Si tu lo deseas, puedes bailar solo tienes que codear:

la historia de dance-mat.js

Ramón Huidobro

@hola_soy_milk

ramonh.dev/dance-mat-js.pdf

@hola_soy_milk

Advertencia:
Música buenísima de los 90

@hola_soy_milk

Me llamo Ramón (él)

De Chile, en Austria

Developer Advocate @ Suborbital

Consultante DevRel

Instructor de Desarrollo

Mozilla tech speaker

Mentor de Carrera Tech

Live Streamer

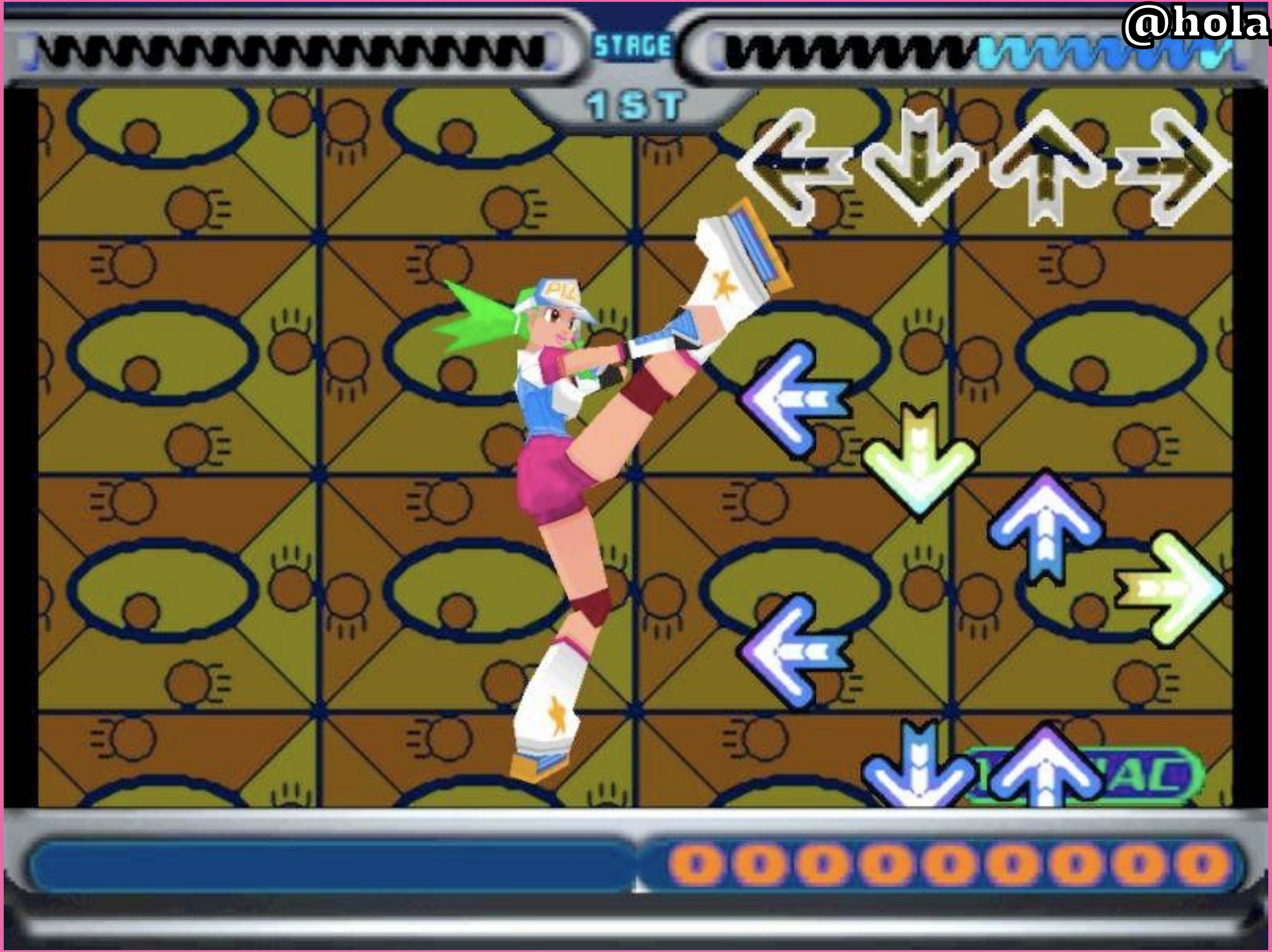
Ponente de keynote

HAPPY
GOOD
JAZZY
OK
CRAZY
PERFECT
FUNKY
GOOD
VIDEOGAMER.COM

@hola_soy_milk
×8
LTV

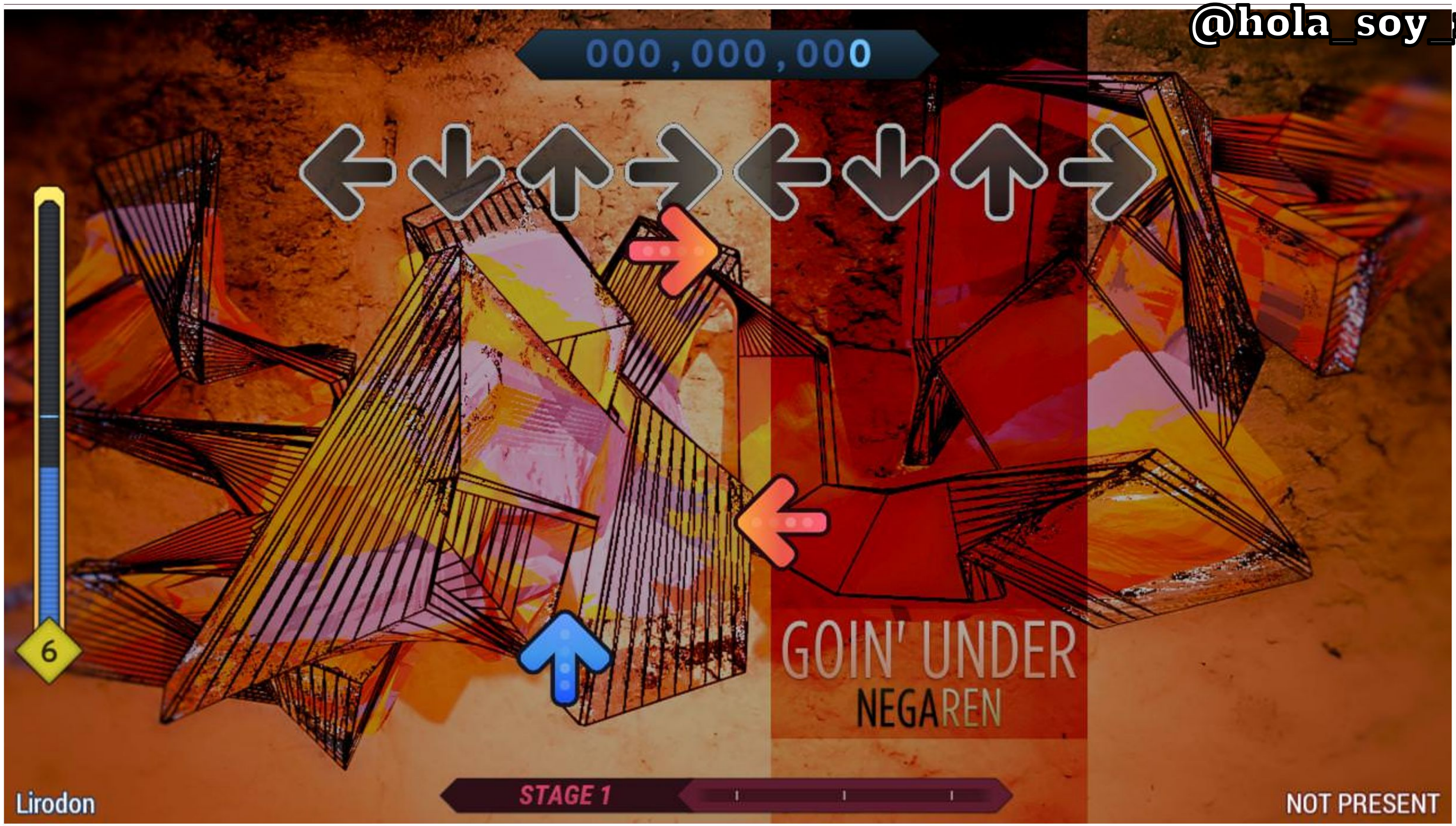
Dance Dance Revolution

DDR


STAGE
1ST
PIA

Y bueno, la música.

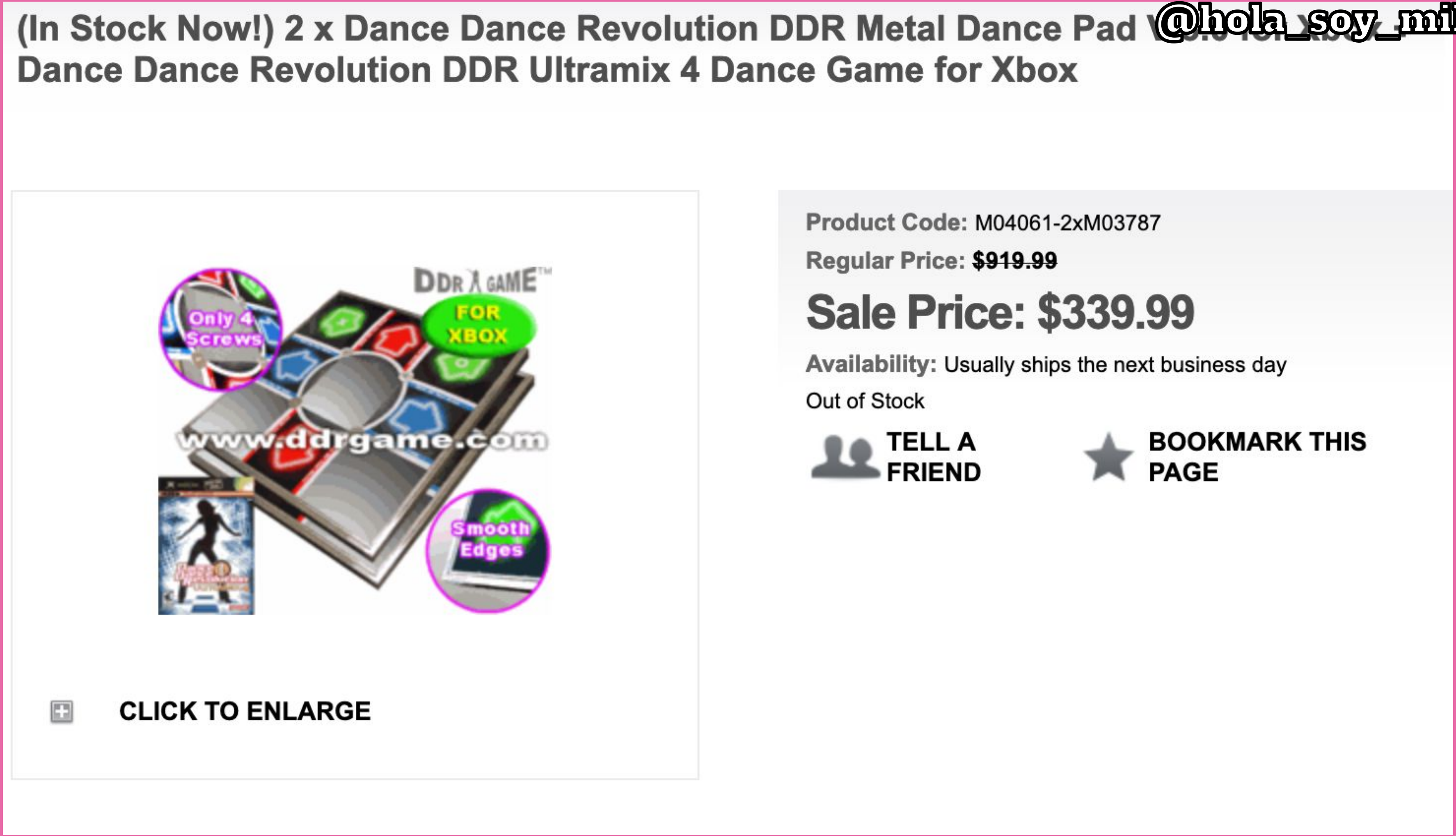
Santo cielo, la música.



https://www.stepmania.com/

START
SELECT
Show Me Your Best
Show Me Your Best
Show Me Your Best
Show Me Your Best
Show Me Your Best
Show Me Your Best
Show Me Your Best
Show Me Your Best
Stay Cool!

www.gamesdatabase.org
Dance Dance Revolution
2ND MIX
GAME OVER
CREDIT(S): 0
CREDIT(S): 0



(In Stock Now!) 2 x Dance Dance Revolution DDR Metal Dance Pad V[illegible]
Dance Dance Revolution DDR Ultramix 4 Dance Game for Xbox

CLICK TO ENLARGE

Product Code: M04061-2xM03787

Regular Price: ~~$919.99~~

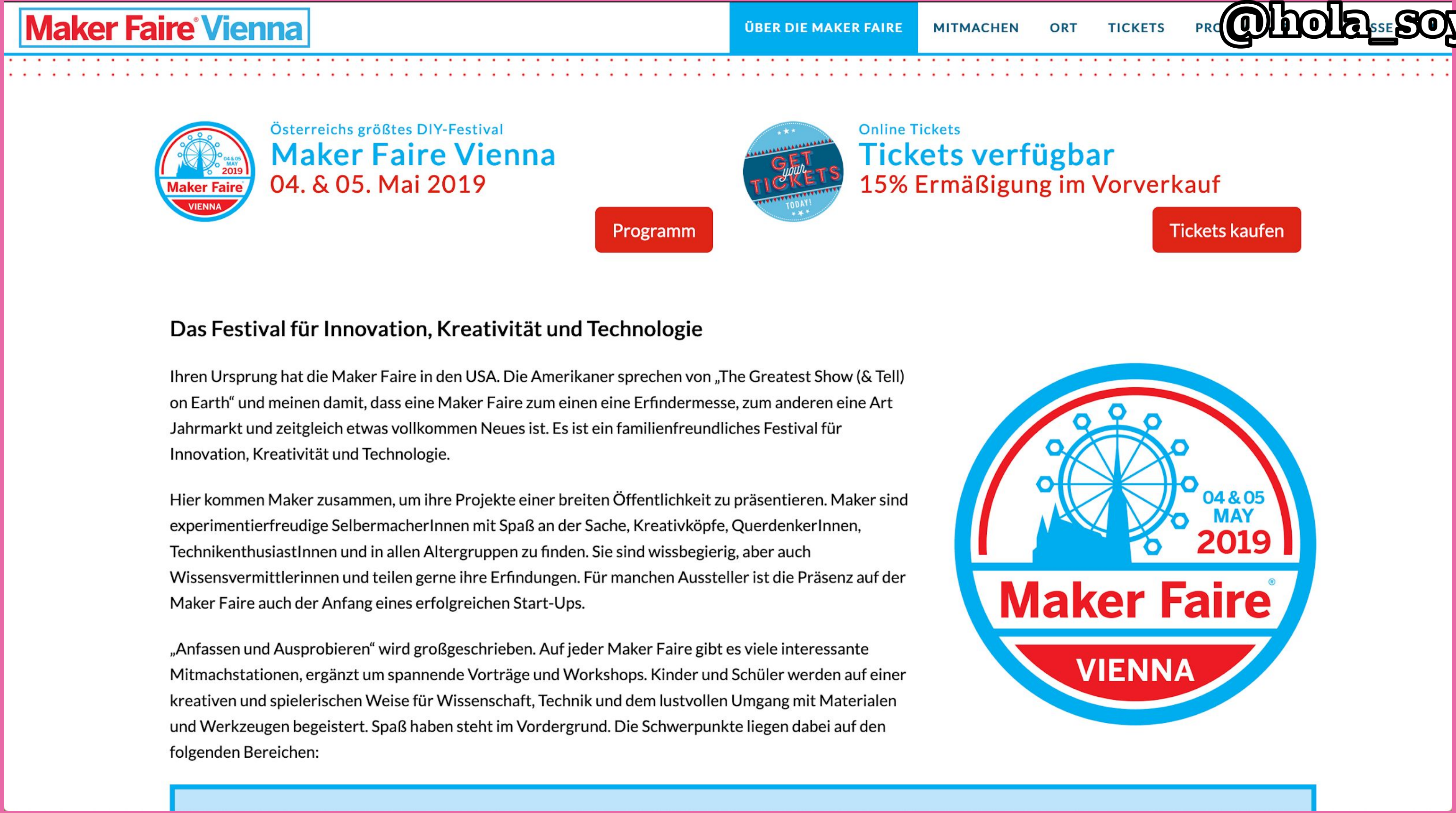
Sale Price: $339.99

Availability: Usually ships the next business day

Out of Stock

TELL A FRIEND

BOOKMARK THIS PAGE



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

Online Tickets
Tickets verfügbar
15% Ermäßigung im Vorverkauf

Das Festival für Innovation, Kreativität und Technologie

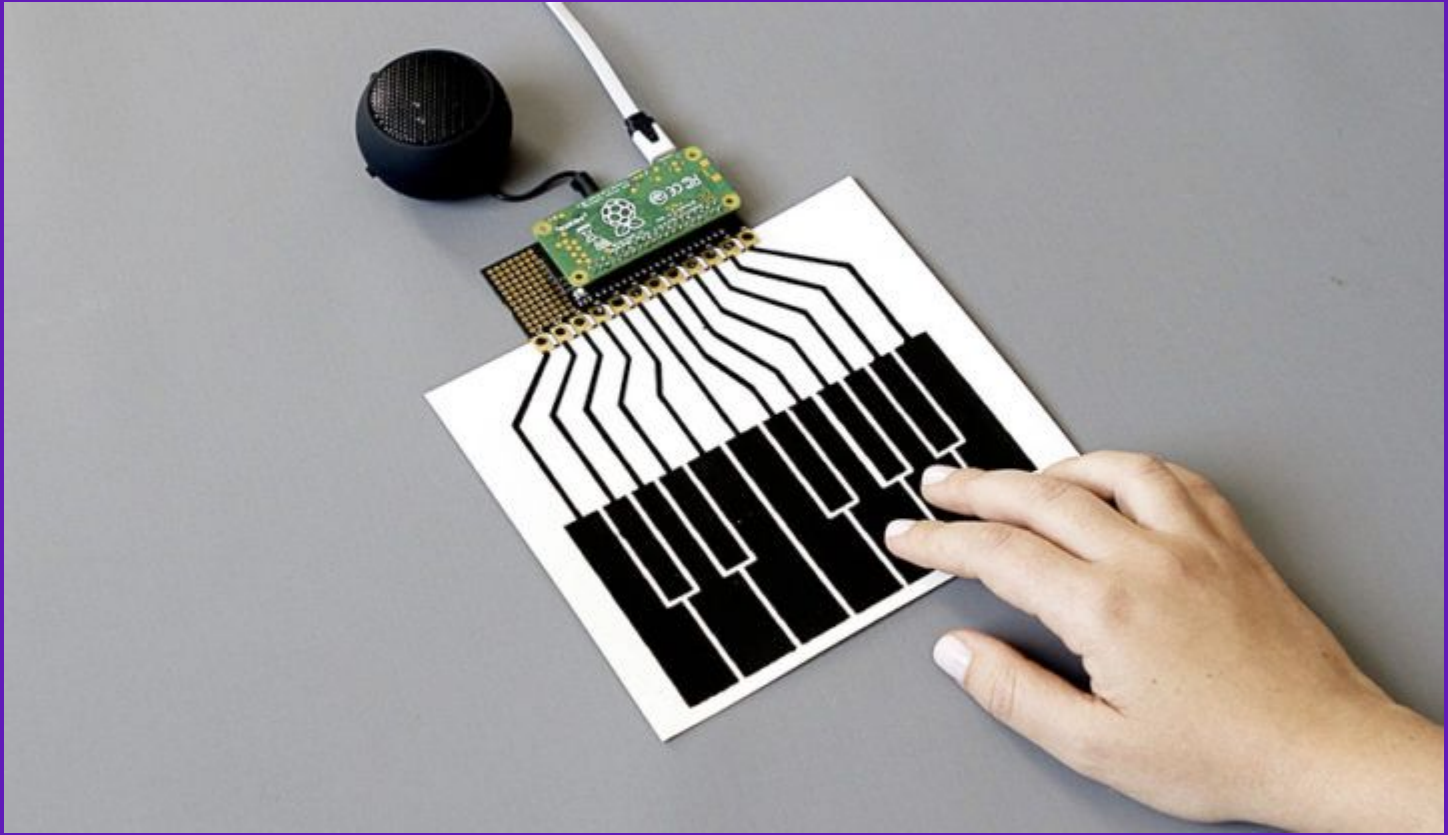
Ihren Ursprung hat die Maker Faire in den USA. Die Amerikaner sprechen von „The Greatest Show (& Tell) on Earth" und meinen damit, dass eine Maker Faire zum einen eine Erfindermesse, zum anderen eine Art Jahrmarkt und zeitgleich etwas vollkommen Neues ist. Es ist ein familienfreundliches Festival für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

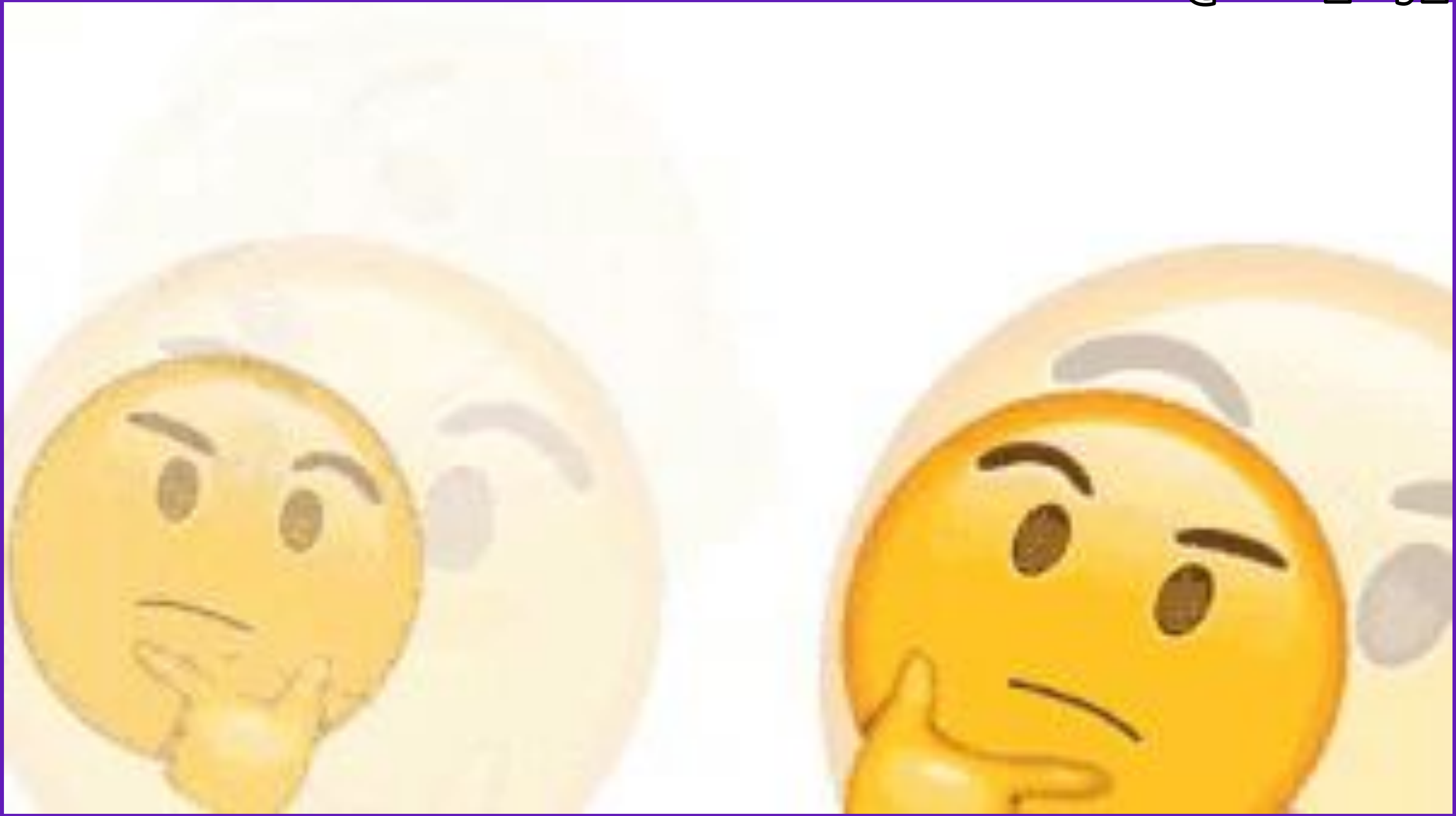
„Anfassen und Ausprobieren" wird großgeschrieben. Auf jeder Maker Faire gibt es viele interessante Mitmachstationen, ergänzt um spannende Vorträge und Workshops. Kinder und Schüler werden auf einer kreativen und spielerischen Weise für Wissenschaft, Technik und dem lustvollen Umgang mit Materialen und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

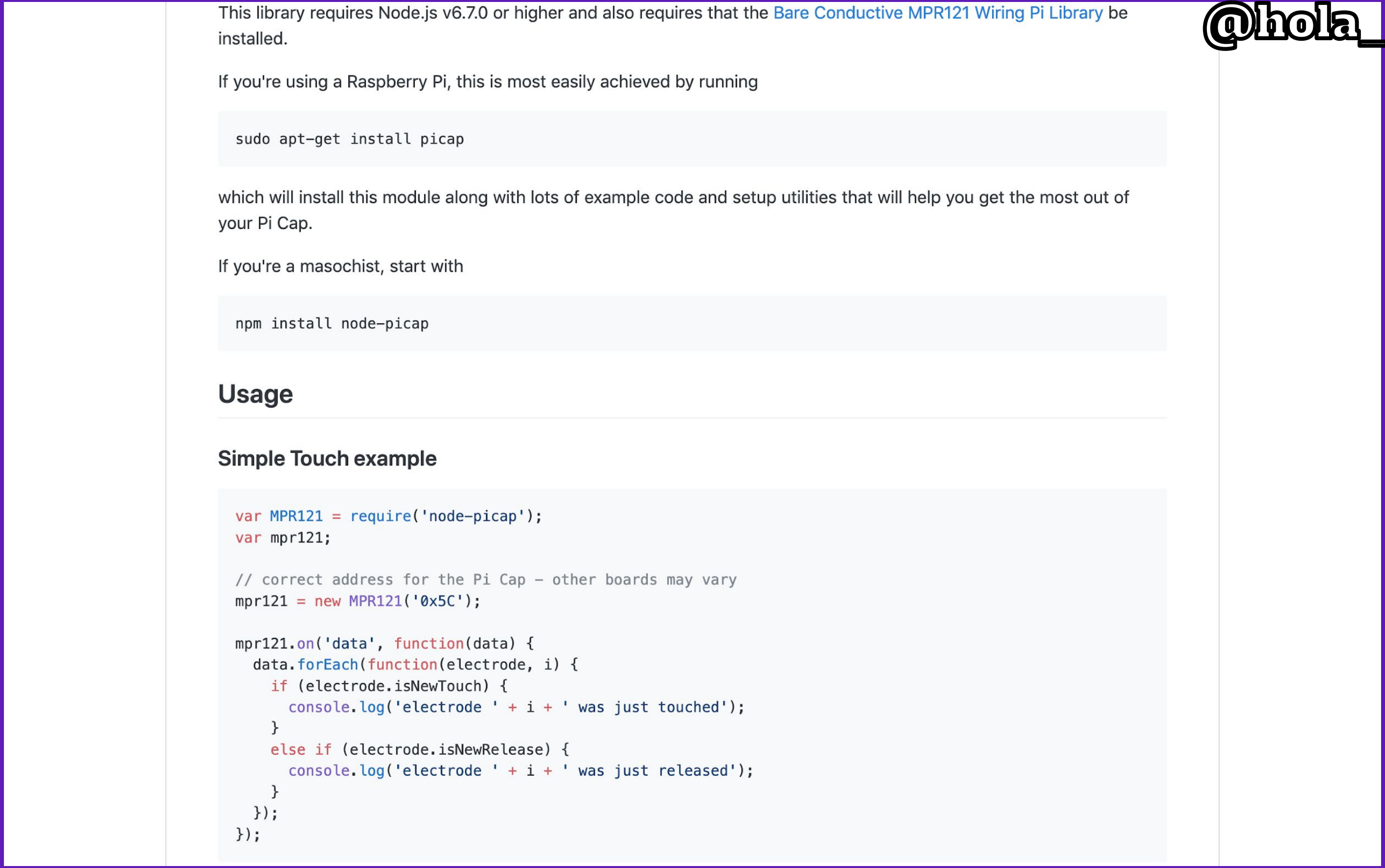
https://www.stepmania.com/

@hola_soy_milk



@hola_soy_milk




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
```

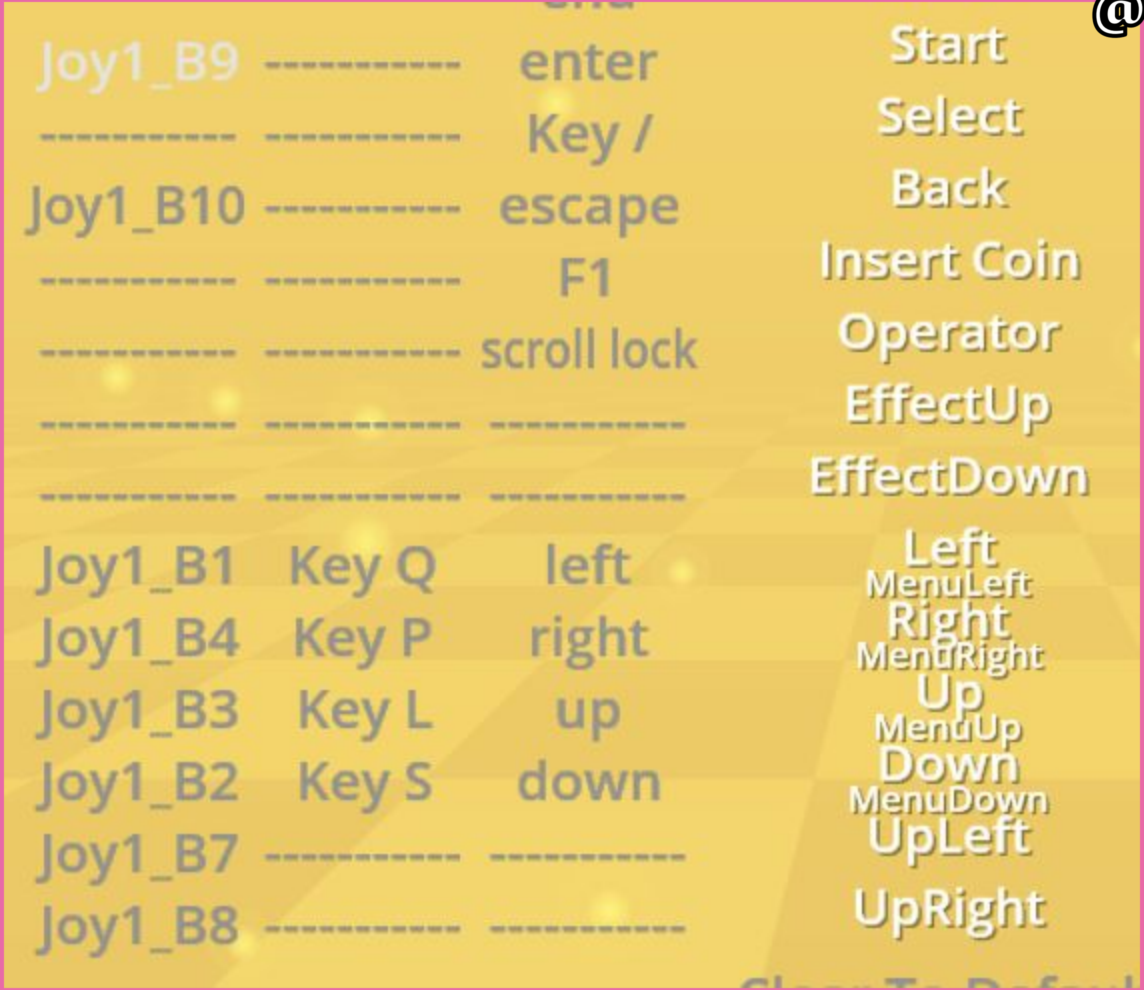
Usage

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');
    }
  });
});
```

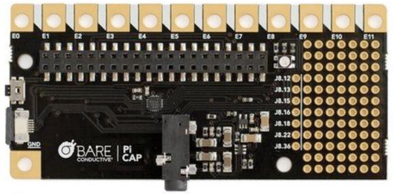
@hola_soy_milk

https://github.com/BareConductive/node-picap



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

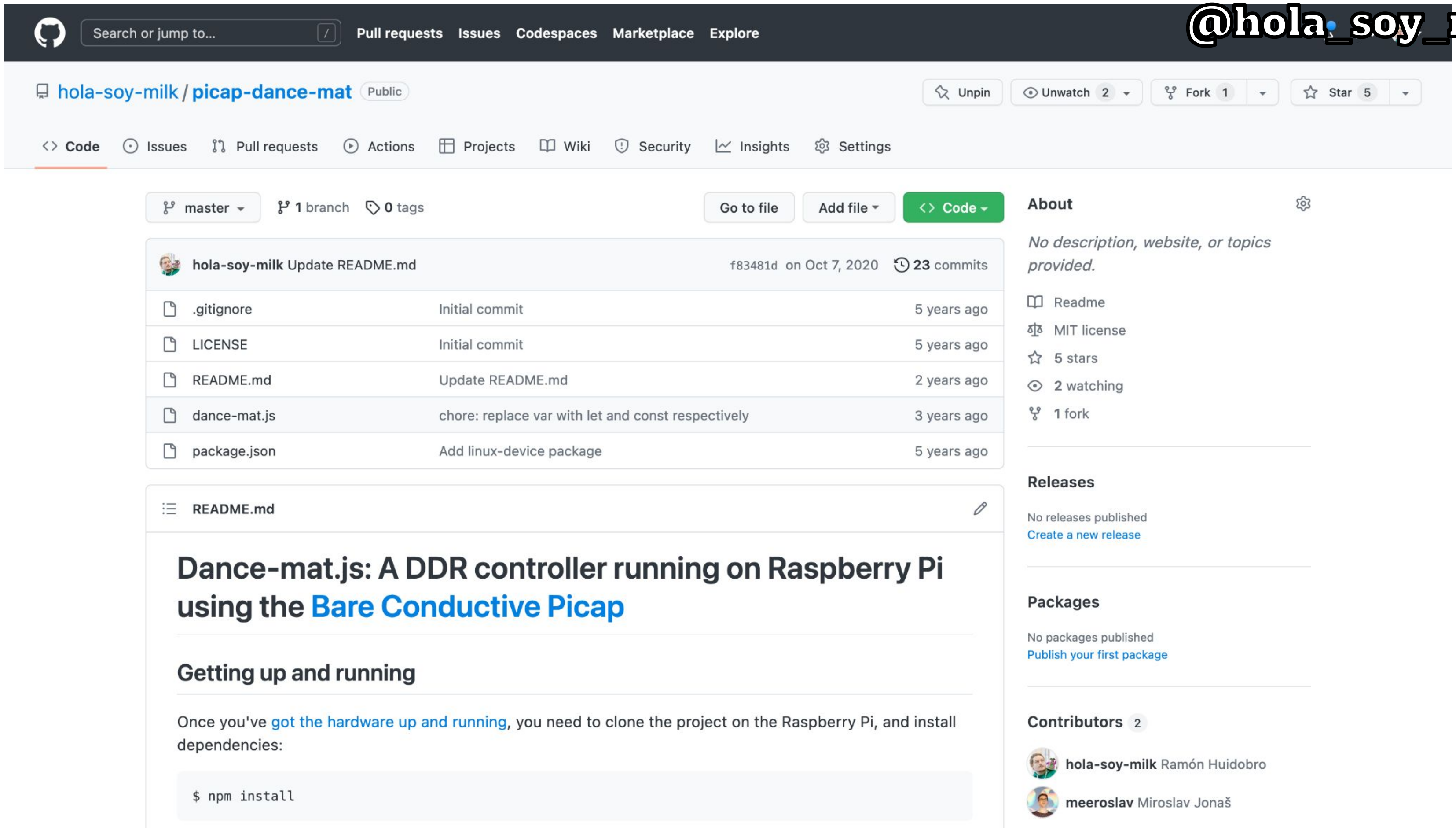
@hola_soy_milk



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



hola-soy-milk / picap-dance-mat Public

hola-soy-milk Update README.md f83481d on Oct 7, 2020 23 commits

.gitignore	Initial commit	5 years ago
LICENSE	Initial commit	5 years ago
README.md	Update README.md	2 years ago
dance-mat.js	chore: replace var with let and const respectively	3 years ago
package.json	Add linux-device package	5 years ago

README.md

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
```

About

No description, website, or topics provided.

Readme
MIT license
5 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 Ramón Huidobro
meeroslav Miroslav Jonaš

https://github.com/BareConductive/node-picap

@hola_soy_milk

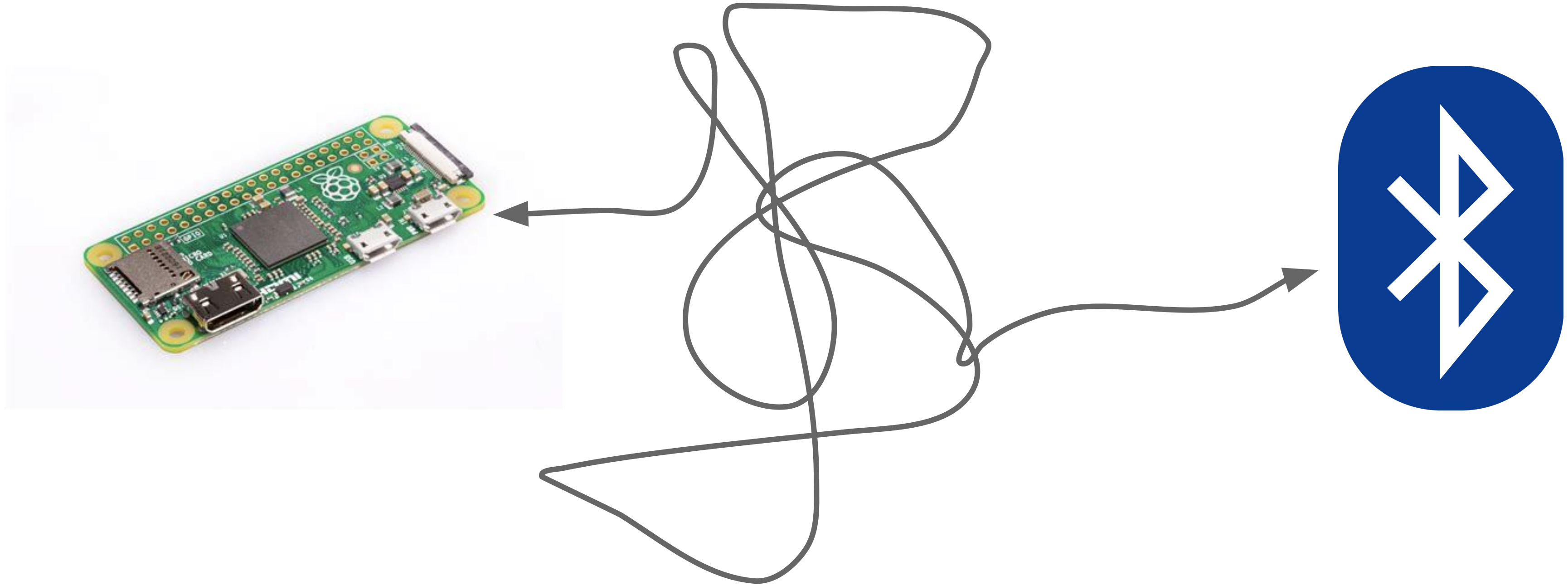
@hola_soy_milk

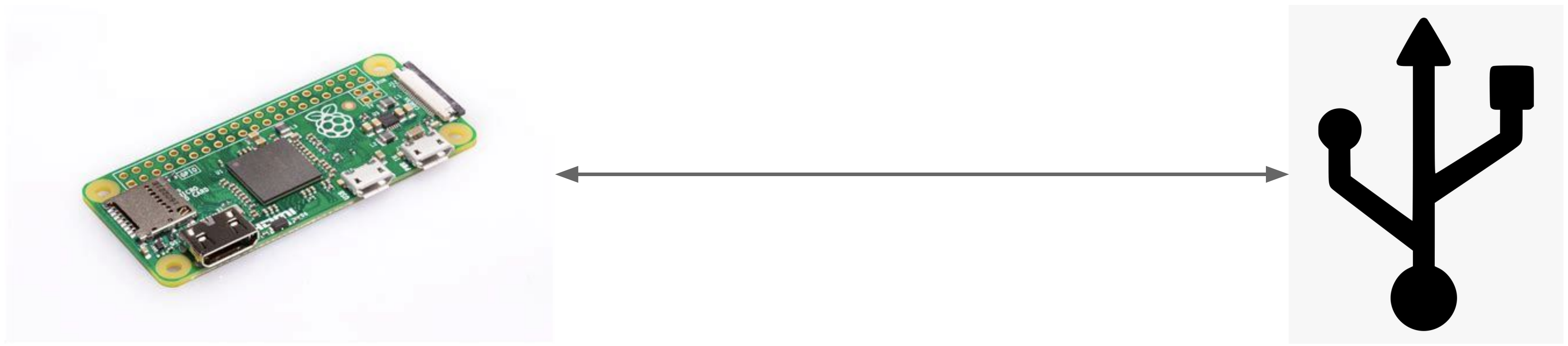
@hola_soy_milk

```
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);
  }
});
```

@hola_soy_milk

@hola_soy_milk





!=

@hola_soy_milk

@hola_soy_milk

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

Gadget de USB Linux gestionado con configfs

@hola_soy_milk

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

1. Activar Raspberry Pi OTG (USB on the go)
2. Añadir una boot script para activar el modo de teclado USB HID con configfs

```
pi@raspberrypi:~ $ echo "dtoverlay=dwc2" | sudo tee -a /boot/config.txt
pi@raspberrypi:~ $ echo "dwc2" | sudo tee -a /etc/modules
pi@raspberrypi:~ $ sudo echo "libcomposite" | sudo tee -a /etc/modules
```

1. ✅ Activar Raspberry Pi OTG (USB on the go)
2. Añadir una boot script para activar el modo de teclado USB HID con configfs

@hola_soy_milk

```bash
#!/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 "Ramon 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\\x0
08\\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
```

@hola_soy_milk

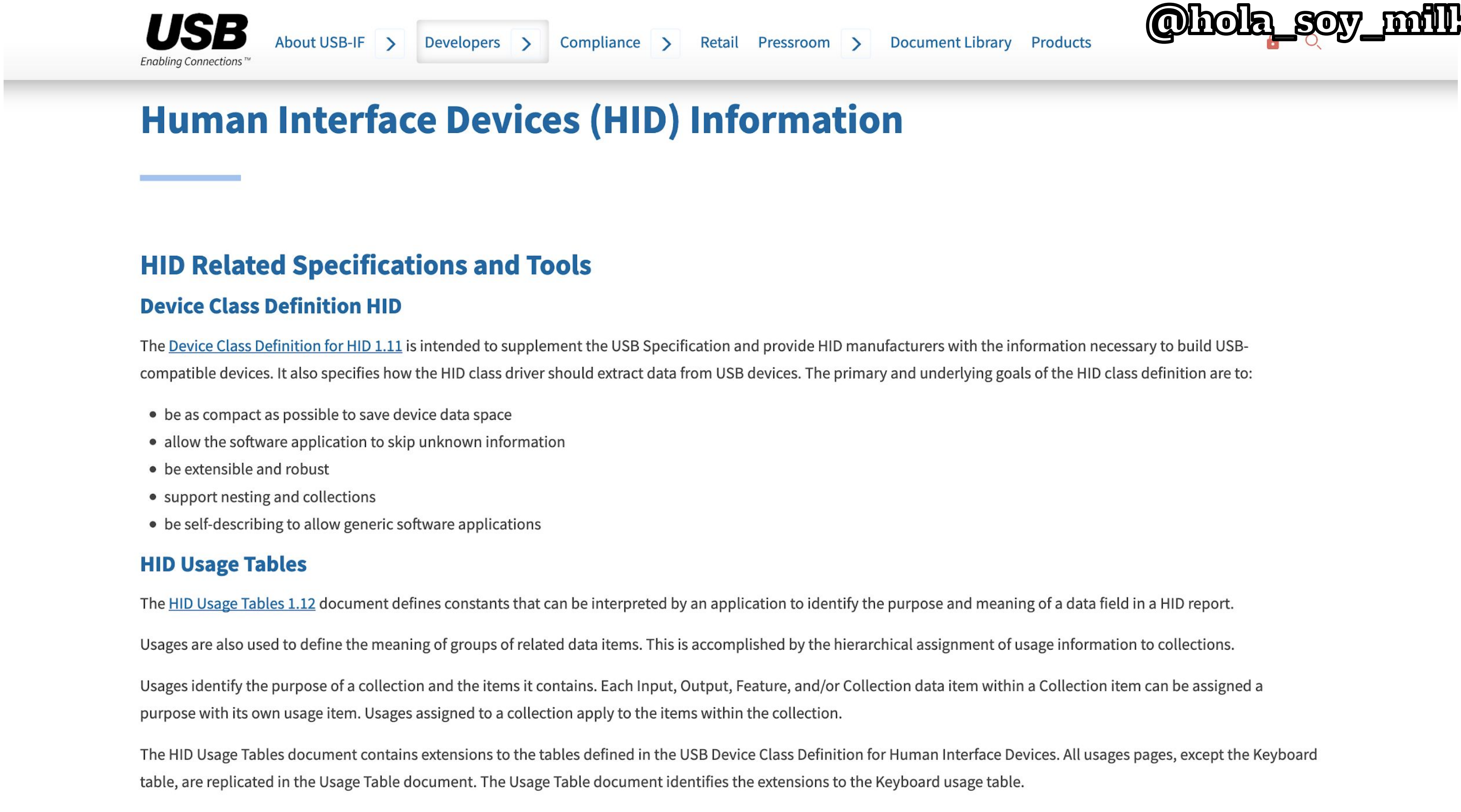
```bash
#!/bin/bash
cd /sys/kernel/config/usb_gadget/
mkdir -p ddrusb
cd ddrusb
echo 0x1d6b > idVendor # Linux Foundation
echo 0x0104 > idProduct # Multifunction Composite Gadget
echo 0x0100 > bcdDevice # v1.0.0
echo 0x0200 > bcdUSB # USB2
mkdir -p strings/0x409
echo "fedcba9876543210" > strings/0x409/serialnumber
echo "Ramon 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\\x0
08\\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 es una filesystem virtual montada al iniciar.

Nos permite gestionar el kernel de Linux.

¡Es increible cuantas opciones nos ofrece la Raspberry Pi OS!

@hola_soy_milk



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

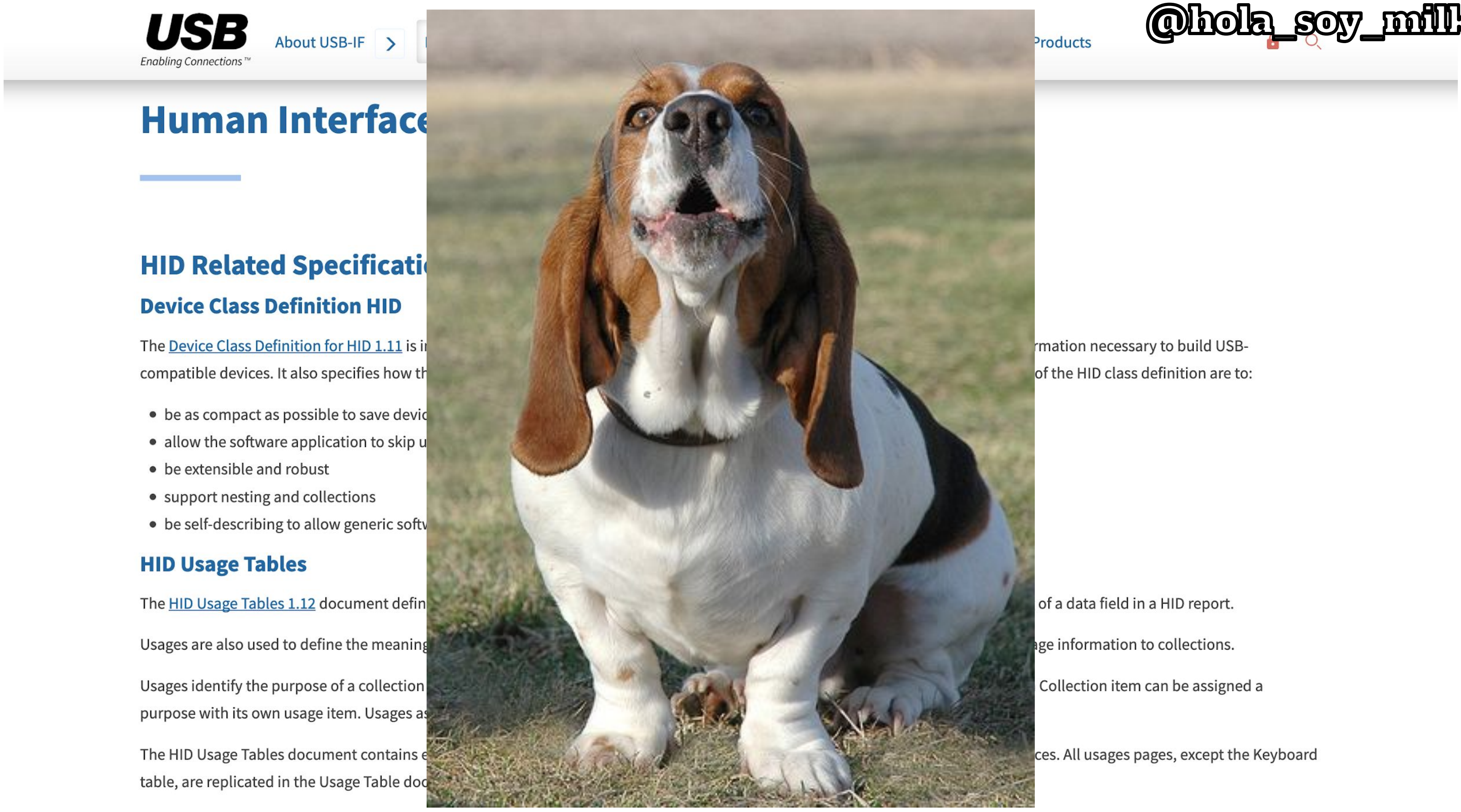
The HID Usage Tables 1.12 document defines constants that can be interpreted by an application to identify the purpose and meaning of a data field in a HID report.

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.

@hola_soy_milk



Human Interface

HID Related Specificati

Device Class Definition HID

The Device Class Definition for HID 1.11 is i rmation necessary to build USB-compatible devices. It also specifies how th of the HID class definition are to:

- be as compact as possible to save devic
- allow the software application to skip u
- be extensible and robust
- support nesting and collections
- be self-describing to allow generic softv

HID Usage Tables

The HID Usage Tables 1.12 document defin of a data field in a HID report.

Usages are also used to define the meaning age information to collections.

Usages identify the purpose of a collection Collection item can be assigned a purpose with its own usage item. Usages as

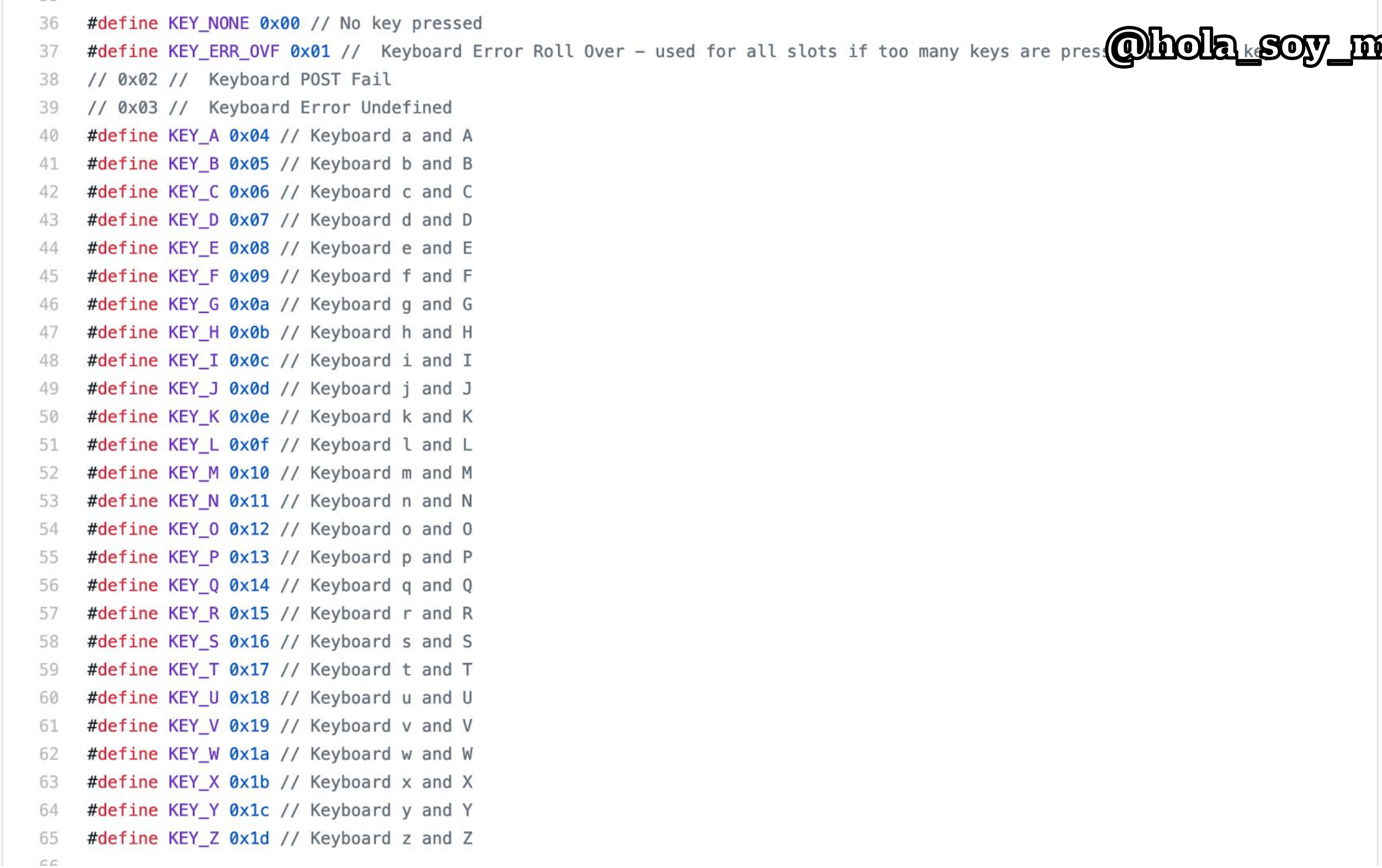
The HID Usage Tables document contains e ces. All usages pages, except the Keyboard table, are replicated in the Usage Table doc

Asking for help is A-OK!

1 pulsada de tecla

=

1 byte array de 8 valores hex



```
#define KEY_NONE 0x00 // No key pressed
#define KEY_ERR_OVF 0x01 //  Keyboard Error Roll Over - used for all slots if too many keys are press
// 0x02 //  Keyboard POST Fail
// 0x03 //  Keyboard Error Undefined
#define KEY_A 0x04 // Keyboard a and A
#define KEY_B 0x05 // Keyboard b and B
#define KEY_C 0x06 // Keyboard c and C
#define KEY_D 0x07 // Keyboard d and D
#define KEY_E 0x08 // Keyboard e and E
#define KEY_F 0x09 // Keyboard f and F
#define KEY_G 0x0a // Keyboard g and G
#define KEY_H 0x0b // Keyboard h and H
#define KEY_I 0x0c // Keyboard i and I
#define KEY_J 0x0d // Keyboard j and J
#define KEY_K 0x0e // Keyboard k and K
#define KEY_L 0x0f // Keyboard l and L
#define KEY_M 0x10 // Keyboard m and M
#define KEY_N 0x11 // Keyboard n and N
#define KEY_O 0x12 // Keyboard o and O
#define KEY_P 0x13 // Keyboard p and P
#define KEY_Q 0x14 // Keyboard q and Q
#define KEY_R 0x15 // Keyboard r and R
#define KEY_S 0x16 // Keyboard s and S
#define KEY_T 0x17 // Keyboard t and T
#define KEY_U 0x18 // Keyboard u and U
#define KEY_V 0x19 // Keyboard v and V
#define KEY_W 0x1a // Keyboard w and W
#define KEY_X 0x1b // Keyboard x and X
#define KEY_Y 0x1c // Keyboard y and Y
#define KEY_Z 0x1d // Keyboard z and Z
```

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

@hola_soy_milk

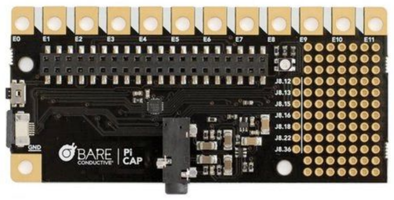
```
const p1Left = 0x04; // A
const p1Right = 0x05; // B
const p1Up = 0x06; // C
const p1Down = 0x07; // D
```

(Picture Credit: Cassie Keenum / 500px/Getty Images)

```
// 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;
}
```

@hola_soy_milk

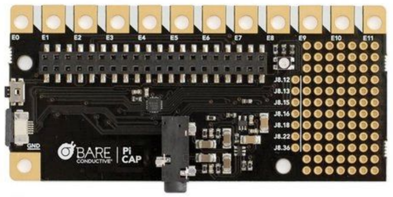


@hola_soy_milk

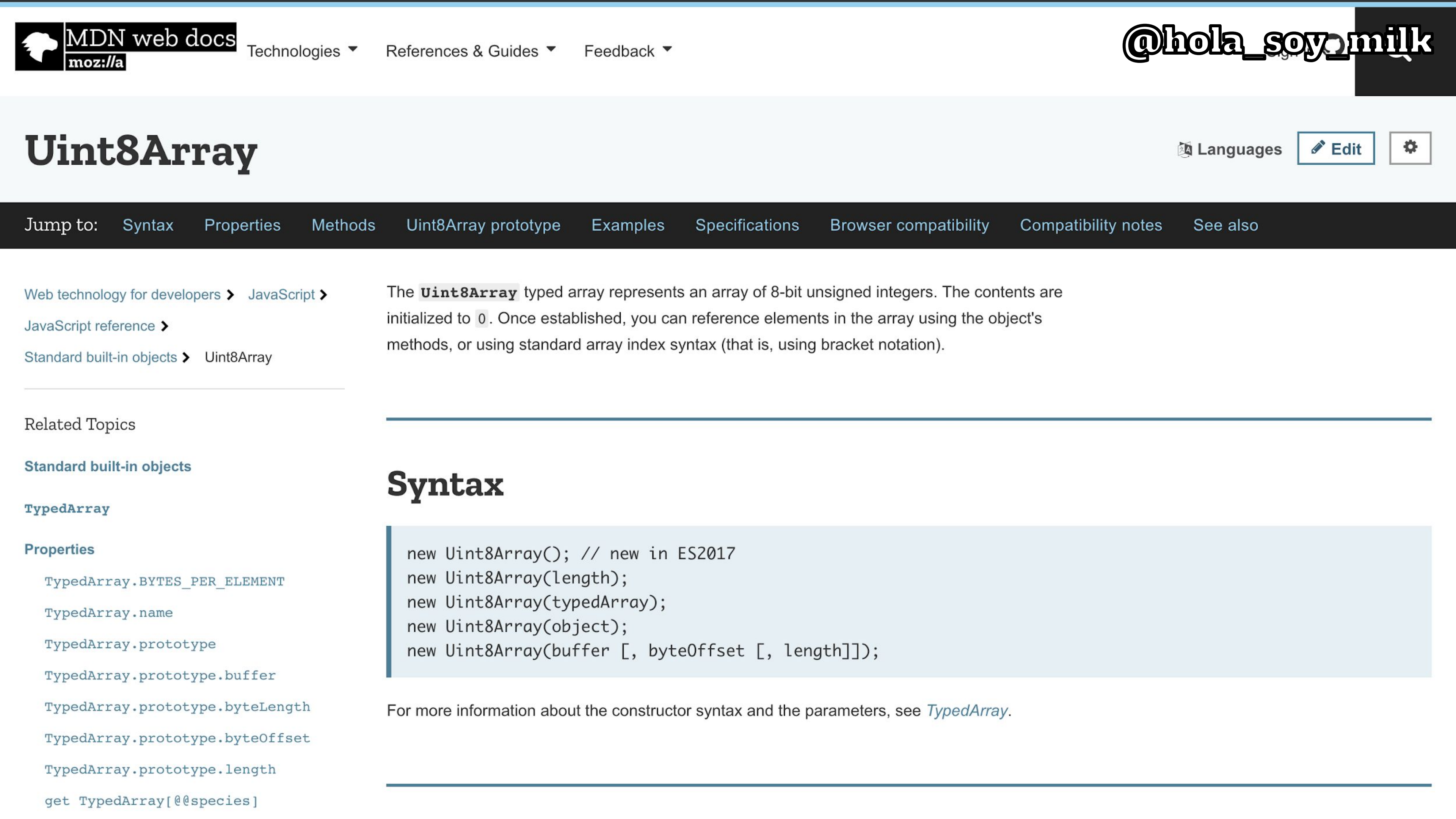
```
// Process touches
mpr121.on('data', (data) => {
  let keys = parsePressedKeys(data);
  let keystroke = keystrokeFromPressedKeys(keys);

});
```

```
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);
});
```



Y ahora, el byte array………….



Uint8Array

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*.

@hola_soy_milk

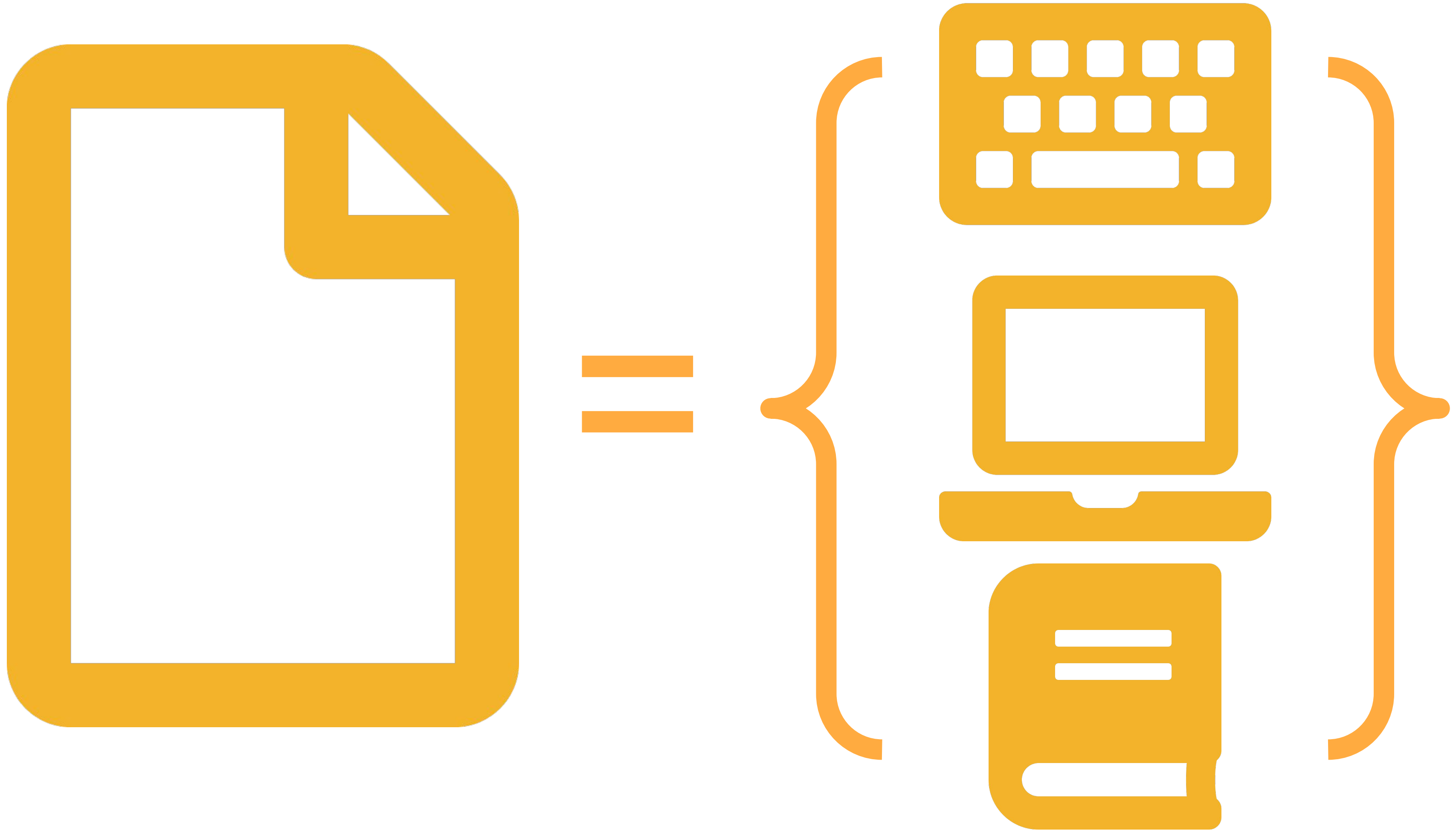
```
// Process touches
mpr121.on('data', (data) => {
    let keys = parsePressedKeys(data);
    let keystroke = keystrokeFromPressedKeys(keys);

    let buffer = Uint8Array.from(keystroke);

});
```

Y de ahí,
A pulsar el teclado...

@hola_soy_milk

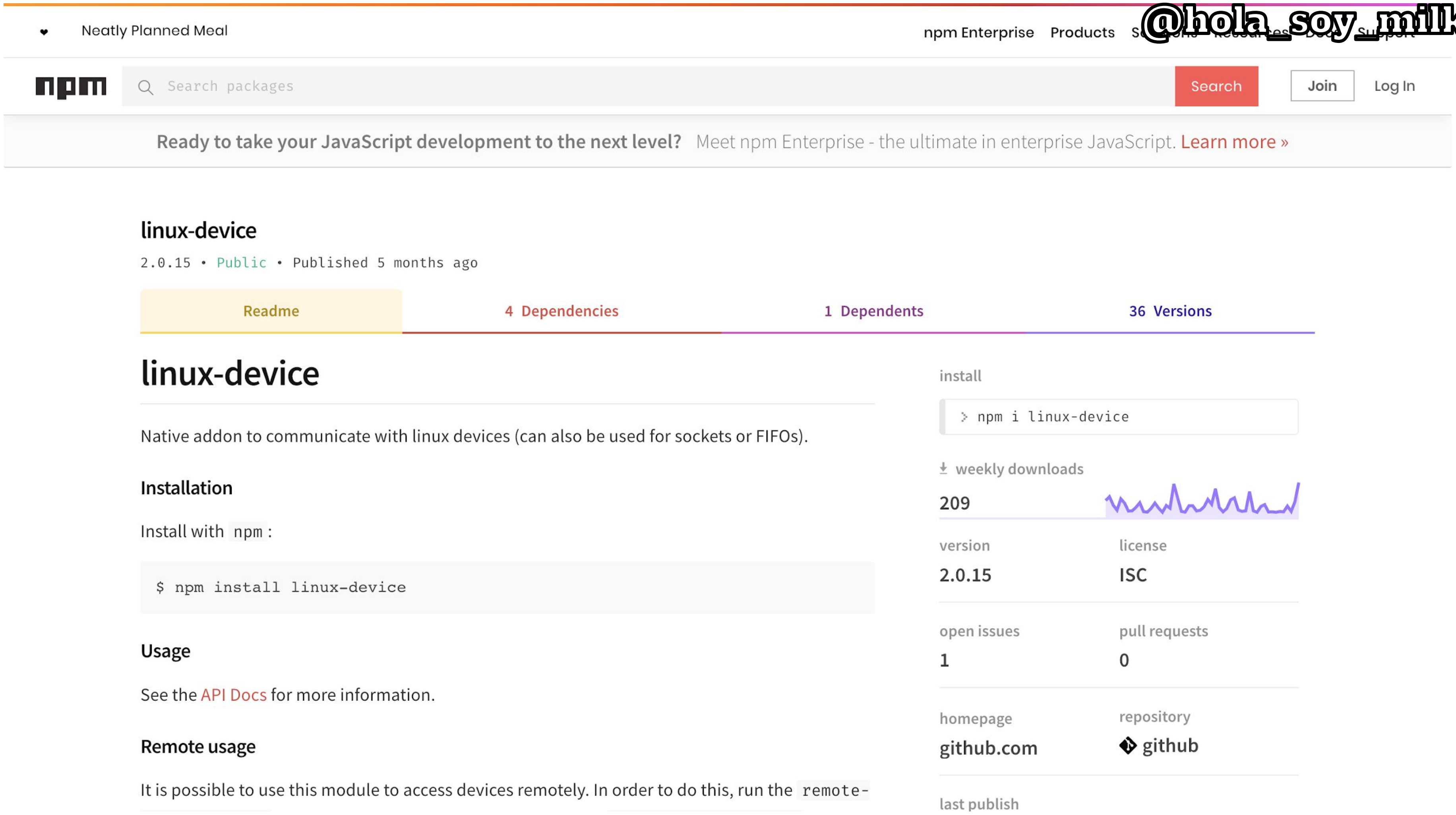
Y de ahí,
A pulsar el
teclado............?

FILE DESCRIPTORS



```
'/dev/hidg0'
```

```
pi@raspberrypi:~ $ echo "blablabla soy un teclado amigues" | sudo tee -a /dev/hidg0
```



Neatly Planned Meal

npm Enterprise Products Solutions Resources Docs Support

npm

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

It is possible to use this module to access devices remotely. In order to do this, run the `remote-`

install

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

weekly downloads

209

version	license
2.0.15	ISC

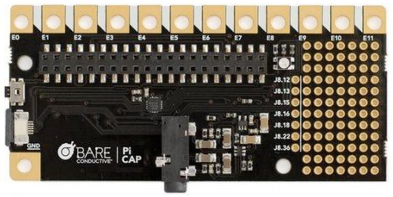
open issues	pull requests
1	0

homepage	repository
github.com	github

last publish

```
const DeviceHandle = require('linux-device');

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

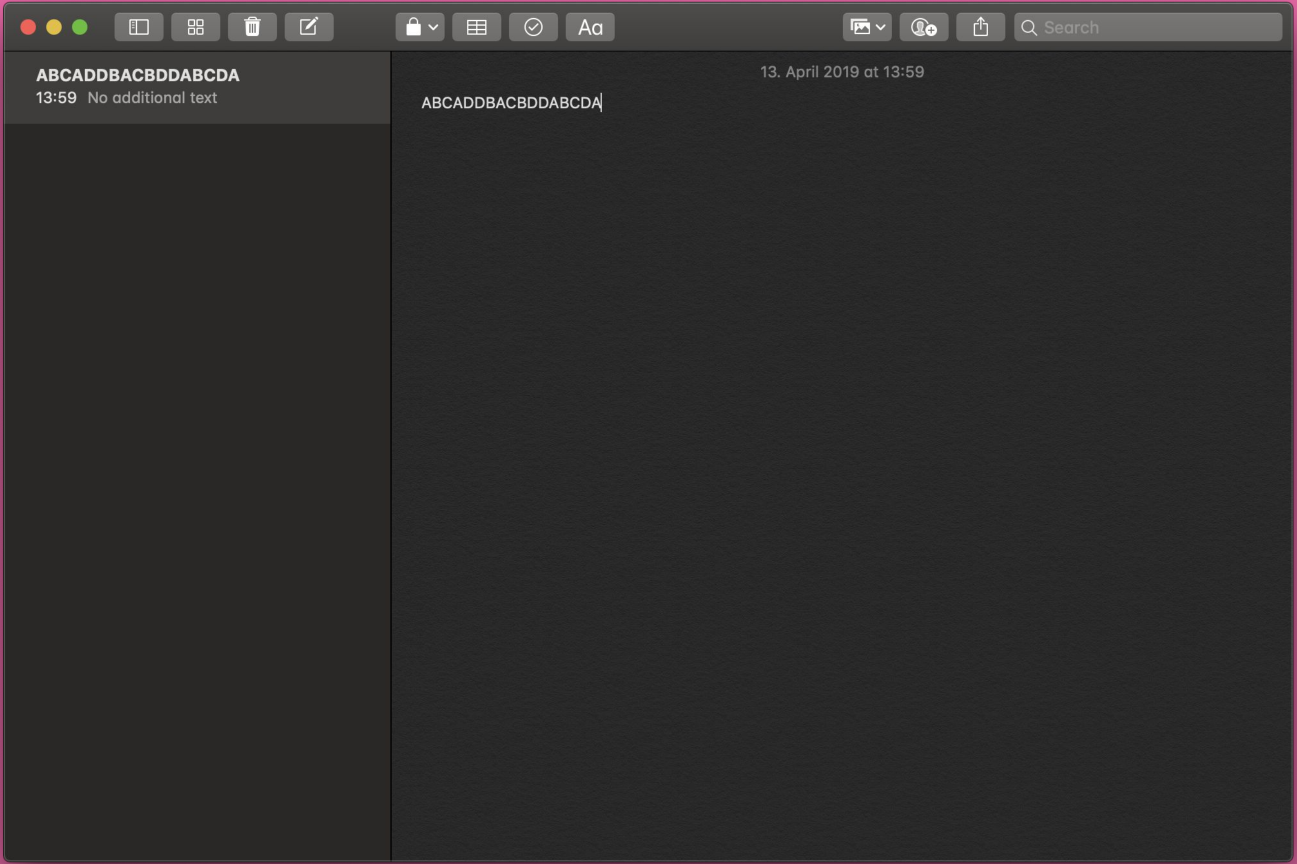
```
// Process touches
mpr121.on('data', (data) => {
   let keys = parsePressedKeys(data);
   keystroke = keystrokeFromPressedKeys(keys);

   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
```

@hola_soy_milk

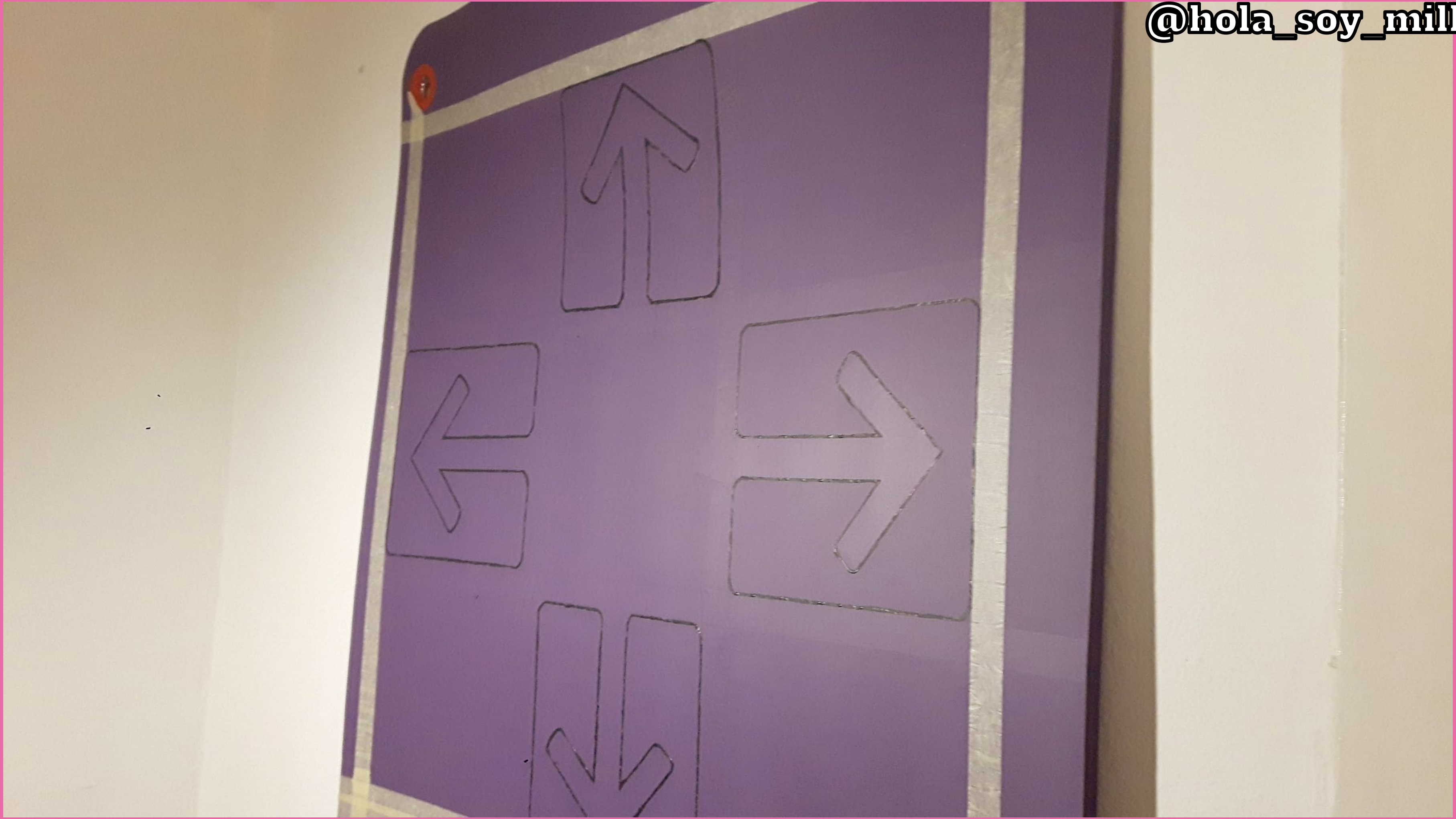


ABCADDBACBDDABCDA
13:59 No additional text

13. April 2019 at 13:59

ABCADDBACBDDABCDA

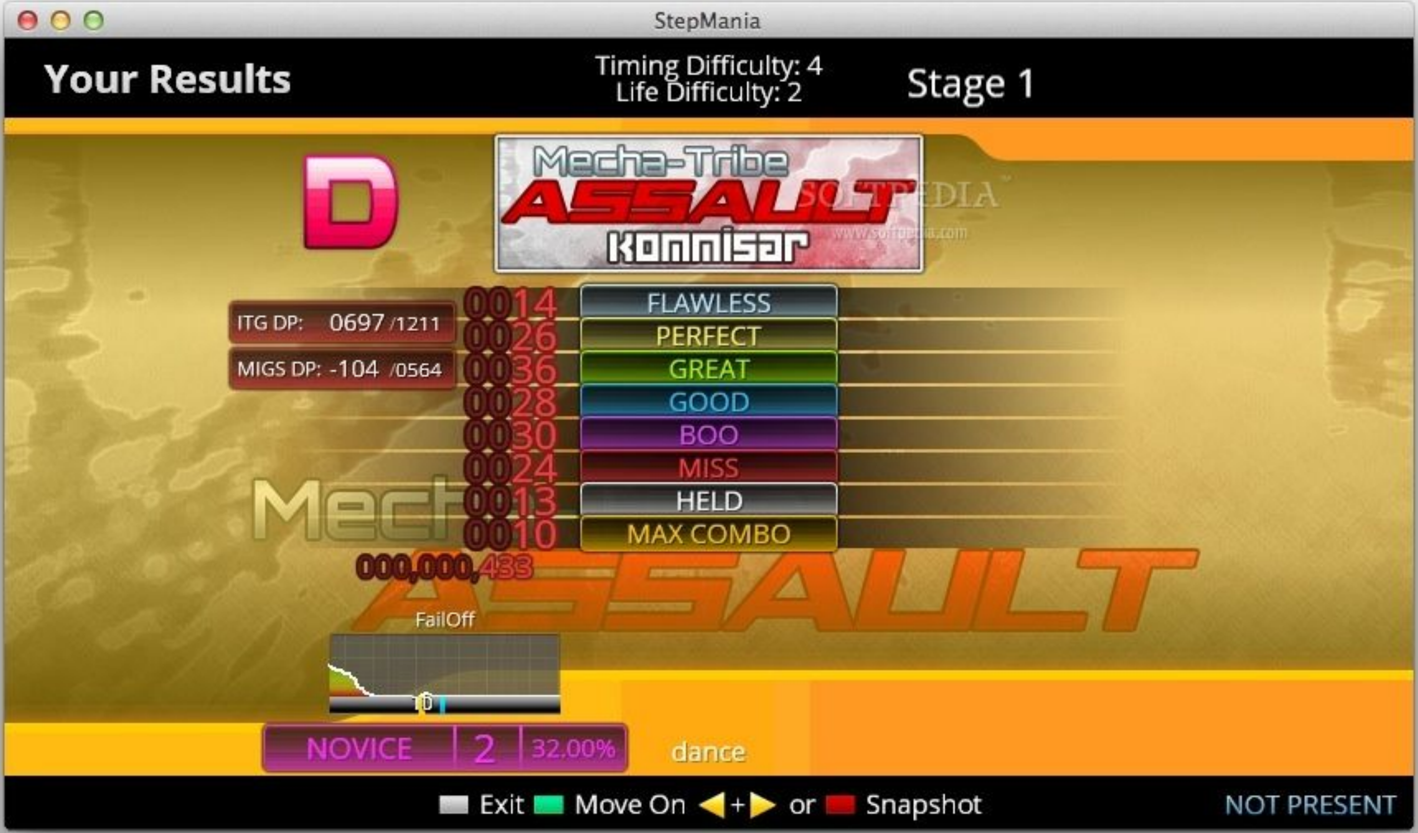
@hola_soy_milk



@hola_soy_milk

@hola_soy_milk

@hola_soy_milk



StepMania

Your Results

Timing Difficulty: 4
Life Difficulty: 2

Stage 1

D

Mecha-Tribe ASSAULT
Kommisar

ITG DP: 0697 /1211
MIGS DP: -104 /0564

0014 FLAWLESS
0026 PERFECT
0036 GREAT
0028 GOOD
0030 BOO
0024 MISS
0013 HELD
0010 MAX COMBO

000,000,433

FailOff

NOVICE | 2 | 32.00%

dance

Exit Move On ◀+▶ or Snapshot

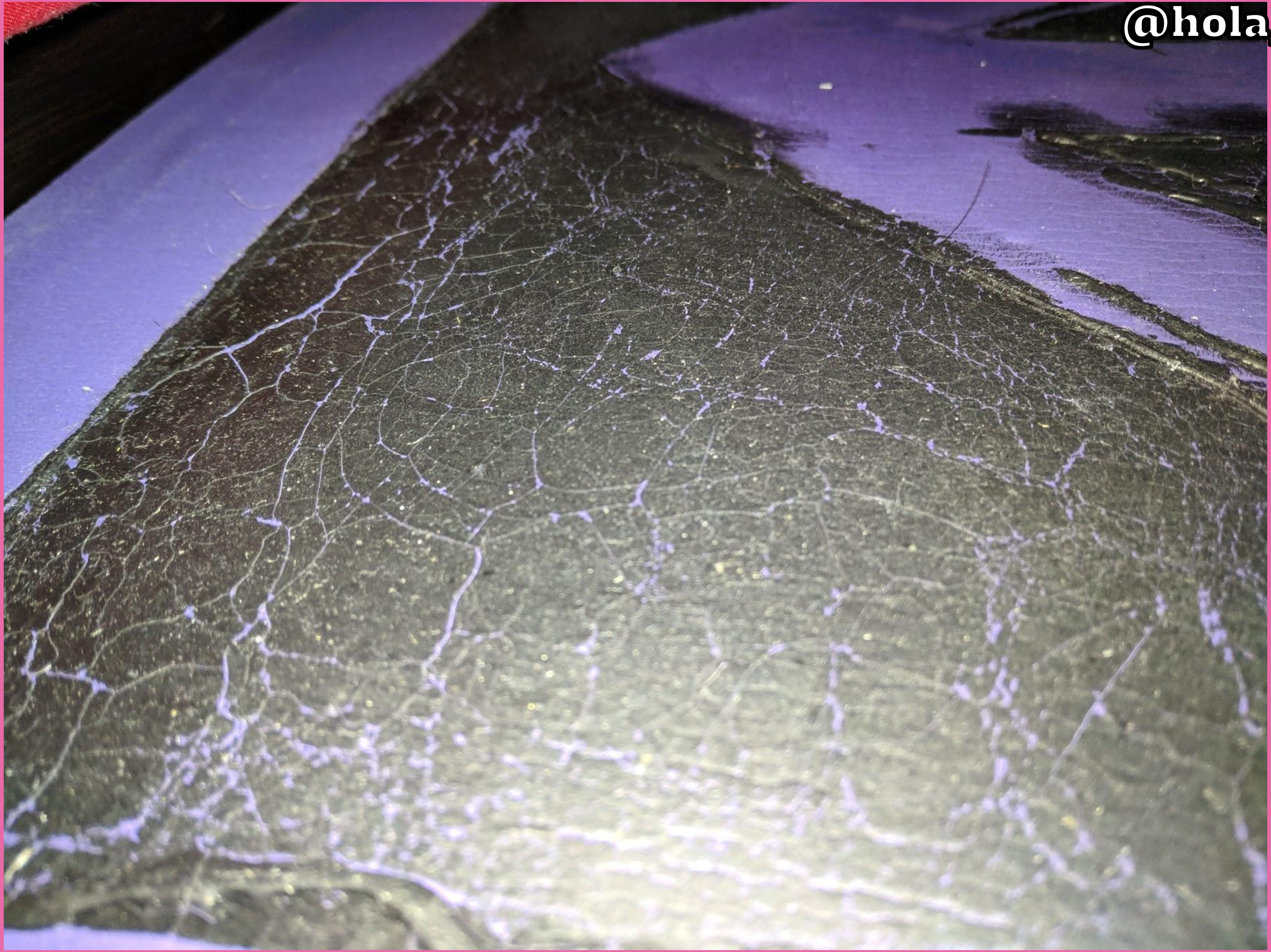
NOT PRESENT

¿O sea ya terminaron de por vida no?

@hola_soy_milk

¿O sea ya terminaron de por vida no?

¯_(ツ)_/¯

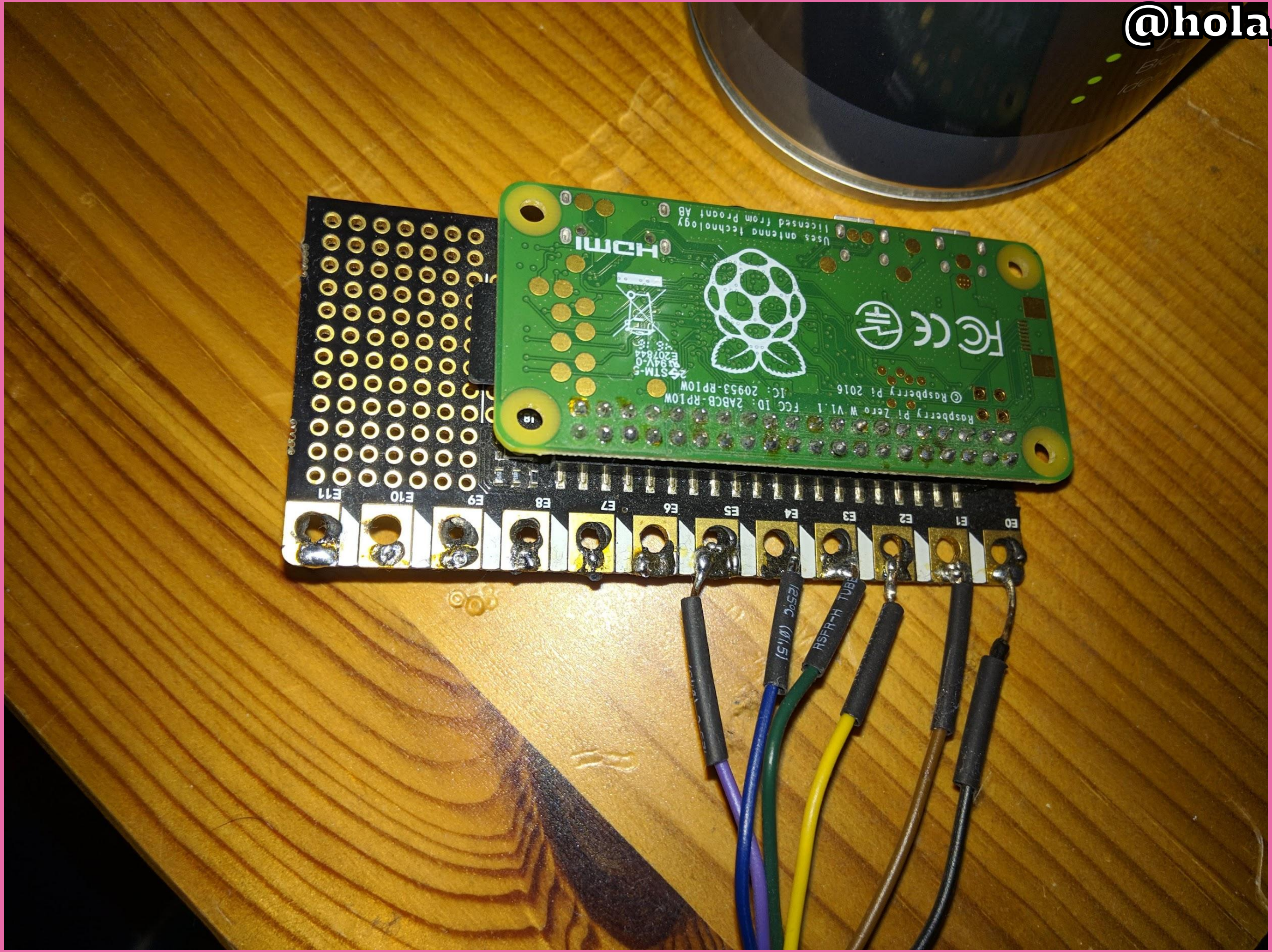


@hola_soy_milk

Si pudiéramos empezar de nuevo...

- Usar un Arduino
- Superficie más suave
- Cables físicos en vez de “cables” de pintura

@hola_soy_milk

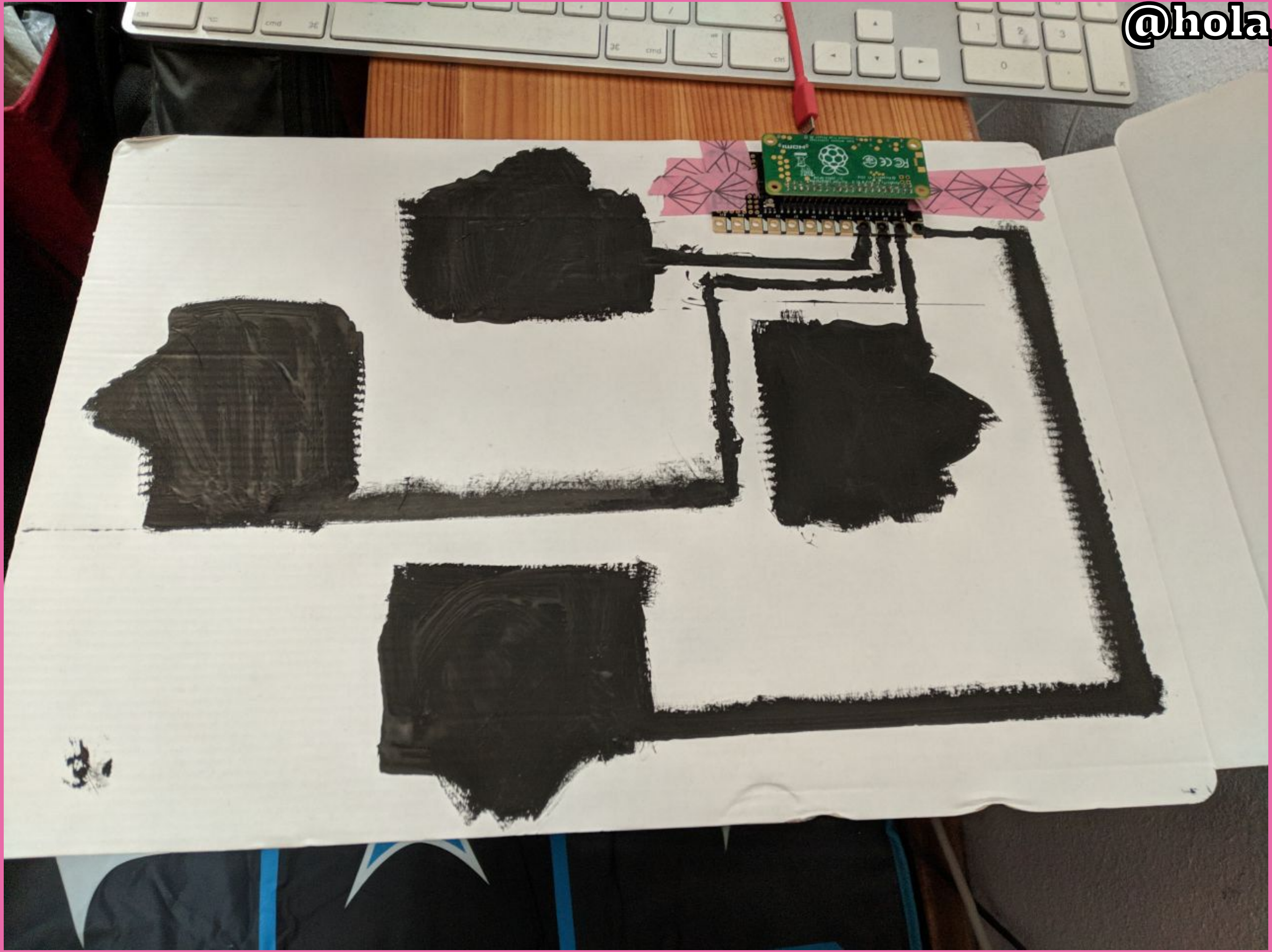

HDMI
Uses antenna technology licensed from Proant AB
Raspberry Pi Zero W V1.1 FCC ID: 2ABCB-RPI0W
IC: 20953-RPI0W
© Raspberry Pi 2016
E11
E10
E9
E8
E7
E6
E5
E4
E3
E2
E1
E0

@hola_soy_milk

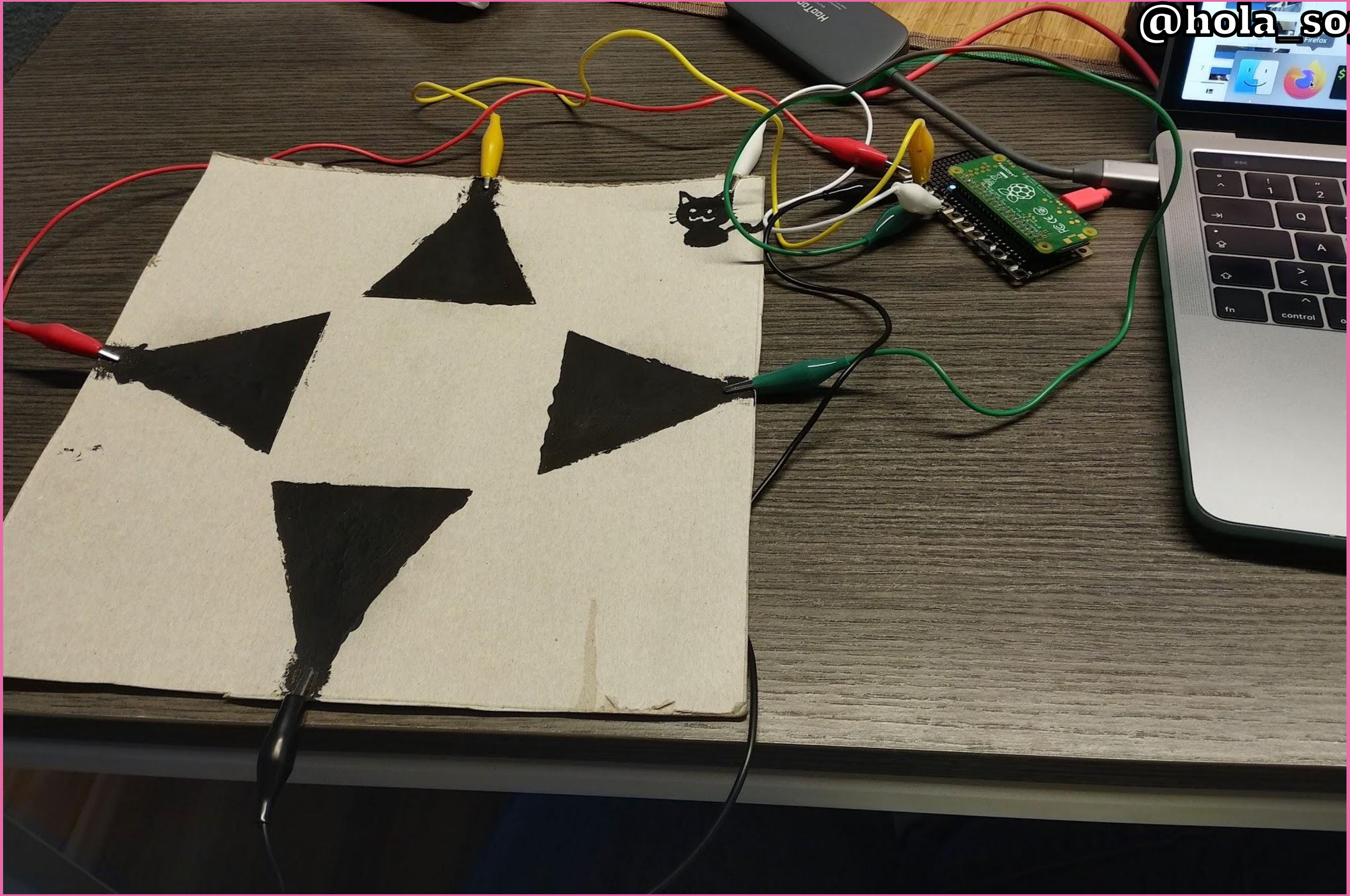
¡Pero bueno! Aprendimos un montonazo

@hola_soy_milk

Ya ya bueno a bailar, Ramón.



@hola_soy_milk



@hola_soy_milk

Original Wagner
Die Backfrische
im Original Wagner Steinofen vorgebacken
MOZZARELLA mit Kirschtomaten & Basilikumöl
LUFTIG-KROSSER BODEN
DURCH ZUGABE VON 24 STD. GEREIFTEM SAUERTEIG UND OLIVENÖL NATIV EXTRA
Backtradition seit 1973
ECHT WAGNER
Pro Portion 175g (1/2 Pizza)
pro 100g: 940 kJ/224 kcal

@hola_soy_milk

Y bueno, al final...

- Hackear hardware suena intimidante, pero hay mucho apoyo
- ¡No temas experimentar con cosas nuevas!
- Investiga los paquetes existentes
- Sin embargo, lo más importante...

@hola_soy_milk

¡Pásala Bien!

¡Links!

@hola_soy_milk

- Dance-mat.js - https://github.com/hola-soy-milk/picap-dance-mat
- Pintura eléctrica - https://www.bareconductive.com/products/electric-paint
- Picap - https://www.bareconductive.com/collections/pi-cap
- Raspberry Pi Zero - https://www.raspberrypi.com/products/raspberry-pi-zero-w/
- Stepmania: https://www.stepmania.com/
- Linux USB Gadget - https://www.kernel.org/doc/Documentation/usb/gadget_configfs.txt
- Teclado Raspberry Pi USB - https://randomnerdtutorials.com/raspberry-pi-zero-usb-keyboard-hid/
- NPM: linux-device - https://www.npmjs.com/package/linux-device

Ramón Huidobro

¡Mil gracias!

hola_soy_milk

hola_soy_milk_