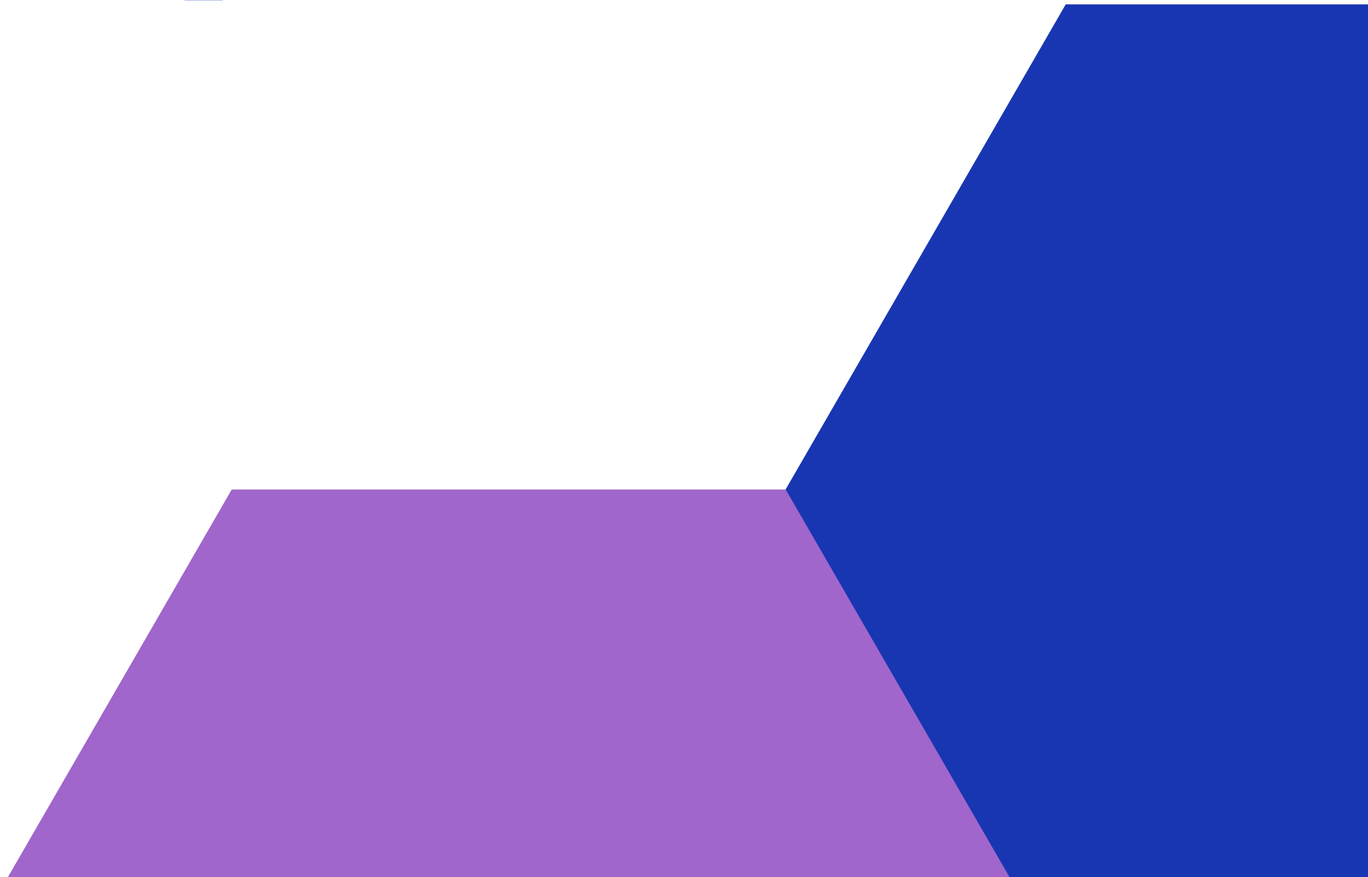
# WEB MIDI API

A primer on MIDI input and output in the browser



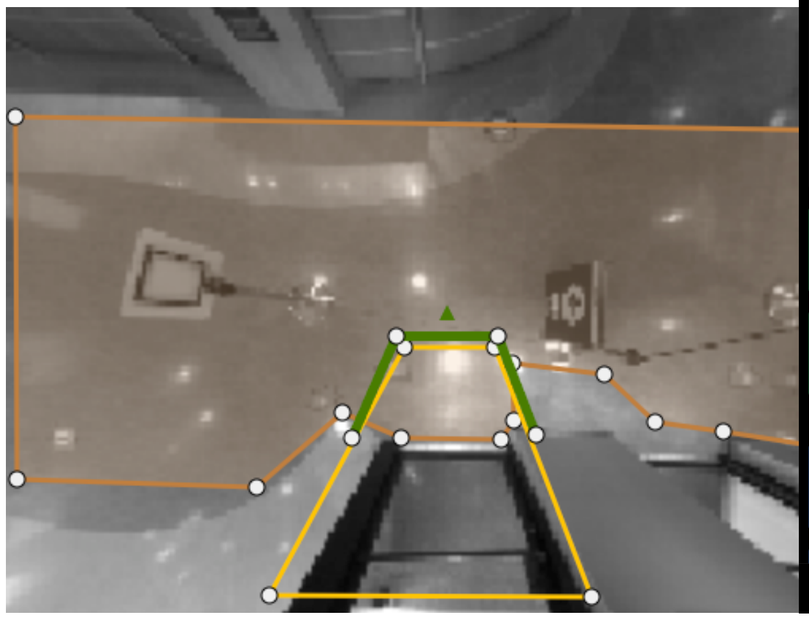
# Introduction

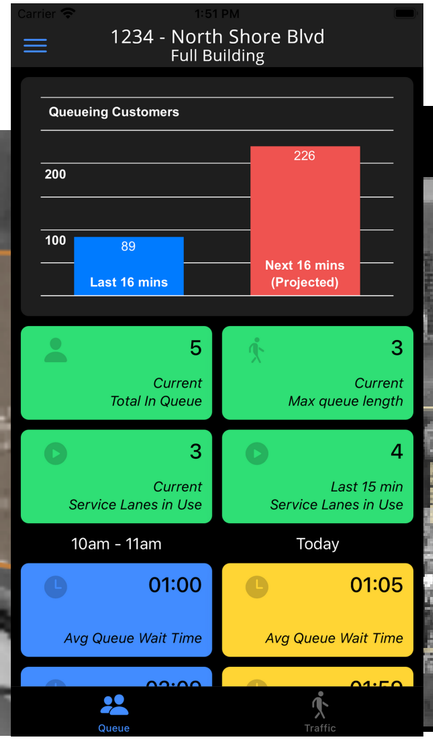

## Jeremy Forsythe

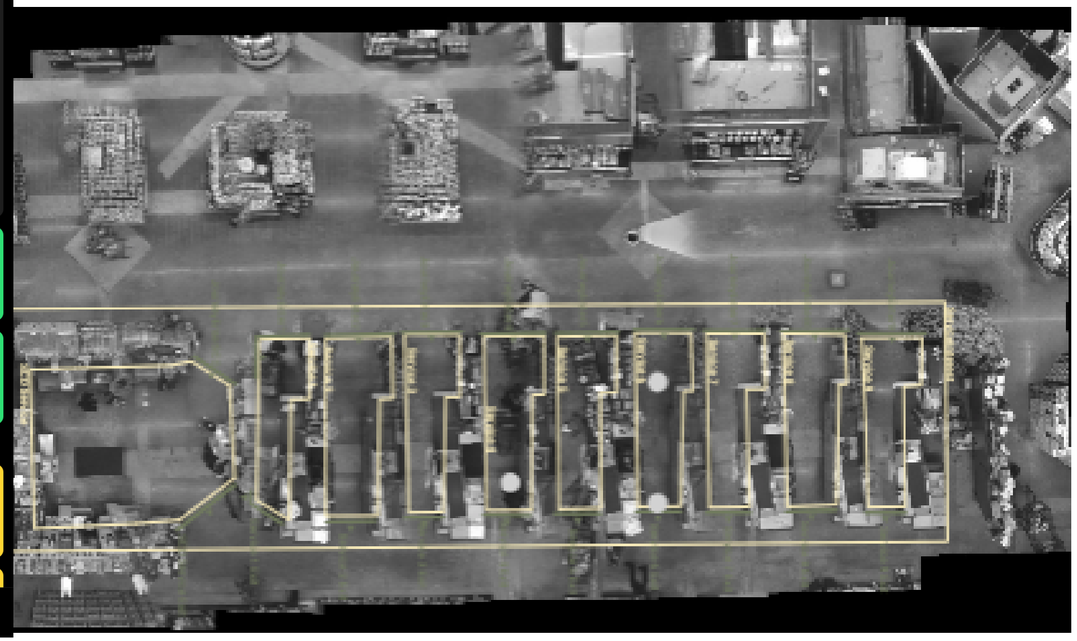
Director of Software Engineering


SENSOURCE

- writing code for 34 years from Basic and Pascal to TypeScript
- mostly TypeScript microservices nowadays
- tooling, DevOps, infrastructure





If you see this, you need to get a Java(tm) enabled browser, like Microsoft Internet Explorer.

Click on the above Java applet or here to hear Today (today.mid).

This page has been accessed times!

Submit a Smashing Pumpkins site for the Bugg Superstar's SP Site Award


Pumpkins Trivia | Pumpkins Chat | Pumpkins Fonts

Pumpkins Pictures | Pumpkins Thumbnail Pictures | Pumpkins Sounds


Last updated: May 8, 1997

Please find out why this site is not very fancy

If you see this, you need to get a Java(tm) enabled browser, like Microsoft Internet Explorer.

Click on the above Java applet or here to hear Today (today.mid).

This page has been accessed times!

Submit a Smashing Pumpkins site for the Bugg Superstar's SP Site Award


Pumpkins Trivia | Pumpkins Chat | Pumpkins Fonts

Pumpkins Pictures | Pumpkins Thumbnail Pictures | Pumpkins Sounds

# What is MIDI?

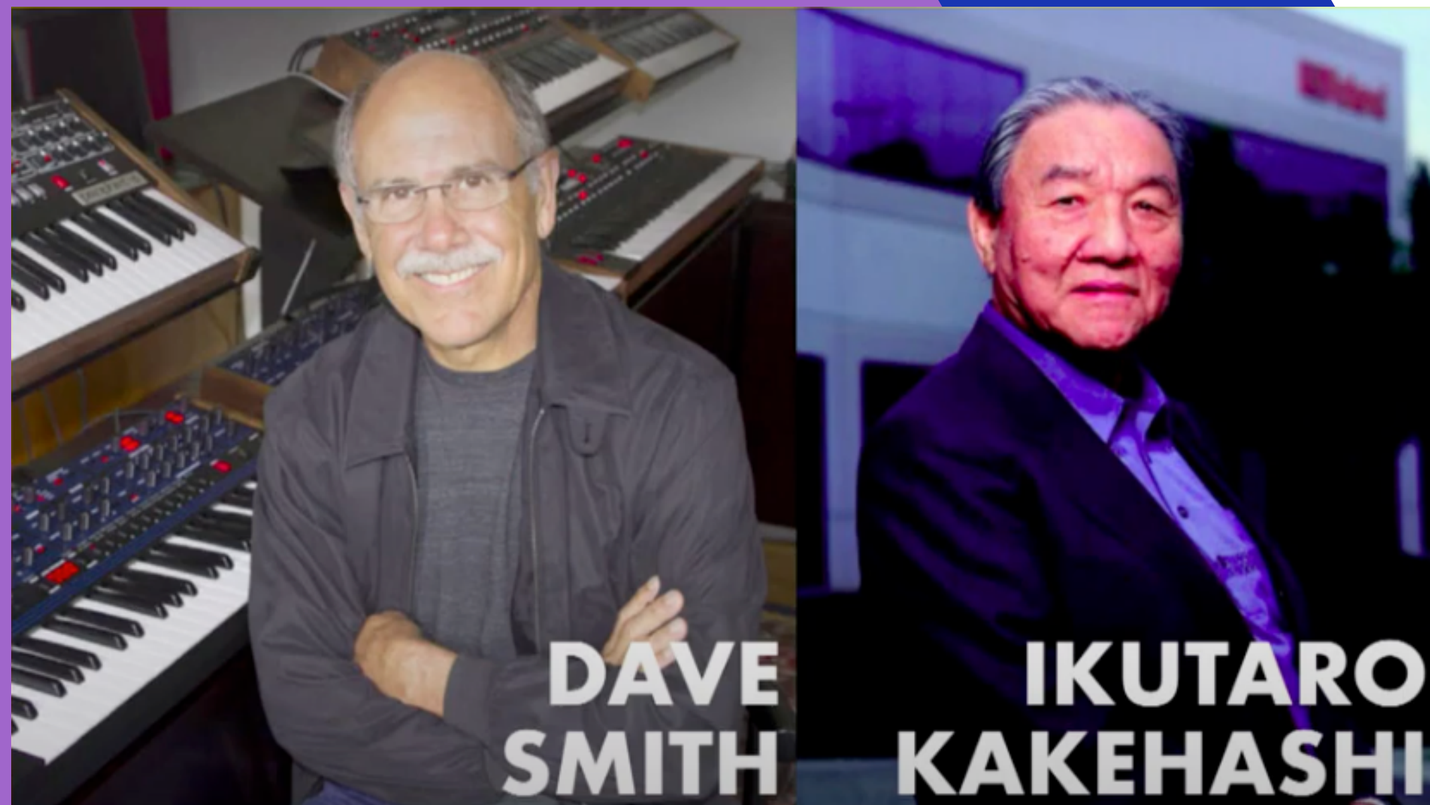
Musical Instrument Digital Interface

A technical standard describing a *communications protocol*, *digital interface*, and *electrical connectors*

# What is MIDI?

**Early commercial electronic instruments**

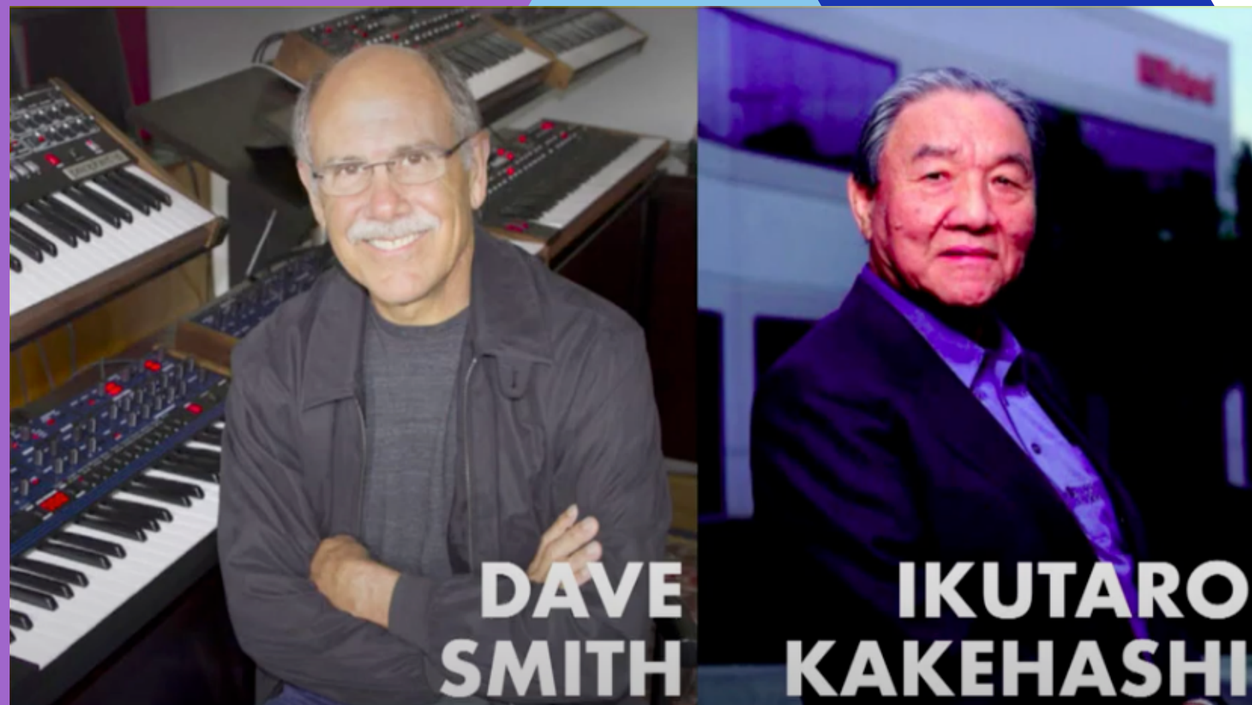
- 1977 Roland MC-8 - Ralph Dyke - Microprocessor sequencer
- 1978 Sequential Circuits Prophet 5 - Dave Smith - Microprocessor synthesizer
- 1979 Oberheim Electronics - Tom Oberheim - first instruments talking over custom protocol



# What is MIDI?

## Standard

- First described by Dave Smith and Chet Wood of Sequential Circuits in October 1981 at *Audio Engineering Society*
- Standardized in 1983 and 1.0 published in 1985
- Owned by MIDI Manufacturers Assn (MMA) and Association of Musical Electronics Industry (AMEI)
- 2.0 introduced in 2020 (Content negotiation / JSON)



# What is MIDI?

5-pin DIN

## History

- First devices: Roland Jupiter-6 ('82), Prophet 600 ('82), Roland TR-909 ('83), Roland MSQ-700 ('83)
- First MIDI-supporting computer, NEC PC-88, in 1982
- General MIDI song files added to standard in 1991
- USB/Firewire compatible MIDI in 1999
- Almost all music you hear today went through at least one MIDI interface

# What is MIDI?

**Protocol is old**

1971 - FTP

1974 - TCP

**1981 - MIDI**

1982 - Ethernet

1991 - HTTP

1995 - SSH

2001 - BitTorrent

2009 - BitCoin

2015 - HTTP/2

# MIDI Protocol

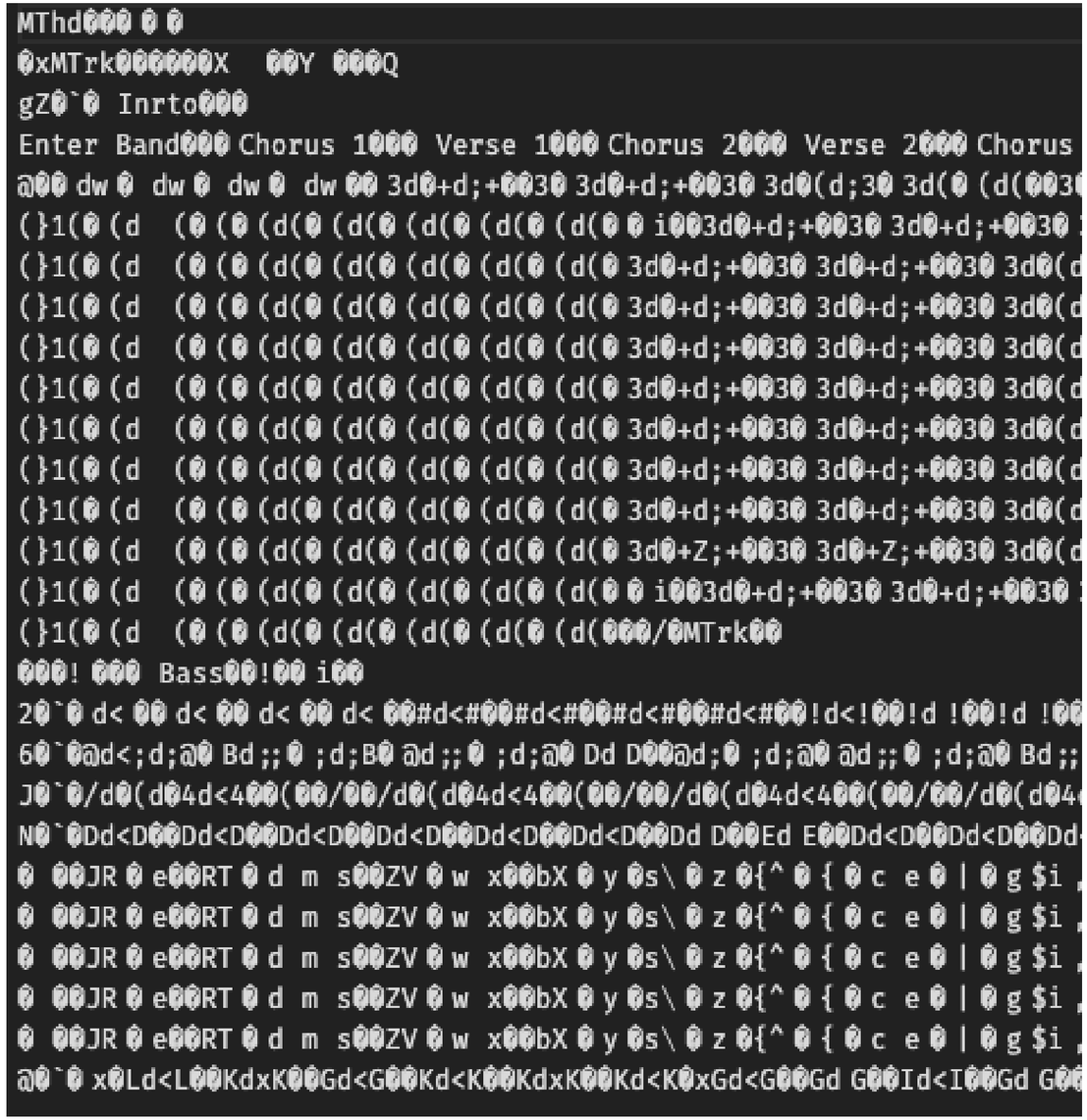
- MIDI is a serial, event-based messaging protocol

- Does not transmit audio signals but instead events that cause the receiving device to perform an action, like playing a sound

- Brief numeric descriptions of an action (pressing keys, turning knobs, wiggling joysticks, etc.)

- 16 channels per MIDI device, one instrument per channel

- Designed to carry messages like notation, pitch, velocity, on/off



# MIDI Protocol

- MIDI is a serial, event-based messaging protocol

- Does not transmit audio signals but instead events that cause the receiving device to perform an action, like playing a sound

- Brief numeric descriptions of an action (pressing keys, turning knobs, wiggling joysticks, etc.)

- 16 channels per MIDI device, one instrument per channel

- Designed to carry messages like notation, pitch, velocity, on/off

```
0, 0, Header, 1, 10, 120
1, 0, Start_track
1, 0, Time_signature, 4, 2, 24, 8
1, 0, Key_signature, 4, "major"
1, 0, Tempo, 681818
1, 480, Marker_t, "Inrto"
1, 2400, Marker_t, "Enter Band"
1, 6240, Marker_t, "Chorus 1"
1, 10080, Marker_t, "Verse 1"
1, 13920, Marker_t, "Chorus 2"
1, 17760, Marker_t, "Verse 2"
1, 21600, Marker_t, "Chorus 3"
1, 25920, Marker_t, "Verse 3"
1, 29280, Marker_t, "Chorus 4"
1, 29280, End_track
2, 0, Start_track
2, 0, MIDI_port, 0
2, 0, Title_t, "Drums"
2, 0, Program_c, 9, 0
2, 0, Control_c, 9, 7, 105
2, 0, Control_c, 9, 10, 64
2, 0, Note_on_c, 9, 31, 100
2, 119, Note_on_c, 9, 31, 0
2, 120, Note_on_c, 9, 31, 100
2, 239, Note_on_c, 9, 31, 0
2, 240, Note_on_c, 9, 31, 100
2, 359, Note_on_c, 9, 31, 0
2, 360, Note_on_c, 9, 31, 100
2, 479, Note_on_c, 9, 31, 0
2, 2400, Note_on_c, 9, 51, 100
2, 2400, Note_on_c, 9, 43, 100
2, 2459, Note_on_c, 9, 43, 0
2, 2459, Note_on_c, 9, 51, 0
2, 2460, Note_on_c, 9, 51, 100
2, 2460, Note_on_c, 9, 43, 100
2, 2519, Note_on_c, 9, 43, 0
2, 2519, Note_on_c, 9, 51, 0
2, 2520, Note_on_c, 9, 51, 100
2, 2520, Note_on_c, 9, 40, 100
2, 2579, Note_on_c, 9, 51, 0
2, 2580, Note_on_c, 9, 51, 100
2, 2609, Note_on_c, 9, 40, 0
```

0x80 (128) - Note off

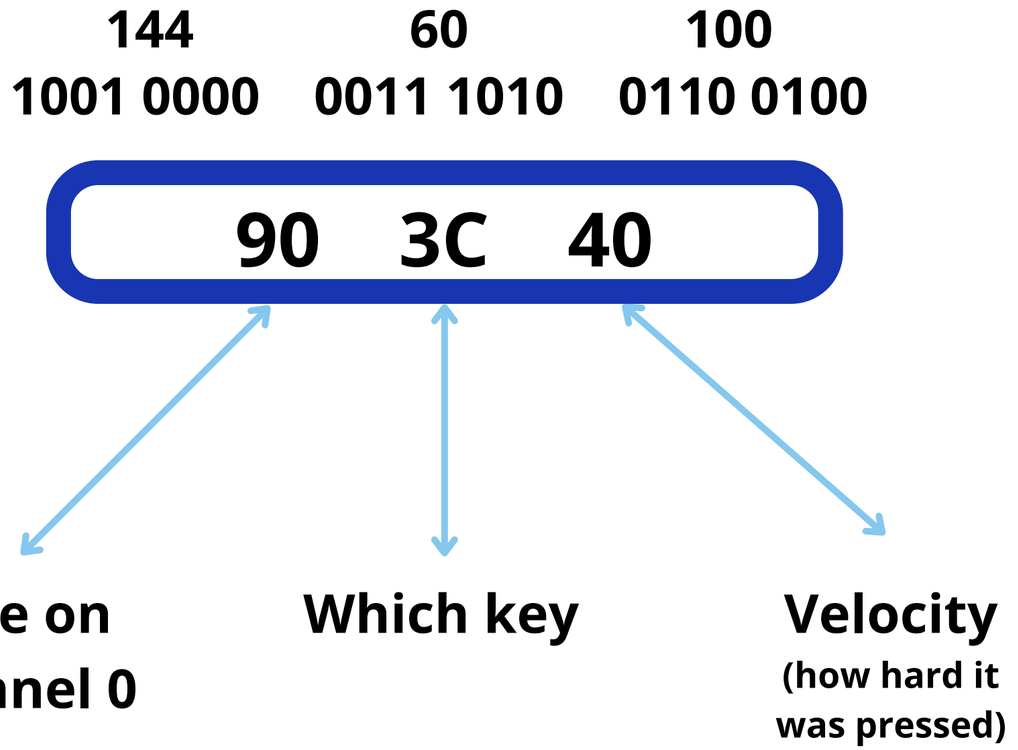
0x90 (144) - Note on

# MIDI Protocol

## General MIDI

**128 Notes (~10 octaves)**
**16 channels**
**128 "programs" (instrument sounds)**

1. **Acoustic Grand Piano**
2. **Bright Acoustic Piano**
3. **Electric Grand Piano**
4. **Honky-tonk Piano**

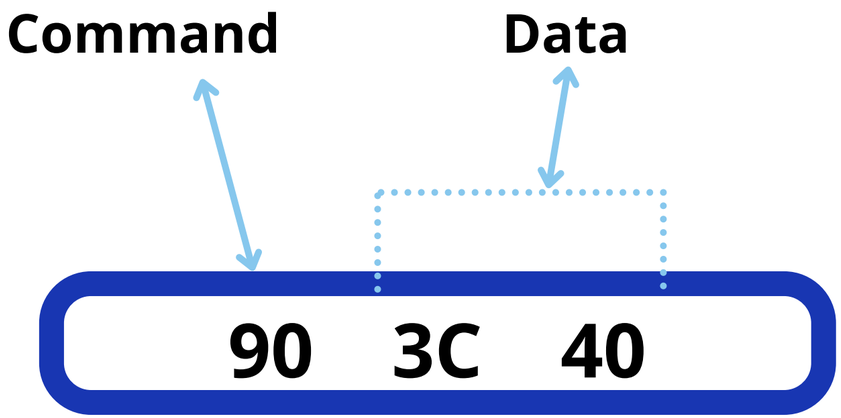



0x**9**_ **1001** 0000 Note on
0x_**0** 1001 **0000** Channel 0

# MIDI Protocol

The original intention was for audio devices, but of course it's just a protocol and can be used to send anything we can fit in the packet

We can also *interpret* the messages and perform any action we want on the receiving side



# Web MIDI API

Working Draft
https://webaudio.github.io/web-midi-api/

# Web MIDI API

Working Draft
https://webaudio.github.io/web-midi-api/

"The Web MIDI API specification defines a means for web developers to enumerate, manipulate, and access MIDI devices"

# Web MIDI API

Working Draft
https://webaudio.github.io/web-midi-api/

It's simply an API for the protocol

# Can I Use?

## Web MIDI API - WD

Usage: % of all users ? — Global 74.69%

The Web MIDI API specification defines a means for web developers to enumerate, manipulate and access MIDI devices

Current aligned | Usage relative | Date relative | Filtered | All

| IE | Edge * | Firefox | Chrome | Safari | Opera | Safari on iOS * | Opera Mini * | Android Browser * | Opera Mobile * | Chrome for Android | Firefox for Android | UC Browser for Android | Samsung Internet |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 12-18 | | 4-42 | | 10-29 | | | | | | | | |
| 6-10 | 79-94 | 2-92 | 43-94 | 3.1-14.1 | 30-79 | 3.2-14.8 | | 2.1-4.4.4 | 12-12.1 | | | | 4-14.0 |
| 11 | 95 | 93 | 95 | 15 | 80 | 15 | all | 94 | 64 | 94 | 92 | 12.12 | 15.0 |
| | | 94-95 | 96-98 | TP | | | | | | | | | |

- **No official Firefox or Safari support**
- **jzz (npm install jzz) allows support in all major browsers and Node.JS**
- **only available in *secure contexts (https, localhost, not iframe)***

# Hello, World

```
navigator.requestMIDIAccess().then(
  (midiAccess) => {
    console.log('Got MIDI access');
  },
  (errMsg) => console.log(`Failed to get MIDI access - ${msg}`)
)
```

```
Got MIDI access ▼MIDIAccess {inputs: MIDIInputMap, ou
                  ▶ inputs: MIDIInputMap {size: 1}
                    onstatechange: null
                  ▶ outputs: MIDIOutputMap {size: 1}
                    sysexEnabled: false
                  ▶ [[Prototype]]: MIDIAccess
```

# Web MIDI API

```
Got MIDI access ▼MIDIAccess {inputs: MIDIInputMap, ou
                  ▶ inputs: MIDIInputMap {size: 1}
                    onstatechange: null
                  ▶ outputs: MIDIOutputMap {size: 1}
                    sysexEnabled: false
                  ▶ [[Prototype]]: MIDIAccess
```

**.inputs** and **.outputs** are **Maps** so calling **.input.values()** returns an *iterator*

for demo simplicity, we assume *one* attached device, so we just use **.inputs.values().next().value** to get the first device

# MIDI Inputs

```
navigator.requestMIDIAccess().then((midiAccess) => {
  const input = midiAccess.inputs.values().next().value;

  console.log(input);
});
```

```
▼MIDIInput {onmidimessage: null, connec
    connection: "closed"
    id: "1392079709"
    manufacturer: "Focusrite A.E. Ltd"
    name: "Launchpad Mini"
    onmidimessage: null
    onstatechange: null
    state: "connected"
    type: "input"
    version: "2"
  ▶[[Prototype]]: MIDIInput
```

# MIDI Input Events

```
navigator.requestMIDIAccess().then((midiAccess) =>
  const input = midiAccess.inputs.values().next().v

  input.onmidimessage = console.log;
});
```

```
▼MIDIMessageEvent {isTrustec
    bubbles: true
    cancelBubble: false
    cancelable: false
    composed: false
  ▶ currentTarget: MIDIInput
  ▶ data: Uint8Array(3) [144,
    defaultPrevented: false
    eventPhase: 0
    isTrusted: true
  ▶ path: []
    returnValue: true
  ▶ srcElement: MIDIInput {co
  ▶ target: MIDIInput {connec
    timeStamp: 1800.099999904
    type: "midimessage"
  ▶ [[Prototype]]: MIDIMessag
```

# MIDI Input Events

```
navigator.requestMIDIAccess().then((midiAccess) => {
  const input = midiAccess.inputs.values().next().value;

  input.onmidimessage = console.log;
});
```

```
▼MIDIMessageEvent {isTrusted
  bubbles: true
  cancelBubble: false
  cancelable: false
  composed: false
▶ currentTarget: MIDIInput
▶ data: Uint8Array(3) [144,
  defaultPrevented: false
```

## Down

```
▼data: Uint8Array(3)
   0: 144
   1: 0
   2: 127
```

Command (Button Press / Note on) 0x90

Note number 0x00

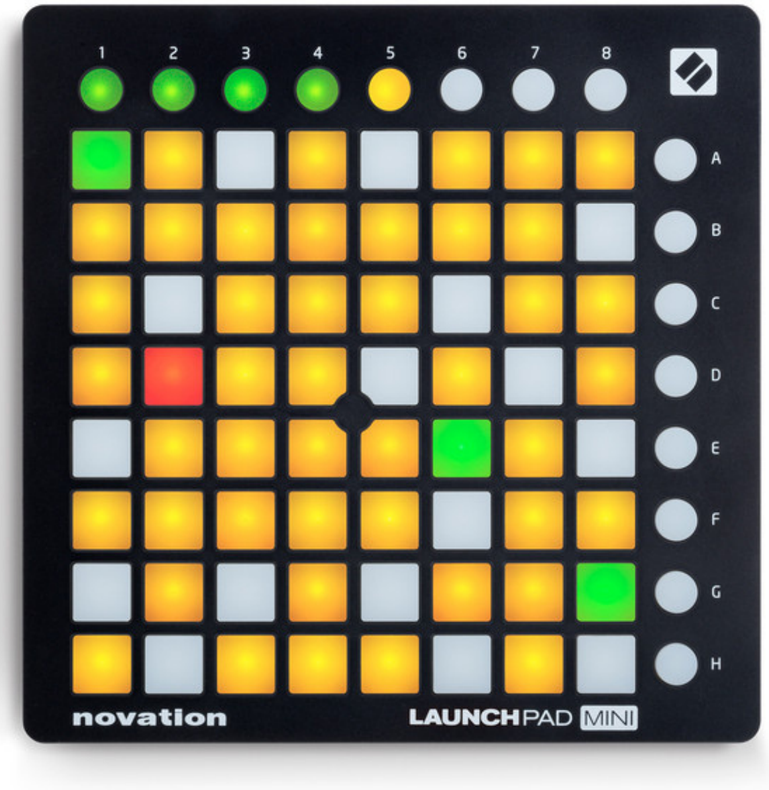
Velocity (0x7f vs 0x00)

Some devices use velocity 0 instead of command 0x80 (note off)

## Up

```
▼data: Uint8Array(3)
   0: 144
   1: 0
   2: 0
```

# Novation MK2 LaunchPad MINI




- USB MIDI controller (input device)
- Created for audio - e.g. Ableton / Fruity Loops
- 8x8 R/G backlit "pads" and 16 R/G backlit buttons (not much difference as far as MIDI is concerned)

# Novation MK2 LaunchPad MINI

Top buttons give command **0xb0**
(continuous controller)

Buttons are numbered:
**0x68 - 0x6f**

Pad area and right buttons give command **0x90**

Buttons are numbered:
**0x00 - 0x08**
**0x10 - 0x18**
**...**
**0x70 - 0x78**

# MIDI Demo 1

(Input) Button Logging

# Novation MK2 LaunchPad MINI

Also a MIDI output device - can set the colors and brightness of the pads/buttons with MIDI commands

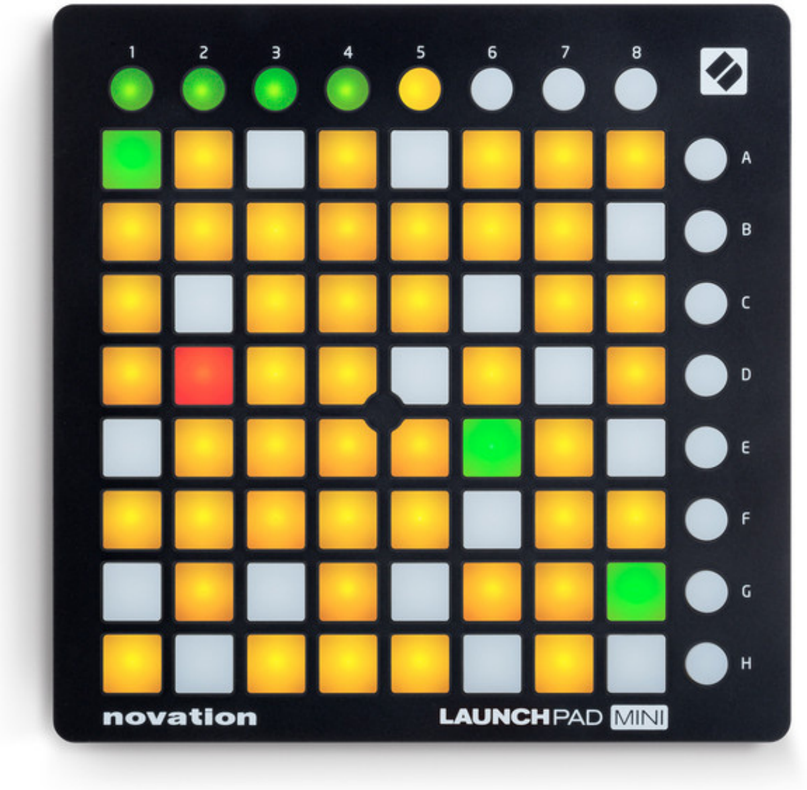
To change color:

[command, note, velocity]

Command is same as input (0xb0 or 0x90)
Note is same as input (0x68-0x6f or 0x00-0x78)
Color is based on some bit math:

```
(
  0b10000 * green +       // 0 - 3
  0b01000 * (0 or 1) +    // clear buffer bit
  0b00100 * (0 or 1) +    // copy to buffer bit
  0b00001 * red           // 0 - 3
)
```

Clear all with **[0xb0, 0x00, 0x00]**



# MIDI Demos 2 & 3

(Output) Marquee and Game of Life

# MIDI Demo 4

(Input) Web MIDI API + Web Audio API

# MIDI Demo 5

(Input) Web MIDI API + Web Audio API
+ ThreeJS Visualization

# MIDI Demo 6 & 7

(I/O) Snake, p5.js, dual controllers

# MIDI Demo 8

Snake + Custom Midi Controller

*Arduino and cannibalized parts*

# Custom MIDI Controller

## Arduino

Custom firmware - mocoLUFA - dual boot Arduino firmware for true USB-MIDI compliant, plug-and-play device - based off the LUFA framework

Default boot into MIDI mode

Connect ICSP 4&6 to boot in "normal" mode (to upload sketch)

Uses Francois Best's Midi Library (midi.h)

Simple sketch to convert button presses to MIDI messages

## Input devices

Cannibalized 6mmx6mm tactile push button switches from the remote for an old, broken drone

Button caps and enclosure 3d-printed with UV curing resin (Elegoo Mars 2 Pro)

# MIDI Demo 9

## Robot Controller

# Sphero Bolt - Educational Ball Robot

## Coding Robot for Kids

Uses a Scratch-like language or JavaScript to program the robot in the official apps

Official JS SDK is archived and didn't support the Bolt model

**spherov2** NPM package allows pairing and communicating

MIDI commands can be interpreted to send directions to the robot

# What's it good for?

The API is very simple, but why would you want to use it?

- **A/V applications in the browser (or any other app that could use physical buttons) - Digital Audio Workstations, Video Editors, etc.**
- **Custom macro pads - a la StreamDeck**
- **Accessibility for web apps - give usability to those with a hard time typing**
- **Custom controllers (Arduino / Pi Pico / etc.)**
- **Playing with robot balls and other fun things**
- **Sending sensor data or any other kind of data**

# Web MIDI API

https://github.com/jdforsythe/web-midi-tests

https://webaudio.github.io/web-midi-api/
https://developer.mozilla.org/en-US/docs/Web/API/Web_MIDI_API
https://www.npmjs.com/package/spherov2



jdforsythe


jdforsythe@gmail.com


@Jeremy Forsythe