

# LEGO Mindstorms のマイコンの走行体を用いた 制御系の Java プログラミング演習の環境構築

## Construction of Environments for Control of LEGO Mindstorms in Java Programming Exercises

劉 世博<sup>\*1</sup>, 中井 智己<sup>\*1</sup>, 富永 浩之<sup>\*1</sup>  
Shibo RYU<sup>\*1</sup>, Tomoki NAKAI<sup>\*1</sup>, Hiroyuki TOMINAGA<sup>\*1</sup>  
<sup>\*1</sup>香川大学工学部  
<sup>\*1</sup>Faculty of Engineering, Kagawa University  
Email: s14g486@stmail.eng.kagawa-u.ac.jp

あらまし : LEGO Mindstorms のマイコンを搭載した走行体の制御プログラミングの演習を, 大学情報系学科の授業で実践している. プロジェクト単位のグループ学習として, 技術要素を含んだゲーム課題に取り組み. 新版 EV3 への移行として, 実行環境を Linux ベースの leJOS に, 開発環境をオブジェクト指向の Java 言語に変更した. 多様なセンサを扱うライブラリとして, 各センサのサブクラスを設け, メソッドを構築した. キャリブレーション用の機能も提供した.

キーワード : LEGO マイコン教材, Java 演習, 制御プログラミング, センサ処理

### 1. はじめに

LEGO Mindstorms は, LEGO 社と MIT が共同開発した教育玩具である. キットは, マイコンおよびモータや各種のセンサを含む LEGO ブロックで構成される. これらを組み合わせ, センサで外部環境を感知し, モータで動作する自律ロボットが制作できる. 制御プログラムは PC で作成し, USB ケーブルでマイコンに転送する. 開発環境は, ビジュアルベースの簡易なものから, 各種のプログラム言語まで提供されている. Mindstorms は, 小中高から大学まで, 幅広い教育現場で使われている.

### 2. EV3 キットと規定ロボット

本研究室では, 大学情報系の専門課程の必修科目「情報環境実験 2」において, 2010 年度から LEGO ロボット制御とゲーム課題による制御プログラミングの演習を実施している<sup>(1)</sup>. 四半期の演習を, 8 週間程度で実施する. 各週で, ほぼ 1 つのプロジェクトとし, 実技認定を行うゲーム課題を含む. 5 つのプロジェクトの後, 最終課題の競技大会には, 3 週間程度で取り組む. 活動を総括するレポートも課す.

本演習では, 走行体として規定ロボットを用意する. 1 グループ 4~5 人とし, 2 台の規定ロボットを与え, 幾つかの実技課題を提示する(図 1). 各課題は, ゲームフィールド上のコースやエリアを走行し, ゲーム感覚の任務要素を実現する(図 2). メンバで分担して, 攻略法の設計, プログラムの実装, 動作の検証を行う. 時間による走行点と, 達成度による任務点を合計して, 実技認定の得点とする.

2014 年度までの旧版の NXT キットに対し, 2015 年度から, 新版の EV3 キットに移行した<sup>(2)</sup>. ハードウェアとしては, 本体のマイコンの性能が大きく向上した. USB と microSD のポートが設置され, Bluetooth を内蔵し, 別売の dongle で WiFi 接続も

可能である. 付属部品には, 大小 2 種類のモータ, 4 種類のセンサ(カラー, タッチ, ソナー, ジャイロ)がある. センサの精度も向上している. 別売で IR センサと IR ビーコンも使える.

EV3 キットで新しい規定ロボットを作成した. 車輪機構, 接触センサと着脱式のバンパー, 床下検知の 2 つの色彩センサ, 前方と側方に付け替えられる反響センサなどは, ほぼ同じである. なお, キットのパーツ構成から, 1 つの後輪が 1 つの自由球になった. しかし, 従来と同じように, 多くのゲーム課題に対処できる. ただし, 現時点では, 手腕機構は実現できていない.

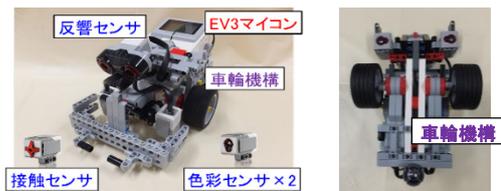


図 1 LEGO 演習の規定ロボット

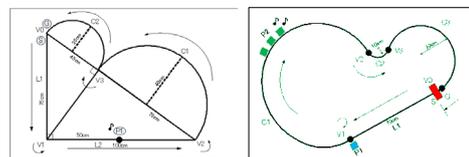


図 2 図形模走と黒線追跡のゲームフィールド

### 3. leJOS 上でのセンサ関連のライブラリ

開発環境も ROBOTC による C プログラミングから leJOS による Java プログラミングに移行した. これにより, 擬似的なマルチタスクから, スレッドによる並列処理が可能となった. leJOS は, Oracle 純正の仮想マシンを含んだ Linux OS である. 組込機器向けに開発された Java ランタイムを使用しているた

め、標準の Java API が利用できる。例えば、スレッドによる並列処理が可能である。また、USB ケーブルによるプログラムの転送が可能のため、簡単に実行できる。そこで、本演習では、leJOS を採用する。

leJOS のライブラリを使用し、センサの値を取得する場合、図 3 の手順で取得する。EV3 のセンサは、NXT と比べ、それぞれの機能が揃っている。例えば、EV3 の色彩センサは、NXT の光量センサの機能も備えている。そのため、手順②で、センサのモードを切り替えている。手順③で、センサの値は、float 型の配列に取得される。色彩センサでは、RGB の 3 値 (0.0~1.0) が各要素に格納される。接触センサでは、真偽値(0/1)が第 0 要素に格納される。

このように、leJOS のライブラリを使用してセンサの値を取得すると、プログラムが冗長になる。そのため、センサを簡潔に制御するためのライブラリを用意した。すなわち、各センサ用のサブクラスを設け、それぞれに適した取得メソッドを用意した<sup>(3)</sup>。サブクラスには、接触センサ、色彩センサ、反響センサなどがある。色彩センサでは、4 つのモード(外光輝度、反射輝度、色彩判別、色彩成分)にも対応した。作成したライブラリは、図 4 のように利用する。

各センサに対し、それぞれに応じた値を取得するためのメソッドを用意する。接触センサでは、ボタンが押されているかを boolean 型で取得する isPress メソッドを実装する。色彩センサでは、反射光を float 型で取得する getLight メソッドと、RGB 値を float 型の配列で取得する getRGB メソッドを実装する。また、RGB の各値を取得する getRed, getGreen, getBlue メソッドも実装する。反響センサでは、物体との距離の相対値を float 型で取得する getDistance メソッドを実装する。

各クラスは、Sensor インタフェースを継承し、getValue メソッドを実装している。getValue メソッドは、センサの値を float 型の配列で受け取る。つまり、もともとの leJOS のライブラリを使用して、センサの値を取得する方法も残している。

#### 4. キャリブレーション用のライブラリ

黒線追跡のライントレースや色彩タイルの判別において、外光の影響を避けて安定的に処理を行うには、色較校正すなわちキャリブレーションが必要である。今回、RGB 値に対するキャリブレーションのため、RGBCalibration クラスを立てた。コンストラクタの引数 n はキャリブレーションの回数で、引数 color は対象とする色彩センサである。オブジェクト変数として、キャリブレーションの回数 k と、各回の RGB 値を格納する二次元配列 calibrator[][]がある。

メソッド executeCalibration()は、各回のキャリブレーションとして、RGB 値を読み取り、結果を格納する。メソッド getCalibData(k)は、k 回目の RGB 値を返却する。メソッド getCalibData(k)は、k 回目の RGB 値を EV3 マイコンの LCD 画面に表示する。

- ① ポートを指定し、センサのインスタンスを作成
- ② センサのインスタンスからモードを取得
- ③ モードから float 型の配列でセンサの値を取得
- ④ float 型の配列から目的の値を取得

図 3 leJOS でのセンサ値の取得の手順

```
EV3TouchSensor touchSensor
    = new EV3TouchSensor(SensorPort.S1);
SensorMode touch = touchSensor.getMode(0);
float value[] = new float[touch.sampleSize()];
touch.fetchSample(value, 0);
float state = value[0];
```

図 4 センサライブラリの利用コード例

```
RGBCalibration rightCalibration
    = new RGBCalibration(3, rightColor);
LCD.drawString("=Right Calibration=", 0, 0);
enterPressWait();
rightCalibration.executeCalibration();
LCD.clear();
```

図 5 キャリブレーションの利用コード例

#### 5. おわりに

応用プログラミングとして、LEGO ロボットの制御とゲーム課題を題材とするグループ演習を提案している。組込制御の基礎、ソフトウェアの開発工程を内容とし、問題解決の手段としてのプログラミングを習得する。新版のキットに対応した規定ロボットを構築し、Java leJOS 環境に移行した。課題を円滑に進めるため、センサを扱うライブラリとキャリブレーションを行うクラスを作成した。教育実践として、大学情報系で四半期の演習を行う。

演習では、RGB によるキャリブレーションでは、うまく動作しないことが多かった。そのため、今後は、色相と明暗の情報を分離しやすい HSV にも対応する。また、技術解説のオンライン教材の構築や、UML によるモデリング設計の教育支援も検討する。

#### 参考文献

- (1) 加藤聡, 富永浩之: "LEGO ロボットの制御プログラミングを題材とした問題解決型の応用演習 -ROBOTC による基本制御の練習問題の教材検討と授業実践-", 情報処理学会 第 78 回全国大会, Vol.78, pp.893-894 (2016).
- (2) 中井智己, 辻健人, 富永浩之: "LEGO ロボットの制御プログラミングとゲーム課題の攻略を題材とする Java グループ演習の授業設計", 情報処理学会 第 78 回全国大会, Vol.78, pp.893-894 (2016).
- (3) 中井智己, 内山豊, 辻健人, 富永浩之: "LEGO EV3 を制御する leJOS 上のライブラリの構築とゲーム課題のための Java サンプルの提供", ゲーム学会 第 14 回全国大会, Vol.14, pp.27-30 (2016).