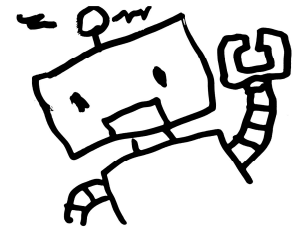


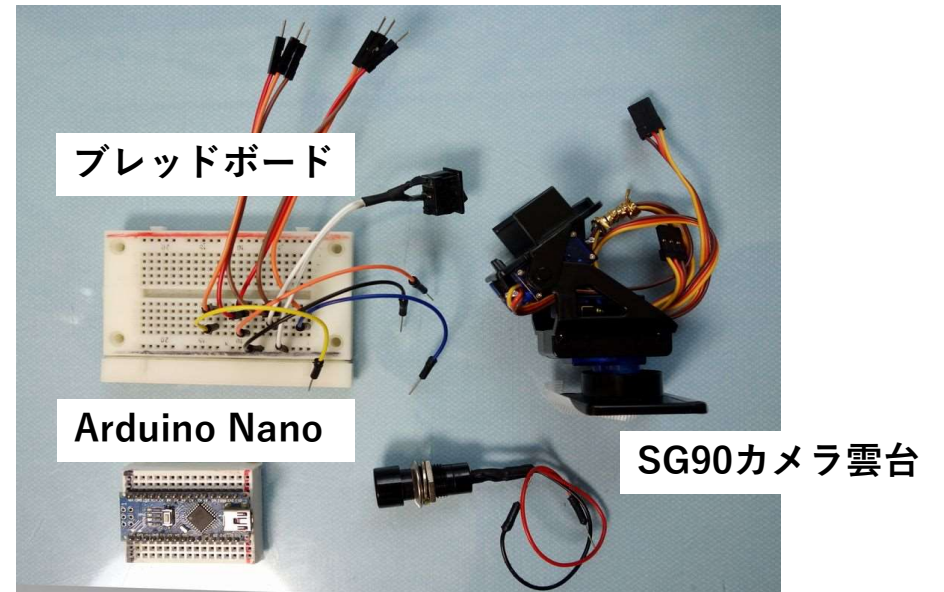
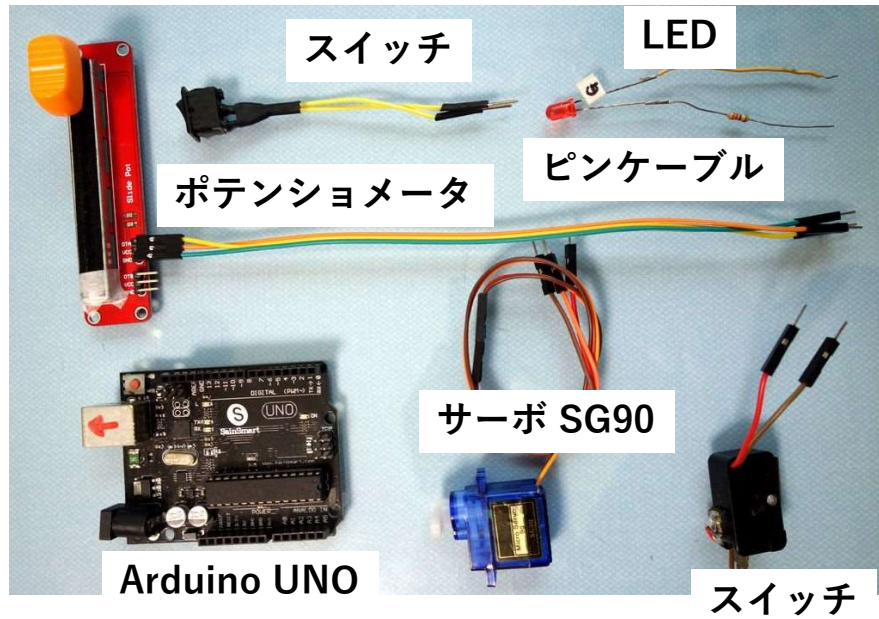
- Googleのパワポビューワでは正しく表示されません
- 必ずダウンロードしてください
- 解説が「ノート」に入っています

# Arduino入門

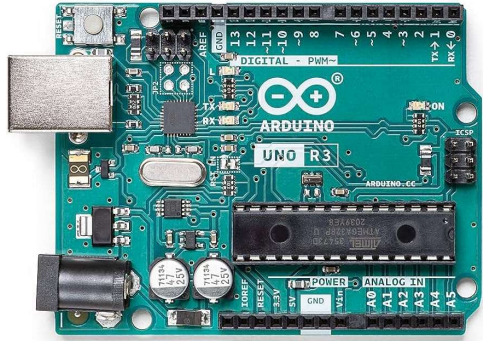


2023/9/15

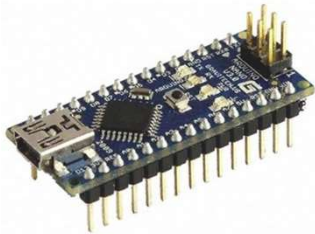
# お品書き



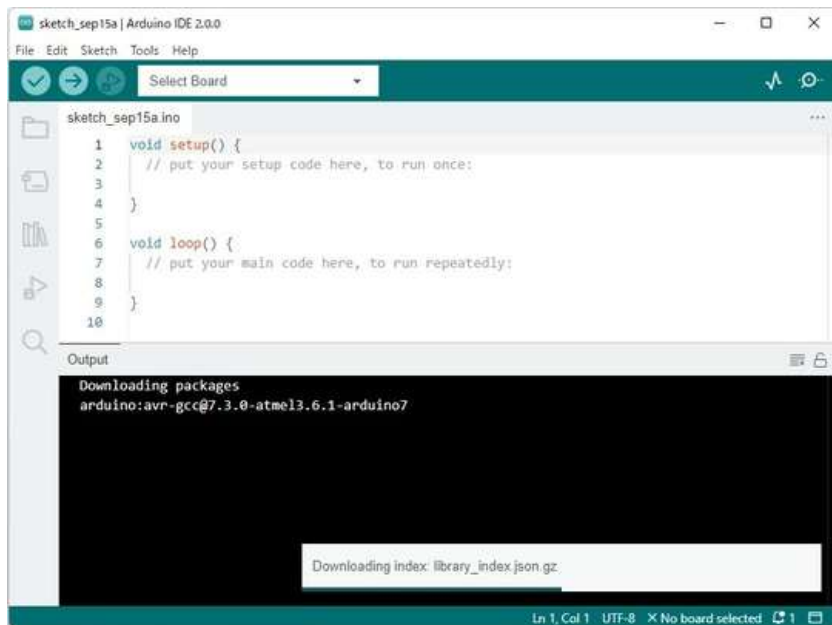
# Arduino (アルドゥイーノ)



- 2005年イタリアで誕生
- オープンソース
- Atmel AVR ARM 第4世代はルネサス
- MCU : Micro Controller Unit  
Risc-CPU, I/O, メモリ等を1チップに格納
- 互換品は500円以下～正規品は4000円弱
- ArduinoIDEでC/C++風言語で開発
- “シールド”を乗せることで拡張可能
- 学習用途やIOTなどで活用

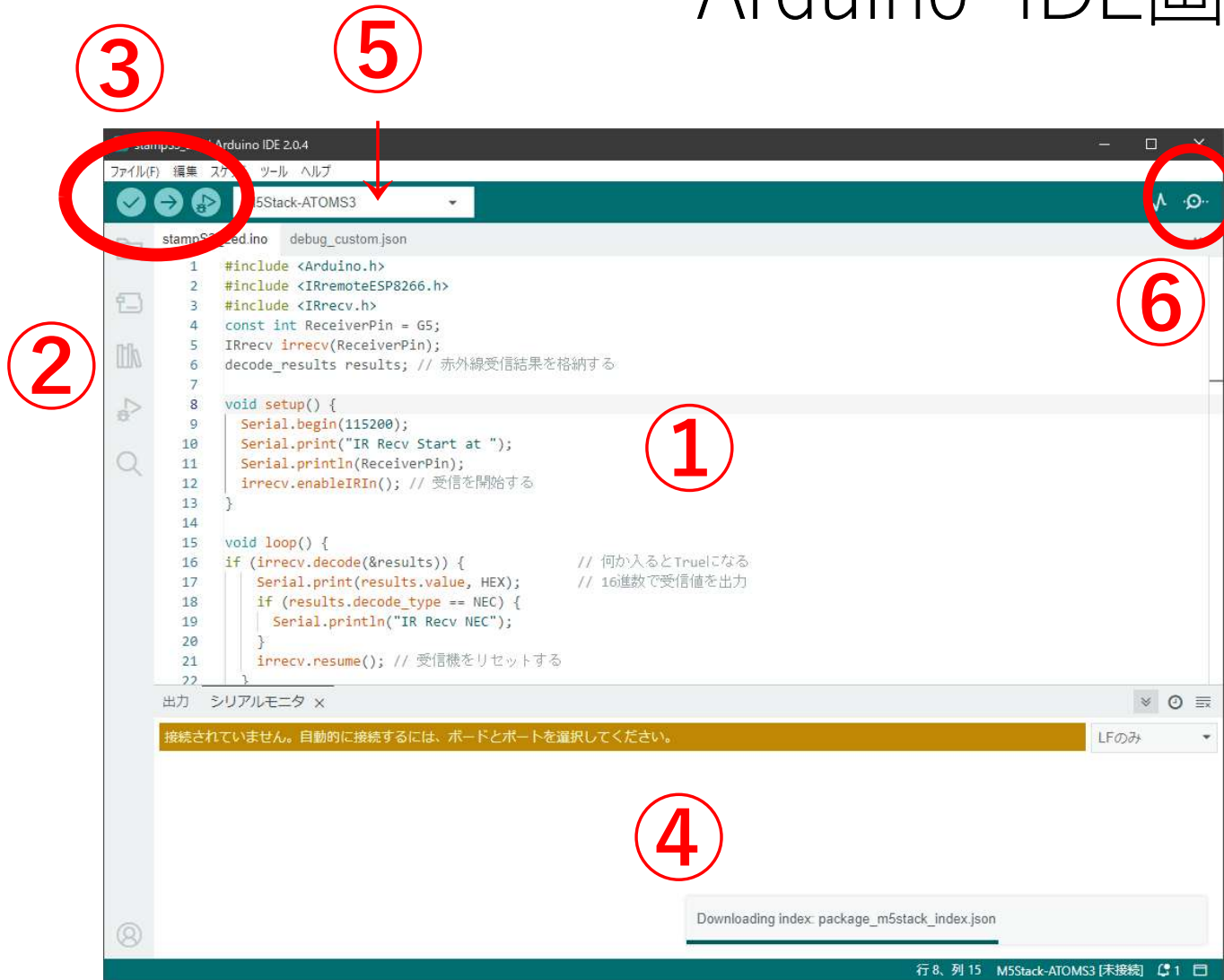


# 開発環境 (Arduino IDE Ver2)



- C/C++に準じた言語
- ネットワークにリンクした統合開発環境
- プログラム管理、ライブラリ管理
- Win,Mac,Linuxで使用可
- 日本語対応は「ファイルメニュー」「Preferences」から
- 「配色は」Darkモードが楽

# Arduino IDE画面



- ① コーディング
- ② ライブラリ
- ③ コンパイル・書き込み
- ④ メッセージ (エラー)
- ⑤ 機器・チャンネル選択
- ⑥ シリアル通信

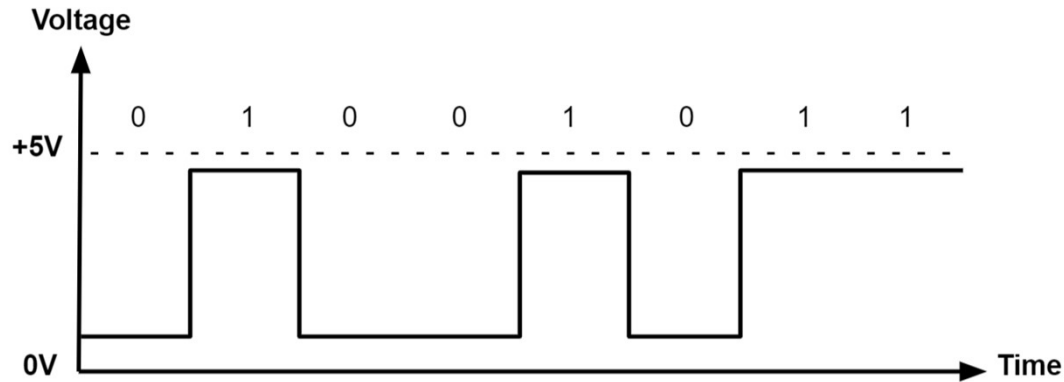
※ノート参照



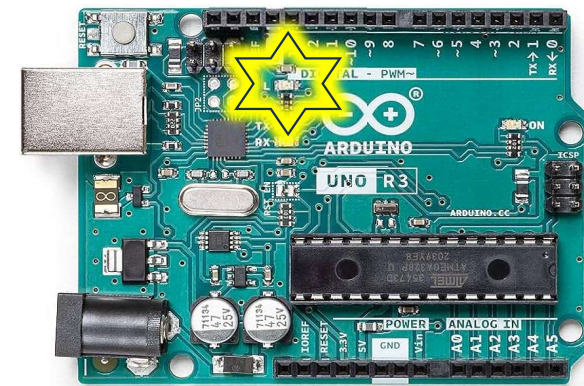
LED



# デジタル信号

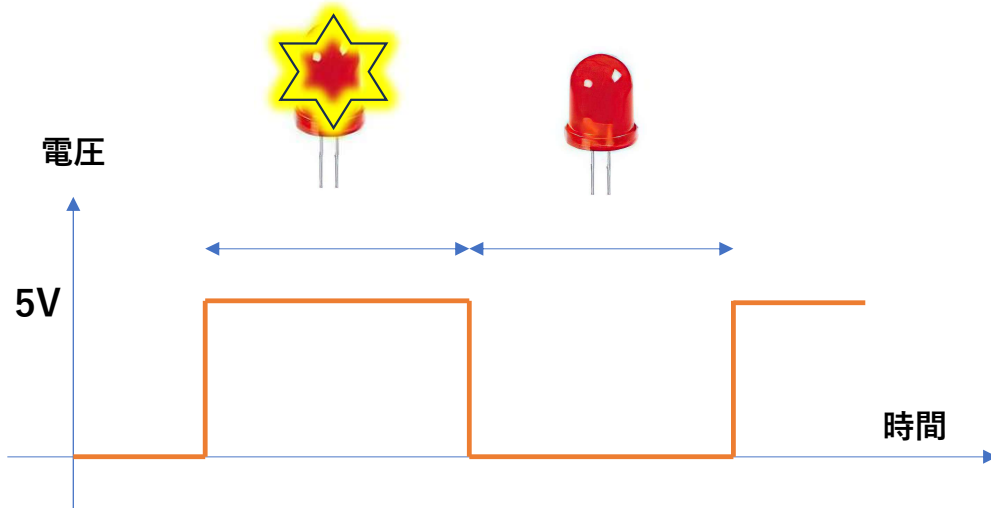


- Arduinoの場合、ピンは2種類  
信号Pin (GPIO) と電源Pin
- 信号は5Vの電圧、0Vの電圧の2種類で構成  
特定のpinを 5 V / 0V にセットすることでデータを表現  
どのぐらいの時間ONにするかプログラミングする。  
組み合わせで文字や数字を伝達することが可能
- PinにLEDが接続されていると発光します。





# Lチカ (LEDチカチカ)



基盤上のLEDは  
Pin13番に割当

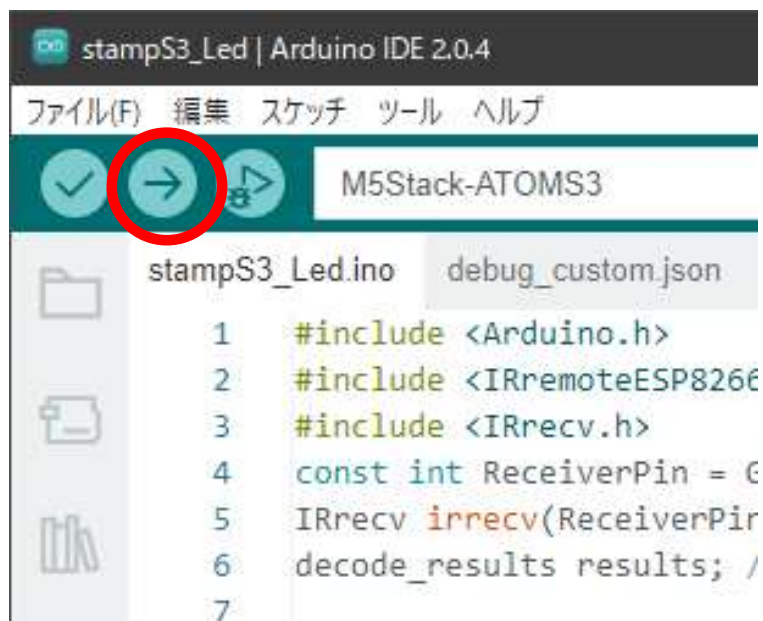
```
void setup() {  
  pinMode(13,OUTPUT);  
}  
void loop() {  
  digitalWrite(13,HIGH);  
  delay(500);  
  digitalWrite(13,LOW);  
  delay(500);  
}
```

※ノート参照

# プログラムの説明

<b>void setup() {</b>	プログラム開始時に一回だけ実行
<b>pinMode(13,OUTPUT);</b>	13番ピンを出力で使います宣言
<b>}</b>	}は関数の終了マーク
<b>void loop() {</b>	loopの中をずっと繰り返します
<b>digitalWrite(13,HIGH);</b>	13番ピンに5Vを印加
<b>delay(500);</b>	500ms実行を停止
<b>digitalWrite(13,LOW);</b>	13番ピンに0Vを印加
<b>delay(500);</b>	
<b>}</b>	loop関数の終わり

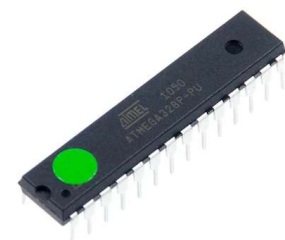
# プログラム転送！ & 実行



- 「CTRL+R」でコンパイル
- 「CTRL+U」でコンパイル&転送

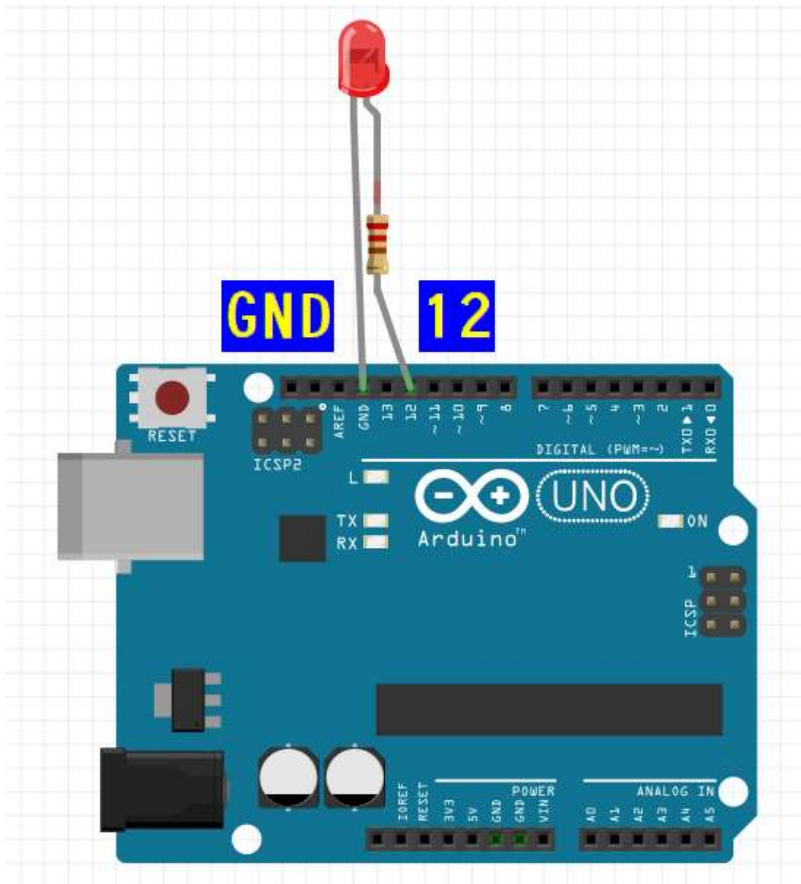
自動的にプログラムが起動します。  
Lチカが動き出します。

- PCと切り離しても動作します。
- ICチップ+電源だけでも稼働



ATMega328P-PU

# GPIOに外部LEDを付ける



```
// -----外付けLED 12番ピン-----  
// 先ほどのソースを12に書き直します
```

```
void setup() {  
  pinMode(12,OUTPUT);  
}  
void loop() {  
  digitalWrite(12,HIGH);  
  delay(500);  
  digitalWrite(12,LOW);  
  delay(500);  
}
```

※ノート参照

# エディタの使い方

- Shift + 矢印キーで範囲を選択（反転）
- Ctrl + Cでクリップボードへコピー
- 範囲選択を解除（カーソル移動）
- ペーストしたい場所へカーソルを移動
- Ctrl + Vで貼り付け
- Ctrl + Xで切り取り
- Ctrl + ZでUndo



```
01 Lチカ.txt - メモ帳
ファイル(E) 編集(E) 書式(O) 表示(V) ヘルプ(H)
nano: atmega328p
12pin は左端、GNDは右2番目

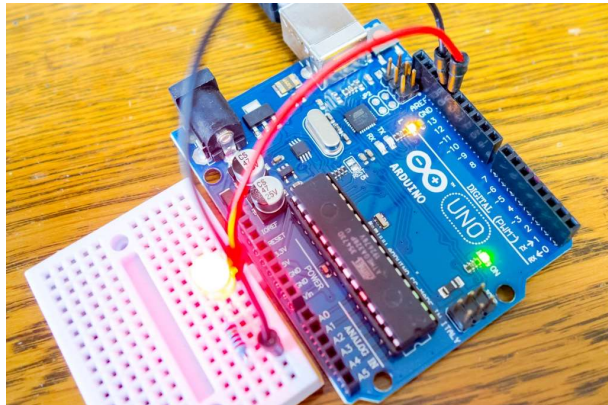
// ----- Lチカ 基盤上13番LED -----
void setup() {
  pinMode(13,OUTPUT);
}
void loop() {
  digitalWrite(13,HIGH);
  delay(500);
}
```

※Notepadで練習

# 12/13PinのLEDを交互に光らせる

```
void setup() {  
  pinMode(12,OUTPUT);  
  pinMode(13,OUTPUT);  
}
```

コピペ



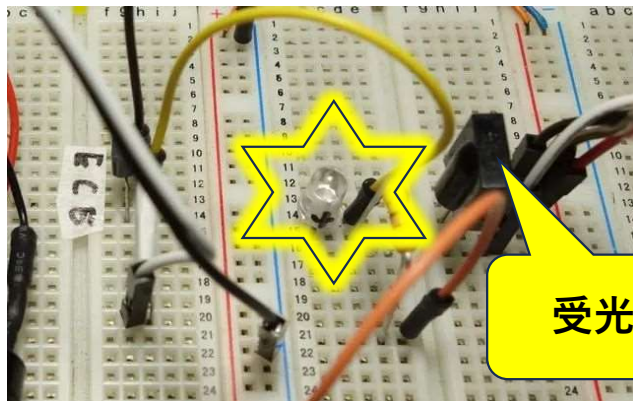
```
void loop() {  
  digitalWrite(12,HIGH);  
  digitalWrite(13,LOW);  
  delay(500);  
  
  digitalWrite(12,LOW);  
  digitalWrite(13,HIGH);  
  delay(500);  
}
```

HIGH / LOW  
交互に

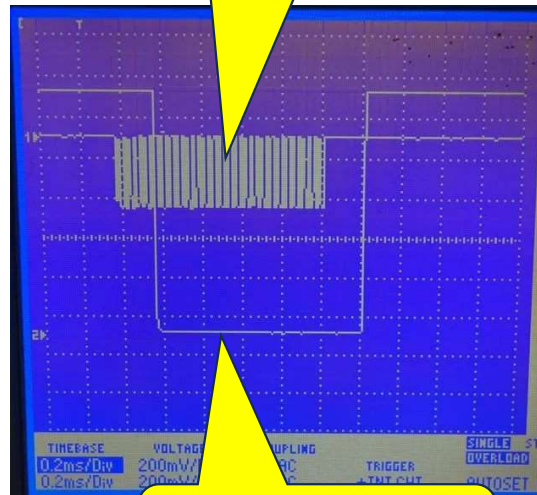
Shift + → / ↓  
Ctrl + C  
Ctrl + V

HIGH / LOW  
反対に

# Lチカ応用・赤外線リモコン



赤外LEDが発光したパルス



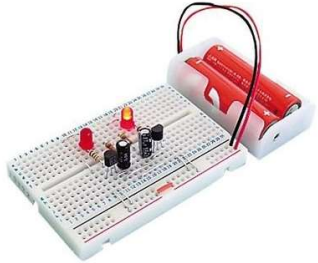
受光部から出力されたデータ

```
出力 シリアルモニタ x
メッセージ ('COM6'のArduino Unoにメッセー
リモコンのキーを押して下さい！
Mid キーON
IRコード: 41B659A6, ビット: 32
UP キーON
IRコード: 41B6F906, ビット: 32
Left キーON
IRコード: 41B619E6, ビット: 32
Down キーON
IRコード: 41B67986, ビット: 32
Right キーON
IRコード: 41B66996, ビット: 32
```



# 趣味で使う周辺機器

ブレッドボード



ピンケーブル



ギアードモータ



SG90 サーボモーター



LCD+キーボード



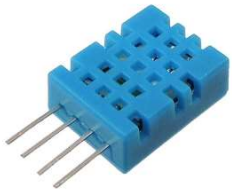
OLED



フルカラーLED



温湿度センサー



超音波測距センサー



人感センサー



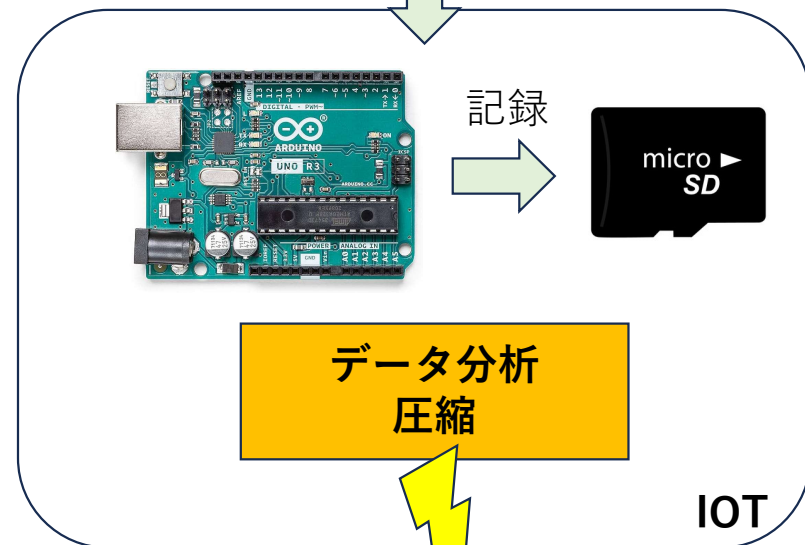
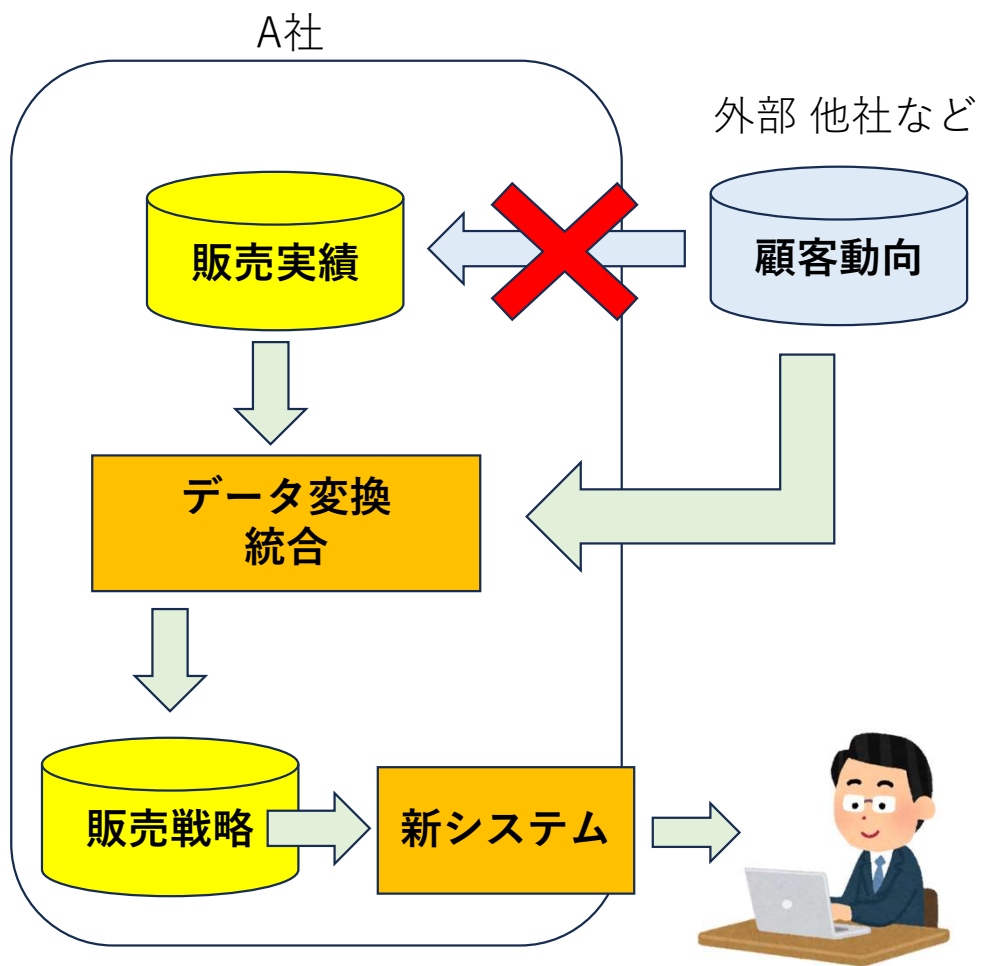
ELEGOO キット



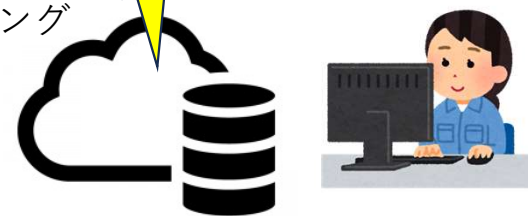
※ノート参照



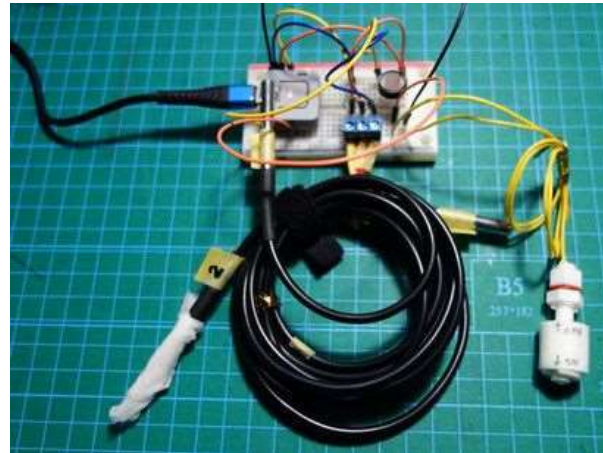
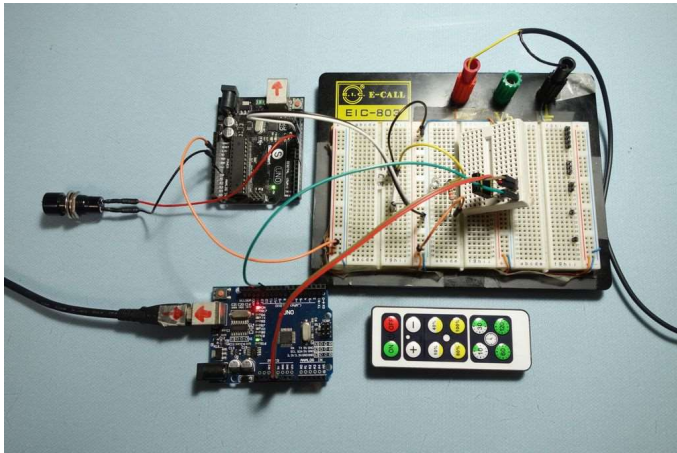
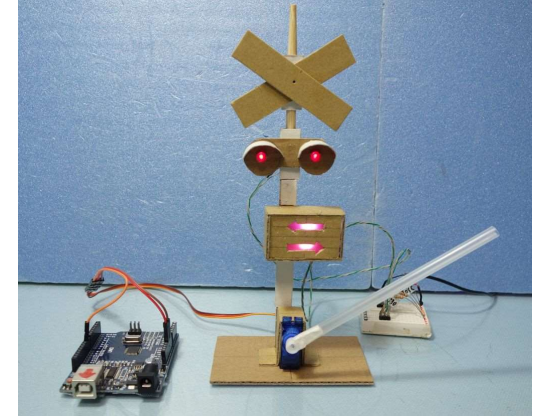
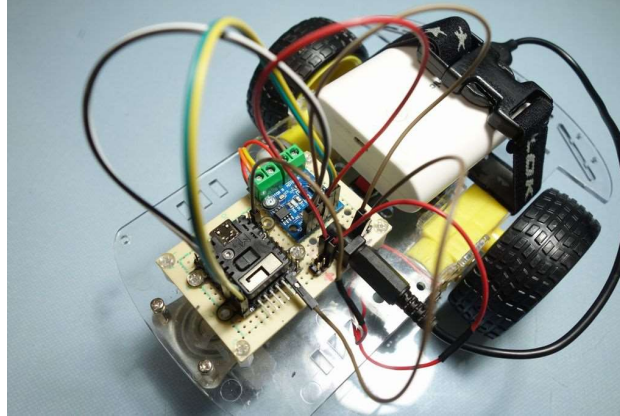
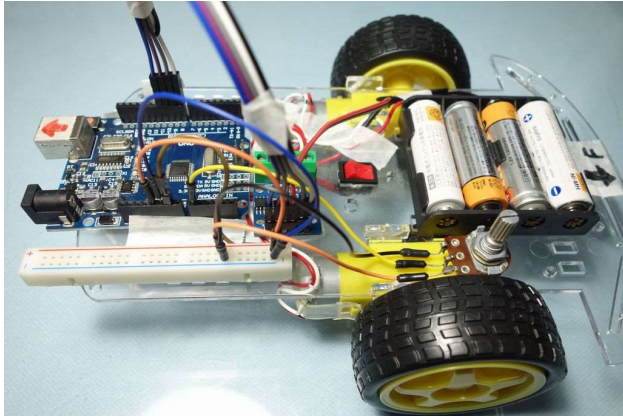
# プログラムが作れると何が出来る？



- エッジコンピューティング
- 通信費の軽減



# Arduino应用工作



# ChatGPTの活用



- 丁寧な指示書を書けばちゃんと動くが・・・
- 例えば煩雑な条件分けで処理を変えるなどは細部で嘘（ごまかし）が入る。
- 小さなルーチンを個別に作らせ、人間が組み合わせる。そういう使い方なら実用に耐える
- 人間側でしっかり検証が必要
- 先生として基本的なことを教えてもらうには適する。

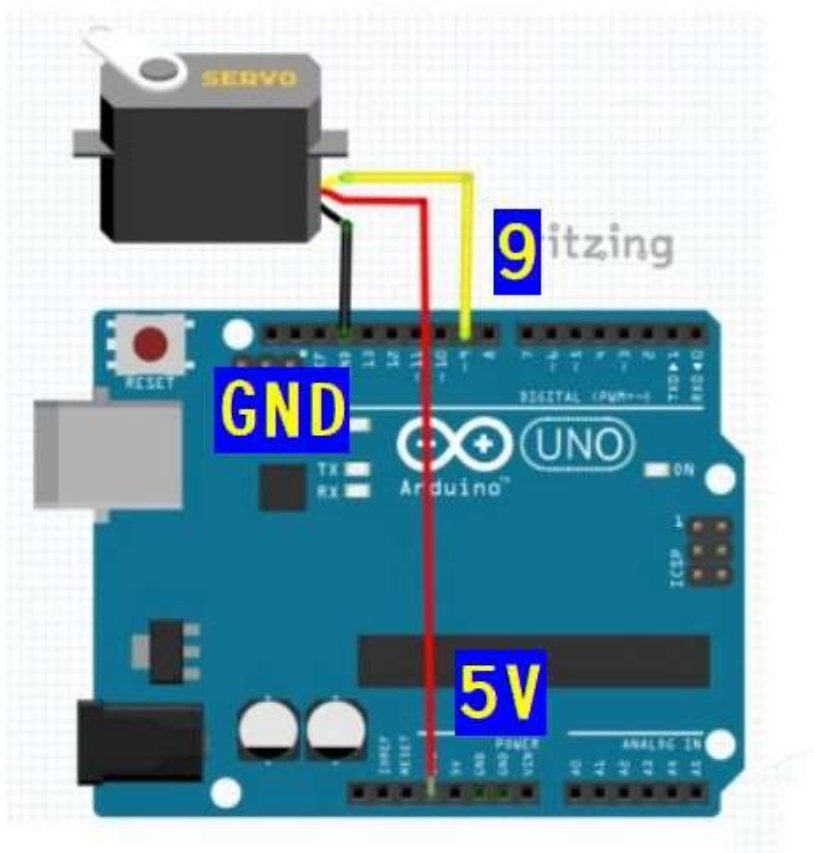


# サーボモータ





サーボモータを繋ぐ



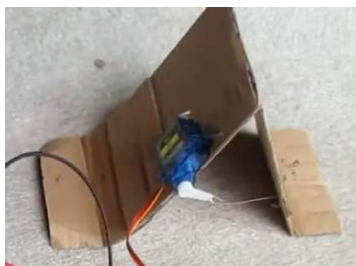
黄色 : 9 赤 : 5V 茶 : GND

```
#include <Servo.h>
Servo myServo;
```

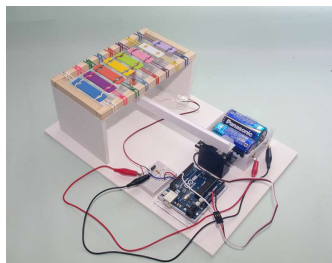
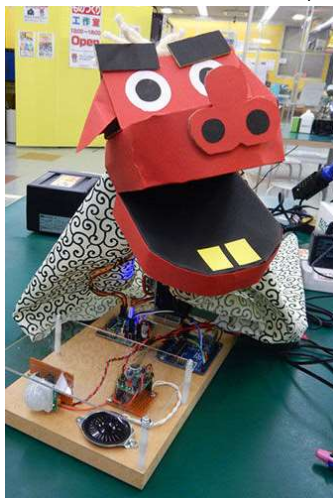
```
void setup() {
  myServo.attach(9);
}
```

```
void loop() {
  myServo.write(0);
  delay(1000);
  myServo.write(90);
  delay(1000);
}
```

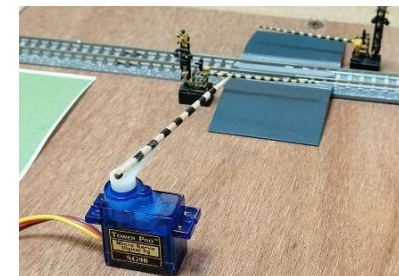
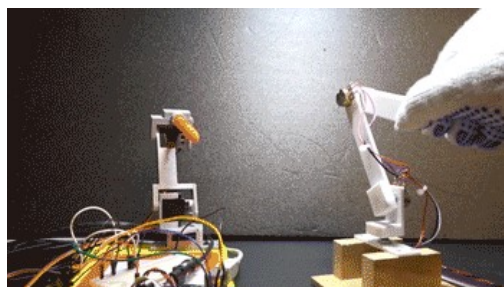
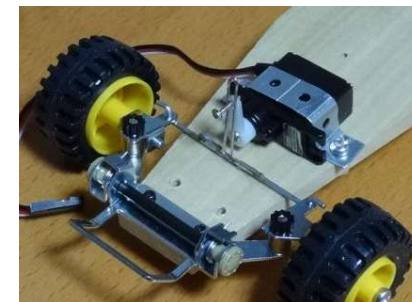
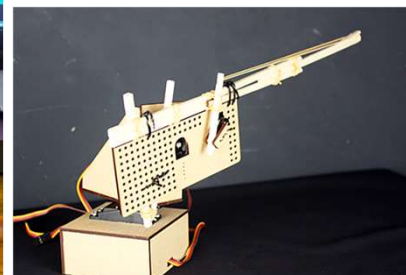
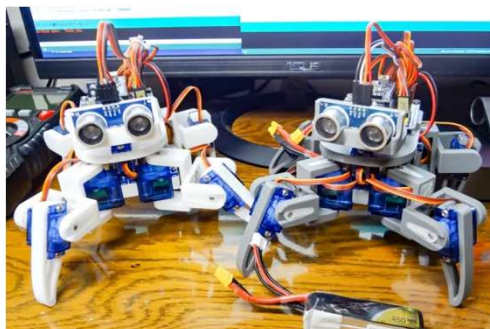
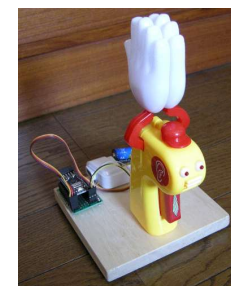
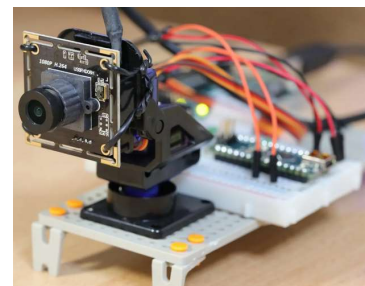
※ノート参照



「イケメン獅子舞君」 <https://www.youtube.com/watch?v=iOJ2VA-u-oU>



※ノート参照





# キットはお買い得



mercari

<p>¥1,400</p> <p>【新品】 Arduino ATtiny85 ゲーム 電子工...</p>	<p>¥1,300</p> <p>【新品】 Arduino D1 Mini 電子工作 プログラミン...</p>	<p>¥1,400</p> <p>【新品】 Arduino ATtiny85 ゲーム 電子工...</p>	<p>¥2,000</p> <p>新品 : Arduino ESP-WROOM-32D 電子工...</p>	<p>¥1,700</p> <p>【新品】 Raspberry Pi Pico 土壌水分センサ...</p>
<p>¥1,800</p> <p>【新品】 Arduino OLED MP3 電子工作 プログ...</p>	<p>¥1,400</p> <p>【新品】 Arduino ATtiny85 ゲーム 電子工...</p>	<p>¥2,800</p> <p>【新品】 ESP32 3個 Arduino 電子工作 プロ...</p>	<p>¥2,300</p> <p>【新品】 Raspberry Pi Pico W 電子工作 プロ...</p>	<p>¥1,800</p> <p>【新品】 Arduino OLED MP3 電子工作 プログ...</p>
<p>¥1,200</p>	<p>¥1,300</p>	<p>¥2,300</p>	<p>¥3,000</p>	<p>¥2,800</p>

おまけ





# C言語

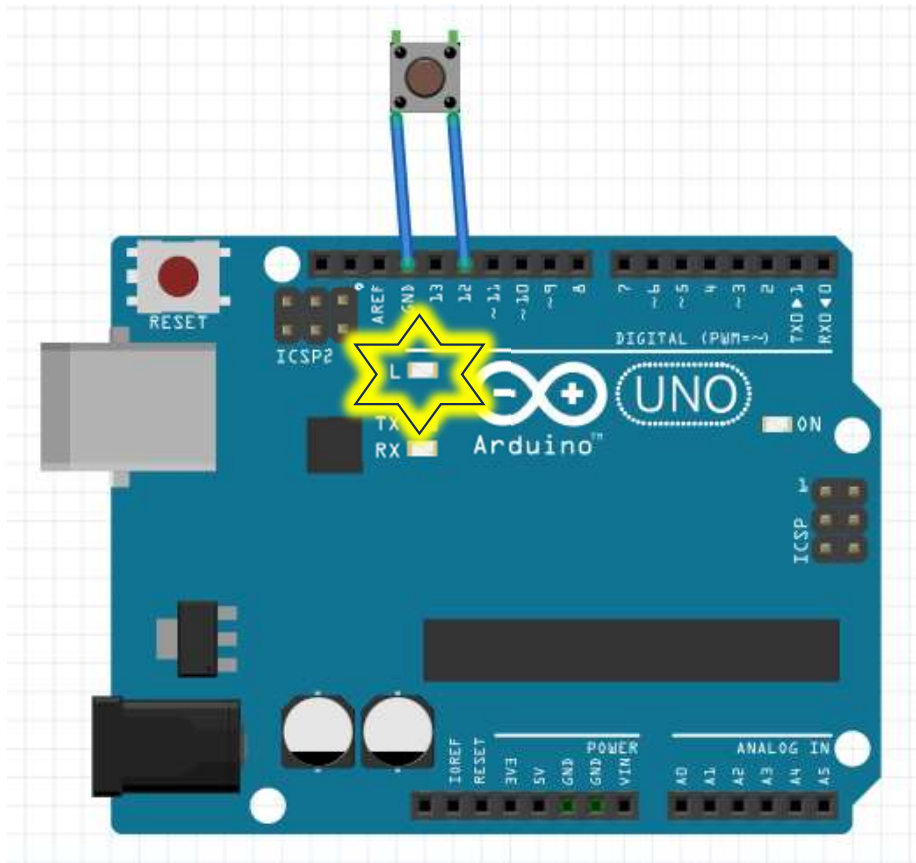
- とほほのC言語入門  
<https://www.tohoho-web.com/ex/c-lang.html>
  - C言語（等）の実習  
<https://paiza.io/ja/projects/new>
- 覚える基本構文は10個程度
  - 組み込み系では文字操作は殆どない
  - C++追加学習は僅かでOK

A screenshot of the Paiza.io online IDE. The interface shows a code editor with a C program. The code is as follows:

```
1 #include <stdio.h>
2 main(){
3     printf("hello world\n");
4 }
5
```

The IDE has a green header bar with the Paiza.io logo and a search bar. Below the code editor, there is a green button labeled "実行 (Ctrl-Enter)" and a link "Cを学ぶ | プログラミングカ診断". At the bottom, there is an output area showing "hello world" and a "コンパイルエラー" section with "0" comments. The execution time is shown as "(0.00 sec)".

# スイッチ押下で内蔵LED点灯・消灯



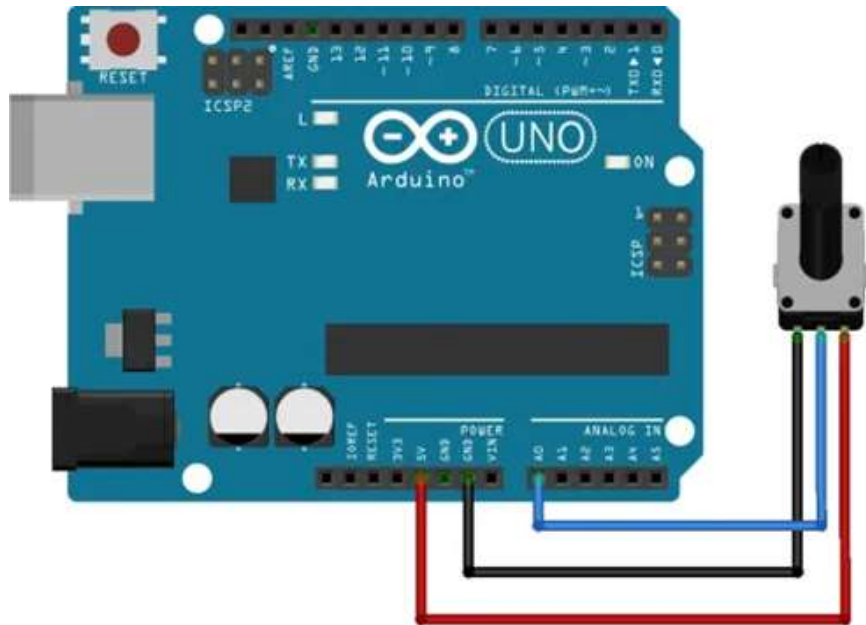
12番PinとGNDに接続

```
void setup() {  
  pinMode(12,INPUT_PULLUP);  
  pinMode(13,OUTPUT);  
}
```

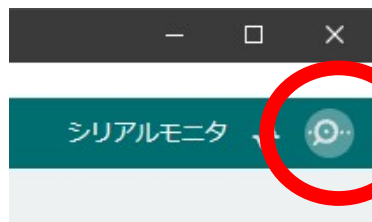
```
void loop() {  
  if (digitalRead(12) == LOW){  
    digitalWrite(13,HIGH);  
  }else{  
    digitalWrite(13,LOW);  
  }  
}
```

※ノート参照

# ポテンシヨメータで値を取得



黄 : A0   オレンジ : 5V   緑 : GND



シリアル通信の表示

```
void setup() {  
  pinMode(A0,INPUT);  
  pinMode(13,OUTPUT);  
}
```

```
  Serial.begin(9600);  
  Serial.println("hello");  
}
```

```
void loop() {  
  int val = analogRead(A0);  
  Serial.println(val);  
  delay(1000);  
}
```

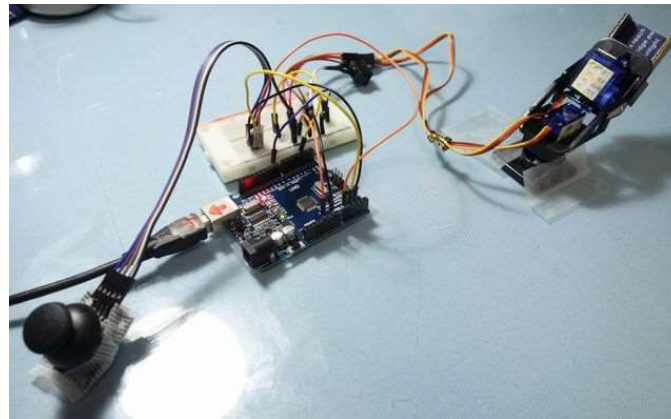
※ノート参照

## 2軸サーボ カメラ雲台をジョイスティックで動かす

```
#include "Servo.h"
Servo servo_rl;
Servo servo_ud;

int x,y,sw;
void getJoystick(){
  x = analogRead(A0);
  y = analogRead(A1);
  sw= analogRead(A2);
  if (sw != 0) sw = 1;
}
void dspJyoData(){
  Serial.print("Y:");
  Serial.print(y);
  Serial.print(" X:");
  Serial.print(x);
  Serial.print(" SW:");
  Serial.println(sw);
}
```

```
void setup() {
  servo_rl.attach(9);
  servo_ud.attach(3);
  pinMode(A0,INPUT);
  pinMode(A1,INPUT);
  pinMode(A2,INPUT);
  Serial.begin(9600);
  Serial.println("Sony JoyStick begin");
}
```



```
int s1a,s2a; // サーボ1,2の角度

void loop() {
  getJoystick();
  displayData();

  if (sw == 1 ){
    s1a = map(x, 0, 1023, 0,180);
    s2a = map(y, 0, 1023, 0,180);
    servo_rl.write(s1a);
    servo_ud.write(s2a);
  } else{
    servo_rl.write(0);
    servo_ud.write(0);
  }
  delay(300);
}
```

※ノート参照（補正プログラム有）

おわり



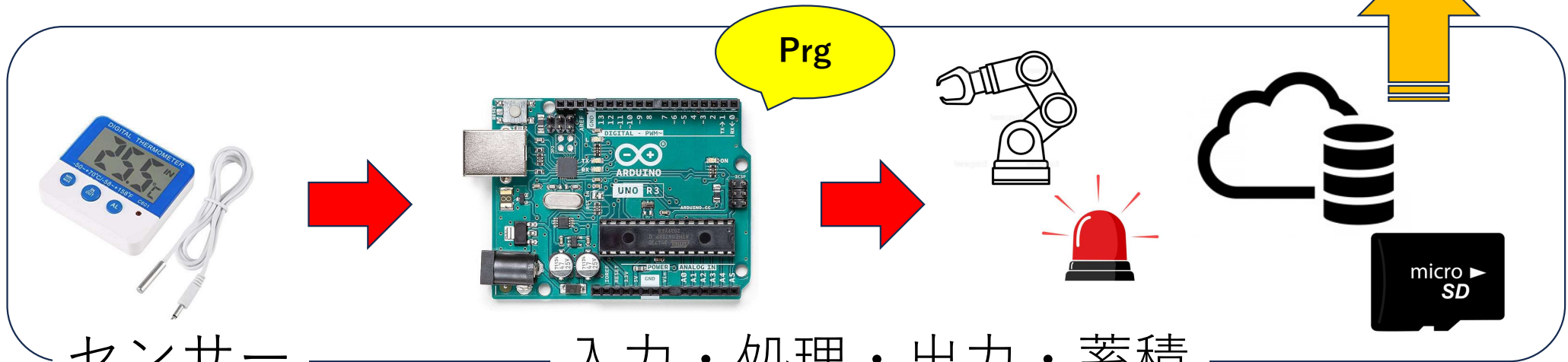
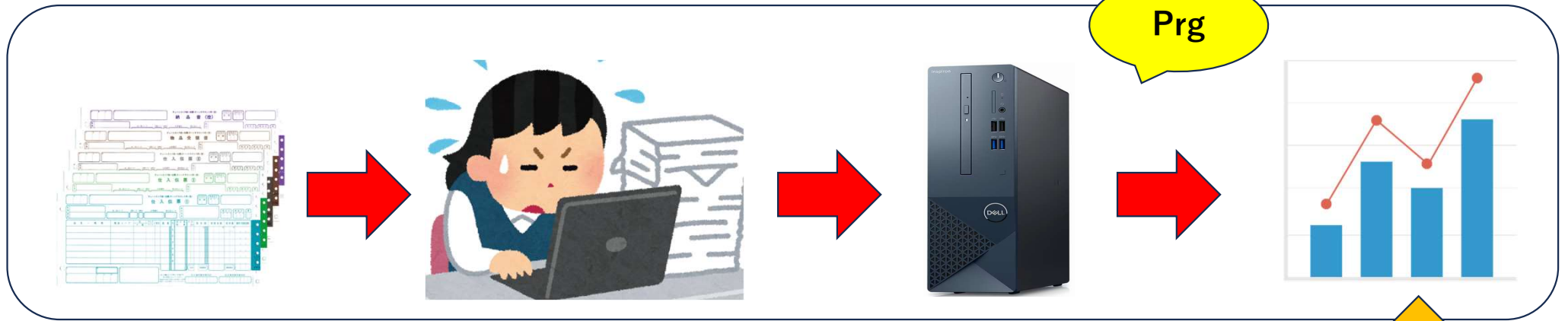
# マイコンが行う処理 (IOT・クラウド連携)

元データ

入力

処理

出力



センサー

入力・処理・出力・蓄積