

「ビュートローバー」「ビュートローバー ARM」

「ビュートミニ ARM」

専用ソフトウェア

ビュートビルダー2

取扱説明書

ヴイストーン株式会社

Version1.3

説明書の読み方について

本説明書は、主にソフトウェアをはじめて使う方のために、基本説明や取り扱い方法を手順に沿って説明する「チュートリアル」と、ソフトウェアの機能について網羅した「リファレンス」の二部構成となっています。以下の案内を参照し、目的に応じてどちらを読むかお選びください。

- 以下に該当する場合、チュートリアル（4～36 ページ）をご覧ください
 - [ビュートビルダー2 やロボット本体を始めて使う方](#)
 - [プログラミングを始めて経験する方・プログラミング初心者](#)
- 以下に該当する場合、リファレンス（37 ページ～）をご覧ください。
 - [チュートリアルを読んでビュートビルダー2 の使い方を一通り理解できた方](#)
 - [モータの速度や拡張機器など、ロボットに関する設定を変更したい方](#)
 - [上級者向けのプログラミング機能を利用したい方](#)

ビュートビルダー2 取扱説明書

チュートリアル

目次

1. はじめに.....	4
1-1 各部の名称.....	4
1-2 動作環境.....	5
1-3 ソフトウェアのインストール.....	7
インストール不要版の起動.....	7
インストーラ版のインストールと起動.....	8
2. プログラムを作ってみよう.....	10
2-1 プログラム画面について.....	10
2-2 ロボットの接続.....	10
2-3 プログラムの作成.....	12
ブロックの追加.....	12
ブロックの削除.....	13
ブロックのつなぎ方.....	13
プログラムの書き込み方.....	14
プログラムの実行.....	15
プログラムの保存・読み込み.....	16
モータのスピード調整.....	17
3. いろいろな機能を使ってみよう.....	18
3-1 モータブロックの使い方.....	18
繰り返しの使い方.....	20
3-2 ブザーとLEDの使い方.....	22
ブザーの使い方.....	22
LEDの使い方.....	23
「続ける」の使い方.....	24
3-3 センサを使った条件分岐.....	26
センサ値の確認.....	26
条件分岐の使い方.....	27
3-4 エンコーダを使った移動.....	30
エンコーダを有効にする.....	30
エンコーダを使った車輪制御の使い方.....	32
4. 次の課題にチャレンジ！.....	33
4-1 高度な機能を使う.....	33
様々な命令を使う.....	33
モータの速度や回転方向を変える.....	33

Bluetooth や IXPBUS などの拡張機器を使用する	33
ファームウェアの書き換えを行う	33
C 言語でプログラミングする	33
4-2 拡張機能を使う	34
落下防止	34
障害物回避	34
ライントレース	35
迷路	35
無線操縦	36
群制御	36
5. プログラムランドとの連携について	37

1. はじめに

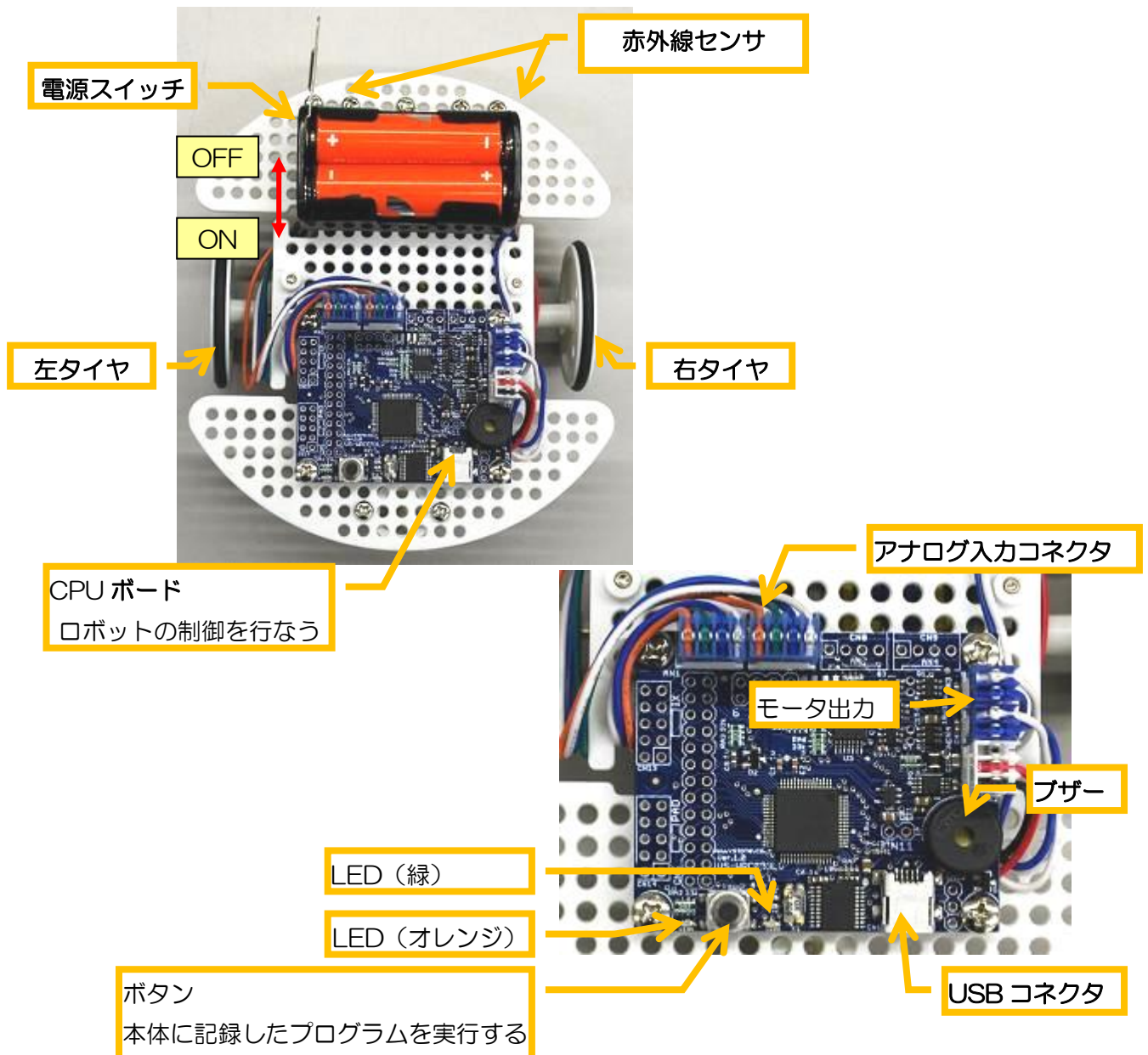
この度は、「ビュートローバー」「ビュートローバーARM」(以下、「ロボット」及び「ロボット本体」と記述)をお買い上げいただきありがとうございます。ビュートビルダー2(以下、「本ソフトウェア」と記述)は、ロボットをPCと接続してロボットのプログラムを作成するためのソフトウェアです。

チュートリアルでは、プログラミング初心者の方や、初めて本ソフトウェアを利用される方のために、操作方法を簡単に説明しています。まずは、本説明書を読みながら、ロボットを実際に触ってみて基本的な使い方を学んでいきましょう。

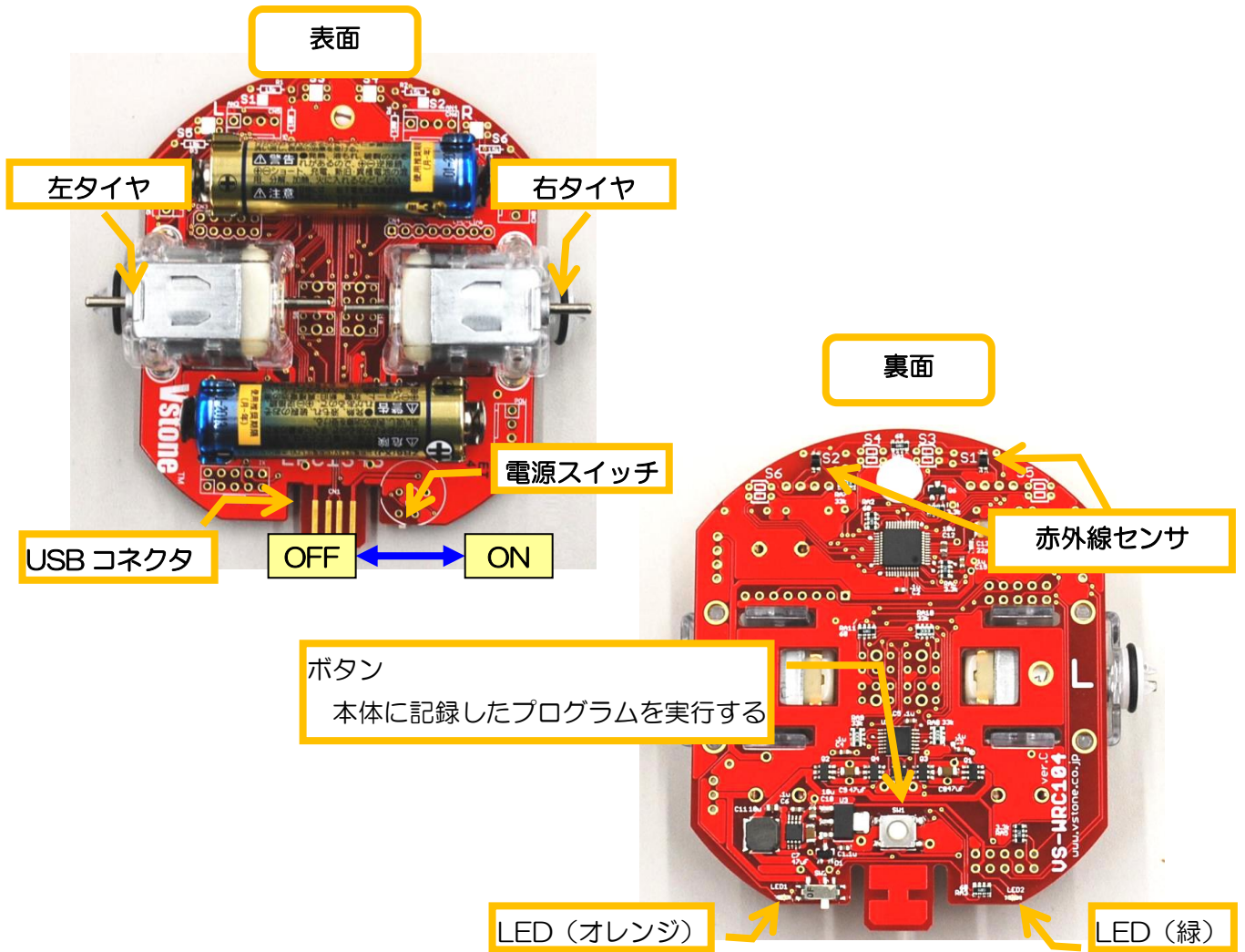
1-1 各部の名称

ロボットの各部の名称について説明します。

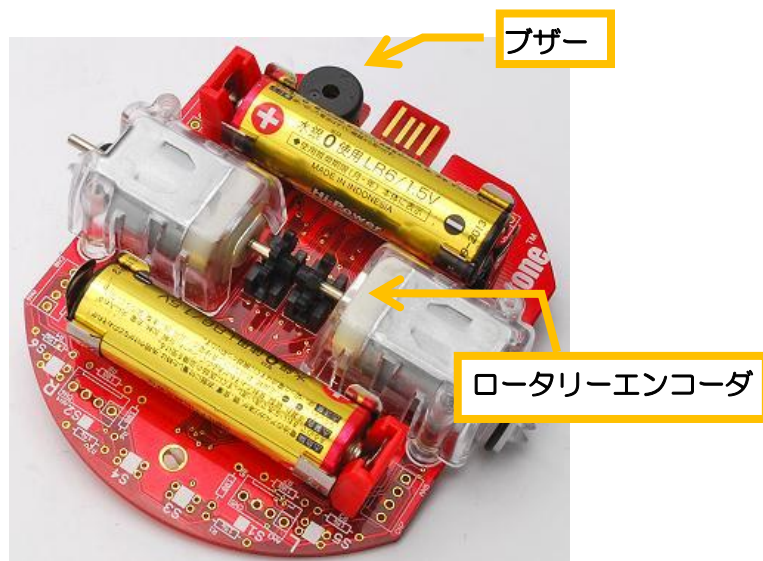
「ビュートローバー」「ビュートローバーARM」は、共に同じ名称です。



ビュートミニ ARM の各部の名称は以下の通りです。



ビュートミニ ARM エンコーダ・ブザー搭載版は、以下の場所にロータリーエンコーダ・圧電ブザーを搭載しています。



1-2 動作環境

本ソフトウェアは下記の環境の PC で動作します。現在、市販されている PC であればほとんどの PC で動作します。

- OS : Windows2000(SP4)/XP/Vista/7
- CPU : Pentium-Ⅲ以降 (1GHz 以上推奨)
- RAM : 128MB
- インターフェース : USB ポート
- 画面サイズ : XGA(1024×768)以上

1-3 ソフトウェアのインストール

ビュートビルダー2には2種類のバージョンがあります。

• インストーラ版

一般的なソフトウェアのように、インストーラを使ってPCでソフトウェアを利用できるようにします。インストーラを使った場合、デスクトップへのショートカットの作成や、スタートメニューへの登録をしてくれるので、ご家庭での使用や、学校のPCにインストール制限がない場合にはこちらがお勧めです。

• インストール不要版

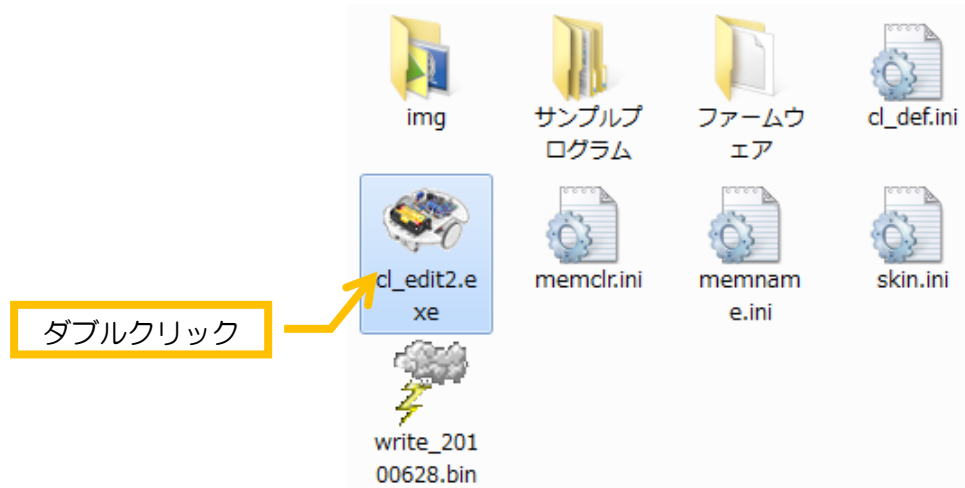
学校のPCでは、インストール制限がかかっていることが多くあります。その場合は、CD-ROMからコピーするだけで使用できる、インストール不要版をご使用ください。ソフトウェアの機能はインストーラ版と同じです。USBメモリなどにコピーして使うと、インストールせずにどのPCでも気軽に利用できるようになります。

インストール不要版の起動

ビュートビルダー2のインストール不要版をCD-ROMからインストールする場合、CD-ROM内の「ソフトウェア¥インストール不要版フォルダ」にある「BeautoBuilder2_***」フォルダをPCにフォルダごとコピーしてください。（インストーラのバージョンによりファイル中の数字が異なる場合があります）



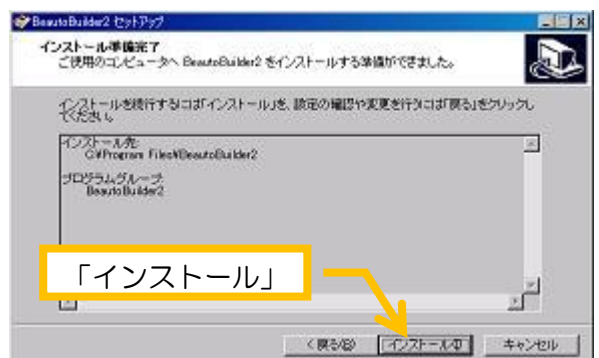
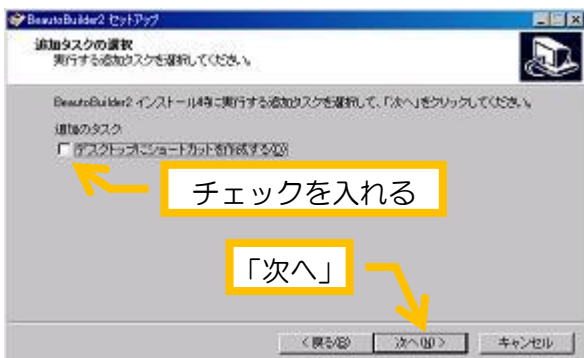
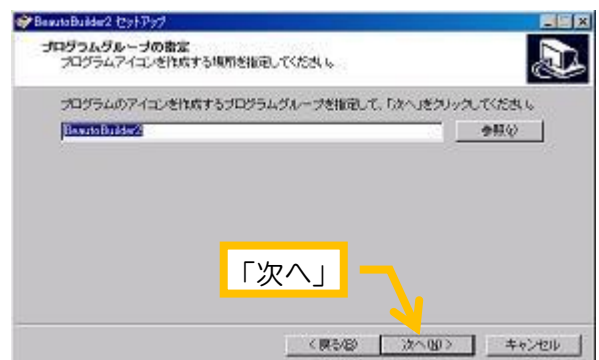
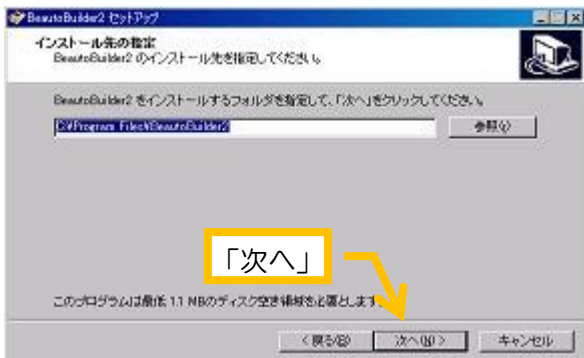
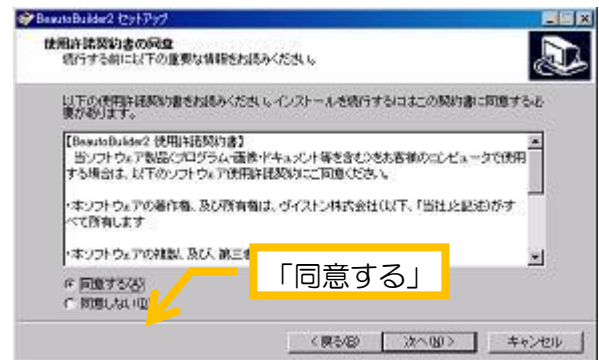
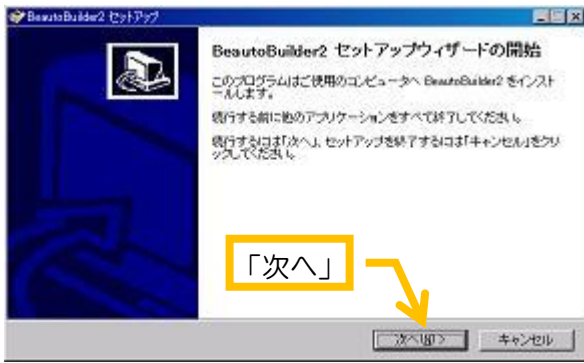
ビュートビルダー2を起動するには、PCにコピーしたフォルダ内にある「cl_edit2.exe」（環境によって、exeは省略されます）をダブルクリックします。起動したら「2. プログラムを作ってみよう」に進んでください。

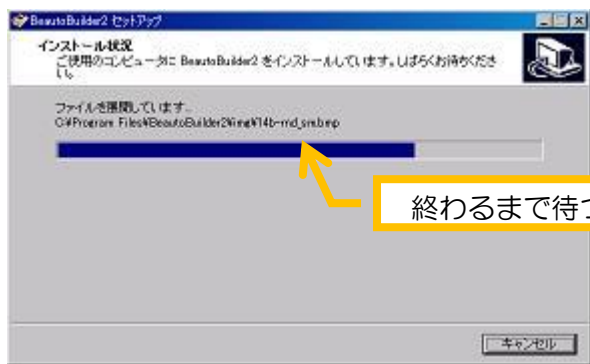


インストーラ版のインストールと起動

ビュートビルダー2をCD-ROMからインストールする場合、CD-ROM内のソフトウェアフォルダにある「BeautoBuilder2_Inst_***.exe」をダブルクリックしてください。ダブルクリックするとインストールが始まります。以降は、図の通りの順番で作業を進めてください。

(インストーラのバージョンによりファイル中の数字が異なる場合があります)





終わるまで待つ

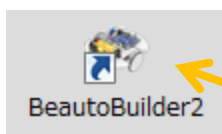


「完了」

これでビュートビルダー2 のインストーラ版でのインストールは完了です。

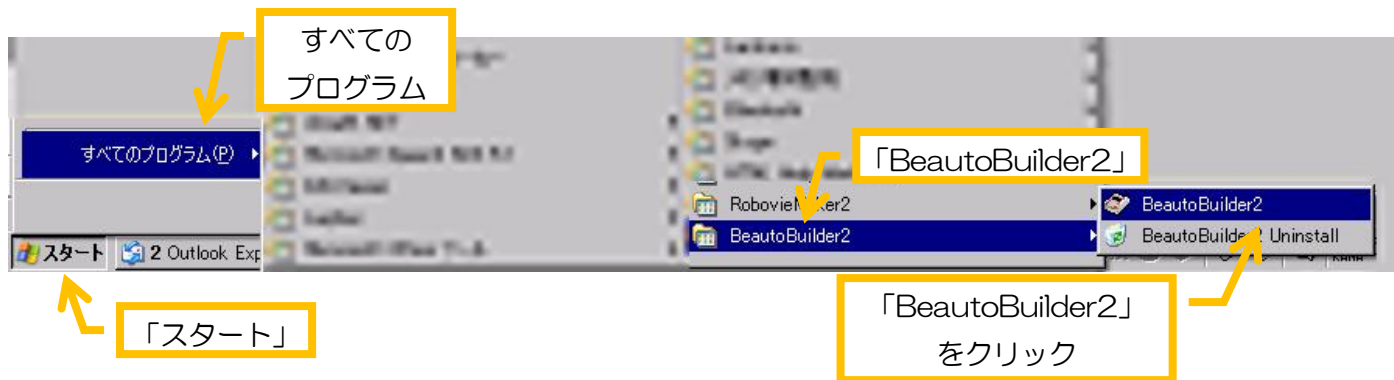
ソフトウェアの起動は、デスクトップのショートカットをダブルクリックするか、OS のスタートメニューから「BeautoBuilder2」を選んでクリックします。

デスクトップからは・・・



ショートカットアイコンをダブルクリック

スタートメニューからは・・・



すべての
プログラム

「BeautoBuilder2」

「スタート」

「BeautoBuilder2」
をクリック

2. プログラムを作ってみよう

2-1 プログラム画面について

ビュートビルダー2の画面は、4つのエリアに分かれます。

プログラムエリア

プログラムエリアでは、「フローチャート」のような形式でアクションブロックをおいていき、それを矢印で結んでプログラムを作成するエリアです。ここで作成したプログラムを、実際にロボットで動作させることができます。

アイコンエリア

プログラムに使用する「アクション」(命令)を選択するエリアで、動作を指示したり、センサによって動作を決めるためのブロックが表示されています。

設定エリア

アクションの詳細を設定するエリアで、各アクションによってさまざまな設定ができます。

センサエリア

ロボットに接続されているセンサ情報を表示するエリアで、センサの値がリアルタイムで確認できます。

The screenshot displays the Beauto Builder 2 software interface. The main workspace shows a flowchart program starting with a '開始' (Start) block, followed by an 'LED 1.00秒...' block, a 'LOOP ID' block, a decision diamond 't1 < 2000', and two action blocks '前' (Forward) and '右前' (Right Forward). The flowchart ends with a '終了' (End) block. The interface is divided into four main areas, each highlighted with a yellow box and an arrow:

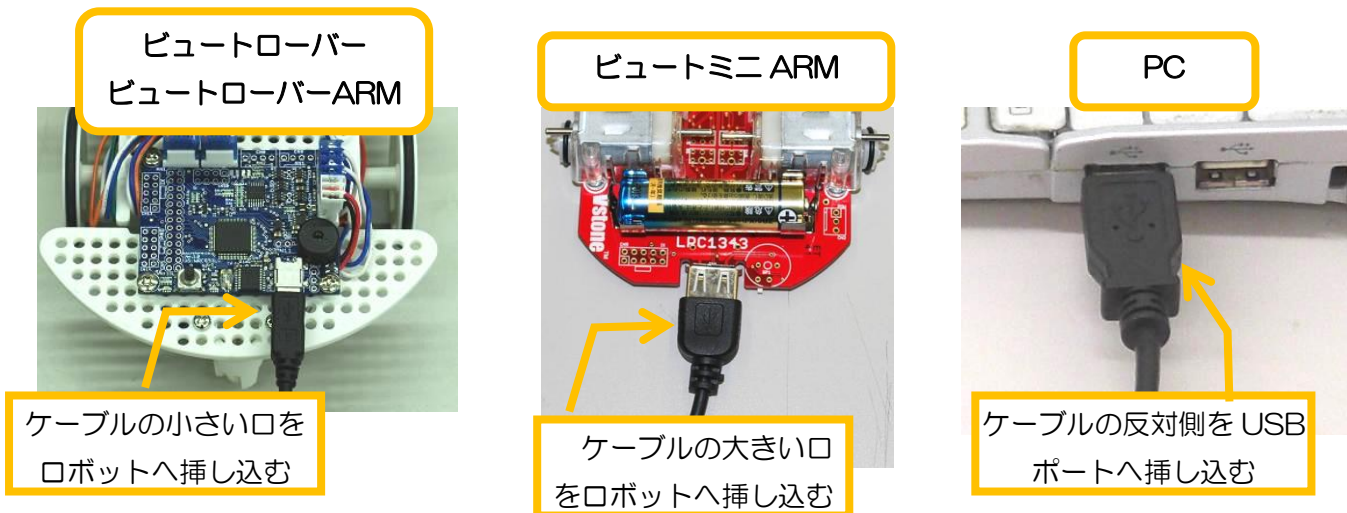
- 「アイコンエリア」** アクションを選択 (Icon Area): A grid of various action blocks like 'LED', 'LOOP', 'IF', 'AND/OR', and movement blocks.
- 「設定エリア」** アクションの詳細を設定 (Setting Area): A panel for configuring the selected action, including a '続ける' (Continue) checkbox and a '時間を決める' (Set time) field.
- 「センサエリア」** センサの数値を表示 (Sensor Area): A panel showing real-time sensor values for 'センサ1' through 'センサ4' and 'ボタン' (Button).
- 「プログラムエリア」** ブロックを組み合わせてプログラムを作成 (Program Area): The central workspace where the flowchart is built.

2-2 ロボットの接続

ロボットのプログラミングをおこなう場合、まずPCにロボットを接続します。

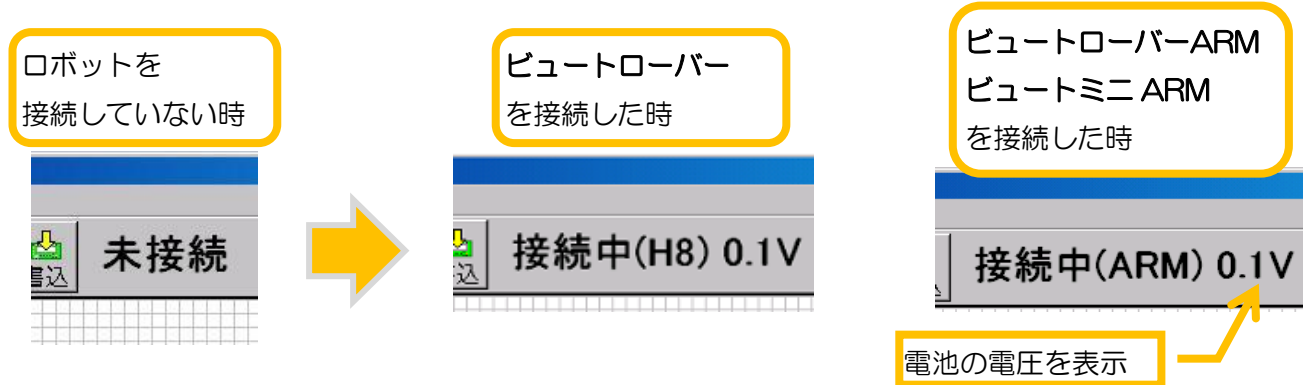
接続は、PCにケーブルの大きい端子を、ロボットへは小さい端子を挿し込みます。どちらも向きがあるので注意してください。

※ ロボットにケーブルを接続して通信する場合（または、書き込みをする場合）、電源が入ってなくてもUSBから電源が供給されるので、通信、書き込みが行えます。



PCとロボットを接続すると、本ソフトウェアは自動的にロボットと通信を開始します。

ロボットと通信できると、画面左上の表示が「未接続」から「接続中(xxx) ??V」に変わります。



数値は現在の電池の電圧を表し、「接続中」の後の文字列は、接続しているロボットの機種を表します。ロボットの機種は「ブートローバー」の場合は「接続中(H8)」、「ブートローバー-ARM」「ブートミニ ARM」の場合は「接続中(ARM)」という表記になります。

※C言語でロボットのプログラムを書き換えた場合

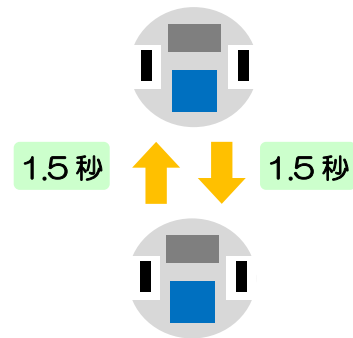
C言語で作成した自作プログラムをロボットに書き込むと、本ソフトウェアと通信できなくなります。改めて本ソフトウェアでロボットをプログラミングしたい場合は、本ソフトウェアから「ファームウェアの書き換え」を行ってください。詳しい手順については、リファレンスの「3-4.ファームウェアの書き換え」をご覧ください。

2-3 プログラムの作成

それでは、実際に簡単なプログラムを作成しながら、本ソフトウェアの使い方を覚えて行きましょう。最初はモータを動かしてロボットを移動させるプログラムを作りましょう。

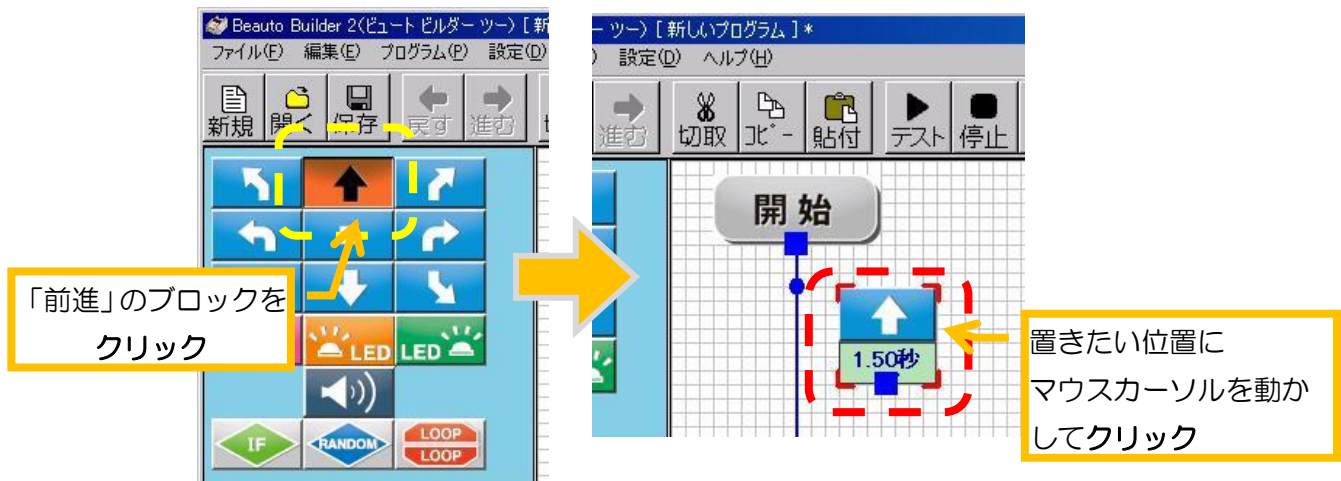
問題 1

- 1.5秒前進
 - 1.5秒後退
- の順番で、ロボットが移動するプログラムを作成しましょう。

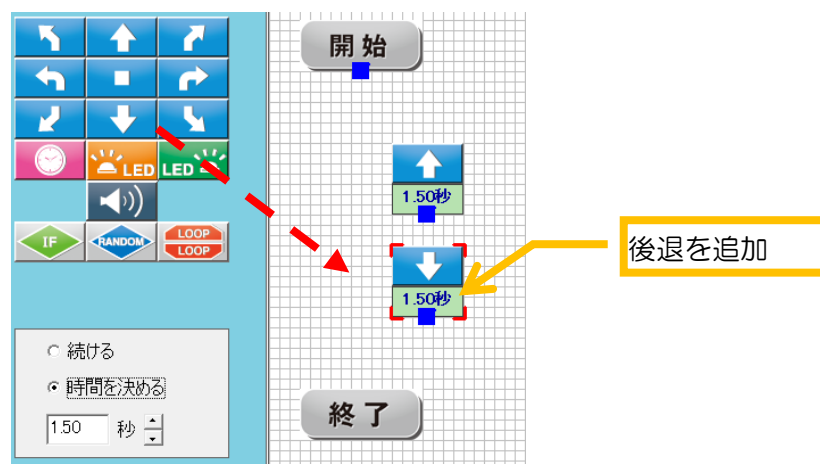


ブロックの追加

ロボットのモータを動かす場合は車輪制御のブロックを使います。
まずは以下の手順で、「前進」のブロックをプログラムに追加してみましょう。



このように、「追加するブロックの種類」→「ブロックを置く場所」の順番で画面をクリックすると、新しいブロックを追加できます。同様に「後退」のブロックも追加しましょう。

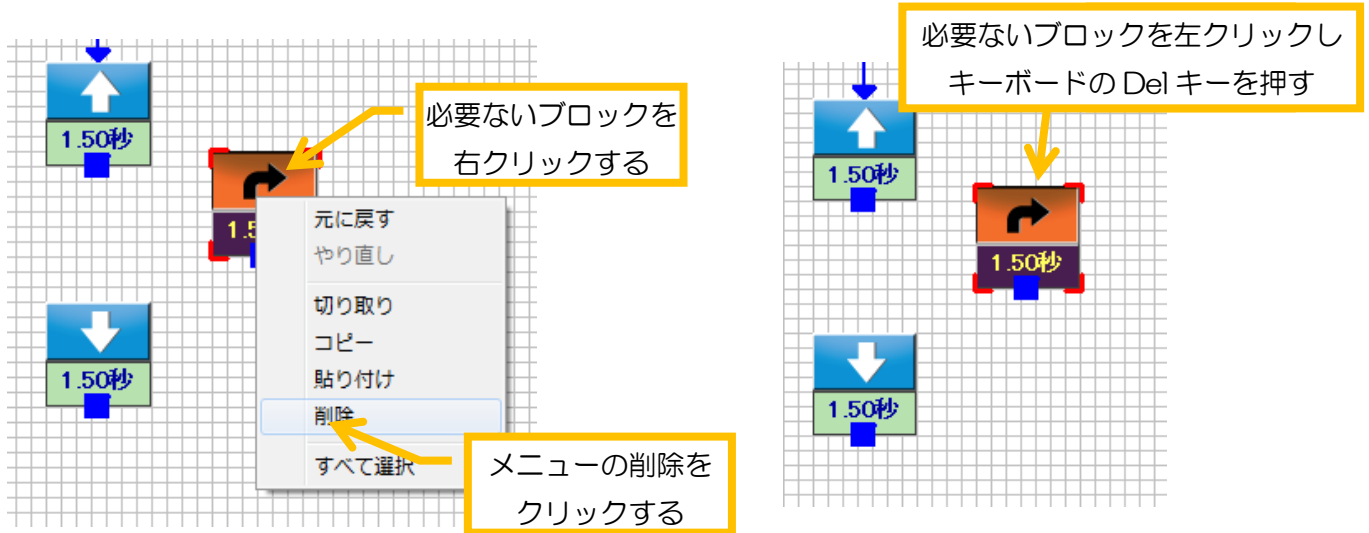


ブロックの削除

プログラムを作成していると、まちがって置いてしまったブロックや、使わなくなったブロックを削除したいことがあります。

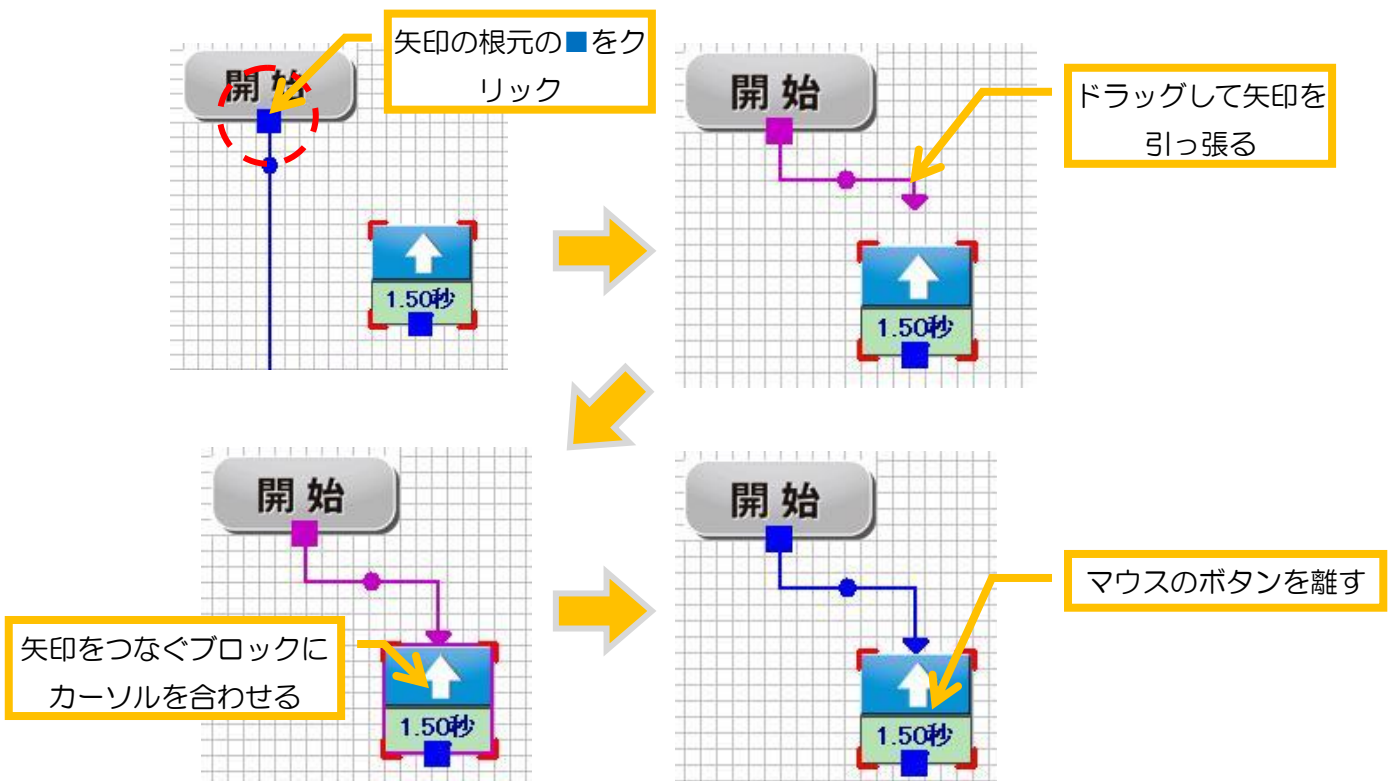
その場合は、以下の手順でブロックの削除を削除しましょう。

削除するには2通りの方法があります。

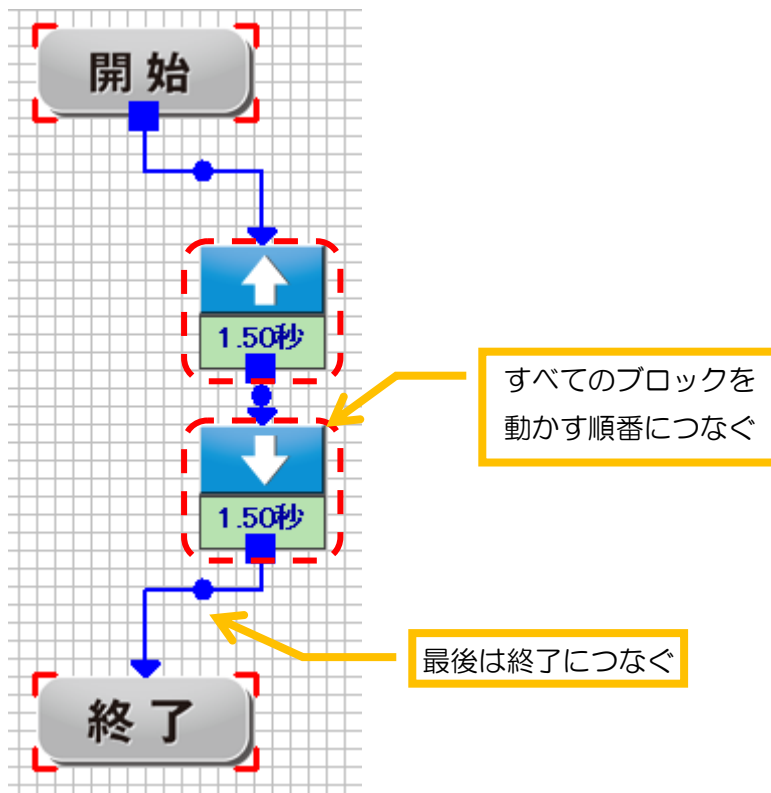


ブロックのつなぎ方

ブロックを追加したら、それを実行する順番に矢印でつないでいきます。ブロックをつながないと、せっかく追加したブロックをロボットが実行しなくなるので注意してください。



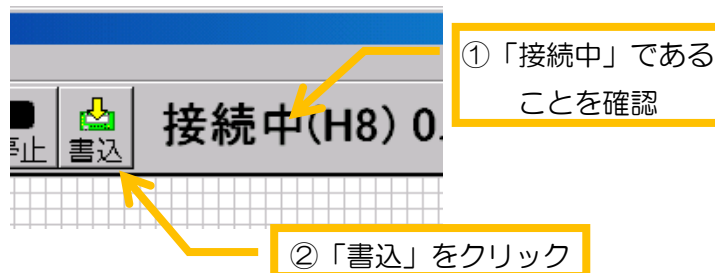
では、すべてのブロックを図と同じように接続しましょう。



プログラムの書き込み方

ブロックの接続が完了したら、プログラムをロボットに書き込んで動かしてみましょう。

まず、PCとロボットを接続して、通信が始まるまで待ちます。(画面に「接続中」と表示されるまで待つ)
通信が始まったら、ツールバーの「書込」ボタンをクリックしてください。



クリックすると、書き込みが開始されて、下のメッセージが表示されます。メッセージ内の青いバーが右端まで届いたら書き込み完了です。完了すると自動的にメッセージが消えます。

⚠ プログラムの書き込み中は、通信ケーブルを外したり、ボタンを押したりしないようにしてください。

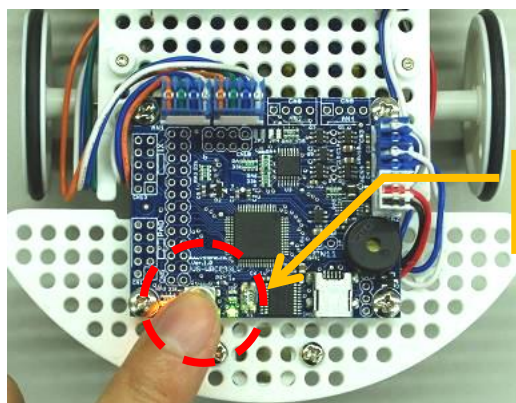
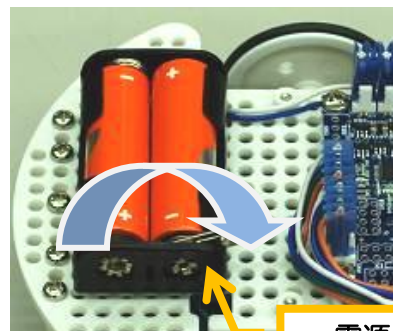
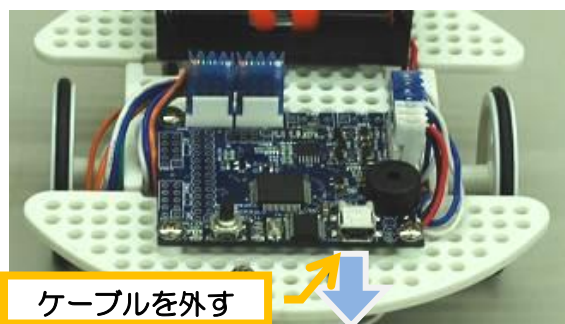


ビュートローバーよりもビュートローバーARM・ビュートミニ ARMの方がロボットに書き込めるプログラムのサイズが大きいため、ビュートローバーARMで書き込めるプログラムがビュートローバーでは書き込めない場合があります。

プログラムの実行

それでは、実際にロボットに書き込んだプログラムを実行させて見ましょう。

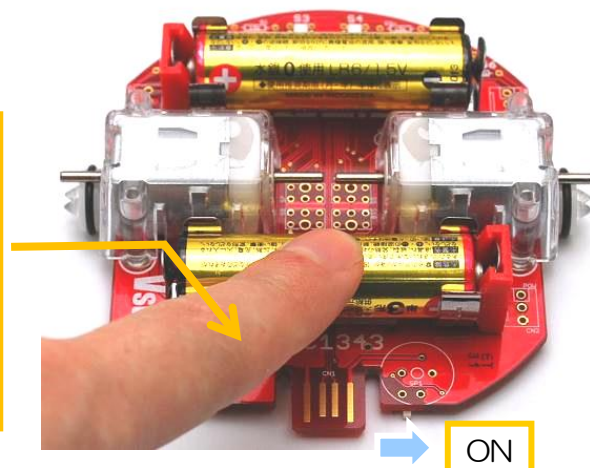
ビュートローバー・ビュートローバーARM



ビュートミニ ARM

- ① ケーブルを外す。
- ② 電源を ON にする。
- ③ 机に置いて、電池を上から軽く押す。
(裏側のボタンを間接的に押す)

裏側のボタンが押されるとプログラムスタート



プログラムを途中で終了させる場合は、ロボットの電源を OFF にしてください。また、ロボットには一度に一つのプログラムしか書き込めません。

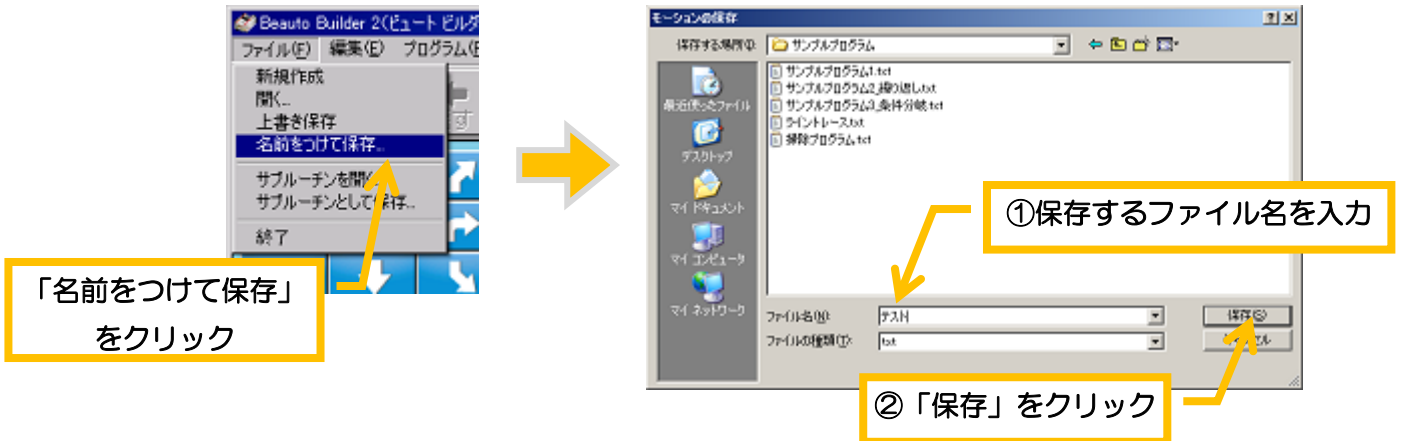
ロボットが課題の通りに動けば、正しくプログラミングできています。課題の通りに動作しない場合は、「ブロックの矢印をつなぎ忘れていないか」などを確認してみましょう。

また、作成したプログラムは次ページの手順で保存、読み込みができます。編集したプログラムが再度利用できるように、ここで一度保存をしておきましょう。

プログラムの保存・読み込み

ビュートビルダー2で作成したプログラムは、PCにファイルとして保存することができます。プログラムをファイルに保存する場合は、メニューより「ファイル」→「名前をつけて保存」をクリックするか、ツールバーの「保存アイコン」をクリックしてください。

クリックするとファイルを保存するフォルダと名前を設定するダイアログを開くので、保存先と名前を設定してダイアログの「保存」をクリックしてください。



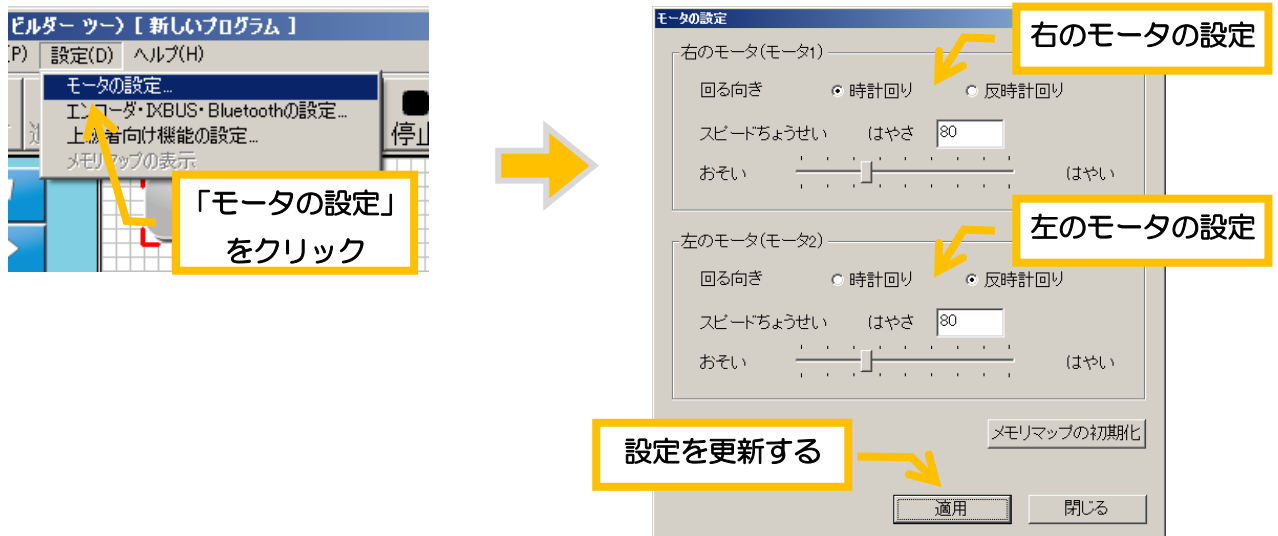
また、ファイルに保存したプログラムを読み込む場合は、メニューより「ファイル」→「開く」をクリックするか、ツールバーの「開くアイコン」をクリックしてください。

クリックすると、読み込むファイルを選択するダイアログを表示するので、読み込むファイルをクリックして「開く」をクリックしてください。

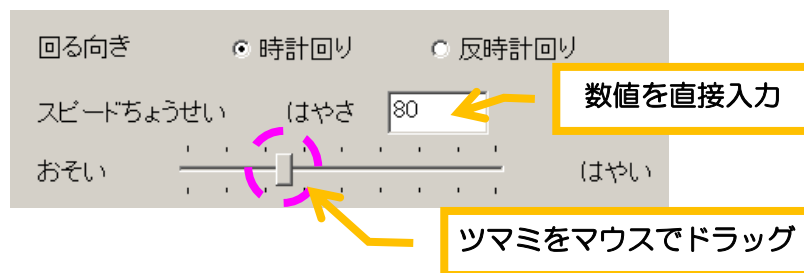


モータのスピード調整

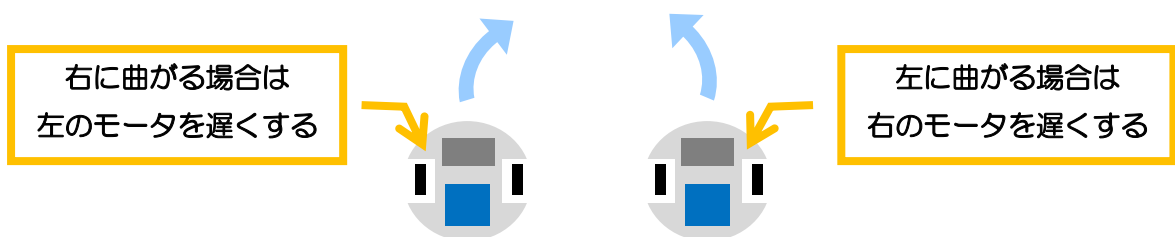
課題のプログラムを実行したときにロボットが多少曲がって走る場合は、真っ直ぐ走るように左右のモータスピードを調整します。モータスピードの調整を行う場合は、ロボットを PC と接続してメニューの「設定」→「モータの設定」をクリックしてください。



モータのスピードは 0~255 の範囲で設定します。数値が大きいほど、モータのスピードが速くなります。設定の変更は、数値を直接キーボードで入力するか、スライダーのツマミをマウスでドラッグしてください。



調整方法は、ロボットの曲がる方向の反対側のモータを遅くするように変更してください。数値は一度に大きく変更せず、3~5 程度の範囲で少しずつ変更して合わせてください。



設定を変更したら「適用」をクリックしてロボットに設定を書き込んでください。続いて、真っ直ぐ走るようになったかプログラムを実行して確認してください。

ロボットが曲がる原因は、モータの個体差・モータシャフトのタイヤや地面との摩擦など様々です。調整を行っても、プログラムを実行する場所や時間が変わると真っ直ぐ走らない場合があるので、プログラムを実行する場所に合わせて適時調整を行ってください。

3. いろいろな機能を使ってみよう

ビュートビルダー2では、モータ、LED、ブザー、センサなどいろいろな機能が搭載されています。これらの問題を解きながら一つずつ使ってみましょう

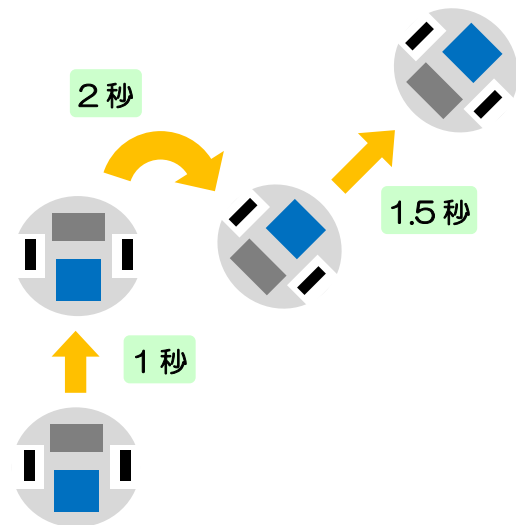
3-1 モータブロックの使い方

まず、モータブロックの機能を使ってみましょう。

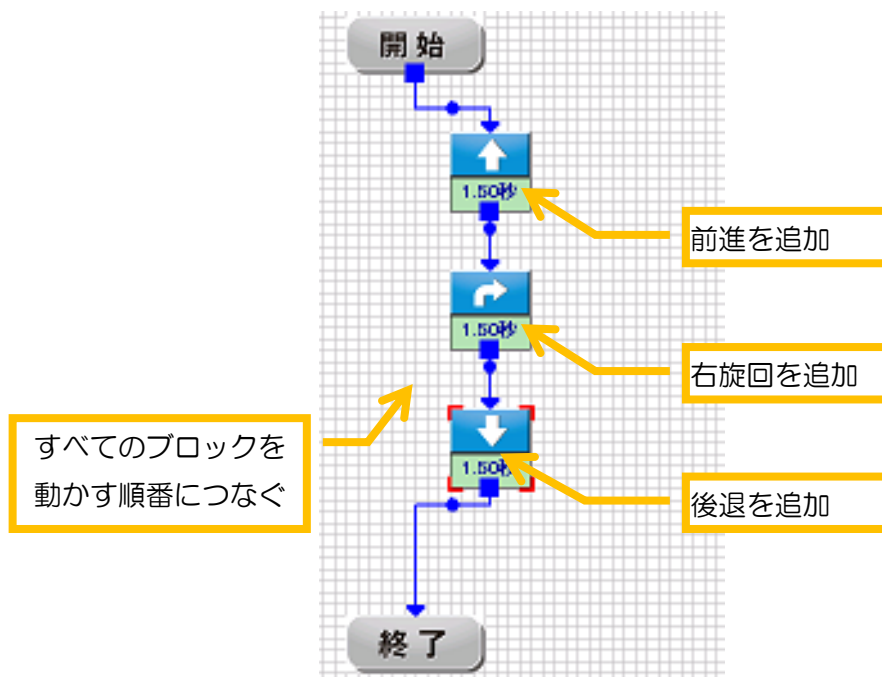
問題2

- 1秒前進
- 2秒右旋回
- 1.5秒後退

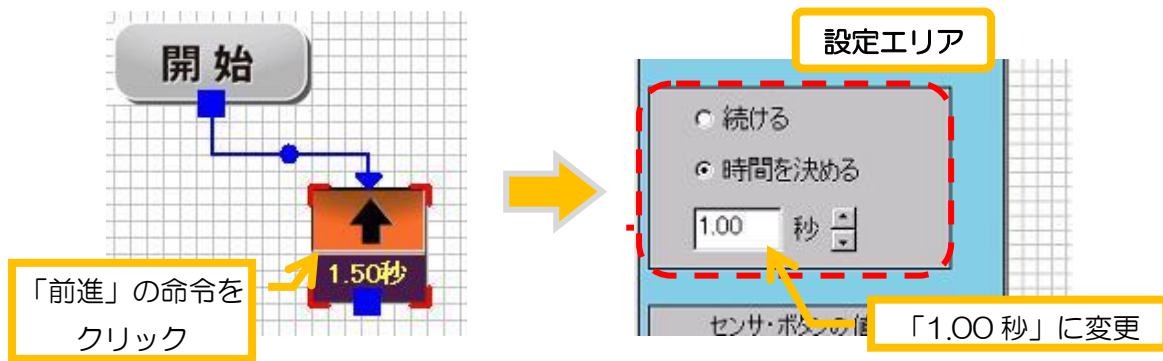
の順番で、ロボットが移動するプログラムを作成しましょう。



まず、問題1で説明した方法と同じように、ブロックを置き、ブロック同士を接続します。

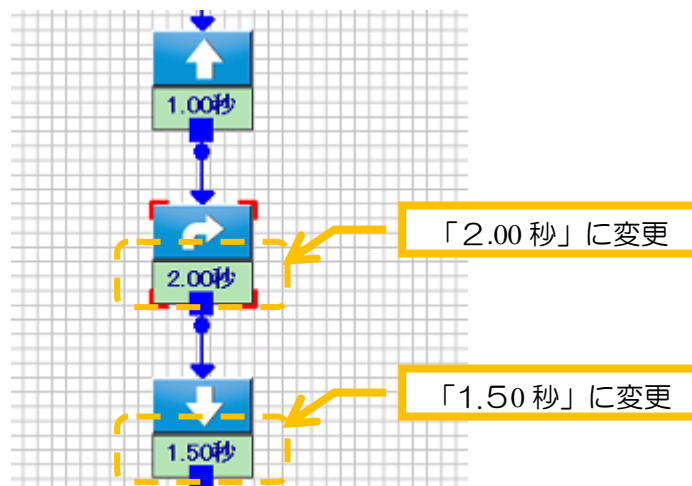


続いて、モータブロックの詳細を設定します。 それぞれのブロックには、実行する時間の設定ができます。



車輪制御のブロックは、モータを動かす時間を設定します。また、「続ける」「時間を決める」の設定は少し説明が難しいので、ここでは変更しないでください。

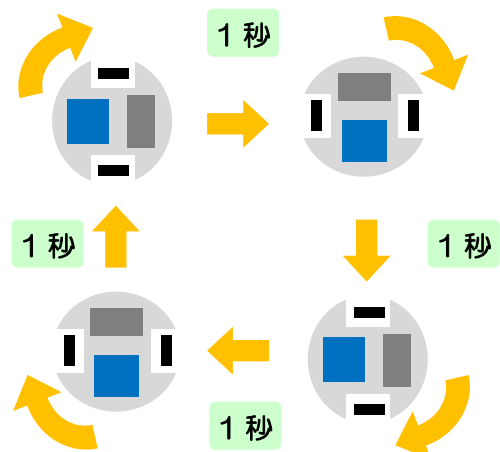
同様に、「右旋回」「後退」のブロックも、例題の通りに秒数を変更してください。



プログラミングできたら、ロボットに書き込んで確認してみましょう。
うまく動作したら、以下の練習問題にもトライしましょう。

練習①

1秒前進 → 右旋回 → 1秒前進 を繰り返しながら、ロボットが正方形を移動するプログラムを作成しましょう。

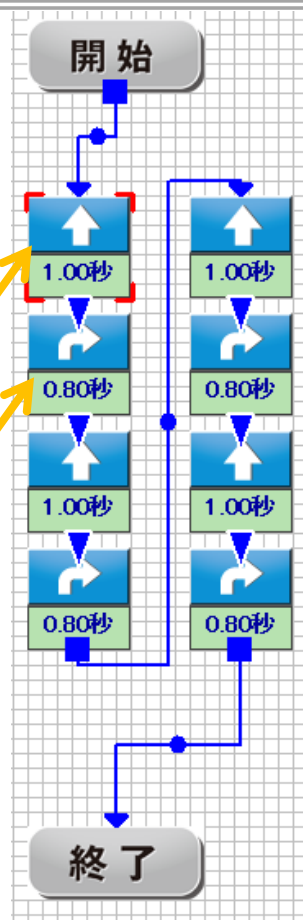


練習①の答え

前進と、右旋回を交互に
4 組つなぎます。

1.0 秒に設定

実際に 90 度回る時間を調べて、
時間を決めてください。

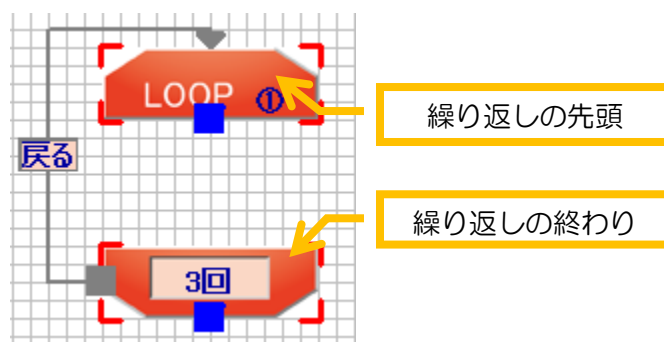


練習①は繰り返しブロックを使うと、もっと簡単なプログラムにすることができます。

繰り返しの使い方

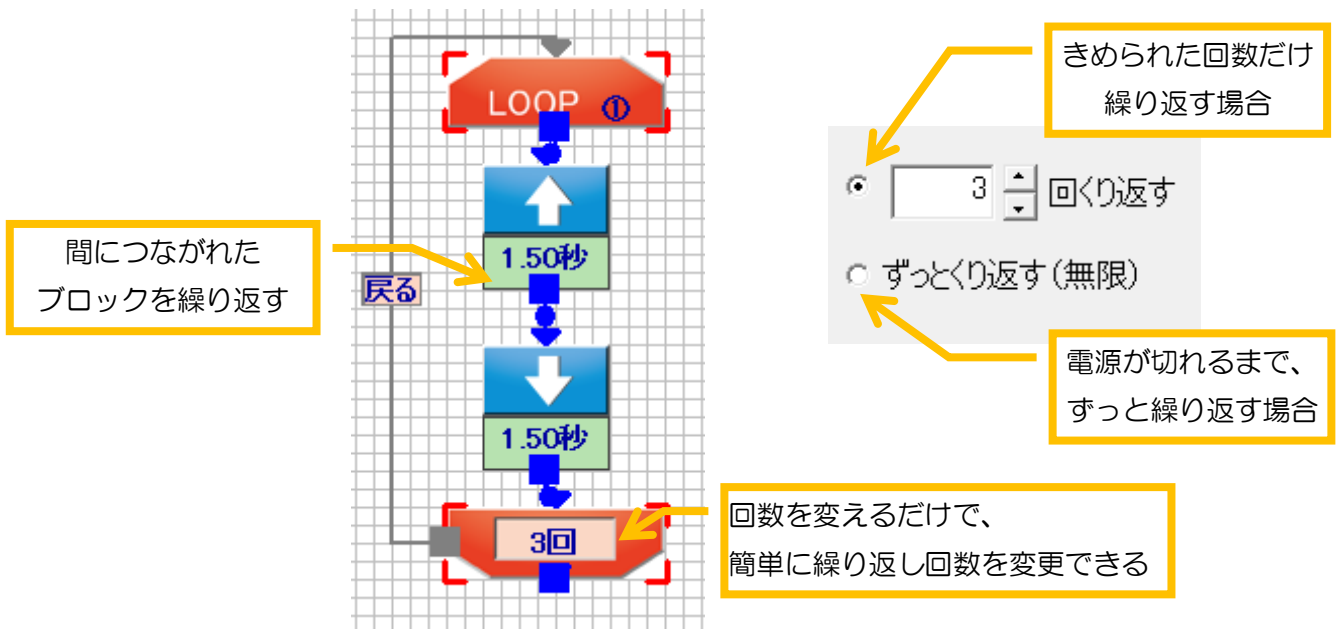
ある回数だけ同じ動作を繰り返す場合、繰り返しブロックを使用します。

繰り返しブロック



繰り返しブロックは、先頭と終わりの2つがセットになっています。この2つの間に繰り返したい動作をいれます。また、繰り返しは回数以外にも、電源が OFF になるまで、ずっと繰り返すように設定することもできます。

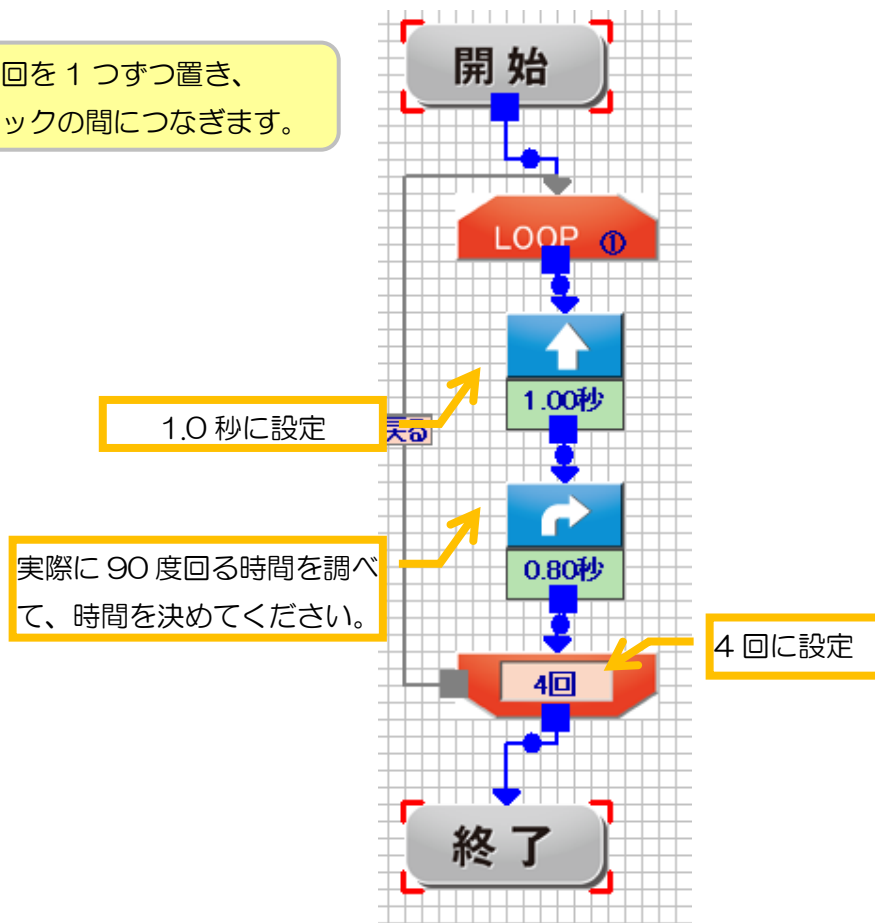
繰り返しブロックを使うことで、ブロックの数を少なくすることができます。100 回四角形を描くプログラムを作ると、400 個ものブロックが必要になりますが、繰り返しブロックを使うと、10 回でも 50 回でも同じ 2 個のアクションブロックと 1 個の繰り返しブロックだけでプログラムを作ることができます。



それでは、練習①のプログラムを、繰り返しブロックを使って作ってみましょう。

練習①の別解

前進と右旋回を1つずつ置き、
ループブロックの間につなぎます。



3-2 ブザーとLEDの使い方

次に、ロボットのブザーとLEDを使ってみましょう。

問題3

0.5秒間隔で「ドレミファソラシド」を鳴らし、最後にLEDをオレンジ・緑の順番で1秒ずつ光らせましょう

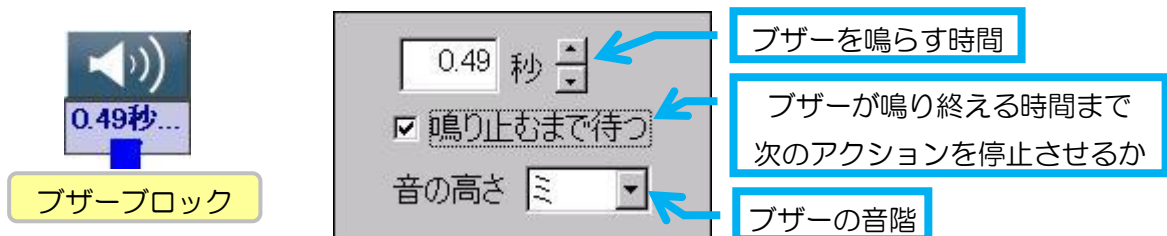


※ビュートミニ ARM(スタンダード)は、ブザーを鳴らす場合別途ブザーオプションの拡張が必要です。

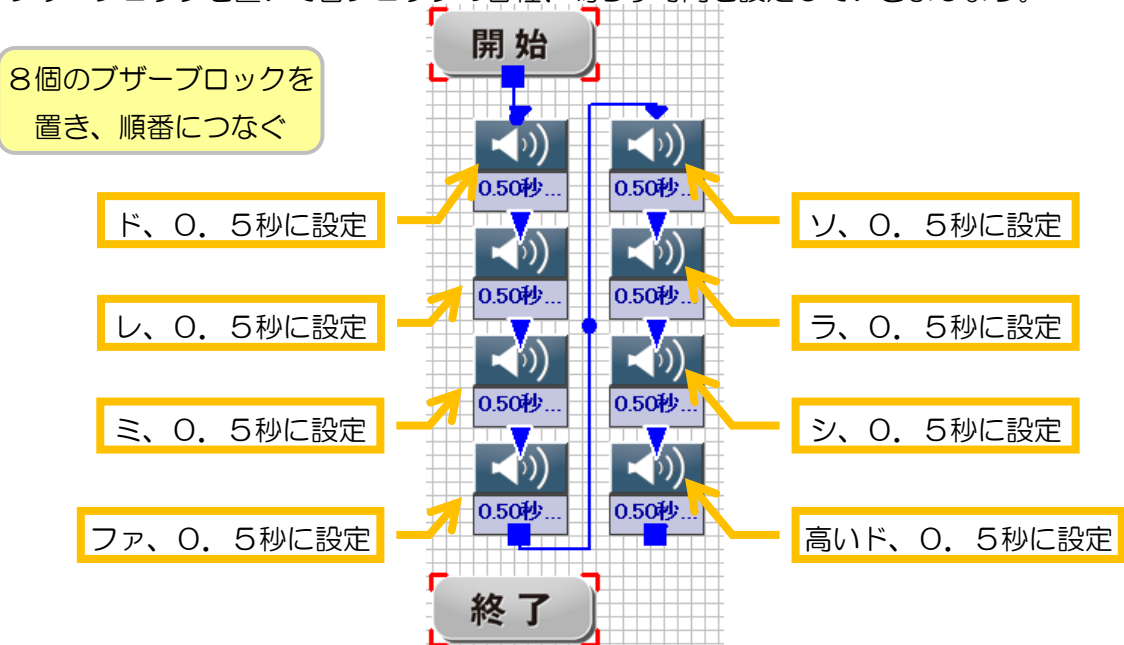
ブザーの使い方

まず、ブザーブロックを追加します。

ブザーは、「指定した音階と時間でブザーを鳴らす」アクションです。音階と時間はブロックの設定より変更できます。また、「鳴り止むまで待つ」のチェックを外すと、ブザーが鳴り終わる前に次のアクションに進むようになり、「ブザーを鳴らしつつモータを動かす」などの複雑な動作も可能になります。



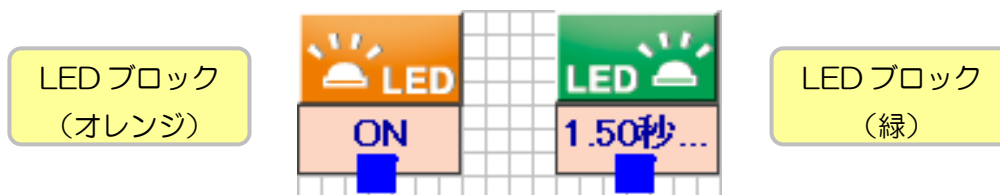
それでは、ブザーブロックを置いて各ブロックの音階、鳴らす時間を設定していきましょう。



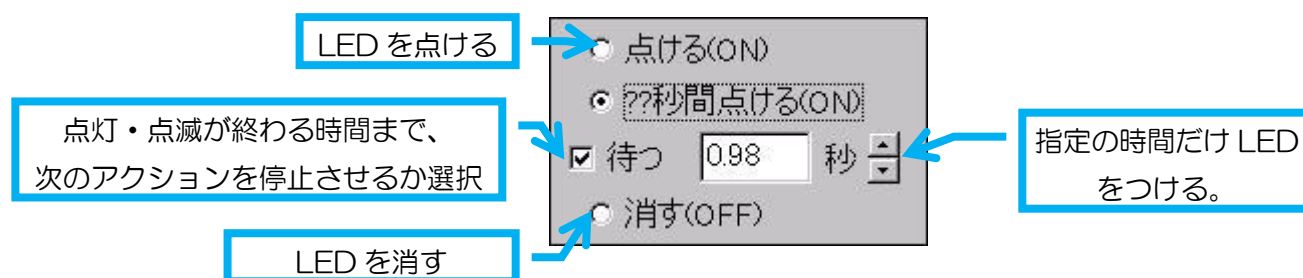
LED の使い方

次に LED ブロックを追加します。

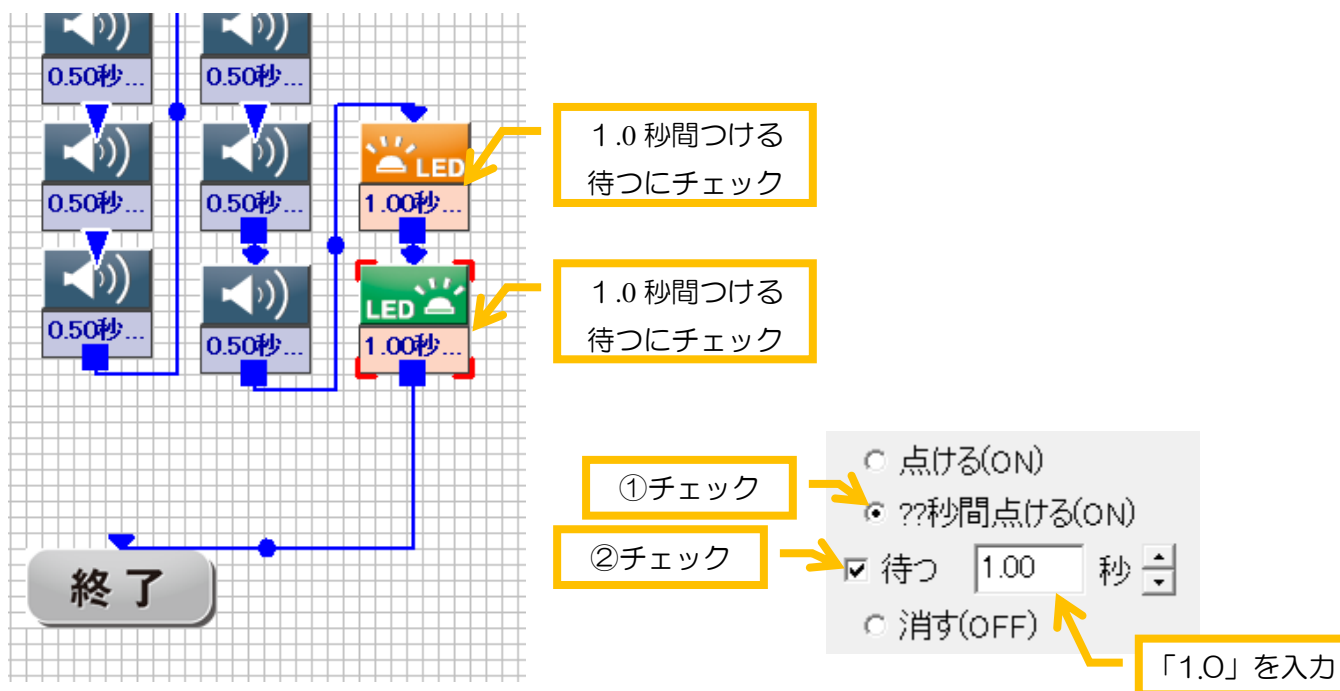
LED ブロックは、「LED を点ける」「LED を消す」「指定の時間だけ LED を点ける」の三通りの動作から選択できます。また、LED の色はブロックによって決まっています。



ブロックの設定にある「待つ」のチェックは、「指定の時間だけ LED を点ける」の際に、LED が消えるまで次のアクションを待つかどうかを表します。



それでは、LED ブロックを置き、ブザーブロックの後に接続してそれぞれのブロックを設定しましょう。



プログラミングできたら、ロボットに書き込んで確認してみましょう。
うまく動作したら、以下の練習問題にもトライしましょう。

この課題ではさっきよりブロックの数が多くなるので、ブロックをドラッグしてなるべくきれいに整列させましょう。見た目にきれいなプログラムを作ると、プログラムに問題があった時に、その問題点を見つけやすくなったり、修正しやすくなります。

練習②

最初に緑のLEDを点けて、ブザーで「ドレミファソラシド」すべての音が鳴り終わるまで、LEDをつけたままに見ましょう



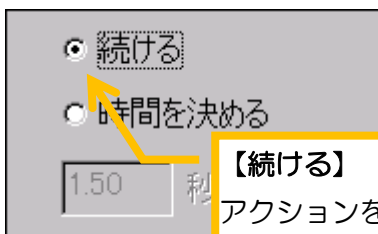
ヒント

この問題は、LEDをつけながらほかの動作をさせる必要があります。設定の次のアクションを待つか、待たないかの設定をおこなう必要があります。

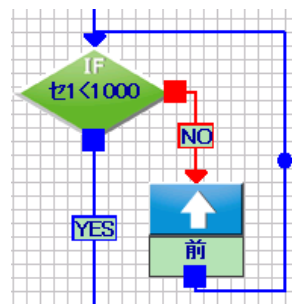
「続ける」の使い方

「続ける」は、アクションブロックの動作を行ったら、待つことなくそのまま次のアクションに進みます。

こちらの設定は、「そのアクションを動作させながら、ほかのアクションを実行する（例えば車輪を動かしながらブザーを鳴らすなど）」という場合に使います。



【続ける】
アクションを動作させながら、次のアクションに進む

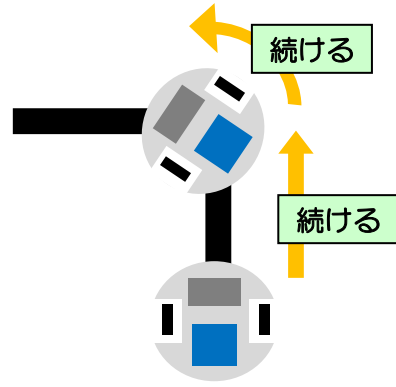
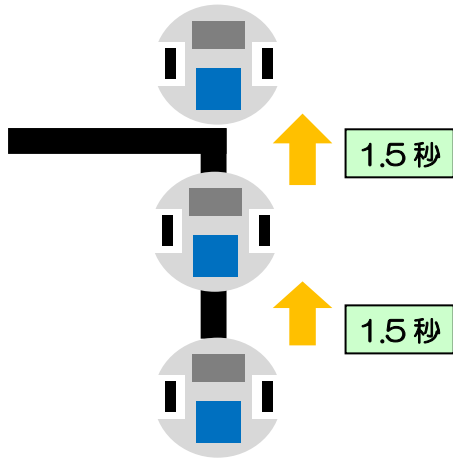


「続ける」に設定することで、センサを取りこぼさずにロボットを動かせる

「時間を決める」で秒数を指定する場合、ロボットは車輪を動かしたら、指定した時間過ぎるのをただ待つだけです。その待ち時間中にセンサの数値が変化しても、それに応じてロボットの動作は変わらないため、ロボットがセンサの反応を取りこぼしてしまいます。

「続ける」に設定を変更すると、ロボットは動かした車輪はその状態のまま、次のアクションを実行していくので、動きながら、センサをチェックすることができます。

「時間を決める」と「続ける」の設定は、プログラムの目的に応じて使い分けが必要です。「決められた距離・時間動作させる」場合、「続ける」は「センサや変数など他の要素と絡めて動作が柔軟に変化する」場合に、それぞれ利用します。



「時間を決める」

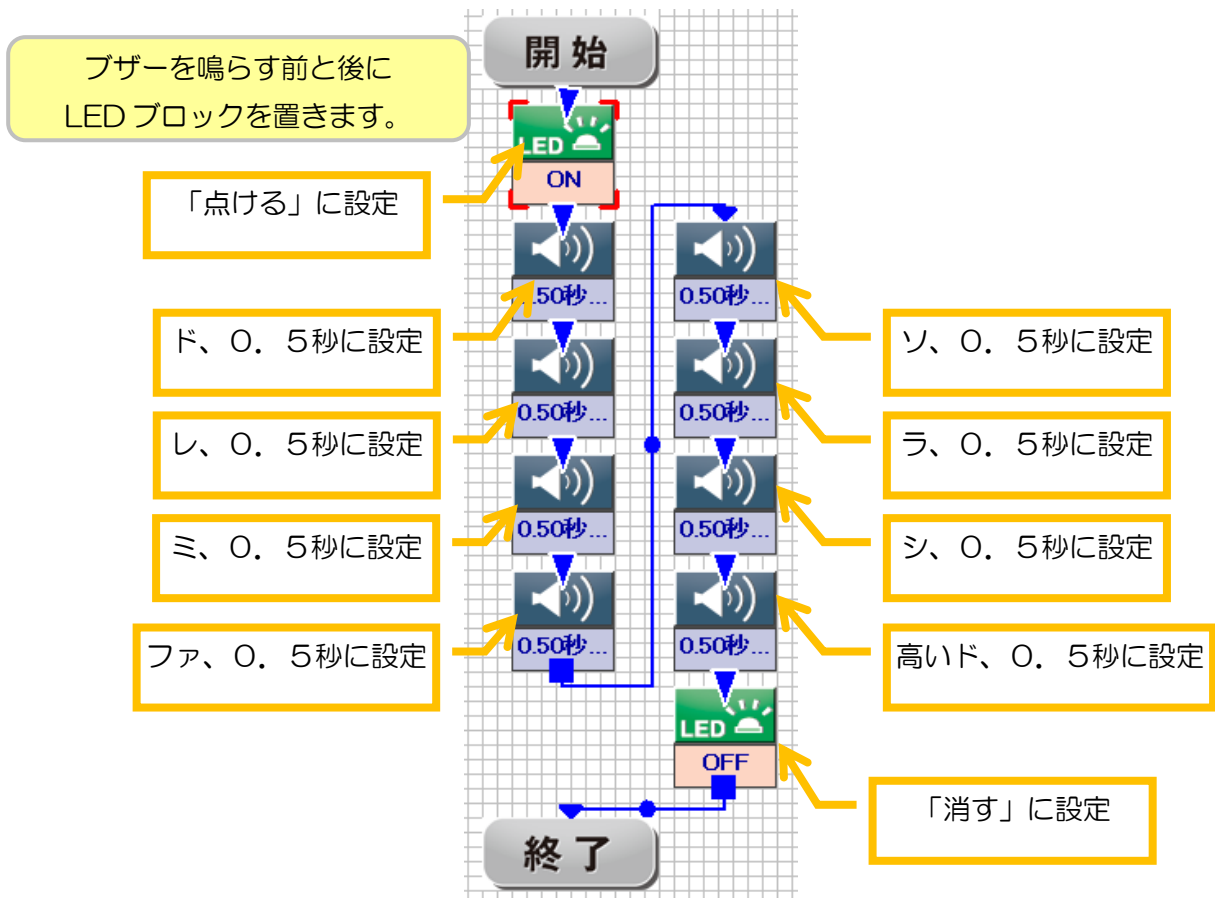
決めた時間ごとに1回しかセンサを見ないため、ラインを見落とす。
決まった距離の走行は適している

「続ける」

車輪を動かしつつ他の命令を実行でき、センサの反応を取りこぼさない。ただし、「時間を決める」に代わる方法で動作時間を決める必要がある

それでは、「続ける」を使って、練習②のプログラムを作ってみましょう。

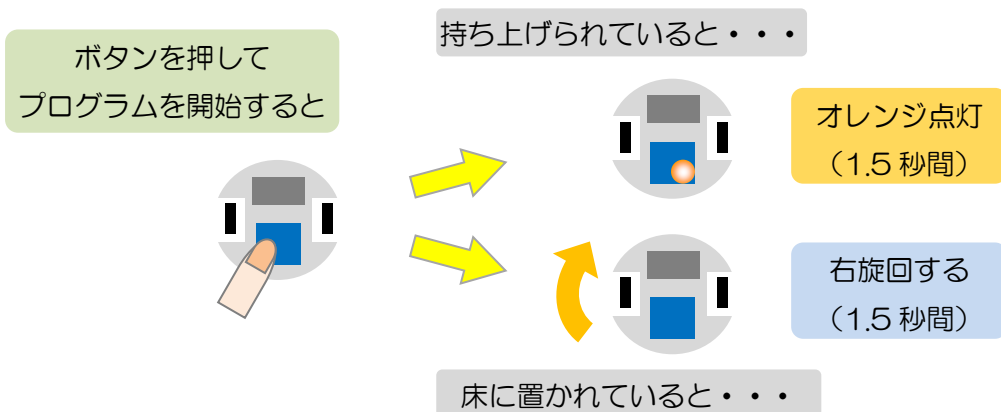
練習②の答え



3-3 センサを使った条件分岐

問題4

ボタンを押したときに、床に置いているときは回転し、持ち上げられていると車輪は回転せずに、オレンジのLEDが光るようにプログラムしてみましょう。

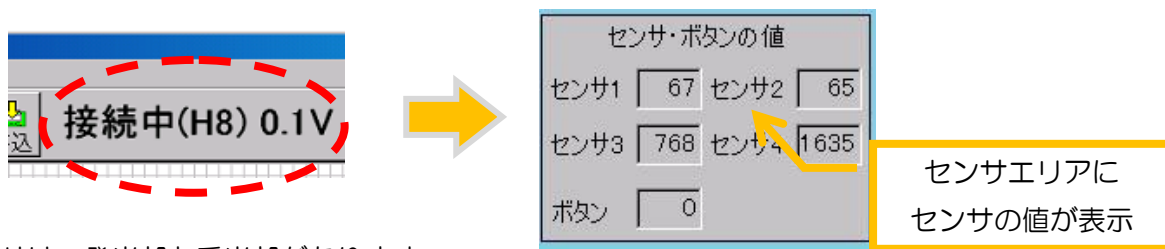


この問題では、ロボットに付いている赤外線センサを使います。

センサ値の確認

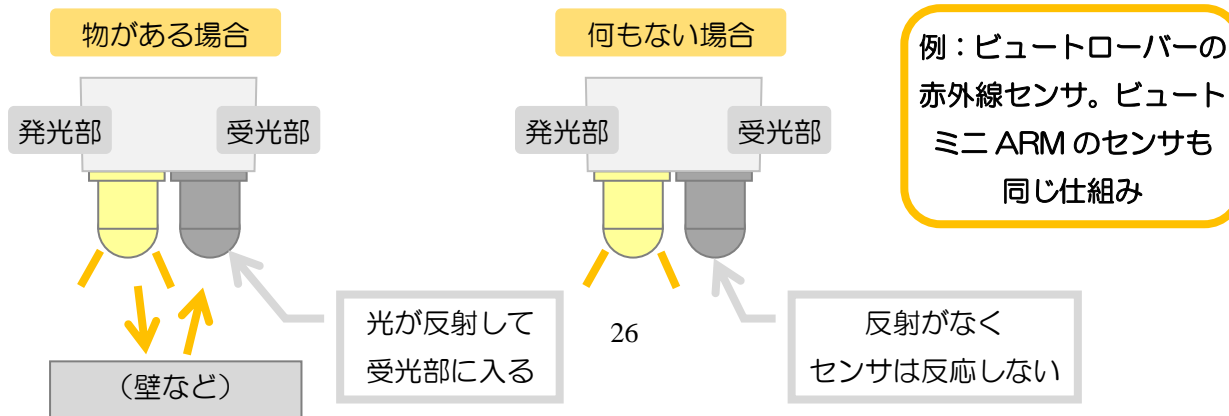
まず赤外線センサの反応を確かめるため、ロボットとPCをつないで、画面左下のセンサエリアの数値を確認してみましょう。

USBをつないで、接続中になると・・・

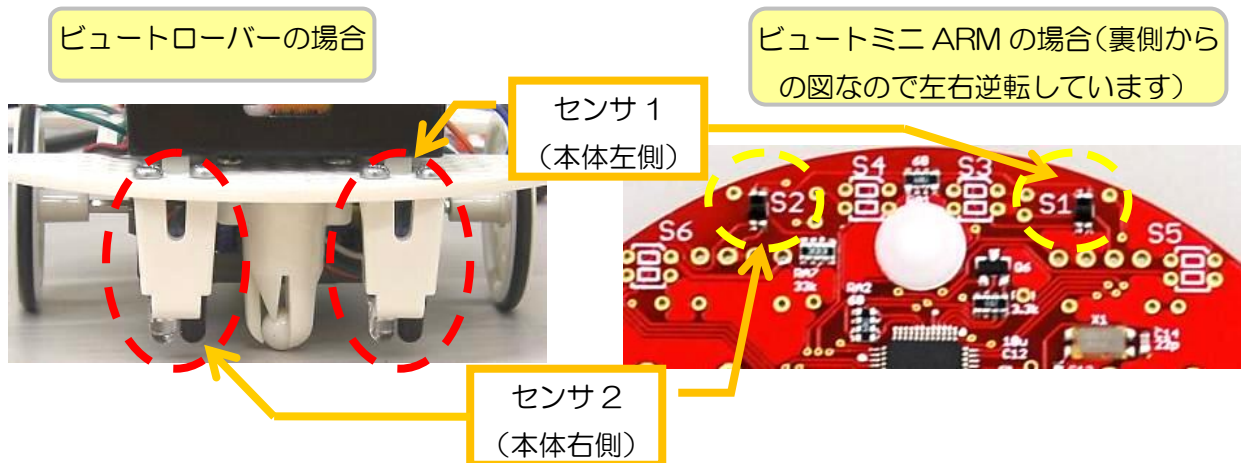


赤外線センサは、発光部と受光部があります。

発光部から赤外線という目に見えない光が出て、物に反射して帰ってきた明るさを、受光部で検出することができるので、「センサが何かの物で塞がれているか」「センサの目の前の色が明るいかどうか」「周囲が明るいかどうか」などを知ることが出来ます。これにより「ライトレース」「障害物回避」などのプログラミングが出来ます。



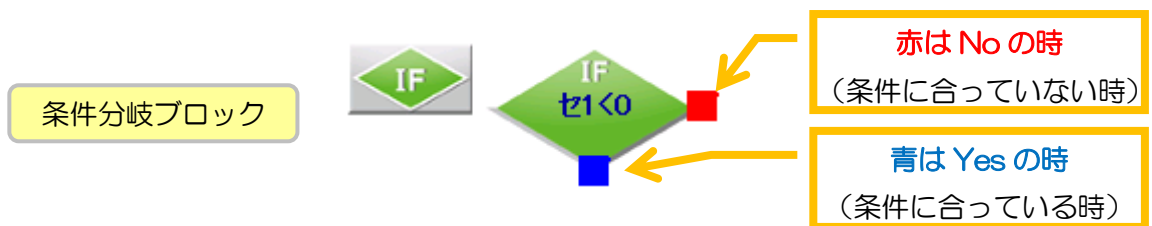
それでは、ロボットを床に置いたときと、持ち上げた時で、センサの値がどのように数値が変化するかを確認して、表を埋めてみましょう



	床に置いた時	持ち上げた時	センサに手を当てた時	センサから手を離れた時
センサ 1				
センサ 2				

条件分岐の使い方

「ロボットが床に置かれているかどうか」を判断するためには、条件分岐ブロックを使います。



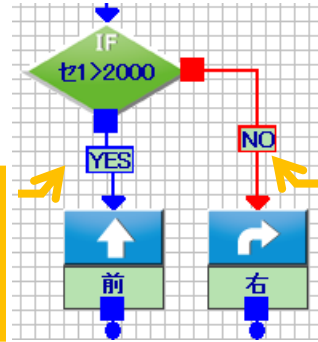
条件分岐ブロックは緑のひし形形で表示され、青と赤の二つの矢印を持っています。ブロックの設定では「センサ 1 の数値が 2000 以上か?」などの条件を決めることができます。ロボットがブロックを実行したときに、決められた条件が成立していれば青い矢印 (YES)、成立していなければ赤い矢印 (NO) に従ってプログラムを進めます。

「センサ 1 の数値が 2000 より大きいか？」
という条件を設定した場合

センサ1 の値は
2000
より大きい?

分岐ブロックの設定エリア

センサ 1 の数値が
2000 より大きければ
YES に進む

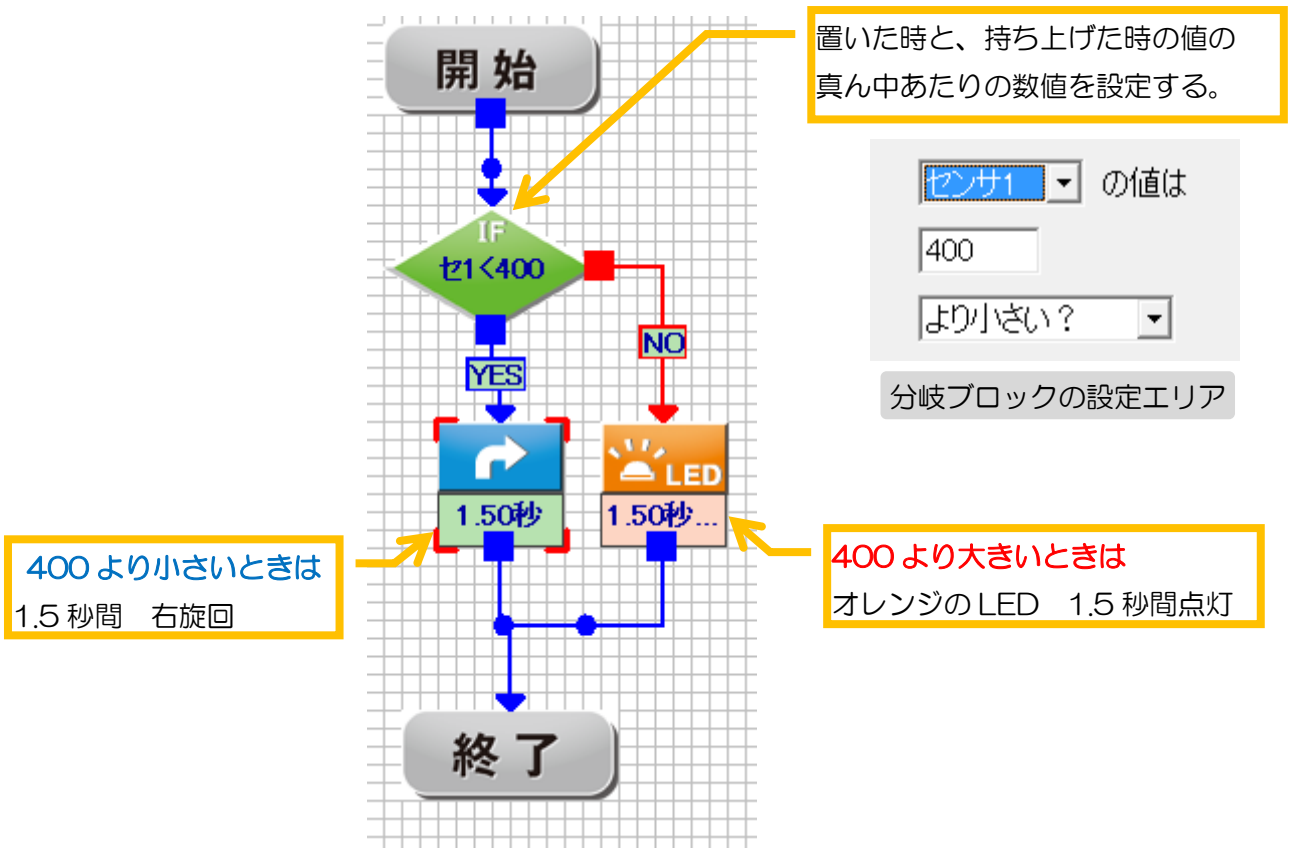


センサ 1 の値が
2000 より小さければ
NO に進む

ブロックの設定では、「どのセンサを使うか」「センサと比べる値はいくつか」「どのような条件で比べるか」を選びます。

それでは、プログラムを作ってみましょう。

条件分岐ブロックと、右回転、LED ブロックをおいて、それぞれ以下のように設定してください。

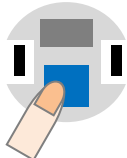


プログラミングできたら、ロボットに書き込んで確認してみましょう。
うまく動作したら、以下の練習問題にもトライしましょう。

練習③

問題 4 と同様のプログラムで、一度ボタンを押したら、机の上では右旋回、持ち上げたらオレンジの LED を光らせる、をずっと繰り返すプログラムを作りましょう。

ボタンを押して
プログラムを開始すると



持ち上げると・・・



オレンジ点灯

床に置くと・・・



右旋回する

ヒント

繰り返すにはループブロックを使います。また、センサの取りこぼしがないように、右旋回は、「続ける」をうまく使いましょう。

練習③の答え

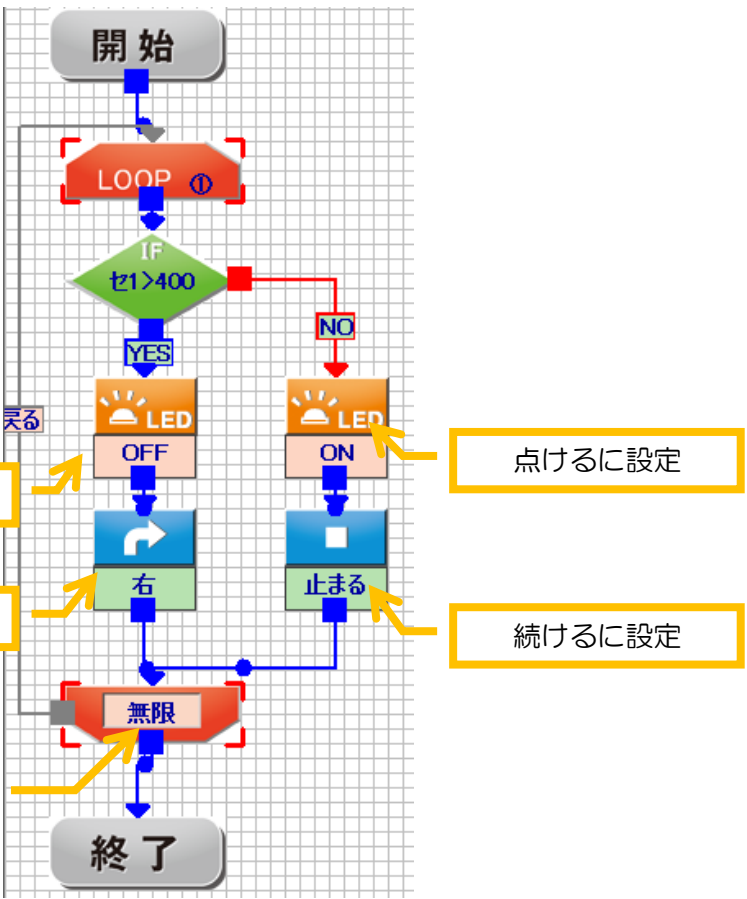
ループブロックの間に、条件分岐ブロックとLED、車輪制御ブロックをつなぎます。

「続ける」を使うと、動作はずっと継続するので、「LEDをOFFにする」、「車輪を停止させる」ブロックが必要になります。

「消す」に設定

続けるに設定

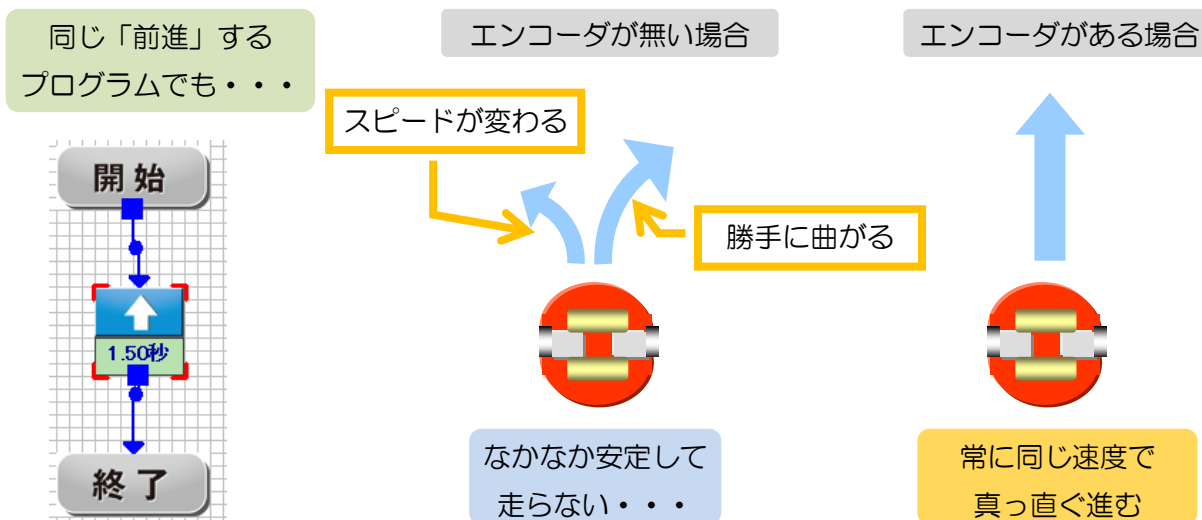
「ずっと繰り返す」に設定



3-4 エンコーダを使った移動

※この項目はビュートミニ ARM 向けです

ビュートミニ ARM のエンコーダ・ブザー搭載版をご利用の場合、本ソフトウェアからエンコーダの設定を有効にすることで、前進命令のときに勝手に曲ってしまう問題を改善したり、電池の残量に関係なく移動速度をほぼ一定に保ったりすることが出来ます。

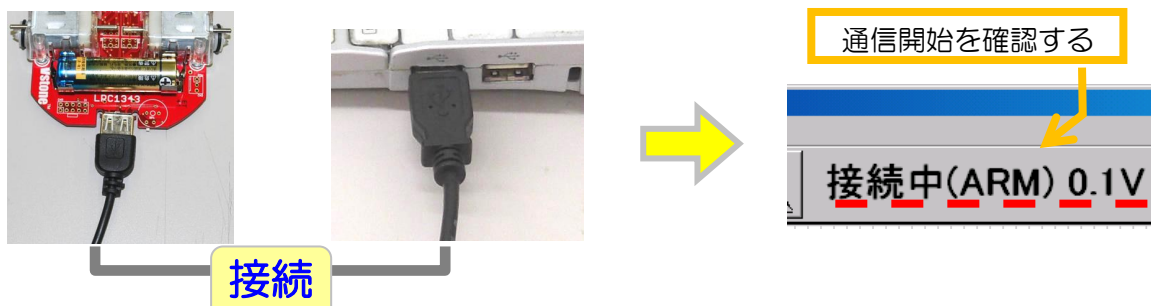


※ エンコーダを有効にした場合でも、電池が少なすぎたり、タイヤが地面やモータシャフトに空回りなどすると、若干の誤差が発生する場合があります。

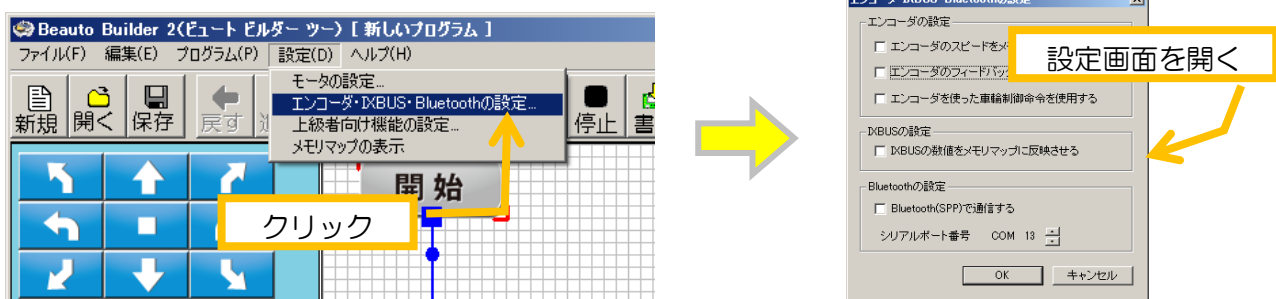
エンコーダを有効にする

エンコーダをビュートビルダー2で使用する場合、以下の手順で設定を変更してください。

1. ロボットをPCと接続してください



2. メニューの「設定」→「IXBUS・エンコーダ・Bluetooth の設定...」をクリックし、設定画面を開いてください。



3. 設定画面の「エンコーダを使った車輪制御命令を使用する」にチェックを入れてください。
 (チェックすると、「エンコーダの設定」の項目全てに自動的にチェックが付きます)。
 続いて「OK」をクリックして設定をロボットに反映させてください。

チェックする

自動でチェックが付く

OK をクリック

このメッセージが
出れば設定完了

以上の設定で、メモリマップにエンコーダの位置・速度情報が反映され、ロボットが移動する際にエンコーダの情報を基に進行方向や速度を補正します。また、距離や角度を指定した車輪制御命令を利用できます。

INBUS0	0	SWORD	
エンコーダA速度	0	SWORD	R/W
エンコーダB速度	0	SWORD	R/W
エンコーダA位置	128	SWORD	R/W
エンコーダB位置	60	SWORD	R/W
ユーザ変数(SWORD)a	0	SWORD	R/W

メモリマップに速度・位置が反映される

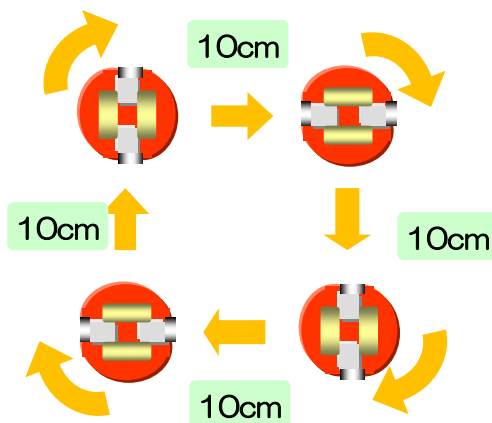
距離・角度を指定して移動する車輪制御命令

エンコーダを使った車輪制御ブロック

メモリマップについては、リファレンスの「上級者向け機能の設定」をご参照ください。

問題6

エンコーダを使って、一辺の長さが10cmの正方形を描いて走るプログラムを作成しましょう。

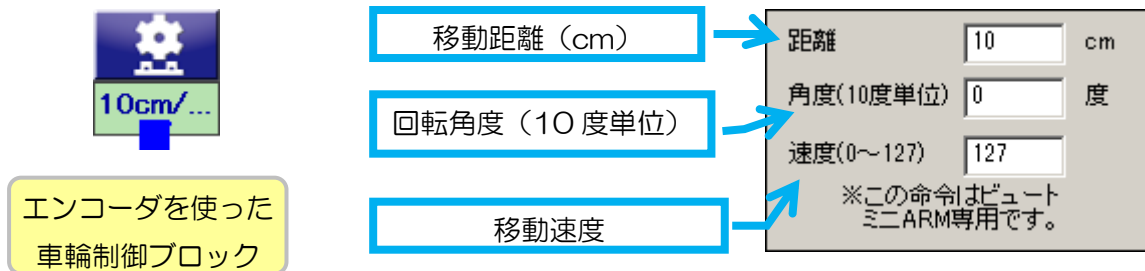


ヒント

エンコーダをつかった車輪制御命令で、正しい角度と距離を走らせましょう。また、ループブロックも使いましょう。

エンコーダを使った車輪制御の使い方

エンコーダを使った車輪制御ブロックは、距離・角度・速度を決めてロボットを移動させることができます。ブロックの設定にある「距離」「角度」「速度」のそれぞれの項目で、ロボットの移動方法を設定します。

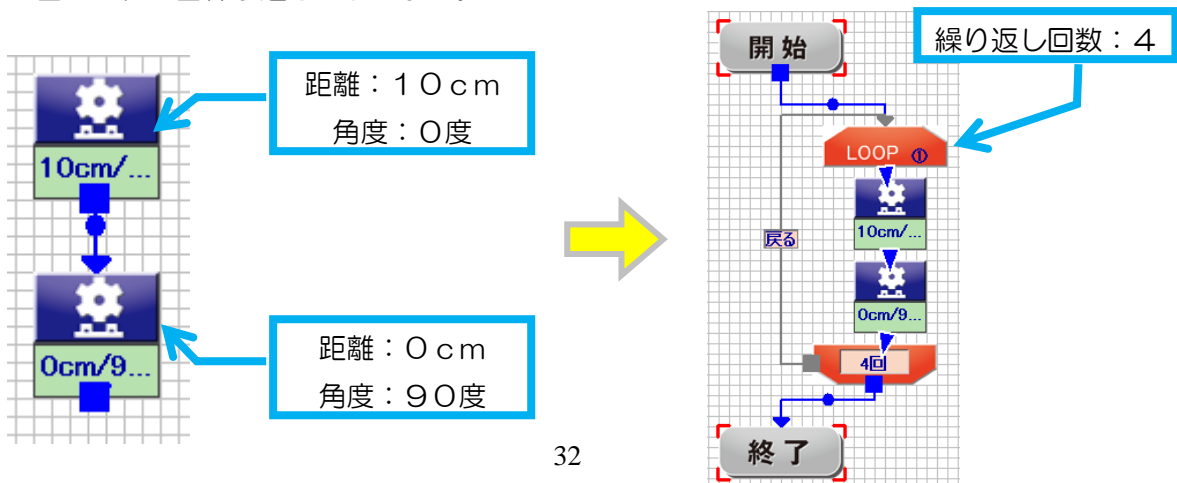


ロボットの移動距離は「距離」の項目で入力します。マイナスの数値を入力すると、ロボットがバックします。ロボットの回転角度は「角度」の項目で入力します。こちらも、マイナスの数値を入力するとロボットが逆方向に回転します。

距離と角度は同時に設定することができます。両方に数値を入力した場合、ロボットが移動しながら回転します。



10cmの正方形の形で走るには、「10cm走る」と「90度回る」という二つの命令を4回繰り返せばできます。まずは、それぞれに対応したエンコーダを使った車輪制御ブロックを追加しましょう。次に、それをループブロックで囲んで、4回繰り返させましょう。



4. 次の課題にチャレンジ！

ビュートビルダー2の基本的な使い方はマスターしましたか？

ロボットにはセンサが2個ついていますので、単体でも様々なプログラミングができます。以下のような課題にもどんどんチャレンジしてみましょう。

紹介しているいくつかの課題は、以下の動画でも紹介していますので、参考にしてください。

<http://www.vstone.co.jp/robot/rover/img/Robver07.wmv> (8.4Mbyte)

また、ロボットや本ソフトウェアにはチュートリアルで説明していない様々な機能が備わっています。これらに関する情報を最後に紹介します。

4-1 高度な機能を使う

本説明書の「リファレンス」には、チュートリアルで取り上げていないプログラムの命令や、ロボットのモータ・拡張機器の設定などについて説明しています。チュートリアルを読んで一通り本ソフトウェアの使い方を理解できたら、リファレンスの以下のページを読んで、より高度な機能を使ってみましょう。

様々な命令を使う

リファレンスの「2-2.アクションブロックの説明」をご参照ください。

モータの速度や回転方向を変える

リファレンスの「3-1.車輪の速度・回転方向の設定」をご参照ください。

Bluetooth や I2C などの拡張機器を使用する

リファレンスの「3-2.エンコーダ・I2C・Bluetooth の設定」をご参照ください。

ファームウェアの書き換えを行う

リファレンスの「3-4.ファームウェアの書き換え」をご参照ください。

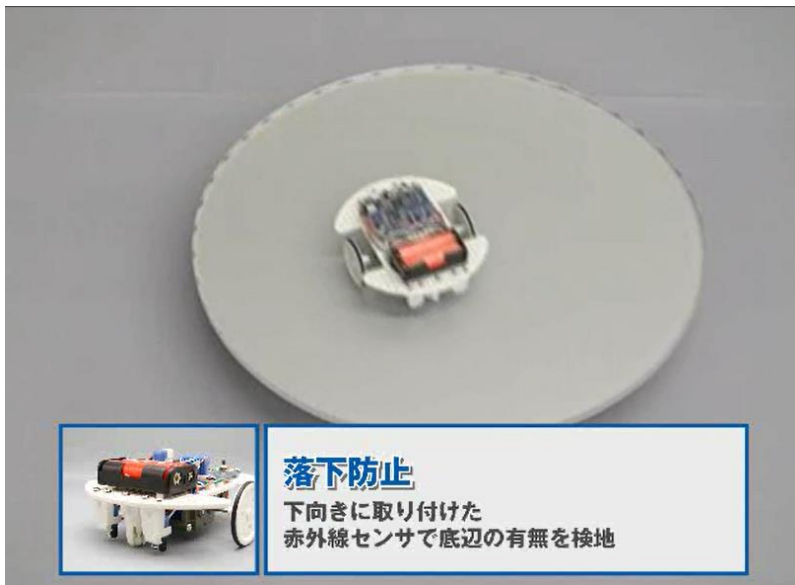
C 言語でプログラミングする

リファレンスではなく、「VS-WRC104_取扱説明書」の「6. 開発環境の導入とサンプルプログラムの実行」をご参照ください。

4-2 拡張機能を使う

ロボット用のオプションとして、様々な拡張部品が販売されています。これらを追加して機能を拡張すると、さらに色々な課題にトライすることができます。なお、ロボットの機種ごとに使用できる拡張部品が異なるのでご注意ください。

落下防止



落下防止

下向きに取り付けた赤外線センサで底辺の有無を検地

下向きにつけた赤外線センサで、床があるかないかを検出して、台から落下しないようにするプログラムです。

テーブルなどから落ちないようにプログラムできますが、まずは低めの台の上から落ちないプログラムを作ってみましょう。

障害物回避



障害物回避

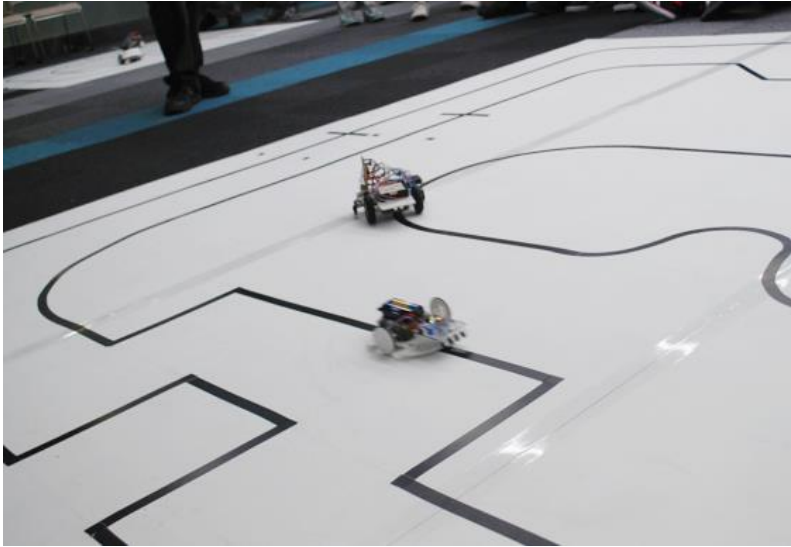
前向きに取り付けた赤外線センサで障害物を検地

赤外線センサを前向きに取り付け、障害物が近づいてきたら避けるプログラムを作ってみましょう。

速いと、障害物を検出するのが難しいので、まずゆっくりと近づくようにプログラムしてみましょう。

ビュートミニ ARM の場合、ビュートローバー用の赤外線センサとビュートミニ ARM 用ユニバーサルボードを使って、水平にセンサを取り付けます。

ラインレース

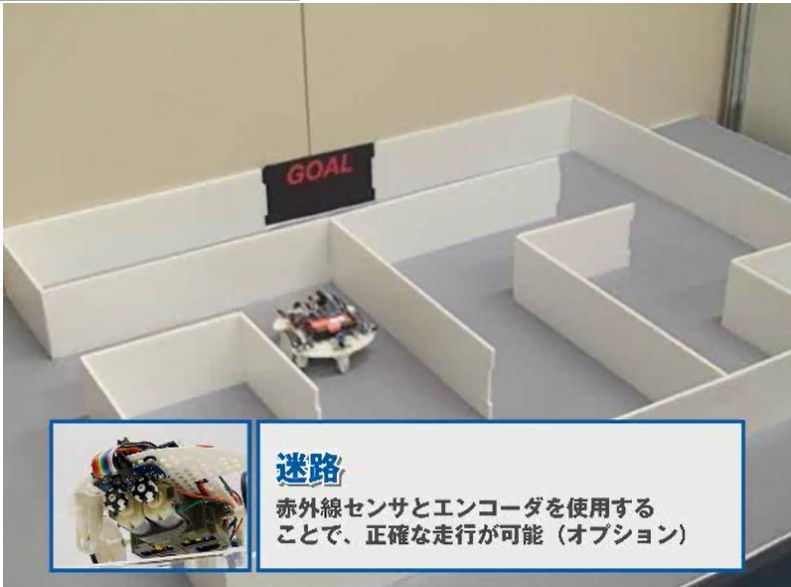


白い床にひかれた、黒いラインに沿ってロボットを走らせるプログラムです。

下向きにつけた2つの赤外線センサでラインを検出し、ラインから外れないように走らせます。

ビュートロボコンなどの競技会でもタイムを競いあう、奥の深いプログラムです。

迷路



ロボットにロータリーエンコーダと赤外線センサ（水平に取り付け）を追加することで、迷路もクリアできるようになります。

ロボットにエンコーダを搭載すると、毎回同じ角度だけまわることができるようになりますので、迷路をクリアしやすくなります。



迷路

赤外線センサとエンコーダを使用することで、正確な走行が可能（オプション）

無線操縦

※ビュートローバー・ビュートローバーARMのみ

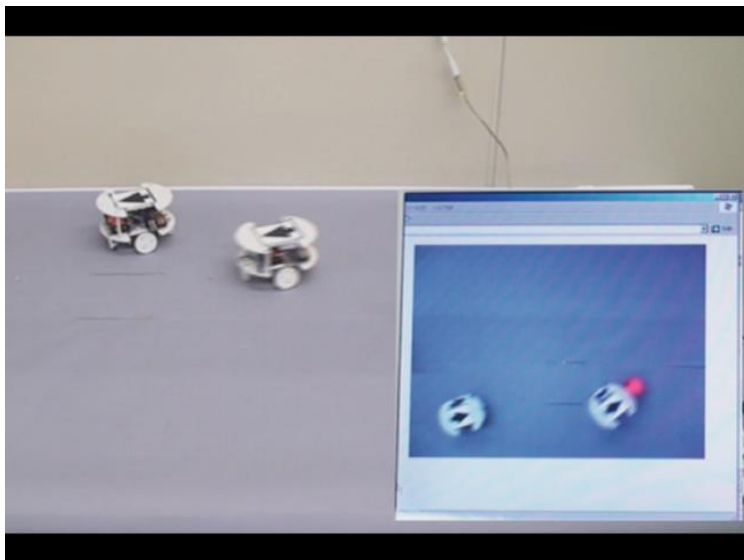


無線操縦セットを拡張することで、無線コントローラで遠隔操縦できるようになります。

各種ロボコンなどで、有線だとケーブルが邪魔でうまく動作できないことがあります。

ビュートローバーなら、ビュートビルダー2で無線コントローラをつかって決まった動作を出すことができます。

群制御



Bluetooth を拡張したロボットを複数用意すると、PC などから複数のロボットを同時に無線で制御できます。

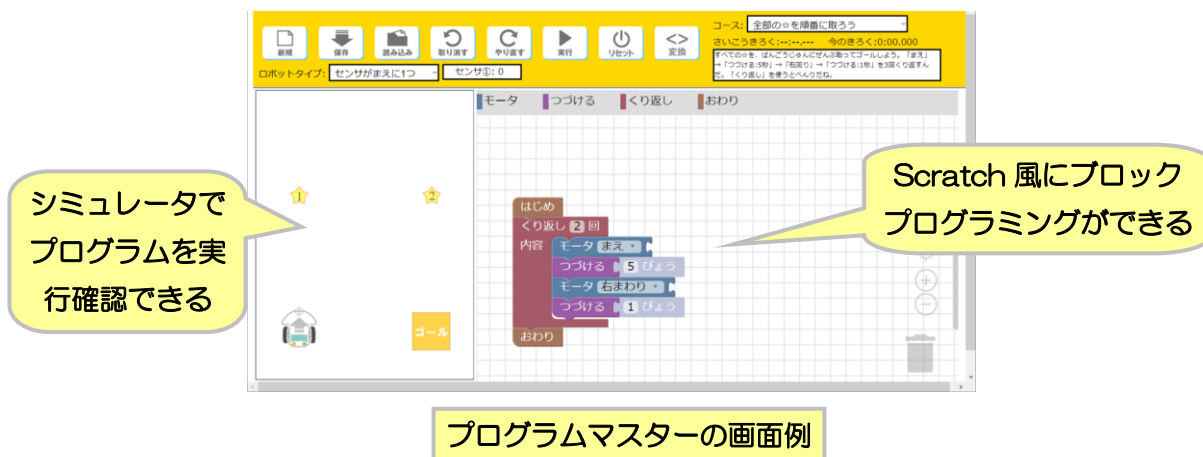
本体に色マーカをつけて、天井カメラから撮影してサッカーをさせたり、エンコーダを搭載して先頭のロボットの動作を追従させたりすることが可能です。

5. プログラムランドとの連携について

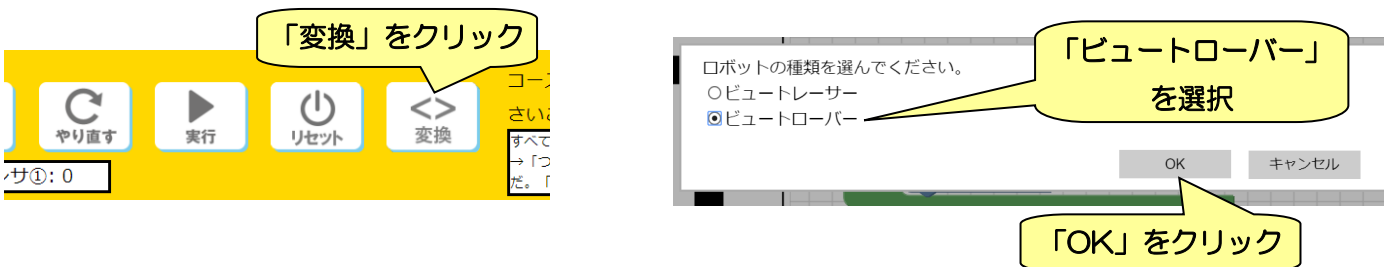
ヴイストーン株式会社の web ページで公開している無料プログラミングツール「プログラムランド」では、本ソフトウェア用のプログラムを出力することができます。

プログラムランドは、Web ブラウザで動作する Scratch スタイルのブロックプログラミングツールです。ブラウザベースのため、PC、スマートフォン、タブレットなど多様なプラットフォームで利用でき、また、PC やスマートフォンなどへのソフトのインストールも不要です。

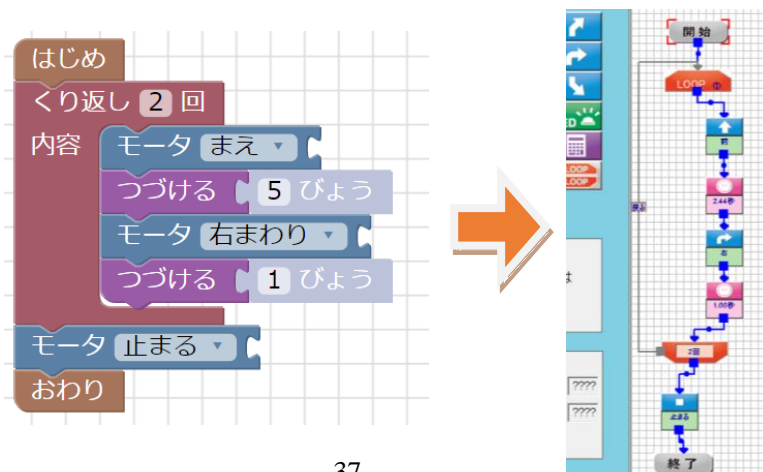
利用 URL:<https://www.vstone.co.jp/programland/>



プログラムランドの画面上にある「変換」ボタンをクリックすると、本ソフトウェア用のプログラムを出力できます。ボタンをクリックし、続いてロボットの種類の選択で「ビュートレーサー」を選んで「OK」をクリックすると、変換したプログラムをダウンロードできます。



ダウンロードしたプログラムは本ソフトウェアから開くことができ、そのままロボット本体に書き込んで実行出来ます。



プログラムランドと本ソフトウェアでは、いくつかの仕様の違いにより、変換の際にいくつかの変更が加えられる場合があります。具体的には以下ようになります。

- センサ値の範囲がプログラムランドは0～255、本ソフトウェアは0～4096と違いがあるので、変換したプログラムはブロック内のセンサ値が範囲に比例して大きくなります。

また、センサの反応はロボット本体と異なるため、センサを使う分岐ブロックが入ったプログラムは、ロボット本体に書き込む前に本体のセンサ値を確認して調整する必要があります。

ビュートビルダー2 取扱説明書

リファレンス

目次

1. はじめに	41
2. プログラムの作成	41
2-1. 基本操作方法の説明	41
画面表示説明	41
アクションブロックの操作説明	43
アクションブロックの追加	43
アクションブロックの詳細表示と設定変更	43
アクションブロックの実行順序の設定（矢印の接続）	44
アクションブロックのコピー・切り取り・貼り付け	45
アクションブロックの選択	45
2-2. 基本のアクションブロック	47
移動のアクションブロック	47
ウェイトのアクションブロック	48
ブザーのアクションブロック	49
LED のアクションブロック	50
分岐のアクションブロック	50
ランダム of アクションブロック	52
繰り返しのアクションブロック	53
2-3. 上級者向けのアクションブロック	55
車輪制御のアクションブロック	55
変数演算のアクションブロック	56
サブルーチンのアクションブロック	57
ゲームパッド分岐のアクションブロック	57
分岐のアクションブロック（上級者向け）	58
エンコーダを使った車輪制御のアクションブロック	59
2-4. プログラムの実行・保存	60
サンプルプログラムの読み込み	60
プログラムのテスト実行	60
プログラムの書き込み・実行	61
プログラムの保存・読み込み	62
サブルーチンの保存・読み込み	63
2-5. ツールバー・メニューの項目と機能	64
ファイル	64
編集	64
プログラム	64

設定.....	65
ヘルプ.....	65
ショートカットキーの活用.....	66
3. ロボットの詳細設定.....	67
3-1. 車輪の速度・回転方向の設定.....	67
3-2. エンコーダ・IXBUS・Bluetooth の設定.....	68
エンコーダの設定.....	69
IXBUS の設定.....	69
3-3. Bluetooth による通信の設定.....	70
PC との接続・シリアルポート番号の確認.....	70
通信方法の変更・シリアルポート番号の設定.....	71
VS-C2 for Android について.....	71
3-4. ファームウェアの書き換え.....	72
ビュートローバーへのファームウェア書き込み.....	72
ビュートローバーARM・ビュートミニ ARM へのファームウェア書き込み.....	73
3-5. 上級者向け機能の設定.....	74
3-6. メモリマップの表示.....	76
4. FAQ.....	78

1. はじめに

リファレンスでは、本ソフトウェアに関する説明を、機能別に記述しています。チュートリアルを読んで本ソフトウェアの操作方法やプログラミングについて一通り理解された後で、本ソフトウェアの様々な機能を活用したい・ロボットに関する設定変更や拡張を行いたいなどの場合に、該当する項目をお調べいただけるようになっています。

なお、以下については、本ソフトウェアに関する基礎的な説明としてチュートリアルに掲載しております。まだチュートリアルを読んでいない方は、必ず一度ご確認ください。

- ・ ロボットの各部名称について・・・4～5ページ
- ・ 本ソフトウェアの動作環境・インストール方法について・・・6～9ページ

2. プログラムの作成

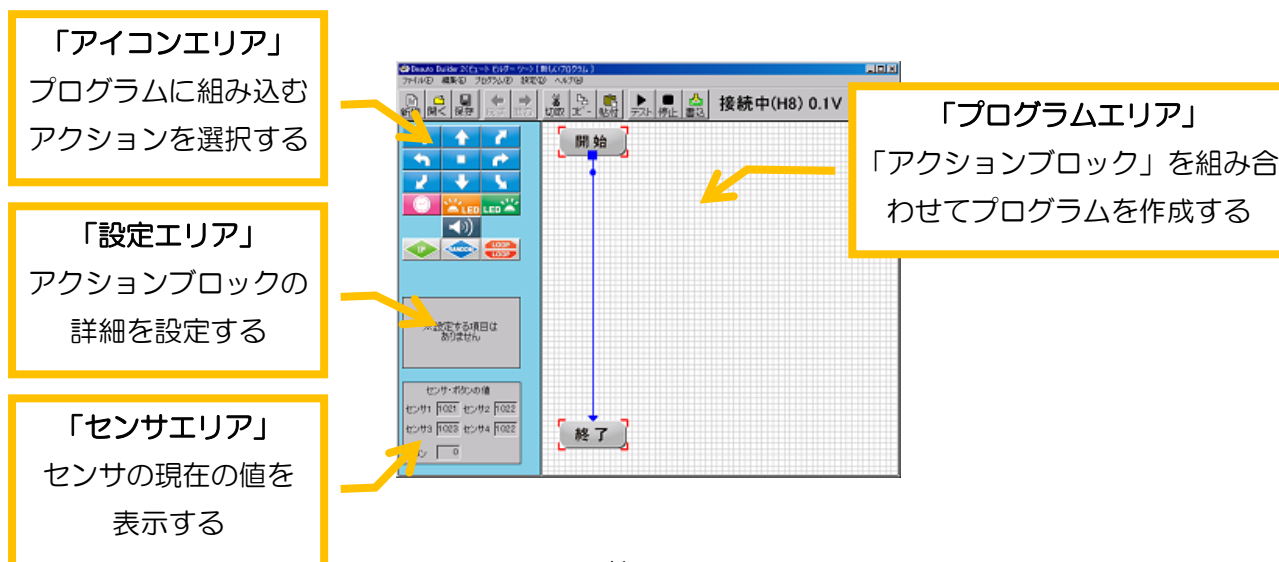
本項目では、本ソフトウェアよりロボット本体のプログラムを作成する方法について説明します。前半では、プログラムの各部分の名前や機能、また、プログラムの1命令を表す「アクションブロック」の追加・配置の方法や、複数のアクションブロックを矢印でつないでプログラムを実行する順序を設定する方法など、本ソフトウェアでプログラムを作成するための基本的な操作について説明します。

後半では、アクションブロックの概要や設定方法に関する細かい説明、作成したプログラムを実行する方法について説明します。また、メニューやツールバーなど、本ソフトウェアに備わっている細かい機能について説明します。

2-1. 基本操作方法の説明

画面表示説明

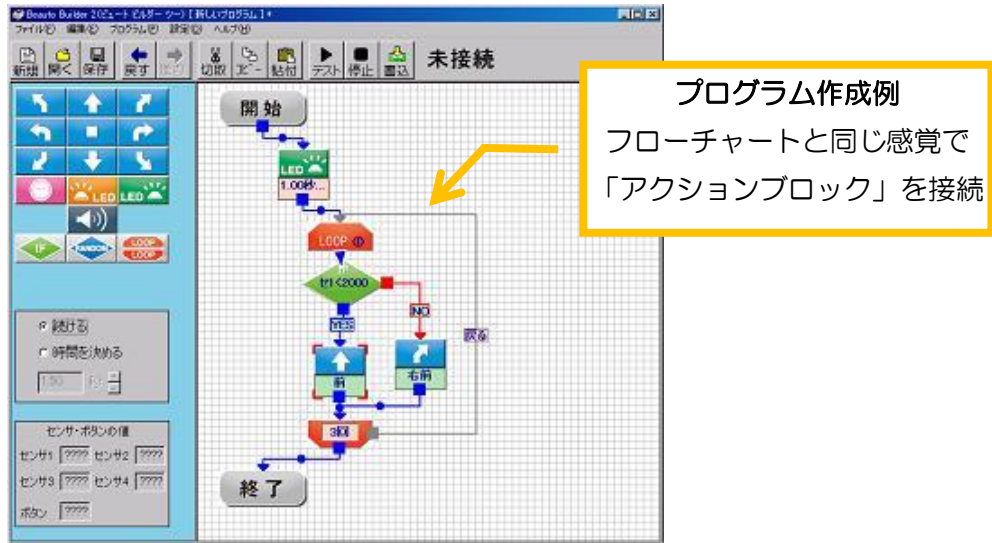
本ソフトウェアの画面構成について説明します。ロボットのプログラム作成は、ほぼ全て下図の範囲内で作成します。各部分の名称と概要は下記の通りです。



画面左側は、上から順番に、プログラムに使用する「アクション」（命令）を選択する「アイコンエリア」、アクションの詳細を設定する「設定エリア」、ロボットの現在のセンサ情報を表示する「センサエリア」の三つに分かれます。画面右側はアクションを組み合わせるプログラムを作成する「プログラムエリア」といいます。

画面右のプログラムエリアには、ロボットに与える命令を種類に応じて色分け・記号化した「アクションブロック」で表示されます。アクションブロックは青や赤の矢印でつながっており、この矢印がプログラムを実行する順番になります。プログラムは必ず「開始」のアクションブロックから始まり、「終了」のアクションブロックで終わります。

このように、本ソフトウェアではプログラムの目的に応じて、様々な種類のアクションブロックをフローチャートのように接続してプログラムを作成します。



本ソフトウェアで使用できる主なアクション（アクションブロック）の種類は下記の通りです。また、ソフトウェアの設定を変更すると、これ以外に上級者向けの複雑な命令も使用することが出来ます。

前進・後退・旋回・停止など、移動を行なう

待つ・ブザーを鳴らす・LEDを光らせる

繰り返し・条件分岐などプログラムの実行順序を指定する

プログラムを開始・終了する

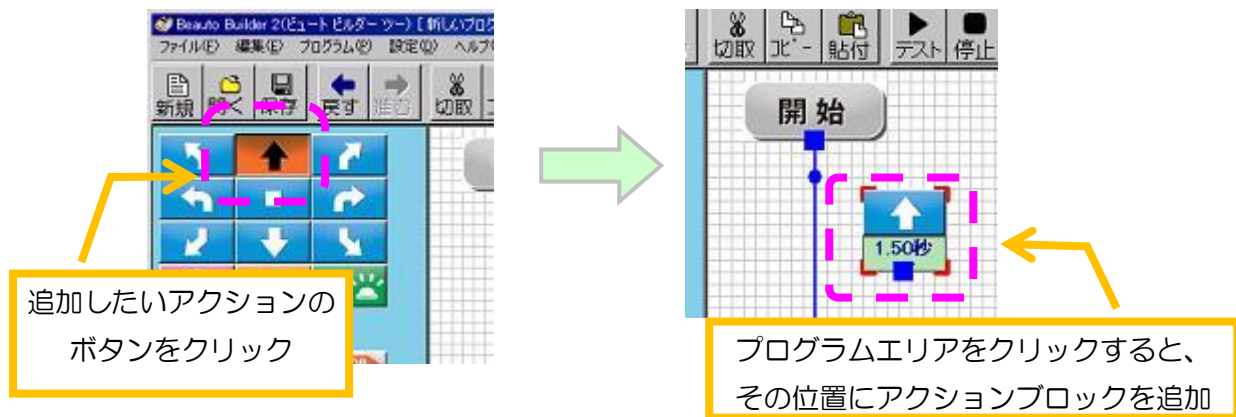
開始

終了

アクションブロックの操作説明

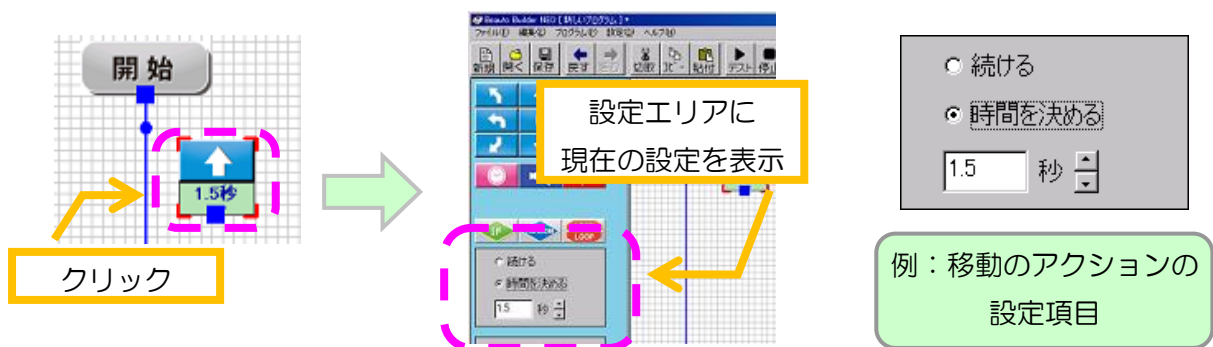
アクションブロックの追加

プログラムエリアにアクションブロックを追加する場合は、画面左側のアクションエリアより追加したいアクションのボタンをクリックします。続いて、画面右側のプログラムエリア中のアクションブロックを挿入したい場所をクリックすると、その場所にアクションブロックが追加されます。



アクションブロックの詳細表示と設定変更

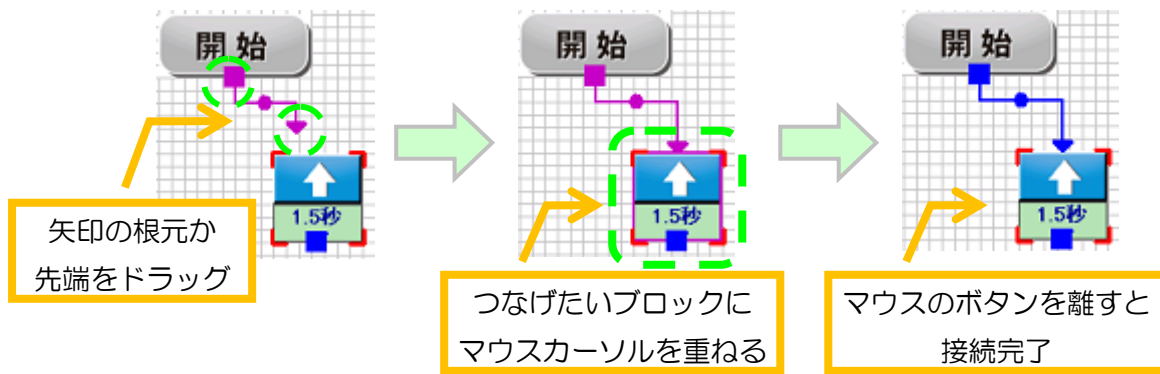
移動のアクションの「車輪を動かす時間」、ブザーのアクションの「音程と発音時間」など、アクションに関する細かい設定は、画面左側中央の「設定エリア」より行ないます。プログラムエリアのアクションブロックをクリックすると、周囲が赤い枠線で囲まれ、設定エリアにクリックしたアクションブロックの詳細と設定項目が表示されます。



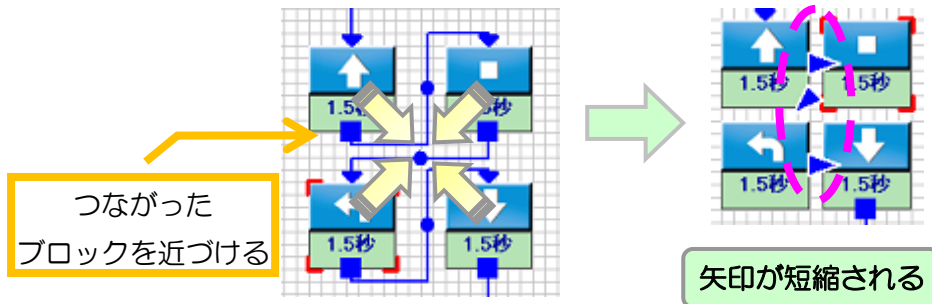
※各アクションブロックの詳細設定につきましては、後述の「2-2.アクションブロックの説明」をご参照ください。

アクションブロックの実行順序の設定（矢印の接続）

プログラムに追加したアクションブロックは、矢印を接続して実行する順序を設定します。下図のようにアクションブロックの小さい四角、もしくは矢印の先端をマウスでクリックしてドラッグすると、アクションブロックから矢印が表示されます。そのまま接続したいアクションブロックにマウスカーソルを重ねると、目的のアクションブロックが周囲に紫色の枠線で囲まれ矢印が自動的につながります。そのままマウスのボタンを離すと接続が完了します（「開始」など一部のアクションブロックは矢印を接続できません）。

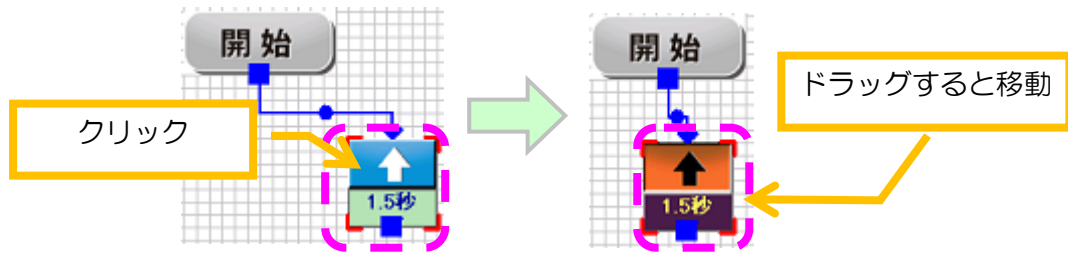


矢印をつないだアクションブロック同士が隣接している場合、下図のように矢印が短縮して表示されます。



アクションブロックの移動と削除

アクションブロックの場所を移動したい場合は、アクションブロックをマウスでクリックしてドラッグしてください。なお、アクションブロックは背景のマス目に沿って移動します。



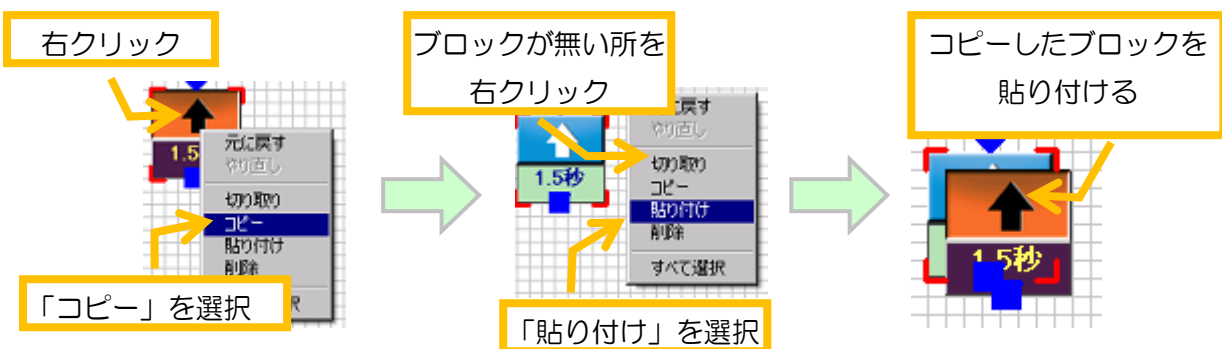
アクションブロックをプログラムエリアから削除する場合は、マウスカursorをアクションブロックに合わせて右クリックします。クリックするとメニューを表示するので、メニューより「削除」を選択してください。削除したいアクションブロックをクリックしてキーボードのDELETEキーを押しても、アクションブロックを削除することができます。



アクションブロックのコピー・切り取り・貼り付け

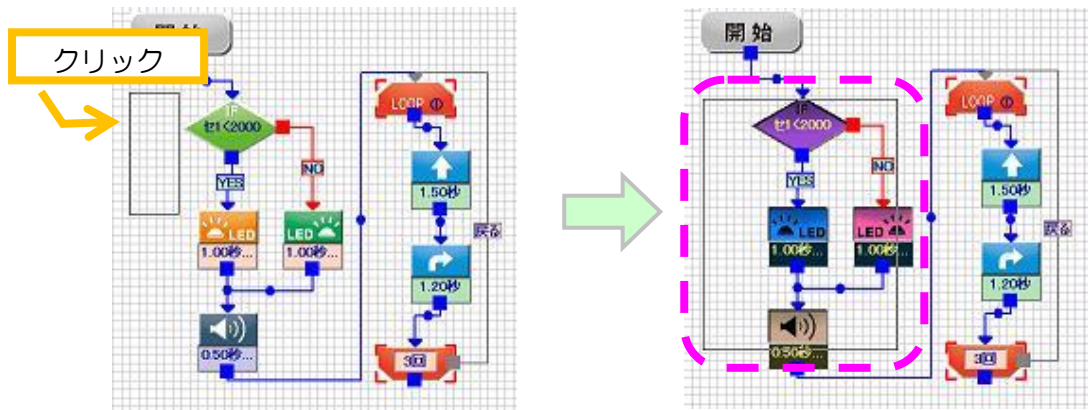
マウスカursorをアクションブロックに合わせて右クリックしてメニューを表示し、メニューより「コピー」を選択すると、右クリックしたアクションブロックをコピーすることができます。また、メニューより「切り取り」を選択すると、右クリックしたアクションブロックを削除してコピーすることができます。

アクションブロックをコピーしてから、プログラムエリアを右クリックして表示されるメニューより「貼り付け」を選択することで、コピーしたアクションブロックをプログラムエリアに貼り付けることができます（開始・終了のアクションブロックはコピーできません）。



アクションブロックの選択

プログラムエリアのアクションブロックが無い部分をクリックしてドラッグすると、下図のように枠線を表示します。この枠線をアクションブロックに重ねることで、アクションブロックをクリックした場合と同じように表示色が変わります。このようにアクションブロックの表示色が変わった状態を「選択状態」といいます。選択状態のアクションブロックは、先ほど説明した場所の移動や削除・コピー・切り取り・貼り付けなどをまとめて行うことができます。



ブロックが無い所をクリックしてドラッグすると、選択用の枠線を表示

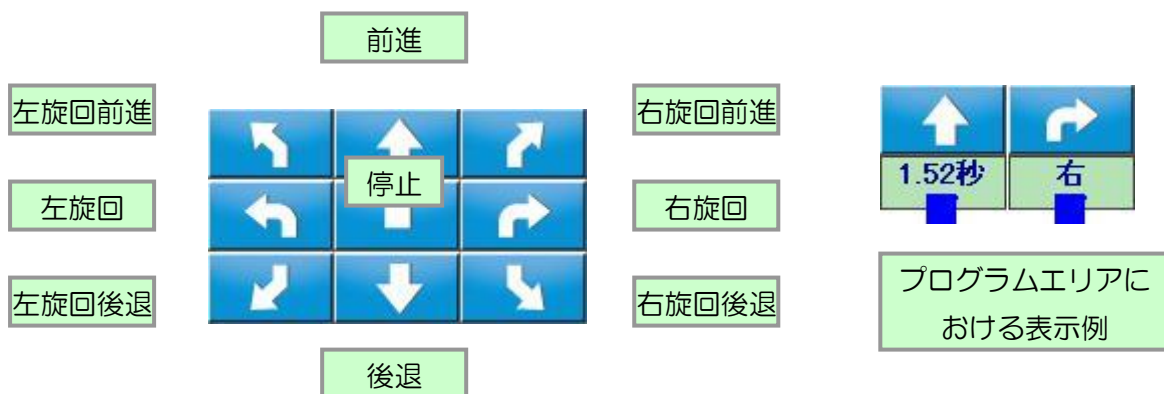
枠線とアクションブロックを重ねるとアクションブロックが「選択状態」になり、まとめて移動や削除・コピー・貼り付けなどができる

2-2. 基本のアクションブロック

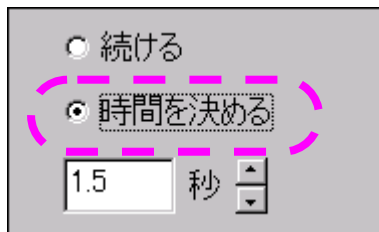
本プログラムに備わっているアクションブロックについて、種類ごとに、機能や設定の変更方法などの詳細を説明します。本項では、頻繁に使用する基本のアクションブロックについて説明します。上級者向けの設定を変更して使えるようになるアクションブロックについては、「2-3. 上級者向けのアクションブロック」をご参照ください。ちなみに、本ソフトウェアをインストールしたフォルダに作成される「¥サンプルプログラム¥基本」のフォルダに、各アクションブロックを用いた非常に簡単なプログラム例を収録しています。そちらのプログラムも合わせてご参照ください。

移動のアクションブロック

移動のアクションブロックは、ロボットの車輪を回して指定の方向に移動させる機能です。移動の方向は「前進」「後退」「右/左旋回」「右/左旋回前進」「右/左旋回後退」「停止」の9種類から選択します。アイコンエリアやプログラムエリアでは、下図のように水色で表示されます。



また、設定エリアに表示される詳細設定項目は下記の通りです。



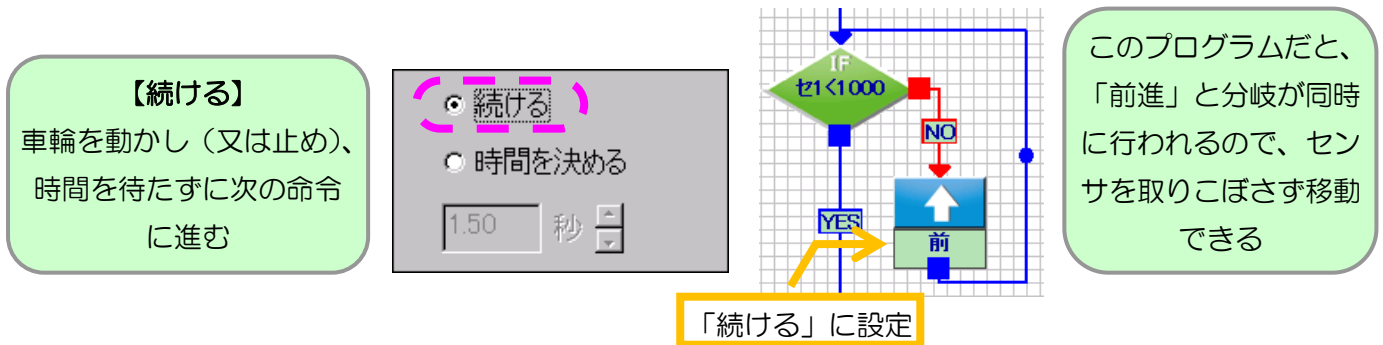
【時間を決める】

指定した時間だけ車輪を回す。時間が来たら車輪を停止する。回す時間も併せて設定する
(時間は 0.01~40.96 秒の範囲)

移動のアクションブロックは、9種類の移動方向に対して、車輪を回す時間を設定します。設定は「続ける」「時間を決める」の二つに大きく分かれ、それぞれの項目をマウスでクリックして選択します。

「時間を決める」は、あらかじめ車輪を動かす時間を決めて、プログラムの実行中はその時間だけ車輪を動かす、指定時間が過ぎたら車輪を停止させて次のアクションに進みます。時間は 0.01~40.96 秒の範囲で、0.01 秒単位で設定できます。なお、設定した時間には車輪の加速・減速に必要な時間も含まれており、同じ時間を設定しても、直前の車輪の回転状況によっては移動距離や動き方が若干異なる場合があります。

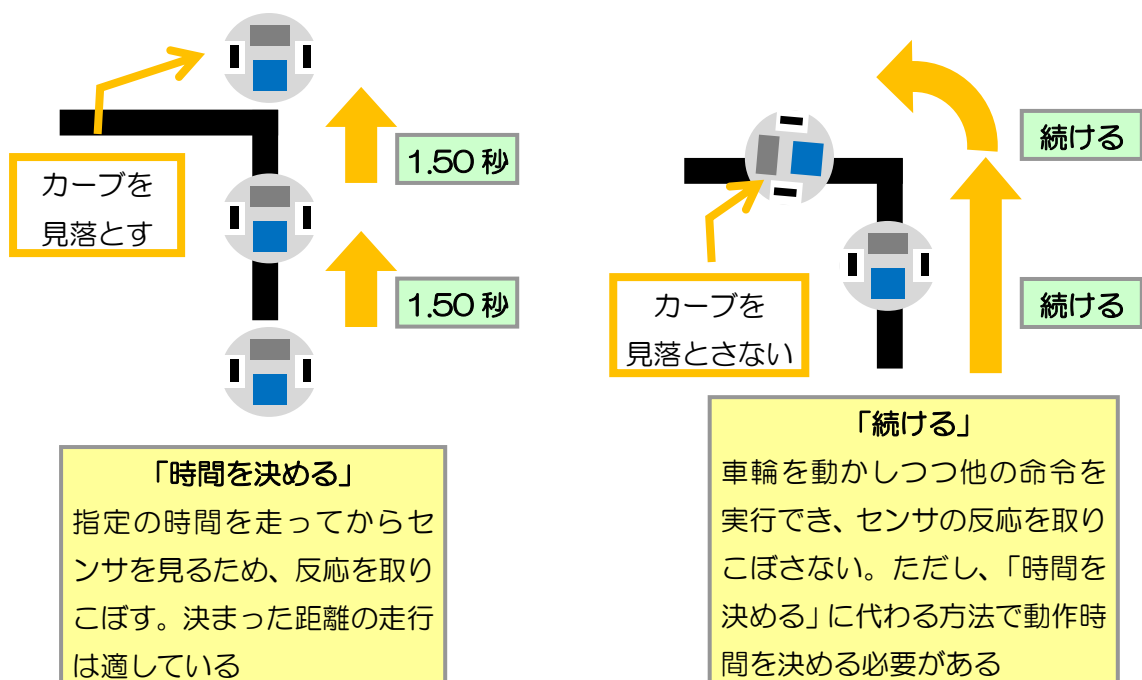
「続ける」は、車輪の動き方を9種類の中から選択したものに設定し、車輪を回したら待つことなくそのまま次のアクションに進みます。こちらの設定は、「車輪を動かしつつ他の命令を実行する」という場合に使います。



前述の「時間を決める」で秒数を指定する場合、ロボットは車輪を動かしたら、他の命令を実行せずに指定した時間過ぎるのを待ちます。この設定の場合、例えばライトレースのプログラムで、その待ち時間中にセンサの数値が変化しても、それに応じてロボットの動作は変わらないため、結果としてロボットがセンサの反応を取りこぼしてしまう可能性があります。

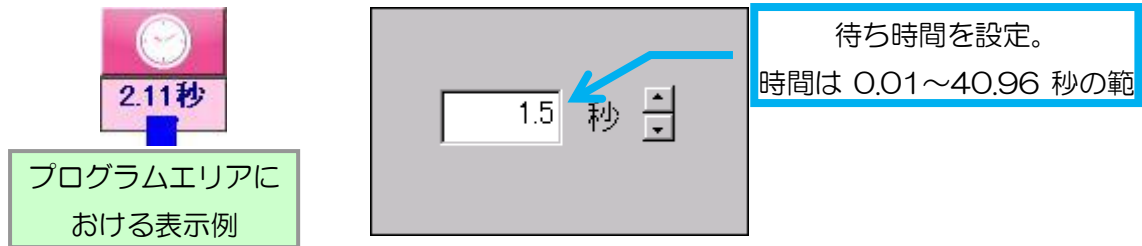
そこで「続ける」に設定を変更すると、ロボットは「車輪を動かした後、一切時間を待たずに次の命令に進む」という動作になります。ここで動かした（又は止めた）車輪は、その状態のままで次の命令を実行していくので、結果として「車輪を動かしつつ（又は止めつつ）他の命令を実行する」ということができます。ライトレースのプログラムなどでも、このようにプログラミングすると、センサの変化を取りこぼすことなくロボットを動かします。ただし、ウェイトなど他の命令を使って、車輪を動かす時間や止めるタイミングを指定する必要があります。

このように、「時間を決める」と「続ける」の設定は、プログラムの目的に応じて使い分けが必要です。前者は「決められた距離・時間動作させる」場合、後者は「センサや変数など他の要素と絡めて動作が柔軟に変化する」場合に、それぞれ利用します。



ウェイトのアクションブロック

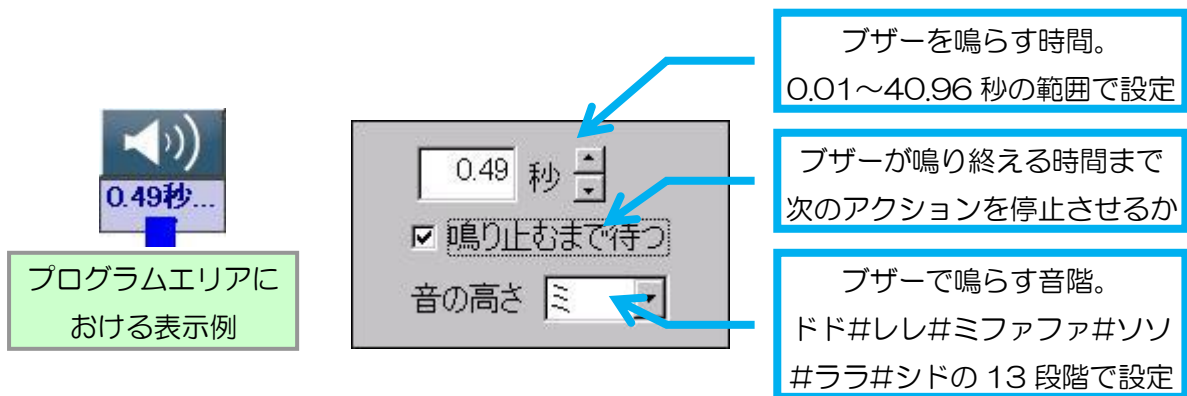
ウェイトのアクションブロックは、「現在のロボットの状態を維持したまま、指定した時間だけ次のアクションに進まずに待つ」という機能です。このアクションは、移動のアクションブロックの「停止」のように車輪を停止したり、ブザーや LED を止めたりせず、それらの状態を維持したままロボットを動作させる命令なので注意してください。アイコンエリアやプログラムエリアでは、下図のようにピンク色で表示されます。また、設定エリアに表示される詳細設定項目は下記の通りです。



ウェイトのアクションブロックは、待ち時間のみを設定します。時間は 0.01~40.96 秒の範囲で、0.01 秒単位で設定できます。

ブザーのアクションブロック

ブザーのアクションブロックは、ロボットの CPU ボードに備わったスピーカーより「鳴らす音の高さ」と「鳴らす時間」を指定してブザー音を鳴らす機能です。アイコンエリアやプログラムエリアでは、下図のように紺色で表示されます。また、設定エリアに表示される詳細設定項目は下記の通りです。



ブザーのアクションブロックは、音程を【ドド#レレ#ミファファ#ソソ#ララ#シド】の 13 種類から選択できます。ブザーを鳴らす時間は 0.01~40.96 秒の範囲で、0.01 秒単位で設定できます。また、「鳴り止むまで待つ」にチェックを入れると、プログラム実行中にアクションを実行する際、指定した時間ブザーが鳴り終わるのを待ってから次のアクションに進むようになります。「鳴り止むまで待つ」のチェックを外すと、指定した時間ブザーを鳴らしますが、プログラム自体はブザーが鳴り始めてからすぐに次のアクションへ進みます。

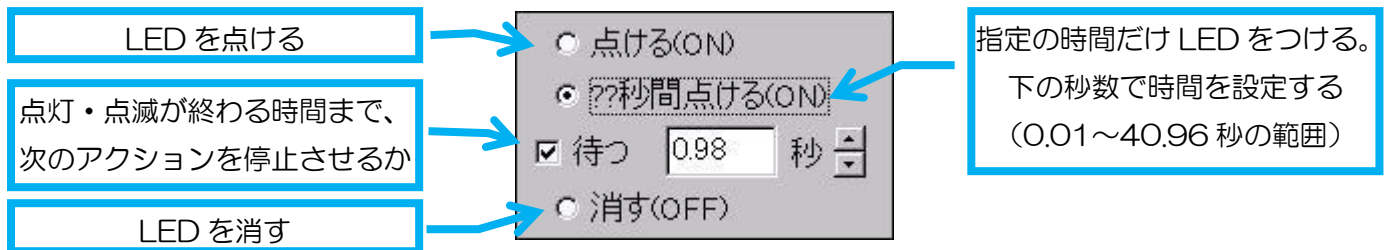
ビュートミニ ARM (スタンダード) でブザーを鳴らす場合は、ブザー拡張オプション (別売) の追加が必要です。

LEDのアクションブロック

LEDのアクションブロックは、ロボットのCPUボードに備わっている二つのLEDを点灯させる機能です。この命令では、LEDをつける(ON)、消す(OFF)、及び、指定した時間だけLEDをつける、の三通りの動作を選択します。この命令はLEDの色(オレンジ・緑)によって二つに分かれます。プログラムエリアでは、それぞれ下図のように表示されます。



また、設定エリアに表示される詳細設定項目は下記の通りです。

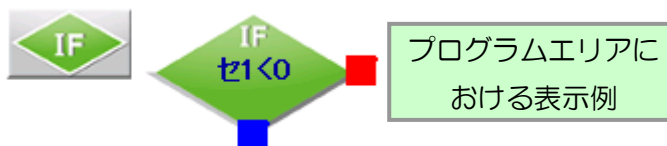


LEDのアクションブロックは、「点ける(ON)」「??秒間点ける(ON)」「消す(OFF)」のいずれかよりアクションを選択します。「点ける(ON)」を選択した場合、LEDを点けた状態で次の命令に進みます。「消す(OFF)」を選択した場合、LEDを消した状態で次の命令に進みます。

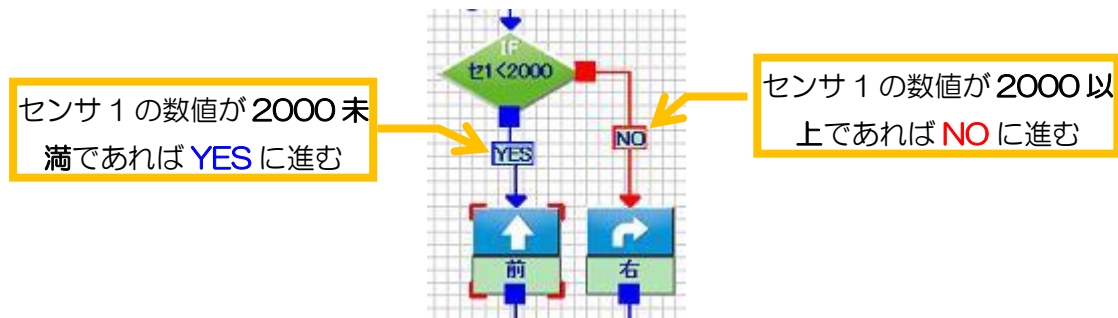
「??秒間点ける(ON)」は、0.01~40.96秒の範囲で指定した時間だけLEDを点けます。この時間の間にも他の命令は通常通り実行され、指定した時間が過ぎると自動的にLEDが消えます。また、指定時間内に他のLEDの命令を実行すると、その設定が適用されます。「待つ」にチェックを入れると、指定した時間が過ぎるのを待って次の命令に進みます。このチェックを外すと、LEDを点けてすぐに次の命令に進みます。

分岐のアクションブロック

分岐のアクションブロックは、センサやボタンの値が指定した数値と比較して「大きいか」「小さいか」「同じ数値か」などの条件を判断し、プログラムを進める経路を選択する機能です。ライトレースなど、センサを使ったプログラムには必ず使用します。アイコンエリアやプログラムエリアでは、下図のように緑のひし形が表示されます。



プログラムエリアでは他のアクションブロックと異なり、青と赤の二つの矢印を持っています。プログラム実行中に分岐のアクションを実行した瞬間、ロボットがセンサの値を読み取って指定の条件が成立するか判断し、成立すれば青い矢印（YES）、成立しなければ赤い矢印（NO）に従ってプログラムを進めます。



また、設定エリアに表示される詳細設定項目は下記の通りです。

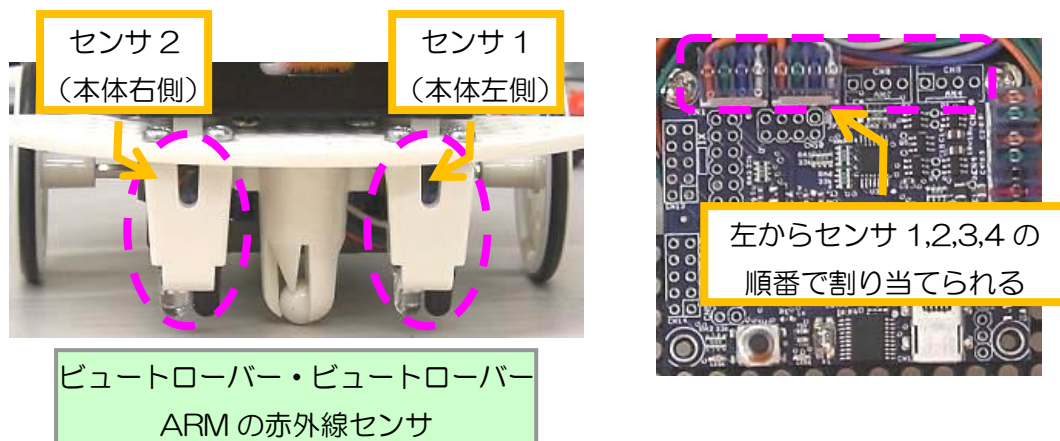
使用するセンサ・ボタンを設定。
センサ 1~4 とボタンから選択

比較する数値を入力。
数値は-32768~32767 の範囲

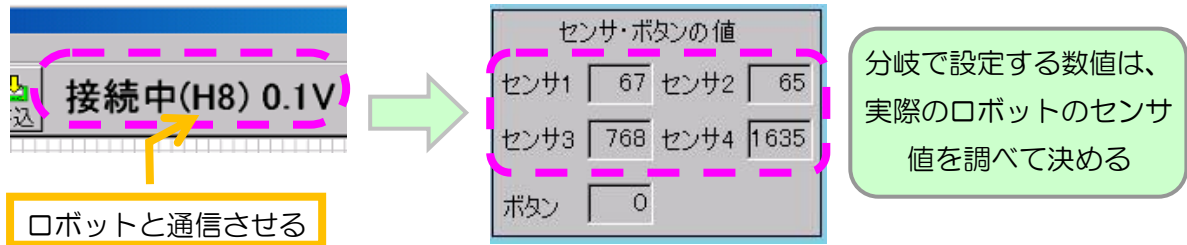
数値を比較する方法を設定。
「より小さい?」「と同じ?」「より大きい?」の三つから選択。

設定する項目は、「どのセンサの値を使用するか」「どの数値と比べるか」「どういう条件で比べるか」の三つになります。

設定項目の 1 行目では、分岐に使用するセンサやボタンを設定します。設定は「センサ 1~4」及び「ボタン」から選んでください。なお、ロボット本体に標準で付属する赤外線センサは、「センサ 1」が本体左側、「センサ 2」が本体右側にそれぞれ該当します。また、CPU ボード上のボタンは「ボタン」に該当します。ロボットにセンサを追加すると、センサ 3,4 にもセンサ値が表示されます。ビュートローバーH8・ARM は Cn8,9 が、ビュートミニ ARM は S3,S4 が、それぞれセンサ 3,4 に該当します。



2行目の項目は、分岐の際に比較させる数値を設定します。項目を設定する場合は、数値をクリックしてカーソルを表示し、キーボードから数値を入力してください。なお、PC がロボットと通信している状態であれば、画面左下の「センサエリア」に、現在のセンサ 1~4 とボタンの値が表示されます。分岐の数値を設定する場合は、実際にロボットを分岐させたい状況にして、そのときのセンサの値を確認して決定してください。



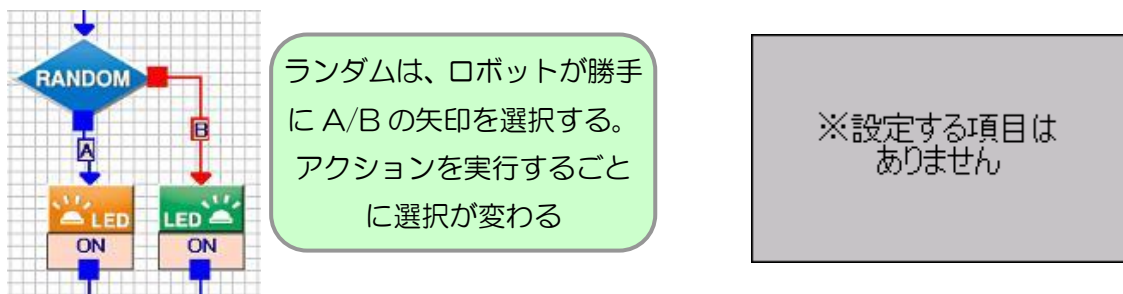
3行目の項目は、使用するセンサ・ボタンと設定した数値をどのように比較させるかを設定します。項目は「より小さい?」「と同じ?」「より大きい?」の三つから選択します。例えばセンサの値が設定した数値より小さいときに YES にしたい場合は「より小さい?」を、反対にセンサの値が設定した数値より大きいときに YES にしたい場合は「より大きい?」を選択してください。

ランダムアクションブロック

ランダムアクションブロックは分岐とよく似ています。ただし、分岐と違って YES と NO に分かれるための条件の設定が存在せず、代わりにロボットがどちらに進むかを勝手に選択します。どちらに進むかはプログラムを実行するたびに変わるので、さいころなどプログラムの中に偶然を組み込む場合に使用します。アイコンエリアやプログラムエリアでは、下図のように青いひし形が表示されます。

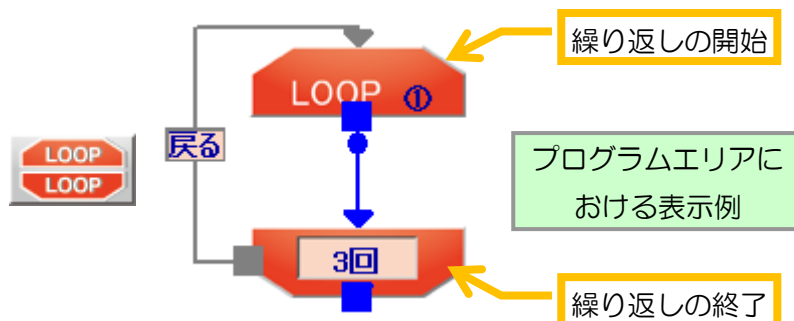


ランダムアクションブロックは、分岐と同じように青と赤の二つの矢印を持っています。プログラム実行中にランダムアクションを実行した瞬間、ロボットが A と B のどちらに進むかを勝手に判断してプログラムを進めます。また、ランダムアクションブロックは設定する項目がありません。



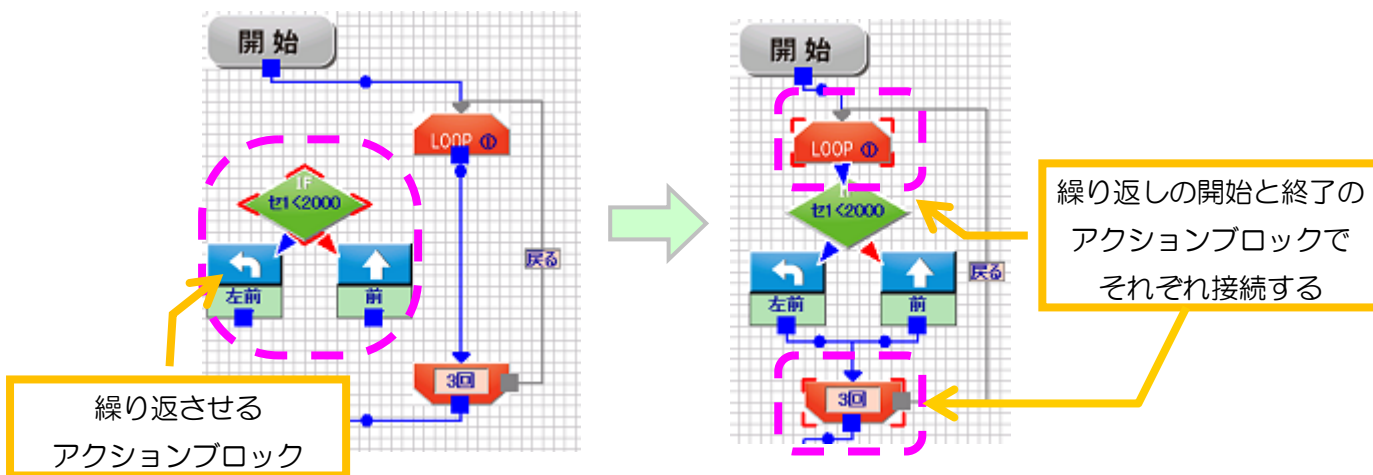
繰り返しのアクションブロック

繰り返しのアクションブロックは、プログラムの一部を、指定した回数だけ繰り返し実行させる機能です。同じアクションを何度か繰り返すプログラムの場合、いくつもアクションブロックを並べずに繰り返しのアクションブロックを使うことで、プログラムを簡単にすることができます。アイコンエリアやプログラムエリアでは、下図のようにオレンジの六角形で表示されます。

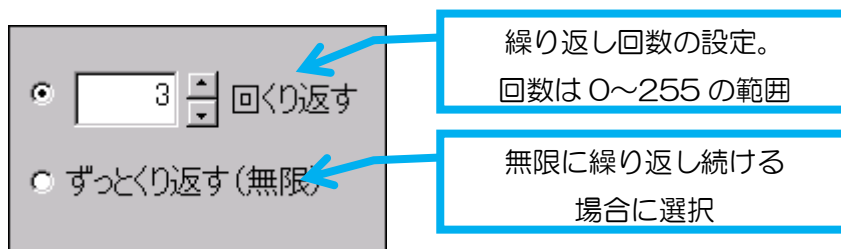


プログラムエリアには、繰り返しの開始と終了の二つのアクションブロックが一度に追加されます。また、繰り返しは一つのプログラムで最大7個までしか使用できません。

繰り返しのアクションブロックをプログラムに組み込む場合は、繰り返すアクションブロックを全て繰り返しのアクションブロックの開始と終了で囲むように矢印を接続してください。必ず繰り返しの開始から終了まで、途中で矢印が途切れることなく接続してください。矢印が途切れていた場合、正しくプログラムを実行できません。

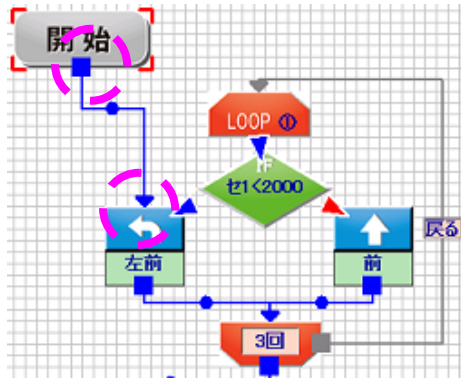


また、設定エリアでは繰り返しの回数を設定します。回数は0~255の範囲で設定し、0の場合は繰り返しの含まれるアクションを一度も実行せずに次に進みます。「ずっと繰り返す」を選ぶと、無限に繰り返し続けます。

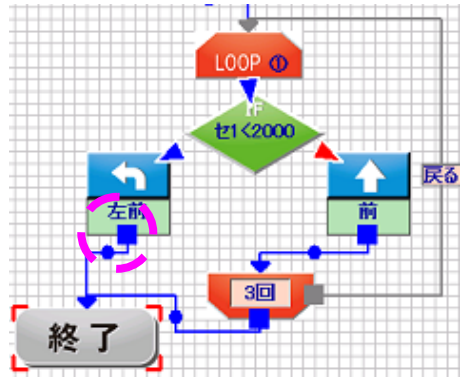


繰り返しのアクションブロックを下図のように接続すると正しく繰り返しが行われません。このような場合、プログラムを実行・書き込みしようとすると画面に警告が表示され、関係するアクションブロックが全て選択状態になります。警告が表示された場合、問題の箇所を確認してプログラムを修正してください。

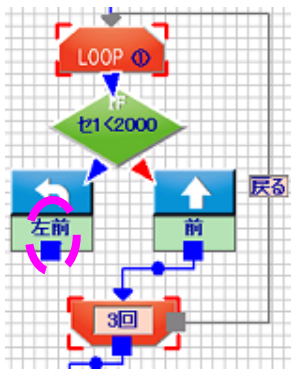
以下の場合には繰り返しが正しく行われません



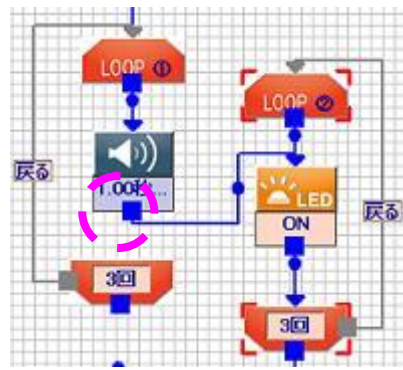
繰り返しの開始ではなく繰り返しの
中身に矢印を接続している



繰り返しの終了を通らず、繰り返しの
外のブロックに接続している



繰り返しの途中で矢印が途切れている



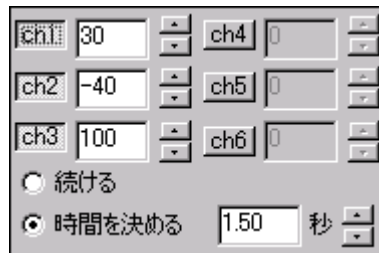
繰り返しの途中で別の繰り返しに
接続している

2-3. 上級者向けのアクションブロック

「3-5. 上級者向け機能の設定」で説明している「上級者向け機能」を変更すると、いくつかのアクションブロックがアイコンエリアに追加されます。これらのアクションブロックは上級者向けの命令で、設定は複雑になりますがロボットをより自由に動かせるようになります。本項では、これらの上級者向けのアクションブロックについて説明します。

車輪制御のアクションブロック

「上級者向け機能」の設定より「速度や旋回量を変更できる車輪制御ブロックを使う」を有効にすることで、モータの速度を柔軟に設定できる移動のアクションブロックを使用できます。この命令を「車輪制御」のアクションブロックといいます。車輪制御のアクションブロックは下記のアイコンで表示されます。また、設定エリアに表示される詳細設定項目は下記の通りです。



どのモータを、どのスピードで動かすかを設定。

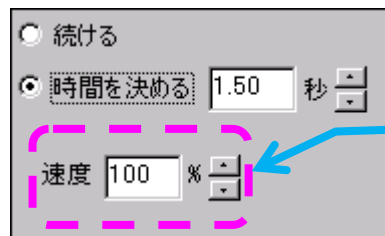
時間を指定して走らせるか否かを設定

車輪制御の命令では、CPU ボードが使用できる全ての数のモータ（最大 6 個）に対応しています。それぞれのモータに対して、回転の方向と速さを個別に設定できます。ch1～ch6 のボタンは、各モータに速度の指示を送るかどうかを選択します。プログラムを実行すると、ボタンが押されているモータだけが指定の速度に変わり、ボタンが押されていないモータは現在の状態を維持します。

各モータのスピードを設定する数値は、-127～127 の範囲で設定します。プログラムを実行すると、この数値が直接各モータ速度のメモリに代入されます。

一番下の「続ける」「時間を決める」の設定は、移動のアクションブロックと同一です。

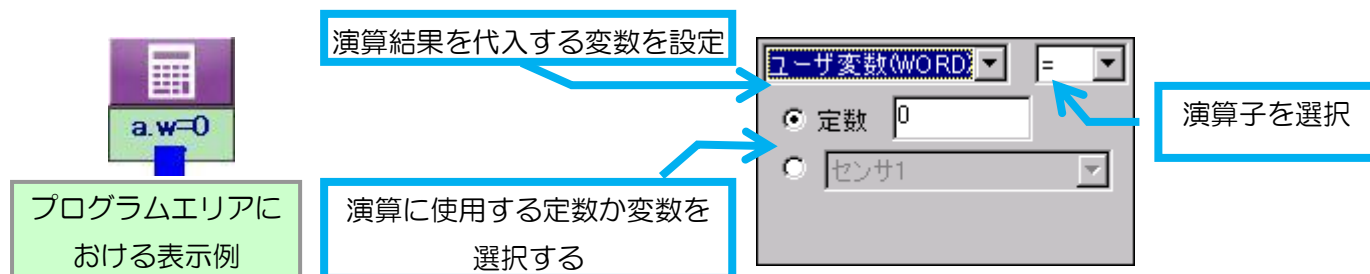
また、この車輪制御ブロックを使用できる状態では、移動のアクションブロックも下図のように変わります。この場合、各方向へ移動する際の車輪の速度を変更できます。数値は 0～100 の範囲で設定し、数値が大きいほど速く移動します。



移動のアクションブロックも、速度を設定できるようになる

変数演算のアクションブロック

「上級者向け機能」の設定より「演算ブロックとメモリマップウィンドウを有効にする」を有効にすることで、ロボットの変数を利用した複雑なプログラミングが可能になります。この命令を使用すると、センサの平均値を求めると、プログラム中に計算を組み込むことができます。変数演算のアクションブロックは下記のアイコンで表示されます。また、設定エリアに表示される詳細設定項目は下記の通りです。



変数演算の主な動作は、「変数一つを選択し、その変数の現在の値に、別の定数、もしくは変数を四則演算し、求まった答えを最初に選択した変数に代入する」というものです。

1行目の左は、演算結果を代入する変数を選択します。1行目の右は四則演算の方法を選択します。「+=」「-=」「×=」「÷=」はそれぞれ加減乗除を表し、「=」は2~3行目で設定する定数・変数をそのまま代入します。また「%=」は、「演算結果に割り算の余りを代入する」命令です。

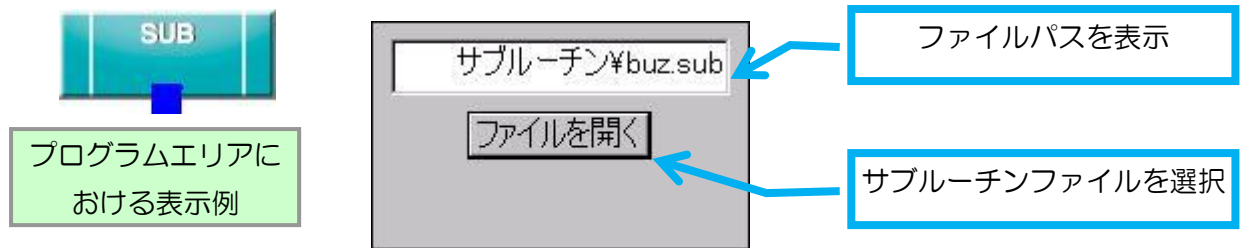
2行目は、演算に使用する定数を設定します。また、3行目は演算に使用する変数を設定します。2行目と3行目は、目的によって選択してください。

なお、各項目で選択できる変数のみが、その項目で使用可能であり、「ある項目では選択できる変数がある項目では選択できない」というケースもあります。また、設定によってはエリア下に「右辺と左辺のサイズが違いため計算できません」又は「右辺と左辺の符号が違いため計算できません」という警告が表示される場合があります。これは、変数同士を演算する場合に、二つの変数のサイズや負の値を使えるかどうか異なる则表示され、このままでは正常にプログラムを実行できません。そのため、どちらかの変数を差し替えて、二つの変数の設定を合わせる必要があります。変数の設定に関する説明については「3-6. メモリマップの表示」もご参照ください。

サブルーチンのアクションブロック

「上級者向け機能」の設定より「サブルーチンブロックを使う」を有効にすることで、別のシーケンスファイルを読み込んでプログラムに組み込めるようになります。ここでファイルより読み込んだシーケンスのまとまりを「サブルーチン」と呼びます。

サブルーチンのアクションブロックは下記のアイコンで表示されます。



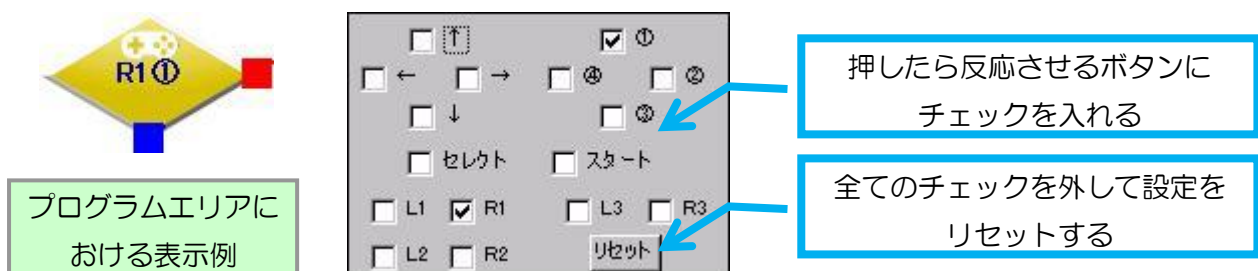
設定エリアには、現在指定しているシーケンスファイルのファイルパスを表示しています。また、「ファイルを開く」ボタンをクリックすると、サブルーチンとして使用するファイルを選択できます。

サブルーチンを使用する場合、あらかじめサブルーチンとして使えるプログラムを作成してファイルに保存する必要があります。プログラムをサブルーチンとして保存する場合は、メニューの「ファイル」→「サブルーチンとして保存」をクリックして行ないますが、注意として「サブルーチンの中で更に別のサブルーチン呼び出す」ということが出来ないため、サブルーチンが含まれているプログラムをサブルーチンとして保存することは出来ません。

また、サブルーチンのファイルは、呼び出し元となるシーケンスファイルとの相対パスの関係で呼び出されます。そのため、サブルーチンのファイルは、それを呼び出すプログラムファイルと同じフォルダ、もしくはその下の階層のフォルダになるべく配置してください。

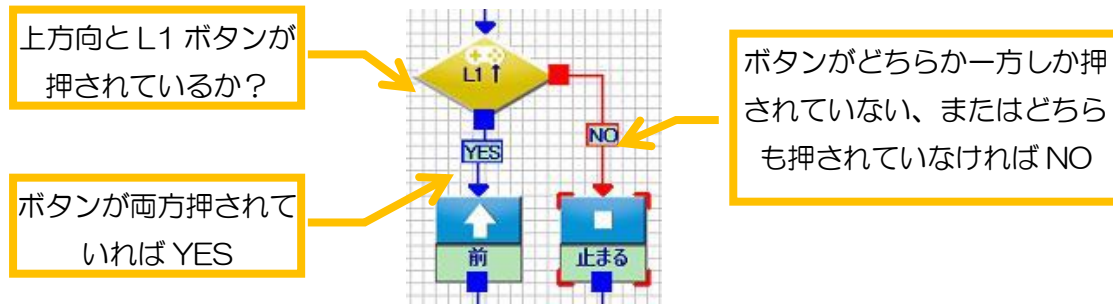
ゲームパッド分岐のアクションブロック

「上級者向け機能」の設定より「ゲームパッドの入力ブロックを使う」を有効にすることで、ゲームパッドのボタン入りに特化した分岐のアクションブロックを使用できます。また、「VS-C2 for Android」で Android 端末から操縦する場合も、このブロックを使うことが出来ます。ゲームパッドの接続方法は、CPU ボードに関する別資料を、また、VS-C2 for Android の詳細については、[VS-BT001 の商品ページ](#)を、それぞれご参照ください。アイコンエリアやプログラムエリアでは、下図のように黄色いひし形で表示されます。

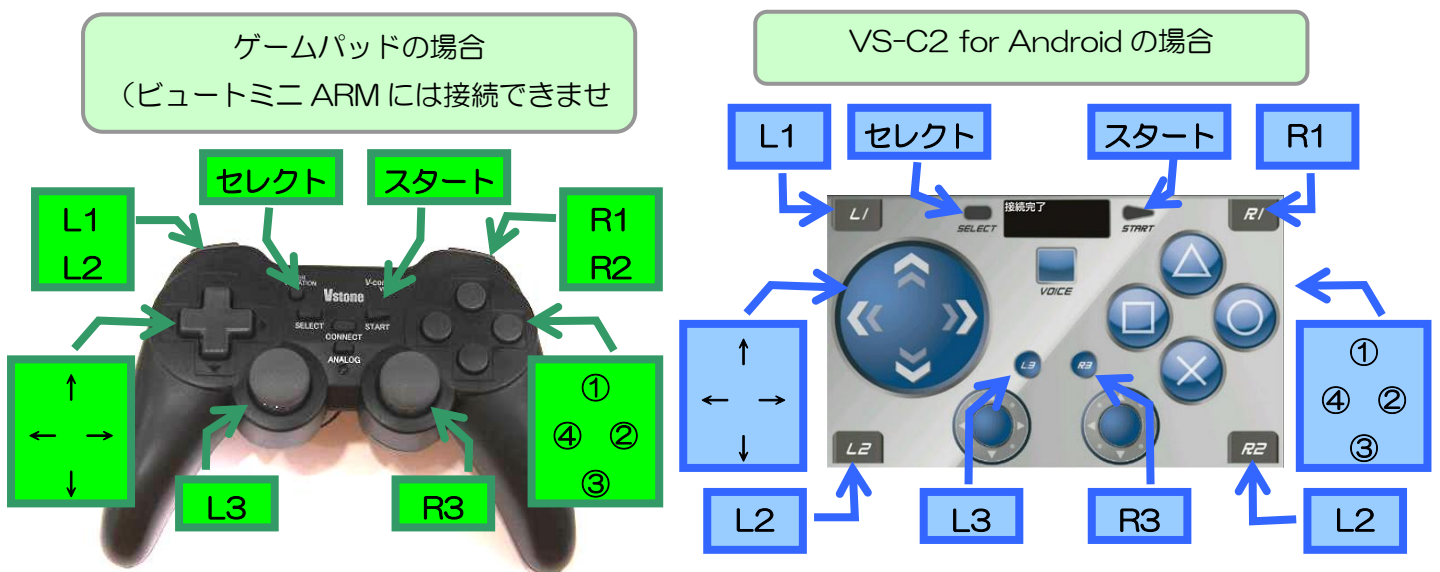


ゲームパッド分岐のアクションブロックは分岐と同じく矢印が二つ備わっています。設定エリアにはゲームパ

ツドのボタンごとにチェックボックスが存在し、チェックを入れたボタンがプログラム実行中に押されているかどうかで分岐します。ボタンは複数同時に入力させることができ、「指定した全てのボタンが入力されていたら YES」「指定のボタンが一つでも押されていないならば NO」にそれぞれ進みます。また、「リセット」のボタンを押すと、ボタンのチェックが全て外れます。



各チェックボックスの記号と、それに対応するゲームパッド上のボタンは以下の通りです。



分岐のアクションブロック (上級者向け)

「上級者向け機能」の設定より「演算ブロックとメモリマップウィンドウを有効にする」を有効にすると、分岐のアクションブロックの設定内容が変数演算に合わせて変化します。このとき設定エリアに表示される詳細設定項目は下記の通りです。

演算ブロックを使用できるようにすると、分岐ブロックの設定も多くの変数を使えるようになる

分岐に使用する変数を選択

比較する定数、もしくは変数を選択

比較する条件を選択

各設定は基本的に通常の分岐と同じです。大きな違いは、センサ・ボタン以外の変数を使用できること、及び、変数同士を比較できることです。なお、選択した変数の種類によっては、2行目で使用できる定数の設定範囲が、-32768~32767ではなく0~65535になる場合があります。この関係で、使用する変数を切り替えた場合にも自動的に定数が切り替わる場合があります。

1 行目は、比較に使用する変数を選択します。2 行目は 1 行目の変数と比較する定数を設定します。3 行目は 1 行目と比較する変数を選択します。2 行目と 3 行目は、変数演算と同じく目的に応じて選択してください。4 行目は二つを比較する方法を選択します。比較する方法は通常の方と同じです。

なお、変数演算と同じく、1 行目と 3 行目では選択できる変数が異なるものもあります。また、設定によってはエリア下に「右辺と左辺のサイズが違いため比較できません」又は「右辺と左辺の符号が違いため比較できません」という警告が表示される場合があります。これは、変数同士を比較する場合に、二つの変数のサイズや負の値を使えるかどうか異なる場合に表示されます。このままでは正常にプログラムを実行できないため、どちらかの変数を差し替えて、二つの変数の設定を合わせる必要があります。変数の設定に関する説明については「3-6. メモリマップの表示」もご参照ください。

エンコーダを使った車輪制御のアクションブロック

ビュートミニ ARM にロータリーエンコーダを取り付け、「エンコーダの設定」より「エンコーダを使った車輪制御命令を使用する」と「エンコーダのフィードバックを有効にする」にチェックを入れると、エンコーダを利用した車輪制御命令を使うことができます。

プログラムエリアにおける表示例

距離 10 cm

角度(10度単位) 0 度

速度(0~127) 127

※この命令はビュートミニARM専用です。

距離を cm で設定。正負の数値を設定できる

角度を設定。10 度単位で正負の数値を設定できる

スピードを設定

エンコーダを使った車輪制御はビュートミニ ARM のみ対応しています。
また、必ずエンコーダのフィードバックを有効にしてください。

エンコーダを使った車輪制御のアクションブロックは、前進や車輪制御などと同じくロボットを移動させるものですが、エンコーダの機能を活かして距離と角度を指定して動かします。

設定エリアの各項目の説明は次の通りです。「距離」は、ロボットの移動距離を 10cm 単位で指定します。数値はプラス・マイナス両方を与えることができ、プラスは前進・マイナスは後退します。「角度」はロボットの回転角度を 10 度単位で与えます。数値はプラス・マイナス両方を与えることができ、プラスは時計周り、マイナスは反時計回りになります。「速度」はロボットの移動速度を 0~127 の範囲で指定します。

モータとタイヤがスリップしたり、地面の状態（凸凹が激しいなど）により、距離・角度に誤差が発生する場合がありますのでご了承ください。

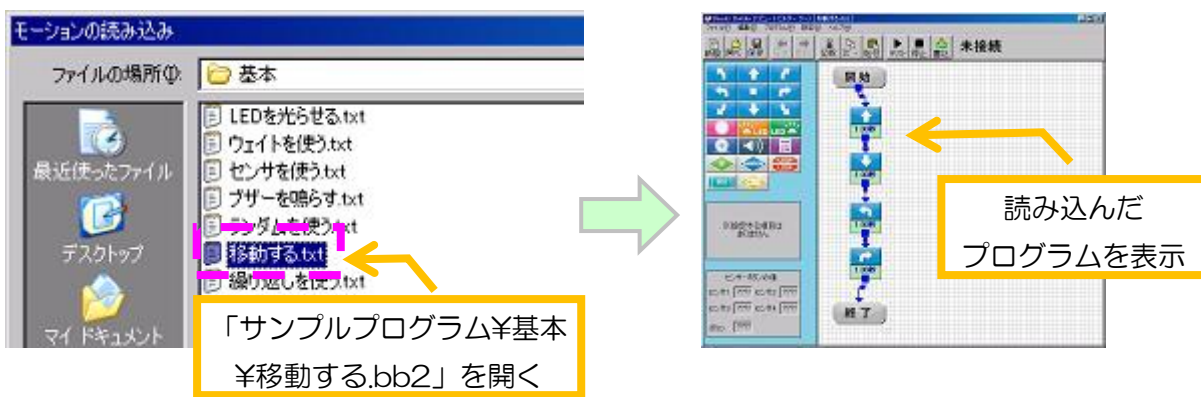
2-4. プログラムの実行・保存

本ソフトウェアで作成したプログラムをロボットに実行させる方法について説明します。なお、本ソフトウェアからロボットにプログラムを実行させる方法は、「テスト実行」と「書き込んで実行」の二通りがあります。それぞれに長所と短所があり、また細かい部分で動作に違いが現れる場合があるので、それぞれの方法の説明をよく読んでください。

サンプルプログラムの読み込み

プログラムを実行させる前に、本ソフトウェアでロボットに書き込むプログラムを作成する必要があります。ここでは例として、本ソフトウェアに含まれる「移動する.bb2」のサンプルプログラムを読み込んでロボットに実行させます。

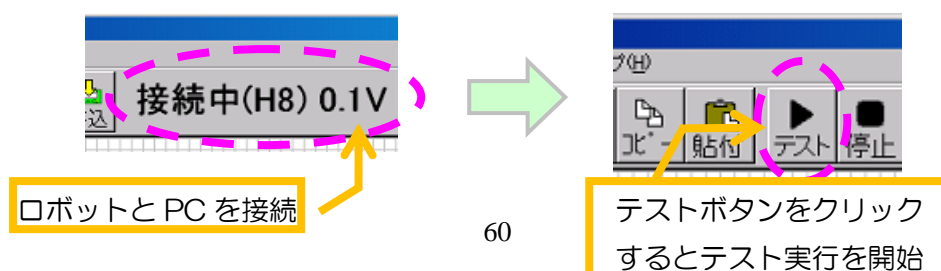
メニューより「ファイル」→「開く」をクリックしてください。クリックするとファイルを選択するダイアログを開くので、「サンプルプログラム」のフォルダをダブルクリックし、更に「基本」のフォルダをダブルクリックしてください。すると、開いたフォルダに「移動する.bb2」というファイルがあるので、このファイルを選択して「開く」をクリックしてください。クリックするとファイルからプログラムを読み込んで、その内容を下記のようにプログラムエリアに表示します。



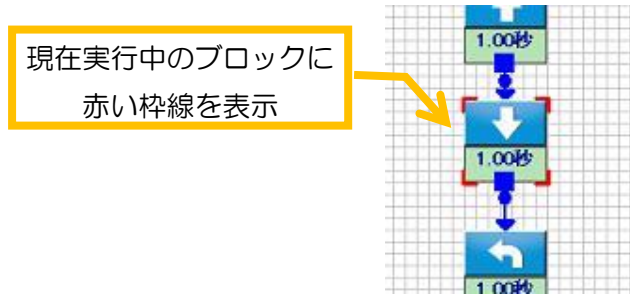
プログラムのテスト実行

「テスト実行」は、PC から通信メッセージでロボットに命令を送りプログラムを実行させます。この場合、ロボット本体に書き込まれているプログラムデータは消えませんが、通信に若干時間がかかるため、ロボット本体にプログラムを書き込んで実行するときよりも全体的に動作が遅くなります。あくまで、作成したプログラムが正しい順序でほぼ想定した動作を行なうか確認のためにご利用ください。

プログラムをテスト実行する場合は、PC とロボットを接続し、ロボットの電源を ON にしてください（ロボットの電源が OFF の場合、車輪が回りません）。続いて、ツールバーのテストボタン（▶のアイコン）をクリックしてください。クリックするとテスト実行を開始します。



プログラムのテスト実行中は、現在ロボットが実行しているアクションに相当するアクションブロックが赤い枠で囲まれます。この枠線を目で追うことで、プログラムの流れの確認が非常に便利になります。



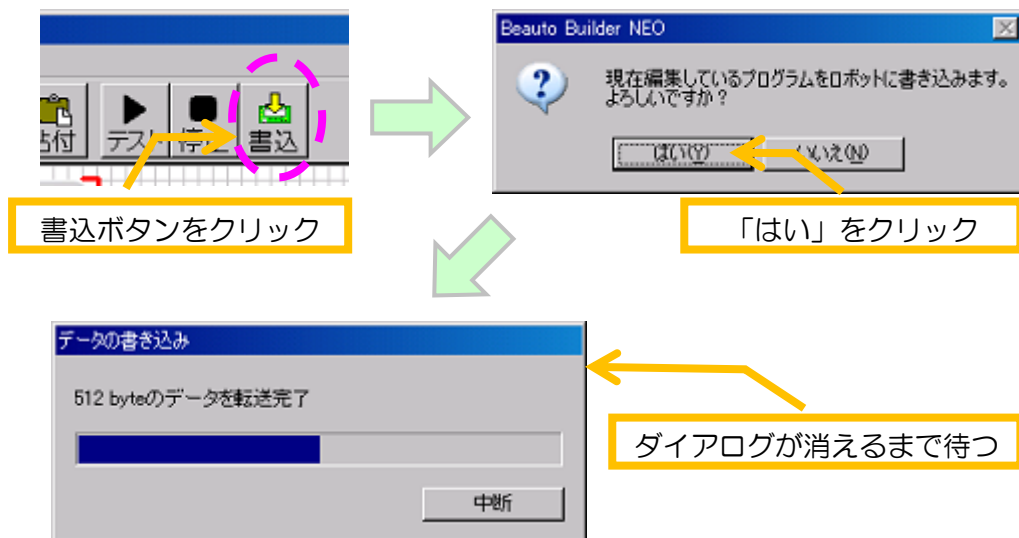
テスト実行を停止する場合は、ツールバーの停止ボタン（■のアイコン）をクリックしてください。

テスト実行中は、ロボットから USB ケーブルが抜けないように注意してください。

プログラムの書き込み・実行

ロボット本体にプログラムを書き込んで実行する方法は、PC にロボットを接続せずに単体でプログラムを実行するものです。この場合、ロボットの電源を切っても書き込んだプログラムが保存されるため、PC の無い場所でもロボットを動かすことができます。また、通信のための余計な時間がかからないため、各アクションに設定した時間通りにプログラムが実行されます。ただし、ロボット本体には 1 種類しかプログラムを保存できないため、新しいプログラムを書き込むと古いプログラムデータが消えます。

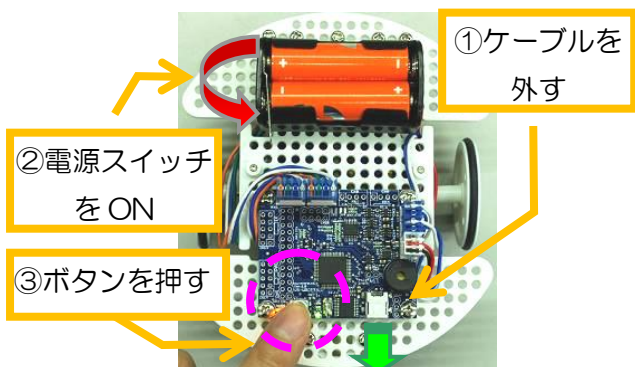
ロボットにプログラムを書き込む場合は、まず、PC とロボットを通信ケーブルで接続してください。接続したら、ツールバーの書き込ボタン（📁のアイコン）をクリックしてください。クリックすると本当に書き込みを行なうか確認のメッセージを表示するので、書き込んで問題なければ「はい」をクリックしてください。クリックすると、画面にプログラムの書き込み状況を表すウィンドウを表示して書き込みが行なわれます。書き込み状況を表すウィンドウが消えたら書き込みが完了です。



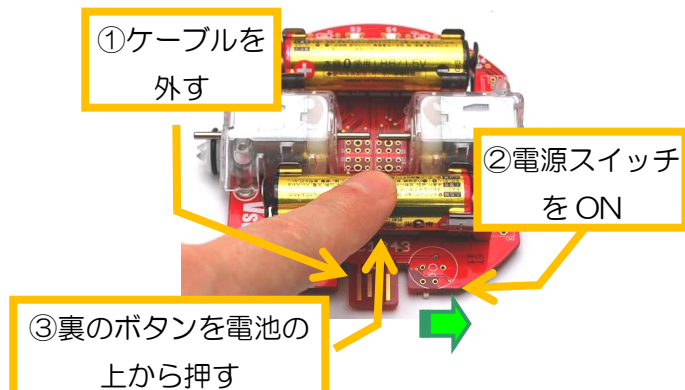
プログラムの書き込み中に、ロボットのボタンを押したり通信ケーブルを抜いたりしないでください。

ロボットに書き込んだプログラムを実行する場合は、ロボットから通信ケーブルを外し、ロボット本体の電源スイッチをONにしてボタンを押してください。ボタンを押すとプログラムを開始します。プログラムを停止する場合はロボットの電源スイッチをOFFにしてください。

ビュートローバー・ビュートローバー-ARM

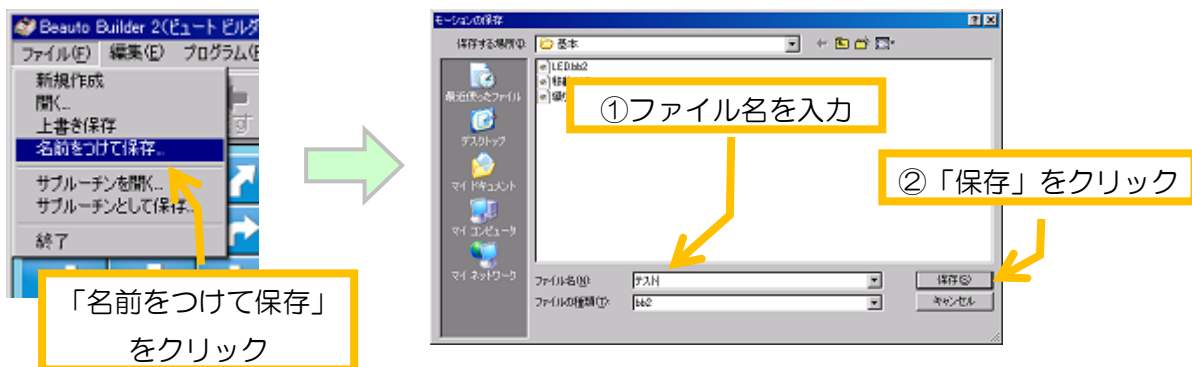


ビュートミニ ARM

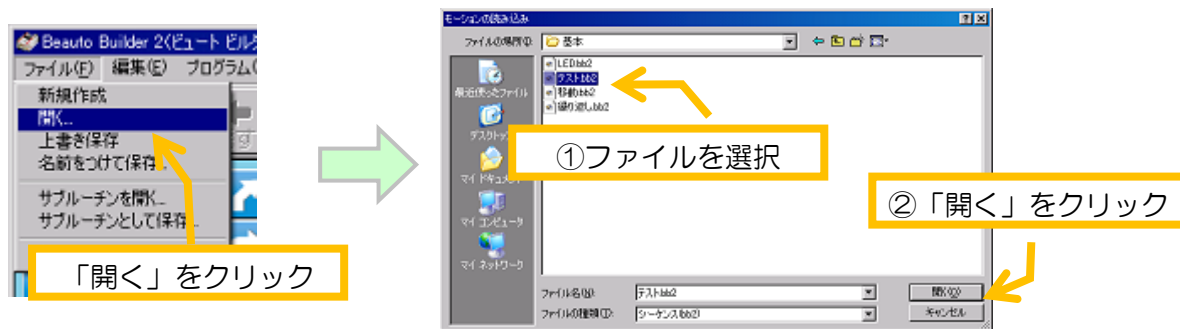


プログラムの保存・読み込み

本ソフトウェアで作成したプログラムは、PC にファイルとして保存することができます。プログラムをファイルに保存する場合は、メニューより「ファイル」→「名前をつけて保存」をクリックしてください。クリックするとファイルを保存するフォルダと名前を設定するダイアログを開くので、保存先と名前を設定してダイアログの「保存」をクリックしてください。

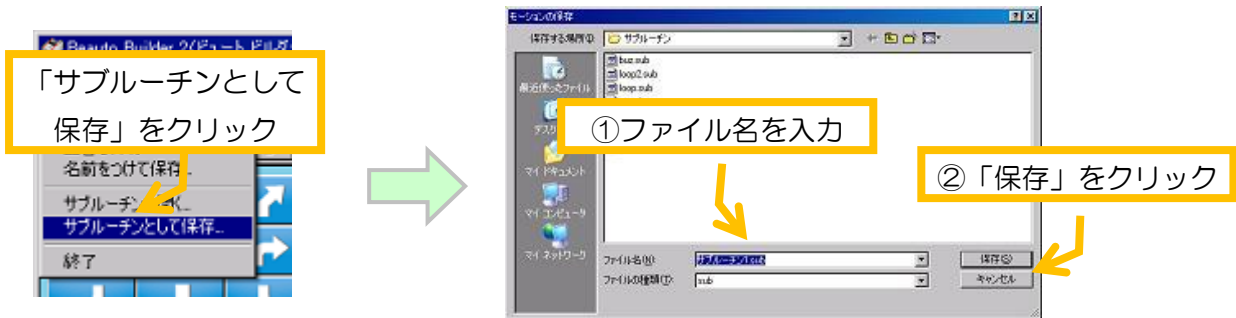


また、ファイルに保存したプログラムを読み込む場合は、メニューより「ファイル」→「開く」をクリックしてください。クリックすると、読み込むファイルを選択するダイアログを表示するので、読み込むファイルをクリックして「開く」をクリックしてください。

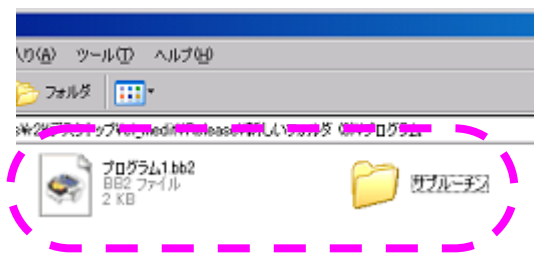
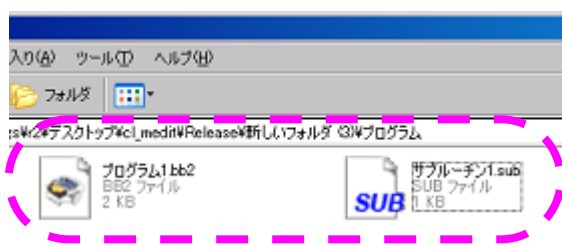


サブルーチンの保存・読み込み

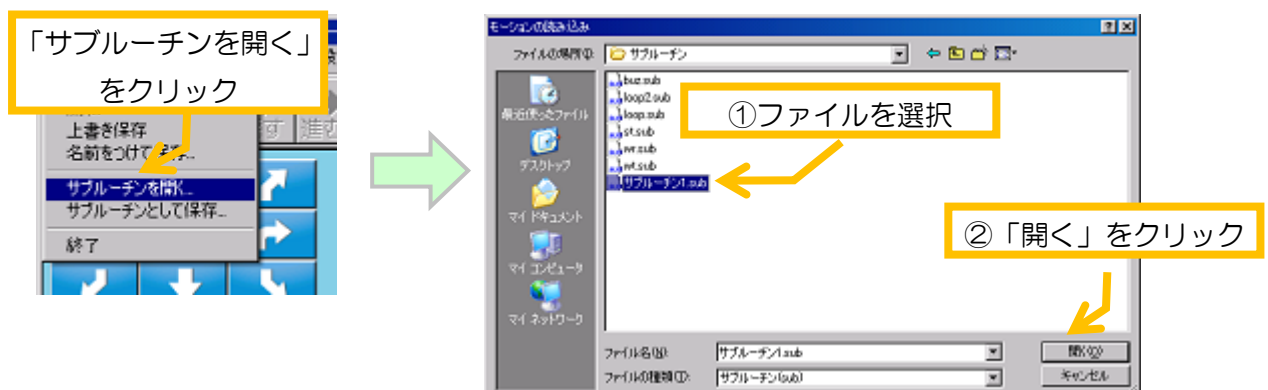
作成したプログラムをサブルーチンとして利用する場合は、サブルーチンの形式でファイルに保存する必要があります。なお、サブルーチンの中に別のサブルーチンを含めることは出来ません。プログラムをサブルーチンの形式で保存する場合は、メニューより「ファイル」→「サブルーチンとして保存」をクリックしてください。クリックすると保存するファイル名を選択するウィンドウを開くので、ファイル名を入力して「保存」をクリックしてください。



注意として、サブルーチンを保存するフォルダは、サブルーチンを使うプログラムと同じ、またはその下の階層のフォルダにしてください。このように保存することで、プログラムとサブルーチンを別のフォルダにコピーしても問題なく読み込みできるようになります。



また、ファイルに保存したサブルーチンを読み込む場合は、メニューより「ファイル」→「サブルーチンを開く」をクリックしてください。クリックすると読み込むファイルを選択するウィンドウを開きます。



2-5. ツールバー・メニューの項目と機能

本ソフトウェアに備わっているメニューの項目とツールバーのボタンの機能について説明します。なお、同一の機能を持つメニュー項目とツールバーのボタンは、項目名の後に括弧でツールバーのアイコンを表示しています。

ファイル

- **新規作成** (📄)・・・現在作成中のプログラムを破棄し、新しいプログラムの作成を開始します。
- **開く** (📁)・・・ファイルに保存したプログラムを読み込みます。
- **上書き保存** (💾)・・・現在作成中のプログラムを上書き保存します。ファイル名の無いプログラムの場合は名前を付けて保存を行いません。
- **名前をつけて保存**・・・現在作成中のプログラムを、名前を変更して保存します。
- **サブルーチンを開く**・・・ファイルに保存したサブルーチンを読み込みます。
- **サブルーチンとして保存**・・・現在作成中のプログラムを、サブルーチンの形式でファイルに保存します。
- **終了**・・・本ソフトウェアを終了します。

編集

- **元に戻す** (↶)・・・現在作成中のプログラムをひとつ前の状態に戻します。
- **やり直し** (↷)・・・「元に戻す」で戻したプログラムの状態を一つ進めます。
- **切り取り** (✂)・・・現在プログラムエリアで選択しているアクションブロックをバッファにコピーして削除します。アクションブロックの切り取り・コピー・貼り付けについては、「2-1-2.アクションブロックの操作説明」をご参照ください。
- **コピー** (📄)・・・現在プログラムエリアで選択しているアクションブロックをバッファにコピーします。
- **貼り付け** (📄)・・・「切り取り」及び「コピー」でバッファにコピーしたアクションブロックをプログラムエリアに貼り付けます。
- **削除**・・・現在プログラムエリアで選択しているアクションブロックを削除します。
- **すべて選択**・・・現在プログラムエリアに存在するすべてのアクションブロックを選択状態にします。

プログラム

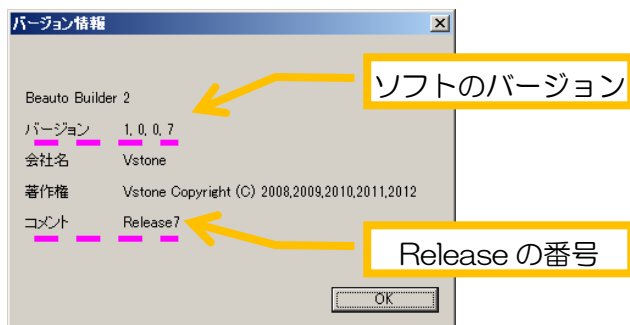
- **テスト実行** (▶)・・・現在作成中のプログラムを本ソフトウェアからテスト実行します。プログラムのテスト実行、プログラムの書き込みについては、「2-3.プログラムの実行・保存」をご参照ください。
- **停止** (■)・・・プログラムのテスト実行を停止します。
- **プログラムの書き込み** (📥)・・・現在作成中のプログラムをCPUボードのROMに書き込みます。

設定

- **モータの設定**・・・・・・・・・・ロボット本体の車輪に関する詳しい設定を行いません。設定の具体的な説明については、「3.ロボットの詳細設定」をご参照ください。
- **エンコーダ・IXBUS・Bluetooth の設定**・・・・・・・・・・ロータリーエンコーダ・IXBUS・Bluetooth など、ロボットの拡張機能に関する設定を行います。設定の具体的な説明については、「3.ロボットの詳細設定」をご参照ください。
- **上級者向け機能の設定**・・・・・・・・・・上級者向けの各種機能を使用するかどうかを設定します。設定の具体的な説明については、「3-5. 上級者向け機能の設定」をご参照ください。
- **メモリマップの表示**・・・・・・・・・・上級者向け機能の一つであるメモリマップウィンドウの表示を有効にしている場合、メモリマップウィンドウを画面に表示します。

ヘルプ

- **バージョン情報**・・・・・・・・・・本ソフトウェアのコピーライト表示、及びバージョン情報を確認します。この項目をクリックすると、下記のダイアログを表示します。



- **ファームウェア書き換え**・・・・・・・・・・ロボット本体のCPUボードに書き込まれたプログラムを更新します。製品のサポートページで新しいバージョンのファームウェアが公開され、お使いのロボットをそのプログラムに書き換える場合に使用します。

ショートカットキーの活用

PCのキーボードより特定のキー入力を行なうことで、本説明書で紹介しているメニューの一部の項目を呼び出したり、テキスト編集に利用したりできます。この機能をショートカットキーといいます。ショートカットキーは、マウスのカーソルがプログラムエリア、及び設定エリアにあるときに、それぞれのエリア上の操作を行います。本ソフトウェアに備わっているショートカットキーは以下の通りです。

プログラムエリアのショートカットキー

- ・ CTRL (コントロールキー) +A・・・「編集」→「すべて選択」
- ・ CTRL+C・・・・・・・・「編集」→「コピー」
- ・ CTRL+V・・・・・・・・「編集」→「貼り付け」
- ・ CTRL+X・・・・・・・・「編集」→「切り取り」
- ・ DELETE キー・・・「編集」→「削除」
- ・ CTRL+Q・・・・・・・・「ファイル」→「終了」
- ・ CTRL+Z・・・・・・・・「編集」→「元に戻す」
- ・ CTRL+Y・・・・・・・・「編集」→「やり直し」

設定エリアのショートカットキー

- ・ CTRL (コントロールキー) +A・・・テキストを全て選択
- ・ CTRL+C・・・・・・・・選択したテキストをコピー
- ・ CTRL+V・・・・・・・・選択したテキストを貼り付け
- ・ CTRL+X・・・・・・・・選択したテキストを切り取り
- ・ DELETE キー・・・選択したテキストを削除
- ・ CTRL+Z・・・・・・・・テキストの変更を元に戻す
- ・ CTRL+Y・・・・・・・・元に戻したテキストの操作をやり直す

3. ロボットの詳細設定

本ソフトウェアには、ロボットに関する設定を変更する機能が備わっています。本項では、それらの機能について紹介します。

各種設定の変更は、必ずロボットを接続した状態で行ってください。
一部の項目は、ロボットが接続されていないと変更できません。

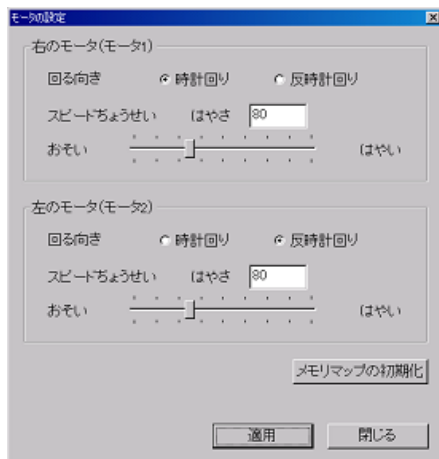
3-1. 車輪の速度・回転方向の設定

車輪に関する設定では、左右の車輪それぞれの回転速度、及び回転の方向を設定できます。左右の車輪の速度が微妙に異なることが原因で、ロボットが正確に前進できない場合に、左右の車輪の速度を合わせる目的などで利用します。また回転方向については、ビュートローバーでギアボックスを組み替えてモータの回転方向が実際の命令と逆転した場合に設定を変更すると、進行方向をプログラムに合わせられます。

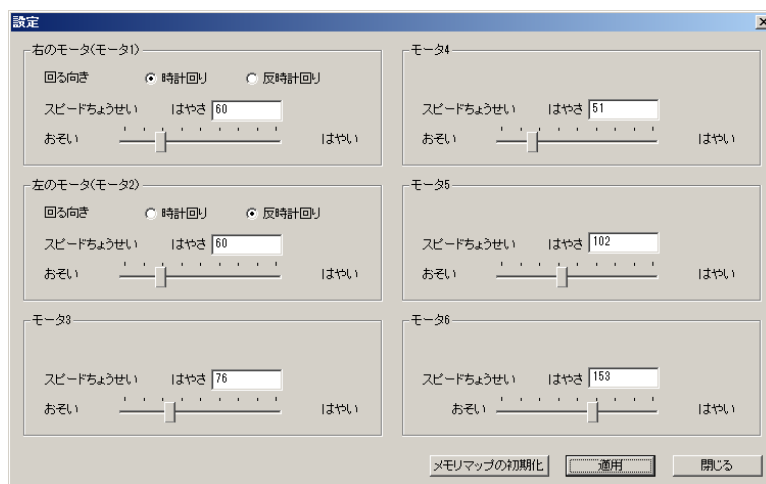
設定を行う場合、メニューの「設定」→「モータの設定」をクリックして「モータの設定」を開きます。

モータの設定は、本ソフトウェアの標準設定では、左下図のように左右車輪のみを設定できますが、上級者向け機能の設定で「速度や旋回量を変更できる車輪制御ブロックを使う」にチェックを入れると、右下図のように6個分設定できます。

左右車輪のみの設定（標準）



6個全てのモータ設定（上級者向け機能設定を変更）

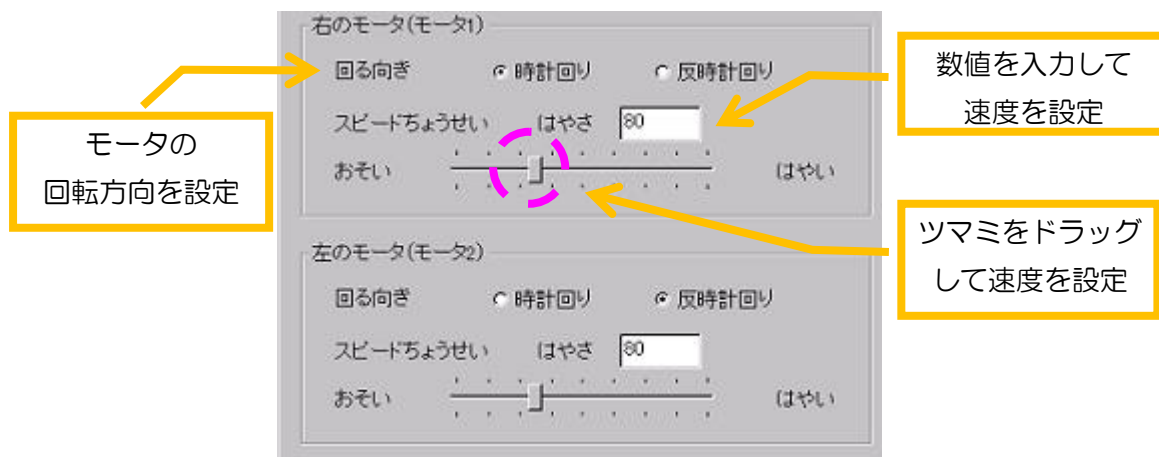


ビュートミニ ARM はモータ3～6に対応していません

車輪の設定は、モータごとに「回る向き」「スピードちょうせい」の二つの項目が備わっています。

「回る向き」は車輪の回る向きです。車輪を側面から見たときの回転方向を設定します。「時計回り」「反時計回り」のいずれかを選択できます。

「スピードちょうせい」は、車輪の回転速度です。「はやさ」の箇所に表示されているのが現在の車輪の速度で、数値が大きいほど速くなります。また、数値の設定範囲は0~255です。下のスライダーのつまみをマウスでドラッグしたり、「はやさ」の横の数字をキーボードで入力したりして速度を調整します。

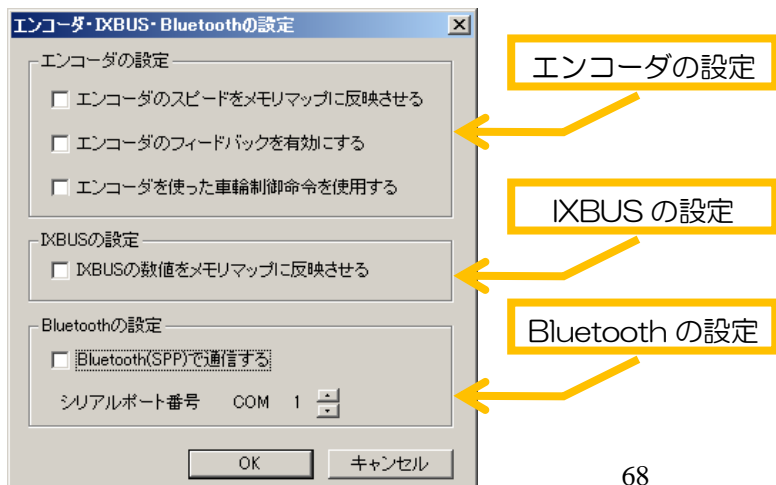


設定を変更したらダイアログ下の「適用」ボタンをクリックしてください。設定を適用すると、自動的にロボット本体にも設定が記録されます。なお、車輪の速度について、あまり数値を小さくすると車輪がまったく回転しなくなる場合がありますのでご注意ください。

3-2. エンコーダ・IXBUS・Bluetooth の設定

ロボットには、別売の各種オプションによって、IXBUS、ロータリーエンコーダ、Bluetoothなどの拡張機能が利用できます。これらを拡張した場合、本ソフトウェアから設定変更することでプログラムから機能を使えるようになります。

これらの設定を行う場合は、メニューの「設定」→「エンコーダ・IXBUS・Bluetooth の設定…」をクリックしてください。クリックすると以下のダイアログを表示します。



ロボットに接続していない拡張機器は設定しないでください。ロボットの動作がおかしくなる場合があります。

エンコーダの設定

ロータリーエンコーダはロボットの車輪の回転速度や位置を取得するセンサデバイスで、「電池の残量や地面の状態に関係なく車輪の速度を均一化する」「移動方法に時間ではなく距離や角度を使用する」などが可能になります。

ロボットにエンコーダを拡張していない場合は、エンコーダの設定を有効にしないでください。
エンコーダを使った車輪制御をつかう場合は、エンコーダの全ての設定をチェックしてください。

ビュートローバーARMでエンコーダをつかう場合、製品の仕様上 6ch 目のモータが使用できなくなりますのでご注意ください

エンコーダに関する各設定項目の説明は次のとおりです。

◆エンコーダのスピードをメモリマップに反映させる

エンコーダの現在の位置と速度がメモリマップに表示されるようになります。

◆エンコーダのフィードバックを有効にする

「前進」「旋回」などの移動のアクションブロックを実行する際に、モータが常に一定の回転数を保つようになり、移動性が向上します（「車輪制御」や「変数演算」の命令で車輪を動かした場合、フィードバックはかかりません）。

◆エンコーダを使った車輪制御命令を使用する

エンコーダを使った車輪制御の命令を使用できるようになります。また、この項目をチェックすると、エンコーダに関するほかの設定項目も自動的にチェックされます（エンコーダを使った車輪制御は、エンコーダの全ての設定をチェックしていないと正常に使えません）。

IXBUS の設定

IXBUS は、IXBUS (I2C) 通信を行う別売の拡張ボードに関する項目です。拡張ボードには「8ch アナログ入力」「2 軸ジャイロ・3 軸加速度センサ」などがあります。これらの情報は、設定を行うことでメモリマップから取得できるようになり、分岐や変数演算の命令でプログラムから使えます。

IXBUS に関する各設定項目の説明は次のとおりです。

◆IXBUS の数値をメモリマップに反映させる

メモリマップに IXBUS の 8ch の情報が表示されるようになります。

3-3. Bluetooth による通信の設定

Bluetooth をロボットに拡張すると、PC やスマートフォンなどを無線で通信できるようになります。これによって、本ソフトウェアから無線でプログラミング・遠隔操作したり、複数のロボットを群制御したりすることが出来ます。ここでは、本ソフトウェアから Bluetooth によって無線でロボットと通信する方法について説明します。

PC との接続・シリアルポート番号の確認

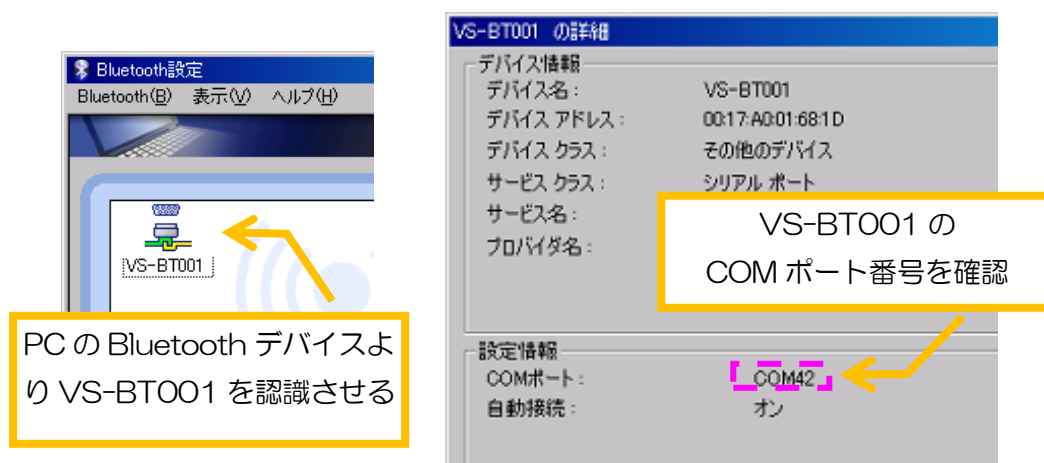
Bluetooth を使う場合、最初に PC の Bluetooth デバイスからロボットを認識させて接続しておく必要があります。また、通信に必要な「シリアルポート番号」という情報も確認が必要です。以下の手順でそれぞれ対応してください。

最初に、ロボット本体のファームウェア最新版を製品サポートページよりダウンロードして、ファームウェアの書き換えを行ってください。また、ビュートローバーH8 をお使いの方は Bluetooth と通信可能な方のファームウェアを書き込んでください。ファームウェアの書き換えの方法については「3-4.ファームウェアの書き換え」をご参照ください。

◆ビュートローバー 製品サポートページ (ダウンロード)

http://www.vstone.co.jp/products/beauto_rover/download.html

お使いの PC の Bluetooth 通信デバイスより VS-BT001 を認識させて、通信を開始してください。また、併せて COM ポート番号をご確認ください (PC 側の Bluetooth 通信デバイスの操作方法につきましては、各デバイスの取扱説明をご確認下さい)。

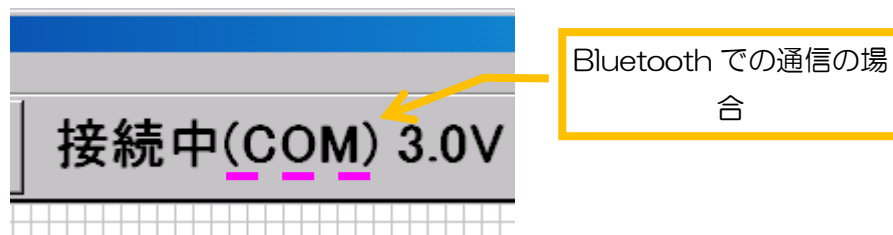


通信方法の変更・シリアルポート番号の設定

BluetoothでPCからロボットに接続し、シリアルポート番号を確認したら、本ソフトウェアの設定を行います。ダイアログの「Bluetoothの設定」の項目で、それぞれ以下のように設定してください。



設定を適用すると、しばらくしてロボットが Bluetooth で通信を開始します。Bluetooth で通信している場合、ウィンドウ右上の通信中を現す表記が「通信中(COM)」となります。

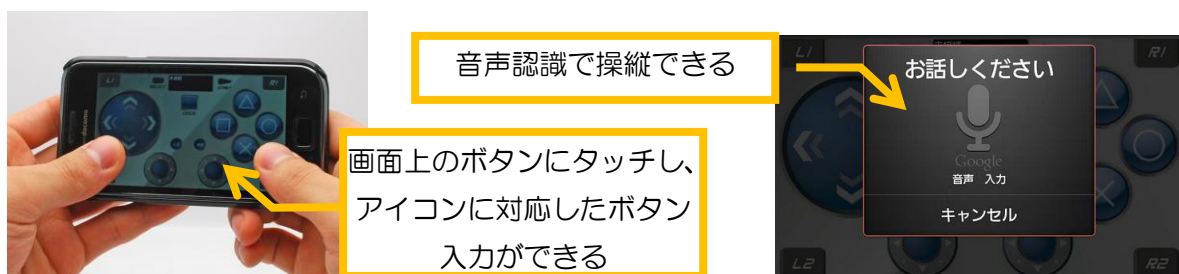


なお、Bluetooth と通信する設定にしている場合でも、USB ケーブルでロボットを接続している場合、そちらが優先的に認識されるので、Bluetooth で通信させるロボットは USB で接続せずにお使いください。

VS-C2 for Android について

ロボットに Bluetooth 機能を拡張することで、スマートフォンなど Android OS を搭載した端末からロボットを操縦できるようになります。[VS-BT001 の製品ページ](#)では、Android 端末対応から Bluetooth でロボットを操縦するソフトウェア「VS-C2 for Android」を公開しています。

VS-C2 for Android は、ゲームパッド「VS-C1」とほぼ同じ操作が出来るようになっています。ソフトウェア上でロボットと Android 端末を Bluetooth で接続し画面上のボタンを押すと、アイコンに対応するゲームパッドのボタンを押したのと同じ操作が出来ます。また、音声認識による操縦にも対応しています。



VS-C2 for Android の導入やその他詳細は、Android マーケットの VS-C2 for Android のページ、及び VS-C2 for Android のヘルプをご参照ください。

3-4. ファームウェアの書き換え

ロボットには、PC と通信したり記録しているプログラムを単独で実行したりするための専用の動作プログラムが書き込まれています。このプログラムを「ファームウェア」といいます。出荷時のロボットには、本ソフトウェア用のファームウェアがあらかじめ書き込まれていますが、今後製品の機能の追加や問題点の解消などを目的にファームウェアが変更されることがあります。また、C 言語で作成したプログラムをロボットに書き込んだ場合、再度本ソフトウェアで使用するためには本ソフトウェア用のファームウェアを書き直す必要があります。

このような場合、製品サポートから新しいファームウェアのファイルをダウンロードして、本ソフトウェアよりお使いのロボットの CPU ボードに書き込むことができます。

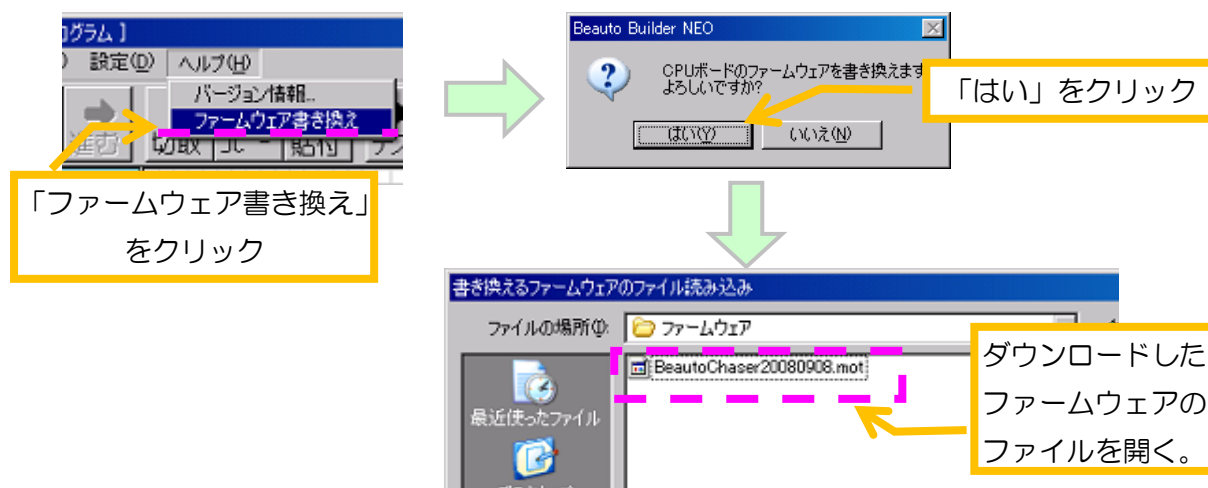
◆ビュートローバー 製品サポートページ (ダウンロード)

http://www.vstone.co.jp/products/beauto_rover/download.html

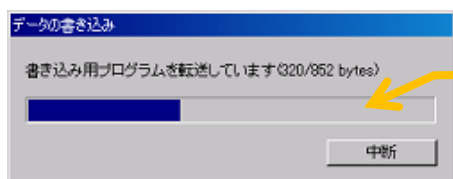
なお、ビュートローバーとビュートローバーARM・ビュートミニ ARM ではファームウェアの書き込み手順が異なるため、それぞれの方法を個別に説明します。

ビュートローバーへのファームウェア書き込み

ビュートローバーをお使いの方で、サポートページからダウンロードしたファームウェアを CPU ボードに書き込む場合は、下記の手順で行ないます。まず、ロボットを PC に接続してください。続いて、本ソフトウェアを起動してメニューより「ヘルプ」→「ファームウェア書き換え」をクリックしてください。クリックすると「CPU ボードのファームウェアを書き換えます。よろしいですか?」というメッセージを表示するので、書き換えを行なう場合は「はい」をクリックしてください。「はい」をクリックすると、更にファームウェアのファイルを選択するダイアログを開くので、ダウンロードしたファームウェアのファイルをクリックして開いてください。

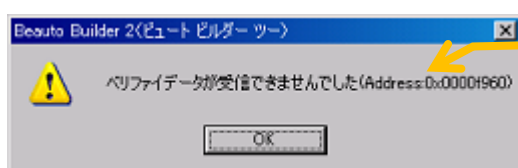


ファームウェアの書き換えを開始します。書き換えの進行状況はダイアログに表示され、書き換えが終了すると終了のメッセージを画面に表示します。なお、書き換えがまったく進まない場合は、一度「中断」ボタンをクリックして書き換えを中断してください。



書き込みが完了するまで待つ。まったく書き込みが進まない場合は「中断」をクリックして書き込みを中断する

書き換えを完了するメッセージが表示されたらファームウェアの書き込みは完了です。ファームウェアの書き換えが正しく行われなかった場合、下記のように問題の概要を警告し、書き込みを中断します。この場合、「ロボットとPCがしっかり接続されているか」「他に本ソフトウェアを起動していないか」などを確認して、一度通信ケーブルを抜き差しした後、もう一度書き込み操作を行ってください。



書き込みに失敗した場合、もう一度最初から書き込む

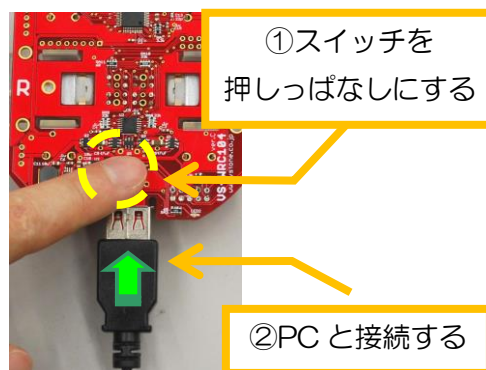
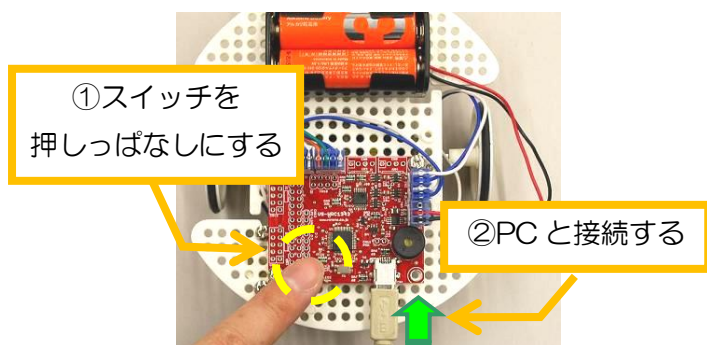
また、ファームウェアの書き込みに失敗したり、途中で書き込みを中断した場合、ロボットのCPUボードに正しくプログラムが書き込まれていない状態になります。そのままでは本ソフトウェアでロボットが使えないため、改めてファームウェアの書き込みを行なってください。

ビュートローバーARM・ビュートミニ ARM へのファームウェア書き込み

ビュートローバーARM・ビュートミニ ARM へファームウェアを書き込む場合は、本ソフトウェアを使わずに行います。まず、本ソフトウェアを終了してください。終了したら、ロボットの電源を切って一度PCから取り外してください。取り外したら、ロボットのスイッチを押しながらもう一度PCに接続してください。

ビュートローバーARM

ビュートミニ ARM



接続して数秒待つと、PC がロボットをマスタストレージデバイス（USB メモリのような新しいドライブ）として認識します。PC のマイコンピュータを開いて、「CRP DISABLE」という名前のドライブが現れるのを待ってください。



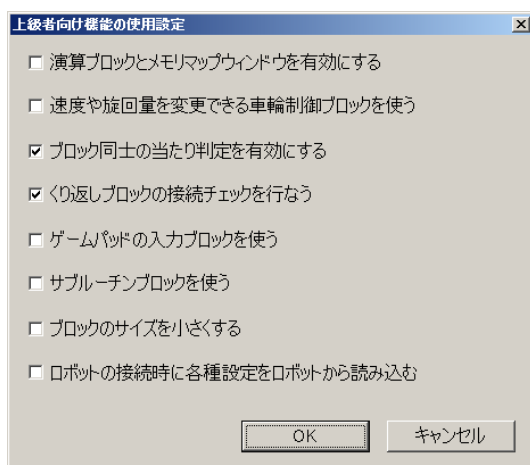
ドライブが現れたら PC から開いてください。ドライブを開くと、中に「firmware.bin」というファイルが入っています。これがロボットに書き込まれているファームウェアです。ファームウェアを書き換える場合、「firmware.bin」を削除して、ダウンロードした新しいファームウェアのファイルをコピーしてください。

コピーしたら、ロボットを PC から取り外して再起動させてください。また、最後にロボットに書き込んだプログラムやモータの設定などは消えているので、改めて本ソフトウェアより書き直してください。

3-5. 上級者向け機能の設定

本ソフトウェアでは、変数を利用した演算やセンサ以外を利用した条件分岐など、プログラミング上級者用の機能が備わっています。これらの機能を使いこなすことでより複雑なプログラミングが可能ですが、扱うために必要となる知識も多いため、初期の設定ではこの機能を利用できないようになっています。

上級者向け機能は、メニューの「設定」→「上級者向け機能の設定」をクリックして開くダイアログより行ないます。ダイアログには複数のチェック項目があり、各項目のチェックの有無で機能の有効・無効を切り替えます。



各項目のチェックを変更することで、
上級者向けの機能を利用できる

◆演算ブロックとメモリマップウィンドウを有効にする

変数演算のアクションブロックを使用するか、また、ロボットとの接続中に、画面にロボットのメモリマップの現在値を一覧で表示する「メモリマップウィンドウ」を開くかどうかの設定です。また、この機能を有効にすると、条件分岐の命令も設定エリアの内容が変化します。これらの命令に関する詳細は「2-2-9.変数演算のアクションブロック」及び「3-6. メモリマップの表示」をご参照ください。

◆速度や旋回量を変更できる車輪制御ブロックを使う

左右の車輪の速度を、別々に自由な速度で動かせる命令を使用するかどうかを設定します。この命令自体に関する詳細は「車輪制御のアクションブロック」をご参照ください。

◆ブロック同士のあたり判定を有効にする

プログラムエリアのアクションブロック同士が重なって表示されないようにするための設定です。この項目にチェックを入れると、アクションブロックをマウスでドラッグ中に他のアクションブロックと重なっているときはブロックの周囲のみを点線で表示し、また、アクションブロック同士が重なった状態でドラッグを終えると、ブロックが重ならない位置まではじかれます。

◆繰り返しブロックの接続チェックを行なう

繰り返しのアクションブロックの接続が正しくない場合に警告を表示するかどうかの設定です。この項目については、初期状態でチェックがついています。この項目にチェックをつけると、プログラムをテスト実行・ロボットに書き込みする際に繰り返しのアクションブロックの接続についてチェックし、接続が正しくない場合は警告を表示します。

◆ゲームパッドの入力ブロックを使う / ◆サブルーチンブロックを使う

それぞれ、ゲームパッド用の分岐のアクションブロック、及びサブルーチンのアクションブロックを使用するかどうかの設定です。これらの命令に関する詳細は「サブルーチンのアクションブロック」及び「ゲームパッド分岐のアクションブロック」をご参照ください。

◆ブロックのサイズを小さくする

プログラムエリアの各アクションブロックのサイズを全体的に縮小表示するかどうかを選択します。この項目にチェックを入れると、アクションブロックのサイズが50%ほどに縮小し、画面を広く使うことができます。

◆ロボットの接続時に各種設定をロボットから読み込む

本ソフトウェアとロボットが通信を始めたときに、モータの速度や IxBUS・エンコーダ・Bluetooth に関する設定（シリアルポート番号は除く）をロボットの中から読み出すか否かを設定します。1 台の PC で複数のロボットを使い分ける場合などに、チェックを入れることで設定を自動的に切り替えることができます。

3-6. メモリマップの表示

ロボットと PC が通信中の場合、本ソフトウェアのセンサエリアに現在のセンサの数値が表示されますが、CPU ボードからはセンサ・ボタンの数値以外にも様々な数値を取得することができます。これらの数値は、「メモリマップウィンドウ」を表示させることでセンサエリアのように数値を表示させられます。また、ロボットから取得してメモリマップウィンドウに表示されるこれらの数値をまとめて「変数」といいます。

メモリマップウィンドウを表示させる場合は、上級者向け機能の設定で「メモリマップの内容を表示する」を有効にすると自動的に表示します。また、メモリマップウィンドウを一度閉じた後に再度表示させる場合は、メニューより「設定」→「メモリマップの表示」をクリックします。



説明	数値	サイズ	R/W
Version	0	SWORD	R/W
右モータゲイン	0	UBYTE	R
左モータゲイン	0	UBYTE	R
モータ3ゲイン	0	UBYTE	R
モータ4ゲイン	0	UBYTE	R
モータ5ゲイン	0	UBYTE	R
モータ6ゲイン	0	UBYTE	R
右モータ速度	0	SWORD	R/W
左モータ速度	0	SWORD	R/W
モータ3速度	0	SWORD	R/W
モータ4速度	0	SWORD	R/W
モータ5速度	0	SWORD	R/W
モータ6速度	0	SWORD	R/W
センサ1	0	SWORD	R
センサ2	0	SWORD	R
センサ3	0	SWORD	R
センサ4	0	SWORD	R
センサ5	0	SWORD	R
センサ6	0	SWORD	R
センサ7	0	SWORD	R
センサ8	0	SWORD	R
ボタン	0	UWORD	R
LED(緑)	0	SWORD	R/W
LED(オレンジ)	0	SWORD	R/W
アザー音量	0	UWORD	R/W
アザーピッチ	0	UWORD	R/W
アザータイム	0	UWORD	R/W
リモコンボタン入力1	0	UWORD	R
リモコン右アナログX	0	SWORD	R
リモコン右アナログY	0	SWORD	R
リモコン左アナログX	0	SWORD	R
リモコン左アナログY	0	SWORD	R
アザーカウンタ	0	UWORD	R

メモリマップウィンドウは、
ロボットと通信している場合に
ロボットの変数の現在値を
確認できる

メモリマップウィンドウには、表形式でメモリマップの名称・数値・及びその他概要が1行ずつ表示されています。それぞれの列の概要は以下の通りです。

• 説明 / 数値

変数の名前と現在の数値です。注意として、「数値」の項目は、ロボットと通信しないと最新の状態に更新されません。

• サイズ

変数が扱うことのできる数値の範囲を表します。この項目は、変数の種類によって「SBYTE」「UBYTE」「SWORD」「UWORD」の4つに変わります。

「SBYTE」「UBYTE」は、1byteの変数であることを表します。また、頭文字の「S」「U」は、マイナスの数値を扱えるか否かを表します。扱える数値の範囲は、「SBYTE」は-128~+127、「UBYTE」は0~255です。

「SWORD」「UWORD」は、2byteの変数であることを表します。また、頭文字の「S」「U」は、マイナスの数値を扱えるか否かを表します。扱える数値の範囲は、「SWORD」は-32768~+32767、「UWORD」は0~65535です。

• R/W

変数に対してデータの読み込み・及び書き込みができるかを表しています。この項目の表記は「R」「R/W」の2種類です。「R」は、その変数がデータの読み込みだけ出来ることを表します。センサや IXBUS などがこれに相当します。分岐や変数演算などの命令でプログラム中から扱う場合、この変数に値を代入することは出来ません。「R/W」は、その変数がデータの読み書きが出来ることを表します。「R」の場合と異なり、変数演算の命令から変数に数値を代入することが出来ます。

変数演算や分岐の命令でプログラムから変数を使う場合、「サイズ」及び「R/W」を良く確認してください。例えば「サイズ」の表記が異なる二つの変数を計算しようとする、変数に代入される値が想定外のものに変化してしまい、ロボットが意図とは異なる動作をする可能性があります。具体的には、「SWORD」のユーザ変数に 40000 を代入しようとする、数値が他の値に変わります。また、LED やブザータイムなど、代入した値が自動的に変化するものもありますのでご注意ください。

なお、モータゲインなど一部の変数は、ソフトウェアの内部設定によりプログラム中で扱うことが出来ないようになっています。

4. FAQ

本ソフトウェアを使用して何か問題が発生した場合は、下記項目をご参照いただき、原因と対処方法をご確認ください。記載されている対処を行っても状況が改善しない、または以下に当てはまる現象が見つからない場合は、お手数ですが末尾の連絡先までお問い合わせください。

Q:ロボット本体の車輪がまわらない。

- A1:ロボットの電池が少ない可能性があります。車輪の回転が弱々しい場合はこれが原因である可能性が高いです。電池を新しいものと交換して改善するかご確認ください。
- A2:ロボットの電源スイッチが OFF になっていることが考えられます。電源スイッチが ON になっているかご確認ください。なお、ロボットを PC と接続している場合は、PC の USB ポートから供給される電力だけで動作している場合があります。この場合、電力不足により車輪の回転が弱い場合があります。
- A3:ピュートミニ ARM をお使いの場合、モータの端子が基板に接触できていない可能性があります。モータ端子の曲げが弱く基板との間に隙間が出来ていないか、または、端子を曲げすぎて基板の反対側から飛び出していないかご確認ください。
- A4:ロボット本体の設定で、車輪の速度が非常に遅い設定になっている可能性が考えられます。「3.ロボットの詳細設定」の項目を参考に、車輪の速度を速く設定してください。
- A5:移動のアクションブロックで、車輪を回す時間を非常に短く設定しているため、車輪が十分加速する前に停止してしまっている可能性があります。時間の指定を長めに設定しなおしてください。
- A6:片方の車輪が何かに引っかかったりするなどして強い負荷がかかっている場合、ロボット全体の電力が不足して車輪が回らなくなる場合があります。車輪が引っかかっている場所などが無いかご確認ください。
- A7:モータ配線を CPU ボードに逆方向に接続していないか、また、途中でモータの配線が断線していないかご確認ください。

Q:プログラムを実行していないのに、勝手にモータが動く

- A:作成したプログラムで、最後に車輪軸にブレーキを与えずに終了した場合、最後の車輪速度の設定が反映されたままになるため、その状態でモータの電源を ON にすると勝手に車輪が回転するなどの現象が発生します。プログラムの最後にモータを停止させるアクションを追加してください。

Q:ロボット本体の動作中に、頻繁に電源が落ちたりリセットがかかったりする

- A1:ロボット本体の電池が少ない可能性があります。新しい電池に交換して状況が改善するかご確認ください。
- A2: ロボットの車輪に無理な力がかかっており、全体の電力が不足している可能性があります。車輪が引っかかっているなどの問題が無いかご確認ください。

Q:ロボット本体から煙・火花・異臭などが発生した

- A1:ロボットの配線がショートしている可能性があります。すぐに電源スイッチを OFF にして通信ケーブルを取り外し、電池を抜いた上で、末尾の宛先までご連絡ください。

Q:プログラムの書き込みの際に「プログラムのサイズがオーバーしました」というエラーが表示されプログラムを書き込むことができない

A1:作成したプログラムがプログラムに書き込めるデータのサイズを越えた場合、ロボット本体にプログラムを書き込むことができません。プログラム中に存在する矢印の接続されていないアクションブロックもロボット本体にデータが書き込まれるので、不要なブロックは削除して、プログラムを書き込めるサイズまで容量を削減してください。

A2:お使いの BeautoBuilder2 及びファームウェアのバージョンを最新のものに書き換えると、プログラムを記録できる容量が増えて改善される場合があります。それぞれのデータを製品サポートページからダウンロードしてお試しください。

Q:センサエリアにセンサの情報が正しく表示されない

A1:センサのコネクタが CPU ボードから抜けていたり、途中で断線していないかご確認ください

A2:センサが CPU ボード上の正しい位置に接続されているかご確認ください。

A3: CPU ボードに C 言語で作成したプログラムを書きこんだ場合、出荷時のプログラムに書き直さないとビューアビルダー2 で通信できません。「[3-4. ファームウェアの書き換え](#)」の手順に従い、CPU ボードのファームウェアを出荷時のものに書き換えてください。

Q:ビュートローバーARM で書きこめるプログラムがビュートローバーでは書き込めない

A1:ビュートローバーとビュートローバーARM では、ロボット本体に記録できるプログラムのデータサイズが異なります。ビュートローバーARM の方が書きこめるサイズが大きいので、大きいプログラムの場合はビュートローバーでは書き込めない場合があります。

Q:VS-BT001 で Bluetooth による通信ができない

A1:お使いの PC の Bluetooth デバイスが VS-BT001 を正しく認識できているかご確認ください。

A2:VS-BT001 の COM ポート番号が正しいかご確認ください。お使いの Bluetooth デバイスによっては、以前認識していた COM ポートと別の番号が割り当てられる場合があります。

A3:ビュートローバーARM をお使いの場合は、一度 USB ケーブルで接続して、モータの設定ダイアログで「Bluetooth (SPP) で通信する」にチェックを入れた状態で「適用」をクリックし、CPU ボードに通信の設定を反映させてください。

A4:ビュートローバーH8 をお使いの場合は、COM ポートによる通信を行うことができるファームウェアが書き込まれているかご確認ください。

Q:ライトレースのプログラムを実行すると、プログラムは正しいのにロボットがコースを外れてしまう

A1:ロボットのモータのスピードが速すぎると、カーブを曲がりきれずロボットがコースアウトすることがあります。モータの設定でモータのスピードを左右両方とも遅めに設定変更してください。

A2:コース上に直射日光が当たっていると、赤外線センサが正しく反応できずコースアウトします。ライトレースを行う際は直射日光が当たらない場所でロボットを動かしてください。

◆ビュートローバー 製品サポートページ:

http://www.vstone.co.jp/products/beauto_rover/

◆ビュートローバー 製品サポートページ (ダウンロード):

http://www.vstone.co.jp/products/beauto_rover/download.html

◆ビュートローバー 製品サポートページ (Q&A):

http://www.vstone.co.jp/products/beauto_rover/qa.html

◆ビュートミニ ARM 製品サポートページ:

http://www.vstone.co.jp/products/beauto_mini_arm/

◆ビュートミニ ARM 製品サポートページ (ダウンロード):

http://www.vstone.co.jp/products/beauto_mini_arm/download.html

◆ビュートミニ ARM 製品サポートページ (Q&A):

http://www.vstone.co.jp/products/beauto_mini_arm/qa.html

■オプションパーツ、関連商品のご購入は・・・

No.1 の品揃え! 各種オプションパーツ、ロボット関連製品のご購入はコチラ

<http://www.vstone.co.jp/robotshop/>

楽天・Amazon・Yahoo の各 Web 店舗、または東京、福岡の各ロボットセンター店頭でもロボット関連商品をお買い求めいただけます。

ロボットセンター東京秋葉原店 (東京支店)
〒101-0021
東京都千代田区外神田 1-9-9 内田ビル 4F

ロボットセンターロボスクエア店 (福岡支店)
〒814-0001
福岡市早良区百道浜 2-3-2
TNC 放送会館 2F ロボスクエア内

商品に関するお問い合わせ

商品の技術的なご質問は、問題・症状・ご使用の環境などを記載の上メールにてお問い合わせください。

E-mail: infodesk@vstone.co.jp

受付時間 : 10:00~17:00 (土日祝日は除く)

ヴイストーン株式会社

〒555-0012 大阪市西淀川区御幣島 2-15-28
TEL : 06-4808-8701 FAX : 06-4808-8702

Vstone[®]
www.vstone.co.jp