

STEAM教育 出前授業

すごいで！マイコンプログラミング講座

IoTバギーカー

鹿児島高専 創造デザイン工学科 I 類

HP: <http://www2.kagoshima-ct.ac.jp/elect/>



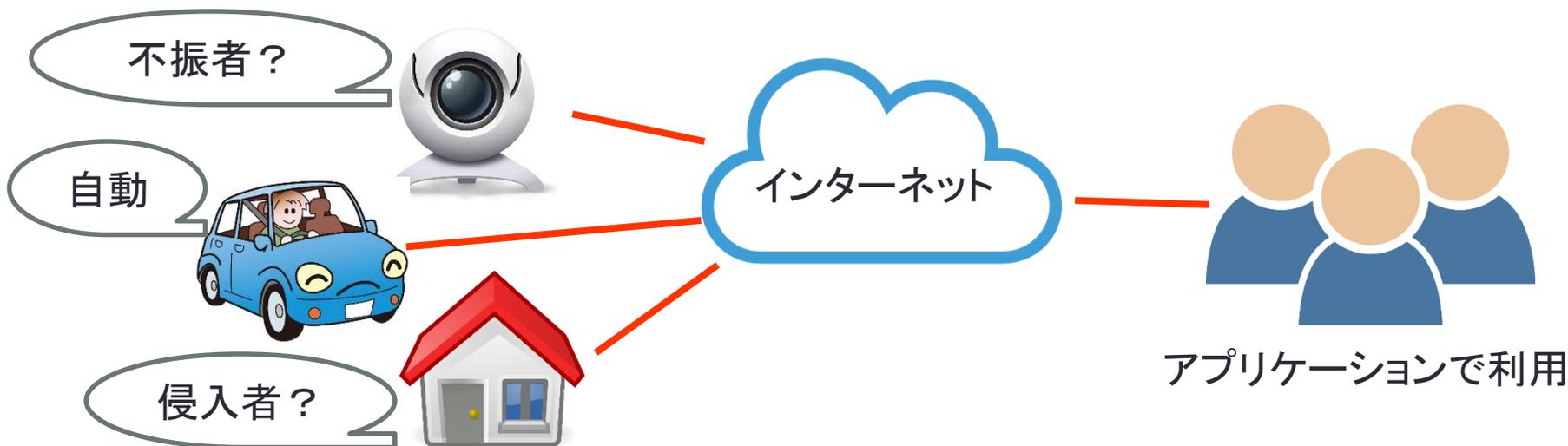
【本日の大まかな内容】

- ✓ IoT、スマホ操作バギーカーの概要(動作)、Webページ
- ✓ ブレッドボード、配線
- ✓ スマホでLEDを制御(Lチカ)
- ✓ スマホでDCモーター、DCサーボモーターを制御
- ✓ 本格的に動作チェック

創造デザイン工学科 I 類
2年生、4年生相当の
実習を体験！

1. IoTとは

- IoT(Internet of Things):モノのインターネット
センサー、カメラ、家など、身の周りの様々なモノがインターネットに接続し、どこでもそのデータを利用できること。



✓ 次世代

人のインターネット化 (IoH): 行動、生体情報、感情など
能力(Ability)のインターネット化 (IoA): 感覚、身体能力の拡張

マイコン

小さなコンピューター

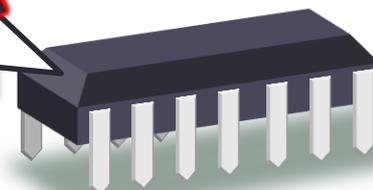
プログラムを書き込むことができる

電気回路につなぐと、いろいろな働きができる

```

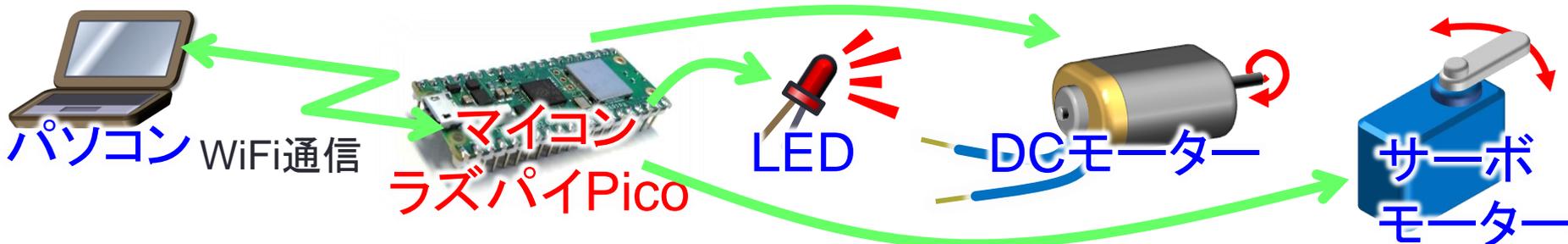
1 package taiken02;
2
3 import java.awt.BorderLayout;
4
5 public class ImageDropPanel {
6
7     //-----|オプドの部品宣言|-----
8     JPanel mainPanel; //メインパネル
9     JPanel buttonPanel; //ボタンパネル
10    ImageDropPanel imPanel1; //①
11    ImageDropPanel imPanel2;
12
13 }
  
```

プログラム



今回のマイコン(Raspberry Pi Pico W)の役割

- LEDを光らせる LEDにつながる線の電気をON/OFF
- WiFi通信 パソコン(スマホ)と無線通信
- DCモーターをコントロール 正転・逆転・停止を命令
- サーボモーターをコントロール 角度を決める

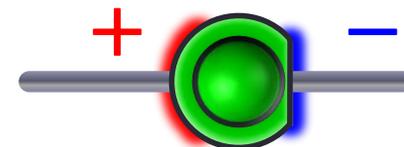
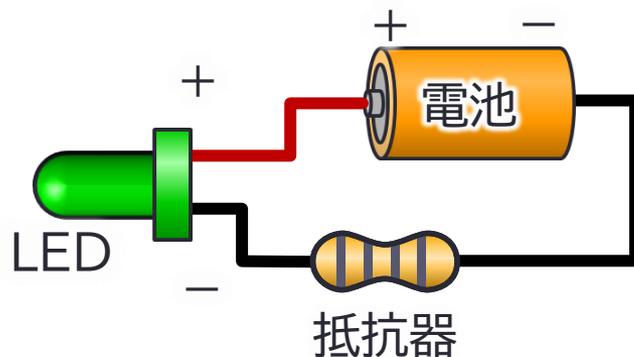
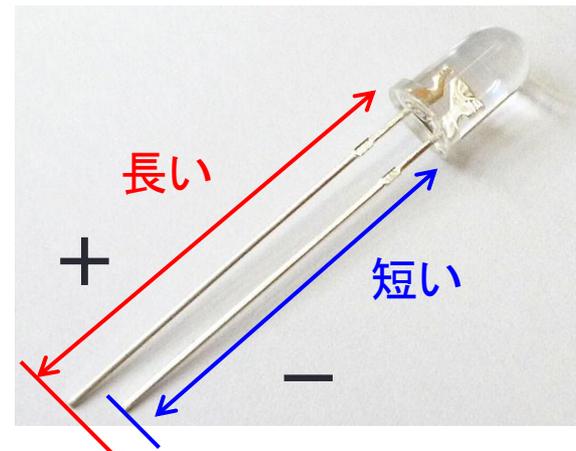


LED(発光ダイオード)

電気を流すと光ります。

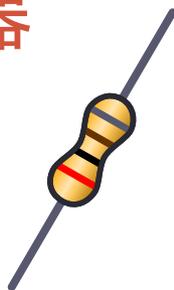
足の長い方が+で、短い方が-。

電池を逆向きにつなぐと光りません
(壊れるかも)。



真上から見た形が
丸い方が+，平たい方が-

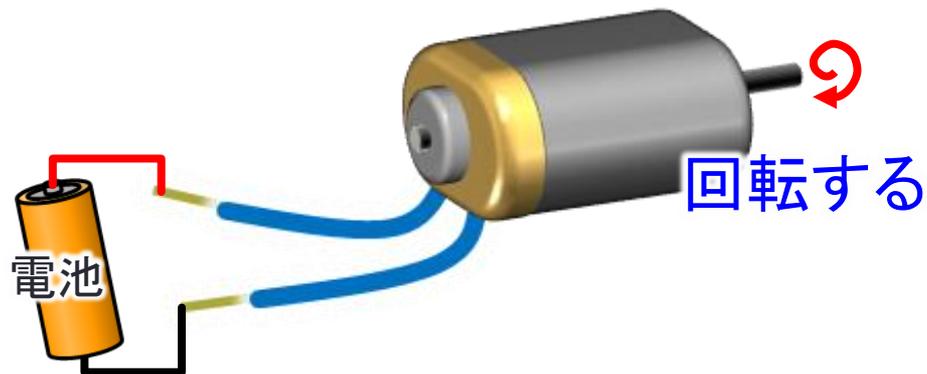
抵抗器



LEDに電気が流れすぎないようにします。
(LEDに大量の電気が流れると壊れる)
抵抗器に+-の向きはありません。

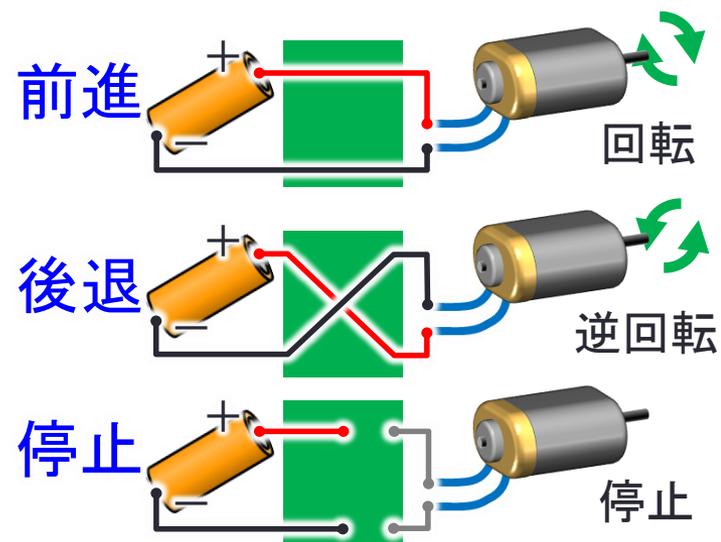
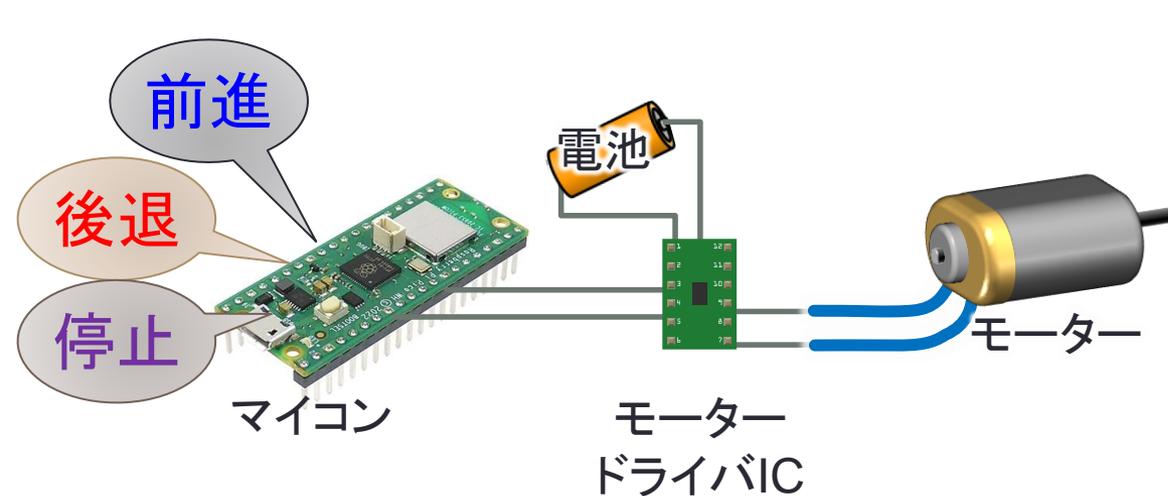
DCモーター

電気を流すと回ります。
電池を逆向きにつなぐと
逆回転します。



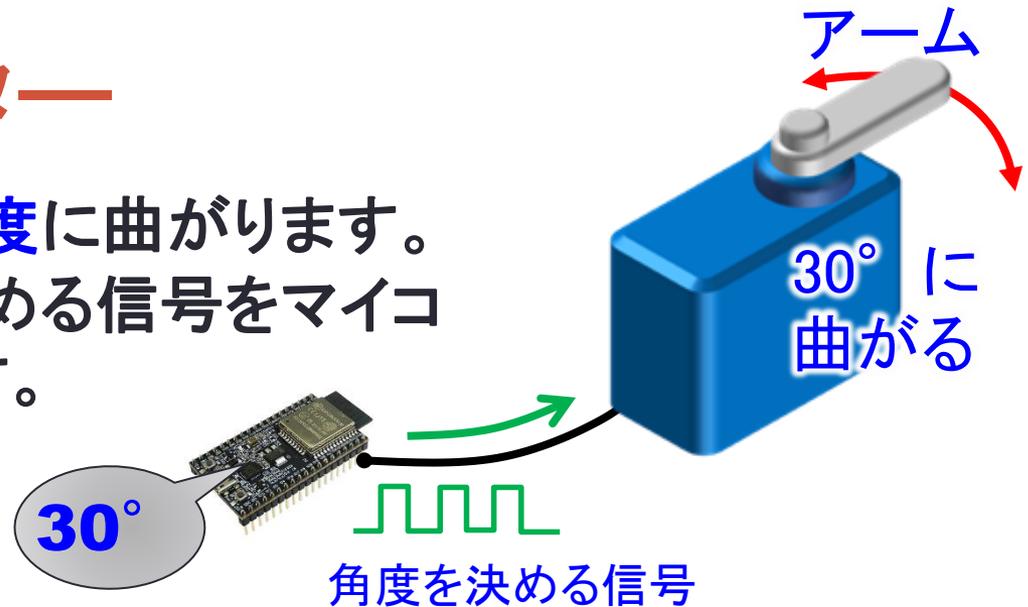
マイコンでコントロール

マイコンが「前進」「後退」「停止」の命令を出し、
モータードライバICがモーターの回路をつなぎ変えます。

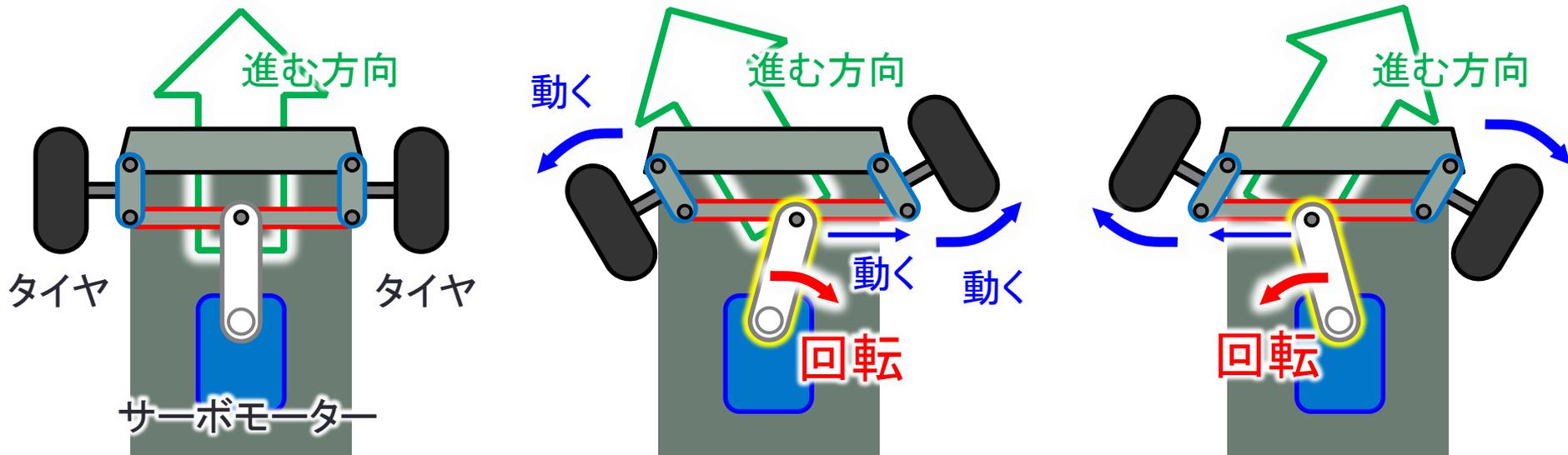


DCサーボモーター

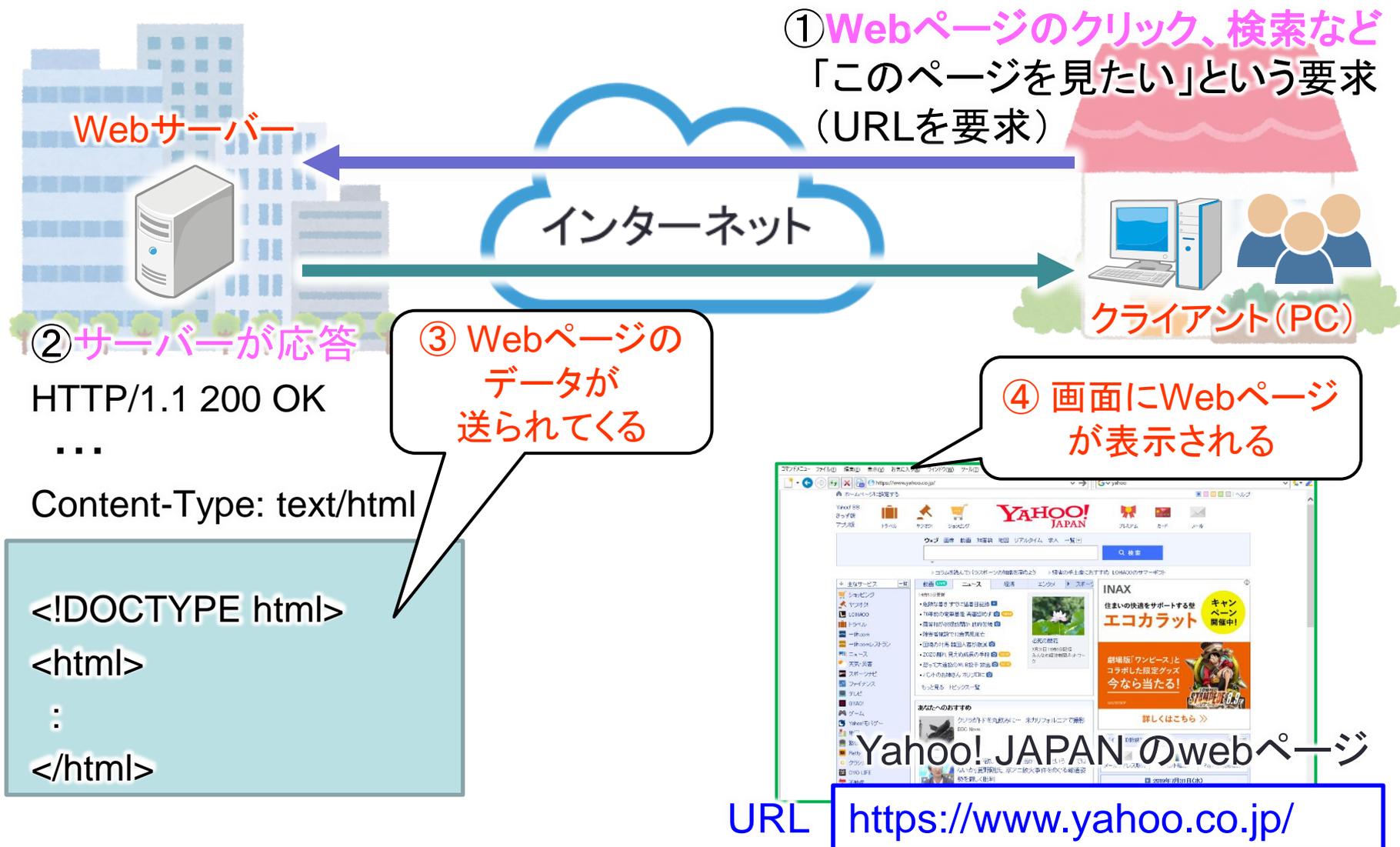
アーム部分が決まった**角度**に曲がります。
電池のほかに、角度を決める信号をマイコンから送る線をつなぎます。



サーボモーターでタイヤの向きを変更(ステアリング)



Webページ



② サーバーが応答

HTTP/1.1 200 OK

...

Content-Type: text/html

```
<!DOCTYPE html>
```

```
<html>
```

```
:
```

```
</html>
```

③ Webページの
データが
送られてくる

④ 画面にWebページ
が表示される



Yahoo! JAPANのwebページ

URL <https://www.yahoo.co.jp/>

IoTバギーカーの概要(動作)

～ IoTを利用 ～

マイコン

(ローカルWebサーバー、WiFiアクセスポイント)

<http://192.168.4.1/>

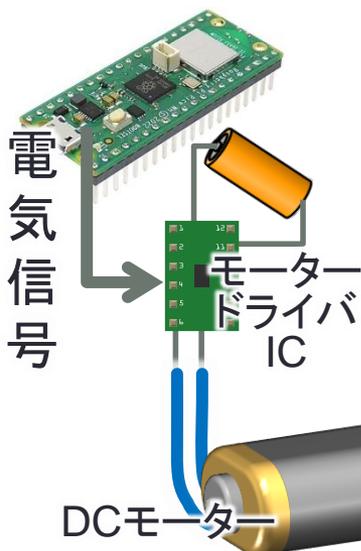
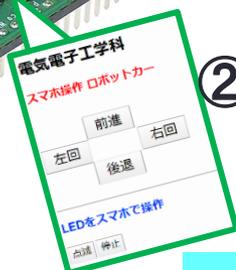
① ブラウザでURL <http://192.168.4.1/> にアクセス

② マイコン(ローカルWebサーバー)が応答。
Webページを送信。

③ 「前進」ボタンを押すと(Webページを要求)

④ ページ要求(前進)に対応した動作:
モータードライバICへ電気信号を送る

⑤ DCモーターが回る!



回転



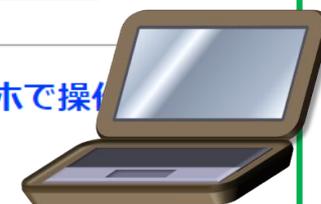
電気電子工学科

スマホ操作 ロボットカー



LEDをスマホで操作

点滅 停止



Webページの中身

- WebページはHTMLで書かれています。
- ブラウザ(プログラム)が解釈して、画像・動画や文字を表示。

HTMLで書かれたWebページ

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,initial-scale=1,minimum-scale=1">
  <title>電気電子工学科 2019</title>
</head>
<body>
  <h1>電気電子工学科</h1>
  <h2>スマホ操作IoTバギーカー</h2>
  <br><br><br>
  <input type= "button" value="前進" style="font-size:32px; width: 100px; position: relative; left:
100px;" onclick="location.href=/forward"><br>
  <input type= "button" value="左回" style="font-size:32px; width: 100px;" onclick="location.href=/left">
  <input type= "button" value="右回" style="font-size:32px; width: 100px; position: relative; left: 96px; "
onclick="location.href=/right"><br>
  <input type= "button" value="停止" style="font-size:32px; width: 100px; position: relative; left: 100px; "
onclick="location.href=/stop">
</body>
</html>
```

ボタンの表示と
リンク先

パソコンに表示される
Webページ

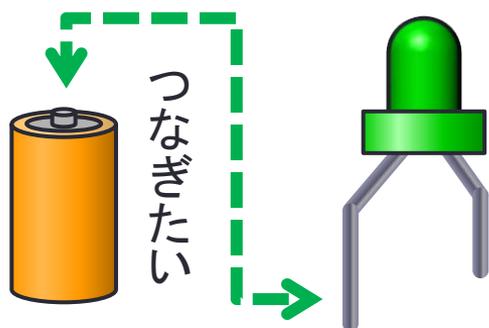
電気電子工学科

スマホ操作 IoTバギーカー

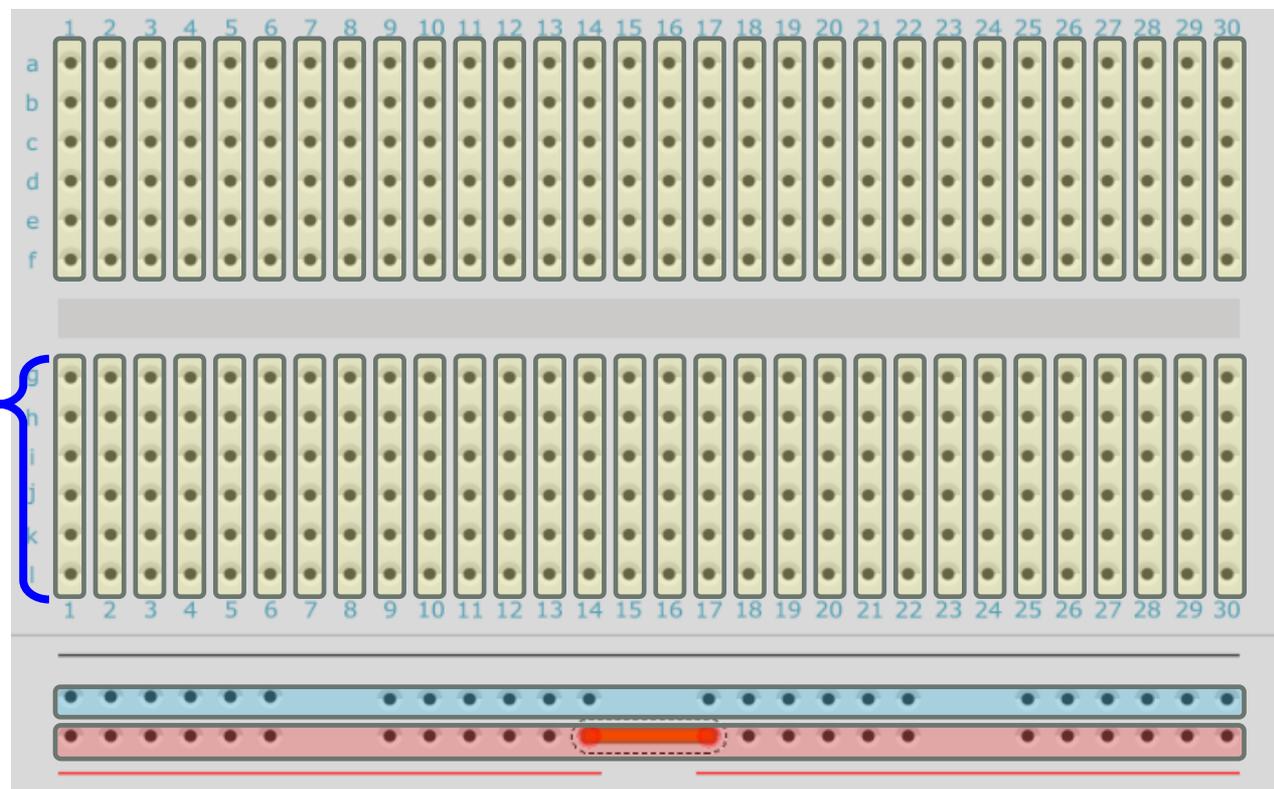
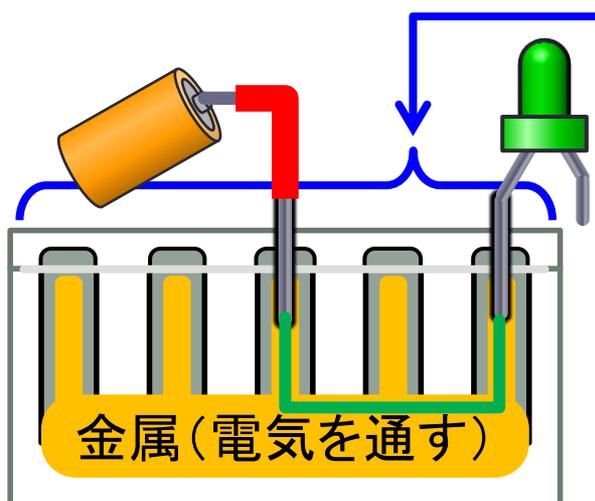


ブレッドボード

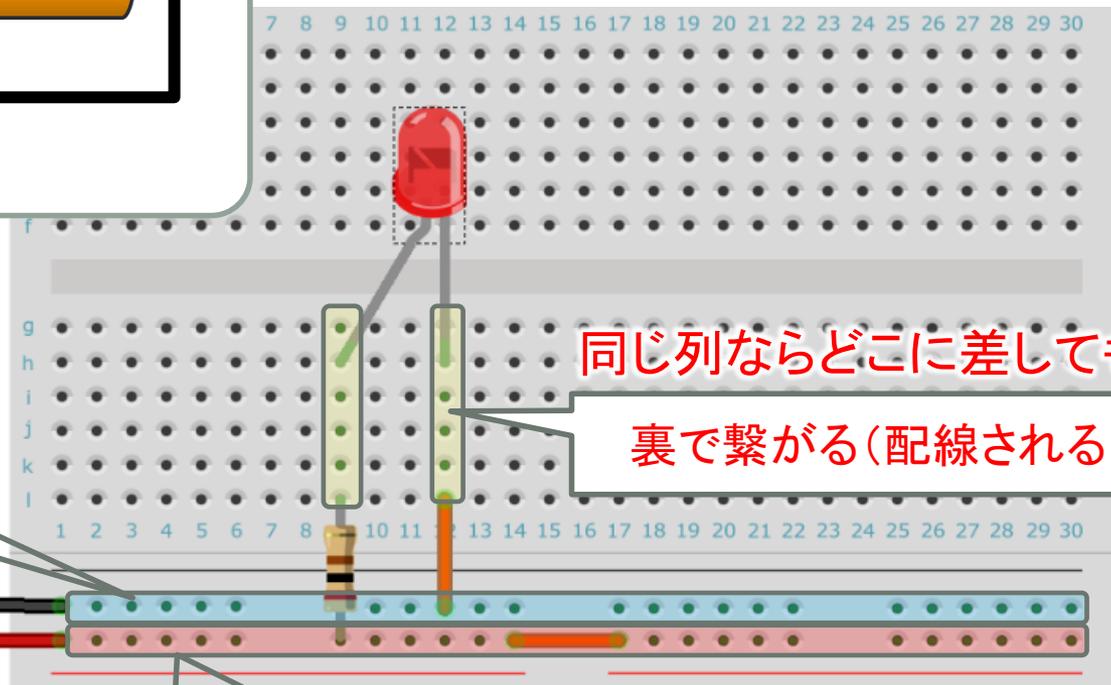
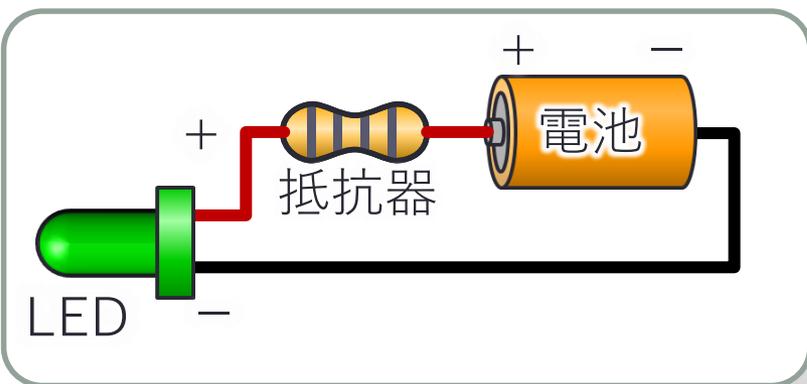
部品を差し込むだけで、回路がつながります。
下の図の縦線と横線が裏でつながっています。



同じ列の穴に差し込む



配線の例



どこも同じ電位: 0V

電池: 0V

電池: 3V



どこも同じ電位: 3V

同じ列ならどこに差ししても同じ

裏で繋がる(配線される)

fritzing

配線(その0) ためしにLEDを光らせる

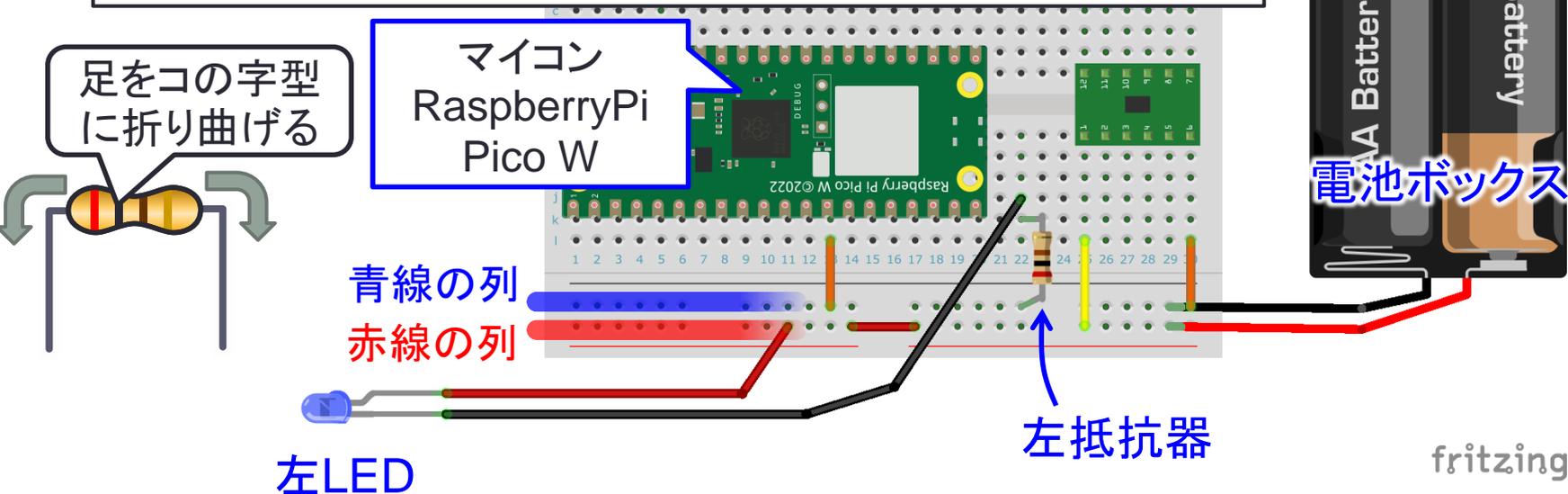
左LEDの+の線 ⇒ ブレッドボード **赤線の列**
 -の線 ⇒ ブレッドボード **下22番の列**

左抵抗器の片方 (済) ⇒ ブレッドボード **下22番の列**
 もう片方 (済) ⇒ ブレッドボード **青線の列**

電池ボックスの黒の線 ⇒ ブレッドボード **青線の列**
 赤の線 ⇒ ブレッドボード **赤線の列**

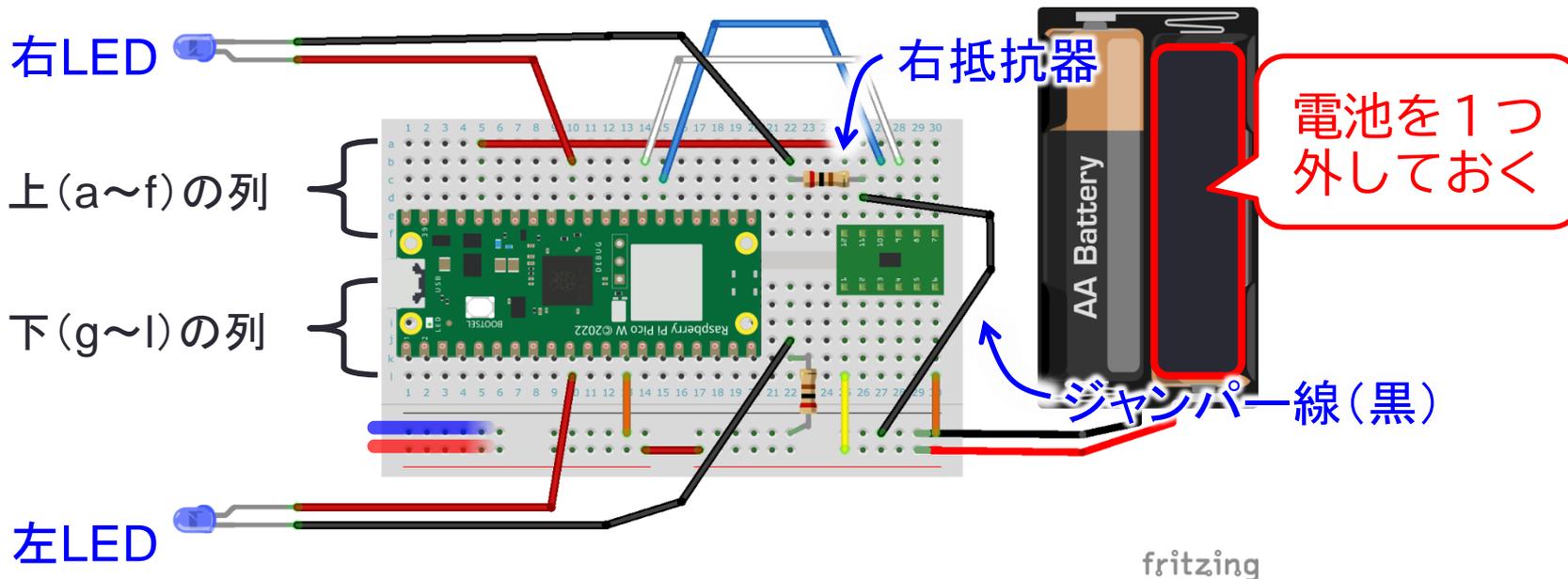
同じ列の
別の穴に入れる

配線が終わってから電池を電池ボックスに入れる



配線(その1) LEDの配線

- 電池ボックス ⇒ 電池を1つ外しておく
- 左LEDの+の線 ⇒ ブレッドボード下10番 (GP7) の列に差し変える
- 右LEDの+の線 ⇒ ブレッドボード上10線 (GP26) の列
 -の線 ⇒ ブレッドボード上22番の列
- 右抵抗器の片方 (済) ⇒ ブレッドボード上22番の列
 もう片方 (済) ⇒ ブレッドボード上26番 (MODE) の列
- ジャンパー線 (黒) ブレッドボードの上26番(MODE)と青線の列(GND)をつなぐ

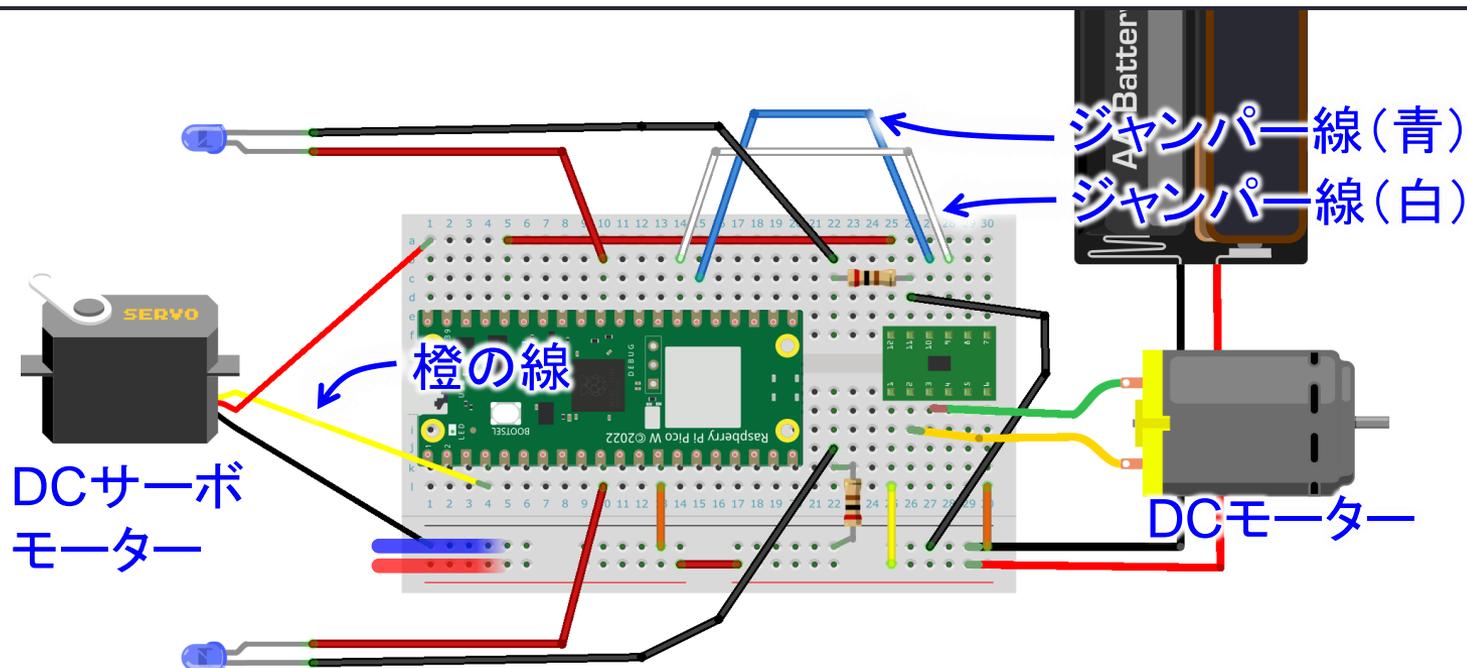


配線(その2) モーターの配線

DCモーター黄の線 ⇒ ブレッドボード下26番(AOUT1)の列
 青(緑)の線 ⇒ ブレッドボード下27番(AOUT2)の列

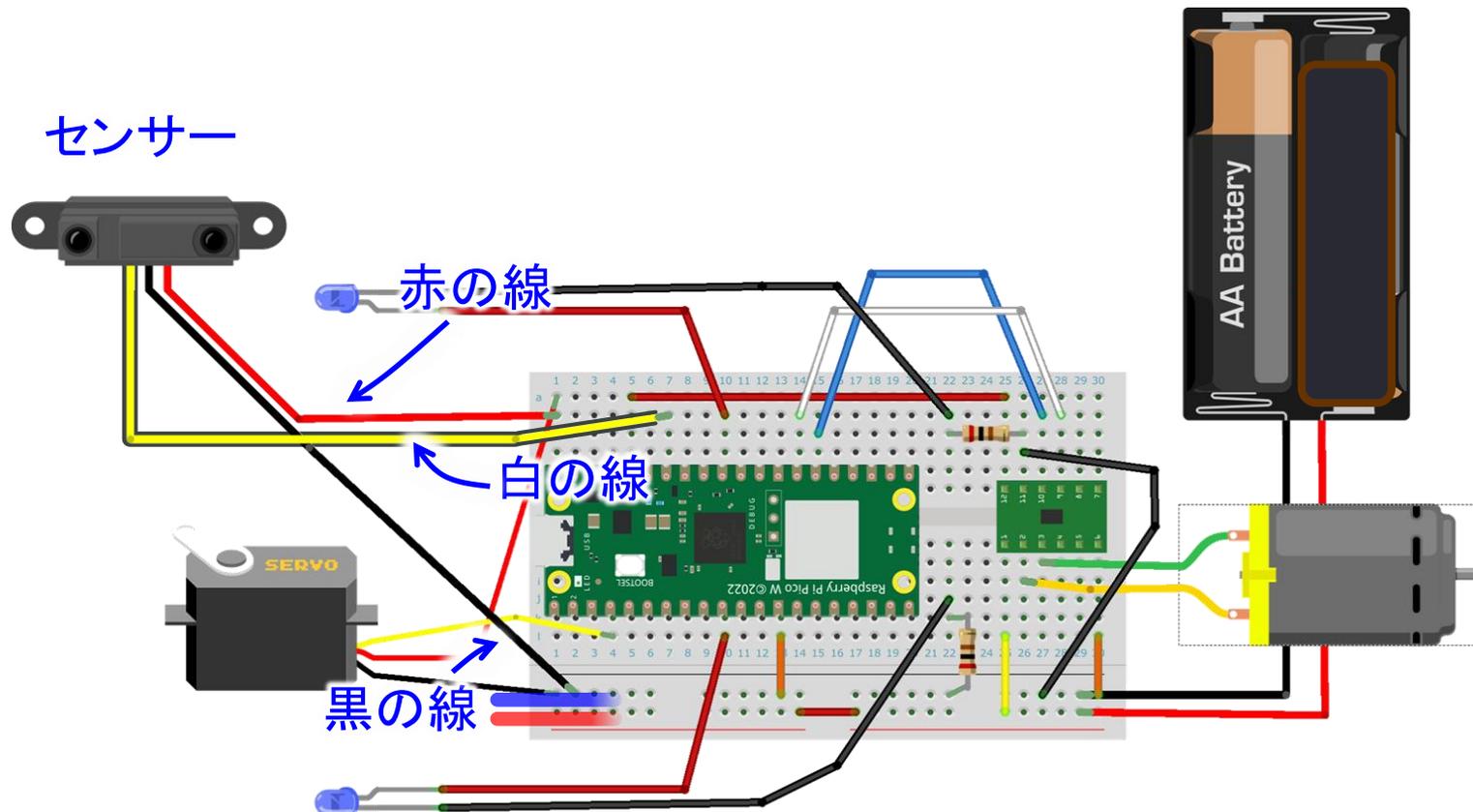
DCサーボ赤の線 ⇒ ブレッドボード上1線(VBUS:5V)の列
 橙の線 ⇒ ブレッドボード下4番(GP2)の列
 黒の線 ⇒ ブレッドボード青線(GND)の列

ジャンパー線(青) ブレッドボードの上15番(GP20)と上27番(AIN1)をつなぐ
 ジャンパー線(白) ブレッドボードの上14番(GP21)と上28番(AIN2)をつなぐ



配線(その3) 測距センサーの配線

測距センサー	白の線	⇒	ブレッドボード上7番(GP28 : ADC2)
	黒の線	⇒	ブレッドボード青線(GND)の列
	赤の線	⇒	ブレッドボード上1線(VBUS:5V)の列

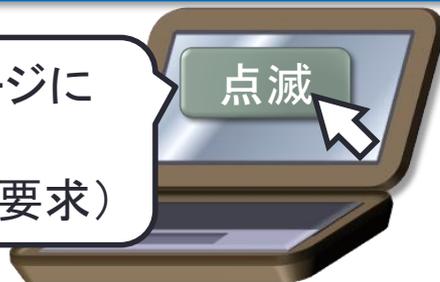


パソコン(スマホ)からリモートでLEDチカチカ！

プログラムでLEDをチカチカ(点滅)させてみます。

Webページの「点滅」ボタンを押す⇒WiFi通信で点滅用のページ(URL:/blink)を要求⇒マイコンが応答⇒ **blink_LED**という関数命令が実行されるように・・・

点滅用のページに
アクセス！
(URL:/blinkを要求)



WiFi通信



点滅用のURL:/blinkが
要求されたら
プログラムの
「**blink_LED**」を実行！

// URLと関数命令の対応関係の設定

```
void ReqURL(){
```

```
  server.on( ...
```

```
  ...
```

// LED操作

```
server.on( "/blink" , blink_LED );
```

```
server.on( ...
```

関数命令追加

--- 関数命令の説明 ---

`server.on("URL", 実行したい関数命令)`

➤ PC、スマホのブラウザから点滅用のページにアクセスされる(URLの要求がある)と、指定した関数を実行する。

「**blink_LED**」のプログラムは次のスライド

blink_LED関数に命令を追加

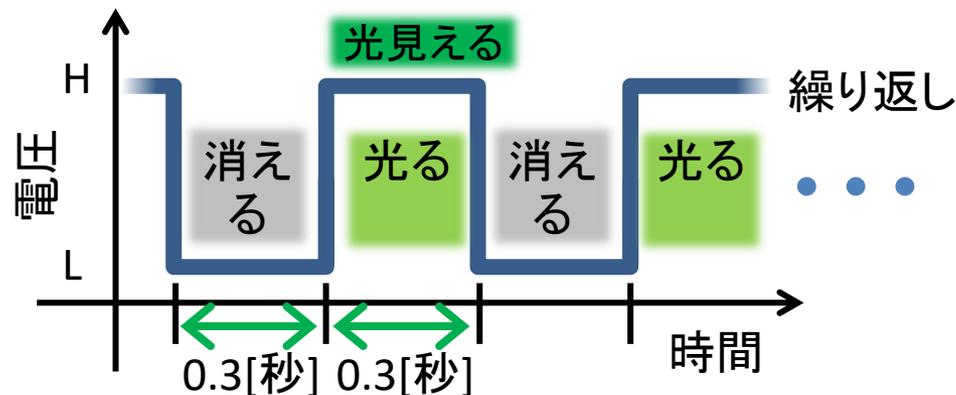
```
// 点滅ボタンに対応
void blink_LED() {
  Server_outo("LED点滅");
  Tenmetu(0.3);
}
```

追加

--- 関数命令の説明 ---

Tenmetu(秒) // 小数可:

- 指定した秒数の間隔でLEDが点滅。
Tenmetu(0.3)だと0.3秒ごとにLEDが点滅。
マイコン内部のタイマー機能を使っています。



プログラムを書き込んで(次の資料)、うまくいったら…

Tenmetu(0.01); など数値を小さくしてみてください。

⇒速すぎて点滅に見えなくなります(残像効果, 蛍光灯と同じ)。

⇒アニメは、1秒間に24枚の静止画で出来ています。

プログラムの書き込み

```

1 package mojiNin_6;
2
3 @import java.awt.*;
4 @import java.awt.event.*;
5
6 public class MNPMain implements ActionListener {
7
8     //アクションコマンド
9     public static final String CMD1 = "コマンド1";
10    public static final String CMD2 = "コマンド2";
11    public static final String CMD3 = "コマンド3";
12    public static final String CMD4 = "コマンド4";
13    public static final String CMD5 = "コマンド5";
14
15    int N = 32; // ニューロン数Nの平方
16    int NN = N*N; // ニューロン数N
  
```



プログラム書き込み



① 接続

- ・ マイクロUSBケーブルをつなぎ、「ボード」と「ポート」を確認し、書き込みボタン → を押す。次からは、ボタン → を押すだけ。

② 「ツール」をクリック

③ 確認する

ボード: Raspberry Pi Pico W
ポート > COM1以外を選ぶ

④ ここを押すとプログラムを書き込む

⑤ 右下に書き込み完了と出る

書き込み完了

Arduino IDE 2.3.6

ファイル 編集(E) スケッチ ツール ヘルプ(H)

自動整形 Ctrl+T

スケッチをアーカイブする

ライブラリを管理...

シリアルモニタ

シリアルプロット

Firmware Updater

SSLルート証明書を書き込み

ボード: "Raspberry Pi Pico W" >

ポート: "COM4" >

Reload Board Data

ボード情報を取得

出力

Debug Level: "None" >

Resetting

Debug Port: "Disabled" >

Converting

C++ Exceptions: "Disabled" >

Scanning for RP2040 devices

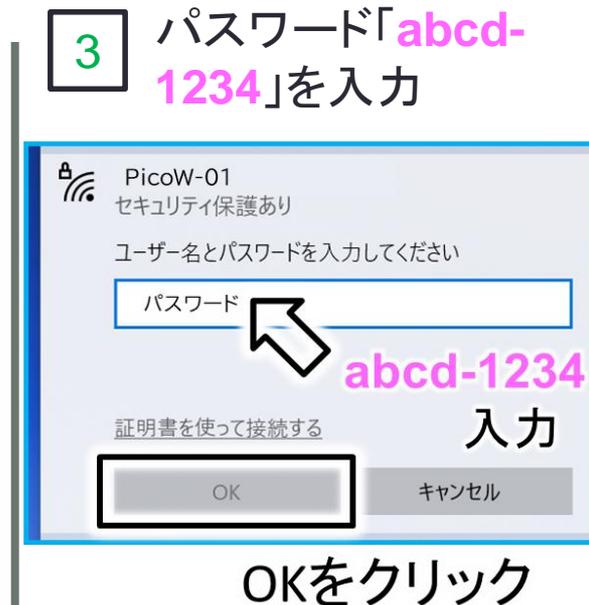
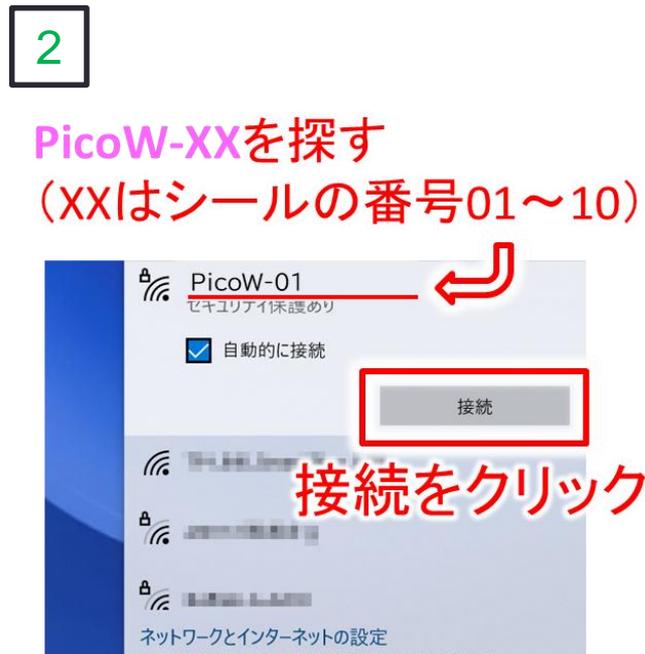
Flashing D: (RPI-RP2)

Wrote 804352 bytes to D:/NEW.UF2

行 95、列 32 Raspberry Pi Pico W COM4D

パソコン(スマホ) と PicoWでWiFi通信手順

- マイコン(WiFiモジュール)の電源を入れると、プログラムが起動し、**アクセスポイント & Webサーバー**になる。



- Edge, chromeなどの**ブラウザ**で「<http://192.168.4.1/>」にアクセスすると、Webページが表示されるので、操作してみる。



リモートで前進(1秒後停止)!

プログラムでDCモーターを前進させてみます。

「前進」ボタンを押す⇒WiFi通信で点滅用のページ(URL: /forward)を要求
⇒マイコンが応答⇒ forward という関数命令が実行されるように・・・

前進用のページ
にアクセスしたい
(URL: /forward)



WiFi通信



前進用のURL: /forward
が要求されたら
プログラムの
「forward」を実行!

// URLと関数命令の対応関係の設定

```
void ReqURL(){
  server.on( ...
  ...
  // バギー操作用
  server.on( "/forward" , forward );
  server.on( ...
}
```

関数命令追加

--- 関数命令の説明 ---

server.on("URL", 実行したい関数命令)

- PC、スマホのブラウザから点滅用のページにアクセスされる(URLの要求がある)と、指定した関数を実行する。
forward

forward関数に命令を追加

プログラムでDCモーターを前進させてみます。

前進用のページ
にアクセスしたい
(URL:/forward)



WiFi通信



前進用のURL:/forward
が要求されたら
プログラムの
「forward」を実行！

```
// 前進ボタンに対応
void forward() {
  Server_ouo("前進");
  ...
  Tenmetu(0.02);
  Motor(1);
  Handle(HC);
  delay(Ton);
  LEDoff();
  ...
}
```

追加

--- 関数命令の説明 ---

Motor(1)

➤ 電気信号でDCモーターを駆動(前進)

Handle(パルス幅)

➤ PWM信号でDCサーボモーターを駆動
(ステアリングが動く)

➤ HCはパルス幅:1.5ms

タイヤがまっすぐ走る向きになるような数値

Handle(xx)は、いじるとステアリン
グ部分が壊れます..

リモートで後進(1秒後停止)!

```
void ReqURL(){
  ...
  server.on( "/forword", forword);
  server.on( "/left", left );
  server.on( "/right", right );
  server.on( "/back", back );
```

変更

```
// 後進ボタンに対応
void back() {
  Server_outo("後退");
  ...
  Motor(2);
  Handle(HC);
  delay(Ton);
  ...
}
```

追加

「left」「right」の
プログラムは
次のスライド



左・右・後用の
ページにアクセス
(URL: /left,
/right, /back)

WiFi通信



各URLが要求されたら
対応するプログラム
「left」「right」「back」を実行

--- 関数命令の説明 ---

Motor(2)

➤ 電気信号でDCモーターを駆動(後進)

Handle(パルス幅)

➤ PWM信号でDCサーボモーターを駆動
(ステアリングが動く)

➤ HCはパルス幅: 1.5ms

タイヤがまっすぐ走る向きになるような数値

リモートでステアリングを右／左へ操作！

```
// 左回ボタンに対応
void left() {
  Server_outo("左回");
  ...
  TenmetuL(0.1);
  Motor(1);
  Handle(HL);
  delay(Ton);
  ...
}
```

TenmetuL(0.1);
Motor(1);
Handle(HL);

追加

```
// 右回ボタンに対応
void right() {
  Server_outo("右回");
  ...
```

TenmetuR(0.1);
Motor(1);
Handle(HR);

追加

--- ステアリングの説明 ---

ステアリングには、DCサーボモーターというものが付いています。

サーボモーターはPWM信号を加えると
思った角度で止まってくれます。

--- 関数命令の説明 ---

Handle(パルス幅)

- PWM信号でDCサーボモーターを駆動
(ステアリングが動く)
- HLはパルス幅:1.7ms
タイヤが左に曲がる向きになるような数値
- HRはパルス幅:1.3ms
タイヤが右に曲がる向きになるような数値

※何かにぶつかって壊れないように、1秒後にDCモーターを停止させています。

衝突防止用の距離センサーをつける！

赤外線センサーを用いて障害物との距離を計測！

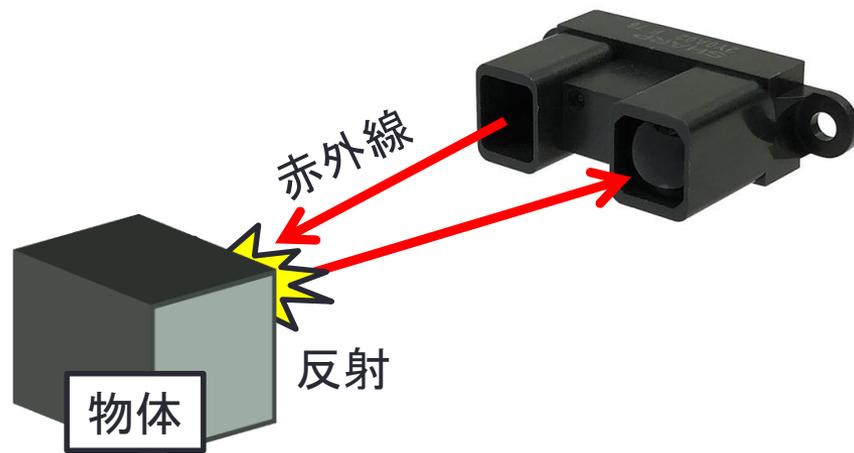
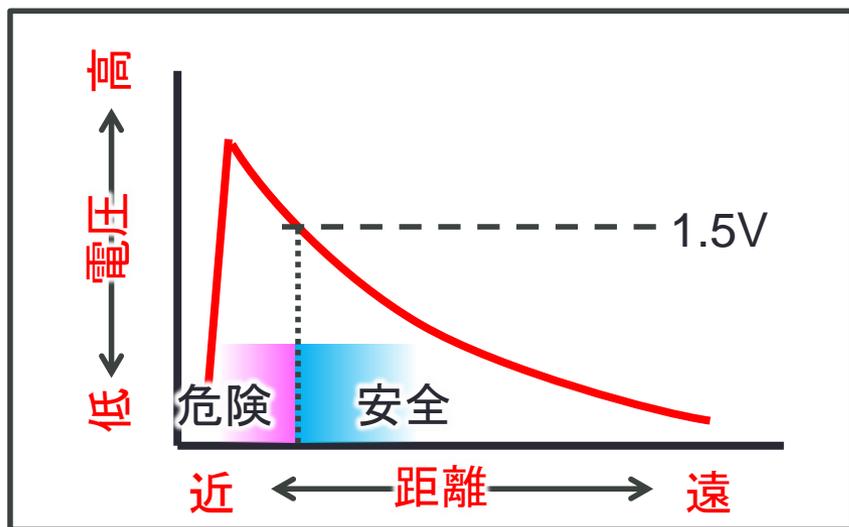
//測距センサ出力が THRESHOLD 以上なら停止 (23行目くらい)

```
const float THRESHOLD = 1.5f; // 距離の判断[v]
```

変更

--- 距離を測るしくみ ---

赤外線とセンサーを使用して非接触で距離を測定することができます。
物体からの赤外線の反射を電圧に変換してマイコンで測定します。
だいたい20cm～100cmくらいの距離を測ることができます！



結果を表示して衝突防止センサーを調整する！

ツール → シリアルモニタ



表示を見て赤外線センサーの値THRESHOLDを調整しよう！



“voltage”の値が THRESHOLD 以下ならPSD_OK が 1 になってモーターが回転できる

モバイルバッテリーで試験走行！

プログラミング用のUSBケーブルを抜いて、WiFiモジュールに**モバイルバッテリー用のUSBケーブル**を繋げて下さい。

床で試験走行し、**DCモーターが回転する時間**をお好みで変更してみてください。

➤ **DCモーターが回転する時間**

```
#define Ton    1000    // DC motor : wait(ms)
```

今は、1000に設定されています(1000ミリ秒=1秒)

お疲れ様でした

現在、様々な電子機器がインターネットに繋がろうとしています。
今回のIoTバギーの原理を応用すると、画像も送れますし、ネットワーク経由で画像認識・音声認識などのもっと高度なことも出来ます。

鹿児島高専(Ⅰ類・Ⅱ類)では、最新の技術にも対応できるように理論はもちろんのこと、実験・実習も充実させています。

◆ 今回の主な使用機器

Raspberry Pi Pico W(完成品)	¥1,375(税込)
ブレッドボード 6穴版 EIC-3901	¥520(税込)
マイクロサーボ SG92R	¥780(税込)
DRV8835使用ステッピング & DCモータドライバモジュール	¥550(税込)
シャープ測距モジュール GP2Y0A21YK	¥550(税込)
TAMIYA バギー工作基本セット(入手困難)	¥2,090(税込)
※バギー工作基本セットを改造しています。	