

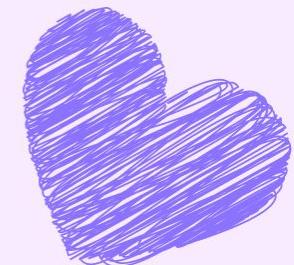
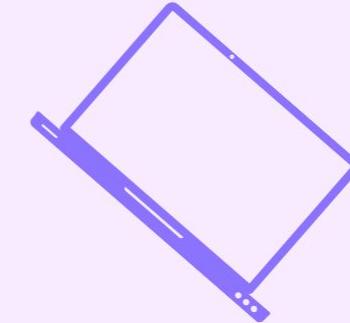


Funky Hardware Beats



The story of dance-mat.js

carmenh.dev



CW: Sick late 90's beats

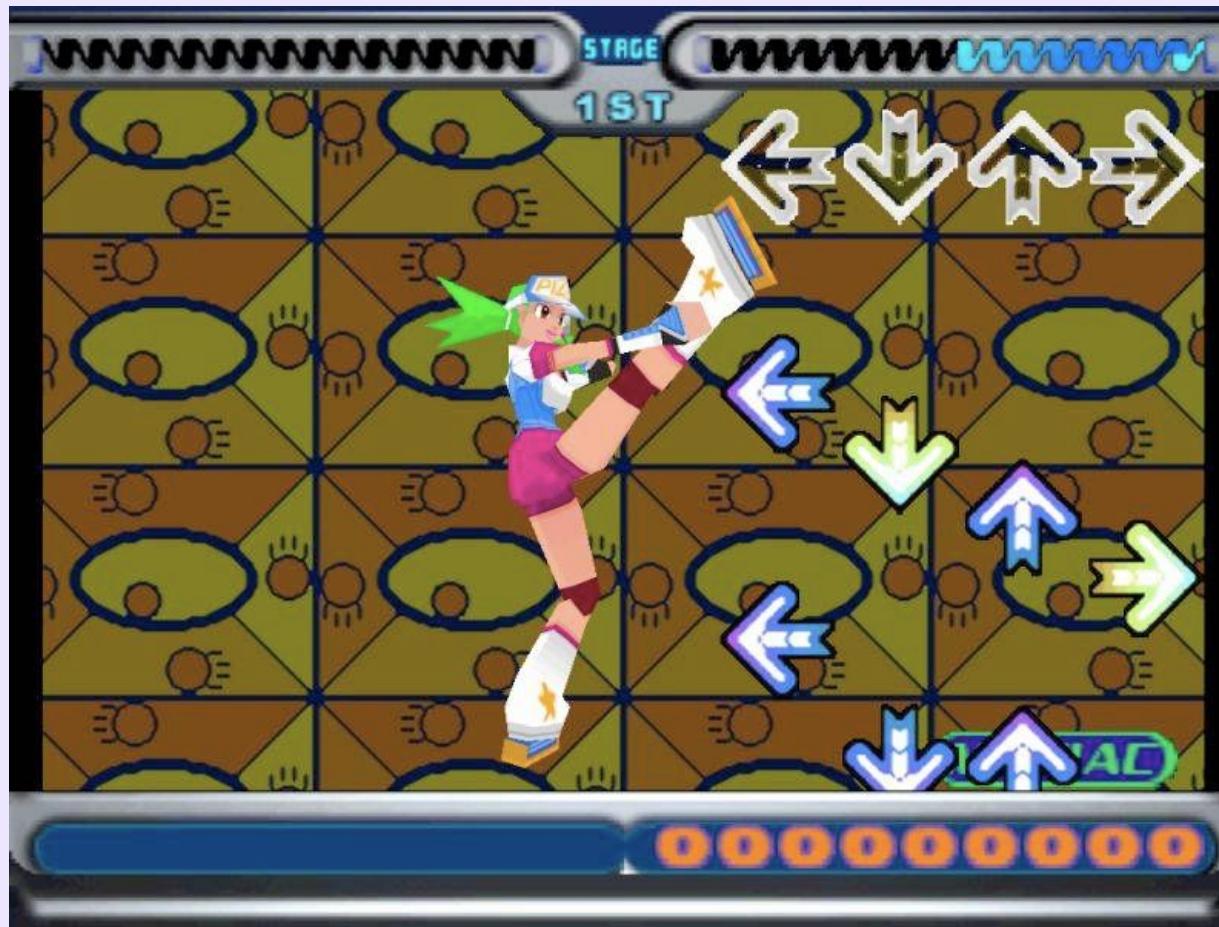




Dance Dance Revolution

DDR



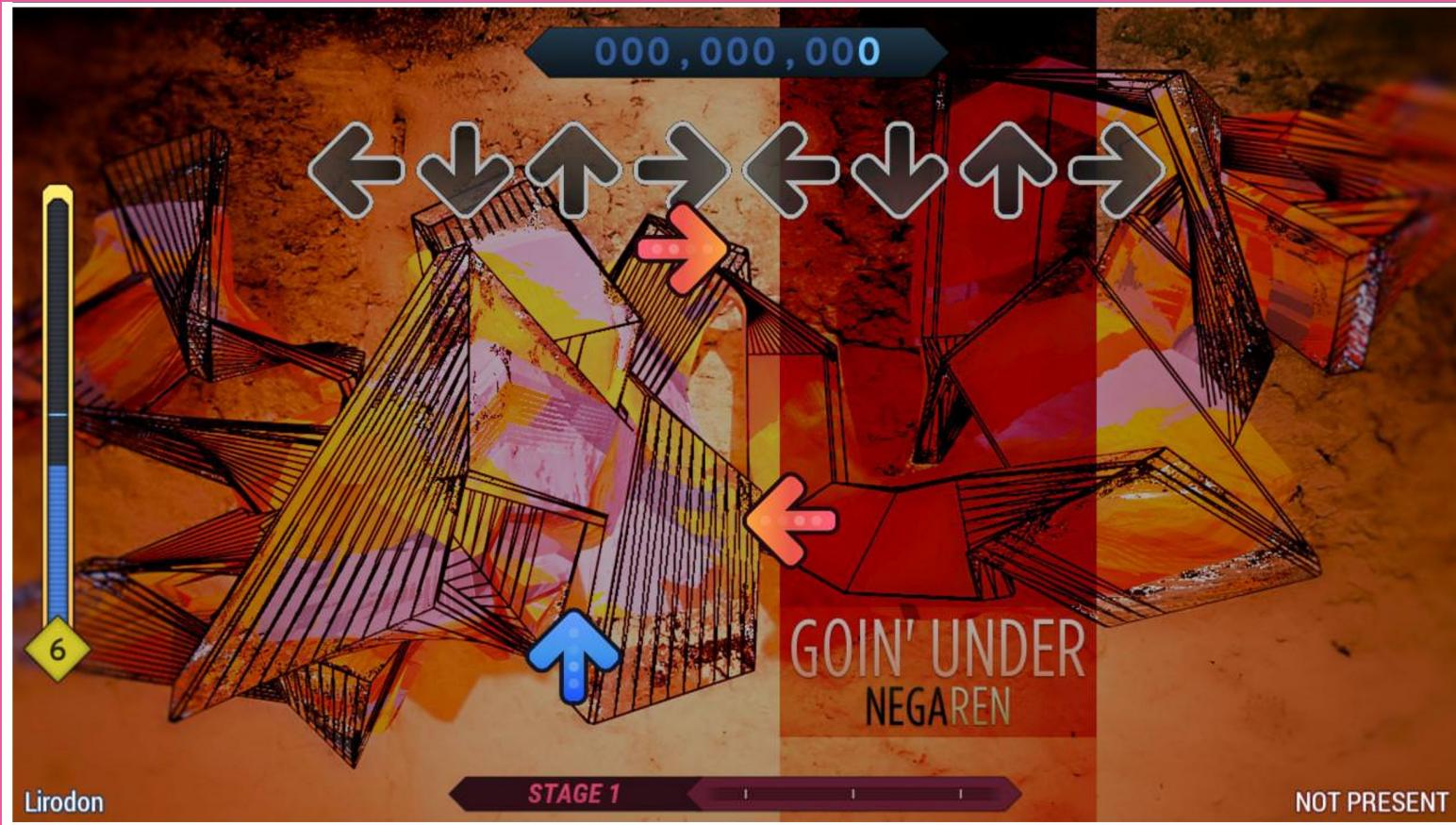




The Music.



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



CLICK TO ENLARGE

Product Code: M04061-2xM03787

Regular Price: \$919.99

Sale Price: \$339.99

Availability: Usually ships the next business day

Out of Stock



**TELL A
FRIEND**



**BOOKMARK THIS
PAGE**

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](#)

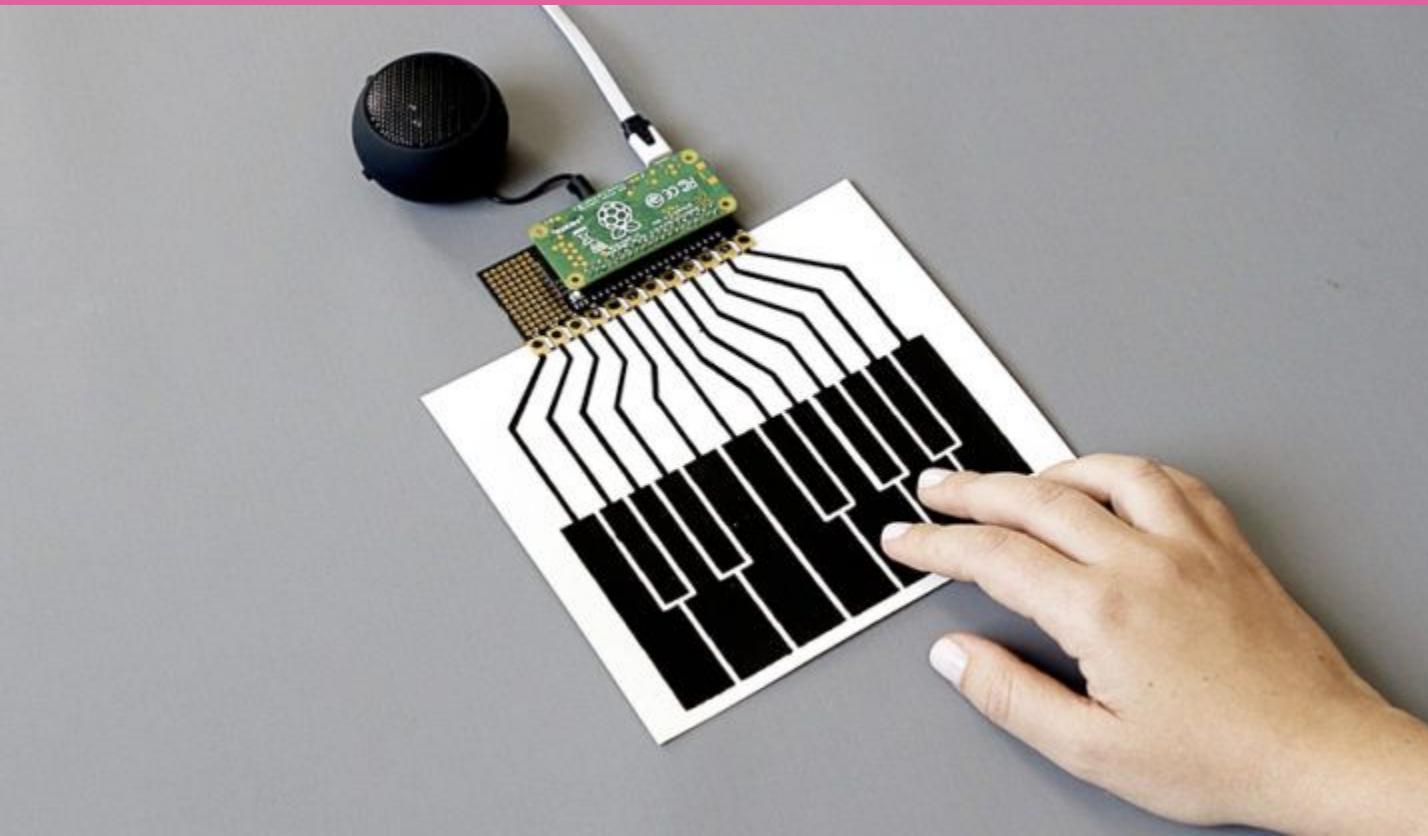
Das Festival für Innovation, Kreativität und Technologie

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 SelbermacherInnen mit Spaß an der Sache, Kreativköpfe, QuerdenkerInnen, TechnikenthusiastInnen und in allen Altergruppen 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
```

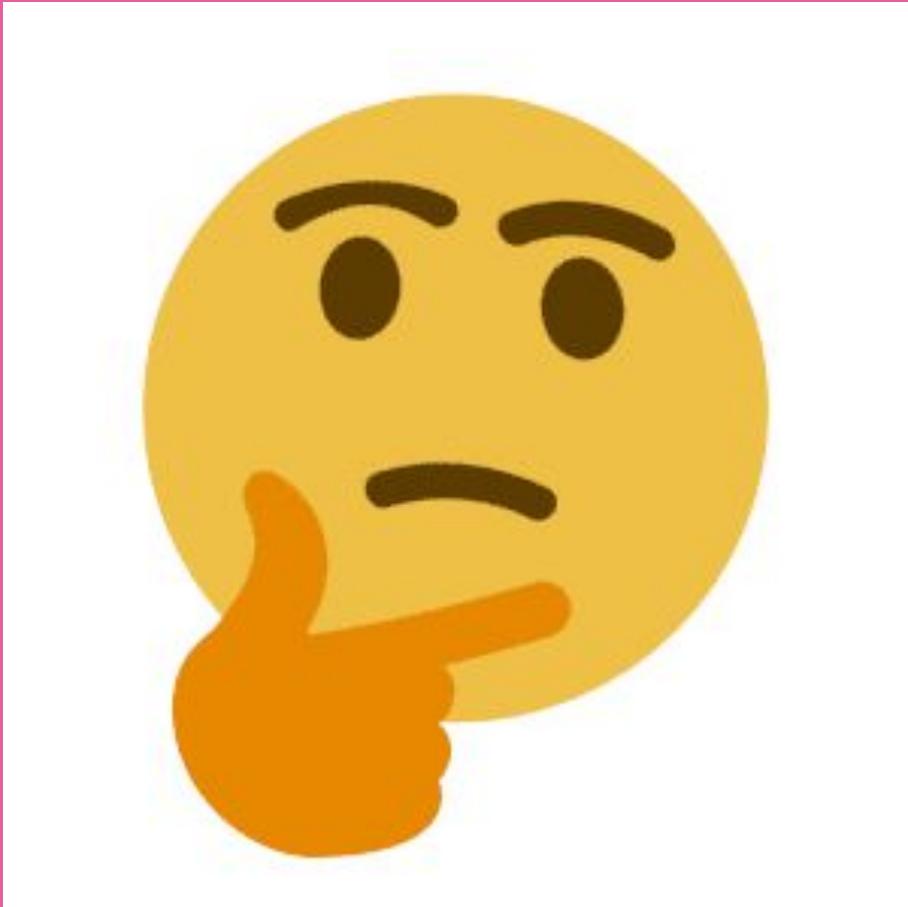
Usage

Simple Touch example

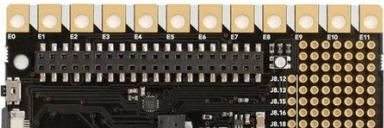
```
var MPR121 = require('node-picap');
var mpr121;

// 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 |
| Joy1_B4 | Key P | right | Right |
| Joy1_B3 | Key L | up | Up |
| Joy1_B2 | Key S | down | Down |
| Joy1_B7 | | | MenuUp |
| Joy1_B8 | | | MenuDown |
| | | | UpLeft |
| | | | UpRight |





Raspberry Pi Zero W

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



hola-soy-milk / picap-dance-mat

Type to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

picap-dance-mat Public

Unpin Unwatch 2 Fork 1 Star 7

master 1 Branch Tags Go to file Add file Code

hola-soy-milk Update README.md f83481d · 4 years ago 23 Commits

.gitignore Initial commit 7 years ago

LICENSE Initial commit 7 years ago

README.md Update README.md 4 years ago

dance-mat.js chore: replace var with let and const respectively 5 years ago

package.json Add linux-device package 7 years ago

About

No description, website, or topics provided.

Readme MIT license Activity 7 stars 2 watching 1 fork

Releases

No releases published [Create a new release](#)

Packages

No packages published [Publish your first package](#)

Contributors 2

hola-soy-milk Carmen Huidobro

meeroslav Miroslav Jonaš

Dance-mat.js: A DDR controller running on Raspberry Pi using the [Bare Conductive Picap](#)

Getting up and running

Once you've [got the hardware up and running](#), you need to clone the project on the Raspberry Pi, and install dependencies:

```
$ npm install
```



I'm Carmen! (she/her)

From , living in 

Co-Founder: BadWebsite.Club

Developer Educator, Directus

DevRel Strategy Consultant

egghead Instructor

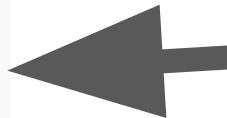
Community member

Mozilla tech speaker alum

Kids' coding coach



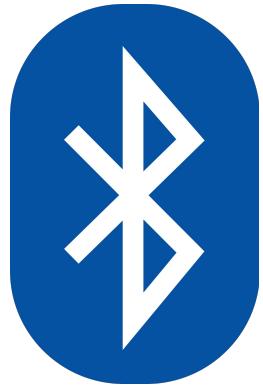
I have no idea what I'm doing

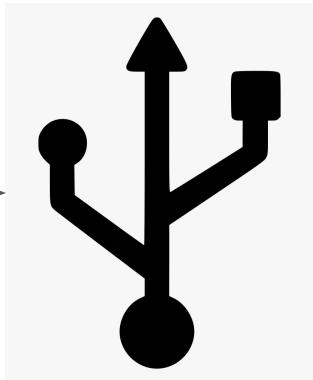


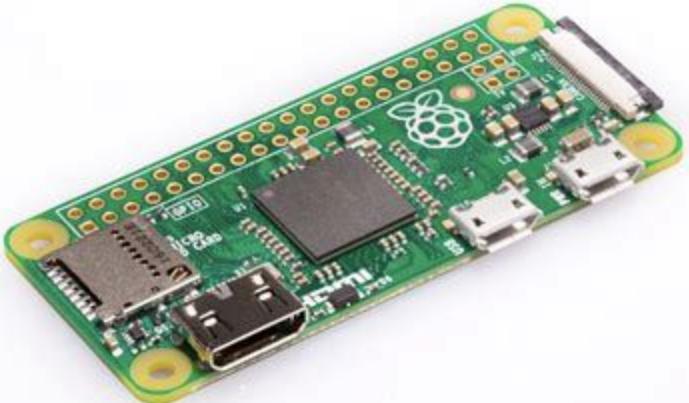
```
const MPR121 = require('node-picap');
const mpr121 = new MPR121('0x5C');

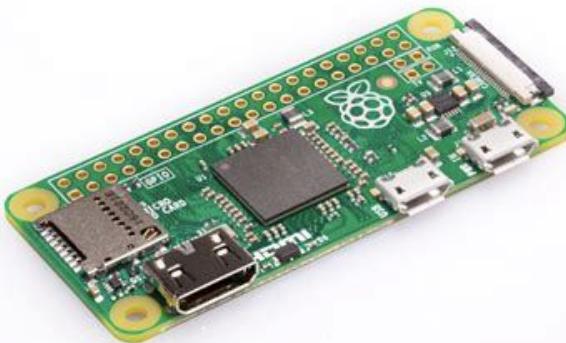
mpr121.setTouchThreshold(40);
mpr121.setReleaseThreshold(20);

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



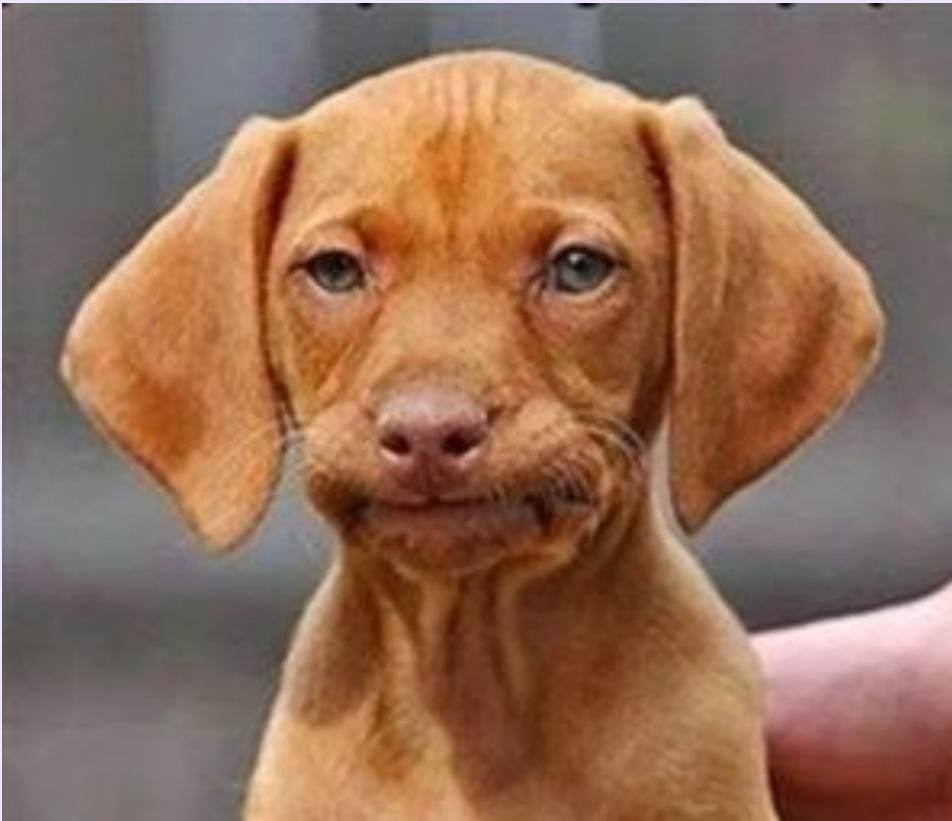






!=







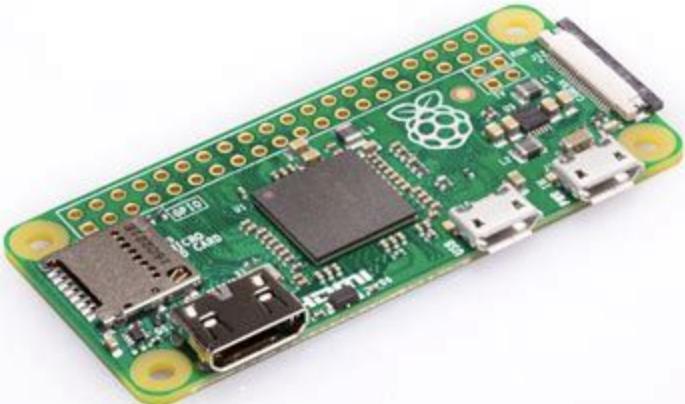
I have no idea what I'm doing

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

Linux USB gadget configured through configfs

<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 configfs**



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

```
#!/bin/bash
cd /sys/kernel/config/usb_gadget/
mkdir -p ddrusb
cd ddrusb
echo 0x1d6b > idVendor # Linux Foundation
echo 0x0104 > idProduct # Multifunction Composite Gadget
echo 0x0100 > bcdDevice # v1.0.0
echo 0x0200 > bcdUSB # USB2
mkdir -p strings/0x409
echo "fedcba9876543210" > strings/0x409/serialnumber
echo "Carmen Huidobro" > strings/0x409/manufacturer
echo "DDR Dance Mat" > strings/0x409/product
mkdir -p configs/c.1/strings/0x409
echo "Config 1: ECM network" > configs/c.1/strings/0x409/configuration
echo 250 > configs/c.1/MaxPower

# Add functions here
mkdir -p functions/hid.usb0
echo 1 > functions/hid.usb0/protocol
echo 1 > functions/hid.usb0/subclass
echo 8 > functions/hid.usb0/report_length
echo -ne
\\x05\\x01\\x09\\x06\\xa1\\x01\\x05\\x07\\x19\\xe0\\x29\\xe7\\x15\\x00\\x25\\x01\\x75\\x01\\x95\\x08\\x81\\x02\\x95\\x08\\x19\\x01\\x29\\x05\\x91\\x02\\x95\\x01\\x75\\x03\\x91\\x03\\x95\\x06\\x75\\x08\\x15\\x00\\x25\\x65\\x05\\x07\\x19\\functions/hid.usb0/report_desc
ln -s functions/hid.usb0 configs/c.1/
# End functions

ls /sys/class/udc > UDC
```



configfs is a virtual filesystem mounted on the Raspberry Pi.

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



About USB-IF >

Developers >

Compliance >

Retail Pressroom >

Document Library Products



Human Interface Devices (HID) Information

HID Related Specifications and Tools

Device Class Definition HID

The [Device Class Definition for HID 1.11](#) is intended to supplement the USB Specification and provide HID manufacturers with the information necessary to build USB-compatible devices. It also specifies how the HID class driver should extract data from USB devices. The primary and underlying goals of the HID class definition are to:

- be as compact as possible to save device data space
- allow the software application to skip unknown information
- be extensible and robust
- support nesting and collections
- be self-describing to allow generic software applications

HID Usage Tables

The [HID Usage Tables 1.12](#) document defines constants that can be interpreted by an application to identify the purpose and meaning of a data field in a HID report.

Usages are also used to define the meaning of groups of related data items. This is accomplished by the hierarchical assignment of usage information to collections.

Usages identify the purpose of a collection and the items it contains. Each Input, Output, Feature, and/or Collection data item within a Collection item can be assigned a purpose with its own usage item. Usages assigned to a collection apply to the items within the collection.

The HID Usage Tables document contains extensions to the tables defined in the USB Device Class Definition for Human Interface Devices. All usages pages, except the Keyboard table, are replicated in the Usage Table document. The Usage Table document identifies the extensions to the Keyboard usage table.



About USB-IF >

Human Interface

HID Related Specifications

Device Class Definition HID

The [Device Class Definition for HID 1.1](#) document defines the requirements for USB-compliant devices. It also specifies how to implement the HID class definition.

- be as compact as possible to save costs
- allow the software application to select the device
- be extensible and robust
- support nesting and collections
- be self-describing to allow generic device drivers

HID Usage Tables

The [HID Usage Tables 1.12](#) document defines the usage items and their meanings.

Usages are also used to define the meaning of a collection item. A collection item can be assigned a usage item.

Usages identify the purpose of a collection item. A collection item can be assigned a usage item with its own usage item. Usages are replicated in the Usage Table.

The HID Usage Tables document contains a table of usages. These usages, along with their descriptions, are replicated in the Usage Table.



cts lock search

ion necessary to build USB-compliant devices. The requirements for the HID class definition are to:

data field in a HID report.

information to collections.

ection item can be assigned a usage item.

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
```

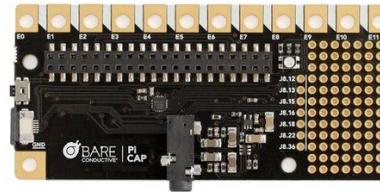
```
const p1Left = 0x04; // A
const p1Right = 0x05; // B
const p1Up = 0x06; // C
const p1Down = 0x07; // D
```

```
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);
}) ;
```

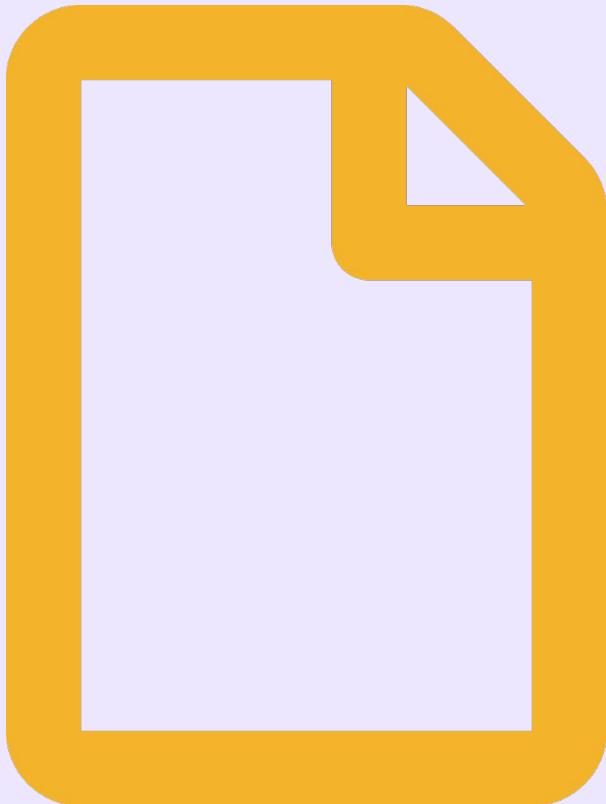


And now, we send the
keystroke...

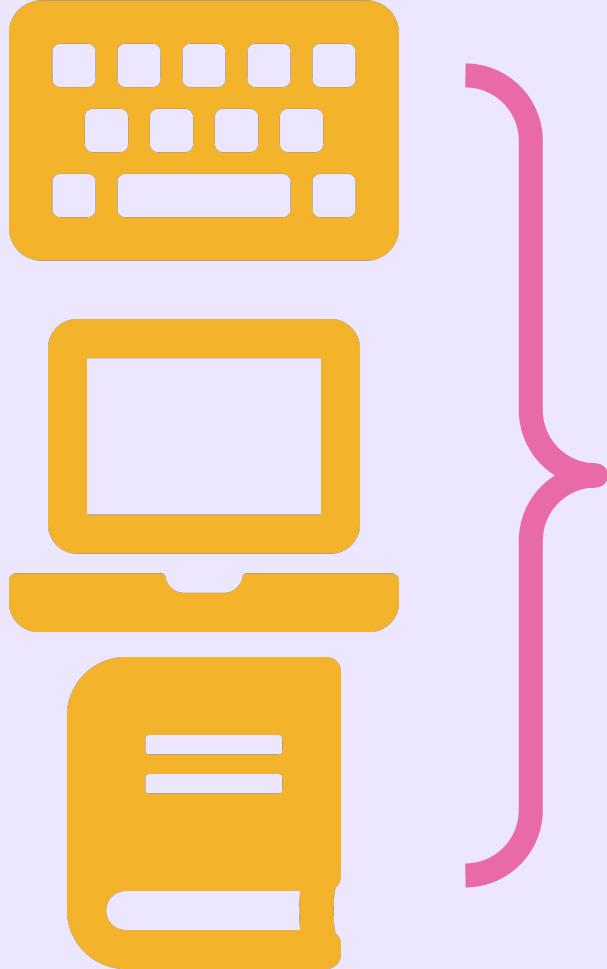


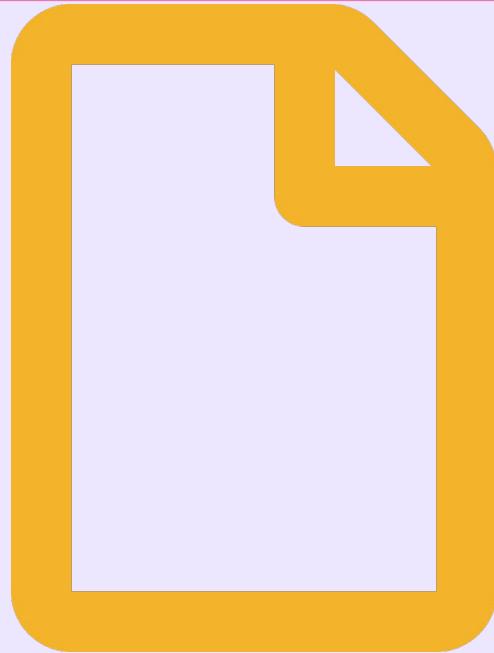
And now, we send the
keystroke...?

🌟 File Descriptors 🌟



=





' /dev/hidg0 '

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



♥ Neatly Planned Meal

npm Enterprise Products Solutions Resources Docs Support



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);

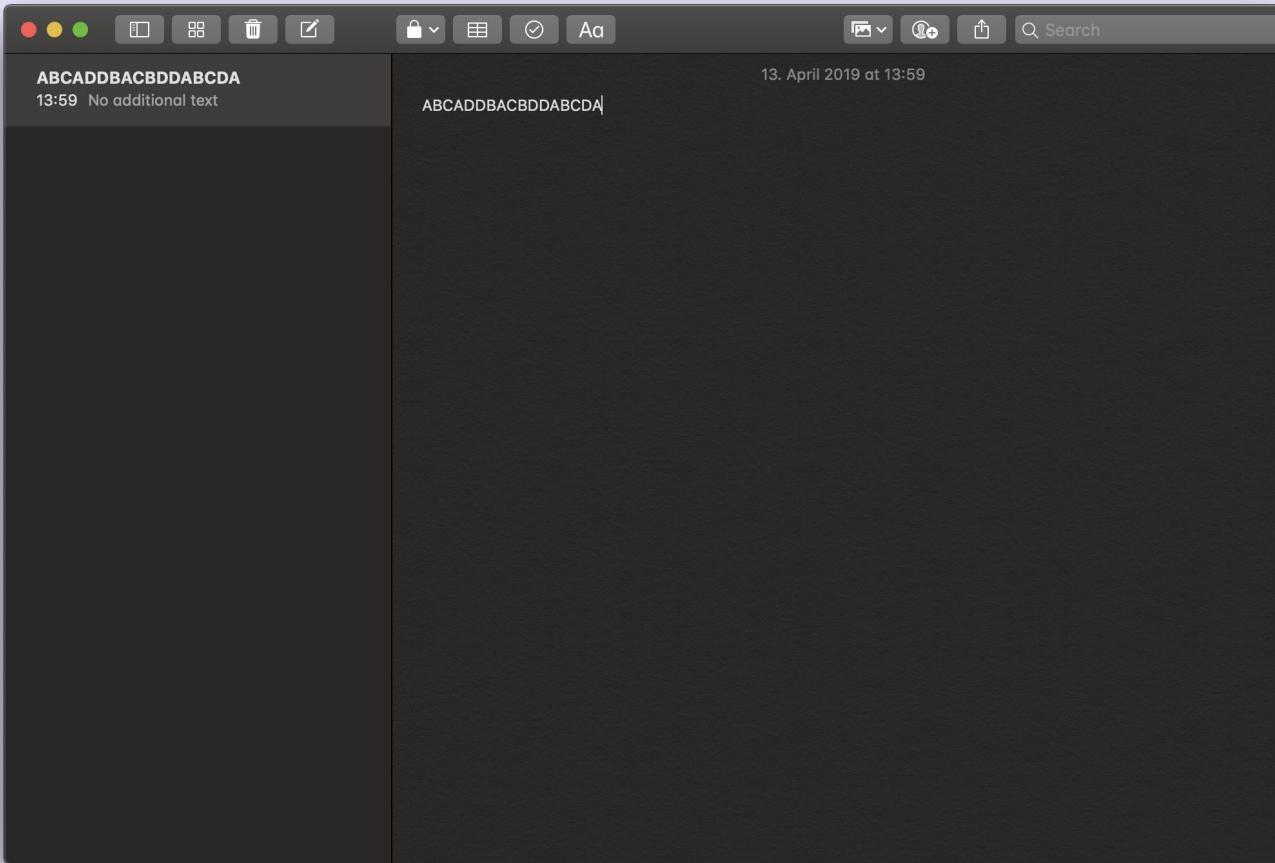
  let buffer = Uint8Array.from(keystroke);

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



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

node dance-mat.js

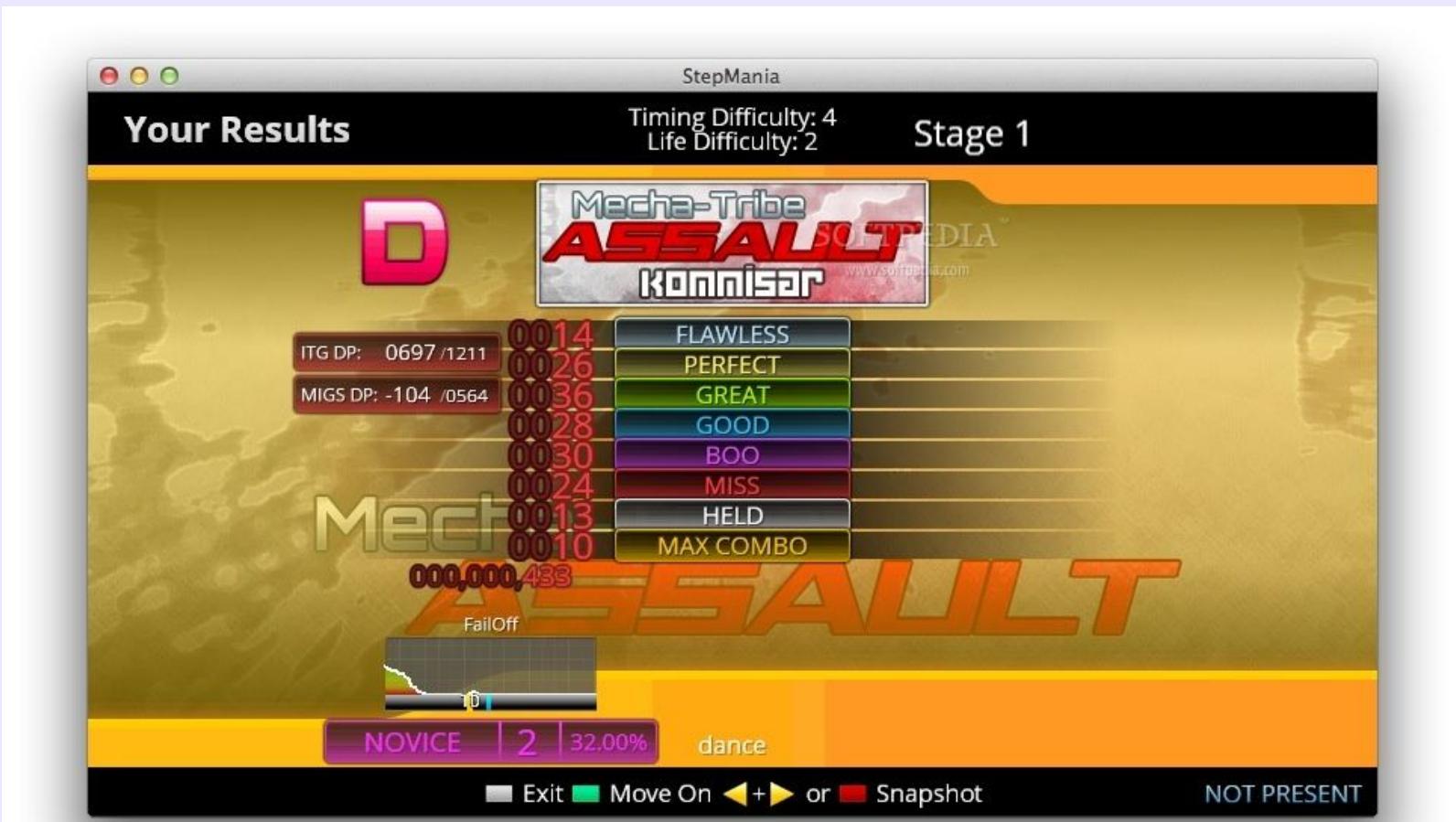














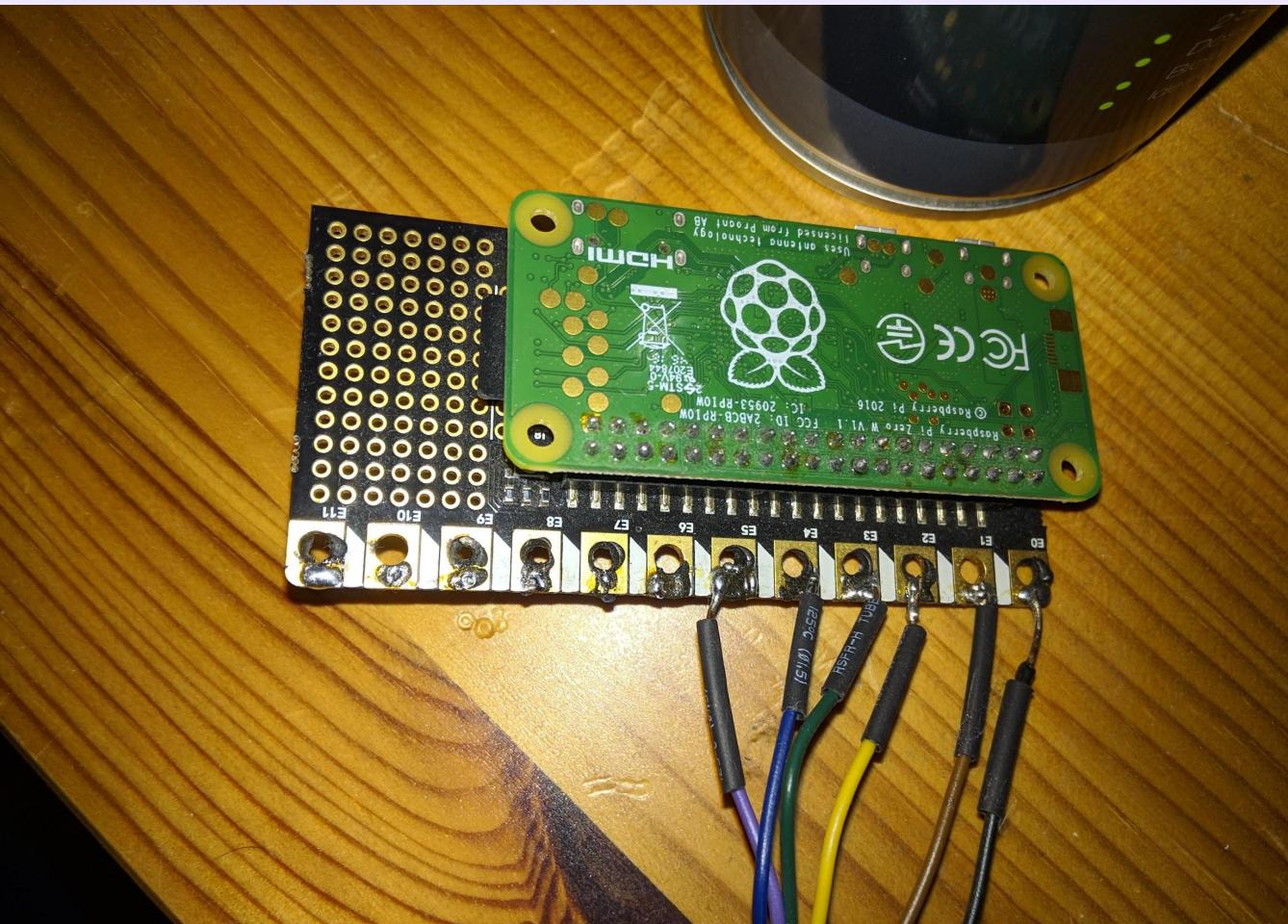
So are we set for life?

So are we set for life?



If we could do
it all over
again...

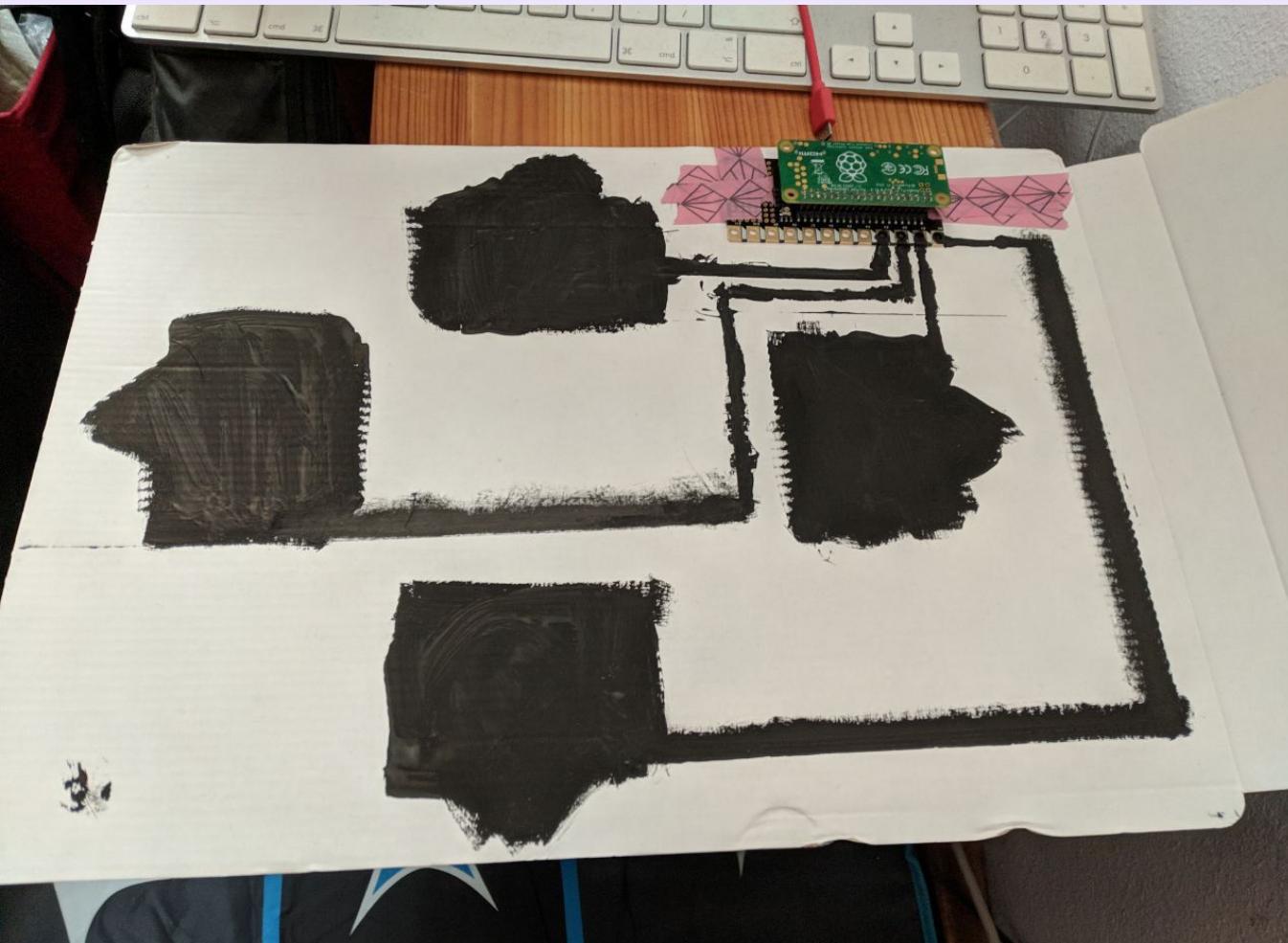
- Use an Arduino
- Smoother surface
- Physical instead of “paint” cabling

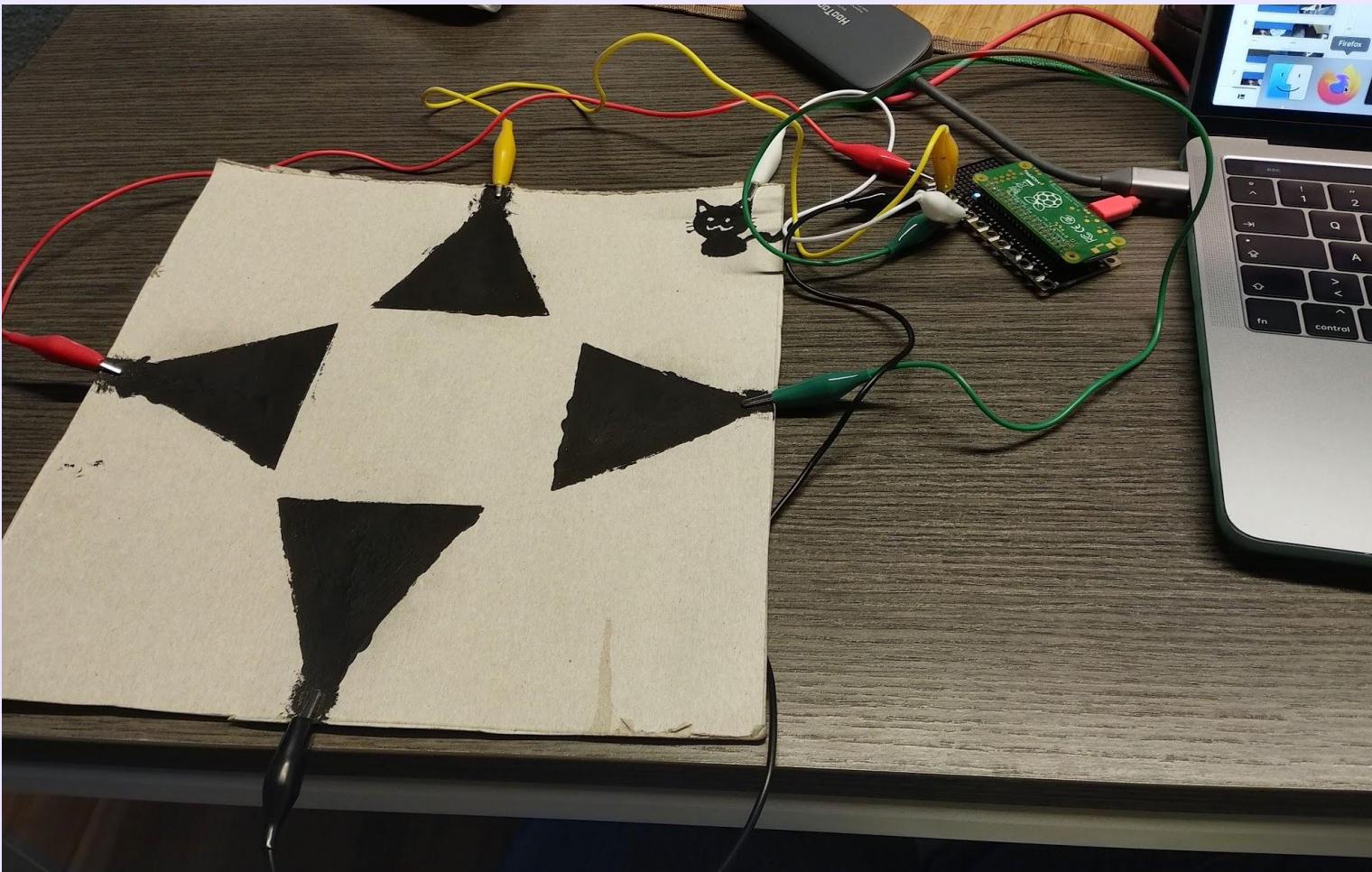


**But hey!
We learned a whole mess
of stuff.**



Dance for the nice people, Carmen.









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!

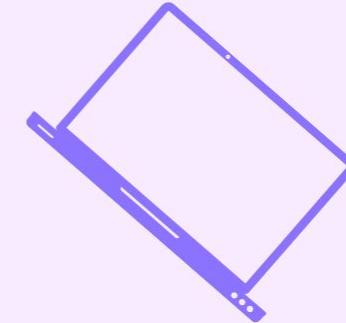


<https://carmenh.dev/dance-mat-js-24.pdf>



Carmen Huidobro

**Thank you,
Friends!**



carmenh.dev/card

