Rapidly Prototyping MIDI Controllers with Web MIDI

Max Vogler
Zurich, Switzerland
max.vogler@gmail.com

ABSTRACT

MIDI (Musical Instrument Digital Interface) is the standard communication protocol to connect audio hardware and software. The protocol is commonly used to send messages from hardware MIDI controllers to software that controls music synthesis and playback. Although a variety of hardware and software MIDI controllers exist, they typically use traditional, skeuomorphic input modes like keys, buttons, faders, and knobs. Since 2015, web browsers started supporting this protocol through the Web MIDI API, opening up an enormous untapped potential of integrations.

This talk guides viewers through quickly prototyping web-based MIDI controllers utilizing the power of web browsers and the JavaScript ecosystem. These prototypes showcase innovative ways of controlling audio software by sending MIDI messages from the web browser, e.g. using a phone's touchscreen with pressure detection as 3-d input to control audio effects (see Figure 1).

The talk consists of a series of example prototypes, written in plain JavaScript and running in Chrome. Prior knowledge of JavaScript or the MIDI protocol is not required, but helpful in fully understanding the examples. The first example simply sends a single MIDI message from the browser to the audio software. Then, examples ramp up in interactivity and complexity, e.g. controlling audio effects through machine-learning models. The important parts of each prototype's code fit onto 1-2 slides (similar to Figure 2) and are followed by a live demonstration (similar to Figure 1).

The slides will be written in HTML and JavaScript[1] themselves to allow neatless transitions from static slides to live examples. Every example contains a link to a fully-functional, open-source prototype[2] that viewers can run right in their browser.

The talk aims to excite the audience about:

- Unlocking novel input sources for audio software by using the web browser and web ecosystem.
- Iterating through prototypes quickly by using vanilla JavaScript with high-level APIs and libraries. No IDE or build process is required. Each prototype can be edited and executed right in the web browser without requiring any extension or special software.
- Collaborating with others through open-source and open standards.

[1] Using a framework like reveal.js.
[2] See https://github.com/max-vogler/midi.

Web Audio Conference WAC-2021, July 5–7, 2021, Barcelona, Spain.

1. CONTENTS

To begin, I present a minimal way of requesting access to MIDI outputs and sending the first MIDI message, explaining the usage of `NOTE ON`, pitch, and velocity to instrument the audio software.

I introduce interactive controls, by listening on mouse inputs and triggering `NOTE ON` and `OFF` on mouse click and release respectively.

This leads to the first real prototype, *TOUCH*, controlling two effects' values through the finger's x-y coordinates on a touch screen. Furthermore, the effect's dry-wet knob is controlled through the finger pressure, making the effect more audible when the finger is pressed harder.

Since the laptops that run audio software do not generally have touch screens, I introduce *GATEWAY*, a bridge from the Web MIDI output to other computers through WebRTC. I show how to open a WebRTC connection, send MIDI messages through WebRTC, and peer two computers by displaying a QR code. This can then be used to feed a phone's touchscreen controls (from the previous prototype) to a different computer that runs the audio software.

To showcase more benefits of the JavaScript ecosystem, I show *CONTROLLER*, a translation from video game controller inputs to MIDI messages using the GamePad API.

In the final example, *HANDPOSE*, I feed the webcam video feed through a pre-trained machine-learning model which detects the pose and position of fingers. This allows the user to control effect values by moving their fingers mid-air in front of the laptop's webcam.

2. ACKNOWLEDGMENTS

This talk focusses on prototyping and technology and is not a musical performance. Unless otherwise requested, the MIDI values will only be displayed in the slides through simple visuals, but not control any live music during the talk.

The talk is exclusively based on open-source resources, notably github.com/max-vogler/midi. It is not sponsored by my employer. Both the open-source code repository and the talk are created in my private time.

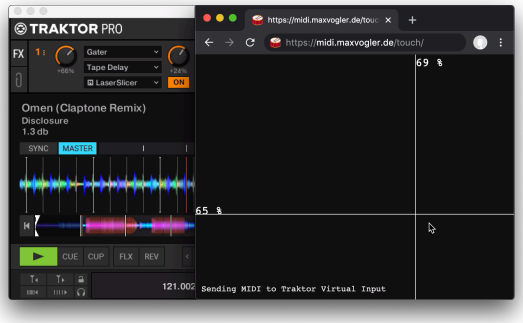

Figure 1: A sample experiment, controlling effect value knobs in Traktor Pro through touch coordinates in the browser.

```
const midi = await navigator.requestMIDIAccess();
const noteOn = (p, v) => midi.outputs[0].send(
  0x90, p, Math.round(v * 127));
document.addEventListener("pointermove",
  ({x, y, pressure}) => {
    noteOn(0, x / document.body.clientWidth);
    noteOn(1, y / document.body.clientHeight);
    noteOn(2, pressure);
  });
});
```

Figure 2: A sample code snippet that translates touch input to MIDI messages.