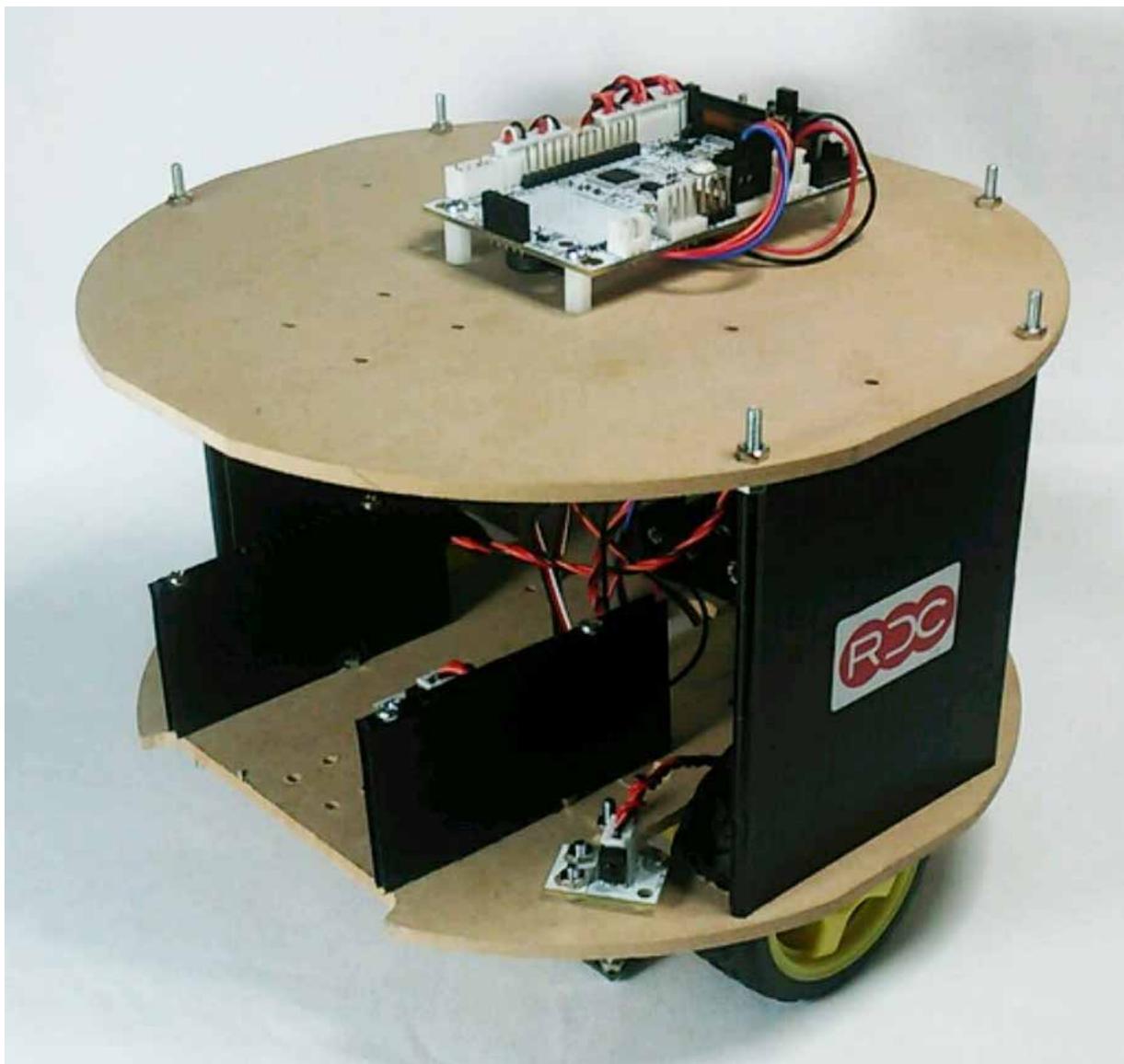


9. ベースロボ Soccer Base Robo RDS-X23_BS3

Ball Sensor 3台仕様



搭載コントローラボードにより

Typel (液晶なし、使用可能モータ数 (DC モータ×2、サーボモータ×8)、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×2、サーボモータ×8)、超音波センサ増設可能)

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)

Typelll+ (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― を搭載しており、これらを利用して各種制御を行えます。

9-1. 部品の見方、使い方

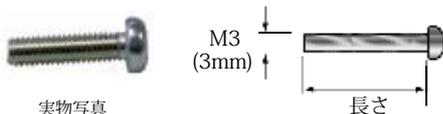
1. 部品サイズ表示

ネジやナットのサイズ表示は以下のとおりです。

■ ^{S C R E W} 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。



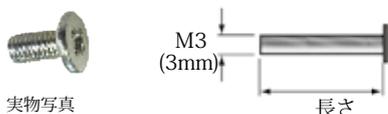
※ネジが切っただけの長さです。

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。



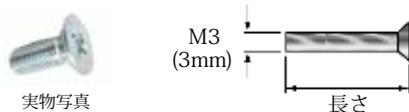
※ネジが切っただけの長さです。

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。



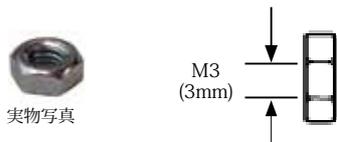
※ネジの頭からの長さです。

Nut

■ ナット

表記：M3

「太さ（直径）3mm のネジ用」という意味です。



■ 共通で使う上記以外のパーツ



・座金付組ネジ ・スペーサ ・バネ座金 ・平座金

2. 長さ測定用スケール



3. 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。

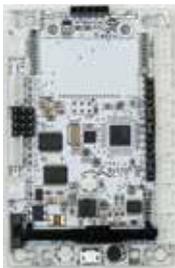
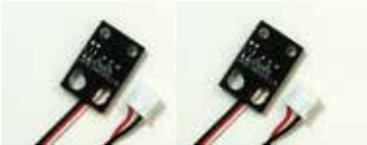
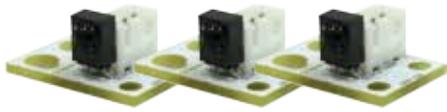
	電子回路に接続する電源電圧のプラス / マイナスを間違えないように注意ください。間違えると部品が壊れます。
--	---

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。
--	--

9-2. 事前準備 部品確認

9.2.1 ベースロボRDS-X23_BS3_TYPEⅢパーツリスト

<p>□コントローラ RDC-103_TYPEⅢ 1台</p>  <p>取付部品:樹脂スペーサ 10mm x 4 ネジ M3x6mm 4個</p> <table border="1" data-bbox="231 627 710 683"> <tr> <td></td> <td>ナベ小ネジ Screw M3x8mm</td> <td>4本</td> </tr> </table>		ナベ小ネジ Screw M3x8mm	4本	<p>□ベースプレート RDP-811_BS3 1枚</p> 	<p>□トッププレート RDP-811_Top23 1枚</p> 																		
	ナベ小ネジ Screw M3x8mm	4本																					
<p>□赤外線反射センサ- RDI-211 2台</p>  <table border="1" data-bbox="231 952 710 1052"> <tr> <td></td> <td>ナベ小ネジ Screw M3x50mm</td> <td>4本</td> </tr> <tr> <td></td> <td>M3ナット Nut</td> <td>12個</td> </tr> </table>		ナベ小ネジ Screw M3x50mm	4本		M3ナット Nut	12個	<p>□ロボット外装材</p>  <p>プラダン W85mm x H100mm 4枚 RDCシール 4枚</p>	<p>□支柱セット</p>  <p>ネジ Screw M3x120mm 6本 スペーサー Spacer 25mm 24本 ナット Nut 12個</p> <p>□双輪キャスター</p>  <table border="1" data-bbox="1061 1120 1468 1310"> <tr> <td></td> <td>皿ネジ Flat Head Screw M3x10mm</td> <td>4本</td> </tr> <tr> <td></td> <td>平座金 Washer M3x 8mm</td> <td>4個</td> </tr> <tr> <td></td> <td>M3ナット Nut</td> <td>4個</td> </tr> </table> <p>□電池ケース</p>  <table border="1" data-bbox="1061 1344 1468 1478"> <tr> <td></td> <td>皿ネジ Flat Head Screw M3x10mm</td> <td>2本</td> </tr> <tr> <td></td> <td>M3ナット Nut</td> <td>4個</td> </tr> </table>		皿ネジ Flat Head Screw M3x10mm	4本		平座金 Washer M3x 8mm	4個		M3ナット Nut	4個		皿ネジ Flat Head Screw M3x10mm	2本		M3ナット Nut	4個
	ナベ小ネジ Screw M3x50mm	4本																					
	M3ナット Nut	12個																					
	皿ネジ Flat Head Screw M3x10mm	4本																					
	平座金 Washer M3x 8mm	4個																					
	M3ナット Nut	4個																					
	皿ネジ Flat Head Screw M3x10mm	2本																					
	M3ナット Nut	4個																					
<p>□変調赤外線センサ- RDI-203JR 3台</p>  <table border="1" data-bbox="231 1243 710 1467"> <tr> <td></td> <td>センサ接続コード 30 c m RDP-831</td> <td>3本</td> </tr> <tr> <td></td> <td>ナベ小ネジ Screw M3x12mm</td> <td>6本</td> </tr> <tr> <td></td> <td>ナベ小ネジ Screw M3x 6mm</td> <td>6本</td> </tr> <tr> <td></td> <td>M3ナット Nut</td> <td>18個</td> </tr> </table>		センサ接続コード 30 c m RDP-831	3本		ナベ小ネジ Screw M3x12mm	6本		ナベ小ネジ Screw M3x 6mm	6本		M3ナット Nut	18個	<p>□マイクロUSBケーブル RDP-824 1本</p> 	<p>□センサ仕切板</p>  <p>プラダン W80 x H80mm 2枚 ネジ Screw M3x50mm 4本 ナット Nut 4個</p>									
	センサ接続コード 30 c m RDP-831	3本																					
	ナベ小ネジ Screw M3x12mm	6本																					
	ナベ小ネジ Screw M3x 6mm	6本																					
	M3ナット Nut	18個																					
<p>□ギアードモータ&ホイールRDO-502 2台</p>  <p>取付部品:マウンター x 2個 ネジ M3x30mm x 4個 ナット x 4個 モータ接続コード30 c m 2本</p> <table border="1" data-bbox="231 1803 710 2027"> <tr> <td></td> <td>ナベ小ネジ Screw M3x8mm</td> <td>4本</td> </tr> <tr> <td></td> <td>ナベ小ネジ Screw M3x 6mm</td> <td>2本</td> </tr> <tr> <td></td> <td>M3ナット Nut</td> <td>2個</td> </tr> <tr> <td></td> <td>レジンスペーサー Resin Spacer</td> <td>2個</td> </tr> </table>		ナベ小ネジ Screw M3x8mm	4本		ナベ小ネジ Screw M3x 6mm	2本		M3ナット Nut	2個		レジンスペーサー Resin Spacer	2個	<p>□開発環境 DISK</p>  <p>□補修用ネジ・ナット</p>  <p>ネジM3x10mm 4本 ナット 10個</p>	<p>◆一緒に使う別売品案内 コンパスセンサ RDL5883L</p> 									
	ナベ小ネジ Screw M3x8mm	4本																					
	ナベ小ネジ Screw M3x 6mm	2本																					
	M3ナット Nut	2個																					
	レジンスペーサー Resin Spacer	2個																					

9.2.2. コントローラ 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)

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)

仕様	The specification	DataSheet URL
マイコン / ATMEGA32U4、発信周波数 8MHz	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/
明るさセンサー	Light sensor Everlight PT12-21C	http://www.everlight.com/file/ProductFile/PT12-21C-TR8.pdf

・ RDC-103 には、いろいろな文字や記号が描かれていますが、大きく分けると、センサコネクター、モータコネクター、電源コネクター、USB コネクターの 4 つです。

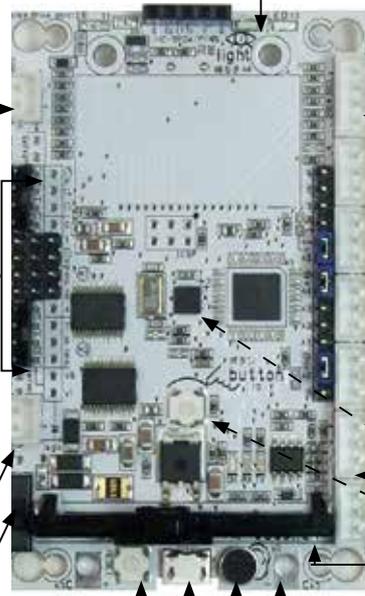
デジタル入出力 Digital in/out

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

RESET スイッチ

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



- 明るさセンサ Light sensor
発光 LED 白色 / 受光 LED
- I²C コネクター 3 SCL, 2 SDA
- アナログ入力 Analog input

A0 (A は Analog の A) から A5 までの計 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	記号	解説
A0	A0	音センサ / アナログ入力コネクター Sound Sensor / Analog input
A1	A1	アナログ入力コネクター Analog input
A2	A2	アナログ入力コネクター Analog input
A3	A3	みの虫クリップ Signal Clip terminal
A4	A4	明るさセンサ / アナログ入力コネクター Light Sensor / Analog input
A5	A5	スライダ / アナログ入力コネクター Slider / Analog input

接続コネクター JST connector XH3B



LED

記号	解説
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

- 加速度 / ジャイロ / 温度センサ (I2C) Accelerometer/Gyroscope
- ボタン button
- スライダ (可変抵抗) Slider resistance
- みの虫クリップ用端子 Terminal for clips (抵抗等を測ります)
- 音センサ Sound sensor
- USB コネクター USB connector

9.2.3 ギアボックス組立て

ギアボックスを左右対象で作成します。

(1) マウンター取付

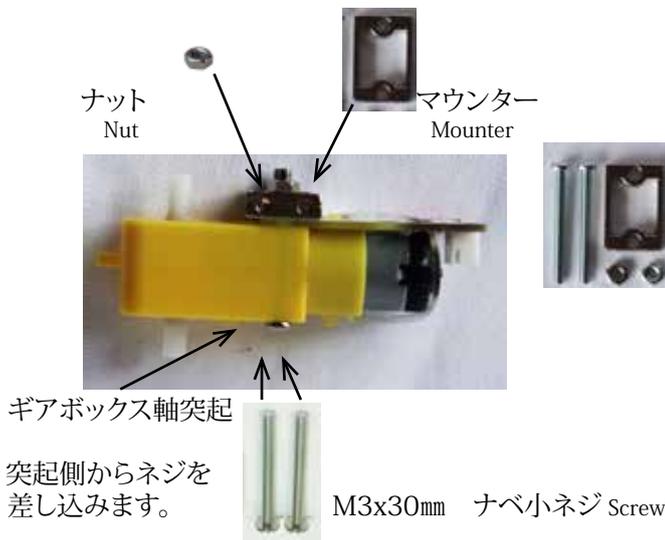
[Assemble of a Mounter]

ギアードモータにマウンターを取り付けます。

(検査のため取り付けられ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。)

(図では左右ともにネジ穴が上を向いています)

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。



(2) タイヤホイールの組み込み

[Assemble of a Tire Wheel]



①モータ出力軸とホイールの差込口の長円形の方を合わせます。



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。

Geared motor assembly

A gearbox is made by symmetricalness.

(1). Mounter assembly
Mounting hardware (mounter) is assembly in the geared motor.



Down view



Side view



完成品: 2 個作ります。

Finished goods: 2 are made.

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。

ホイールの組み込み図 Fig Wheel assembly

9-3. ロボットベースプレートへの組み込み

9.3.1 ギアードモータを取り付ける。

1. 準備で作製したモーターを、ベースプレートに取り付けます。

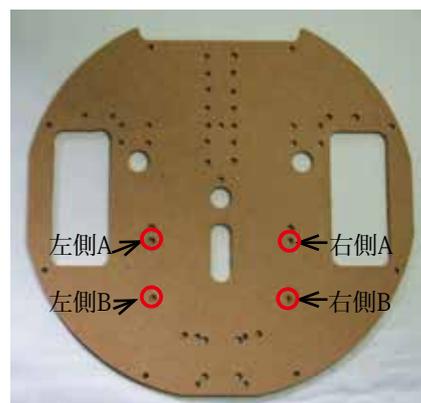
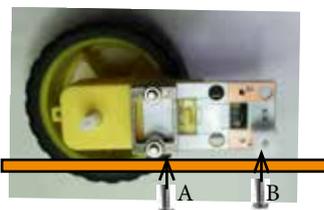
2. 取り付け位置は、図の配置を参照ください。

A: M3x8mmネジを、マウンターのネジ穴位置に合わせて仮止めします。

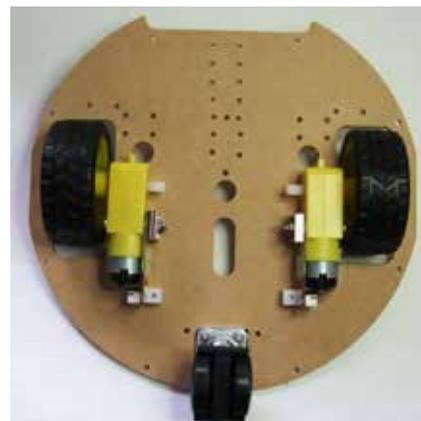
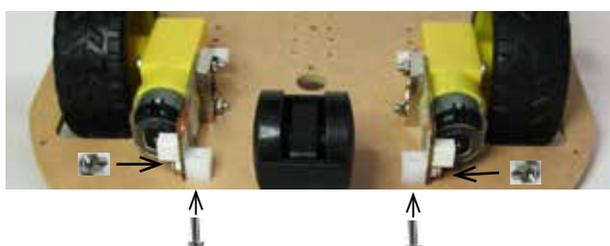
B: レジンスパーサ8mm角をベースプレートBの位置にM3x8mmのネジで取り付けます。

3. M3x6mmのネジにナットを取り付けます。

4. 作成したM3x6mmナット付きのネジを、基板の穴を通してレジンスパーサーへ取り付けます。

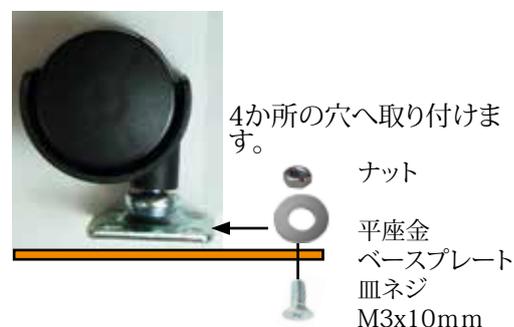
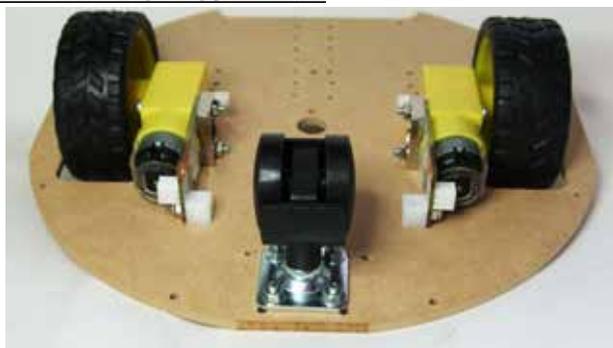


(ネジ長さ調整)



A,Bの位置での仮止めが完了後、モーター取付角度で左右タイヤの並行性を確認して、A,B左右で4か所のネジを締め付けて固定します。

9.3.2 双輪キャストを取り付ける。



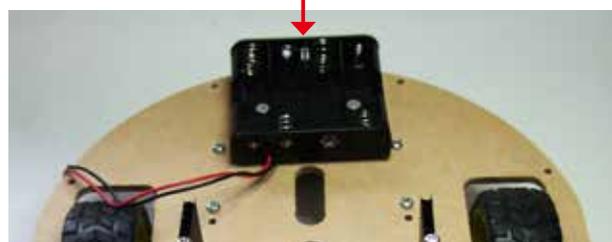
9.3.3 電池ボックスを取り付ける。

1. 電池ボックスに皿ネジを差し込み、裏面にナットを取り付けます。

2. 作成したネジ付の電池ボックスを、双輪キャストの反対面に取り付けます。

3. 電池ボックスのネジへ取り付けしたナットの厚み部分がスペーサの役目を果たし、双輪キャスト取付ねじ頭厚み(皿ネジで、厚みは少ない)の空間ができますので、電池ボックス/キャスター上下の取り付けが可能です。

※双輪キャスト、電池ボックスの取付には、皿ネジを使用してください。

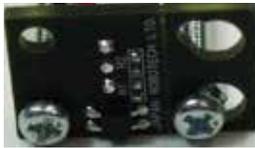


9.3.4 赤外線反射センサーを取り付ける。



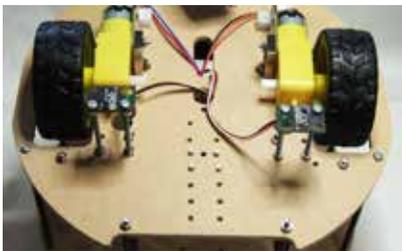
部品名		サイズなど	使用数
	ナベ 長ネジ	M3x50mm	4個
	ナット	M3	12個

1. 赤外線反射センサーを組み立てます。



・長ネジを赤外線反射センサーに取り付けます。基板の穴は丸穴と瓢箪型穴があります。瓢箪型穴へネジを取り付ける時は小さい穴側へネジを取り付けます。

2. モーター前方の図の位置に、組み立てた赤外線反射センサーを取り付けます。



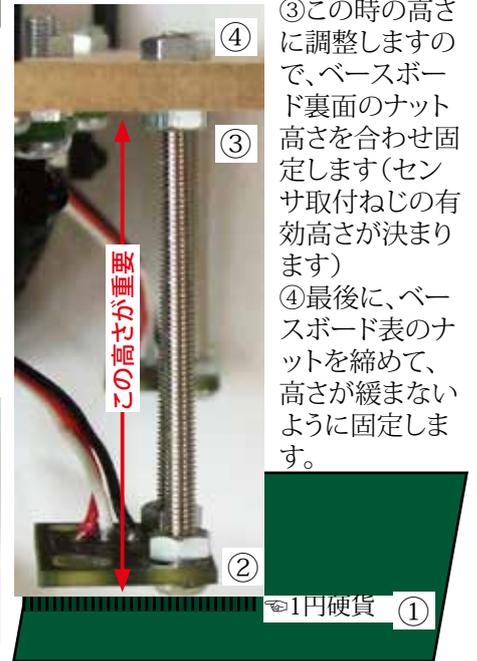
👉 前から見た図



👉 横から見た図

取付手順【高さ調整】

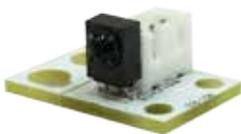
- ①競技台の上に1円硬貨を置き
- ②センサー取付ねじ頭を1円硬貨に接触させる



- ③この時の高さに調整しますので、ベースボード裏面のナット高さを合わせ固定します(センサー取付ねじの有効高さが決まります)
- ④最後に、ベースボード表のナットを締めて、高さが緩まないように固定します。

9.3.5 変調赤外線センサーを取り付ける。

ボールセンサ 3個使用



部品名		サイズなど	使用数
	ナベ小ネジ	M3x12mm	6本
	ナベ小ネジ	M3x6mm	6本
	ナット	M3	18個

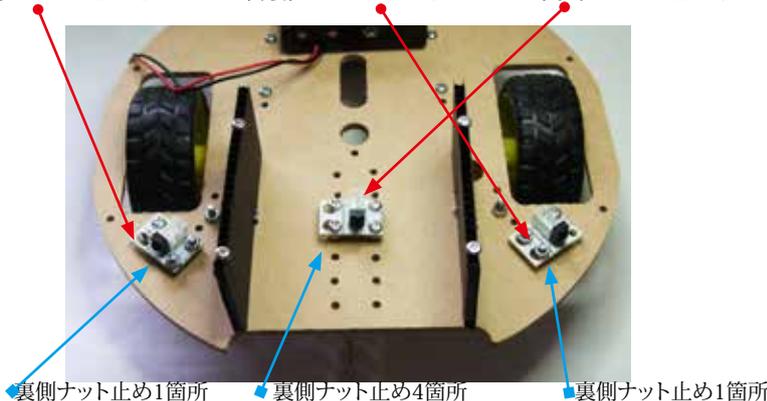
1. 左側、右側、中央の3箇所のセンサは各々、ネジ足の使い方が異なります。下図を確認し、ネジを取り付けます。



左側ボールセンサ

右側ボールセンサ

中央ボールセンサ



👉 裏側ナット止め1箇所

👉 裏側ナット止め4箇所

👉 裏側ナット止め1箇所

センサ調整

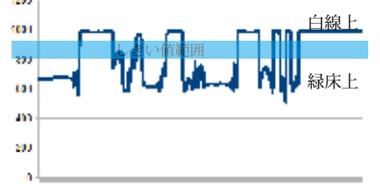
赤外線反射センサー表面から、床までの距離を2～3mmになるように調整してください。



- ・日本の1円硬貨が厚さ約1.5mm、ネジ先端からセンサ表面まで約0.5mmあります。図のように調整すると床からセンサ表面まで2mmの隙間になります。
- ・データでは、焦点距離は1mmの規格になっていますが、サッカー競技用ロボでは、フィールド床緑色とライン白色の色の違いでの反射出力データ差の違いが出るように実験にて高さ調整を決めます。



- ・床までの距離が離れすぎると、データは得られなくなります。近すぎると、白と緑の差が出なくなります。
- ・ネジを回して、センサ高さを変更して、シリアルモニターでデータを取得し、上図のような差が得られるようにセンサ高さを調整をします。
- ・センサ高さ調整後、得られたデータに基づき、「しきい値」を考察し、プログラムに反映します。



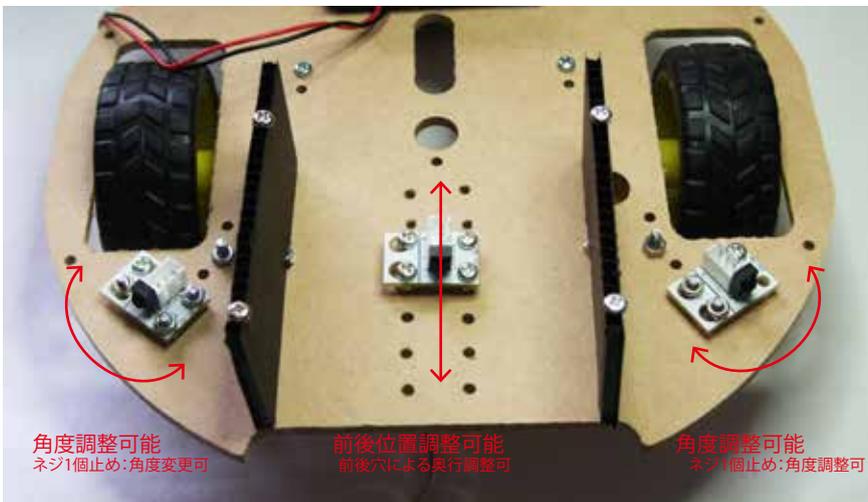
9.3.6 センサ仕切板を取り付けます。

部品名		サイズなど	使用数
	プラダン	40x80mm	2枚
	ナベ 長ネジ	M3x50mm	4個
	ナット	M3	4個

1. ボール捕捉性向上を目指して、センサ指向性を調整する仕切板を取り付けます。

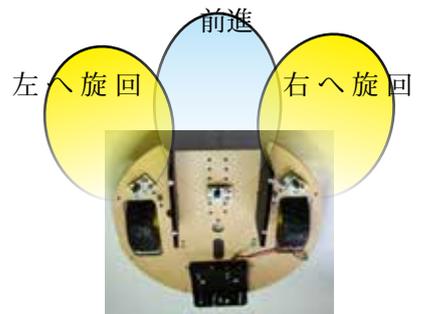


プラダン仕切板取付位置に、ベースボード下部より部品取付時のネジ先端がありますので、左図のようなナット高さ分の切れ込みを作ります。ネジ部はプラダンに入ります。切れ込み位置は現物合わせで決めます。



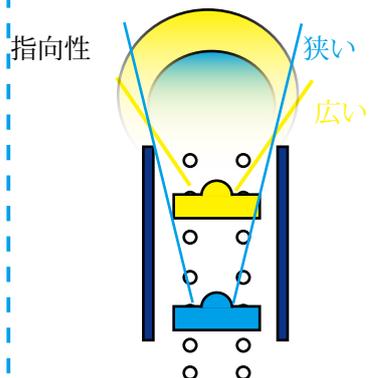
★仕切板のはたらき

\\(◎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、右センサの指向性を仕切板で制限することにより、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。

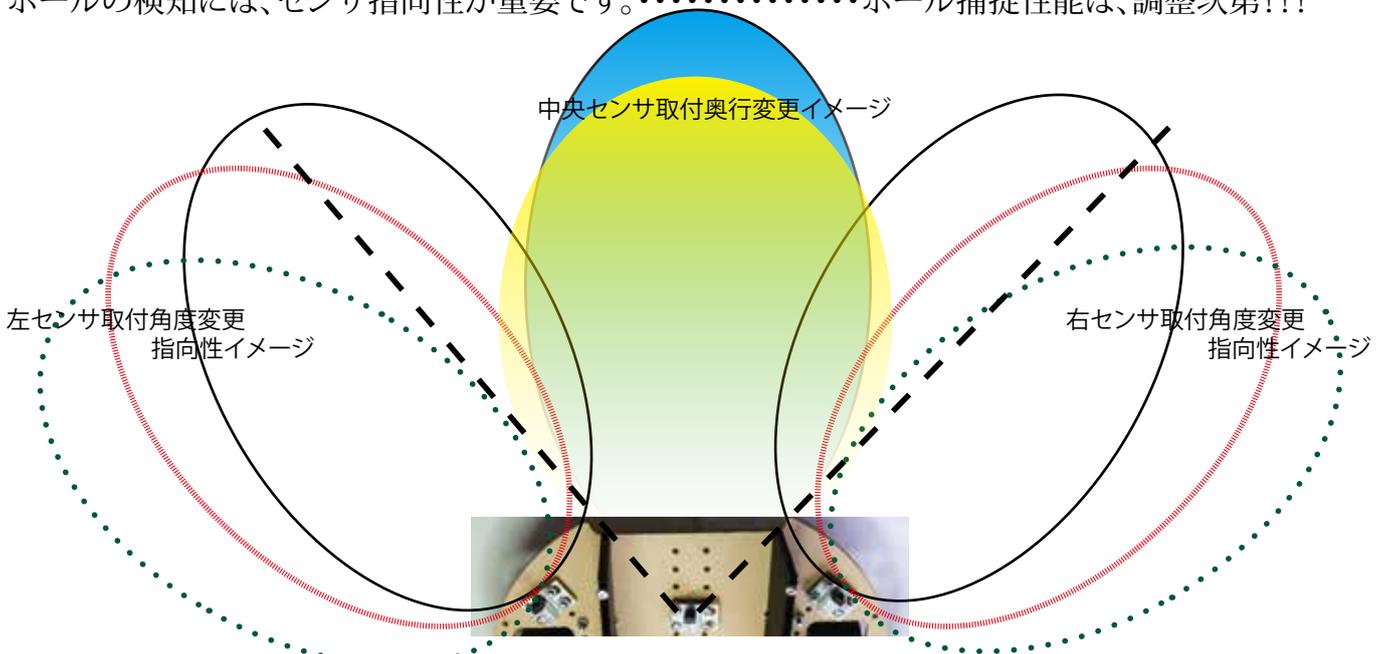


ボール検知範囲とボール捕捉行動の関係図

☞中央センサの奥行きで、センサ指向性の調整ができます。



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!!!



9-4. ロボットトッププレートへの組み込み

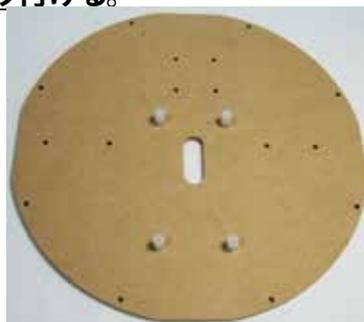
9.4.1 コントローラ用樹脂スペーサを取り付ける。

部品名	サイズなど	使用数
 ネジ Screw	M3x8mm	6本
 樹脂スペーサー Resin Spacer	M3x10mm	6本

樹脂スペーサーは、M3x6mmネジでコントローラに取り付けてありますので、取り外して使用します。



※基板に取り付けてあるネジM3x6mmは、あとで基板取付時に使用しますので、保管ください。



9-4. Top plate is assembled.

A resin spacer for controllers is assembled.

9-5. ロボットを組み立てる

9.5.1 ベースプレートに支柱を立てる。

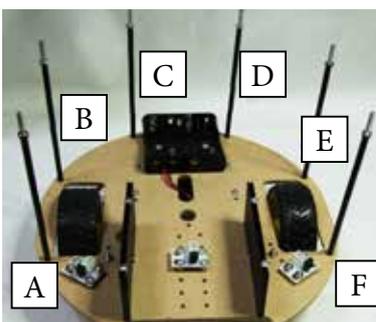
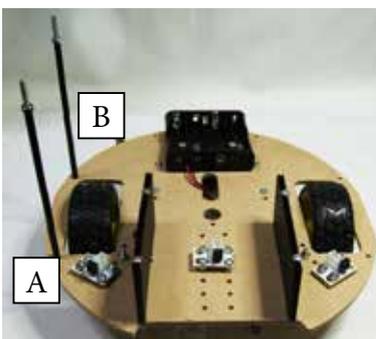
部品名	サイズなど	使用数
 長ネジ	M3x120mm	6本
 黒色スペーサ	M3x25mm	24本
 ナット	M3	6個

1. 図の位置6箇所支柱を立てます。まずは、Aの位置の支柱を立て、ナットを締めて仮固定して、次にBの位置の支柱を立てる、次、C、D～Fという順序で1本ごとに立てていきます。

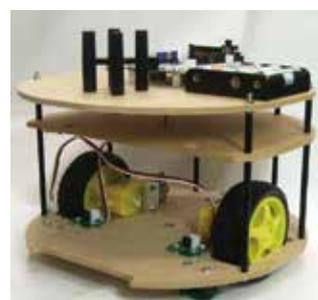


2. 支柱の立て方:

- イ) ベースプレートの下からネジM3x120mmを差し込みます。
- ロ) 黒色スペーサを、支柱1本あたり4個差し込みます。
- ハ) ナットを、ネジ先端からはめ込み黒色スペーサを支柱に固定します。



《多段構成支柱について》



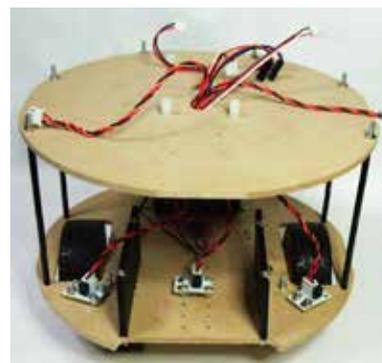
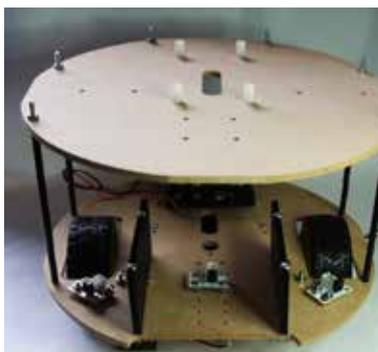
将来、機能追加などへの対応が可能なようにロボットを多段構成で設計します。その時には、スペーサの間に柵板を追加して必要機能を付加してください。上図は、3段構成にした時の画像で、3段目・4段目スペーサの間に柵板追加した画像です。また、画像のようにモータ取付面を表裏逆にして組み立てると、低床タイプのロボットになります。

支柱組立図



9.5.2 トッププレートを取り付ける。

- ・トッププレートを取り付けます。
- ・ロボット各部品の配線ケーブルをトッププレート中央の穴から、上部へ取り出します。



9.5.3 コントローラへの配線接続

部品名	接続ポート
ボールセンサ左 変調赤外線センサ	<u>A0</u>
ボールセンサ中央 変調赤外線センサ	<u>A1</u>
ボールセンサ右 変調赤外線センサ	<u>A2</u>
フロアセンサ 左 赤外線反射センサ	<u>A4</u>
フロアセンサ 右 赤外線反射センサ	<u>A5</u>
モータ 左	<u>M1</u>
モータ 右	<u>M2</u>
電池ボックス	<u>V1</u>

servo

M4

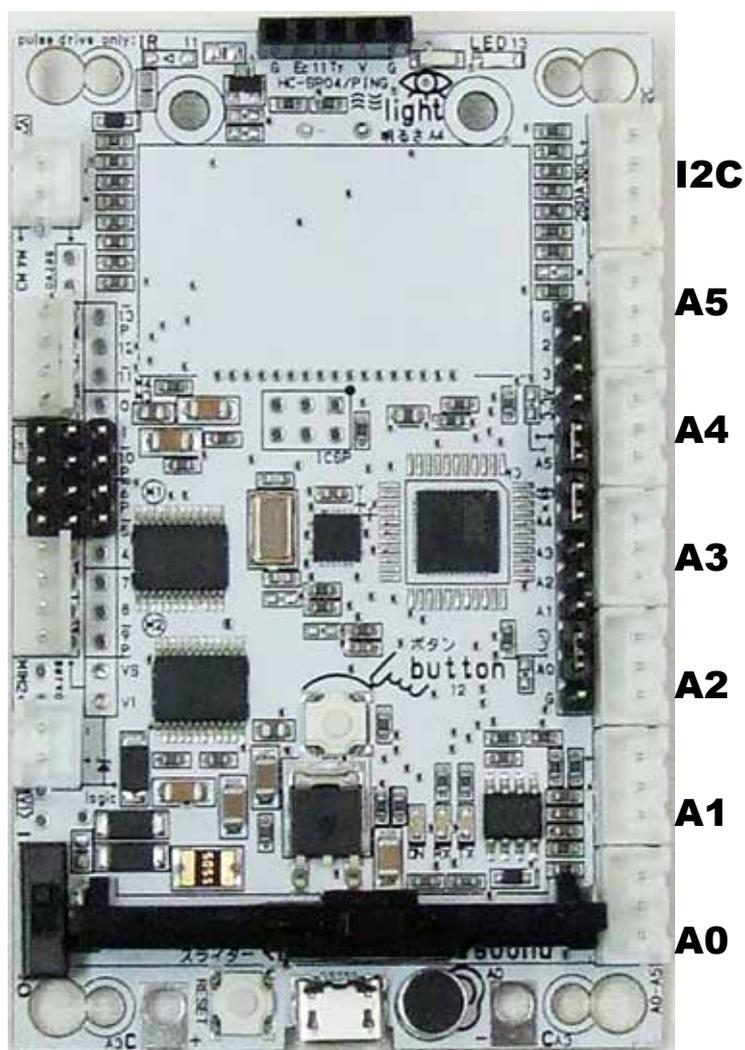
M3

M1

M2

V1

SW



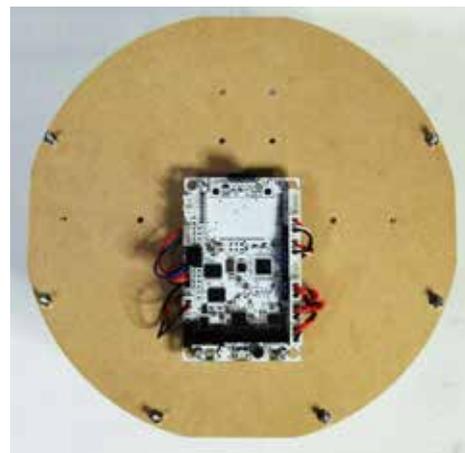
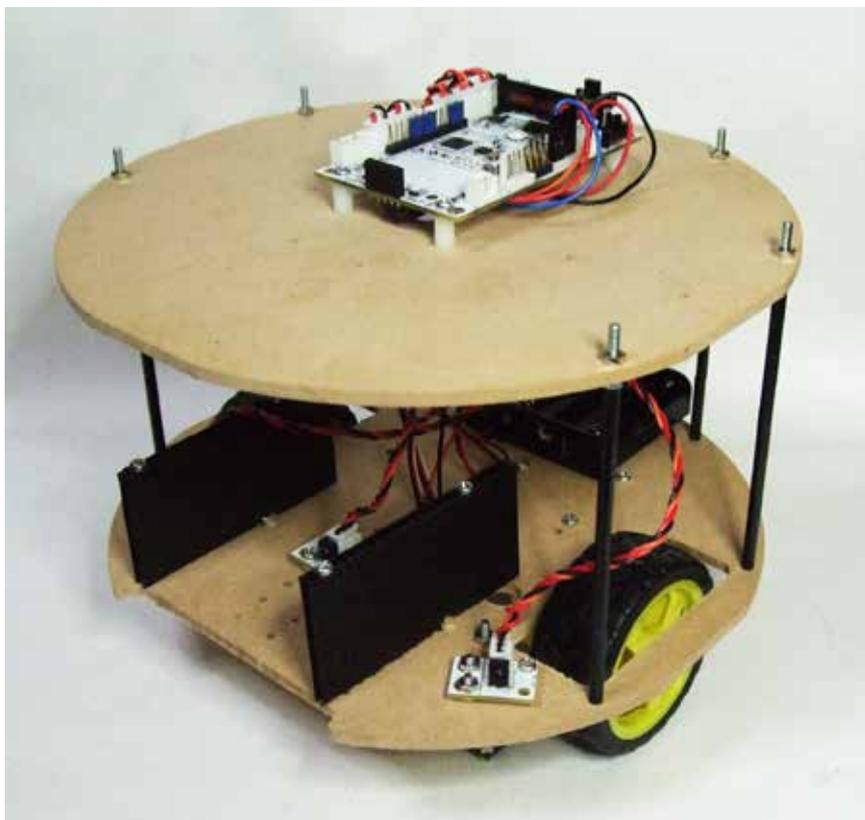
2. 接続は、コネクタ部分を持って抜き差しします。



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。

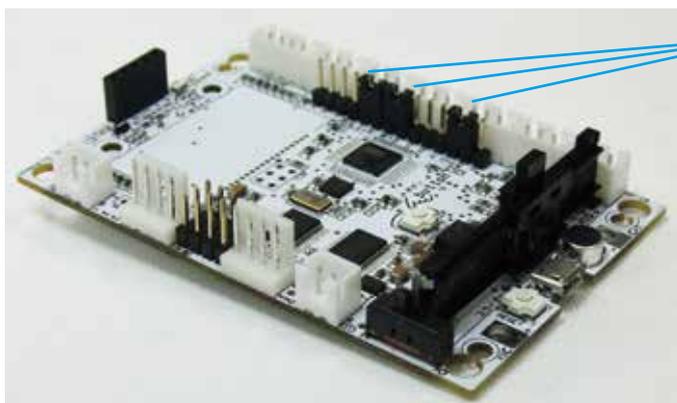
9.5.4 コントローラ取付

1). コントローラボードを、M3x6mmネジ4本で取り付けます。



👉 完成直前図

2). 基板外部センサ利用のため、回路設定用ショートピンを引き抜いて外し、回路を外部接続設定で使用します。



👉 A0,A4,A5のショートピン部分が取り外し(抜き差し)できます。説明用に青色をしましたが、製品は黒色ショートピンを使っています。

・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。

・基板搭載の音センサ(A0)、明るさセンサ(A4)、スライダ(A5)を使用する場合は、ショートピンが必要ですので、保管しておいてください。

9-6. ロボットの動作確認をする

9.6.1. コントローラボードの電源を入れます。

9.6.2. 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。
(Windows の場合、デバイスドライバインストール時に使用設定した USB 接続端子に接続します。他の端子では誤動作します)
2. 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。
・マイコンコンピュータ ▶ スタート ▶ 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。
3. (COM と LPT) COM] に認識出現している STEM Du RDC-102(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。

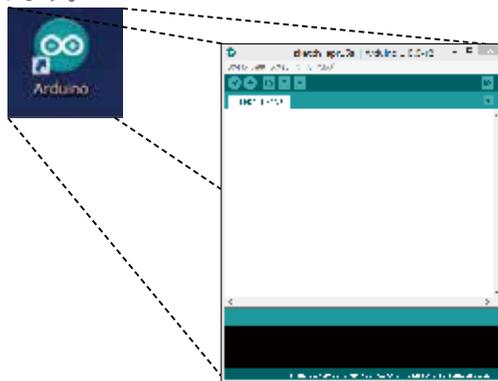


プログラム転送処理時に COM 番号が必要となります。

※プログラムが書き込めない場合は、COM 番号を確認してください。

9.6.3. プログラム開発環境の起動

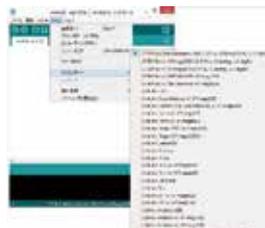
1. デスクトップに作成したショートカット arduino.exe をダブルクリックして arduino を起動します。



起動中の arduino 画面

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。

[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。



指定を間違えると、マイコンボードが誤動作します。

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

[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。

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

9-6. 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.

* My computer Start ▶ A 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.

3. 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。

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



4. 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。

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



5. ArduBlock がスタートします。

5. ArduBlock starts.



・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が始まる前に、RDC-103 コントローラの RESET スイッチを「ダブルクリック」してください。

↑ RESET

このタイミングです。

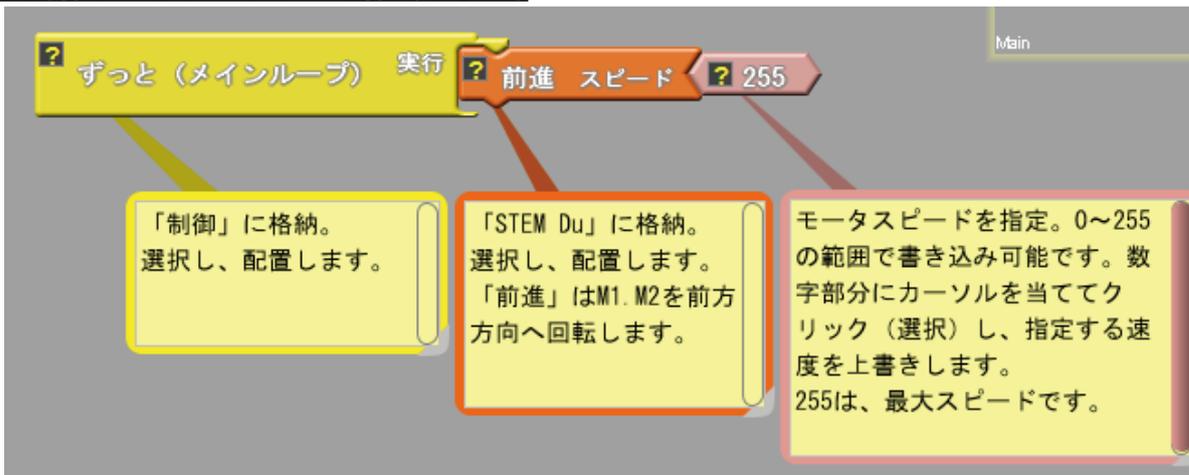
プログラムエラー

- ・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。

9.6.4. 動作テストプログラムを作成します。

[4]. A movement test program is made.



- 上記の図を参考にしながら、前進するだけの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。

1. You make the program to which you just move while consulting a figure above-mentioned.
 * 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.



- 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。
- 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。

2. "It's uploaded to Arduino.", it's done.
 3. Arduino - IDE compile, an executable file-ized 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.



****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、台座の上に置く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。**

**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.

エラーメッセージ Couldn't find a Leonardo on the selected port 出現時 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」 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が始まる前に、RDC-103 コントローラの RESET スイッチを「ダブルクリック」してください。

***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.

When upload doesn't work, you double-click the RESET switch of RDC-103 controller., please.

[Error Message]
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.

9.6.5. 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

1. 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。
2. 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
ギアードモータ 左側	Gear motor Left side	M1
ギアードモータ 右側	Gear motor Right side	M2
電池ケース (単 3 × 4)	Battery housing (AA battery x 4)	V1

3. [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。
 4. 電池ケースに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。



[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。・配線の間違いを含めて、最初の重要な確認ですから、注意して観察しましょう。

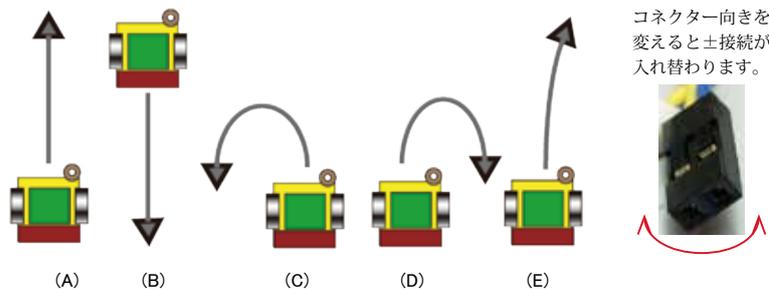


Fig.9.4.5 動作点検 Movement check

(2). 動きに応じた対策

- (A) の動き： 正しく動いています。
- (B) の動き： M1,M2 配線の極性 (プラス・マイナス) 接続間違いです。
⇒ 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。
- (C) の動き： M1 モータ配線の極性 (プラス・マイナス) 接続間違いです。
⇒ 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。
- (D) の動き： M2 モータ配線の極性 (プラス・マイナス) 接続間違いです。
⇒ 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。
- (E) の動き： 正しく動いています。左右のモータ個体差の影響で、許容範囲内です。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、正しく回転しくなくなります。

* 1. 左右モーターの個体差が影響します、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。自律型ロボットではセンサ-の情報により、常に動きが変化します、長い距離の直進性は必要ないでしょう。

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

[1]. Execution of the program forwarded to a substrate

- (1). Robot movement preliminary confirmation of preparations
1. A production is moved and it's checked to confirm the state of the connection of the robot.
 2. It's confirmed whether an input/output device is connected to the following setting street.
 3. It's confirmed that [circuit power supply switch (SW)] becomes off.
 4. 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.

(2). Measure according to the movement

Movement of (A): It's moving right.

Movement of (B): A polarity (plus minus) connection mistake of a motor wiring terminal.

⇒ Please replace a+connection of the left side motor and connect again.

Movement of (C): A polarity (plus minus) connection mistake of a motor wiring terminal.

⇒ Please replace a+connection of the right side motor and connect again.

Movement of (D): It's moving right. It's influence of the motor individual difference in the left and right and is in the latitude.

* 1. Is the individual difference in the motors of left and right influential and is it that it's very difficult even to advance where straight perfectly?

9.6.6. ロボットとプログラム

1). ロボットが白線により止まる方法

1. ロボットがどのような方法で白線を検知し、停止行動をしようとして動いているのか・・・そのアルゴリズムを理解しておきましょう。
2. ロボットに取り付けた赤外線センサには、発光用LEDと受光用モジュールがあり、左図のように赤外線を発射して、周囲からの反射光の強さを計測します。
3. 赤外線センサは、フォトIC（光起電力素子）を使用した受光センサです。光の波長の中でも赤外線周辺の帯域をもっともよく検出します。
4. 光の量で出力電圧が変化しますから、光源からの距離を計れば「距離センサ」として、色の反射率を計れば床の図形などの「読み取りセンサ」として使うことができます。より単純に、デジタル入力的に赤外線の「ある・なし」の判定に使うこともできます。
5. このように赤外線センサは届いた光を電気信号に変換してロボット本体へ情報を送ります。
6. ロボットに搭載しているコントローラボードでは、皆さんが命令したプログラム分岐条件に従い、反射光の強さがどのくらい以上であればモータを回転し、以下であれば、モータを停止するなどの動きをします。
7. 光の性質として、白い部分からは強い反射光があり、黒い部分は光を吸収して反射光は弱くなります。動作実験には、白色の床の床に描かれた黒色の線などを使用する方が、反応が大きく変化しますので、計測がしやすくなります。
8. 反射光の強さの違いによるアナログ赤外線センサの出力の変化を計算して、白なのか、黒なのかを判断し、動くことが必要で、この判断基準になる値（分岐条件）を、「閾（しきい）値」と呼びます。
9. 右の図は、プログラムを表しているフローチャートです。このように整理すると、どのような動きになるかが分かりやすくなります。
10. モータのスピードも大きな要素で、速すぎると、停止線を越えてしまい衝突するロボットになります。速すぎず、遅くない、スムーズに停止するロボットづくりに挑戦します。

白線に反応するのであれば、黒線にも反応しそうですね。

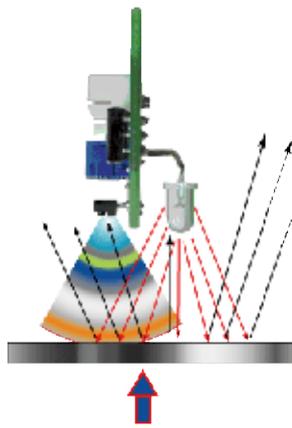


Fig.9.6.1 IR センサ原理図

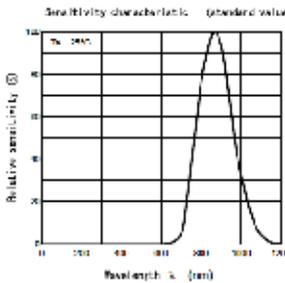
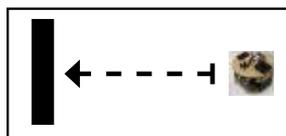
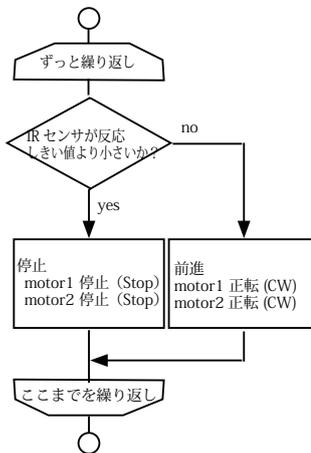
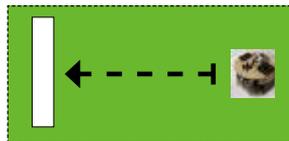


Fig.9.6.2 IR センサ特性図

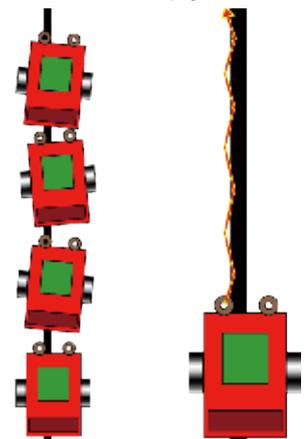


2). 白線をよける動きができます。

1. 白線をとらえて、停止したり、動いたりできるということは、うまく組み合わせていけば、競技場に設定された白線枠内で動作する自律型ロボットを制作できそうです。
2. 白線を正確に認識できる調整ができること、白線を認識したときに進行方向を変更できるかということが、大切な要素になります。

3). ライントレースをすることができます。

1. 緑色と白色、色による光の反射率を利用して、センサ反射信号の違いを見つけることが出来るようになりますが、白色と黒色の色違いでも、取り出せるセンサ信号に違いがあることとなります。
2. 白い大きなシートに、黒いテープでラインを引いて、その上をたどりながら動くロボットを作ることができます。



3. 黒い線で大きな輪を作り、その一周を走るロボットに挑戦するのはいかがでしょうか。
4. 大きな黒い線でできたトラックを、うまく早いスピードで駆け抜けるロボット作りに挑戦してみてください。

4). まず第一歩です、「前進中、センサ1個の情報で、白線で停止するロボット」を作ります。

1. グリーンカーペット切れ端に白色のテープを貼り、実験場を準備します。
2. 次頁のデータ計測を行い、緑色/白色に反応したセンサ信号値を測ります。
3. プログラムの「しきい値」を調整して、白線に差し掛かったら停止するロボットを作ります。

9.6.7. 白線で停止/旋回するプログラムを作成

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



(1) 上図を参考に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。赤外線反射センサ RDI-211 を使用して床の明暗に反応して動くプログラム例です。白いラインがあったら旋回してフィールドに戻ります。

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。

- ・赤外線反射センサ RDI-211 A4/A5 変数 ir1, ir2
- ・モータ RDO502 M1/M2

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。

※作成したプログラムは、「名前をつけて保存」します。

作成完了後、「Arduinoへアップロード」して、コントローラへ書き込みます。

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。

1. データを調べる時は、USB ケーブルを接続します。(Windows の場合、デバイスドライバインストール時に使用設定したUSB 接続端子に接続します。他の端子では誤動作します)

2. Arduino-IDE[ツール]で[マイコンボード]、[シリアルポート]を確認。

[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。



指定を間違えると、マイコンボードが誤動作します。

(1). Please make a program by making reference to a figure. The program example which reacts to the light and shade of the floor color using RDI-211 of a infrared reflectiveness sensor and moves.

Use connector : A4 / A5/ M1/ M2
Variable ir1, ir2

An output value of RDI-211 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.

* 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 code of [24_01_Base_Floor_sample] is stocked in the sample folder arranged to PC/MyDocuments/Arduino/ArduBlock_Examples by the file name [24_01_base_floor_sample.abp].

[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。



指定を間違えると、通信ができなくなります。
When you make a mistake in designation, you can't communicate any more.



3. シリアルモニター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] を使ってコントローラ接続センサの値を調べることができます。

ir1: 左赤外線反射センサー (床を見るセンサー)



ir2: 右赤外線反射センサー (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。



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. シリアルモニターでセンサ値を確認しながらグリーンカーペットの上、白線の上、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. 計測したデータを、シリアルモニターから表計算ソフトなどにコピー (ctrl+C) して数値をグラフ化処理して、考察の上、中間値を「しきい値」として使います。

※ 1. USB ケーブルを抜いて、ロギングを停止してからコピーしてください。

(3). 各種パラメータ調整

- 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量が変化します。
 - 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変化するかを調べます。
 - 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。
- 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。

2. Data of a floor sensor is checked.

1. When checking a dater, a USB cable is connected.

2. Please checked Serialport, Microcomputer board list.

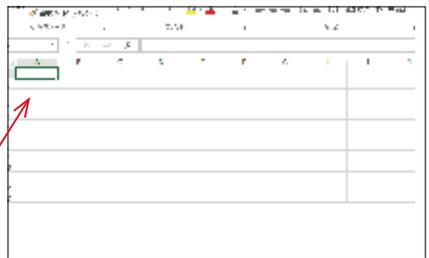
3. It's possible to check the value of the sensor using a serial monitor of ArduBlock at the state that a program of a controller is executed. (A sample program is being made in order to monitor a serial.)

4. When [serial monitor] of ArduBlock is clicked, a serial monitor screen stands up, and sensor value is indicated in real time.

5. 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)

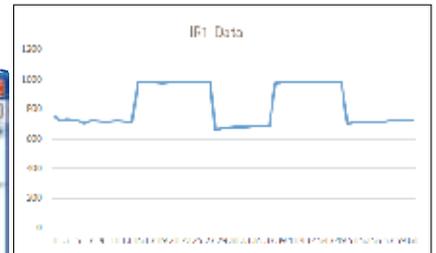
6. Intermediate value of measured data is used as "threshold value".

7. When graphing makes spreadsheet software copy from a serial monitor, and deals with a figure, it becomes easy to understand.

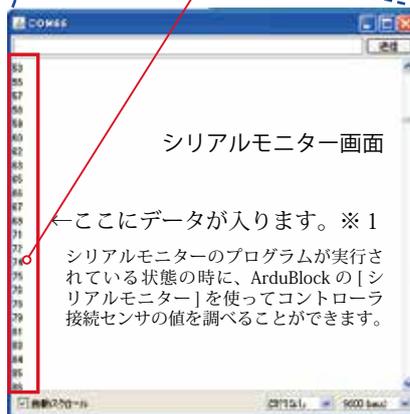


データを表計算ソフトにコピーペーストベースト時に出現するダイアログのメニューで、「スペースで区切り」を指定すると文字と数値が区別してセルに配置されます。

📊 グラフ化処理



データの傾向が把握でき、分岐条件の考察が容易になります。

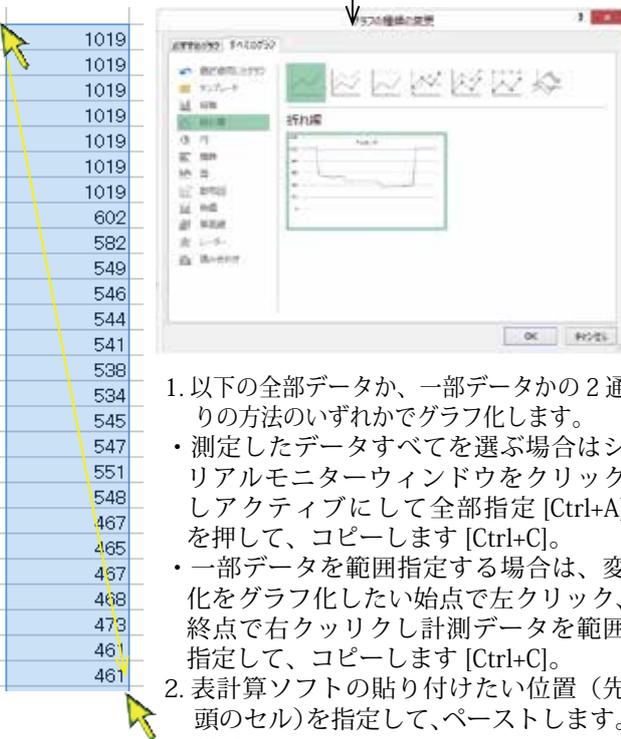
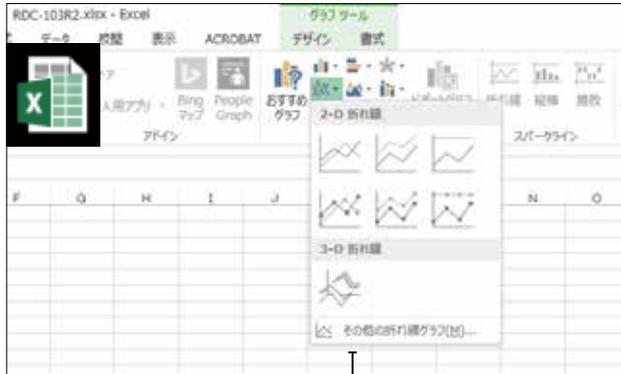


ここにデータが入ります。※ 1 シリアルモニターのプログラムが実行されている状態の時に、ArduBlock の [シリアルモニター] を使ってコントローラ接続センサの値を調べることができます。

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、USB ケーブルを外して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。

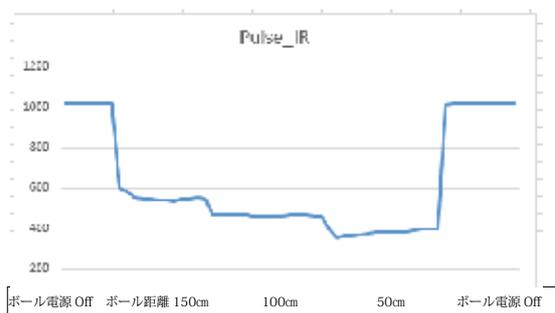
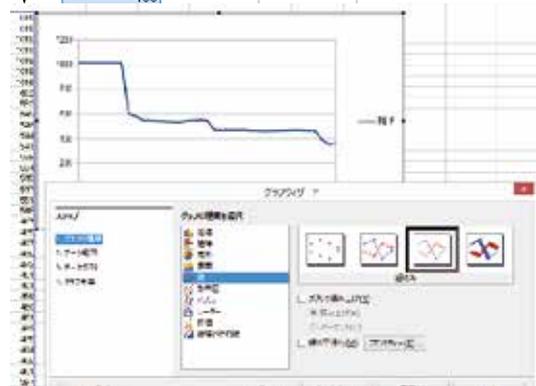
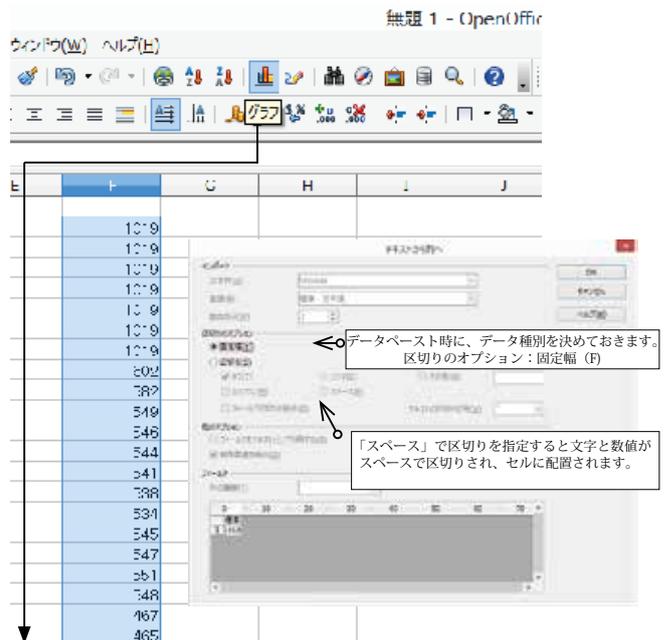
(4). 表計算ソフトの使い方

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通りの方法のいずれかでグラフ化します。
 - ・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全部指定 [Ctrl+A] を押して、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. 表計算ソフトの貼り付けたい位置 (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。

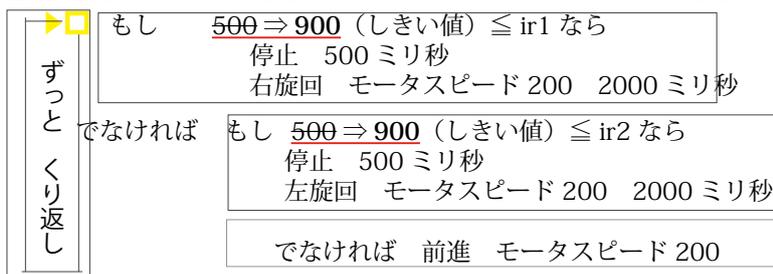
- ・グラフで見ると、距離ごとに計測数値が違うことも理解でき、プログラムの調整に役立ちそうです。
- ・ボールから離れているとき、中間距離時、近い距離など、ボールからの距離ごとにデータが違うことが分かり、プログラムを工夫して、ボールとの距離でロボットの行動を変えることが可能です。

(5). 調べたデータで、プログラムの「しきい値」を書き換えます。

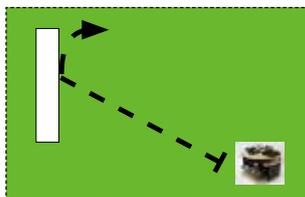


○サンプルでは、ir1, ir2の「しきい値」を500と仮に決めていますので、今回、調べた実測値に基づき「しきい値」を書き換えます。ここでは、前々頁の計測結果に基づき「しきい値」を900に書き換えてみます。

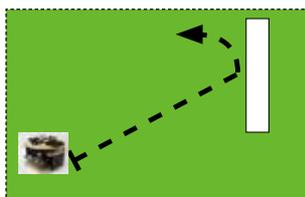
ir1, ir2の値がしきい値以上になったら停止／旋回します。



※「しきい値」に、計測し計算仮設定した900以外に、920, 940, 960, 980 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
左側から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。



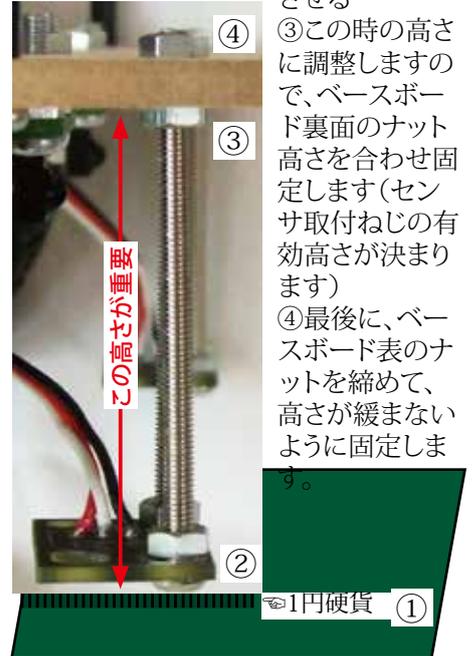
2. 次に、右側から斜めに白線に到達すると、右側の ir2 が白線に反応します。
もし
右側の反射赤外線センサが「しきい値」以上の数値になったら 0.5 秒停止して左旋回を 2 秒間 200 のスピードで行います。

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。

上手くデータが得られないとき、多くの原因は高さ調整不十分です。

- ①競技台の上に1円硬貨を置き
- ②センサ取付ねじ頭を1円硬貨に接触させる



- ③この時の高さに調整しますので、ベースボード裏面のナット高さを合わせ固定します(センサ取付ねじの有効高さが決まります)
- ④最後に、ベースボード表のナットを締めて、高さが緩まないように固定します。

フロアセンサ調整

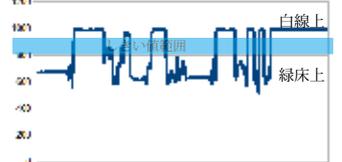
赤外線反射センサ表面から、床までの距離を 2~3mm になるように調整してください。



- 日本の 1 円硬貨が厚さ約 1.5mm、ネジ先端からセンサ表面まで約 0.5mm あります。図のように調整すると床からセンサ表面まで 2mm の隙間になります。
- データでは、焦点距離は 1mm の規格になっていますが、サッカー競技用ロボでは、フィールド床緑色とライン白色の色の違いでの反射出力データ差の違いが出るように実験にて高さ調整を決めます。

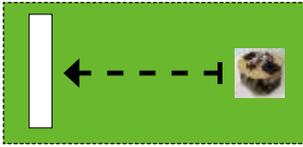


- 床までの距離が離れすぎると、データは得られなくなります、近すぎると、白と緑の差が出なくなります。
- ネジを回して、センサ高さを変更して、シリアルモニターでデータを取得し、上図のような差が得られるようにセンサ高さ进行调整をします。
- 得られたデータに基づき、「しきい値」を考察し、プログラムに反映します。



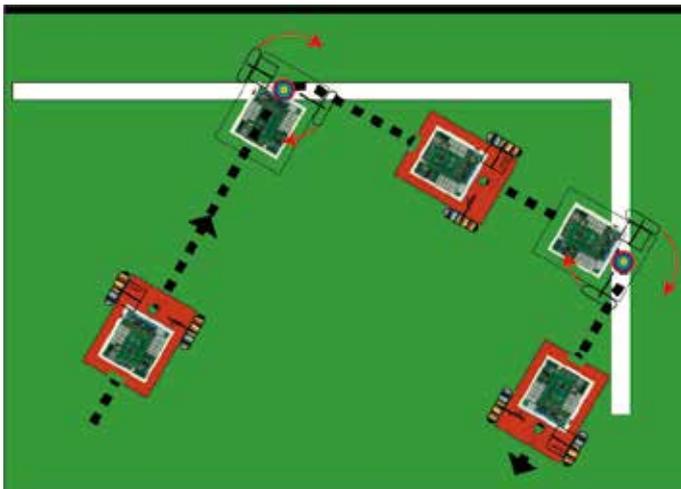
(6). プログラムを変更し、動きを変えてみます。

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。

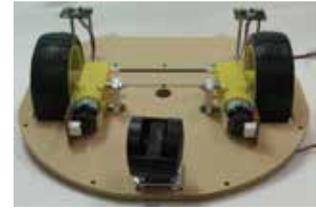


2. サンプルプログラムの巡回する時間を変更すると、180度回転するようになります。2000ミリ秒だったら回転角度はいくらだったでしょう？変更時間は何秒に設定したら、180度ピタッと回転してくれるようになりましたか？
もし $900 \text{ (しきい値)} \leq ir1$ なら
停止 500ミリ秒
右巡回 モータスピード 200 2000ミリ秒

でなければ もし $900 \text{ (しきい値)} \leq ir2$ なら
停止 500ミリ秒
左巡回 モータスピード 200 2000ミリ秒
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。



高さ調整は、後部キャスターヘナットを入れることでできます。ナット1枚0.8mm



(7). プログラムをしてみる。

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…→「左巡回」「右巡回」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。

※ Arduino 画面に、直接、C 言語を記入する時は、「半角英字」と「半角数字」のみで行ってください。「スペース」は人間に見えませんが、PC では、見えていますので、エラーの原因として一番多い現象です。

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。

9.6.8. パルス変調赤外線ボールを探索するプログラムを作成します。

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



★ボールセンサ3個のはたらき
 \ (◎o◎) /!

左センサ、正面センサ、右センサの役割図のイメージでプログラムして、ボール検知方向へロボットを向かわせます。

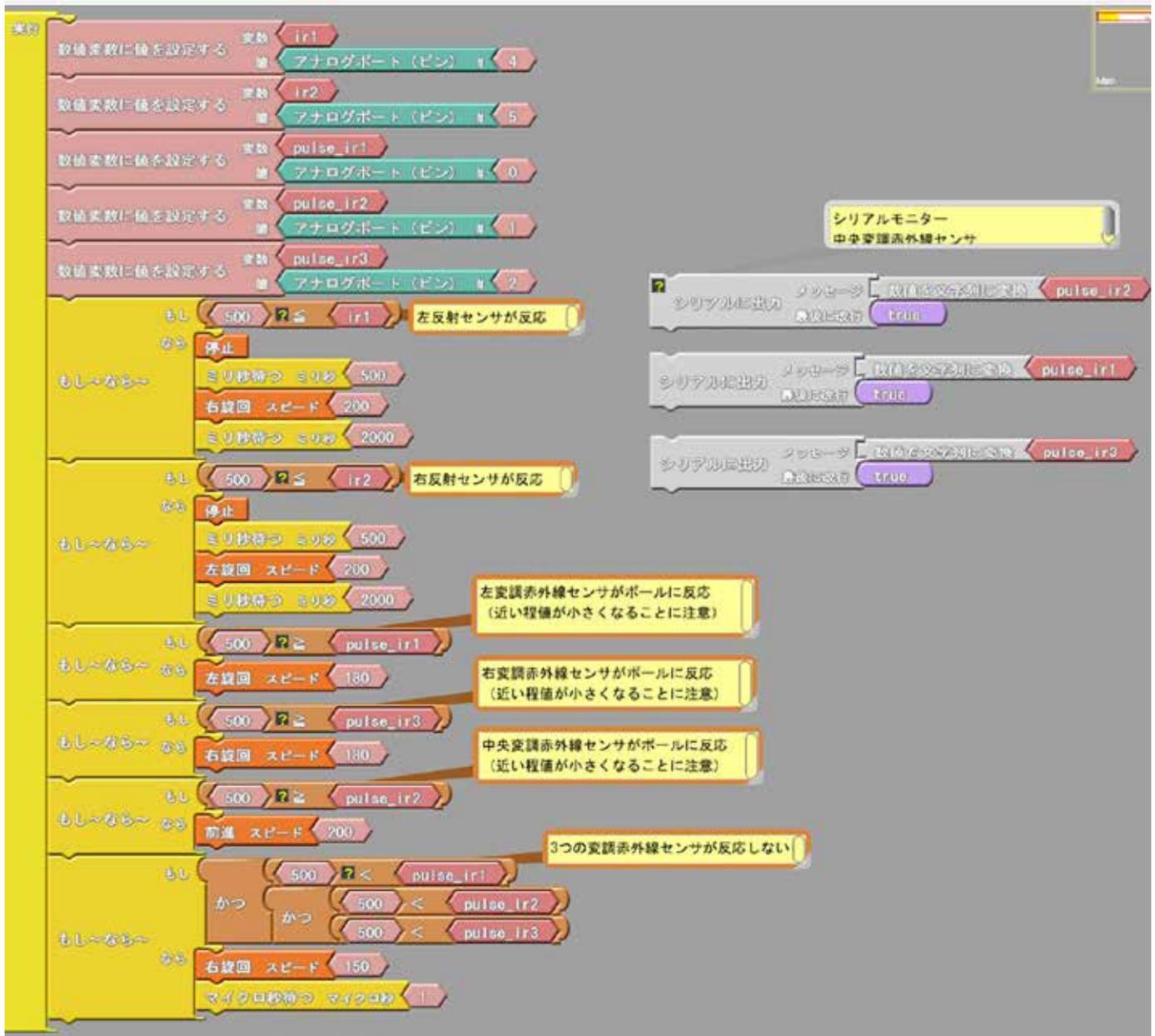
ボール検知範囲とボール捕捉行動の関係図
 中央センサの奥行きで、センサ指向性の調整ができます。

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/ArduBlock Examples の中に配置したサンプルを開くと確認できます。

9.6.9. 白線を回避しボール探査するプログラムを作成。



上図を参考に、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。
ベースロボサンプルプログラムです。

白いラインがあったら旋回してフィールドに戻ります。旋回しながらボールを探索し、左前方にボール発見時は左旋回、右前方にボール発見時は右旋回をし、前方にボールを発見したら前進します。
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。

- ・赤外線反射センサ(フロアセンサ) RDI-211 A4/A5 変数 ir1, ir2
- ・変調赤外線センサ(ボールセンサ) RDI-203JR A0/A1/A3 変数 pulse_ir1, pulse_ir2, pulse_ir3
- ・モータ RDO-502 M1/M2

「しきい値」は、前項までに、ボールを探索、ラインを認識、の実験をして、成功したときの「しきい値」を使用します。
ボールの距離によってセンサの値が変わります。

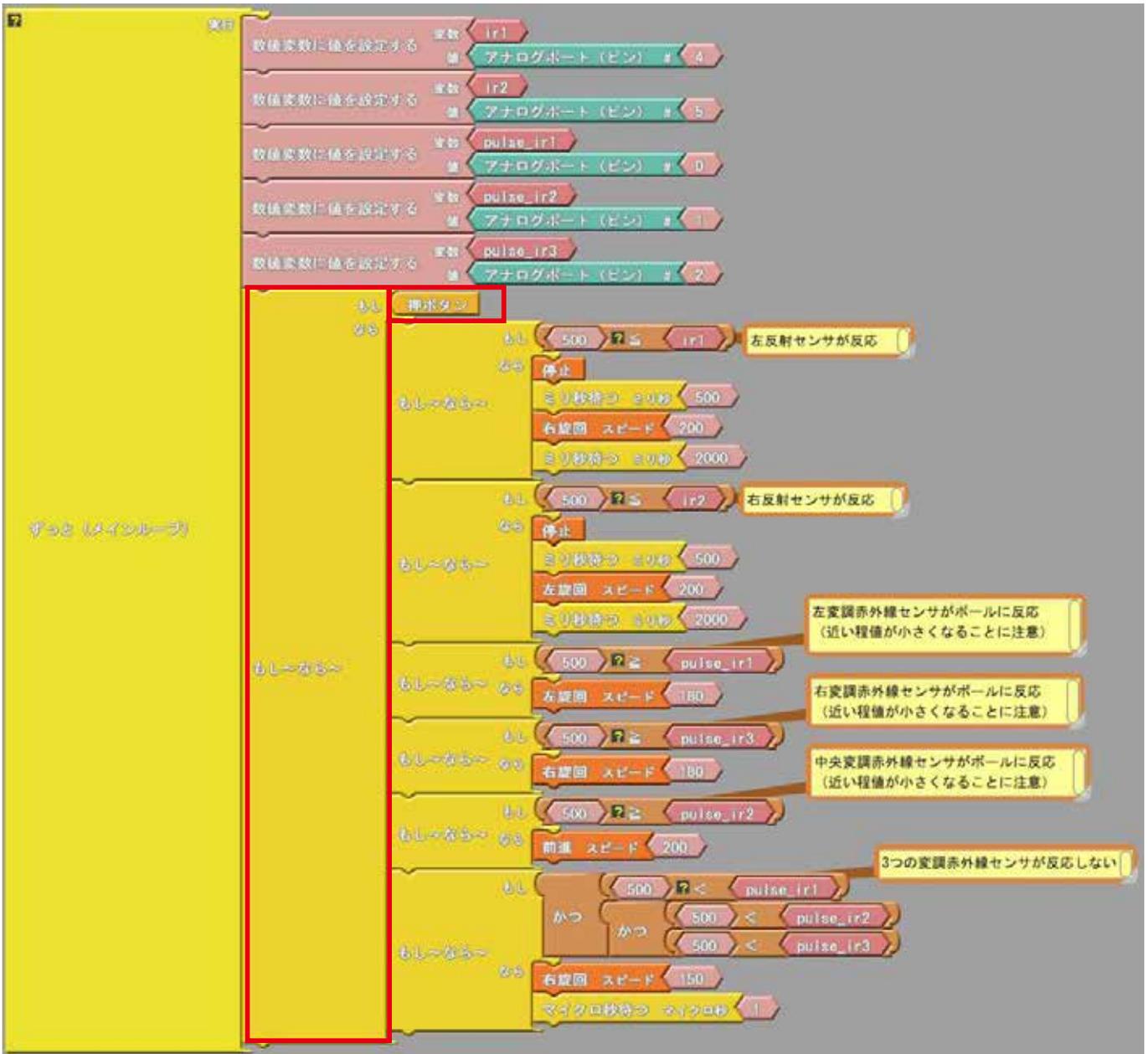
遠距離からのボール探索と、近づいたときの動作はサンプルのしきい値調整だけでは両立できないでしょう。
条件分岐を追加するなど、工夫してみてください。

- ※ RDI-203JR はボールの距離に比例して出力が変化します。近いと出力が低くなりますので注意してください。
- ※ 作成したプログラムは、「名前をつけて保存」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【23_03_base_floor_ball_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。

9.6.10. 白線を回避しボール探査するプログラムを**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、前ページのプログラムを押ボタンスタートにしたプログラム。

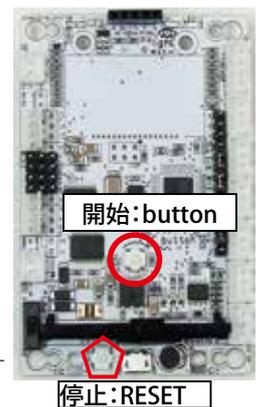


前ページプログラムに追加したアイコン



- ・「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。

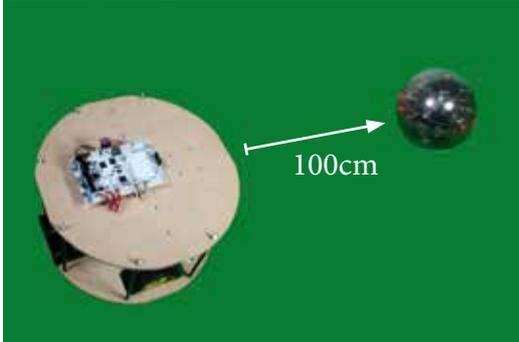
実験:前進を押ボタンスタートさせるプログラムです。



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【23_04_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。

9.6.11. プログラム調整

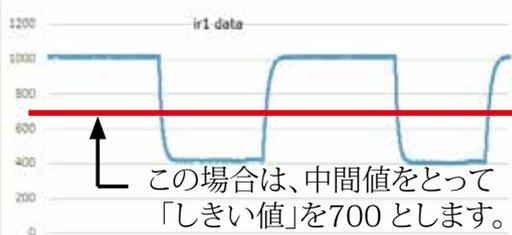
- ①. ボール探査用変調赤外線センサ「しきい値」は、シリアルモニタを使って左/中央/右ボールセンサの値をそれぞれに確認して設定します。
 - ・競技状態を想定したボールとの間隔を作り左ボールセンサ-pulse_ir1のデータ確認を行います。



- ・ボールとの距離は100cm前後で計測します。(競技場コートサイズの半分くらい)
- ・カーペット材質の床で計測します。(床の反射影響を競技場に近づけるためです)



↑ボールとの距離80cm、ボールの電源をON/OFFした測定データ



- ②. pulse_ir1 の設定後に、「シリアルに出力して改行」のセンサをpulse_ir2 へ変更してプログラムのアップロードを実施し、マイコンボードへ書き直しを行い、シリアルモニタで中央ボールセンサpulse_ir2の値を確認して設定します。



「pulse_ir2」へ変更し、アップロードをして、マイコンボードを書き直します。

- ③. 右ボールセンサpulse_ir3も、同じく設定を行います。

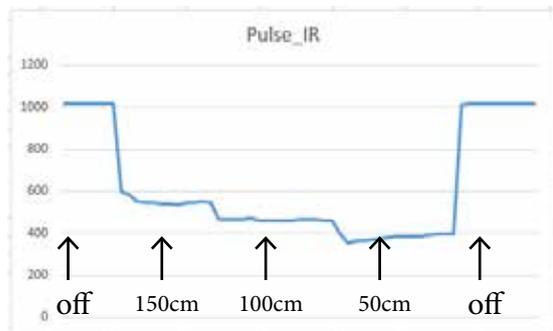


□床白線用センサir1,ir2 の「しきい値」は、前項で調べた「しきい値」を設定します。

※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【23_03_base_floor_ball_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。

◆ロボカップジュニアサッカー公式赤外線発光ボールの特徴

公式赤外線ボールには定常波/パルス波の切替が設けてあり、競技会によって赤外線を切り替えるルールになっています。通常使用されるのは赤外線パルス波で、ボールの距離によってセンサの値が変わるのが特徴です。(競技会開催時に主催者が発表します)

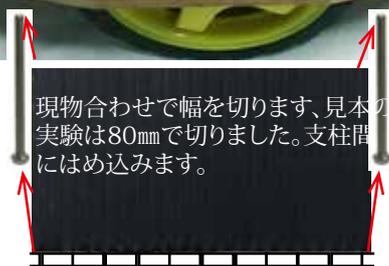


プログラムによりボール距離でロボットの動作を変化させることも可能です。遠距離からのボール探索と、近づいたときの動作はサンプルのしきい値調整だけでは両立できないでしょう。条件分岐を追加するなど、工夫してみてください。
 ※ RDI-203JR はボールの距離に比例して出力が変化します。近いと出力が低くなりますので注意してください。

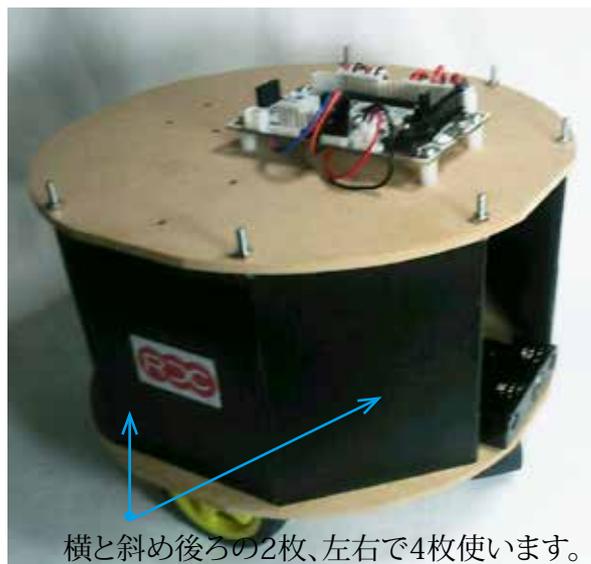
実測データ	
Pulse_ir	Data
off	1019
	1019
	1019
	1019
150cm	602
	582
	569
	544
	538
	545
100cm	548
	467
	465
	468
	473
	461
50cm	459
	397
	352
	377
	384
	398
off	397
	1010
	1018
	1019

9.6.12. ロボット仕上げ

競技会では、ロボット同士が衝突するなどが発生し、ロボットが損傷することもあります。ロボットの周囲を壁で覆うことにより、相手のロボの突起物などが、ロボット内部に突っ込むことでの損傷を防ぐ守りを施します。壁の構造:重量を増やさない為に、プラダンを利用します。



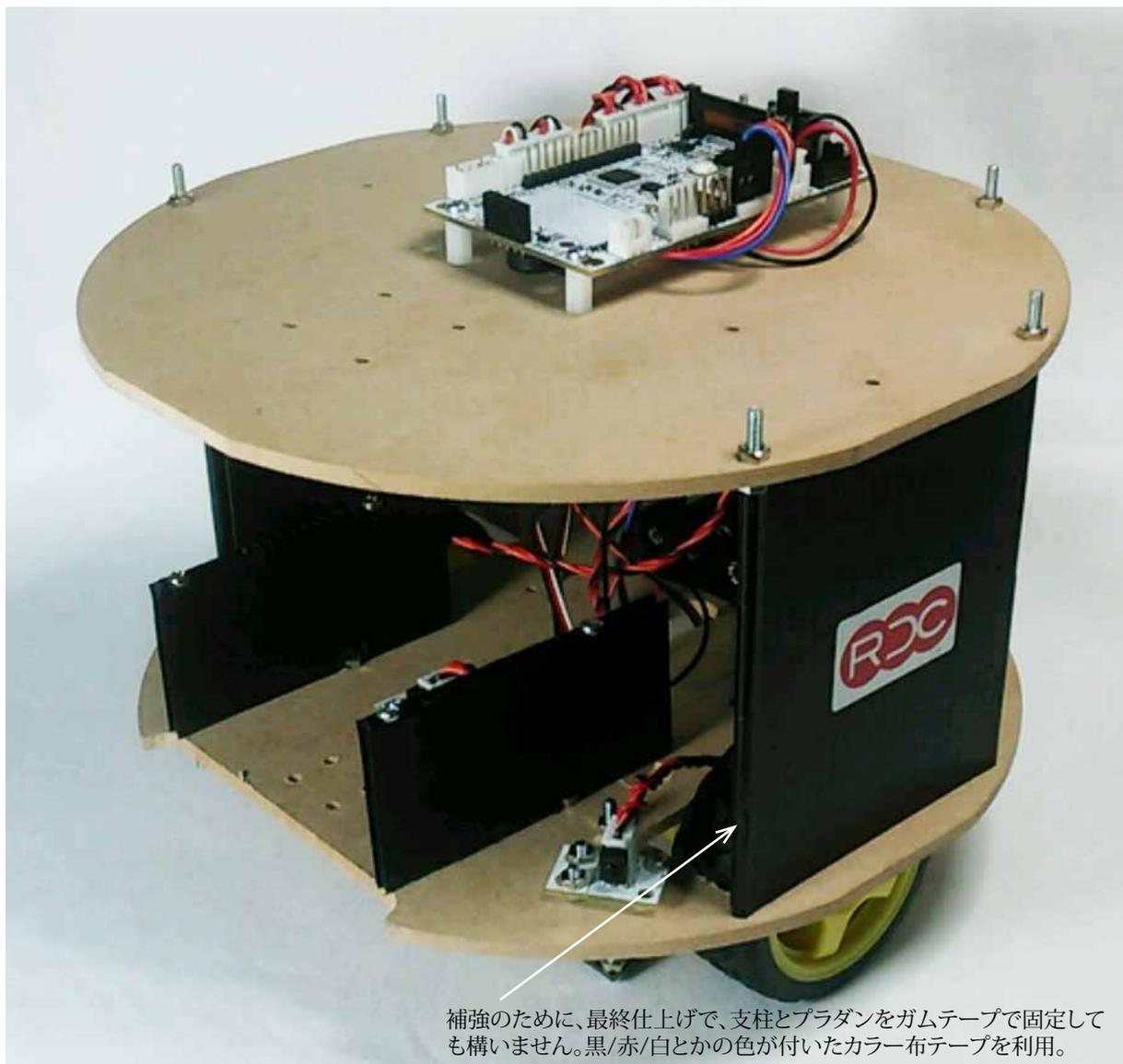
現物合わせて幅を切ります、見本の実験は80mmで切りました。支柱間にはめ込みます。



横と斜め後ろの2枚、左右で4枚使います。

壁のシールは、自分のロボットをアピールするための目印です。部品のシールを張ってもよいですが、自分らしさを演出するために、オリジナルなシールを作成して貼りましょう。

【注意】競技ルールでゴールの色(青・黄)は使えません。



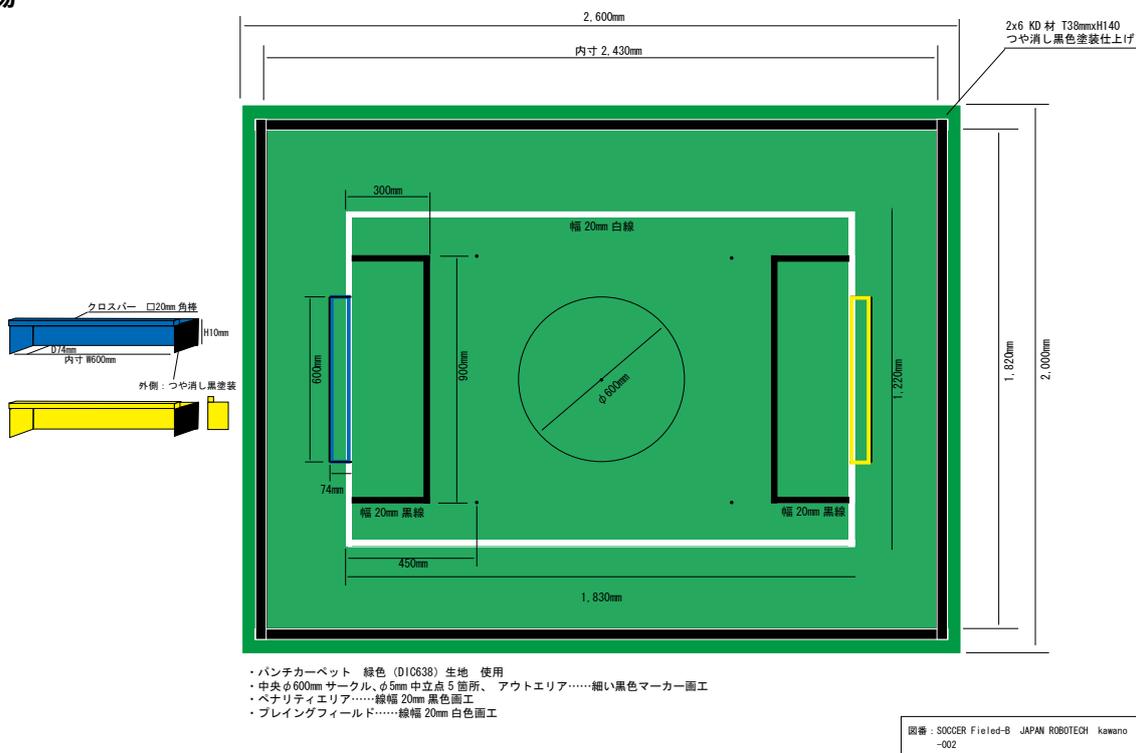
補強のために、最終仕上げで、支柱とプラダンをガムテープで固定しても構いません。黒/赤/白とかの色が付いたカラー布テープを利用。

9.6.13 競技ルール

競技会では、競技ルール、ロボット規格などがあり、出場時にはルールに適したロボットを持参する必要があります。

競技会主催者が、大会ルールを事前に発表しますので、注意をもって確認します。

・競技場



・ロボットサイズ規定・・・直径22cm円筒に納まること、高さ14cm以上



車検シリンダーが準備されています。



競技フィールド

・ロボットハンドル

競技中に審判がロボットに触れるために、ロボットには持ち上げたりするハンドルを取り付けることが要求されます。



本ロボットには、部品追加が可能なようにM3x10mmのネジをTOPに出していますので、六角高ナットを取り付けることで、設計を拡張できます。

この状態で、高さ18cm
重さは700gほどです。

重さ、使用電圧、使用部品制限等々、チャレンジするリーグでルールが異なります。出場するリーグの大会ルールを確認ください。