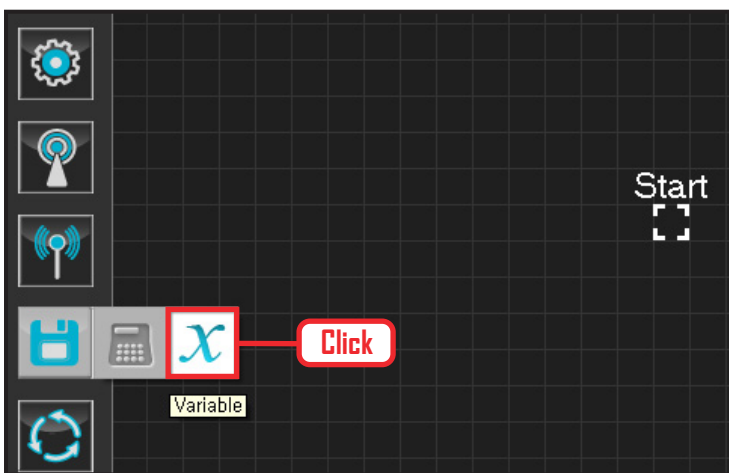


Move Example Step by Step

Example Description

Move module loads the robot motion saved in the DRC controller and applies it to the program. Robot motion can be loaded by the number, and if required, names can be checked from DR-SIM. This program will repeat running the motion created by DR-SIM on the robot indefinitely. This is a relatively complicated program useful for reliability test or for demonstration purposes.

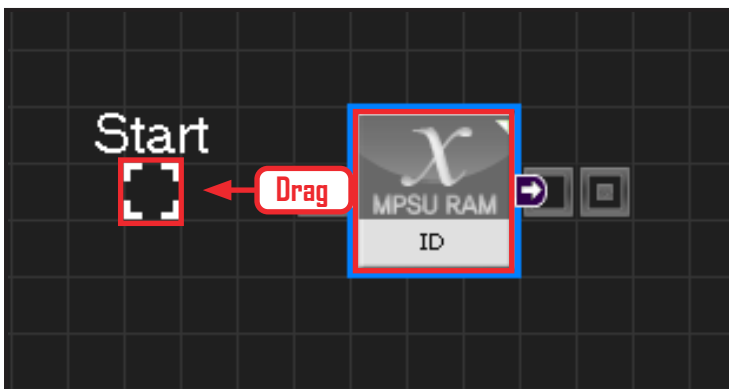
* Motions and Motion numbers used in this example are not same as the provided basic motion. This example assumes motion was created by DR-SIM and downloaded to the DRC. To download motion go to www.hovis.co.kr/guide



01 Assign Variable

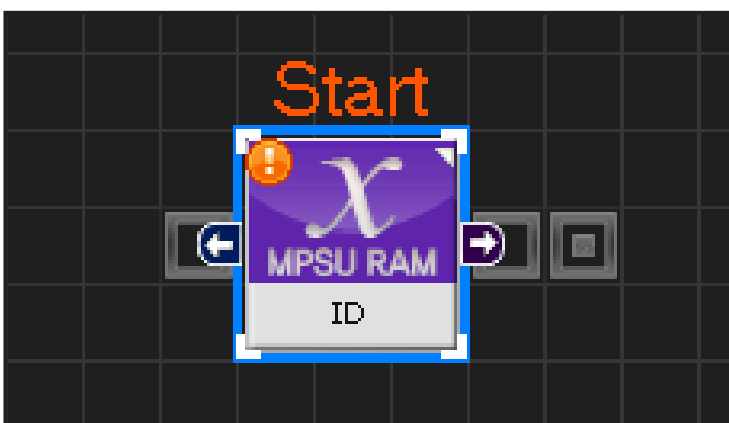
Operating the robot is same as operating the robot servo motor. Value has to be assigned so that servo will be able to operate.

Click Data > Variable module.



02 Start

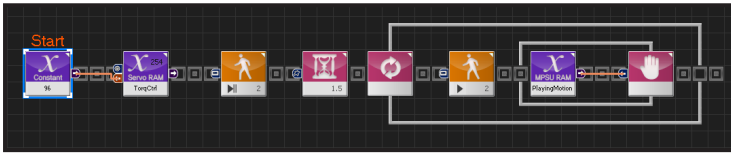
Click and drag the connecting line located at left side of the module to the Start Point and dock.



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

Loads the saved motion and duplicates 저장된 모션을 가져와 일정하게 반복 시키는 프로그래밍입니다. Motion ready 값에 주의합니다.

```

1 void main()
2 {
3     SERVO_TorqCtrl[254]=96
4     motionready(2)
5     delay(1500)
6     while(true)
7     {
8         motion(2)
9         waitwhile(MPSU_PlayingMotion)
10    }
11 }

```

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 following the clicked module, making it easy to see the module changing to text.

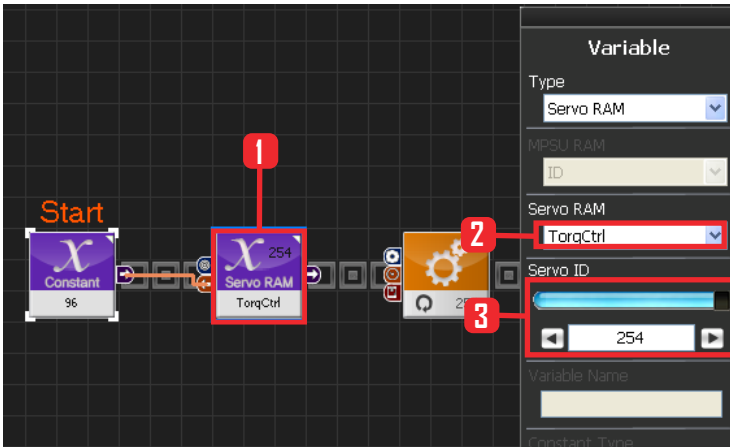
1 Click Start

2 Select

3 Input

06 Variable Setup

This section makes 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 TorqControl register, servo becomes ready to operate. This value is sent to the torque value of the next module through the output connector.



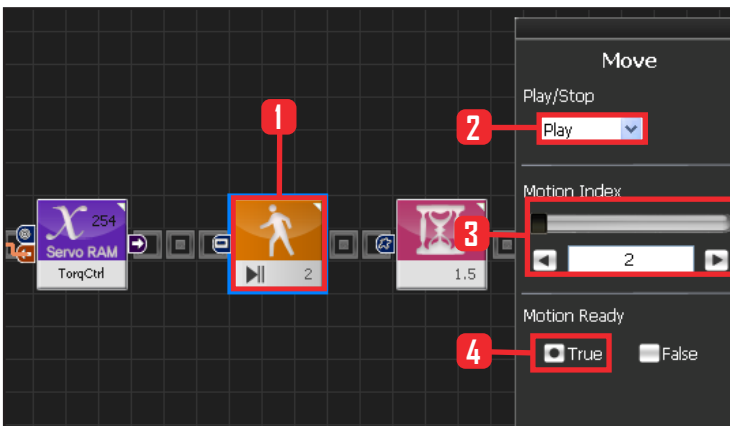
07 Apply to All Servos

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..



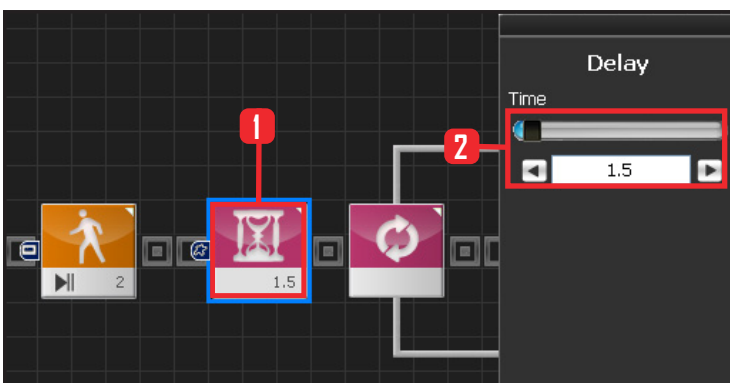
08 Motion Ready

When the motion is loaded, robot may make a sudden movement or motion change from the current position. If the difference between the current position and the start of the motion is drastically different, it may cause stress to the motors or pose danger to the user. To prevent such an occurrence 'Motion Ready' is used to give time for motion to start.

Select Motion > Move .

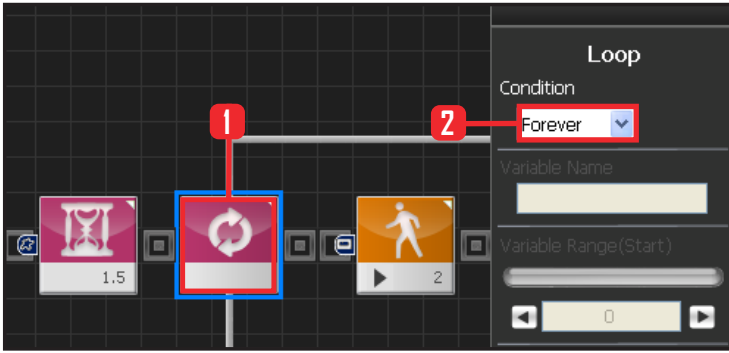
Select Play/Stop : Play .

Select Motion Index : 2, Load motion No 2. As a reference, motion No 2 in this program makes the robot sit and stand. It does not necessarily have to be No 2. User can select another motion No to use. Select Motion Ready : True . When True is selected, robot will slowly change to starting position of the motion.



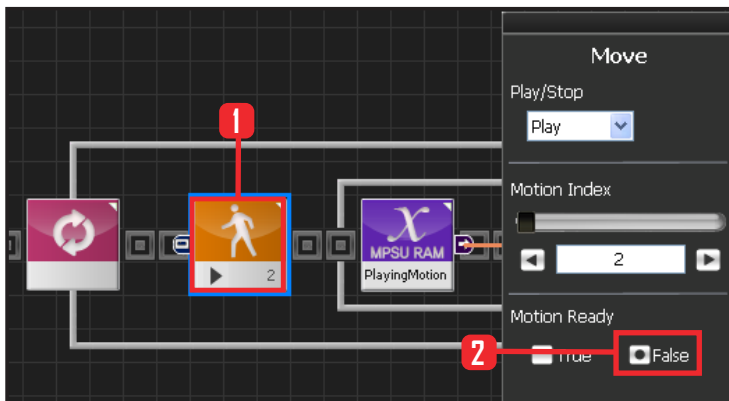
09 Delay

To prevent the motion from starting before Motion Ready ends, set Delay value to 1.5s.



10 Loop

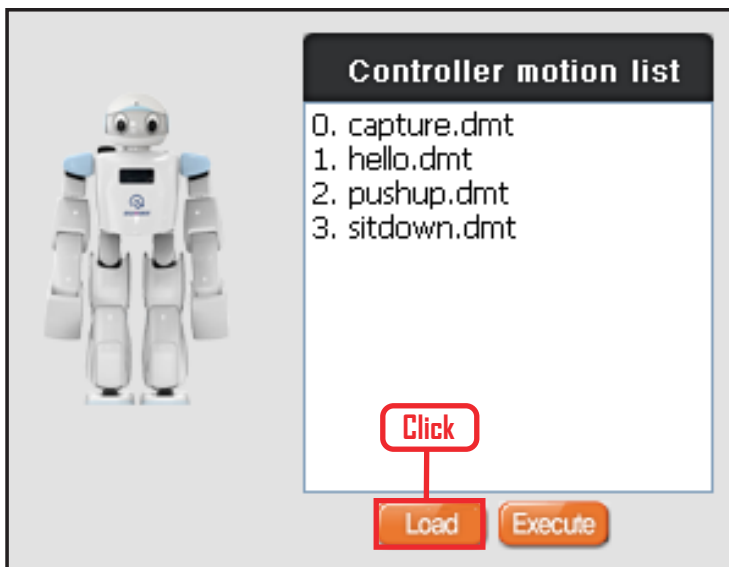
Set Forever infinite loop.



11 Motion Movement

If Motion Ready value is set to False, motion will run from start to finish.

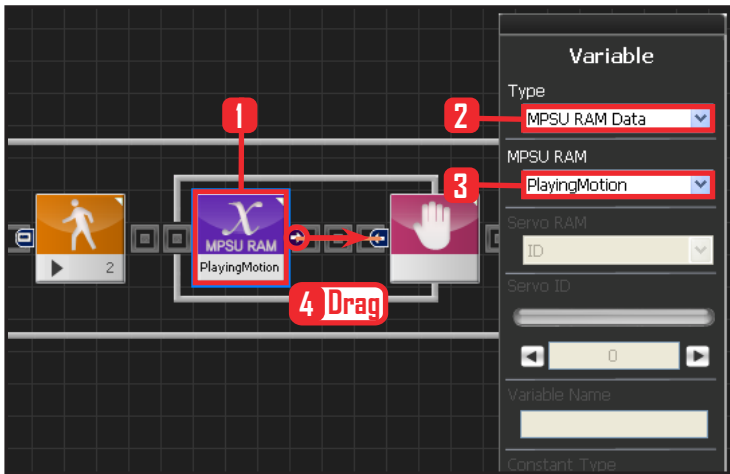
Select Motion Ready : False .



Reference: View Motion

To view the list of motions in the controller, connect to the robot and click robot setting from DR-SIM.

No 2 motion Robot sits and stands.

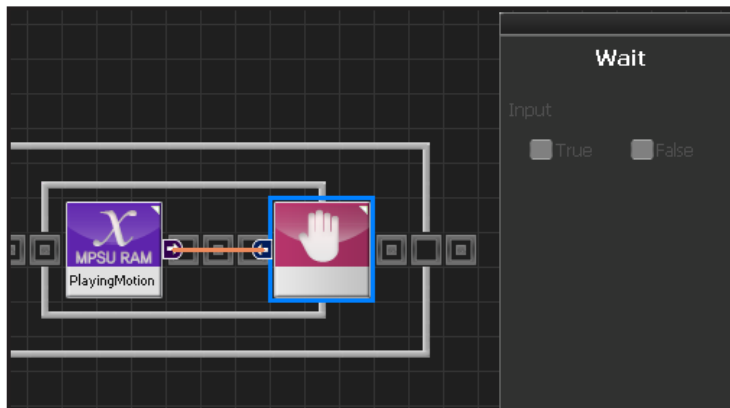


12 Check Motion Movement

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'.

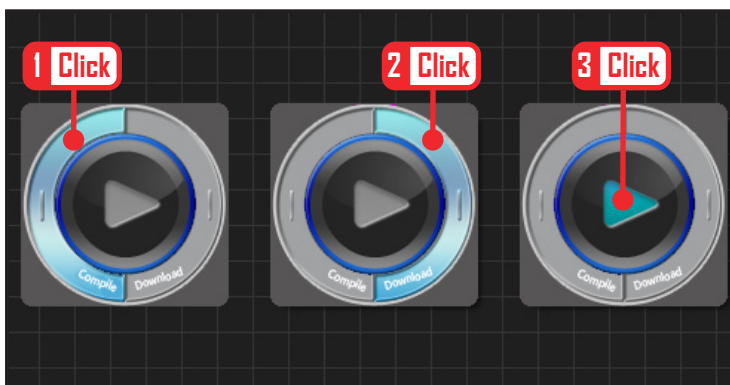
Select Data > Variable Module.
 Select Type : MPSU RAM Data
 Select MPSU RAM : Playing Motion
 Add Wait module to the output connector.



13 Wait

Wait until the motion ends.

Go to the beginning and repeat when motion ends.



14 Compile, Download, Run

Click left icon to compile. If no compile error is found, click right icon and download to robot. After the download is complete, click the arrow like run button in the middle to apply the program to the robot.

1



2



15 Robot Motion

Robot will continuously repeat sit and stand motion.