ソフトウェア：DR-Visual Logic

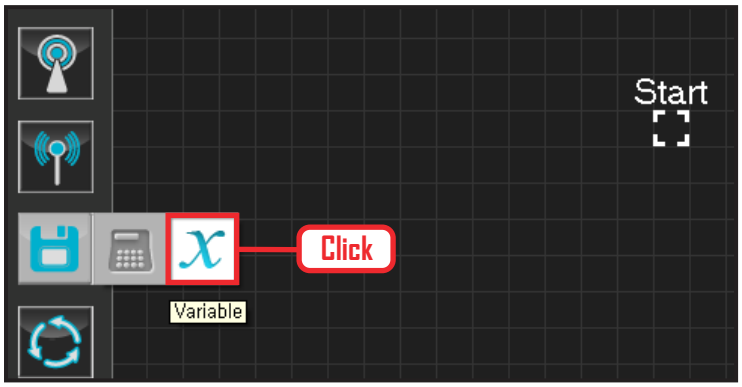
モジュール別プログラミング：センサー（Sensor）> Digtal Distance

Digital Distance Sensorの例

例の説明

アナログ距離センサはすべての距離測定が可能ですが、デジタル距離センサは特定の距離を基準に近いか遠いかだけを判断します。車輪付きのロボットは壁を検出するのではなく、崖の検出に多く使用されています。ヒューマノイドでは足が動くので、崖の検出ではなく壁を検出するプログラミングをします。アナログ距離センサと比較してください。

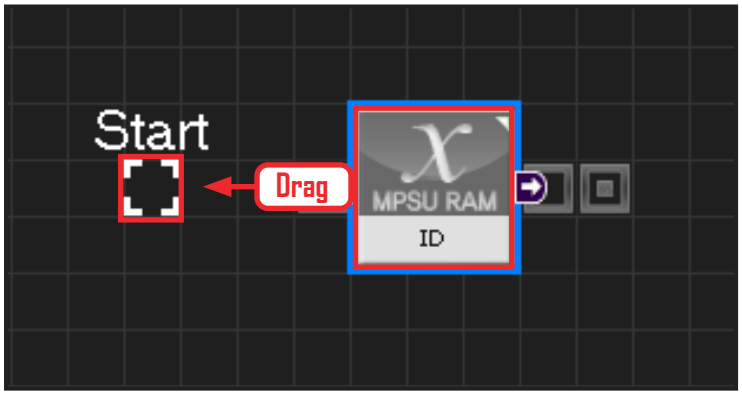
壁が一定距離に近づくと、後ずさりしてから方向を変えて前進するプログラムです。この例を実行する場合はADCポート1番（左）に、デジタル距離センサを装着する必要があります。



01 変数の指定

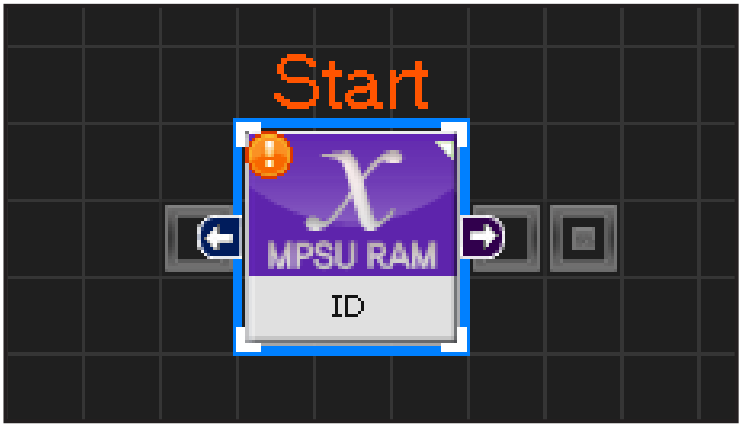
ロボットを動作させるということはロボットのサーボモーターを動作させるということです。サーボが自ら動くことができる状態の値を指定しなければなりません。

Data > Variableモジュールをクリックします。



02 開始

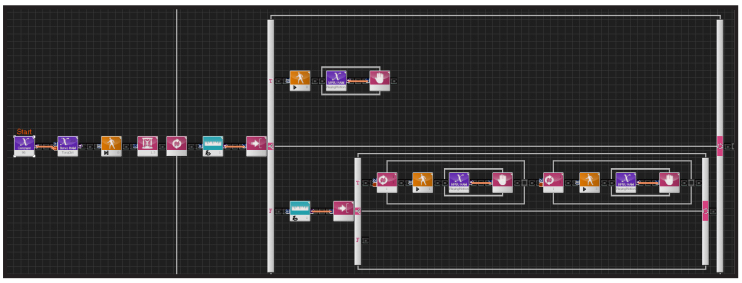
モジュールの左側の連結線をStartPointにドラッグして正確にドッキングさせます。



03 プログラミングの開始

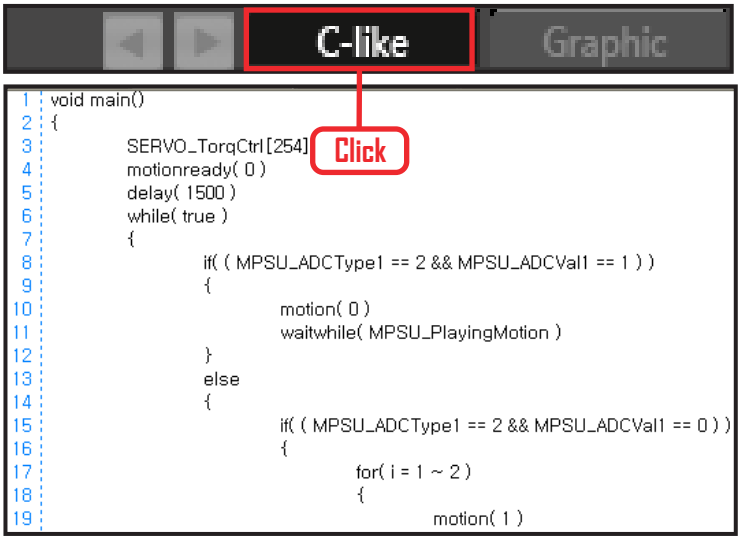
モジュールとStart Pointが正確にドッキングすると、左側のように有効化されたカラーイメージのモジュールに変わります。

これはプログラミングが開始されたという意味です。



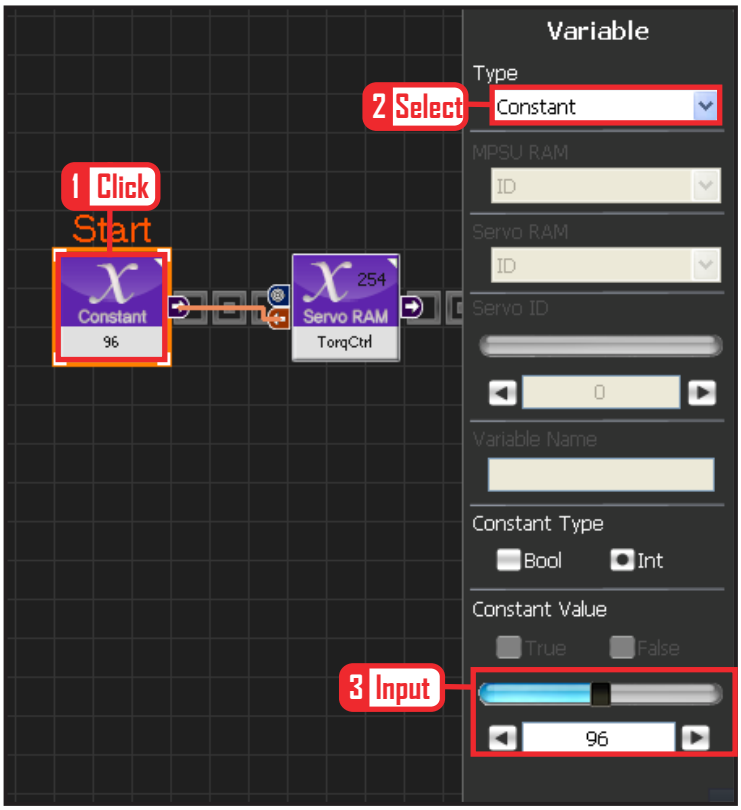
04 完全プログラミング

デジタル距離センサを利用した完全なプログラムです。



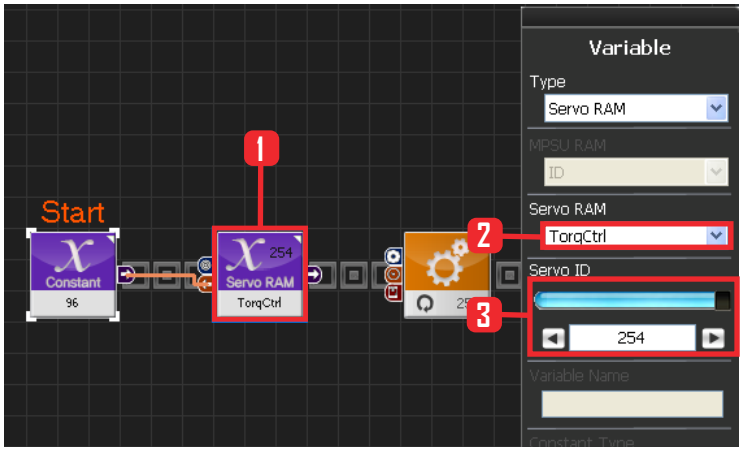
05 C-Like

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。
デジタル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

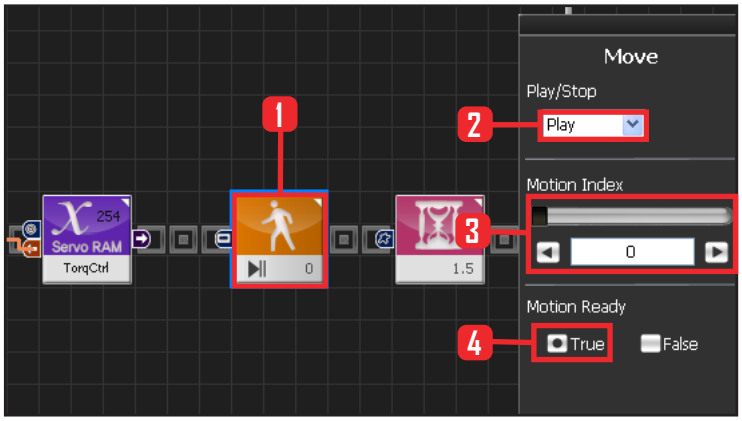
サーボモーターを自分で動かすことができる状態にする方法です。
Variable TypeでConstantを選択します。
プロパティー中のConstant　Value値を96に設定します。サーボのTorqControlレジスタに96(0x60)という値が入ると、サーボを動かせることができる状態になります。
その値はOutputコネクタを通して後ろのモジュールのトルク値に渡されます。



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。

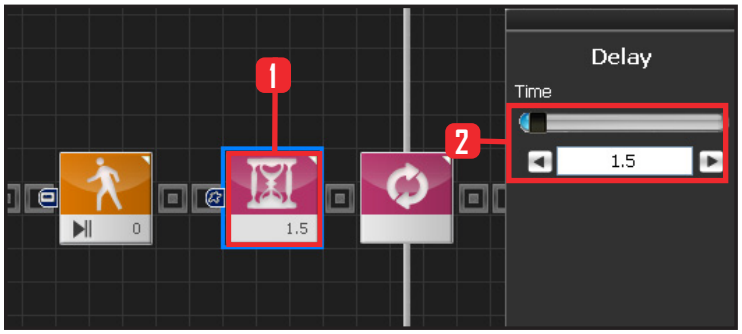
Variable > Type：Servo RAMを選択します。
Servo RAM：TorqCtrlを選択します。
Servo ID：254を選択します。 254は接続されているすべてのサーボに適用するという意味です。



08 モーション動作の準備

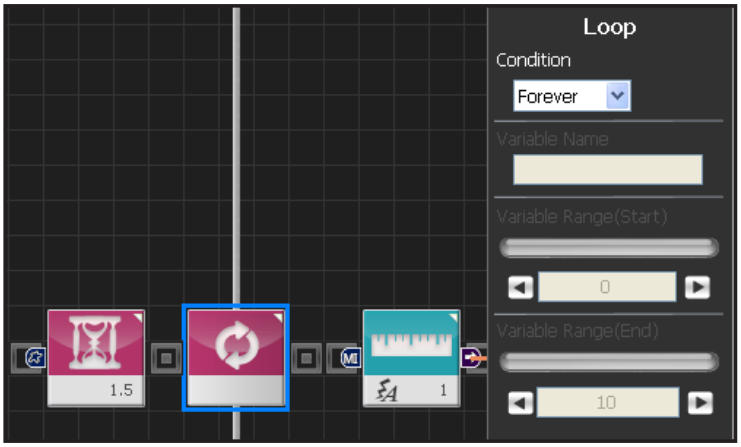
モーション動作のための準備過程です。ロボットのモーションが急に変動した場合、ロボットに無理がかかります。したがって、現在実行しようとするロボットの最初のモーションからゆっくりと移動させます。
Motion　ReadyがTrueの場合、モーションの最初のシーンを準備し、Falseの場合モーションが動作します。

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。



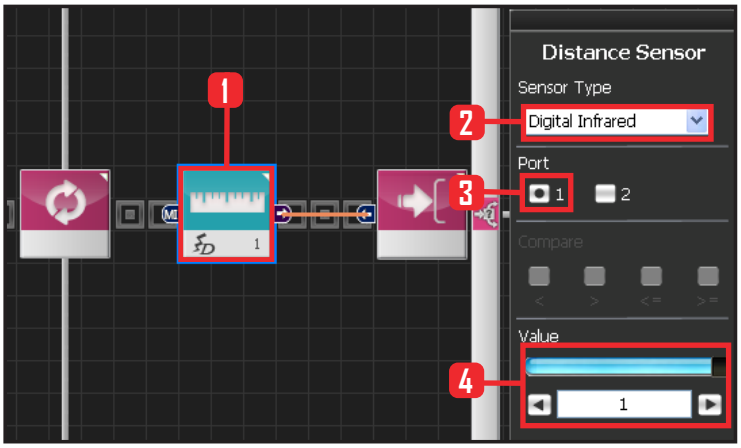
09 Delay

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

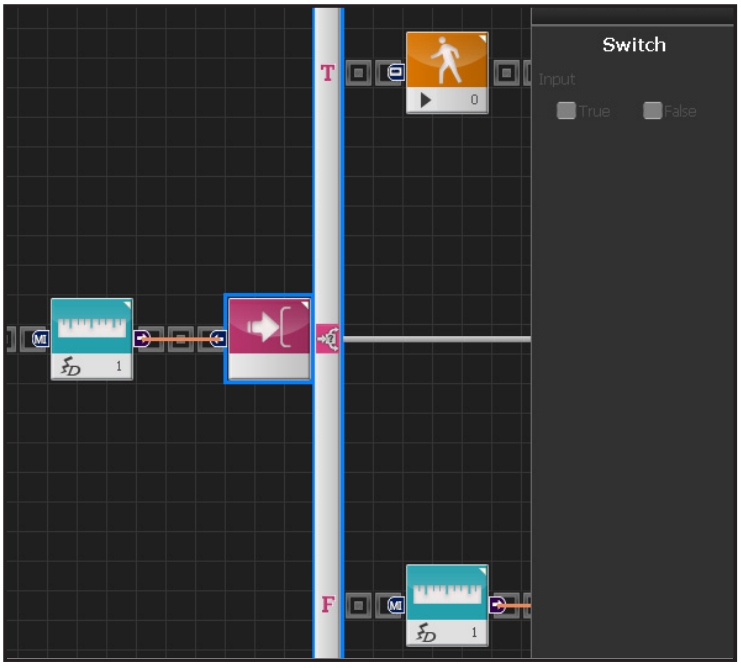
Forever無限反復を設定します。



11 デジタル距離センサの設定

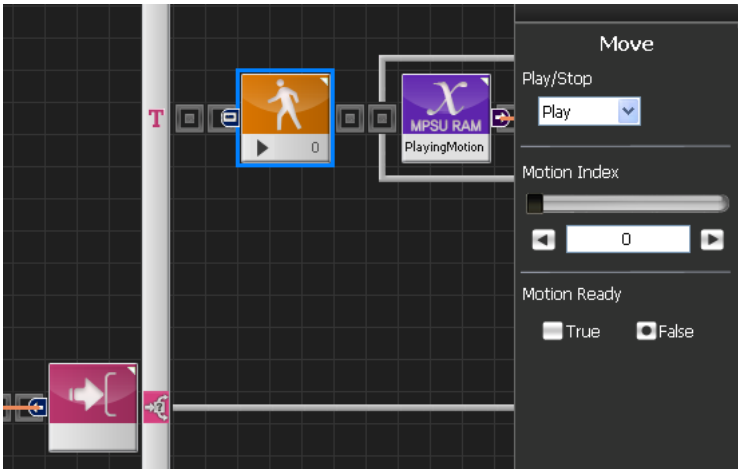
デジタル距離センサは製品ごとに測定距離が異なります。ここでは20cm単位で設定されています。

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。
Value:1を選択します。 10cm以上離れていることを意味します。



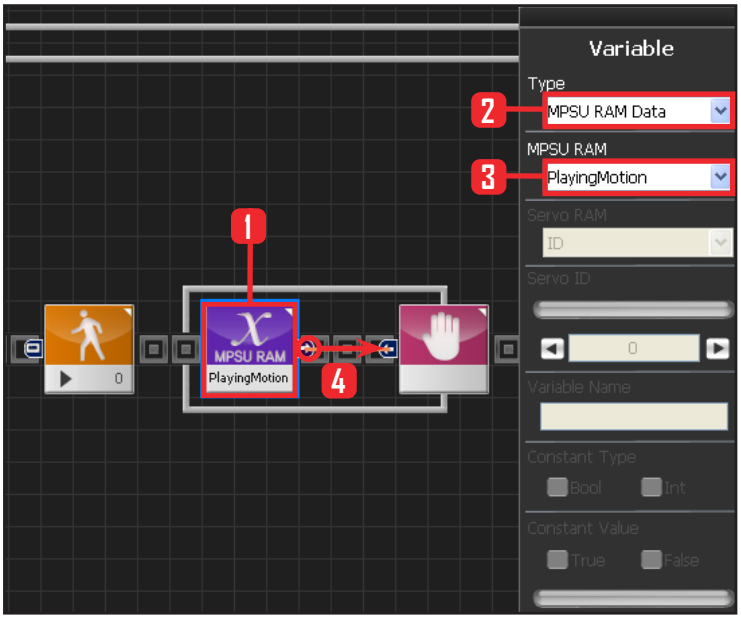
12 If 分岐文

Trueの場合は前進し、Falseの場合は次の条件文に入ります。



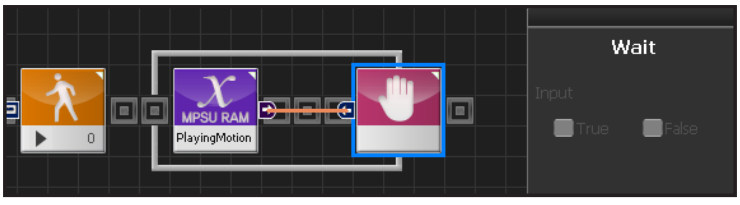
13 前進する

10cmより遠い距離なので、ロボットが前進します。
Motion　Ready値でFalseを選択すると、ロボットは前進モーションを実行します。



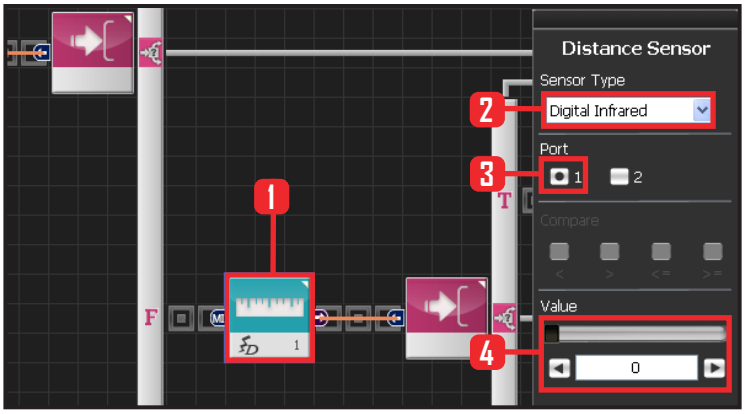
14 モーション動作の確認

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。



15 Wait

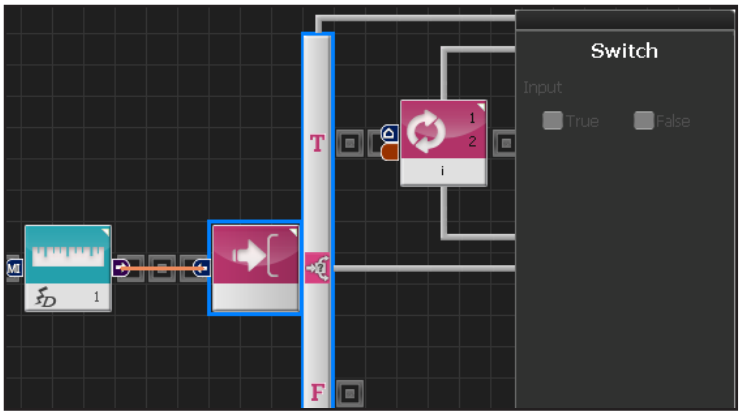
モーションが終わるまで待ちます。
モーションが終われればまた最初に戻ってモーションを繰り返します。



16 壁に接近する時の動作

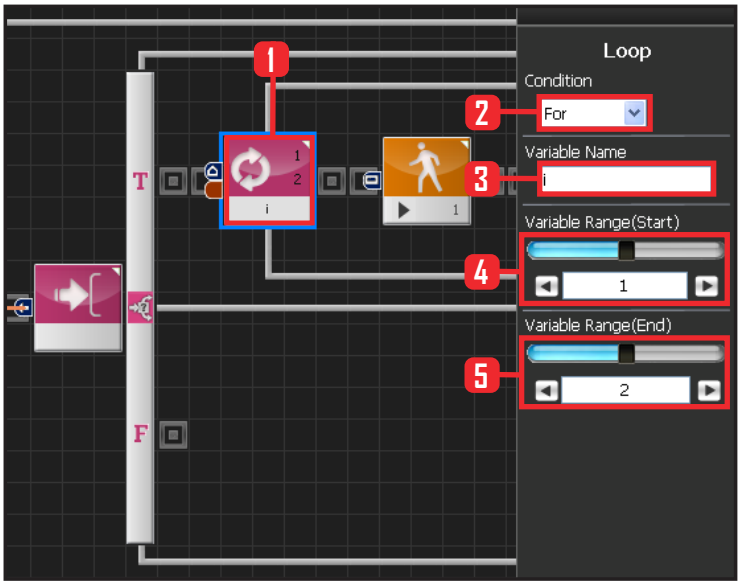
ロボットが壁の10cm以内に接近した時はロボットを後進歩行させ、その後、右折させるプログラムです。

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。
Value : 0を選択します。 10cm内に入ったことを意味します。



17 If 分岐文

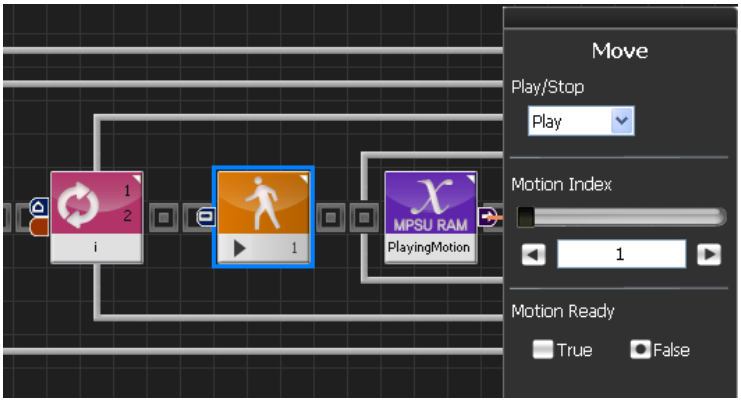
10cmより近いと、Trueの中の文を実行します。



18 For 反復

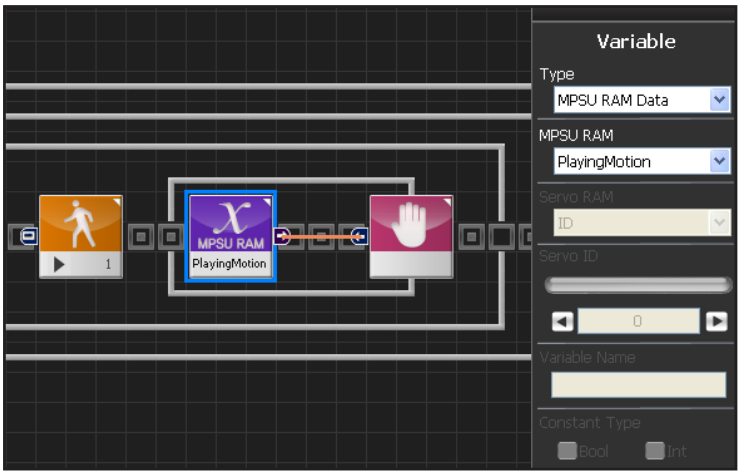
ロボットのモーションは同じ動作を繰り返して、必要なモーションまでの動作を取ることができます。
モーション1番はロボットが後進するモーションです。後進は左足と右足を一度だけ後ろに移動させることです。
そのモーションに必要な数だけ繰り返しを与えることができるFor文を用いて、移動したい位置まで移動させることができます。

Flow > Loopモジュールを選択します。
Condition : Forを選択します。
Valuable Nameに ｉ を入力します。
Valuable Range(Start)に1を入力します。
Valuable Range(End)に2を入力します。
モーションを2回実行させるという意味です。



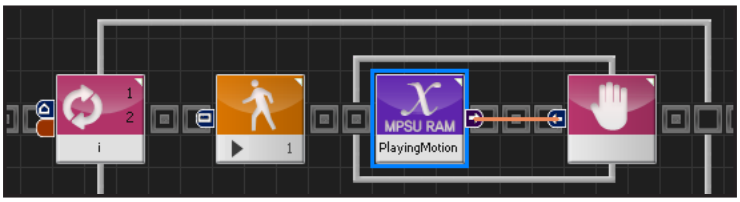
19 後ろにしりぞく

モーション1番はロボットが後進するモーションです。
Falseを選択すると、ロボットが後進モーションを実行します。



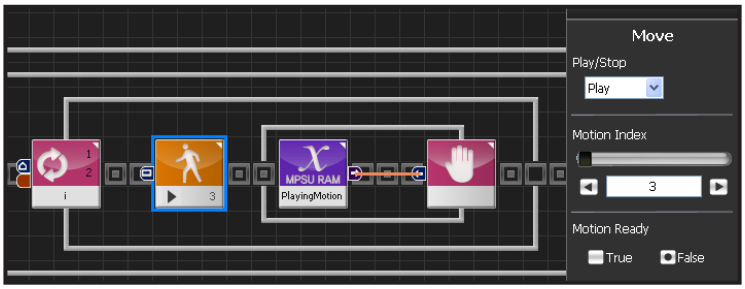
20 動作感知

Playing Motionでロボットの動作を検出し、終われば またfor文の最初の部分に戻ります。



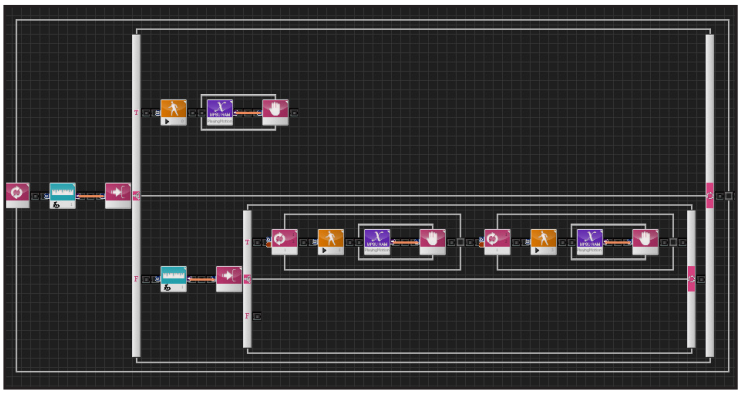
21 二回反復の後進

ロボットを二回後進させるプログラムです。



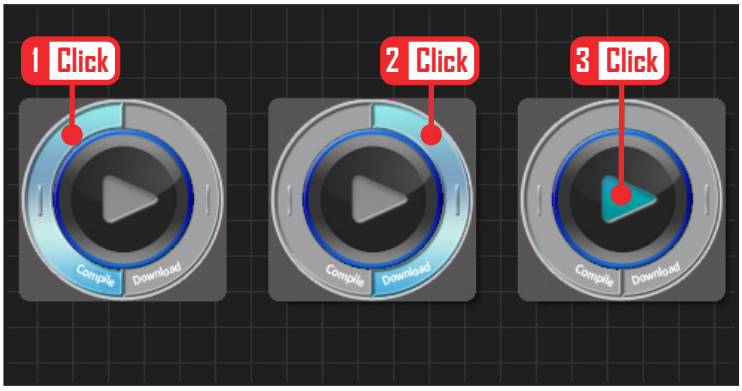
22 右折させる

3番モーションはその場で左折する動作です。右折動作も同じようにFor文を使って調整することができます。Motionで3番を選択して、For文を1～3まで設定して、上記のようにプログラムします。



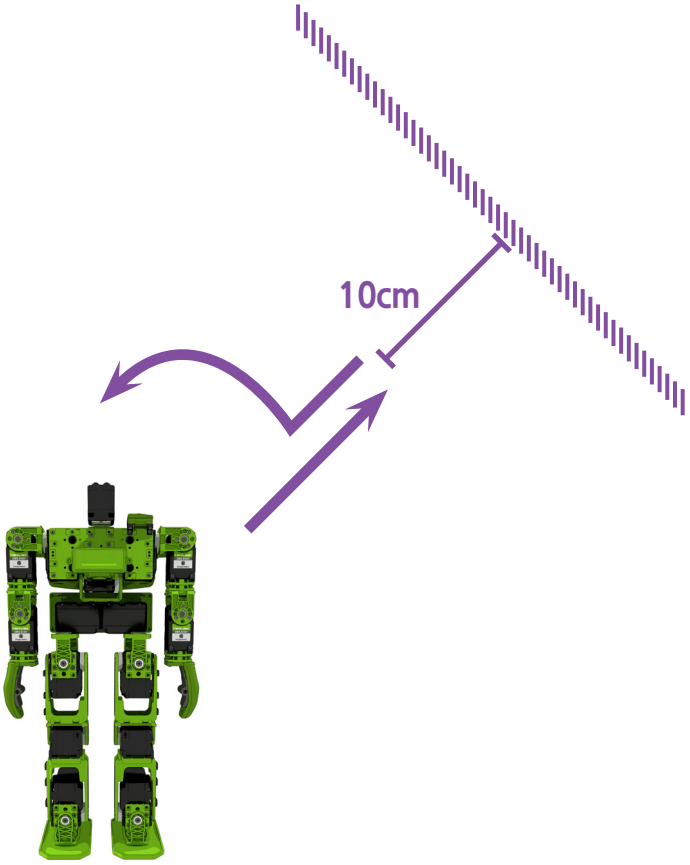
23 全体を見る

10cmよりも遠い場合にはロボットをそのまま前進させ、10cmより近い場合には、For文で後進と右に曲がるモーションを絶えず繰り返して壁を回避するプログラミングです。



24 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。



25 ロボット動作

ロボットは10cm前の壁を検出すると、後進してから右折し、そのまま前進します。