ソフトウェア：DR-Visual Logic

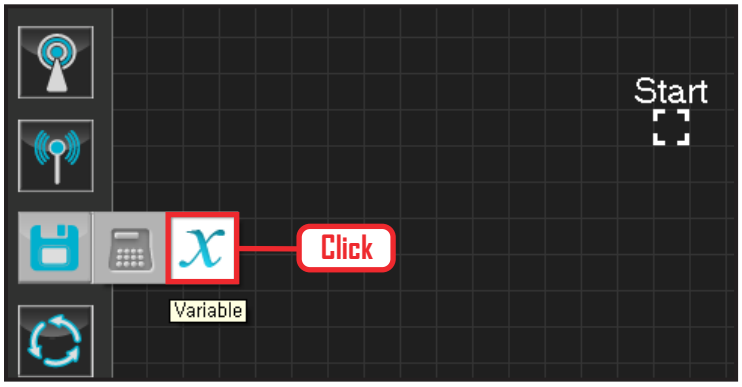
モジュール別プログラミング：センサー（Sensor）＞ Analog Distance

Analog Distance Sensorの例

例の説明

アナログ距離センサを用いて壁に近づくと左折して壁を回避するプログラムです

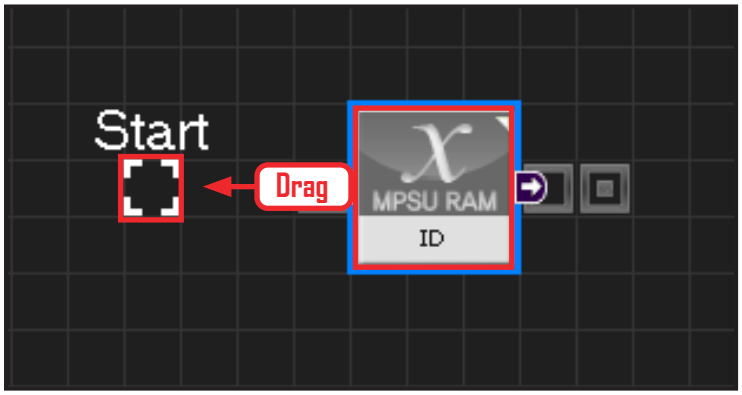
距離センサはアナログセンサとデジタルセンサがあります。デジタルセンサは一定の距離（10cm）を基準に内側と外側の両方の領域のみを検出するのに対しアナログセンサは距離別に距離（6～80cm）を検出することができます。この例を実行するときはADCポート1番（左）にPSDセンサを装着する必要があります。



01 変数の指定

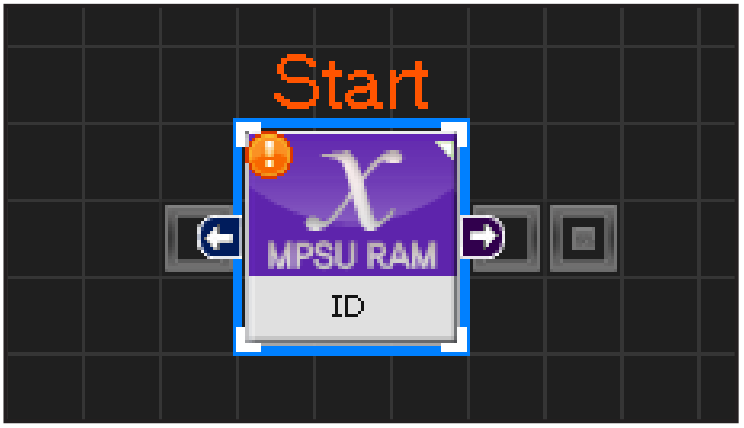
ロボットを動作させるということはロボットのサーボモーターを動作させるということです。サーボが自ら動くことができる状態の値を指定しなければなりません。

Data ＞ Variableモジュールをクリックします。



02 開始

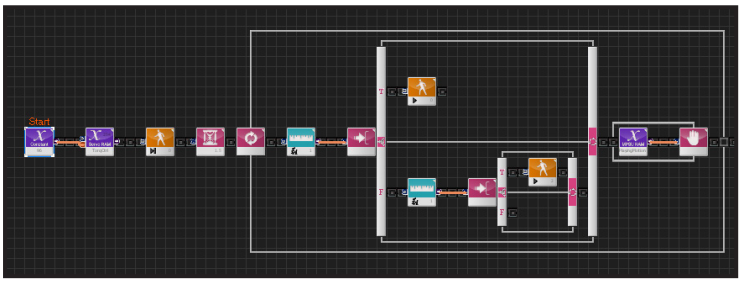
モジュールの左側の連結線をStartPointにドラッグして正確にドッキングさせます。



03 プログラミングの開始

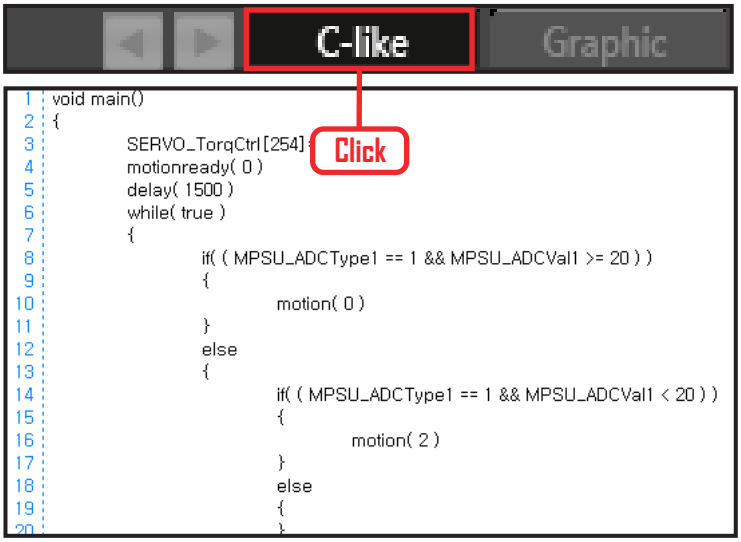
モジュールとStart Pointが正確にドッキングすると、左側のように有効化されたカラーイメージのモジュールに変わります。

これはプログラミングが開始されたという意味です。



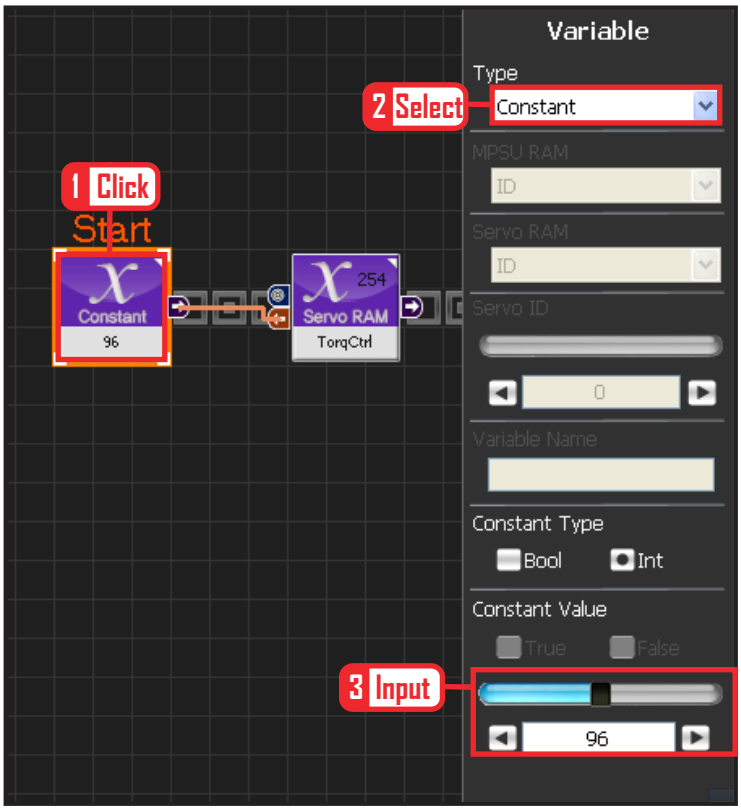
04 完全プログラミング

アナログ距離センサを利用した完全なプログラムです。



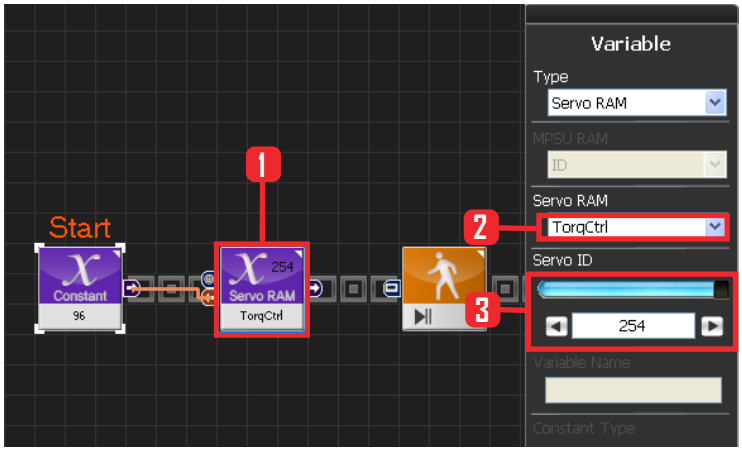
05 C-Like

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。
アナログ距離センサを利用して完全な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

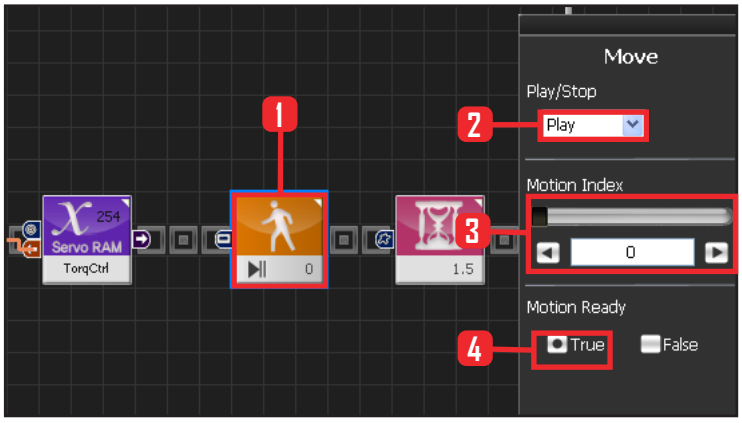
サーボモーターを自分で動かすことができる状態にする方法です。
Variable TypeでConstantを選択します。
プロパティー中のConstant　Value値を96に設定します。サーボのTorqControlレジスタに96(0x60)という値が入ると、サーボを動かすことができる状態になります。
その値はOutputコネクタを通して後ろのモジュールのトルク値に渡されます。



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。

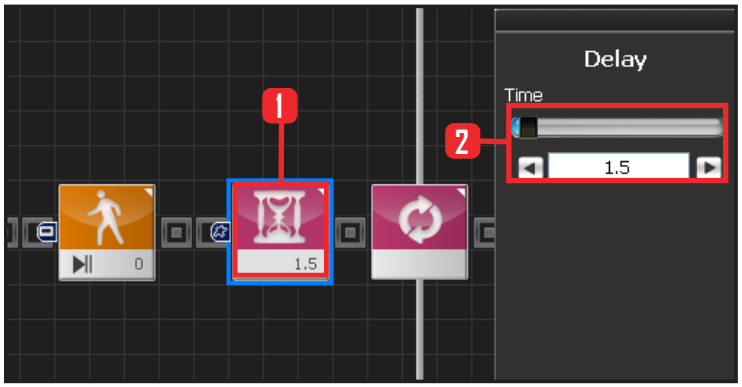
Variable > Type：Servo RAMを選択します。
Servo RAM：TorqCtrlを選択します。
Servo ID：254を選択します。 254は接続されているすべてのサーボに適用するという意味です。



08 モーション動作の準備

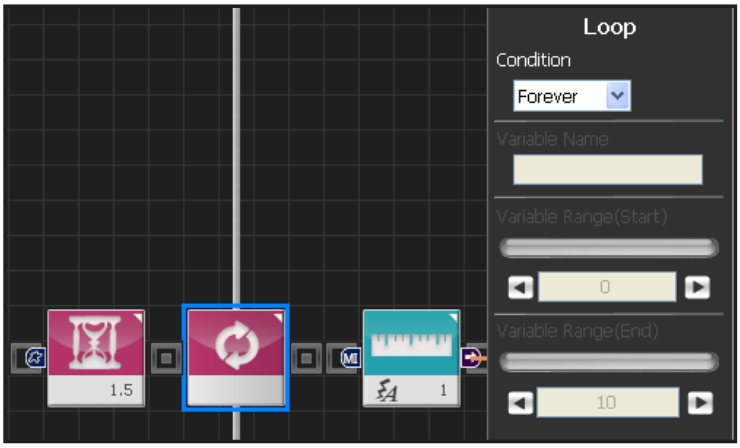
モーション動作のための準備過程です。ロボットのモーションが急に変動した場合、ロボットに無理がかかります。したがって、現在実行しようとするロボットの最初のモーションからゆっくりと移動させます。
Motion　ReadyがTrueの場合、モーションの最初のシーンを準備し、Falseの場合モーションが動作します。

Motion > Moveモジュールを選択します。
Play/ Stop：Playを選択します。
Motion Index：0番を選択します。0番は前進しながら歩くモーションです。
Motion Ready：Trueを選択します。



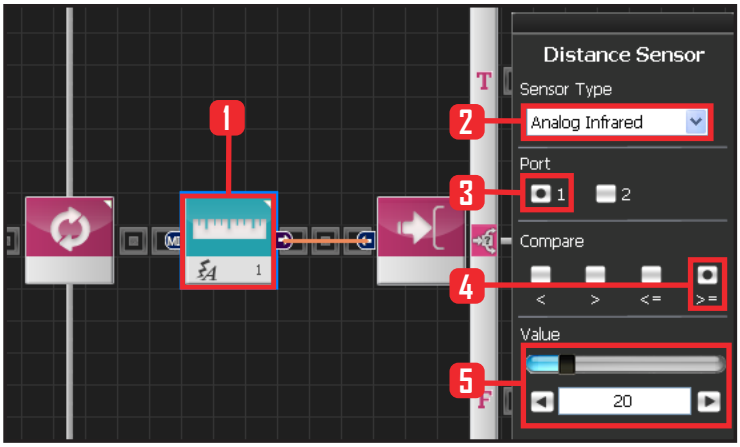
09 Delay

MotionReady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 反復

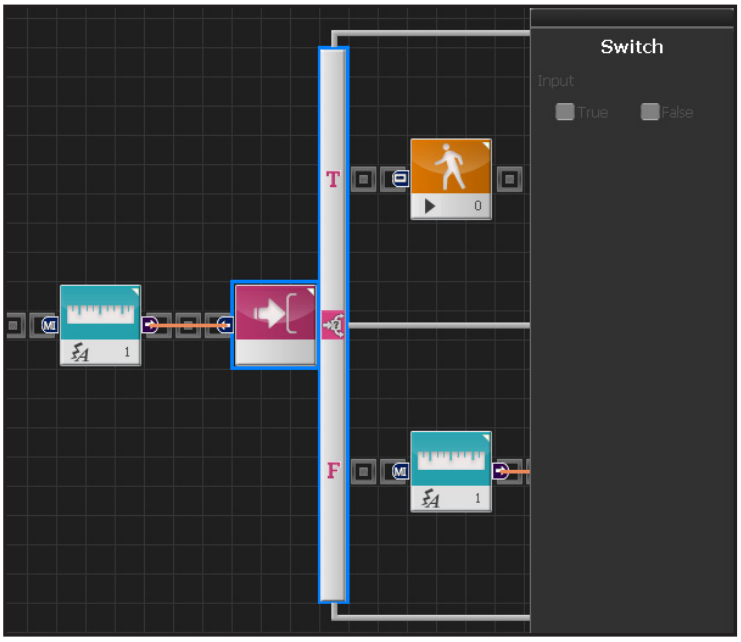
Forever無限反復を設定します。



11 アナログ距離センサの設定

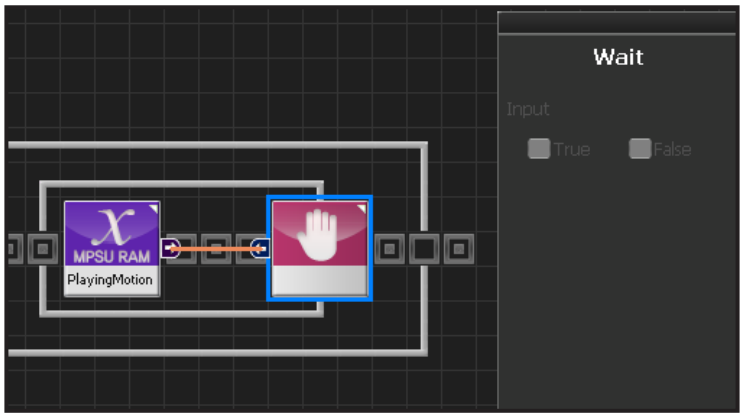
20cm基準に設定します。

Sensor > Distance Sensorモジュールを選択します。
Sensor Type：Analog Infraredを選択します。
Port：1番を選択します。
Compare：>=　を選択します。基準よりも同じか高い時はTrueです。
Value：20を選択します。20cmを意味します。



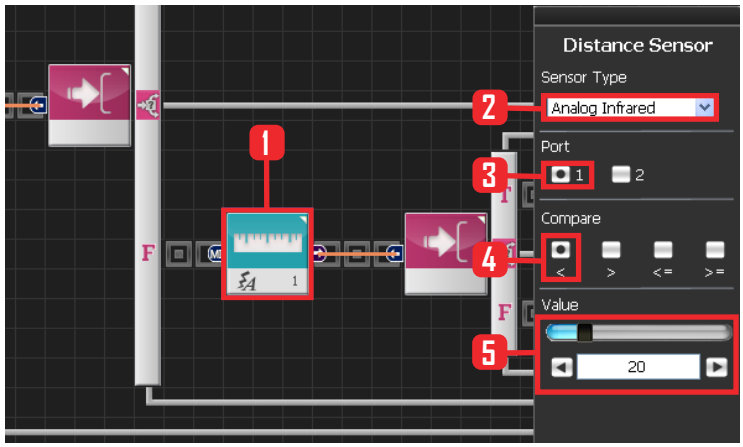
12 If 分岐文

Trueの場合は前進し、Falseの場合は次の条件文に入ります。



13 前進する

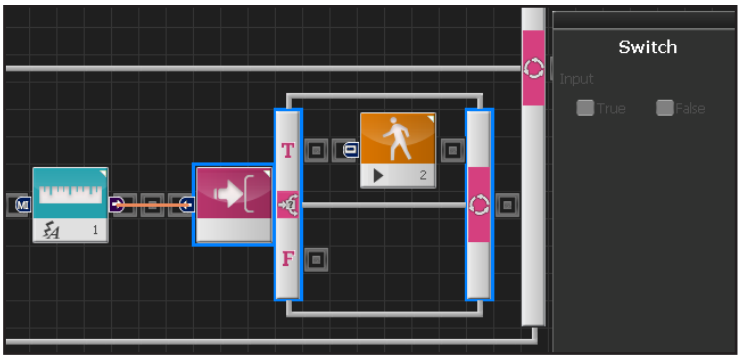
20cmより遠い距離なので、ロボットが前進させます。
Motion　Ready値をFalseに選択すると、ロボットは前進モーションを実行します。



14 壁に接近する時の動作

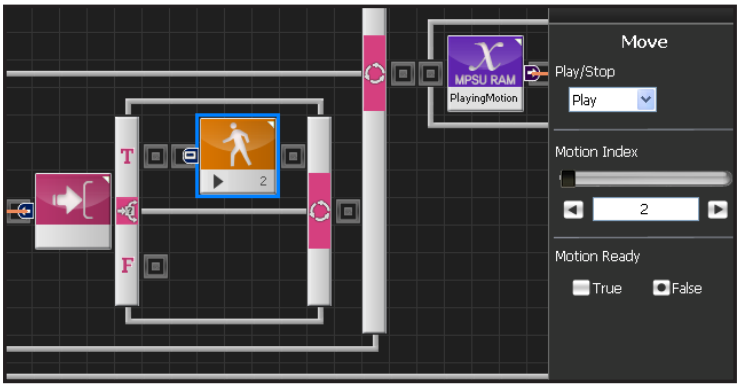
ロボットが壁の20cm以内に接近した時はロボットを右折させるプログラムです。

Sensor＞Distance Sensorモジュールを選択します。
Sensor Type：Digital Infraredを選択します。
Port：1番を選択します。
Compare：＜を選択します。基準より低い場合にTrueを返します。
Value：20を選択します。20cmを意味します。



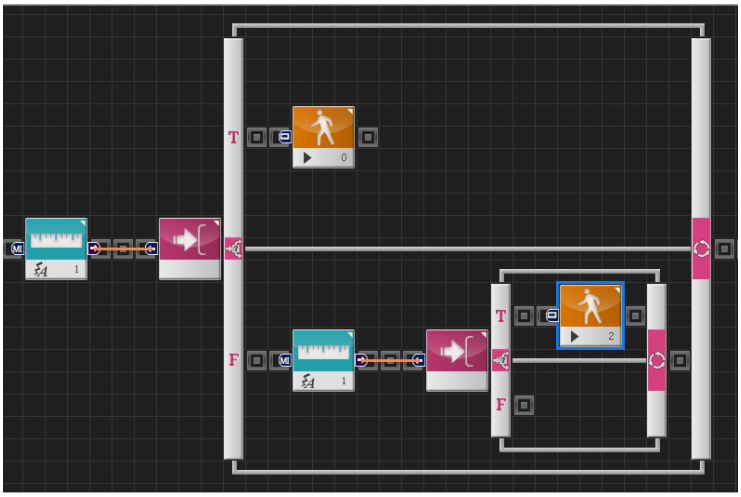
15 If 分岐文

20cmより近いと、Trueの中の文を実行します。



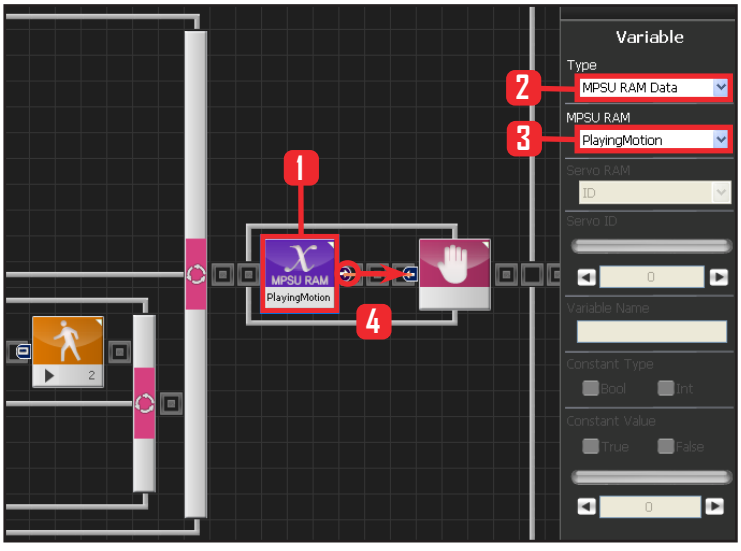
16 左折させる

モーション2番は左折モーションです。
Motion　Readyの値がFalseの場合は左折モーションを実行します。



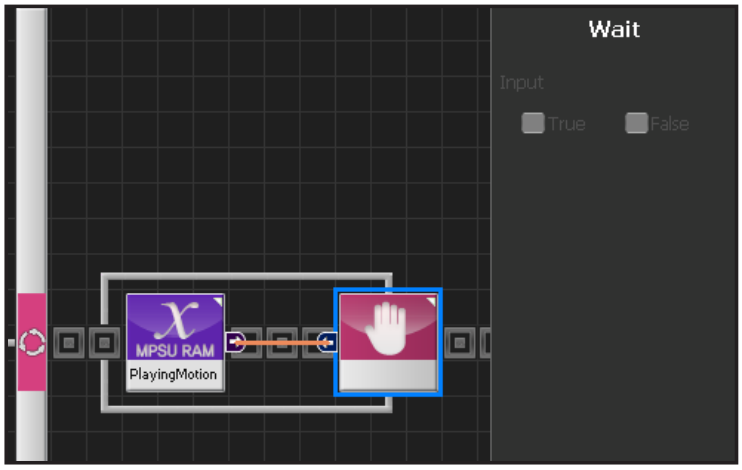
17 距離別の動作

20cmより遠い場合はロボットが前に前進し、20cm未満の場合はロボットが左折します。



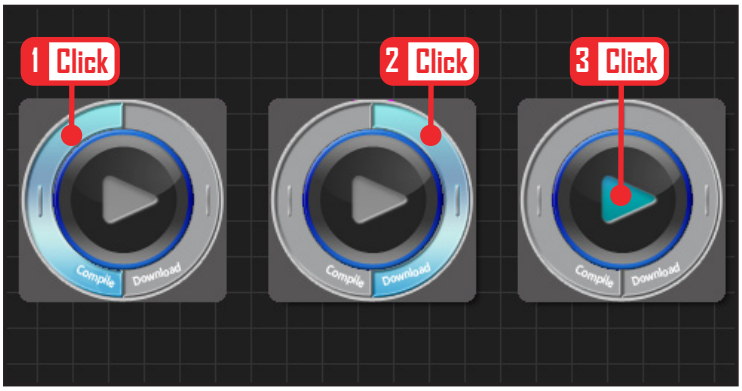
18 モーション動作の確認

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、完了するまでは時間がかかるので、loop内にMoveモジュールの一つだけを入れて実行すると、モーションが既に実行中央にもかかわらず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確実です。
Variable > MPSU RAM Dataに入ると、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。



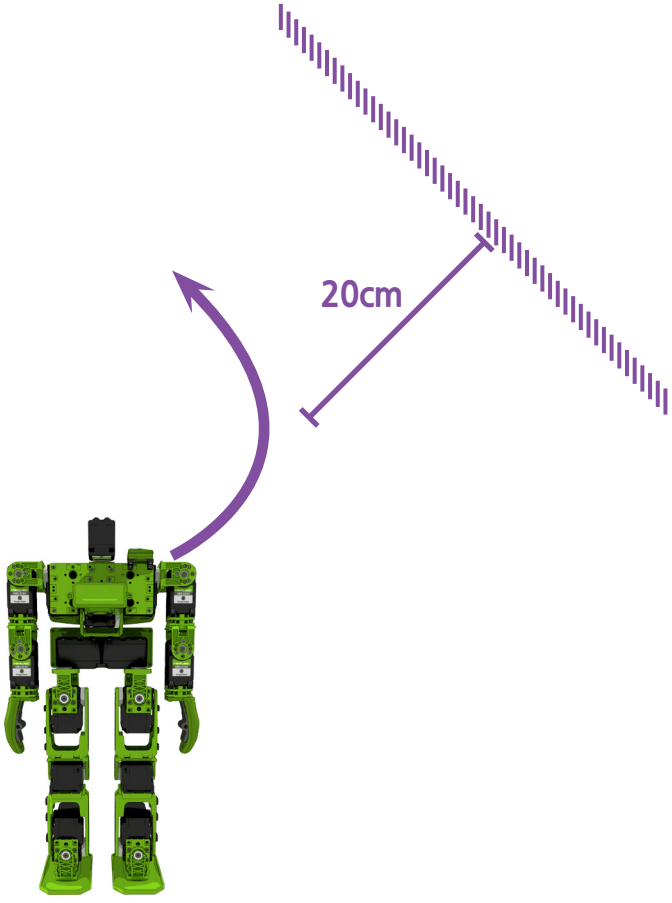
19 Wait

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



20 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。



21 ロボット動作

ロボットは壁に向かって前進し、20cm内に近づくと、左折します。