

ロボット博士養成講座

ロボティクスプロフェッサーコース

オムニホイールロボット③

第6回

ロボット競技大会

講師用

目 次

0. ロボット競技大会

0.0. 「ロボット競技大会」でやること

0.1. 必要なもの

0.2. 操縦プログラムをあらためて読みとく

1. 競技大会

1.0. スラローム走行タイムアタック

1.1. 玉ころがしリレー

1.2. サッカー大会

2. まとめ

○ 授業開始にあたって

授業のはじめは、着席させ、大きな声であいさつしてから始めます。

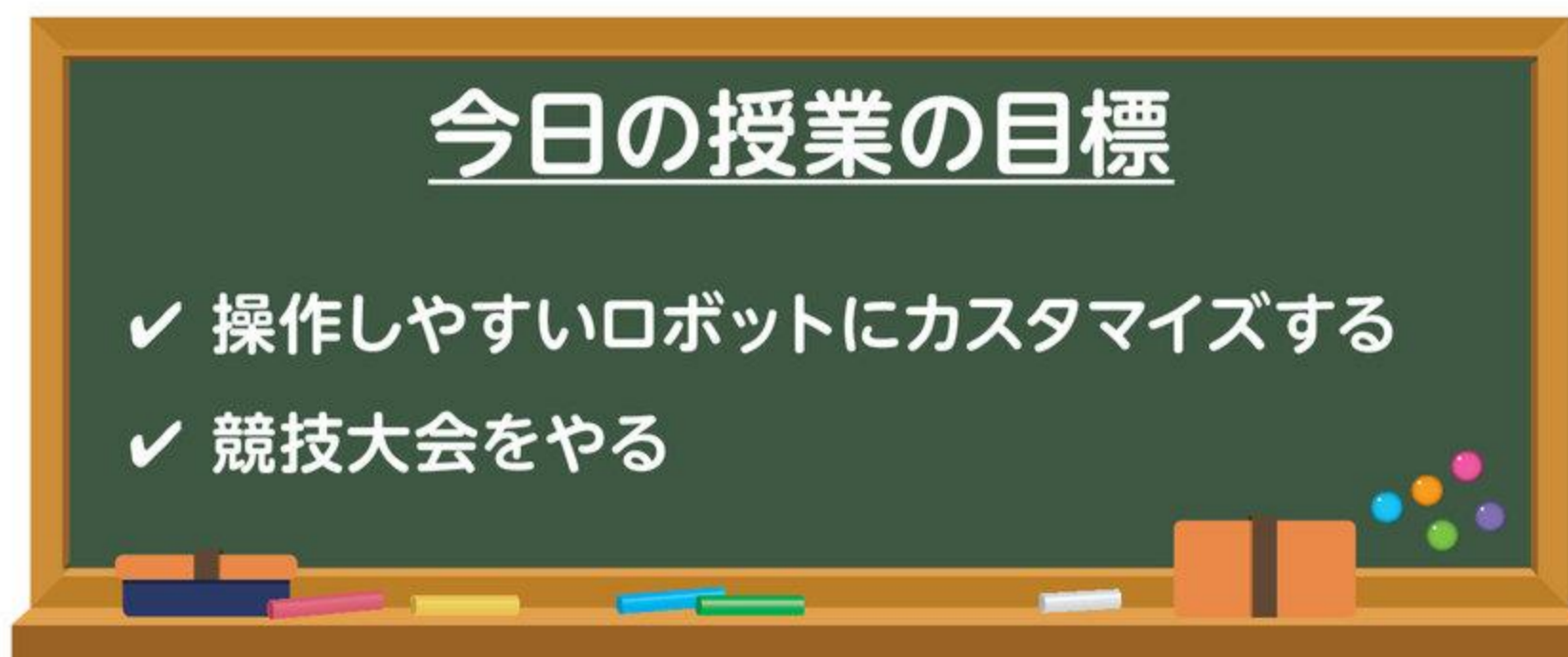
○ 今回の目標をパネルで用意するか、黒板に予め書いておきます。

(授業の目標を明確化することは大変重要なことですので、生徒によく理解させます)

目安時間は授業時間 120 分のうち、休憩 10 分程度取ることを想定しています。
生徒の進捗状況により、休憩時間などを調整して授業を行ってください。

0. ロボット競技大会（目安40分）

0.0. 「ロボット競技大会」でやること



前回までの授業では、オムニホイールのしくみから、力の合成やセンサーの扱い^{あつか}についてなど、いろいろなことをマスターしてきました。

今回は、これまでの総復習をかねて、オムニホイールロボットを使った競技大会をします。これまでに学んだ内容を存分に生かして、オムニホイールロボットをカスタマイズし、自分好みのロボットに仕上げましょう！ 目指せ！ 最強ロボット！！

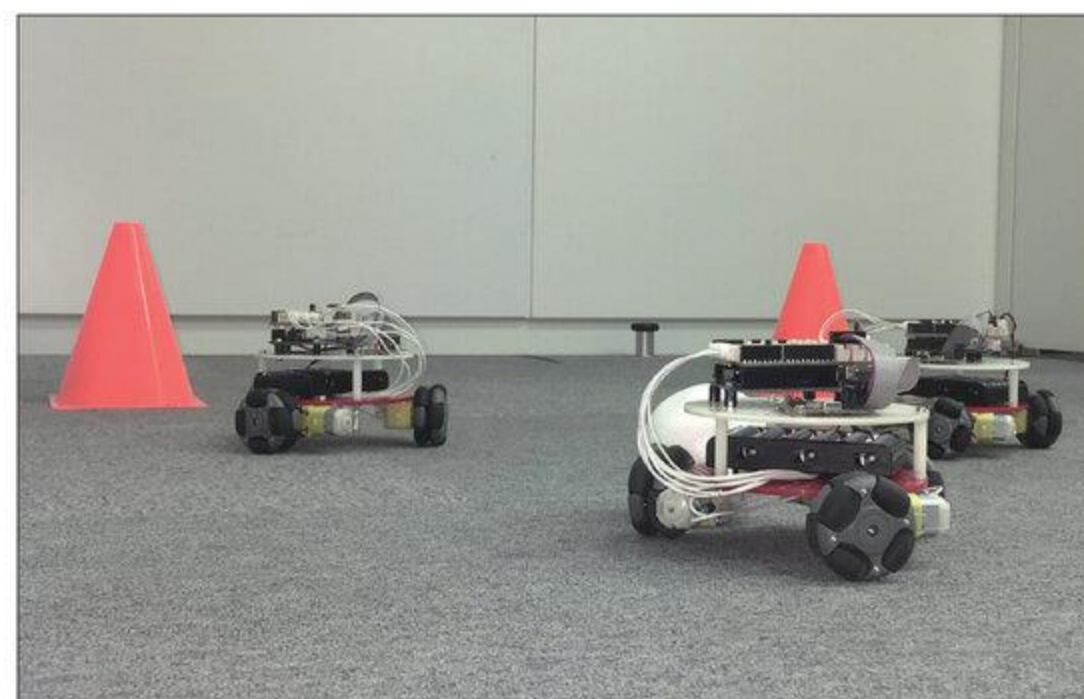
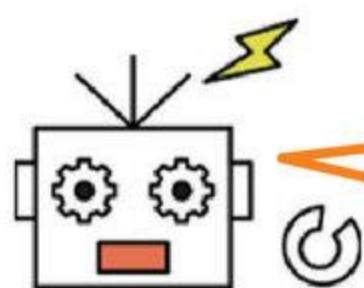


図0-0 サッカー大会



ロボット競技大会の開幕だ！ 誰かこの僕の高みまで登ってコイ！

講

写真はイメージとなります。教室の空きスペースをできるだけ確保して競技を行わせてください。またテーブルの上で実施される場合は落下防止の囲いを固定して落下による故障が発生しないようご注意ください。

オムニホイールロボットの速度調整の際には、急発進に備えホイールを地面から離して状態を確認させてください。

0.1. 必要なもの

前回使ったロボットと、以下のパーツを準備しておきましょう。

前回使ったタッチセンサーは取り外し、リボンケーブルを接続しておきましょう。




| USB ケーブル | 1 | リボンケーブル | 1 | コントローラー | 1 |
|---|---|--|---|---|---|
|  | |  | |  | |

図 0-1 必要なもの

講

障害物とボールを準備してください。障害物は、三角コーンやペットボトル、ボールは 100 円均一ショップで手に入るような発泡スチロールのボールやカプセルトイのカプセルなど、どのようなものでも構いません。

0.2. ^{そうじゅう}操縦プログラムをあらためて読みとく

1) プログラムの確認

前回の授業では if 文を学びました。これは、ある条件を満たすときとそうでないときで、ちがう動作をさせたいときに使う命令でしたね。

今回は if 文の知識を活用し、これまで扱ってきたプログラムをより深く読み込んでみましょう。そして、より狙い通りの動作になるよう、プログラムやロボットを改造する練習を試みましょう。

まず、以下のプログラムを書き込んでみましょう。

コントローラーでオムニホイールロボットを^{そうじゅう}操縦するためのプログラムです。



プログラムの書き込み

RoboticsProfessorCourse1 > OmniWheelRobot4 > Remote3

左のアナログスティックで前後左右の移動を、右のアナログスティックで回転を^{せいぎょ}制御できますね。

実はこのプログラムも、if 文を利用しています。void loop() 内を見てみましょう。

プログラム「Remote3」より^{ぼっすい}抜粋

```
void loop(){
```

(中略)

```
if(ps2x.Button(PSB_L1) || ps2x.Button(PSB_R1)){  
  // L1/R1ボタンをおすと高速モード (数字を変更するともっと速度が上がるよ!)  
  omniBot.move(xx / 2, yy / 2, ww / 2);  
}
```

```
else if(ps2x.Button(PSB_L2) || ps2x.Button(PSB_R2)){  
  // L2/R2ボタンをおすとゆっくりモード  
  omniBot.move(xx / 5, yy / 5, ww / 5);  
}
```

```
else{ // 普通モード  
  omniBot.move(xx / 3, yy / 3, ww / 3);  
}
```

```
delay(10);
```

```
}
```


スティックを倒すとき、同時に L1 ボタンか R1 ボタンを押していればスピードが上がり、L2 ボタンか R2 ボタンを押していればスピードが下がります。つまり、このプログラムには「高速モード」「低速モード」「普通モード」の3種類のモードがあることがわかります。3つのモードの切りかえをフローチャートにまとめてみると、次の図のようになりますね。

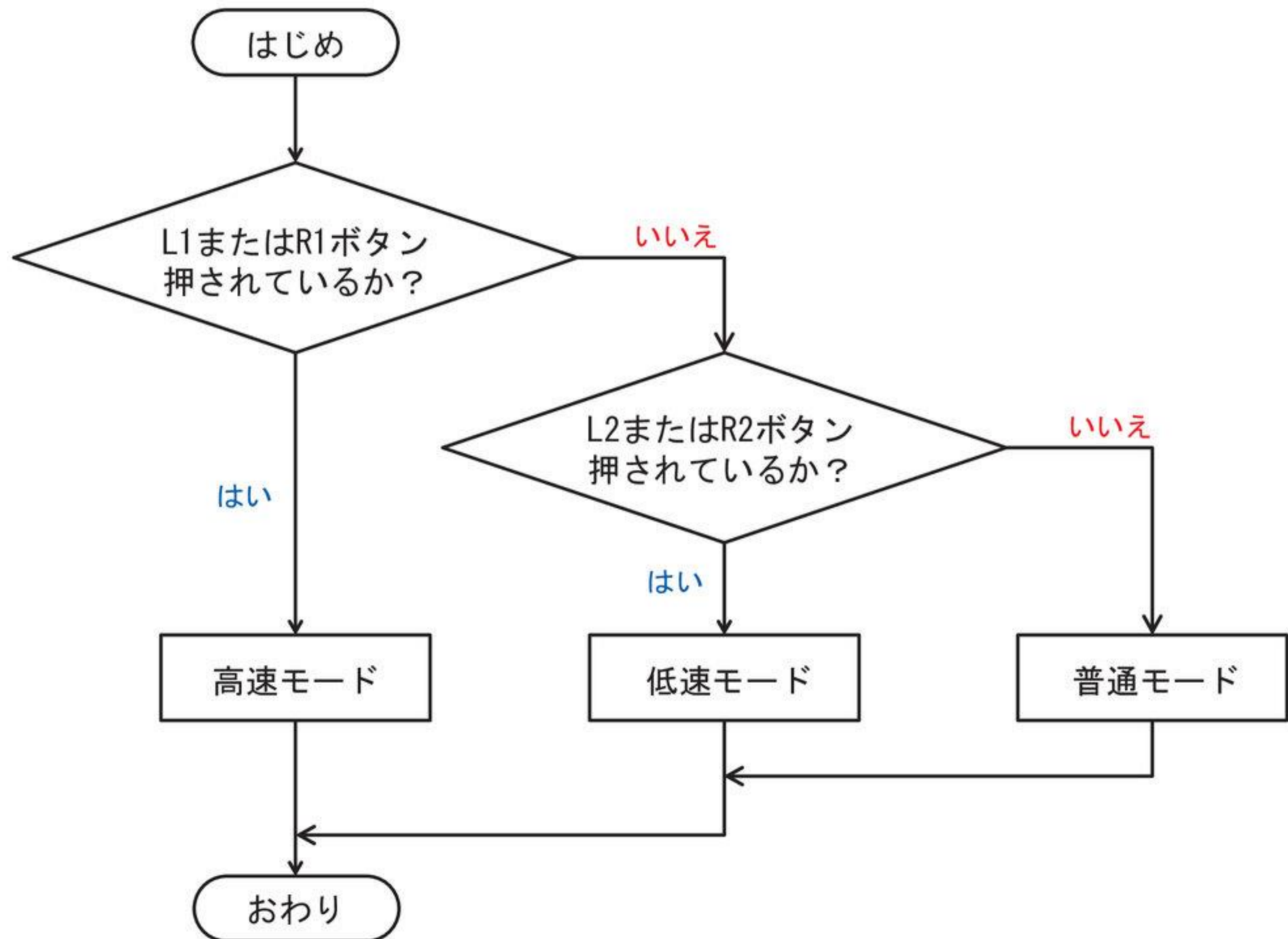


図 0-2 「Remote3」モード決定フローチャート

モードを決定するのに、「L1・R1の判定」と「L2・R2の判定」で、最大2回枝分かれをします。

L2・R2の判定は、L1・R1の判定が「いいえ」だったときだけすればよいですね。

2) else if

ここで、緑枠の部分に注目してみましょう。

命 令 [else if]

動作内容：直前の if が「いいえ」だったとき、指定の条件の判定を行う

使 い 方：else if(touchSensor(D1) == ON){

// もし直前の if が「いいえ」で、D1 センサーが ON ならば

[else if] は、はじめの if が「いいえ」だったとき、さらに他の条件で分岐をさせたいときに使う命令です。[if] と [else] だけでは2通りの分岐しかつくれませんが、[else if] をいくつもかけばさらに多くの動作に分岐させることができます。

[else if] も [else] と同じく、[if] の直後にしかかけません。また [else] とちがい、[else if] は [()] の中に「判定する条件」をかく必要があります。

高速モード、低速モード、普通モードがどのように速度を分けているのか見てみましょう。

□ プログラム「Remote3」より抜粋 ぼっすい

```
omniBot.move(xx / 2, yy / 2, ww / 2);
```

(中略)

```
omniBot.move(xx / 5, yy / 5, ww / 5);
```

(中略)

```
omniBot.move(xx / 3, yy / 3, ww / 3);
```

どのモードでも、左右の移動は `xx`、前後の移動は `yy`、回転は `ww` というのをもとに速度を決めています。

`xx`、`yy`、`ww` は「変数」というもので、左右のアナログスティックを倒すことで様々な数字に置きかわる部分です。

左のアナログスティックを左右に倒すと `xx` が、上下に倒すと `yy` が、右のアナログスティックを左右に倒すと `ww` が、それぞれ -127 ~ 128 の範囲内のいずれかの数字になります。

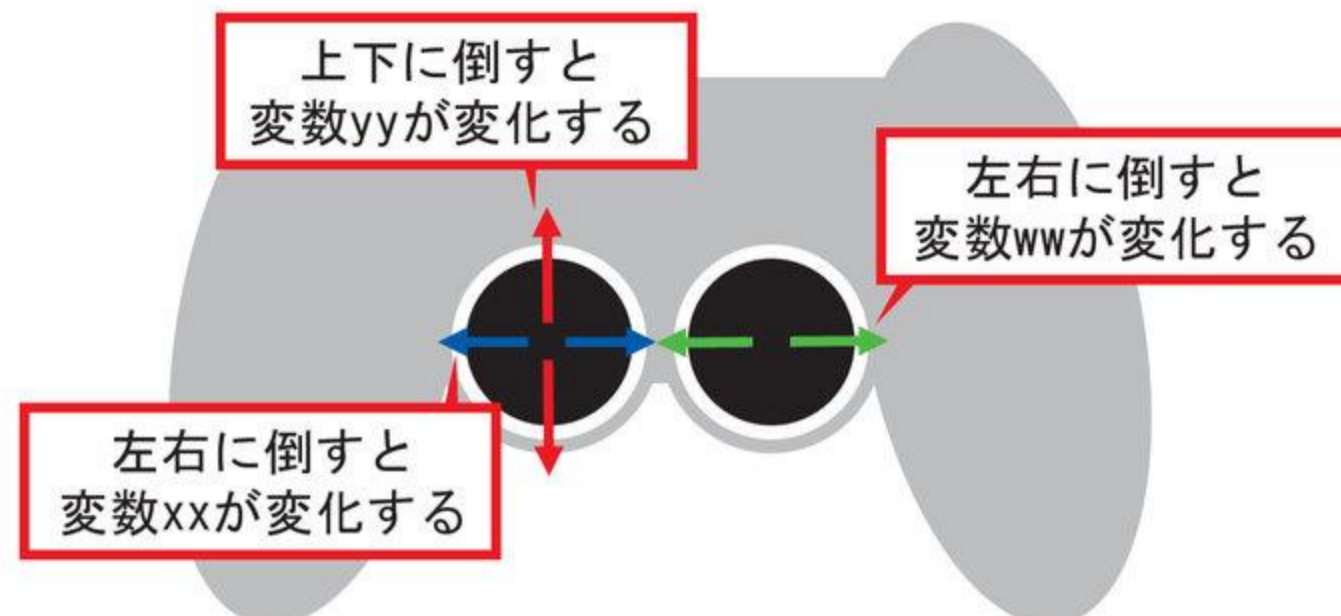


図 0-3 アナログスティックを倒す方向

たとえば、右に目いっぱい倒すと `xx` が 128 になります。上下方向には倒れていないので、`yy` は 0 のままです。

同様に、上に目いっぱい倒せば `xx` は 0 のまま、`yy` が 128 になります。



図 0-4 各変数の値を変化させる

左スティックだけを上に目いっぱい倒したときを考えてみましょう。[xx] が 0、[yy] が 128、[ww] が 0 です。

この場合、高速モードの `omniBot.move(xx / 2, yy / 2, ww / 2);` はどうなるでしょうか。

プログラムの世界では \square は^{じょざん}除算、つまり「わり算」の記号を表します。そのため、`yy / 2` という文は [yy] の値を 2 で割ることと同じです。今回の [yy] の値は 128 なので、前後速度は $128 \div 2 = 64$ となりますね。[xx] や [ww] はともに 0 だったので、2 で割っても 0 のままです。

まとめると、`omniBot.move(xx / 2, yy / 2, ww / 2);` という命令が `omniBot.move(0, 64, 0);` に置きかわることがわかります。

やってみよう!

同じ状況で、低速モードのとき、普通モードのときには、`omniBot.move` の命令はどのように置きかわるか計算してみよう。

💡 ヒント

わり算の結果が小数になるときは、小数点以下の値は切り捨てになるよ!

低速モード : `omniBot.move( 0 , 25 , 0);`

普通モード : `omniBot.move( 0 , 42 , 0);`

スティックの倒し具合が同じでも、モードによって速度が変化することがわかりますね。ちなみに、スティックをななめに倒せば [xx] も [yy] も値が変化するので、ロボットもななめに移動することになります。これは、第3回で学んだ「力の合成」を使えば考えることができますね。



図 0-5 スティックをななめに倒した場合

やってみよう！

プログラム「Remote3」をかきかえ、高速モードをより高速に、低速モードをより低速にしてみよう！

💡 ヒント

`xx / 2` の2はわる数だね。分数で言えば分母にあたる部分だよ！
ちなみに、プログラムの世界で乗算（かけ算）の記号は `*` だから、活用してみると面白いかもしれないね。

講

`xx / 2` を `xx` に直すなど、分母を小さい数にすれば速度は大きくなります。また乗算（かけ算）記号は `*` なので、`xx * 2` などとすればさらに大きい数にすることも可能です。

ただし、速度が100を超える場合はプログラム実行時100に変換されてしまいます。たとえば `omniBot.move(0, 128, 0);` になってしまうような状況だと、強制的に `omniBot.move(0, 100, 0);` として実行されるため、スティックを倒しきる前に最高速度に到達することにご注意ください。

なお、低速モードの分母を大きい数にすれば速度を小さくすることも可能です。ただし小さくし過ぎるとモーターの回転に十分な速度が確保できない可能性があります。

3) 回転操作

なお、プログラム「Remote3」では、直線的な動き以外にも、右のアナログスティックを左右にたおすことでその場で回ることも可能ですね。つまり「並進走行（左のアナログスティック）」と「回転走行（右のアナログスティック）」を組み合わせることにより、自由自在の走行が可能になっているわけです。

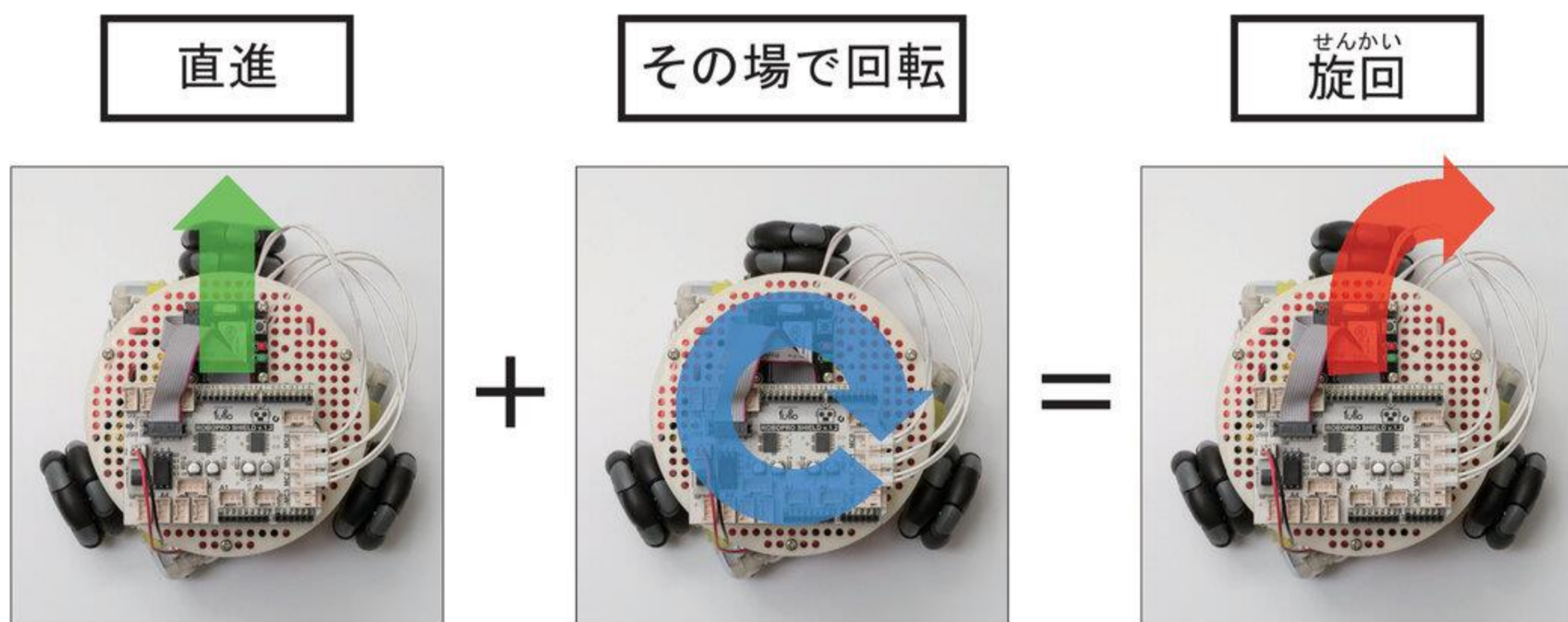


図0-6 せんかい 旋回の動き

講

次のページに入る前に、スペースの広い場所で、三角コーンやペットボトルを並べて、スラローム用のコースをご準備ください。

1. 競技大会（目安 60 分）

1.0. スラローム走行タイムアタック

まずは、うで試しです！ ペットボトルや三角コーンを並べたスラロームコースをつくって、友達と競争しましょう！ 写真では三角コーンを使ってコースをつくっていますが、他の物でも大丈夫です。「スラローム走行なんて簡単すぎる！」という人は、アイデアを出し合っ
て、難易度の高いコースをつくったり、独自ルールを決めたりして競争してみましょう。



POINT

競技例

・タイムアタック

ストップウォッチで走行時間を計測し、最速ラップを目指しましょう！
時間差がつきにくい場合は、何往復かしましょう！

・バトルレース

改造アリ、体当たりアリの無差別レースです！
とにかく先にゴールすれば勝ちです。

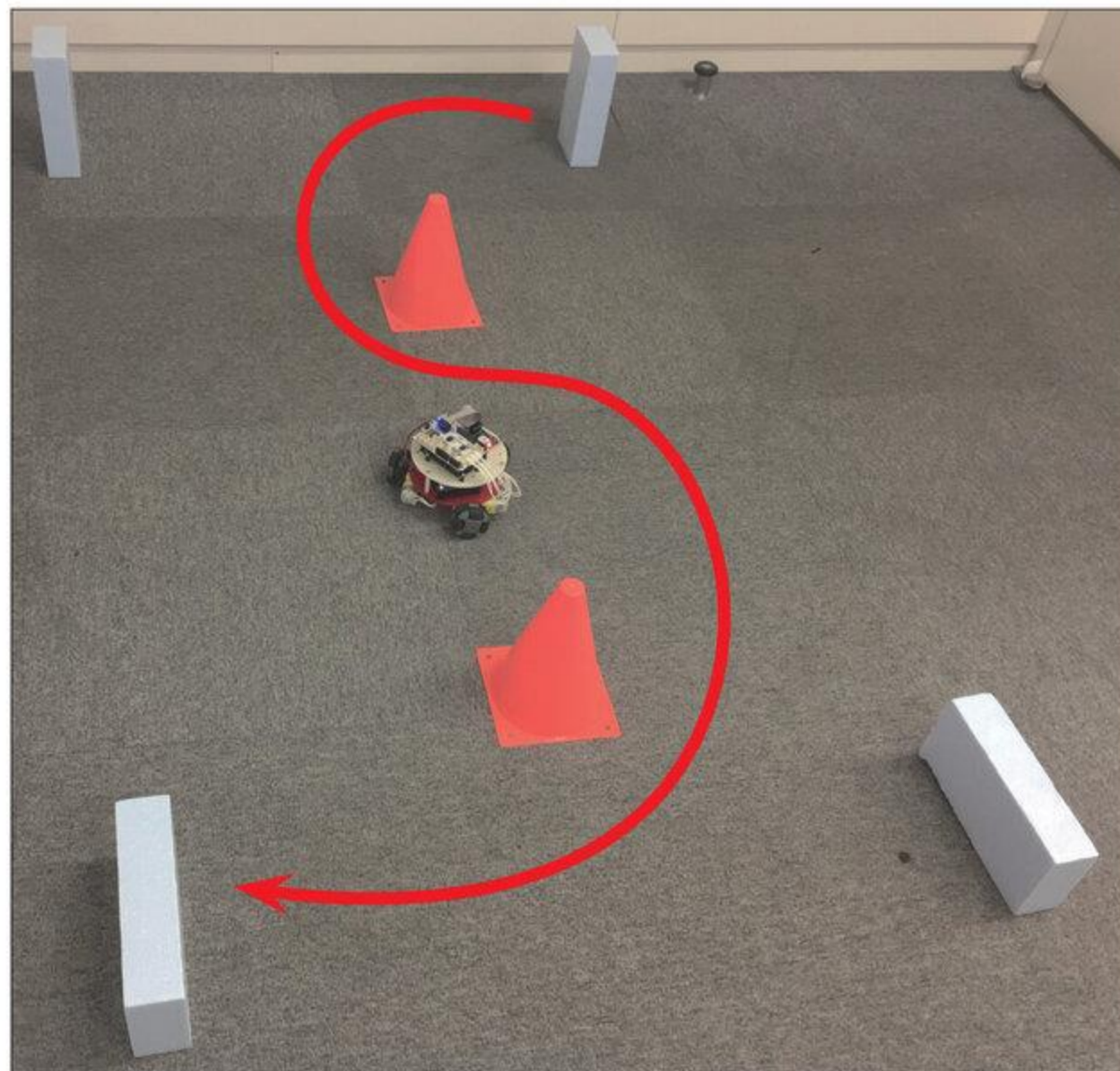


図 1-0 スラロームコースの例

講

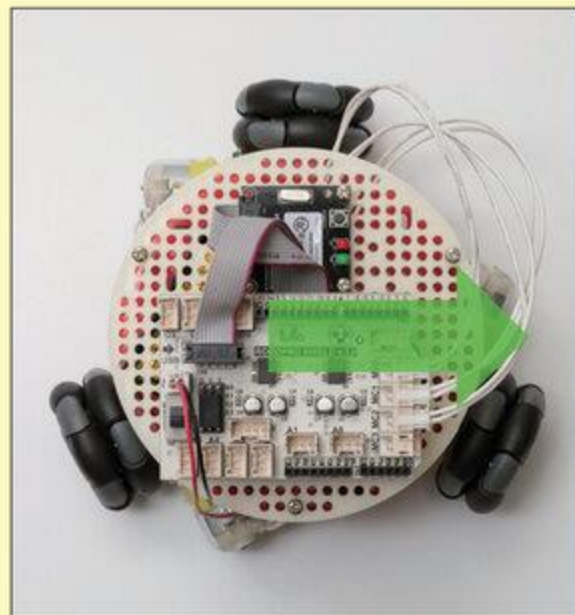
ストップウォッチで、それぞれの生徒のタイムを計ります。タイムを競い合わせる
ことにより、カスタマイズに対する意欲が高まります。



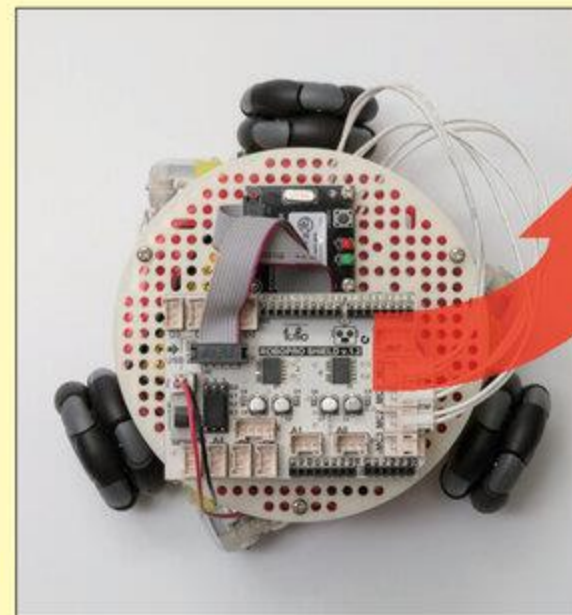
POINT

プログラムを修正しながら気付いた人がいるかもしれませんが、回転速度が大きくなるようにしても、マイコンがモーターに流せる電流には限界があります。もちろん、モーターの速度にも限界があります。また、たとえば、真横に動く速度をだんだんと上げていくと、3個のモーターのうち、速く回転する1個のモーター（MC0モーター）の速度が一番先に限界に達してしまいます。そこから速度指令をどんどん上げていっても、まっすぐ横方向へ進まなくなってしまいます。

真横に進む



曲がってしまう



このように回転のつり合いが取れず曲がってしまうことがあるので、カスタマイズの際は、注意しましょう。

また、速度の命令が正しくても、モーターどうしのほんの小さな性能の差（個体差）が原因で進行方向がずれてしまうこともあります。

以前のテーマでも登場した調整値を、今回も活用しましょう。

```
RPomniDirect omniBot(1.0f, 1.0f, 1.0f, 50.0f);  
// オムニホイール調整用パラメータ
```


1.1. 玉ころがしリレー

続いて、ボールをバトンに見立てたリレー競技です！ ボールをオムニホイールロボットでころがしながら、**図1-1**のような8の字コースを走り、2つのペットボトルの手前でボールの受けわたしをします。やみくもに最大速度でボールを持ち帰っても、バトンリレーに失敗して大きなタイムロスに！ この競技では、速度だけでなく、^{びみょう}微妙な操作も求められます。また、チーム競技ですから、速さ重視の人も正確な動きを重視する人も、一丸となりましょう。

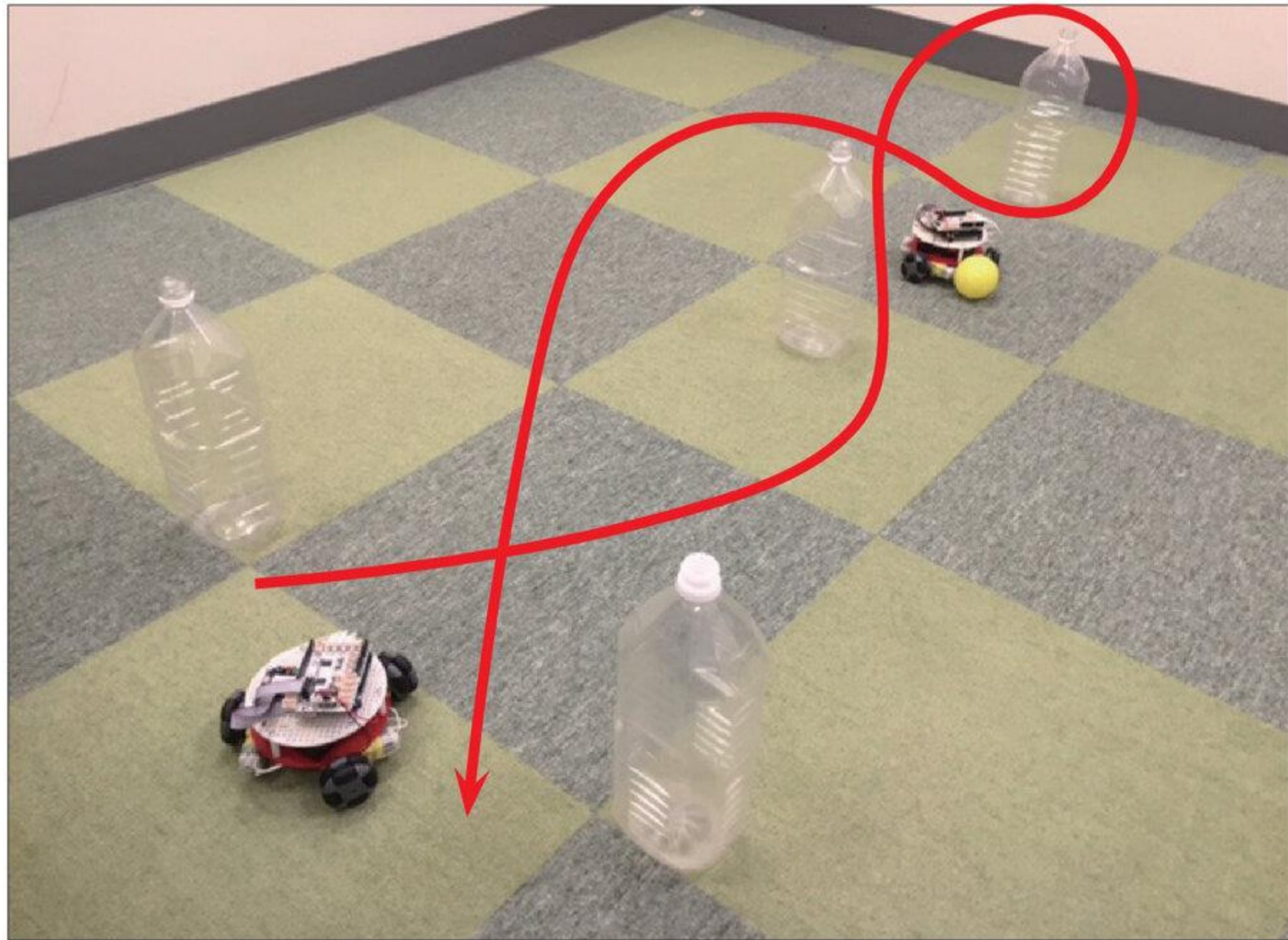


図1-1 玉ころがしリレー

講

ボールの扱いが難しい場合は、センサーL字ステイをオムニホイールロボットの赤円形ボードに取り付けるなど、使用していないパーツで、ボールを保持出来るように改良をさせてください。

1) ロボットの向きが目印

折り返しのあるコースでは、行きと帰りで、ロボットの前後左右が頭の中でごちゃごちゃになってしまいます。どうしても、ロボットの向きがわからなくなって操縦が難しいと思う人は、ロボットに目印をつけてみましょう。ロボットのパーツや色テープ、色画用紙など、目印をつけて、どちらが前かわかりやすくしましょう。

講

ロボットの前方がわかるような工夫を考えさせましょう。写真はセンサーL字ステイとユニバーサルバーでフラッグを取り付けた様子です。

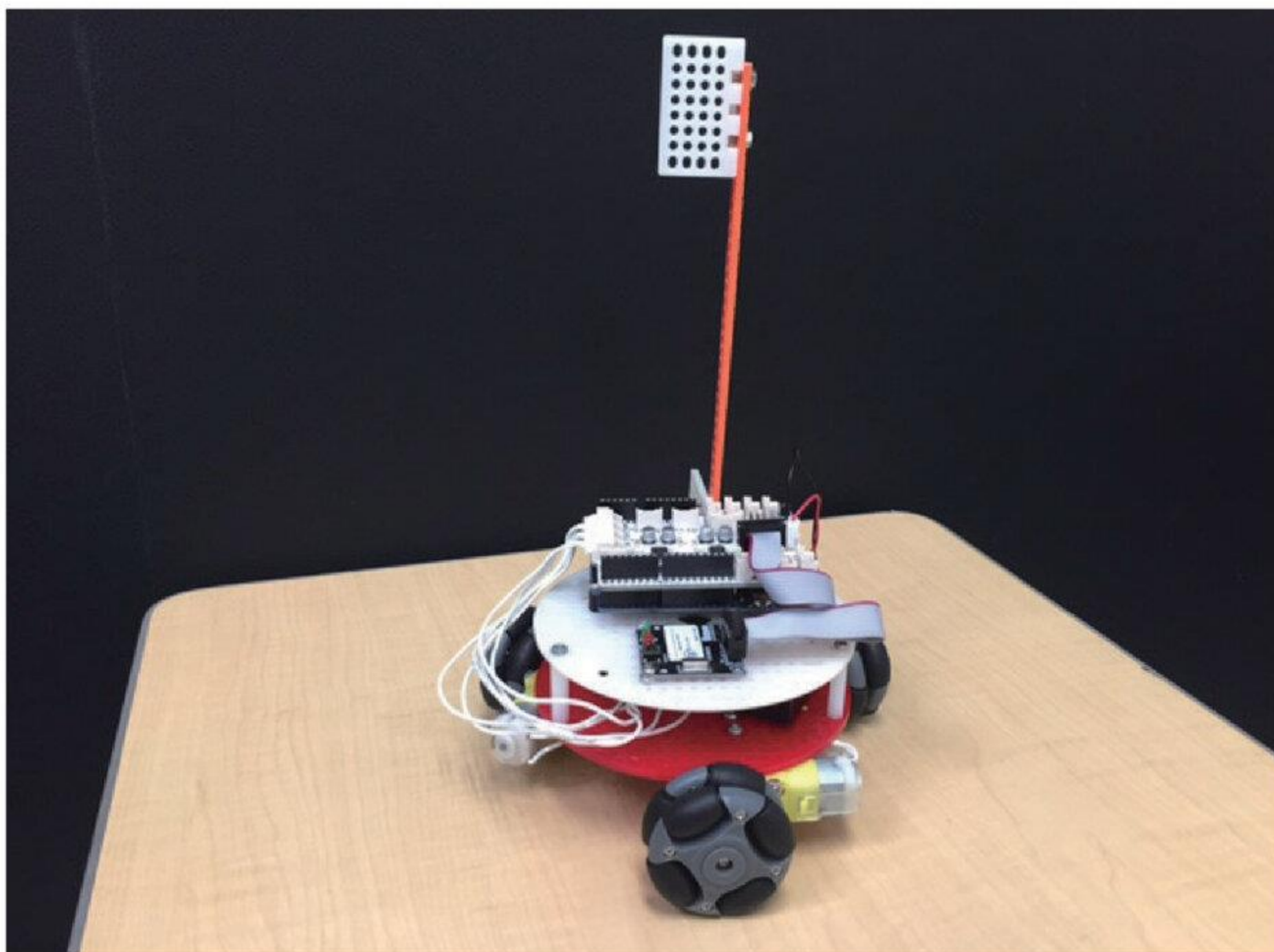


図 1-2 ロボットの向き

2) 速度バランス

速度を重視するか、操縦のしやすさを重視するかは、みなさんの好きな方を選んでください。先ほど説明した、真横に進むときに曲がってしまうという問題も、前に直進するときには起こりません。これは、進ませる方向によって操縦性に差があるということです。多少の操作性が悪くても、速度重視の人は、アナログスティックの操作量に対して、速度を速めに設定してもよいでしょう。

操縦テクニックで補える部分もありますが、細やかな操作が求められるような競技では、L1・R1、L2・R2 ボタンを押しながらモードを切りかえる使いこなしも、勝負を分けるポイントです。

講

競技に応じて、プログラム中の値を調整させます。

1.2. サッカー大会

これまでの競技で、かなり^{そうじゅう}操縦のテクニックが上がっているはずです。

次はいよいよ、サッカーの試合に挑戦です。

ゴールはペットボトルや椅子のあしの間、積み重ねた本など、なんでも構いません。

コートのはしは人数などに応じて調整しましょう。

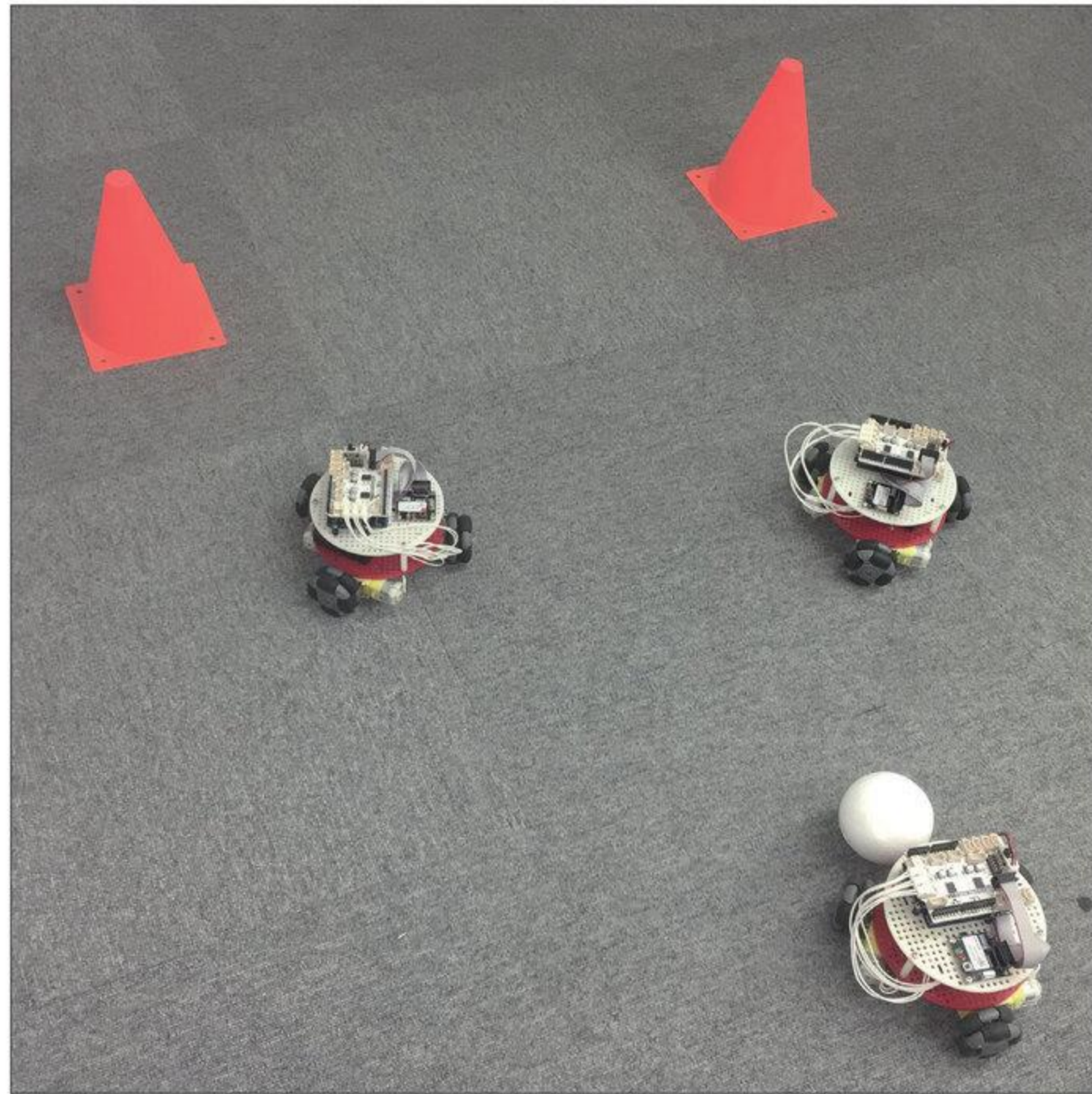


図1-3 サッカー大会

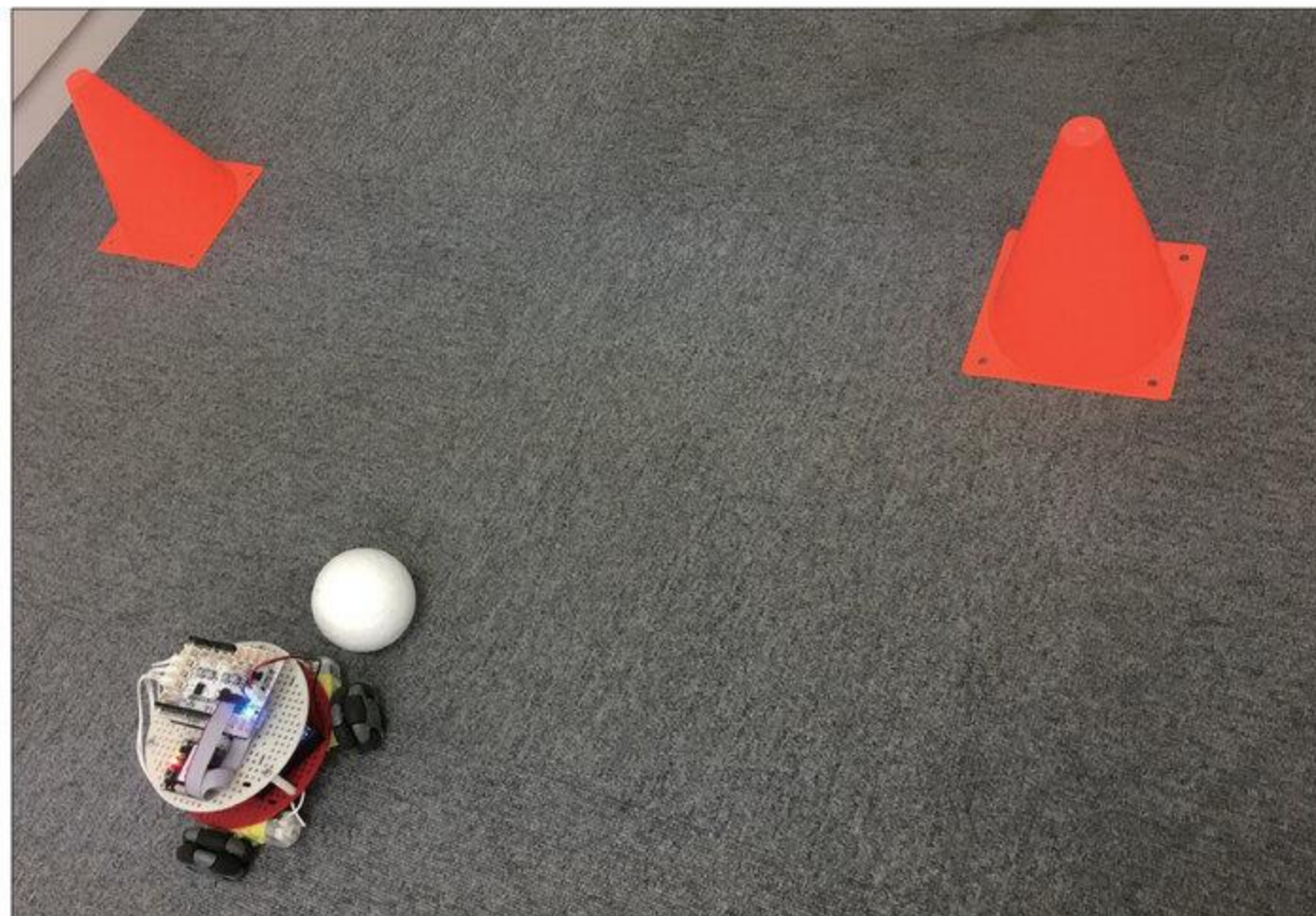


図1-4 ドリブル&シュート

1) ボールの種類

ボールは、サイズや、重さのちがうもの、少しいびつな形をしたものなど、いろいろなものでやってみるのも面白いでしょう。また、一度に2個や3個使っても楽しい試合になりますよ！



図 1-5 いろいろなボール

講

ボールを変えて対戦したり、ルールを変更したりするなど、いろいろな工夫を生徒と一緒に考えましょう。

2) ボールのキープ力

オムニホイールロボットでサッカーをしようとする、ドリブルを失敗してしまうことが多いのではないのでしょうか。

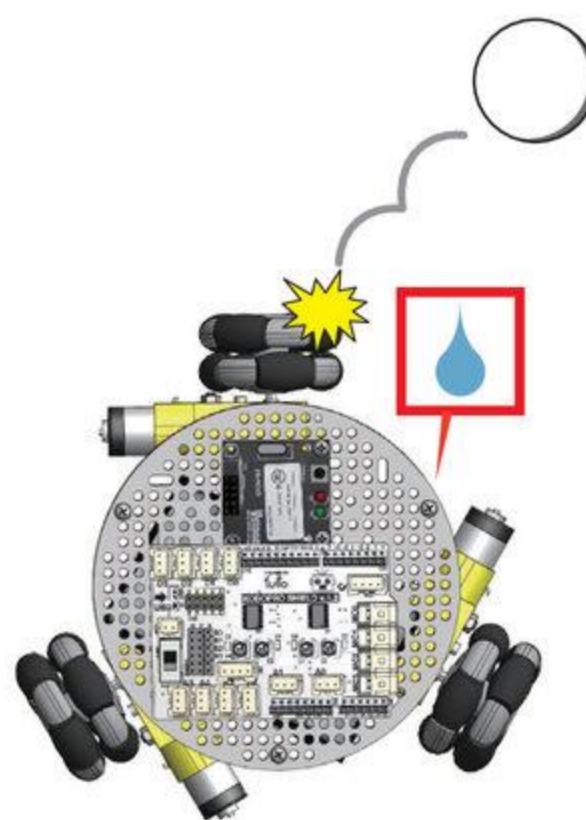


図 1-6 ドリブルに失敗するロボット

オムニホイールロボットは円形なので、もともとボールの扱いは得意ではありませんが、工夫しだいでは今よりはうまくボールを操れるようになるかもしれませんね。

チャレンジ課題

ロボットやプログラムを改造し、よりドリブルに強いロボットをつくってみよう！

 ヒント

そもそも「なぜドリブルが苦手なのか」「失敗の直接の原因はどの部分か」をしっかりと見極める必要があるね！
どこを直すべきかをハッキリさせた方が、改造すべきポイントもわかりやすいよ！

講

もとのロボットだと、① 前方に配置されたオムニホイールによりドリブルの方向が狂いやすい。 ② そもそもロボットが円形なのでボールが横方向に弾かれやすいなどといった問題点があります。

①の問題はロボットを 180 度反転させて走行させれば解決します。ただ操作も反転し非常に難しくなるので、`omniBot.move(xx / 2, yy / 2, ww / 2);` を `omniBot.move(-xx / 2, -yy / 2, ww / 2);` とするなど、各速度の変数にマイナスをつけるとわかりやすくなります。

②の問題であれば、左右に逃げていくボールを捕まえられるようなパーツを増設するという方法も考えられます。

クワガタのハサミ(アゴ)のように左右に棒状のパーツを取り付けるだけでもそれなりにホールドできます。

パーツが長く突き出ている方がホールドはしやすいですが、パーツが折れるなど破損の危険が高まること、ゲーム性が失われることなどから、ある程度の長さ制限を設けるとよいでしょう。

実際のロボットサッカー大会でも、ボールをホールドするための窪みは 2~3cm 程度の深さまでしか認められていないものがあります。

「失敗を分析し、問題点を具体的に見つけられるか」「見つけた問題に的確な対処ができるか」という、PDCAでCとAにあたる部分の力を伸ばすことが目的の課題です。問題点や改善案の内容やクオリティは問いません。生徒が積極的に、かつ主体的に改良に取り組めるようご指導お願いいたします。



コラム いろいろなロボット競技会「ロボコン（ロボットコンテスト）」

ロボコンは、さまざまな競技でロボットを競わせる競技会です。いずれの大会も、競技はちがえど、大会を通じて、ロボット技術のレベルを向上させることを目的に開催されています。

●NHKロボコン（大学ロボコン・ABUロボコン・高専ロボコン）

ときどきテレビでも放送されているロボットコンテストで、毎年変わるさまざまな課題にチャレンジします。アジア太平洋地域の国と地域が参加する国際大会「ABUロボコン」の日本代表選考大会として、「大学ロボコン」が開催されています。また「高専ロボコン」では、全国から若いエンジニアが東京に集まり、対決します。

●つくばチャレンジ

みなさんの生活する「実世界」で、ロボット自身が環境を認識し、自ら行動を決めて動けるようにするための技術を競い合う大会です。人の行き交う実際のつくば市街地の中で、さまざまなミッションをあたえられたロボットが自動走行にチャレンジします。千葉工業大学未来ロボット技術研究センター（fuRo）のロボットは、この大会に参加して、3年連続で完走を果たしました。

●マイコンカーラリー競技会

ラインが引かれたコースを走り、最速ラップを競う大会です。車体の先端についたセンサーでラインを認識しながら、目にもとまらぬ速さで駆けぬけるマイコンカーの走行は圧巻です。「ジャパンマイコンカーラリー」と「ルネサスマイコンカーラリー競技大会」の2つの大会が開かれています。

●マイクロマウス大会

はば9センチの通路でつくられた複雑な迷路の中を、ロボットが自ら探索し、自分の力だけでゴールまで達する最短時間を競う競技です。光センサーや超音波センサーを使って、かべや迷路の構造を理解して、最速タイムを目指します。手の平にのるほどの小さなロボットが、目を見張るスピードで迷路を駆け回り、ゴールに到達します。

●RoboCup レスキューリーグ

RoboCupはロボットにサッカーをさせる大会ですが、レスキューロボットが仮設の災害現場で災害救助活動のスピードと精度を競い合う「レスキューロボットリーグ」というカテゴリーもあります。RoboCup世界大会で優勝した実績もある千葉工業大学未来ロボット技術研究センター（fuRo）のレスキューロボットは、東日本大震災後、福島原発建屋内に国産ロボットとして初めて投入された災害対応ロボットです。

2. まとめ（目安5分）

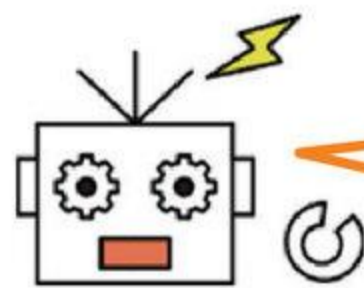
オムニホイールロボットの授業はいかがでしたか？

ロボットづくりを通じて、機械の組み立てや構造、プログラミングの基礎、「感じて」「考えて」「動く」というロボットの要素、力の合成やモーメントといった物理学の知識、if-then ルールに代表される「条件分岐」の考えかたなど、いろいろなことを学びましたね！
また、ドリブルの失敗とその改良から、「問題点を見つける」「見つけた問題に対応する」という経験を積むことができましたね！

この「やってみて、失敗して、分析して、改良する」という流れは、ロボットづくりだけでなく様々な場面で登場します。ロボプロの授業は、ロボットの知識だけでなく、生きていくなかでとても大切な力を伸ばすことができるのです！

もちろん、ロボプロで取り扱うのはオムニホイールロボットだけではありません。これからも色々なロボットを製作するので、その中で広く深く、さまざまな力をつけることができますよ！

次回の授業を楽しみにしててください！



ロボットは、なにもオムニだけじゃないぜ！！

講

- 以下の授業の目標を再確認します。
 - ・操作しやすいロボットにカスタマイズする
 - ・競技大会をやる
- 今回のタームで学んだ感想や面白かったことなどを、生徒から聞いてみましょう。
- 次回、1年目コースは「不思議アイテム I -1」、2年目コースに進級する場合は「アームロボット」になります。

《次回必要なもの》

次回は、以下のパーツを持ってきてください。



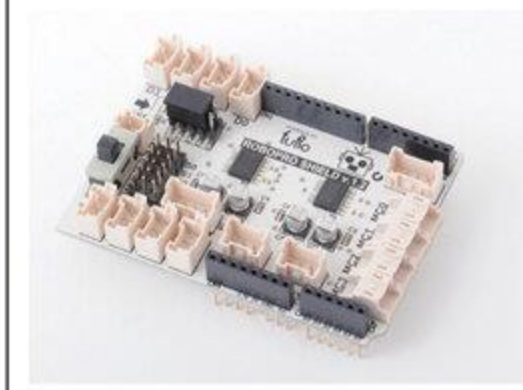


| | | | |
|--|--|---|---|
| USB ケーブル 1 | マイコンボード 1 | ロボプロシールド 1 | マトリクスLEDシールド 1 |
|  |  |  |  |
| マトリクスLED 1 | | | |
|  | | | |

図 2-0 次回必要なもの