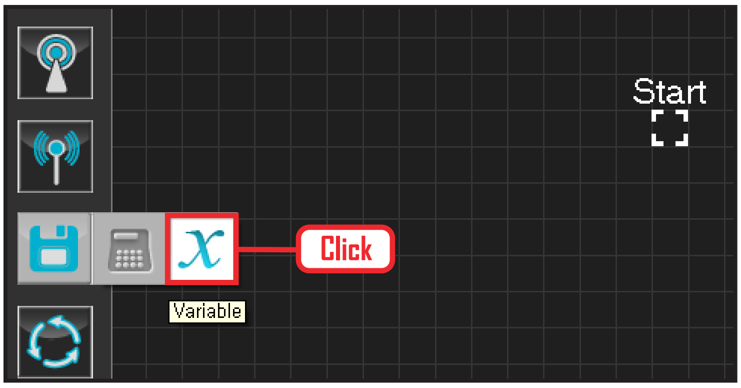
Digital Distance Sensor Example Step by Step

Example Description

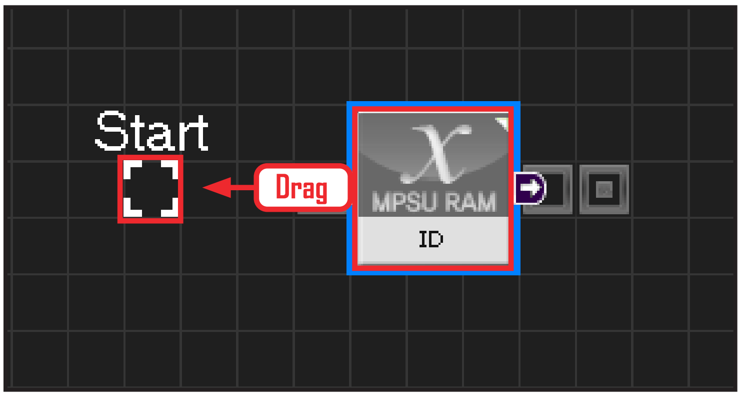
Analog sensor is capable of detecting the actual distance from an object whereas digital sensor uses specific distance as a reference to judge how far or near it is from the reference distance. Robots with wheels use the sensor for cliff detection more often than for object avoidance and humanoid robots with moving legs use the sensor for object avoidance rather than for cliff detection. This example will use the sensor for object detection and avoidance. Compare the program and the result with the analog sensor program. When the robot nears the wall, it will move backwards, change direction and move forward again. This example requires digital distance sensor to be installed at ADC port #1 (left).



01 Assign Variable

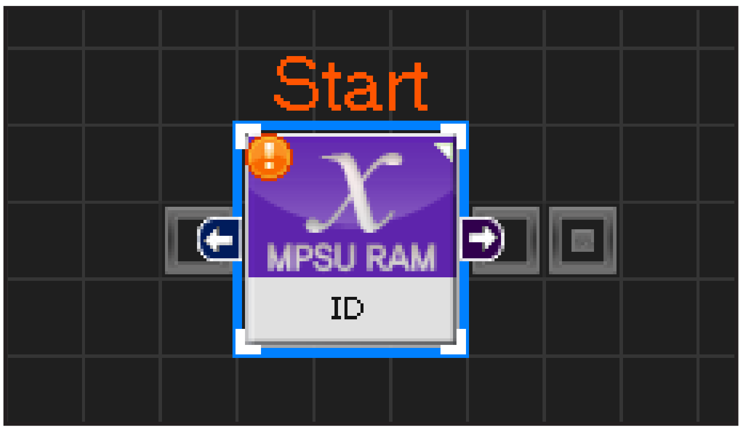
Operating the robot is same as operating the robot servo motor. Value has to be assigned so that servo will be able to operate.

Click Data 〉 Variable module.



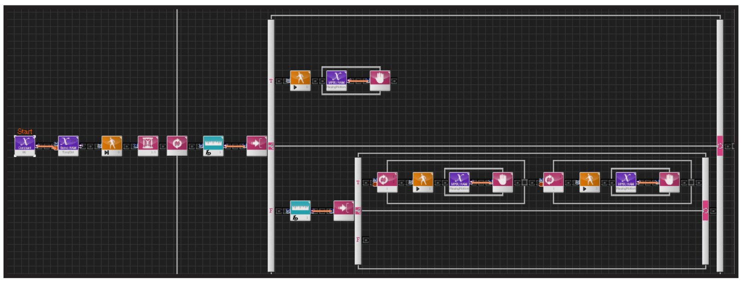
02 Start

Click and drag the connecting line located at left side of the module to the Start Point and dock



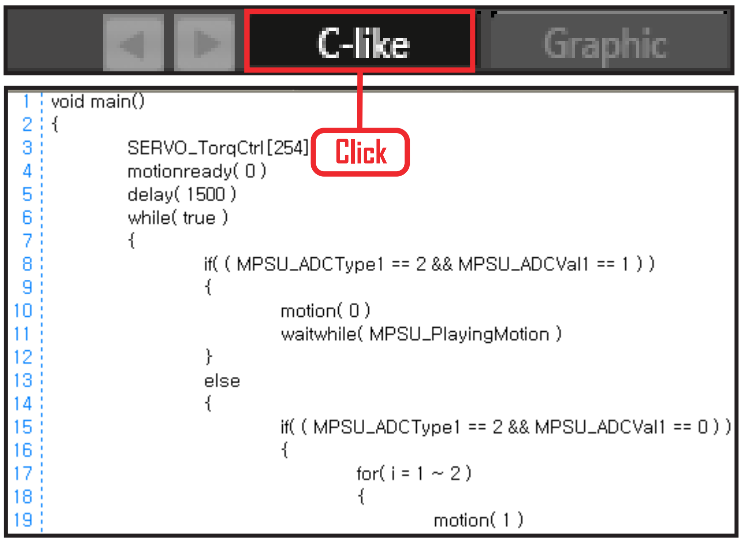
03 Start Programming

When the module and the Start Point is docked properly, module will become active and change color as seen in the photo to the left.This means programming has started..



04 Entire Program

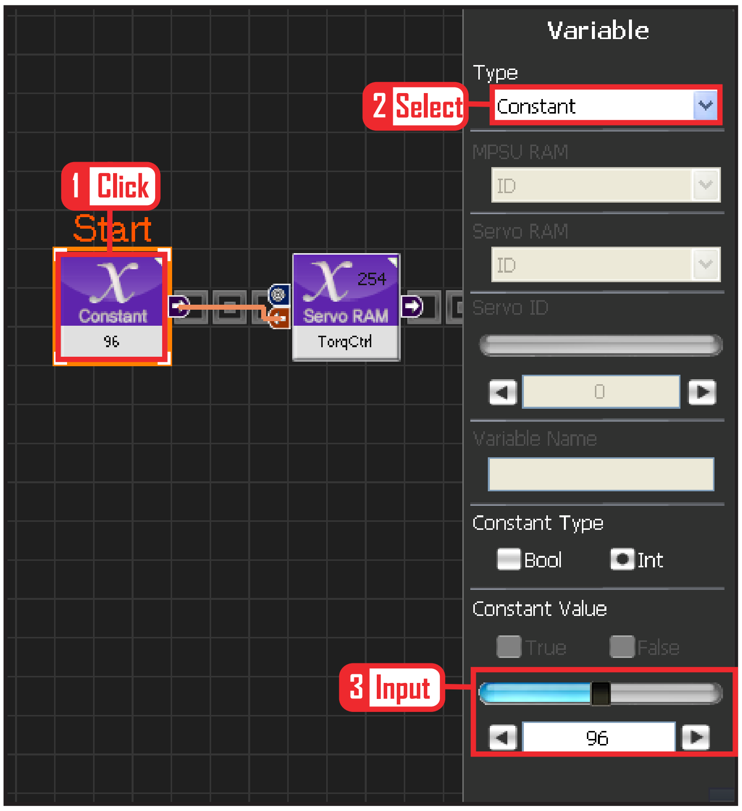
Entire program using the digital sensor.



```
void main()
{
        SERVO_TorqCtrl[254]
        motionready( 0 )
        delay( 1500 )
        while( true )
        {
                if( ( MPSU_ADCType1 == 2 && MPSU_ADCVal1 == 1 ) )
                {
                        motion( 0 )
                        waitwhile( MPSU_PlayingMotion )
                }
                else
                {
                        if( ( MPSU_ADCType1 == 2 && MPSU_ADCVal1 == 0 ) )
                        {
                                for( i = 1 ~ 2 )
                                {
                                        motion( 1 )
```

05 Viewing C-Like

Click the 'C-like' tab near the top right and task programming window will open as shown in the photo to the left. This is the task window of the entire program. Codes are very similar to the C language structure so studying the codes will help the user become familiar with the C language structure. Cursor will jump follwing the clicked module, making it easy to see the module changing to text.



Type: Constant

Constant Type: Bool [✓ Int]

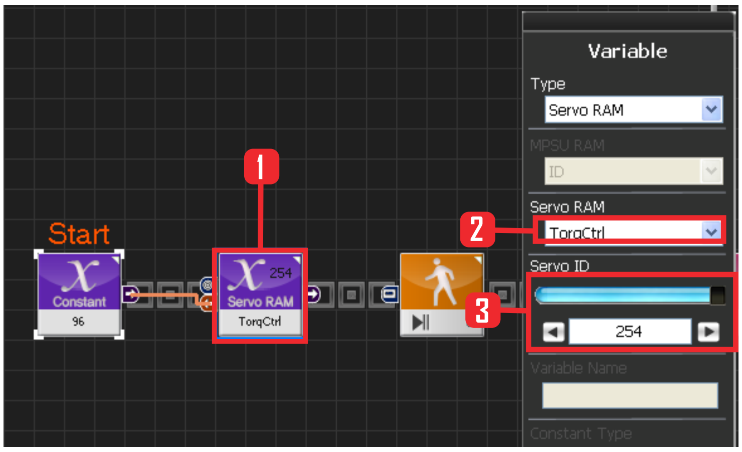
96

06 Setup Constant

This section allows the servo motor to operate on it's own.

Select Constant as the Variable Type. In properties, set constant value as 96.

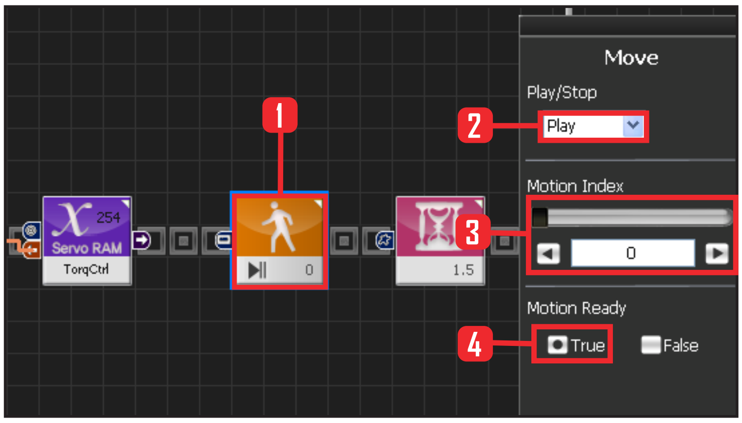
When 96(0x60) is entered in the servo TorgControl register, servo becomes ready to operate. This value is sent to the torque value of the next moduel through the output connector.



07 Apply to All Servos

This section applies contact value 96 to all servos.

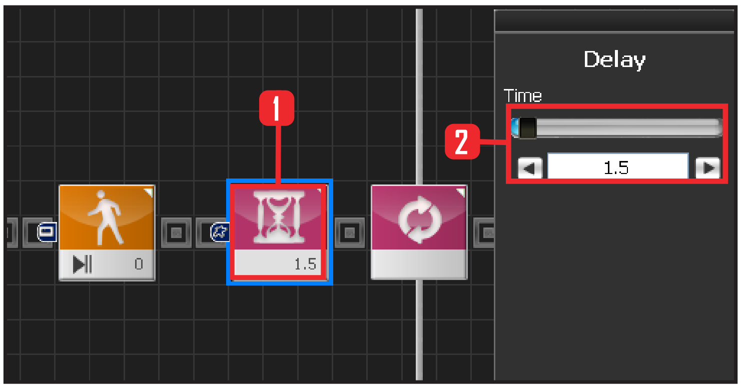
Select Variable 〉 Type : Servo RAM.
Select Servo RAM : TorqCtrl.
Set Servo ID : 254. 254 means it will be applied to all connected servos.



08 Motion Ready

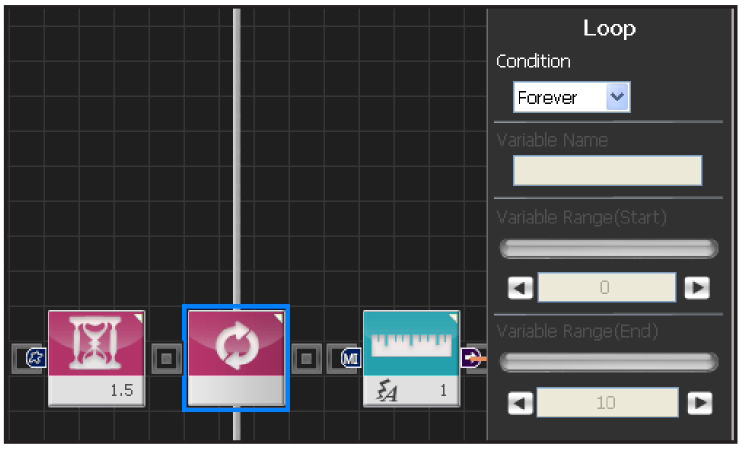
Robot goes through a prepatory stage before starting the next motion. This prepatory stage allows the robot to move slowly to the the initial position of the motion to be run. This prevents stress or damage from sudden change in motion.
IF Motion Ready is True prepare for next motion. Run next motion if False.

Select Motion 〉 Move module.
Select Play/Stop : Play.
Set Motion Index : 0 . walk forward
Select Motion Ready : True.
Motion Ready Stage



09 Delay

Set delay to 1.5s to prevent next step from staring before Motion Ready ends.



Loop

Condition: Forever

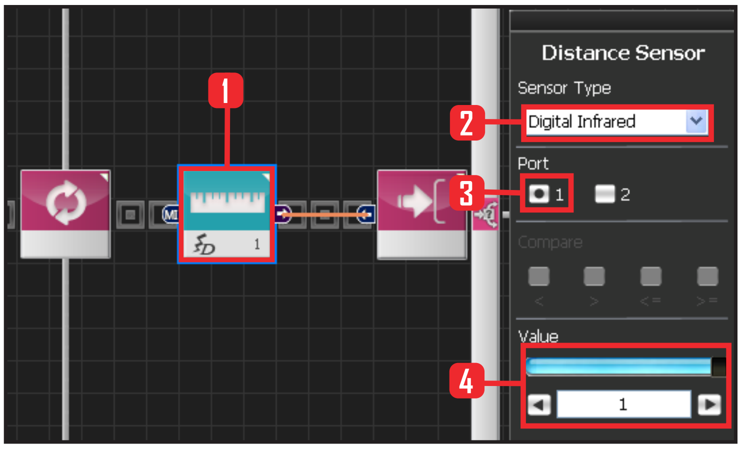
Variable Name

Variable Range(Start): 0

Variable Range(End): 10

10 Loop

Select Loop: Forever
Infinite loop.



Distance Sensor

Sensor Type: Digital Infrared

Port: 1 2

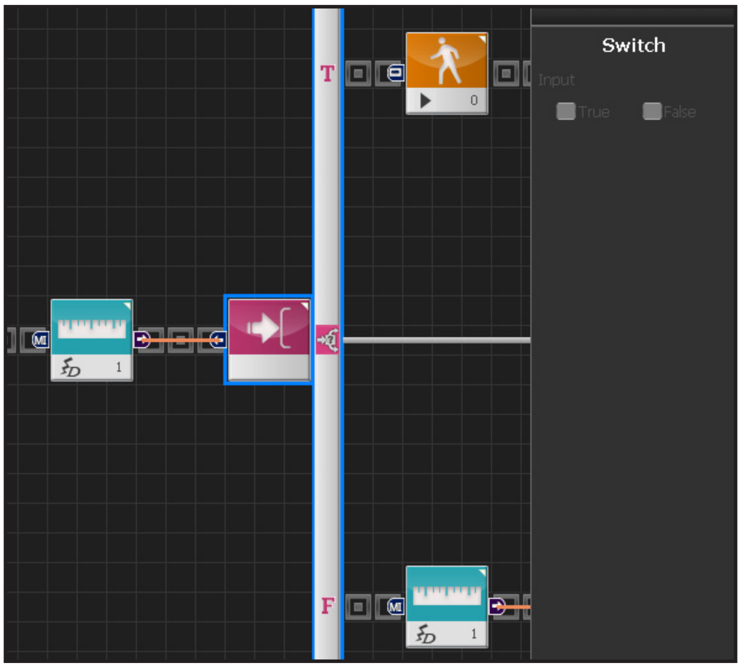
Compare: < > <= >=

Value: 1

11 Setup Digital Distance Sensor

Digital sensors have different measuing distance.
Setup with 20cm as standard.

Select Sensor 〉Distance Sensor module.
Select Sensor Type : Digital Infrared.
Select Port : 1.
Set Value : 1 . farther than 10cm.



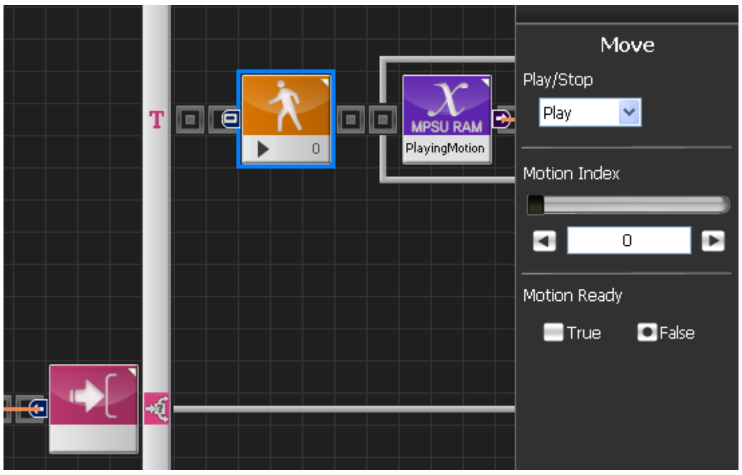
Switch

Input

True False

12 If Conditional Statement

Proceed if True, go to next conditional statement if False.



Move

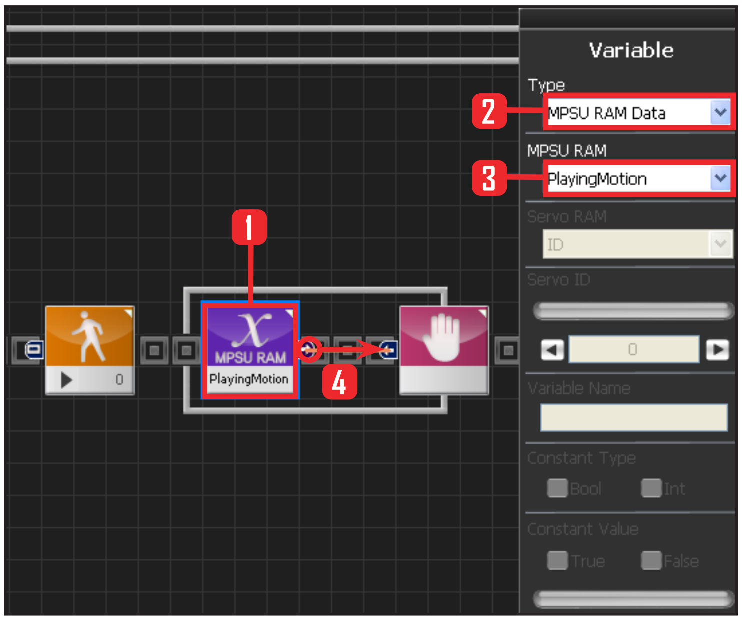
Play/Stop: Play

Motion Index: 0

Motion Ready: True [✓ False]

13 Forward

Robot will move forward since the distance is greater than10cm.
When False is selected as Motion Ready value, robot will proceed with forward motion.



Variable

Type: MPSU RAM Data

MPSU RAM: PlayingMotion

Servo RAM: ID

Servo ID: 0

Variable Name

Constant Type: Bool Int

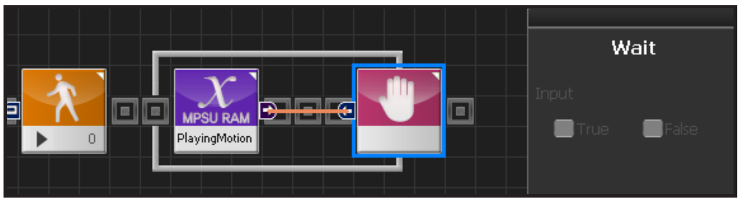
Constant Value: True False

14 Motion Movement Check

Loop refers to continuous repetition. It takes time for the actual motion to complete after Move command has been issued, but loop with single move module will continue to run and give motion command even while the previous motion is still running. The lag in actual motion will result in difference between the number of motion commands given by the move module and the number of actual motions. To correct this difference, loop will need to wait for the motion to complete before repeating the process.'Playing Motion' is found within Variable > MPSU RAM Data. 'Playing Motion' is a variable that checks whether the motion is in process. Loop will wait for the current motion to end if 'wait' is added to the 'Playing Motion'.

Select Data > Variable Module.
Select Type : MPSU RAM Data
Select MPSU RAM : Playing Motion
Add Wait module to the output connector.

Data > Variable.
Type : MPSU RAM Data.
MPSU RAM : Playing Motion.

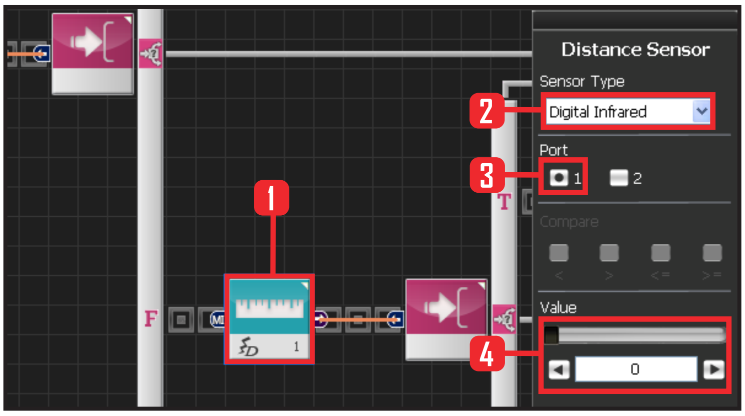


Wait

Input: True False

15 Wait

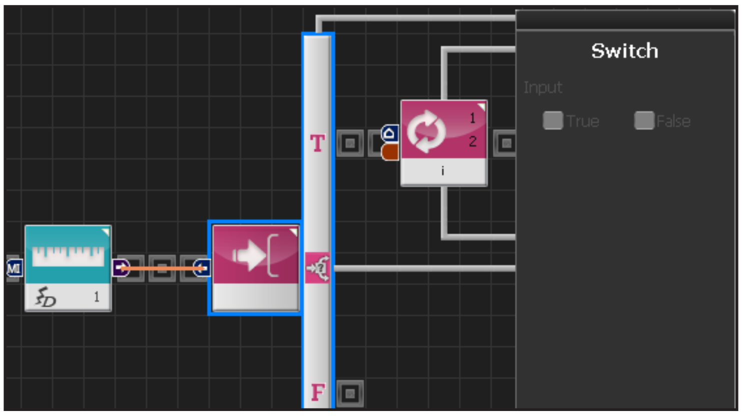
Wait untill the motion ends.
Go to the begining and repeat when motion ends.



16 Motion Near The Wall

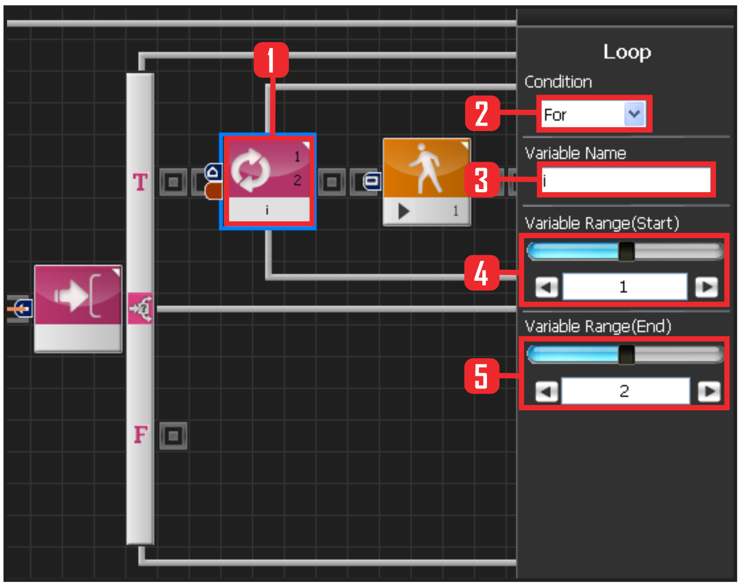
When the robot is less than 10cm from the wall, program will make the robot walk backwards and change direction.

Select Sensor 〉Distance Sensor module.
Select Sensor Type : Digital Infrared .
Select Port : 1.
Set Value : 0 . Within 10cm distance.



17 If Conditional Statement

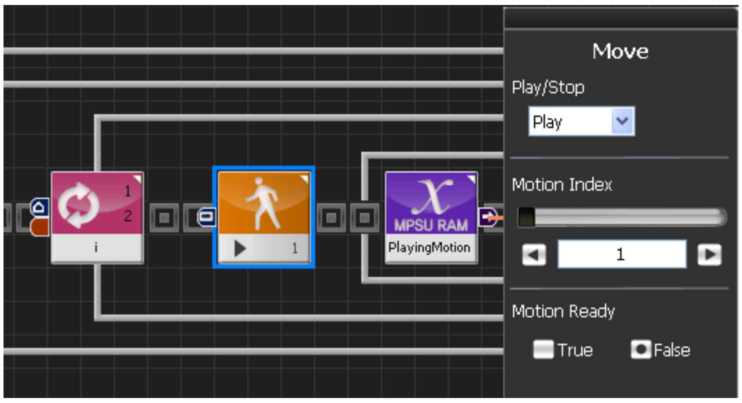
Run statement within True if less than 10cm from the wall.



18 For Loop

Repeat certain motion untill the condition is met. Motion #1 is a walk backwards motion. walk backwards motion makes the robot take one step backward each using left and rigt feet.
Robot can be moved to the desired location by adding For statement to the motion to repeat the motion desired number of times.

Select Flow 〉Loop module.
Select Condition : For .
Set Variable Name: i .
Set Variable Range(Start) 1 .
Set Variable Range(End) 2 .
Repeat motion twice.



Move

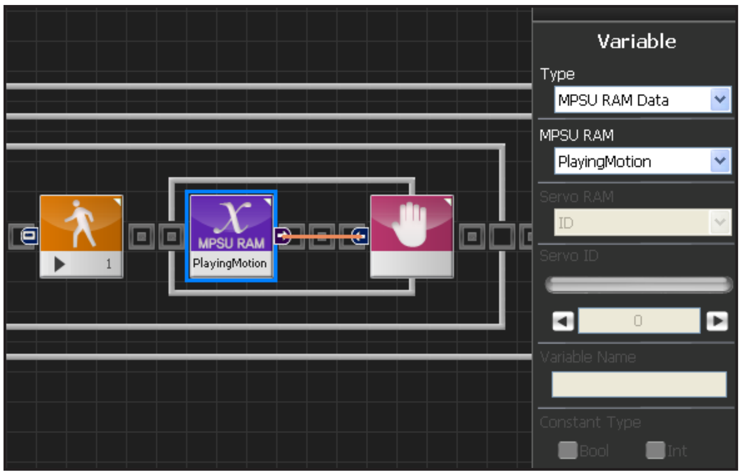
Play/Stop: Play

Motion Index: 1

Motion Ready: True [✓ False]

19 Walk Backwards

#1is a walk backwards motion.
Robot will run the walk backwards motion if False is selected.



Variable

Type: MPSU RAM Data

MPSU RAM: PlayingMotion

Servo RAM: ID

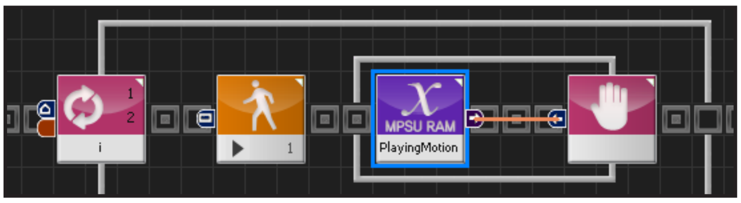
Servo ID: 0

Variable Name

Constant Type: Bool Int

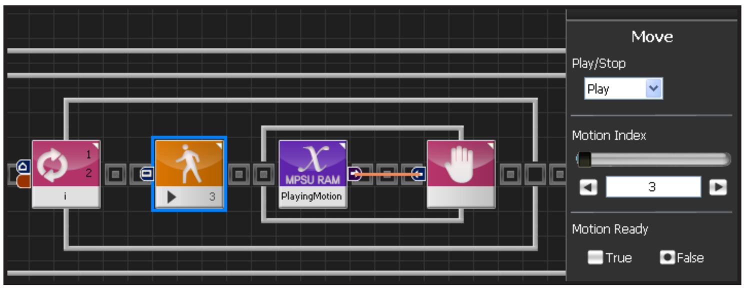
20 Check Motion

Use Playing Motion to check the robot motion. When the motion ends, return to the start of the For statement.



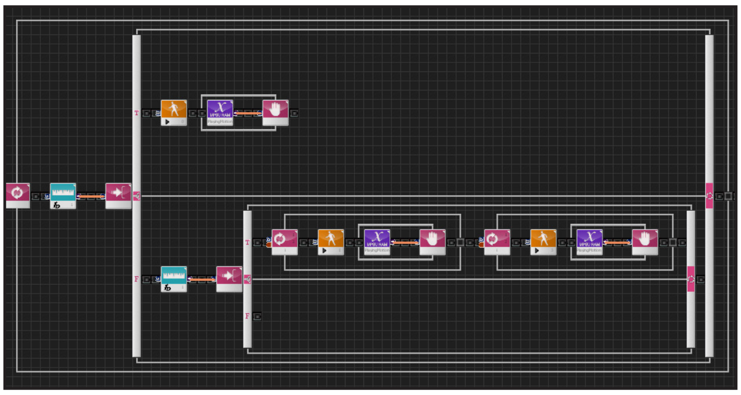
21 Repeat Backwards Motion Twice

Program makes the robot repeat the walk backwards motion twice.



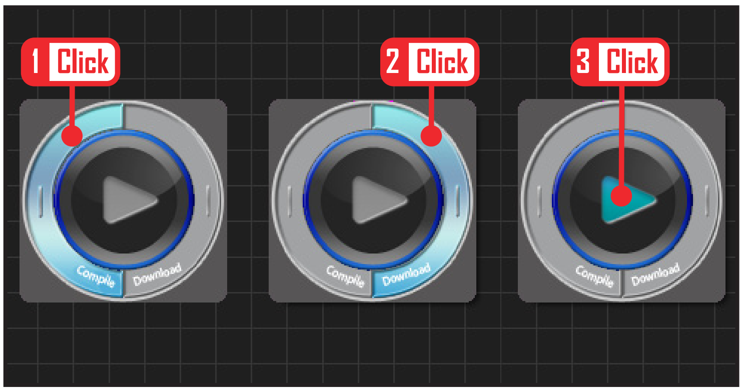
22 Right Turn

Robot motion #3 makes the robot change direction to the right. Right turn motion can be controlled by using the For statement . Seletct motion #3, set For statement from 1–3 and program as above.



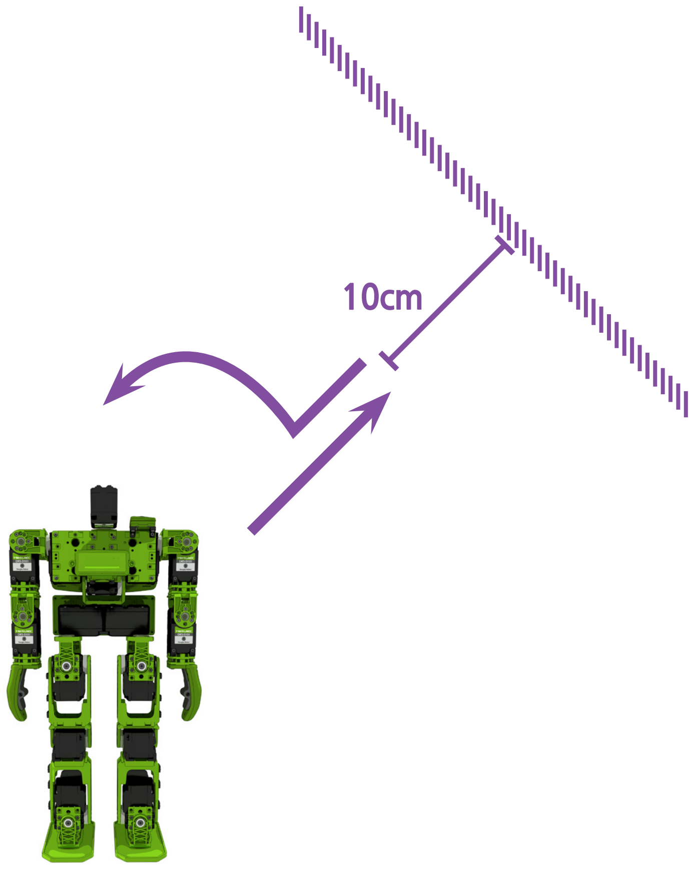
23 Entire Program

Program make the robot walk froward when the distance to the wall is greater than 10cm . If the distance is less than 10cm, robot will repeat the backward and right turn motion according to the For statement and avoid the obstacle.



24 Compile, Download, Run

Click 'Compile'. Click 'download' on the right if there is no compilation error. Download to robot. Click 'Run' button (Arrow button) after the download.



25 Robot Motion

When detects a wall within 10cm, it will walk backwards, change direction to the right and start walking forward again.