ソフトウェア：DR-Visual Logic

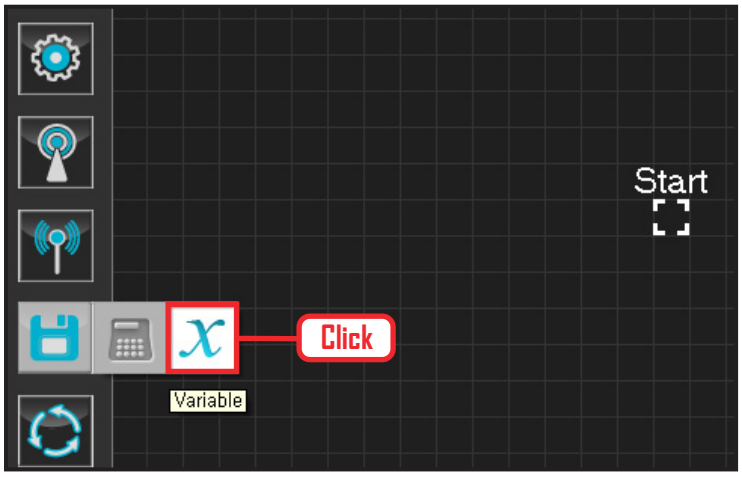
モジュール別のプログラミング：モーション(Motion) > Move

Moveの例

例の説明

MoveはDRCコントローラーに保存されたロボットのモーションを取得し、プログラムに適用します。ロボットのモーションも番号でインポートし、番号ごとの動きの名前はDR-SIMで確認することができます。 DR-SIMで編集された特定のモーションをロボットが無限に繰り返すプログラミングをします。ロボットの信頼性テストや展示などで活用度が高く、難易度が高いプログラムなので、一つずつ実行してください。

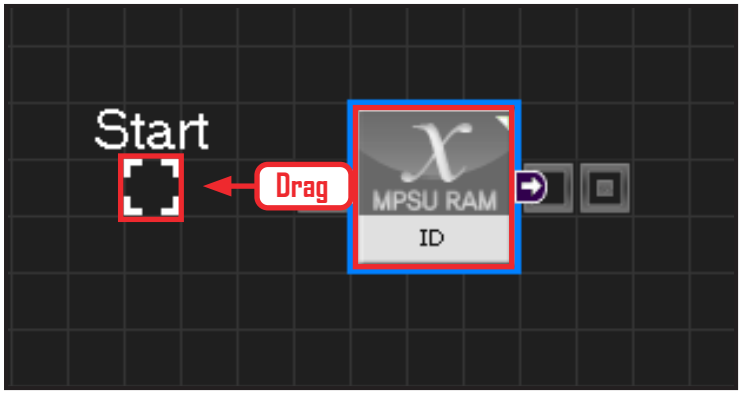
*この例で使用するモーションとモーション番号は基本提供するモーションとは異なります。 DR-SIMを使用して編集したモーションをDRCにダウンロードした場合を想定して作成された例です。関連モーションは www.hovis.co.kr/guide からダウンロードしてください。



01 変数の指定

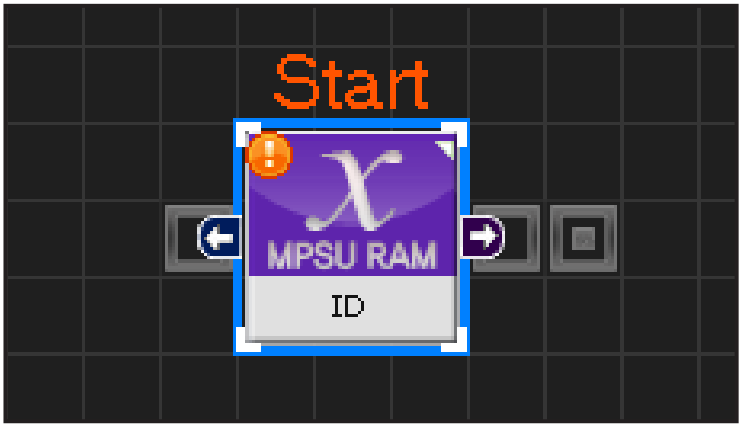
ロボットを動作させるということはロボットのサーボモーターを動作させるということです。サーボが自ら動くことができる状態の値を指定しなければなりません。

Data > Variableモジュールをクリックします。



02 開始

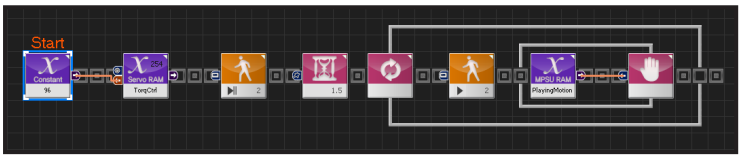
モジュールの左側の連結線をStartPointにドラッグして正確にドッキングさせます。



03 プログラミングの開始

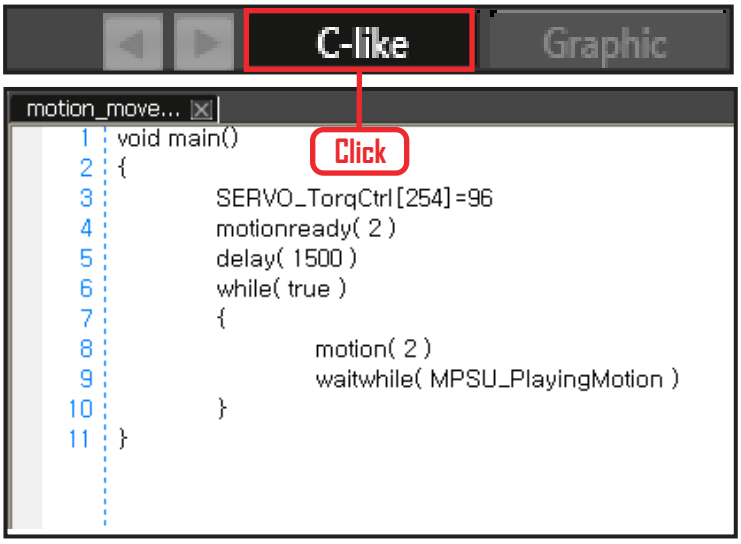
モジュールとStart Pointが正確にドッキングすると、左側のように有効化されたカラーイメージのモジュールに変わります。

これはプログラミングが開始されたという意味です。



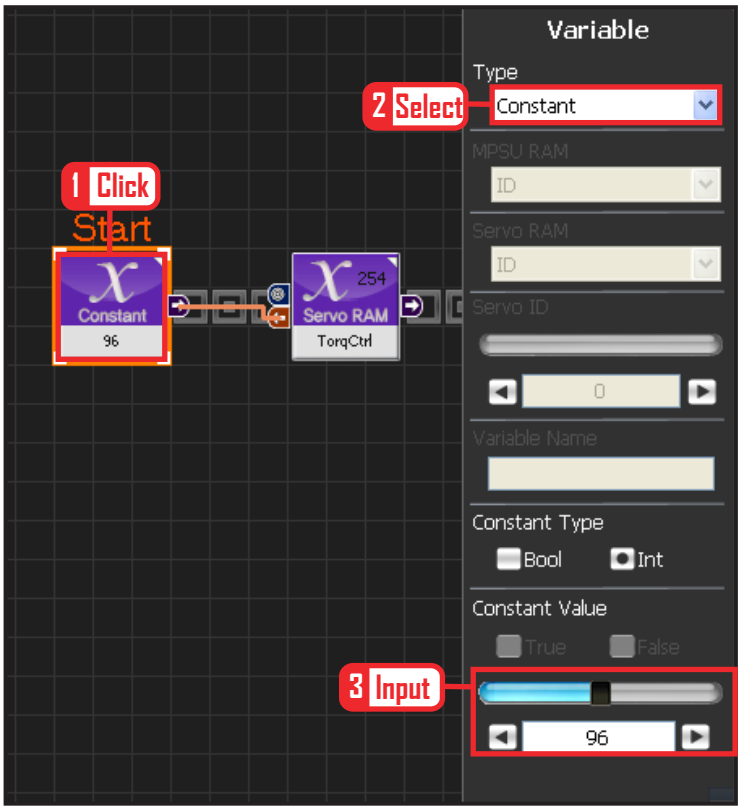
04 完全プログラミング

保存したモーションをインポートして繰り返させるプログラムです。Motion ready値に注意してください。



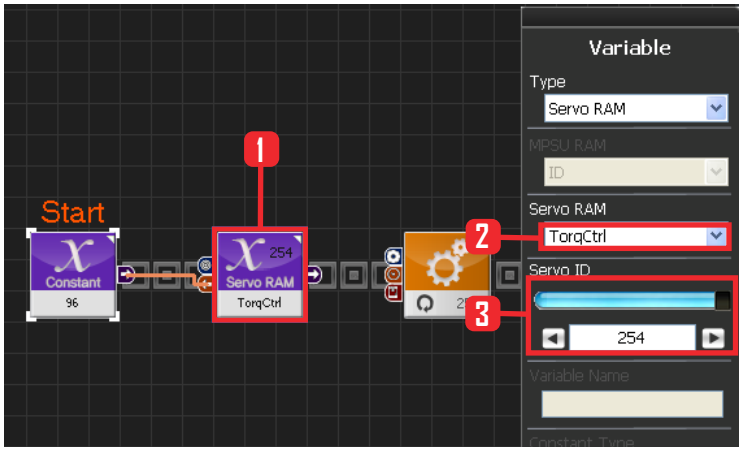
05 C-Like

右上のGraphicタブで、C-likeタブをクリックすると、左側のようなTaskプログラミングの画面が表示されます。
保存したモーションを利用して完全プログラミングをする画面です。
Cと似ている文法構造を持っているのでC文法の先行学習の効果もあります。
各モジュールごとにクリックすると、カーソルが今のプログラムの位置を示しながらモジュールごとにTextにどのように変換するかを確認することができます。



06 定数の設定

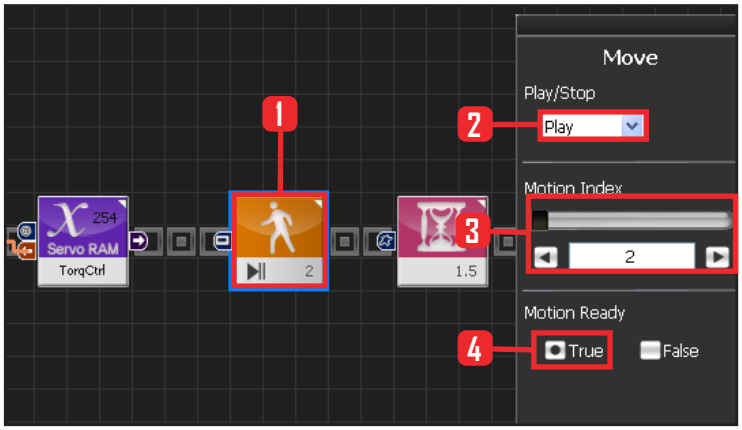
サーボモーターを自分で動かすことができる状態にする方法です。
Variable TypeでConstantを選択します。
プロパティー中のConstant　Value値を96に設定します。サーボのTorqControlレジスタに96(0x60)という値が入ると、サーボを動かすことができる状態になります。
その値はOutputコネクタを通して後ろのモジュールのトルク値に渡されます。



07 すべてのサーボに適用

前にもらった定数値96をすべてのサーボに適用する方法です。

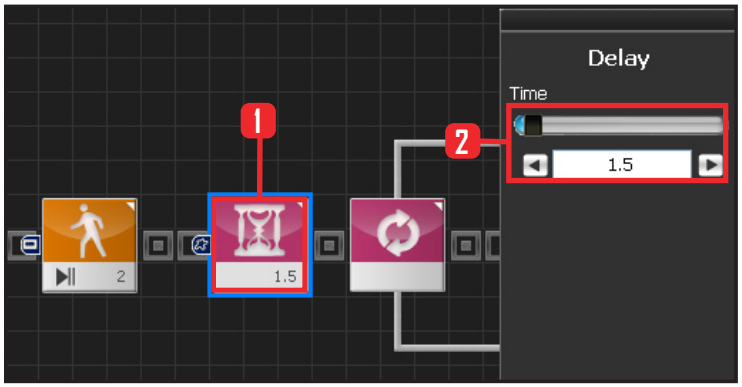
Variable > Type：Servo RAMを選択します。
Servo RAM：TorqCtrlを選択します。
Servo ID：254を選択します。 254は接続されているすべてのサーボに適用するという意味です。



08 Motion Ready

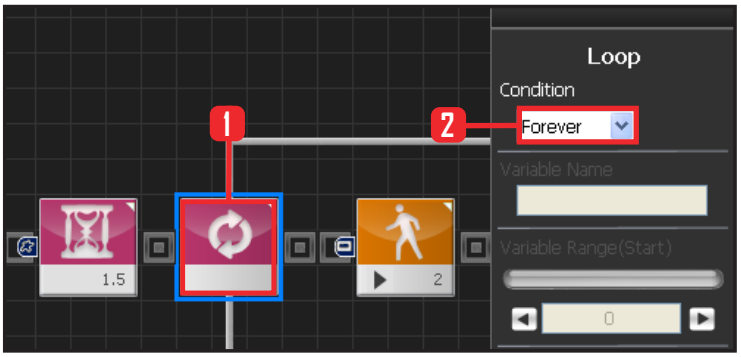
保存したモーションをインポートする時、ロボットの現在状態で、急に動きが変動する場合があります。現在状態とモーションのスタート状態があまりにも違うとモーターに無理がかかり、ユーザーに危険が及びます。そこでMotion Ready、つまり、モーションの動作準備時間を設定する必要があります。

Motion > Moveを選択します。
Play/Stop：Playを選択します。
Motion Index：2を選択します。 2番のモーションを持ってくるという意味です。ちなみにこのプログラムの2番モーションは座った後に立ち上がるモーションです。必ず2番のモーションを選択する必要はなくて、ユーザーが実行したいモーションの番号を入力してください。
Motion Ready：Trueを選択します。 Trueを選択すると、動作させようとするモーションの初期状態に徐々に移動します。



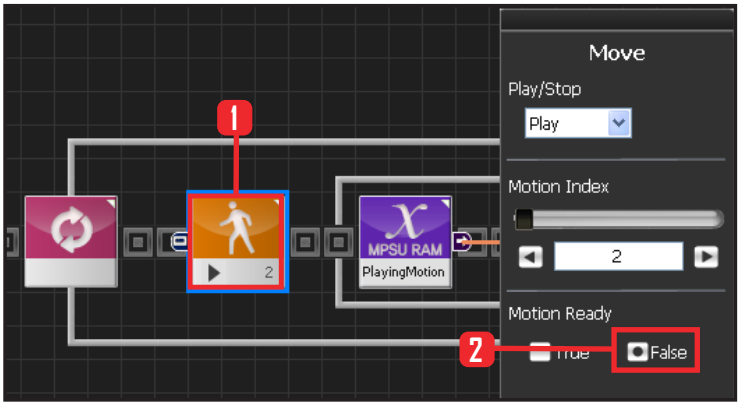
09 Delay

Motion Ready 動作が終了する前にモーションが進行することを防止するためにDelay値を1.5秒に設定します。



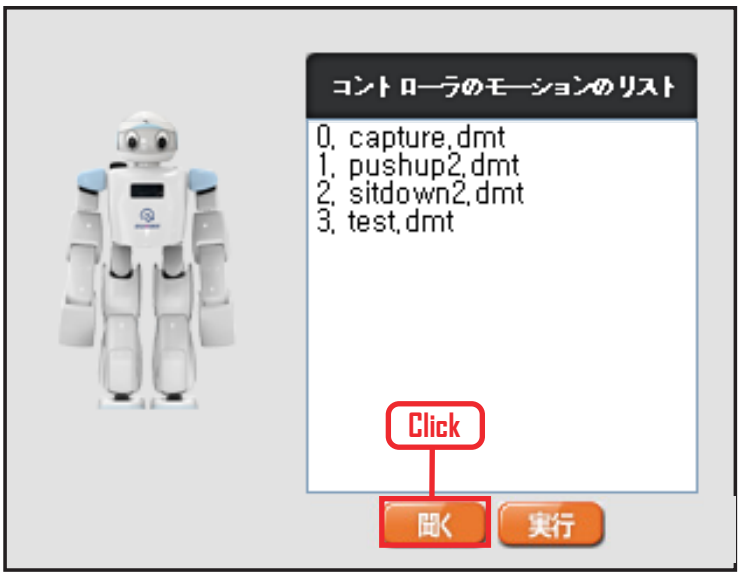
10 繰り返し

Forever無限ループを設定します。



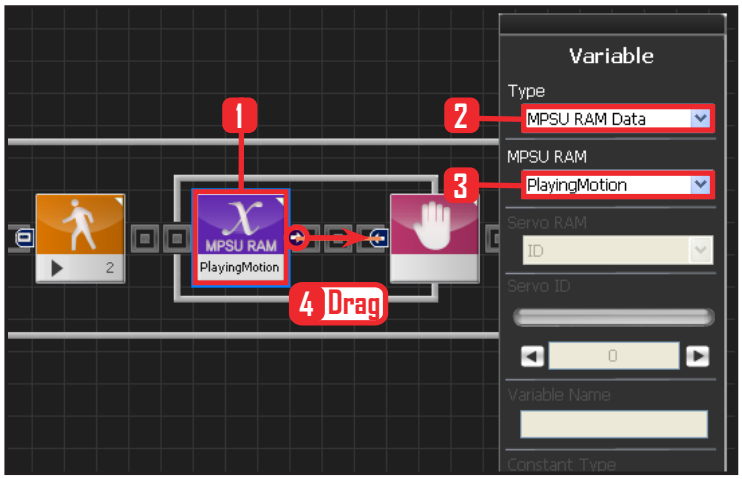
11 モーション動作

Motion　Ready値をFalseに設定すると、モーションを最初から最後まで動作させるという意味になります。
Motion Ready:Falseを選択します。



参照:モーションを見る

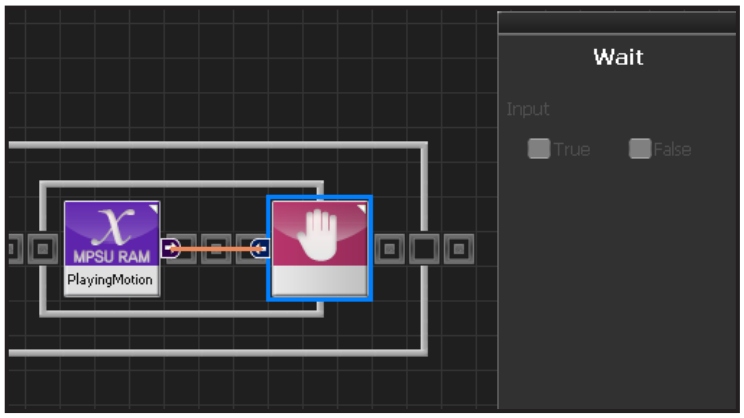
DR-SIMプログラムでロボットと接続して、ロボット設定をクリックすると、現在のコントローラーにあるロボットのモーションを確認できます。
現在、2番のモーションは両腕を広げて座った後、立ち上がるモーションです。



12 モーション動作の確認

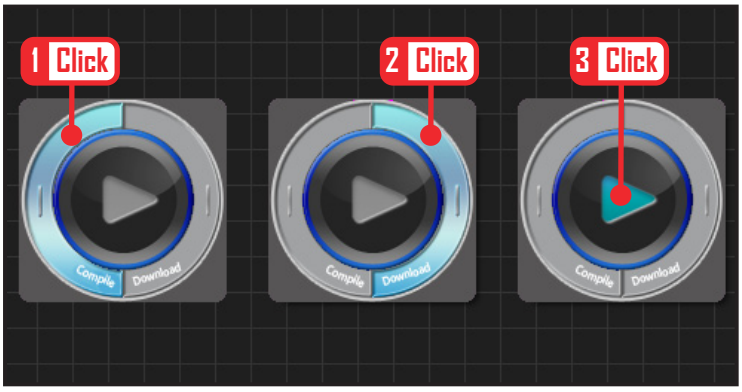
Loopは継続的な繰り返しを意味します。 Move命令を下してから、実際のモーションが実行され、完了するまでには時間がかかるので、loop内にMoveモジュールを一つだけ入れて実行すると、モーションが既に実行中であるにもかかわらずloopを続けてモーションを実行する命令を繰り返すことになります。
この場合、Moveモジュールを実行した回数と実際のモーションを実行する回数が異なります。
したがって実行したモーションが終わるまで待ってから再びloopの最初に戻す方がより正確です。
Variable > MPSU RAM Datalに入れば、Playing Motion があります。 Playing Motionはロボットのモーションを実行しているかを確認する変数です。そのPlaying MotionにWaitをかければ、ロボットの動作が終わるまでLoopを待ちます。

Data > Variableモジュールを選択します。
Type:MPSU RAM Dataを選択します。
MPSU RAM:Playing Motionを選択します。
Outputコネクタの値の後ろのWaitモジュールに接続します。
モーションが終わるまで待つことを意味します。



13 Wait

モーションが終わるまで待ちます。
モーションが終わればまた最初に戻ってモーションを繰り返します。



14 コンパイル、ダウンロード、実行

左をクリックしてコンパイルします。エラーがなければ右クリックして、ロボットにダウンロードします。ダウンロードが完了すると、中央の矢印の実行ボタンを押して、ロボットで実行させます。

1



2



15 ロボット動作

ロボットが座って立ち上がるモーションを繰り返し実行します。