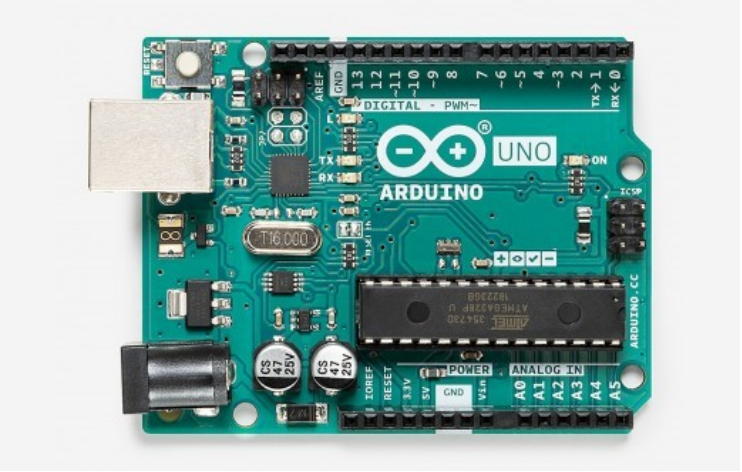
Drums and Touch (WS003)

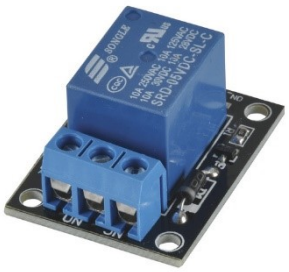
This worksheet contains a range of projects to detect "touch" and use it as a trigger for playing musical notes. It includes three great "ready to go" boards that are very easy to get started with and have a wide range of other educational uses too. It also includes several projects to get percussive sounds out of relays and servos and use them to create music.

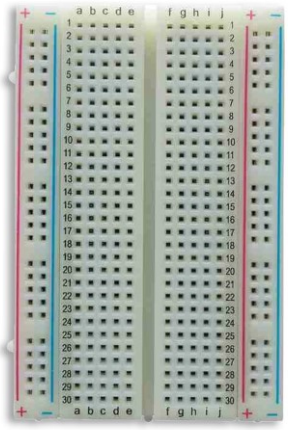
Kevin



Parts List – WS003

- Arduino Uno (USB lead and power)
- Arduino Relay Modules (several)
- Miniature Servos (several)
- 1x 220Ω resistor (red-red-brown)
- 13x 1MΩ resistors (brown-black-green)
- 5x 1N4733A Zener diodes (5.1V)
- 5x piezo disk sensors and "drum pads" (plastic lids)
- 5-pin 180 DIN plug or socket
- Foil, copper tape, or other "touchable" surfaces
- Solderless Breadboard(s) and jumper wires
- 8Ω Loudspeaker or old headphone speaker





Additional modules or boards (optional)

- MPR121 based "touch module"
- Adafruit Circuit Playground Express
- Bare Conductive Touch Board
- BBC Micro:bit and Keyboard add-on module

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:

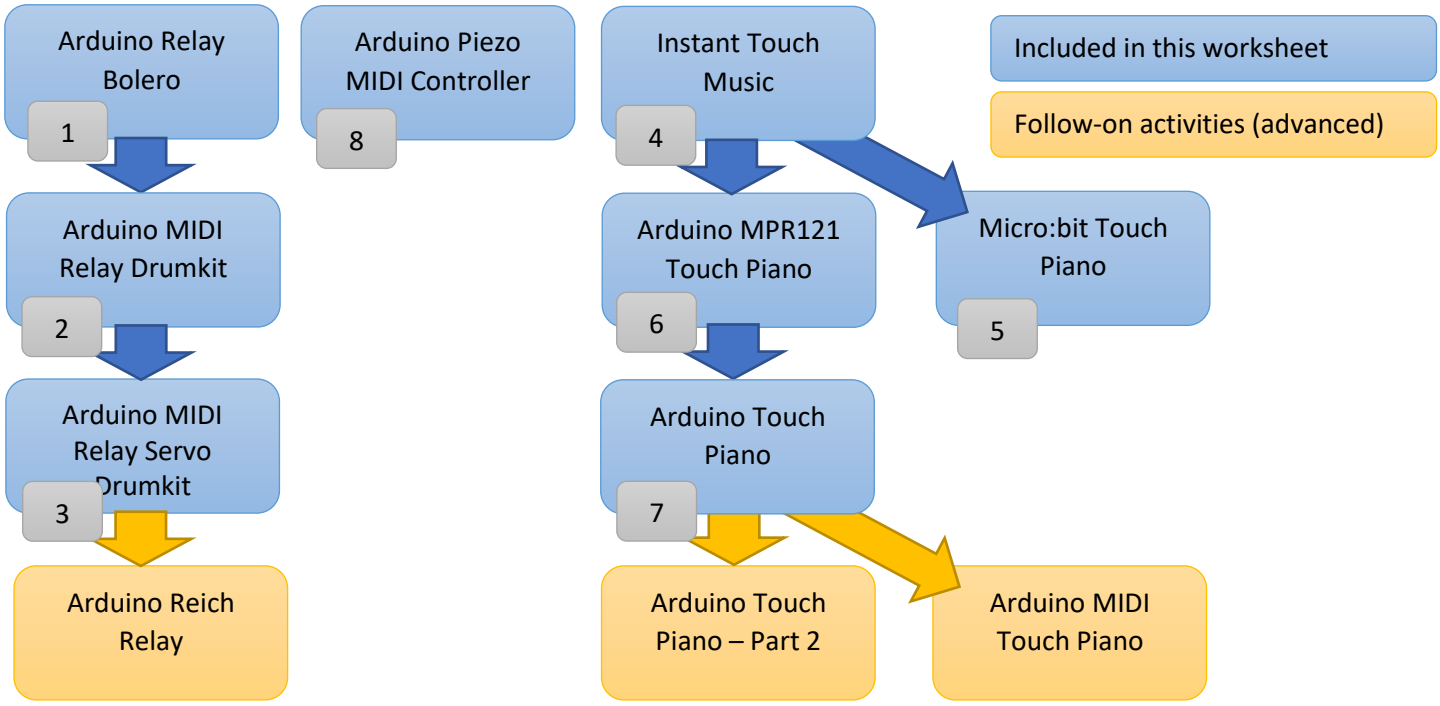
> Lady Ada's Learn Arduino - https://learn.adafruit.com/ladyadas-learn-arduino-lesson-number-0
> The official Arduino Getting Started pages - https://www.arduino.cc/en/Guide/HomePage

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…

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

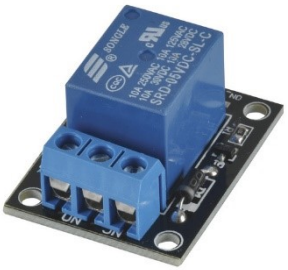
There are many more projects to try on the above links too.

Arduino Relay Bolero (Project 1)

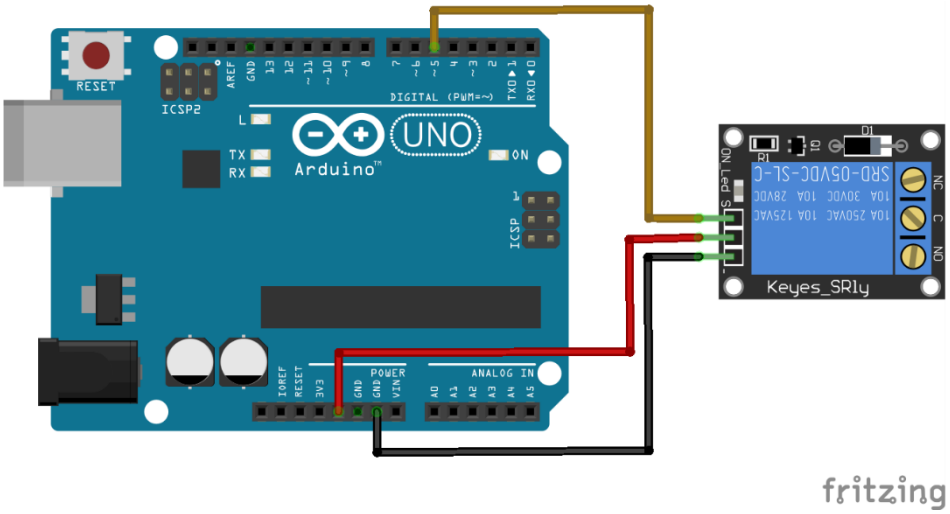
Full details: https://diyelectromusic.wordpress.com/2020/06/02/arduino-relay-bolero/
Code: https://github.com/diyelectromusic/sdemp/tree/master/src/SDEMP/ArduinoBolero

Parts List

- Arduino Uno
- Relay Module
- Breadboard and jumper wires



The Circuit



The relay module needs connecting to 5V, GND and one fo the Arduino digital output pins. Note that the pinouts on relay modules can be quite different, so check (and double check) your wiring.

Basic Principles

This project uses the "click" of a relay as the percussive component of a rhythm. In this case the repetitive pattern of the snare in Ravel's Bolero. The relay isn't controlling anything itself, I'm just using the click so to play a note the code just has to toggle the relay – if it was already open, close it; if it was already closed, then open it.

The timing of the rhythm is achieved using the Arduino delay() function, so the rhythm to be played is simply a series of values to be passed to the delay to create a pattern between the toggles.

As the "Bolero" rhythm includes triplets a crotchet is set to 450 milliseconds and (triplet) quavers are 150 each, although it is fairly easy to change the values in the code to experiment with different timings.

It is possible to drive several relays to create more complex rhythms. I used five to recreate Steve Reich's "Music for Pieces of Wood" in the "Arduino Reich Relay" project listed on my website.

Further Reading

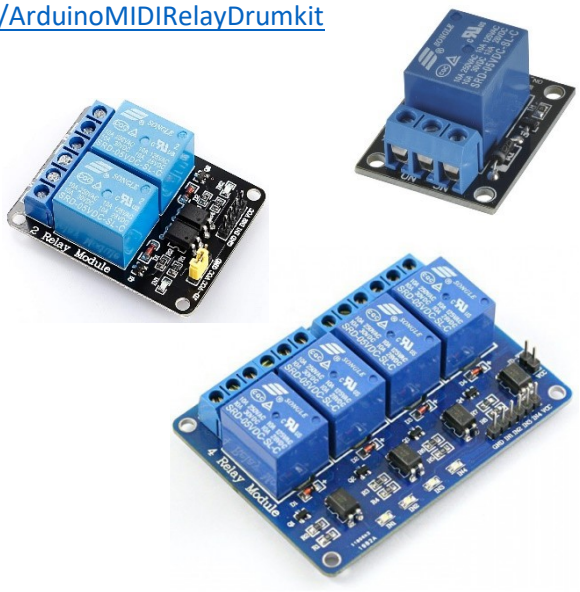
- Arduino Digital Pins - https://www.arduino.cc/en/Tutorial/DigitalPins
- Arduino Relay Tutorial - https://arduinogetstarted.com/tutorials/arduino-relay

Arduino MIDI Relay Drumkit (Project 2)

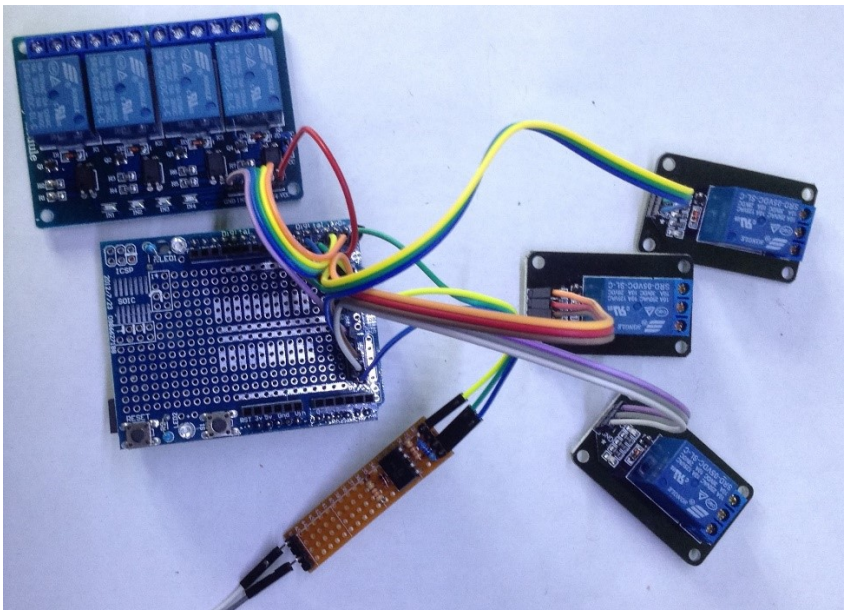
Full details: https://diyelectromusic.wordpress.com/2020/06/14/arduino-midi-relay-drumkit/
Code: https://github.com/diyelectromusic/sdemp/tree/master/src/SDEMP/ArduinoMIDIRelayDrumkit

Parts List

- Arduino Uno
- Arduino Relay Modules (several)
- Jumper wires
- Some kind of MIDI In circuit (see WS002).



The Circuit



This is essentially several copies of the circuit used in the first project. Each relay will be connected to a common 5V and GND, but the signal pins must be connected to different Arduino digital output pins.

Basic Principles

A number of relays are wired up to the Arduino digital output pins and a MIDI receive module is hooked up to the RX pin. The code listens for MIDI note messages on channel 10 (which is typically used for drum tracks) and different notes will activate different relays.

To get different timbres from the relays they can be attached to resonant surfaces such as plastic boxes or pieces of wood – get inventive! Just don't short out any of the electrical links on the circuit board by attaching it to anything conductive.

The code has to keep track of the state of the different drums, which digital pins are related to which drums and which MIDI note messages to listen out for, for each drum. The notes used correspond to the General MIDI standard drum notes for MIDI channel 10, but it is fully configurable by changing the values in the code. I've configured one relay as the "bass drum", one as a "snare" and several as different "toms".

Further Reading

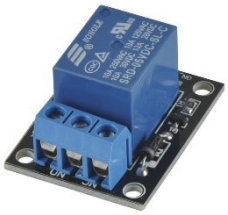
- Arduino Digital Pins - https://www.arduino.cc/en/Tutorial/DigitalPins
- Arduino Relay Tutorial - https://arduinogetstarted.com/tutorials/arduino-relay
- Arduino MIDI Library - https://github.com/FortySevenEffects/arduino_midi_library

Arduino MIDI Relay Servo Drumkit (Project 3)

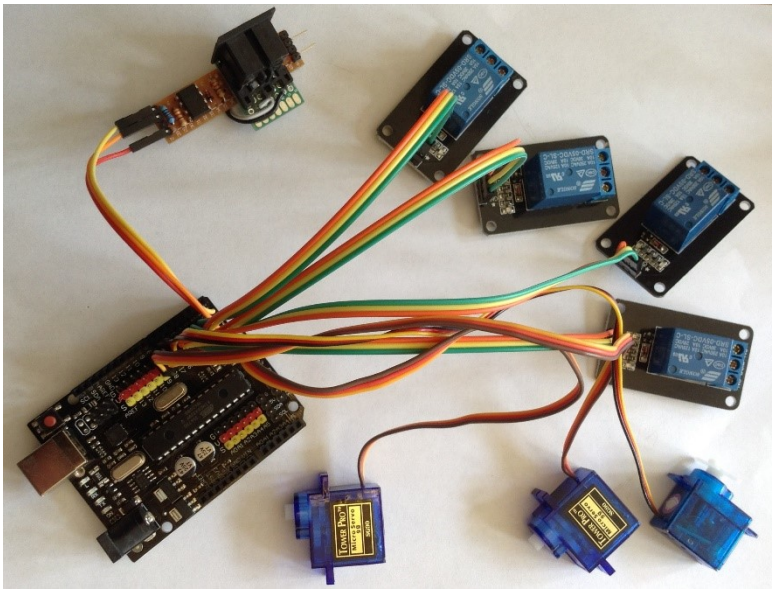
Full details: https://diyelectromusic.wordpress.com/2020/06/16/arduino-midi-relay-servo-drumkit/
Code: https://github.com/diyelectromusic/sdemp/tree/master/src/SDEMP/ArduinoMIDIRelayServoDrumkit

Parts List

- Arduino Uno
- Arduino Relay Modules (several)
- Arduino Servos (several)
- Jumper wires
- Some kind of MIDI In circuit (see WS002).
- Breadboard and jumper wires



The Circuit



Like the relay modules, miniature servos require three wires – 5V, GND and signal. They can draw a fair bit of power if driving a physical load, but as we're just using them "as is" for the sound, we don't need to worry about that here.

Basic Principles

As with the previous project, this code listens for MIDI note messages on channel 10 and translates those into actions to play a "drum" (relay) or "cymbal" (servo). Servos need to be given a rotation, so the noise is created by making it rotate a short way for one "beat" and when the next note is required, it rotates back.

A larger number of lists are required to keep track of all the different states of relays, servos, notes, positions and Arduino IO pins to drive them all, but as in the previous project the notes to be used, number of servos and relays, pins to connect them to, is fully configurable in the code.

Further Reading

- Arduino Digital Pins - https://www.arduino.cc/en/Tutorial/DigitalPins
- Arduino Relay Tutorial - https://arduinogetstarted.com/tutorials/arduino-relay
- Arduino Servo Sweep Tutorial - https://www.arduino.cc/en/tutorial/sweep
- Arduino MIDI Library - https://github.com/FortySevenEffects/arduino_midi_library

Instant Touch Music (Project 4)

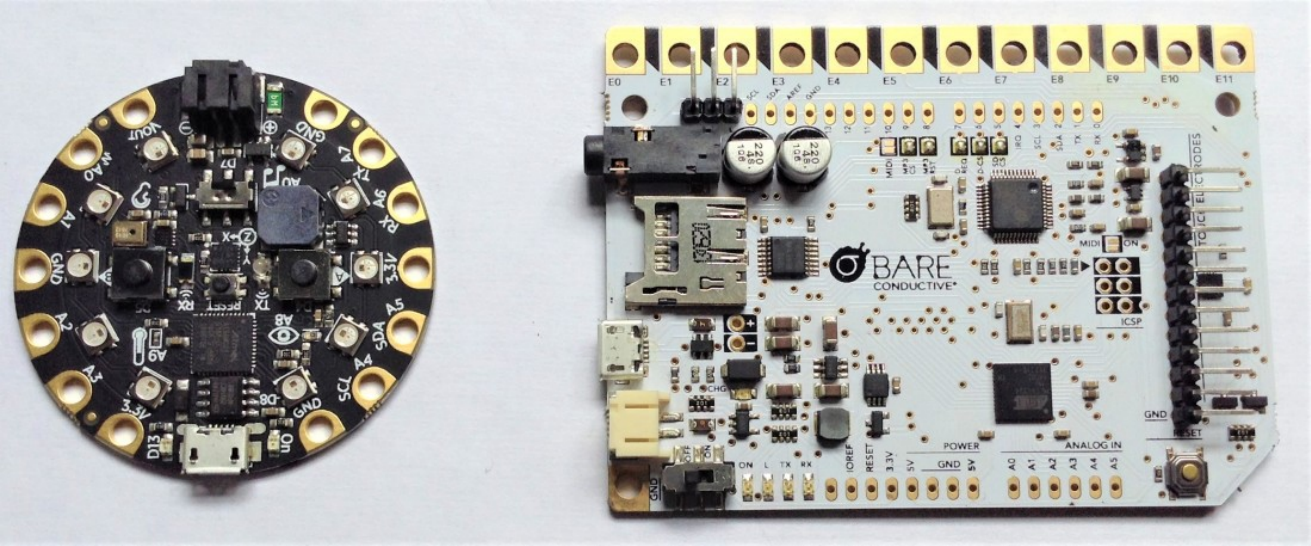
Full details: https://diyelectromusic.wordpress.com/2020/08/23/instant-touch-music/

Parts List

- Adafruit Circuit Playground Express or Bare Conductive Touch Board
- Wires and "touchable" objects

For a third alternative module, see the next project too.

The Circuit

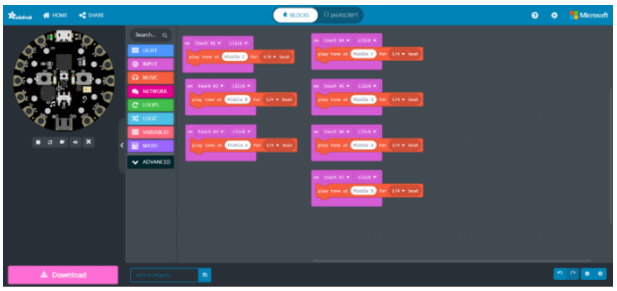


Basic Principles

Both of these devices are designed to get beginners playing with electronics (and as a side effect for us, electronic music) pretty quickly and easily.

The Circuit Playground Express (CPX) is the cheaper of the two, and much more of a general purpose educational device, but is limited to 7 touch inputs. The Touch Board is more expensive but is designed for "touch music" (and other) applications right out of the box and supports a full chromatic octave's worth of notes. They also sell a range of conductive paint which allows you to create a whole range of "touchable" surfaces for use with your board.

The CPX can be programmed using a block-level language (shown right), using the Python language, or the Arduino environment.



The Touch Board can be used 'as is' as a "touch piano", but if you want to get programming is fully programmable using the Arduino environment.

Both devices can have wires attached using crocodile clips (or simply bending wires into the holes) and those wires can be used "bare" or attached to other items. It seems very popular to create "fruit pianos" on the Internet!

Note: I have no affiliation with either of these companies. These are just two really great "beginner" boards for "instant touch music" applications! As is the board described in the next project…

Further Reading

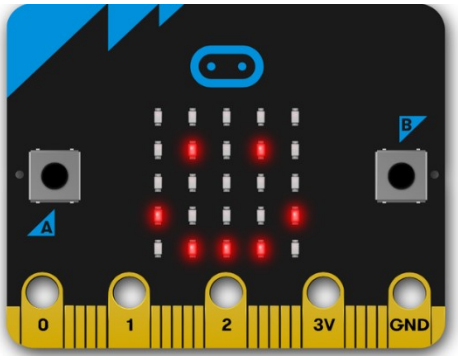
- Adafruit Circuit Playground Express - https://learn.adafruit.com/adafruit-circuit-playground-express/overview
- Bare Conductive Touch Board - https://www.bareconductive.com/make/introducing-the-touch-board/

Micro:bit Touch Piano (Project 5)

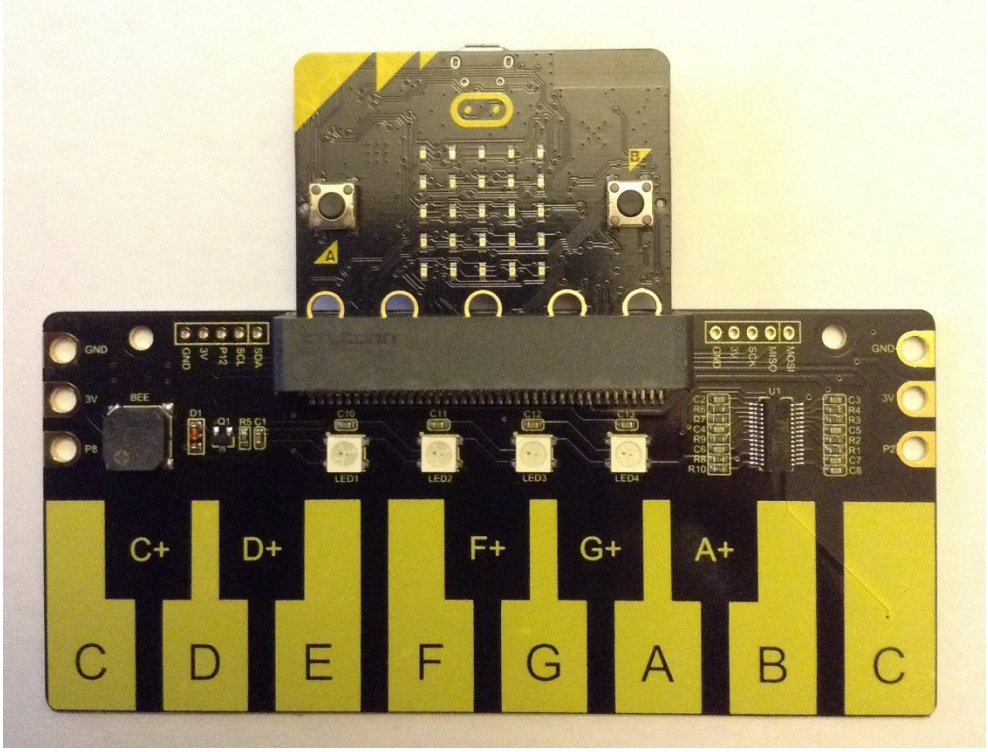
Full details: https://diyelectromusic.wordpress.com/2020/10/29/microbit-touch-piano/

Parts List

- BBC Micro:bit
- BBC Micro:bit "Music Keyboard" module



The Circuit



Basic Principles

As with the previous project, the BBC Micro:bit is designed as a cheap, educational electronics module to introduce beginners to programming, computers and electronics. It also happens to be even cheaper than both of those boards.

A detailed introduction isn't possible here (and wouldn't do it justice) but you can get a great feel for what the device can offer simply by browsing some of the projects on https://microbit.org/.

Above is a "touch music keyboard" add-on provided by Keyestudio. There are several variants available. At the time of writing, version 2 of the Micro:bit has just been released, but I believe the module used here would work with either version of the board – if not then I'm sure there are alternative keyboards available too.

As with the Adafruit CPX there is a nice "block-like" language which can be used to programme the board. To use this keyboard an add-on set of "blocks" are required – full details are provided on my website. Once the blocks are installed, then it is a simple matter of including the "Play Piano" block to start making some noise!

Then there are many options for further experimentation – the keyboard also includes four RGB programmable LEDs and provides access to the Micro:bit's other sensor pins, not forgetting the built-in buttons, 5x5 LED matrix and sensors on the Micro:bit board itself.

Further Reading

- Micro:bit Getting Started Guide - https://microbit.org/get-started/first-steps/introduction/

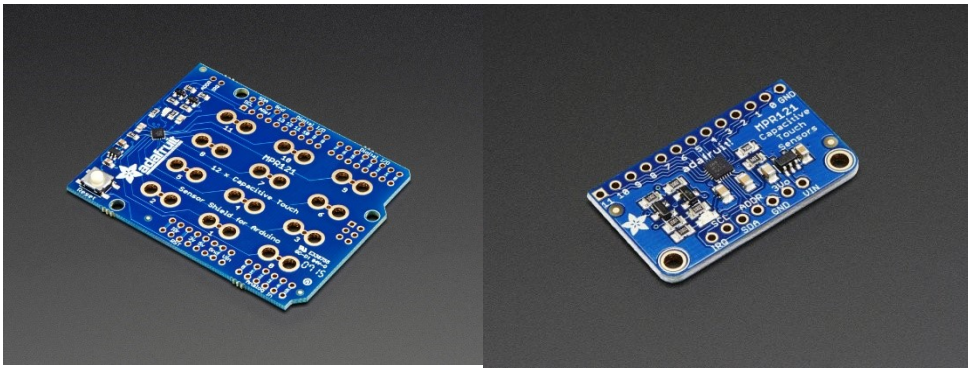
Arduino MPR121 Touch Piano (Project 6)

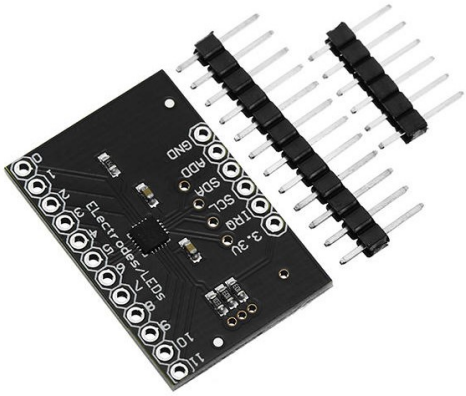
Full details: https://diyelectromusic.wordpress.com/2020/08/25/arduino-mpr121-touch-piano/
Code: https://github.com/diyelectromusic/sdemp/tree/master/src/SDEMP/ArduinoMPR121TouchPiano

Parts List

- Arduino Uno
- MPR121 Based Touch Module, for example:
  - Adafruit 12x Capacitive Touch Shield
  - Adafruit 12-Key Capacitive Touch Sensor Breakout
  - Generic MPR121 sensor breakout
- Conductive "touchable" objects and wires
- 1x 220Ω resistor
- 8Ω Loudspeaker or old headphone speaker

The Circuit





By far the simplest way to get started with touch interfaces with an Arduino is to use the Adafruit 12x Capacitive Touch Shield. This just plugs in and you're ready to go!

If you are using one of the break out modules, then you'll need to follow the instructions to hook them up to the right pins on the Arduino. They use the Arduino's I2C bus support, so need linking up to SDA and SCL (which can be found either broken out independently on more recent Arduino boards, or on A4 and A5 respectively for the Arduino Uno).

The speaker is connected to Arduino pin 12 via a 220Ω resistor and to the Arduino GND. Then it is a simple matter of connecting your touchable surfaces to the input pins on the shield or module and off you go.

Basic Principles

The code uses the Arduino tone() function to generate sounds. Which sound is produced is determined by which input from the MPR121 module is triggered. It uses the Adafruit MPR121 library to handle all the details of talking to the modules.

As the modules use the I2C interface it is possible to connect several modules to a single Arduino which in theory allows you to "play" a wider range of notes. Details of how to do this are provided in the full tutorial (but you may need additional modules and circuitry to support it).

Further Reading

- Adafruit MPR121 Breakout Tutorial - https://learn.adafruit.com/adafruit-mpr121-12-key-capacitive-touch-sensor-breakout-tutorial/overview
- MPR121 Hookup Guide - https://learn.sparkfun.com/tutorials/mpr121-hookup-guide
- Arduino Tone Melody - https://www.arduino.cc/en/Tutorial/toneMelody

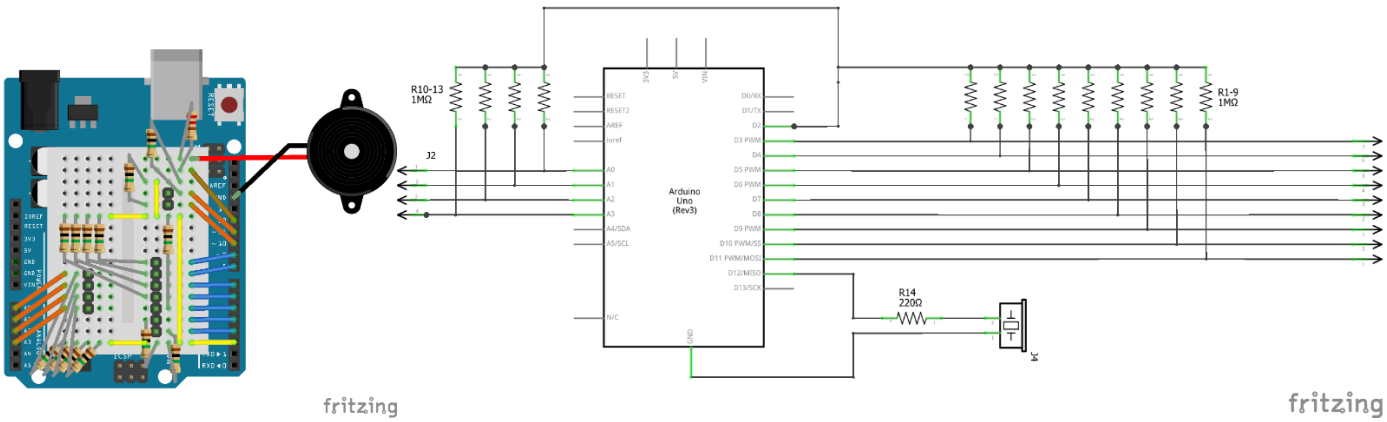
Arduino Touch Piano (Project 7)

Full details: https://diyelectromusic.wordpress.com/2020/06/07/arduino-touch-piano/
Code: https://github.com/diyelectromusic/sdemp/tree/master/src/SDEMP/ArduinoTouchPiano

Parts List

- Arduino Uno
- 13x 1MΩ resistors
- 1x 220Ω resistor
- 8Ω Loudspeaker or old headphone speaker
- Conductive surfaces (e.g. tin foil or copper tape)
- Breadboard and jumper wires

The Circuit



It is possible to do the "touch thing" without extra boards, just using the digital pins of an Arduino. This circuit illustrates the principles.

Basic Principles

The basic idea is that a signal is sent out of one Arduino pin and detected at another. The time it takes for the signal to propagate from the sender to the receiver depends on a resistor and capacitor in the circuit between the two where the main contribution to the capacitance comes from the capacitance of the human body. In this way it is possible to detect proximity to or the touching of a sensor attached to the receiving pin.

Thankfully all this is hidden behind the Capacitive Sensor library for the Arduino.

In our case we use the sensed signal as indication of a "key" on a music "keyboard" and use it to set the frequency for an Arduino tone which is output on pin 12 (via the 220Ω resistor).

As a more advanced activity, you could build the circuit onto a shield and create a music keyboard out of copper tape or foil to connect to the sensors. This is described in "Arduino Touch Piano – Part 2".

Another extension would be to use it as a MIDI controller and generate MIDI note messages rather than Arduino tones. This is described in "Arduino MIDI Touch Piano".

Further Reading

- Arduino Tone Melody - https://www.arduino.cc/en/Tutorial/toneMelody
- Arduino Capacitive Sensor - https://playground.arduino.cc/Main/CapacitiveSensor/

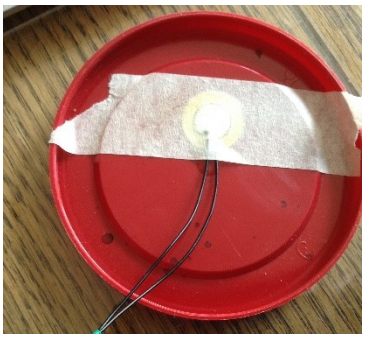
Arduino Piezo MIDI Controller (Project 8)

This is a repeat of the same project listed in worksheet WS001.

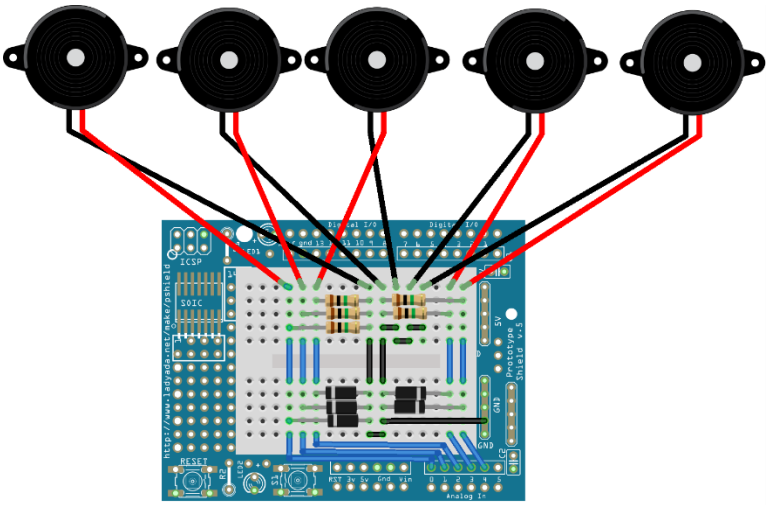
Full details: https://diyelectromusic.wordpress.com/2020/07/19/arduino-piezo-midi-controller/
Code: https://github.com/diyelectromusic/sdemp/tree/master/src/SDEMP/ArduinoPiezoMIDIController

Parts List

- Arduino Uno
- 5x 1MΩ resistors (brown-black-green)
- 5x 1N4733A Zener diodes (5.1V)
- 5x piezo disk sensors and "drum pads" (plastic lids)
- 5-pin DIN plug or socket
- Breadboard and jumper wires



The Circuit



Basic Principles

This project is a little more complex and will require a proper reading of the full description. The basic idea is that the Arduino is watching five piezo sensors. These sensors will generate a voltage if tapped or hit so they make a great basis for an electronic drumkit. In fact these are exactly the type of sensors you'll find in the types of drumkits available for playing games. Depending on which sensor is hit the Arduino will send out a MIDI note message on channel 10, which for any typical "general MIDI" setup is the percussion channel.

The diagram above made this on a "proto shield" – an Arduino shield designed to take a breadboard and let you easily connect it to the Arduino's pins without extra wires. It can be made just as easily using a normal breadboard too.

The MIDI "out" circuitry isn't shown, but it can use the same "two resistors in a DIN plug" MIDI out circuitry we've used several times already.

Use is fiddly again, with a calibration stage required to get the sensitivity of the sensors right. And once again that sensitivity is dependent on the physical design of your sensors. For a really good example of a quality set of home-made drum sensors, I recommend "Marco's DIY kit mentioned below.

Future Reading

- Arduino Knock - https://www.arduino.cc/en/Tutorial/Knock
- Marco's DIY Kit – https://arduinoplusplus.wordpress.com/2020/05/06/diy-midi-percussion-kit-part-1/

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

ARDUINO
OPEN-SOURCE
COMMUNITY

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

https://shop.pimoroni.com/

https://thepihut.com/

https://hobbytronics.co.uk/

Thanks for playing along.

Kevin