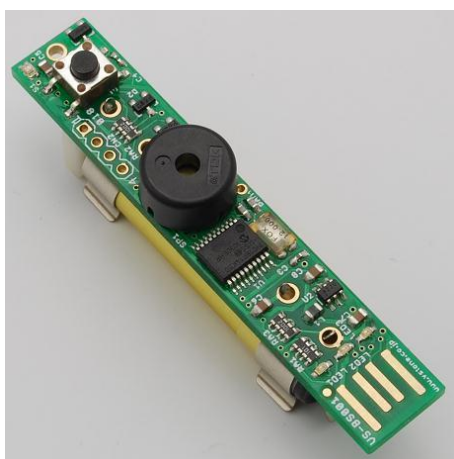


(2012.7.30)

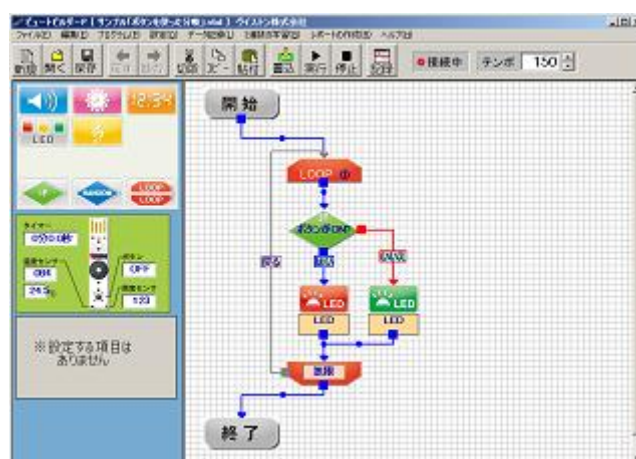
# 計測制御プログラマー用ソフトウェア ビュートビルダーP プログラム学習の手引き

ヴイストン株式会社

計測制御プログラマーは、PC と接続して温度や明るさをセンサから調べたり、フローチャート形式のプログラムを作って動かすことが出来るなど、中学校技術家庭科の「計測と制御」を学習するのに最適な教材です。この説明書では、初めて計測制御プログラマーを使う人のために、付属のソフトウェア「ビュートビルダーP」を使って簡単なプログラムが作れるようになるまでを説明します。



**計測制御プログラマー**  
説明では「本体」と記述します  
(写真は試作品のため、  
実際の製品と異なります)



**ビュートビルダーP**  
説明では「ソフトウェア」と記述します

※ビュートビルダーPに関する詳しい説明は、「ビュートビルダーP 取扱説明書」をご参照ください。

# 1.PC の操作や基本的な用語の確認

本説明書では、マウスの使い方やファイル・フォルダのコピーなど、PC の操作方法や用語については詳しく説明していません。次の用語や操作方法がどのようなものか、あらかじめ調べて、自分で出来るようにしておきましょう。

## 本説明書で登場する、PC の基本的な操作方法や用語

1. マウスに関する用語・操作方法
  - (ア) クリック、ダブルクリック、右クリック
  - (イ) ドラッグ
  - (ウ) マウスカーソル
2. ファイルやフォルダに関する用語・操作方法
  - (ア) フォルダ
  - (イ) マイコンピュータ (Windows7 の場合は「コンピューター」)
  - (ウ) デスクトップ
  - (エ) フォルダを開く
  - (オ) ファイルのコピー、フォルダのコピー

# 2.必要なものを準備する

計測制御プログラマーでプログラムを作るには、次のものがが必要です。まずはこれらを準備しましょう。

## 準備するもの

- |                   |     |
|-------------------|-----|
| 1.計測制御プログラマー本体    | 1 個 |
| 2.単四乾電池           | 1 個 |
| 3.PC (下の条件を満たすもの) | 1 台 |
- ・ OS が Windows 2000/XP/Vista/7
  - ・ メモリが 128MB 以上
  - ・ CPU が Pentium-Ⅲ以上 (1GHz 以上のものが良いです)
  - ・ 画面のサイズが XGA (1024x768) 以上
  - ・ USB2.0 の接続ポートが一つ以上付いている

その他、接続用の USB 延長ケーブルがあると便利です。

### 3.ソフトウェアをインストールする

必要なものが準備できたら、ソフトウェア「ビュートビルダーP」をPCにインストールしましょう。

①web ページでダウンロードできる「ビュートビルダーP\_Release???.zip」というファイルがインストールもとのファイルです。



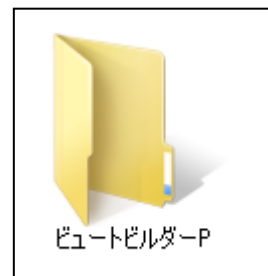
ビュートビルダー  
P\_Release01.zip

数値は  
バージョンを表す



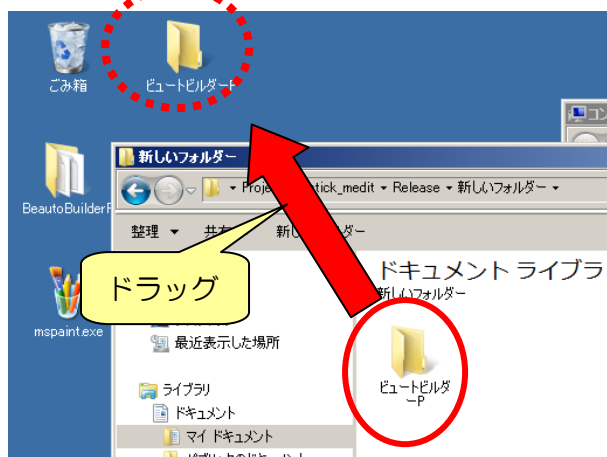
OS によって  
アイコンが異なる

②ファイルは ZIP 形式で圧縮されているので、展開用ソフトなどで中身を全て展開してください。展開すると、「ビュートビルダーP」というフォルダが出来ます。



ビュートビルダーP

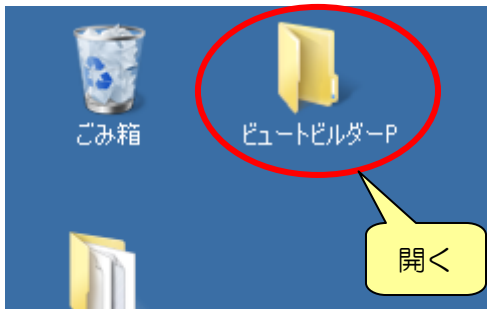
③出来上がったフォルダは、使いやすいようにデスクトップへドラッグして移動しましょう



## 4.ソフトウェアを起動し本体と通信する

ソフトウェアをインストールしたら、起動して本体と通信できるか確認してみましょう。

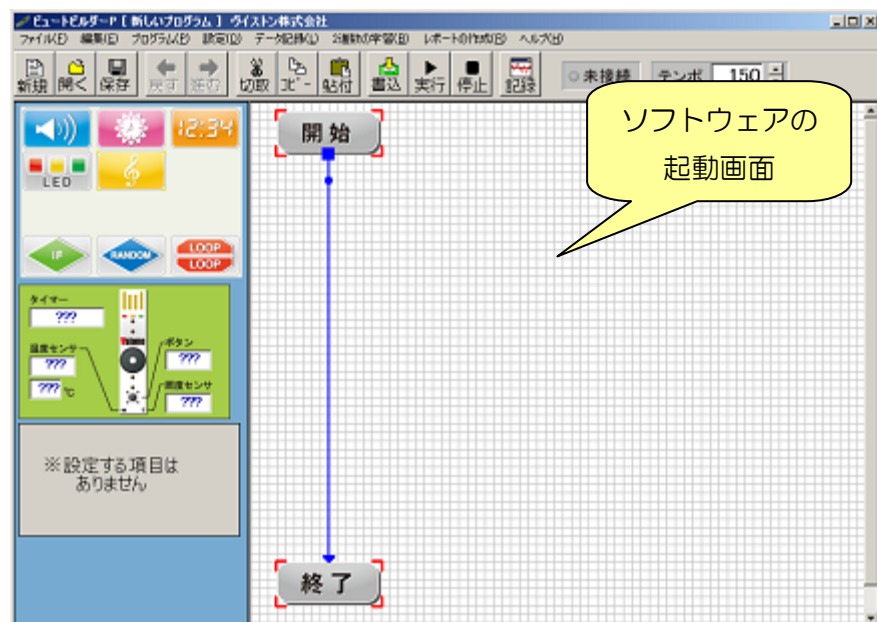
①デスクトップにコピーしたフォルダ「ビュートビルダーP」を開きます。



②フォルダの中にある「cl\_edit\_stk.exe」というプログラムをダブルクリックして実行します。



③ソフトウェアが起動し、下のような画面を表示します。

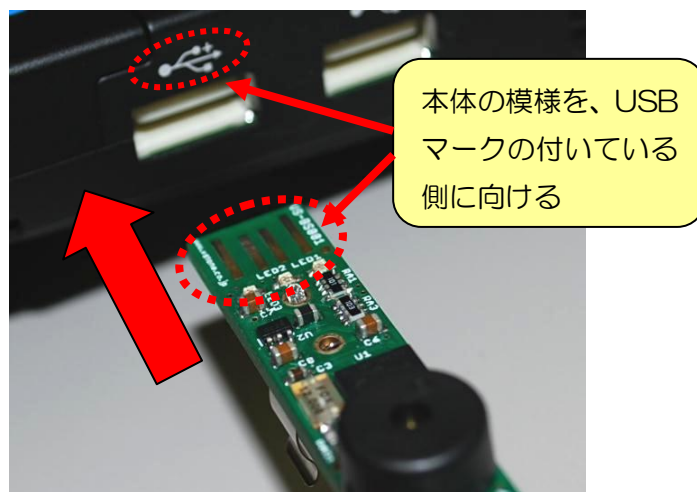


正しく起動できたら、今度は本体を PC と接続して通信させてみましょう。

①PC の「USB ポート」を探します

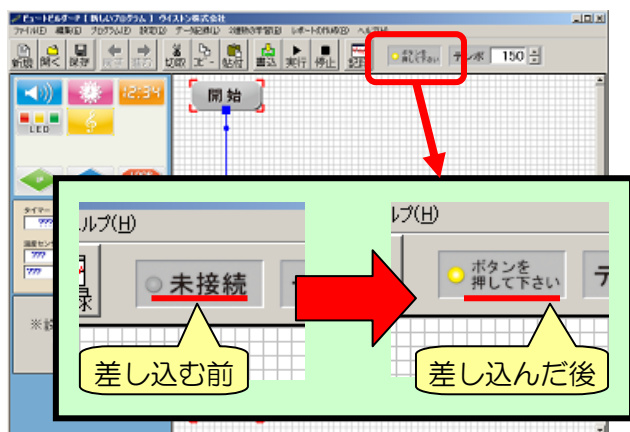


②向きに注意して、本体を差し込みます

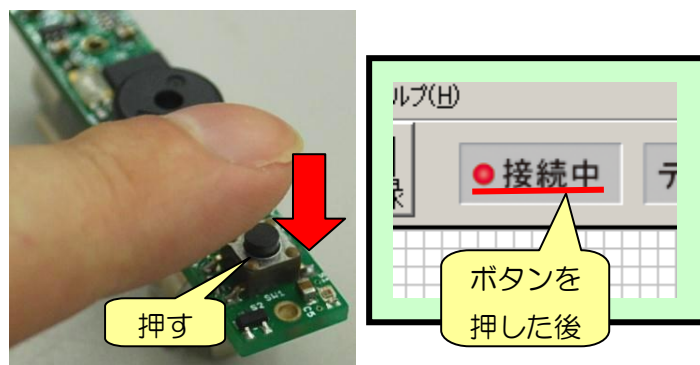


※差し込む向きを間違えると通信できません。

③接続したら、画面の「未接続」の表示が  
「ボタンを押してください」に変わります。



④本体のボタンを押して、電源を ON にします。  
ON にすると、画面の表示が「接続中」になります



プログラムを実行するとき・センサの数値を見るときは本体と通信させましょう。それ以外のときは、電池を節約するため、本体のボタンを 2 秒以上押してから離し、電源を OFF にしておきましょう。

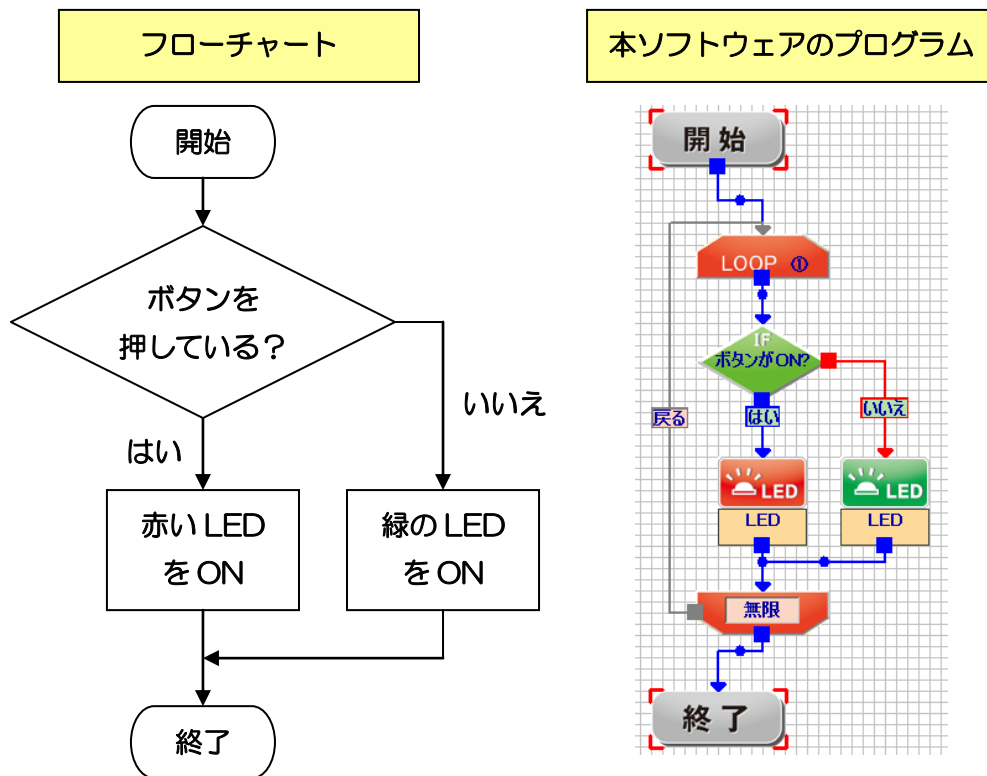
《Q.こんなときは?》

差し込んででも表示がずっと「未接続」のまま・・・差し込む向きが正しいか確認してください。

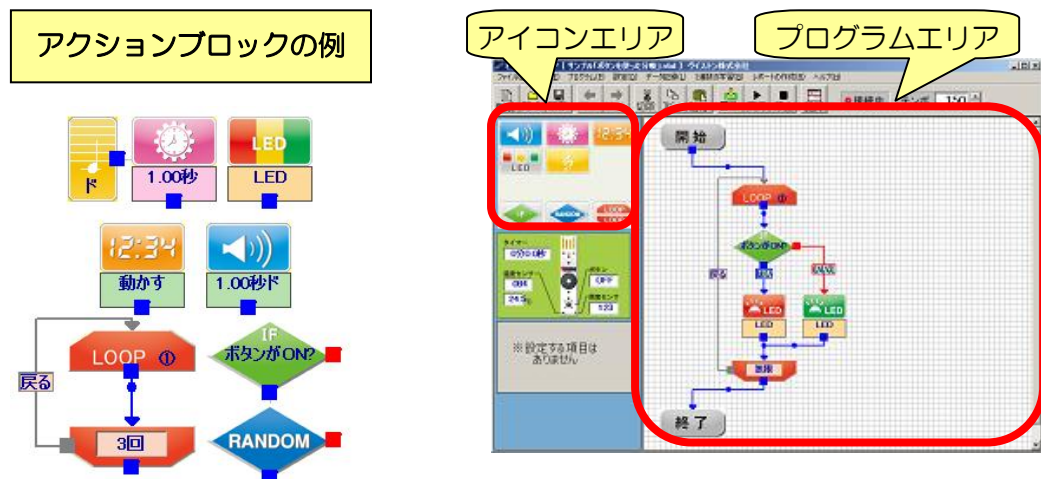
ボタンを押しても「接続中」にならない・・・もう一度本体を抜き差ししてください。

## 5.ブザーを鳴らすプログラムを作る

ここまでの説明でプログラミングの準備が整ったので、早速プログラムを作ってみましょう。本ソフトウェアのプログラムは、「フローチャート」に似せて作られています。フローチャートはプログラムの設計図のようなもので、命令や手順を四角や丸の図形で描き、実行する順番に合わせて矢印でつないでいきます。

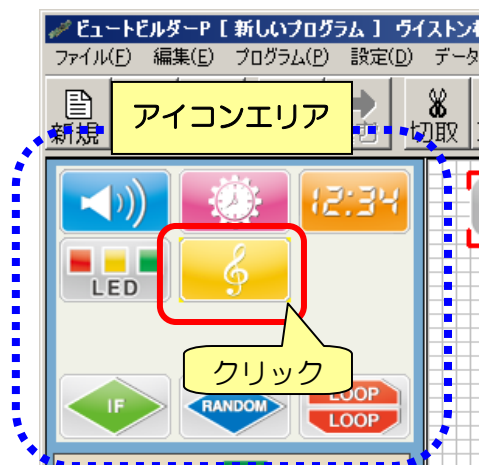


本ソフトウェアでは、この「命令や手順をあらわす図形」を「アクションブロック」または「ブロック」と呼びます。また、アクションブロックを並べてプログラムを作る部分を「プログラムエリア」、アクションブロックの種類を選ぶ部分を「アイコンエリア」といいます。

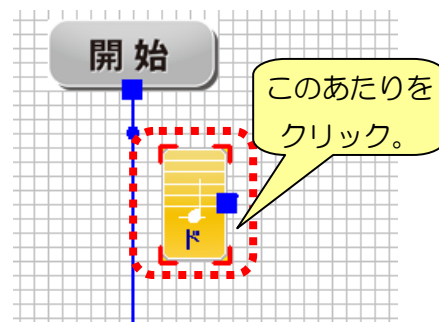


では、一番簡単な「ブザーを 1 音鳴らす」というプログラムを作りますので、皆さんも説明を読んで同じプログラムを作ってみましょう。

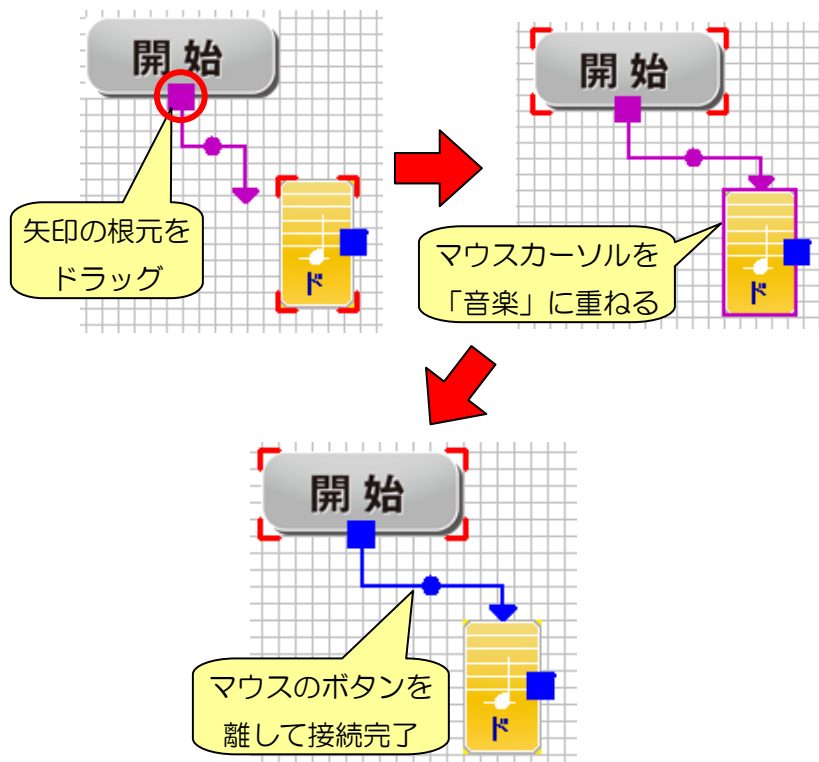
①「アイコンエリア（画面左上）」から、ト音記号のボタンをクリックします。



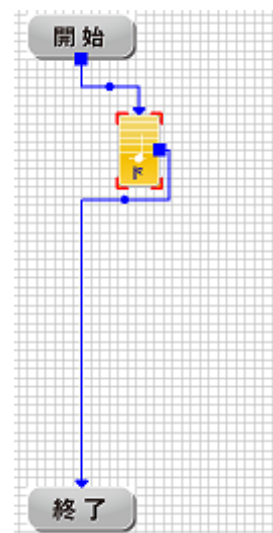
②「プログラムエリア（画面右）」の「開始」の下辺りをクリックし、音楽のアクションブロックを追加します。



③「開始」のアクションブロックから出ている矢印の根元をマウスでドラッグし、音楽のアクションブロックへつなげます。



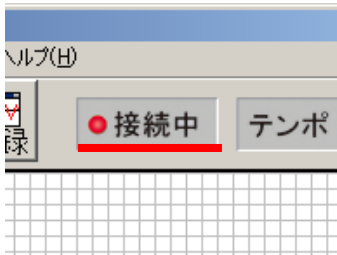
④音楽のブロックも、「終了」のブロックへ矢印をドラッグしてつなげます。



## 6.プログラムを実行する

作成したプログラムを実行する場合は、本体をPCと接続して、PCからデータを書き込みます。次の説明どおりに、作成したプログラムを書き込んで実行してみましょう。

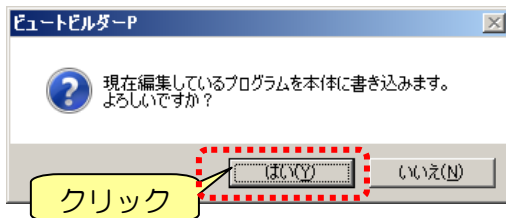
①本体とPCをつないで「接続中」にします。



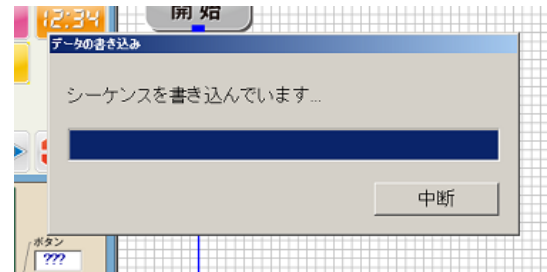
②「ツールバー（画面上側）」の「実行」ボタンをクリックします。



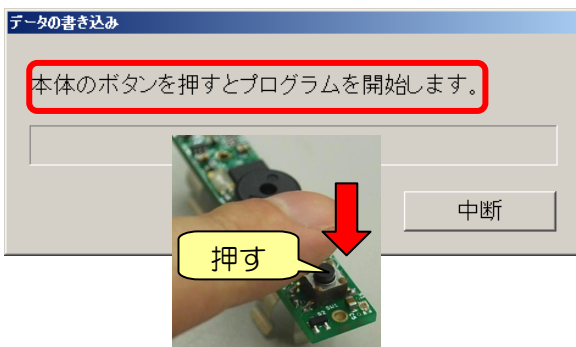
③確認画面を表示するので、「はい」をクリックします。



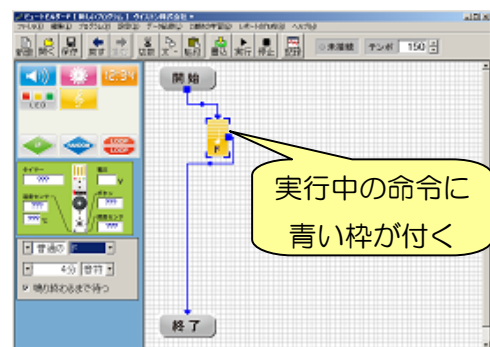
④書き込み中の画面を表示するので、しばらく待ちます。



⑤書き込み中の画面が下図のように切り替わったら、本体のボタンを押します。



⑥プログラムが始まります。実行中はプログラムエリアで進み具合がわかります。

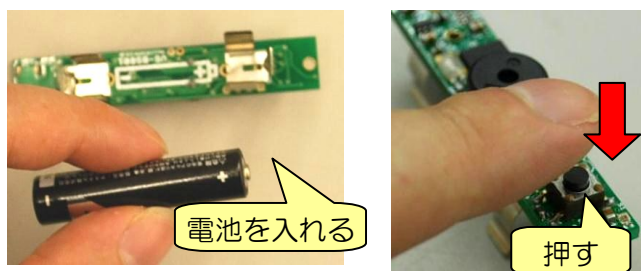


書き込みがなかなか終わらない、ボタンを押してもプログラムが始まらないなどの状態になったら、一度本体を抜き差しして、もう一度書き込みをやり直してみてください。

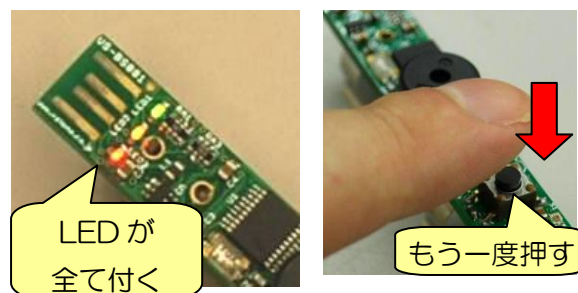


一度本体からプログラムを書き込んでいれば、本体だけでも同じプログラムを実行できます。

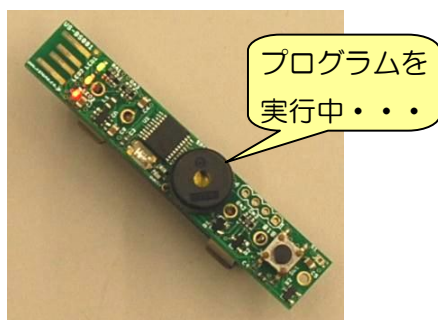
①本体を PC から外して、本体に単四乾電池を 1 本入れ、ボタンを押します。



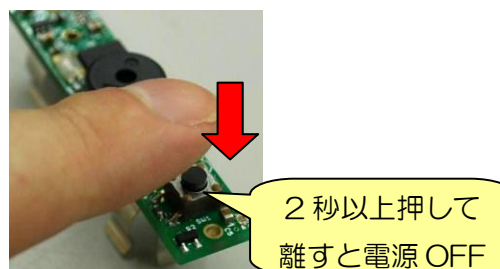
②ブザーが鳴って LED が全て光った状態になるので、もう一度ボタンを押します。



③プログラムを開始します。最後まで実行したら、②の状態に戻ります。

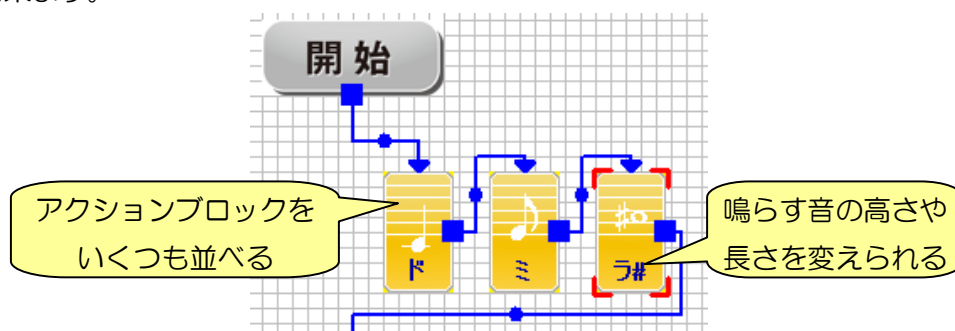


④使い終わるときは、ボタンを 2 秒以上押し、電源を OFF にしてください。



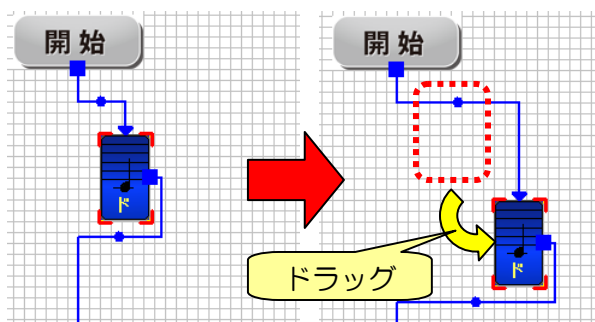
## 7. プログラムを修正する

今作ったプログラムは、アクションブロックを一つしか使っていませんが、アクションブロックはいくつも並べてつなげることが出来ます。また、今は 4 分音符の長さで「ド」の音が鳴りましたが、アクションブロックの設定を変更すれば、音の長さや高さを変えることが出来ます。

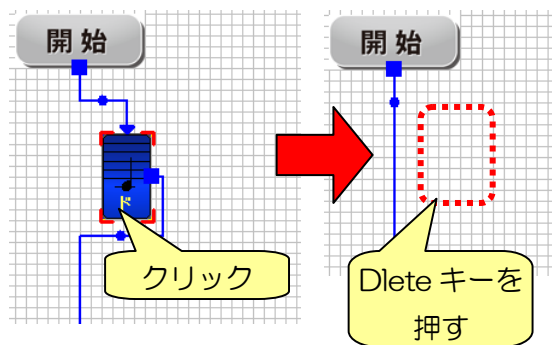


アクションブロックをいくつも使う場合は、マウスでブロックをドラッグして、見やすい位置に動かすようにしましょう。また、いらぬブロックは削除しましょう。

アクションブロックをドラッグすると位置を動かせます。

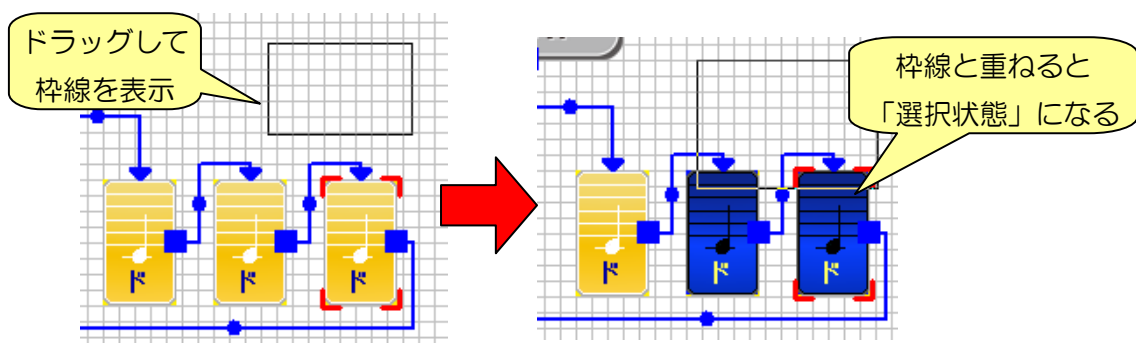


いらぬアクションブロックはクリックしてからキーボードのDeleteキーを押すと削除できます

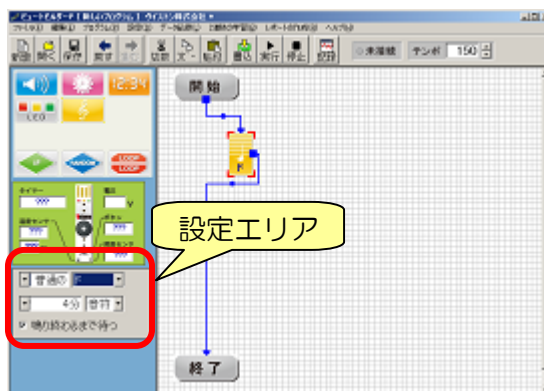


アクションブロックの移動や削除は、「選択状態」の機能を使うと一度にまとめて行うことができます。

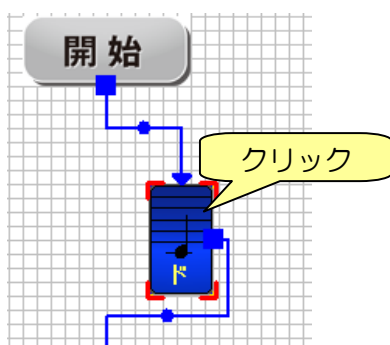
プログラムエリアの何も無いところをドラッグすると枠線を表示します。枠線にアクションブロックを重ねると、色が変わって「選択状態」になります。選択状態のアクションブロックは、まとめて移動や削除ができます。



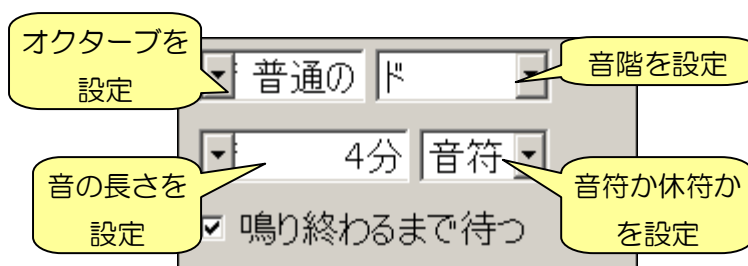
アクションブロックの設定は、画面左下の「設定エリア」で行います。次の説明を参考に、音の高さや長さを変えてプログラムを実行してみましょう。



設定を変更したいアクションブロックをクリックします。

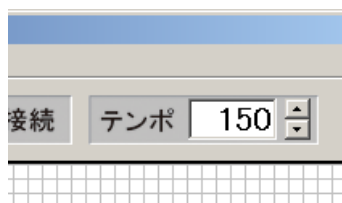


設定エリアで設定の変更を行います。音楽のブロックは次のような設定ができます。



※「鳴り終わるまで待つ」の設定は難しいのでここでは説明しません

音楽のブロックは、設定エリア以外に画面上側の「テンポ」の項目で、音楽のテンポを設定できます。



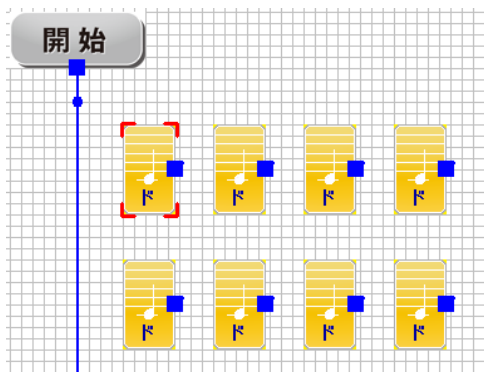
設定エリアの内容は、設定するブロックの種類によって変わります。また、「開始」「終了」のように設定が出来ないアクションブロックも存在します。

# 課題：音楽の命令で順次型プログラムを学習

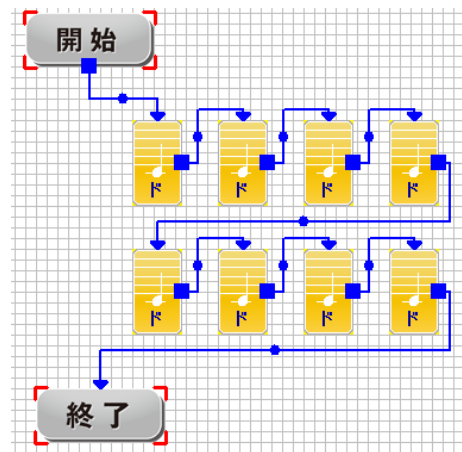
それでは、これまでの説明を元に課題に挑戦してみましょう。まずは例題を一つ解いてみましょう。例題の答えがわからない人は、解説の通りにプログラムを作りましょう。

例題：ブザーを「ドレミファソラシド」の順番で鳴らすプログラムを作ってみましょう。  
ただし、一つの音は8分音符の長さで、テンポは170で作みましょう

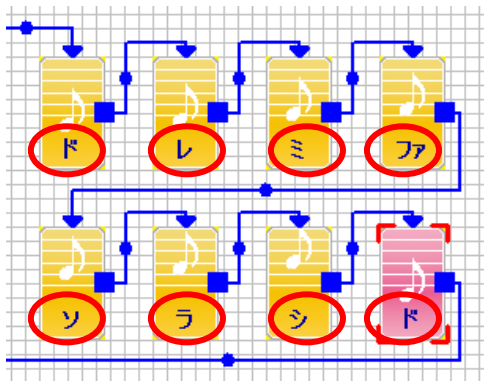
①最初に、必要ブロックを全てプログラムエリアに作ります。図のようにきちんと整列して置くと見やすくなります。



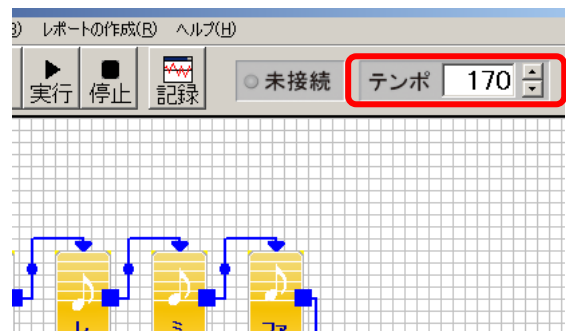
②次に、矢印を接続していきます。最後は忘れず「終了」につなげるようにしましょう。



③設定エリアで、一つずつブロックの設定を変更しましょう。最後のドは、オクターブの設定で「高い」を選択しましょう。



④テンポを設定しましょう。  
設定したら正しくプログラムが出来ているか実行して確認しましょう。



課題1：「かえるのうた」を鳴らすプログラムを作ってみましょう。

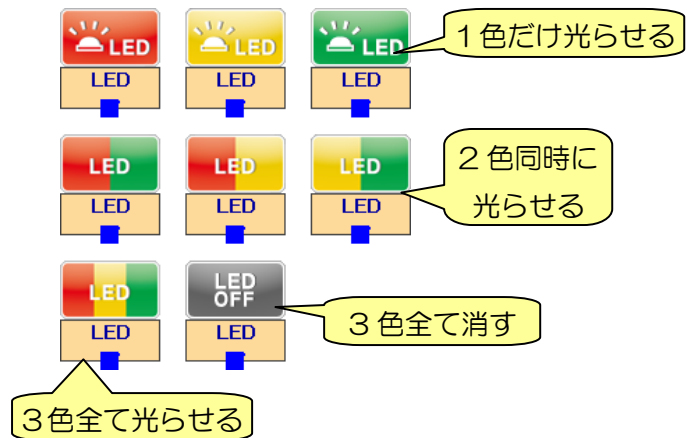
## 8.LEDを使う

本体には、赤・黄・緑の三つのLEDが備わっており、これらはブザーと同じくプログラムからON/OFFを切り替えることができます。基本的にはこれまで説明した音楽と同じように使うことができるので、説明を参考に色々なプログラムを作って見ましょう。

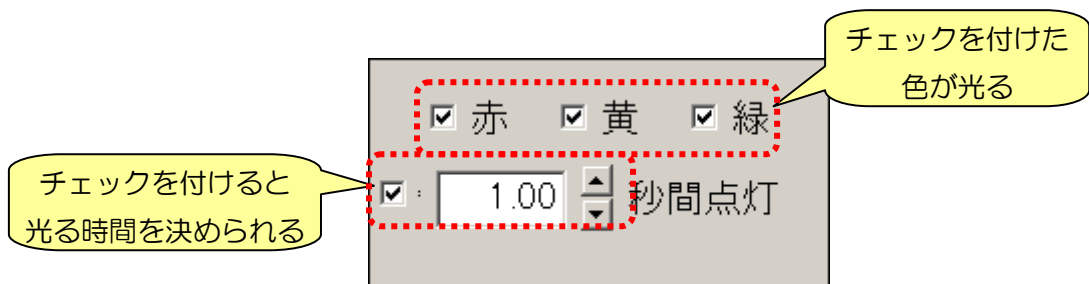
アイコンエリアのLEDアクションブロックは、下図のボタンになります



LEDのアクションブロック。設定によって見た目が変わります。

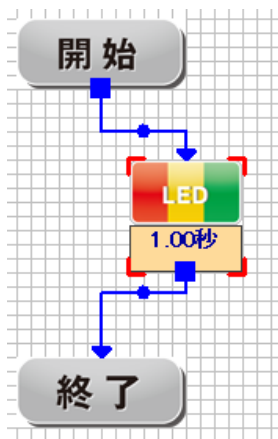


設定エリアの内容は次の通りです。LEDのブロックは、「光る色」と「光る時間」の二つを設定します。

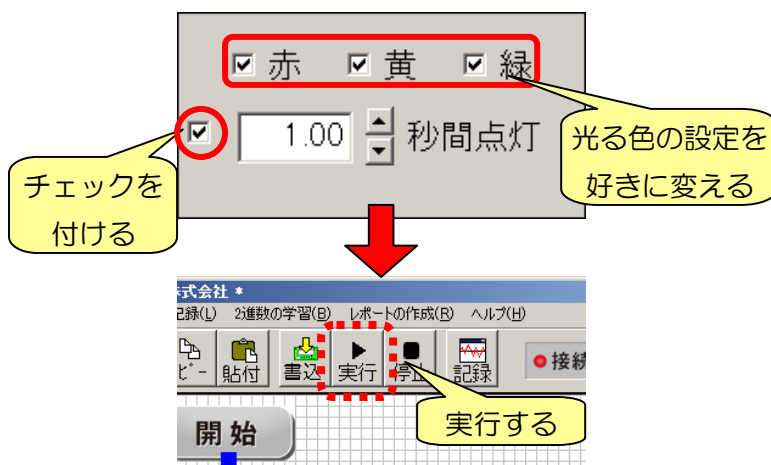


まずは、「光る色」の設定を確認してみましょう。

①プログラムエリアで次のようなプログラムを作りましょう。

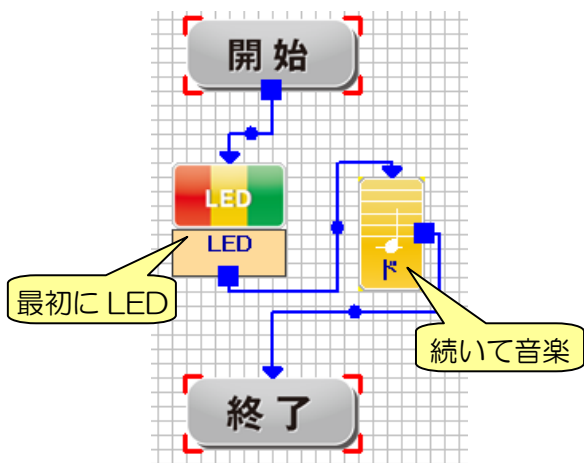


②「光る時間」にチェックを付けて、「光る色」の設定を色々変えてみましょう。設定を変えたら実行してみましょう

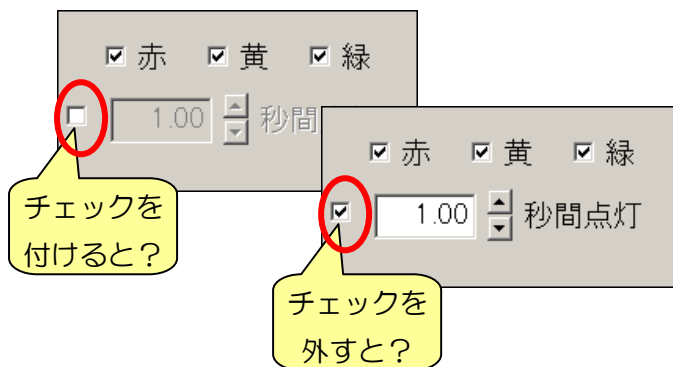


同じく、「光る時間」の設定を確認してみます。こちらの設定は少し難しいところがあるので、本体の動きを確認してしっかり理解しましょう。

①プログラムエリアで次のようなプログラムを作りましょう。

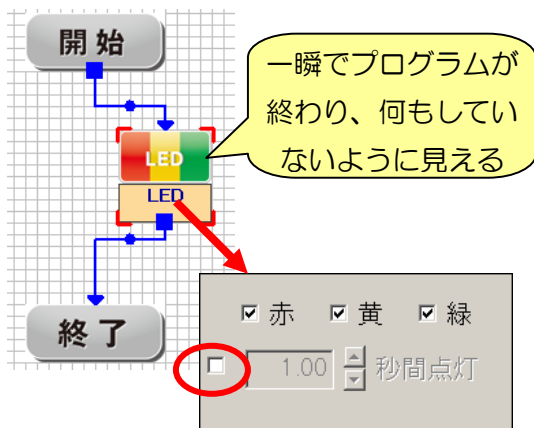


②「光る時間」のチェックを付けた場合と外した場合で、それぞれ実行してみましょう。

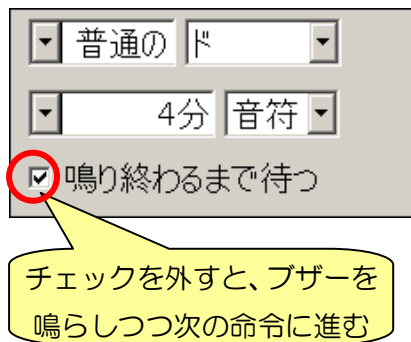


「光る時間」にチェックを付ける・・・LED が光った後にブザーが鳴る  
「光る時間」のチェックを外す・・・LED が光ると同時にブザーが鳴る  
チェックを外すと、LED の命令に時間の設定そのものがなくなるので、LED の命令を実行したらすぐに次の命令に進みます。また、「何秒後に LED を消す」という設定もなくなるので、LED をつけっぱなしでプログラムが進みます。

もし光る時間を設定しないLEDだけのプログラムを作ると、本体は何もせずすぐにプログラムが終わってしまいます



音楽の命令の「鳴り終わるまで待つ」も同じ設定です。チェックを外すと、ブザーを鳴らしつつ次の命令に進みます (ただし、音の長さは設定どおりに鳴ります)

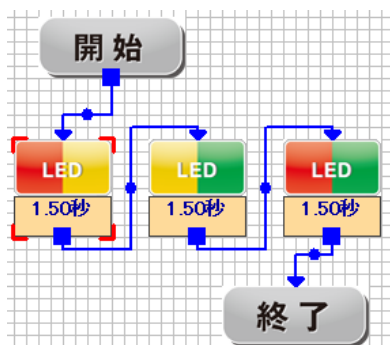


## 課題：LEDのプログラム

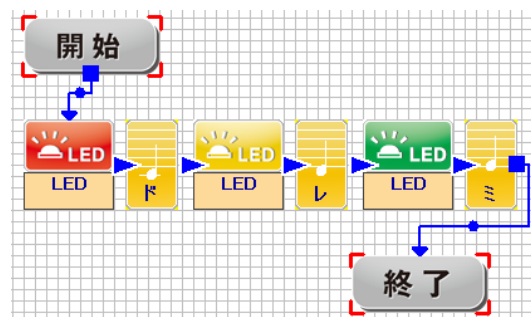
LEDの命令を使ったプログラムや、音楽ブロックと組み合わせた課題にチャレンジしてみましょう。

- 例題1：LEDを赤黄・黄緑・緑赤の順番で1.5秒ずつ光るプログラムを作しましょう。  
 例題2：ブザーをドレミの順番で鳴らしつつ、LEDを赤・黄・緑の順番で光らせてみましょう。  
 ただし、LEDとブザーは同時に切り替わるようにしましょう。

例題1：ここまでの説明を理解していれば、簡単に出来ます。



例題2：LEDの命令で光る時間のチェックを外しましょう。



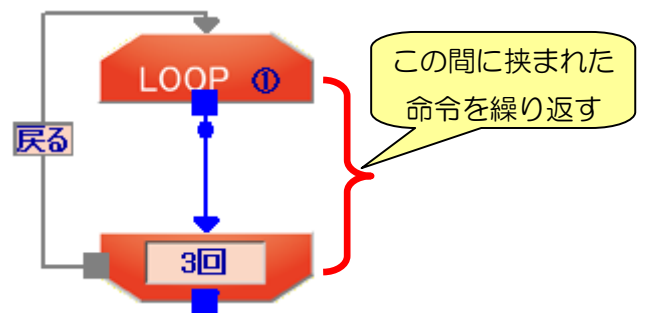
## 9.繰り返しを使う

長いプログラムを作成していると、同じような内容を何回も実行する場面が出てくると  
思います。そういう場合には「繰り返し」を使います。次に説明するサンプルプログラム  
を作りながら、繰り返しの使い方を覚えましょう。

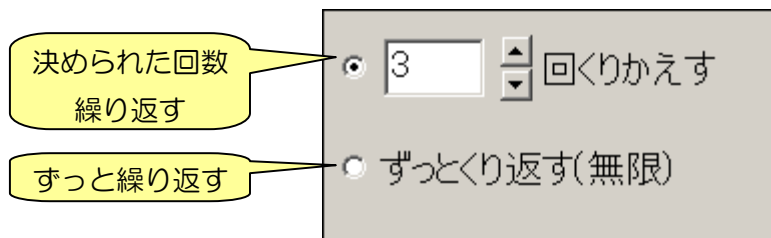
アイコンエリアの繰り返しのアクション  
ブロックは、下図のボタンになります



繰り返しのアクションブロック。  
繰り返す範囲と回数を設定します。

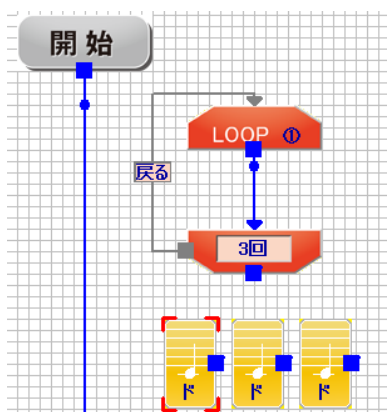


設定エリアの内容は次の通りです。

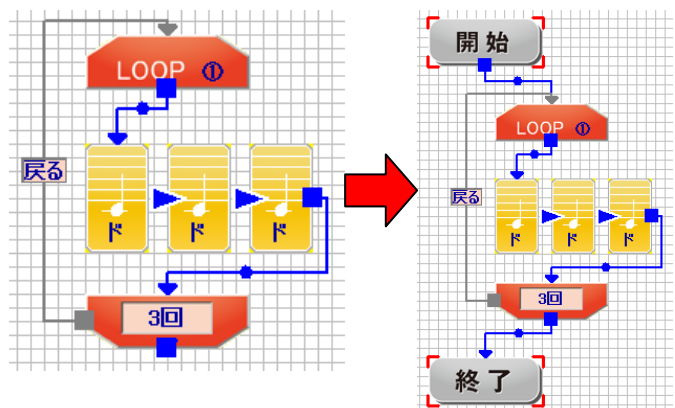


それでは、例として「ドレミ」と4回鳴らすプログラムを作ってみましょう。

①必要なアクションブロックを  
追加します。

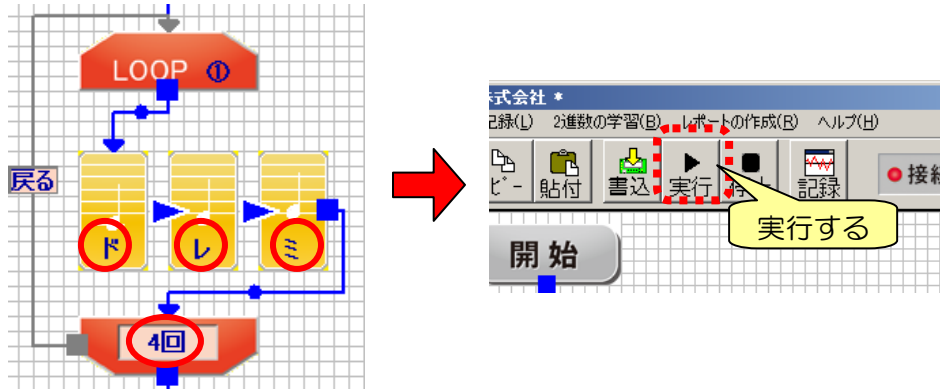


②矢印を接続します。繰り返しのブロックの中に  
音楽のブロックをつなぎ、開始・終了の  
ブロックを繰り返すにつなぎましょう。



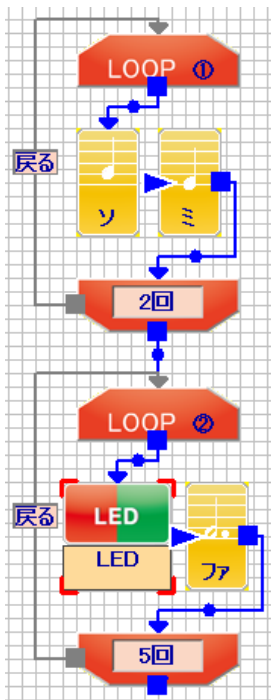


③繰り返しと音楽の設定をそれぞれ変更します。  
設定が終わったら実行してみましょう。

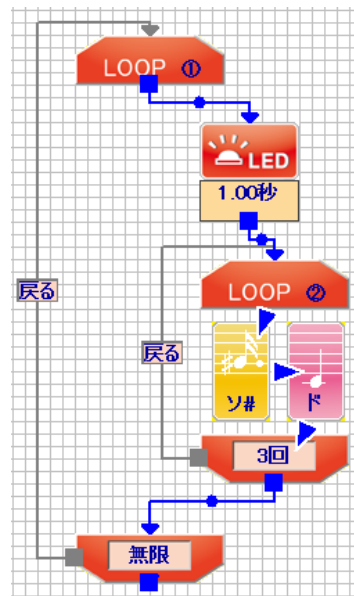


繰り返しのアクションブロックは、最大で7個まで使えます。また、「繰り返しを連続で並べる」「繰り返しの中に繰り返しを入れる」ということも出来ます。

繰り返しを連続で並べることが  
出来ます。



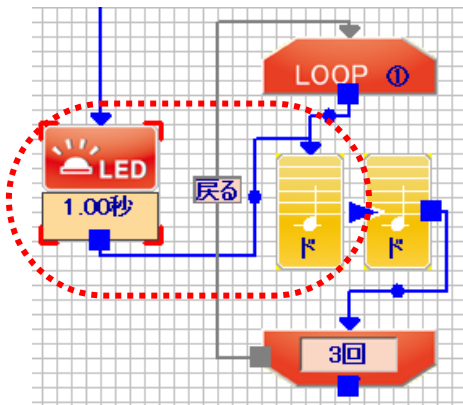
繰り返しの中に繰り返しを  
入れることが出来ます。



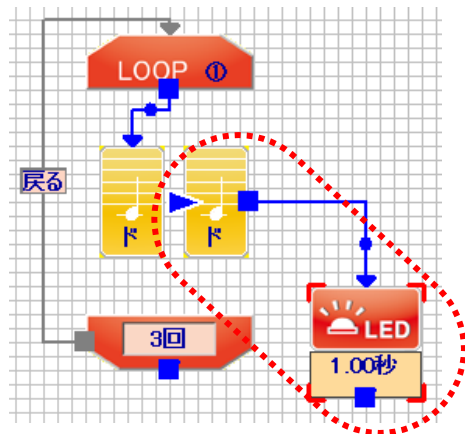
# 10.間違えやすい繰り返し

繰り返しのアクションブロックは、矢印のつなぎ方を間違えると正しく動かなくなりま  
す。繰り返しを使う場合、次のような間違いをしないように注意しましょう。

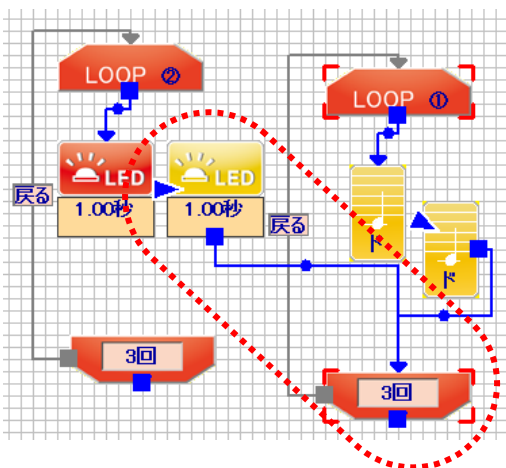
繰り返しの開始ブロックを飛ばして  
つなぐと、正しい回数繰り返しません



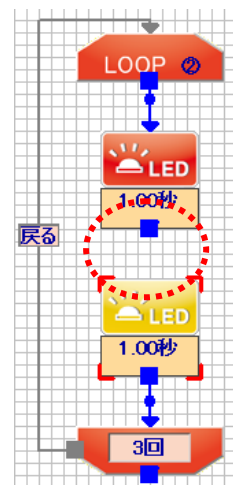
繰り返しの終了ブロックを飛ばして  
つなぐと、そこで繰り返しが終わります。



繰り返しの開始・終了ブロックの組み合わせ  
を間違えると正しい回数繰り返しません



繰り返しの開始・終了ブロックの途中で矢印が  
途切れていると、そこでプログラムが終了します



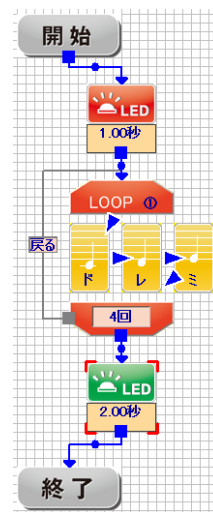
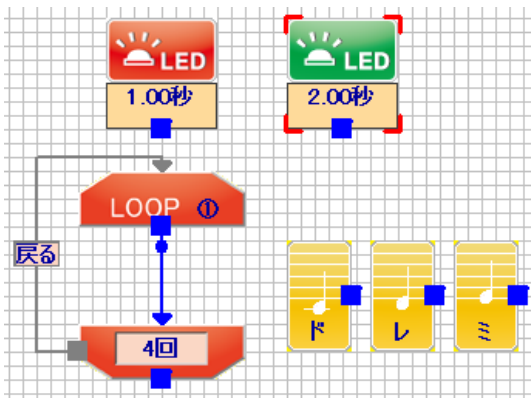
# 課題：繰り返しを使ったプログラムの課題

繰り返しを使ったプログラムの課題にチャレンジして見ましょう。

例題：最初に赤いLEDを1秒光らせ、ブザーで「ドレミ」を4回繰り返し、最後に緑のLEDを2秒光らせるプログラムを作りましょう

①必要な命令をプログラムエリアに追加し、設定を行います。

②矢印をつなぎます。繰り返しに入るもの・入らないものをしっかり区別しましょう。



課題：繰り返しを使って歩行者用信号を再現してみましょう。  
LEDの点滅が次の順番で切り替わるのを繰り返すプログラムにしましょう。

1. 緑が10秒間ON（出来ればブザーも鳴らしてみましょう）
2. 緑が0.5秒間隔で5秒間点滅
3. 赤が10秒間ON

《ヒント》繰り返しは、全体の繰り返し・点滅の繰り返しの2つ使います。

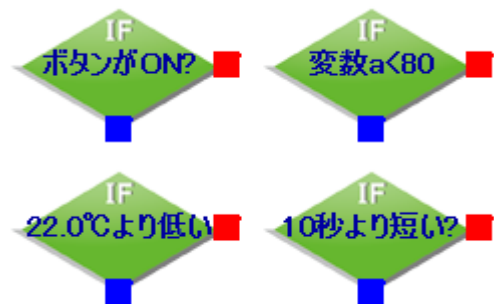
# 11.分岐を使う

本体には、ボタン・タイマー（時間）・温度センサ・明るさセンサなど、色々な情報を測る機能が備わっており、それらの状況に応じてプログラムの途中で動作を変えることができます。

アイコンエリアの分岐アクションブロックは、下図のボタンになります



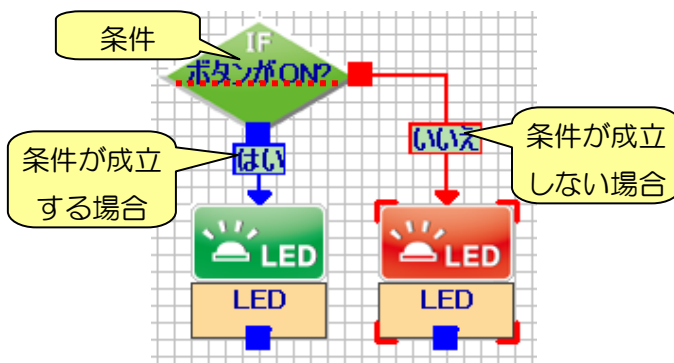
分岐のアクションブロック。ボタン・タイマー・温度・明るさの4つに対して反応できます。



分岐のブロックには、「はい」「いいえ」の二つの矢印があります。また、設定エリアでは「ボタンが押されているか?」「温度が25°C以上か?」など、何らかの条件を一つ決めます。プログラムを実行したら、そのときの本体の状態によって、条件が成立する場合は「はい」、成立しない場合は「いいえ」に進みます。

分岐のブロックには「はい」「いいえ」の二つの矢印があり、プログラムを実行したときに、ブロックで決めた条件が成立するかどうかで、どちらに進むかが決まります。

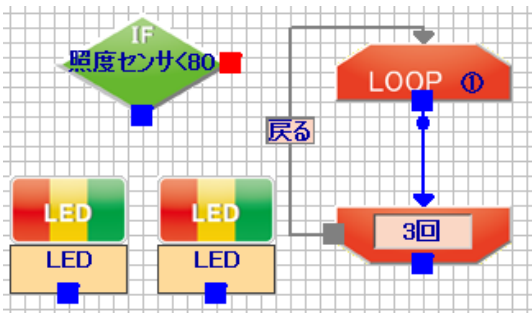
分岐のアクションブロックは、設定エリアで分岐の条件を設定します。下図は、一番簡単なボタンのON/OFFに関する条件です



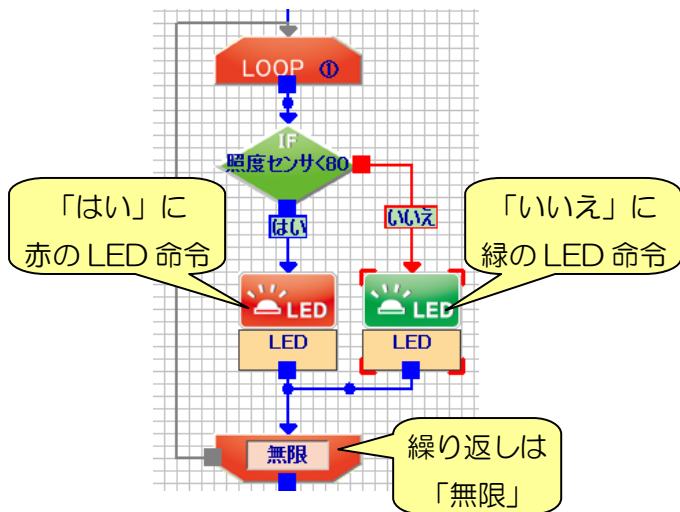
- 温度 ○ 時間 ○ ボタン
  - 明るさ・その他
- ボタンの状態が
- ON ○ OFF
- であるかどうか

それでは、簡単なサンプルとして「ボタンを押したら赤いLEDをON・押しなければ緑のLEDをONにする」というプログラムを作ってみましょう。

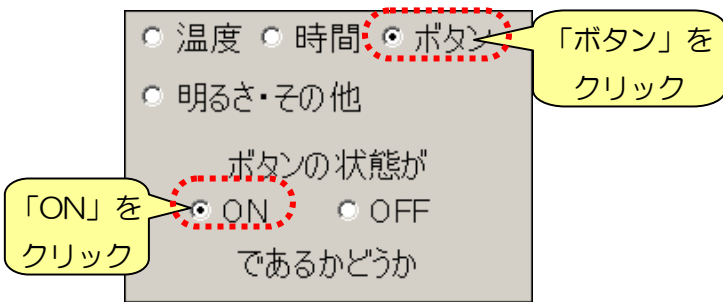
①必要な命令をプログラムエリアに追加します。すぐにプログラムが終了しないよう、繰り返しも使います



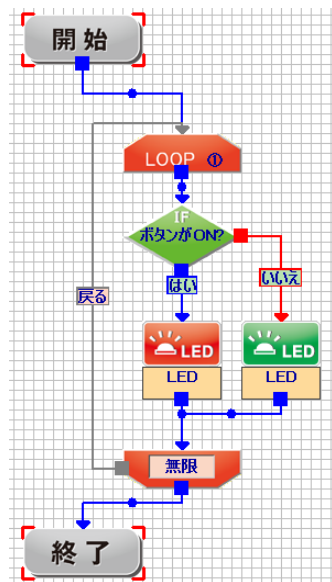
②矢印をつないで、LEDと繰り返しの設定を図の通りに変更します。



③分岐ブロックの設定で、「ボタン」をクリックし、条件として「『ON』であるか」をクリックします。



④プログラムを実行します

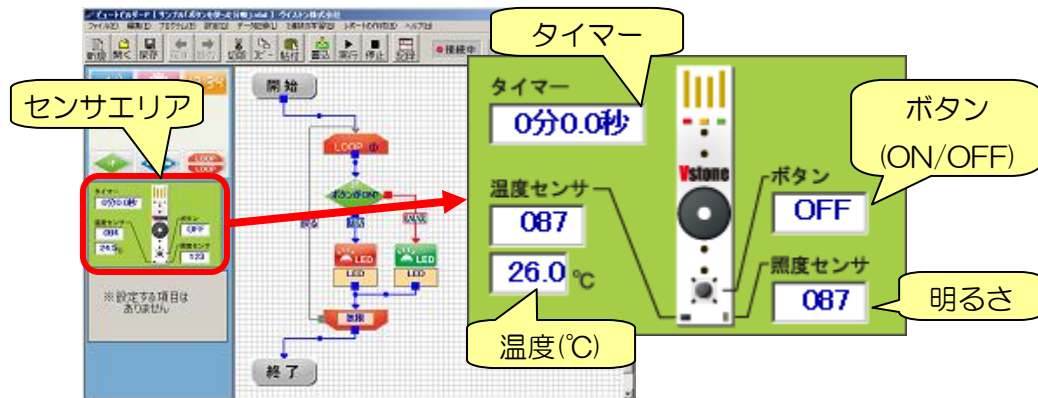


※2秒以上ボタンを押すと電源がOFFになるので、ボタンは短めに押しましょう。

正しく動くのを確認できたら、次の変更をするとどうなるか調べてみましょう。  
 Q 「はい」と「いいえ」の矢印をつなぎ変えると・・・？  
 Q 条件の設定を「ボタンの状態が『OFF』であるか」に変えると・・・？

## 12. センサの数値を調べてみる

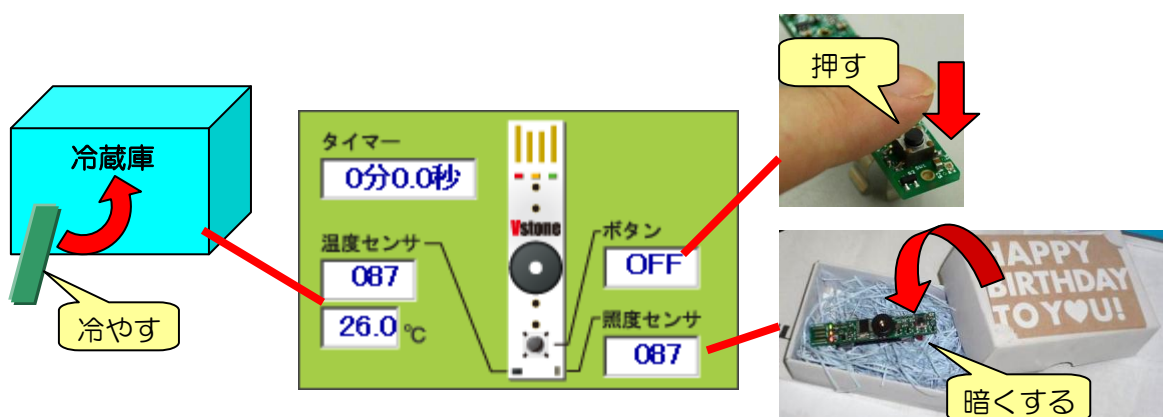
分岐のアクションブロックの説明でも書いていますが、本体にはボタン・タイマー・明るさ・温度を測ることが出来ます。これらの情報は、本体がPCと通信しているときに、画面中の「センサエリア」に表示されます。



この中で、ボタン・温度・明るさは、本体の周りの状況に応じて数値が変わります。

本体をPCと通信させて、次のことを試してみましょ。

1. ボタンを押してみる・・・ONになりますか？
2. 冷蔵庫に入れてみる・・・温度が下がりますか？
3. 箱にしまってみる・・・明るさが小さくなりますか？



### 《注意！》

- ※ 温度はセンサを直接触ると値が不安定になります。
- ※ ボタンを長く押しと電源がOFFになります。
- ※ 本体を動かすと通信が途切れる場合があります。センサエリアの数値がまったく変わらなくなったら、一度本体を抜き差ししてください。

# 13.温度に反応するプログラムを作る

先ほどの説明の通り、分岐の命令は温度・明るさにも反応することが出来ます。それでは、温度や明るさに反応した分岐のプログラムを作ってみましょう。

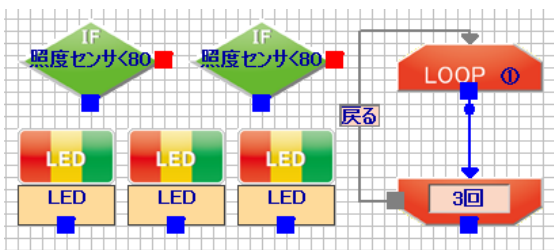
ボタンを使った分岐では、「ボタンがONか」または「OFFか」という条件を設定しました。しかし、温度や明るさは、ON/OFFのようにはっきりした二つだけの状態ではなく、いろんな数値が出てきます。そこで、これらを使った分岐では、「センサの値が決められた数値よりも大きいか・小さいか・同じか」という条件を使います。

例：「温度が 20℃よりも大きいですか？」「明るさは 100 よりも小さいですか？」  
これを数式になおすには？ → 不等号（<、>）を使って考えてみましょう。

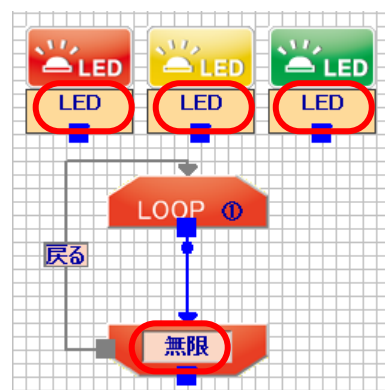
例：「温度 > 20℃ ?」「明るさ < 100 ?」

サンプルとして、温度に反応してLEDの光り方が変わるプログラムを作ってみましょう。また、温度は ON/OFF 以外の状態にもなることを利用して、「15℃より低ければ緑、15℃～20℃なら黄、20℃より高ければ赤のLEDを光らせる」というプログラムにしてみたいと思います。

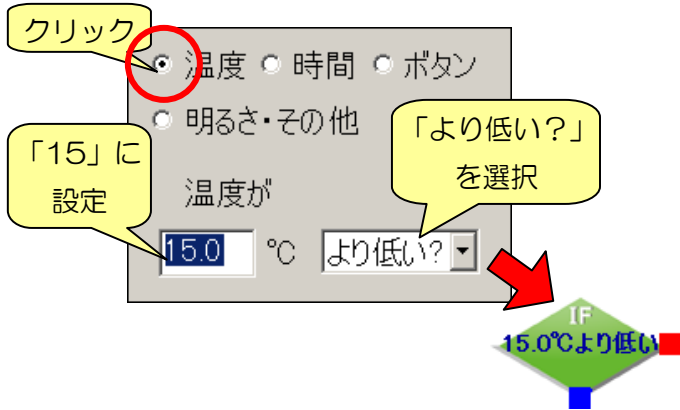
①プログラムに必要なブロックを追加しましょう。今回は分岐ブロックが二つ必要です。



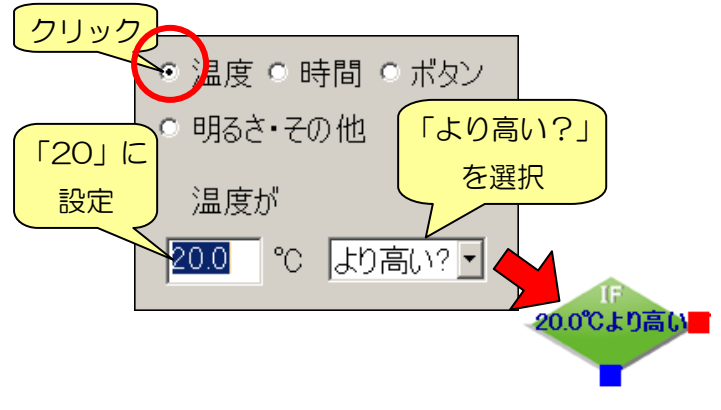
②まずは、LEDと繰り返しの設定を図の通りに行います。



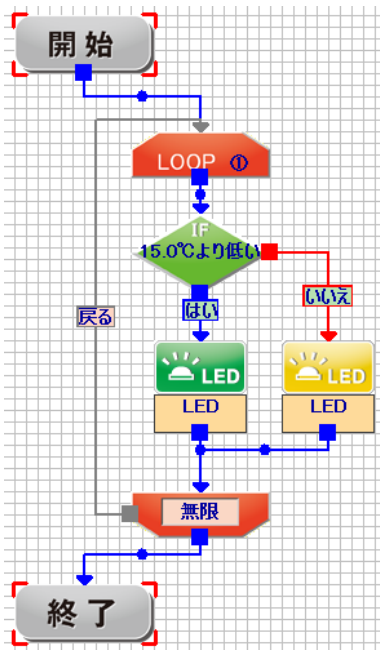
③一つの分岐ブロックを「温度が 15 度より低い?」という条件に設定しましょう。  
(設定方法は図を参考にしましょう)



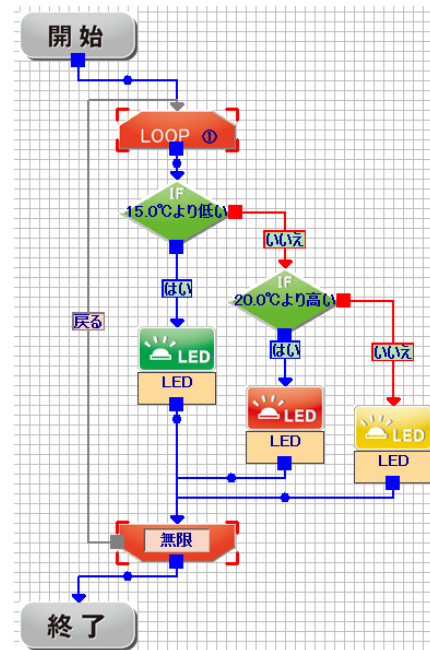
④もう一つの分岐ブロックを「温度が 20 度より高い?」という条件に設定しましょう。  
(設定方法は図を参考にしましょう)



⑤次に矢印を接続します。まずは下図のようにつないで実行してみましょう。このプログラムは「15℃より低ければ緑・そうでなければ黄のLEDが光る」というプログラムですね。



⑥更に、まだつながっていないブロックを下図のようにつなぎましょう。これでプログラムは完成です。正しく動作するか確認してみましょう。



このプログラムでは、二つの条件をつなげることでより複雑な条件に対応させています。  
次の説明のように、分岐をつなげると、最初の分岐の条件が結果的に後の分岐にも反映される仕組みです。

- 1. 「15℃より低い?」 → はい=緑 (緑り返しへ) いいえ=黄か赤 (2へ)
- 2. 「15℃以上で20℃より高い?」 → はい=赤 いいえ=黄

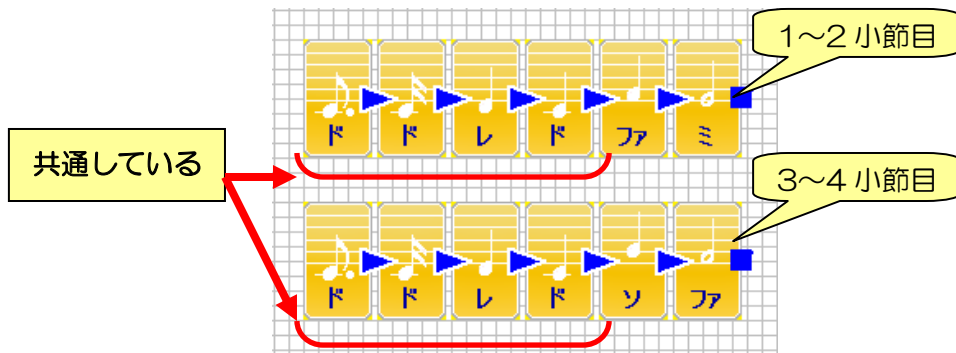
《注意!》分岐の条件の設定、矢印のつなぎ方など、どれか一つでも間違えると思ったプログラムになりません。設定した条件と「はい」「いいえ」の接続先がかみ合っているかよく確認しましょう。



# 14. ブザーで音楽を作ってみる

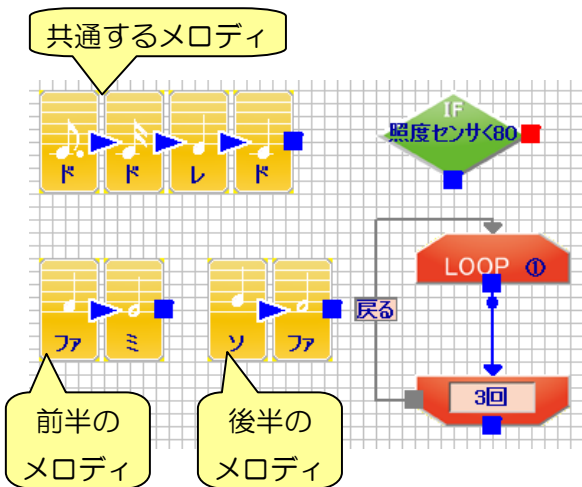
皆さんの良く知っている音楽には、同じフレーズを繰り返したり、途中まで同じフレーズを演奏して、最後の方で少し違う演奏をするような曲が多くあります。このような音楽をプログラムで演奏するとどうなるでしょうか？

下のプログラムは、ハッピーバースデーの音楽を 4 小節目まで作ったものです。よく見ると、1~4 番目と 7~10 番目の音がそれぞれ共通しています。

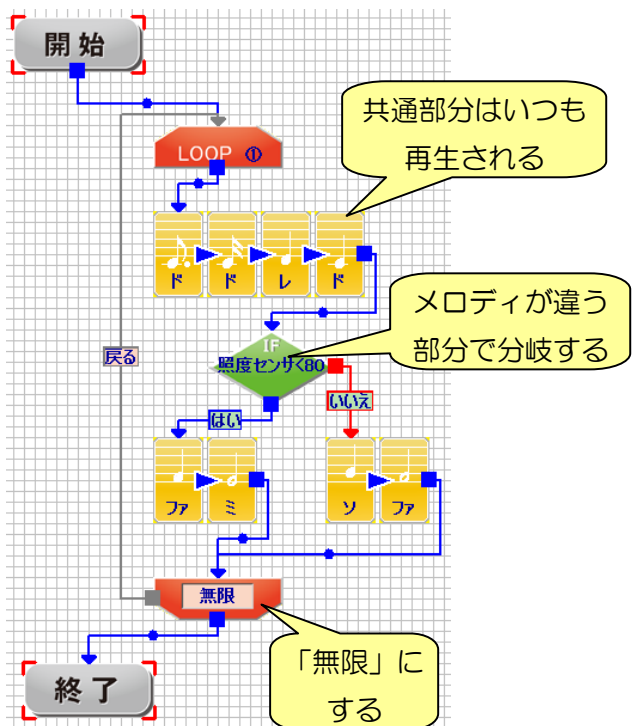


共通した部分は繰り返しを使ってブロックの数を節約し、メロディーが異なるところは分岐を使って区別すると、うまく音楽を鳴らせます。繰り返す回数で分岐をさせる場合は「変数」という難しい機能を使う必要がありますが、今回は代わりに明るさを使って「明ければ 3~4 小節目・暗ければ 1~2 小節目を演奏」というプログラムにします。

①必要なブロックをプログラムに追加します。また、音楽を作っておきます。



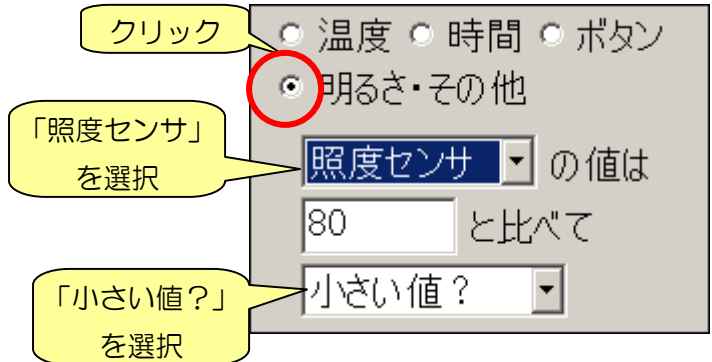
②矢印をつなぎます。共通するところとしないところの区別をつけましょう。また、繰り返しは「無限」に設定します。



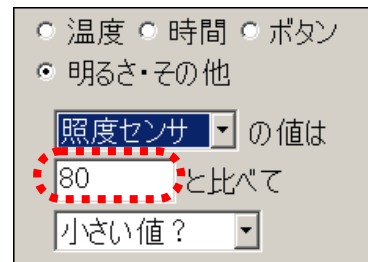
③分岐の条件を設定します。まずは、  
 明るいときと暗いときでセンサの値が  
 いくつになるか確認しましょう  
 (※数値はちゃんと自分の目で見て確  
 認しましょう！)



④分岐の条件を設定します。  
 まずは数値以外を、下図の通りに設定します。

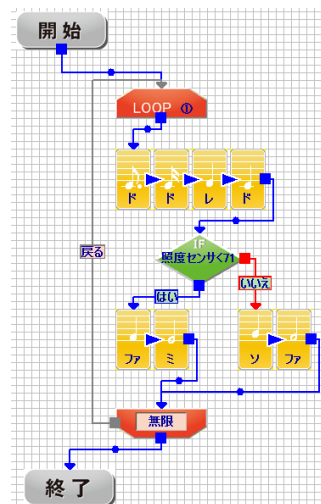
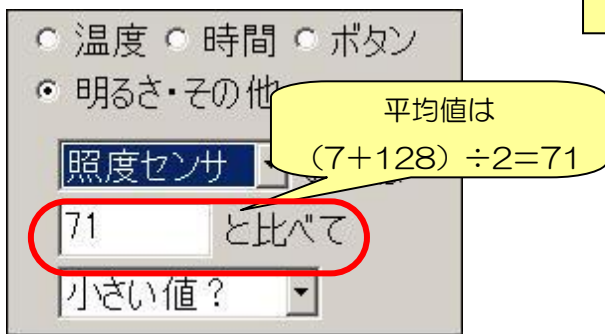


⑤最後に条件に使う数値を設定しますが、  
 いくつにすればうまく動くでしょうか?



先ほど調べた数値では、明るさが暗い時は「7」、明るいときは「128」になりました。  
 →じゃあ、8以上なら明るい? / 127以下なら暗い?  
 《暗くても10くらい出るかもしれない・明るくても110くらい下がるかもしれない!》  
 どちらにもまず間違われることの無いよう、真ん中くらいの値が良い=平均値を求めよう!

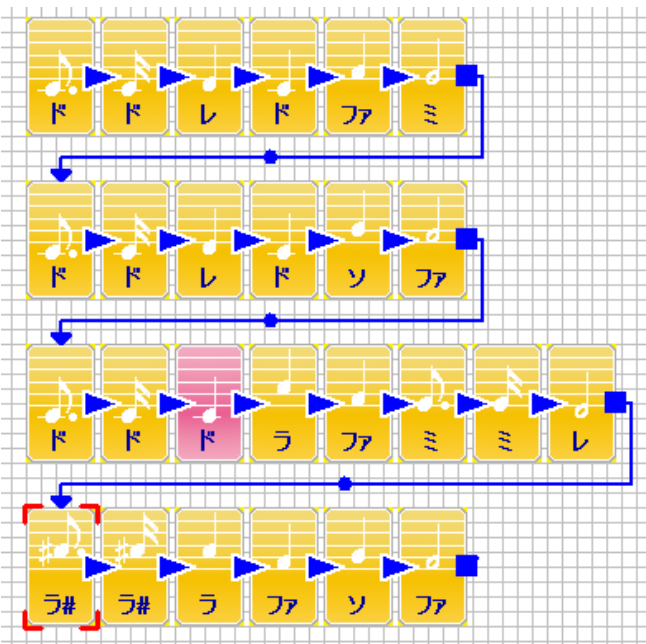
プログラムが出来たら実行してみましょう



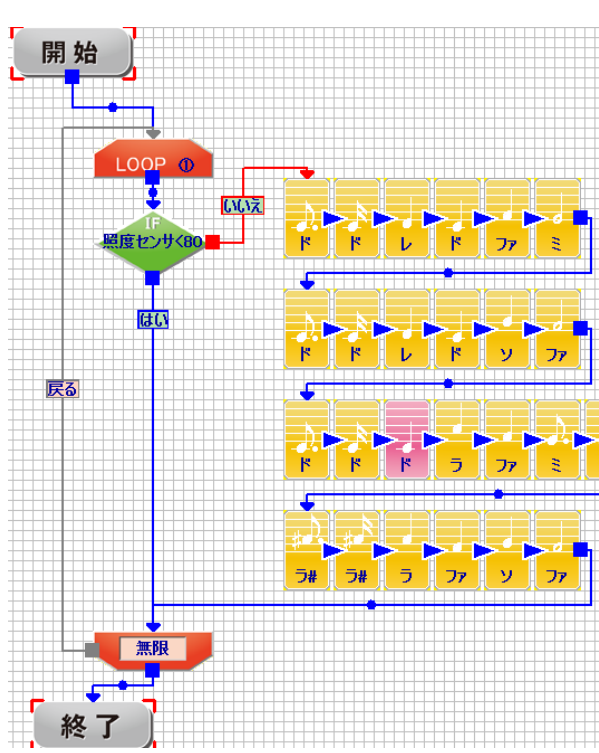
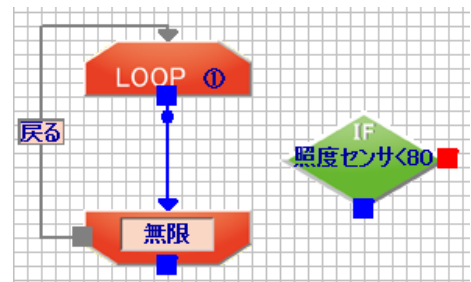
# 課題：電子オルゴールを作ってみる

ここまで学習してきたことを全て使って、電子オルゴールを作成してみましょう。オルゴールは、箱を開けるとスイッチが入って音が鳴り出す仕掛けですが、本体にはスイッチがありません。そこで、「箱が開いたかどうか」を明るさで区別します。

①まずハッピーバースデーの音楽を作りましょう。先ほどの例題では繰り返しや分岐を使って作りましたが、今回は全て音楽ブロックを並べて作ります



②他に必要な命令として、分岐と繰り返しを一つずつ追加します。これで全てのブロックが出揃いました。



③ブロックの矢印を接続します。「条件が成立するかどうか」で音楽を鳴らす場合は、左図のような接続になります。

④分岐の条件を設定します。数値以外を、下図の通りに設定します。  
数値は、先ほどの例題の通り「箱のふたを閉じる・開ける」のそれぞれの場合でセンサエリアの明るさを確認し、平均値を計算して設定しましょう。

クリック

温度  時間  ボタン  **明るさ・その他**

「照度センサ」を選択

照度センサ の値は

と比べて

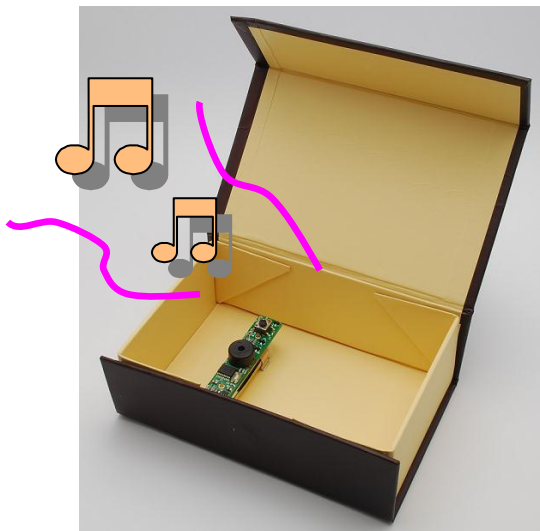
「小さい値?」を選択

小さい値?

ふたを開けた時

ふたを閉じた時

二つの明るさの平均値を計算



完成したら、箱に飾り付けをして友達に見せてみましょう。

計測制御プログラマーとビュートビルダーPには、ここで紹介していない便利な機能がまだまだ備わっています。「ビュートビルダーP 取扱説明書」を参考にして、より難しいプログラムにもチャレンジしてみましょう。