

# ロボカップジュニア サッカーロボット作成講習会

## 第二回 ロボット制御プログラム1

# 講習会の進め方

---

- 第一回 ロボット制作
- 第二回 ロボット制御プログラム1
  - ロボットを動かす/センサから外部の情報を取り込む
- 第三回 ロボット制御プログラム2
  - プログラムのテクニック

# 第二回の予定

---

- 10:00～10:30 導入 ロボットとは
- 10:30～11:30 LEDの点滅/プログラムの管理  
(昼休憩)
- 12:30～13:30 モータを動かす
- 13:30～14:30 センサの値を読む
- 14:30～15:15 課題演習
- 15:15～15:30 質疑応答/連絡事項
- 15:30～ 解散

(フィールドで自由にロボットを動かしてみてください)

# ロボカップの目的

---

- 目標
  - 「FIFAルールに従い2050年までにワールドカップチャンピオンチームに11台のヒューマノイドロボットで勝利する」
- 目的
  - この過程で開発される多くの技術が様々な分野に応用されることにより、より豊かな社会が実現されることを期待する

# ロボカップジュニアの目的

---

- 最終目標に向けた次世代研究者の育成
- ロボカップを通じたものづくり教育
  - 次のことを期待
    - 科学知識の習得
    - 科学的、論理的な考え方を身につける
    - 自ら工夫し挑戦する
    - だれとでも(大人・外国人)会話できる

# 本講習会の目的

---

ロボカップジュニアの大会に出場するために

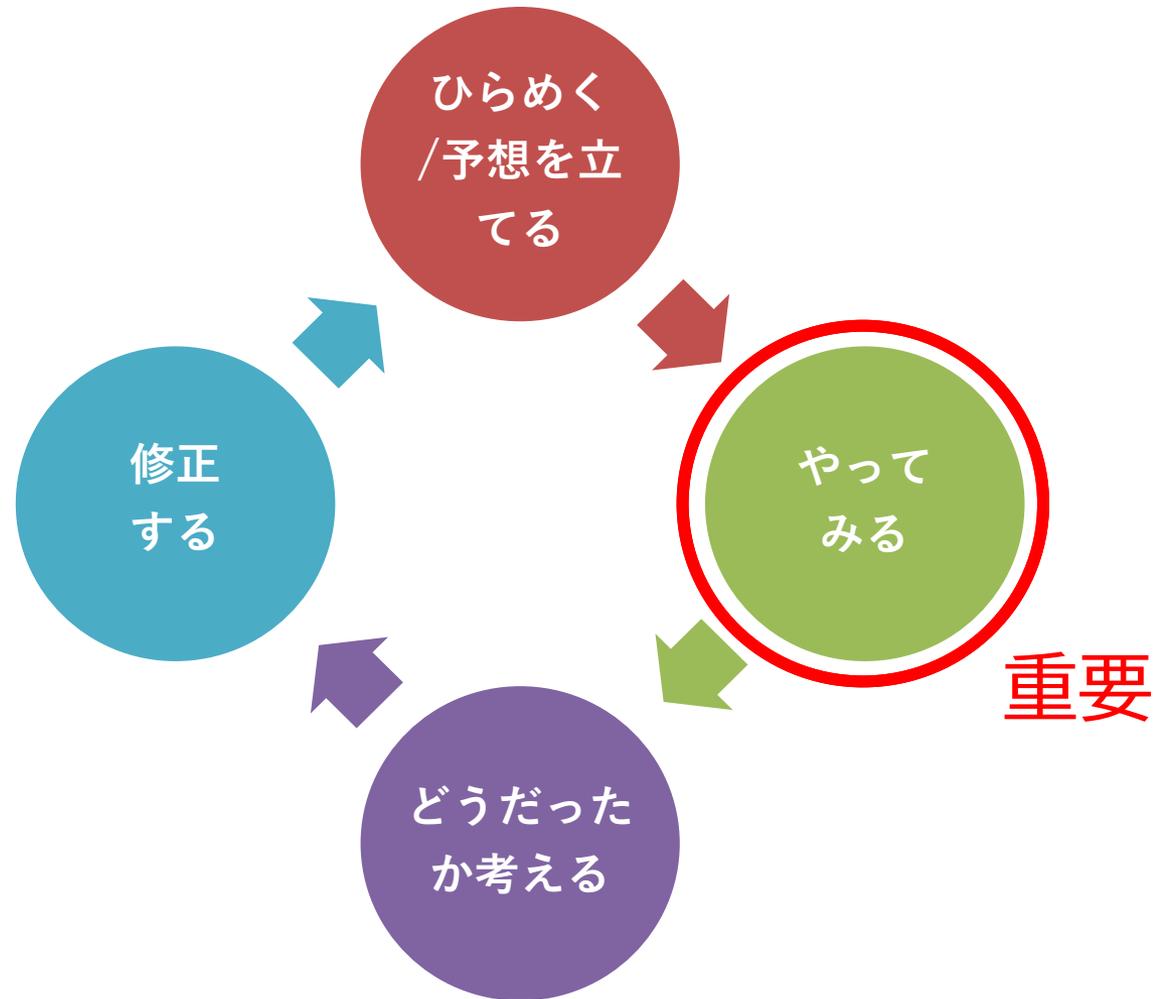
1. ロボットの基本的なしくみを理解する
2. ロボットを思いどおりに動かせるようにプログラミング
3. ロボットの改造
4. ルールと戦略(どう戦えばいいか)を理解

# 学習のポイント1

---

- メモする
  - 目的、目標
  - やること、やったこと(計画)
  - 観察したこと(事実)
  - ひらめいたこと、思いついたこと(思い)
  - 聞いたことなど 何でも

# 学習のポイント2



# お願い 子供たちに

---

質問があるときは手をあげて  
自分で質問して下さい

# お願い 保護者の方へ

---

- 子供は要領よくできませんが、  
できるだけ見守ってやってください
  - コツを教えてあげてください
  - 力が足りないときは手伝ってやってください
- スタッフを呼ぶのはお子さんに  
させてください

# サッカーロボットのしくみ

# ロボットとは？

---

- 人間に類似した形態をもち、自動的に作業を行う機械装置
- 感じる、判断する、動くの三つがそなわっている人工物
- 環境に溶け込み人の役に立つもの(見えないロボット)
  - 例: エアコン、洗濯機、テレビ、車

※出典 東嶋和子著『ロボット教室』光文社

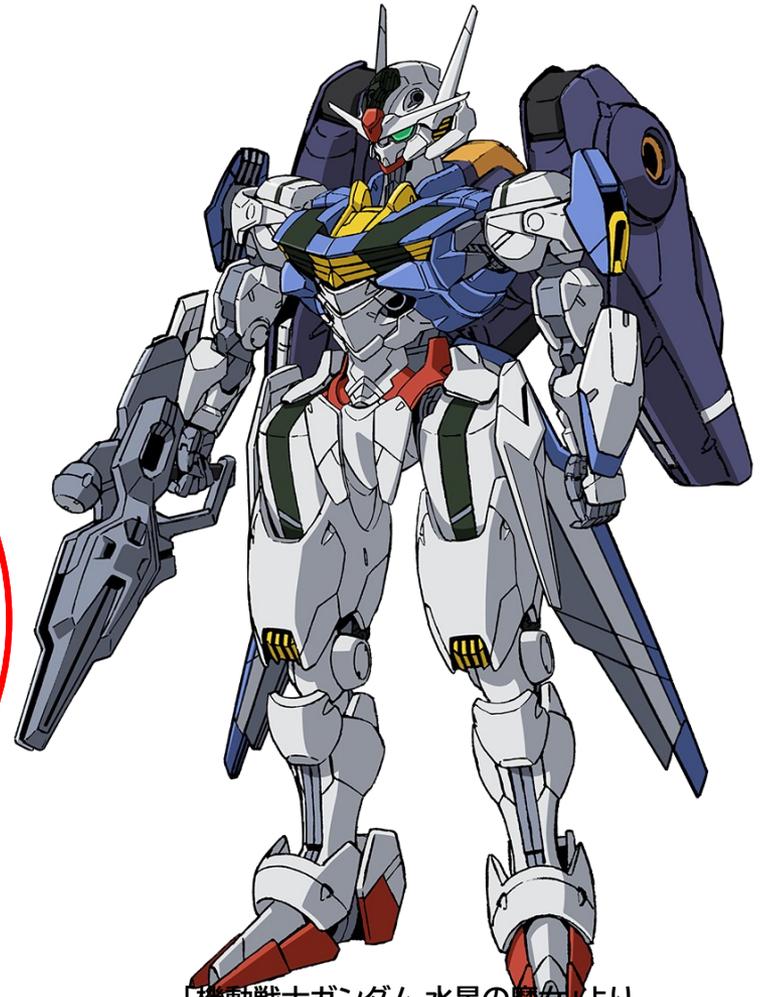
# ロボットって？

こっち



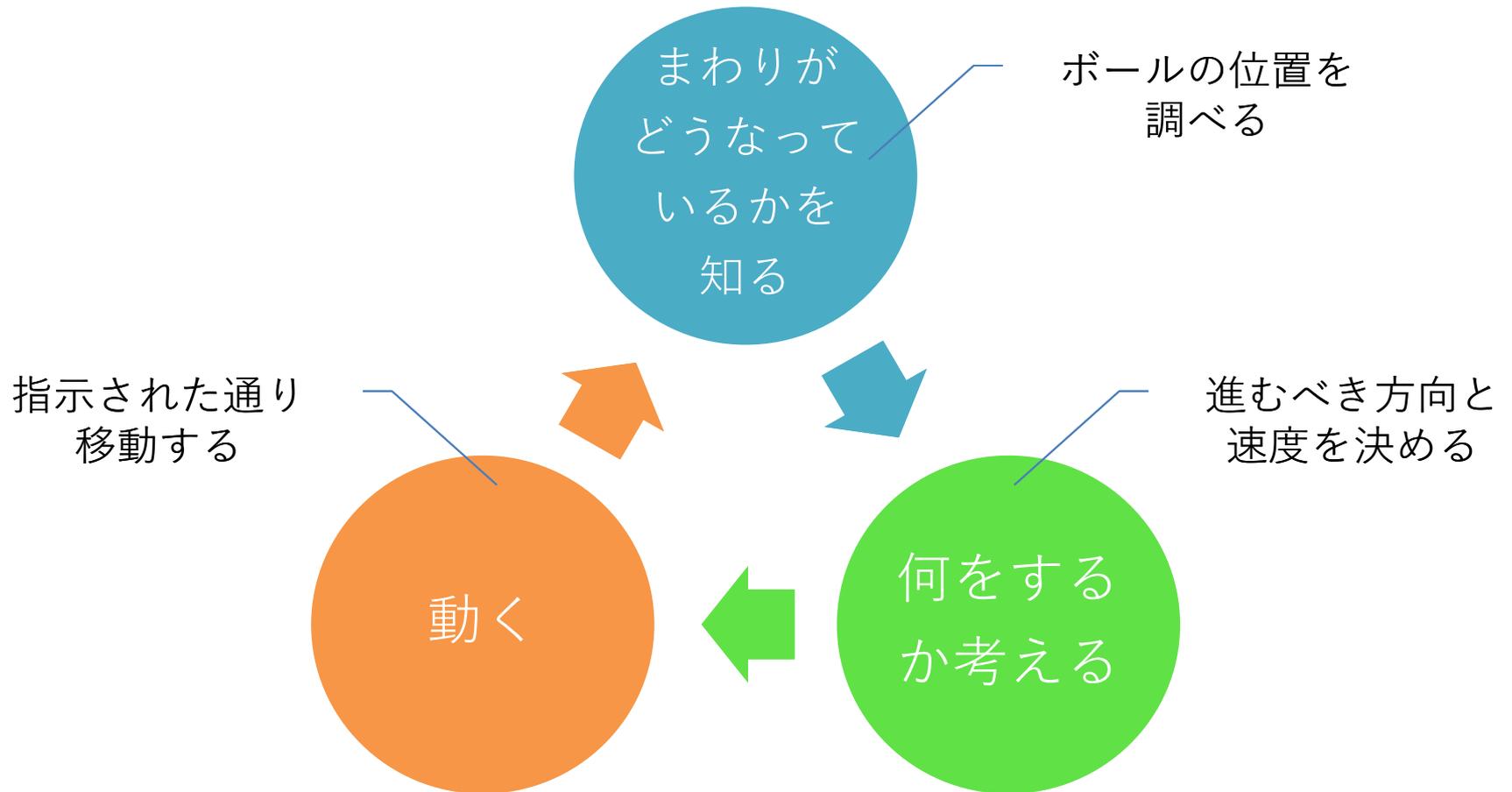
「Dr.スランプ」より

「PLUTO」より



「機動戦士ガンダム 水星の魔女」より

# 自律行動



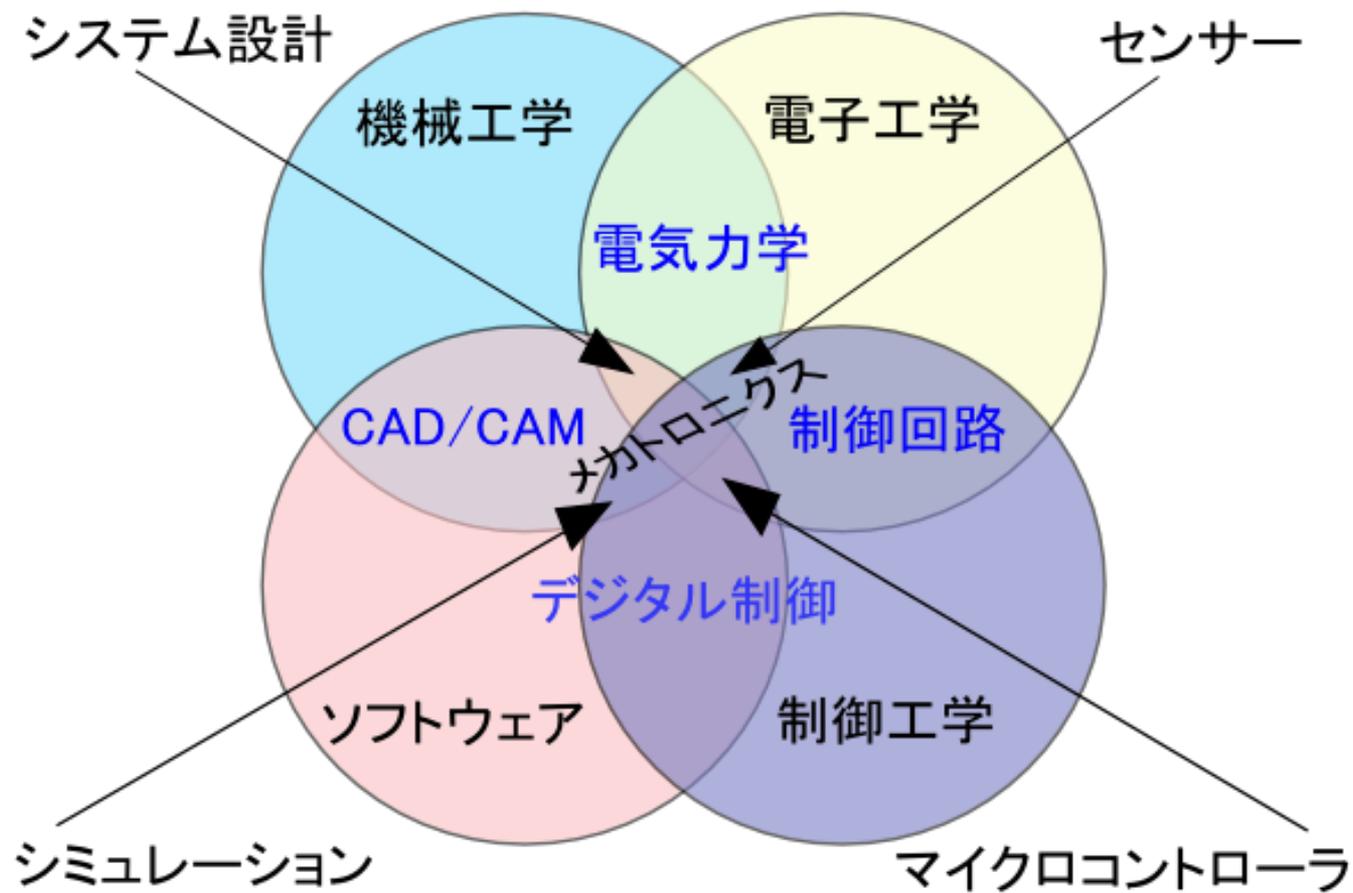
# ロボットに自律行動をさせるには？

---

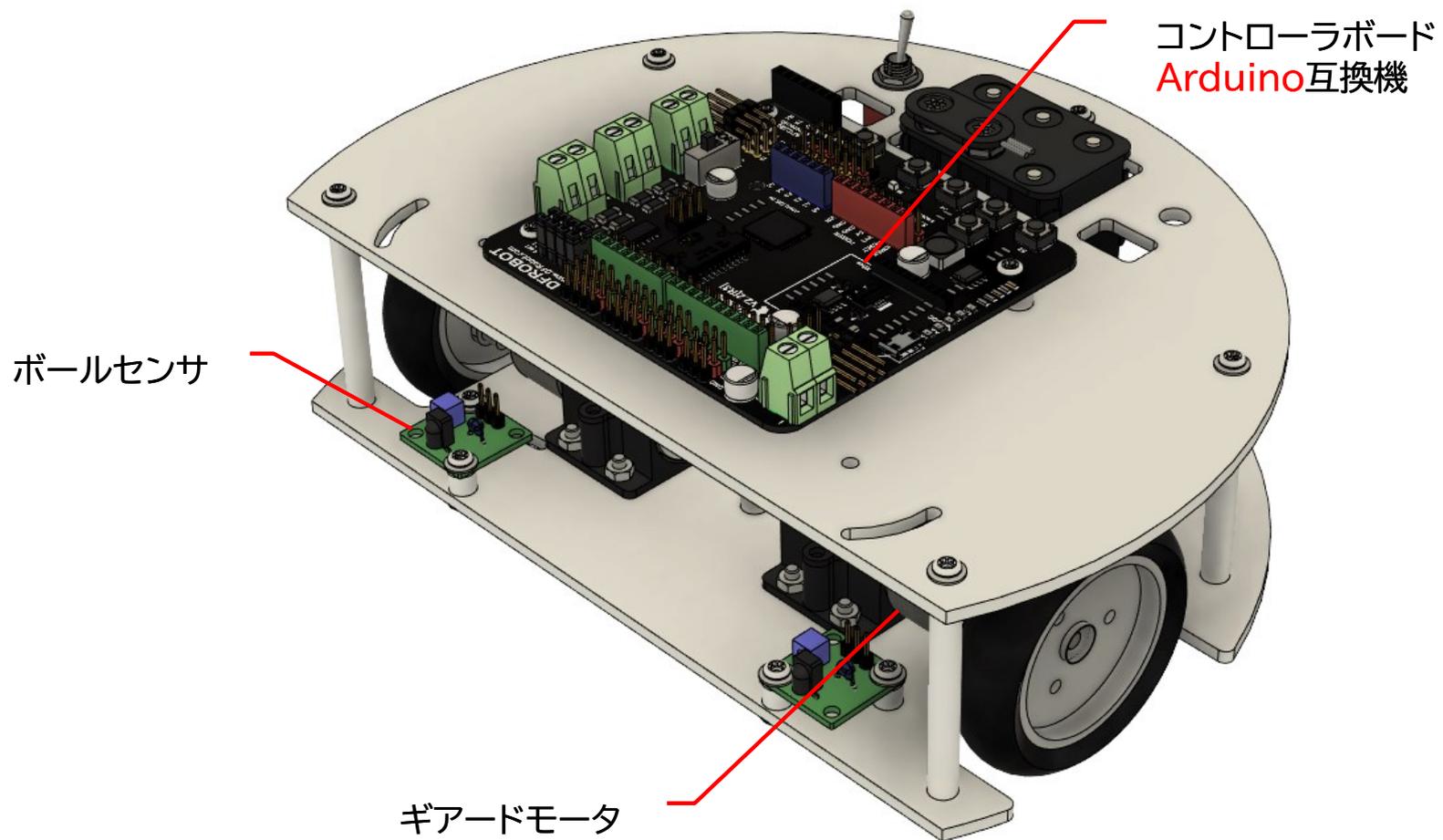
- 課題を解くための手順を考える
  - ⇒ アルゴリズム
  
- 手順をロボットにわかる言葉に書きかえる
  - ⇒ プログラミング

# ロボット工学

Wikipediaより



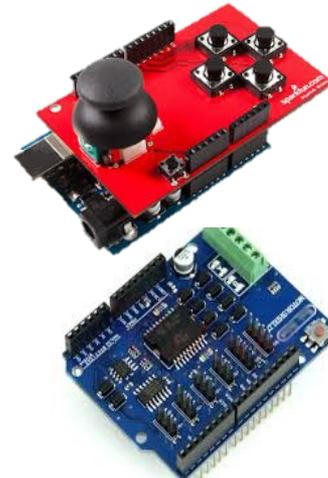
# ORJAロボ



# Arduinoとは

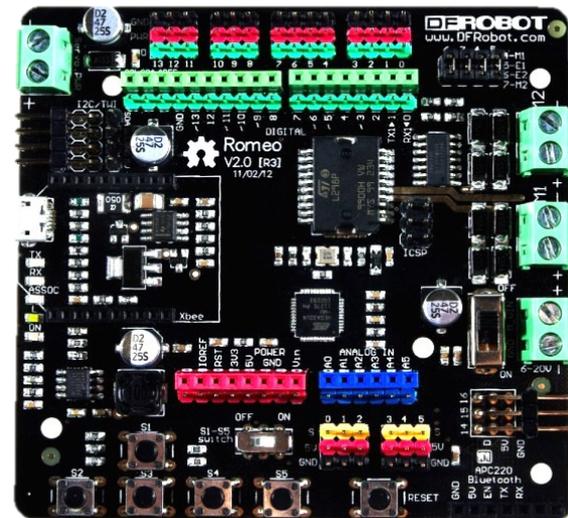
イタリア生まれの教育用マイコンとプログラミング環境

- デザイン科の学生向けに作られたのでプログラムしやすい
- ハードウェア/ソフトウェアとも誰でも使用料なしで勉強できるようになっている
  - 知りたいことがインターネットで見つかる
  - たくさんの増設ボードが販売されている



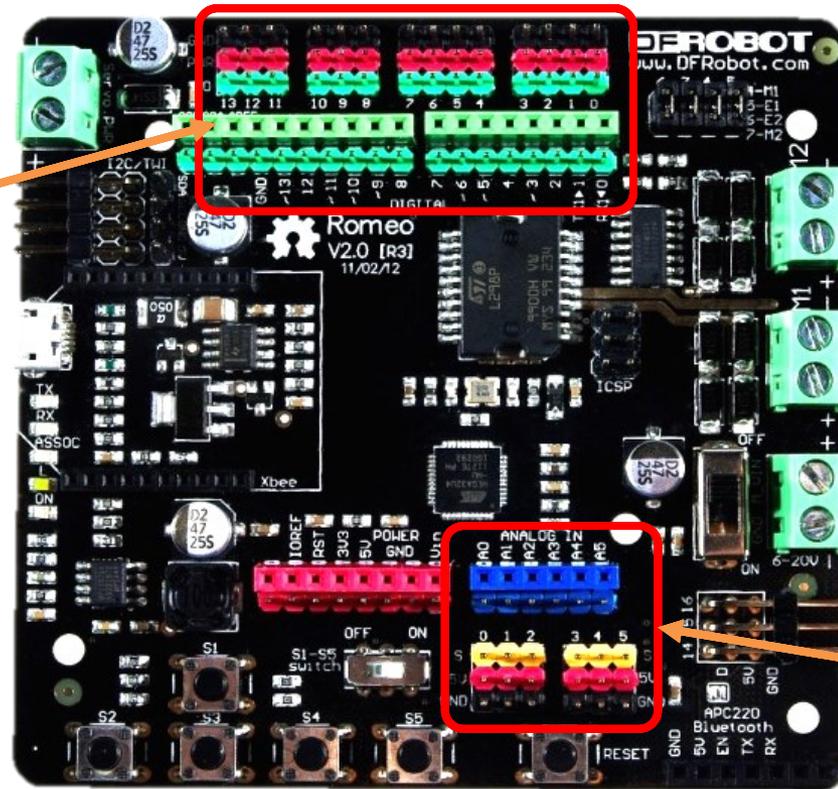
# コントローラボード

- Arduino<sup>ごかんき</sup>互換機 DFRobot社RomeoV2
  - Arduino Leonard互換
    - Arduino用のプログラム・増設ボードを使える
  - 標準機から追加されている機能
    - モータを動かす回路(2個)
    - ボタン(5個)
    - センサー接続ピン
      - 各信号線に5V/GNDを増設
      - I<sup>2</sup>C用専用ピン増設(3セット)
    - 無線通信増設口



# コントローラボード

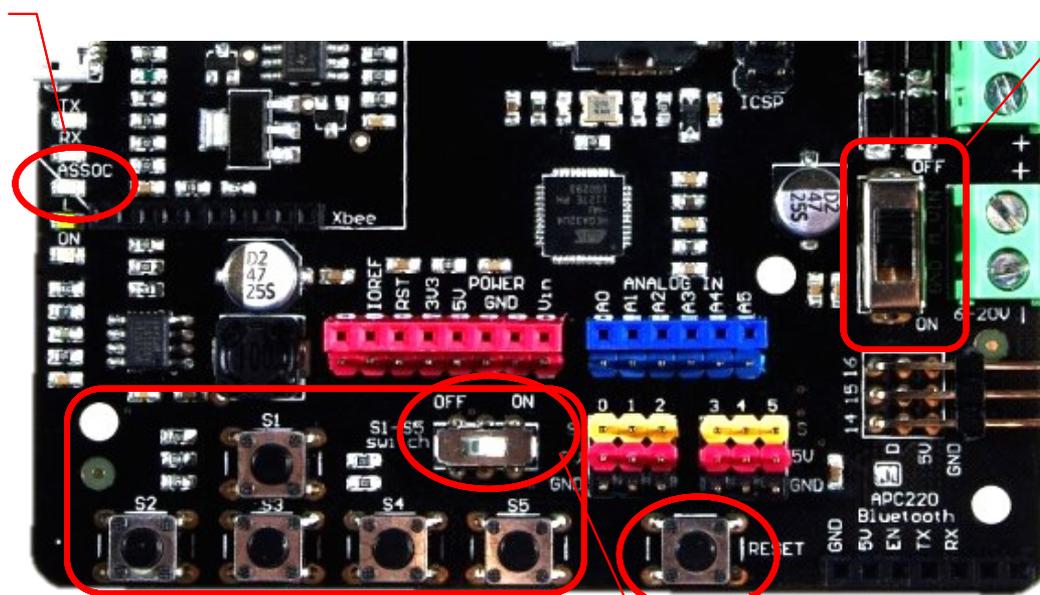
デジタル入出力ピン  
0～13



アナログ入力ピン  
0～5

# 準備 LEDとボタン

デジタル13番ピンにつながつたLED



電池の電力を電子回路に使うかどうかのスイッチ (ONに設定)

アナログ0番ピンにつながるボタン

アナログ0番ピンとボタンをつなぐかどうかの切替スイッチ (ONに設定)

リセットスイッチ

# アルゴリズム

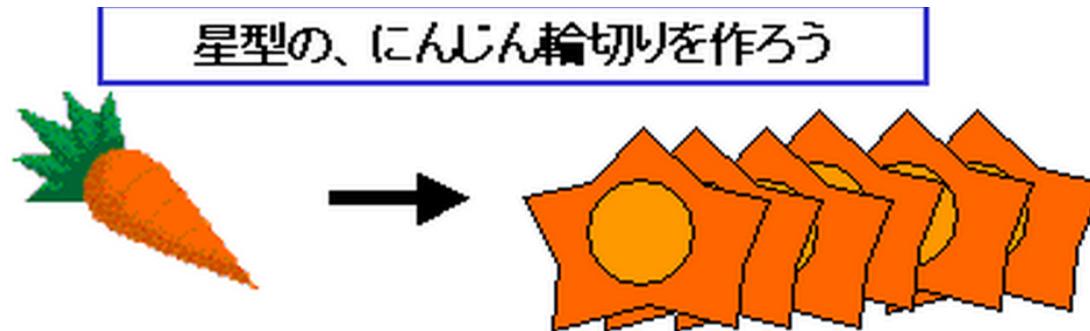
# アルゴリズムとは？

---

- コンピュータで計算を行うときの「計算方法」
- 何か物事をやるときの「やり方」
  
- 「やり方」を工夫して、よりよいやり方を見つけていきます

国立情報学研究所のホームページ  
[http://research.nii.ac.jp/~uno/algo\\_3.htm](http://research.nii.ac.jp/~uno/algo_3.htm)

# より良いアルゴリズムとは



星形にんじんを30枚作るには、  
何回包丁を使う？

国立情報学研究所のホームページ  
[http://research.nii.ac.jp/~uno/algo\\_3.htm](http://research.nii.ac.jp/~uno/algo_3.htm)

# より良いアルゴリズムとは

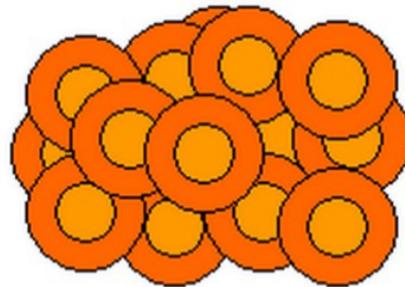
星型の、にんじん輪切りを作ろう

30枚作るには？



輪切り：  
31回包丁をいれる

星型切り：  
1枚ずつ10回包丁をいれる

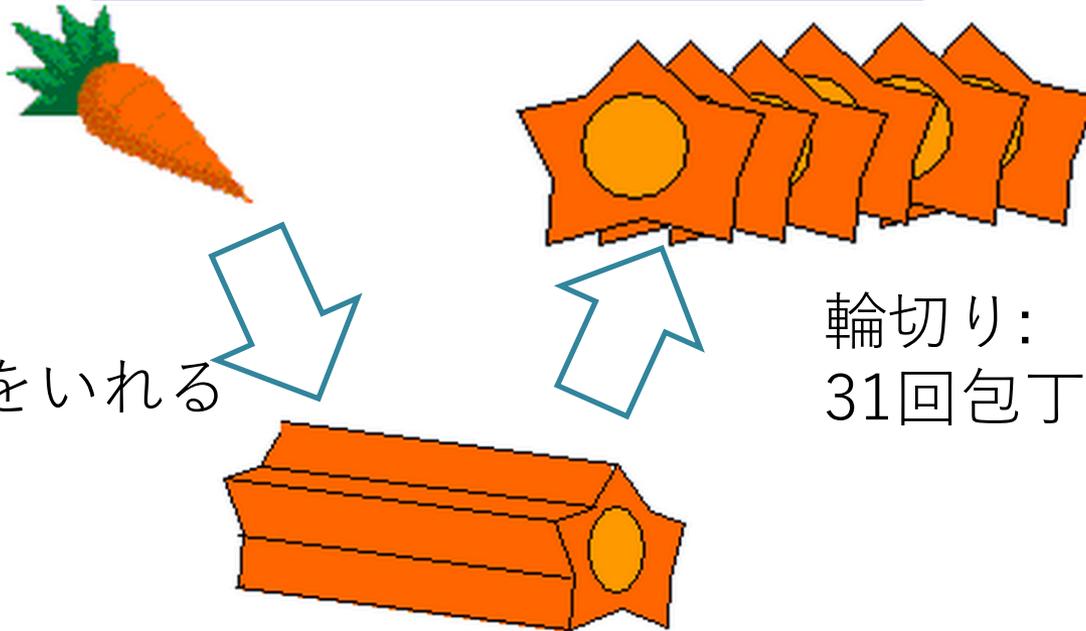


合計 **331**回 包丁をいれる (31回+10回×30枚)

# より良いアルゴリズムとは

星型の、にんじん輪切りを作ろう

30枚作るには？



星型切り:  
10回包丁をいれる

輪切り:  
31回包丁をいれる

合計**41**回 包丁をいれる(10回 + 31回)

方法1(331回)に比べ、**290回少ない**

# パソコンの使い方と ロボットへのプログラムを入れる準備

# マウスの使い方

- クリック：左ボタン(人差し指)を押してすぐに離す
- ダブルクリック：クリックをすばやく2回する
- ドラッグ：左ボタンを押したままマウスを移動する



マウスカーソル



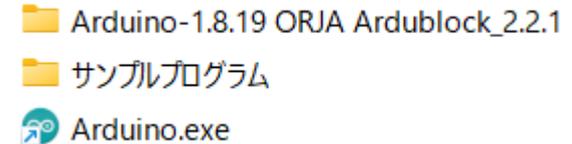
# Arucublockを動かす1

- パソコンの画面を“デスクトップ”といいます
- デスクトップにある「ORJA講習会」フォルダのなかの Arduino.exe をダブルクリックします



.exeが表示されていない  
パソコンもあります

- ArduinoIDEが動き出します
  - “IDE”とは  
Integrated Development Environment  
(統合開発環境)の略称です



```
sketch_sep06a $
digitalWrite(7, LOW);
}

void loop()
{
  _ABVAR_1_Ba11 = analogRead(5);
  if (( ( _ABVAR_1_Ba11 ) < ( 500 ) ))
  {
    setRomeoMotor(1, 95);
    setRomeoMotor(2, 95);
  }
  else
  {
    setRomeoMotor(1, 80);
    setRomeoMotor(2, -80);
  }
}
```

# ArduBlockを動かす2

- ArduinoIDEのメニューから  
[ツール]-[ArduBlock]を選びます



- ArduBlockが動きます

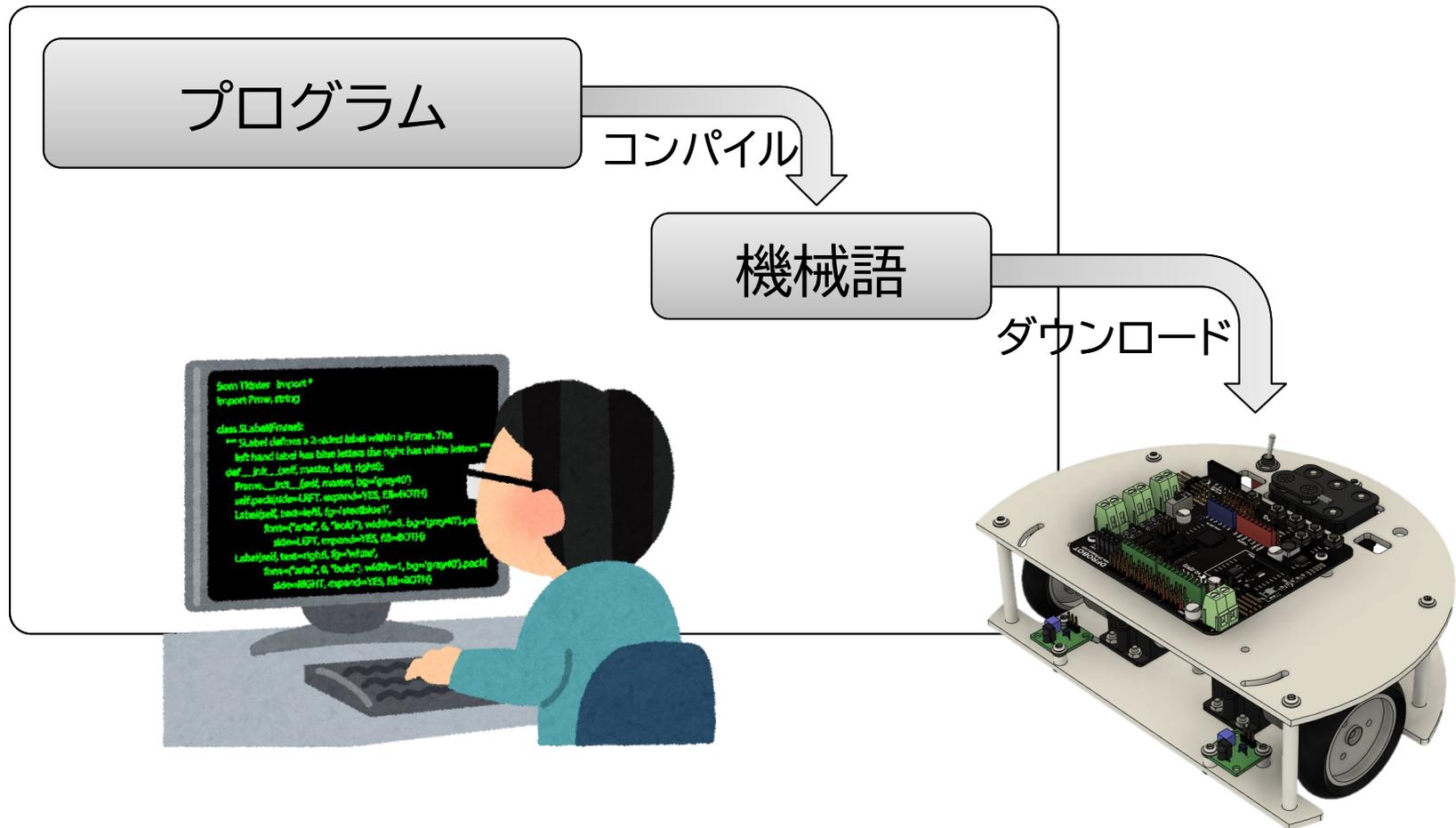


# プログラム実行までの流れ

---

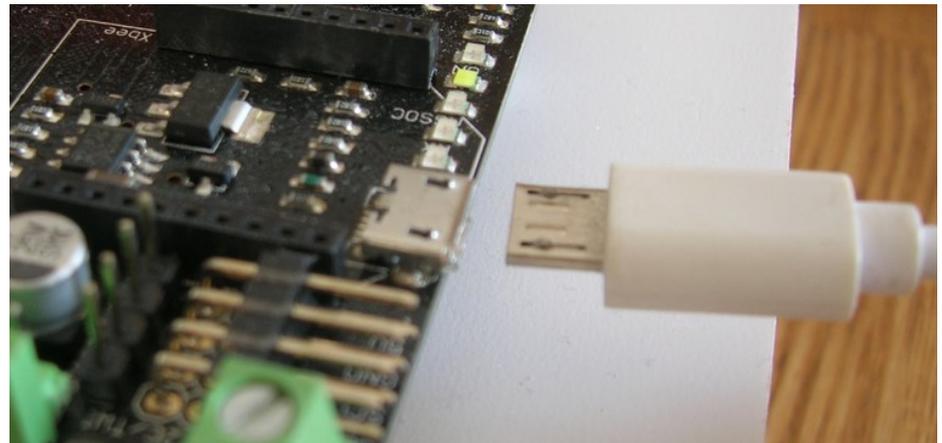
1. パソコン上でプログラムを作成
  - 人がわかる言葉・図形で表現
2. コンパイル・リンク
  - ロボットがわかる言葉に変換
  - あらかじめ用意されているプログラムと合体
3. ロボットへダウンロード(アップロード)
  - パソコンからロボットへプログラムを送る
4. ロボット上でプログラムを動かす

# プログラム実行までの流れ



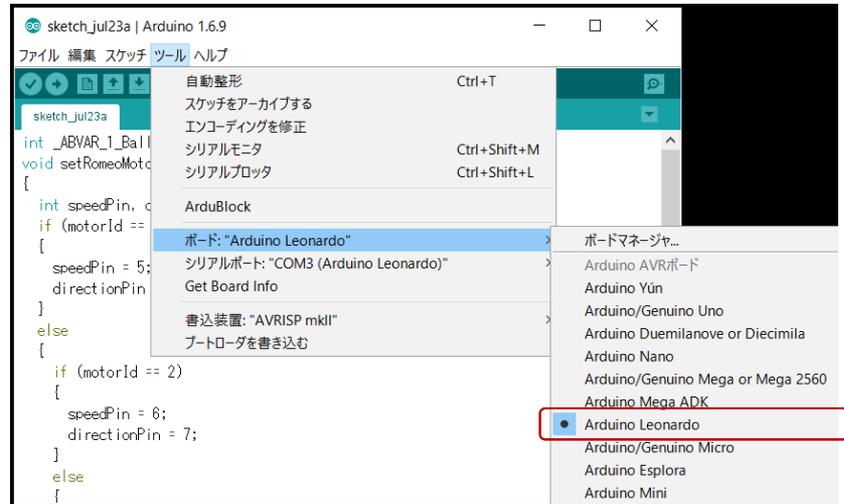
# ロボットにプログラムを書込む準備1

- ロボットとパソコンをUSBケーブルで接続してください
  - ケーブルをつなぐ前にロボットの電源がオフになっていることを必ず確認してください  
ロボットが急に動き出し机から落ちることがあります

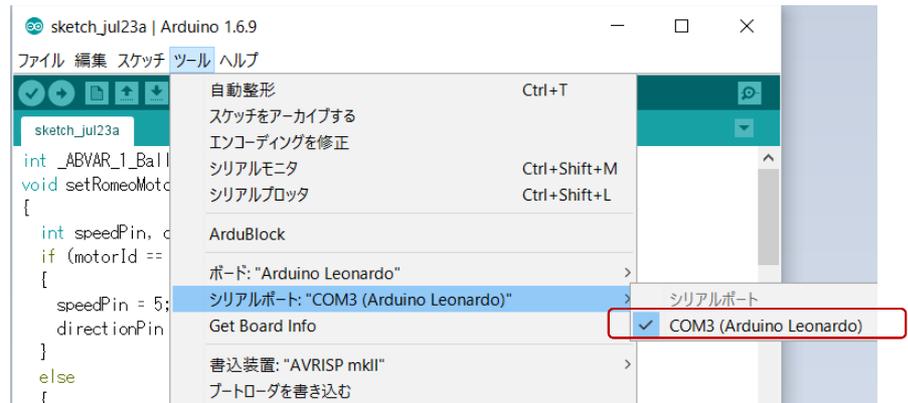


# ロボットにプログラムを書込む準備2

ArduinoIDEのメニュー[ツール]-[ボード]から“Arduino Leonardo”が選ばれていることを確認します



Arduinoメニューの[ツール]-[シリアルポート]から“COMx (Arduino Leonardo)”が選ばれていることを確認します



# プログラミング

## LED点滅 (Lチカ)

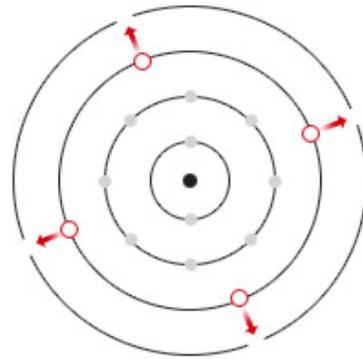
# LED

- LED(Light Emitting Diode):発光ダイオード

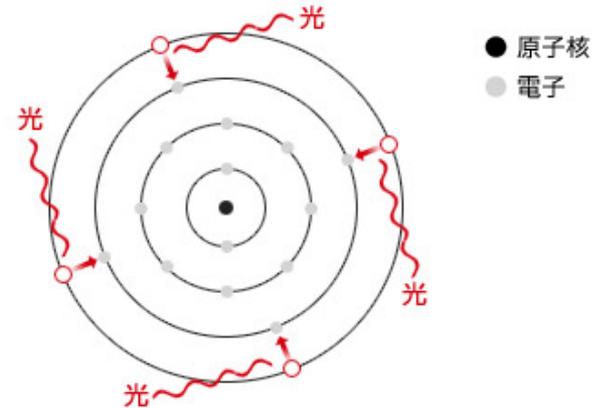


# LEDの原理

## • 励起発光



外部からエネルギーを与えると、電子は軌道を飛び出し、高エネルギー状態になる。

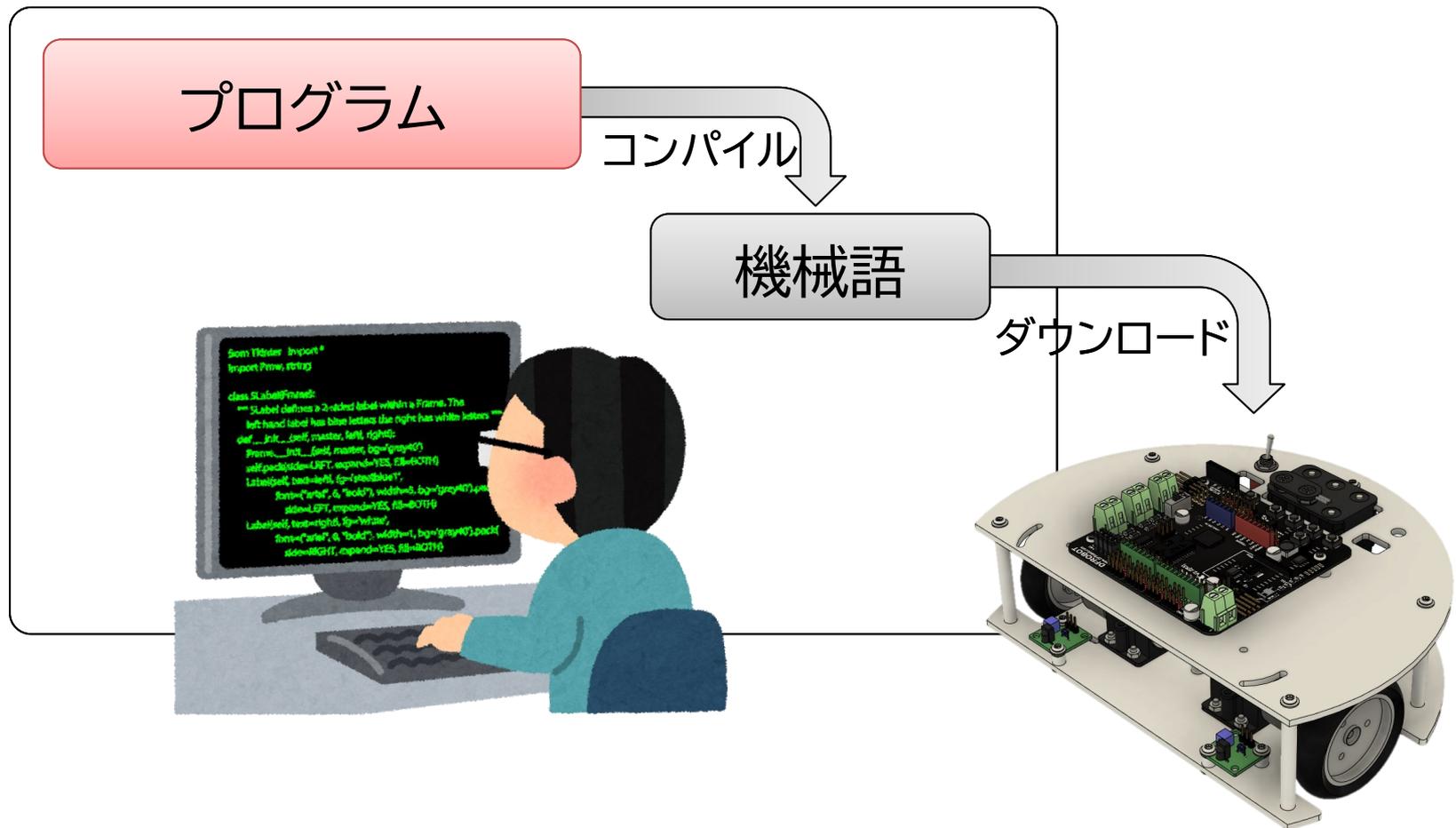


電子が元の軌道に戻るとき、過剰なエネルギーが光として放出される。

## • エネルギー変換効率

- |          |        |
|----------|--------|
| - ろうそく   | ~0.04% |
| - 白熱電球   | ~5%    |
| - 蛍光灯    | ~15%   |
| - 高電圧放電管 | ~25%   |
| - LED照明  | ~40%   |
| - 蛍      | ~40%   |

# プログラムを作ってみよう



# LEDを1秒明るくするアルゴリズム

- 動きを細かく分析、分解する



# LEDを点灯するプログラム

- LEDを1秒間点灯するプログラム



# LEDを点灯するプログラム

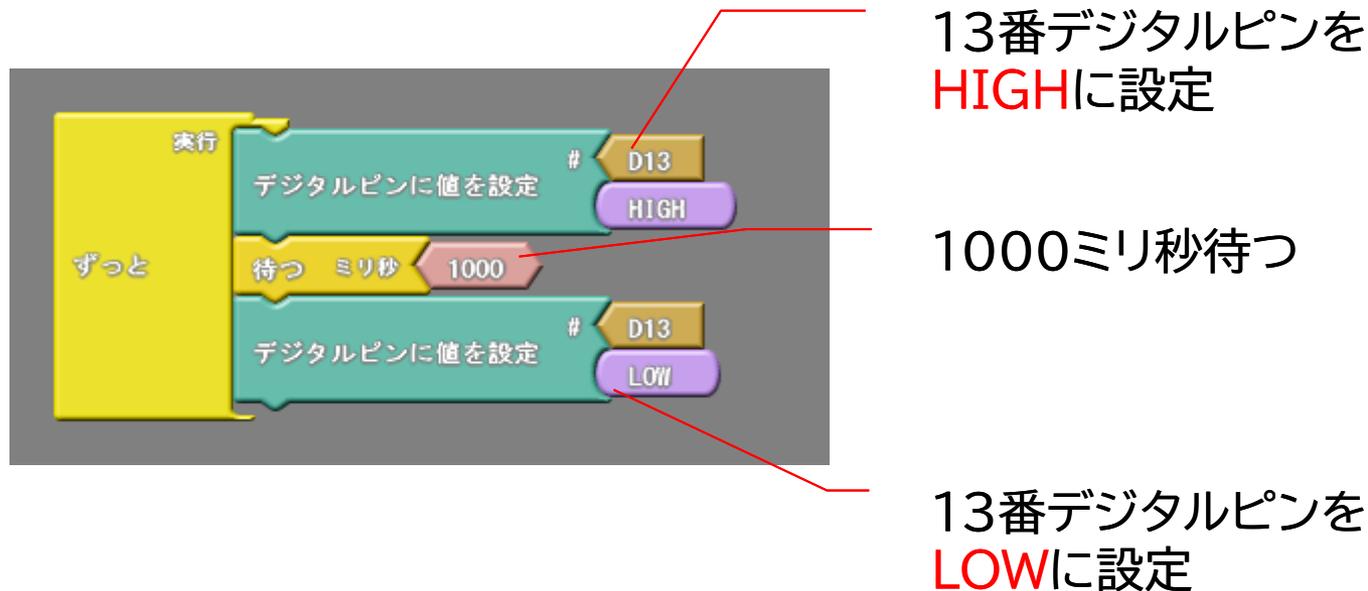
## LEDを一秒間点灯するプログラム



処理の流れ

プログラムは上から順番にひとつずつ実行される

# プログラム解説



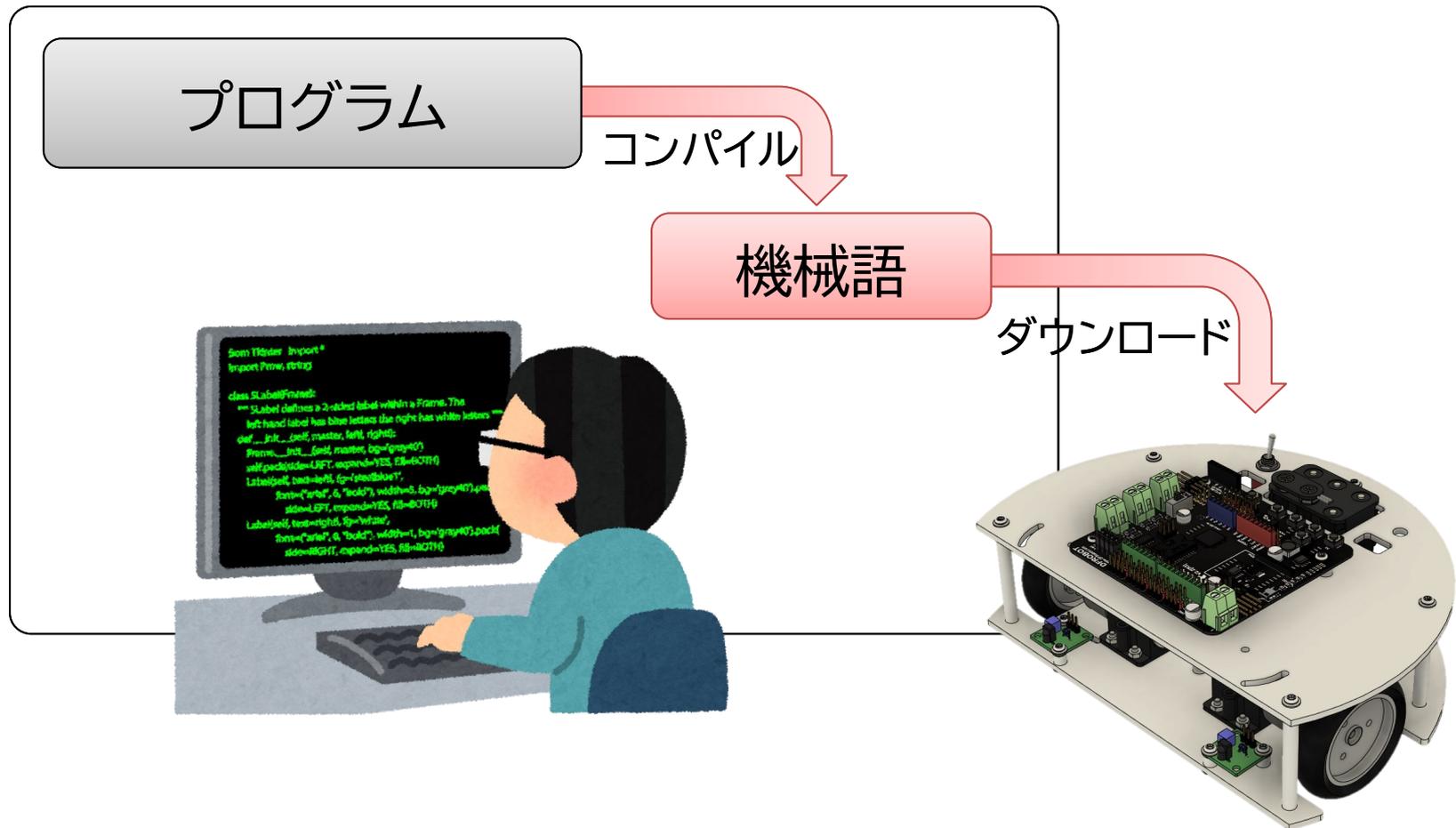
『デジタル回路』は“0ボルト”と“電源電圧値”の2つの電圧値だけを扱う  
0ボルトを**LOW**、電源電圧値(RomeoV2は5ボルト)を**HIGH**と呼ぶ

# 単位の接頭辞について

- セント
  - cm(センチメートル)
  - 100分の1
- ミリ
  - mm(ミリメートル)
  - mg(ミリグラム)
  - 1000分の1(1/1000)
- マイクロ
  - $\mu\text{m}$ (マイクロメートル)
  - $\mu\text{g}$ (マイクログラム)
  - $\mu\text{秒}$ (マイクロ秒)
  - 100万分の1(1/1000000)
- %(パーセント)
  - パーとは割ることを意味している
  - セントで割る  
→百で割る
- ‰(パーミル)
  - 千で割る

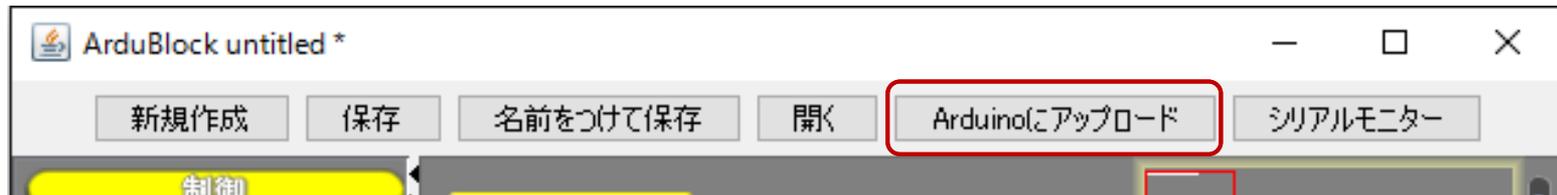


# コンパイル・ダウンロード



# ロボットにプログラムを書込む

[Arduinoにアップロード]ボタンを押します



「ボードへの書き込みが完了しました。」と表示されたら、正しくロボットにプログラムが入っています

ボードへの書き込みが完了しました。

最大28,672バイトのフラッシュメモリのうち、スケッチ  
最大2,560バイトのRAMのうち、グローバル変数が14

# 動作確認

---

自分の思ったとおりの動きをしましたか？

# デバッグ

自分の思いとは違う動きをした場合はプログラムを修正する→これを「**デバッグ**」という

Arduinoはプログラムを繰り返し実行する



# LEDを点滅するプログラム

- LEDを1秒ごとに、つけたり消したりするプログラム

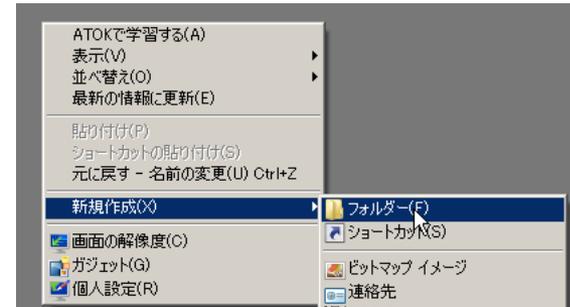


# プログラムの管理

# プログラムの管理

- 最初にプログラムの置いておく場所を決める
  - ここではデスクトップに”robot” というフォルダを作ることとします(名前は好きな名前でもOK)

- デスクトップ上でマウスを右クリックし  
[新規作成]-[フォルダー]を選択



- フォルダが出来たら名前を  
「robot」に変えます



# プログラムの保存1

- 最初にプログラムの保管する場所をするとき

ウィンドウタイトルが「ArduBlock untitled \*」(プログラムに名前が付けられていない)

[保存]ボタンを押す

robot  
フォルダを表示

名前をつけて[保存]ボタンを押す

保存が完了するとウィンドウタイトルが変わります

# プログラムの保存2

- 保存しているプログラムを修正し保存をするとき
  - プログラムの保存後、内容を少しでも修正すると、ウィンドウタイトルの末尾に”\*”マークが表示される



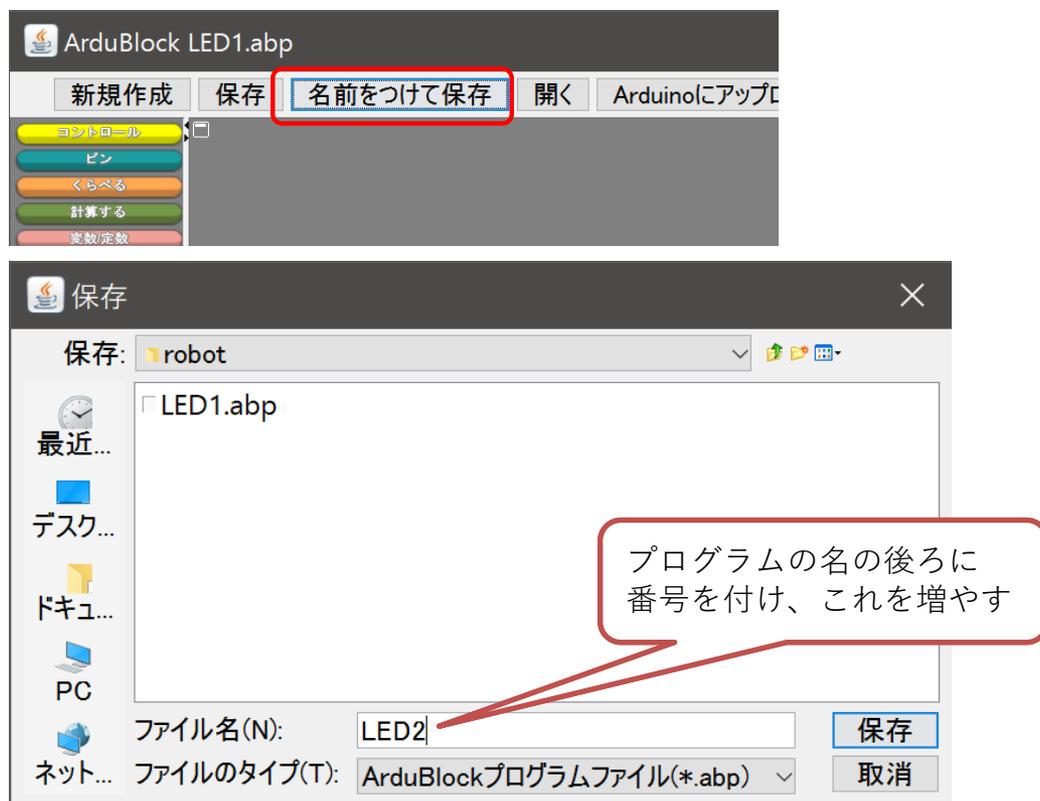
- ある程度プログラムを修正したら、念のためプログラムを保管する



[保存]ボタンを押すと、  
上書きで内容が保存される

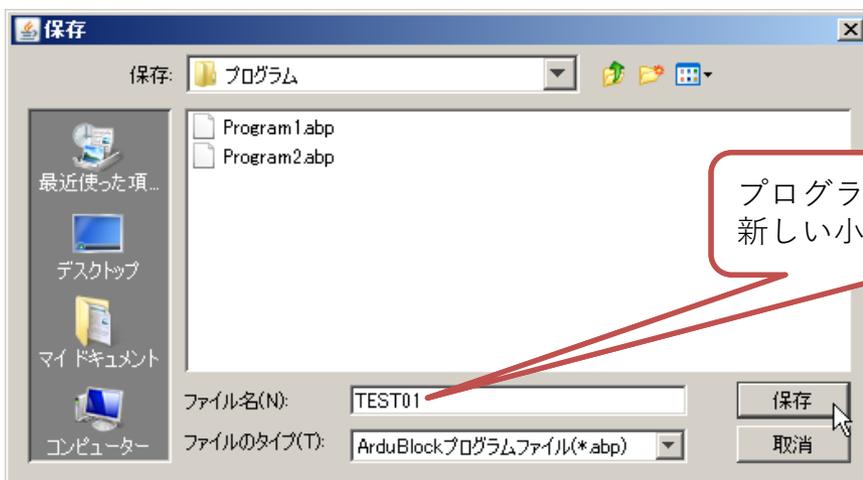
# プログラムの保存3

- しっかりと改造できたプログラムが完成したとき
  - プログラムに新しい番号を付け上書き保存する



# プログラムの保存4

- 今あるプログラムから、  
大きな改造をしようと思ったとき
  - 最初にプログラムに新しい名前を付け上書き保存する

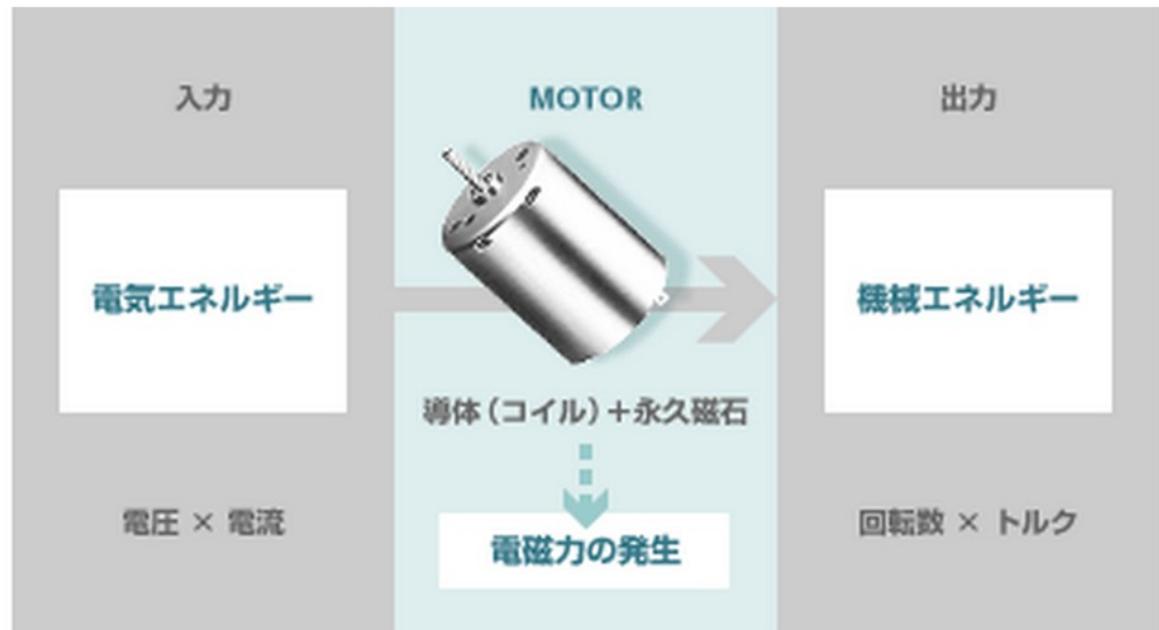


# プログラミング

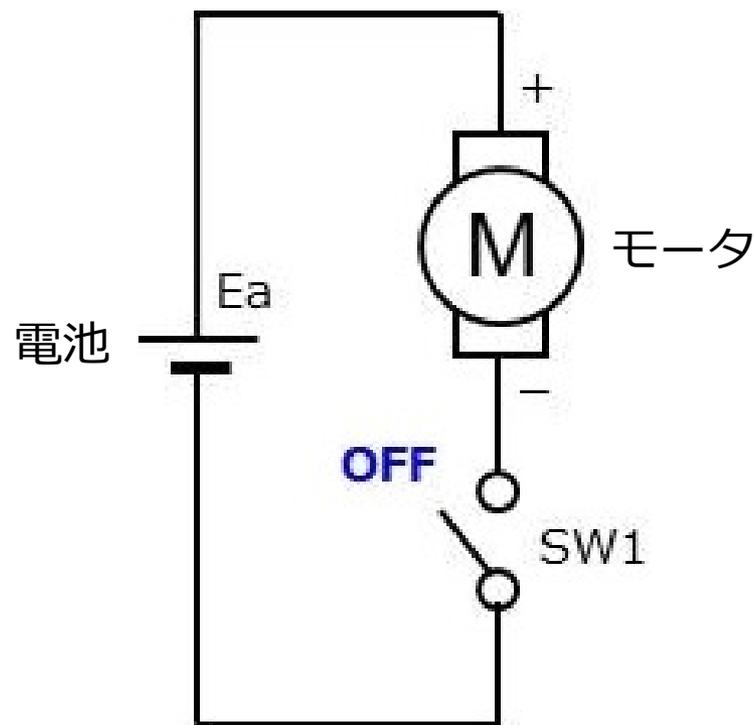
ロボットを走らせる

# モータ

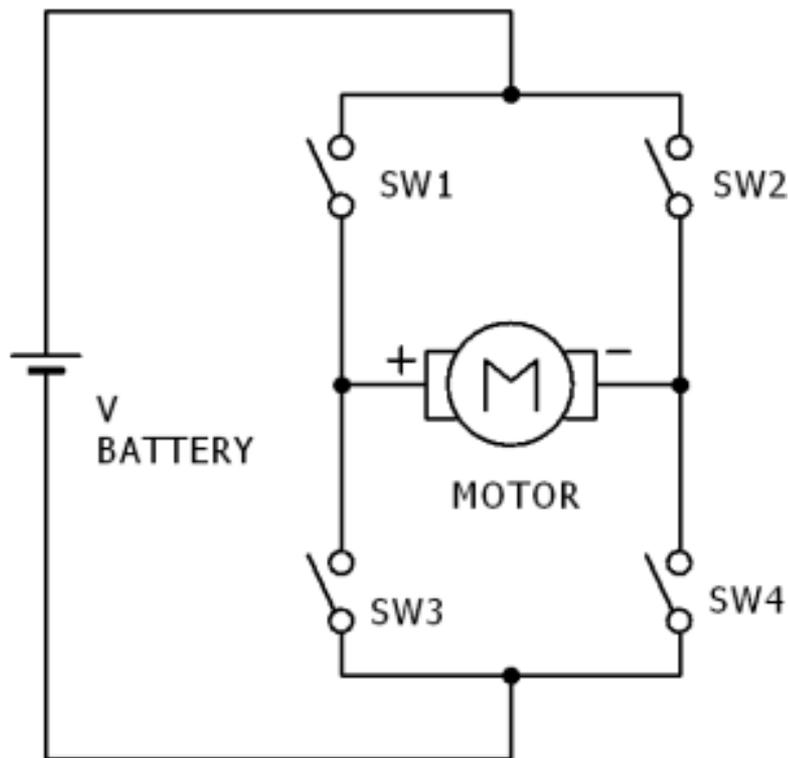
「モータ」とは、  
電気エネルギーを  
機械エネルギーに  
変換する装置です。



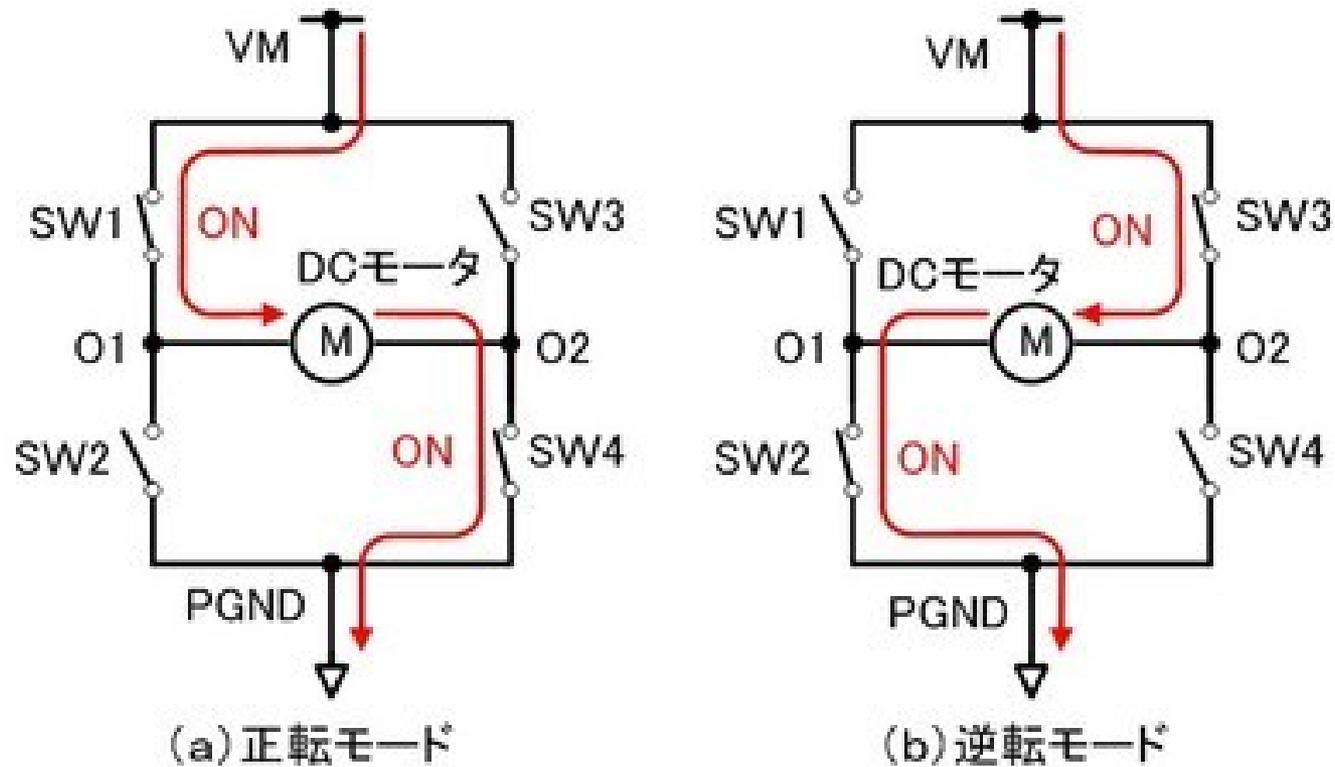
# モータの回転を制御するには



# モータの回転を制御するには

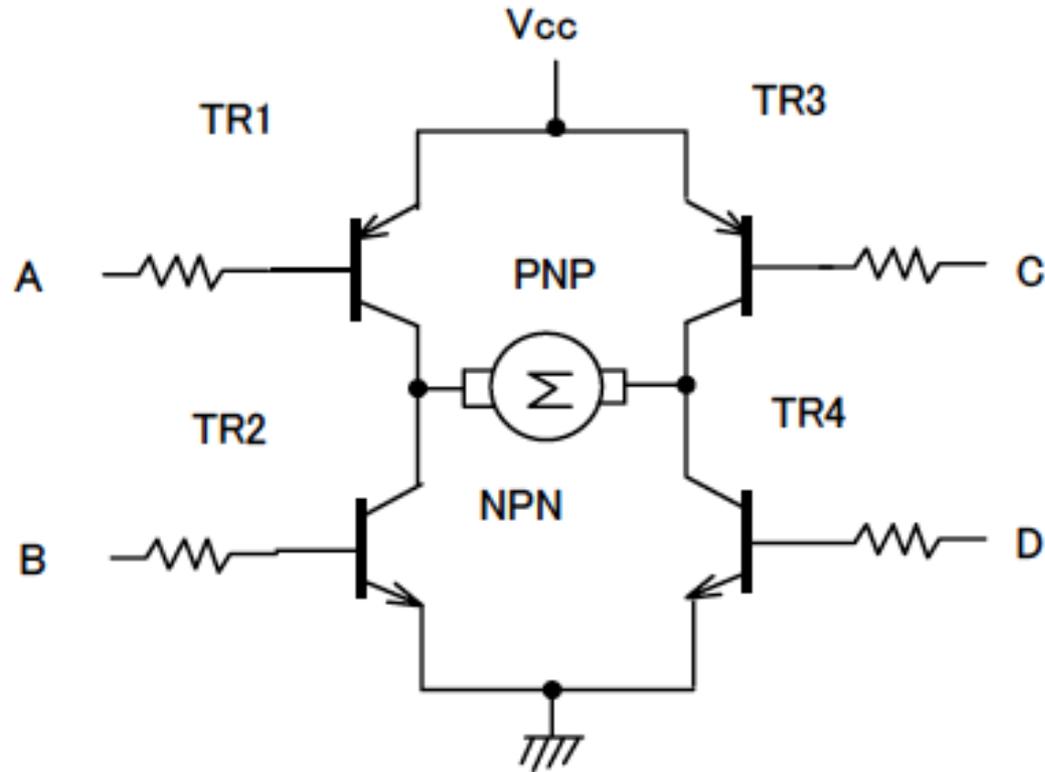


# モータの回転を制御するには

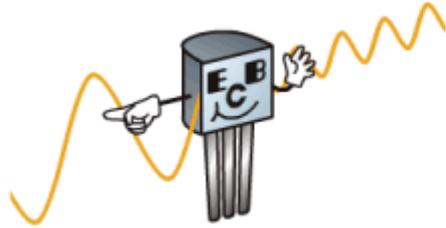


# Hブリッジ

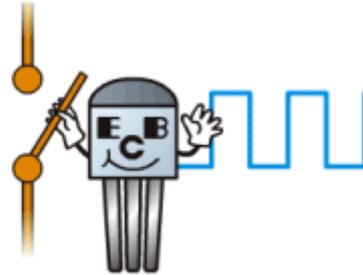
- モータの回転方向を制御する



# トランジスタとは



小さな信号を何倍にも大きくする



信号によって電気を流したり、  
止めたりする

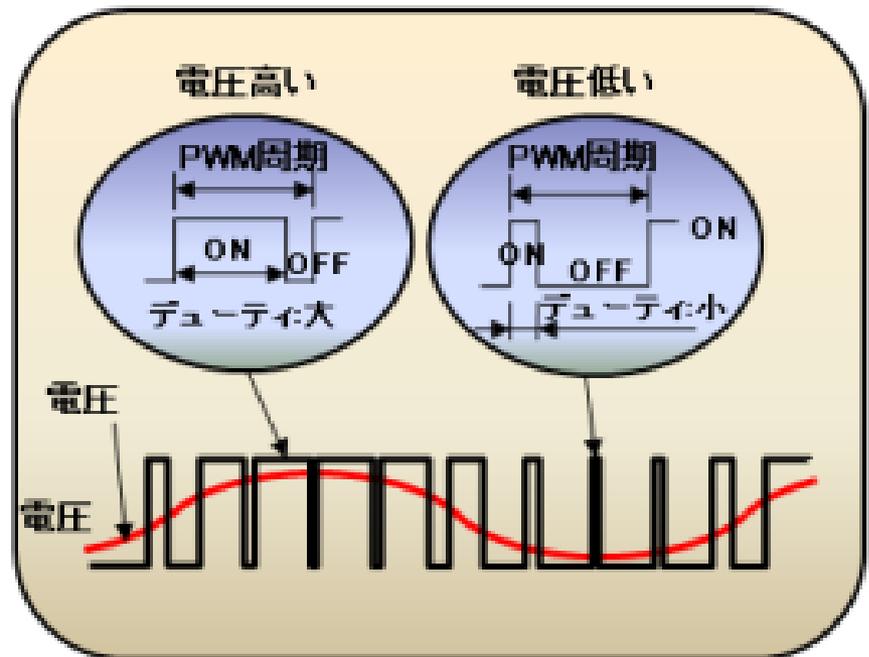
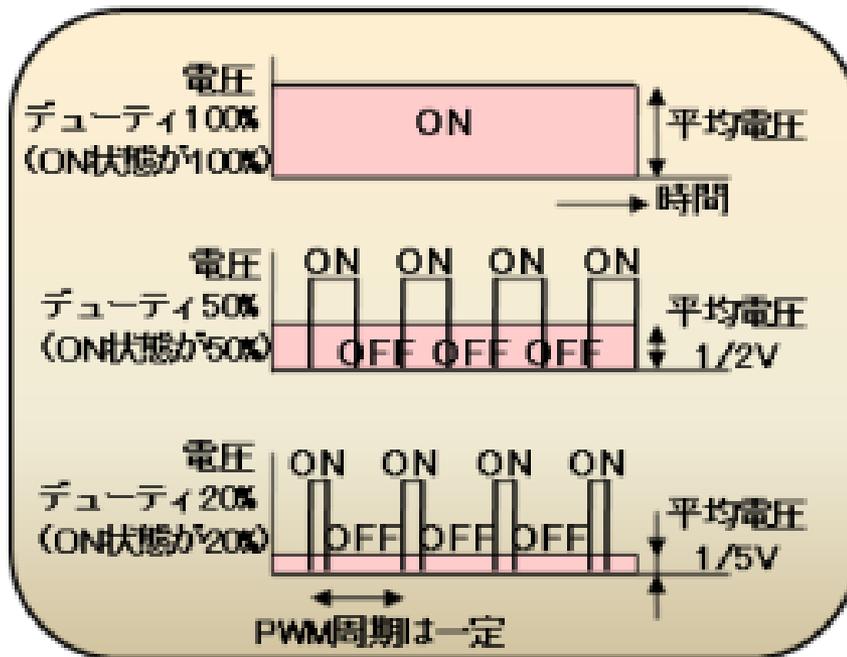
※「20世紀最大の発明」といわれている



IC(集積回路)には無数の  
トランジスタが入っています

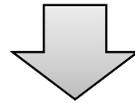
# PWM

## Pulse Width Modulation



# モータドライバ

モータを強く駆動するための電子回路  
コントローラの出力行では  
電力が足りずモータを強力に駆動できない

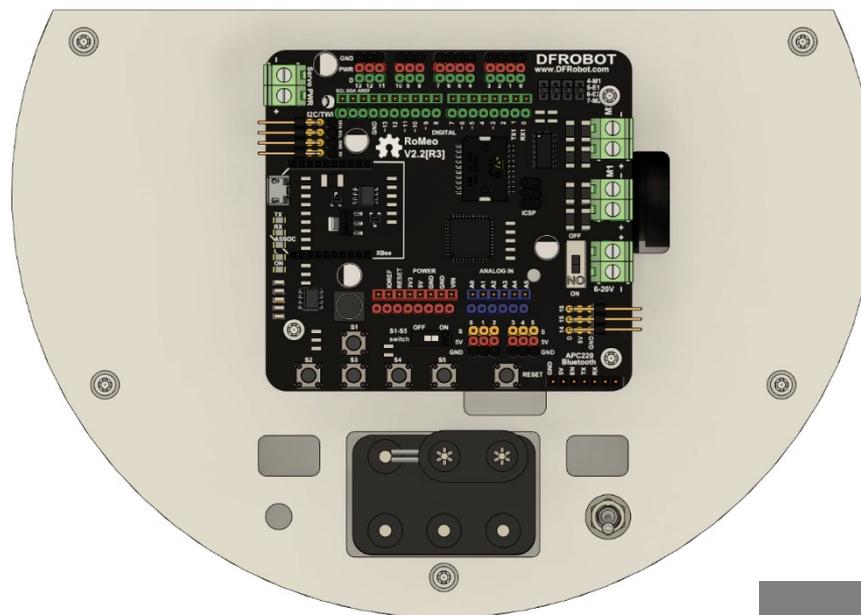


モータドライバを用意する必要あり



# ORJAロボ

左モータ  
(M2)

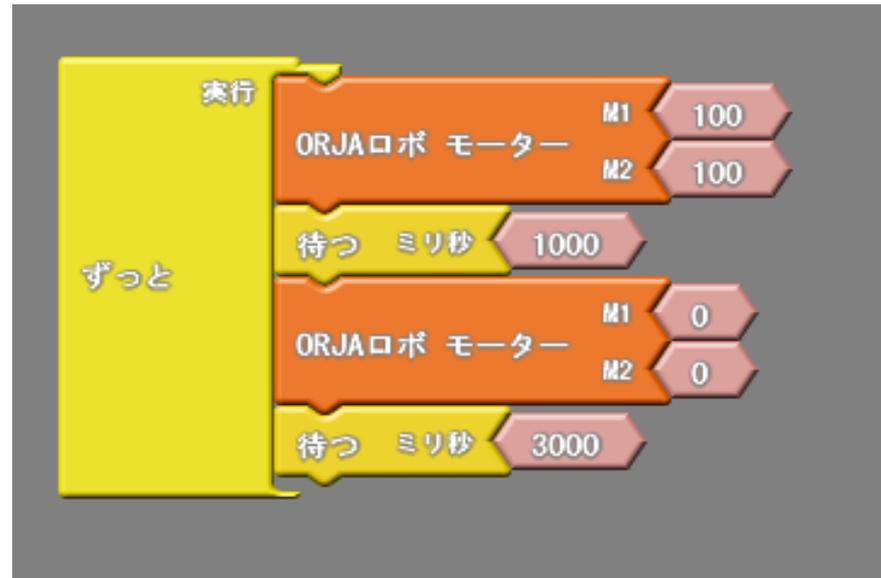
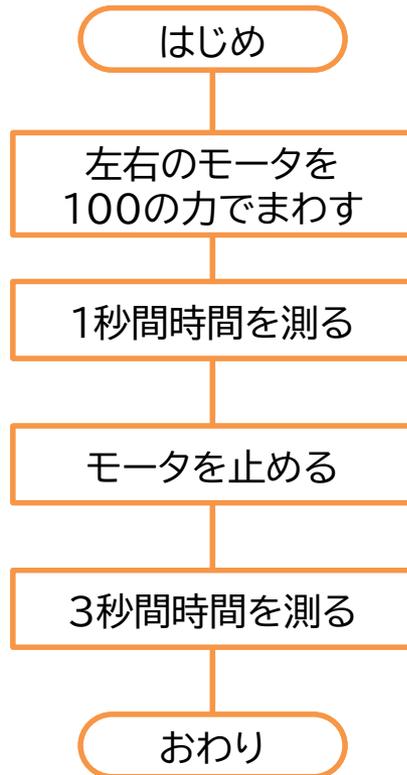


右モータ  
(M1)

ORJAロボ モーター	I11	50
	I12	50

# 前進させてみよう

- 1秒間前進して3秒止まるアルゴリズム例



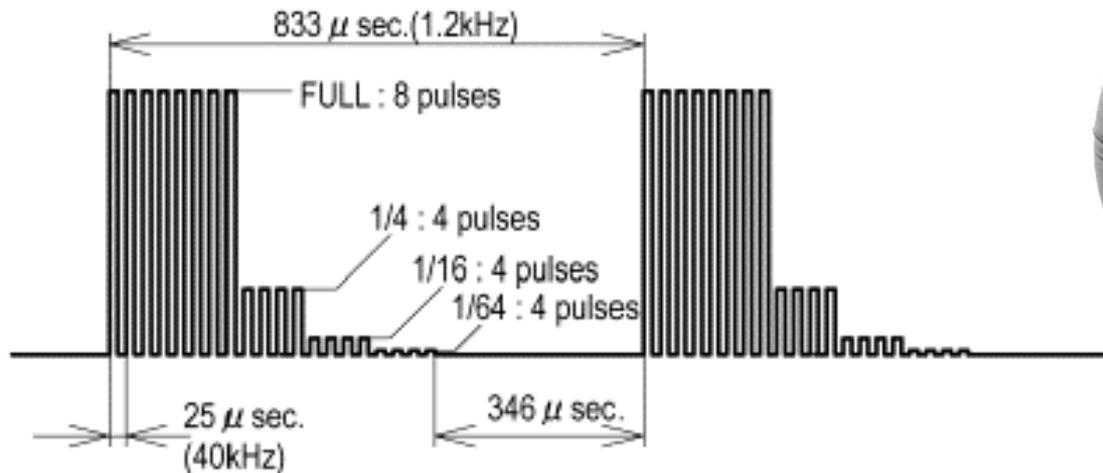
- 
- ここからは資料をみて問題を解いてみてください。  
(あせらなくても だいじょうぶです)
  - 質問がある人は手を挙げて下さい
  - 質問するときは、自分は何をどうしたいのか  
今どうなってしまうのか説明をして下さい
  - できた人は近くの先生に見せて下さい

# ボールセンサ

ボールを見つけたらLEDを光らす

# ボールとボールセンサの仕組み

- 家電製品のリモコンの赤外線発光素子が使われている
  - ボールが発する信号は40kHzの周波数で変調された赤外線(日本のリモコンは38kHz)
  - 赤外線LEDが波長940nmの赤外線を出している

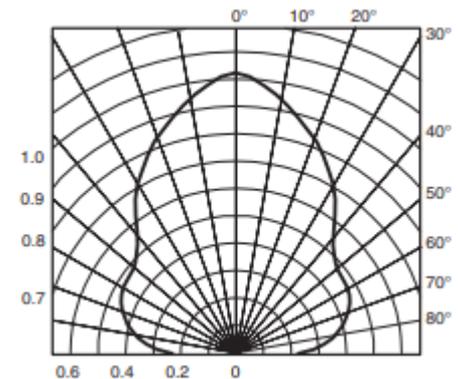
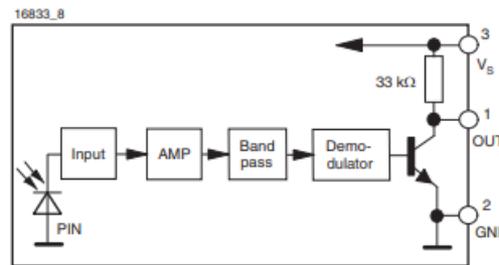


# ボールとボールセンサの仕組み

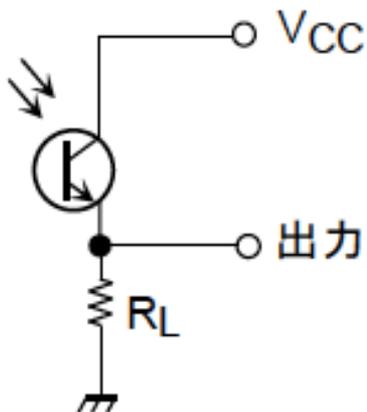
- リモコン受光素子をボールセンサとして利用
  - 皆さんのロボットではVishay社” TSSP58038”を利用



電子部品を購入するときは  
あわせてデータシートも入手する



# センサの基本



外部の情報により抵抗値が変化するものをセンサとして利用する

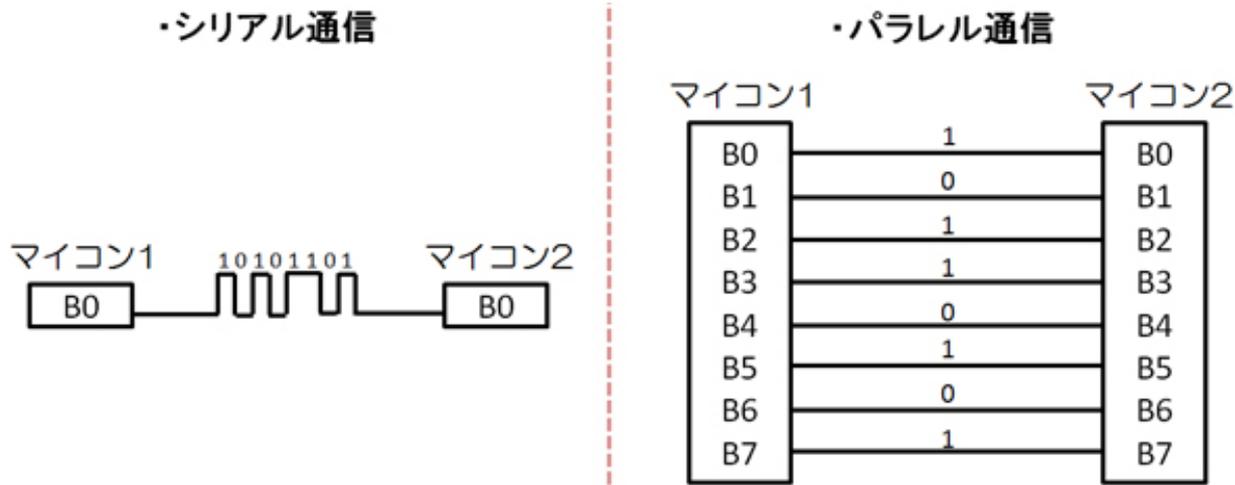
外部環境の変化により抵抗値が変化することで  
出力される電圧が変化する

# シリアル通信と シリアルモニタ/シリアルプロッタ

信号の値を見る

# シリアル通信

- デジタル信号を1bitずつやり取りすること
- デジタルでの最も基本的なデータ取り扱い
  - ごく簡単な装置でほとんどのデジタル機器についている



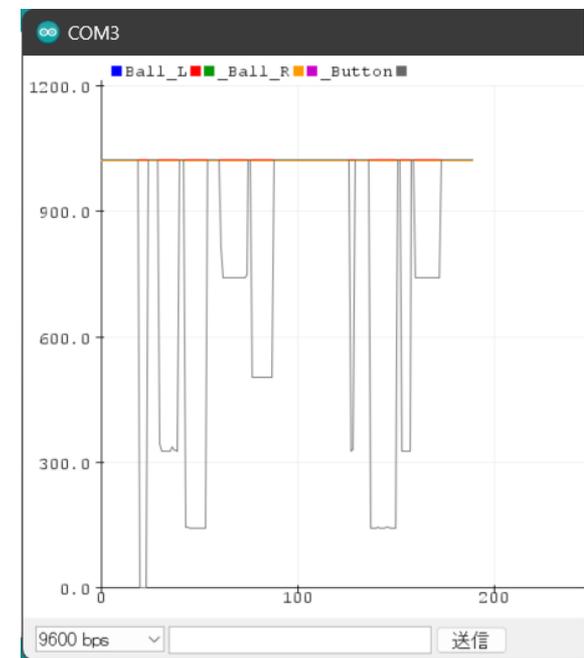
# シリアルモニタ/シリアルプロッタ

- 1bitずつ送られてくるデータを表示する
- 人間がロボットの取り込んだ情報を知ることが出来る

シリアルモニタ



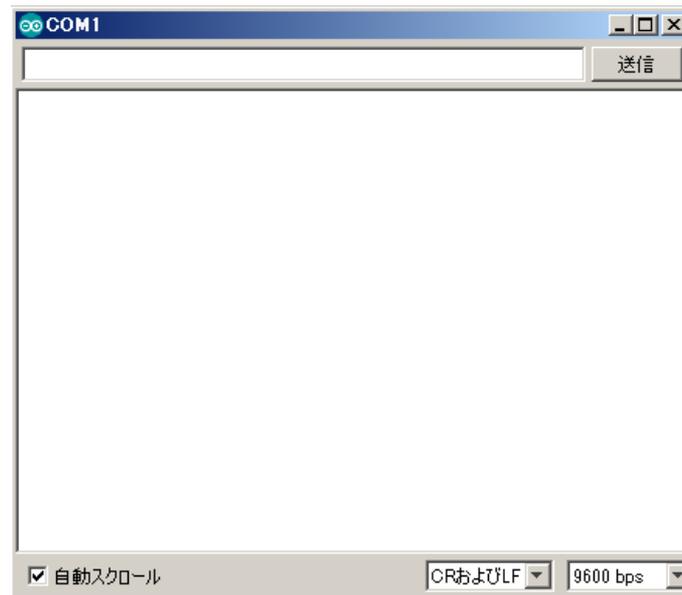
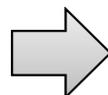
シリアルプロッタ



# センサの値を見てみよう

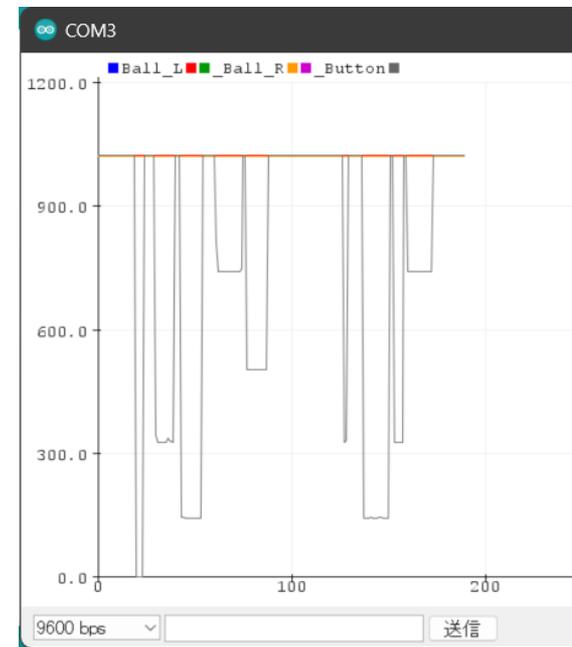
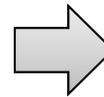
# センサの値を読む1

- センサの値を見るためにArduinoのシリアルモニタを利用します
  - Ardublockでセンサの値をパソコンに送る
  - ArduinoIDEで送られてくる値を表示する



# センサの値を読む2

- センサの値を見るためにArduinoのシリアルプロッタを利用します
  - Ardublockでセンサの値をパソコンに送る
  - ArduinoIDEで送られてくる値を表示する

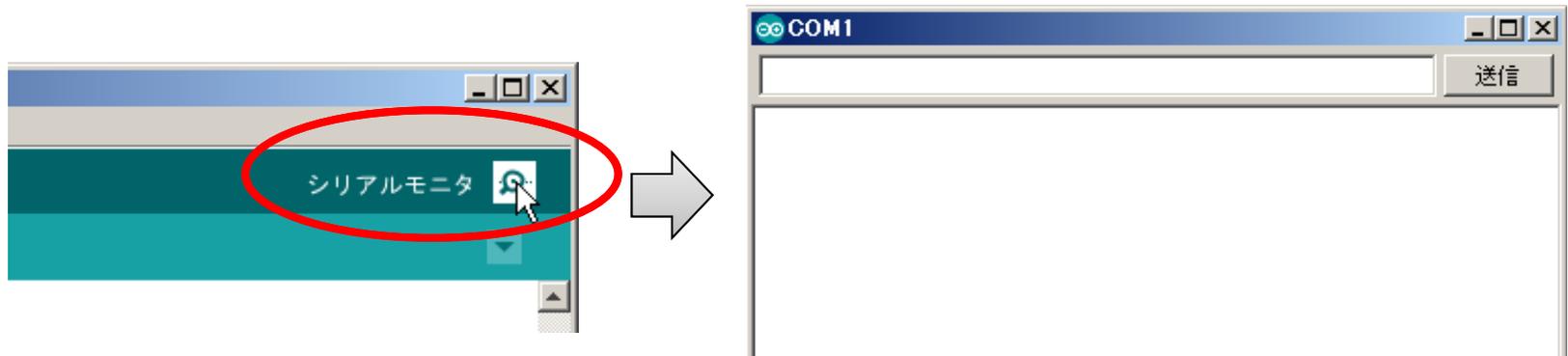


# Ardublockのシリアル通信

下図のようにブロックを組んでロボットへアップロードして下さい

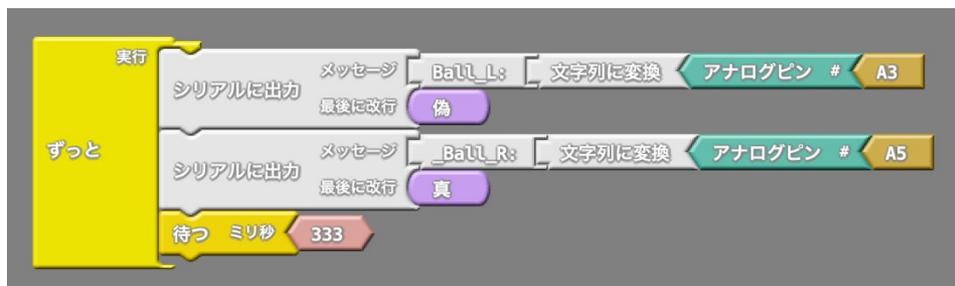


アップロードできたらシリアルモニタを見てみましょう

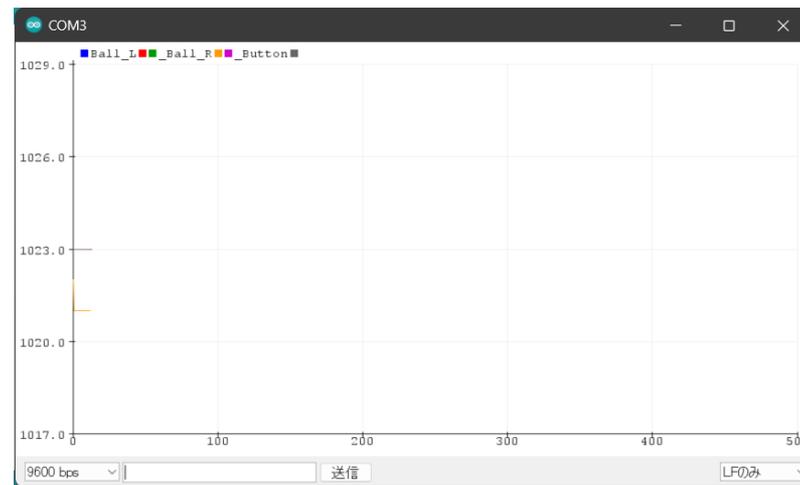
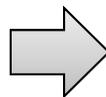


# Ardublockのシリアル通信

下図のようにブロックを組んでロボットへアップロードして下さい

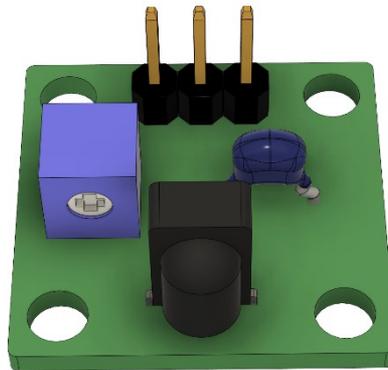


アップロードできたらシリアルプロッタを見てみましょう



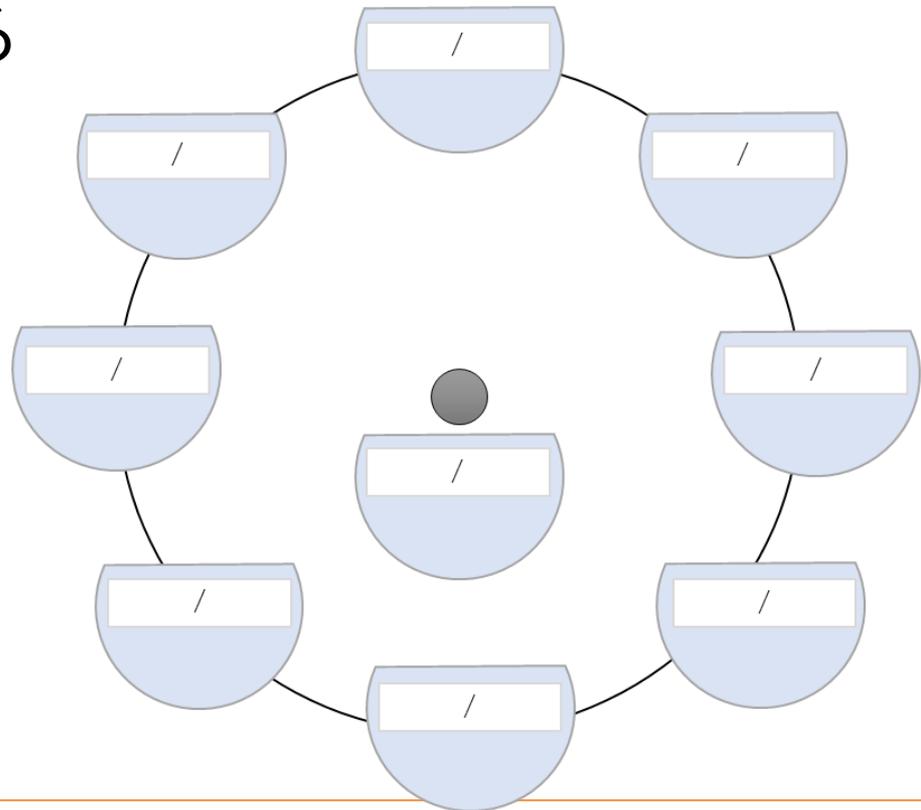
# 準備1 ボールセンサの調整

1. シリアルモニタを表示
2. どちらかのボールセンサについている半固定抵抗を回してみ、おおよそ中央に設定する
3. もう一方のボールセンサの値を見ながら、先程の値とほぼ同じ値になるように、半固定抵抗を左右に少しずつつまわし調整する



# 準備2 ボールセンサ値の確認

1. シリアルモニタを表示
2. ロボットの位置によって左右のセンサ値がどうなるか記録する



# センサの値 よくある質問1

---

「センサの信号を見ると、何もしていなくても、  
増えたり減ったりして安定していない」



信号は安定しない それが当たり前

# センサの値 よくある質問2

---

「センサの数値は何を表している？  
距離？単位は何？」



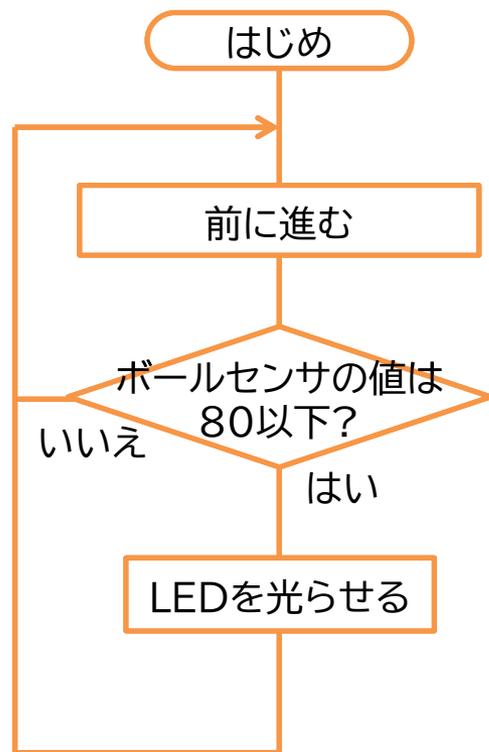
電圧の大小を10bit・1024段階で表現している  
単位は無し(具体的な意味はない)

# 練習問題1

---

ボールに向かってロボットを前進させ、右ボールセンサの値が80より小さくなったらLEDを光らせましょう

# ボールが近いとLEDを光らせる



## 条件分岐を使う

もし〇〇なら△△する



# 練習問題2

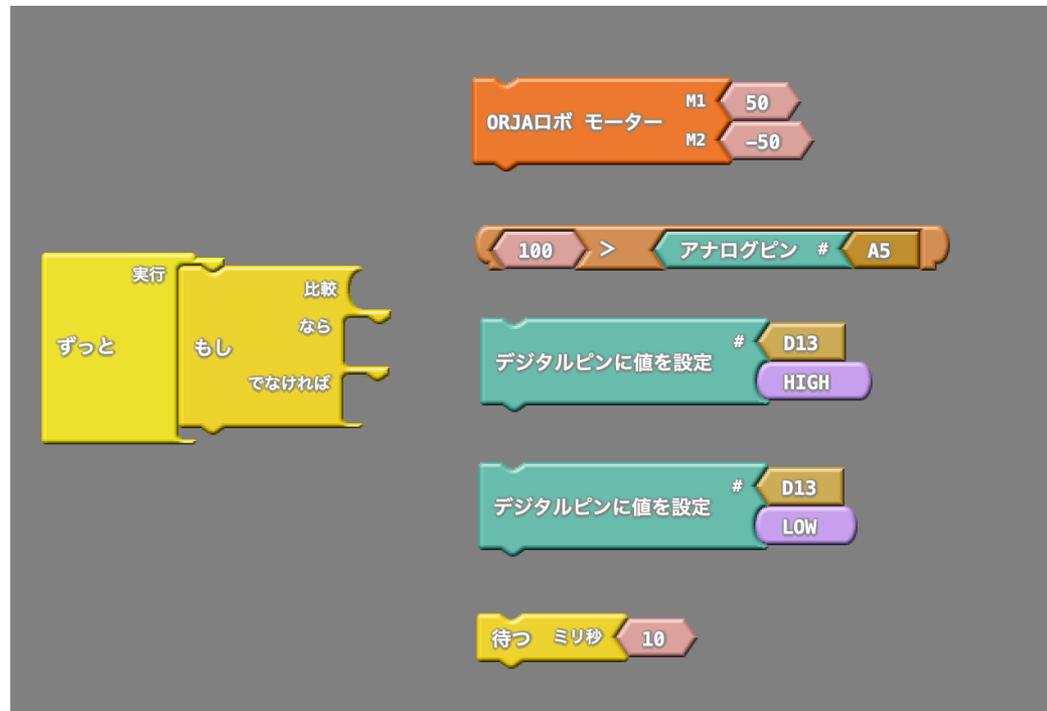
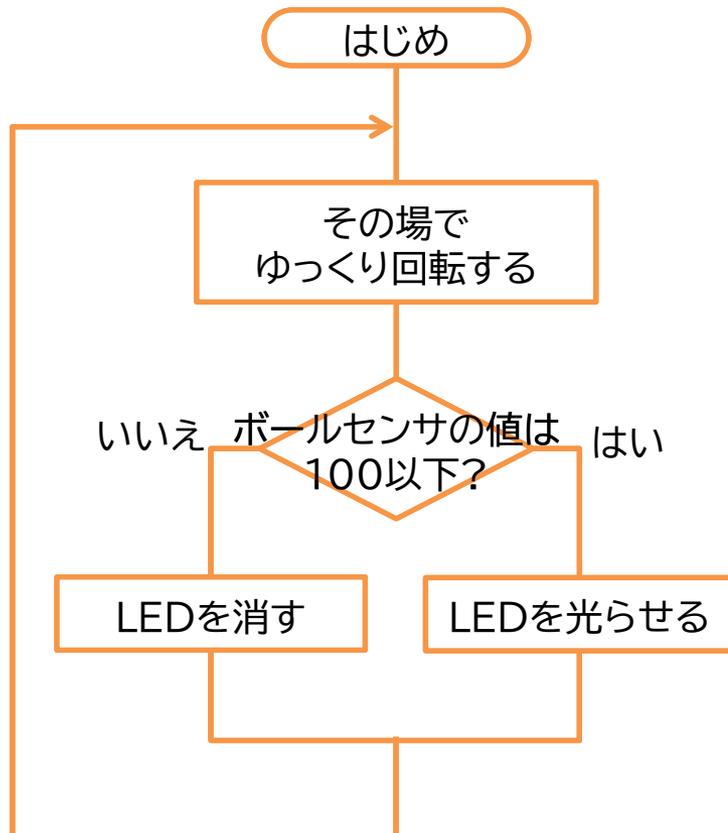
---

ロボットをその場で回転させて、右ボールセンサの値が**100**より小さいときはLEDを光らせ、**100**より大きいときはLEDを消しましょう

# ボールが近いとLEDを光らせる

## 条件分岐を使う

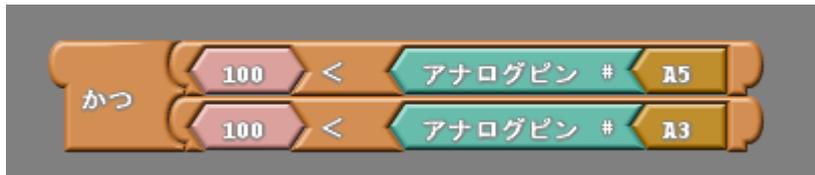
もし〇〇なら△△する  
そうでないときは××する



# 練習問題3

ロボットをその場で回転させて、左右のボールセンサの値が両方とも**100**より小さいときはLEDを光らせ、そうで無い時はLEDを消しましょう

ヒント:AND条件(かつ条件)を使います



# 練習問題4

ロボットをその場で回転させて、左右のボールセンサの値の差が**30**以下の場合LEDを光らせましょう

ヒント:計算式を使います



# 練習問題5

---

ロボットをその場で回転させて、左右のボールセンサの値の差が**30**以下の場合、ロボットを前進させましょう

# 練習問題6

---

左右のボールセンサの値の差が100以下のとき、ロボットを前進させましょう

そうで無いときはボールのある方向にロボットを回転させましょう

# 次回の準備

- 電池単三型6本×数セット

- 乾電池(一次電池)でも充電式電池(二次電池)でも可

- 電池がなくなったとき、六本のうち一本だけ交換することは止めましょう。節約のつもりがかえってたくさんお金が掛かります

- ドライバー

- ロボットのネジを確認して各自判断

- 今日の配付物 + パソコン

- 他、自分で必要と思ったもの

ただし  
リチウムイオン  
電池は禁止