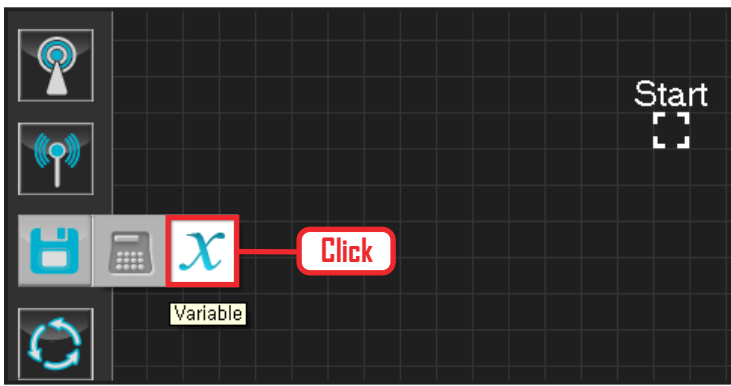


Analog Distance Sensor 예제 따라하기

예제설명

아날로그 거리센서를 이용하여, 벽이 가까워지면 좌회전 하면서 벽을 회피하는 프로그래밍입니다

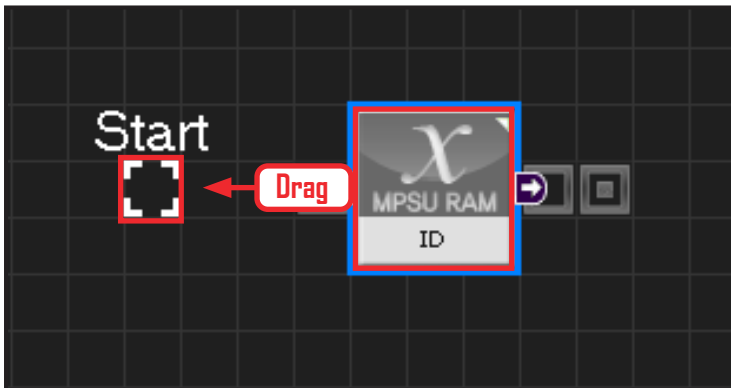
거리센서에는 아날로그 센서와 디지털 센서 가 있습니다. 디지털 센서는 일정거리(10cm) 를 기준으로 안쪽과 바깥쪽 두 영역만 감지하는 반면, 아날로그 센서는 거리별로(6~40cm) 위치를 감지할 수 있습니다. 이 예제를 실행하려면 ADC 포트 1번 (좌측)에 PSD센서를 장착한 상태여야 합니다.



01 변수 지정

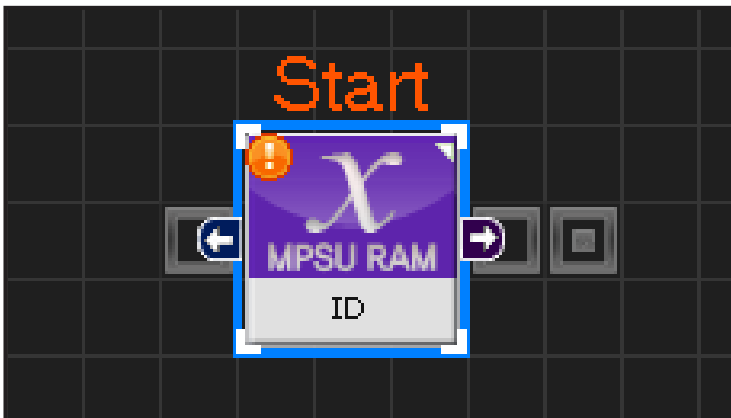
로봇을 동작시킨다는 것은 로봇의 서보 모터를 동작시킨다는 의미입니다. 서보가 스스로 움직일 수 있는 상태로 값을 지정해주어야 합니다.

Data > Variable 모듈을 클릭합니다.



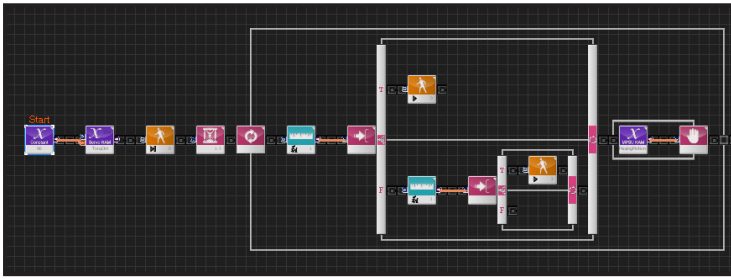
02 시작

모듈의 왼쪽 연결선을 Start Point 에 드래그하여 정확히 도킹을 시킵니다.



03 프로그래밍 시작

모듈과 Start Point 가 정확히 도킹하면 왼쪽과 같이 활성화된 칼라 이미지 모듈로 변합니다. 그럼 프로그래밍이 시작되었다는 의미입니다.



C-like Graphic

```

1 void main()
2 {
3     SERVO_TorqCtrl[254]
4     motionready( 0 )
5     delay( 1500 )
6     while( true )
7     {
8         if( ( MPSU_ADCType1 == 1 && MPSU_ADCVal1 >= 20 ) )
9         {
10            motion( 0 )
11        }
12        else
13        {
14            if( ( MPSU_ADCType1 == 1 && MPSU_ADCVal1 < 20 ) )
15            {
16                motion( 2 )
17            }
18            else
19            {
20            }
21        }
22    }
23 }
    
```

1 Click Start

2 Select Type: Constant

3 Input Constant Value: 96

04 전체 프로그래밍

아날로그 거리센서와 전진걸기 모션, 좌로돌기 모션을 이용하여 전체 프로그래밍해봅니다.

05 C-Like 보기

오른쪽 상단의 Graphic 탭에서 C-like 탭을 클릭하면 왼쪽과 같은 Task 프로그래밍 화면이 나옵니다. 아날로그 거리센서를 이용한 전체 프로그래밍 화면입니다.

C와 유사한 문법 구조를 가지고 있으므로 C 문법 선행학습 효과도 있습니다.

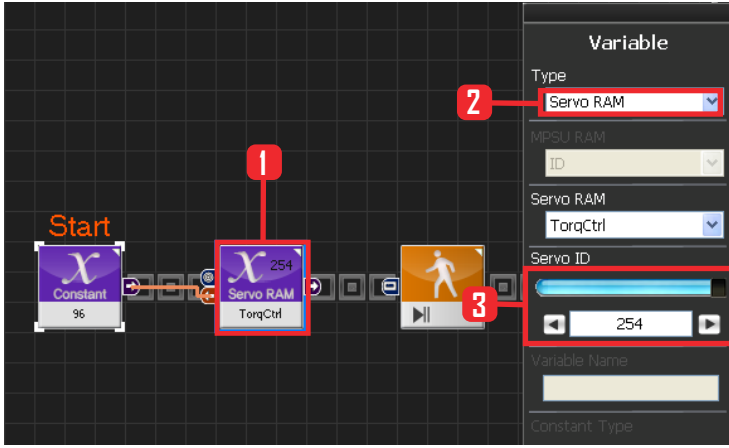
각 모듈별로 클릭하면 커서가 따라서 움직이므로 모듈별로 Text로 어떻게 변환하는지 확인할 수 있습니다.

06 상수 설정

서보 모터를 스스로 움직일 수 있는 상태로 만드는 과정입니다.

Variable Type 을 Constant 로 선택합니다.

속성중에 Constant Value 값을 96 으로 설정합니다. 서보의 TorqControl 레지스터에 96(0x60) 이라는 값이 들어가면 서보가 움직일 수 있는 상태가 됩니다. 그 값은 Output 커넥터를 통하여 뒤 모듈의 토크값에 전달합니다.



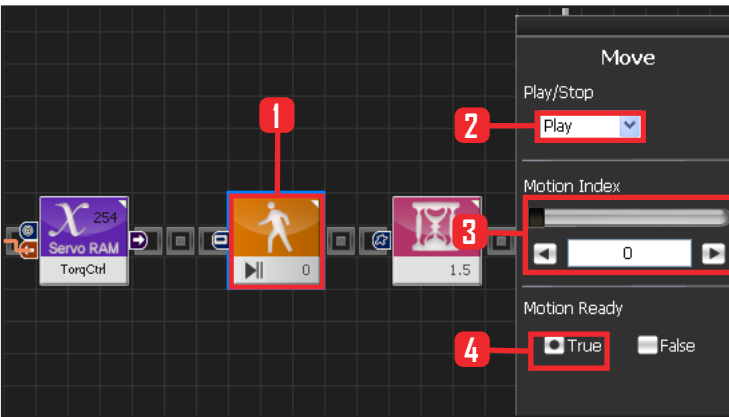
07 모든 서보에 적용

앞에서 받은 96 상수값을 모든 서보에 적용하는 과정입니다.

Variable > Type : Servo RAM을 선택합니다.

Servo RAM : TorqCtrl 을 선택합니다.

Servo ID : 254 를 선택합니다. 254는 연결되어있는 모든 서보에 적용하겠다는 의미입니다.



08 모션 동작준비

모션이 동작하기 위해서는 준비과정을 거칩니다. 이전 로봇 모션에서 갑작스럽게 변동하면 로봇에 무리가 가해질 수 있습니다. 따라서 현재 실행하고자 하는 로봇의 첫 모션으로 천천히 이동시키는 과정입니다. Motion Ready 가 True 이면 모션 첫 장면을 준비하는 것이고, False 이면 모션이 동작합니다.

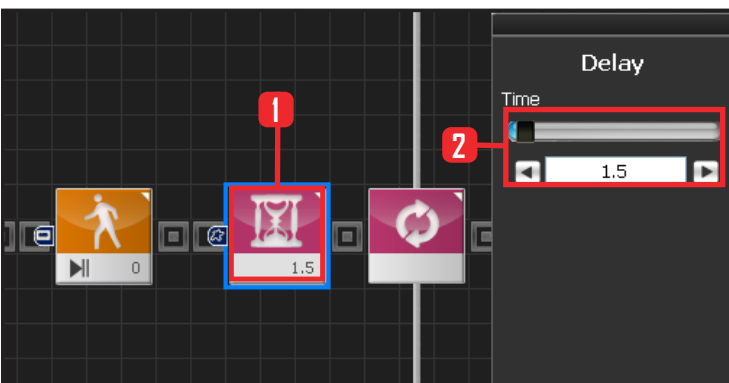
Motion > Move 모듈을 선택합니다.

Play/Stop : Play 를 선택합니다.

Motion Index : 0 번을 선택합니다. 0번은 전진하면서 걷기 모션입니다.

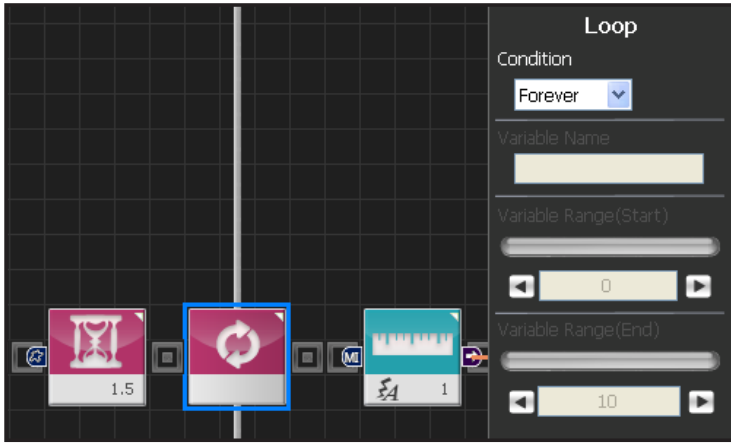
Motion Ready : True 를 선택합니다.

모션 동작 준비 과정입니다.



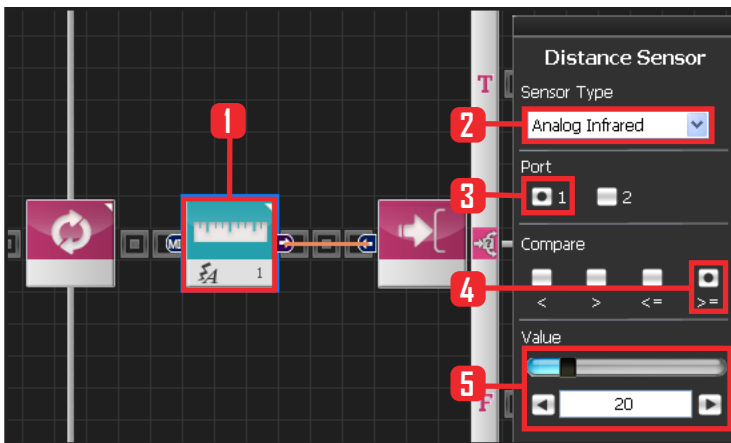
09 Delay

Motion Ready 동작이 끝나기 전에 진행하는 것을 방지하기위해 Delay 값을 1.5 초로 설정합니다.



10 반복

Forever 무한 반복을 설정합니다.



11 아날로그 센서 설정

20cm 기준으로 설정합니다.

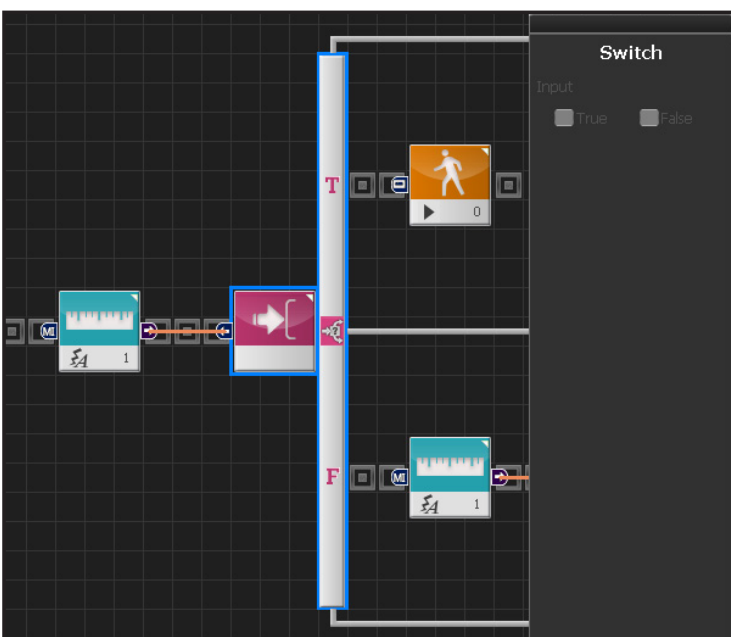
Sensor > Distance Sensor 모듈을 선택합니다.

Sensor Type : Analog Infrared 를 선택합니다.

Port : 1번을 선택합니다.

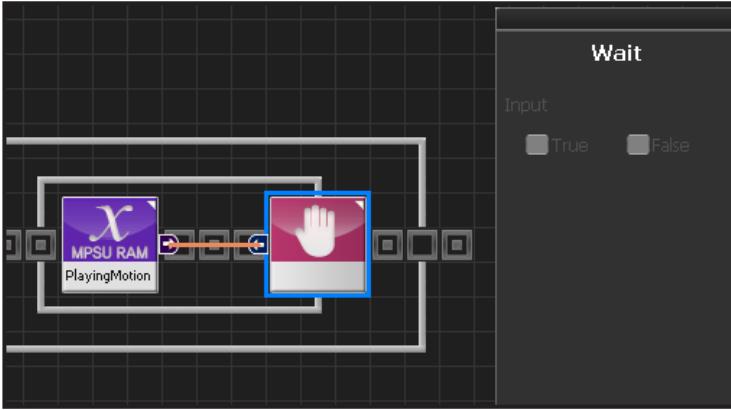
Compare : >= 를 선택합니다. 기준보다 같거나 높았을 때 True 입니다.

Value : 20 을 선택합니다. 20cm 를 의미합니다.



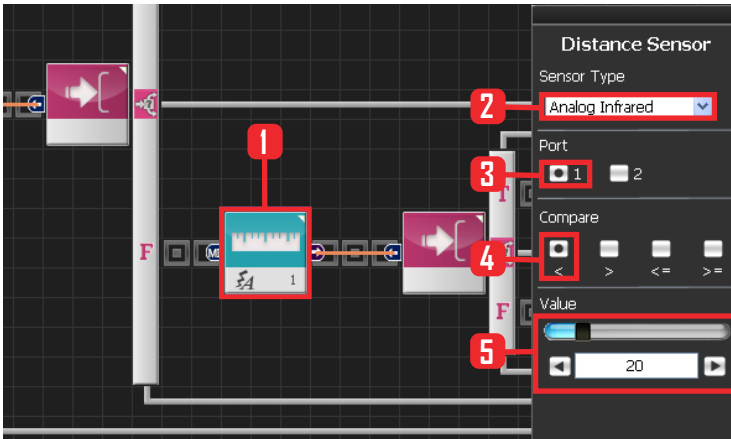
12 if 분기문

True 일때는 전진하고, False 일때는 다음 조건문으로 들어갑니다.



13 전진하기

20cm 보다 먼거리이기 때문에 로봇은 전진합니다. Motion Ready 값을 False 로 선택하면 로봇은 전진 모션을 실행합니다.

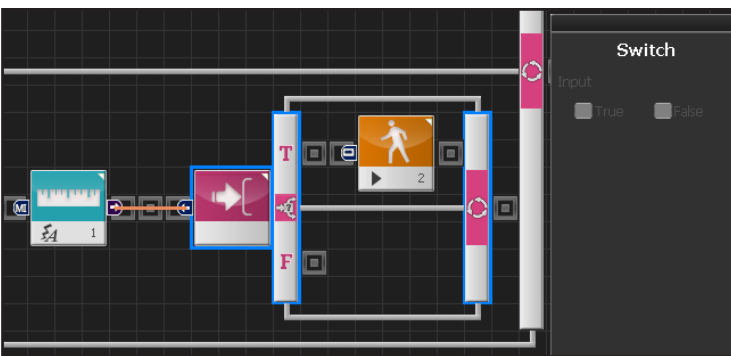


14 벽 근접시 동작

로봇이 벽과 20cm 이내로 근접시에는 좌회전 시킵니다.

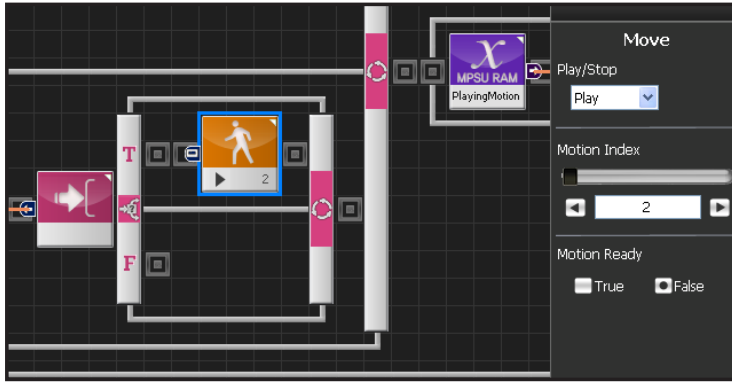
Sensor > Distance Sensor 모듈을 선택합니다. Sensor Type : Analog Infrared 를 선택합니다. Port : 1번을 선택합니다.

Compare: <를 선택합니다. 기준보다 낮을 때 True 입니다. Value : 20 을 선택합니다. 20cm 를 의미합니다.



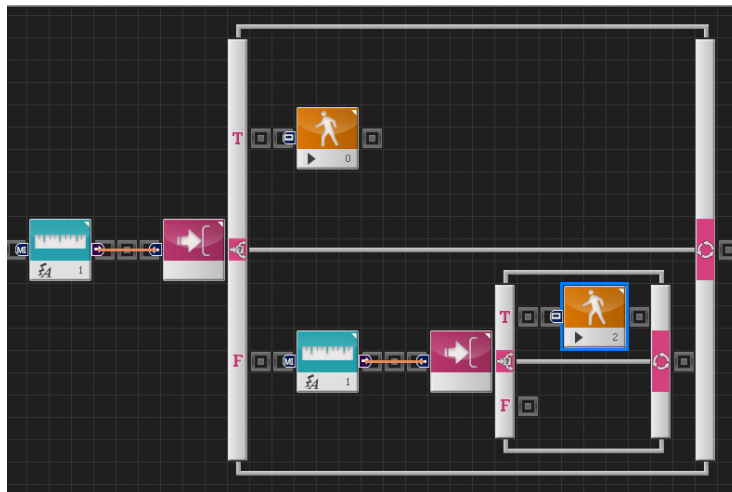
15 If 분기문

20cm 보다 더 가까울 때 True 안의 문을 실행시킵니다.



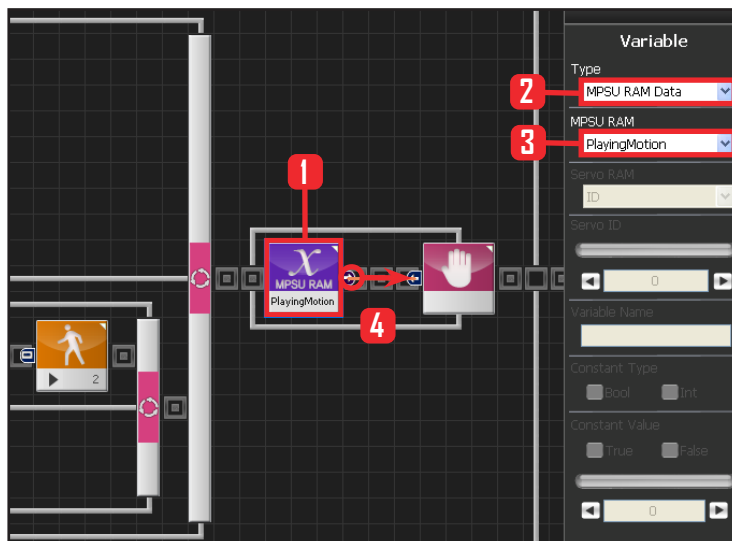
16 좌회전 시키기

Motion 2번은 좌회전 모션입니다.
Motion Ready 값이 False 이면 좌회전 모션을 실행합니다.



17 거리별 동작

20cm 보다 멀면 로봇의 계속 앞으로 전진하고, 20cm 보다 작으면 로봇의 좌회전 합니다.



18 모션 동작 확인

Loop 는 계속적인 반복을 의미합니다. Move 명령을 내리고 나서 실제 모션이 실행되어 완료되기 까지는 시간이 걸리므로 loop안에 Move모듈 하나만을 넣고 실행하면 모션을 이미 실행중임에도 loop를 계속 돌면서 모션실행 명령을 반복하게 됩니다.

이렇게 되면 Move모듈을 만난 횟수와 실제모션을 실행한 횟수가 달라집니다.

따라서 실행한 모션이 끝날때까지 기다렸다가 다시 loop의 처음으로 돌아가게 하는 편이 더 정확합니다.

Variable > MPSU RAM Data 에 들어가면 Playing Motion 이 있습니다. Playing Motion 은 로봇이 모션을 실행중인지 확인하는 변수입니다. 그 Playing Motion에 Wait 를 걸어주면 로봇의 동작이 끝날 때까지 Loop는 기다려줍니다.

Data > Variable 모듈을 선택합니다.

Type : MPSU RAM Data 를 선택합니다.

MPSU RAM : Playing Motion 을 선택합니다.

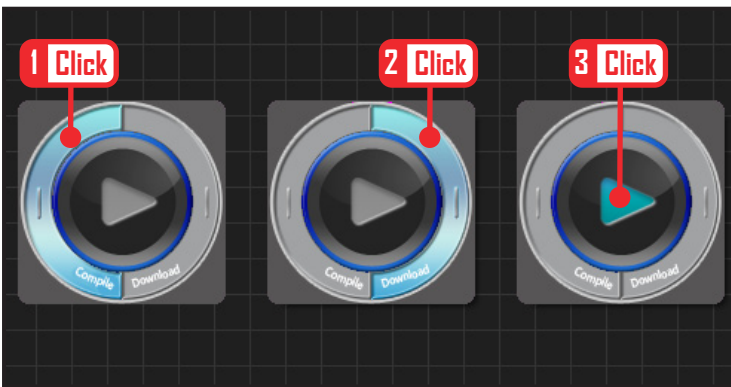
Output 커넥터 값을 뒤에 Wait 모듈에 연결합니다.

모션이 끝날 때까지 기다리겠다는 의미입니다.



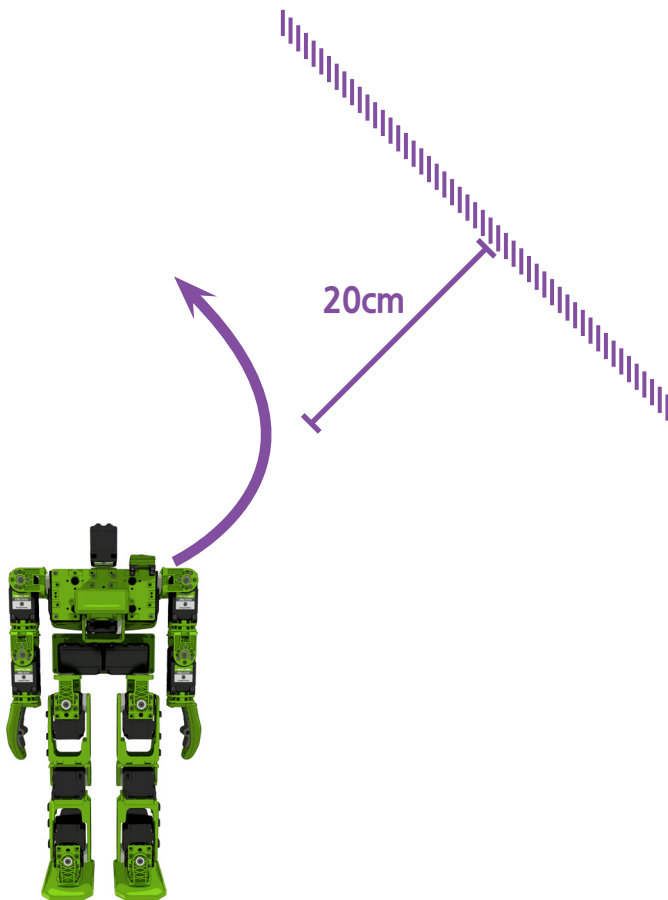
19 Wait

모션이 끝날때까지 기다립니다.
모션이 끝나면 다시 처음으로 돌아가 모션을 반복합니다.



20 컴파일, 다운로드, 실행

왼쪽 클릭하여 컴파일 시킵니다. 에러가 없으면 오른쪽 클릭하여 로봇에 다운로드 시킵니다. 다운로드 완료되면 가운데 화살표 실행버튼을 눌러 로봇에서 실행시킵니다.



21 로봇동작

로봇이 벽을 향해 전진하다가 20cm 안으로 근접하면 좌회전 합니다.