

10. チャレンジロボ Challenge Robo RDS-TEC34



ドリブルサッカー競技をできるように目指したオリジナルロボット製作セットです。

- ①. 競技場のフィールドに示された白線ラインを識別し、ラインからはみ出さないで動作をします。
- ②. 変調赤外線を発光するボールを目標として捕捉行動
- ③. 目標(ボール)をドリブルして行動する等

ドリブル機構を装備し、実践競技の場で、本格的に活躍できるロボット作りにチャレンジします。

構造体にハニカム構造の素材を使用し、高強度・超軽量を目指しています。

完成後には、方向を把握する「コンパス(地磁気)センサ」を加えて、OWNゴールを防いで競技するロボットへの改造ヒントを掲載しています。

重量:本体640g
(電池搭載時990g)

搭載コントローラボードにより

Typelll (液晶なし、使用可能モータ数 (DCモータ×4、サーボモータ×4)、超音波センサ増設可能)

Typell+ (液晶付、使用可能モータ数 (DCモータ×4、サーボモータ×4)、超音波センサ増設可能)

のモデルがあります。

すべてのモデルに

・音センサ ・明るさセンサ ・加速度/ジャイロセンサ ・スライダー を搭載しており、これらを利用して各種制御を行えます。

10.1. 部品の見方、使い方

構造体ハニカム構造板



ロボット構造体に、高強度・超軽量なハニカム構造の熱可塑性樹脂板を使っています。
六角形のセルの集合体で、力学上もとてもすぐれたサンドイッチ部材です。板の表面(スキン)は、とても薄いPP材ですので、簡単に穴など加工がしやすく、錐やドリル、ハンダこて先などで穴加工が可能です。

ロボットプレートには5mm角の方眼模様を印刷したシートを張っていますので、部品配置などに利用できます。方眼シートは剥がすことが可能です。(注:シート印刷は水性です、濡らすと、インキがにじみます。)

1. 部品サイズ表示

ネジやナットのサイズ表示は以下のとおりです。

Screw

■ナベネジ: 頭がナベを伏せたような形の名称です。

表記:M3 × 10mm

「太さ(直径)3mm、長さ10mm」という意味です。



長さ
※ネジが切ってあるところの長さです。

Low Head Machine Screw

■超低頭ネジ: 頭が低く平らな形のネジの名称です。

表記:M3 × 4mm

「太さ(直径)3mm、長さ4mm」という意味です。



実物写真

※ネジが切ってあるところの長さです。

Flat Head Screw

■皿ネジ: 頭が皿のように平らな形のネジの名称です。

表記:M3 × 10mm

「太さ(直径)3mm、長さ10mm」という意味です。



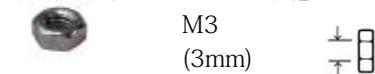
実物写真

長さ
※ネジの頭からの長さです。

■ナット Nut

表記:M3

「太さ(直径)3mmのネジ用」という意味です。



実物写真

■共通で使う上記以外のパー



・座金付組ネジ・スペーサ・バネ座金・平座金

RDS-TEC34 チャレンジロボ

2. 電子基板使用時の注意

①. 基板表面のピン同士をショートさせないこと。

	表面に装備しているサーボピン等をショートさせないようにしてください。ショートさせると基板が壊れます。
--	----------------------------------------------------



②. 裏面には必ず隙間を空けて使う。

	電子基板は裏面にも微小な部品や回路パターンが配置されています。圧力を加えると破壊され、また金属製の物体に触れるとショートして基板が壊れるなどの原因になります。
	ネジ・ナットを使って隙間を設けるなど工夫して、裏面の部品や回路パターンが、取付個所などに接触しないように注意してください。

③. 規格範囲内の電圧で使用する。

	電子回路に使用されている電子部品は、規定値の電圧で動作するように設計されています。電源電圧2V～5Vと指定されている場合、指定範囲より電圧が低いと動作が誤ったり、高すぎると回路が破損したりしますので指定範囲内の電圧でご使用ください。
--	----------------------------------------------------------------------------------------------------------------------

本製品のコントローラボードの電源電圧規定値は

回路用:	4.5V ~ 6.0V
モータ用:	2 電源での使用時: 使用モータの適正電圧を加えてください。

④. 電源電圧極性を間違えない。

	電子回路に接続する電源電圧のプラス/マイナスを間違えないように注意ください。間違えると部品が壊れます。
--	-----------------------------------------------------

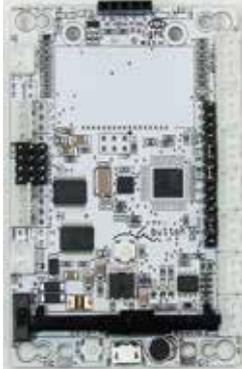
⑤. 水分大敵! 電子回路は、水をかけると壊れます。

	ロボット製作を行っているとき、夢中になって気がつかないうちに、近くに置いていたカップ容器などを倒したりすると、中に入っていた液体が電子回路にかかり、回路がショートして壊れるなどの事故があります。同じテーブルや机の上に、液体が入っているカップ容器は置かないようにします。
--	----------------------------------------------------------------------------------------------------------------------------------------

10-2. 事前準備 部品確認

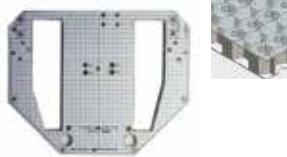
10.2.1a. チャレンジロボRDS-TEC34TYPEIII パーツリスト

□コントローラRDC103_TYPEIII 1台



取付部品	樹脂スペーサ 10mm	4本
	ナベ小ネジ M3x6mm	4本
付属部品	ワッシャー M3x8	8個
	ナベ小ネジ M3x10mm	4本

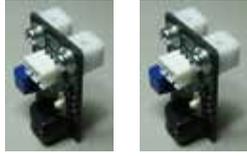
□ベースプレート RDP-TEC_Base34



□トッププレート RDP-TEC_Top34



□アナログ赤外線センサー JES-7023VAD 2台



下記の部品を取り付けています。

センサ接続ケーブル 30cm RDP-832W	2本
レジンスペーサー8mm角 VAD-308	4個
ネジM3x6mm座金組込	4本
ネジ M3x12mm	4本
ワッシャー M3x8	4個

□支柱セット



ネジM3x120mm	6本
樹脂スペーサ黒色20mm	12本
金環スペーサ黒色25mm	18本
ナットM3	6個
平ワッシャー M3x8	24枚

□変調赤外線センサー RDI-203JR 3台



センサ接続ケーブル 30cm RDP-832W	3本
ネジ M3x12mm	12本
ナット M3	24個

□電池ケース (単2x4) 6V



ナベ子ネジ M3x20mm	2本
ワッシャー M3	2個
ナット M3	2個

□ギアードモータ&ホイール RDO-502 2台



モーターケーブル30cm	2本
ナベ小ネジM3x8mm	2個
下記の部品を取り付けています。	
マウンター金具	2個
ナベ子ネジM3x30mm	4本
座金組込ネジM3x10mm	6本
ナット M3	4個
レジンスペーサー8mm角	2個
座金組込ネジM3x6P=3	2個
ワッシャー M3x8	14

□プログラム開発環境DISK



DISK収録内容

- ・プログラム開発環境
- ・インストールガイド
- ・ユーザーガイド

□マイクロUSBケーブル RDP-824 1本



□補修用ネジナット

M3x12mm	8本
M3x15mm	8本
M3ナット	16個

10.2. 1b ドリブル部品リスト

□ミドルプレート RDP-TEC_Middle34 1枚



□ドリブル機構 1台



座金組込ネジ M3x15	4本
ワッシャー M3	4個
ナット M3	8個

□外装プラスチック段ボール 1セット



75x90mm
2枚

□測距センサ 1セット



ナベ子ネジ M3x12	4本
トラス木ネジ M3x6	2個
ナット M3	2個

□ハニカムプラダンプレート



35x100mm	1枚
35x45mm	2枚

□スイッチ付電池ケース (単3x3本) 4.5V



皿ネジ M3x12mm	2本
ワッシャー M3	2個
ナット M3	2個

◆◆◆ 一緒に使う別売品 ◆◆◆

◆I²Cコンパスセンサ RDL-5883L



I²Cセンサケーブル付

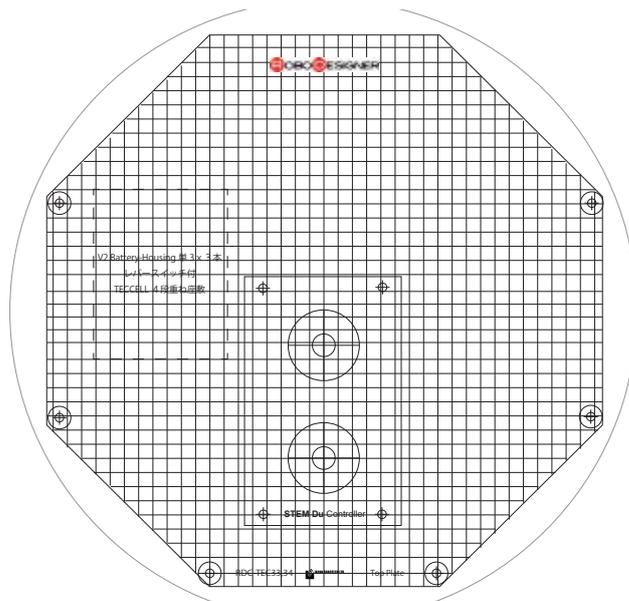
◆ソナーセンサ (超音波) RDI-HCSR04



差し込みピン式 (ソケットはコントローラに装備)

10.2.2. RDS-TEC34チャレンジロボプレート設計図

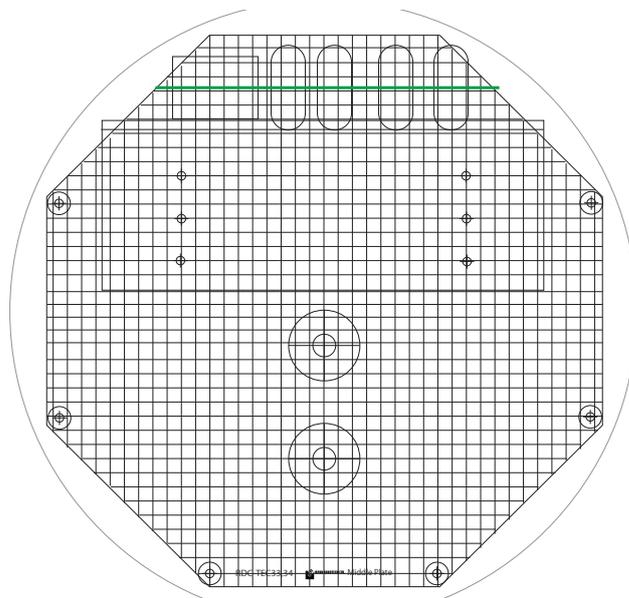
1. トッププレート (3段目)



配置部品

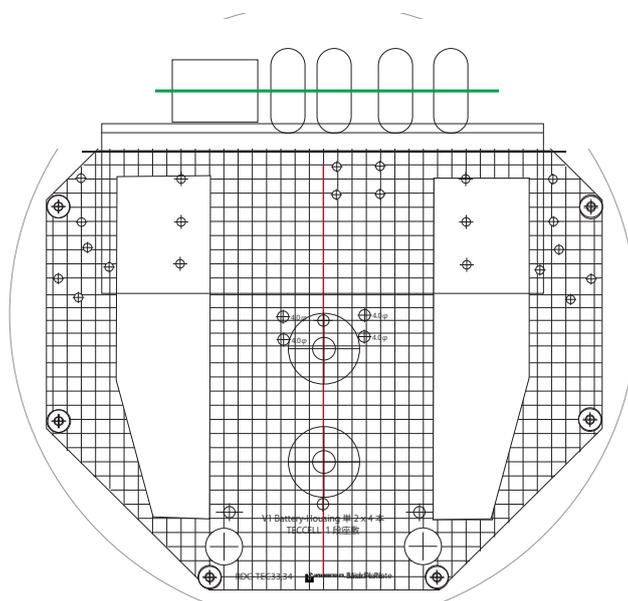
- コントローラ
- M3、M4 電源

2. ミドルプレート (2段目)



- ドリブル機構
- ボール距離センサ

3. ベースプレート (1段目)



- 駆動機構
- ボールセンサ
- フロアセンサ
- 回路・駆動用主電源

10.2.3. RDC – 103TYPE III仕様

- ・4 個のDC モーターを使用し、PC から独立して動く自律型ロボットを作成可能。
- ・LED / 光センサー、音センサー、加速度 / ジャイロセンサー、スライダを基板ボード上に搭載しており、これらを利用して各種制御を行えます。
- ・外部超音波センサー 増設可能 ・I²C
- ・外部アナログセンサー 6 個まで接続可
- ・通常はスケッチ(プログラム)に合わせて配線します。
- ・サーボモータ 4 個まで接続可能
- ・2 電源式(M3,M4 モータ電源 最大12V まで使用可能)
- ・LCD モニター必要な場合は、RDC-103TYPE III +をご利用ください。

Control board RDC-103TYPE III

- ・It's possible to make the autonomous robot which becomes independent of a PC using 4DC motor-and moves.
- ・It's equipped with LED / Light sensor-, Sound sensor-and the Acceleration / Gyro sensor and a Slider on the board, using these, you can control variously.
- ・It's possible to connect Ultrasonic Sensor- (sensor socket use), I²C----(separate sale part)
- ・It's even possible to connect 6 of outside analogue sensor (A0,A1,A2,A3,A4,A5).
- ・It's even possible to connect 4 servomotors.
- ・2 power supply system (Even at most 12 V of motor power supply is practicable. M3,M4)

マイコン / ATMEGA32U4	MCU / ATMEGA32U4 Clock 8MHz	http://media.digikey.com/pdf/Data%20Sheets/Atmel%20PDFs/ATmega16U4,32U4.pdf
加速度センサ / ジャイロ	Acceleration / Gyro sensor MPU-6050	http://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/
音センサ	Sound Sensor SPI XCM6035P	http://www.buzzer.com.hk
スライダボリューム	SlidePotentiometers Alps RS30H121	http://www.alps.com/WebObjects/catalog.woa/J/HTML/Potentiometer/SlidePotentiometers/
明るさセンサ	Light sensor Everlight PT12-21C	http://www.everlight.com/file/ProductFile/PT12-21C-TR8.pdf

デジタル入出力 Digital in/out

入出力端子を使用したい時はピンで接続します。

ピン番号	ヘッダーピン	記号	解説
13		13P	サーボ / 白色LED/PWM 出力可能 R/C servo motor / White LED / PWM output
12	M4	12	サーボ / ボタン R/C servo motor / Button
11	M3	11	サーボ / 超音波 / 赤外線LED R/C servo motor / UltraSonic / InfraRed
0		0	サーボ / ブザー / シリアルRX R/C servo motor / Buzzer / Serial RX
1		1	サーボ / LCD RS / シリアルTX R/C servo motor / LCD RS / Serial TX
10		10P	サーボ / LCD CS / PWM 出力可能 R/C servo motor / LCD CS / PWM output
6			サーボ / M1 PWM 制御 / PWM 出力可能 R/C servo motor / M1 PWM control / PWM output
5	M1 (0.5A 程度)		サーボ / M1 制御 / PWM 出力可能 R/C servo motor / M1 control / PWM output
4			M1 制御 M1 control
7	M1		M2 制御 M2 control
8	M2 (0.5A 程度)		M2 制御 M2 control
9			M2 PWM 制御 / PWM 出力可能 M2 PWM control / PWM output
電源コネクタ			
	V2		電源コネクタ M3,M4 Power Connector
	V1		電源コネクタ 回路 Power Connector M1,M2
	○		電源スイッチ Power Swith

加速度 / ジャイロ / 温度センサ (I²C)

Accelerometer/Gyroscope

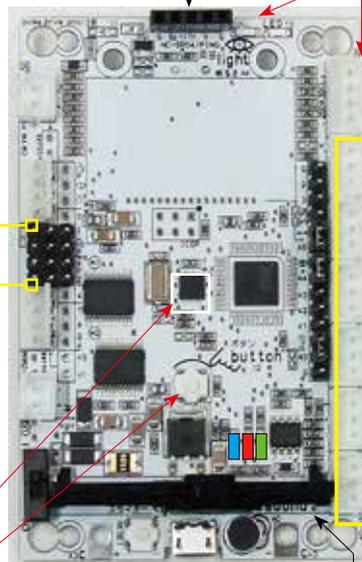
ボタン button

LED

ON	青色	電源確認Blue LED
RX	赤色	通信確認Red LED
TX	緑色	通信確認Green LED

増設可能

★超音波センサ Ultrasonic sensor (差し込んで使用します。)



- リセットスイッチ Reset Button
- USB コネクタ USB connector
- 音センサ Sound Sensor
- みの虫クリップ用端子 Terminal for clips (抵抗等測定)
- スライダ (可変抵抗) Slider resistance

明るさセンサ Light sensor

発光 白色LED / 受光 フォトトランジスタ

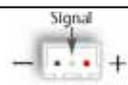
I²C コネクタ 3 SCL, 2 SDA

アナログ入力 Analog input

A0からA5 の6 ポートがあります。0 から電源電圧 (3.3V) までの入力電圧を1024 段階で読み取ります。センサやボリュームなどを接続することができます。また、スケッチで設定を変更するとデジタル入出力ピンとして使うことができます。

ピン番号	ヘッダーピン	記号	解説
		G	⊖ライン
		2	SDA
		3	SCL
		3.3V	電源 ⊕ライン Power
	↔		ボード搭載スライダ接続端子 Slider connect pin
		A5	アナログ入力ポート Analog input
			ボード搭載明るさセンサ接続端子 Light sensor connect pin
		A4	アナログ入力ポート Analog input
		A3	アナログ入力ポート(プルアップ) Analog input (PULL UP)
		A2	アナログ入力コネクタ Analog input terminal
		A1	アナログ入力コネクタ Analog input terminal
			ボード搭載音センサ接続端子 Sound sensor connect pin
		A0	音センサ Sound Sensor
		G	⊖ライン

接続コネクタ JST
connector XH-3B

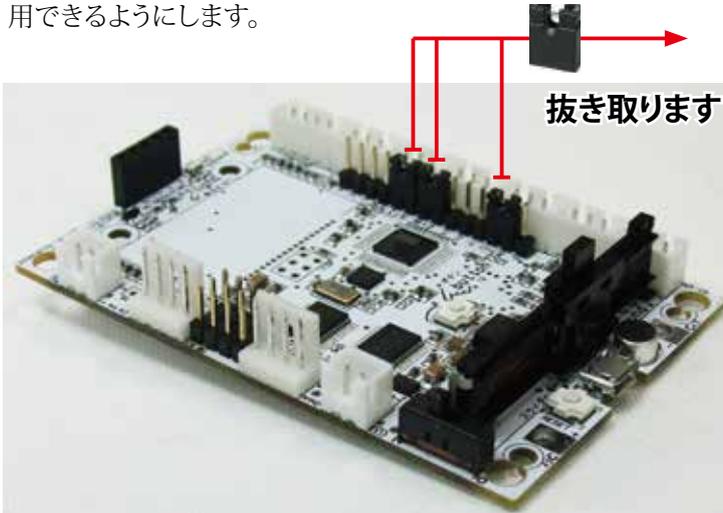


10.2.4. 部品の準備

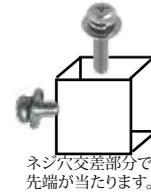
(1). コントローラボード入力端子設定

本機で接続するセンサが使用できるように、コントローラボード入力を設定します。
 * 製品出荷時は、基板搭載センサ(音センサ、明るさセンサ、スライダ)に使用設定していますので、設定用ジャンパーピンを外し、基板外部接続センサが使用できるようにします。

抜き取り外した設定用ジャンパーピンは、後日のために保管しておいてください。基板搭載センサ使用設定時に必要になります。



取り付け時点の使い方



(2). アナログ赤外線センサ(フロアセンサ) JES7023VAD



(3). ボールセンサ制作

①. 変調赤外線センサの4隅にねじを立て、スペーサとしてナットを1段取り付けます。
 使用ネジ:M3x12mm



※ 回路の保護のため、ナット1段のスペーサを必ず入れてください。



- ①. 画像の部品を使って、左のアナログ赤外線センサを組み立てます。
- ②. 垂直取付のために、正方形スペーサ取付時には、ネジ穴が上に向いているように組付けます。
- ③. ロボット組み立て時に使うネジと正方体の中で交差する時にねじ同士の干渉を防ぐ目的でネジの長さ調整をするために、座金組込ナベネジM3x6P=3に、ワッシャーを1枚加えて取り付けます。

8mm正方形スペーサ 発光用LED

(4). ハニカム構造の熱可塑性樹脂板加工方法

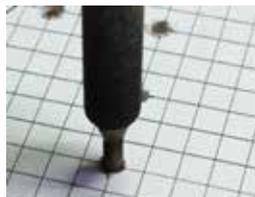
六角形のセルの集合体で、力学上もっともすぐれたサンドイッチ部材ハニカム構造板の加工。

板の表面(スキン)は、とても薄いPP材ですので、簡単に穴など加工がしやすく、キリヤドリル、ハンダこて先などで穴加工が可能です。



・カッターナイフ

定規を当ててカットします。数度切れ込みを入れて、切ります。



・はんだこて

こて先の熱を使って穴加工に使用します。穴は、こて先径の大きさで開きます。こて先径4.0mmが多い。



ねじギリ

穴を大きくしたい時使用。ネジ穴が作れます。



丸ギリ

ケガキ作業・精密・細作業に使える、鋭い丸形先端。



彫刻刀(切出刀)

両手で押し出しながら切ります。木を彫刻する時に使用。



ドリル

M3ビスを、ねじ込む時の穴は先端2.5mm径のドリル刃を使用。

(4). ギアボックス組立て ギアボックスを左右対称で作成します。

①. マウンター取付 [Assemble of a Mounter]

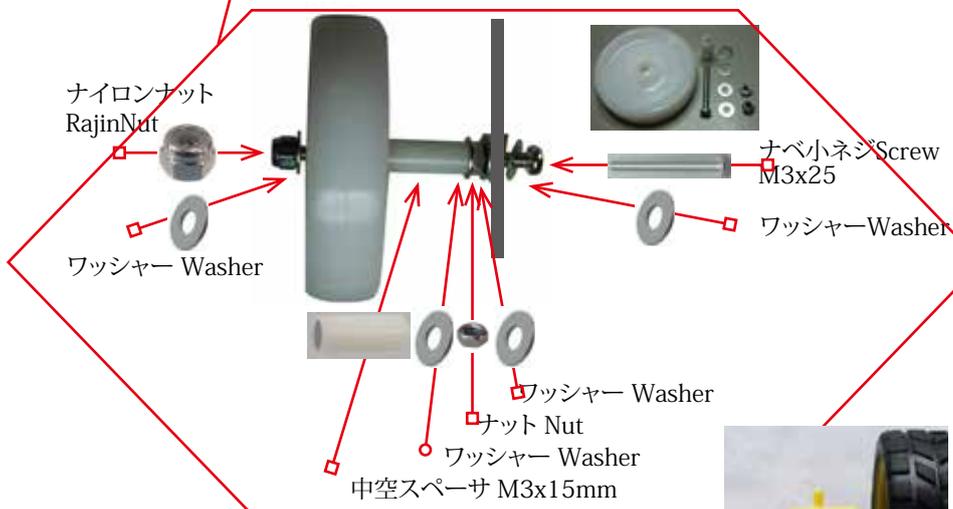
ギアードモータにマウンターとタイヤを取り付けます。

(検査のため取り付けている場合もありますが、その場合ナットを外して、マウンターネジ穴の方向性を図に合わせて入れ替えてください。)

マウンター金具には方向性があります。図のネジ穴の方向を確認して取り付けてください。左右対称になるように作ります。(左右のマウンターのネジ穴が上を向く)



②. 後輪タイヤの組み込み [Assemble of a Tire Wheel]



③. 前輪タイヤ組み込み

1). モータ出力軸とホイールの差込口の長円形の方
向を合わせます。

2). 回転軸を支えてホイールを差し込みま
す。

3). 最後に、図のように、出力軸の反対側を台に
当てて、タイヤを上から手の平で押さえて差し
込みます。

ギアボックス内部ギアに無理な力を
加えないように注意ください。



ホイールの組み込み図 Fig Wheel assembly

Geared motor assembly



Down view



Side view



完成品: 2 個作ります。

左右対称: マウンター穴位置
Finished goods: 2 are made.
A gearbox is made by
symmetricalness.

組み込んだホイールを取り外
したい時



ギアボックスとホイールの間に
ラジオペンチの先を差し込み、
ラジオペンチをゆっくりとこ
ねてホイールを真上へ抜きま
す。

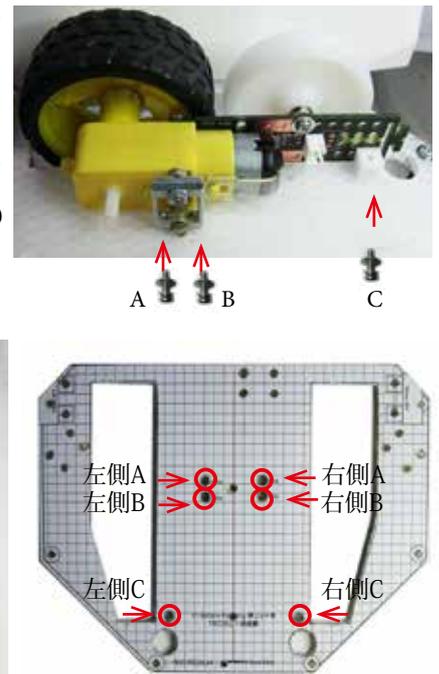
※手で無理やり、タイヤを曲げ
るとシャフトが折れて壊れま
す。

10-3. ロボットベースプレートへの組み込み

10.3.1 ギアードモータを取り付ける。

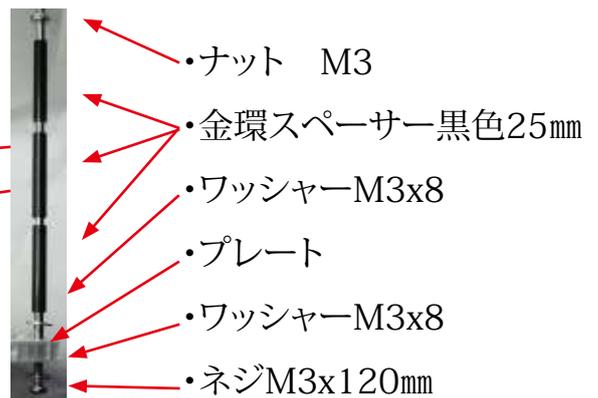
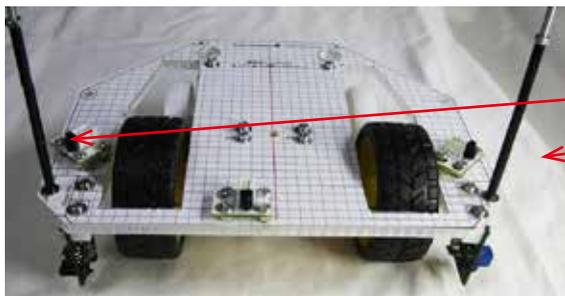
1. 準備で作製したモータを、ベースプレートに取り付けます。
2. 取り付け位置は、図の配置を参照ください。
 A: M3x10mm座金組込ネジにワッシャーを通して、マウンターのネジ穴位置に合わせて仮止めします。(A,Bの位置)
 B: レジンスペーサ8mm角(Cの位置)にM3x10mm座金組込ネジをワッシャーを通して取り付けます。
 A,B,Cのネジ穴位置を合わせた後に、ネジを締め付けて固定します。

A,B,Cの位置での仮止めが完了後、モータ取付角度で左右タイヤの並行性を確認して、A,B,C左右で6か所のネジを締め付けて固定します。



10.3.2 アナログ赤外線センサを取り付ける。(フロアセンサ)

1. 図の位置の支柱2本を取り付けます。

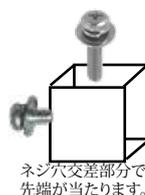


2. アナログ赤外線センサを図のように垂直に取り付けます。床を計測するフロアセンサとして使います。

①ネジM3x12mm+ワッシャーで取り付けます。

②ネジ穴の位置が合わない時は、ヤスリなどを利用してベース板の穴を少し大きくしてください。

③垂直方向のネジを取り付けた後に、基板を固定する水平方向のネジを締めて固定してください。ネジは樹脂スペーサ8mm角のネジ穴中で少し干渉するような長さです。強く締めることでネジが互いに押し合うことでゆるみ止めとなり競技中のロボット振動などで緩まなくなります。



10.3.3. 変調赤外線センサー取り付け

部品名	サイズなど	使用数
ボールセンサ 変調赤外線センサ RDI-203JR		3 個
ネジ	M3 x 12mm	12 個
ナット	M3	24 個
ネジ・ナットはセンサと同じ袋に入っています。		



1. 準備で作製した変調赤外線センサーを3 方向に差し込み、裏側からナットで固定し、取り付けます。

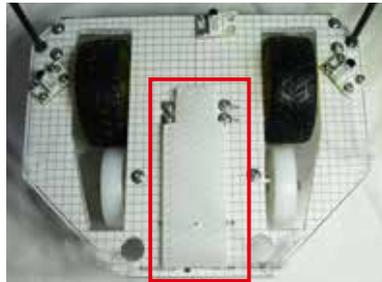
10.3.4. 電池ケース取り付け

1. 図の位置に、取り付ける電池ケースはハニカム板1枚を下敷きにして、ねじ頭などの干渉を防ぎます。



2. ハニカム板を図のように配置してみて、ねじ頭が干渉する部分を切り取り凸型に形を作ります。ハニカム板35x100mmを使います。

ハニカム板はプラスチック板です。ニッパーを使うか、カッターナイフで数度切込みを入れて切断するかの方法で加工します。



3. ベースプレートの図の位置に、単2 電池4 本ケースを取り付けます。

	ナベ小ネジ M3x15mm	2本
	ナット M3	2個



ネジを電池ケースの穴から通し、ベースプレートの裏でナットで固定します。ネジ穴は先端が尖った工具でハニカム板を突き刺して穴を開けます。

10.3.5. 支柱の組み立て要領

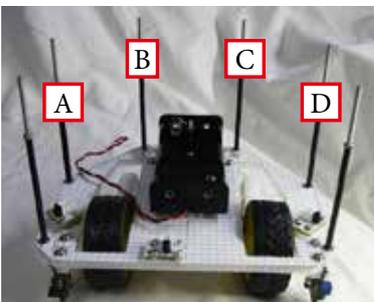
部品名	サイズなど	使用数
ネジ	M3 x 120mm	6本
金環スペーサ	黒色 3 x 25mm	18個
ワッシャー	M3x8	12個
◆ネジ・ナットは、25mm 黒色スペーサーと一緒に入っています。		

1. フロアセンサ取付時に立てた支柱以外の図の位置A ~ D の4 箇所、支柱を作ります。

2. 支柱の作り方は、右図を参照ください。

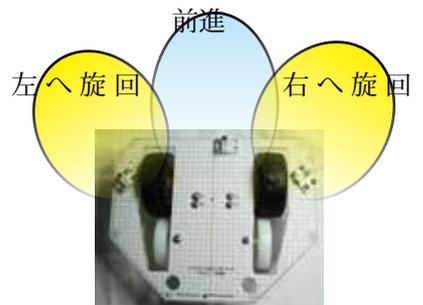
3. 6 本の支柱の仮固定が終わったら、先端に取り付けたナットを締めて固定します。

4. 6 本とも、締め付けが終わった段階で、ロボット筐体基部が完成しますので、ぐらつきなどがいないか確認します。



★センサ指向性のはたらき

\\(◎o◎)/!
センサには検知できる範囲を示す指向性がありますが、左センサ、正面センサ、右センサの角度を利用することにより、各センサのボール検知に範囲を与え、プログラムに従ったボールの位置に対する動きを自律行動しやすくする働きがあります。



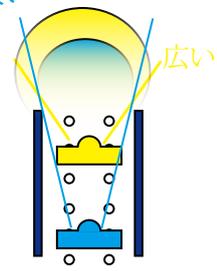
ボール検知範囲とボール捕捉行動の関係図



改造ヒント

中央センサの奥行きで、センサ指向性の調整ができます。

指向性 狭い



支柱組み込み

☞ ナット

☞ 金環スペーサ
黒色25mm径

☞ 金環スペーサ
黒色25mm径

☞ 金環スペーサ
黒色25mm径

☞ ワッシャー

☞ ハニカムプレート

☞ ワッシャー

☞ ナベネジ
M3x120mm



筐体(きょうたい)Cabinet, Case, 外箱:ここでは、ロボットフレームのことを言っています。:機械・電気機器を中に収める部品の位置固定や、ホコリや衝撃からの保護の役割をします。

10-4. ミドルプレートへの組み込み

10.4.1. ドリブラー取り付け

ドリブラー	写真参照	1台
座金組込ネジ	M3x12mmP=4	4本
ワッシャー	M3x8	4個

1. トップ、ミドル、ベース、3段のプレートは微妙な上中下の位置関係になりますので、図の取付位置を確認しながら取り付けます。

2. ネジ穴に座金組込ネジM3x12mmとワッシャーを4箇所にし込み、プレート裏側でナットを嵌め込み固定します。

3. ドリブラーを配置し、裏側をナットで固定します。

4. 最所に取り付けたナットと、ドリブラー取付に使用するナットは、重なり2段になります。プレートにナット1段分のすき間ができますので、モーターコードの配線に利用します。

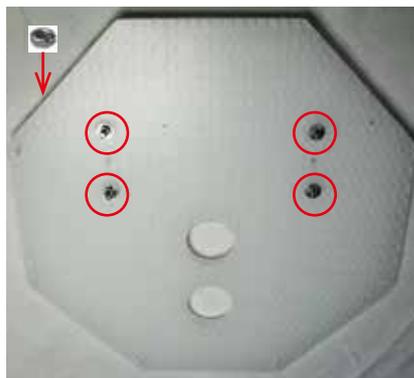
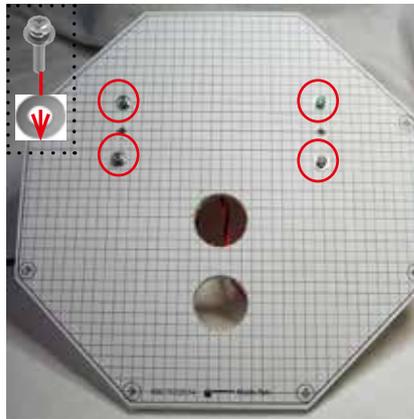
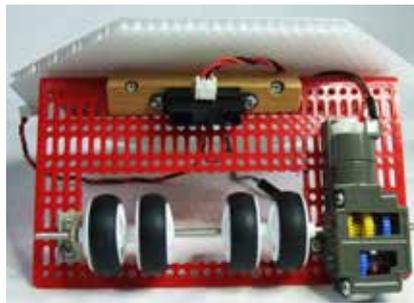
10.4.2. 測距センサー取り付け

ドリブラー	写真参照	1台
ナベ子ネジ	M3x12mm	2本
ナット	M3	2個

1. ドリブラー動作時のボール検知用に測距センサーを使用します。

2. 右準備2.写真のように、ユニバーサルプレートにネジM3x12mmで取り付けます。

3. ドリブル動作を開始するために、駆動タイヤに接触するボールを検知する重要なセンサーです。確実にボール検知できる方法を考えて取り付けます。



準備1. ドリブラーの回転確認



電池2本をつなぎ、ドリブラーがスムーズに回転することを調べます。

●回転しない時は、ギアボックス内の出力軸を固定しているイモネジのゆるみか原因です。ドリブラー部品と同梱している六角レンチを使用してイモネジを締め込み固定します。



●ギアの回転がきしむなど回転音が高いときは、付属のオイルをモーターのピニオンギアに接触している1段目の黄色ギアの歯の部分に注入します。



準備2. 測距センサ(レンジセンサ) 取付



ドリブラーのほぼ中央に、測距センサ(レンジセンサ)を取り付ける。
M3x12ネジ使用

準備. 測距センサを組み立てます。



✓トラスネジ 3x6mm



10-5. トッププレートへの組み込み

10.5.1. コントローラ用樹脂スペーサ取付

部品名	サイズなど	使用数
ネジ	M3 x 8mm	4 個
樹脂スペーサ	M3 x 10mm	4 個
ワッシャー	M3 x 8	8個
樹脂スペーサはM3x6mm ネジでコントローラに付けてありますので、外して使用します。 ◆基板取付ネジM3x6mm は、後で基板取付時に使用しますので、保管ください。		

1. 樹脂スペーサを取り付けます。
2. 樹脂スペーサを取り付けた面が、コントローラを搭載する表面になります。

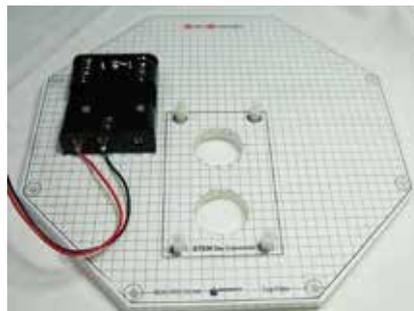
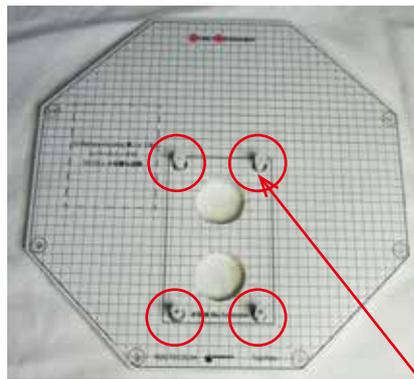
10.5.2. 電池ケース取り付け

1. 図の位置に、電池ケース単3x3本を取り付けます。

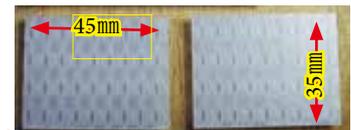
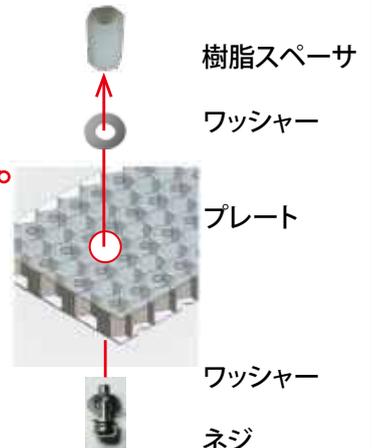
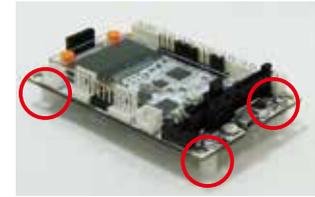
部品名	サイズなど	使用数
皿ネジ	M3 x 12mm	2個
ナット	M3	2 個
ネジ・ナットは、電池ケースに同梱。		

10.5.3 コンパスセンサなど拡張部品取り付け台座

1. 図の位置に、ハニカム構造の樹脂版小片を両面テープで貼付けます。
2. 2枚重ね、あるいは1枚を両面テープなどで貼付けます。

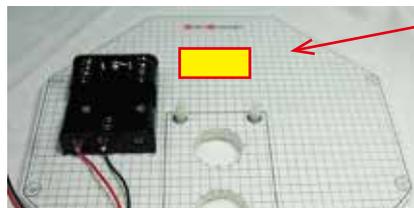


準備1. コントローラボードから、樹脂スペーサを取り外す。



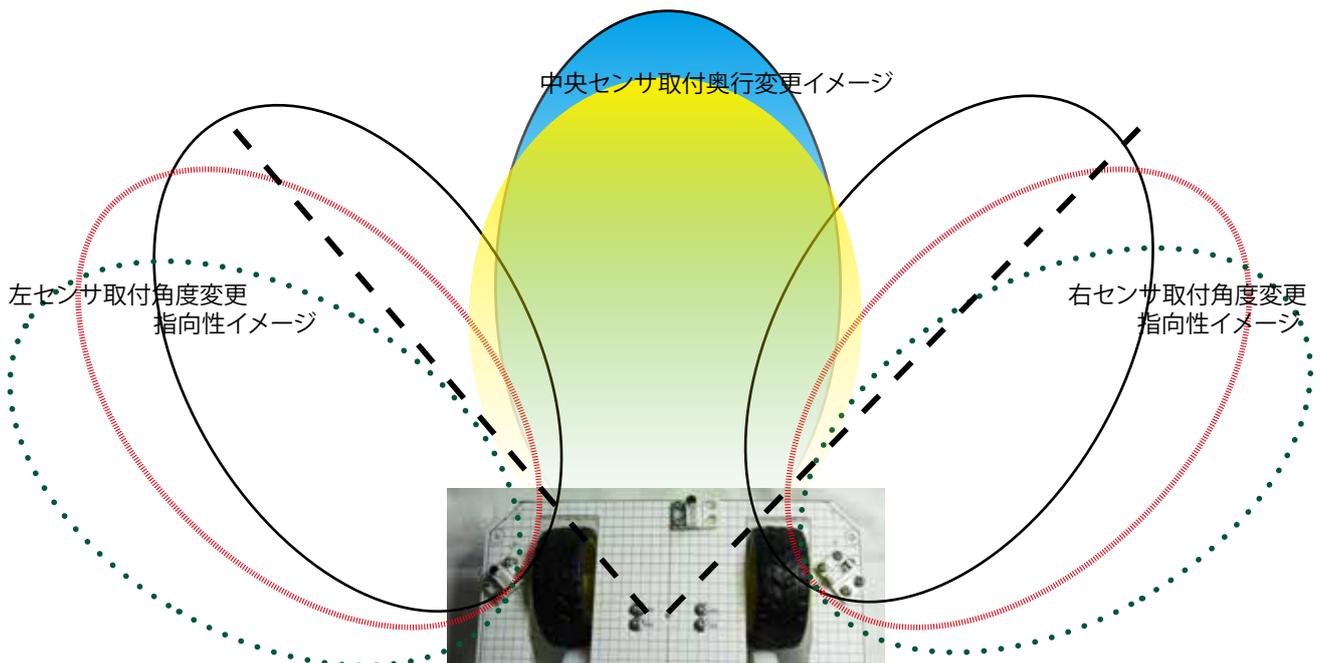
ハニカム小片2枚

貼付けに必要な両面テープを準備ください。
2.5mmの穴を開けると、ネジM3で取り付け可能です。



(拡張時)

ボールの検知には、センサ指向性が重要です。……………ボール捕捉性能は、調整次第!!!



このロボットの動きの特徴

10-6. ロボットを組み立てる

10.6.1 総合組立

1. 準備で作製したベース→ミドル→トッププレートの順に、重ねて取り付けます。
2. 組み立てた6本の支柱に、プレート
の穴を差し込んで取り付けます。



10.6.2. ミドルプレート取り付け

部品名	サイズなど	使用数
樹脂スペーサ	黒色20ミリ六角	6個
ワッシャー	M3x8	6個

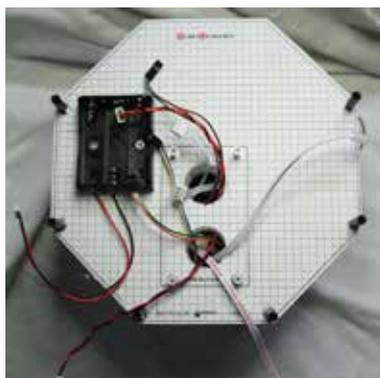
1. 準備で作製したミドルプレートのドリブラーを下面にして、取り付けます。
2. ミドルプレートの上から、ワッシャーをはさみ樹脂スペーサー20mmをねじ込みます。
3. ケーブルを中央の通し穴から、通しておきます。



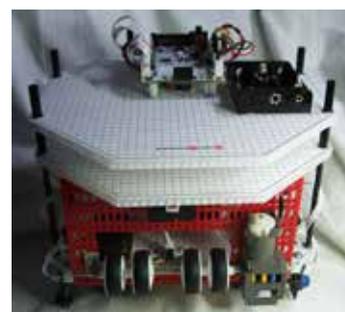
10.6.3. トッププレート取り付け

部品名	サイズなど	使用数
樹脂スペーサ	黒色20mm六角	6個
ワッシャー	M3x8	6個

1. コントローラ設置部分に、接続ケーブルを集めます。
2. トッププレート中央の通し穴から、ケーブルを通しておきます。
3. トッププレートの周囲の穴を、支柱に差し込みます。
4. 支柱先端にトッププレートの上から、ワッシャーをはさみ樹脂スペーサー20mmをねじ込み固定します。



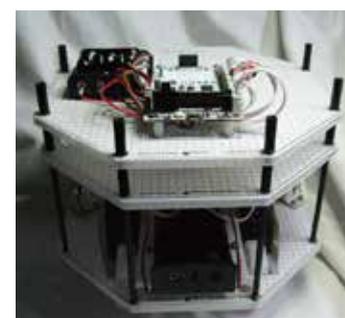
組立完成図



Front fig.



Side fig.



Back fig.

10.6.4. コントローラへの配線

	部品名	コントローラ接続ポート
右側	床センサ 右	A5
	床センサ 左	A4
	レンジセンサ (測距センサ)	A3
	ボールセンサ 右	A2
	ボールセンサ 中	A1
	ボールセンサ 左	A0
左側	電池ケース ドリブラ用	V 2
	ドリブラー	M3
	モータ 左	M1
	モータ 右	M2
	電池ケース マイコン回路用	V 1

V2→
(M3,M4電源)

M4→
M3→

SERVO

M1→
M2→

V1→
(回路電源)

電源
SW

←I²C

←A5

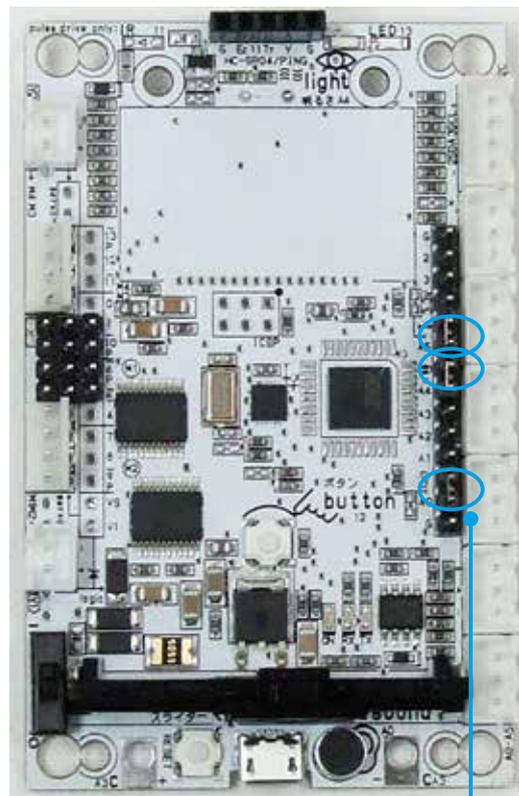
←A4

←A3

←A2

←A1

←A0



1. コントローラセンサ切替ピンを引き抜いて外し、回路を外部接続設定にして、使用します。

A0,A4,A5 のショートピン

このロボットでは、センサ接続のために取り外して使用します。

※外さないでセンサ信号が伝わらず誤動作します。

2. 配線表を確認し、コントローラへ接続します。

3. 接続は、コネクタ部分を持って抜き差しします。



線を引っ張ると、コネクタ接続部で引きちぎれることがあります。

4. 配線を先に行い、コントローラボードの取付は配線完了後に行います。

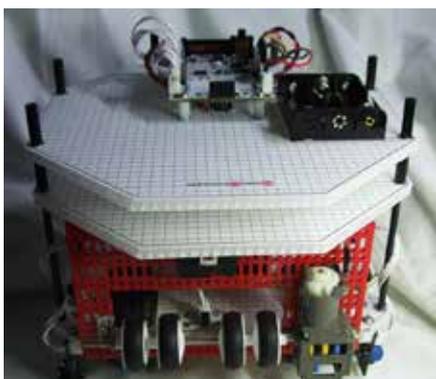
10.6.5. コントローラ取り付け

部品名	サイズなど	使用数
ネジ	M3x6mm	4 個

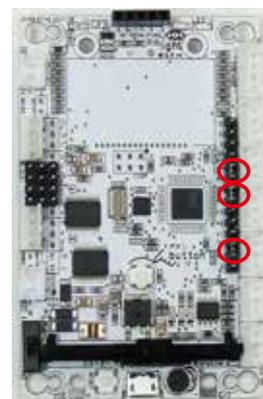
ネジは、コントローラRDC103 に樹脂スペーサを取り付けていた部品です。

1. コントローラボードを、M3x6mm ネジで取り付けます。

2. 電池取付は、配線終了後に行います。



・コントローラに外部センサーを接続する場合は、コントローラセンサ切替ピンを引き抜いて外し、回路を外部接続設定にして、使用します。A0,A4,A5 のショートピン



・A0,A4,A5 で、搭載センサを使用する時に、ショートピンが必要ですので、保管しておいてください。



10-7. ロボットの動作確認

10.7.1. 開発環境起動の事前準備…COMポート番号の確認

1. PCとマイコンボードを、マイクロUSBケーブルで接続します。
2. コントローラに電源が入ったことを確認します。
3. PCがマイコンボードを感知し、PC側の「COMポート」が設定されますので、次の手順に従い、COM番号を調べてください。
 - ・Windows：マイコンコンピュータ > コントロールパネル > ハードウェアとサウンド > デバイスマネージャーの順で開いていき、[ポート (COMとLPT)COM] を見つけます。
4. (COMとLPT)COM] に認識出現している USBシリアルデバイス (COM番号)を確認し、COM記号の後ろにある数字(ポート番号)をメモします。
5. プログラム転送処理時にCOM番号が必要となります。
 - ※プログラムが書き込めない場合は、必ずCOM番号を確認してください。

10.7.2. プログラム開発環境の起動

1. デスクトップに作成したショートカット arduino.exe をダブルクリックして arduino を起動します。
2. Arduino-IDE [ツール] で [マイコンボード]、[シリアルポート]を確認します。
3. [マイコンボード]
 - Arduino-IDE [ツール] ▶ [マイコンボード] をクリックし、出現するマイコンボードリストで、[STEM Du/RoboDesigner+ RD C-102 w/ ATmega32U4 3.3V 8MHz] を選択・クリック指定を行います。リスト左端に●印が付きます。
4. [シリアルポート]; Arduino-IDE の [ツール] ▶ [シリアルポート] をクリックし、出現するサブウィンドウで、先ほど デバイスマネージャーで調べた COM 番号の通信ポートをクリック指定し、☑マークがついたことを確認します。

指定を間違えると、通信ができなくなります。

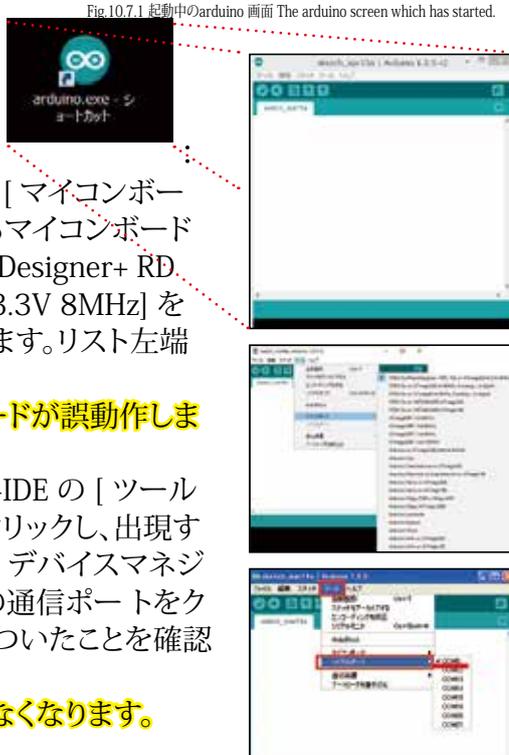


Fig.10.7.1 起動中のarduino画面 The arduino screen which has started.

10-7. Operations check of a robot

- [1]. Power supply switch of a controller board is turned on.
- [2]. Preliminary preparations of a development environment start--- The communication port number is confirmed.
 - 1.PC and a microcomputer board are connected by Micro USB cable.
 - 2.PC senses a microcomputer board, and "communication port" on the PC side is established, so please check the COM number with the next procedure.
 - * Windows : My computer Start > Control Panel > Hardware and sound > It's being held by the order of the device manager and [port COM(COM and LPT)] is found.
 3. You check STEM Du RDC-102 from which recognition emerges in [(COM LPT) COM] (portnumber), and take notes of the number which is behind the COM symbol (portnumber).
 4. The COM number is needed at the time of program upload.

※ When a program can't be written in, please be sure to confirm the COM number.

[3]. Start of a program development environment.

1. The short cut made in a desktop arduino.exe is double-clicked and arduino is started.
2. [Microcomputer board] and [serial port] are confirmed by Arduino-IDE [tool]. [Microcomputer board]: of Arduino- IDE [Tool] > [STEM Du/RoboDesigner+ RD C-102 w/ ATmega32U4 3.3V 8MHz] is chosen by the microcomputer board list which clicks [microcomputer board] and appears, and click designation is performed. A ● mark sticks to the list left end.

When you make a mistake in designation, a microcomputer board malfunctions.

COM checked by a device manager a short while ago by a subwindow A communication port of the number is designated and it's confirmed that a ☑ mark stuck.

When you make a mistake in designation, you can't communicate any more.

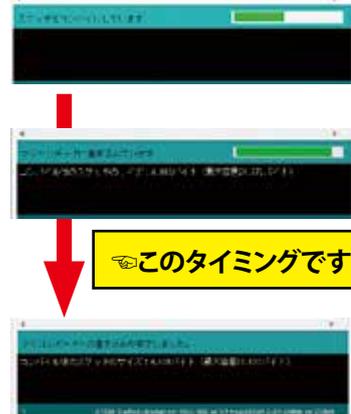


エラーメッセージ Couldn't find a Leonardo on the selected port. Check that you have the correct port selected. If it is correct, try pressing the board's reset button after initiating the upload.

出現時には、通信エラーが発生し、マイコンボードへの書き込みが失敗しています。

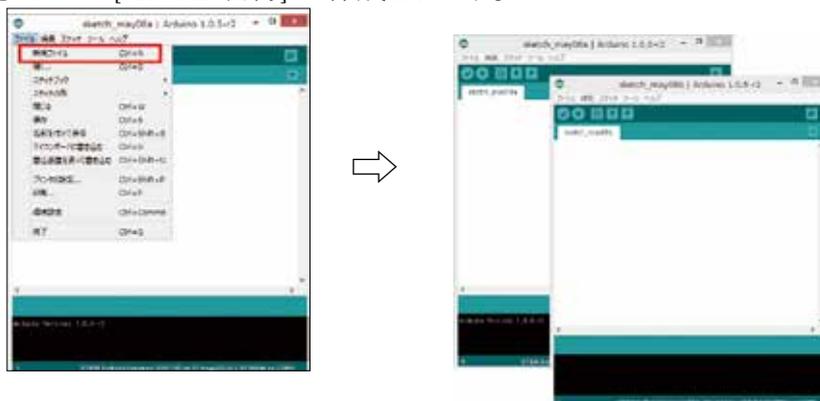
- 1)USB ケーブル接続確認
- 2)Arduino > [ツール] > 「マイコンボード」でRDC-102 に●マーク、「シリアルポート」接続COM 番号に☑マークを確認ください。

※アップロードがうまくいかない場合は、コンパイルの後、マイコンボードに書き込み中に、RDC-103 コントローラのRESET スイッチを「ダブルクリック」し**強制アップロード**してください。



このタイミングです。

5. Arduino-IDE の[ファイル] > [新規ファイル] をクリックします。
新規ファイル[Sketch. 日付] が作成されます。



5. The [new file] in the [file] of Arduino-IDE is clicked. A new file [Sketch. Date] is made.

6. 新規ファイルのarduino-IDE で、[ツール] にある[ArduBlock] をクリックします。



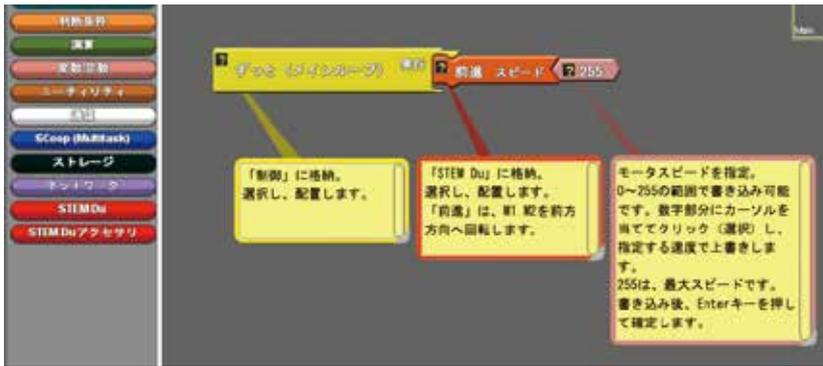
6. [ArduBlock] in [tool] is clicked in arduino of a new file-IDE.

7. ArduBlock がスタートします。



7. ArduBlock starts.

10.7.3. 動作テストプログラムを作成します。



- 上記の図を参考にしながら、前進するプログラムを作成します。
 - 例は、「前進」を使っていますが、他の動きでテストを行なうこともできます。
 - モータスピードは255 が最大値です。かなり速いスピードで動きますので、最初は、少し遅めが良いと思います。半分くらいのスピードを指定してみます。
- 「Arduino へアップロード」します。



3. Arduino-IDE では、コンパイルを行い実行ファイル化され、コントローラボードに書き込みます。

4. Arduino-IDE の下部メッセージ画面に、動作状態の表示がされます。「マイコンボードへ書き込みが完了しました。」メッセージが表示されると、書き込み完了です。

****RDC-103 コントローラは、電源が入ると、すぐにプログラムが実行されます。言い換えれば、電源スイッチをON にするとロボットの動作が開始します。ロボットが動作しても安全な位置に置いて、動作開始させます。**

***** 机の上に置いたまま、電源を入れるとロボットが動き出し、床に落下して衝撃で壊れるなど事故が起こりますので、台座の上に置きタイヤを宙に浮かすなどロボットの設置場所に十分配慮して、ロボットを動かします。**

A movement test program is made.

1. You make the program to which you just move while consulting a figure abovementioned.

* An example uses “advance”, but it’s possible to test by other movements.

* 255 is the greatest for the motor speed. It moves by the quite fast speed, so the beginning, it’s rather a little late, I think it’s good. You’ll designate the speed which is about half.

2. “It’s uploaded to Arduino.”, it’s done.

3. Arduino-IDE compile, an executable file-sized and write in a controller board in Arduino-IDE.

4. Operating state is shown to the lower part message screen of Arduino-IDE. “I have finished writing notes in a microcomputer board.”

When a message is indicated, it’s writing in completion.

****When RDC-103 controller is turned on, a program will be executed immediately. When power supply switch is turned on in other words, movement of a robot begins it. Even if I move, a robot puts it in the safe location and makes them begin to move.**

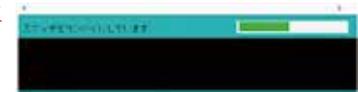
*****When You turn on the power while placing it on the desk, a robot begins to move, and such as falling in the floor and breaking by an impact, an accident happens, so such as placing it on the pedestal, it’s considered sufficiently in an installation site of a robot and a robot is moved.**

エラーメッセージ Couldn't find a Leonardo on the selected port. Check that you have the correct port selected. If it is correct, try pressing the board's reset button after initiating the upload.

出現時には、通信エラーが発生し、マイコンボードへの書き込みが失敗しています。

- 1)USB ケーブル接続確認
- 2)Arduino > [ツール] > 「マイコンボード」でRDC-102 に●マーク、「シリアルポート」接続COM 番号に☑マークを確認ください。

※アップロードがうまくいかない場合は、コンパイルの後、マイコンボードに書き込み中に、RDC-103 コントローラのRESET スイッチを「ダブルクリック」し**強制アップロード**してください。



このタイミングです。



10.7.4. はじめてロボットを動作スタート、各種動作点検

(1). 基板に転送したプログラムの実行

1. ロボット動作事前準備確認

- a. ロボットの接続状態を確認するために実機を動かして点検を行います。
- b. 下記の設定通りに入出力機器の接続がなされているかを確認します。

部 品	コントローラボード
タイヤ付ギアボックス 左側Left Gear Box	M1
タイヤ付ギアボックス 右側Right Gear Box	M2
電池ケース(単2 × 4)Battery housing	V1

- c. [回路電源スイッチ(SW)] がOFF になっていることを確認します。
- d. 電池が入っている事を確認します。

2. プログラム実行: [電源スイッチ(SW)] をON にします。

プログラムが実行されます。

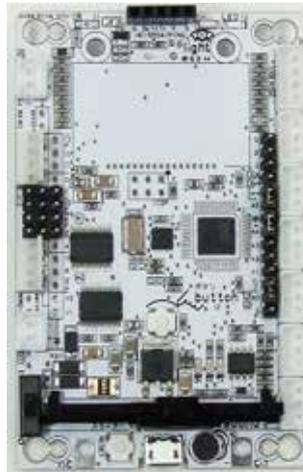
3. プログラム停止: [回路電源スイッチ(SW)] をOFF にします。

左モータLeftmotor M1→

右モータRightmotor M2→

電池ケースBatteryhousing V1→

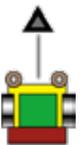
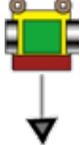
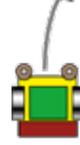
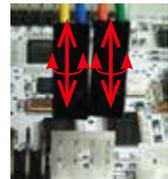
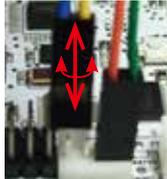
電源スイッチPowersupply switch→



RESET スイッチ ↑ ↑ USB

(2). 初めてのロボット動作点検

(1). プログラムを実行し、ロボットの動作をスタートして、前進するかどうかを確認します。配線の間違ひを含めて、最初の重要な確認ですから、注意して観察しましょう。

前進する	バックする	左回転	右回転	少しカーブする
				
正しく動いています。	M1,M2 配線の極性(プラス・マイナス)接続間違い	M1 モータ配線の極性(プラス・マイナス)接続間違い	M2 モータ配線の極性(プラス・マイナス)接続間違い	正しく動いています。
そのまま使います。 It's connected right. I'll use it just as it is.	M1,M2 ともにコネクタの向きを反対にして(±接続を入れ替えて)、接続し直してください。 Replace a ± connection of the left and right side motor and connect again.	左側モーターのコネクタの向きを反対にして(±接続を入れ替えて)、接続し直してください。 Replace a ± connection of the left side motor and connect again.	右側モーターのコネクタの向きを反対にして(±接続を入れ替えて)、接続し直してください。 Replace a ± connection of the right side motor and connect again.	左右のモーター個体差の影響で、許容範囲内です。 It's moving right. It's influence of the motor individual difference in the left and right and is in the latitude.*1
				プレートにタイヤが接触していないかを確認してください。擦れているとブレーキをかけた状態になり、正しく回転しなくなります。*1.

*1. 左右モーターの個体差が影響し、完全にどこまでもまっすぐ進むことは、大変難しいことです。

* 長い距離を走らせると左右のどちらかに少しづつ曲がっていきます。

** 違う構造では、たとえば1個のモーターの両側に車輪を取り付け走らせるなどの実験を行うとまっすぐ進みますが、プログラムにより自在に方向転換ができなくなります。

*** 解決策としてモーターの個体差をなくす方法も考えられますが、数万個の生産の中から個体差が少ないモーターを探し出すことになり、大変高価なコストになります。

**** 今回取り組んでいる自律型ロボットはセンサ情報を得てプログラムに基づき、常に方向を変えて動きますので、長い距離での直線性が大きく問題となりませんので、ご安心ください。
エンコーダモータを利用し、左右の回転を制御することにより、直進性を得ることもできます。

Fig.10.7.1 動作点検 Movement check

A movement start and all kinds' movement check a first robot.

(1). Execution of the program forwarded to a substrate

1. Robotmovement preliminary confirmation of preparations

a. A production is moved and it's checked to confirm the state of the connection of the robot.

b. It's confirmed whether an input/output device is connected to the following setting street.

c. It's confirmed that [circuit power supply switch (SW)] becomes off.

d. It's confirmed that a battery housing contains a battery.

2. A program is executed: [Power supply switch (SW)] is turned on.

A program is executed.

3. Program stop: [Circuit power supply switch (SW)] is turned off.

(2). First robot movement check

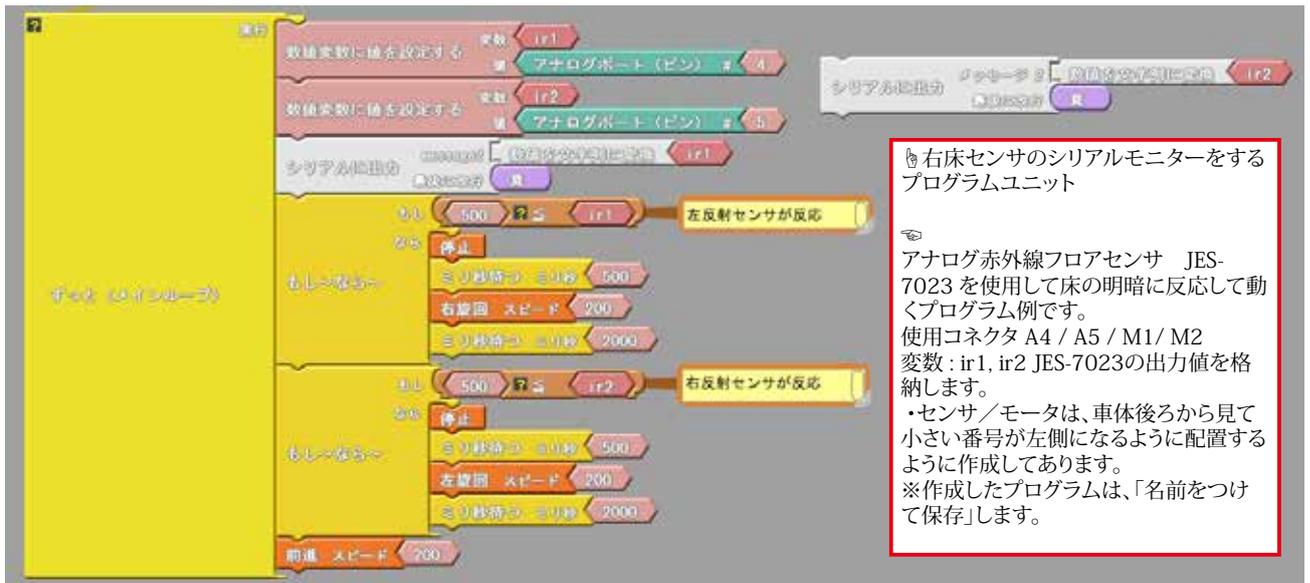
1. A program is executed and movement of a robot is started, and it's confirmed whether you move ahead.

* Because it's the first important confirmation you put into effect including a mistake of your wiring, please be careful and observe.

10.7.5. 白線で停止/ 旋回するプログラムを作成

(1) プログラムを作成し、コントローラへアップロードしてください。

Robot detects a white line, and makes the program which suspends a movement.



動作手順

- 床の色が暗ければ前進します。
 - ir1, ir2 の値がしきい値以上になったら停止/旋回します。
- ソースコードは、PC/MyDocuments/Arduino/へ配置したサンプルフォルダ[ArduBlock Examples]に、ファイル名【24_01_base_floor_sample.abp】で格納されています。Arduino/ArduBlock で、PC/MyDocuments/Arduino/ArduBlock Examples の中に配置したサンプルを開くと確認できます。

(2). 床センサのデータを調べます。

- データーを調べるときは、USB ケーブルを接続します。
- Arduino-IDE[ツール] で[マイコンボード]、[シリアルポート] を確認します。
 - [マイコンボード]: Arduino-IDE の [ツール] > [マイコンボード] をクリックし、出現するマイコンボードリストで、[STEM Du/RoboDesigner+ RD C-102 w/ATmega32U4 3.3V 8MHz] を選択・クリック指定を行います。リスト左端に●印が付きます。
 - [シリアルポート]; Arduino-IDE の [ツール] > [シリアルポート] をクリックし、出現するサブウィンドウで、デバイスマネージャーで調べた COM 番号の通信ポートをクリック指定し、☑ マークがついたことを確認します。



- コントローラのプログラムが実行されている状態の時に、ArduBlock の[シリアルモニター] を使ってセンサの値を調べることができます。

[シリアルに出力] ブロックを使ったプログラム行を図のように作成挿入し、シリアルモニターできるようにします。

ir1: 左アナログ赤外線センサー(床を見るセンサー)
 ir2: 右アナログ赤外線センサー(床を見るセンサー)
 プログラムを書き換えてコントローラボードへアップロードして、白線を感知するセンサーの左右をそれぞれに測ります。

(1). Please make a program by making reference to a figure.

The program example reacts to the light and shade of the floor color using JES-7023 of an infrared analog sensor and moves.

Use connector : A4 / A5 / M1 / M2
 Variable : ir1, ir2 : An output value of JES-7023 is stocked.

* A sensor/a motor is made as it is seen from the body rear, and it may be arranged as the small number will be the left side.

※ Your made program names and preserves it. After making completion, it "is uploaded to Arduino", it's done and it's written in controller board.

* When the color of the floor is dark, Forward.

* If the signal level of ir1, ir2 will be beyond a threshold value, it stops/it circles.

* Program source cord of [base_floor_sample] is stocked in the sample folder arranged to PC/MyDocuments/Arduino/ArduBlock_Examples by the file name [13_01_base_floor_sample.abp].

2. Data of a floor sensor is hecked.

- When checking a dater, a USB cable is connected.
- Please checked Serial port , Microcomputer board list.
- It's possible to check the value of ArduBlock at the state that a program of a controller is executed. (A sample program is being made in order to monitor a serial.)
- When [serial monitor] of ArduBlock is clicked, a serial monitor screen stands up, and sensor value is indicated in real time.
- They're on the green carpet and a white line while checking the sensor value by a serial monitor and a data collection is performed respectively. (For a USB cable, condition of a connection)
- Intermediate value of measured data is used as "threshold value".
- When graphing makes spreadsheet software copy from a serial monitor, and deals with a figure, it becomes easy to understand.



4. ArduBlock の[シリアルモニター]をクリックするとシリアルモニター画面が立ち上がり、リアルタイムでセンサ値が表示されます。

5. ロボットを手で移動させグリーンカーペットの上、白線の上にセンサーの位置をあててシリアルモニターでセンサ値を確認しながら、それぞれデータ収集を行います。(USBケーブルは接続のまま)

6. 計測したデータの間中値を、「しきい値」として使います。

7. 測定データをシリアルモニタからコピー (Ctrl+C) し、表計算ソフトなどにペースト (Ctrl+V) して数値をグラフ化処理するとセンサ出力傾向値が分かり易くなり、分岐条件(しきい値)考察が容易になります。
※ 1. USB ケーブルを抜いて、ロギングを停止してから、データをカーソルで範囲指定しコピーしてください。

(3). 各種パラメータ調整

1. 床面の色の違いで、反射率の影響を受け、センサに届く赤外線の色量が増減します。

- ・シリアルモニターを使って、計測をしながら、ロボットを少しずつ移動していき、緑色の床と、白色の床で、色の違いにより、どのようにセンサ出力が増減するかを調べます。
- ・データは、数回調べて、緑色床、白色床それぞれに平均値を調べます。

2. センサ出力(緑色床からの反射データと、白色床からの反射データ)の違いを調べて、その中間値が「しきい値」となります。

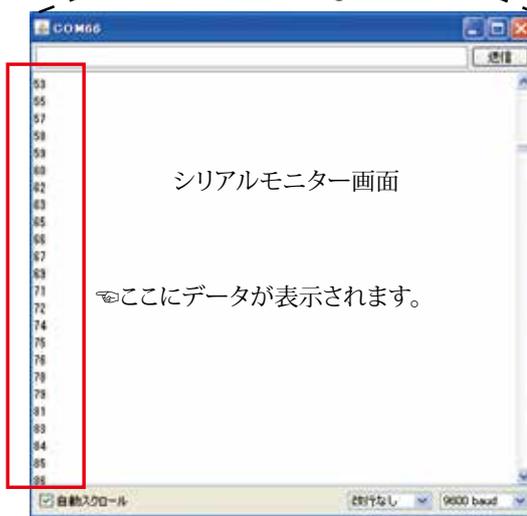
(4). 「表計算ソフトの利用」をお勧めします。 次頁詳細説明

・右は、例として、グリーンカーペットに白線を施した床で、グリーンカーペットと白線の上を交互に計測したシリアルモニターのデータを、グラフ化した図です。どれほどの信号の大きさか一目で理解でき、「しきい値」の検討などに役に立てることができます。大きい値と小さい値の中間値位を「しきい値」とします。図の実測例の場合、900 くらいが「しきい値」となります。

データ数は、1秒間で4000個ほどの大きな量です。調べる時間は1秒ごとにグリーン上、白線と交互にロボットを移動して数回調べます。

The number of acquisition data is the big amount of about 4000 a second.

It's green and it's on the white line. It's checked every 1 second. You move a robot alternately and check it several times.



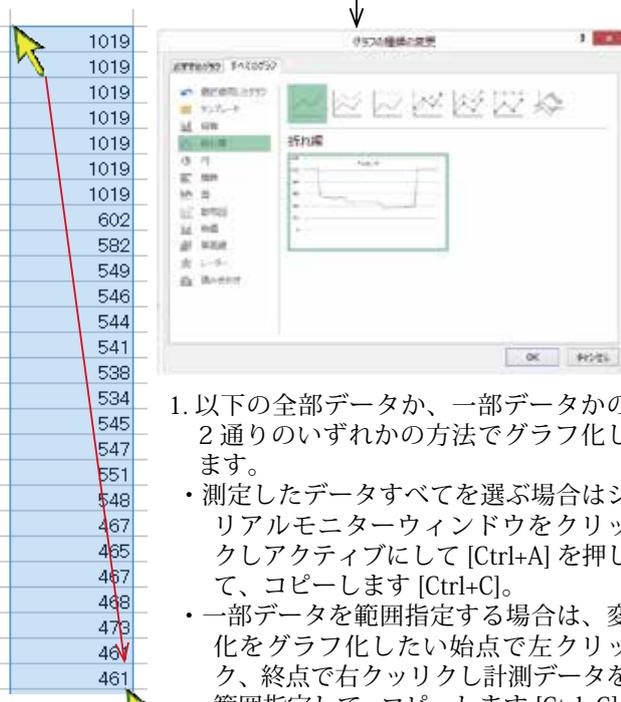
シリアルモニターを長い時間継続すると、取得データがオーバーフローし、PCがフリーズすることがあります。このような場合、データ取得を中止し、コントローラのリセット、Arduinoの再立ち上げを行ってください。

お使いのP/Cによっては、P/Cの再起動が必要な場合もあります。When data overflowed, a PC is reset, and there is a case which needs a restart.



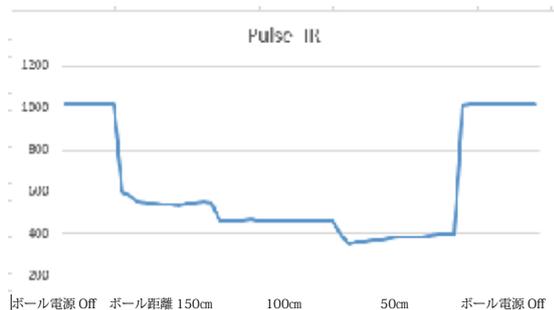
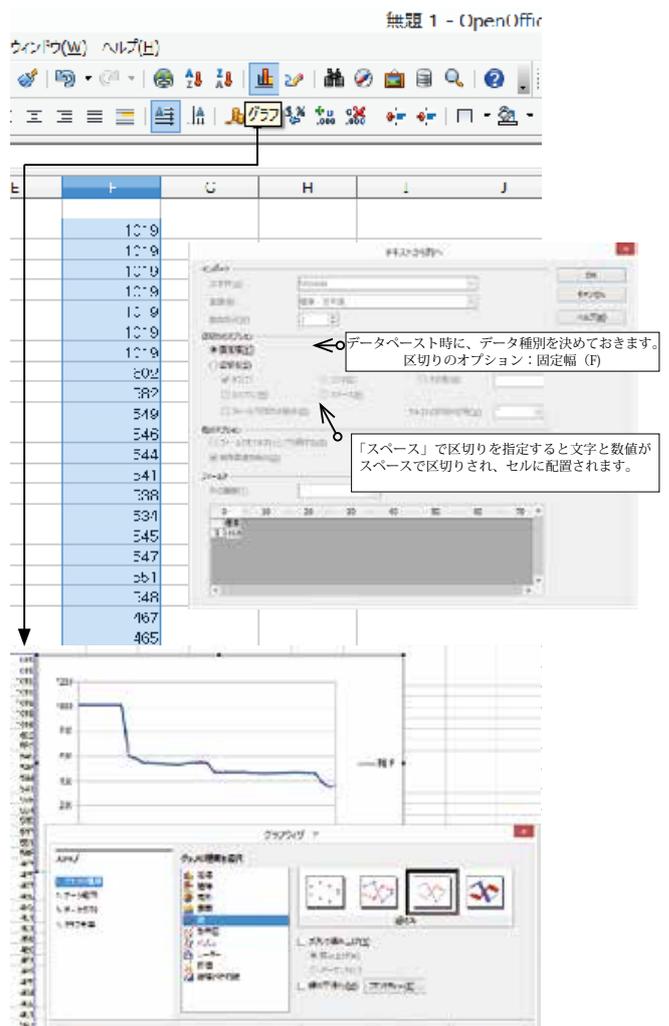
(5). 表計算ソフトの使い方

1. シリアルモニタのデータは、1秒で4000個超のカウント数になります。プログラムの分岐条件に使用するしきい値は取得データを「表計算ソフト」などを利用してグラフ化し、分岐点を考察します。
 - 代表的な表計算ソフトとして **EXCEL** (有料ソフトウェア) があります。その他無料でインターネットから入手できるソフトウェアの例として **OpenOffice** もあります。



1. 以下の全部データか、一部データかの2通りのいずれかの方法でグラフ化します。
 - 測定したデータすべてを選ぶ場合はシリアルモニターウィンドウをクリックしアクティブにして [Ctrl+A] を押し、コピーします [Ctrl+C]。
 - 一部データを範囲指定する場合は、変化をグラフ化したい始点で左クリック、終点で右クリックし計測データを範囲指定して、コピーします [Ctrl+C]。
2. 貼り付けたい表計算ソフトの位置 (先頭のセル) を指定して、ペーストします。 [Ctrl+V]
3. グラフ化したいデータを範囲指定して、グラフ化処理をします。
4. グラフの形などは自分が分かりやすい形を選びます。

※シリアルモニタの出力を表計算ソフトで処理する場合は、変数の前のテキストをカンマやスペースに置き換えてください。
OpenOffice の場合は、区切りのオプション「固定幅 (F)」を選択指定してデータをペーストしてください。ペースト時に出現するダイアログのメニューで、「スペースで区切り」を指定すると文字と数値が区切られてセルに配置されます。



- 図は、ロボカップジュニア公式競技ボールを変調赤外線センサで計測したデータです。
- グラフで見ると、距離ごとに計測数値が違うことも理解でき、プログラムの調整に役立ちそうです。
 - ボールから離れているとき、中間距離時、近い距離など、ボールからの距離ごとにデータが違うことが分かり、プログラムで工夫をしてロボットの行動を変えることが可能です。

(6). 調べたデータで、プログラムの「しきい値」を書き換えます。

○サンプルプログラムでは、ir1, ir2 の「しきい値」を500 と仮に決めていますので、今回、調べた実測値に基づき「しきい値」を書き換えます。

ここでは、計測結果に基づき プログラムの「しきい値」を250に書き換えてみます。

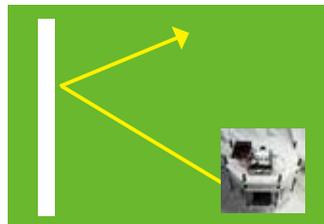
計測結果が次のような場合
 緑床上:120~170
 白線上:300~400



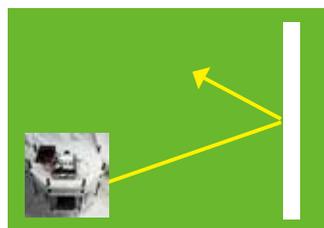
ir1, ir2 の値がしきい値以上になったら停止／旋回します。

- もし $500 \Rightarrow 250(\text{しきい値}) \leq ir1$ なら
 - 停止 500 ミリ秒
 - 右旋回 モータスピード200 2000 ミリ秒
- もし $500 \Rightarrow 250(\text{しきい値}) \leq ir2$ なら
 - 停止 500 ミリ秒
 - 左旋回 モータスピード200 2000 ミリ秒
- でなければ 前進 モータスピード200

※「しきい値」に、計測仮設定した250 以外に、180, 200, 270, 300というように前後の数値にも書き換えて実験してみます。図のような円滑な動きになる最適な「しきい値」を実動テストにより見つけます。



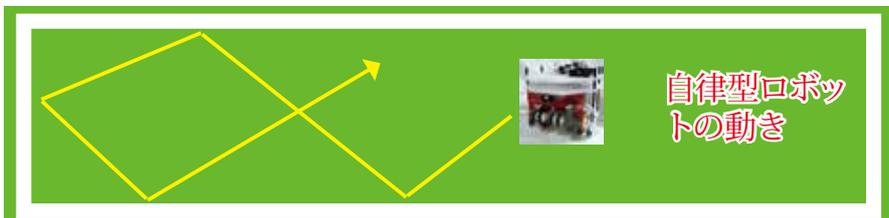
1. 左図の場合で
 左側へ斜めに白線に到達すると、左側のir1 が先に白線に反応します。
 もし
 左側の反射赤外線センサが「しきい値」以上の数値になったら0.5 秒停止して右旋回を2 秒間 200 のスピードで行います。



2. 次に、右側へ斜めに白線に到達すると、右側のir2 が白線に反応します。
 もし
 右側の反射赤外線センサが「しきい値」以上の数値になったら0.5 秒停止して左旋回を2 秒間 200 のスピードで行います。
 とサンプルプログラムがなっていますので、

3. 緑色床と白線の違いがわかる「分かれ目(しきい値)」を調べて、サンプルのir1,ir2 のしきい値を変更することにより、現在実験している環境(周囲の明るさなどの影響)下で、白線を回避するロボットが作成できるようになります。

4. プログラムとロボットの関係が、分かってくると、大変楽しくなります。いろいろなプログラム作りをして、思ったような動きにできるロボットを作りましょう。



赤外線フロアセンサの原理



センサ裏面の発光LEDから赤外線を発光し、床に照射します。

床からの反射波を受光LEDで計測するセンサです。

白が大きく反射、黒は反射が少ないなど色の違いにより光の反射量は変化します。

反射波の強さの違いを計測することで、プログラムにより行動を変化させます。

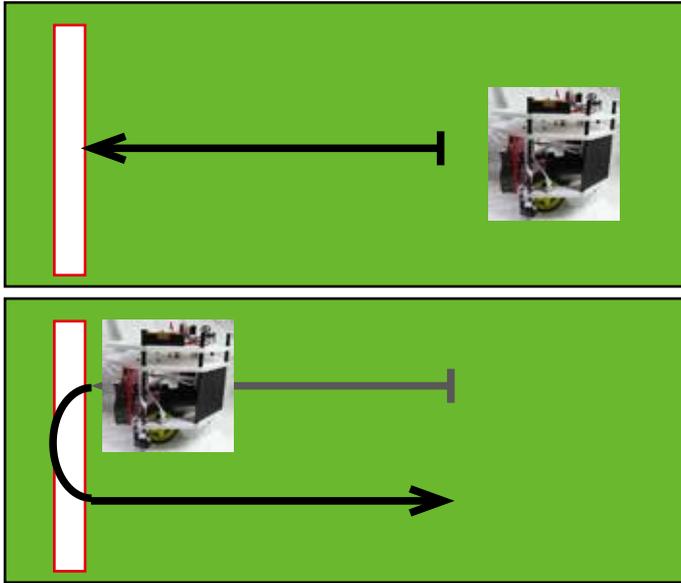
フロアセンサ調整

センサの半固定ボリュームで出力調整が可能です。



(7). プログラムを変更し、動きを変えてみます。

1. では、正面から進んでいき、白線に差し掛かったら、停止して、ほぼ、180度回転し、他の方向へ進んでいく、ロボットを作成してください。



サンプルプログラムを変更します。

2. サンプルプログラムの
もし $200(\text{しきい値}) \leq \text{ir1}$ なら
停止 500 ミリ秒
右旋回 モータスピード200 2000 ミリ秒
でなければもし $200(\text{しきい値}) \leq \text{ir2}$ なら
停止 500 ミリ秒
左旋回 モータスピード200 2000 ミリ秒
の旋回する時間を変更すると、180 度回転するようになります
2000 ミリ秒だったら回転角度はいくらだったでしょう？
変更時間は何秒に設定したら、180 度ピタッと回転してくれるようになりましたか？
3. 上手く動作するようになるまで、「実走」→「観察」→「プログラム変更」→「ロボットへ書き込み」→「実走」→「観察」を繰り返します。

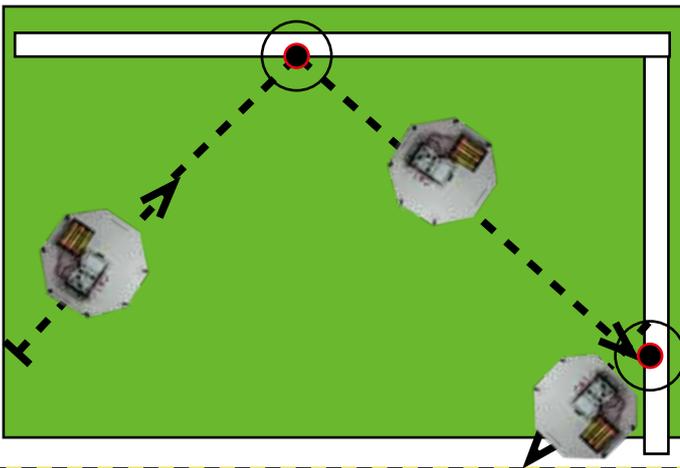
4. 電池残量、グリーンカーペットの厚さ等各種の要因が影響しますので、実物の動きが180 度回転するようになったときの設定時間が正解です。

5. 実動テストをして、得られた、しきい値、モータ回転方向、モータスピード、モータ回転時間等の各要素のデータがロボットが動いている内容となり、パラメーターと呼びます。記録をして次の実験を行います。

(8). プログラムを試みる。

1. プログラムとロボットの関係が、分かってくると、大変楽しくなります。いろいろなプログラム作りをして、思ったような動きができるロボットを作ります。

2. グリーン床に白線を引いて(白色クラフトテープ、電気用絶縁テープ等)、枠内を超えないで方向転換する自律型ロボットをプログラミングします。



- 白線に反応させる…→「しきい値」を指定します。
 - 白線反応後に回転する方向…→「左旋回」「右旋回」を指定します。
 - 回転する角度…→「回転スピード」と「ミリ秒(時間)」の組み合わせで指定します。
が、調整ポイントです。
3. 各パラメータを書き直して動きを調整します。
 4. 成功したプログラムは、別名保存します。

プログラムエラー

• プログラムタイピングミスの場合など、Arduino の該当行が黄色ハイライト表示で警告されます。

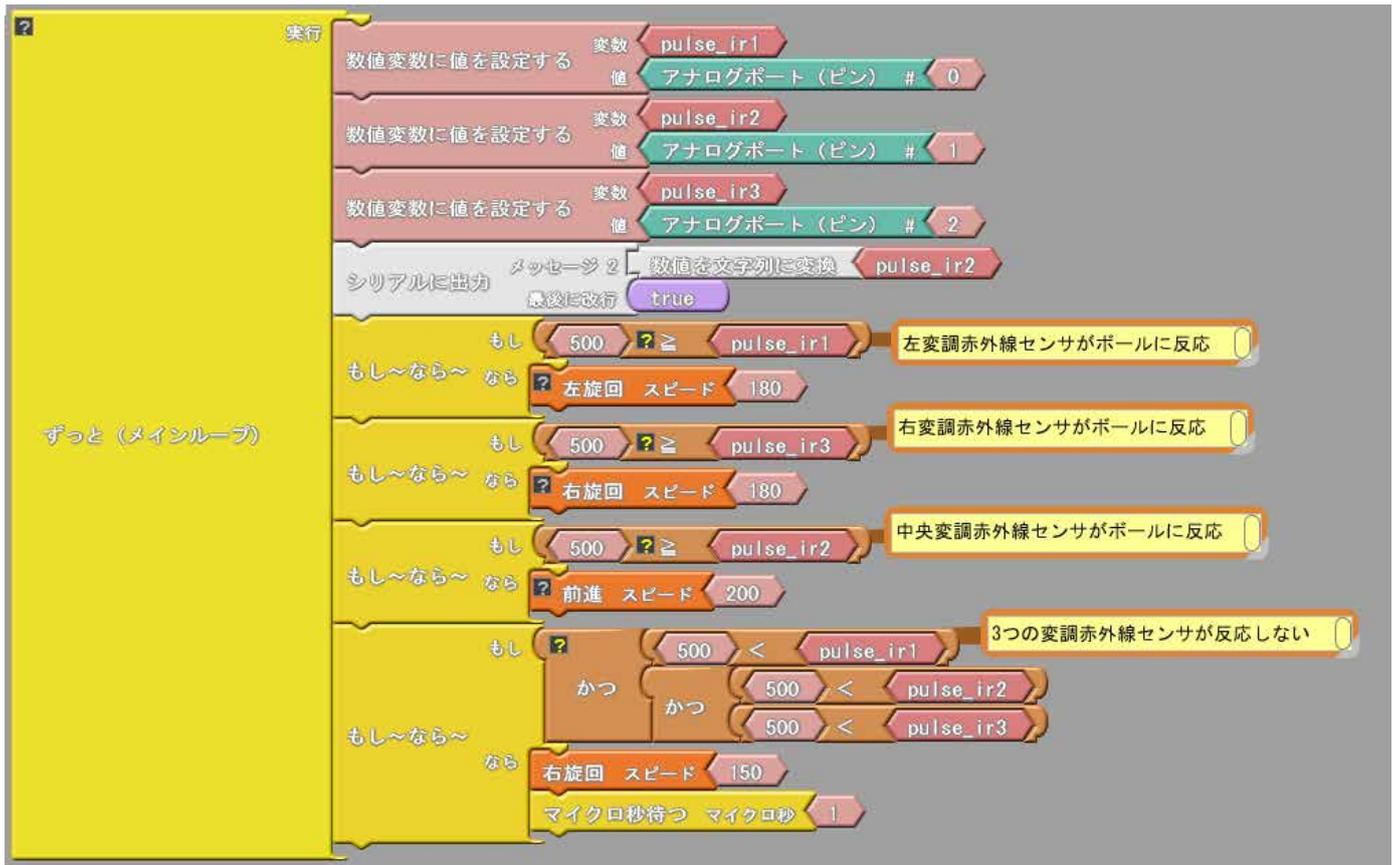
• プログラムで使用できる文字は「半角英文字」と「半角数字」のみです。全角文字は、プログラム文ではエラーとなり、該当行が黄色マークで警告されます。

◆ エラーメッセージを確認して、対策を施して、問題を解決した後で、再度、マイコンへの書き込みを行います。

エラーメッセージ
エラー内容が表示されます。



10.7.6. パルス変調赤外線ボールセンサーを探索するプログラムを作成します。



変調赤外線センサー RDI-203JR を使用して赤外線ボールを探すプログラム例です。

上図を参考にプログラムを作成し、コントローラにアップロードしてください。

使用コネクタ A0 / A1 / A2 / Motor1 / Motor2

変数 pulse_ir1, pulse_ir2, pulse_ir3 RDI-203JR の出力値を格納します。

・センサ／モータは、車体後ろから見て小さい番号が左側になるように配置するように作成してあります。

※作成したプログラムは、「名前をつけて保存」します。

動作手順

- ・pulse_ir1 がしきい値以下になったら左旋回します。
- ・pulse_ir2 がしきい値以下になったら前進します。
- ・pulse_ir3 がしきい値以下になったら右旋回します。
- ・pulse_ir1, pulse_ir2, pulse_ir3 とともに、しきい値以上の場合は右旋回でボールを探します。

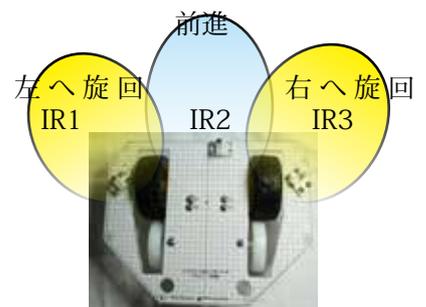
※「しきい値」は、ツール>シリアルモニタを使って pulse_ir1,2,3 の値を確認して設定します。シリアルモニタのデータをコピー (Ctrl+C) し、表計算ソフトなどにペースト (Ctrl+V) して数値を処理すると分かり易くなります。サンプルは しきい値 を 500 と仮に記していますが、ロボットを動作させる環境によって変化しますので、皆さんの現状で調べて しきい値 を決定してプログラムしてください。

※ RDI-203JR はボールの赤外線が強くなるほど出力が低くなりますので注意してください。

ソースコードは、PC/MyDocuments/Arduino/へ配置したサンプルフォルダ [ArduBlock Examples] に、ファイル名 [24_02_base_ball_X24_sample.abp] で格納されています。Arduino/ArduBlock で、PC/MyDocuments/Arduino/ArduBlockExamples の中に配置したサンプルを開くと確認できます。

★センサ指向性のはたらき

∪(●o●)∩!
センサには検知できる範囲を示す指向性があり、左センサ、正面センサ、右センサの角度を利用することにより、各センサのボール検知に範囲を与え、プログラムに従ったボールの位置に対する動きを自律行動しやすくする働きがあります。



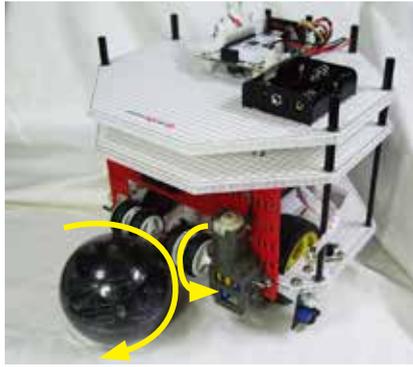
ボール検知範囲とボール捕捉行動の関係図

10-8. ドリブルロボット ドリブル駆動力調整

1). ドリブル回転数調整

1. ドリブラで、手前側へ回転させる力をボールに加えることにより、ボールはロボット側へ転がり、ロボットから離れません。逆回転させるとボールは弾かれて、ロボットから離れます。

2. ドリブラの回転数を高速にしたり、低速にしたりすることで、ボールがロボットに引っつく強さが変わります。ドリブルに使っているモータの回転速度を変更することで強さに変化を与えることができますので、実験をして好みの強さに仕上げてください。

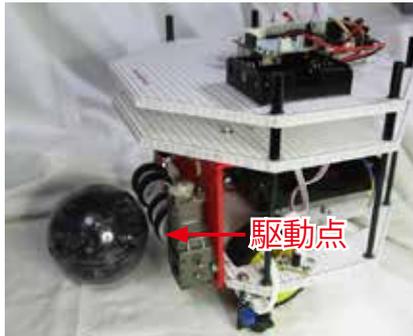


2). ドリブル駆動点高さ調整

1. 駆動点が高すぎるとボールにロボットが乗り上げてしまい、接点が低すぎると、ボールを把持出来なくなります。

2. 駆動点の高さを変更する場合は、ギアボックスとシャフト先端を取り付けているネジ/ナットの位置(4か所のネジ)を変えることで、変更します。

3. ドリブルがうまく働かどうかの調整ポイントです。何度も調整実験を行い、仕上げてください。



3). 測距センサの指向性調整

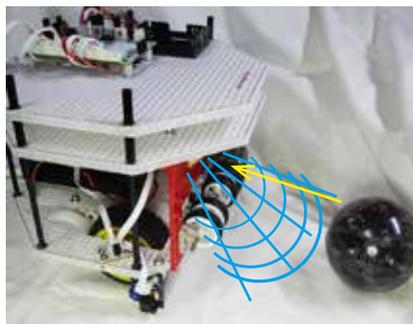
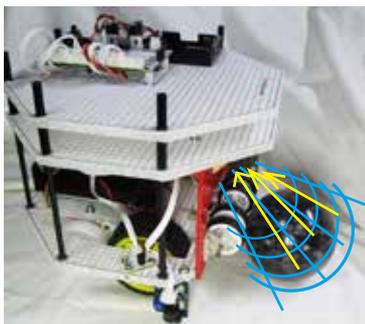
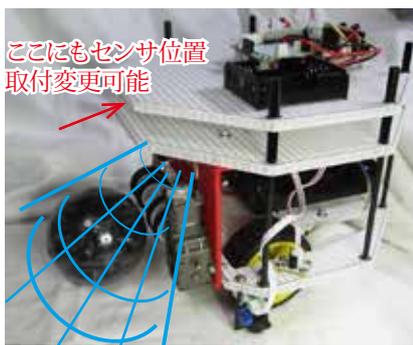
1. 図のイメージのように、ある範囲の角度(指向性)の中を、計測しますので、ボールを捕捉できる取り付け位置を考え、指向性に合わせます。

2. 実測をして、データに基づき、取り付け位置を変更します。

3. ボールの有無をレンジセンサ(測距)で測定し、ボールがある場合はドリブラーが高速回転してボールを把握します。

4. 実測し、センサデータに基づき「しきい値」を設定します。

5. ボールが直近と離れた時でのシリアルモニター計測値が730から600などへと変化します。このような場合レンジセンサ(測距)の「しきい値」を700としプログラムを書き換えます。



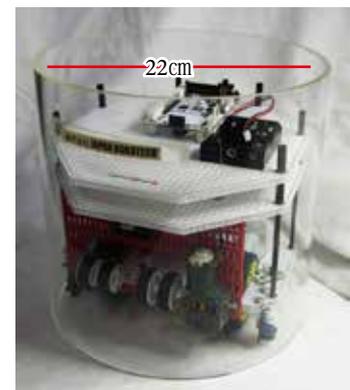
4). 大きさの調整

1. 作成後、ロボットサイズの規定にかかるときには、大きさの調整をします。



ロボット競技会の一つであるロボカップジュニア競技会では、ロボットの大きさ規定があり、直径22cmの円筒に納まること、高さ14cm以上あることとなっています。

図のような検査用車検シリンダーが準備されていて、試合開始前に実施される車検に合格することが必要です。

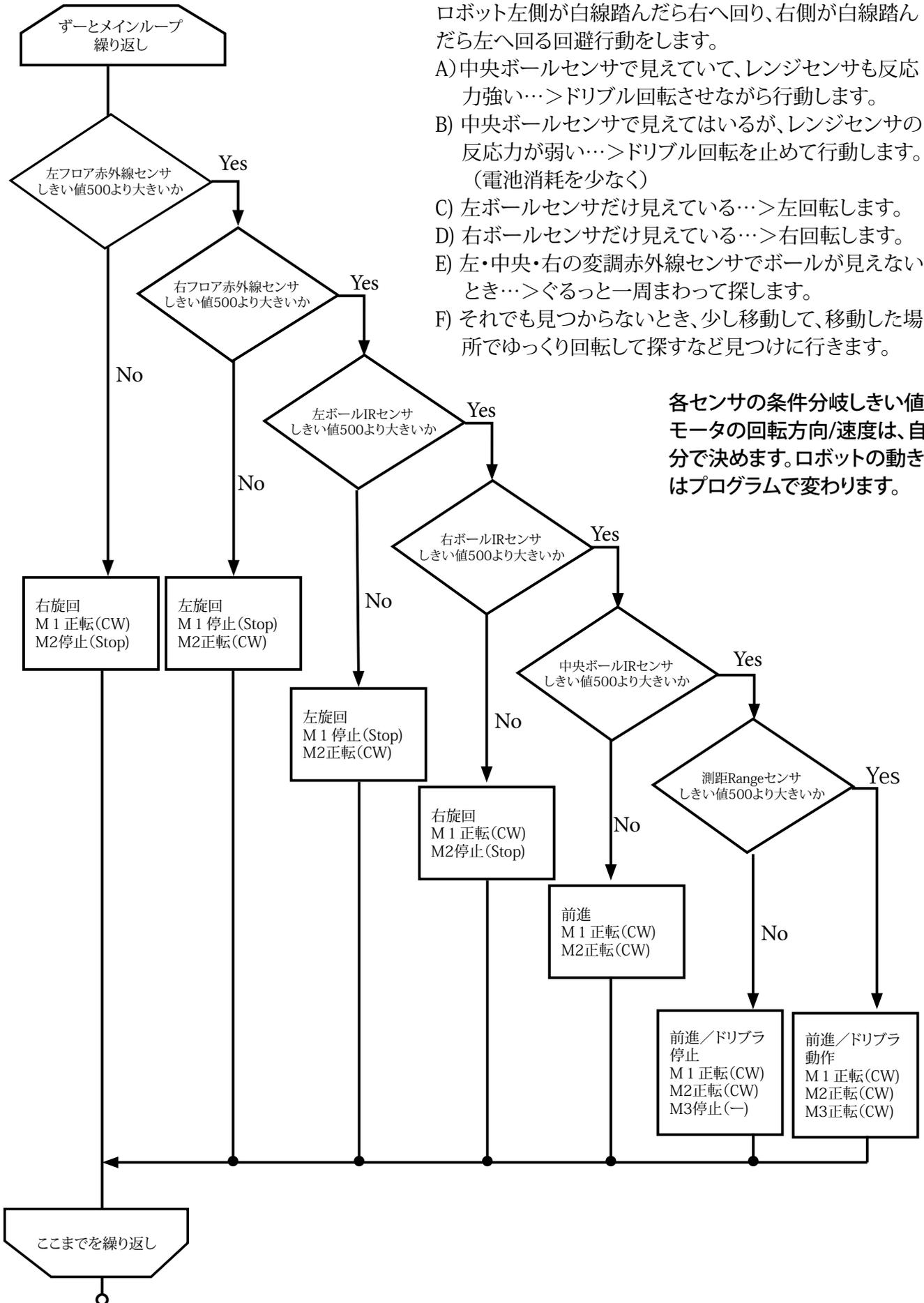


この組立ガイド通りに作成すると、この図の収まる大きさになりますが、オリジナル部品を追加するなどされる時、規定にパスするかどうか、仕上げサイズを確認してください。

※重さ規定はリーグにより異なります、ルールを確認ください。

10-9. ドリブルロボットのプログラム

1). 例:フローチャート図



基本:白線を越えないで、回避行動をします。

ロボット左側が白線踏んだら右へ回り、右側が白線踏んだら左へ回る回避行動をします。

A) 中央ボールセンサで見えていて、レンジセンサも反応力強い…>ドリブル回転させながら行動します。

B) 中央ボールセンサで見えてはいるが、レンジセンサの反応力が弱い…>ドリブル回転を止めて行動します。(電池消耗を少なく)

C) 左ボールセンサだけ見えている…>左回転します。

D) 右ボールセンサだけ見えている…>右回転します。

E) 左・中央・右の変調赤外線センサでボールが見えないとき…>ぐるっと一周まわって探します。

F) それでも見つからないとき、少し移動して、移動した場所でゆっくり回転して探すなど見つけに行きます。

各センサの条件分岐しきい値/モータの回転方向/速度は、自分で決めます。ロボットの動きはプログラムで変わります。

2). ドリブルロボットのプログラム作成 ・図を参考にプログラムを作成し、コントローラにアップロードしてください。

「base_floor_ball_sample」に測距センサを使ったドリブル動作を加えたプログラムです。
 センサ／モータは、小さい番号をロボット後ろから見て左側に配置するように作成してあります。

- ・アナログ赤外線センサ(フロア) JES-7023 A4/A5 変数 ir1, ir2
- ・変調赤外線センサ(ボール) RDI-203JR A0/A1/A2 変数 pulse_ir1, pulse_ir2, pulse_ir3
- ・モータ M1/M2/M3
- ・測距センサ(レンジ) RDI-209 A3

※ RDI-203JR はボールの距離が近いと出力が小さくなります。
 ※ RDI-209 はボールの距離が近いと出力が大きくなります。約10cm以内からは逆に出力値が下がりますので注意してください。

動作手順

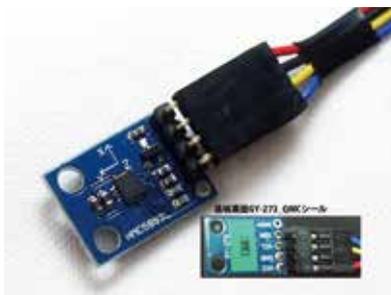
- ・ir1, ir2 の値が「しきい値」以上になったら停止／旋回します。
- ・床の色が暗い場合は赤外線ボールを探します。
- ・左変調赤外線センサpulse_ir1 が「しきい値」以下になったら左に旋回します。
- ・右変調赤外線センサpulse_ir3 が「しきい値」以下になったら右に旋回します。
- ・中央変調赤外線センサpulse_ir2 が「しきい値」以下になったら前進してドリブラーを回転させます。
- ・range が「しきい値」以上になったら停止して、ドリブラーでボールを保持したまま旋回します。
- ・左、右、中央の変調赤外線センサが3 つとも反応しないときには、右旋回します。

各パラメータを調整して、動作実験をします。
 JES-7023/RDI-203JR/RDI-209 の「しきい値」設定
 M1 / M2 / M3 のモータスピード設定
 ボールの距離によってセンサの値が変わります。
 遠距離からのボール探索と、近づいたときの動作はサンプルのしきい値調整だけでは両立できないでしょう。
 条件分岐を追加するなど、工夫してみてください。

※ソースコードは、PC/MyDocuments/Arduino/へ配置したサンプルフォルダ [ArduBlock Examples] に、ファイル名【24_02_challenge_range_kick_sample.abp】で格納されています。
 Arduino/Ardublock で、PC/MyDocuments/Arduino/ArduBlock Examples の中に配置したサンプルを開くと確認できます。

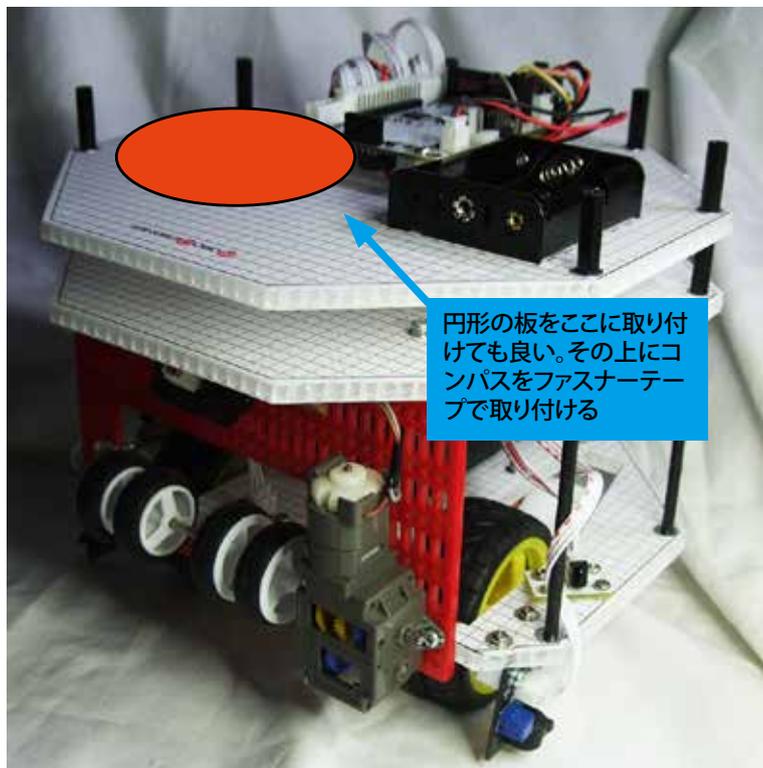
10-10. コンパスセンサを追加

競技場のゴールの色をカラーセンサで認識して行動するロボットもありますが、カラーセンサが高価なので、ここでは地磁気を利用するコンパスセンサについて調べます。



■RDI-5883L_QMC

1). 取付方法



①ファスナーテープ(接着タイプ)を準備ください。(100均店舗などで販売されています)

②上図のように、ファスナーテープの一方をコネクタ裏面に貼付け、ロボットの取り付けたい位置に片方の面を貼り付けます。

③右図のように、円形の板を取り付け、その上にコンパスを取り付けるのも良いでしょう。板はホームセンターなどでアクリル円形板が販売されています。

円形の板をここに取り付けても良い。その上にコンパスをファスナーテープで取り付ける

③ある方位に対して攻勢に出るなどのプログラムをして、競技開始前に、自陣でロボットを設置して、コンパスセンサの角度を攻勢方角へ設定し、ファスナーテープで貼り付けて固定します。

④競技では、前半戦と後半戦で、コートチェンジをしますので、コンパスセンサの方向を変えて張り替えることで、ロボットに方位情報変更を知らせます。(前半戦、後半戦で方位認識が変わり、オウンゴールを防げます。)

この時に、付けたり外したりが簡単にできるようにファスナーテープ利用を推奨しています。ファスナーテープ(別名マジックテープ)

コンパスのソースコードは、PC/MyDocuments/Arduino/ ファイル/スケッチブックへ配置したサンプルフォルダ[RDS-X Examples C]に、ファイル名【RDC-QMC-5883】で格納されている【QMC-5883_Compass】のサンプルを開くと確認できます。

サンプルプログラム C言語で準備

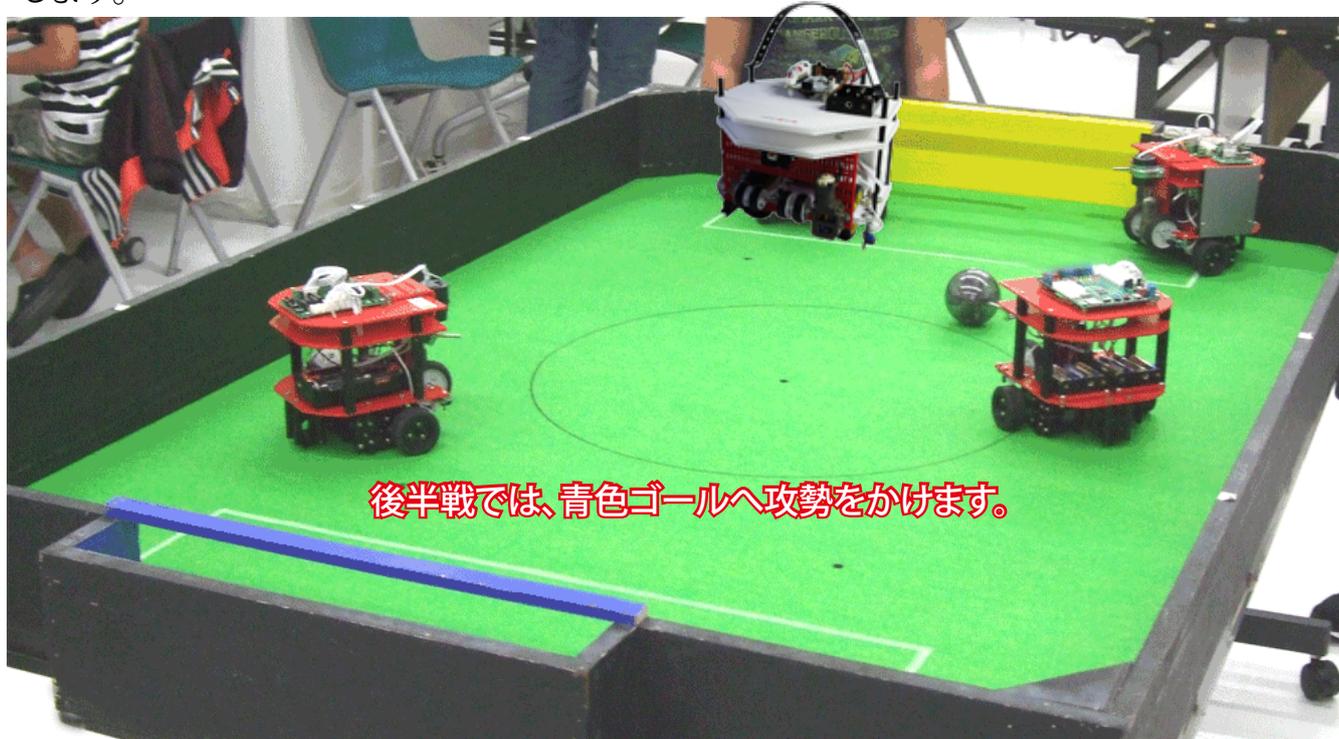
2). 競技で方向認知

前半戦と後半戦で、コートチェンジをします。方向認識がないと、プログラムに従いボールを必死で追いかけて、ついオウンゴールをしてしまいます。方向を認知させることが、競技の決め手になります。



前半戦では、黄色ゴールへ攻勢をかけます。

ある方位に対して攻勢に出るなどのプログラムをしておき、競技開始前に、自陣でロボットを設置して、コンパスセンサの角度を攻勢方角へ設定し、ファスナーテープで貼り付けて固定します。



後半戦では、青色ゴールへ攻勢をかけます。

競技では、後半戦で、コートチェンジをしますので、ハーフタイム中にコンパスセンサの方向を前半戦と真逆の方位に変えて張り替えることで、ロボットに方位情報変更を知らせます。(前半戦、後半戦で方位認識が変わり、オウンゴールを防げます。)プログラム変更しなくてコンパスの方向を張り替えるだけで対策が済みます。

2). コンパスセンササンプルコード例

```

/ * !
 * @file QMC5883_compass.cpp
 * @brief The program shows how to realize the
function compass. When the program runs, please
spin QMC5883 freely to accomplish calibration.
 * @n 3-Axis Digital Compass IC
 *
 * @copyright [DFRobot] (http://www.dfrobot.
com), 2017
 * @copyright GNU Lesser General Public License
 *
 * @author [dexian.huang] (952838602@qq.com)
 * @version V1.0
 * @date 2017-7-3
 */

#include <Wire.h>
#include <DFRobot_QMC5883.h>

DFRobot_QMC5883 compass;

void setup()
{
    Serial.begin(115200);
    while (!compass.begin())
    {
        Serial.println("Could not find
a valid QMC5883 sensor, check wiring!");
        delay(500);
    }

    if (compass.isHMC()) {
        Serial.println("Initialize
HMC5883");
        compass.setRange(HMC5883L_
RANGE_1_3GA);
        compass.setMeasurement-
Mode(HMC5883L_CONTINUOUS);
        compass.setDataRate(HMC5883L_
DATARATE_15HZ);
        compass.setSamples(HMC5883L_
SAMPLES_8);
    }
    else if (compass.isQMC()) {
        Serial.println("Initialize
QMC5883");
        compass.setRange(QMC5883_
RANGE_2GA);
        compass.setMeasurement-
Mode(QMC5883_CONTINUOUS);
        compass.setDataRate(QMC5883_
DATARATE_50HZ);
        compass.setSamples(QMC5883_
SAMPLES_8);
    }
}

```

```

    }
}

void loop()
{
    Vector norm = compass.readNormal-
ize();

    // Calculate heading
    float heading = atan2(norm.YAxis,
norm.XAxis);

    // Set declination angle on your
location and fix heading
    // You can find your declination on:
http://magnetic-declination.com/
    // (+) Positive or (-) for negative
    // For Bytom / Poland declination
angle is 4'26E (positive)
    // Formula: (deg + (min / 60.0)) /
(180 / M_PI);
    float declinationAngle = (4.0 +
(26.0 / 60.0)) / (180 / PI);
    heading += declinationAngle;

    // Correct for heading < 0deg and
heading > 360deg
    if (heading < 0) {
        heading += 2 * PI;
    }

    if (heading > 2 * PI) {
        heading -= 2 * PI;
    }

    // Convert to degrees
    float headingDegrees = heading *
180/M_PI;

    // Output
    Serial.print(" Heading = ");
    Serial.print(heading);
    Serial.print(" Degress = ");
    Serial.print(headingDegrees);
    Serial.println();

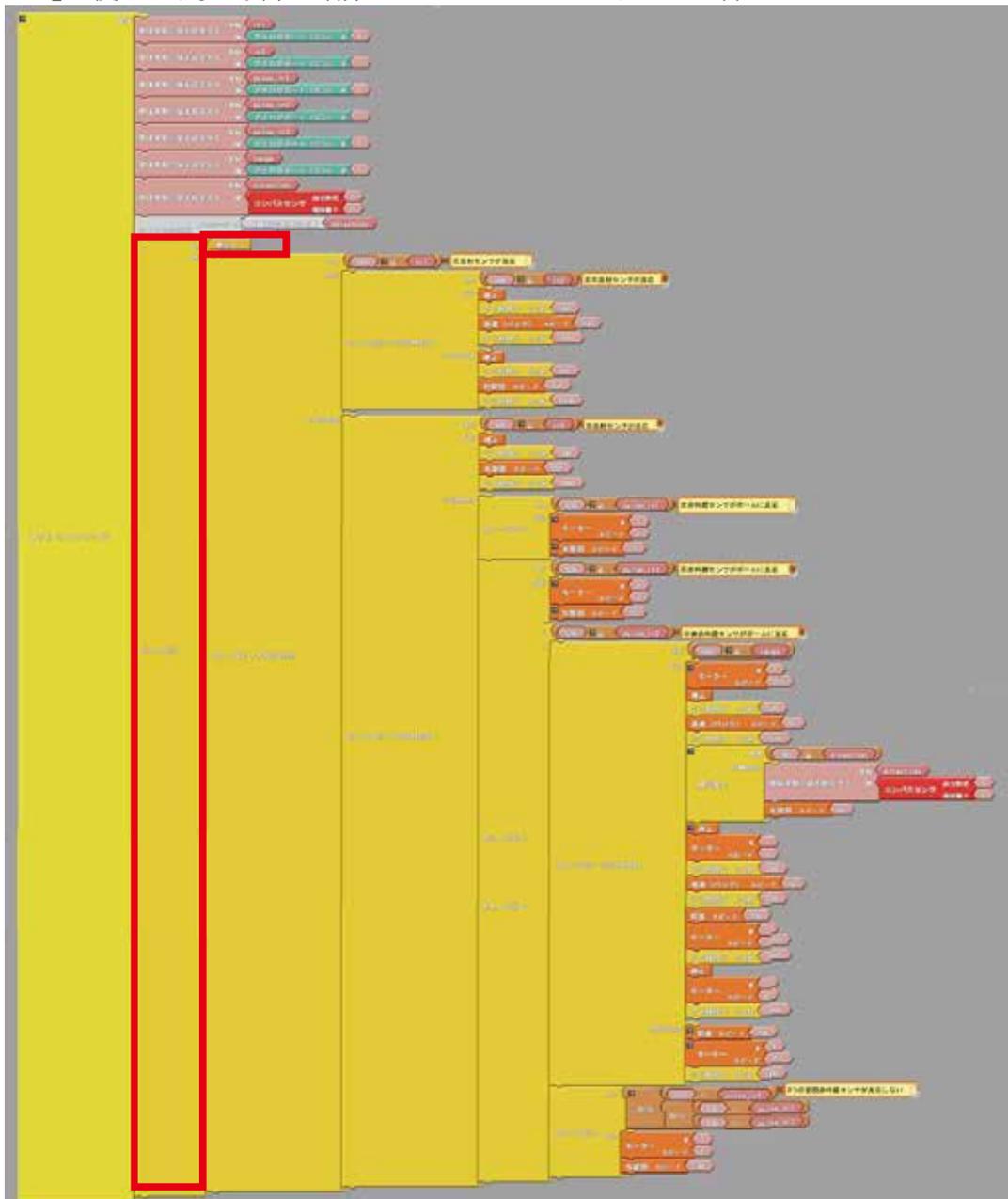
    delay(100);
}

```

コンパスのソースコードは、PC/MyDocuments/Arduino/ スケッチブックへ配置したサンプルフォルダ[RDS-X Examples C] に、ファイル名【RDC-QMC-5883】で格納されている【QMC-5883_Compass】のサンプルを開くと確認できます。

10-11. プログラムをボタンでスタートさせる時。

ロボットは動作スタートスタンバイ状態で、競技試合スタート合図で動作開始する時などは「**押ボタン**」「**押ボタン**」を使います。以下は、前ページのドリブルプログラムを押ボタンスタートにしたプログラム。



ドリブルロボットプログラムに追加したアイコン

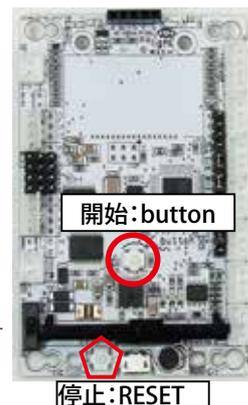


- ・「押ボタン」は、STEM Duパレットの中に格納されています。
- ・「もし~なら」は、制御パレットの中に格納されています。

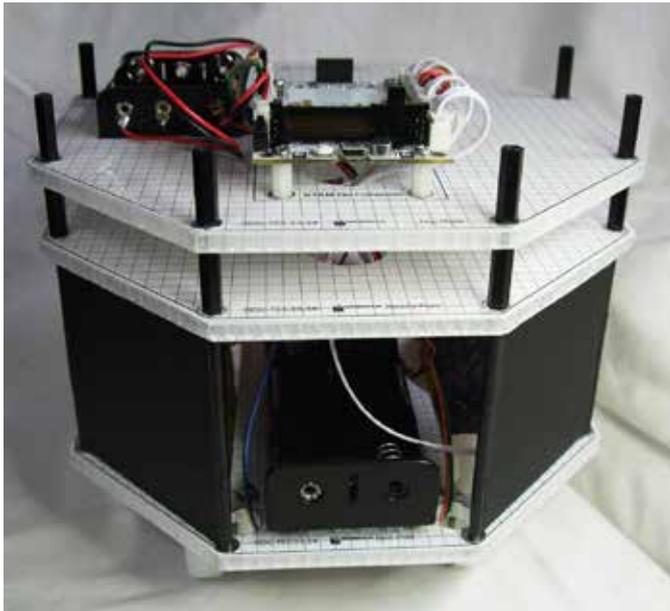
実験:前進を押ボタンスタートさせるプログラムです。



ソースコードは、PC/MyDocuments/Arduino/へ配置したサンプルフォルダ[ArduBlock Examples]に、ファイル名【24_05_base_floor_ball_button_sample.abp】で格納されています。Arduino/ArduBlockで、PC/MyDocuments/Arduino/ArduBlock Examplesの中に配置したサンプルを開くと確認できます。

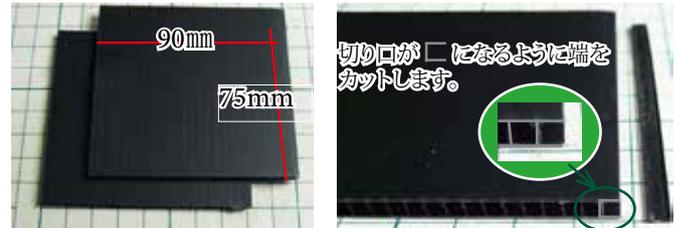


10-12. バックカバーを取り付け、ロボットの形を完成させる。



組み立ての最後に、バックカバーを取り付けます。

1. ロボット背後からの光の影響などを受けないようにするための光遮蔽目的があります。
2. 競技中に、他のロボットの突起物がロボット体内に突っ込まれないようにする保護目的もあります。



10-13. ロボットに独自のデザインをする

1. カバーに使用する色は、青と黄色以外を使います。青、黄色は、ロボカップジュニアサッカー競技のゴールの色として指定されており、ゴール以外では使用できないルールになっています。

ロボットからLEDの青色、黄色の可視光を発光することも禁止されています。

カラーセンサを搭載したロボットが試合に出るとゴールをカラーで判断しますので、ゴール以外に同じ色があると誤解して動作するからです。ロボットだけでなく、競技出場者も同じ色の服装は着用してはいけません。

2. 使用する材料は、サンプルとして同梱されていたプラダンなどが手頃で良いでしょう。ホームセンターなどで90cm×180cmサイズが販売されています。白やグレー色、黒色、赤色など多くの色が揃っています。
 - ・ペイントする時に使用する材料は、ホームセンターなどで売られている水性塗料などが良いでしょう。
 - ・塗膜が乾いて使用可能な強度になるまでに、24時間ほど必要ですので、作業計画を考えながら、塗装に着手しましょう。

10-14. ロボット完成直前の総合調整

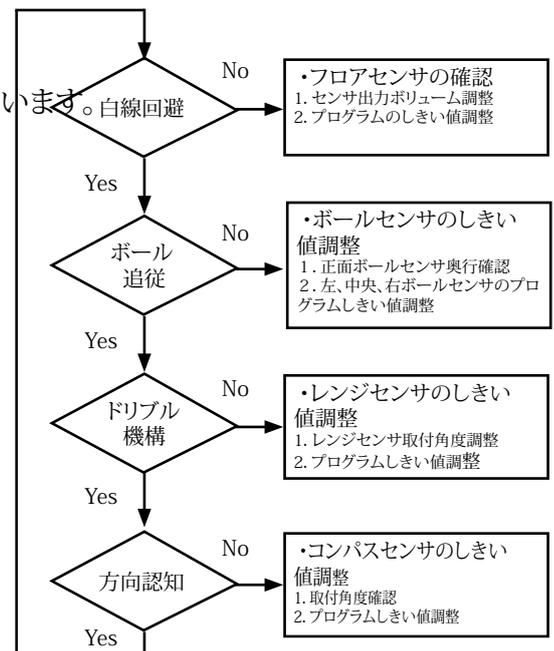
1. 総合調整では、各センサの動きを1個ごとに確認することを行います。白線回避

- ① 白線を越えずに活動できるか:フロアセンサ確認
- ② ボールへの追従性は満足できるか:ボールセンサ調整
- ③ ドリブルは満足できるか:レンジセンサ調整
- ④ オウンゴールを防げるか:コンパスセンサ調整

3. 各動作の1個ごとに調整を施します。

各センサ接続ケーブルを外して実行してみる

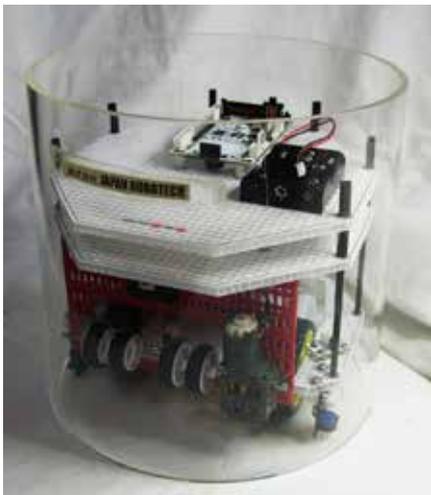
4. 重要なのは、ロボットを動作させる環境と同じ状態で、調整なされていること:電池容量、照度、電磁波などの影響で地磁気は乱れていないか等々、要素は多くありますが、環境に合わせた調整をロボットに施します。



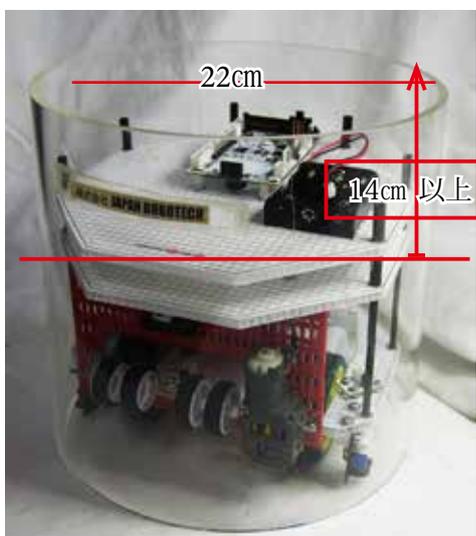
10-15. 競技会規定



・サッカー競技車検シリンダー



・車検シリンダーの中に入る大きさにします。

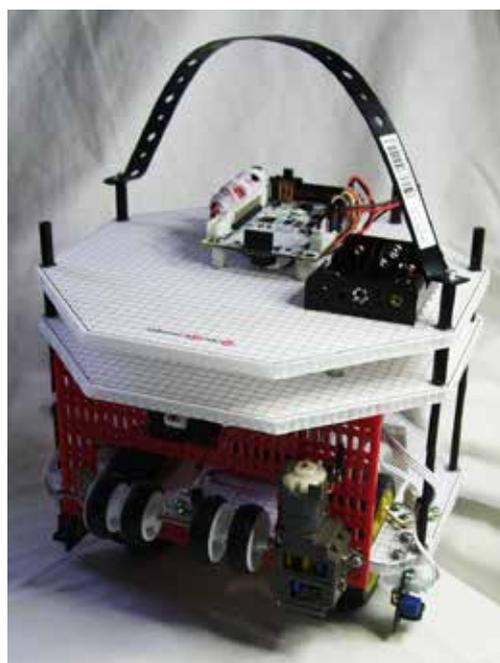


・ビギナーズリーグ



・サッカーBリーグ

・電源電圧/ロボット重さなど、競技会のリーグごとに発表される参加規程を確認ください。



ロボット試合出場時には、持ち手を取り付けることが必要です。ホームセンターで購入した幅15mmx長さ289mm厚さ1mmの曲げ板金物を参考に付けてみました。樹脂スペーサー六角支柱にM3ネジで取り付け可能です。