Building a simple MIDI controller

Requirements:

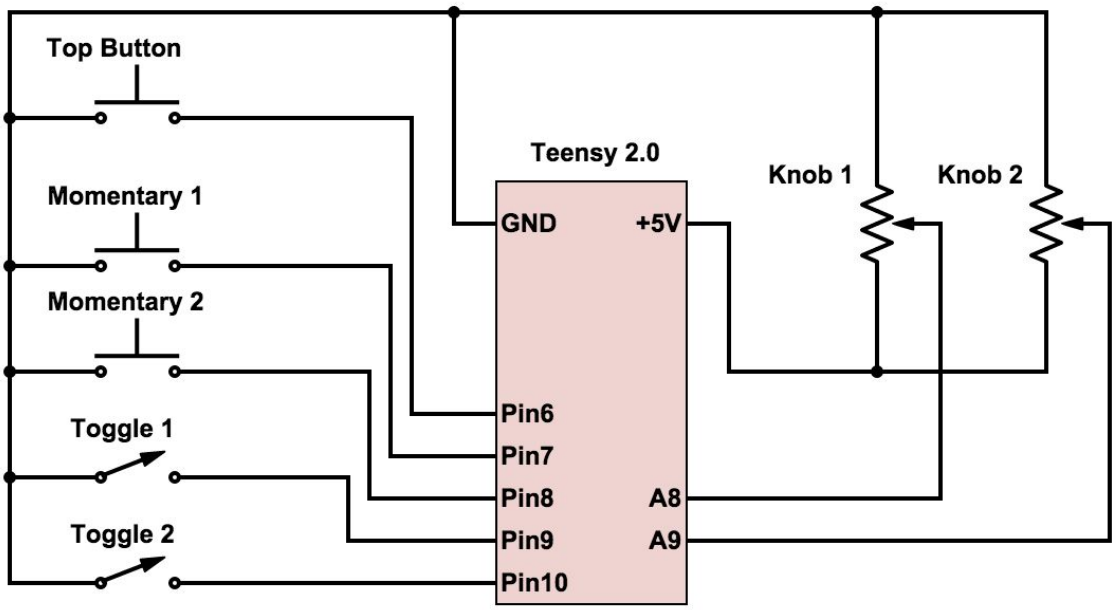
Hardware:

- Teensy microcontroller
    - We used a Teensy 2.0 since it was the most inexpensive. Any Teensy model should work.
- Switches
    - Single Pole Single Throw
    - 3 Pushbutton type - Top button and Momentary buttons
    - 2 Toggle type
- Potentiometers (Knobs)
    - Linear Taper
- Wires, solder etc.
- Enclosure
- Usb cable

Software:

- Arduino software
    - https://www.arduino.cc/en/Main/Software
- Teensyduino - a plugin for Arduino software
    - https://www.pjrc.com/teensy/td_download.html
- Source code
    - https://github.com/soumam/MIDI-controller

Wiring Schematic:

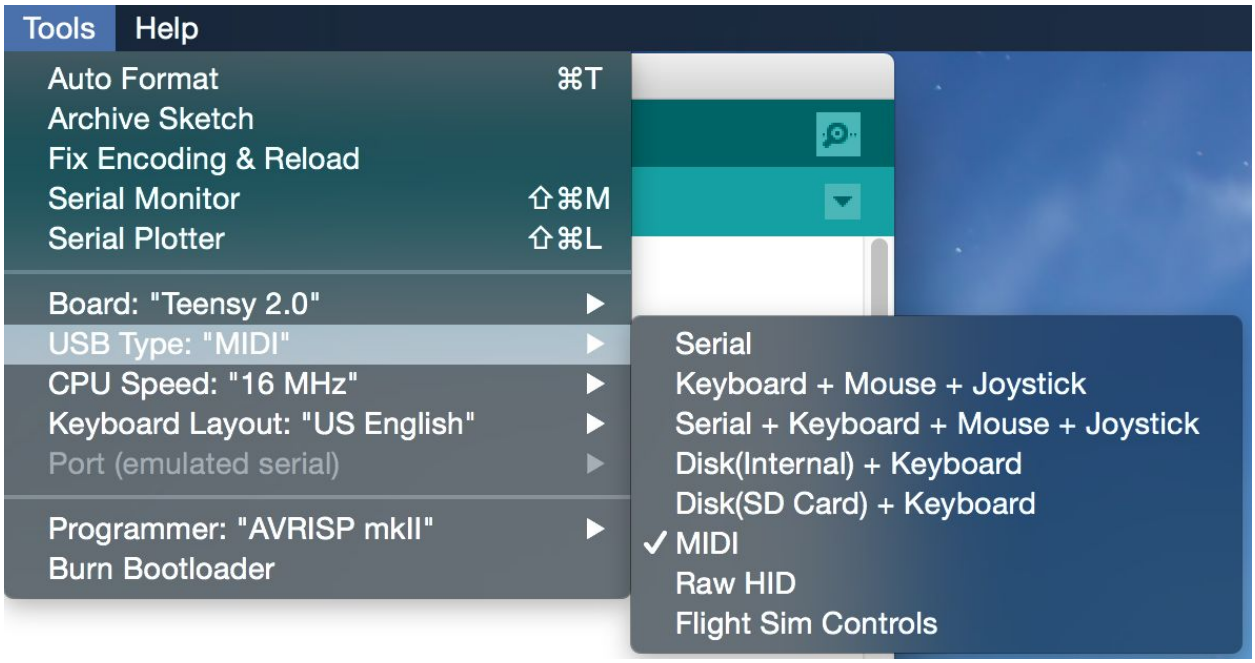


Notes:

- When wiring potentiometers, make sure to not have GND and +5V reversed. Doing so will reverse the direction of the knob. It is best to test first before soldering wires.
- Similarly, check that your toggle switches are oriented in the correct direction.

Programming the Teensy:

- Download/Install Arduino software
- Download/Install Teensyduino
- The Teensy website has excellent tutorials on how to test and program the Teensy microcontroller:
    - https://www.pjrc.com/teensy/first_use.html
- Be sure to set USB Type to “MIDI” to send MIDI data



Notes on the source code:

- For our momentary 1 and 2 buttons we accidentally purchased pushbuttons that will switch from ON to OFF after each press. What we were expecting was pushbuttons that will turn ON upon a press and then OFF upon a release. This is why Momentary Button 1 and 2 have different logic from all the other buttons. We had to recreate a “press and release” action. We implemented this by listening for either a rising or a falling edge and sending a midi ON, waiting for a few cycles, then sending a midi OFF.
- In the likely case that the correct buttons are purchased, simply copy the logic from either the toggle buttons or the top button (they are identical) but replace the variable name of the button with momentary_1 and momentary_2. for example:

```
// Momentary Button 1
if (momentary_1.fallingEdge()) {
  usbMIDI.sendNoteOn(MOMENTARY_1_MIDI_CC, MIDI_NOTE_ON, CHANNEL);
} else if (momentary_1.risingEdge()) {
  usbMIDI.sendNoteOff(MOMENTARY_1_MIDI_CC, MIDI_NOTE_OFF, CHANNEL);
}
```

- All the MIDI related constants can be changed. For simplicity’s sake, it is recommended that CHANNEL remain equal to 1.