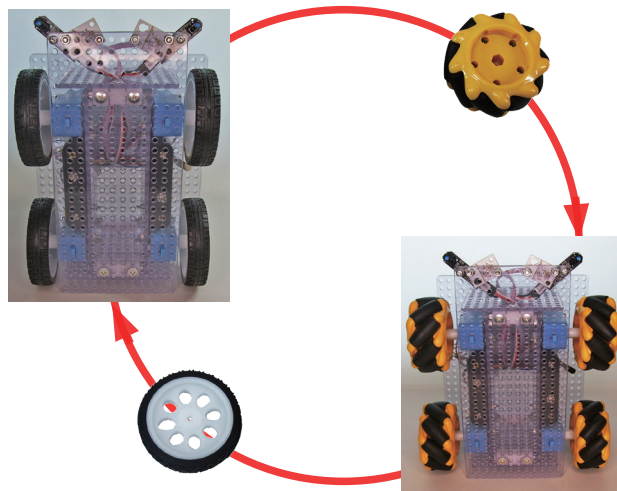
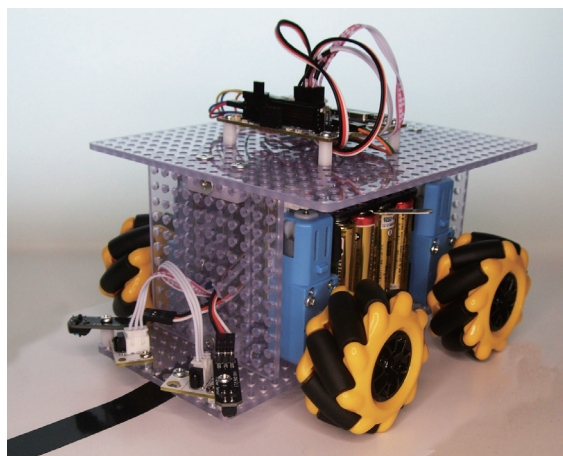


15 Omnidirectional 全方向移動制御ロボ



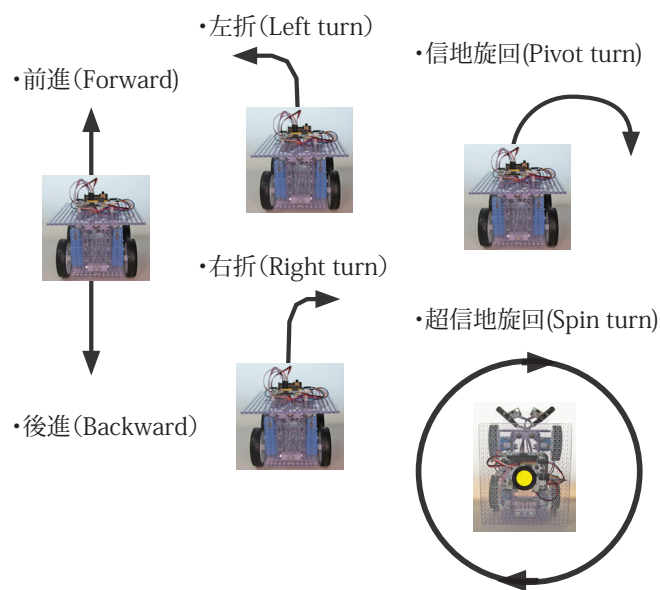
全方向移動制御スターロボット概要

駆動部にメカナムホイールと標準車両型ホイールの2刀流ドライブを備え、目的に合わせてホイールを交換する、トランスフォームロボットを作り、全方向移動を実現、プログラミングします。

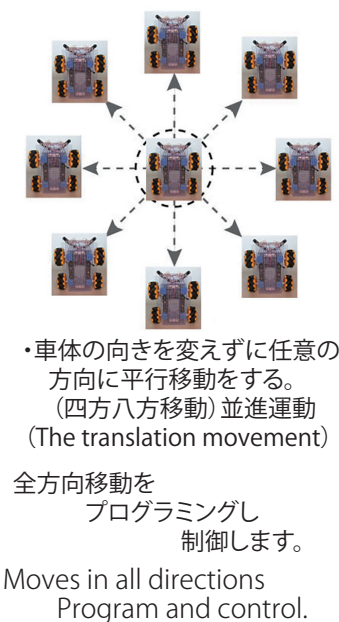
Omnidirectional Control Stater Robot Overview

Duals wheeld drive with Mecanum wheel and standard vehicle type wheel in the drive part. A robot that moves in all directions by selecting and incorporating according to target that.

自律移動、車両型ホイール動き Vehicle type move



メカナムホイール動き Mecanum wheel move



RDS4WD55_104TypeIII

(液晶なし、使用可能モータ数 (DC モータ×4、サーボモータ×8)、超音波センサ増設可能)
(without LCD, Number usable motors (DC motor x 4, servo motor x 8) ultrasonic sensor can be added)

RDS4WD55_104TypeIII +

(液晶付、使用可能モータ数 (DC モータ×4、サーボモータ×8)、超音波センサ増設可能)
(with LCD, Number of usable motors (DC motor x 4, servo motor x 8), ultrasonic sensor can be added)

RDS4WD55_103TypeIII +

(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)
(with LCD, Number of usable motors (DC motor x 4, servo motor x 4), ultrasonic sensor can be added)

のモデルがあります。

すべてのモデルに

- ・赤外線反射センサ
 - ・変調赤外線センサ
 - ・音センサ
 - ・明るさセンサ
 - ・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。

- ・超音波距離センサ
- ・加速度/ジャイロセンサ増設可能

For all models:

Equipped with infrared reflectiveness sensor, irregularity infrared radiation sensor, sound sensor, brightness sensor and slider, various controls can be performed using these.

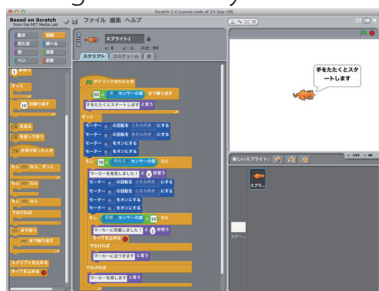
* Ultrasonic range sensor, Acceleration/gyro sensor can be added.

Outline of a system

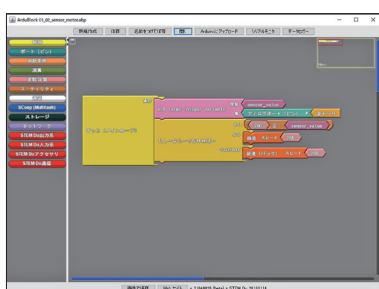
Program control practice robot, Move to omnidirectional.

The outline of a system of RDS-4WD55 is the following figure.

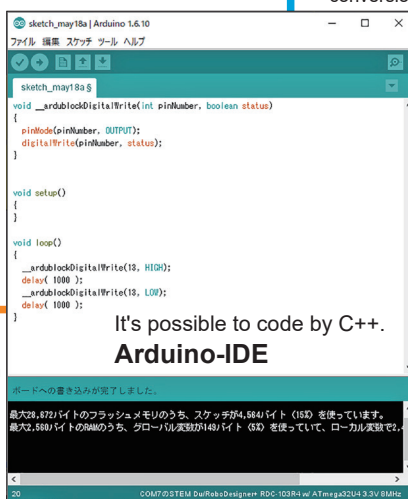
A program development environment can make Arduino a core and use visual program environment by ArduBlock, and a Scratch. Circuit composition consists of a microcontroller, a DC motor (with metal gear) and a motor driver. A power supply and a drive course are chosen as 6 [V] power supply, and 6 [V] is making a control system move.



Scratch [Visual programming language, USB connection use]



ArduBlock [Visual programming language]



To a C language, automatic program conversion.

To a C language, automatic program conversion.

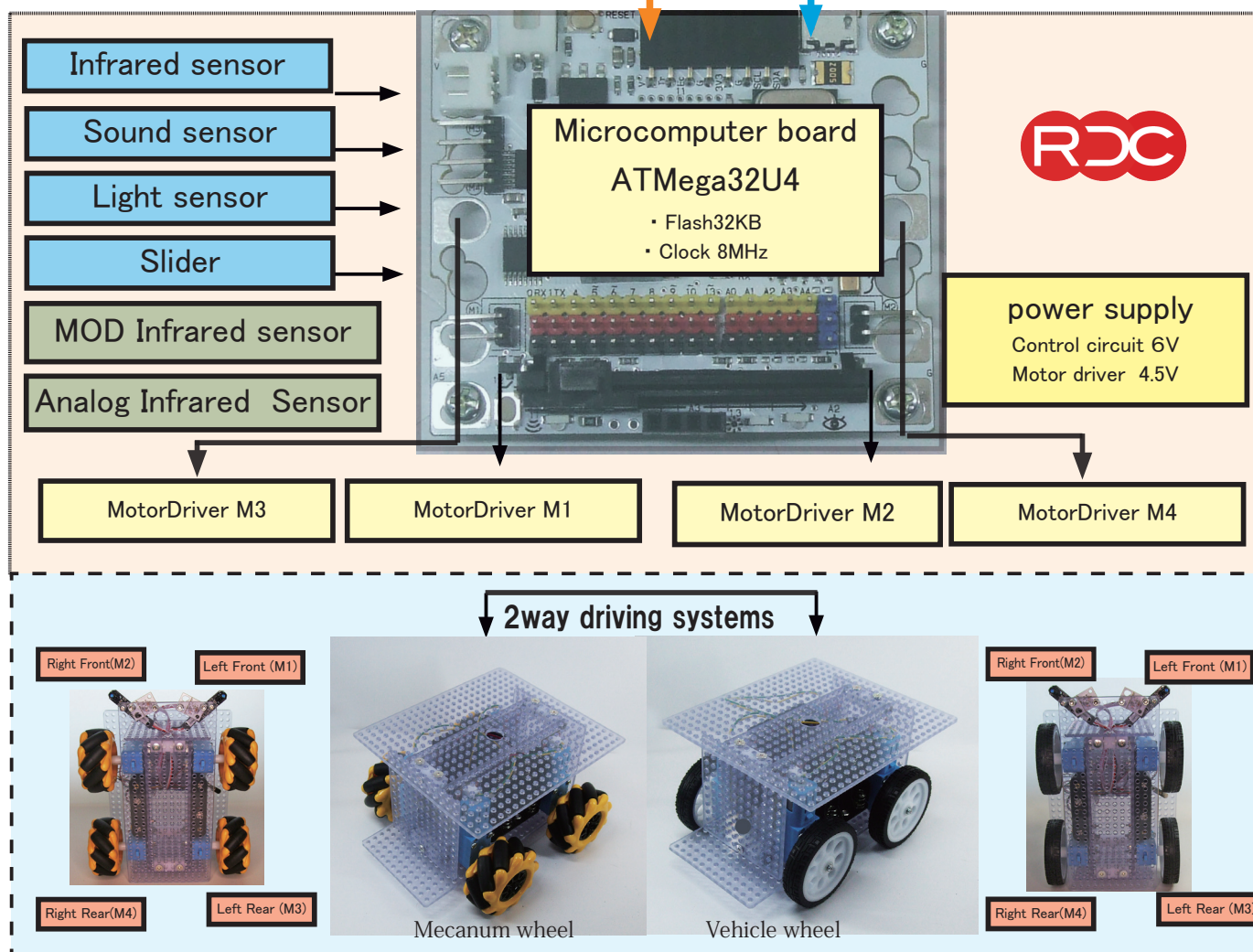
It's possible to code by C++.

Arduino-IDE

Arduino program environment is CC BY-SA.

[Creative Commons License.]

All circuit information inclusion is opened.



15.1. Omnidirectional RDS-4WD55 構成部品

A product of this mark can check details by URL of Japan Robotech Ltd.,
<https://www.japan-robotech.com>

15.1.1. Omnidirectional movement 2way control Robot RDS-4WD55 List of parts

☐ Microcomputer board
☐ マイコンボード  RDC-104 1枚

TYPE3

セットモデルで、付属マイコンボードが変わります。

Attached microcomputer board changes on set model

RDS-4WD55_104Type3:RDC104TypeⅢ

RDS-4WD55_104Type3+:RDC-104_TYPEⅢ+

Type3+

Mounting parts:

Resin spacer M3x10mm 4 pcs

Pan head (machine) screw M3x6mm 4pcs

Accessories:

Pan head machine screw M3x8mm 4 pcs

・音センサ・明るさセンサ・スライダ搭載

Equipped with irregularity infrared radiation sensor, sound sensor, brightness sensor and slider,

・モータドライバーM1,2,3,4搭載

Motor driver M1,M2,M3,M4

☐ Microcomputer board
☐ マイコンボード  RDC-103 1枚

セットモデルRDS-4WD55_103Type3+は、こちらのマイコンボードが付属します。

Set model RDS-4WD55_103Type3+ comes with this microcomputer board

☐ MOD_Infrared Sensor
☐ 変調赤外線センサ  RDI-203JR

・赤外線リモコンの変調信号(38kHz搬送波)を受信するためのセンサです。デジタル/アナログジャンパがあります。

The sensor to receive modulating signal of an infrared remote control (38kHz carrier). There is digital/analog setting jumper..

Accessories:

30cm sensor cable

Pan head machine screw M3x8mm 4 pcs

Nut 8pcs

☐ Analog infrared reflectiveness sensor
☐ アナログ赤外線反射センサ  RDI-ICRT5000A

・ライントレース等に使用します。

・色の明暗濃度違いによる反射光量を検出計測します。

* It's used for linear trace.

* The catoptric light amount by the bright and dark side density difference in the colors is detected, a measured sensor.

Accessories:

30cm sensor cable

Pan head machine screw M3x8mm 2 pcs

RajinSpacer AS-310 2pcs

☐ 機構部シャーシセット Mechanism chassis set

出荷時:標準ホイール取り付け自律移動車両型移動機構

Shipment: Mobile mechanism with the vehicle type standard wheel for autonomus system movement

全方向移動ロボ用メカナムホイール交換部品

Mecanum wheel replacement parts for Omni-directional moving robots

| | |
|---|---|
|  | Dummy battery It's the same shape as a AA battery, there is no electromotive force. |
| ・ダミー乾電池1個(V2用) 取付 単3電池と同じ形状、起電力はありません。 | |
| ・22cmモータケーブル2本組 4セット付き | 22cm motor cable 2 pcs set with 4 sets |
| ・プチ袋(ポリエチレン製気泡緩衝材)入り | Comes with a petit bag (polyethylene bubble wrap) |

| | | |
|-----------------------|-----|----------------------------------|
| ①ホイールキャップ | 4個 | Wheel rajin cap Diameter30mm |
| ②M3 x 30ナベネジバネ座金付 | 4本 | M3x30mm pan screws 4pcs |
| ③丸ワッシャーM10 | 4枚 | washer 11x22x1.5 4pcs |
| ④ホイールL® | 各2台 | Mecanum wheel L® 2 each |
| ⑤丸ワッシャー8.5 x 32 x 3.2 | 8枚 | washer 8.5x32x3.2 8pcs |
| ⑥異形変換シャフト | 4本 | Irregular conversion shafts 4pcs |
| ⑦ボールキャスターOW-011 | 1個 | Ball caster OW-011 1Pcs |
| M3x15座金組み込みネジ | 2本 | M3x15 washer built-in screw 2pcs |
| ナット ワッシャー | 各2個 | Nuts washer 2 each |

☐ プログラミング開発環境  CD-ROM 1枚

USB cable A-microB

☐ マイクロUSBケーブル  RDP-824 1本

☐ ケース Store case

ロボット組み立て時収納ケース
w226×D188×H174mm

☐ 鉄製貼付けウエイト5/10 g シート 2枚
Balance weight

Spare parts

☐ 補修用ネジ・ナット 1袋

・ナット Nut M3 10個

・ナット Nut M4 10個


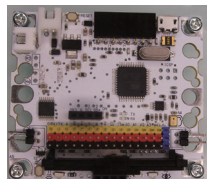


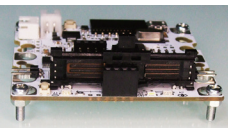

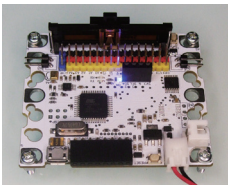

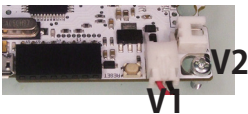

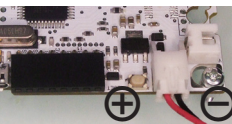


・ネジ pan Screw M3x10mm 6本

・ネジ pan Screw M4x8mm 6本

15.1.2. 部品の見方、使い方

How to use the part.

| 構造体パンチングボード ユニバーサルプレート | Structure punching board universal plate |
|---|---|
| <p>ロボット構造体に、格子状に穴をあけた硬質塩ビ板を使います。 透明色です、塗装でオリジナル可能です。</p> <p>板厚3.0mm Thickness 穴サイズ3.5mm Hole diameter 穴ピッチ7.0mm Interval</p> <p>切削加工は、ドリルやジグソー（電動鋸（のこぎり））等で加工できます。</p> <p>分解や拡張工作メモ TopBase、UnderBase、RearBase、SideBase のボードは、7mm間隔格子状に3.5mm径穴を全面に開けています。</p> | <p>Hard PVC board with holes in a grid pattern is use for robot structure It's transparent color, the original is possible by painting. Hard Polyvinyl chloride board.</p> <p>• 3.5 mm diameter hole in 7 mm grid line.</p> <p>Cutting is done with drills & jigsaw (electric saw) Can be processed with a saw.</p> <p>Disassembly & expansion work memo TopBase、UnderBase、RearBase、SideBase board is 7 mm 3.5 mm diameter hole is drilled on the entire surface in a grid pattern.</p> |
| <p>追加拡張部品の取付に利用できます。 拡張時のネジ取り付け例・・・・・・・・・・＞</p> <p>① M4 ネジ・・・穴を下穴にして タッピングでメネジ穴を作成 M4 をネジ込めます。 ② M4 タッピングトラスネジ 直接締結 ③ M4 タッピングナベネジ 直接締結 ④ M4 ナベドリルネジ 直接締結 ⑤ M4 超低頭タッピングネジ 直接締結 ⑥ M3 ナベネジ ネジ・ネジ・ナットで取付</p> | <p>It can be used to install additional expansion parts .</p> <p>① When using M4 pan screws: M4 can be screwed using holes as pilot holes (You can make a female screw hole using M4 tap tool.) ② M4 Tapping truss screw----Direct tighten a screw. ③ M4 Tapping pan screw----Direct tighten a screw. ④ M4 Pan drill screw----Direct tighten a screw. ⑤ M4 Super-bow tapping screw----Direct tighten. ⑥ M3 pan screws・・・mounting with screws and nuts</p> |
| <p>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。</p> | <p>Part size indication The size indication of a screw and a nut is as follows.</p> |
| <p>■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diameter)3mm、長さLength10mm」という意味です。</p> | <p>Pan Screw: As the name it looks like the head is lying down. M3 = Diameter 3mm, Length=10mm</p> |
| <p>■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diameter)3mm、長さLength4mm」という意味です。</p> | <p>Low Head Machine Screw: It's modest and is the name of the screw with the flat shape. M3=Diameter 3mm, Length=10mm</p> |
| <p>■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。</p> | <p>Flat Head Screw: A head is the name of the screw with the flat shape like a dish. M3=Diameter 3mm, Length=10mm</p> |
| <p>■タッピングネジ：部材にめねじが切られていなくても締結が可能なネジで部材に直接締結します。 表記:M2.3 × 10mm 「太さ(直径)2.3mm、長さ10mm」の意味。</p> | <p>Tapping screw: Even if a female thread isn't cut into a part, it's concluded directly in a part by the screw it's possible to conclude.</p> |
| <p>■ナット Nut 表記:M3 「太さ(直径)3mm のネジ用」という意味。</p> | <p>Nut Transcription :M3 The meaning as "for screws of 3 mm of thick (diameter)".</p> |
| <p>■共通で使う上記以外のパーツ</p> <p>・座金付組ネジ・スペーサ・バネ座金 ・平座金・プッシュピンリベット</p> | <p>Parts other than the above used is common ・ screws with washers・spacer ・ spring washer・plain washer・push rivet</p> |

| 2. 電子基板使用時の注意 | | Precautions when using electronic board. |
|--|---|---|
|  ①. 基板表面のピン同士をショートさせない。 | | Do not short the pins on the surface of the board. |
| 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。 |  | Please do not short the servo pins present on the surface. If it short-circuits the board will be break. |
|  ②. 裏面には必ず隙間を空けて使う。 | | Be sure to leave a gap on the back side. |
| 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。 |  | Electronic board also has minute parts & circuit on the backside. The pattern is arranged. If you apply pressure or touch the metal object it will be destroyed. It may cause the board to break. |
| 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。 |  | We have provided a gap using screws and nuts, the parts and circuit patterns on the backside are installed. Please be careful not to touch the parts. |
|  ③. 規格範囲内の電圧で使用する。 | | Use at a voltage within the standard. |
| 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。 |  | Electronic components used in electronic circuits are designed to operate at a specific voltage of 2V to 5V. If right it will operate if voltage is lower than the specified range. If wrong or too high the circuit will be damaged. |
|  本製品のマイコンボードの電源電圧規定値は | | Microcomputer board' s specific power supply voltage is. |
| 回路,M1,M2用V1: 6.0V モータM3,M4用V2: 4.5V |  | V1; For circuits : 6.0V V2; For motor M3,M4 4.5V |
|  ④. 電源電圧極性を間違えない | | Do Not mistake power supply voltage polarity. |
| 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。 |  | Please be careful not to make mistake with plus/minus of the power supply voltage connected to the circuit board. If you make mistake the parts will break. |
|  ⑤. 水分大敵！ 電子回路は、水をかけると壊れます。 | | Moisture enemy ! Electronic circuits break when exposed to water. |
| 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。 |  | When making a robot you are absorbed in it. Before you know it you knock the container etc., liquid will enter the circuit board and stays there. There are accidents such as breaking. Do not place cup containers, work memos on the same table as liquids. |

15.2. マイコン ボード概要

Overview of microcomputer board

| RDC-104TYPE III仕様 | Microcomputer board RDC-104TYPE III Spec |
|--|---|
| ・ 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。 | ・ It's possible to make the autonomous robot which becomes independent of a PC using 4DC motor-and moves. |
| ・ 赤外線センサ、LED / 光センサ、音センサ、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えます。 | ・ It's equipped with LED / Light sensor-, Sound sensor and a Slider on the board, using these, you can control variously. |
| ・ 外部超音波センサー / 加速度・ジャイロセンサ増設可能 (I ² C) | ・ It's possible to connect Ultrasonic Sensor, and the Acceleration / Gyro sensor (sensor socket use), I ² C----(separate sale part). |
| ・ 外部アナログセンサー 6 個まで接続可 | ・ It's even possible to connect 6 of outside analogue sensor (A0,A1,A2,A3,A4,A5). |
| ・ 2 電源式 制御回路、M1,M2 モータ電源 6V 供給 (M3,M4 モータ電源 最大 12V まで使用可能) | ・ 2 power supply system For circuit and M1,M2 power supply 6V (Even at most 12 V of motor power supply is practicable. M3,M4). |
| ・ サーボモータ 10 個まで接続可能 | ・ It's even possible to connect 10 servomotors. |
| ・ 通常はスケッチ (プログラム) に合わせて配線します。 | ・ Wire according to the sketch (program). |
| ・ LCD モニター必要な場合は、RDC-104TYPE III + plus をご利用ください。 | ・ When a LCD monitor needs, please use RDC-104TYPE III + plus. |

| 仕様 | The specification | DataSheet URL |
|-------------------|-----------------------------------|---|
| マイコン / ATMEGA32U4 | MCU / ATMEGA32U4 Clock 8MHz | http://media.digikey.com/pdf/Data%20Sheets/Atmel%20PDFs/AT-mega16U4,32U4.pdf |
| 音センサ | Sound Sensor SPU0410HR5H-1 | https://media.digikey.com/pdf/Data%20Sheets/Knowles%20Acoustics%20PDFs/SPU0410HR5H-1.pdf |
| スライダーボリューム | SlidePotentiometers Alps RS30H121 | http://www.alps.com/WebObjects/catalog.woa/J/HTML/Potentiometer/SlidePotentiometers/ |
| 明るさセンサ | Light sensor Everlight PT12-21C | http://www.everlight.com/file/ProductFile/PT12-21C-TR8.pdf |
| 赤外線センサ | Vishay Semiconductor VSMB10940 | https://www.digikey.jp/product-detail/ja/VSMB10940/VSMB-10940TR-ND/3915205 |

デジタル入出力 Digital in/out

端子を使用したい時はピンで接続します。

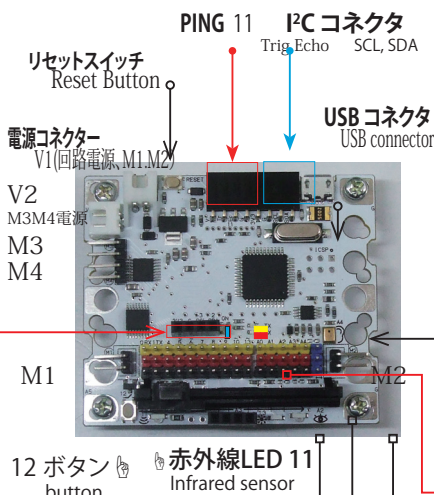
| ピン番号 No. | ヘッダーピン | 記号 symbol | 解説 (Commentary) |
|----------|--------|----------------------|--|
| 13 | | M4 (0.5A 程度) | サーボ / 白色LED/PWM 出力可能 R/C servo motor / White LED / PWM output |
| 12 | | M4 (0.5A 程度) | ボタン Button |
| 10 | | | サーボ / 超音波 / 赤外線LED R/C servo motor / UltraSonic / InfraRed |
| 0 | | M3 (0.5A 程度) | サーボ / シリアルRX R/C servo motor / Serial RX |
| 1 | | M3 (0.5A 程度) | サーボ / シリアルTX R/C servo motor / Serial TX |
| 11 | | | サーボ / LCD CS / PWM 出力可能 R/C servo motor / LCD CS / PWM output |
| 6 | | M1 (0.5A 程度) | サーボ / M1 PWM 制御 / PWM 出力可能 R/C servo motor / M1 PWM control / PWM output |
| 9 | | M1 (0.5A 程度) | サーボ / M1 制御 / PWM 出力可能 R/C servo motor / M1 control / PWM output |
| 4 | | M1 制御 | M1 control |
| 7 | | M2 制御 | M2 control |
| 8 | | M2 制御 | M2 control |
| 5 | | 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 | 回路、M1,M2 Power Conector |
| | V2 | 電源コネクタ | M3,M4 Power Conector |

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

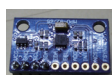


増設可能

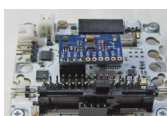
★超音波センサ Ultrasonic sensor (差し込んで使用します。別売品)



加速度/ジャイロ/温度センサ Accelerometer/Gyroscope (別売品)



加速度/ジャイロセンサ取付状態図



| ピン番号 No. | ヘッダーピン | 記号 symbol | 解説 (Commentary) |
|------------------|--------|-----------|---------------------------------------|
| PING | | V | 電源 ⊕ライン Power |
| | | Tr | Trig |
| | | Ec | Echo |
| | | G | ⊖ライン |
| I ² C | | 3.3V | 電源 ⊕ライン Power |
| | | G | ⊖ライン |
| | | SCL | SCL |
| | | SDA | SDA |
| | | A5 | アナログ入力ポート Analog input |
| | | A4 | 音センサ Sound Sensor / Analog input |
| | | A3 | スライダー:アナログ入力ポート Slider / Analog input |
| | | A2 | 明るさセンサ/アナログ入力ポート Light / Analog input |
| | | A1 | アナログ入力ポート Analog input |
| | | A0 | アナログ入力ポート Analog input |
| | | G | ⊖ライン |

音センサ Sound Sensor A4 ジャンパーワイヤ接続
アナログ入力 Analog input A0~A4 5ポート
ピンで接続します。A は Analog の A) 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。
Connect with pins. A reads the input voltage from A) 0 of Analog to the power supply voltage (3.3V) in 1024 steps. You can connect sensors, volumes, etc. You can also use it as a digital input / output pin by changing the settings in the sketch.

みの虫クリップ用端子 Terminal for clips (抵抗等測定) A5

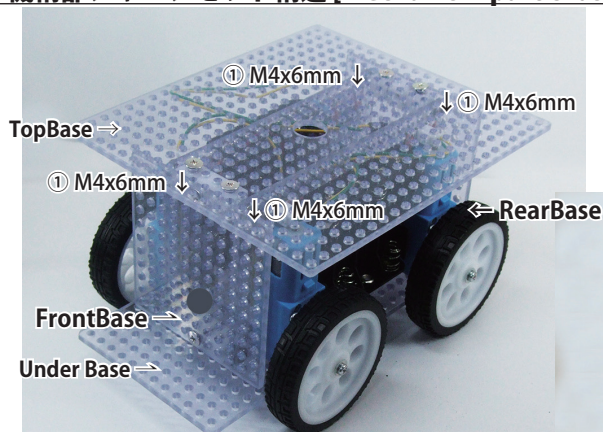
スライダー Slider resistance A3 ジャンパーワイヤ接続

明るさセンサ Light sensor A2 ジャンパーワイヤ接続

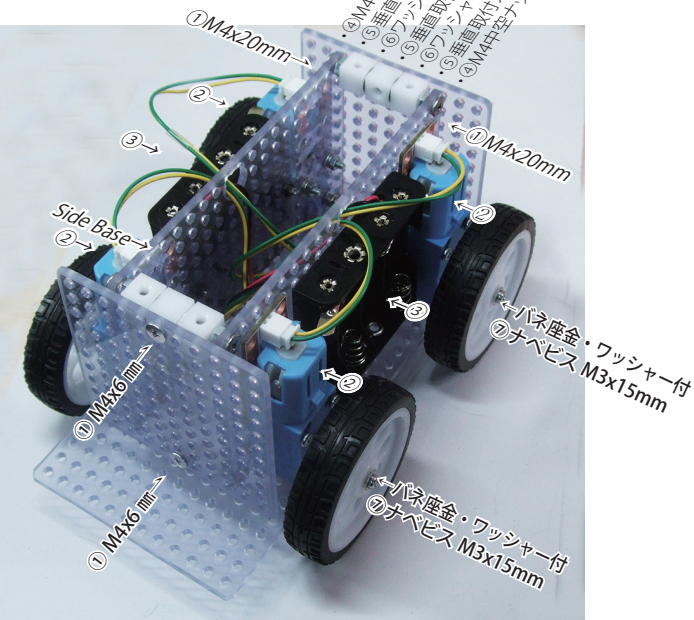
ボード内臓の音センサ、スライダー、明るさセンサ使用時には、ジャンパーワイヤで接続設定します。
A connection is established by a jacket wire at the time of a sound sensor of board internal organs, a slider and brightness sensor use.

15.3. 走行台車組み立て [Assemble of a Vehicles]

15.3.1 機構部シャーシセット構造 [Mechanism part chassis set structure]



← 自律走行標準ホイール組立図 →



| | | |
|---|----------|--------------------------|
| ① | 超低頭ネジ | Low head machine screw |
| ② | ギアードモータ | DC gearedmotor |
| ③ | 電池ケース | Battery holder |
| ④ | M4 中空ナット | Hollow nut |
| ⑤ | 垂直取付スペーサ | Vertical mounting spacer |
| ⑥ | ワッシャー | washer |
| ⑦ | ナベビス | Pan screw |

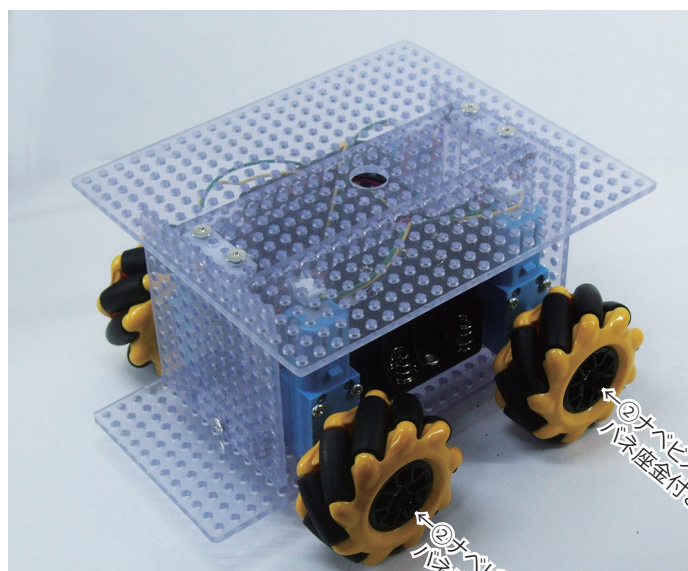
分解や拡張工作メモ

Disassembly & expansion work memo

TopBase、UnderBase、RearBase、SideBase のボードは、7 mm 間隔格子状に 3.5 mm 径穴を全面に開けています。追加拡張部品の取付に利用できます。

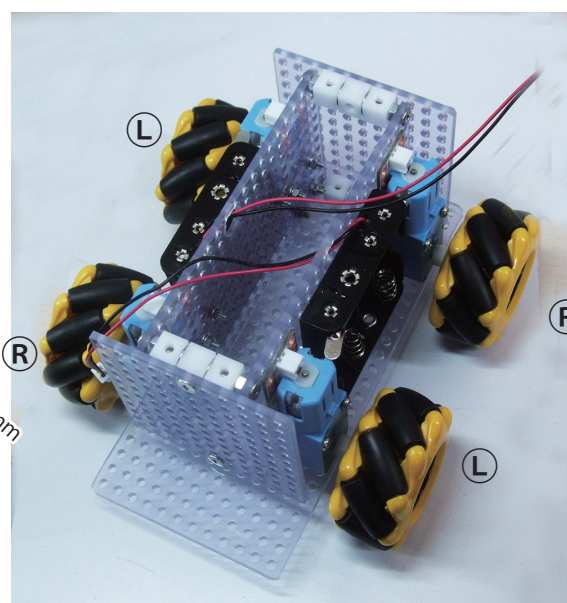
7 mm 3.5 mm diameter hole is drilled on the entire surface in a grid pattern. It can be used to install additional expansion parts.

15.3.2 タイヤホイール交換

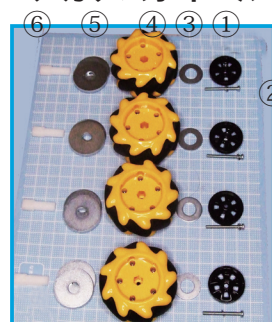


メカナムホイールはⓁⓇがあります。記号マークを確認して取り付けます。

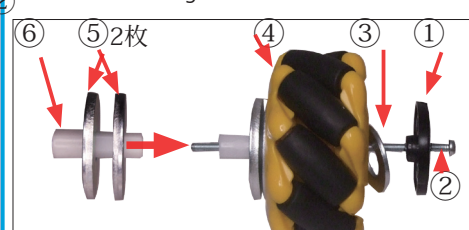
There are Ⓛ and Ⓡ symbol on Mecanum wheel. The symbol is confirmed and it's installed.



メカナムホイール取付

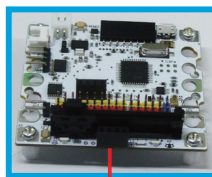


ホイール組み図
Combination figure



| | | |
|------------------------|-----|----------------------------------|
| ①ホイールキャップ | 4個 | Rajin wheel cap Diameter30mm |
| ②M3 x 30ナベネジパネ座金付 | 4本 | M3x30mm pan screws 4pcs |
| ③丸ワッシャー-M10 | 4枚 | washer 11x22x1.5 4pcs |
| ④ホイールⓁⓇ | 各2台 | Mecanum wheel ⓁⓇ 2 each |
| ⑤丸ワッシャー-8.5 x 32 x 3.2 | 8枚 | washer 8.5x32x3.2 8pcs |
| ⑥異形変換シャフト | 4本 | Irregular conversion shafts 4pcs |

15.3.3 マイコンボード・センサー取り付け Assemble of Microcomputer Control board/ Sensor



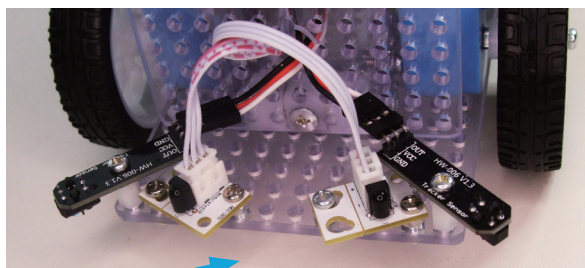
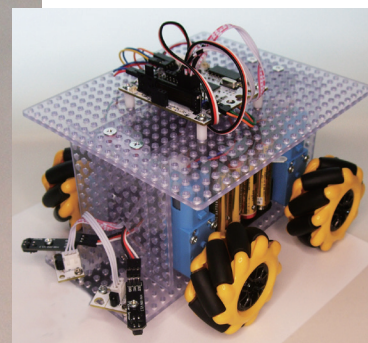
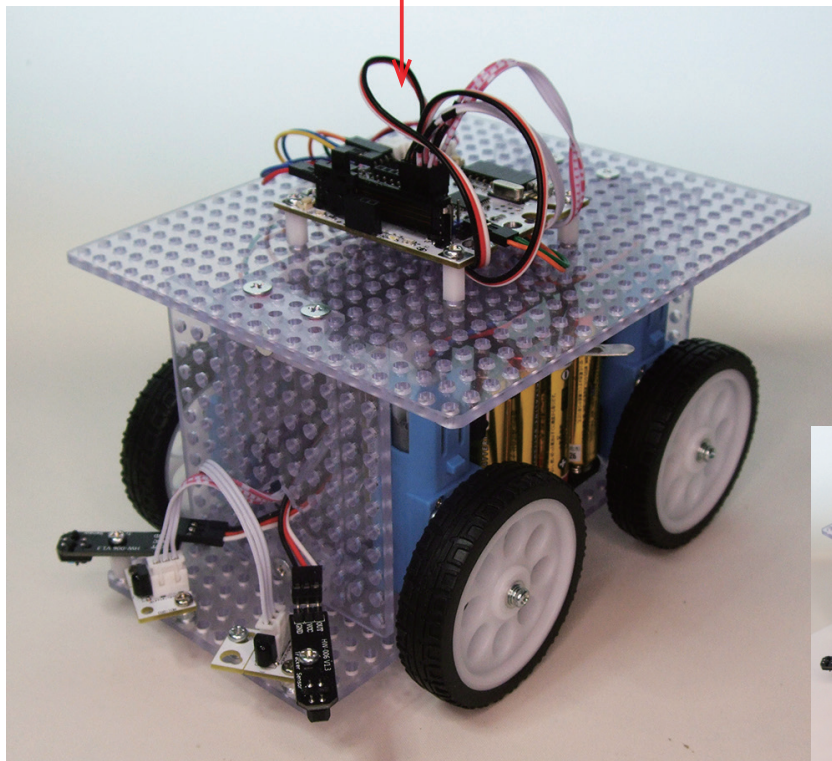
マイコンボード・・・4隅にスペーサ取り付け

Microcomputer board・・・

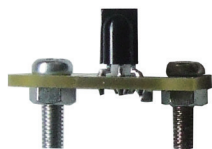
••spacers attached to 4 corners

ジュラコンスペーサM3x10mm

Resin Spacer M3x10mm



参考図のように取り付けます。It's attached as shown in the figure.



•M3x10mmネジを取り付けます。

•対角穴に取り付けます。

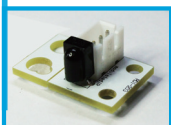
•この角度違いが、センサー指向性を確保します。

•Attach a M3x10mm screw.

*It's attach in the opposite hole.

*This angle difference secures the sensor directivity.

変調赤外線センサ Modulate infrared sensor

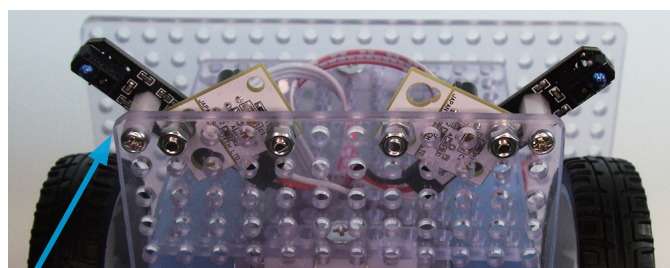


•ロボカップジュニアサッカー公式ボールが発光する変調赤外線に反応します。

•赤外線リモコンの変調信号(38kHz搬送波)を受信するためのセンサです。

*MO sensor is measured, RoboCup junior soccer ball emitted infrared light.

*The sensor to receive modulating signal of an infrared remote control (38kHz carrier).



裏面から見た取り付け図The attached figure seen from the under.



•樹脂スペーサM3x10mmを取り付けます。

Attach Resin spacer M3x10mm.

赤外線反射センサ Infrared reflectiveness sensor



•搭載LEDより赤外線を発光、床などからの反射光を受光LEDで検知します。

•センサ出力・・・アナログ信号

•測定反応距離25mm～30mmです、ロボット制作時に注意します。

* Than a equipped LED, infrared rays, emission of light and catoptric light from a floor are detected by a receiving LED.

* Sensor output... analog signal.

* It's 25 mm-30 mm of measurement reaction distance.

15.4. 配線 wiring

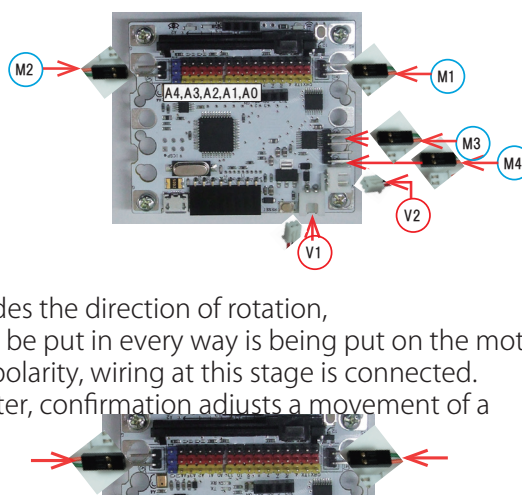
1. 下記の接続一覧表を参考に、各部品とコントローラの配線をします。 Wiring connection

| 部 品 | マイコンボード | 接 続 | |
|---|---------|------------------------------------|---|
| 前列左側ギアードモータ Front Left side Geared-motor | M1 | モータケーブル* 2 line Motor cable | *モータケーブルは、極性があります。差し込みが逆になると、モータ回転が逆になります。・・・ ・モータ回転を逆にする時はマイコン側差し込みを逆にします。 * When the insertion polarity is becomes reverse to a motor cable a motor rotation becomes reverse. ... when making the motor rotation reverse, the microcomputer side plug is made reverse. |
| 前列右側ギアードモータ Front Right side Geared-motor | M2 | モータケーブル* 2 line Motor cable | |
| 後列左側ギアードモータ Back Left side Geared-motor | M3 | モータケーブル* 2 line Motor cable | |
| 後列右側ギアードモータ Back Right side Geared-motor | M4 | モータケーブル* 2 line Motor cable | |
| 左側変調赤外線センサ left side MO-infrared sensor | A2 | 3芯センサケーブル** 3 line sensor cable | ** センサケーブルには、極性があります。ピンヘッダー(接続コネクタ)のG 黒色⊖、V赤色⊕、S(信号)黄色をあわせて差し込みます。 A sensor cable has polarity. The one of the connector Ground->Black⊖, Voltage->Red⊕, and Signal->Yellow is put in all together. |
| 右側変調赤外線センサ Right side MO-infrared sensor | A3 | 3芯センサケーブル** 3 line sensor cable | |
| 左側赤外線反射センサ Left side Infrared reflective sensor | A0 | 3芯センサケーブル** 3 line sensor cable | |
| 右側赤外線反射センサ Right side Infrared reflective sensor | A1 | 3芯センサケーブル** 3 line sensor cable | |
| 電池ケース(単3 x 4) Battery holders (AA battery x4) | V1 | 赤/黒ケーブル Red/Black cable | マイコンボード、M1、M2動作用 For microcomputer board, M1 and M2 movement. |
| 電池ケース(単3 x 4) 1本のダミー電池と3本の乾電池を実装 holders (AA battery x3+Dummy1) | V2 | 赤/黒ケーブル Red/Black cable | M3、M4動作用 単3電池3本と、ダミー電池1本を合わせて4本を取り付け For M3 and M4 movement. |

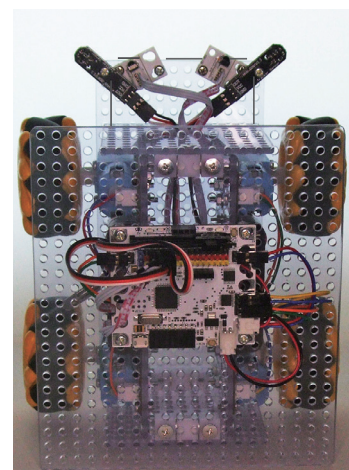
***モータ極性** モータには、回転方向を決める極性がありますが、モータケーブル先端には、どちら方向にでも差し込める無極性のコネクタを付けています、この段階での配線は極性を気にせずに接続します。後ほど、実動テストを行うときにロボットの動きを確認調整します。

* motor polarity

A 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 motor cable point, without worrying about polarity, wiring at this stage is connected. When doing an actual working test later, confirmation adjusts a movement of a robot.



ロボットの前面側 Front side



ロボットの後面側 Back side

****電池** 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。

** Battery_

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

・V1接続電池ケース↓



単3アルカリ乾電池4本
AA alkaline batteries_4pcs

・V2接続電池ケース↓



単3アルカリ乾電池3本と
ダミー乾電池1本
AA alkaline batteries_3pcs
and Dummy battery1pcs

*****電源スイッチ** 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。

*****Power switch** Power supply switch is cut until you'll be ready for a program.

電源スイッチは2か所です。

V1用



V2用

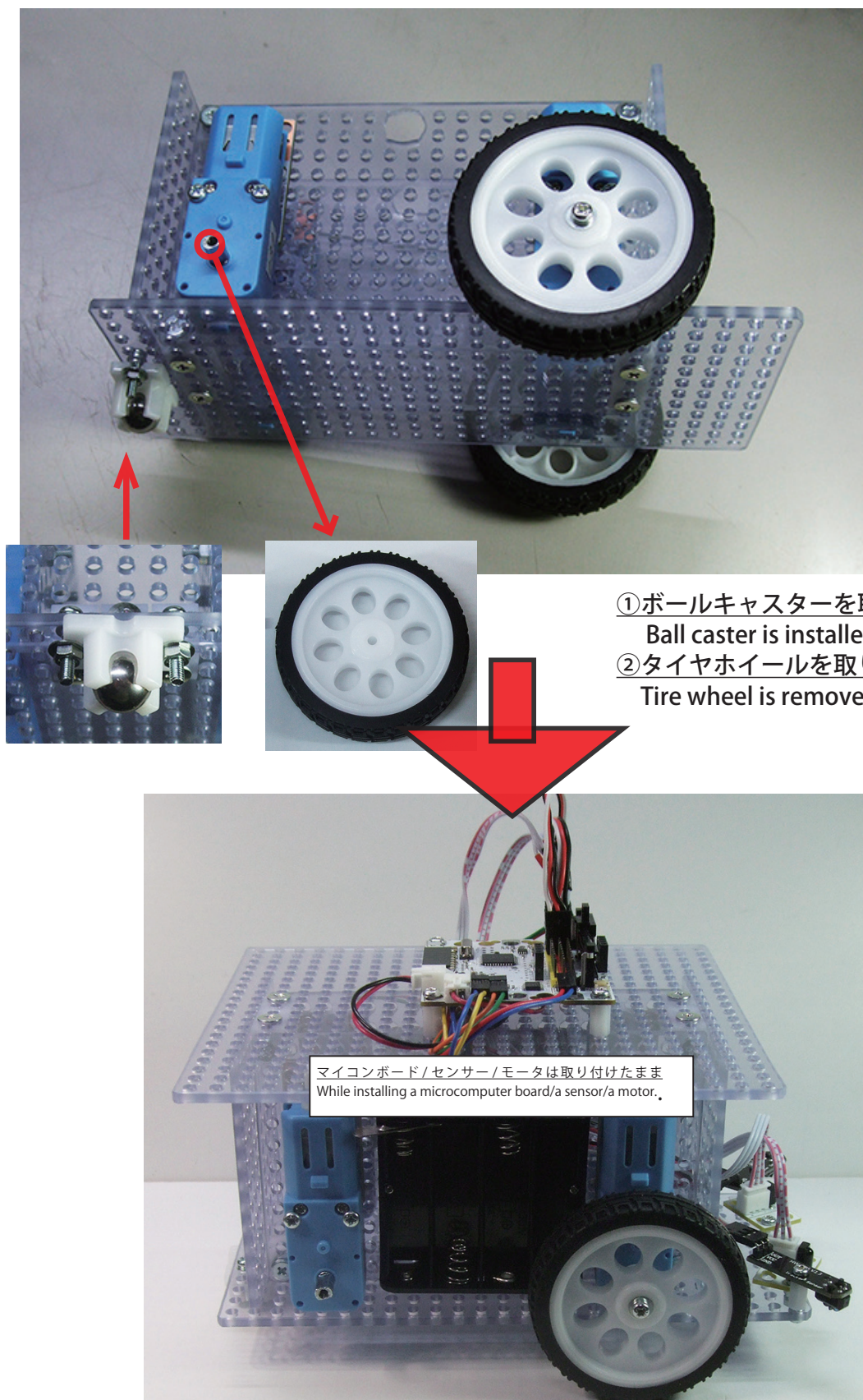


15.5. 動作確認含め、2 輪でのテスト用ロボ作成

Robot making for 2 wheel tests for an operations check

マイコンボード / センサー / モータは取り付けたままで、後輪タイヤホイールを外し、2 輪へ改造します。

A rear wheel is removed and it's remodeled into 2 wheels while installing a microcomputer board/a sensor/a motor.



①ボールキャスターを取り付ける。

Ball caster is installed.

②タイヤホイールを取り外す。

Tire wheel is removed.

マイコンボード / センサー / モータは取り付けたままで
While installing a microcomputer board/a sensor/a motor.

改造後の 2 輪テスト用ロボット (Test robot after remodeling)

15.6. ロボットの動作確認

Operations check of a robot

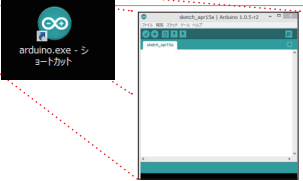
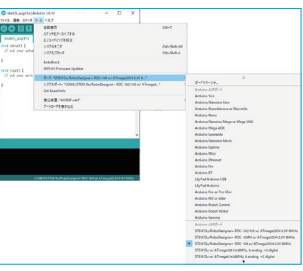
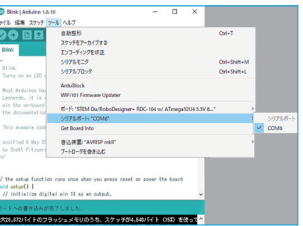
15.6.1. 開発環境起動の事前準備…COM ポート番号の確認

Confirmation of the communication port number.

| | |
|--|--|
| 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。 | 1. PC and a microcomputer board are connected by Micro USB cable. |
| 2. コントローラに電源が入ったことを確認します。 | 2. Power supply switch of a microcomputer board, is turned on. |
| 3. PC がマイコンボードを感知し、PC 側の「COM ポート」が設定されますので、次の手順に従い、COM 番号を調べてください。 ・Windows : マイコンコンピュータ > コントロールパネル > ハードウェアとサウンド > デバイス マネ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. A 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 : It's being held by My computer > Control Panel > Hardware and the sound > Device manager and [port (COM and LPT) COM] is found. |
| 4. (COM と LPT) COM] に認識出現している USB シリアル デバイス (COM 番号) を確認し、COM 記号の後ろにある数字 (ポート番号) をメモします。 | 4. (COM and LPT) You check the USB serial communication device from which recognition emerges in COM number and take notes of the number which is behind the COM symbol (port number). |
| 5. プログラム転送処理時に COM 番号が必要となります。 | 5. The COM number is needed at the time of program transmission upload. |
| ※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。 | ※ When a program can't be written in, please be sure to confirm the COM number. |

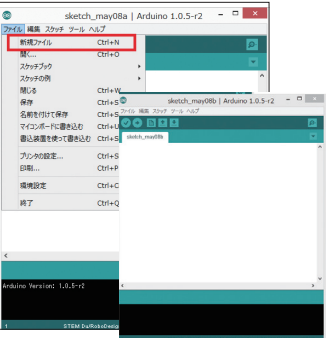
15.6.2. プログラム開発環境の起動

Start of a program development environment

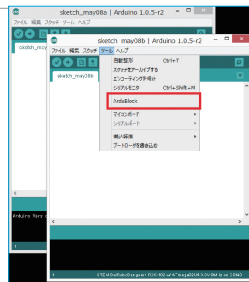
| | | |
|---|---|---|
| 1. デスクトップに作成したショートカット arduino.exe をダブルクリックして arduino を起動します。 |  | 1. The short cut made in a desktop arduino.exe is double-clicked and arduino is started. |
| 2. Arduino-IDE [ツール] で [マイコンボード]、[シリアルポート] を確認します。 | 起動中の arduino 画面 The arduino screen which has started. | 2. By Arduino-IDE [tool]. [Microcomputer board], [serial port] is confirmed. |
| 3. [マイコンボード]: Arduino-IDE [ツール] ▶ [マイコンボード] をクリックし、[ボードマネージャー] に出現するマイコンボードリストで、[STEM Du/RoboDesigner+ RDC-104w/ 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。 |  | 3. [Microcomputer board] : Arduino-IDE [tool] > [STEM Du/RoboDesigner+ RDC-104w/ ATmega32U4 3.3V 8MHz] is chosen and clicked by the microcomputer board list which clicks [microcomputer board] and appears in [Board manager], and it's designated. A ● mark sticks to the list left end. |
| 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. |
| 4. [シリアルポート]; Arduino-IDE の [ツール] ▶ [シリアルポート] をクリックし、出現するサブウィンドウで、先ほどデバイスマネージャーで調べた COM 番号の通信ポートをクリック指定し、チェックマークがついたことを確認します。 |  | 4. [Serial port] of ;Arduino-IDE [Tool] > COM which clicked [serial port] and checked it by a device manager a short while ago by the subwindow from which I emerge A click designates a communication port of the number, and it's confirmed that a check mark stuck. |
| 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. |

プログラム開発環境の準備

Preparations of a program development environment

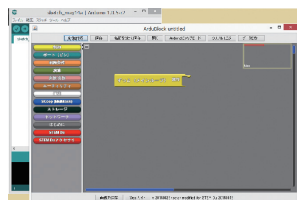
| | | |
|--|---|--|
| 5. Arduino-IDE の [ファイル] > [新規ファイル] をクリックします。新規ファイル [Sketch. 日付] が作成されます。 |  | 5. The [new file] in the [file] of Arduino-IDE is clicked. A new file [Sketch. Date] is made. |
|--|---|--|

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



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

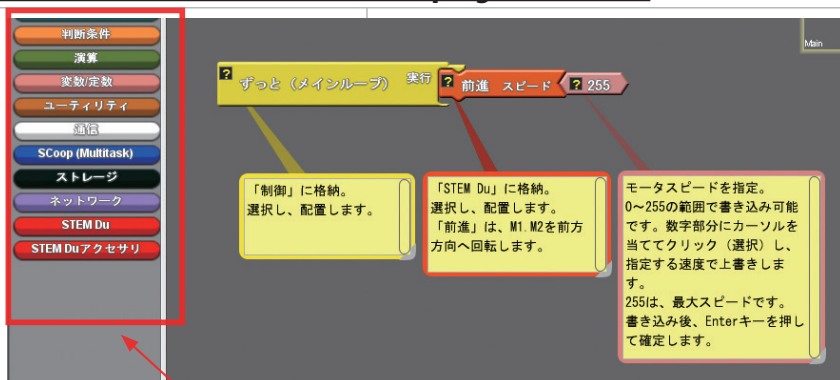
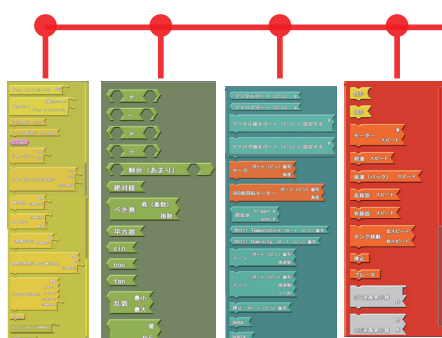
7. ArduBlock がスタートします。



7. ArduBlock starts.

15.6.3. 動作テストプログラムを作成します。

test program is made.



- 上記の図を参考にしながら、前進するプログラムを作成します。
 ・例は、「前進」を使っていますが、他の動きでテストを行なうこともできます。
 ・モータースピードは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、半分くらいのスピードを指定してみます。

アイコンパレットにプログラムブロックが入っています。
 An icon palette contains a program block.

- A figure above-mentioned is consulted, Forward, a program is made.
 * An example uses "forward", 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 will designate the speed which is about half.

- 「Arduino へアップロード」します。



- "It's uploaded to Arduino." it's done.

- Arduino-IDE では、コンパイルを行い実行ファイル化して、マイコンボードに書き込みます。



- Arduino - IDE compiles, becomes an executable file and writes in a microcomputer board.

- 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。



- 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.

***マイコンボードは、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。電池ボックスのレバースイッチを電源ON/OFFに使います。

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

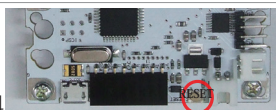
***When microcomputer board is turned on, a program will be executed immediately. When power supply switch is turned on in other words, movement of a robot begins it. Even if I move, a robot puts it in the safe location and makes them begin to move.

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

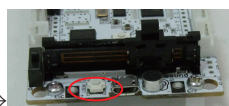
15.6.4. トラブルシューティング

【troubleshooting】

| | |
|--|---|
| 1 Windowsで、使用中にコムポートの認識が外れる。 (Windows起因の現象) | Recognition in a com port comes off during use by Windows. (Phenomenon of Windows cause) |
| OS(Windows7, 8, 10)にもよりますが、おおむね下記の手順で解決することがあります。 | It also depends on OS (Windows7, 8, 10), it's settled by the following procedure for the most part. |
| ①. 使用中のアプリを、いったん閉じて再立ち上げてみてください。 | Please close and restart the application which is being used once. |
| ②. 使用しているマイコンボードのResetボタンをダブルクリックしてみてください。 | Please double-click a Reset button of the microcomputer board which is being used. |
| ③. USBケーブルを接続したまま、再起動してみてください。 | Please restart while connecting a USB cable. |
| ④. 書き込み成功したときに、直ぐに2度目の重ね書きをしてください。PCの通信ポート接続の認識力が高まる場合があります。(コムポート認識が外れにくくなる場合があります) | When writing and succeeding, please do the 2nd time of stack describing immediately, The recognition power of the communication port connection of a PC sometimes rises. (Recognition in a com port sometimes becomes difficult to come off.) |



RDC-104



RDC-103

| | |
|--|---|
| 2 コムポートを認識しなくなった。 Windows起因のトラブルではなく、コントローラの能力を超える大きなプログラムを書き込む、通信中にUSBケーブルを抜き差しして電気ショックを与えたなど、無理な使い方により搭載マイコンが対応できずに処理機能を果たせなくなりCOMポート認識ができなくなる場合があります | A com port wasn't recognized any more. By the how to use which is impossible such as inserting and removing a USB cable to whole communication in which a big program beyond the ability of the controller is written, not trouble of Windows cause and giving electronic shocks, a equipped microcomputer, it can correspond, without, the process function can't be achieved any more and communication port recognition can't sometimes be done any more. |
| この場合のArduino画面のメッセージ | Message of this case's Arduino screen |

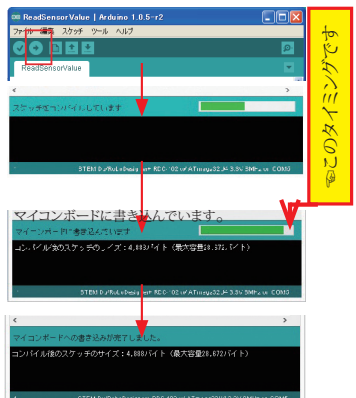
ポート認識不
成立/容量
サイズ越え
エラー

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.

コンパイル後のスケッチのサイズ: 〇〇,〇〇〇バイト (最大容量28,672バイト) processing.app.debug.Runner-Exception: 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.



| | |
|---|---|
| 最大容量以上のプログラムを書き込むことはできません。エラーを起こし、通信ができなくなります。(COMポート認識が外れます) | It isn't possible to write in a program beyond the maximum capacity, You can't make an error and communicate any more. (Com port recognition comes off.) |
| 【対応策】 次の手順により、回復することがあります。 | [Countermeasure] You get back by the next procedure. |
| 最大容量より、小さいサイズのプログラムを下記の方法で書き込みます。 | A program with size smaller than the maximum capacity is written in by the following way. |
| プログラムは、サンプルプログラムを使う方が妥当でしょう。(プログラムのタイプミスなどの要素がない方がテストに適しているが理由です) | The one using a sample program will be proper for a program, (The person who doesn't have an element of a type mistake of a program is suitable for a test, but, it's a reason.) |
| Arduino>ファイル>スケッチの例>01. Basics>Blinkを読み込みます。(プログラムサイズはBlinkで5,000バイトくらいです) | Arduino> file >Example of Sketch >01. Basics>Blink is read. (The program size is Blink, about 5000 bytes.) |
| USBケーブルを接続します(この場合COMポートの認識はしていなくてよいです) | A USB cable is connected (You don't have to recognize a com port in this case.) |
| 「マイコンボードに書き込む」ボタンを押してコンパイルが進行し、 マイコンボードに書き込みが始まった直後(書き込み完了前)に、コントローラのRESETスイッチをダブルクリックします。 強制アップロードです。 | <div><p>Compulsion upload.</p></div> <p>"It's written in a microcomputer board." just after compilation presses a button, and progresses, and writing in has started with a microcomputer board, in (before writing in completion). The RESET switch of the controller is double-clicked.</p> |
| 回復しない場合、USBケーブルを接続したまま再起動させると、COMポート認識が回復する場合があります。→この手順のよ うに、小さいサイズのプログラムを書き込みます。 | When you make them restart while connecting a USB cable, when not getting back, communication port recognition is sometimes convalescent, a program with small size is written in like this procedure. |



15.6.5. はじめて2輪テストロボットを動作スタート、各種動作点検

Test robot movement check

(1). 基板に転送したプログラムの実行 Execution of the program downloaded in a microcomputer board

| | |
|-------------------------------------|--|
| 1. ロボット動作事前準備確認 | Robot movement confirmation of preparations |
| a. ロボットの接続状態を確認するために実機を動かして点検を行います。 | a. production is moved and it's checked to confirm the state of the connection of the robot. |
| b. 下記の設定通りに入出力機器の接続がなされているかを確認します。 | b. It's confirmed whether an input/output device is connected to the following setting street. |

| 部 品 | microcomputer board | parts |
|----------------------|---------------------|---|
| タイヤ付ギアボックス 左側 | M1 | Left Gear Box with wheel |
| タイヤ付ギアボックス 右側 | M2 | Right Gear Box with wheel |
| 電池ケース(単3 × 4) | V1 | Battery holder AA battery x4 |
| 電池ケース(単3×4) 1個はダミー電池 | V2 | Battery holder AA battery x 3 and dummy 1 |

| | |
|---|--|
| c. 電池が入っている事を確認します。 | c. It's confirmed to include a battery. |
| ・V1接続電池ホルダー ⇩ V1 connection battery holder AA battery x 4  | ・V2接続電池ホルダー ⇩ V2 connection battery holder AA battery x 3 Dummy battery x1  |
| 単3アルカリ乾電池4本 | 単3アルカリ乾電池3本とダミー乾電池1本 |

2. プログラム実行：

A program is executed.

| | | |
|---|---|---|
| [V1接続電池ボックスのレバースイッチ(SW)] を ON にします。→プログラムが実行されます。 |  | Turn on the [lever switch of] V1 connection battery holder (SW). → A program is executed. |
| ※ V2接続M3,M4動作 今回はOFFのままにします |  | V2 connection is for M3,M4 movement. It's done while being off this time. |

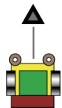
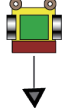
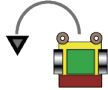
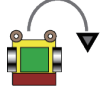


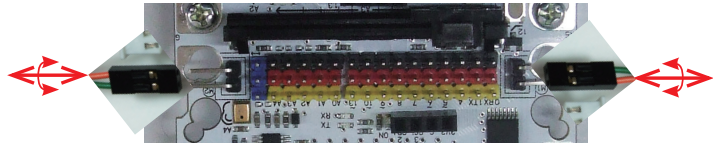
3. プログラム停止： [電池ボックスのスイッチ(SW)] をOFF にします。

Stop the program: Power supply sw is cut.

(2). 初めての2輪ロボット動作点検

(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、注意して観察します。

| 前進する Forward | 後進する Backward | 左旋回 Left pivot turn | 右旋回 Right pivot turn | 少しカーブする Little turn |
|---|---|--|--|---|
|  前進 |  後進 |  左回転 |  右回転 |  カーブする |
| 正しく動いています。 | M1,M2 配線の極性(プラス・マイナス)接続間違い | M1 モータ配線の極性(プラス・マイナス)接続間違い | M2 モータ配線の極性(プラス・マイナス)接続間違い | 正しく動いています。*1 |
| It's moving right. | M1,M2 Polarity (plus minus) connection mistake of wiring. | M1,M2 Polarity (plus minus) connection mistake of wiring. | M1,M2 Polarity (plus minus) connection mistake of wiring. | It's moving right. |
| そのまま使います。 | 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)、接続してください。 | 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)、接続してください。 | 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)、接続してください。 | 左右のモータ個体差の影響で、許容範囲内です。 |
| It's connected right. You'll use it just as it is. | Replace a ± connection of the left and right side motor and connect again. | Replace a ± connection of the left side motor and connect again. | Replace a ± connection of the right side motor and connect again. | It's moving right. It's influence of the motor individual difference in the left and right and is in the latitude.*1 |
|  |  | | | プレートにタイヤが接触していないかを確認してください。擦れているとブレーキをかけた状態になり、正しく回転しくなります。 |
| | | | | Please confirm whether a tire doesn't touch a basic plate, it can't be in the state to which the brakes was hit and rotation right. |

*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ずつ曲がっていきます。

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

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

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

*1. The individual difference in the motors of left and right is influential and is that it's very difficult to advance the long distance straight.

* 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. It's possible to get direct advance by using an encoder motor and controlling a rotation in left and right.

15.6.6 車輪の動き・ロボットの動きとプログラム

Movement of a wheel and programming



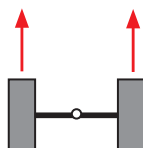
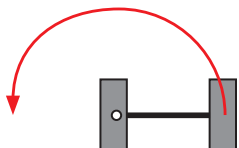
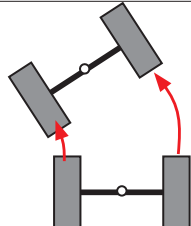
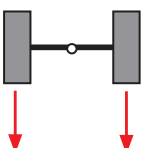
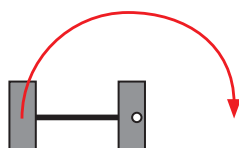
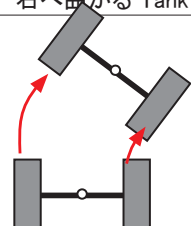
円周率 [The circular constant; pi (sinbol: π)]
 $\pi \approx 3.14159\ 26535\ 89793\ 23846\ 26433\ 83279\ 50288\ \dots$

66mm×円周率(π) \approx
1回転 207mm

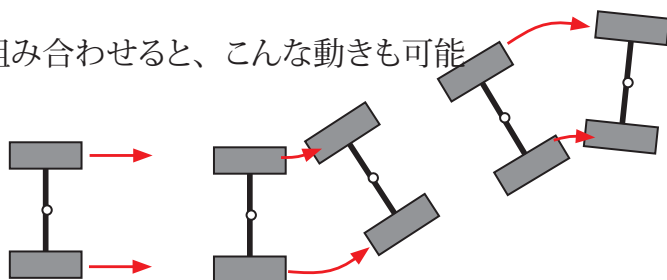
タイヤ回転数
150rpmの時
1分間に
3,105mm

分速 3 m/m
時速 180m/h

| | |
|--|--|
| 車はタイヤが一回転すれば、タイヤの直径×円周率だけ進みます。66 mm×円周率 \approx 207 mm | When a tire does a car one rotation, only the diameter of a tire x pi is developed. 66 mm x pi \approx 207mm |
| 左右の車輪の回転速度を制御することで、ロボットを前進させたり、旋回させたりすることができます。 | You make a robot move ahead by controlling the rotating speed of the wheel in left and right, and it's possible to make them circle. |
| 片輪だけが回転すれば旋回します。 | When only a one loop circulates, it circles. |

| 前進 Forward | 左旋回 Left pivot turn | タンク移動 左へ曲がる Tank left turn |
|---|--|---|
|  <p>左右のタイヤが同じスピードで回転 A tire in left and right rotate by the same speed.</p> <p>前進 スピード 255</p> <p>STEMDU_robot.forwardM1M2(255)</p> |  <p>左を停止、右側だけが回転 The left is suspended. Only the right side rotate.</p> <p>左旋回 スピード 255</p> <p>STEMDU_robot.leftM1(255)</p> |  <p>左右の異スピードの回転差で曲がる It bends by the rotation difference in the different speed of the left and right.</p> <p>タンク移動 左スピード 30 右スピード 150</p> <p>スピード設定は 10 以上 255 まで</p> <p>STEMDU_robot.tankM1M2(30,150)</p> |
| 後進 Backward | 右旋回 Right pivot turn | タンク移動 右へ曲がる Tank right turn |
|  <p>左右のタイヤが同じスピードで回転 A tire in left and right rotate by the same speed.</p> <p>後進 (バック) スピード 255</p> <p>STEMDU_robot.backwardM1M2(255)</p> |  <p>右を停止、左側だけが回転 The right is suspended. Only the left side rotate.</p> <p>右旋回 スピード 255</p> <p>STEMDU_robot.rightM2(255)</p> |  <p>左右の異スピードの回転差で曲がる It bends by the rotation difference in the different speed of the left and right.</p> <p>タンク移動 左スピード 255 右スピード 50</p> <p>スピード設定は 10 以上 255 まで</p> <p>STEMDU_robot.tankM1M2(255,50)</p> |

組み合わせると、こんな動きも可能



ArduBlockでのスピードPWM設定

前進 スピード 255

後進 (バック) スピード 255

⚠ モータ固体速度より速くは設定できません。255が限界値です。

タンク移動

左スピード 30

右スピード 150

| ブロックのスピード設定 | 255 | 150 | 100 | 50 | -255 | -150 | -100 | -50 |
|----------------|------|-----|-----|-----|------|------|------|-----|
| モータ速度へのデューティ比: | 100% | 60% | 40% | 20% | 100% | 60% | 40% | 20% |
| 回転方向⇒ | CW | CW | CW | CW | CCW | CCW | CCW | CCW |

⚠ DCモータギアボックス固体性能の回転速度に対してのデューティ比です。

ギアボックスでのギア比(減速比)・使用モータの回転数などにより、違いがありますが、出力回転数で130～200RPMがあります。



ロボットの動きチェック

実行

タンク移動

左スピード 255

右スピード 255

ミリ秒待つ

ミリ秒 100

・この命令をロボットへ転送して動作を確認します。
上記動作にならない時は、モータケーブルなど配線が違う場合がありますので、⊕ ⊖ チェックをします。
* You forward this order to a robot and make sure of movement, when it isn't the above movement, when wiring is different, a motor cable is here, ⊕ ⊖ It's checked.

15.6.7. ライントレースのプログラムに必要な分岐条件パラメータ「しきい値」を調べる。

赤外線センサを床に反射させ、床からの反射赤外線強度を測定して、赤外線量の強弱に基づき、マイコンの処理を分岐させる分岐条件「しきい値」の設定がプログラミングに重要な要素です。

• Infrared rays are applied to a floor and the strength of the reflective infrared rays from a floor is measured. Setting of the branch condition which makes processing of a microcomputer branch "threshold value" is an element important to a programming based on the strength of the infrared amount.

センサを指定して arduBlock のリアルモニターの命令プログラムを実行することで接続しているセンサのデータが計測できます。

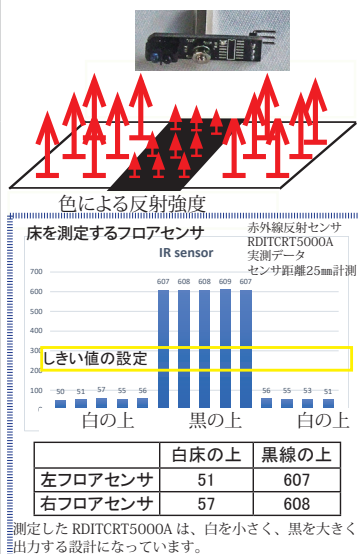


左図は、機体搭載フロアセンサ RDI-TCRT5000A の出力、実験室での測定値です。

動作環境下でセンサ出力を計測して "しきい値" を適切に設定します。このデータでは、「しきい値」は 200 位が良いでしょう。

プログラム文は
200 > センサ出力なら ○○ する。
でなければ △△ する。
条件分岐プログラムを作成します。

• 反射赤外線量は、室内照明違いや、太陽光などの差し込み、競技台紙の光沢違いによる反射度合い等により変化しますので、動作させる実環境で測定しプログラムに反映します。
• 測定データは表計算ソフト(例:EXCEL等)を使い、グラフ化すると分岐条件を判断しやすくなります。



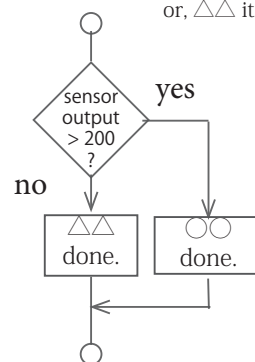
• Data of the sensor connected by designating a sensor and executing an order program of a serial monitor of arduBlock can be measured.

• Left figures are output of floor sensor RDI-TCRT5000A and the measure at the laboratory with a fuselage.

• Sensor output is measured under the working environment and "threshold value" is established appropriately. This data would be fine for about 200 for "threshold value".

A program statement?

Conditional branch program is made.
200 > sensor output is ○○ done.
or, △△ it's done.



ライントレース動作させる「モータ速度 (PWM)」を調べる

黒を感知した なら 停止する プログラム

しきい値200 > アナログセンサポート 1 (200より小さいセンサ出力) なら前進する。

でなければ右旋回する。

と条件分岐プログラムで、ライントレースを行っても、なかなかうまくいきません。

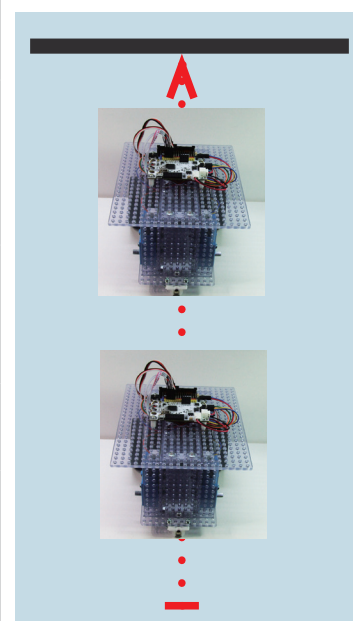
多くの場合モータスピードとしきい値のバランスが悪く、センサが計測したデータをマイコンボードが処理しモータへ動きを指示する間に、ロボットがさらに先へ動いてしまい、センサ値が変化してしまうことがあります。

現場環境で計測し、分岐条件(しきい値)の設定を決定したら、その「しきい値」を使って「黒を感知した なら 停止する プログラム(上記ブロックプログラム)」を作成し、ロボットで黒帯線に向けて動作させます。

スピード設定(PWM値)を変化させながら、黒帯線に差し掛かったらピタッと止まる動作を実験で見つけます。(赤外線反射センサが黒帯線に差し掛かった止まる位置です。)

この時のスピード設定値が、ロボットがライントレースするベストのスピード設定(PWM値)です。

動作時の床の滑り具合、電池残量の違いなどによっても変化しますので、うまくライントレースができない時は、この「黒を感知した なら 停止する プログラム(上記ブロックプログラム)」で実走させ、黒帯線にピタッと止まるかを確認し、調整してください。



The most suitable motor speed is checked by a practice.

Measure by a site and decide about setting of a branch condition (threshold value).

The programming which "is suspended" if black was sensed, is made using the "threshold value".

Make them move to the black belt line.

I maneuver while changing speed setting (PWM value). Duty ratio of the speed which stops at a black belt line is found.

(A infrared reflectiveness sensor is the location which stops at a black belt line.)

The speed set value of that time is the best speed setting by which a robot traces a line (PWM value).

It also changes by the slipping condition of the floor when moving, and the difference in the battery remaining amounts, so please adjust it as when linear trace can't be done well, I make them move by this program, and it may stop at a black belt line.

15.7. ライトレースロボのプログラム

15.7.1 条件分岐プログラムのアルゴリズム

Algorithm of Conditional branch program

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、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るでしょう。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でしょう。

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. マイコンボードは、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。

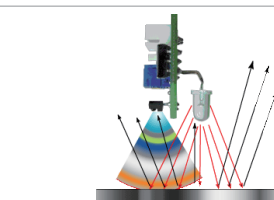
・ロボットが動く環境下で計測して、白色 / 黒色それぞれの反射光の強さを計測します。センサ値計測のソフトを使ってマイコンボードで測ります。

・白色からと、黒色からの、それぞれの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。

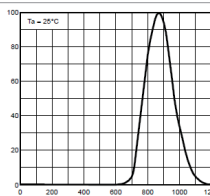
エッジ走行・・・境界の縁をたどります
分かりやすい一言で説明すると「エッジ走行」です。

ロボットは、直線だけでなく、曲線でも進めます。
フロアセンサ 2 個でライン両側のエッジ（白と黒の境界）を辿ります。左曲がり、右曲がり曲線をトレースできます。

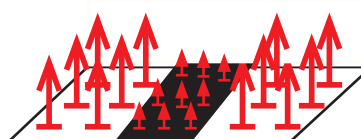
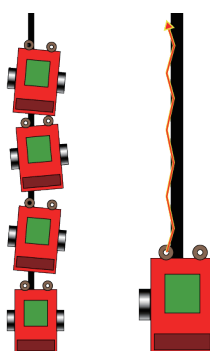
センサが 1 つの場合は、一方向に曲がる曲線しか進めません。



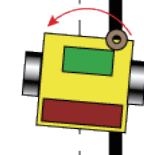
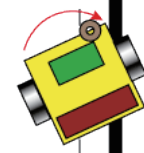
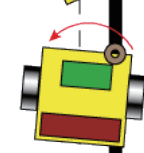
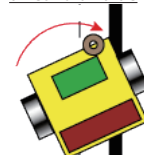
IRセンサ原理図



IRセンサ特性図



色による反射光強度
Light reflection intensity with the color



1 sensor 2 sensor

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. Microcomputer board has an LED for emission flight 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 microcomputer controller board rotates 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.

The respective catoptric light data is measured from black from white.

You decide about the middle of the mean as a threshold value (branch condition). It's written in a program.

Edge run...it's followed in case of a boundary. When it's explained in a plain word, it's "edge run".

Tracer can follow a curved line as well as a straight line.

An edge on linear both sides (white and black boundary) is followed by 2 floor sensors.

The left curve and the right curve curved line can be traced.

A sensor can trace one in case of 1 to only the curved line which bends in one-way.

15.7.2. フロアセンサ 2 個使用での ArduBlock プログラミング

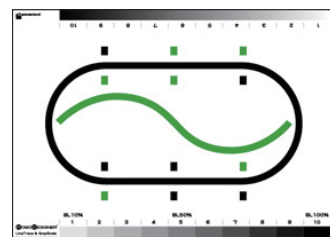
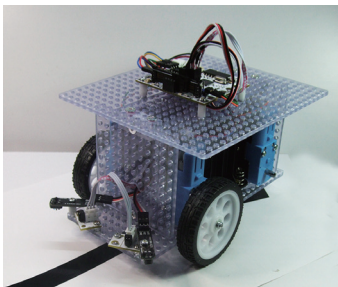
[2]. Programming of linear trace

1. 取り出せるセンサ出力数値が、光の反射率により、白色と黒色で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。



↑ ライントレースシート (A1 サイズ)

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.



ArduBlock ライントレースプログラム作成……… しきい値は 200 で作成します。

Create ArduBlock line trace program…… Create a threshold value of 200.

上図を参考に、プログラムを作成し、マイコンボードへアップロードします。

Create a program and upload it to microcomputer board by referring to the above figure.

分岐条件「しきい値」を 200 に設定します。

Set the branch value threshold to 200.

テスト動作実験室での測定で上記表の本機に使用しているアナログ赤外線反射センサ RDI-TCRT5000A は、白床の上で 51 ほどの値、黒線の上では 607 ほどの測定値であったので、分岐条件は「しきい値」を 200 に設定してサンプルを作成。

下記サンプルプログラムは、機体搭載フロアセンサ RDI-TCRT5000A の出力値を、以下で仮定しています。

The analog infrared reflection sensor RDI-TCRT5000A used in this machine in the above table for measurement in the Lab had a measured value of about 51 on the white floor and about 607 on the black line. The program sets a "threshold" of 200 set to create a sample.

| (実験室計測値) | | |
|----------|------|------|
| | 白床の上 | 黒線の上 |
| 左フロアセンサ | 51 | 607 |
| 右フロアセンサ | 57 | 608 |

動作環境下でのセンサ出力を計測して「しきい値」を適切に設定します。

実動テストの際には、実際使用環境下でのラインレース実施場で、フロアセンサのデータを計測して、計測データの分析で「しきい値」を決定します。環境の明るさ、ラインを書き込んだ紙の光沢状況などで計測データが変動しますので、使用環境下での計測に基づき、プログラムを作成します。

During the actual operation test this time, the floor sensor data will be measured at the line trace implementation site under the actual usage environment, and the "threshold value" will be determined by analysing the measurement data. Since the measurement data varies depending on the brightness of the environment and the glossiness of the paper on which the lines are written, create a program based on the measurement under the usage environment.

| | |
|---|--|
| <p>arduino > ファイル > サンプルの例 > 01.Basics > AnalogReadSerial を開いて、マイコンボードに書き込んで、A0 と A1 に接続したアナログ赤外線反射センサを、白床、黒線でそれぞれ計測して、上記表のような計測値をまとめる表に作り、その中で、しきい値を考察してください。</p> <p>条件分岐を決める値ですので、重要な条件設定です。シリアルモニター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。</p> <p>・ 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【RDC-104_linetrace_2sensor_test.abp】で格納されています。</p> <p>・ 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。</p> <p>シリアルモニター命令ブロックを使うことでも、センサーデータの取得が可能です。</p> | <p>arduino > File > Sample example > 01. Basics > Open analog read serial and write on the microcomputer board measure the analog infrared reflection sensor connected to A0 and A1 with the white floor and the black line respectively.</p> <p>Make a table that summarises the values such as a memorandum and consider the threshold in it.</p> <p>It is important condition setting because it is a value that determines the conditional branch. When make copy measured data from a cereal monitor, and the paste (Ctrl + V) is made spreadsheet software and graphing is processed, the tendency value can be grasped, and it becomes capable of turning point (threshold value) consideration.</p> <p>* Source cord is stocked in the sample folder arranged to " PC/MyDocuments/Arduino/ [ArduBlock Examples] " by the file name [RDC-104_linetrace_2sensor_test.abp].</p> <p>* It can be confirmed that the sample arranged in PC/ MyDocuments/Arduino/ArduBlock Examples is opened in Arduino/ArduBlock.</p> <p>It's able to acquire sensor data even to use a sirial monitor block.</p> |
|---|--|



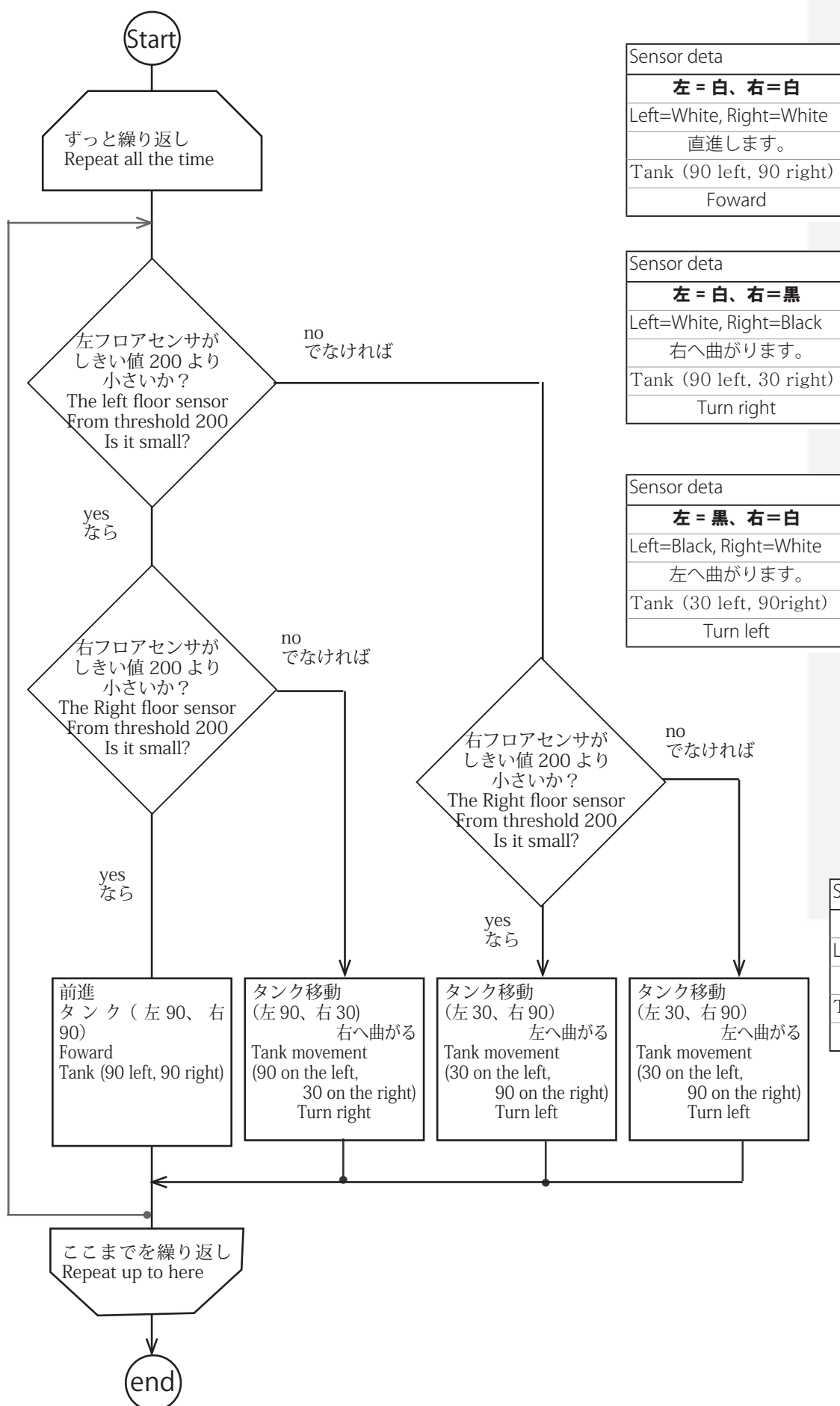
今回のプログラムのアルゴリズム

Algorithm of this program

| 条件分岐 1 Conditional branch 1 | 条件分岐 2 Conditional branch 2 | 判断 Judgement | 処理 Processing |
|--|--|--|---|
| もし、アナログポート A0 の入力 (左フロアセンサー出力) が「しきい値」以下の時 (白の時) | アナログポート A1 の入力 (右フロアセンサー出力) が「しきい値」以下 (白の時) | なら 左 = 白、右 = 白 If Left = white, right = white | 直進します。(左PWM90、右PWM90) Go straight. (Left PWM90, right PWM90). |
| If the input of analog port A0 (left floor sensor output) is When it is less than or equal to the key value. (When white). | The input of analog port A1 (right floor sensor output) Below "threshold". (When White). | でなければ 左 = 白、右 = 黒 Otherwise Left = white, right = black | タンク移動 (左PWM90、右PWM30) をします。 (右へ曲がる) Tank movement (left PWM90, right PWM30). It was (Turn right). |
| もし、アナログポート A0 の入力 (左フロアセンサー出力) が「しきい値」以下でなければ (黒の時) | アナログポート A1 の入力 (右フロアセンサー出力) が「しきい値」以下 (白の時) | なら 左 = 黒、右 = 白 If Left = black, right = white | タンク移動 (左PWM30、右PWM90) をします。 (左へ曲がる) Tank movement (left PWM30, right PWM90). It was (Turn left). |
| If the input of analog port A0 (left floor sensor output) is Unless it is less than or equal to the key value. (When black). | The input of analog port A1 (right floor sensor output) Below "threshold". (When white). | でなければ 左 = 黒、右 = 黒 Otherwise left = black, right = black | タンク移動 (左PWM30、右PWM90) をします。 (左へ曲がる) Tank movement (left PWM30, right PWM90). It was (Turn left). |

アクティビティ図

Activity diagram

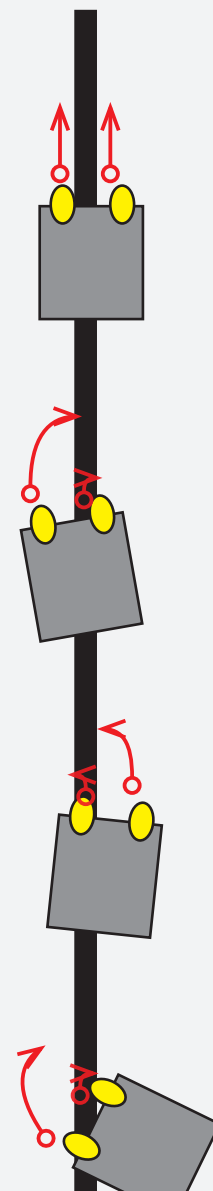


| Sensor deta | |
|--------------------------|--|
| 左 = 白、右 = 白 | |
| Left=White, Right=White | |
| 直進します。 | |
| Tank (90 left, 90 right) | |
| Foward | |

| Sensor deta | |
|--------------------------|--|
| 左 = 白、右 = 黒 | |
| Left=White, Right=Black | |
| 右へ曲がります。 | |
| Tank (90 left, 30 right) | |
| Turn right | |

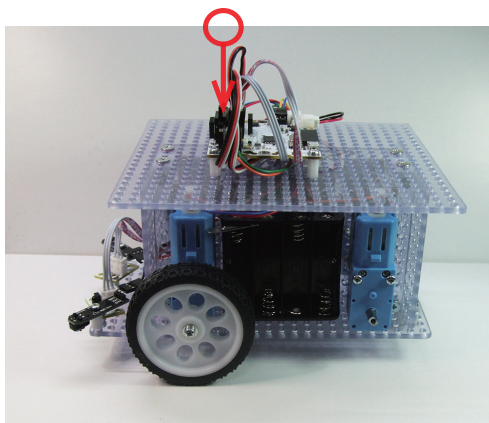
| Sensor deta | |
|-------------------------|--|
| 左 = 黒、右 = 白 | |
| Left=Black, Right=White | |
| 左へ曲がります。 | |
| Tank (30 left, 90right) | |
| Turn left | |

| Sensor deta | |
|--------------------------|--|
| 左 = 白、右 = 黒 | |
| Left=White, Right=Black | |
| 右へ曲がります。 | |
| Tank (90 left, 30 right) | |
| Turn right | |

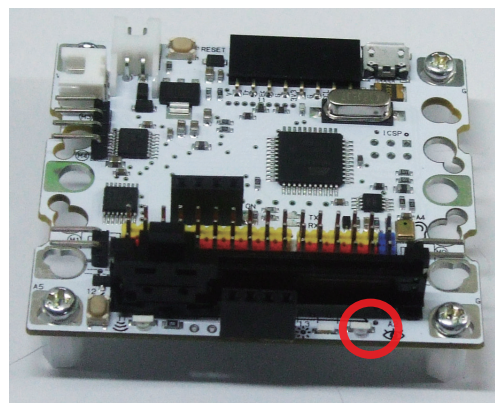


15.8. 明るさに反応するロボットを作る。

Create the robot which reacts to the brightness.



<全体図 Top view>



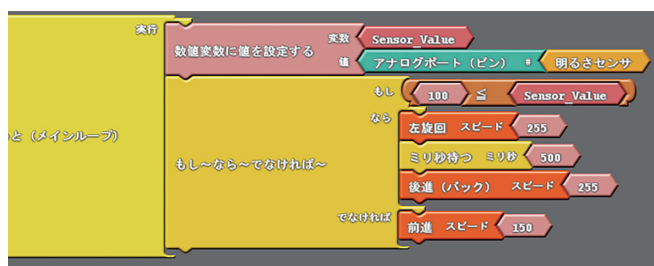
<センサ位置 Sensor location>

ボード搭載の明るさセンサを使用する時は、
明るさセンサ位置のピンと A2 信号ピンをジャンパー
ケーブル 1 本で接続し回路設定をします。

When a brightness sensor of board loading is used.
In the brightness sensor location, A2 signal pin is
connected by a of jump cable and circuit setting is
done tightly.

The motor speed is made a low-speed forward according to the data of a brightness sensor and it's made back of high-speed.

明るさセンサの値に応じてモータを低速前進したり、高速後進をしたりします。



使用コネクタ 明るさセンサ

RDC-104: (A2)/(M1)/(M2)

RDC-ESP32: (A6)/(M1)/(M2)

変数 sensor_value 搭載光センサの出力値を格納します。
分岐分岐のしきい値を変えると反応する明るさが変わります。

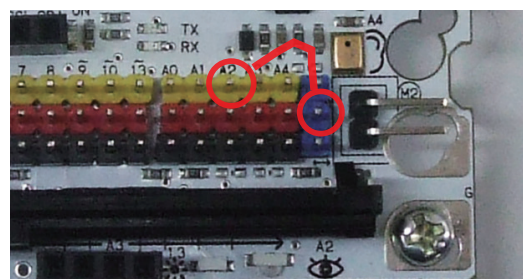
動作手順

- ・周囲の明るさがしきい値(100)より小さいとき 150のスピードで前進します。
- ・でなければ 255のスピードで後進します。

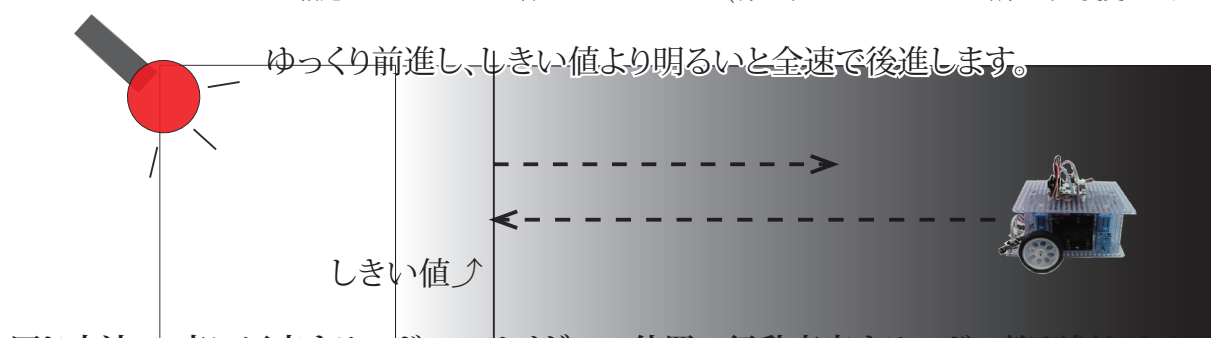
※作成したプログラムは、「名前をつけて保存」します。

作成完了後、「Arduino ヘアアップロード」して、マイコンボードへ書き込みます。

■Arduino Cのサンプルプログラムは、[Arduino] > [ファイル-スケッチの例-01. Basics-AnalogReadSerial]を開いて、アナログセンサの指定をマイコンボードに合わせてRDC-104=A2(明るさ) RDC-ESP32=A6(明るさ)に変更してください。



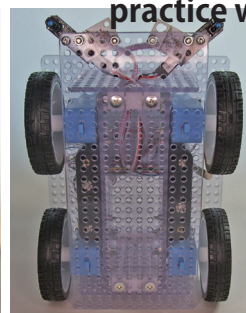
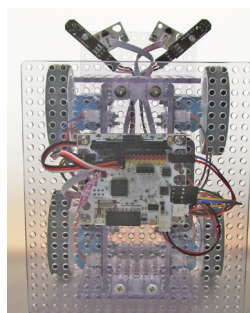
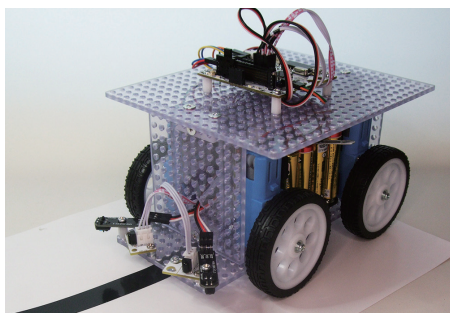
| | 搭載センサ On board sensor | 裏面表記 Print symbol to the back | 接続ピン Pin No. |
|---|-----------------------------|----------------------------------|-----------------|
| 搭載センサ 回路 接続表 Sensor connection table | マイク Mic | | A4 |
| | 明るさセンサ Brightness sensor | | A2 |
| | スライダ Slider | | A3 |



同じ方法で、音に反応するロボット スライダーの位置で行動変容するロボット等が創れます。

15.9.4 輪ロボのライトレースプログラム

15.9.1 4WD ロボットを制作します。



How many branch condition value practice way

15.9.2. 分岐条件「しきい値」を調べる。

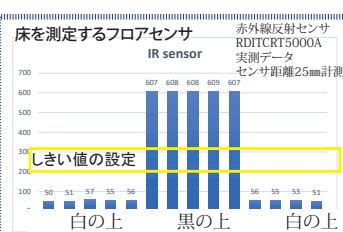
赤外線センサを床に反射させ、床からの反射赤外線ของความ強さを測定して、赤外線量の強弱に基づき、マイコンの処理を分岐させる分岐条件「しきい値」の設定がプログラミングに重要な要素です。

• Infrared rays are applied to a floor and the strength of the reflective infrared rays from a floor is measured. Setting of the branch condition which makes processing of a microcomputer branch "**threshold value**" is an element important to a programming based on the strength of the infrared amount.

センサを指定して arduBlock のリアルモニターの命令プログラムを実行することで接続しているセンサのデータが計測できます。

右図は、機体搭載フロアセンサ RDI-TCRT5000A の出力、実験室での測定値です。

動作環境下でセンサ出力を計測して「しきい値」を適切に設定します。このデータでは、「しきい値」は 200 位が良いでしょう。



| | 白床の上 | 黒線の上 |
|---------|------|------|
| 左フロアセンサ | 51 | 607 |
| 右フロアセンサ | 57 | 608 |

測定した RDI-TCRT5000A は、白を小さく、黒を大きく出力する設計になっています。

• Data of the sensor connected by designating a sensor and executing an order program of a serial monitor of arduBlock can be measured.

• Left figures are output of floor sensor RDI-TCRT5000A and the measure at the laboratory with a fuselage.

• Sensor output is measured under the working environment and "threshold value" is established appropriately. This data would be fine for about 200 for "threshold value".

ライトレース動作させる「モータ速度 (PWM)」を調べる 黒を感知した なら 停止する プログラム

しきい値200>アナログセンサポート 1 (200より小さいセンサ出力) なら前進する。でなければ右旋回する。と条件分岐プログラムで、ライトレースを行なっても、なかなかうまくいきません。

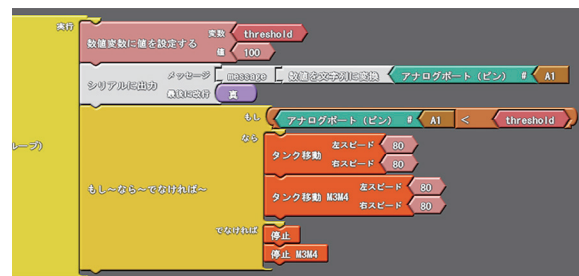
多くの場合モータスピードとしきい値のバランスが悪く、センサが計測したデータをマイコンボードが処理しモータへ動きを指示する間に、ロボットがさらに先へ動いてしまい、センサ値が変化してしまうことがあります。

動作環境で計測し、分岐条件(しきい値)の設定を決定したら、その「しきい値」を使って「黒を感知した なら 停止する プログラム(上記ブロックプログラム)」を作成し、ロボットで黒帯線に向けて動作させます。

スピード設定(PWM値)を変化させながら、黒帯線に差し掛かったらピタッと止まる動作を実験で見つけます。(赤外線反射センサが黒帯線に差し掛かった止まる位置です。)

この時のスピード設定値が、ロボットがラインレースするベストのスピード設定(PWM値)です。

動作時の床の滑り具合、電池残量の違いなどによっても変化しますので、うまくライトレースができない時は、この「黒を感知した なら 停止する プログラム(上記ブロックプログラム)」で実走させ、黒帯線でピタッと止まるかを確認し、調整してください。



The most suitable motor speed is checked by a practice.

Measure by working environment and decide about setting of a branch condition (threshold value).

The programming which "is suspended" if black was sensed, is made using the "threshold value".

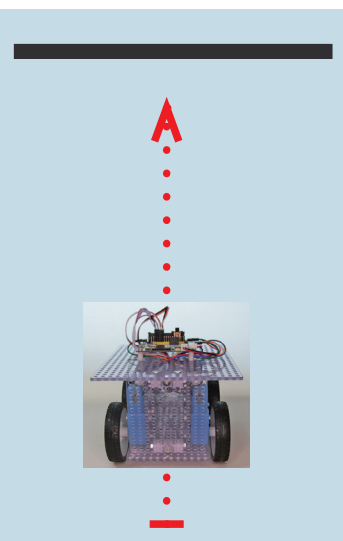
Make them move to the black belt line.

Change speed setting (PWM value), Duty ratio of the speed which stops at a black belt line is found.

(A infrared reflectiveness sensor is the location which stops at a black belt line.)

The speed set value of that time is the best speed setting by which a robot traces a line (PWM value).

It also changes by the slipping condition of the floor when moving, and the difference in the battery remaining amounts, so please adjust it as when linear trace can't be done well, I make them move by this program, and it may stop at a black belt line.



黒帯線は、黒色の電工用絶縁テープ幅19mm又は梱包用ガムテープ幅50mmを使います。

15.9.3 4WD ロボ車輪の動きとプログラム

Movement and programming of a 4WD robot wheel

一般的な4輪構成では、車輪が取り付けられているホイールベースの対角線にほぼ直角に力を加えます。

The power is added to a diagonal line of the chassis in which a wheel is installed perpendicularly mostly by a general four-wheel system.

各車輪の回転速度と回転方向を変えることで任意の方向に移動し、回転させることができます。

It's possible to move to the optional direction by changing the rotating speed and the direction of rotative of each wheel and make them rotative.

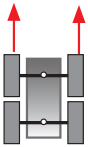
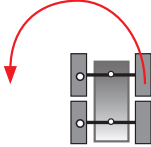
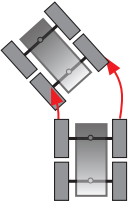
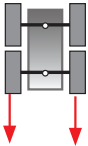
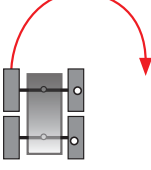
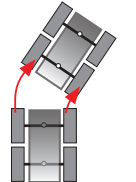
4つのモータを全て同じ方向に動かすと、前後に動きます。一方の車輪を反対側の車輪と反対方向に動かすと旋回運動になります。

When all 4 motors are moved to the same way, it moves at the time. When one wheel is moved to the wheel opposite and an opposite direction, it'll be turning motion.

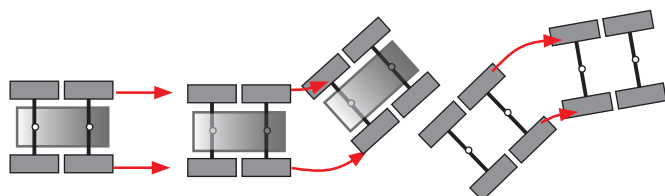
左右の車輪の回転速度を制御することで、ロボットを前進させたり、旋回させたりすることができます。片輪だけが回転すれば旋回します。

An advance can circle over a robot by controlling the rotating speed of the wheel in left and right.

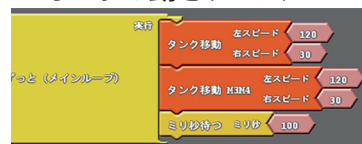
When only one wheel rotates, turn.

| 前進 forward | 左旋回 left turn | タンク移動 左へ曲がる tank |
|---|--|--|
|  <p>左右のタイヤが同じスピードで回転 A tire in left and right rotate by the same speed.</p> <div data-bbox="183 936 523 1055"> <p>前進 スピード 255</p> <p>前進 M3M4 スピード 255</p> </div> <p>STEMDU_robot.forwardM1M2(255) STEMDU_robot.forwardM3M4(255)</p> |  <p>左を停止、右側だけが回転 Only the right side suspends the left and rotate.</p> <div data-bbox="630 936 970 1055"> <p>左旋回 スピード 255</p> <p>左旋回 M3M4 スピード 255</p> </div> <p>STEMDU_robot.leftM1M2(255) STEMDU_robot.leftM3M4(255)</p> |  <p>左右の異スピードの回転差で曲がる It bends by the rotate difference in the different speed of the left and right.</p> <div data-bbox="1085 936 1425 1093"> <p>タンク移動 左スピード 30 右スピード 120</p> <p>タンク移動 M3M4 左スピード 30 右スピード 120</p> </div> <p>スピード設定は 10 以上 255 まで The speed setting is 10 to 255.</p> <p>STEMDU_robot.tankM1M2(30,120) STEMDU_robot.tankM3M4(30,120)</p> |
| 後進 backward | 右旋回 right turn | タンク移動 右へ曲がる tank |
|  <p>左右のタイヤが同じスピードで回転 A tire in left and right rotate by the same speed.</p> <div data-bbox="183 1597 523 1697"> <p>後進 (バック) スピード 255</p> <p>後進 (バック) M3M4 スピード 255</p> </div> <p>STEMDU_robot.backwardM1M2(255) STEMDU_robot.backwardM3M4(255)</p> |  <p>右を停止、左側だけが回転 Only the left side suspends the right and rotate.</p> <div data-bbox="630 1585 970 1704"> <p>右旋回 スピード 255</p> <p>右旋回 M3M4 スピード 255</p> </div> <p>STEMDU_robot.rightM1M2(255) STEMDU_robot.rightM3M4(255)</p> |  <p>左右の異スピードの回転差で曲がる It bends by the rotate difference in the different speed of the left and right.</p> <div data-bbox="1085 1585 1425 1742"> <p>タンク移動 左スピード 120 右スピード 30</p> <p>タンク移動 M3M4 左スピード 120 右スピード 30</p> </div> <p>スピード設定は 10 以上 255 まで The speed setting is 10 to 255.</p> <p>STEMDU_robot.tankM1M2(120,30) STEMDU_robot.tankM3M4(120,30)</p> |

組み合わせると、こんな動きも可能



ロボットの動きチェック



・この命令をロボットへ転送して動作を確認めます。
 上記動作にならない時は、モータケーブルなど配線が違う場合がありますので、⊕⊖チェックをします。

15.9.4. 4WD ライントレースの ArduBlock プログラミング

ArduBlock programming of 4WD linear trace.

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。

・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、格納されているファイル名【13_TEC36_2floorsensor_LineTrace.abp】に、M3/M4を追加した状態です。
・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。

Try to make a robot that can run through a truck made of big black lines at a good speed.

・ The source code is stored in the sample folder [ArduBlock Examples] located in PC/MyDocuments/Arduino/. M3 and M4 are added to the file name 【13_TEC36_2floorsensor_LineTrace.abp】
・ In Arduino/ ArduBlock you can check it by opening the sample placed in PC/MyDocuments/Arduino/ArduBlock Examples.

ラインレースプログラム作成

Make linear trace programming.

※前進の時には、[タンク移動]+[タンク移動 M3M4] 同スピードの組み合わせのほか、[前進]+[前進 M3,M4] の組み合わせで4輪前進をすることができます。

※左右のスピードに差をつけて、回転差で曲がりたい時には [タンク移動] を利用してください。

※動作の良し悪しは、パラメータ調整にかかっています。

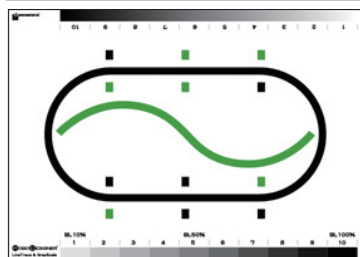
(配置した全てのブロックの各々の スピード PWM、しきい値の調整を繰り返し行ないます。)

※ When it's an forward, [tank movement] + [tank movement M3M4] it's possible to do a four-wheel forward by in addition to the combination of the identical speed and combination of [forward] + [forward M3,M4].

※ When you'd like to have the lead and make them bend by the rotate difference, please use [tank movement] for the speed of the left and right.

※ Good or bad of movement depends on parameter tuning.

Each speed PWM and threshold value in all arranged blocks are adjusted repeatedly.



・自作の場合 プラダン 90x180 cmに、黒色ビニールテープでラインのループを作成し競技場で使用します。

* When selfmaking, a linear circle is made by a black tape and it's made a field in a board of 180 cm x 90 cm.

ラインレース競技トラック例：RDP-971

プログラミングの上達は、演習計時で記録します。
Programming progress will be recorded at the exercise timekeeping

| | 1回 | 2回 | 3回 |
|---|-----|-----|-----|
| A | sec | sec | sec |
| B | sec | sec | sec |
| C | sec | sec | sec |

実機動作周回時間をストップウォッチで、計測し記録します。

The actual operation lap time is measured and recorded with a stop watch.

後日に演習した時も、同じ記録紙に書き込みます。

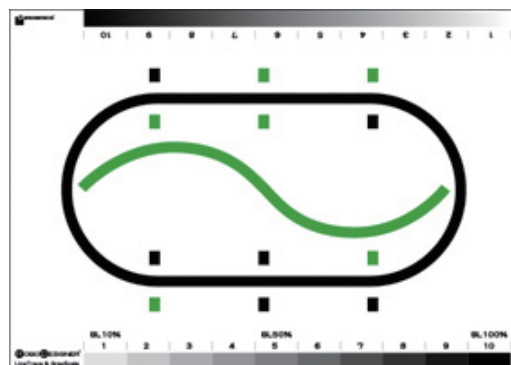
Look back by writing down the parameters during exercise in the remarks column.

備考欄に演習時のパラメータをメモすることにより、振り返りに役立てて、プログラミング力向上になります。

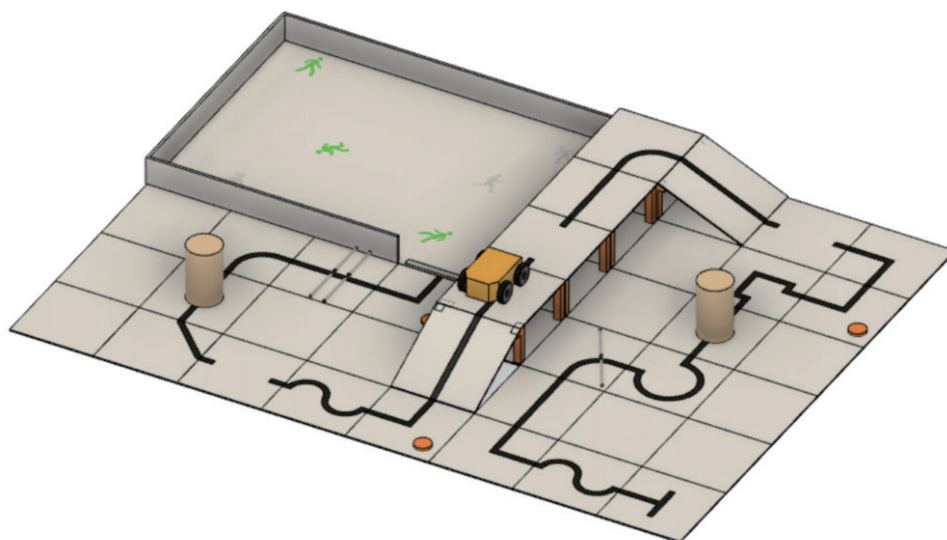
It will help you to improve your programming skills .

15.9.5 ライントレース競技例 Example of Line trace competition

ライントレース実験シート RDP-971 A 1 Size
Line trace experiment sheet



競技例 1. ロボカップジュニアレスキューリーグ；障害物、傾斜路、路面悪路、ライン切れ、被害者救出が課題
Competition E.g.: 1. Robocup Junior rescue league; obstacles, ramps, rough roads, broken lines, rescue of victims.

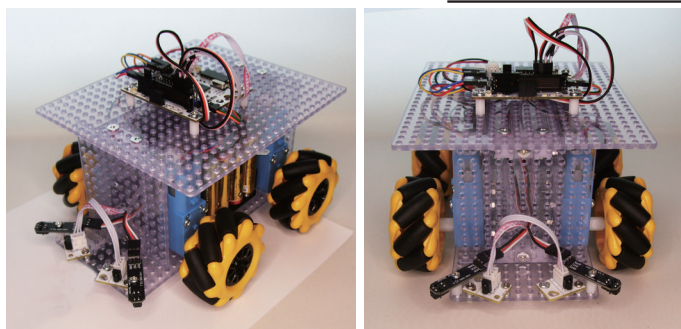


競技例 2. グループ対抗競技会場；障害物、ライン切れ、路面悪路が課題
Competition E.g.. 2. Group competition venue; obstacles, line breaks, rough roads are issue.



15.10. 全方向移動 4WD ロボ

Omni-directional movement 4WD Robot



メカナムホイールには方向性があります。
The Mecanum wheel is directional.

45 度樽 (パレル) の角度が左右対称に取り付けてあります。
The angle of the 45 degree rotor wheel is Symmetricalness.

左取付用⇒

For Left

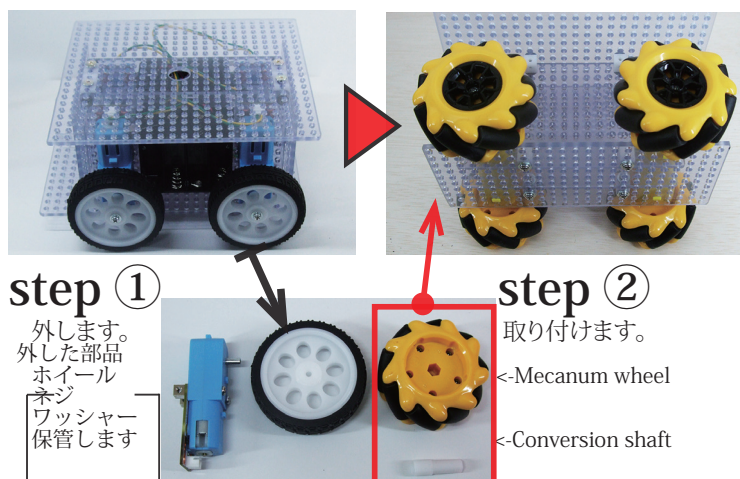


⇐右取付用

For Right



15.10.1. 全方向移動 4WD メカナムホイール取り付け



step ①

外します。
外した部品
ホイール
ネジ
ワッシャー
保管します

step ②

取り付けます。

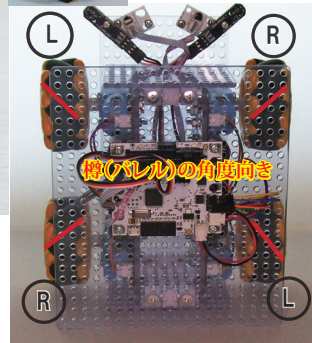
←Mecanum wheel

←Conversion shaft

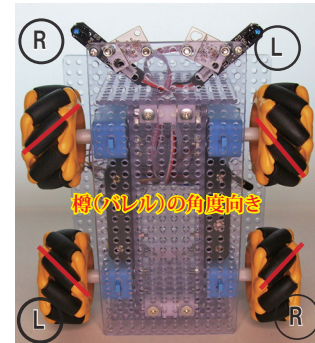


内側に刻印ⓁⓇがあります。

There is a stamp ⓁⓇ on the inside.



上から見た図
The figure seen from the top



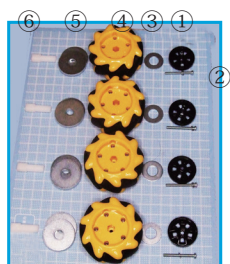
下から見た図
The figure seen from the underside

① 自律標準ホイールを外します。(外した部品は保管します)

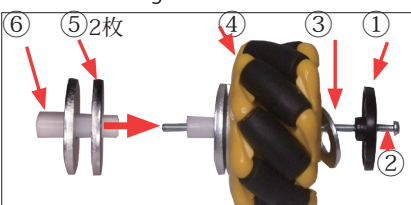
① Standard wheel is removed.
(A removed part is kept.)
Standard wheel 4pcs
Pan screw M3x15mm 4pcs

② 異形変換シャフトを差し込み、メカナムホイールを取り付けます。

② Insert the deformed conversion shaft and attach the Mecanum wheel.



ホイール組み図
Combination figure



| | | |
|------------------------|-----|----------------------------------|
| ①ホイールキャップ | 4個 | Rajin wheel cap Diameter30mm |
| ②M3 x 30ナベネジバネ座金付 | 4本 | M3x30mm pan screws 4pcs |
| ③丸ワッシャー-M10 | 4枚 | washer 11x22x1.5 4pcs |
| ④ホイールⓁⓇ | 各2台 | Mecanum wheel ⓁⓇ 2 each |
| ⑤丸ワッシャー-8.5 x 32 x 3.2 | 8枚 | washer 8.5x32x3.2 8pcs |
| ⑥異形変換シャフト | 4本 | Irregular conversion shafts 4pcs |

15.10.2. 電池が入っている事を確認します。

Make sure the batteries are installed.

| ・V1接続電池ケース V1connection battery case. | ・V2接続電池ケース V2connection battery case. | 電池入れ方違いあり | The difference by which a battery is how to put it in. |
|--|--|---|---|
| ↓ output: 6V | ↓ output: 4.5V | V1は、基板内の電子回路で電気を消費しながらモータ電力をM1,M2へ出力、V2はほぼ直接M3,M4へ出力しますので、M1～4へ供給する電気力を同じにする目的でV2電池ケース内にダミー電池を1本差し込みます。 | V1 is a motor power while consuming electricity in the electronic circuit in the board. The force is output to M1 and M2, and V2 is almost directly output to M3 and M4. So, V2 battery for the purpose of making the electric force supplied to M1 ~ 4 the same, Insert one dummy battery into the case. |
| 単3アルカリ乾電池4本 4 AA alkaline batteries . | 単3アルカリ乾電池3本 とダミー乾電池1本 3 AA alkaline batteries and 1 dummy battery. | 確認したい時は、基板のM1,M2,M3,M4の出力ピン端子を測定器(テスター)で計測して確認します。 | When you want to check, measure the output pin terminals of M1, M2, M3, M4 on the board with a measuring instrument (tester) and check. |



15.10.3 全方向移動 4WD メカナムホイールロボ車輪の動きとプログラム

Movement and programming of an omnidirectional movement 4WD Mecanum-wheel

メカナムホイールは、車輪の表面 (円周上) が車軸に対して 45°傾けた樽 (バレル) によって覆われています。

The surface of the wheel (on circumference) tilted mecanum wheel 45 deg to an axle, a barrel, it's covered by (barrel) up.

モーターの駆動力の伝達により、従来の車輪と同じ動きをすることに加え、円周上の樽がフリーになっているため 45°の方向に移動できます。

As well as doing the same movement as a conventional wheel by communication of drive force of a motor, on the circumference, a barrel, it becomes free, so you can move to 45 deg of direction.

4つのモーターの回転方向と速度制御を調整することで、車輪の回転と円周上の樽による動きとのコンビネーションを生み出し、全方向移動を実現します。

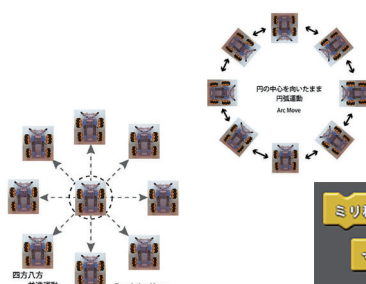
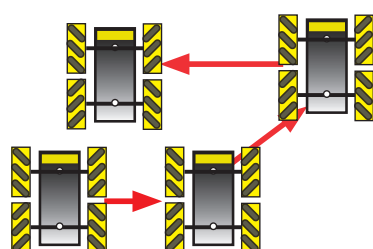
On the direction of rotation of 4 motors, the rotation which is to adjust speed control and a wheel and the circumference, a barrel, the combination with the chosen movement is invented and omnidirectional movement is achieved.

使用ギアボックスは減速比 1/90、出力シャフトの回転数は 130 ~ 140rpm (個体差あり) です。

A use gearbox is reduction gear ratio 1/90, and the number of rotations of the output shaft is 130-140rpm (An individual, it's partial.)

| 前進 Forward | 左旋回 Left turn | 左へ移動 Move left | 左斜め前へ移動 |
|--|--|--|--|
| <p>：車軸回転方向 : ベクトル ↑ : 移動方向</p> | <p>：車軸回転方向 : ベクトル ↑ : 移動方向</p> | <p>：車軸回転方向 : ベクトル ↑ : 移動方向</p> | <p>：車軸回転方向 : ベクトル ↑ : 移動方向</p> |
| タンク移動 Tank move M1,M2,M3,M4 が ⊕ 設定 左右が同じスピードで回転 | タンク移動 Tank move M1,M3 が ⊖ PWM 設定 M2,M4 が ⊕ PWM 設定 | タンク移動 Tank move M1,M4 が ⊖ PWM 設定 M2,M3 が ⊕ PWM 設定 | モーター単独設定 Motor independent M2,M3 を ⊕ 設定 M1,M4 は停止 |
| <p>タンク移動 左スピード 255 右スピード 255</p> <p>タンク移動 M3M4 左スピード 255 右スピード 255</p> | <p>タンク移動 左スピード -255 右スピード 255</p> <p>タンク移動 M3M4 左スピード -255 右スピード 255</p> | <p>タンク移動 左スピード -100 右スピード 255</p> <p>タンク移動 M3M4 左スピード 255 右スピード -100</p> | <p>モーター 2 スピード 255</p> <p>モーター 3 スピード 255</p> |
| スピード設定は 10 以上 255 まで STEMDU_robot.tankM1M2(255,255) STEMDU_robot.tankM3M4(255,255) | スピード設定は 10 以上 255 まで STEMDU_robot.tankM1M2(-255,255) STEMDU_robot.tankM3M4(-255,255) | スピード設定は 10 以上 255 まで STEMDU_robot.tankM1M2(-100,255) STEMDU_robot.tankM3M4(255,-100) | スピード設定は 10 以上 255 まで STEMDU_robot.motor(2,255) STEMDU_robot.motor(3,255) |
| 後進 Backward | 右旋回 Right turn | 右へ移動 Move right | 右斜め前へ移動 |
| <p>：車軸回転方向 : ベクトル ↑ : 移動方向</p> | <p>：車軸回転方向 : ベクトル ↑ : 移動方向</p> | <p>：車軸回転方向 : ベクトル ↑ : 移動方向</p> | <p>：車軸回転方向 : ベクトル ↑ : 移動方向</p> |
| タンク移動 Tank move M1,M2,M3,M4 が ⊖ 設定 左右が同じスピードで回転 | タンク移動 Tank move M1,M3 が ⊕ PWM 設定 M2,M4 が ⊖ PWM 設定 | タンク移動 Tank move M1,M4 が ⊕ PWM 設定 M2,M3 が ⊖ PWM 設定 | モーター単独設定 Motor independent M1,M4 を ⊕ 設定 M2,M3 は停止 |
| <p>タンク移動 左スピード -255 右スピード -255</p> <p>タンク移動 M3M4 左スピード -255 右スピード -255</p> | <p>タンク移動 左スピード 255 右スピード -255</p> <p>タンク移動 M3M4 左スピード 255 右スピード -255</p> | <p>タンク移動 左スピード 255 右スピード -100</p> <p>タンク移動 M3M4 左スピード -100 右スピード 255</p> | <p>モーター 1 スピード 255</p> <p>モーター 4 スピード 255</p> |
| スピード設定は 10 以上 255 まで STEMDU_robot.tankM1M2(-255,-255) STEMDU_robot.tankM3M4(-255,-255) | スピード設定は 10 以上 255 まで STEMDU_robot.tankM1M2(255,-255) STEMDU_robot.tankM3M4(255,-255) | スピード設定は 10 以上 255 まで STEMDU_robot.tankM1M2(255,-100) STEMDU_robot.tankM3M4(-255,100) | スピード設定は 10 以上 255 まで STEMDU_robot.motor(1,255) STEMDU_robot.motor(4,255) |

組み合わせると、こんな動きも可能



移動距離は「ミリ秒待つ・マイクロ秒待つ」の時間指定で行なう。



15.10.5. 車輪の滑り、空転対策

ロボット動作環境（板床、カーペット、タイル等）での実験です。15-11 章で詳しく解説しますが、ここでは、本機（メカナムホイール使用）が、しっかり接地し、空転しないようにするために、ロボットに重りを載せる実験をします。運動学モデル制御を行なうためにも、ホイール外周に取り付けてある樽（バレル）が接地するようになるための、加重量を実験で求めています。

Experiment with robot working environment (boards, carpets and tiles, etc.). This machine lands tightly here, and I decide not to stall. The experiment which puts a sinker on a robot is done. Kinematics model control has been also installed in the performed purpose in wheel outskirts, a barrel, (barrel) will find the weighting amount to start to land, by an experiment.

加重量実験には、「鉄製貼付けウェイト 5/10g シート」を使います。平らにして置くことができますが、三角なども作れます。切り分けて使う時は、すき間をハサミやカッターナイフ等で切断することが可能です。

まずは、ロボットの上部にそのまま載せます。最初は置くだけで良いでしょう。

下記の Omni-directional Robot

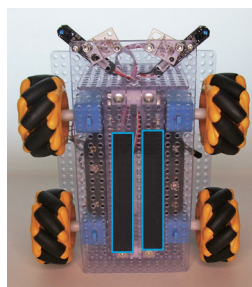
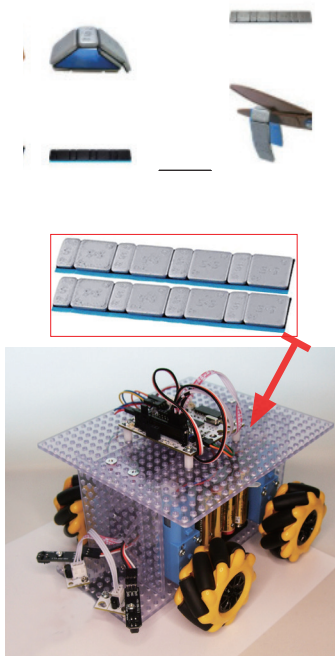
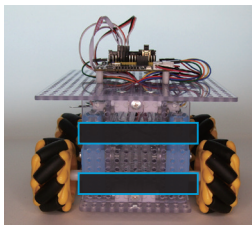
Square parallel movement で実験しましょう。

STEM 技術② 全方向移動プログラム（車体の向きを変えずに、任意の方向に平行移動します。）四方八方移動運動を使って、四方八方に動くことを確認してください。

「バランスウェイト」で調整しながら、プログラムに従い水平に動くかなどを確認し、空転したり、滑ることが無いようになる重さを見つけてみます。

移動距離はプログラムの「ミリ秒待つ」内の時間指定で行なえます。

最適な重さを確認後に、底面や、裏面などに貼り付けます。



"Iron sticking weight 5/10g seat" is used for the weighting amount experiment.

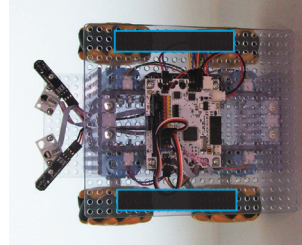
It's possible to make it flat and put it, but a triangle can be made. When cutting and dividing and using it, it's possible to cut a gap by scissors and a retractable knife, etc..

First, it's put on the upper part of a robot just as it is. You'd need only to put the beginning.

You'll experiment in following Omni-directional Robot Square parallel movement. STEM technology(2) The whole way movement programming (Without changing the direction of the body, I move in parallel in the optional direction.) All directions, please confirm that it moves to square every direction using movement movement. As it is programed while adjusting it by "balance weight", the weight by which it starts to confirm whether it moves and be no stalled and slippery things levelly is found.

The displacement can be performed by designation time of the inside where a program "waits for the millisecond".

The most suitable weight is stuck on a base and the back plate after confirmation.



Omni-directional Robot

Square parallel movement

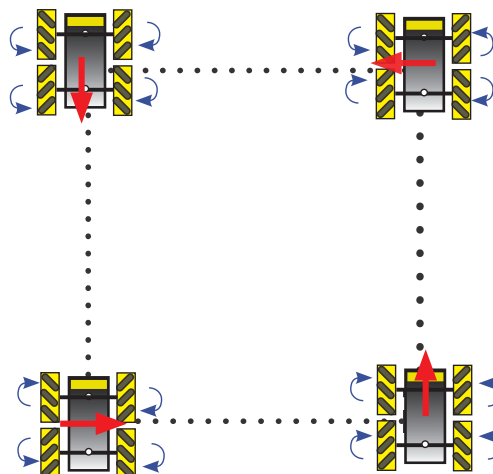
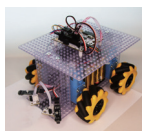
Arduinoで変換したC言語

```
STEMDU _STEMDU_robot = STEMDU();
```

```
void setup()
{
}
```

```
void loop()
{
```

```
  _STEMDU_robot.tankM1M2(200,200);
  _STEMDU_robot.tankM3M4(200,200);
  delay(2000);
  _STEMDU_robot.tankM1M2(-200,200);
  _STEMDU_robot.tankM3M4(200,-200);
  delay(2000);
  _STEMDU_robot.tankM1M2(-200,-200);
  _STEMDU_robot.tankM3M4(-200,-200);
  delay(2000);
  _STEMDU_robot.tankM1M2(200,-200);
  _STEMDU_robot.tankM3M4(-200,200);
  delay(2000);
}
```

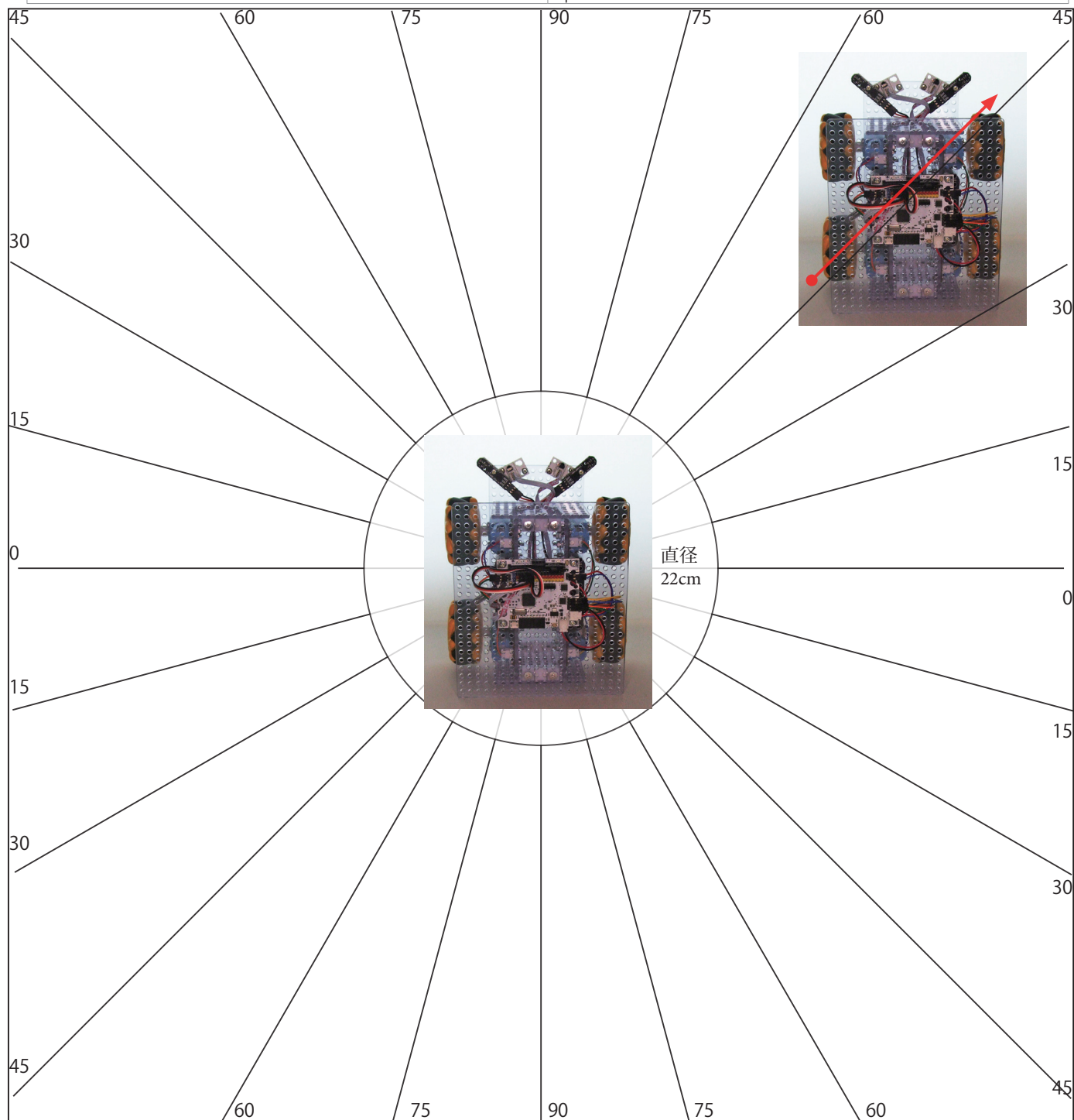


15.10.5. 全方向移動のプログラムに必要なホイール回転方向・速度パラメータを調べる。

Examine the wheel rotation direction and velocity parameters required for omni-directional movement program

4つのモータの回転方向を制御することにより、前後、左右、斜め等あらゆる方向の動きが実現できるメカナムホイールを使用するロボットのプログラミングに必要なパラメータ「ホイールの回転方向と回転速度」を調べます。

By controlling the rotation direction of 4 motors vector composition can be performed, on front-back, left-right, diagonal etc., The parameters required for programming a robot that uses Mecanum wheels that can move in a direction. Check the "wheel rotation direction and rotation speed".



| | 0 | 15 | 30 | 45 | 60 | 75 | 90 | 105 | 120 | 135 | 150 | 165 | 180 |
|----|---|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|
| M1 | | | | | | | | | | | | | |
| M2 | | | | | | | | | | | | | |
| M3 | | | | | | | | | | | | | |
| M4 | | | | | | | | | | | | | |

回転方向・スピード PWM・移動距離に関する「ミリ秒待ち」時間等記録します。

15.10.6 移動機構違いを比較する プラクティス

Omni-directional movement robot practice

下記の 4 角形の辺を動く移動機構違いのロボを作成、演習し、比較します。

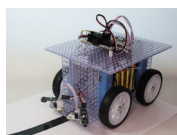
Create and practice an omni-directional moving robot that moves on the sides of the following quadrangle.

| | | |
|--|---|---|
| | <input type="checkbox"/> 100x100 cm 四角形 | 100x100 cm Square |
| | <input type="checkbox"/> 線幅 20 mm 線色：黒色 | Line width 20 mm : Line color : Black, |
| | <input type="checkbox"/> 床色：白色, クリーム色 | Floor : White, Cream |
| | <input type="checkbox"/> 周回 2 周 | Laps 2 |
| | <input type="checkbox"/> 試技回数 3 回 | No. Attempts 3 |
| | <input type="checkbox"/> 3 回の試技中のベストタイムを、演習の結果とします。 | The best time during the 3 trials will be the result of the practice. |
| | <input type="checkbox"/> 演習に利用可能な STEM 技術 | STEM technology available for the practice. |
| | ① 4WD ロボでライトレースプログラム | ① Line trace program by 4WD robo. |
| | ② オムニディレクショナルロボで、四方八方移動プログラム | ② Omni-directional movement program using vector composition. |

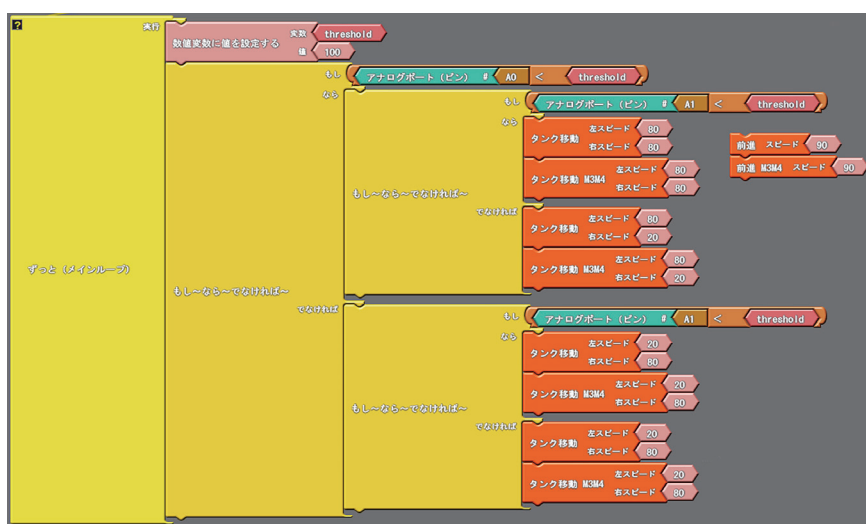
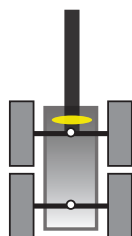
試技集計表 Trial summary table

| | Trial 1 | Trial 2 | Trial 3 |
|--------|---------|---------|---------|
| TEAM A | | | |
| TEAM B | | | |
| TEAM C | | | |

STEM 技術① 4WD ライトレースプログラム (ラインのエッジを辿ります。)



4WD ロボ



STEM 技術② 全方向移動プログラム (車体の向きを変えずに、任意の方向に平行移動します。)

移動距離は「ミリ秒待つ」内の時間指定で行ないます。

Omni-directional Robot

Arduino で変換した C 言語

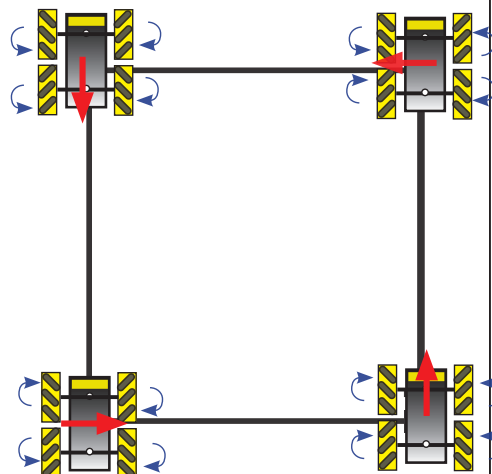
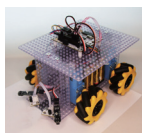
```

STEMDU_STEMDU_robot = STEMdu();

void setup()
{
}

void loop()
{
  _STEMDU_robot.tankM1M2(200,200);
  _STEMDU_robot.tankM3M4(200,200);
  delay( 2000 );
  _STEMDU_robot.tankM1M2(-200,200);
  _STEMDU_robot.tankM3M4(200,-200);
  delay( 2000 );
  _STEMDU_robot.tankM1M2(-200,-200);
  _STEMDU_robot.tankM3M4(-200,-200);
  delay( 2000 );
  _STEMDU_robot.tankM1M2(200,-200);
  _STEMDU_robot.tankM3M4(-200,200);
  delay( 2000 );
}

```



Square parallel movement

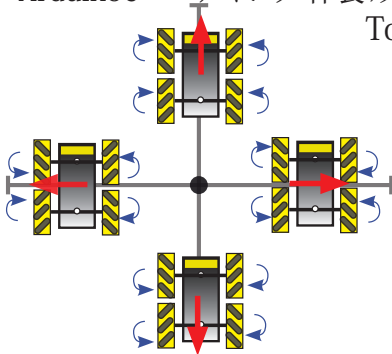
15.10.7. Omnidirectional sample cord 四方八方運動

前後左右への動き(四方八方移動)

The translation movement

ArduinoCコーディング 作製;東京都 鋤先星汰氏

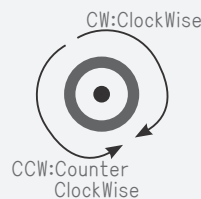
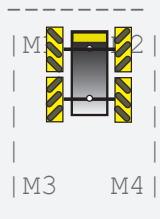
Tokyo Seita Sukisaki



/*

右側面(M2-M4の見える方)から見た時にモーターが全てCW回転することで前進すると仮定する。

/ * Suppose that all the motors rotate CW when viewed from the right side (the side where M2-M4 can be seen) to move forward.



*/

```
void setup() {
    // put your setup code here, to
    run once:
    pinMode(0, OUTPUT);
    pinMode(1, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);
    pinMode(8, OUTPUT);
    pinMode(9, OUTPUT);
    pinMode(10, OUTPUT);
    pinMode(11, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(13, OUTPUT);
    Serial.begin(115200);
}

void loop() {
    // put your main code here, to
    run repeatedly:
    Stop();
    delay(1000);
}
```

```
Fwd(255);
delay(1000);
Back(255);
delay(1000);
Right(255);
delay(1000);
Left(255);
delay(1000);
Fwd_Right(255);
delay(1000);
Back_Right(255);
delay(1000);
Fwd_Left(255);
delay(1000);
Back_Left(255);
delay(1000);
Brk();
delay(1000);
}

int M1_CW(int pwm) {
    digitalWrite(9, LOW);
    digitalWrite(4, HIGH);
    analogWrite(6, pwm);
}

int M1_CCW(int pwm) {
    digitalWrite(9, HIGH);
    digitalWrite(4, LOW);
    analogWrite(6, pwm);
}

int M1_stop() {
    digitalWrite(9, LOW);
    digitalWrite(4, LOW);
}

int M2_CW(int pwm) {
    digitalWrite(7, LOW);
    digitalWrite(8, HIGH);
    analogWrite(5, pwm);
}

int M2_CCW(int pwm) {
    digitalWrite(7, HIGH);
    digitalWrite(8, LOW);
    analogWrite(5, pwm);
}

int M2_stop() {
    digitalWrite(7, LOW);
    digitalWrite(8, LOW);
}

int M3_CW(int pwm) {
    digitalWrite(0, LOW);
    digitalWrite(1, HIGH);
    analogWrite(11, pwm);
}
```



```

}
int M3_CCW(int pwm) {
    digitalWrite(0, HIGH);
    digitalWrite(1, LOW);
    analogWrite(11, pwm);
}
int M3_stop() {
    digitalWrite(0, LOW);
    digitalWrite(1, LOW);
}
int M4_CW(int pwm) {
    digitalWrite(12, LOW);
    digitalWrite(10, HIGH);
    analogWrite(13, pwm);
}
int M4_CCW(int pwm) {
    digitalWrite(12, HIGH);
    digitalWrite(10, LOW);
    analogWrite(13, pwm);
}
int M4_stop() {
    digitalWrite(12, LOW);
    digitalWrite(10, LOW);
}
int Fwd(int pwm) {
    M1_CW(pwm);
    M2_CW(pwm);
    M3_CW(pwm);
    M4_CW(pwm);
}
int Back(int pwm) {
    M1_CCW(pwm);
    M2_CCW(pwm);
    M3_CCW(pwm);
    M4_CCW(pwm);
}
int Left(int pwm) {
    M1_CCW(pwm);
    M2_CW(pwm);
    M3_CW(pwm);
    M4_CCW(pwm);
}
int Right(int pwm) {
    M1_CW(pwm);
    M2_CCW(pwm);
    M3_CCW(pwm);
    M4_CW(pwm);
}
int Fwd_Left(int pwm) {
    M1_stop();
    M2_CW(pwm);
    M3_CW(pwm);

```

```

    M4_stop();
}
int Back_Left(int pwm) {
    M1_stop();
    M2_CCW(pwm);
    M3_CCW(pwm);
    M4_stop();
}
int Fwd_Right(int pwm) {
    M1_CW(pwm);
    M2_stop();
    M3_stop();
    M4_CW(pwm);
}
int Back_Right(int pwm) {
    M1_CCW(pwm);
    M2_stop();
    M3_stop();
    M4_CCW(pwm);
}
int Turn_Left(int pwm) {
    M1_CCW(pwm);
    M2_CW(pwm);
    M3_CCW(pwm);
    M4_CW(pwm);
}
int Turn_Right(int pwm) {
    M1_CW(pwm);
    M2_CCW(pwm);
    M3_CW(pwm);
    M4_CCW(pwm);
}
int Stop() {
    M1_stop();
    M2_stop();
    M3_stop();
    M4_stop();
}
int Brk() {
    digitalWrite(9, HIGH);
    digitalWrite(4, HIGH);
    digitalWrite(7, HIGH);
    digitalWrite(8, HIGH);
    digitalWrite(0, HIGH);
    digitalWrite(1, HIGH);
    digitalWrite(12, HIGH);
    digitalWrite(10, HIGH);
}

```

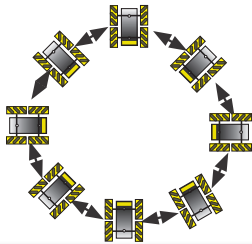
15.10.8. Omnidirectional sample cord 円弧運動

大きな円での円弧運動Cプログラム

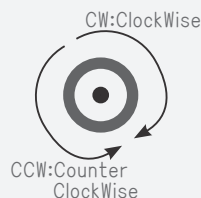
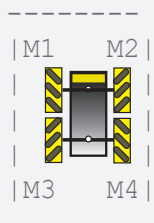
Arc movement

ArduinoCコーディング 作製;東京都 鋤先星汰氏

Tokyo Seita sukisaki



```
/*
  右側面 (M2-M4の見える方) から見た時にモーター
  が全てCW回転することで前進すると仮定する
  /* Suppose that all the motors rotate CW when
  viewed from the right side (the side where M2-M4
  can be seen) to move forward.
```



```
*/

float deg = 0;
float rad = 0;

void setup() {
  // put your setup code here, to
  run once:
  pinMode(0, OUTPUT);
  pinMode(1, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
  Serial.begin(115200);
}

void loop() {
  for (int i = 0; i < 360; i++) {
    Move(100, i, 0); // (スピー
    ド、角度(0~359度)、0)で入力
    delay(10);
```

```

  }
}
/*
  Stop();
  delay(1000);
  Fwd(255,255,255,255);
  delay(1000);
  Brk();
  delay(1000);
  Back(255,255,255,255);
  delay(1000);
  Brk();
  delay(1000);
  Right(255,255,255,255);
  delay(1000);
  Brk();
  delay(1000);
  Left(255,255,255,255);
  delay(1000);
  Brk();
  delay(1000);
  Fwd_Right(255,255);
  delay(1000);
  Brk();
  delay(1000);
  Back_Left(255,255);
  delay(1000);
  Brk();
  delay(1000);
  Back_Right(255,255);
  delay(1000);
  Brk();
  delay(1000);
  Turn_Right(255,255,255,255);
  delay(1000);
  Brk();
  delay(1000);
  Turn_Left(255,255,255,255);
  delay(1000);
  Brk();
  delay(1000);
  */

int Move(int spd, int deg, int rot)
{
  Serial.print(deg);
  Serial.print("\t");
  int deg2 = 90 - deg;
```



```

rad = deg2 * 3.1415926 / 180;

int pwm_1 = 100;
int pwm_2 = pwm_1 * (tan(rad) +
1) / (tan(rad) - 1);

if(deg == 225){
    pwm_1 = 100;
    pwm_2 = 10000;
}
float rt = (float)abs(pwm_2) /
(float)abs(pwm_1);

Serial.print(pwm_1);
Serial.print("\t");
Serial.print(pwm_2);
Serial.print("\t");

if (rt < 1) {
    pwm_1 = 255;
    pwm_2 = 255 * rt;
} else {
    pwm_1 = 255 / rt;
    pwm_2 = 255;
}

if (45 < deg && deg < 225 ) {
    pwm_1 = pwm_1 * (-1);
}
if (135 < deg && deg < 315 ) {
    pwm_2 = pwm_2 * (-1);
}
int pwm_3 = pwm_1;
int pwm_4 = pwm_2;
pwm_1 = pwm_1 * spd / 100;
pwm_2 = pwm_2 * spd / 100;
pwm_3 = pwm_3 * spd / 100;
pwm_4 = pwm_4 * spd / 100;

Serial.print(rad);
Serial.print("\t");
    Serial.print(rt);
Serial.print("\t");
Serial.print(pwm_1);
Serial.print("\t");
Serial.print(pwm_2);
Serial.print("\t");
Serial.print(pwm_3);
Serial.print("\t");
Serial.println(pwm_4);

if (pwm_1 > 0) {

```

```

    M1_CW(pwm_1);
} else {
    pwm_1 = pwm_1 * (-1);
    M1_CCW(pwm_1);
}
if (pwm_2 > 0) {
    M2_CW(pwm_2);
} else {
    pwm_2 = pwm_2 * (-1);
    M2_CCW(pwm_2);
}
if (pwm_3 > 0) {
    M3_CW(pwm_3);
} else {
    pwm_3 = pwm_3 * (-1);
    M3_CCW(pwm_3);
}
if (pwm_4 > 0) {
    M4_CW(pwm_4);
} else {
    pwm_4 = pwm_4 * (-1);
    M4_CCW(pwm_4);
}

}

int M1_CW(int pwm_1) {
    digitalWrite(9, LOW);
    digitalWrite(4, HIGH);
    analogWrite(6, pwm_1);
}

int M1_CCW(int pwm_1) {
    digitalWrite(9, HIGH);
    digitalWrite(4, LOW);
    analogWrite(6, pwm_1);
}

int M1_stop() {
    digitalWrite(9, LOW);
    digitalWrite(4, LOW);
}

int M2_CW(int pwm_2) {
    digitalWrite(7, HIGH);
    digitalWrite(8, LOW);
    analogWrite(5, pwm_2);
}

int M2_CCW(int pwm_2) {
    digitalWrite(7, LOW);
    digitalWrite(8, HIGH);
    analogWrite(5, pwm_2);
}

}

```

```

int M2_stop() {
    digitalWrite(7, LOW);
    digitalWrite(8, LOW);
}
int M3_CW(int pwm_3) {
    digitalWrite(0, HIGH);
    digitalWrite(1, LOW);
    analogWrite(11, pwm_3);
}
int M3_CCW(int pwm_3) {
    digitalWrite(0, LOW);
    digitalWrite(1, HIGH);
    analogWrite(11, pwm_3);
}
int M3_stop() {
    digitalWrite(0, LOW);
    digitalWrite(1, LOW);
}
int M4_CW(int pwm_4) {
    digitalWrite(12, LOW);
    digitalWrite(10, HIGH);
    analogWrite(13, pwm_4);
}
int M4_CCW(int pwm_4) {
    digitalWrite(12, HIGH);
    digitalWrite(10, LOW);
    analogWrite(13, pwm_4);
}
int M4_stop() {
    digitalWrite(12, LOW);
    digitalWrite(10, LOW);
}
int Fwd(int pwm_1, int pwm_2, int
pwm_3, int pwm_4) {
    M1_CW(pwm_1);
    M2_CW(pwm_2);
    M3_CW(pwm_3);
    M4_CW(pwm_4);
}
int Back(int pwm_1, int pwm_2, int
pwm_3, int pwm_4) {
    M1_CCW(pwm_1);
    M2_CCW(pwm_2);
    M3_CCW(pwm_3);
    M4_CCW(pwm_4);
}
int Left(int pwm_1, int pwm_2, int
pwm_3, int pwm_4) {
    M1_CW(pwm_1);
    M2_CCW(pwm_2);
    M3_CW(pwm_3);

```

```

    M4_CCW(pwm_4);
}
int Right(int pwm_1, int pwm_2, int
pwm_3, int pwm_4) {
    M1_CCW(pwm_1);
    M2_CW(pwm_2);
    M3_CCW(pwm_3);
    M4_CW(pwm_4);
}
int Fwd_Left(int pwm_1, int pwm_3)
{
    M1_CW(pwm_1);
    M2_stop();
    M3_CW(pwm_3);
    M4_stop();
}
int Back_Right(int pwm_1, int
pwm_3) {
    M1_CCW(pwm_1);
    M2_stop();
    M3_CCW(pwm_3);
    M4_stop();
}
int Fwd_Right(int pwm_2, int pwm_4)
{
    M1_stop();
    M2_CW(pwm_2);
    M3_stop();
    M4_CW(pwm_4);
}
int Back_Left(int pwm_2, int pwm_4)
{
    M1_stop();
    M2_CCW(pwm_2);
    M3_stop();
    M4_CCW(pwm_4);
}
int Turn_Left(int pwm_1, int pwm_2,
int pwm_3, int pwm_4) {
    M1_CW(pwm_1);
    M2_CCW(pwm_2);
    M3_CCW(pwm_3);
    M4_CW(pwm_4);
}
int Turn_Right(int pwm_1, int
pwm_2, int pwm_3, int pwm_4) {
    M1_CCW(pwm_1);
    M2_CW(pwm_2);
    M3_CW(pwm_3);
    M4_CCW(pwm_4);
}

```



```

}
int Stop() {
    M1_stop();
    M2_stop();
    M3_stop();
    M4_stop();
}
int Brk() {
    digitalWrite(9, HIGH);
    digitalWrite(4, HIGH);
    digitalWrite(7, HIGH);
    digitalWrite(8, HIGH);
    digitalWrite(0, HIGH);
    digitalWrite(1, HIGH);
    digitalWrite(12, HIGH);
    digitalWrite(10, HIGH);
}

```

15.11. 運動学

メカナムホイールロボットの速度制御 ～運動学モデルにもとづいた制御～

東洋大学 理工学部 機械工学科
山川聡子

自動車や自転車の車輪は、中心の軸（＝車軸）に対して回転します。回転する車輪を地面に接地させると車輪は一方向に進みます。このとき、車輪が移動していく方向は、車軸に対して垂直な方向です（図1）。ですから、自転車でハンドルを真ん中に固定して前輪と後輪の車軸が平行にすると、自転車は直進します。自転車の進行方向を変えたいときは、ハンドルを切って前輪の車軸の向きを変える必要があります。このように通常車輪は進む方向が決まっていますから、自転車や自動車は真横に動くことができません。真横に移動するためには、前後に移動しながら切り返し動作などを行う必要があります。

これに対して、メカナムホイールでは車輪の外周に複数の「ころ」がついています。ころの回転軸は車輪の車軸に対して傾いており、ころはこの回転軸まわりに自由に回転します。そのため、メカナムホイールの接地点は、車輪の回転による速度（図2の青線）にころの回転によって生じる速度（図2の赤線）を合わせた図2の紫の矢印の方向へ移動します。しかし、このメカナムホイールを同じ向きでロボットに取り付けてしまうと、モータを止めてもころが勝手に転がってロボットの動きを止めることができません。そこで、ロボットに取り付けるときは勝手に転がらないように複数のメカナムホイールの向きを変えて取り付けます。

では、自由に回転するころがついたメカナムホイールを用いて、ロボットを目標の方向へ動かすためには、それぞれのモータをどう回転させたら良いのでしょうか？運動学や数学の知識（ベクトル）を使って考えていきましょう。

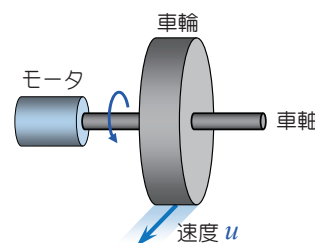


図1 車輪の速度

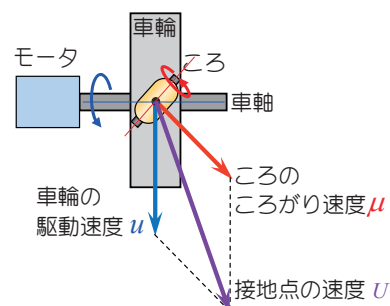


図2 メカナムホイールの速度

15.11.1. メカナムホイールの速度

まず、一つの車輪について見てみましょう。車輪の車軸にはモータが接続されていて、モータを回すと車輪も回転します。このとき、車輪は地面に対してすべらないものとします。モータを回転させたときに車輪の接地点が進む方向を x 軸、車軸の方向を y 軸とした座標系を考えます。車輪の接地点が地面に対して u という速さで前進するとき、この速度をベクトルを使って、

$$\mathbf{u} = \begin{pmatrix} u \\ 0 \end{pmatrix} \quad (1)$$

と表すことにします。ベクトルの1行目が速度の x 軸方向成分、2行目が y 軸方向の成分です。なお、通常、ベクトルを記号で表すときは(1)の左辺のように太字を使います。 u はモータを回すことによって発生する速度なので、以下では駆動速度と呼ぶことにします。

一方、ころの回転によって生じる地面に対する速さを μ とし、以下ではころがり速度とよぶことにします。ころの軸にはモータは付いていませんから、このころがり速度 μ は能動的に設定することはできません。ころの接地点が地面に対して進む方向は、ころの軸に垂直な方向です。つまり、 x 軸に対して 45° 傾いた方向です。この速度をベクトルで表すと、

$$\boldsymbol{\mu} = \begin{pmatrix} \mu \sin 45^\circ \\ \mu \cos 45^\circ \end{pmatrix} \quad (2)$$

となります（図3参照）。

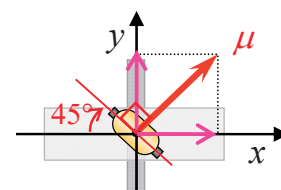


図3 転がり速度

車輪を回転させているときにもころも転がると、車輪の接地点の速度 U はこれらのベクトルの和となります。

$$\begin{aligned} U &= \begin{pmatrix} u \\ 0 \end{pmatrix} + \begin{pmatrix} \mu \sin 45^\circ \\ \mu \cos 45^\circ \end{pmatrix} \\ &= \begin{pmatrix} u + \mu \sin 45^\circ \\ \mu \cos 45^\circ \end{pmatrix} \end{aligned} \quad (3)$$

15.11.2. ロボットの速度

あらためて、ロボットの中心を原点、前方を x 軸、横方向を y 軸としてロボットに固定した座標系を考えましょう。ロボットに対して、4つのメカナムホイールを図4のように取り付けます。図中の[1]は車輪の番号を表します。前後の車軸間距離は $2l_x$ [m]、左右車輪の中心間距離は $2l_y$ [m]です。図中のころの向きは車輪の接地点側（つまり床側）から見た方向で描いています。

この座標系において各車輪の駆動速度をそれぞれ u_1, u_2, u_3, u_4 、ころのころがり速度を $\mu_1, \mu_2, \mu_3, \mu_4$ とします。このとき、車輪1の接地点の速度 U_1 は前述の通り、

$$U_1 = \begin{pmatrix} u_1 + \mu_1 \sin 45^\circ \\ \mu_1 \cos 45^\circ \end{pmatrix} \quad (4)$$

です。車輪2～3についてもころの向きに注意すると、

$$U_2 = \begin{pmatrix} u_2 + \mu_2 \sin 45^\circ \\ -\mu_2 \cos 45^\circ \end{pmatrix} \quad (5)$$

$$U_3 = \begin{pmatrix} u_3 + \mu_3 \sin 45^\circ \\ -\mu_3 \cos 45^\circ \end{pmatrix} \quad (6)$$

$$U_4 = \begin{pmatrix} u_4 + \mu_4 \sin 45^\circ \\ \mu_4 \cos 45^\circ \end{pmatrix} \quad (7)$$

となります。

つぎに、ロボットの運動について考えてみましょう。ロボットが回転せずに x 軸方向に速度 v_x で移動している場合、各車輪の接地点の速度も同じ v_x です。もし一つでも車輪の速度が異なっていれば、ロボットは回転してしまうか、動かないかどちらかです。これは x 軸方向だけでなく、あらゆる方向について同じことが言えます（図5）。ロボットの中心が移動する速度を並進速度と呼ぶこととし、 x 軸、 y 軸方向の成分をそれぞれ v_x [m/s]、 v_y [m/s] とおくことにします。このとき、各車輪の速度は、

$$U_1 = U_2 = U_3 = U_4 = \begin{pmatrix} v_x \\ v_y \end{pmatrix}$$

を満たす必要があります。

つぎに、ロボットの中心位置を固定したまま、その中心まわりに回転している場合を考えてみましょう。反時計回りを正として回転の角速度が ω [rad/s] であるとし、このとき、各車輪は中心とその接地点を結んだ直線に対して垂直な方向、つまり円周方向の速度を持っています（図6）。この速度は中心からの距離 $R = \sqrt{l_x^2 + l_y^2}$ に比例していて $R\omega$ です。車輪1の速度について図7のような幾何学的な関係に注意して x 軸、 y 軸成分の速度成分を求めると、

$$U_1 = R\omega \begin{pmatrix} -\frac{l_y}{R} \\ \frac{l_x}{R} \end{pmatrix}$$

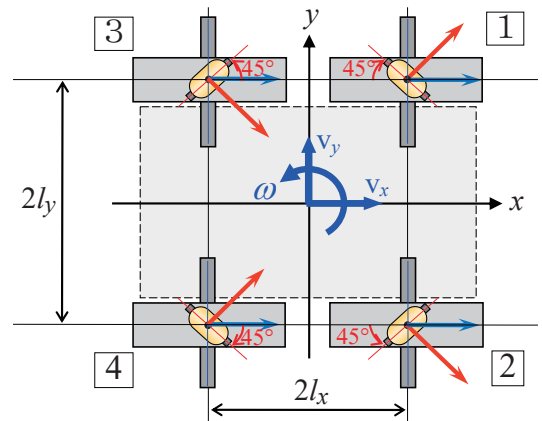


図4 メカナムホイールロボットの速度

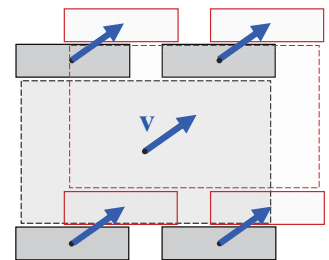


図5 並進運動

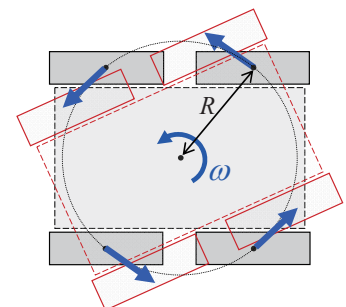


図6 回転運動

となります。同様に、他の車輪についても向きに注意すると、各車輪の接地点での速度は、

$$U_1 = \begin{pmatrix} -l_y \omega \\ l_x \omega \end{pmatrix}, U_2 = \begin{pmatrix} l_y \omega \\ l_x \omega \end{pmatrix}, U_3 = \begin{pmatrix} -l_y \omega \\ -l_x \omega \end{pmatrix}, U_4 = \begin{pmatrix} l_y \omega \\ -l_x \omega \end{pmatrix}$$

を満たす必要があります。

以上のことから、ロボットを並進速度 v_x, v_y , かつ回転角速度 ω で移動させるためには、各車輪の接地点の速度が

$$U_1 = \begin{pmatrix} v_x - l_y \omega \\ v_y + l_x \omega \end{pmatrix} \quad (8)$$

$$U_2 = \begin{pmatrix} v_x + l_y \omega \\ v_y + l_x \omega \end{pmatrix} \quad (9)$$

$$U_3 = \begin{pmatrix} v_x - l_y \omega \\ v_y - l_x \omega \end{pmatrix} \quad (10)$$

$$U_4 = \begin{pmatrix} v_x + l_y \omega \\ v_y - l_x \omega \end{pmatrix} \quad (11)$$

を満たす必要があることが分かります。

式(4)-(7)が式(8)-(11)を満たすために必要な駆動速度 $u_1 \sim u_4$ を求めましょう。例として、 U_1 についてみると、式(4)と式(8)のそれぞれ、1行目、2行目どうしが等しくなればよいので、

$$u_1 + \mu_1 \sin 45^\circ = v_x - l_y \omega$$

$$\mu_1 \cos 45^\circ = v_y + l_x \omega$$

という2つの連立方程式になります。2つ目の式から $\mu_1 = (v_y + l_x \omega) / \cos 45^\circ$ なので、これを1つ目の式に代入すると、

$$u_1 + (v_y + l_x \omega) \frac{\sin 45^\circ}{\cos 45^\circ} = v_x - l_y \omega$$

$$u_1 = v_x - v_y - (l_x + l_y) \omega$$

と得られます。同様に他の車輪についての式も解くと、ロボットを (v_x, v_y) , ω で動かすために必要な各車輪の駆動速度は、

$$\begin{aligned} u_1 &= v_x - v_y - (l_x + l_y) \omega \\ u_2 &= v_x + v_y + (l_x + l_y) \omega \\ u_3 &= v_x + v_y - (l_x + l_y) \omega \\ u_4 &= v_x - v_y + (l_x + l_y) \omega \end{aligned} \quad (12)$$

と得られます。

なお、各車輪をこの駆動速度で回転させているとき、ころは接地点で

$$\begin{aligned} \mu_1 &= \sqrt{2} (v_y + l_x \omega) \\ \mu_2 &= \sqrt{2} (-v_y - l_x \omega) \\ \mu_3 &= \sqrt{2} (-v_y + l_x \omega) \\ \mu_4 &= \sqrt{2} (v_y - l_x \omega) \end{aligned} \quad (13)$$

という速度で転がっています。

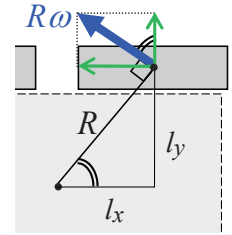


図7 回転速度

15.11.3. サンプルプログラム

ロボット固定座標に対して、前方向の速度 v_x 、横方向の速度 v_y 、中心まわりの旋回速度 w を与えて走行するためのサンプルプログラムを示します。前述の説明は車輪が地面に対してすべらない場合に成り立ちますので、床面に対してメカナムホイール4つがしっかり接地していることが必要です。もし、浮いている（空転している）ようであれば、ロボットに重りを載せると接地します。

また、ロボットへの入力値はモータへの PWM 入力値であり、直接モータの駆動速度を与えることはできません。そのため、あらかじめ、与えた PWM 入力値に対してモータの駆動速度がどうなるか、その関係を調べておく必要があります。以下のサンプルプログラムでは、(PWM 値)=(モータの速度)×係数という比例の関係であるとしています。モータの特性がおおよそ同じであれば、同じ係数を掛ければおおよそ理論通り動くはずですが、しかし、モータ特性がばらついている場合は、それぞれの係数を調整する必要があります。

サンプルプログラムの中で、**ロボットの車軸間距離**や**各モータの係数**など（サンプルプログラムの中の赤字）は使用しているロボットに合わせて調整してください。また、各モータへの最終的な PWM 入力値が限界値（255）を超えてしまうような目標速度を与えると、4つの車輪の速度比が崩れて目標の動きをしなくなってしまうます。一方で、入力値が小さすぎると摩擦のせいでモータが回りません。ハードウェアに合わせて、適切な目標速度（サンプルプログラムの青字）を設定してください。

サンプルプログラムでは、前進、横移動、斜め移動、その場回転、円軌道走行、斜め移動を繰り返します。目標速度 v_x 、 v_y 、 w を変えて、ロボットの走行軌道が変わる様子を確認してみてください。

```

/*****
Speed control of mecanum wheel robot    by S.Yamakawa 20220113
メカナムホイールロボット(robo designer)用、開発環境はArduino 1.6.10
並進速度(vx, vy)と回転速度wを指定してロボットを動かすプログラム

----座標系とモータ番号-----
      x
      ↑
    |M1  M2|
y ← |      |
    |M3  M4|
*****/

//=====ロボットのパラメータ（※ロボットに合わせて変更）=====
float lx=0.095; //前後タイヤの車軸間距離
float ly=0.13;  //左右タイヤの車軸間距離
/*****
  初期化处理
*****/

void setup() {
  pinMode(0, OUTPUT);
  pinMode(1, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
  Serial.begin(115200);
}

```

```

/*****
繰り返しメインループ
*****/
void loop() {
    motorctrl(0.16,0.0,0.0); //前に 2 秒進んで
    delay(2000);
    motorctrl(0.0,0.0,0.0); //1 秒止まって
    delay(1000);
    motorctrl(0.0,0.16,0.0); //左に 2 秒進んで
    delay(2000);
    motorctrl(0.0,0.0,0.0); //1 秒止まって
    delay(1000);
    motorctrl(-0.08,-0.08,0.0); //右斜め後ろへ 2 秒進んで
    delay(2000);
    motorctrl(0.0,0.0,0.0); //1 秒止まって
    delay(1000);
    motorctrl(0.0,0.0,1.5); //その場で 2 秒反時計回りに回転して
    delay(2000);
    motorctrl(0.0,0.0,0.0); //1 秒止まって
    delay(1000);
    motorctrl(0.16,0.0,0.5); //2 秒反時計回りに円軌道で走行して
    delay(2000);
    motorctrl(0.0,0.0,0.0); //1 秒止まって
    delay(1000);
    motorctrl(-0.08,0.08,0.0); //左斜め後ろへ 2 秒進んで
    delay(2000);
    motorctrl(0.0,0.0,0.0); //1 秒止まる
    delay(1000);
}

/*****
各車輪の速度計算 (与えられた vx (m/s), vy (m/s), w (rad/s) を達成するためのモータ指令値)
*****/
int motorctrl(float vx, float vy, float w){
    float u1,u2,u3,u4;
    float temp;
    int umax=255;
    u1=(vx-vy-(lx+ly)/2*w)*800;
    u2=(vx+vy+(lx+ly)/2*w)*800;
    u3=(vx+vy-(lx+ly)/2*w)*790;
    u4=(vx-vy+(lx+ly)/2*w)*850;
    //指令値がモータの限界 umax を超えるときは、減速して移動方向は保持する
    temp=min(u1,u2);
    temp=min(temp,u3);
    temp=min(temp,u4);
    temp=max(-temp,u1);
    temp=max(temp,u2);
    temp=max(temp,u3);
    temp=max(temp,u4);
    if (temp>umax){
        u1=u1/temp*umax;
        u2=u2/temp*umax;
        u3=u3/temp*umax;
        u4=u4/temp*umax;
    }
    motorM1(u1);
    motorM2(u2);
    motorM3(u3);
    motorM4(u4);
}

```

```

/*****
モータ関数
*****/
int motorM1(int pwm) {
    if(pwm > 0){
        digitalWrite(9, LOW);
        digitalWrite(4, HIGH);
    }else{
        pwm=pwm*-1;
        digitalWrite(9, HIGH);
        digitalWrite(4, LOW);
    }
    analogWrite(6, pwm);
}
int motorM2(int pwm) {
    if(pwm > 0){
        digitalWrite(7, HIGH);
        digitalWrite(8, LOW);
    }else{
        pwm=pwm*-1;
        digitalWrite(7, LOW);
        digitalWrite(8, HIGH);
    }
    analogWrite(5, pwm);
}
int motorM3(int pwm) {
    if(pwm > 0){
        digitalWrite(0, HIGH);
        digitalWrite(1, LOW);
    }else{
        pwm=pwm*-1;
        digitalWrite(0, LOW);
        digitalWrite(1, HIGH);
    }
    analogWrite(11, pwm);
}
int motorM4(int pwm) {
    if(pwm > 0){
        digitalWrite(12, LOW);
        digitalWrite(10, HIGH);
    }else{
        pwm=pwm*-1;
        digitalWrite(12, HIGH);
        digitalWrite(10, LOW);
    }
    analogWrite(13, pwm);
}

```


15. 12. 拡張機能使い方 How to use the extension facility.

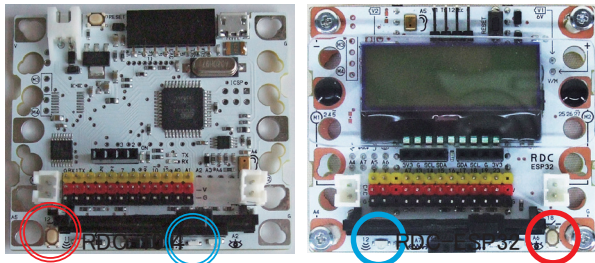
15.12.1. マイコンボード搭載センサー How to use the sensor with a microcomputer board.

①ボタン Button・・・ボタンを押すと LED が点灯する。 A button is pressed and an LED is set.

●ArduBlockのサンプルプログラム Sample program of ArduBlock: Button_LED

・ArduBlockソースコードは、PC/MyDocuments/Arduino/へ配置したサンプルフォルダ[RDC_ArduBlockSamples]に、ファイル名[b01_RDC_Button_LED.abp]で格納されています。

* ArduBlock source cord is stocked in the sample folder arranged to PC/MyDocuments/Arduino/ [RDC_ArduBlockSamples] by the file name [b01_RDC_Button_LED.abp].



| | | |
|-----------|---|--|
| RDC-104 | ボタン(12番ピン \odot)を押すとLED(13番ピン \odot)が点灯します。 | If a button (12th pin \odot) is pressed, an LED (13th pin \odot) is lit. |
| RDC-ESP32 | ボタン(18番ピン \odot)を押すとLED(19番ピン \odot)が点灯します。 | If a button (18th pin \odot) is pressed, an LED (19th pin \odot) is lit. |
| | ボタンを放したら消灯します。 | If a button is released, Turn off the LED lights. |
| | シリアルモニタにボタンの状態を出力します。(押すと0、放すと1) ボタンはプルアップするので、放した状態が1 (HIGH)です。 | The state of the button is output for a serial monitor. (Pushed =0 and released =1) A button is pull up --and the released position is 1 (HIGH). |

■Arduino Cのサンプルプログラム Sample program of Arduino C: DigitalInputPullup

・Arduinoスケッチ例> 02.Digital > DigitalInputPullup(ボタンを押すとLEDが点灯する)を読み込み

RDC-104: pinMode(2, INPUT_PULLUP) と digitalWrite(2)のピン番号を2から12に変更する。

RDC-ESP32:ピン番号を2から18に変更する。

```
void setup() {
  //start serial connection
  Serial.begin(9600);
  //configure pin2 as an input and enable the internal pull-up resistor
  pinMode(2, INPUT_PULLUP); RDC-104: ピン番号を2から12に変更 The pin number is changed in 12 from 2.
                             RDC-ESP32: ピン番号を2から18に変更する The pin number is changed in 18 from 2.

  pinMode(13, OUTPUT); RDC-ESP32: ピン番号を13から19に変更する The pin number is changed in 19 from 13.
}

void loop() {
  //read the pushbutton value into a variable
  int sensorVal = digitalRead(2);
  RDC-104: ピン番号を2から12に変更 The pin number is changed in 12 from 2.
  RDC-ESP32: ピン番号を2から18に変更する The pin number is changed in 18 from 2.

  //print out the value of the pushbutton
  Serial.println(sensorVal);

  // Keep in mind the pullup means the pushbutton's
  // logic is inverted. It goes HIGH when it's open,
  // and LOW when it's pressed. Turn on pin 13 when the
  // button's pressed, and off when it's not:
  if (sensorVal == HIGH) {
    digitalWrite(13, LOW); RDC-ESP32: ピン番号を13から19に変更する The pin number is changed in 19 from 31.
  } else {
    digitalWrite(13, HIGH); RDC-ESP32: ピン番号を13から19に変更する The pin number is changed in 19 from 13.
  }
}
```

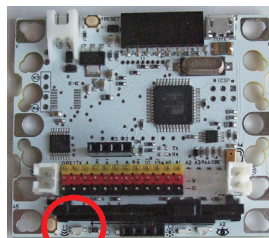
②内臓赤外線距離センサ …赤外線 LED と明るさセンサで測距

Infrared range sensor of board internal organs---Distance surveying by an infrared LED and a brightness sensor,

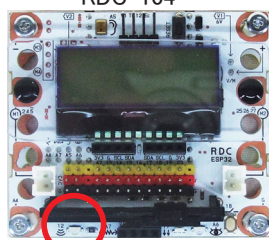
● ArduBlock サンプルプログラム Sample program of ArduBlock: IR distance

・ArduBlock ソースコードは、PC/MyDocuments/Arduino/へ配置したサンプルフォルダ [RDC_ArduBlockSamples] に、ファイル名【b04_RDC_IRdistance.abp】で格納されています。

* ArduBlock source cord is stocked in the sample folder arranged to PC/MyDocuments/Arduino/ [RDC_ArduBlockSamples] by the file name [b04_RDC_IRdistance.abp].



RDC-104



RDC-ESP32

内臓赤外線距離センサ



| | |
|---------------------------------------|--|
| ボードの赤外線 LED と明るさセンサを使って対象物までの距離を測ります。 | The distance to the target thing is measured using an infrared LED of a board and a brightness sensor. |
| 10cm 程度までの距離を計測できます。 | The distance up to about 10 cm can be measured. |
| 数値 (単位はありません) をシリアルモニタに出力します。 | The numerical value (There are no units.) is output for a serial monitor. |
| 対象物が黒いなど、赤外線を反射しないと計測できません。 | Such as a target thing is black, when infrared rays aren't reflected, you can't measure. |

■ Arduino C のサンプルプログラム Sample program of Arduino C: IR distance

スケッチブック >RDC_samples>b04_RDC_IR_distance (赤外線 LED と明るさセンサで測距する)
(Which do a ranging are an infrared LED and a brightness sensor.)

```
// Definitions.
const int BUFFER_SIZE = 45;          // use an odd number

// Global variables.
double buf[BUFFER_SIZE];             // Analog readings at 100khz & stored here
double out[BUFFER_SIZE];             // output of filter stored here.
int buffer_index;                    // Interrupt increments buffer
boolean buffer_full;                 // Flag for when complete.

double a0,a1,a2,b1,b2;               // filter kernel poles
double f,bw;                         // frequency cutoff and bandwidth
double r,k;                          // filter coefficients

int LEDonoff;
// the setup routine runs once when you press reset:
void setup() {
  for( int i = 0; i < BUFFER_SIZE; i++ ) {
    buf[i] = 0;
    out[i] = 0;
  }
  buffer_index = 0;
  buffer_full = false;

  // initialize serial communication at 9600 bits per second:
  Serial.begin(57600);
  while(!Serial) {                    // For STEM Du RDC
  }
  pinMode(11,OUTPUT);
  pinMode(12,INPUT_PULLUP);
```

```
// Lets sort out the filter variables before we end setup.

// Cut-off frequency.
// We are looking for a 38khz IR TV remote.
// F is fraction of the sample frequency
// It has to be between 0 and 0.5. Therefore, the interrupt
// needs to be at least *double* the bandpass frequency.
// I picked 100khz as a nice number to scale from.
// So, f = (100khz * 0.38) = 38Khz
//f = 0.38;
f = 0.4;
// Bandwidth (allowance) of bandpass filter.
// Same principle as above (fraction of 100khz).
// We are using this filter to get rid of ambient environment
// noise. 20khz seems like a big band, but I wouldn't expect
// there to be much in the khz. You can fine tune downwards.
//bw = 0.2;
bw = 0.2;

// Maths. Read the book. Does the trick.
r = 1 - ( 3 * bw );
k = 1 - ( 2 * r * cos( 2 * PI * f ) ) + ( r * r );
k = k / ( 2 - ( 2 * cos( 2 * PI * f ) ) );

a0 = 1 - k;
a1 = ( 2 * ( k - r ) ) * ( cos( 2 * PI * f ) );
a2 = ( r * r ) - k;
b1 = 2 * r * cos( 2 * PI * f );
b2 = 0 - ( r * r );

LEDonoff = 0;
}

// the loop routine runs over and over again forever:
void loop() {
    float output;

    // read the input on analog pin A2 of Ph.T:
    //int sensorValue = analogRead(A2);

    if( buffer_index >= BUFFER_SIZE ) {
        buffer_index = 0;
        buffer_full = true;
    }
    else if ( buffer_full == false ){
        buf[ buffer_index ] = (double)analogRead(A2); // For STEM Du RDC-104.
        A4 for RDC_103. A6 for RC_ESP32
        buffer_index++;
    }

    // print out the value you read:
    //Serial.println(sensorValue);

    if( buffer_full == true ) {

        // Run the input buffer through the filter
        output = doFilter();

        // We are going to transmit as an integer
        // Move up the decimal place
        //output *= 1000;
    }
}
```



```

//output -= 50;
output *= 1000;
output -= 600; // Offset tuning
//Serial.print( (int)output );
//Serial.print(",");
output = 12000/sqrt(output); // Amplify
Serial.println( (int)output );

// Reset our buffer and interrupt routine
buffer_index = 0;
buffer_full = false;
}

if(digitalRead(12)<1){
    analogWrite(11,0);
}
else{
    int slider = analogRead(A3); // For STEM Du RDC-104. A5 for RDC-103.
    A7 for RDC-ESP32
    analogWrite(11,map(slider,0,1023,0,255));
}
}

// This filter looks at the previous elements in the
// input stream and output stream to compound a pre-set
// amplification. The amplification is set by a0,a1,a2,
// b1,b2. Please see the linked book, above.
double doFilter() {
    int i;
    double sum;

    // Convolute the input buffer with the filter kernel
    // We work from 2 because we read back by 2 elements.
    // out[0] and out[1] are never set, so we clear them.
    out[0] = out[1] = 0;

    for( i = 2; i < BUFFER_SIZE; i++ ) {
        out[i] = a0 * buf[i];
        out[i] += a1 * buf[i-1];
        out[i] += a2 * buf[i-2];
        out[i] += b1 * out[i-1];
        out[i] += b2 * out[i-2];
    }

    // Bring all the output values above zero
    // To get a well reinforced average reading.
    for( i = 2; i < BUFFER_SIZE; i++ ) {
        if( out[i] < 0 ) out[i] *= -1;
        sum += out[i];
    }
    sum /= BUFFER_SIZE -2;

    return sum;
}

```

③スライダ Slider

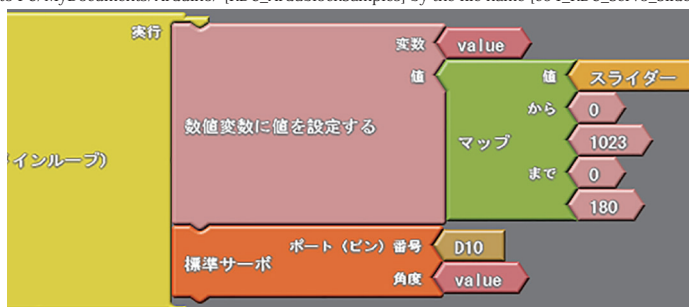
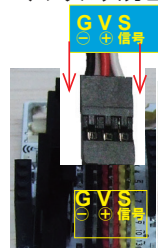
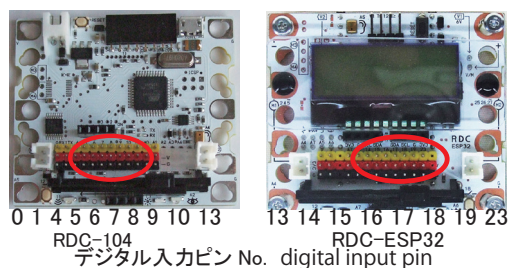
サーボ II(デジタル入力ピン接続)・・・スライダの位置に応じてサーボが回転

Servo II (digital input pin connection)・・・Servo rotates according to the location of the slider.

● ArduBlock サンプルプログラム Sample program of ArduBlock: Servo_Slider

スケッチブック >RDC_samples>c04_RDC_Servo_Slider ・ArduBlock ソースコードは、PC/MyDocuments/Arduino/へ配置したサンプルフォルダ [RDC_ArduBlockSamples] に、ファイル名【c04_RDC_Servo_Slider.abp】で格納されています。

* ArduBlock source cord is stocked in the sample folder arranged to PC/MyDocuments/Arduino/ [RDC_ArduBlockSamples] by the file name [c04_RDC_Servo_Slider.abp].



| | |
|----------------------------------|---|
| STEM Du 出力系の標準サーボブロックでサーボを動かします。 | A servo is moved at a standard servo block of the STEM Du output subpopulation. |
| スライダの値をサーボの駆動範囲にマッピングして出力します。 | A data value of a slider is output. The mapping is made the drive reach of the servo. |

■ Arduino C のサンプルプログラム Sample program of Arduino C: Knob

スケッチの例 > Servo > Knob (スライダの位置に応じてサーボが回転する)

(A servo rotates according to the location of the slider.)

myservo.attach(13) のピン番号をサーボを接続したピンの番号に変更する。

potpin = 0 のピン番号を 0 から A3 に変更する。

```
#include <Servo.h>
Servo myservo; // create servo object to control a servo

int potpin = A3; // analog pin used to connect the potentiometer

int val; // variable to read the value from the analog pin

void setup()
{
  myservo.attach(13); // attaches the servo on pin 9 to the servo object
}

void loop()
{
  val = analogRead(potpin); // reads the value of the potentiometer (value
  // between 0 and 1023)
  val = map(val, 0, 1023, 0, 179); // scale it to use it with the servo (value
  // between 0 and 180)
  myservo.write(val); // sets the servo position according to the scaled
  // value
  delay(15); // waits for the servo to get there
}
```

RDC104= ピン番号をスライダピン番号 A3 に変更する。 The pin number is changed in slider pin number A3.

RDCESP32= ピン番号をスライダピン番号 A7 に変更する。 The pin number is changed in slider pin number A7.

RDC104= ピン番号をサーボ接続したピン番号に変更する It's changed in the pin number which did servo juncture of the pin number.

RDCESP32= ピン番号をサーボ接続したピン番号に変更する It's changed in the pin number which did servo juncture of the pin number.

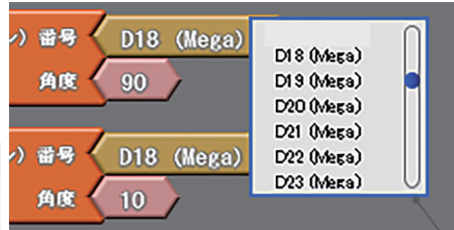
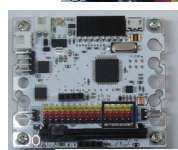
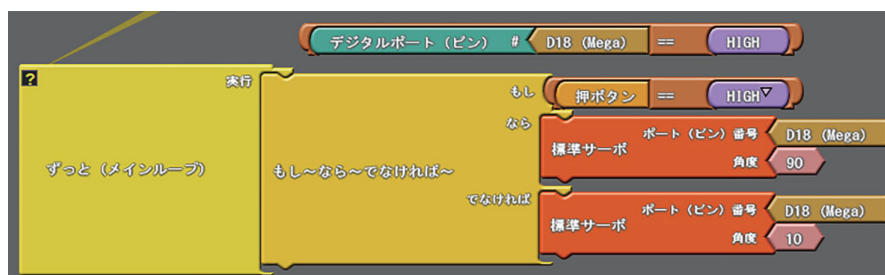
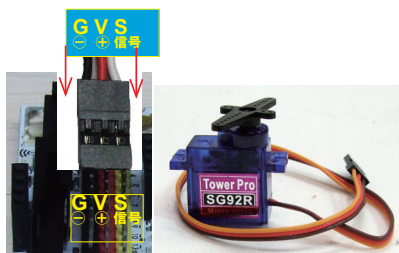
15.12.2. 別売オプション利用 Option use example

①サーボ (アナログ入力ピン接続) Servo connected to an analog input pin terminal

● ArduBlock サンプルプログラム Sample program of ArduBlock: **ServoAnalogin**

・ArduBlock ソースコードは、PC/MyDocuments/Arduino/へ配置したサンプルフォルダ [RDC_ArduBlockSamples] に、ファイル名【b09_RDC_ServoAnalogin.abp】で格納されています。

* ArduBlock source cord is stocked in the sample folder arranged to PC/MyDocuments/Arduino/ [RDC_ArduBlockSamples] by the file name [b 09_RDC_ServoAnalogin.abp].

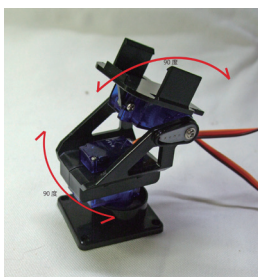


STEM Du 出力系の標準サーボブロックでサーボを動かします。
ボタンやアナログポートにセンサをつないで、入力が HIGH か LOW かで角度を変えます。

Servo is moved at a standard servo block of the STEM Du output subpopulation.

Sensor is connected with a button and an analog port, and the angle is changed by whether input is HIGH or LOW.

2 軸サーボ用クランプの例
An example of a clamp for 2 axis servos



アナログ端子：プルダウン設定
Analog terminal: pulldown setting

| | |
|----|-----------|
| A0 | D18(Mega) |
| A1 | D19(Mega) |
| A2 | D20(Mega) |
| A3 | D21(Mega) |
| A4 | D22(Mega) |
| A5 | D23(Mega) |

◆ Arduino で変換した C プログラム

```
#include <STEMDu.h>
```

```
#include <Servo.h>
```

```
STEMDu _STEMDU_robot = STEMDu();
```

```
Servo servo_pin_18 setup()
```

```
{
    servo_pin_18.attach(18, 530, 2600);
}
```

```
void loop()
```

```
{
    if ( ( _STEMDU_robot.readPush() ) == ( HIGH ) )
    {
        servo_pin_18.write( 90 );
    }
    else
    {
        servo_pin_18.write( 10 );
    }
}
```

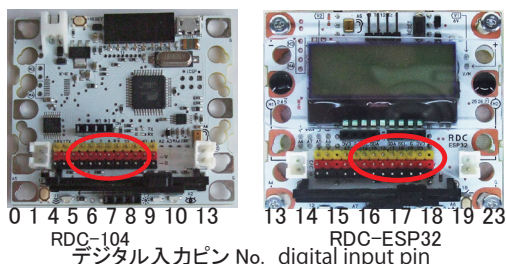

サーボ (デジタル入力ピン接続)

Servo connected to an Dijital input pin terminal

● ArduBlock サンプルプログラム Sample program of ArduBlock: **ServoDigitalin**

・ArduBlock ソースコードは、PC/MyDocuments/Arduino/へ配置したサンプルフォルダ [RDC_ArdublockSamples] に、ファイル名【b06a_RDC_Servo.abp】で格納されています。

* ArduBlock source cord is stocked in the sample folder arranged to PC/MyDocuments/Arduino/ [RDC_ArdublockSamples] by the file name [b 06a_RDC_Servo.abp].

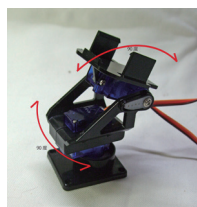


** サーボケーブルには、極性があります。ピンヘッダー(接続コネクタ)のG 黒色⊖、V 赤色 ⊕、S(信号)黄色をあわせて差し込みます。

A servo cable has polarity. The one of the connector Ground->Black⊖, Voltage ->Red⊕, and Signal ->Yellow is put in all together.

2 軸サーボ用クランプの例

An example of a clamp for 2 axis servos



STEM Du 出力系の標準サーボブロックでサーボを動かします。

単に待ち時間を入れるとサーボ固有の速度で指定位置まで動きます。

リピートを使って細かいステップごとに待ち時間を入れるとゆっくり動かすことができます。

Servo is moved at a standard servo block of the STEM Du output subpopulation.

When the waiting time is just put in, a servo moves to the designation location with peculiar speed.

When the waiting time is put in every small step using repeat, it can be moved slowly.

■ Arduino C のサンプルプログラム サーボが回転する Sample program of Arduino C: **Sweep**

スケッチの例 > Servo > Sweep (サーボが回転する)

myservo.attach(9) のピン番号をサーボを接続したピンの番号に修正する。

```
#include <Servo.h>

Servo myservo; // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0; // variable to store the servo position

void setup()
{
  // The pin number is changed to the pin number connected from 9.
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180
```

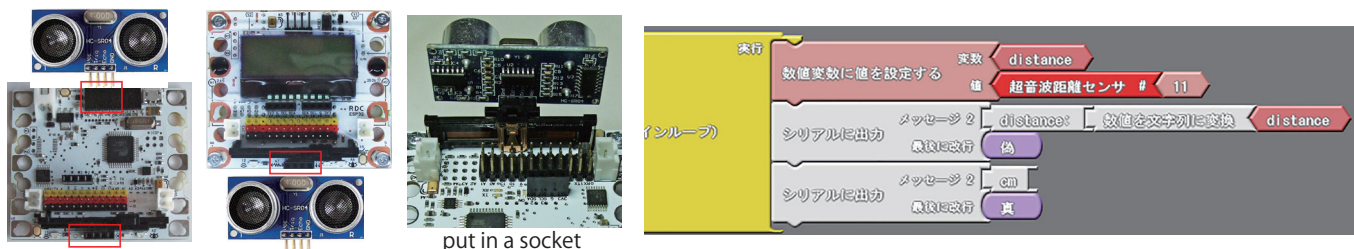
```
degrees
// in steps of 1 degree
myservo.write(pos); // tell servo to go to position in variable 'pos'
delay(15); // waits 15ms for the servo to reach the position
}
for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0
degrees
myservo.write(pos); // tell servo to go to position in variable 'pos'
delay(15); // waits 15ms for the servo to reach the position
}
}
```

②.超音波測距センサ・・・超音波距離センサで測距 By an ultrasonic range sensor, ranging.

● ArduBlock サンプルプログラム Sample program of ArduBlock: Ping

- ArduBlock ソースコードは、PC/MyDocuments/Arduino/へ配置したサンプルフォルダ [RDC_ArduBlockSamples] に、ファイル名【b03_RDC_Ping.abp】で格納されています。

* ArduBlock source cord is stocked in the sample folder arranged to PC/MyDocuments/Arduino/ [RDC_ArduBlockSamples] by the file name [b03_RDC_Ping.abp].



RDC-104

RDC-ESP32

put in a socket

| | |
|---|--|
| 超音波距離センサHC-SR04で距離を測ります。 | The distance is measured in ultrasonic range sensor HC-SR04. |
| ソケットに超音波距離センサを差し込みます。 | An ultrasonic range sensor is put in a socket. |
| ピン名をよく確認して取り付けてください。 | Please often confirm the pin name and install it. |
| (RDC-104:Echo/Triggerはどちらもピン11に接続されます) | (Where is RDC-104:Echo/Trigger also connected to pin 11?) |
| (RDC-ESP32:Echo/Triggerはどちらもピン12に接続されます。) | (Where is RDC-ESP32:Echo/Trigger also connected to pin 12.) |
| 距離をシリアルモニタに出力します。 | The distance is output for a serial monitor. |

■ Arduino C のサンプルプログラム Sample program of Arduino C: Ping

スケッチの例 > 06.Sensors > Ping (HC-SR04 超音波センサで測距する)

RDC-104:const int pingPin = 7 のピン番号を 7 から 11 に修正する。

RDC-ESP32: ピン番号を使用するセンサのピン番号に 12 に設定する。

```
// this constant won't change. It's the pin number
// of the sensor's output:
const int pingPin = 7;

void setup() {
// initialize serial communication:
Serial.begin(9600);
}

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);
```

RDC-104: ピン番号を 7 から 11 に変更 The pin number is changed in 11 from 7.

```

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 = microsecondsToInches(duration);
cm = microsecondsToCentimeters(duration);

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

delay(100);
}

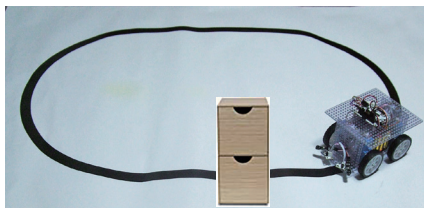
long microsecondsToInches(long microseconds) {
// According to Parallax's datasheet for the PING))) , there are
// 73.746 microseconds per inch (i.e. sound travels at 1130 feet per
// second). This gives the distance travelled by the ping, outbound
// and return, so we divide by 2 to get the distance 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;
}

```

◆超音波測距センサを障害物回避とライントレースに使う例。 Practics

The example which uses an obstacle ranging sensor for obstacle avoidance and linear trace.



【アルゴリズム】

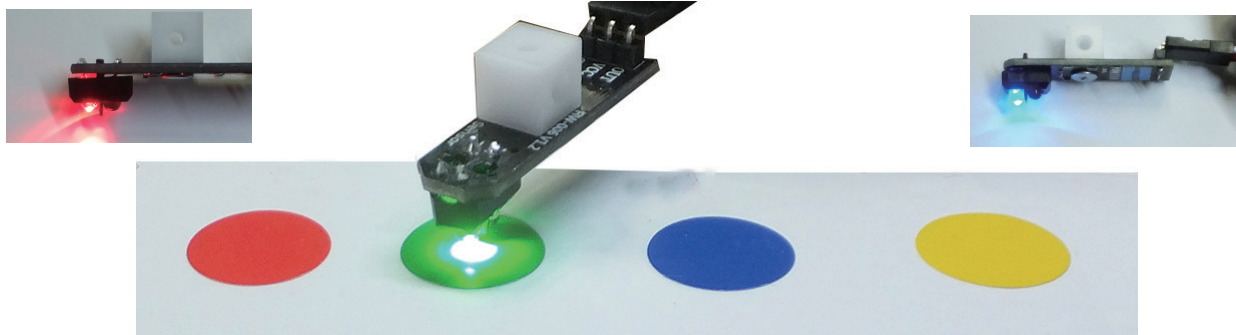
フロアセンサでライントレースをしながら前進し、超音波センサで前方に障害物を発見すると、20 cm手前で回避行動を行い、障害物の先へ回り込み、その先でライントレースを行うという行動をプログラムします。

| | |
|--|---|
| ・分岐条件になる「ラインの計測データ値」「障害物の計測データ値」など動作させる環境下での実計測データに基づき[しきい値]を決めます。 | * [Threshold value] is decided based on the actual measured data at the environment bottom which makes "the linear measured data value" "the measured data value of the obstacle" move which become a branch condition. |
| ・動作速度、回転時間なども調整し、プログラムします。 | The working speed and the rotationving hour are adjusted and programed. |
| ・環境に合わせて、実機を作りこむことが重要です。 | A production is made according to the environment, and it's important to be crowded. |
| ・動作命令「旋回」を使用していますが、「タンク移動」を使う事もできます。 | A sample of the next page is using "turning" for a movement order, but it's possible to use "tank movement". |


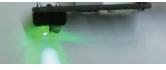

■参考プログラムは、Sec15.9.4 に掲載しています。参照ください。

Program is carried in Sec15.9.4. Please refer to it.

③ . 可視光センサ・・・色を認識 / プログラムで動作 The color is recognized , It's used for the program condition.



色別可視光センサによる反射光測定値 The catoptric light measure by the color another light sensor.

| 対象物との距離5 mmで測定 | 赤ラベル Red | 緑ラベル Green | 青ラベル Blue | 黒ラベル Black | 黄ラベル Yellow | 白地 White |
|--|-------------|---------------|--------------|---------------|----------------|-------------|
| RDI-221Red  | 140 | 810 | 790 | 890 | 180 | 400 |
| RDI-222Green  | 670 | 310 | 680 | 680 | 85 | 95 |
| RDI-223Blue  | 830 | 750 | 630 | 810 | 760 | 410 |

可視光センサを使用することで、対象物の色を計測することが可能です。

It's possible to be to use a light sensor and measure the color of the object.

床に塗られた色を認識させ、条件分岐によるプログラム行動することが可能になります。

A program by a conditional branch makes them recognize the color applied a floor, and becomes possible to behave.

■ Arduino C のサンプルプログラム Sample program of Arduino C: **AnalogReadSerial**

スケッチの例 > 01.Basics > AnalogReadSerial (アナログ入力をシリアルモニタに出力する)
AnalogRead(A0) のピン番号を RDC104 ヘセンサ接続使用したピン番号に変更する。

```
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(1);        // delay in between reads for stability
}
```

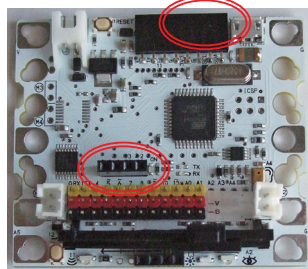
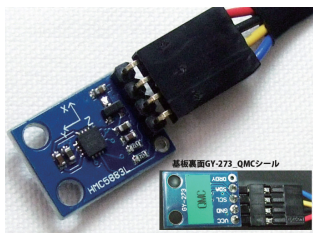
使用するピン番号に変更する

④ .I2C コンパスセンサ・・・1度単位で角度を取得できます。

● ArduBlock サンプルプログラム

電子コンパスモジュール 3 軸コンパス 磁気センサー

- ・ Block ソースコードは、PC/MyDocuments/Arduino/へ配置したサンプルフォルダ [RDC_ArduBlockSamples] に、ファイル名【b08_RDC_QMC5883.abp】で格納されています。



RDC-104

| マイコンボード | コンパスセンサ |
|---------|---------|
| 3V3 | VCC |
| 3SCL | SCL |
| 2SDA | SDA |
| G | GND |

コンパスセンサはピンの表示を合わせて I2C ソケットに挿します。

！コンパスセンサブロックは製作中です。（エラーが出てコンパイルできません）

！C 言語のサンプルを使用してください。

| X 方位 | 出力値 | 分解能 |
|------|-----|----------|
| 北 N | 0 | 360 |
| 東 E | 90 | 1 度単位で出力 |
| 南 S | 180 | |
| 西 W | 270 | |



STEM Du アクセサリのコンパスセンサブロックを使います。QMC5883 コンパスセンサの値をシリアルモニタに出力します。

Serial print例

Heading = 3.76 Degress = 215.19
 Heading = 3.02 Degress = 172.90
 Heading = 0.81 Degress = 46.65

■ Arduino C のサンプルプログラム Sample program of Arduino C: QMC5883_compass

スケッチの例 >[DFRobot_QMC5883]>【QMC5883_compass】で格納されています。

■同じ場所に 3 軸を計測するサンプルソースコード【QMC5883_readRaw】が格納されています。

```

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

```

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

DFRobot_QMC5883 compass;

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

    if(compass.isHMC()){
        Serial.println("Initialize HMC5883");
        compass.setRange(HMC5883L_RANGE_1_3GA);
        compass.setMeasurementMode(HMC5883L_CONTINUOUS);
        compass.setDataRate(HMC5883L_DATARATE_15HZ);
        compass.setSamples(HMC5883L_SAMPLES_8);
    }
    else if(compass.isQMC()){
        Serial.println("Initialize QMC5883");
        compass.setRange(QMC5883_RANGE_2GA);
        compass.setMeasurementMode(QMC5883_CONTINUOUS);
        compass.setDataRate(QMC5883_DATARATE_50HZ);
        compass.setSamples(QMC5883_SAMPLES_8);
    }
}

void loop()
{
    Vector norm = compass.readNormalize();

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

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

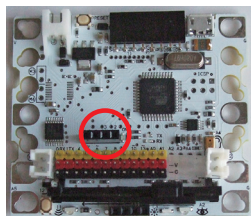
    // Correct for heading < 0deg and heading > 360deg
    if (heading < 0){
        heading += 2 * PI;
    }
}
```


⑤ . 加速度 / ジャイロセンサ ・ ・ ・ MPU-6050 シリアルモニタ

● ArduBlock サンプルプログラム

・ Block ソースコードは、PC/MyDocuments/Arduino/へ配置したサンプルフォルダ [RDC_ArduBlockSamples] に、ファイル名【b09_RDC_MPU6050.abp】で格納されています

STEM Du 入力系のブロックを使います。加速度／ジャイロセンサ MPU6050 からシリアルモニタに出力します。



RDC-104

Ⓐ RDI-9250 4本のヘッダーピンで取り付け使用します。

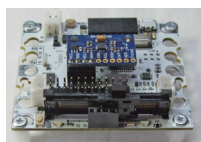


VCC, GND, SCL, SDA コントローラソケットの記号と合わせて差し込み接続をして使用します。

！ 加速度センサブロックは、現状プログラムの中で一つの値しか出力することができません。

！ 複数入れるとコンパイルでエラーになります。

！ 複数のパラメータを使用する場合は C 言語で記述してください。



接続状態図

■ Arduino C のサンプルプログラム Sample program of Arduino C: **MPU6050_test**
スケッチブック >RDC_samples>c07b_RDC_MPU6050_test

```
// MPU-6050 Short Example Sketch
// By Arduino User JohnChi
// August 17, 2014
// Public Domain
#include<Wire.h>
const int MPU=0x68; // I2C address of the MPU-6050
int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;
void setup() {
    Wire.begin();
    Wire.beginTransmission(MPU);
    Wire.write(0x6B); // PWR_MGMT_1 register
    Wire.write(0); // set to zero (wakes up the MPU-6050)
    Wire.endTransmission(true);
    // Wire.beginTransmission(MPU);
    // Wire.write(0x1B); //FS_SEL register selects the full scale range of the gyroscope
    // outputs
    // Wire.write(0); // set to +-250deg/s 00011000 +-2000deg/s
    // Wire.endTransmission(true);
    Serial.begin(9600);
}
void loop() {
    Wire.beginTransmission(MPU);
    Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
```

```

Wire.endTransmission(false);
Wire.requestFrom(MPU, 14, true); // request a total of 14 registers
AcX=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
AcY=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
AcZ=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)

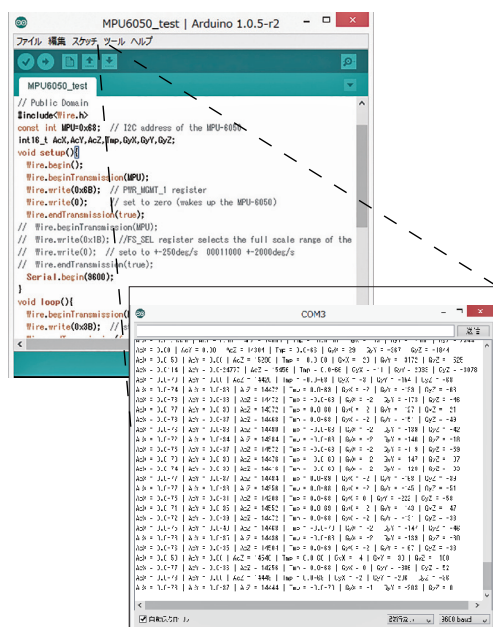
Tmp=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
GyX=Wire.read()<<8|Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
GyY=Wire.read()<<8|Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
GyZ=Wire.read()<<8|Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)

Serial.print("AcX = "); Serial.print(AcX / 1638.4);
Serial.print(" | AcY = "); Serial.print(AcY*0.0054);
Serial.print(" | AcZ = "); Serial.print(AcZ);
Serial.print(" | Tmp = "); Serial.
print(Tmp/340.00+36.53); //equation for temperature in degrees C from datasheet
Serial.print(" | GyX = "); Serial.print(GyX / 131);
Serial.print(" | GyY = "); Serial.print(GyY);
Serial.print(" | GyZ = "); Serial.println(GyZ);
delay(333);
}

```

3). センサーデータ計測

1. 【MPU6050_test.ino】を書き込んだマイコンボードのプログラムが実行されている状態の時に、Arduino 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)



2. Arduino の [ツール] → [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データ収集を行います。(USB ケーブルは接続のまま)
4. USB ケーブル接続を外して、データ計測を停止します。シリアルモニターウィンドウの計測値表示が停止しますので、確認が容易になります。
5. 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分かり易くなります。
6. 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。

図は、ArduinoC のサンプル MPU6050test でのシリアルモニターです。

