

# 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

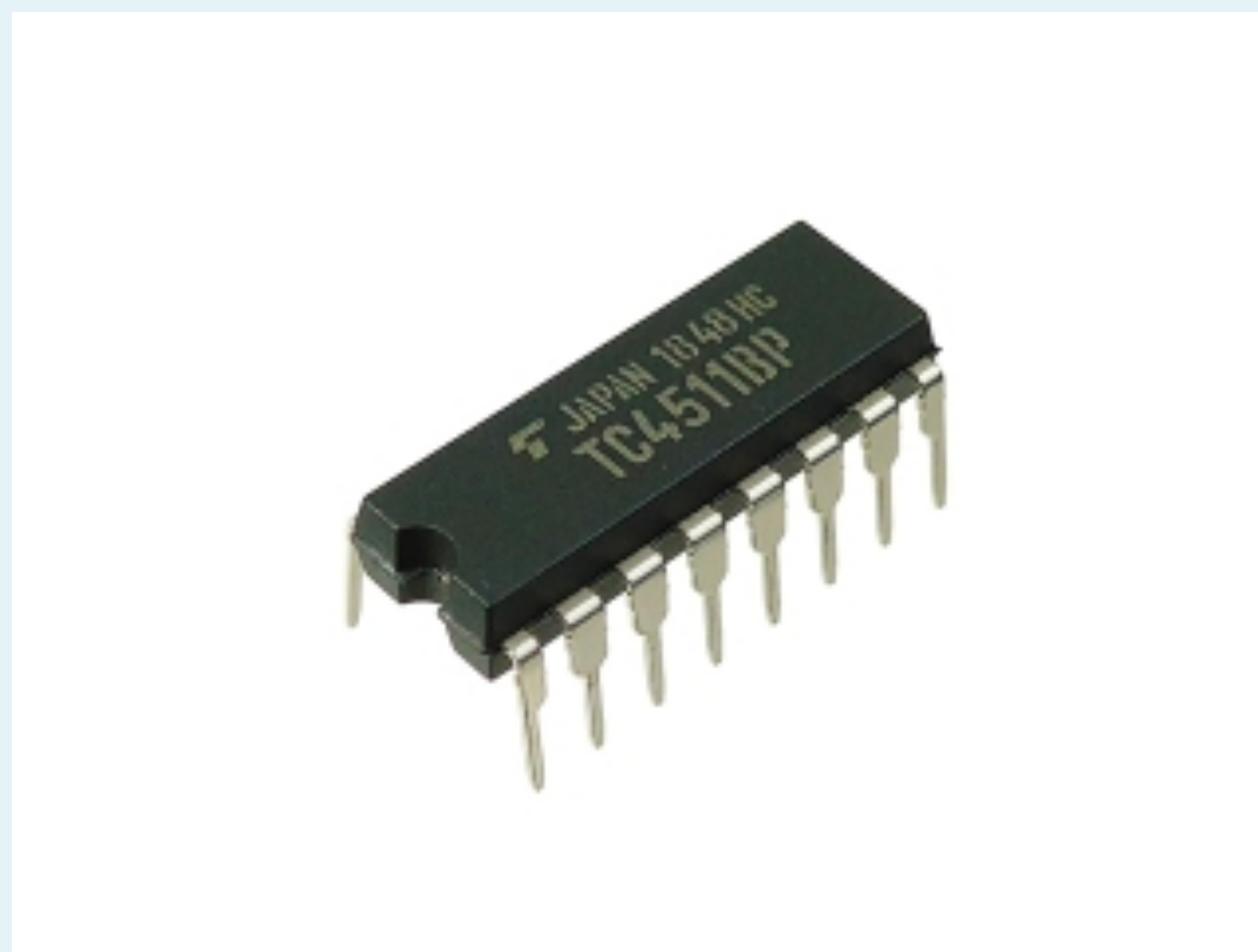
2023.05.19 松浦知也 ([matsura.tomoya@noc.geidai.ac.jp](mailto:matsura.tomoya@noc.geidai.ac.jp) [teach@matsuuratomoya.com](mailto:teach@matsuuratomoya.com))



# 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

## 真理値表

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

\* : Don't care

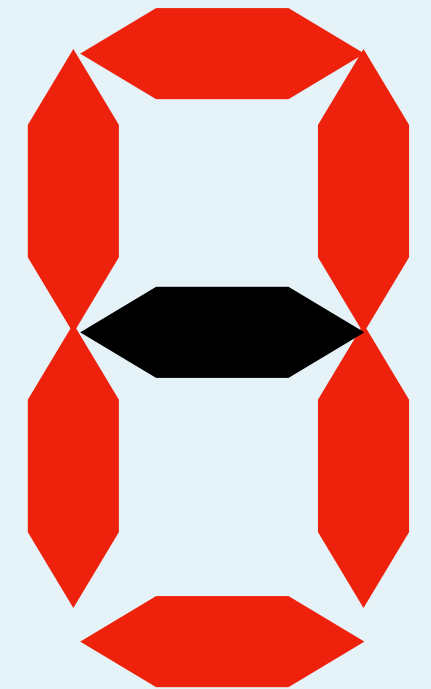
ΔΔ : Depends upon the BCD code previously applied when LE "L"

## 真理値表

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

\* : Don't care

ΔΔ : Depends upon the BCD code previously applied when LE "L"

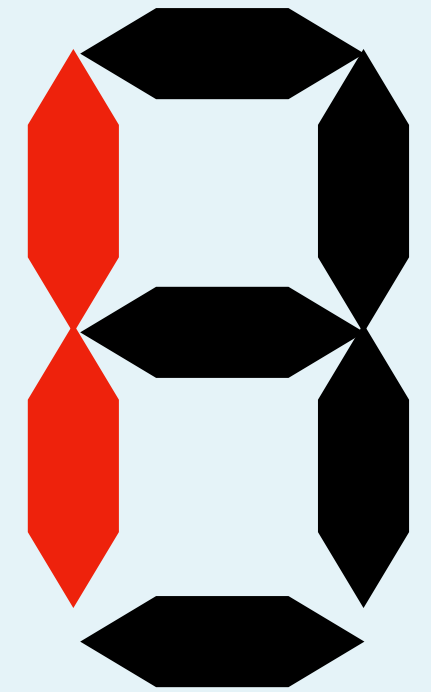


## 真理値表

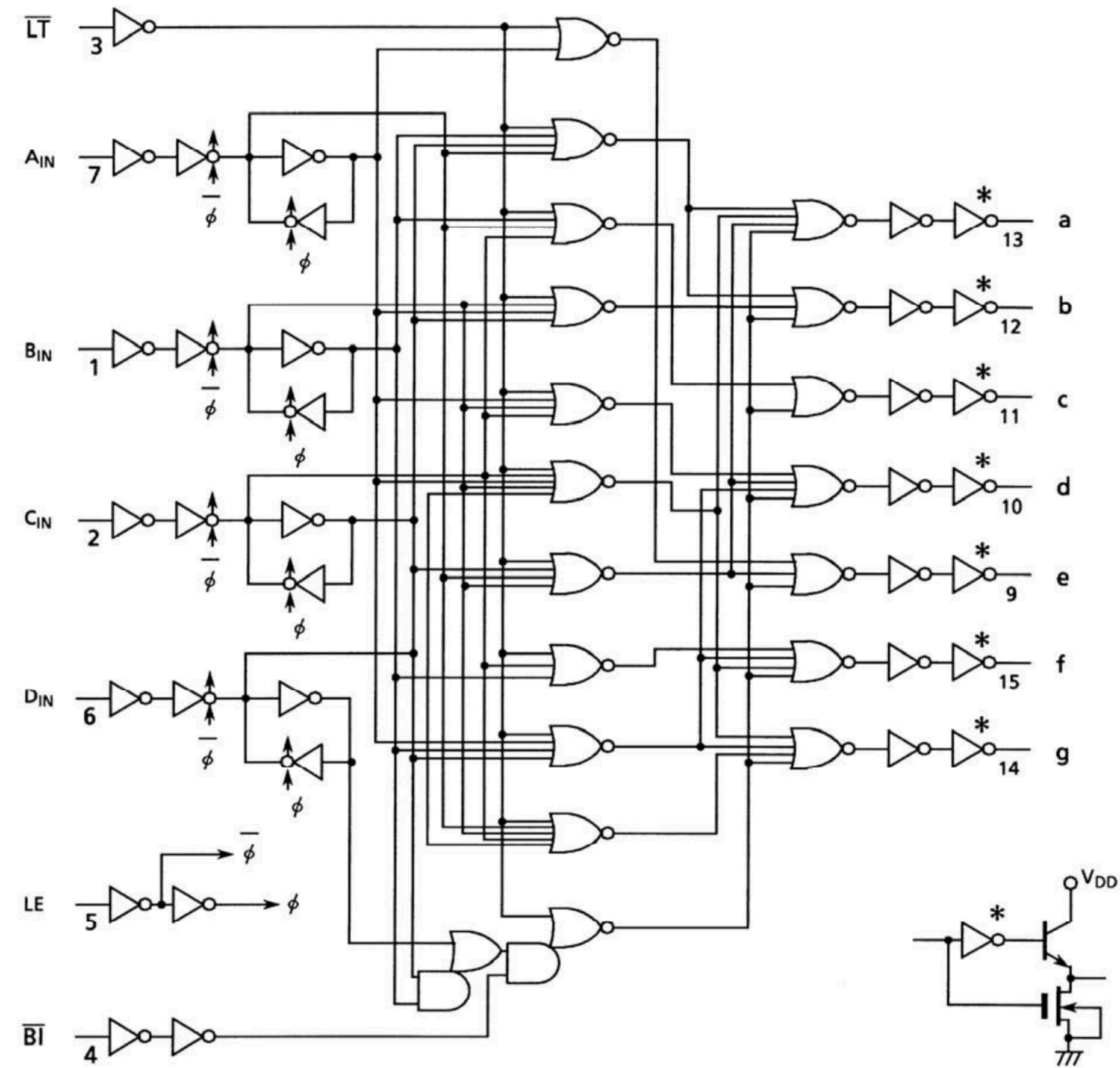
Inputs							Outputs							Display Mode
LE	$\overline{BI}$	$\overline{LT}$	D	C	B	A	a	b	c	d	e	f	g	
*	*	L	*	*	*	*	H	H	H	H	H	H	H	8
*	L	H	*	*	*	*	L	L	L	L	L	L	L	Blank
L	H	H	L	L	L	L	H	H	H	H	H	H	L	0
L	H	H	L	L	L	H	L	H	H	L	L	L	L	1
L	H	H	L	L	H	L	H	H	L	H	H	L	H	2
L	H	H	L	L	H	H	H	H	H	H	L	L	H	3
L	H	H	L	H	L	L	L	H	H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

\* : Don't care

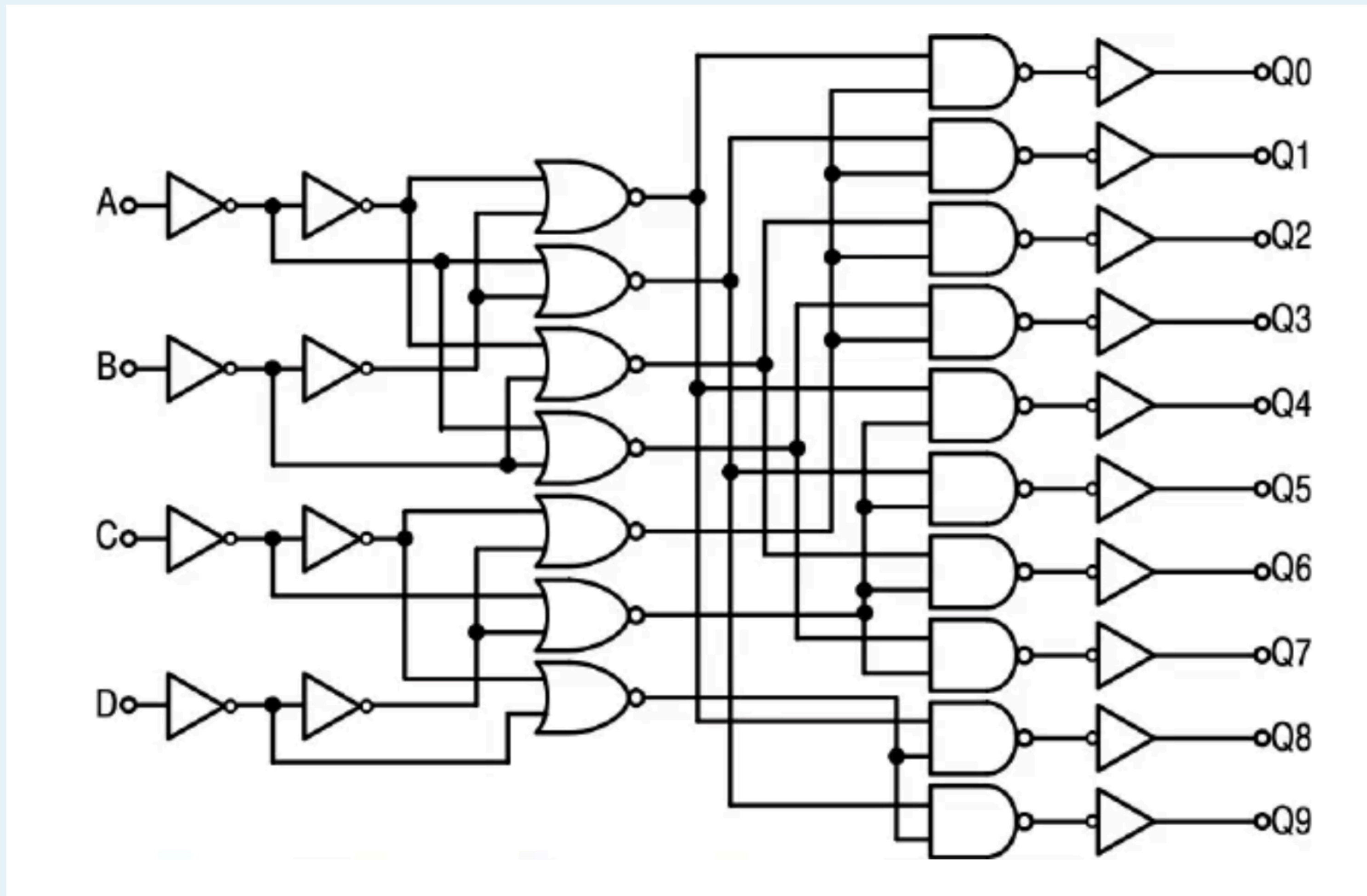
ΔΔ : Depends upon the BCD code previously applied when LE "L"



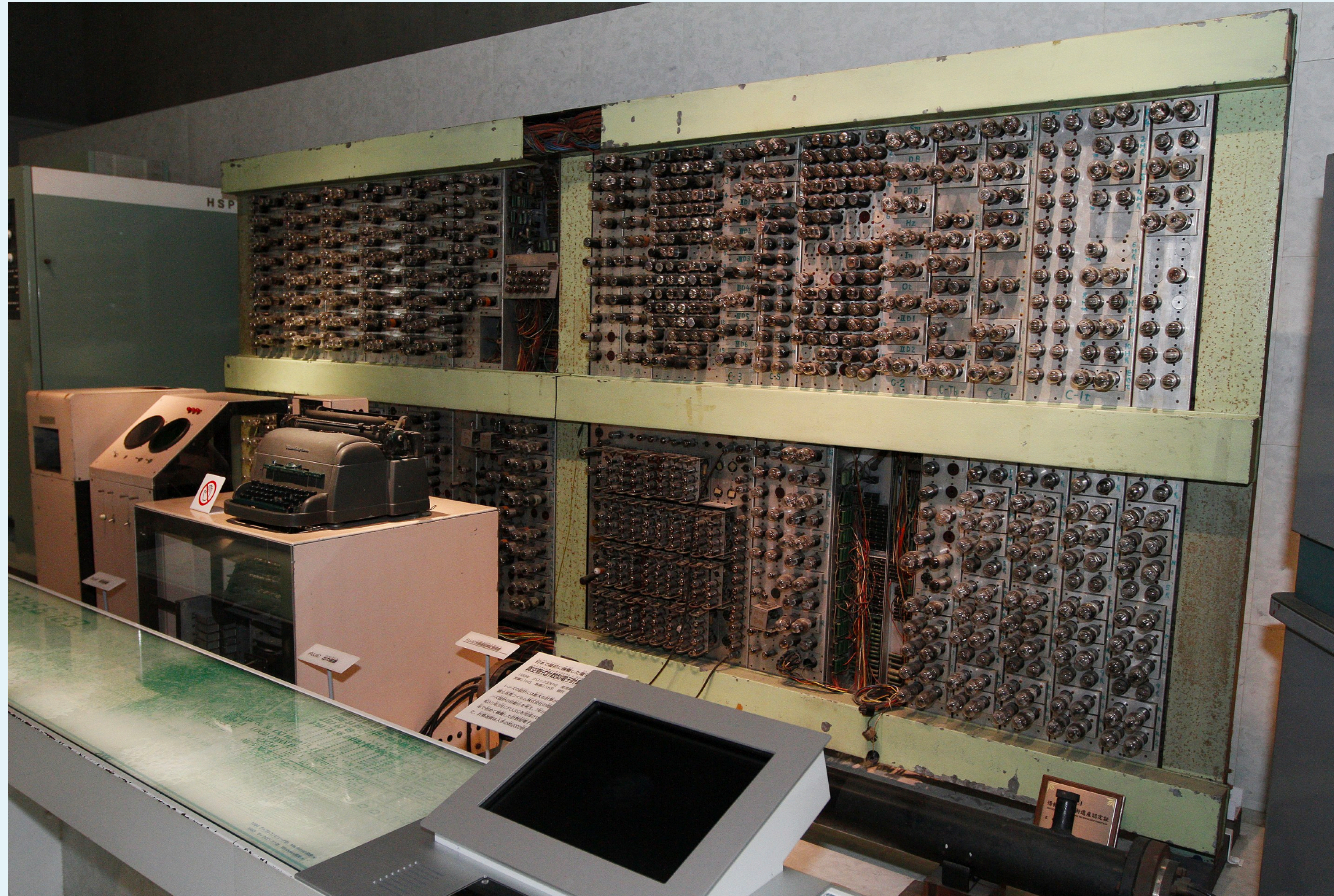
論理図







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)

Morio - 投稿者自身による著作物, CC 表示-継承 3.0, <https://commons.wikimedia.org/w/index.php?curid=12663684>

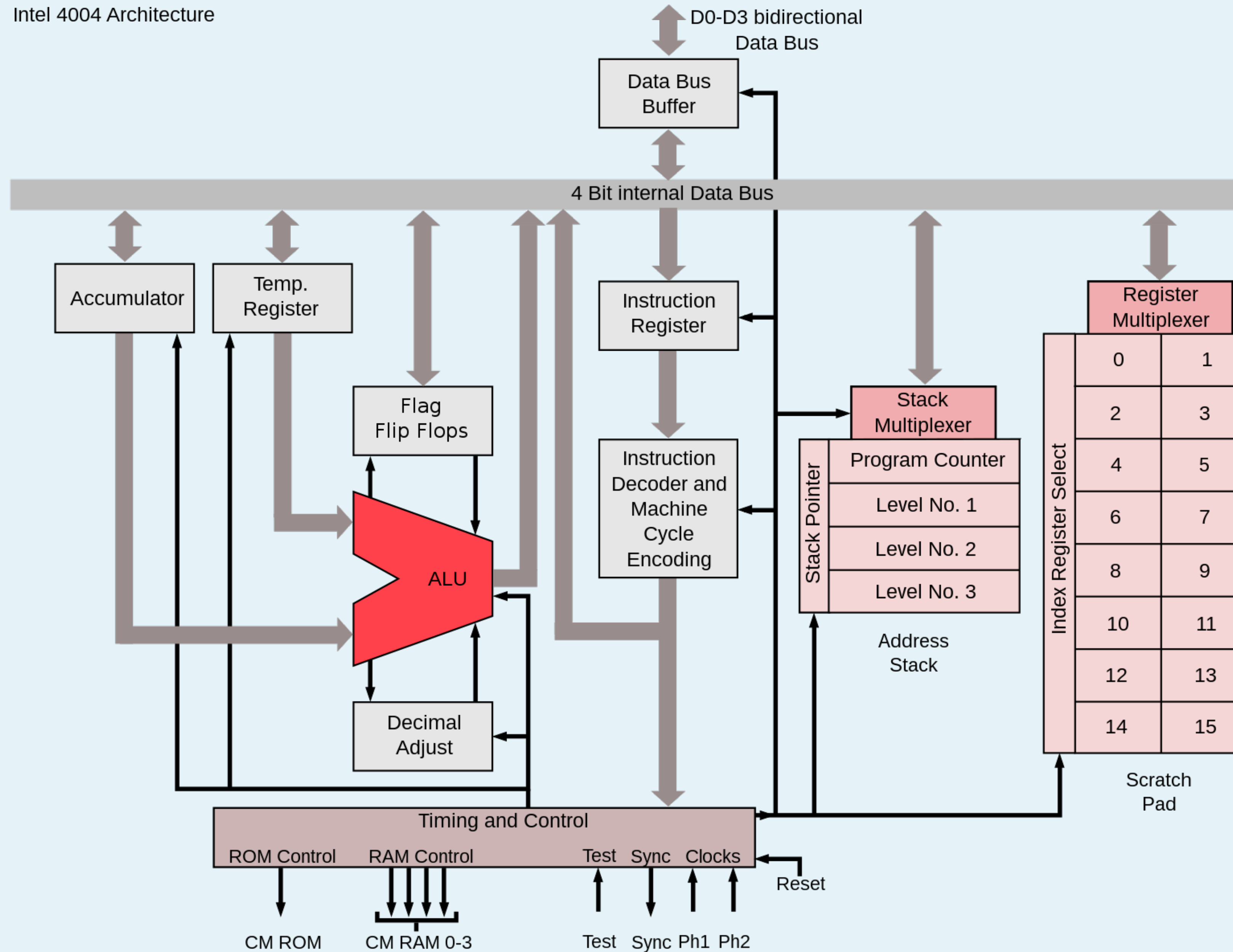


FUJIC(日本で最初のコンピューター、上野の科学博物館で展示されています)

Morio - 投稿者自身による著作物, CC 表示-継承 3.0, <https://commons.wikimedia.org/w/index.php?curid=12663684>

# 電卓とコンピューターの違い

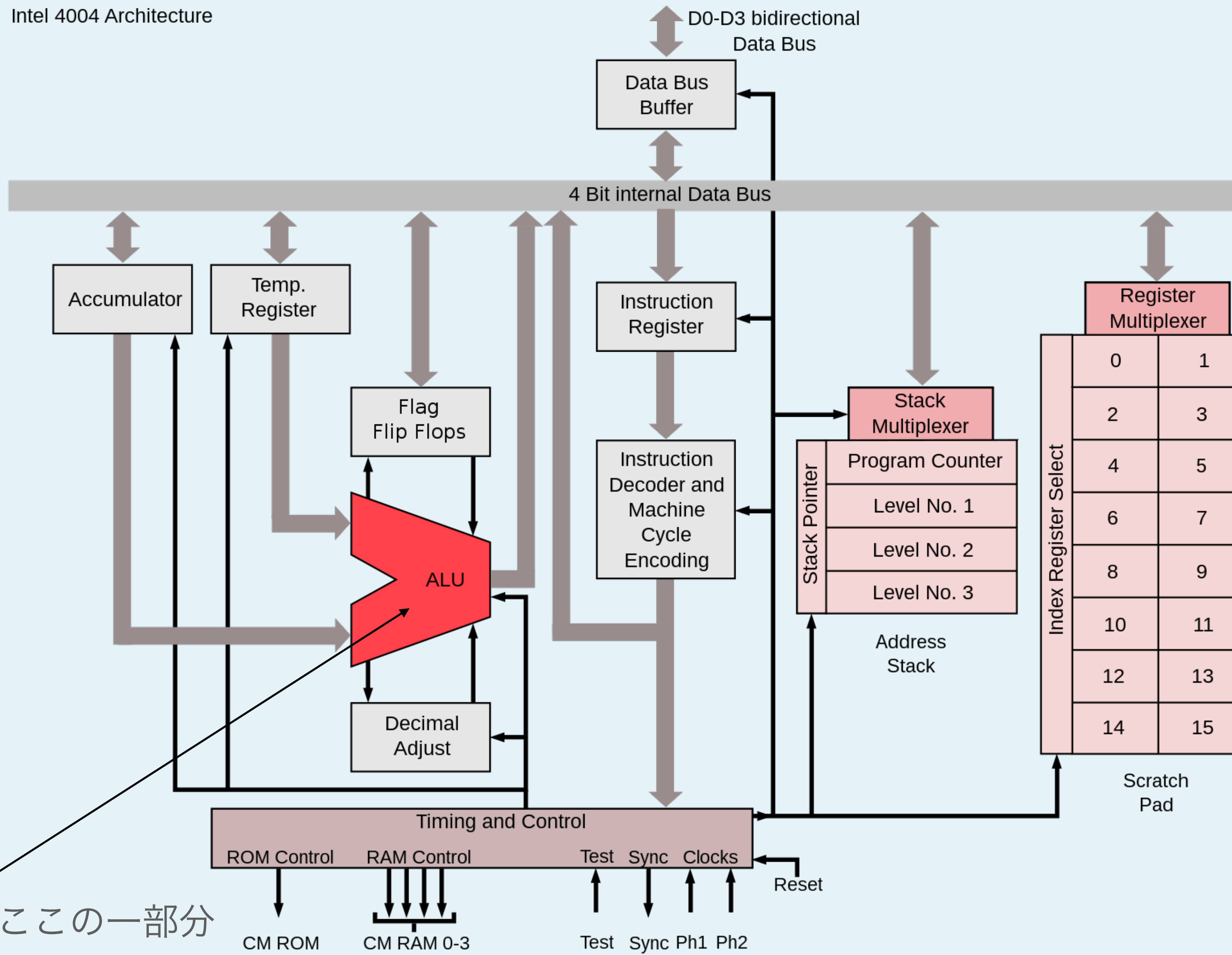
Intel 4004 Architecture



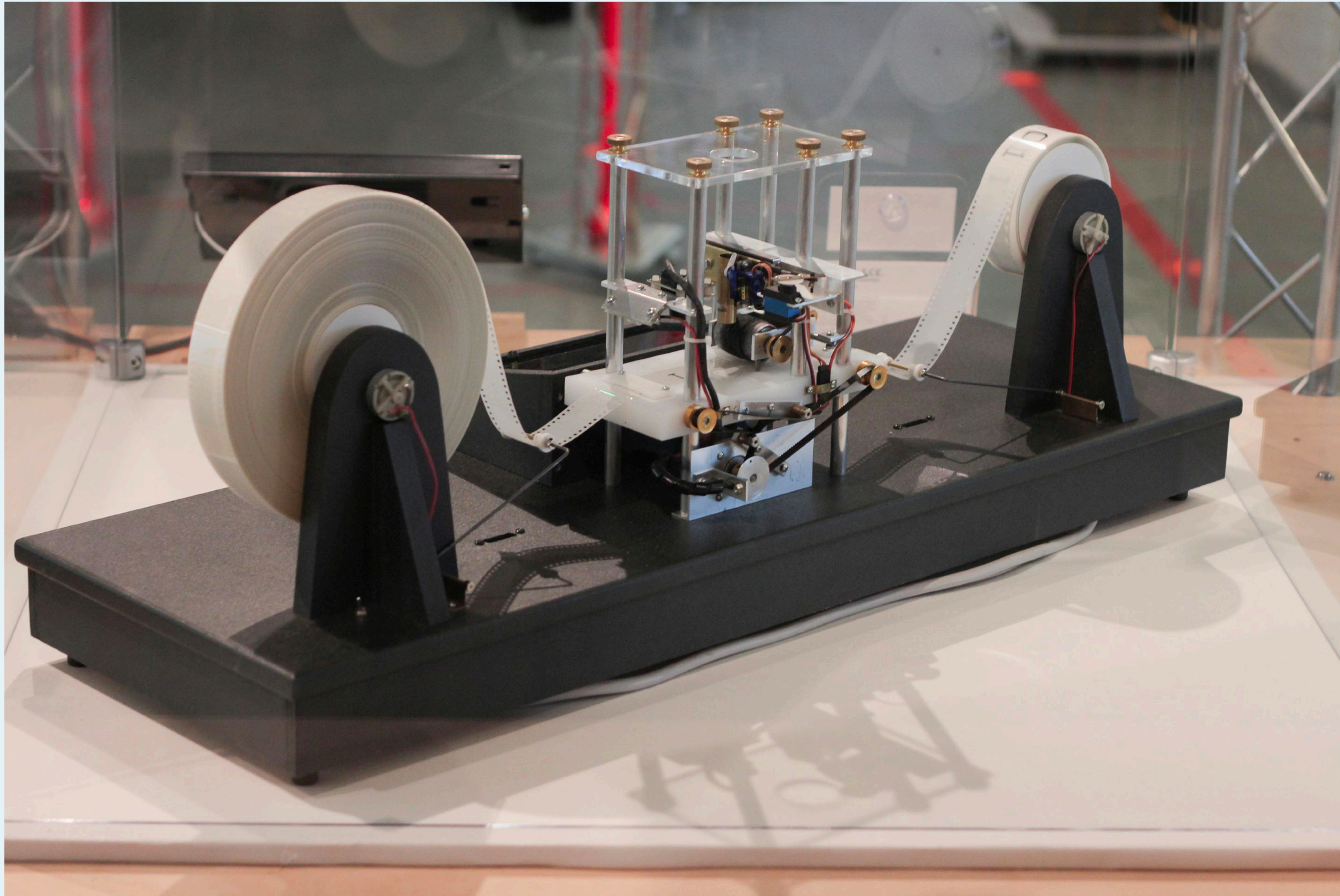
Appaloosa, Intel 4004 architecture, CC 表示-継承 3.0, [https://ja.wikipedia.org/wiki/Intel\\_4004#/media/](https://ja.wikipedia.org/wiki/Intel_4004#/media/)

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

+

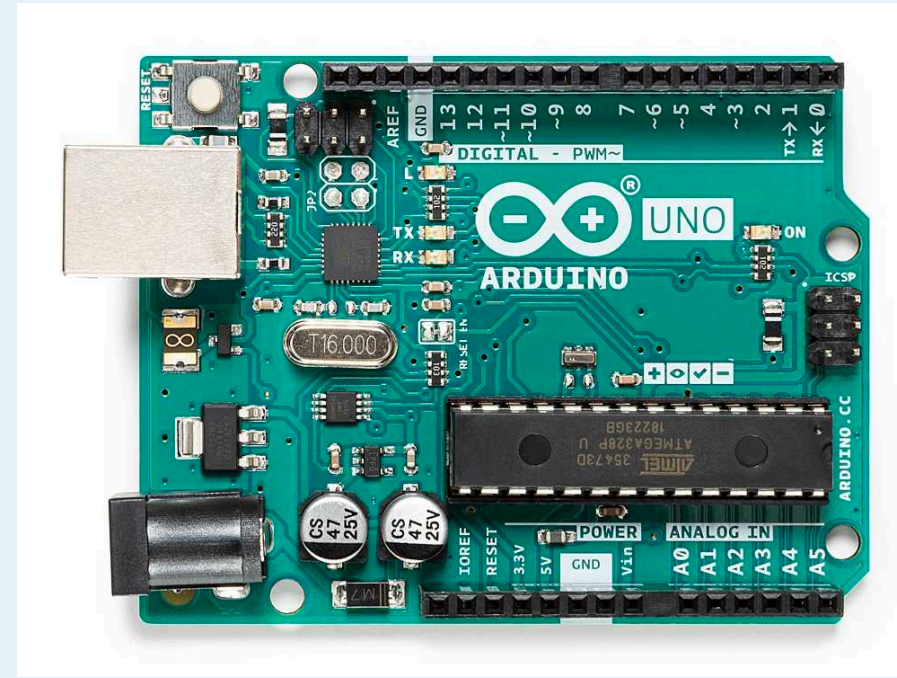


簡単+複数のボードに対応したプログラミング環境

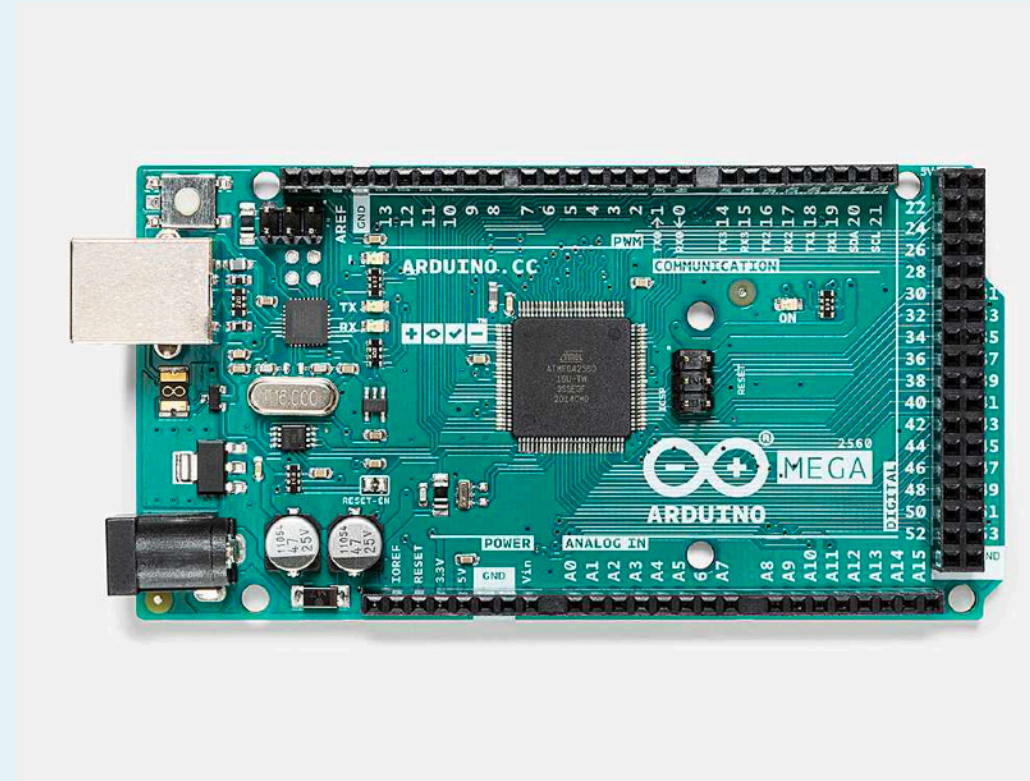
C、C++を簡易化した言語を使用

拡張子は.ino

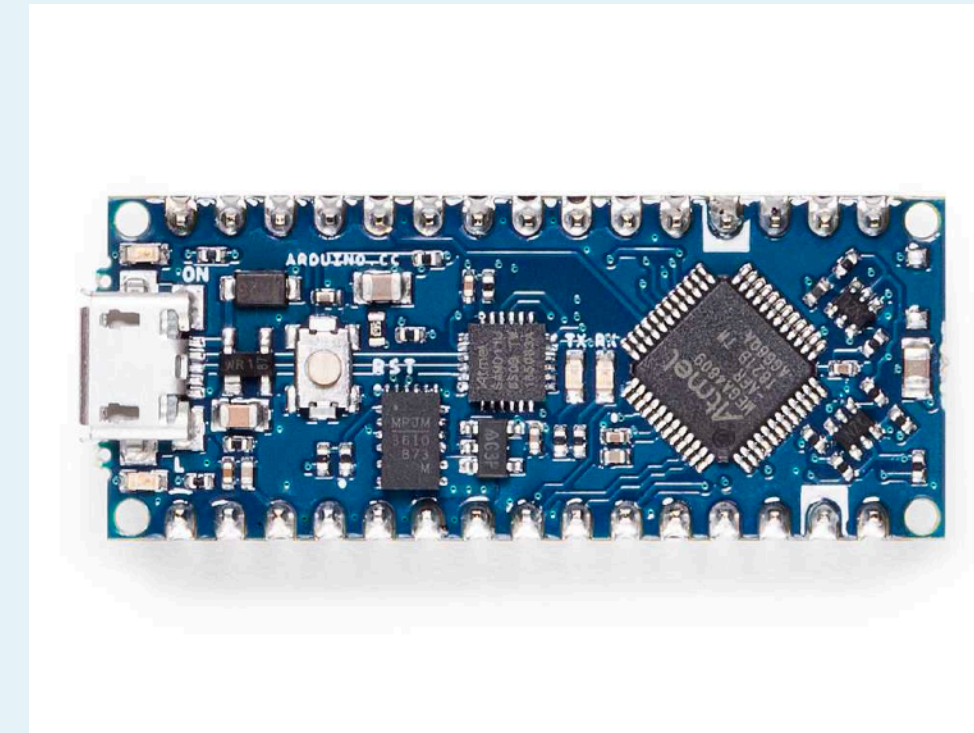
# Arduino公式シリーズ



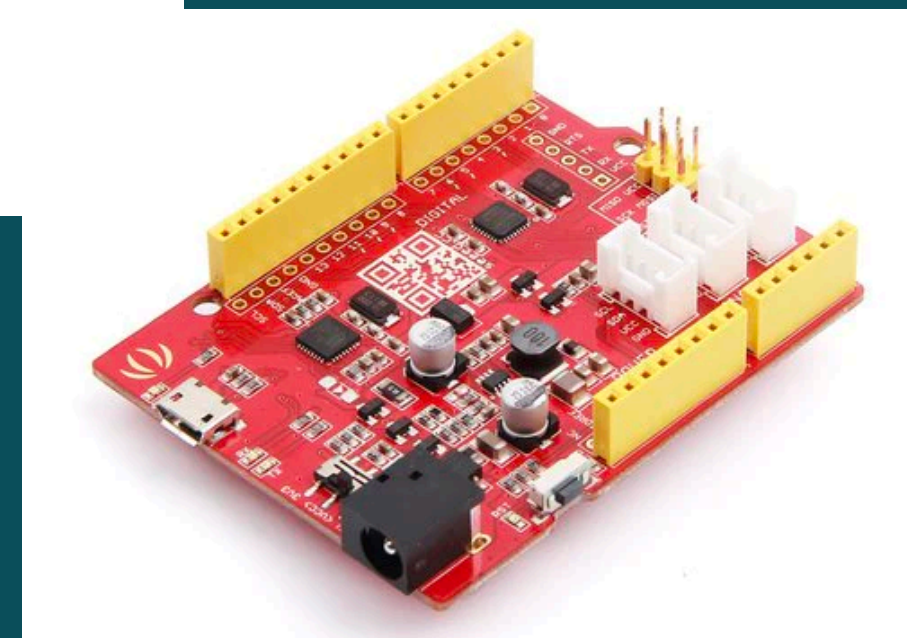
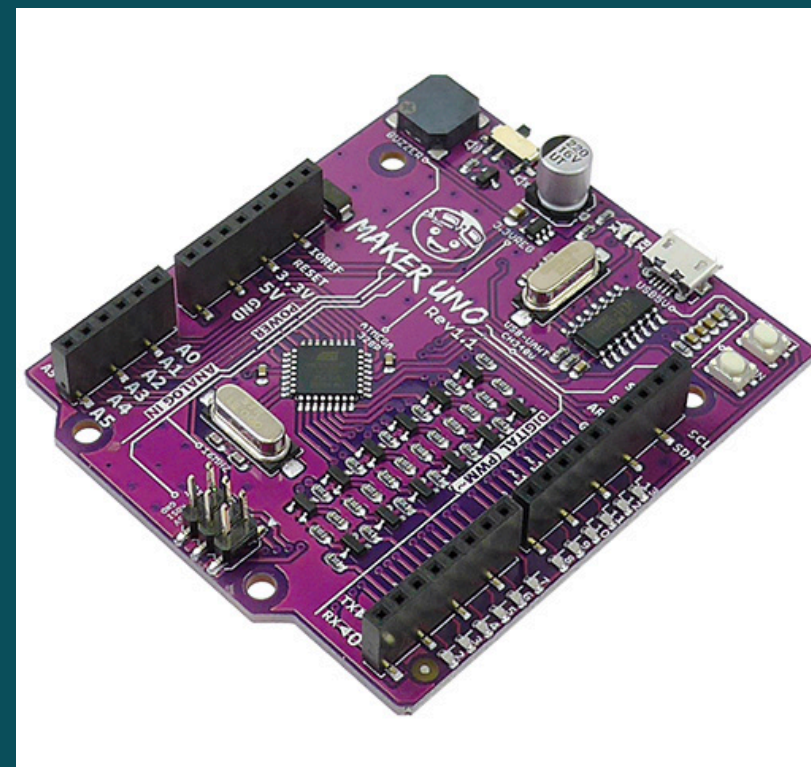
Arduino UNO



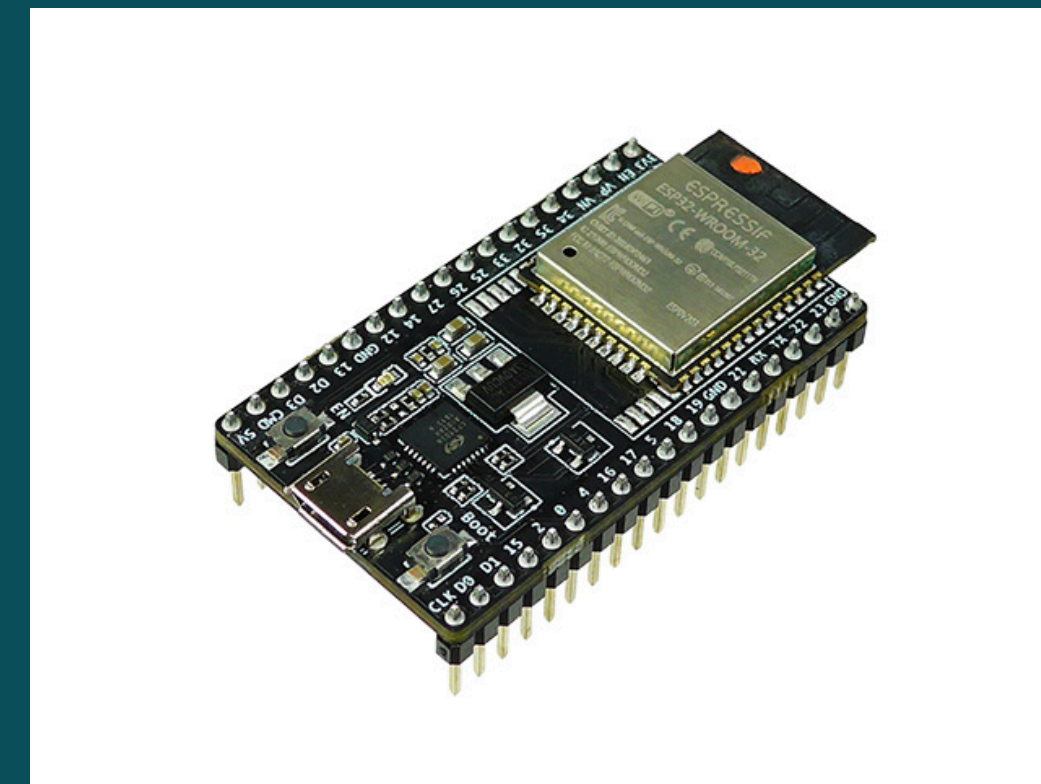
Arduino Mega



Arduino Nano Every

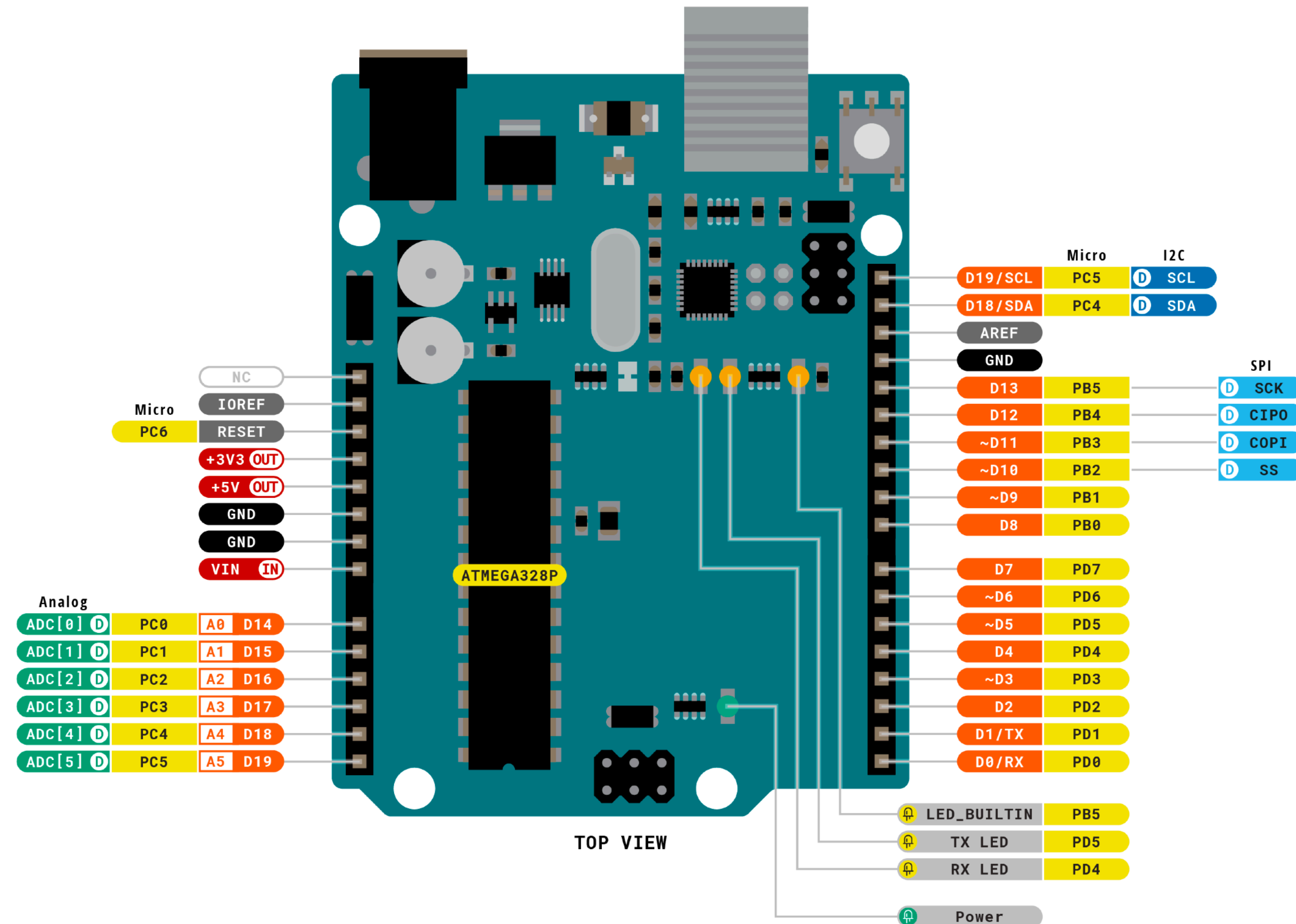


ピン配置互換性




Arduino IDEでの開発対応



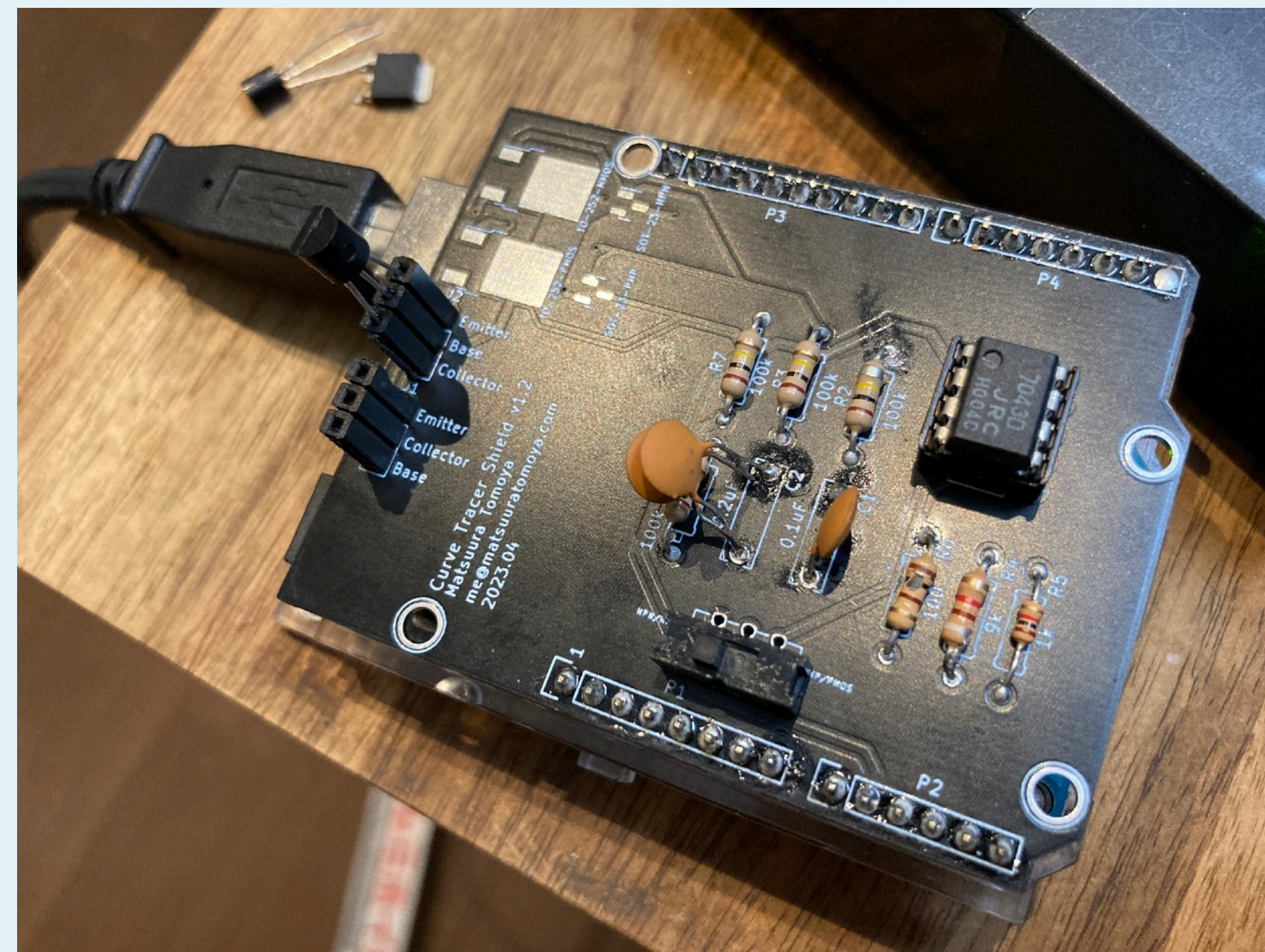
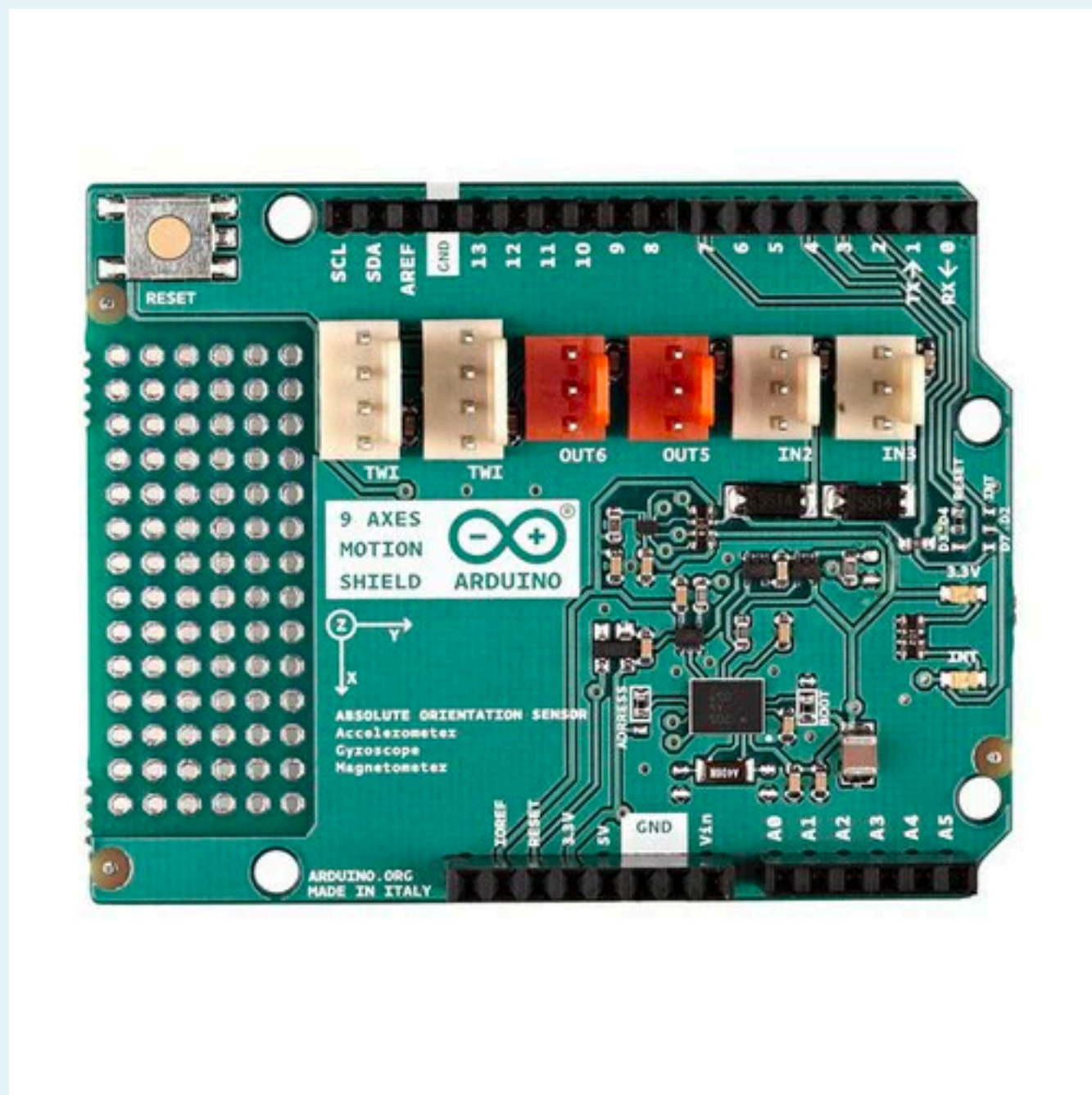


<b>Legend:</b>	<span style="color: orange;">■</span> Digital	<span style="color: blue;">■</span> I2C
<span style="color: red;">■</span> Power	<span style="border: 1px solid orange; display: inline-block; width: 10px; height: 10px;"></span> Analog	<span style="color: lightblue;">■</span> SPI
<span style="color: black;">■</span> Ground	<span style="background-color: yellow; border: 1px solid black; display: inline-block; width: 10px; height: 10px;"></span> Main Part	<span style="color: green;">■</span> Analog

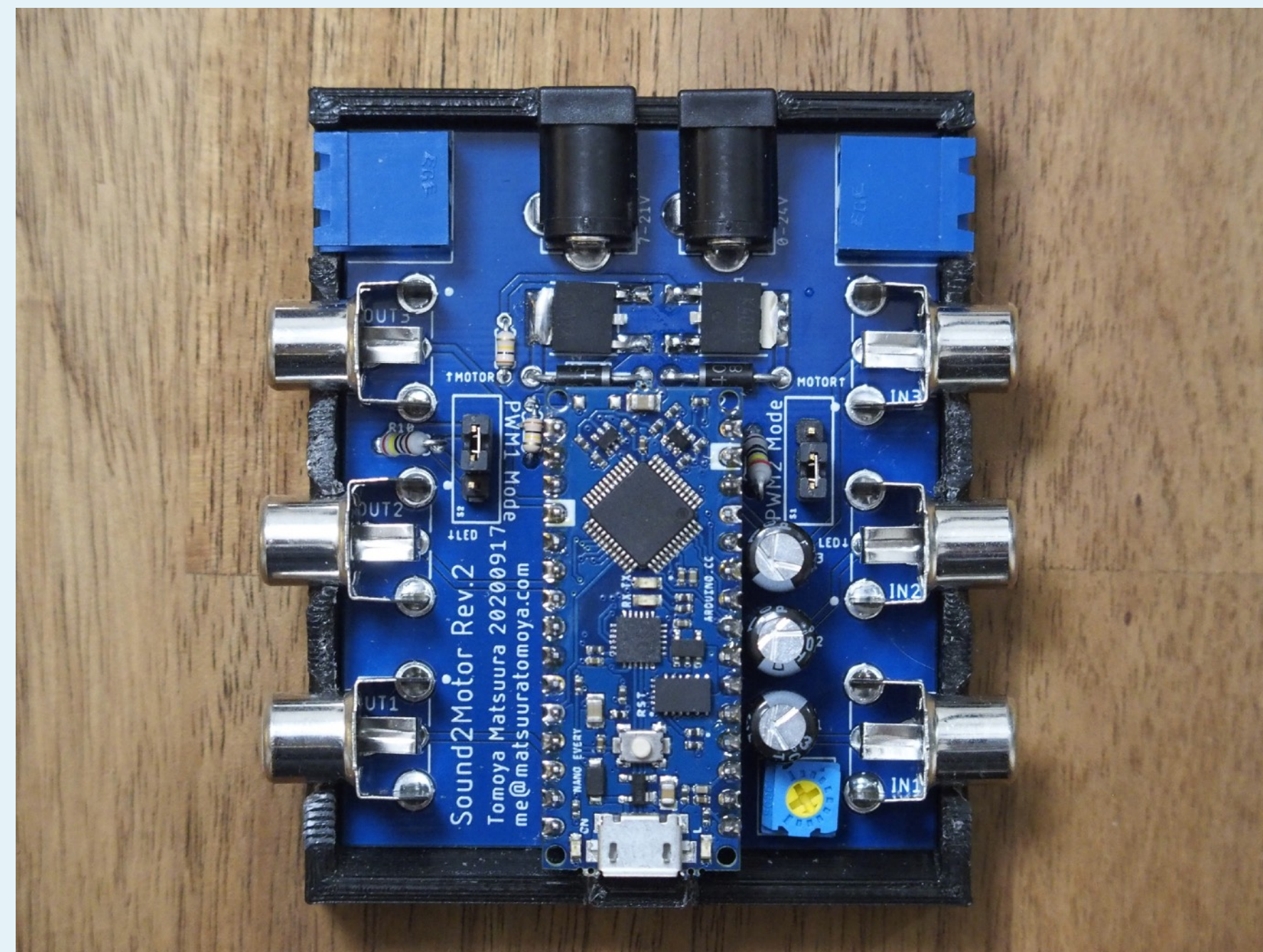
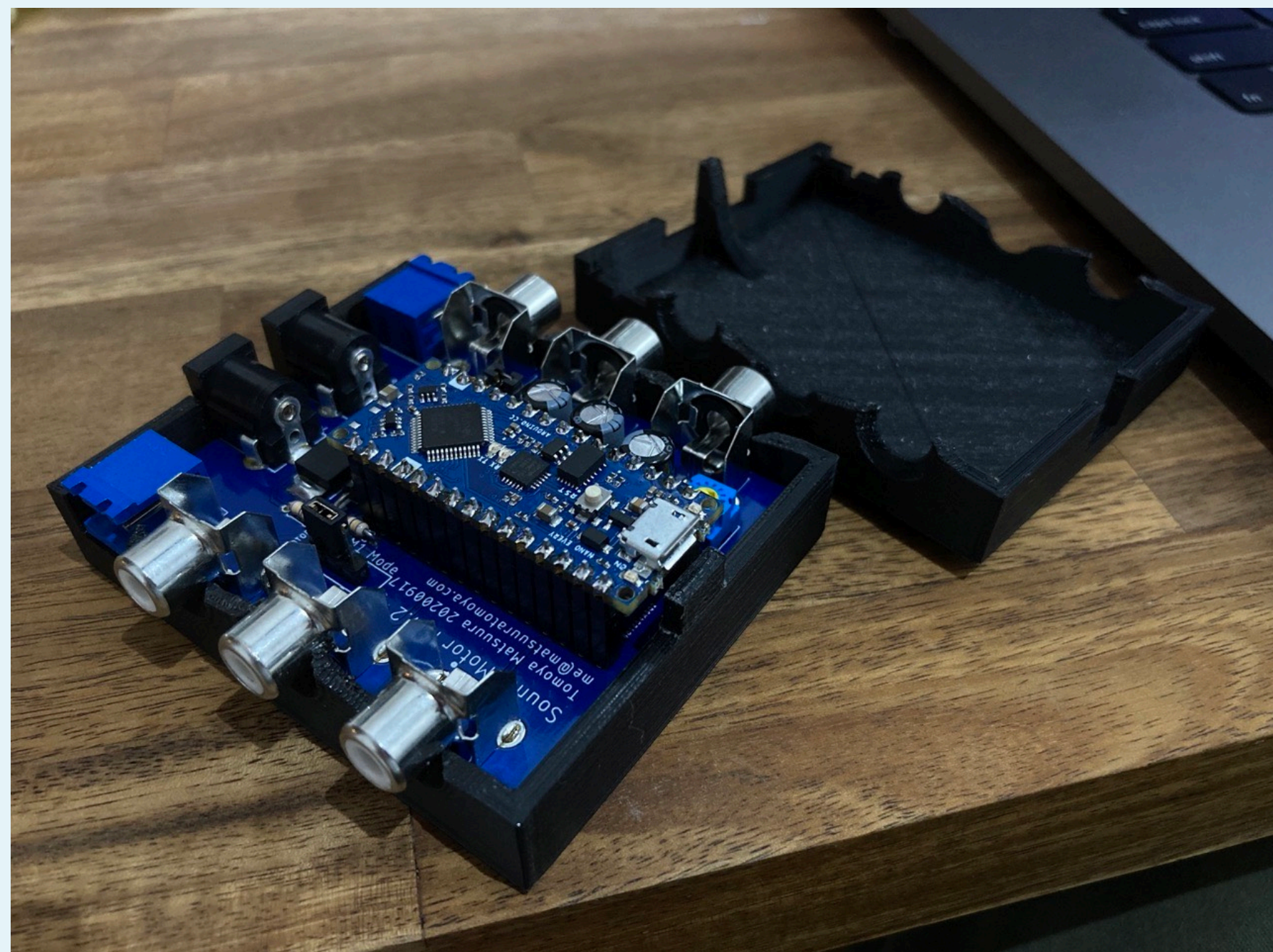


ARDUINO UNO REV3  
SKU code: A000066  
Pinout  
Last update: 6 Oct, 2022

<https://docs.arduino.cc/hardware/uno-rev3>



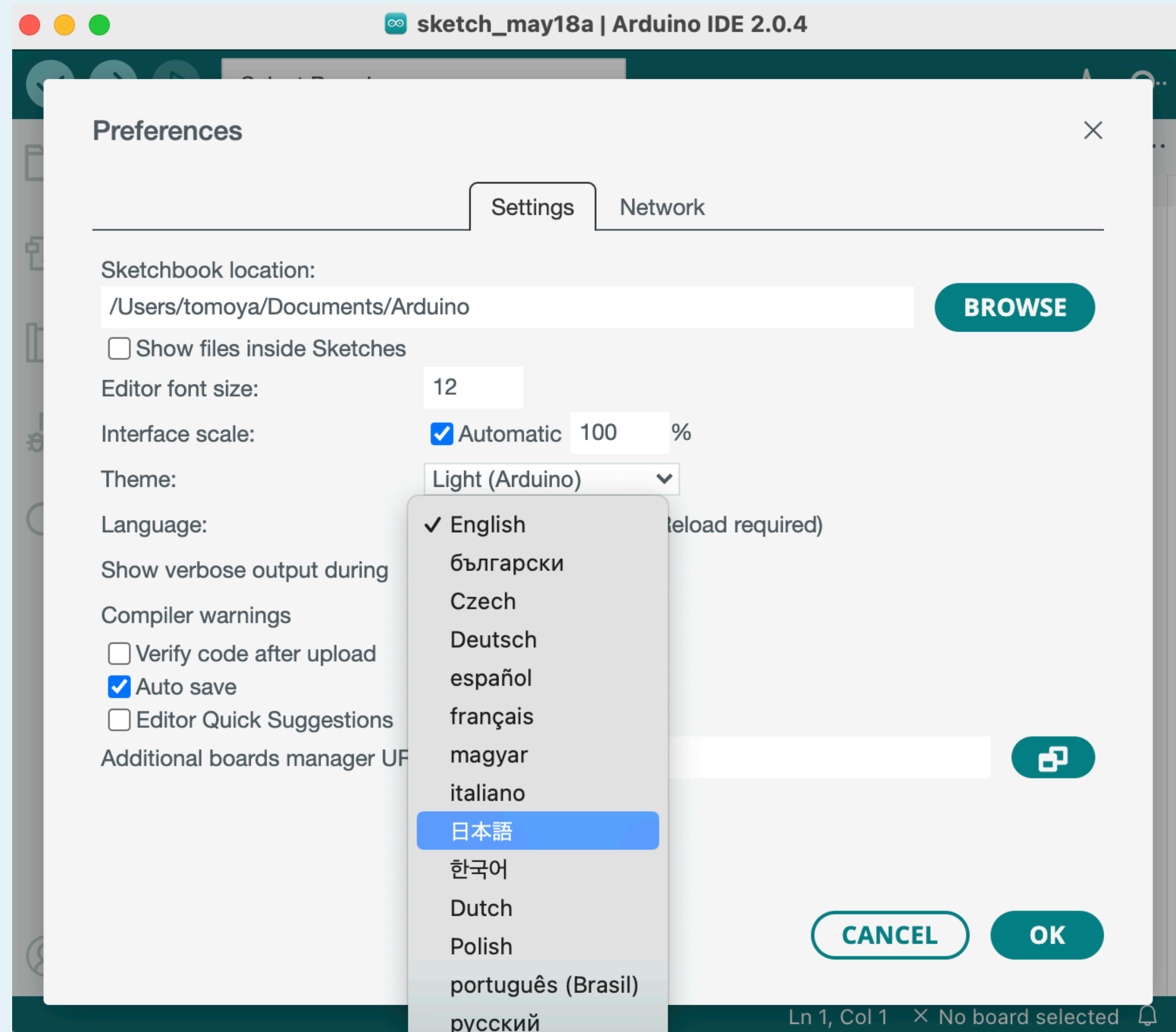
シールドと呼ばれる拡張基板を組み合わせることができる  
(自分でも作れる)



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

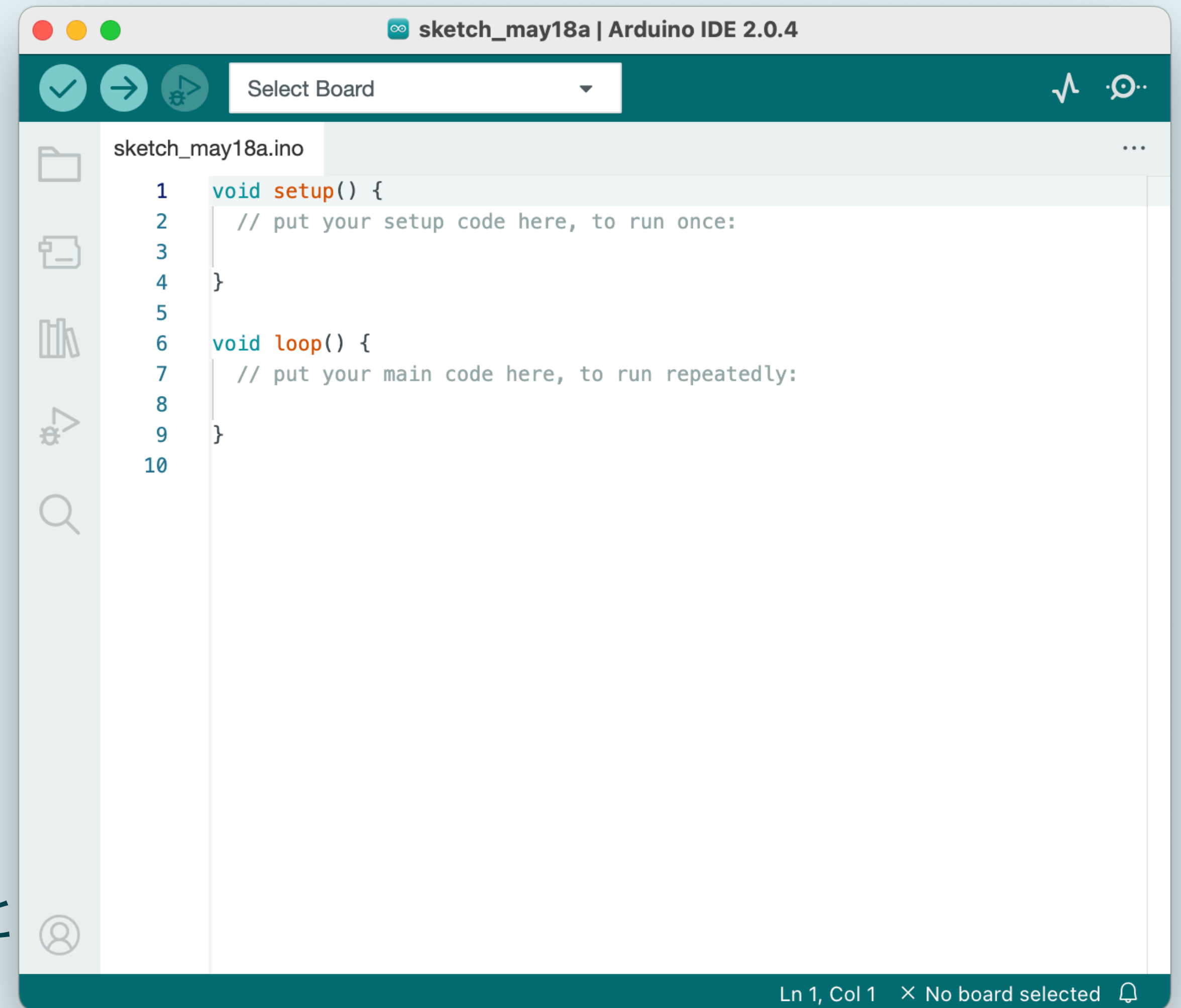
# Arduinoのプログラミング入門



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



```
sketch_may18a.ino
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
```

Ln 1, Col 1 × No board selected

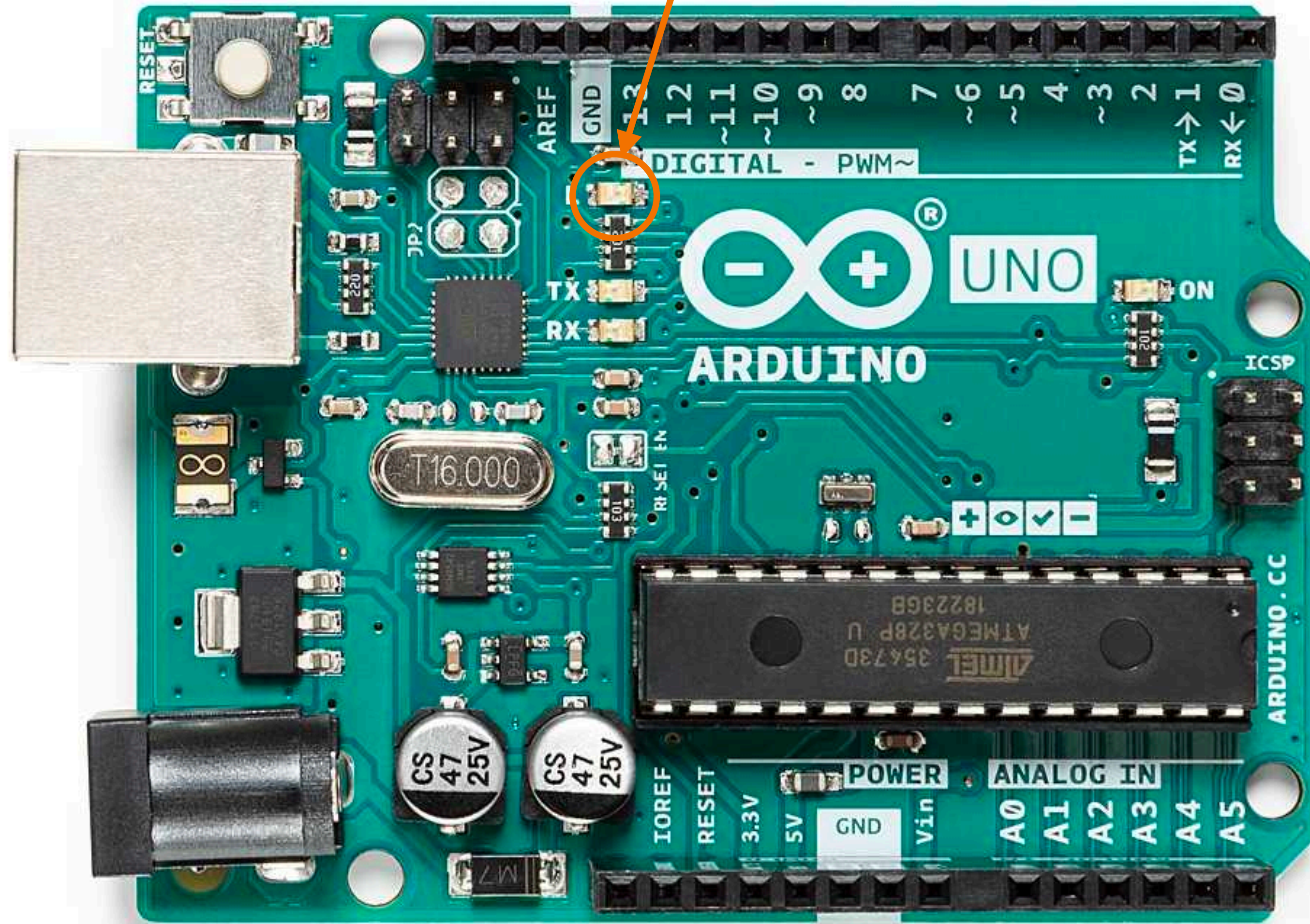


# スケッチ

```
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よう

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 (見やすさ重視で好きに)
- //以降はコメントアウト

```
sketch_may18a.ino
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
```

Output

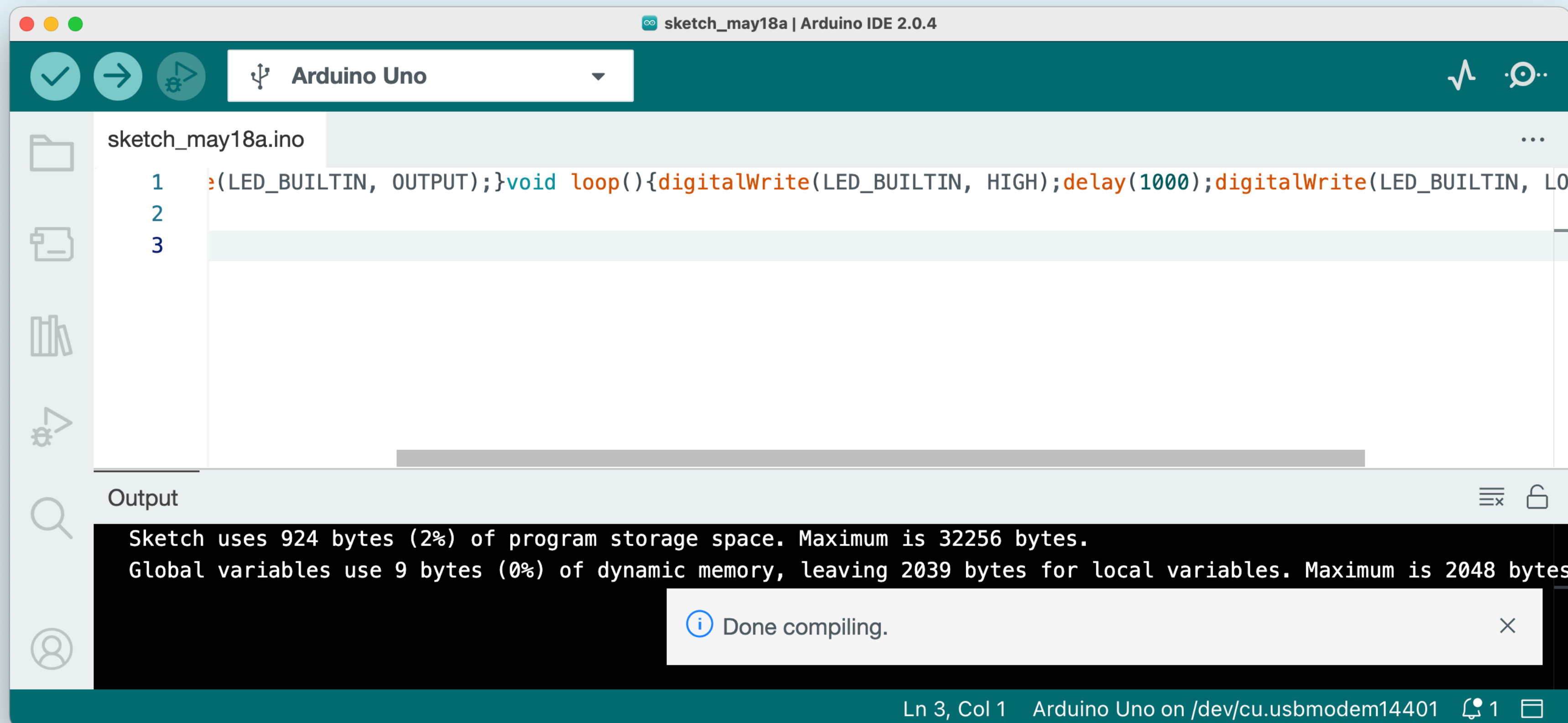
```
^~~~~
/private/var/folders/b3/gftgtjkd1cnczy04xj60w76r0000gn/T/.arduinoIDF
Delay(1000);
^~~~~
delay
```

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/選択範囲を上下と入れ替える
- `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



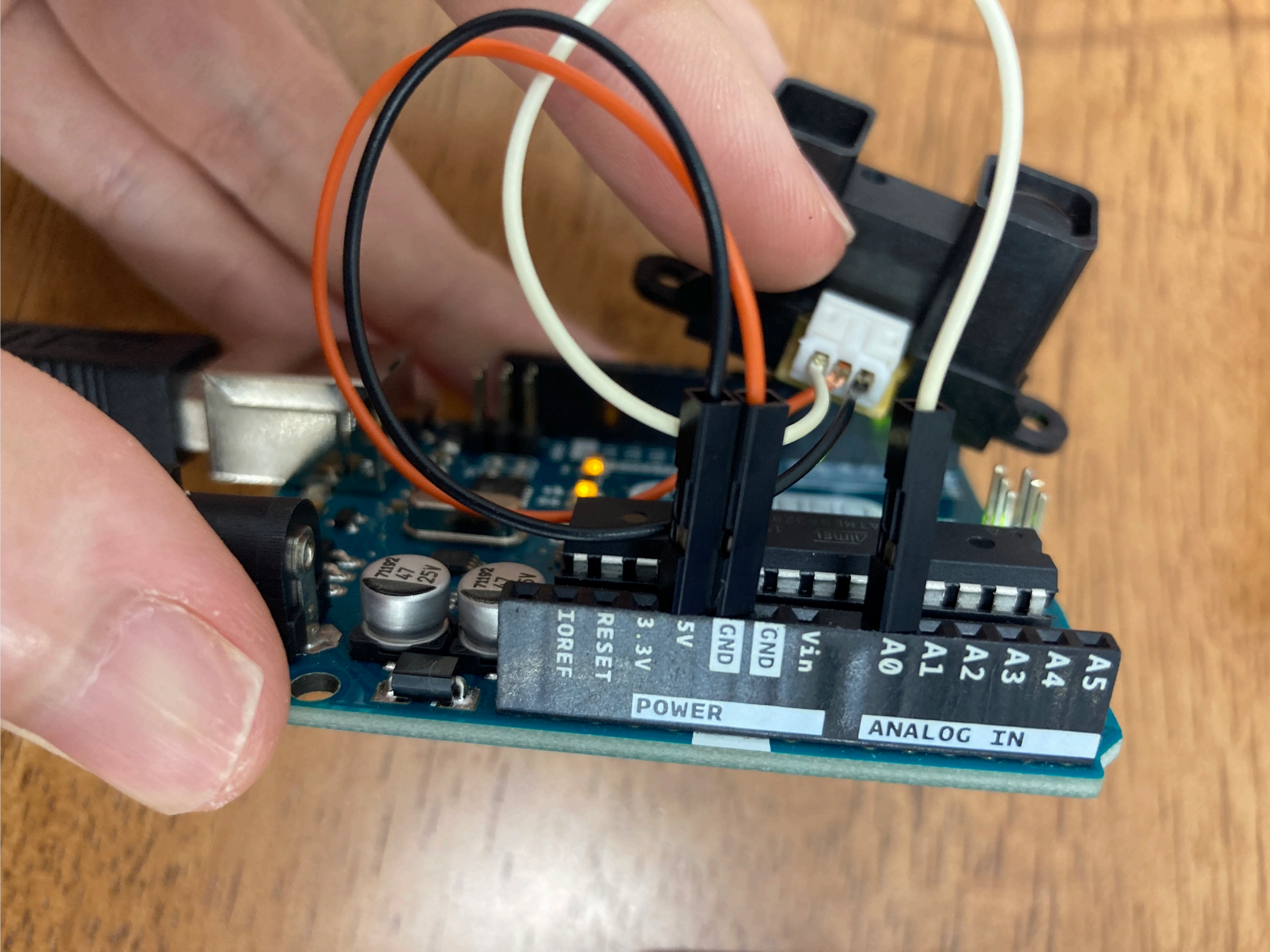
20cm~150cm測れる

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

The screenshot shows the Arduino IDE 2.0.4 interface. The top toolbar contains several icons: a checkmark, a right arrow, a play button, a USB icon, a dropdown menu showing 'Arduino Uno', a 'シリアルプロッタ' (Serial Plotter) button with a graph icon, and a help icon. An arrow points from the text above to the Serial Plotter button. The main editor window displays the following code for 'sketch\_may19b.ino':

```
1 void setup() {  
2   pinMode(A0, INPUT);  
3   Serial.begin(9600);  
4 }  
5  
6 void loop() {  
7   int distance = analogRead(A0);  
8   Serial.println(distance);  
9   delay(20);  
10 }  
11
```

At the bottom right, a notification box says '通知はありません' (No notifications). The status bar at the bottom shows '行 8、列 29 Arduino Uno /dev/cu.usbmodem144401の'.



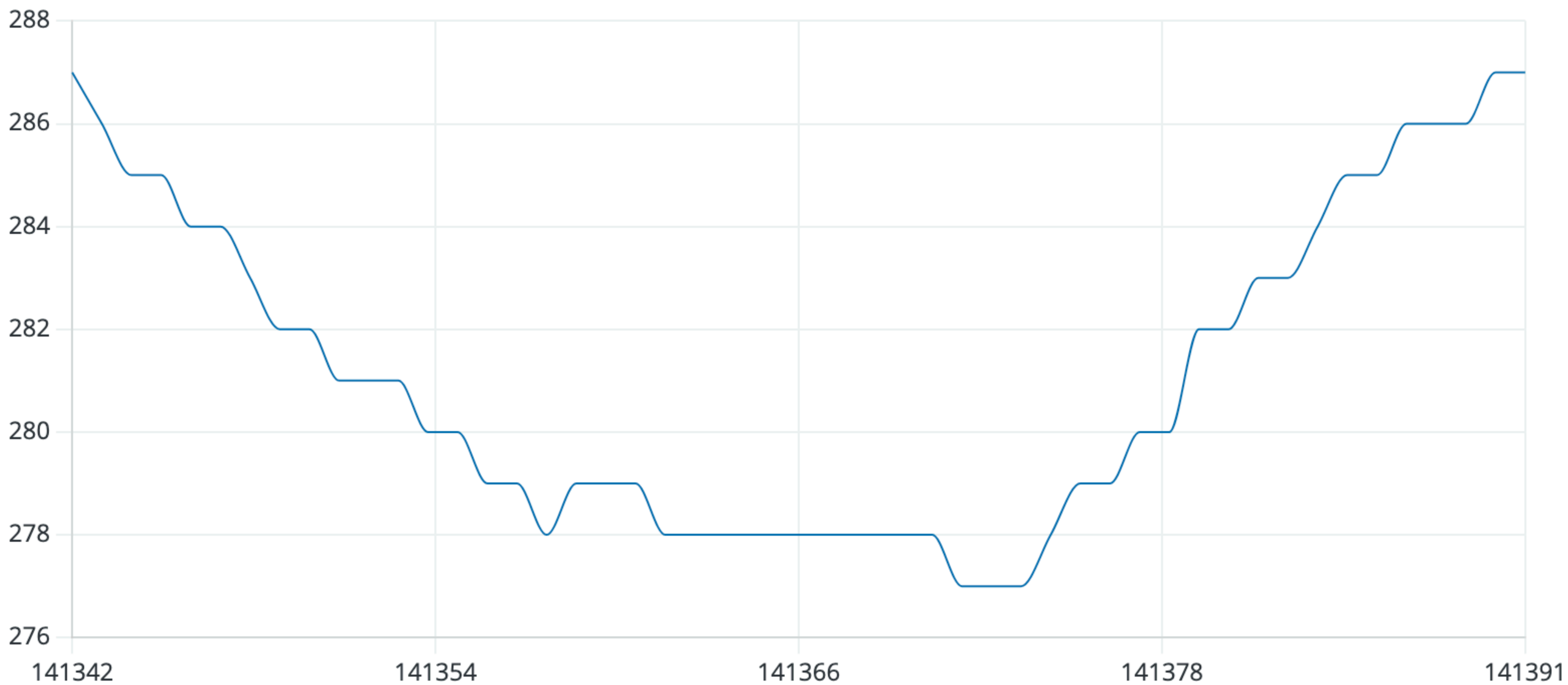
/dev/cu.usbmodem144401

value 1

Interpolate



STOP



Type Message

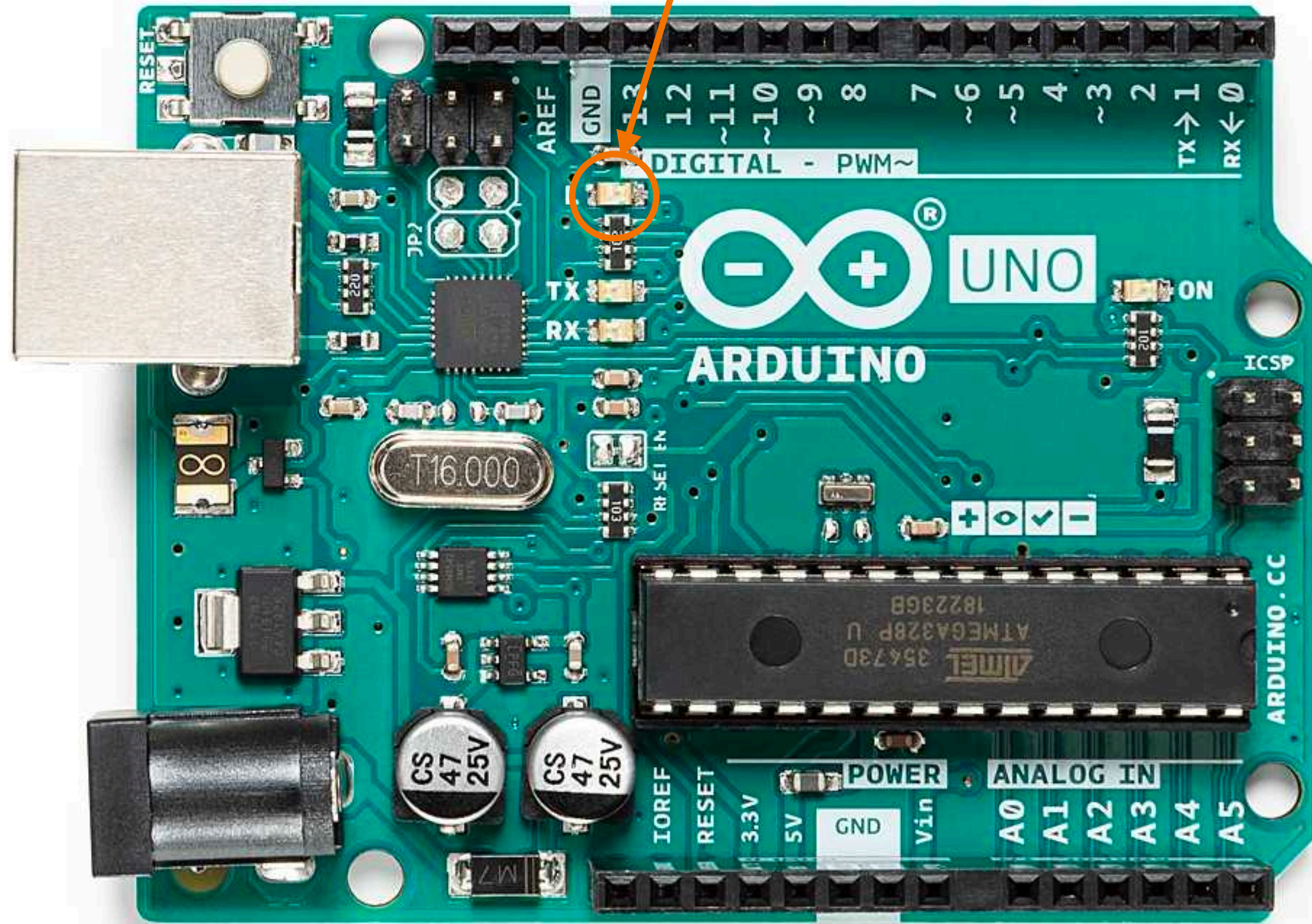
SEND

New Line

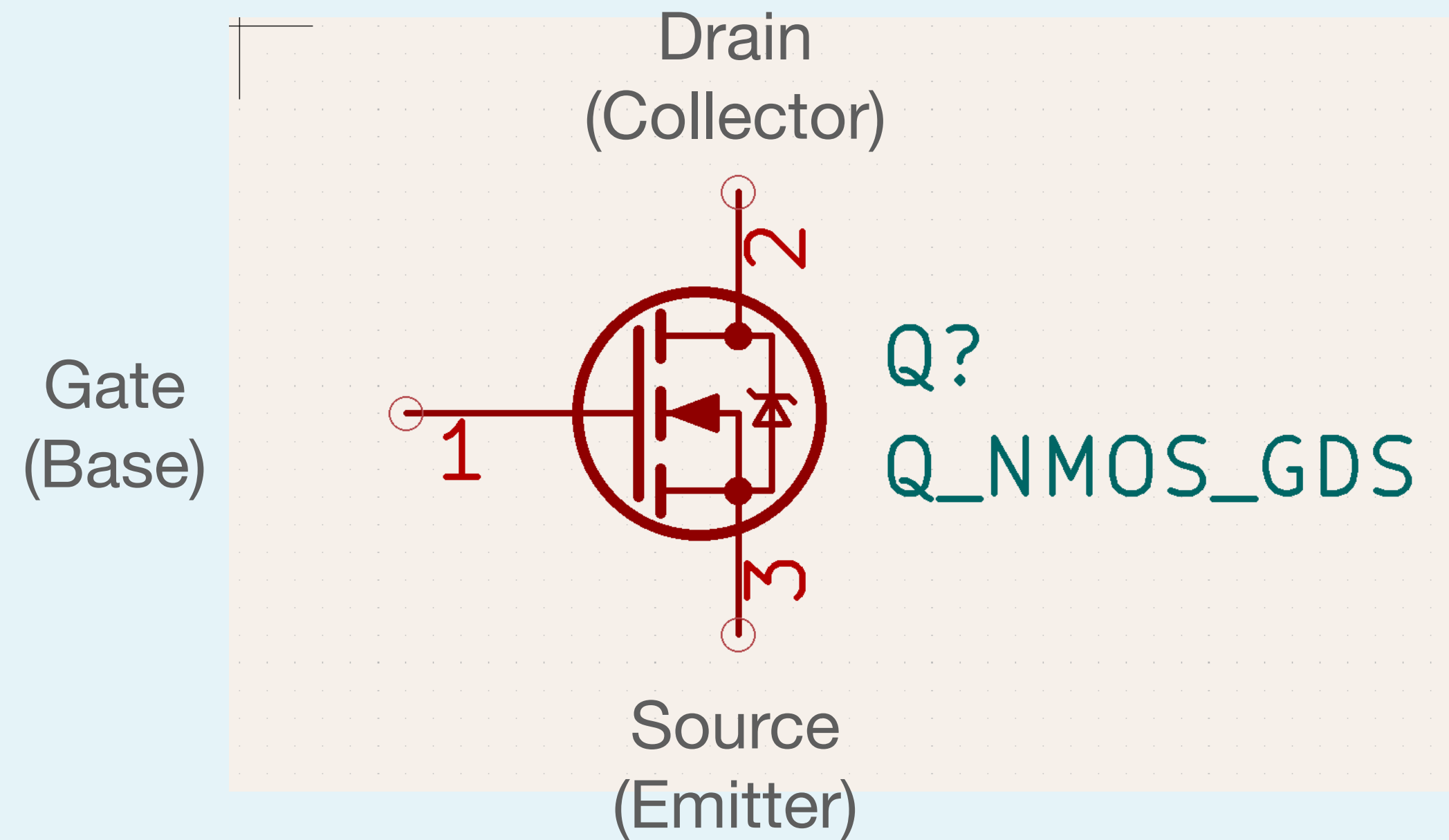
9600 baud

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

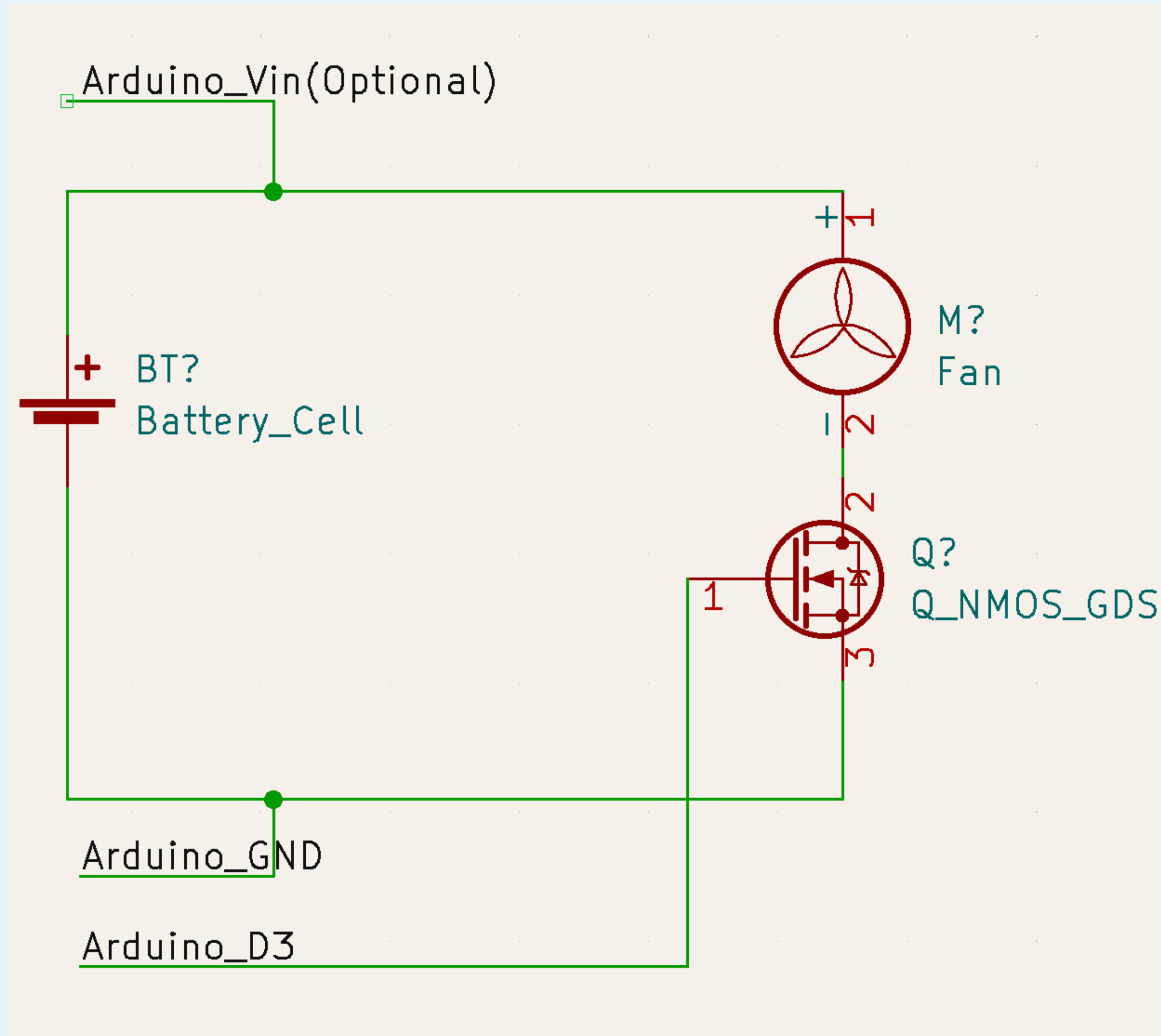


# K2232 MOSFET



<https://akizukidenshi.com/catalog/g/gI-02414/>

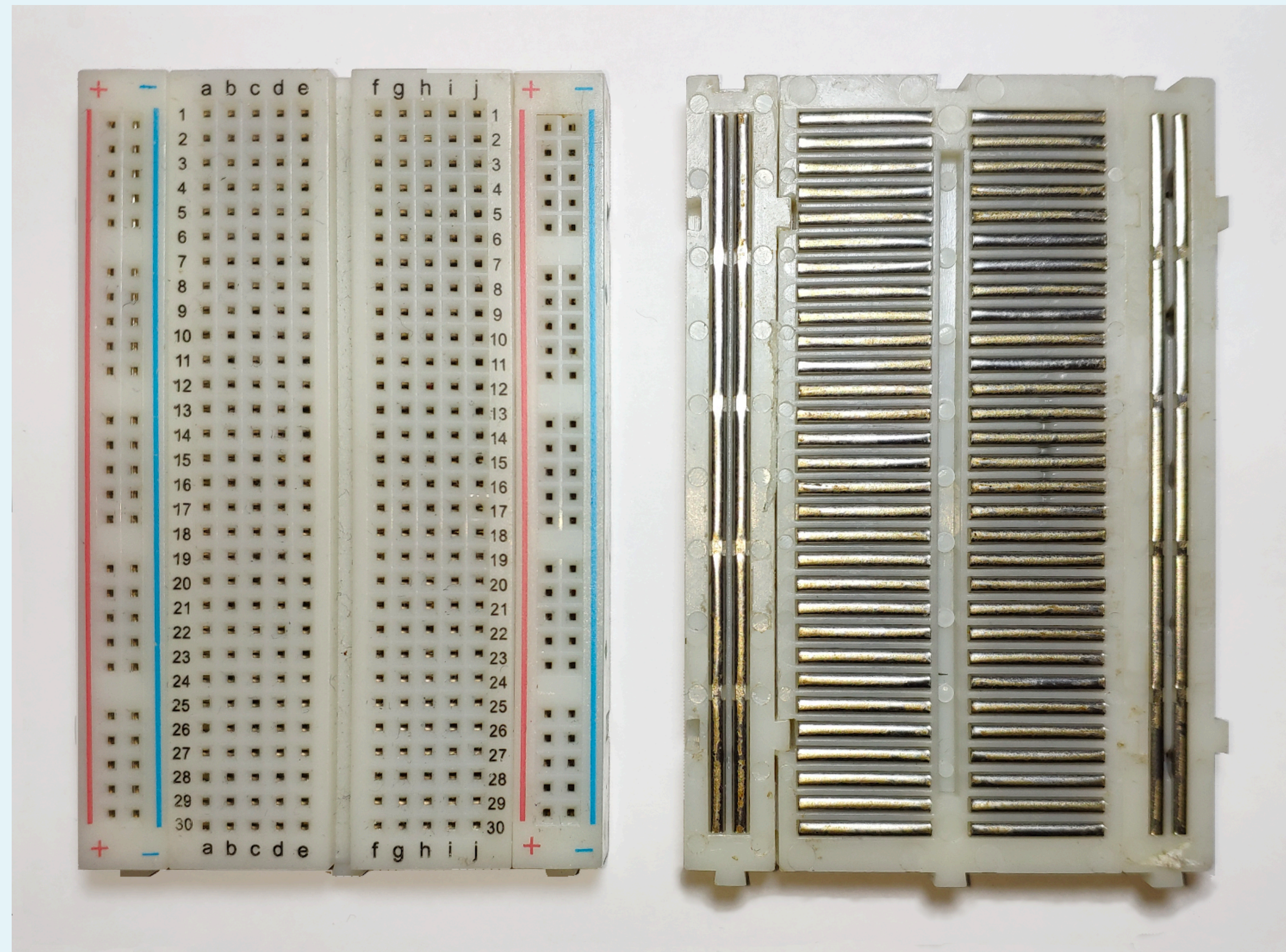
トランジスタとほぼ同じだが、ゲートからソースには電流が流れない



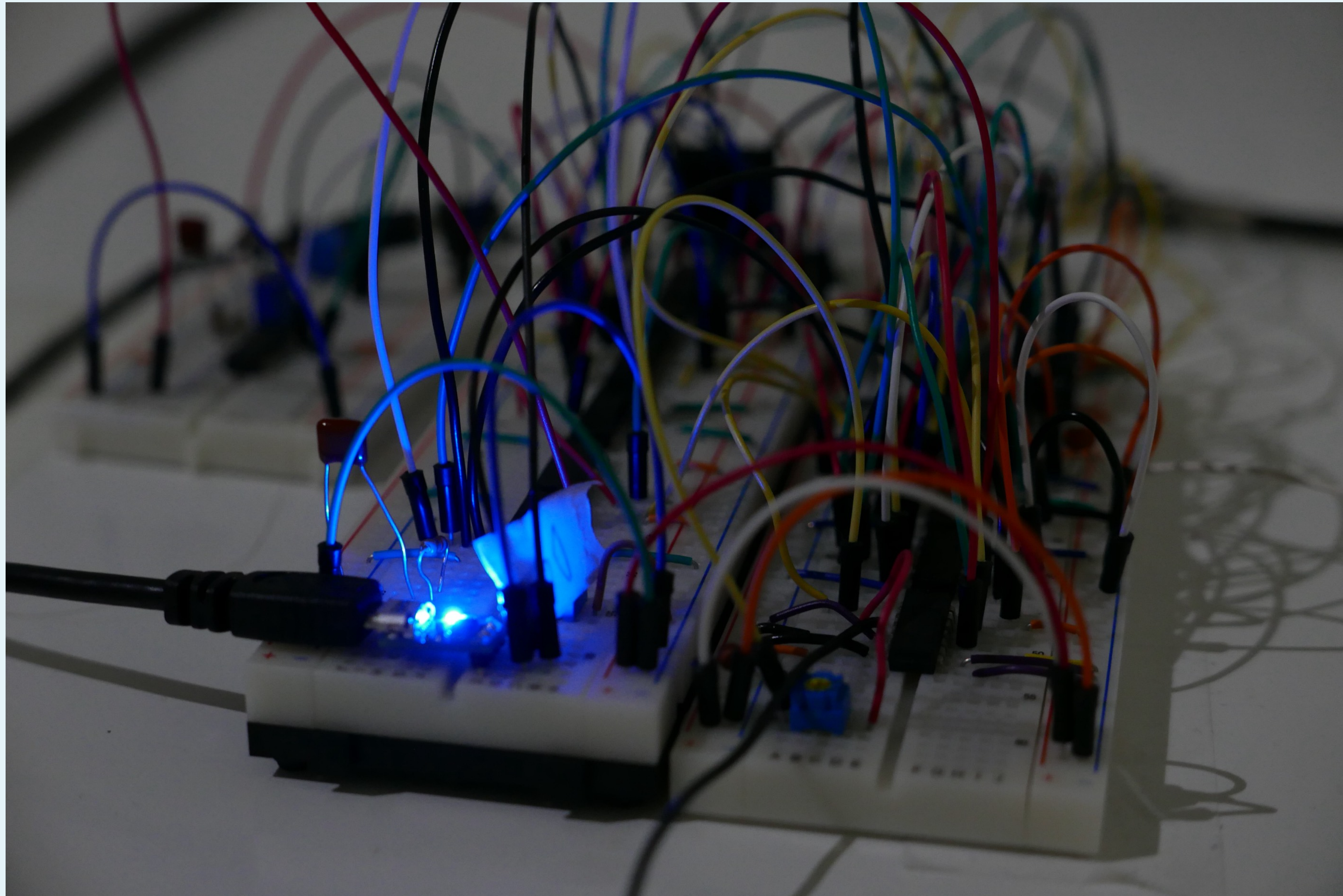
※ファンから逆流する電流について考慮していないので不適切です、今度補足



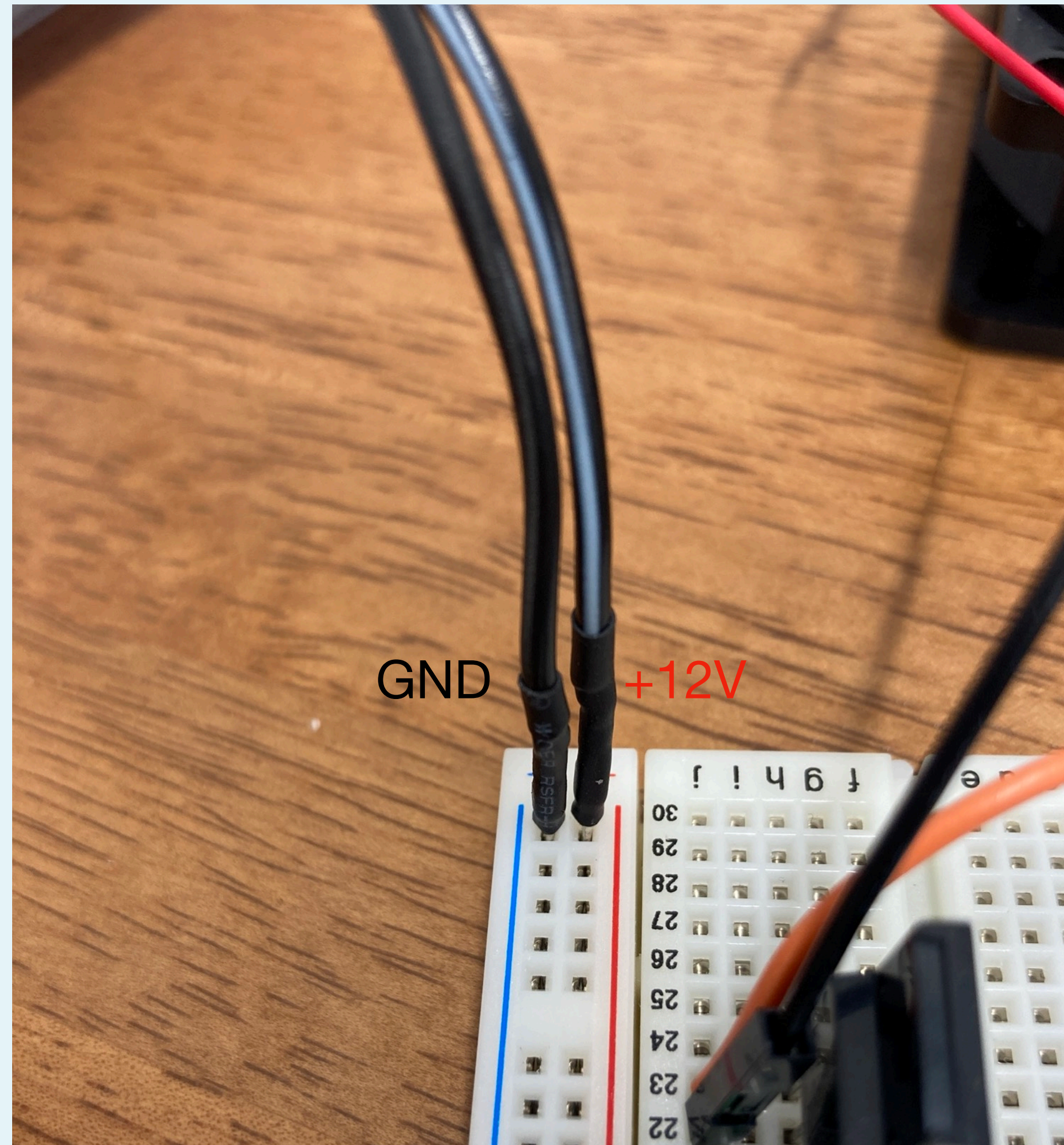
# ブレッドボード



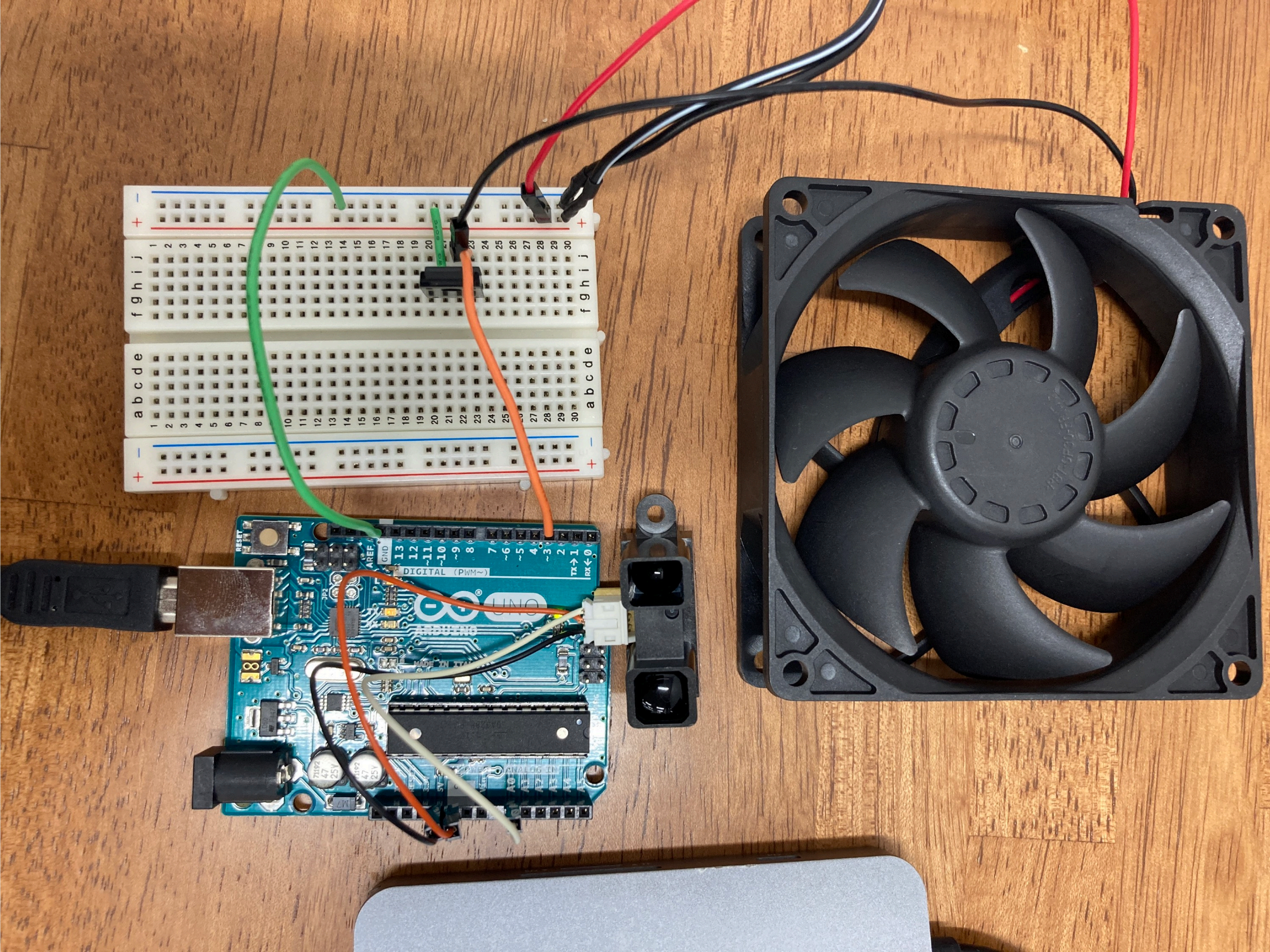
- はんだ付けなしで簡単に回路を組みたい人用



<https://matsuuratomoya.com/works/adm/>



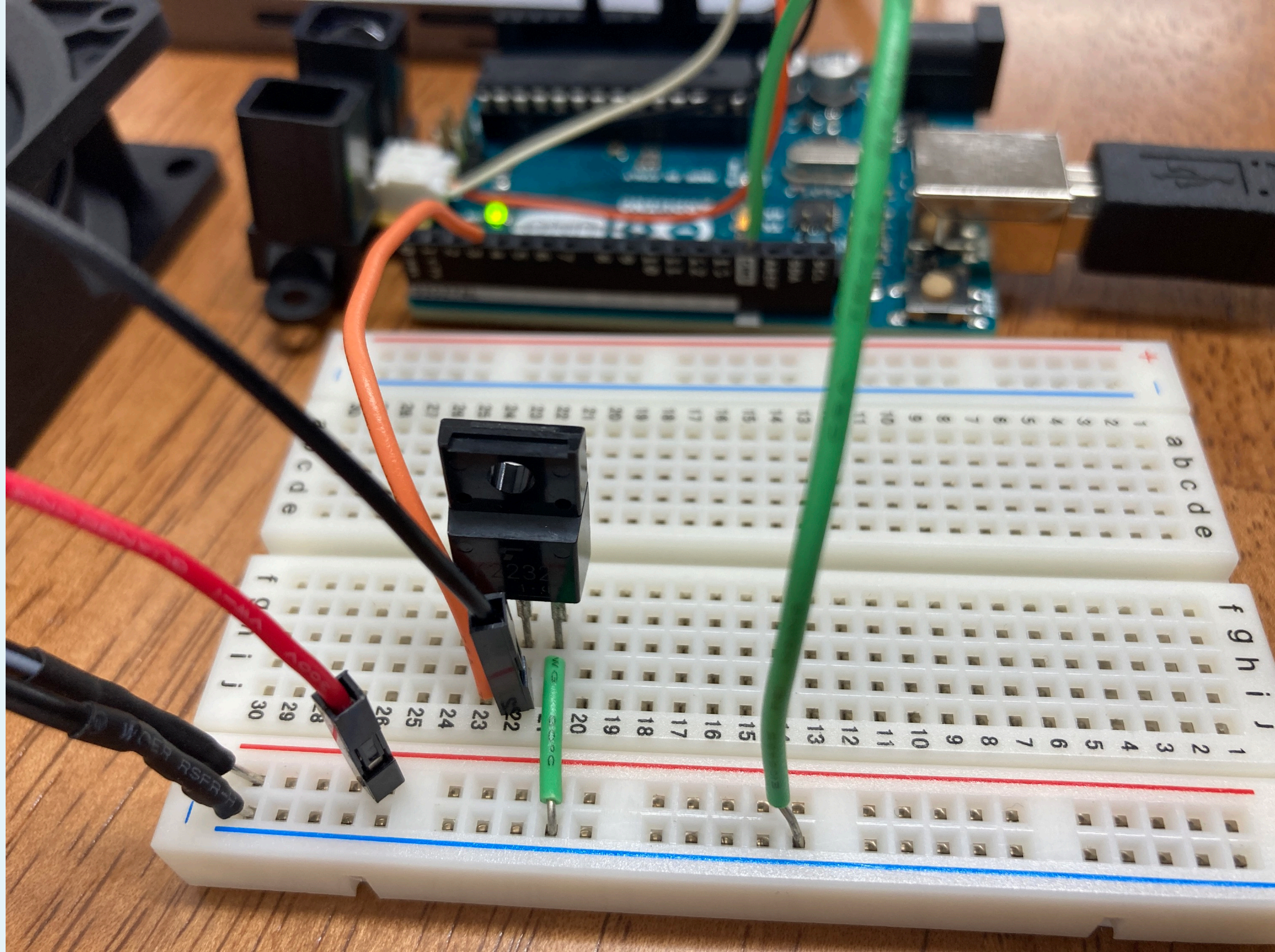
今回は、白い線が入ってる方が+  
(場合によりけりなので、きちんとテスターで確かめること)



breadboard pin headers: +, -, abcde, fghij, 1-30, fghij, -, +

Arduino Uno board labels: RESET, AREF, GND, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, TX, RX, ANALOG IN, 0-5, ARDUINO, GND, DIGITAL (PWM ~)



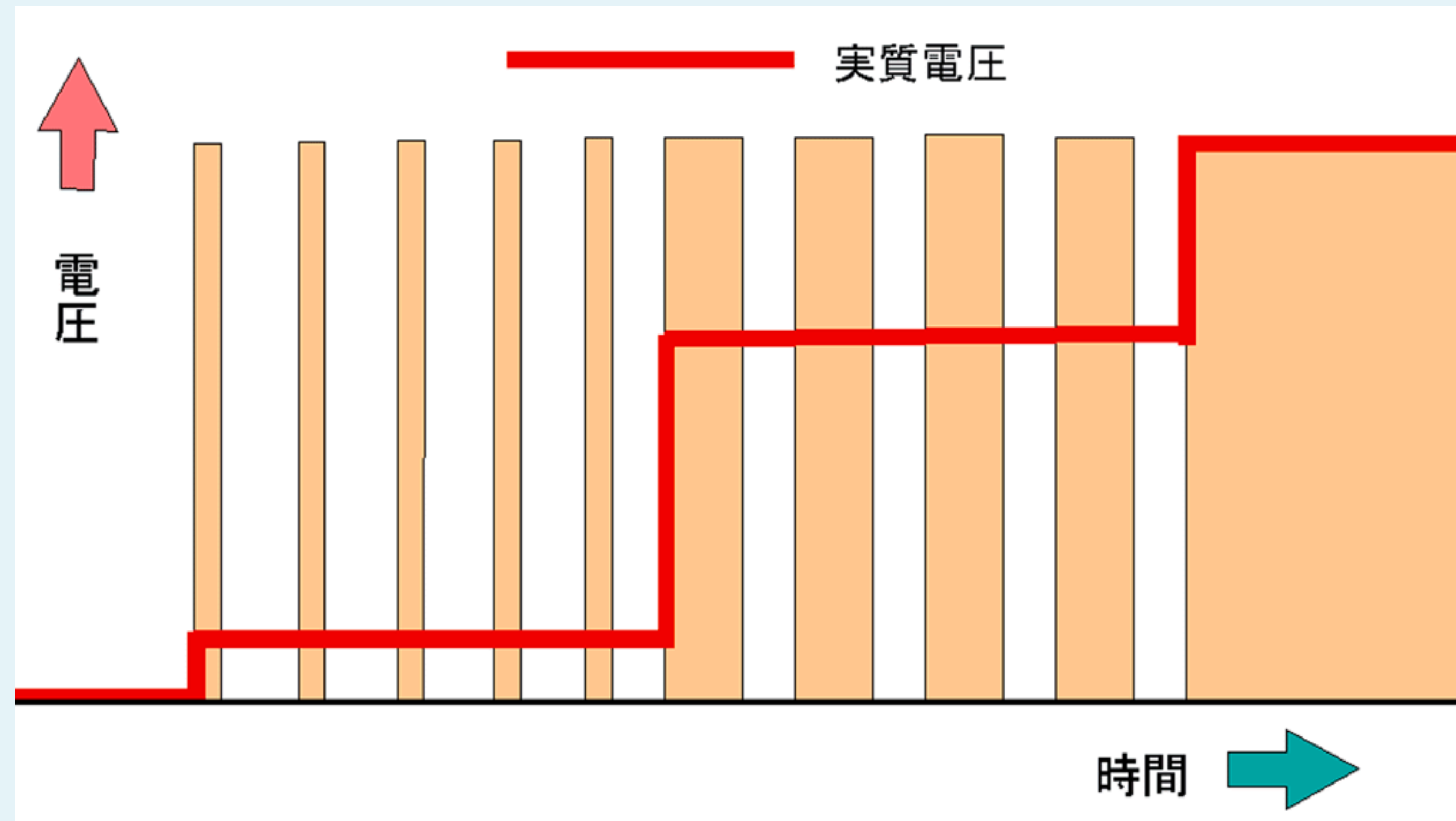


```
void setup() {
  pinMode(A0, INPUT);
  pinMode(3, OUTPUT);
  Serial.begin(9600);
}

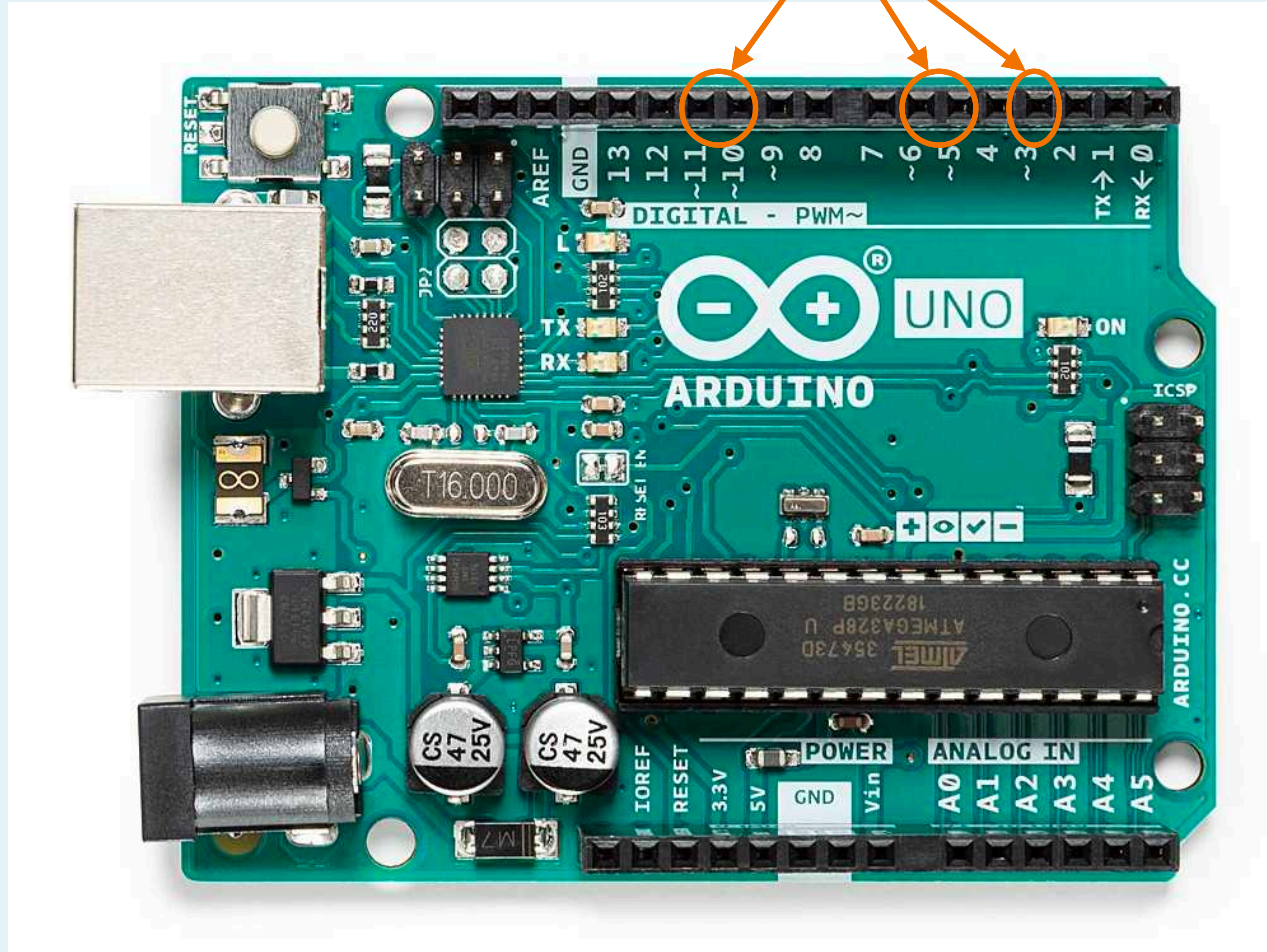
void loop() {
  int distance = analogRead(A0);
  Serial.println(distance);
  if(distance > 400) {
    digitalWrite(3, HIGH);
  } else {
    digitalWrite(3, LOW);
  }
  delay(20);
}
```

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)

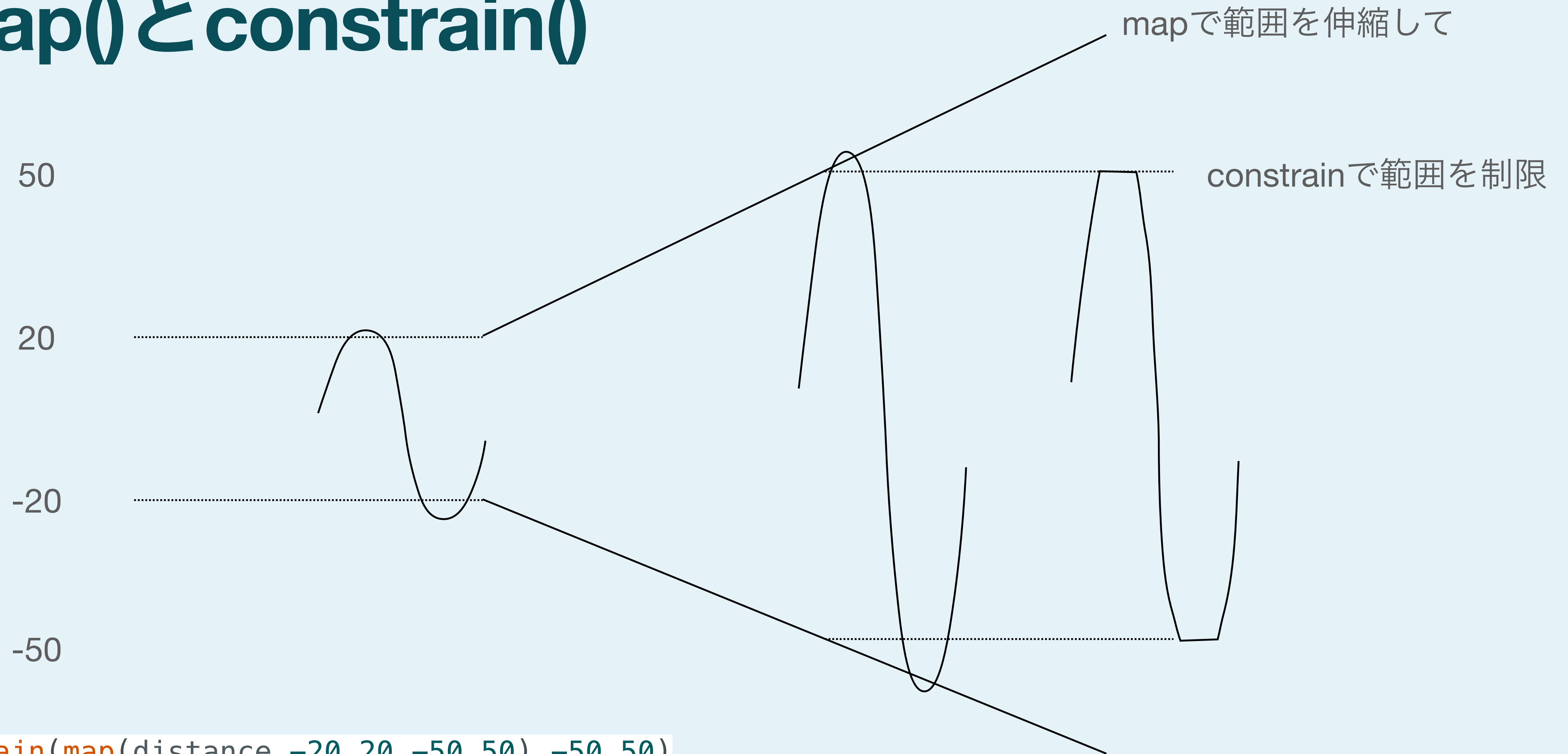




```
void setup() {  
  pinMode(A0, INPUT);  
  pinMode(3, OUTPUT);  
  Serial.begin(9600);  
}  
  
void loop() {  
  int distance = analogRead(A0);  
  int output = constrain(map(distance, 0, 500, 0, 255), 0, 255);  
  analogWrite(3, output);  
  Serial.print(distance);  
  Serial.print(',');  
  Serial.println(output);  
  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(', '); ← カンマ区切りでシリアルにprintすると、
  Serial.println(output); シリアルプロッタで複数の値を観測できる
  delay(40);
}
```

analogWriteを使って連続的に変化させる