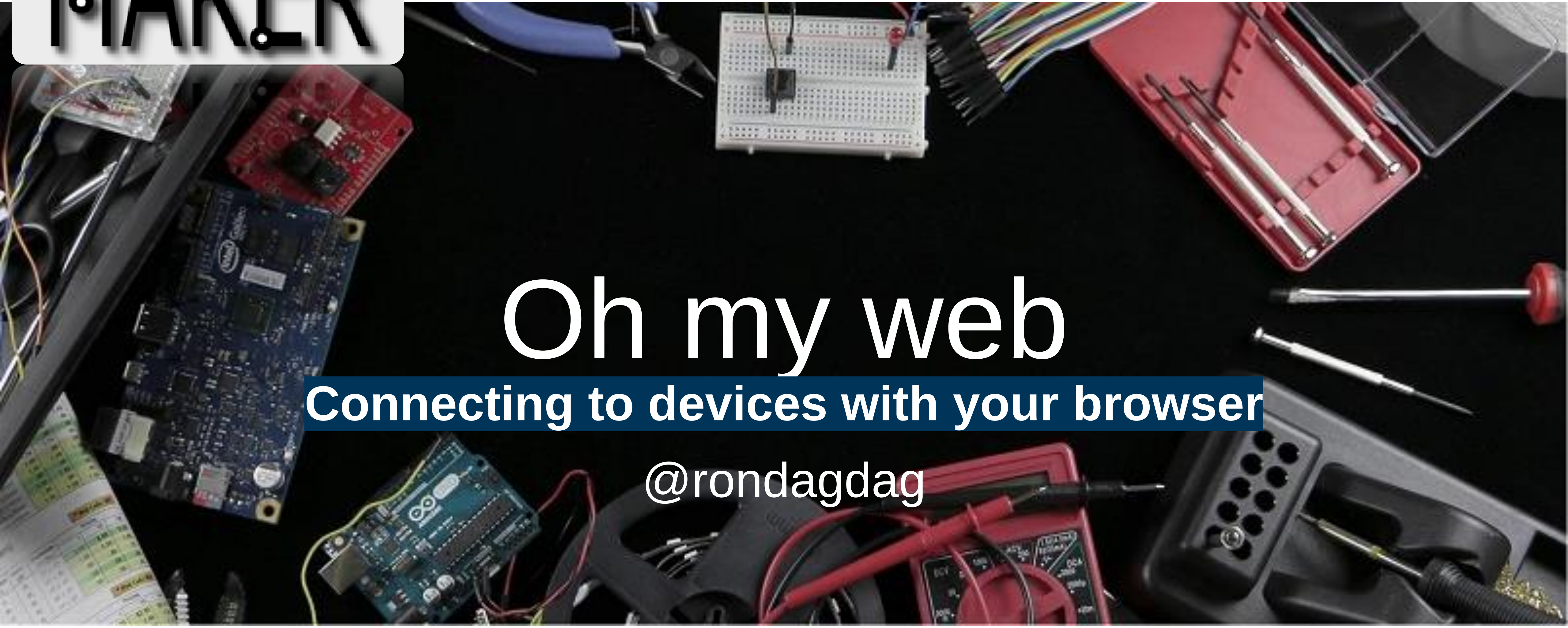
MAKER



Oh my web

Connecting to devices with your browser

@rondagdag

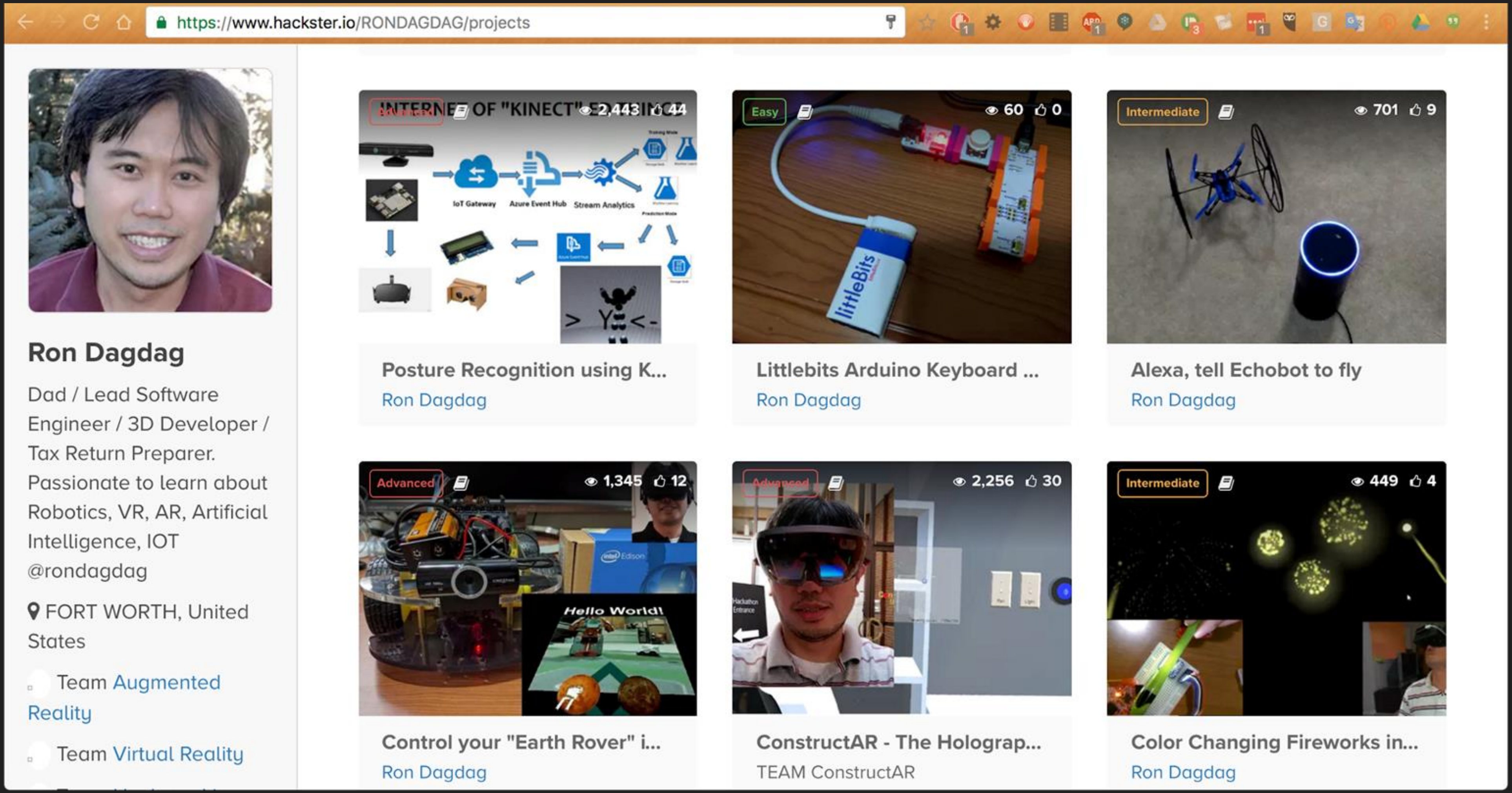
Ron Dagdag

Audience Survey

- Web Developers?
- UX Designers?
- VR Developers?
- IoT Engineers?

Hackster Portfolio

Ron Dagdag @rondagdag



https://www.hackster.io/RONDAGDAG/projects

Ron Dagdag

Dad / Lead Software Engineer / 3D Developer / Tax Return Preparer. Passionate to learn about Robotics, VR, AR, Artificial Intelligence, IOT @rondagdag

FORT WORTH, United States

Team Augmented Reality

Team Virtual Reality

Project	Author	Difficulty	Views	Likes
Posture Recognition using K...	Ron Dagdag	Advanced	2,443	44
Littlebits Arduino Keyboard ...	Ron Dagdag	Easy	60	0
Alexa, tell Echobot to fly	Ron Dagdag	Intermediate	701	9
Control your "Earth Rover" i...	Ron Dagdag	Advanced	1,345	12
ConstructAR - The Holograp...	TEAM ConstructAR	Advanced	2,256	30
Color Changing Fireworks in...	Ron Dagdag	Intermediate	449	4

My Story

ASP Active Server Pages

ASP.NET

ASP.NET Core

HTML JS CSS

node JS

The Web Eats Everything in its Path

-Graphics

-Animation

-Location

-Motion Input

-Real-Time 3D

-Mixed Reality

-Camera

-Messaging

-Real-Time Messaging

-IoT/Wearables

-Robotics

IoT use cases

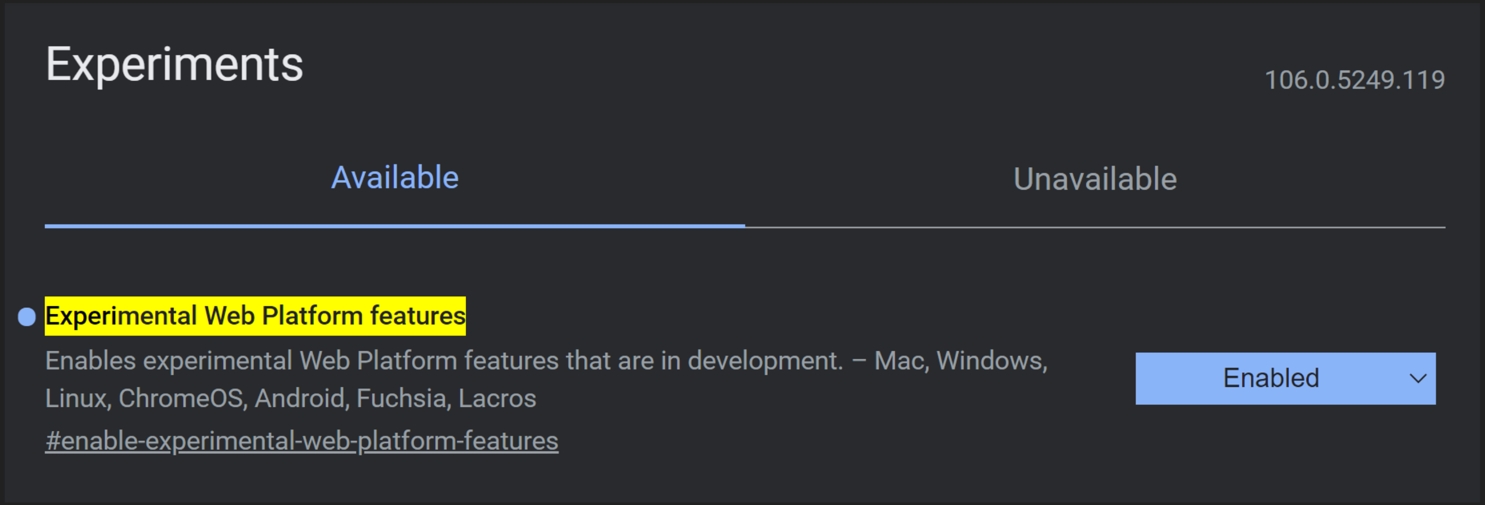
Monitoring

Control

- Not everything needs to be connected to the Internet all the time.
- Only connect when it’s needed
- by allowing sandboxed code to request limited access to a device

Web MIDI, Web Bluetooth, Web USB, Web NFC

chrome://flags/#enable-experimental-web-platform-features



Experiments

106.0.5249.119

Available | Unavailable

● Experimental Web Platform features

Enables experimental Web Platform features that are in development. – Mac, Windows, Linux, ChromeOS, Android, Fuchsia, Lacros

#enable-experimental-web-platform-features

Enabled

What is MIDI?

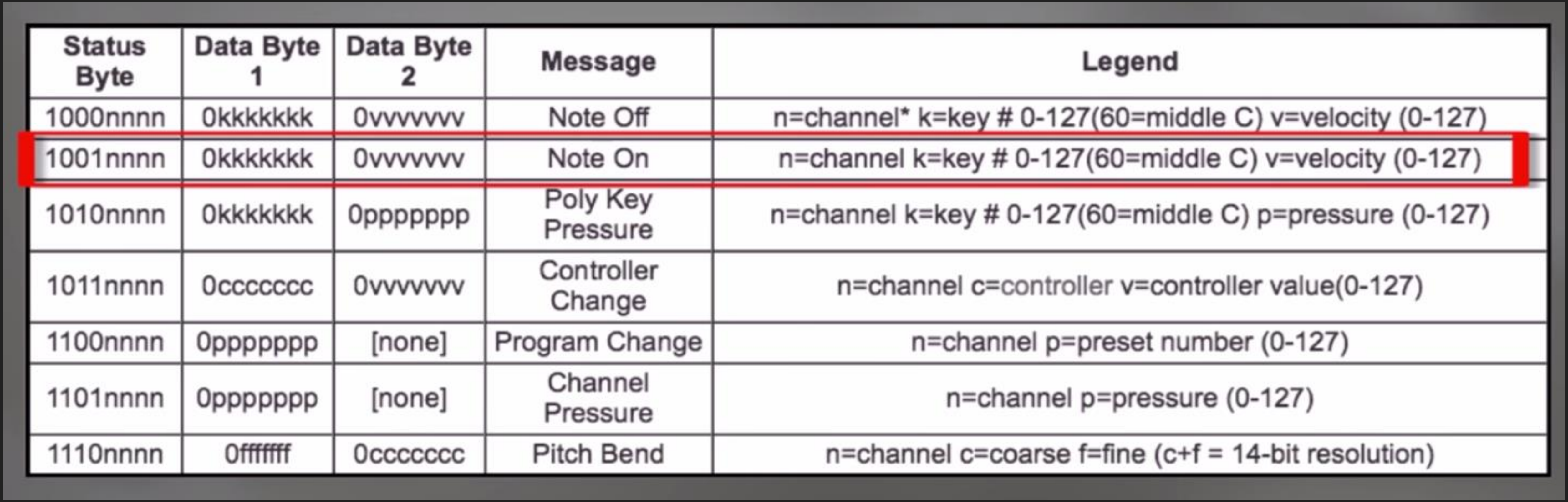
MIDI

- Musical Instrument Digital Interface
- 1981 by founder of Roland, Ikutaro Kakehashi
- Industry standard music technology protocol
- connects products like
  digital musical instruments,
  computers, tablets, and smartphones

How to use the WebMIDI API?

1. Request access
2. Scan for Relevant Device
3. Add Event Listener
4. Decode the message

MIDI



Status Byte	Data Byte 1	Data Byte 2	Message	Legend
1000nnnn	0kkkkkkk	0vvvvvvv	Note Off	n=channel* k=key # 0-127(60=middle C) v=velocity (0-127)
1001nnnn	0kkkkkkk	0vvvvvvv	Note On	n=channel k=key # 0-127(60=middle C) v=velocity (0-127)
1010nnnn	0kkkkkkk	0ppppppp	Poly Key Pressure	n=channel k=key # 0-127(60=middle C) p=pressure (0-127)
1011nnnn	0ccccccc	0vvvvvvv	Controller Change	n=channel c=controller v=controller value(0-127)
1100nnnn	0ppppppp	[none]	Program Change	n=channel p=preset number (0-127)
1101nnnn	0ppppppp	[none]	Channel Pressure	n=channel p=pressure (0-127)
1110nnnn	0fffffff	0ccccccc	Pitch Bend	n=channel c=coarse f=fine (c+f = 14-bit resolution)

How to use the WebMIDI API?

1. Request access

```
let midiAccess = await navigator.requestMIDIAccess();
```

1. Scan for Relevant Device

```
const inputs = midiAccess.inputs.values();
```

How to use the WebMIDI API?

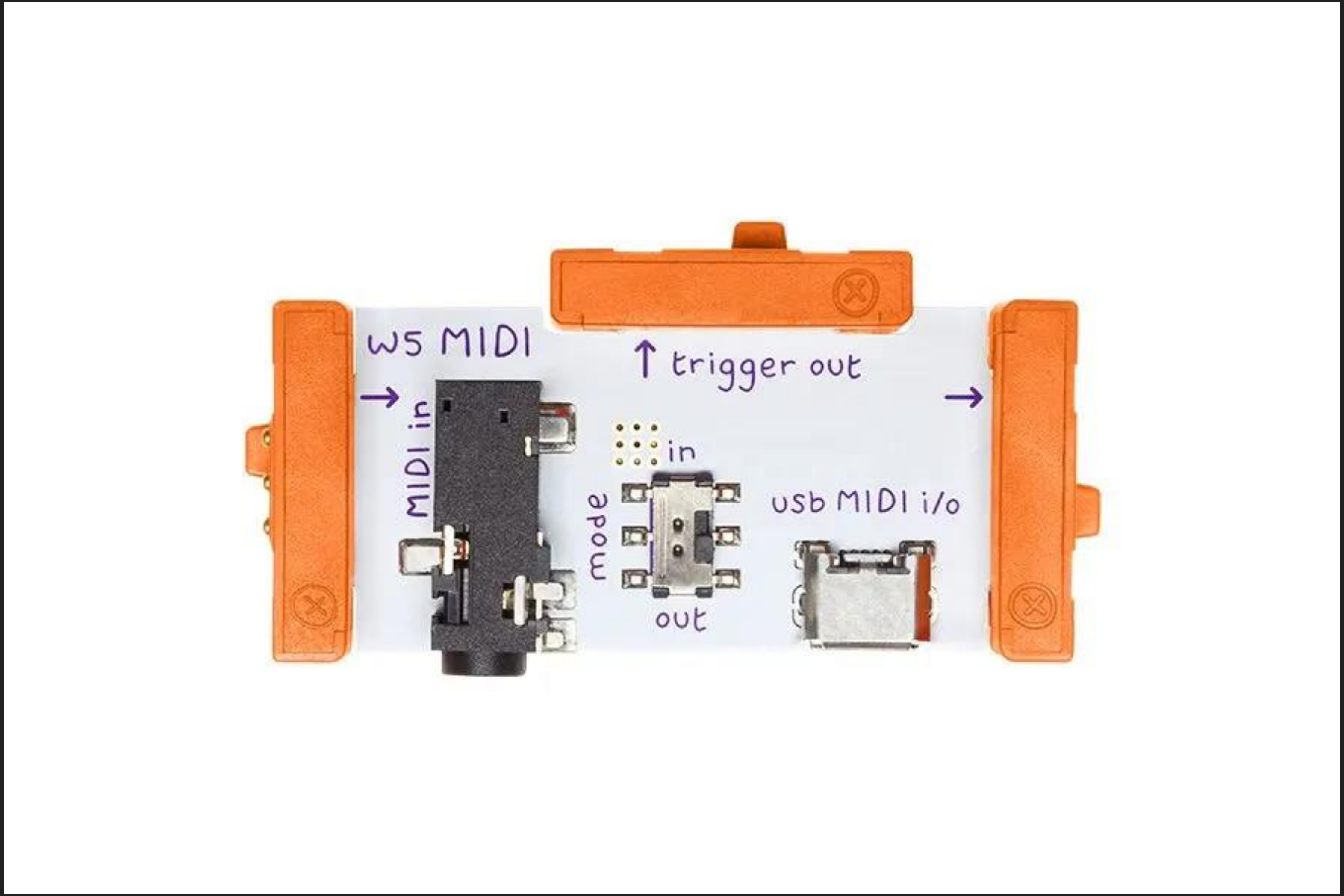
3. Add Event Listener

```
    input.addEventListener("midimessage",
                MIDIMessageEventHandler);
```

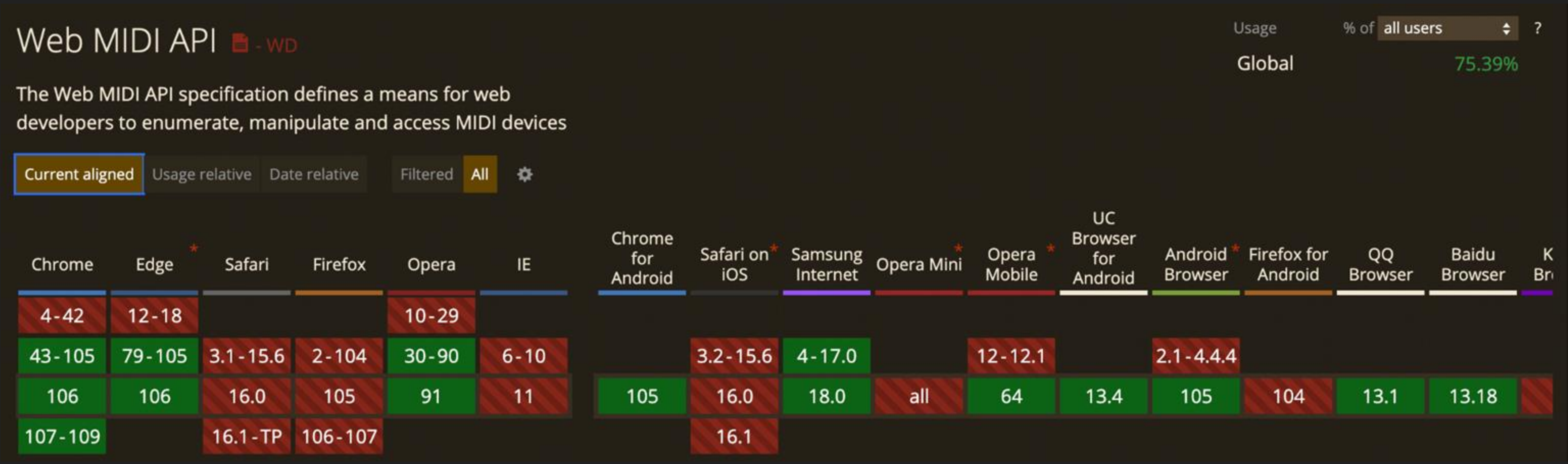
4. Decode the message

```
    const cmd = event.data[0] >> 4; //on or off
 const pitch = event.data[1];
 const velocity = event.data.length > 2 ? event.data[2] : 1;
      // if velocity == 0, fall thru: it's a note-off.
```

Littlebits MIDI?



Web MIDI



Web MIDI API - WD

Usage % of all users ?

Global 75.39%

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

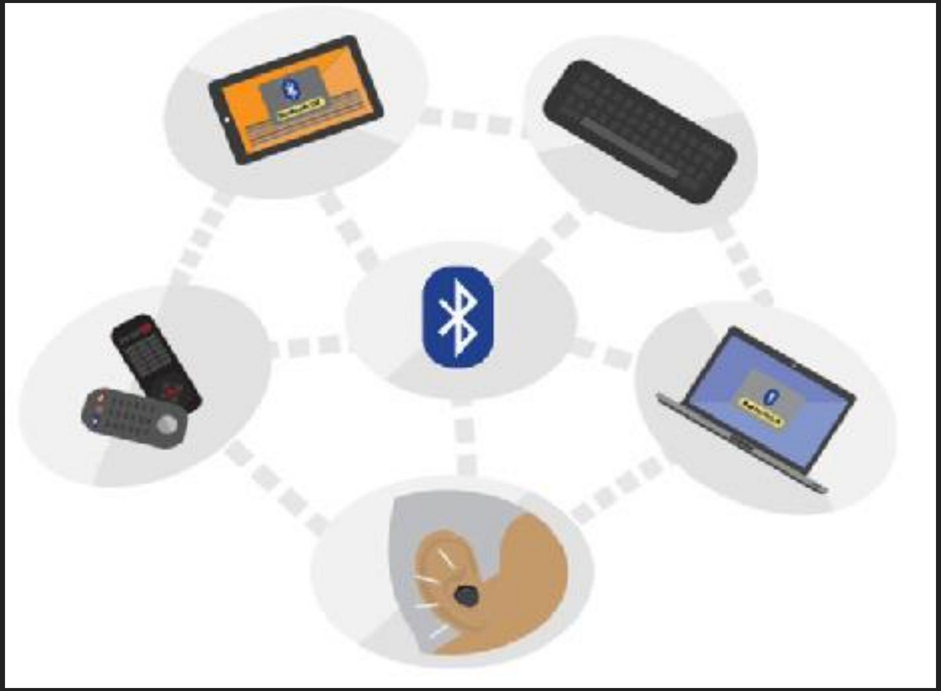
Current aligned | Usage relative | Date relative | Filtered | All

Chrome	Edge *	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	Opera Mobile *	UC Browser for Android	Android Browser *	Firefox for Android	QQ Browser	Baidu Browser	K Br
🟥 4-42	🟥 12-18			🟥 10-29												
🟩 43-105	🟩 79-105	🟥 3.1-15.6	🟥 2-104	🟩 30-90	🟥 6-10		🟥 3.2-15.6	🟩 4-17.0		🟥 12-12.1		🟥 2.1-4.4.4				
🟩 106	🟩 106	🟥 16.0	🟥 105	🟩 91	🟥 11	🟩 105	🟥 16.0	🟩 18.0	🟥 all	🟩 64	🟩 13.4	🟩 105	🟥 104	🟩 13.1	🟩 13.18	🟥
🟩 107-109		🟥 16.1-TP	🟥 106-107				🟥 16.1									

https://caniuse.com/#feat=midi

What is Bluetooth?

- Standard (specification)
- wireless communication standard
- allows electronic devices to connect and interact with each other
- short distances less than about 10m or 30ft
- Bluetooth 5 - maximum of around 800 feet



In the beginning

* Ericsson 1994
* Replacement for RS-232
* Original name:
  * Short link radio technology
* 1999 got the name Bluetooth
* Bluetooth Special Interest Group
  * More than 20k members

Bluetooth®
SPECIAL INTEREST GROUP

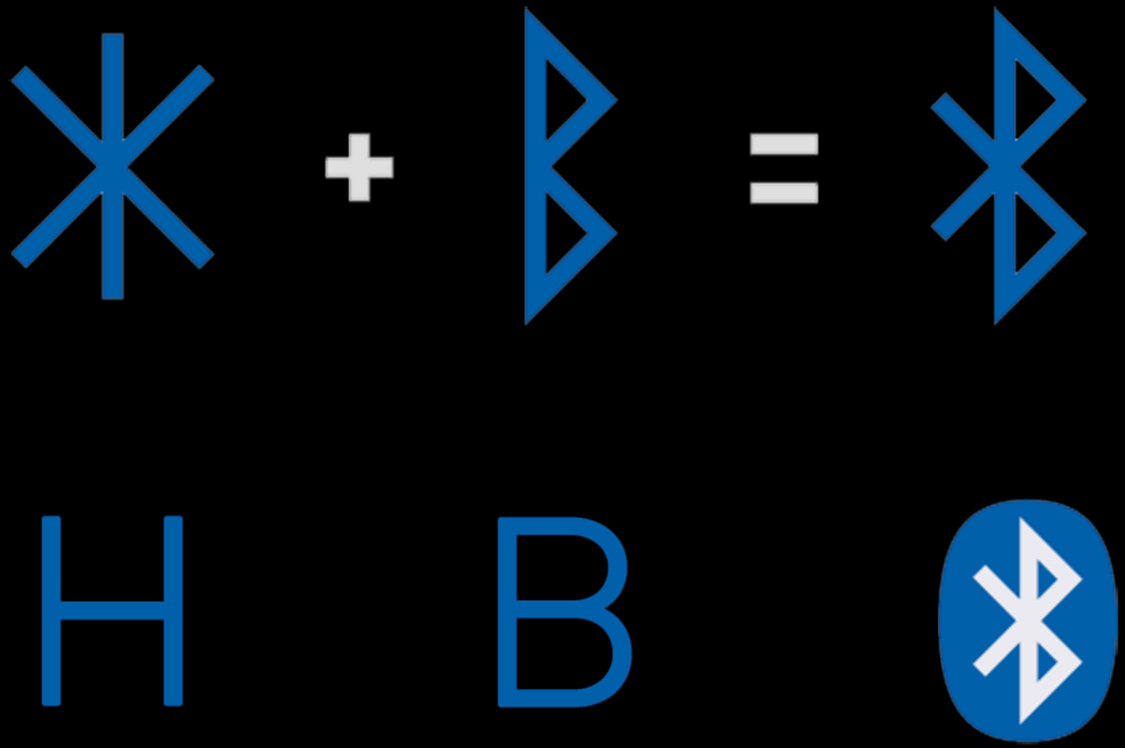
ERICSSON

10 million

Bluetooth enabled devices shipping **EVERY DAY**

Source: ABI Research, via Martin Woolley

Fun Fact...



Norse runes for Harald Bluetooth, 10th century King of Denmark

Classic v.s. BLE (smart)

3.0 Classic

* Connection session (connected all the time
* Connection time higher
* Voice capable

* Pairing

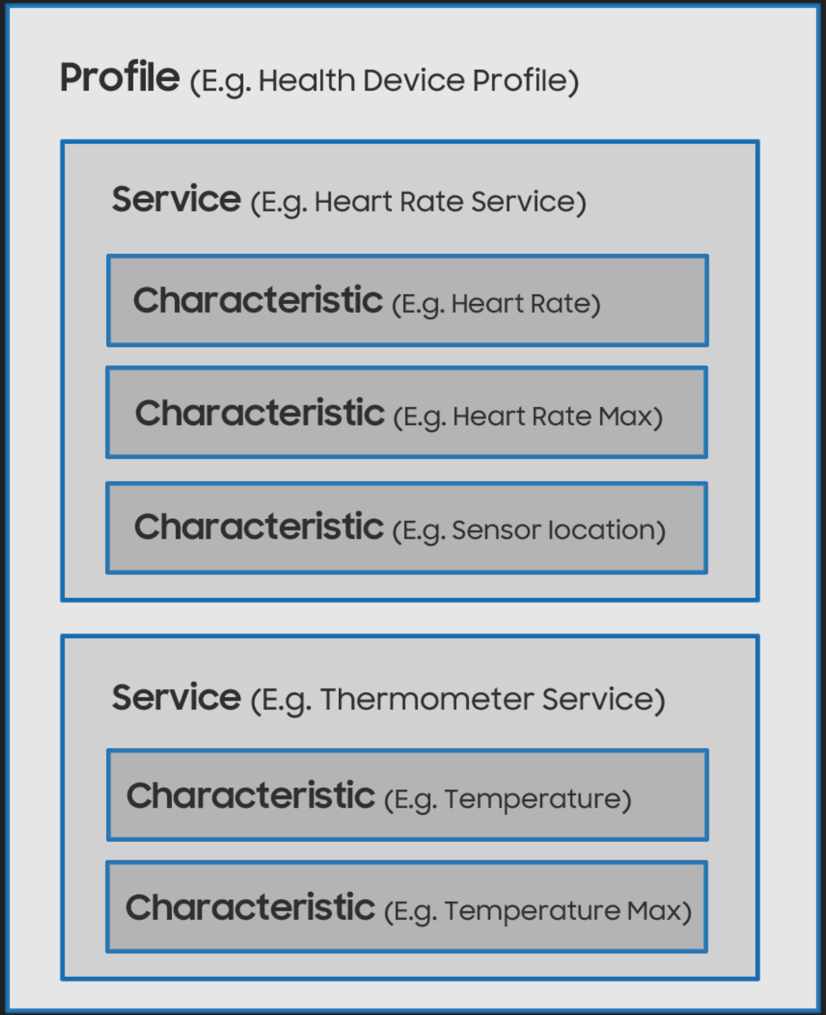
4.0 Low Energy

* On/Off

* Fast connection (3ms)
* No voice (some unidirectional for hearing aids)
* Beacons
  * 32 bytes

Generic Attribute Profile (GATT)

- Generic Attribute Profile
- Peripheral (Server)
- Central (Client)
- Read
- Write
- Notify
- Indicate (Ack)



Profile (E.g. Health Device Profile)
Service (E.g. Heart Rate Service)
Characteristic (E.g. Heart Rate)
Characteristic (E.g. Heart Rate Max)
Characteristic (E.g. Sensor location)
Service (E.g. Thermometer Service)
Characteristic (E.g. Temperature)
Characteristic (E.g. Temperature Max)

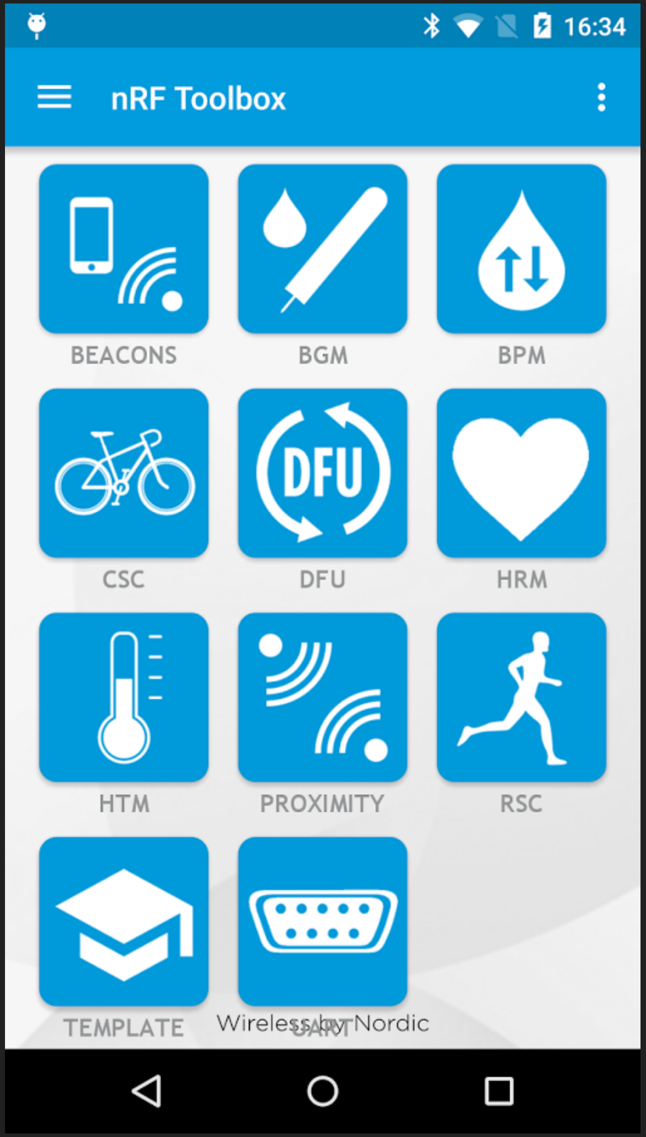
GATT Services

https://www.bluetooth.com/specifications/gatt/services

- Alert Notification Service
- Automation IO
- Battery Service
- Blood Pressure
- Body Composition
- Bond Management
- Continuous Glucose Monitoring
- Current Time Service
- Cycling Power
- Cycling Speed and Cadence
- Device Information
- Environmental Sensing
- Generic Access
- Generic Attribute
- Glucose
- Health Thermometer
- Heart Rate
- HTTP Proxy
- Human Interface Device
- Immediate Alert
- Indoor Positioning
- Internet Protocol Support

GATT Services

- Location and Navigation
- Next DST Change Service
- Object Transfer
- Phone Alert Status Service
- Pulse Oximeter
- Reference Time Update Service
- Running Speed and Cadence
- Transport Discovery
- Tx Power
- User Data
- Weight Scale



https://play.google.com/store/apps/details?id=no.nordicsemi.android.nrftoolbox

Web Bluetooth

- Control BLE devices directly from the web
- HTTPS only
- Security-First, User Interaction + Approval required
- ES6 Promise-based API

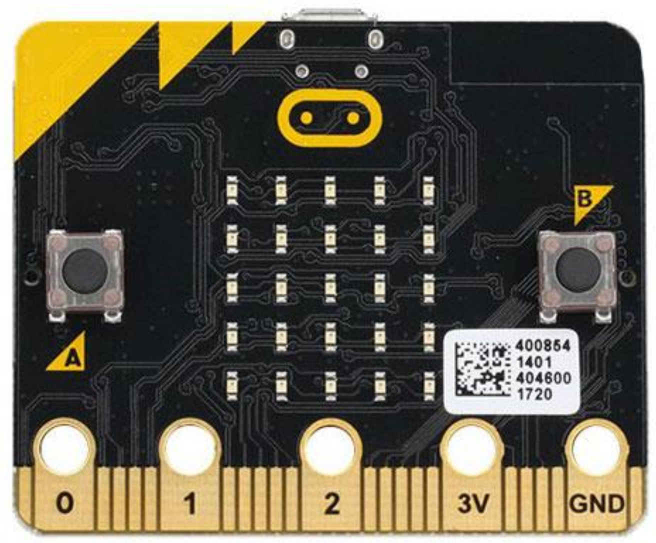
The Web Bluetooth API

* Available through navigator.bluetooth
* Can only be invoked through user interaction (e.g. button click)
* We need to specify filters – specific services/ device names we are interested in

How to use the WebBluetooth API?

1. Device has to be paired first before chromium can connect
2. Scan for a relevant Device
3. Connect to it
4. Get the Service you are interested in
5. Get the Characteristics you are interested in
6. Read / Write / Subscribe to the Characteristics

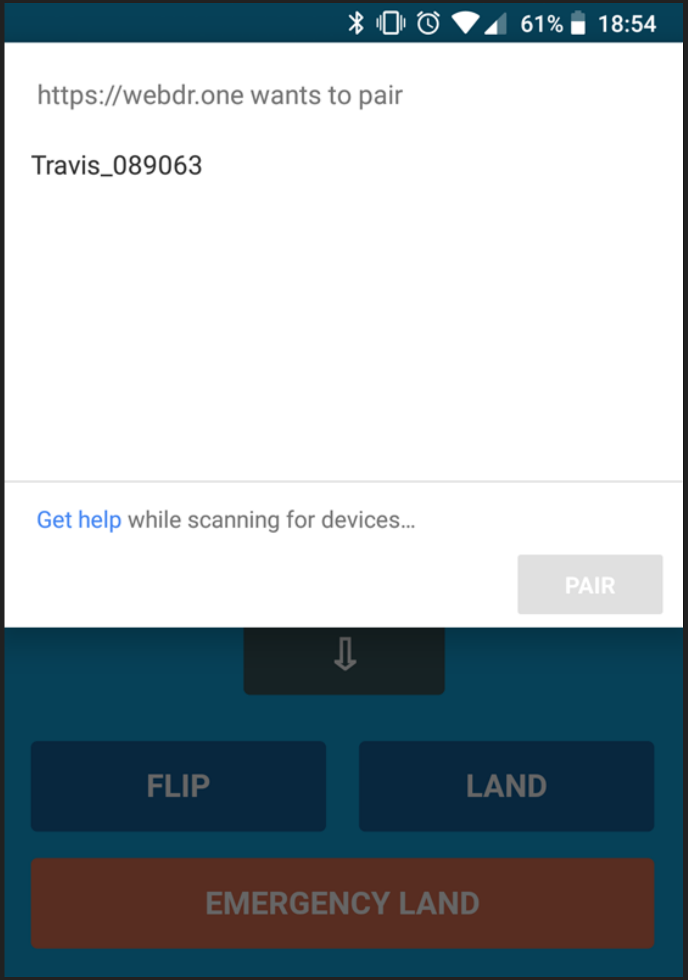
Micro:bit



Step 1 – Find a matching Device

```
targetDevice = await
navigator.bluetooth.requestDevice({
        // filters: [...] <- Prefer filters to
save energy & show relevant devices.
        filters: [{ services: [LED_SERVICE] },
{ namePrefix: "BBC micro:bit" }]
      });...
```

* Asks the user to choose a device from a list
* Returns a promise for the selected Device object


61% 18:54

https://webdr.one wants to pair

Travis_089063

Get help while scanning for devices...

Step 2 – Connect to the Device

```
.then(device => device.gatt.connect())
```

* Returns a promise for the GATT Server object, which you can query for Services

Step 3 – Get the Service

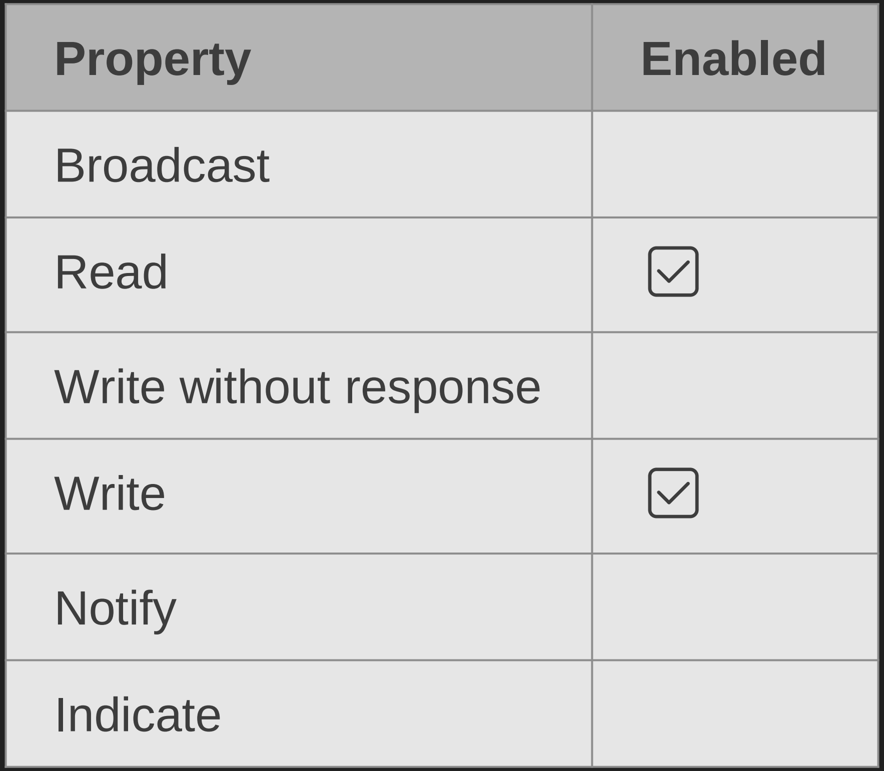
```
.then(server => {
  // Get Service...
  return server.getPrimaryService(serviceUUID);
})
```

* Returns a promise for the Service object

Step 4 – Get the Characteristic

```
.then(service => {   // Get Characteristic...
  return service.getCharacteristic(characteristicUUID);
})
```

* Returns a promise for the Characteristic object



Property	Enabled
Broadcast	
Read	✓
Write without response	
Write	✓
Notify	
Indicate	

Step 5 – Read

```
.then(characteristic => {
  return characteristic.readValue();
})
.then(value => {
  console.log('Value is ' + value.getUint8(0));
})
.catch(error => { console.log(error); });
```

* Returns a promise for DataView, which gives access to individual bytes

Step 5 – Write

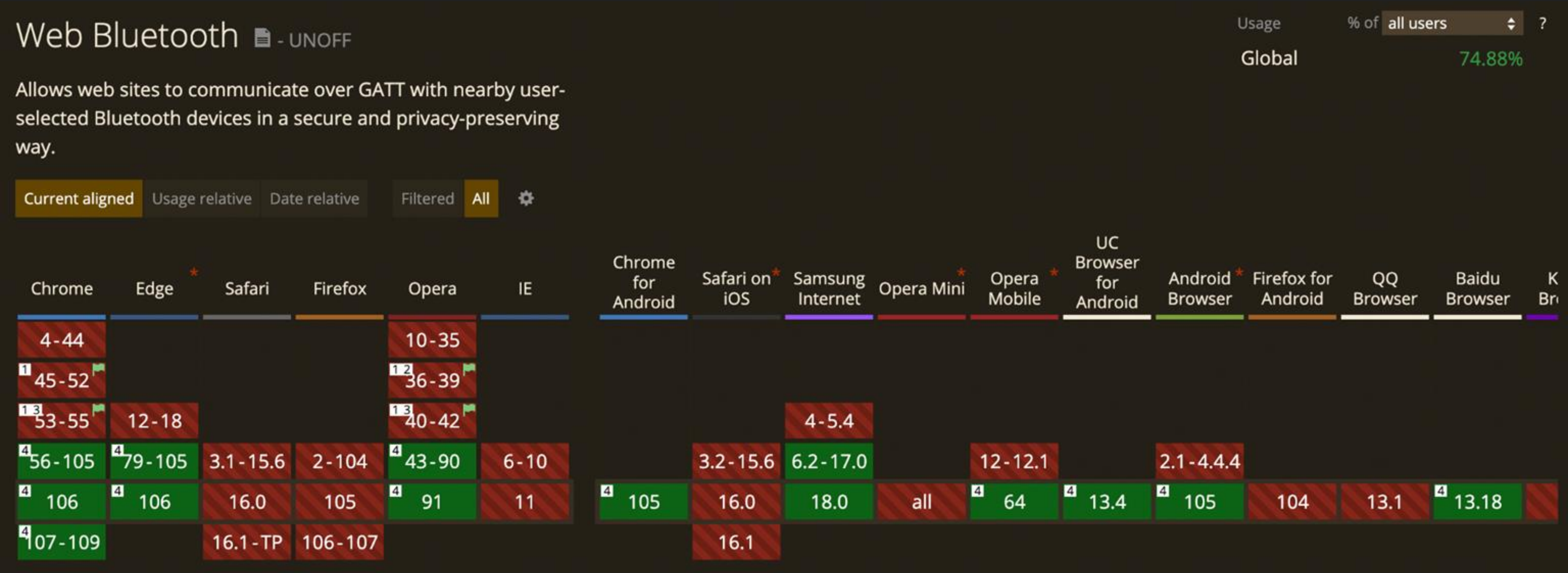
```
const data = new Uint8Array([0x55, 0x70])
characteristic.writeValue(data)
```

* Returns a promise which will be resolved after the value has been written

Chrome Debugging Tools

chrome://bluetooth-internals

Web Bluetooth



Web Bluetooth - UNOFF

Usage % of all users ?

Global 74.88%

Allows web sites to communicate over GATT with nearby user-selected Bluetooth devices in a secure and privacy-preserving way.

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

Chrome	Edge	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS	Samsung Internet	Opera Mini	Opera Mobile	UC Browser for Android	Android Browser	Firefox for Android	QQ Browser	Baidu Browser	K Br
🟥 4-44				🟥 10-35												
🟥 [1] 45-52				🟥 [1 2] 36-39												
🟥 [1 3] 53-55	🟥 12-18			🟥 [1 3] 40-42				🟥 4-5.4								
🟩 [4] 56-105	🟩 [4] 79-105	🟥 3.1-15.6	🟥 2-104	🟩 [4] 43-90	🟥 6-10		🟥 3.2-15.6	🟩 6.2-17.0		🟥 12-12.1		🟥 2.1-4.4.4				
🟩 [4] 106	🟩 [4] 106	🟥 16.0	🟥 105	🟩 [4] 91	🟥 11	🟩 [4] 105	🟥 16.0	🟩 18.0	🟥 all	🟩 [4] 64	🟩 [4] 13.4	🟩 [4] 105	🟥 104	🟥 13.1	🟩 [4] 13.18	🟥
🟩 [4] 107-109		🟥 16.1-TP	🟥 106-107				🟥 16.1									

What is USB?

➔ Universal Serial Bus

➔ standard type of connection for many different kinds of devices

➔ protocol for connecting peripherals to a computer

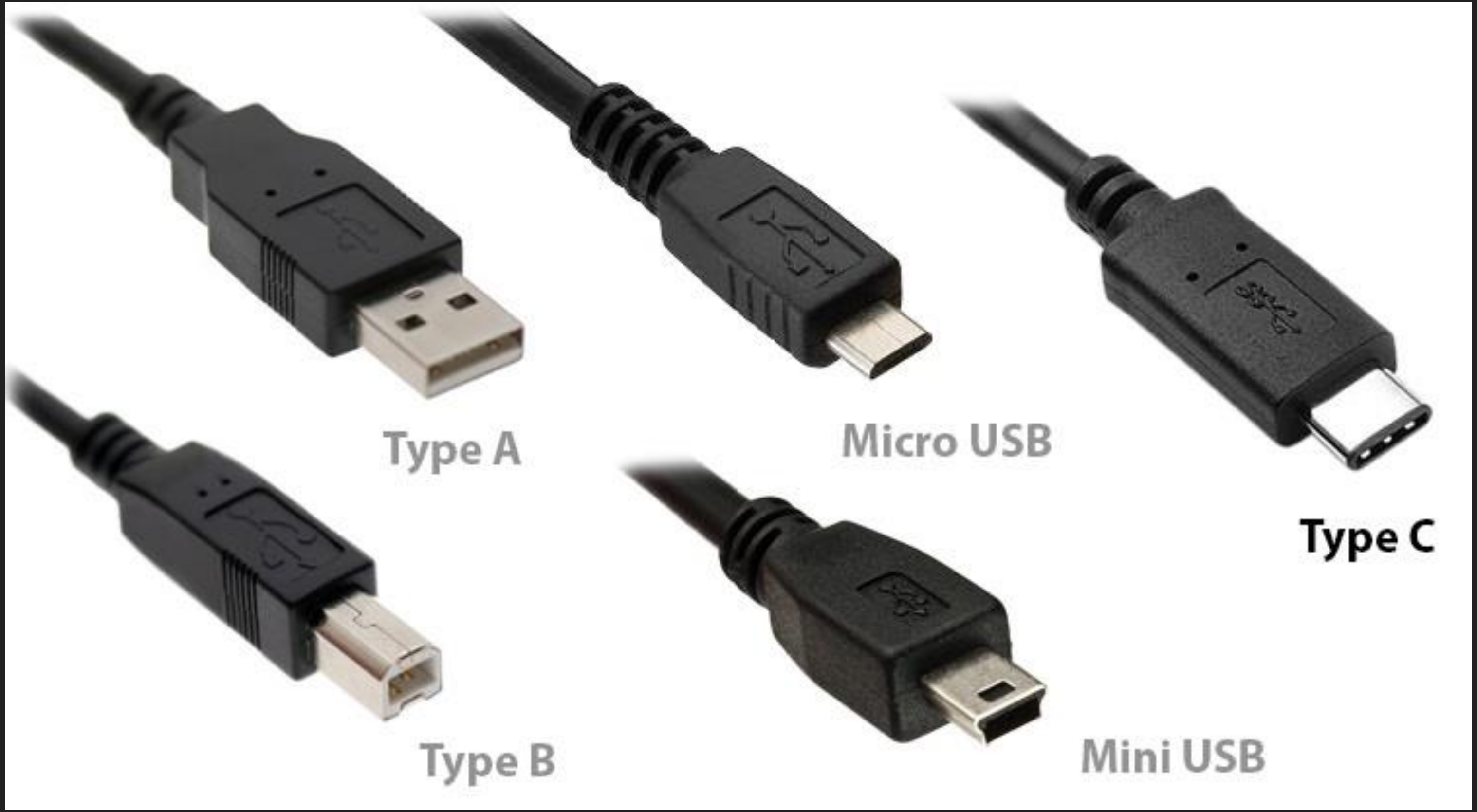
➔ de-facto standard for wired peripherals

➔ 1994 - co-invented by Ajay Bhatt of Intel and the USB-IF (USB Implementers Forum, Inc)

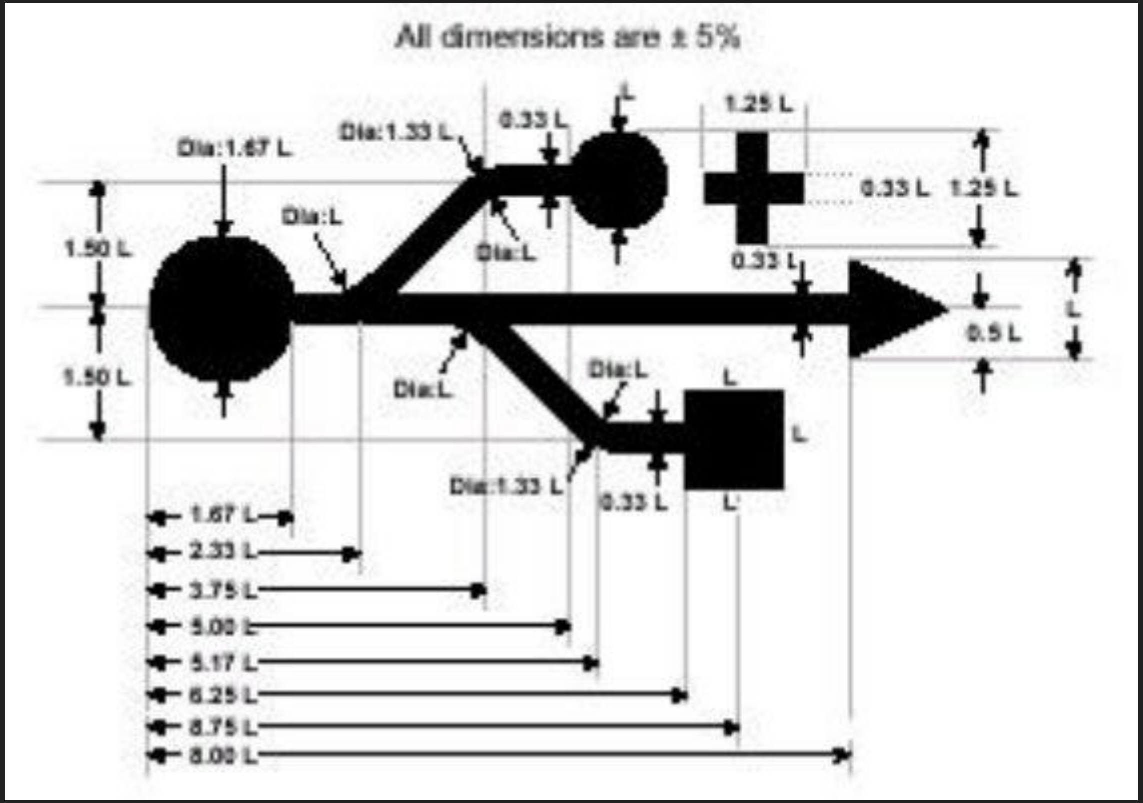
USB Versions

1. USB 4.0:
   * transfer data at 40 Gbps.
2. USB 3.1: Called Superspeed+
   * transfer data at 10 Gbps (10,240 Mbps).
3. USB 3.0: Called SuperSpeed USB,
   * maximum transmission rate of 5 Gbps (5,120 Mbps).
4. USB 2.0: Called High-Speed USB,
   * maximum transmission rate of 480 Mbps.
5. USB 1.1: Called Full Speed USB,
   * maximum transmission rate of 12 Mbps.

Types of USB



USB Logo


All dimensions are ± 5%
Dia:1.67 L
Dia:1.33 L
0.33 L
L
1.25 L
Dia:L
Dia:L
0.33 L
1.25 L
1.50 L
1.50 L
0.33 L
0.5 L
L
Dia:L
Dia:L
L
L
Dia:1.33 L
0.33 L
L
1.67 L
2.33 L
3.75 L
5.00 L
5.17 L
6.25 L
6.75 L
8.00 L

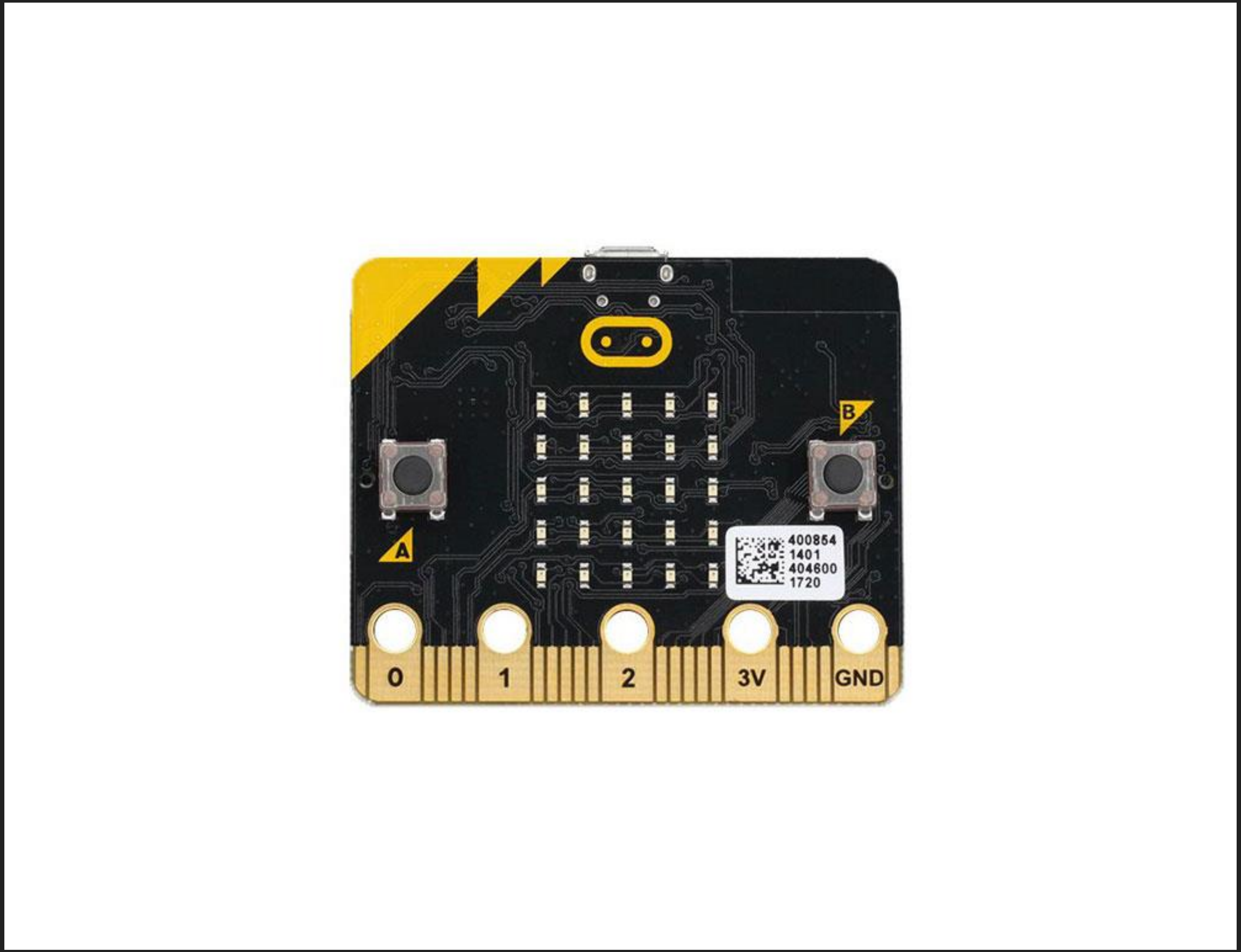
How to use the WebUSB API?

1. You have to understand how the USB standard works in order to be able to use this API.
2. uses Cross-Origin Resource Sharing (CORS)

How to use the WebUSB API?

1. Request devices
2. Connect
3. Select configuration
4. Claim interface
5. Control transfer
6. Transfer

Micro:bit



How to use the WebUSB API?

1. Request devices

```
    let devices = await navigator.usb.getDevices();
```

2. Connect

```
 const filters = [
        { vendorId: 0x2341, productId: 0x8036 }
//Arduino Leonardo
      ];
      let device = await navigator.usb.requestDevice({
filters: filters });
```

How to use the WebUSB API?

3.Select configuration

```
await this.device_.open(); // Begin a session.
 await this.device_.selectConfiguration(1);
      // Select configuration #1 for the device.
```

4.Claim interface

```
await this.device_.claimInterface(2);
   // Request exclusive control over interface #2.
```

How to use the WebUSB API?

5.Control transfer

```
await this.device_.controlTransferOut({
      requestType: "class",
      recipient: "interface",
      request: 0x22,
      value: 0x01,
      index: 0x02
     }); // Ready to receive data
```

How to use the WebUSB API?

6.Transfer

```
let readLoop = async () => {
    try {
            let result = await
    this.device_.transferIn(5, 64);
            this.onReceive(result.data);
            readLoop();
        } catch (error) {
          this.onReceiveError(error);
        }
      };
```

Chrome Debugging Tools

1. chrome://device-log
2. chrome://usb-internals

Resources

micro:bit over USB

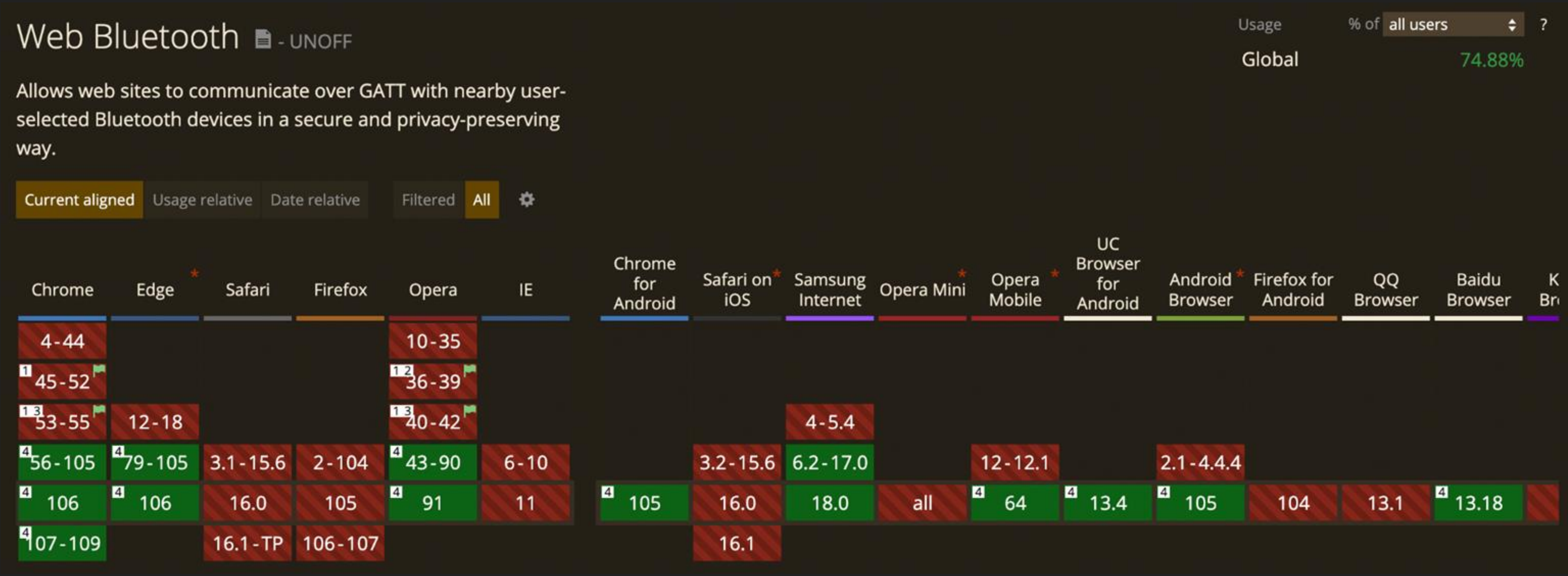
https://github.com/bsiever/microbit-webusb

https://bsiever.github.io/microbit-webusb/

WebUSB Codelab

https://codelabs.developers.google.com/codelabs/web-serial

Web USB



Web Bluetooth - UNOFF

Usage % of all users ?

Global 74.88%

Allows web sites to communicate over GATT with nearby user-selected Bluetooth devices in a secure and privacy-preserving way.

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

Chrome	Edge *	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	Opera Mobile *	UC Browser for Android	Android Browser *	Firefox for Android	QQ Browser	Baidu Browser	K Br
🟥 4-44				🟥 10-35												
🟥 [1] 45-52				🟥 [1 2] 36-39												
🟥 [1 3] 53-55	🟥 12-18			🟥 [1 3] 40-42				🟥 4-5.4								
🟩 [4] 56-105	🟩 [4] 79-105	🟥 3.1-15.6	🟥 2-104	🟩 [4] 43-90	🟥 6-10		🟥 3.2-15.6	🟩 6.2-17.0		🟥 12-12.1		🟥 2.1-4.4.4				
🟩 [4] 106	🟩 [4] 106	🟥 16.0	🟥 105	🟩 [4] 91	🟥 11	🟩 [4] 105	🟥 16.0	🟩 18.0	🟥 all	🟩 [4] 64	🟩 [4] 13.4	🟩 [4] 105	🟥 104	🟥 13.1	🟩 [4] 13.18	🟥
🟩 [4] 107-109		🟥 16.1-TP	🟥 106-107				🟥 16.1									

Other

- ➔ Web HID - provides access to HID input/output devices

  higher level of abstraction than the WebUSB and Web Bluetooth APIs
- ➔ Web NFC – ability to read and write to NFC tags

  only works on android phone via chrome

  limited to NFC Data Exchange Format (NDEF)

Summary

- ➔ Web MIDI - easiest to learn and pick up, MIDI message format
- ➔ Web Bluetooth – device has to be paired first, learn GATT
- ➔ Web USB – understand how the USB standard works first

About Me

Ron Dagdag

- Director of Software Engineering at Spacee
- 6th year Microsoft MVP awardee
- Personal Projects
  www.dagdag.net
- @rondagdag
- www.linkedin.com/in/rondagdag/
- Feedback is appreciated

https://linktr.ee/rondagdag