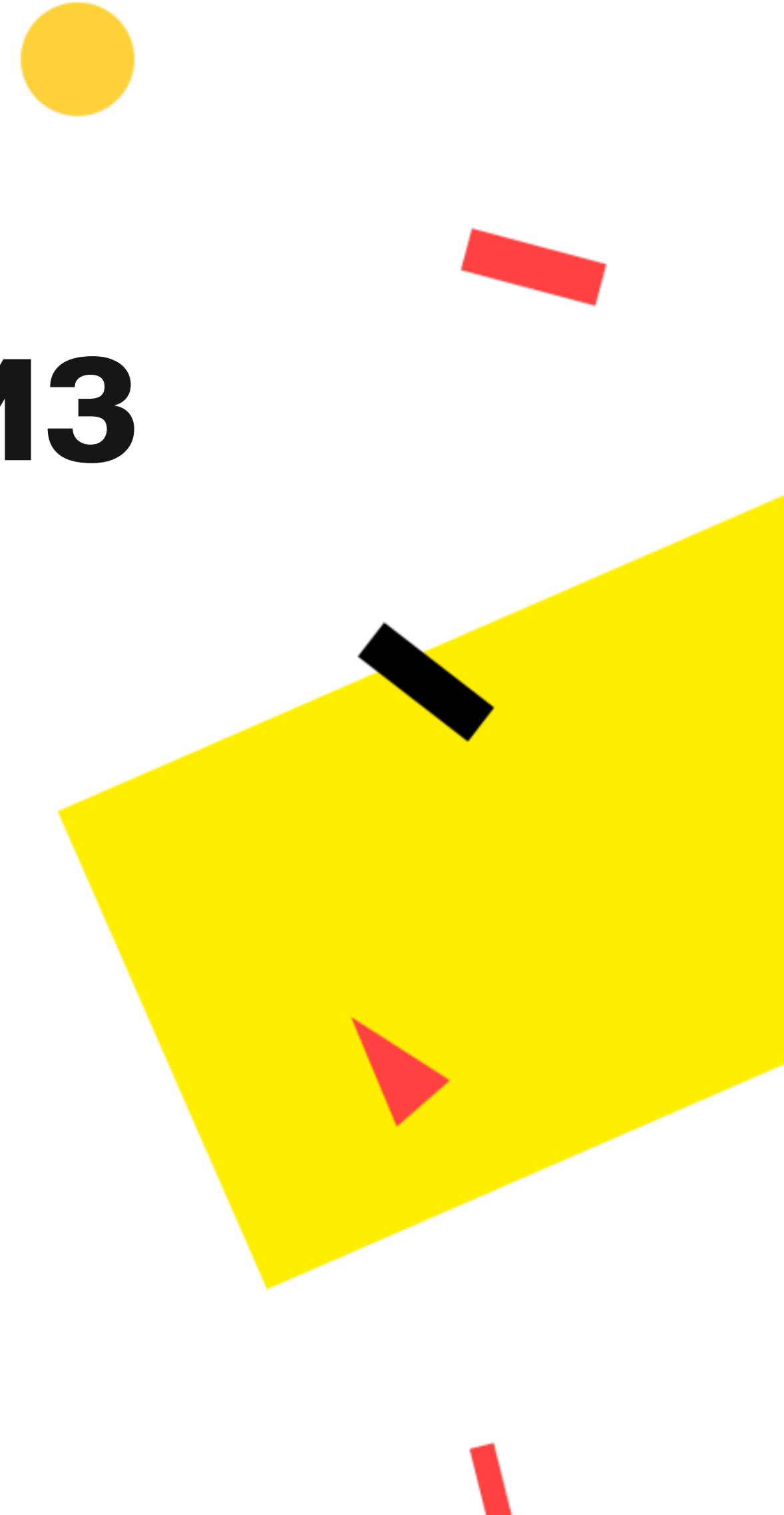


Я – фронтендер. Я управляю JS-машинкой из браузера через Bluetooth

Илья Чертопыльский



Frontend
Conf 2021



JS – Write Once Run Anywhere

Илья Чертопыльский



О себе:

Райффайзен банк - Senior Community Lead

Telegram (и т.д.) – Tetragius

Прошлые выступления:

FC-2020 – WebAssembly UI- фреймворки



Frontend
Conf 2021

О чём этот доклад?



О чём этот доклад?

Bluetooth

USB

Serial



Frontend
Conf 2021

О чём этот доклад?

WebBluetooth

WebUSB

WebSerial



Frontend
Conf 2021

О чём этот доклад?

WebBluetooth

WebUSB

WebSerial

WebHID

Gamepad

PWA

TypedArrays

WebAssembly

WebGPU

WebXR

WebAudio

WebRTC

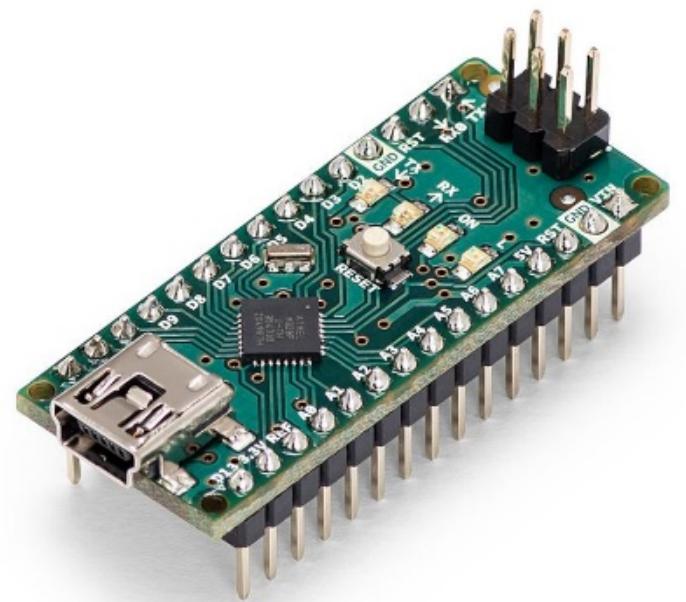
WebCrypto



Frontend
Conf 2021

О чём этот доклад?

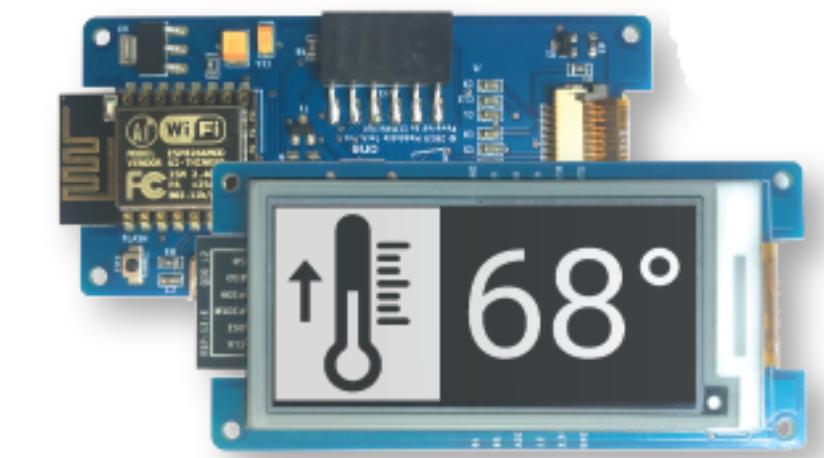
Arduino



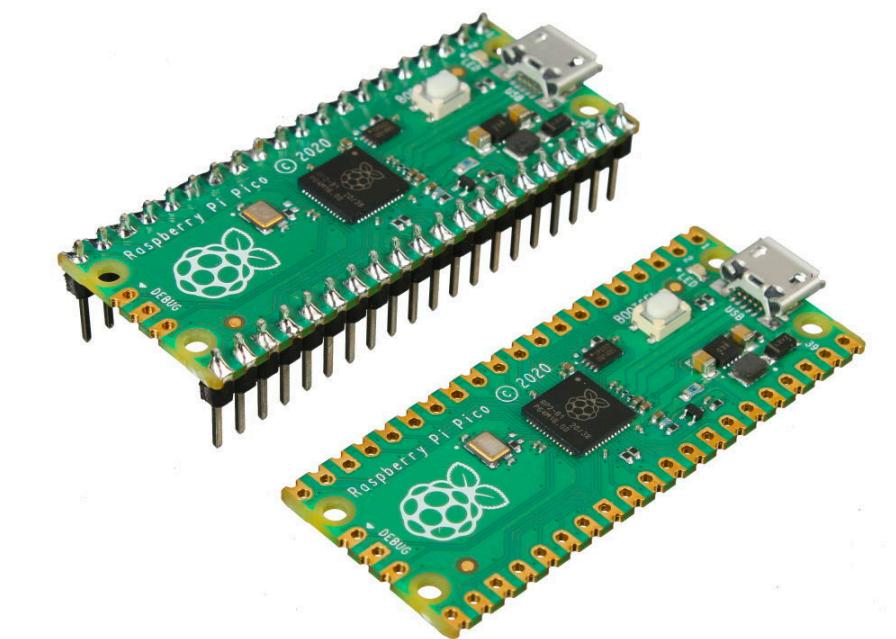
Espruino (tiny-js)



Moddable (Kinoma)



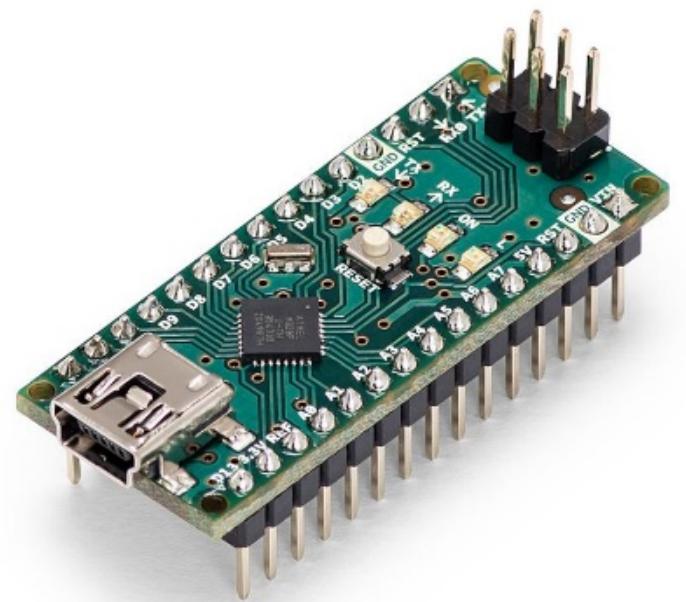
Raspberry Pico



Frontend
Conf 2021

О чём этот доклад?

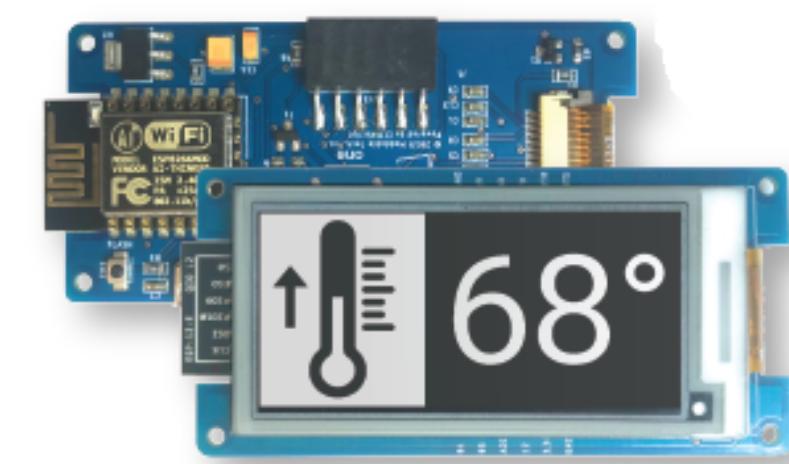
Arduino



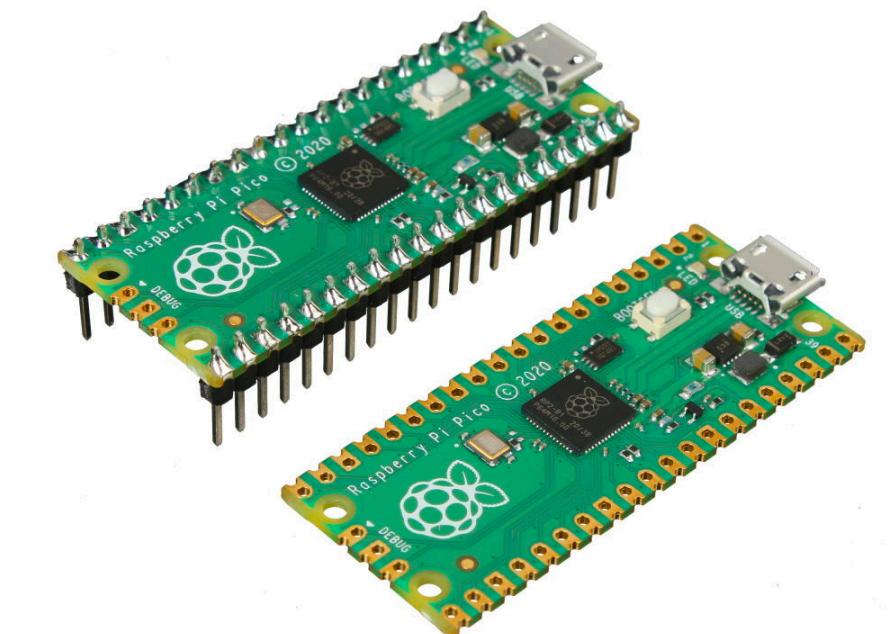
Espruino (tiny-js)



Moddable (Kinoma)



Raspberry Pi Pico



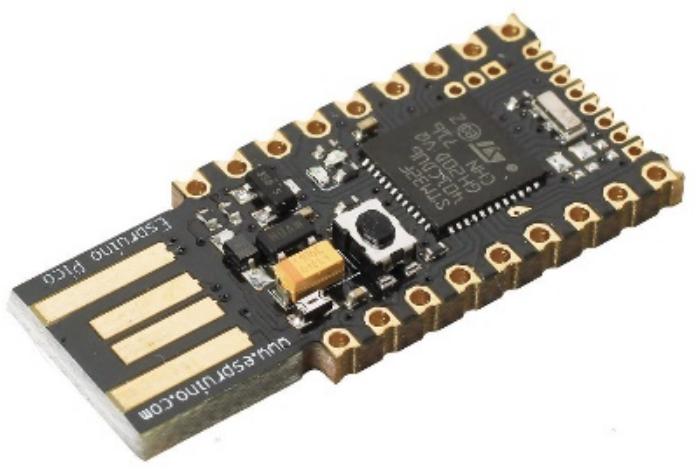
Quickjs
mJS
Duktape
SophonJS

SophonJS
IoT.js (JerryScript)
Throwback
MuJS
V7

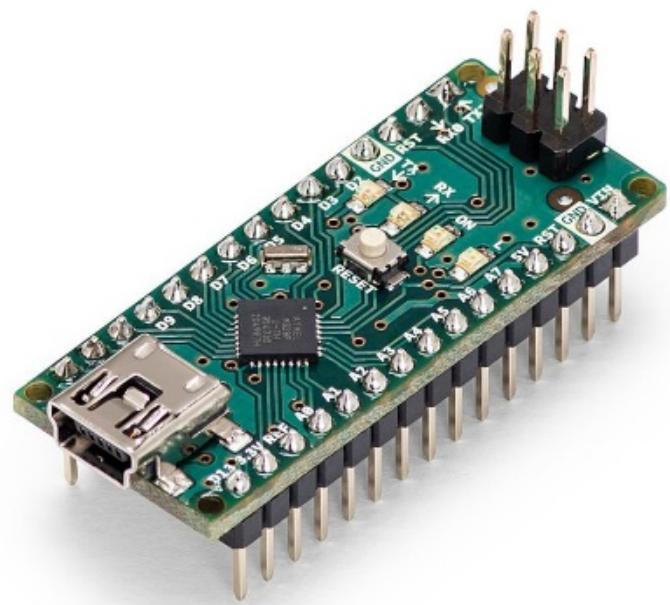
Espruino
Elk.js
Moddable
Johny-five
Kaluma

О чём этот доклад?

Espruino (tiny-js)

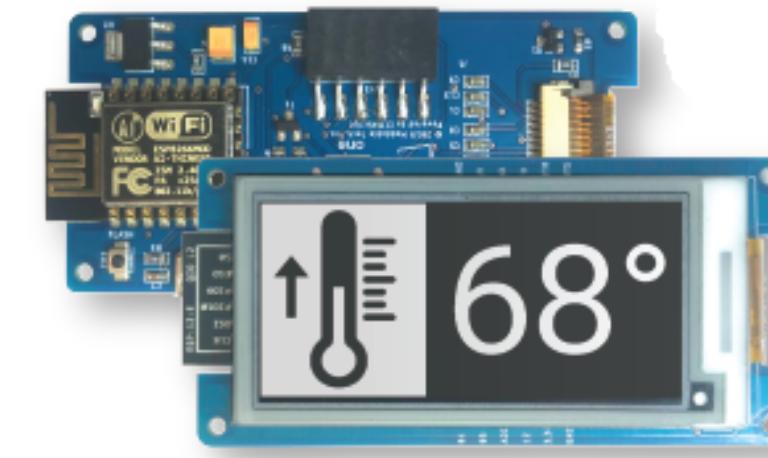


Arduino

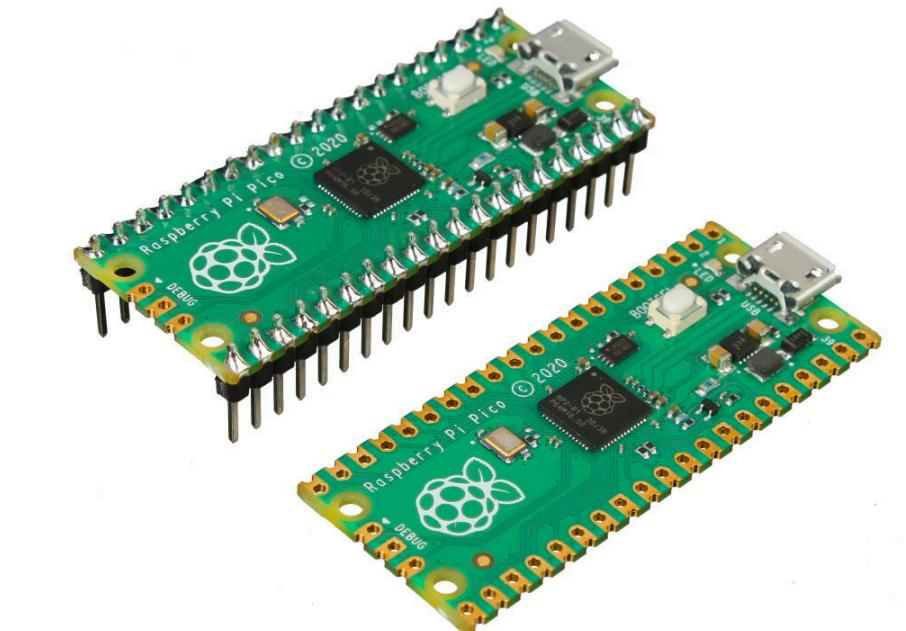


Espruino
Elk.js
Moddable
Johny-five
Kaluma

Moddable (Kinoma)



Raspberry Pi Pico



Frontend
Conf 2021

Браузерная часть



Общее для всех API

Требуется HTTPS

Требуется пользовательское действие **click** или **touch**

Расположение – `navigator.bluetooth` ; `navigator.usb` ; `navigator.serial`



Frontend
Conf 2021

WebBluetooth



Frontend
Conf 2021

WebBluetooth

Bluetooth



Frontend
Conf 2021

WebBluetooth

Bluetooth Low Energy
(BLE)

GAP
(Generic Access Profile)

GATT
(Generic Attribute Profile)

ATT
(Attribute Protocol)

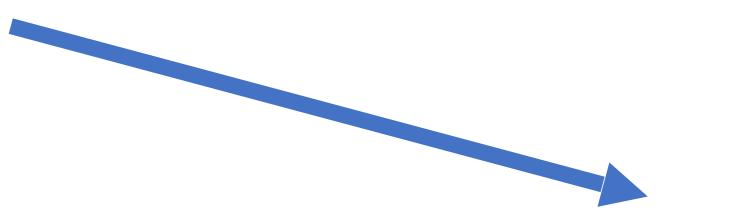
- В каком режиме работают устройства
- Как они соединяются между собой
- Определяют роль устройства
- Определяют параметры соединения



Frontend
Conf 2021

WebBluetooth

Bluetooth Low Energy
(BLE)



GATT
(Generic Attribute Profile)

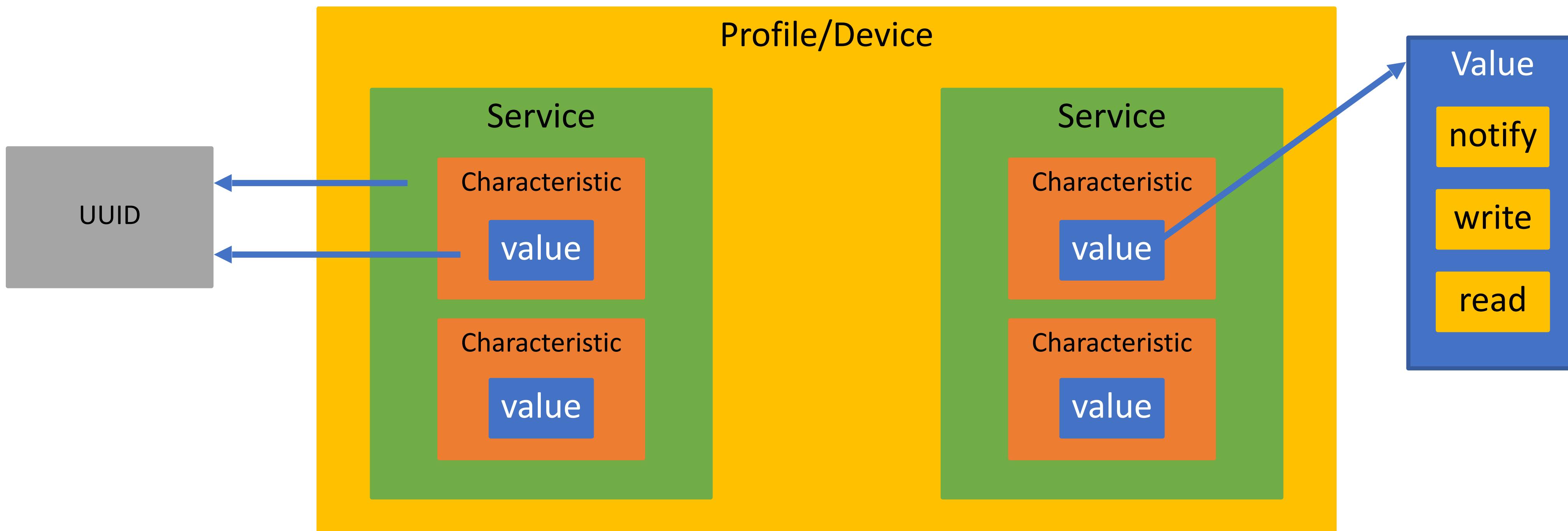
ATT
(Attribute Protocol)



Frontend
Conf 2021

WebBluetooth

Bluetooth Low Energy
(BLE)



16-bit: 0000180F-0000-1000-8000-00805F9B34FB – **Battery Service**

128-bit: 0BD51666-E7CB-469B-8E4D-2742F1BA77CC

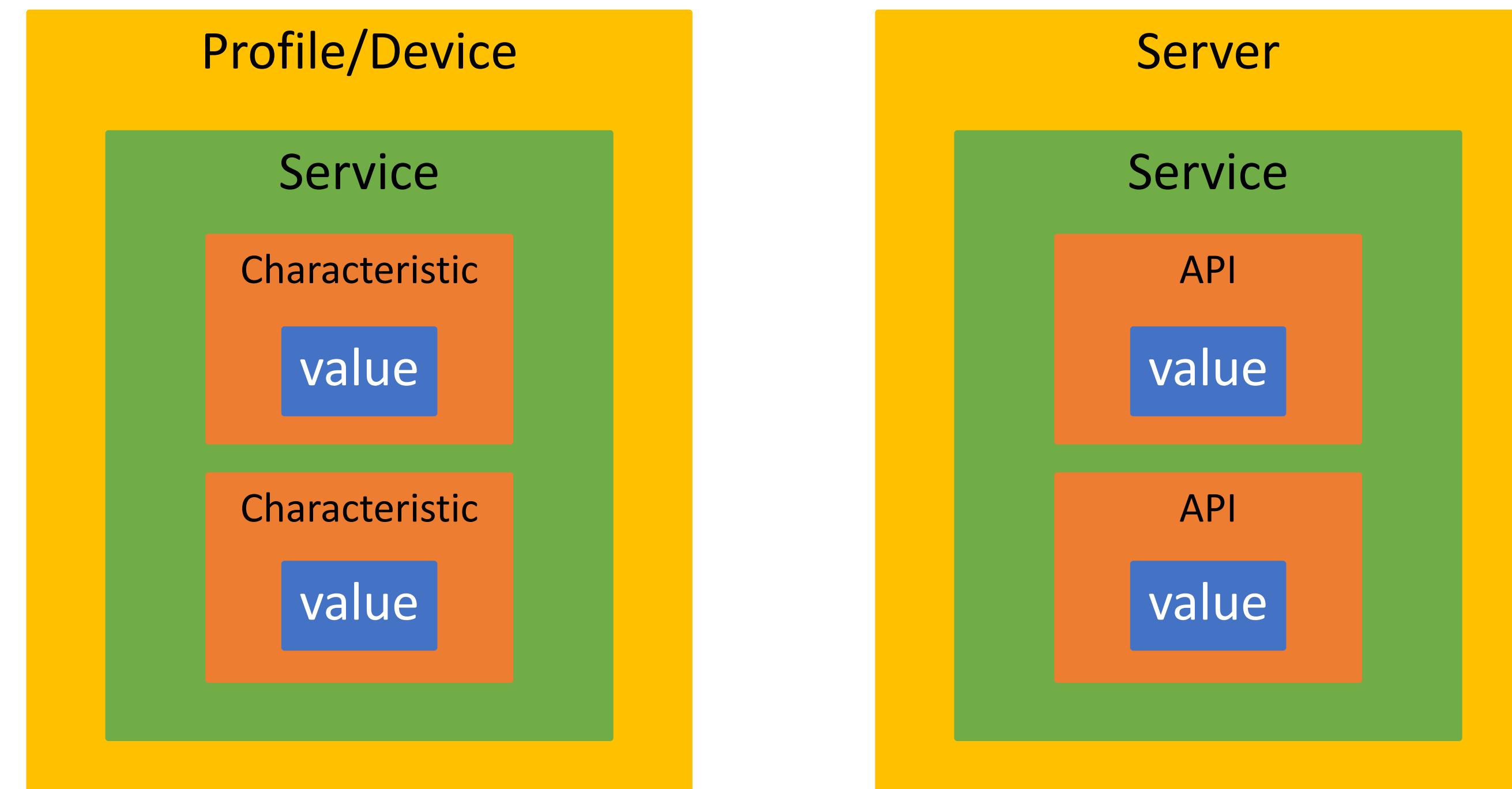


Frontend
Conf 2021

WebBluetooth

Bluetooth Low Energy
(BLE)

Client



Server



Frontend
Conf 2021

WebBluetooth

```
let device = await navigator.bluetooth.requestDevice({ filters: [ { namePrefix: 'name' } ], optionalServices: [ 0xff0f ] });
let service = await device.getPrimaryService(0xff0f);
let characteristic = await service.getCharacteristic(0xffffc);
```



```
let value = await characteristic.readValue(); // uint8Array
characteristic.writeValue( new Uint8Array([...]) );
```

```
characteristic.addEventListener( 'characteristicvaluechanged', e => { /* e.target.value */ } );
characteristic.startNotifications();
```



chrome://bluetooth-internals – полезная отладочная информация



WebBluetooth

```

navigator.bluetooth
  .requestDevice({ /* ... */ })
  .then((device) => {
    this.#disconnect = () => device.gatt.disconnect();
    this.device = device;
    return device.gatt.connect();
  })
  .then((server) => server.getPrimaryService(this.#deviceId))
  .then((service) => service.getCharacteristic(this.#characteristicId))
  .then((characteristic) => {
    this.#send = (buf) => {
      if (this.#sended) {
        this.#sended = false;
        characteristic
          .writeValue(buf)
          .then(() => (this.#sended = true));
      }
    };
    this.#characteristic = characteristic;
    this.#read = () => Promise.resolve(characteristic.value);
  });
listen = (callback) => {
  this.#characteristic.startNotifications();
  this.#characteristic.addEventListener(
    'characteristicvaluechanged',
    () => {
      this.read().then(callback);
    }
  );
}
}

```

```

stick?.addEventListener('change', ({ detail }) => {
  if (this.#BLEinstanse.device) {
    this.#BLEinstanse.send(new Uint8Array([detail.relativeX, detail.relativeY]));
  }
});

```

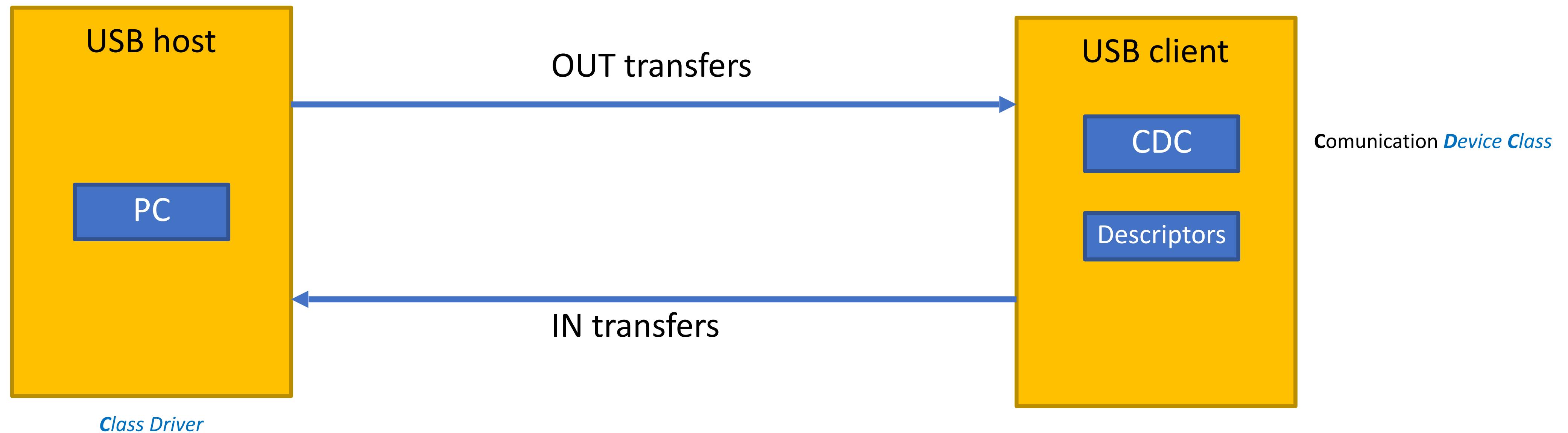


WebUSB



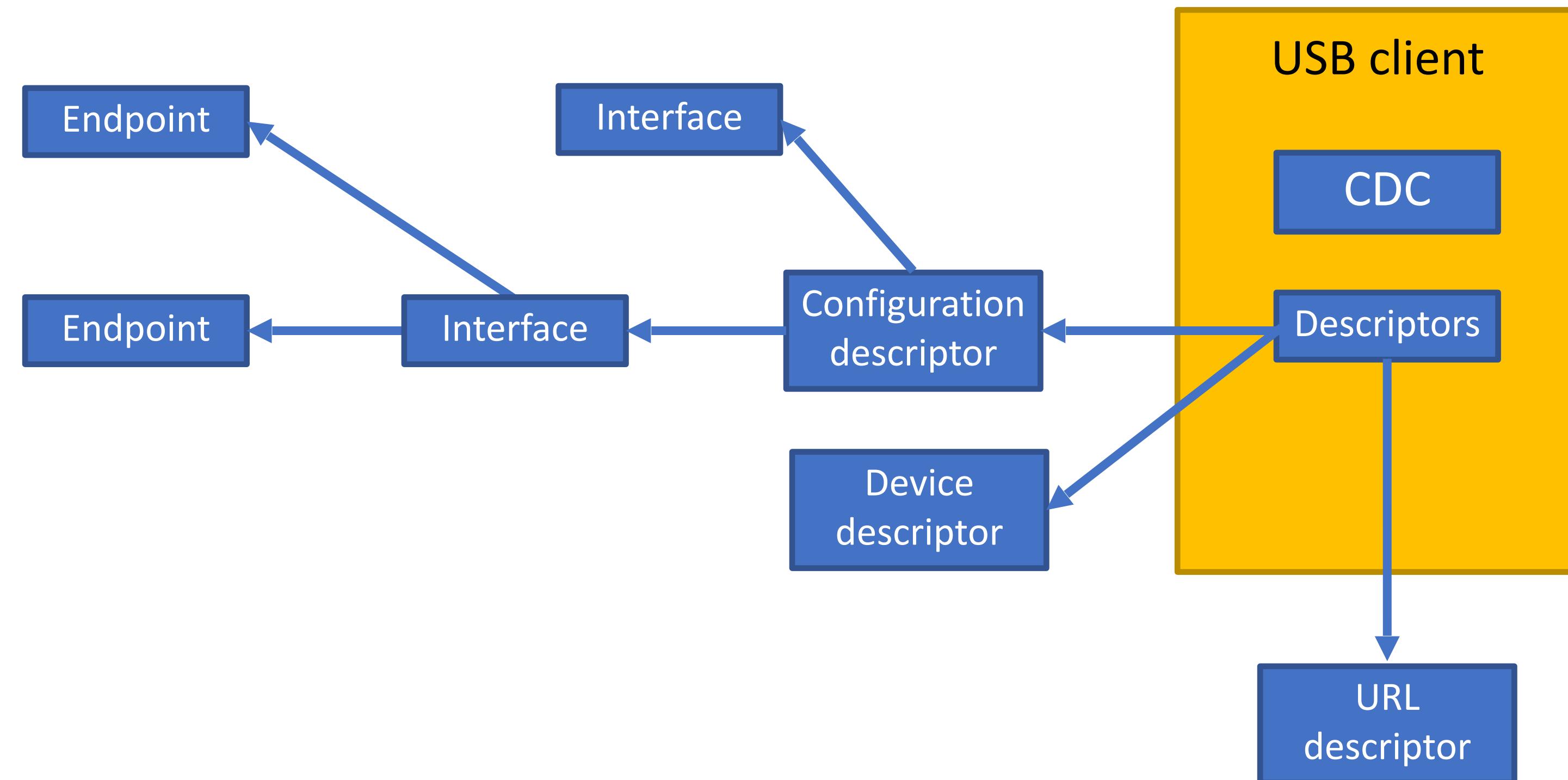
Frontend
Conf 2021

WebUSB



Frontend
Conf 2021

WebUSB



WebUSB

§ 4.3.1. URL Descriptor

This descriptor contains a single URL and is returned by the [Get URL](#) request.

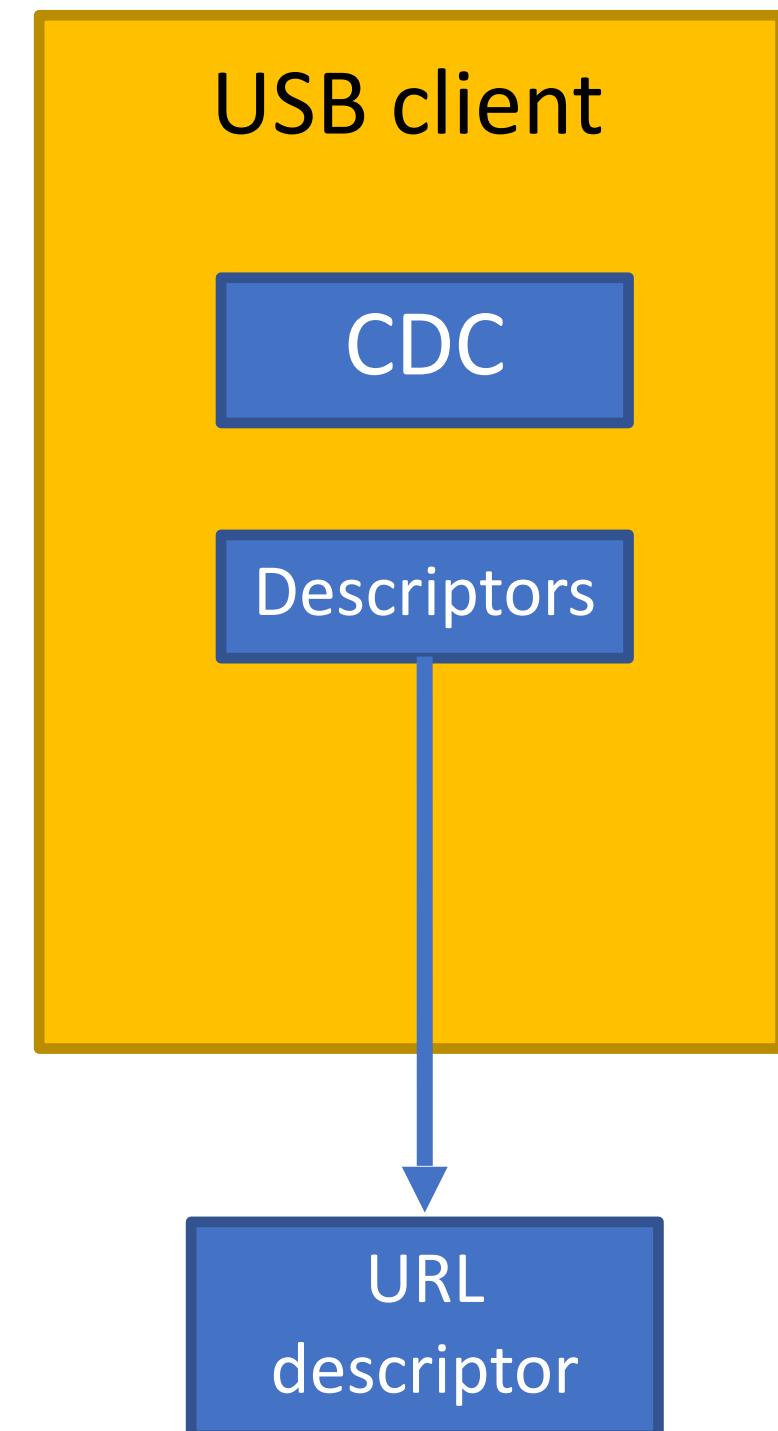
Offset	Field	Size	Type	Description
0	bLength	1	Number	Size of this descriptor.
1	bDescriptorType	1	Constant	WEBUSB_URL.
2	bScheme	1	Number	URL scheme prefix.
3	URL	Variable	String	UTF-8 encoded URL (excluding the scheme prefix).

The `bScheme` field MUST be one of these values:

URL Prefixes

Value	Prefix
0	"http://"
1	"https://"
255	""

The special value 255 indicates that the entire URL, including scheme, is encoded in the `URL` field.



WebUSB

```
#include <WebUSB.h>
```

```
WebUSB WebUSBSerial(1 /* https:// */, "webusb.github.io/arduino/demos");
```

URL descriptor

URL Prefixes

Value	Prefix
0	"http://"
1	"https://"
255	""



Frontend
Conf 2021

WebUSB

```
let device = await navigator.usb.requestDevice({ filters: [ { vendorId: 0x2341 } ] });
let device = await navigator.usb.getDevices();

device.open();
device.selectConfiguration(1);
device.claimInterface(2);
device.controlTransferOut({ ... });
device.transferIn(5, 64);
```

↑
endpoint
↓

```
device.transferOut(5, [TypedArray]);
```

chrome://usb-internals – полезная отладочная информация



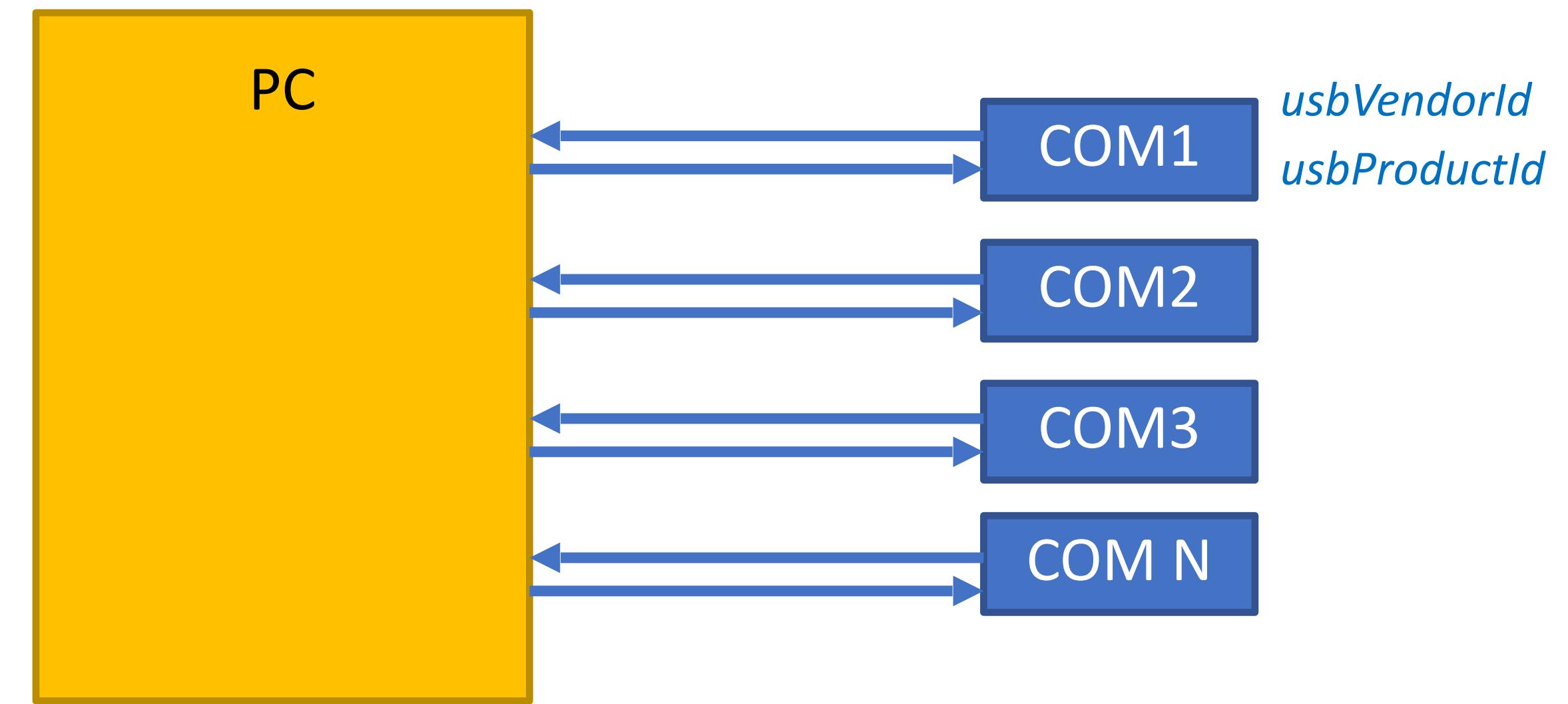
Frontend
Conf 2021

WebSerial



Frontend
Conf 2021

WebSerial



WebSerial

```
let port = await navigator.serial.requestPorts({ filters });
```

```
let port = await navigator.serial.getPorts();
```

```
await port.open({ baudRate: 9600 });
```

```
const reader = port.readable.getReader();
```

```
const { value, done } = await reader.read();
```

```
const writer = port.writeable.getWriter();
```

```
await writer.write([Uint8Array]);
```

chrome://device-log – полезная отладочная информация



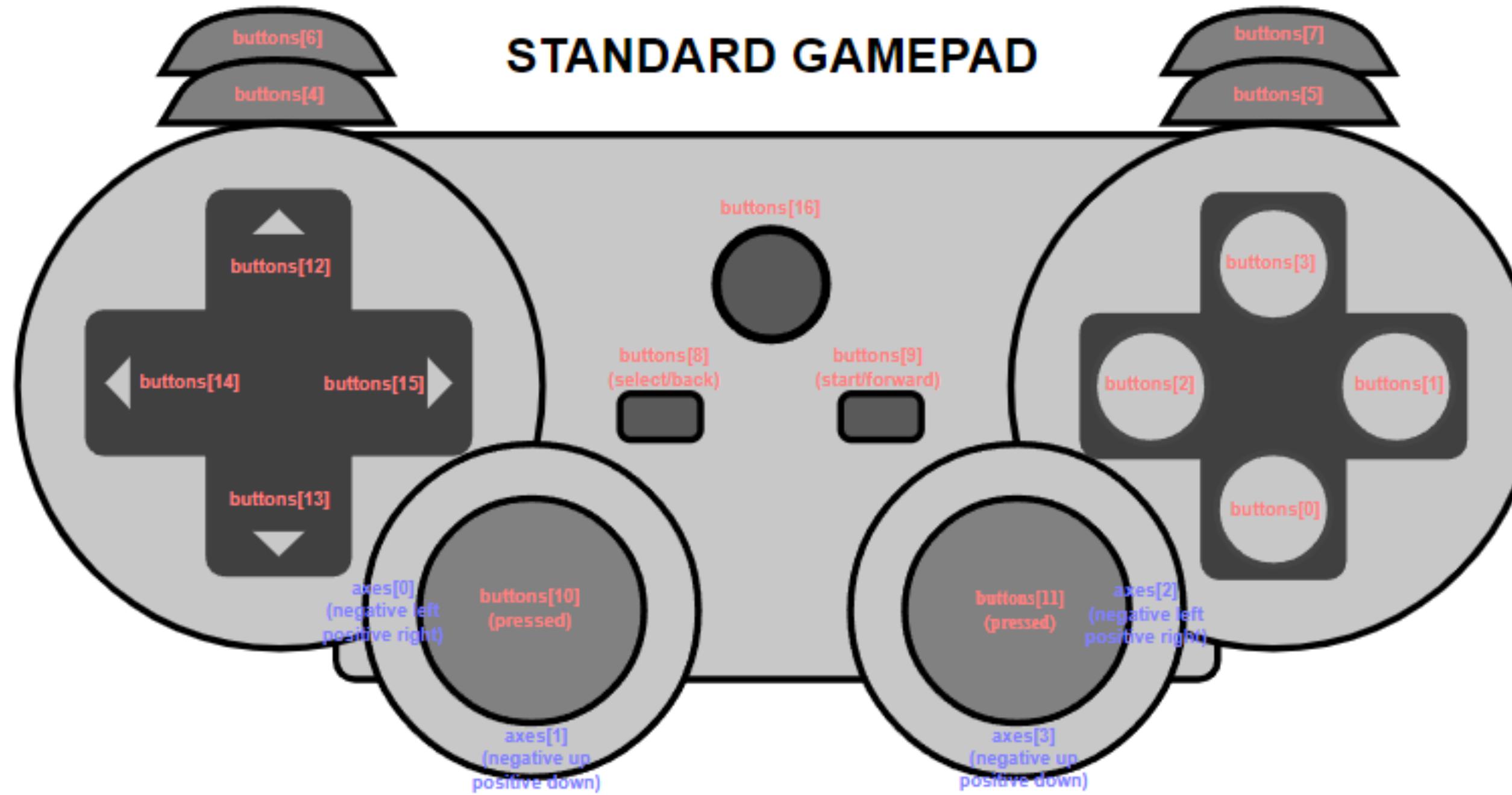
Frontend
Conf 2021

Gamepad



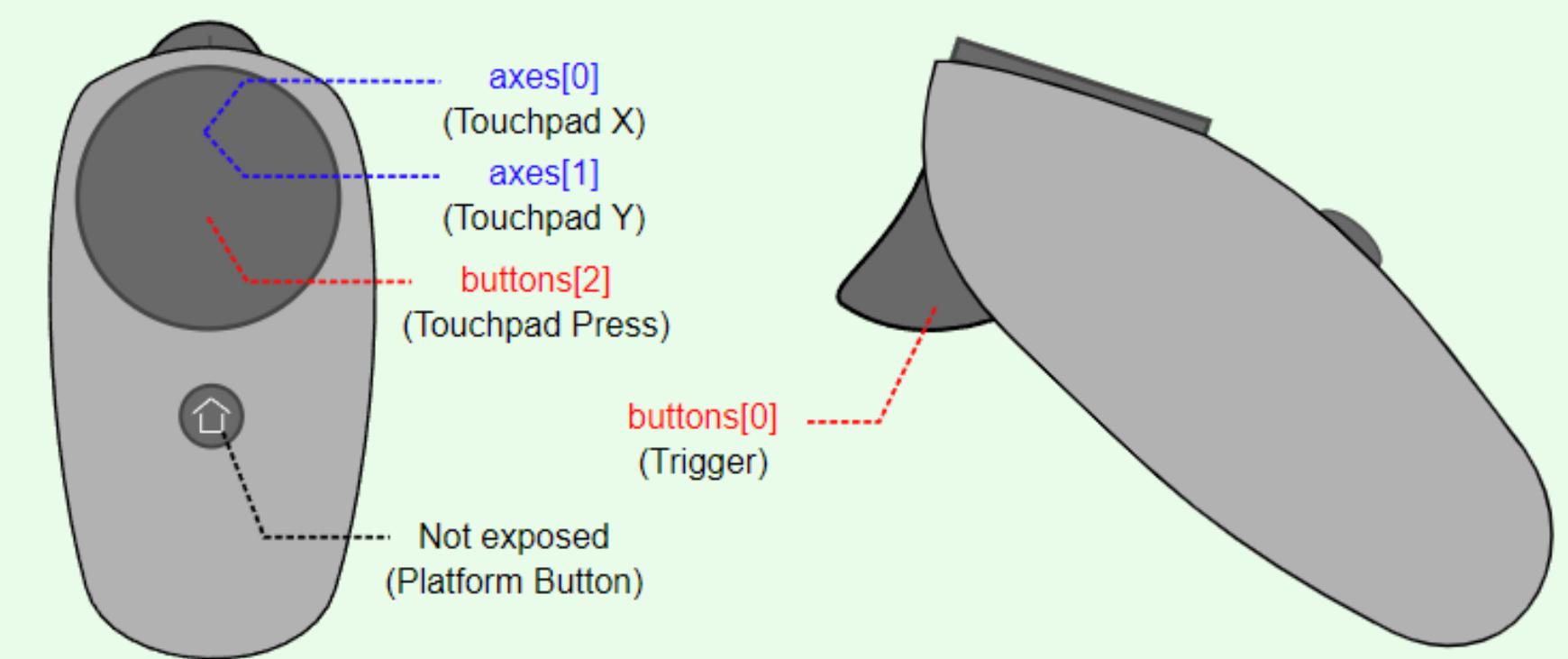
Frontend
Conf 2021

Gamepad

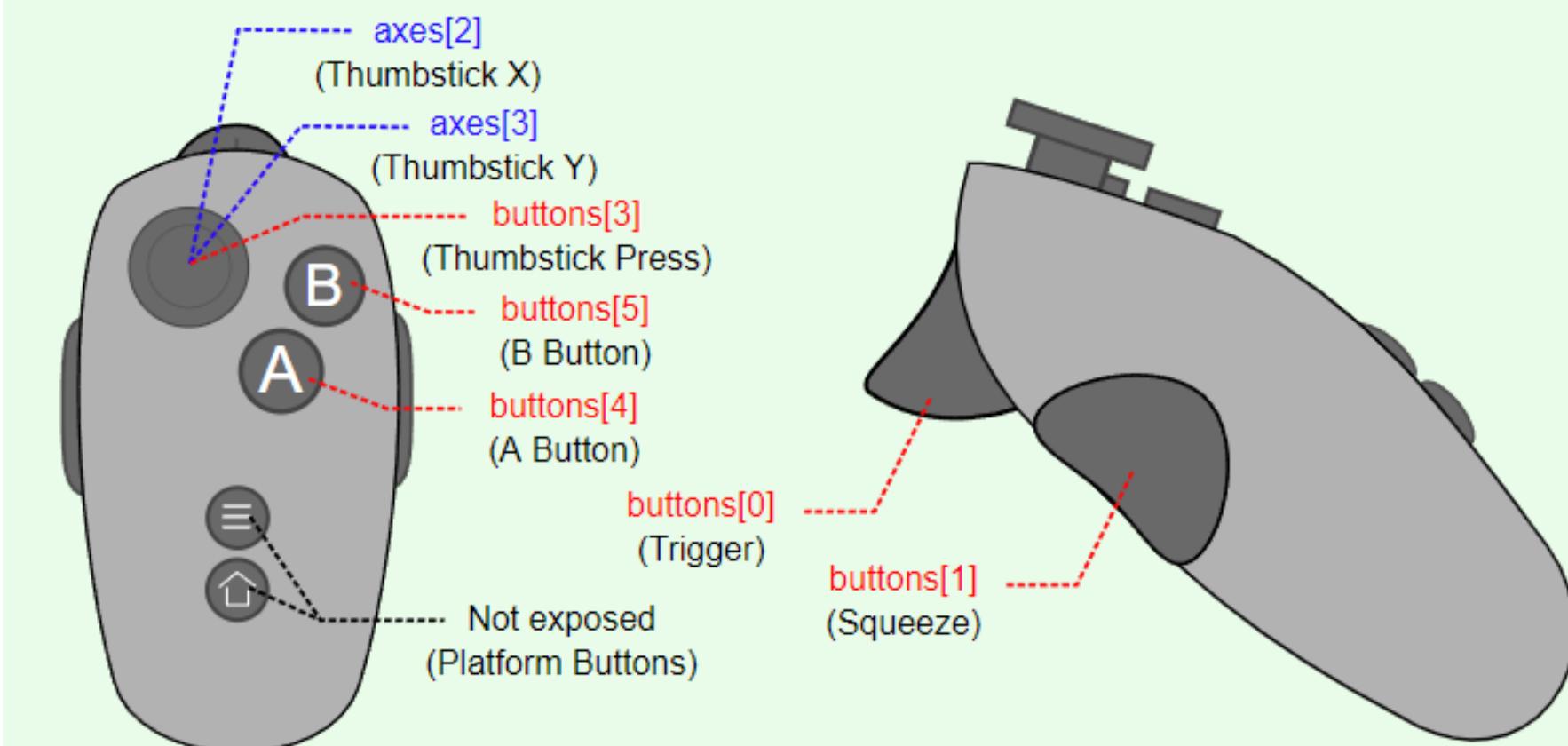


```
const gamepad = navigator.getGamepads()[0];
console.log(gamepad.buttons[0].value);
```

Example basic 'xr-standard' controller



Example complex 'xr-standard' controller



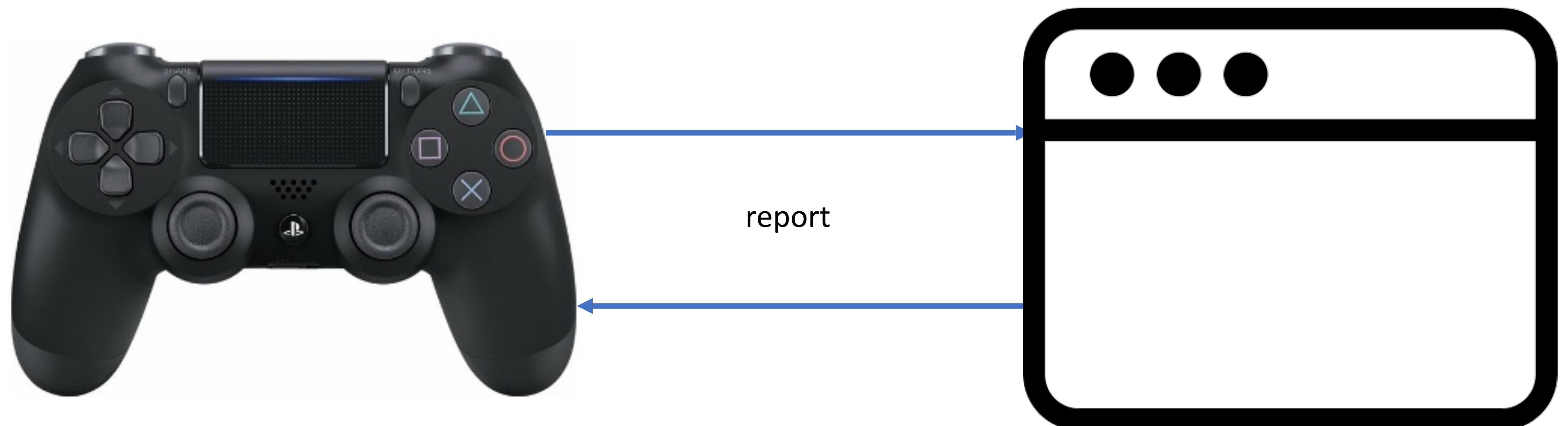
NOTE: For this device axes[0], axes[1], and buttons[2] are placeholders, since they are reserved for Touchpad inputs

WebHID



Frontend
Conf 2021

WebHID



WebHID

```

const devices = await navigator.hid.requestDevice({});
const controller = devices[0];
await controller.open();
controller.oninputreport = (e) => {
  const report = e.data;
  // Кнопки с пиктограммами
  const mainButtons = report.getInt8(4);
  data['triangle'] = !(mainButtons & 0x80);
  data['circle'] = !(mainButtons & 0x40);
  data['cross'] = !(mainButtons & 0x20);
  data['square'] = !(mainButtons & 0x10);
  // Кнопки со стрелками
  const dPad = mainButtons & 0x0F;
  data['dPadUp'] = dPad === 7 || dPad === 0 || dPad === 1;
  data['dPadRight'] = dPad === 1 || dPad === 2 || dPad === 3;
  data['dPadDown'] = dPad === 3 || dPad === 4 || dPad === 5;
  data['dPadLeft'] = dPad === 5 || dPad === 6 || dPad === 7;
};

```

byte index	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
[0]	Report ID (USB)							
[1]		X axis (0 = left)						
[2]		Y axis (0 = up)						
[3]		Right Stick axis X						
[4]		Right Stick axis Y						
[5]					D-PAD : hat format (0x08 is released)			
	1000							
	0111	0110	0101	0100	0011	0010	0001	0000
	7=NW	6=W	5=SW	4=S	3=SE	2=E	1=NE	0=N
	↑←	←	↓←	↓	↓→	→	↑→	↑
[6]					R2 (8)	L2 (4)	R1 (2)	L1 (1)



WebHID

```

const mainButtons = report.getInt8(4);
// [item] → item → 8bit → 00000000
// 01000100 → parseInt('01000100', 2) → 68;
data['circle'] = !(mainButtons & 0x40);
// 68 & 0x40 = 64
// 68 → 01000100 &
// (0x40).toString(2) → 01000000 ⇒ true
const dPad = mainButtons & 0x0F;
// 68 → 01000100 &
// (0x0F).toString(2) → 00001111;
// 00000100;

data['dPadDown'] = dPad === 3
|| dPad === 4 // ! 100 ⇒ 4
|| dPad === 5;

```

byte index	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
[0]	Report ID (USB)							
[1]	X axis (0 = left)							
[2]	Y axis (0 = up)							
[3]	Right Stick axis X							
[4]	Right Stick axis Y							
D-PAD : hat format (0x08 is released)								
1000								
[5]	△	○	×	□				
	0111	0110	0101	0100	0011	0010	0001	0000
	7=NW	6=W	5=SW	4=S	3=SE	2=E	1=NE	0=N
	↑↔	↔	↓↔	↓	↓↔	→	↑↔	↑
[6]	R3	L3	OPTIONS	SHARE	R2 (8)	L2 (4)	R1 (2)	L1 (1)



WebHID

```

const devices = await navigator.hid.requestDevice({
  filters: [ ... ]
});
const controller = devices[0];
await controller.open();
let flag = true;
const set = () => {
  const report = new Uint8Array(16);
  report[0] = 0x05; // Report ID
  report[1] = 0xF0 | 0x01 | 0x02; // Моторчики (0x01), Светодиод (0x02)
  report[4] = 0x00; // Малый моторчик
  report[5] = 0x00; // Большой моторчик
  report[6] = flag ? 0x00 : 0xFF; // Красный
  report[7] = 0x00; // Зеленый
  report[8] = !flag ? 0x00 : 0xFF; // Синий
  flag = !flag;
  controller.sendReport(report[0], report.slice(1));
}
setInterval(set, 500);

```



Аппаратная часть



JS – интерпретируемый

Чтобы запустить JS нужен интерпретатор

Интерпретатор может быть написан на чем угодно



Интерпретатор может быть запущен на чем угодно

Значит JS может быть запущен почти на любом устройстве



Frontend
Conf 2021

Как запустить JavaScript на устройстве?

1. Скачать Node.js (или Deno)
2. Набрать node example.js
3. PROFIT!!!



Справедливо для одноплатных компьютеров



Frontend
Conf 2021

Как запустить JavaScript на устройстве?

1. Выбрать МК
2. Выбрать «встраиваемый» движок
3. Изучить API
4. Скачать toolchain
5. Прошить МК интерпретатором
6. Прошить МК полезным кодом
7. Повторить шаги с 1 по 6 при необходимости



Frontend
Conf 2021

Микроконтроллеры и микропроцессоры

Микропроцессоры

Микроконтроллеры

SoC



Frontend
Conf 2021

Микроконтроллеры и микропроцессоры

Микропроцессоры	Микроконтроллеры	SoC
CPU Central processing unit	MCU Micro-controller unit	System on Chip



Frontend
Conf 2021

Микроконтроллеры и микропроцессоры

Микропроцессоры	Микроконтроллеры	SoC
CPU Central processing unit	MCU Micro-controller unit	System on Chip
ARM (Advanced RISC Machine)	AVR (Advanced Virtual RISC)	



Frontend
Conf 2021

Микроконтроллеры и микропроцессоры

Микропроцессоры

CPU
Central processing unit

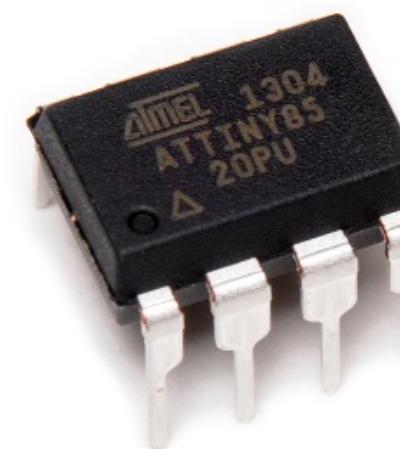
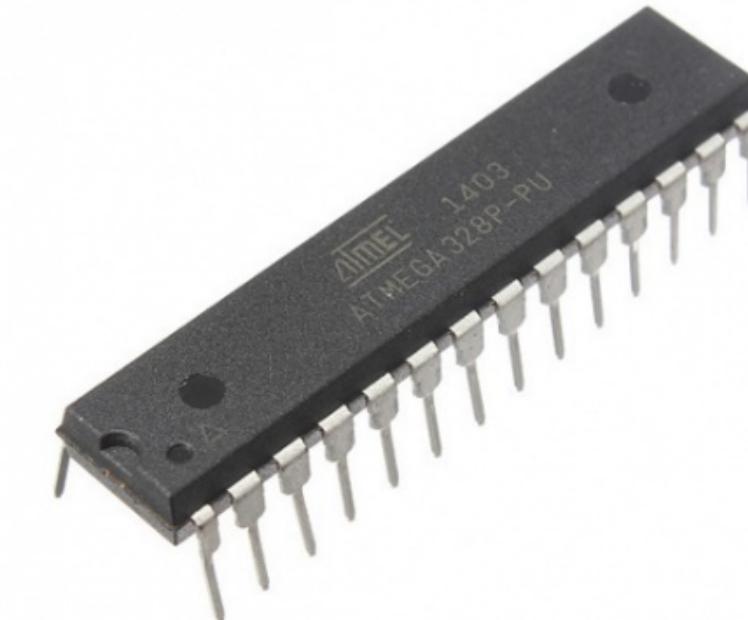


Только процессор

Микроконтроллеры

MCU
Micro-controller unit

Имеют память и периферические интерфейсы, таймеры, счетчики



Frontend
Conf 2021

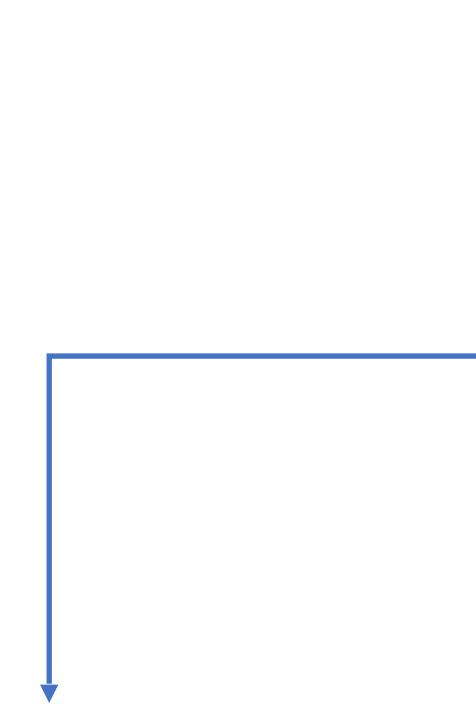
Микроконтроллеры и микропроцессоры

Микропроцессоры	Микроконтроллеры	SoC
CPU Central processing unit	MCU Micro-controller unit	System on Chip
Только процессор	Имеют память и периферические интерфейсы, таймеры, счетчики	
	Arduino Arduino-подобные	Espruino Moddable Kaluma



Johny-five (cylonjs)

Скетч с Firmata (есть в Arduino IDE)



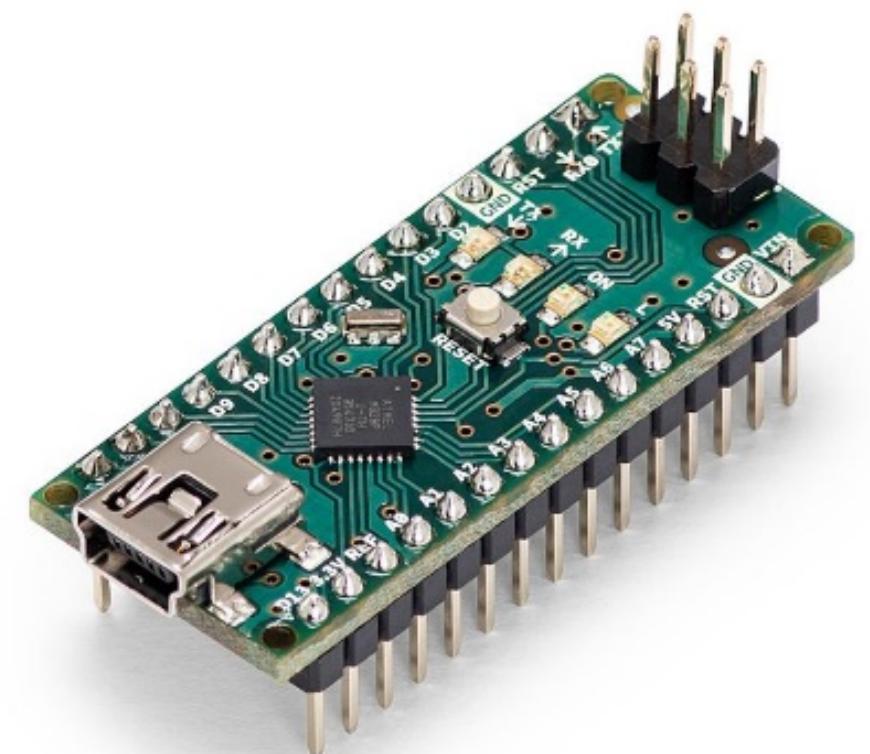
```
const { Board, Led } = require('johnny-five');
const board = new Board({port: 'COM6'});
board.on('ready', () => {
  const led = new Led(13);
  led.blink(500);
});
```



Frontend
Conf 2021

Elk.js

```
#include "elk.h"
void myDelay(int milli) { delay(milli); }
void myWrite(int pin, int val) { digitalWrite(pin, val); }
void myMode(int pin, int mode) { pinMode(pin, mode); }
}
char buf[300];
void setup() {
    struct js *js = js_create(buf, sizeof(buf));
    jsval_t global = js_glob(js), gpio = js_mkobj(js);
    js_set(js, global, "gpio", gpio);
    js_set(js, global, "delay", js_import(js, (uintptr_t) myDelay, "vi"));
    js_set(js, gpio, "mode", js_import(js, (uintptr_t) myMode, "vii"));
    js_set(js, gpio, "write", js_import(js, (uintptr_t) myWrite, "vii"));
    js_eval(js, "let pin = 13;" +
        "gpio.mode(pin, 1);"
        "while (true) {" +
        "  delay(300);"
        "  gpio.write(pin, 1);"
        "  delay(300);"
        "  gpio.write(pin, 0);"
        "}"
        "~0);
}
```



Espruino

```

E.on('init', function () {

  const servo = new ServoHW(D33);

  servo.write(90);

  pinMode(D2, 'output'); D2.reset(); analogWrite(D2, 0);
  pinMode(D5, 'output'); D5.reset(); analogWrite(D5, 0);

  Serial2.setup(115200, { tx: D4, rx: D15 });

  const US = new Ultrasonic({ trigPin: D12, echoPin: D14 });

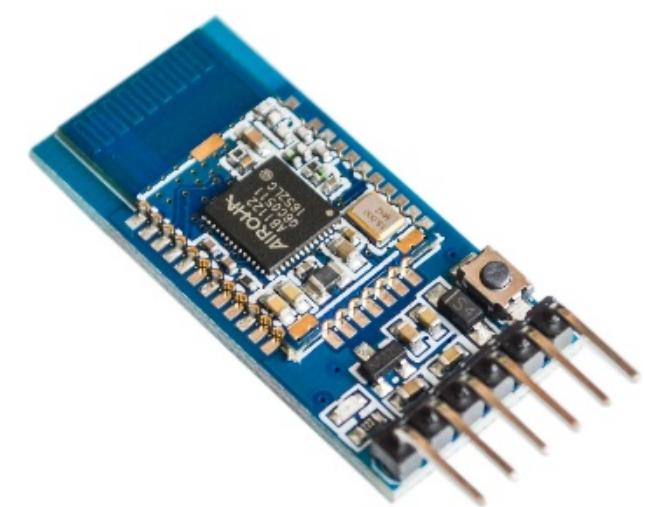
  setInterval(function () {
    US.ping(function (err, value) {
      if (!err) {
        Serial2.print(value);
      }
    }, 'mm');
  }, 200);

  Serial2.on('data', function (data) {
    const a = parseInt(data.charCodeAt(0) * 0.71);
    const v = (data.charCodeAt(1) * 0.008) - 1;

    if (v > 0) { analogWrite(D2, v); analogWrite(D5, 0); }
    if (v === 0) { analogWrite(D2, 0); analogWrite(D5, 0); }
    if (v < 0) { analogWrite(D2, 0); analogWrite(D5, -1 * v); }

    servo.write(a);
  });
});

```

Espruino

```

const wifi = require("Wifi");
const lcd = require("SSD1306");
const http = require('ws');

I2C1.setup({ scl: D27, sda: D25 });
const g = lcd.connect(I2C1, () => { }, { rst: D26 });

const pageHandler = (req, res) => {
    const url = req.url;
    const file = E.openFile(`/${rootPath}/${url}`, 'r');
    res.writeHead(200, {
        'Content-Type': 'text/html',
        'Content-Encoding': 'gzip',
        'Vary': 'Accept-Encoding',
        'Cache-Control': 'public, max-age=31536000'
    });
    file.pipe(res, { chunkSize: 1024 });
};

};


```

```

function onInit() {
    E.connectSDCard(SPI1, D16);
    wifi.connect(ssid, option, () => {
        const address = wifi.getIP(() => {
            const srv = http.createServer();
            srv.on('websocket', wsHandler);
            srv.listen(80);
        });
        print(address.ip);
    });
    wifi.stopAP();
}

```



Espruino

```
function drawClock() {  
    var t = new Date();  
    var h = t.getHours();  
    var m = t.getMinutes();  
    var time = (("0" + h).substr(-2) + ("0" + m).substr(-2)).split('');  
    g.setFont("8x16", 4);  
    g.setFontAlign(0, 0, 0);  
    g.setColor("#fff");  
    g.drawString(time[0], 48, 140);  
    g.drawString(time[1], 96, 140);  
    g.drawString(time[2], 154, 140);  
    g.drawString(time[3], 200, 140);  
    g.setFont("8x16", 2);  
    g.setFontAlign(-1, -1, 0);  
    g.setColor("#fff");  
    g.drawString('Raiffeisen', 52, 190);  
}  
  
g.clear();  
Bangle.setUI("clock");  
Bangle.loadWidgets();  
Bangle.drawWidgets();  
setInterval(drawAll, 1E4);  
drawAll();
```



Kaluma и Moddable

The screenshot shows the Espruino IDE interface. At the top, there's a navigation bar with icons for file operations (New, Open, Save, etc.), a search bar, and settings. The main area has a dark theme with a sidebar on the left containing a logo, the title "Espruino IDE", a version "Web App, v0.75.11", and a "Recent News" section listing various posts with dates. Below this is a "Bangle.js 2 is now on KickStarter!" banner with a link to check it out. The right side features a code editor with a scrollable text area showing a snippet of JavaScript code for a Bangle.js watch face. The code handles date and time calculations, color schemes for different seconds markers, and drawing hands and seconds on a circular canvas. A status bar at the bottom says "READY...".

moddable

[Home](#) [Moddable on GitHub](#) [License](#) [About ▾](#) [Blog](#)

JavaScript for IoT

Tools to create open IoT products using standard JavaScript on low cost microcontrollers

Articles and Resources

Development Hardware by Moddable

Moddable has three development boards featuring beautiful IPS screens. Check out the order pages for the hardware specifications and our [GitHub repository](#) for the latest software.

[Purchase](#)

[Products](#)

IoT Development for ESP32 and ESP8266 with JavaScript

Not sure where to start? Our new book is the fastest way to learn how to use Moddable SDK in your commercial IoT product or weekend project. There's even an introduction to modern JavaScript just for professional embedded developers working in C. [Read more](#)

JavaScript for Embedded

Moddable is the only embedded engine provider on the JavaScript language committee. We joined to ensure JavaScript remains a great language for use in embedded. Here's our original presentation to the committee. [Read more](#)

Licensing

Licensing terms are a key factor in selecting embedded software. To meet the full range of licensing needs, we offer two licensing options for the Moddable SDK: a FOSS (Free and Open Source Software) license and a traditional commercial license. [Read more](#)



Экосистема



Frontend
Conf 2021

<https://wokwi.com/>
<https://www.tinkercad.com/dashboard>
<https://create.arduino.cc/editor/>

} Симуляторы микроконтроллеров

<https://www.circuito.io/>
<http://falstad.com/circuit/>
<http://falstad.com/circuit/avr8js/>
<https://everycircuit.com/app>
<http://opencircuits.net/register>
<https://www.partsim.com/simulator>

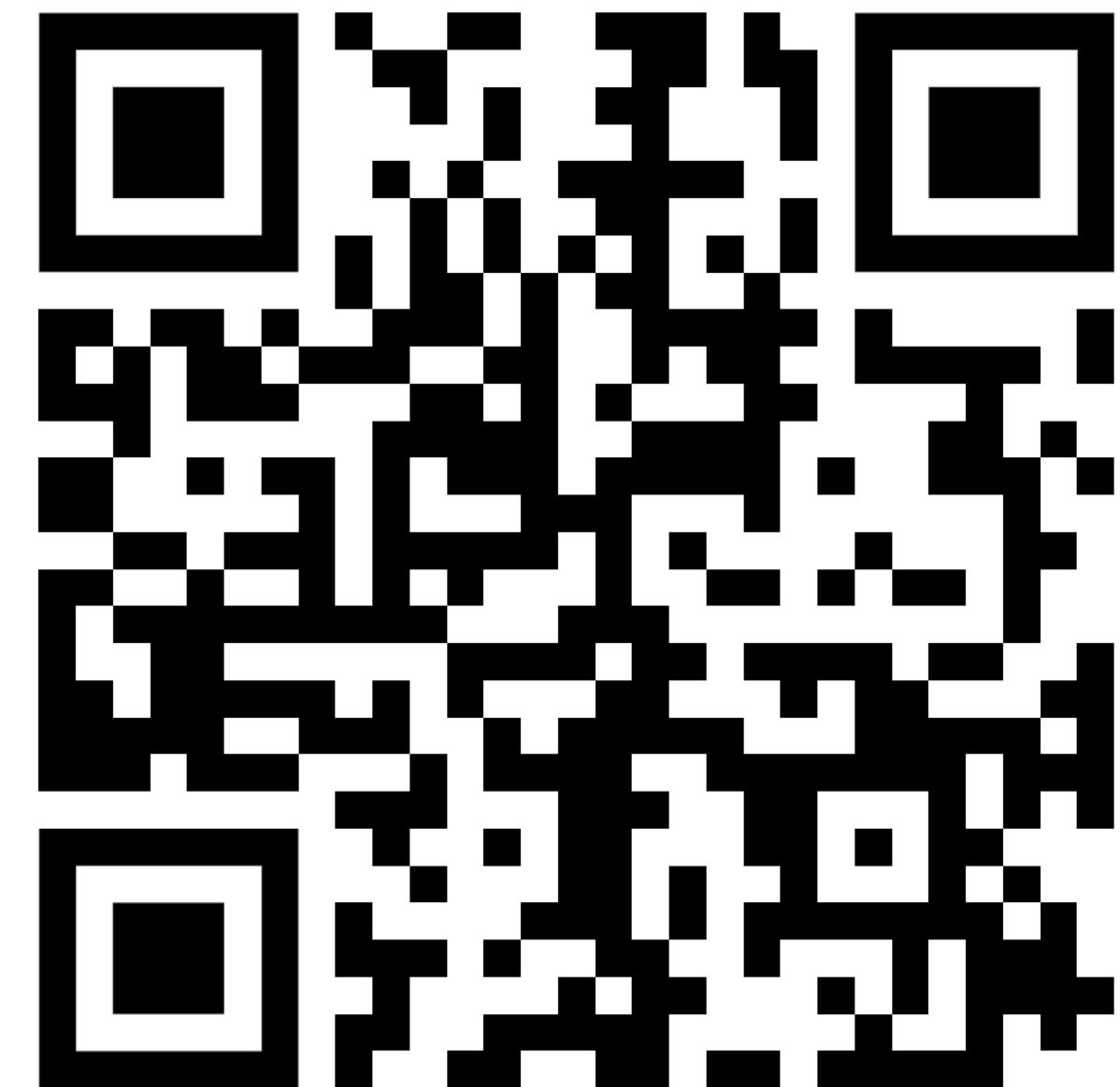
} Симуляторы схем

<https://copy.sh/v86>
<https://github.com/noopkat/avrgirl-arduino>
<https://github.com/thingsSDK/flasher.js>

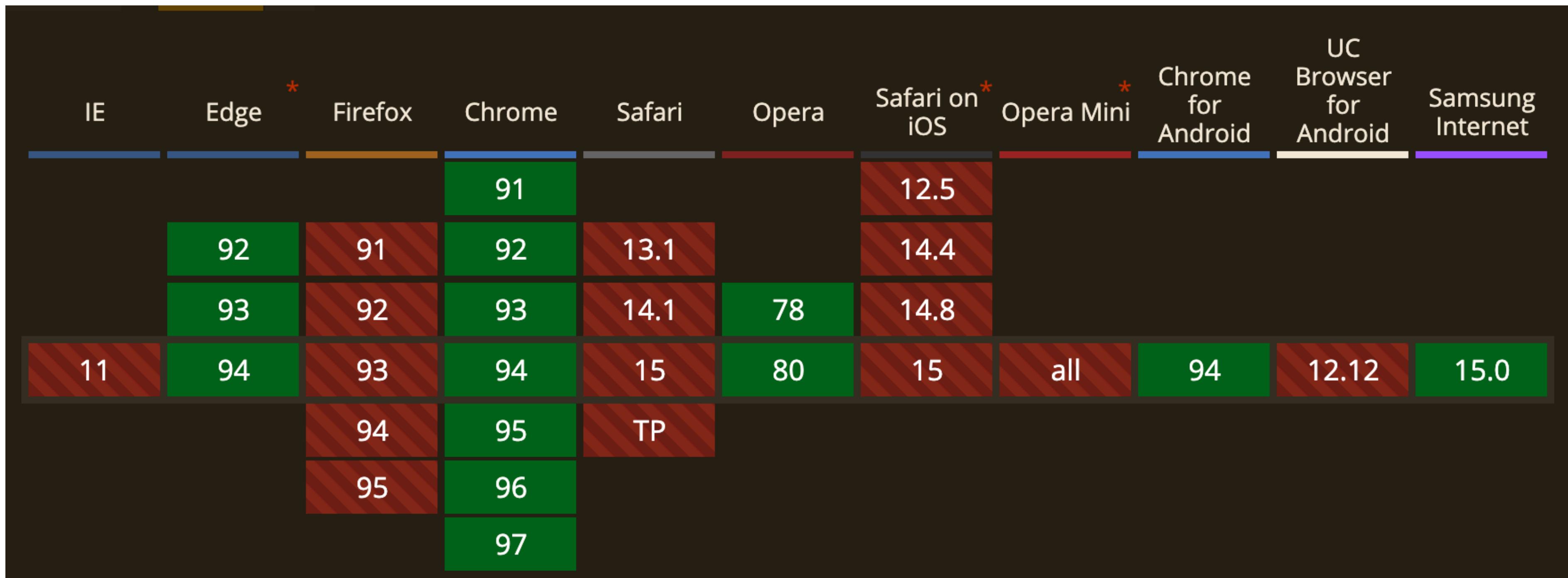
} Разное и интересное



Заключение



Frontend
Conf 2021





Frontend
Conf 2021

Write Once Run Anywhere



JavaScript™



Frontend
Conf 2021

Спасибо!