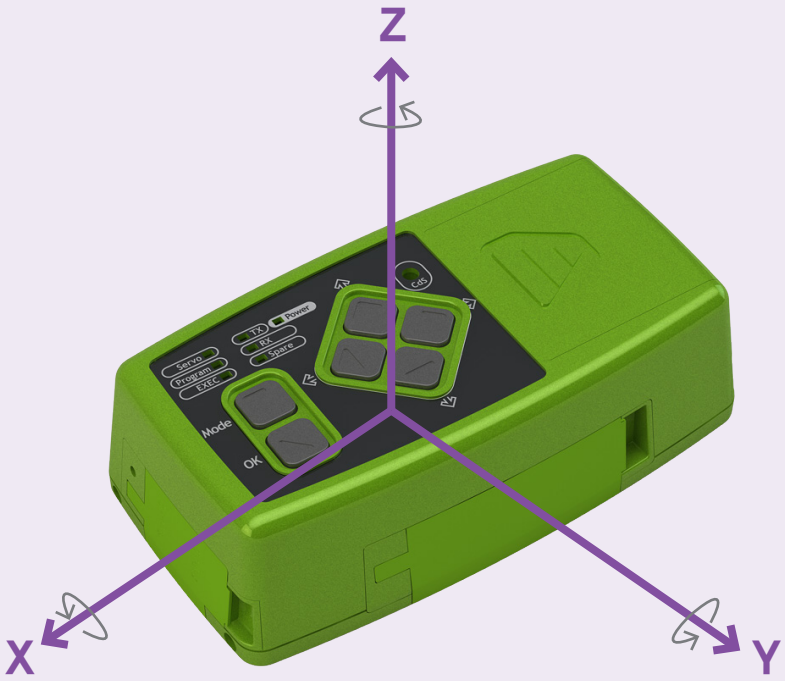
Acceleration Sensor Example Step by Step

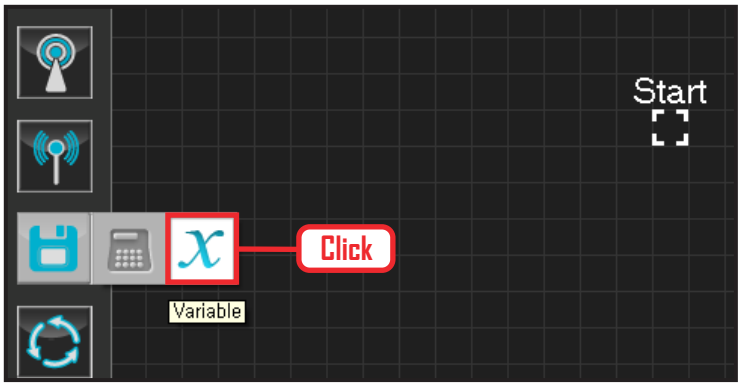
Example Description

Use the Acceleration sensor to make the robot stand when it falls forward or backward.

Acceleration sensor is attached to a module type board that also has Gyro sensor attached to it. Sensor module can be installed inside the controller by opening the controller back cover.



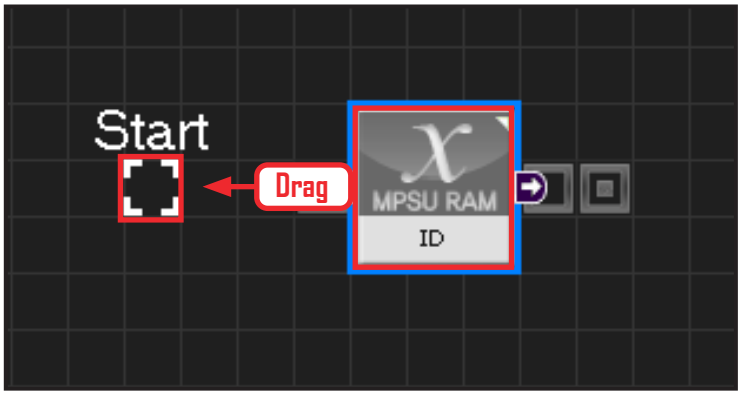
- When the robot is in prone position (lying face down),
  Z axis "+" acclerates and it's value is approximately +256.
- When the robot is in supine position (lying on the back),
  Z axix "-" accelerates and it's value is approximately -256. (256 is approximately 1g force of gravity value.)



01 Assign Variable

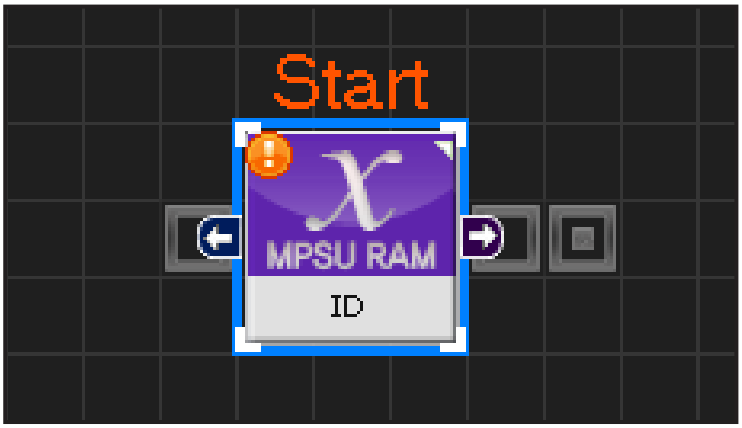
Operating the robot is same as operating the robot servo motor. Value has to be assigned so that servo will be able to operate.

Click Data 〉 Variable module.



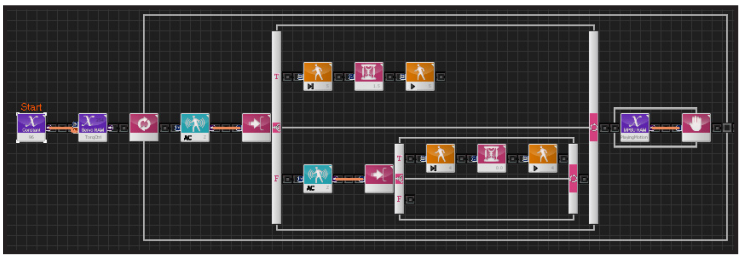
02 Start

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



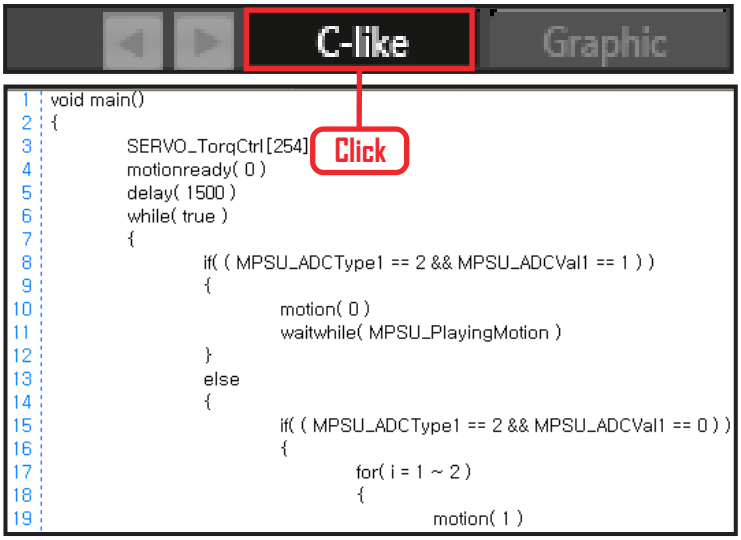
03 Start Programming

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..



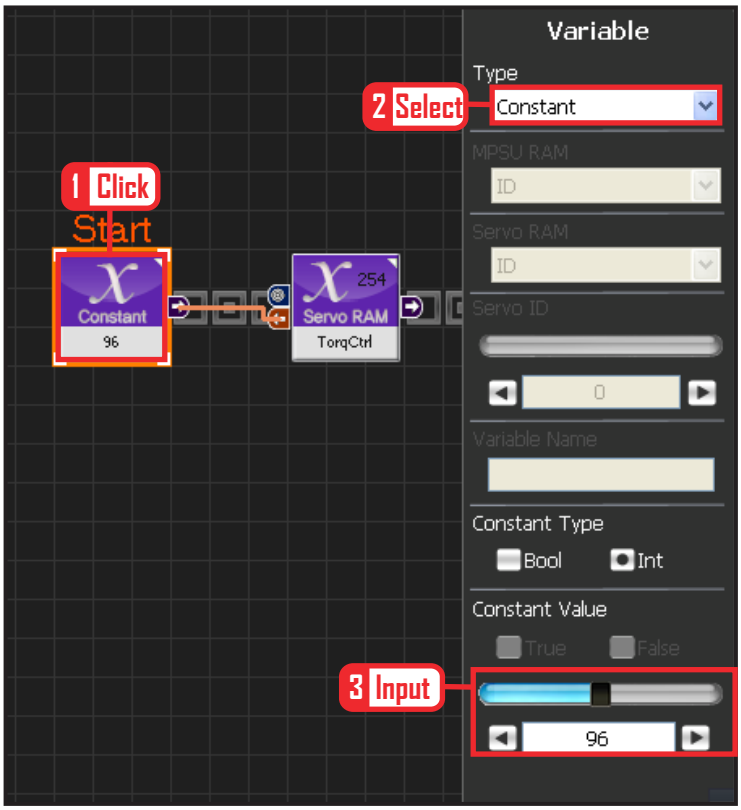
04 Entire Program

Entire program using the accleration sensor to make the robot stand after falling.



05 View 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.

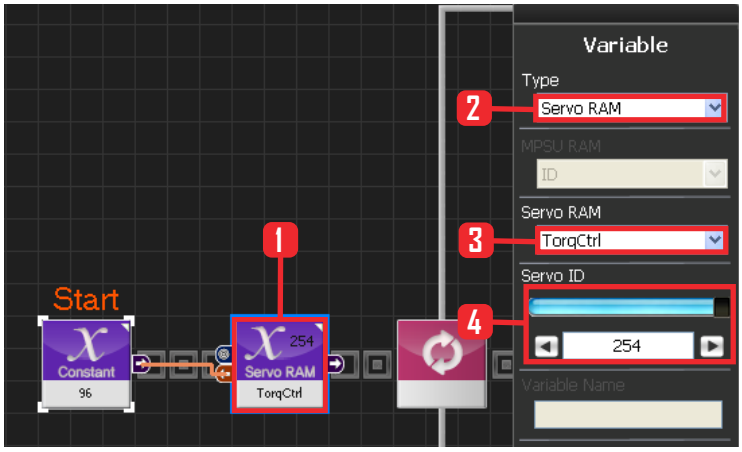


06 Setup Constant

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.

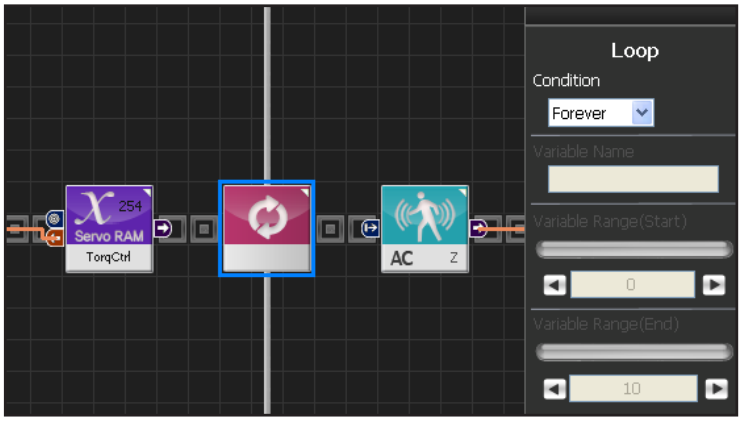
When 96(0x60) is entered in the servo TorgControl register, servo becomes ready to operate. This value is sent to the torque value of the next moduel through the output connector.



07 Apply to All Servos

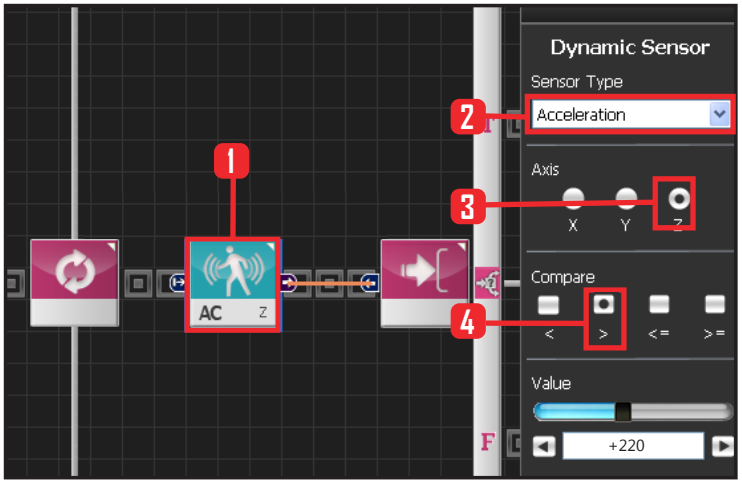
This section applies contact value 96 to all servos.

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



08 Loop

Select Flow 〉Loop module.
Select Condition: Forever.
Infinite loop.

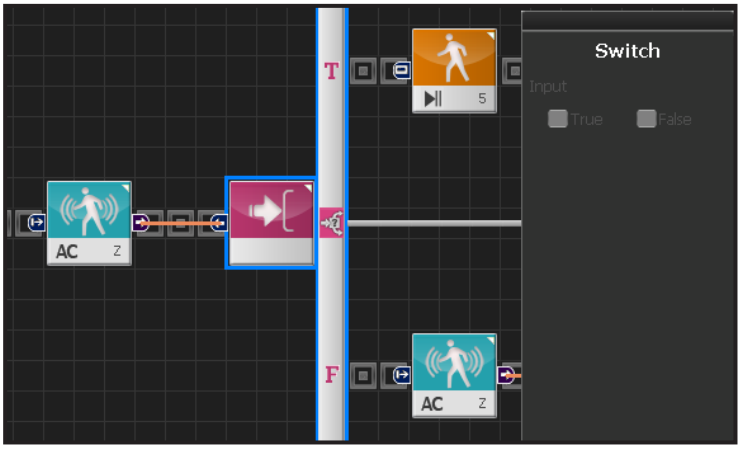


09 Acceleration Setup (Prone)

Acceleration has value of 0 when the robot is standing up straight.
When the robot is in prone position it has value of +256 and –256 when in supine position.
If the accleration value is near +256, it can be assumed that the robot has fallen forward. Set +220 as standard value.

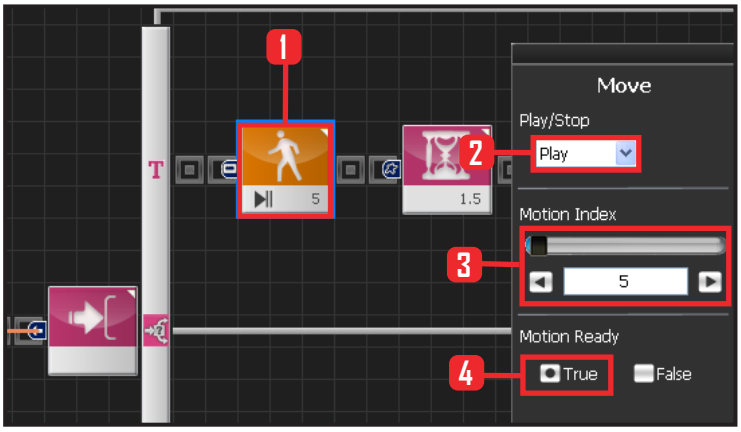
if the value is less thant +220 robot is assumed to have fallen forward.

Select Sensor 〉Dynamic Sensor moduleSelect
Sensor Type : Acceleration
Select Axis : Z
Select Compass : 〉
Set Value : +220



10 If Conditional Statement

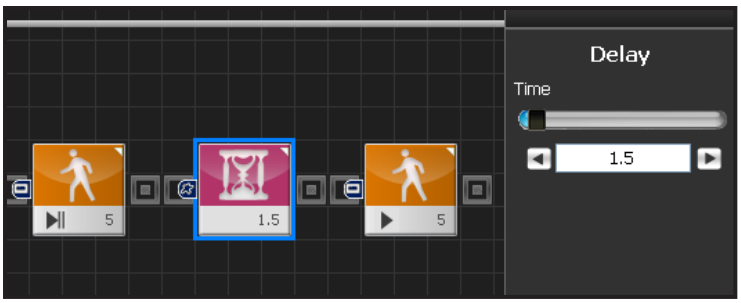
Robot gets up backward when True, Proceed to next conditional statement if False.



11 Run Up Backwards Motion

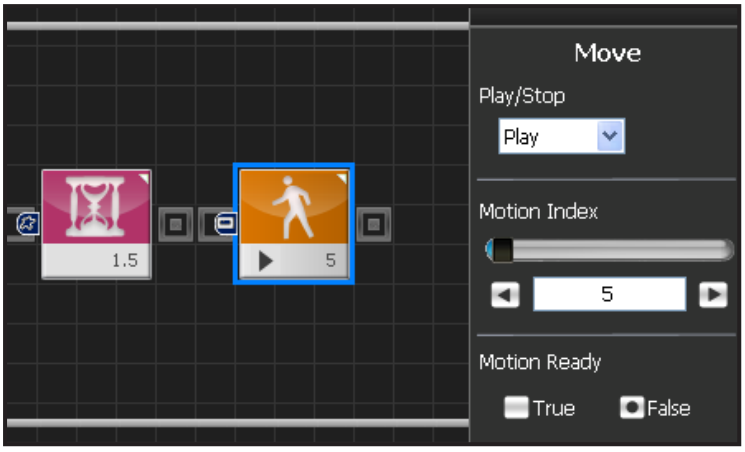
Insert Up Backwards motion when the robot is in prone position.
Motion #5 is up backward motion.

Select Motion 〉Move module
Select Play/Stop : Play
Set Motion Index : 5
Select Motion Ready : True
Prepatory stage for motion



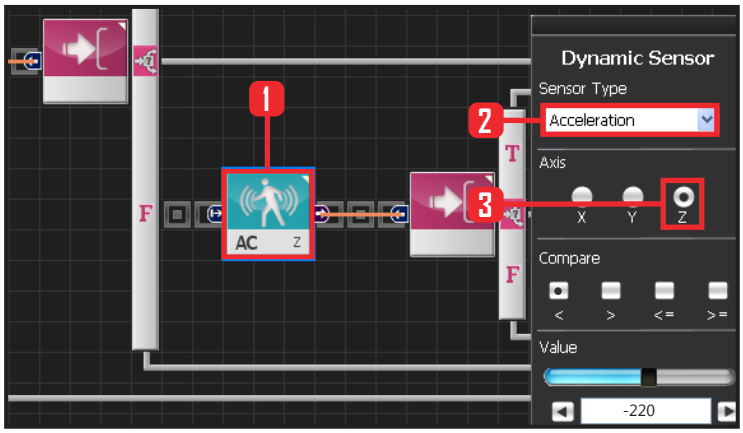
12 Delay

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



13 Run Up Backwards Motion

When False is selected as Motion Ready value, robot will run the up backwards motion.

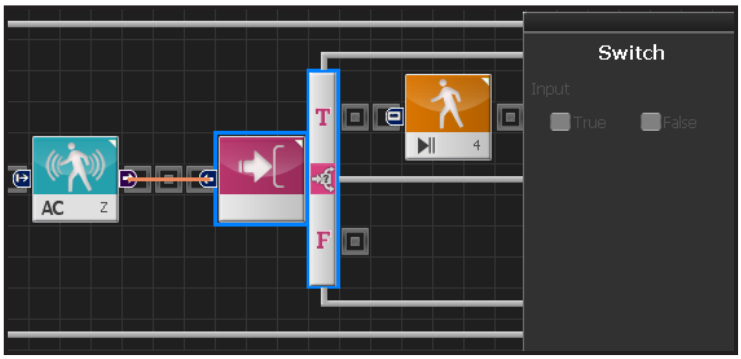


14 Setup Gravity Acceleration (Supine Position)

Gravity acceleration has value of 0 when the robot is standing up straight.
When the robot is in prone position it has value of +256 and −256 when in supine position.
If the accleration value is near −256, it can be assumed that the robot has fallen backward. Set −220 as standard value.

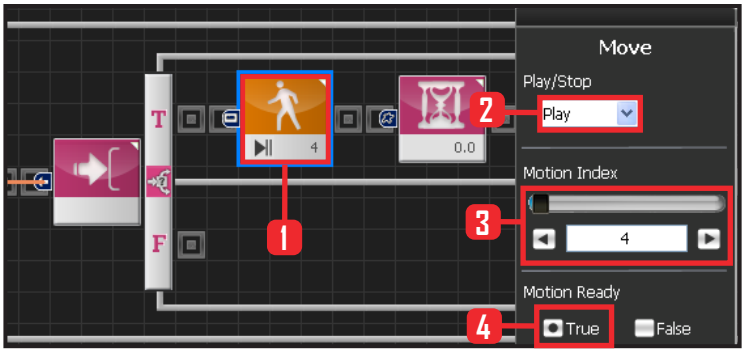
Select Sensor 〉Dynamic Sensor module

Select Sensor Type : Acceleration
Select Axis : Z
Select Compass : 〉
Set Value : −220



15 If Conditional Statement

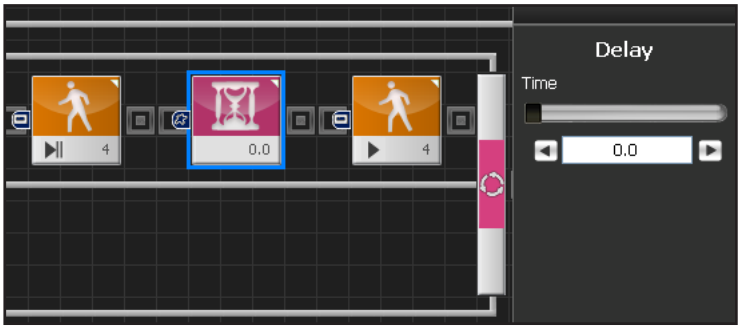
Robot gets up if True.



16 Motion Ready

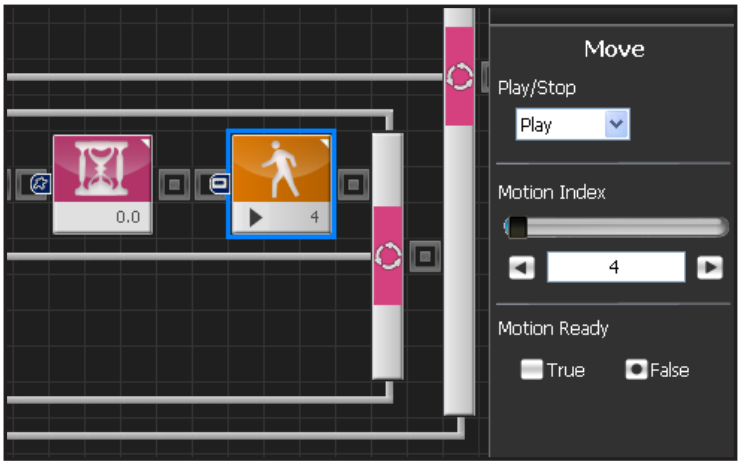
Robot goes through a prepatory stage before starting the next motion. This prepatory stage allows the robot to move slowly to the the initial position of the motion to be run. This prevents stress or damage from sudden change in motion. IF Motion Ready is True prepare for next motion. Run next motion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

Select Motion 〉Move module
Select Play/Stop : Play
Select Motion Index : 4 . Motion # 4, robot gets up forward
Select Motion Ready : True
Motion ready stage.



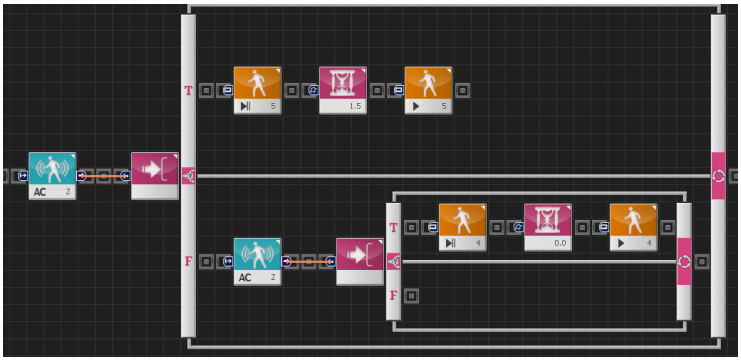
17 Delay

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



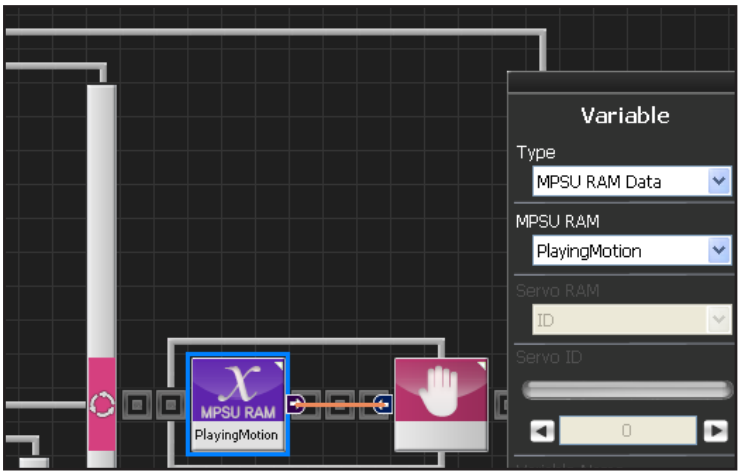
18 Run Up Forward Motion

When False is selected as Motion Ready value, robot will run the up forward motion.



19 Getting Back Up

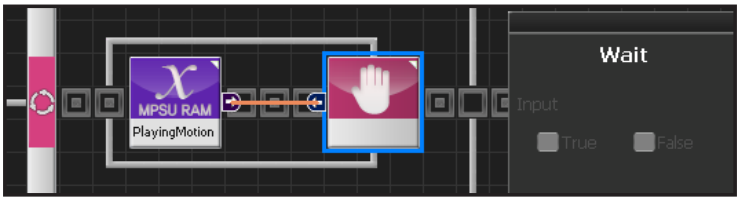
Robot determines if it has fallen by referencing the Z axix acceleration value and runs the appropriate motion to get back up.



20 Motion Movement Check

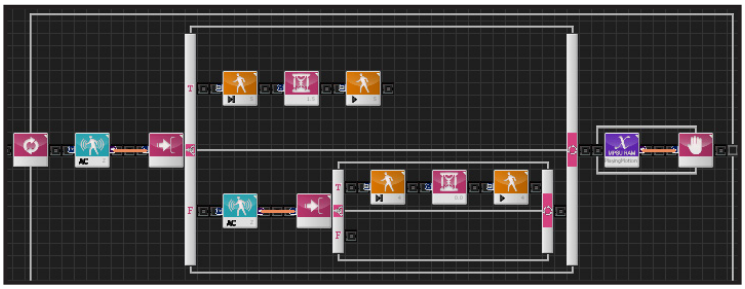
Loop refers to continuous repetition. It takes time for the actual motion to complete after Move command has been issued, but loop with single move module will continue to run and give motion command even while the previous motion is still running. The lag in actual motion will result in difference between 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.



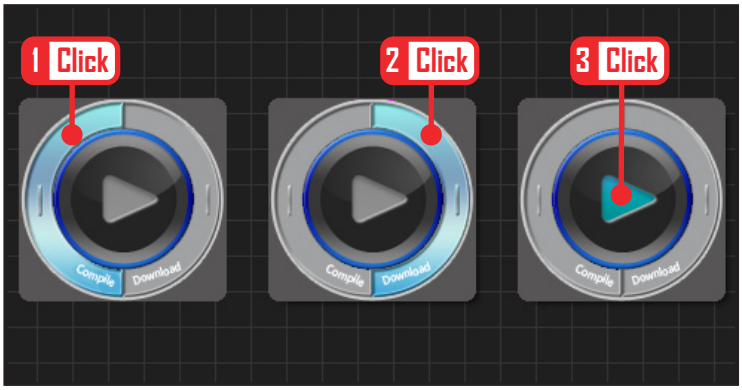
21 Wait

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



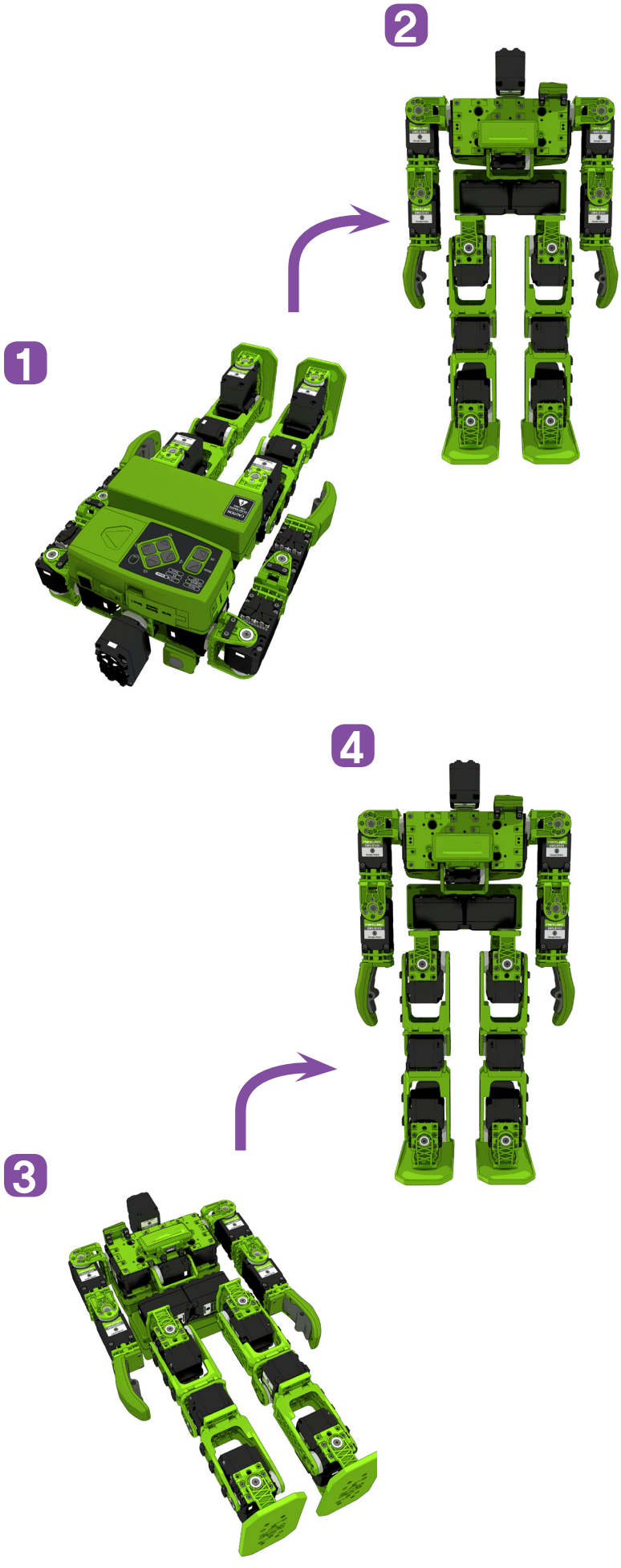
22 Entire Program

Robot detremines if it has fallen backwards or forward and runs the appropriate motion to get back up.



23 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.



24 Robot Motion

If the robot is in prone position, it gets back up backwards. If it is in supine position, it gets back up forward.