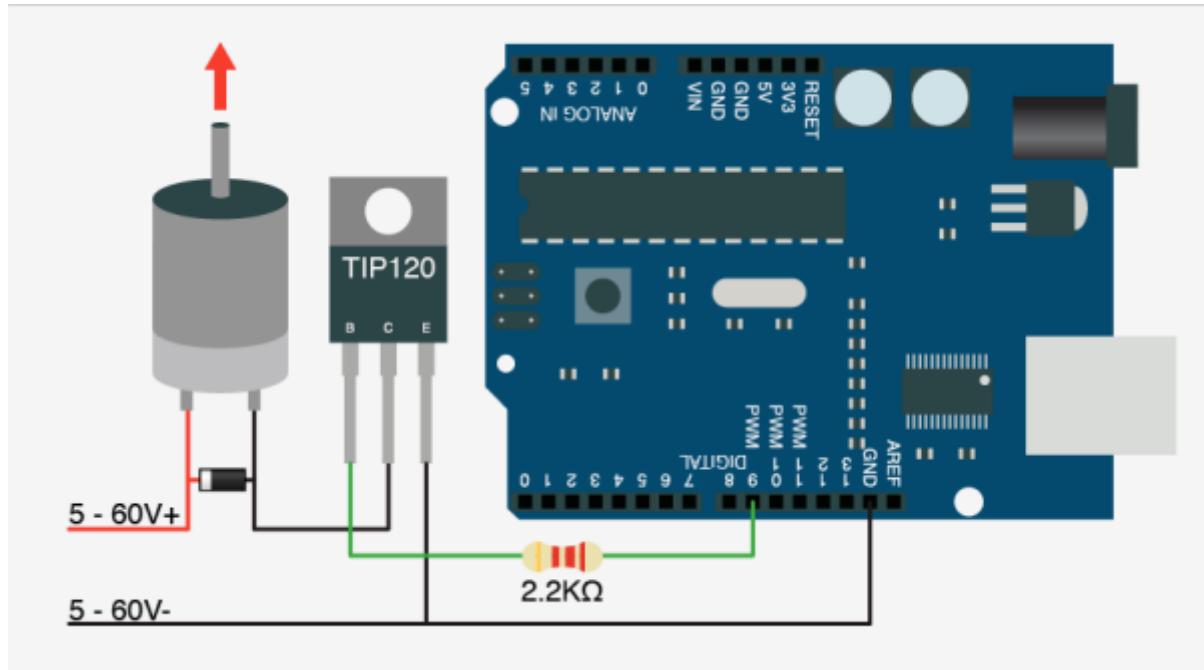




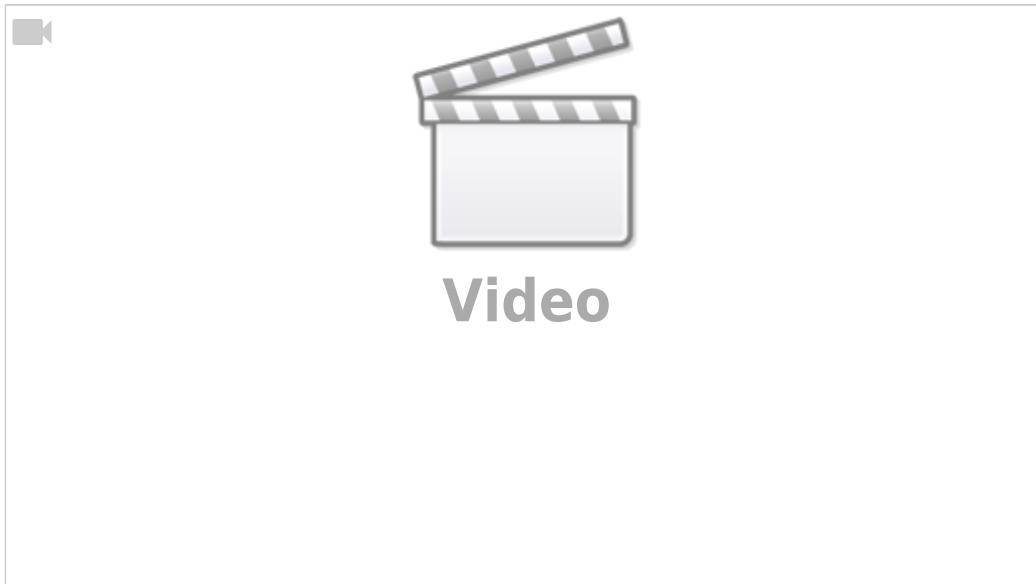
Para llevar a cabo este proceso se experimenta con varios modelos, alturas y voltajes que llegan a los actuadores hasta encontrar la posición ideal. Posteriormente se construye un circuito electrónico que permite accionar cada percutor y este a su vez; es controlado eléctricamente por un microcontrolador(Arduino).





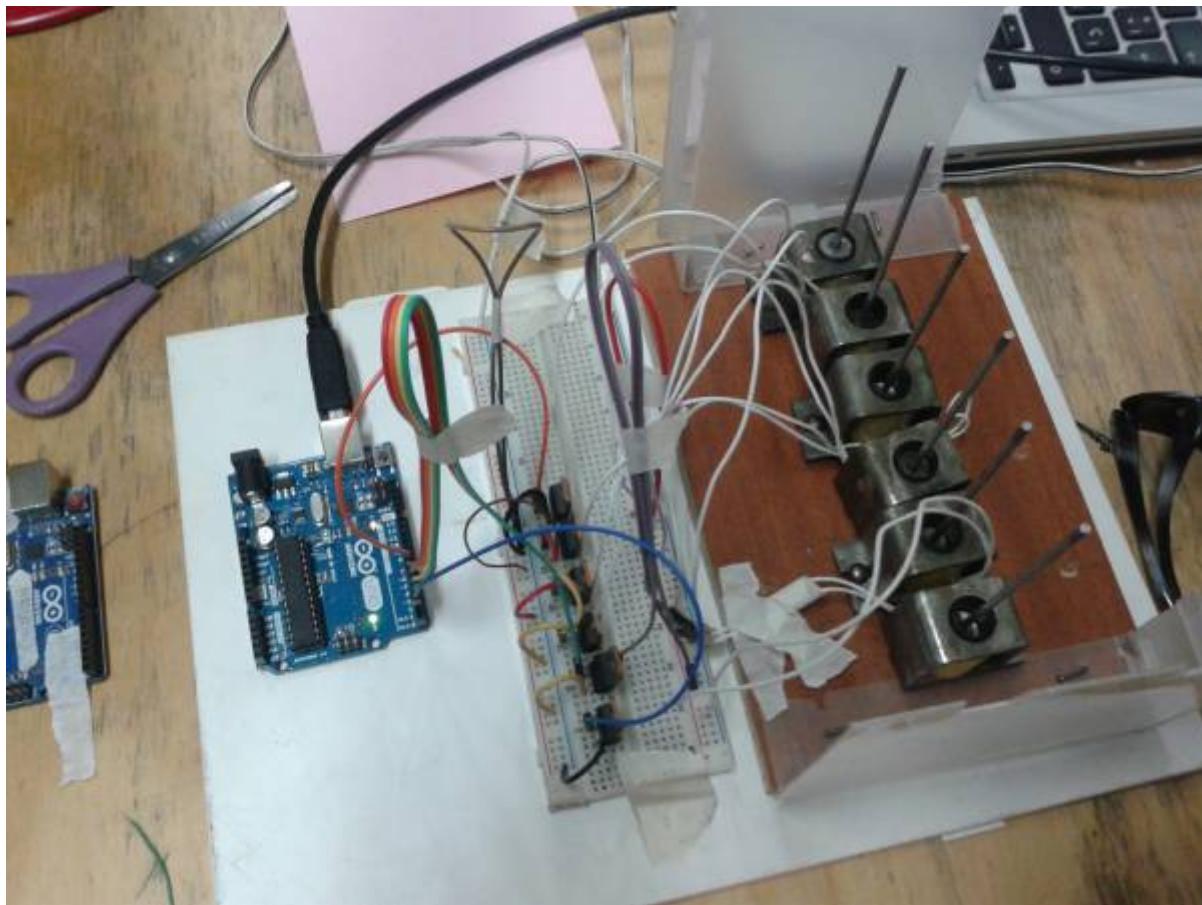


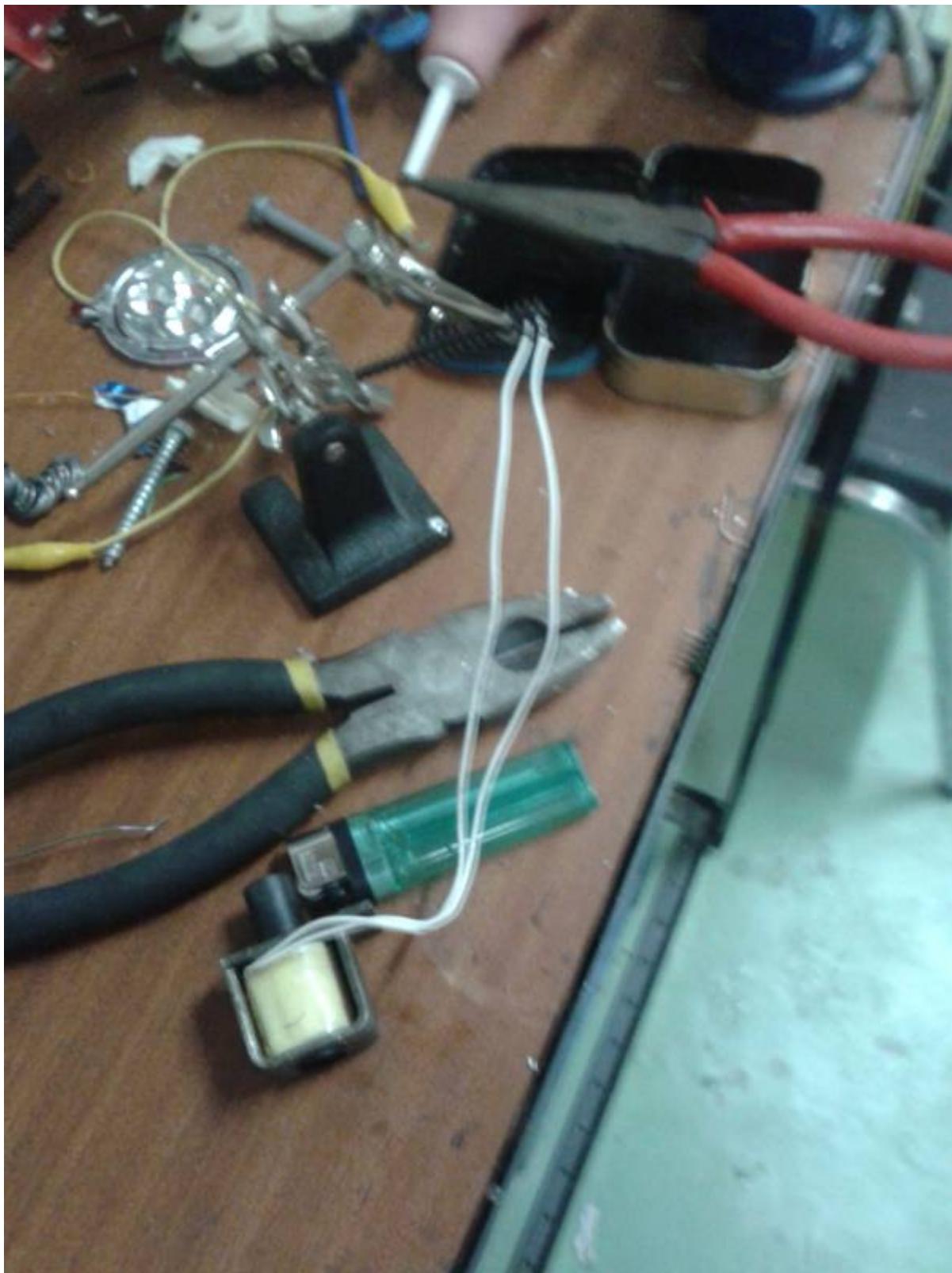
Este microcontrolador es el encargado de convertir notas musicales (datos digitales) en voltaje o datos analogos, que seran los respectivos percutores de cada nota del xilofono.



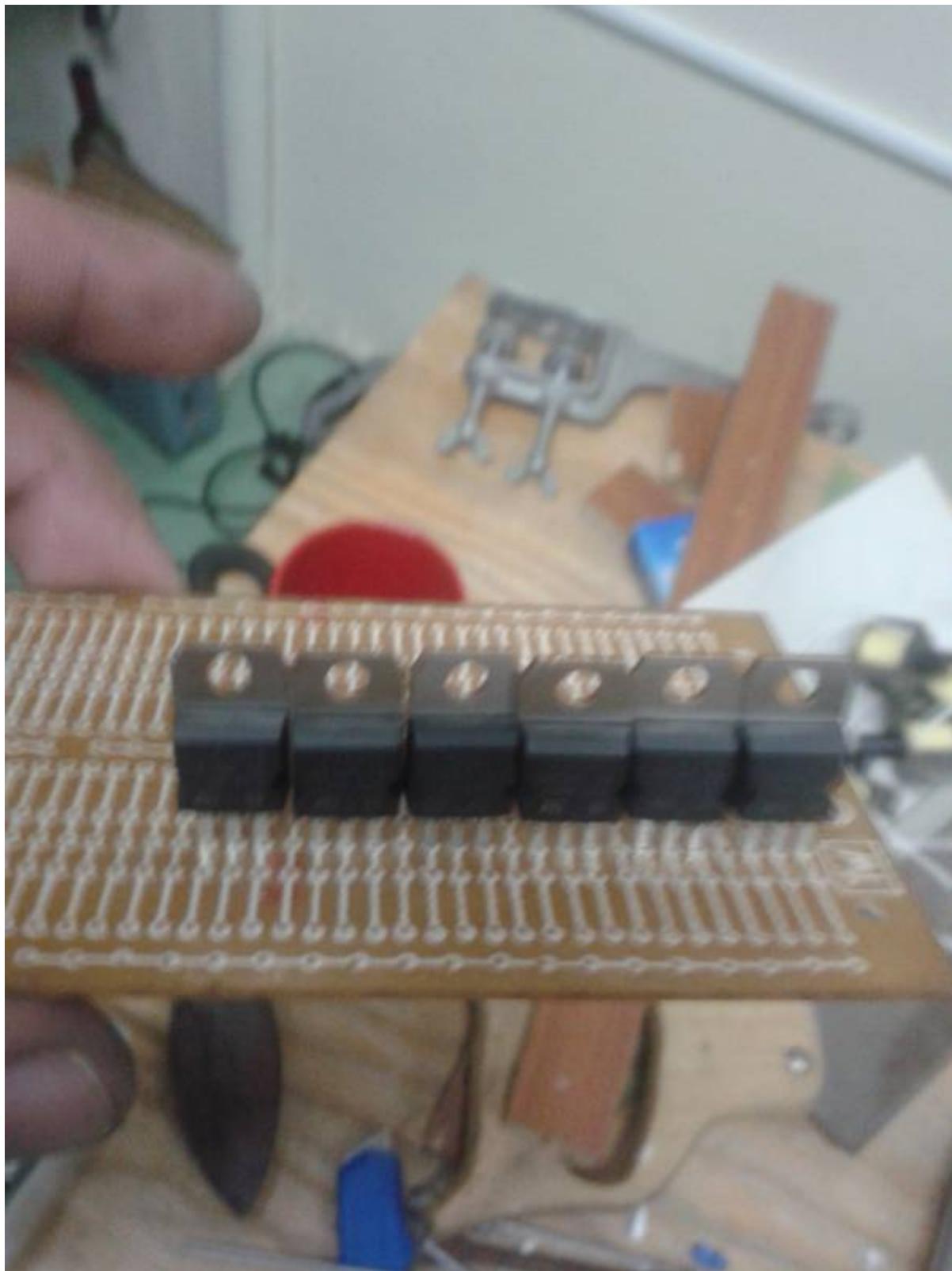
### Elaboracion del circuito electronico

Para que el instrumento pueda funcionar es necesario realizar un circuito para controlar y alimentar los actuadores y estos a su vez, puedan golpear las laminas del xilofono como vimos en el anterior video.



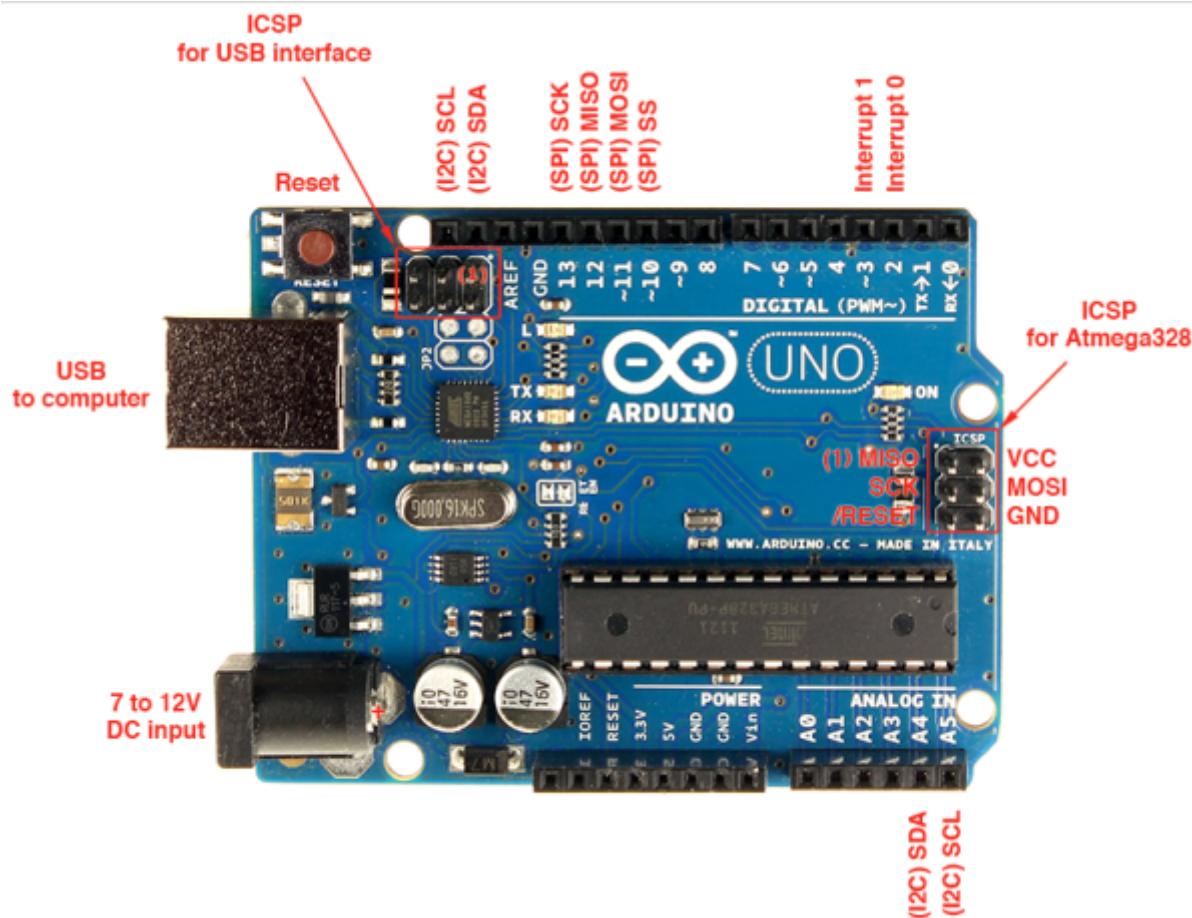


El circuito esta compuesto de varios transistores TIP122, ademas de un adaptador de 9v y algunos diodos para que no hayan cortos en el proceso de alimentacion de los actuadores. Los transistores seran nuestro puente entre la comunicació n del Arduino y el circuito.



### Cambio de firmware en el chip original del Arduino

La placa Arduino que usamos en este proyecto cuenta con un microcontrolador ATmega16U2, que cumple la función de traducir una información a otra, es decir; es un conversor USB-serial. Para que este chip pueda llevar a cabo el proceso de transducción, este es previamente programado a un lenguaje de programación de bajo nivel, y este tipo de programas se llaman Firmware.

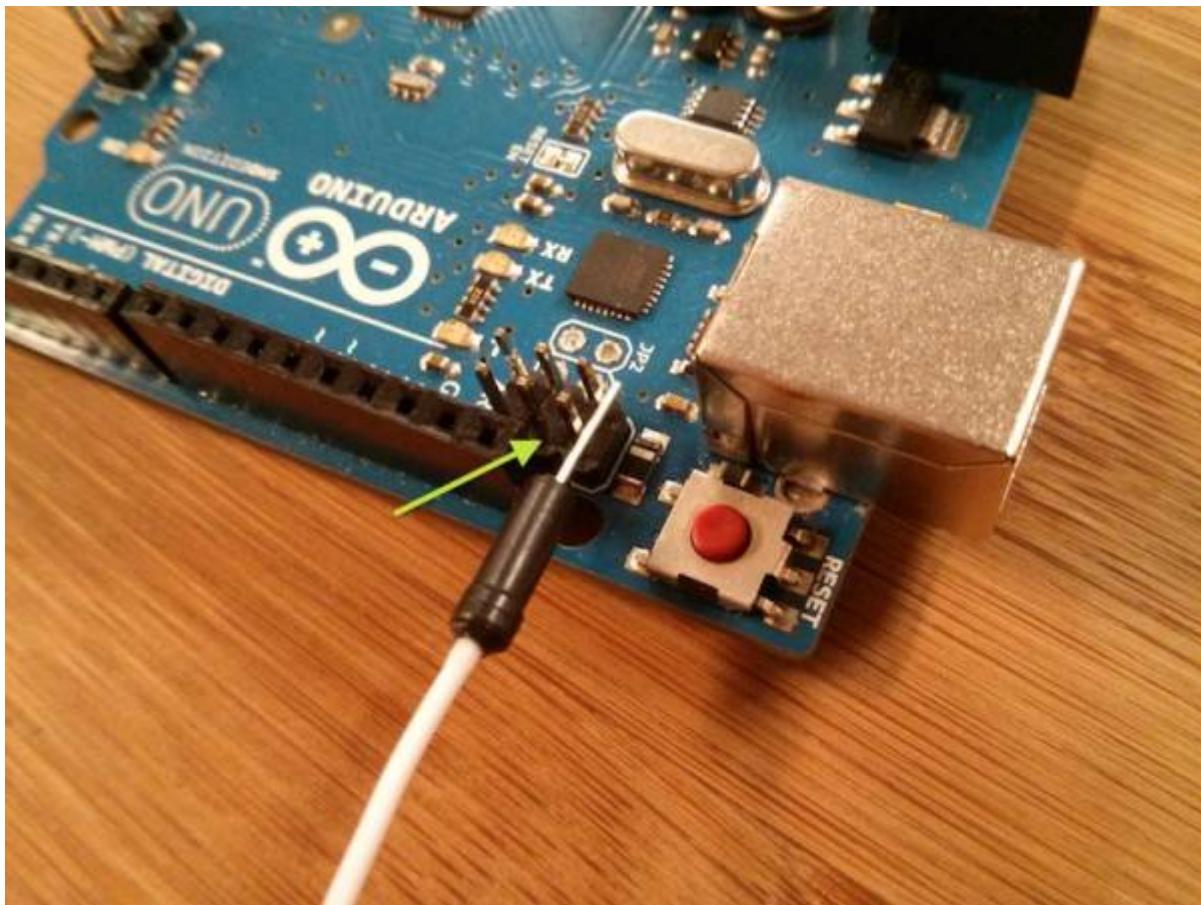


Hiduino es un firmware del chip Atmega16U2 que permite que la placa Arduino sea reconocida por computadores como dispositivos HID, es decir; teclados, mouses y perifericos conectados al computador. (human interface device)

Para flashear el Arduino se pueden tomar diferentes caminos, en este caso usamos Ubuntu 14.04 LTS.25

En general se siguen los siguientes pasos:

1. Primero se abre una terminal, luego se escriben los siguientes comandos: [lubusb-1.0-0 lubusb-1.0-0-dev libusb-dev]
2. Luego instalamos el dfu-programmer escribiendo: [sudo apt-get install dfu-programmer]
3. Conectamos el Arduino por usb y lo ponemos en modo dfu, que consiste en conectar dos pines entre si de la interfaz ISP.



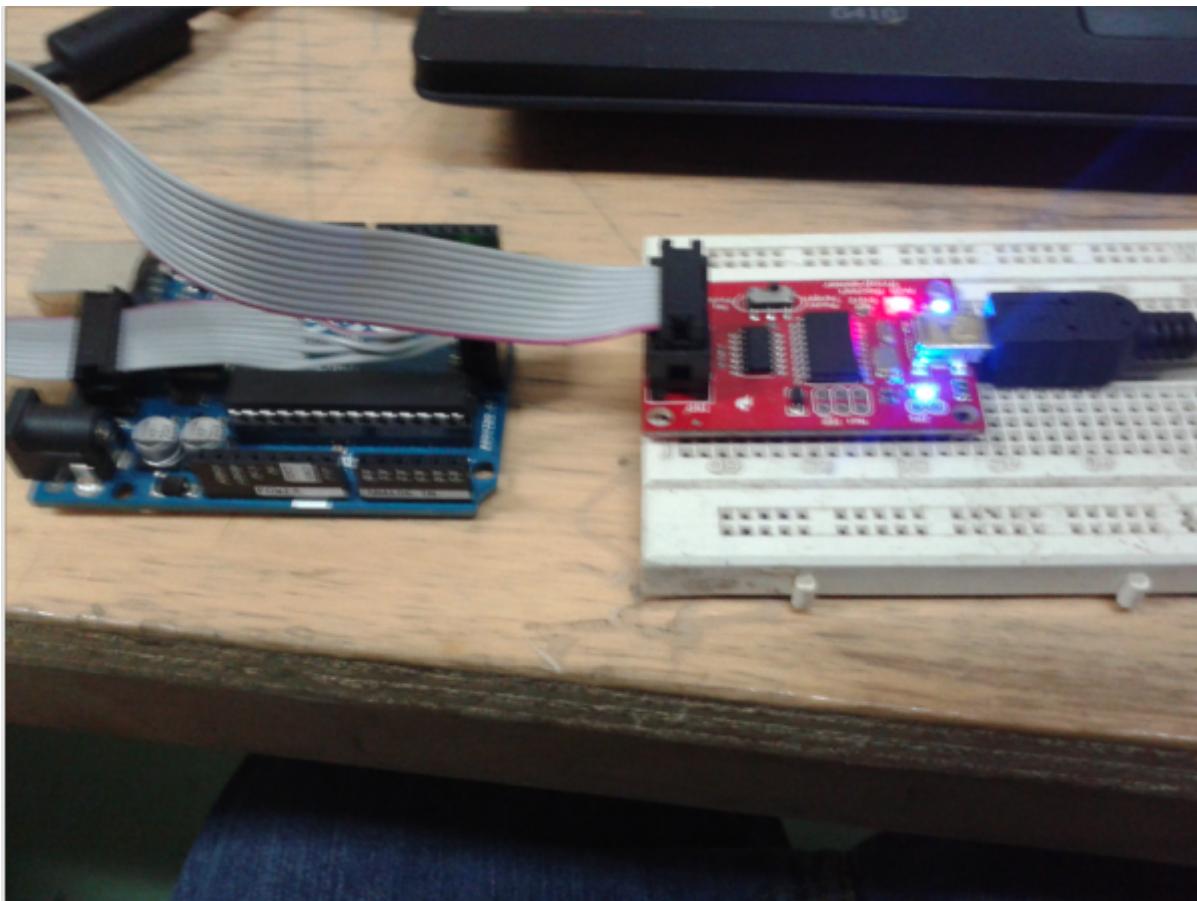
1. Borramos el firmware: [sudo dfu-programmer atmega16u2 erase]
2. Cargamos el nuevo firmware: [sudo dfu-programmer atmega16u2 flash MYFIRMWARE.hex]
3. Reiniciamos el chip: [sudo dfu-programmer atmega16u2 reset]

```

↳ lsusb
Bus 002 Device 002: ID 8087:8000 Intel Corp.
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 8087:8008 Intel Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 004: ID 0bda:0129 Realtek Semiconductor Corp. RTS5129 Card Reader Controller
Bus 003 Device 002: ID 0bda:5728 Realtek Semiconductor Corp.
Bus 003 Device 019: ID 03eb:2fef Atmel Corp.
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
[unloquer@unloquer-2-2014]-(~/Escritorio/MIDUINO/hiduino/Compiled Firmwares)-[git://master ✘]-
↳ ls
dev          usbserial_due_16u2.hex  usbserial_mega_8u2.hex  usbserial_arduino_8u2.hex
HIDUINO_MIDI.hex  usbserial_mega_16u2.hex  usbserial_arduino_16u2.hex
[unloquer@unloquer-2-2014]-(~/Escritorio/MIDUINO/hiduino/Compiled Firmwares)-[git://master ✘]-
↳ sudo dfu-programmer atmega16u2 flash HIDUINO_MIDI.hex
Checking memory from 0x0 to 0xCFF... Empty.
0%           100% Programming 0xD00 bytes...
[>>>>>>>>>>>>>>>>>] Success
0%           100% Reading 0x3000 bytes...
[>>>>>>>>>>>>>>>>>] Success
Validating... Success
0xD00 bytes written into 0x3000 bytes memory (27.08%).
[unloquer@unloquer-2-2014]-(~/Escritorio/MIDUINO/hiduino/Compiled Firmwares)-[git://master ✘]-
↳ lsusb
Bus 002 Device 002: ID 8087:8000 Intel Corp.
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 8087:8008 Intel Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub

```

Despues de hacer este proceso, es conveniente programar el chip Atmega que controla los pines por medio de un programador AVR usando las entradas ISP del Arduino.



El programador ISP es necesario porque una vez el arduino es flasheado, **no pueden subirse programas por el puerto USB**. Ahora el paso a seguir es programar los Pines del Arduino usando una librería Midi, para que la placa reconozca los mensajes Midi provenientes del DAW.

### **Programacion de la placa Arduino para que funcione como Input-MIDI debido a las condiciones motoras del Instrumento.**

Luego de que el chip ATmega esta flasheado, procedemos a programar las salidas digitales de la placa segun nuestras necesidades. Este proceso se lleva a cabo mediante el Ide proporcionado por Arduino para programar la placa.

```
HIDUINO_MIDI_Example02
/*
 * HIDUINO_MIDI_Example02
 * Example: MIDI Input
 * by Dimitri Diakopoulos (http://www.dimitridiakopoulos.com)
 * Last Updated: 17 January 2013
 */

#include <MIDI.h>

void HandleNoteOn(byte channel, byte pitch, byte velocity) {
    // Do something here with your data!
}

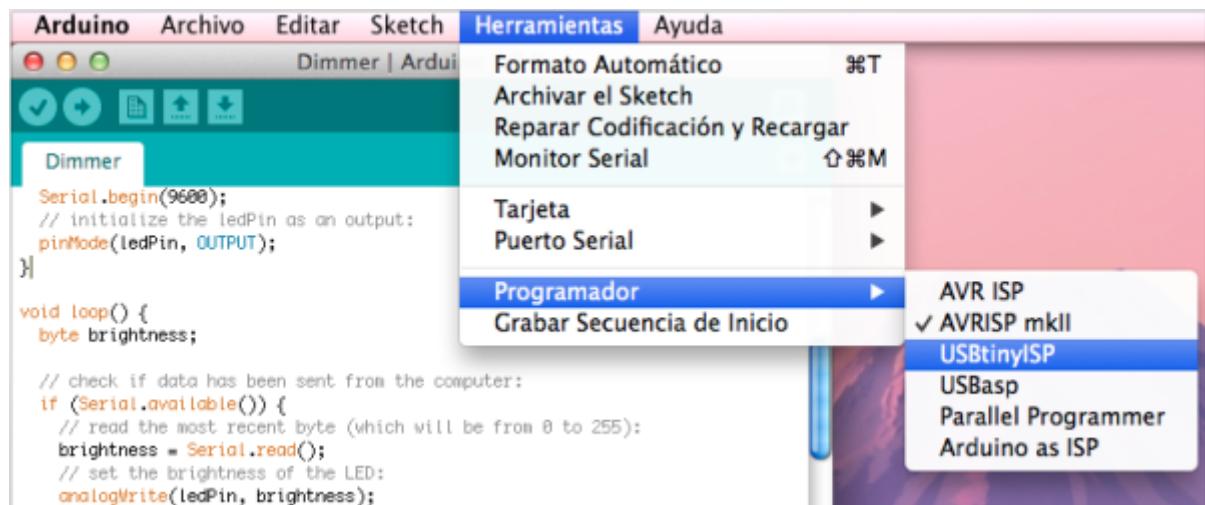
void HandleCC(byte channel, byte pitch, byte velocity) {
    // Do something here with your data!
}

void HandleNoteOff(byte channel, byte pitch, byte velocity) {
    // Do something here with your data!
}

void setup() {
    MIDI.begin(MIDI_CHANNEL_OMNI);

    // As of the MIDI Library v3.1, the lib uses C style function
    // pointers to create a callback system for handling input events.
    MIDI.setHandleNoteOn(HandleNoteOn);
    MIDI.setHandleControlChange(HandleCC);
    MIDI.setHandleNoteOff(HandleNoteOff);
}
```

En este caso necesitamos reconocer notas midi que seran programadas en el software Ableton Live. La programacion del codigo se basa en escribir funciones que reconozcan velocity, channel y pitch, que son variables del protocolo Midi para reconocer que nota es, que dinámica tiene esa nota, y un canal que le dice al computador cual es ese instrumento Midi.



Cuando este código en software esta listo, procedemos a conectar el programador ISP y luego abrimos la interfaz de Arduino, es muy importante elegir el programador a usar, porque ya no estamos usando la interfaz USB. Para seleccionar el programador adecuado, por favor mire la imagen de arriba.

Con los pasos descritos anteriormente hemos finalizado la construcción del instrumento robotico, pasamos entonces a la composición en el software LSDJ.

Aqui un video del flasheo del arduino.

[programando\\_isp.mp4](#)

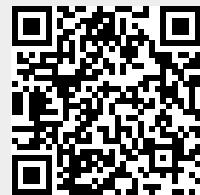
# Programación y composición de la Música en el Tracker de la Gameboy

From:

<https://wiki.unloquer.org/> -

Permanent link:

<https://wiki.unloquer.org/proyectos?rev=1477260180>



Last update: **2016/10/23 22:03**