

Simple DIY Electronic Music Projects

MIDI In, MIDI Monitors and Effects (WS002)

This worksheet contains a range of projects related to receiving MIDI messages with your Arduino. There are three MIDI monitoring projects, projects to turn your Arduino into a “tone module”, several project ideas for building or using MIDI shields, USBMIDI to MIDI converters and a MIDI effects unit.

Kevin

Parts List – WS002

- Arduino Uno (USB lead and power)
- Solderless Breadboard(s) or Stripboard
- Jumper Wires
- 19x 220Ω Resistors (red-red-brown)
- 4k7Ω Resistor (yellow-purple-red)
- 8x Red LEDs
- 8x Extra Red LEDs or 2x “LED Bar Displays”
- 1N914 Diode
- 6N138 Optoisolator
- 5-pin DIN Socket
- 8Ω Loudspeaker or old headphone speaker

Optional Shields

- MIDI Shield
- USB Host Shield
- LED Shields

All projects in this worksheet can be made using the above parts, but the quantities listed assume that parts will be re-used between projects.

Introduction to Arduino

You'll need a computer running the Arduino software. Details of how to get up and running can be found here:

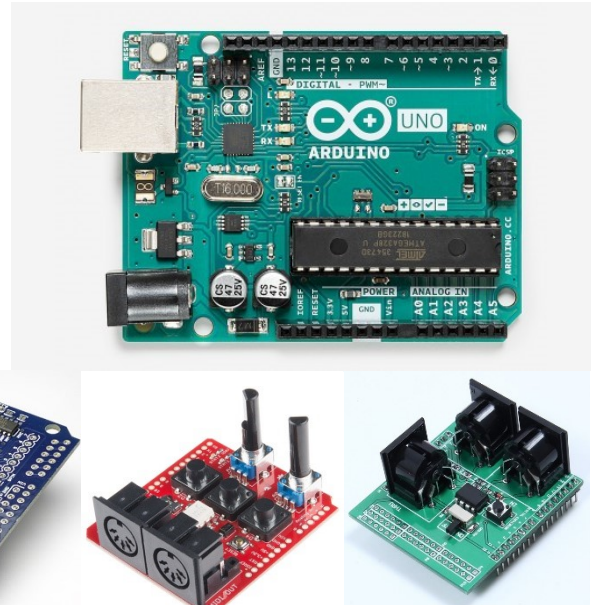
Lady Ada's Learn Arduino - <https://learn.adafruit.com/ladyadas-learn-arduino-lesson-number-0>

The official Arduino Getting Started pages - <https://www.arduino.cc/en/Guide/HomePage>

Caution!

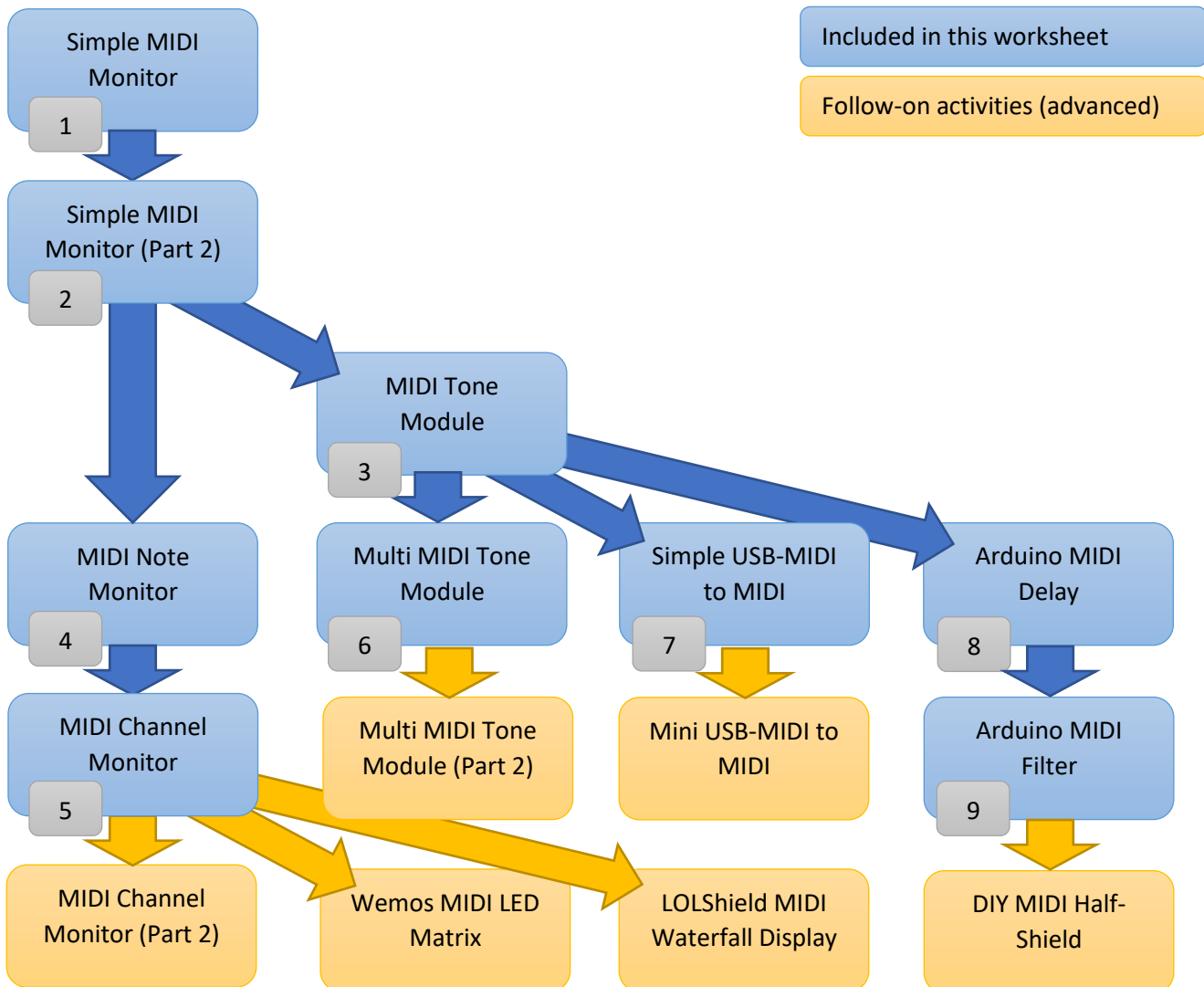
1. NEVER let your home-made electronics anywhere near live mains.
2. I strongly recommend powering your Arduino using a separate power supply – NOT your computer or any expensive music equipment with a USB port.
3. Always double check your wiring. Ideally test for any short circuits, especially between 5V and GND, with a multi-meter.
4. I strongly recommend getting hold of a second hand, cheap (ideally “sacrificial”) MIDI keyboard for your experiments.

WARNING: You undertake these projects at your own risk – I will not be held responsible for damage to expensive electronic, computer, music, or any other, equipment as a result of these experiments. Re-read point 4 above...



Simple DIY Electronic Music Projects

Suggested Project Sequence



The full technical details supporting all of these projects can be found on:

<https://diyelectromusic.wordpress.com/>

All the source code is available to download from GitHub on:

<https://github.com/diyelectromusic/sdemp>

There are many more projects to try on the above links too.

Simple DIY Electronic Music Projects

Simple MIDI Monitor (Project 1)

Full details: <https://diyelectromusic.wordpress.com/2020/06/11/simple-midi-monitor/>

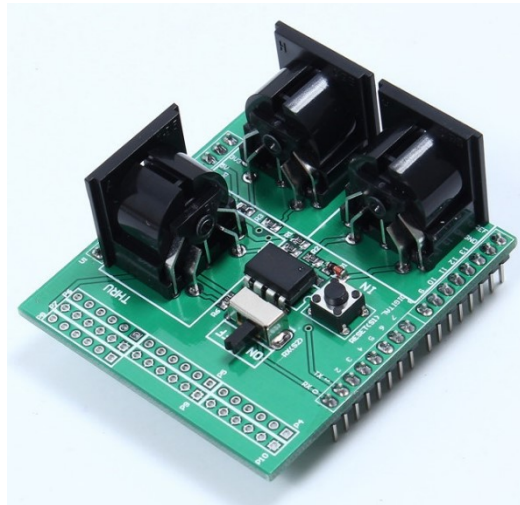
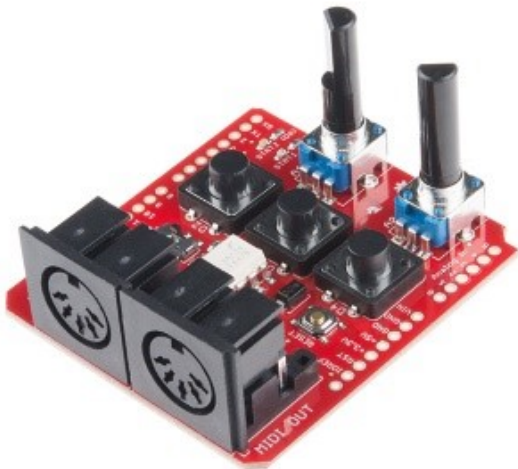
Code: <https://github.com/diyelectromusic/sdemp/tree/master/src/SDEMP/SimpleMIDIMonitor>

Parts List

- Arduino Uno
- MIDI Shield

The Circuit

This project uses an off-the-shelf MIDI shield for an Arduino Uno. Two examples are shown below.



The first is the MIDI Shield from Sparkfun, which you can find from a number of suppliers locally. The second is a generic MIDI Shield that you can find relatively cheaply using online auction sites.

The MIDI links are attached to the Arduino's RX and TX pins, which are also needed when uploading sketches, so you'll often find MIDI Shields include a switch to disable the MIDI links when uploading.

The Sparkfun shield has additional MIDI buttons and potentiometers which makes it easy to create MIDI projects. The unbranded shield includes a MIDI THRU port.

Basic Principles

This project utilises the built-in LED of the Arduino to show when it receives MIDI data. Depending on the shield being used the built-in LED might actually be underneath the shield so it is fairly easy to switch to using an LED (and associated resistor) on a different Arduino pin if required.

The provided code will poll the MIDI port as part of the Arduino's loop() function and if any MIDI note messages are received it will turn the LED on for a short period of time.

A small update would turn the LED on when receiving a Note On message and turn it off when receiving Note Off.

Further Reading

- Arduino MIDI Library - https://github.com/FortySevenEffects/arduino_midi_library
- Arduino Blink - <https://www.arduino.cc/en/Tutorial/Blink>

Simple DIY Electronic Music Projects

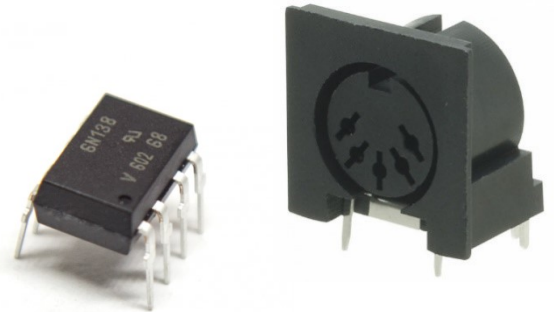
Simple MIDI Monitor Part 2 (Project 2)

Full details: <https://diyelectromusic.wordpress.com/2020/06/11/simple-midi-monitor-part-2/>

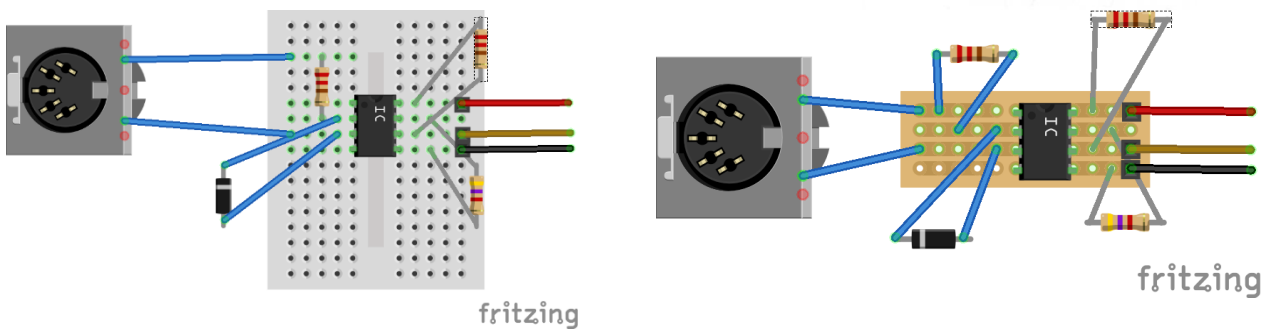
Code: <https://github.com/diyelectromusic/sdemp/tree/master/src/SDEMP/SimpleMIDIMonitor>

Parts List

- Arduino Uno
- 2x 220Ω Resistors (red-red-brown)
- 4k7Ω Resistor (yellow-purple-red)
- 6N138 Optoisolator
- 1N914 Diode
- Breadboard or Stripboard and jumper wires



The Circuit

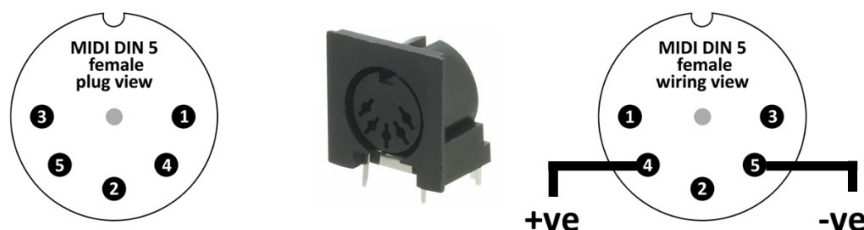


Basic Principles

The functionality of this project is the same as the previous one, but this time we are making our own MIDI in (receive) circuit. There are two things to note when building this circuit:

- Make sure the diode goes the right way round, with the “line” on the case as indicated connecting to pin 4 of the MIDI connector and pin 2 of the 6N138.
- Make sure you use the right pins on the 5-pin DIN connector:
 - Pin 5 (negative) goes to the cathode pin of the 6N138 (pin 3);
 - Pin 4 (positive) goes to the anode pin of the 6N138 (pin 2) via the 220Ω resistor.

If the polarity of the diode seems backwards, don't worry – it is there to protect the optoisolator in case something is plugged in that is wired incorrectly. It is “reverse polarity protection” as described in the MIDI spec.



The whole thing is powered from the +5V and GND links back to your Arduino, and the signal is connected to the Arduino RX pin. If you build it on stripboard it makes for a nicely self-contained module that you can use over and over.

Further Reading

- Arduino MIDI Library - https://github.com/FortySevenEffects/arduino_midi_library
- Midi.org Electrical Specification - <https://www.midi.org/specifications/item/midi-din-electrical-specification>

Simple DIY Electronic Music Projects

MIDI Tone Module (Project 3)

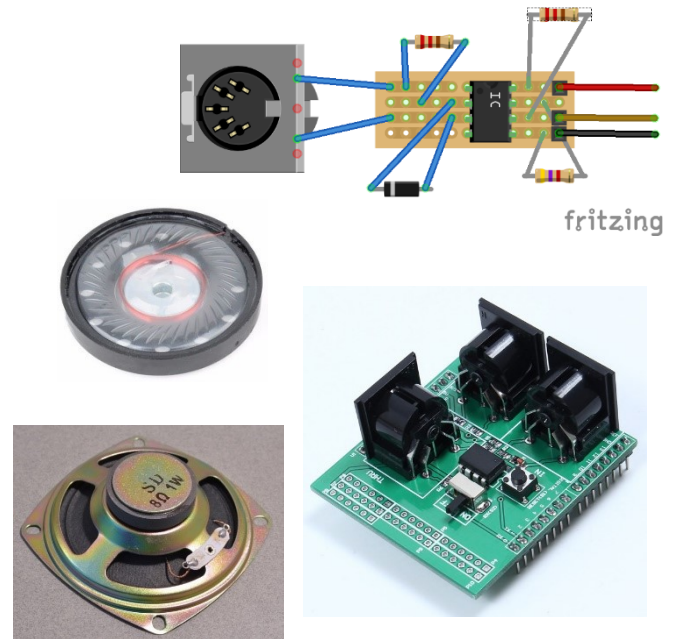
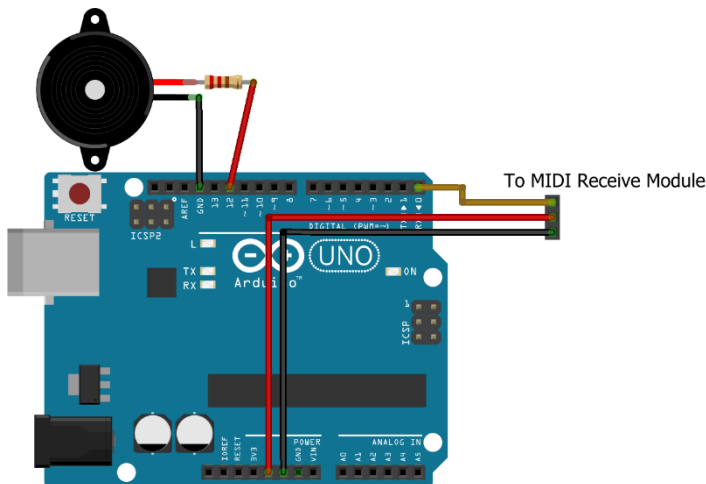
Full details: <https://diyelectromusic.wordpress.com/2020/06/12/arduino-midi-tone-module/>

Code: <https://github.com/diyelectromusic/sdemp/tree/master/src/SDEMP/ArduinoMIDIToneModule>

Parts List

- Arduino Uno
- 1x 220Ω Resistor (red-red-brown)
- 8Ω Loudspeaker or old headphone speaker
- Breadboard and jumper wires
- MIDI Receive Module (see project 8)

The Circuit



Basic Principles

The Arduino is listening for MIDI Note On and Note Off messages via its serial port (the RX pin). This is connected either to a standalone MIDI receive module or a MIDI shield. When it receives a note message it is turned into a frequency which is then passed into the Arduino tone() function which will play a note at that frequency which is sent to pin 12. This can be directly connected, via a resistor, to a small loudspeaker.

This demonstrates a slightly more sophisticated way of handling MIDI notes – using the idea of “callback” functions. These are functions that are registered with the MIDI library and automatically called (“called back”) when a note is received.

This “module” can only play one note at once.

Further Reading

- Arduino Tone Medley - <https://www.arduino.cc/en/Tutorial/ToneMelody>
- Arduino MIDI Library - https://github.com/FortySevenEffects/arduino_midi_library/wiki/Using-Callbacks

Simple DIY Electronic Music Projects

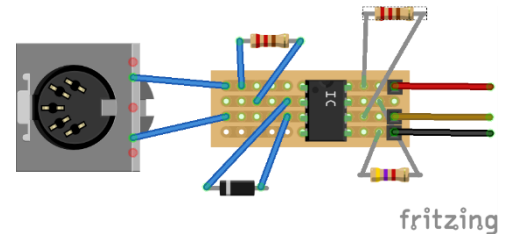
MIDI Note Monitor (Project 4)

Full details: <https://diyelectromusic.wordpress.com/2020/06/13/arduino-midi-note-monitor/>

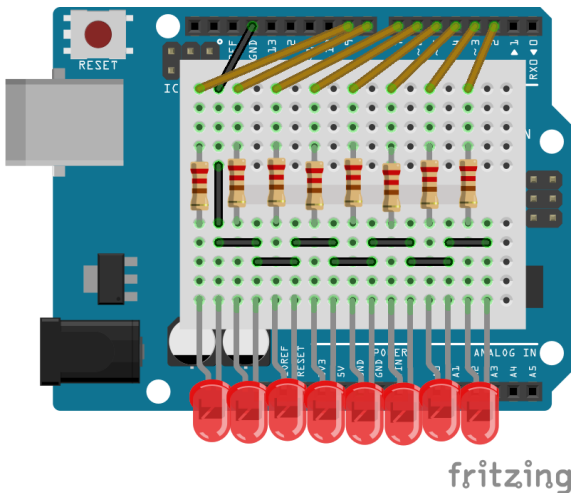
Code: <https://github.com/diyelectromusic/sdemp/tree/master/src/SDEMP/ArduinoMIDIKeyMonitor>

Parts List

- Arduino Uno
- 8x Red LEDs
- 8x 220Ω Resistors (red-red-brown)
- MIDI Receive Module (see project 8)
- Breadboard and jumper wires



The Circuit



Basic Principles

The Arduino is listening for Note On and Note Off messages using “callbacks” from within the Arduino MIDI Library but instead of playing a tone it now lights up an LED. The code has a list of which pins are connected to LEDs and has a list of MIDI notes to respond to for each LED.

This example uses 8 LEDs but there is no reason why you couldn't have more if you wanted, but eventually you will run out of IO pins on the Arduino! There are many techniques for using a lot more LEDs with an Arduino. I particularly liked using the “Lots of LEDs Shield” (“LOLShield”) to make a great “note waterfall” effect. For some details, look up the “LOLShield Waterfall Display” and the “Wemos MIDI LED Matrix” in the advanced projects area of the website.

You can also get breakout boards with more LEDs on them. Some will include resistors and some won't so be sure to check.

If you have different coloured LEDs that is fine too, but you will need to use the correct resistor value that matches the LEDs you have. There are lots of LED resistor calculators on the Internet. Remember this will be running with a 5V power supply to the LED and you need to keep the current to less than 20mA so you don't overload the Arduino output pins.

Further Reading

- Arduino Blink - <https://www.arduino.cc/en/Tutorial/Blink>
- Arduino MIDI Library - https://github.com/FortySevenEffects/arduino_midi_library

Simple DIY Electronic Music Projects

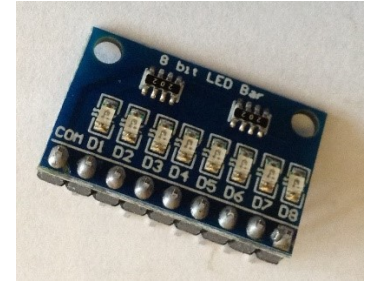
MIDI Channel Monitor (Project 5)

Full details: <https://diyelectromusic.wordpress.com/2020/07/20/arduino-midi-channel-monitor/>

Code: <https://github.com/diyelectromusic/sdemp/tree/master/src/SDEMP/ArduinoMIDIChannelMonitor>

Parts List

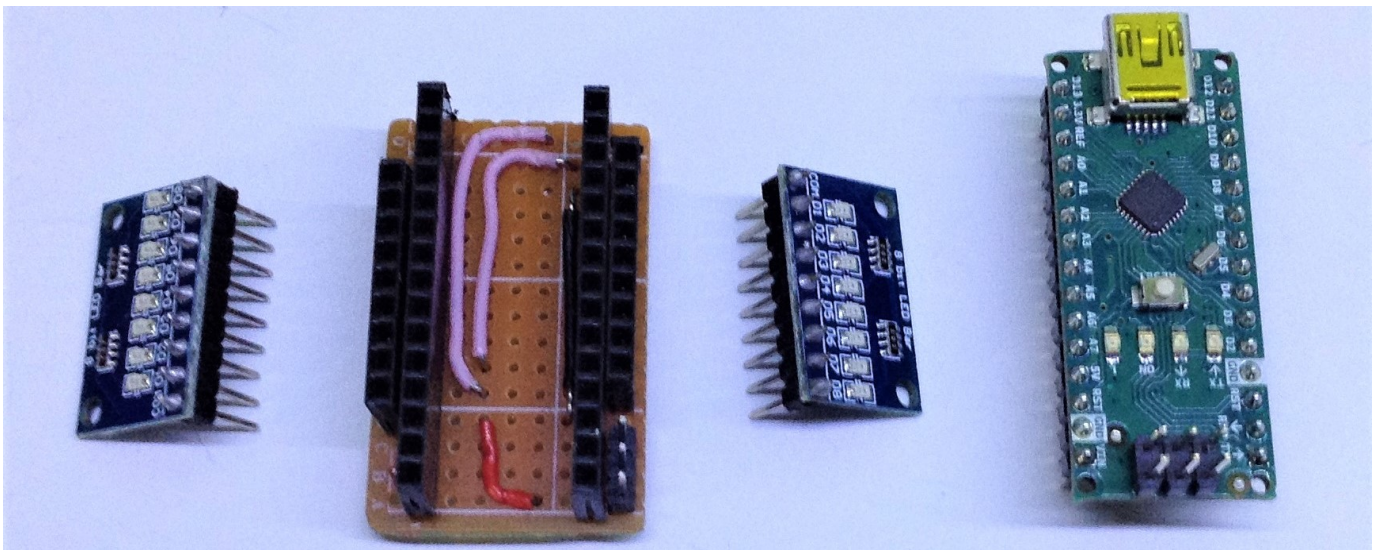
- Arduino Uno or Arduino Nano
- 16x Red LEDs or 2x 8-LED common cathode “8-bit bars”
- 16x 220Ω Resistors (red-red-brown) if using Red LEDs
- MIDI Receive Module (see project 8)
- Breadboard and jumper wires



The Circuit

This is using the same circuit as the MIDI Note Monitor (project 10), but with double the number of LEDs. You can continue to use an Arduino Uno and discrete LEDs, but I found it easier to use an Arduino Nano and two breakout boards each containing 8 LEDs in a “common cathode” configuration and their associated resistors as shown below. Although I had to bend out the common “ground pin” so it could be connected to the Arduino’s ground (see the project pages for more details).

If you are comfortable soldering, then this is a good project to commit to stripboard (more details in Part 2 on the project pages).



Basic Principles

This uses the Arduino MIDI Library in “polling” mode, it has the loop() function checking for MIDI messages every time it is run and if it sees any it will turn on the LED corresponding to the MIDI channel it was receiving on. The only slight complication is including some additional code to determine how long the LED stays on for.

Using two of these 8-bit “bars” there is one LED for each MIDI channel and on startup it counts through each channel LED so you can see the ordering of the channels.

Further Reading

- Arduino Blink - <https://www.arduino.cc/en/Tutorial/Blink>
- Arduino MIDI Library - https://github.com/FortySevenEffects/arduino_midi_library

Simple DIY Electronic Music Projects

Multi MIDI Tone Module (Project 6)

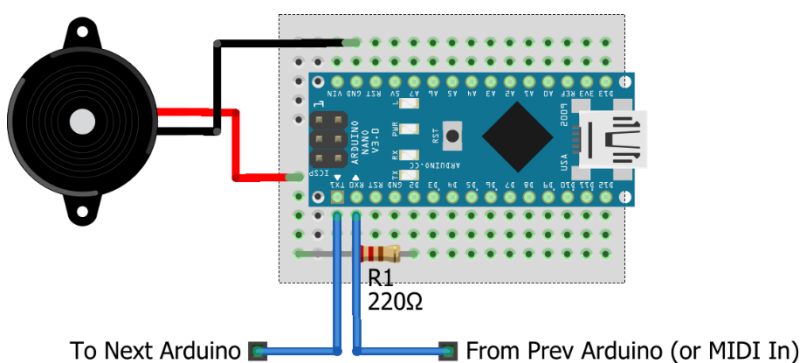
Full details: <https://diyelectromusic.wordpress.com/2020/08/02/arduino-multi-midi-tone-module/>

Code: <https://github.com/diyelectromusic/sdemp/tree/master/src/SDEMP/ArduinoMultiMIDIToneModule>

Parts List

- Arduino Uno or Arduino Nano (two or more)
- 8Ω Loudspeaker or old headphone speaker per Arduino
- 1x 220Ω resistor per Arduino
- MIDI Receive Module (see project 8)
- Breadboard and jumper wires

The Circuit



fritzing

The above circuit should be made for each Arduino. The basic idea is to link the transmit (TX) of one Arduino to the receive (RX) of the next, with the receive from first Arduino in the chain linked to the MIDI in circuit.

Basic Principles

Each Arduino will listen for MIDI messages and respond to Note On messages to play a tone. If it plays a tone, the MIDI message goes no further. If it receives a Note On message but is already playing a tone then it will retransmit the message through to the next Arduino in the chain using its TX pin so that one can play the tone instead. This carries on until we run out of Arduinos!

There are some limitations, perhaps the most curious being that if you play a chord you might get the different notes of the chord played on different Arduinos each time, depending on the ordering of the received messages. But it is simple and you can play as many notes at the same time as you have Arduinos (or at least until the delay in passing notes down the chain becomes noticeable)!

A further extension is described on the website – how to build a simple “chaining” stripboard to make the wiring and power a bit simpler.

Further Reading

- Arduino Tone Melody - <https://www.arduino.cc/en/Tutorial/ToneMelody>
- Arduino MIDI Library - https://github.com/FortySevenEffects/arduino_midi_library/wiki/Using-Callbacks

Simple DIY Electronic Music Projects

Simple USBMIDI to MIDI (Project 7)

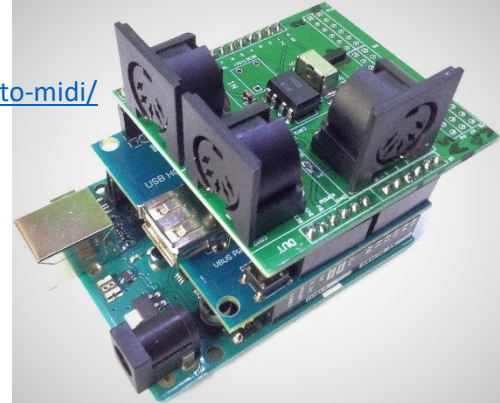
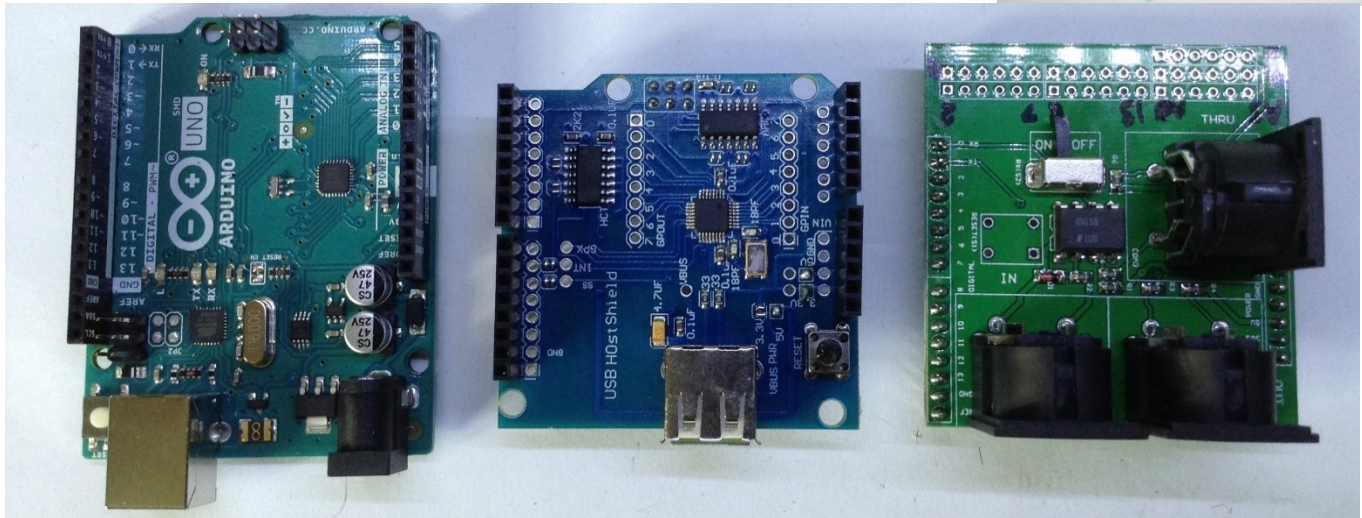
Full details: <https://diyelectromusic.wordpress.com/2020/07/29/simple-usb-midi-to-midi/>

Code: Uses example code from the USBMIDI Library

Parts List

- Arduino Uno
- USB Host Shield
- MIDI Shield

The Circuit



This project uses two “off the shelf” shields for the Arduino.

On the right is a common MIDI shield and centre is a USB Host Shield. When combined with an Arduino Uno they can be used to create a USBMIDI to MIDI converter. It may be possible to create a MIDI to USBMIDI converter too, but this project just looks at receiving USBMIDI messages and passing them out over standard MIDI.

The one thing to watch out for with this project is making sure the USB Host Shield is correctly configured via jumpers for the Arduino’s 5V power supply. Full details are provided on the project pages.

Basic Principles

There is a library available for the USB Host Shield that includes MIDI functionality which can be found here:

- https://github.com/felis/USB_Host_Shield_2.0

One of the examples is a USB_Midi_Converter and it literally just works.

As an extension project, I was able to use a mini USB host shield, a 3.3V Arduino Pro Mini and some homemade electronics to construct a smaller, all-in-one “MIDIUSB in, standard MIDI out” device. See the advanced project pages for details.

Further Reading

- Arduino USB Host Shield Projects - https://chome.nerpa.tech/arduino_usb_host_shield_projects/

Simple DIY Electronic Music Projects

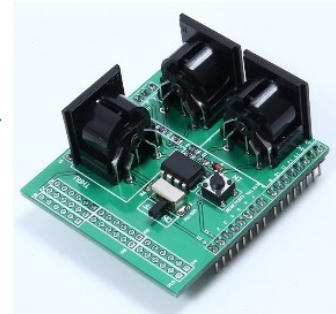
Arduino MIDI Delay (Project 8)

Full details: <https://diyelectromusic.wordpress.com/2020/08/11/arduino-midi-delay/>

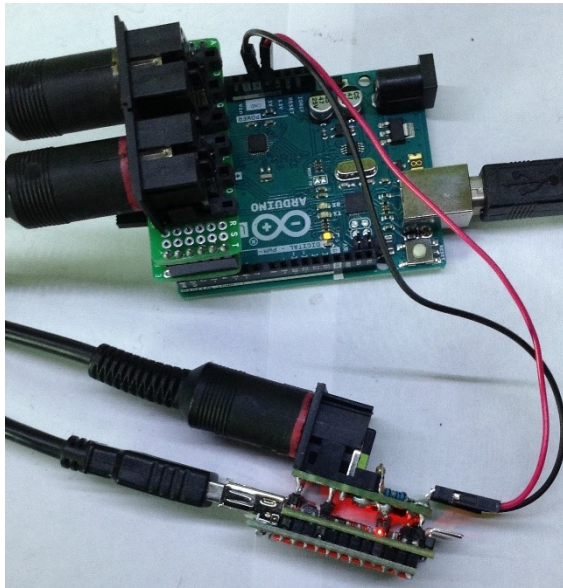
Code: <https://github.com/diyelectromusic/sdemp/tree/master/src/SDEMP/ArduinoMIDIDelay>

Parts List

- Arduino Uno
- MIDI Shield (or modules supporting receive and transmit)

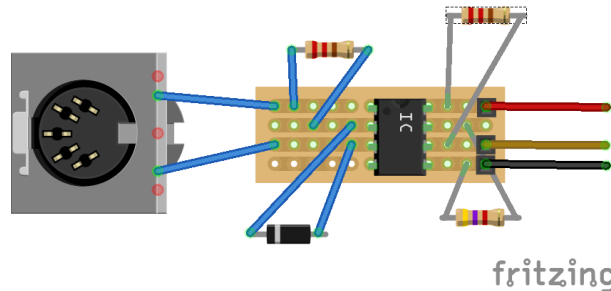


The Circuit



This is primarily a software project, using either the DIY MIDI receive module, the off-the-shelf shield or even the USBMIDI to MIDI interface.

In the photo to the left, I'm using the mini USBMIDI to MIDI converter (powered from the Arduino and linked via MIDI) and a home-made "half MIDI Shield" – details for both of these are provided in the advanced projects section of the website.



Basic Principles

This implements a MIDI delay function by keeping track of which notes have been played and replaying them back after a fixed delay. The code as written can store up to 16 notes at a time but there are a number of configuration parameters at the top of the file to experiment with, including the number of echoes, the timings, which MIDI channel to listen out for and so on. It is even possible to change the pitch of the echoes.

The code makes use of the Arduino's built-in timers, using the TimerOne library, to give me a basic "tick" within which I can decide if I am listening for notes, counting down time intervals, or playing echoes. The "tick" was configured to run at 1kHz i.e. 1000 times each second. Full details of the internals are provided on the project website, but you don't need to know any of this to make use of this code and experiment with different effects!

Once you have the basic idea of receiving, processing and re-transmitting a MIDI stream, there are lots of other things you can try too.

Further Reading

- Arduino Timer1 - <https://playground.arduino.cc/Code/Timer1/>
- Arduino MIDI Library – https://github.com/FortySevenEffects/arduino_midi_library/wiki/Using-Callbacks
- Arduino noInterrupts - <https://www.arduino.cc/reference/en/language/functions/interrupts/nointerrupts/>

Simple DIY Electronic Music Projects

Arduino MIDI Filter (Project 9)

Full details: <https://diyelectromusic.wordpress.com/2020/08/11/arduino-midi-delay/>

Code: <https://github.com/diyelectromusic/sdemp/tree/master/src/SDEMP/ArduinoMIDIDelay>

Parts List

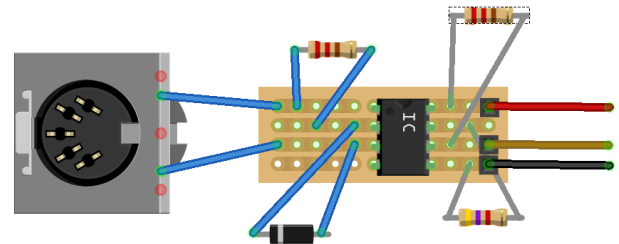
- Arduino Uno
- MIDI Shield (or modules supporting receive and transmit)



The Circuit

This too is primarily a software project, using either the DIY MIDI receive module, the off-the-shelf shield or even the USBMIDI to MIDI interface.

As you'd expect the MIDI module is connected to the Arduino's RX and TX pins.



fritzing

Basic Principles

This implements a MIDI filter by having a configured set of MIDI channels to listen out for and only passing on messages received if they match the list of channels we are interested in.

The list of channels is defined using a bit pattern at the top of the file, so complex combinations of multiple channels are possible.

This is a great way to reduce the processing overhead of some of the other modules and works particularly well with the "multi tone" project described above reducing the processing overheads on the first Arduino of examining and then rejecting any MIDI messages it doesn't need to see.

Further Reading

- Arduino MIDI Library – https://github.com/FortySevenEffects/arduino_midi_library/wiki/Using-Callbacks

Simple DIY Electronic Music Projects

The Back Page

If you've found this useful there are plenty more projects to take a look at on my site:

<https://diyelectromusic.wordpress.com/>

Any feedback, thoughts, comments or ideas are welcome.



diyelectromusic@gmail.com



<https://twitter.com/diyelectromusic>



<https://www.facebook.com/diyelectromusic/>

If you like what you see, then please considering buying me a Ko-fi:



<https://ko-fi.com/diyelectromusic>

Everything you see has been built on the "shoulders" of those who share their tools, techniques, know-how, code and expertise openly via the Internet.

In particular, this wouldn't be possible without the Arduino ecosystem. You can support this ecosystem directly with financial support here:

<https://www.arduino.cc/en/Main/Contribute>



Official distributors for Arduino boards and kits in the UK include, among others:

<https://shop.pimoroni.com/>

<https://thepihut.com/>

<https://hobbytronics.co.uk/>

All content and code Copyright © 2020 diyelectromusic (Kevin). All code is released under a MIT license.

Thanks for playing along.

Kevin

