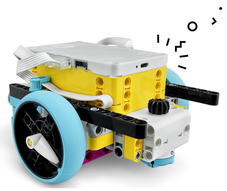
# ■ LEGO Mindstormsについて

# RIS, NXT, EV3, SPIKE

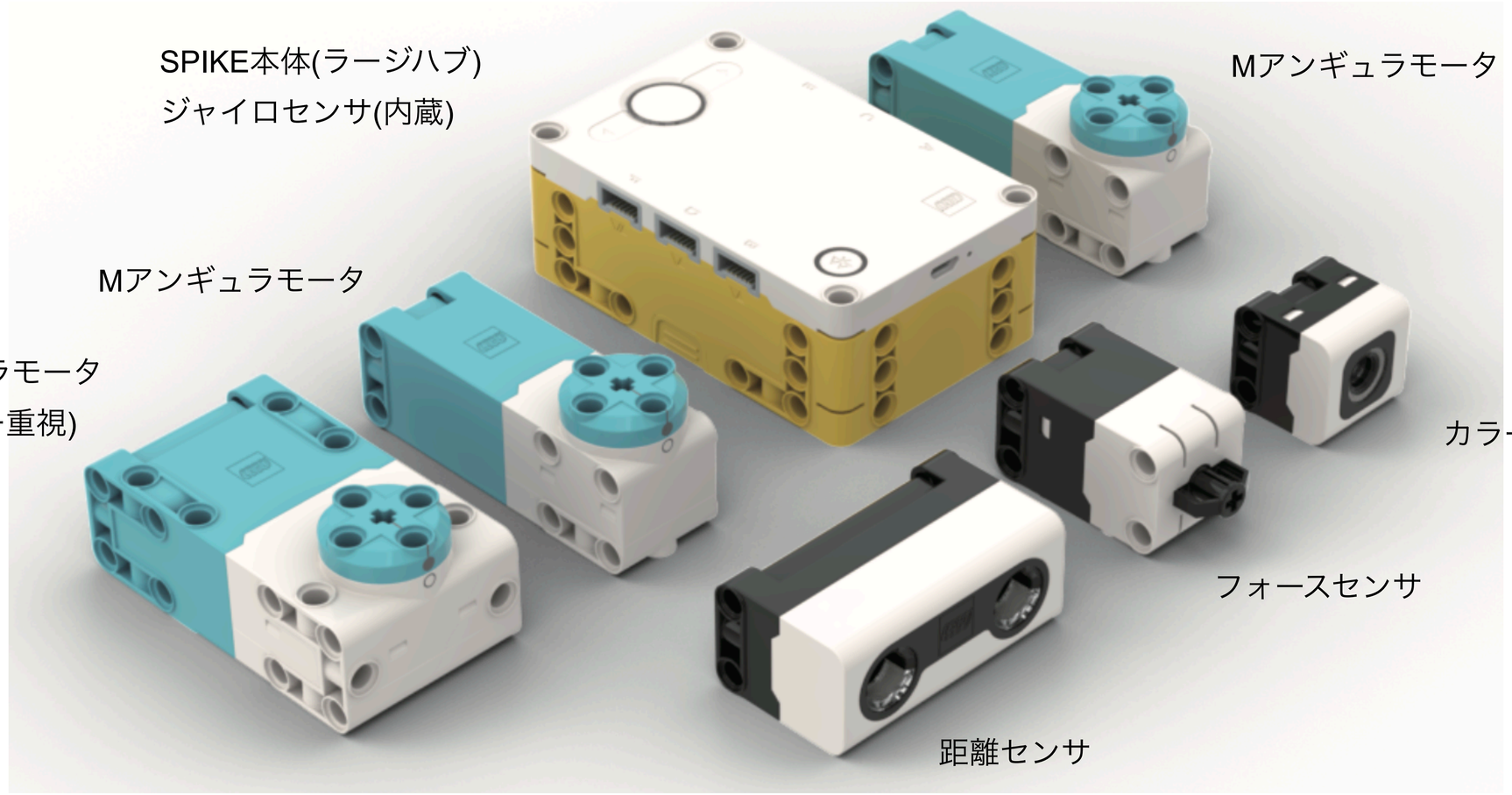


| | RIS | NXT | EV3 | **SPIKE** |
|---|---|---|---|---|
| 発売時期 | 1998年 | 2006年 | 2013年 | 2020年 |
| CPU | H8（8 bit） | ARM7（32bit） | ARM9（32bit） | ARM M4（32bit） |
| クロック周波数 | 16MHz | 48MHz | 300MHz | 100MHz |
| RAM | 32KB | 64KB | 64MB | 320KB |
| フラッシュメモリ | なし | 256KB | 16MB | 32MB |
| 転送方法 | 赤外線通信 | USB/Bluetooth | U/B + WiFi | USB/Bluetooth |
| ポート数 | 入力:3 出力:3 | 入力:4 出力:3 | 入力:4 出力:4 | 入出力： 6 |
| 駆動 | 電池 | 電池／バッテリー | 電池／バッテリー | バッテリー |

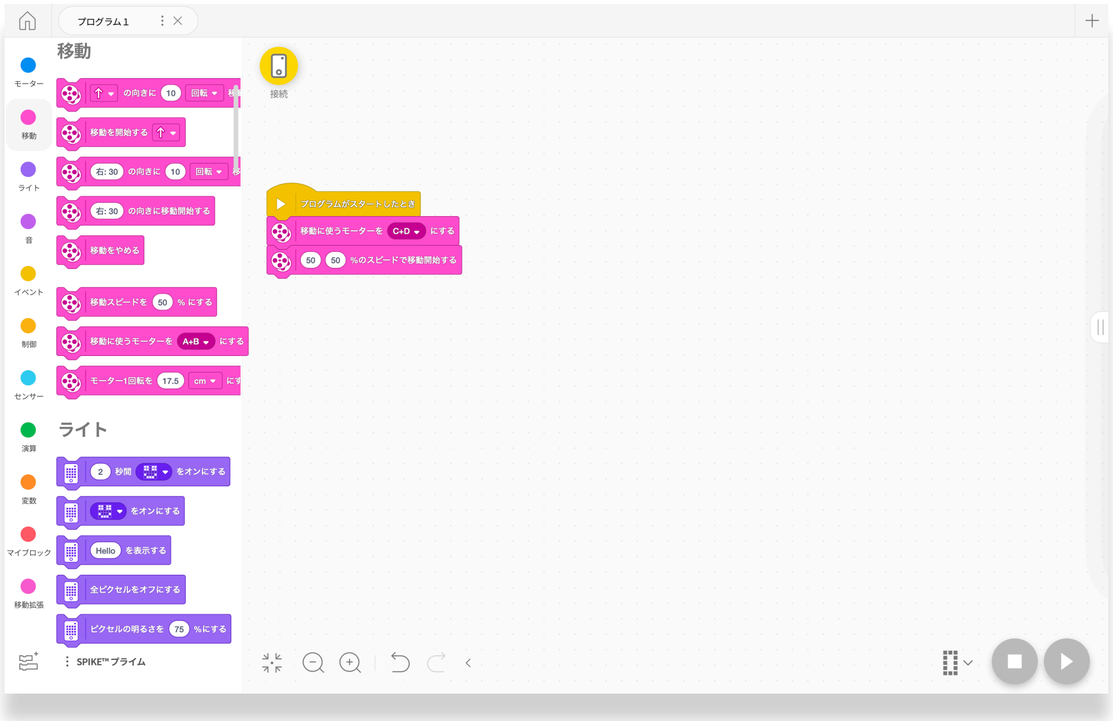
# LEGOロボットSPIKEの構成

- 入力：距離センサ、カラーセンサ、フォースセンサ、ジャイロセンサ(ハブ内)
- 出力：Lアンギュラモータ、Mアンギュラモータ

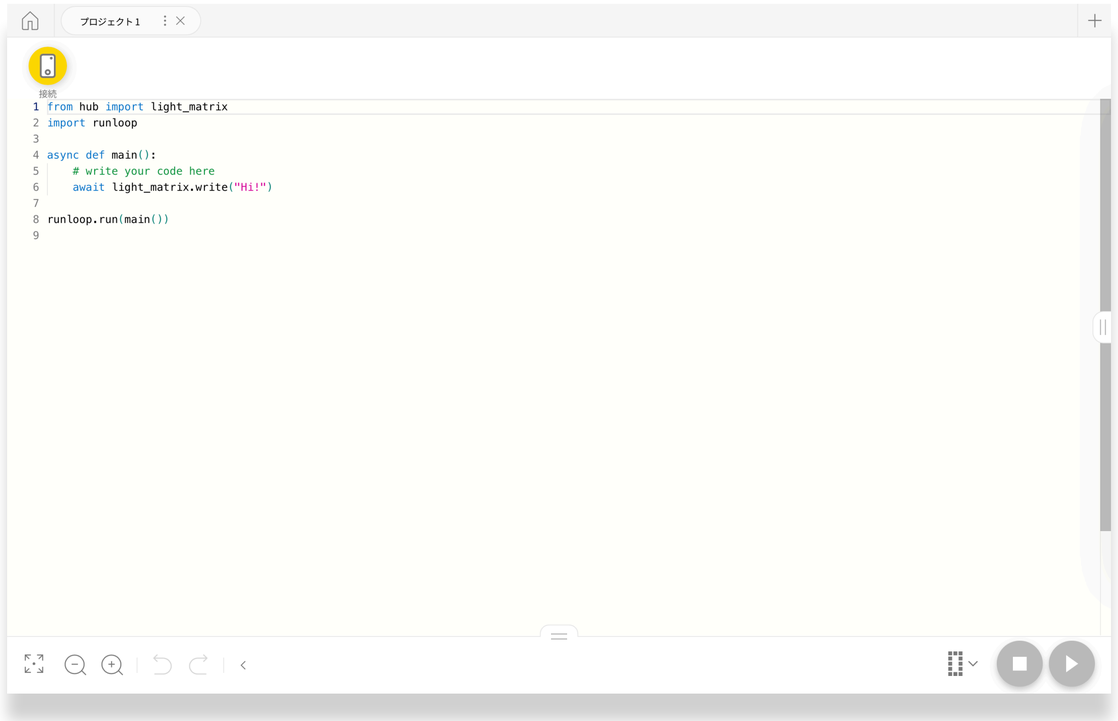


# プログラミング環境：SPIKE アプリ

- GUIとCUIの開発環境




ワードブロック(SPIKE-WB)




Python言語

## ■ ロボットの組み立て

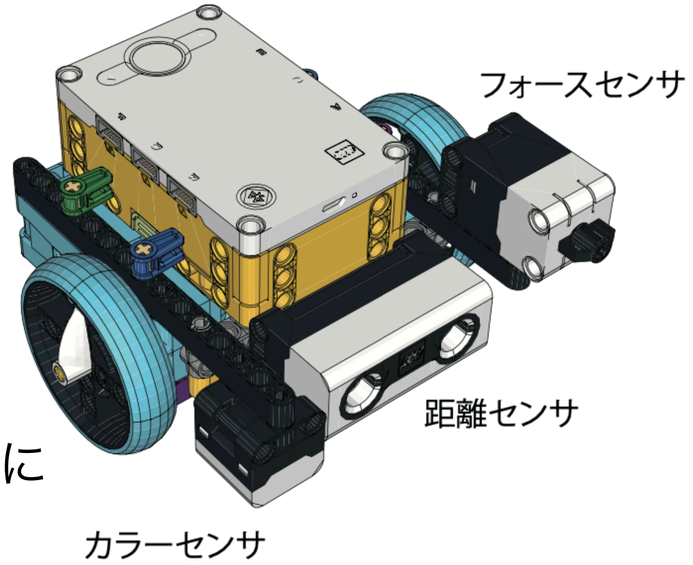
# ロボットの組み立て

SPIKE App ⇒ 組み立てガイド ⇒ **ドライビングベース３＋センサ**


組み立てガイドを参考にドライビングベースとセンサを組み立てる

① **ベース・ロボット：手順 1～26**
② **カラーセンサ：手順 1～4**
③ **フォースセンサ**※
④ **距離センサ**※

を組み立てよう！




※ フォースセンサと距離センサはセンサ裏に

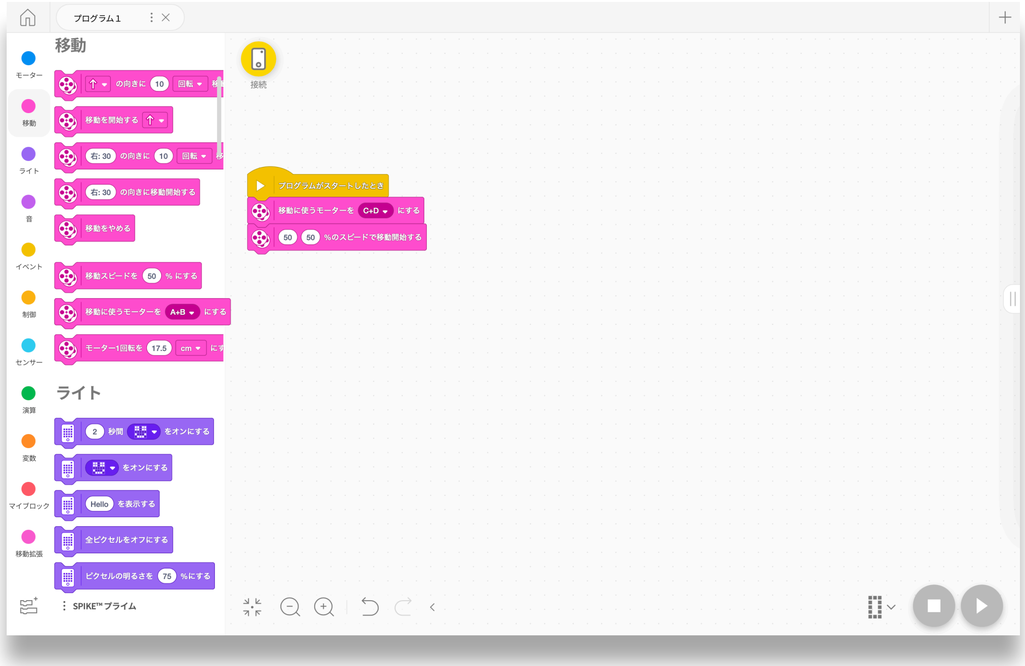
をはめてイラストのように取り付ける

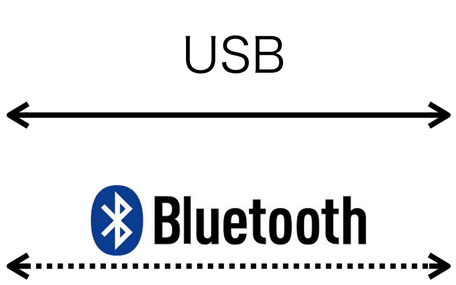
フォースセンサは**ポートA**に
距離センサは**ポートE**に接続

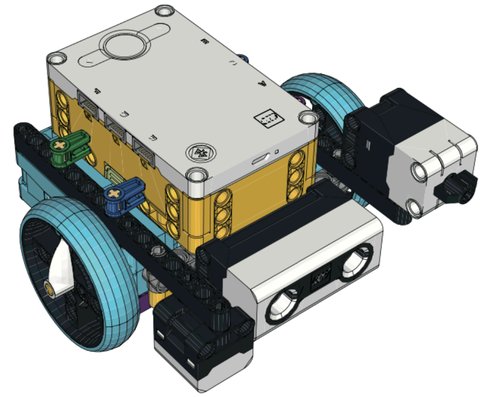
## ■ ロボットを動かすには

# プログラム実行までの流れ

- PC上でソフトウェア(SPIKE App)を用いてプログラムを作成
- USB/Bluetoothでロボットへダウンロード
- ロボット上でプログラムを実行

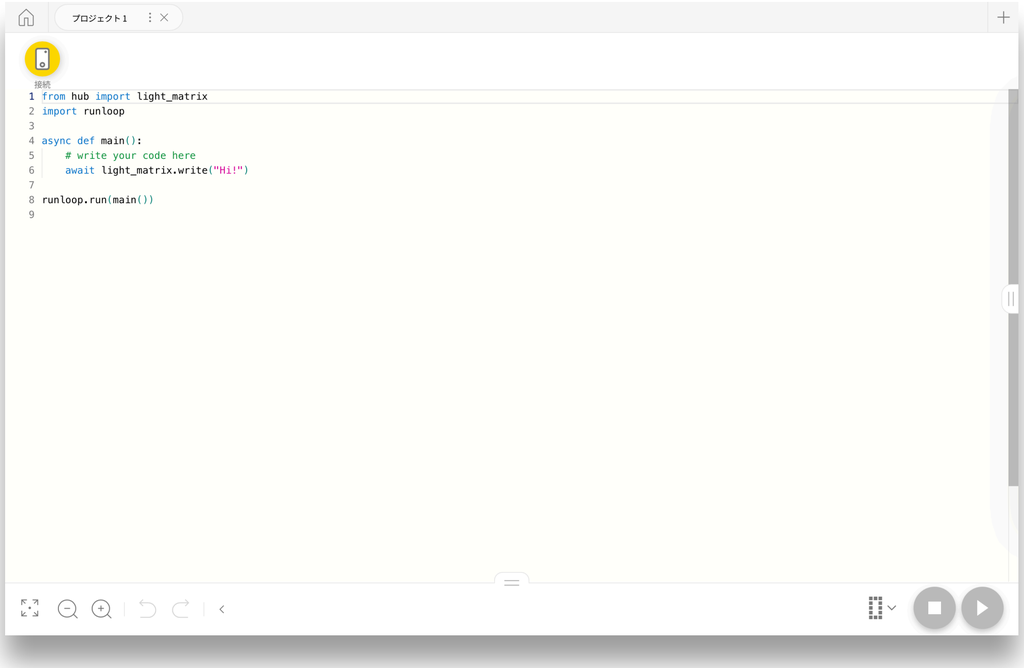


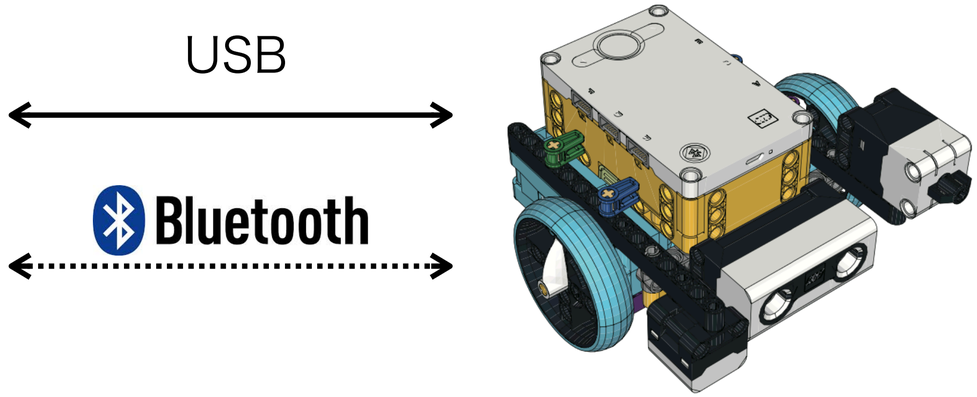



# プログラム実行までの流れ

- PC上でソフトウェア(SPIKE App)を用いてプログラムを作成
- USB/Bluetoothでロボットへダウンロード
- ロボット上でプログラムを実行



# SPIKEアプリ：ワードブロック

A. SPIKEアプリを起動

B. 新しいプロジェクトの作成

– 新しいプロジェクト⇒プログラムの種類を選択 ⇒ 作成する

# SPIKEアプリ：Python言語

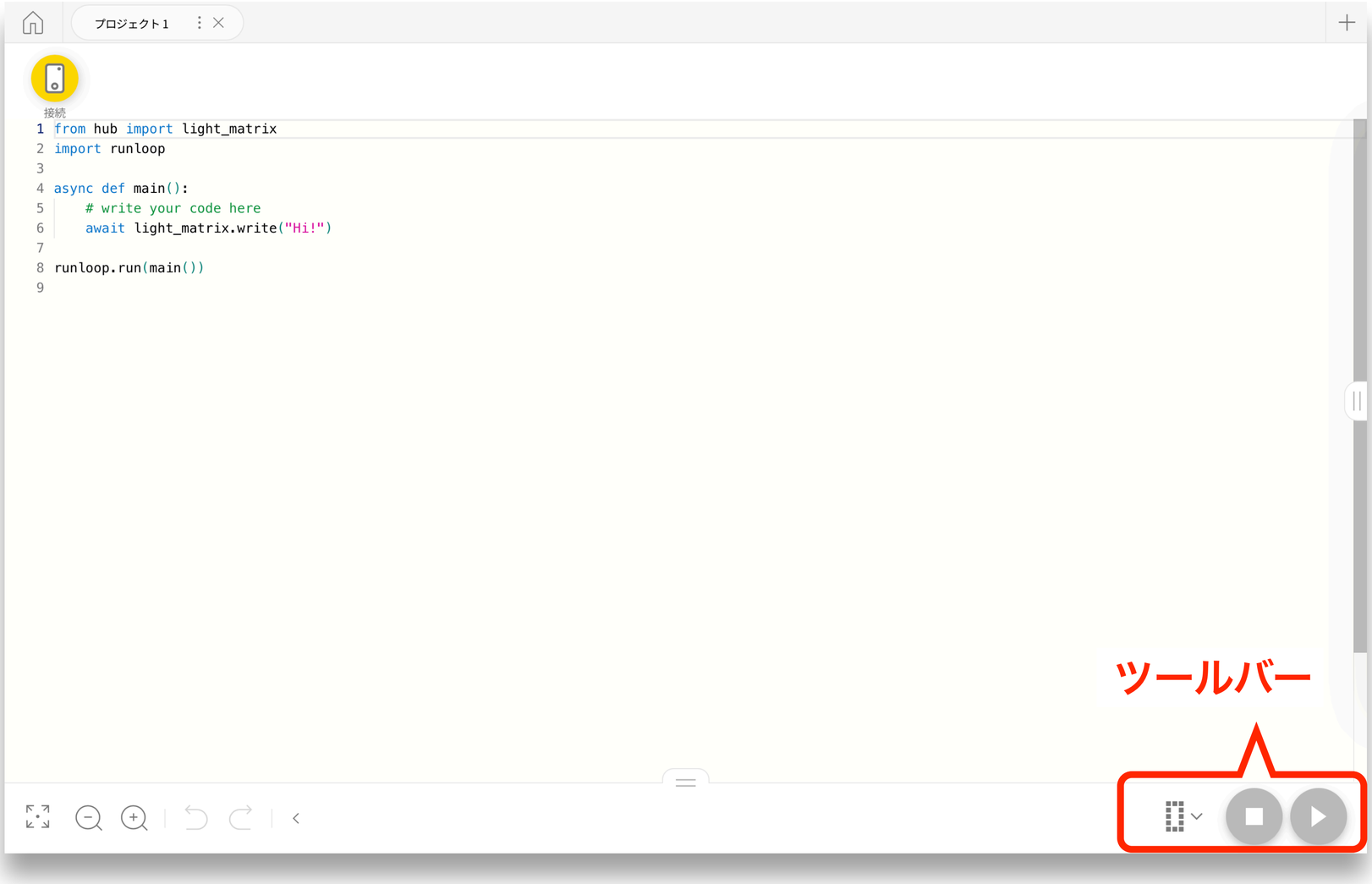
A. SPIKEアプリを起動

B. 新しいプロジェクトの作成

– 新しいプロジェクト⇒プログラムの種類を選択 ⇒ 作成する

# SPIKEアプリの画面構成

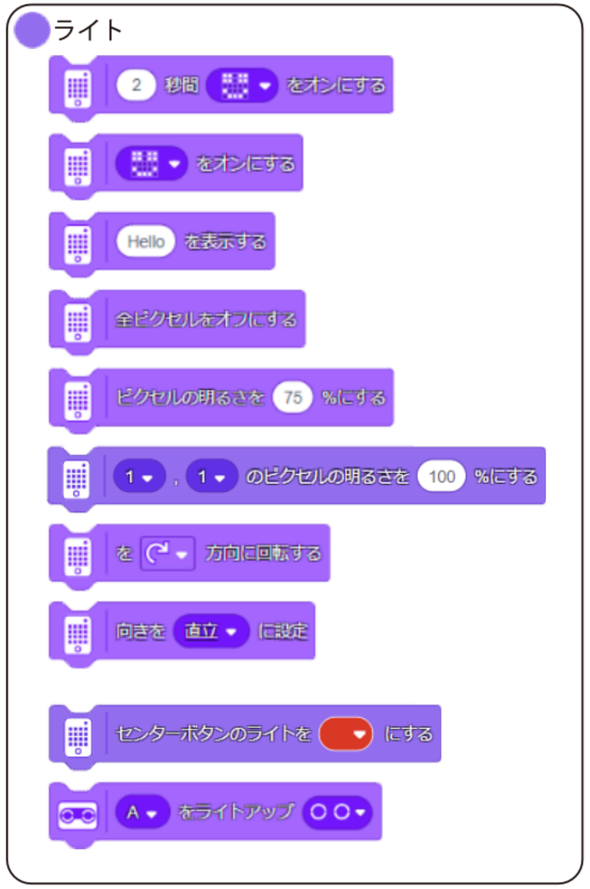
# SPIKEアプリの画面構成

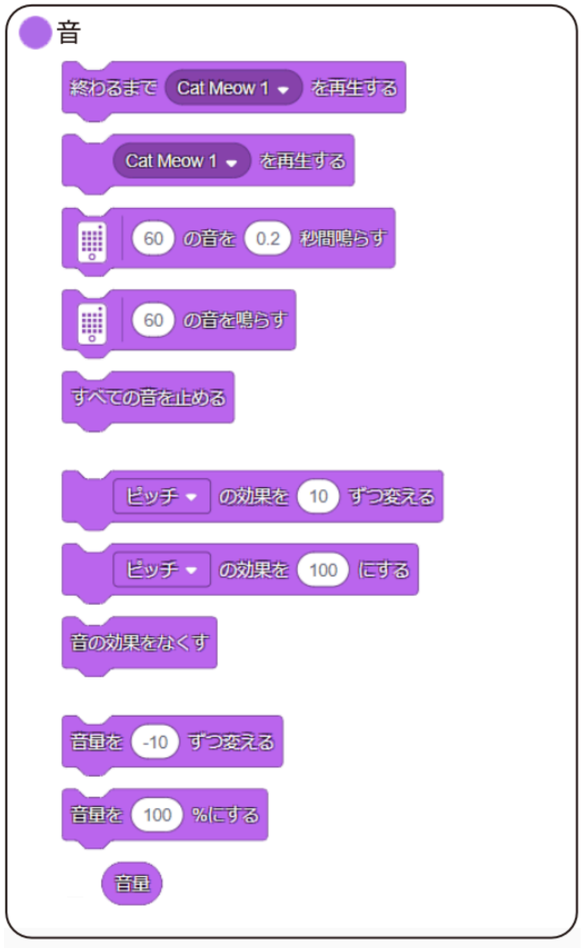


# 感じる・判断する・**動く**

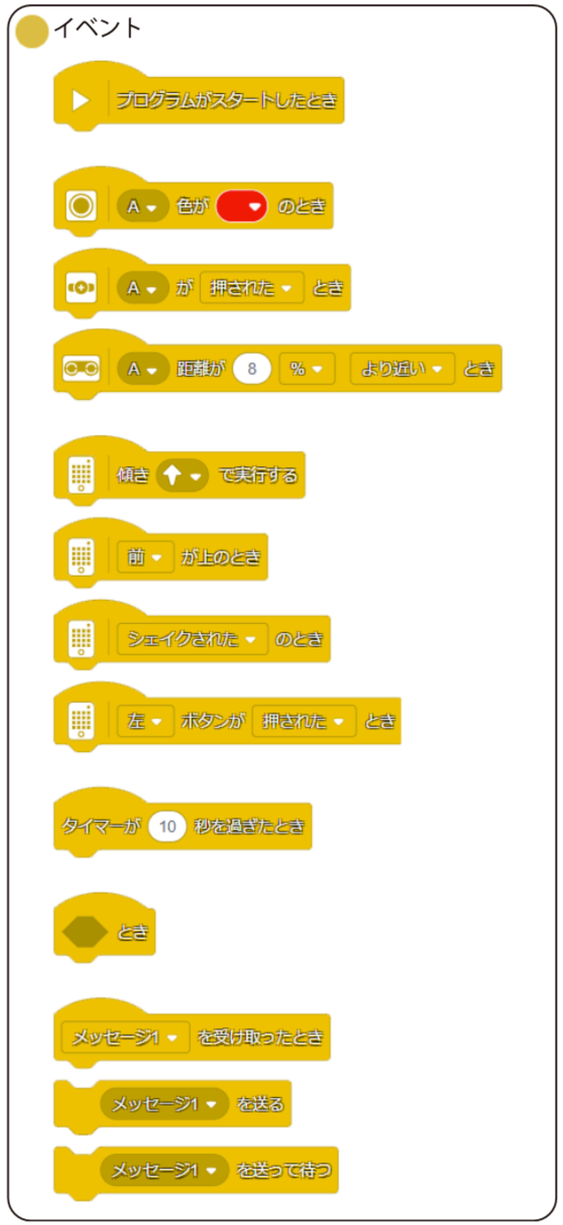

モーター
A を の方向に 1 回転 回す
A 最短経路で 位置 0 に行く
A を 方向に回す
A のモーターを停止する
A のスピードを 75 %にする
A 位置
A スピード

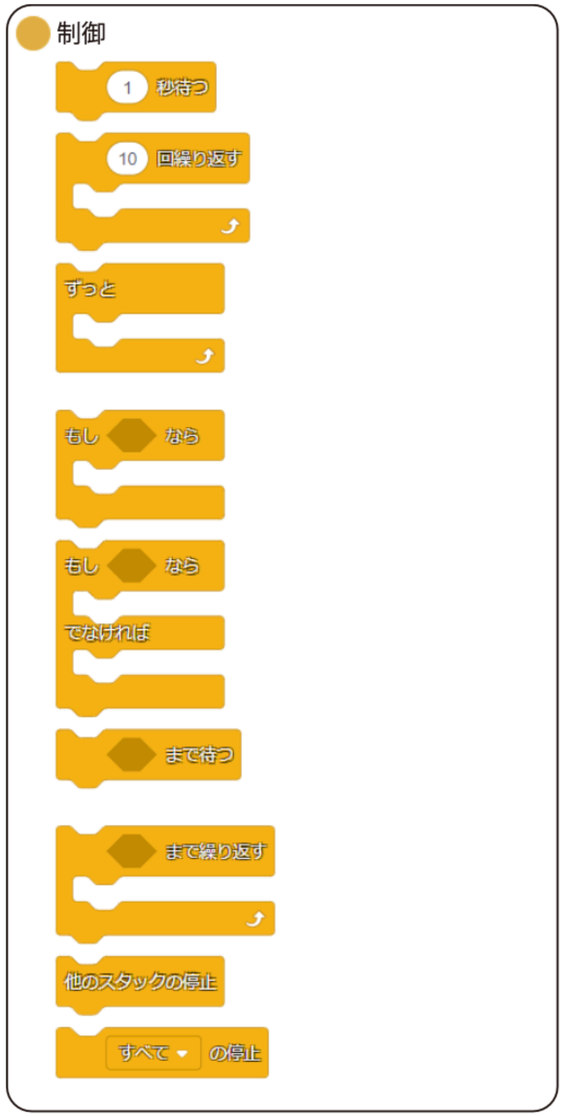

移動
の向きに 10 回転 移動する
移動を開始する
右: 30 の向きに 10 回転 移動する
右: 30 の向きに移動開始する
移動をやめる
移動スピードを 50 % にする
移動に使うモーターを A+B にする
モーター1回転を 17.5 cm にする

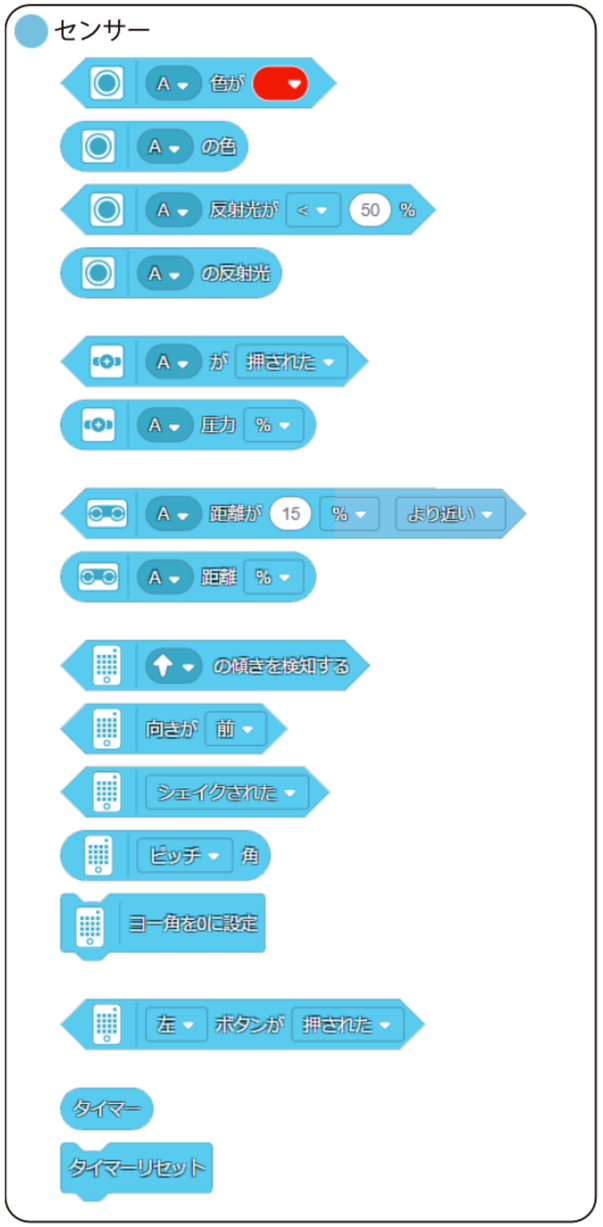

ライト
2 秒間 をオンにする
をオンにする
Hello を表示する
全ピクセルをオフにする
ピクセルの明るさを 75 %にする
1 1 のピクセルの明るさを 100 %にする
を 方向に回転する
向きを 直立 に設定
センターボタンのライトを にする
A をライトアップ


音
終わるまで Cat Meow 1 を再生する
Cat Meow 1 を再生する
60 の音を 0.2 秒間鳴らす
60 の音を鳴らす
すべての音を止める
ピッチ の効果を 10 ずつ変える
ピッチ の効果を 100 にする
音の効果をなくす
音量を -10 ずつ変える
音量を 100 %にする
音量

# 感じる・判断する・動く


イベント
プログラムがスタートしたとき
A 色が のとき
A が 押された とき
A 距離が 8 % より近い とき
傾き で実行する
前 が上のとき
シェイクされた のとき
左 ボタンが 押された とき
タイマーが 10 秒を過ぎたとき
とき
メッセージ1 を受け取ったとき
メッセージ1 を送る
メッセージ1 を送って待つ

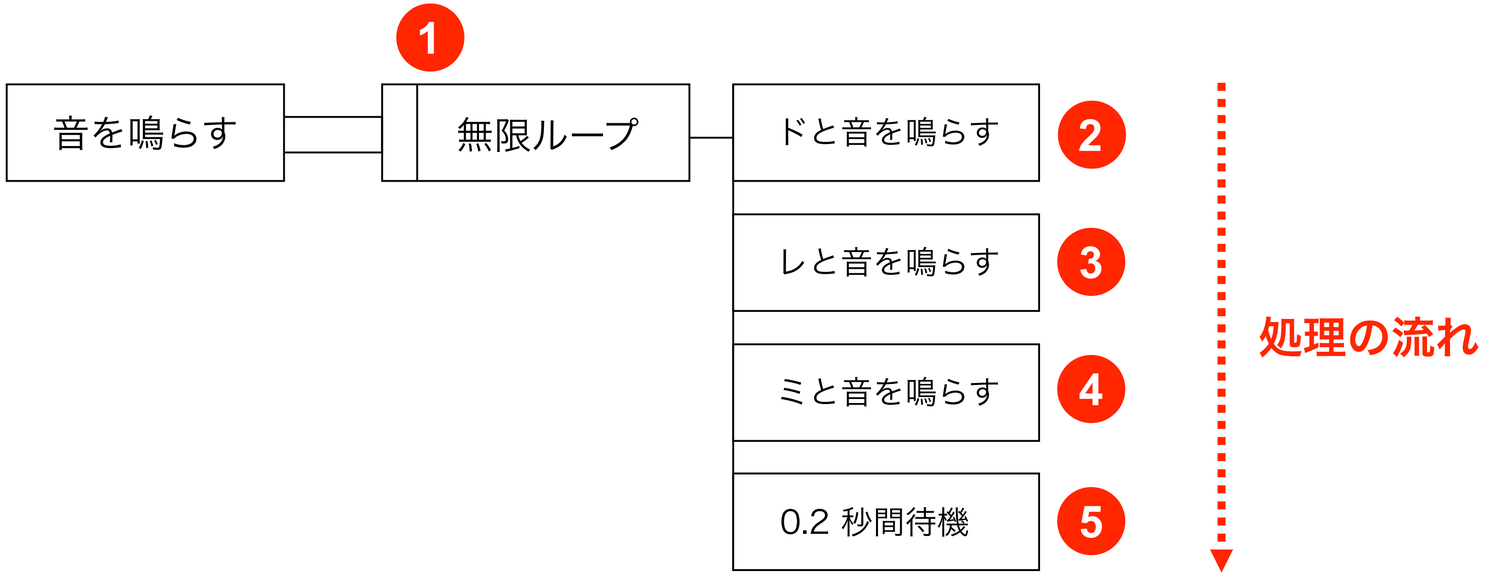

制御
1 秒待つ
10 回繰り返す
ずっと
もし なら
もし なら
でなければ
まで待つ
まで繰り返す
他のスタックの停止
すべて の停止


センサー
A 色が
A の色
A 反射光が < 50 %
A の反射光
A が 押された
A 圧力 %
A 距離が 15 % より近い
A 距離 %
の傾きを検知する
向きが 前
シェイクされた
ピッチ 角
ヨー角を0に設定
左 ボタンが 押された
タイマー
タイマーリセット

## ■音をならしてみよう

# 音を鳴らすプログラムの設計図（PAD）

・指定したサウンドファイルを無限ループで再生




PAD(Problem Analysis Diagram)はプログラムの設計図

PADの構成部品：

処理：

選択：

反復：

# 練習問題1

- テキスト14ページの演習問題を実施しよう

- テキスト29ページ「ドレミの音を鳴らすアルゴリズム」のPADをノートに書き写して流れを理解しよう

# 音を鳴らすプログラム

## 1. 指定したサウンドを再生

p.29：melody.llsp3

# 音を鳴らすプログラム（Python）

1. 指定したサウンドを再生

p.30：melody.py

```
from hub import sound
import runloop
import time
sound.stop()

async def main():
    while True:
        # play a song
        await sound.beep(262, 500, 100) #ド 261.63 Hz, 500ms, 音量100
        await sound.beep(294, 500, 100) #レ 293.67 Hz, 500ms, 音量100
        await sound.beep(330, 500, 100) #ミ 329.63 Hz, 500ms, 音量100
        time.sleep_ms(200)

runloop.run(main())
```

# while文を用いたループ

- while True (字下げ：インデント)とすると無限ループ

```
while True:
    字下げ
        繰り返す処理

次の命令
```

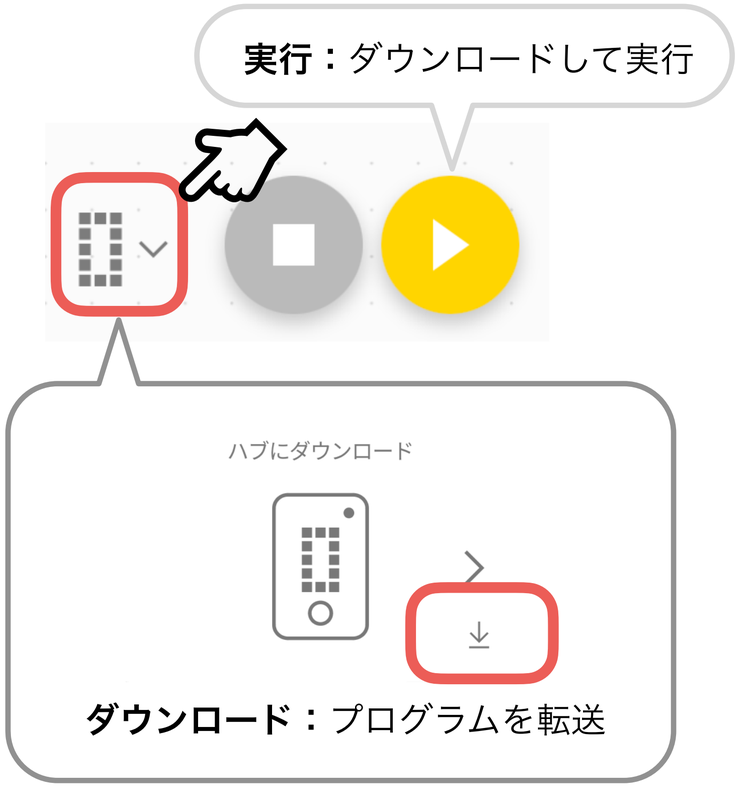
無限ループのプログラムを作成する時は：
プログラムの緊急停止（無限ループ脱出）のための処理を必ず入れておくこと
無限ループ脱出後の処理として、プログラムを停止する命令(モータ停止など)を追加すること

# プログラムの転送

- プログラム名の変更
- ロボットが接続されているか確認　→ ×: 

○: 

- ダウンロード
- ロボット上でプログラムを実行



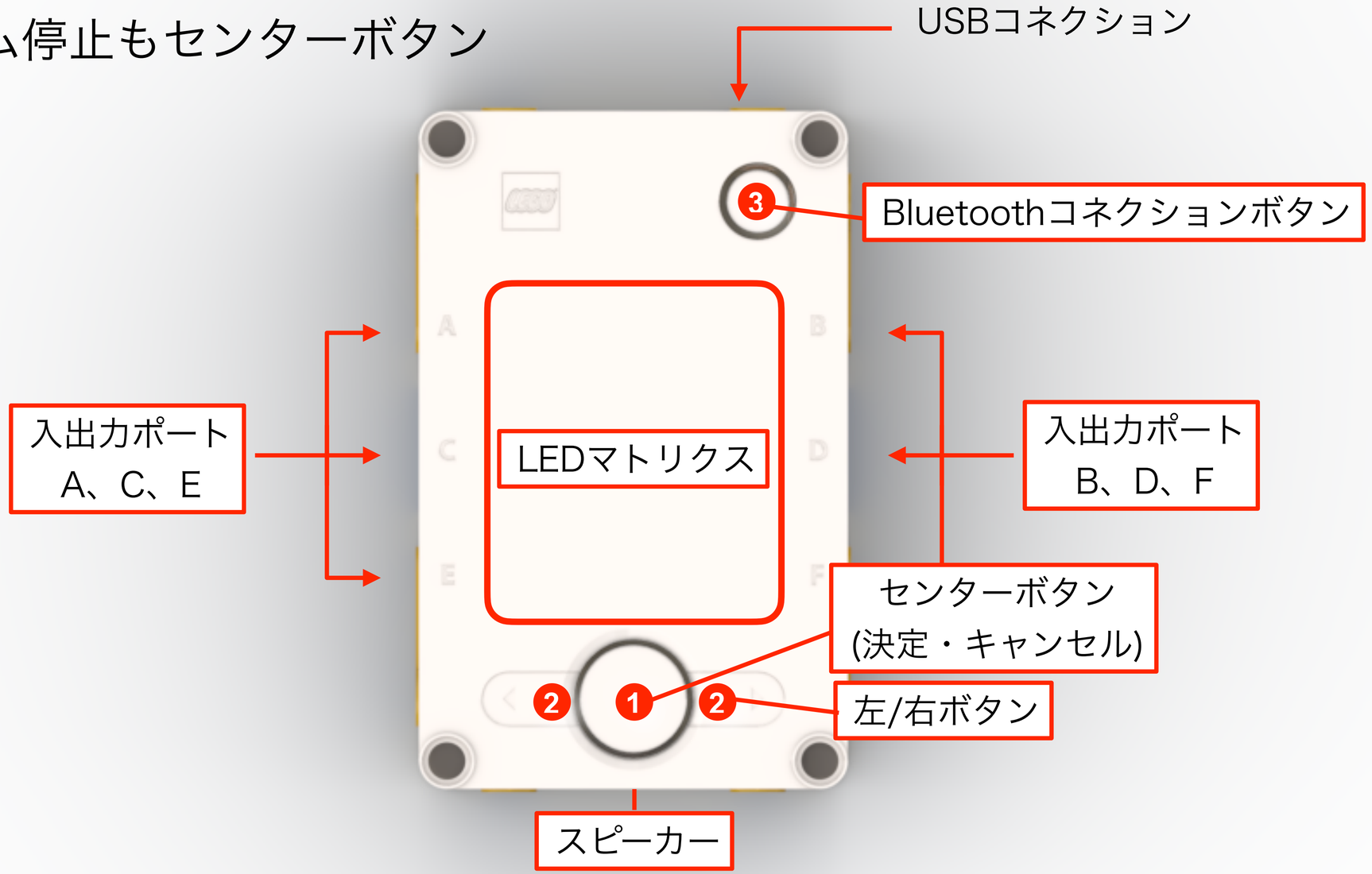

**注意：**

実行ボタンを押すとダウンロードと実行

→ 急にロボットが動く場合があるので注意

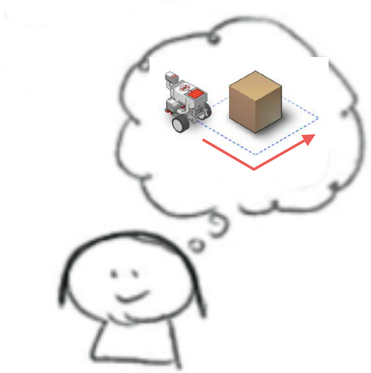
# プログラムの実行

- プログラム番号(０～)を選択してセンターボタンで実行
  - プログラム停止もセンターボタン

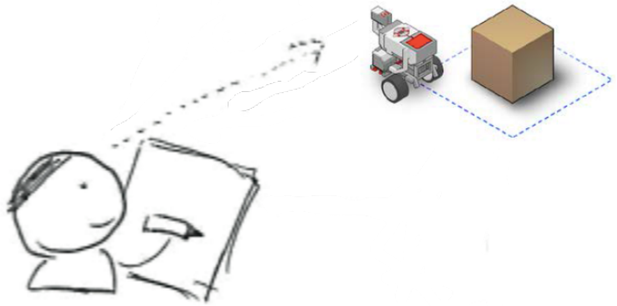


# プログラムの実行

- 実行前にすべきこと
  - ケーブルを外して、安全確認
  - プログラムのアルゴリズムを頭の中で実行

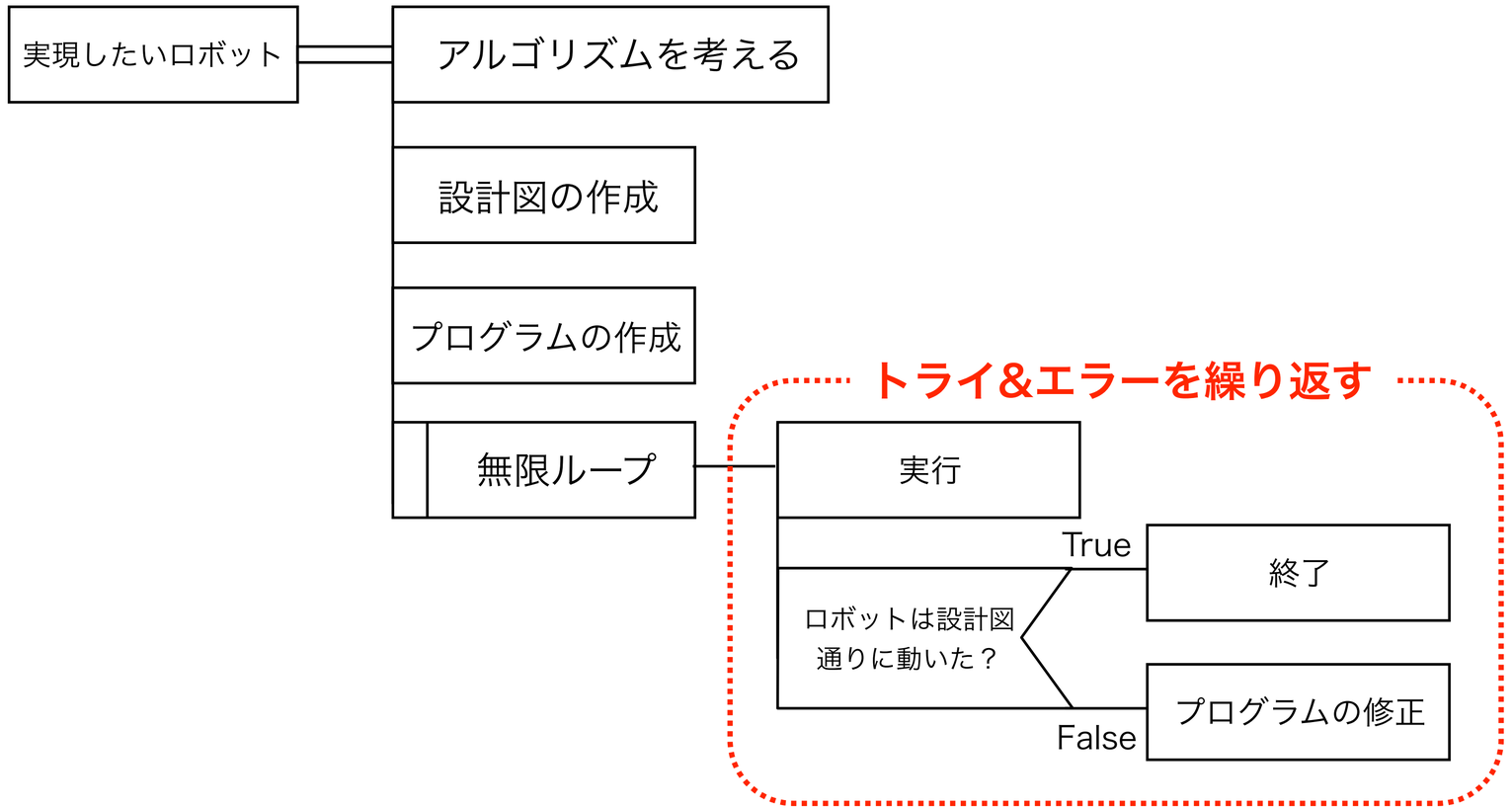


- 実行時の注意
  - ロボットの動作をよく観察し、思った通りに動いているかを確認
  - ロボットが思い通りに動かないときは、ロボットがどこまで設計図通りに動いたかを調べ、プログラムを修正（デバッグ）する



# トライ＆エラーによる問題解決

- ロボットを思い通りに動かすには



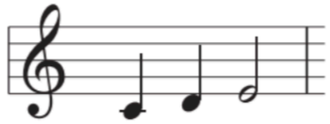

→設計図通りにロボットが動くまでトライ&エラーを繰り返して問題解決

# 練習問題２

・ テキスト35ページの演習問題を実施しよう

**コラム 4：テンポ (BPM)**

音符でメロディを奏でるには，テンポである BPM を決定する必要があります．BPM は，**Beat Par Minute** であり，1 分間あたりの四分音符の数を表します．



$$\text{四分音符の長さ（秒）} = \frac{60}{\text{BPM}}$$

BPM= 120 とすると，四分音符の長さは $60 \div 120 = 0.5$ 秒となります．八分音符は，その半分の 0.25 秒，二分音符は 1 秒となります．BPM によって，各音符の長さが変化します．

| | | | 周波数 | 秒 | (msec) |
|---|---|---|---|---|---|
| ド | 四分音符 | → | 523.25 Hz | 0.5 秒 | 500 |
| レ | 四分音符 | → | 587.33 Hz | 0.5 秒 | 500 |
| ミ | 二分音符 | → | 659.26 Hz | 1.0 秒 | 1000 |