

Cuadernillo de Prácticas

Laboratorio de Digital – Cuadernillo 3b

ÁREA DE TECNOLOGÍA ELECTRÓNICA - URJC

DESCRIPCIÓN DEL CUADERNILLO

En este cuadernillo como fabricar vuestro propio piano controlador usando Arduino y MIDI. Muchos diréis ¿Qué es eso del MIDI? ¿Para qué sirve? Pues muy sencillo, MIDI (Musical Instrument Digital Interface) es un protocolo de comunicación serial que permite conectar ordenadores, sintetizadores, controladores y demás elementos destinados a la creación de sonidos. Dicho protocolo data de los años ochenta pero aún se sigue usando hoy en día y de manera muy extendida, nos podemos encontrar gente que lo usa hasta para controlar luces y servos.

CONSULTA los Anexos para completar tu trabajo (flashear Arduino para MIDI y led RGB). Estos anexos sirven para realizar partes optativas que sumarán cada una 1 punto adicional a tu nota.

Realizar el piano Fase 1 hasta la simulación es 1 punto. Además, si consigues bajar a placa HW la Fase 1 tendrás +1 punto (Esto se debe realizar en casa y entregarlo en físico en la sesión 3 – optativo).

Realizar el flasheo del piano Fase 2 y bajar a placa la Fase 2 son 2 puntos. Esto se debe realizar en casa y entregarlo en físico en la sesión 3 – optativo.

Realizar el piano Fase 3 son 4 puntos máximo. Por un lado, la simulación son 2 puntos. Por otro lado, bajar a placa la Fase 2 son otros 2 puntos (Esto último de bajar a placa, si no te da tiempo en el aula, se puede realizar en casa y entregarlo en físico en la sesión 3 – optativo).

Parte 1: Piano a

3. Piano electrónico

3.1. NOMBRE Y DESCRIPCIÓN DEL SISTEMA: Piano Electrónico

Desarrollaremos un piano electrónico. Este cuadernillo se rellena de manera diferente al resto. Tienes que completar los apartados que te dice: "A completar".

Lo que haremos será usar varios botones conectados al Arduino que leerá cuando se pulsan. En el caso de que se pulse uno haremos que Arduino envíe a nuestro ordenador por medio del cable USB unos datos indicando que botón ha sido pulsado y qué nota debe enviar (un Do, un Re,...). Una vez en el ordenador deberemos usar algún programa como Traktor, Ableton, Cubase, Nuendo, Reason,... que interpretará los datos MIDI recibidos y generará un sonido que podremos escuchar por los altavoces de nuestro ordenador. Este último paso es importante ya que **MIDI es solamente un protocolo**, es decir, **no suena por sí solo**.

Como Arduino no es un dispositivo pensado para estos fines, no puede ser reconocido por los programas de audio mencionados anteriormente como si fuera un dispositivo midi comercial. Una de las formas es emplear unos convertidores Serial a MIDI y unos puertos virtuales (cosa muy extendida por Internet, como HairLess y MidiYoke). Sin embargo, lo que vamos a hacer es **reflashear nuestro Arduino UNO con un nuevo firmware llamado Hiduino**, lo que nos permitirá crear un dispositivo "Plug & Play" **como si fuera un controlador MIDI comercial**.

3.2. MATERIAL HARDWARE NECESARIO

Material necesario para el desarrollo de la Fase 1 y la Fase 2:

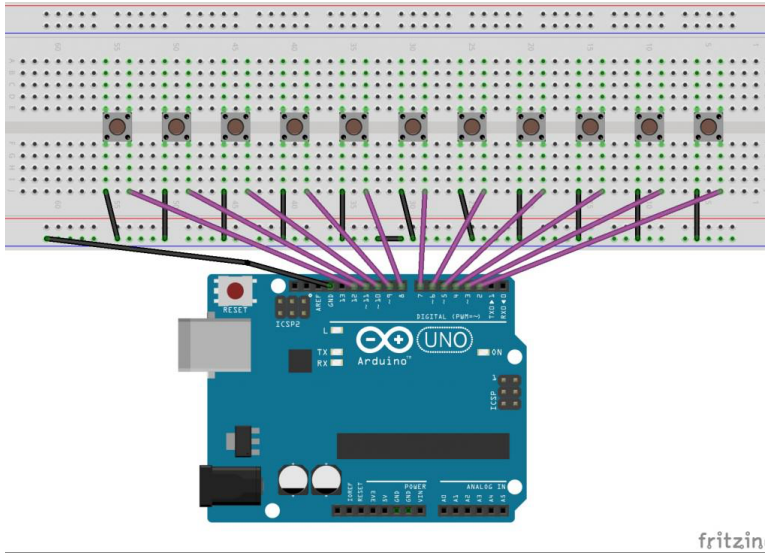
- Kit de prácticas.
- La placa de Arduino.
- x11 Switches (también se pueden usar menos)
- x01 Resistencia 10k Ohm
- Arduino UNO y cable USB
- ProtoBoard
- Programa Arduino
- DFU-Programmer (sobre Linux o Mac)
- Hiduino Firmware
- Librería MIDI
- Cables de inserción.
- Placa protoboard.

3.3. CONEXIONES

Lo primero **conectar los 11 botones**.

Como podéis ver la conexión es muy sencilla, conectamos cada botón entre GND y el pin correspondiente (del 2 al 12). Para que se pueda funcionar así, será

necesario **habilitar las resistencias de pullup internas** (se hace desde el código). **Ojo si usáis el pin 13**, que requiere una resistencia de pullup externa.



fritzing

3.3. CÓDIGO DE VERIFICACIÓN SIN MIDI- FASE 1

Vamos a cargar un código que **verifique el correcto funcionamiento** de todos los botones. **Aún no hemos empezado con MIDI. TABULA EL CÓDIGO QUE VIENE A CONTINUACIÓN CORRECTAMENTE. RELLENA POR TI MISMO LO QUE PONE XXX**

*/** Este ejemplo es para ver en el monitor serial los botones pulsados/despulsados En el siguiente ejemplo veremos como enviar midi Ojo con usar mas de 11 botones empleando la resistencia de pullup interna Para el pin 13 debe usarse una resistencia de pullup externa */*

```
const int numBotones=XXX;    // Definir aqui el numero de botones (11 max) int
valorLeido[numBotones];    // Array con los valores leidos de cada boton int
valorAnterior[numBotones]; // Array con el valor anterior leido

void setup() {
  Serial.begin (XXX);        // Inicializamos la comunicacion serial  for(int i=0;
  i<numBotones; i++){
    pinMode(i+2, INPUT_PULLUP); // Recorremos cada pin y lo ponemos como
    entrada  valorAnterior[i]=1; // Inicializamos el valor anterior a 1 (no pulsado)
  }
} void loop(){
  for(int i=0; i<numBotones; i++)
  {  valorLeido[i]=digitalRead(i+2); // Leer valor    // Solo enviamos cuando ha existido
    un cambio, para no sobrecargar el puerto serial
```

```

if((valorLeido[i]==0) && (valorLeido[i]!=valorAnterior[i])){
    Serial.print("Pulsado el boton: ");
    Serial.println(i);
    Serial.println("-----");
} else if ((valorLeido[i]==1) && (valorLeido[i]!=valorAnterior[i])){
    Serial.print("Ya no el boton: ");
    Serial.println(i);
    Serial.println("-----");
}   valorAnterior[i]=valorLeido[i]; // El valor anterior es el nuevo valor para la siguiente
iteracion
}
}

```

Para probarlo, debes abrir el puerto serial con un **baud rate de 9600** e ir pulsando y liberando los botones. Podréis ver como al pulsarlo se nos indica por el monitor serial que botón ha sido pulsado, y cuando se libera cual ha sido liberado.

Recuerda que hasta aquí no hemos hecho nada de MIDI, simplemente estamos testando el hardware e introduciendo el código que usaremos.

3.4. DIAGRAMA DE BLOQUES FASE 1 – A completar

Introduce aquí el diagrama de bloques para que explique el funcionamiento del código

3.5. RECOMENDACIONES PARA LA FASE 1

Para este montaje te debería valer al menos el montaje del semáforo. Queremos que funcione el código del semáforo o el del piano en la misma placa hardware.

Si lo presentas sólo en el simulador, no es necesario que se cumpla este requisito.

NOTA: ANTES DE BAJARLO A PLACA - Te recomendamos que primero lo hagas con el Simulador antes de hacerlo en la placa real.

3.6. SIMULACIÓN SIN MIDI– FASE 1

Introduce aquí el enlace del vídeo o fotos de la simulación.

3.7. CÓDIGO DE VERIFICACIÓN CON MIDI– FASE 2 – optativo (+2 puntos)

A continuación debemos cargar el siguiente código. Este código hace uso de la [librería MIDI de Arduino](https://github.com/ddiakopoulos/hiduino). (**Descargar la librería e instalarla**). Simplemente se emplea un baud rate de 31250 y se envía mensajes serial con la sintaxis de MIDI. Nos centraremos en el mensaje de NoteOn es decir, nota activada. Este mensaje lleva información de la nota pulsada (Do, Re,...), con que intensidad se pulsó esta nota (127 es el máximo, nosotros usaremos el máximo ya que nuestros botones no tienen sensibilidad) y en que canal MIDI se quiere enviar el mensaje (hay 16 disponibles). **Cargamos el código, nada más.**

```
#include // Incluimos la libreria MIDI /** * * Este ejemplo sirve
para enviar midi a nuestro ordenador * Sera necesario emplear el
firmware hiduino para que funcione correctamente *
https://github.com/ddiakopoulos/hiduino * Ojo con usar mas de 11
botones empleando la resistencia de pullup interna * Para el pin 13
debe usarse una resistencia de pullup externa */

const int numBotones=11;

int valorLeido[numBotones]; // Array con los valores leidos de cada
boton int valorAnterior[numBotones]; // Array con el valor anterior
leido MIDI_CREATE_DEFAULT_INSTANCE();

void setup() {

    MIDI.begin(); // Inicializamos la comunicacion
midi (realmente es serial)

    for(int i=0; i<numBotones; i++){

        pinMode(i+2, INPUT_PULLUP); // Recorremos cada pin y lo ponemos
como entrada valorAnterior[i]=1; // Inicializamos el valor
anterior a 1 (no pulsado) }

}

void loop(){

    for(int i=0; i<numBotones; i++){
valorLeido[i]=digitalRead(i+2); // Leer valor // Solo enviamos
cuando ha existido un cambio, para no sobrecargar el puerto serial
if((valorLeido[i]==0) && (valorLeido[i]!=valorAnterior[i])){
MIDI.sendNoteOn(48+i, 127, 1); //48 corresponde a C es decir D0.

} else if ((valorLeido[i]==1) &&
(valorLeido[i]!=valorAnterior[i]))

{ MIDI.sendNoteOff(48+i, 0, 1); }
valorAnterior[i]=valorLeido[i]; // El valor anterior es el nuevo valor
para la siguiente iteracion }

}
```

Se puede apreciar que el código es muy parecido al anterior, solo que cuando se detecta botón pulsado/liberado se envía un mensaje de nota activada o desactivada. El valor de nota enviado por el primer botón es 48 que según el protocolo MIDI corresponde a DO. Por lo tanto el siguiente botón enviará DO Sostenido, el siguiente RE,...

Ahora ya no tiene sentido visualizar el puerto serial, ya que veremos caracteres sin aparente sentido. Lo que tenemos que hacer ahora es hacer que nuestro Arduino sea reconocido como un dispositivo MIDI, para lo que **deberemos reflashear el Arduino con un nuevo firmware**.

3.4. DIAGRAMA DE BLOQUES FASE 2 – A completar

Introduce aquí el diagrama de bloques para que explique el funcionamiento del código

3.5. RECOMENDACIONES PARA LA FASE 2

Para este montaje te debería valer al menos el montaje del semáforo. Queremos que funcione el código del semáforo o el del piano en la misma placa hardware.

Si lo presentas sólo en el simulador, no es necesario que se cumpla este requisito.

NOTA: ANTES DE BAJARLO A PLACA - Te recomendamos que primero lo hagas con el Simulador antes de hacerlo en la placa real.

3.6. SIMULACIÓN CON MIDI– FASE 2

Introduce aquí el enlace del vídeo o fotos de la simulación.

3.7. CÓDIGO FINAL DE LA FASE 1, 2 Y ENTREGA

Para que la entrega esté aprobada hay que mostrar el funcionamiento siguiendo las mismas instrucciones que te hemos planteado al profesor en la fecha establecida por el mismo (VER NORMATIVA DE PRÁCTICAS – campus virtual).

La FASE 1 se puede mostrar al profesor en el simulador. Mientras que la FASE 2 es optativa y tendrás que adquirir tu propia placa (+2 puntos).

A parte, se habilitará una entrega en el campus virtual de los resultados de este cuadernillo en el que tienes que respetar la fecha de entrega de lo siguiente.

La entrega se debe realizar en el MISMO ZIP (NumeroGrupoX-sed.zip) que empezaste para el Semáforo añadiendo lo siguiente:

- Contenido: Debes agregar en el fichero .docx que tenga las siguientes secciones:
 - o Sección 1- PIANO: La primera página con el nombre y apellidos de los miembros del grupo.

- Sección 2- PIANO: Circuito electrónico tipo Figura 1 de tu sistema de la Fase 1 y Fase 2 junto con la imagen o fotografía de tu circuito (a ser posible hecha desde el 123dcircuits).
- **Sección 3- PIANO: Incluir los bloques a completar.**
- **(NOTA: SU ENTREGA SE HABILITARÁ DESPUÉS DE QUE FINALICE LA SESIÓN 2 DEL LABORATORIO).**

Parte 1: Piano B

4. Piano electrónico

4.1. NOMBRE Y DESCRIPCIÓN DEL SISTEMA: Piano Electrónico

Desarrollaremos un piano electrónico con diferentes elementos hardware. Este cuadernillo se rellena de manera diferente al resto. Tienes que completar los apartados que te dice: “A completar”.

Este ejemplo servirá para que veas cómo se realizan funciones y llamadas a Funciones desde Arduino. Fíjate en el código de ejemplo que viene en la sección 4.4

4.2. MATERIAL NECESARIO

- Protoboard
- Placa Arduino
- 5 botones
- Potenciómetro
- Sensor de luz
- Buzzer
- Resistencia
- Led
- Cable telefónico

4.3. CONEXIONES

Lo primero es **conectar los 5 botones**.

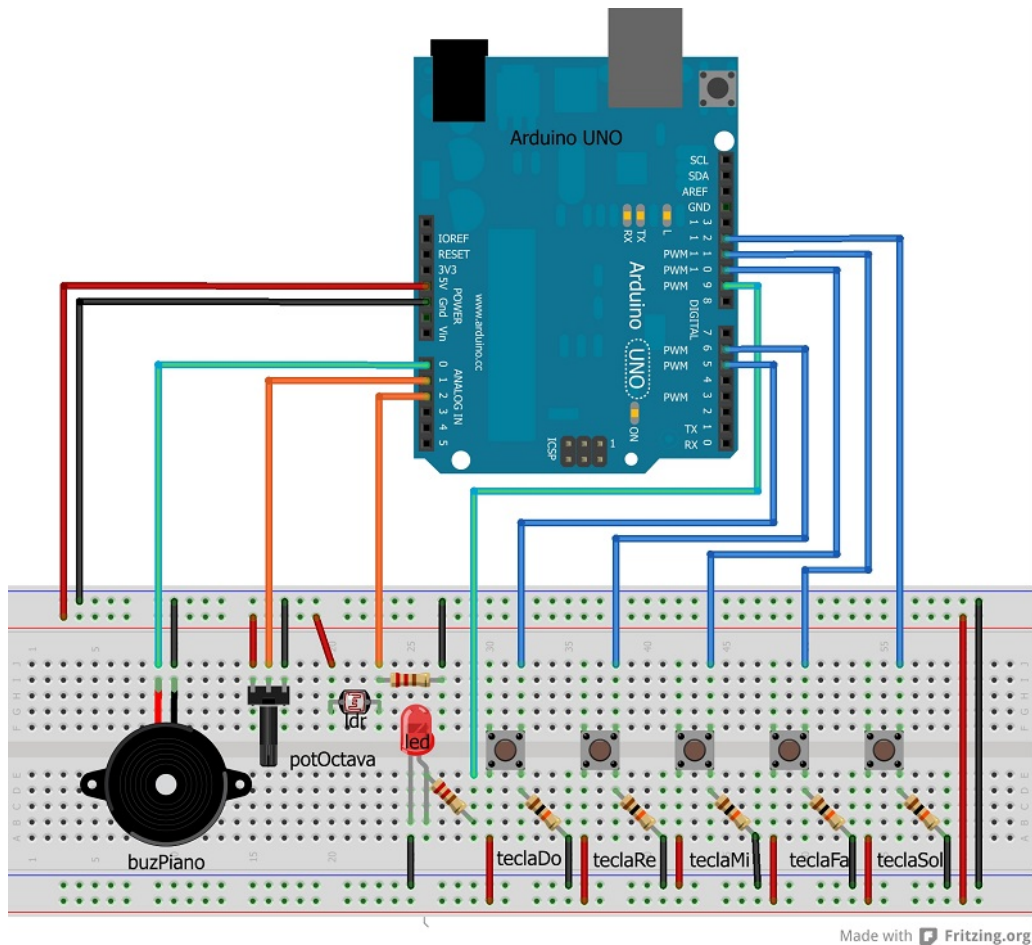
Como podéis ver en la siguiente figura la conexión es muy sencilla, conectamos cada botón entre GND y el pin correspondiente (del 2 al 12).

El objetivo es implementar un mini-piano con cinco botones (como si fueran las teclas), correspondientes a las teclas do, re, mi, fa y sol.

El potenciómetro permitirá subir la frecuencia de los sonidos para cambiar de octava.

Además, se incluirá un sensor que encienda un led para iluminar al piano cuando no haya luz.

Comprueba las conexiones de la siguiente figura, realiza las conexiones y procede a subir el código de la siguiente subsección.



4.4. CÓDIGO DE VERIFICACIÓN – FASE 3

Escribe el siguiente código en un fichero de Arduino que se llame mipiano.ino

Debes copiar las librerías Boton que te dejamos a tu disposición. Tabula el código, no está tabulado del todo. Para ello, utiliza la opción de insertar librería de Arduino.

Completa el código que falta marcado con **XXX**

```
#include <Boton.h>
```

```
/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
```

```
/*XXXXXXXXXXXX PIANO XXXXXXXXXXXXXXX*/
```

```
/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
```

```
/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
```

```
/** NOTAS MUSICALES **/
```

```
#define SILENCIO 0
```

```
#define NOTA_DO 32
```

```

#define NOTA_RE 37
#define NOTA_MI 41
#define NOTA_FA 44
#define NOTA_SOL 49
/** PARÁMETROS **/
//Establece el tiempo de separación entre notas para que no suenen seguidas.

#define DURACION_SEPARACION 10
/** DATOS DEL PROGRAMA **/
//PINES DE ENTRADA ANALÓGICOS

int ldr = A2;
int potOctava = A1;

//TECLAS DEL PIANO

Boton teclaDo(XXX);
Boton teclaRe(XXX);
Boton teclaMi(XXX);
Boton teclaFa(XXX);
Boton teclaSol(XXX);

//PINES DE SALIDA

int led = 9;
int buzPiano = 14;

//OTRAS VARIABLES
int octava = 1;

void setup() {
  pinMode(led, OUTPUT);
  pinMode(buzPiano, OUTPUT);
}

void loop() {

  tocaNota();
  estableceOctava();
  compruebaLuz();
}

/**
 * Comprueba si hay suficiente luz y enciende
 * los leds proporcionalmente a la luz.
 */

void compruebaLuz(){
  int luz = analogRead(ldr);
  Serial.println(luz);
  if(luz > 255){
    analogWrite(led, 0);
  }
}

```

```

}else{
analogWrite(led, 255-luz);
}
}

/**
 * Establece la octava en función de
 * lo que marque el potenciómetro.
 */

void estableceOctava(){

int potencia = analogRead(potOctava);
octava = map(potencia, 0, 1023, 1, 8);

}

/**
 * Si hay algún pulsador activo, toca la nota que le corresponde.
 */

void tocaNota(){

if(teclaDo.leerBoton() == HIGH){
    tone(buzPiano, NOTA_DO*octava);
}
else if(teclaRe.leerBoton() == HIGH){
    tone(buzPiano, NOTA_RE*octava);
}
else if(teclaMi.leerBoton() == HIGH){
    tone(buzPiano, NOTA_MI*octava);
}
else if(teclaFa.leerBoton() == HIGH){
    tone(buzPiano, NOTA_FA*octava);
}
else if(teclaSol.leerBoton() == HIGH){
    tone(buzPiano, NOTA_SOL*octava);
}
else{
    noTone(buzPiano);
}

}

}

```

4.5. DIAGRAMA DE FLUJO – FASE 3 – A completar

Realiza el diagrama de flujo del código.

4.6. SIMULACIÓN – FASE 3 – A completar

Realiza la simulación con el simulador tinkercard

4.7. CÓDIGO FINAL DE LA FASE 3 Y ENTREGA

Para **que la entrega esté aprobada** hay que mostrar el funcionamiento siguiendo las mismas instrucciones que te hemos planteado al profesor en la fecha establecida por el mismo (VER NORMATIVA DE PRÁCTICAS – campus virtual).

La FASE 3 se puede mostrar al profesor en el simulador (máximo 1 punto). Si llegas a esta fase en clase, pide al profesor que te preste el material para montarlo en hardware. De ser así conseguirás 1 punto adicional, y por tanto, los 2 puntos máximos de la Fase 3.

A parte, se habilitará una entrega en el campus virtual de los resultados de este cuadernillo en el que tienes que respetar la fecha de entrega de lo siguiente.

La entrega se debe realizar en el MISMO ZIP (NumeroGrupoX-sed.zip) que empezaste para el Semáforo añadiendo lo siguiente:

- Contenido: Debes agregar en el fichero .docx que tenga las siguientes secciones:
 - Sección 1- PIANO: La primera página con el nombre y apellidos de los miembros del grupo.
 - Sección 2- PIANO: Circuito electrónico tipo Figura 1 de tu sistema de la Fase 1 y Fase 2 junto con la imagen o fotografía de tu circuito (a ser posible hecha desde el 123dcircuits).
 - **Sección 3- PIANO: Incluir los bloques a completar.**
 - **(NOTA: SU ENTREGA SE HABILITARÁ DESPUÉS DE QUE FINALICE LA SESIÓN 2 DEL LABORATORIO).**

ANEXO I: DOCUMENTACIÓN REFLASHEAR

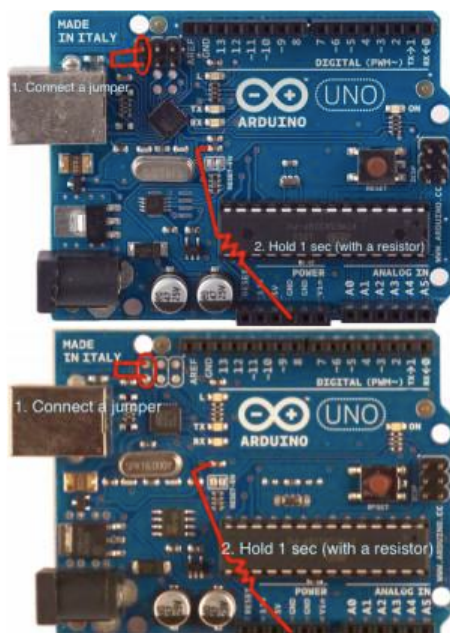
Reflasheando el Arduino con Hiduino. (Desde Linux, en Mac es parecido).

Lo primero que deberemos hacer es **descargarnos el firmware de Hiduino**. Vayamos a la carpeta de "Compiled Firmware" y localicemos el archivo que nos interesa "**HIDUINO_MIDI.hex**". Pon ese archivo en el escritorio por ejemplo, es simplemente por tener una ruta mas fácil para después.

Ahora nos tenemos que bajar el "**dfu-programmer**". Si estamos en Linux:
sudo apt-get install dfu-programmer

A continuación necesitaremos que nuestro Arduino entre en modo DFU para que se nos permita introducirle el código. Antes de nada es muy **importante que hayáis cargado el código anterior** (el del MIDI). 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:

- Mira tu versión: La de un condensador o la de dos (**ver fotos**)
- **Conecta un jumper** entre los terminales que se ven en la figura
- **Toca con la resistencia** entre GND y el pad izquierdo del condensador
- Deberás ver como una luz parpadea y se apaga, retira las conexiones anteriores.
- Estás en modo DFU.



Si todo ha salido correctamente y dejasteis el **HIDUINO_MIDI.hex** en el **Escritorio**, deberías de poder hacer lo siguiente. Repito, desde **consola Linux**:

```
cd \Desktop
```

```
sudo dfu-programmer at90usb82 erase
```

```
sudo dfu-programmer at90usb82 flash HIDUINO_MIDI.hex
```

```
sudo dfu-programmer at90usb82 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.

ANEXO II: LED RGB

En este anexo te ponemos un ejemplo para usar un LED RGB. En el kit de prácticas no dispones de uno, pero puedes adquirirlo en cualquier tienda de electrónica o utilizar el simulador.

5.1. NOMBRE DEL SISTEMA

Led RGB.

En este último capítulo, se muestra un sencillo ejemplo para utilizar Led RGB.

5.2. DISEÑO DEL SISTEMA

Se basa en un led RGB, capaz de reproducir diferentes colores en una escala cromática de 0 a 256. El sistema consiste en la alternación de diferentes colores de forma continua.

5.3. PSEUDOCÓDIGO

VARIABLES

Entero pinLed, pinLDR, valorLDR.

FUNCIÓN CONFIGURACIÓN

Función setup

Inicio

Pin digital pinLed ← OUTPUT;

Espacio Serial ← 9600;

Fin

FUNCIÓN CÍCLICA

Función loop

Inicio

pinLed ← LOW;

valorLDR ← leer pinLDR;

Imprimir pantalla ← valorLDR;

Si valor > 256 entonces

pinled ← HIGH;

Si no

pinled ← LOW;

Fin si;

espera (200);

Fin

4. CÓDIGO FINAL

A continuación, te mostramos un ejemplo que puedes usar para probar este tipo de Leds. Estos Leds no los tenemos en el kit de las prácticas pero puedes adquirirlo en cualquier tienda de electrónica y en Amazon.

```
sketch_apr26a $
//Primero asignamos las distintas patas del led RGB (red-green-blue) a los pines
//de la placa 9, 10 y 11. La pata restante se conecta a tierra (gnd).

int ledRojo=9;
int ledVerde=10;
int ledAzul=11;

void setup() {

// Establecemos las patas de los leds conectados a los pines como
//salidas.

  pinMode(ledRojo,OUTPUT);
  pinMode(ledVerde,OUTPUT);
  pinMode(ledAzul,OUTPUT);

}

void loop() {

//Llamamos a la funcion color, y asignamos valores a la intensidad
//de cada uno de los principales (RGB) para obtener mezclas:

  color(255,0,0); //ROJO (principal)
  delay(1000);
```

```
color(0,255,0); //VERDE (principal)
delay(1000);

color(0,0,255); //AZUL (principal)
delay(1000);

color(255,255,255); //BLANCO (rojo+verde+azul)
delay(1000);

color(255,0,255); //MAGENTA (rojo+azul)
delay(1000);

color(255,128,0); //NARANJA (rojo+un poco de verde)
delay(1000);

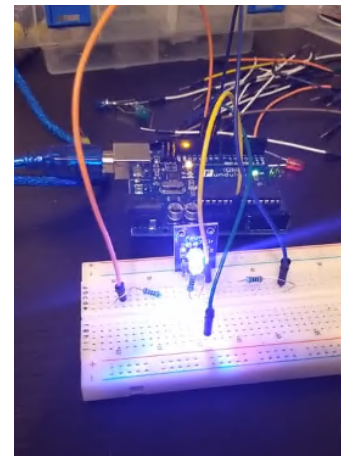
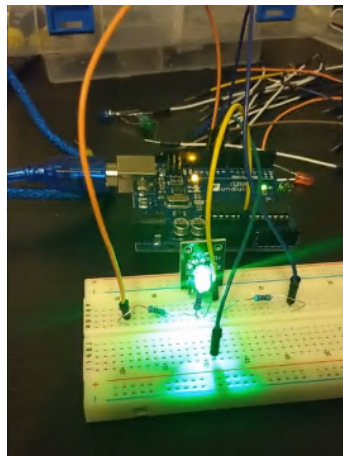
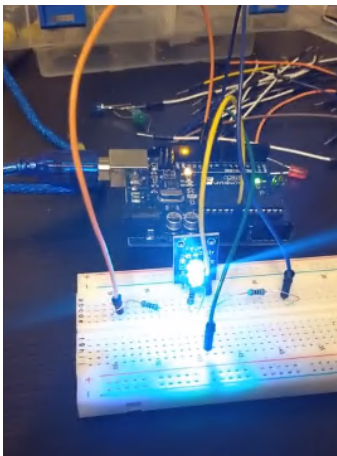
}

void color(int rojo, int verde, int azul) {

    //Funcion que manda la señal de cada uno de los colores RGB

    analogWrite(ledRojo,255-rojo);
    analogWrite(ledVerde,255-verde);
    analogWrite(ledAzul,255-azul);
}
```

5.5. FOTOS DEL SISTEMA FINAL



5.6. VÍDEO EJEMPLO DEL SISTEMA FINAL

<https://www.youtube.com/watch?v=p8zsrkCPoQA>

Nota: Para ver el vídeo se pone el cursor encima del enlace y se le da a Ctrl+Clic (clic izquierdo).

