



Mindstorm 講習会

東京工業大学レゴ同好会・日本女子大学レゴ同好会

2016/7/7 (Thu) 東京工業大学 大岡山キャンパス

制作 東京工業大学レゴ同好会

目次

- 基本
- 構文（条件分岐と繰り返し処理）
- 構文の組み合わせ
- インテリジェンスブロックの利用
- センサの取り扱い
- 変数の取り扱い
- 拘束時間・並列処理（おまけ項目）

基本 1

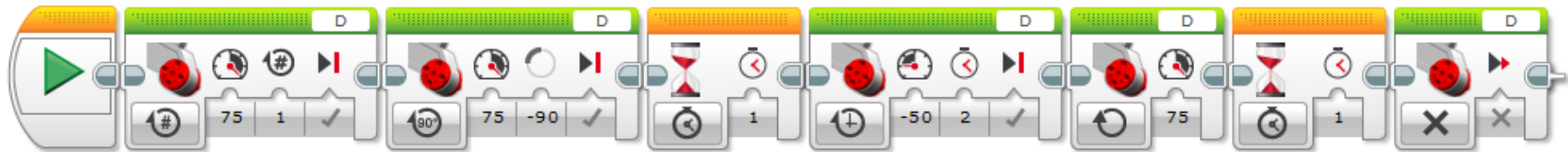
The screenshot shows the LEGO MINDSTORMS EV3 Home Edition software interface. The window title is "LEGO MINDSTORMS EV3 Home Edition". The menu bar includes "File", "Edit", "Tools", and "Help". The interface is divided into several sections:

- Project and Program Tabs:** Located at the top left, showing "Project" and "Program" tabs. Annotations point to these tabs with the labels "プロジェクトファイル" (Project File) and "プログラム" (Program).
- Toolbar:** Located at the top right, containing icons for undo, redo, zoom, and search. Annotations point to the undo icon with the label "一つ戻る (undo)" and to the zoom icon with the label "ズームとか" (Zoom, etc.).
- Edit Area:** The central workspace for building programs, annotated with "エディット画面" (Edit Screen).
- Programming Blocks:** A palette at the bottom left containing various functional blocks, annotated with "プログラミングブロック" (Programming Blocks).
- EV3 Connection Panel:** A panel at the bottom right showing the connection status of the EV3 brick, including "EV3", "Firmware: V1.03E", and "Connection Type: USB". Annotations point to this panel with the labels "EV3との接続画面" (EV3 Connection Screen) and "EV3へ書き込み" (Write to EV3).

The Windows taskbar is visible at the bottom of the screen, showing the time as 22:35 on 2016/07/04.

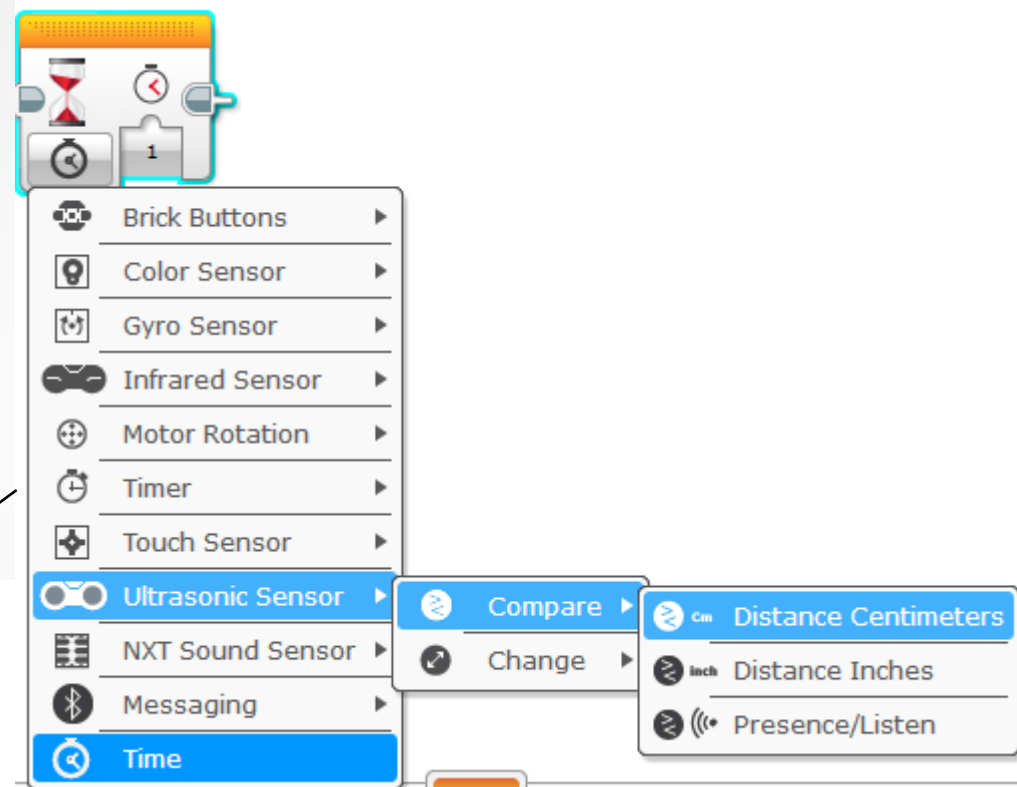
基本 2

- プログラミングブロックを順に並べる
→ EV3に書き込んで実行
→ 「前から」 「ひとつずつ」 実行される



基本 3

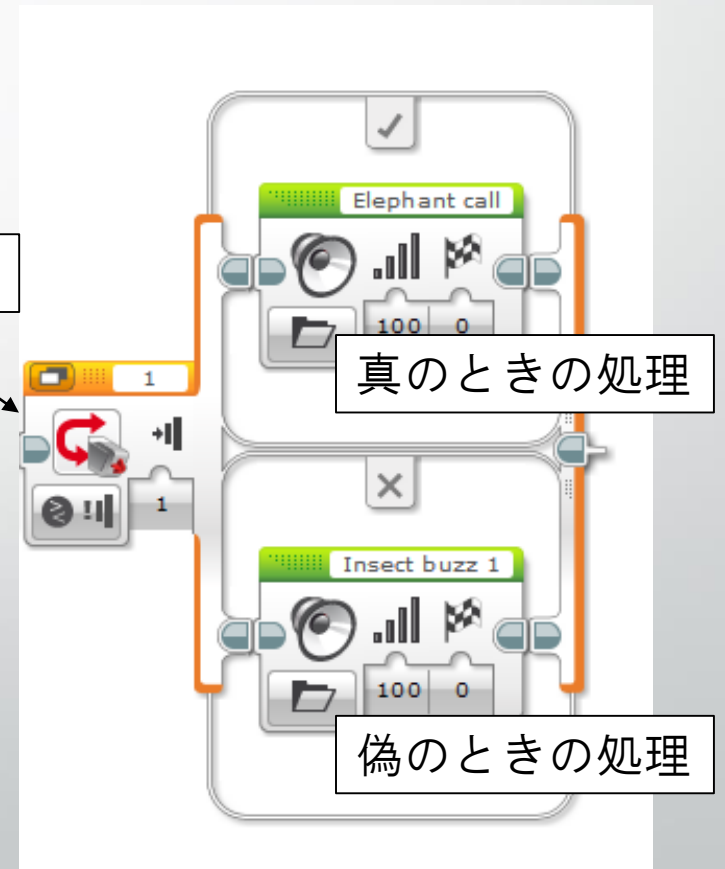
- 「センサ」 + 「待機」 → 「入力待ち」



分岐（if 構文）

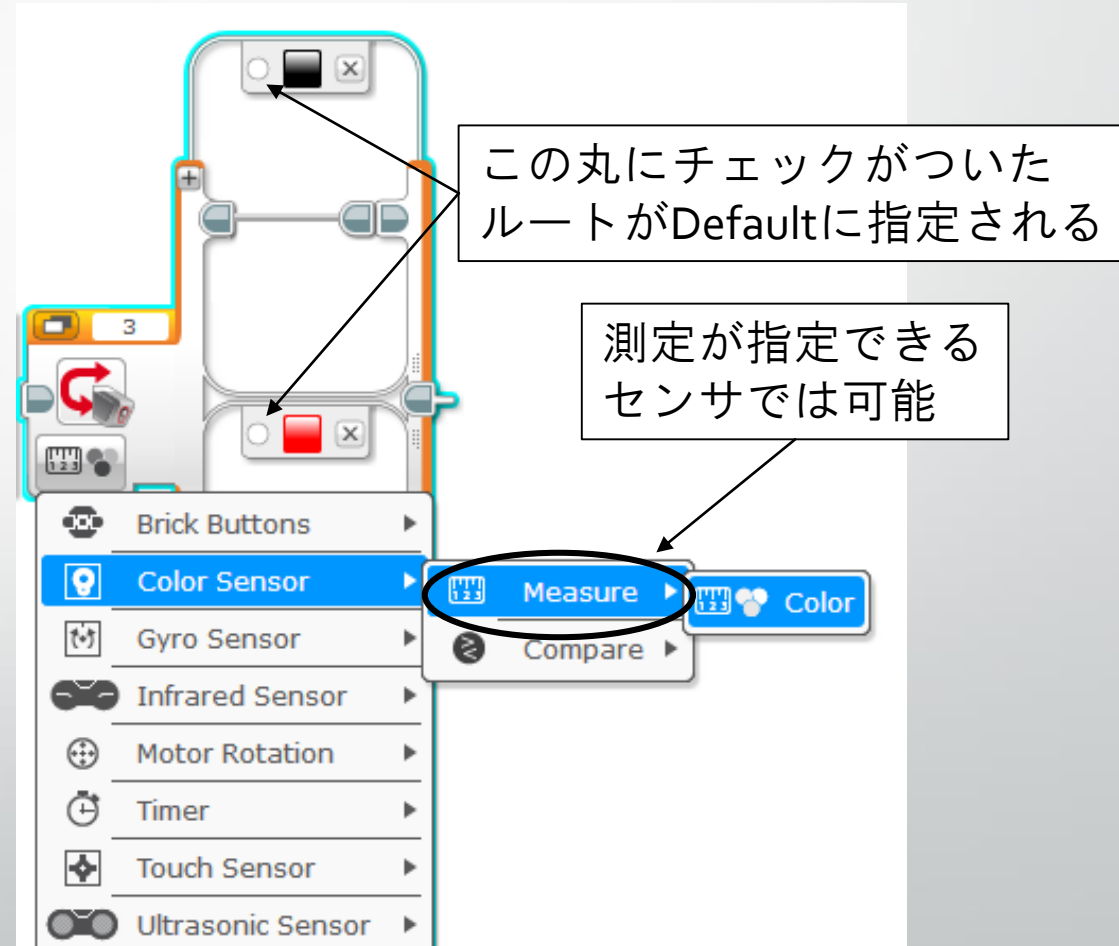
- 条件（状態・数式など）を設定
→ 満たしていれば「真（true）」
→ 満たしていなければ「偽（false）」
- 必ずどちらかに一方にのみ入る

分岐の条件



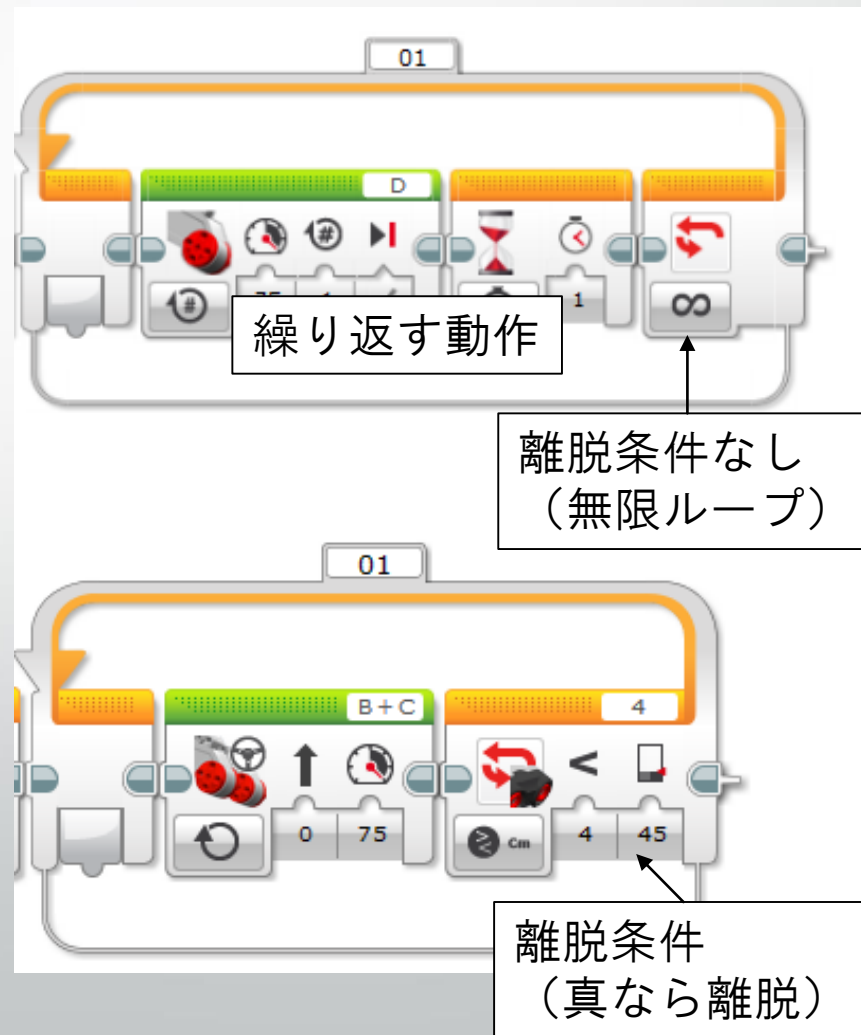
分岐（if 構文）

- 3つ以上の分岐
=分岐条件が真・偽ではないとき
(Ex: カラーセンサで色を判別)
- MindstormEV3では、
「センサ→測定」と「テキスト」、
「数値」で利用可（2016/7/7時点）
- 「偽」に相当する「default」を
必ず指定する必要がある



ループ処理

- 特定の処理を繰り返すことができる
＝「繰り返す回数」と「繰り返す動作」
- 「ループの離脱条件」を指定
→離脱条件が「真」になると離脱して
次のブロックへ



ループ処理（Break処理）

- 「ループを離脱する」という処理をするブロックが存在
→条件分岐との組み合わせが必須
- なぜBreakが存在??
 - ・ 離脱条件を複数指定できるようになる
 - ・ 離脱条件の判定タイミングを指定できる
 - ・ プログラムをより見やすくできることがある（switch-case文式）
 - ・ 離れた位置にあるループの中断ができる（並列処理などで便利）



構文の組み合わせ

- 分岐と分岐の組み合わせ → 樹形図、Akinator ...
- 多重ループ → 掛け算九九表 ...
- 分岐とループの組み合わせ → だいたいのもの

- 所望の動作を分岐とループに分解して順に書き下せるかがカギ

練習問題 1

- (1) 2つのタッチセンサを用いて、
Lモータの正転、逆転、停止を操作するコントローラを作る

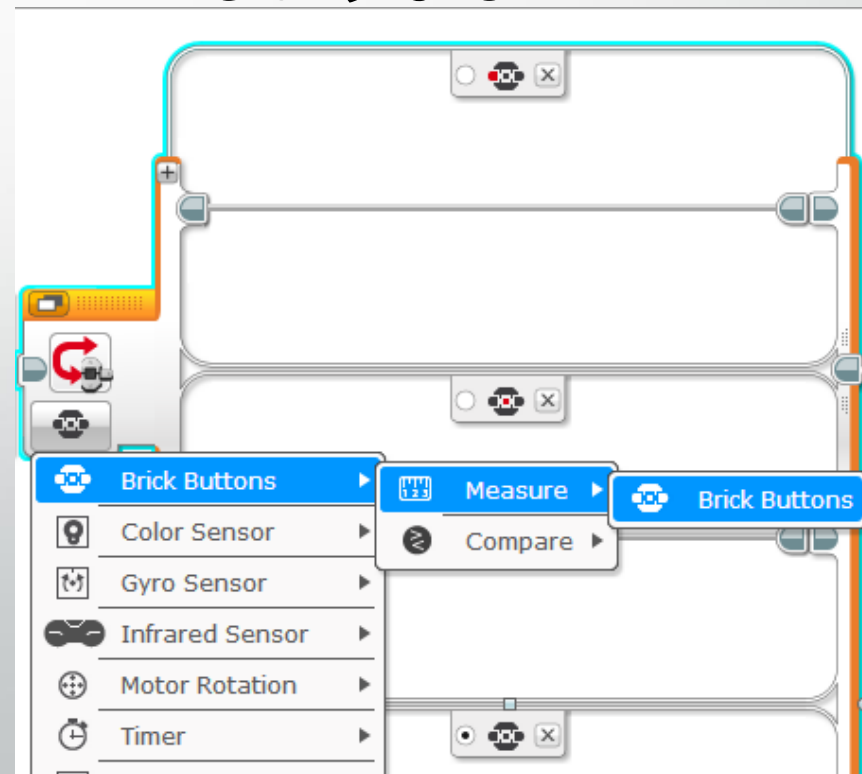
- (2) 超音波センサを用いて物体との距離を測定し、
①30 cm 以下のとき、②30 ~ 50 cm のとき、③50 cm 以上のとき
で異なる音が鳴る楽器を作る (※音量に注意)

インテリジェンスブロック

- インテリジェンスブロック=EV3本体
→画面とボタンとLEDとスピーカーが付いている=利用できる
- ここでは画面とボタンについて説明
(LEDやスピーカーは簡単なので暇なときにやってみて下さい)

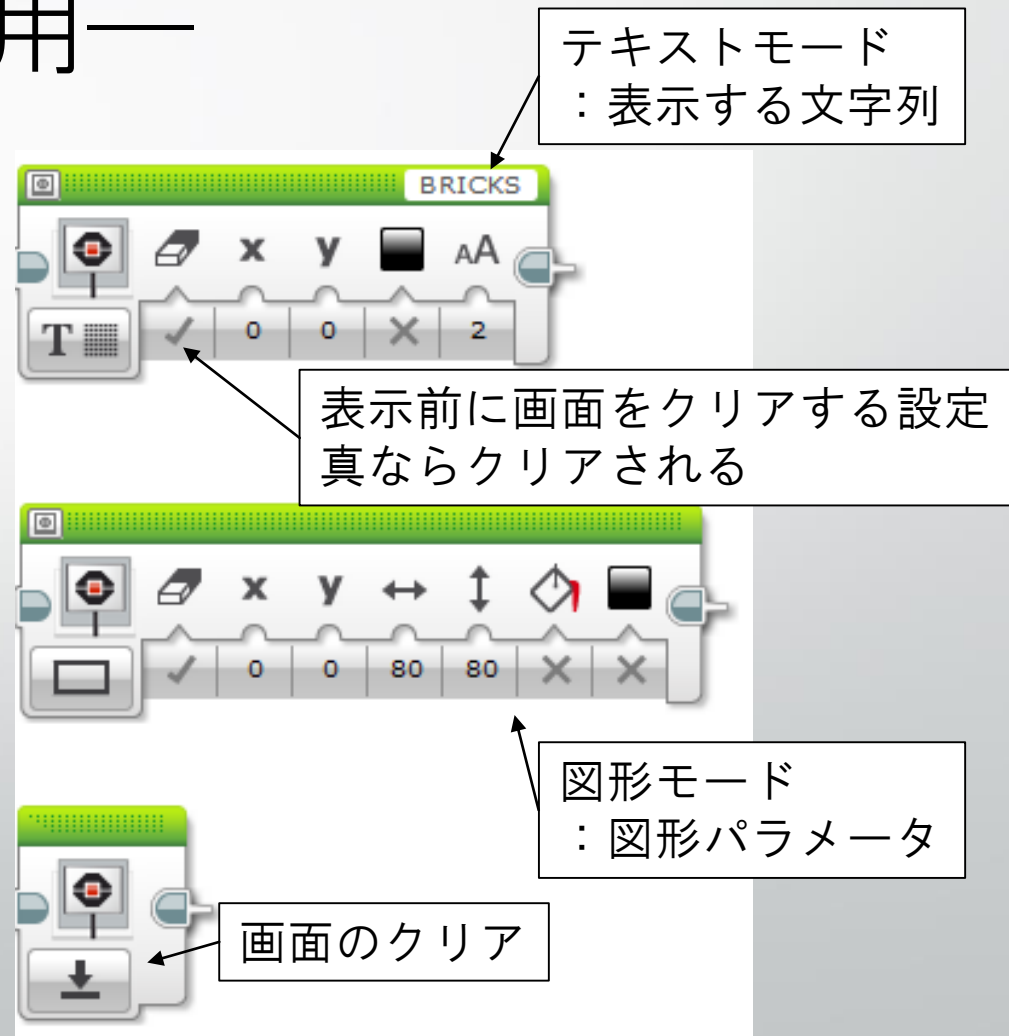
インテリジェンスブロック —ボタンの利用—

- 大雑把に言うと、タッチセンサが5個ついているようなもの
→センサポートの節約ができる
- 特徴
 - ・ 多岐分岐を利用しやすい
 - ・ 操作には向く反面、計測には不向き



インテリジェンスブロック —画面の利用—

- 画面に文字や図形を表示できる
- テキスト、図形など各モードを選択し
必要なパラメータを設定する
(画面のクリア設定や座標、大きさなど)



センサの利用

- そもそもセンサとは…??
＝何かを測定する装置
- 測定 → データが出てくる → データを利用して何かをする
- EV3ではどうやって測定したデータを利用すればいいのか

データワイヤー

- データワイヤー
= EV3 (LabVIEW) において、
数値などのデータを引き出して
他の場所で利用できるようにする
- 右の例は、超音波センサで測定
した距離を画面に表示するもの

テキストモード

: 表示する文字列をワイヤーで受け取るモード



下向き凸みみたいな形が引き出し可能の目印



測定した数値がワイヤーで
表示ブロックに送られる

練習問題 2

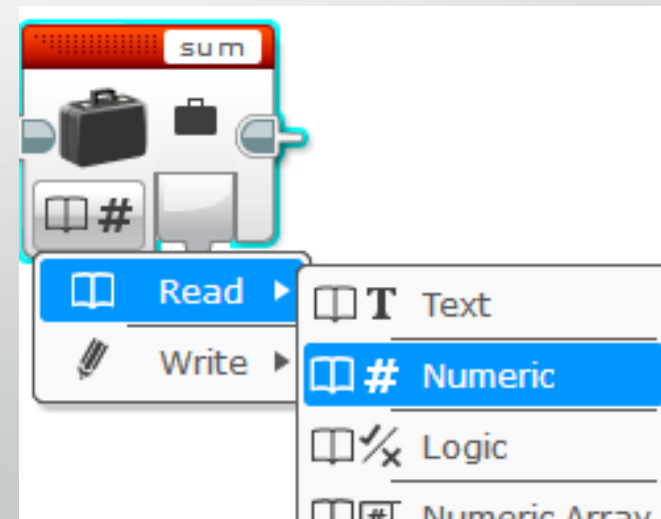
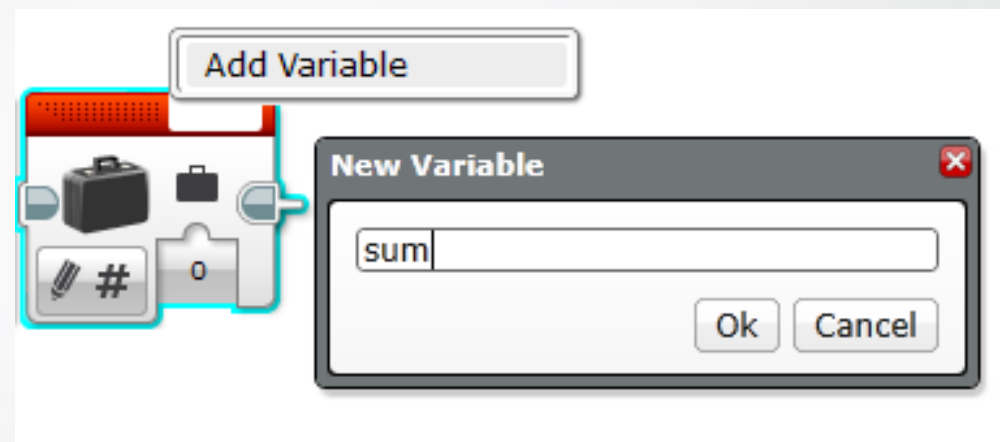
- (1) 超音波センサを用いて対象物との距離を測定し、それをリアルタイムで画面に表示する

変数の利用

- 変数とは...??
＝数字などのデータを記憶しておくことのできる箱のようなもの
- 特徴
 - ・ 名前を付けられる
 - ・ ひとつの変数はひとつのデータしか保持できない
 - ・ 何度でもデータを代入（データを上書きすること）できる
 - ・ 途中でデータの種類を変更することはできない

変数の利用

- まずは変数を用意してみる
「sum」という名前の数値型の
変数を用意、いまは“0”
- 変数「sum」から数字を引き出してみる
→読み込みモード数値型にして、
データワイヤーを利用して画面表示
→“0”が表示されればOK



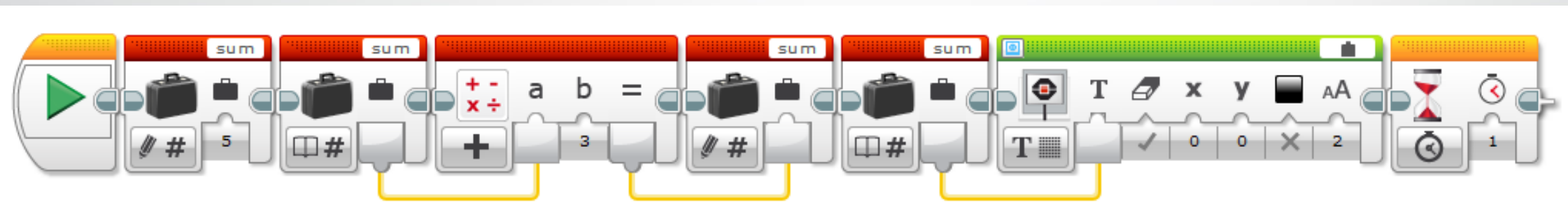
変数の利用

- 変数の値を更新する
 - ・ 先のプログラムの後ろに、「sumに5を代入して表示する」ようなプログラムを足してみよう
 - “0” 表示後、“5” が表示されればOK
 - ＝無事に変数の値が5に更新され、sumに0の値は残っていない



変数の利用

- 計算ブロックを利用して変数の値に手を加える



- 四則演算に限らず、複雑な計算も可能（拡張モード）
- Question：なぜ一番初めにsum = 5 のブロックを置いている??

→ ~~未知の値を用いて計算はできない（初期値の設定）~~

※LabVIEW上でEV3に書き込む場合、変数に値が格納されていない場合は数値型であれば”0”が予め代入されていることが分かりました。

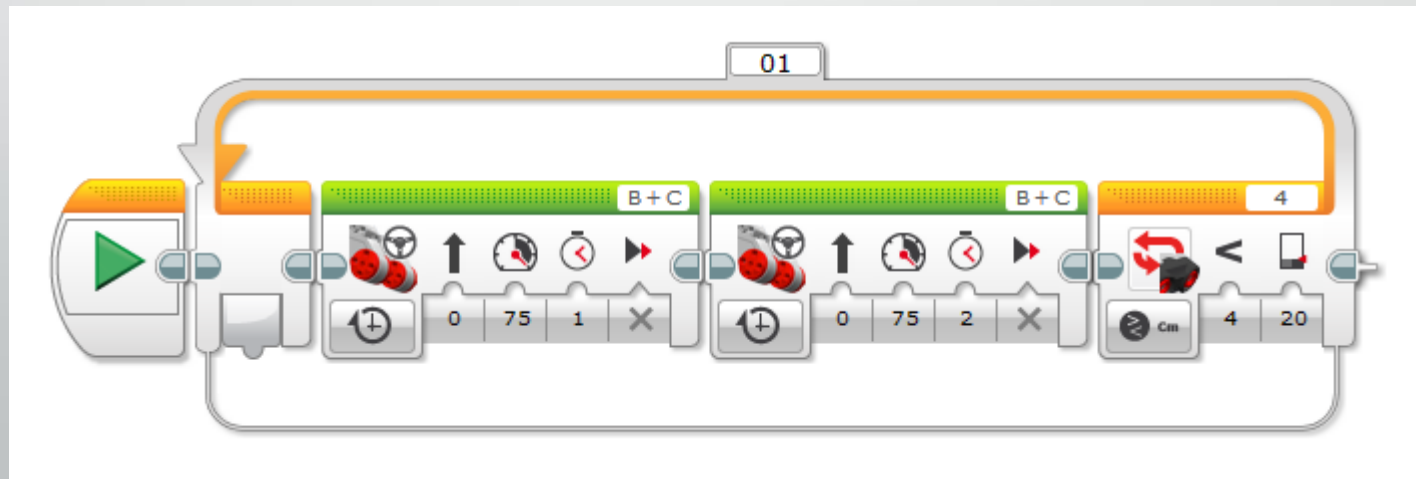
よって、ここでは問題ないこととなります。言語によってはエラーになるので注意

練習問題 3

- (1) 変数はプログラム内で複数扱うことができる（ただし同名であってはならない）。2つの数値型変数を用いて、それらの和差積商を4つ同時に表示する
- (2) タッチセンサを用いて、カウントアップするカウンターを作る。余力があれば、カウントダウン機能も追加する
- (3) 上記のカウンターを改造して、合計が3の倍数になるとバカになる「世界のナベアツロボット」を作る（3のつく数に反応する必要はなし）
- (4) 【練習問題2（1）】を改造して、表示する数値の単位系を[cm]と[inch]で入れ替えられるようにする

拘束時間

- プログラムは「前から」「順に」「ひとつずつ」実行される
- =前の動作が終わらないと次の動作は始まらない
- 下図は、「超音波センサで距離を測り、近くに物がくると止まる」
- プログラムの悪例。何が問題...??



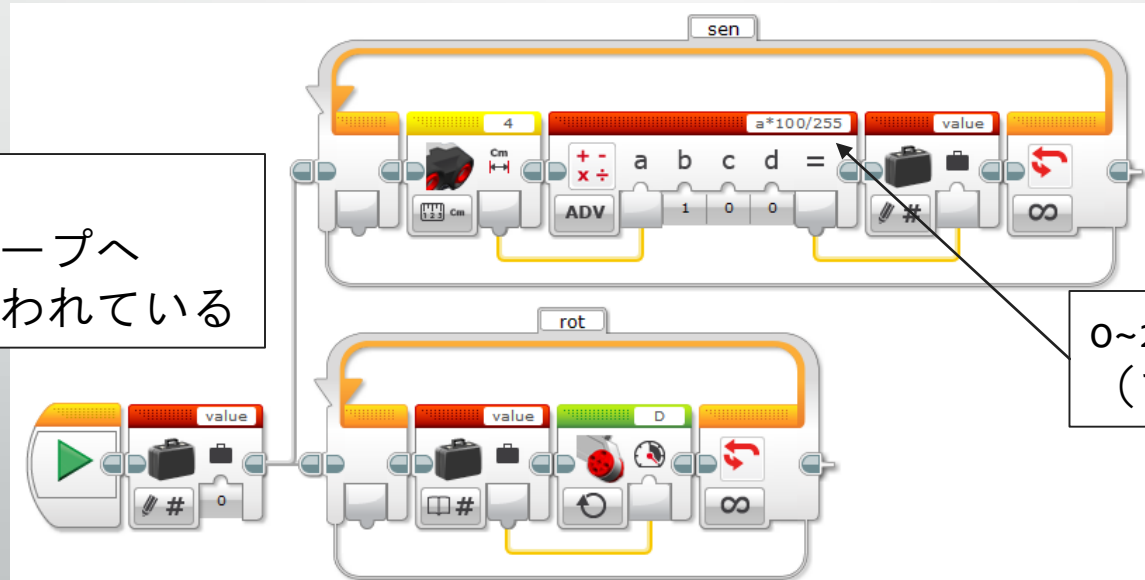
並列処理

- プログラムは必ずしも一本道である必要はない
→途中で分かれ道を作り、2つ以上の処理を並行させることができる
- 並列先へのデータ渡しは変数、並列先のループ中断はBreakが便利
- 便利な反面、闇が深いので注意
(ミスが起こりやすい、何が起きているか把握するのが大変など)

並列処理

- 簡単な並列の例：超音波センサで測定した距離に応じ回転速度が変化
上のループは測定用で、センシングしながらその値を変数valueに代入
下のループは動作用で、変数valueを参照してPowerに入れて駆動
- センシング→駆動のラグがとても短くなり、応答がよくなる

変数"value"を介して、
測定ループから駆動ループへ
測定値の受け渡しが行われている



0~255を0~100に変換している
(マッピングという)