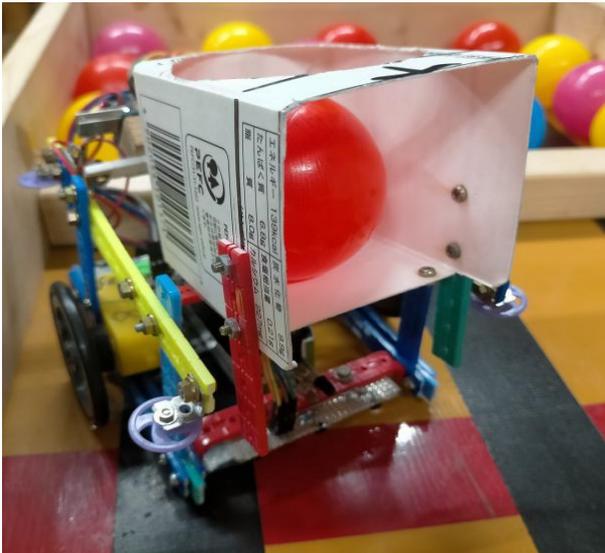


所属団体名	東京都 中野区立 第七中学校 ロボコン部
ふりがな	わっこす
チーム名	わっこす
ロボコンルール名称 (URL https://...)	ワクワク!ロボット定植 ( <a href="https://ajgika.ne.jp/~robo/ru/R5/R5_seigyoo.pdf">https://ajgika.ne.jp/~robo/ru/R5/R5_seigyoo.pdf</a> ) )
製作期間	西暦 2023年 9月頃 ~ 西暦2024年 1月頃
製作時間 (構想から試作完成までの 全ての時間)	90 時間
ロボットに関する 写真と図  必ず、ロボットの概要 や機構等の特徴がわ かる写真や図等を、1 ~4枚程度で掲載し ましょう。  写真や図に記号等を 書き込み、この下の枠 「ロボットのアイデア 概要」で解説しましよ う。	
ロボットの アイデア概要  【報告書要約】 どのような動きを実現 するために、具体的 にどのような素材や 機構を用いて実現し たのか説明してくださ い。	<p>ロボットの動きはプログラム(コード)で表せます。どのようなコードがいろいろな動きを作っているのか、昨年から引き継いだコード、開発したコードを示し、説明します。</p> <ol style="list-style-type: none"> <li>1)モータを回す基礎となる MOL、MOR、BAL、BAR 変数</li> <li>2)ロボットを直進、後進させる FO1、BACK2 サブルーチン</li> <li>3)コードからモータの回転までの流れ</li> <li>4)最高速でカーブするための KAXX 変数</li> <li>5)カーブの方向を決めるサブルーチン</li> <li>6)ライトレースのサブルーチン</li> <li>7)リミッターサブルーチン</li> <li>8)ロータリーエンコーダのサブルーチン</li> <li>9)中央のボールを置きに行くコード</li> <li>10)トラブルリカバー</li> </ol>
参考資料 製作上参考にしたロ ボット等の情報を文 章と URL 等を用いて 掲載しましょう。	<p><a href="https://youtu.be/tTD08HkEwGE">https://youtu.be/tTD08HkEwGE</a> わっこすの動き</p> <p><a href="https://youtu.be/0nO_IsD1DUg">https://youtu.be/0nO_IsD1DUg</a> 先輩の動画</p> <p><a href="https://youtu.be/Rh8NPjuLoiw">https://youtu.be/Rh8NPjuLoiw</a> 先輩の動画</p>

※参考資料が書かれていないなど、未記入の項目がないようにしましょう。

※報告書の2枚目以降にさらに詳しく自由フォーマットで記入しましょう。この表紙を入れて6枚以内で報告書をお願いします。

※この報告書はクリエイティブ・コモンズ表示4.0国際ライセンスの下に提供されます。<https://creativecommons.org/licenses/by/4.0/deed.ja>

## はじめに

このロボットは、「創造アイデアロボットコンテスト・計測部門」で全国決勝で 72 点を取ったロボットです。このロボットのプログラムは、先輩から教わり、引き継いだコード。ロボコン部内で共通で使用したコードが土台となり、その上に左右中央の穴にボールを運ぶコードをチームごとに作成しました。ロボットの動きから、今どのコードを実行中かが分かります。大会では予選ではガタガタでしたが、コードを調整改良したので、決勝では高得点を取れました。それでは「わっこす」のすべてのコード説明します。

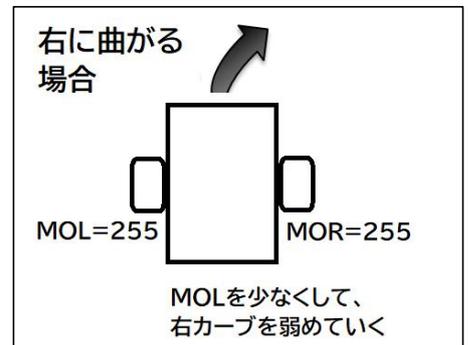
使用制御ボードは「あそぼ〜ど」。言語は Arduino IDE 内で動く ArduBlock でコードを書きました。あわせて 2022 年の報告書「中野七中」を読むと分かりやすいと思います。

### 1)モータを回す基礎となる MOL、MOR、BAL、BAR 変数



変数名の MO は前進回転の略。BA は後進回転の略です。L は左モータ、R は右モータです。前進の回転量は 1~255。後進は-1~-255段階になり、0 から離れるほど速く回転します。左図では MOL 変数に 255 を、MOR 変数にも 255 を代入しているのです、このままモータに出力すれば、全力で前進回転します。同じように、後進には BAL、BAR 変数を使います。しかし、モータにはバラツキがあり、同じ回転段階をモータに出力してもまっすぐ進みません。解決策としては、下

図のように競技前に MOL、MOR に 255 を代入してロボットを前進させてみます。前進させてみて、右にカーブしてしまう場合、右モータが弱いからです。これは左モータが強いとも言えます。そこで、MOL を 255 から減らして 245 に変えて、左モータを弱くして、もともと弱い右の回転に合わせるように MOL の値を調整します。まっすぐ前進させるには手間がかかりますが、この調整はとても大切です。ロボットの動きのコードはすべて MOL、MOR、BAL、BAR の値から計算して作り出しているからです。たとえモータの回転が変動しても、競技直前に直進調整すれば、これまで作成した動き(直進調整済)のコードに対応できます。



### 2)ロボットを直進、後進させる FO1、BACK2 サブルーチン

サブルーチンとはいくつかのコードを一つのブロックにまとめることです。右図では、まとめるためのブロックは上側に、実行形のブロックは下側にあります。

FO1 サブルーチンは全速前進ブロックです。全速前進させるために LLMOTER(左モータへの出力ブロック)に MOL を入れ、RRMOTER に MOR を入れて、左右のモータに前進回転をさせています。

BACK2 サブルーチンは全速後進ブロックです。BAL BAR をモータへ出力しています。

右図の下のコードは実行形サブルーチンを用いたコードです。1 秒間前進し、1 秒間後進するコードです。



### コードからモータへの流れ



### 3)コードからモータの回転までの流れ

全速前進ブロック FO1 の左モータの回転量は MOL の数値で決められます。左図は左モータの説明です。コードで回転方向や回転量を指定し、LLMOTER サブルーチンで、D5 ピンに回転方向の

指定。D9 ピンに回転量をアナログ出力します。

モータドライバはその指定された電圧をモータ端子間に出力します。右モータの場合も同じ流れですが、D6 ピンに回転方向、D10 ピンは回転量をアナログ出力します。回転方向は回転量が正の数か負の数かでデジタルピンの出力を low か high と指定します。

2 つの移動モータを制御するのに D5、D6、D9、D10の各デジタルピンを使うことになります。

次ページに LLMOTER サブルーチンのコードを示します。

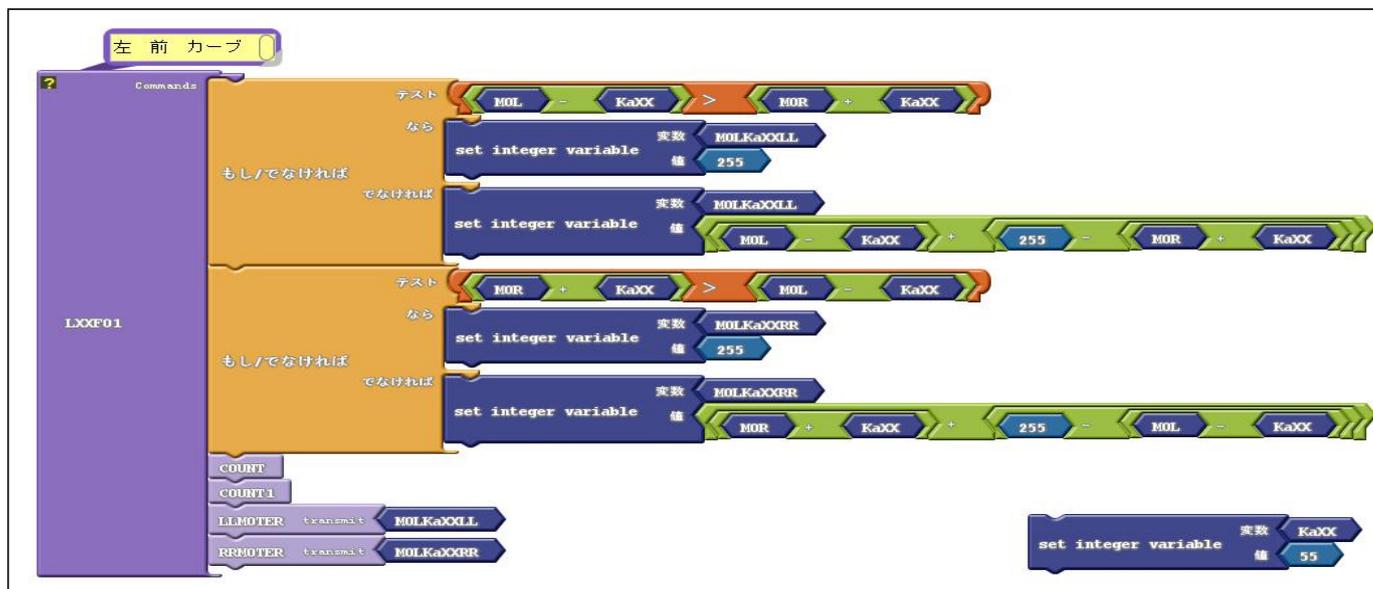


デジタル信号でアナログ出力を 255 段階で出すために PWM 制御方法で出力しています。これは 0V 状態と 5V 状態を入れ切れし、その割合で 255 段階の電圧を作り出し、モータの回転量を制御します。

#### 4) 最高速でカーブするための KAXX 変数

カーブの度合いは KAXX 変数で表します。カーブの種類は左前進、右前進、左後進、右後進の 4 つです。

カーブしてもロボットのスピードを落とさないようにするため、カーブする方向のモータのみ減速させます。下図は左前にカーブする LXXFO1 サブルーチンの

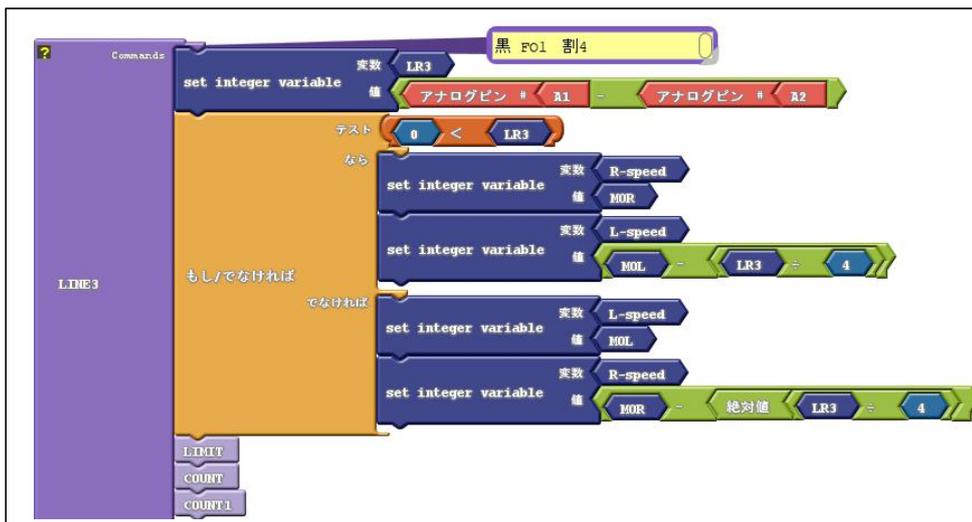


内容です。カーブする右と左のモータ出力の段階値を求めています。

右下の青いブロックはカーブの度合いを決める KaXX 変数を入れるブロックです。このブロックを先に置き、次に LXXFO1 サブルーチンを置きます。これでどのくらいカーブするかは決められますが、カーブをどのくらい続けるかはセンサーのしきい値やロータリーエンコーダの値で決めていくことでフィードバック制御でコントロールできます。

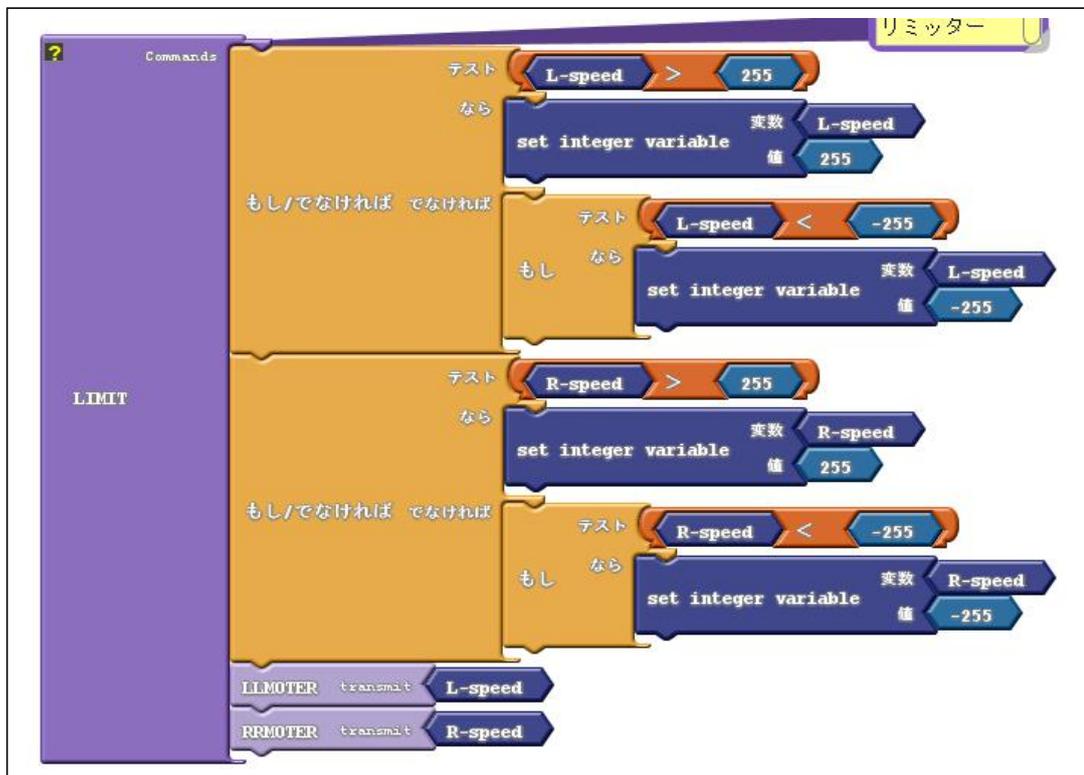
#### 6) ライトレースのサブルーチン

ライトレースのコードは比例制御です。比例制御とは、色のついたラインを 2 つのセンサーで左右からはさむようにして、(左のセンサー値-右のセンサー値)を計算し、その差分をモータの動きに反応させ、ラインを進ませます。反応する感度も調整します。感度が強いと首振り状態となりますが、感度が強い分だけラインからかなりずれていても



強力にトレースに入ることができます。そこで、ライトレースの入り際は感度を強くして、振りトレースをさせて、次に感度を下げて直進性を高めました。また、黄色の床もラインと見立てて、ライトレースさせたり、後進時も黄色の床をトレースさせてスタート地点にもどるようにしました。周囲の壁に触れるとタイムロスがあったり、壁に触れると四隅につけたローラーが壁に当たり、その当たり具合でカーブして壁から離れる時にカーブの度合いがズれてしまいます。ロボットの移動コースは動きの誤差を減らすため壁に触れることがすくないようにしました。

### 7)リミッターサブルーチン



このサブルーチンはモータ出力が-255～255の間から出ることがないように監視するものです。

モータ出力値をさまざまな計算で求める場合、特に比例制御などの場合は想定外のモータ値になってしまい、誤動作の原因となります。そこで、もしモータ値が255を越えても、モータ値が255以上にならないようにコードを組んでいます。

### 8)ロータリーエンコーダのサブルーチン



ロータリーエンコーダは、穴の数を12コに増やし、1.7 cm単位でタイヤのコントロールができるようになりました。

タイヤ円周÷12=1.7 cm  
このカウントは光の当たり始めるタイミングで

COUNT 変数をカウントしています。ロータリーエンコーダ装置は 2023 年の中野七中レポートにあります。

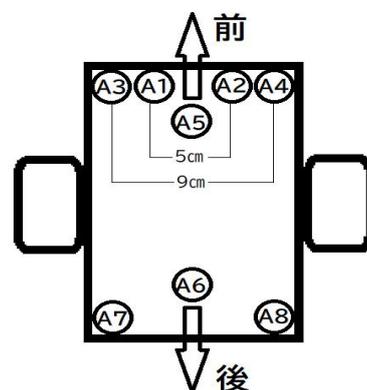
### 9)中央のボールを置きに行くコード

これまで説明したコードを組み合わせた「中央のボールを置きに行くコード」を説明します。

コード作りで大切にしてきた点を説明します。

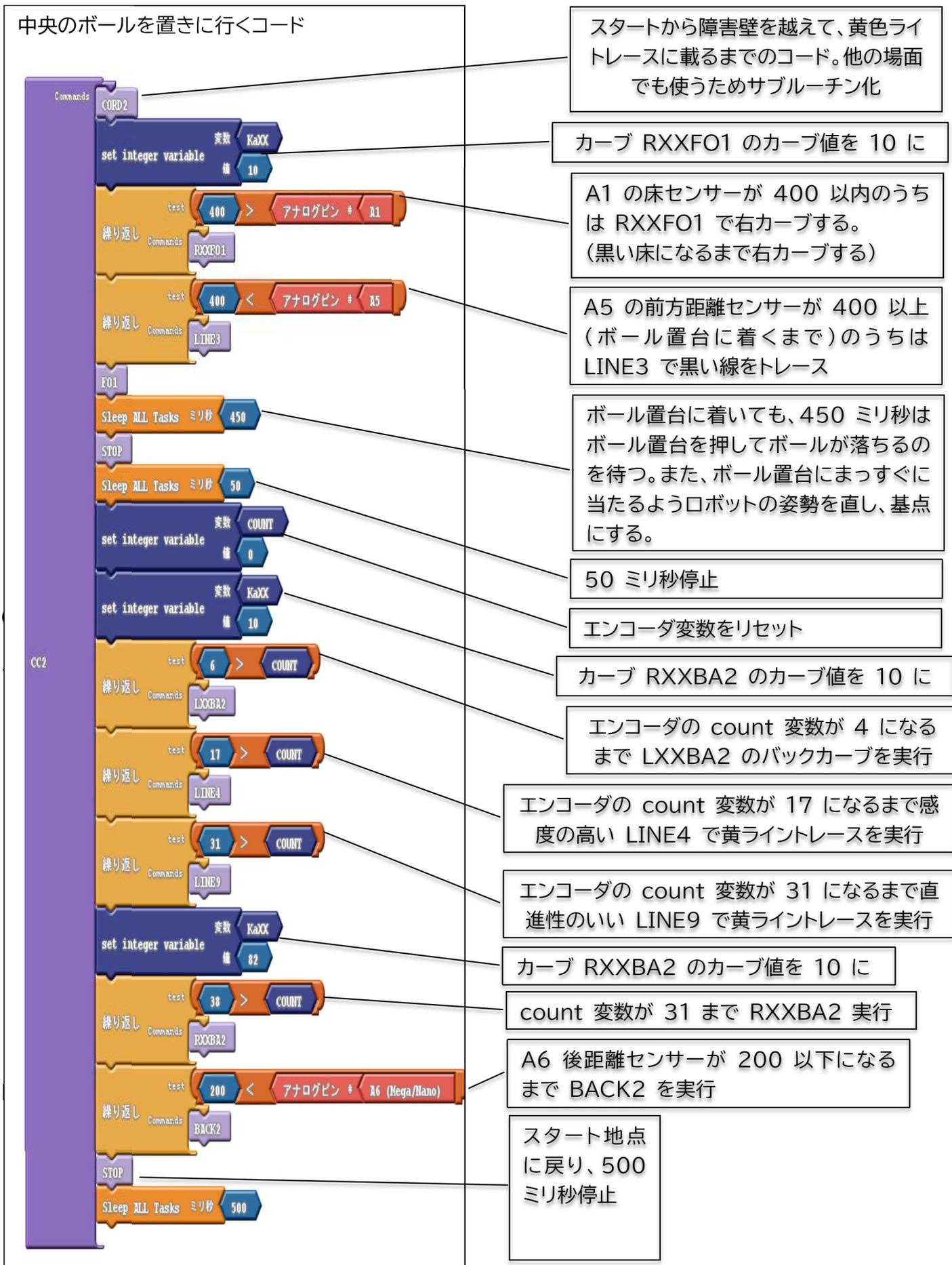
- ・より高速で移動するコードにする。

センサーの配置とピン番号

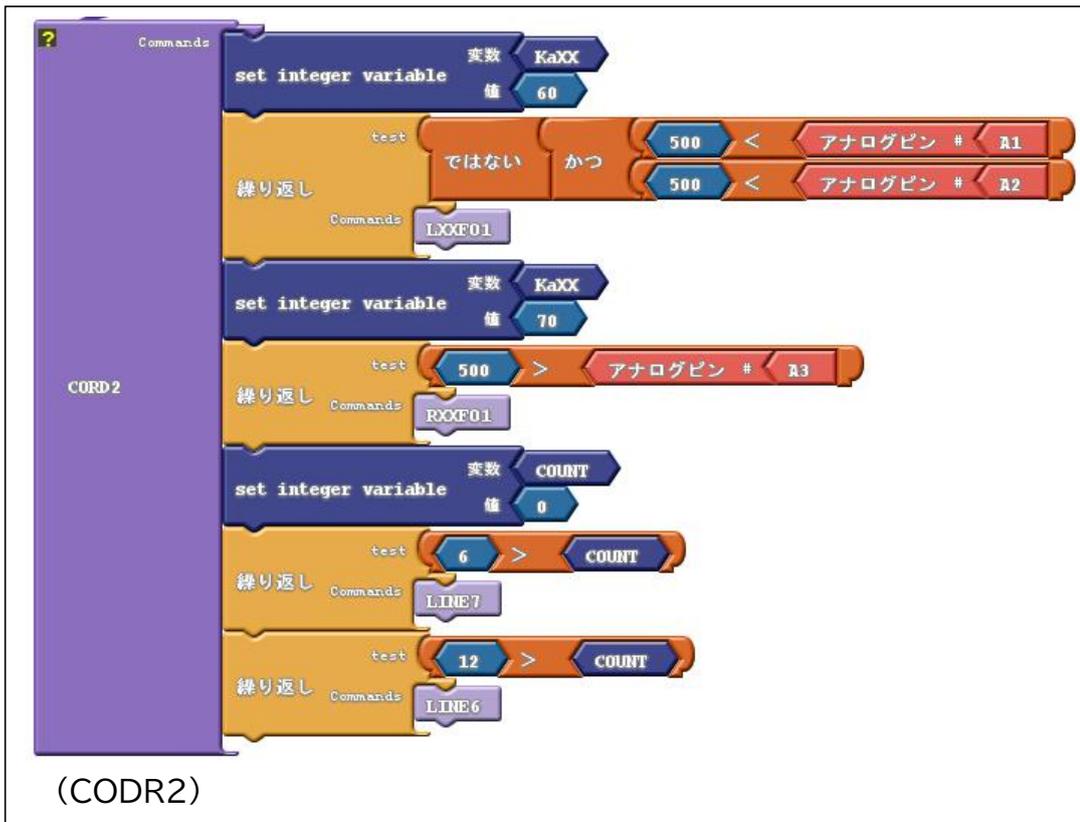


- ・基本はセンサー値やロータリーエンコーダの count 値をしきい値にしてフィードバック制御にすること。
- ・壁にはなるべく触れずに短距離で移動したり、壁に触れることでコースからずれることをなるべく減らす。
- ・動きに誤差が出ないように、コース上に基点を作る。基点はスタート地点、ボールを置いた地点になる。ライントレース中は基点にはならないが、幅方向の位置は基準線となり、誤差を減らせる。

中央のボールを置きに行くコード



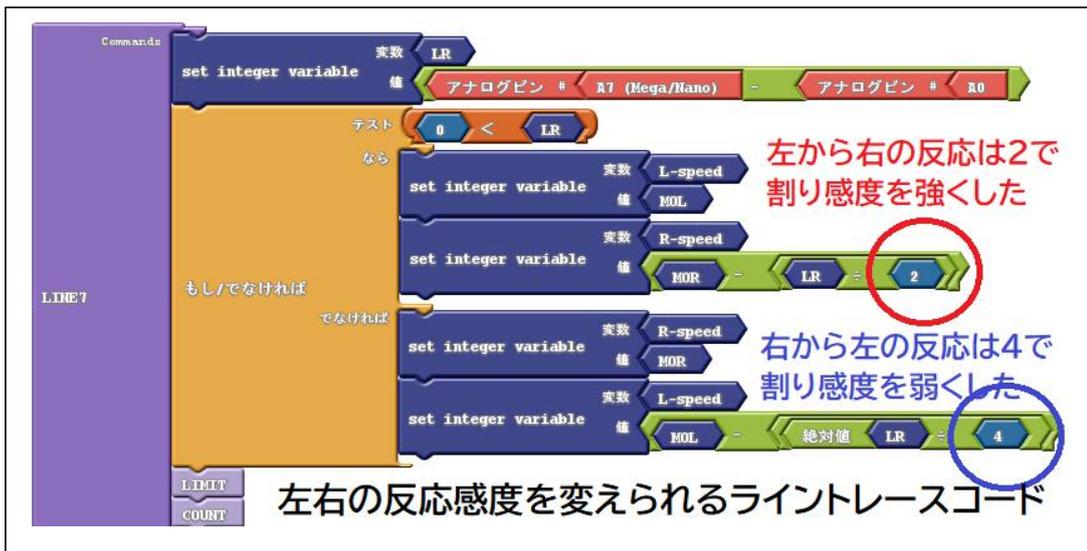
## 10)トラブルリカバー



これは往路で使う CORD2 サブルーチンです。スタート時から障害物板を越えるまでのコースをコード化しました。黄色ラインに強く反応してトレースさせるコードになってます。このコードの目的は、最短コースで障害物板を越えるとともにゴール方向へ正面を向けさせ、右と中央のどの穴にも進んでいくことです。

しかし全国大会の予選で異変が起きました。前ページのコード

実行中の黄色のラインをトレースし、中央の穴に向かうはずなのに、黄色のラインを乗り越えて左に進み、左の穴に向かってしまう現象です。



こちらは LINE7 サブルーチンコードです。LINE7は CORD2 の中に入っているコードで、強力に反応して黄色のラインをトレースするコードです。この LINE7をいじくればいいと直感しました。LINE7のコードは右から左に傾いた時の反応値と左から右傾

いた時の反応値は左右ともに大きいのですが、右～左に傾いたときの反応値だけ鈍感にすれば、左に向きづらくなると予測し、反応量を左側だけ 4 で割り、数値を下げてみたところ、左に行かず、中央の穴に向かうコースに向かわせることができました。これは左右に曲がる反応値を変えることができるという技術を開発したことになります。

最後に

コードを開発して、ロボットが思い描いた動きをすとうれしいし、予期しない動きをすると、なぜそうなるのかと強い関心を持つようになりました。コード開発は無限にあります。いろいろなアイデアをコードで表現できます。私も先輩の残したコードを改良してきました。そしてこの工夫をこの報告書でいろいろな人に見ていただき、その価値を伝えたいと思っています。これで中野第七中学校『わっこす』の発表を終わります。