バージョン: 0.1.1

ETロボコンアーカイブス

２０１５～２０１６活動

|  |  |  |
| --- | --- | --- |
| 承認 | 確認 | 作成 |
|  |  |  |

**改訂履歴**

|  |  |  |  |
| --- | --- | --- | --- |
| 日付 | Ver. | 内容 | 制作者 |
| 2016/09/30 | 0.0.1 | 新規作成 | 大塚　信晶 |
| 2016/12/31 | 0.1.1 | 2016/09/30中間レビュー適用 | 大塚　信晶 |

**用語一覧**

|  |  |
| --- | --- |
| 名称 | 意味 |
| EV3 | 大会で使用した走行体本体を指す。 |
| API | 大会提供の関数群・クラス群を指す。バランス制御用フィードバック制御関数もこれに含まれる。 |
| サンプルコード | 大会提供サンプルコード、特に「ev3way-cpp」を指す。 |
| プロジェクト | 本大会で自チームが作成したコード・アプリケーション・UML設計資料全般を指す。 |
| 尻尾 | 走行体後部の主に本体を支える用途で使用する可動式の支え棒を指す。 |
| 走行体角度 | 走行体の向く走行方向を走行体角度または走行角度とする。 |

目次

[1. はじめに 7](#_Toc471690891)

[1.1. 目的 7](#_Toc471690892)

[1.2. 資料構成 7](#_Toc471690893)

[1.2.1. ２０１５年・２０１６年の活動 7](#_Toc471690894)

[1.2.2. 気づきと対策 7](#_Toc471690895)

[1.2.3. ETロボコンのパターンとリスク 8](#_Toc471690896)

[1.2.4. ソフトウェア設計思想 8](#_Toc471690897)

[1.2.5. 今後の課題 8](#_Toc471690898)

[1.2.6. おわりに 8](#_Toc471690899)

[2. ２０１５年・２０１６年の活動 9](#_Toc471690900)

[2.1. ２０１５年 9](#_Toc471690901)

[2.1.1. 活動テーマ「全競技内容をクリアする」 9](#_Toc471690902)

[2.2. ２０１６年 9](#_Toc471690903)

[2.2.1. 活動テーマ「ライントレース機能の実装」 9](#_Toc471690904)

[3. 気づきと対策 10](#_Toc471690905)

[3.1. バランス制御（デバイス特性） 10](#_Toc471690906)

[3.1.1. 問題 10](#_Toc471690907)

[3.1.2. 原因と対策 10](#_Toc471690908)

[3.2. バランス制御（プログラム） 10](#_Toc471690909)

[3.2.1. 問題 10](#_Toc471690910)

[3.2.2. 原因と対策 10](#_Toc471690911)

[3.3. 光学センサの値 11](#_Toc471690912)

[3.3.1. 問題 11](#_Toc471690913)

[3.3.2. 原因と対策 11](#_Toc471690914)

[3.4. 環境による光学センサの取得値の変化 11](#_Toc471690915)

[3.4.1. 問題 11](#_Toc471690916)

[3.4.2. 原因と対策 11](#_Toc471690917)

[3.5. 走行速度の壁 11](#_Toc471690918)

[3.5.1. 問題 11](#_Toc471690919)

[3.5.2. 原因と対策 11](#_Toc471690920)

[3.6. 尻尾の破損（１回目） 13](#_Toc471690921)

[3.6.1. 問題 13](#_Toc471690922)

[3.6.2. 原因と対策 13](#_Toc471690923)

[3.7. 尻尾の破損（２回目） 13](#_Toc471690924)

[3.7.1. 問題 13](#_Toc471690925)

[3.7.2. 原因と対策 13](#_Toc471690926)

[3.8. API「GyroSensor.getAnglerVelocity()」の未解決問題 14](#_Toc471690927)

[3.8.1. 問題 14](#_Toc471690928)

[3.8.2. 原因と対策 14](#_Toc471690929)

[3.9. 障害物突破に避けて通れない決め打ち走行 14](#_Toc471690930)

[3.9.1. 問題 14](#_Toc471690931)

[3.9.2. 原因と対策 14](#_Toc471690932)

[3.10. コーディングとパラメータ調整の分離 15](#_Toc471690933)

[3.10.1. 問題 15](#_Toc471690934)

[3.10.2. 原因と対策 15](#_Toc471690935)

[3.11. コンセプトシート再提出 15](#_Toc471690936)

[3.11.1. 問題 15](#_Toc471690937)

[3.11.2. 原因と対策 15](#_Toc471690938)

[3.12. 走行開始時のバランス制御 16](#_Toc471690939)

[3.12.1. 問題 16](#_Toc471690940)

[3.12.2. 原因と対策 16](#_Toc471690941)

[3.13. 走行中のバランス制御のON/OFF 16](#_Toc471690942)

[3.13.1. 問題 16](#_Toc471690943)

[3.13.2. 原因と対策 16](#_Toc471690944)

[4. ETロボコンのパターンとリスク 17](#_Toc471690945)

[4.1. 試走会の順番待ちと言う空白の時間 17](#_Toc471690946)

[4.1.1. パターン 17](#_Toc471690947)

[4.1.2. リスク 18](#_Toc471690948)

[4.2. アナウンスの不足 18](#_Toc471690949)

[4.2.1. パターン 18](#_Toc471690950)

[4.2.2. リスク 18](#_Toc471690951)

[4.3. 常連はなぜ有利か 18](#_Toc471690952)

[4.3.1. パターン 18](#_Toc471690953)

[4.3.2. リスク 19](#_Toc471690954)

[4.4. 運営が歓迎するショートカット 19](#_Toc471690955)

[4.4.1. パターン 19](#_Toc471690956)

[4.4.2. リスク 19](#_Toc471690957)

[4.5. 部品忘れ 19](#_Toc471690958)

[4.5.1. パターン 19](#_Toc471690959)

[4.5.2. リスク 19](#_Toc471690960)

[4.6. UML資料の評価 20](#_Toc471690961)

[4.6.1. パターン 20](#_Toc471690962)

[4.6.2. リスク 20](#_Toc471690963)

[4.7. 試走１・試走２・地区大会での環境光の変化 20](#_Toc471690964)

[4.7.1. パターン 20](#_Toc471690965)

[4.7.2. リスク 20](#_Toc471690966)

[5. ソフトウェア設計思想 21](#_Toc471690967)

[5.1. 21](#_Toc471690968)

[6. 今後の課題 22](#_Toc471690969)

[6.1. ライントレース 22](#_Toc471690970)

[6.2. 障害物「ルックアップゲート」 22](#_Toc471690971)

[6.3. 障害物「段差または階段」 22](#_Toc471690972)

[6.4. UML資料 23](#_Toc471690973)

[7. おわりに 24](#_Toc471690974)

# はじめに

## 目的

本資料は、２年に渡りETソフトウェアデザインコンテスト（以降ETロボコンと

記載）で活動した中で発見した「気づき」と「対策」、２度の大会出場で見た

「パターン」と「リスク」、そしてそれら全てから対応すべき「今後の課題」を

文書として残し、今後のHAL大阪におけるETロボコンの活動を支援することを

目的とする。

## 資料構成

### ２０１５年・２０１６年の活動

２０１５年と２０１６年の活動を簡単に振り返り、各年度の達成目標と

その成果を振り返る。

### 気づきと対策

開発中に経験した問題や失敗、それらに対する策を記載する。項目は以下。

Q１　：バランス制御（初期設定）

Q２　：バランス制御（プログラム）

Q３　：光学センサの値

Q４　：環境による光学センサの取得値の変化

Q５　：走行速度の壁

Q６　：尻尾の破損（１回目）

Q７　：尻尾の破損（２回目）

Q８　：API「GyroSensor.getAnglerVelocity()」の未解決問題

Q９　：障害物突破に避けて通れない決め打ち走行

Q１０：コーディングとパラメータ調整の分離

Q１１：コンセプトシート再提出

Q１２：走行開始時のバランス制御

Q１３：走行中のバランス制御のON/OFF

### ETロボコンのパターンとリスク

勉強会・試走会・地区大会での運営・自チームを含めた参加者の動きのパターン、

そしてそこから考えられるリスクを記載する。項目は以下。

P１：試走会の順番待ちと言う空白の時間

P２：アナウンスの不足

P３：常連はなぜ有利か

P４：運営が歓迎するショートカット

P５：部品忘れ

P６：UML資料の評価

P７：試走１・試走２・地区大会での環境光の変化

### ソフトウェア設計思想

２０１６年度ソフトウェアの設計思想を記述する。

・設計テーマ

・現状分析

・機能

### 今後の課題

以上の内容より。2年間を通して、特に２０１６年度大会でのフィードバックを

重点に置き、今後の達成すべき課題を考察する。

### おわりに

２年間の活動の所感とETロボコンをチーム活動とする上で重要と思うところを

述べる。

# ２０１５年・２０１６年の活動

## ２０１５年

### 活動テーマ「全競技内容をクリアする」

開発難易度の把握、大会内容の把握を重きに置いた。

実感として初年度の活動にはUML設計、進行管理に1名、プログラミング、開発、デバックに２名が適した人員であると思われる。

特に開発環境構築、資料分析、API理解には授業進捗的に2年生後半～３年生前半の能力が適切である。UML設計、進行管理は最も活動に長時間を費やすことが可能な人材が適切であり、学年は問わずとも担当可能である印象。

活動開始時は多数のメンバーが参加したが開発環境構築が大きな壁となり、実活動メンバーは1名となる。

ライントレース機能の実装は諦め、大会コースを用意できたことを利用して記録した走行を行うプログラムを作成。

布製のコースの張りや曲がりでテスト条件が毎度変化し、競技突破可能な状態には至らなかった。

UML設計のテーマは「最短距離でのコース突破」である。

## ２０１６年

### 活動テーマ「ライントレース機能の実装」

前年度活動で実装できなかった機能の実装と、2016年度では大会コースを用意できない背景からライントレース機能の実装は絶対必須項目となった。

前年度の実活動メンバーをUML設計・進行管理専任とし、開発・プログラミングを別途リーダーとメンバーを決めるチーム構成をとった。

前年度問題となった開発環境構築をメンバーに必須とせずに、コーディングのみをメンバーに依頼し、コンパイルチェック・動作確認をリーダークラスで行うようにする。

これにより設計からコーディングまではスムーズに進行するが、コンパイルから動作確認を行える人材が育たず、この作業をUML設計・進行管理役が行う。

この間に夏季休暇に突入し、コーディングも１名で行う状態になり、その後開発は一人で行うことになる。

成果物としては

・PIDを用いたライントレース機能

・コースの現在進行方向の算出機能

・設定ファイルによるパラメータ管理機能

である。

# 気づきと対策

## バランス制御（デバイス特性）

### 問題

２０１５年初参加時に自身でオリジナルのコードを生成した際、バランス制御が成立せず、すぐに走行体が転倒してしまう現象が発生した。

協会提供のサンプルプログラムと同様のAPIを使用しているので、取得値やフィードバック制御のアルゴリズムに間違いはないはずである。

### 原因と対策

EV3のジャイロセンサデバイスには、個体特性により常に一定の値が誤差として検出されてしまう使用がある。

大会に使用した走行体も完全静止時に４[deg/sec]が検出されていた。

対策としてアプリケーション起動時の本体が静止していることが保証された時にジャイロ値を取得し、それをカット値として検出値から常にカットし続ける構造をとった。

これを起因として本プロジェクトではアプリケーション起動時に角加速を加えるとバランス制御が不安定ないしは不能になる特徴を持った。

## バランス制御（プログラム）

### 問題

２０１５年度に開発を進めていく中で開発初期の状態よりも、開発後半の方がバランス制御の精度が劣化する問題が発生した。

### 原因と対策

開発が進む中でジャイロセンサ値を取得する前に行う処理が肥大化し、一定の周期でのジャイロセンサ値取得が行えておらず、それがバランス制御に影響を及ぼしていた。

２０１６年度の開発ではセンシング・アクチュエイト指示が常に一定間隔に行われるよう変更し、この問題を解消している。

## 光学センサの値

### 問題

APIより「getBrightness()」で反射光を取得した際、回収した値が低い（白検知で４０程度）の為、ライントレース制御等で使用する際に値の変化幅が狭すぎてEV3が望む挙動を示さない。

### 原因と対策

コースの白と黒の変化では望む値の変化幅を得ることは不可能である。EV3のカラーセンサ機能「getRawColor()」を使用することで内臓LED３色を同時発光し、その取得値を足し合わせることで大きな値の変化幅を得た。

## 環境による光学センサの取得値の変化

### 問題

会場は試走会・地区大会・本大会でそれぞれ異なる環境光を持っているが、それらはライントレース制御、グレー検知に大きな影響をもたらす要素である。

### 原因と対策

大会側が行う環境光の調整は主に会場にある橙色のライトの光量調整である。

前述「3.3.2原因と対策」でのカラーセンサ機能でRGB値の内のR値G値、特にR値が影響を受けることを考えると、各値にゲインを設定し適宜調整を行うことでこの影響を軽減できる。

これは更に検証の必要がある。

## 走行速度の壁

### 問題

バランス制御中に走行速度を上昇させる中で、走行速度指定が６０を上回るころから走行速度が上昇しなくなる。

優勝候補チームと比較しても、走行速度のMAX値が明らかにあちらの方が高く、解決可能かつ解決必要性の高い問題の一つである

### 原因と対策

API機能によりバランス制御を取る際、自身の走行速度が角加速に影響することを考慮し、API機能としてバランサーが走行速度を制御している。

「balancer\_param.c」内「K\_THETADOT」を上昇させることで、この制限を緩和できる。

しかしこれによりバランス制御能力が低下してしまう次の問題が発生する。

#### バランス能力の低下（走行速度上昇）

##### 問題

走行速度確保の為にバランス制御の内部パラメータを操作したところバランサー性能が劣化してしまった。

尻尾による支えのある状態からバランス走行に移行するところまでは可能だが、走行速度の指示を６０程度に上昇させた際、一定の走行速度に達した後に転倒する。

##### 原因と対策

指示速度の制限を緩和したことで速度指示６０以上時のバランス制御が低下し、自身の急激な速度上昇自体が転倒要素となり、その要素を吸収できずに転倒している。

自身の速度変化を転倒要素とさせない為に、６０以上の走行速度を指定する際は、急激に指定速度を変更するのではなく、現在の指定走行速度から徐々に指定走行速度を変化させるようした。

ただ走行速度１００から０のような指示を行う際はこの対策を取らなくても転倒しないことから、対策の範囲は「低速→高速」のパターンのみで、「高速→低速」の際には必要ない。

## 尻尾の破損（１回目）

### 問題

２０１５年活動中、尻尾のモータの内部ギアを破損させてしまった。

テスト中に尻尾を９０度位置にして脱力した走行体の尻尾を作業者が手で0度位置へ尻尾を戻そうとしたときに発生。

### 原因と対策

作業者がモータの許容を超す力、または回数モータを手で回転させた為モータの内部ギアを破損させた。

走行体を完全停止させた際は直前の操作を継続して停止するので、再度走行体を起動させる際は例えば0度以外の位置にある尻尾は原位置である0度位置に戻してから起動させなければならない、この為テスト内容によっては頻繁に尻尾モータを手で一方箇所・一方方向に回転させた。

この為、尻尾モータを原位置に復帰する動作を作成し、走行体完全停止時前にこれを実施するようした。

## 尻尾の破損（２回目）

### 問題

走行体を机上にて耐久テストを行っていたところ、テストを終了し走行体を完全停止させようとBluetootchによるコマンド操作を行った際、送信するコマンドを誤り本体の走行用モータが回転し、机から墜落した際に尻尾を床に衝突させモータの内部ギアを破損させた。

### 原因と対策

机の上に走行体が墜落可能な不安全な状態にあった「不安全状態」を解消する為に、走行体を机に置くときは机の真ん中、走行対称面を下に向けて置くようにした。

## API「GyroSensor.getAnglerVelocity()」の未解決問題

### 問題

プログラム処理中に不規則に処理が停止したような挙動を取る。

特に走行体は起動しているが、走行中でない状態のときに顕著である。

### 原因と対策

ジャイロセンサ値を取得するメソッド「GyroSensor.getAnglerVelocity()」を行う際、走行体が走行中でないような角加速が加わらない条件下であると、処理終了に１sec近くの時間がかかる異常が不規則に発生する、タイマ割込みのような高優先度タスクに当該メソッドを記述していると、他の処理を妨げてしまう。

走行体のバランス制御中のような角加速が発生し且つ角加速度の計測が必要な時のみにしか当該メソッドを使用しないようにすることで、当問題は出現頻度を大きく減少させることができる。

## 障害物突破に避けて通れない決め打ち走行

### 問題

障害物突破のために定量を決め打ちで走行を行う際、直線を指定して走行を行うと右ないし左に反れて走行する。

またライントレースを抜けて決め打ち走行に入った際も障害物に対し走行体が正面に向いているとは限らない。

これにより走行計画の計算が合わなくなり、障害物を突破できない。

### 原因と対策

左右モータに全く同じ値を指定してもそれぞれのデバイス的な個体差により左右モータの回転数が異なる。これにより左右どちらかにそれて走行してしまう。

直線走行中は左右モータの回転数を比較して差が発生しないように左右モータの回転指定値を変更することで対応できる。

またコースの直線ライン上をトレースする際に走行体の角度を記録することで、その走行体角度を元に決め打ち走行時の走行体角度を決めれば、障害物に対し正しい侵入方向から突入することが可能である。

## コーディングとパラメータ調整の分離

### 問題

開発終盤は各パラメータ値の調整が大半になるが、その度にコードの書き換え・再コンパイル・書き込み・再起動を行うことは決定的な開発速度の遅延につながる。

### 原因と対策

EV3はファイルシステムによるSDカードへのファイルの書き込みやBluetooth通信によるファイルのやり取りが可能なので、各パラメータを構造体に収めたものをファイル化し、それをBluetooth通信でやり取りすることで高速に各パラメータの変更を行うことが可能となる。

## コンセプトシート再提出

### 問題

コンセプトシート提出後、再提出依頼が大会よりあった。

データはメールで、紙媒体は地区大会時の再提出となった。

### 原因と対策

チームロゴをデザインした際、申請で「チームHAL大阪」としたところを「チームHALオオサカ」とした為に誤った表記とみなされた。

UML設計資料の再提出は任意となった為、コンセプトシートとUML設計資料で扱いが異なる。

コンセプトシートはデザインしたロゴ画像ではなく、テキストデータとした。

## 走行開始時のバランス制御

### 問題

走行開始前の走行体は尻尾による支持で起立しているが、この状態で尻尾を上げバランス制御を開始すると後方に後退してから走行を開始する。これによりタイムロスが発生し、場合によりコースアウトする。

### 原因と対策

本体が後方に傾いた状態からバランスを取り始めるためその傾きを補完する為に後退してしまう。

その為にバランス走行前に尻尾で床を少し蹴り上げ本体を垂直あるいは前方に傾斜させた状態からバランス走行を開始させることで後退しなくなる。

## 走行中のバランス制御のON/OFF

### 問題

バランス制御を一度停止した状態から再度バランス制御を再開させるとバランス走行が失敗する。

### 原因と対策

API機能を使用していない期間のモータの回転数が再開時にAPI内部の記録値とモータの回転数で計算が合わなくなる。

その為、バランス制御ON/OFFの切り替わり時にAPIをリセットし、その時点のモータ回転数をカット値として記録して、再度バランス制御開始時にはすべてのパラメータが初期状態から開始するようした。

# ETロボコンのパターンとリスク

## 試走会の順番待ちと言う空白の時間

### パターン

試走会は2時間ほどの時間に、大会で実際に使用するコースを最大８チーム程で

一斉に使用します。

チーム数に対してコースはLコースRコースを別に考えても２コースしかない為、当然順番待ちが発生します。運営側が特に順番の整理を行ってはいないので、参加者側が適宜譲り合ってコースを使用します。

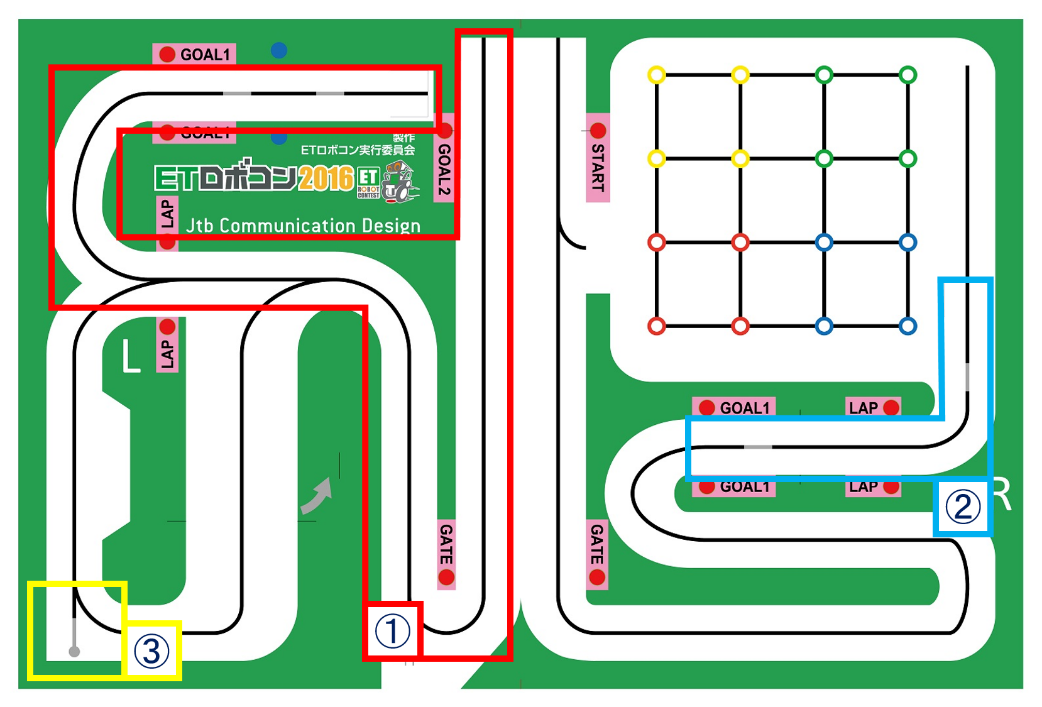
よって、あらかじめテストの計画を立てる必要があります。その為には試走会時の他の参加者の動きを知っていることが有効です。

図 4.1‑1「参加者の動き：2016年度」

1. １コース全てを使用してライントレース―障害物突破を一括してテストするチームです。一見コース全てを占有しているように見えますが、動きが予想し易いのでこのチームが障害物突破中にライントレースのテストを始めてしまうことも可能です。
2. 障害物突破を集中的にテストするチームです。時間を要する障害物突破を繰り返しテストするので、最もコースを占有しているチームです。
3. コースの光量を計測するチームです。基本的に走行コース外を使用していますが、スタート位置での計測を行うこともあります。

### リスク

コースの利用可能状況は常に変動しますので、順序通りにテストが行えることは無いと考えるべきです。

よって、各テスト内容を要素別に分別して常に何かしらのテストをコース上で行えるようにすると良いでしょう。また、各センサの取得値をSDカードに記録して持ち帰り検証することも有効でしょう。

## アナウンスの不足

### パターン

進行の遅れで予定時刻通りに競技が進まないことが多々あります。

その場合は運営側のアナウンスが頼りになりますが、あまり親切なアナウンスではありません。他参加者の動き、特に常連チームの動きを見ることや運営チームに直接問い合わせることが必須です。

### リスク

大会中は全チームが一斉に動くので、アナウンスを聞き漏らしたとしても他に合わせて動けば変更時間に対応できますが、試走会ではチームの動きが読み辛いのに合わせて、次の会場使用が詰めて予定されている為に時間の融通がつくことも無いので、貴重な試走会の機会を逃すことになりかねません。

試走会の入退場に関わる時間変更があった場合は随時、直接運営に問い合わせるようにして下さい。

## 常連はなぜ有利か

### パターン

常連の有利な要素とは、

・活動計画が既に練られている

・過去の成果物が受け継がれる

・会場に慣れている

・運営に頼らずとも大会の流れを把握した人間がいる　　…etc

が考えられます。

特に早期の活動計画の立案は参考にするべきでしょう。他チームがSNS上にアップロードしている活動記録の収集などは、大会の開催発表前から行えると制度の高い計画を早期に行えるでしょう。

### リスク

常連の動きの最も脅威となるところは、過去の成果物の受け継ぎでしょう。

これにより開発日程を大きく削減し、開発の中で特に重要な要素にのみ注力することを可能にしています。

今年度より我がチームも成果物の受け継ぎを行える状態になりましたが、それに慢心せずに開発内容の優先順位や負荷を分析し、適切な活動計画を立てることが必要です。

## 運営が歓迎するショートカット

### パターン

2015・2016年度ともに常連チームがコースのショートカットを行っています。

特に2016年度では主催者側がショートカットを評価する発言をしています。

通常ショートカットを行うにはランダムに配置されるオブジェクトへの接触がリスクになると考えられましたが、これにより運営側がわざとショートカット可能なコースを設計し、オブジェクトを配置している可能性を示唆しています。

### リスク

ショートカットが公式となれば、通常通りにコースの走行を行うチームは走破時間の上で当然不利となります。

つまりショートカット機能の組み込みが必須となれば、ラインを無視するショートカットの実現の為に大会コースによるテストが絶対的に必須となり、コースの用意の可否が直接的に順位に関わる危険があります。

## 部品忘れ

### パターン

大会では車検が行われ、組違いや部品の破損があればその場での部品交換が必要となります。部品の貸し出しは行われない為、対象の部品が無ければ出場は不可となります。

他チームからの部品の貸借は可能なので、稀に他チームにパーツを借りるチームもあります。

### リスク

注意すべきは組違いよりもパーツの破損です。我がチームも大会参加2年を終えましたので、そろそろ各パーツの破損があってもおかしくありません。

プラスチックパーツのヒビ割れや電子パーツの動作不良には注意が必要です。

## UML資料の評価

### パターン

UML資料の点数配分は走行タイムと同等とされています。

UML資料の評価基準は大会提示資料の中に記されていますが、ざっくり言うと

プライマリークラスでのUML評価基準とは、

・設計テーマにそったUML資料であるか

・正しいUMLの使い方がされているか

・UML以外の表記に頼らずに表記しているか

です。

### リスク

計算の詳細さや正しさ表現の奇抜さなどは評価基準にならないようです。

（※PID計算式を説明なしに定石と異なるアルゴリズムを記述しても全く指摘なしでした。）

その為、学校での発表資料作成の様な価値基準とは若干異なる価値基準で資料作成を行はなくてはなりません。

レビューの都度、UML資料の評価基準を見直していつもの資料作成の価値観と混同しないようにしなくてはなりません。

## 試走１・試走２・地区大会での環境光の変化

### パターン

大会運営側は試走会と大会では、会場の環境光を変化させることを明言しています。

関西地区においては会場の白色灯と暖色灯を調整することで環境光の変化を行っている様です。

### リスク

EV3から採用されているカラーセンサの機能を使えば正直言って影響は少ないです。RBG値が等しく変動する白色灯とR値が特に変動する暖色灯を調光することが分かりきっているので、取得RGB値に適宜ゲインをかけて平均化すればほとんど影響がありません。

よってリスクとは

・青色、緑色の調光設備が会場に設置される

・規約上でカラーセンサの機能を制限される

・本大会に進んだ場合の未知の会場設備への対応力不足

となるでしょう。

# ソフトウェア設計思想

## 設計テーマ

### 設計テーマ

「追加・修正の容易な設計」とする。

### テーマ選定理由

2015年度開発時の問題点として開発後期に差し当たって、機能の追加や動作内容の変更を加える度に各クラスの依存関係が複雑に絡み合い、結果として解析不能のコードとなってしまった。

2016年度ではその失敗を活かし、各クラスの役割を明確にして依存関係を最小限に収め、多少の動作内容の変更の際もソースコードの書き換えを行わずに外部ファイルの編集により、変更可能とするものを目指す。

## 現状分析

現状を分析し、競技突破に必要な機能を立案する。

### デバイス能力

EV3のCPUクロック周波数は「」RAM領域は「」と非常に高速であり、またマルチタスクにも対応している。

これらの条件により高容量の変数の宣言・浮動小数を使用した高負荷の計算には特別な制限を設けない。しかし、構造体配列等の大規模な変数の宣言やループ文中に高負荷の計算を行う際は、その変更の直前のバックアップを取った後、十分な動作確認を行った後にのみ使用する。また、その時に作成したバックアップは動作確認が取れた後も破棄しない。

### 制約

ジャイロセンサの値取得には「」ミリ秒、超音波センサの値取得には「」ミリ秒の検出時間を要する。これを超えて連続しての値取得は行えない為、タイマ割込みを用いて連続取得を制限する。（2016年度開発では超音波センサは用いない）

またフィードバック制御にはセンサ値を用いるが、上記制約によりセンサ値がタイマ割込みにより制限される都合上、センサ値更新タイミングと同じ時に行われなければならない。これはフィードバック計算結果によって算出される出力操作も同様である。

この為「センサ値更新」「フィードバック計算」「アクチュエーター操作」は同タイマ割込みタスク内に記述される。

### 条件

マップ無し　会場光量

2016年度は大会コースが用意できなかった為、実際コースを使用したテストがほとんど行えない。その為ライントレース能力には高度な追随性能と、それでいながら大会を高順位で突破できるだけの高速走行性能が必要である。

また、障害物突破時のような一時的にラインの監視が困難になるような状況下においては、マップ上の自身の現在位置を正確に把握している必要がある。

ライントレースを行う上でラインを監視する光学センサはトレース精度の要であるが、環境光は校内でのテスト時・試走会・大会で毎度異なる。

## 開発機能選定

現状分析で立案された機能を評価し開発優先度を点け、実際に開発を行う機能を選別し開発順序を決定する。

コース位置推定機能は開発難度が高いため開発優先度が低く評価されるが、期待効果の高さが無視できないことから、機能を分割し再評価を行う。

また設定ファイル管理機能にはBluetooth通信による設定ファイルの更新機能を付与し、再評価を行う。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 機能名 | 重要度 | 期待効果 | 開発易さ | 得点 |
| 入出力分離 | ◎ | 〇 | ◎ | 13 |
| 高精度ライントレース | ◎ | ◎ | △ | 12 |
| 高速バランス走行 | ◎ | ◎ | △ | 12 |
| 設定ファイル管理 | 〇 | 〇 | ◎ | 11 |
| コース位置推定 | 〇 | ◎ | × | 9 |
| 環境光誤差自動除去 | 〇 | 〇 | △ | 8 |

表 5.3‑1　「マトリックス分析\_開発機能選定」

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 機能名 | 重要度 | 期待効果 | 開発易さ | 得点 |
| コース走行距離類推 | 〇 | 〇 | ◎ | 11 |
| コース走行角度類推 | 〇 | ◎ | 〇 | 11 |
| コース照査 | 〇 | ◎ | × | 9 |

表 5.3‑2　「マトリックス分析\_コース位置推定機能再評価」

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 機能名 | 重要度 | 期待効果 | 開発易さ | 得点 |
| 設定ファイル管理機能 | 〇 | ◎ | ◎ | 13 |

表 5.3‑3　「マトリックス分析\_設定ファイル管理機能再評価」

◎：5点　〇：3点　△：2点　×：1点

## 機能

### 入出力分離機能

### 高精度ライントレース機能

### コース走行角度類推機能

### 設定ファイル管理機能

### 高速バランス走行機能

# 今後の課題

## ライントレース

走行速度は２０１６年度大会で大会突破に十分な速度に達した為、以降は

[1]．「高速走行」と「カーブ用低速走行」の切り替えタイミング。

[2]．[1]を達成するためのコースの把握。

[3]．[2]を達成する手段の確保。

これらが重要となる。特に[3]においては原寸大コースでのテストが有効である。

大会コースが購入できない場合は、大判刷りなどでコースを再現することを早期に考えるべきである。試走会での実証ではテスト精度が著しく落ちる。

## 障害物「ルックアップゲート」

走行体をバックで走行させると尻尾が負荷に負けて尻尾が微妙に上下し、本体を振動させてしまう。これによりタイヤが宙に浮いてしまい正常な走行を行えず、ルックアップゲートの突破と次のガレージのクリアに問題となる。

解決手段として

[1]．尻尾角度保持アルゴリズムの見直し。

[2]．バックせずに方向転換を行う。

[1]は２０１６年度活動で試みたがテスト期間が確保できずに完成していない。[2]は方向転換を行う時間のロスが次のガレージのクリアに影響する。

開発の状況によりルックアップゲートのダブル通過を放棄することも考慮に入れて計画すべきである。

## 障害物「段差または階段」

段差の乗り越えは現状バランス走行のままでも突破は可能だが、スピンを行うと本体傾斜の補完動作とスピン動作が干渉して回転軸がずれて段差からの落下のリスクが高まる。

この問題を回避する為にバランス走行を止め、尻尾支持走行で段差乗り越え・スピンを行うことが大会では通例である。他にはバランス走行のスピンでも回転軸のずれを補完する機能の付与が考えられる。

## UML資料

UML資料の作成自体は提出時には終了しているはずだが、提出書類はテーマに沿ったわかり易いUML資料が必要になる為、単純に全体の設計を記述したUML資料や、チーム内配布を目的とした口頭での追加説明前提の資料では書き換えや作り直しが発生しやすい。

この為、UML提出資料の作成期間は多く取るべきである。しかしUML設計資料の提出時期は開発のピークであることから考えると、開発初期に別途提出資料の作成を済ませておくか、完全分業することを考える必要がある。

実運用のUML資料が完成しているからといって提出資料作成期間を短く見積もると作業計画がひっ迫し、ブラッシュアップの機会を失うことになる。

# おわりに

紆余曲折ありましたが、コースを大会突破県内で走行できるレベルには到達したと判断します。

ライントレース精度・バランス走行速度・設定ファイル管理機能をうまく使えば、現状のソースコードの編集をしなくても、それなりの結果は残せるでしょう。

今から必要なのはそれなりの結果ではなく最高の結果を求める活動です。その為には綿密な活動計画が不可欠です。

特に開発内容の取捨選択に気を付けてください。これからは過去の成果物が利用できます。これを最大限利用し、しかし頼り過ぎずに有効に活用しなくてはなりません。

プログラミングによる開発が一定の目途を見た現状、これから重要になるのはUML設計資料の完成度でしょう。

恐らく活動メンバー全員が初めてUMLを最初から勉強するところから始まるでしょうが、それだけに全員が対等な立場からスタートできるはずです。

このチャンスをうまく生かせば多くのメンバーがこのETロボコンに参加できるはずです。その為にはもしかすると、リーダーとかUML経験者とかをチームに含まない方がより多くのメンバーが参加しやすい環境を作れるかもしれません。

注意しなければならないのは夏季休暇です。例年ここを超えるとメンバーの参加率は著しく減少します。

ここを何かしらの対策を打てたならば、ETロボコン活動としては成功したと言っても過言ではないでしょう。

UMLを用いた開発とは多人数によるシステム開発を前提とされています。それを一人で活動していたとなると得られる学びも本当の学びとは言い難いでしょう。

これからの活動は、沢山のメンバーで学びと結果を残せるものであることを願います。