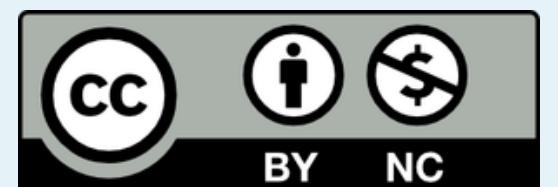


# コードとデザイン

東京藝術大学 芸術情報センター開設科目 金曜4-5限 第6週

2023.05.12 松浦知也 ([matsura.tomoya@noc.geidai.ac.jp](mailto:matsura.tomoya@noc.geidai.ac.jp) [teach@matsuuratomo.ya.com](mailto:teach@matsuuratomo.ya.com))



# 本日のスケジュール

- 質問コーナー：二進数からの出力（15分）
- Arduinoの入門

- 二進数で計算したものを最終的にどうやって人間の読めるフォーマットで出力する？



7セグデコーダーIC

**TOSHIBA****TC4511BP/BF****真理値表**

LE	$\bar{B}I$	$\bar{LT}$	Inputs			Outputs							Display Mode
			D	C	B	A	a	b	c	d	e	f	
*	*	L	*	*	*	*	H	H	H	H	H	H	8
*	L	H	*	*	*	*	L	L	L	L	L	L	Blank
L	H	H	L	L	L	L	H	H	H	H	H	H	0
L	H	H	L	L	L	H	L	H	H	L	L	L	1
L	H	H	L	L	H	L	H	H	L	H	H	L	2
L	H	H	L	L	H	H	H	H	H	L	L	H	3
L	H	H	L	H	L	L	H	H	L	L	H	H	4
L	H	H	L	H	L	H	H	L	H	L	H	H	5
L	H	H	L	H	H	L	L	H	H	H	H	H	6
L	H	H	L	H	H	H	H	H	L	L	L	L	7
L	H	H	H	L	L	H	H	H	H	H	H	H	8
L	H	H	H	L	L	H	H	H	L	L	H	H	9
L	H	H	H	L	H	L	L	L	L	L	L	L	Blank
L	H	H	H	L	H	H	L	L	L	L	L	L	Blank
L	H	H	H	H	*	*	L	L	L	L	L	L	Blank
H	H	H	*	*	*	*	$\Delta\Delta$						

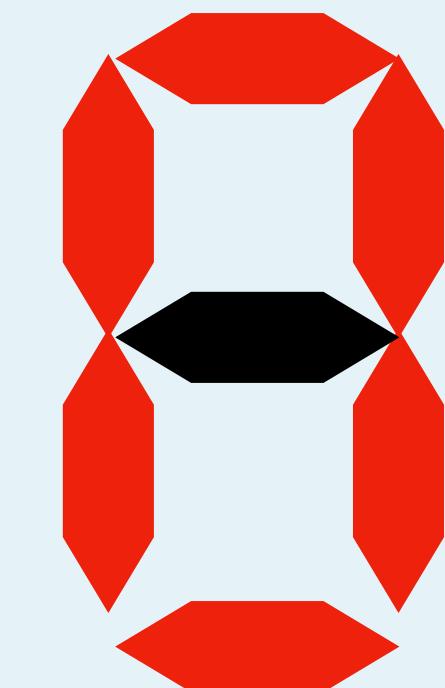
\*: Don't care

 $\Delta\Delta$ : Depends upon the BCD code previously applied when LE "L"

**TOSHIBA****TC4511BP/BF****真理値表**

LE	$\bar{B}I$	$\bar{LT}$	Inputs			Outputs							Display Mode
			D	C	B	A	a	b	c	d	e	f	
*	*	L	*	*	*	*	H	H	H	H	H	H	8
*	L	H	*	*	*	*	L	L	L	L	L	L	Blank
L	H	H	L	L	L	L	H	H	H	H	H	L	0
L	H	H	L	L	L	H	L	H	H	L	L	L	1
L	H	H	L	L	H	L	H	H	L	H	H	L	2
L	H	H	L	L	H	H	H	H	H	L	L	H	3
L	H	H	L	H	L	L	H	H	L	L	H	H	4
L	H	H	L	H	L	H	L	H	H	L	H	H	5
L	H	H	L	H	H	L	L	H	H	H	H	H	6
L	H	H	L	H	H	H	H	H	L	L	L	L	7
L	H	H	H	L	L	H	H	H	H	H	H	H	8
L	H	H	H	L	L	H	H	H	L	L	H	H	9
L	H	H	H	L	H	L	L	L	L	L	L	L	Blank
L	H	H	H	L	H	H	L	L	L	L	L	L	Blank
L	H	H	H	H	*	*	L	L	L	L	L	L	Blank
H	H	H	*	*	*	*	$\Delta\Delta$						

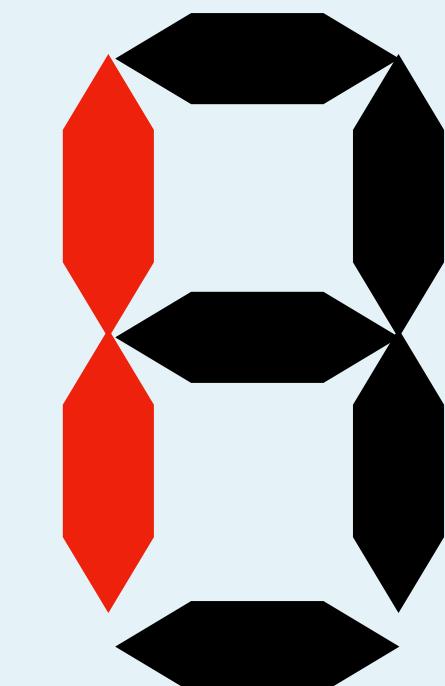
\*: Don't care

 $\Delta\Delta$ : Depends upon the BCD code previously applied when LE "L"

**TOSHIBA****TC4511BP/BF****真理値表**

LE	$\bar{B}I$	$\bar{LT}$	Inputs			Outputs							Display Mode
			D	C	B	A	a	b	c	d	e	f	
*	*	L	*	*	*	*	H	H	H	H	H	H	8
*	L	H	*	*	*	*	L	L	L	L	L	L	Blank
L	H	H	L	L	L	L	H	H	H	H	H	H	0
L	H	H	L	L	L	H	L	H	H	L	L	L	1
L	H	H	L	L	H	L	H	H	L	H	L	H	2
L	H	H	L	L	H	H	H	H	H	L	L	H	3
L	H	H	L	H	L	L	H	H	L	L	H	H	4
L	H	H	L	H	L	H	L	H	H	L	H	H	5
L	H	H	L	H	H	L	L	H	H	H	H	H	6
L	H	H	L	H	H	H	H	H	L	L	L	L	7
L	H	H	H	L	L	H	H	H	H	H	H	H	8
L	H	H	H	L	L	H	H	H	L	L	H	H	9
L	H	H	H	L	H	L	L	L	L	L	L	L	Blank
L	H	H	H	L	H	H	L	L	L	L	L	L	Blank
L	H	H	H	H	*	*	L	L	L	L	L	L	Blank
H	H	H	*	*	*	*	$\Delta\Delta$						

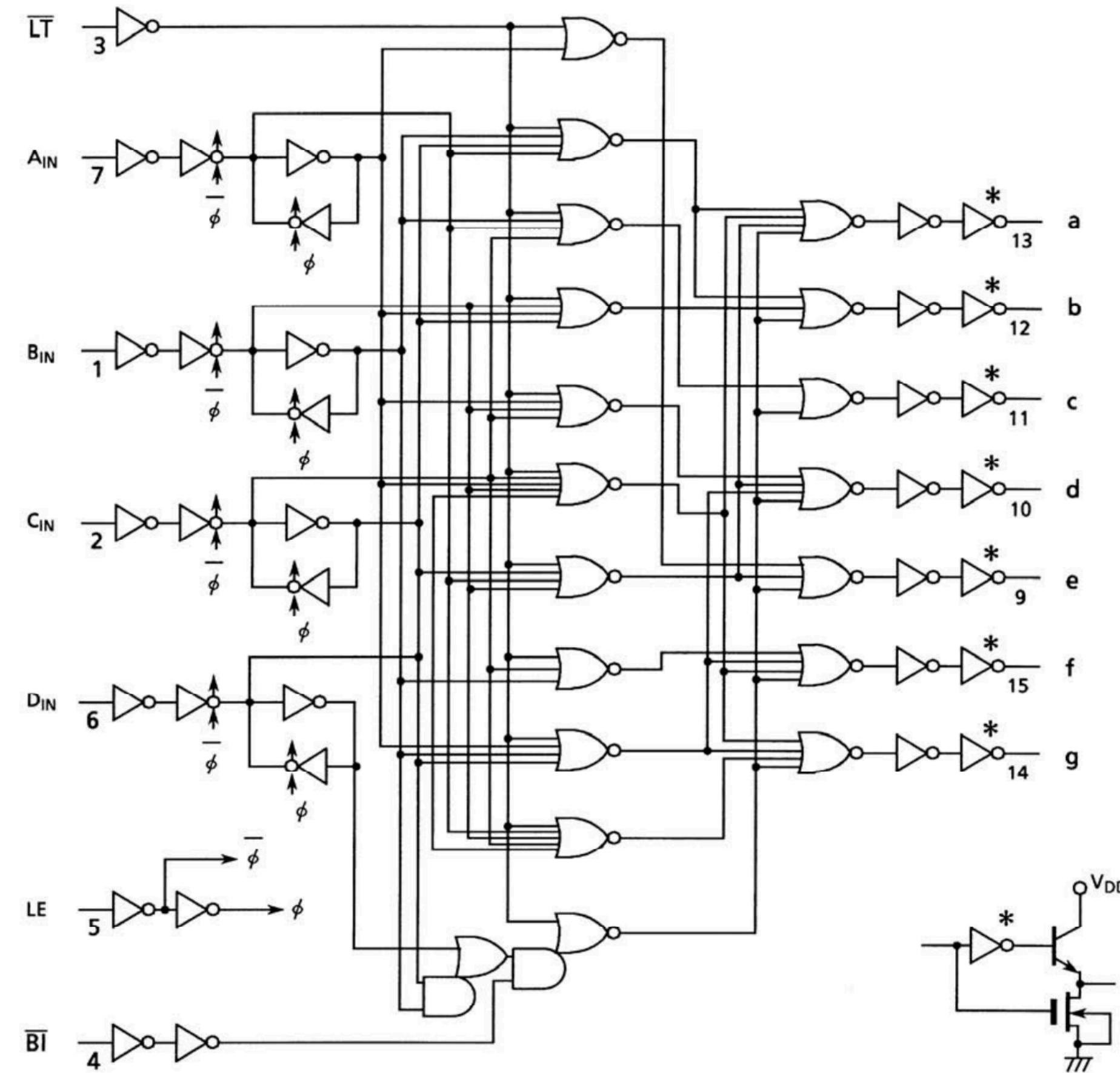
\*: Don't care

 $\Delta\Delta$ : Depends upon the BCD code previously applied when LE "L"

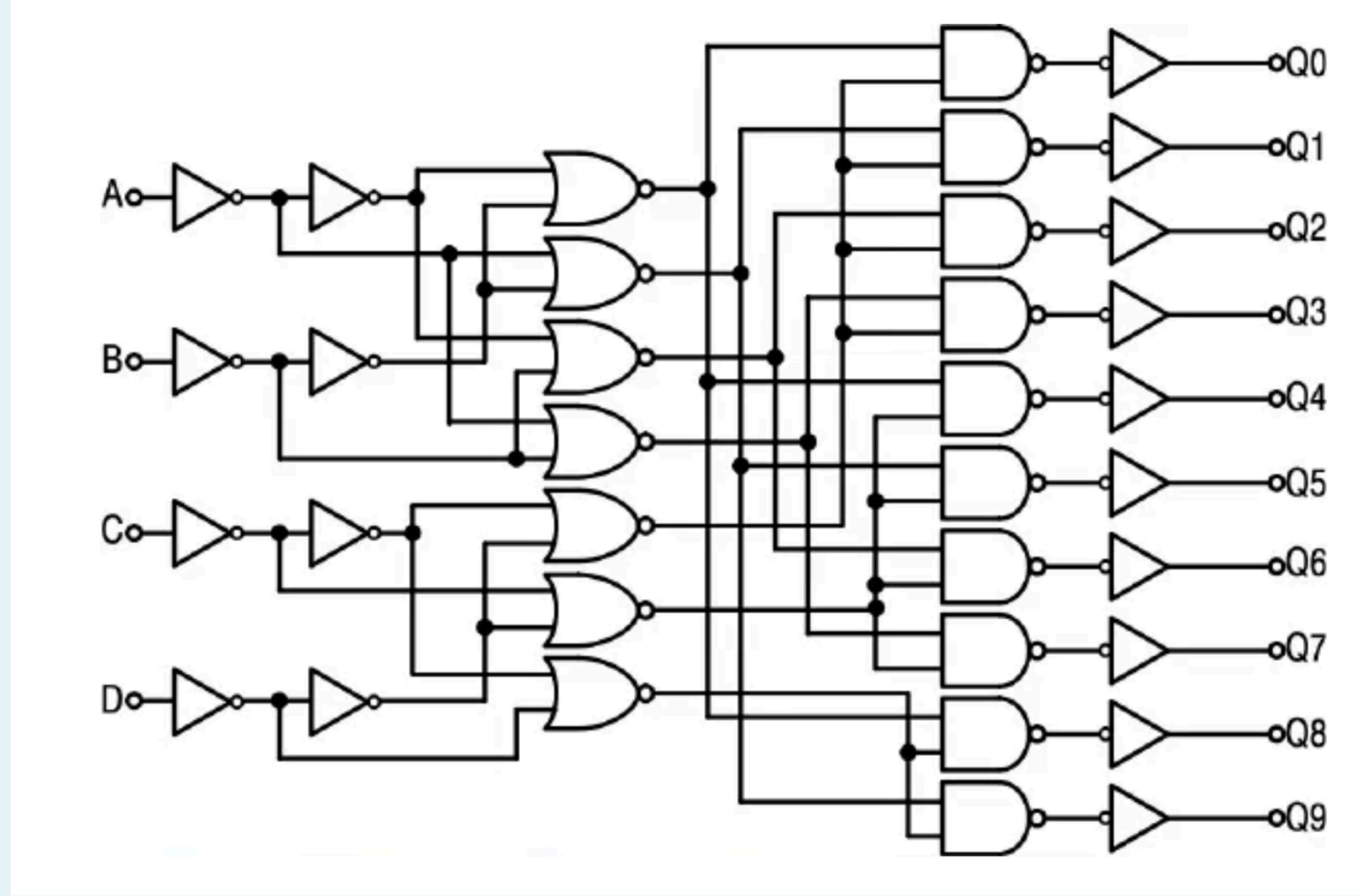
**TOSHIBA**

**TC4511BP/BF**

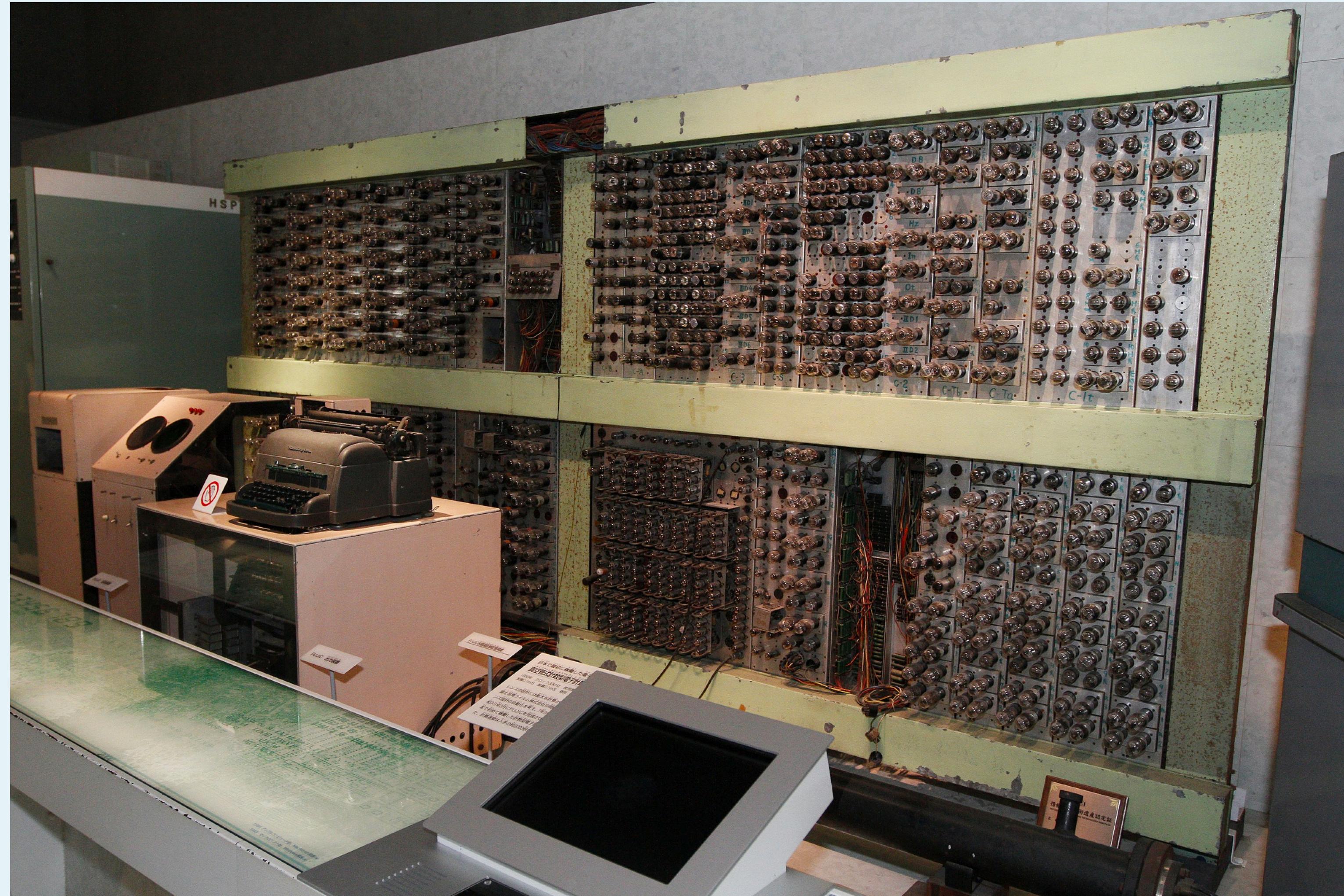
論理図



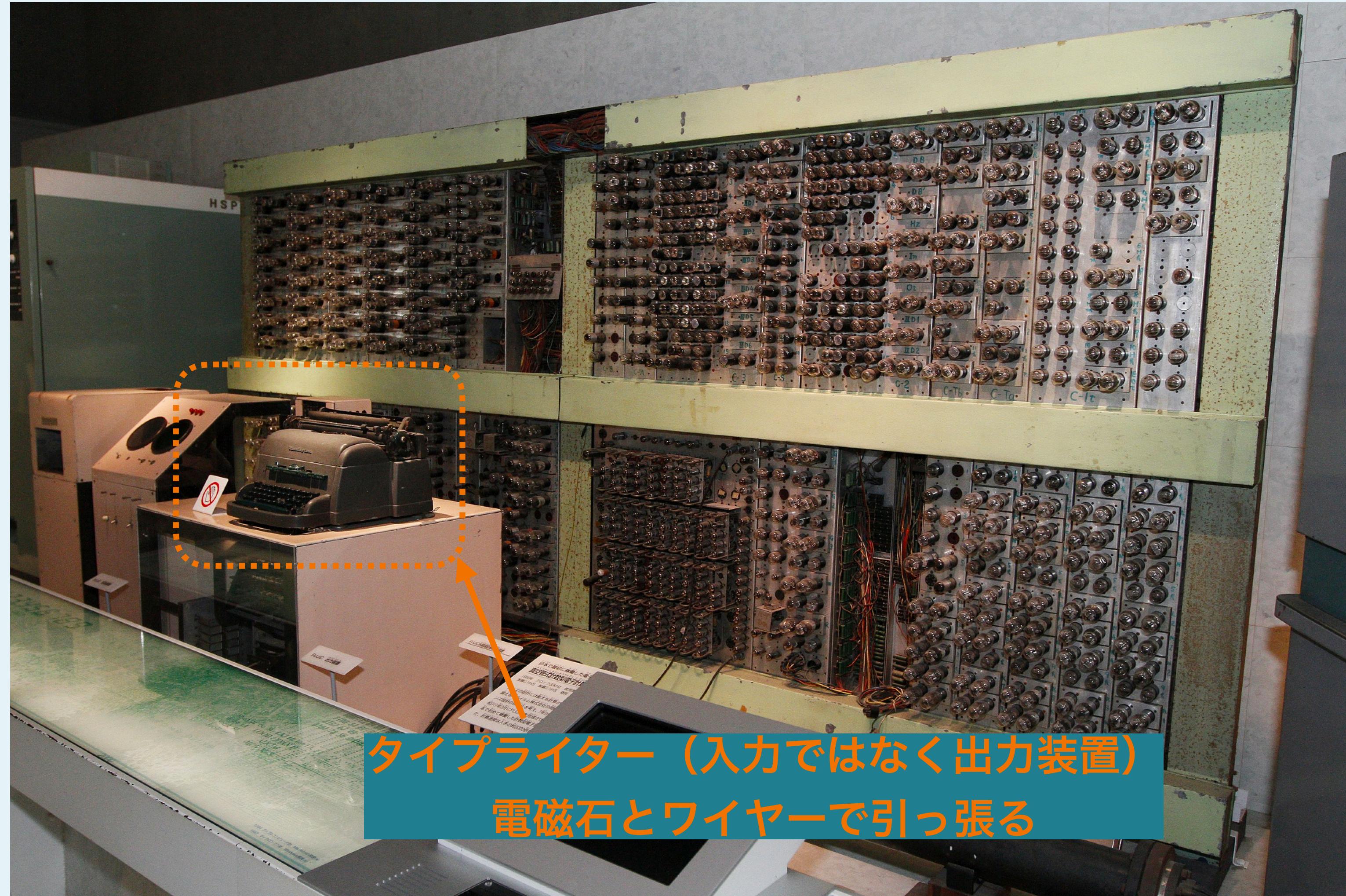
TOSHIBA TC4511BP/BF データシート(2006)より



4-Bit Binary To Decimal Decoder (BDD), Harsh Burwar(2008) <https://www.scribd.com/doc/23119807/4-bit-Binary-to-Decimal-Decoder-BDD> より。



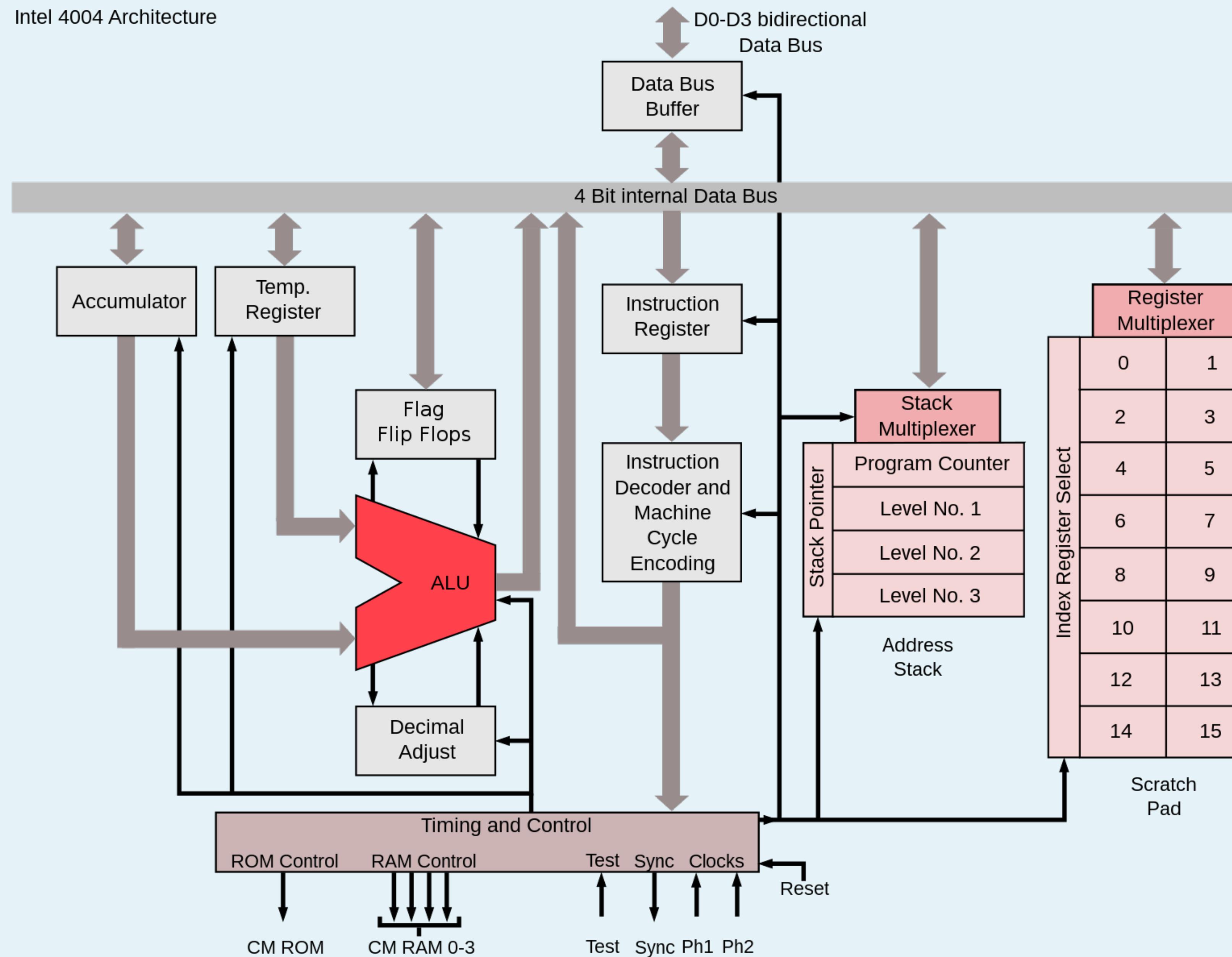
FUJIC(日本で最初のコンピューター、上野の科学博物館で展示されています)



FUJIC(日本で最初のコンピューター、上野の科学博物館で展示されています)

# 電卓とコンピューターの違い

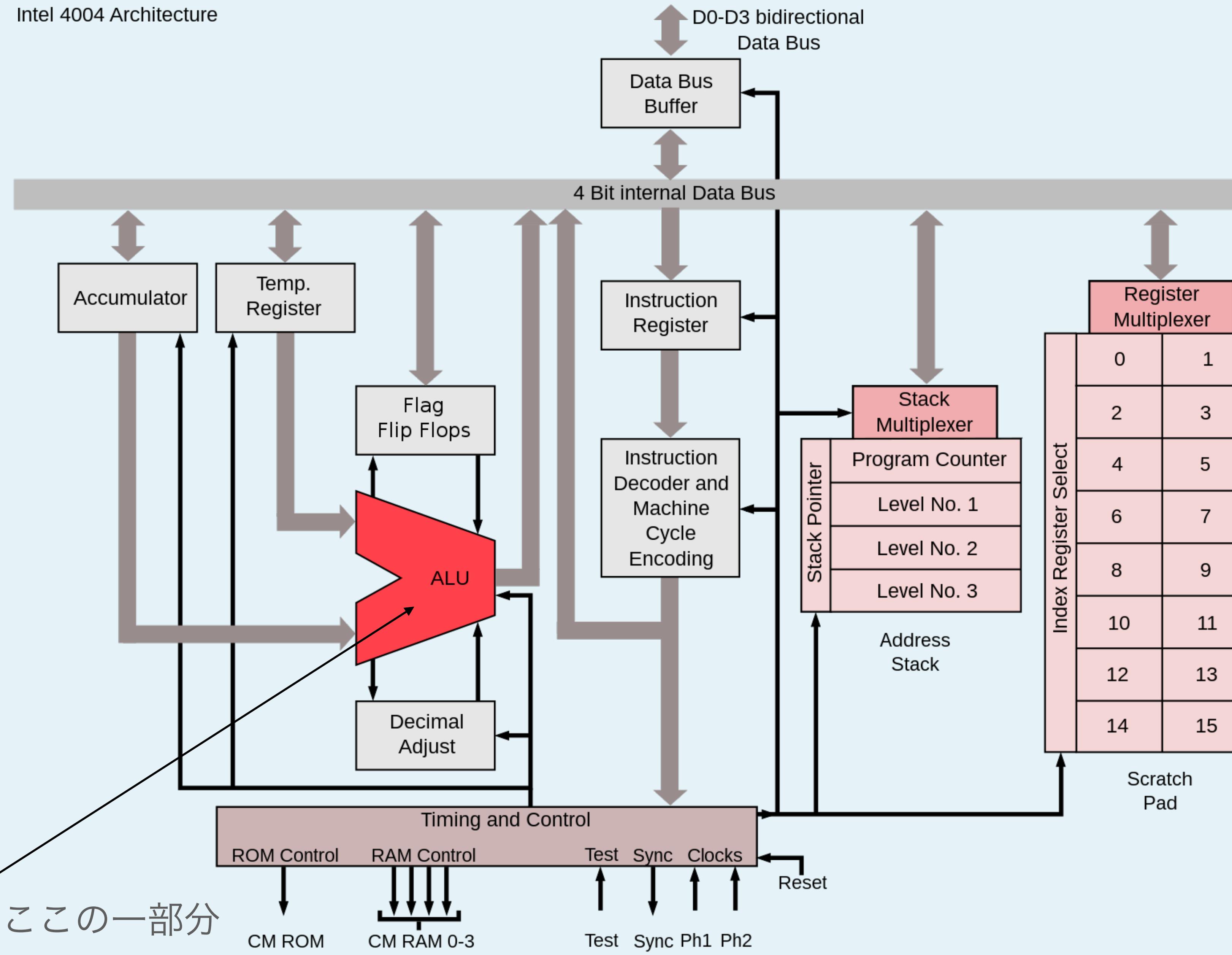
Intel 4004 Architecture



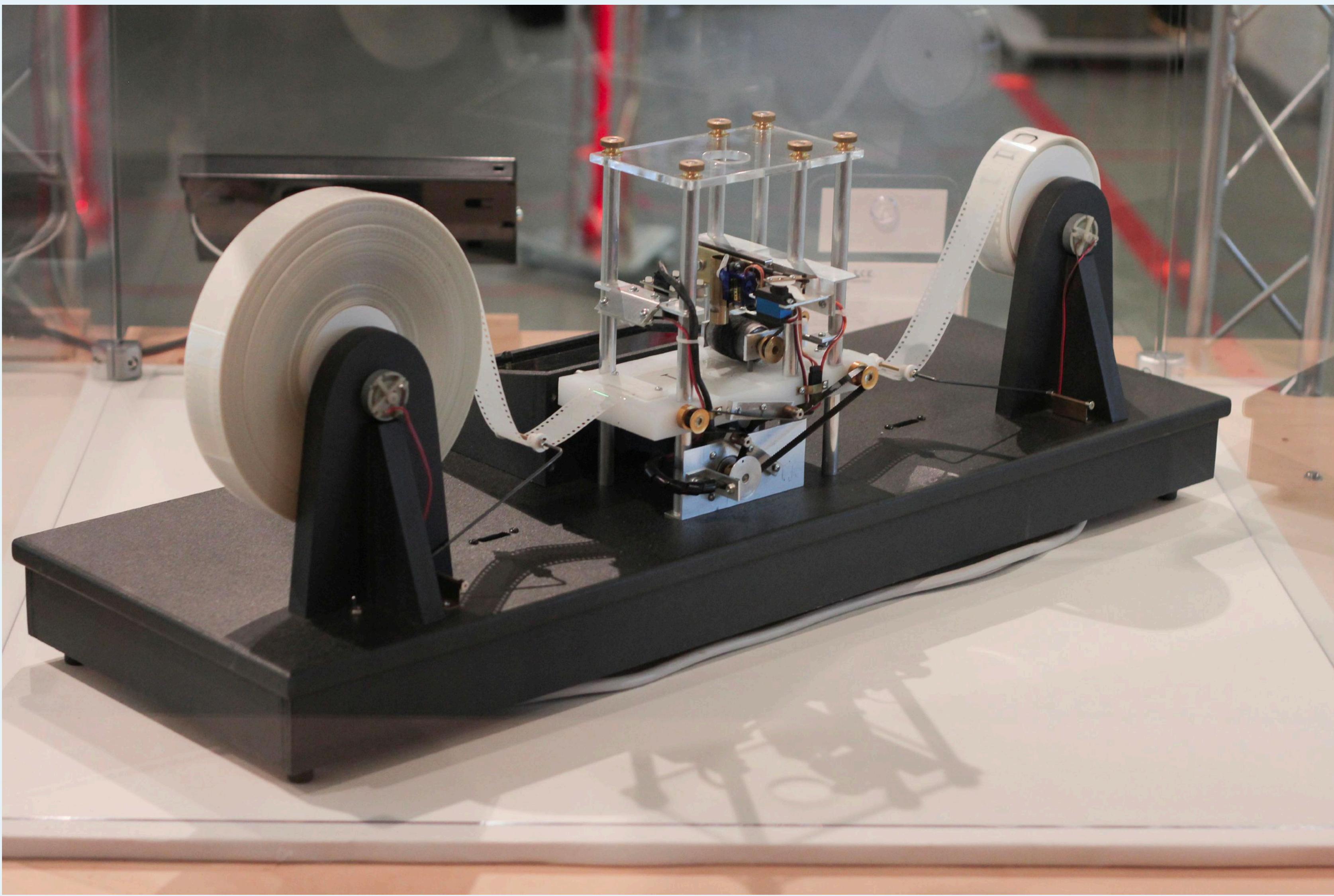
Appaloosa, Intel 4004 architecture, CC 表示-継承 3.0, [https://ja.wikipedia.org/wiki/Intel\\_4004#/media/](https://ja.wikipedia.org/wiki/Intel_4004#/media/)

%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB:4004\_arch.svg

## Intel 4004 Architecture



前回まで作ったのはこここの一部分

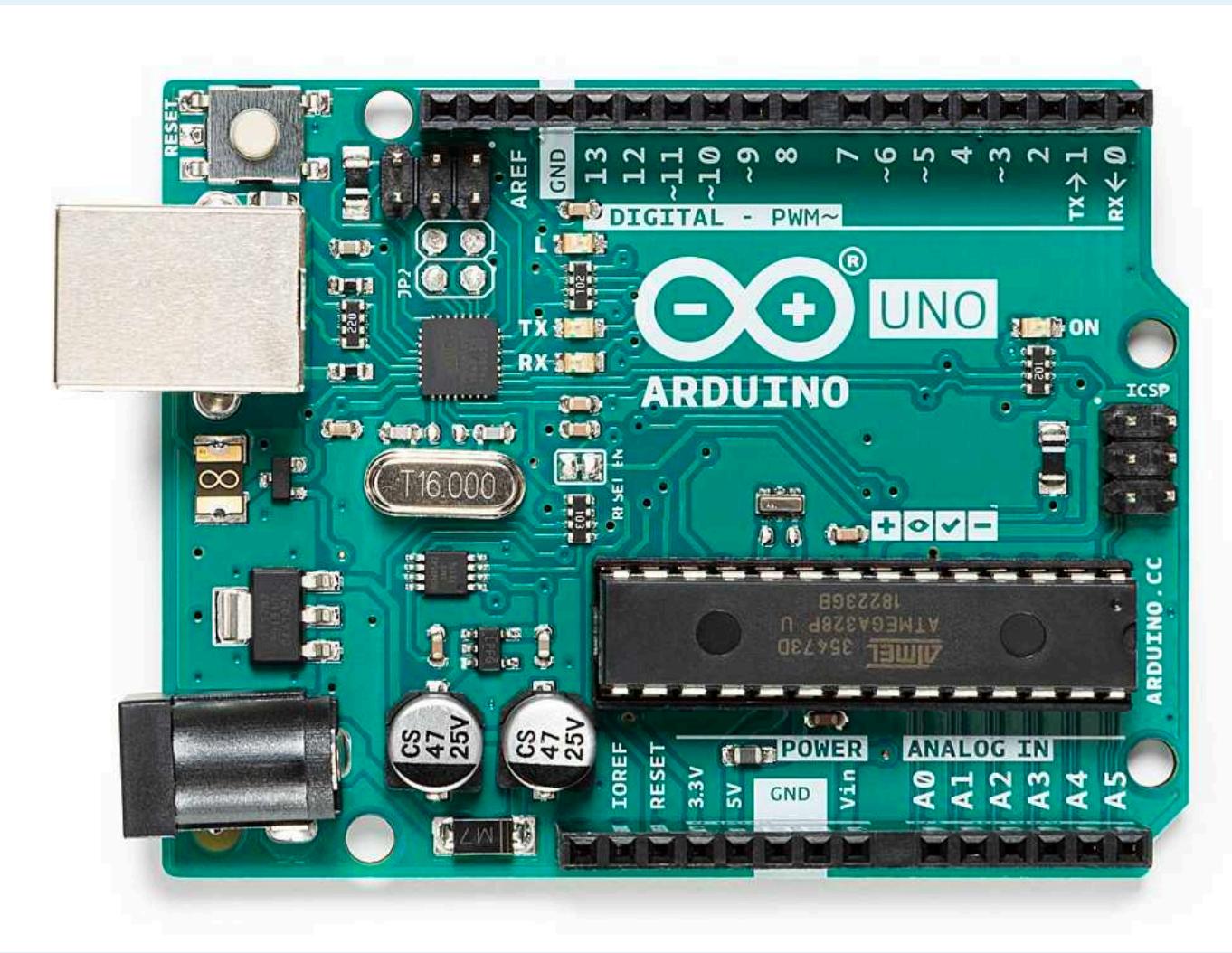


## チューリングマシンの物理モデル

Turing Machine, reconstructed by Mike Davey as seen at Go Ask ALICE at Harvard University, Rocky Acosta - Own work, CC-BY-3.0(2012),  
[https://en.wikipedia.org/wiki/Turing\\_machine#/media/File:Turing\\_Machine\\_Model\\_Davey\\_2012.jpg](https://en.wikipedia.org/wiki/Turing_machine#/media/File:Turing_Machine_Model_Davey_2012.jpg)

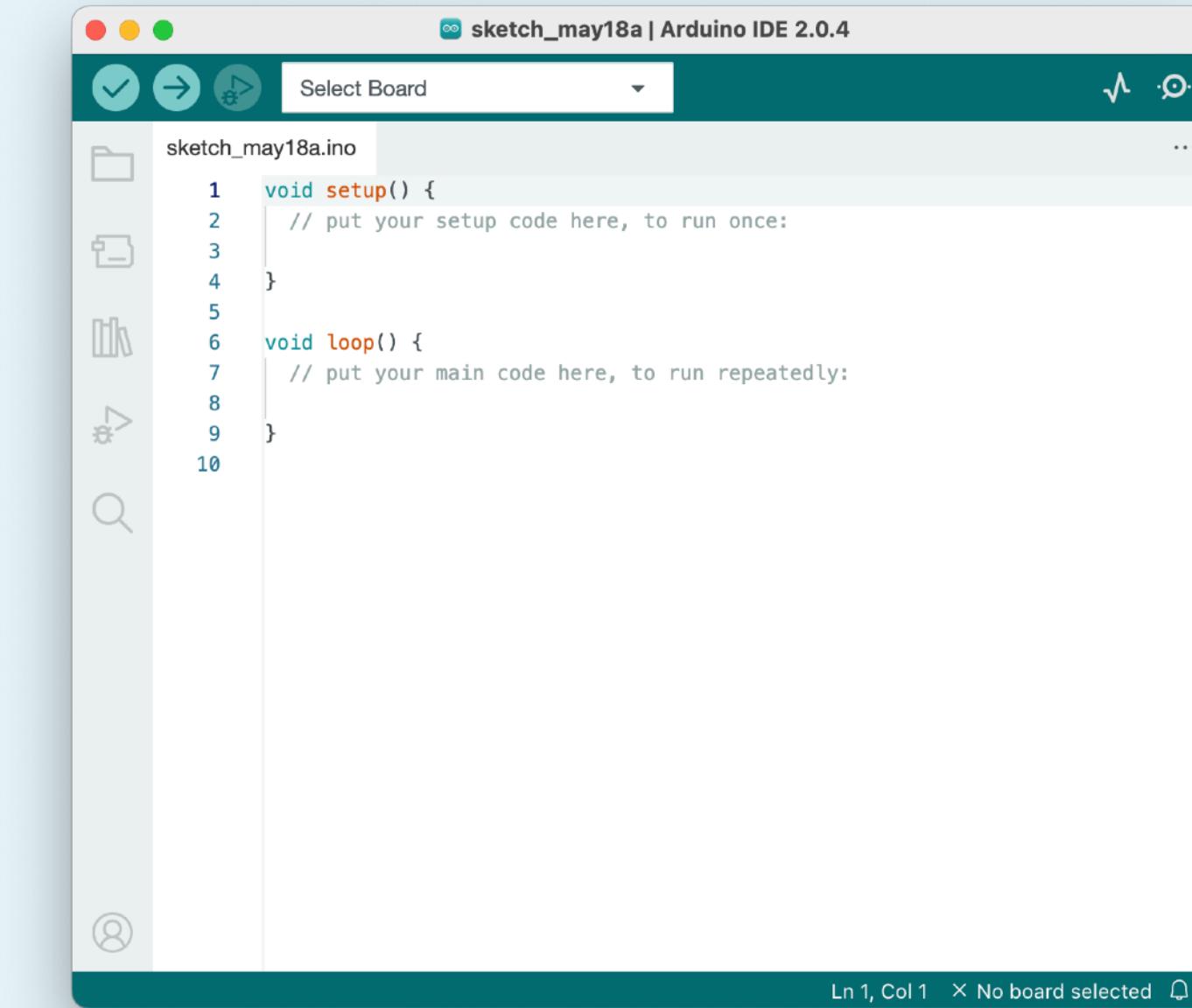
# Arduino入門

# Arduino



規格化された入出力のボード

+

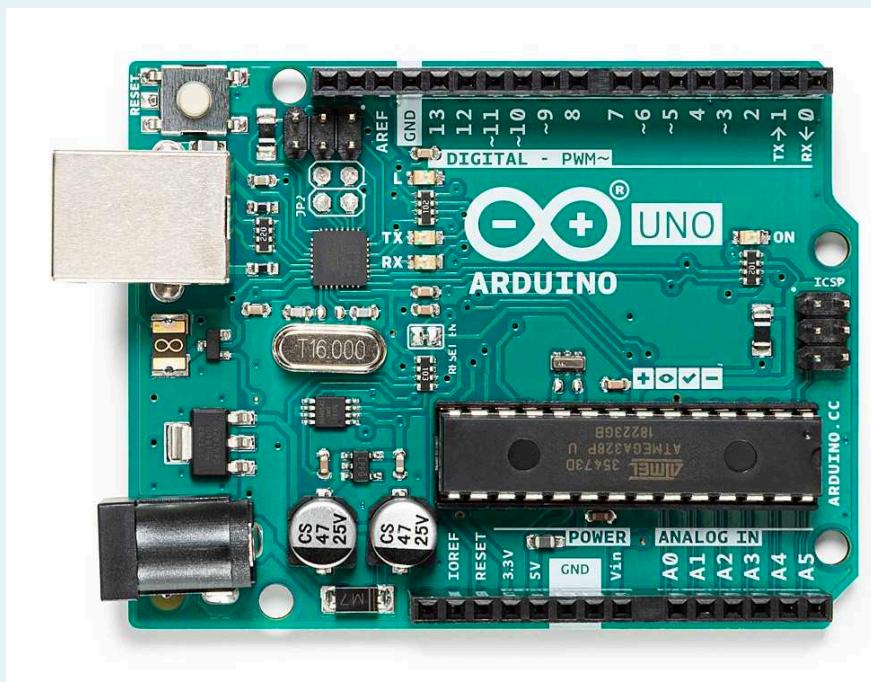


簡単 + 複数のボードに対応したプログラミング環境

C、C++を簡易化した言語を使用

拡張子は.ino

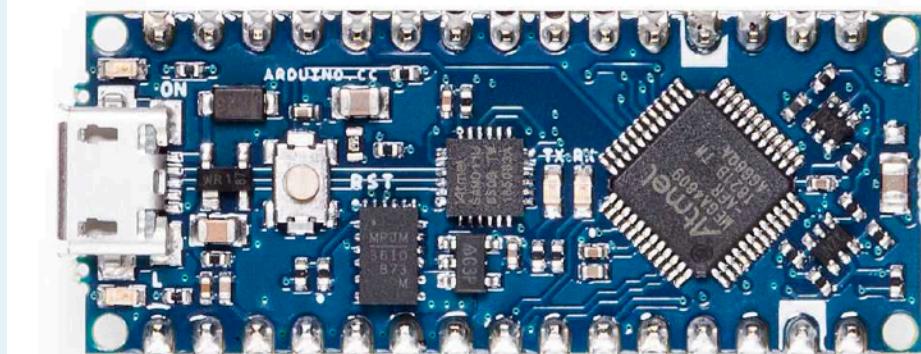
# Arduino公式シリーズ



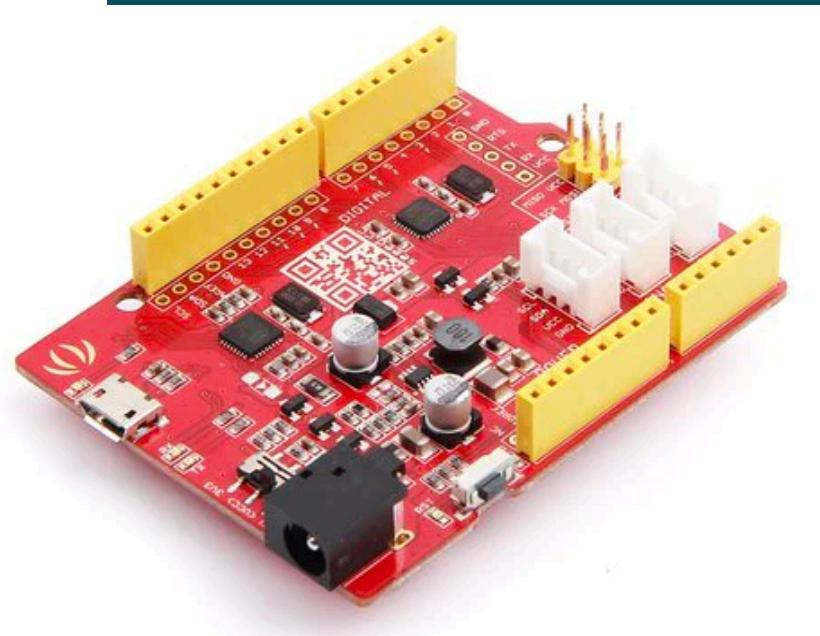
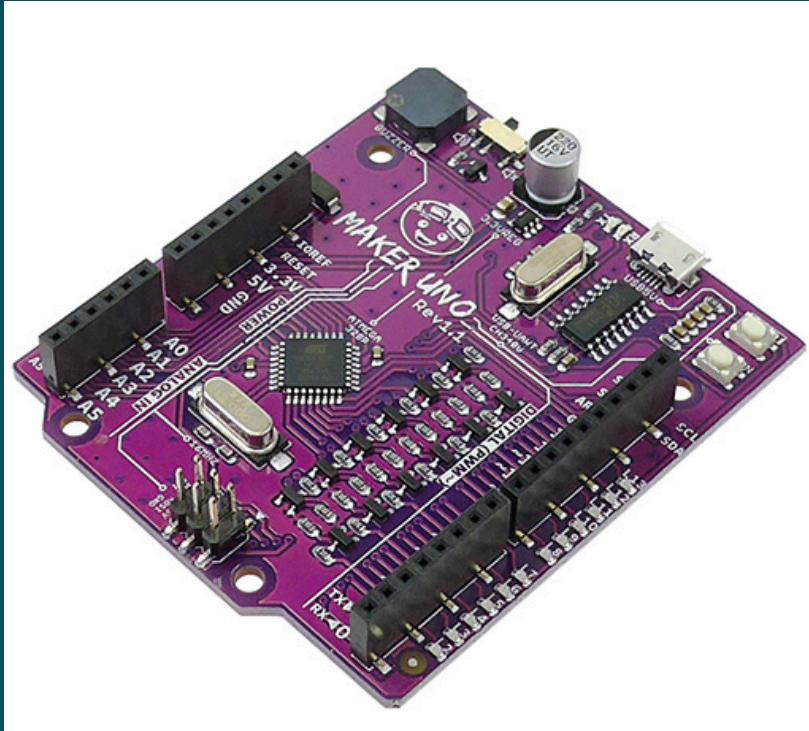
Arduino UNO



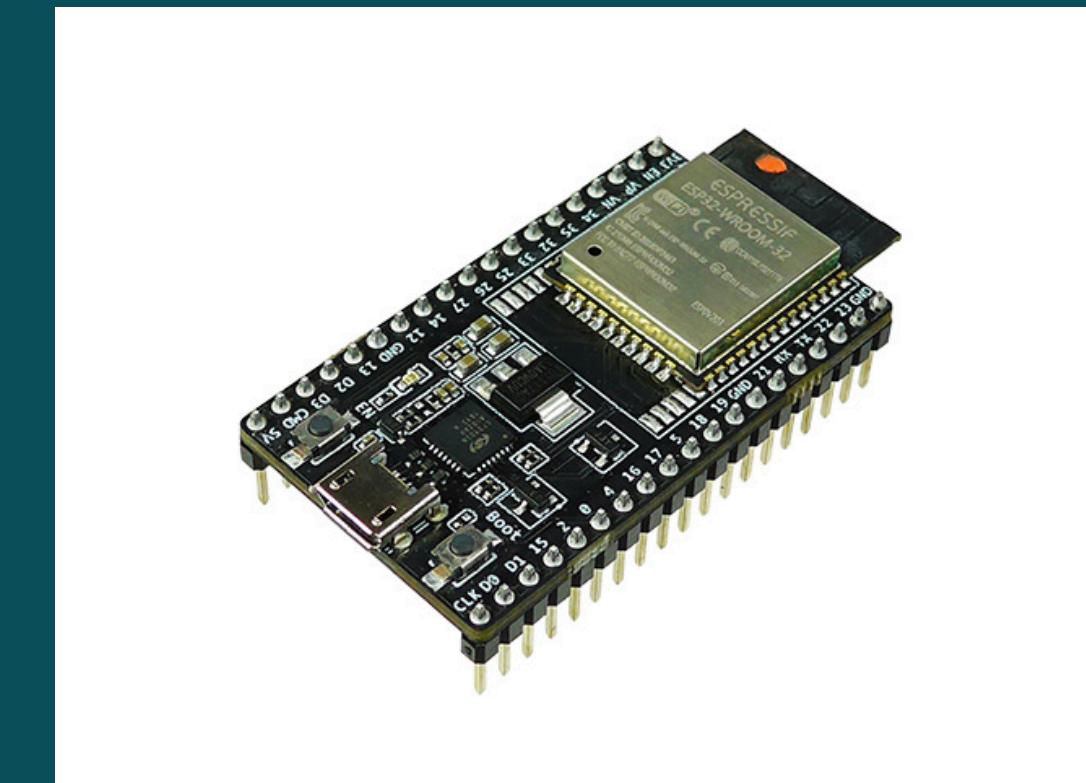
Arduino Mega



Arduino Nano Every



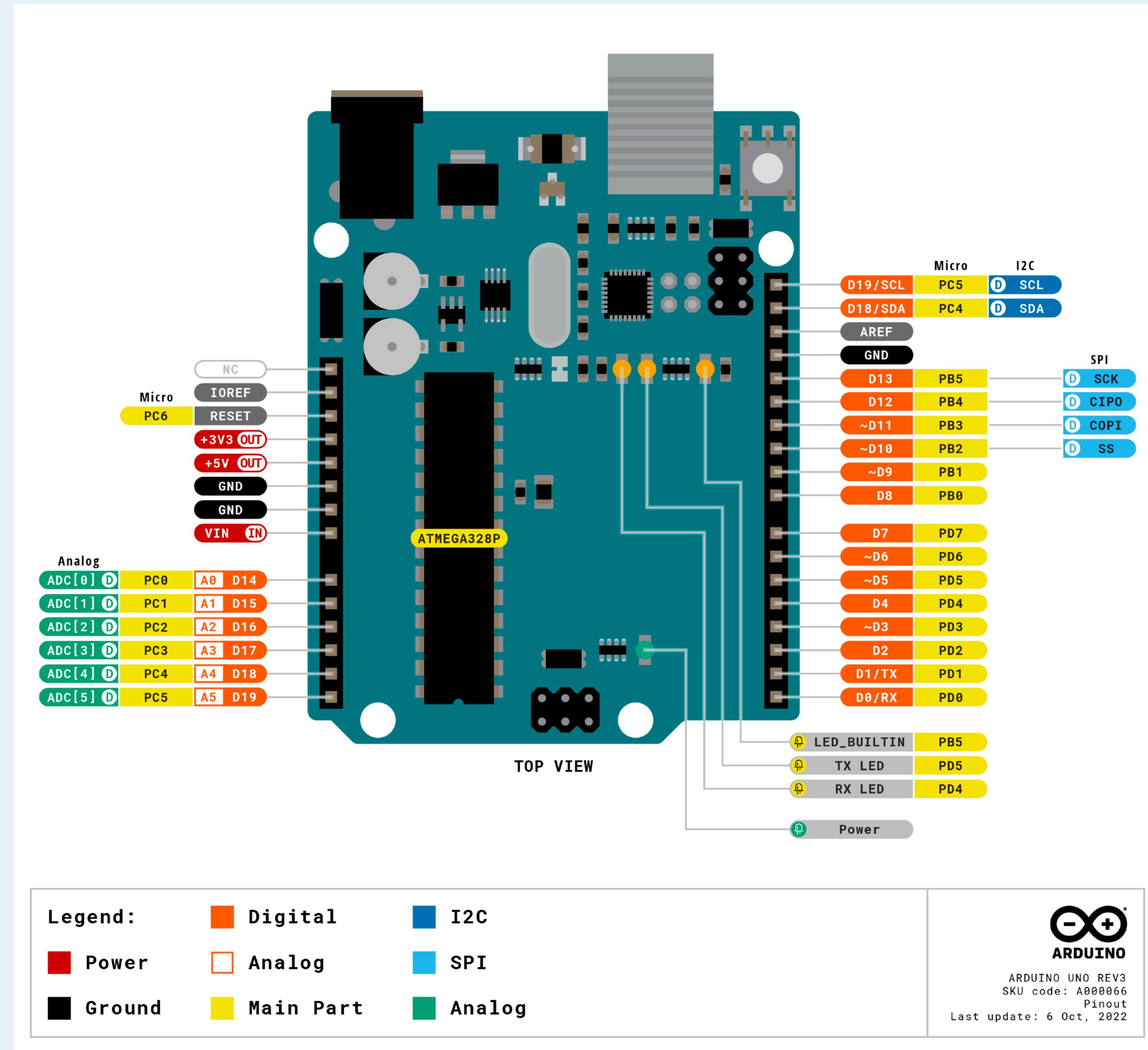
ピン配置互換性

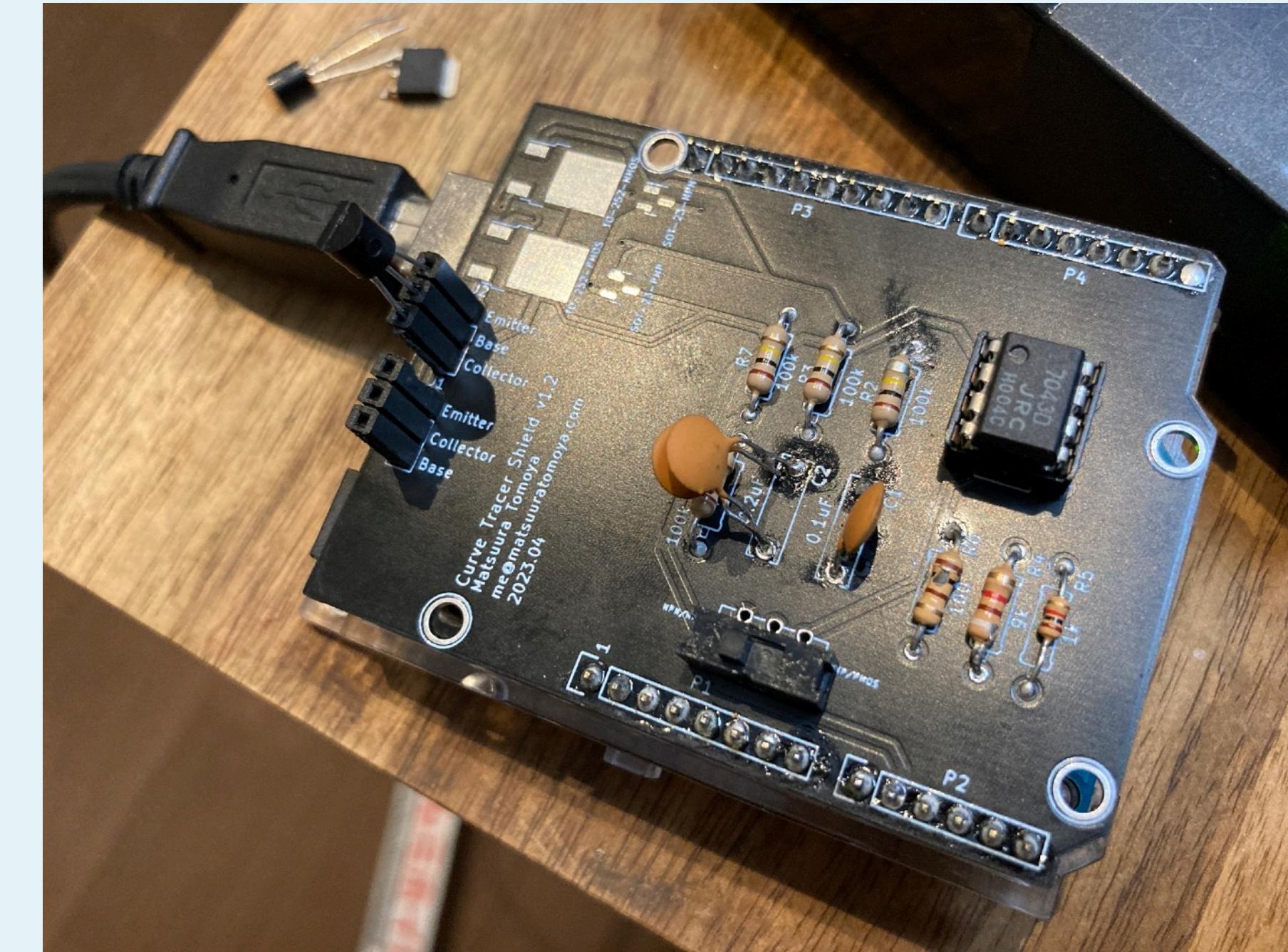
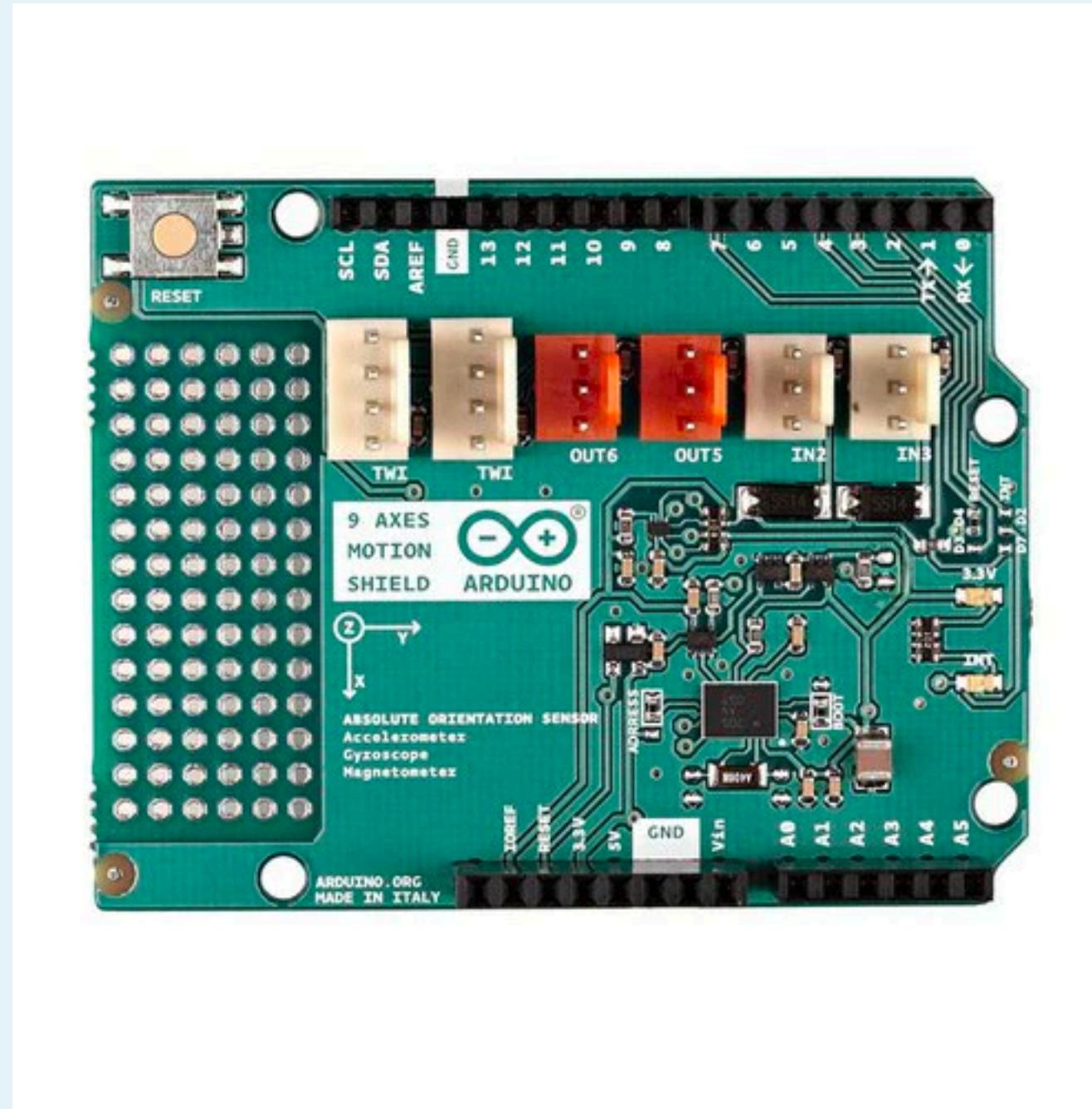


Arduino IDEでの開発対応

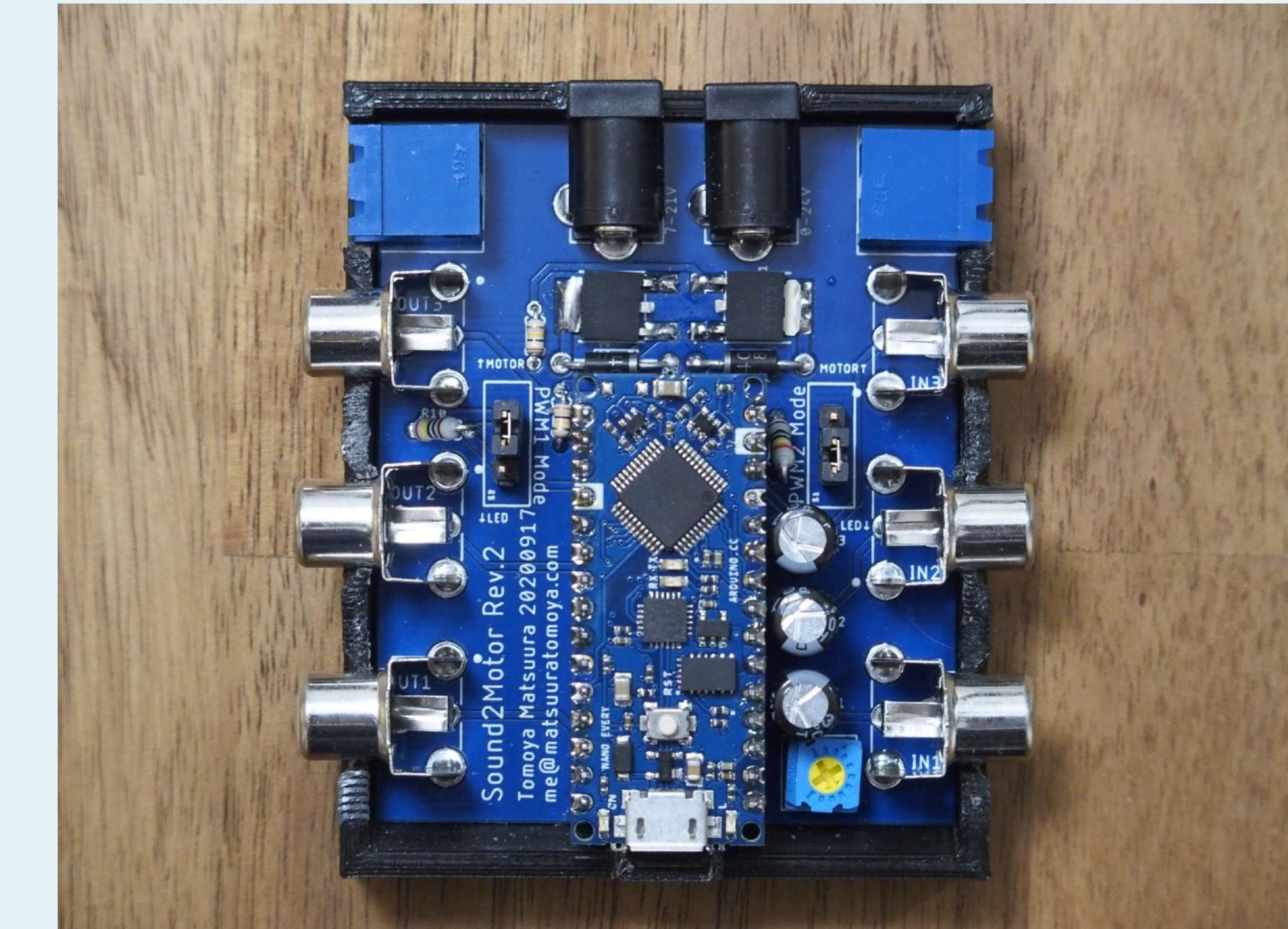


M5STACK





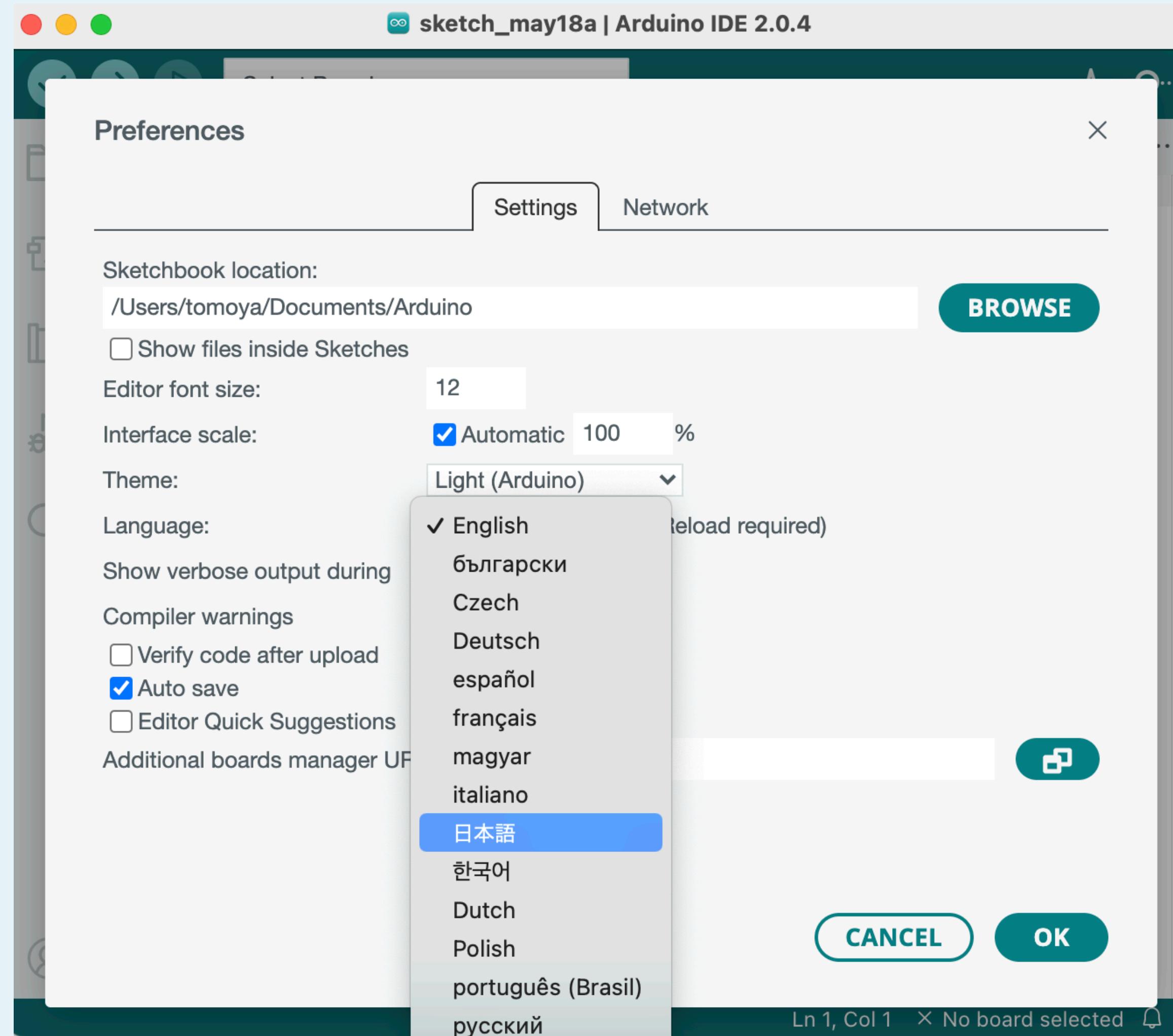
シールドと呼ばれる拡張基板を組み合わせることができる  
(自分で作れる)



音量に応じてLEDとモーターを起動するためのデバイス（2020年）

Arduino Nano Every用の拡張基板として実装

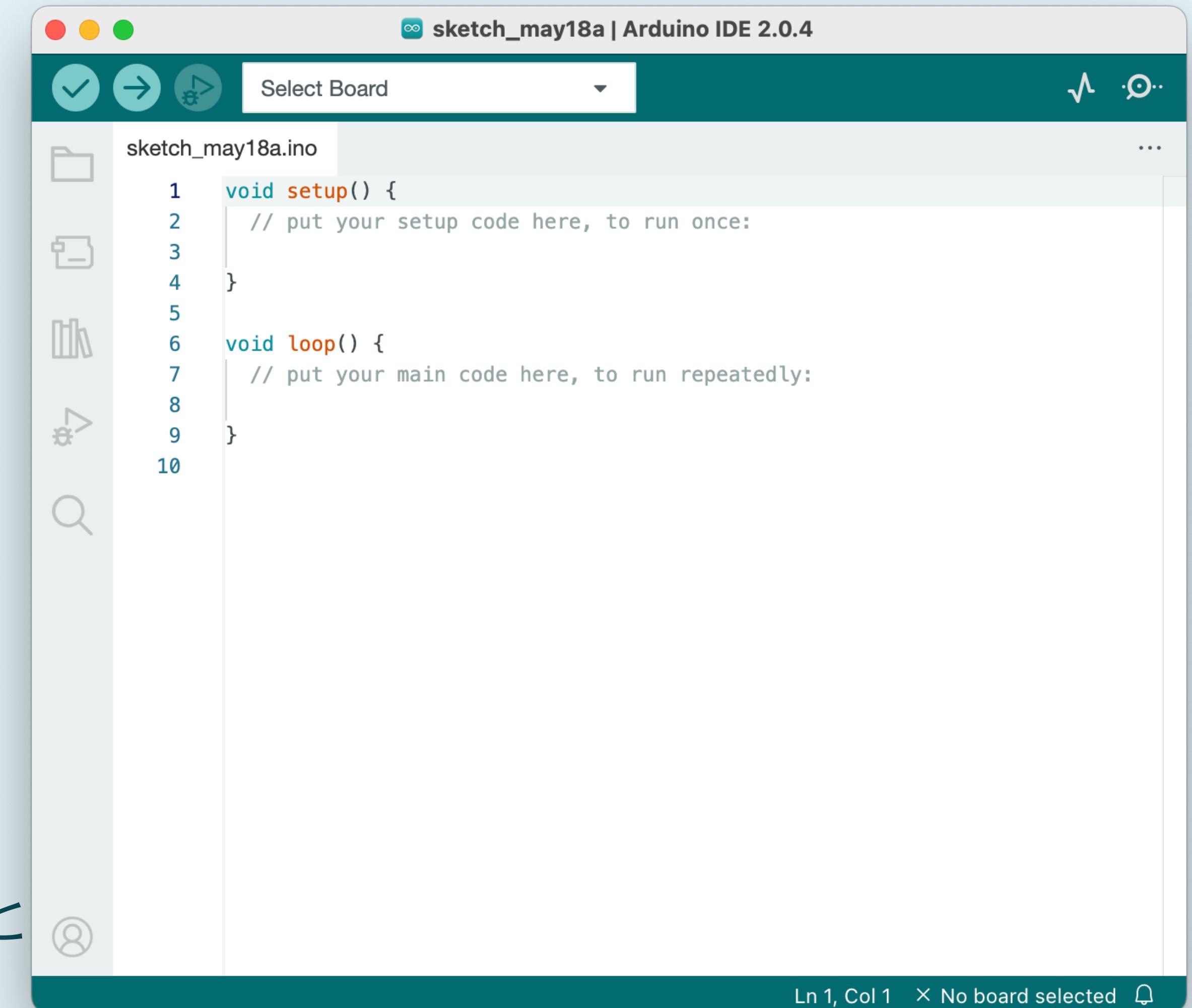
# Arduinoのプログラム入門



環境設定(Cmd+,)⇒Languageで日本語化できます

# スケッチ

- Arduinoのソースコードのファイルは「スケッチ」と呼ばれる
- Arduino IDE.appを立ち上げると自動で日付が入ったスケッチが開かれる
- ドキュメント/Arduino 以下に保存されるので、AMCの端末を使用する方は適宜USBメモリなどに保存すること



The screenshot shows the Arduino IDE 2.0.4 interface. The title bar reads "sketch\_may18a | Arduino IDE 2.0.4". The main window displays a new sketch template named "sketch\_may18a.ino". The code editor contains the following C-like pseudocode:

```
1 void setup() {
2     // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7     // put your main code here, to run repeatedly:
8
9 }
10
```

The IDE includes standard toolbars and a sidebar with icons for file operations like Open, Save, and Find.

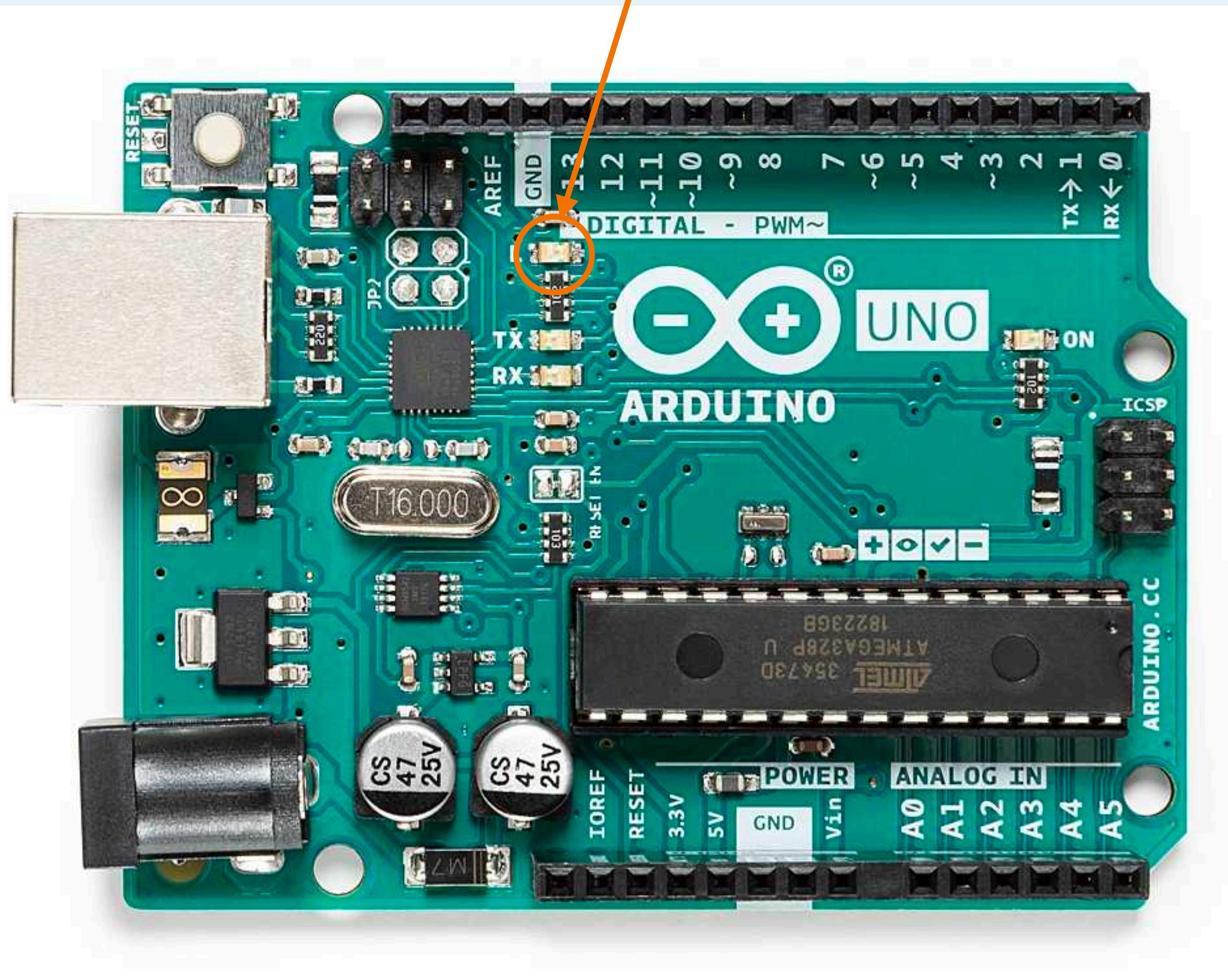
# スケッチ

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

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

とりあえず写経してみましょう

ここが1秒おきに点滅する



# C言語のポイント

- 全角でタイプしない
- 大文字小文字を間違えない
- 行末のセミコロンを忘れない
- 2つ以上のスペースは全て1つのスペースと同じ  
(これも見やすさ重視)
- 改行はあってもなくても処理には関係ない（見  
やすさ重視でお好きに）
- //以降はコメントアウト

The screenshot shows the Arduino IDE interface with the title bar "sketch\_may18a | Arduino IDE 2.0.4". The sketch file "sketch\_may18a.ino" contains the following code:

```
1 void setup() {
2   pinMode(LED_BUILTIN, OUTPUT);
3 }
4
5 void loop() {
6   digitalWrite(LED_BUILTIN, HIGH);
7   Delay(1000);
8   digitalWrite(LED_BUILTIN, LOW);
9   delay(1000)
10}
11
12
```

The "Output" window below shows the command-line interface with the following text:

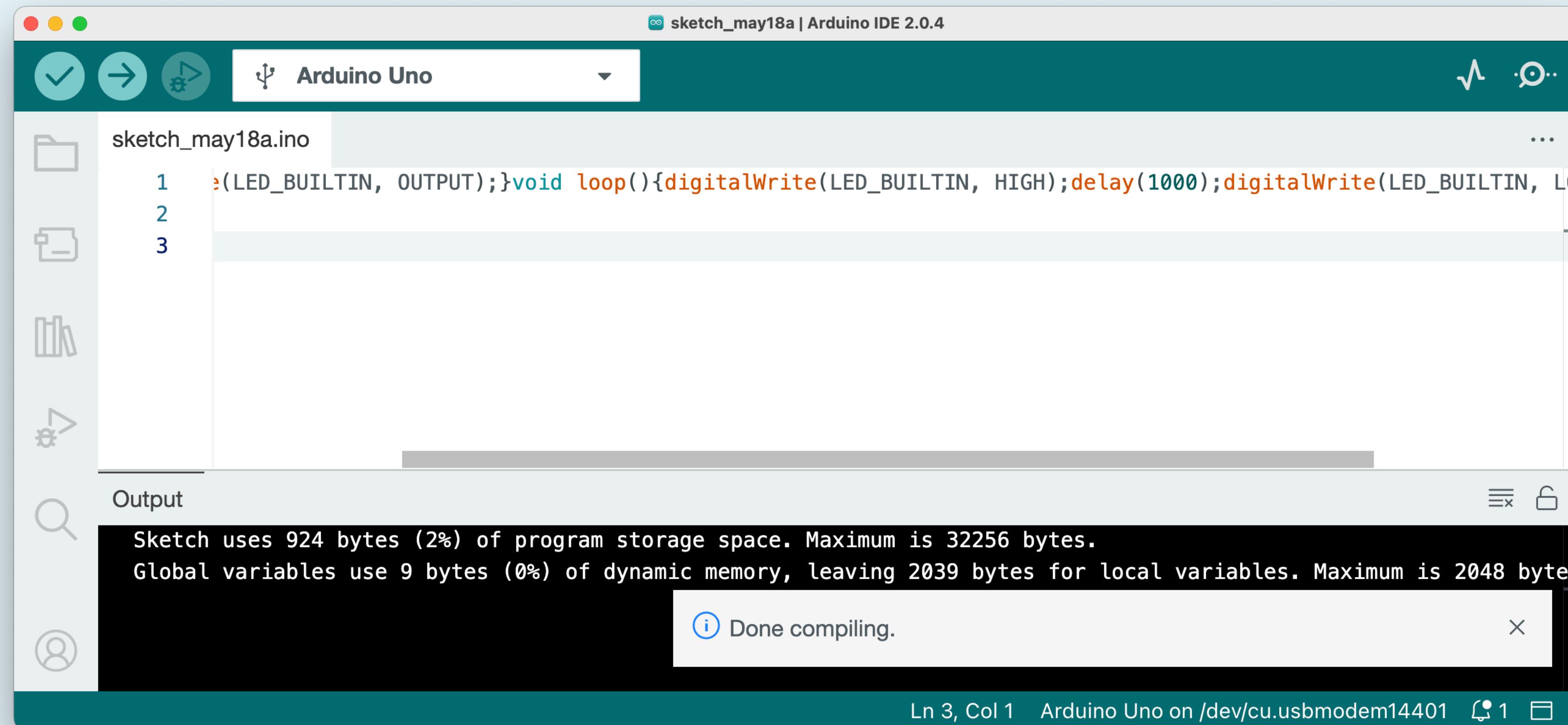
```
| | ~~~~  
| /private/var/folders/b3/gftgtjkd1cnczy04xj60w76r000gn/T/.arduinoIDE  
| | Delay(1000);  
| | ~~~~  
| | delay
```

At the bottom of the IDE, the status bar displays "Ln 2, Col 3 Arduino Uno on /dev/cu.usbmodem14401".

# 覚えておくと便利なショートカット

- Cmd+←→：行頭行末へカーソルをジャンプ
- Option+←→：単語単位でカーソルをジャンプ
  - +Shiftと組み合わせて選択範囲を変える
- Option+↑↓：カーソルのある行/選択範囲を上下に入れ替える
- Cmd+ /：カーソルのある行or選択範囲をコメントアウト、コメント解除

Visual Studio Codeなど、モダンなエディタでは大体共通して使える機能



仕様を理解してれば、一見めちゃくちゃ読みづらくても動く

# 数値をいじってみよう

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

void loop() {
    digitalWrite(LED_BUILTIN, HIGH);
    delay(500);
    digitalWrite(LED_BUILTIN, LOW);
    delay(100);
}
```

Delayの数値はミリ秒単位！

入力：センサーを使ってみよう

# シャープ 測距モジュール

## GP2Y0A02YK



赤外線LED

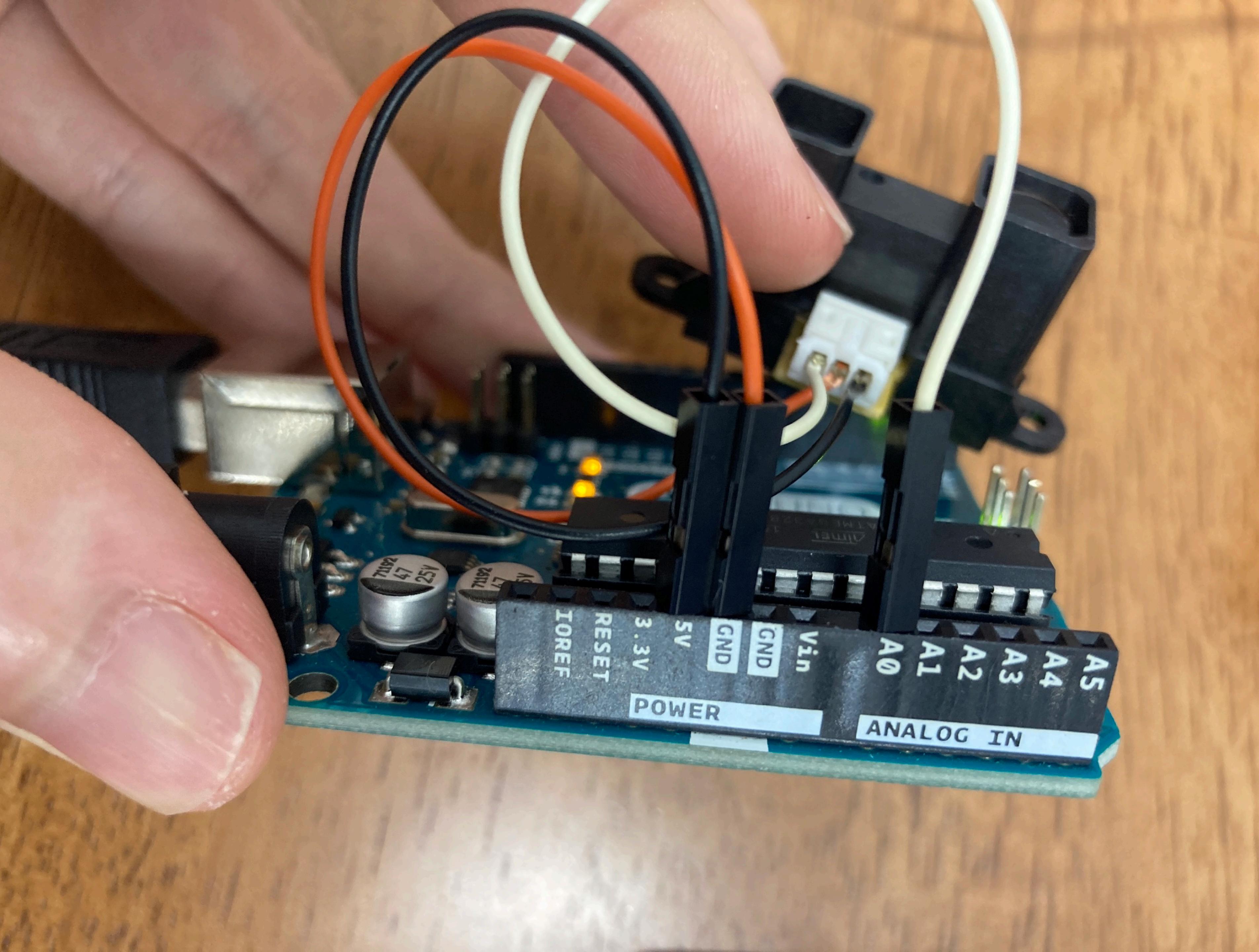
フォトトランジスタ

20cm~150cm測れる

白：距離に応じた電圧出力

オレンジ：GND

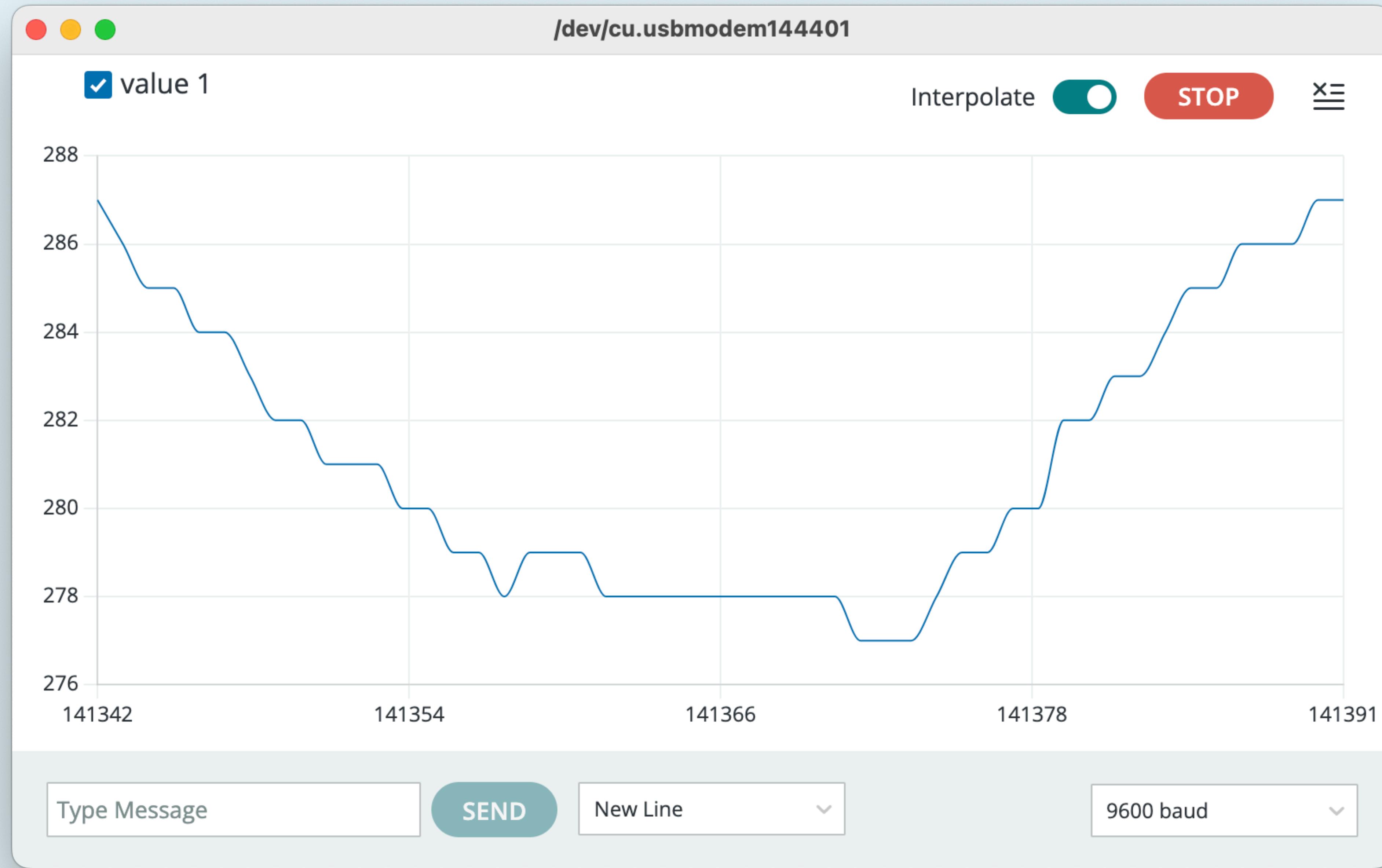
黒：電源(+5V)



```
void setup() {  
    pinMode(A0, INPUT);  
    Serial.begin(9600);  
}  
  
void loop() {  
    int distance = analogRead(A0);  
    Serial.println(distance);  
    delay(20);  
}
```

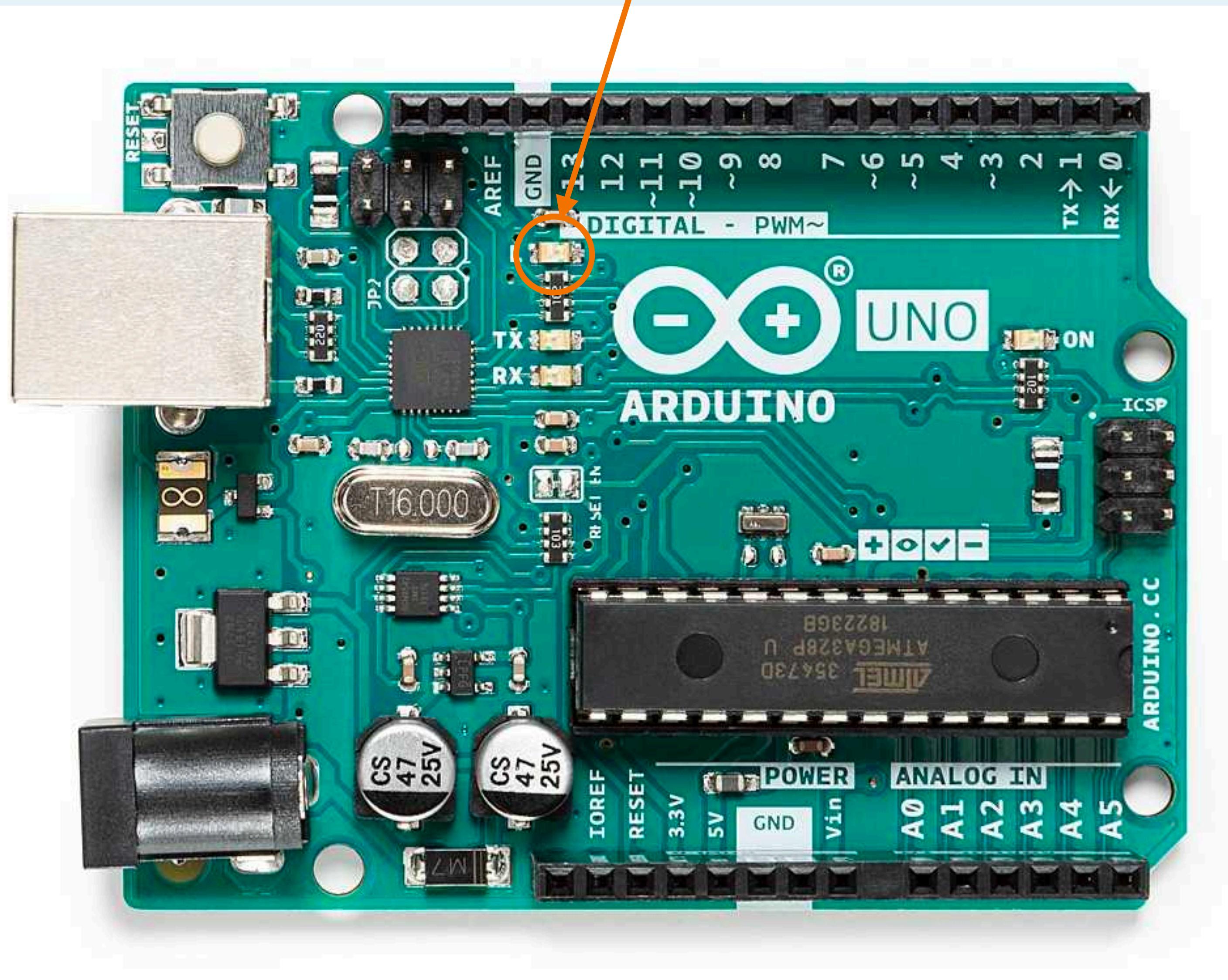
書き込んだらここをクリック



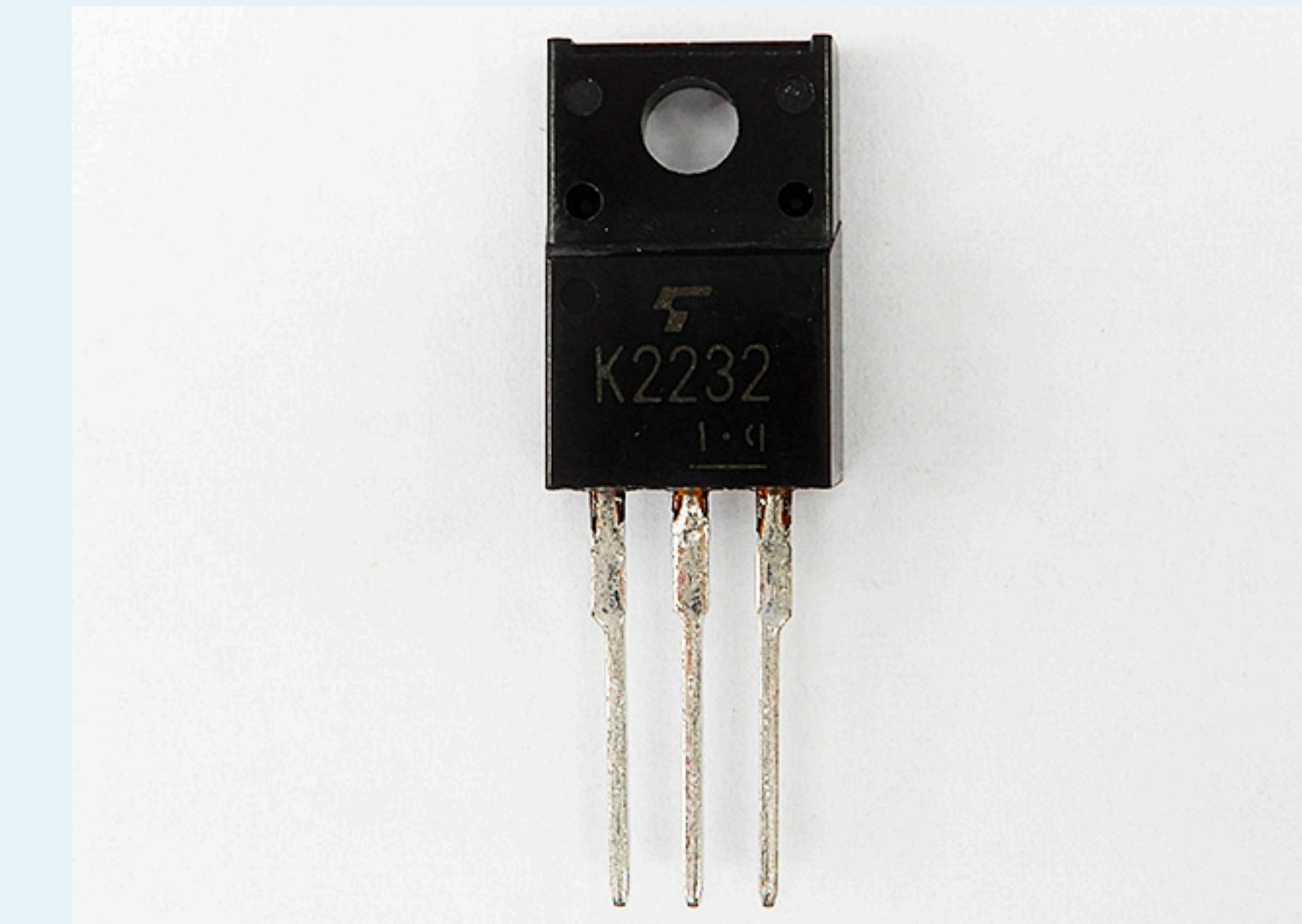
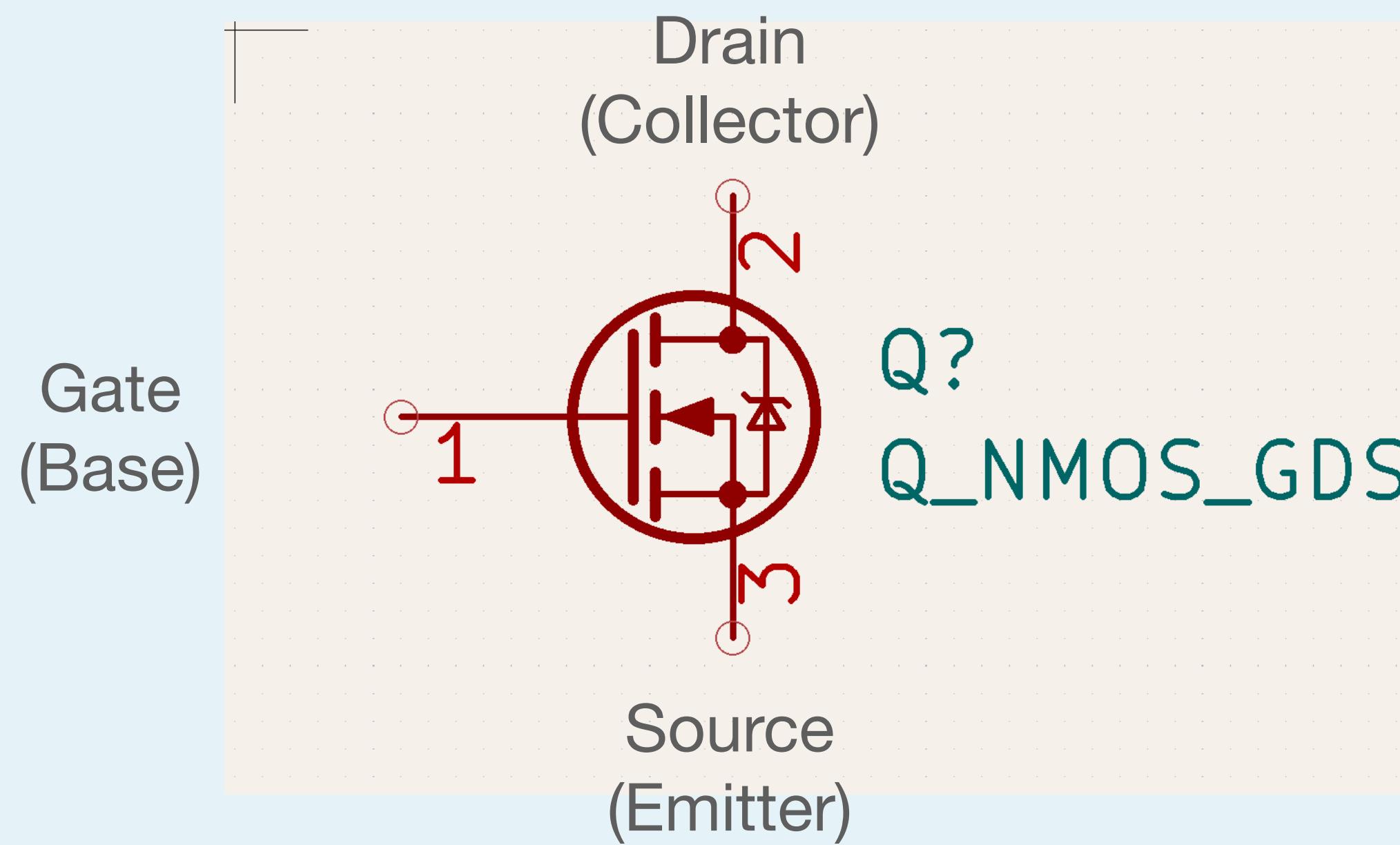


出力:電流消費の激しい物を扱う

ここが1秒おきに点滅する

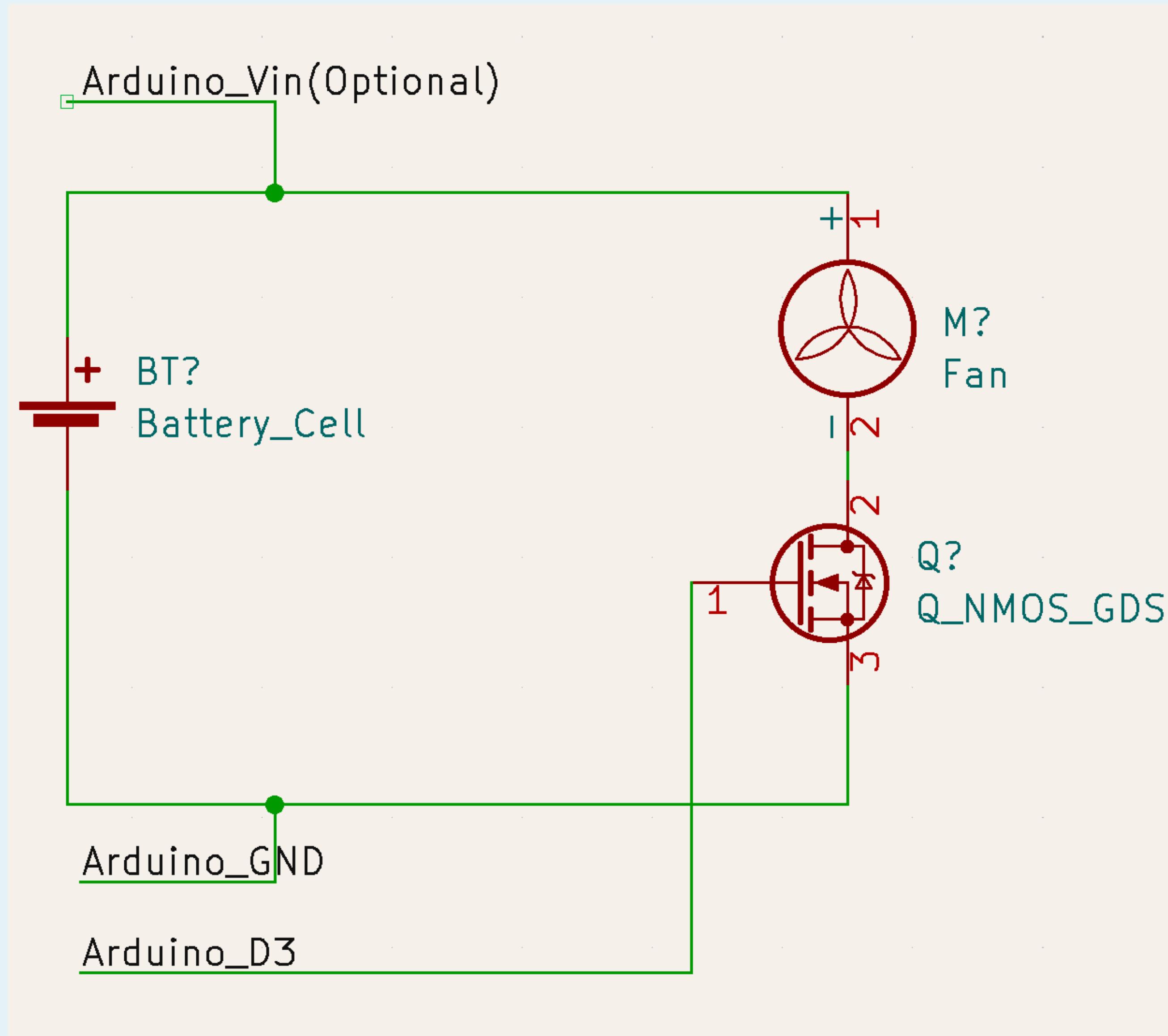


# K2232 MOSFET



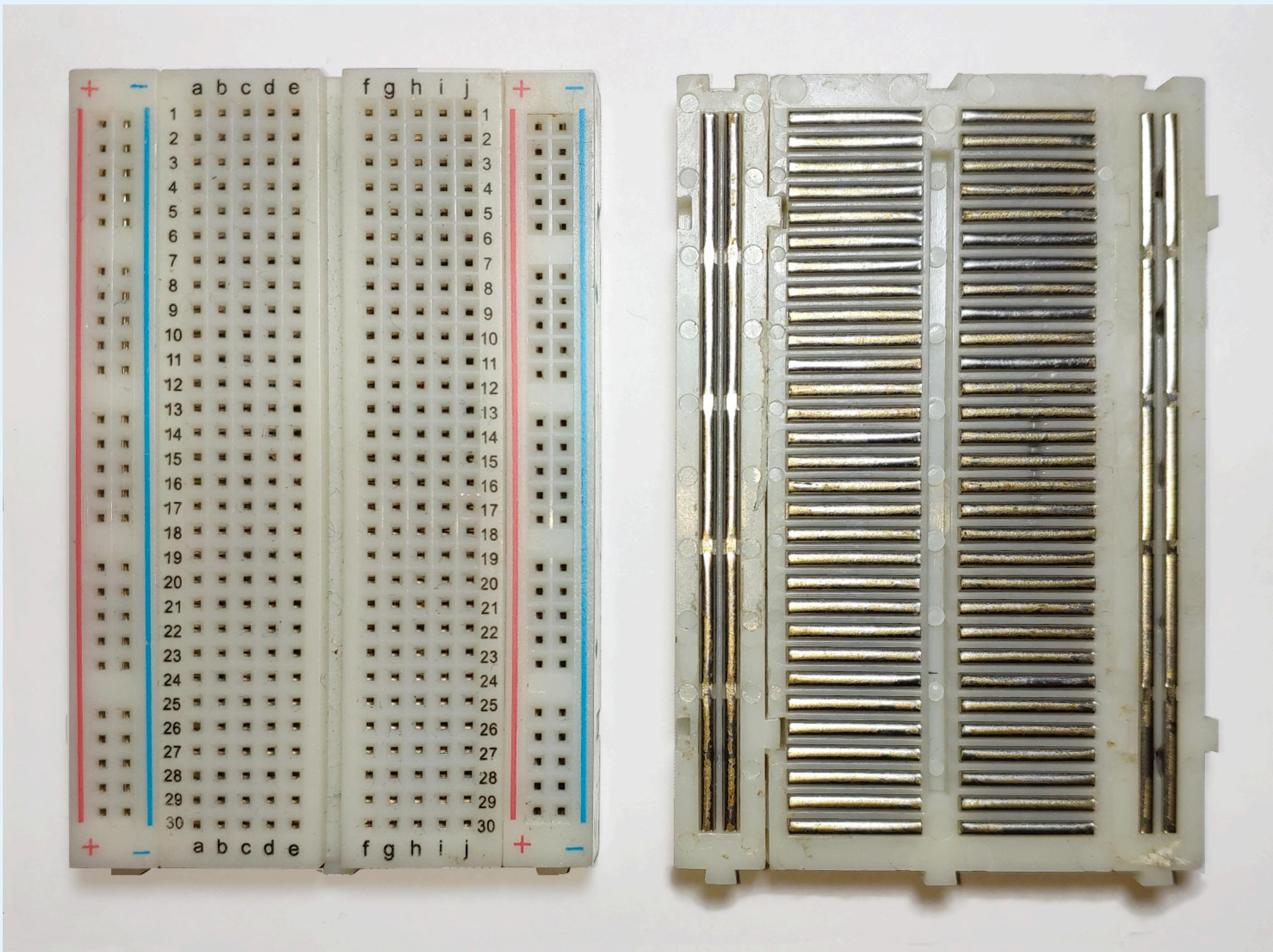
<https://akizukidensi.com/catalog/g/gl-02414/>

トランジスタとほぼ同じだが、ゲートからソースには電流が流れない

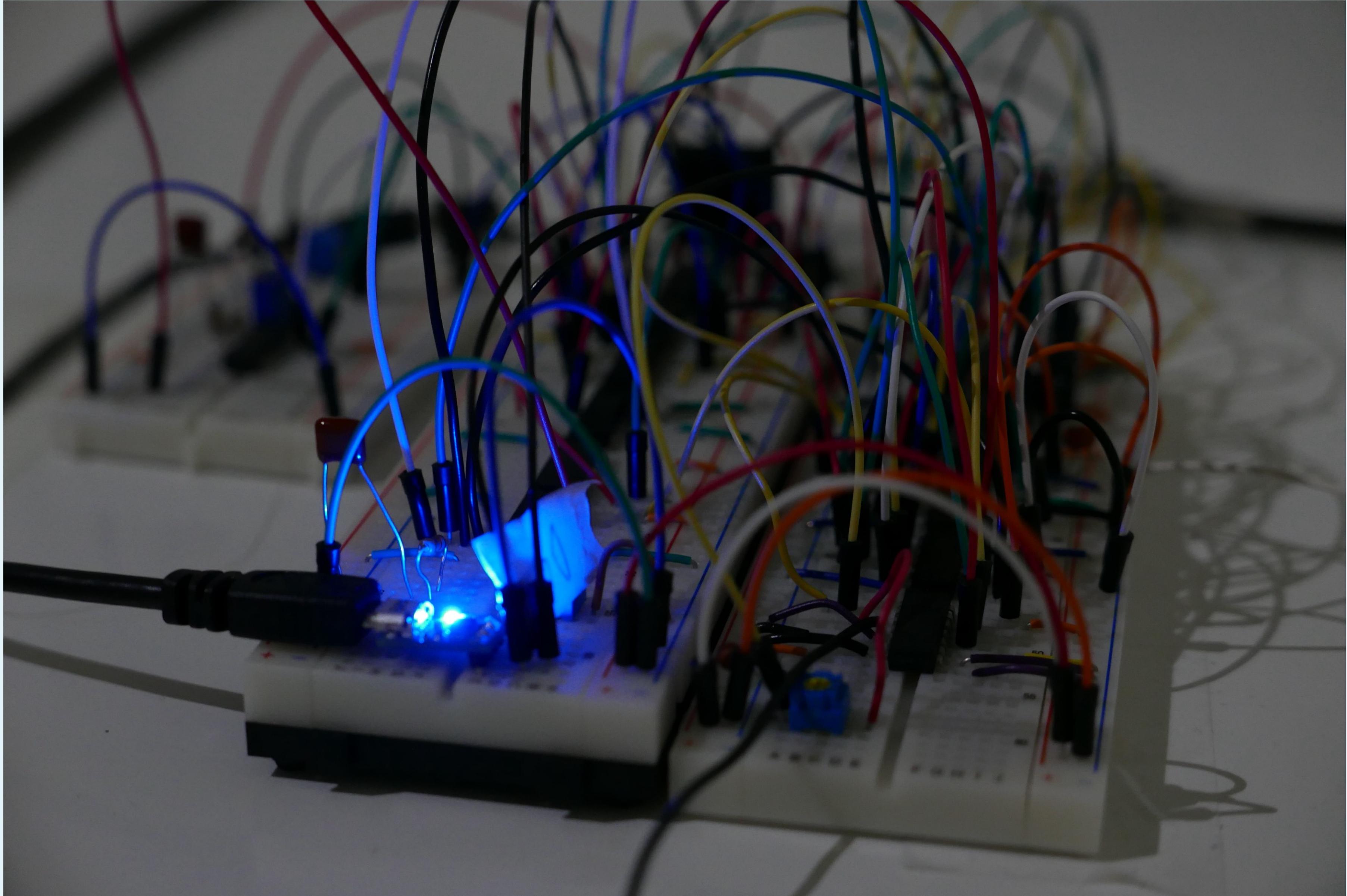


\*ファンから逆流する電流について考慮していないので不適切です、今度補足

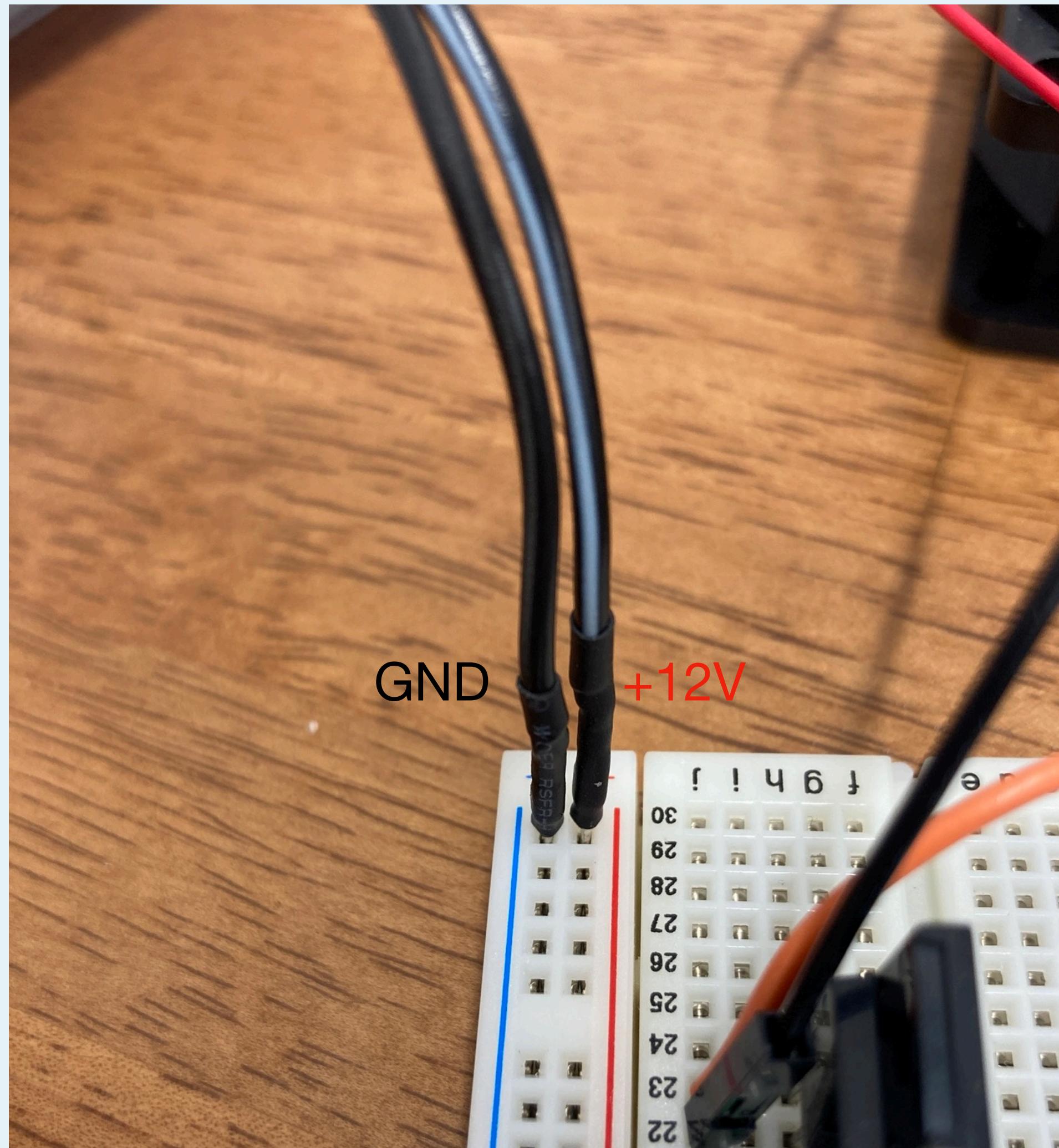
# ブレッドボード



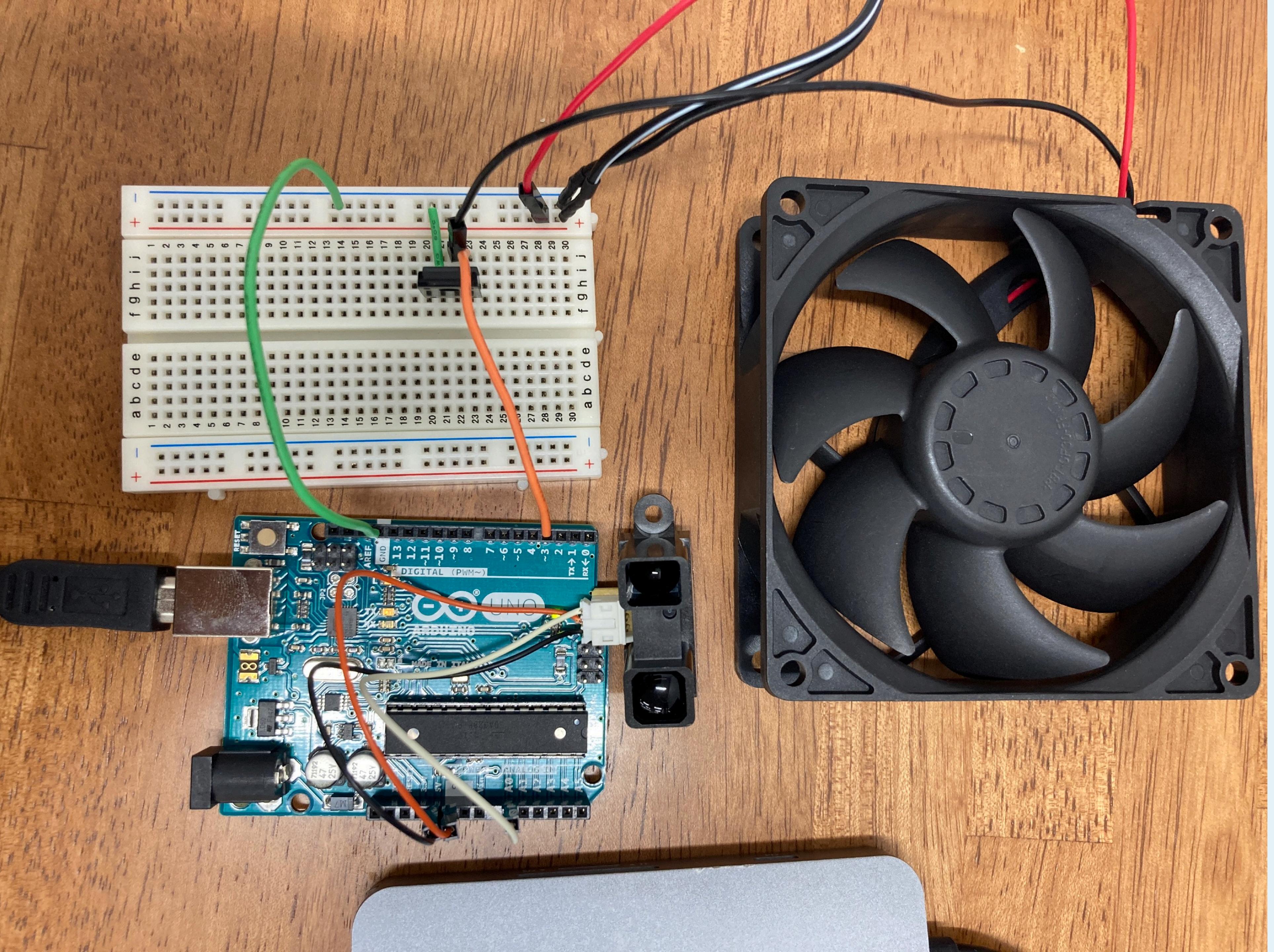
- はんだ付けなしで簡単に回路を組みたい人用

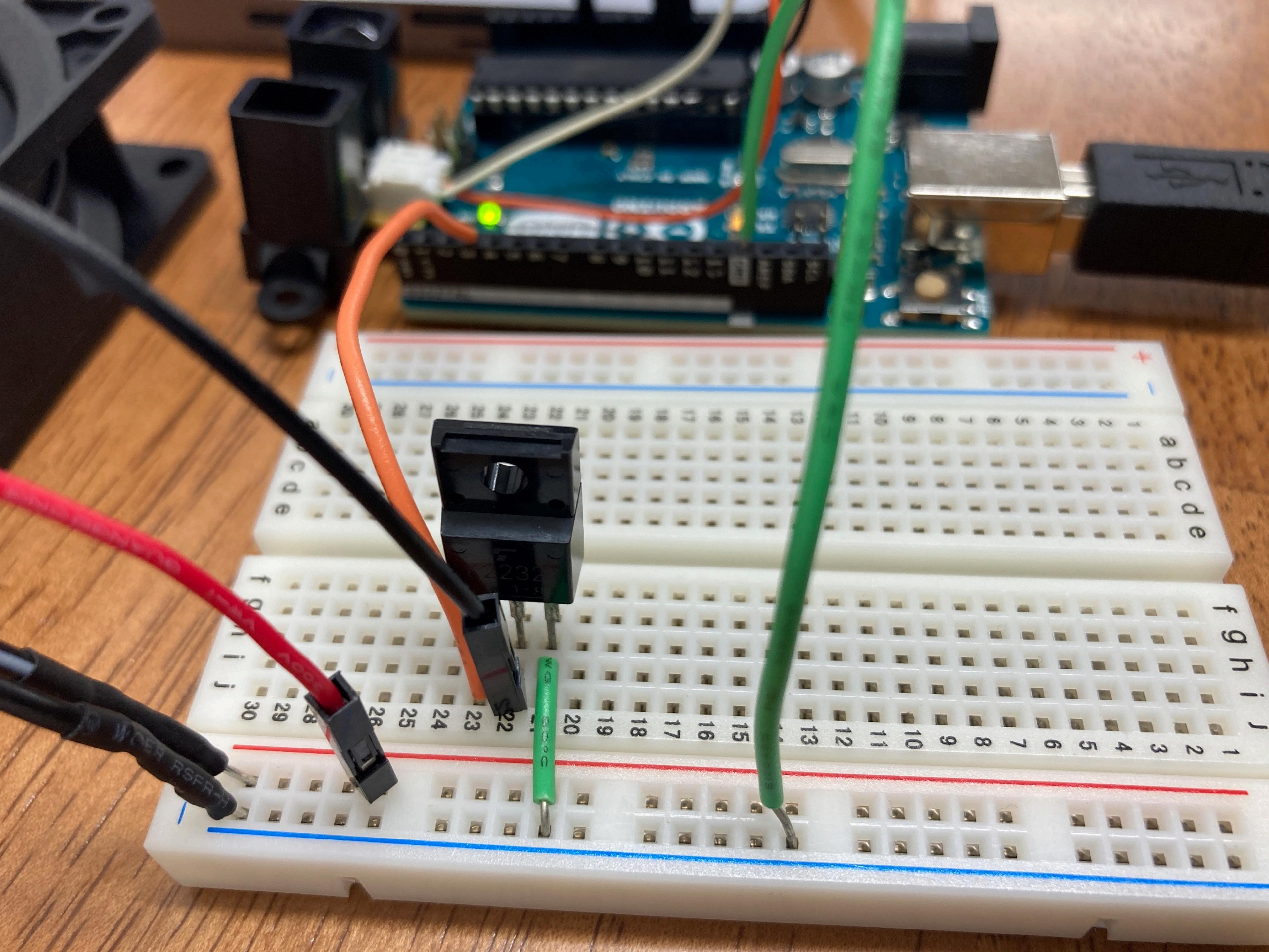


<https://matsuuratomo.ya.com/works/adm/>



今回は、白い線が入ってる方が+  
(場合によりけりなので、きちんとテスターで確かめること)



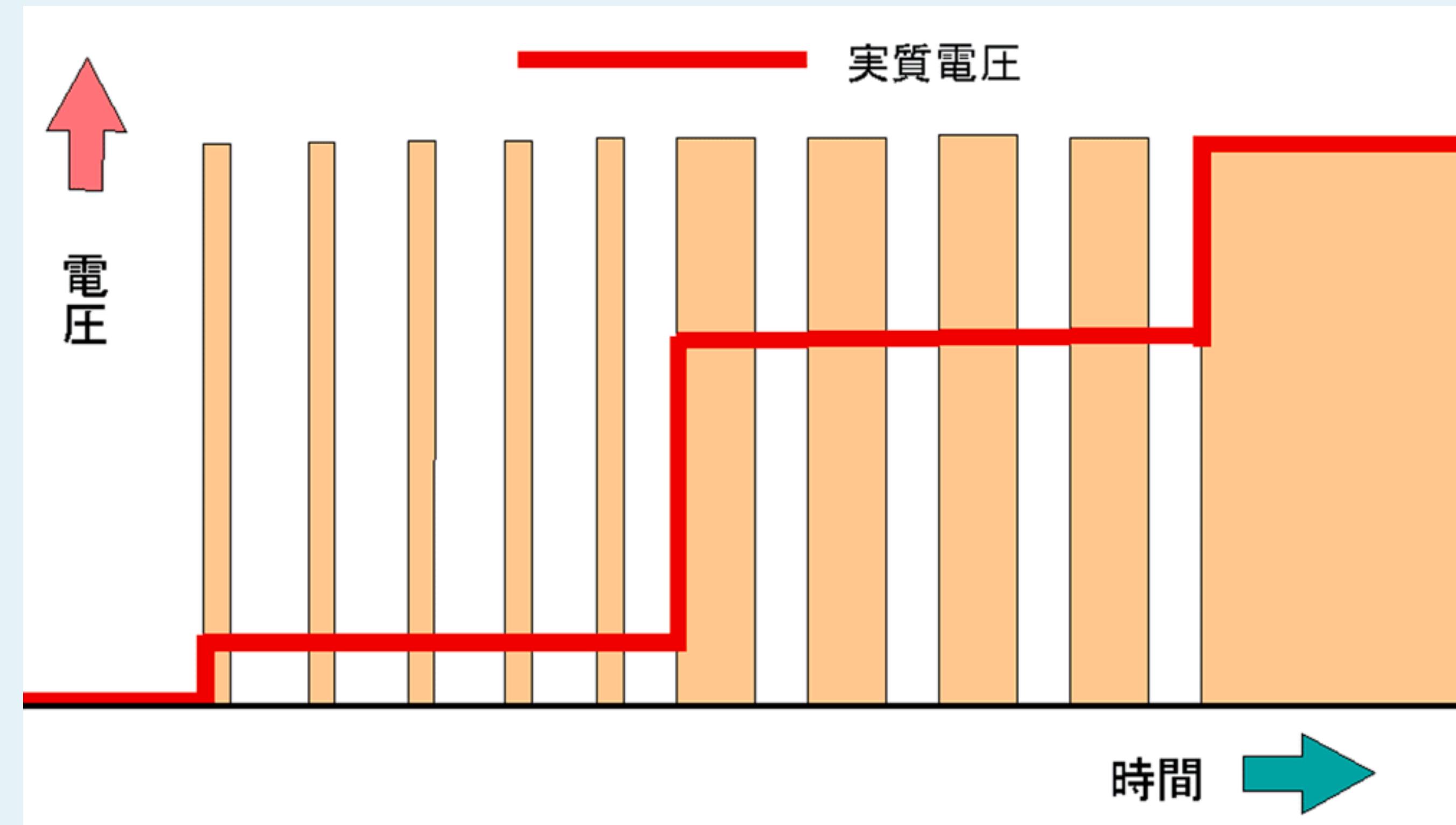


```
void setup() {
    pinMode(A0, INPUT);
    pinMode(3, OUTPUT);
    Serial.begin(9600);
}

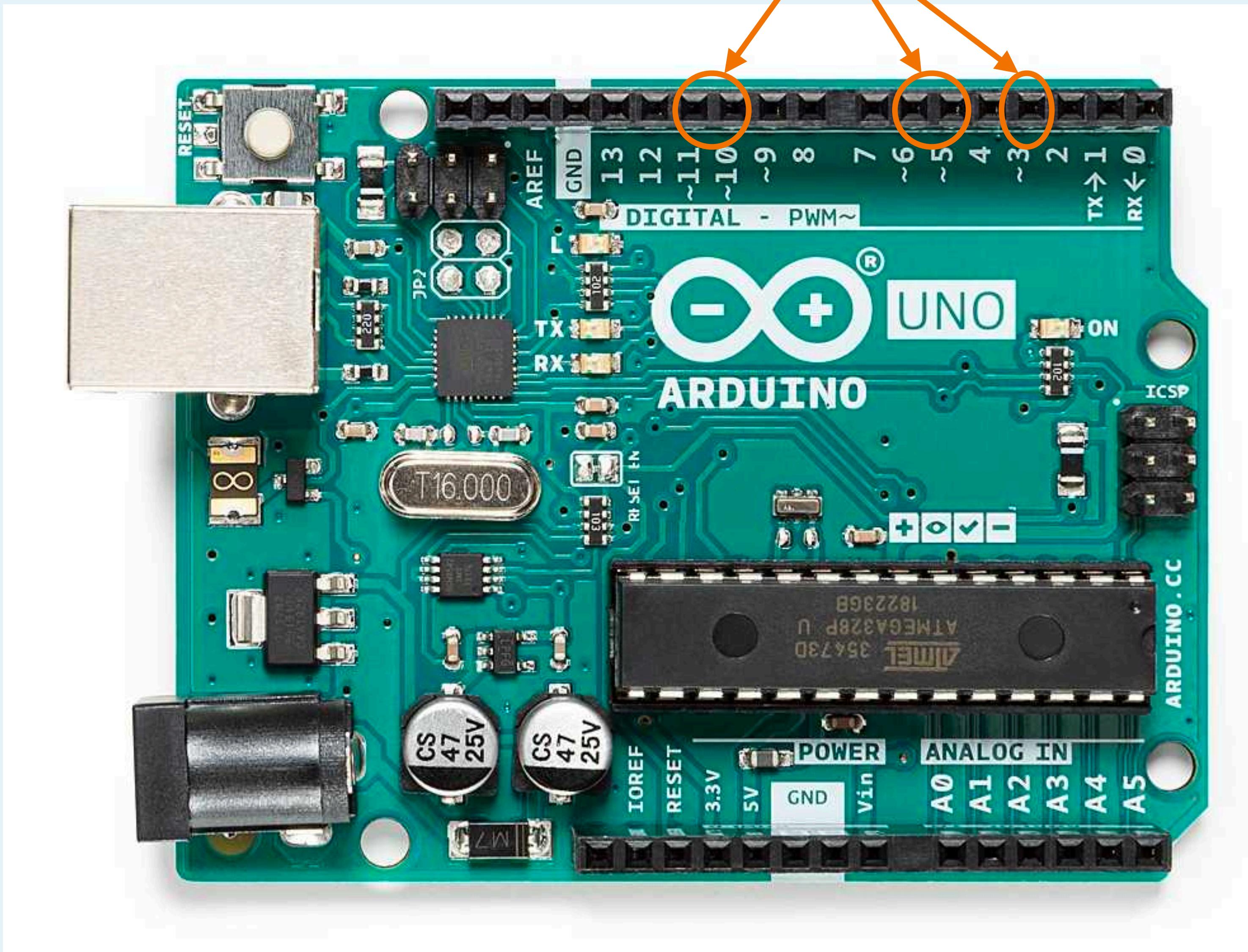
void loop() {
    int distance = analogRead(A0);
    Serial.println(distance);
    if(distance>400){
        digitalWrite(3,HIGH);
    }else{
        digitalWrite(3,LOW);
    }
    delay(20);
}
```

If文でオンオフ切り替え

# PWM制御



analogWrite()でPWM出力が使用できるピン (~がついたもの)

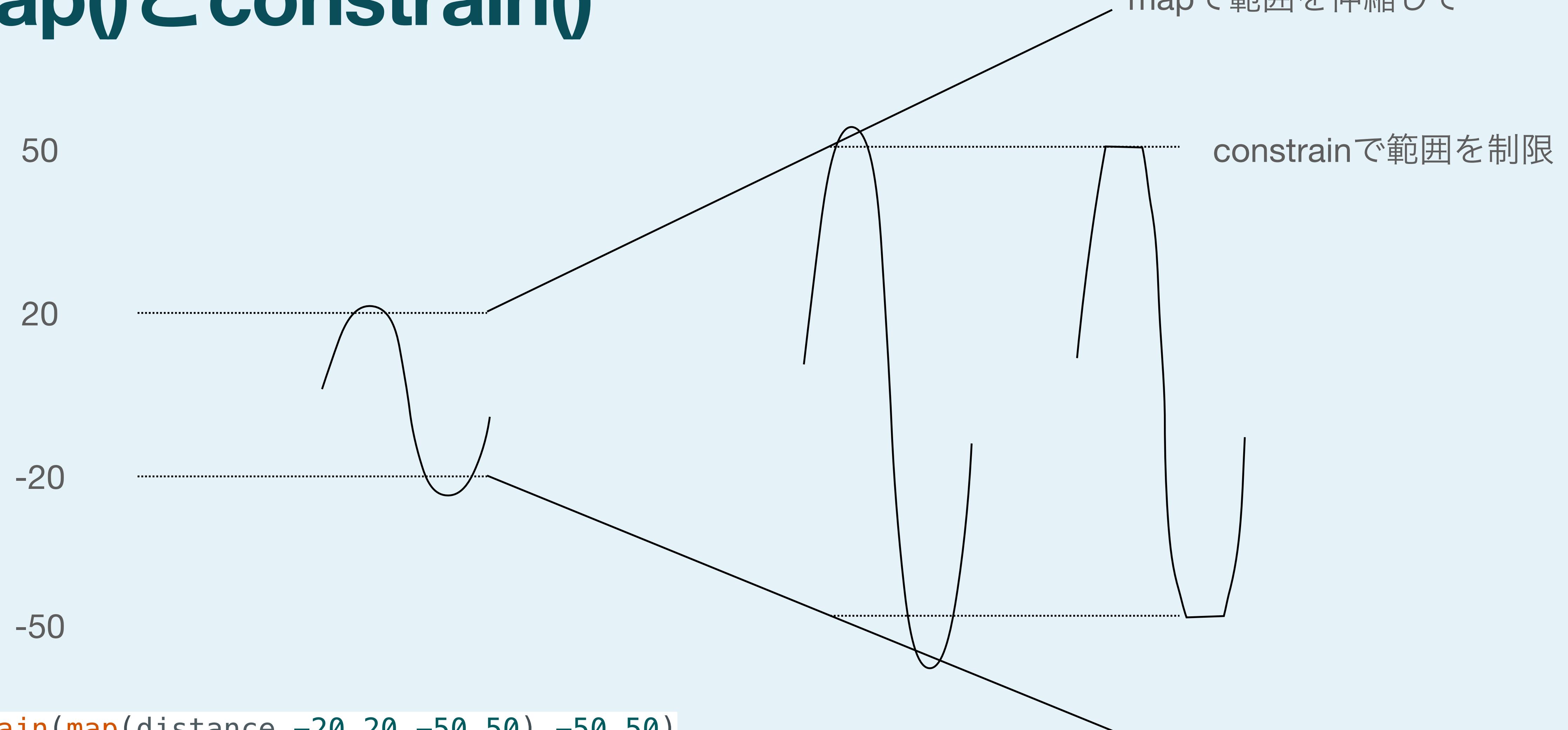


```
void setup() {
  pinMode(A0, INPUT);
  pinMode(3, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int distance = analogRead(A0);
  int output = constrain(map(distance, 0, 500, 0, 255), 0, 255);
  analogWrite(3, output);
  Serial.print(distance);
  Serial.print(',');
  Serial.println(output);
  delay(40);
}
```

analogWriteを使って連続的に変化させる  
出力範囲は0~255なので範囲の変換が必要

# map()とconstrain()



`constrain(map(distance, -20, 20, -50, 50), -50, 50)`

に対して、入力-30~30ぐらいの値が入ってきた時のイメージ

```
void setup() {
  pinMode(A0, INPUT);
  pinMode(3, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int distance = analogRead(A0);
  int output = constrain(map(distance, 0, 500, 0, 255), 0, 255);
  analogWrite(3, output);
  Serial.print(distance);
  Serial.print(',');
  Serial.println(output);
  delay(40);
}
```

カンマ区切りでシリアルにprintすると、  
シリアルプロッタで複数の値を観測できる

analogWriteを使って連続的に変化させる