ON A KEYBOARD RUBY ON BOARD

PICORUBY AND PRK FIRMWARE

HASUMIKIN

RUBYCONF 2021 / NOV 8-10

Wake up, Ruby...

The Micon has you...

Follow the white rabbit.

Hi!

PicoRuby
and
PRK Firmware

PicoRuby
is a Ruby interpreter
for Microcontorollers

PRK Firmware
is a keyboard firmware
framework in PicoRuby

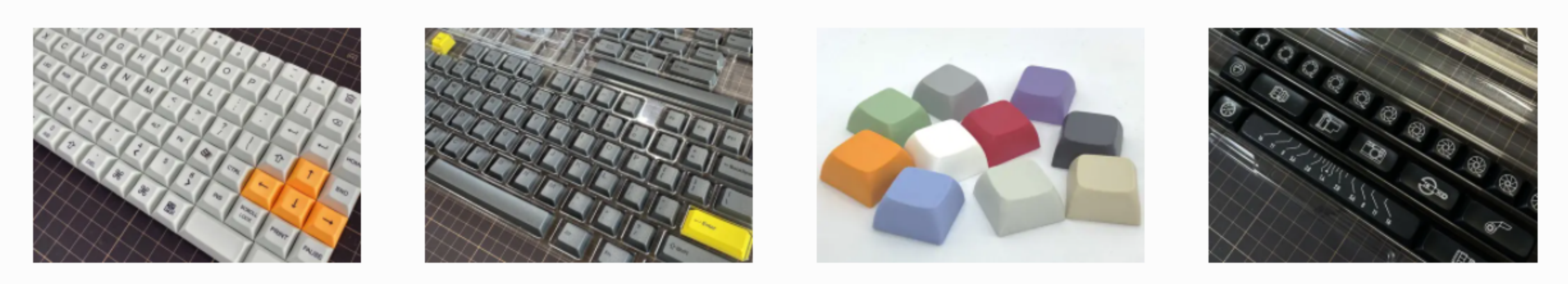
PRK Firmware

Hardware became familiar

- You can design your PCB even on a web browser
- PCB manufacturers are willing to accept your PCB data via the internet
- You can receive your PCB or even PCBA (assembled PCB) that arrives from overseas

When you build a keyboard

- You can choose
  - Keycaps for appearance and touch



https://talpkeyboard.net/?category_id=59be183f428f2d49120007b1

When you build a keyboard

- You can choose
  - Keycaps for appearance and touch
  - Switches for touch

https://talpkeyboard.net/?category_id=59cf8860ed05e668db003f5d

When you build a keyboard

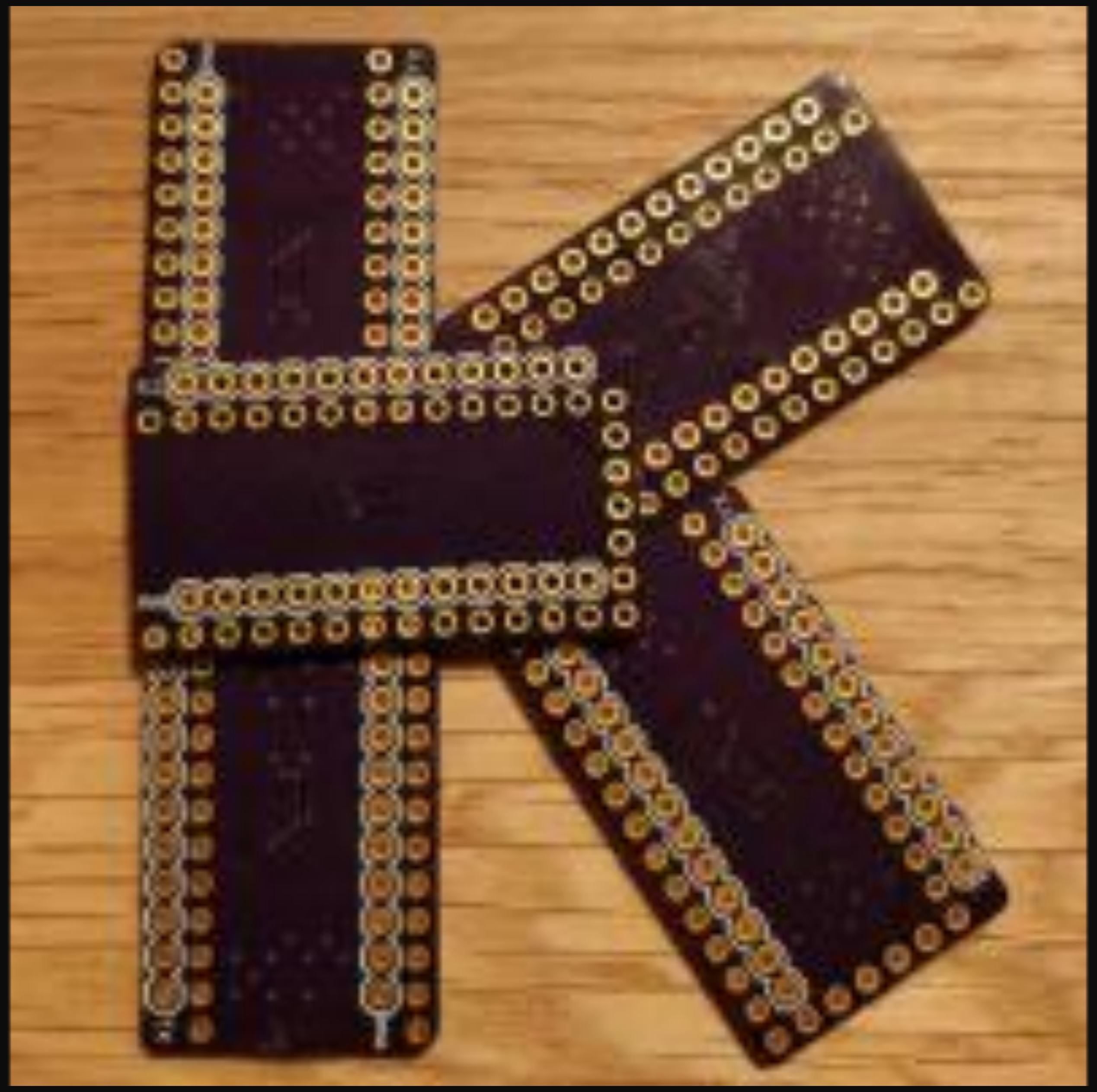
- You can choose
  - Keycaps for appearance and touch
  - Switches for touch
  - Layout for usability and appearance

When you build a keyboard

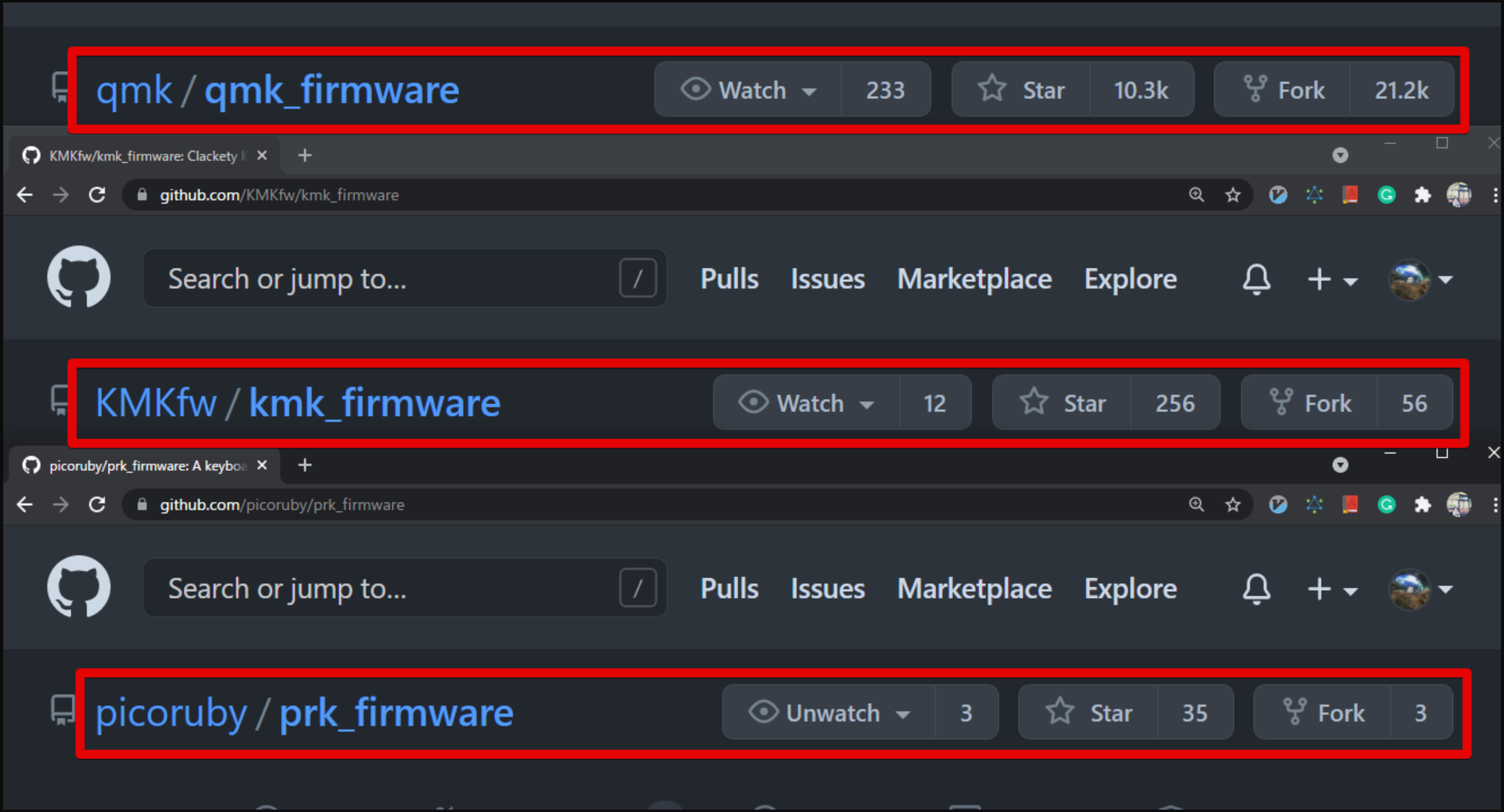
- You can choose
  - Keycaps for appearance and touch
  - Switches for touch
  - Layout for usability and appearance
  - Firmware for programmability

Keyboard firmware

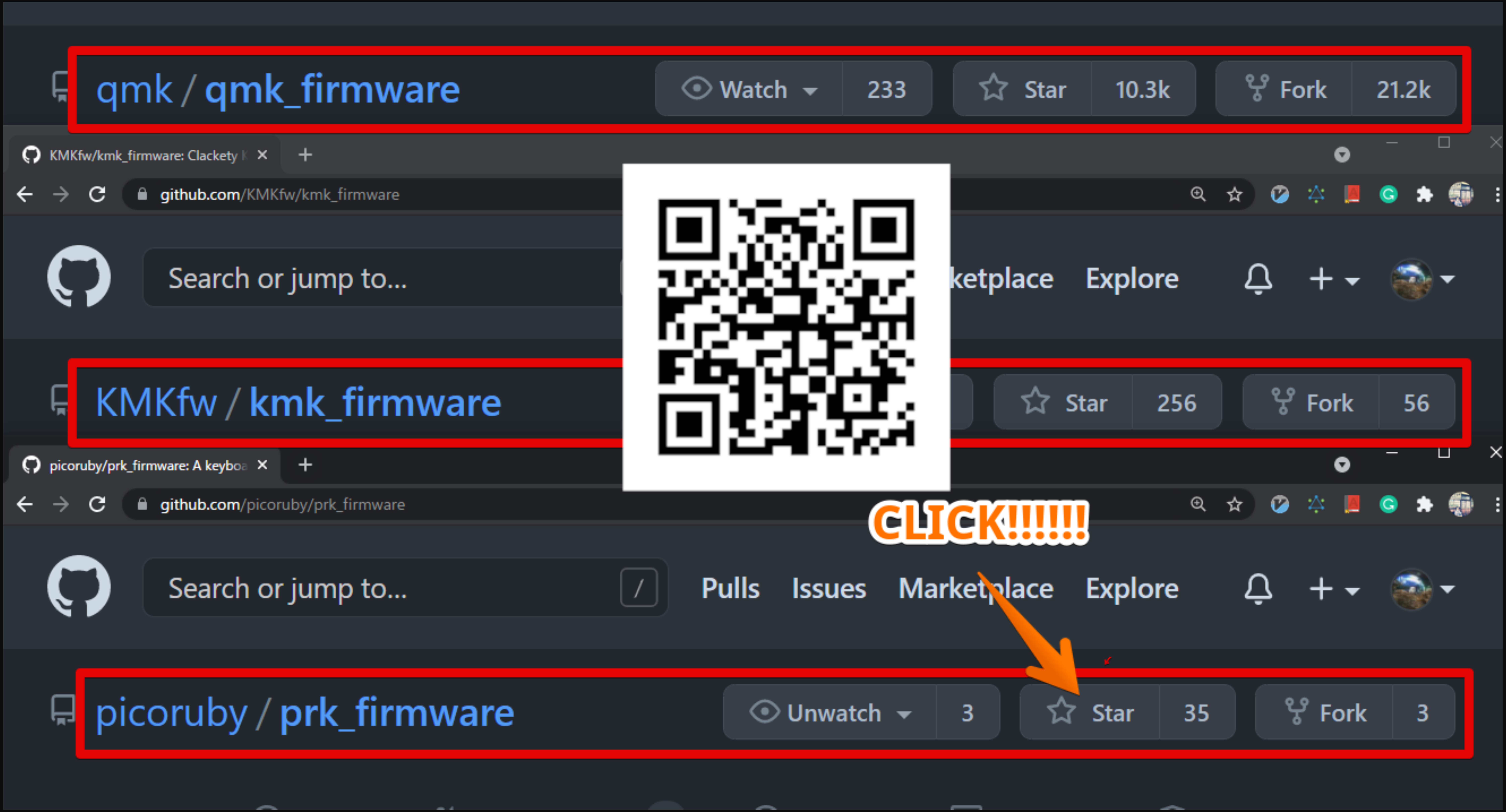
- QMK Firmware ... C
- KMK Firmware ... Python
- PRK Firmware ... **Ruby**



Keyboard firmware



Stargaze at picoruby/prk_firmware!



e.g.) picoruby/prk_pipigherkin

- Gherkin for the Raspberry Pi Pico
  - PCB available on talpkeyboard.net
  - Easy to explain but **hard for newbies** to build

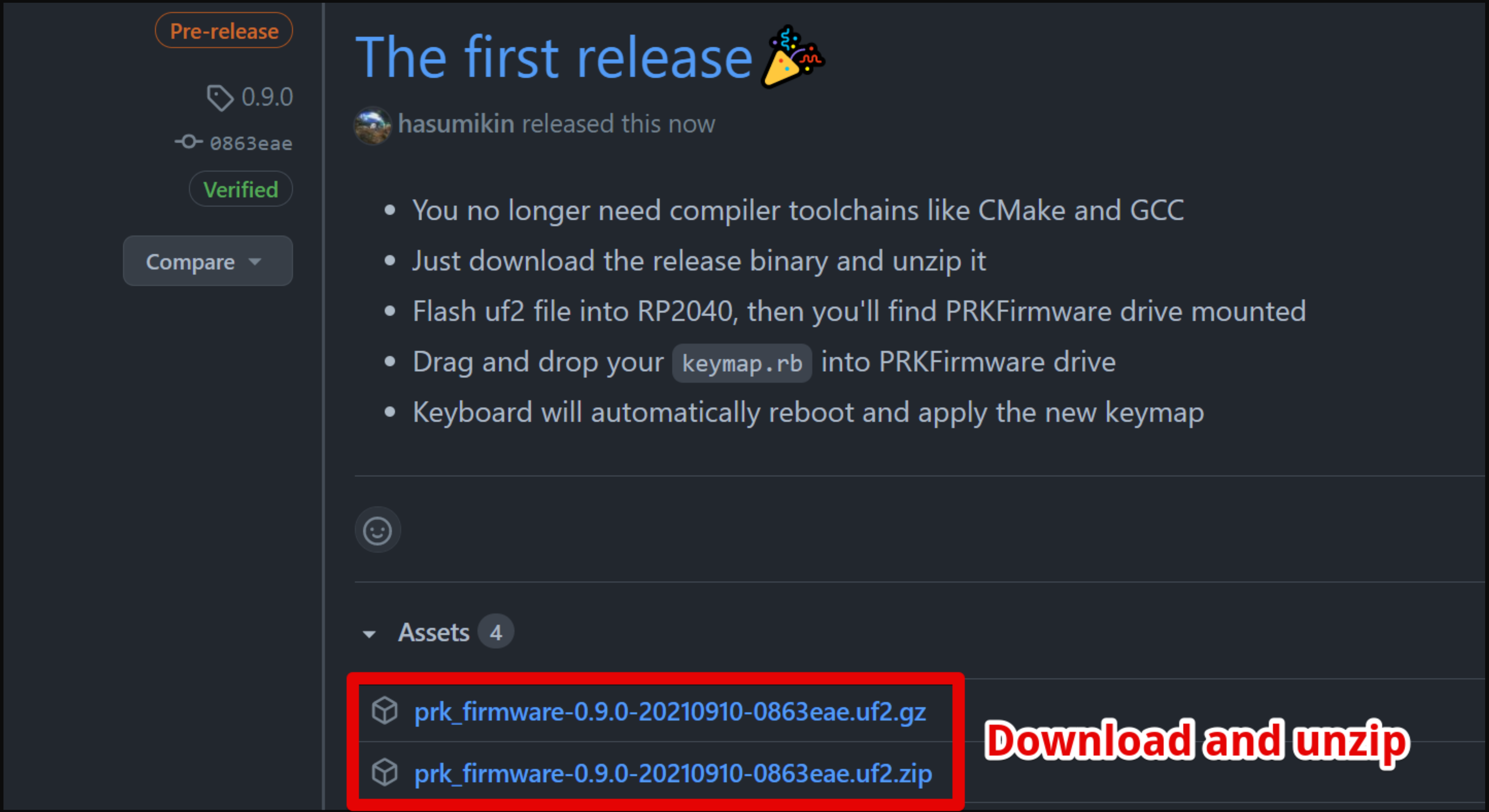
e.g.) prk_pipigherkin/keymap.rb

```ruby
# Initialize Keyboard
kbd = Keyboard.new
# Initialize GPIO pins
kbd.init_pins(
  [ 12, 11, 10, 9, 8 ], # row0, row1,... respectively
  [ 7, 6, 5, 4, 3, 2 ]  # col0, col1,... respectively
)
```

e.g.) prk_pipigherkin/keymap.rb

```ruby
# Default keymap
kbd.add_layer :default, %i(
  KC_Q      KC_W    KC_E        KC_R     KC_T       KC_Y       KC_U       KC_I      KC_O     KC_P
  KC_A      KC_S    KC_D        KC_F     KC_G       KC_H       KC_J       KC_K      KC_L     KC_BSPACE
  Z_LSFT    X_LGUI  C_LALT      V_LCTL   SPC_LOWER  ENT_RAISE  B_RCTL     N_RALT    M_RGUI   UNDS_RSFT
)
kbd.add_layer :raise, %i(
  KC_EXLM   KC_AT   KC_HASH     KC_DLR   KC_PERC    KC_CIRC    KC_AMPR    KC_ASTER  KC_EQUAL KC_PLUS
  KC_LABK   KC_LCBR KC_LBRACKET KC_LPRN  KC_MINUS   KC_LEFT    KC_DOWN    KC_UP     KC_RIGHT KC_BSPACE
  KC_RABK   KC_RCBR KC_RBRACKET KC_RPRN  ADJUST     ENT_RAISE  KC_BSLASH  KC_COMMA  KC_DOT   KC_SLASH
)
kbd.add_layer :lower, %i(
  KC_1      KC_2    KC_3        KC_4     KC_5       KC_6       KC_7       KC_8      KC_9     KC_0
  KC_TAB    KC_NO   KC_QUOTE    KC_DQUO  KC_MINUS   KC_GRAVE   KC_TILD    KC_PIPE   KC_COLON KC_SCOLON
  KC_ESCAPE KC_LGUI KC_LALT     KC_LCTL  SPC_LOWER  ADJUST     KC_RCTL    KC_RALT   KC_RGUI  KC_RSFT
)
# Mode keys
kbd.define_mode_key :Z_LSFT,    [ :KC_Z,     :KC_LSFT, 150, 150 ]
# ...
kbd.define_mode_key :UNDS_RSFT, [ :KC_UNDS,  :KC_RSFT, 200, 150 ]
kbd.define_mode_key :ENT_RAISE, [ :KC_ENTER, :raise,   150, 150 ]
kbd.define_mode_key :SPC_LOWER, [ :KC_SPACE, :lower,   300, 250 ]
```

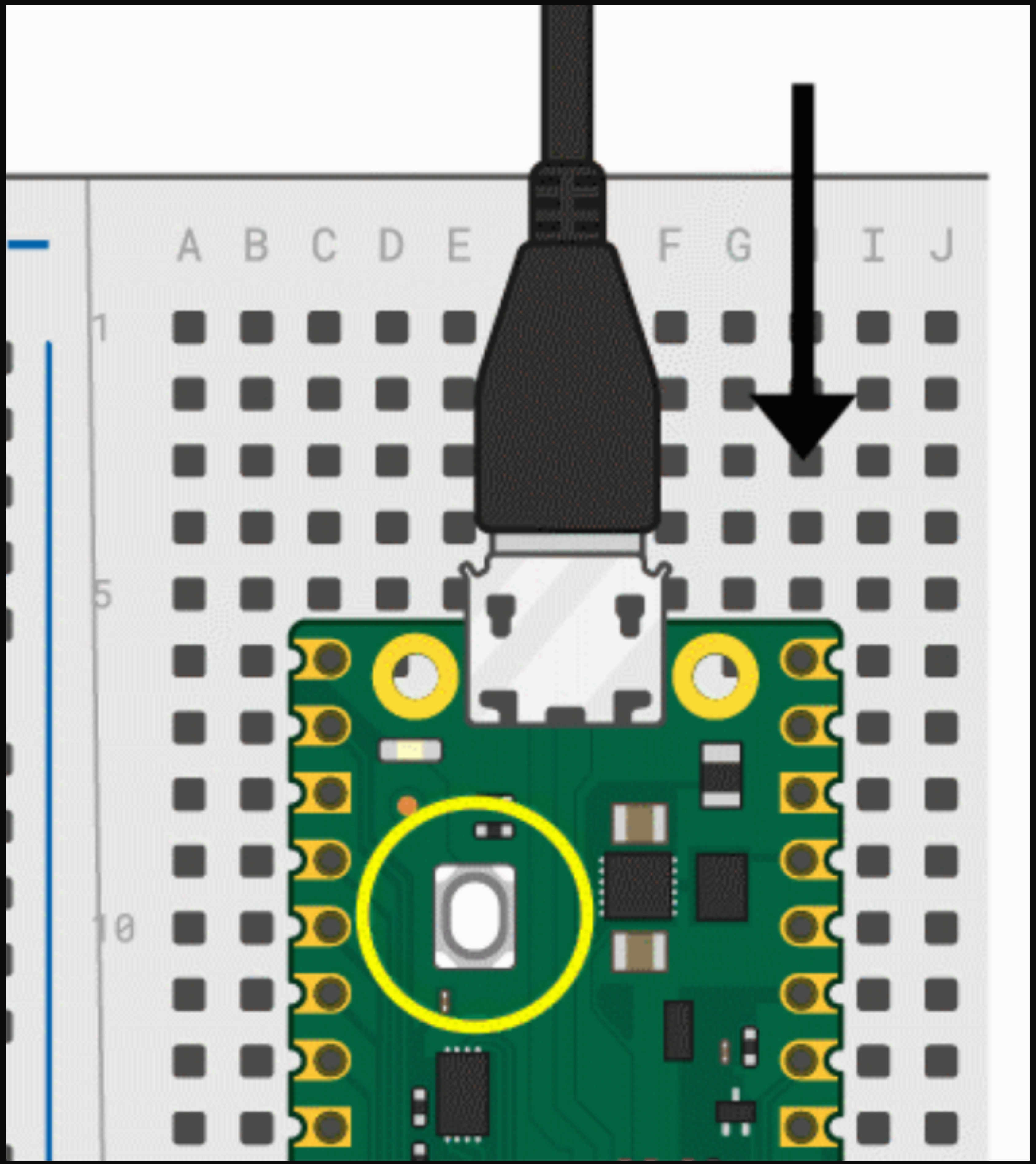
picoruby/prk_firmware/releases



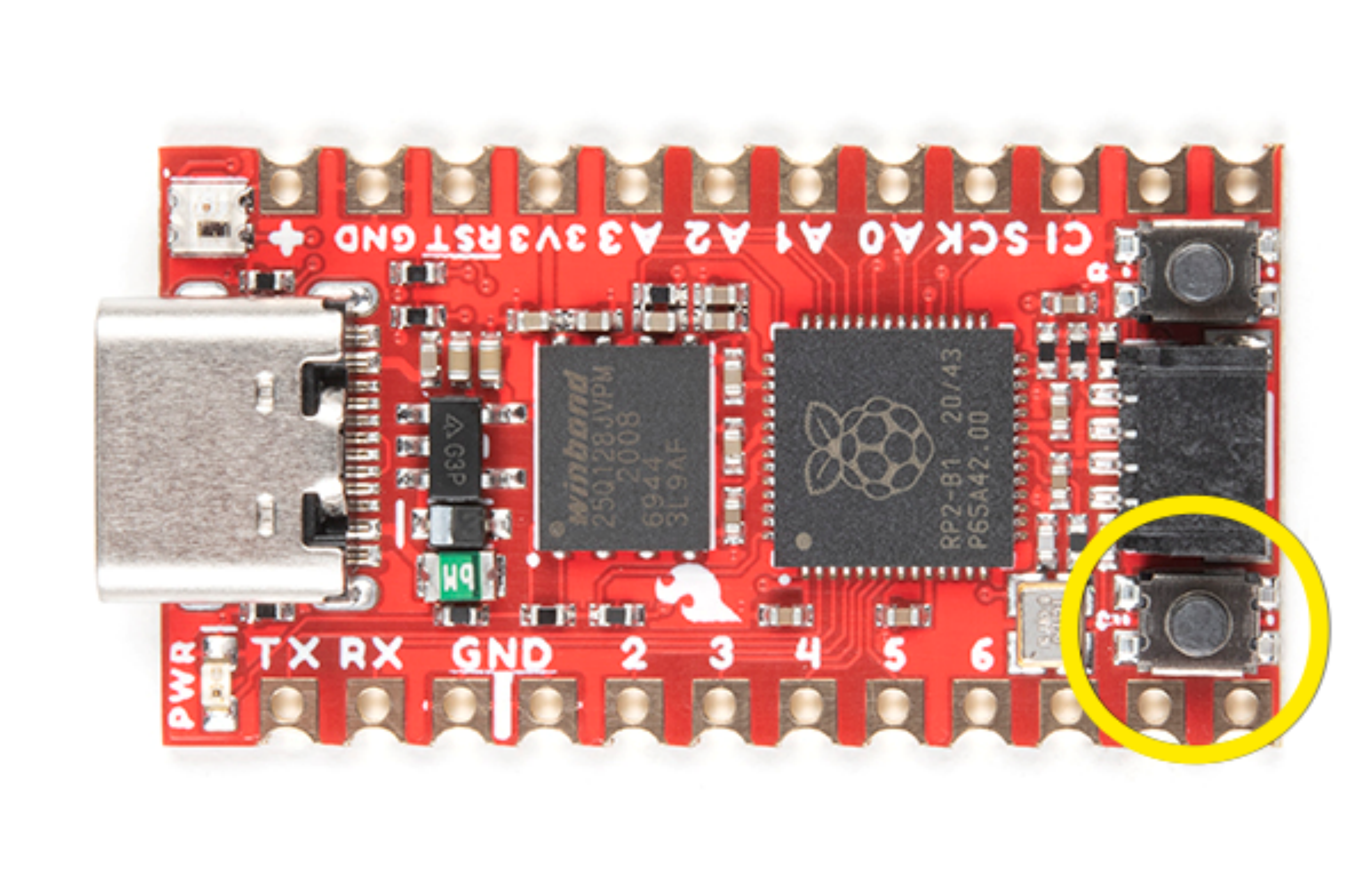
Download and unzip

Install into the microcontroller

- Connect USB cable while pressing "boot button"
- RPI-RP2 will be mounted as storage, then

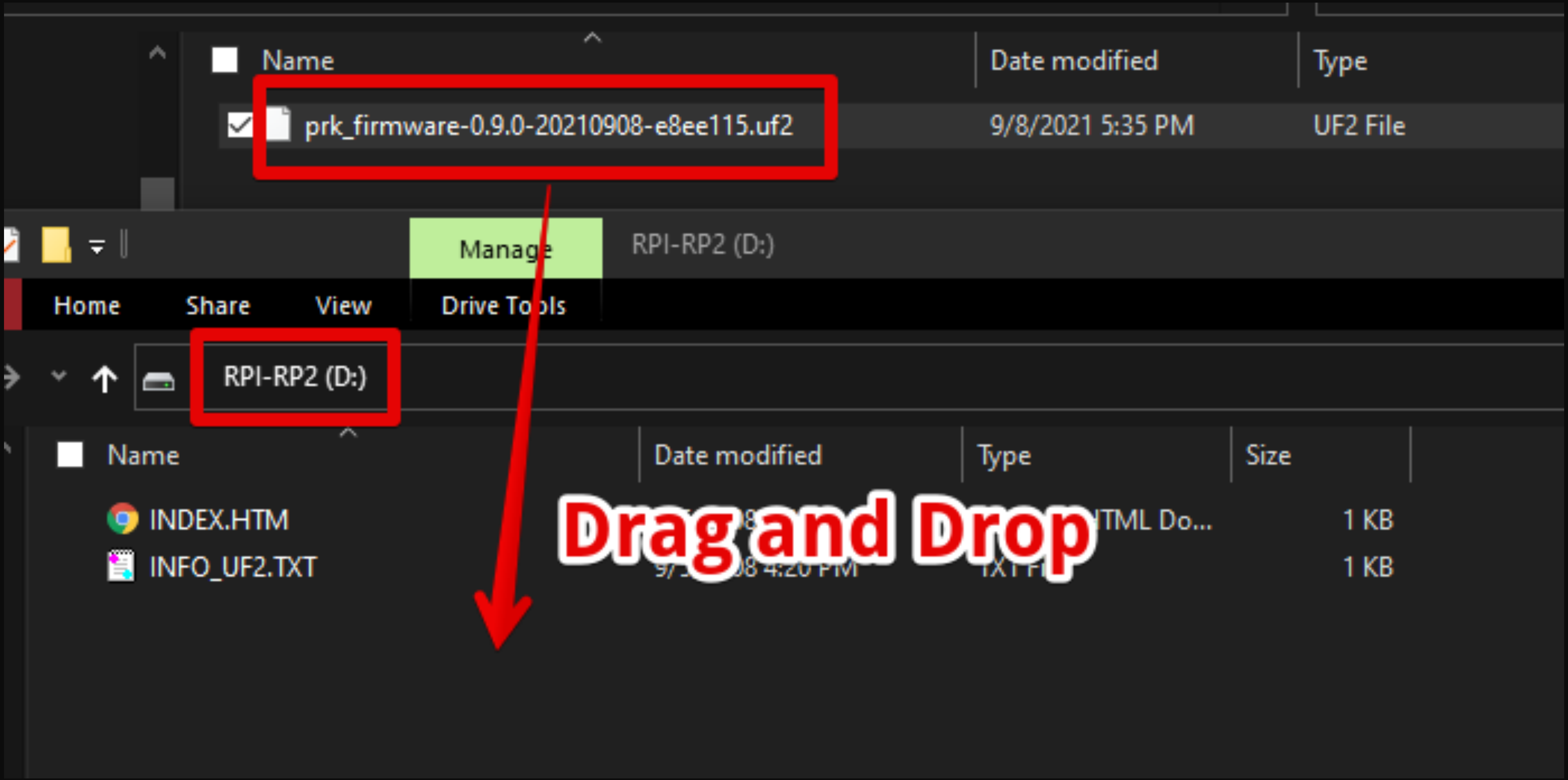


Install into the microcontroller

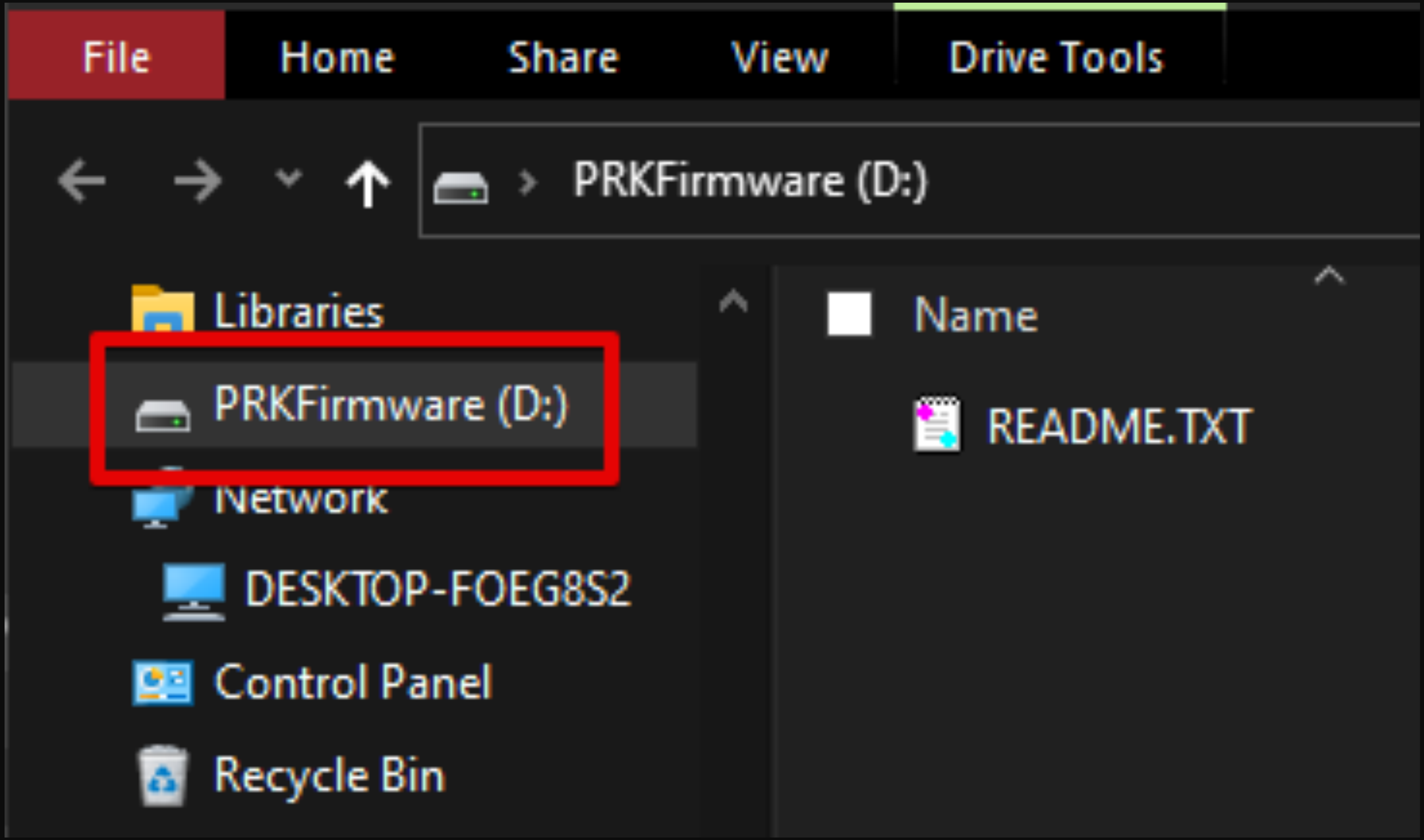


Boot button on Pro Micro RP2040

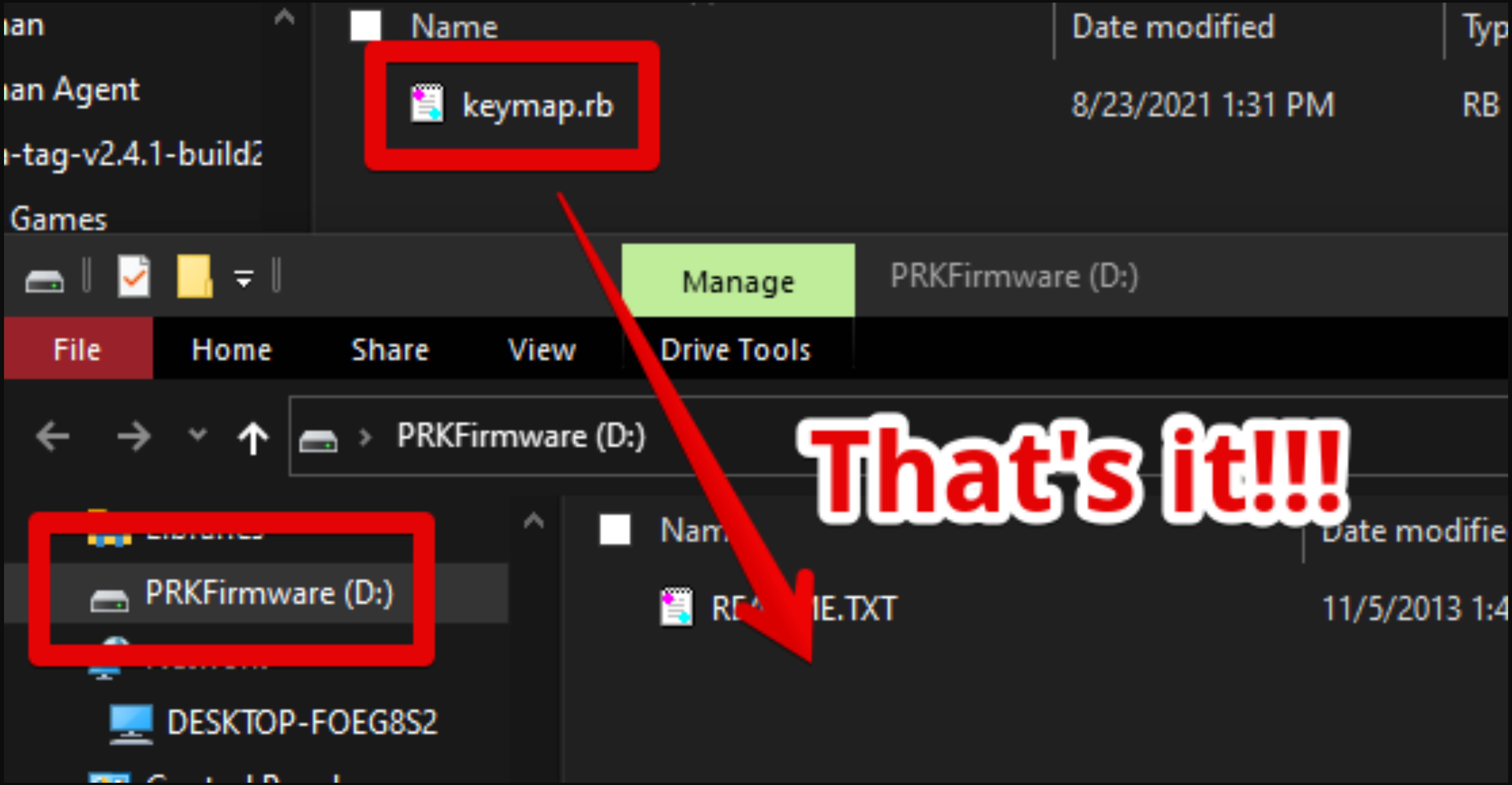
Install into the microcontroller



A new drive automatically mounted



Drag & drop keymap.rb

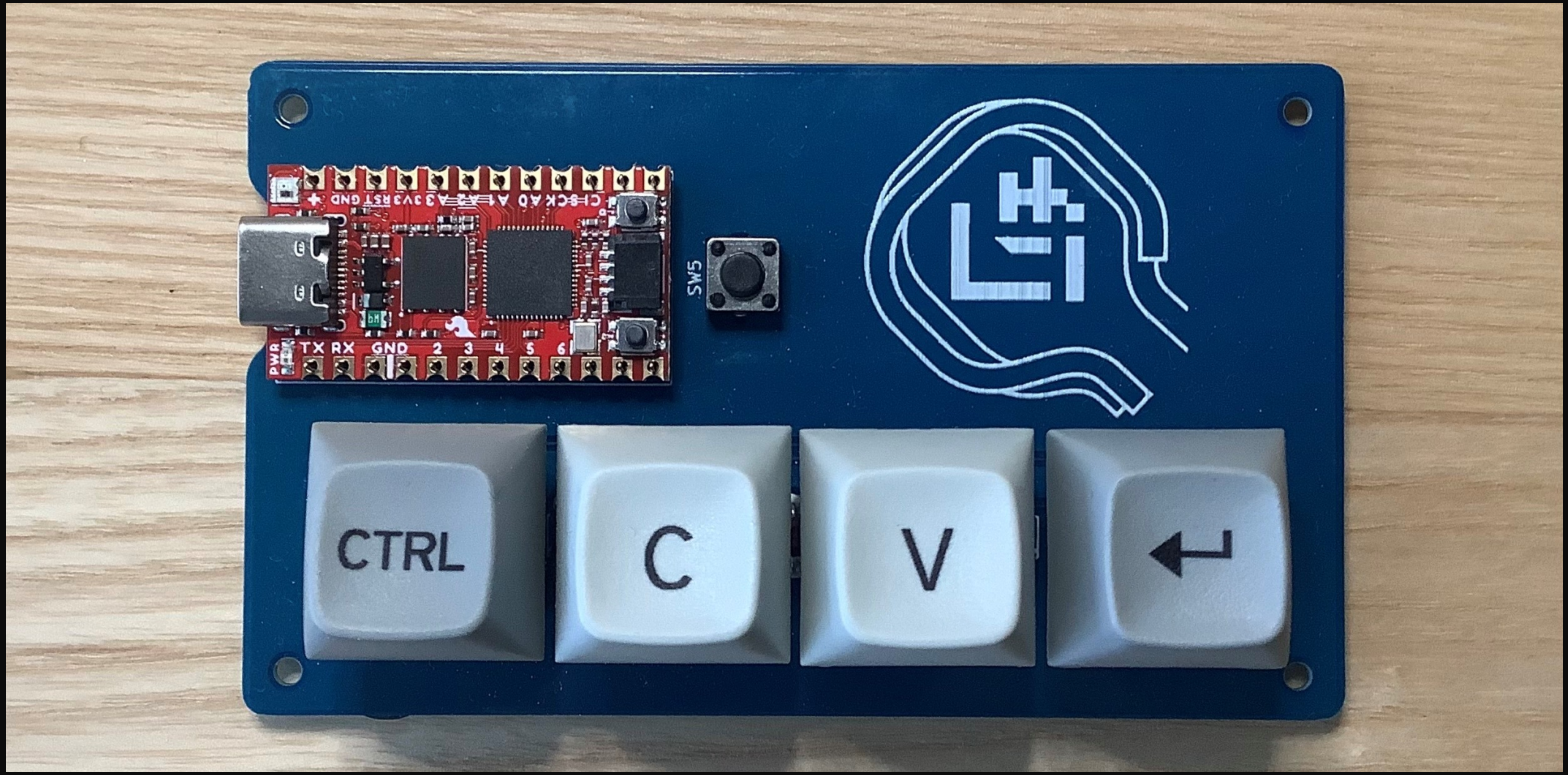


THE

KEYMAP RELOADED

Unlike QMK Firmware,
You don't need
to **compile**
anything!

Demonstrations



- meishi keypad
  - meishi means "name card"
  - Four keys macro pad

Fibonacci number
(DEMO)

Fibonacci in keymap.rb

```ruby
class Fibonacci
  def initialize
    @a = 0 ; @b = 1
  end
  # Omits F0 and F1
  def take
    result = @a + @b
    @a = @b
    @b = result
  end
end
fibonacci = Fibonacci.new
kbd.define_mode_key :FIBONACCI,
                    [ Proc.new { kbd.macro fibonacci.take },
                    :KC_NO, 300, nil ]
```

Password generator (DEMO)

Password generator in keymap.rb

```ruby
class Password
  def initialize(c = nil)
    @c = c || 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_!@#$%^&*()=-+/[]{}<>'
  end
  def generate
    unless @srand
      srand(board_millis) # generate rand()'s seed by board_millis
      @srand = true
    end
    password = ""
    while true
      i = rand % 100
      password << @c[i].to_s
      break if password.length == 8
    end
    return password
  end
end
passwd = Password.new
kbd.define_mode_key :PASSWD,
                    [ Proc.new { kbd.macro passwd.generate, [] },
                      :KC_NO, 300, nil ]
```

Use it at your own risk

Extend your keyboard

If you wanna input ";" with SHIFT and ":" without SHIFT,

```
kbd.before_report do
  kbd.invert_sft if kbd.keys_include?(:KC_SCOLON)
end
```

You no longer need to modify .vimrc

Extend your keyboard

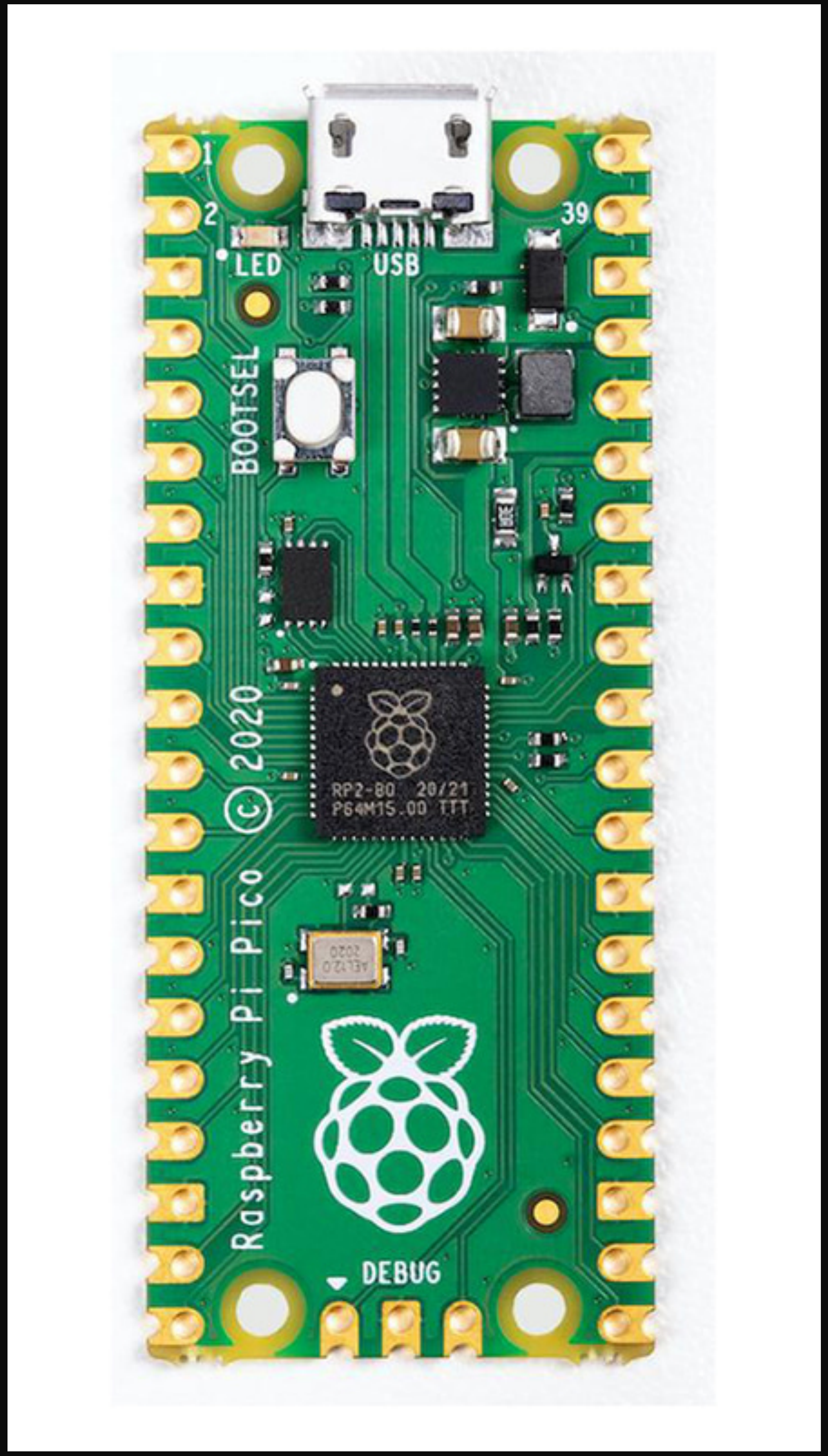
```ruby
class Keyboard
  def invert_ctl
    #       KC_LCTL: 0b00000001 👈
    #       KC_LSFT: 0b00000010
    #       KC_LALT: 0b00000100
    #       KC_LGUI: 0b00001000
    #       KC_RCTL: 0b00010000 👈
    #       KC_RSFT: 0b00100000
    #       KC_RALT: 0b01000000
    #       KC_RGUI: 0b10000000
    if (@modifier & 0b00010001) > 0
       @modifier &= 0b11101110
    else
       @modifier |= 0b00000001
    end
  end
end
kbd.before_report do
  kbd.invert_ctl if kbd.keys_include?(:KC_V)
end
```

Ruby-mode key
(DEMO)

THE KEYBOARD REVOLUTIONS

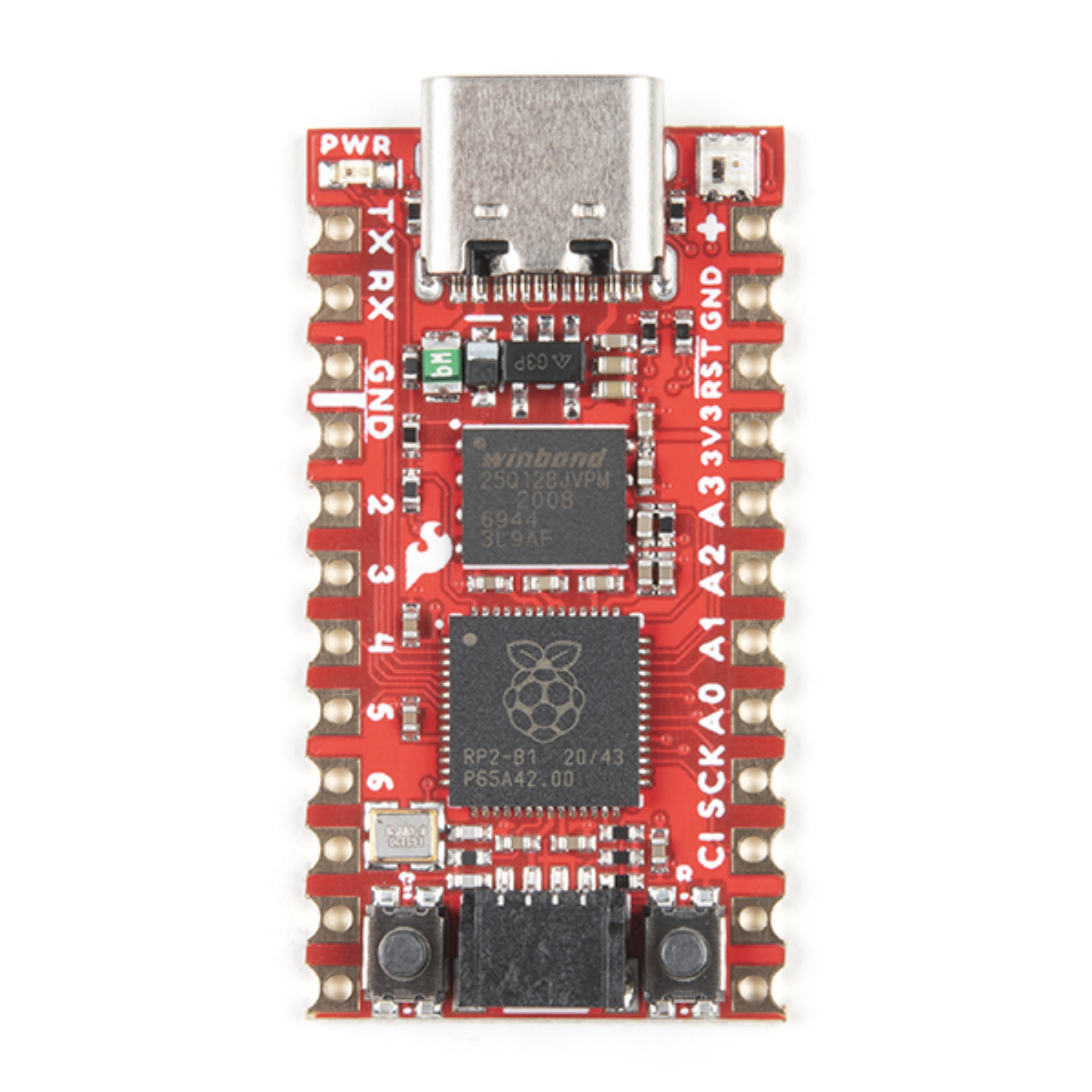
Micon of PRK Firmware

- Target MCU is "RP2040"
  - Raspberry Silicon
  - Implemented on Raspberry Pi Pico
  - 264KB RAM
  - Arm 32bit Cortex-M0+ (dual)



Micon of PRK Firmware

- RP2040 assembled on Pro Micro
  SPARKFUN PRO MICRO - RP2040
  [model no. DEV-18288]



https://www.sparkfun.com/products/18288

PicoRuby

https://ruby.or.jp/en

Ruby Association

News | About the Organization | Certification

Ruby Association > Past News > Call For Grant Proposals 2021

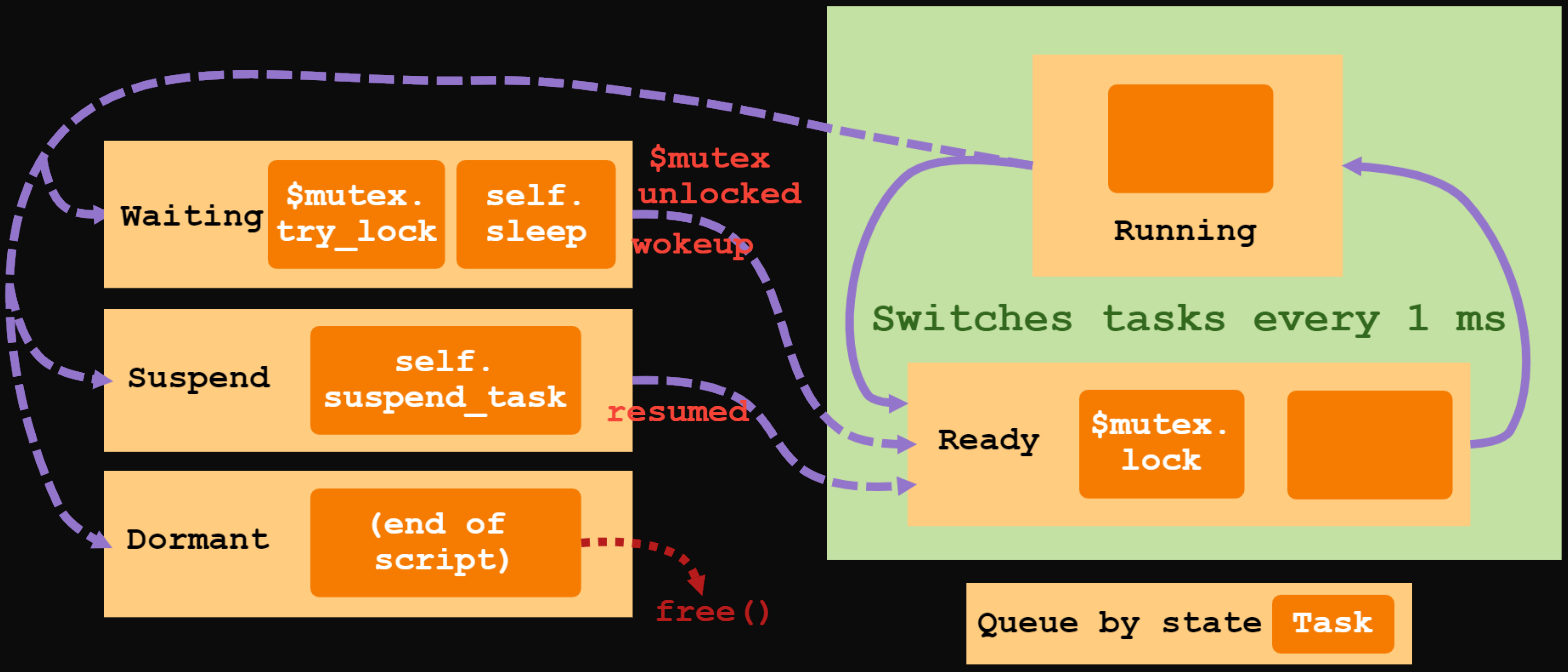
2021 Call for Grant Proposals

Ruby Association grant overview

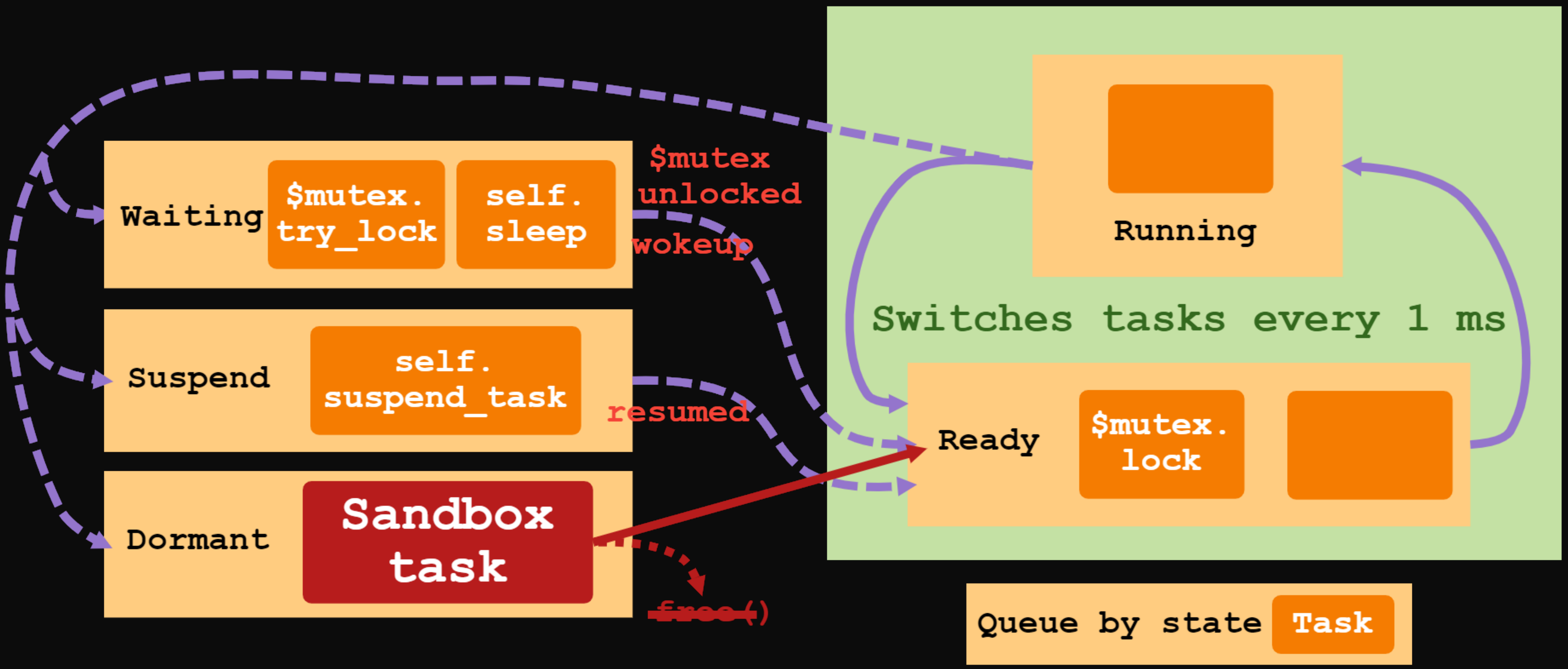
The Ruby Association is looking to give grants for development projects related to Ruby implementations, libraries and frameworks, including projects that improve existing software.

If you have a good idea, please submit a proposal. However, please note that we do not intend to fund development projects of Ruby applications for specific purposes because the aim of this grant is to improve Ruby and its environment.

Multitasking

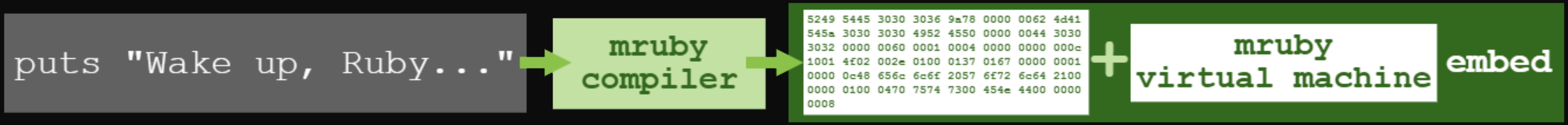


Multitasking



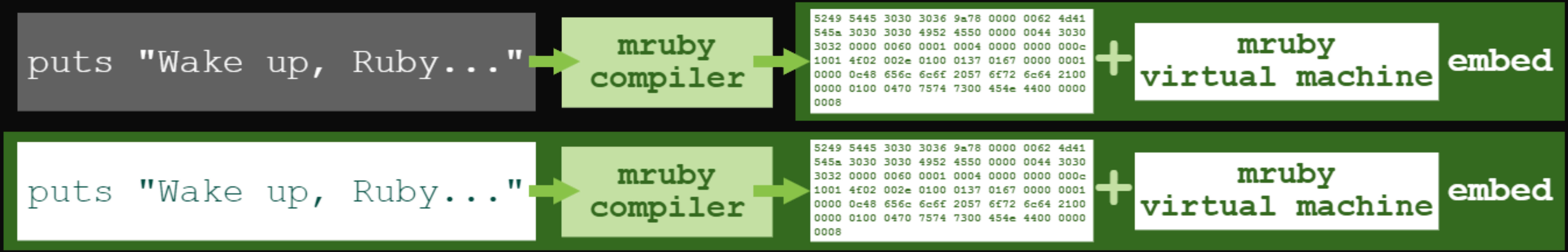
THE RUBY RESURRECTIONS

Architecture



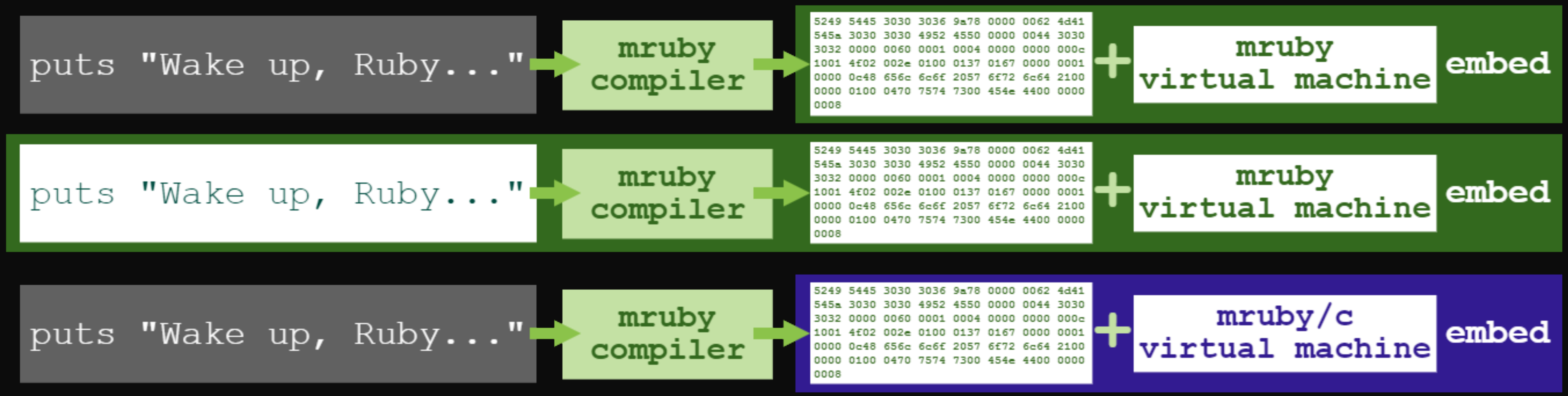
- An mruby app generally consists of **VM code** and **VM**
  - mruby-compiler has somehow big footprint
  - It isn't assumed to be embedded

Architecture



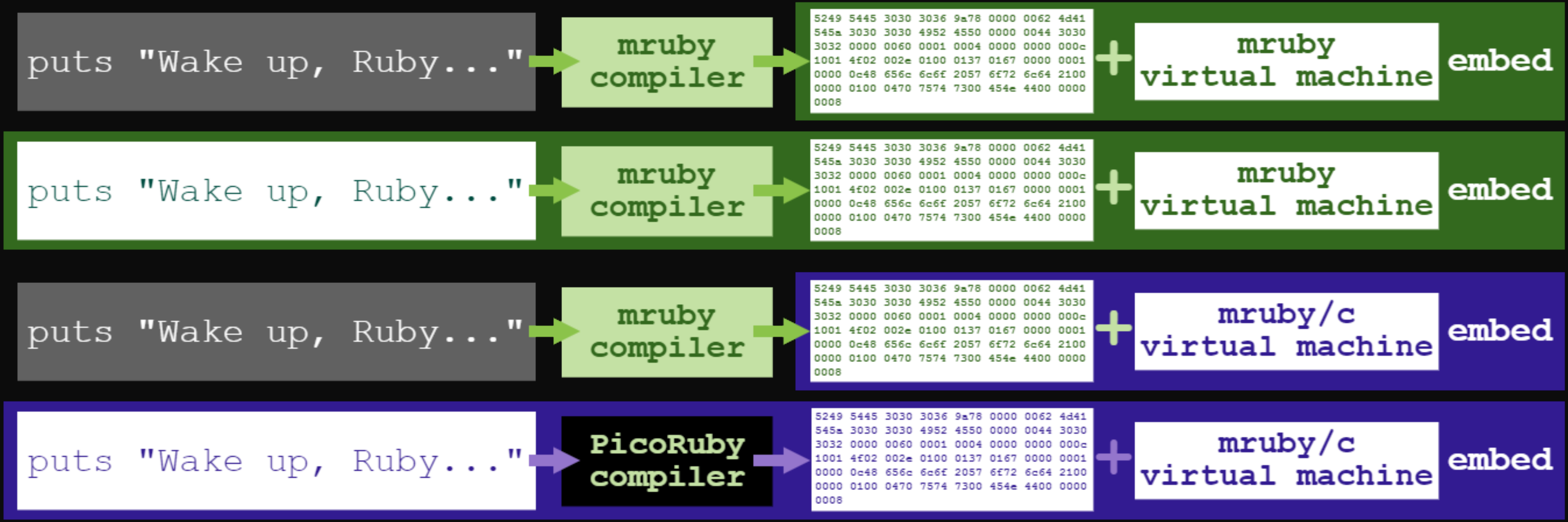
- Technically, you can embed also **Ruby script** and **mruby-compiler**
  - It makes sense if the computing resource is big enough

Architecture



- mruby/c is a smaller VM for one-chip microcontrollers
  - Naturally, we don't embed mruby-compiler with it

Architecture



PicoRuby = PicoRuby-compiler + mruby/cVM

Architecture

- As you know, the syntax of Ruby is complicated
- Easy to imagine that a Ruby compiler consumes large memory

Architecture

- You might think that you can use a microcontroller that has a big RAM
- The answer is "Yes and No"
  - You should do it if you can
- The bigger RAM, the bigger electric energy consumption
- The bigger resource, the more expensive in general

To make a small Ruby compiler

- mruby-compiler depends on mruby
  - The main reason for big footprint
  - PicoRuby compiler should be coded from scratch
- To make PicoRuby compiler small
  - Every fine logic should be minimal
  - Lemon parser generator instead of Bison/Yacc
    - A part of SQLite project

RAM consumption

```
$ valgrind                        \
   --tool=massif                  \
   --stacks=yes                   \
   mrbc hello_world.rb

$ valgrind                        \
   --tool=massif                  \
   --stacks=yes                   \
   picorbc hello_world.rb

$ ms_print massif.out.xxx | less
```

hello_world.rb (20 bytes)

```
puts "Hello World!"
```

mrbc hello_world.rb -> 133.5 KB

```
--------------------------------------------------------------------------------
Command:            mrbc hello_world.rb
Massif arguments:   --stacks=yes
--------------------------------------------------------------------------------
    KB
133.5^                                                                      #
     |                                                                     #
     |                                                                     #
     |                                                                     #
     |                                                                     #
     |                                                                     #
     |                                                                     #
     |                                                                     #
     |                                                                     #
     |                                                                @:::@#:
     |                                             :::@:::::::@:::@:::@:::@#::
     |                       ::::::@:::::@::@:::@::@::@:: ::::@:::@:::@:::@#::
     |                @@@@::::::: :@:: ::@::@: :@: @::@:: ::::@:::@:::@:::@#::
     |                @   :: :::: :@:: ::@::@: :@: @::@:: ::::@:::@:::@:::@#::
     |                @   :: :::: :@:: ::@::@: :@: @::@:: ::::@:::@:::@:::@#::
     |                @   :: :::: :@:: ::@::@: :@: @::@:: ::::@:::@:::@:::@#::
     |                @   :: :::: :@:: ::@::@: :@: @::@:: ::::@:::@:::@:::@#::
     |                @   :: :::: :@:: ::@::@: :@: @::@:: ::::@:::@:::@:::@#::
     |                @   :: :::: :@:: ::@::@: :@: @::@:: ::::@:::@:::@:::@#::
     |               @@   :: :::: :@:: ::@::@: :@: @::@:: ::::@:::@:::@:::@#::
   0 +----------------------------------------------------------------------->ki
     0                                                                   936.3
```

picorbc hello_world.rb -> 16.98 KB

```
--------------------------------------------------------------------------------
Command:            picorbc hello_world.rb
Massif arguments:   --stacks=yes
--------------------------------------------------------------------------------
    KB
16.98^                                                        #
     |                                                   @::@# :::@::  @
     |                                                  :@::@#::::@::::@ :::
     |                                                  :@::@#::::@::::@::::
     |                                                  :@::@#::::@::::@::::
     |                                                  :@::@#::::@::::@::::
     |                                                  :@::@#::::@::::@::::
     |                                                  :@::@#::::@::::@::::@
     |                                                  :@::@#::::@::::@::::@
     |                                                  :@::@#::::@::::@::::@
     |                                               @@::@::@#::::@::::@::::@
     |                                               @ ::@::@#::::@::::@::::@
     |                                               @ ::@::@#::::@::::@::::@
     |        ::                                     @ ::@::@#::::@::::@::::@
     |        :  :                                   @ ::@::@#::::@::::@::::@
     |        :  :                                   @ ::@::@#::::@::::@::::@
     |        : ::                                   @ ::@::@#::::@::::@::::@
     |        : ::                                   @ ::@::@#::::@::::@::::@
     |        : ::                                   @ ::@::@#::::@::::@::::@
     |       :: :: :::::::::::::::::::::::::::@:::@: @ ::@::@#::::@::::@::::@:
   0 +----------------------------------------------------------------------->ki
     0                                                                   222.0
```

keymap.rb of meish2 (2388 bytes)

```ruby
while !$mutex
  relinquish
end
kbd = Keyboard.new
kbd.init_pins(
  [ 6, 7 ],    # row0, row1
  [ 28, 27 ]   # col0, col1
)
kbd.add_layer :default, %i[ RUBY_GUI  KC_1     RAISE_ENTER LOWER_SPACE  ]
kbd.add_layer :raise,   %i[ FIBONACCI PASSWD   RAISE_ENTER ADJUST       ]
kbd.add_layer :lower,   %i[ KC_E      KC_F     RAISE_ENTER LOWER_SPACE) ]
kbd.add_layer :adjust,  %i[ KC_SCOLON KC_LSFT  RAISE_ENTER ADJUST       ]
kbd.define_mode_key :RAISE_ENTER, [ :KC_ENTER,              :raise,   200, 150 ]
kbd.define_mode_key :LOWER_SPACE, [ :KC_SPACE,              :lower,   300, 400 ]
kbd.define_mode_key :ADJUST,      [ nil,                    :adjust,  nil, nil ]
kbd.define_mode_key :RUBY_GUI,    [ Proc.new { kbd.ruby }, :KC_RGUI, 300, nil ]
class Fibonacci
  def initialize
    @a = 0 ; @b = 1
  end
  def take
    result = @a + @b
    @a = @b
    @b = result
  end
end
fibonacci = Fibonacci.new
kbd.define_mode_key :FIBONACCI, [ Proc.new { kbd.macro fibonacci.take }, :KC_NO, 300, nil ]
class Password
  def initialize
    @c = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_!@#$%^&*()=-+/[]{}<>'
  end
  def generate
    unless @srand
      # generate seed with board_millis
      srand(board_millis)
      @srand = true
    end
    password = ""
    while true
      i = rand % 100
      password << @c[i].to_s
      break if password.length == 8
    end
    return password
  end
end
password = Password.new
kbd.define_mode_key :PASSWD, [ Proc.new { kbd.macro password.generate, [] }, :KC_NO, 300, nil ]
kbd.before_report do
  kbd.invert_sft if kbd.keys_include?(:KC_SCOLON)
end
kbd.start!
```

mrbc keymap.rb -> 206.1 KB

```
--------------------------------------------------------------------------------
Command:            mrbc keymap.rb
Massif arguments:   --stacks=yes
--------------------------------------------------------------------------------
    KB
206.1^                                                                    ##
     |                                                                  @#
     |                                                                  @#
     |                                                                  @#
     |                                                                  @#
     |                                                                  @#
     |                                                                 :@#
     |                                                                 :@#
     |                                                                 :@#
     |                                                              ::::@#
     |                                                              :: :@# :
     |                                          @::::::::::::::::::::: :@# :
     |                                          @::: :: ::: : :: ::::: :@# :
     |                                    @:::::@::: :: ::: : :: ::::: :@# :
     |                  ::::::::@@::::::::@:: : @::: :: ::: : :: ::::: :@# ::
     |         @@::::::::: : :: @ :: : :: @:: : @::: :: ::: : :: ::::: :@# ::
     |         @ : :: : :: : :: @ :: : :: @:: : @::: :: ::: : :: ::::: :@# ::
     |         @ : :: : :: : :: @ :: : :: @:: : @::: :: ::: : :: ::::: :@# ::
     |         @ : :: : :: : :: @ :: : :: @:: : @::: :: ::: : :: ::::: :@# ::
     |         @ : :: : :: : :: @ :: : :: @:: : @::: :: ::: : :: ::::: :@# ::
   0 +----------------------------------------------------------------------->Mi
     0                                                                  1.476
```

picorbc keymap.rb -> 61.98 KB

```
--------------------------------------------------------------------------------
Command:            picorbc keymap.rb
Massif arguments:   --stacks=yes
--------------------------------------------------------------------------------
    KB
61.98^                                                                       #
     |                                                                     @#
     |                                                                  @  @@#
     |                                                                 @@:@@#
     |                                                             :   @   @@:@@#
     |                                                          @::::@:@:::@@:@@#
     |                                                   :      @::::@:@:::@@:@@#
     |                                              @@:@:::@::::@::::@:@:::@@:@@#
     |                                              @@:@:::@::::@::::@:@:::@@:@@#
     |                                     ::::@:::::@@:@:::@::::@::::@:@:::@@:@@#
     |                                     ::: @: :: @@:@:::@::::@::::@:@:::@@:@@#
     |                     @@@:::    :::::::: @: :: @@:@:::@::::@::::@:@:::@@:@@#
     |                     @@ :::@:::::: ::: @: :: @@:@:::@::::@::::@:@:::@@:@@#
     |          @:::: ::::@@ :::@:::::: ::: @: :: @@:@:::@::::@::::@:@:::@@:@@#
     |          @:::::::: @@ :::@:::::: ::: @: :: @@:@:::@::::@::::@:@:::@@:@@#
     | @@: :::::@:::::::: @@ :::@:::::: ::: @: :: @@:@:::@::::@::::@:@:::@@:@@#
     | @ ::: :::@:::::::: @@ :::@:::::: ::: @: :: @@:@:::@::::@::::@:@:::@@:@@#
     | @ ::: :::@:::::::: @@ :::@:::::: ::: @: :: @@:@:::@::::@::::@:@:::@@:@@#
     | @ ::: :::@:::::::: @@ :::@:::::: ::: @: :: @@:@:::@::::@::::@:@:::@@:@@#
     | @ ::: :::@:::::::: @@ :::@:::::: ::: @: :: @@:@:::@::::@::::@:@:::@@:@@#
   0 +----------------------------------------------------------------------->Mi
     0                                                                   7.101
```

Summary

#	script size	mrbc	picorbc
hello_world.rb	20 bytes	133.5 KB	16.98 KB
keymap.rb(meishi2)	2388 bytes	206.1 KB	61.98 KB

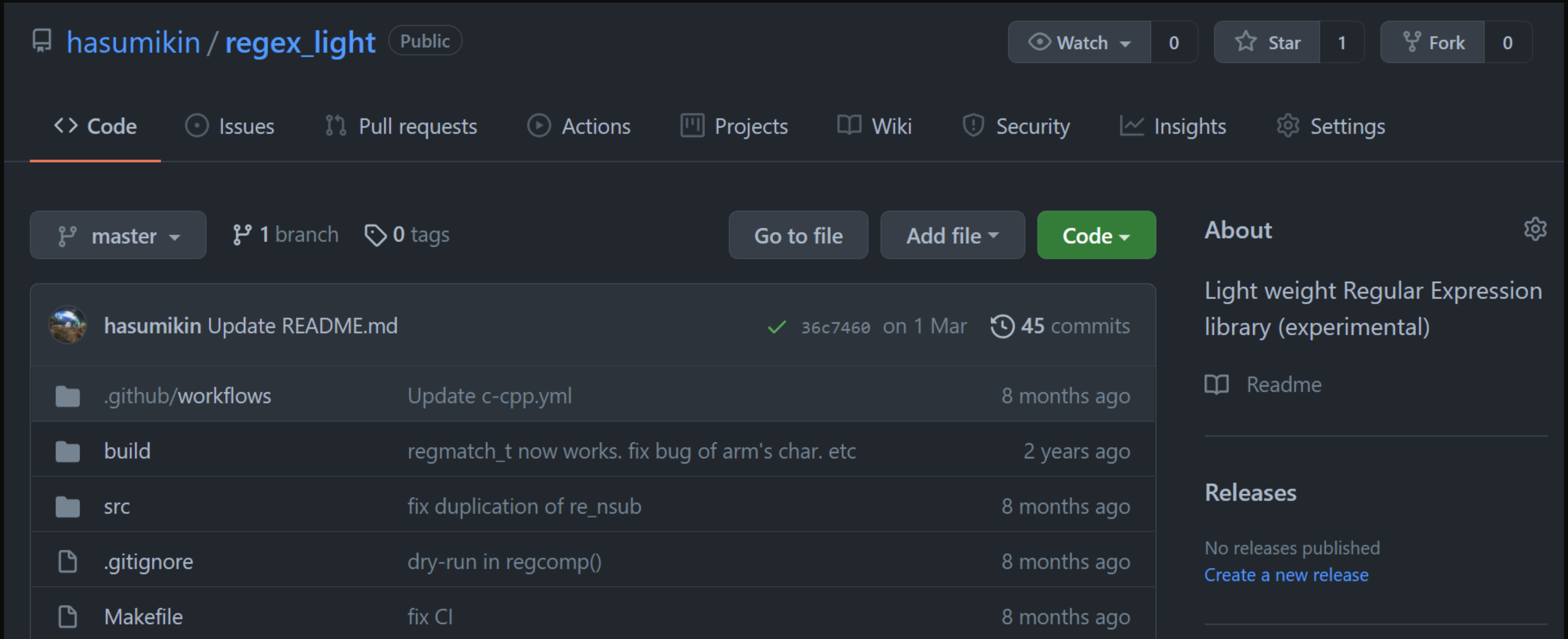
Note: Figures are measured in 64 bit Ubuntu

Implementation

- Depending on only libc and less
- Considering paddings and Pooled allocation
- Freeing heap memory in a loop instead of recursion

Depending on only libc and less

- glibc for desktop/server application
  - newlib for embedded application
    - Small code size
    - Doesn't have regex.c



Padding in a struct

```
struct LinkedList {
  struct LinkedList *next;  // 4 bytes (in 32-bit architecture)
  uint8_t           value;  // 1 byte
}

sizeof(LinkedList);
=> 5 🙅 No
=> 8 🙆 Yes
```

✔Data structure alignment (at least in C99)

pointer[4] + uint8_t[1] + **padding[3]** = sum[8]

👉You need to pack them well

Paddings waste memomy

```
LinkedList *top;
(...)
top->next->next->next->next; // 5 items in the list
```

- ❗Consumes 40 bytes to store 5 values of uint8_t
- ❗Total size of pointers will be 1 KB if there are 250 items in a list

Pooled allocation 💡

Pooled allocation

```
                                                        [n]: bytes

*next[4] -----------> *next[4] -----------> *next[4] ------------>
size[2]               size[2]               size[2]
index[2]              index[2]              index[2]
data[sizeof(Data)]    data[sizeof(Data)]    data[sizeof(Data)]
data[sizeof(Data)]    data[sizeof(Data)]    data[sizeof(Data)]
...                   ...                   ...
data[sizeof(Data)]    data[sizeof(Data)]    data[sizeof(Data)]
```

- Reduces paddings of data structure alignment
- Reduces the number of pointers
- Reduces fragmentation

Before improvement -> 53.20 KB

```
--------------------------------------------------------------------------------
Massif arguments:   --stacks=yes
--------------------------------------------------------------------------------
    KB
53.20^                                                                 #
     |                                                                 #
     |                                                                @#  :::
     |                                                                @#  : :
     |                                                                @#  : :
     |                                                               @@#  : :
     |                                                               @@# :: :
     |                                                               @@#::: :
     |                                                               @@#::: :
     |                                                           @@::@@#::: :
     |                                                           @@: @@#::: :
     |                                               @@:@:::@::::@@: @@#::: ::
     |                                         @::@::@ :@:: @:: :@@: @@#::: ::
     |                                 @:::@@@:@::@: @ :@:: @:: :@@: @@#::: ::
     |                           @@:@::@:: @@ :@::@: @ :@:: @:: :@@: @@#::: ::
     |                       ::@@@ :@: @:: @@ :@::@: @ :@:: @:: :@@: @@#::: ::
     |             :::::: :::::@ @ :@: @:: @@ :@::@: @ :@:: @:: :@@: @@#::: ::
     |      :::::@@:: :::::: ::@ @ :@: @:: @@ :@::@: @ :@:: @:: :@@: @@#::: ::
     |      : :: @ :: :::::: ::@ @ :@: @:: @@ :@::@: @ :@:: @:: :@@: @@#::: ::
     |      : :: @ :: :::::: ::@ @ :@: @:: @@ :@::@: @ :@:: @:: :@@: @@#::: ::
   0 +----------------------------------------------------------------------->Mi
     0                                                                   1.616
```

After improvement -> 24.21 KB

```
--------------------------------------------------------------------------------
Massif arguments:   --stacks=yes
--------------------------------------------------------------------------------
    KB
24.21^                                                                #
     |                                                        @ @:@:@:#
     |                                                    @:@@@:@:@:@:#:     @
     |                                               @@:::@:@@@:@:@:@:#::::::@
     |                                               @ :::@:@@@:@:@:@:#::::::@
     |                                               @ :::@:@@@:@:@:@:#::::::@
     |                                               @ :::@:@@@:@:@:@:#::::::@
     |                                               @ :::@:@@@:@:@:@:#::::::@
     |         @@:::::::::::::::@::::: ::::@::@::::::@ :::@:@@@:@:@:@:#::::::@
     |         @ ::::::: :::: : @:: ::@:: :@::@:: :: @ :::@:@@@:@:@:@:#::::::@
     |         @ ::::::: :::: : @:: ::@:: :@::@:: :: @ :::@:@@@:@:@:@:#::::::@
     |         @ ::::::: :::: : @:: ::@:: :@::@:: :: @ :::@:@@@:@:@:@:#::::::@
     |         @ ::::::: :::: : @:: ::@:: :@::@:: :: @ :::@:@@@:@:@:@:#::::::@
     |         @ ::::::: :::: : @:: ::@:: :@::@:: :: @ :::@:@@@:@:@:@:#::::::@
     |         @ ::::::: :::: : @:: ::@:: :@::@:: :: @ :::@:@@@:@:@:@:#::::::@
     |         @ ::::::: :::: : @:: ::@:: :@::@:: :: @ :::@:@@@:@:@:@:#::::::@
     |         @ ::::::: :::: : @:: ::@:: :@::@:: :: @ :::@:@@@:@:@:@:#::::::@
     |         @ ::::::: :::: : @:: ::@:: :@::@:: :: @ :::@:@@@:@:@:@:#::::::@
     |         @ ::::::: :::: : @:: ::@:: :@::@:: :: @ :::@:@@@:@:@:@:#::::::@
     |     ::::@ ::::::: :::: : @:: ::@:: :@::@:: :: @ :::@:@@@:@:@:@:#::::::@
   0 +----------------------------------------------------------------------->Mi
     0                                                                   1.082
```

Freeing in a loop

- Freeing long linked list in recursion causes "stack spike"
  - Because the call stack becomes deep until the recursive call reaches the bottom of the list
- Freeing in a loop instead of recursion
  - The call stack doesn't stack up at the cost of a bit complicated code

Summary

```
#                     script size      mrbc       picorbc
=================================================
hello_world.rb          20 bytes   133.5 KB    16.98 KB
-------------------------------------------------
keymap.rb(meishi2) 2388 bytes   206.1 KB    61.98 KB
-------------------------------------------------
```

Note: Figures are measured in 64 bit Ubuntu

Finally, I could make
PicoRuby work on the Micon!

The ~~Matrix~~

Micon

is Everywhere.

ON A KEYBOARD
RUBY ON BOARD

written and presented by
HASUMIKIN

supported by

RUBY ASSOCIATION and MONSTARLAB

mentor

MATZ

follow the

PICO
RUBY

an ESSENTIALLY RUBY production