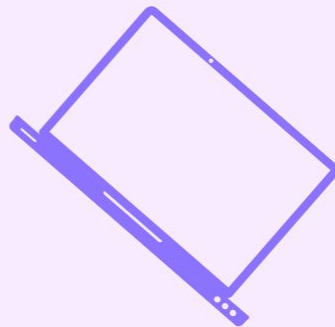


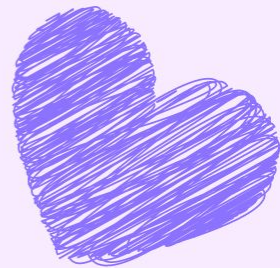


Funky Hardware Beats



The story of `dance-mat.js`

ramonh.dev



EW: Sick late 90's beats



I'm Ramón. (he/him)

From 🇨🇱, living in 🇸🇰

Co-Founder: [BadWebsite.Club](#)

DevRel Strategy Consultant

egghead Instructor

Community member

Mozilla tech speaker alum

Kids' coding coach

Coding live streamer

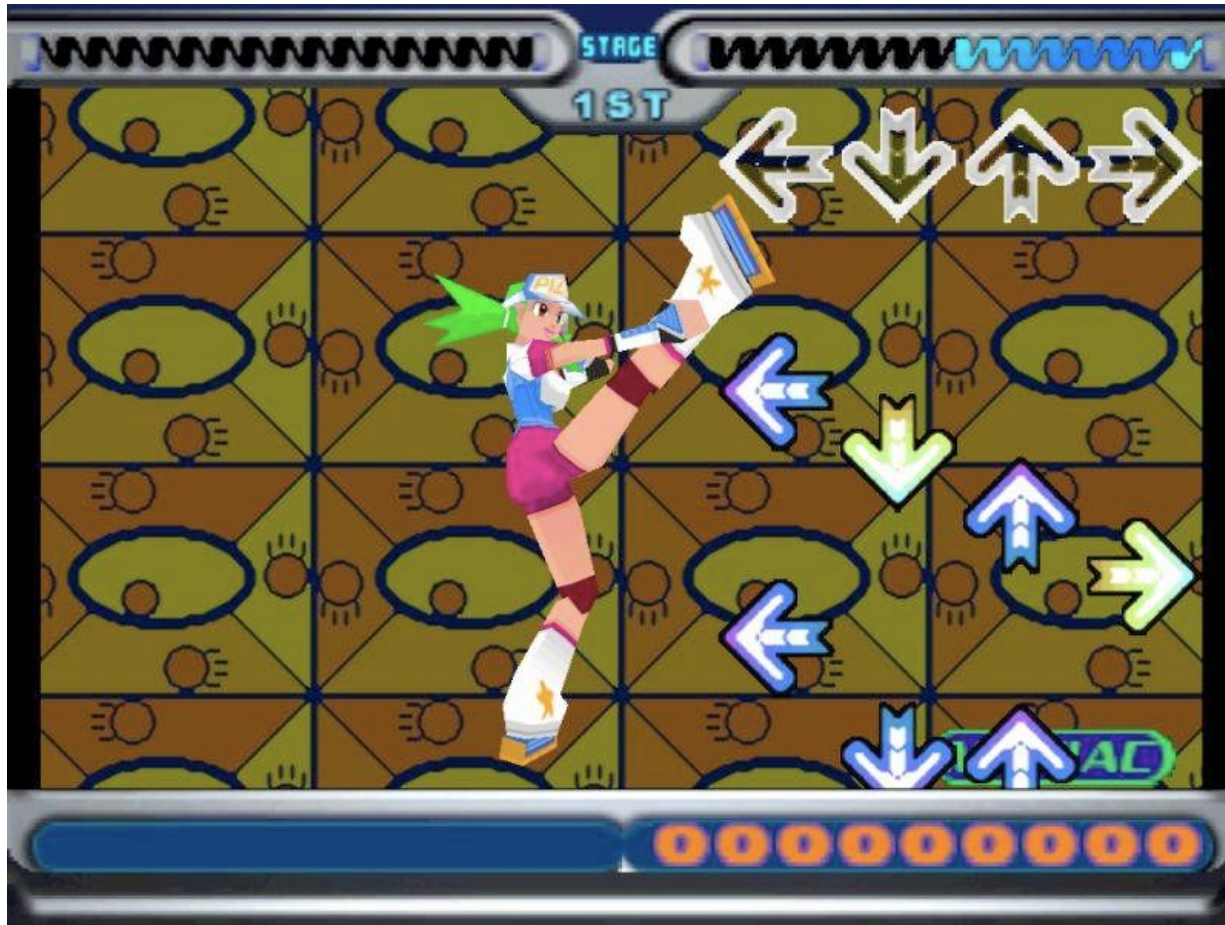




Dance Dance Revolution

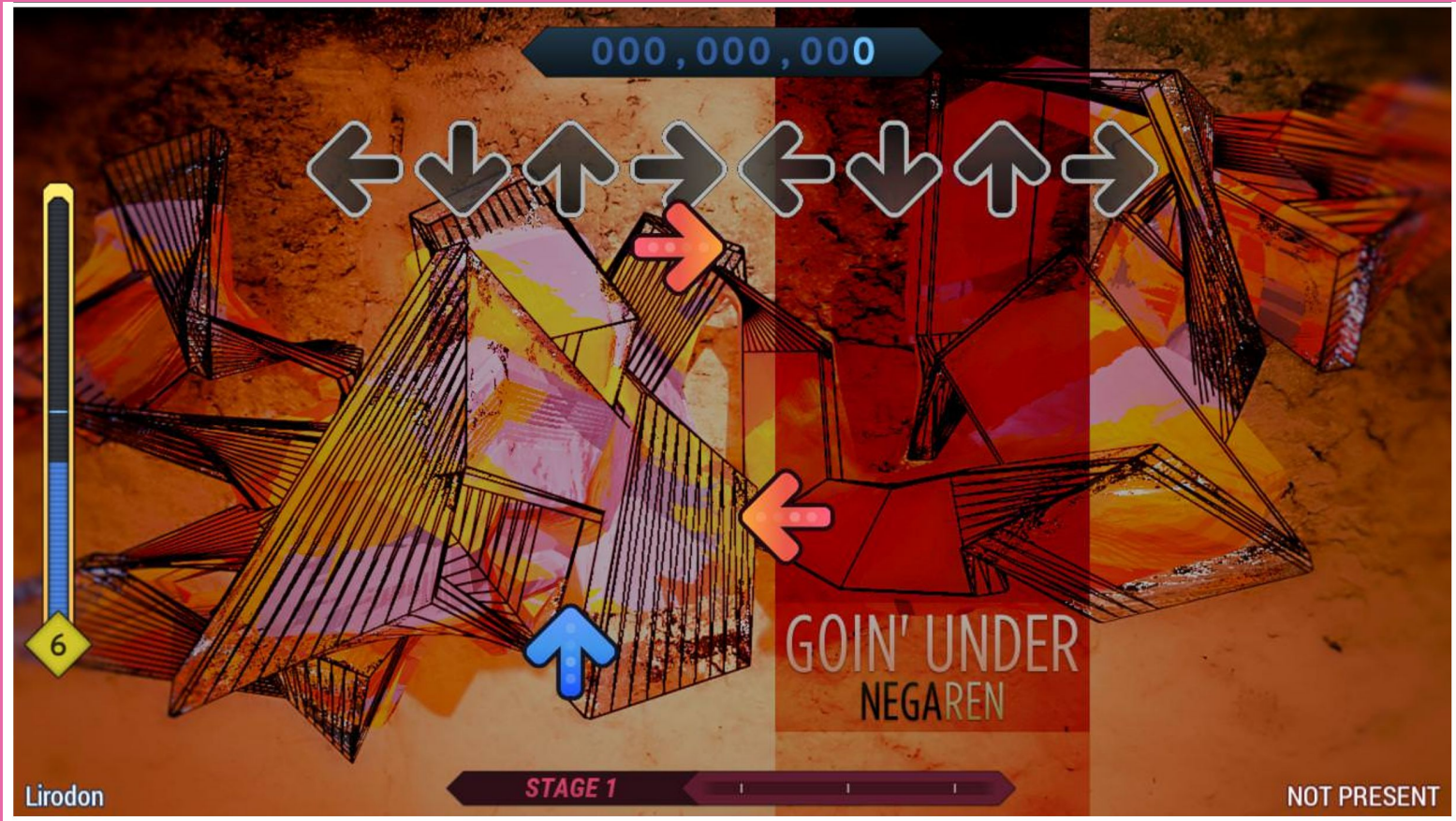
DDR





The Music.

Y'all, the **music.**







(In Stock Now!) 2 x Dance Dance Revolution DDR Metal Dance Pad V 3.0 for Xbox + Dance Dance Revolution DDR Ultramix 4 Dance Game for Xbox



Product Code: M04061-2xM03787

Regular Price: ~~\$919.99~~

Sale Price: \$339.99

Availability: Usually ships the next business day

Out of Stock



Maker Faire Vienna

ÜBER DIE MAKER FAIRE

MITMACHEN

ORT

TICKETS

PROGRAMM 2019

PRESSE



Österreichs größtes DIY-Festival
Maker Faire Vienna
 04. & 05. Mai 2019

[Programm](#)



Online Tickets
Tickets verfügbar
 15% Ermäßigung im Vorverkauf

[Tickets kaufen](#)

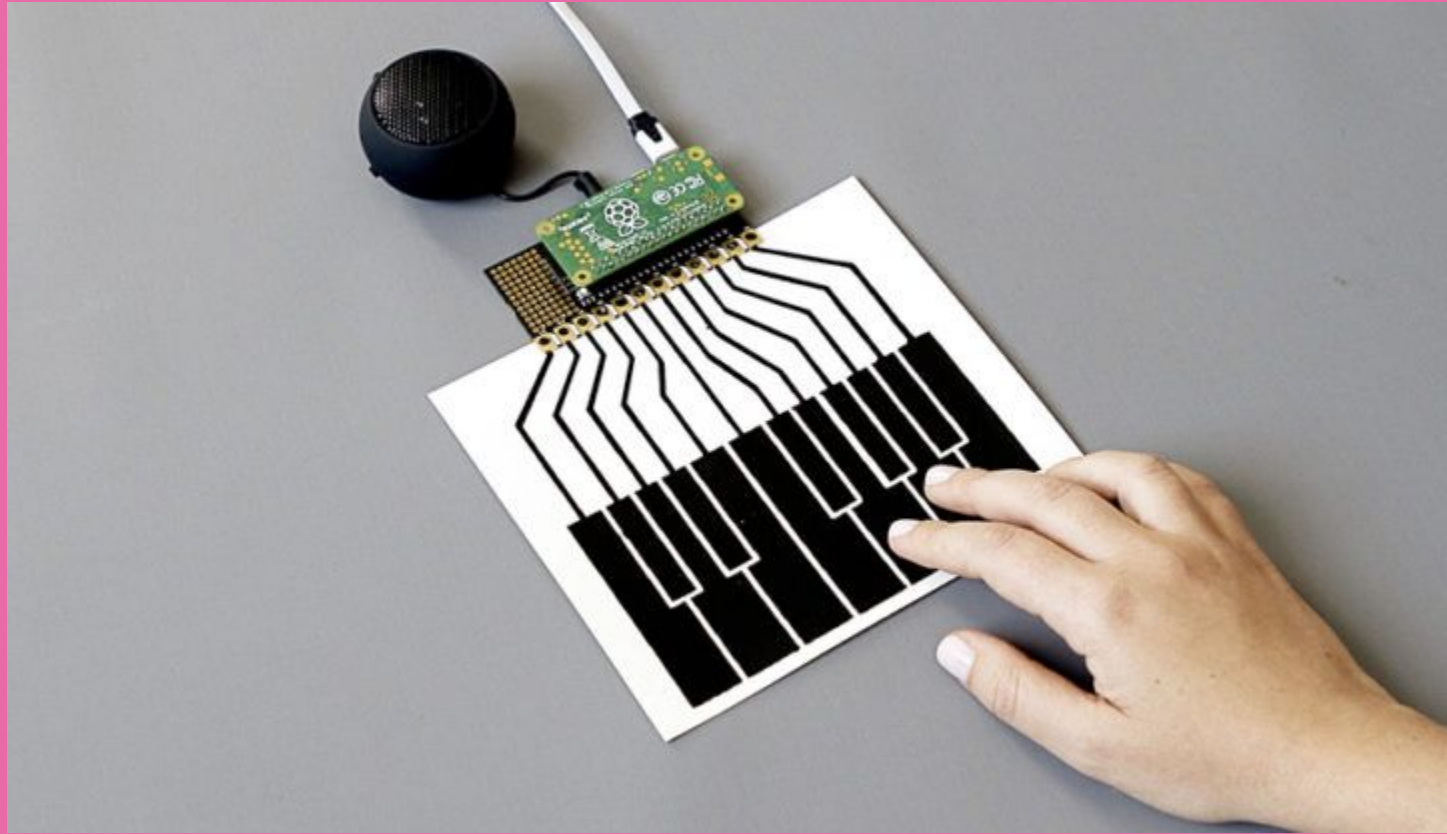
Das Festival für Innovation, Kreativität und Technologie

Ihren Ursprung hat die Maker Faire in den USA. Die Amerikaner sprechen von „The Greatest Show (& Tell) on Earth“ und meinen damit, dass eine Maker Faire zum einen eine Erfindermesse, zum anderen eine Art Jahrmarkt und zeitgleich etwas vollkommen Neues ist. Es ist ein familienfreundliches Festival für Innovation, Kreativität und Technologie.

Hier kommen Maker zusammen, um ihre Projekte einer breiten Öffentlichkeit zu präsentieren. Maker sind experimentierfreudige SelbsterfinderInnen mit Spaß an der Sache, Kreativköpfe, QuerdenkerInnen, TechnikenthusiastInnen und in allen Altersgruppen zu finden. Sie sind wissbegierig, aber auch WissensvermittlerInnen und teilen gerne ihre Erfindungen. Für manchen Aussteller ist die Präsenz auf der Maker Faire auch der Anfang eines erfolgreichen Start-Ups.

„Anfassen und Ausprobieren“ wird großgeschrieben. Auf jeder Maker Faire gibt es viele interessante Mitmachstationen, ergänzt um spannende Vorträge und Workshops. Kinder und Schüler werden auf einer kreativen und spielerischen Weise für Wissenschaft, Technik und dem lustvollen Umgang mit Materialien und Werkzeugen begeistert. Spaß haben steht im Vordergrund. Die Schwerpunkte liegen dabei auf den folgenden Bereichen:







This library requires Node.js v6.7.0 or higher and also requires that the [Bare Conductive MPR121 Wiring Pi Library](#) be installed.

If you're using a Raspberry Pi, this is most easily achieved by running

```
sudo apt-get install picap
```

which will install this module along with lots of example code and setup utilities that will help you get the most out of your Pi Cap.

If you're a masochist, start with

```
npm install node-picap
```

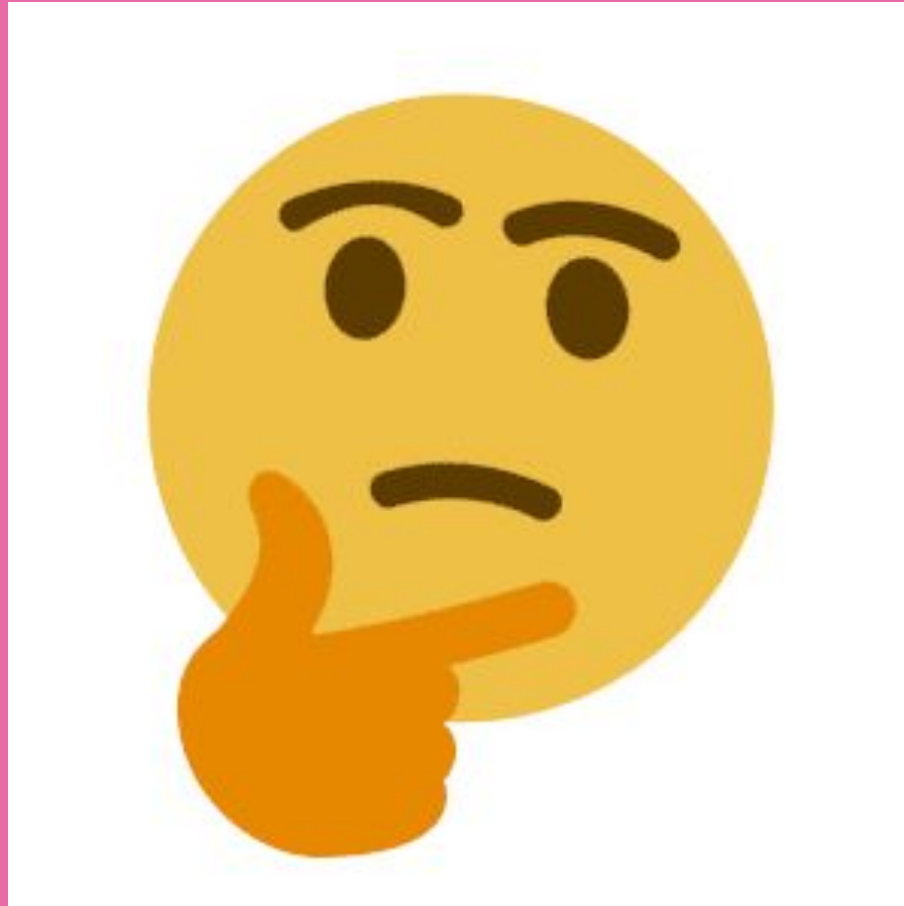
Usage

Simple Touch example

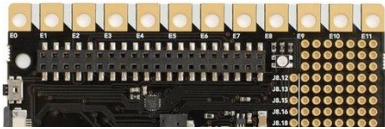
```
var MPR121 = require('node-picap');
var mpr121;

// correct address for the Pi Cap - other boards may vary
mpr121 = new MPR121('0x5C');

mpr121.on('data', function(data) {
  data.forEach(function(electrode, i) {
    if (electrode.isNewTouch) {
      console.log('electrode ' + i + ' was just touched');
    }
    else if (electrode.isNewRelease) {
      console.log('electrode ' + i + ' was just released');
    }
  });
});
```



Joy1_B9	-----	enter	Start
-----	-----	Key /	Select
Joy1_B10	-----	escape	Back
-----	-----	F1	Insert Coin
-----	-----	scroll lock	Operator
-----	-----	-----	EffectUp
-----	-----	-----	EffectDown
Joy1_B1	Key Q	left	Left MenuLeft
Joy1_B4	Key P	right	Right MenuRight
Joy1_B3	Key L	up	Up MenuUp
Joy1_B2	Key S	down	Down MenuDown
Joy1_B7	-----	-----	UpLeft
Joy1_B8	-----	-----	UpRight





Raspberry Pi Zero W

The Raspberry Pi Zero W extends the Pi Zero family and comes with added wireless LAN and Bluetooth connectivity.



Search or jump to...

Pull requests Issues Marketplace Explore



ramonh / picap-dance-mat

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

No description, website, or topics provided.

Edit

Manage topics

18 commits 1 branch 0 releases 1 contributor MIT

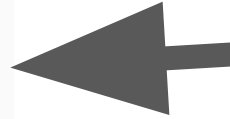
Branch: master New pull request Create new file Upload files Find File Clone or download

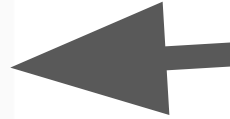
ramonh	Fix keys	Latest commit 8219289 on Nov 10, 2018
.gitignore	Initial commit	2 years ago
LICENSE	Initial commit	2 years ago
README.md	Initial commit	2 years ago
dance-mat.js	Fix keys	5 months ago
package.json	Add linux-device package	2 years ago

README.md

picap-dance-mat



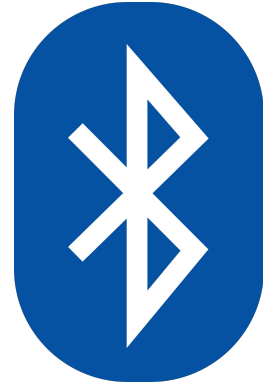


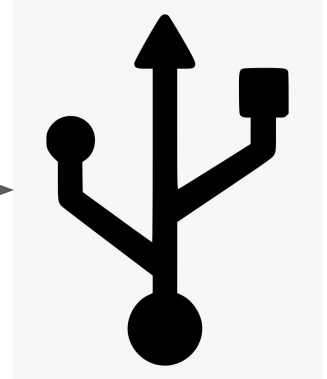


```
const MPR121 = require('node-picap');
const mpr121 = new MPR121('0x5C');

mpr121.setTouchThreshold(40);
mpr121.setReleaseThreshold(20);

// Process touches
mpr121.on('data', (data) => {
  try {
    // SEND DATA TO PC
  });
  catch(e) {
    console.log("ERROR: ", e);
  }
});
```













https://www.kernel.org/doc/Documentation/usb/gadget_configs.txt

Linux USB gadget configured through configs

<https://randomnerdtutorials.com/raspberry-pi-zero-usb-keyboard-hid/>

1. Enable Raspberry Pi OTG (USB on the go)
2. Add boot script to enable Keyboard HID device on the configs



- ~~1. Enable Raspberry Pi OTG (USB on the go)~~
2. Add boot script to enable Keyboard HID device on the configs

configs is a virtual filesystem mounted on the Raspberry Pi.

Raspberry Pi OS comes full of configuration options out of the box!



About USB-IF >

Human Interfa

HID Related Specifica Device Class Definition HID

The [Device Class Definition for HID 1.11](#) defines the requirements for HID compatible devices. It also specifies ho

- be as compact as possible to save c
- allow the software application to s
- be extensible and robust
- support nesting and collections
- be self-describing to allow generic

HID Usage Tables

The [HID Usage Tables 1.12](#) document c

Usages are also used to define the mea

Usages identify the purpose of a collec purpose with its own usage item. Usag

The HID Usage Tables document conta table, are replicated in the Usage Table



ucts



tion necessary to build USB-
e HID class definition are to:

data field in a HID report.

nformation to collections.

ection item can be assigned a

All usages pages, except the Keyboard

Asking for help is A-OK!

Each keystroke is a byte array of 8 hexadecimal keys.

```
33 * KEY_ERR_OVF in all slots to indicate this condition.
34 */
35
36 #define KEY_NONE 0x00 // No key pressed
37 #define KEY_ERR_OVF 0x01 // Keyboard Error Roll Over – used for all slots if too many keys are pressed ("Phantom key")
38 // 0x02 // Keyboard POST Fail
39 // 0x03 // Keyboard Error Undefined
40 #define KEY_A 0x04 // Keyboard a and A
41 #define KEY_B 0x05 // Keyboard b and B
42 #define KEY_C 0x06 // Keyboard c and C
43 #define KEY_D 0x07 // Keyboard d and D
44 #define KEY_E 0x08 // Keyboard e and E
45 #define KEY_F 0x09 // Keyboard f and F
46 #define KEY_G 0x0a // Keyboard g and G
47 #define KEY_H 0x0b // Keyboard h and H
48 #define KEY_I 0x0c // Keyboard i and I
49 #define KEY_J 0x0d // Keyboard j and J
50 #define KEY_K 0x0e // Keyboard k and K
51 #define KEY_L 0x0f // Keyboard l and L
52 #define KEY_M 0x10 // Keyboard m and M
53 #define KEY_N 0x11 // Keyboard n and N
54 #define KEY_O 0x12 // Keyboard o and O
55 #define KEY_P 0x13 // Keyboard p and P
56 #define KEY_Q 0x14 // Keyboard q and Q
57 #define KEY_R 0x15 // Keyboard r and R
58 #define KEY_S 0x16 // Keyboard s and S
59 #define KEY_T 0x17 // Keyboard t and T
60 #define KEY_U 0x18 // Keyboard u and U
```

```
const p1Left = 0x04; // A
const p1Right = 0x05; // B
const p1Up = 0x06; // C
const p1Down = 0x07; // D
```



```
const p1Left = 0x04; // A
const p1Right = 0x05; // B
const p1Up = 0x06; // C
const p1Down = 0x07; // D
```



```
// Process touches  
mpr121.on('data', (data) => {  
  let keys = parsePressedKeys(data);  
});
```

```
parsePressedKeys = (data) => {  
  var pressedKeys = [];  
  data.forEach((electrode, i) => {  
    if (electrode.isTouched) {  
      switch(i) {  
        case 0:  
          pressedKeys.push(p1Left);  
          break;  
        case 1:  
          pressedKeys.push(p1Right);  
          break;  
        case 2:  
          pressedKeys.push(p1Up);  
          break;  
        case 3:  
          pressedKeys.push(p1Down);  
          break;  
      }  
    }  
  });  
  return pressedKeys;  
}
```



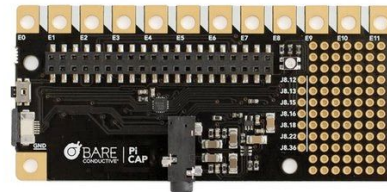
```
// Process touches
mpr121.on('data', (data) => {
  let keys = parsePressedKeys(data);
  let keystroke = keystrokeFromPressedKeys(keys);
});
```

```
keystrokeFromPressedKeys = (pressedKeys) => {  
  var keystroke = [0x00, 0x00];  
  pressedKeys.forEach((key) {  
    keystroke.push(key);  
  });  
  while(keystroke.length < 8) {  
    keystroke.push(0x00);  
  }  
  return keystroke.slice(0, 8);  
}
```



```
// Process touches
mpr121.on('data', (data) => {
  let keys = parsePressedKeys(data);
  let keystroke = keystrokeFromPressedKeys(keys);

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



**And now, the byte
array...**

Uint8Array

 Languages Edit

Jump to: [Syntax](#) [Properties](#) [Methods](#) [Uint8Array prototype](#) [Examples](#) [Specifications](#) [Browser compatibility](#) [Compatibility notes](#) [See also](#)

[Web technology for developers](#) > [JavaScript](#) >

[JavaScript reference](#) >

[Standard built-in objects](#) > [Uint8Array](#)

Related Topics

[Standard built-in objects](#)

[TypedArray](#)

Properties

`TypedArray.BYTES_PER_ELEMENT`

`TypedArray.name`

`TypedArray.prototype`

`TypedArray.prototype.buffer`

`TypedArray.prototype.byteLength`

The **Uint8Array** typed array represents an array of 8-bit unsigned integers. The contents are initialized to 0. Once established, you can reference elements in the array using the object's methods, or using standard array index syntax (that is, using bracket notation).

Syntax

```
new Uint8Array(); // new in ES2017
new Uint8Array(length);
new Uint8Array(typedArray);
new Uint8Array(object);
new Uint8Array(buffer [, byteOffset [, length]]);
```

For more information about the constructor syntax and the parameters, see [TypedArray](#).

```
// Process touches
mpr121.on('data', (data) => {
  let keys = parsePressedKeys(data);
  let keystroke = keystrokeFromPressedKeys(keys);

  let buffer = Uint8Array.from(keystroke);

});
```

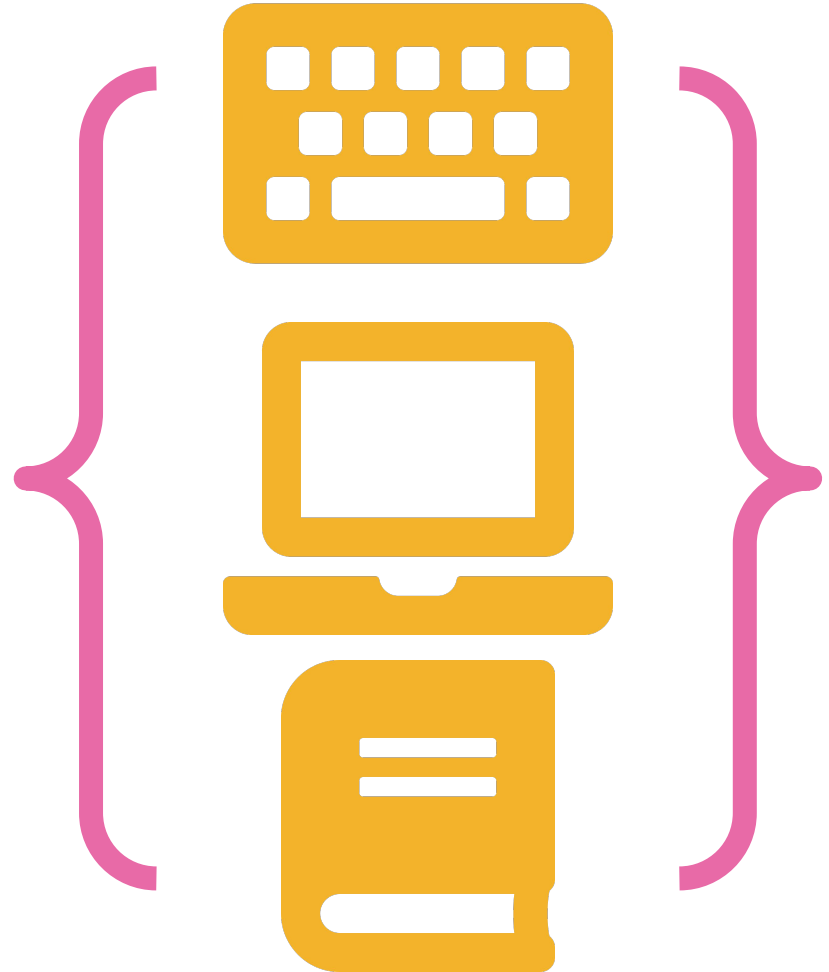
**And now, we send the
keystroke...**

**And now, we send the
keystroke...?**

🌟🌟 File Descriptors 🌟🌟



=





' /dev/hidg0 '

```
pi@raspberrypi:~ $ echo "blablabla I am a keystroke" | sudo tee -a /dev/hidg0
```



Search packages

Search

Join

Log In

Ready to take your JavaScript development to the next level? Meet npm Enterprise - the ultimate in enterprise JavaScript. [Learn more »](#)

linux-device

2.0.15 • Public • Published 5 months ago

Readme

4 Dependencies

1 Dependents

36 Versions

linux-device

Native addon to communicate with linux devices (can also be used for sockets or FIFOs).

Installation

Install with npm :

```
$ npm install linux-device
```

Usage

See the [API Docs](#) for more information.

Remote usage

install

```
> npm i linux-device
```

weekly downloads

209



version

2.0.15

license

ISC

open issues

1

pull requests

0

homepage

github.com

repository

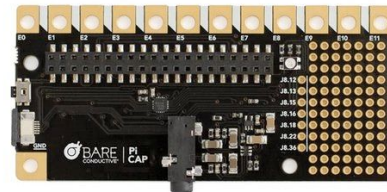
github

```
const DeviceHandle = require('linux-device');  
  
// Open up access to the USB interface  
const device = new DeviceHandle('/dev/hidg0', true, 16);
```

```
// Process touches
mpr121.on('data', (data) => {
  let keys = parsePressedKeys(data);
  keystroke = keystrokeFromPressedKeys(keys);

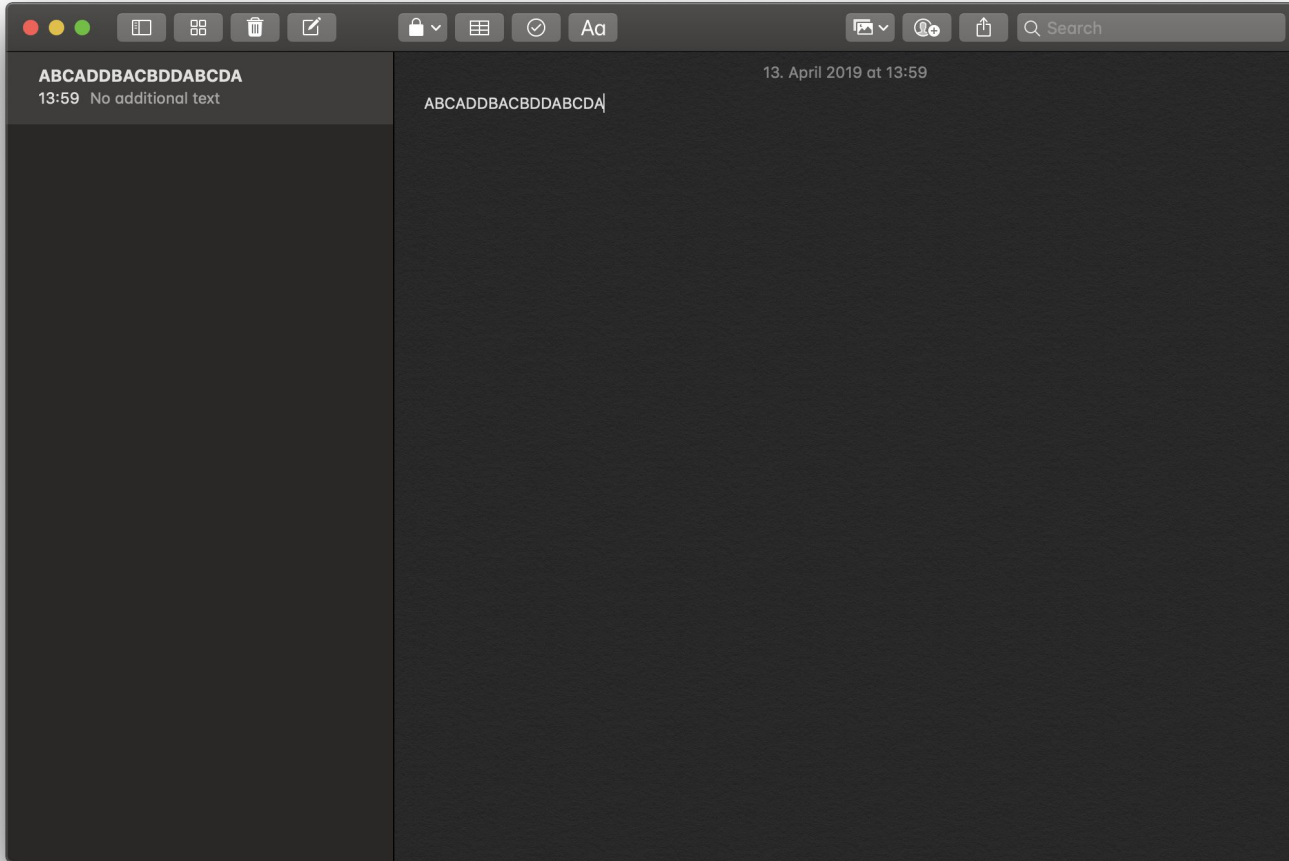
  let buffer = Uint8Array.from(keystroke);

  device.write(buffer);
});
```



```
const exec = require('child_process').exec;
process.on('SIGINT', () => {
  device.close();
  process.exit(0);
});
```


node dance-mat.js












StepMania

Your Results Timing Difficulty: 4 Stage 1
 Life Difficulty: 2

D



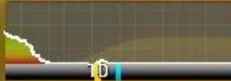
Mecha-Tribe
ASSAULT
Konnisar

ITG DP: 0697 /1211	0014	FLAWLESS
MIGS DP: -104 /0564	0026	PERFECT
	0036	GREAT
	0028	GOOD
	0030	BOO
	0024	MISS
	0013	HELD
	0010	MAX COMBO

000,000,433

ASSAULT

FailOff



NOVICE

2

32.00%

dance

Exit
Move On
◀ ▶ or
Snapshot
NOT PRESENT

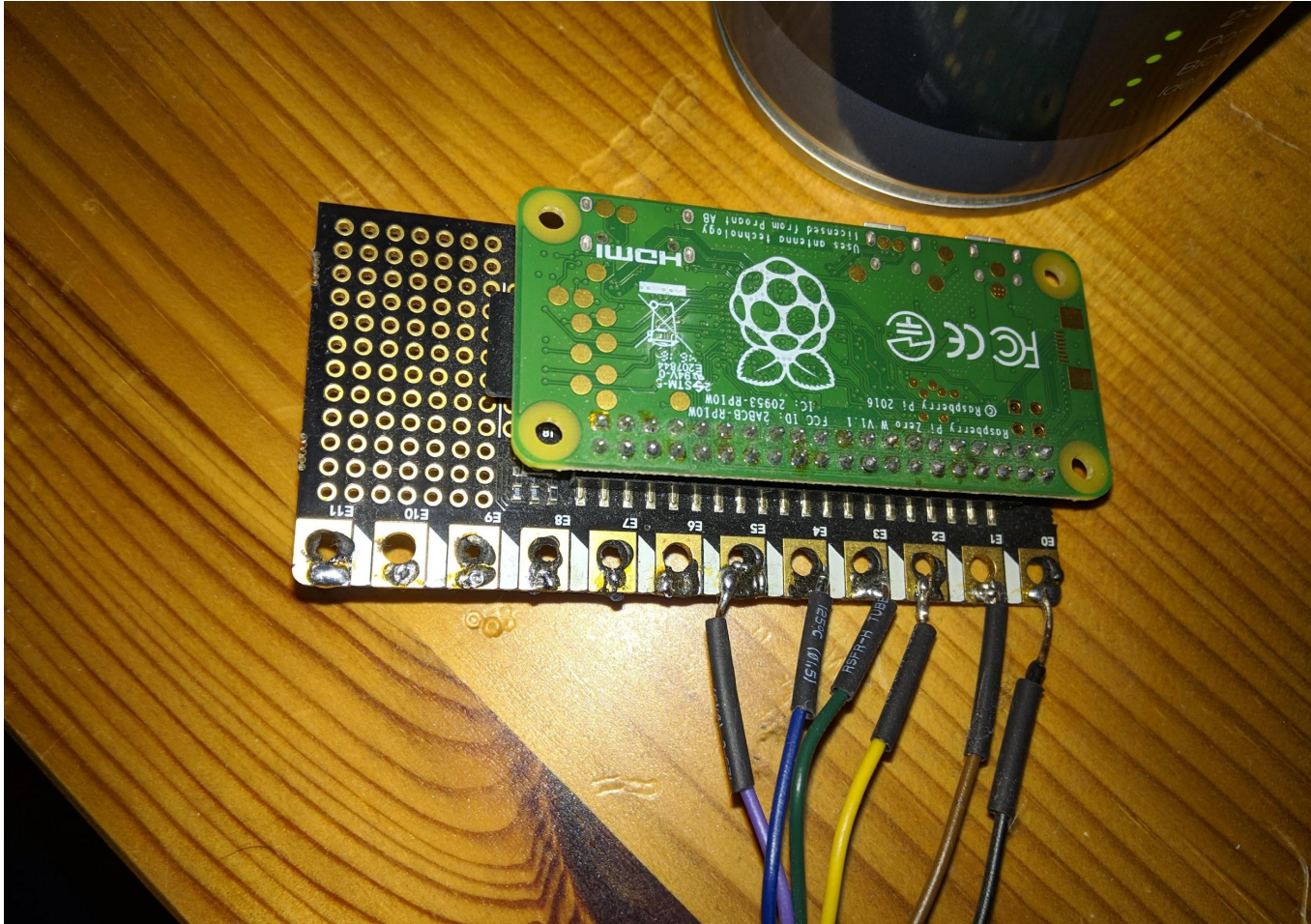
So are we set for life?

So are we set for life?

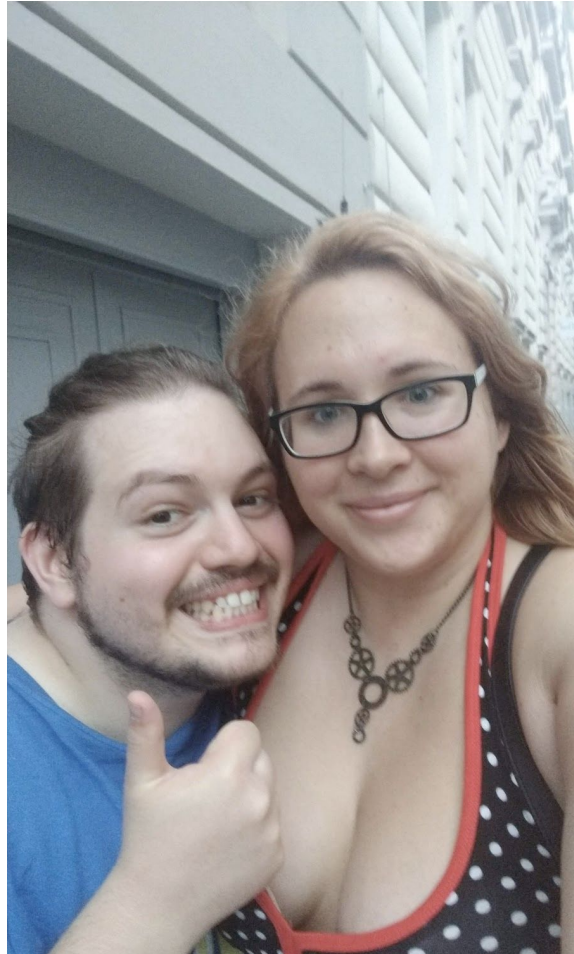


**If we could do
it all over
again...**

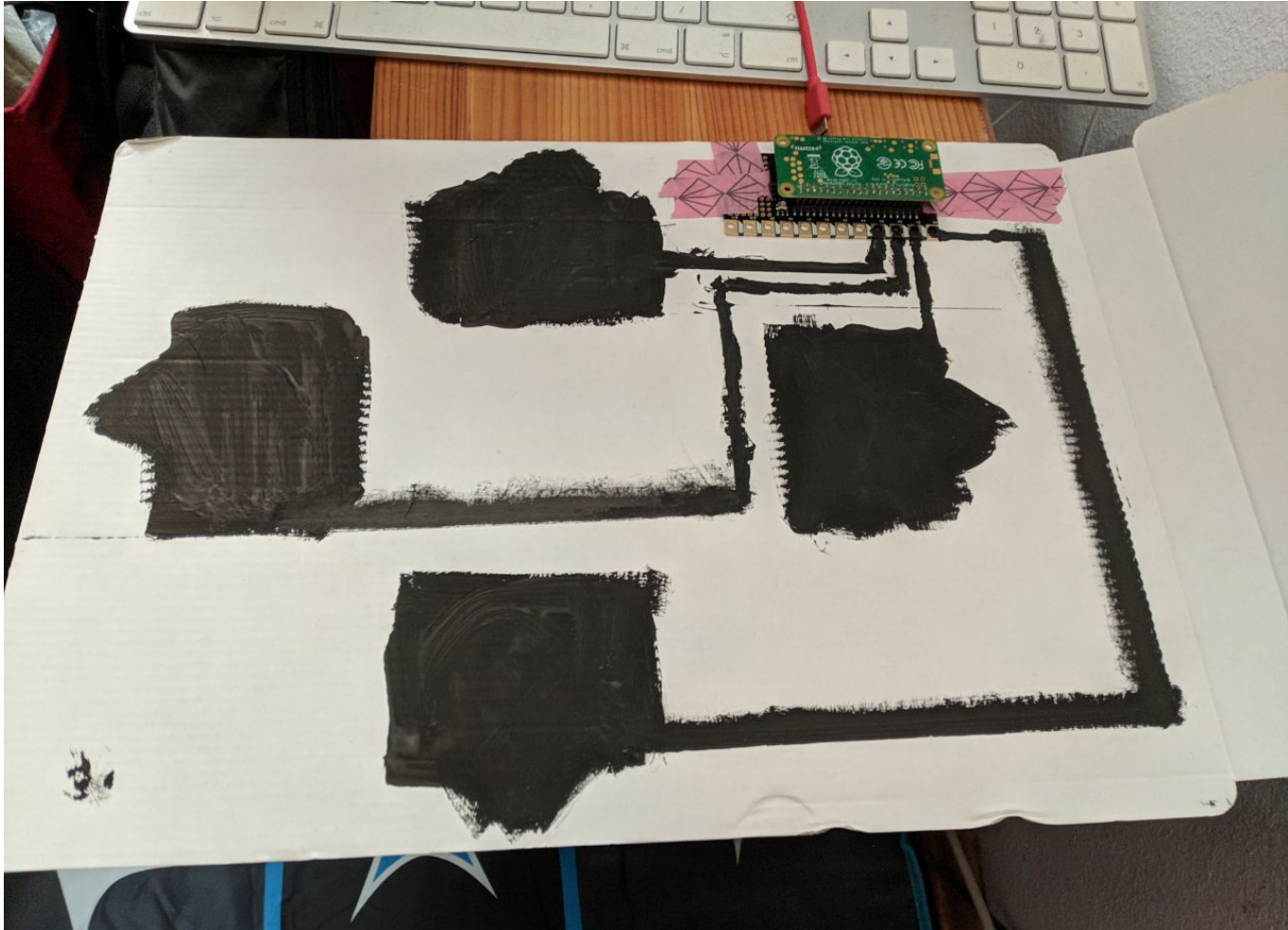
- Use an Arduino
- Smoother surface
- Physical instead of “paint” cabling

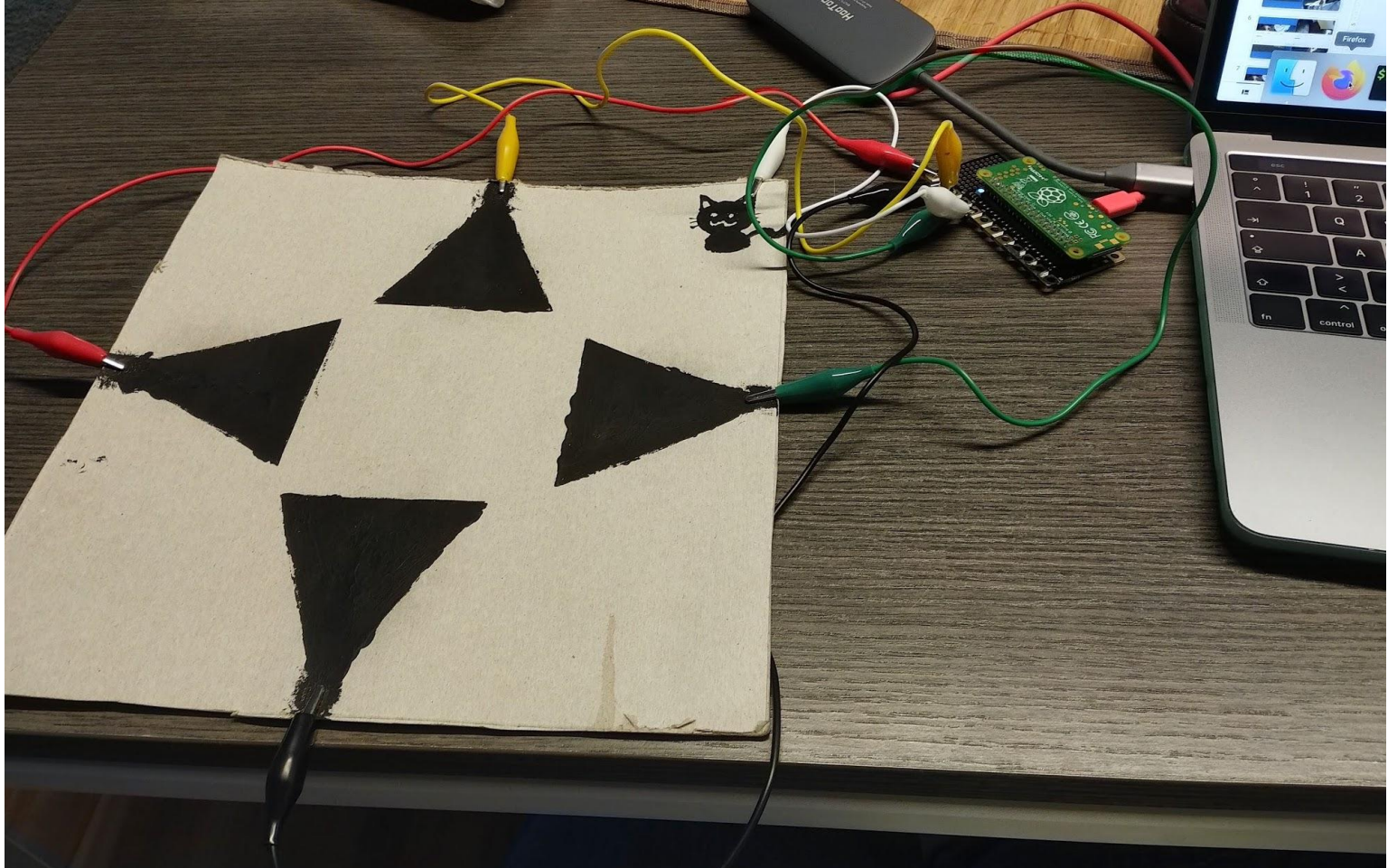


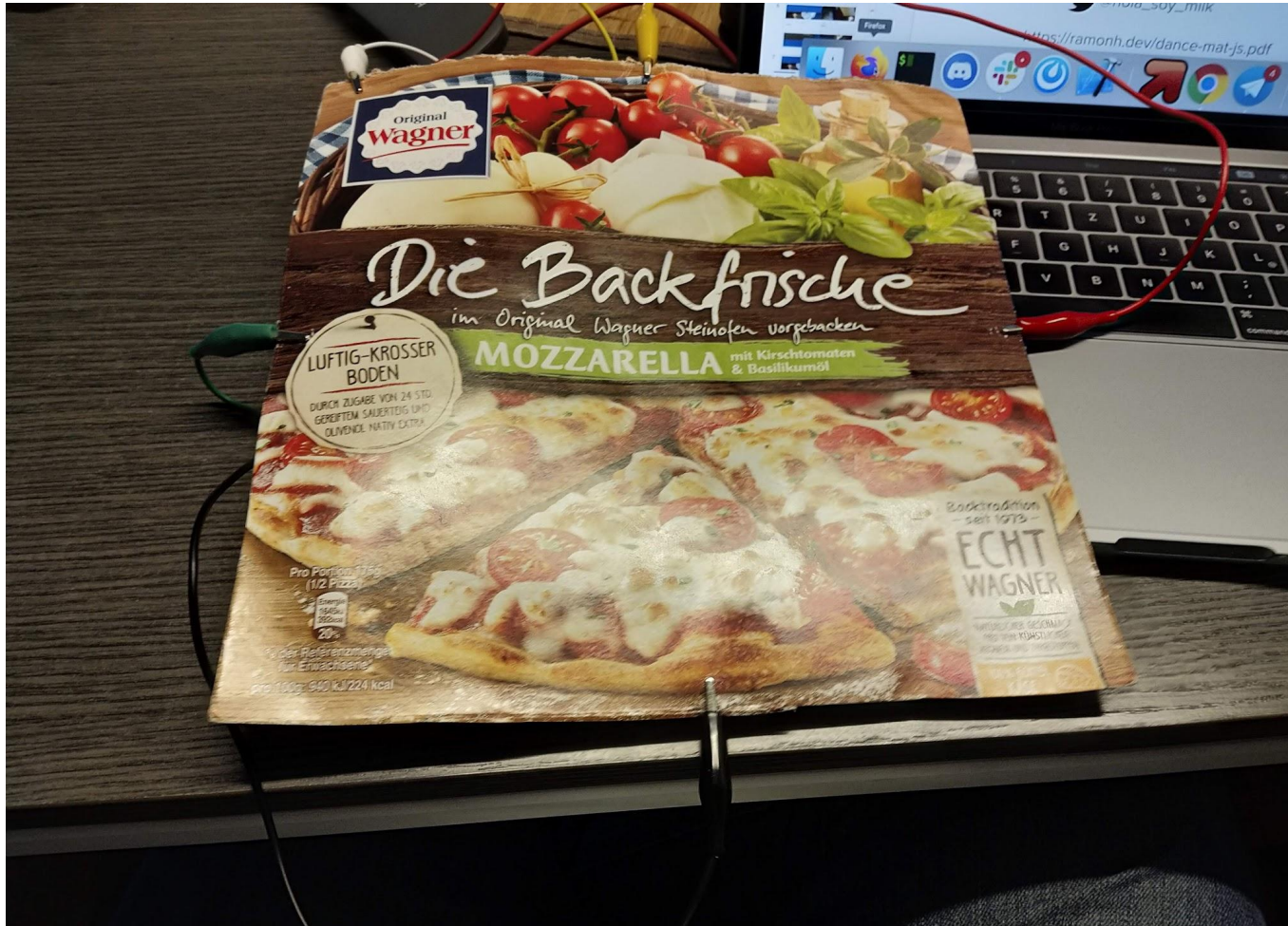
But hey!
**We learned a whole mess
of stuff.**



Dance for the nice people, Ramón.







Takeaways

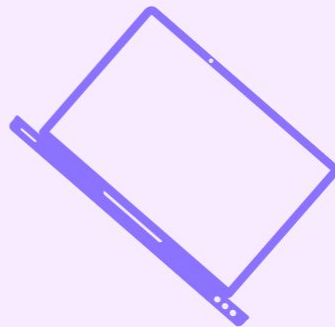
- Hardware hacking looks daunting, but there's a ton of support out there
- Play around with things!
- There's packages to help you
- But most importantly...

Just go for it!

<https://ramonh.dev/dance-mat-js.pdf>



Ramón Huidobro



**Thank you,
Friends!**



ramonh.dev/card

