Sintetizadores

Bricolaje MIDI con Arduino

por emiliomm | 10/09/2013

¿Te gustaría crear un dispositivo MIDI original y creativo? En este tutorial te enseñaremos cómo hacerlo con Arduino. El tutorial primero ofrece una visión general de Arduino y cómo se puede aplicar al mundo musical. Posteriormente, explica cómo crear (1) un dispositivo capaz de enviar diferentes tipos de mensajes MIDI, y (2) un dispositivo capaz de recibir mensajes MIDI. Por último, se plantea un ejercicio para combinar todo lo aprendido y se proponen ideas para inspirar a aquellos valientes que decidan construirse su propia locura MIDI.

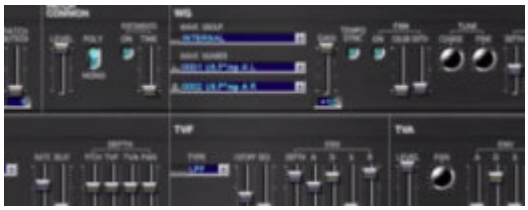
0. Sobre este tutorial...

Este tutorial ha sido elaborado por Emilio Molina bajo la supervisión de Ana María Barbancho (profesora responsable) como práctica de la asignatura *'Laboratorio de Equipos de Audio'*, de la titulación de *Grado en Ingeniería*

ARTÍCULOS RELACIONADOS

MIDI y latencia: da más solidez a tus bases

Control mediante NRPNs en MIDI



SysEx (I): el MIDI más 'SexSy'

MIDI: cambios de programa y

creciente, y para completarlo es conveniente tener unos conocimientos mínimos de electrónica y de programación.

1. ¿Qué es Arduino?

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares. Las placas se pueden adquirir en su página web (o montarlas a mano...) y el entorno de desarrollo integrado libre se puede descargar gratuitamente. Al ser open-hardware, tanto su diseño como su distribución es libre; es decir, puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin haber adquirido ninguna licencia. En consecuencia, es una forma muy sencilla de introducirse en el mundo de la electrónica, con una amplia documentación en Internet, y no es necesario tener grandes conocimientos de electrónica para crear cosas sorprendentes.

La página web oficial de Arduino es http://arduino.cc/ (la versión española es http://arduino.cc/es, pero es menos completa), y en ella puede encontrarse el software necesario, tutoriales y documentación, así como un enlace a la Arduino Store (http://store.arduino.cc) donde se puede encontrar un gran surtido de placas (los precios rondan los 30-60€). En este tutorial, nos centraremos en la placa Arduino Uno, porque dispone de todas las características

MIDI Local Control on/off y los modos MIDI (mode change)

WiFi MIDI, realidad y espejismo

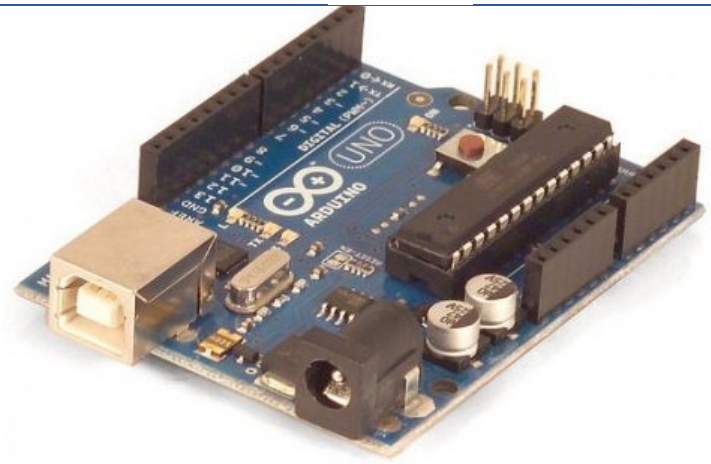


Figura 1 - Arduino Uno (20€ aproximadamente)

2. MIDI y Arduino

Arduino Uno está basado en el microcontrolador ATmega328, el cual funciona a 16Mhz, tiene una memoria flash de tan sólo 32KB (lo justo para almacenar el programa a ejecutar) y una memoria SRAM de 2KB. Estas bajas prestaciones lo limita para procesar audio de alta calidad, pero lo hace muy apropiado para trabajar con mensajes MIDI o para sintetizar sonidos de baja calidad. No obstante, los modelos más recientes de Arduino (como Arduino Yun) poseen mejores prestaciones y merecería la pena seguir indagando en sus posibilidades para aplicarlas a la música. En este tutorial nos centraremos sólo en el manejo de mensajes MIDI en Arduino Uno (y superiores), ya que la síntesis de audio es un tema diferente y daría para un tutorial mucho más largo que este (ejemplo). Para comprender mejor qué es el MIDI, recomiendo este fantástico tutorial posteado en Hispasonic.

interfaz MIDI con sintetizadores u otro tipo de controladores MIDI. Este tipo de elementos se pueden encontrar en tiendas de electrónica, o se pueden encargar por Internet.

Figura 2 - Enfoque típico para elaborar dispositivos MIDI en Arduino.

3. Instalación de Arduino Uno y ejecución del ejemplo 'Blink.ino'

La instalación de Arduino es bastante sencilla: es necesario instalar el entorno de desarrollo y configurar los drivers de la placa. En la versión española de la web oficial, se puede encontrar las instrucciones detalladas para instalar Arduino: http://arduino.cc/es/Guide/Windows

/Windows.

Siguiéndolas, ya deberíamos tener **Arduino funcionando**. Ejecuta el IDE de Arduino, selecciona el tipo de tarjeta (Arduino UNO) y el puerto (en mi caso COM4) en el menú Herramientas. Abre el ejemplo 'Blink.ino', en Archivo>Ejemplos>01.Basics>Blink y cargalo en la placa mediante el botón Cargar. Si el LED integrado en la placa del pin 13 parpadea, tu Arduino está correctamente programado.

Nota: Una vez que Arduino ya se ha programado a través del puerto USB, el PC es prescindible. En nuestro caso el USB lo utilizamos para alimentar la placa, pero con un alimentador externo de entre 7 y 12V, la placa funcionaría de forma autónoma igualmente.

Para aprender a conectar LEDs, botones, potenciómetros... **Muy recomendado**: http://arduino.cc/es/Tutorial/HomePage

4. Cableando un conector MIDI de salida y ejecución del ejemplo 'MIDI.ino'

En este apartado vamos a construir un mini-secuenciador MIDI y lo vamos a conectar a un sintetizador, o a la interfaz MIDI de nuestro ordenador para escuchar las notas generadas. La conexión de un conector MIDI a Arduino es muy simple, y el único material necesario es una resistencia de 220Ω, varios cablecitos y un soldador. El conector MIDI de salida se conecta a Arduino tal y como se muestra en la siguiente

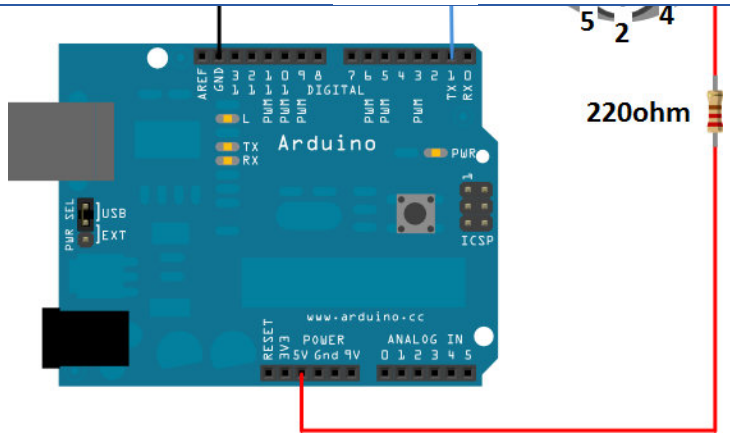


Figura 3 - Cableado de un conector MIDI OUT en Ardunio

En mi caso, la resistencia la he soldado en el cable directamente y funciona muy bien. Este esquema ha sido extraído de: http://arduino.cc/es/Tutorial/Midi

¡IMPORTANTE!

Es muy importante conectar los pines tal y como se muestra en la figura. Por si existe alguna duda de cuáles son los pines 5, 2, y 4 en un conector real, véase la siguiente foto:

Figura 4 - ¡Cuidado con la numeración de los pines!

Si estas conexiones no se realizan correctamente, nuestro Arduino no emitirá

forma, acabamos de construir un MIDI OUT para el Arduino. Con un cable MIDI estándar:

Figura 5 - Cable MIDI estándar

Conectamos el MIDI OUT del Arduino al MIDI IN de una interfaz MIDI para ordenador o a un sintetizador. A continuación, ejecuta el ejemplo Archivo>Ejemplos>04.Communication>MIDI y cárgalo. Este ejemplo envía secuencialmente una escala cromática ascendente. Para escuchar estos mensajes MIDI, es necesario algún tipo de instrumento virtual que actúe de sintetizador.

5. Extra: Análisis de mensajes MIDI con la herramienta MIDI-OX

MIDI-OX es una herramienta muy útil para monitorizar los mensajes MIDI que entran y salen del PC. Es gratuita y permite analizar con detalle la información que entra y sale por los puertos MIDI asociados a nuestro PC. Puedes descargarla de http://www.midiox.com/ y utilizarla para depurar tus programas de

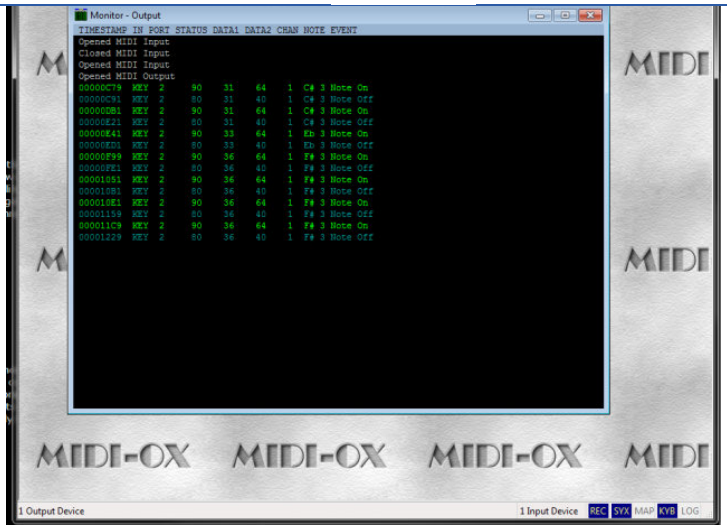


Figura 6 - Captura de pantalla de MIDI-OX

En nuestro caso, abriremos el Monitor - Input (View>Input Monitor). En el Monitor - Input veremos todos los campos de los mensajes MIDI entrantes.

Nota: Puede ser necesario configurar los puertos MIDI de entrada y salida en Options>Midi devices para seleccionar los de la interfaz MIDI que estemos usando.

6. Elaboración de diferentes tipos de mensajes MIDI

6.1. Note on- Note off

1000nnnn	0kkkkkkk 0vvvvvvv	Note Off event. This message is sent when a note is released (ended). (kkkkkkk) is the key (note) number. (vvvvvvv) is the velocity.
1001nnnn	0kkkkkkk 0vvvvvvv	Note On event. This message is sent when a note is depressed (start). (kkkkkkk) is the key (note) number. (vvvvvvv) is the velocity.

Estos mensajes constan de 3 bytes, tal y como se indica en la tabla superior. Vamos a elaborar dos funciones llamadas noteOn y noteOff, encargadas de enviar estos mensajes. Hemos

```
void noteOn(int channel, int note, int velocity)
{
// channel comprendido entre 1 y 16.
Serial.write(B10010000+channel-1);
Serial.write(note);
Serial.write(velocity);
}
```

Código 1 - Función alternativa a la ofrecida en 'MIDI.ino' para generar NoteOn. El uso de los canales se ha facilitado y están comprendidos entre 1 y 16.

Nota: En Arduino, un número binario lleva el prefijo B (válido para un máximo de 8 bits), y un número hexadecimal el prefijo 0x, e.g. 15 = B1111 = 0xFF.

```
void noteOff(int channel, int note) {
Serial.write(B10000000+channel-1);
Serial.write(note);
Serial.write(0);
}
```

Código 2 - Función para generar un NoteOff según indica el protocolo MIDI.

Nota: El protocolo MIDI contempla dos formas de hacer un noteOff. La primera utiliza el mensaje Note Off que comienza por B1000nnnn. La segunda utiliza el mensaje Note

Modifica ahora el ejemplo MIDI que se utilizo anteriormente y envía algunas notas utilizando estas funciones. Por ejemplo, puedes hacer uso de diferentes canales.

6.2. Control change

1011nnnn	0ccccccc 0vvvvvvv	Control Change. This message is sent when a controller value changes. Controllers include devices such as pedals and levers. Controller numbers 120-127 are reserved as "Channel Mode Messages" (below). (ccccccc) is the controller number (0-119). (vvvvvvv) is the controller value (0-127).

Cualquier sintetizador tiene numerosos parámetros que pueden ser controlados vía MIDI. Un parámetro puede asignarse a cualquier número de control MIDI (0ccccccc), aunque existen algunos valores que siempre están asociados a características concretas del instrumento. Algunos controles comunes son:

- Modulation (CC #1)
- Breath (CC #2)
- Foot Pedal (CC #4)
- Portamento Time (CC #5)
- Volume (CC #7)
- Pan (CC #10)
- Expression (CC #11)
- Sustain Pedal (CC #64)
- Soft Pedal (CC #68)

En este ejemplo, vamos a modificar algunos controles del instrumento virtual o sintetizador que estemos usando. Vamos a hacer una función que envíe un cambio de control y comprobaremos qué sucede al asignarle distintos tipos de controles.

```
void controlChange(int channel, int control,
```

```
Serial.write(control);
Serial.write(value);
}
```

Código 3 - Función para generar un mensaje de cambio de parámetro de control

Ejercicio: Haz un programa que modifique el estado del pedal de sustain cada 1000ms. Comprueba que el estado del pedal de sustain va variando. Recuerda que el control correspondiente al pedal de sustain es el CC #64.

Control Number (2nd Byte Value)			Control Function	3rd Byte Value	
Decimal	Binary	Hex		Value	Used As
64	01000000	40	Damper Pedal on/off (Sustain)	≤63 off, ≥64 on	---

6.3. Pitch wheel change

1110nnnn	0lllllll 0mmmmmmm	Pitch Wheel Change. 0mmmmmmm This message is sent to indicate a change in the pitch wheel. The pitch wheel is measured by a fourteen bit value. Center (no pitch change) is 2000H. Sensitivity is a function of the transmitter. (lllllll) are the least significant 7 bits. (mmmmmm) are the most significant 7 bits.

El protocolo MIDI contempla la posibilidad de 'desafinar' las notas para simular técnicas como el vibrato, el bending o el glissando. Para ello, es posible enviar un mensaje de "cambio de rueda de tono" (muy común en los controladores MIDI). Vamos a implementar una función que envíe este tipo de parámetros:

```
void PitchWheelChange(int channel, int value)
{
// value: [-8192, 8191] (14 bits)
```

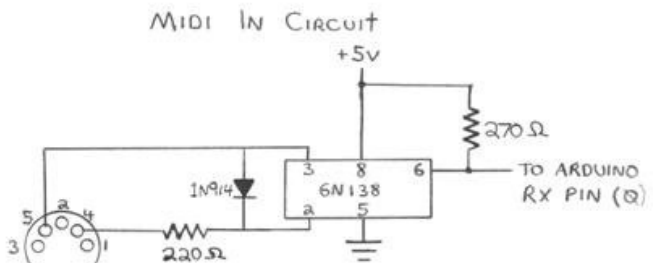
```
unsigned char high = (change >> 7) & 0x7F; // High 7 bits

Serial.write(B11100000+channel-1);

Serial.write(low);

Serial.write(high);

}
```

Código 4 - Función que genera un mensaje de cambio de rueda de pitch. Observa cómo hay que dividir la variable 'value' en byte bajo y byte alto para enviarlo por el puerto serie.

Puedes hacer algunas pruebas enviando diferentes valores de pitch bending y comprobar de qué forma afecta al resultado del sonido sintetizado.

7. Cableado del conector MIDI de entrada

Aunque la configuración de MIDI OUT también funciona como MIDI IN, utilizaremos una configuración algo más compleja para evitar daños por exceso de corriente. Esta configuración es la recomendada en la mayoría de los tutoriales. Por lo tanto, haremos uso de un optoacoplador para proteger al dispositivo:



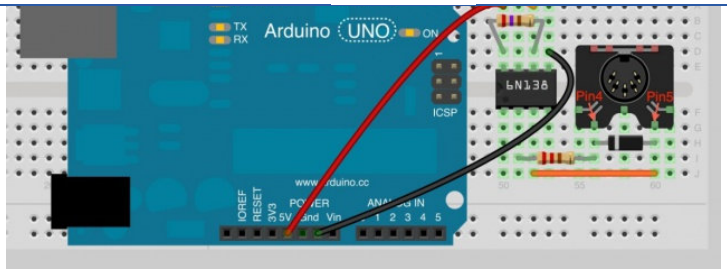
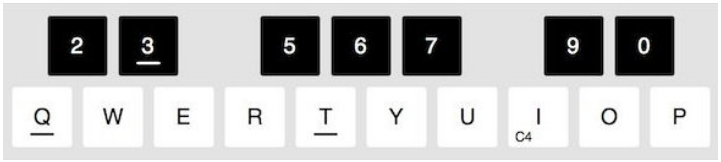


Figura 7 - Esquema y montaje de un conector MIDI de entrada con optoacoplador para evitar daños.

Nota: El diodo común de pequeña señal 1N4148 podría sustituir al diodo 1N914.

Envío de mensajes MIDI con la herramienta MIDI-OX

Si no se dispone de un controlador MIDI externo, la herramienta MIDI-OX puede ser útil para enviar mensajes MIDI a la placa Arduino. Activa la opción Actions>Keyboard, y el teclado del ordenador pasará a funcionar como un teclado MIDI según la distribución siguiente:



Nota: La tecla Q, por defecto, envía la nota MIDI C4 (número 60).

8. Gestión en Arduino de mensajes MIDI entrantes

Esta sección del tutorial creo que es la más valiosa y la que más horas de trabajo tiene, ya que la gestión de mensajes MIDI no es fácil y cuesta mucho trabajo encontrar documentación para realizarla correctamente. Para recibir los

Serial.available(): Devuelve el número de bytes (caracteres) disponibles para ser leidos por el puerto serie. Se refiere a datos ya recibidos y disponibles en el buffer de recepción del puerto (que tiene una capacidad de 128 bytes).

· **Serial.read():** Devuelve el primer byte disponible recibido por el puerto serie (devuelve -1 si no hay datos disponibles).

Ninguna de estas funciones es bloqueante, por lo que es necesario usarlas correctamente para asegurarse que se leen los mensajes MIDI sólo cuando ya han sido recibidos. A continuación se presenta la implementación de una función que devuelve los parámetros de un mensaje MIDI entrante de 3 bytes. Si se detecta el primer byte de un mensaje MIDI válido (byte de status), la función espera hasta recibir el mensaje MIDI completo, en caso contrario no es bloqueante.

```
void readMIDI(byte* channel ,byte* type,
byte* data1, byte* data2)

{

int aux;

while (Serial.available()>0){

aux=Serial.read();

if ((aux >= B10000000)&&(aux <=
B11101111)){

// Si es byte STATUS legal:

*channel=(aux&B00001111)+1; //canales
entre 1 y 16
```

```
// los Serial.read() devolverán -1:
while (Serial.available() < 2){
}
*data1 = Serial.read();
*data2 = Serial.read();
break; // Sale porque ha leído un mensaje MIDI correcto
}
}
}
```

Código 5 - Función para leer mensajes MIDI haciendo uso de la gestión de buffers para evitar leer mensajes incompletos.

Nota: Se han definido los parámetros de la función como punteros para pasar los parámetros por referencia (parámetros de entrada/salida).

A continuación se muestra un **programa completo** que lee un mensaje MIDI, y si es un NoteOn en el canal 1 correspondiente a la nota C3 (nota 60, tecla Q en MIDI-OX) se enciende el LED integrado del pin 13 durante 1 segundo.

¡¡ Importante !! : Es imprescindible desconectar totalmente los pines 0 (RX) y 1 (TX) mientras se carga el programa a la placa Arduino, puesto que la programación se realiza a través del mismo puerto serie conectado al USB.

```
*/
byte type;
byte channel;
byte data1;
byte data2;
void setup() {
// Set MIDI baud rate:
Serial.begin(31250);
pinMode(13,OUTPUT);
}
void loop() {
channel=0;
type=0;
data1=0;
data2=0;
readMIDI(&channel,&type, &data1,&data2);
//type 0x90: Note On
//data1 60: C4 (tecla Q en MIDI-OX)
if ((channel==1)&&(type==0x90)&&
(data1==60)){
digitalWrite(13,HIGH);
delay(1000);
}
else
{
```

```
}

void readMIDI(byte* channel ,byte* type,
byte* data1, byte* data2)

{

int aux;

while (Serial.available()>0){

aux=Serial.read();

if ((aux >= B10000000)&&(aux <=
B11101111)){

// Si es byte STATUS legal:

*channel=(aux&B00001111)+1; //canales
entre 1 y 16

*type=(aux&B11110000);

// Esperamos porque si aún no han llegado
los datos,

// los Serial.read() devolverán -1:

while (Serial.available() < 2){

}

*data1 = Serial.read();

*data2 = Serial.read();

break; // Sale porque ha leído un mensaje
MIDI correcto

}

}

}
```

Código 6 - Programa completo para leer

En este apartado se propone un ejercicio que utiliza simultáneamente el MIDI IN y el MIDI OUT que hemos construido sobre nuestro Arduino.

Importante: Si estás usando MIDI-OX, comprueba no está realizando un eco entre la entrada y la salida MIDI de tu interfaz MIDI en View>Port routings... Si existiese una conexión entre la entrada y la salida MIDI de tu interfaz, con el botón derecho sobre ella puede eliminarse para evitar realimentaciones indeseadas.

Se propone la elaboración de un programa en Arduino que funcione como un MIDI-thru selectivo. El objetivo es que los mensajes entrantes por nuestro MIDI-IN se vuelvan a enviar por el MIDI-OUT si cumplen alguna condición (por ejemplo, se pueden ignorar los Note Off para simular un pedal de sustain).

Si el programa funciona correctamente, se podrán enviar mensajes con el QWERTY en MIDI-OX y escucharlos en el sintetizador que estemos usando.

Nota: Recuerda que los canales en el programa los estamos manejando entre 1 y 16, pero al enviarlos con Serial.write() es necesario restarle uno, para que estén comprendidos entre 0 y 15.

Sugerencias para un 'MIDI hack' en Arduino

Un 'MIDI hack' es algún invento creativo que combina MIDI con otro tipo de elementos.

Algunas ideas:

- Construcción de un controlador MIDI con botones y potenciómetros.
- Iluminar una serie de LEDs en función de los mensajes MIDI recibidos desde un teclado controlador externo.
- Utilizar las salidas PWM para sintetizar los mensajes MIDI entrantes y escucharlos a través de un altavoz (con su correspondiente amplificador).
- Programación de un arpegiador que recibe una serie de notas por el MIDI IN y envía una secuencia a un tempo determinado por el MIDI OUT.
- Programación de un secuenciador programable a partir de botones, luces y potenciómetros (ver http://monome.org/).
- Utilizar piezoeléctricos pegados sobre el cuerpo para detectar golpes sobre ellos y enviar mensajes MIDI. Esto puede funcionar como controlador MIDI de percusión corporal.

COMENTARIOS

1 2

#1 por **insula** el 10/09/2013

fantastico tutorial...

1

#2 por **Mordus** el 10/09/2013

Gracias!
Muy interesante 

#3 por **Man with the x-ray eyes** el 10/09/2013

Gracias por compartir el tutorial.
Por mi parte, aporto un par de ejemplos completos que colgué en hispa en su día de cómo hacer controladores midi con arduino: teclado controlador y pads de percusión:
https://www.hispasonic.com/foros/piano-midi-arduino/419226
https://www.hispasonic.com/foros/paduino-01-pads-midi-basados-arduino/414667
Saludos!

1

#4 por **Guti** el 10/09/2013

Me encanta! gracias a Hispasonic.

#5 por **pablofcid** el 10/09/2013

Estupendo que nos hayas puesto tan a primera vista la sencillez y posibilidades de arduino para MIDI (y otras cosas). Aplaudo tu artículo y va mi +1.

Pero un par de comentarios (aunque he hecho una lectura rápida). Me parece que tu programación no tiene en cuenta algunas cosas importantes como el running status, o los mensajes real time y system exclusive, por ejemplo.
Para poder funcionar en un entorno real de sintes, secuenciadores y equipos MIDI sería imprescindible que gestione sin llevar a situaciones de error cualquier tipo de mensaje. Es sencillo (la norma MIDI afortunadamente se pensó para que fuera factible un hardware muy simple, pero exige programar una pequeña máquina de estados, al menos para llevar el control de running status.
O también para tolerar correctamente mensajes pitch bend con sólo el MSB. El aftertouch mono tampoco usa 3 bytes en el mensaje,...
Como digo son detalles que más allá de un ejemplo de desarrollo para laboratorio habría que tener en cuenta, pero que no suponen realmente complicar gran cosa (sí complicarían el artículo, así que bienvenido como está, pero al menos queda la advertencia).

2

la manera de hacerme un secuenciador midi de 8-16 pasos basado en arduino

#7 por **SOHAM** el 10/09/2013

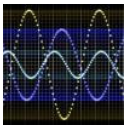
Me encantan las DIY aunque no se si tengo tiempo para meterme en algo así, las posibilidades que brinda (conociendo bien la norma MIDI que prácticamente desconozco) son infinitas y muy creativas, y el lenguaje parece C o similar no? (esto no sería un problema).

#8 por **semabr** el 10/09/2013

Aquí os dejo un tutorial de como hacer un controlador MIDI con arduino, pero más orientado a ableton o traktor (al estilo Midi Fighter). Espero qu a alguien le sea útil http://www.instructables.com/id/Arcade-Button-MIDI-Controller/

1

#9 por **emiliomm** el 10/09/2013



Gracias a todos! Pablofcid, es muy interesante lo que comentas. ¿Cómo harías un programa para recibir mensajes MIDI "avanzados"? ¿Mas sentencias "if"?

Otro tutorial para cacharrear.

Sea bienvenido.

#11 por **parker25** el 10/09/2013

muchisimas gracias

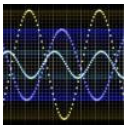
#12 por **pablofcid** el 10/09/2013

#9
Lo ideal es leerse el documento original de la primera norma MIDI. Es corto y sencillo de entender, y es 'compatible hacia adelante' (ya podían aprender otros estándares). Hay está muy claro el formato, longitudes, la cuestión del running status, etc. Básicamente yo soy enemigo de eso de leer mensajes enteros y prefiero analizar según llegan, byte a byte y con una pequeña máquina de estados multinivel. la UART resuelve la recepción de cada byte. una primera capa de tu soft puede resolver la cuestión de identificar cada mensaje MIDI. necesita recordar en qué punto estás (reconociendo el posible running status, haciendo excepción para permitir real time messages a itad de otros, todo ese tipo de cosas mejor que queden encapsuladas en una capa. otra capa analizaría ya mensajes completos.

Como curiosidad te diré que antes de que pudiera comprarme un primer entorno de desarrollo con micros,

realtime intercalado, sysex,...
Y lo bueno es que estaba hecho 'a pelo', mediante lógica. No tenía ni UART (mejor dicho, la implementé desde cero). Me refiero a que todo estaba hecho con puertas y biestables. Eso si, era en una CPLD, de manera que en lugar de soldaduras y cien integrados, estaba todo en un sólo chip de lógica programable.

#13 por **emiliomm** el 11/09/2013



Seria interesante disponer de un esquema de todos los estados y las condiciones de transición correspondiente a la norma MIDI completa

#14 por **roncaron** el 11/09/2013

Introducing 123D Circuits.io // https://vimeo.com/73973905

1

#15 por **--430038--** el 11/09/2013

a marcadores!, siempre me parecio una utopia el poder montarme mi propio sistema arduino, pero a medida que voy recabando informacion presiento estar mas cerca de dar el paso. muchas gracias por compartir.

información complementaria, de mi proyecto de una controladora midi hecha con arduino:http://jjchop.wordpress.com/

#17 por **Alberto Serrano** el 12/09/2013

Precisamente he estado montando un display y router MIDI para mi pedalera de guitarra durante este verano y ya lo tengo terminado y funcionando:

http://albertoserrano.fuckopfamily.net/wp-content/uploads/2013/09/101_5250-1024x768.jpg
http://albertoserrano.fuckopfamily.net/wp-content/uploads/2013/09/101_5252-1024x768.jpg

En cuanto lo tenga montado en la pedalera lo incluyo en mi blog: http://albertoserrano.fuckopfamily.net

#18 por **Alberto Serrano** el 12/09/2013

Y el circuito montado en su caja:

http://albertoserrano.fuckopfamily.net/wp-content/uploads/2013/09/101_5260-1024x768.jpg

http://albertoserrano.fuckopfamily.net/wp-content/uploads/2013/09/101_5262-1024x768.jpg

#19 por **sapristico** el 13/09/2013

1

#20 por **SOHAM** el 13/09/2013

#18 Alberto he mirado tu blog y vaya manitas estás hecho! vaya equipazo te has montado resistencia a resistencia.. por cuanto saldría montar un ampli como ese? tiene una pinta tremenda.

El router arduino midi no lo acabo de comprender, 1 In 2 Out con display?

#21 por **Lisboetas** el 13/09/2013

#19
+100000

Quiero tiempo...quiero otra vida entera para empezar desde cero con estas movidas

Edito: por cierto, vaya nivelungo que ha cogido Hispasonic...Yo trasteo por webs guiris con cierta frecuencia y NO HAY nada de este nivel.....bueno no se si en coreano o japones

#22 por **Alberto Serrano** el 13/09/2013

SOHAM, el plexi me salió por unos 800€ solo el material. Lleva una semana de trabajo aproximadamente montarlo.

También he montado un Bassman, que uso con el grupo, y un JCM800 de

nombre del preset lo uso para ruteal, es decir, con un cambio de programa en la entrada envío diferentes cambios de programa y en diferentes canales MIDI en la salida.

#23 por **SOHAM** el 16/09/2013

#22 Ostras! sale caro hasta montándoselo uno mismo.. jaja.

Ahora si entiendo el ruteo midi. Con vistas a que aplicación real la hiciste? Compatibilidad entre hardware midi?

#24 por **Gaston Betancor** el 16/09/2013

Hay una librería MIDI que funciona muy bien y es muy fácil de usar:

http://playground.arduino.cc/Main/MIDILibrary

1

#25 por **Alberto Serrano** el 17/09/2013

#23 Es para controlar varios aparatos con una sola pedalera MIDI.

1