



# AudioとガジェットをWebで遊ぶ

## - Web Audio/MIDI, Web Bluetooth -

Yamaha Corporation of America  
Ryoya Kawai



Yamaha Corporation of America

# Ryoya Kawai



## 得意技

- Web Audio/MIDI API
- Polymer (Web Components)
- Web Bluetooth はじめました

## HTML5な活動

- HTML5Experts.jp #55
- Web Music Developers JP 管理人



[@ryoyakawai](https://twitter.com/@ryoyakawai)



[google.com/+RyoyaKawai](https://plus.google.com/+RyoyaKawai)



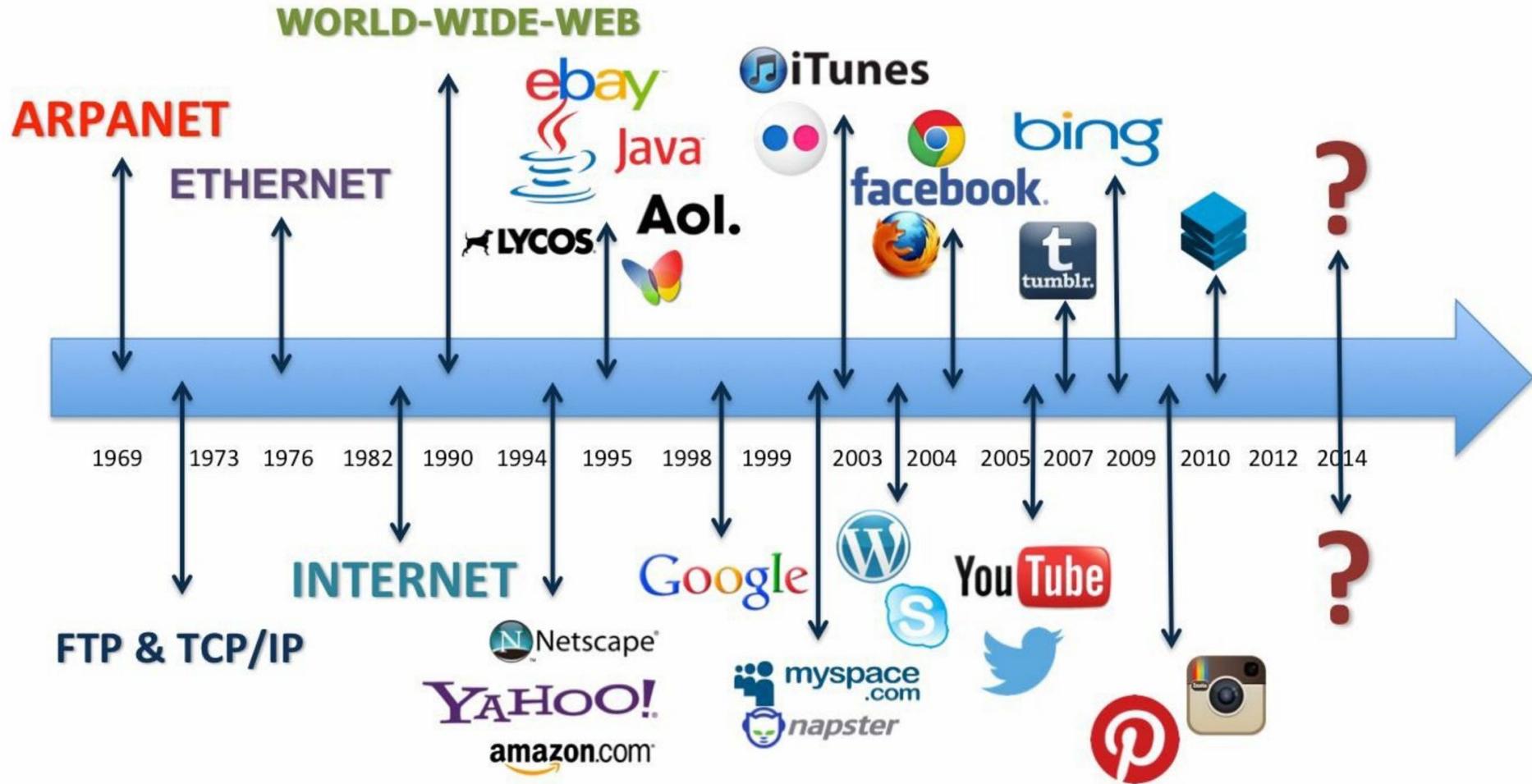
みでやっぴー by @g200kg  
(W3C Unofficial Web MIDI API Mascot)

本日のコードはこちら：



<https://goo.gl/kBoRzZ>

# 25年の変遷

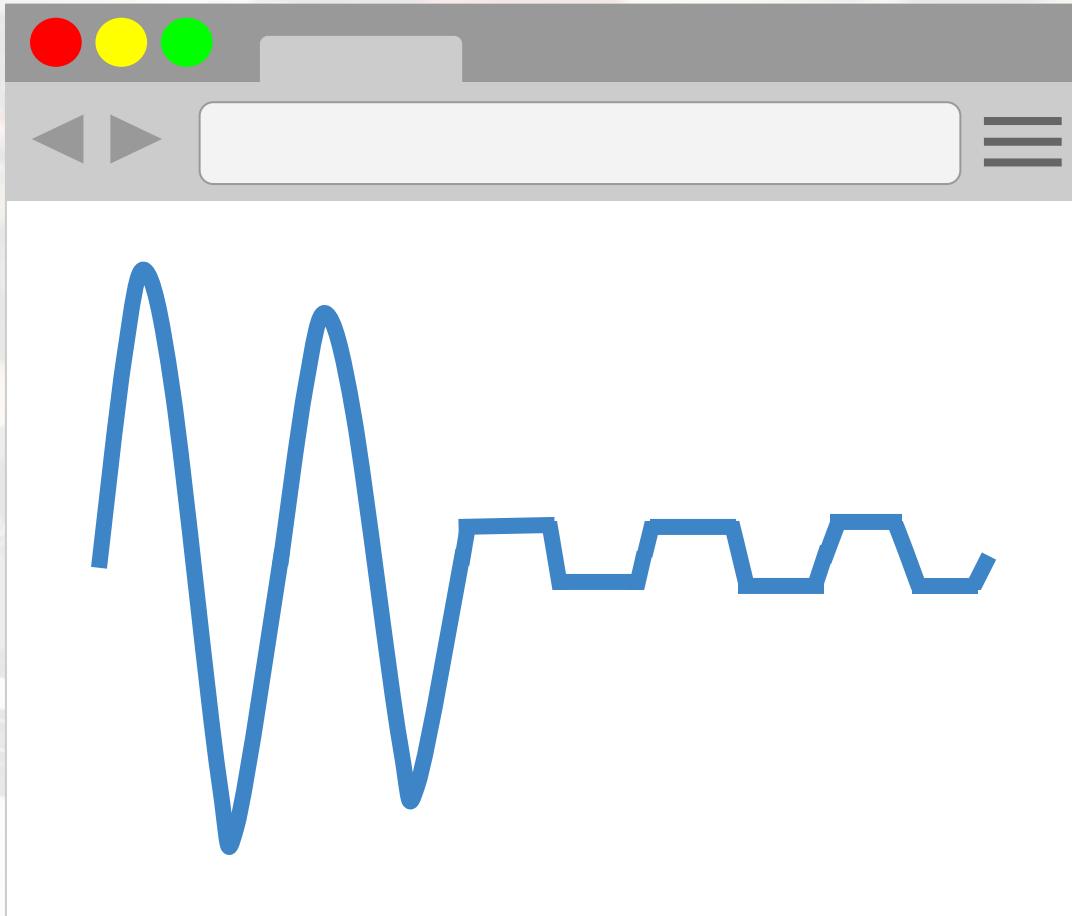


# **Sound!**

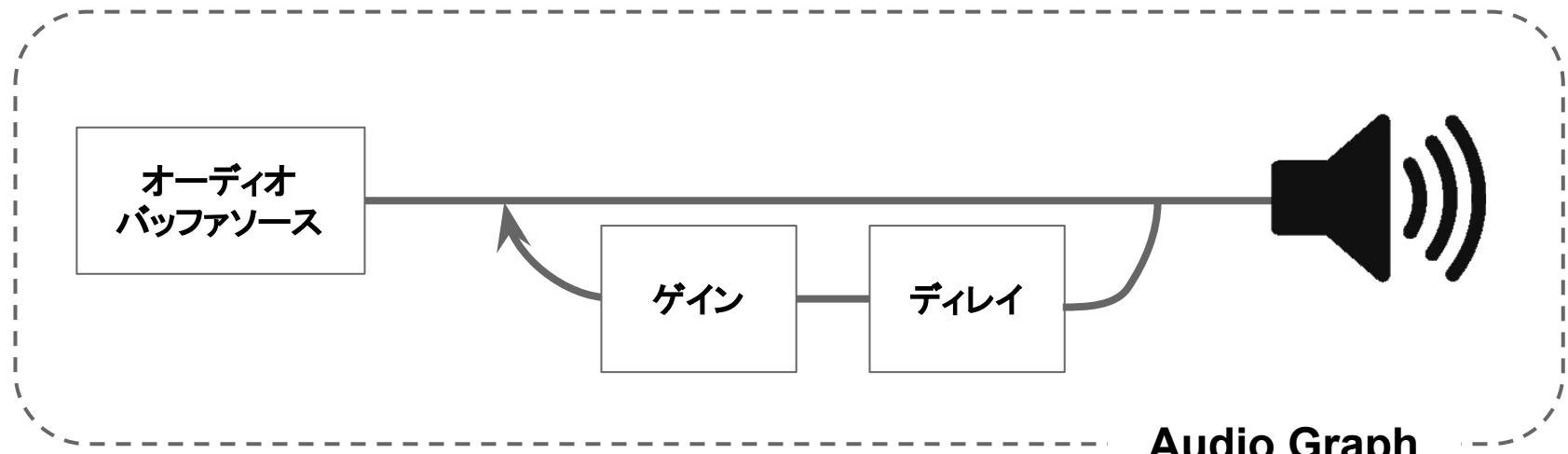
# Web Audio API



# Webブラウザ上で 信号処理を可能にした API



# Audio Graph



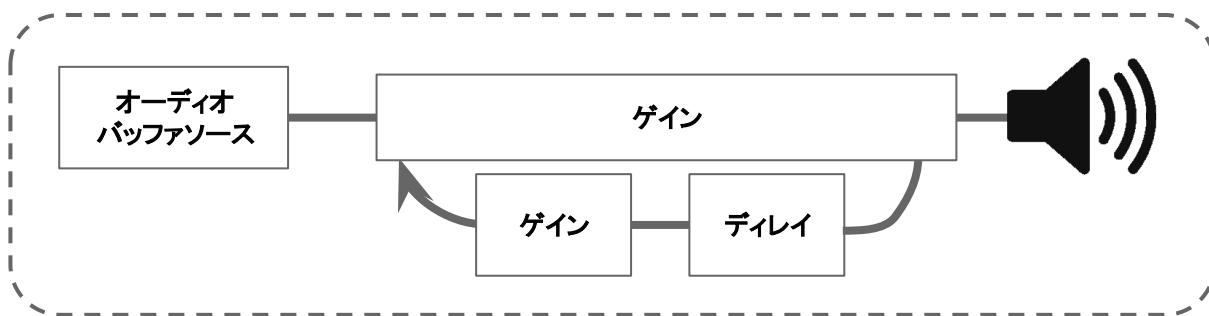
予め用意されているNode(最小単位)

- オシレーター
- オーディオバッファソース
- ゲイン
- フィルター
- ディレイ
- パン
- コンプレッサー
- コンボルバー
- アナライザー
- ウェーブシェイパー

• **AudioWorklet**

(現: ScriptProcessor)

# Delay



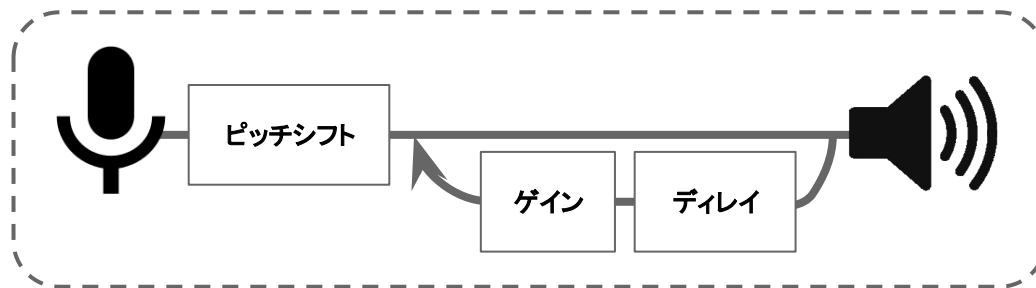
```
<button onclick="Play()" id="playsound" disabled>Play</button>
<script type="text/javascript">
var audioctx = new AudioContext();
var buffer = null;
LoadSample(audioctx, "./snare.wav");
function Play() {
    var src = audioctx.createBufferSource();
    src.buffer = buffer;
    gain = audioctx.createGain();
    var delay = audioctx.createDelay();
    var gainD = audioctx.createGain();
    delay.delayTime.value=0.2;
    gainD.gain.value=0.5;

    src.connect(gain).connect(audioctx.destination);
    gain.connect(delay).connect(gainD).connect(gain);

    src.start(0);
}
```

Demo : [/webaudiomidi/delay/](http://webaudiomidi/delay/)

# Pitch Shift



```
var ctx = new AudioContext();
var mic, ps, delaySw, localStream;
var micout=ctx.createGain();
var psin=ctx.createGain(), psout=ctx.createGain();
var psdry=ctx.createGain(), delaydry=ctx.createGain();
var delayin=ctx.createGain(), delayout=ctx.createGain();
var intersection0=ctx.createGain();
var type=false; zPHMJ

var pshift=new pitchShift(ctx);
var delay=new delayProcess(ctx);

psin.connect(pshift.getSrc()).connect(psout)
  .connect(intersection0).connect(delayin)
  .connect(delay.getSrc());

delay.connect(delayout);
delayout.connect(ctx.destination);

psin.connect(psdry).connect(intersection0)
  .connect(delaydry).connect(ctx.destination);
```

[Demo](#) : /webaudiomidi/pitch/js/app.js

# Pitch Shift

```
var pitchShift=function(ctx) {
    this.ctx=ctx;
    this.rate = 2.0;

    uprice this.stream_length = 4096;
    this.pitchShifter=this.ctx.createScriptProcessor(this.stream_length, 1, 1);
    this.pitchShifter.onaudioprocess=this.onaudioprocess.bind(this);
    this.fft = new FFT(this.stream_length, this.ctx.sampleRate);
    this.hann = new WindowFunction(DSP.HANN);
    this.a_real = new Array(this.stream_length);
    this.a_imag = new Array(this.stream_length);

};

pitchShift.prototype={
    getSrc: function() {
        return this.pitchShifter;
    },
    connect:function(node) {
        this.pitchShifter.connect(node);
    },
    off: function() {
        to create and this.rate=1;
    },
    togglePitch: function() {
        if(this.rate==0) this.rate=0.7;
        this.rate = (2.0 + 0.7) - this.rate;
    },
    pshift: function(val,apindata) {
        // (.. Processing Pitch Shift ..)
    },
    onaudioprocess: function(event) {
        var sin = event.inputBuffer.getChannelData(0);
        var sout = event.outputBuffer.getChannelData(0);
        // 1.0:normal 2.0:oct up 0.5:oct down
        var data = this.pshift(this.rate, sin);
        for (var i = 0; i < sin.length; i++) {
            sout[i] = data[i];
        }
    }
}
```

Eventhandler



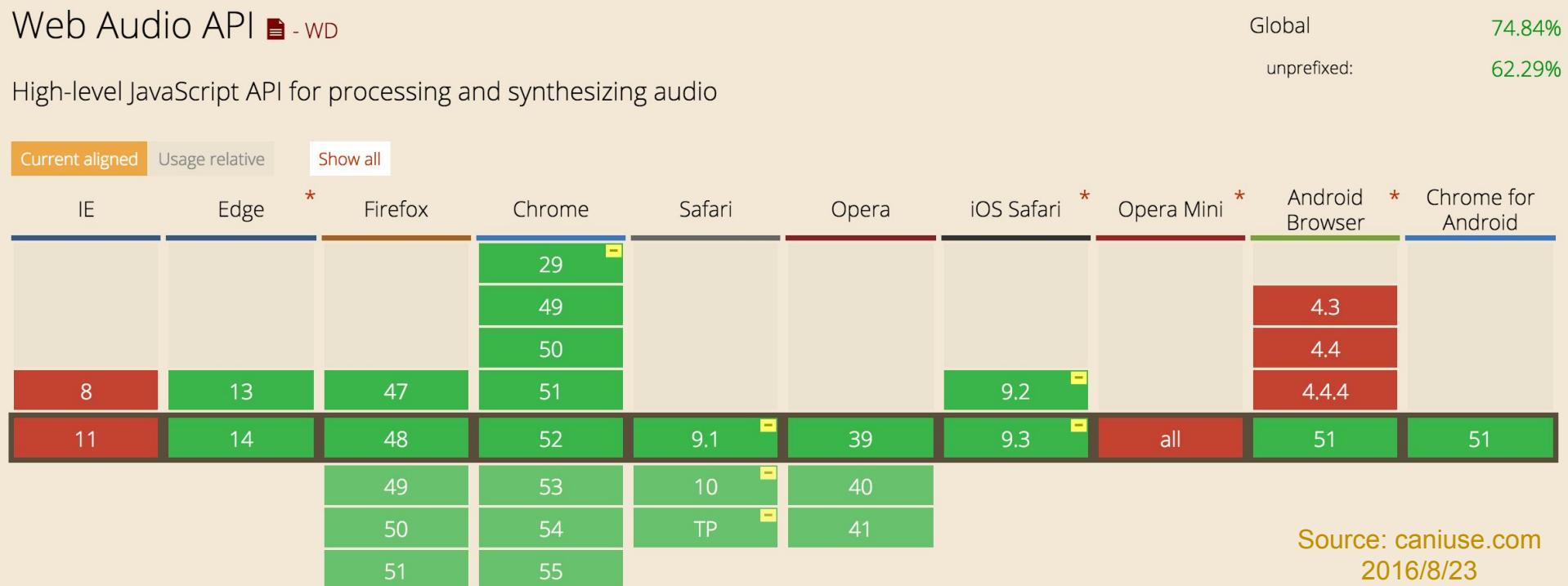
~~AudioWorker~~



AudioWorklet

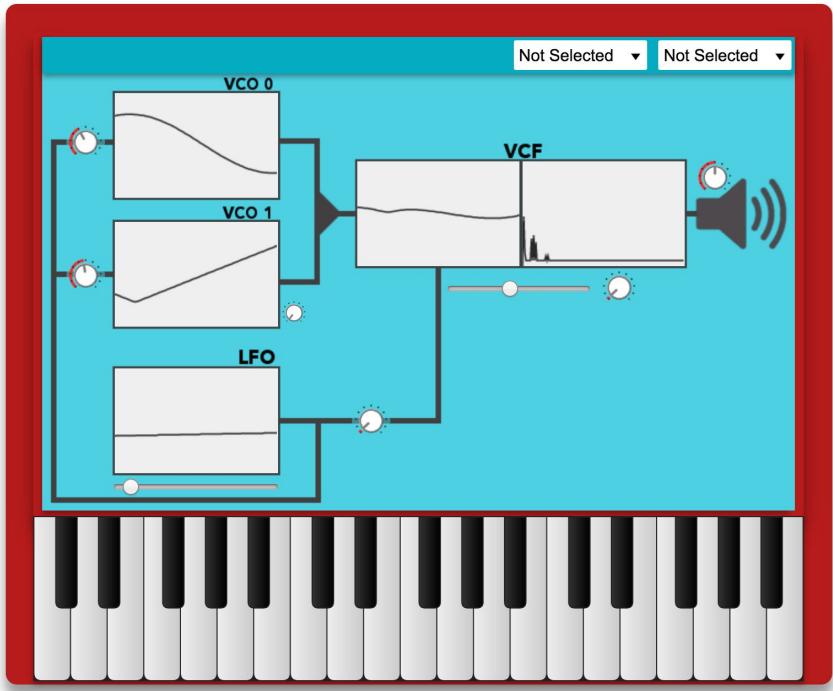
↑イマココ

# Web Audio APIを利用可能なブラウザ



# 应用例

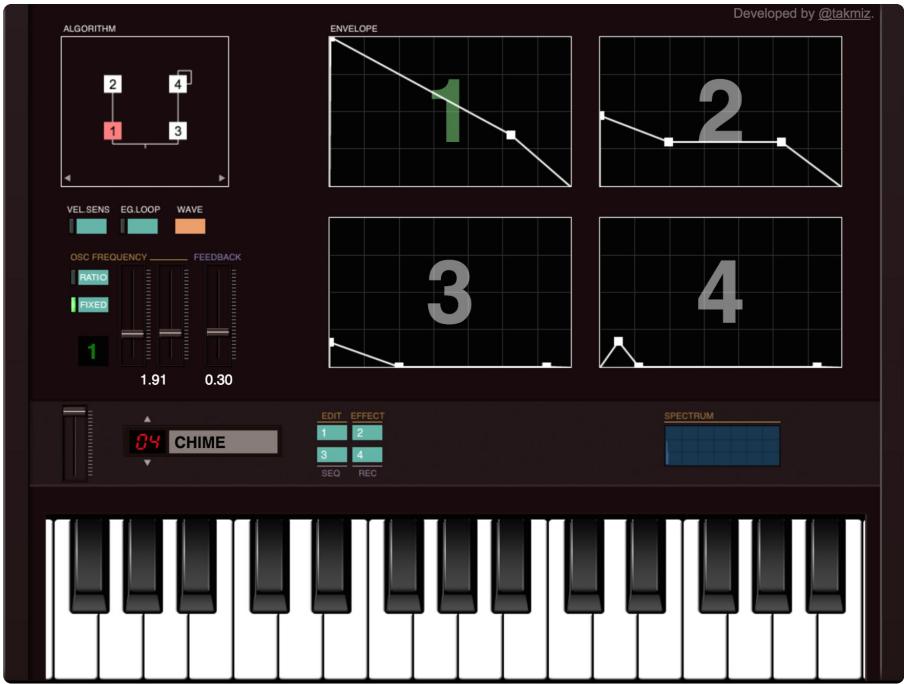




## アナログシンセサイザー



<https://goo.gl/ODtD8n>



## FMシンセサイザー

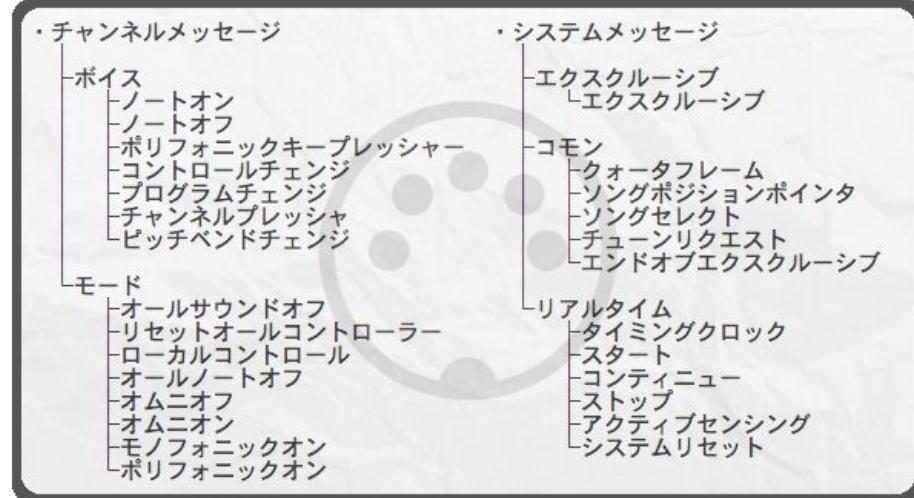
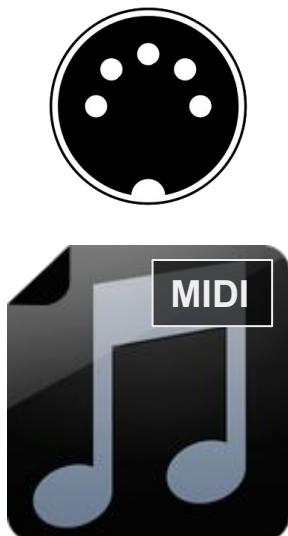
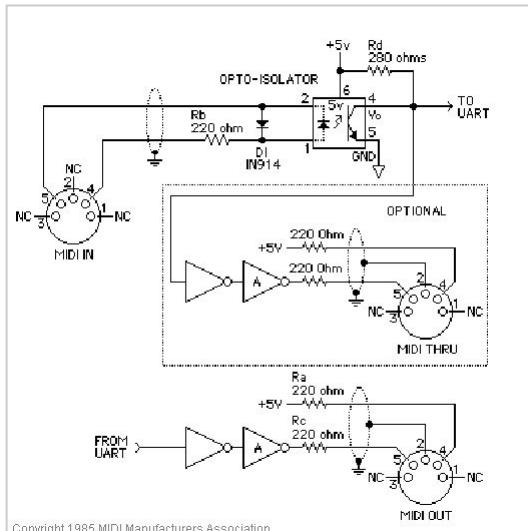
Developed by @takmiz

# Web MIDI API



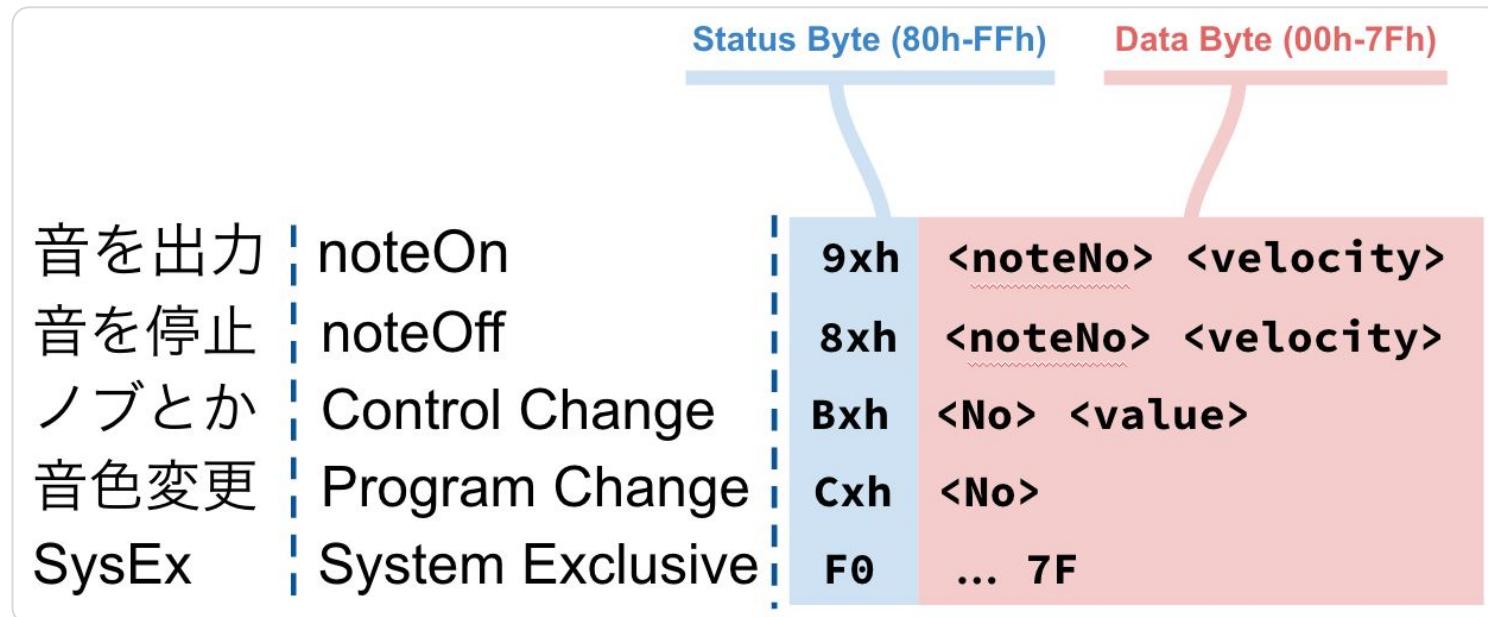
# MIDIとは

- **Musical Instrument Digital Interface** (公開: 1982年10月)
  - 演奏データを機器間でデジタル転送する規格
    - 物理的な送受信回路
    - インターフェイス
    - プロトコル
    - ファイルフォーマット、等



# MIDIメッセージ

- 8ビットのメッセージの集合
- 16進数
- 16チャンネル
- 0～127で数値を表現



Google

midi メッセージ



[Demo](#) : /webaudiomidi/midimsgv/

# コードの説明

```
<html>
  <head>
    <title>Web MIDI Demo | HTML5 Conference 2016</title>
  </head>
  <body>
    <div><span>INPUT:</span> <select id="midiinput"><option value="0">Select One</option></select></div>
    <div><span>OUTPUT:</span> <select id="midioutput"><option value="0">Select One</option></select></div>
    <script type="text/javascript">
      var inputs=[], outputs=[];
      var inSelected=false, outSelected=false;
      navigator.requestMIDIAccess({sysex: true}).then(successCallback, errorCallback);

      function successCallback(midi) {
        // input port
        var inputItr=midi.inputs.values();
        for(var i=inputItr.next(); !i.done; i=inputItr.next()) {
          inputs.push(i.value);
        }

        // output port
        var outputItr=midi.outputs.values();
        for(var i=outputItr.next(); !i.done; i=outputItr.next()) {
          outputs.push(i.value);
        }

        midi.onstatechange=function(event) {
          // do something when connection state is changed
        };

        function onmidimessage(event) {
          if(outSelected!==false) outputs[outSelected].send(event.data);
        }
      }

      function errorCallback(msg) {
        console.error("Error occurred during enumerating MIDI devices", msg);
      }
    </script>
  </body>
</html>
```

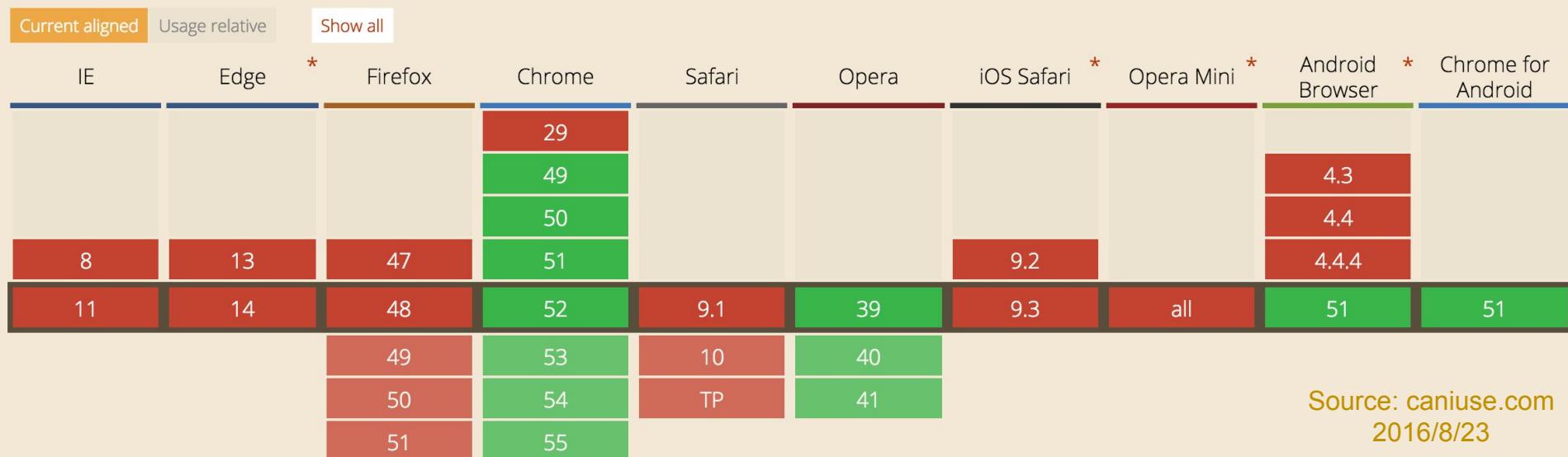
# Web MIDI APIを利用可能なブラウザ

Web MIDI API [- WD](#)

Global

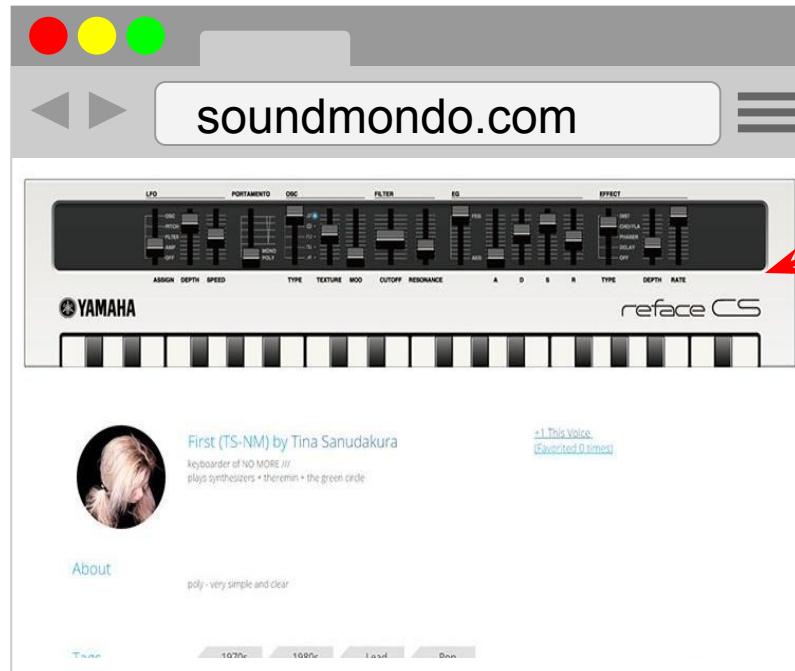
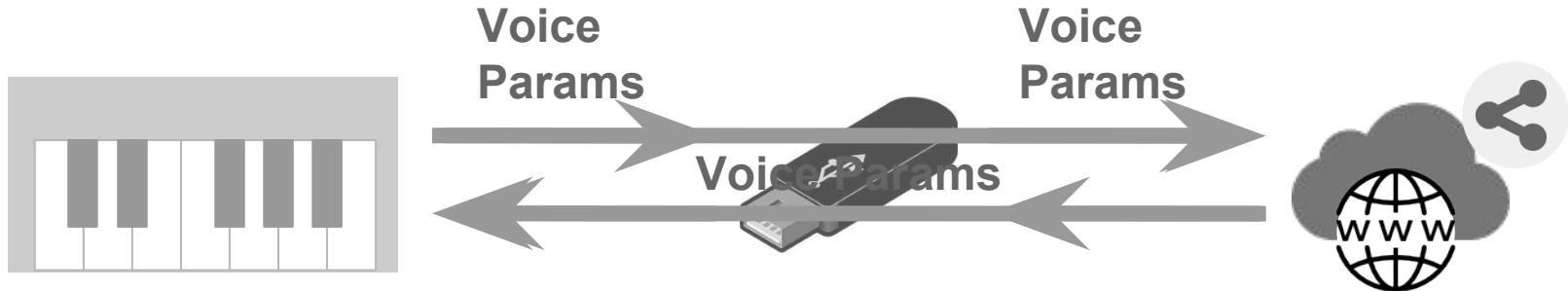
52.8%

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



# Soundmondo

Web MIDI APIを使ったシンセサイザー用音色共有サイト



# madeon (EDM Artist in France)

BUY  
ALBUM

madeon  
L R V E K + V X E

INFO

TOUR  
DATES

STREAM  
MUSIC



SHARE

WE MADE THIS

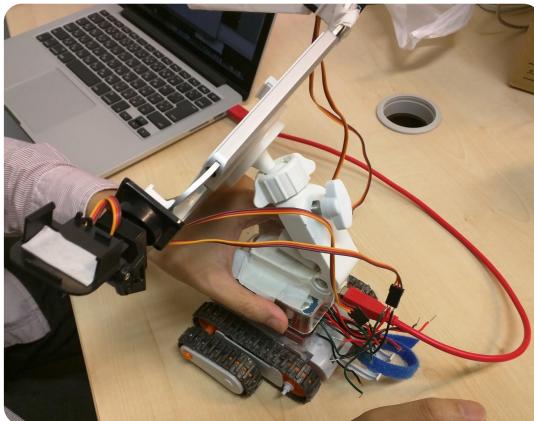
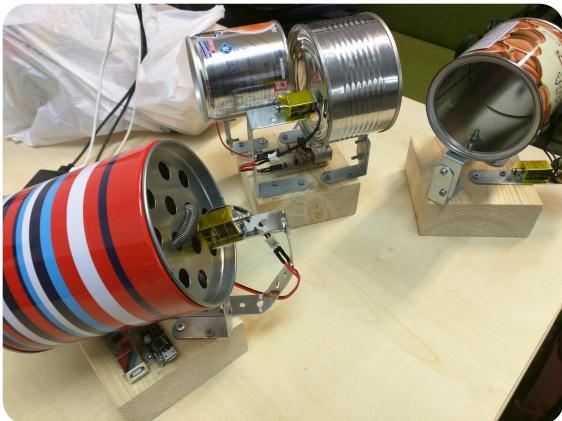
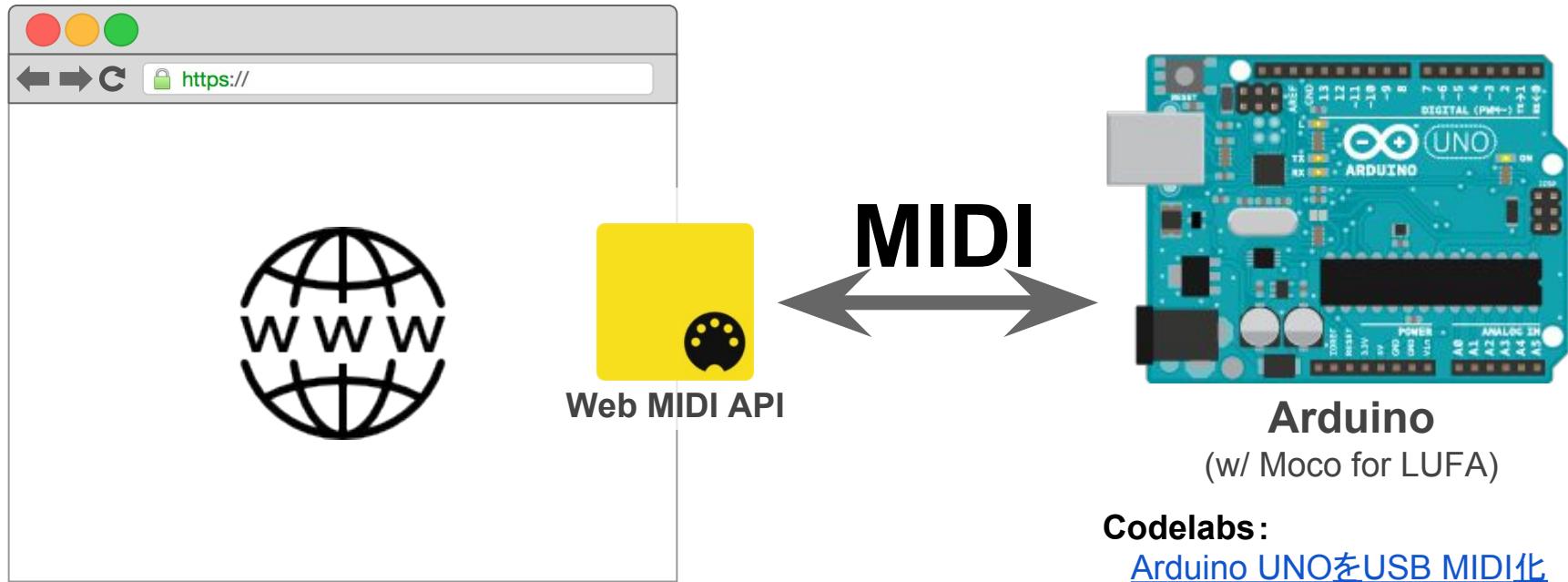
<https://www.madeon.fr/adventuremachine/>

# Web Music ハッカソン #5 (2016/7/30)

コミュニティ主催のハッカソンからの作品の紹介



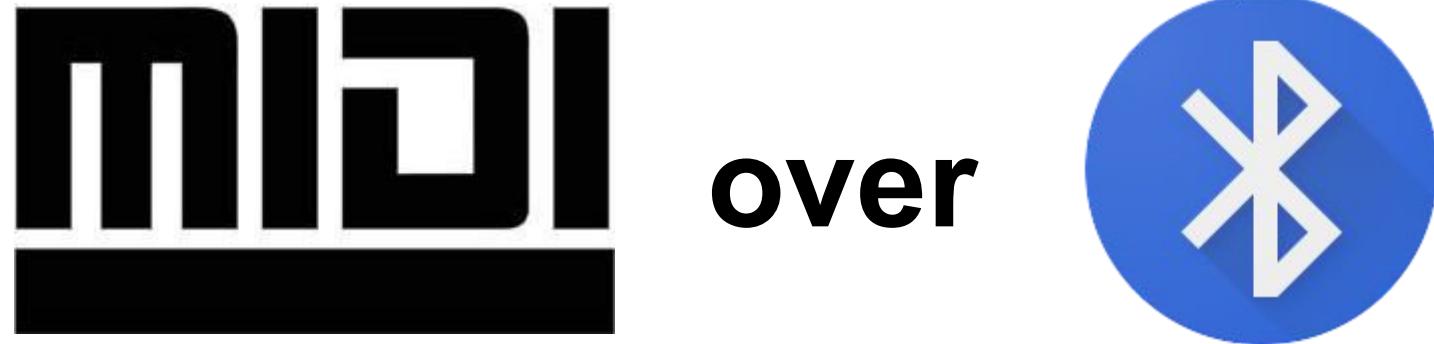
# Web MIDI API と DIY



DIYで使うなら改善したい点

- 手順がHacky
- 有線…
- デバッグが…
- Resolutionが…
- In/Outが別々

# MIDIがBluetoothに対応(Bluetooth MIDI)

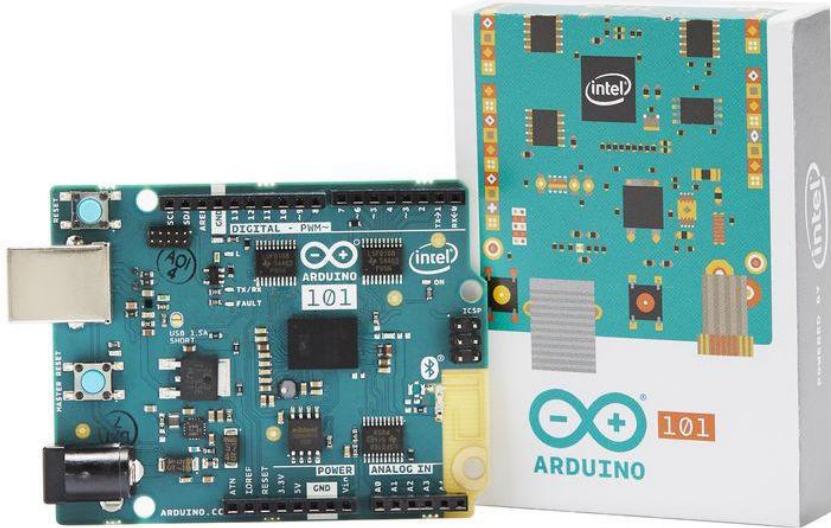


## DIYで使うなら改善したい点

- 手順がHacky
- 有線…
- デバッグが…
- Resolutionが…
- In/Outが別々

[Demo : /webaudiomidi/fmsynth/](#)

# ArduinoがBluetooth化(Arduino 101)

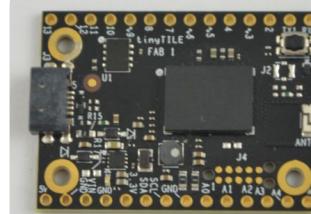


Intel社製のCurieを搭載

- 6軸加速度センサ
- ジャイロセンサ
- Bluetooth LE (Bluetooth Smart)

DIYで使うなら改善したい点

- 手順がHacky
- 有線…
- デバッグが…
- Resolutionが…
- In/Outが別々



**TinyTILE**  
Arduino 101 互換ボード  
(Comming soon?)

Demo : /webaudiomidi/blemidicon/



# Web Bluetooth

# Wi-FiとBluetoothの特徴



- ネットワーク的なつながり
- 設定が必要
- ルータ経由で繋がる
- 近距離のつながり
- 範囲内ならだいたい繋がる
- 直接繋がる

# 直接繋がる

## デモ: PLAY BULB candle

ホーム&キッチン > 家電 > 照明 > 屋内照明 > キャンドルライト



PLAYBULB candle 【ルームキャンドル アロマキャンドル Bluetoothコントロール LED】 BTL300 (1)  
PLAY BULB  
★★★★★☆ 6件のカスタマーレビュー

価格: ¥ 2,180 通常配送無料 詳細

残り17点 ご注文はお早めに 在庫状況について

住所からお届け予定日を確認 153-0064 - 東京都目黒区下目黒 詳細

8/30 火曜日 にお届けするには、今から1時間3分以内に「お急ぎ便」または「当日お急ぎ便」を選択して注文を確定してください (有料オプション)。Amazon プライム会員は無料)

この商品は、超激安堂が販売し、Amazon.co.jp が発送します。この出品商品にはコンビニ・ATM・ネットバンキング・電子マネー払いが利用できます。

新品の出品: 13 ¥ 1,200より

パッケージ商品数: 1

1 ¥ 2,180	3 ¥ 4,590
--------------	--------------

- お部屋やお店のムード作りにおすすめ
- アロマ使用可能。癒し効果も発揮します。
- Bluetoothでコントロールできるキャンドル。一台のデバイスでキャンドル5台まで (推奨) ~最大10台まで可能
- レインボーカラーに点灯したり、お好みのパターンでライティング効果を調整出来ます。



PLAYBULB CANDLE

- No Effect
- Candle Effect
- Flashing
- Pulse
- Rainbow
- Rainbow Fade

Battery: 100%

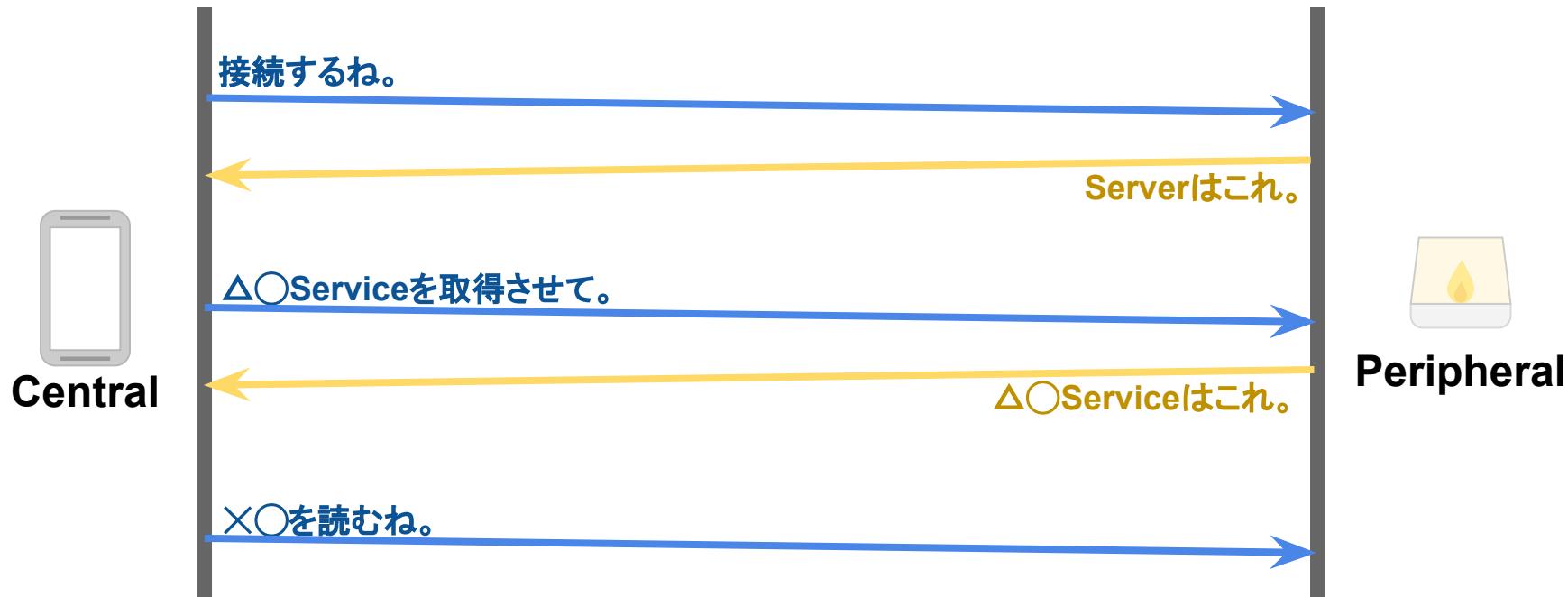
Code is on GitHub

# Bluetoothの種類、接続手順

- 種類

- **Central**: 探索する側(スマホ、PC等)
- **Peripheral**: 探索される側(心拍センサ、ビーコン等)

- 接続シーケンス



# コードの説明

/webbluetooth/candymagic/index.html

```
<button id="find" >Connect</button>
<script type="text/javascript">16
function handleDevice (characteristic) {
    // do something
}
function service (service) {
    console.log(service);
    Promise.all([
        service.getCharacteristic('19b10001-e8f2-537e-4f6c-984fee0f808e')
    ]).then(handleDevice, error);
```

UUIDは小文字で

```
}

function gatt (server) {
    server.device.ongattserverdisconnected=function(msg){
        console.error("[server disconnected] ", msg);
    };
    server.getPrimaryService('19b10000-e8f2-537e-4f6c-984fee0f808e').then(service, error);
}
```

```
}

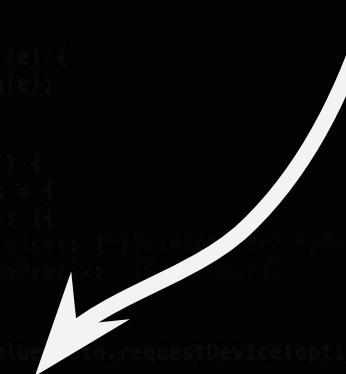
function found (device) {
    device.gatt.connect().then(gatt, error);
}
```

```
21
function error (e) {
    console.log(e);
}22
Web Bluetooth
```

```
function find () {
    var options = {
        filters: [
            services: ["19b10000-e8f2-537e-4f6c-984fee0f808e"],
            namePrefix: '[SearchWord]'
        ]
    };
    navigator.bluetooth.requestDevice(options).then(found, error);
}
```

```
23
document.getElementById("find").addEventListener("click", find, false);
</script>
```

開始にはユーザーのインタラクションが必須



# Web Bluetooth と Arduino 101 でデモ



地球グルグル

/webbluetooth/eearthgrugru/

(Peripheral → Central)



Candy Magic

/webbluetooth/candymagic/

(Peripheral ← Central)

# HTML

唐突ですが、HTML5 Conference なので！

# Progressive Web APP

NativeアプリのようなUXを提供するWebアプリ

## 含まれる技術要素

- Service Worker
- Add to Home Screen
- App Manifest
- Background Sync 等



## 積極的に使いましょう！

- NativeアプリのようなUXを提供したい
- 必ずしもネットワーク必要としないアプリが多い
- 使いたい場所にネット、携帯が利用可能とは限らない

# Progressive Web APP 実装してます！



以下の動作が確認可能です！

- Offlineでの動作
- Add to Home Screen

15:20 - 16:00

Progressive Web Apps の次の一步～アイデンティティとペイメント

keywords Credential Management API, Web Payments, Progressive Web Apps

プログレッシブウェブアプリの次の一步として、ログインと支払いを最適化する新しいAPIについてお話しします



えーじ  
Google

詳細はえーじさんのセッションで！  
@ ルームA(1F)

# まとめ

## Web Audio API

- ブラウザで信号処理

## Web MIDI API

- ブラウザと外部MIDIデバイスが直接つながる
- Bluetooth MIDIを使えば無線にできる

## Web Bluetooth

- Bluetooth LE機器が接続可能
- Arduino 101 を使えばオリジナル機器の作成が可能

## Progressive Web APP

- 是非使いましょう！！

## 本日のコード



<https://goo.gl/kBoRzZ>



# Enjoy Hacking !

ご清聴ありがとうございました！