



Notes and Volts

Electronics, Guitars, and Geekery

[Home](#)
[Steal My Music](#)
[Contact](#)

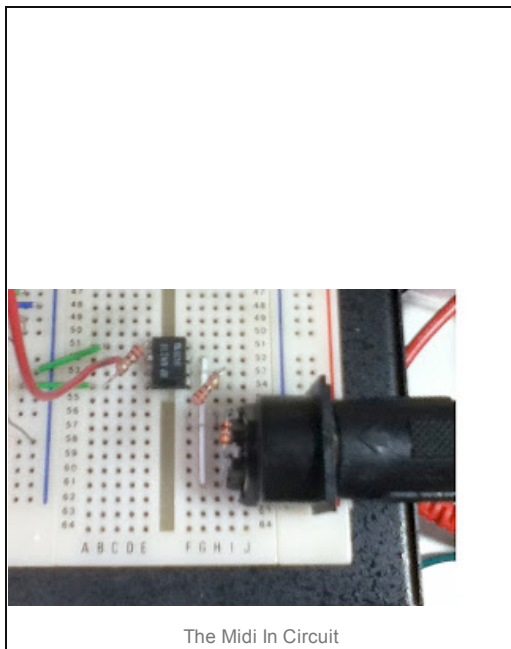
SATURDAY, 28 JANUARY 2012

Fun with Arduino - Midi Input Basics

Midi In - "Hello World!"

For a real world look at this circuit in action, check out the [NaV-1 Arduino synth](#) construction series.

Have you ever been working on a Arduino project and suddenly thought 'This thing could really use a MIDI Input!' This exact thing just happened to me. Not wanting to re-invent the wheel, I began searching around for a Library that would help. I came across the Arduino Midi Library which seemed to fit the bill. It took me a while to actually get the thing going, so I thought I would write a quick post outlining the steps to get a simple test circuit working. The following program and circuit will simply Flash the LED connected to Pin 13 on the Arduino Board when you press a note on a Midi Keyboard. But, that's really all you really need to confirm that you are correctly receiving Midi commands with your Arduino.



FOLLOW NOTES AND VOLTS

[Follow @NotesAndVolts](#)

Notes and Volts on

[Follow me on Pinterest](#)

LABELS

- [Music](#) (12)
- [Electronics](#) (11)
- [Geekery](#) (5)
- [Guitar](#) (5)

BLOG ARCHIVE

- ▼ [2012](#) (13)
 - ▶ [September](#) (1)
 - ▶ [August](#) (1)
 - ▶ [June](#) (2)
 - ▶ [May](#) (1)
 - ▶ [April](#) (2)
 - ▶ [March](#) (1)
 - ▶ [February](#) (1)
 - ▼ [January](#) (4)

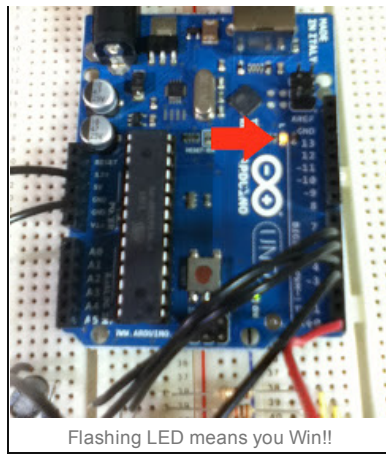
1000th Visitor!!!

[Fun with Arduino - Midi Input Basics](#)
[Chromatic Side-slipping with the Pentatonic Scale](#)
[Happy New Year from Notes and Volts](#)

- ▶ [2011](#) (10)

TOTAL PAGEVIEWS

~~8~~ ~~6~~ ~~5~~ ~~9~~



The Arduino Midi Library

First go to this [Link](#) and download the Arduino Midi Library files.

Unzip the downloaded folder. There are two folders inside. For Windows, copy the folder called "MIDI" and paste it into your Arduino "libraries" folder. Mine was located inside the "arduino-0022" folder. Quit and restart your Arduino IDE program. Go to the Menu and open Sketch > Import Library. You should see "MIDI" as one of the choices.

Copy and paste the code at the end of this article into a new Sketch. The code is commented, so give it a quick read through.

Here are some of the key commands:

MIDI.begin(MIDI_CHANNEL_OMNI);

This initializes the Midi Library. The MIDI_CHANNEL_OMNI parameter sets the library to listen to all Midi Channels. MIDI.begin(2) would set it to listen to Channel 2 only.

MIDI.setHandleNoteOn(MyHandleNoteOn);

This is an import command! The Arduino Midi Library uses something called 'Callbacks'. When a Midi event occurs, the Library will Call a function to handle it. This command tells the Library to call the 'MyHandleNoteOn' function when a **'Note On'** Midi event is detected. There are many callback functions in the Library to handle the many types of Midi events (Clock, Pitch Bend, Program Change, Etc..). Use the MIDI.set... command to point to the functions you require.

void MyHandleNoteOn(byte channel, byte pitch, byte velocity)

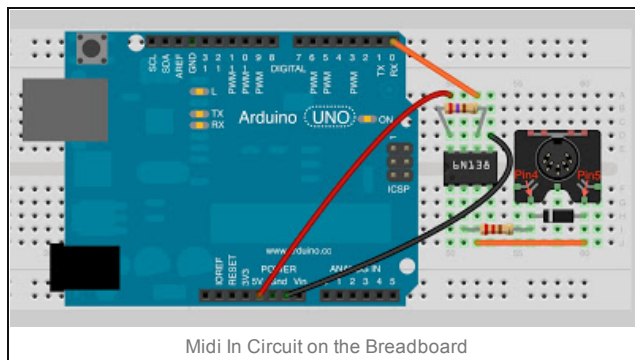
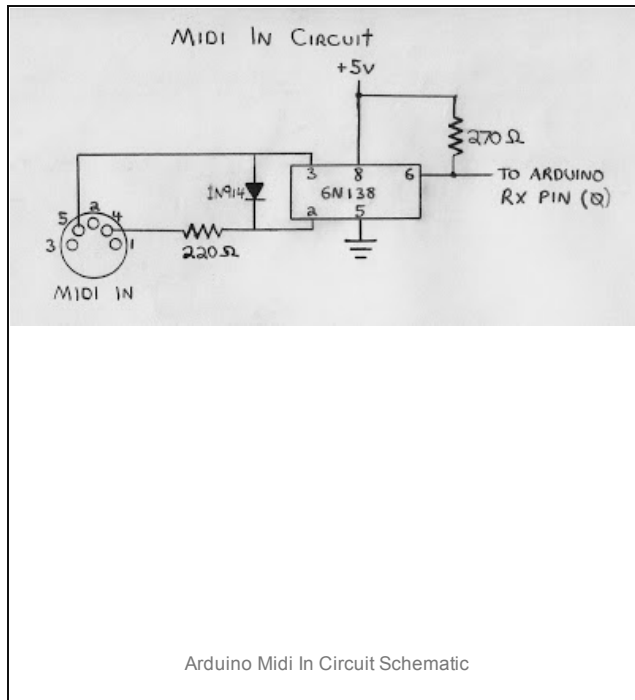
This is the function I created to be called when a Midi **Note On** event is detected. This is the 'meat' of your program. In this test program, I just have it flash the LED on the Arduino board. But you could just as easily have it play a note on your home made Synth circuit, Flash a spotlight on your Midi controlled lighting rig, or even command your Midi controlled Robotic Gorilla to enter 'Rampage' Mode. The sky is the limit.

MIDI.read();

This is the only function in the main loop of the program. It just checks the input buffer for any received Midi commands and passes them to the correct function.

The Hardware

The MIDI standard spells out the circuit that should be used for a MIDI INPUT so lets look at that first.



This very simple circuit uses a 6N138 optocoupler chip. This device basically electrically isolates your circuit from the incoming Midi signal. The 1N914 diode protects the chip from an incorrectly wired Midi cable. Plug the output of the optocoupler (Pin 6) into the RX (Pin 0) socket on your arduino board. **Note: Be sure to correctly identify pin 4 and 5 on the Midi In Jack.**

Also note that the RX/TX pins and the USB Port on the Arduino Board share the same signals. So, you will need temporarily remove the wire from the RX Pin 0 on the Arduino Board to upload a program. Then remove the USB Cable from the computer and replace the RX wire when you run the program with a Midi Input.

That's really all there is to it! Connect the Midi Out port on a Midi Keyboard to the Midi In Jack on your Arduino. If the little LED next to Pin 13 on the Arduino Board flashes when you hit a key, you are correctly receiving the Midi Data. Have fun!

For a real world look at this circuit in action, check out the [NaV-1 Arduino synth](#) construction series.

Copy the following code and paste into a new Arduino Sketch:

```
#include <MIDI.h> // Add Midi Library

#define LED 13 // Arduino Board LED is on Pin 13

// Below is my function that will be called by the Midi Library
// when a MIDI NOTE ON message is received.
// It will be passed bytes for Channel, Pitch, and Velocity
void MyHandleNoteOn(byte channel, byte pitch, byte velocity) {
    digitalWrite(LED,HIGH); //Turn LED on
    if (velocity == 0) { //A NOTE ON message with a velocity = Zero is actually a NOTE OFF
        digitalWrite(LED,LOW); //Turn LED off
    }
}

void setup() {
    pinMode (LED, OUTPUT); // Set Arduino board pin 13 to output
    MIDI.begin(MIDI_CHANNEL_OMNI); // Initialize the Midi Library.
    // OMNI sets it to listen to all channels.. MIDI.begin(2) would set it
    // to respond to channel 2 notes only.
    MIDI.setHandleNoteOn(MyHandleNoteOn); // This is important!! This command
    // tells the Midi Library which function I want called when a Note ON command
    // is received. in this case it's "MyHandleNoteOn".
}

void loop() { // Main loop
    MIDI.read(); // Continually check what Midi Commands have been received.
}
```

Posted by Dave at 1:27 AM

+1 Recommend this on Google

Labels: [Electronics](#), [Music](#)

16 comments:



--((()))--- 1 April 2012 03:37

excellent!

[Reply](#)

[Replies](#)



Dave 2 April 2012 01:18

Thanks! Glad you enjoyed it.



Anonymous 12 April 2012 18:52

Your breadboard diagram has an error. The diode should be at the optocoupler input not at the connector.



Dave 17 April 2012 22:53

You are correct! I totally missed that. Thx. Functionally, it should be fine either way but I'll try to correct the diagram soon.

[Reply](#)

**werebears** 28 August 2012 22:55

I have no idea what i can't seem to get this to work. Is there something I'm missing? I'm more or less following the breadboard diagram since, you know, i'm mostly an idiot.

[Reply](#)[Replies](#)**Dave** 29 August 2012 00:05

Hi WB. Sorry you're having trouble. What version of the Arduino IDE and Midi Library are you using? I wrote this with IDE .0022 and Midi Library V3.1.1. Are you able to compile and upload the program successfully? Here are a few things to check. Make sure that the pins on your Midi jack are correct. Remember to remove the wire from the Arduino RX pin to upload the program, then replace it to run. Make sure your keyboard is actually set to send Midi data. Let me know if this helps.

**werebears** 29 August 2012 13:58

Hey Dave, Well at one point when I pressed on note on my Korg the light turned on never to turn off again. Now I'm not getting anything at all, I'm afraid I've messed up the wiring. I'm using Midi Library V3.2 and I'm not sure what version of IDE i'm using I suppose 1.0

**Dave** 29 August 2012 14:29

Hey WB. If you like, send some pictures of your circuit to the email address in the "Contact" tab at the top of the page. I'd be glad to take a look.

**werebears** 29 August 2012 17:09

Hey Dave, Sent it out. If you need anymore photos let me know. I really appreciate you helping with this.

**Anonymous** 16 September 2012 13:36

I'm running into the same problem with my circuit, did you figure out WB's problem?

**Dave** 16 September 2012 13:53

I believe I fixed his problem. You want to check the orientation of your Midi jack and make sure you have it connected correctly. The breadboard diagram picture may be a bit confusing as the Midi jack is shown facing toward you (this was the only midi jack graphic I could find). I added the actual pin numbers in red underneath to try to make it clearer. If you look at the actual picture of the circuit at the top of the page, you can see how the jack is actually oriented on my board. Let me know if this helps.

[Reply](#)**Dave** 29 August 2012 14:28

This comment has been removed by the author.

[Reply](#)**jkor** 7 September 2012 09:16

hey, nice setup! Do you think it could be use to control that kind of strip with midi information : <http://www.adafruit.com/products/306> ?

[Reply](#)**jkor** 7 September 2012 09:18

sorry I meant that one : <http://www.adafruit.com/products/683>

[Reply](#)[Replies](#)**Dave** 7 September 2012 11:27

Hi jkor. You could easily control that via Midi. Just replace the code inside the "MyHandleNoteOn" function with the code to drive the LEDs. Every time a Midi note is received the MyHandleNoteOn function is called.

[Reply](#)**jkor** 13 September 2012 09:26

Hi Dave,
I think that this rgb led strip looks too weak for a stage show. I'm thinking of using a blinkM/MaxM chain with your Midi IN arduino setup. Do you have example of the <http://arduino.cc> midi library interfacing with blinkM/MaxM?
Thanks for your help!

[Reply](#)

Enter your comment...

Comment as: Google AccountPublishPreview

Links to this post

[Create a Link](#)[Newer Post](#)[Home](#)[Older Post](#)Subscribe to: [Post Comments \(Atom\)](#)

