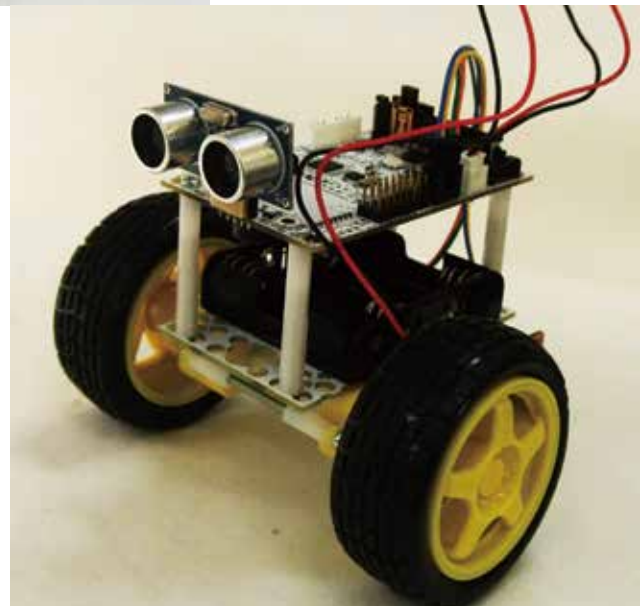
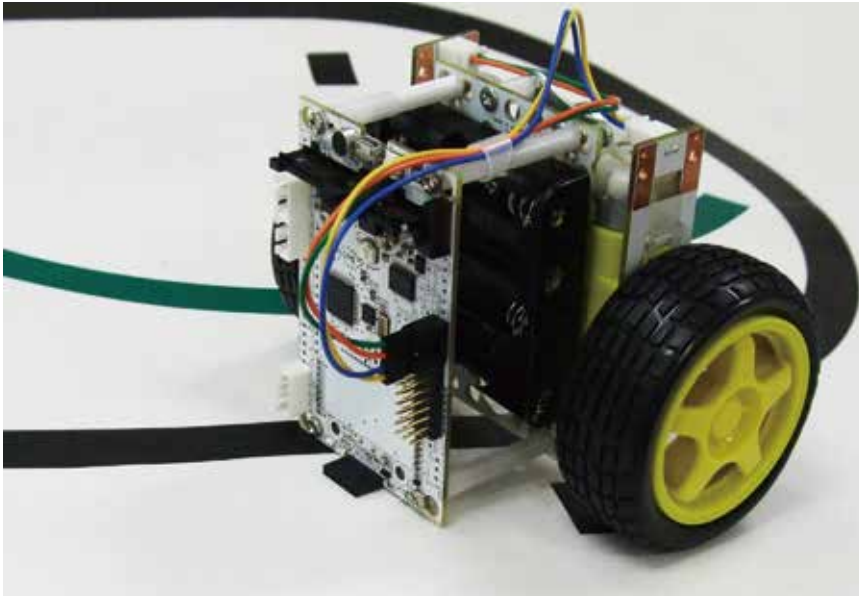


7. ライトレース/超音波障害物回避ロボ RDS-X21



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。

7-1. RoboDesigner RDS-X21 構成部品

7.1.1. パーツリスト

<p>□コントローラボード RDC-103 1枚</p>  <p>セットモデルで、付属コントローラが変わります。</p> <p>RDS-X21TYPE1 :RDC-103TYPE I RDS-X21TYPE2 :RDC-103TYPE II RDS-X21TYPE3 :RDC-103TYPE III RDS-X21TYPE3+ :RDC-103TYPE III+</p>  <p>下記の部品を取り付けています。</p> <ul style="list-style-type: none"> □樹脂製高ナット M3x10mm 4本 □ナベネジ M3x6mm 4本 	<p>□130型ギアードモータ RDO-502 2個 定格電圧 4.5V</p>   <p>組立てに必要なネジ一式が袋に入っています。</p> <ul style="list-style-type: none"> □ギアードモータ 2個 □タイヤホイール 2個 □コネクタ付モーターケーブル 2本  <p>組立てに必要なネジ一式が袋に入っています。</p> <ul style="list-style-type: none"> □マウント金具 2個 □長ネジ M3x30mm 4本 □ナット M3 4個 	<p>□スタビライザー作成部品</p>  <table border="1"> <tr> <td></td> <td>全ネジ M3 × 30mm</td> <td>1個</td> </tr> <tr> <td></td> <td>樹脂スペーサ Resin spacer M3x15mm</td> <td>2本</td> </tr> <tr> <td></td> <td>ナベネジ Screw M3x6mm</td> <td>2本</td> </tr> </table> <p>□補修用 M3 ネジ・ナットセット 1袋</p> <ul style="list-style-type: none"> □ナット M3 10個 □M3x10mm 4本 □M3x15mm 4本  <p>※各部品の取り付け時に必要なネジ・ナットは、それぞれの部品袋に入っています。</p> <p>□ケース</p> 		全ネジ M3 × 30mm	1個		樹脂スペーサ Resin spacer M3x15mm	2本		ナベネジ Screw M3x6mm	2本
	全ネジ M3 × 30mm	1個									
	樹脂スペーサ Resin spacer M3x15mm	2本									
	ナベネジ Screw M3x6mm	2本									
<p>□ユニバーサルシャーシプレート 1枚</p>  <p>取付に必要なネジが袋に入っています。</p> <ul style="list-style-type: none"> □樹脂製高ナット M3x20mm 8本 □長ネジ M3x30mm 4本 □超低頭ネジ M3x4mm 6本 	<p>□キャスタ 1セット</p>  <p>組立てに必要なネジ一式が袋に入っています。</p> <ul style="list-style-type: none"> □φ25mmキャスター 1個 □樹脂製高ナット M3x10mm 4本 □ナット M3 4個 □ナベ小ネジ M3x6mm 4本 	<p>■一緒に使う別売品案内…</p> <p>サーボモータ SG 92R</p> 									
<p>□超音波センサー HC-SR04 1個</p>  <p>差し込みピン式 (ソケットはコントローラに装備)</p>	<p>□マイクロ USB ケーブル RDP-824 1本</p> 	<p>■一緒に使う別売品案内…</p> <p>I²C コンパスセンサ RDI-5883L</p>  <p>I²C センサケーブル付</p>									
<p>□電池ボックス単 3 × 4 RDP-8093x4P 1個</p>  <p>取付に必要なネジが袋に入っています。</p> <ul style="list-style-type: none"> □皿ネジ M3x10mm 2本 □ナット M3 2個 	<p>□開発環境 CD-ROM 1枚</p> 										

7.1.2. 部品の見方、使い方

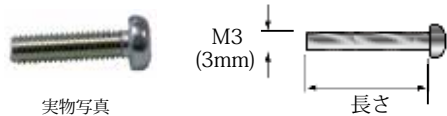
1. 部品サイズ表示

ネジやナットのサイズ表示は以下のとおりです。

■ **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。



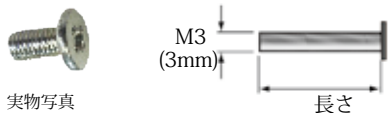
※ネジが切っただけの長さです。

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 × 10mm

「太さ（直径）3mm、長さ 10mm」という意味です。



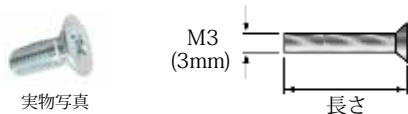
※ネジが切っただけの長さです。

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。



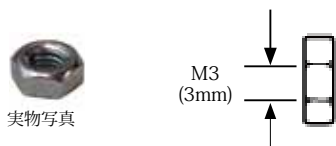
※ネジの頭からの長さです。

Nut

■ **ナット**

表記：M3

「太さ（直径）3mm のネジ用」という意味です。

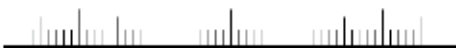


■ 共通で使う上記以外のパーツ



・ 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。
--	--

7-2. コントローラボード概要

7.2.1. RDC – 103TYPE I

- Scratch を使って 2 つのモータまで動かすことができ、常にパソコンと接続して使用します。※ 1
 - LED / 光センサー、音センサー、ジャイロ/加速度センサ、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います。
 - 外部アナログセンサー 2 個まで接続可能。
 - サーボモータ 8 個まで接続可能
 - RDC-103TYPE I には、M3、M4、小型液晶モジュールは搭載していません。
- ※ 1 ・ USB 端子からの電源供給でモータ 1 個が動作可能です。モータ 2 個を動かすには、電池を接続して電源を供給してください。

CONTROL BOARD RDC-103TYPE I OUTLINE

- It's even possible to move 2 motors using Scratch, and always it's connected with a PC and it's used. ※ 1.
 - It's equipped with LED / Light sensor-, Sound sensor-and the Acceleration/a Gyro sensor and a Slider on the board, using these, you can control variously.
 - It's even possible to connect 2 of outside analogue sensor (A1,A2).
 - It's even possible to connect 8 servomotors.
 - M3, M4 and a LCD module aren't loaded into RDC-103TYPE I.
- ※ 1) 1 motor can move by power supply from a USB terminal. Please connect a battery and supply me a power supply to move 2 motors.

仕様	The specification	DataSheet URL
マイコン / ATMEGA32U4、発信周波数 8MHz	MCU / ATMEGA32U4 Clock 8MHz	http://media.digikay.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-、モータコネクタ-、電源コネクタ-、USB コネクタ-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。

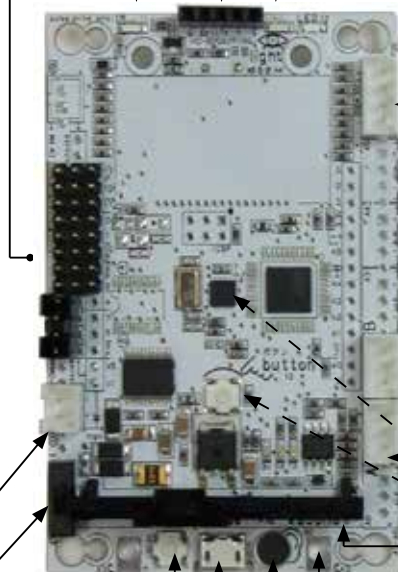
ピン番号	記号	解説
13		サーボ / 白色 LED/PWM 出力可能 R/C servo motor / White LED / PWM output
12		サーボ / ボタン R/C servo motor / Button
11		サーボ / 超音波 / 赤外線 LED R/C servo motor / UltraSonic / InfraRed
0		サーボ / ブザー / シリアル RX R/C servo motor / Buzzer / Serial RX
1		サーボ / LCD RS / シリアル TX R/C servo motor / LCD RS / Serial 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 (0.5A 程度 / 1 個)	M2 制御 M2 control
8		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-		
V1		電源コネクタ- Power Conctor
-		電源スイッチ Power Switch

RESET スイッチ

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



赤外線 LED
Infrared Sensor



明るさセンサ Light sensor

発光 LED 白色 / 受光 LED

I²C コネクタ- 3 SCL, 2 SDA

アナログ入力 Analog input

A1 (A は Analog の A) と A2 の 2 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 Sound Sensor
A1	A1	アナログ入力コネクタ- Analog input
A2	A2	アナログ入力コネクタ- Analog input
A3	A3	みの虫クリップ Signal Clip terminal
A4	A4	明るさセンサ Light Sensor
A5	A5	スライダ- Slider

接続コネクタ- 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 / ジャイロセンサ (I2C) accelerometer

ボタン button

スライダ- (可変抵抗) Slider resistance

みの虫クリップ用端子 (抵抗等を測ります)

音センサ Sound sensor

USB コネクタ- USB connector

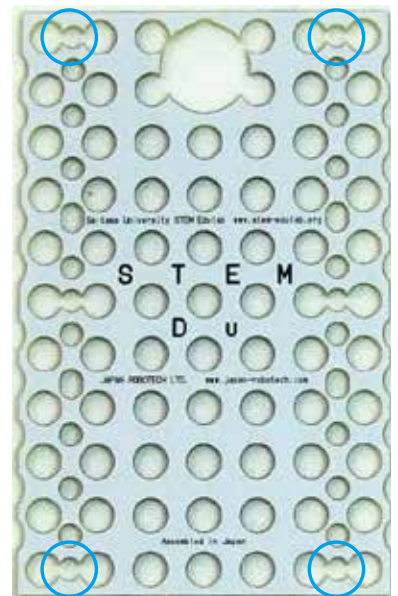
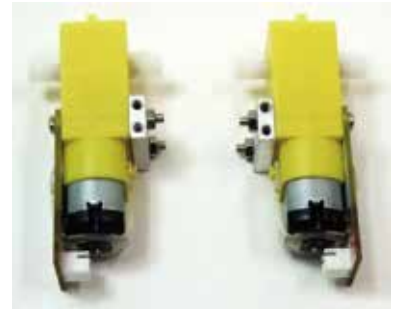
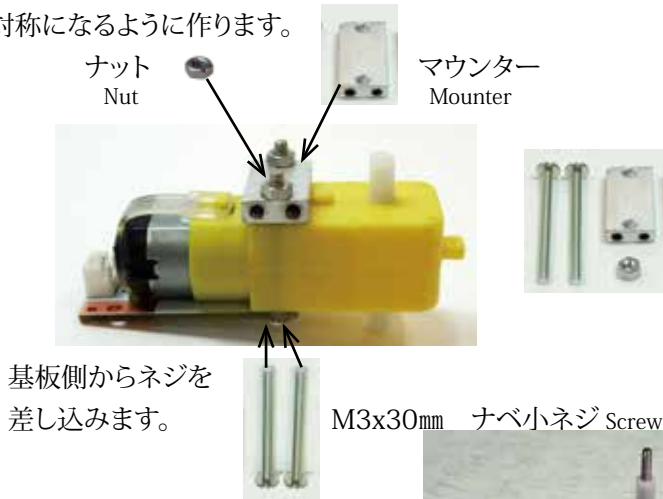
7-3. 走行台車組み立て [Assemble of a Vehicles]

7.3.1 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。A gearbox is made by symmetricalness.

ギアードモータにマウンターを取り付けます。[Assemble of a Mounter]

(検査のため取り付けし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。右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。

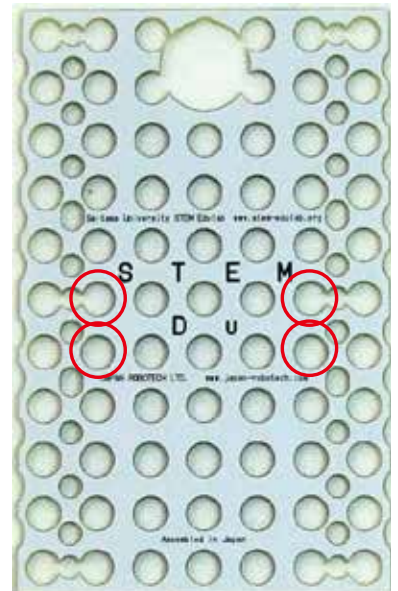
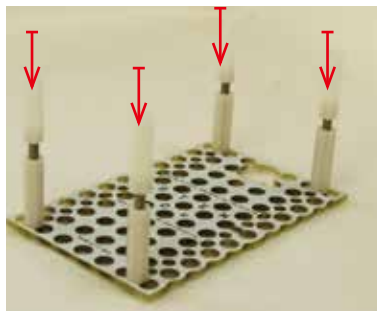
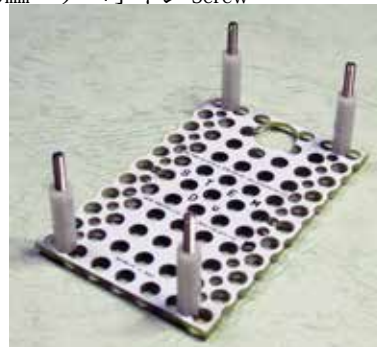


7.3.2. 支柱の組み込み [Assemble of a pillar]

1. 右図シャーププレート上の青丸位置に、支柱を組み付けます。

- ・ナベネジ30mm 4本を青丸位置に差し込み、樹脂製スペーサー20mmを取り付けます。
- ・樹脂製スペーサー20mm 4本を 2段目にはめ込みます。

	使用部品名	使用数
	支柱:樹脂製スペーサー Resin spacer M3x20mm	8本
	ナベネジ Screw M3x30mm	4個

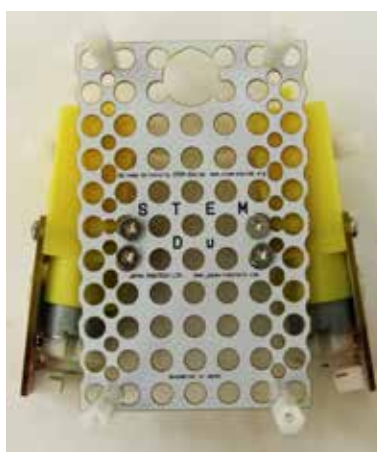


7.3.3. モータ取付 [Assemble of a motor]

1. 右図の赤丸位置に、マウンターの穴位置を合わせてモータを取り付けます。

- ・モータの取付方向にご注意ください。左右方向性あります。
- ・モータコード差し込みソケットが左右対称になるようにします。



	使用部品名	使用数
	超低頭ネジ Low Head Machine Screw M3x4mm	4本



・この段階では、ネジは位置を確認する程度にとどめ、締め付けないようにします。次段階で左右のモータを補強支持棒で合体させた後で位置決定をして締め付け固定します。

7.3.4. 後部キャスト取り付け

1. キャスターを右図の赤丸位置に取り付けます。


使用部品名	使用数
 支柱取り付けキャスター Caster 車輪径 25mm	1 個
 超低頭ネジ Low Head Machine Screw M3x4mm	2 本

・超低頭ネジは、シャーシプレートに同梱しています。
※とても小さいので見逃さないようにしてください。

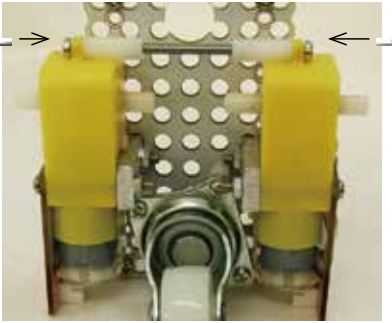
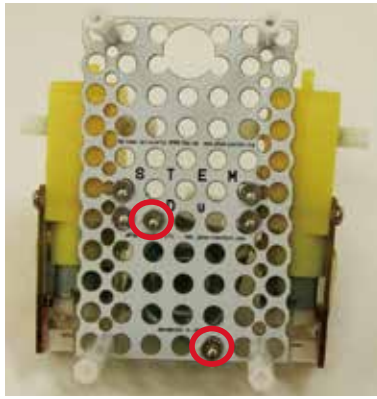
※支柱は 4 本ですが、シャーシプレートへの取付固定は 2 か所のネジで行います。

※モーター間スペースが狭く、キャスターが入りづらいですが、モーター取り付けねじを緩くしてモーター角度を広げたりしながら、取り付けを行います。

2. スタビライザーを加え、駆動部取付を安定させます。

使用部品名	使用数
 ナベ小ネジ Screw M3x6mm	2 本

準備 2 で作成するスタビライザーをモーター前部のボス間にはめ込み、両側よりナベネジ M3x6mm で取り付けます。仮止めです、ゆるく締めておきます。



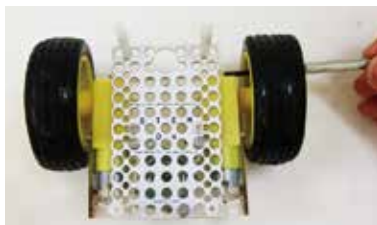
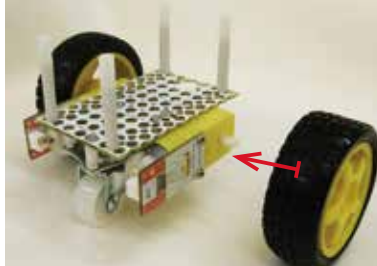
準備 1. キャスターに支柱を取付

使用部品名	使用数
 キャスター Caster 車輪径 25mm	1 個
 樹脂スペーサ Resin spacer M3x10mm	4 本
 ナット Nut	4 個
 ナベ小ネジ Screw M3x6mm	4 本



7.3.5. タイヤ取り付け

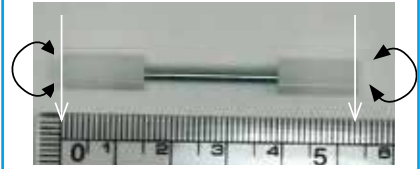
- ・車輪にタイヤを差し込みます。
- ※回転軸（シャフト）を支えてホイールを差し込みます。ギアボックス内部ギアに無理な力を加えないようにご注意ください。



準備 2 スタビライザー作成

使用部品名	使用数
 全ネジ M3 × 30mm	1 個
 樹脂スペーサ Resin spacer M3x15mm	2 本

全ネジの両側からスペーサをねじ込み仕上げ長さ 53mm に調整します。



7.3.6. タイヤ角度調整

1. 左右のタイヤが並行になるように角度を調整確認後に、マウンター取付ネジを締め付けて固定します。
2. スタビライザーの固定ネジを締め付けて固定します。



7.3.7. 電池ボックス取り付け

使用部品名	使用数
ネジを差し込んだ電池ボックス Battery housing	1 個
ナット Nut	2 個

1. シャーシプレートの図の位置に取付ます。
2. 前部の取付ねじは裏側に見えていますので、ナットドライバーでナットを取り付けます。
3. 後部の取付ねじは、キャスターと位置が同じ場所ですので、ラジオペンチなどでナットを挟み隙間から差し込みネジ先端に合わせておいて、ネジ頭をドライバーで回して、ネジ締めを行い、固定します。



準備 3. 電池ボックスにネジを取付

Attaching a screw in the battery housing.

使用部品名	使用数
 電池ボックス Battery housing 単 3 × 4 本	1 個
 皿ネジ Flat Head Screw M3x 1 0 mm	2 本




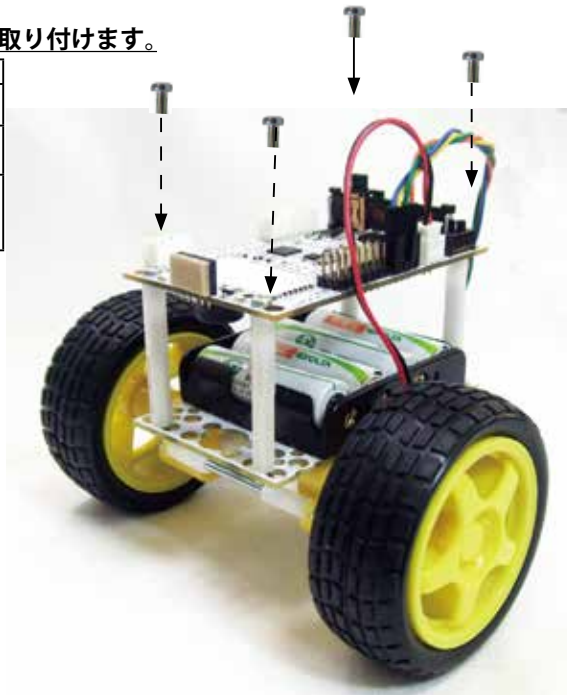
表面 裏面

ネジ取付位置

・ M3x10mm 皿ネジは、電池ボックスに同梱しています。

7.3.8. コントローラを取り付けます。

使用部品名	使用数
コントローラ Controller board	1 枚
ナベネジ Screw M3x 6 mm	4 本
	



※超音波センサ-ソケット側面に貼り付けている「すべり」テープは、床面接触時のすべり効果を出す目的とともに、本機をライントレースロボットとして使うときのセンサー面と床面との隙間調整を決めるためのテープです。取り外したりなさらないようにご注意ください。

3. ATTACHING CONTROLLER BOARD

It's used for the purpose that the "slip" tape which is being stuck to an ultrasonic sensor socket side takes out the slip effect which is at the time of floor surface contact.

The sensor page when using a this machine as a Linear tracing robot, and the tape to decide a gap adjustment with resting face.

Please be careful so as not to remove.

7-4. 超音波障害物回避ロボの組立 [Attaching Ultrasonic sensor]

7.4.1. 超音波距離センサ取付

- ・センサ基板の超音波距離センサ取り付け用ソケットの穴に、表示を合わせてセンサーピンを差し込みます。

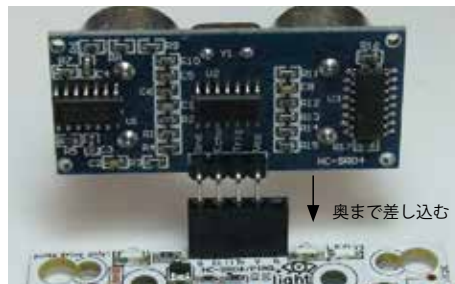
超音波 HCSR04 センサ端子表示	RDC-103 ソケット端子表示
Gnd	G
Echo	Ec11
Trig	Tr
Vcc	V
—	G

ATTACHING SENSOR

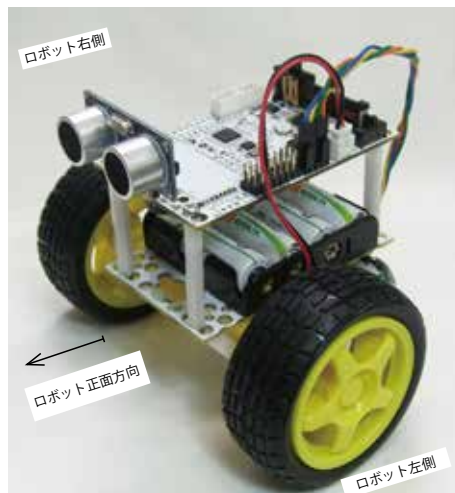
Put a sensor pin in a hole of a socket for ultrasonic range sensor installation of a sensor substrate.

Make a sensor terminal indication symbol and a controllerboard terminal indication symbol agree.

超音波センサ取り付け前



超音波センサ取り付け後



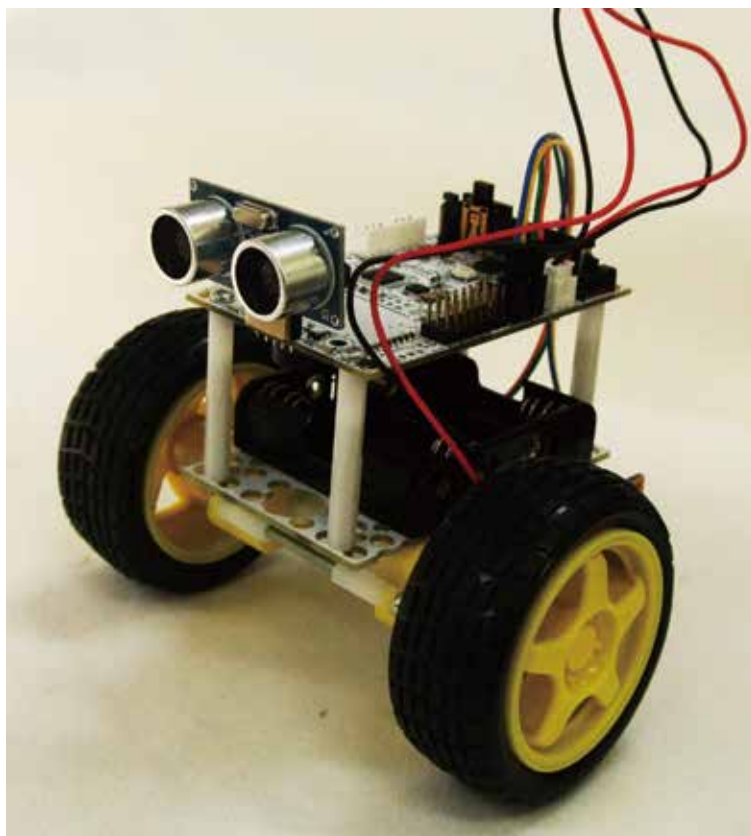
7-5. 配線

1. 下記の接続一覧表を参考に、各部品とコントローラの接続配線をします。

部 品	コントローラボード	接続
ギアードモータ 左側 Geared motor Left side	M1	モータケーブル ※1 Motor cable
ギアードモータ 右側 Geared motor Right side	M2	モータケーブル ※1 Motor cable
電池ケース (単 3 × 3) Battery housing (AA battery x 3)	V1	赤 / 黒ケーブル Red/black cable

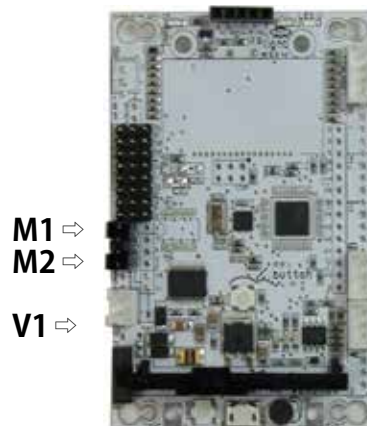
* モータには、回転方向を決める極性がありますが、モータケーブル先端には、どちら方向にでも差し込める無極性のコネクタを付けています、この段階での配線は極性を気にせず接続します。後ほど、実動テストを行うときにロボットの動きを確認して、差し込みを調整します。

2. 接続に間違いがないかどうか再確認後、電池ケース底面に記載されている極性表示に合わせて電池を実装します。
3. プログラムの準備ができるまで、電源スイッチを切っておきます。



WIRING

1. Connection wiring of each part and a controller is done by making reference to the following connection list.



*Motor has the polarity which decides the direction of rotation, but polarity-less connector which can be put in every way is being put on the cable point.

Without worrying about polarity, wiring at this stage is connected.

When doing an actual working test later, the movement of a robot is confirmed and insertion is adjusted.

2. After reconfirming whether it's without mistakes in a connection, a battery is mounted according to the polarity indication indicated on a battery housing base.

3. Power supply switch is cut until We'll be ready for a program.

組み込んだホイールを取り外したい時

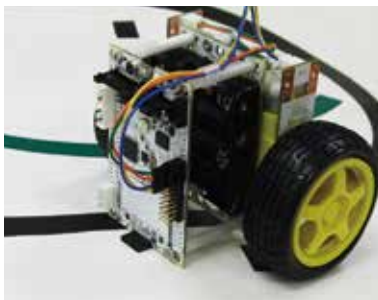


ギアボックスとホイールの間にラジオペンチの先を差し込み、ラジオペンチをゆっくりとこねてホイールを真上へ抜きます。

※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。

7-6. ライントレースロボ機体組立

1. 前項で作成した「超音波障害物回避ロボ」から、超音波センサ HC-SR04 と、後部キャスターを取り外します。
2. 説明写真のように、コントローラ先端の [明るさセンサ] を床面に向けて、ライントレースロボとして使います。明るさセンサの仕様を考慮し、床面からセンサ面までの距離を 3 mm に設定して構造計算をしています。
3. 搭載センサを利用し、構造を他の方法で組み立てられる場合は、センサ距離を 3 mm 前後になるように調整ください。

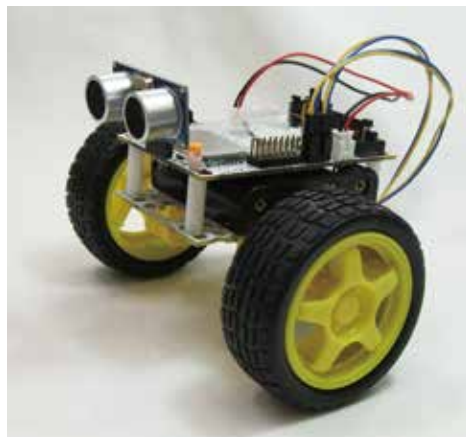
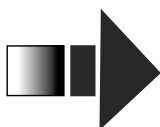
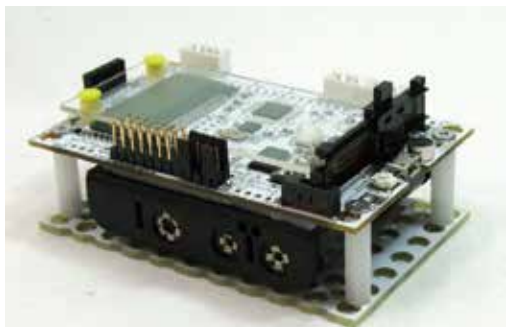


Assembly of an Linear tracing robot

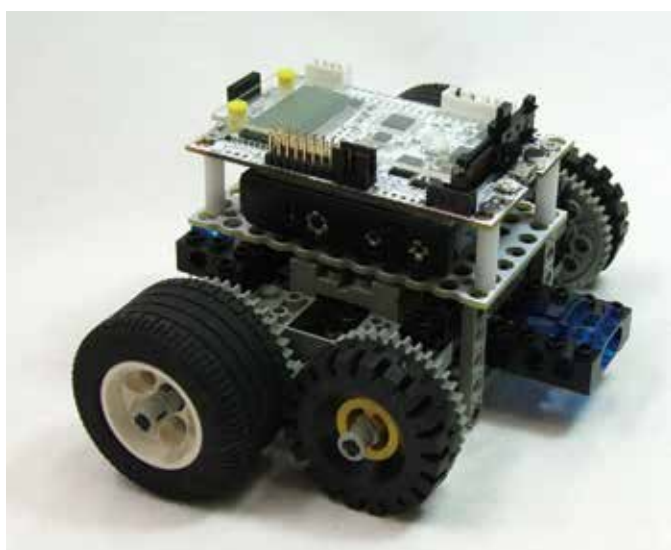
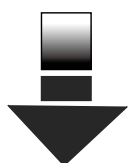
1. Ultrasonic sensor HC-SR04 and a rear caster are removed from the "ultrasonic obstacle avoidance robot" made by the preceding clause.
2. [Brightness sensor] in a controller point is used as a linear trace robot for resting face. The specification of brightness sensor is being considered, the distance from resting face to the sensor face is set as 3mm and structure calculation is being done.
3. When We can use equipped sensor and construct the structure by other ways, please adjust the sensor way as it'll be about 3 mm.

MEMO

コントローラは、組み合わせでいろんなロボットに利用できます。



樹脂スペーサ 1 段でスリムなロボット



ブロック教材と合体させることもできます。



ユニバーサルシャーシプレートは、ブロック教材との互換性を考慮して設計。走行台車にハマ込むことが可能です。

7-7. ロボットの動作確認をする

7.7.1. コントローラボードの電源スイッチを ON にして、電源を入れます。

7.7.2. 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、デバイスドライバー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。
・ Windows : マイコンピュータ ▶ 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。
4. プログラム転送処理時に COM 番号が必要となります。

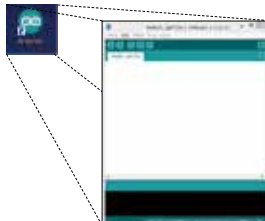


※プログラムが書き込めない場合は、必ず COM 番号を確認してください。

7.7.3. 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。

Fig.7.7.1 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。

[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。

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.

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。
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☑マークがついていることを確認してください。

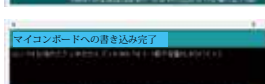
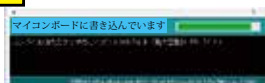
・アップロードがうまくいかない場合は、コンパイルの後、マイコンボードに書き込みが始まる前に、RDC-103 コントローラの RESET スイッチを「ダブルクリック」してください。



↑ RESET



このタイミングです。☑



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 ▶ 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].

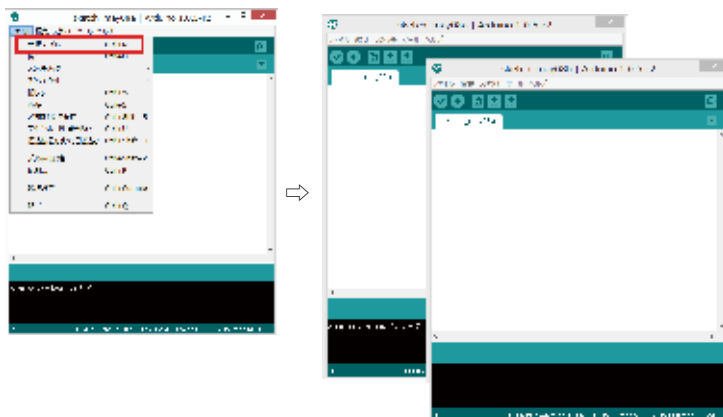
[Microcomputer board]: of Arduino-IDE [Tool] ▶ [STEM Du/RoboDesigner+ RD C-102 w/ ATmega32U4 3.3V 8MHz] is chosen by the microcomputer board list which clicks [microcomputer board] and appears, and click designation is performed. A ● mark sticks to the list left end.

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. [File] of Arduino - IDE [It's filed newly.] 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 -IDE of a new file.

5. ArduBlock がスタートします。

5. ArduBlock starts.



7.7.4. 動作テストプログラムを作成します。

A movement test program is made.



- 上記の図を参考にしながら、前進だけのプログラムを作成します。
 - 例は、「前進」ですが、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ますので、最初は、半分くらいの少し遅めが良いと思います。



- 「Arduino へアップロード」します。
- 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。」とメッセージが表示されると、書き込み完了です。

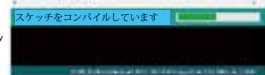


エラーメッセージ Couldn't find a Leonardo on the selected port...出現時通信エラーが発生し、マイコンボードへの書き込みが失敗しています。Arduino ▷ [ツール] ▷ 「マイコンボード」RDC-102 に●マーク、「シリアルポート」接続 COM 番号に✓マークがついていることを確認ください。

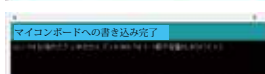
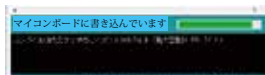
・アップロードがうまくいかない場合は、コンパイルの後、マイコンボードに書き込みが始まる前に、RDC-103 コントローラの RESET スイッチを「ダブルクリック」してください。



↑ RESET



このタイミングです。☞



1. 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.

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.

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.

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

7.7.5. はじめてロボットを動作スタート、各種動作点検

[1]. コントローラボードに転送したプログラムの実行

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

- [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。
- 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: [RESET スイッチ (SW)] を押します。

Left Motor
左モーター → M1
Right motor
右モーター → M2

Battery housing
電池ケース → V1

Power supply switch
電源スイッチ →



RESET スイッチ ↑ ↑ USB

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]. 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。

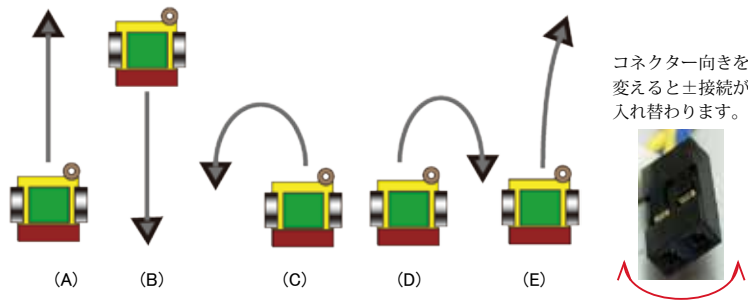


Fig.9.4.5 動作点検 Movement check

(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 ともにコネクター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ー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ー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。

*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。
- ***** エンコーダ付のモータに取り換えることで、



モータの回転数を調べて、左右が常にぴったり同じになるよう制御調整する PID 制御の方法などもあります。

モータ後部にエンコーダが付いている例



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きますので、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。

[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

- (A): It's moving right.
 (B): A polarity (plus minus) connection mistake of a motor wiring terminal.
 ⇨ Please replace a + - connection of the left side motor and connect again.
 (C): A polarity (plus minus) connection mistake of a motor wiring terminal.
 ⇨ Please replace a + - connection of the right side motor and connect again.
 (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?

- * The long distance, dispatch if I'll turn to one of them in left and right a little.
- ** For example a wheel is installed in both sides of 1 motor by the different structure, dispatch of when an experiment is made, I advance straight, but you can't turn any more freely by a program.
- *** The way to lose the individual difference in the motors as a countermeasure is also considered, but the individual difference will find a little motor from the inside of tens of thousands of of production, and it'll be the very expensive cost.
- **** The autonomous robot on which you're working this time gets sensor information and changes the constancy direction based on a program, and moves, so a straight line by the long distance won't be a problem big, so please be relieved.

When RDC-103 controller is turned on, a program will be executed immediately.

When power supply switch is turned on, movement of a robot begins it.

Even if movement is begun, put a robot in the safe location and let me begin to move, please.

7-8. ライントレースロボのプログラム

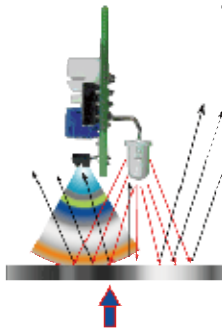


Fig.7.8.1 IR センサ原理図

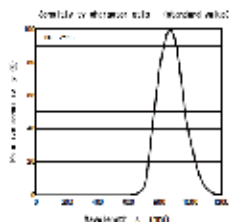


Fig.7.8.2 IR センサ特性図

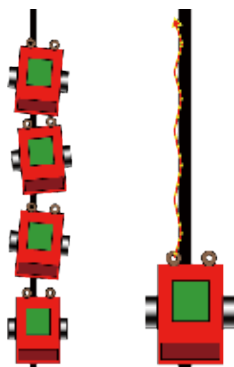
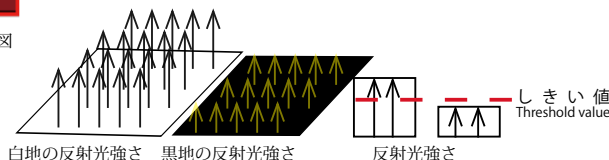


Fig.7.8.3 トレースイメージ図

7.8.1 条件分岐プログラム

1. ロボットがどのような方法で黒線を検知し、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コントローラには、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、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. コントローラボードは、プログラムに従い、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測して、白なのか、黒なのかを判断し、動くことが必要で、この分岐条件になる値を、「閾（しきい）値」と呼びます。



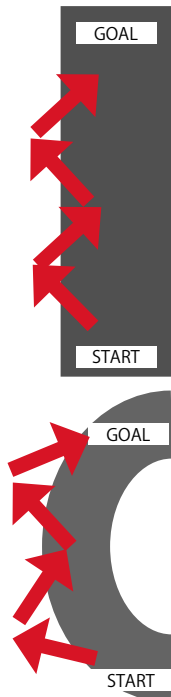
- ・ロボットが動く環境下で計測して、白色・黒色それぞれの反射光の強さを計測します。センサ値計測のソフトを使ってコントローラで測ります。
- ・白色反射光データ平均値と黒色反射光データ平均値の中間をしきい値（分岐条件）として決定しプログラムに書き込みます。

エッジ走行・・・境界の縁をたどります

分かりやすい一言で説明すると「エッジ走行」です。

- 線上にある。黒から白へ左斜め走行
- ↑
- 線上に無い、白から黒へ右斜め走行
- ↑
- 線上にある。黒から白へ左斜め走行
- ↑
- 線上に無い、白から黒へ右斜め走行

ラインは、直線だけでなく、曲線でもたどれます。



Program of a Linear tracing robot

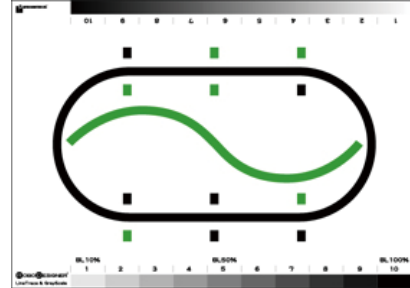
Conditional branch program

1. A black line will be detected and you're going to do stop behavior, and it's moving or... by what kind of way does a robot understand the algorithm.
2. A controller has an LED for emission of light and a module for receiving light, and infrared rays are launched like a left figure and the strength of the catoptric light is measured.
3. Infrared radiation sensor is a receiving light sensor with photo IC (photovoltaic device). One in the wavelength of the light detects a band in the infrared circumference most often.
4. Because the output voltage changes by the amount of the light, when measuring the distance from the light source, when measuring the reflectivity of the color as "range sensor", it would be possible to use it as "reading sensor" of a figure on a floor. More simply, it can be used for judgment of an infrared presence.
5. Infrared radiation sensor changes the light which has reached to an electronic signal and sends information to the robot main body this.
6. A controller board revolves around a motor according to the size of the sensor input value with a program, it stops, a movement is done.
7. There is strong catoptric light from a white part as the nature of the light, and a black part absorbs light, and catoptric light becomes weak. A reaction is big and the person who uses a drawn black line for a floor of the white floor changes into a movement experiment, so it becomes easy to measure.
8. A change in output of analog infrared radiation sensor by the difference in the strength of the catoptric light is measured, a white one or a black one is judged, to move is needed and the numerical value which becomes this branch condition is called "the threshold value".

Please decide about the middle of the white catoptric light data mean and the black catoptric light data mean as a threshold value (branch condition) and write notes in a program.

7.8.2. ライントレースのプログラミング

1. 光の反射率により、白色と黒色で取り出せるセンサ出力数値に違いが出ることを利用し、ラインエッジをトレースします。
2. 白い大きなシートに、黒いテープでラインを引いて、そのエッジ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。
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。



[2]. Programming of linear trace

1. That the difference goes out to the sensor output numerical value which can be taken out by white and black is used and a linear edge is traced by the reflectivity of the light.
2. A line is pulled to a white big seat by a black tape, and it's possible to make the robot which moves while following on it.
3. Make the robot which makes a big circle and runs through its one round with a black line.
4. Please try robot making which runs through a truck made of a big black line by the early speed well.

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【21_02_Line_board-lightsensor_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。

↑ ライントレースシート (A1 サイズ)



7.8.3. ライントレースプログラム作成

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。

明るさセンサを使用して床の白黒を判断しながら行動します。黒線上で、floor 1 (明るさセンサー)の出力が「しきい値 (分岐条件)」以下なら、左旋回します。

でなければ、右旋回し前進します。

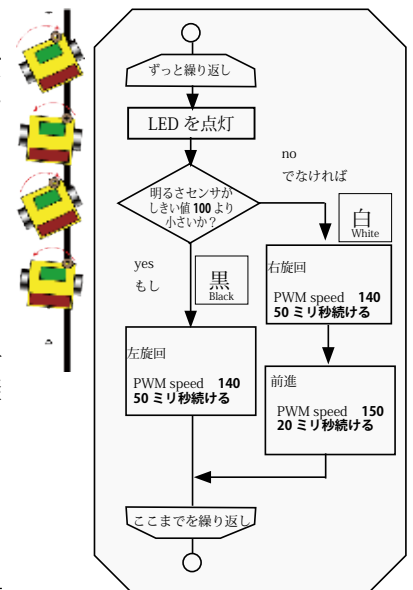
変数 floor1 明るさセンサーの出力値を格納します。

しきい値は、ツール>シリアルモニタを使って floor1 のデータ値を確認して設定します。シリアルモニタから計測データをコピー (Ctrl+C) し、表計算ソフトなどにペースト (Ctrl+V) してグラフ化処理すると傾向値が把握でき、分岐点 (しきい値) 考察が容易になります。

※投射している赤外線と比較して周囲が明るいと「前進」と「旋回」をしきい値で分けることができません。

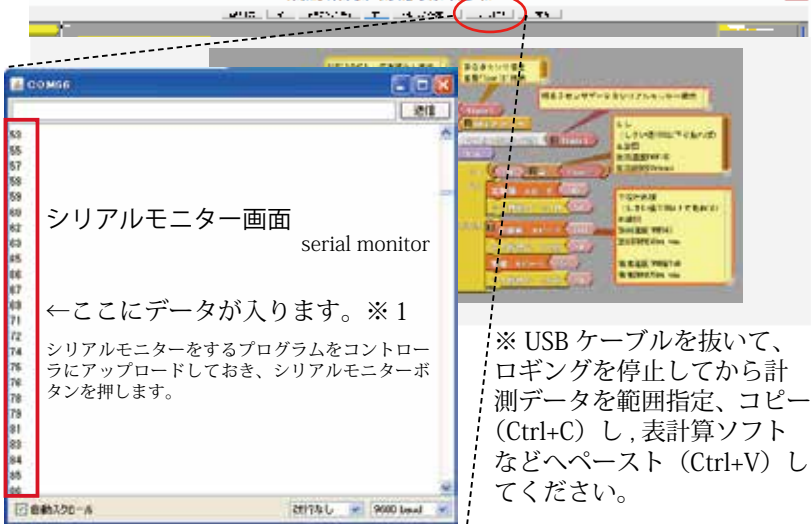
7.8.4. フローチャート図

右側のフローチャートは、上のサンプルプログラム ArduBlock_Line_board-lightsensor_sample をチャート図で表してみました。



7.8.5. プログラム調整ロボット作り込み

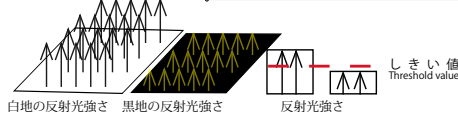
1. センサデータを調べます。…USB ケーブルを接続して計測します。
 1. コントローラのプログラムが実行されている状態の時に、ArduBlock の [シリアルモニター] を使ってセンサの値を調べることができます。(シリアルモニターできるように sample プログラムを作成しています→コントローラへアップロードします)
 2. ArduBlock の [シリアルモニター] をクリックするとシリアルモニター画面が立ち上がり、リアルタイムでセンサ値が表示されます。



3. シリアルモニターでセンサ値を確認しながら白床の上、黒線の上、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)
4. 計測したデータの間間値を、「しきい値」として分岐条件に使います。
5. 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)、ペースト (Ctrl+V) して数値をグラフ化処理するとセンサー値傾向が分かり易くなり、分岐条件 (しきい値) 考察がしやすくなります。…・参照：次頁 - 表計算ソフト使い方

2. 各種パラメータ調整

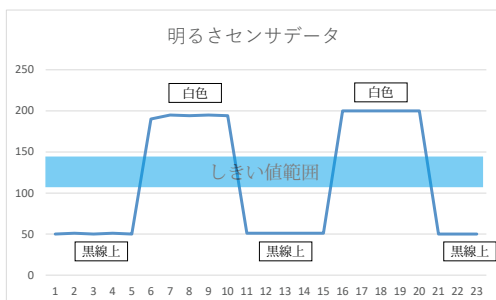
1. 床面の色の違いで、センサに届く赤外線量が変化します。
 - ・シリアルモニターを使って、計測をしながら、ロボットを少しずつ移動していき、黒の線と、白色の床で、色の違いにより、どのように出力が変化するかを調べます。
 - ・データは、数度にわたって調べて、黒色、白色床それぞれに平均値を調べます。
2. センサデータ (黒色からの反射データと、白色からの反射データ) の違いを調べて その中間値を「しきい値」とします。



データ数は 1 秒間で 4000 個を超えるカウント数量です。反射光 1 個ごとは、ばらつきがあり判定がしにくいので、グラフ化すると分かりやすくなります。

3. 「表計算ソフトの利用」が便利です。

・以下は、例として、前頁ライトレースシートで、黒線と白色床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。図の場合、100 ~ 150 くらいが「しきい値」に適しています。

参照：次頁 - 表計算ソフト使い方

Data of a sensor is checked.... A USB cable is connected and measured.

1. It's possible to check output data of a sensor using [serial monitor] of ArduBlock at the state that a program of the microcomputer board in which "Line_base_floor_sample" was written is executed.
2. When [serial monitor] of ArduBlock is clicked, a serial monitor screen stands up, and the sensor value is indicated in real time.
3. They're on the white floor and a black line while confirming the sensor value by a serial monitor and a data collection is performed respectively. (For a USB cable, condition of a connection)
4. The value of the middle of the measured data is used as "threshold value".
5. When graphing makes spreadsheet software copy from a serial monitor, and deals with a figure, it becomes easy to understand.

All kinds' parameter tuning

1. With the color of the resting face, to be different, more, the amount of the infrared rays which reach a sensor undergoes influence of the reflectivity and changes.
 - * You're moving a robot a little and measure using a serial monitor, and it's the color in case of a black line and a white floor, please examine how output changes to be different.
 - * Please check data over several times and check a mean in each of black and white floor.
2. The intermediate value will check the difference in the sensor data (each reflectance data in a black part and a white part), and is "threshold value".

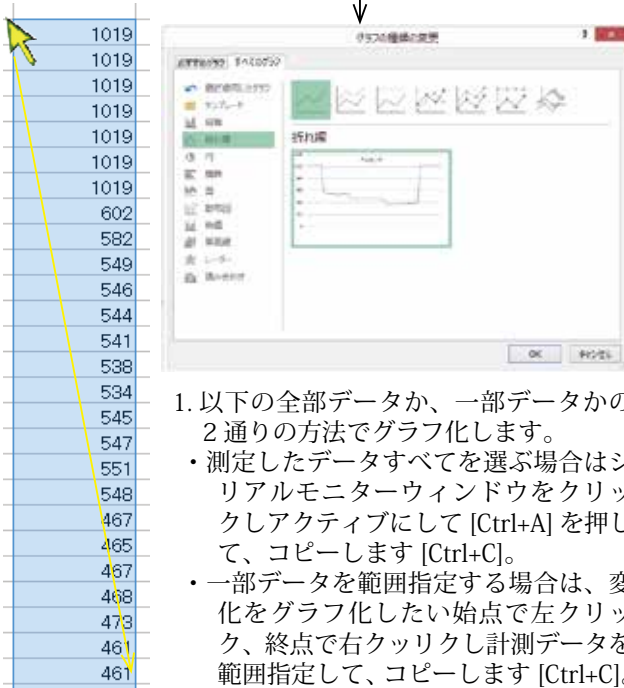
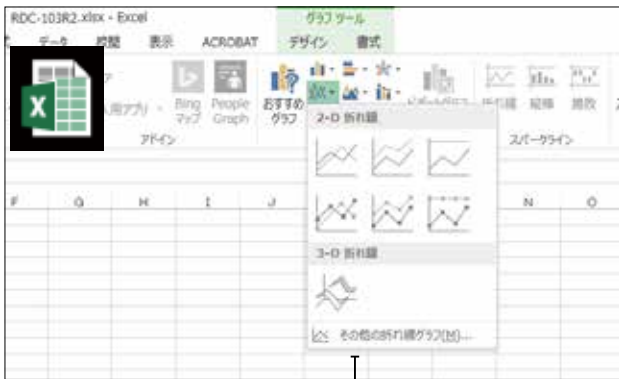
"Practice of spreadsheet software" is needed.

* Below is previous page linear tracing paper and is a figure which graphs data of the cereal monitor who measured black and white as an example. It can be understood and is it possible to be able to be useful for consideration of "the threshold numerical value" by how much signal size or look?

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、コントローラのダブルリセット、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.

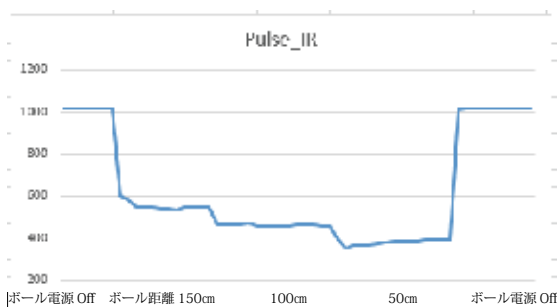
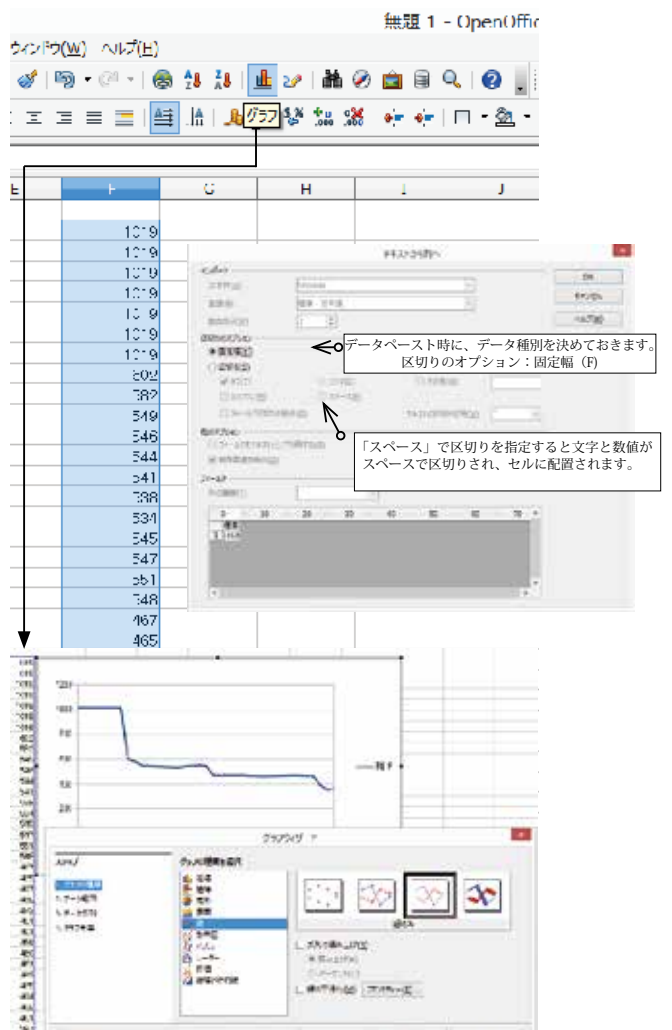
4. 表計算ソフトの使い方

- 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。



- 以下の全部データか、一部データかの2通り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]。
- 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]
- 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。
- 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スペースで区切られ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。

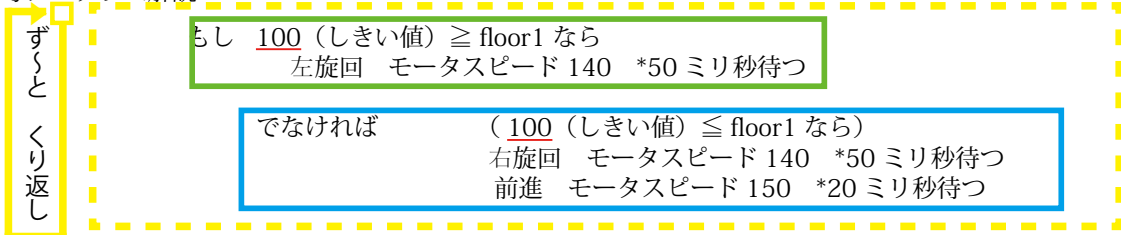
- グラフで見ると、距離ごとに計測数値が違うことも理解でき、プログラムの調整に役立ちそうです。
- ボールから離れているとき、中間距離時、近い距離など、ボールからの距離ごとにデータが違うことが分かり、プログラムで工夫をしてロボットの行動を変えることが可能です。

5. 調べたデータで、プログラムの分岐条件「しきい値」を書き換えます。 [パラメータ調整]

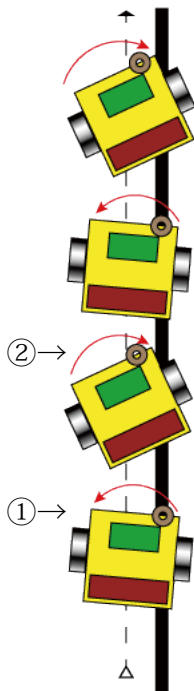
- サンプルでは、floor1 の「しきい値」を 100 と仮に決めていますので、今回、調べた実測値に基づき「しきい値」を書き換えます。



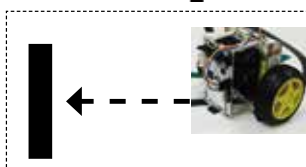
・参考プログラム解説



>>>>> 計測では、黒だと 50 前後、白であれば 200 前後のデータでした。



- 左図の場合で
黒線上で、明るさセンサが「しきい値 (分岐条件)」以下の数値になったら PWM140 のモータスピードで左旋回を 50 ミリ秒間行います。----> ①
* [ミリ秒待つ_ミリ秒] は、結合アイコンの動作時間を設定するパラメータで、継続動作するという意味です。
- 左旋回をすると、黒線上からセンサが離れ、白色床上に到達するので、明るさセンサ出力が「しきい値 (分岐条件)」以上のデータに変化します。
明るさセンサが「しきい値 (分岐条件)」以上の数値になったら 右旋回を 50 ミリ秒間 PWM140 のスピードで行い
モータスピード PWM150 で 20 ミリ秒間前進します。-----> ②
①と②をくり返して黒と白を境界をたどりながら前へ進んでいきます。
白色床と黒線の違いがわかる「分かれ目 (しきい値) 分岐条件」を調べて、サンプルの明るさセンサ数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、黒線をトレースするロボットが作成できるようになります。
- 左旋回、右旋回の時間設定は、左図のロボット振れ幅を設定するパラメータです。
振れを繰り返しながらラインの境界をトレースしますので、振れは必要です。
振れ幅が小さい方が、トレース時間は少なくなります。
- 左旋回、右旋回のモータスピード PWM 値は回転速度を設定するパラメータです。
- 前進のモータスピードと時間設定は、ラインレースコースを進む速さのパラメータです。
- 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。
- 周回トラックを作成し、3 周回を何秒で完走できるかなど、周回時間を測ってみましょう。
- トラックが少し複雑なカーブを持っているなどのコース設定を行うことで、ラインレースプログラミングの難しさにもチャレンジしてみましょう。



5. プログラムを変更し、動きを変えてみます。

- では、正面から進んでいき、黒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。

7.8.6. コンパイル

ArduBlock_Line_board-lightsensor_sample を Arduino へアップロードし、C 言語へ自動変換したコード ArduBlock で作成したプログラムを「Arduino へアップロード」すると、Arduino IDE 画面に、変換された C ソースが出来上がります。

- 以下、前頁で作成した ArduBlock_Line_board-lightsensor_sample を「Arduino へアップロード」し、Arduino IDE 画面に出来上がったソースコードです。
- 直接 C ++ 言語で、コーディングも可能です。

```

Program source code.
ArduBlock_Line_board-lightsensor_sample | Arduino 1.0.5 r2-
#include <STEMDu.h>

STEMDu _STEMDU_robot = STEMDu();
int _ABVAR_1_floor1 = 0;

void setup()
{
  Serial.begin(9600);
}


void loop()
{
  _STEMDU_robot.led((bool)HIGH);
  _ABVAR_1_floor1 = _STEMDU_robot.readLight();
  Serial.print(_ABVAR_1_floor1);
  Serial.println();
  if (( ( 100 ) >= ( _ABVAR_1_floor1 ) ) )
  {
    _STEMDU_robot.leftM1M2(140);
    delay( 50 );
  }
  else
  {
    _STEMDU_robot.rightM1M2(140);
    delay( 50 );
    _STEMDU_robot.forwardM1M2(150);
    delay( 20 );
  }
}
  
```

Upload

1. Upload-controller board writing in is performed automatically, but it's possible to upload manual.
 * When choice clicks an upload button (->) icon of Arduino IDE, writing in to a microcomputer board is begun.

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。

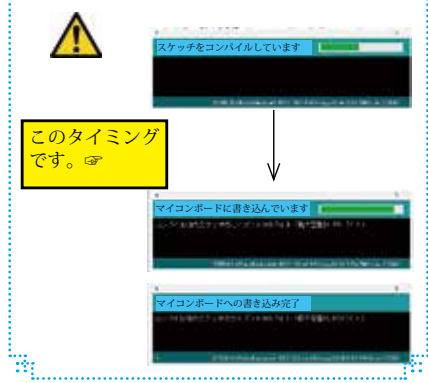


◆エラーメッセージを確認して、対策を施して、問題を解決した後で、再度、マイコンへの書き込みを行います。

・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。

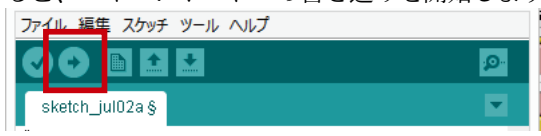


↑ RESET



7.8.7. アップロード

- アップロード～コントローラボード書き込みは、自動で行われますが、手動でもアップロード可能です。
 - Arduino IDE のアップロードボタン (→) アイコンを選択クリックすると、マイコンボードへの書き込みを開始します。



- Arduino IDE 画面では、プログラムの変更、修正も行えます。
 - 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。

7-9. 超音波障害物回避ロボのプログラミング

7.9.1. 超音波距離センサー HC-SR04

- ・超音波の反射時間を利用して非接触で測距するモジュールです。外部からトリガパルスを入力すると超音波パルス（8波）が送信され、出力された反射時間信号をマイコン（Arduino等）で計算することによって距離を測ることができます。
- ・特性を利用して、障害物回避ロボットなどにすることが可能です。

◆主な仕様

- ・測距範囲：2～180cm
（センサー基板正面を中心とした15度の範囲、分解能：0.3cm）
 - ・電源電圧：DC 5.0V
 - ・動作電流：15mA
 - ・動作周波数：40kHz
 - ・トリガ信号：10μS（TTLレベルのパルス波）
 - ・エコー出力信号：反射（往復）時間
 - ・サイズ：45×20×15mm
- ※通電時はGND端子が最初に接続されるようにしてください

7.9.2. 超音波距離センサー取付

- ・センサー基板の超音波距離センサー取り付け用ソケットの穴に、表示を合わせてセンサーピンを差し込みます。

超音波 HCSR04 センサー端子表示	RDC-103 ソケット端子表示
Gnd	G
Echo	Ec11
Trig	Tr
Vcc	V
—	G

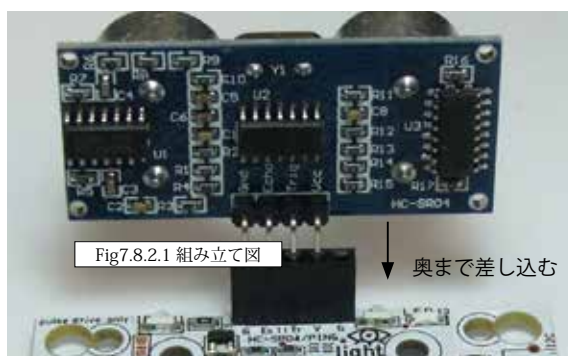


Fig.7.8.2.1 組み立て図

奥まで差し込む

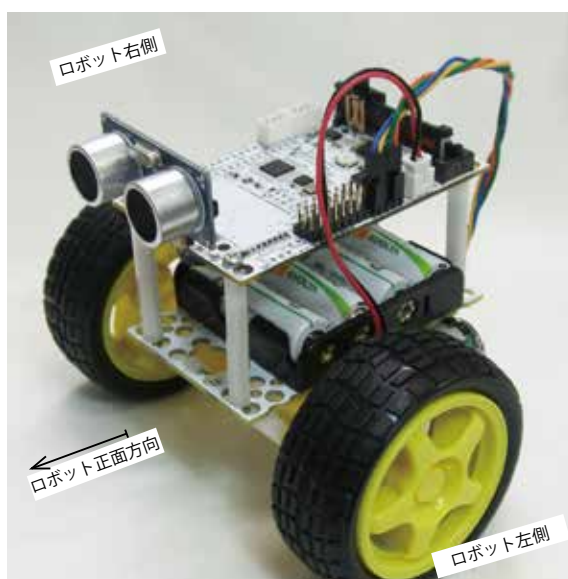


Fig.7.9.2.2 組み立てた状態図

Ultrasonic distance sensor HC-SR04

* It's non-contact using the ultrasonic reflective hour and is the module which does ranging.

When a trigger pulse is input from outside, ultrasonic pulse (8 wave) is sent, and it's possible to measure the distance by calculating an output reflective time signal by a microcomputer (Arduino).

◆ The main specification

- * Distance surveying area : 2-180cm
(15 times of area where it was made the center in front of the sensor substrate and resolution :0.3cm)
- * Line voltage : DC 5.0V
- * Movement electric current : 15mA
- * Operation frequency : 40kHz
- * Trigger signal : 10 μ S
(wave pulse of the TTL level)
- * The echo output signal : reflective
(round trip) hour
- * Size: 45 x 20 x 15 mm
- ※ When turning on, please make sure that the GND terminal will be connected first.

Assembly of an ultrasonic range sensor.

* A sensor pin is put in a hole of a socket for ultrasonic range sensor installation of a sensor substrate together with indication.

7.9.3 プログラム (ArduBlock) を作成します。



上図を参考にプログラムを作成してください。
 超音波センサからの距離情報が変数 threshold よりも
 小さくなったならモータを停止します。
 走行速度との停止位置 (threshold) をうまく調整する
 必要があります。

・参考プログラム解説

- ①. 超音波センサの値を「変数」distance に代入するプログラム部分。



- ②. 停止位置設定プログラム部分です、動作環境でシリアルモニターにて障害物回避させたい距離を計測して、しきい値を決め、threshold の [値] 数値に書き込みます。



- ③. シリアルモニタープログラムユニットを挿入すると、超音波センサ距離情報確認が容易になります。



- ④. 超音波センサ距離情報 [distance] がしきい値 [threshold] 以下になったら [≥], 停止します。でなければ前進 (モータ PWM 値 100) します。



前進スピード (走行速度) と停止位置の調整をしてください。早すぎると、停止させたい位置をオーバーランした状態になり、障害物に衝突します。
 ・停止した後に、どのように動作させるかは、皆さんでプログラミングしてください。

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【21_01_clash_avoidance-HCSR04.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。

- 21_01_clash_avoidance-HCSR04_sample を Arduino へアップロードし、C言語へ自動変換したコード
 ・直接C++言語で、コーディングも可能です。

```

Program source code
#include <STEMDU.h>

int _ABVAR_1_threshold = 0 ;
int _ABVAR_2_distance = 0 ;
STEMDU_STEMDU_robot = STEMDU();
int ardublockUltrasonicSensorCodeAutoGeneratedReturnCM(int pingPin)
{
    long duration;
    pinMode(pingPin, OUTPUT);
    digitalWrite(pingPin, LOW);
    delayMicroseconds(2);
    digitalWrite(pingPin, HIGH);
    delayMicroseconds(5);
    digitalWrite(pingPin, LOW);
    pinMode(pingPin, INPUT);
    duration = pulseIn(pingPin, HIGH);
    duration = duration / 59;
    if ((duration < 2) || (duration > 300))
return false;
    return duration;
}

void setup()
{
    digitalWrite( 11 , LOW );
}

void loop()
{
    _ABVAR_1_threshold = 10 ;
    _ABVAR_2_distance = ardublockUltrasonicSensorCodeAutoGeneratedReturnCM( 11 ) ;
    if ( ( ( _ABVAR_1_threshold ) >= ( _ABVAR_2_distance ) ) )
    {
        _STEMDU_robot.stopM1M2 ();
    }
    else
    {
        _STEMDU_robot.forwardM1M2(100);
    }
}
        
```

7.9.4. 超音波障害物回避プログラム(C)

Cで作成してみます。

※ サンプルのソースコードは、PC/MyDocuments/Arduino/へ配置したサンプルフォルダ [RDS-X Examples_C] に、ファイル名【ultrasonic_sample_avoidance】で格納されています。

ソースはC言語で記述されています。

※ Arduinoを起動後、[ファイル]→[スケッチブック]→[RDS-X Examples_C]→[ultrasonic_sample_avoidance]を開くと確認できます。

※以下は、サンプルコードの内容です。薄い色の文字はコメント文です。



※ After starting Arudino, [file,-> [sketchbook] can confirm that-> [RDS-X Examples_S]-> opens by [ultrasonic_sample_avoidance].

```

crash-avoidance_ping
for JAPANROBOTECH RDC-103
Souce code of ultra sonic distance sensor control
from Arduino Sample Sketch "Ping".
*/

// 機体とポート類の接続は 左：小さい番号<->大きい番号：右
//A connection of a fuselage and port kinds The left: The small number The big number: The right?

// 機体に接続するもの：モータ M1,M2 超音波センサ HC-SR04
// Something to connect to a fuselage: Motor M1,M2, Ultrasonic sensor HC-SR04
// MotorDriver Pin Assign on RDC-103.
int M1_1 = 4;
int M1_2 = 5;
int M1_PWM = 6;
int M2_1 = 7;
int M2_2 = 8;
int M2_PWM = 9;

// pin number of the sensor's output:
int pingPin = 11;

int LEDPin = 13;

int PWM_Value = 200; // how speed the Motor is
int distance = 15; // how long from target object, unit is cm

void setup() {
// initialize serial communication:
Serial.begin(9600);
// initialize the digital pin as an output.
pinMode(M1_1, OUTPUT);
pinMode(M1_2, OUTPUT);
pinMode(M1_PWM, OUTPUT);
pinMode(M2_1, OUTPUT);
pinMode(M2_2, OUTPUT);
pinMode(M2_PWM, OUTPUT);

```

```

pinMode(LEDPin, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
// establish variables for duration of the ping,
// and the distance result in inches and centimeters:
long duration, inches, cm;

// The PING))) is triggered by a HIGH pulse of 2 or more microseconds.
// Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
pinMode(pingPin, OUTPUT);
digitalWrite(pingPin, LOW);
delayMicroseconds(2);
digitalWrite(pingPin, HIGH);
delayMicroseconds(5);
digitalWrite(pingPin, LOW);

// The same pin is used to read the signal from the PING))) a HIGH
// pulse whose duration is the time (in microseconds) from the sending
// of the ping to the reception of its echo off of an object.
pinMode(pingPin, INPUT);
duration = pulseIn(pingPin, HIGH);

// convert the time into a distance
inches = icrosecondsToInches(duration);
cm = microsecondsToCentimeters(duration);

if(cm >= distance){
// turn the Motor on
analogWrite(M1_PWM, PWM_Value);
digitalWrite(M1_1, HIGH);
digitalWrite(M1_2, LOW);
analogWrite(M2_PWM, PWM_Value);
digitalWrite(M2_1, HIGH);
digitalWrite(M2_2, LOW);
digitalWrite(LEDPin, LOW); // turn
the LED off by making the voltage LOW
}
else{
// turn the Motor off

```

```
digitalWrite (M1_1, LOW);
    digitalWrite (M1_2, LOW);
digitalWrite (M2_1, LOW);
    digitalWrite (M2_2, LOW);
    digitalWrite (LEDPin, HIGH); //
turn the LED on (HIGH is the voltage level)
}

Serial.print (inches);
Serial.print ("in, ");
Serial.print (cm);
Serial.print ("cm");
Serial.println ();

delay (100);
}

long microsecondsToInches (long
microseconds)
{
// According to Parallax's datasheet for the
(PING)), there are
// 73.746 microseconds per inch (i.e. sound
travels at 1130 feet per
// second). This gives the distance travelled by
the ping, outbound
// and return, so we divide by 2 to get the
distance of the obstacle.
// See: http://www.parallax.com/dl/docs/prod/
acc/28015-PING-v1.3.pdf
return microseconds / 74 / 2;
}

long microsecondsToCentimeters (long
microseconds)
{
// The speed of sound is 340 m/s or 29
microseconds per centimeter.
// The ping travels out and back, so to find the
distance of the
// object we take half of the distance travelled.
return microseconds / 29 / 2;
}
```

(2). プログラム動作後、[ツール] → [シリアルモニター]で確認できます。After program movement, it can be checked by [tool]-> [serial monitor].

シリアルモニターウィンドウのデータをコピーした内容です。A measured data example of an ultrasonic sensor by a sample program

サンプルプログラムによる超音波センサの実測データ例（対象物までの距離を in, cm で出力）(The distance to the target thing is output by in, cm.)

93in, 239cm	26in, 67cm
94in, 240cm	28in, 71cm
59in, 150cm	30in, 77cm
31in, 80cm	32in, 82cm
28in, 71cm	47in, 122cm
28in, 72cm	51in, 131cm
31in, 81cm	26in, 66cm
26in, 67cm	25in, 64cm
26in, 68cm	30in, 78cm
19in, 49cm	7in, 20cm
19in, 49cm	5in, 15cm
22in, 56cm	131in, 334cm
31in, 80cm	

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。



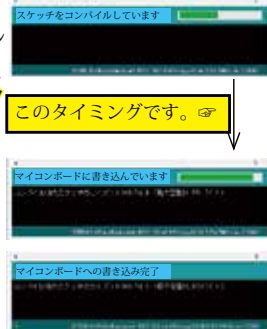
◆エラーメッセージを確認して、対策を施して、問題を解決した後で、再度、マイコンへの書き込みを行います。

エラーメッセージ 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。



↑ RESET



Error Message

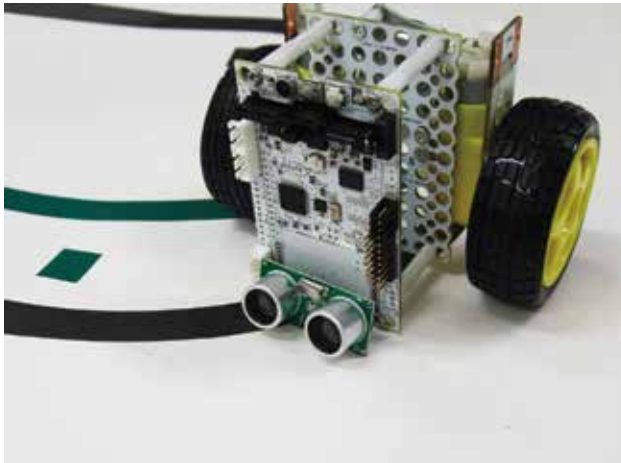
Couldn't find a Leonardo on the selected port. Check that you have the correct port selected. If it is correct, try pressing the board's reset button after initiating the upload.

7-10. 製作例② 障害物回避 と ライントレース をするロボ

障害物がラインコース上に配置されている時には、ライントレースと障害物回避の2機能をロボットに備えます。超音波センサの信号ピンを写真のように曲げます。

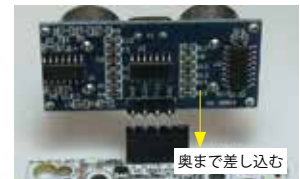


・ピンの曲げ角度は、障害物の高さを考慮して調整します。
【注!】曲げ加工を繰り返すと、金属疲労を起し、ピンが折れて壊れます。

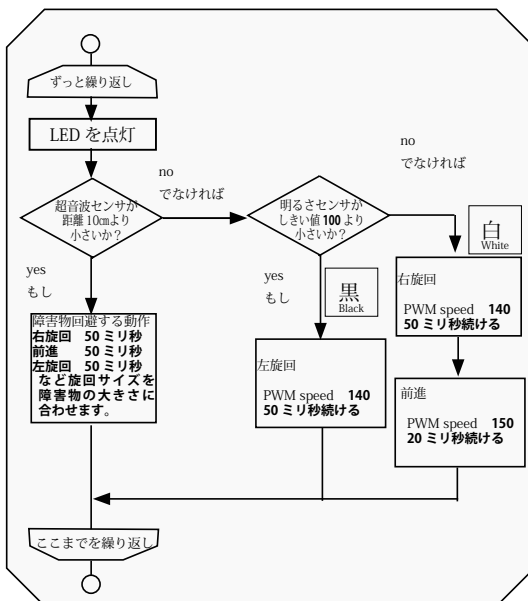
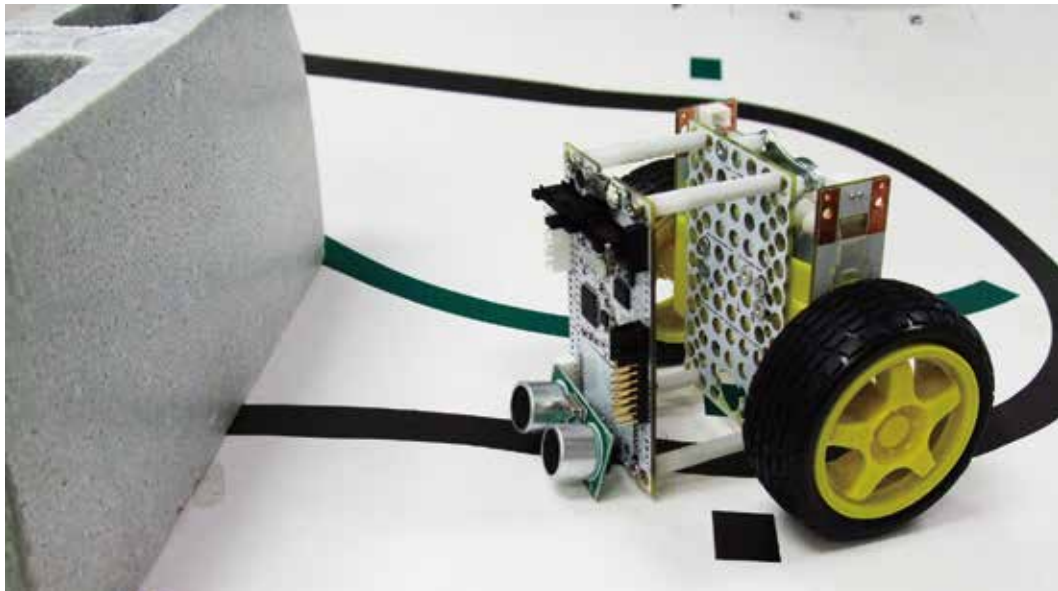


信号ピンを曲げ加工した超音波センサを、センサ基板の超音波距離センサ取り付け用ソケットの穴に、表示を合わせてセンサーピンを差し込みます。

超音波 HCSR04 センサ 端子表示	RDC-103 ソケット 端子表示
Gnd	G
Echo	Ec11
Trig	Tr
Vcc	V
—	G



奥まで差し込む



・超音波センサで、障害物回避をします。



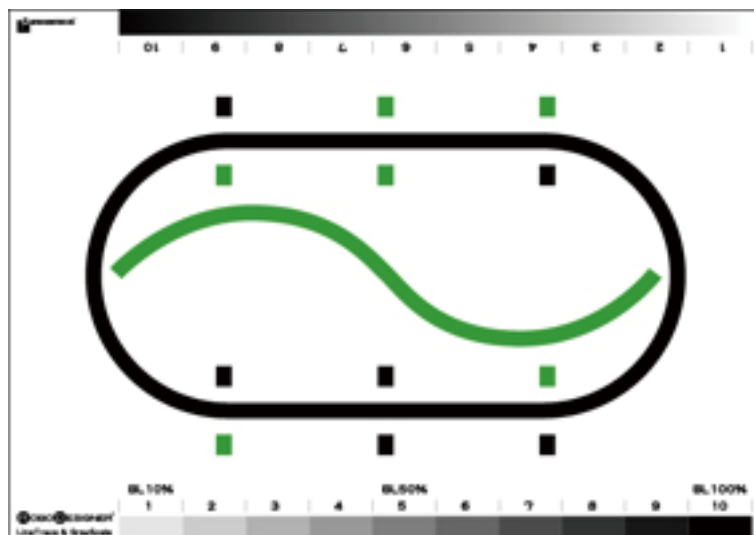
・基板搭載するセンサでライントレースをします。

7.8 章 ライントレースロボ と 7.9 章 障害物回避ロボのプログラムを組み合わせ、プログラムを作成します。左のフローチャート図がアルゴリズムです。

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、障害物回避：ファイル名【21_01_clash_avoidance-HCSR04.abp】で格納されています。ライントレース：ファイル名【21_02_Line_board-lightsensor_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。1 サンプルごとに開いて確認し、組み合わせて使用します。

7-11. ライントレース競技例

ライントレース実験シート
RDP-971
シートサイズ A1



競技例 [1]

プラクティクス競技場例；S字カーブ、交差路、直角路を課題に入れています。



競技例 [2]

グループ対抗競技会場；障害物、ライン切れ、路面悪路が課題に入っています。

