

# Motion Works 取扱説明書

ヴイストーン株式会社

2020.07.21

## 目次

1.はじめに .....	3
1-1. 概要 .....	3
1-2. CPU ボードの仕様 .....	3
1-3. CPU ボードの部位説明 .....	4
1-4. microSD カードの準備 .....	5
2.導入方法 .....	7
2-1. インストールと起動 .....	7
2-2. ロボットプロジェクトの作成 .....	8
3.モーションの作成 .....	11
3-1. ポーズエリア .....	11
3-2. ポーズスライダ .....	12
3-3. サーボモータの位置補正 .....	13
3-4. ポーズの作成 .....	16
3-5. モーションの再生 .....	17
3-6. ポーズの遷移時間の調整 .....	18
4. ブロックの説明 .....	20
4-1. スタートブロック・エンドブロック .....	22
4-2. ポーズブロック .....	22
4-3. ウェイトブロック .....	23
4-4. 演算ブロック .....	23
4-5. 分岐ブロック .....	24
4-6. ゲームパッドブロック .....	25
4-7. サブルーチンブロック .....	25
4-8. 音声ブロック .....	26
4-9. キャンセルブロック .....	27
4-10. ファイル書き込みブロック .....	27
4-11. ファイル読み込みブロック .....	28
4-12. コメントブロック .....	28
5. スタンドアローンでの動作 .....	29
5-1. モーション・関連ファイルの転送 .....	29
5-2. 初期化モーションの例 .....	30
5-3. 操作マップの作成 .....	31

6. メモリマップ .....	34
7. ポーズに関する機能.....	35
7-1. ポーズスライダの設定 .....	35
7-2. 基準ポーズの設定 .....	37
7-3. ポーズの補間器について .....	39
7-4. ポーズ作成に便利な機能 .....	40
7-5. ポーズの一括操作 .....	41
8. モータに関する設定.....	42
8-1. モータ ID の設定 .....	42
8-2. 個別のモータの出力設定 .....	43
9. 拡張機器に関する設定.....	45
10. その他の設定・機能説明 .....	46
10-1. CPU の設定.....	46
10-2. 画面表示の設定.....	46
10-3. 平滑化の演算.....	48
10-4. シリアル通信.....	48
10-5. ファームウェアの更新.....	49
11. FAQ.....	50

# 1.はじめに

## 1-1. 概要

Motion Works for VS-RC026 は、ロボット用小型 CPU ボード「VS-RC026」を扱うためのソフトウェアです。VS-RC026 の各種設定（サーボモータやコントローラでの操作、拡張機器などの設定）は、すべて Motion Works から行います。また、VS-RC026 を搭載したロボットのモーション（動作）を作成する場合にも Motion Works を使用します。ロボットを正しくお使いいただくために、この「Motion Works 取扱説明書」をお読みいただき、使用方法をご理解ください。

Motion Works を実行するには、以下の条件を満たす PC の動作環境が必要です。

- OS : Windows7/8/8.1/10(32bit/64bit)
- インターフェース : USB ポート
- 画面サイズ : XGA(1024×768)以上

なお、以降の説明では、Motion Works を「本ソフトウェア」、現在お読みいただいている「Motion Works 取扱説明書」を「本説明書」、CPU ボード「VS-RC026」を「CPU ボード」とそれぞれ記述します。

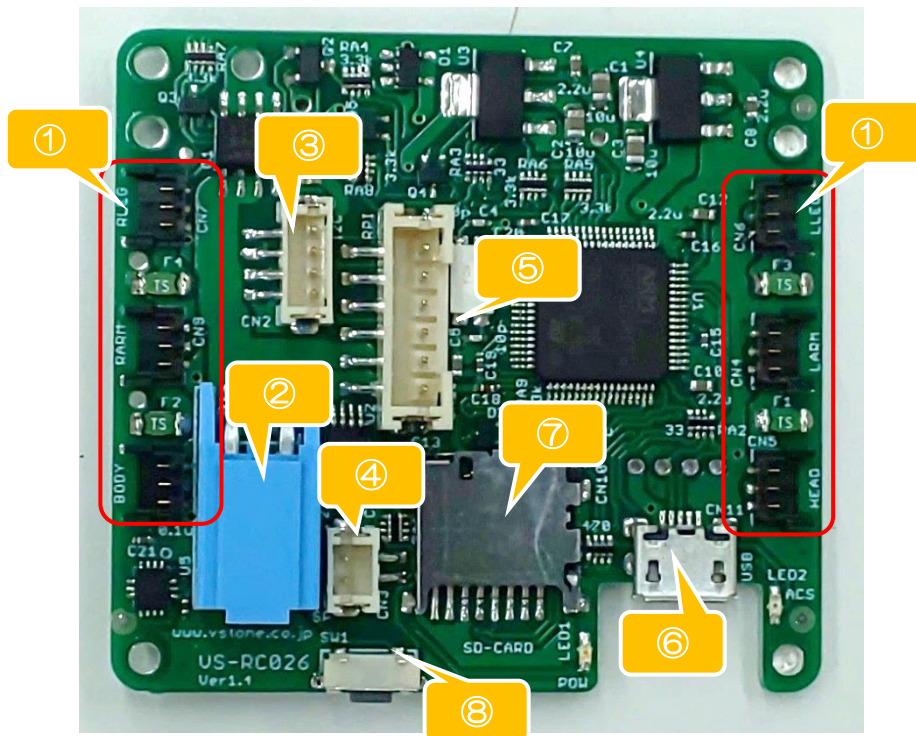
※Motion Works・モーションワークスは、ヴイストーン株式会社の登録商標です。

※その他、本説明書中に登場するすべての商品名、商標、登録商標については、すべての権利が各所有者に帰属するものとします。

## 1-2. CPU ボードの仕様

- 対応サーボモータ:シリアルサーボモータ(双葉電子社プロトコル)
- 動作電力:7~12v
- 搭載センサ: 6 軸センサ(3 軸加速度センサ/3 軸ジャイロセンサ)
- 対応拡張機器(2020 年 7 月現在)
  - VS-C3                      無線コントローラ

### 1-3. CPU ボードの部位説明



#### ① サーボモータコネクタ

サーボモータを接続する端子です。一部のポートは回路的に同じバスに接続されています(LARM と HEAD、RARM と BODY がそれぞれ同一バスです)。

#### ② 電源コネクタ

駆動用の電力を接続します。なお、CPU ボードのファームウェアは USB からの供給電源でも動作しますが、サーボモータ等一部の周辺機器は電源コネクタからの電力供給が無いと動作しません。

#### ③ 拡張機器コネクタ

拡張機器を接続します。

#### ④ 音声出力

モノラル音声を出力します。スピーカは別途ご用意ください。

#### ⑤ Raspberry Pi4 用拡張コネクタ

Raspberry Pi4 との接続に用いられます。

#### ⑥ USB コネクタ

PC との通信を行います。コネクタの形状は mirco-B です。

#### ⑦ microSD カードスロット

ロボットの設定・動作データなどを記録する microSD カードを装着します。FAT 形式でフォーマットされた SD カードをご利用ください。

#### ⑧ 電源スイッチ

CPU ボードの電源を ON/OFF します。電源が OFF の状態で長押し(約 1 秒以上)すると ON になり、ON の状態で押しすと OFF になります。

## 1-4. microSD カードの準備

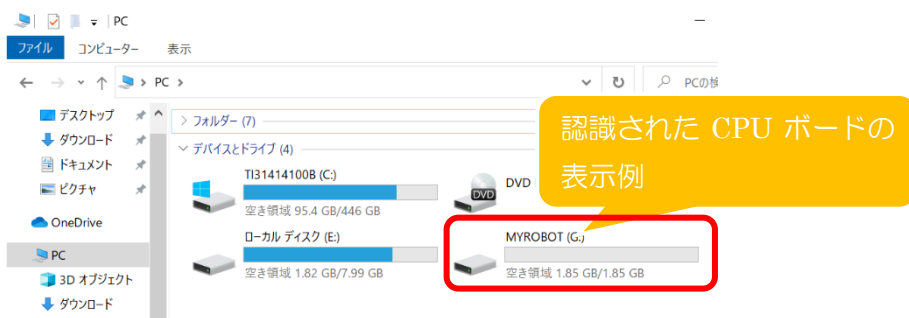
CPU ボードを使用する場合、FAT 形式でフォーマットした SD カードを CPU ボードの microSD カードスロットに装着します。既に製品に専用の SD カードが付属している場合はこの手順は必要ありませんが、SD カードが付属していない場合、もしくはバックアップ目的などで予備の SD カードを準備する場合は、下記の手順をご参照ください。

SD カードのフォーマットは PC から行います。尚、FAT 形式でフォーマットできるのは、**容量が 4GB までの SD カード**です。これに合う SD カードをご用意いただくか、容量の大きい SD カードに対して後述の方法でディスクイメージを書き込む等の方法で、別途 SD カードをご用意ください。

PC に SD カードを入れ、認識したドライブを右クリックして「フォーマット」をクリックしてください。フォーマットのウィンドウでは、「ファイルシステム」を「FAT」に指定してください。また、「ボリュームラベル」にロボット名などを入力しておくとも判別しやすくなります。「フォーマットオプション」で「クイックフォーマット」にチェックを入れても問題ありません。



SD カードを CPU ボードに装着し、USB(microB)ケーブルで PC と接続すると、PC が自動的に CPU ボードを認識します。CPU ボードは PC からは一般的な SD カード等のようにマスタストレージデバイス(ドライブ)として表示されます。実際に表示されるデータは、SD カードに書き込まれたデータと同一です。



CPU ボードが認識されない場合は、SD カードが正しくフォーマットされているか、また CPU ボードの SD カードスロットに正しく差し込まれているかご確認ください。

4GB より容量が多い SD カードしかご用意できない場合は、「Win32 Disk Imager」というツールを使ってディスクイメージを書き込んで下さい。

Win32 Disk Imager 配布 URL(ページ内の「Download」よりダウンロードしてください)。

<https://sourceforge.net/projects/win32diskimager/>

ディスクイメージは、本製品のサポートページ(下記 URL)にて配布しています。

[https://vstone.co.jp/products/robovie\\_z/index.html](https://vstone.co.jp/products/robovie_z/index.html)

ディスクイメージを書き込んだら、既に FAT フォーマットされた状態の SD カードが出来上がります。尚、イメージ書き込み後は、本来のディスク容量よりも利用できるサイズが小さくなりますので、ご注意ください。

## 2.導入方法

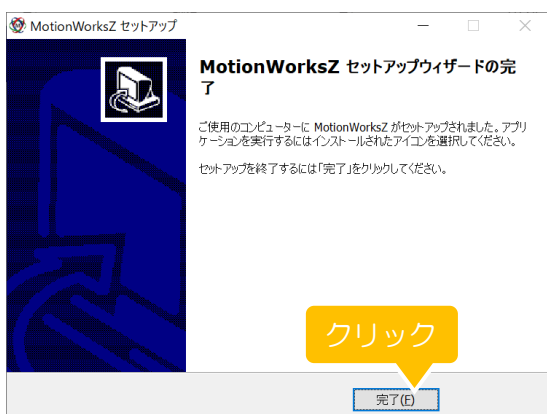
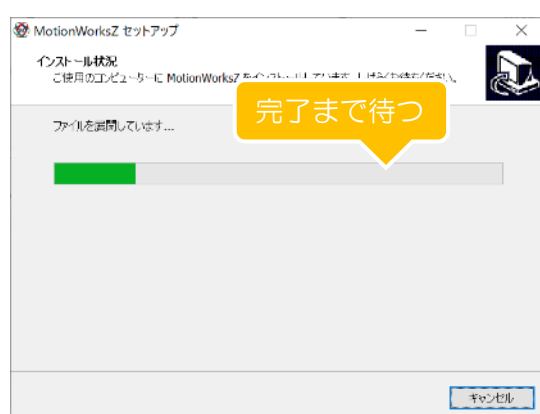
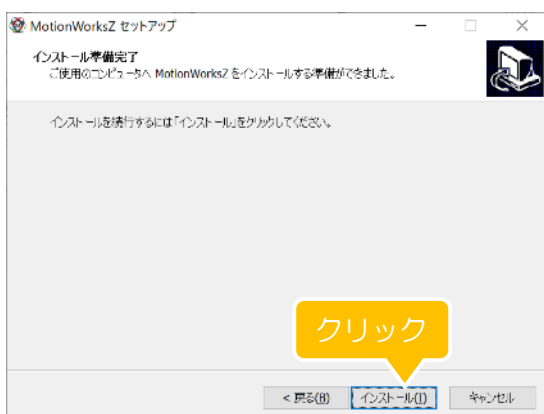
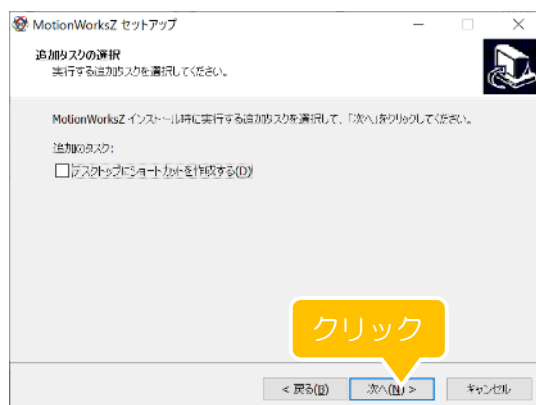
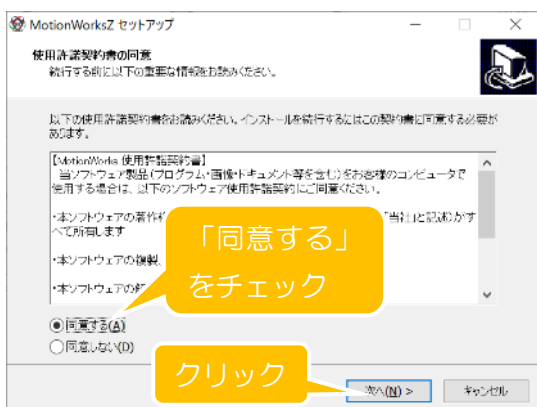
本ソフトウェアをインストールしてモーションを作成できる状態にするまでの手順を説明します。

### 2-1. インストールと起動

弊社の製品ページ(後述)より最新版のインストーラをダウンロードしてください。インストーラのファイル名は、「**MotionWorks\_for\_VSRC026\_Inst\_???.exe** (???には三桁の数字が入ります)」という形式です。ファイルを実行すると以下のようなダイアログが開くので、ダイアログの指示に従ってインストール作業を進めてください。



MotionWorks\_for\_VSRC026\_Inst\_???.exe  
を実行



本ソフトウェアを起動する場合、スタートメニューを開いて「Motion Works for VS-RC026」グループ内の「Motion Works」をクリックしてください。



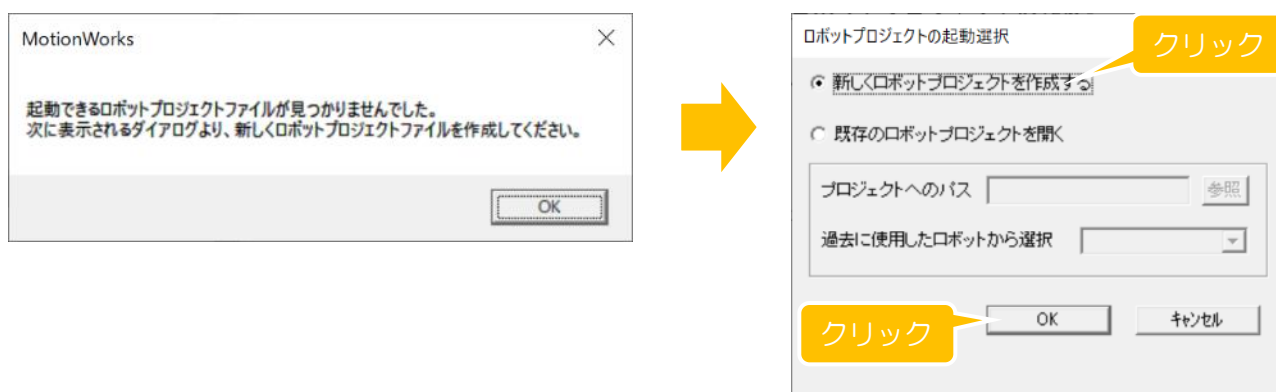
インストール時に「デスクトップにショートカットを作成する(D)」にチェックを入れた場合は、PCのデスクトップに「Motion Works」というショートカットができるので、それをダブルクリックしても起動します。また、スタートメニューの「Motion Works Uninstall」をクリックすると本ソフトウェアをPCからアンインストールします。

## 2-2. ロボットプロジェクトの作成

本ソフトウェアでは「ロボットプロジェクト」という概念が存在し、ロボットの個体や機種に応じてモーション作成や設定を切り分けることができます。最初にお使いのロボットに合わせて新しいロボットプロジェクトを作成します。

インストール直後等、起動するためのロボットプロジェクトが見つからなかった場合は、その旨のメッセージを表示するので、「OK」をクリックしてください。

続いて、ロボットプロジェクトの新規作成・読み込みを選択するダイアログを表示します。今回のように初めてロボットプロジェクトを作成する場合は「新しくロボットプロジェクトを作成する」にチェックを入れて「OK」をクリックしてください。



既にロボットプロジェクトが存在し、そちらを開く場合は、「既存のロボットプロジェクトを開く」にチェックを入れ、「参照」ボタンをクリックしてロボットプロジェクトファイル (\*.rpz) へのパスを指定してください。



続いて作成するプロジェクトの情報入力画面を開くので、下図の説明に従い情報を入力してください。全ての設定が完了したら「作成」をクリックしてください。

プロジェクトの新規作成

ロボットの名前  **ロボット名を入力  
(半角英数 16 文字以内)**

他のロボットとの識別を行なうために必要な設定です。  
16文字以内の半角英数字を入力してください  
また、「¥/ :\* ? < > |」の各文字は使えません

プロジェクトの作成場所  **プロジェクトを作成する  
フォルダを設定**

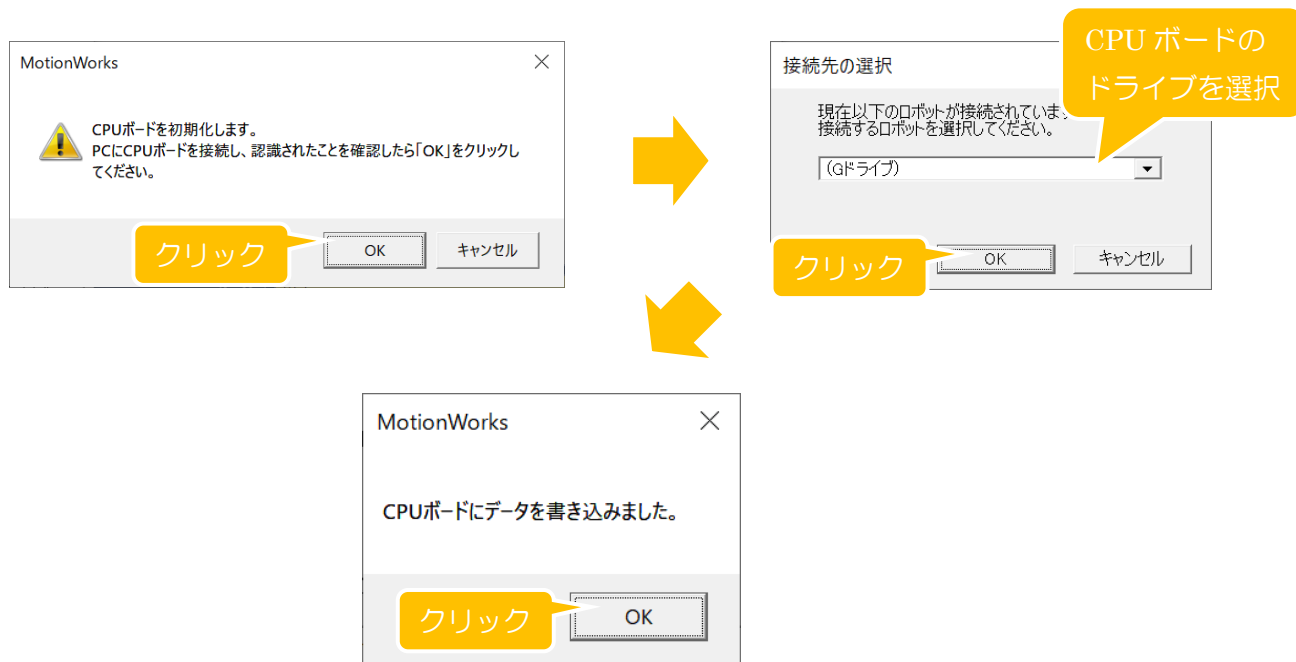
ロボットの種類 (オリジナルロボット)  **ロボットの機種を選択**

ご購入頂いたロボットの商品名をお選びください。

CPUボードを初期化する **チェックする**

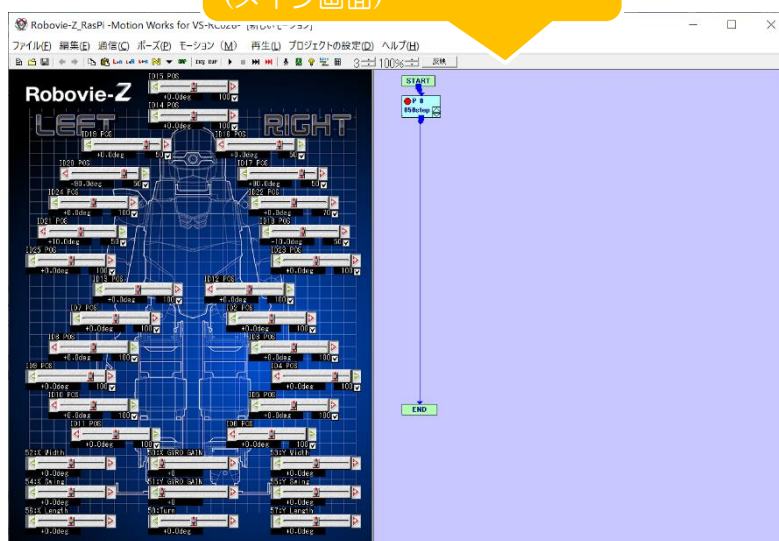
**クリック**

「作成」をクリックすると、指定したフォルダにプロジェクトのファイルを作成し、続いて CPU ボードを初期化します。メッセージの案内に従い、PC に CPU ボードを接続して認識されたことを確認してから、「OK」をクリックしてください。続いて、初期化先の CPU ボードのドライブ名を選択します。



以上でロボットプロジェクトの作成は完了です。全ての手順が完了すると、新しく作成したロボットプロジェクトを開いた状態で本ソフトウェアを起動し、下記のようなウィンドウ(メイン画面)を表示します。次回以降、本ソフトウェアを起動すると、先ほど作成したロボットプロジェクトを開いて本ソフトウェアを起動します。

本ソフトウェアの起動画面例  
(メイン画面)



また、ロボットプロジェクトの作成時に「プロジェクトの作成場所」に指定したフォルダには、「**‘ロボットの名前’ .rpz**」というファイルが作成されます。このファイルをダブルクリックすると、そのプロジェクトを読み込んで本ソフトウェアを起動することができます。



プロジェクトのフォルダにある「\*.rpz」ファイルを  
ダブルクリックしても起動できる

### 3. モーションの作成

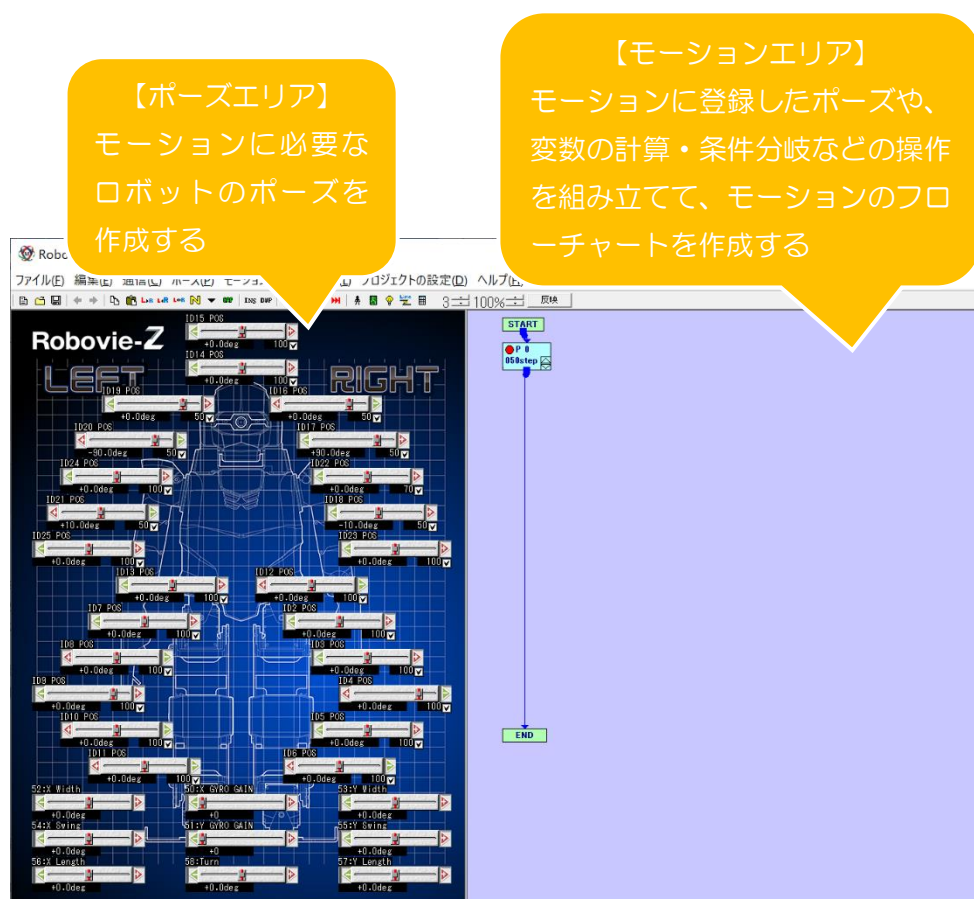
本ソフトウェアの主な機能の一つに、「歩行」「起き上がり」などのロボットの動作を作成する機能が挙げられます。これらのロボットの動作を「モーション」と呼びます。

本項目ではモーションの作成に必要な基礎知識について記述しています。始めてモーションを作成する際は、本項目を順にお読みいただき、モーション作成の流れをご確認ください。

#### 3-1. ポーズエリア

モーションの作成は、メイン画面上の「ポーズエリア」と「モーションエリア」で行います（下画像参照）。モーションの構造はアニメーションと同じで、モーションにおける、ある時点でのロボットの姿勢（ポーズ）を複数作成し、一定の時間間隔でそれらのポーズを順番に実行させることで、ひとつのモーションを成します。

本ソフトウェアでモーションを作成する場合、モーションに使用するポーズをポーズエリアで作成し、それをモーションエリアに登録します。モーションエリアでは、登録したポーズをロボットに実行させる順序や、ポーズごとの遷移時間（動作の速度）を設定します。また、モーションエリアには、ポーズエリアで作成したポーズ以外にも、条件分岐の設定や変数の計算などを登録することが可能です。



### 3-2. ポーズスライダ

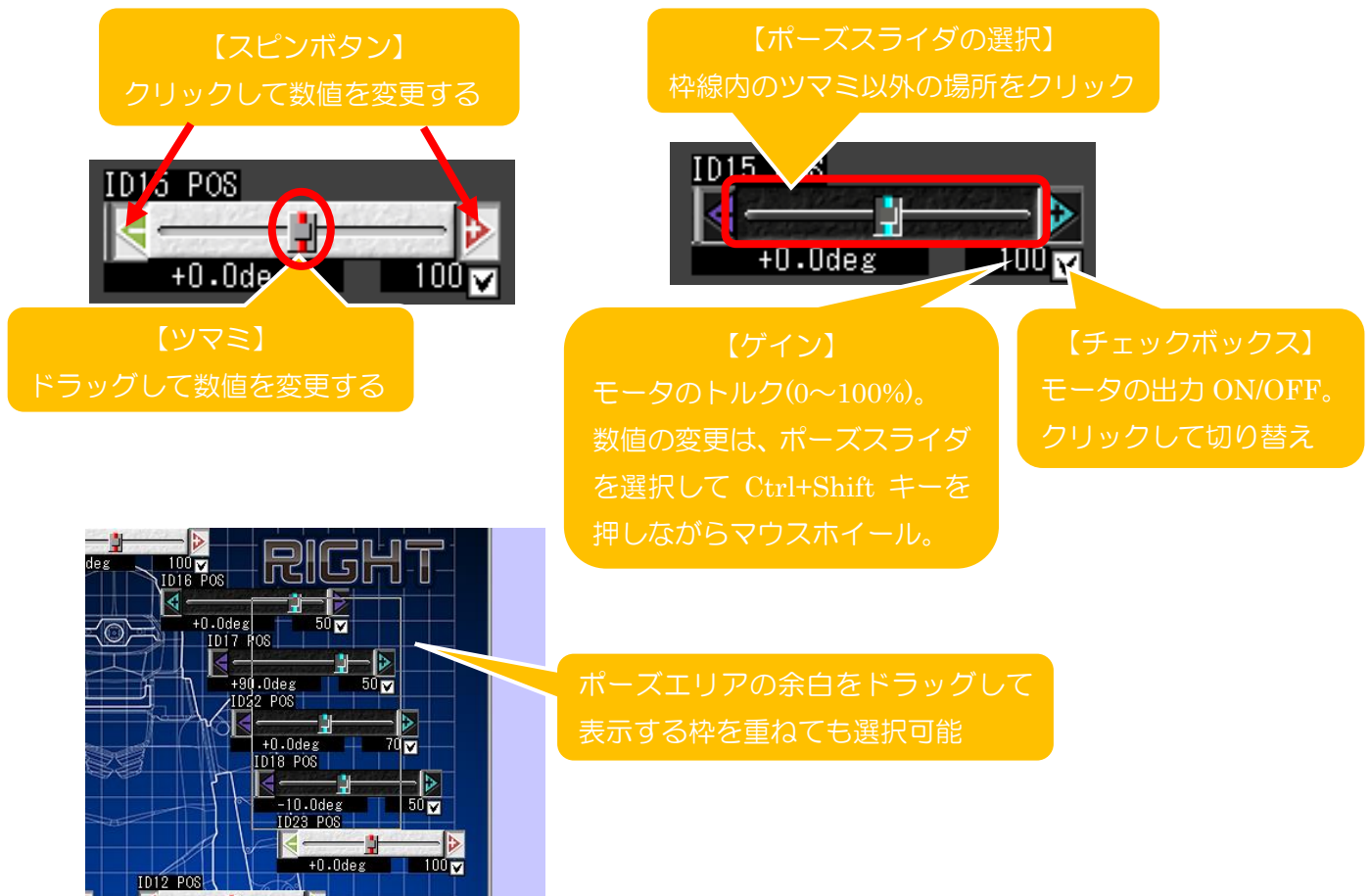
ポーズエリアには、サーボモータの角度やゲイン(保持力)を設定する「ポーズスライダ」(下図参照)がモータごとに表示されています。ポーズスライダの配置や名称・設定範囲などはロボットプロジェクト作成時に適時設定されています。また、サーボモータ以外にセンサのゲインや LED の明るさと言った数値もポーズスライダに割り当てて、モーションに組み込むことが可能です。



ポーズスライダの数値を操作する場合、つまみをドラッグ、またはスピノボタンをクリックします。数値を大きく動かす場合は前者、小刻み・正確に動かす場合は後者で、それぞれ操作してください。

ポーズスライダの特定箇所をクリックしたり、ポーズエリア内の余白部分をドラッグして表示する枠線に重なると、ポーズスライダを選択状態にします。ポーズスライダを選択してマウスホイールを操作すると、複数のポーズスライダをまとめて操作できます。

また、モータに割り当てられたポーズスライダには、「ゲイン」と「チェックボックス」が含まれます。ゲインを操作する場合は、ポーズスライダを選択し、キーボードの Ctrl+Shift キーを押しながらマウスホイールを操作します。チェックボックスはマウスでクリックして操作します。



- 【スピノボタン】**  
クリックして数値を変更する
- 【つまみ】**  
ドラッグして数値を変更する
- 【ポーズスライダの選択】**  
枠線内のつまみ以外の場所をクリック
- 【ゲイン】**  
モータのトルク(0~100%)。数値の変更は、ポーズスライダを選択して Ctrl+Shift キーを押しながらマウスホイール。
- 【チェックボックス】**  
モータの出力 ON/OFF。クリックして切り替え
- ポーズエリアの余白をドラッグして表示する枠を重ねても選択可能**

### 3-3. サーボモータの位置補正

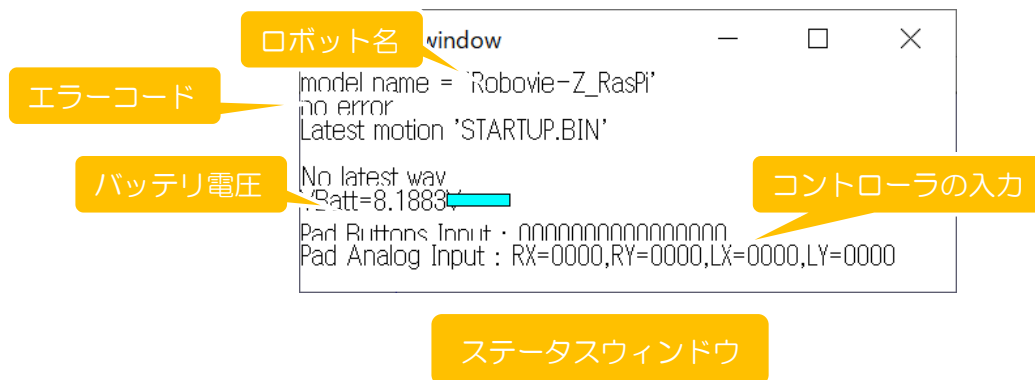
ロボットのポーズやモーションを作成する前に、サーボモータに存在する角度誤差を補正します。まず、すべてのポーズスライダを調整の基準となるポーズ(以後「基準ポーズ」と表記)に合わせます。ポーズスライダを二つも選択しない状態で、ツールバーの **N** ボタンをクリックしてください。



続いて、CPU ボードと通信を開始します。CPU ボードと PC を USB ケーブルで接続し、PC が CPU ボードを認識したことを確認してから、ツールバーの **通信** ボタンをクリックしてください。通信を開始すると、ツールバーの **通信** ボタンがへこんだ状態になります。通信を終了する場合は、もう一度 **通信** ボタンをクリックします。



また、プロジェクトの設定によっては、通信状態を示す「ステータスウィンドウ」を表示します。ステータスウィンドウでは、バッテリーの電圧・コントローラの入力状態など CPU ボードから読み込んだ各種情報が表示されます。



サーボモータの位置補正は、サーボモータに電源を入れて、実際に関節を動かしながら行います。このとき「モータロック」が発生しそれが長時間続くとモータが破損する恐れがありますので、あらかじめ下記の注意事項をご確認の上、適時モータの状態を確認して作業を行ってください。

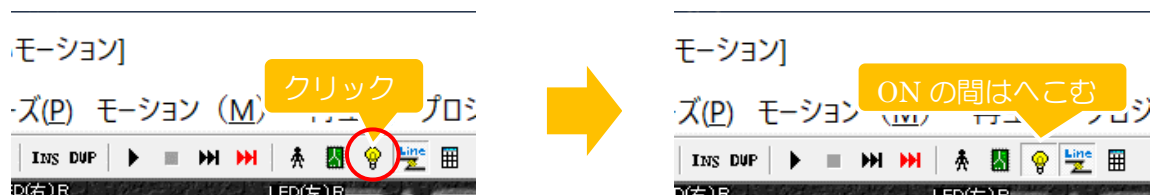
### モータロックに関する注意事項

サーボモータが本来指定された角度とは違う位置に無理やり合わせられて、大きな負担がかかっている状態を「モータロック」といいます。モータがロックすると「普段より振動や駆動音が大きくなる」「熱を帯びてくる」などの現象が見られ、これが長時間続くと、モータが発熱して故障したり、場合によっては発煙・発火などの危険も招きます。

モータロックが見られたらすぐに電源を切り、サーボモータが熱くなっている場合は冷めるまで休ませてください。

サーボモータは、CPU ボードの電源コネクタから供給される電力で動作します。CPU ボードにバッテリーを接続してください。続いて、モータ ON の際にロボットが転倒しないようにロボットを手で持ち上げる・支えるなどしてください。既定の機種種のロボットの場合、指の挟みこみなどの事故が発生しない正しい持ち方が紹介されています。

ここまで準備できたら、ツールバーの💡ボタンをクリックすると、モータが緩やかに ON になります。もう一度💡ボタンをクリックするとモータが OFF になります。また、CPU ボードからバッテリーを取り外しても OFF になります。

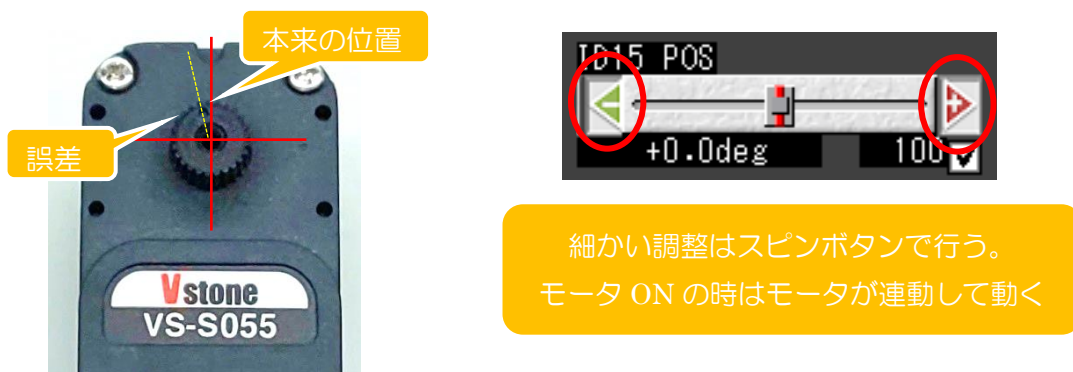


モータを ON にしている間は、モータロック・ロボットの転倒・指の挟みこみ等の事故に十分注意し、問題が発生したらすぐにモータを OFF にできるように準備してください。また、モータが ON の間はバッテリーを大きく消耗するので、不要な場面ではこまめに OFF にすることを推奨します。



モータを ON にしたら、各モータにおける基準ポーズとの誤差を目で確認し、ポーズスライダから正しい位置に補正してください。補正の場合は細かくモータを動かすため、ポーズスライダのスピンのボタンをクリックして少しずつ調整してください。

多くのモータは出力軸の原点( $\pm 0^\circ$  の位置)が基準ポーズになりますが、ロボットによっては特定の基準点などを参考に合わせる場合があります。お使いの機種種の調整方法を参照し、すべての軸を補正してください。



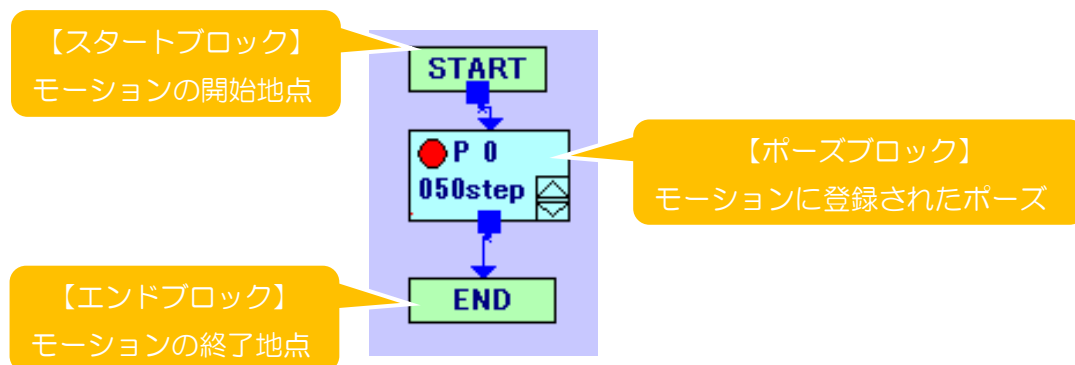
補正したら、ツールバーの歩ボタンをクリックすると、現在の位置から誤差を補正します。補正後は、各ポーズスライダの値が基準ポーズに戻ります。



サーボモータの誤差は個体によって異なるため、故障などでモータを交換した場合は、交換したモータにのみ、再度同じ手順で位置補正を行ってください。また、誤差が極端に大きい場合はモータの個体差ではなくサーボホーンの取り付け(歯車のかみ合わせ)が間違っている場合がありますので、そこに問題が無いかご確認ください。

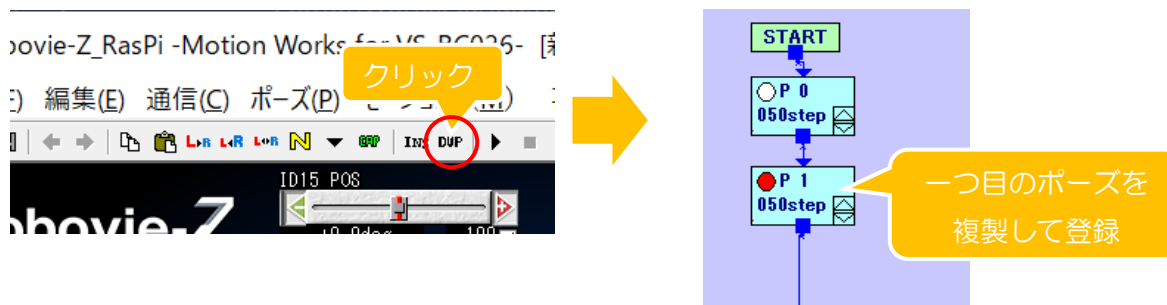
### 3-4. ポーズの作成

プロジェクトの起動直後等、新しいモーションを作り始めた際は、モーションエリアに「ポーズブロック」として最初のポーズがひとつ登録され、ポーズエリアでそのポーズが編集できるようになっています。この時のポーズは基準ポーズになっています。

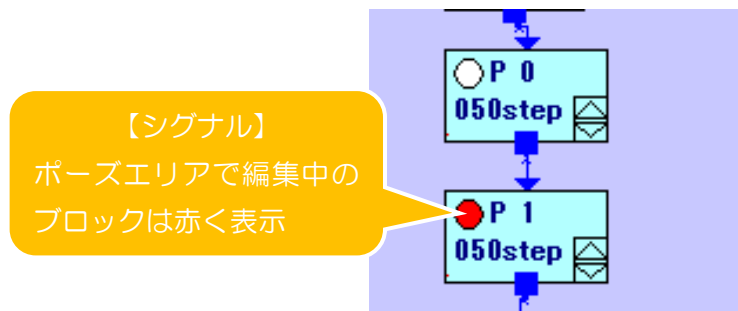


簡単なモーションの作成を例に、モーション作成の流れについて説明していきます。まず、モーションは「基準ポーズで始まり基準ポーズで終わる」という原則で作成します。このようにモーションを作ることによって、個別のモーション間のポーズが統一され、どのモーションをどの順番で実行しても安全に開始・終了できる利点があります。

上記の原則に従い、最初に存在する基準ポーズは「始まりの基準ポーズ」とするため、2番目のポーズを追加します。ポーズの追加にはいくつかの方法がありますが、連続したポーズを作る場合は、ツールバーのDUPボタンをクリックして「現在編集しているポーズを次のポーズとしてコピーする(複製)」という手順を推奨します。



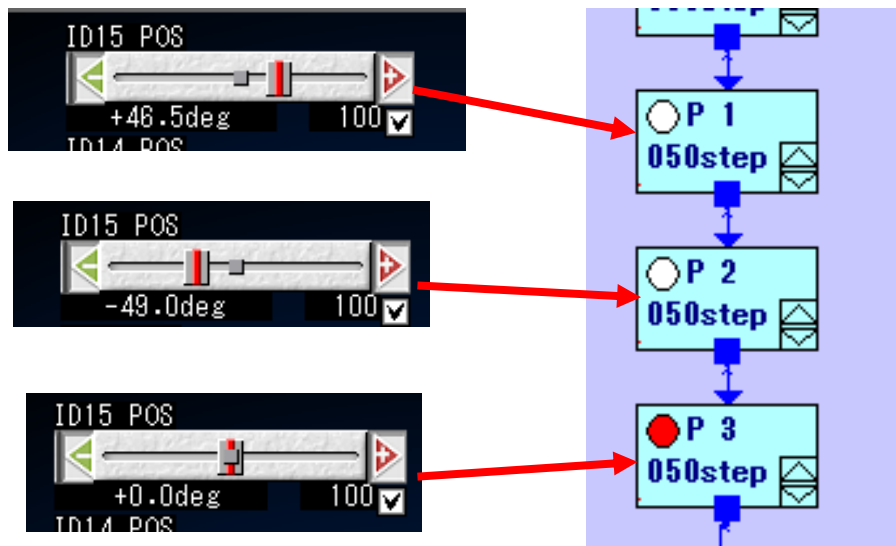
また、ポーズブロック左上の丸印は「シグナル」と言い、ポーズエリアで編集中のポーズは赤色で表示されます。編集するポーズを変更する場合、そのポーズブロックをクリックして、シグナルを赤にしてください。





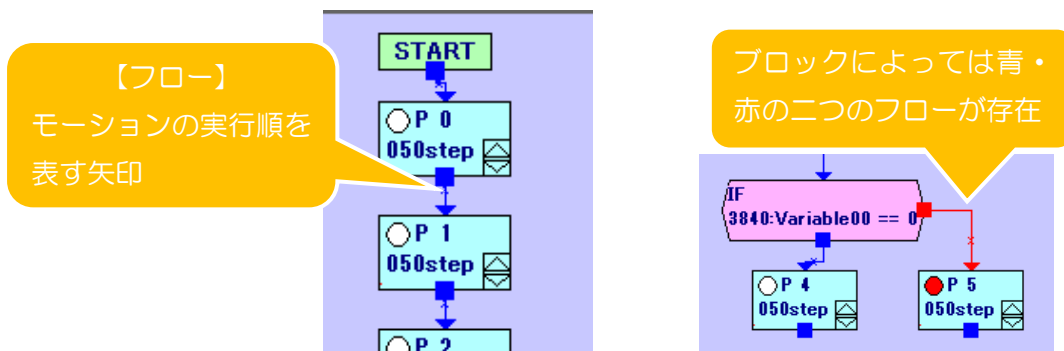
それでは、ポーズスライダを操作してロボットのポーズを自由に作成してみてください。初めてモーションを作成する場合は、「首を振る」「腕を上げる」など、腕や頭などあまり体のバランスに影響を及ぼさない関節だけを動かして体全体の動きが少ない簡単なモーションをお勧めします。ポーズを作成したらツールバーのDUPボタンで複製し、次のポーズを作ります。最後のポーズは基準ポーズに戻してください(ツールバーのNボタンをクリック)。

下図のモーションは、ID12のモータを少し動かすモーションのサンプルです。



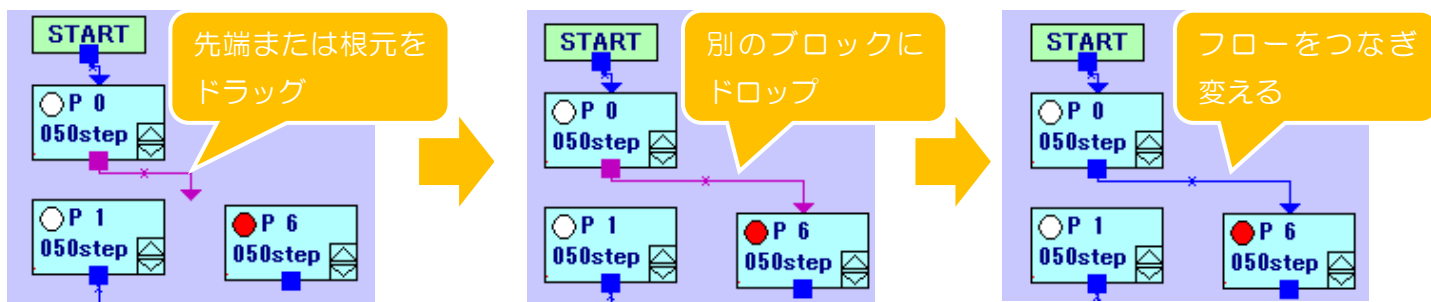
### 3-5. モーションの再生

モーションを作成したら、ツールバーの▶ボタンで再生できます。クリックするとスタートブロックから、各ブロックにつながっている青い矢印の順番通りにポーズが実行されます。この矢印を「フロー」と呼び、モーションはフローの接続に従って実行されます。一部のブロックには赤い矢印のフローもあり、条件によって進む先が分かります。



モーションの再生は、エンドブロックに到達するか、どこにもフローが繋がっていないブロックを実行すると終了します。また、モーションの再生を中断する場合はツールバーの■ボタンをクリックします。

フローはマウスドラッグでつなぎ変えることができます。矢印の先端及び根元をマウスでドラッグすると、色が紫に変わり矢印を伸ばせます。その状態で別のブロックにドロップするとフローをつなぎ変えられます。どこにもつなげずにドラッグを終えるとフローは切り離されます。

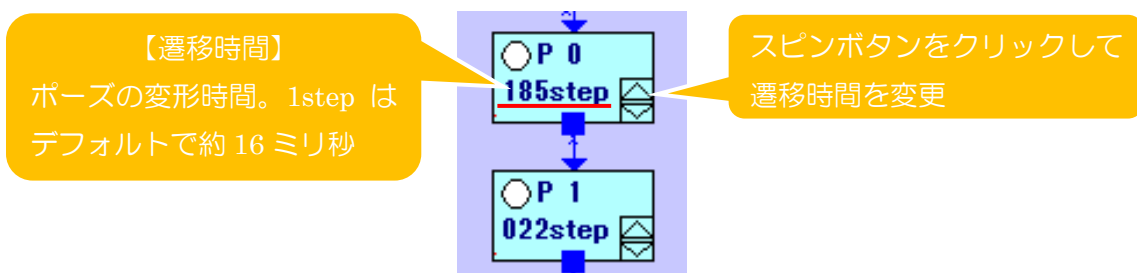


また、Alt キーを押しているとき、ブロックのどの場所をドラッグしてもフローをドラッグした状態になります。同様に、Shift+Alt キーを押しているとき、赤いフローをドラッグした状態になります。

モーションを再生すると、ロボットがモーションを実行している最中に画面のモーション再生は既に終了することがあります。また、そもそも 3~4 ポーズ程度のモーションは画面上でのモーション再生が一瞬で終わることもあります。これは、CPU ボードが数ポーズ分先行して実行するポーズを予約に積み上げる仕様のためです。モーション再生中に赤のシグナルが順番に移っていくと思いますが、これは CPU ボードに送信済みのポーズを表します。ポーズブロックの設定によっては、1 ポーズごとに実行し終えてから次のポーズに進むこともできます。詳細は「[4-2. ポーズブロック](#)」をご参照ください。

### 3-6. ポーズの遷移時間の調整

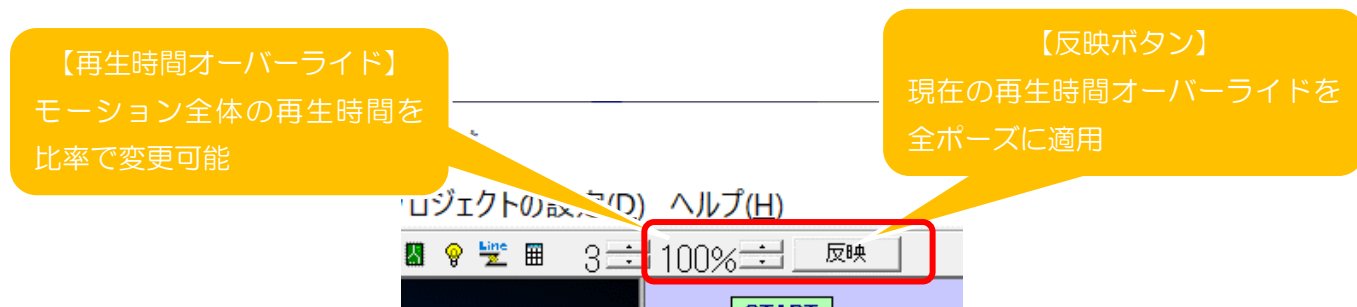
ポーズには「遷移時間(変形に要する時間)」の要素がそれぞれ存在し、ポーズごとに遷移時間を調整することで、モーションに緩急をつけられます。遷移時間はポーズブロックに表示されており、設定の変更はポーズブロックのスピンボタンで行います。



ポーズブロックはポーズスライダーと同様にマウス操作で選択でき、ポーズブロックを選択してマウスホイールを操作すると、複数のポーズの遷移時間をまとめて変更できます。



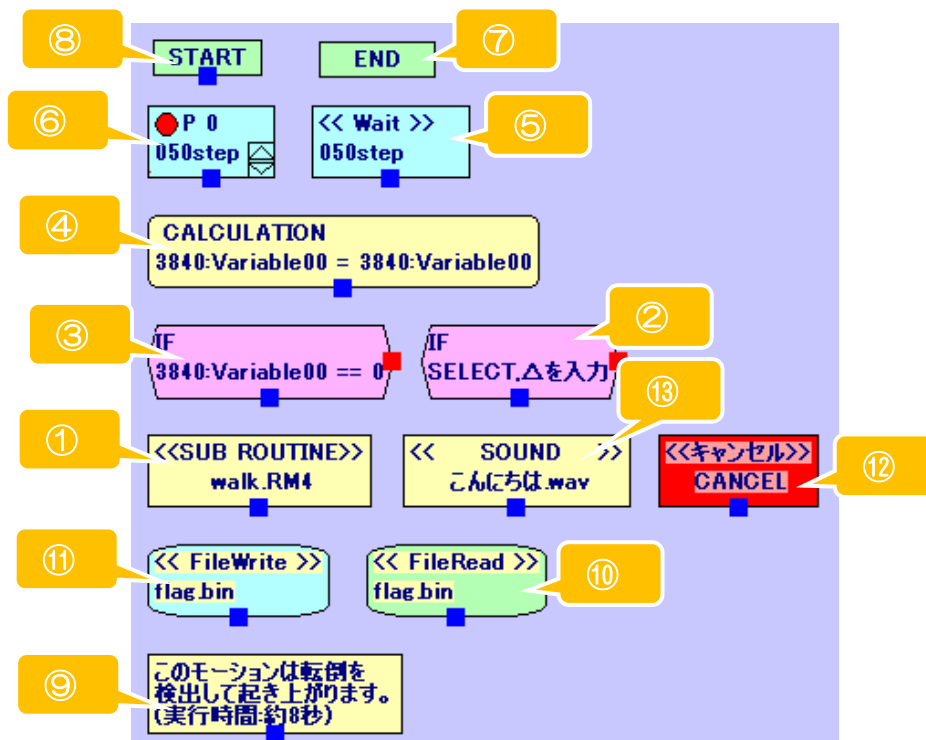
モーション全体の遷移時間を一定の割合で一括変更したい場合は、画面左上の「再生時間オーバーライド」を使用します。こちらの数値を 200%にすると本来の 2 倍の時間、50%にすると本来の半分の時間でモーションを再生します。再生時間オーバーライドは本ソフトウェアからテスト再生する場合のみ有効ですが、隣の「反映」ボタンをクリックすると、現在の設定を全ポーズの遷移時間に適用します。



モーション作成の基本的な流れや作業に必要な最低限の説明は以上で完了です。以降の項目では、ポーズブロック以外のブロックの種類やモーションの作成に便利な各種機能について説明します。

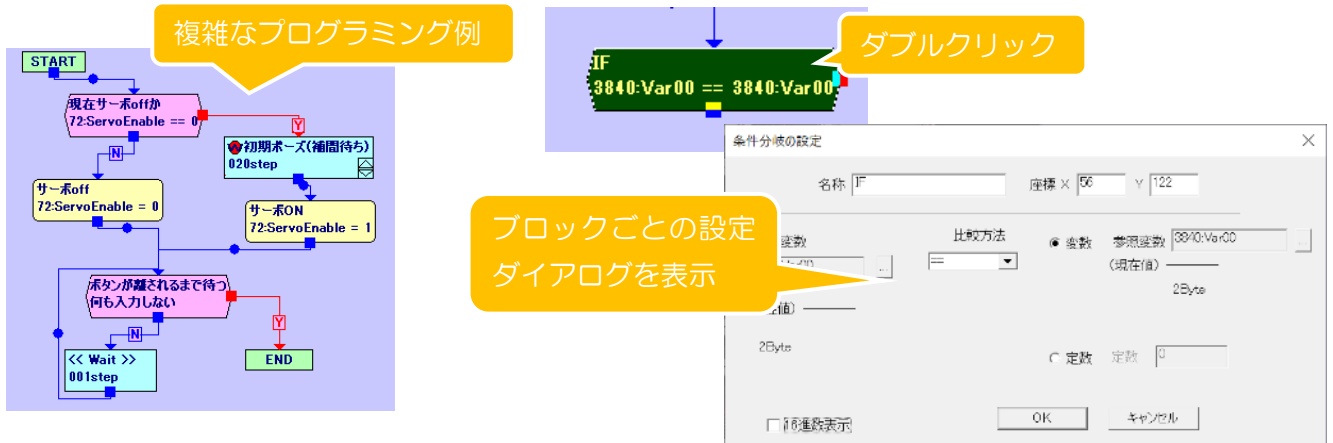
## 4. ブロックの説明

モーションエリアには、これまで説明したポーズブロック・スタートブロック・エンドブロックをはじめ、様々な機能を持つ「ブロック」が存在します。使用可能なブロックの名称と概要は下記の通りです。



- ① [スタートブロック](#)…モーションの開始地点
- ② [エンドブロック](#)…モーションの末端
- ③ [ポーズブロック](#)…ポーズを実行する
- ④ [ウェイトブロック](#)…指定時間待つ
- ⑤ [演算ブロック](#)…演算を行いメモリマップに結果を代入する
- ⑥ [分岐ブロック](#)…指定の条件でフローを分岐する
- ⑦ [ゲームパッドブロック](#)…ゲームパッドの特定のボタンの入力状態でフローを分岐する
- ⑧ [サブルーチンブロック](#)…別ファイルのモーションを呼び出す
- ⑨ [音声ブロック](#)…wave 形式の音声を再生する
- ⑩ [キャンセルブロック](#)…ゲームパッドの特定ボタンの入力状態でモーション再生を中断する
- ⑪ [ファイル書き込みブロック](#)…メモリマップの指定の領域を SD カード内のファイルに書き出す
- ⑫ [ファイル読み込みブロック](#)…SD カード内のファイルを読み込み、メモリマップの指定の領域に展開する
- ⑬ [コメントブロック](#)…特に機能は無いが、任意のコメントをモーションエリアに表示できる

これらのブロックをフローチャートの要領で組み合わせると、繰り返しや分岐・その他かなり複雑なプログラミングも可能になります。また、各ブロックは一部を除きダブルクリックすると詳細な設定ダイアログを開きます。



設定ダイアログ上部の「名称」「座標」は全ブロックで共通の設定項目です。「名称」はブロックの1行目に表示するコメント(30文字以内)、「座標」はブロックの表示位置をそれぞれ入力します。

ブロックの追加は、モーションエリアの余白部分を右クリックして、表示されたメニューより「ブロックの追加」をクリックし、一覧表示された種類から選びます。また、モーション内に存在するブロックを選択し、コピー&ペーストすることも可能です。同じく、ブロックを選択して Delete キーを押すと、不要なブロックを削除できます。



## 4-1. スタートブロック・エンドブロック

スタートブロックはモーションの開始地点を表し、モーションは必ずこのブロックから開始します。モーションに必ず1個だけ存在し、コピーや削除はできません。エンドブロックはモーションの末端を表し、モーションの末端を表します。エンドブロックが繋がっていない場合でも、どこにもフローが繋がっていないブロックを実行したらモーションは終了します。

これらのブロックは、詳細設定ダイアログが存在しません。

## 4-2. ポーズブロック

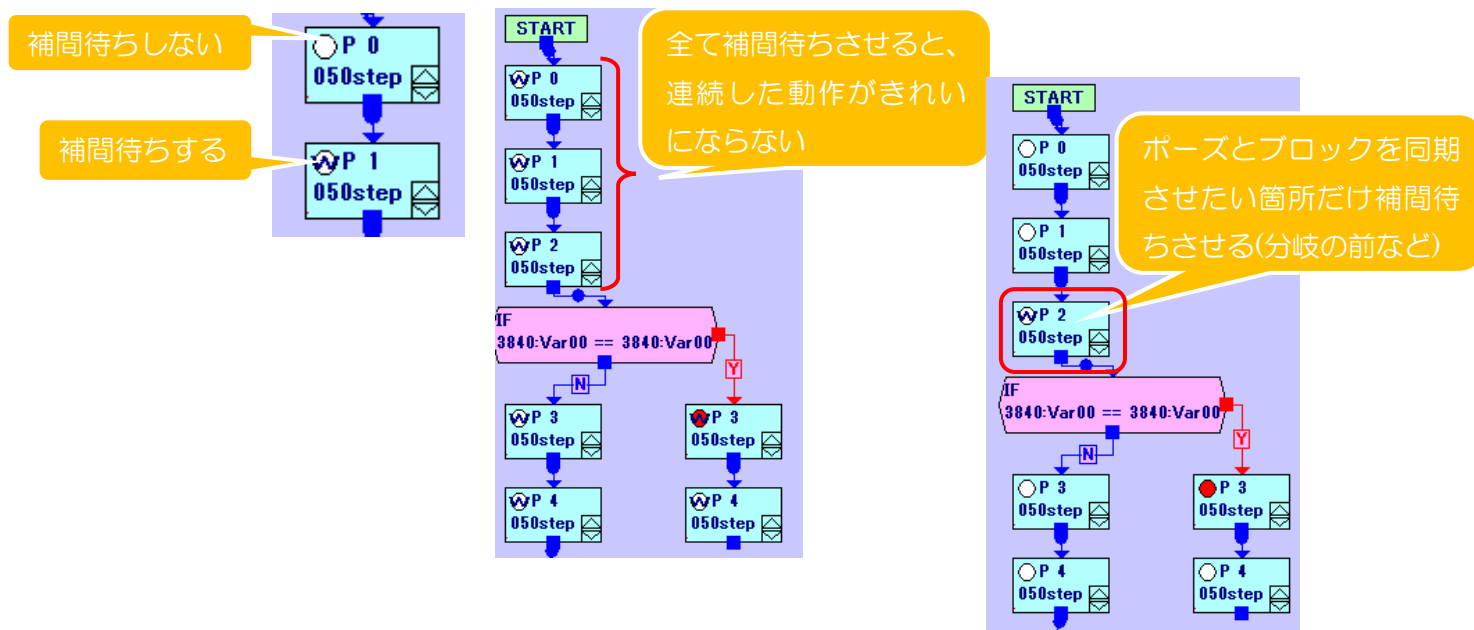
ロボットのポーズを設定します。ポーズの作成方法は「[3.モーションの作成](#)」をご参照ください。ブロックの詳細設定ダイアログでは、遷移時間とモーションの補間を同期するか否かの項目があります。



「[3-5. モーションの再生](#)」でも記述していますが、CPU ボードはポーズ実行の際に1ポーズずつ処理するのではなく、「あらかじめ数ポーズ先まで実行するポーズを積み上げて、順次実行していく」という処理を行います。これはポーズを滑らかに補間するための仕組みですが、ポーズの実行とモーション内のブロックの実行が同期せず、プログラムによっては不都合な場合があります。このような場合、「補間待ちを行う」にチェックを入れることで、予約されている全てのポーズを実行し終えるまで待つことができ、ポーズとブロックの進行を同期させることができます。

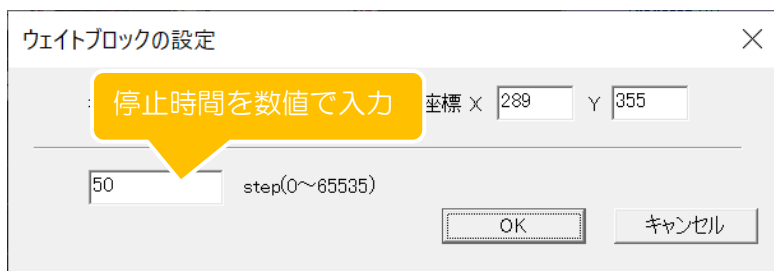
ポーズを補間待ちさせると、ポーズとブロックの進行が統一されわかりやすくなる反面、連続した動作の場合でもポーズごとに区切られたような動作になります。そのため、ポーズとブロックの進行のタイミングを一致させたいタイミングのみ、補間待ちを行うようにすることを推奨します。

補間待ちの有効・無効の状態は、ポーズブロックのシグナルのデザインで区別できます。



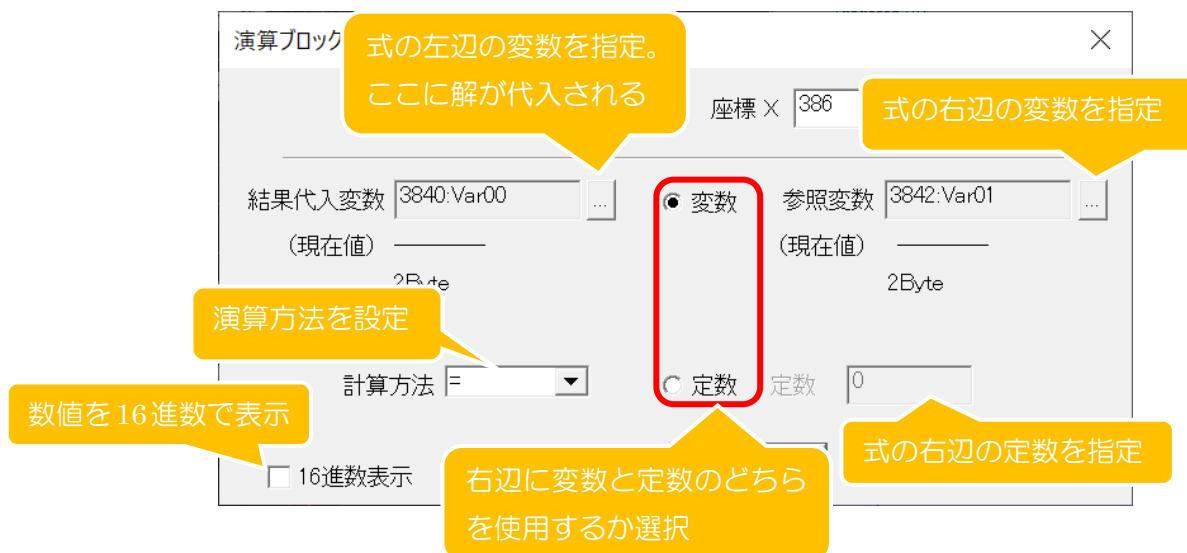
### 4-3. ウェイトブロック

指定の時間だけモーションの進行を停止します。CPU ボードが予約しているポーズの実行は停止しません。時間の単位はポーズブロックと同じです(デフォルトで 1step≒16 ミリ秒)。ブロックの詳細設定ダイアログでは、停止時間を設定します。



### 4-4. 演算ブロック

演算ブロックは「A=3」「B+=C」のようなプログラミングの代入式と同じ考えで、CPU ボードのメモリマップの任意の変数に対して演算を行います。メモリマップについては「[6. メモリマップ](#)」をご参照ください。設定ダイアログでは、左辺(変数)と右辺(変数または定数)の項、及び演算子を指定します。

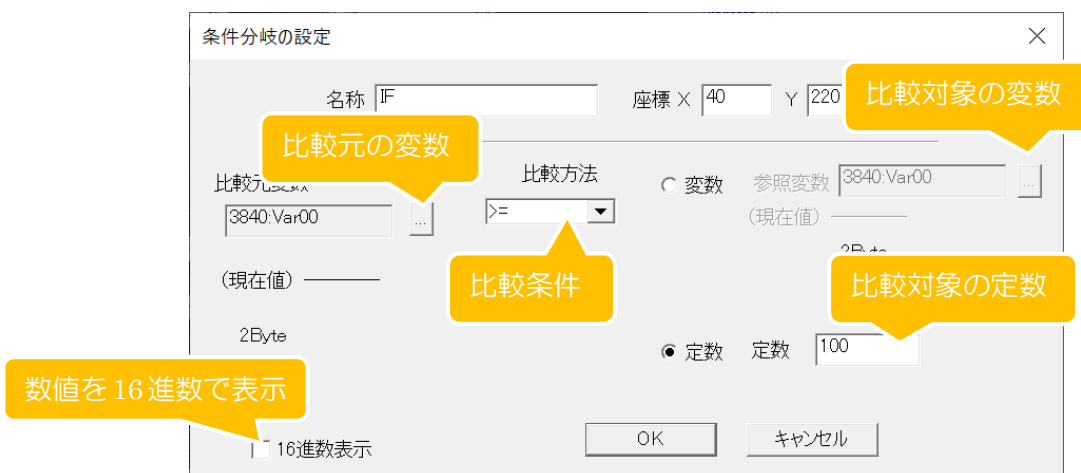


演算方法は、代入(=)、四則演算と剰余(+, -, \*, /, MOD=)、論理演算(AND=, OR=, XOR=, NOT=)を使用できます。また、変数は 1byte, 2byte, 4byte の 3 種類のサイズがありますが、解が変数のサイズをオーバーフローした場合は、あふれた桁が切り捨てられます。

「数値を 16 進数で表示」にチェックを入れて設定ダイアログを開き直すと、変数のアドレス値・変数の現在値・定数等が 16 進数で表示されます。また、定数に「0x」をつけて入力すると 16 進数の数値として換算します(例:「0x0a」と入力すると 10 進数の 10 に換算)。

## 4-5. 分岐ブロック

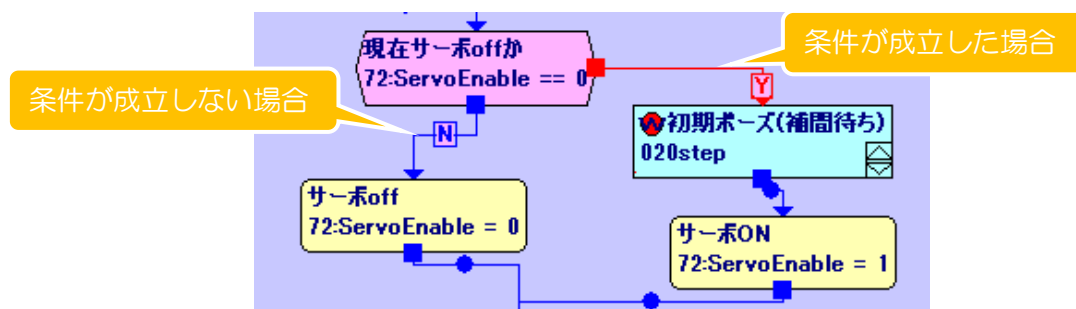
CPU ボードのメモリマップの任意の変数を参照し、指定した条件が成立するかでフローを分岐します。メモリマップについては「[6. メモリマップ](#)」をご参照ください。設定ダイアログでは、比較元の変数、比較対象の変数または定数、比較条件を設定します。



比較条件には、一致・不一致(==,!/=)、以上・超過(>=,>)、以下・未満(<=,<)、剰余の有無(MOD)、論理比較(AND,XOR)、常に一定(TRUE,FALSE)から選択できます。

「数値を16進数で表示」にチェックを入れて設定ダイアログを開き直すと、変数のアドレス値・変数の現在値・定数等が16進数で表示されます。また、定数に「0x」をつけて入力すると16進数の数値として換算します(例:「0x0a」と入力すると10進数の10に換算)。

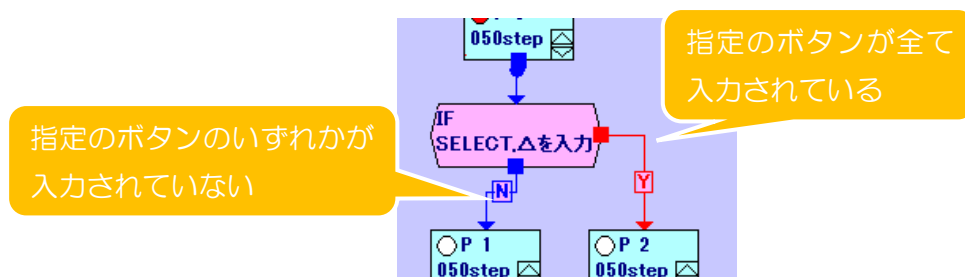
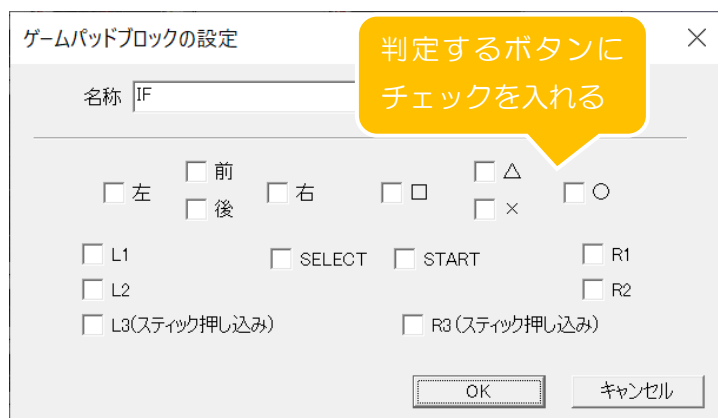
条件が成立する場合は赤いフロー、成立しない場合は青いフローに進みます。





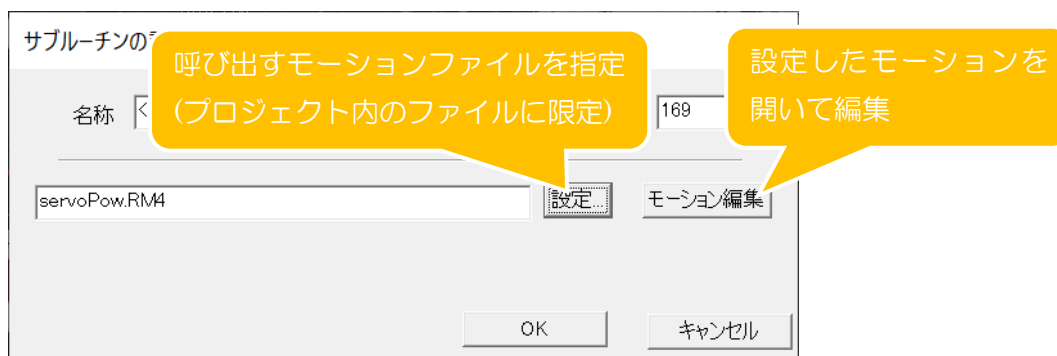
## 4-6. ゲームパッドブロック

ゲームパッドのボタン入力に特化した分岐ブロックです。現在のゲームパッドのボタン入力を参照し、チェックがつけられたボタンが全て入力されている場合は条件成立と判断します。

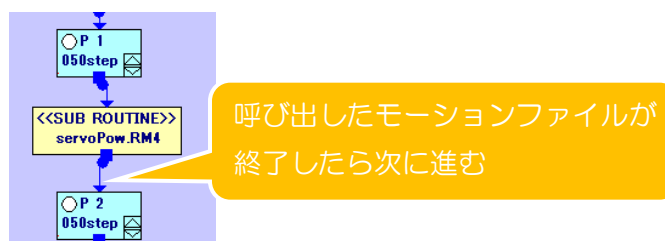


## 4-7. サブルーチンブロック

ファイルに保存した別のモーションを呼び出します。設定ダイアログでは呼び出すモーションファイルを指定します。モーションはプロジェクトに含まれるファイルのみ指定可能です。また、「モーション編集」をクリックするとサブルーチンに指定したモーションを開いて編集します。

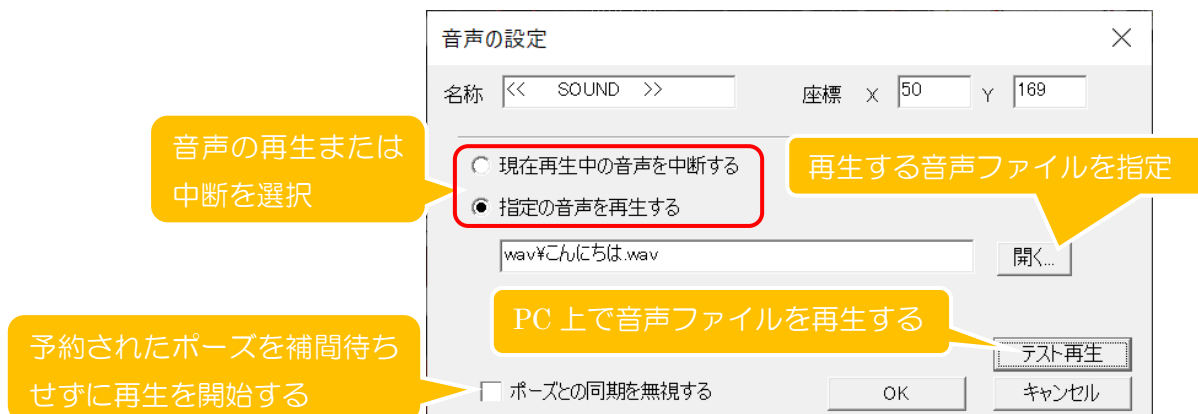


呼び出したモーションが終了したら、現在のモーションに戻って続きを実行します。サブルーチンは多重に呼び出すことが可能ですが、ネスト可能なのは 16 段階までです。

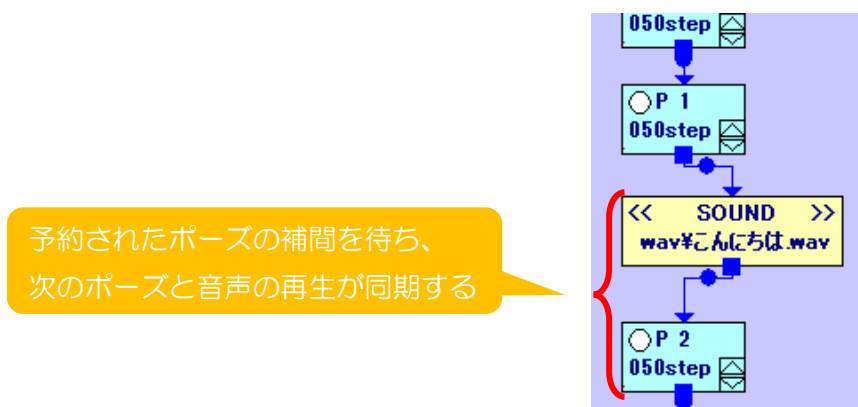


## 4-8. 音声ブロック

Wave 形式の音声を再生します。音声データは 44kHz までのモノラル形式の Wave データに対応しています。また、音声ファイルはプロジェクトのフォルダに含まれるものに限定されます。なお、CPU ボードは再生する音声ファイルを SD カードから読み込むため、SD カード内にプロジェクトと同じパスで再生する音声ファイルをコピーしておく必要があります。音声ファイルのコピーは手動で行う以外に、「[5. スタンドアローンでの動作](#)」に従って本体にモーションを転送すると、必要な音声ファイルを自動的にコピーします。

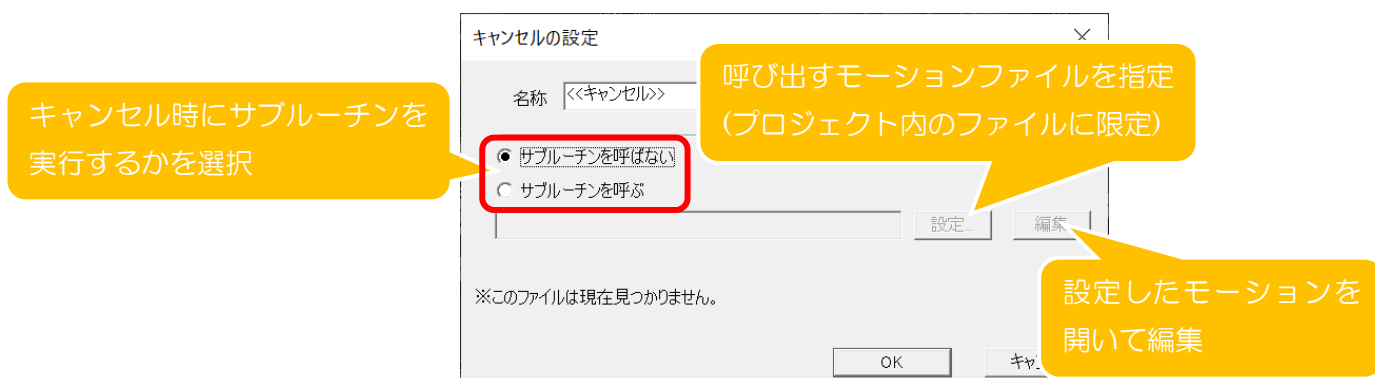


音声とポーズ実行を同期させるため、音声再生前に CPU ボードが予約しているポーズを全て補間し終えてから音声を再生します。もし補間待ちをせずにブロックを実行する場合は、「ポーズとの同期を無視する」をチェックしてください。



#### 4-9. キャンセルブロック

ゲームパッドの入力を読み取り、キャンセルボタンに割り当てられた入力を得られた場合に、モーションを終了します。キャンセルボタン(キャンセルとみなす入力)は「[拡張機器に関する設定](#)」より任意に設定できます。また、キャンセルが行われた場合に任意のモーションをサブルーチンとして実行してからモーションを終了できます。

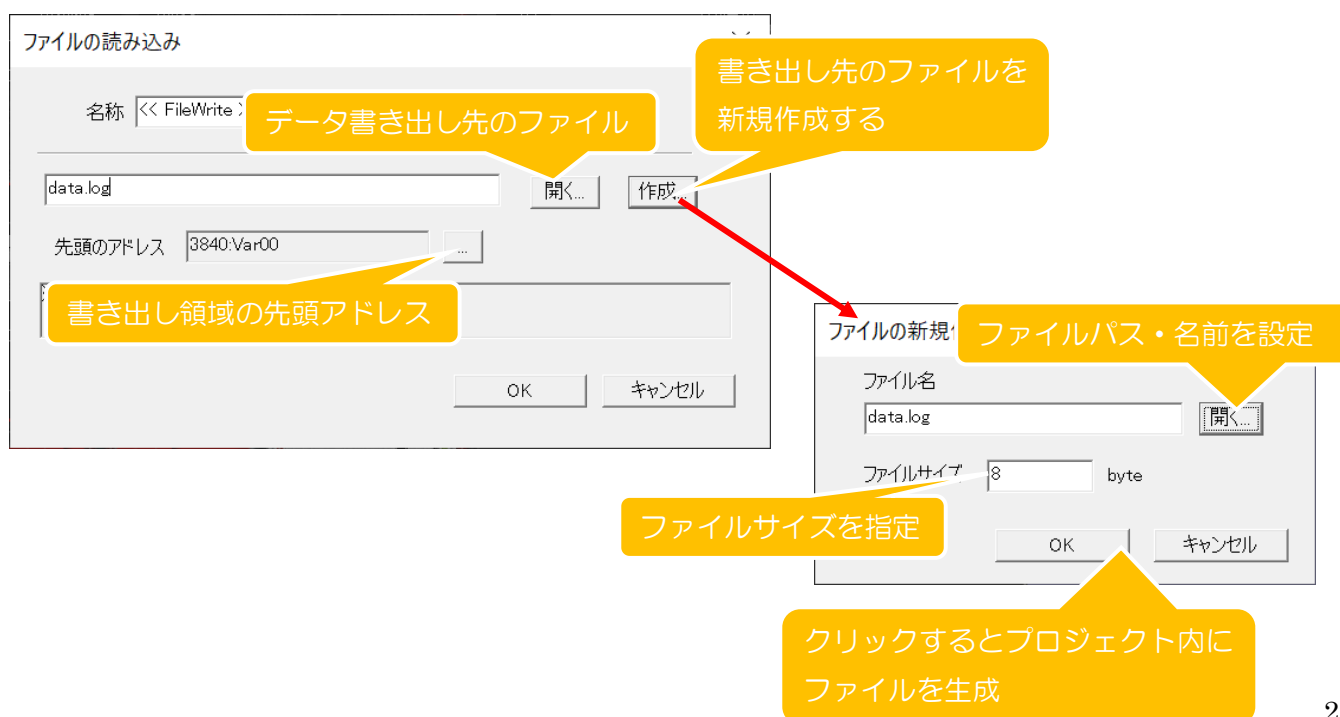


なお、ゲームパッドブロックなどとは異なり、キャンセルブロックではボタンの入力を取りこぼさないように、ボタンの入力を「押している間」(MU16\_PSPAD\_BTN の変数)で判断するのではなく、「一度でも押された場合」(MU16\_PSPAD\_BTNH の変数)で判断します。そのため、複数のボタン入力を設定している場合、状況によっては個別にボタンを押した場合でもキャンセルとして判定される場合があります。

#### 4-10. ファイル書き込みブロック

メモリマップの連続する領域の内容を SD カードに出力します。メモリマップについては「[6. メモリマップ](#)」をご参照ください。出力先のファイルはあらかじめプロジェクト内に作成し、SD カードにコピーする必要があります。ファイルに書き出す内容は、設定ダイアログで指定した先頭アドレスから始まる、書き出す先のファイルサイズ分の連続領域です。

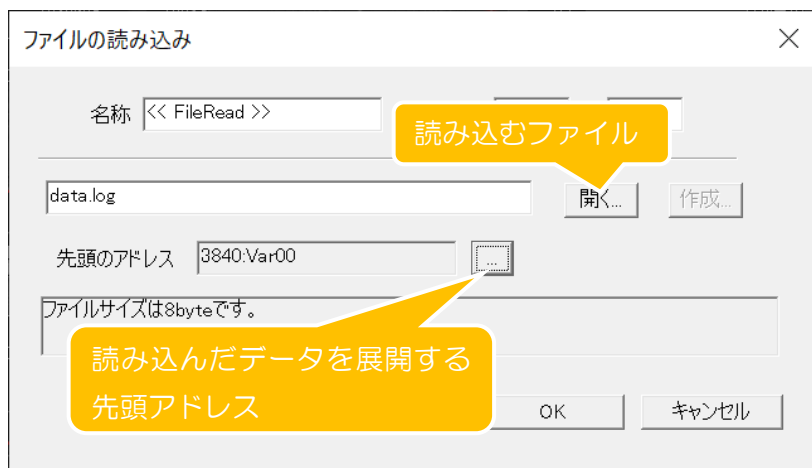
書き出すファイルのコピーは手動で行う以外に、「[5. スタンドアローンでの動作](#)」に従って本体にモーションを転送すると、必要なファイルを自動的にコピーします。



#### 4-11. ファイル読み込みブロック

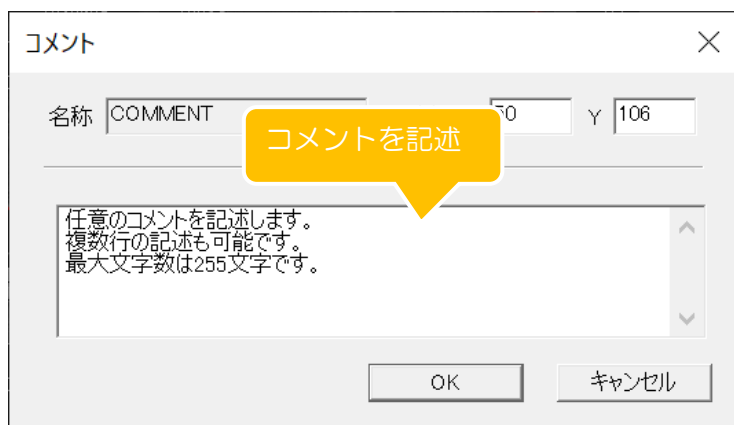
SD カードからバイナリファイルを開いて、読み込んだ生値をメモリマップの特定のアドレスに展開します。メモリマップについては「[6. メモリマップ](#)」をご参照ください。読み込むファイルはあらかじめプロジェクト内に作成し、SD カードにコピーする必要があります。読み込んだデータは、設定ダイアログで指定した先頭のアドレスから、読み込んだファイルサイズ分の連続領域に展開します。

読み込むファイルのコピーは手動で行う以外に、「[5. スタンドアローンでの動作](#)」に従って本体にモーションを転送すると、必要なファイルを自動的にコピーします。




#### 4-12. コメントブロック

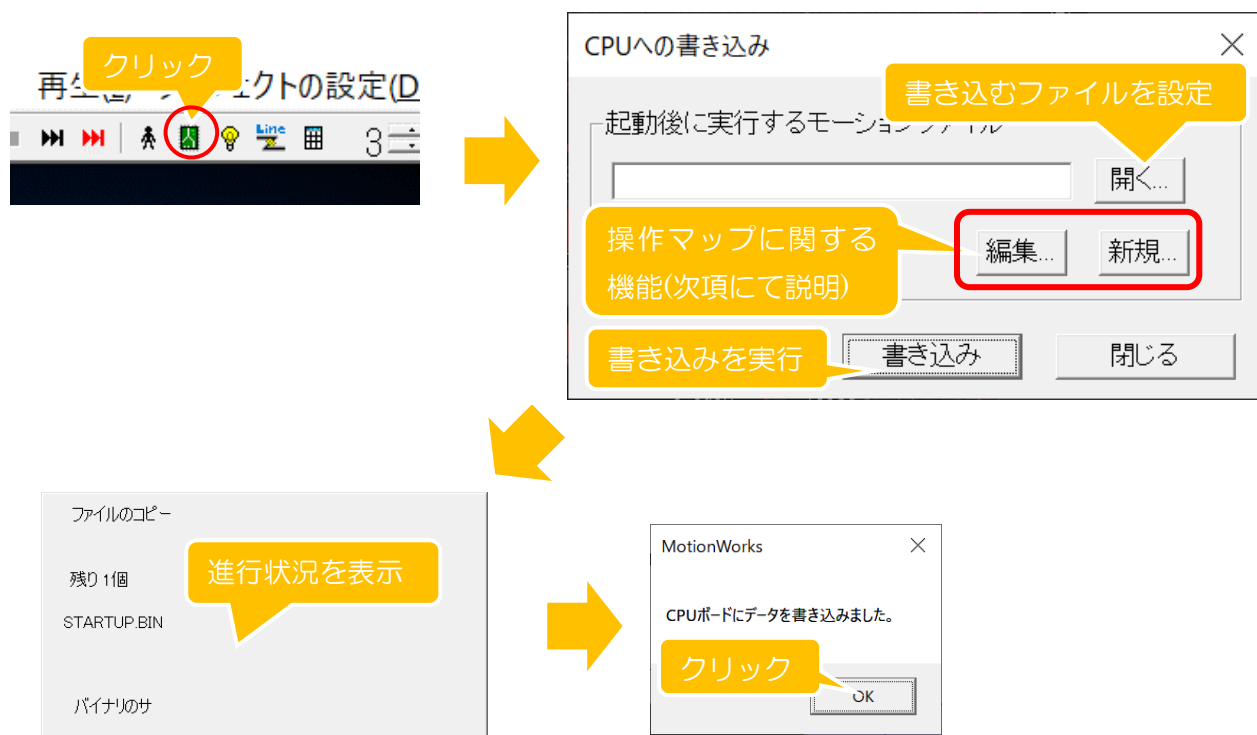
モーションエリア上にコメントを表示するために使います。あくまで編集時のメモ用途であり、コメントの内容は実行コードに含まれず、モーション再生でコメントブロックを通過しても何もせずそのまま次のブロックに進みます。



## 5. スタンドアローンでの動作

### 5-1. モーション・関連ファイルの転送

作成したモーションを CPU ボードに書き込むことで、スタンドアローンで動作させることができます。モーションの書き込みは、ツールバーより  ボタンをクリックして書き込みダイアログを開き、書き込むモーションファイルを指定して「書き込み」をクリックします。書き込みを開始すると、指定のモーションファイルを実行コードに変換し、実行コードファイル及び関連するすべてのモーションで使用する外部ファイル(音声・読み書きするデータ等)を転送します。



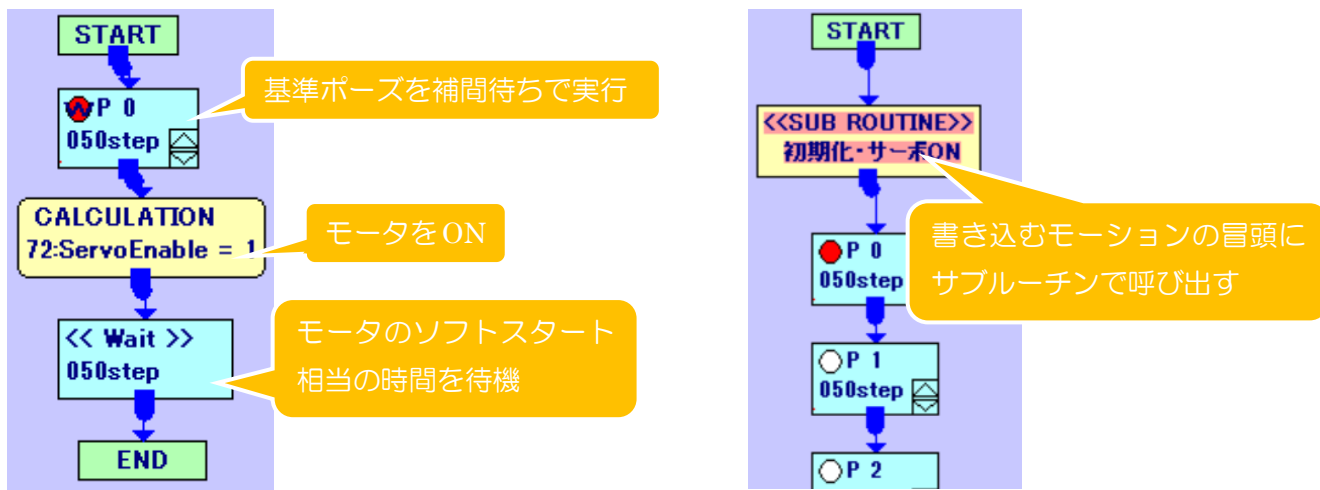
ロボットに書き込めるモーションは一種類のみで、別のモーションを書き込むとそれまで設定していたモーションが上書きされます。そのため、様々なモーションを呼び出したい場合は、サブルーチンなどを組み合わせたモーションファイルを自分でプログラムしてください。また、モータ ON など一部の初期化をユーザがモーションで作成して組み込む必要があります。詳細は「[初期化モーションの例](#)」をご参照ください。

書き込み中に ESC キーを押すと、書き込みを中断できますが、書き込みが始まった後の場合はモーションや関連ファイルが新しいもので上書きされている場合があります。

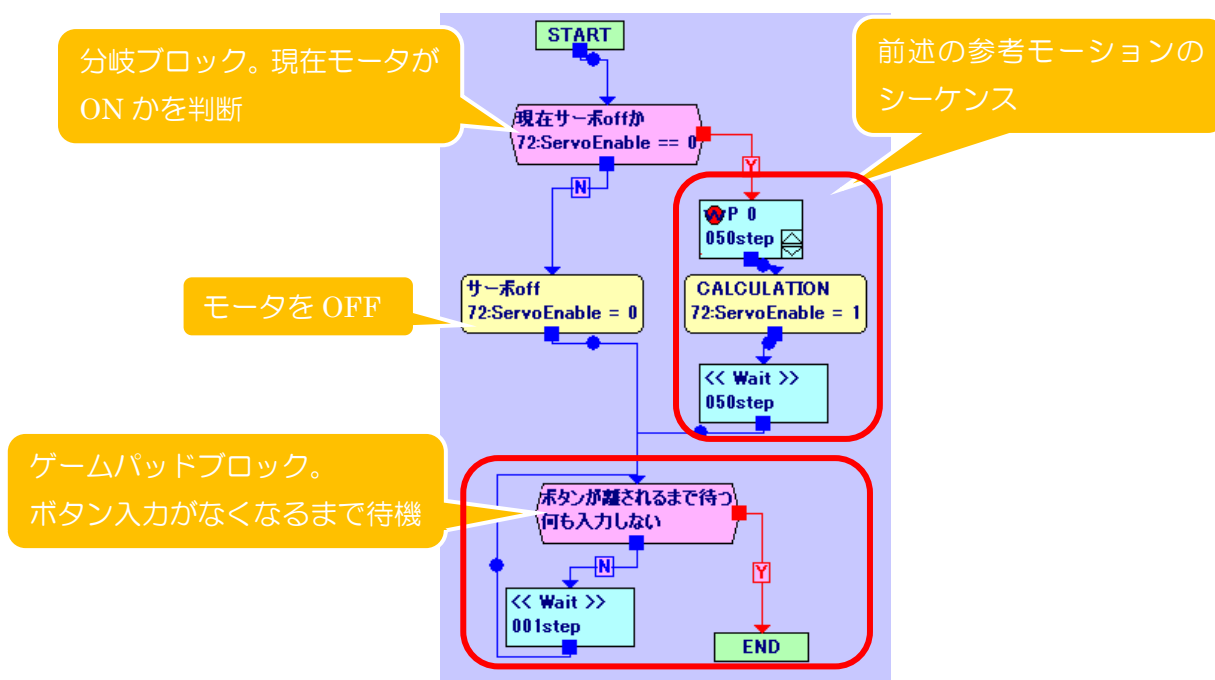
## 5-2. 初期化モーションの例

CPU ボードに関する基本的な設定は、モーションファイルの書き込み時に合わせて書き込まれますが、ポーズの初期化やサーボモータの ON と言った処理はユーザに委ねられるため、適時これらのモーションを作成して実行するモーションに組み込む必要があります。

基準ポーズ状態で安全にモータを ON にするモーションの参考例は下記になります。このモーションを、書き込むモーションの冒頭にサブルーチンとして実行したり、操作マップの起動モーションに組み込むと、ロボットが起動後緩やかに基準ポーズになり、完全に基準ポーズになった後で以降のモーションが実行されます。

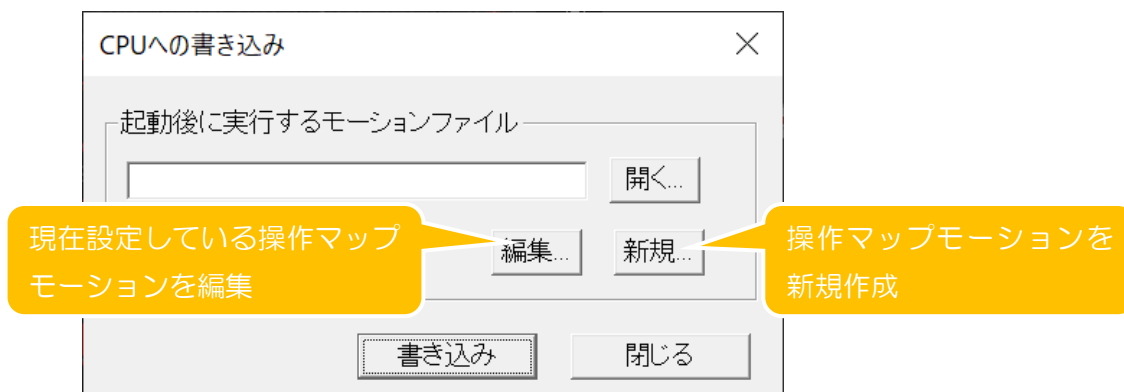


また、操作マップのボタン入力でモータの ON/OFF を切り替えたい場合は、下記のようなモーションを作成します。現在サーボモータが ON か否かを確認し、OFF の場合は前述の初期化モーションと同じシーケンスを実行します。ON の場合はメモリマップからモータを OFF にします。各処理の実行後、いきなり次のモーションに進まないように、コントローラのボタン入力がなくなるまで待機させます。

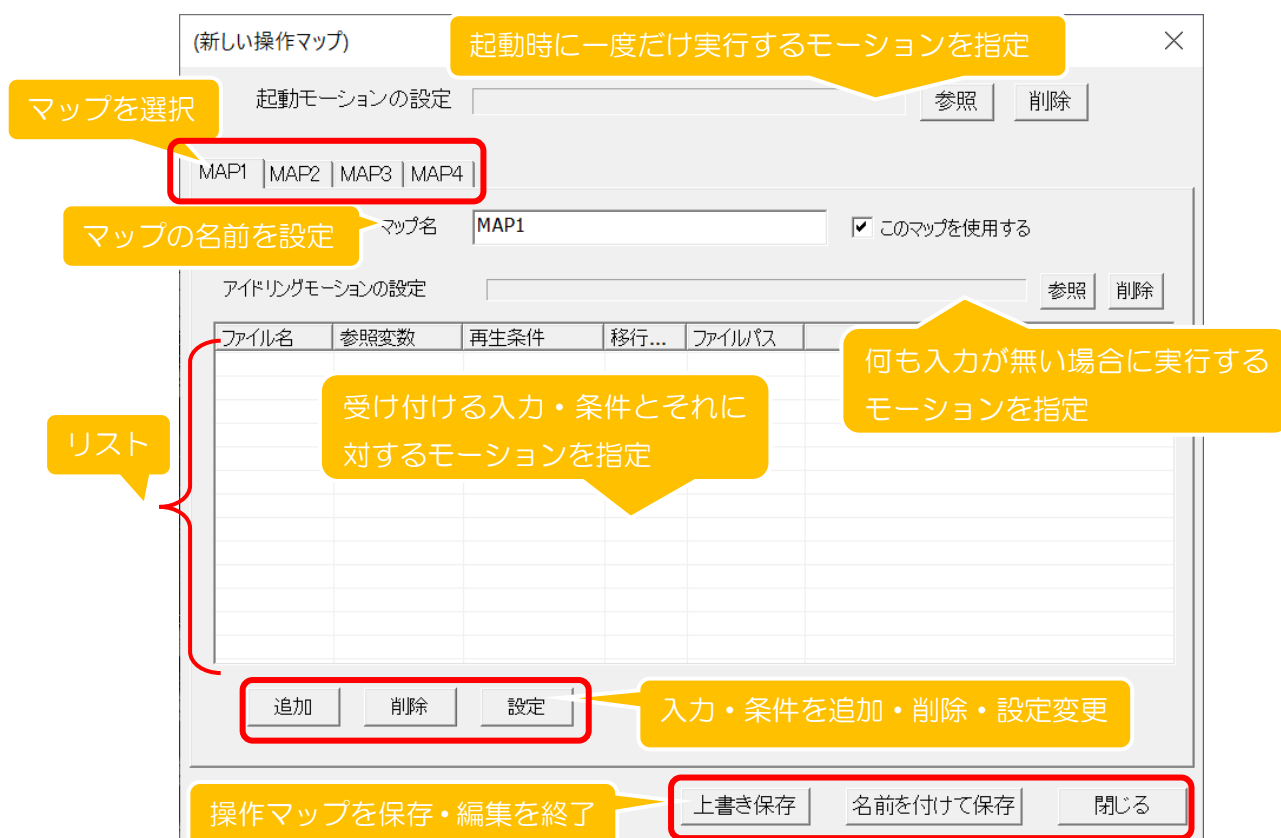


### 5-3. 操作マップの作成

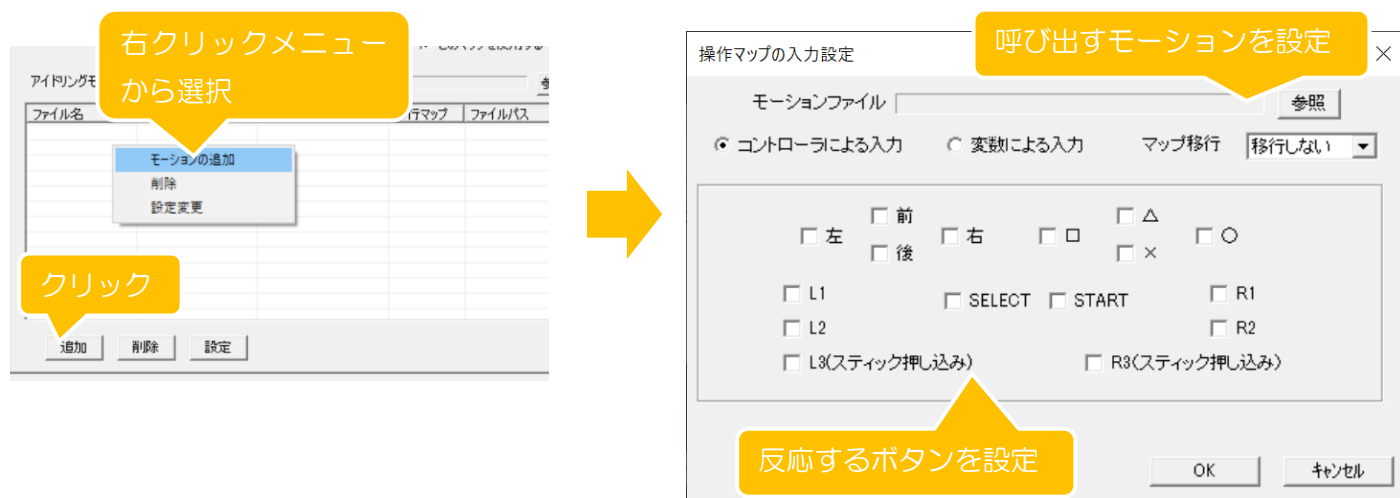
主にゲームパッドでの操縦に特化したモーションをダイアログから自動生成する機能です。書き込みダイアログの「新規」をクリックすると新しく操作マップのモーションを作成開始し、「編集」をクリックすると、既に設定している操作マップモーションを編集できます(操作マップモーション以外が設定されているとエラーが出ます)。



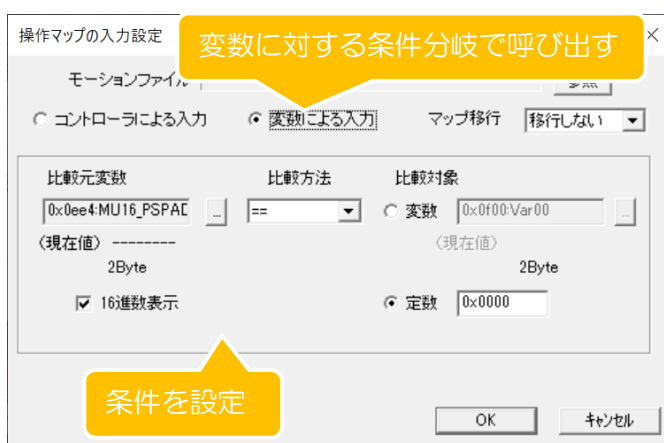
操作マップでは、ゲームパッドのボタンの入力状態やメモリマップの変数に対する条件を設定し、それに対して呼び出すモーションを割り当てます。作成した操作マップモーションを書き込んで実行すると、コントローラのボタン入力やセンサの値に反応してモーションを呼び出すことができます。



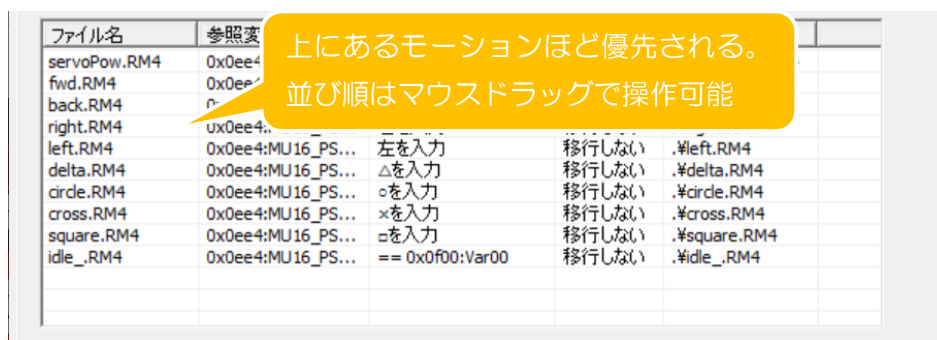
モーションを追加する場合は「追加」ボタンをクリック、もしくはダイアログ中央の「リスト」を右クリックしてメニューを表示し、「モーションの追加」をクリックします。クリックすると、反応するボタンとそれに対するモーションファイルを設定するダイアログを開くので、それぞれを設定します。複数のボタンをチェックすると、そのボタンが全て押された場合にモーションが呼び出されます。



「変数による入力」をクリックすると、メモリマップの変数に対する条件分岐によってモーションを呼び出す設定になります。条件の設定方法は[分岐ブロック](#)と同じです。



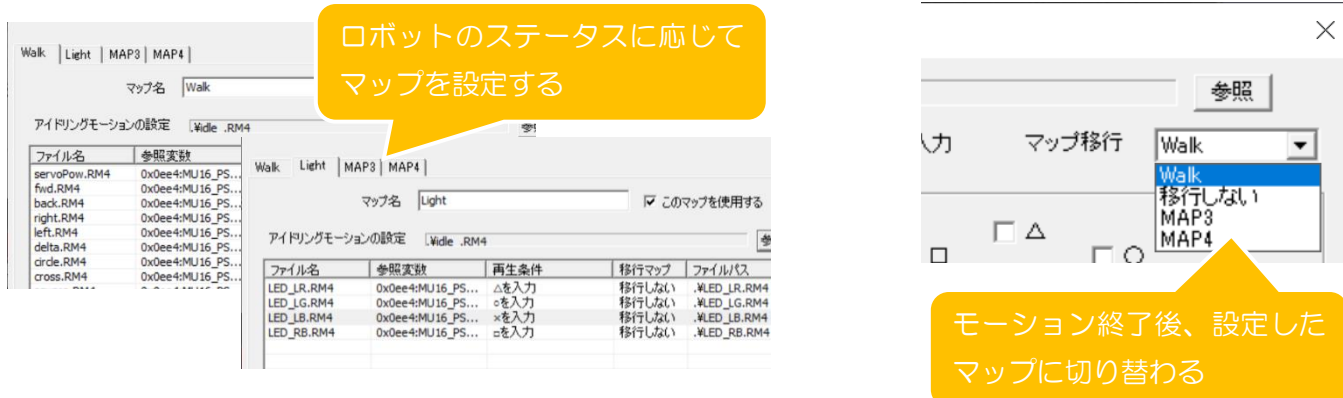
設定を完了すると、設定内容がリストに追加されます。このような形で、各ボタンにモーションを割り当てていきます。実際にモーションを呼び出す場合は、リストの上にあるモーションが優先して実行されます。リストの並び順を変更する場合は、項目をマウスでドラッグしてください。





「アイドリングモーション」はリスト内に合致するモーションが無い場合に実行します。また「起動モーション」はロボットの起動時に一度だけ実行します。こちらは「[初期化モーションの例](#)」で説明しているような初期化のためのモーションを割り当てます。

操作マップには「マップ」の概念があり、例えば「直立状態のモーション群」「しゃがみ状態のモーション群」と言った異なる状況のモーションを使い分けたいときに、マップを利用することで明確にロボットの状態を切り分けることができます。ボタン入力・条件を設定するダイアログの「マップ移行」を設定すると、そのモーションを実行した後にマップが切り替わります。また、ロボットの起動後は必ずマップ 1 になります。




コントローラでの操縦時に、ボタンを押してもすぐにモーションが実行されなかったり、数秒前に入力したボタンが指を離してから認識されたりする場合があります。これは、[ポーズブロックの説明](#)などで記述している、ブロックとポーズが同期せずに実行される仕組みによるものです。モーションの再生時には、予約に積まれたポーズはブロックの実行から遅れて実行される一方、コントローラの入力を読み取るブロックの実行はポーズより先行して行われるため、このような遅延が発生します。

対策として各モーションの最後のポーズを補間待ちさせると、モーション終了のタイミングでポーズとボタンの読み取りが同期するため、意図しないような先行入力は防止できます。

また、ボタン入力のレスポンスを向上させる場合、アイドリングモーションのポーズ数を減らしたり、遷移時間を小さくする方法や、メモリマップから現在予約に積まれているポーズ数を取得できるので、極力ポーズ数を減らすなどの対策があります。

## 6. メモリマップ

CPU ボードはメモリマップド I/O の仕組みを持っており、モーション等から任意の変数を読み書きできます。ツールバーの  ボタンをクリックするとメモリマップの一覧を表示します。ロボットと通信している場合は、リアルタイムで数値を確認できます。ダイアログ下部の検索ウィンドウに探したい変数の名称の一部を入力すると、その文字列が含まれる変数のみがリストアップされます。



また、演算ブロック・分岐ブロックなどで変数を選択する場合、リストアップされた変数をダブルクリックすると、その変数を選択できます。

メモリマップダイアログの内容は、プロジェクト内の ini フォルダに入っている memmap.csv に「アドレス、サイズ(byte)、名前」の csv 形式で定義されています。また、詳細説明は別資料にて配布しております。

memmap.csv はプロジェクトの内容に応じて任意に書き換えることができますが、あらかじめ機能が割り当てられたアドレスに別の目的を定義すると、意図しない挙動を引き起こす可能性があるため、定義するアドレスをよく確認して割り当ててください。

## 7. ポーズに関する機能

### 7-1. ポーズスライダの設定

ポーズスライダを右クリックして「ポーズスライダのプロパティ」をクリックすると、ポーズスライダの設定を行います。設定ダイアログには下図の区分の項目が備わっています。

The screenshot shows the 'ポーズスライダの設定' (Pose Slider Settings) dialog box. It is divided into several sections:

- 設定するポーズスライダ:** A dropdown menu showing '28-ID15 POS'.
- テキスト設定:** Includes a text field for 'スライダ名' (Slider Name) set to 'ID15 POS', buttons for 'フォント' (Font) and '文字背景色' (Text Background Color), and text fields for 'スライダ横の文字列' (Slider Label) with '上限側' (Upper Limit Side) and '下限側' (Lower Limit Side) options.
- 数値設定:** Includes radio buttons for '書式' (Format): '10進数' (Decimal), '16進数' (Hexadecimal), '角度(deg)' (Angle), and 'パーセント' (Percentage). It also has fields for '可動範囲制限(下限~上限)' (Range Limitation) set to '-1500 ~ 1500', 'ステップ分解能' (Step Resolution) set to '5', '対応表示角度 1度=' (Corresponding Display Angle 1 degree) set to '10.0000', and '小数点以下の桁数' (Number of Digits After Decimal Point) set to '1'.
- 左右設定:** Includes radio buttons for '属性' (Attribute): '設定無し' (None), '中心' (Center), '右' (Right), and '左' (Left). It also has a dropdown for '対となるスライダ' (Opposite Slider) set to '00-ID1 POS'.
- スライダの位置/フラグ設定:** Includes text fields for '表示座標 X' (Display Coordinate X) set to '240' and 'Y' (Display Coordinate Y) set to '30'. It also has several checked checkboxes: 'スライダ有効' (Slider Enabled), '表示反転' (Display Inverted), '符号反転' (Sign Inverted), '基準ポーズ相対値' (Reference Pose Relative Value), 'ポーズスライダ名表示' (Pose Slider Name Display), and 'スライダ横の文字列表示' (Slider Label Display).

Four yellow callout boxes point to specific sections:

- '表示名やテキストに関する設定' (Settings for display name or text) points to the 'テキスト設定' section.
- '数値単位や動作範囲に関する設定' (Settings for numerical units or operating range) points to the '数値設定' section.
- '左右対称軸に関する設定' (Settings for left-right symmetry axis) points to the '左右設定' section.
- '表示位置と各種フラグに関する設定' (Settings for display position and various flags) points to the 'スライダの位置/フラグ設定' section.

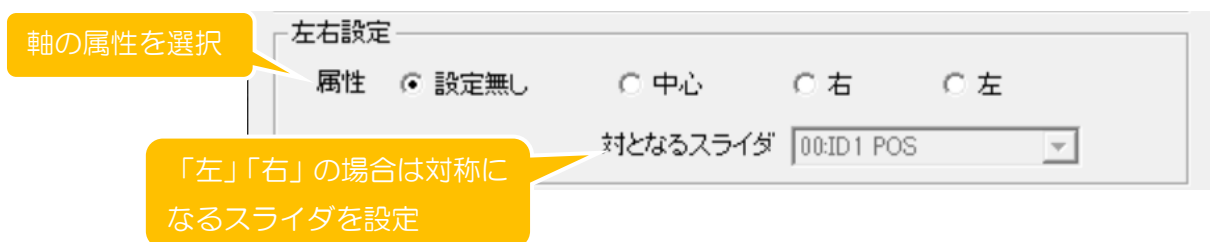
表示やテキストに関する設定の例です。スライダ横の文字列は、フラグ設定で表示の有無を切り替えられます。

This diagram illustrates the relationship between the settings in the dialog box and the resulting slider UI. On the left, a 'テキスト設定' (Text Settings) section shows 'スライダ名' (Slider Name) set to '首ヨー軸' (Neck Yaw Axis), 'スライダ横の文字列' (Slider Label) set to '左向き' (Leftward), and '下限側' (Lower Limit Side) set to '右向き' (Rightward). A yellow arrow points from these settings to the slider UI on the right. The slider UI shows the name '首ヨー軸' at the top, '左向き' (Leftward) on the left and '右向き' (Rightward) on the right, a central slider with a red indicator, and numerical values '+0.0deg' and '100' with a checked checkbox.

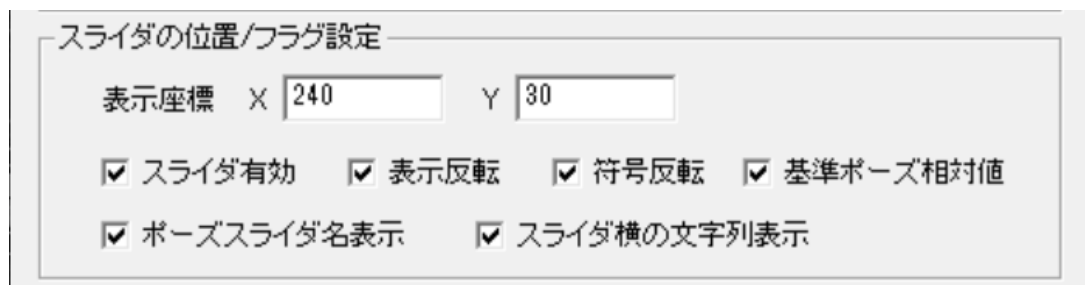
ポーズスライダは、内部的には符号付 16bit(-32768~32767)の数値ですが、表示の際に角度や割合(パーセント)を選択できます。角度の場合は「対応表示角度」に指定した比率、割合の場合は可動範囲内における現在値の比率を表示します。それ以外は生値単位で表示します。可動範囲は生値ベースで設定します。



後述の半身コピーや鏡像反転機能のため、ポーズスライダに対称となる組み合わせや反転可能かどうかを設定できます。首や腰等、出力軸が体の中央を通るものは「中心」、手足等二つのポーズスライダが対称関係となるものは「右」「左」を、それぞれ設定します。後者は対称となるポーズスライダも選択します。対称となるポーズスライダは、一方を設定すると他方は自動的に設定が書き換わります。



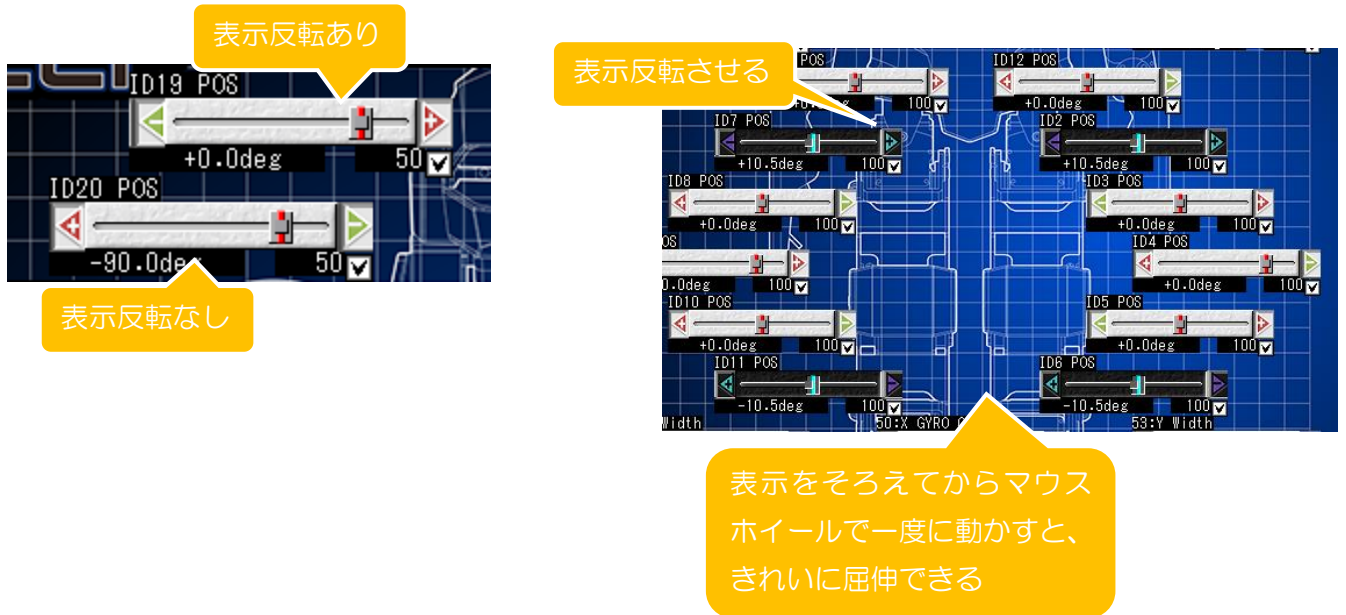
表示座標・フラグ設定では、ポーズスライダの描画等に関するフラグを設定します。各フラグの概要は後述の通りです。



- スライダ有効…ポーズスライダの表示の有無を選択。
- 表示反転…ツマミとスピンドットの表示位置を反転する。
- 符号反転…表示されている数値の符号を反転する。左右半身コピー・鏡像反転にも影響するため、対称となるスライダ同士はこのフラグを同じ設定にする。

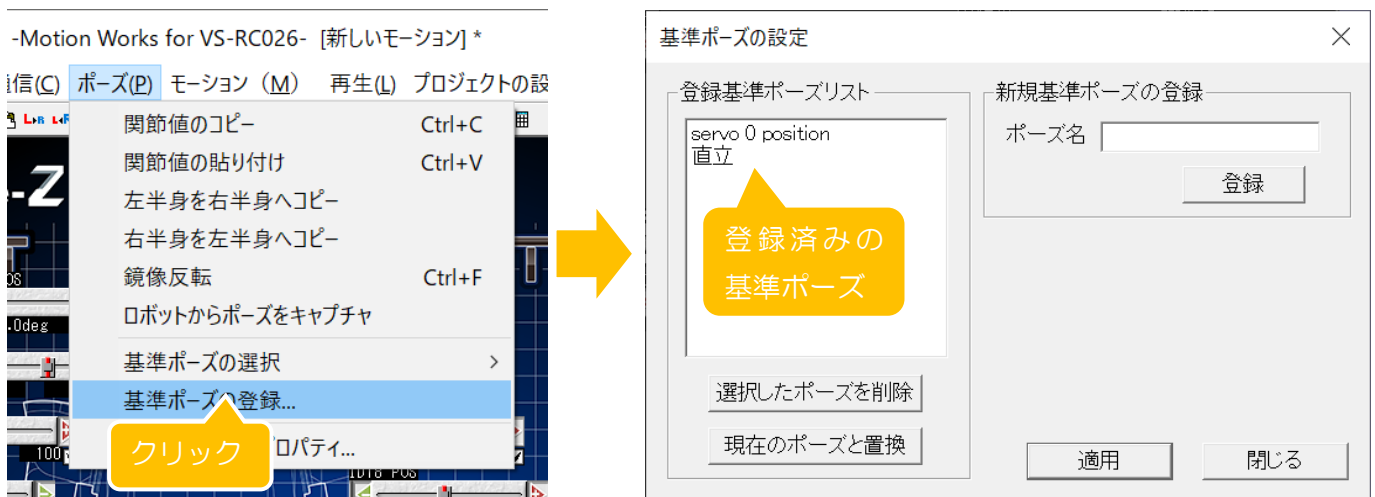
- 基準ポーズ相対値…数値の原点(±0)を、生値の±0 とするか基準ポーズの位置を±0 とするかを選択。
- ポーズスライダ名表示…スライダ上部に名称を表示するかを選択。
- スライダ横の文字列表示…スライダ左右に文字列を表示するかを選択。

表示反転の用途例としては、マウスホイールの操作で動くつまみの方向は常に同じなため、体の重心移動で大腿と足首のピッチ軸やロール軸のように対称となる軸に対して、軸の正逆転の方向に応じて表示を反転すると、きれいに重心移動ができます。



## 7-2. 基準ポーズの設定

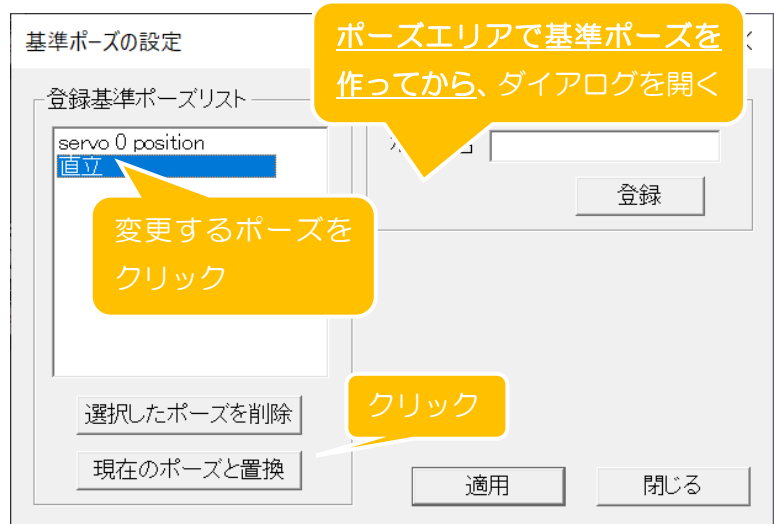
基準ポーズを追加・変更する場合は、メニューの「ポーズ」→「基準ポーズの登録」をクリックして設定ダイアログを表示します。



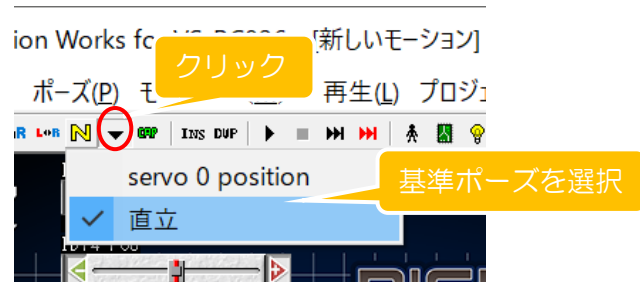
基準ポーズを新規登録する場合は、あらかじめポーズエリアで登録するポーズを作成したうえで、基準ポーズの設定ダイアログより「ポーズ名」を設定して「登録」をクリックします。



既存の基準ポーズを変更する場合は、同じくポーズエリアで変更後のポーズを作成したうえで、基準ポーズの設定ダイアログを開き、変更したいポーズをリストからクリックして「現在のポーズと置換」をクリックしてください。



複数登録した基準ポーズから一つを選択する場合は、ツールバーの **N** ボタン横の ▼ をクリックし、一覧の中からポーズを選んでください。以降は選択したポーズが基準ポーズとして使われます。





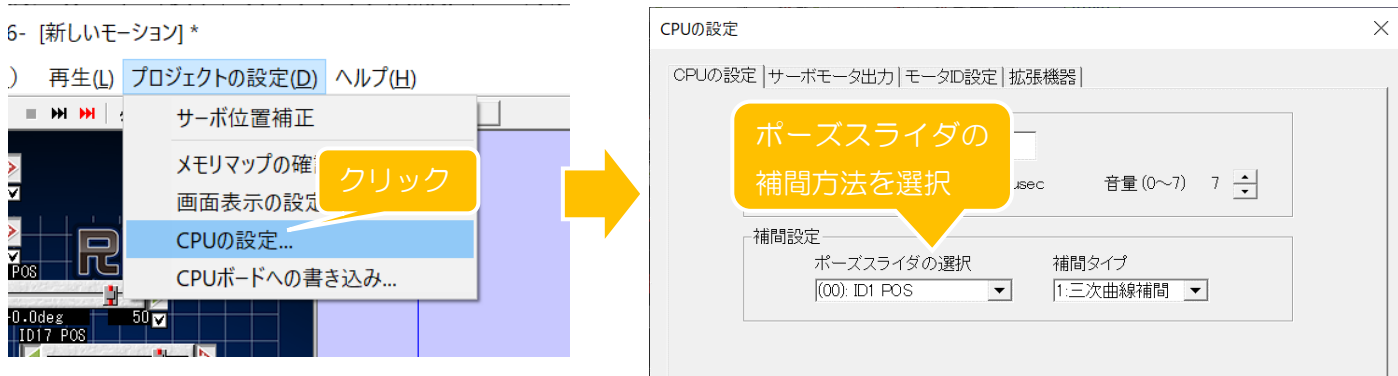
### 7-3. ポーズの補間器について

CPU ボード内では、ポーズスライダの指示値に対して常に補間を行っており、ポーズスライダの操作やモーションの実行等で指示値と遷移時間が与えられると、決められた補間方法で現在値から指示値に向けて補間を開始します。

補間方法はポーズスライダで個別に設定でき、下記の中から選択できます。

- 補間しない…指示値に対して一切補間をしません。特定のポーズスライダが動かない場合は補間方法がこれになっていないかご確認ください。
- 三次曲線補間…2 ポーズ先までの指示値をベースに滑らかに補間します。
- 直線補間…指示値に対して直線的に補間します。
- 遷移前同期切り替え…指示値に瞬時に切り替わり、与えられた遷移時間まで待ちます
- 遷移後同期切り替え…与えられた遷移時間まで補間せず、時間が経過したら瞬時に指示値に切り替わります。

補間方法の設定は、メニューより「プロジェクトの設定」→「CPU の設定」をクリックし、「CPU の設定」タブの「補間設定」で行います。



ロボットと通信中は、CPU ボードの補間値はポーズスライダ上にツマミの影のような状態で表示されます。ただし、モータが割り当てられたポーズスライダは、補間値ではなくモータからキャプチャされた現在値が表示されます。



## 7-4. ポーズ作成に便利な機能

ポーズの作成に便利な機能を紹介します。

ポーズスライダを選択して Ctrl+C キーを押すと現在のポーズをコピーし、Ctrl+V キーを押すとコピーしたポーズを貼り付けます。

「[ポーズスライダの設定](#)」で紹介した左右設定を行うと、ツールバーの **L/R** ボタンをクリックすると右半身のポーズを左半身にコピーします(**L/R** ボタンで逆に左半身から右半身へコピー)。ツールバーの **L/R** ボタンをクリックすると、右半身と左半身のポーズを鏡像反転します。これらの機能もポーズスライダを選択すると、そのスライダのみに対して行います。

ロボットと通信してツールバーの **OFF** ボタンをクリックすると、現在のロボットのポーズを CPU ボードから読み取ってポーズスライダに反映させます。モータを OFF にし、手でロボットのポーズを作ってキャプチャすることが可能です。

ツールバーの **Ins** ボタンをクリックすると、現在編集中のポーズとその次のポーズの間に、両者の中間の（両者を平均した）ポーズを挿入します。ポーズ間に別の動作を挿入したり、動きが不十分で関節が干渉するといった問題が見られる場合は、この機能で間の動きを細かく指定します。

ポーズスライダを選択してマウスホイールを操作すると一度に値を変更できますが、この時 SHIFT キーを押していると、増減値が約 10 倍になります。Ctrl キーを押していると増減値が約 1/8 になります。

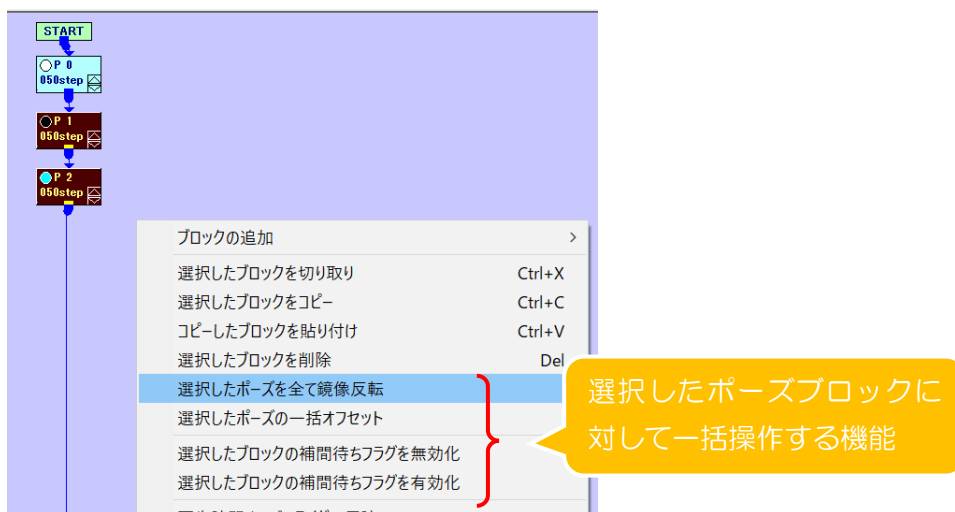
モーションエリアとポーズエリアの境界をマウスでドラッグすると、それぞれのエリアの幅を変更できます。また、それぞれのエリアの余白部分をダブルクリックすると、モーションエリアを最小化できます(再度ダブルクリックすると元に戻ります)。

ポーズを再生せずに一つずつ進めて見たい場合は、キーボードのカーソルキー(←、→キー)を押すと、フローに従ってポーズを次に進めます。また、基本的には青のフローに従って進みますが、SHIFT キーを押しながら操作すると、赤のフローに従って進めることができます。



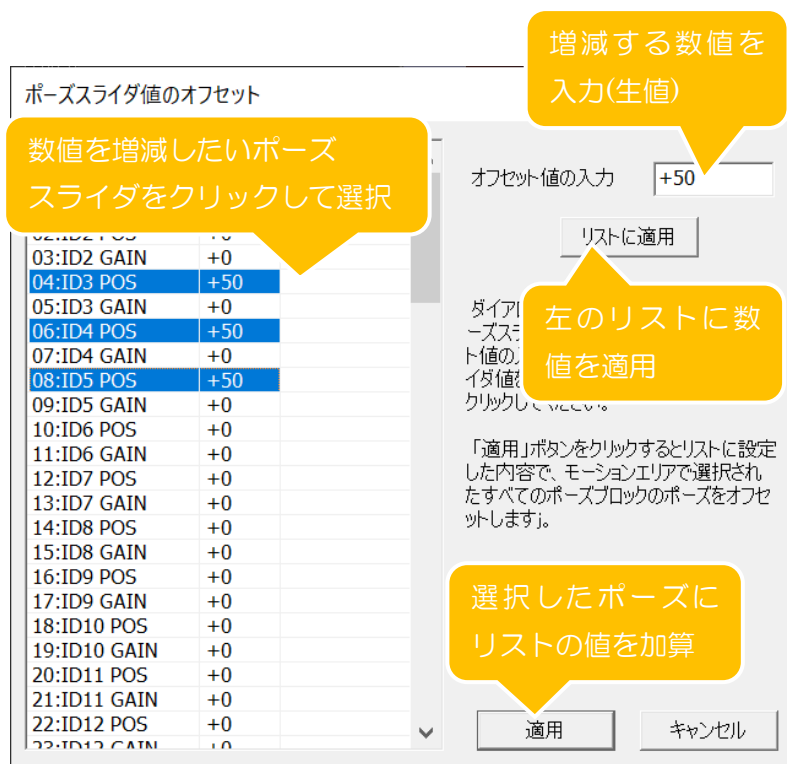
## 7-5. ポーズの一括操作

モーションエリアで複数のポーズブロックを選択していると、右クリックメニューの一部の項目でそれらのポーズに対して一度に同じ操作を適用できます。



「選択したポーズをすべて鏡像反転」は、選択したポーズすべてにツールバーの ボタンと同じ操作を行います。「選択したブロックの補間待ちフラグを無効化/有効化」は、選択したポーズの補間待ちの設定を一括で変更できます。

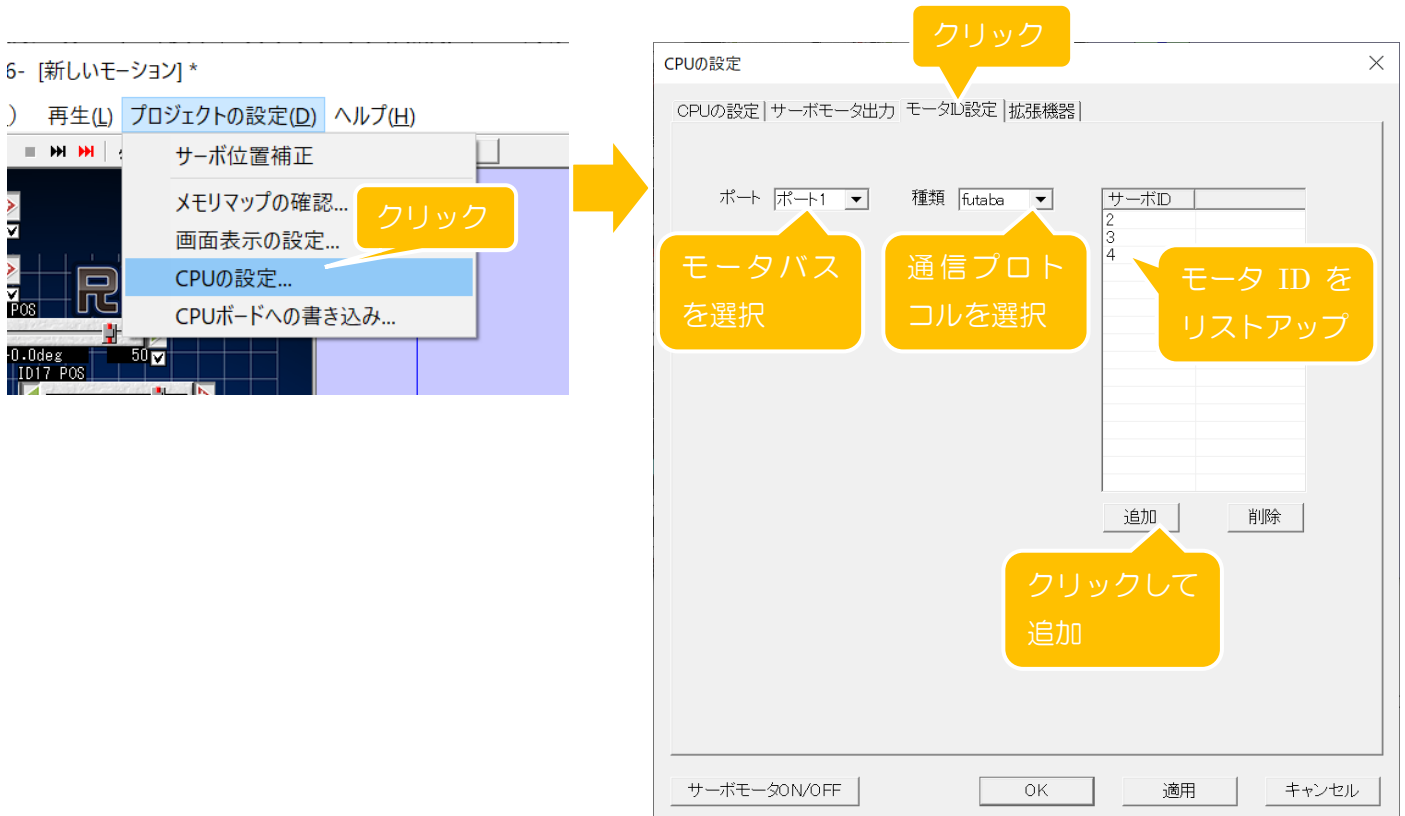
「選択したポーズの一括オフセット」はポーズスライダごとに増減値を設定し、それを選択したポーズに一括して加算する機能です。ここでの数値は全て生値(符号付 16bit)単位です。



## 8. モータに関する設定

### 8-1. モータ ID の設定

CPU ボードからモータを制御する場合は、CPU ボード上の各モータ出力バスに対して、接続されたモータの ID を指定する必要があります。モータの ID の設定は、メニューより「プロジェクトの設定」→「CPU の設定」をクリックし、表示されたダイアログの「モータ ID 設定」のタブをクリックします。



「ポート」は、設定するポートを選択します。CPU ボードには 4 つのモータバスがあり、うち 2 つは基板上の出力コネクタが 2 か所に分かれています。ポート番号とコネクタ名の対応は以下の通りです。

- ポート 1:LARM, HEAD
- ポート 2:RARM, BODY
- ポート 3:LLEG
- ポート 4:RLEG

「種類」は各バスにつないだモータの通信プロトコルを設定します。双葉電子社製のモータの場合は「futaba」を選んで下さい。また、同じバスに異なるプロトコルのモータを混在させることはできません。

「サーボ ID」は、バスにつながっているモータの ID をリストアップします。「追加」をクリックして ID の数値を入力してください。なお、一つのバスに接続できるモータ数は最大で 14 個ですが、通信速度などの関係から、極力一つのバスに多数のモータを集中して接続せず、適度に分散して接続することを推奨します。また、CPU ボード自体が扱えるモータ数は、全バスの合計で 32 個までです。

本ソフトウェア上のモータ ID は、接続の順番やバスの場所に関係なく、実際のモータ ID と一致します。

## 8-2. 個別のモータの出力設定

各モータには、どのポーズスライダをアサインするか、また、センサ値のフィードバックを行うかなどの設定が存在します。基本的な項目は最初からプロジェクトに設定されていますが、モーション内の演算結果や拡張機器の入力と連動させるといった発展的な使い方をする場合は、別途必要な設定を追加します。

設定を行う場合は、メニューより「プロジェクトの設定」→「CPU の設定」をクリックし、表示されたダイアログの「サーボモータ出力」のタブをクリックします。

この図は、ソフトウェアの「CPUの設定」ダイアログボックスを示しています。左側には「プロジェクトの設定」メニューから「CPUの設定...」を選択する様子があり、右側には「サーボモータ出力」タブがアクティブな設定画面が示されています。

設定画面の「基本設定」セクションには、**サーボモータ名** (ID1) と **サーボモータのソフトスタート時間** (60 サイクル) が設定されています。

「出力値設定」セクションには、 $x = [3072:Unkown] + 0$  (位置補正值) と、4つの乗算項が定義されています。各乗算項は  $0 * [0:Unkown] / 256 * [0:Unkown] / 256$  の形式で、すべて「常に有効」の状態です。

「キャプチャ変数/ゲイン変数設定」セクションには、**キャプチャ変数** ([3456:ID01 Positio]) と **ゲイン参照変数** ([3074:Unkown]) が指定されています。

出力値範囲は  $-32767 \leq x \leq 32767$  と設定されています。

ボタンには「サーボモータON/OFF」、「OK」、「適用」、「キャンセル」があります。

基本設定ではサーボモータ名とソフトスタート時間を設定します。サーボモータ名はこのダイアログのみで有効な名前です。ソフトスタート時間は、モータを ON にした際に徐々に立ち上がる (ON になる) 機能のことで、完全に立ち上がるまでの時間を設定します。ここでの単位 (サイクル) は、ポーズの遷移時間の単位 (step) と同等で、1 サイクル  $\approx$  16 ミリ秒なので、60 サイクルの場合約 1 秒でモータが ON になります。

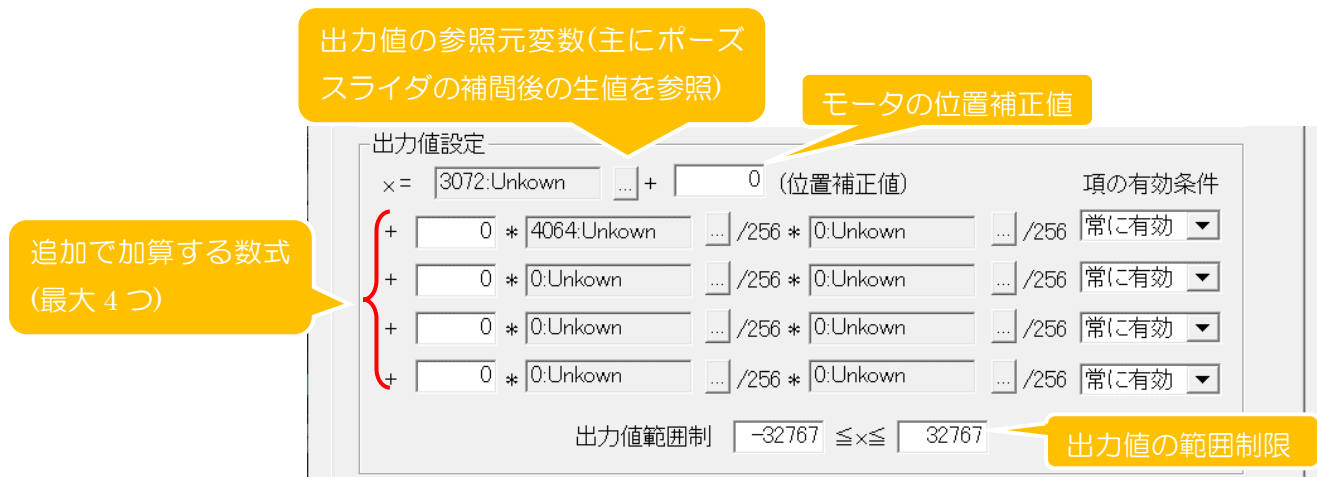
この図は、「基本設定」セクションの詳細を示しています。

**サーボモータ名** (ID1) は、ダイアログ上のモータの表示名を設定するためのフィールドです。

**サーボモータのソフトスタート時間** (60 サイクル) は、モータ ON 時の立ち上がり時間を設定するためのフィールドです。

出力値設定では、モータへの出力値を求める各種設定を行います。モータの出力は、任意の変数の値(主にポーズスライダの生値(補間器の値))に対して、「[3-3. サーボモータの位置補正](#)」で調整したモータの個体差の補正値を加算して求められます。また、更に最大4つの数式を加算できます。また、最終的に求めた値に対して範囲制限をかけることができます。

なお、本項目で使用する変数・定数は全て符号付16bitとして扱われます。仕様・サイズの異なる変数を指定した場合、意図しない演算結果になるため、ご注意ください。



加算できる数式は、全て以下の形式となります。このうち、定数・変数 A・変数 B の三つの数値を設定できます。

$$[\text{定数}] \times [\text{変数 A}] \div 256 \times [\text{変数 B}] \div 256$$

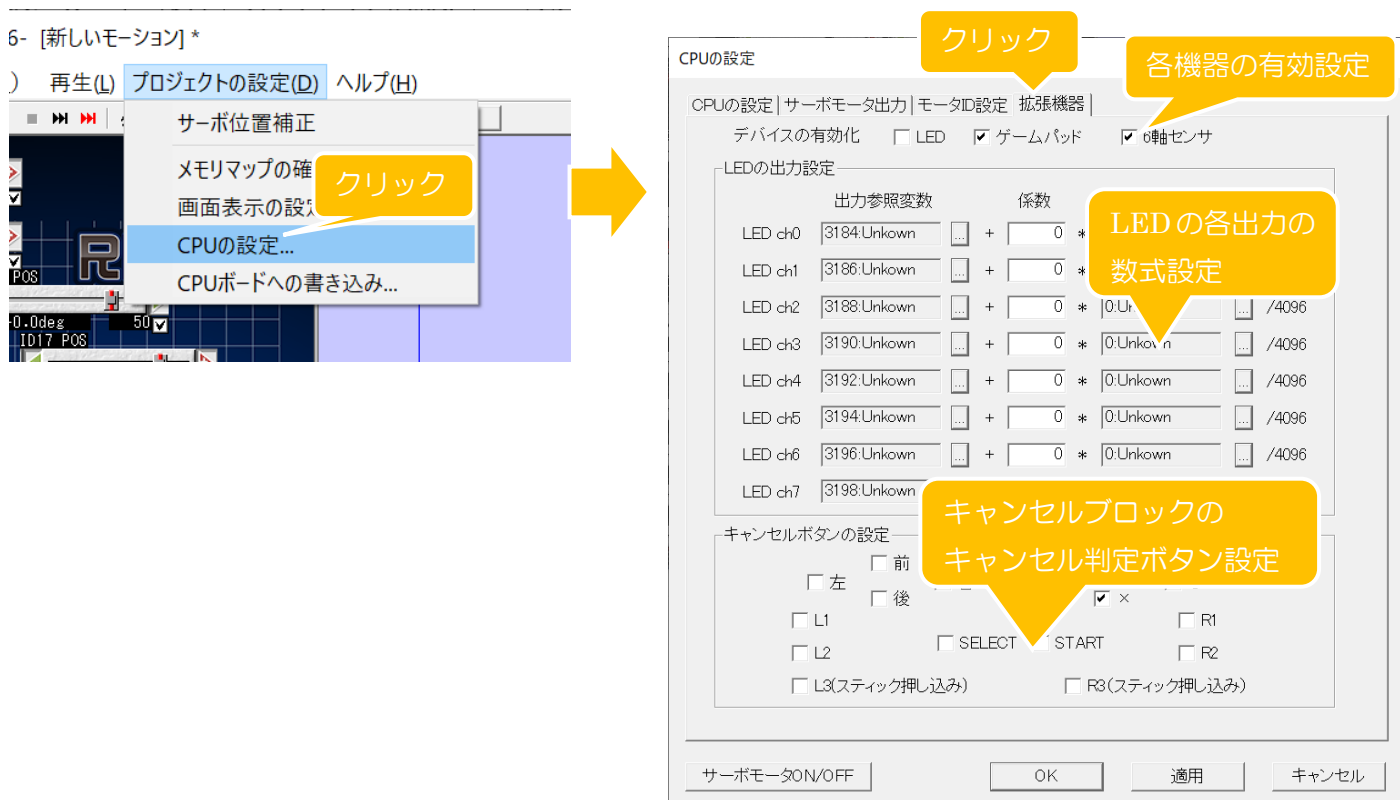
また「項の有効条件」は、変数 B の数値が 0 以上・0 以下のいずれかの場合のみ数式を有効にする設定です。

キャプチャ変数/ゲイン変数設定では、モータから取得したポジションキャプチャの値を代入する変数、及びモータのゲイン(出力トルク)として参照する変数をそれぞれ設定します。基本的に、前者はメモリマップの MS16\_VAR\_RPOS0(0x0d80~0x0dbf)、後者は一部の偶数のポーズスライダがデフォルトで設定されており、新たに設定し直す必要はありません。



## 9. 拡張機器に関する設定

拡張機器に関する設定は、メニューより「プロジェクトの設定」→「CPU の設定」をクリックし、表示されたダイアログの「サーボモータ出力」のタブをクリックします。CPU ボードに接続した拡張機器を有効にする場合は、本ダイアログよりデバイスの有効化を行ってください。また、LED の出力値の算出方法、及び[キャンセルブロック](#)を使用した場合のキャンセル判定入力について、補助設定があります。



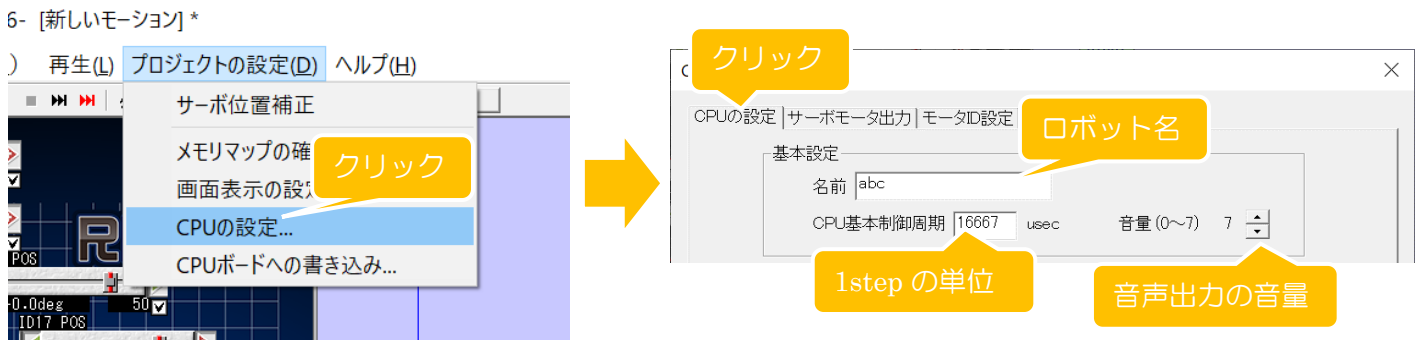
LED の出力設定では、LED 基板の 8ch の各出力について、参照する変数及び加算する数式を設定します。なお、LED の出力で使用する変数・定数は全て符号付 16bit として扱われます。仕様・サイズの異なる変数を指定した場合、意図しない演算結果になるため、ご注意ください。

キャンセルボタンの設定は、[キャンセルブロック](#)を実行したときにキャンセルとして判定されるボタン入力を設定します。チェックされたボタンが全て押されている場合にキャンセルと判定されます。なお、キャンセルブロックの説明で記述している通り、判定として参照する変数は「一度でも押されたボタンのビットフラグ」(MU16\_PSPAD\_BTNH の変数)のため、複数のボタン入力を設定している場合、状況によっては個別にボタンを押した場合でもキャンセルとして判定される場合があります。

# 10. その他の設定・機能説明

## 10-1. CPU の設定

メニューより「プロジェクトの設定」→「CPU の設定」をクリックし、表示されたダイアログの「サーボモータ出力」のタブをクリックすると、CPU ボードに関する設定を開きます。ダイアログでは、ロボット名・基本制御周期・音量の3項目を設定します。

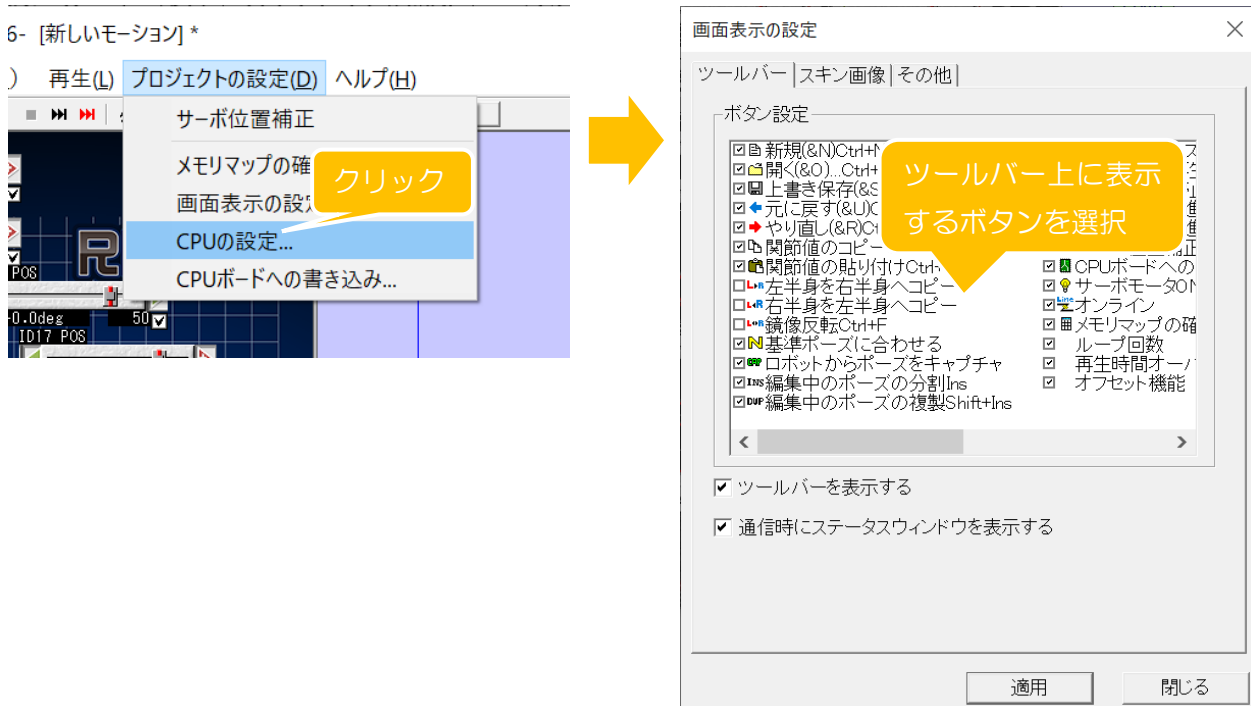


基本制御周期は、ポーズなどの1stepの時間に関する設定です。数値の補間やメモリマップの更新はここで設定した周期で行われます。デフォルトで16667マイクロ秒が設定されており、約60Hzで処理が実行されます。設定を変更すると、モータの通信速度などにも影響が発生するため、あまり短くするとモータに対して正しく数値が読み書きされない可能性があります。

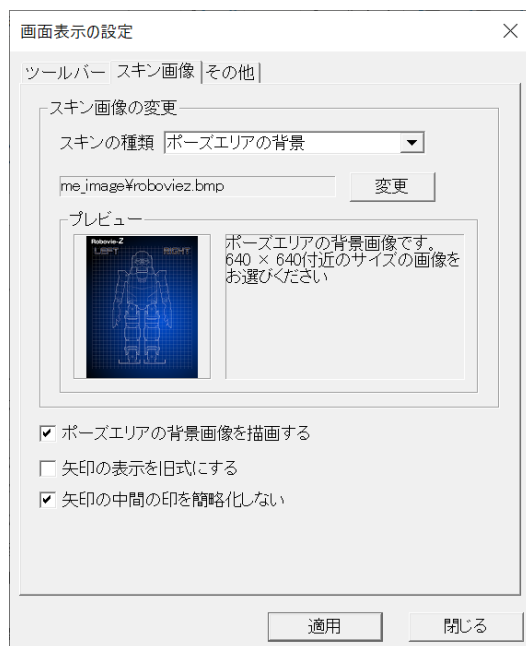
## 10-2. 画面表示の設定

メニューより「プロジェクトの設定」→「画面表示の設定」をクリックすると、ツールバー・ポーズ/モーションエリアのデザイン等表示に関する設定ダイアログを開きます。

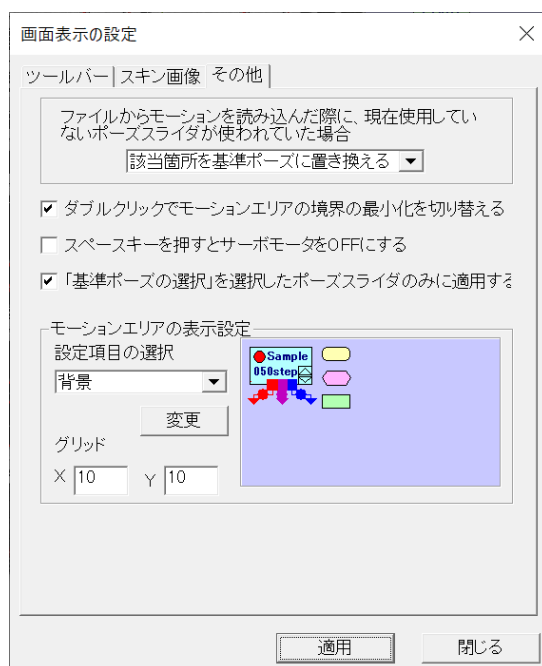
「ツールバー」のタブでは、ツールバーに表示するボタンの選択やステータスウィンドウの表示の有無などを設定します。



「スキン画像」のタブでは、ポーズエリアで使用する各種画像を設定します。試用できる画像は 24bit の WindowsBMP 形式です。また、変更する場合は極力元の画像に近いサイズの物を設定してください。



「その他」のタブでは、フィル読み込み時や各種動作に関する挙動の選択、モーションエリアのブロックの色の設定などを行います。



### 10-3. 平滑化の演算

メモリマップのアドレス 0x600~0x800 に設定された数値の平滑化機能はノイズや離散的な信号を平滑化する機能で、任意の入力値を与えると CPU ボード内で自動的に演算して出力を得られます。

計算式は以下のようになっています。N を大きくするほどより滑らかに平滑化されますが、入力値に対する出力値の反応が鈍くなります。

$$\text{出力値} = (\text{前回の出力値} * (N-1) + \text{入力値}) / N$$

平滑化機能は 32ch 存在し、複数の入力値を並列で演算できます。各 ch には 4 種類のパラメータが備わっており、それぞれ下記のように定義されています(数値のサイズはいずれも 2byte です)。

CU16\_IDXX\_TYPE    0:平滑化 OFF    1:平滑化 ON  
CU16\_IDXX\_ADRIN    入力値のメモリマップ上のアドレス  
CU16\_IDXX\_SM\_N    平滑化係数。大きいほどより平滑化を行う。  
CU16\_IDXX\_SM\_OUT    出力値

### 10-4. シリアル通信

CPU ボードは Raspberry Pi4 よりシリアル通信でメモリマップを読み書きできます。通信ポートの仕様は下記の通りです。

- ボーレート:115200bps
- データビット:8bit
- パリティビット:none
- ストップビット:1bit
- フロー制御:none

シリアル通信では下記のメッセージでメモリマップの読み書きが可能です。通信仕様として、CPU ボードは送信された文字をエコーバックします。また「#」の文字をプロンプトとして用います。メッセージの仕様として、数値は小文字の 16 進数表記になります。メモリマップはリトルエンディアンのため、2,4byte のデータを扱うときは注意してください。

#### ●数値の書き込み

**w AAAA BB CC...¥r**

「w」は書き込みコマンド開始を意味する記号です。「AAAA」は書き込み先のアドレスを記述します。「BB」、「CC」は実際に書き込むデータで、1byte(2 文字)ごとにスペースで区切って記述します。データは任意のサイズ分記述できます。各要素の間には必ず 1 つ以上スペースを入れてください。末尾に「¥r」を記述して送信すると CPU ボードがコマンドを実行します。

コマンドを実行すると、メモリマップ上の指定のアドレスから始まる連続した領域に、送信したデータが順番に書き込まれます。



## ●数値の読み込み

**r AAAA BB¥r**

「r」は読み込みコマンド開始を意味する記号です。「AAAA」は読込先のアドレスを記述します。「BB」は読み込むデータサイズを指定します。各要素の間には必ず1つ以上スペースを入れてください。末尾に「¥r」を記述して送信するとCPUボードがコマンドを実行します。

コマンドを実行すると、ローカルエコーに続いてCPUボードより下記の内容のメッセージが返信されます。

**# aaaa bb cc dd ee ff gg hh ii jj kk ll mm nn oo pp qq llllllllllllll**

「#」はプロンプトです。「aaaa」は、読み込み先のアドレスです。「bb」～「qq」は1byteごとのメモリマップの内容です。例では16個分を記述していますが、それよりも小さいサイズを読み込む場合は指定のサイズだけを返します。「llllllllllllll」はデータをアスキー文字で表示したものです。

17byte以上の数値を読み込む場合は、16byte区切りで表示が仕切り直されます。また、読み込み時にサイズを指定しない場合、指定アドレスから128byte分の数値が得られます。

## 10-5. ファームウェアの更新

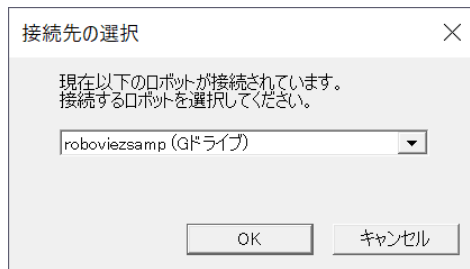
メニューの「ヘルプ」→「ファームウェアアップデート」より、CPUボードのファームウェアを更新することができます。



ファームウェアのバージョンは、CPUボードと通信してメモリマップよりリビジョンの数値を確認してください。

## 11. FAQ

Q：CPU ボードとの通信を開始するとき、以下のようなダイアログが表示される。



- A1：CPU ボードの初期化やデータ書き込みを行った後、RAM に設定が反映されていない可能性があります。CPU ボードを PC から取り外して電源を入れ直し、CPU ボードを立ち上げ直してから通信をおためしてください。
- A2：別のロボットの名前で初期化されている CPU ボードを使用している可能性があります。正しいプロジェクトをお持ちの場合は「キャンセル」をクリックして通信を中止し、改めて正しいプロジェクトを開いてそこから通信を行ってください。意図的に別のプロジェクトを CPU ボードに上書きする場合は「OK」をクリックして通信を開始し、モーションデータを書き込んでください。

Q：本ソフトウェアと CPU ボードが正しく通信しない

- A1：PC が CPU ボードを正しく認識できているかご確認ください。認識できていない場合は、通信ケーブルを接続しなおしたり、SD カードの接続を確認してください。
- A2：CPU ボードの各コネクタや端子に正しく機器を接続しているかご確認ください。コネクタを逆向きに接続している場合、CPU ボードが正しく動作しなくなる場合があります

Q：CPU ボードに接続したサーボモータが動かない

- A1：「[モータ ID の設定](#)」を参照して、サーボモータの ID が正しく設定されているか、また、正しいコネクタに正しい ID のモータを接続しているか、モータの ID が意図した番号になっているかについて、それぞれご確認ください。
- A2：CPU ボードにモータが動作するのに十分な電力が供給されているかご確認ください。
- A3：「[ポーズの補間器について](#)」を参照し、該当するポーズスライダの補間タイプが「0:補間しない」になっていないかご確認ください

**Q：CPU ボードに接続したコントローラが動作しない**

A1：コントローラの送受信機が正しくペアリングされているかご確認ください。ペアリング方法については、コントローラの付属説明書をご参照ください。

**Q：CPU ボードから音声が出力されない**

A1：CPU ボードにはスピーカを内蔵していません。CPU ボードにスピーカを接続していない場合は、音声出力用のスピーカを別途ご用意ください

A2：再生するファイルが CPU ボードの SD カードの正しいパスに存在しない可能性があります。本ソフトウェアよりモーションの書き込みを行うか、手動で正しいパスに音声ファイルを書き込んでください。

A3：音声ファイルの形式に CPU ボードが対応していない可能性があります。音声データの周波数・ビット数・モノラル設定の確認、及び wave ファイルのヘッダの不要な項目を削除するなどして改善するかお試しください。

**Q：CPU ボードの SD カードに対してファイルを読み書きできない**

A1：操作するファイルが CPU ボードの SD カードの正しいパスに存在しない可能性があります。本ソフトウェアよりモーションの書き込みを行うか、手動で正しいパスにファイルを書き込んでください。