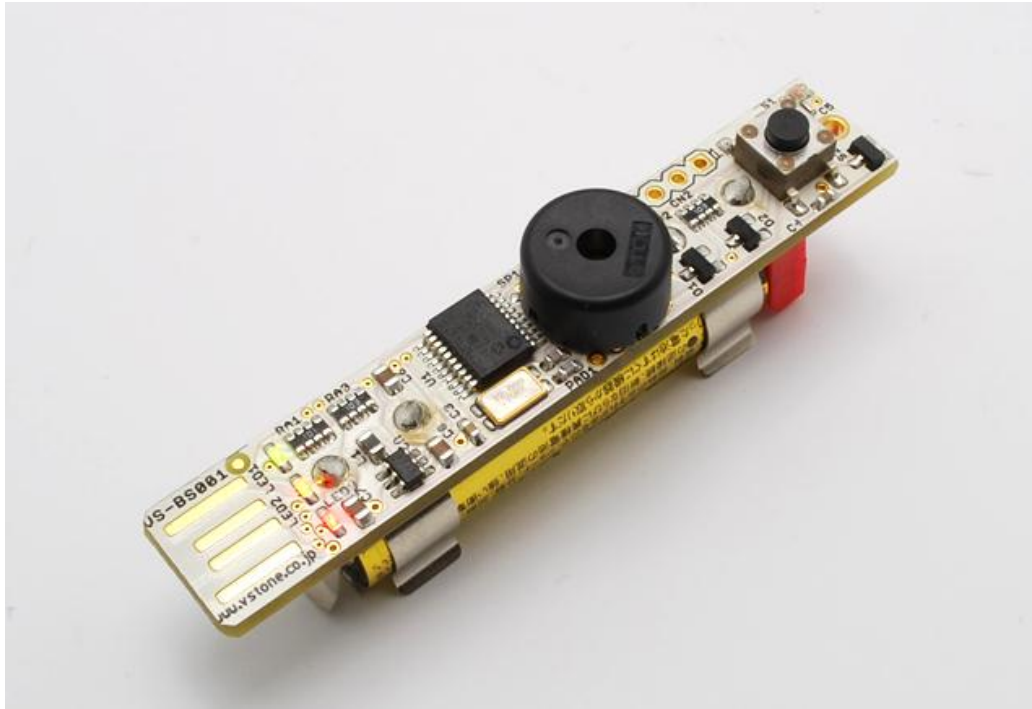
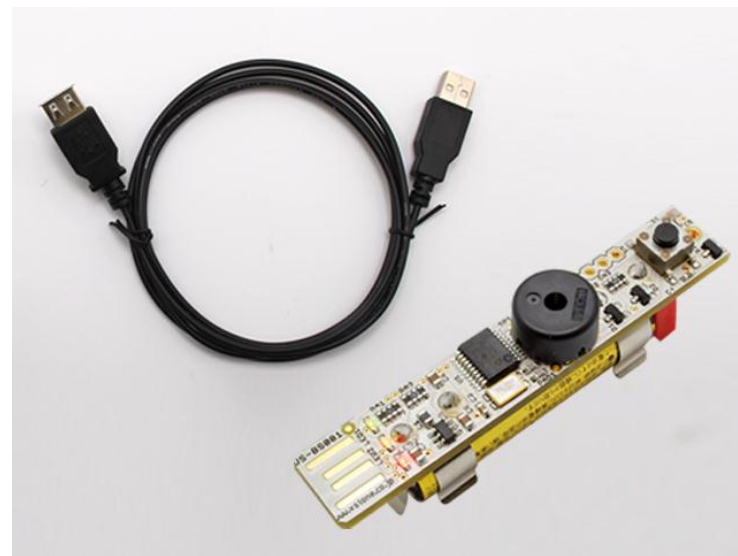


情報とコンピュータ ～プログラムによる計測と制御～



- コンピュータで制御されているものを見つけよう
- 計測制御プログラマー、動作確認
- ソフトウェアの操作方法の説明
- 順次処理の課題に挑戦



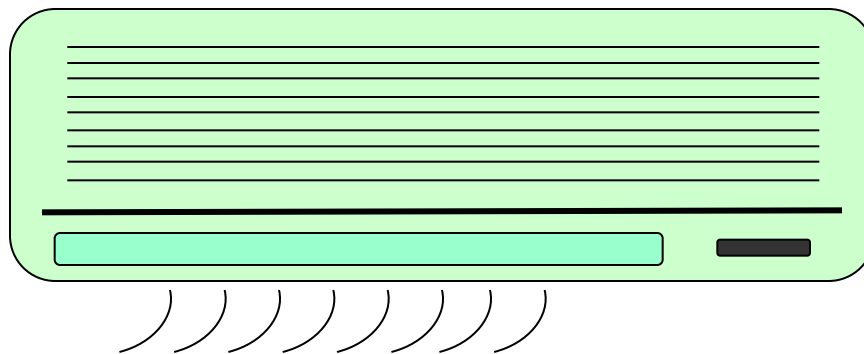
コンピュータは身の回りのいろいろなところで使用されています。

たとえば、エアコン

部屋の温度を感知
(入力)

設定よりも暑いので
冷房を強めよう
(判断・命令)

冷房を強める
(出力)



部屋の室温を感知(計測)し、
コンピュータで判断し
エアコンの働きを制御している。

コンピュータで制御されているものを探してみましょう。

- 自動ドア
 - 人を感知し、扉の開閉を行なう。
- 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る。

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る。**

エアコンの場合

部屋の温度を感知
(入力)

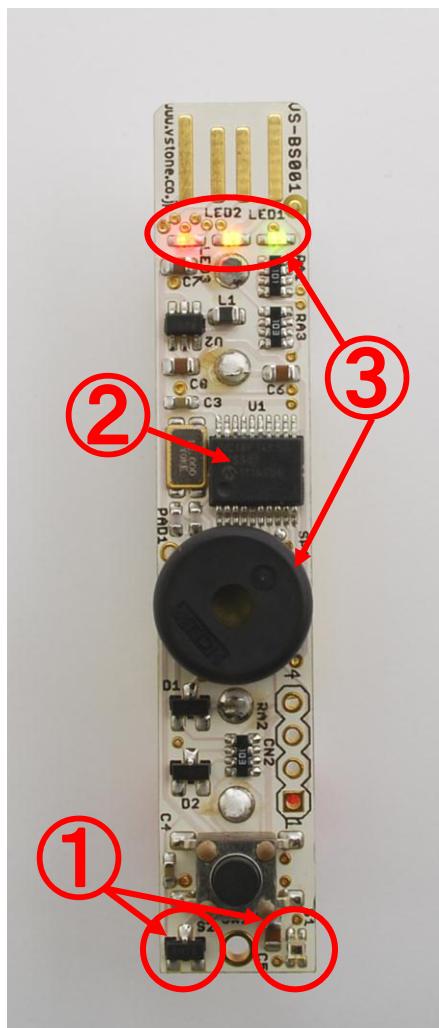


設定よりも暑いので
冷房を強めよう
(判断・命令)



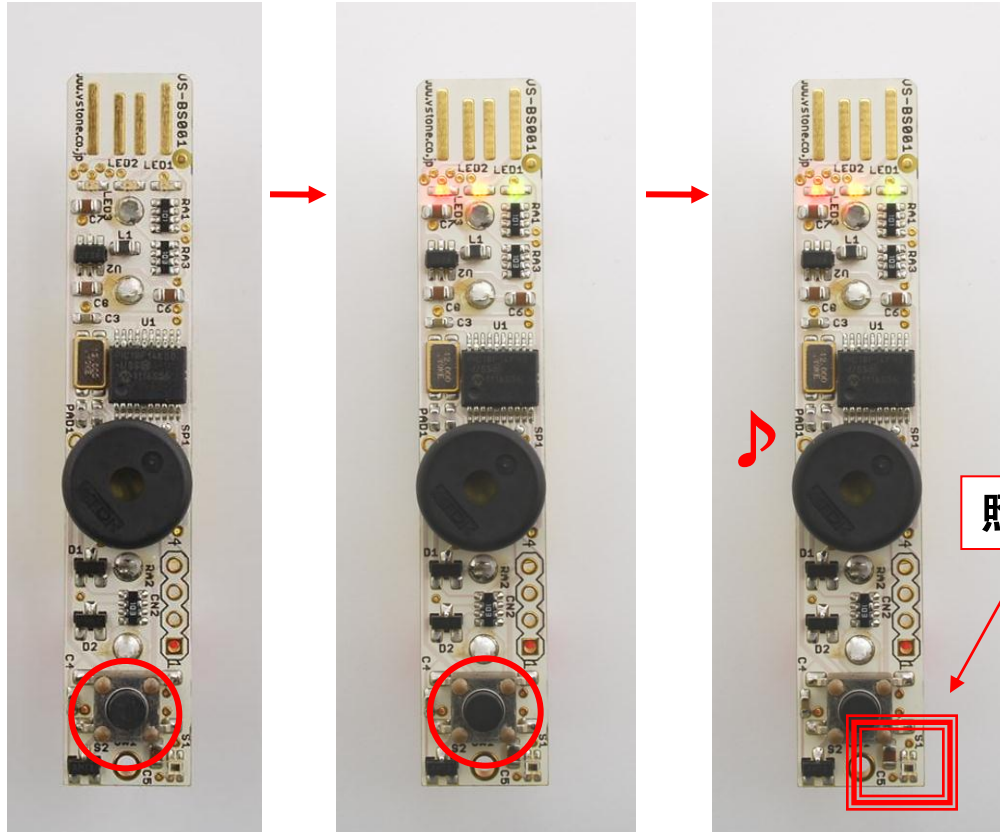
冷房を強める
(出力)

- ①. センサ
状況を感知する。
→温度センサ、光センサ、傾斜センサ
- ②. コンピュータ
状況を判断し、命令を行う。
- ③. アクチュエータ
動作・仕事を行う。
→クーラー、ヒーター、モーター
ブザー、LED



計測制御プログラマー

- ①. **センサ** = 照度センサ、温度センサ
明るさや温度の状況を見る。
- ②. **コンピュータ** = PICマイコン
命令を覚えたり、行動を考える。
- ③. **アクチュエータ** = ブザー、LED
音を出す。
点灯・点滅する。



プログラムの実行方法

0. 電池を入れます。
1. ボタンを押します。
スタンバイモードになります。
2. もう一度、ボタンを押します。
プログラムが実行します。

照度センサ

出荷時のプログラムでは、明るい場合、LEDが点灯し、暗い場合(照度センサを手で覆う、箱の中に入れる等)、ブザーから曲が流れるようになっています。

- ①ボタンを押す →電源がON
②LEDが全て点灯 →もう一度ボタンを押す
③プログラムが開始

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P」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 'ビュートビルダーP' (Vstone Builder P) software interface. The window title is 'ビュートビルダーP [LEDサンプル.vbd] ワ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(O)', 'データ記録(L)', '2進数の学習(B)', 'レポートの作成(R)', and 'ヘルプ(H)'. The toolbar contains icons for '新規', '開く', '保存', '戻る', '進む', '切取', '北', '貼付', '書込', '実行', '停止', and '記録'. A status bar shows '接続中' and 'テンポ 150'. The main workspace displays a flowchart starting with a '開始' block, followed by three 'LED' blocks (red, yellow, green) each with a '1.00秒' duration. This is followed by three 'LED' blocks (red, yellow, green) each with a '0.50秒' duration. Then, there are two 'LED' blocks (grey 'LED OFF' and green 'LED') each with a '2.00秒' duration. The flowchart ends with a '終了' block. On the left side, there are control panels for 'LED' (with a musical note icon), 'IF', 'RANDOM', and 'LOOP LOOP'. Below these is a 'タイマー' (Timer) section showing '0分0.0秒' and sensor data: '温度センサ 086', '湿度センサ 25.5 °C', 'ボタン OFF', and '照度センサ 127'. A note at the bottom left states '※設定する項目は ありません'.

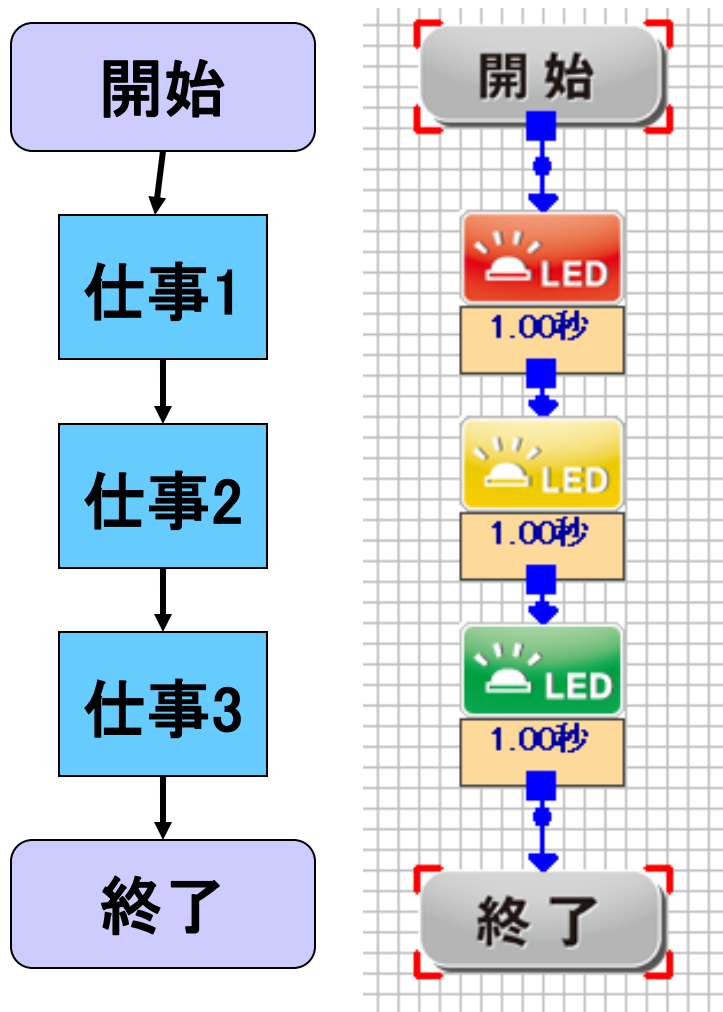
これは、LEDの点灯を行うプログラムです



プログラミングには主に

- ・順次
- ・繰り返し
- ・分岐

の3つの手順があります。



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。

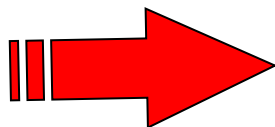
ソフトウェア「ビュートビルダーP」の起動



サンプルプログラム



cl_edit_stk.exe
Vstone



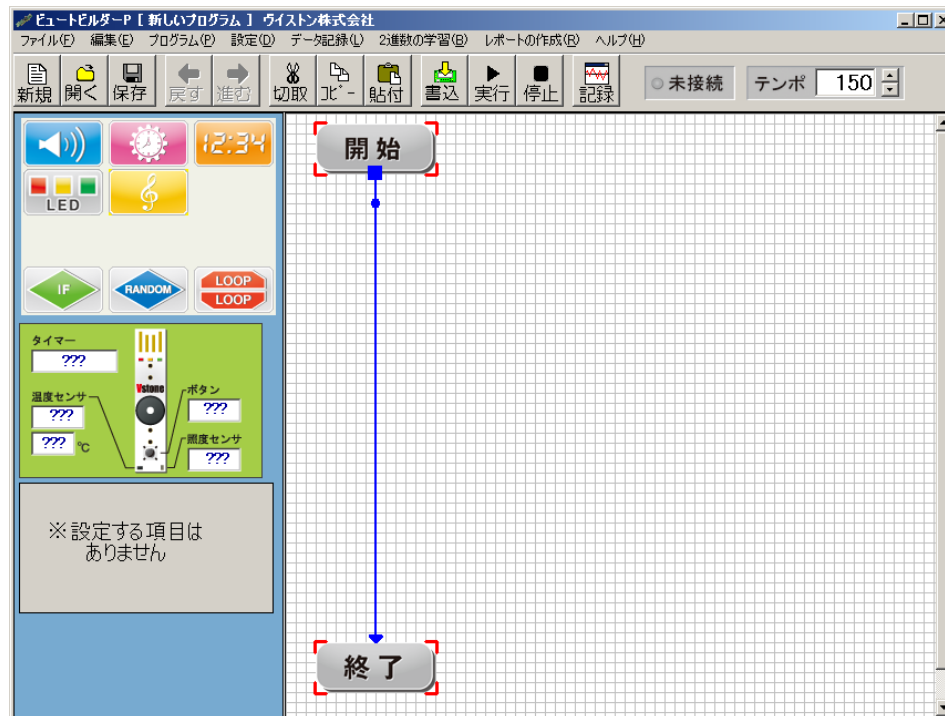
skin.ini
構成設定
1 KB

ダブル
クリック!

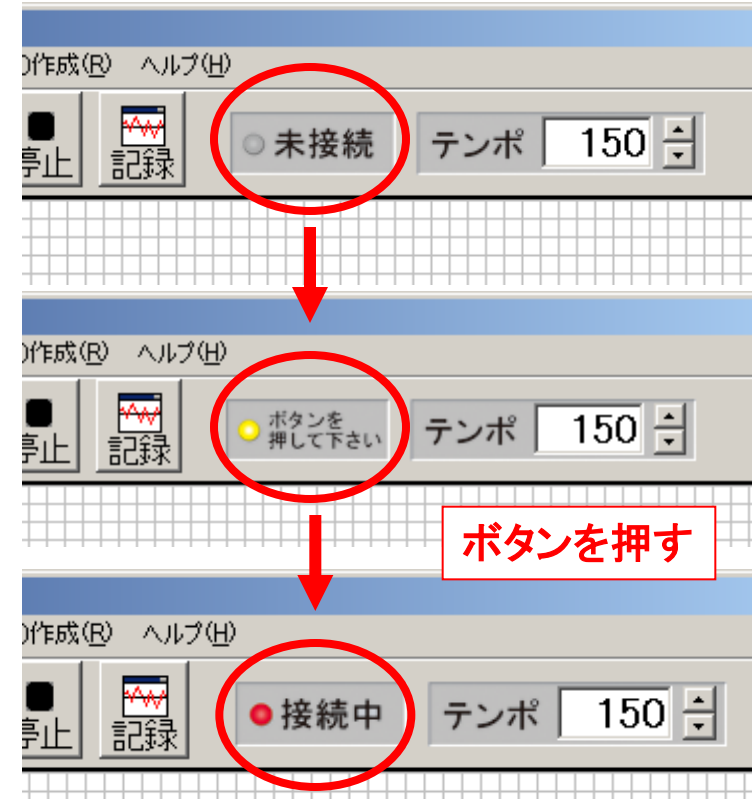
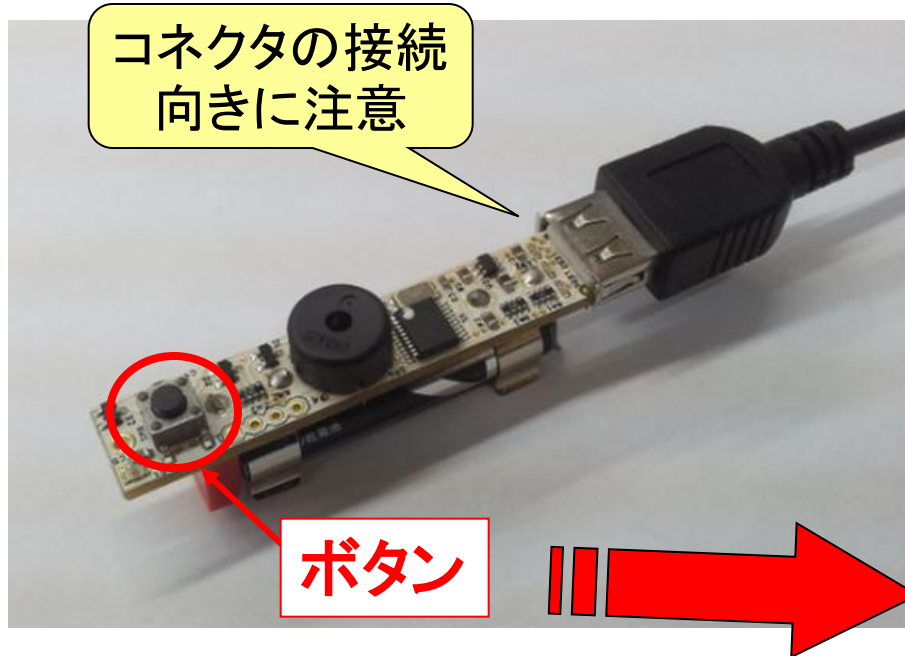


skin_tbar.ini
構成設定
1 KB

ビュートビルダーPフォルダ内の
「cl_edit_stk.exe」を
ダブルクリックします。



ビュートビルダーPの画面



※計測制御プログラマーをケーブルにつないだ状態でグラグラ動かしたりすると、ケーブルの接触が不安定になって通信が途切れることがあります。

順次のプログラミング 「ド」と鳴らす

The screenshot displays the Vstone software interface for programming a sequence. The window title is "ビュートビルダーP [新しいプログラム] ラ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)", "データ記録(L)", "2進数の学習(B)", "レポートの作成(R)", and "ヘルプ(H)". The toolbar contains icons for "新規", "開く", "保存", "戻す", "進む", "切取", "北", "貼付", "書込", "実行", "停止", and "記録". A status bar shows "接続中" and "テンポ 150".

The main workspace is a grid with a flowchart. It starts with a "開始" (Start) block, followed by a "ド" (Do) note block, and ends with a "終了" (End) block. The "ド" block is highlighted with a red dashed box. The left sidebar contains various controls: a speaker icon, a clock showing "12:34", an "LED" indicator, a treble clef icon, "IF" and "RANDOM" logic blocks, "LOOP" blocks, a "タイマー" (Timer) set to "0分0.0秒", a "温度センサ" (Temperature Sensor) showing "093" and "30.0℃", a "ボタン" (Button) set to "OFF", a "照度センサ" (Illuminance Sensor) showing "126", a dropdown menu set to "普通の", a "4分 音符" (4-measure note) dropdown, and a checked checkbox "鳴り終わるまで待つ" (Wait until finished).

プログラムの作り方

コンピュータビルダーP [新しいプログラム] ウイストーン株式会社 *

ファイル(F) 編集(E) プログラム(P) 設定(D) データ記録(L) 2進数の学習(B)

新規 開く 保存 戻す 進む 切取 貼付 書込

開始

終了

クリックで選択!



プログラム] ウイストーン株式会社 *

設定(D) データ記録(L) 2進数の学習(B)

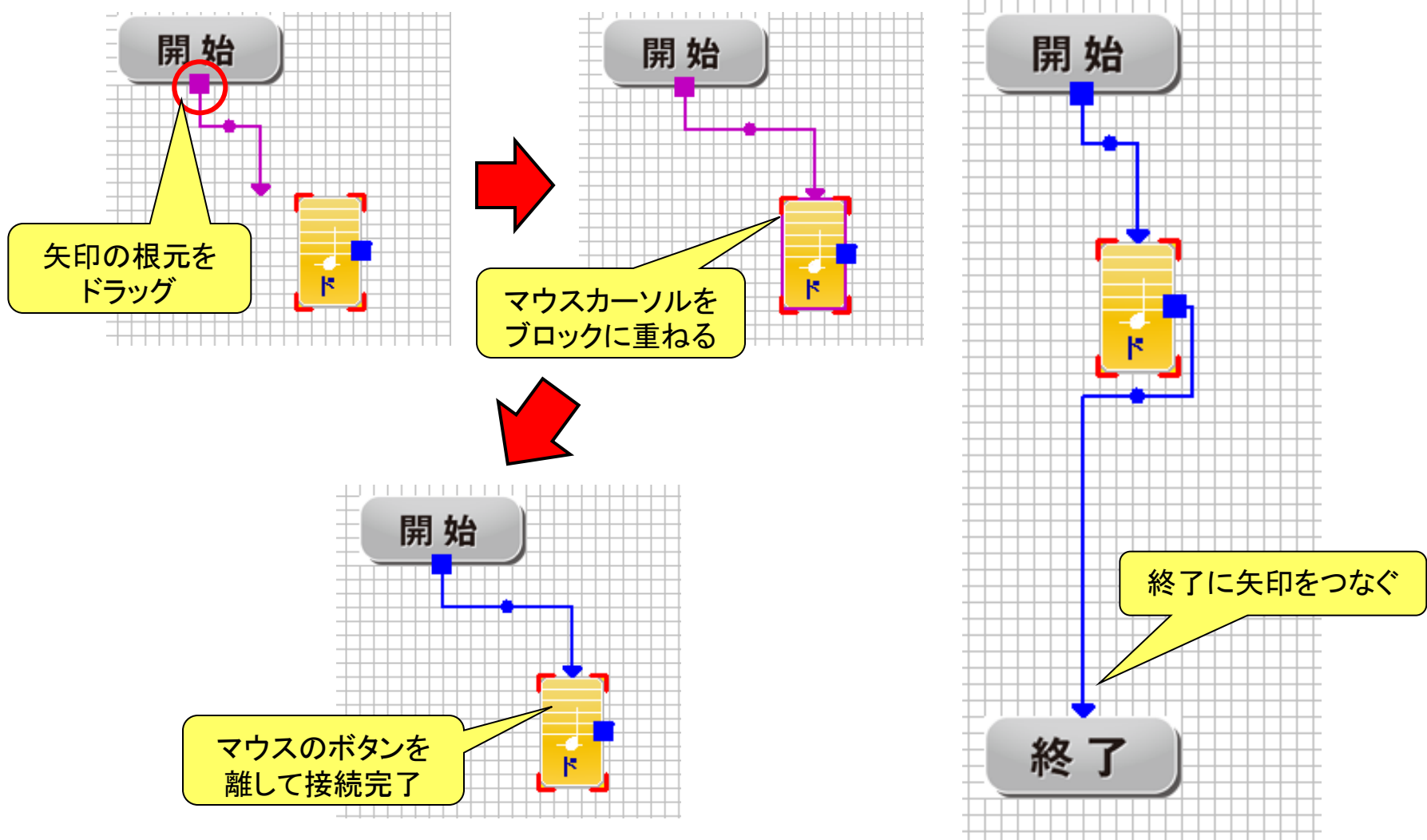
進む 切取 貼付 書込

開始

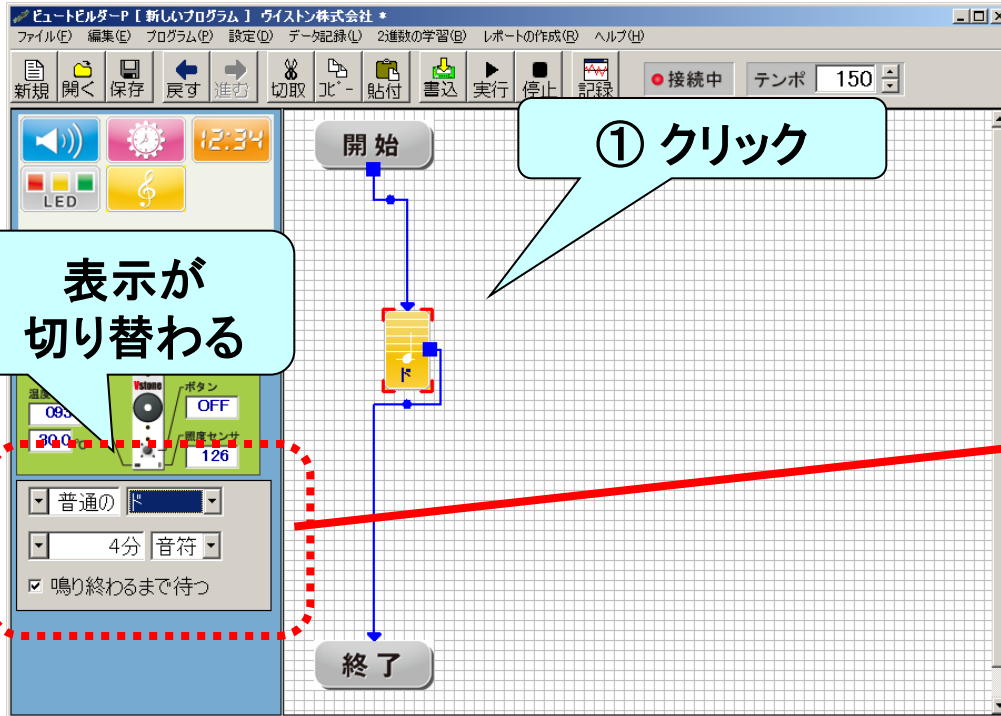
終了

クリックするとブロックを追加!

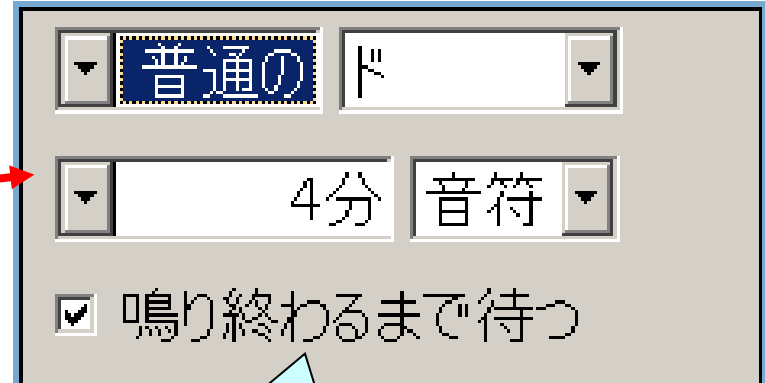
【例題】「ド」のプログラムの作り方



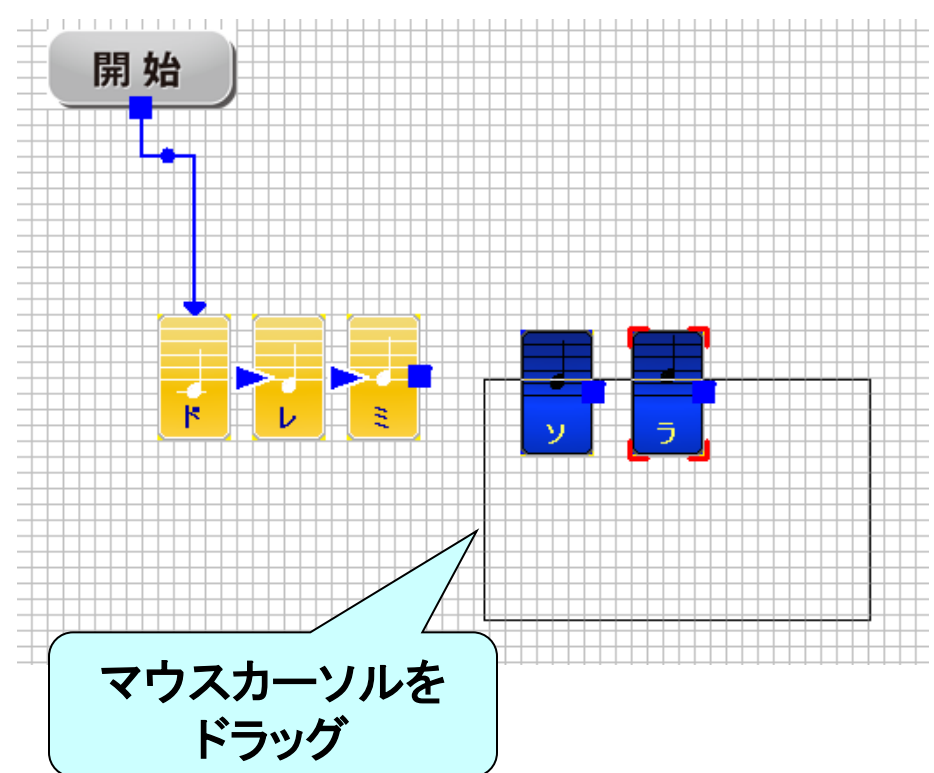
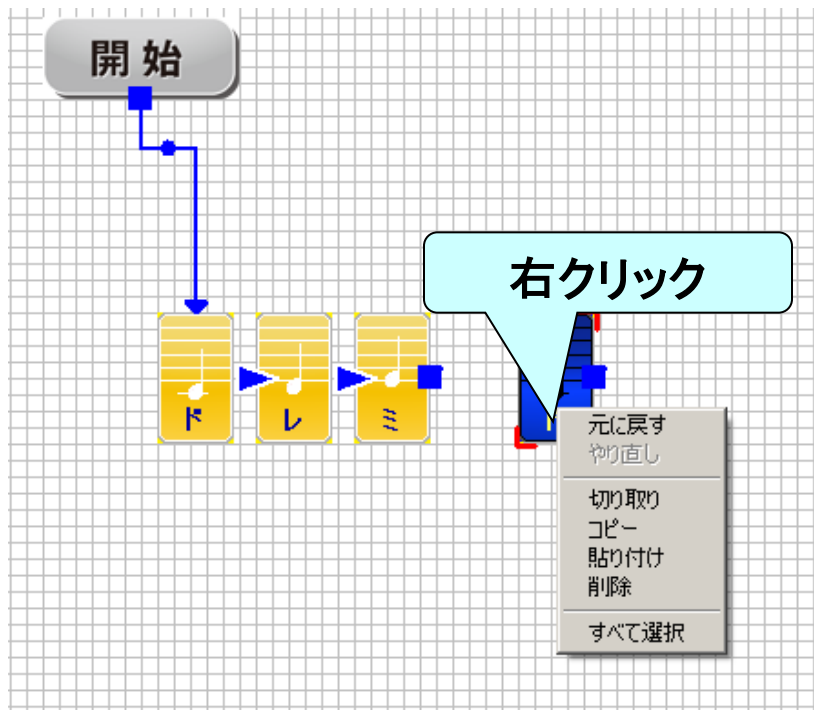
音符のプログラム 設定の変更



ブロックをクリックすると、左下の「設定エリア」の表示が変わります。



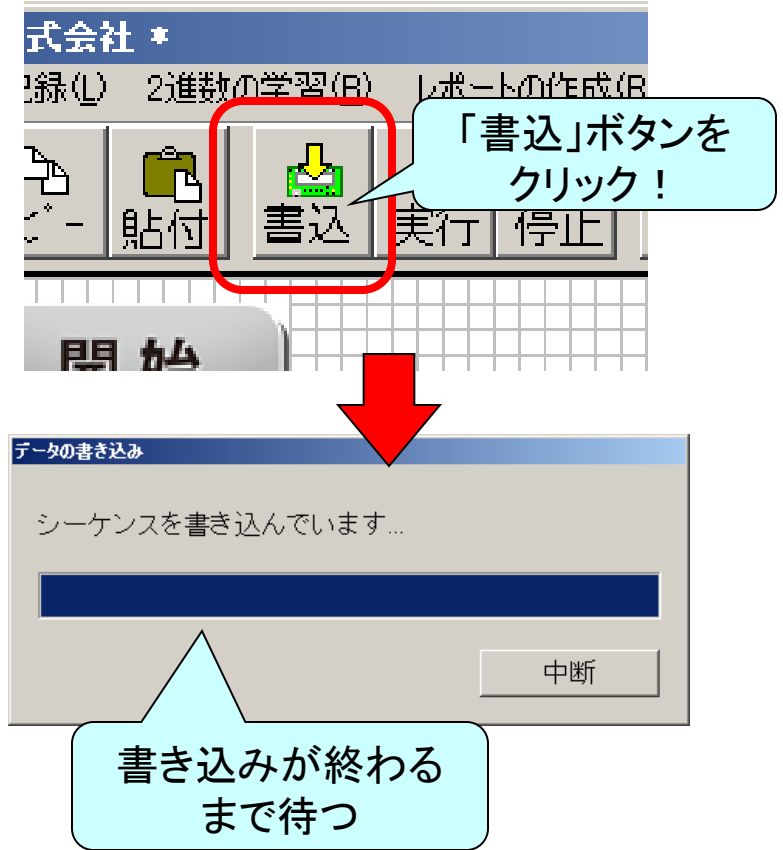
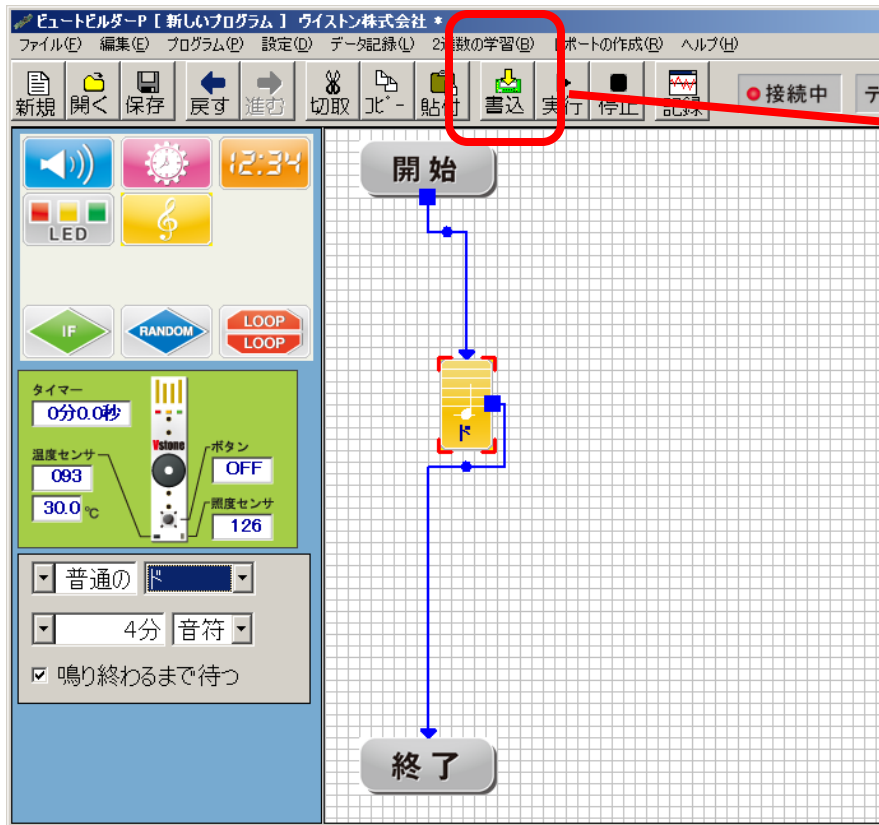
ブロックごとに命令の設定を
変更することができます。



削除したいブロック上で、右クリックし、削除を選択する。

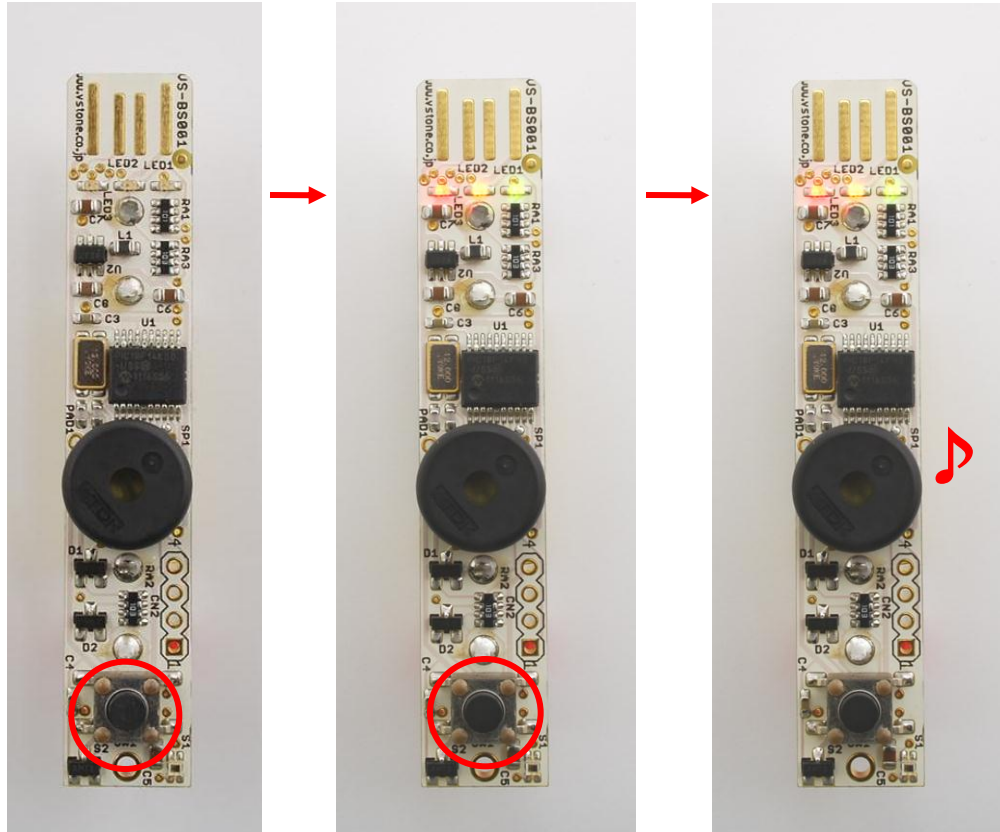
プログラムエリアでカーソルをドラッグすると複数のブロックを選択することができます。選択後、Deleteボタンを押すとブロックを削除することができます。

プログラムの書き込み



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。
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す

USBより計測制御プログラマーを抜きます。



プログラムの実行方法

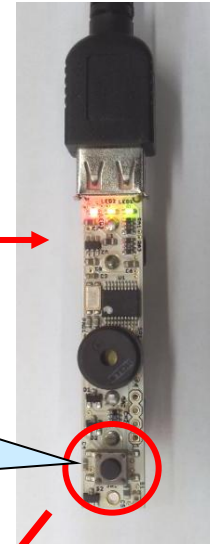
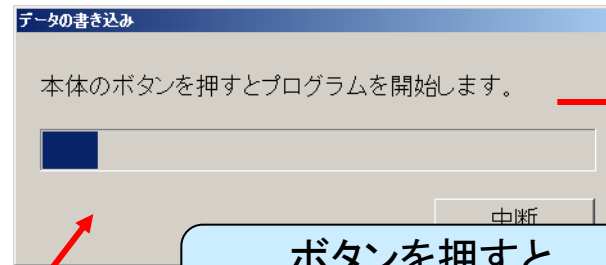
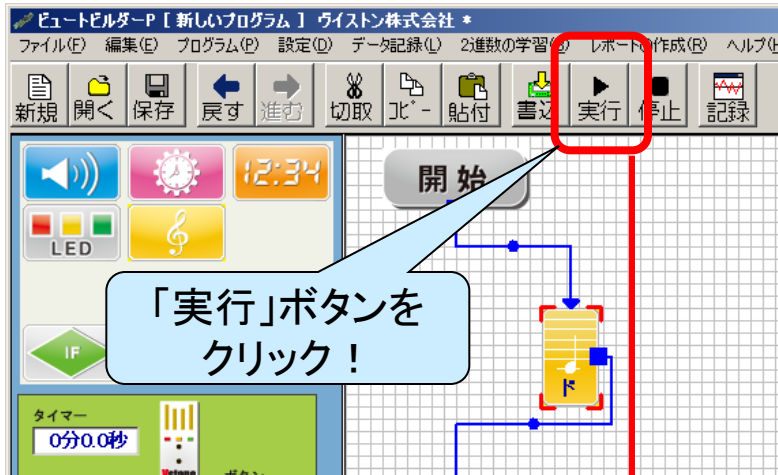
0. 電池を入れます。
1. ボタンを押します。
スタンバイモードになります。
2. もう一度、ボタンを押します。
プログラムが実行します。

プログラムが終了すると
スタンバイモードになります。

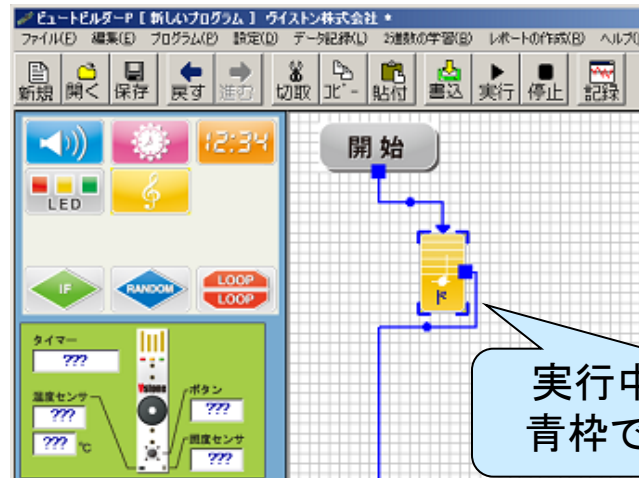
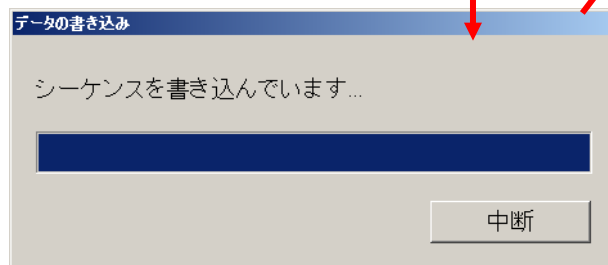
- ①ボタンを押す →電源がON
- ②LEDが全て点灯 →もう一度ボタンを押す
- ③プログラムが開始

プログラムのテスト実行

PC と接続してプログラムをテスト実行すると、実行中の命令が青枠で囲まれて表示するため、プログラムの進み方が常に確認できます。

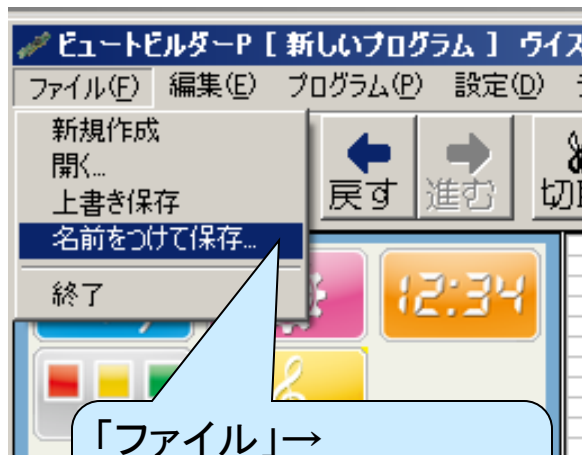


ボタンを押すと
テスト実行を開始

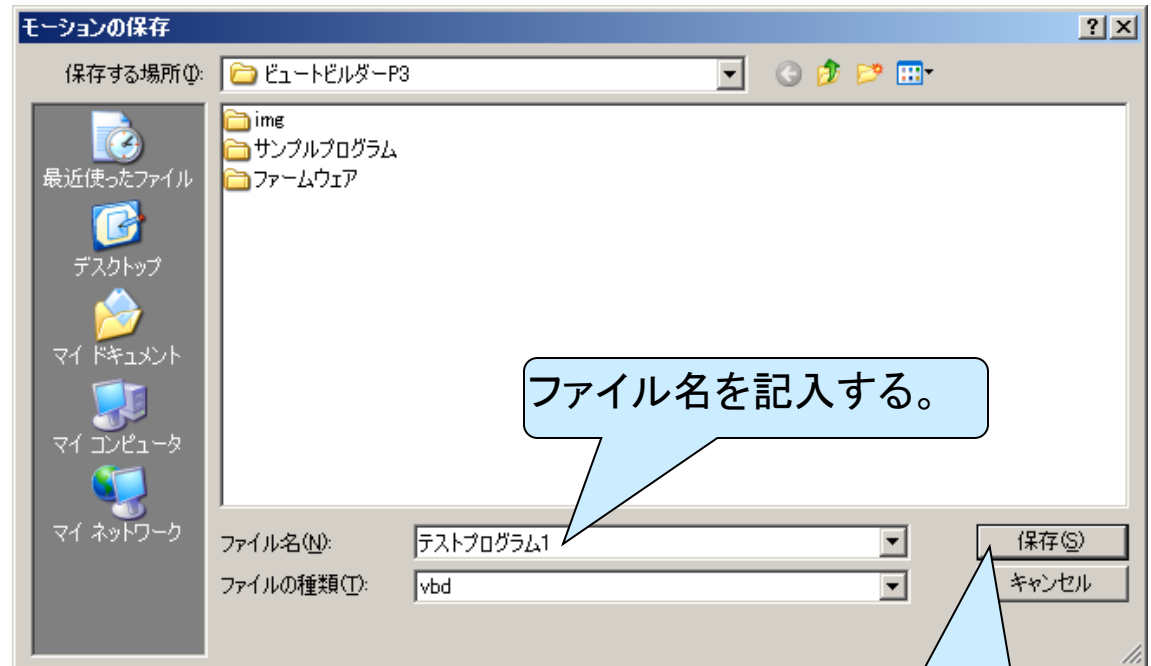


実行中の命令が
青枠で囲まれる。

プログラムファイルの保存



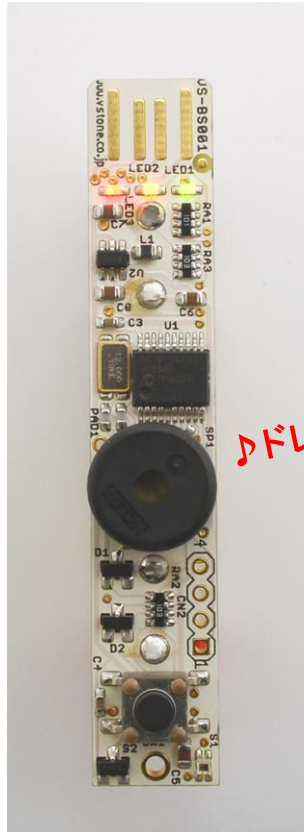
「ファイル」→
「名前をつけて保存」を
選択する。



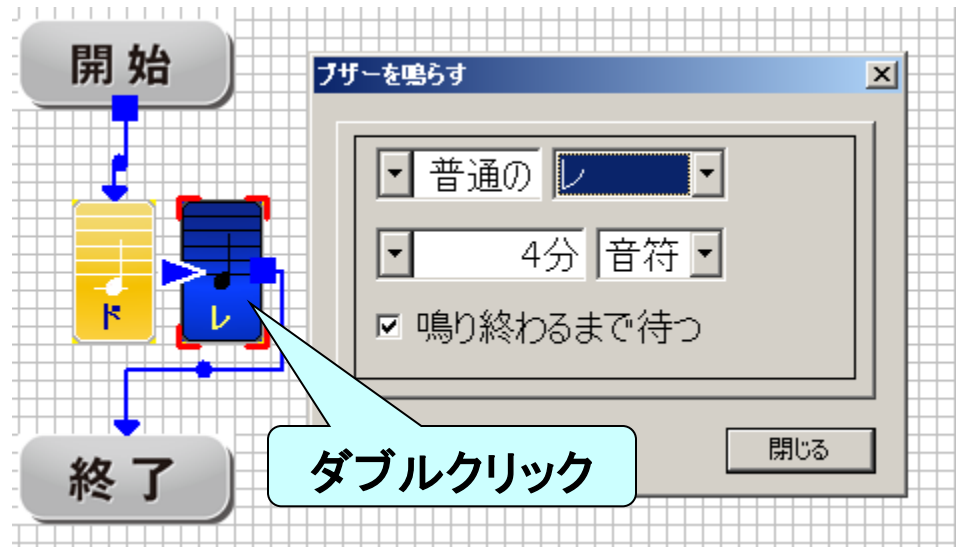
ファイル名を記入する。

保存をクリックする。

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。)

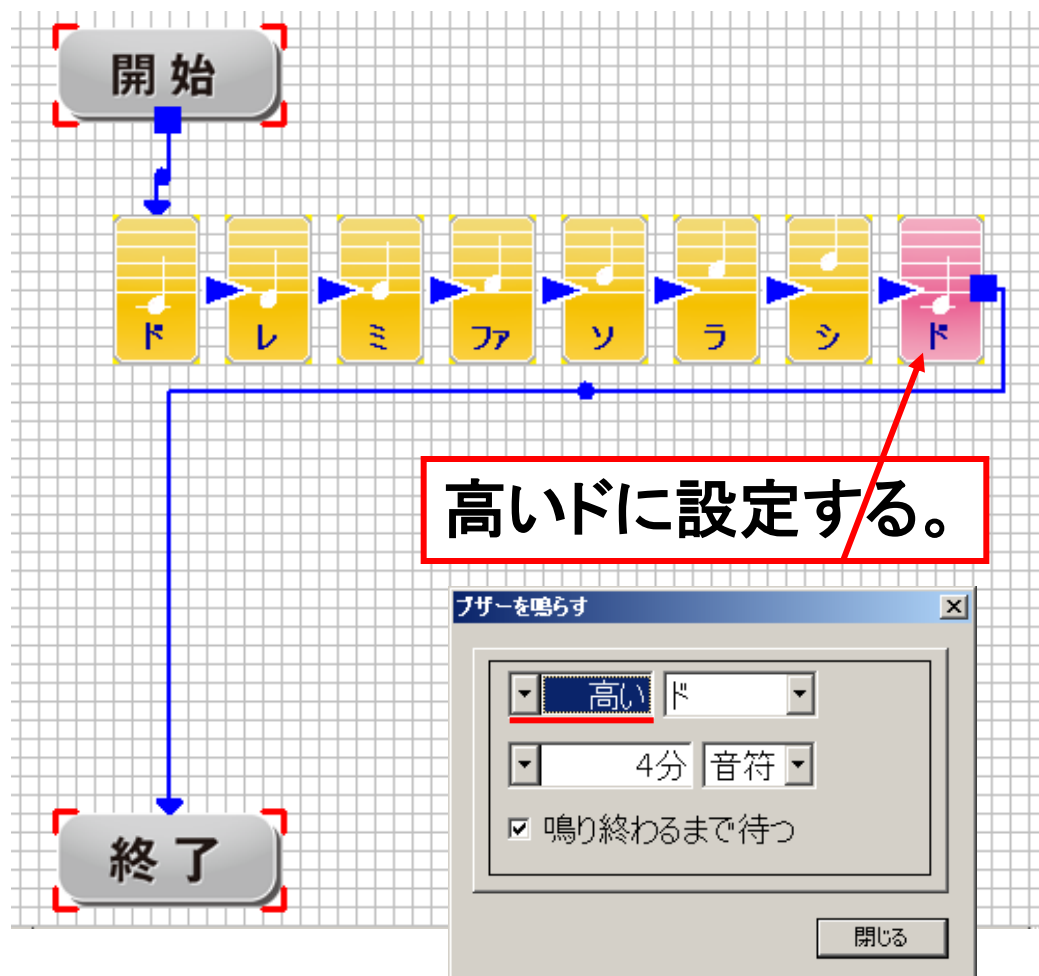


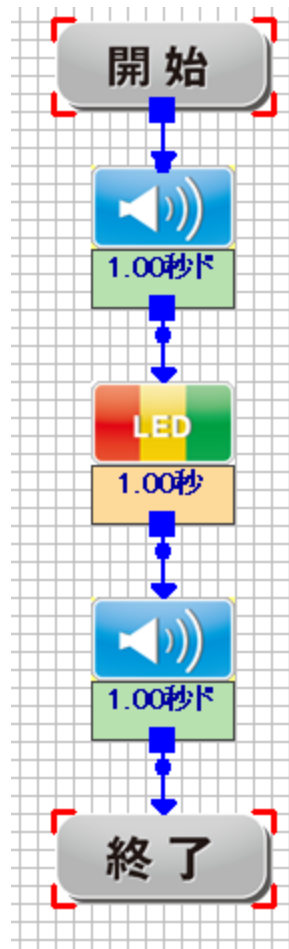
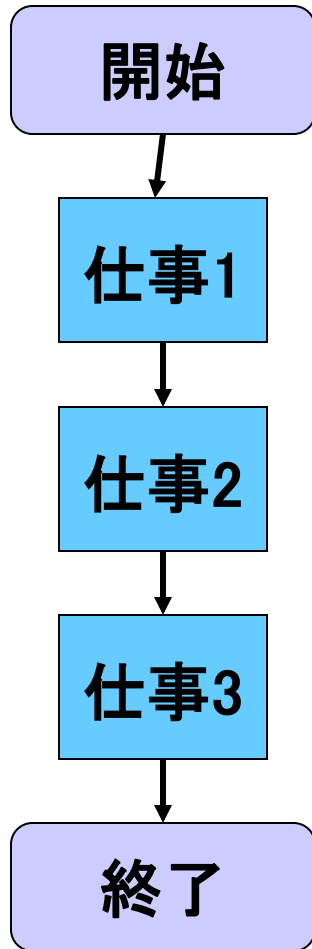
音符ブロックは、ブロックをダブルクリックすると
ブロック近くに設定を表示させることができます。

ドレミファソラシド(高)と鳴らすプログラム



♪ドレミファソラシド♪



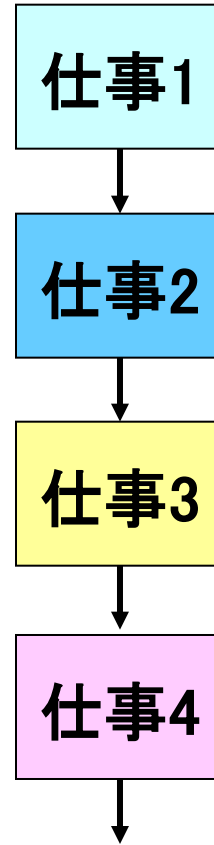


身の回りにはコンピュータで制御されている機器がたくさんあります。

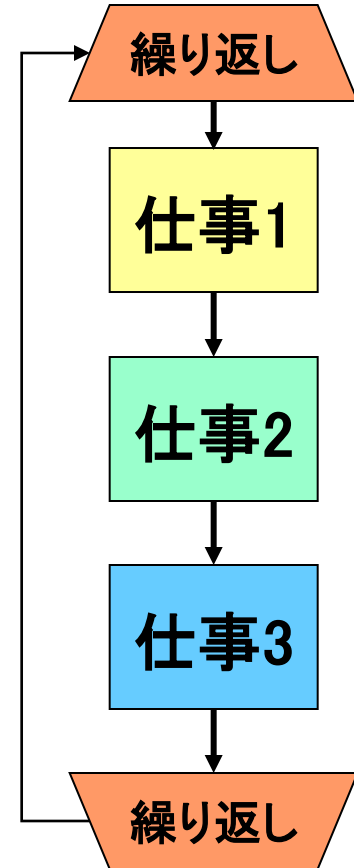
それらの機器は、**センサ・コンピュータ・アクチュエータ**という要素を持っています。

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。

- LEDのプログラム
- 繰り返しのプログラム



【順次】



【繰り返し】

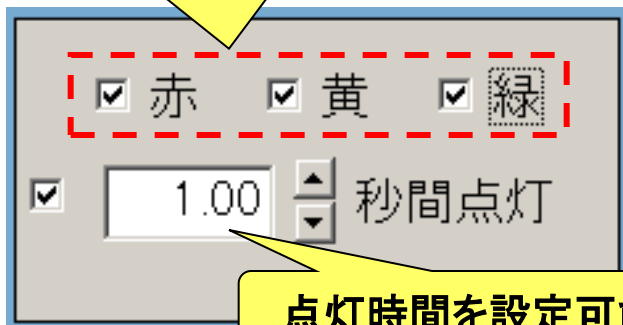
LEDの命令ブロックについて

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。



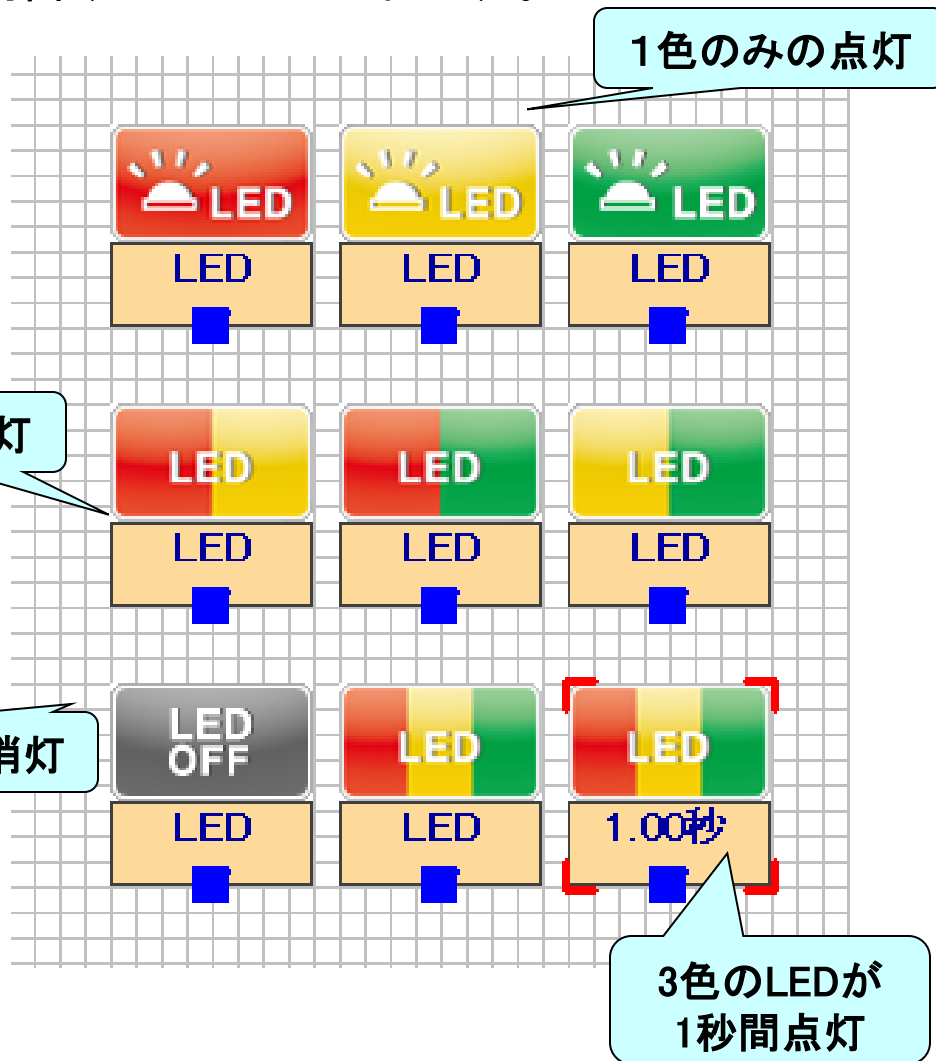
LED命令ブロック

チェックを付けた色が点灯

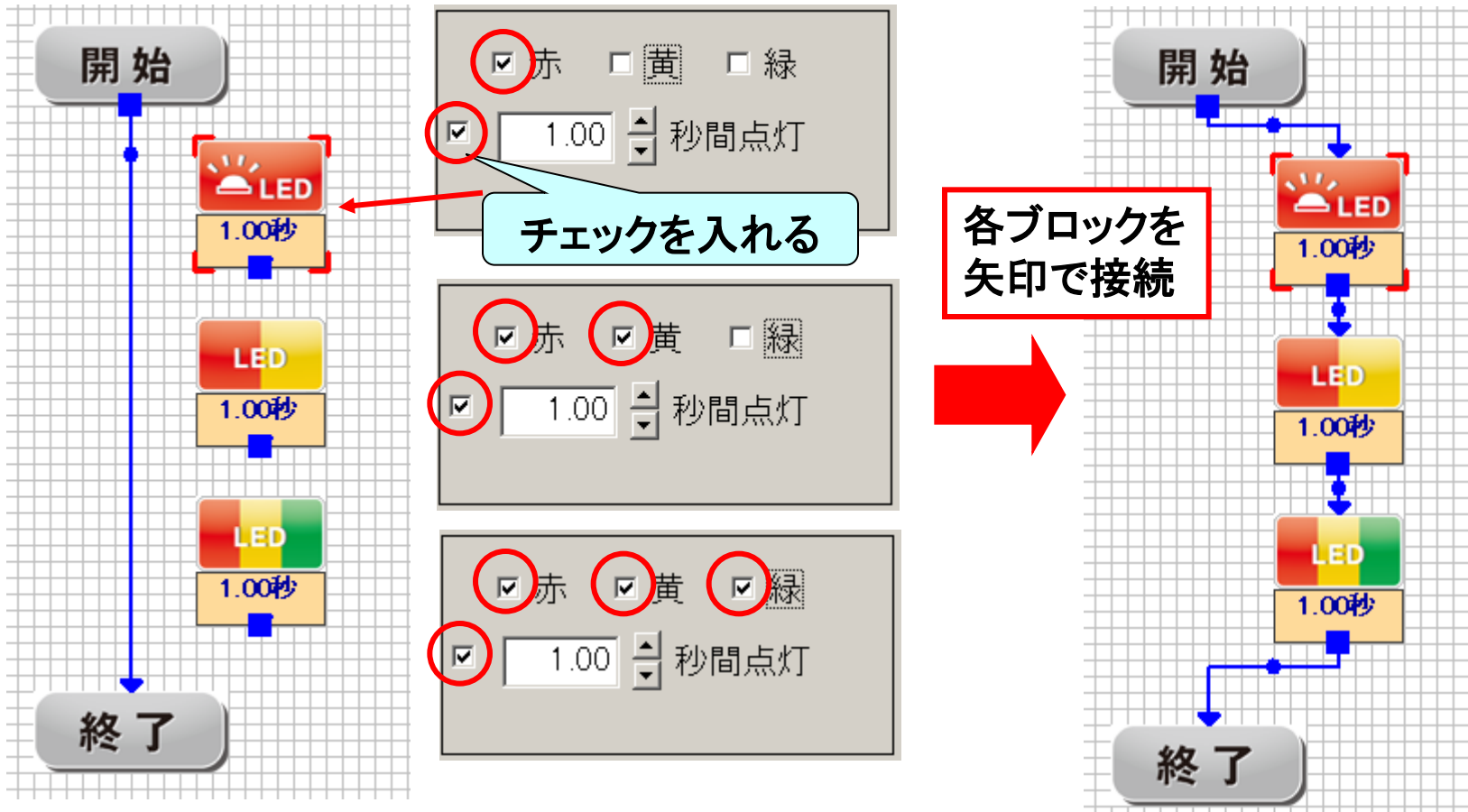


点灯時間を設定可能

<LEDブロックの設定>

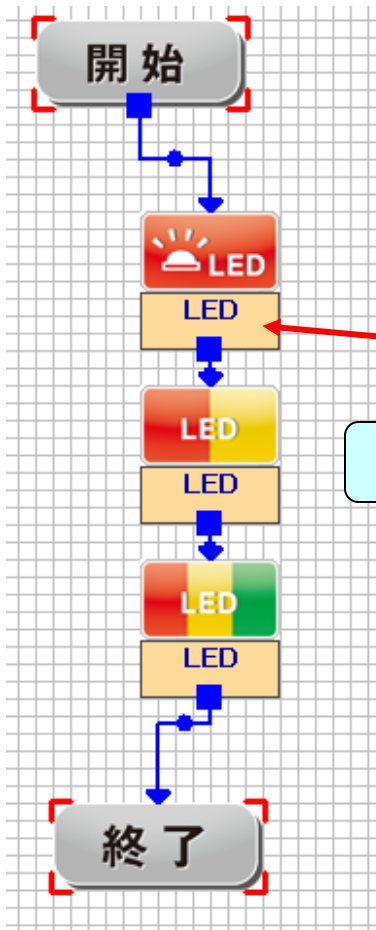


赤1秒、赤+黄色1秒、すべて点灯1秒を行うプログラムを作成しよう



プログラムが完成したら、書込みを行い 実行してみよう。

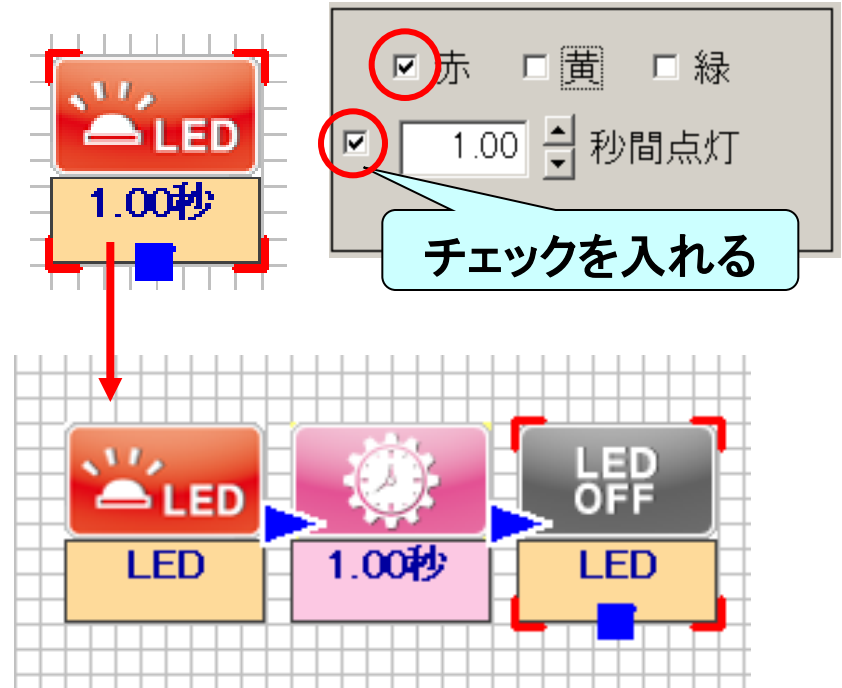
下記のようなプログラムを実行した場合、
赤→赤・黄→赤・黄・緑 というような点灯にはならない。



赤 黄 緑
 1.00 秒間点灯

チェックを入れない

命令のみのブロックになり、
すぐに次の命令ブロックに
移行する。
左のようなプログラムでは、
すぐにプログラムが終了し、
実行後すぐにスタンバイ状
態になってしまう。



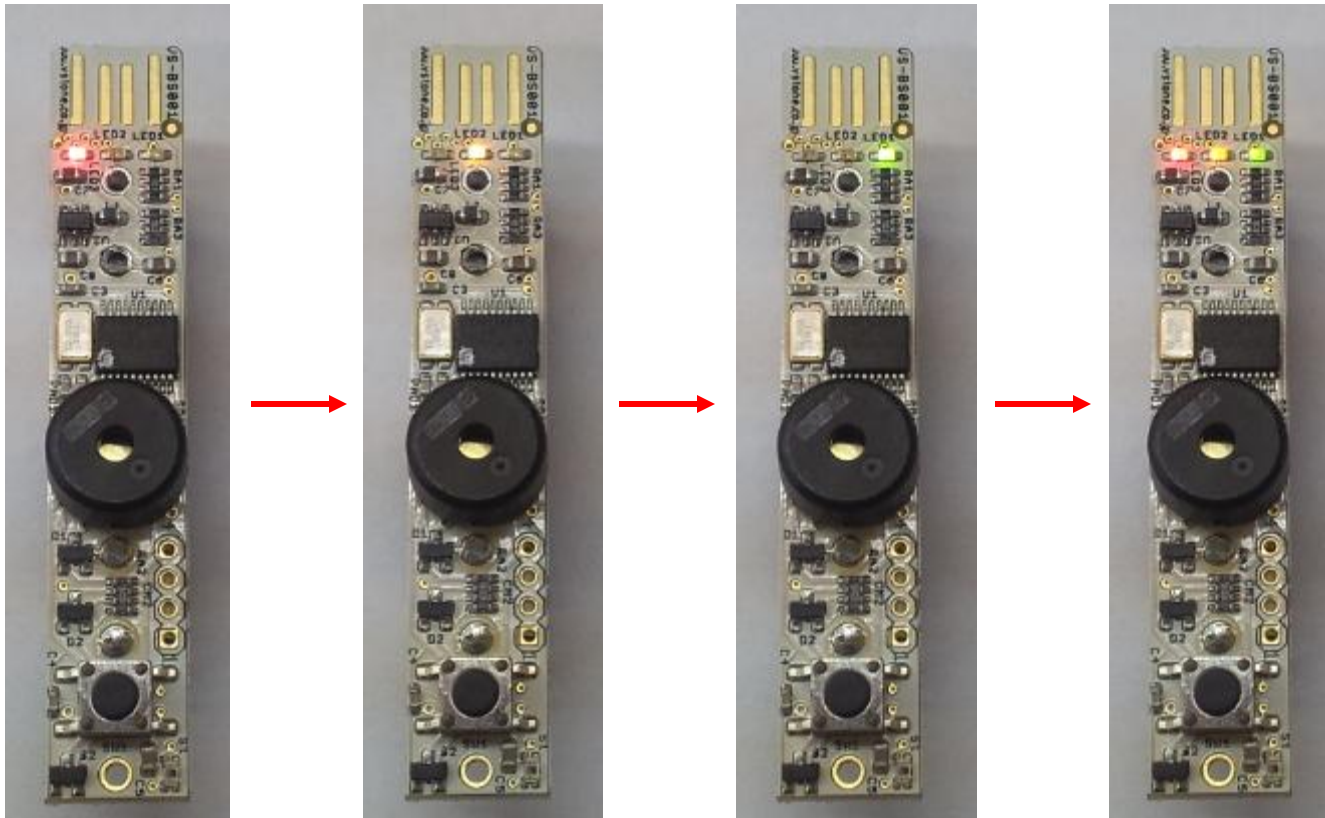
赤 黄 緑
 1.00 秒間点灯

チェックを入れる

赤LED(1秒間点灯)というブロックは、3つの命令
ブロックを合わせたものと同じ内容になっている。
赤LED点灯→1秒間継続→LED消灯

課題2 赤1秒・黄1秒・緑1秒・すべて点灯1秒行うプログラム **Vstone**TM

プログラムをスタートしたら、赤1秒間点灯→黄1秒間点灯→
緑1秒間点灯→すべて点灯1秒間を行うプログラムを作りましょう。

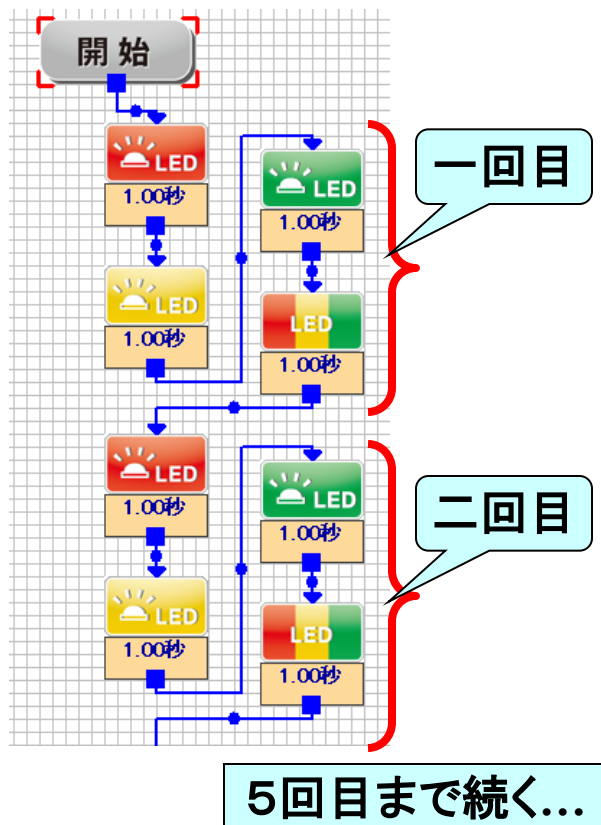


プログラムをスタートしたら、赤1秒間点灯→黄1秒間点灯→緑1秒間点灯→すべて点灯1秒間を行うプログラムを作りましょう。



・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？

「赤1秒間点灯→黄1秒間点灯→緑1秒間点灯→すべて点灯1秒間」を5回繰り返すプログラムを作成しよう。

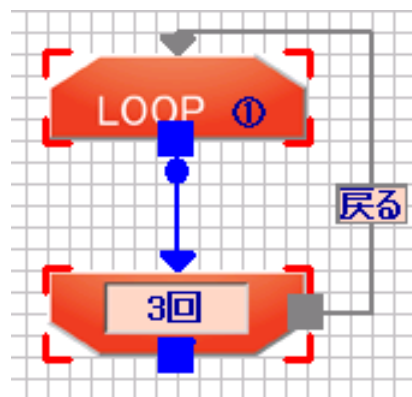


同じ命令を5回分並べると作成できますが、手間がかかりプログラムも見づらくなります。また、もし10回繰り返したい場合は、更に手間がかかります。

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。



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

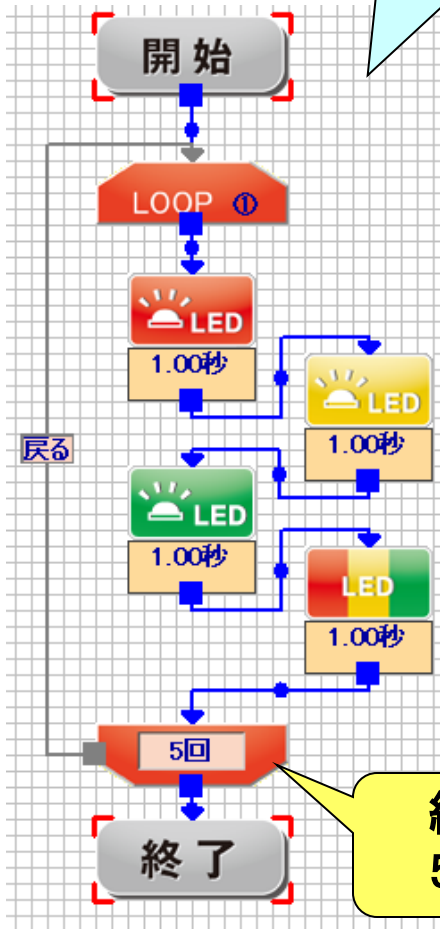


繰り返し回数を設定

- 3 回くりかえす
- ずっとくり返す(無限)

繰り返しブロックで挟まれた命令を設定した回数だけ繰り返します。

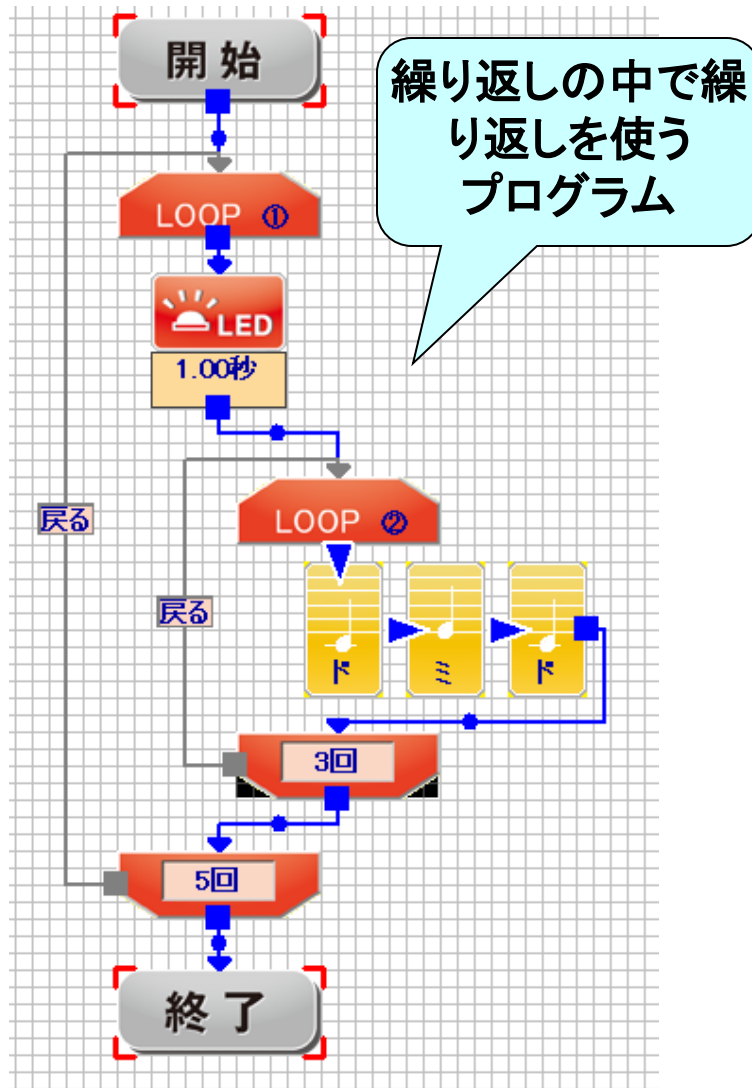
課題2を5回繰り返すプログラム



課題2で作成したプログラムに
繰り返しブロックを追加してみよう。

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てみよう。

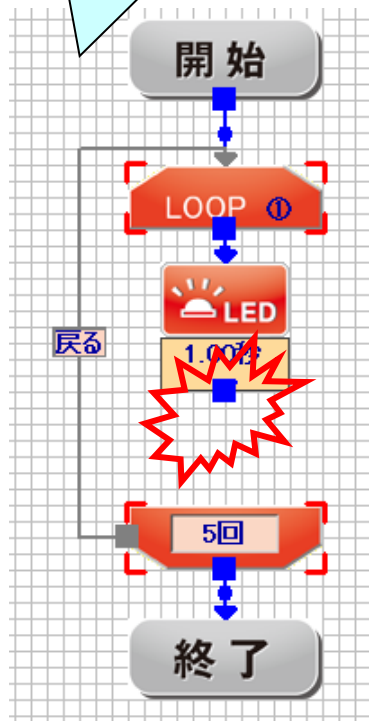
繰り返しは
5回に設定



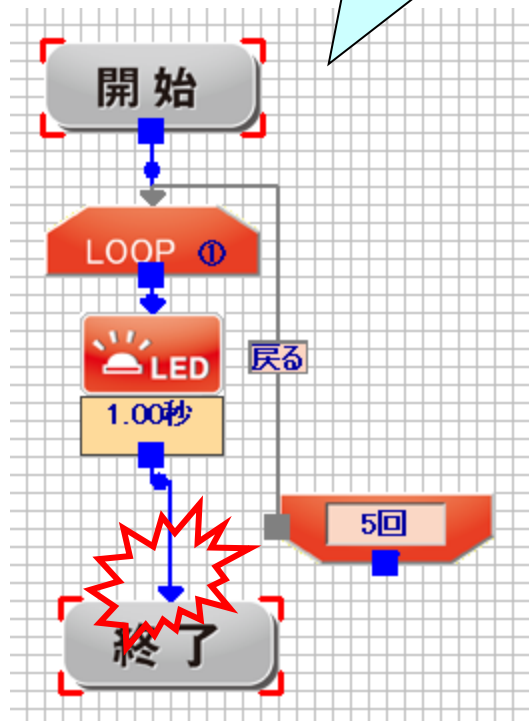
「繰り返し」の中に「繰り返し」を使うというような複雑な構造もできます。

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

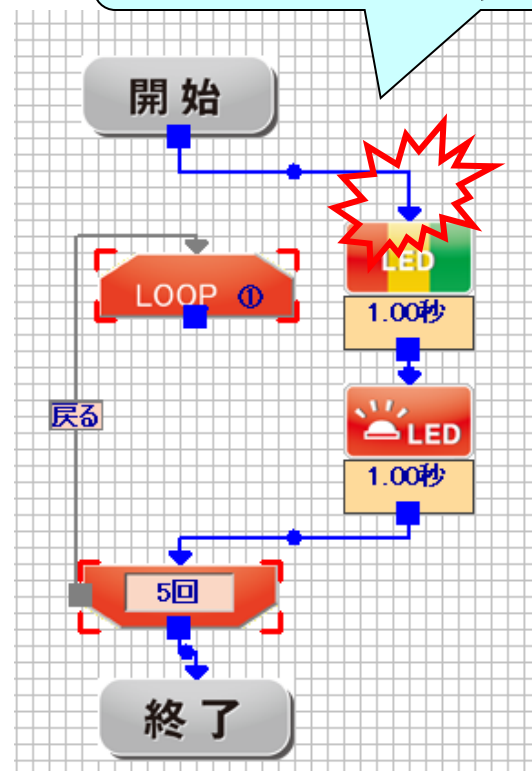
繰り返しの途中で止まってしまう



一度も繰り返さずに終了してしまう

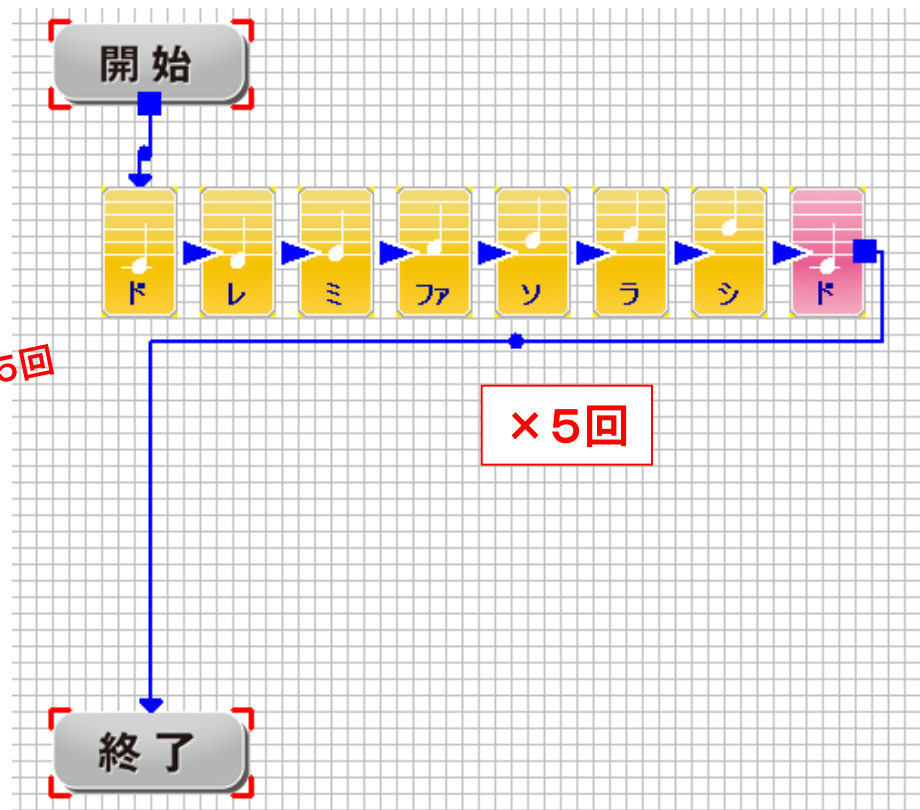
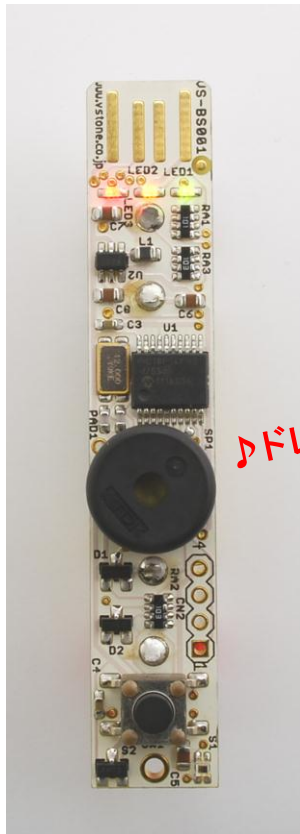


繰り返しのはじめがおかしくなってしまう



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。

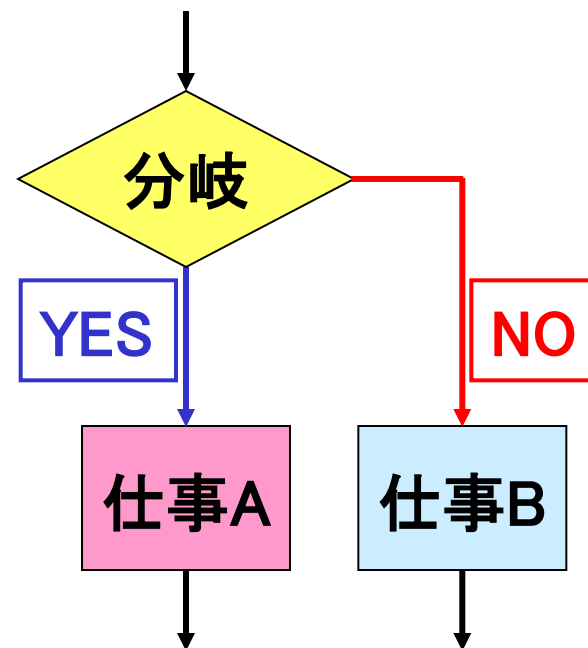


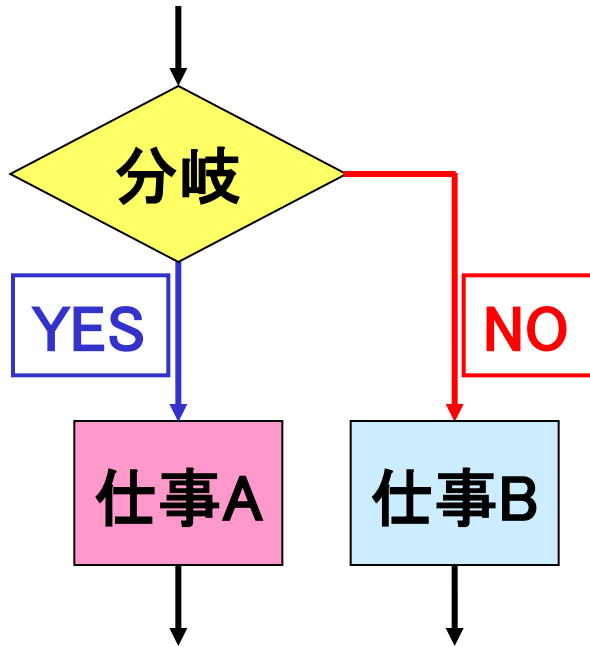
課題3 解答例

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。

The screenshot shows the Vstone software interface with a flowchart for a program that plays the melody 'ドレミファソラシド' (Do-Re-Mi-Fa-Sol-La-Si-Do) 5 times. The flowchart starts with a '開始' (Start) block, followed by a 'LOOP' block. Inside the loop, there is a sequence of notes: 'ド', 'レ', 'ミ', 'ファ', 'ソ', 'ラ', 'シ', and 'ド'. A '戻る' (Return) arrow connects the end of the note sequence back to the start of the loop. Below the loop is a counter block set to '5回' (5 times). A callout box points to this counter with the text '繰り返し回数を5回に設定する' (Set the number of repetitions to 5). 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 control panel on the left with various settings like volume, LED, and sensors.

- 分岐のプログラム
- センサ値を記録しよう





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、センサで感知しドアの前に人がいる場合、「開く」ドアの前に人がいない場合、「閉じる」という処理が行なわれています。

次にセンサを用いた分岐のプログラミングを行ないます。

→照度センサを手で覆ったときに赤LEDが点灯し、手で覆わないときは緑LEDが点灯するプログラムを作成する

分岐ブロック





温度センサ

照度センサ

温度センサ (サーミスタ)

温度変化により抵抗値が変化し、温度を計測することができます。
→ バッテリーの温度管理や家電製品に使用されています。

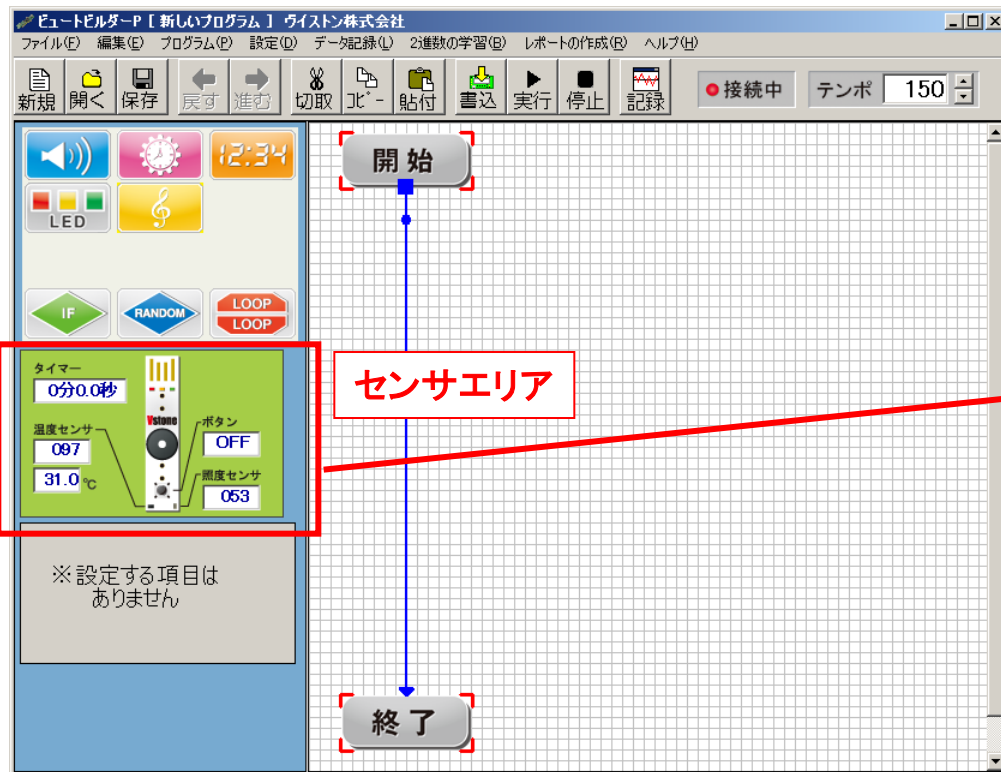
照度センサ (フォトランジスタ)

照度センサに当たる光があたると、電流が流れ(流れやすくなります)。この変化を読み取ることにより、明暗を検知することができます。
→ 携帯電話の液晶の制御などに使用されています。

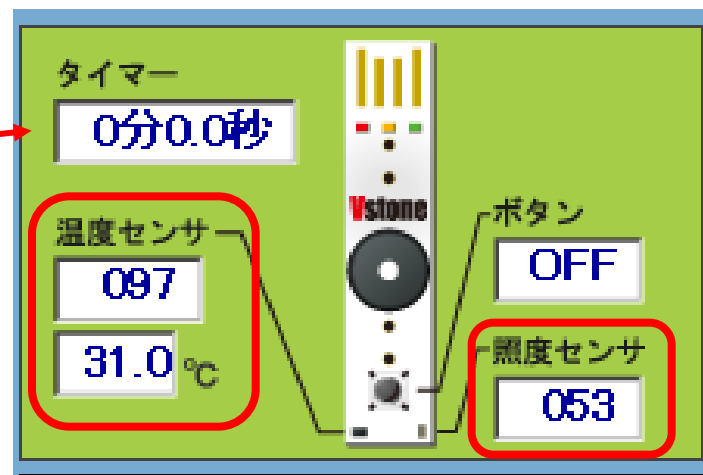
インターフェースが、赤外線センサのアナログ信号を、コンピュータが処理するためにデジタル信号に変換しています。

※ インターフェースとは情報のやり取りを仲介するもの、その仕組みのこと

センサ - **インターフェースA** - **コンピュータ** - **インターフェースB** - アクチュエータという関係になります。



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。



手で照度センサを覆ったときの反応を確かめましょう。

●センサを手で覆うと、数値はいくつになる？ → センサ値 []

●手をはなすと、数値はいくつになる？ → センサ値 []

分岐ブロック【しきい値の決定方法について】

開始

IF

照度センサ<80

終了

タイマー
0分0.0秒

温度センサ
094
29.5℃

ボタン
OFF

照度センサ
053

○ 温度 ○ 時間 ○ ボタン
● 明るさ・その他

照度センサ の値は
80 と比べて
小さい値?

しきい値

- ・手でセンサを覆っているとき
センサ値=0付近
- ・手を離しているとき
センサ値=120付近

センサ値は、ノイズや電源の影響で、実際には若干変化します。

ノイズなどの影響を踏まえ、「手でセンサを覆っているとき」と「手を離しているとき」の状態を分けるしきい値は2つのセンサ値の真ん中にするようにします。

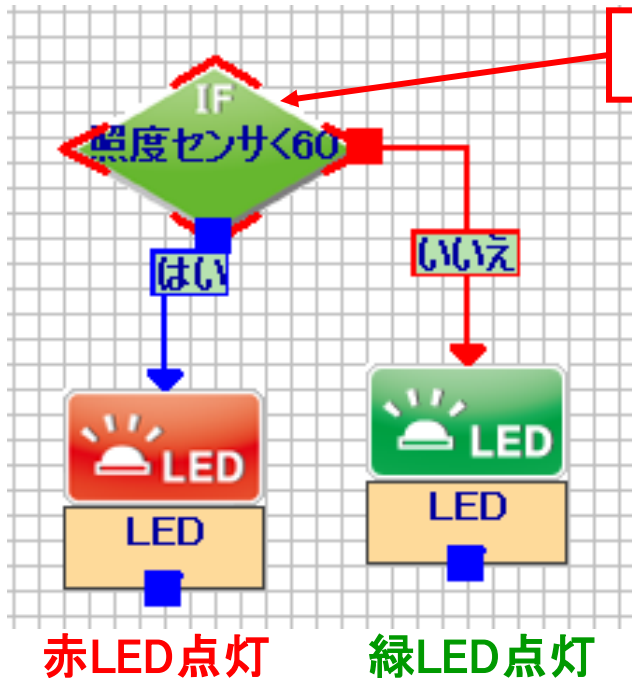
しきい値=60

$$(0+120) \div 2=60$$

$$0 \sim 60 \sim 120$$
$$+60 \quad +60$$

照度センサを用いた分岐のプログラム

照度センサを手で覆ったときに赤LEDが点灯し、
手で覆わないときは緑LEDが点灯するプログラムを作成しよう。



しきい値は60

- 温度
- 時間
- ボタン
- 明るさ・その他

照度センサ の値は

60 と比べて

小さい値?

【 照度センサ<60 】

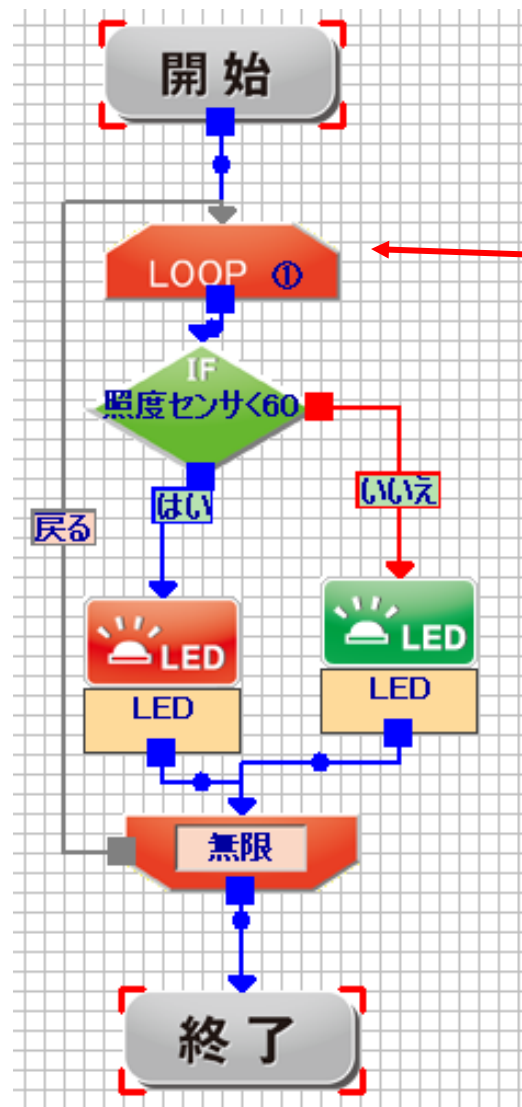
・センサを手で覆う(0付近)

照度センサが60よりも小さい→赤LED点灯

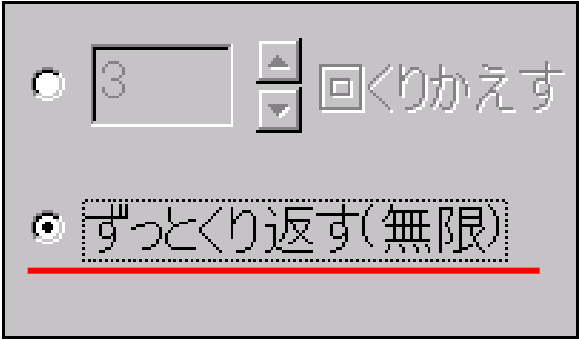
・センサから手を離す(120付近)

照度センサが60よりも大きい→緑LED点灯

照度センサを用いた分岐のプログラム



プログラムを繰り返すのでループを使用する。(無限)



設定できたらプログラムを書き込み、実行してみましょう。

課題4 照度センサを用いた分岐のプログラム

照度センサを用い、手(指)で覆ったときにドレミ(四分音符)と鳴り、手で覆わなかったときはすべてのLEDが点灯するプログラムを作ろう。

照度センサを覆う場合
ドレミと鳴る



照度センサを覆わないとき
全てのLEDが点灯



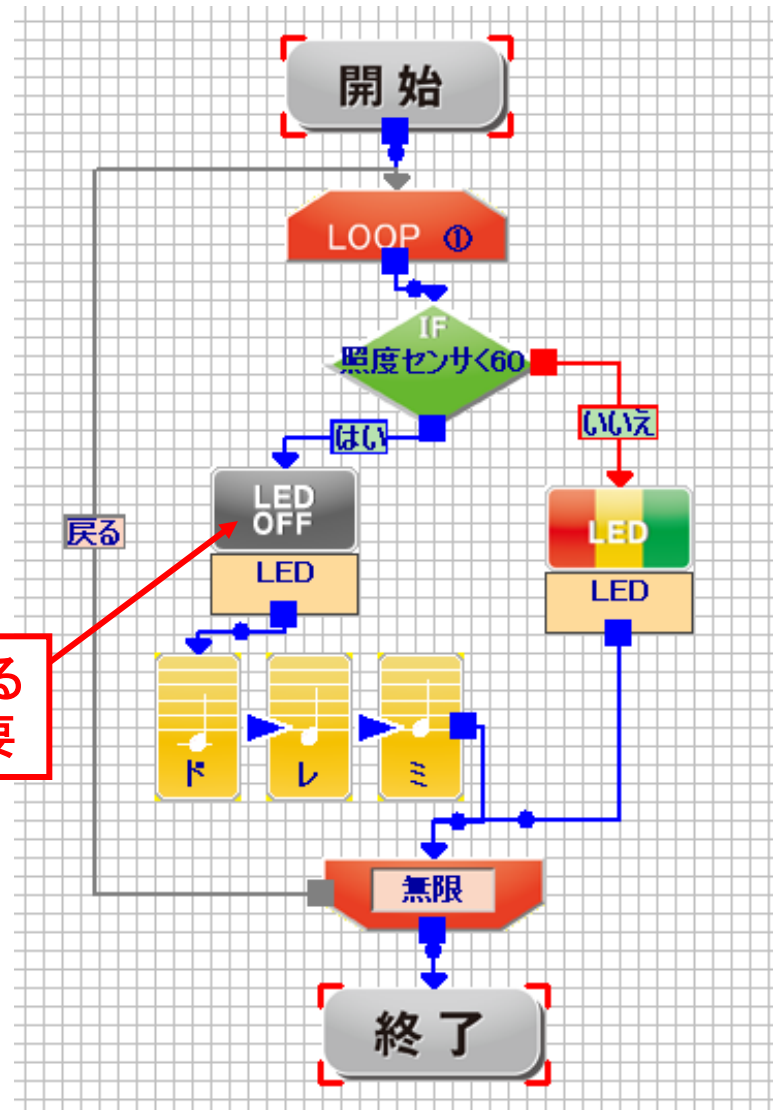
照度センサ

課題4 解答例

手(指)で覆った場合
→ ドレミ(四分音符)と鳴る

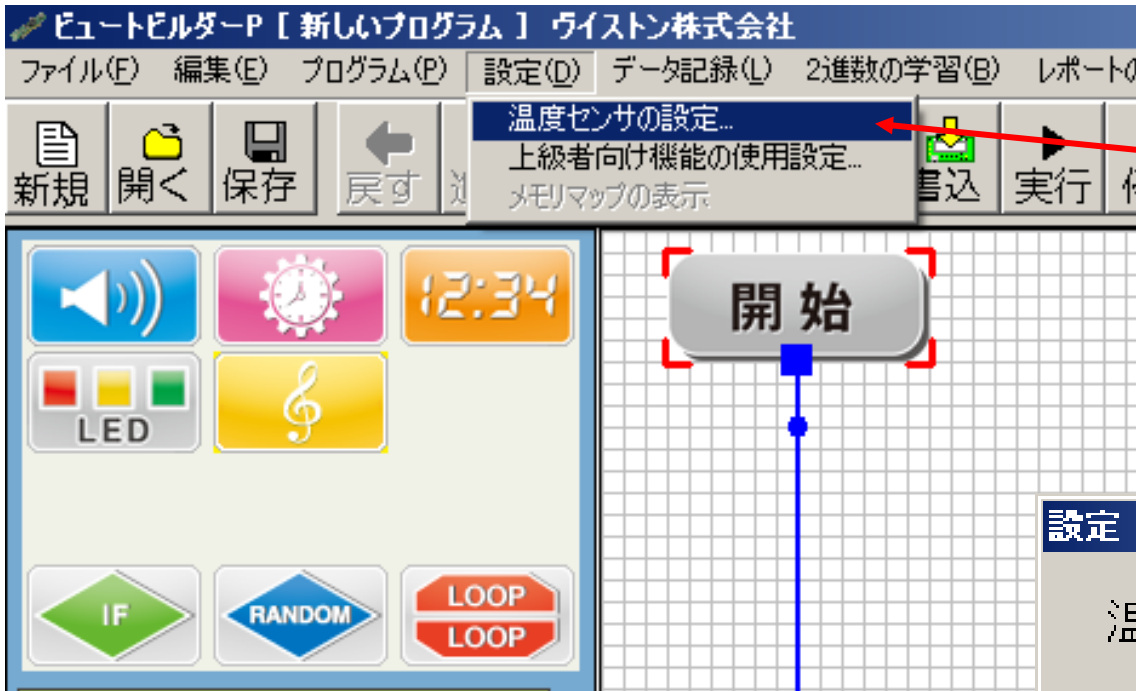
手で覆わなかった場合
→ 全てのLEDが点灯する

LEDを消灯する
ブロックが必要



温度センサの調整方法について

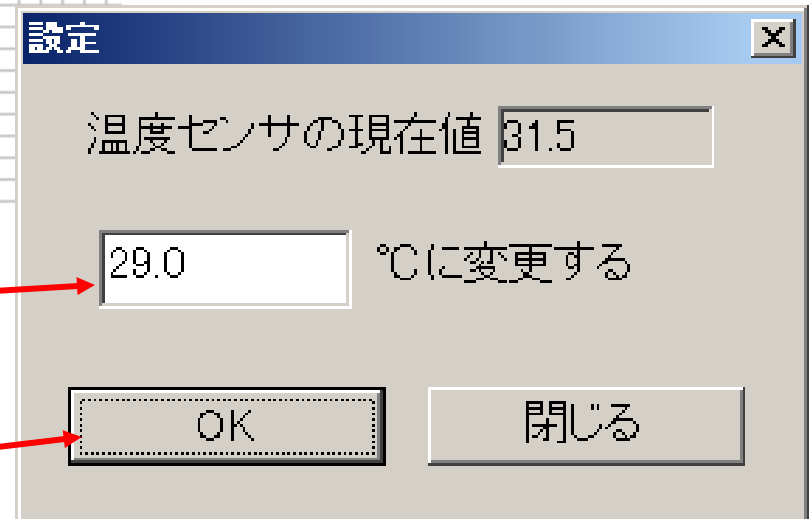
温度センサには個体差があり、センサ値の補正が必要な場合があります。



1. 設定 →
温度センサの設定を選択

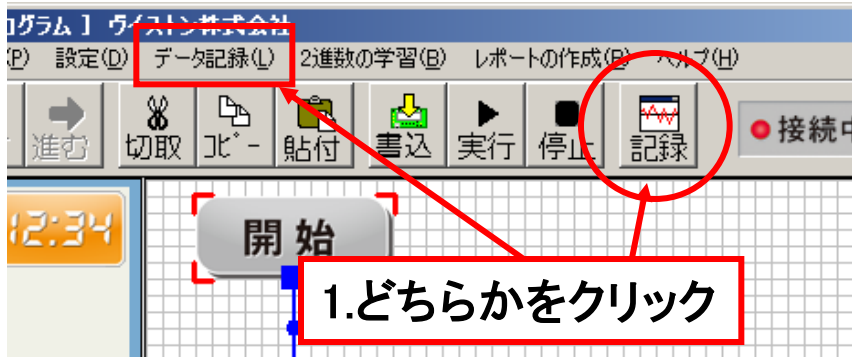
2. 温度計などで実測した
温度を入力する。

3. OKをクリックする。

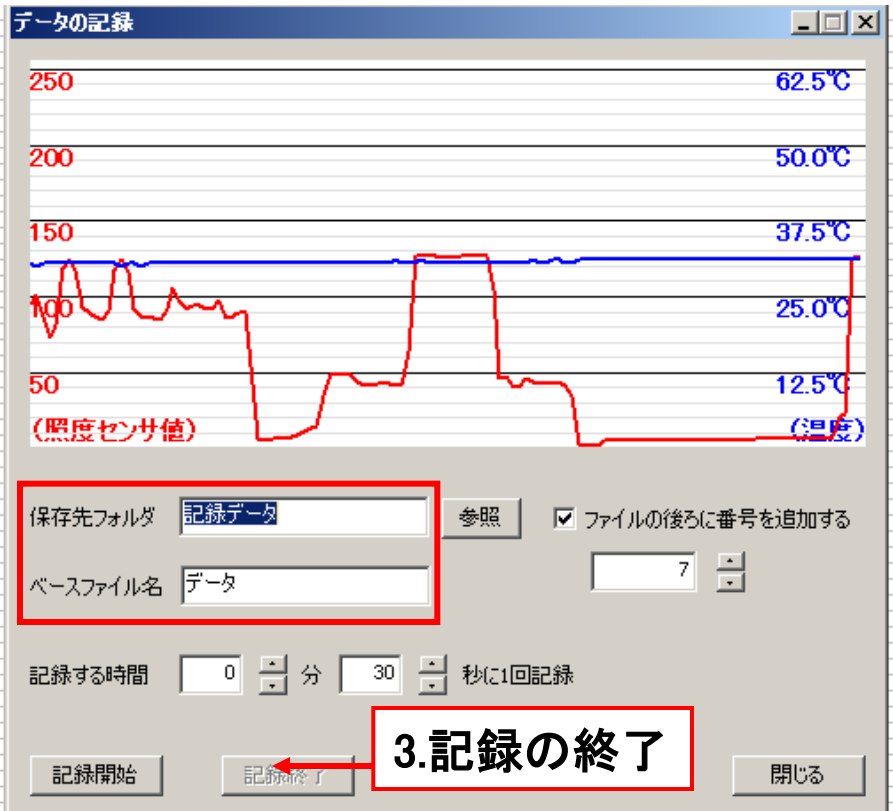


センサの情報をパソコンに記録しよう

パソコンと計測制御プログラマーを接続すれば、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

センサの情報をパソコンに記録しよう

ビュートビルダーPのフォルダ内に記録データが保存されています。

The image shows two overlapping Windows Explorer windows. The top window is titled 'ビュートビルダーP3' and shows the directory 'C:\Documents and Settings\Administrator\Desktop\ビュートビルダーP3'. It contains several folders: 'img', 'ファームウェア', 'サンプルプログラム', and '記録データ'. A red arrow points from a red-bordered text box to the '記録データ' folder. The bottom window is titled '記録データ' and shows the contents of that folder, including a file named 'データ8.csv' (Microsoft Excel CSV File, 1 KB). A red arrow points from another red-bordered text box to this file.

1. フォルダが作成されている。
ダブルクリックする。

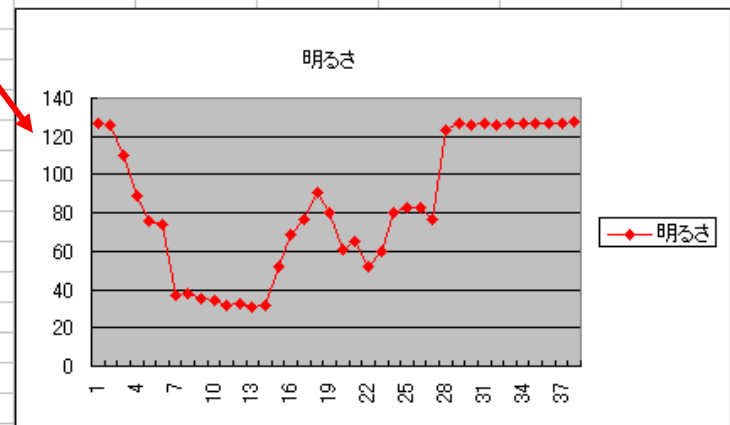
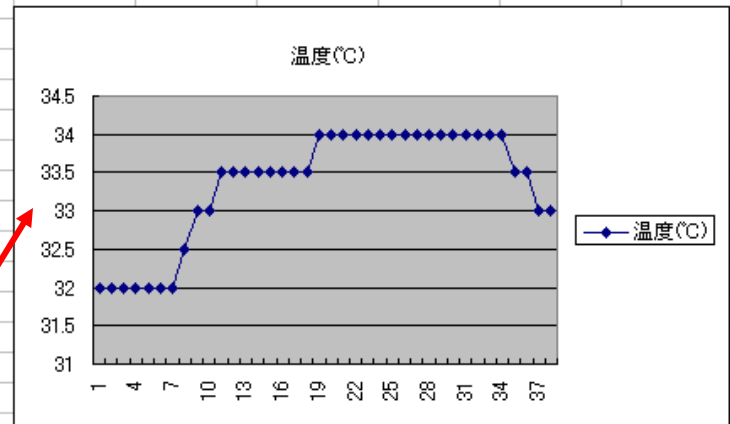
2. データファイルを開きます。
Excelで開くことができます。

センサの情報をパソコンに記録しよう。

記録データには、日付、時間、温度、明るさの計測結果が保存されています。

	A	B	C	D	E	F
1	年月日	時分秒	温度(°C)	明るさ		
2	2012/5/8	17:31:53	32	127		
3	2012/5/8	17:31:54	32	126		
4	2012/5/8	17:31:55	32	110		
5	2012/5/8	17:31:56	32	89		
6	2012/5/8	17:31:58	32	76		
7	2012/5/8	17:31:59	32	74		
8	2012/5/8	17:32:00	32	37		
9	2012/5/8	17:32:01	32.5	38		
10	2012/5/8	17:32:02	33	35		
11	2012/5/8	17:32:03	33	34		
12	2012/5/8	17:32:04	33.5	32		
13	2012/5/8	17:32:05	33.5	33		
14	2012/5/8	17:32:06	33.5	31		
15	2012/5/8	17:32:08	33.5	32		
16	2012/5/8	17:32:09	33.5	52		
17	2012/5/8	17:32:10	33.5	69		
18	2012/5/8	17:32:11	33.5	77		
19	2012/5/8	17:32:12	33.5	91		
20	2012/5/8	17:32:14	34	80		
21	2012/5/8	17:32:15	34	61		
22	2012/5/8	17:32:16	34	65		
23	2012/5/8	17:32:17	34	52		
24	2012/5/8	17:32:18	34	60		
25	2012/5/8	17:32:19	34	80		
26	2012/5/8	17:32:20	34	83		
27	2012/5/8	17:32:22	34	83		
28	2012/5/8	17:32:23	34	77		
29	2012/5/8	17:32:24	34	123		
30	2012/5/8	17:32:25	34	127		

グラフ化も可能です。



センサの情報をパソコンに記録しよう。

温度・明るさの計測結果があると、どのようなことができるかな？

【温度】

- 1日の温度変化がわかれば、植物を日陰に入れるタイミングがわかる。
- 1日の温度変化がわかれば、エアコン使用するのに適した時間がわかる。

【明るさ】

- 1日の明るさがわかれば、目覚まし時計を作ることができる。
- 1日の明るさがわかれば、電灯をつけるタイミングがわかる。
- 1日の明るさの変化を見れば、侵入者の有無を調べることができる。

【温度・明るさ】

- 植物栽培で栽培に適した環境かどうか調べることができる。
- 明るさだけの情報よりも性能がいい目覚まし時計をつくることができる。

課題5 温度センサを用いた分岐のプログラム

温度センサとLEDを用いて、温度計を作成しよう。

温度が 25°C より低い
→ 緑色LEDが点灯

温度が $25\sim 29^{\circ}\text{C}$ のとき
→ 黄色LEDが点灯

温度が 29°C より高い
→ 赤色LEDが点灯



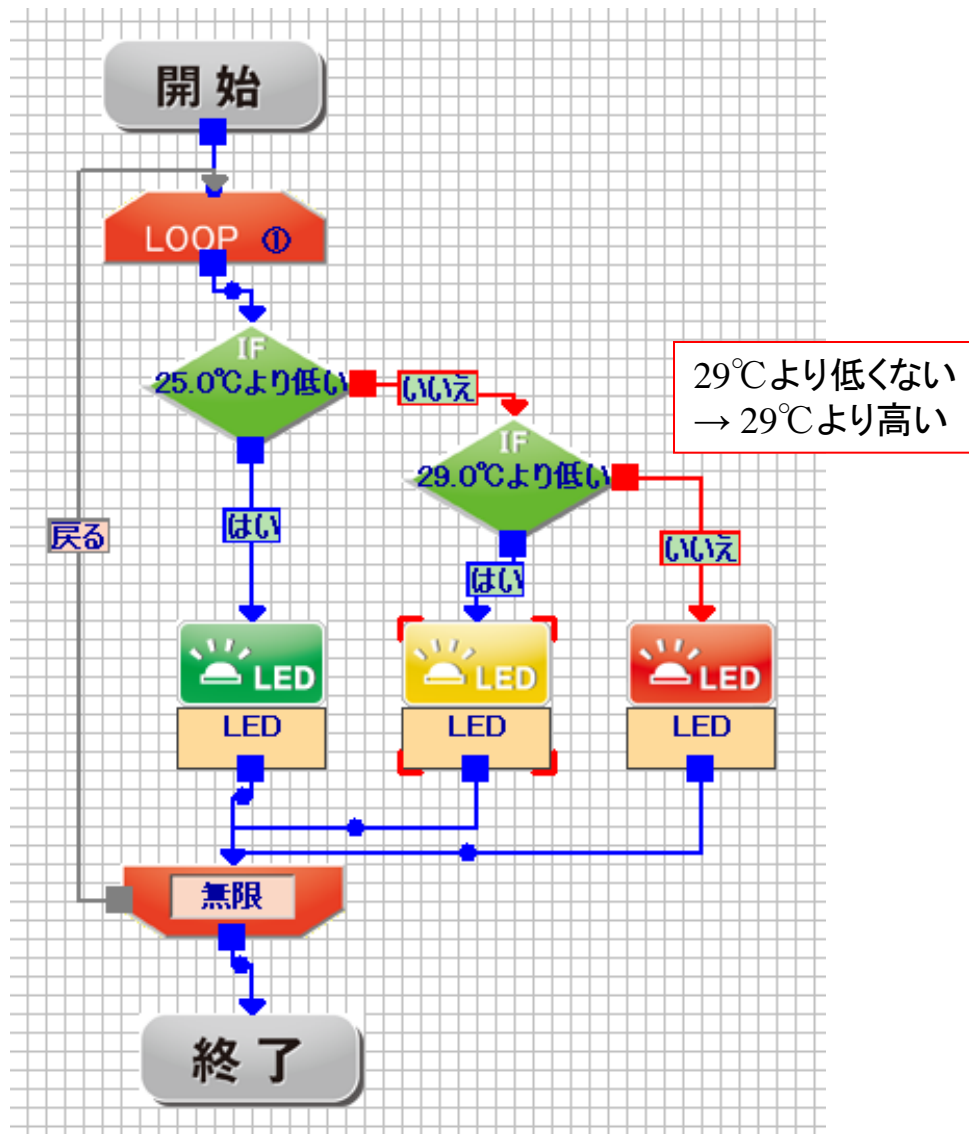
課題5 解答例 温度計を作成しよう

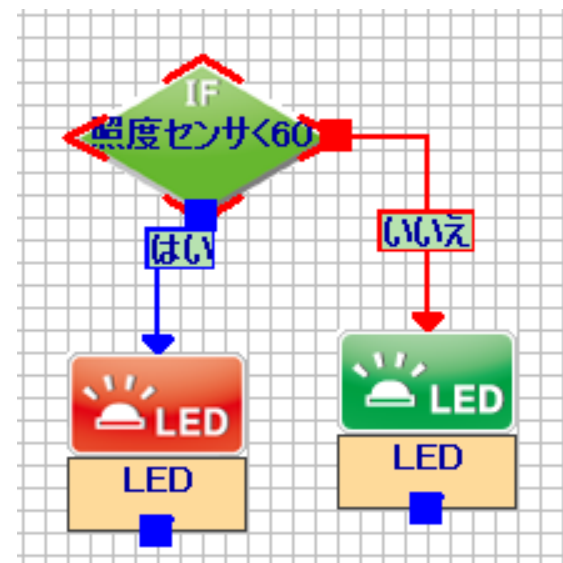
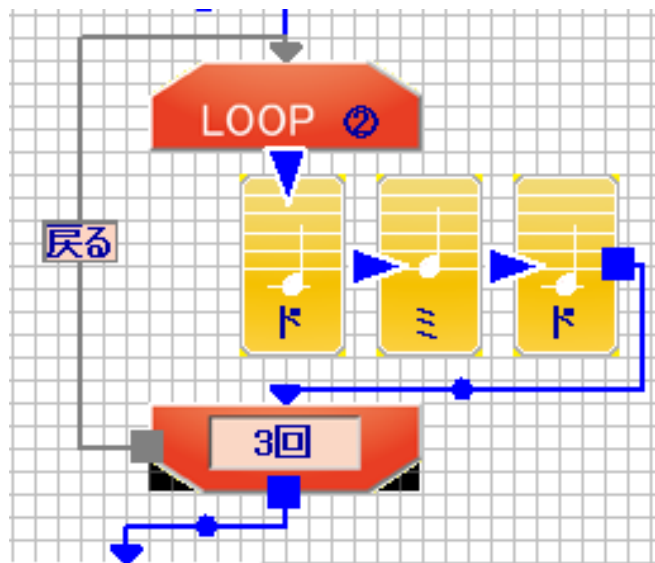
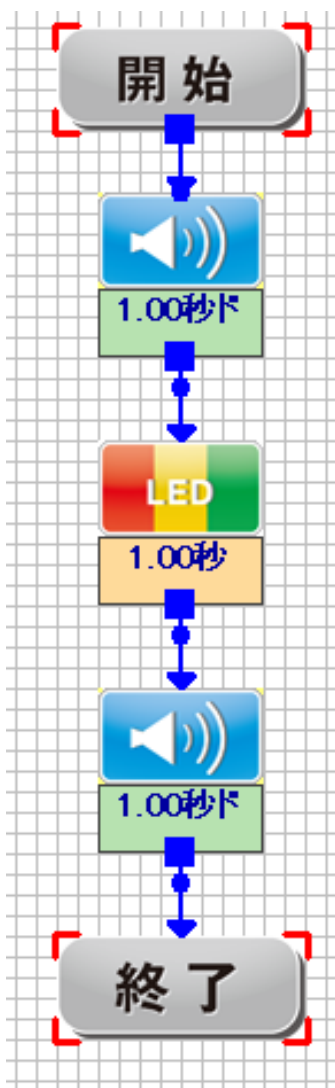
分岐ブロックを2つ使用します。

温度が25°Cより低い
→ 緑色LEDが点灯

温度が25~29°C
→ 黄色LEDが点灯

温度が29°Cより高い
→ 赤色LEDが点灯





プログラミングには主に

- ・順次
- ・繰り返し
- ・分岐

の3つの手順があります。

- 音楽を作成してみよう。
- 電子オルゴールを作成しよう。



音楽を作成しよう

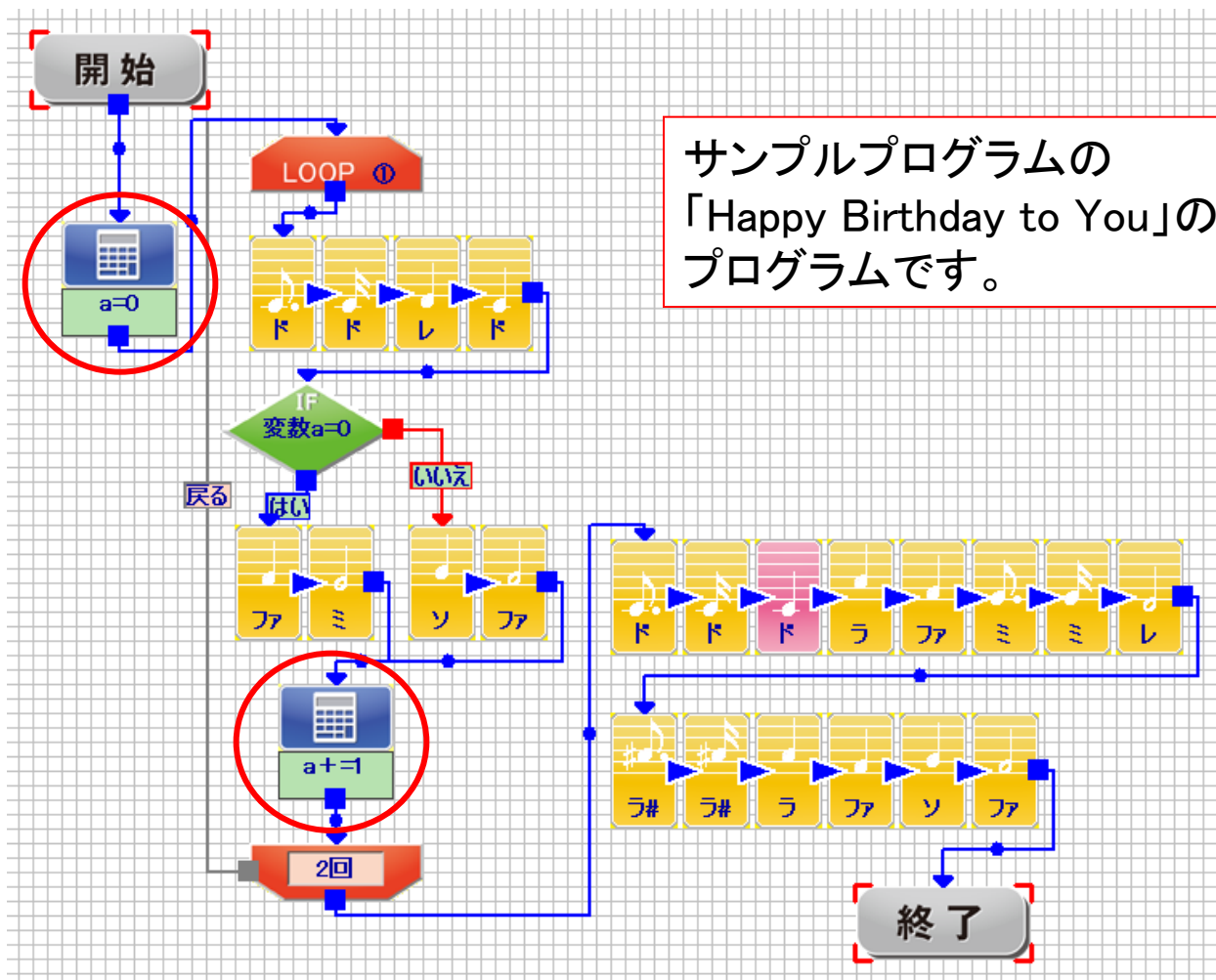
好きな曲や気になる曲など、音楽を作成してみよう。

※ 1つのプログラムで使用できるブロックの数は限られています。(音符ブロックのみの場合、42個まで) 繰り返しブロックや分岐ブロックを使用し、プログラムをまとめるようにしてください。

The screenshot shows the Vstone software interface for creating music. The window title is "ビュートビルダーP [音楽サンプル「さくら」.vbd] ラ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(O)", "データ記録(L)", "2進数の学習(B)", "レポートの作成(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", "貼り付け", "書き込", "実行", "停止", and "記録". A "テンポ" (Tempo) control is set to 150. The main workspace is a grid with a flowchart. It starts with a "開始" (Start) block, followed by a "LOOP ①" block containing three "ラ" (La) blocks. Below this is a "2回" (2 times) block. The flowchart then enters a larger loop structure labeled "LOOP ②". Inside this loop, there are two rows of notes: the first row contains "ラ", "シ", "ド", "シ", "ラ", "シ", "ラ", "ファ" (La, Si, Do, Si, Ra, Si, Ra, Fa), and the second row contains "ミ", "ド", "ミ", "ファ", "ミ", "ミ", "ド", "シ" (Mi, Do, Mi, Fa, Mi, Mi, Do, Si). A "戻る" (Return) block is connected to the end of the loop. Below the loop is another "2回" (2 times) block. On the left side of the interface, there are control panels for "LED", "IF", "RANDOM", "LOOP LOOP", and a "タイマー" (Timer) section with "温度センサ" (Temperature sensor) and "湿度センサ" (Humidity sensor) inputs. A note at the bottom left states "※設定する項目はありません" (There are no items to set).

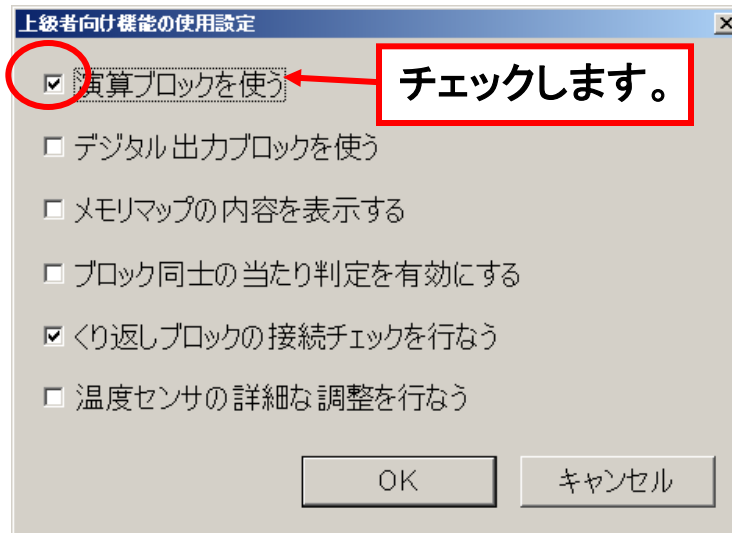
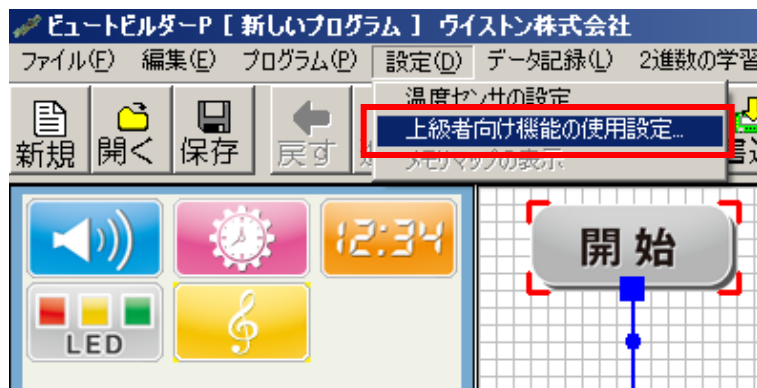
サンプルプログラムの「さくら」のプログラムです。

※ サンプルプログラムの「上級者向け Happy Birthday to You」では、
上級者設定の変数演算ブロックが使用されています。

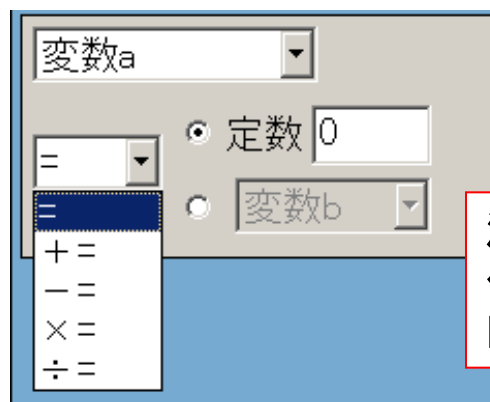


演算ブロックについて

演算ブロックを使用する場合は、上級者向け機能設定で「演算ブロックを使う」を有効にする



演算ブロックのアイコンが表示される

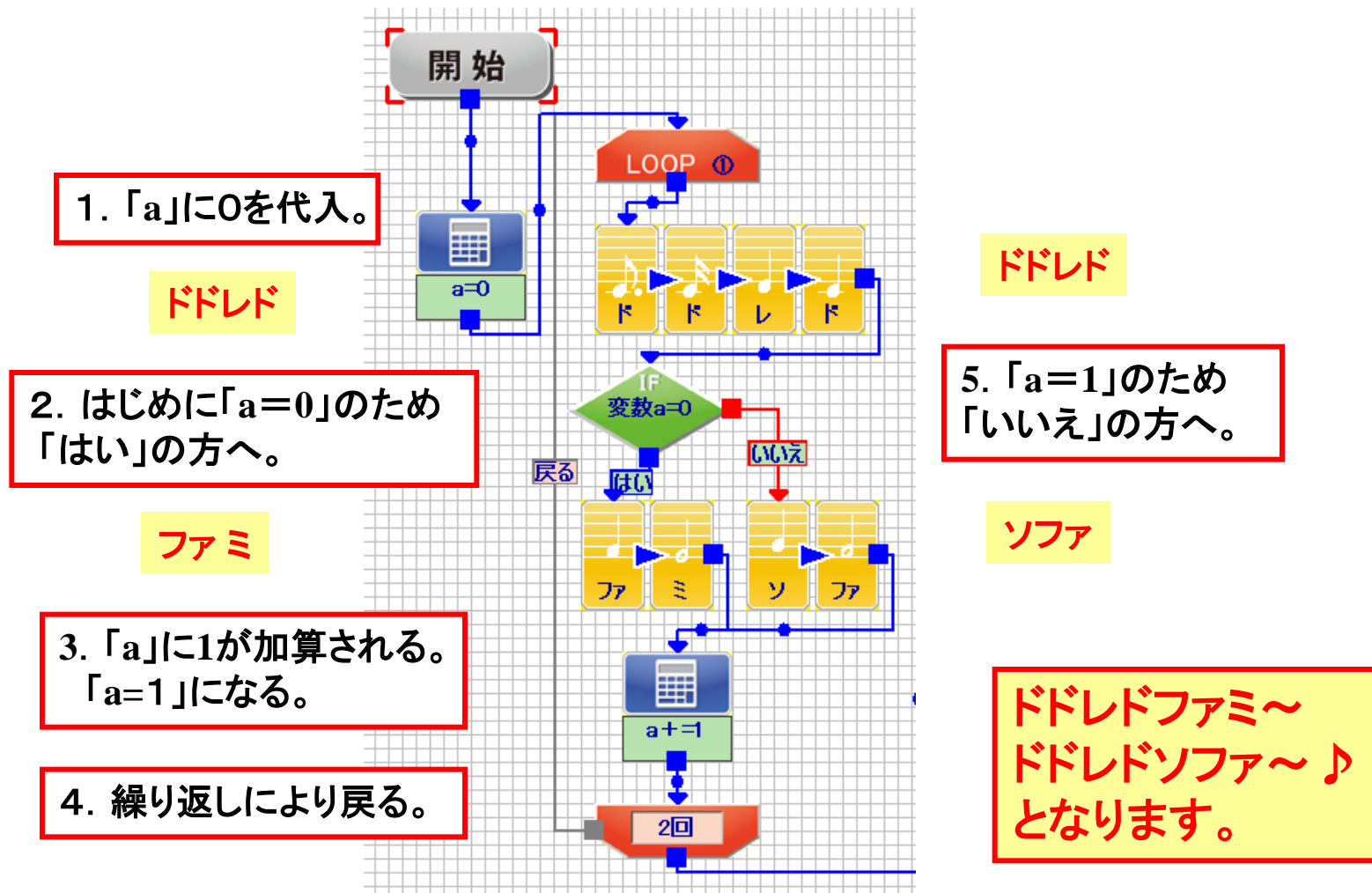


演算ブロックでは、代入や加算などの四則演算が可能。

＜演算ブロックの設定エリア＞

音楽を作成しよう。

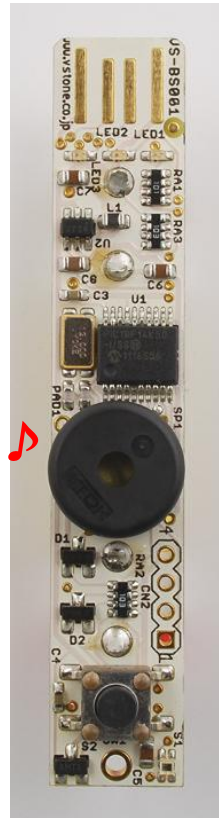
サンプルプログラムの「上級者向け Happy Birthday to You」について



電子オルゴールを作成しよう。

箱を開けたら、「Happy Birthday to You」が流れるプログラムを作成しよう。

箱を開ける(明るい)



「Happy Birthday to You」

ドドレドファミ

ドドレドソファ

ドドド(高)ラファミミレ

＃ラ＃ララファソファ

(サンプルプログラムを参考にしよう)

箱の中(暗い)

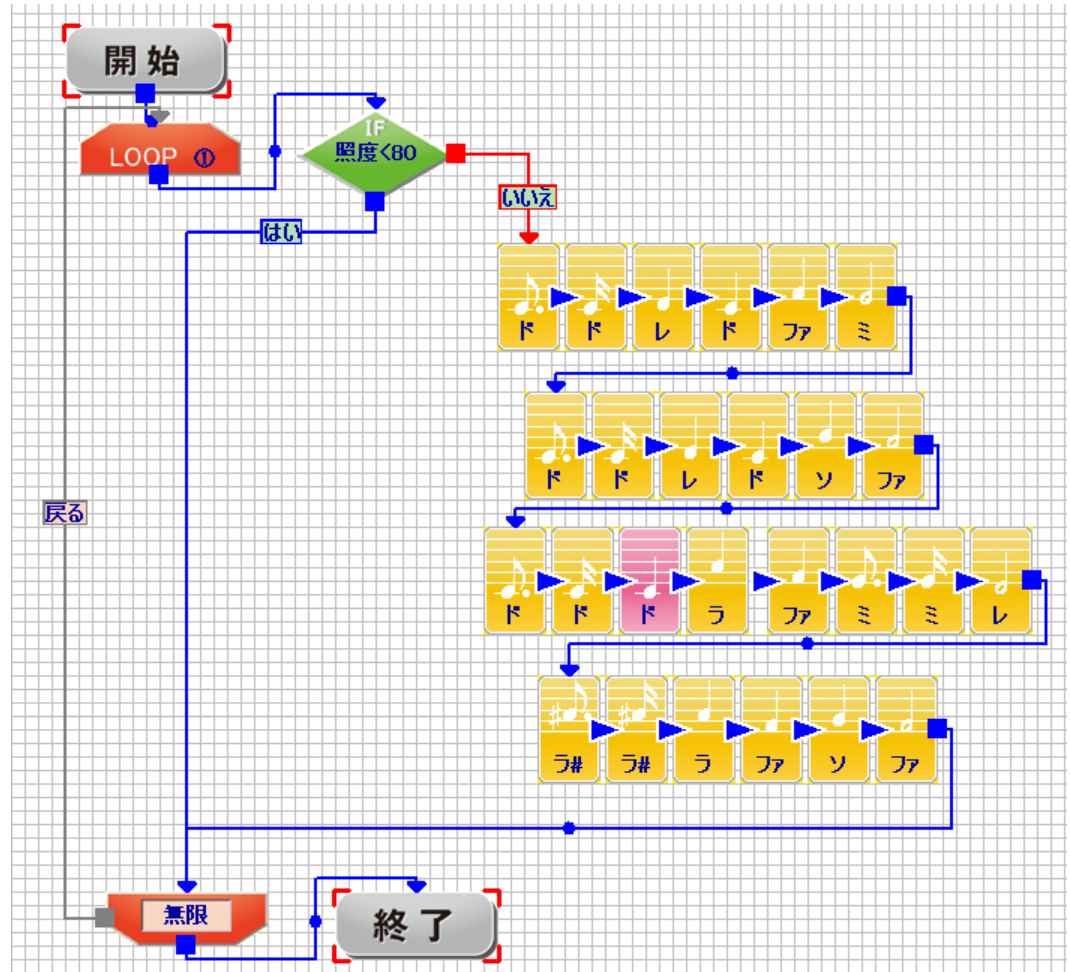


照度センサ

電子オルゴールを作成しよう。

箱を開けたら、「Happy Birthday to You」が流れるプログラムを作成しよう。

照度センサが80以下(暗い) はい
→ 何もしない。
照度センサが80以下ではない いいえ
→ Happy Birthday to You ♪

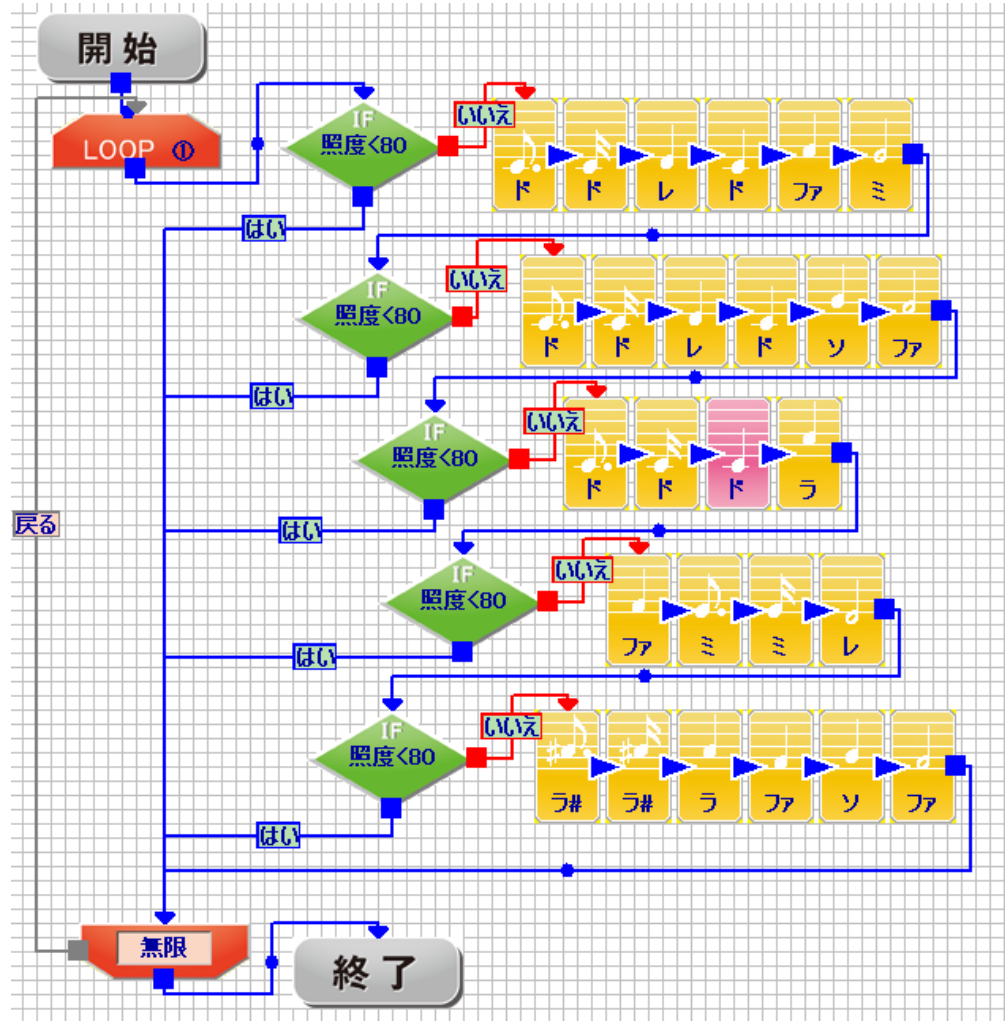


※ 箱を閉じた後も曲が終わるまで音が鳴ってしまう。

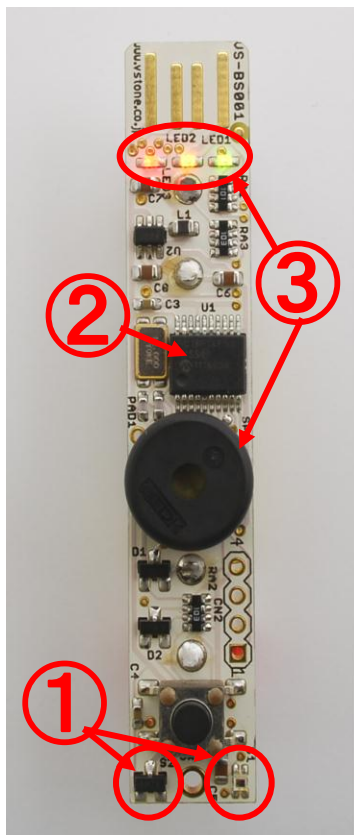
電子オルゴールを作成しよう。

箱を開けたら、「Happy Birthday to You」が流れるプログラムを作成しよう。

曲を分割し、明暗を判定する分岐ブロックをたくさん使用すると、箱を閉じたときの反応が早くなります。



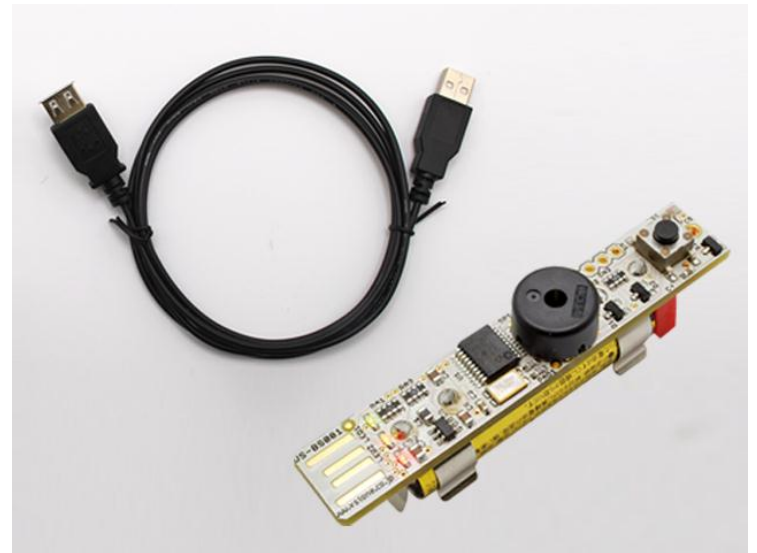
計測制御プログラマーの機能を用い、どんなことができるかな。
生活で役に立つものを考えて、作ってみよう。



- ① **センサ** = 照度センサ、温度センサ
明るさや温度の状況を見る
- ② **コンピュータ** = PICマイコン
命令を覚えたり、行動を考える
- ③ **アクチュエータ** = ブザー、LED
音を出す。点灯・点滅する。

コンピュータは、順次や繰り返し・分岐などのプログラムによって処理を行ない、我々の生活において計測や制御で重要な役割を果たしています。

今後もコンピュータの発展に伴い、生活に役立つたくさんのコンピュータ制御機器が出てくることでしょう。



普段0～9までの10種類の数字の組み合わせで、様々な数値を10進数で表しています。

コンピュータでは「0と1」の2種類の数字だけで数値を表現する2進数が使用されています。

10進数	2進数
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010

スタートビルダーP [新しいプログラム] ウイストーン株式会社
ファイル(E) 編集(E) プログラム(P) 設定(O) データ記録(L) 2進数の学習(B) レポートの作成(R) ヘルプ(H)

新規 開く 保存 戻す 進む 切取 北- 貼付 書込 実行 停止 記録 接続中 テンポ 150

開始

①クリックします。

2進数の学習

LEDの状態

	B2 (赤)	B1 (黄)	B0 (緑)
LEDの状態	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2進数	1	1	1
10進数	7		0~7の順番で点灯デモ

2進数→10進数
 $1 \times 2^2 \text{ 乗} + 1 \times 2^1 \text{ 乗} + 1 \times 2^0 \text{ 乗}$
 $= 1 \times 4 + 1 \times 2 + 1 \times 1$
 $= 7$

閉じる

終了

②ウインドウが現れます。

※設定する項目はありません

2進数の学習

LEDの状態

	B2 (赤)	B1 (黄)	B0 (緑)
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2進数	1	1	1
10進数	<input type="text" value="7"/>		(0~7)

0~7の順番で点灯デモ

2進数→10進数
 $1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$
 $= 1 \times 4 + 1 \times 2 + 1 \times 1$
 $= 7$

閉じる

チェックをはずすと、2進数や10進数の数値が変わります。
計測制御プログラマーのLEDも変化します。



演算ブロックを用い、LEDに数値(10進数)を代入することで、LEDの制御が可能です。

