

Ruby vs Kickboxer

Selena Small &
Michael Milewski

@selenasmall88 @saramic



What is Ruby vs Kickboxer?

Kickboxer 

COVID 

Lockdown 

No sparring ↔

Idea



What is Ruby vs Kickboxer?

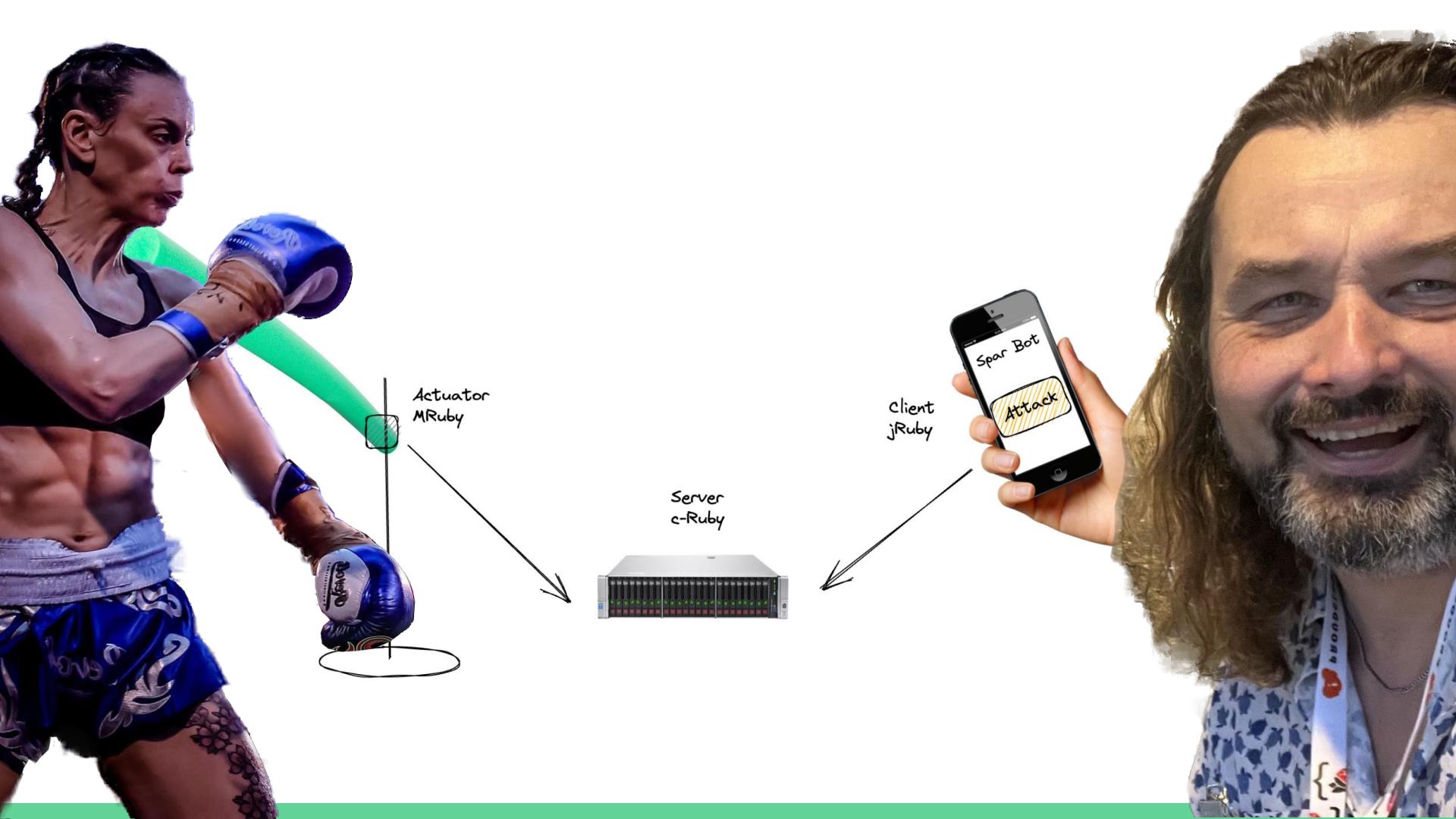


What is Ruby vs Kickboxer?



The plan

How will the different moving parts work together?



Actuator

(mRuby)

Hopefully this works?

Actuator - mRuby

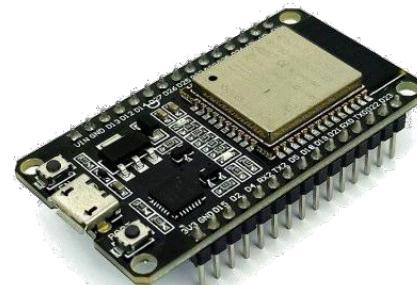
```
→ kickboxer-vs-ruby git:(main) asdf mruby
Unknown command: `asdf mruby`
No plugin named mruby
→ kickboxer-vs-ruby git:(main) █
```

Actuator - mRuby

- C project files
- mRuby source files
- mRuby components
- ESP-IDF
- Push code to ESP32



 **ESPRESSIF**



Actuator - mRuby

ESP32 C

```
#define LED_BUILTIN 2          // ESP32 built in led

void setup() {
    pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
    digitalWrite(LED_BUILTIN, HIGH);
    delay(1000);
    digitalWrite(LED_BUILTIN, LOW);
    delay(1000);
}
```

mRuby

```
led = ESP32::GPIO::GPIO_NUM_2 # ESP32 default Led pin

ESP32::GPIO.pinMode(led, ESP32::GPIO::OUTPUT)

loop do
    ESP32::GPIO.digitalWrite(led, ESP32::GPIO::HIGH)
    ESP32::System.delay(1000)
    ESP32::GPIO.digitalWrite(led, ESP32::GPIO::LOW)
    ESP32::System.delay(1000)
end
```

Actuator - mRuby

ESP32 C

```
#include "EspMQTTClient.h"
// a WiFi.h gets installed with EspMQTTClient and ArduinoJson
#ifndef ESP32
#include <WiFi.h>
#else
#include <ESP8266WiFi.h>
#endif
#include "secrets.h"
#include <HTTPClient.h>
#include <WiFiClientSecure.h>
#include <ESPmDNS.h>
#include <PubSubClient.h>

WiFiClientSecure net = WiFiClientSecure();
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
long hitStart = 0;
char actuator_topic[] = "kick/xx:xx:xx:xx:xx:xx"; // buffer
char serverIp[] = "xxx.xxx.xxx.xxx"; // buffer
char clientName[] = "ESP32/xx.xx.xx.xx.xx.xx"; // buffer

#define LED_BUILTIN 2 // ESP32 builtin led

void connectWifi() {
    WiFi.mode(WIFI_STA);
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

    Serial.println("Connecting to Wi-Fi");

    int count = 0;
    while (WiFi.status() != WL_CONNECTED) {
        count++;
        delay(500);
        Serial.print(".");
        if (count % 20 == 0) {
            count = 0;
            Serial.println();
            WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
            Serial.println("Connecting to Wi-Fi");
        }
    }
}
```

mruby

```
# frozen_string_literal: true

last_msg = 0
hit_start = 0

wifi = ESP32::WiFi.new

puts "Connecting to wifi"

wifi.connect("loki", "password") # TODO: how best to externalise secrets
# client_name = "ESP32/#{wifi.mac_address}" # TODO: why no client name
actuator_topic = "kick/wifi.mac_address"

puts "Connected"

led = ESP32::GPIO::GPIO_NUM_2 # ESP32 default led pin
ESP32::GPIO.pinMode(led, ESP32::GPIO::OUTPUT)

mqtt = ESP32::MQTT::Client.new("failure-driven.local", 1883)
mqtt.connect
mqtt.publish("kick/manage", '{ "message": "OK", "actuator": "' + actuator_topic + '" }')
mqtt.subscribe(actuator_topic)

loop do
    now = ESP32::Timer.get_time
    if (now - last_msg) > 5_000_000 # 5 second ping
        last_msg = now
        mqtt.publish("kick/manage", '{ "message": "OK", "actuator": "' + actuator_topic + '" }')
    end
    if hit_start > 0 && now - hit_start > 170_000 # 170 milliseconds
        hit_start = 0
        ESP32::GPIO.digitalWrite(led, ESP32::GPIO::LOW)
    end
    ESP32::System.delay(20)

    # this seems to block
    topic, _message = mqtt.get
    if topic == actuator_topic
        puts "HIT"

        ESP32::GPIO.digitalWrite(led, ESP32::GPIO::HIGH)
        hit_start = ESP32::Timer.get_time
    end
end
```

Actuator Ruby

ESP32 C

```
#include "EspNetClient.h"
// a lot of code here

    }
}

char * m
{
    IPAddress ip;
    Serial
    {
        IPAddress
        Serial
        "for"
        void re
        void lo
        void t
        long l
        long h
        char a
        char c
        char o
        #defin
        void co
        WiFi
        WiFi
        void se
        // pu
        Serial
        Serial
        int (*ESP3
        while
        == 24
        ("kick"
        de
        Se
        if
        connect
        Serial
        if (!N
        Serial
        while
        de
        }
        if (Str
        Serial
        for (in
        Serial
        messa
        messa
        }
        Serial.
        if (Str
        Serial
        Serial.println("HIT");
    }
}
```

mruby

```
# frozen_string_literal: true

last_msg = 0
hit_start = 0

wifi = ESP32::WiFi.new

puts "Connecting to wifi"
# client_name = "Esp32" # DO: why no client name
actuator_topic = "ki

puts "Connected"

led = ESP32::GPIO::G
ESP32::GPIO.pinMode(led,
                    :OUTPUT)

mqtt = ESP32::MQTT::Client.new("192.168.1.10", 1883)
mqtt.connect
mqtt.publish("kick/master", "meow")
actuator_topic + '')')
mqtt.subscribe(actuator_topic)

loop do
  now = ESP32::Timer.get_millis
  if (now - last_msg) >= 170
    last_msg = now
    mqtt.publish("kick/master", "meow", "actuator":'"' +
actuator_topic + '"')
  end
  if hit_start > 0 && now - hit_start > 170 # 170 milliseconds
    hit_start = 0
    ESP32::GPIO.digitalWrite(led, :HIGH)
  end
  ESP32::System.delay(20)

  # this seems to block
  topic, _message = mqtt.get
  if topic == actuator_topic
    puts "HIT"
    ESP32::GPIO.digitalWrite(led, :HIGH)
    hit_start = ESP32::Timer.get_millis
  end
end
}
```

Actuator - mRuby

ESP32 C

```
void connectWifi() {
    WiFi.mode(WIFI_STA);
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

    Serial.println("Connecting to Wi-Fi");

    int count = 0;
    while (WiFi.status() != WL_CONNECTED) {
        count++;
        delay(500);
        Serial.print(".");
        if (count % 20 == 0) {
            count = 0;
            Serial.println();
            WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
            Serial.println("Connecting to Wi-Fi");
        }
    }
}
```

mruby

```
wifi = ESP32::WiFi.new

puts "Connecting to wifi"

wifi.connect("kickboxer", "password")
```

Actuator - mRuby

ESP32 C

```
char serverIp[] = "xxx.xxx.xxx.xxx"; // buffer

char * mdns = NULL;

IPAddress resolve_mdns_host(const char * hostname)
{
    IPAddress ip = MDNS.queryHost(hostname, 2000); // 2 second timeout
    Serial.printf(
        "found failure-driven: %s.%s.%s.%s\n",
        String(ip[0]),
        String(ip[1]),
        String(ip[2]),
        String(ip[3])
    ); // how to IPAddress toString
    return ip;
}

...
if (!MDNS.begin("ESP32_Browser")) {
    Serial.println("Error setting up MDNS responder!");
    while (1) {
        delay(1000);
    }
}
resolve_mdns_host(MQTT_SERVER_NAME).toString().toCharArray(serverIp,
16);
Serial.printf("found %s: %s\n", MQTT_SERVER_NAME, serverIp);
```

mruby

```
mqtt = ESP32::MQTT::Client.new(
    "failure-driven.local", 1883,
)
mqtt.connect
```

Actuator - mRuby

ESP32 C

```
void connectMQTT(const char * mqtt_server) {
    client.setServer(mqtt_server, 1883);
    client.setCallback(callback);
    reconnect();
    Serial.println("MQTT IoT Connected!");
}

void loop() {
    long now = millis();
    if (now - lastMsg > 5000) { // 5 second ping
        lastMsg = now;
        Serial.println("management ping");
        char buffer[60];
        strcpy(
            Buffer,
            "{\"message\":\"OK\", \"actuator\":\"\"}");
        strcat(buffer, actuator_topic);
        strcat(buffer, "\"}");
        client.publish("kick/manage", buffer);
    }
}
```

mRuby

```
mqtt = ESP32::MQTT::Client.new(
    "failure-driven.local", 1883
)
mqtt.connect
mqtt.subscribe(actuator_topic)

loop do
    now = ESP32::Timer.get_time
    if (now - last_msg) > 5_000_000 # 5 second ping
        last_msg = now
        mqtt.publish(
            "kick/manage",
            '{ "message": "OK", "actuator": "' + actuator_topic + '" }',
        )
    End
```

Server (cRuby)

Piece of cake!

Server - cRuby

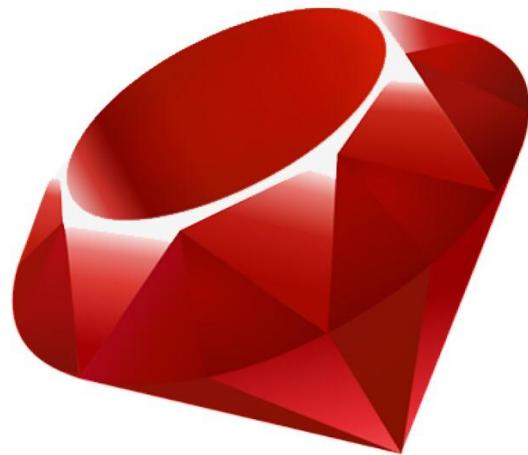
A screenshot of a terminal window with two tabs. The top tab shows the command `demo_using_mqtt git:(main) docker-compose up`. The bottom tab shows the command `demo_using_mqtt git:(main) mosquitto_sub -h localhost -t \# -d`.

```
demo_using_mqtt git:(main) docker-compose up
demo_using_mqtt git:(main) mosquitto_sub -h localhost -t \# -d
```

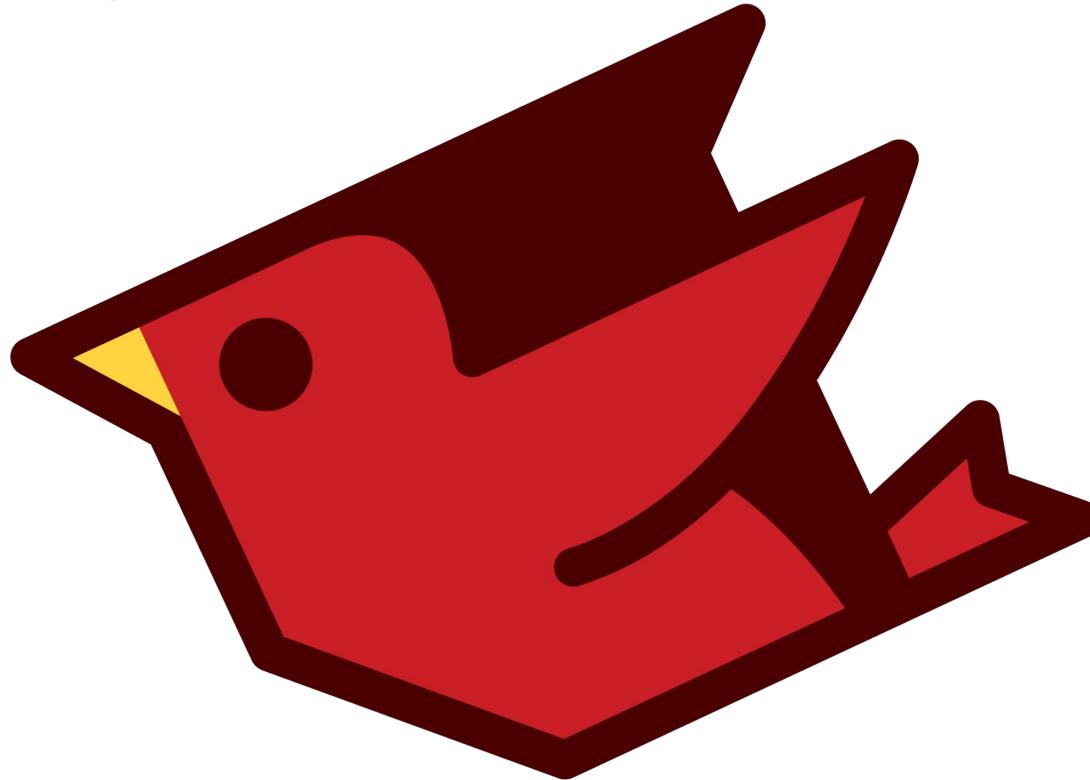
Client
(jRuby)

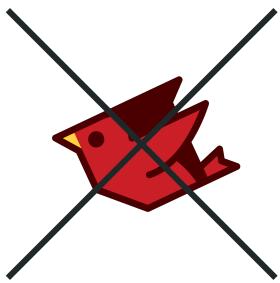
Surely it just works on
Android devices

Client - Ruby



Client - jRuby





Client - Ruboto



Client - Ruboto



← → C Not Secure | ruboto.org

GoDaddy®

ruboto.org

has expired and is parked free, courtesy of GoDaddy.com.

Get This Domain

A screenshot of a web browser window. The address bar shows "Not Secure | ruboto.org". The main content area displays the GoDaddy logo and the text "ruboto.org has expired and is parked free, courtesy of GoDaddy.com." Below this is a black button with the text "Get This Domain".

Related searches

- Dicom Viewer >
- Dicom Viewer Free >
- Flights >

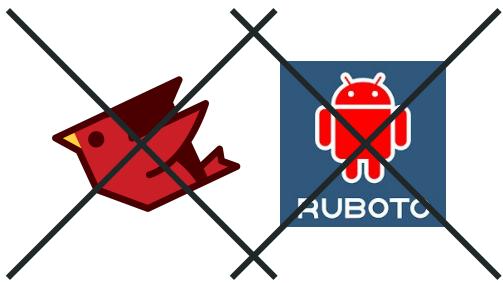
A screenshot of a search results page. It features a header "Related searches" followed by three blue rectangular buttons, each containing a magnifying glass icon and a search term with a right-pointing arrow. The terms are "Dicom Viewer", "Dicom Viewer Free", and "Flights".

Client - Ruboto



```
$ export ANDROID_HOME=/Users/michael/Library/Android/sdk
$ export PATH=/Users/michael/Library/Android/sdk/platform-tools/:$PATH
$ asdf shell ruby jruby-9.4.2.0
$ ruby -v
  jruby 9.4.2.0 (3.1.0) 2023-03-08 90d2913fda OpenJDK 64-Bit Server VM 20+36-2344 on 20+36-2344 +jit
[arm64-darwin]
$ gem install ruboto
  Fetching ruboto-1.6.1.gem
  Successfully installed ruboto-1.6.1
  1 gem installed
$ ruboto gen app --package com.failure-driven --name Test --target android-30

/Users/michael/.asdf/install/ruby/jruby-9.4.2.0/lib/ruby/gems/shared/gems/ruboto-1.6.1/lib/ruboto/util/objectspace.rb:5: warning: ObjectSpace impacts performance. See
https://github.com/jruby/jruby/wiki/PerformanceTuning#dont-enable-objectspace
/Users/michael/.asdf/install/ruby/jruby-9.4.2.0/lib/ruby/gems/shared/gems/ruboto-1.6.1/lib/ruboto/util/main_fix.rb:7: warning: finalizer references object to be finalized
F, [2023-05-05T08:57:08.616086 #32463] FATAL -- : uninitialized constant Main::GetoptLong::FALSE (NameError)
org/jruby/RubyModule.java:4309:in `const_missing'
/Users/michael/.asdf/install/ruby/jruby-9.4.2.0/lib/ruby/gems/shared/gems/main-5.3.2/lib/main/getoptlong.rb
:73:in `initialize'
```



Client - RubyMotion



Client - RubyMotion



download <http://www.rubymotion.com/download>



Client - RubyMotion

```
download http://www.rubymotion.com/download
```

```
motion android-setup
```

```
January 16, 2019
```

```
-----  
Accept? (y/N): Y
```

```
[=====] 100% Unzipping... android-13/framework  
[=====] 100% Unzipping... x86_64/VerifiedBootP  
[=====] 100% Unzipping... src/tck/java/time/TC  
[=====] 100% Unzipping... android-13/llvm-rs-c  
[=====] 100% Unzipping... platform-tools/NOTIC
```



Client - RubyMotion

```
[=====] 100% Unzipping... android-13/llvm-rs-c
[=====] 100% Unzipping... platform-tools/NOTIC
January 16, 2019
-----
Accept? (y/N): Y
[=====] 100% Unzipping... android-ndk-r21e/wra
[=====] 100% Unzipping... android-8.0.0/templa
[=====] 100% Unzipping... src/android/telecom/
[=====] 100% Unzipping... android-8.0.0/zipali
[=====] 100% Unzipping... android-8.1.0/build.
...
[=====] 100% Unzipping... android-12/framework
[=====] 100% Unzipping... src/tck/java/time/TC
[=====] 100% Unzipping... android-12/llvm-rs-c
[=====] 100% Computing updates...
[=====] 100% Computing updates...
[=====] 100% Computing updates...
To install legacy android sdks (API < 26), run `motion android-setup-legacy`
=====
WARNING: Additional commands need to be downloaded. Please run: `motion repo`.
```

Client - RubyMotion



```
To install legacy android sdks (API < 26), run `motion android-setup-legacy`  
=====  
WARNING: Additional commands need to be downloaded. Please run: `motion repo`.  
=====  
motion repo  
Cloning RubyMotion templates. Feel free to browse /Users/michael/.rubymotion/rubymotion-templates to see how  
they're built.  
Cloning into '/Users/michael/.rubymotion/rubymotion-templates'...  
remote: Enumerating objects: 1244, done.  
remote: Counting objects: 100% (26/26), done.  
remote: Compressing objects: 100% (14/14), done.  
remote: Total 1244 (delta 13), reused 12 (delta 12), pack-reused 1218  
...  
Cloning into '/Users/michael/.rubymotion/rubymotion-command'...  
remote: Enumerating objects: 70, done.  
remote: Counting objects: 100% (10/10), done.  
remote: Compressing objects: 100% (8/8), done.  
remote: Total 70 (delta 4), reused 7 (delta 2), pack-reused 60  
Receiving objects: 100% (70/70), 19.36 KiB | 9.68 MiB/s, done.  
Resolving deltas: 100% (23/23), done.  
INFO: Repository synchronization performed on: 2023-05-05.
```



Client - RubyMotion

```
$ motion create --template=android ruby_motion_mqtt_blink
[!] You may want to run `motion repo` if it's been a while since you've updated templates.

Create ruby_motion_mqtt_blink
Create ruby_motion_mqtt_blink/.gitignore
Create ruby_motion_mqtt_blink/Gemfile
Create ruby_motion_mqtt_blink/Rakefile
Create ruby_motion_mqtt_blink/app/main_activity.rb
Create ruby_motion_mqtt_blink/spec/main_spec.rb
```



Client - RubyMotion

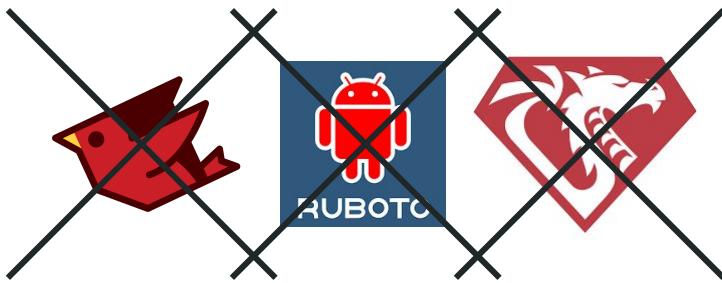
```
$ motion create --template=android ruby_motion_mqtt_blink
[!] You may want to run `motion repo` if it's been a while since you've updated templates.

Create ruby_motion_mqtt_blink
Create ruby_motion_mqtt_blink/.gitignore
Create ruby_motion_mqtt_blink/Gemfile
Create ruby_motion_mqtt_blink/Rakefile
Create ruby_motion_mqtt_blink/app/main_activity.rb
Create ruby_motion_mqtt_blink/spec/main_spec.rb

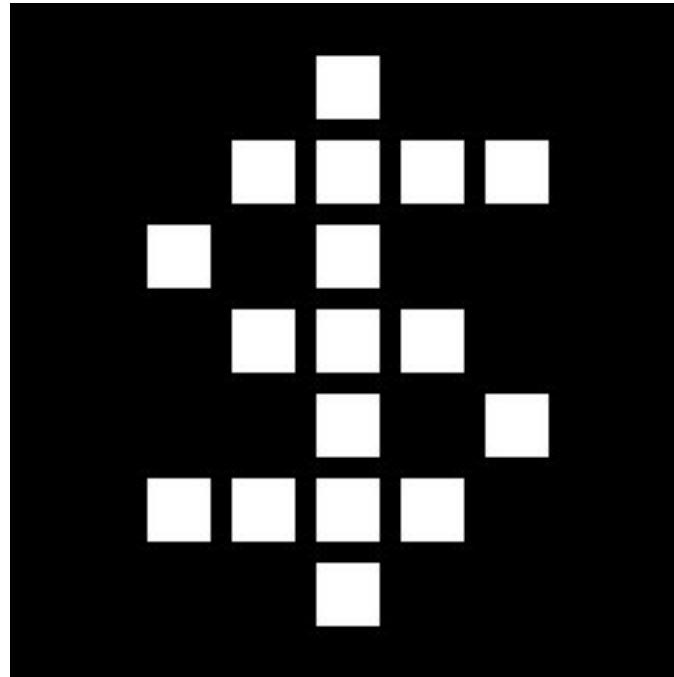
$ cd ruby_motion_mqtt_blink

$ rake

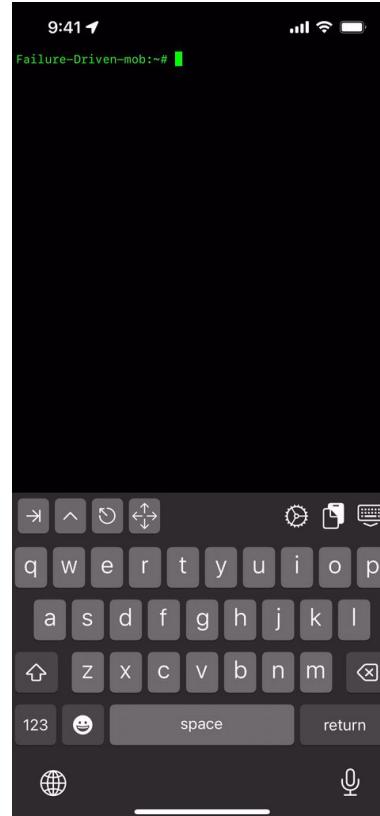
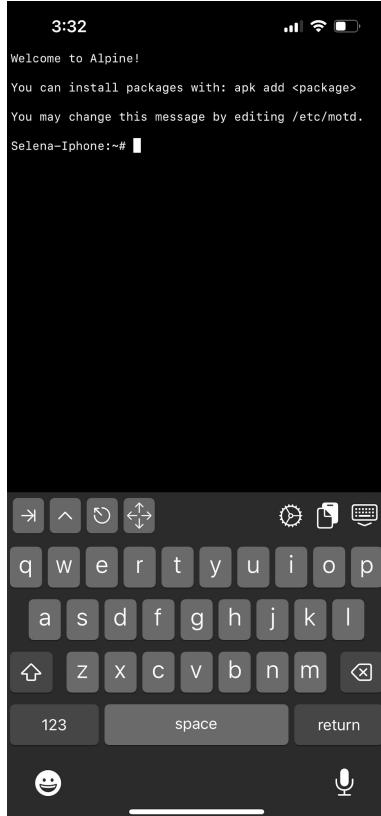
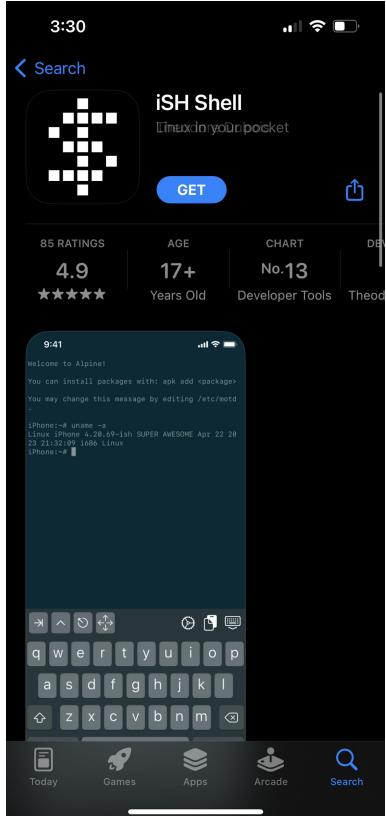
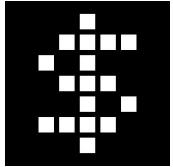
Resolving dependencies...
ERROR! app.sdk_path should point to a valid Android SDK directory. Run 'motion android-setup-legacy' to
install the latest SDK version.
```



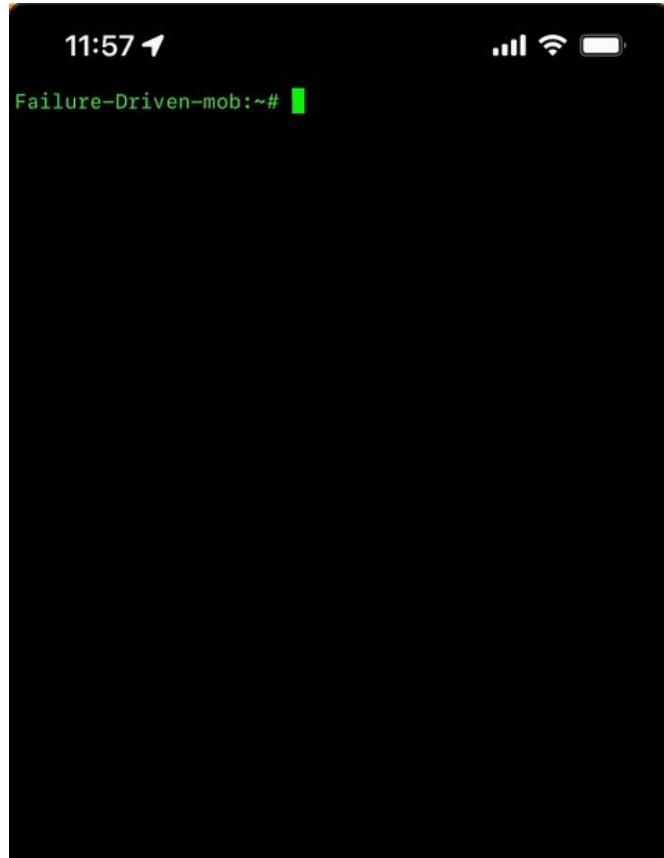
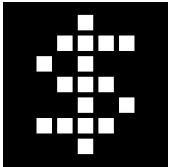
Client - iSH

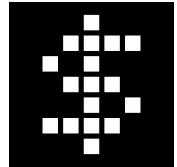


Client - iSH



Client - iSH

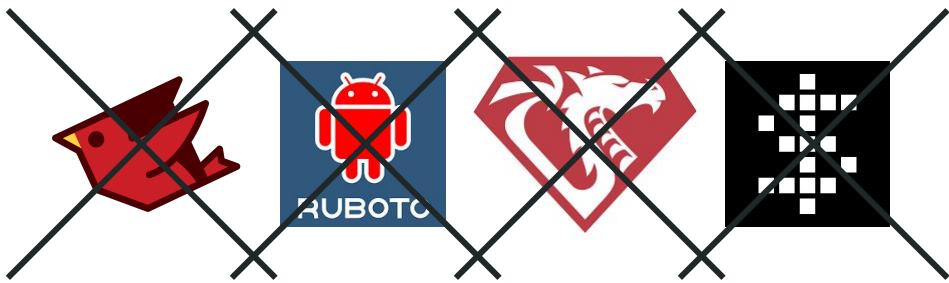




Client - iSH



A screenshot of a smartphone displaying a terminal window. The top status bar shows the time as 9:41, signal strength, Wi-Fi, and battery level. The terminal window has a black background with white text. It displays the command line: "Failure-Driven-mob:~/kickboxer-vs-ruby-main/apps/demo_using_mqtt#". A green cursor is visible at the end of the command line.



Client - Rubyist



Client - Rubyist



7:19

Search

Rubyist - Ruby Scripting
Run and Automate Ruby Code

OPEN

2 RATINGS 5.0 AGE 4+ CATEGORY Developer Tools DEV

What's New Version History

Version 1.1.2 1y ago

* New side-by-side console on iPad
* Fixes share menu on iPad

Preview

Today Games Apps Arcade Search

7:46

Output Done

undefined method 'require' (NoMethodError)

8:10 Hello World

```
1 class Greeter
2   def initialize(name)
3     @name = name.capitalize
4   end
5
6   def salute
7     puts "Hello #{@name}!"
8   end
9
10 g = Greeter.new("world")
11 puts g.salute
12
```

7:26 Done

Documentation

Rubyist MRuby

A

- Alert Constructs and presents alerts.
- Alert::Action Represents an action button for an alert.
- Alert::Textfield Represents a text field for an alert.

B

- Base64 Encodes and decodes Base64 data.
- Browser Opens a website.

C

- Clipboard Copies and pastes data from the device's clipboard.
- Color Represents a Color.
- Confirm Presents an alert asking for confirmation.

D

- Data

7:22 App Store More Done

Share

AUTOMATE

- Siri
- Shortcuts
- Automation Triggers

RUBYIST

- Documentation
- Support Rubyist
- About

Client - Rubyist



7:28

Widget.rb

```
35 # You can combine these few basic
36 # concepts to display almost any information
37 # on your home screen.
38 btc_widget = WidgetUI.new do
39   API = "https://api.coindesk.com/v1/bpi/
40   currentprice/USD.json"
41   # socket = TCPSocket.new("failure-
42   # driven.local", 1883)
43   # socket = TCPSocket.new("10.0.0.166",
44   # 1883)
45   a = TCPSocket.getaddress("failure-
46   driven.local")
47   r = JSON.parse(HTTPRequest.get(API).body)
48   rate = r["bpi"]["USD"]["rate"]
49
50   HStack {
51     Spacer()
52     VStack(alignment: :center) {
53       Text("BTC").size(65)
54       Text(a).size(14)
55     }
56     Spacer()
57   }.background(
58     Gradient.new([Color.hex("F14034"),
59     Color.hex("AD2323")])
60   ).frame(
61     width: :infinity,
62     height: :infinity
63   )
64   .padding(15)
65 end
66
67 # Uncomment this line to preview the second
68 # widget
69 btc_widget.present
70
71 # Please share your creations with me on
72 # twitter @RubyistApp
73 # hello@rubystartup.app
74 # - Sebastian
```

Widget.rb

Widget UI Preview

Output Done

Large Widget

Kickboxer 🎾 - Rubyist

kick/left

kick/right

RESUME - 49 MIN LEFT - 21 APRIL

The Art of Personalized, Custom Experiences with Fac...

RESUME - 3 HRS 14 MIN. LEFT - 24 APRIL

1549 - "AI Gore Rhythms"

THURSDAY

4

TOMORROW

Pay rent Felix

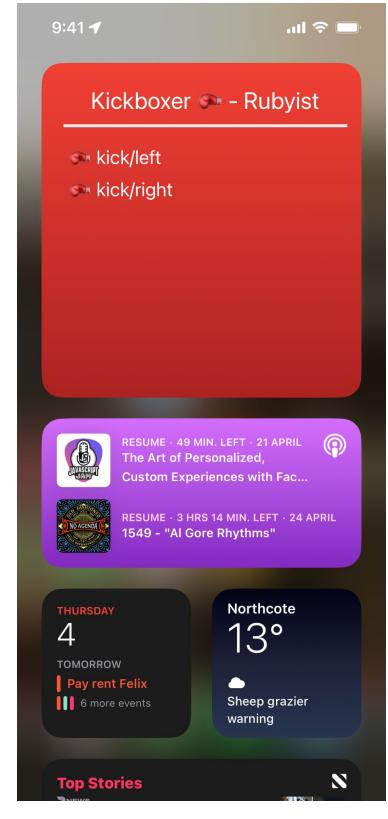
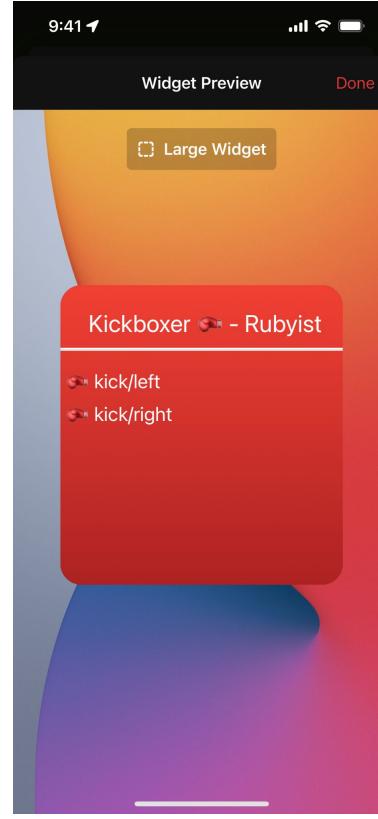
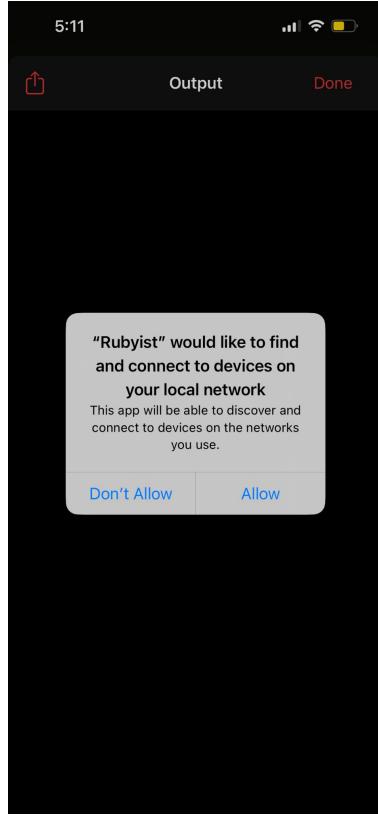
6 more events

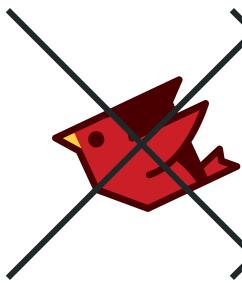
Northcote

13°

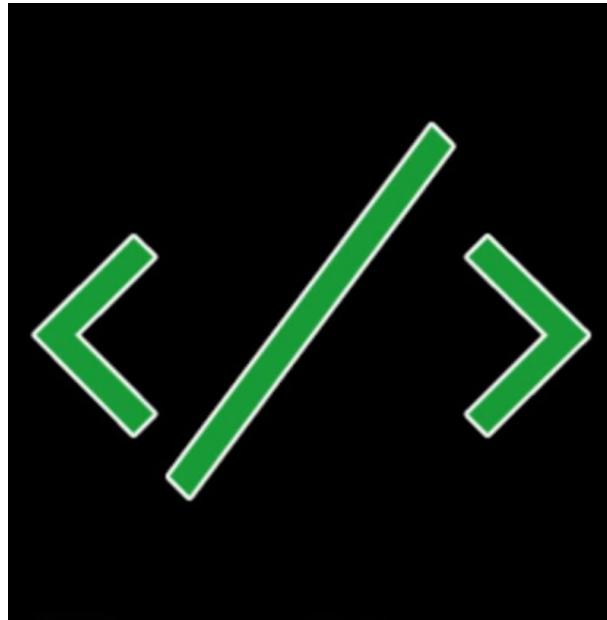
Sheep grazier warning

Top Stories





Client - Termux



Client - Termux



Termux

Docs Posts Packages Donate Privacy Policy

en

Termux is an [Android terminal emulator and Linux environment app](#) that works directly with no rooting or setup required. A minimal base system is installed automatically - additional packages are available using the APT package manager.

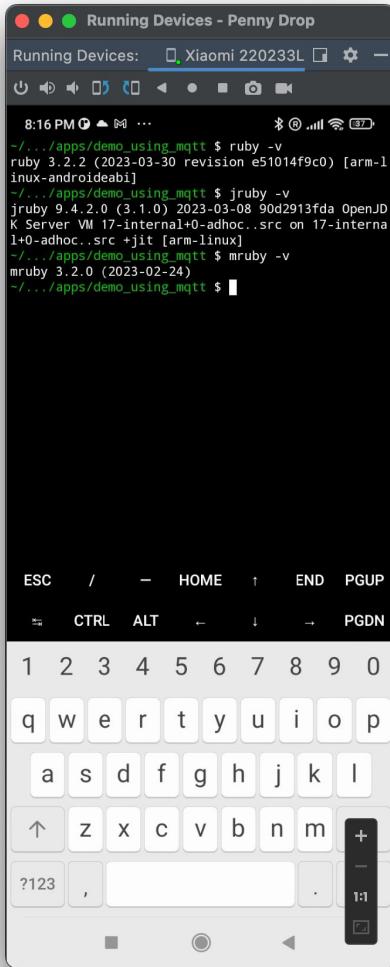
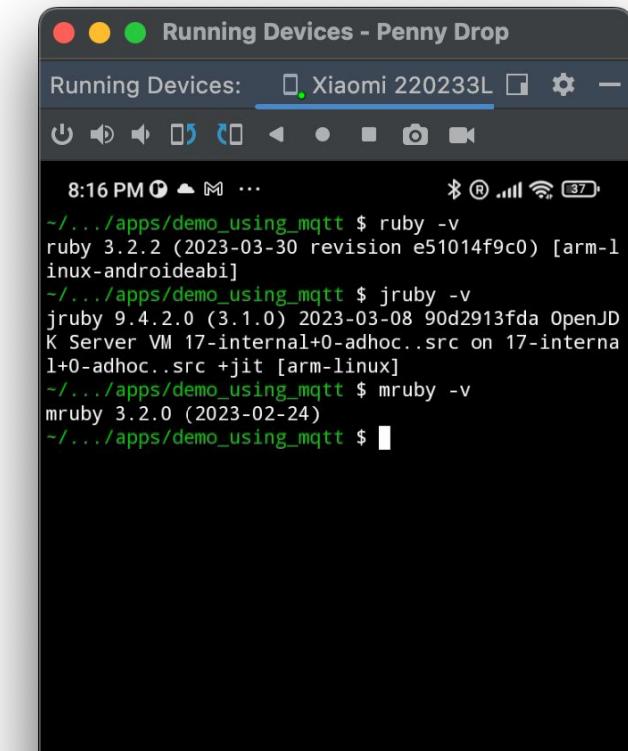
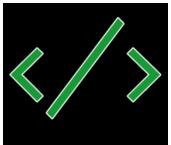


Secure. Access remote servers using the ssh client from OpenSSH. Termux combines standard packages with accurate terminal emulation in a beautiful open source solution.

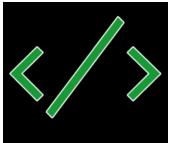
Feature packed. Take your pick between Bash, fish or Zsh and nano, Emacs or Vim. Grep through your SMS inbox. Access API endpoints with curl and use rsync to store backups of your contact list on a remote server.



Client - Termux



Client - Termux

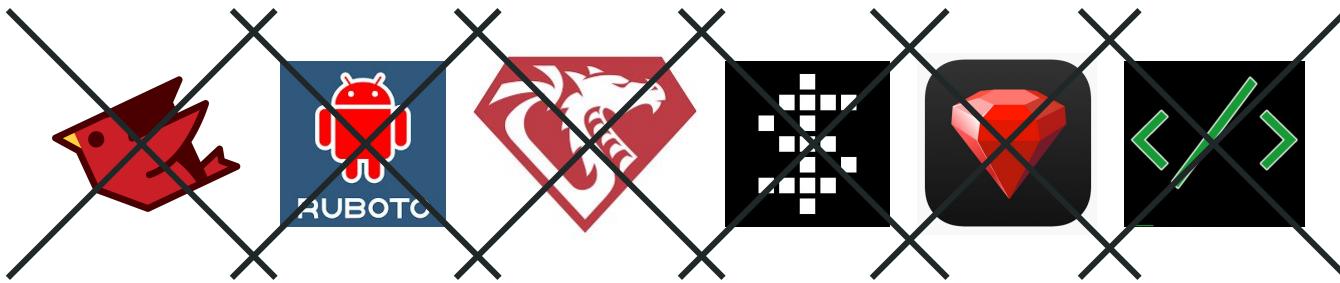


ruby-3.2.2

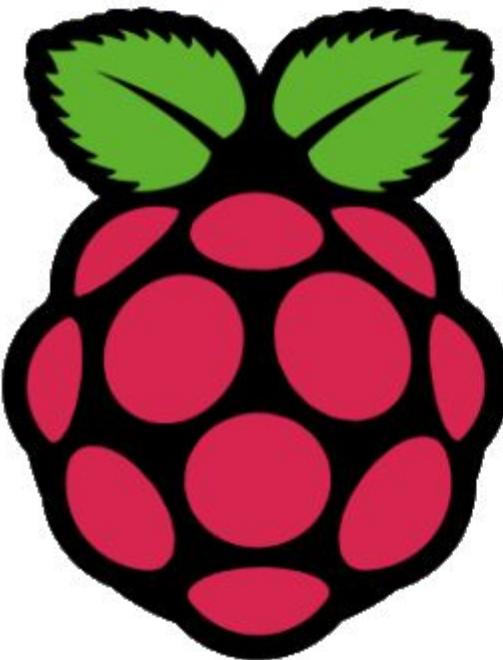
A screenshot of a Termux window titled "Running Devices". The title bar shows "Running Devices" and the device name "Xiaomi 220233L". The status bar at the top displays the time "8:17 PM", battery level "36%", and signal strength. The main terminal area shows the command "ruby version" followed by the output "ruby version: ruby-3.2.2". Below the terminal are sections for "actuators" and "Event Log". The "Event Log" section contains several entries starting with "kick/manage", each with a message field containing "OK". A green arrow points from the "ruby-3.2.2" label to the terminal output.

jruby-3.1.0

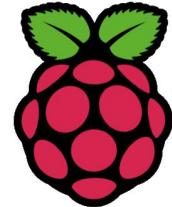
A screenshot of a Termux window titled "Running Devices". The title bar shows "Running Devices" and the device name "Xiaomi 220233L". The status bar at the top displays the time "8:19 PM", battery level "36%", and signal strength. The main terminal area shows the command "ruby version" followed by the output "ruby version: jruby-3.1.0". Below the terminal are sections for "actuators" and "Event Log". The "Event Log" section contains several entries starting with "kick/manage", each with a message field containing "OK". A green arrow points from the "jruby-3.1.0" label to the terminal output.



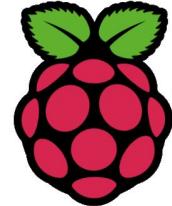
Client - jRuby on RasPi



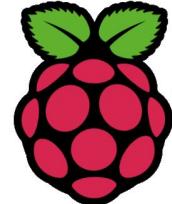
Client - jRuby on RasPi

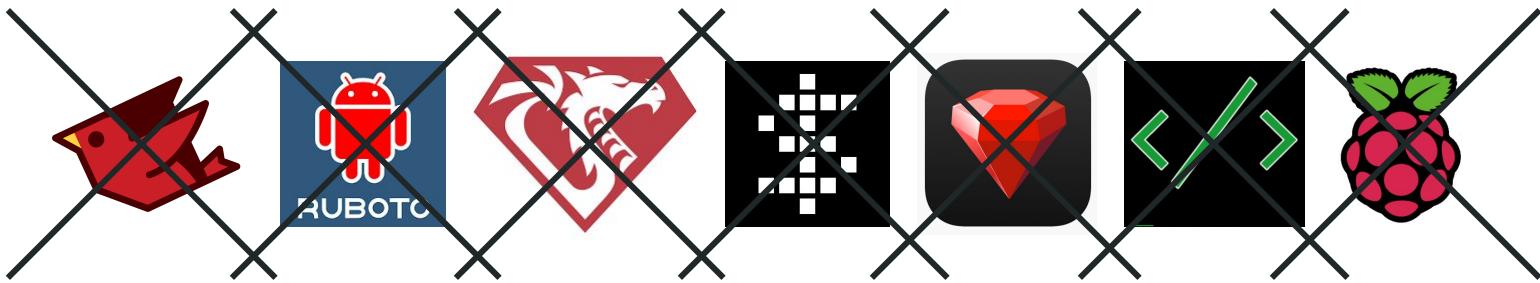


Client - jRuby on RasPi



Client - jRuby on RasPi





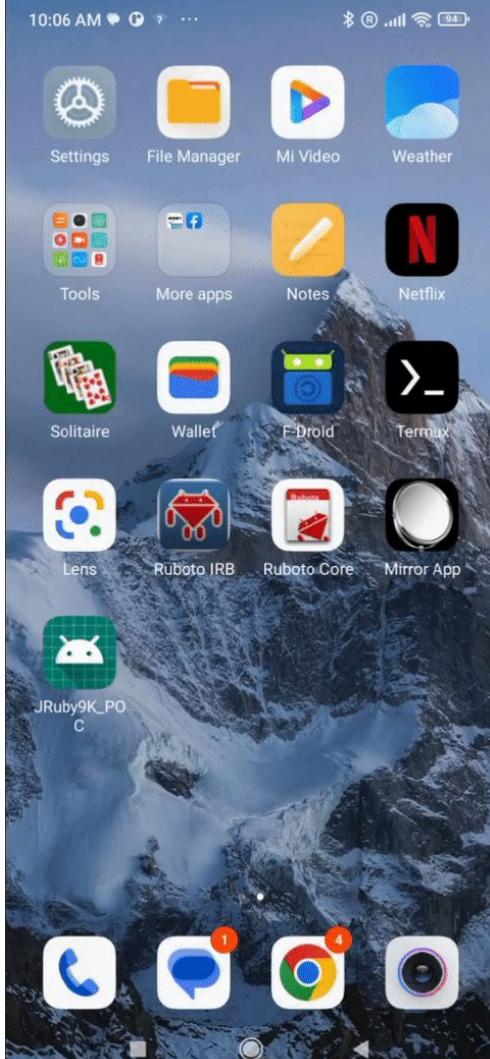
Client - Ruboto + Arduino



+



Client Ruboto + Arduino



Client - Ruboto + Arduino



Demo - placeholder

What worked

1. mRuby on ESP32 actuator
2. MQTT messaging service
3. cRuby on iSH (iOS)
4. mRuby on Rubyist (iOS)
5. cRuby and jRuby on Termux (Android)



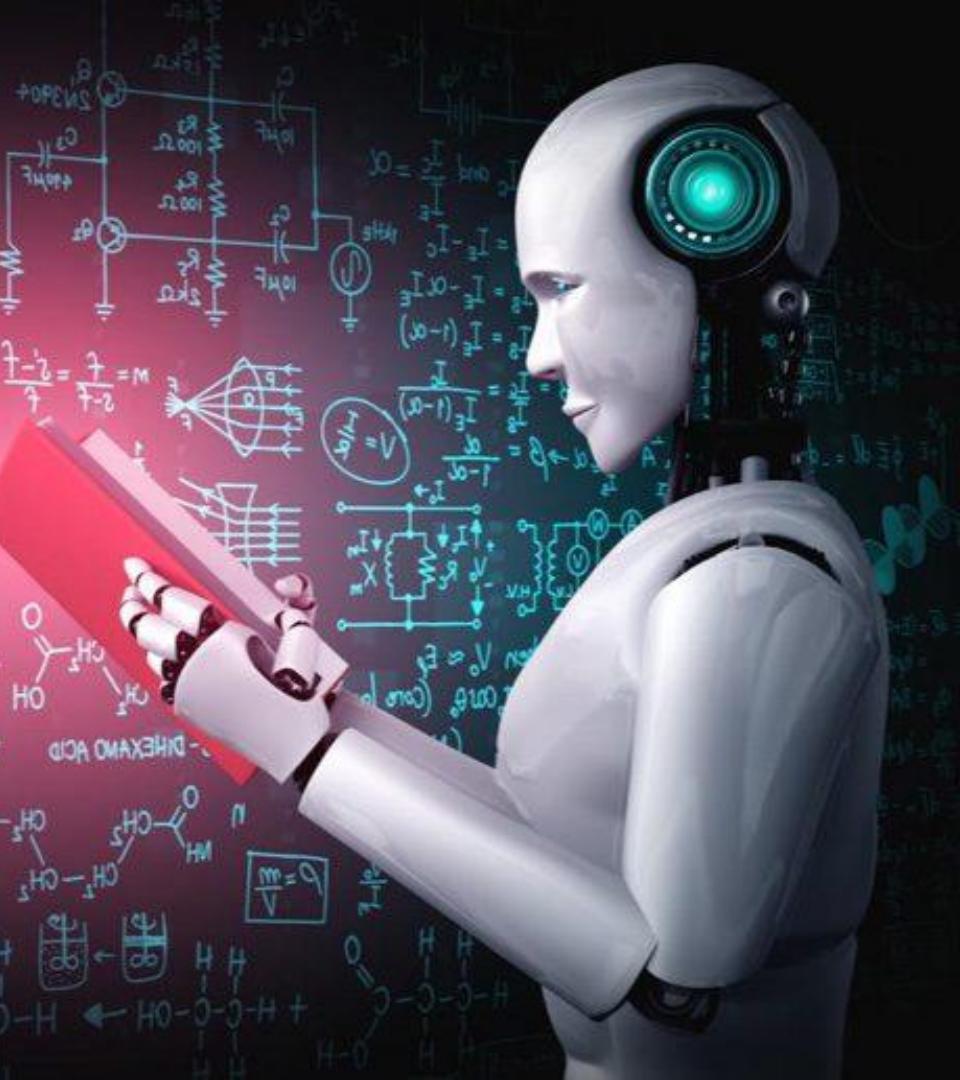
What didn't work

1. Ruboto - needs updating
2. Ruby on Mobile? Is it a good idea



What's next?

- Machine Learning - iRuby



What's next?

- Machine Learning - iRuby
- Actually rewrite C code in Ruby



What's next?

- Machine Learning - iRuby
- Actually rewrite C code in Ruby
- Contribute to Ruboto



What's next?

- Machine Learning - iRuby
- Actually rewrite C code in Ruby
- Contribute to Ruboto
- Get involved in mRuby



Selena Small & Michael Milewski

@selenasmall 88

FRESHO



@saramic

zepto

