強化学習を用いたライントレースの高速化

Classic部門 1+1=2 徳満陽喜

上位通過を目指すため、タイムを短縮したい！

去年はそのためにいくつか試してそれらを比較したが、今回はもっと様々なパターンを試していきたい。
そのために今回は最近学んだ**強化学習**（テーブル方式）という方法を使うことにした。

強化学習とは、**AI**の学習方法の一つで、うまくできていた時にご褒美を与え、そうでなかった時には**罰**を加えることで成長させていく手法。AlphaGo(囲碁のAI)などにも強化学習が使われている。

テーブル方式とは、モーターの動かし方としていくつか用意しておき、その動かし方組み合わせが最良なものを探す方式

3周する時間を比較すると…

現在使っているもの　:89.3秒
去年一番速かったもの:51.2秒
AIが考えた速いもの　:61.3秒

ご褒美の与え方や出力のパターン数が少なかったことがあまり速くなかった原因かもしれない。テーブル方式だけではなく深層強化学習（ディープラーニング）も出来るようになりたいのでこれからももっと勉強していきたいと思った。

電池や光に左右されない制御を目指し！

チーム名：APOⅢ（アポスリー）小島昌子　CLASSIC部門

＜背景＞
・時間(wait_ms)を多くプログラムに使用➡練習するたびに結果が違った
・9Dセンサーが中心のプログラム➡周回を重ねるごとに角度が不安定

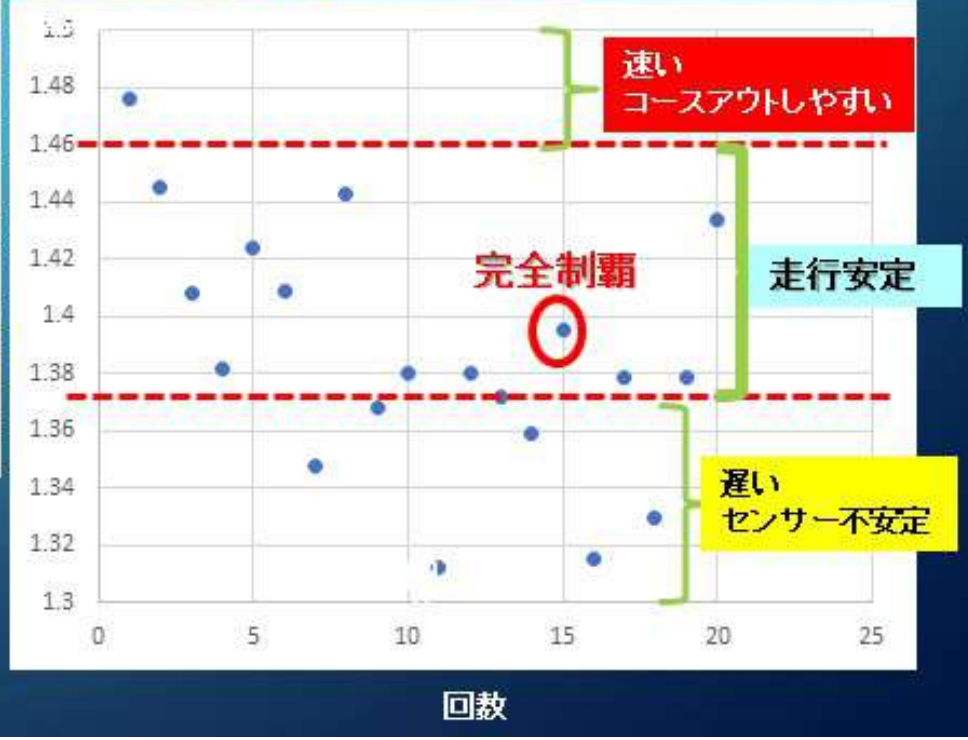
＜考察＞
電池の残量がモーター制御とセンサーに影響しているのではないか

＜対策・実験＞
対策：光の影響を最小限にするため"While if"と"PID制御"のプログラム
実験：周回前に電池電圧を記録、成功した際の値を調べた

周回前に測定した電圧の値(V)

	1回目	2回目	3回目	4回目	5回目	6回目	7回目	8回目	9回目	10回目
1本目の電池	1.488	1.455	1.414	1.327	1.481	1.42	1.331	1.4	1.359	1.381
2本目の電池	1.475	1.446	1.411	1.41	1.41	1.389	1.313	1.483	1.367	1.394
3本目の電池	1.466	1.434	1.399	1.409	1.38	1.418	1.4	1.446	1.378	1.366
合計	4.429	4.335	4.224	4.145	4.271	4.227	4.044	4.329	4.104	4.141
平均	1.476	1.445	1.408	1.382	1.424	1.409	1.348	1.443	1.368	1.380
	11回目	12回目	13回目	14回目	15回目	16回目	17回目	18回目	19回目	20回目
1本目の電池	1.299	1.381	1.376	1.365	1.398	1.334	1.405	1.323	1.405	1.432
2本目の電池	1.323	1.394	1.431	1.319	1.408	1.307	1.316	1.34	1.316	1.427
3本目の電池	1.315	1.366	1.308	1.394	1.38	1.304	1.416	1.326	1.416	1.442
合計	3.937	4.141	4.115	4.078	4.186	3.945	4.137	3.989	4.137	4.301
平均	1.312	1.380	1.372	1.359	1.395	1.315	1.379	1.330	1.379	1.434
					完全制覇					

電池電圧三本の平均



＜物資を運ぶロボット、優先すべきは？＞

たくさんの水や食料などを届けた＝量

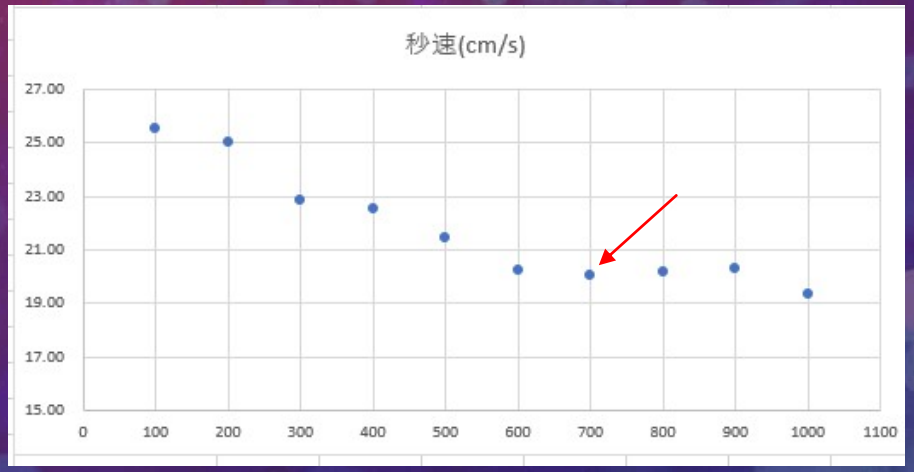
一刻も早く届けたい＝速さ

～重さと速さの関係～

おもりの重さを変えて、秒速を調べる

おもりの重さ(g)	秒速(cm/s)
100	25.56
200	25.04
300	22.88
400	22.56
500	21.48
600	20.26
700	20.04
800	20.18
900	20.32
1000	19.36

※秒速は同じ実験を5回行い、その平均を使用



予想：重さが重くなるにつれ速度は落ちると思う

＜まとめ＞

はじめは重さが重くなるにつれ単調に速度が落ちていっていたが700gを境に一度、速度が上がった。個人的には慣性の法則が原因と考えるが正確には原因はわからない。よって、現時点ではまだ重さと速さの関係の規則性はわからなかった。次回は1kg以上のおもりの重さの場合も試し、さらに重さと速さの関係性について調べていけたらと思う。

中学三年　福村桃菜　Classic

電池残量と3秒間に進む距離の関係

Clover　幸田匠史

動機

　私は今まで充電できないアルカリ電池を使っていましたが、今回の大会から充電できるニッケル水素電池に変えました。
　そこで、充電式の電池では残量によってロボットの進む距離がどのように変化するのか調べました。

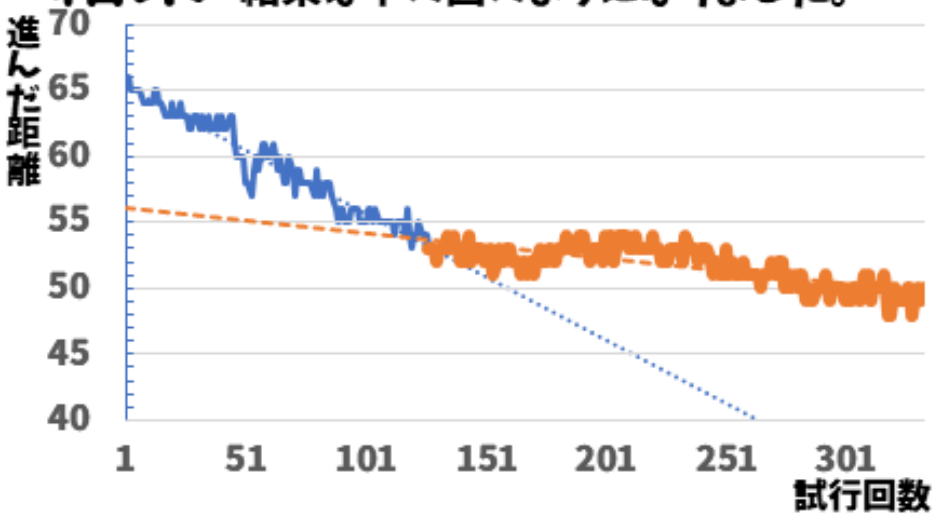
実験の説明

　次のような手順で実験を行いました。

① 電池を完全に充電する
② 3秒間前進させて、進む距離を測る
　（モーターの出力は左右ともに50%とする）
③ ②を繰り返す

結果

結果は下の図のようになりました。



考察・結論

　途中から進む距離の変化が少なくなっている(グラフ中の橙色の部分)ので、速さが安定していると考えられます。
　今回の実験で、電池を完全に充電したときよりも、ある程度電池が減っているときのほうがロボットの進む速さが安定することがわかりました。

CONCON

立命館慶祥中学校 2年 岡田 結衣
Classic出場
プログラミング歴：1年半
前回の成績⇒プレゼンベスト16

電池の温度と動く速さには関係があるのか？

昨年、大阪で行われたSRC17全国大会。
予選でできたことが出来なかった。
大会での大阪と北海道の気温差は 約20℃！
温度が変わることで、ロボットの速さはどうなるのか？

【予想】

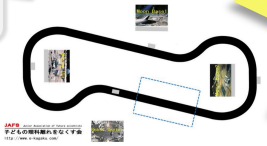
寒さは遅くなる原因か？

-5℃　寒すぎると遅くなる
15℃　安定する
35℃　暑すぎるともっと遅くなる

暑さはロボットにも悪いのかも？

【結果】

ライントレース5回分のタイムを計測！



	-5℃	15℃	35℃
1回目	28	25	23
2回目	27	24	22
3回目	26	24	22
4回目	27	23	23
5回目	26	24	22

35℃が一番速い！

ライントレースの平均時間（秒）

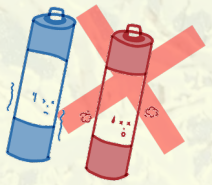
まとめると

どんな温度にも適応できるプログラムに！

温度の変化を小さくする！

秒数指定を
なるべく使わない！

電池の温度変化を
最小限に！！
→移動の際はケースを工夫



ラインセンサーの値の変化

なぜこのテーマを選んだか。

予選で練習ではできたのに本番ではできないことがあった。
→練習と本番のコースがある部屋が違ったのでそれぞれ部屋の向き、窓の位置が違い、
部屋の環境が全くちがった。環境が変わってできないということはラインセンサーの値が変わったから？
と思い、『どういう環境でラインセンサーの値に変化が起こるのか』ということを詳しく調べてみようと思った。

① 電気の種類とラインセンサーの値

電気には主に 3 つの種類がある。

1、　電球色・・・一番暗く暖色系の色。
2、　昼白色・・・太陽の明るさに最も近い色。
3、　昼光色・・・白っぽく青みがかった 1 番明るい色。

この 3 つの電気の種類に着目し白、黒、銀のセンサーの値をそれぞれ 10 回ずつ同じ場所で測った。

⇓

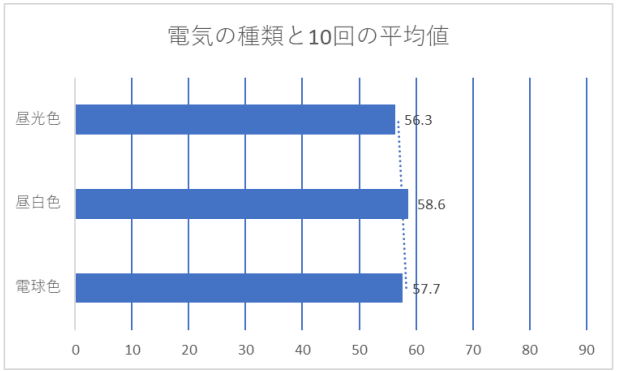
白、黒、銀どれもほとんどラインセンサーの値に変化はなかった。（グラフ 1）

② 日光の強さとラインセンサーの値の変化

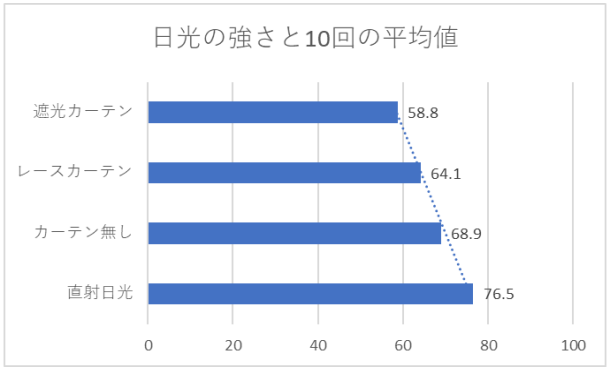
直射日光、窓あり、レースカーテン、遮光カーテン、の 4 つの違う環境でラインセンサーの値を測った。

⇓

黒と銀の値はどれも変化しなかったけれど、白の値は日光が強くあたるほど高くなることがわかった。（グラフ 2）



グラフ 1



グラフ 2

この 2 つからわかること

1、　電気の色や種類、明るさなどは、ラインセンサーの値に変化をおよぼさない。
2、　日光が強くあたるほどラインセンサーの値が高くなる。

⇓

これからロボットを動かすときは、直射日光が当たっているかをチェックするようにする。

ほんだ　こうすけ
Esperanza　　本多　功典

初めまして私は古川裕人です

私は小学生５年生です。４月から小学生６年生です。
私のチーム名はHです。チーム名の由来は名前の頭文字から取りました。福岡市科学館のSSJでロボットプログラミングを半年ぐらい勉強してSRC18Classic第1次予選で2000点以上取って
第二次予選まで来れました。今回の大会で頑張った３の事は、１つ目は秒数を使わなくてセンサーを使うことで安定して走れます。2つ目はなるべくスピードを速くしたことでタイムがはやくなります。３つ目はなるべく荷物を取ることで高得点を取れます。

これが私のロボットです

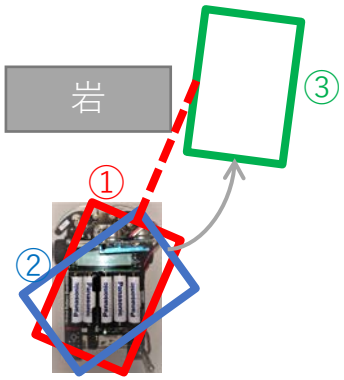
HARUTO

六甲小学校3年生　梶　陽人

- 使うロボットはCキュービック
- センサーはライントレース用の赤外線センサーと、岩よけ用の超音波センサー

- 1次予選では3100点を取った
- 完全制覇（せいは）は難しかったので、ボーナスポイント（岩よけ、３周で完全停止）で点数をかせいだ

- ボーナスポイントの岩よけで工夫したこと
  - ①岩よけでは、超音波センサーを使って、岩がなくなるまで回る
  - ②そこからまっすぐ進むと、岩にぶつかってしまうので、さらに0.2秒回る
  - ③カーブしながら進む



日光とライントレースの速度の関係

動機

ラインセンサーは、周りの環境や光量によって値が変化するため。

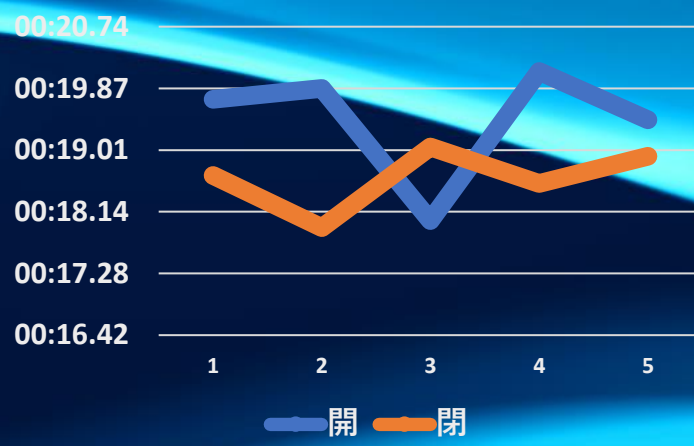
実験内容

カーテンを開けてコースに日光を当てた時と閉めて日光を当てない時の完走時間の違い。

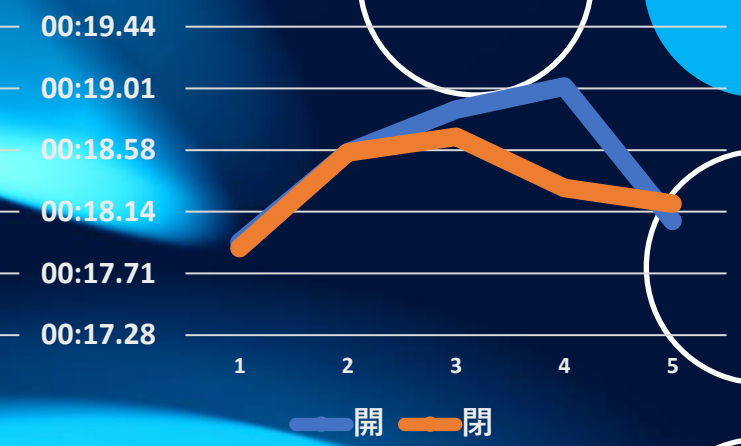
結果

- 12時カーテンを開けた時、完走できなかった
- 14時・16時は完走できたが、どちらも閉めた時の方が完走時間が短かった

《完走時間》 14時



16時



まとめ

自宅でのコース設置をする時は光の調節が重要

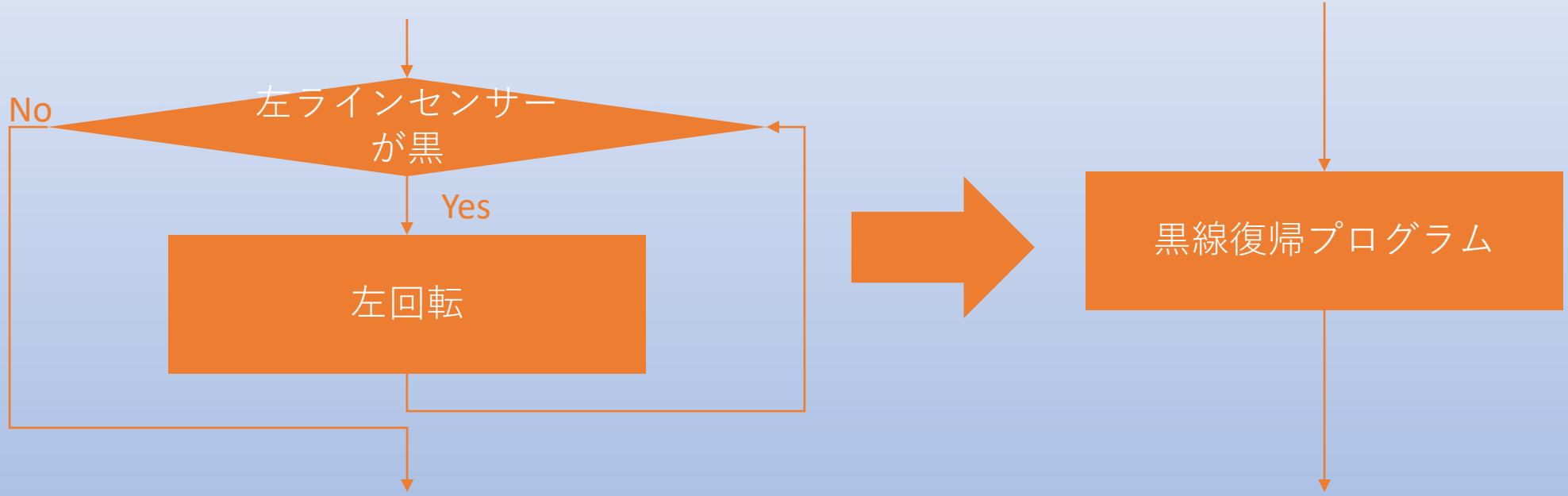
日光の強い光でロボットの動作が変化するため、カーテンを閉めることでライントレースを速くすることができる

Hi-ro- 川口大輝 Classic

黒線復帰プログラム

Hydrogen
Classic

今回、どの角度からでも黒線復帰できる黒線復帰プログラムを作成しました。



このように黒線復帰プログラムを使うと楽にプログラム制作が出来ました。

チームKDM

名前　工藤　優拓
学年　小学5年生

工夫したいこと

　サブプログラムなどを使い、見やすいプログラムにすることで、問題が起きた時にプログラムを書き換えやすくして、はやく修正できるようにしたいです。

　電池を拡張して、より長く電池が持つようにしたいです。

　前回の大会で、サーボモーターがこわれてしまったので、アームを軽くして、モーターがこわれないようにしました。

電池を拡張した後とする前で、速さはどれだけ違うのか

条件

電池の本数を、３本と５本にしてひょうたんコース一周にかかる秒数をしらべました。

予測

電池が５本のほうが３本のときよりもはやくなる。

結果と考察

2倍ほど速くなっていたので、電池を少し増やすだけでもロボットの動きに影響を及ぼすことがわかった。

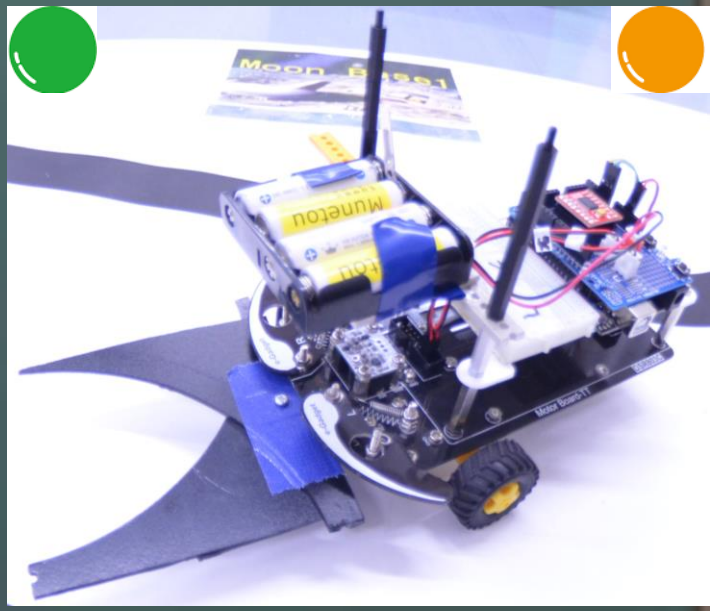
	1回目	2回目	3回目	平均
3本	16.16秒	15.8秒	16.15秒	約16.04秒
5本	33.28秒	34.86秒	34.77秒	約34.3秒

チーム名　Kouno試行　名前　河野雄大　SRC出場部門　Classic

所属　明法高校　学年　1年生

限りなく失敗をゼロへ

ALLPERFECT を目指したロボット製作



●Arduino 使用のプログラム製作

SRC　Classicは、過去にも複数回出場したことがあったが、その時使用したプログラミングのシステムは、C-cubicだった。しかし、今回は、数字や英語を打ち込んでプログラムを製作するArduinoを選択した。書くのが大変そう、と思う人も存在するのではないか。でもなぜArduinoにしたのか。

利点①　細かい秒数調節可

C-cubic→0.1秒単位まで調整可

↓100分の1の細かさ！

Arduino→0.001秒単位まで調整可

→秒数指定を必要とする際に
　動作の継続時間を正確に設定

```
delay(1000);  //←1秒間
delay(100);   //←0.1秒間
delay(10);    //←0.01秒間
delay(1);     //←0.001秒間
```

利点②　電池個数調整可

C-cubic→本体の3本の電池にパワーアップモジュールを付けることによって電流を上げられるが5本一気に追加することになる。（1，2，3，4本だけ追加はできない）→制御難しい

Arduino→電池ボックスの電池の個数で調整

3,4,5,6…の物が存在

→個数調整しやすい

→ちょうどいい出力

●コンテナを取る際の角の工夫

角→学校に置いてあった塩化ビニル板の切れ端（廃材）で作成

少しでも捨てることなく再利用

12 つくる責任 つかう責任

SDGsへの意識

SRC で頑張った３つのポイント

チーム名：LEVEL1　出場カテゴリー：(classic)　名前：古川 清玄

１．ひょうたんコースからＳ字コースに渡る部分のプログラム

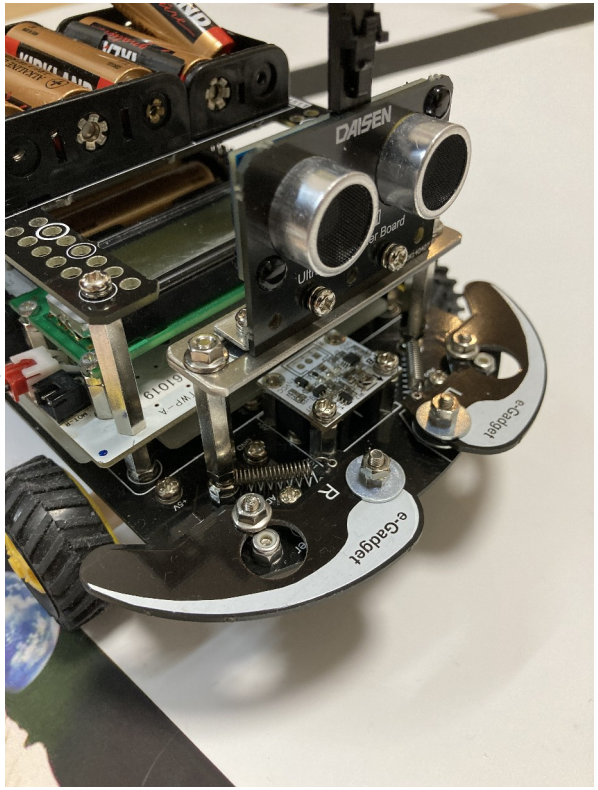
ひょうたんコースからＳ字コースに渡る場所では、ちゃんと銀を狙って３つのセンサーがどんなふうに反応しても、渡ることができるようにたくさん if 文を書きました。

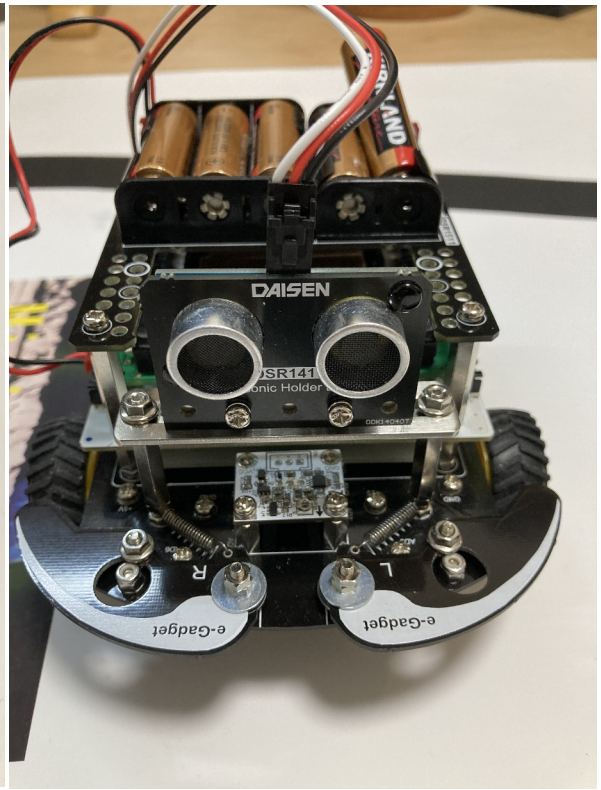
２．たくさんの変数を使うプログラム

普段使う変数は１個や２個程度で、大会で使うプログラムはたくさんの変数を使うので、初めはよくわからずにとても苦労しました。変数の使い方をエンジニアのお父さんに教えてもらって、銀を見つけた数を数えたり、ブロックを運び中かどうかを判定したり出来るようになりました。

３．超音波センサーの位置

ロボットに最初からついている部品では、超音波センサーを真ん中につけることが出来ませんでした。ホームセンターに行ってしっかりとした金属の部品で、ロボットにくっつけられそうな部品をを探して、なんとか真ん中に取り付けることが出来ました。





大会が終わって思ったこと

今回の大会でたくさんの新しいプログラムを知りました。学んだことをこれから先、活用していけるように頑張ろうと思います。そして、学校で習った四大公害病の元になったような環境問題を、いつか自分の作ったプログラムとロボットで調査して解決したいです。

SRCクラシック二次予選Mr.Planeのプログラムについて

明法中学校　細川昊資

二次予選の工夫

僕が二次予選で工夫したいことはWhile ifを使うことと、できるだけ最短コースでゴールを目指すことです。
最短コースを目指す理由は電池の残量を長持ちさせることです。
While ifは電池の残量によってプログラムの動作が変化することを防ぐために使います。

最短コースを目指すために

僕は今まで、確実にコンテナ回収やコンテナ置きをするために普通のルートではなく少し特殊なコースにロボットを走らせていました。しかしそれを今回はできるだけ特殊なコースを少なくして最短コースを目指したいです。

S字コースについて

前回僕はS字コースに行けずに終わってしまいました、なので今回はS字コースに行けるように頑張りたいと思います。
どのようにS字コースに行くかというとWhile ifを使ってS字コースの線にたどり着くまで進むというプログラムを作ってS字コースに行きたいと思います。

SRC18 classic

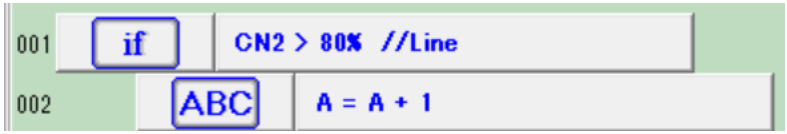
吉岡 暖人

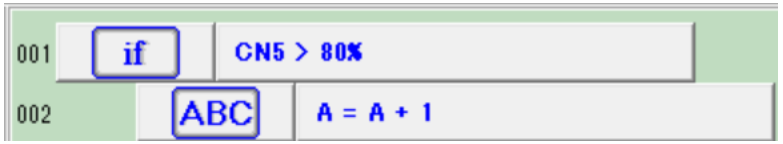
My name is Yoshioka

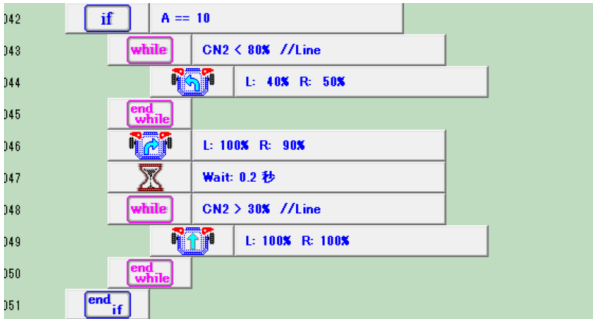
大会の抱負

僕は、今回のSRCで予選突破を目指して頑張ります。

僕が前回の予選で1番考えたプログラムはこちらです。







このプログラムでは、中央が銀の時と、左が銀の時に、センサーが反応し、数値が1増えるという仕組みになっています。こうすることで、3周目のS字に行く際に、Moon Base1のところでロボットのスピードを下げることができ、安定してS字コースに行くことができます。

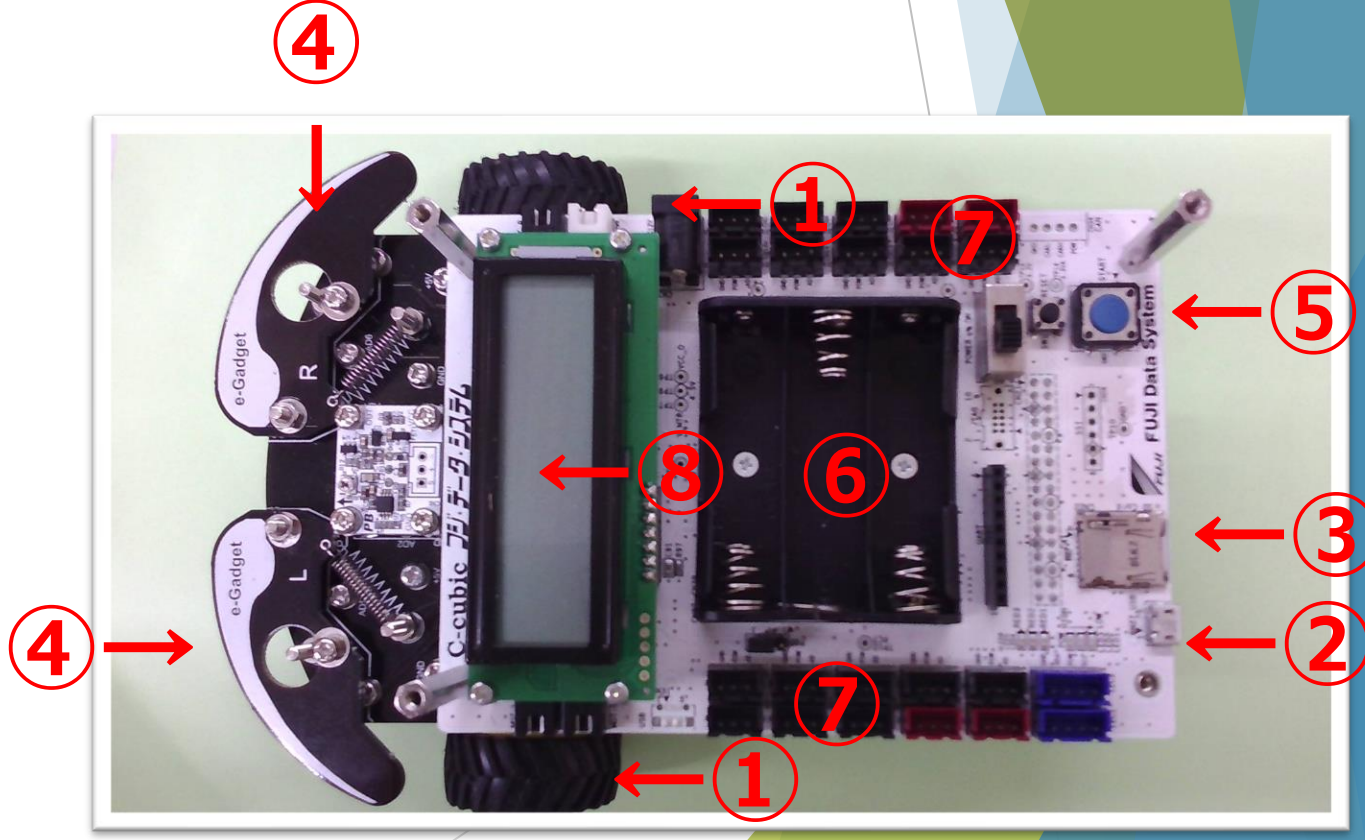
ありがとうございました。

Team Nekoyouray

立命館慶祥中学校
一年一組　工藤伶

ロボットの解説

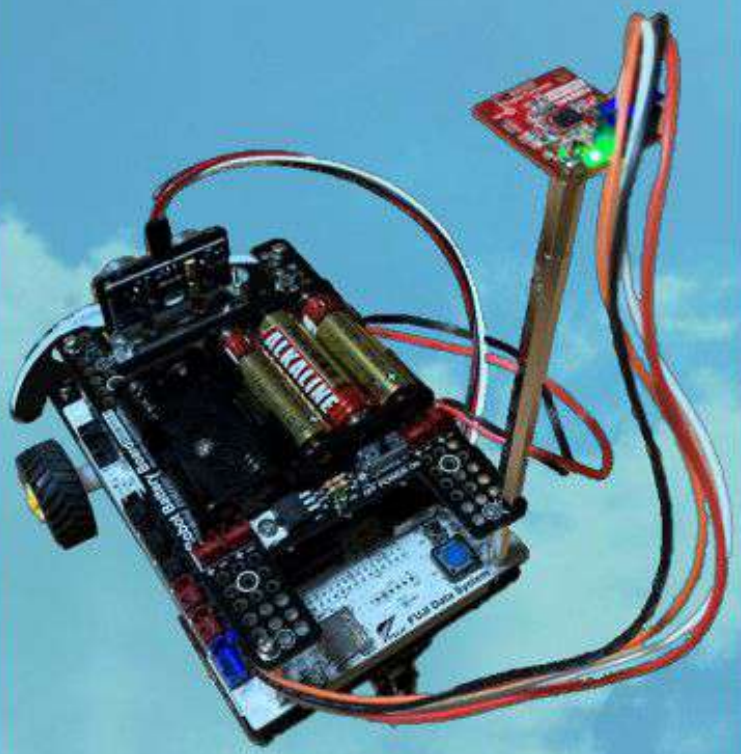
①タイヤ
②USB差込口
③SDカード差込口
④タッチセンサー
⑤スタートボタン
⑥電池ケース
⑦追加するセンサーやモーターをつなげる場所
⑧モニター



PISTOLSTAR

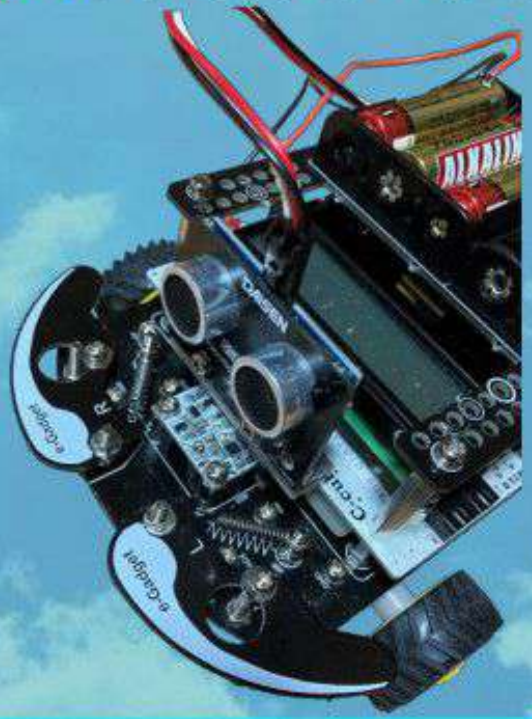
頑張ります.

今回使うロボット



超音波センサーの改良

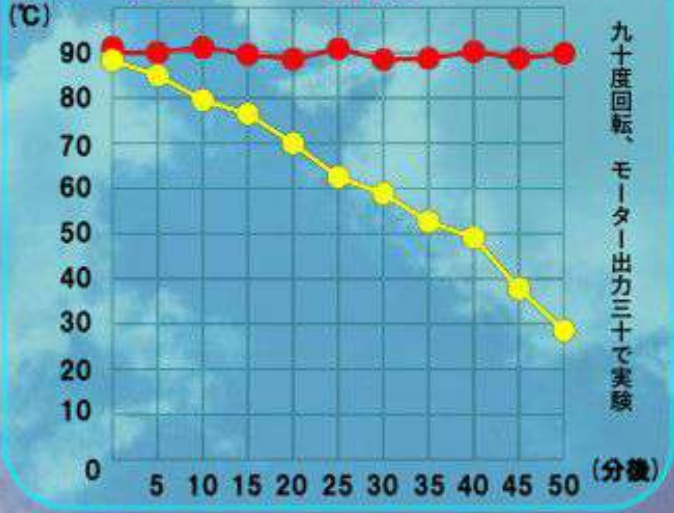
超音波センサーの設置をできるだけ中央かつ低くすることにより、2か所の固定でより安定して、多くの場合に対してセンサーが使えるように改良しました。



方位センサーを利用した回転プログラム

方位センサーを利用して、一定の度数の回転をより安定して行えるようにしました。

下の図は秒数で指定して回転したものと、
方位センサーを利用して回転したものを比較した実験です。
時間がたっても赤はあまり変化していないことが分かります。
電池の消耗に左右されないので、安定した走行ができます。



PUMA

SRC18　Classic

榊　亮太

(今大会で工夫したところ)

・電磁残量をはかりどの電池残量での走行がうまくいくか調べた。

理由
前回、新しい電池で走行したときにロボットのコントロールがうまくいってなかったから。

（分かったこと）
電池残量が変わっても秒数は変わらないが、1.27Vから1.22Vの間がコントロールし易い。

←これで計測しました

（電池残量をはかり同じプログラムで3回ずつ走行した結果の表）

電池残量	時　間	結果
1.53V	1周目：　- 2周目：　- ゴール：　-	1周目でコースアウト
1.41V	1周目：　- 2周目：　- ゴール：　-	1周目でコースアウト
1.33V	1周目：　44秒 2周目：　1分24秒 ゴール：　-	2周目で2回 3周目で1回コースアウト
1.27V	1周目：　41秒 2周目：　1分21秒 ゴール：　2分17秒	完全制覇(3回中2回成功)
1.22V	1周目：　42秒 2周目：　1分22秒 ゴール：　2分22秒	完全制覇(3回中2回成功)

TEAM RAINBOW BELL

明法中学J1鈴木　虹喜

SRC18classic2次予選

最初の行
の変数⇓

	Timer 1: Start
ABC	M = 400 //回転1/4
ABC	L = 100 //モーター左の出力
ABC	R = 100 //モーター右の出力

修正時間5分

短くなった➔

修正時間30秒

¼回転箇所
が全部で
15箇所ある

僕のプログラムは、
電圧が変わるごとに変更が
必要な15箇所の値を全て
変数に置き換えて、1箇所で
全て変更できるようにしました。

チーム名　RT

名前：所　拓真
出場カテゴリー：
Classic

～悔いがないよう、全力を尽くす～

今回行った実験

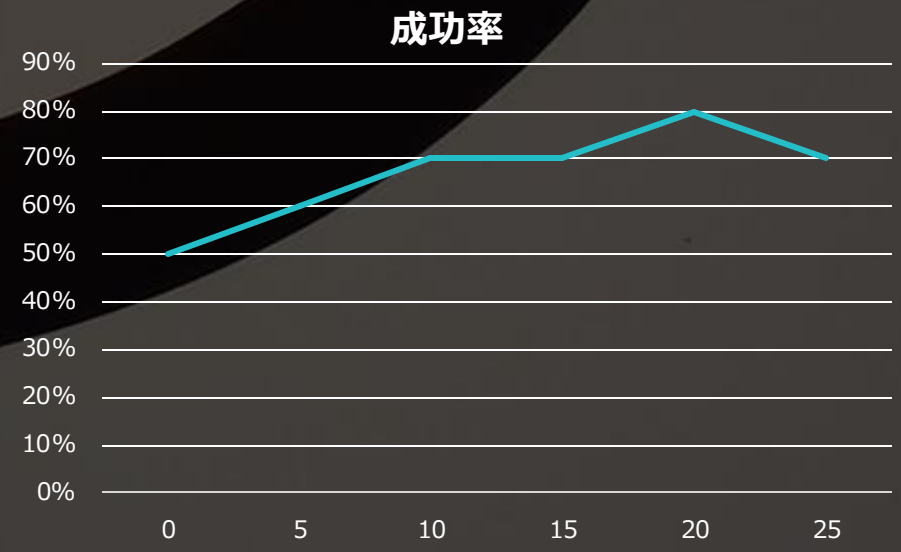
どういう充電池が一番成功率が高いのか

実験方法

充電が完了した後に充電池を**0、5、10、15、20、25分と放置する**。それぞれの充電池で10回瓢箪コースのプログラムで動かして、どれが一番成功率が高いのかを調べる。

実験結果

結果、**充電完了後に20分放置した充電池を使用すると、一番成功率が高い**ことが分かった。



周囲温度とモーター速度の関係

TEAM SAE

名前　　小西紗愛（小学6年）
ロボットプログラミング歴　3年
将来の夢　医師（プログラミングを活用したい）
予選の練習では完全制覇できたが、本番では良い結果を残せなかったので、二次予選を通過できるように頑張りたいと思います。

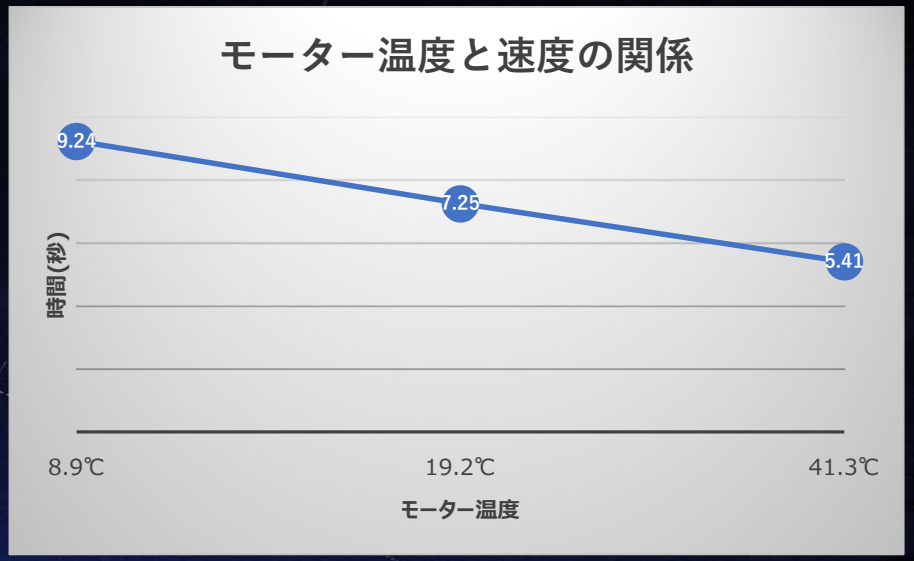
Introduction

周囲温度を変更することでモーター動作温度を変化させ、ロボットの速度に対して温度の影響があるのか調査した。

Method

ロボットの周囲温度を変えることにより、モーターの温度を可変させて、ロボットを一定距離走行させ時間を測定した。周囲温度目標は、10℃,20℃,40℃とし、1.5mの距離を前進（左50,右50）走行を5回行った。
電池はフル充電で、常温のものを使用した。

Result

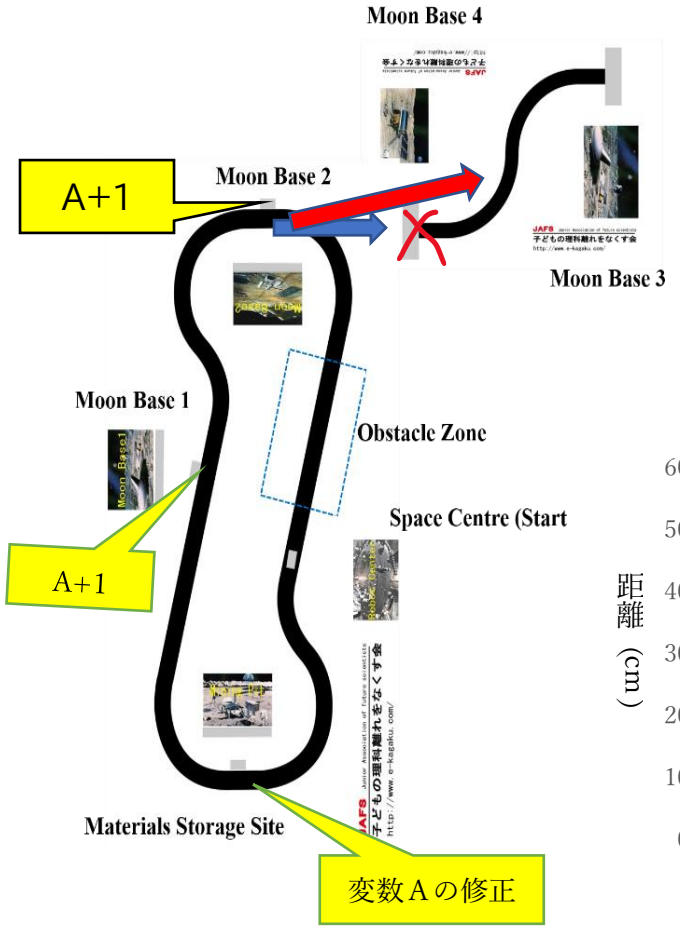


・モーターの温度が高い程、ロボットの走行速度（モーターの回転速度）は速かった。
・各温度で測定した走行速度は、低温時に5回の測定時バラツキが大きかった。
・9～41℃の範囲においては、モーター温度とロボットの走行速度は比例していた。

チーム名　sirokuma

出場カテゴリー　Classic

藤堂　怜那

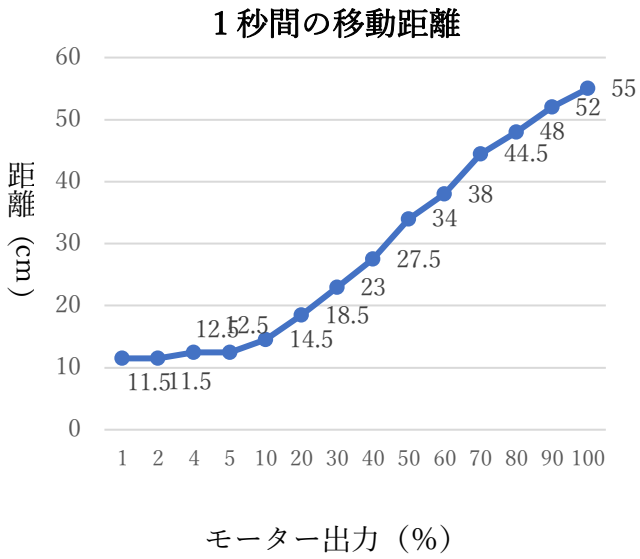


工夫したところ

①　左銀を通った後、タイマーを使って変数Aが1だけ増えるようにしました。もし増えすぎた時やスルーしてしまった時は、変数Aの数を修正するプログラムを入れました。

②　S字にわたる時、銀テープではなく黒ラインに行くようしました。

③　脱線した時のためにタッチセンサーを使って2周目3周目からスタートできるように工夫しました。



発見したこと

モーター出力による1秒間で進む距離を調べました。モーター出力の1～10%は、ほぼ変わりません。10%以上はモーター出力にほぼ比例して距離が伸びます。

TEAM NAME

Sky-Blue!!

明法中学校　J1B-酒井　蒼

SRC18 Classic 明法予選

SRC明法予選を振り返って

反省点

プログラム

本番になってプログラムが上手くいかず、S字コースまで行けなかった。

ポスター

内容がありきたりになってしまった。

改善点

プログラム

もしも失敗したときのためにあらかじめプログラムを複数個用意しておこうと思う。

ポスター

プログラムについて工夫したことをもっと書けるようにしたい。

二次予選も全力を尽くします!!

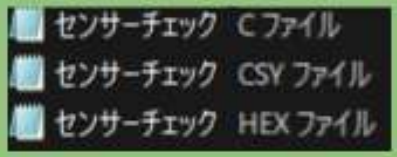
チーム名

ねこねこ

完全制覇出来るように頑張ります!

Cファイルについて

C-Style for e-Gadgetで、プログラムを
ダウンロードした後に保存したファイルを見ると、このような感じに(↓)なっている。
CファイルとCSYファイルとHEXファイルの3つがあるらしい。
今日はこのCファイルなどのことについて調べて見た。



・**Cファイル** C言語で書かれたソースファイル(何らかの処理の入力に使うデータが記録されたファイル)
・**CSYファイル** CSY=Canvas Symbol Formatビルドなどをしないで保存するとこのファイルができる。
CSYファイルが1番大事で、これがないと作ったプログラムが全部消えてしまう。
・**HEXファイル** 16進ソースファイル拡張子ビルドするとCファイルと一緒に出来る。

有名な拡張子 .pdf .zip .txt .jpg .jar .mp3 .mp4 .html .exe .png .cmd

感想「プログラムを支えるプログラムがある。」

チーム名

TERUTO

田中瑛士

Classic

方位センサーを使う回転

①

```
while(!(judge_bno(0,mukuhougaku,10))){
   gV[VAR_A]=get_bno(0);
   a=mukuhougaku%2;
   gV[VAR_B]=mukuhougaku;
   lcd_puts_var2(1,VAR_A,VAR_B);
   if(a==0){
     motor(50,-50);
     wait_ms(1);
   }else{
     motor(-30,30);
     wait_ms(1);
   }
}
```

②

```
int kihonhougaku=190;
```

①のプログラムは回転するときのサブプログラムです。変数「mukuhougaku」が向きたい方向の角度です。また、変数「a」が0だと右回り、1だと左回りになります。②のプログラムはメインプログラムにあります。「190」はコースの方角です。「wait_ms」を使わず、方位センサーを使ったプログラムを使うことで、電池の残量に左右されず **正確に回転する** ことができます。

YUHI

電池の残量が減ると具体的にロボットにどのくらい影響があるのか？

【方法】電池をためて同じ速さ、同じ時間を設定してロボットを走らせ、回数ごとにどのくらい進んだ距離が減るかを調べる

【結果】1回目と10回目を比べると

約4.5cmもの差がうまれた

今回は直進させただけですが、本番ではもっと動かすのでもっと差ができる。

今回の結果から**4回おき**に電池を変えるのがいいということが分かった。

1回目	2回目	3回目	4回目	5回目
76cm	75.8cm	74cm	74cm	73.5cm
6回目	7回目	8回目	9回目	10回目
73.5cm	72.6cm	72.3cm	72cm	71.5cm

Classic　大塚雄日

SRC18Classic　　チーム名：アノマロ　（中村優太）

僕は一次予選にギリギリ通過することできました。

二次予選は自信がありませんがやってみたいと思います。

僕のチーム名「アノマロ」の由来を紹介します。

アノマロカリス

今から５億４０００万年前のカンブリア紀に生息していた動物です。当時最強の捕食者と言われています。僕はこの動物が大好きです。

これからどんどん勉強して立派なプログラマーになりたいです。

よろしくお願いします。

滑らかに走れる後輪を探せ！！

アライグマ　　Classic部門　石山高校1年　小松琢磨

背景

・普段、ライントレースをしているときに、後輪の鉄球がギーギーうるさかったり、滑らかに走らそうとしてねじを緩めると、取れてしまうというアクシデントが起こりその点をを直したいと思ったから。

方法

・後輪をベイブレードのドライバーに変更する。

ベイブレードへの交換方法

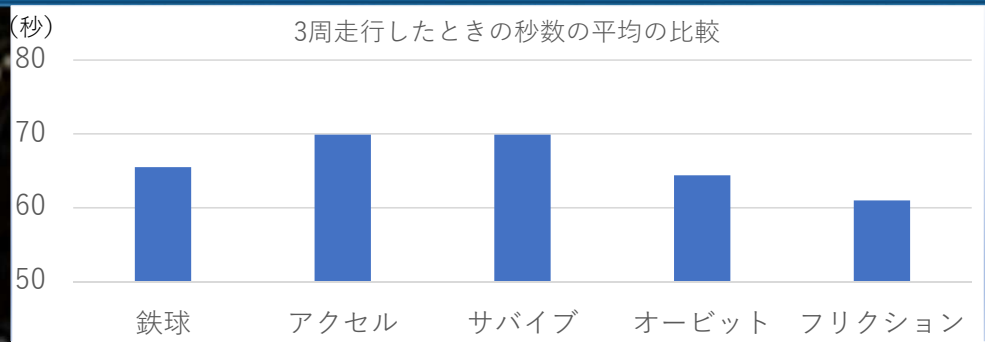
1,レイアーを最小限に小さくなるまで削る。
2,鉄球とそこについている部品をすべて外す。
3,レイアーを鉄球があったところにネジで固定する。
4,レイアーにドライバーをセットする。

ベイブレードとは

・1997年からタカラトミー様が販売している現代版のベーゴマのことである。
・今回使用したのはベイブレードバーストという2015年から続いている第３世代のものである。
・上の部分のレイアー、真ん中の金属部分をディスク、一番下の部分をドライバーという。

比較実験と結果

・ロガーの後輪を鉄球とそれぞれ特徴の違う4種類のドライバーを設置して、自作のひょうたんコースを三周されて、そのタイムを競った。
・フリクションは鉄球の時よりも4.5秒もタイムが速くなり、滑らかに進むことがわかった。



シャインマスカット　Classic　佐藤蒼海

　僕は、このロボットコンテストに出場するにおいて、僕はまず、一次予選では、自分のプログラムにあり再現力が無く、第二走行では全く想定していなかった所でコースアウトしてしまいました。なので、それが自分の日常生活にも関連していないか考え、その点に気を付けて生活したりしました。

そのため、二次予選では、なるべくプログラムに「時間」ではなく、「条件付きループ」を使うことで、その再現力を高めました。

しかし、今回は再現力にこだわり過ぎたせいで、自分の考えを表現するのに時間がかかってしまい、おまけにそれで焦って、起こった問題に対してその一時的な解決策ばかり求めてしまって、正常な判断が下せませんでした。

よって、今後の日常生活では、物事を論理的に考えるように気を付け、もし次二次予選に出場する機会があれば、自分がどれだけの時間があればプログラムを完成させられるか見極めてから、その時間に沿って行動したいです。

チーム せせやめみゆ

立命館慶祥中学校
3年5組　30番
船倉光平

目標

今まで二次予選を乗り越えたことがないので、今年こそは良い結果を残し、三次予選に進みます。

自分のロボット

今回は9Dコンパスセンサー使っているので、正確な走行を目指します。

実験　湿度によってモーターの動きは変わるのか？

概要　湿度の変化によって、モーターの回転速度に違いが出るのかを調べ、今後の参考にしようと思ったから。

方法　風呂で湿度だけ変え、同じ高さから同じ重りを、タコ糸とモーターに括り付けて巻き上げ、その時間を計測する。

結果(結果は秒)

	高い	低い
一回目	8.25	8.3
二回目	8.85	9.2

実際の様子
重りには養生テープを使っている

まとめ　湿度が高いほうが巻き上げの速度が速くなることに驚いたが、これで湿度が高い時の調整の仕方が分かった。

TEAM Strawberry

名前：まみ
学年：５年

努力は必ず実る
～努力は天才を超える～

私が、今回実験をしたい内容は私がいつも使っているこのロボットがどれだけの量を持って運べるのか荷物を運べるロボットを作ってみたいと思ったからです。

内容

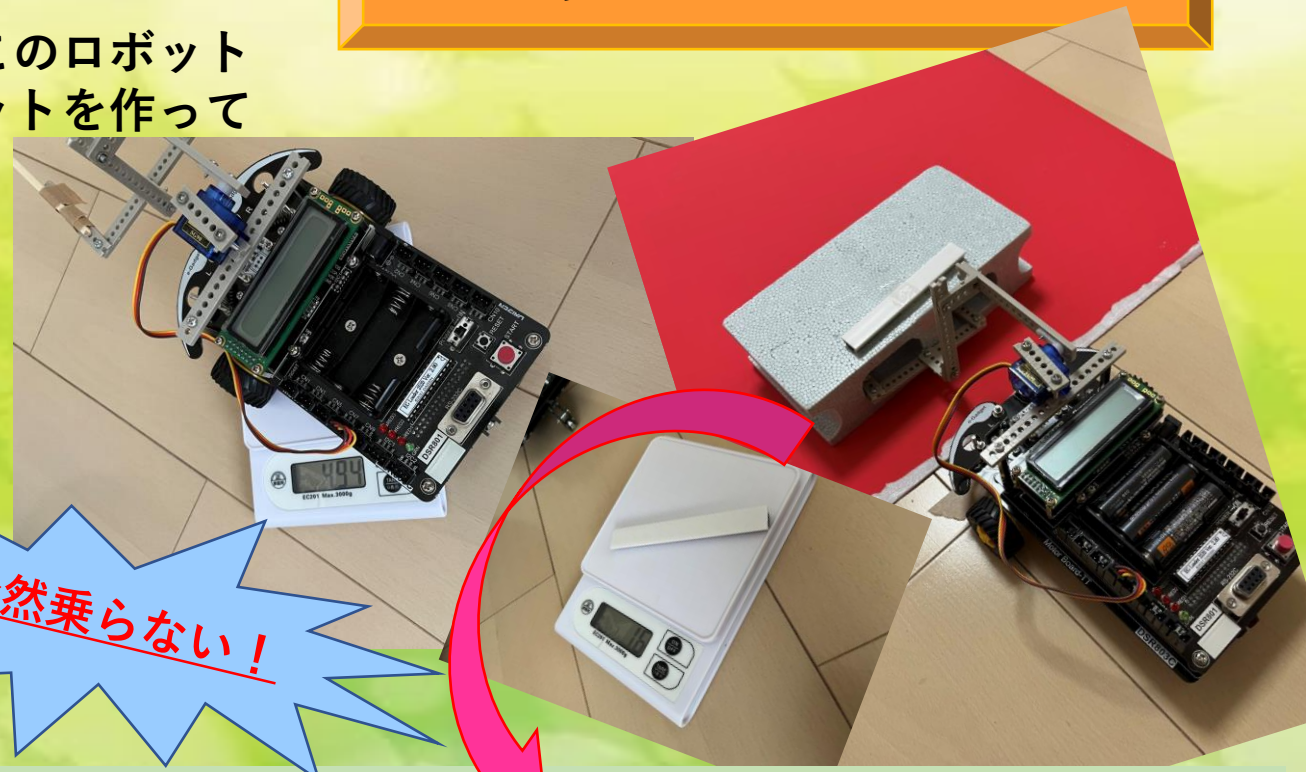
いつも使っているロボット
アームを付け、ブロックにおもりになる磁石は１個１６グラム
坂の高さは５センチ。
どれだけのおもりを付けて坂を上下りが出来るかを調べた。

感想

まだ、ロボットを走らせる前の予想は、１０個ぐらいは余裕で持ち運んでくれるものだと思っていたが５個という結果で大変驚いている。

私の考え

今回の結果を通してわかった事は、大きい荷物をもったり運んだりする為には、モーターの量を増やしたりする必要が重要で。物を運ぶことは想像しているよりも遙かにモーターの力が必要なんだと学習出来ました。
人の助けになるようなロボットがこれからは必要になると思うから、色々な実験をしていきたいと思えた事です。



全然乗らない！

おもり１個…１６グラム　　坂の高さは５センチ

☆１個・２個の場合・・・スムーズに**出来ました**。
☆３個の場合・・・１個２個と違い少し**重い**動き。
☆４個の場合・・・３個よりも車のお尻が上がり**危ない走行**
☆５個の場合・・・重すぎでアームが**上がらず**走行不可！

なんと４個しか乗せられませんでした！！！

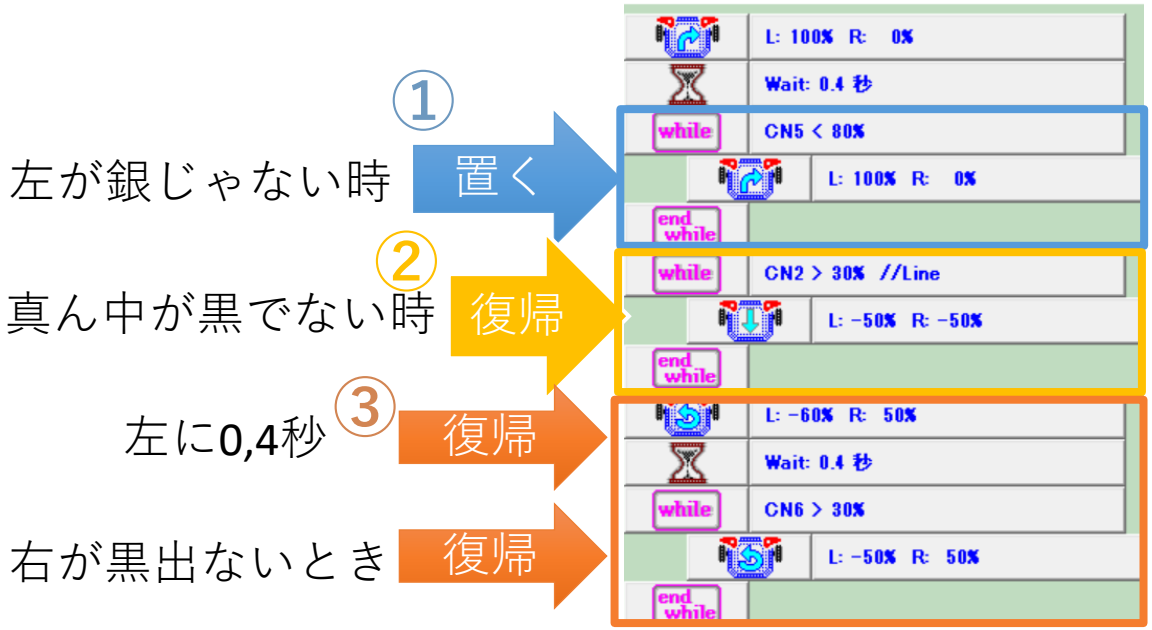
ＳＲＣ二次予選 Classic　　チーム　ポンカン　戸髙龍馬

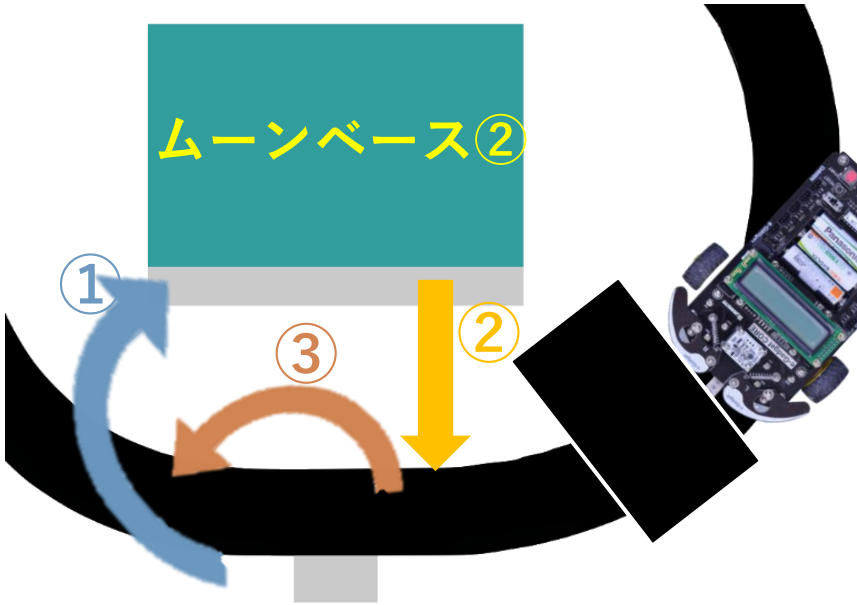
工夫点

僕は最初ほとんど秒数で回収したり設置したり、S字に移動させたりするプログラムを作成していました。
　しかし、秒数でプログラムを作ると電池の残量でロボットの動きが結構変わってきてしまうのでwhile ifをふんだんに使っております。でももちろん秒数も普通に使っている部分もあります。

例えば…

下のプログラムは僕が三番目か四番目ぐらいに苦戦したムーンベース2にコンテナを置くものです。
何に苦戦したのかというと、どれだけ正確にコンテナを置いて復帰できるかということです。十割できるようにwhile ifを使って工夫しました。





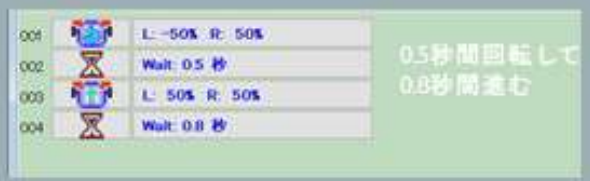
今回こだわったこと

チームカズシ　李一志

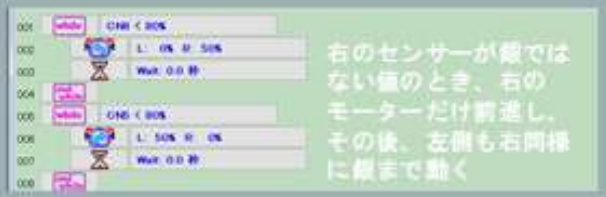
・できるだけ秒数で動くのではなく条件付きループを使うこと

秒数ですべて動きを決めてしまうと電池の残量に大きく影響される。
そのため「〇〇秒間〇〇する」ではなく、「〇〇の間〇〇する」というプログラムを組むことが重要

秒数



条件付きループ



・コースからロボットが外れた時に復帰するサブプログラム

使える場面

・ブロックをとる動作の後

・ブロックを置く動作の後

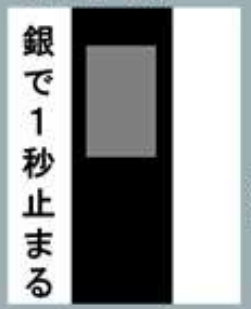
・S字コースへの移動のプログラム

など

Moon base1にブロックを置くプログラムでの使用例

・銀でセンサーが2回反応しないためのプログラム

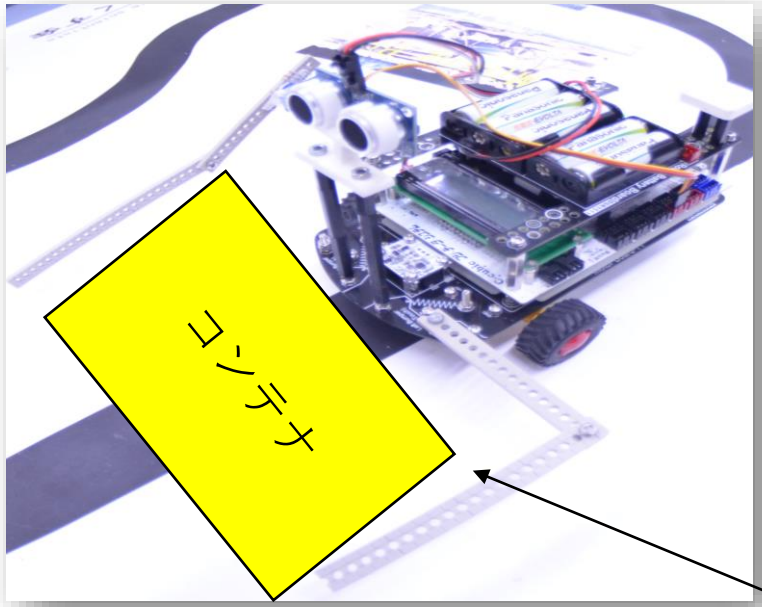
右上のような銀の場所で１秒だけ止まるプログラムを書くとき、１秒止まった後、２度止まらないように銀ではない場所までセンサーを反応させずに進まなければいけない。
右のようにすれば２度反応する可能性は大きく下がる。



チーム制覇

CLASSIC

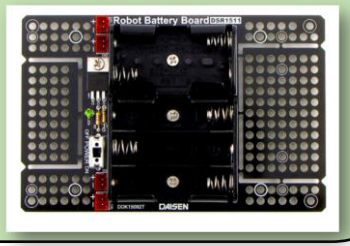
山中歩輝



ロボット紹介

1, **サーボモータ**を使用し、コンテナを取りやすくした

2, **バッテリボード**を使い電圧が上がった（7,5V）



コンテナを取る動作

1, 回転が全て軸回転

2, while ifを使い正確に動く

ロボットのメリット・デメリット

メリット：左のように取りやすく取り逃しにくい
→正確性抜群！

デメリット：コンテナを取るのに時間がかかる
→タイムに影響を与える…

意気ごみ

１次予選は完全制覇したので２次も完全制覇目指し頑張ります！

チーム名　でこ　佐藤優真　Classic

離脱のプログラムの工夫した点

下の写真を見たらわかるかもしれませんが、僕は秒数で離脱をしています。工夫して変えたところ、「002」の秒数を0.1秒だけ変えたことにより、成功率が50%以上上がりました。また、「003」は直線にするためにわざと、右にパワーを強くしました。

2つ目に工夫した点は、前回の弱点であったムーンベース3で、銀に反応しないところを改善しました。

どのように改善したかというと、簡単に説明すれば、最初にもし「パターン1」に入ったら、そのまま「パターン2」に移行し、逆に最初に「パターン2」に入ったら、「パターン1」に移行できるようなプログラムを作りました。

チーム名

にゅうさんきん

classic部門

メンバー

弥富　右恭

ロボット

目標

二次予選突破！！！！

実験

用意するもの

・A4方眼用紙1枚　・c-cubo　・割箸などの棒　・電池

方法

1. 方眼用紙のマス目に沿って長さを書き込む
2. 方眼紙を棒に巻きつける
3. 片側のモーターを100%で１秒間動かすプログラムをロボットにダウンロードする
4. ロボットのタイヤに棒をくっつけてロボットを動かす
5. 4を10回繰り返す
6. もう片側も10回繰り返す

結果

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

結果は右側の平均値が7.43cm,左側が7.98cmと

左側側のモータのほうが、モーターのパワーがおよそ1.1倍ほど

強いことが分かった。

感想

電池の関係で右側のほうが強いと思ったが違ってびっくりした。

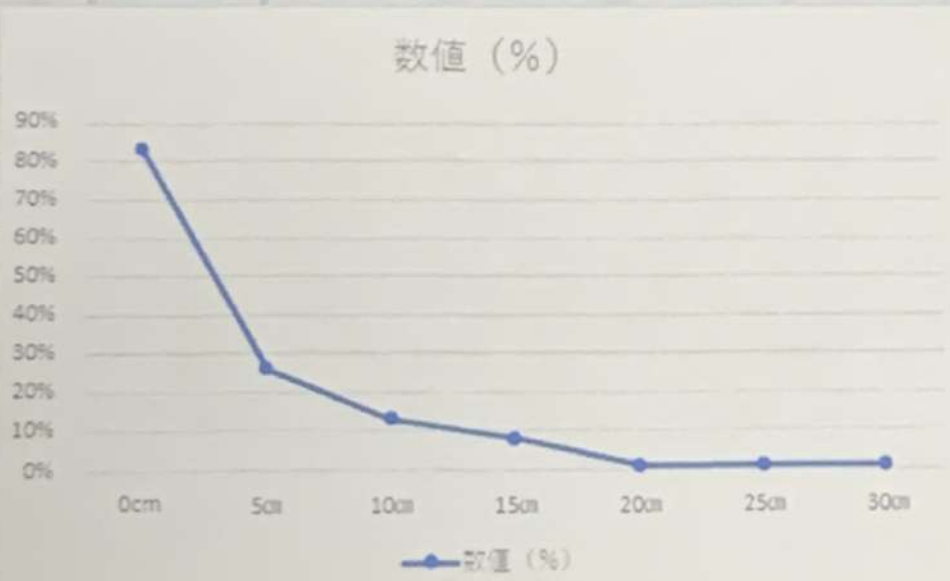
チーム名：ねこですよろしくおねがいします

自己紹介

名前：小西璃空

年齢：13歳

高さ	数値(%)
0cm	83%
5cm	26%
10cm	13%
15cm	8%
20cm	1%
25cm	1%
30cm	1%



ー実験ー

計る高さを変えたときにラインセンサーの値に変化はあるのか

ー実験方法ー

床に銀の折り紙をおいて本を積んで高さを変えて数値を測る(タイヤと地面の接触面を０cmとする)

ー結果ー

グラフの様に測る高さが高くなればなるほど数値は下がったが０になることはなかった

SRC CLASSIC

ファルコン

きれいなプログラムコードのこだわり

立命館慶祥中学校
１年　波方珀人

Subprogramの多用

メリット

① プログラミングのスピードアップ

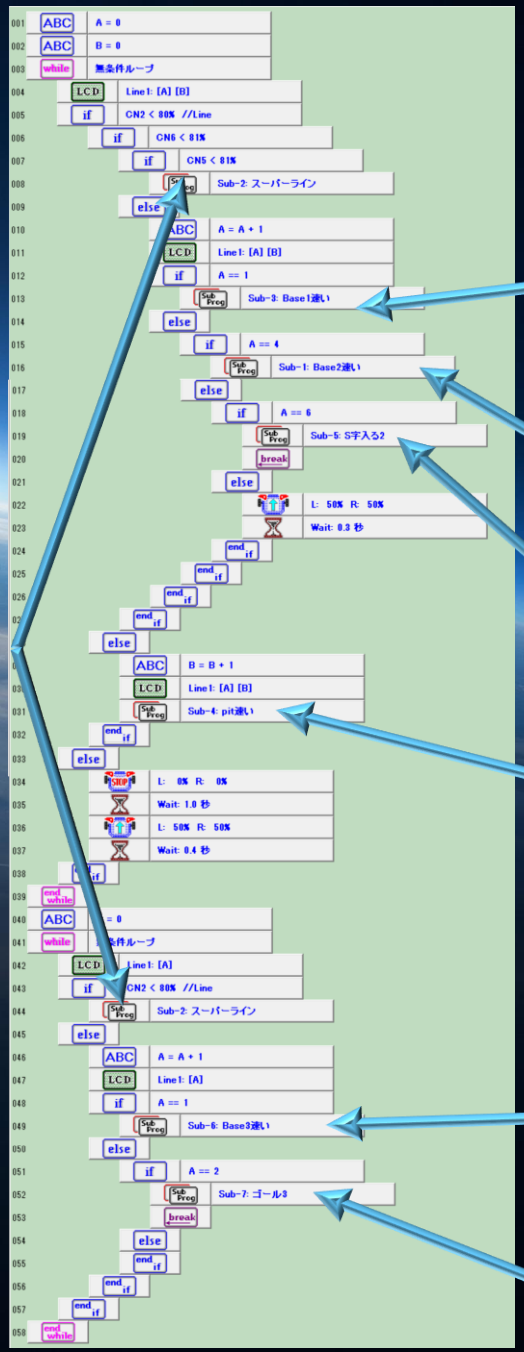
同じ動作のプログラムが複数必要な場合に便利。
修正するときも１つのSubprogramを修正するだけでよい。

② 修正するところがすぐにわかる

大会の本番や、本番前の練習時間に一度でおかしな動作をしたところがわかり、修正時間が短くて済む。

③ 人と共有しやすい。

見やすいプログラム。
どこでどんな動作をしているのがわかりやすい。
subprogram単位で渡すことができる。



ラインとレースをするプログラム

Base1のコンテナをおくサブプログラム。

Base2のコンテナをおくサブプログラム。

S字コースに入るサブプログラム。

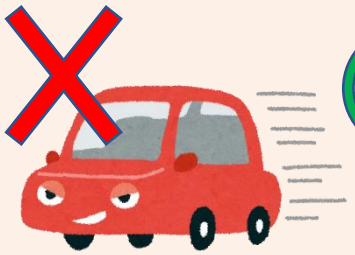
コンテナを運ぶサブプログラム。

Base3のコンテナをおくサブプログラム。

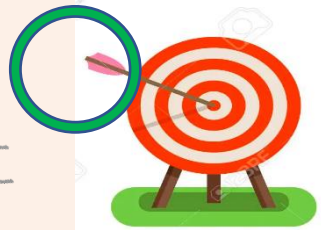
ゴールするサブプログラム

～早さより正確さを重視～

チーム名　フラミンゴ



早さ



正確さ

理由

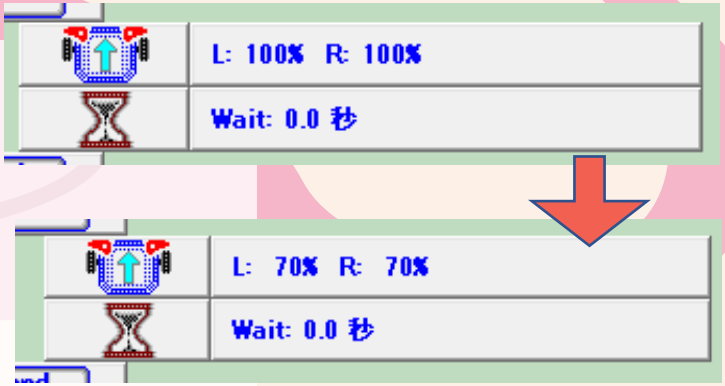
スピードが速いとコースアウトする確率が大きくなるからです。

白と銀の区別がしずらくなり、コースアウトをしてしまいます。

工夫したこと

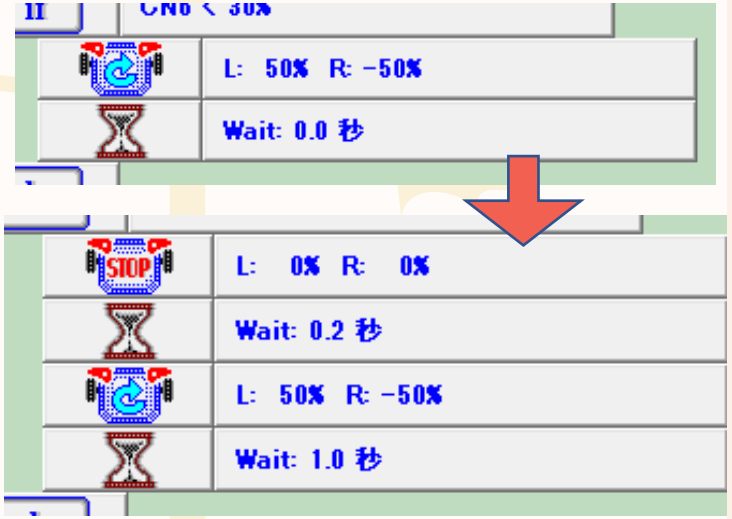
１．ラインとレースの直進のスピードを３０％下げてみました。

コースアウトする確率は２／３と変化はあまりありませんでした。



２．銀がきたら、すぐ回転するのではなく、０．２秒間停止してから回転させるようにしました。

コースアウトする確率は大幅に少なくなりました。



初めての大会で、知識はすごく少ないですが、

Trial and error

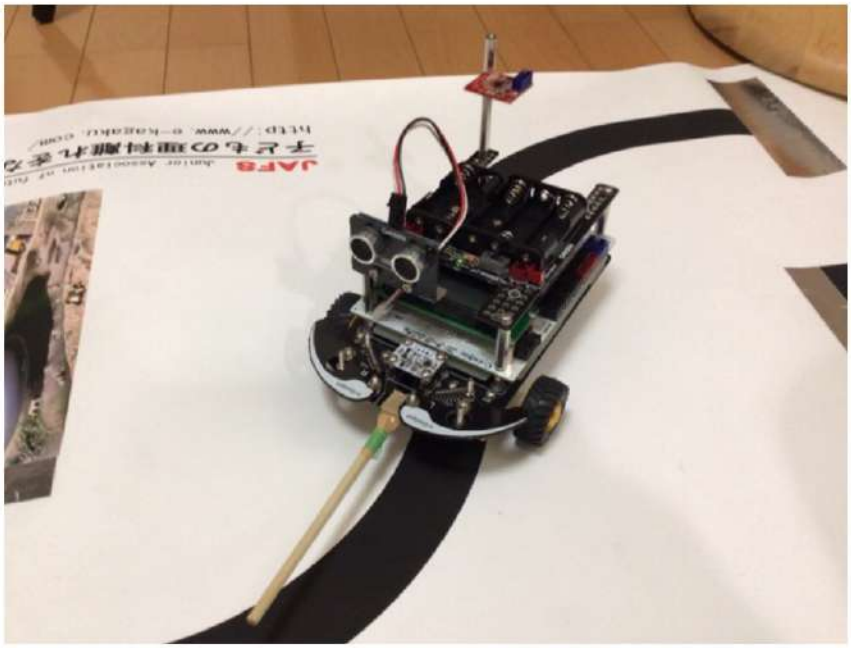
（トライアンドエラー）の繰り返しだなあと思いました。

SRC18　classic

チーム名【へのへのもへじ】内藤長汰

※工夫したところ（ロボット）

- 割り箸（外れないようにした）
- 余計なものをつけないようにした（simple is best）→プログラムでも、ロボットでも、シンプルな方がわかりやすいから



※意気込み

- 去年、体調不良で2次予選行けなくて悔しかったので、今年が最終回。なので、頑張って、絶対に優勝したいです！！！

シートプレゼン　チーム名（めだか）　出場カテゴリー（Classic）　　作成者（鈴木悠斗）

目標

見やすく少ないプログラムで、正確な動きができるようにすること

工夫した点

１、サブプログラムを使うことで、同じ動作をまとめてコード量を減らしたり、見やすく修正しやすいようにした。

２、コンテナの回収に使うパーツはユニバーサルプレートを使うことで、シンプルにして余計な動作をしないようにした。

３、コンテナの回収と設置の動作は、センサー値をもとに秒数を決めることで、少ない動作で確実に行えるようにした。

今回見つかった課題

明るさの変化でセンサー値が変化するので、それに伴い閾値が変わってしまい、正しく動作しないことがあった。

検証

・日光の当たる部屋でコースの白色をラインセンサーではかり、閾値を求めた。

・１分おきにラインセンサーの値を調べることを１０分つづけ、３０分後に同じことをした後に前半１０分と後半１０分の白、黒の閾値を求めた。黒の値は４５から変化しなかった。

結果

閾値は変化しやすく、動作に大きな影響が出る。

検証の結果

10時から	10時40分から
2685	2989
2600	2988
2633	2850
2626	2875
2708	2833
2719	2563
2706	2853
2707	2932
2645	2944
2677	2951
閾値	閾値
1357.8	1461.4

今後について

・センサー値を変数にして、自動で閾値を修正できるプログラムを作りたい。

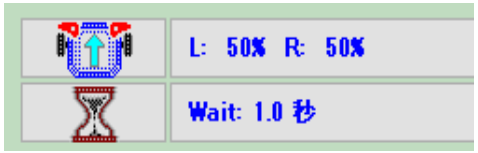
・今回と同じ目標を持つほかに、自分で課題を見つけて工夫をすることでより良い結果を出せるようにしたい。

SRC 1次予選で工夫したこと

チーム名：ユージン03（Classic）氏名：黒政　侑史

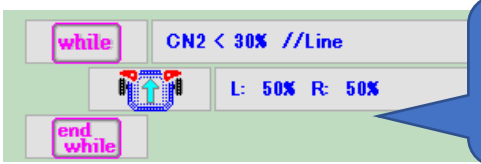
課題

S字コースへの移動の確率が低い
→直進する距離を秒数で制御していた



改善①

条件付きWhileを使い「CN2が黒になるまで前進」に変更



成功率50%にUP
しかし、100%
ではない

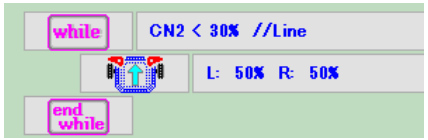
問題点

「L:50% R:50%」にしていても右側へコースアウトする
⇒ このロボットは右側へ曲がる癖があることが判明！

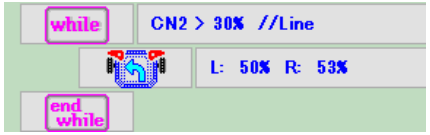
改善②

直進から少し左に曲がる（R:50→53%）
プログラムに変更

Before

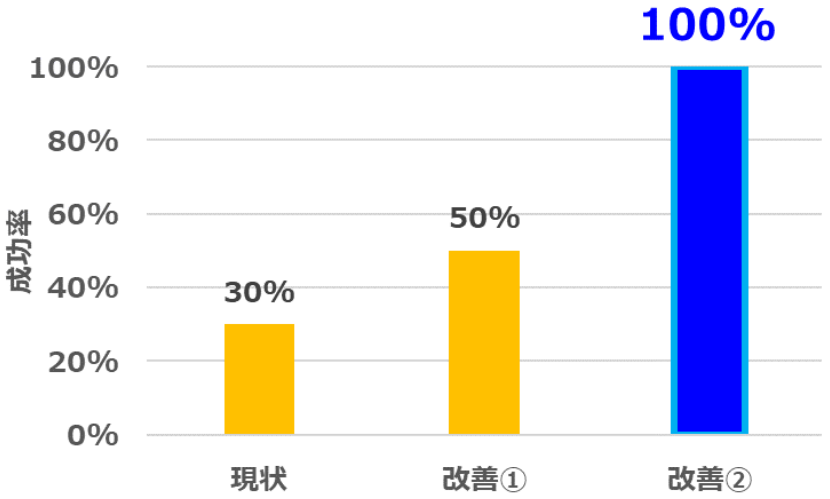


After



結果

S字コースへの移動成功率

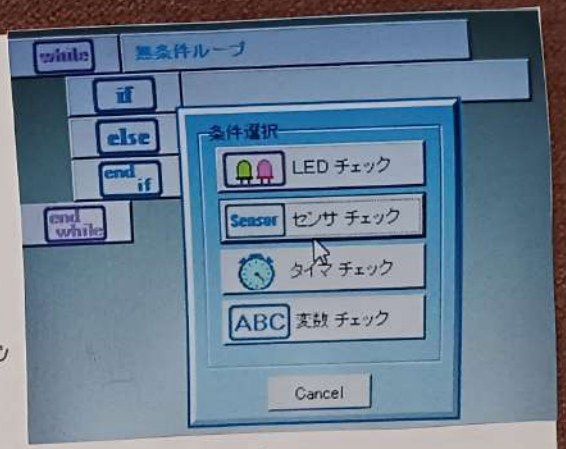


課題解決することに成功！

チーム、ロボ　(谷口 陸仁) Classic

障害物を上手によけるために…

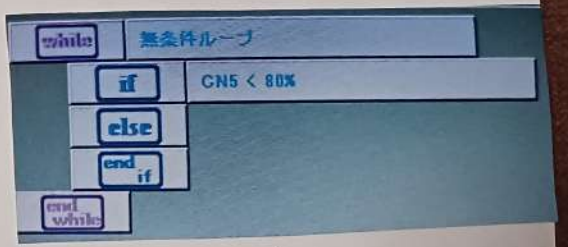
ぼくは、タッチセンサーを使いました。①の写真のセンサーチェックを使いますセンサーチェック CN3 CN4 を使います。それで、タッチセンサができます」



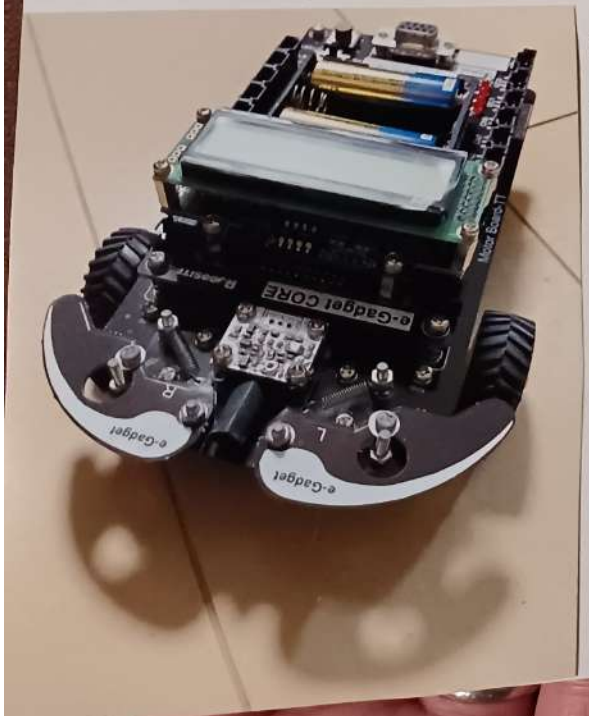
↑
①の写真

おかしくなった事

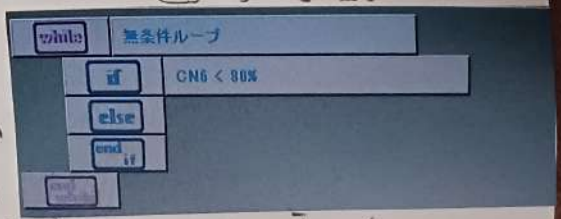
ぼくは、ビデオプレゼンでは、4つの問題で動かなくなると言っていましたがぼくは、ある事に苦しみました。それは、Aの所からライントレースにもどれない事です。ずっとバックしたりして、苦められました。ぼくは、これは絶対に合っていると思っていました。なので、おかしいなど思っていました。そこからぼくは、また1からやると直したので、2次予選では、自分は間違っているかもと思うのが大切だとしりました。



↑
②の写真



③の写真



チームさふぃーる

目指せ　2000点!!

初めてclassic部門に挑戦します。

ロボットの回転する角度の調整を何度もしました。

コンテナの置き方アシストの仕方もなかなかうまくいかず練習しました。

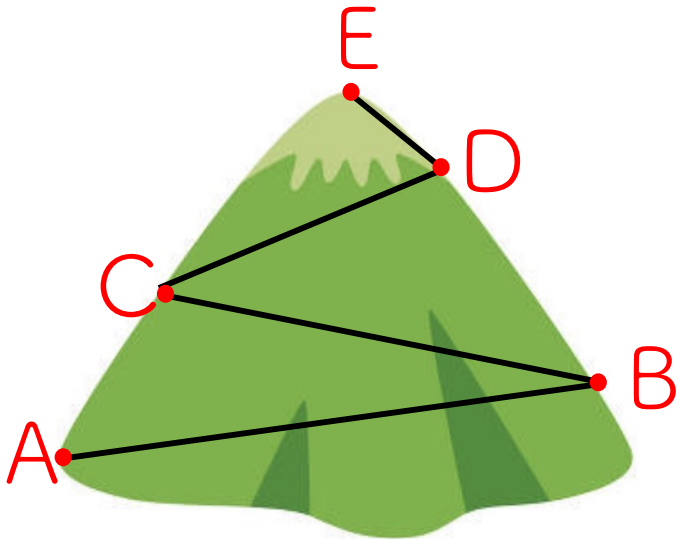
もう一つ工夫したのは、電池の減り具合で、パワーとスピードが変わるので、たくさん試しました。

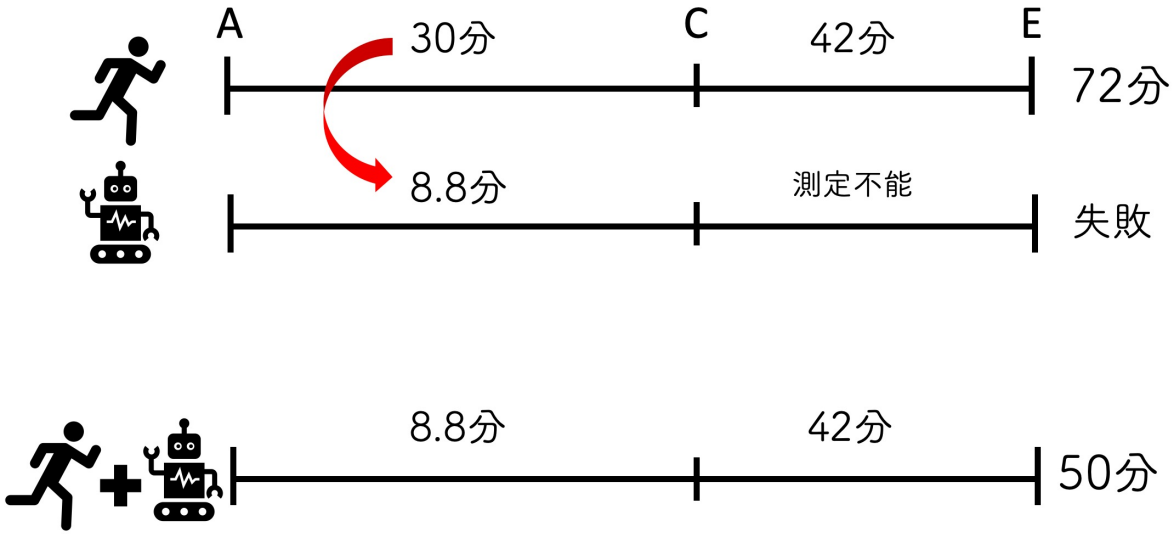
あせらないで
ひとつひとつ
がんばります!!

下克上^2

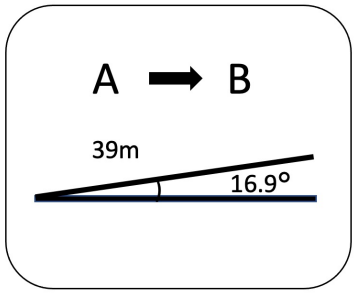
高校2年生 駒野和音
中学2年生 駒野生光

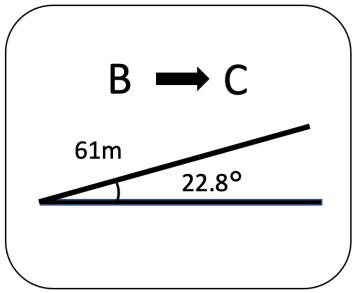
山のそれぞれの地点を村と仮定した上で、E村（頂上）に物資を届けることを想定し研究してみました。

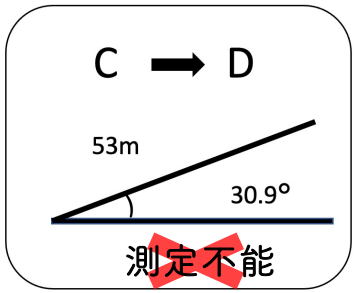


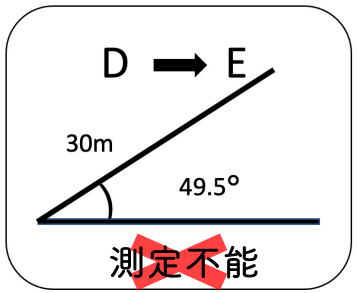


C村まではロボットが運び、C村からは人間が運ぶことで効率が良い。









C村までなら人間よりロボットの方が**４分の１の時間**で運べる。

使用したロボットは**30° 以上の斜面は登ることができなかった**

細かいことはこのQRコードを読み込んでもらうとわかります！→

化学反応（Classic出場）

安井瑞生・飯田一真

＜実験＞

前進しながら曲がるのと、後退しながら曲がるのは、どちらが曲がりやすいのか？

モーターの出力比を60:30に固定した状態で、左右それぞれ前進・後退90°の4つを5回ずつとりその平均を求めました。

複数回実験をすることでより適切な値に近づきます。

その結果・・・前進右折　1.5秒
前進左折　2.1秒
後退右折　1.9秒
後退左折　3.0秒　となりました。

→前進右折と後退左折には2倍もの差がある

＜結論＞

右モーターと左モーターには大きな出力差がある
本番では左モータを強めに設定する必要がある。

＜チーム紹介＞

チームメンバーの二人は中学三年間同じクラスで確かな連携が可能です。また二人ともzoomやskypeなどのオンラインツールを使用するスキルがあるので、離れていても連携が可能です。

ちなみに本番はdiscordを使用してプログラムに関して議論する予定です。

＜工夫＞

安井は秒数制御、飯田は角度制御が双方得意です。

すなわち二人がそれぞれプログラムを考えると全く別物のプログラムが完成します。その後各部分ごとにどちらのプログラム、機材が適しているかを考え丁寧に自分たちのプログラムを決めます。

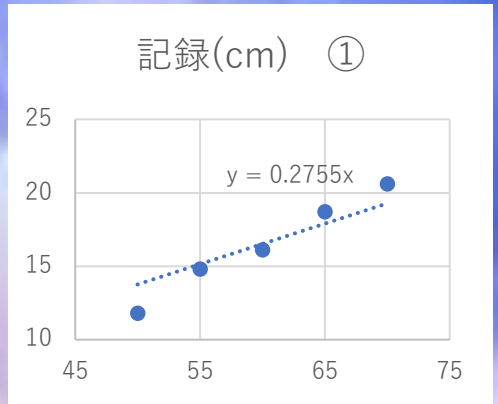
これにより高確率で自分たちの最高のパフォーマンスを発揮でき、万が一失敗してももう一方のプログラムがあるのでトラブルにも強いです。

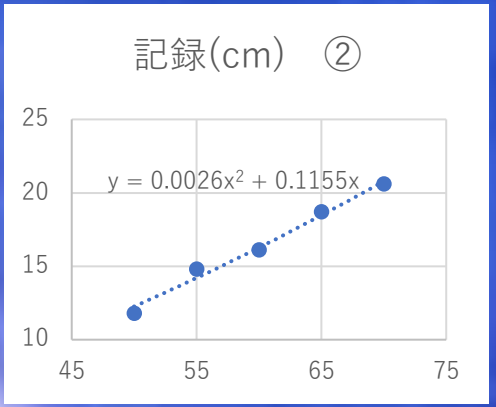
〈光芒（こうぼう）〉

Classic部門

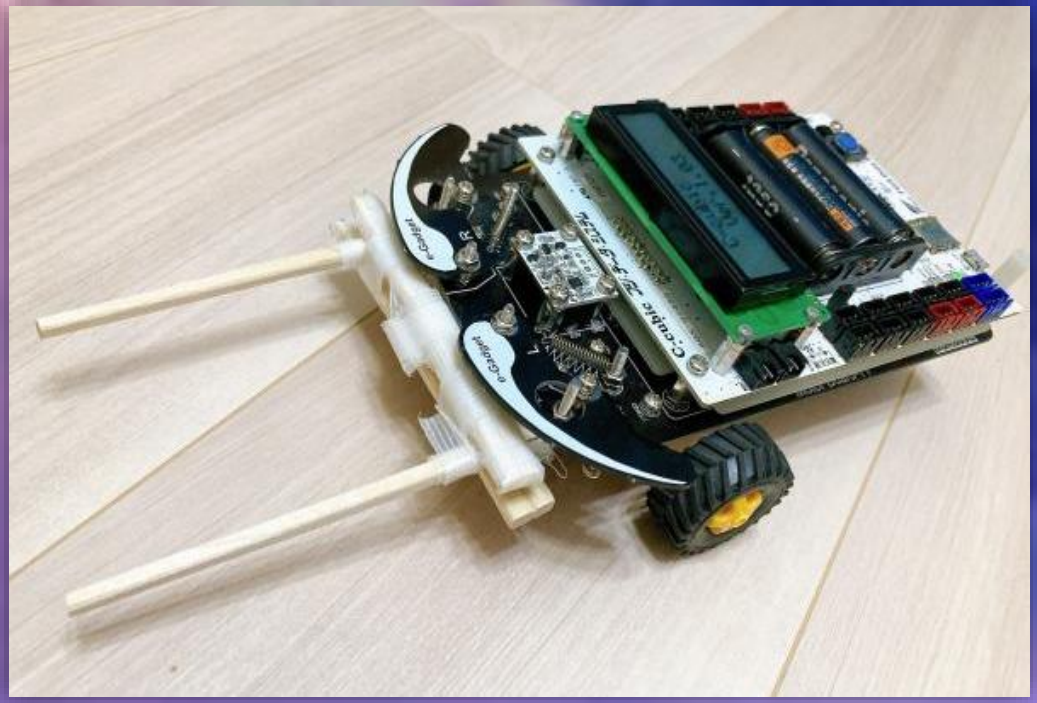
〈Name〉鶴田　獅央(14)
石嶋　海　(15)

【実験】
ロボットの出力パーセンテージと走行距離との関係を調べるためにモーターの出力値(%)と走行距離(cm)の関係を測定して表にし、それをグラフに表した。
なお、左右のモーターの出力値が違うのは、モーターに個体差があり、私たちが使用したモーターで直進させるためにこのような値になった。





【結果・考察】
左右どちらも０％のときは走行しないのでy切片を０に固定した。このとき近似曲線を一次関数にすると（①）、そのグラフと測定値に大きな誤差が生じてしまう。しかし近似曲線を二次関数にすると（②）、一次関数のときと比べて誤差が減った。
このことから、これらの２つの関係を表すにはy=axの形ではなくy=ax^2+bxの形で表すべきであることが分かった。

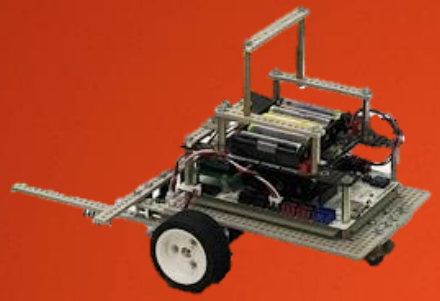


(表)

出力(%)	L:54 R:46	L:65 R:55	L:72 R:58	L:77 R:63	L:82 R:68
左右の平均値(%)	50	55	60	65	70
記録(cm)	11.8	14.8	16.1	18.7	20.6

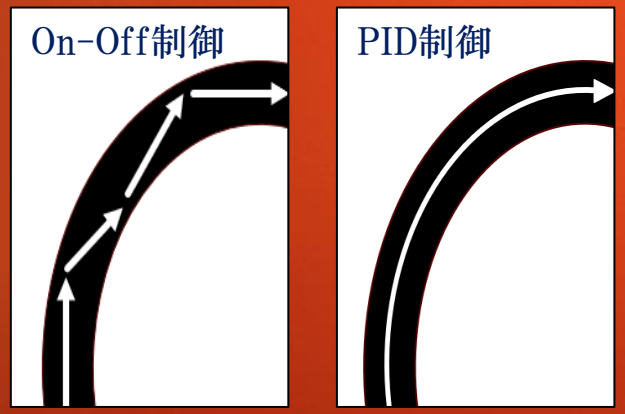
Team　揚げたこ

選手名：荒川 拓海
出身：京都府
年齢：16歳
出場競技：classic
経歴：予選大会二回連続“完全制覇”
モットー：百戦百勝



ライントレース

　タイム短縮のためOn-Off制御から滑らかなライントレースができるPID制御へと移行した。



完全制覇を確実なものに

　完全制覇を難化させているのは電池残量の変化による移動距離、特に回転角度の変化である。これを克服するためのアルゴリズムを考案した。

アルゴリズム

　ロボットは、黒ラインの縁とラインセンサーが垂直であることを用いて90度曲がることができる。90度曲がるのに要する時間を計り、定数をかけることで、30度回転や100mm前進などを電池残量に関わらず行うことができる。
私は計測を各周回の荷物取りの際に行い、電池残量をリアルタイムでプログラムに反映させた。

目標

　全課題 ““ 完全制覇 ””

出動カテゴリー：Classic
高根　将伍

僕が使っているロボットの性能について

TEAM ： 高根ひょうたん

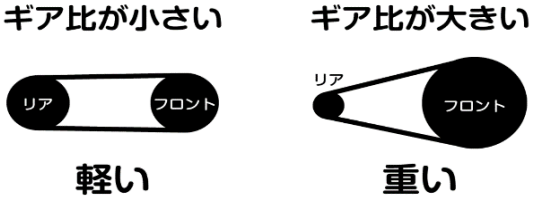
いつも僕が使っているロボットに使っているモーターの性能について調べてみました。
モーターの性能は何で決まるのか調べたところ、

『　ギ　ヤ　比　』

で決まってることがわかりました。

ギヤ比とは**タイヤが１回転するために必要なモーターの回転数**で表します。
特徴としては

モーターの回転数が多くなればパワーは強くなるけどスピードが落ちる
モーターの回転数が少なくなればパワーは弱くなるけどスピードが速くなる

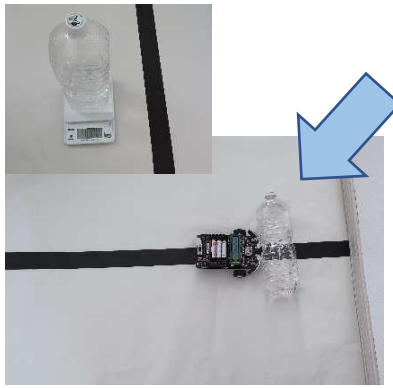


僕たちが使っているロボットのモーターのギヤ比は

48：１　だとわかりました。

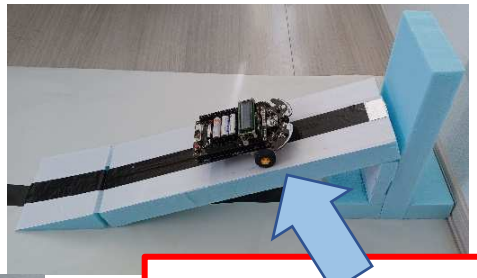
そこでこのモーターのパワーを確認するために２個の実験をしました。

１　どのくらいの重量を押せるか



結果！
900ｍｌ以上は押すことができませんでした

２　どのくらいの斜度を登れるか



結果！
最大で　２０度まで登ることが出来ました

では、ロボットのパワーをもっと上げるにはギヤ比を上げるだけではない？！

プレゼンテーション

チーム名：Lonely　　チームメンバー/選手名：根本　泰地

機体紹介

・使用機体　　c-style for c-cubic

・使用言語　　C 言語

・使うプログラム（仮）　　ライントレース＋荷物の回収

・工夫点　　機体自体に特に工夫はない
だが、プログラムに関して、安定した動作をしてもらうために
「条件付き while(センサーが一定 以下・以上 の場合一つの動作を繰り返す)」
を採用している

今回の目標・意気込み

目標

・二次予選の突破　・決勝への出場　・フルスコアの獲得

意気込み

今回が初めての二次予選ですが、どんどん挑戦して二次予選も三次予選も突破していって、決勝まで勝ち残りたいと、思っています！

今まで習ってきた知識や使える技術を総動員して、全力で抗います！

SRC18 Classic 2次予選

佐藤SRC

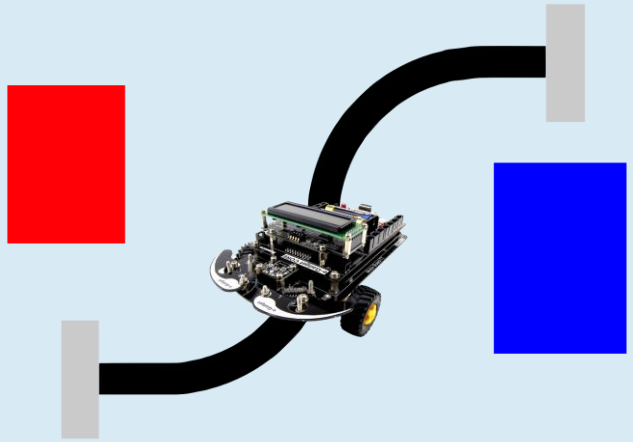
佐藤　俊武根　明法中学校１年

僕は今回の挑戦で、プログラムの正確さを上げられるようにしようと思います。

前回の挑戦では、コースアウトしてしまう確率が高めだったため、今回の挑戦でほぼすべての可能性に対応できるようにしたいです。

今回の取り組みでは、よりコースから一時出るときのスピードを落として、センサーが感知しやすいようにしたいと思います。

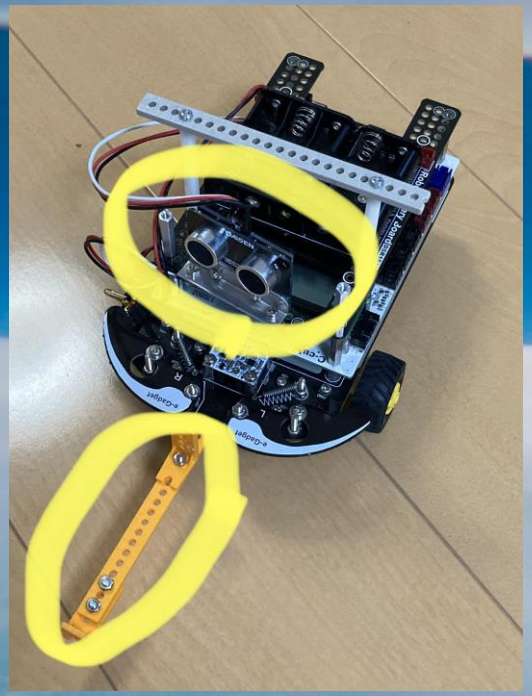
また、もう一つ意識する点としては、ライン上の移動時にカーブがあるため、コースアウトしないように、少しだけ全体的な速度を落としておこうと思います。



走行予想速度

通常	80～85
コース復帰時	50～70

ＳＲＣプレゼンシート



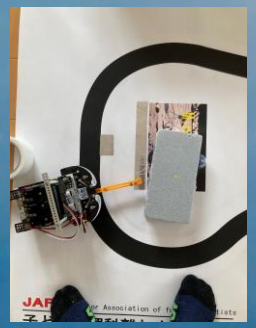
工夫した点

超音波センサーを少し横に向けてコンテナを読み取るようにしたこと

コンテナが引っ掛かりやすいように取っ手を付けたこと

意気込み

ＳＲＣ１７では第二次予選完全制覇は出来なかったので今度こそ完全制覇を成し遂げれるようにしたいです



発見したこと

一番コンテナが入りやすい置き方を探してみました

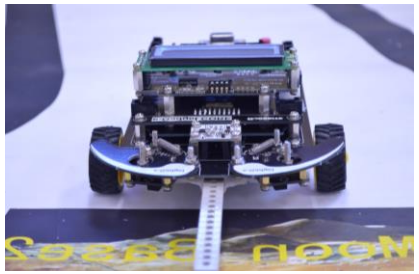
左はあまり曲げなかったのでうまく入りましたが右の写真では斜めにしすぎて引っかかっているのでコンテナにあたって取れていません

SRC18　クラシック　2次予選

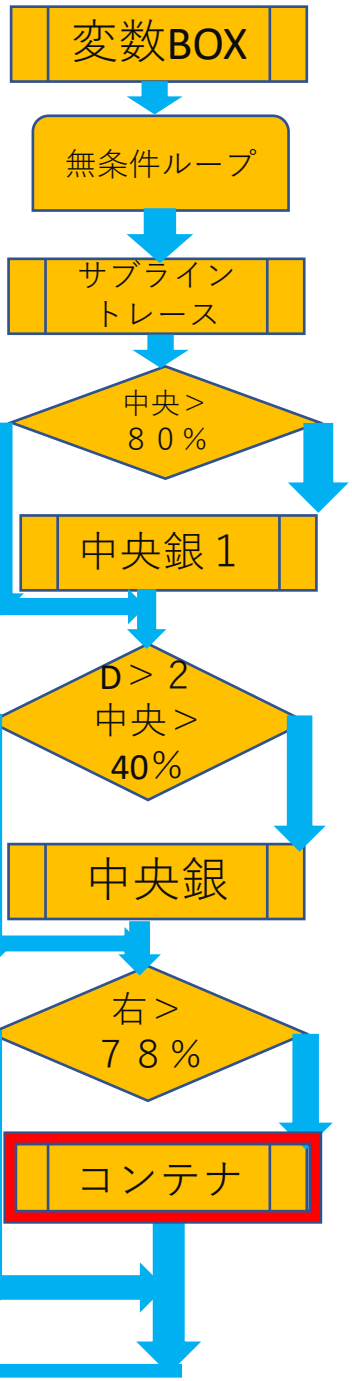
チーム

漆黒のスライムソード7

明法中学1年　三宅惟葵



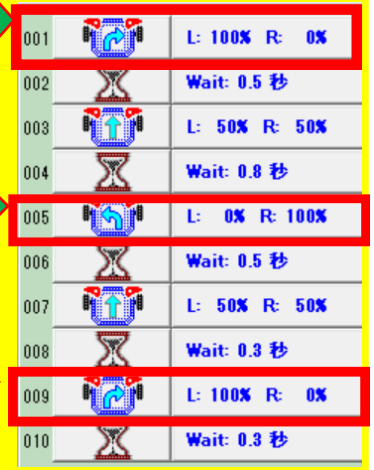
My プログラム



工夫ポイントは**２つ！！**

１.コンテナの取り方

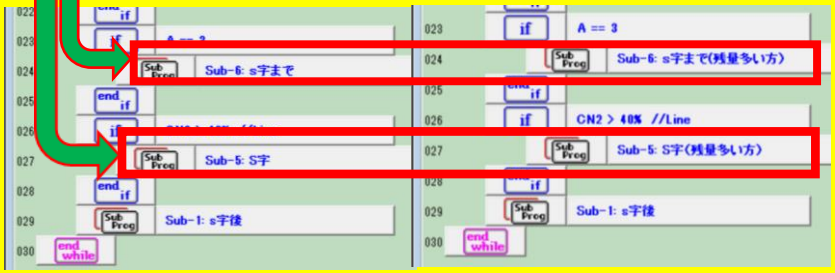
曲がるときに片輪が止まるように
プログラムしている。



2.プログラムの入れ替え

秒数でプログラムしているので、**電池がなくなると大変**

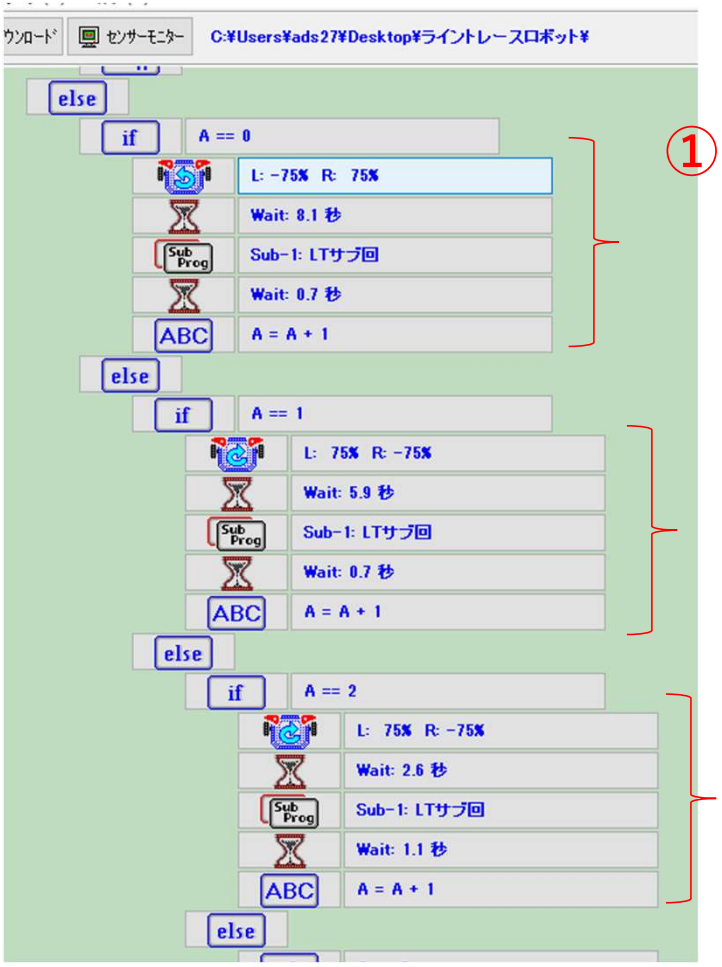
電池の残量によって、プログラムの秒数を変えるようにしている

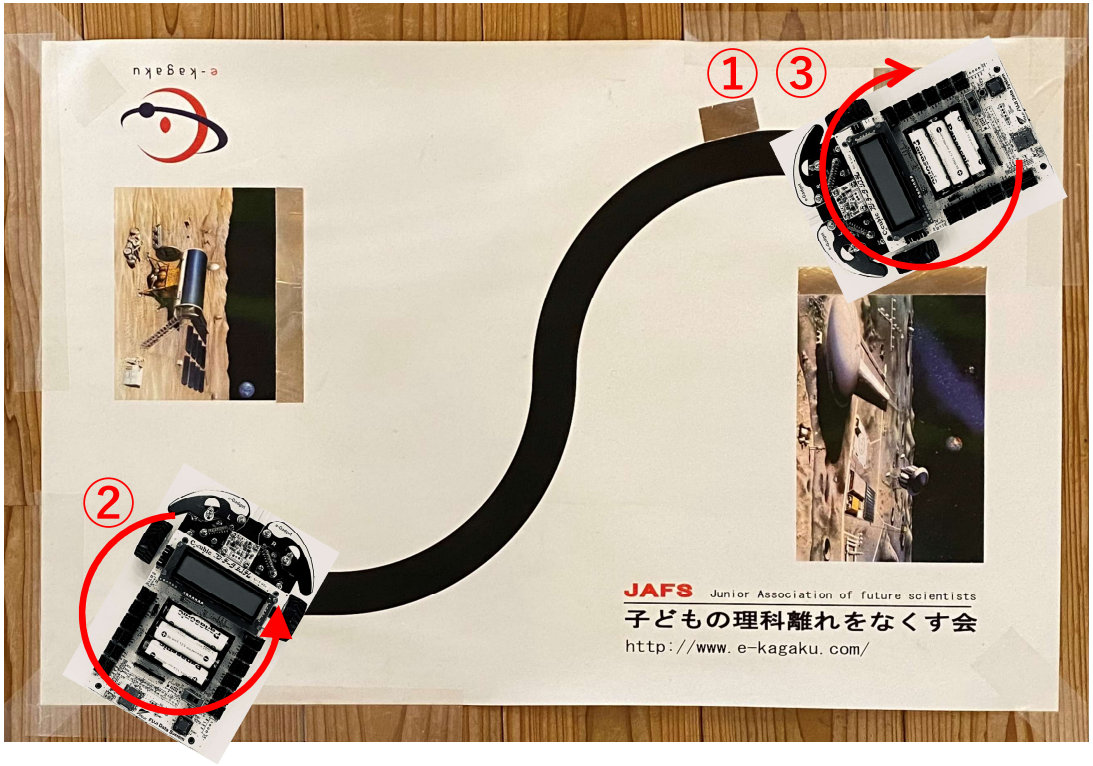


チーム　ペクス

Classic　森下陽斗　小6

挑戦したこと：
苦手なタイマーを上手く使うこと





①②はクリア出来たが③が上手く出来なかった

何故出来なかったのか
・タイマーが0.1秒違っても動きが全然違ったから
・③まで到達出来た回数が少なく、調整が上手く出来なかったから

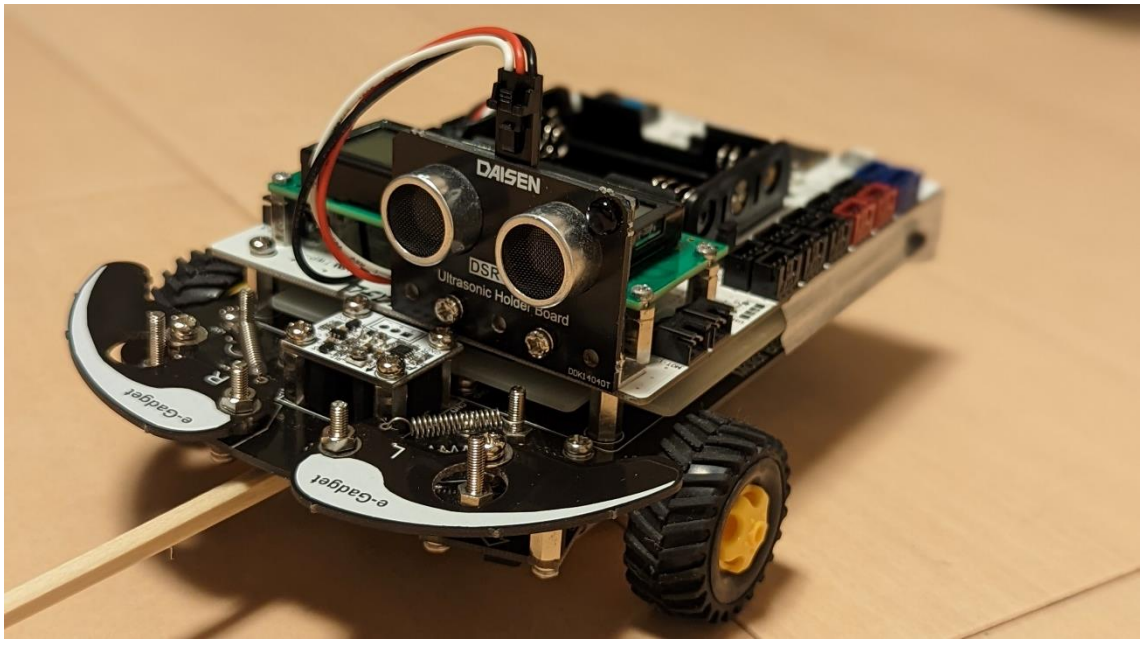
この事からタイマー調整の難しさが分かった

チーム名:ロワン　名前:白水瑛章　カテゴリー:Classic

私が工夫した点は、
コンテナを置く、ライントレースをする、S 字コースに行くなどのプログラムをメインプログラム(void user_main)に全て詰め込むのではなく、Leftcontainer・rightcontainer などの分かりやすい名前のサブプログラムを作ることによって、他の人にも伝わるようにしたことです。また、プログラムに誤りがあった　場合には Atom の一纏まりごとにプログラムを折り畳む機能を使用することによってスクロールする手間を省き、トラブルシューティングを容易にする狙いもあります。
二つ目は、できる限り秒数待ち(wait_ms)を減らしたことです。なぜ秒数待ちを使うことは避けたほうが望ましいかというと、電池残量によって待っている間に進む距離や角度が変わり、プログラムの失敗につながるからです。

```
void gobase3(void){=}
void rightcontainer(void){=}
void leftcontainer(void){=}
void pickupcontainer(void){=}
void linetrace(void){=}
void user_main(void){
  while(TRUE){
    if(gAD[CN6]>3272){
     pickupcontainer();
    }else if(gAD[CN5]>3228){
      gV[VAR_A]=gV[VAR_A]+1;
      wait_ms(300);
      lcd_puts_var1(1, VAR_A);
      if(gV [VAR_A]==1){
        leftcontainer();
      }else if(gV [VAR_A]==4){
        rightcontainer();
      }else if(gV [VAR_A]==6){
        gobase3();
      }else if(gV [VAR_A]==8){
          mb3set();
      }else{
          while(gAD[CN5]>3272){
            motor(50,50);
          }
      }
    }else{
      linetrace();
    }
  }
```

こちらがロボットの写真です。
ロボットにはコンテナを持つための適切な長さに揃え、ロボットが後退するときには意図したようにコンテナが抜ける位置に割りばしをつけています。また、超音波センサーは取り付け場所を可能な限り下げることによって小さい障害物も検知できるようにしました。
以上です。

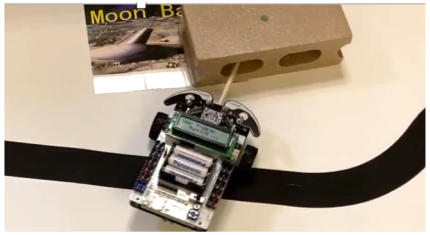


素数少年

中島聡太（10歳）
SRC出場回数：2回
classicコース

問題

荷物を置く動作で，荷物が
割りばしに引っかかってしまう



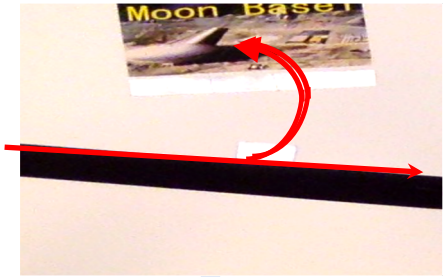
荷物を置く動作は
どのようにしたら
いいのだろう？

方法

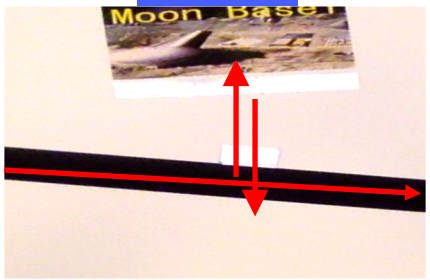
荷物の置き方3通り×20回ずつ 実験して成功率を比べた

＊成功条件：ブロックの中心に貼ったシールの全てがMoon Baseに入る

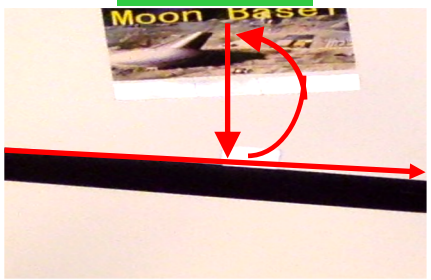
C字型



I字型



D字型



結果

成功率

C字型 45%

I字型 95%

D字型 100%

結論

D字型を使うと良い

オポチュニティー

SRC18の目標

トロフィー（3位以内）を目指します！

工夫した事

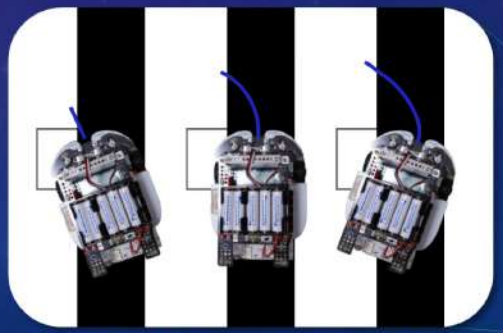
・モータを秒数で動作させると、電池残量でモーターの速度が変わってしまうので、ラインセンサーを使った条件付きループで動作を決めた。

・プログラムを修正することは多くあるので、プログラムを見やすくする為に大部分のまとまりをサブプログラムに入れる事で、プログラム全体を見やすくした。

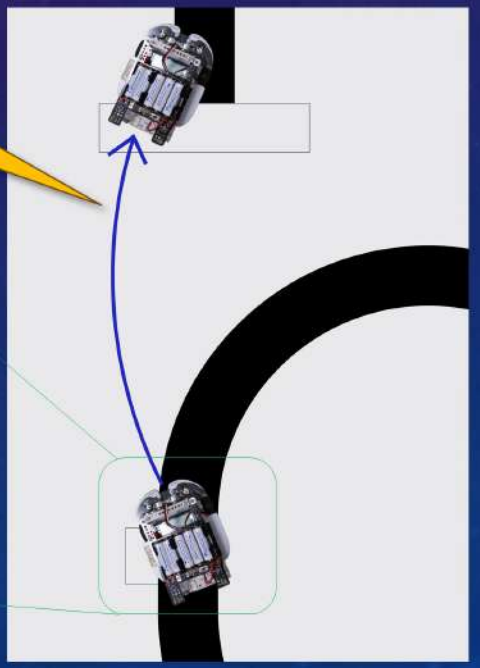
・S字コースにジャンプをするとき、成功率を上げる為にジャンプ前で微調整をするようにした。（右図参照）

・ラインセンサーを使った動作を確認するための物(下図参照)を作成した。

微調整したことで
ジャンプが安定した！



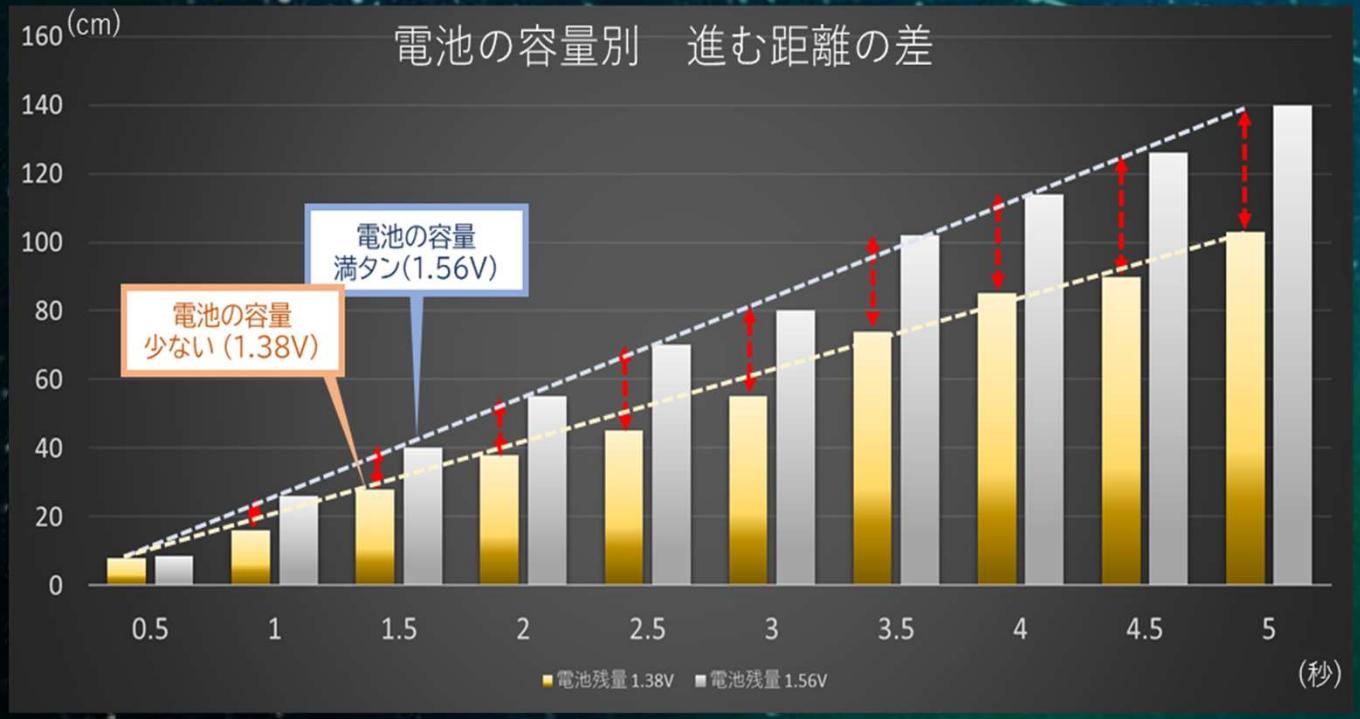
銀テープを検知した時のロボットの姿勢から、左斜め前を向くように微調整するようにした



チーム名　《流れ星★》NAGAREBOSHI 藤田悠暉（CLASSIC部門）

工夫、改善した点

①ロボットが走ったときの滑りを良くするため、後方のボールのネジを緩めました。
②電池が満タンのときと少ないときで進む距離に差があるかどうか検証用のプログラムを作成して調査しました。調査の結果が下のグラフです。



調査の結果、秒数が長くなるにつれ、距離の差が広がる事がわかりました。そのため、電池の容量に左右されないプログラムが必要だと考えました。

秒数ではなく条件付きループを使ったプログラムにすれば、電池の残量に関わらずロボットの進む距離を一定にできる。

抱負

良い記録を出せるように頑張ります！