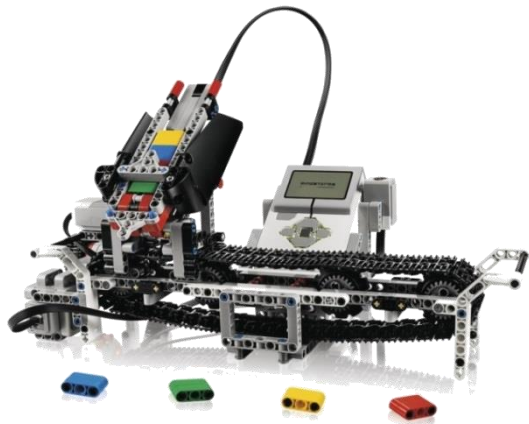


# mruby 学習に最適な、組込み素材、 教育版レゴ マインドストームEV3

軽部 禎文

- 教育版レゴ マインドストームの 正規販売代理店





## 明るい未来社会をつくろう

未来を作るのは「人」

人材育成と未来社会づくりに貢献します。

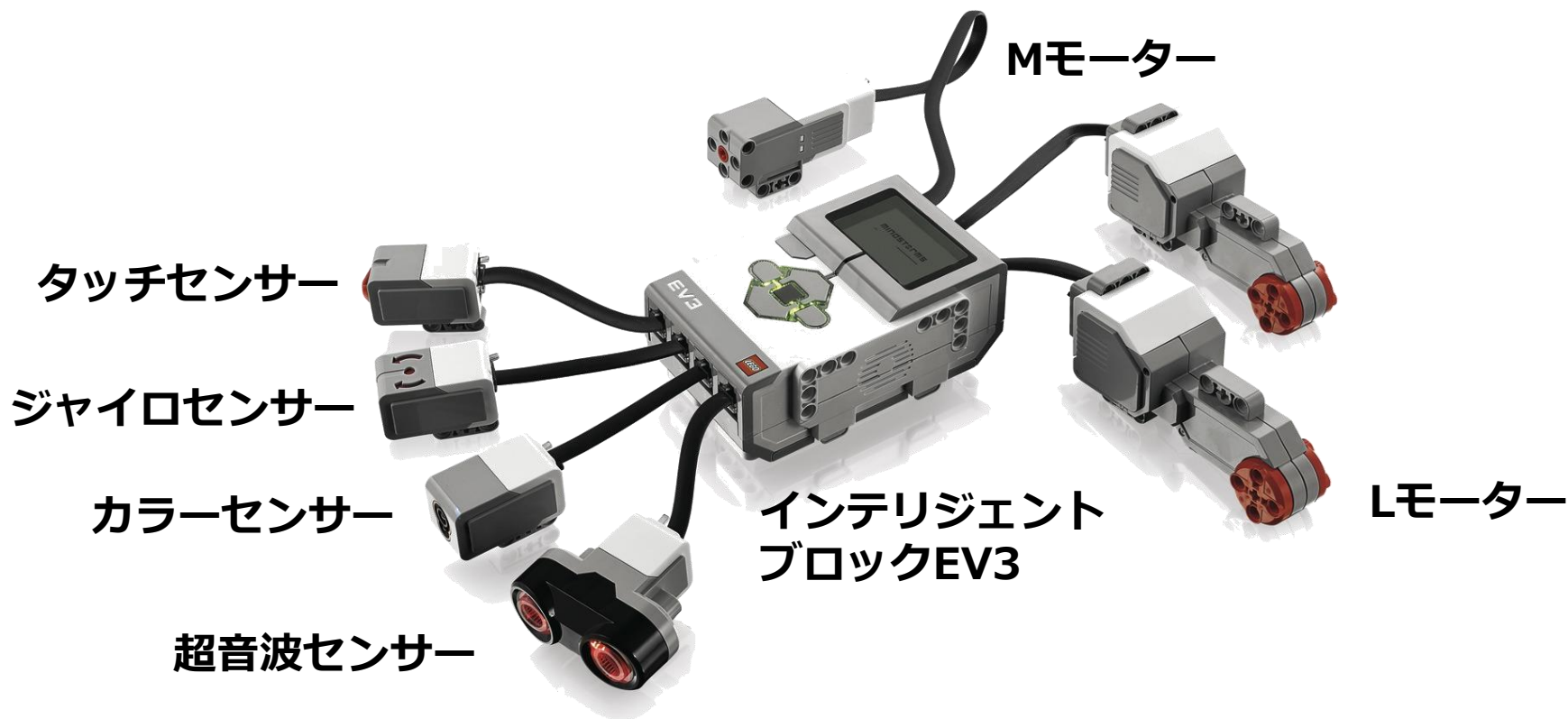
## 教育版レゴ マインドストームを活用した人材育成

- 教材の作成（テキストなど）
  - 子供向けの授業用
  - 先生向けの授業用
  - 企業研修用
- ワークショップの開催
  - 親子ロボット教室 etc...
- ロボコンの企画運営等
  - WRO
  - ETロボコン etc...



# 教育版レゴ マインドストームEV3

- コンピューター、センサー、モーターの組み合わせ
- 自律型のロボットを作成
- 自由に動くから、なにより、楽しい



- 教育版EV3ソフトウェア



- プログラミングの敷居が低い！
- 学びやすい。

- TOPPERSプロジェクト
  - TOPPERS/HRP2カーネル上でmrubyが動く環境を開発されました。

## ■サウンド関連

### サウンド制御 API

戻り値	メソッド
nil	<b>Speaker.volume=vol</b> 音量を調整する。 volume: 0~+100 <b>Speaker.tone(frequency, duration)</b> 周波数と時間を指定してトーン出力する。再生中のサウンド frequency: 周波数 duration: 音を鳴らす時間 [ミリ秒]

## ■バッテリー関連

### サウンド制御 API

戻り値	メソッド
Numeric (電流[mA])	<b>Battery.ma</b> バッテリーの電流を取得する。
Numeric (電圧[mV])	<b>Battery.mv</b> バッテリーの電圧を取得する。

## ■LCD 関連

シンボル: フォントのサイズ (font)	シンボル: 矩形の色 (color)
:small	:white
:medium	:black

### LCD 制御 API

戻り値	メソッド
nil	<b>LCD.font=font</b> フォントを設定する。 font: フォントのサイズを指定する。(上表参照)
nil	<b>LCD.draw(str, x, y)</b> 指定位置で文字列を描く。 str: 表示する文字列 x: 水平方向の表示位置 (小数点以下切り捨て) y: 垂直方向の表示位置 (小数点以下切り捨て) x, y の単位は文字。フォントによって同じ値でも表示位置が変わる。
nil	<b>LCD.fill_rect(x, y, w, h, color)</b> 矩形を描いて色を塗る。 x: 水平方向の位置 (小数点以下切り捨て) y: 垂直方向の位置 (小数点以下切り捨て) w: 矩形の幅 (小数点以下切り捨て) h: 矩形の高さ (小数点以下切り捨て) color: 矩形の色を指定する。(上表参照)
nil	<b>LCD.draw_line(x0, y0, x1, y1)</b> 指定座標で線を描く。 x0: 始点の水平方向の位置 y0: 始点の垂直方向の位置 x1: 終点の水平方向の位置 y1: 終点の垂直方向の位置

## ■モーター関連

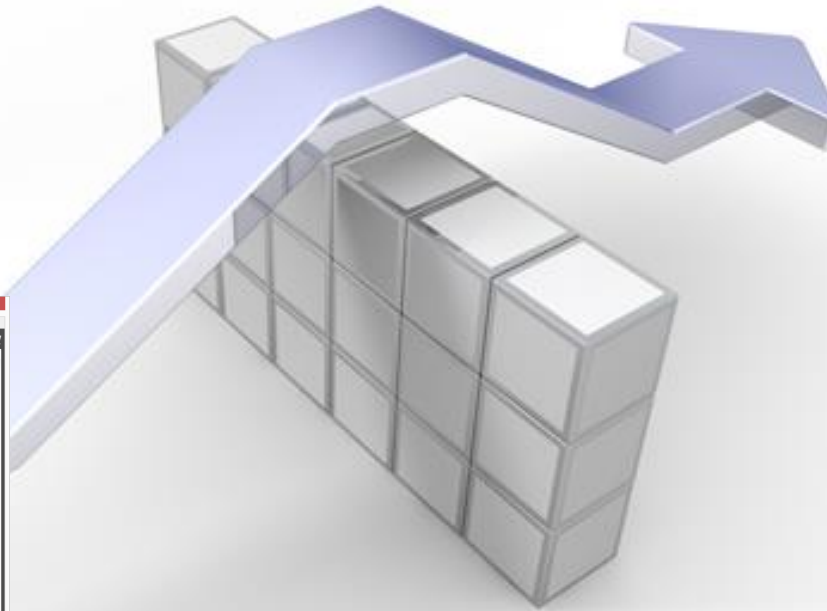
シンボル: モーターのポート (MotorPort)	シンボル: モーターのタイプ (MotorType)
:port_a	:medium
:port_b	:large
:port_c	
:port_d	

シンボル: モーターのタイプ (MotorType)	シンボル: モーターのタイプ (MotorType)
:medium	M モーター
:large	L モーター

### モーター制御 API

戻り値	メソッド
nil	<b>initialize(port, type=:large)</b> モーターポートとポートに接続しているモーターのタイプを設定する。 port: モーターのポートを指定する。(上表参照) type: モーターのタイプを指定する。(上表参照)
MotorType (モーターのタイプ)	<b>type</b> モータータイプを取得する。
nil	<b>powerpwm</b> モーターのパワーを設定し、回転させる。 pwm: -100~+100 (PWM 値) マイナス値でモーターが逆回転する
Numeric (モーターのパワー)	<b>power</b> モーターのパワーを取得する。
Numeric (モーターの角位置)	<b>count</b> モーターの角位置を取得する。(単位は度)
nil	<b>reset_count</b> モーターの角位置をゼロにリセットする。

- アイコン型で学んだ後、言語系を学ぶ



```

/mruby-on-ev3rt+tecs/rb-tecs/workspace/mruby_app
include EV3RT_Tecs
class Tank
  def initialize(l_port, r_port)
    @left_port = l_port
    @right_port = r_port
    @left_motor = Motor.new(@left_port)
    @right_motor = Motor.new(@right_port)
  end

  #左モーターポート
  def get_left_port
    return @left_port
  end

  #右モーターポート
  def get_right_port
    return @right_port
  end

  #イメージ制御
  def drive_time(leftpow, rightpow, time)
    #左モーターの回転速度を初期化
    @left_motor.reset_count
    @right_motor.reset_count

    @left_motor.power = leftpow
    @right_motor.power = rightpow
    RTOS.delay(time)

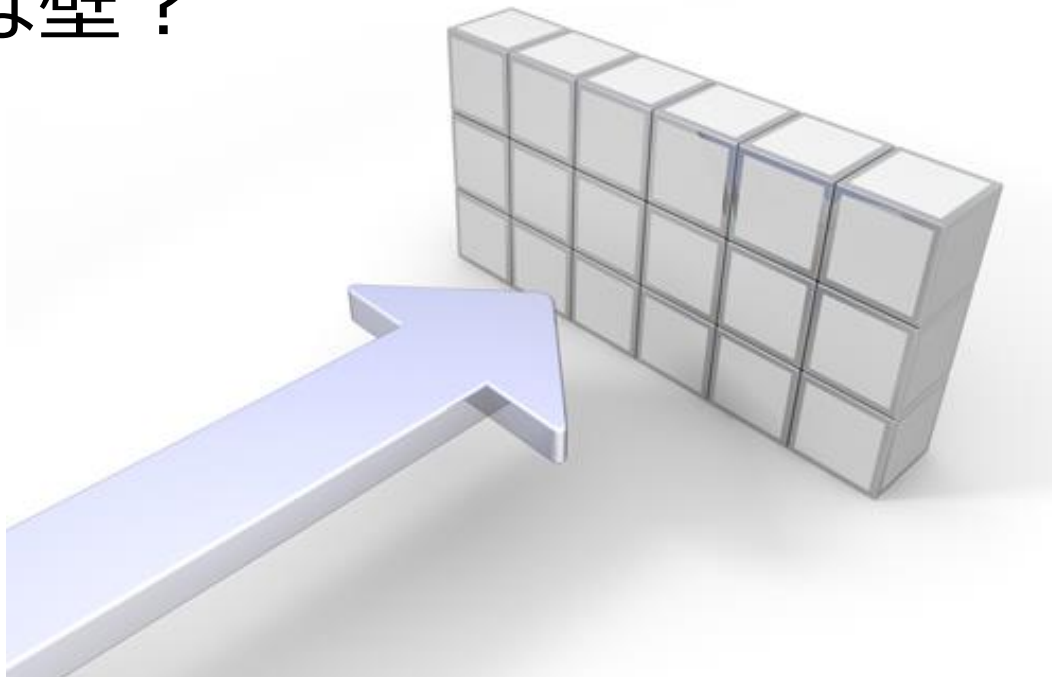
    @left_motor.stop(true)
    @right_motor.stop(true)
  end
end
1,1

```

- アイコン型で学んだ後に、言語系を学ぶ間に、壁がある



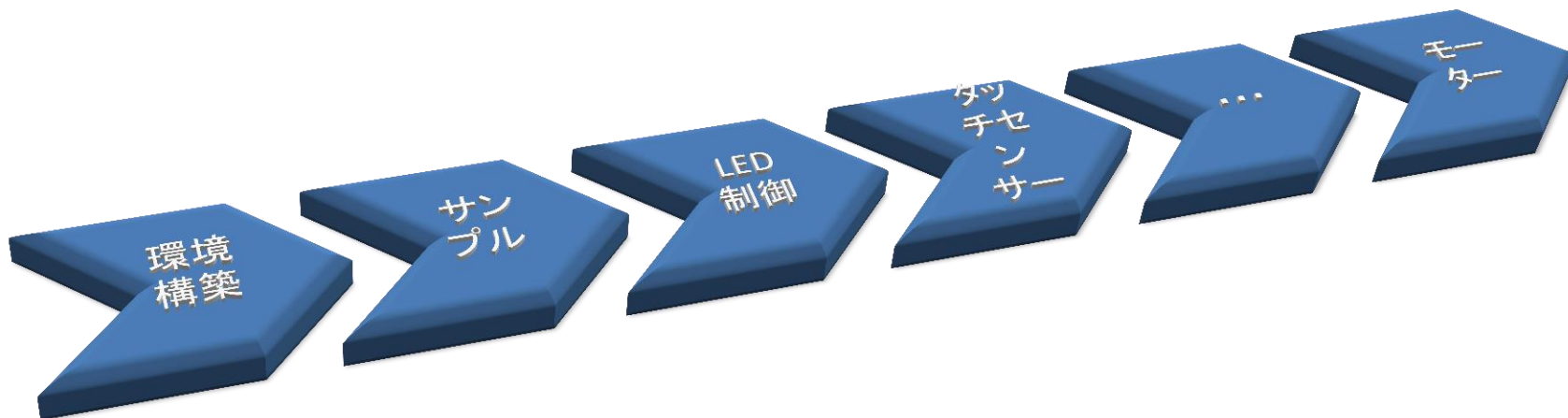
- どんない壁？



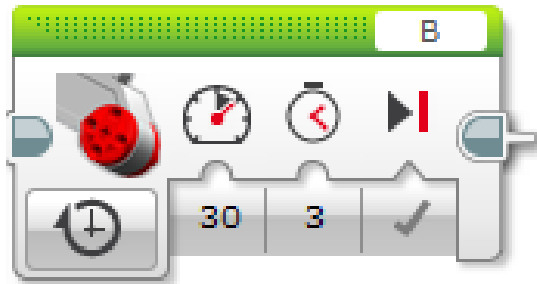
1. 指導者の問題
2. 電子・電気系、機械系の知識の問題
3. ソフトウェア開発環境の問題

# 1. 指導者の問題

テキストを読み進めていく事で学ぶ



## – アイコンのプログラムと、言語のプログラムの対比



モーターBを  
パワー30で3秒前進して止まる



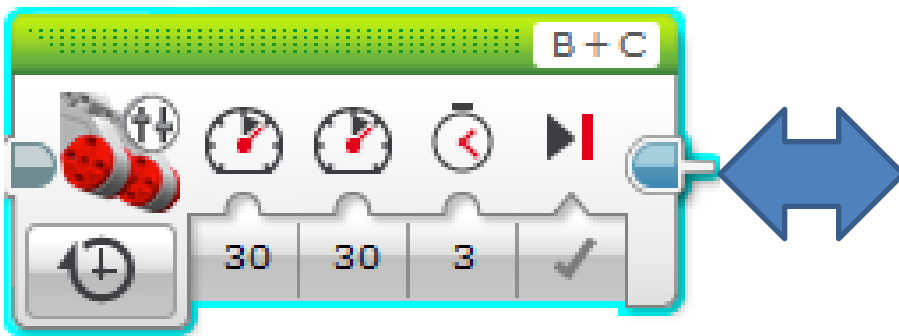
```
#ポートの設定
right_port = :port_b

$right_motor = Motor.new(right_port)

#モーターを3秒間前進
$right_motor.power = 30
RTOS.delay(3000)

$right_motor.stop(true)
```

## – アイコンのプログラムと、言語のプログラムの対比



BとCのモーターを  
パワー30で3秒前進して止まる

```
class Tank
  def initialize(l_port, r_port)
    @left_port = l_port
    @right_port = r_port
    @left_motor = Motor.new(@left_port)
    @right_motor = Motor.new(@right_port)
  end

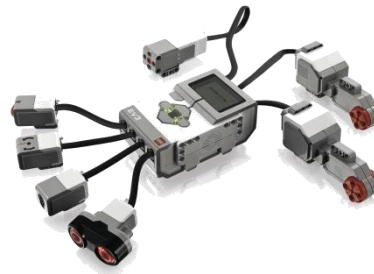
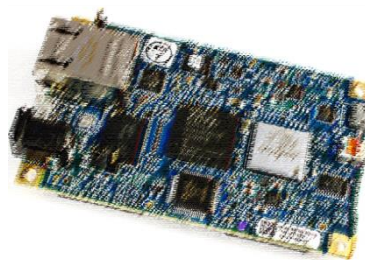
  #タイマー制御
  def drive_time(leftpow, rightpow, time)
    #各モータの各位置を初期化
    @left_motor.reset_count
    @right_motor.reset_count

    @left_motor.power = leftpow
    @right_motor.power = rightpow
    RTOS.delay(time)

    @left_motor.stop(true)
    @right_motor.stop(true)
  end
end
```

## 2. 電子・電気系、機械系の知識の問題

- センサー、モーターをケーブルでつなぐだけ



- アイコン系のソフトウェアで動かしてきた子供たちにとっては、慣れ親しんだハードウェア



- 言語の学習に注力できる

### 3. ソフトウェア開発環境の問題

- やっぱり、ちょっと大変。
- いくつものソフトウェアをインストールしたり、設定したり・・・
- 動かして楽しむ前に、楽しくない。



- 手順を明確に、分かりやすく。

- 初めてmrubyを学びたいエンジニアの方や、組み込み系でない方にとっても良いと思いませんか？

まつもとゆきひろ 様より

mrubyは組み込みでも使えるRuby処理系です。  
ロボットを含むデバイスシステムでもWebアプリ開発で好評のRuby言語の強力なパワーが  
使えるようになります。  
Rubyがもたらすプログラマーの自由をお楽しみください。

- 初めて学ぶ言語系プログラミング言語  
mruby Ruby
- 楽しさ
- 将来のソフトウェアエンジニアとして育てて欲しい。