Hiduino - 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. Elaboración y diseno del instrumento Robotico.
2. Programación y composicion en el Tracker de la Gameboy
3. Programación y composicion en el Tracker de la Gameboy.
4. Sincronización midi Gameboy como MidiOut y controlador USB (Hiduino) como esclavo.
5. Programación y composicion en el computador con la musica construida en la Gameboy.
6. Composición de la musica final añadiendo las partes y melodias del instrumento robotico.

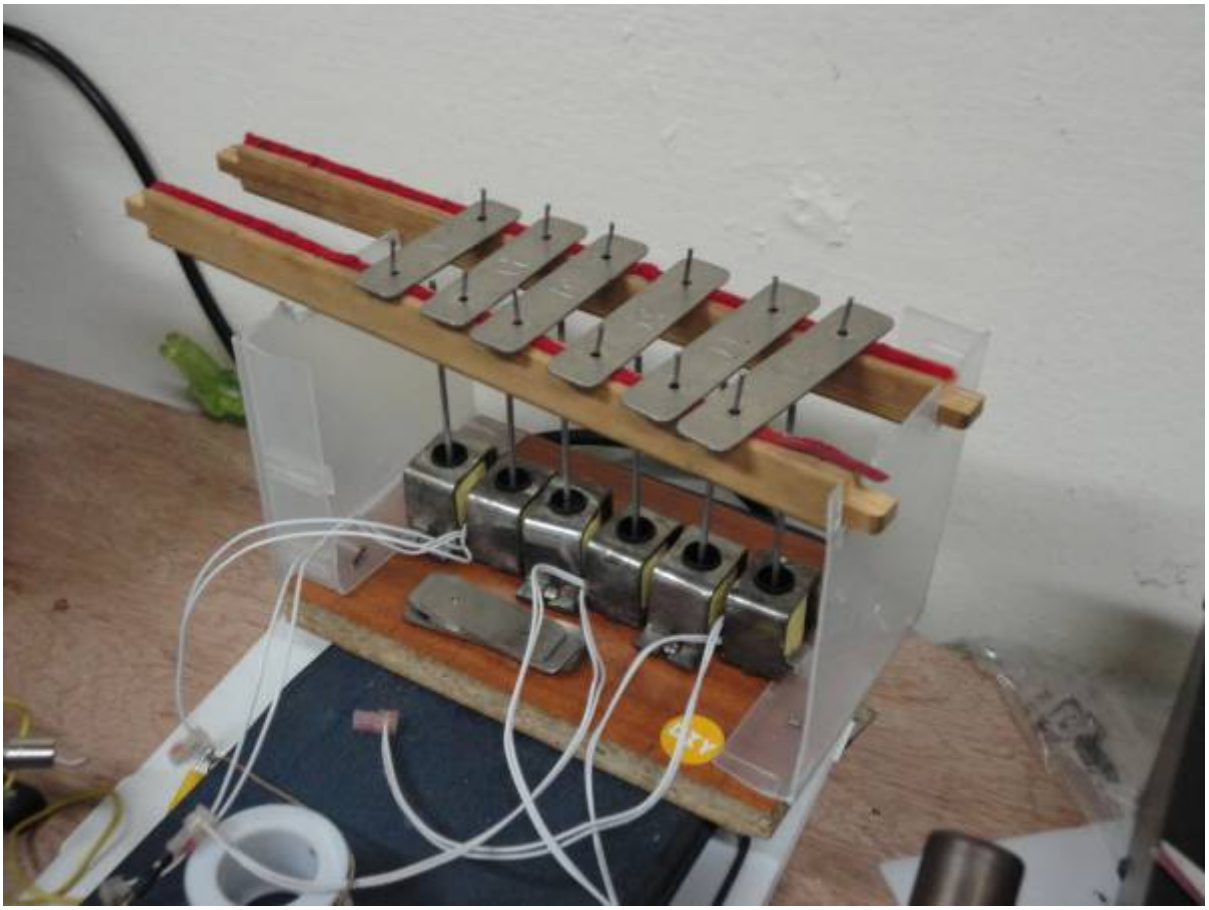
Elaboración y diseño del instrumento robotico

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

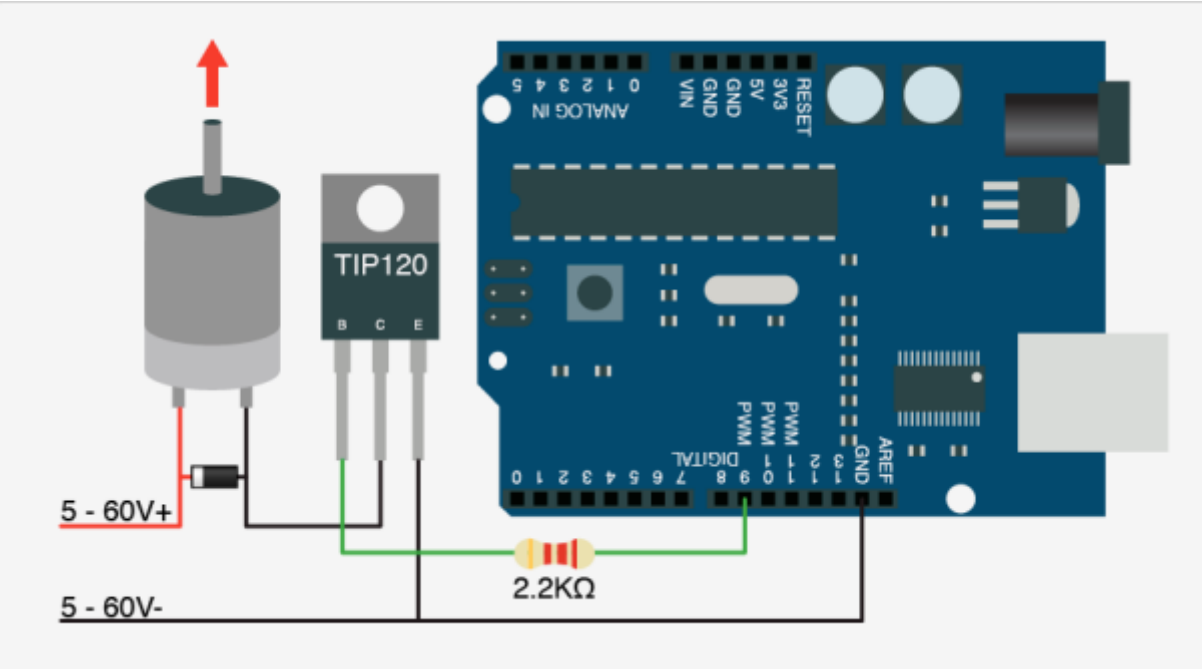
Construcción del instrumento

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

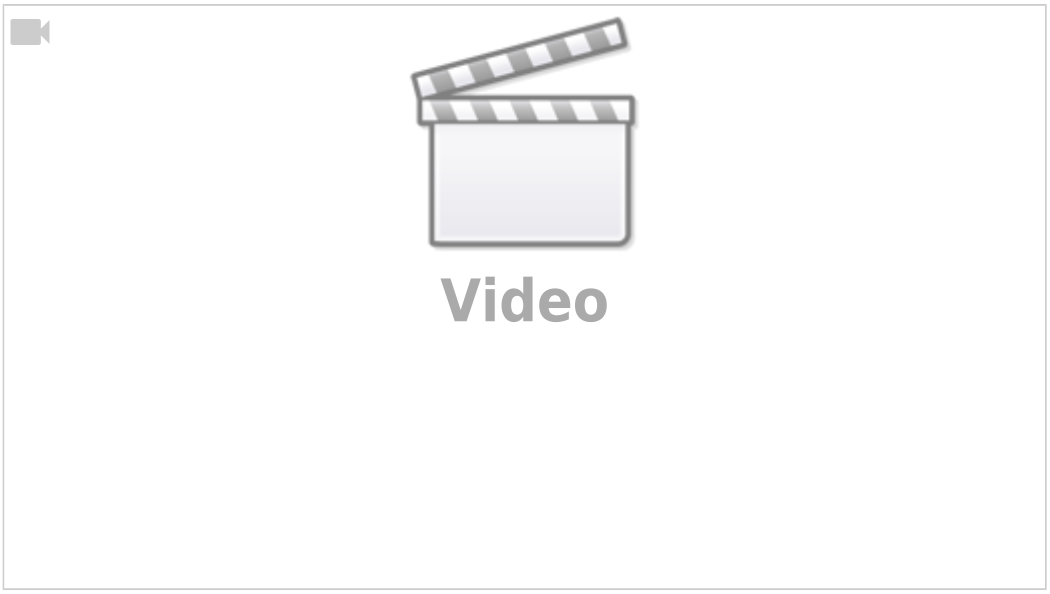
Para llevar a cabo este proceso se experimenta con varios modelos, alturas y voltajes que llegan a los actuadores hasta encontrar la posicion ideal. Posteriormente se construye un circuito electrónico que permite accionar cada percutor y este a su vez; es controlado electricamente por un microcontrolador(Arduino).



000
090

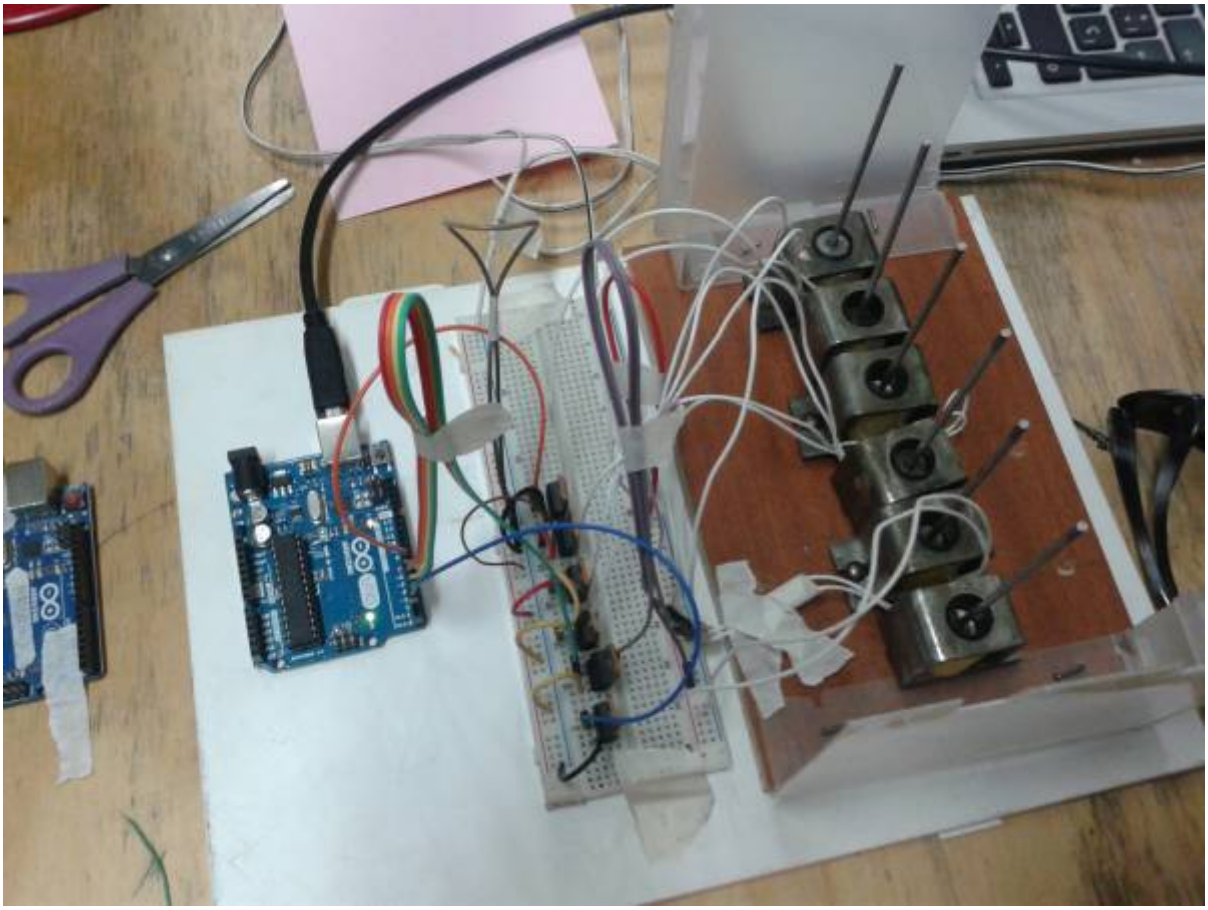


Este microcontrolador es el encargado de convertir notas musicales (datos digitales) en voltaje o datos analogos, que seran los respectivos percutores de cada nota del xilofono.

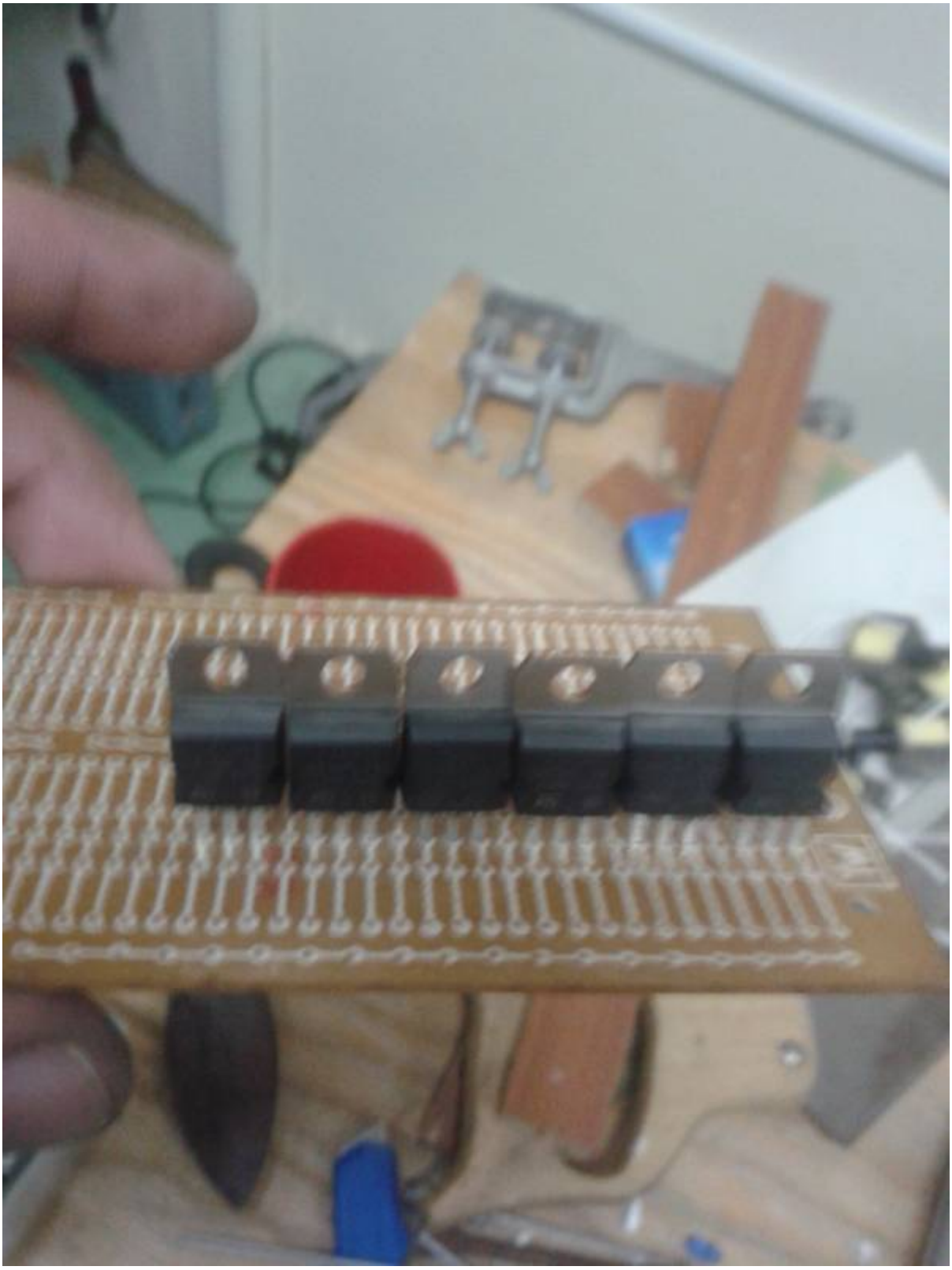


Elaboracion del circuito electrónico

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.

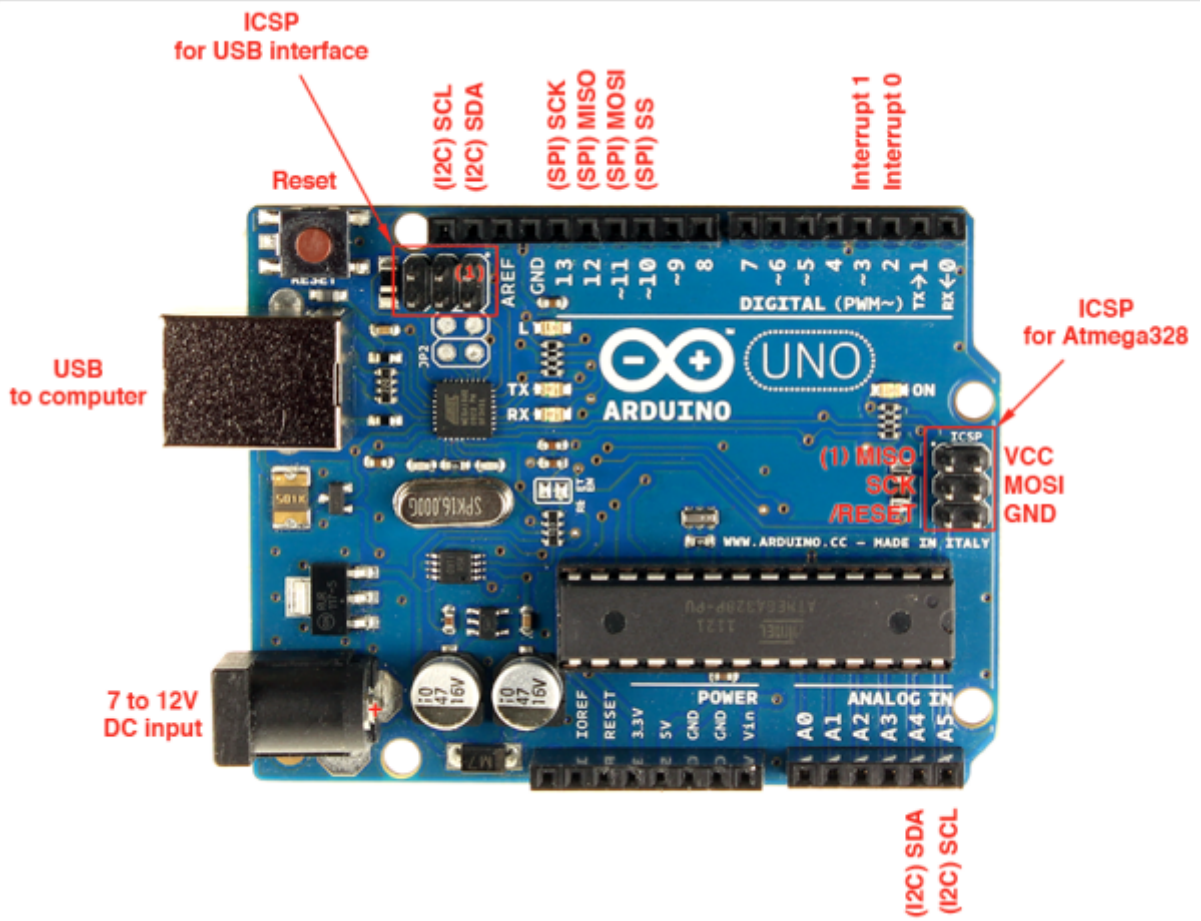


El circuito esta compuesto de varios transistores TIP122, además de un adaptador de 9v y algunos diodos para que no hayan cortos en el proceso de alimentacion de los actuadores. Los transistores serán nuestro puente entre la comunicación del Arduino y el circuito.



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 los datos de serial a usb, es decir; es un conversor USB-serial. Para que este chip pueda llevar a cabo el proceso de transduccion, este es previamente programado a un lenguaje de programacion de bajo nivel, y este tipo de programas se llaman Firmware.

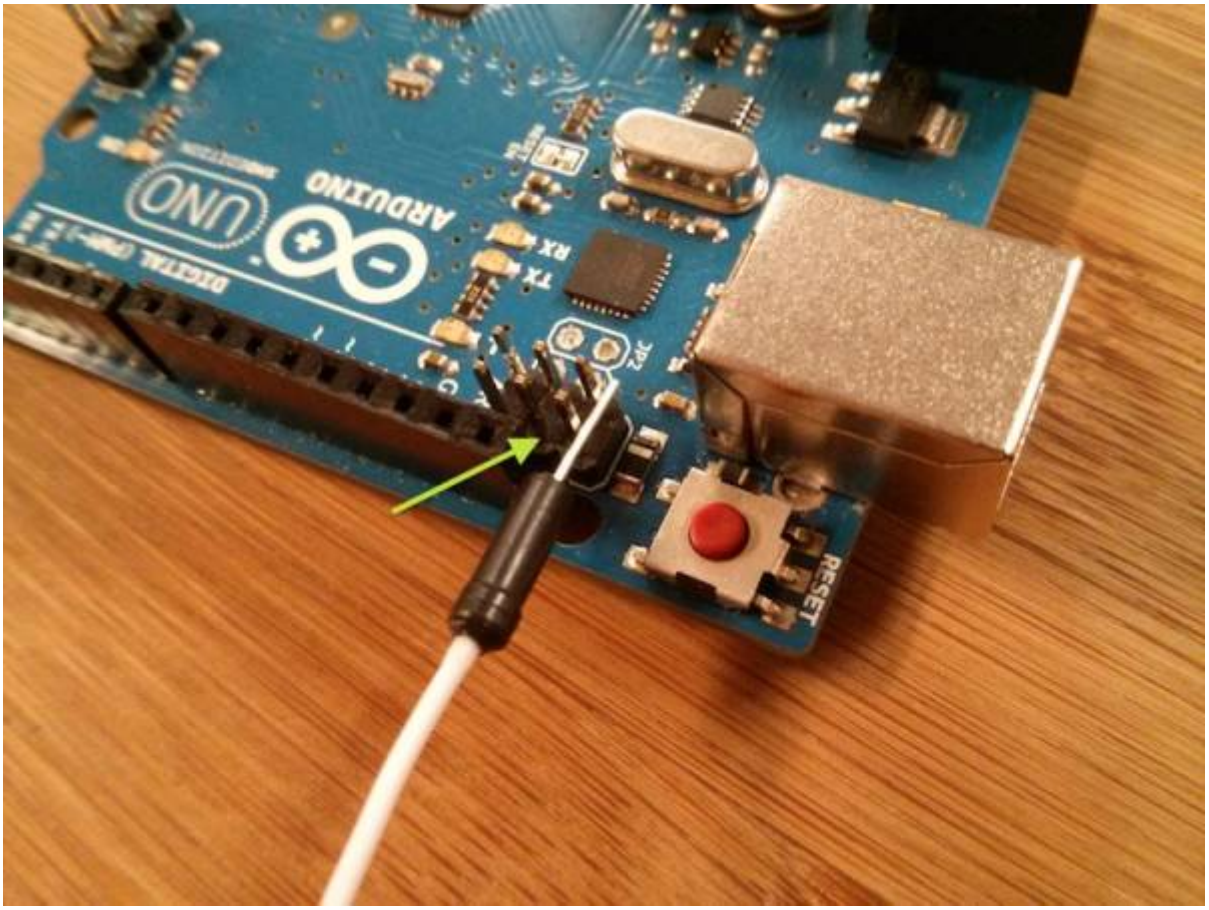


Hiduino es un firmware del chip Atmega16U2 que permite que la placa Arduino sea reconocida por computadores como dispositivos HID, es decir; teclados, mouses y perifericos conectados al computador. (human interface device)

El proceso de cambiar un firmware a otro se le conoce como *flashear*. 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:

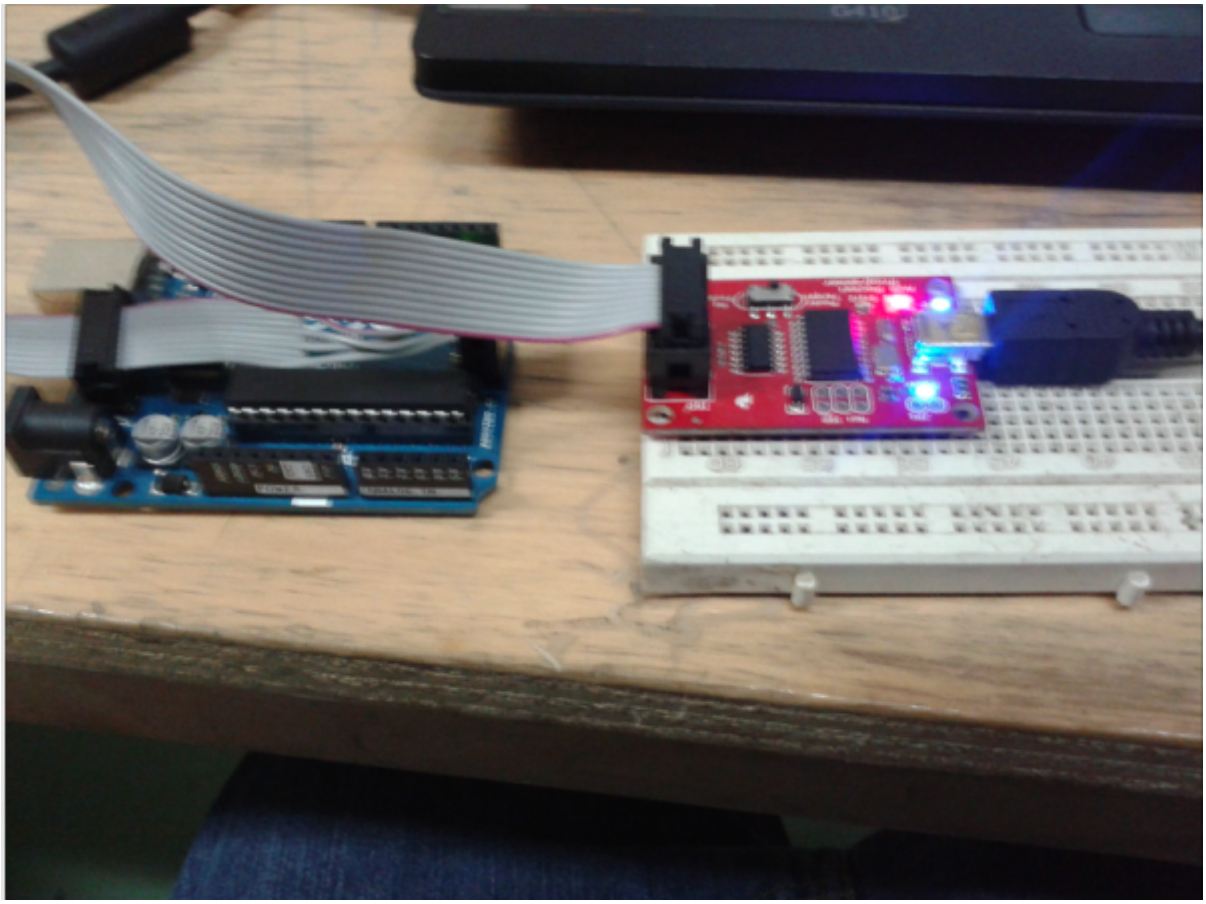
1. Primero se abre una terminal, luego se escriben los siguientes comandos: [lubusb-1.0-0 libusb-1.0-0-dev libusb-dev]
2. Luego instalamos el dfu-programmer escribiendo: [sudo apt-get install dfu-programmer]
3. Conectamos el Arduino por usb y lo ponemos en modo dfu, que consiste en conectar dos pines 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_uno_8u2.hex
HIDUINO_MIDI.hex   usbserial_mega_16u2.hex  usbserial_uno_16u2.hex
┌[                  -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%).
┌[                  -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 libreria Midi, para que la placa reconozca los mensajes Midi provenientes del DAW.

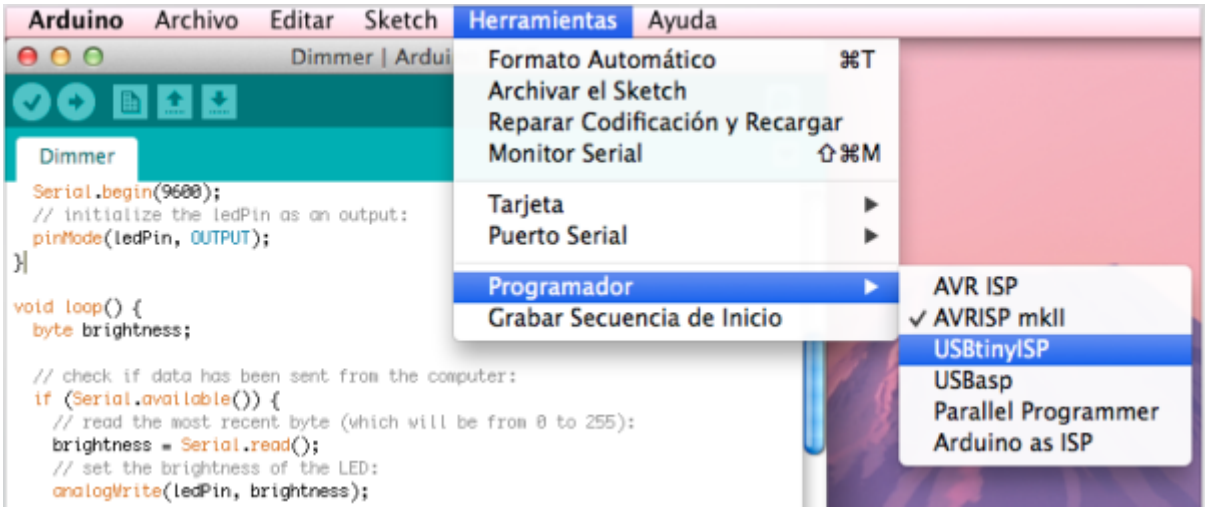
Programacion de la placa Arduino para que funcione como Input-MIDI debido a las condiciones motoras del Instrumento.

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 programación del código 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.

El ISP es una placa electrónica que nos permite programar el Arduino por los pines ISP que dispone. Es muy importante elegir el programador a usar en el IDE del Arduino, porque ya no estamos usando la interfaz USB para hacerlo comunmente. Para sellecionar el programardor 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

Este proceso se lleva a cabo mediante el software LSDJ corriendo en la Gameboy, los tracker's son programas que dependiendo de su interfaz de entrada diversifican su funcionalidad y funcionamiento para llevar a cabo la composición. De este modo, el ejercicio de la composición varia dependiendo del tracker y de su hardware.

INSTRUMENT 00
PU1
NAME
TYPE
PULSE
♪128
ENVELOPE
A8
WAVE
1
OUTPUT
LR
2
LENGTH
UNLIM
W
SWEEP
FF
N
VIB. TYPE
HF
PU2 TUNE
00
PU FINE
0
AUTOMATE
OFF
TABLE
OFF
S
LCPIT
G

LSDJ es un tracker que tiene la posibilidad de correr en un cartucho de la Gameboy, este recibe comandos en sistema hexadecimal para escribir y componer partes de música, estas letras y numeros se escriben en diferentes ventanas del programa, para identificar al usuario si esta escribiendo una melodía, o esta creando un instrumento o si esta procesando algún instrumento. En la vista Song es en donde se escriben pequeños compases que contienen diferentes instrumentos y melodías, dentro de cada dato que agregamos, podemos crear pequeños loops musicales o rítmicos dependiendo del canal en el cual estemos componiendo.

Ademas de esto el software LSDJ proporciona otras ventanas para crear sonidos sintetizados, disparar muestras de sonido o modificar el groove del tempo.

Todo el proceso de composición en la Gameboy por lo general se lleva a cabo siguiendo algunos pasos descritos, por consiguiente; cuando se tiene alguna cantidad de música escrita en la Gameboy, se procede a la sincronización Midi.

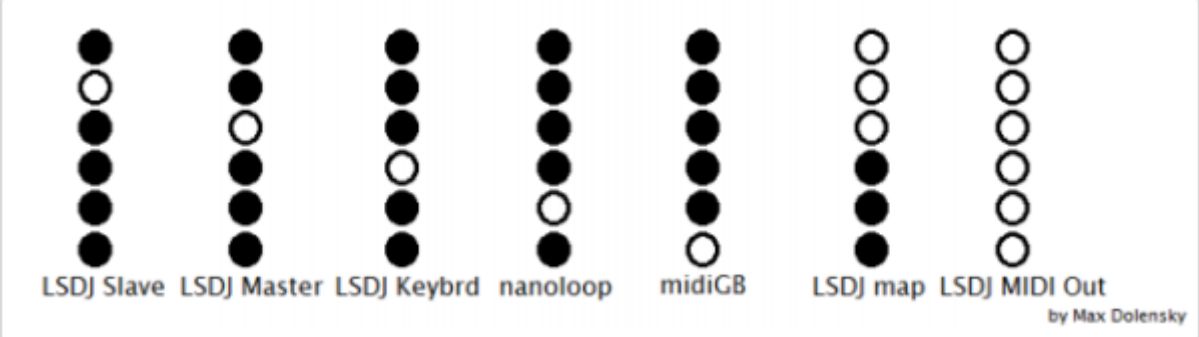
Sincronización Midi Gameboy como MidiOut y controlador USB-boy como esclavo

Cuando la música es finalizada o se tiene parte de ella, se procede entonces a sincronizar el Gameboy

y la computadora por medio del protocolo MIDI. Para llevar a cabo este proceso se necesitan dos herramientas: un cable de interconexión para Gameboy y una interfaz usb para hacer la traducción de mensajes hexadecimales a mensajes MIDI. (Arduino Boy o USB boy)

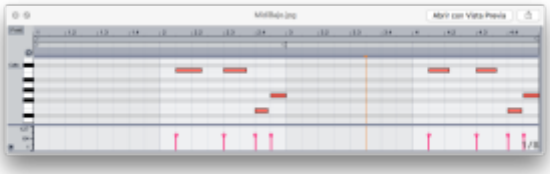
Primero entonces se procede a elegir el modo Midi-output en LSDJ, este modo permite programar comandos en el tracker que serán conducidos por una interfaz usb a nuestro DAW, que para este proyecto es Ableton Live.

EL proceso de sincronización es relativamente sencillo. El dispositivo USB-boy se dispone en modo LSDJ-OUTPUT para que pueda entender los mensajes midi hexadecimales provenientes del Gameboy,esto se hace presionando el botón “mode” que tiene la interfaz, y en Ableton Live buscamos la interfaz usb que tiene el nombre Teensy Midi.Después de tener estos pasos realizados, el computador esta sincronizado remotamente con el Gameboy.



Programación y composición en el computador con la música construida en la Gameboy

Después de que la sincronización esta realizada, procedemos entonces a programar los diferentes sonidos que serán acompañantes de la Gameboy. Esto se hace componiendo diferentes sonidos en Ableton Live, reciclando pequeñoos loops sonoros y componiendo ademas los efectos, ambientes, bajos y pads que retocaran el ambiente musical que proporciona el Gameboy.



Composición de la música final añadiendo las partes y melodias del instrumento robotico

Cuando iniciamos la labor de programar musicalmente el instrumento robotico, establecemos cuales serán las notas MIDI entrantes para hacer que los solenoides se muevan, paso realizado anteriormente, como ya conocemos esas notas, solo falta ir a las preferencias de Ableton Live y seleccionar el controlador “Hiduino”.

Seleccionamos en Ableton Live la salida MIDI de nuestro instrumento y procedemos a programar las notas coo se ve en las imagenes de arriba. Para programar el instrumento robotico se llevan a cabo

los mismos procesos cuando se agregan o graban notas midi en un instrumento virtual. Después de grabar algunas notas en un canal de Ableton Live, se puede escuchar los siguientes resultados.

Este es un audio de la música en el Gameboy.

gameboy-solo.wav

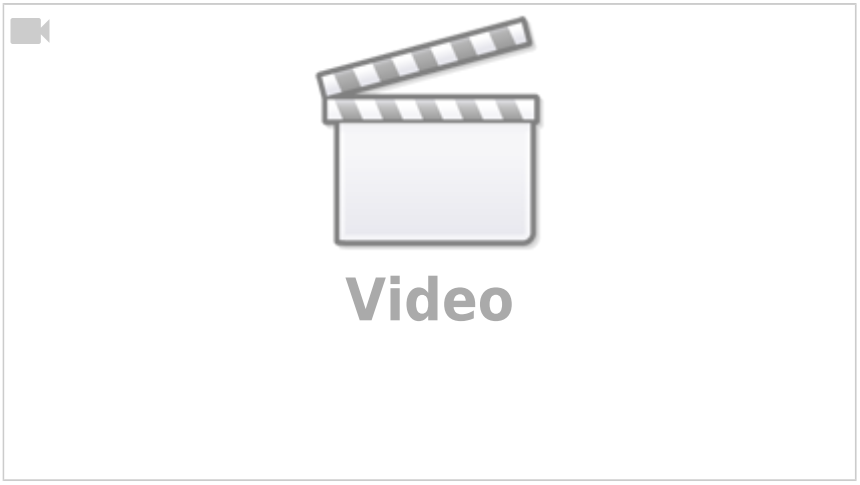
Este es un audio de la música en el Gameboy acompañada del instrumento robotico.

gameboy_instrumento.wav

Finalmente tenemos todo junto, gameboy, instrumento y música compuesta en la computadora para reforzar todo.

gameboy_instrumento_ableton.wav

Esta es una pequeña muestra de la interpretación en vivo.



From:
https://wiki.unloquer.org/ -

Permanent link:
https://wiki.unloquer.org/personas/johnny/proyectos/hiduino

Last update: **2016/10/26 21:07**