

From picoruby to Twitch

or the inspiration you get after bringing 500 packages
of Ahoj Brause to Japans Ruby Kaigi

Joschka Schulz

Who am I

- Joschka
- working at Penseo GmbH
- love Ruby <3



Who am I

- burn myself soldering iron
- drinking Ahoj Brause with a lot of people in Japan
- I like to combine hardware and software



The Inspiration



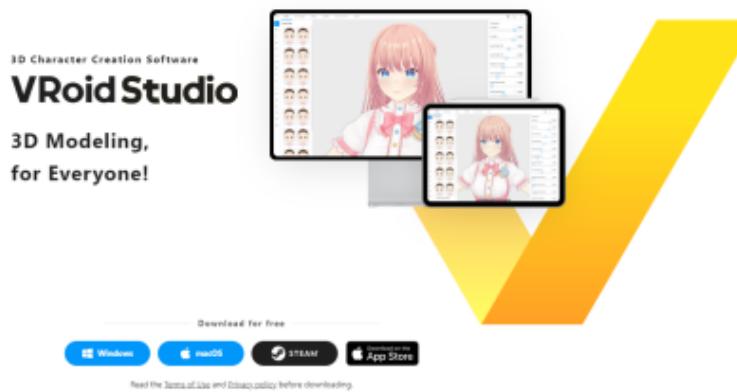
The Idea

- build a keyboard running ruby
- connect it somehow to something that can be animated
- use it as kind of Avatar on Twitch
- show how easy it is
- use as much ruby as possible



Excuse Vtuber/PNGtuber

- no need to be camera ready
- anonymity



VRoid Studio from pixiv

Lets go deeper in the rabbit whole

What has gone down the rabbit hole..



Never quite returns from the rabbit hole.

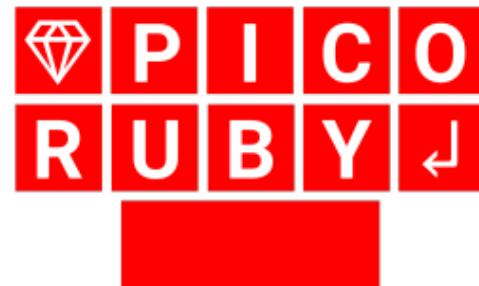
mruby

- for embed or linking in applications
- ruby 3.x syntax
- nice for rubists that like to solder



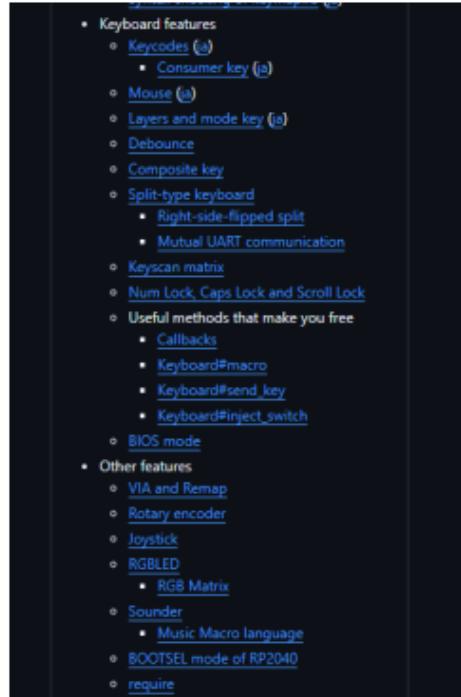
picrouby

- alternative mruby implementation
- aims for a perfect fit for raspberry pi pico
- even smaller then mruby



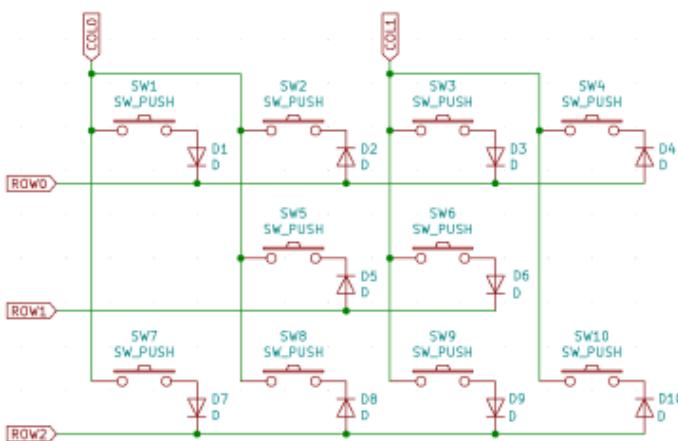
PRK

- keyboard firmware
- written in picoruby
- easy installation
- nice features



PRK settings

- 6 buttons
- direct scan
- really not that complicated



RP2040

- small single chip computer
- can run picoruby
- 30 pins
- use like a usb drive



How to install PRK on the RP2040

- holding BOOTSEL button pressed
- plug in
- drag and drop prk firmware on it

README greeting

PRK Firmware 0.9.23 (20231119 revision 0c8a999)

Welcome to PRK Firmware!

Usage:

- Drag and drop your `keymap.rb` into this directory
- Then, your keyboard will be automatically rebooted. That's all!

Notice:

- Make sure you always have a backup of your `keymap.rb` because upgrading prk_firmware-*.uf2 may remove it from flash

https://github.com/picoruby/prk_firmware

prk keymap

```
kbd = Keyboard.new

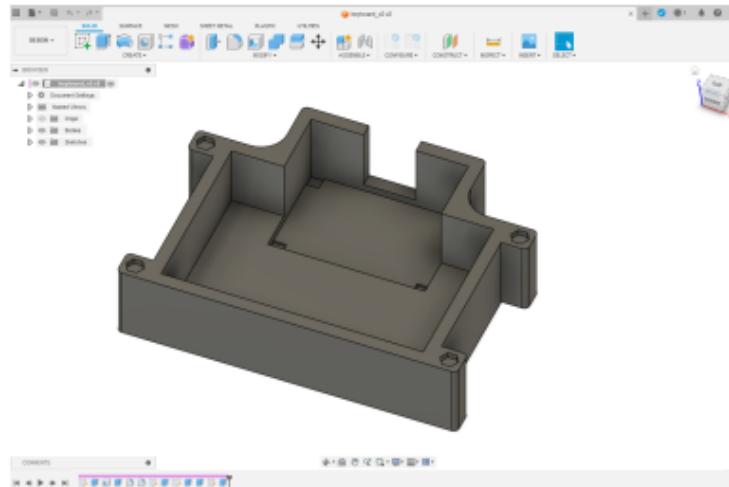
kbd.init_direct_pins(
  [
    3, 4, 5,
    0, 1, 2
  ]
)

kbd.add_layer :default, %i(
  KC_F13 KC_F14 KC_F15
  KC_F16 KC_F17 KC_F18
)

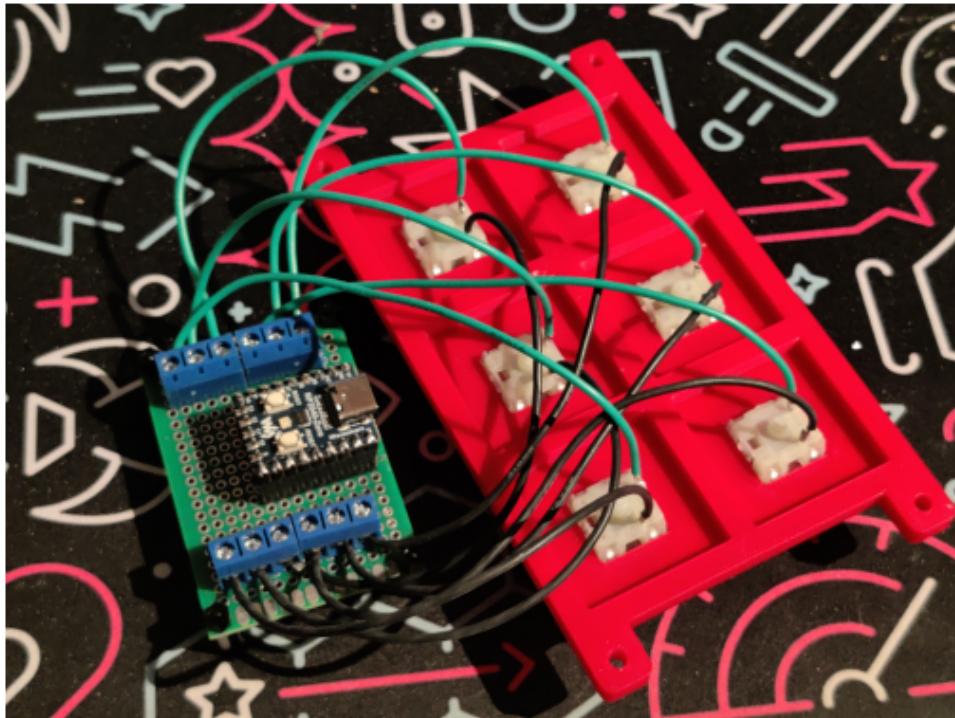
kbd.start!
```

The Keyboard Case

- 3d printed
- fits the controller
- and a lot of wires

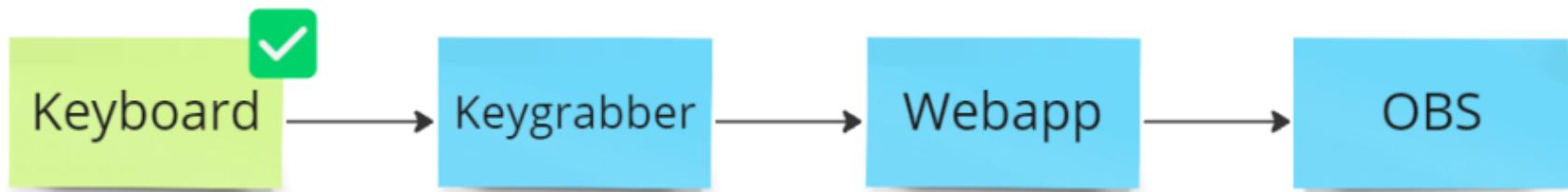


Lots of wires



what's next?

- running keyboard
- We need some key grabber



how to grab keys in windows

- get information about keypressed
- need to be in the background

first try

- lets search for gems
- most libraries just work in foreground

second try

- autohotkey the solution?
- ruby api
- could be the solution!
- or not?

autohotkey and games

- banned in some games, because of anti cheat
- sadly exactly the game I want to play



third try

- do I really need that all?
- ffi should be enough
- just on windows

key grabber

- lets look inside the windows api

GetAsyncKeyState function (winuser.h)

Article • 08/04/2022

Feedback

In this article

- Syntax
- Parameters
- Return value
- Remarks

Show 3 more

Determines whether a key is up or down at the time the function is called, and whether the key was pressed after a previous call to [GetAsyncKeyState](#).

Syntax

C++

```
SHORT GetAsyncKeyState(
    [in] int vKey
);
```

Copy

key grabber

```
require 'ffi'

class KeyStateChecker
  extend FFI::Library
  ffi_lib :user32

  attach_function :GetAsyncKeyState, [:int], :short

  def f13_pressed?
    get_key_state(0x7C) != 0
  end
end
```

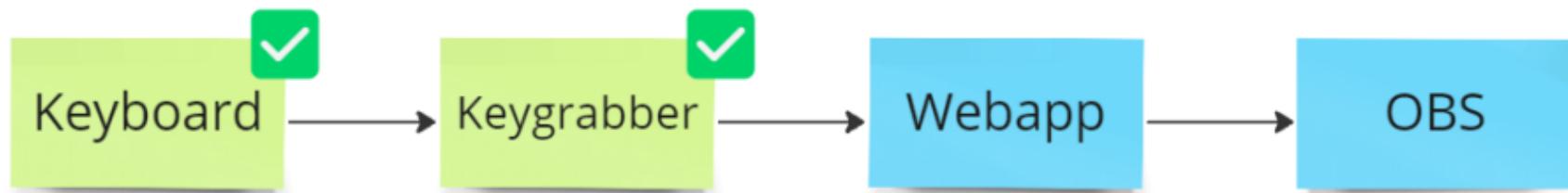
key grabber fire webhooks

- idea is to have a rails app that accept webhooks
- notify webhook on key press

```
if key_state_checker.f13_pressed?  
  puts "F13 was pressed"  
  HTTParty.get('http://localhost:3000/characters/1/move_head')  
end
```

what's next?

- a rails application
- accepting webhooks for animations



webapp

- start a new rails application
- I think that everyone here has done it already

webapp controller

- scaffold the webhook controller in rails

Avatar

- long long time ago
- Lila Laune Bär



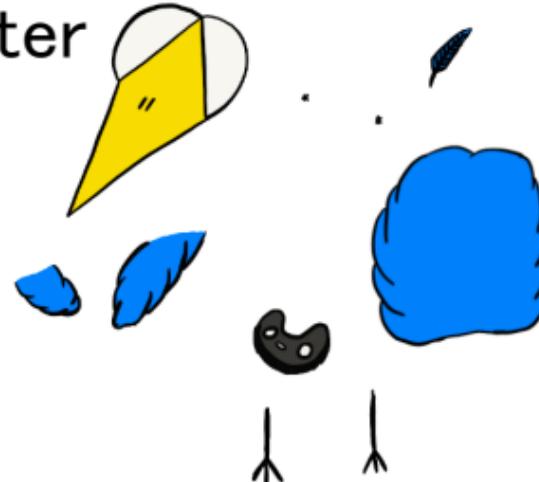
Bird of Metty

- Metty drawings
- my kid drawings



Bird of Joschka

- lot of small pieces
- used for animation later
- do you see the similarities?



Stuck together

- put parts with css together
- add css animation for movement



Idea of animation

- attach css classes via webhook
- controller already exists
- include action cable

controller – move_head endpoint

```
def move_head
  message = { action: 'move_head' }
  ActionCable.server.broadcast("avatar_channel", message.to_json)
  head :ok
end
```

javascript

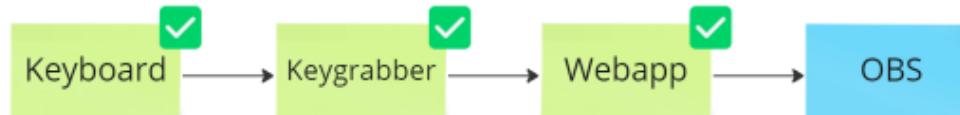
```
function addHeadShakeAnimation() {  
    addClassToElement('avatar-head', 'avatar-head-animation-shake');  
    setTimeout(function() {  
        removeClassFromElement('avatar-head', 'avatar-head-animation-shake');  
    }, 6000);  
}
```

css animation

```
.avatar-head-animation-shake {  
    animation: 3s infinite shake-head alternate;  
}  
  
@keyframes shake-head {  
    80%,  
    100% {  
        transform: rotate(5deg);  
    }  
}
```

from ruby hardware to animation in browser

- what do we have now
- a keyboard running picoruby
- a key grabber written in ruby
- a web application using rails
- sadly the next step won't include ruby



get the bird in the stream

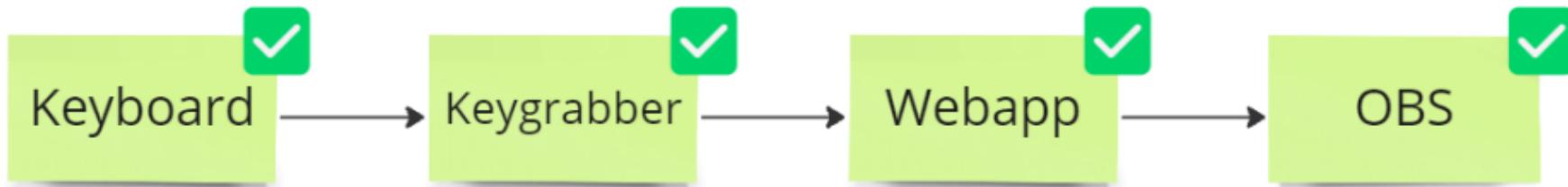
- obs
- used by a lot of streamers
- could also be used in video calls ;)

bird in obs

- just create a browser source
- put the url to the avatar inside
- crop it to size

final result

- bird in stream
- controlled by ruby



The End

- Thank you for listing