

# mbed入門

質問: [tech-support-m1@ogata-lab.jp](mailto:tech-support-m1@ogata-lab.jp)

## mbedとarduinoの違い

- CPUアーキテクチャ (コンパイルしたバイナリファイルの互換性がない。)
  - arduino . . . ATMELのAVRマイコン
  - mbed . . . NXP社のLPCマイコン(ARM Cortex M3)
- 開発環境 (ソースコードやライブラリも互換性がほとんどない)
- ピン数 (ピンが多いとサイズも大きく、消費電力も大きい傾向)
- 動作電圧
  - arduino . . . 5V系、3.3V系のarduinoもある
  - mbed . . . 3.3V系

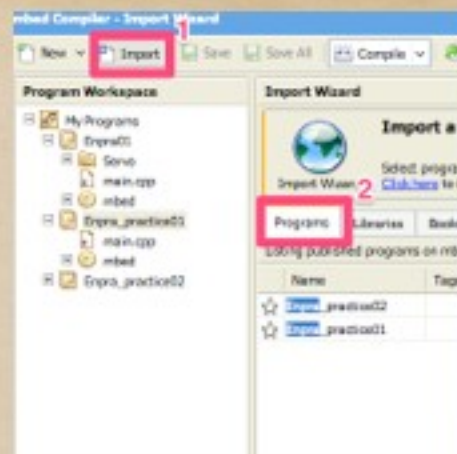
# mbed 開発環境

- ◆ クラウドコンパイル (ウェブ上にある開発環境内ですべて可能)
  - ◆ ソースコード管理
  - ◆ ライブラリ管理・Import/Export
  - ◆ コンパイル

## import

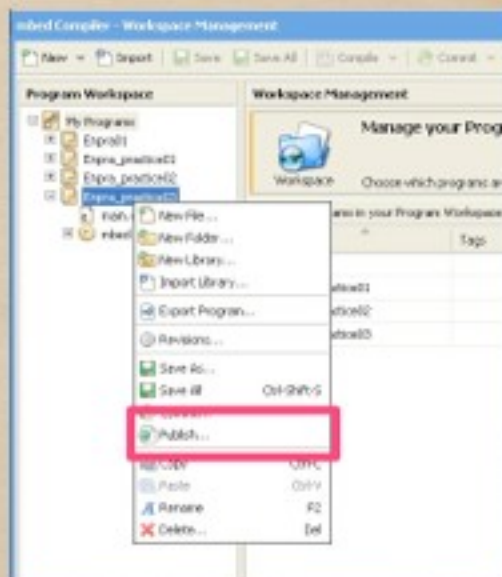
誰かが作ったプログラムをimport  
できる。

Search >> "Enpra"



# publish

自分の作成したプログラムを公開  
できる



## LEDs

Dの $V_f$ は 2.0[V] (Red)

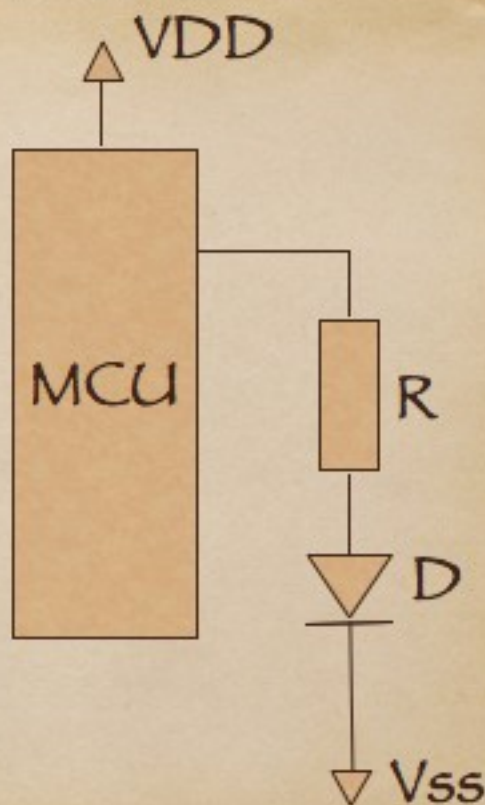
$$I = (V_{dd} - V_f) / R \approx 10 \text{ mA} \text{ くらい}$$

Pin = HIGH ==> LED = ON

Pin = LOW ==> LED = OFF

mbed上のLEDはこれ

Enpra\_practice01 / 02



```

#include "mbed.h"

DigitalOut leds[4] = (LED1, LED2, LED3, LED4);

int main() {
    while() {
        // Turn on LEDs sequentially.5
        for (int i = 0; i < 4; i++) {
            leds[i] = 1;
            wait(0.5);
        }
        // Turn off LEDs.
        for (int i = 0; i < 4; i++) {
            leds[3-i] = 0;
            wait(0.5);
        }
    }
}

```

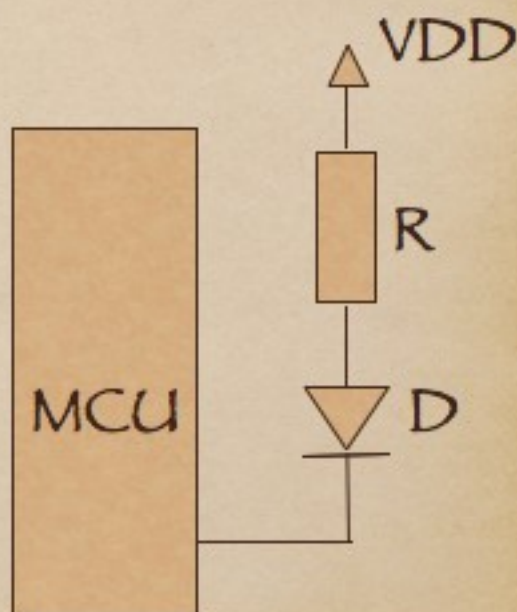
## LEDs

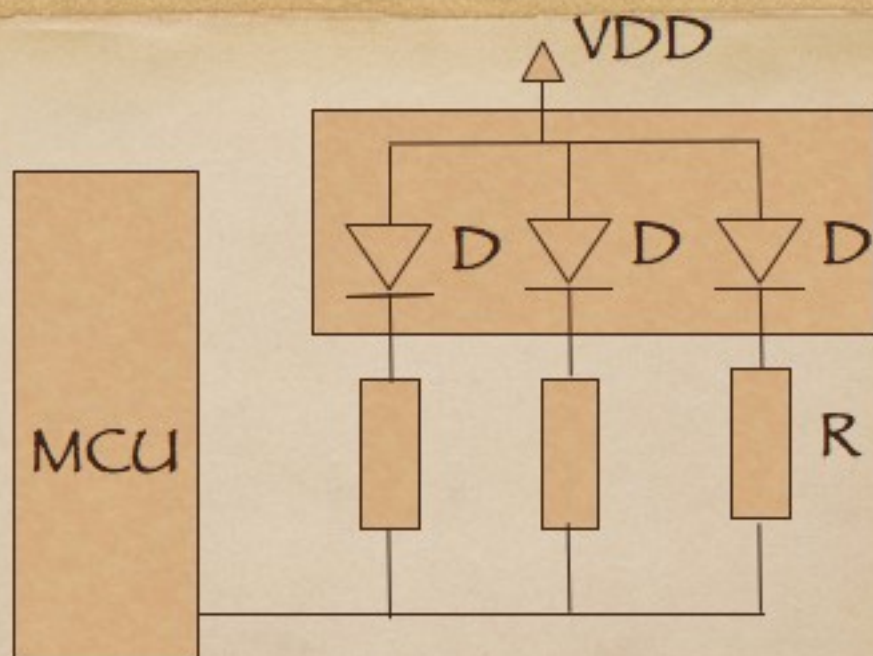
DのVfは 2.0[V] (Red)

$$I = (V_{dd} - V_f) / R \approx 10 \text{ mA} < \text{らい}$$

Pin = HIGH ==> LED = OFF

Pin = LOW ==> LED = ON





## Fullcolor LED

```
#include "mbed.h"
#include "HSVtoRGB.h"

PwmOut r(p21), g(p22), b(p23);

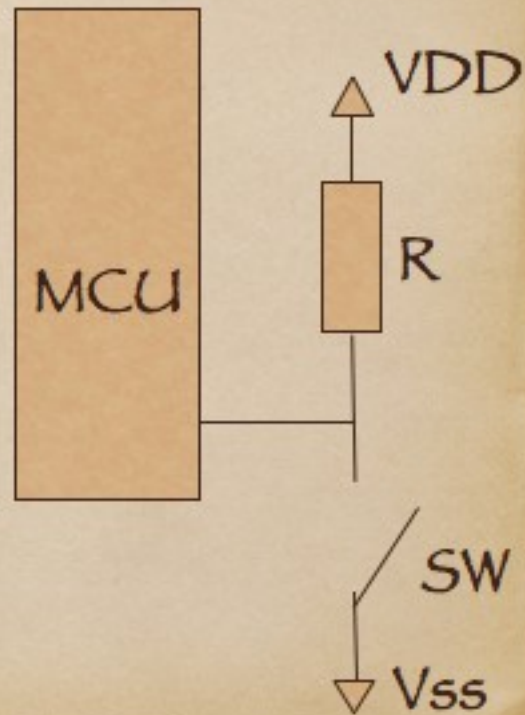
int main() {
    r.period(0.02);
    g.period(0.02);
    b.period(0.02);
    while(1) {

        for(int i = 0; i < 100; i++) {
            HSV hsv;
            hsv.h = 3.6 * i;
            hsv.s = 255;
            hsv.v = 255;
            RGB rgb = HSV2RGB(hsv);
            r.write(1.0f - rgb.r / 255.0f);
            g.write(1.0f - rgb.g / 255.0f);
            b.write(1.0f - rgb.b / 255.0f);
            wait(0.1);
        }
    }
}
```

# SWITCHES

ON ==> Vss (0 V)

OFF ==> VDD

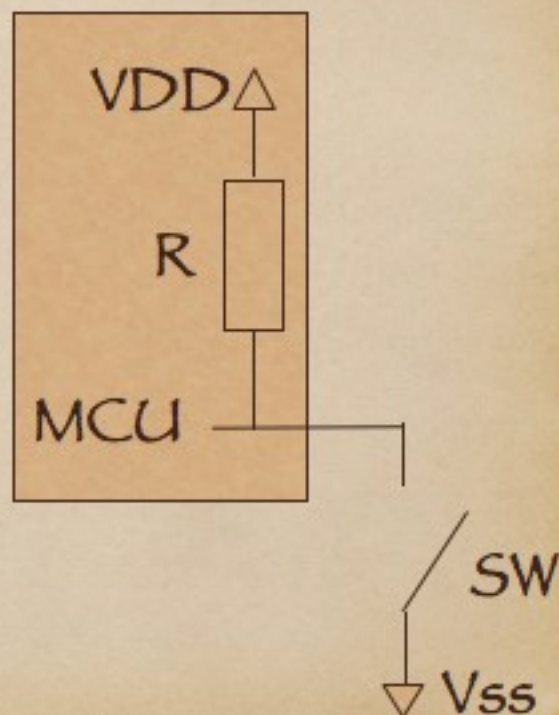


# Internal Pullup

ソフトウェアで設定可能.

部品点数を減らせる

Enpra\_practice02



```

#include "mbed.h"

DigitalOut led1(LED1);

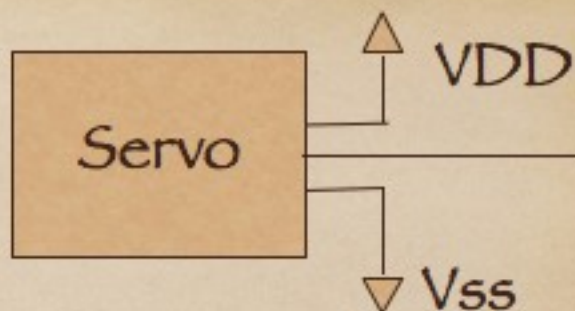
// Pin name can also be used to initialize DigitalIn and DigitalOut.
DigitalIn sw1(p20);

int main() {
    // Inner pull up on
    sw1.mode(PullUp);

    while(1) {
        if (sw1 == 0) { // If Pushed
            led1 = 1;
        } else { // Else (if not pushed)
            led1 = 0;
        }
    }
}

```

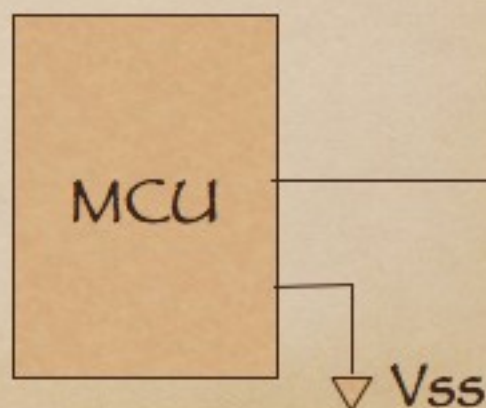
SERVOs



Import "Servo"  
(使用数の多いヤツが安全)

※ Vssを共通にする！！

Enpra\_practice04



```
#include "mbed.h"
#include "Servo.h"

DigitalIn sw01(p20);
DigitalOut led01(LED1);
Servo servo01(p20);

bool pose = false;
bool old_sw = false;

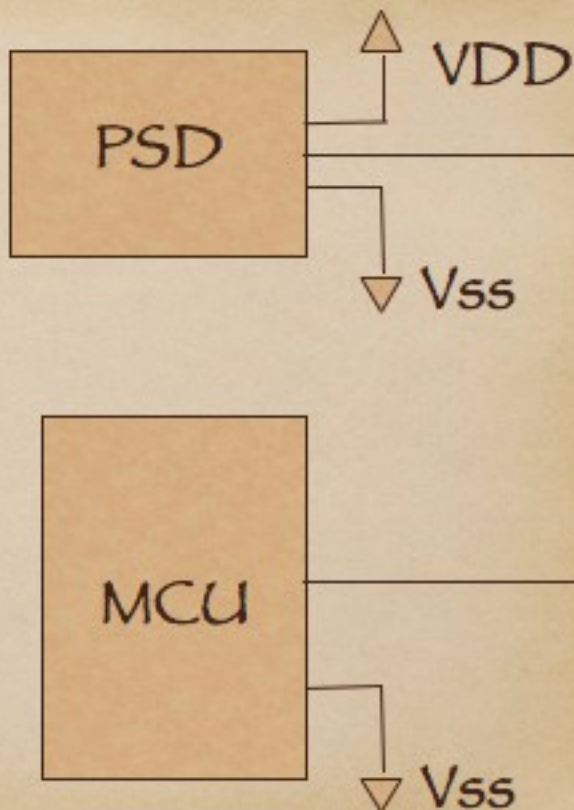
int main() {
    sw01.mode(PullUp);
    while(1) {
        bool now_sw = (sw01 == 0);
        if (now_sw) {
            led01 = 1;
        } else {
            led01 = 0;
        }
    }
}
```

```
if (old_sw && !now_sw) {
    pose = !pose;
}
old_sw = now_sw;
if (pose) {
    servo01.write(0.5);
} else {
    servo01.write(0.0);
}
}
```

PSDs  
(Analog In)

※ Vssを共通にする！！

Enpra\_practice05





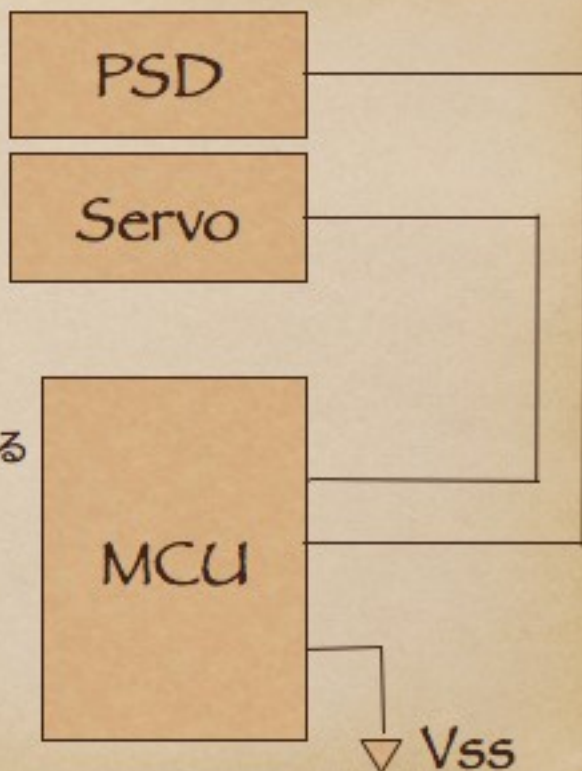
```
#include "mbed.h"

AnalogIn ain01(p19);
DigitalOut leds[4] = (LED1, LED2, LED3, LED4);

int main() {
  while() {
    int i;
    for (j = 0; j < 4 * ain01; j++) {
      leds[j] = 1;
    }
    for (j < 4; j++) {
      leds[j] = 0;
    }
  }
}
```

# Exercise

2つのコードを足して一つにする



```
#include "mbed.h"
```

```
#include "Servo.h"
```

```
DigitalIn sw01(p20);
```

```
DigitalOut led01(LED1);
```

```
Servo servo01(p21);
```

```
AnalogIn analog01(p19);
```

```
bool pose = false;
```

```
bool old_sw = false;
```

```
int main() {
```

```
    sw01.mode(PullUp);
```

```
    while() {
```

```
        bool now_sw = (sw01 == 0);
```

```
        if (now_sw) {
```

```
            led01 = 1;
```

```
        } else {
```

```
            led01 = 0;
```

```
        }
```

```
        if (old_sw && now_sw) {  
            pose = !pose;
```

```
        }
```

```
        old_sw = now_sw;
```

```
        if (pose) {
```

```
            servo01.write(analog01);
```

```
        } else {
```

```
            servo01.write(0.1);
```

```
        }
```

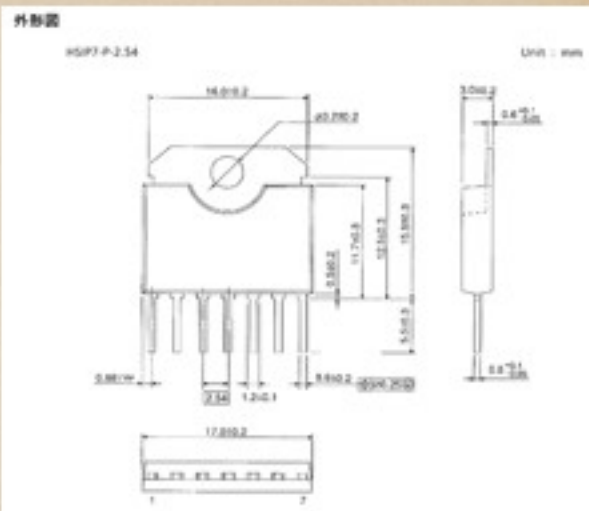
```
    }
```

```
}
```

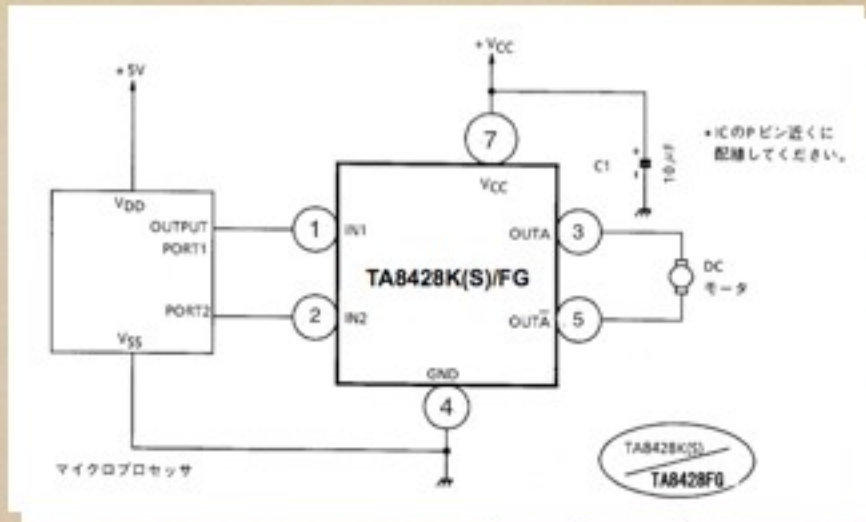
# Motor Driver

TA8428S

ICの型番の末尾は、  
パッケージの形状や  
精度等級を表す



Enpra\_practice06



マイコンから IN1, IN2ピンを  
制御することで、  
モータを正転, 逆転できる。

IN1	IN2	
L	L	ブレーキ
L	H	CCW
H	L	CW
H	H	ハイインピーダンス

```

#include "mbed.h"
#include "TA8428.h"

DigitalIn sw01(p19);
TA8428 driver(p21, p22);

bool old_sw = false;
bool direction_cw = true;

int main() {
    while(1) {
        bool new_sw = sw01 == 0;
        if (old_sw && new_sw) {
            direction_cw = !direction_cw;
        }
        old_sw = new_sw;

        if (direction_cw) {
            driver.drive(0.5);
        } else {
            driver.drive(-0.5);
        }
    }
}

```

# Serial Communication

Windowsの場合はドライバのインストールが必要

<http://mbed.org/handbook/Windows-serial-configuration>

PC, Mac

mbed

USB

```
graph LR; PC[PC, Mac] --- USB[USB]; USB --- mbed[mbed];
```

# Serial Communication

PC, Mac側はコンソール起動  
ConsoleOutput: 文字列を表示  
ConsoleInput: キーボード入力

<http://mbed.org/handbook/Terminals>

PC, Mac

mbed

USB

```
graph LR; PC[PC, Mac] --- USB[USB]; USB --- mbed[mbed];
```

# Console Output

Enpra\_practice07

```
#include "mbed.h"
#include "Serial.h"

Serial usbSerial(USBTX, USBRX);

int main() {
    usbSerial.printf("Hello World!");
    int i = 0;
    while(1) {
        usbSerial.printf("Counter = %d\n", i);
        i++;
    }
}
```

# Console Input

Enpra\_practice08

```
#include "mbed.h"
#include "Serial.h"

Serial usbSerial(USBTX, USBRX);

DigitalOut leds[4] = (LED1, LED2, LED3, LED4);

int main() {
    usbSerial.printf("Hello World!");
    while(1) {
        char c = usbSerial.getc();
        switch(c) {
            case '1':
                leds[0] = 1;
                break;
            case '2':
                leds[1] = 1;
                break;
            case '3':
                leds[2] = 1;
                break;
            case '4':
                leds[3] = 1;
                break;
            default:
                leds[0] = 0;
                leds[1] = 0;
                leds[2] = 0;
                leds[3] = 0;
                break;
        }
    }
}
```

## 課題：シリアル通信でLEDを制御

- ◆ ConsoleInputのプログラムの動作確認をしてこること

## 課題：下記の言葉について調査

- ◆ 「プロトコル」とは何か。
  - ◆ なぜ「プロトコル」が必要なのか
  - ◆ mbed型でロボットを作り、PCと通信する場合を例にとって答えよ