

# mruby/cを用いた プログラミング教育向けデバイスの 開発

---

株式会社ビジネスバンクグループ

牧 俊男

# 会社紹介

株式会社ビジネスバンクグループ



ミッション：

日本の開業率を10%に引き上げます！

主な事業内容：

起業家をターゲットとした起業支援コンサルティング  
経営者向け教育・セミナーの開催  
中小企業向け経営ツール「ALL-IN」(SaaS) の開発



経営をアップグレードしよう。

ALL-INは企業に必要なシステムがオールインワン、  
ビジネスを驚くほど効率化するクラウドサービスです。



# 自己紹介

牧 俊男

株式会社ビジネスバンクグループ  
ALL-IN事業部 CTO

- ALL-IN開発の方針決定、開発チームの取りまとめ
- RubyWorld Conference (2010, 2016)
- 表参道.rb 運営メンバー  
(次回11/8 弊社開催)
- その他大学院講師、書籍執筆の活動等



# CODE FOR 2030 TECH SUMMER CAMP

- 中高生をメインターゲットとして、マイ  
ンド、英語、プログラミング等のこれから  
を生き抜くための術（コード）をレク  
チャーする。
- 福島のホテルで2泊3日のキャンプ。  
(プログラミングに割り当てられた時間  
は約8時間)
- プログラミング未経験者が対象
- プログラミング分野の講師として、教材  
作成から担当
- 今回は「電子工作」をテーマにした

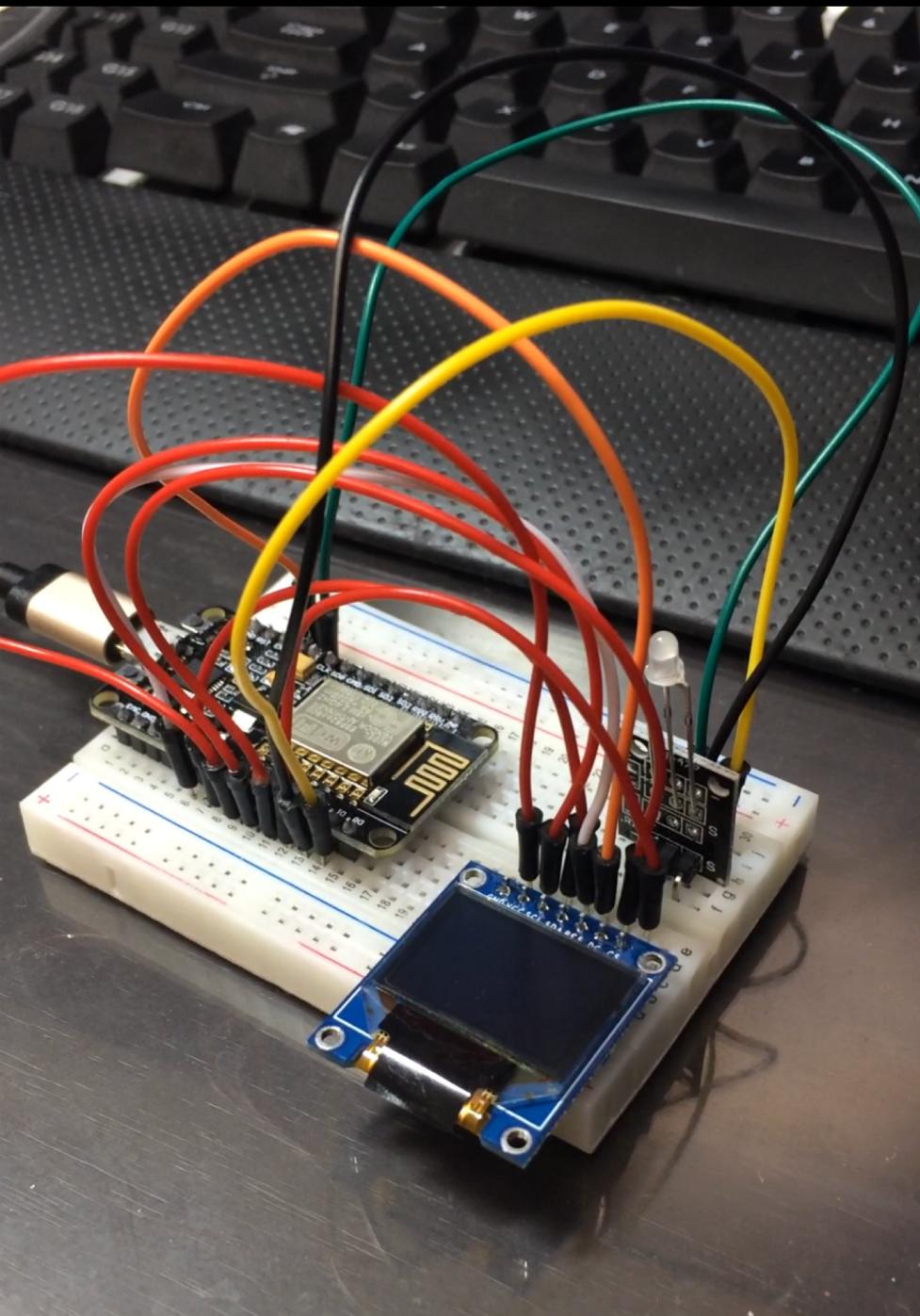
# 電子工作？

コンピュータにおける入出力が電気回路に電流が流れる／流れないので、仕組みが単純

ArduinoやRaspberry PI向けの教育用の入出力デバイスも安価に手に入るのと、基本が分かれば、他のデバイスにも適用できる。

使った教材は全部家に持ち帰ってもらうので、家に帰って続きをできるし、成果物が目に見えやすい

自分自身電子工作に興味があった

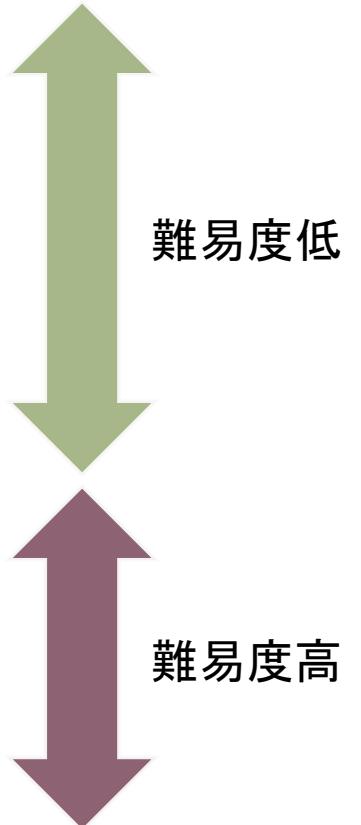


# 今回利用したデバイス

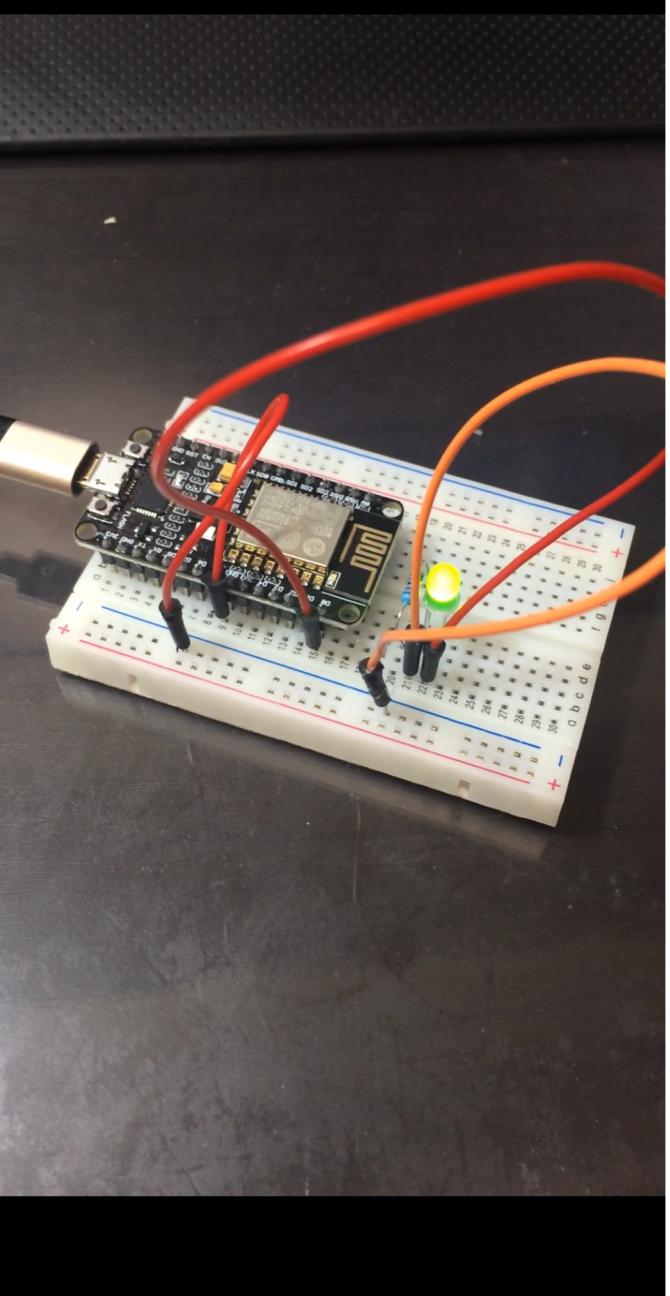
- NodeMCU-32S (ESP32S)
- Bluetooth ・ 無線LAN搭載、SRAM 520KB
- ブレッドボード上にマイコンボードと各種モジュールを配置する
- Arduinoでプログラミング

# 取り扱ったテーマ

1. 2色LEDを使ってLチカ
2. フルカラーLEDを使ってLチカ
3. タクトスイッチを使った入力制御
4. OLEDを使ったディスプレイの制御
5. 温度・湿度センサーから情報を取得し、ディスプレイに情報を表示する



# 2色LEDを使ってLチカ



```
#include <Arduino.h>

#define LED 7

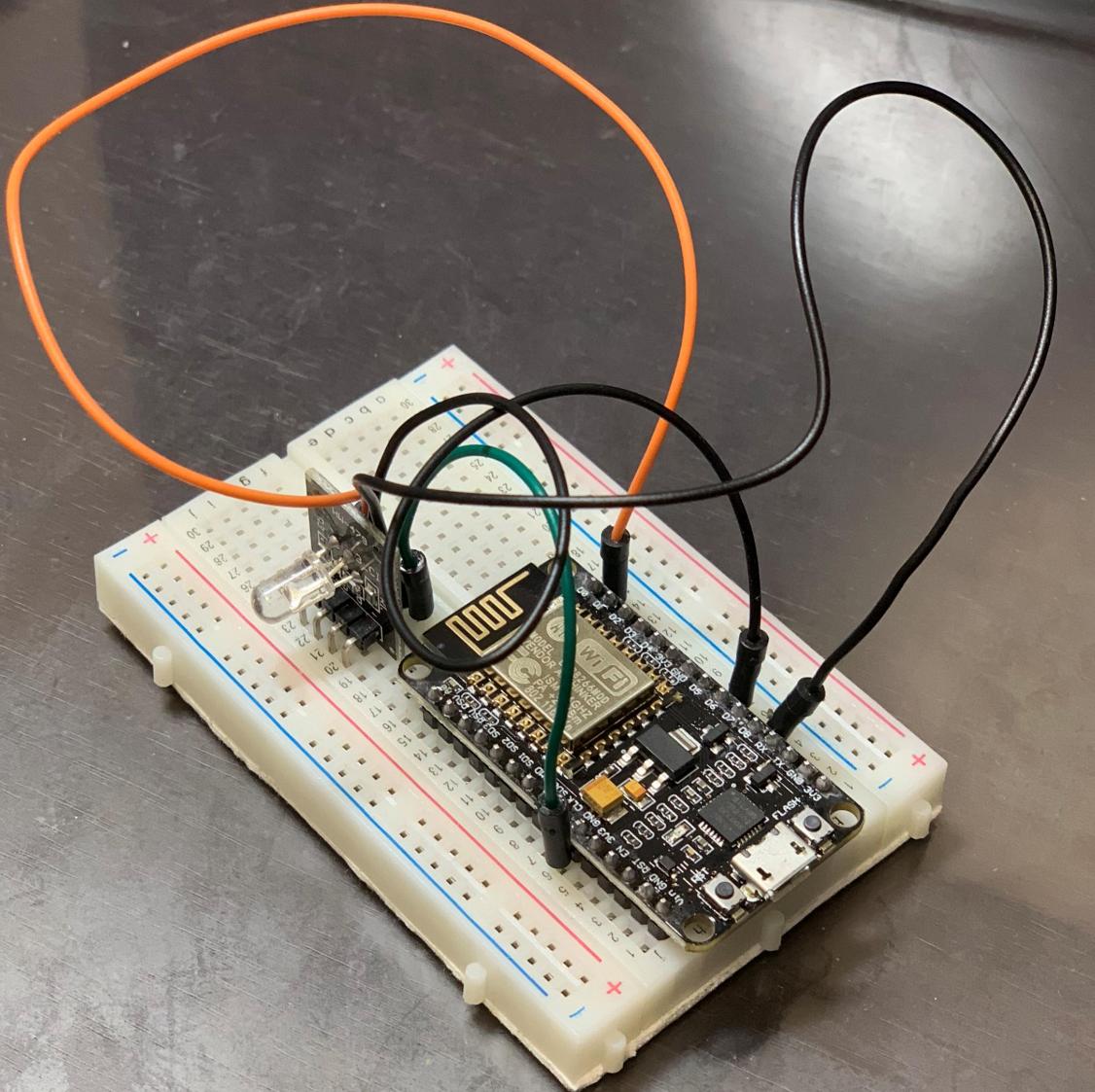
void setup()
{
    pinMode(LED, OUTPUT);
}

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

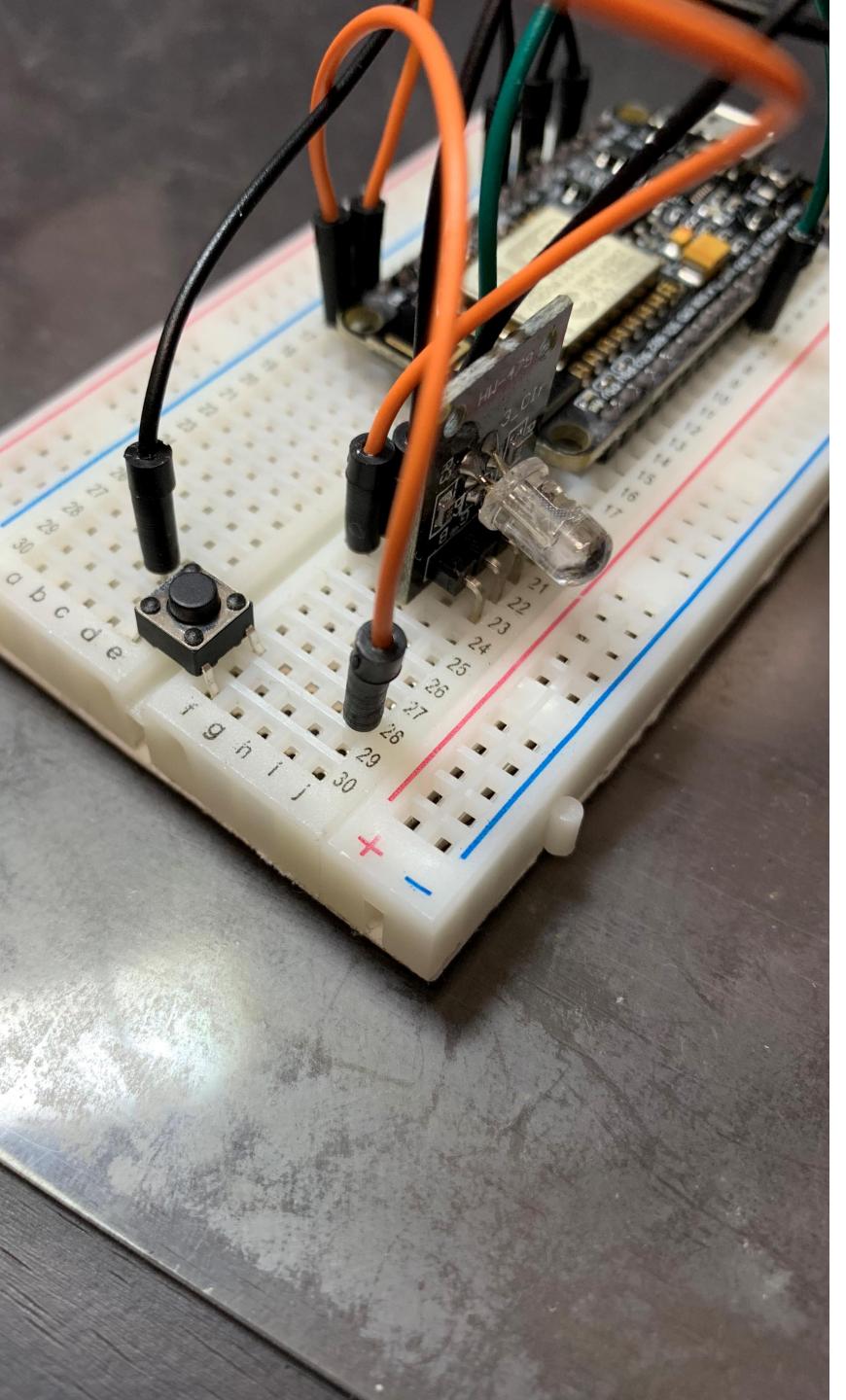
- 配線はシンプル
- 組み込み機器のHello World

# フルカラーLED を使ってLチカ

- RGBに1本ずつ、合計4本のピンを接続して、それぞれ異なる電圧をかけて、LEDを発光させる
- analogWriteというメソッドを使用し、かかる電圧をコントロールする。
- for文での繰り返しで発色が変わるので、ここで繰り返しを学ぶ



# タクトスイッチを使った入力制御

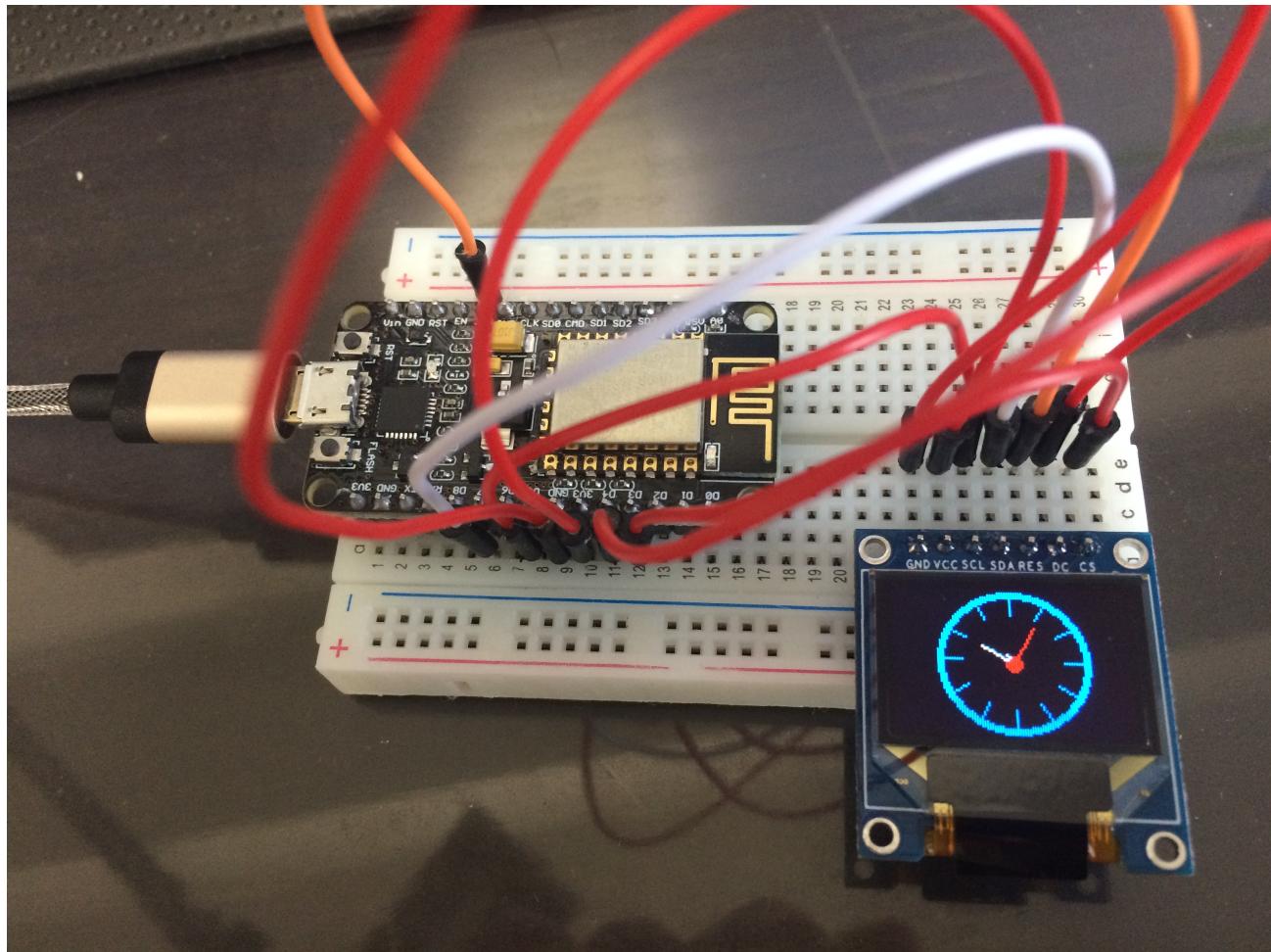


```
void setup() {  
    pinMode (0,OUTPUT);  
    pinMode(4,OUTPUT);  
    pinMode(5,OUTPUT);  
    pinMode(14,INPUT_PULLUP);  
}  
  
int flag;  
  
void loop() {  
    int r=random(1204);  
    int g=random(1203);  
    int b=random(1204);  
    int a=digitalRead(14);  
    if(a==LOW) {  
        if(flag==0){  
            flag=1;  
            delay(500);  
        }else{  
            flag=0;  
            delay(500);  
        }  
    }  
  
    if(flag==0){  
        r=0;  
        g=0;  
        b=0;  
    }  
    analogWrite(0,g);  
    analogWrite(4,r);  
    analogWrite(5,b);  
    delay(5);  
}
```

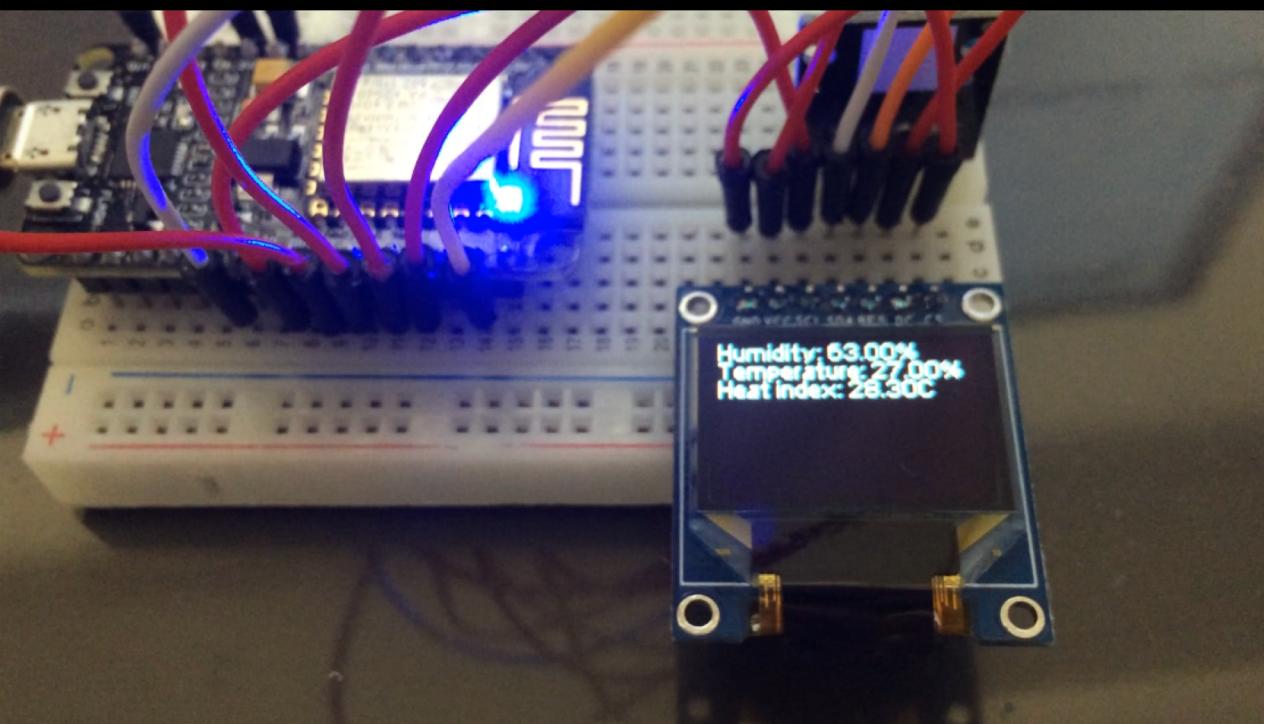
- ・ タクトスイッチを配線して、LEDのON・OFF制御を行う
- ・ フラグ管理とif文について学ぶ

# OLEDを使ったディスプレイの制御

- SSD1331というOLEDディスプレイを制御する
- 制御用のライブラリをプログラムに組み込む。
- 7本の線を配線する。この部分はどう繋げばいいかをレクチャーする。
- ライブラリのインストールや、ディスプレイの初期化作業など、プログラムの本質に関係ない部分が多く、難易度が高い。
- この辺りからmruby/cの適用を考え始めた。



## 温度・湿度センサーから情報を取り得し、ディスプレイに情報を表示する



- SSD1331に加え、DHT11という温度・湿度センサーを制御する
- DHT11の温度・湿度に関する入力信号を数値に変換する部分はライブラリに任せる
- 配線が複雑で10本のピンを使用する。
- DHT11も初期化コードなど、本質とは関係なく書かないといけないコードが多い。

# mruby/cの適用

mrbcでrubyのコードを  
バイトコードに変換して、  
プログラムにコピー&  
ペーストする

コンパイル済みmrubyバイトコード



mruby/c  
initialize code

mruby/c ⇄ library bridge  
(C++)

OLED用  
Library

各種センサ  
Library

mruby/c ⇄ Arduino bridge  
(C++)

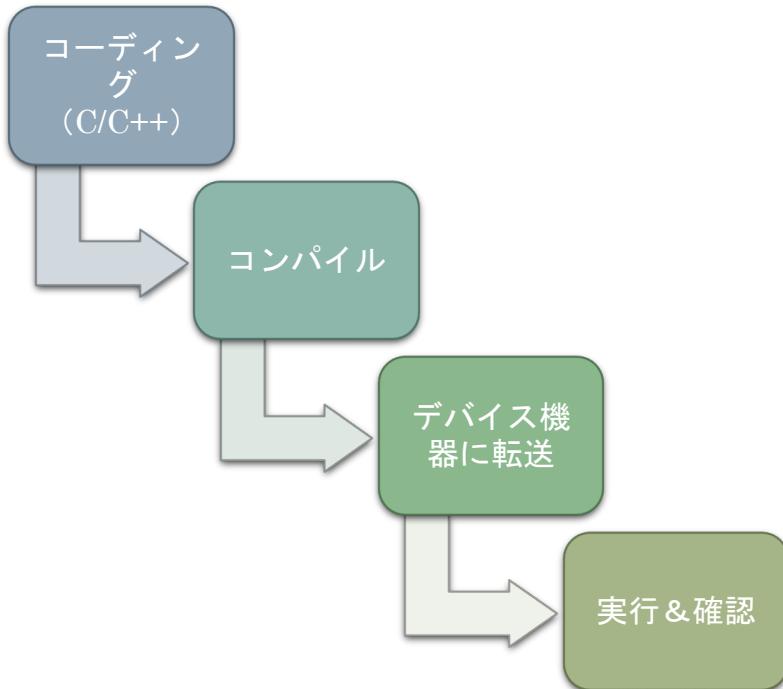
コンパイルしてできた  
バイナリをデバイス機  
器に転送する

ESP32S for Arduino Core

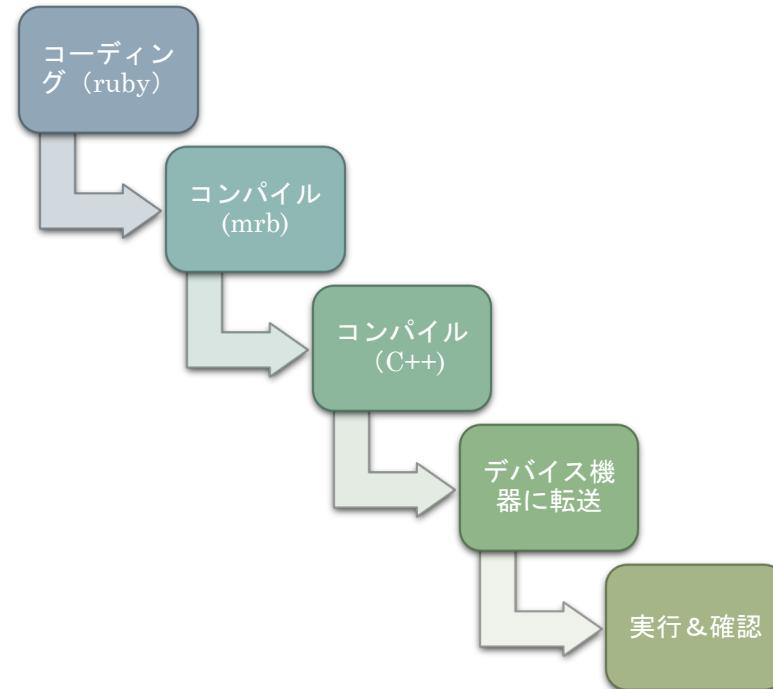
ESP32S

# mruby/cを使うことで逆に手順が増える？

従来方式



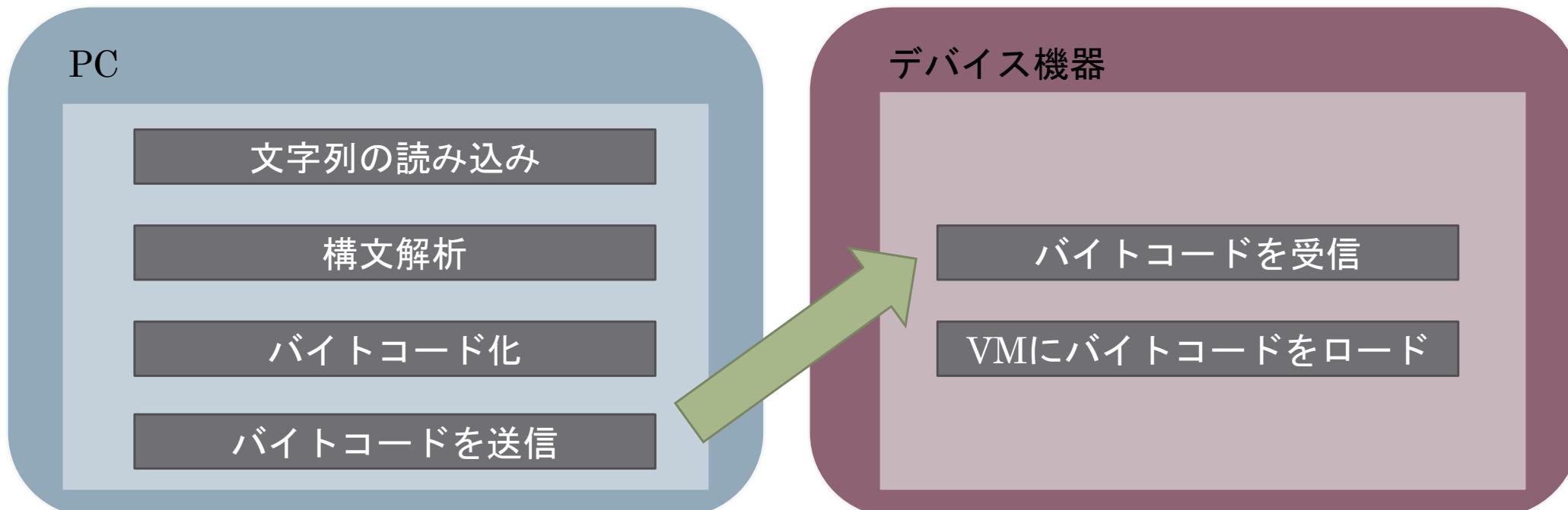
mruby/c



より ruby らしく手軽に動かすことはできないか？

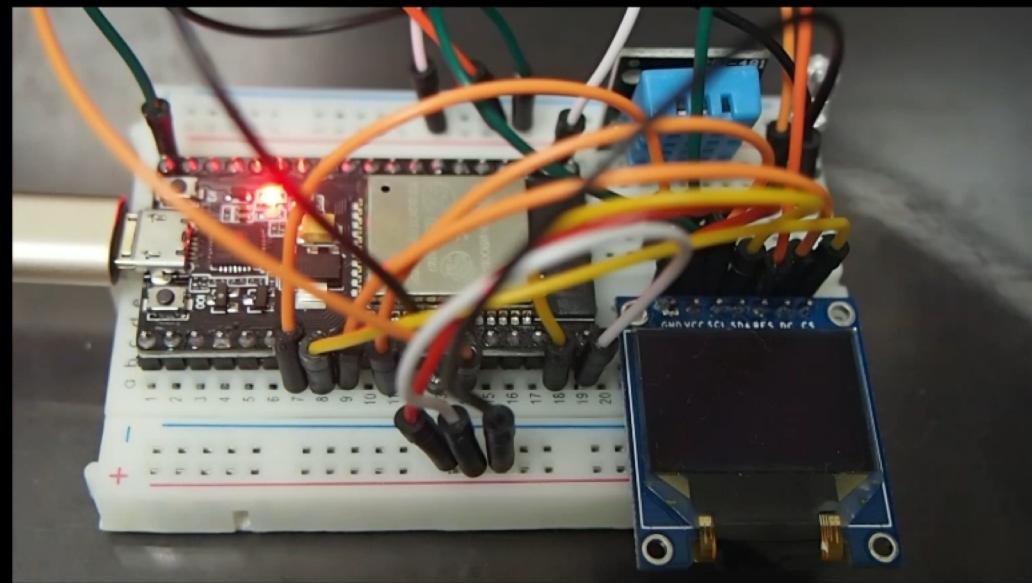
# rmirbとの出会い

- M5Stack meetupにて@kishimaさんが発表
- mruby上のmirbをリモート対応にしたもので、デバイス機器側でTCP/IPで待ち受けて、mirbで逐次バイトコードに変換したものを、デバイス側に送信して実行できる



C:\Users\kirik\Documents\mruby\bin>\_

Lチカ



## 適用結果

- 実は適用できていない
- mruby/c側でループが上手く動作しないバグがあった
- 最終日にバグが解決したので、デモを披露した
- 用意したお題は、全員Cで頑張って書いた。全員Cで苦労したので、とても反響があった。
- 会場のネットワーク環境に左右されたり、まだまだ課題がある





# 今後の話

- ・教材として提供したセンサーデバイスすべてを  
mruby/cから取り扱えるようにする  
<https://github.com/kirikak2/nodemcu32s-meets-mrubbyc>
- ・手間のかからない開発環境（obnizを参考にしている）

# Special Thanks

- @kishimaさん

今回の適用に当たり、多大なご協力をいただき、心より感謝しております。

- ALL-IN開発チーム

ALL-IN開発と並行して行っていたプロジェクトで、タイトなスケジュールの中、精神的支えになってくれたことに対して感謝しています。

- TECH SUMMER CAMP主催グループ

チャレンジングな機会を与えていただいたことに心より感謝いたします。

