

情報とコンピュータ
プログラムによる計測と制御

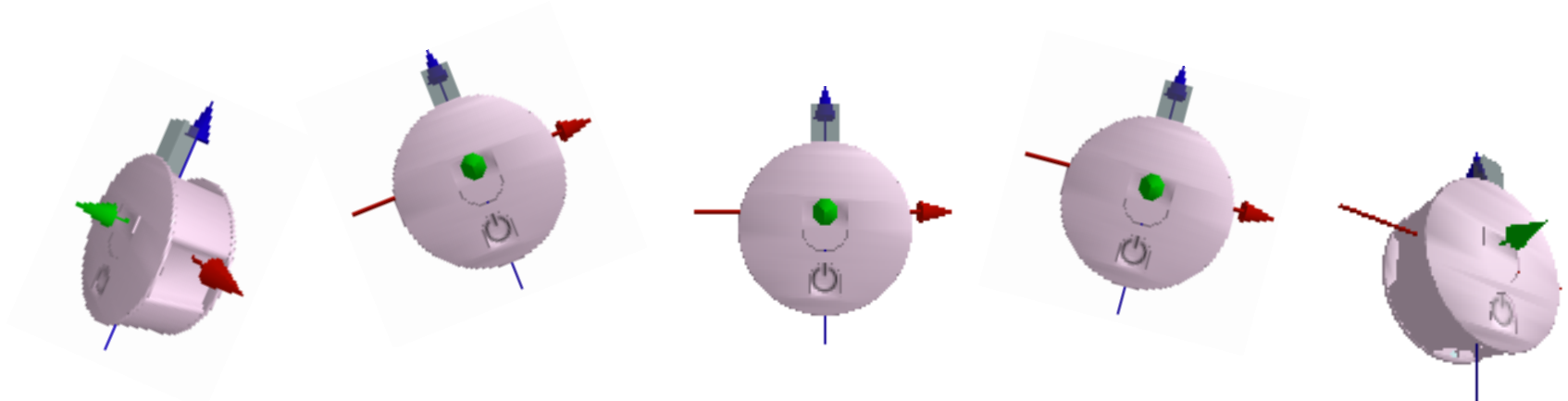
プログラミング学習教材

加速度センサプログラマー



1時間目

- 「コンピュータ」と「制御」の説明
- センサの反応の確認
- ソフトウェアの操作方法の説明
- 順次処理の課題に挑戦

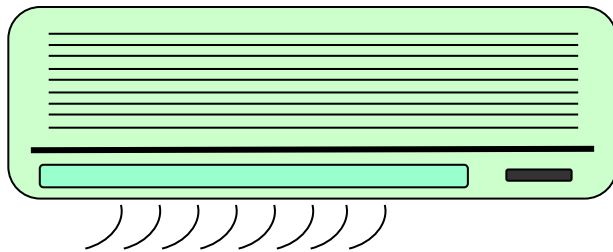


「コンピュータ」と「制御」の説明

そもそも、「コンピュータ」とは・・・？

「①入力された情報から、②状況を判断して、何かを③出力する仕事ができるもの」を言います。

例：エアコン



①部屋の室温を感知(計測)

②コンピュータで判断

③エアコンの働きを制御

それぞれの要素を、別の言葉に置き換えると...

エアコンの場合...

部屋の温度を感知
(①入力)



設定よりも暑いので
冷房を強めよう
(②判断・命令)



冷風を出す(強める)
(③出力)

①入 力 = センサ

状況を感じ取るもの

→温度センサ、光センサなど

②判断・命令 = コンピュータ

状況を判断し、命令を行うもの

③出 力 = アクチュエータ

動作・仕事を行うもの

→クーラー、ヒーター、モーター
ブザー、LEDなど

どんなものに「コンピュータ」が使われている？

センサ・コンピュータ・アクチュエータを持った機械を挙げてみましょう

- 自動ドア

- 人を感知し、扉の開閉を行ないます。

- デジタルカメラ

- センサで手ブレを補正したり、ピントを合わせたり自動的にきれいな画像を撮影できます。

他に「自動車」「炊飯器」「冷蔵庫」「自動改札」など
→実生活にはコンピュータ制御機器があふれています。

本教材にもコンピュータが使われています。



加速度センサプログラマー

- ① **センサ** = 加速度センサ
「傾ける」「振る」などの動きを調べます。
- ② **コンピュータ** = PICマイコン
命令を覚えたり、行動を考えます。
- ③ **アクチュエータ** = ブザー、LED
音を出します。点灯・点滅します。

本体を動かす方法は二通りあります。

・PCと接続して動かす方法

電池不要。

・本体単体で動かす方法

電池が必要です。

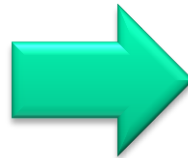


単体で動かす場合は、次のスライド(2ページ分)を見て、動作確認プログラムを動かしてみましょう。

単体でプログラムを実行する場合、電池が必要です。

使用できる電池: **単4アルカリ乾電池**

または単4ニッケル水素充電電池



① プラスドライバー(#2)で
ねじを緩め、カバーを開けます。

② 向きを確認して電池を入れ、
カバーを戻します。

電源ボタンをONにして操作ボタンを押すと、プログラムが開始します。

(USBケーブルで接続している場合、本体からUSBケーブルを抜いてください)

②操作ボタン
を押す



①電源ボタン
を押す

購入時の本体には動作確認用プログラムが記録されています。

傾けた場合：

→常に上方向のLEDが光ります

操作ボタンを押した場合：

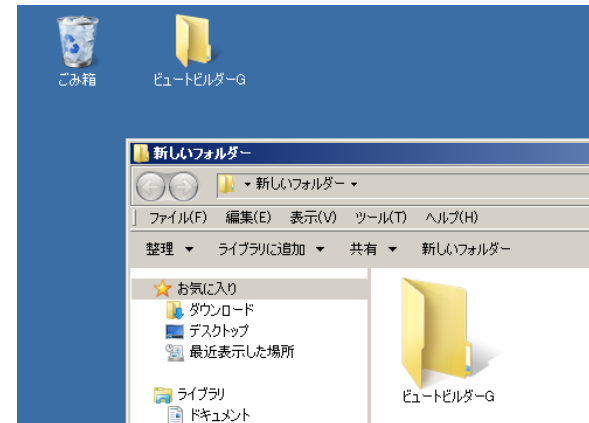
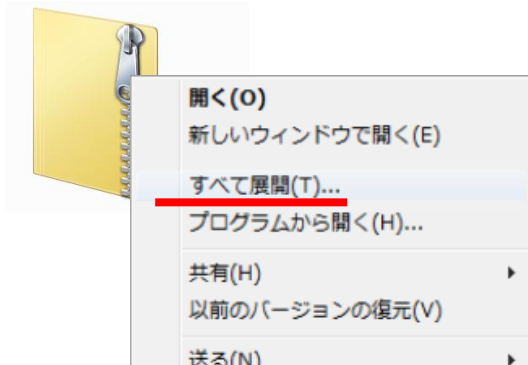
→ブザーが鳴ります。傾きに応じて音の高さが変わります

すぐに電池がなくならないよう、使用しない場合は電源を切りましょう。

ソフトウェア「ブートビルダーG」の導入

学校でブートビルダーGを使うときは、「ZIP圧縮版」を使います。

1. ブートビルダーG (ZIP圧縮版) をダウンロード
2. 右クリックし「すべて展開」をクリック
3. 展開してできたフォルダを、教室の全PCにコピー

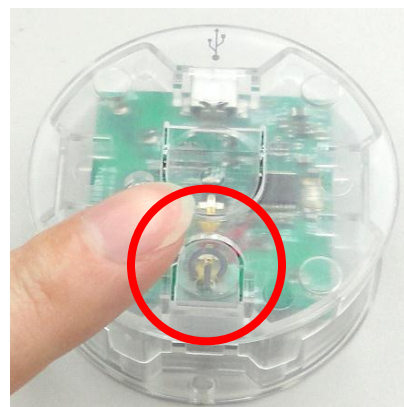


フォルダの中にあるcl_edit_acc.exeをクリックしてソフトを起動します。

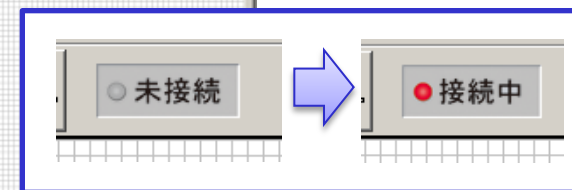
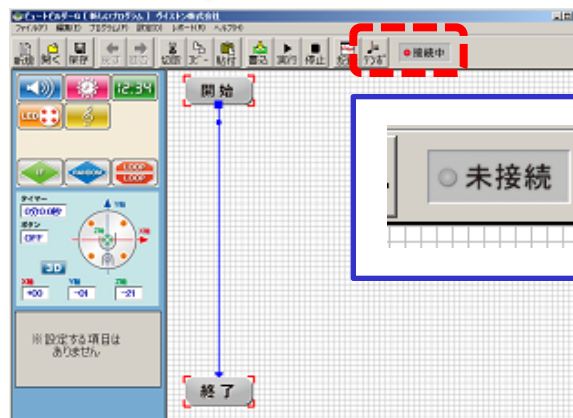
パソコンと本体の接続



①付属のUSBケーブルをPCと本体に接続します。

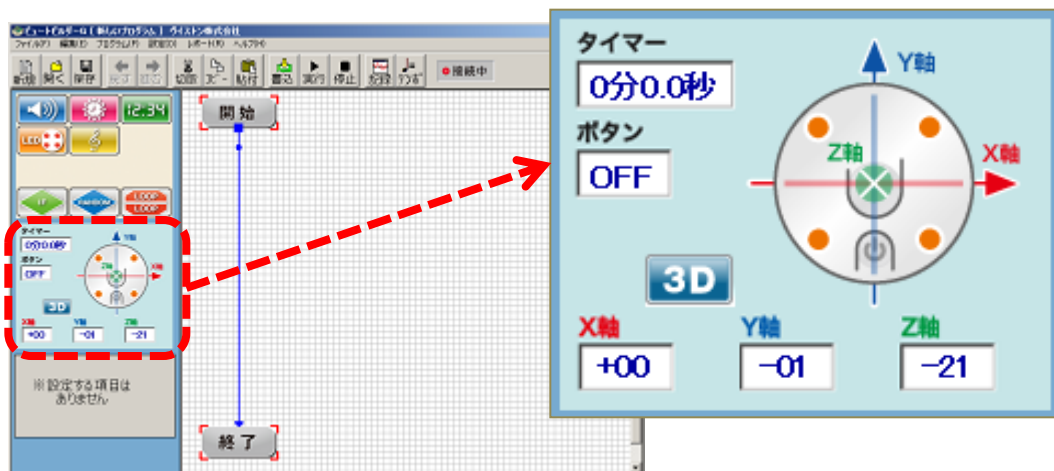


②本体の電源ボタンを押します。

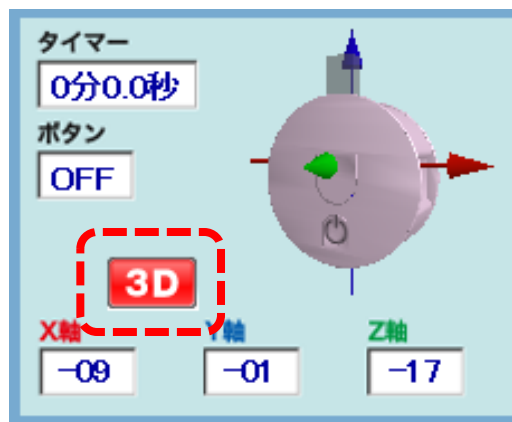


③右上の表示が「未接続」から「接続中」に変わります。

「センサエリア」でセンサの反応を確認しましょう。



加速度センサの数値や
操作ボタンのON/OFF
を表示



「3D」ボタンをクリックすると、
本体に合わせて動く3DCG
を表示します。

必ずケーブルを奥向き
にして持ってください



これから、制御を行なうためのプログラミングを学習していきます。
プログラミングは専用ソフトウェア「ビュートビルダーG」を用い、
命令のブロックを並べることで行ないます。

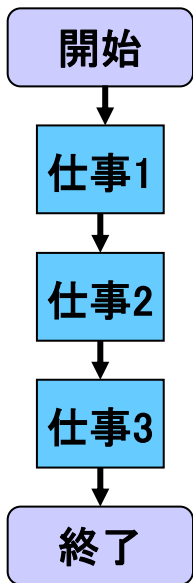
The screenshot shows the Vstone software interface with a flowchart on a grid background. The flowchart starts with a '開始' (Start) block, followed by a 'LOOP' block. An 'IF' decision block contains the condition 'X軸 < +0'. If the condition is true (true), the flow goes to an 'LED' block. If false (false), it goes to another 'LED' block. Both LED blocks then lead to an '無限' (Infinite) loop block, which loops back to the start of the 'LOOP' block. The flowchart ends with a '終了' (End) block. The software interface includes a menu bar, a toolbar with icons for file operations and execution, and a left sidebar with various control blocks like 'IF', 'RANDOM', and 'LOOP'. A 3D coordinate system is also visible in the sidebar.

これは、左右傾けた方の
LEDを光らせるプログラム
です。

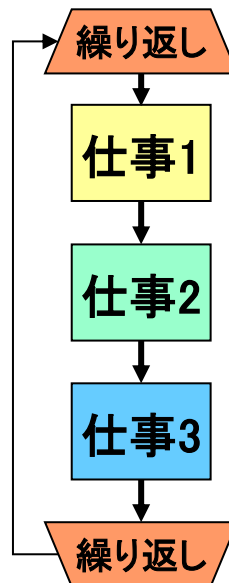
プログラムは、次の三つの構造でできています。

- ① 順次
- ② 繰り返し
- ③ 分岐

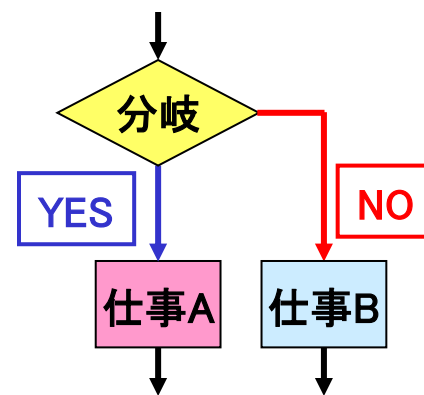
【順次】

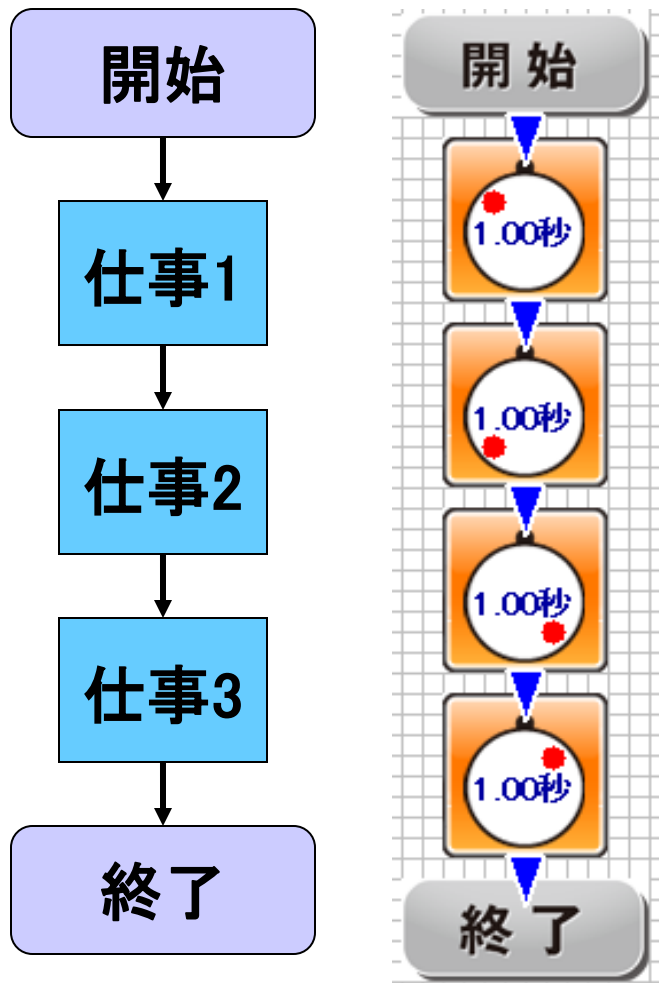


【繰り返し】



【分岐】





仕事を順番に並べ、一つずつ実行するようなプログラムを「**順次**」のプログラムといいます。

プログラムの構造の中では、もっとも単純なものです。

順次のプログラミング 「ドレミ」と鳴らす

The screenshot shows the Vstone software interface for programming a sequence. The window title is "ビュートビルダー-G [新しいプログラム] ヲイストーン株式会社". The menu bar includes "ファイル(F)", "編集(E)", "プログラム(P)", "設定(D)", "レポート(R)", and "ヘルプ(H)". The toolbar contains icons for "新規", "開く", "保存", "戻す", "進む", "切取", "コピー", "貼付", "書込", "実行", "停止", "記録", and "テンポ". A "未接続" (Disconnected) button is also present.

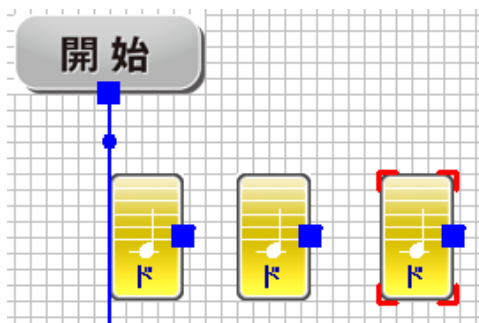
The main workspace is a grid with a flowchart for a sequence. It starts with a "開始" (Start) button, followed by three yellow boxes representing notes: "ド" (Do), "レ" (Re), and "ミ" (Mi). The "ミ" box is highlighted with a red border and contains three dots, indicating it is the current step. The flowchart ends with a "終了" (End) button.

The left sidebar contains various controls:

- Speaker icon, clock icon, and a digital display showing "12:34".
- LED icon and a treble clef icon.
- Control buttons: "IF", "RANDOM", and "LOOP LOOP".
- Timer section: "タイマー" with a "???" input field, "ボタン" with a "???" input field, and a 3D coordinate system with "X軸", "Y軸", and "Z軸" axes, each with a "???" input field.
- Dropdown menu: "普通の" (Normal).
- Dropdown menu: "4分 音符" (4-measure note).
- Checkbox: "鳴り終わるまで待つ" (Wait until finished).

使用するブロック(命令)の追加

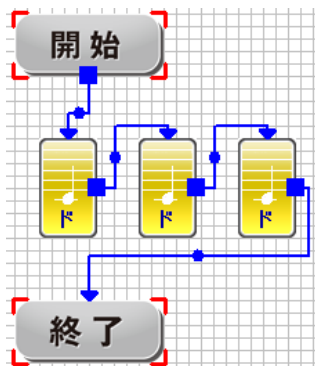
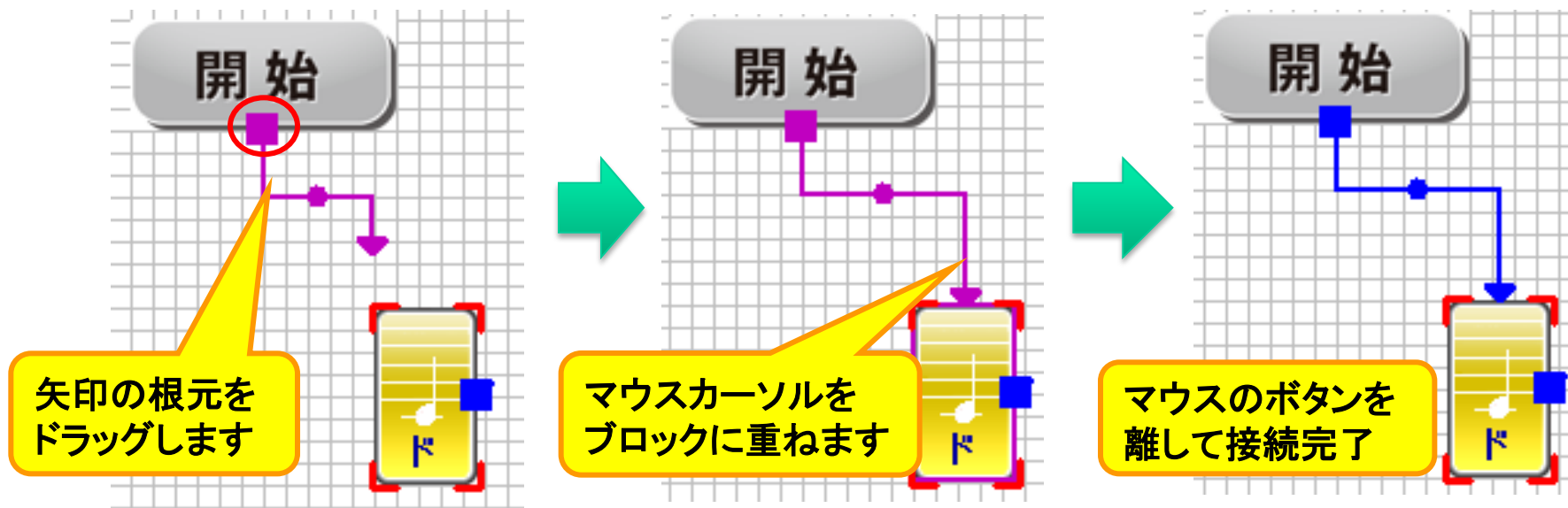
最初に、プログラムで使うブロックを選びます。



同じブロックを三つ作りましょう。

ブロック(命令)の実行順序の設定

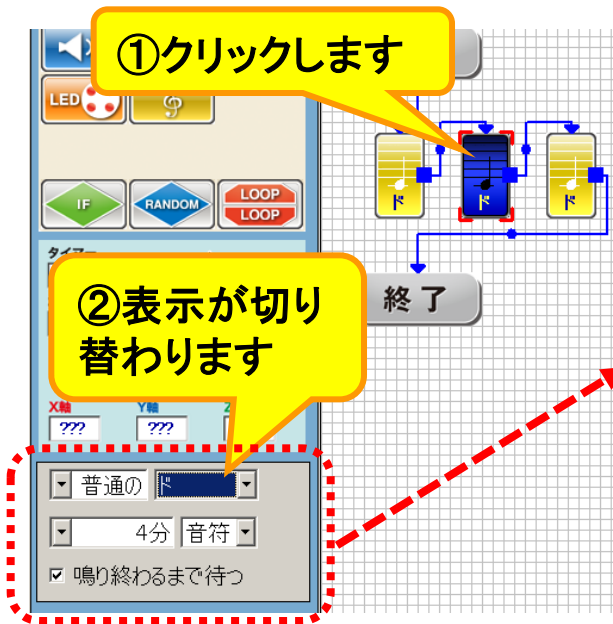
続いて、追加したブロックを矢印でつなぎます。



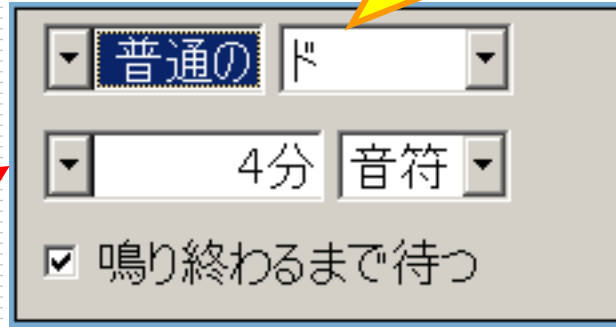
全てのブロックに矢印をつないでみましょう。

ブロック(命令)の設定の変更

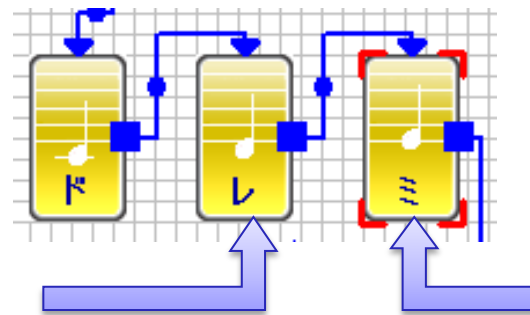
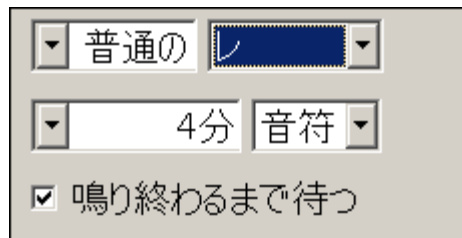
最後に、それぞれの音程を変更します。



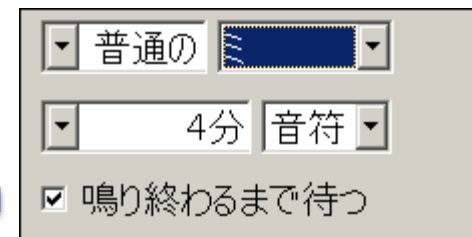
ブロックごとに命令の設定を
変更することができます



ブロックをクリックすると、
左下の設定エリアで内容
を変えられます。



2、3番目のブロックを
「レ」「ミ」の音に変えましょう。



プログラムの書き込み・実行①

プログラムができたので実行してみましょう。

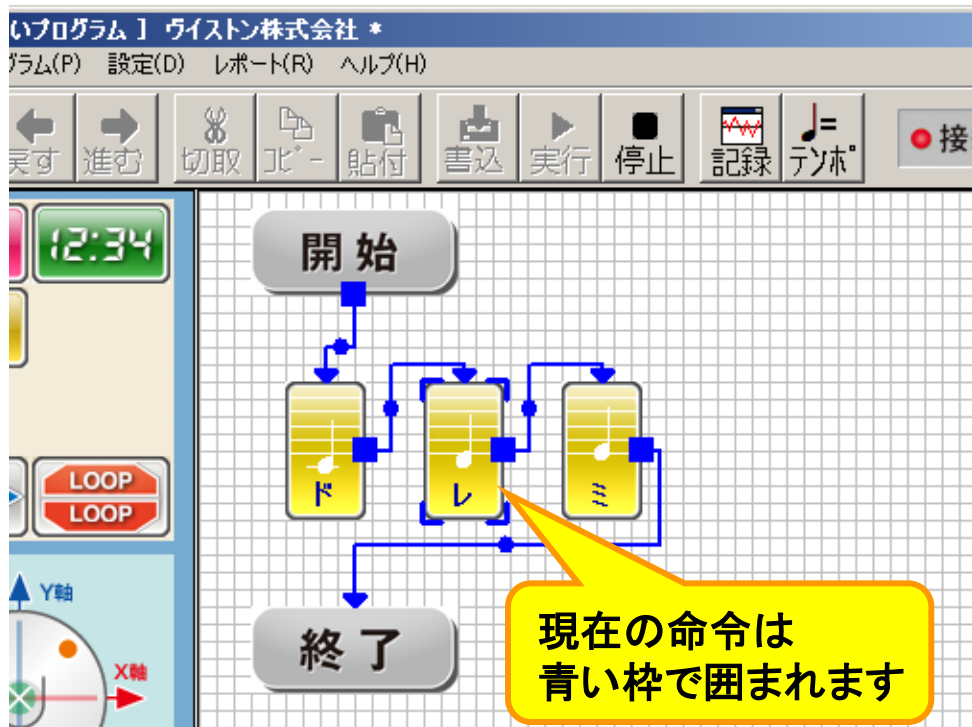
「実行」ボタンをクリック

「はい」をクリック

書き込みが終わるまで待ちます

青いメーターが一瞬で右端まで届き、画面が消えたら書き込み終了です。
※書き込みに10秒以上かかる場合は失敗しているので、「中断」をクリックし、一度ケーブルを抜き差しして書き込みなおしてください。

書き込みが終わると、自動的にプログラムを実行します。



プログラムの実行中は、青い枠で実行中のブロックが囲まれます。

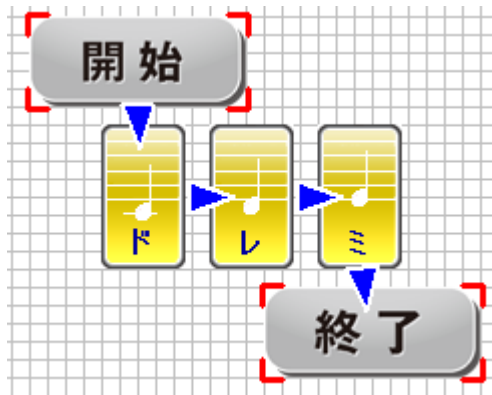
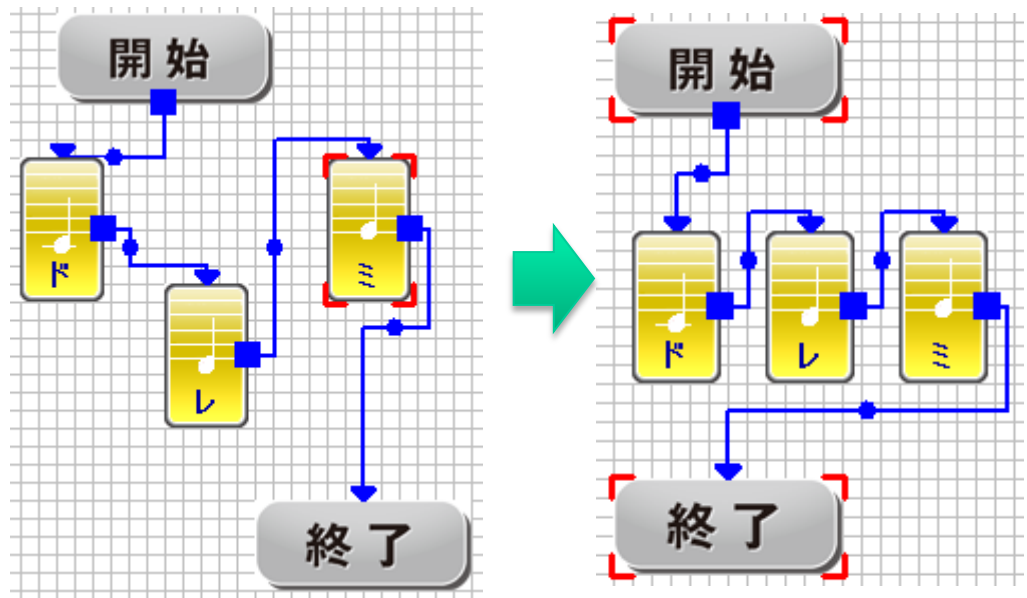
青い枠の流れを見て、思った通りのプログラムが実行されているか確認しましょう。

一度実行したプログラムは、「単独で動かす」(8~9ページ)の手順で、単体でも実行できます。

※本体にプログラムを書き込むと、既にかき込まれていたプログラムが消えます。

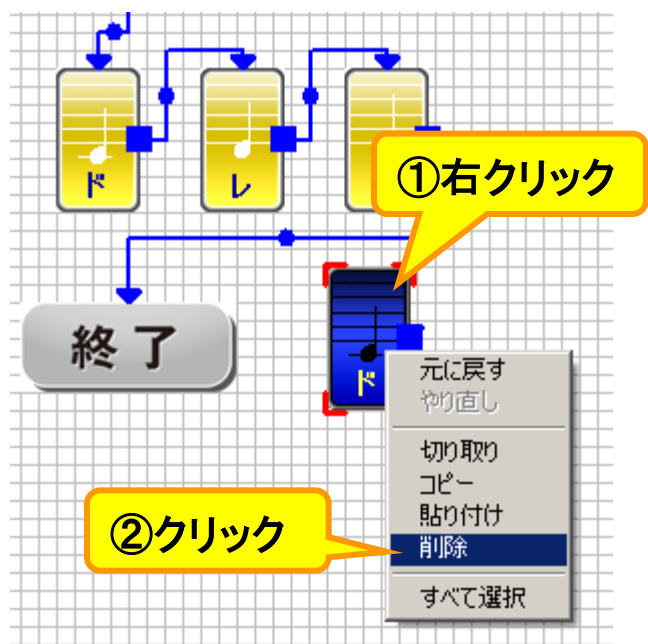
ブロックの整列

ブロックをドラッグすると位置を動かせるので、ブロックを整列して、見やすいプログラムを作成しましょう。

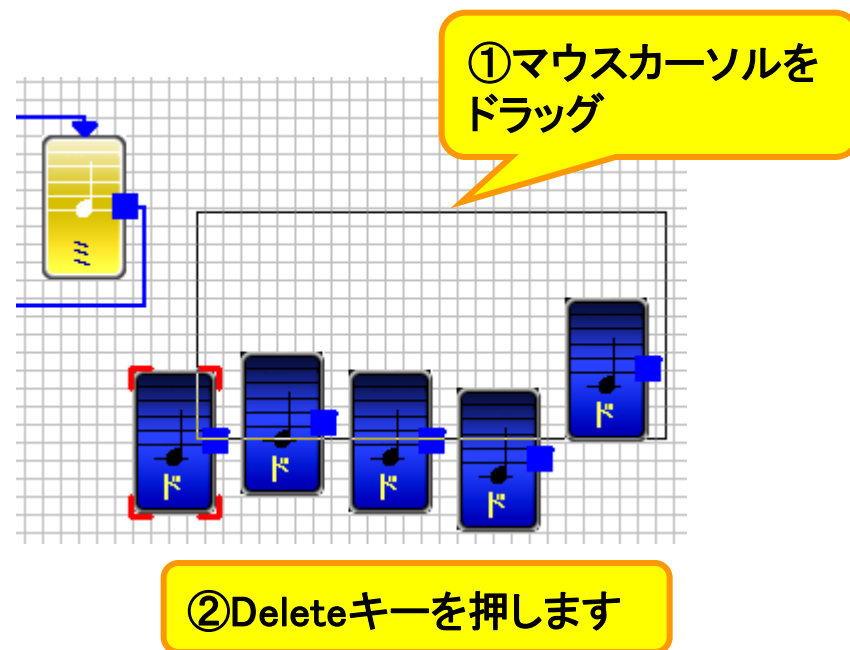


矢印がつながったブロックを近づけると、矢印が△印に省略されます。

不要なブロックは、次の操作で削除しましょう。

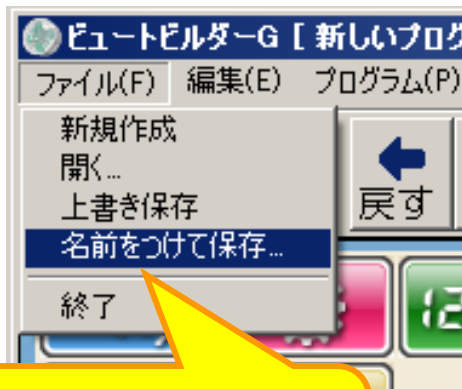


削除するブロックを右クリックし、メニューの「削除」を選択します。

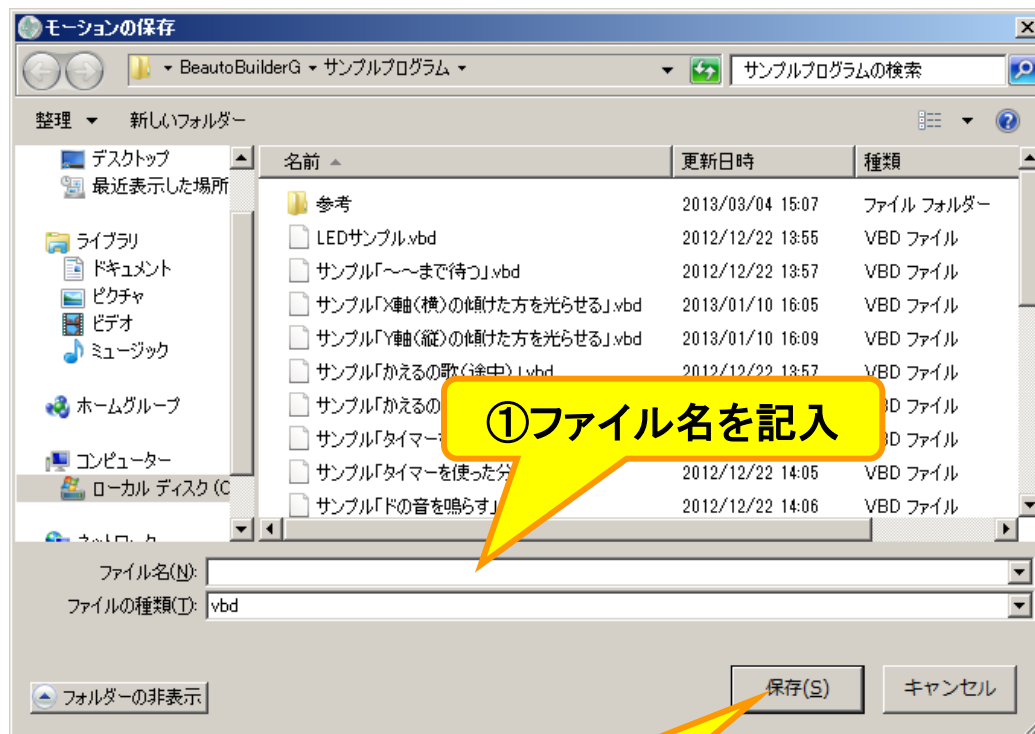


プログラムエリアでカーソルをドラッグすると複数のブロックを選択することができます。選択後、Deleteボタンを押すとブロックを削除することができます。

作成したプログラムはファイルに保存できます。



「ファイル」→
「名前をつけて保存」を
選択します。



①ファイル名を記入

②「保存」をクリック

その他便利な機能

- ・操作を間違えた場合:「戻る」「進む」でやり直せます。
- ・プログラムを新しく作り直す場合:「新規」で作成できます。



課題1 ドレミファソラシド(高)と鳴らすプログラム **Vstone™**

プログラムをスタートしたら、ドレミファソラシド(高)となるプログラムを作りましょう。(四分音符を使用します。)

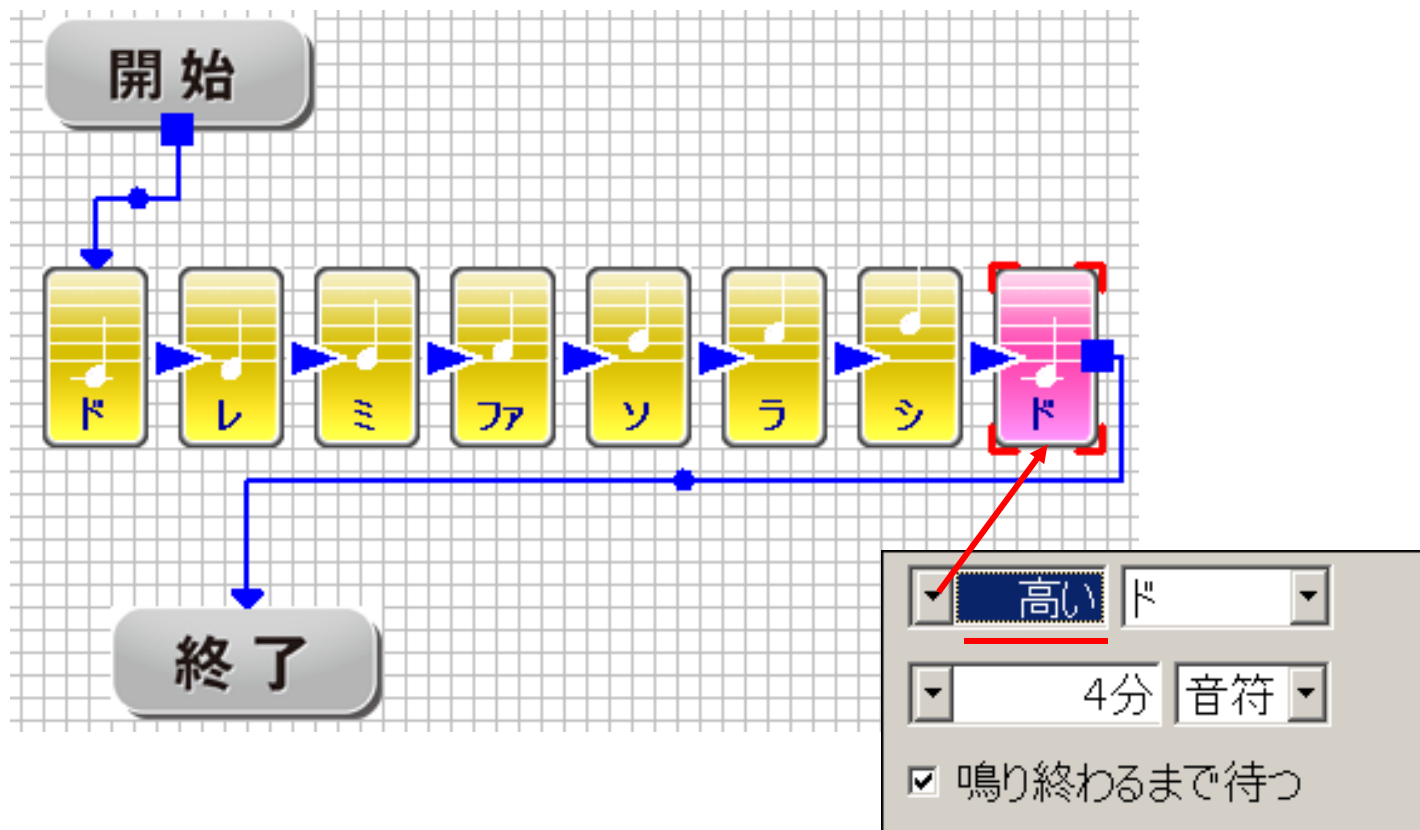
The screenshot displays the Vstone programming environment. The top menu bar includes options like 'ファイル(F)', '編集(E)', 'プログラム(P)', '設定(O)', 'レポート(R)', and 'ヘルプ(H)'. The toolbar contains icons for '新規', '開く', '保存', '戻る', '進む', '切取', 'コピー', '貼付', '書き込', '実行', '停止', '記録', and 'テンプレート'. A '接続中' indicator is visible in the top right.

The main workspace is a grid with a '開始' (Start) button at the top and a '終了' (End) button at the bottom. A vertical blue line connects them. A sequence of eight musical notes is placed on the grid: 'ド', 'レ', 'ミ', 'ファ', 'ソ', 'ラ', 'シ', and 'ド'. Each note is represented by a yellow square with a blue stem and a note head. The final 'ド' note is highlighted with a red border.

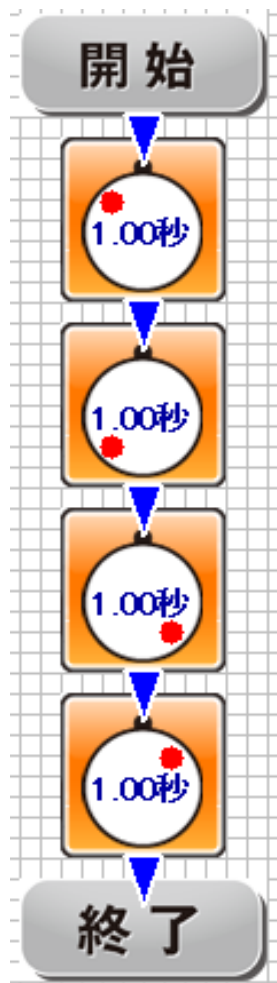
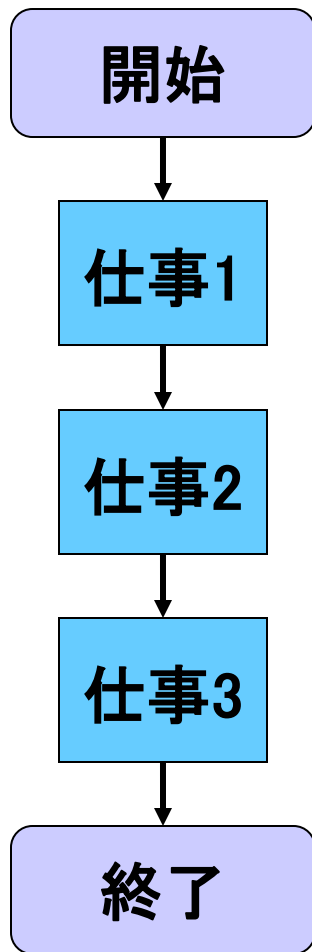
The left sidebar contains various controls:

- Speaker icon, LED icon, and a clock showing 12:34.
- IF, RANDOM, and LOOP buttons.
- Timer section: 'タイマー' set to 0分0.0秒, '操作ボタン' set to OFF, and a 3D coordinate system with X, Y, and Z axes.
- 3D coordinate values: X軸 +01, Y軸 -03, Z軸 -22.
- Dropdown menu for '普通の' (Normal) with a note icon.
- Dropdown menu for '4分' (Quarter) with a note icon.
- Checkbox '鳴り終わるまで待つ' (Wait until finished).

ドレミファソラシド(高)と鳴らすプログラム



高い「ド」に設定します。



身の回りにはコンピュータで制御されている機器がたくさんあります。それらの機器は、

- ・センサ
- ・コンピュータ
- ・アクチュエータ

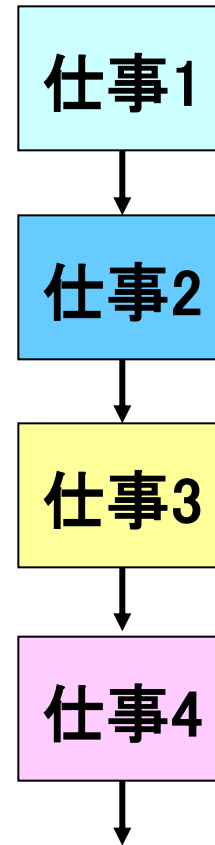
という要素を持っています。

制御のためのプログラムで仕事を順番に並べ、一つずつ実行するようなプログラムを、「**順次**」プログラムといいます。

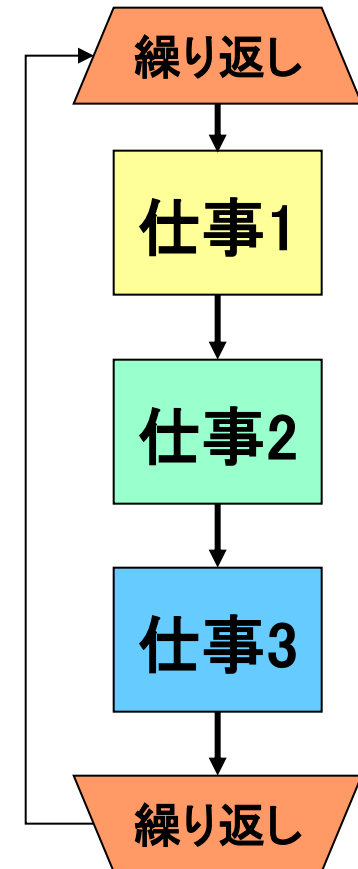
2時間目

- 繰り返しを使うプログラム
- LEDを使うプログラム

【順次】



【繰り返し】



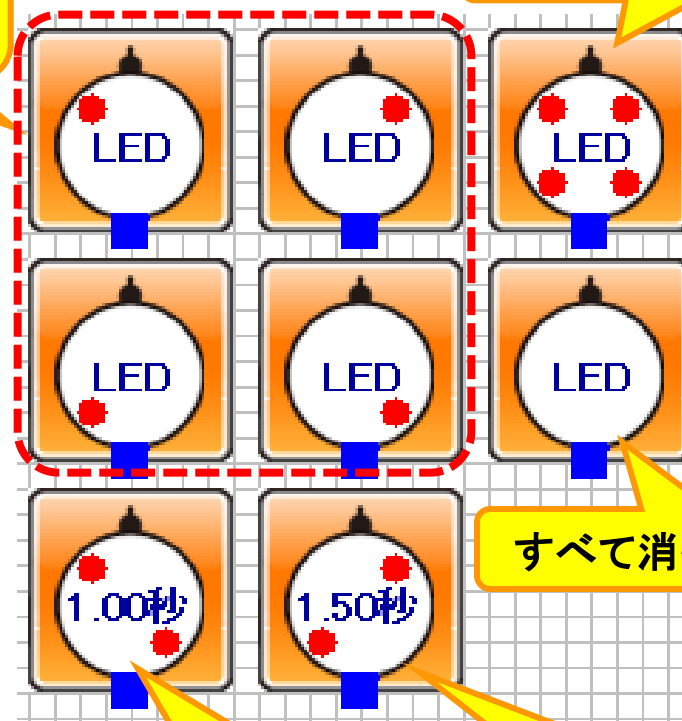
LEDの命令ブロックでLEDの点灯を制御することができます。



LED命令ブロック

それぞれ点灯
左上 右上
左下 右下

すべて点灯



チェックを付けた
場所が点灯

すべて消灯

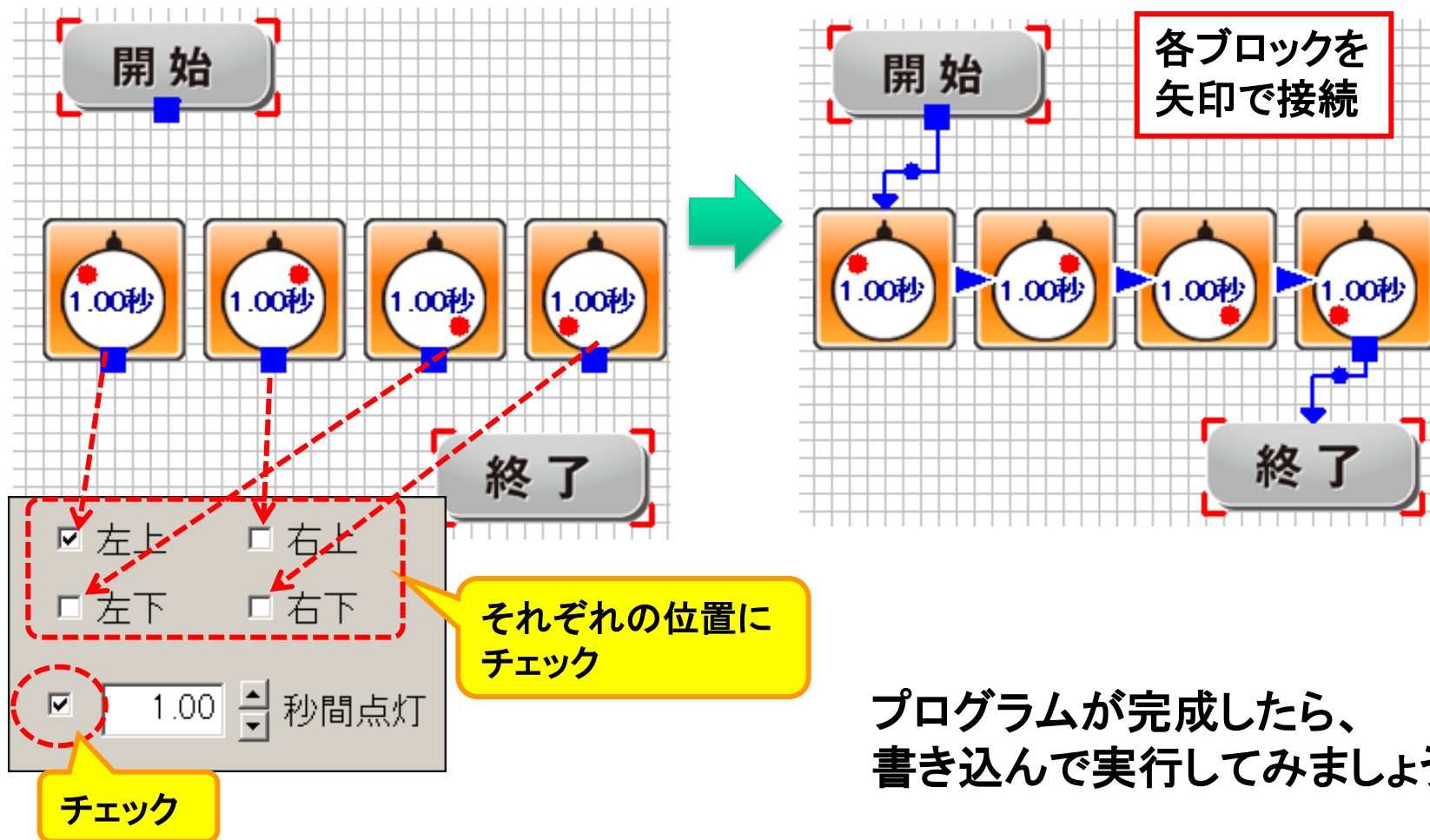
1秒間点灯

1.5秒間点灯

<input checked="" type="checkbox"/>	左上	<input checked="" type="checkbox"/>	右上
<input checked="" type="checkbox"/>	左下	<input checked="" type="checkbox"/>	右下
<input checked="" type="checkbox"/>	1.00	秒間点灯	

点灯時間を設定可能

LEDを時計回りで点灯するプログラムを作成しましょう。

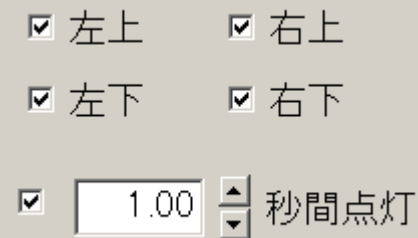


LEDの時間の設定

LEDブロックは点灯時間を設定できます。

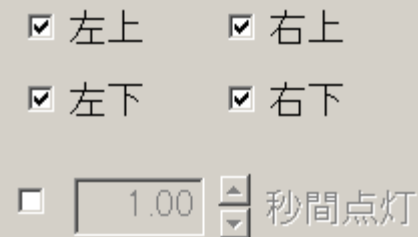
点灯時間を使う場合：

LEDを設定した時間光らせ、次の命令に進むときにLEDを消します。



点灯時間を使わない場合：

LEDを光らせたらずぐに次の命令に進みます(LEDはつけっぱなし)。

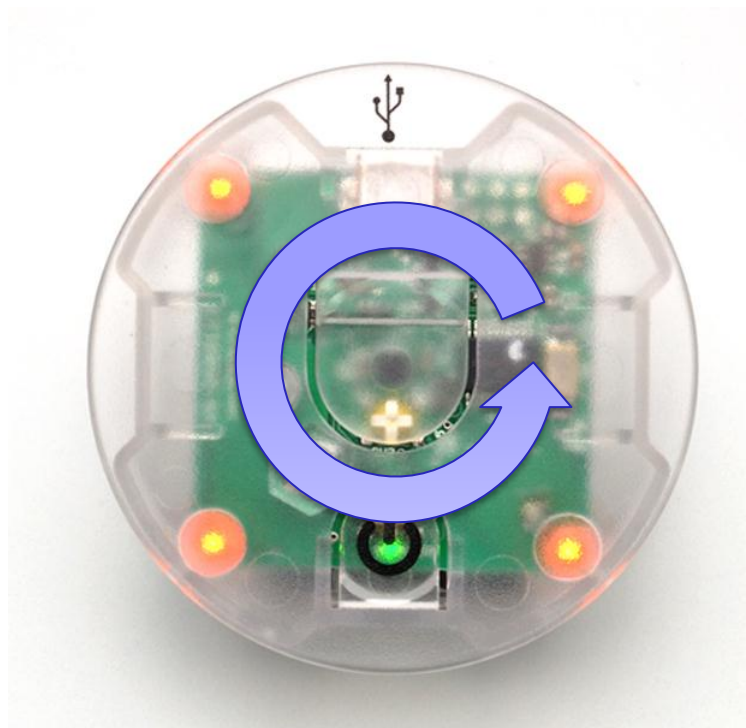


点灯時間を使わない設定は、LEDとブザーやセンサを同時に使うときや、各LEDのON/OFF状態だけを変えたいときに選びます。

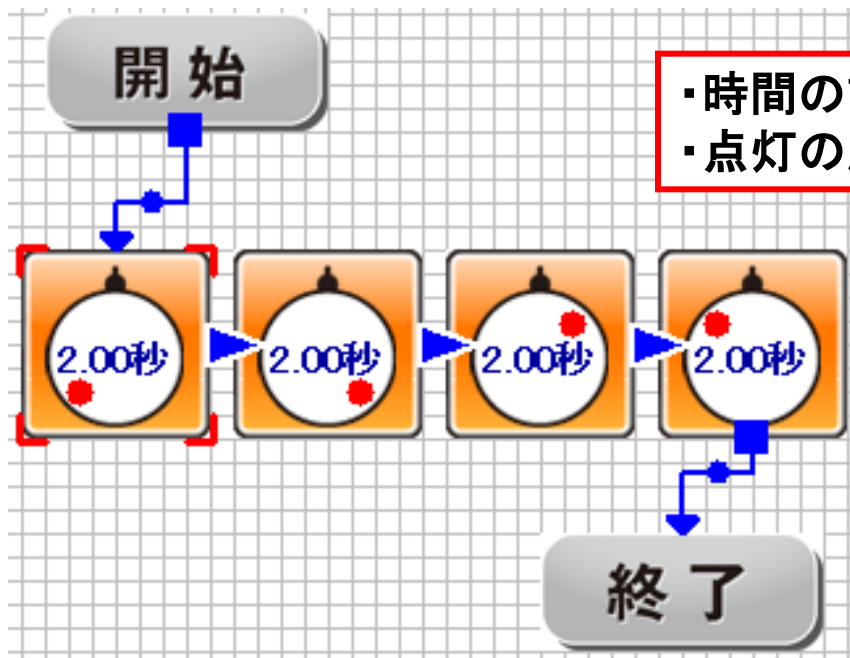
点灯時間を使わない設定は、後の分岐のプログラムで行います。

課題2 反時計回りで点滅する(各2秒)プログラム

プログラムをスタートしたら、LEDを反時計回りに点灯するプログラムを作りましょう。LEDは、一つにつき2秒光らせましょう。

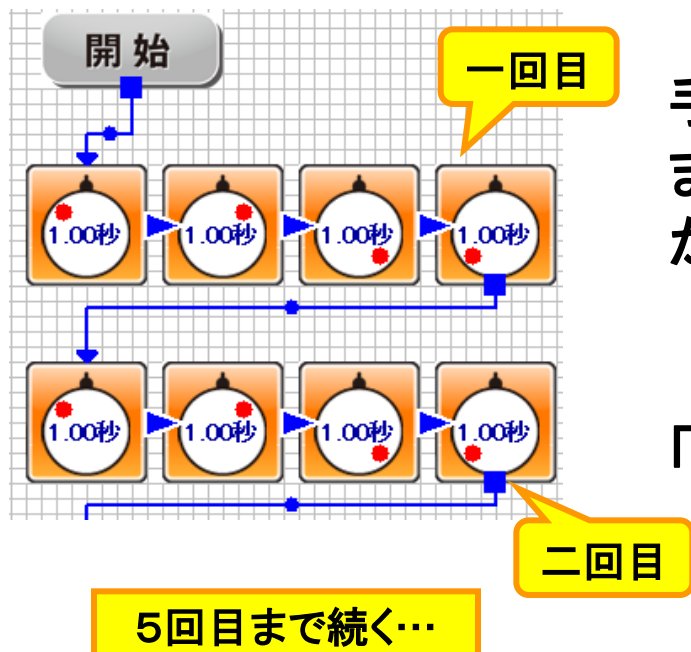


LEDを反時計回りに点灯するプログラム



- ・時間の設定を忘れずに行いましたか？
- ・点灯の順番は正しかったですか？

LEDを時計回りで5周光らせるプログラムを作成しましょう。



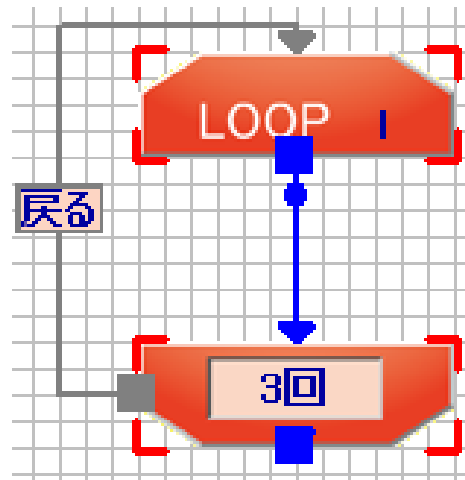
同じ命令を5回分並べると作成できますが、手間がかかりプログラムも見づらくなります。また、10回繰り返したい場合は、更に手間がかかります。

このように同じ命令を何度も実行する場合は、「**繰り返し**」を使います。

繰り返しブロックの使い方

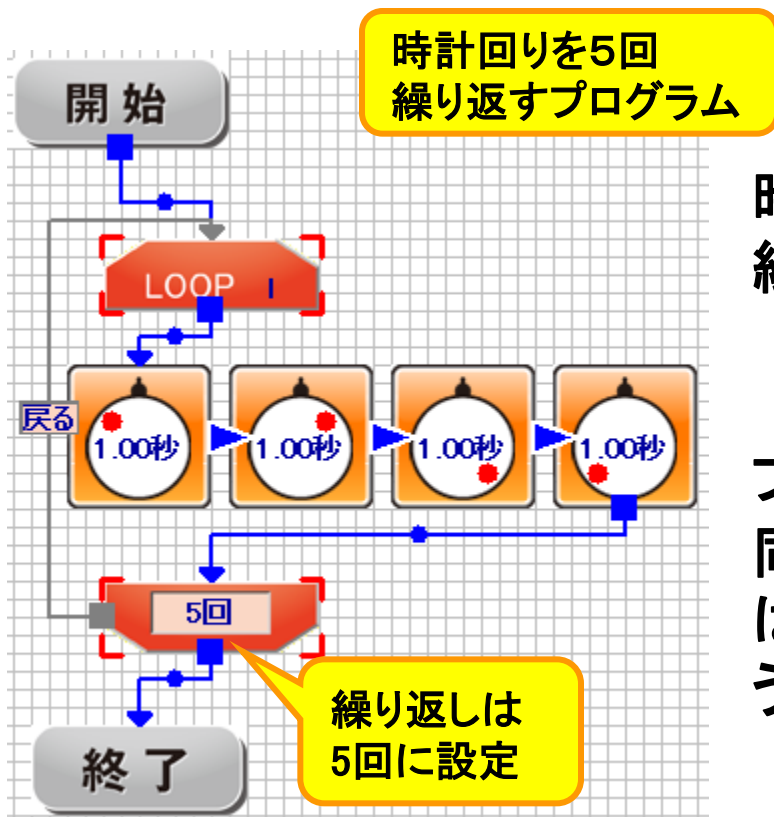


繰り返しブロックは、「繰り返しの始まり」と「終わり」の二つがセットになっており、繰り返す回数を自由に設定できます。



繰り返し回数を設定

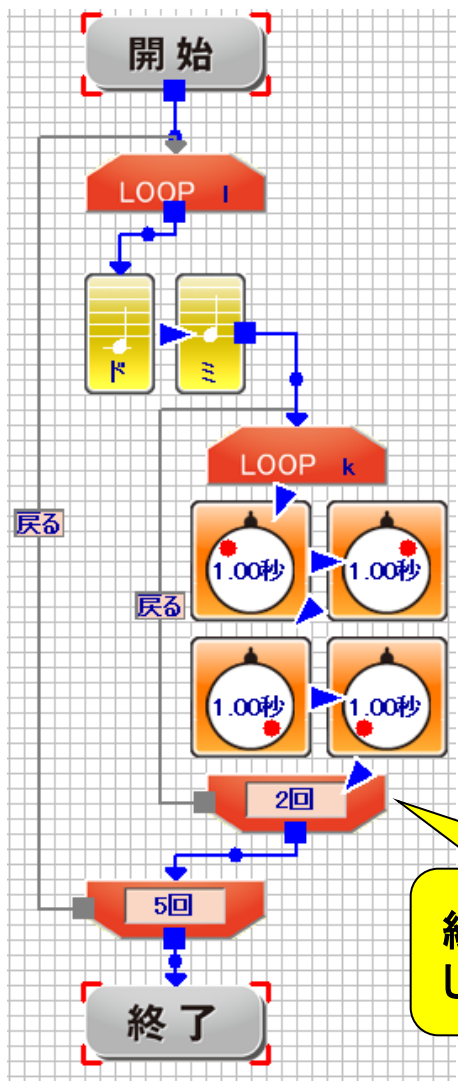
- 3 回くりかえす
- ずっとくり返す(無限)



時計回りにLEDを光らせるプログラムに、繰り返しブロックを追加します。

繰り返しの命令を活用すると、効率よくプログラミングできます。

同じような命令がいくつも出てくる場合は、共通する部分を繰り返しの命令でうまくまとめましょう。

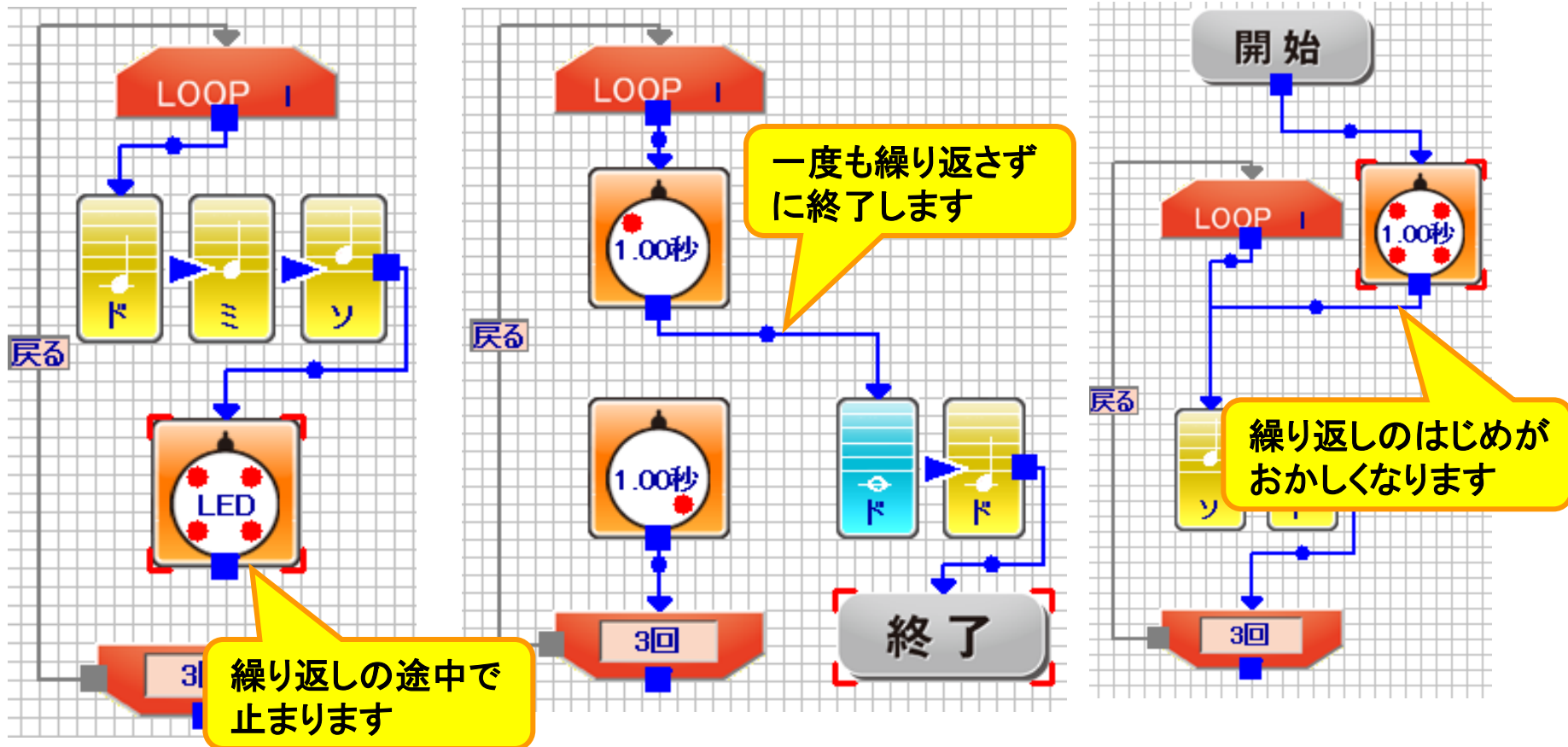


「繰り返し」の中に「繰り返し」を使うという
ような複雑な構造もできます。

繰り返しの中で繰り返
しを使うプログラム例

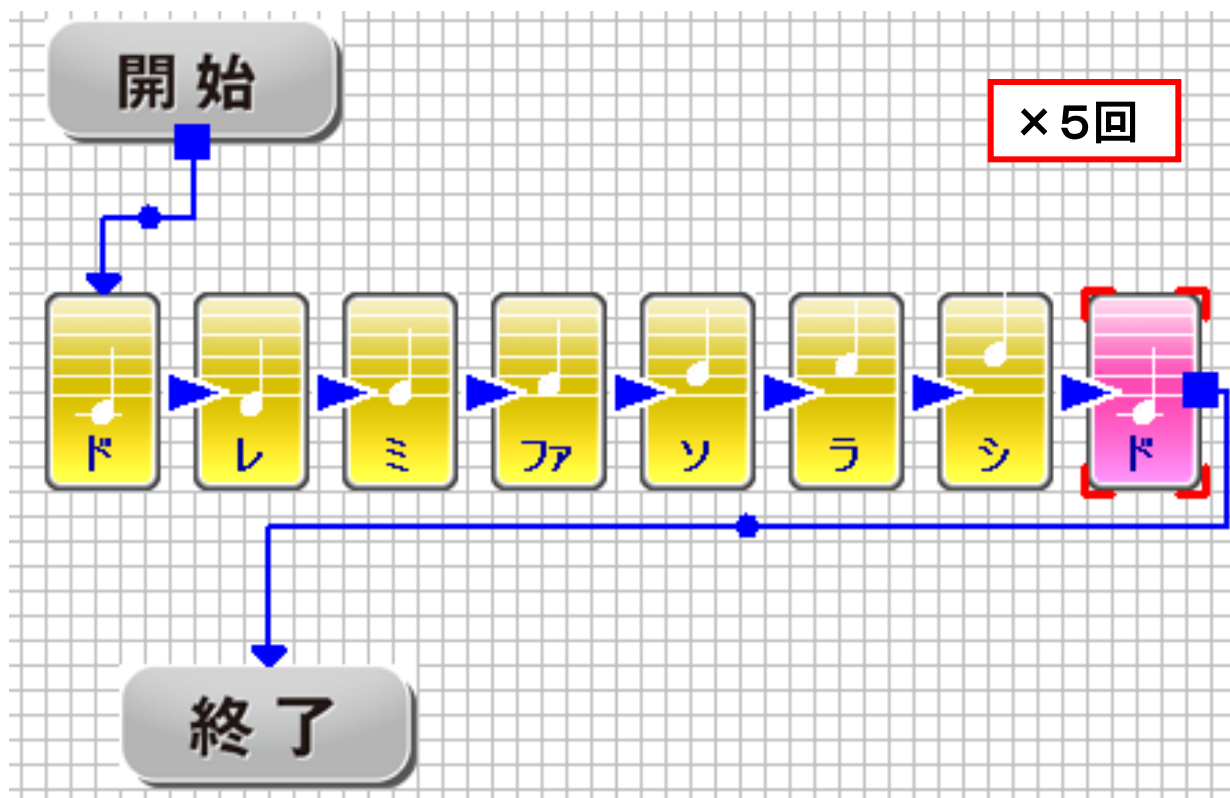
繰り返し命令の注意点

繰り返し命令は、「繰り返しの始まり」から「終わり」に正しくつなげないと、おかしい動作になります。

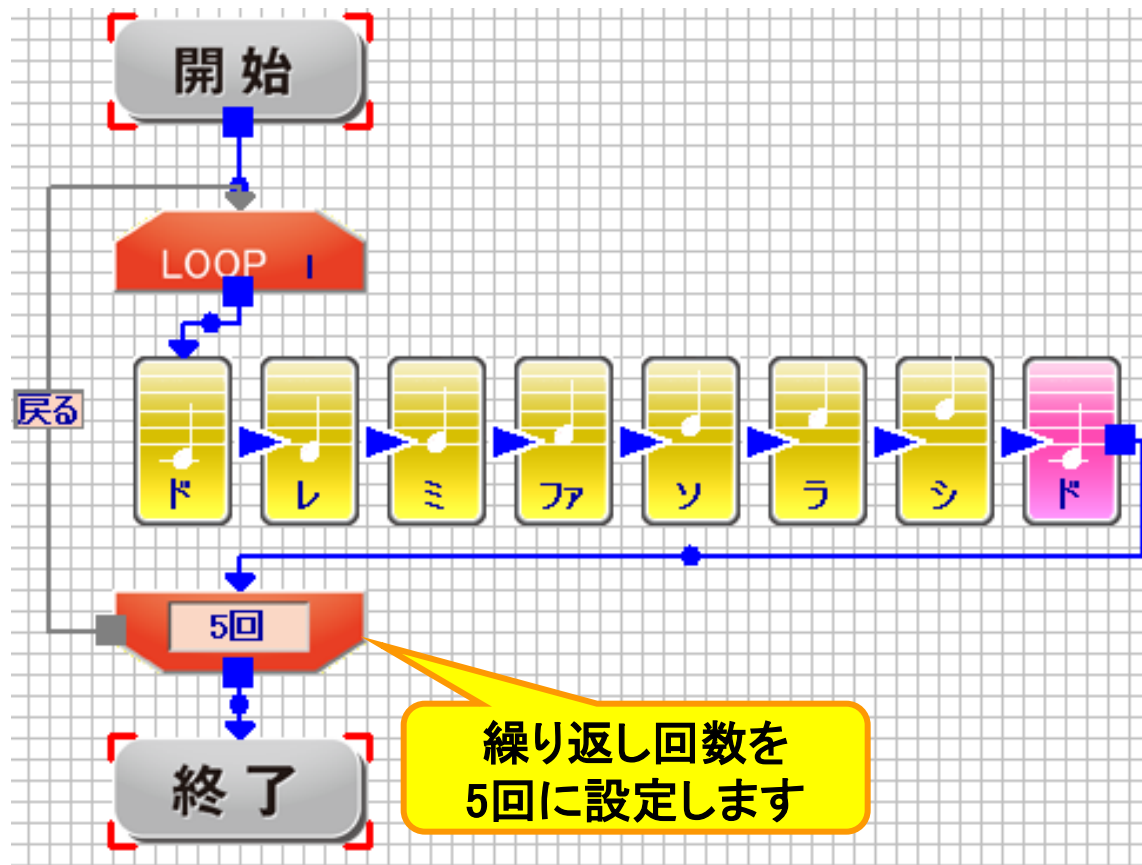


課題3 ドレミファソラシドを5回繰り返すプログラム

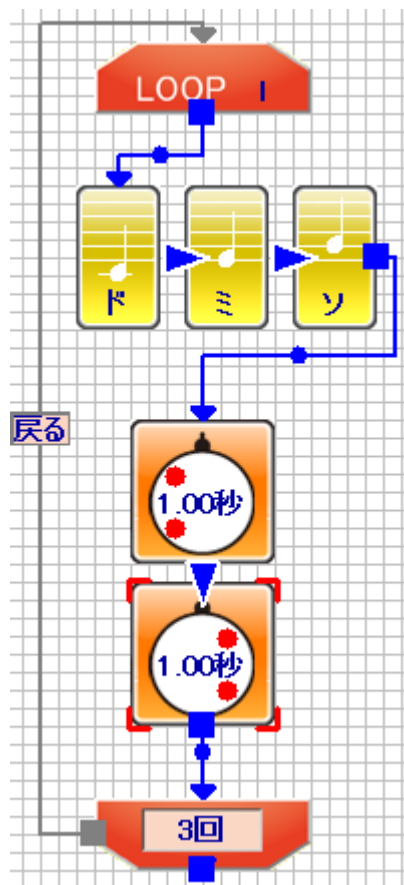
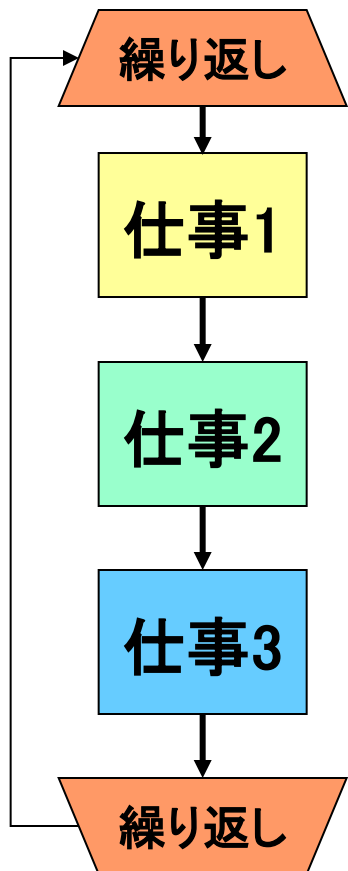
ドレミファソラシド（課題1）を5回繰り返すプログラムを作成しましょう。繰り返しブロックを1つ使用します。



ドレミファソラシド(課題1)を5回繰り返すプログラム



【繰り返し】

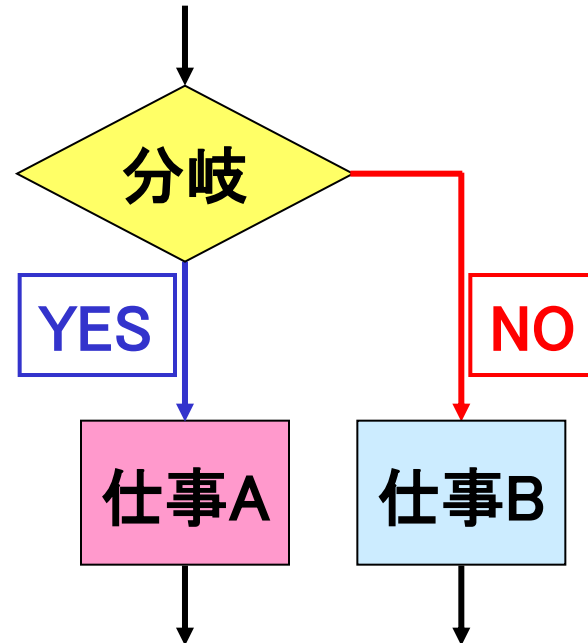


同じ命令を何度も繰り返して実行するプログラムを「**繰り返し**」のプログラムといいます。

繰り返しは便利な命令ですが、矢印をつなぎ間違えると、正しくプログラムが動かないため注意が必要です。

3時間目

- 加速度センサ
- デジタルとアナログ
- 分岐のプログラム
- センサ値の記録

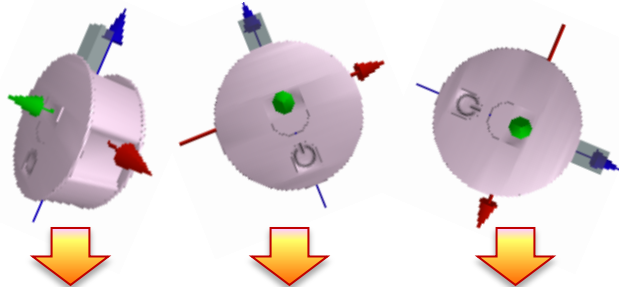


加速度センサについて

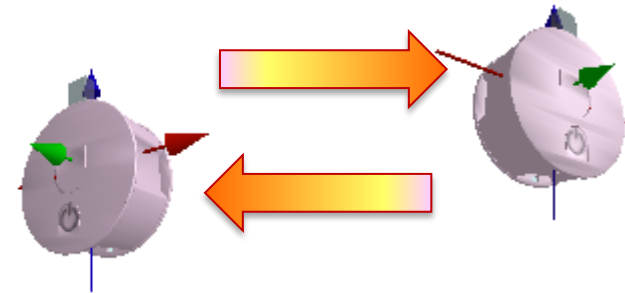
本体に搭載している加速度センサは「加速度」を調べられます。

加速度とは・・・？ 一言で表すと「**加速の大きさ**」

加速度の例



①重力(重力加速度)
物体の向きに関わらず、常に下方向へ
加速度が発生。

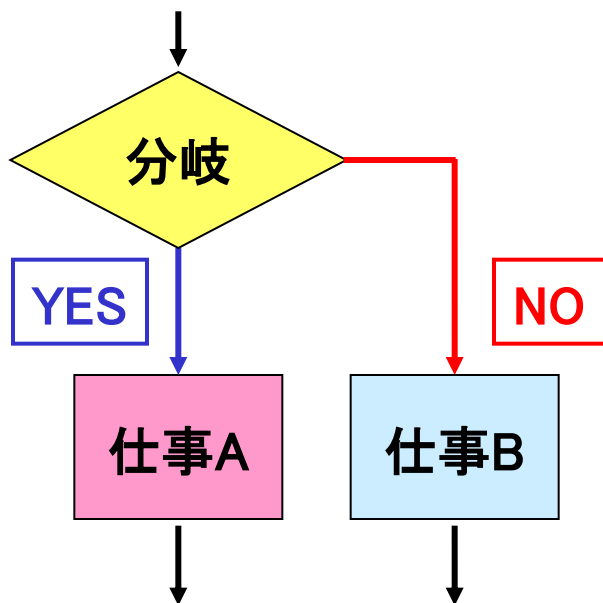


②物体の移動・振動
動いた方向に対して加速度が発生。

加速度センサが使われている機器：
スマートフォンやゲームのコントローラなどの最新電子機器

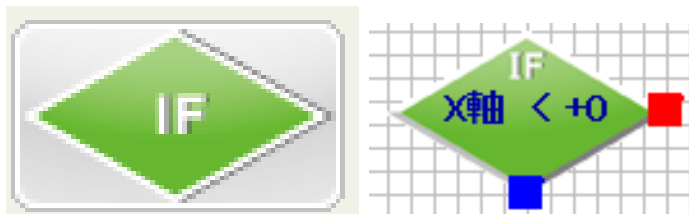
分岐のプログラム

センサを使って、状態によって動きが変わるプログラムを作りましょう。



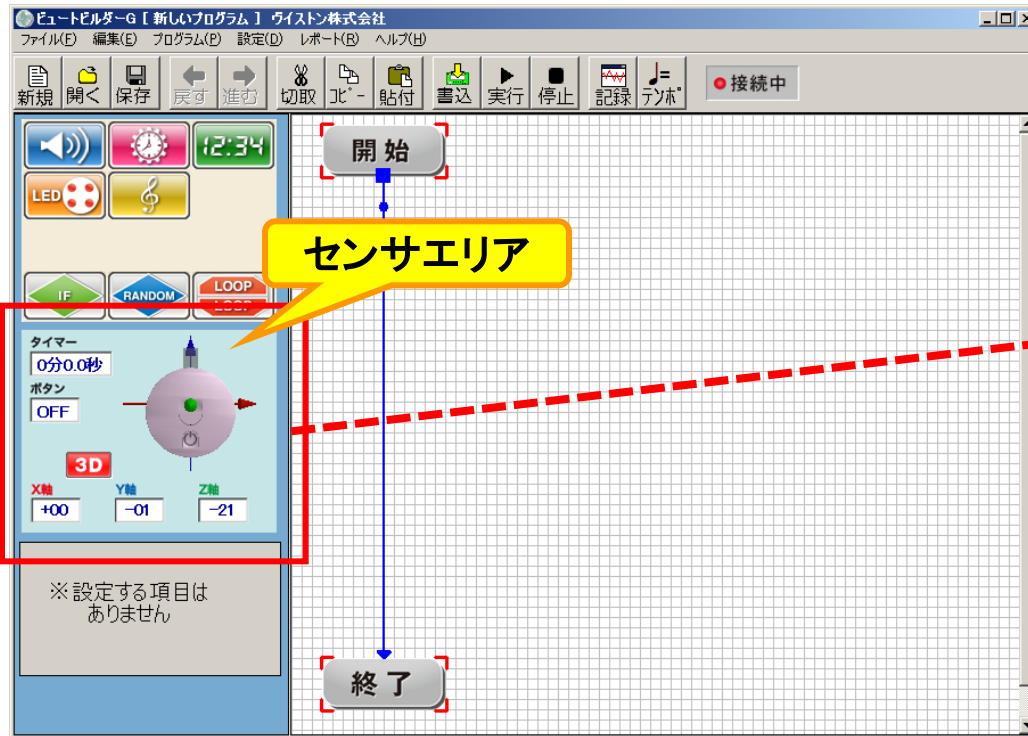
条件によって処理の内容を変えたい場合に、「分岐」を使います。

例えば、自動ドアでは、センサで感知して
・ドアの前に人がいる場合、「開く(仕事A)」
・ドアの前に人がいない場合、「閉じる(仕事B)」
という処理が行なわれています。

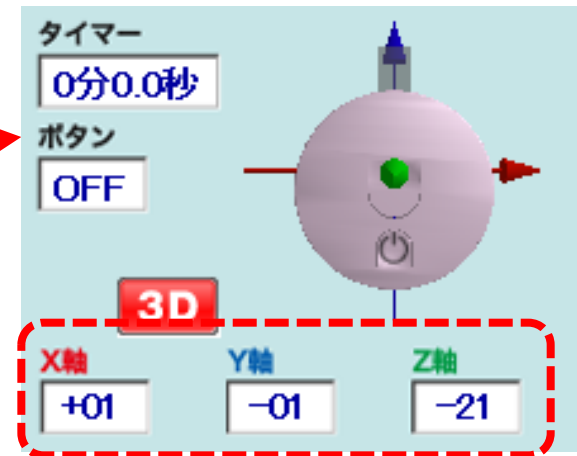


本体を左右に傾けたら、傾けた方向のLEDが光るプログラムを作りましょう。

センサ値の計測



パソコンに接続するとセンサ値が
センサエリアに表示されます。



左右に傾けたときの反応を確かめましょう。

- ・左に傾けると、「X軸」はいくつになるのでしょうか？
- ・右に傾けると、「X軸」はいくつになるのでしょうか？

左と右に傾けたときの状態を区別するには？

タイマー
0分0.0秒

ボタン
OFF

3D

X軸 Y軸 Z軸
-02 +01 -21

● 加速度センサ ● ボタン
● 時間 ● その他

X軸 が
0 より小さい?

しきい値

IF
X軸 < +0

終了

- ・左に傾ける: X軸 = 0より小さい
- ・右に傾ける: X軸 = 0より大きい

左右に傾いた状態が切り替わる境目は、X軸が「0」の時です。

このように、二つの状態を分ける数値を「しきい値」と言います。

→分岐の条件を、図のように『X軸』が『0』『より小さい?』にしましょう。

条件は「**X軸が0より小さい?**」です。

0より小さい→「はい」

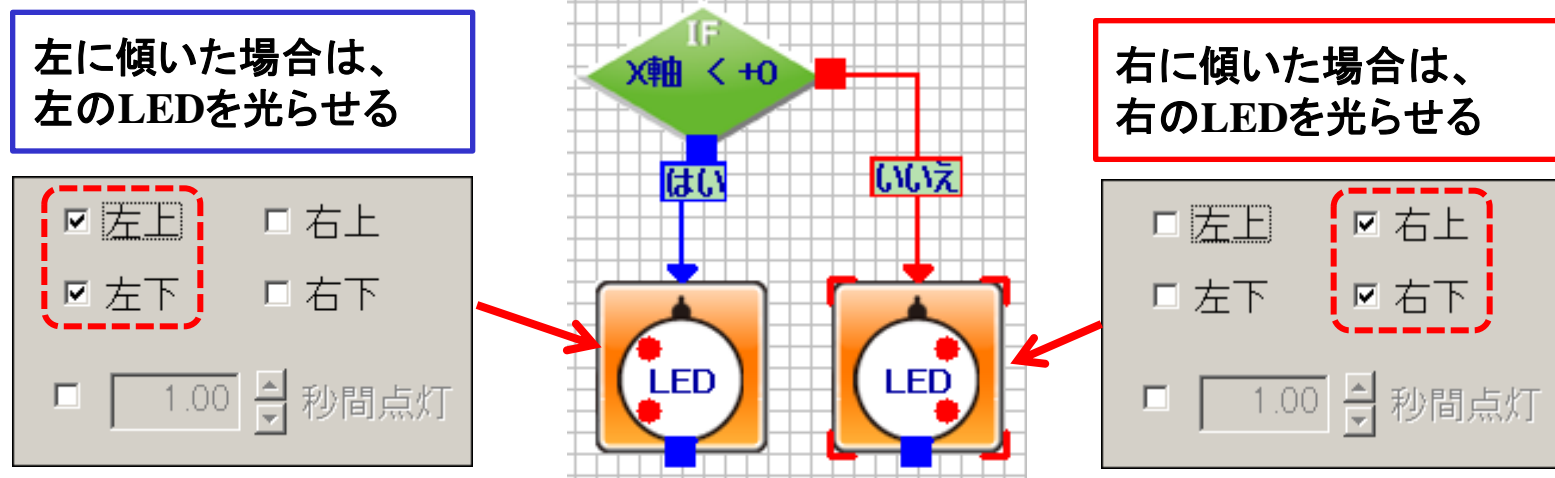
0より大きい→「いいえ」

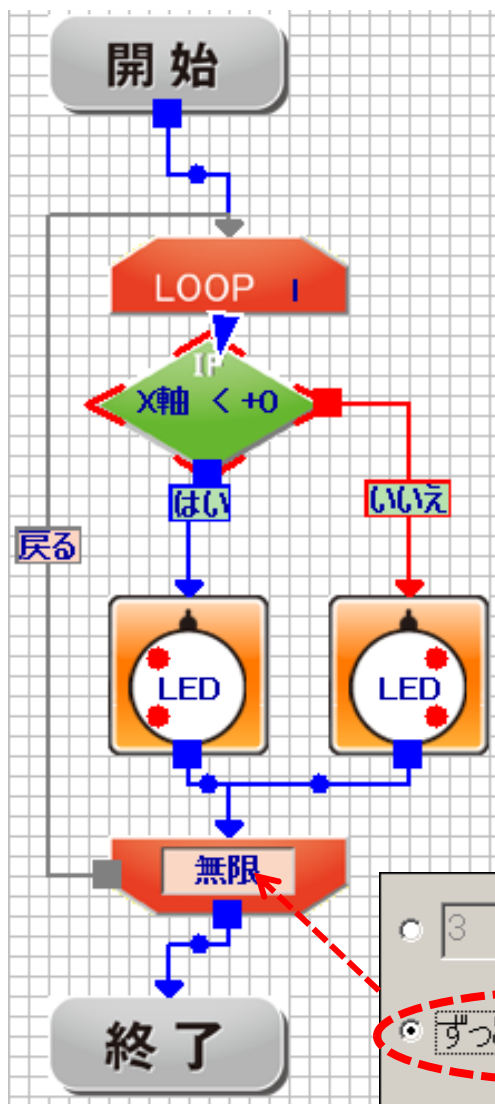
つまり…

→左に傾いたら「はい」、右に傾いたら「いいえ」に進みます。

図に従って、分岐の先にLEDブロックをつないで設定しましょう。

※LEDの点灯時間は使わないようにしましょう。





最後に繰り返しを入れましょう。

繰り返しが無いと・・・

→分岐を1回だけして終わってしまいます。

繰り返し(無限)を入れると

→何度も戻って分岐を実行します。

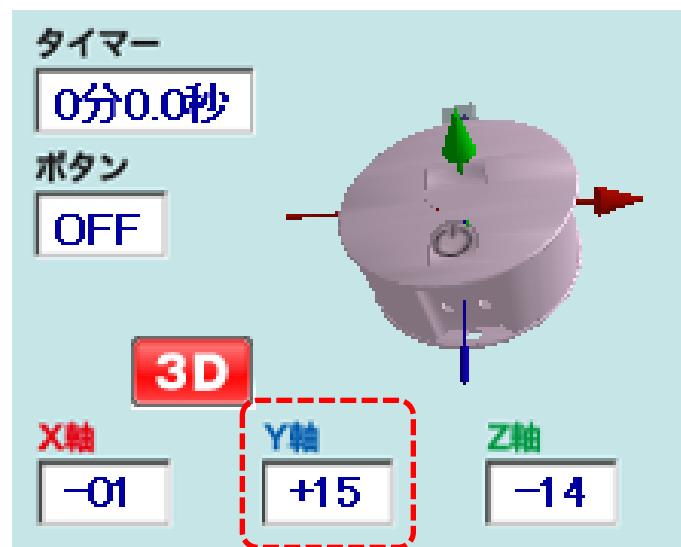
プログラムができたなら実行してみましよう。



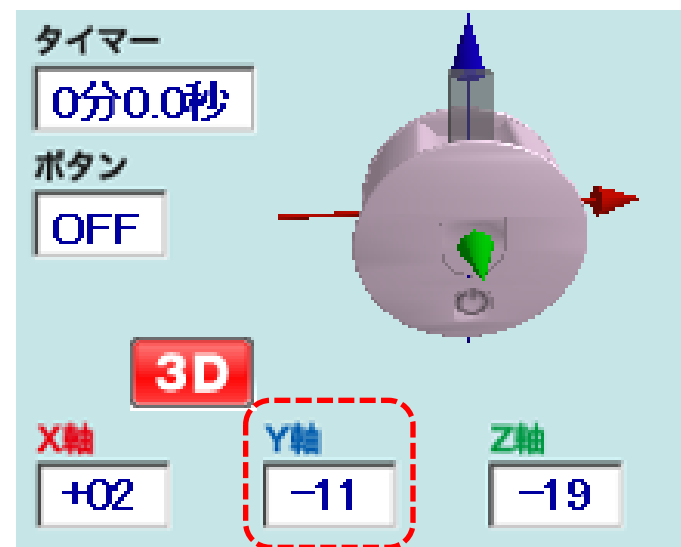
本体を前後に傾けて、傾けた方向のLEDを光らせるプログラムを作りましょう。

加速度センサのY軸を調べます

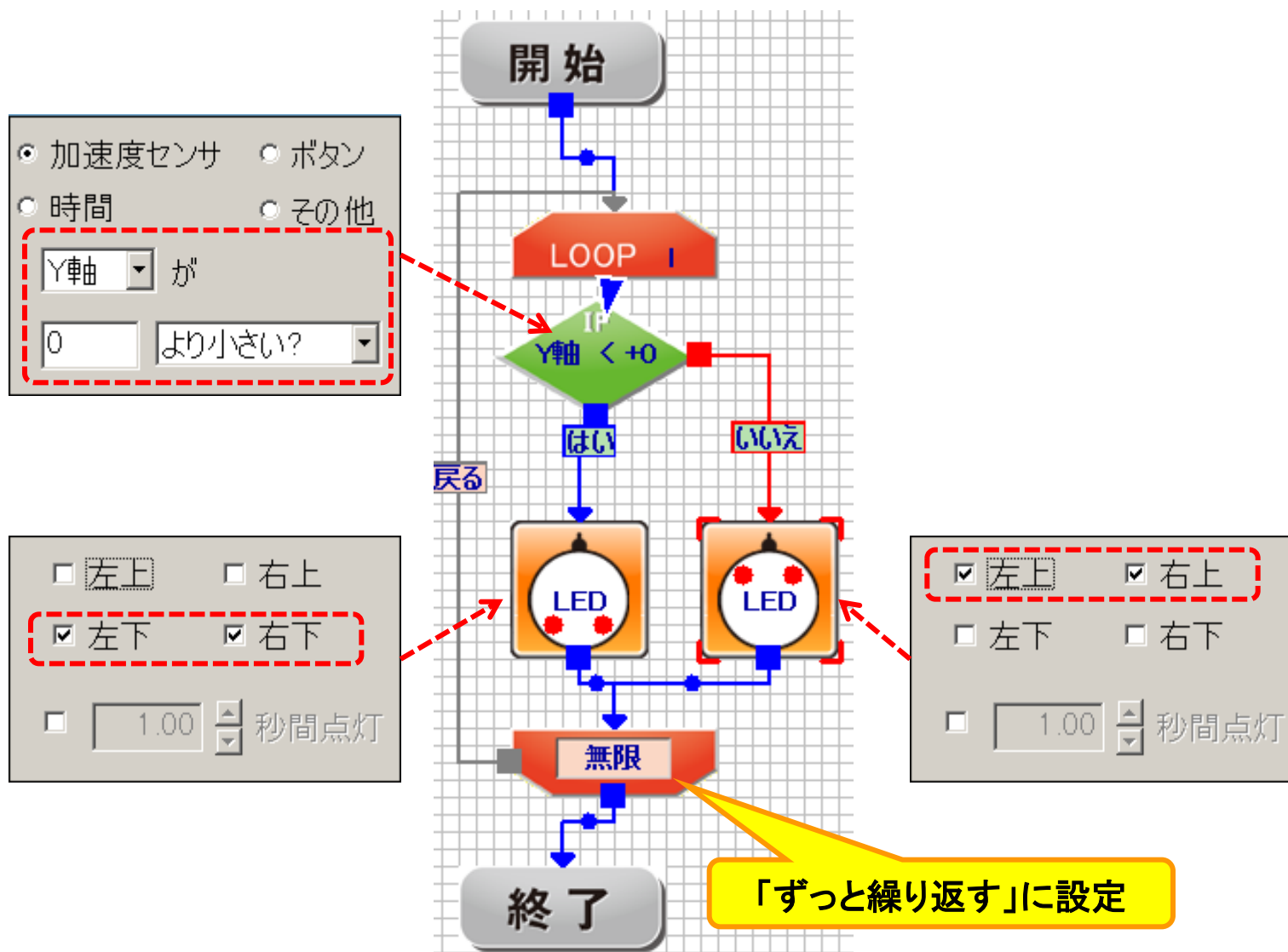
前に傾けた場合



後ろに傾けた場合



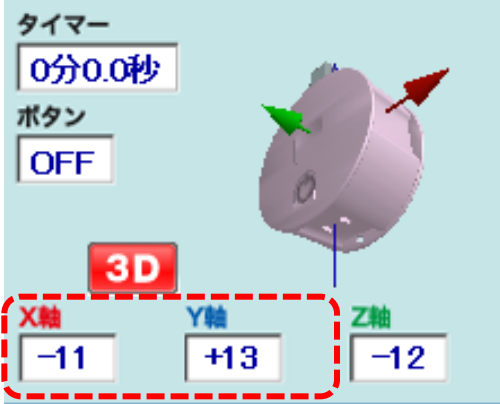
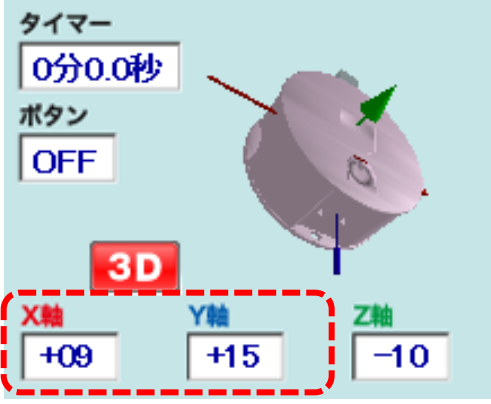
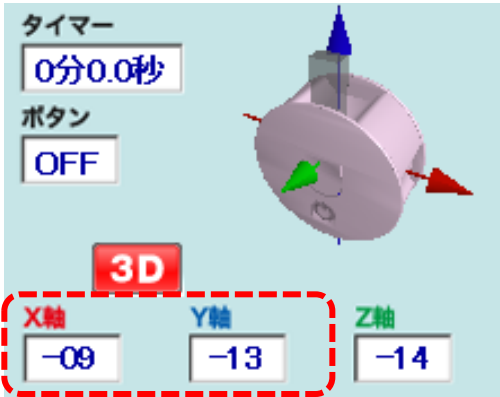
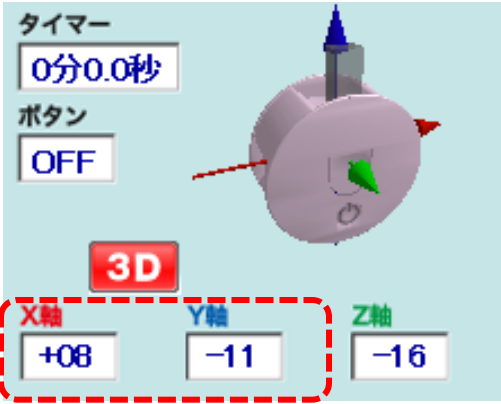
課題4 解答例



課題5 分岐を使った複雑なプログラム①

本体を前後左右に傾けて、傾けた方向のLEDを光らせるプログラムを作りましょう。

前後左右の区別は、X軸・Y軸を組み合わせます。

①	 <p>タイマー 0分0.0秒 ボタン OFF</p> <p>3D</p> <table border="1"><tr><td>X軸</td><td>Y軸</td><td>Z軸</td></tr><tr><td>-11</td><td>+13</td><td>-12</td></tr></table>	X軸	Y軸	Z軸	-11	+13	-12
X軸	Y軸	Z軸					
-11	+13	-12					
②	 <p>タイマー 0分0.0秒 ボタン OFF</p> <p>3D</p> <table border="1"><tr><td>X軸</td><td>Y軸</td><td>Z軸</td></tr><tr><td>+09</td><td>+15</td><td>-10</td></tr></table>	X軸	Y軸	Z軸	+09	+15	-10
X軸	Y軸	Z軸					
+09	+15	-10					
③	 <p>タイマー 0分0.0秒 ボタン OFF</p> <p>3D</p> <table border="1"><tr><td>X軸</td><td>Y軸</td><td>Z軸</td></tr><tr><td>-09</td><td>-13</td><td>-14</td></tr></table>	X軸	Y軸	Z軸	-09	-13	-14
X軸	Y軸	Z軸					
-09	-13	-14					
④	 <p>タイマー 0分0.0秒 ボタン OFF</p> <p>3D</p> <table border="1"><tr><td>X軸</td><td>Y軸</td><td>Z軸</td></tr><tr><td>+08</td><td>-11</td><td>-16</td></tr></table>	X軸	Y軸	Z軸	+08	-11	-16
X軸	Y軸	Z軸					
+08	-11	-16					

センサの値を確認

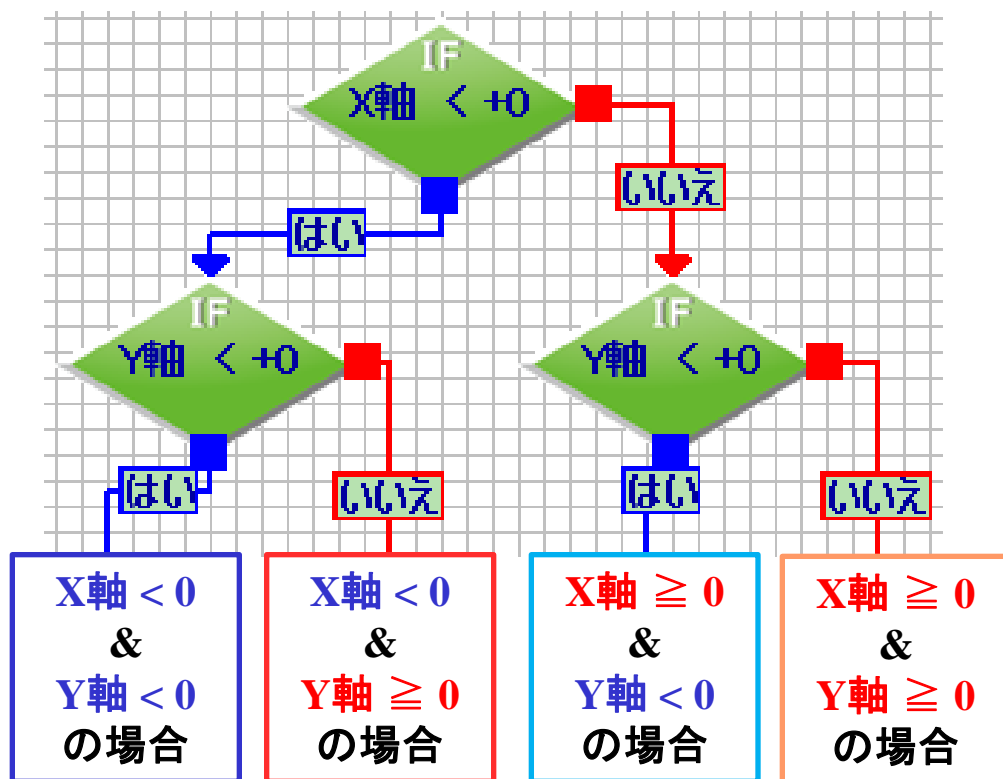
- ①左前に傾けた場合：
 $X < 0 \ \& \ Y \geq 0$

- ②右前に傾けた場合：
 $X \geq 0 \ \& \ Y \geq 0$

- ③左後ろに傾けた場合：
 $X < 0 \ \& \ Y < 0$

- ④右後ろに傾けた場合：
 $X \geq 0 \ \& \ Y < 0$

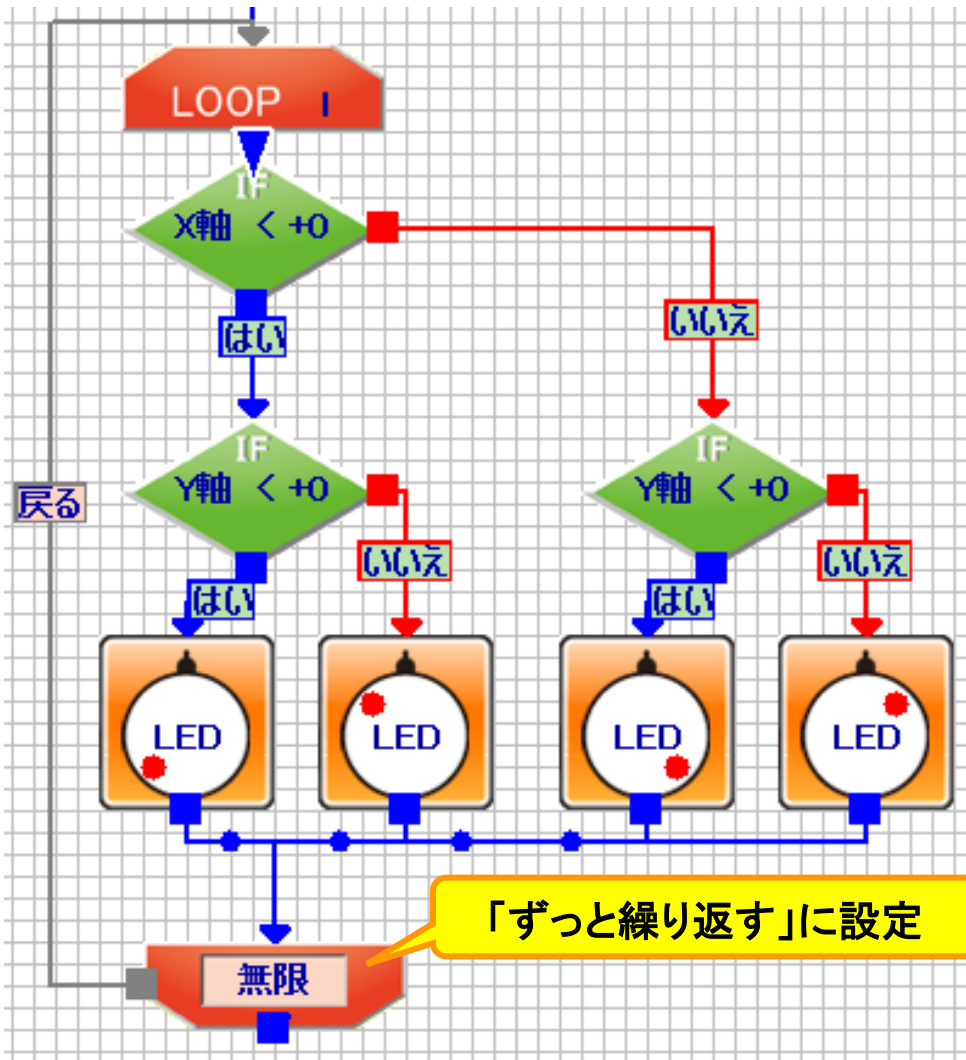
分岐を重ねることで、複雑な分岐を作ることができます。



分岐を重ねると、複数の条件を使ったプログラムが可能です。

前ページの図も参考に、傾けた方向のLEDを光らせるプログラムを作りましょう。

課題5 解答例



課題6 裏返すとLEDがすべて消えるプログラム **Vstone**TM

課題5のプログラムを使用して、前後左右に傾けた方のLEDが光り、本体を裏返すとLEDがすべて消えるプログラムを作りましょう。



タイマー
0分0.0秒
ボタン
OFF
3D
X軸 -02 Y軸 +01 Z軸 -21



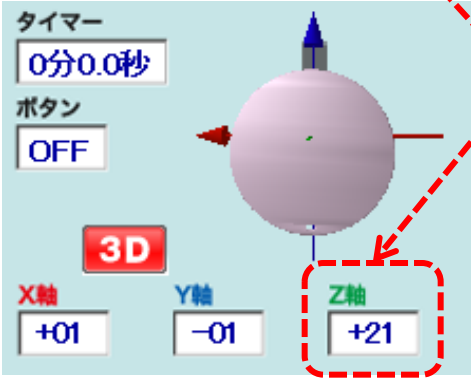
タイマー
0分0.0秒
ボタン
OFF
3D
X軸 -11 Y軸 +13 Z軸 -12

タイマー
0分0.0秒
ボタン
OFF
3D
X軸 +09 Y軸 +15 Z軸 -10

タイマー
0分0.0秒
ボタン
OFF
3D
X軸 -09 Y軸 -13 Z軸 -14

タイマー
0分0.0秒
ボタン
OFF
3D
X軸 +08 Y軸 -11 Z軸 -16

表向きの場合
更に前後左右の
4通りの場合に分岐



タイマー
0分0.0秒
ボタン
OFF
3D
X軸 +01 Y軸 -01 Z軸 +21

裏表はZ軸を
調べます

裏向きの場合
「LEDを全て消す」
の1通りだけ

課題6 裏返すとLEDがすべて消えるプログラム **Vstone**TM

分岐ブロックを3つ連続で使用します(分岐ブロック自体は合計4個)。

・1つ目の分岐

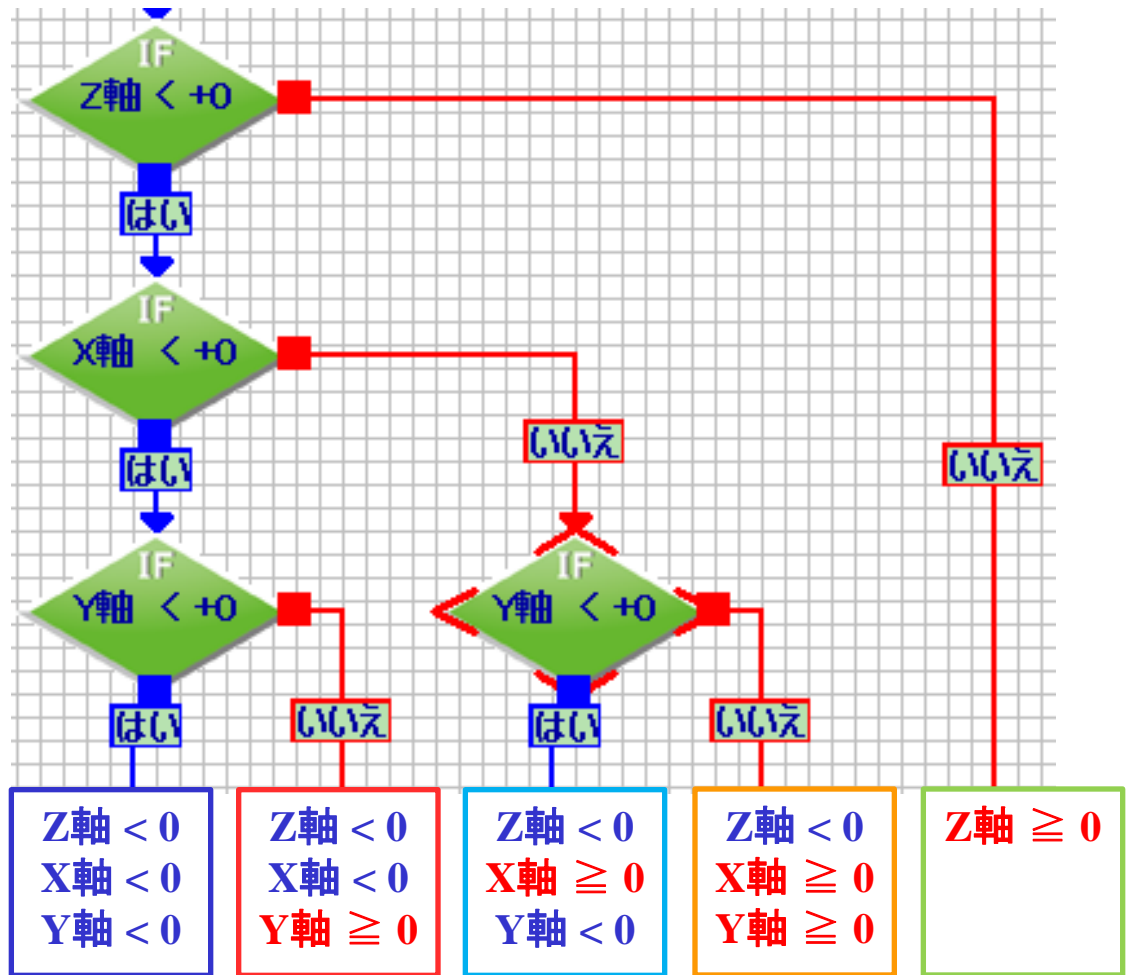
→ 表向きか裏向きか

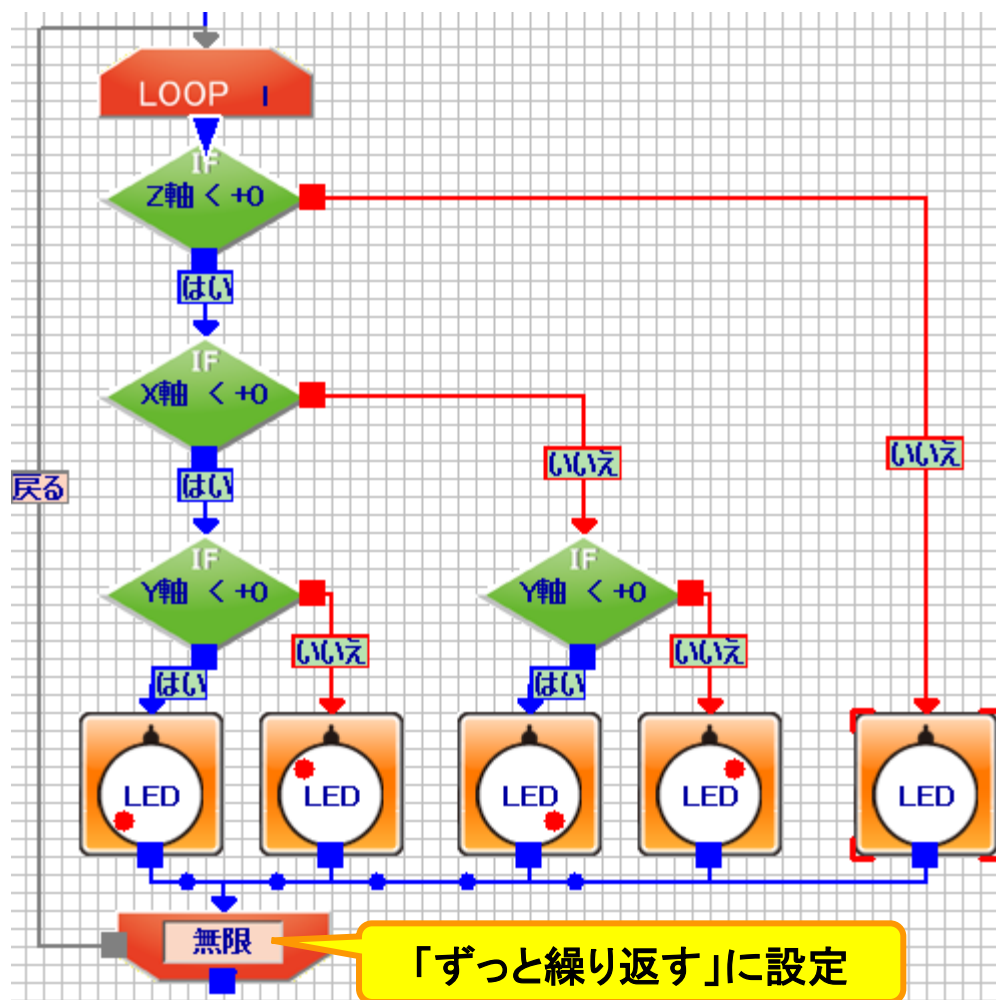
・2つ目の分岐

→ 左向きか右向きか

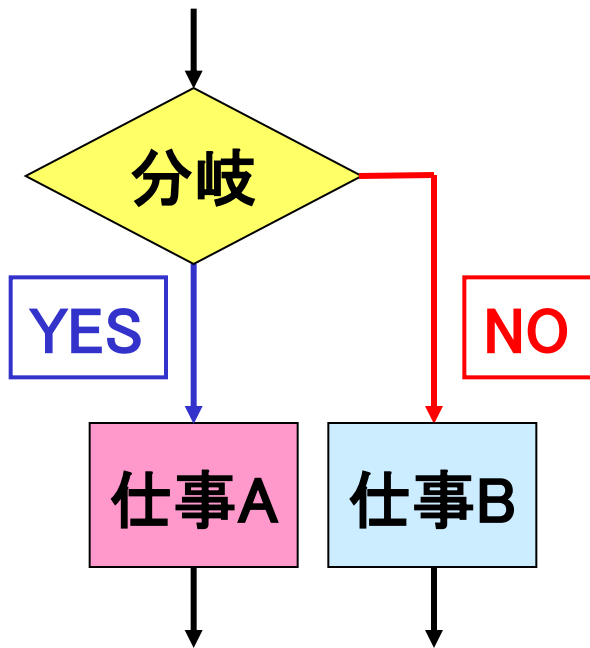
・3つ目の分岐

→ 後ろ向きか前向きか





【分岐】



決められた条件に応じて処理を切り替えるプログラムを「**分岐**」のプログラムと言います。

二つの状態の境目を表す数値を「**しきい値**」といい、これを使って「**条件**」を作ります。

分岐を使うときは、次の三つを考えて正しく組み合わせる必要があります。

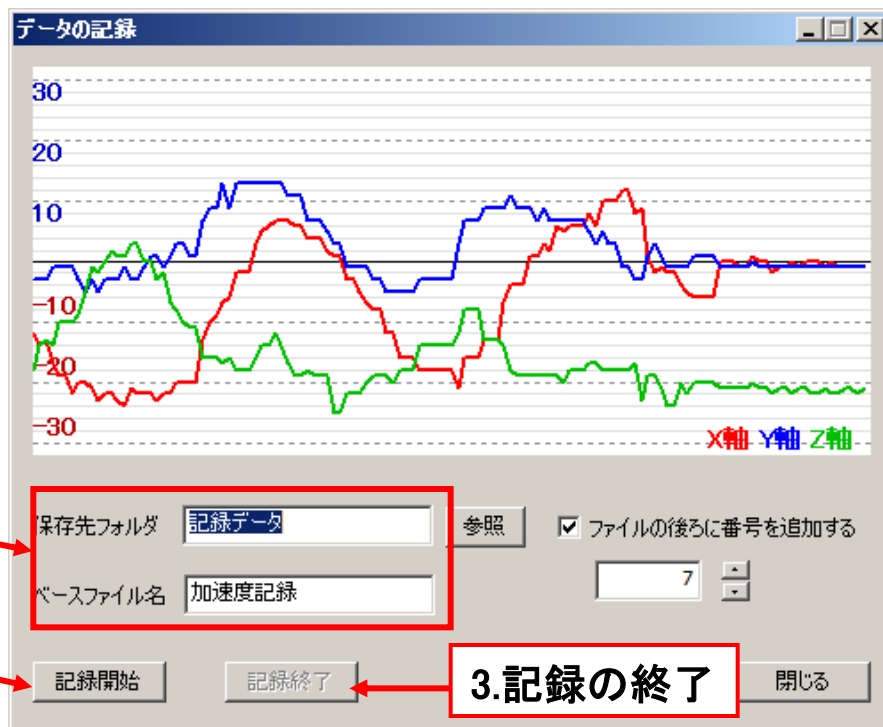
- ① **分岐の条件は何？**（加速度センサ、赤外線センサなど）
- ② **その条件で、それぞれどんな場合に「はい」、「いいえ」に振り分けるか**
- ③ **「はい」「いいえ」に進んだ先に、どんな命令をつなげるか**

センサの情報をパソコンに記録①

パソコンと本体を接続すれば、センサ値の記録が可能です。



センサ値のグラフが表示されます



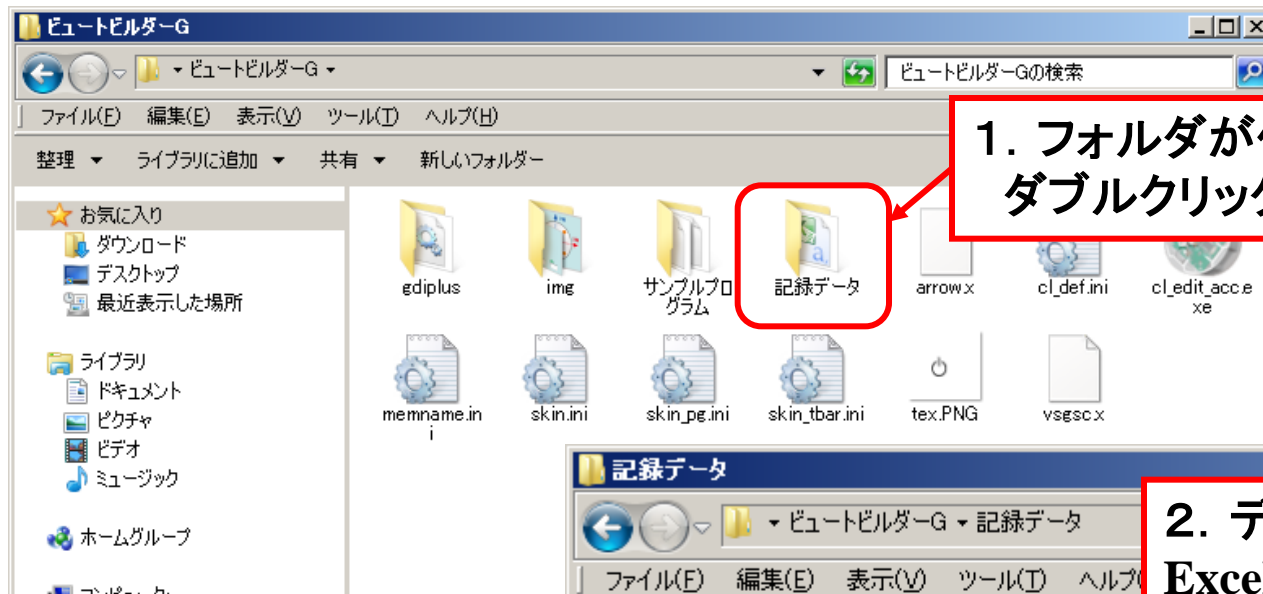
記録データというフォルダが作成され、データ* (ファイル名は変更可能) というファイルに一定時間毎にセンサ値が記録されます。

2. 記録開始をクリック

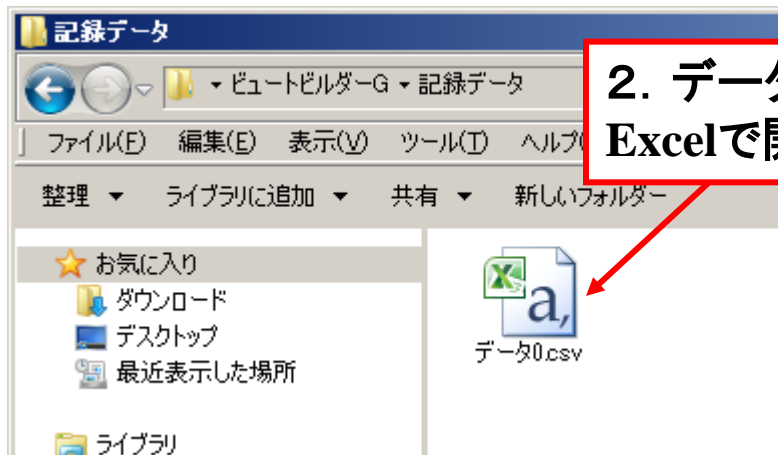
3. 記録の終了

センサの情報をパソコンに記録②

ビュートビルダーGのフォルダ内に記録データが保存されています。



1. フォルダが作成されている。
ダブルクリックします。

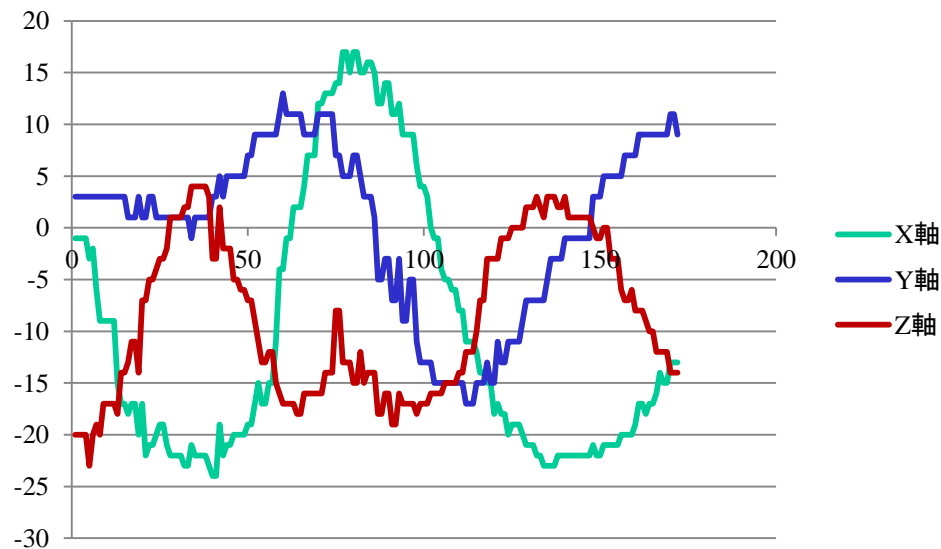


2. データファイルを開きます。
Excelで開くことができます。

記録データには、記録年月日/時刻と、加速度センサの数値が保存されています。

	A	B	C	D	E
1	年月日	時分秒	X軸	Y軸	Z軸
2	2013/3/19	12:018:073	-1	3	-20
3	2013/3/19	12:018:103	-1	3	-20
4	2013/3/19	12:018:133	-1	3	-20
5	2013/3/19	12:018:173	-1	3	-20
6	2013/3/19	12:018:203	-3	3	-23
7	2013/3/19	12:018:233	-2	3	-20
8	2013/3/19	12:018:263	-6	3	-19
9	2013/3/19	12:018:303	-9	3	-20
10	2013/3/19	12:018:334	-9	3	-17
11	2013/3/19	12:018:364	-9	3	-17
12	2013/3/19	12:018:404	-9	3	-17
13	2013/3/19	12:018:434	-9	3	-17
14	2013/3/19	12:018:464	-15	3	-18
15	2013/3/19	12:018:494	-17	3	-14
16	2013/3/19	12:018:534	-17	3	-14
17	2013/3/19	12:018:564	-18	1	-13
18	2013/3/19	12:018:595	-17	1	-11
19	2013/3/19	12:018:635	-17	1	-11
20	2013/3/19	12:018:665	-20	3	-14
21	2013/3/19	12:018:695	-17	1	-7
22	2013/3/19	12:018:725	-22	1	-7
23	2013/3/19	12:018:765	-21	3	-5
24	2013/3/19	12:018:795	-21	3	-5

グラフ化も可能です。



加速度の計測で、どのようなことができるでしょうか？

【人の動き】

本体を持つなどして、体操してみましよう。

→ どんな動きが取れるでしょうか？

※USBケーブルを無理に引っ張ったり、抜けないように注意して下さい。

【振り子・ヨーヨー・落下】

本体ケースの穴にひもを通して、振り子のように動かしたり、穴にゴムを通して弾ませてみましよう。

→きれいな波が現れるでしょうか？

※ひもは短めにして、ぶついたり落としたりしないように注意してください。

【乗り物の動き】

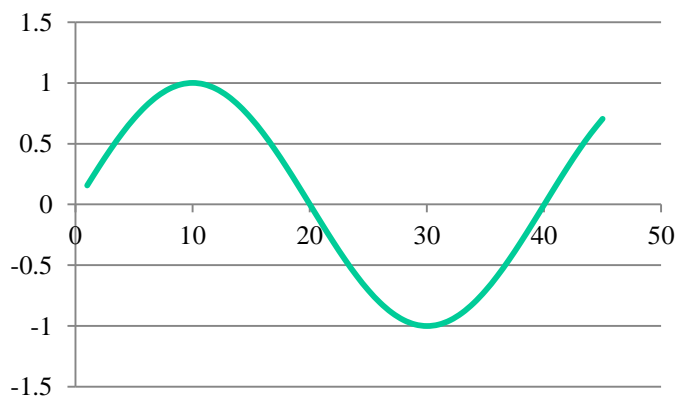
車などに乗ってデータを記録してみましよう(ノートPCなどを利用)。

→アクセル・ブレーキ・カーブなどに合わせて数値が変化するでしょうか？

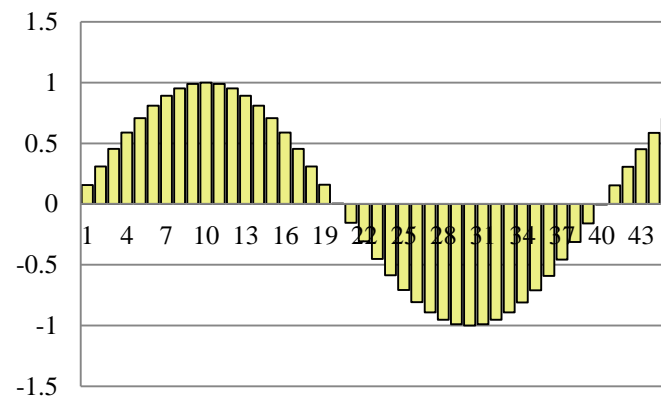
アナログ信号とデジタル信号について

センサ・アクチュエータの多くは、「**アナログ信号**」(電圧など)を使います。
一方、コンピュータは「**デジタル信号**」しか使えません。

そこで、間に**デジタルとアナログを変換する仕組み**を挟みます。



【アナログ信号】



【デジタル信号】

センサのアナログ信号 → **A/D変換** → デジタル信号
コンピュータのデジタル信号 → **D/A変換** → アナログ信号
コンピュータには、このように間で仲介を行う「**インターフェース**」を挟みます。

【メモ】

VstoneTM