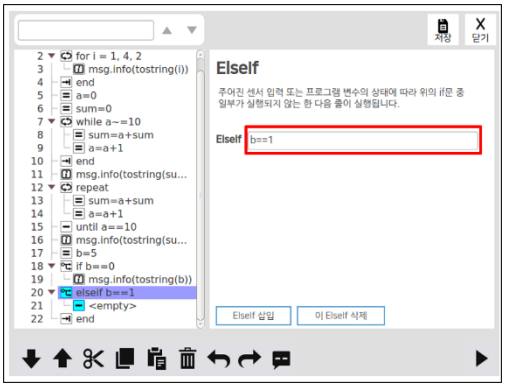
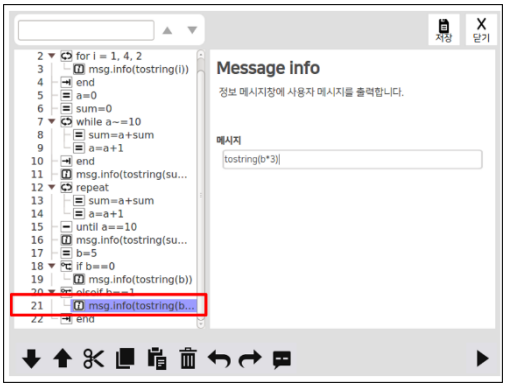
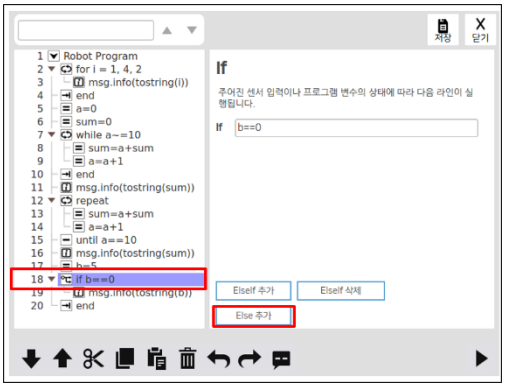
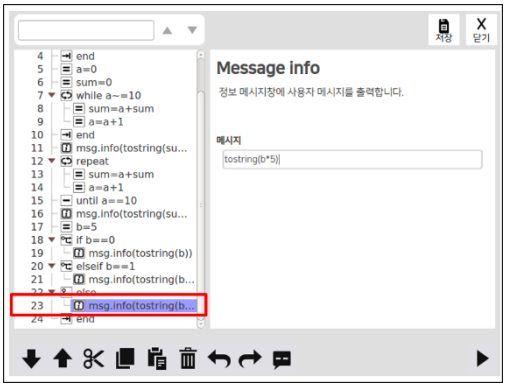



• Elseif 에 맞는 조건 입력이 되었으면 해당 조건일 때 실행하는 라인을 선택하여 구문을 여러 가지 명령어를 통해 입력합니다.

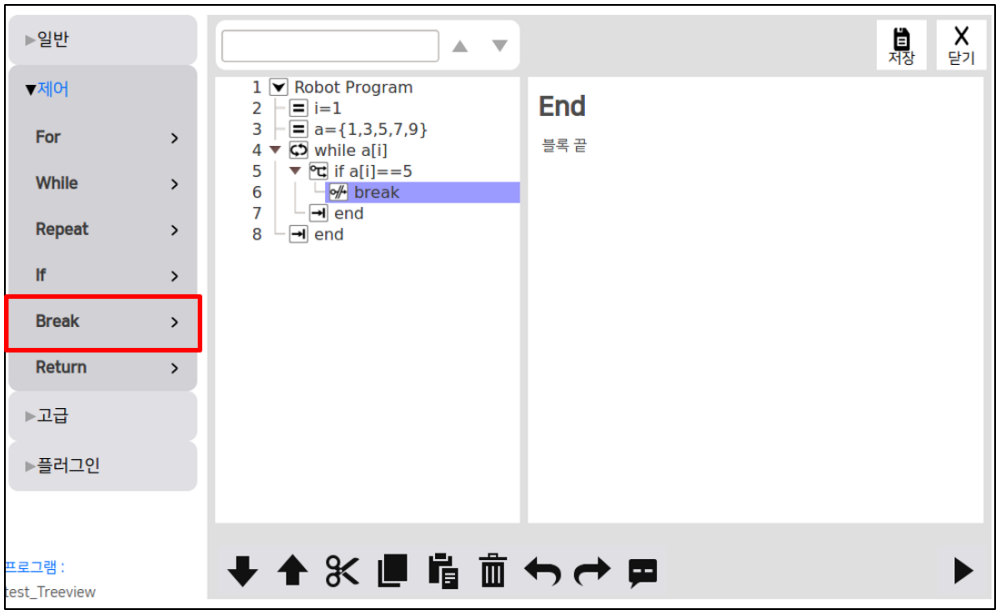



• Else 구문을 추가하여 If 와 Elseif 조건이 true 가 아닐 때 실행할 구문을 입력합니다.

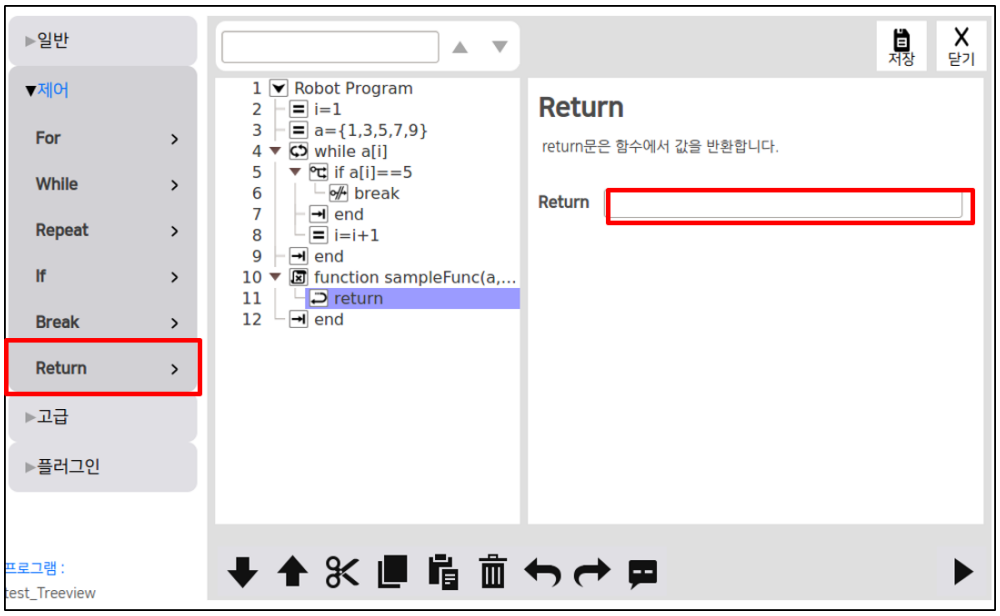
### 1.1.2.5. Break

- 블록에서 빠져나가고자 할 때 사용하는 명령어 입니다.
- 반복문을 종료할 때 사용합니다.
- Break 문이 실행되면 프로그램은 중지된 반복문 이후의 시점에서 실행을 계속합니다.

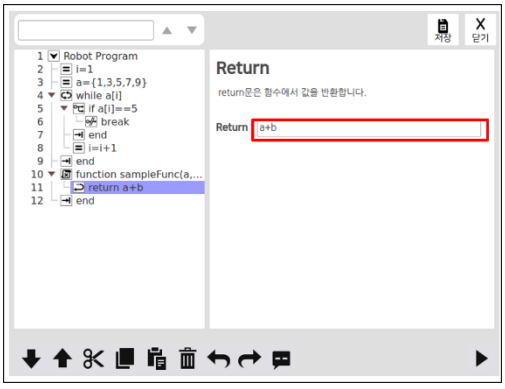


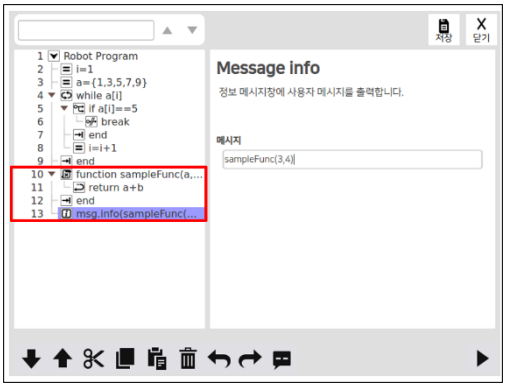
### 1.1.2.6. Return

- 블록에서 빠져나가고자 할 때 사용하는 명령어 입니다.
- Return 문은 함수로부터 결과를 반환할 뿐만 아니라 함수를 종료할 때 사용 가능합니다.
- 모든 함수의 끝에는 암시적인 return 문이 자연스럽게 있으므로 값을 반환하지 않는 함수 끝에는 return 을 사용할 필요가 없습니다.




- 함수를 추가하고 Return 명령어를 클릭하여 추가합니다.
- Return 문을 입력하기 위해 입력창을 클릭하고 키보드를 통해 입력합니다.



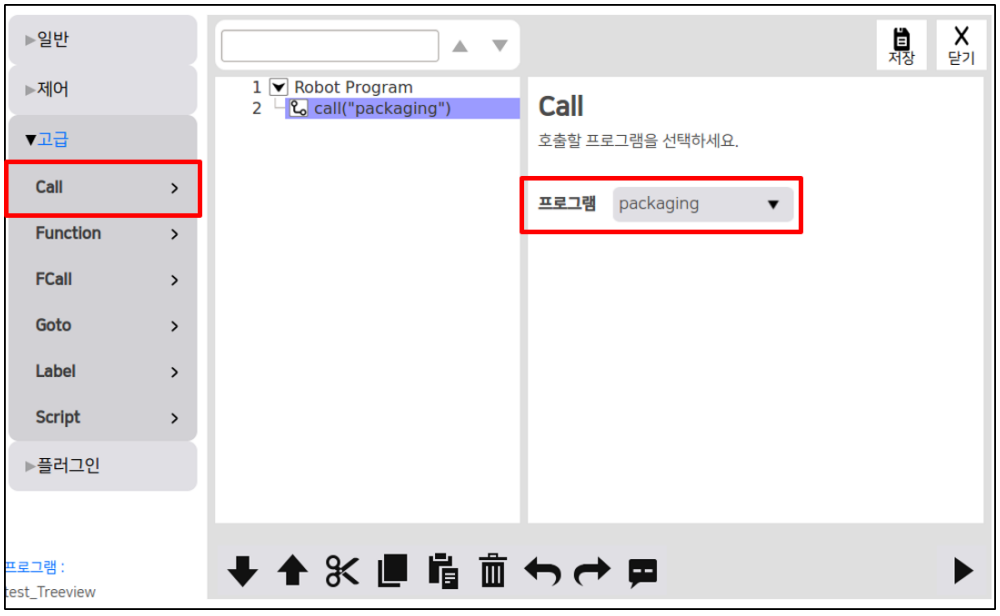

- Return 문을 입력하고 함수와 파라미터를 입력한 후 메시지창을 띄워서 Return 값을 확인할 수 있습니다.

### 1.1.3 고급

함수 생성과 다른 프로그램을 호출할 수 있는 명령어로 구성되어 있습니다.

#### 1.1.3.1. Call

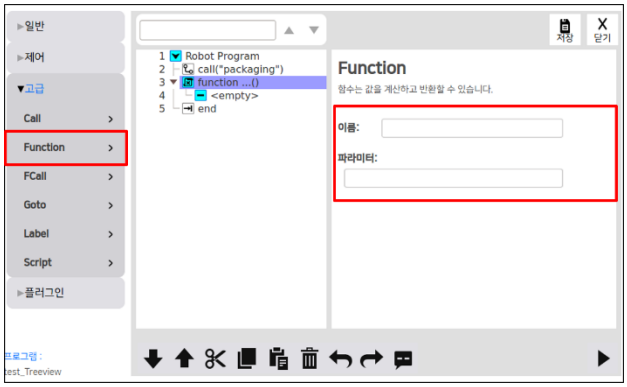
- 현재 작성하고 있는 프로그램 외 다른 프로그램을 호출하는 명령어 입니다.

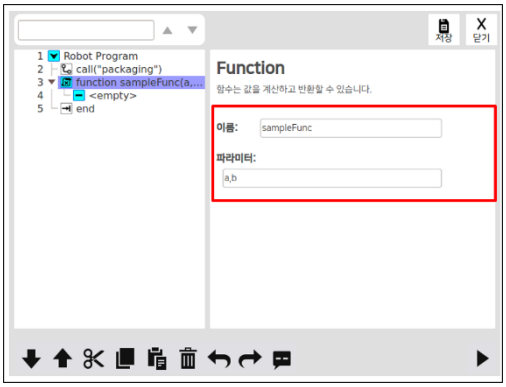



- Call 명령어를 클릭하면 트리뷰에 명령어가 추가되고 설정 화면이 나타납니다.
- 콤보 박스에 현재 작성되어 있는 프로그램들이 보여집니다. 그 중에서 호출할 프로그램을 선택합니다.

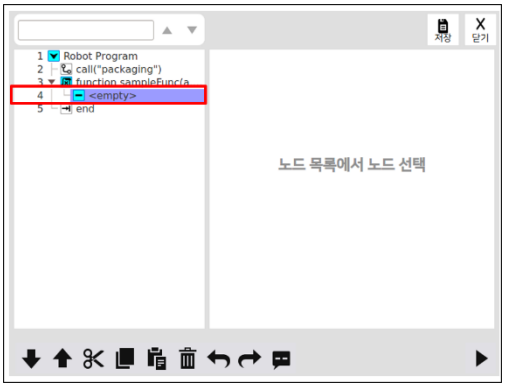
#### 1.1.3.2. Function

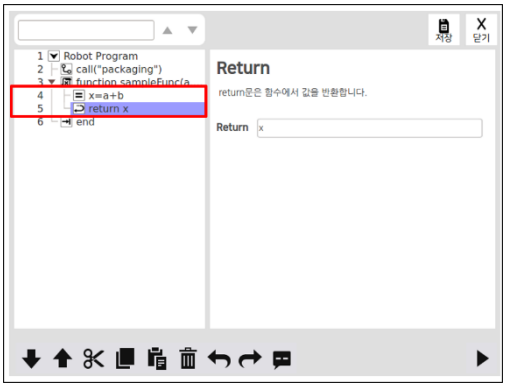
- 프로그램 내 함수를 생성하는 명령어 입니다.



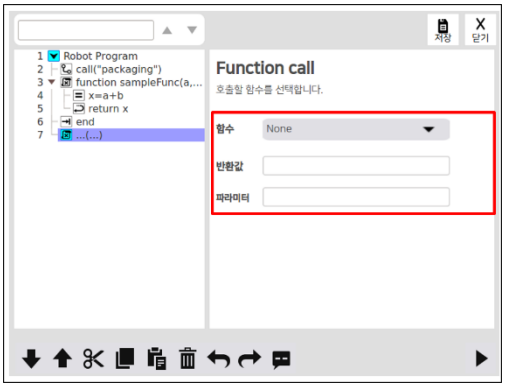

- Function 명령어를 클릭하면 트리에 function 함수 명령어가 추가됩니다.
- 이름과 파라미터에 각각 키보드를 통해 입력합니다. 반환할 함수 내 구문을 입력합니다.

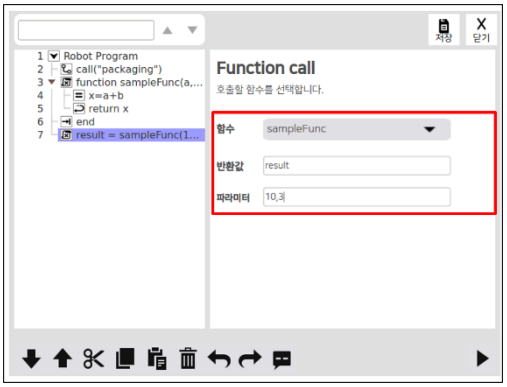



### 1.1.3.3. Function call

- 프로그램 내 함수를 호출하는 명령어 입니다.

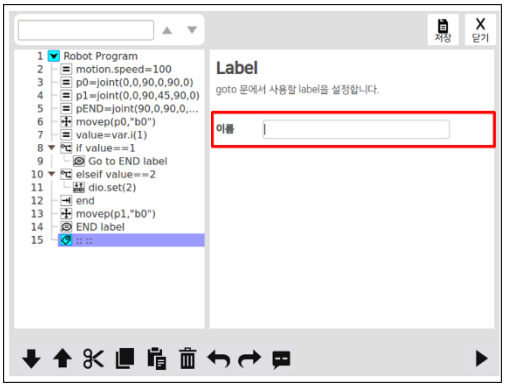


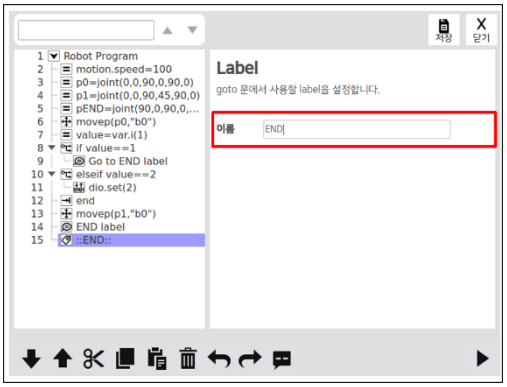




- FCall 명령어를 클릭하면 트리에 Function call 함수 명령어가 추가됩니다.
- 프로그램 내에 있는 함수를 호출하므로 입력되어 있는 함수가 없으면 콤보박스에 나타나지 않습니다.
- 키보드를 통해 호출할 함수의 반환값과 파라미터를 입력합니다.

### 1.1.3.4. Label

- Goto 문에서 사용할 label 을 설정하는 명령어 입니다.

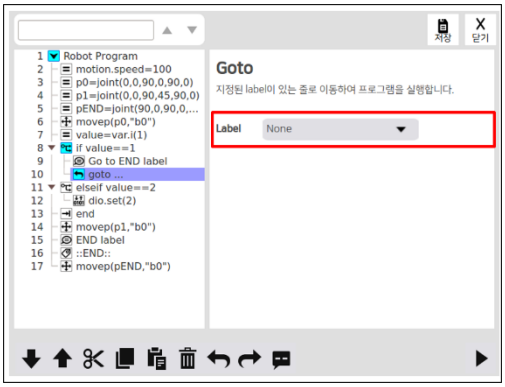


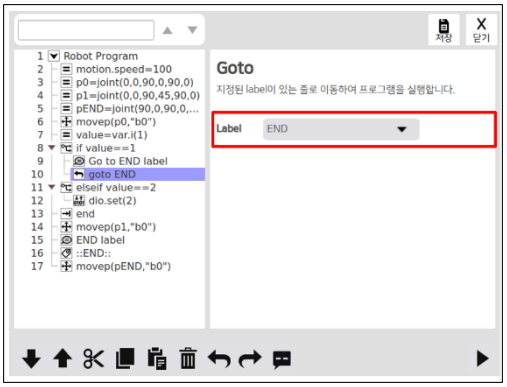




- Label 명령어를 클릭하면 트리에 label 함수 명령어가 추가됩니다.
- 키보드를 통해 Label 을 입력합니다. 입력한 Label 은 '::입력값::' 형태로 입력됩니다.

### 1.1.3.5. Goto

- 지정될 label 이 있는 줄로 이동하여 프로그램을 실행하는 명령어 입니다.

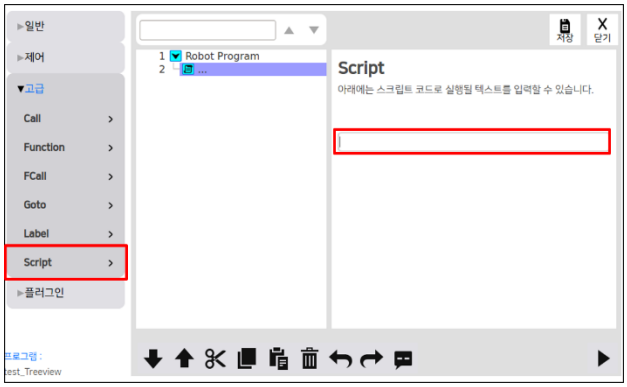


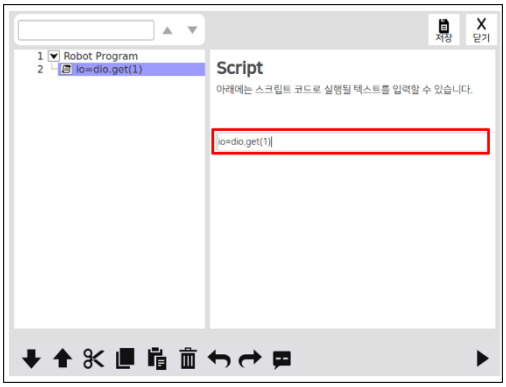




- Goto 명령어를 클릭하면 트리에 goto 함수 명령어가 추가됩니다.
- 콤보박스를 통해 프로그램 내에 입력되어 있는 label 을 선택합니다.

### 1.1.3.6. Script

- 스크립트 코드로 실행될 텍스트를 입력할 수 있는 명령어 입니다.

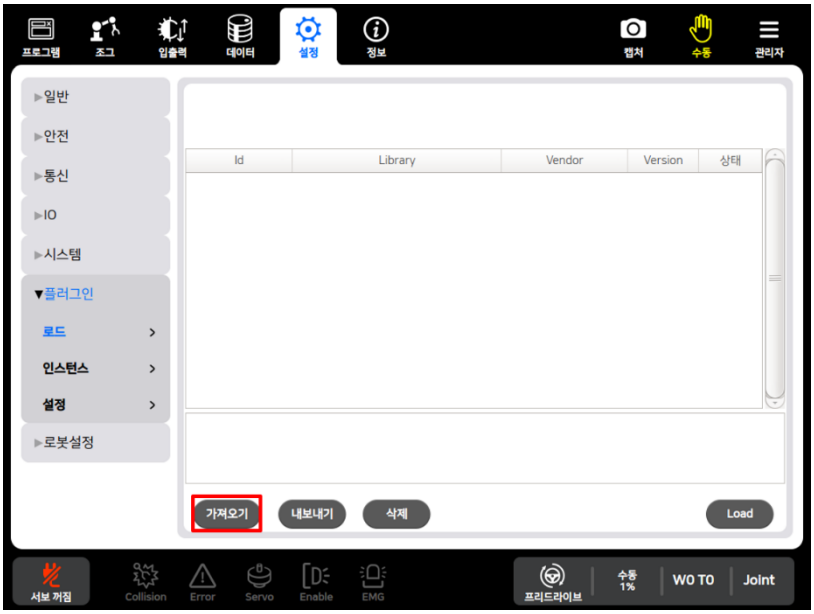


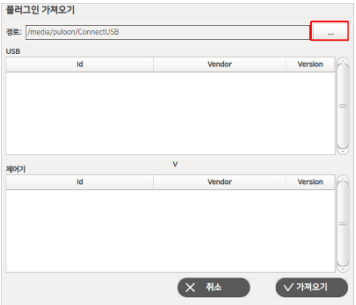

- 키보드를 통해 텍스트를 입력합니다.

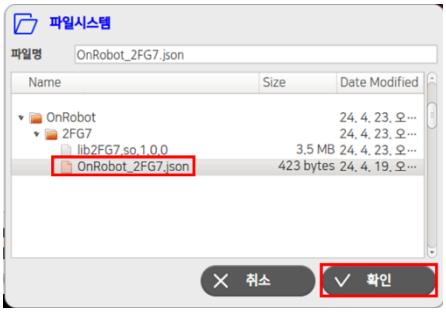
### 1.1.4 플러그인

- 플러그인 명령어를 사용하기 위해서 메인 메뉴 '설정'에서 플러그인 메뉴에서 플러그인 인스턴스를 생성해야합니다.
- 순서는 플러그인을 로드→ 인스턴스 생성을 위해 초기 설정→ 인스턴스를 생성입니다.

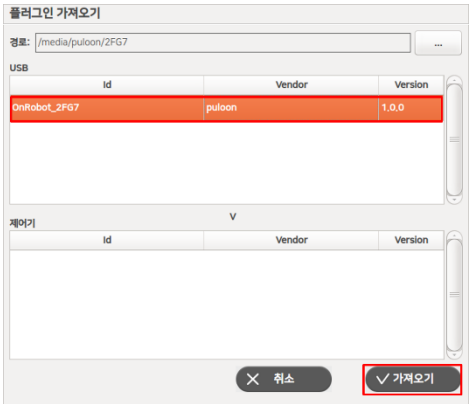



- '설정' 메뉴의 '가져오기' 버튼을 통해 플러그인 파일을 가져옵니다.

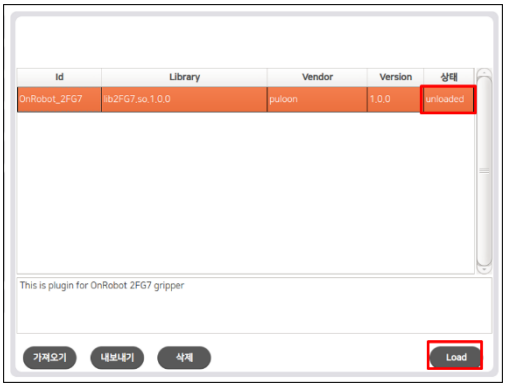


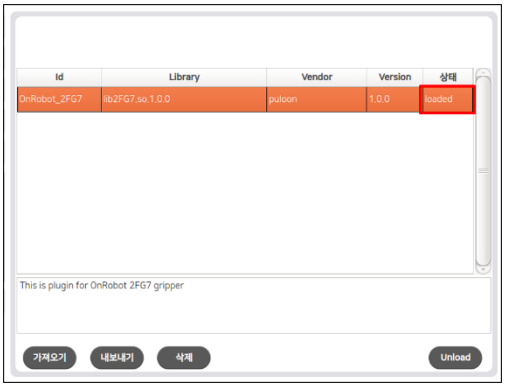




- 플러그인 가져오기에서 '...' 버튼을 클릭하고 플러그인 파일(*.json)을 선택한 후 '확인' 버튼을 클릭합니다.

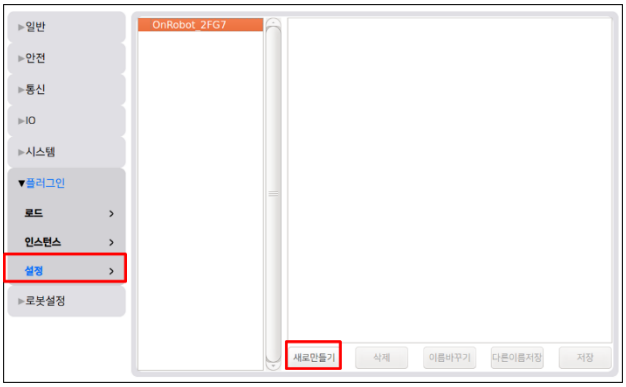


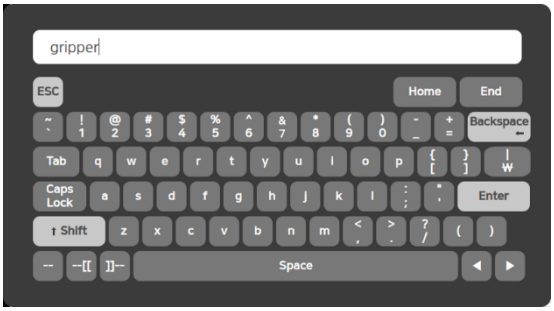

- 플러그인 가져오기 화면에 선택한 플러그인 파일이 추가되면 그 파일을 선택하고 하단에 '가져오기' 버튼을 클릭합니다.



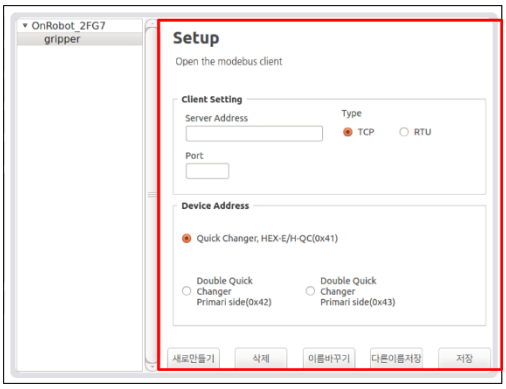



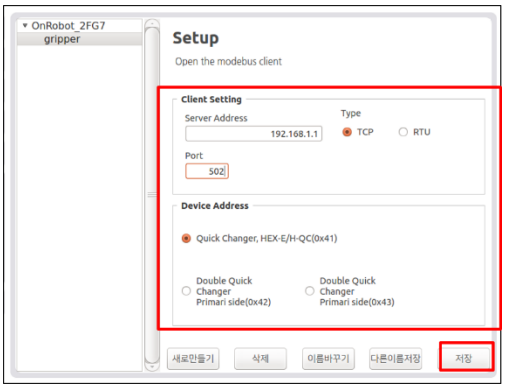

- 가져오기가 완료된 플러그인 라이브러리를 사용하기 위해 하단의 'Load' 버튼을 클릭합니다.
- 정상적으로 로드가 되면 플러그인의 상태가 'loaded'로 변경됩니다.



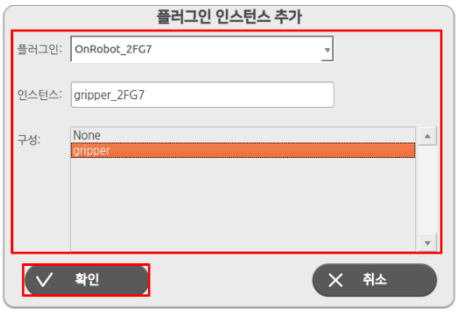




- 플러그인 인스턴스 생성을 위해서 플러그인의 초기 설정이 필요합니다. 플러그인 메뉴의 설정 화면으로 이동합니다. 설정 화면에 앞서 Load 한 플러그인이 리스트에 표시되며, '새로만들기' 버튼을 클릭하여 새로운 플러그인 설정을 생성합니다.

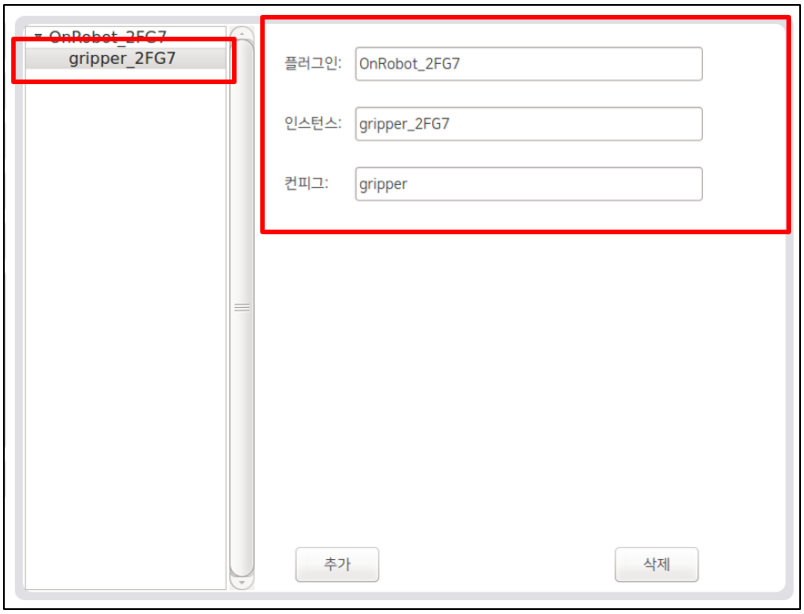


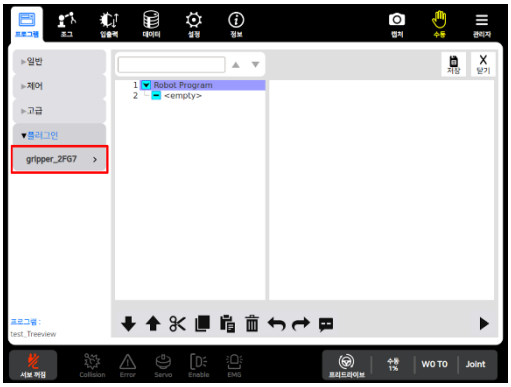

- 서버 주소, 유형, 포트 번호 등 관련 설정 정보들을 입력한 후 '저장'버튼을 클릭합니다.

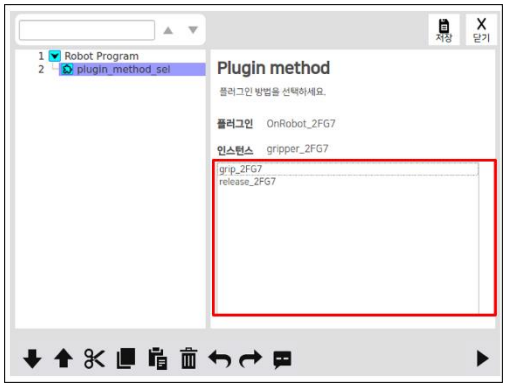


- 실제로 트리뷰 프로그램을 작성할 때 쓰일 플러그인 객체를 생성합니다.
- 초기 설정까지 저장했으면 플러그인 인스턴스를 생성하기 위해 '인스턴스' 화면으로 이동합니다. '추가' 버튼을 클릭하면 플러그인 인스턴스 추가 화면이 팝업됩니다. 플러그인 항목과 앞서 설정에서 저장한 플러그인 설정 정보를 선택하고 인스턴스 이름도 입력합니다. 모든 항목을 입력했으면 '확인' 버튼을 클릭합니다.

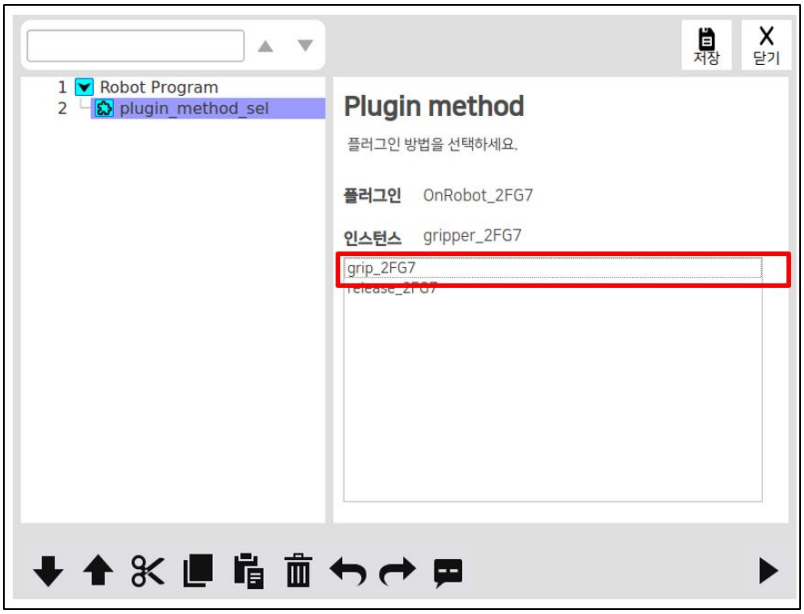



- 입력한 설정값대로 인스턴스 생성이 된 것을 확인할 수 있습니다.

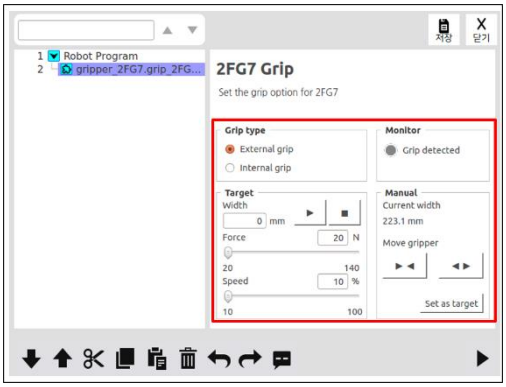


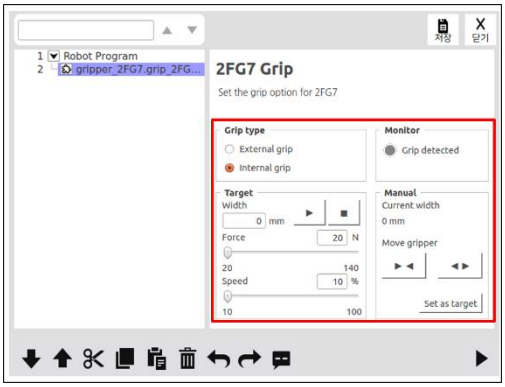




- 메인 메뉴 '프로그램'으로 이동하여 플러그인 명령어를 클릭합니다. 설정 메뉴에서 생성한 인스턴스의 이름이 표시됩니다. 해당 인스턴스를 클릭하면 플러그인이 제공하는 명령 목록이 표시되며 선택한 명령어에 따른 설정 화면이 표시됩니다.



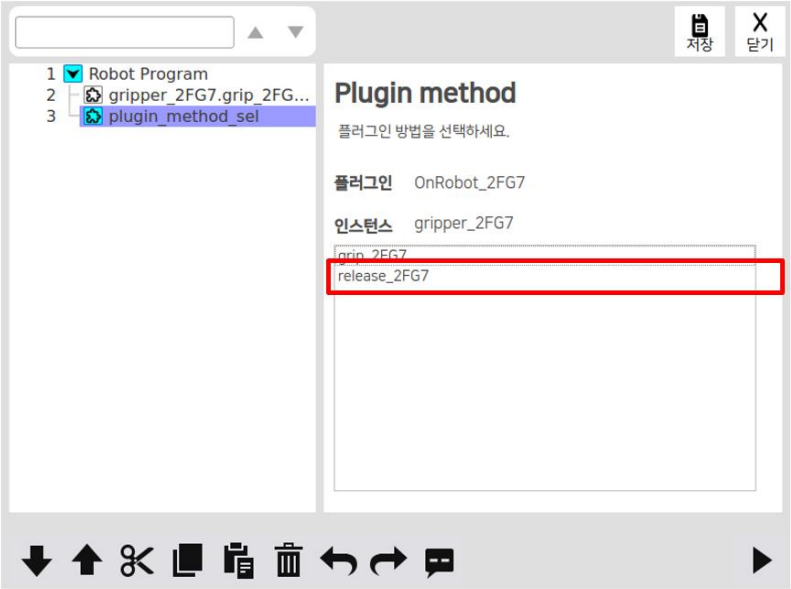

• 우선 'grip' 명령어를 선택합니다.



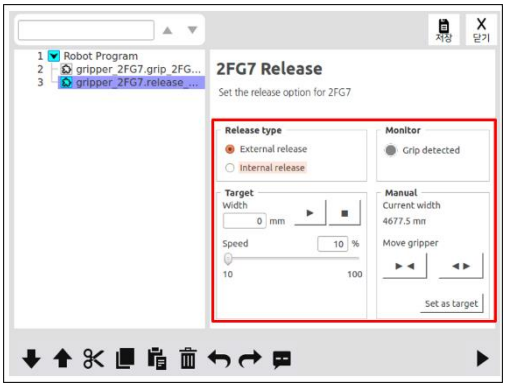

• 'grip' 옵션에 설정할 화면이 팝업됩니다. 그립 유형, 목표값 등 설정 값을 입력합니다.

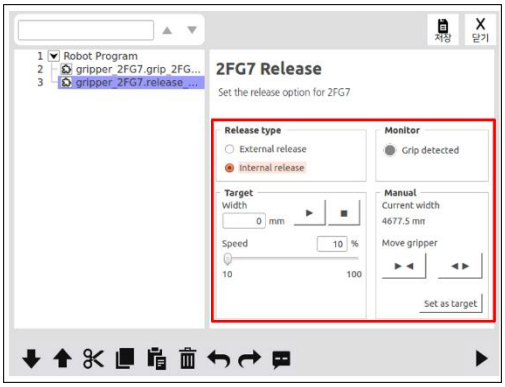
• 입력이 완료되면 플러그인 노드가 하늘색에서 하얀색으로 변경됩니다.




• 좌측의 명령어에서 인스턴스 명령어를 다시 클릭하여 트리뷰에 명령어를 추가합니다.

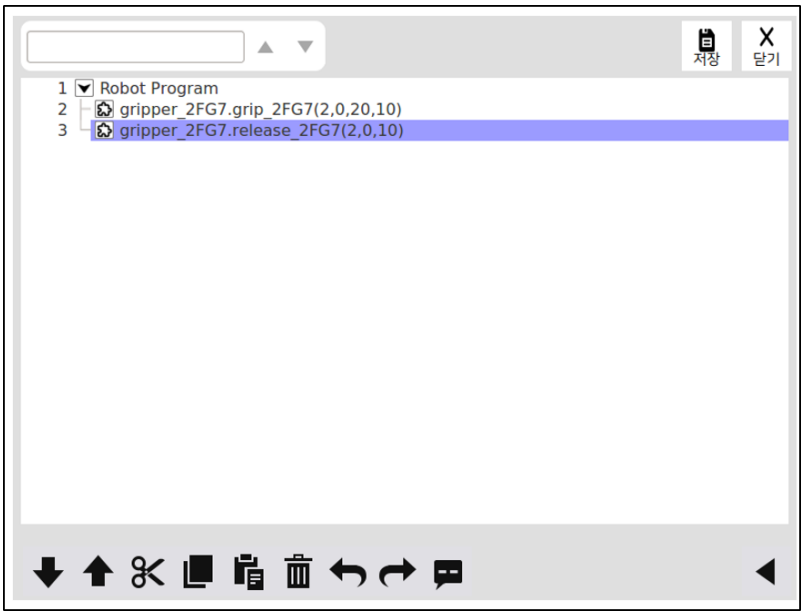
• ‘release’ 명령어를 선택합니다.




• ‘release’ 옵션에 설정할 화면이 팝업됩니다. 릴리즈 유형, 목표값 등 설정 값을 입력합니다.

• 입력이 완료되면 플러그인 노드가 하늘색에서 하얀색으로 변경됩니다.



# 2 스크립트 프로그램

## 2.1. 협동로봇 스크립트 언어

### 2.1.1. 기본 프로그램 구조

협동로봇 제어기는 인터프리터 언어인 **LUA언어(버전 5.1.4)**로 설계되어 있습니다. 따라서, PRL(Puloon Robot Language)은 LUA와 동일한 문법을 사용하지만, 모든 기능을 포함하지는 않습니다. 따라서, 본 매뉴얼을 따르지 않을 경우 오류가 발생할 수 있습니다.

※ PRL은 스크립트 저장 시 **정적검사와 동적검사를 수행**하도록 설계되어 있습니다.

※ 정적검사는 함수의 매개변수의 **인자 입력 개수, 인자의 유형(Type)**등을 검사합니다.

※ 동적검사는 함수의 매개변수에 대하여 **최대/최소 값**을 검사합니다.

PRL은 아래와 같은 구조로 작성되기를 바라지만, 이 방법은 필수적이지 않습니다.

① **전역변수 초기화**: 프로그램에서 사용되는 전역변수를 초기화 합니다.

```
-- Global variables define

-- Integer Variable
a = 3
b = 5

-- Float Variable
mathPI = 3.1415926

-- String Variable
stringVar = "Hello"

-- Table Variable
exArray = {}
for i = 1, 3 do
        exArray[i] = 0
end
```

② **함수 선언**: 프로그램에서 사용되는 함수를 작성합니다.

```
-- Global function defines
```

```
function sum_two_number(a, b)
    return a + b
end
```

③ **메인 프로그램 작성**: 스크립트가 실행할 메인 프로그램을 작성합니다.

```
movep(lp0)
movep(lp1)
c = sum_two_number(a, b)
msg.info(tostring(c))
movel(lp2)
movel(lp3)
```

※ lp 는 local point 를 의미합니다.

### 2.1.2. 프로그램 실행 순서

PRL은 인터프리터 언어로 코드를 한 줄씩 **순차적으로 실행하는 언어**입니다. 그림 2-1과 같은 구조로 스크립트가 작성될 경우 실행순서는 다음과 같습니다.

프로그램 실행 순서: ① → ② → ③ → ① → ④

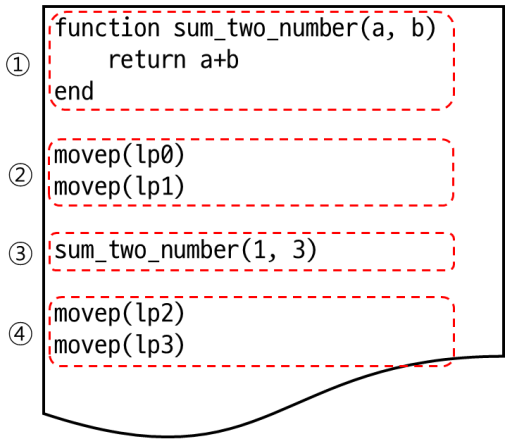



**그림 2-1 순차적 실행 구조**

### 2.1.3. 기본 문법

PRL은 쉽고 간결하며 문법 체계를 가진 프로그래밍 언어입니다. 따라서, 초보자도 쉽게 익힐 수 있다는 장점을 가지고 있으며, 널리 사용되는 파이썬과 비교하여도 문법이 더 간결합니다.

※ 인터프리터 언어로 용량이 작고 빠릅니다.

※ 절차지향적, 객체지향적, 함수형으로 스크립트를 구성할 수 있습니다.

#### 2.1.3.1. 청크(Chunk)

청크(Chunk)는 순차적으로 실행되는 명령이나 문장의 단위로, **프로그램의 최소 실행 단위**를 의미합니다. 연속되는 명령문 사이에 구분자를 쓰지 않아도 되나, 프로그램의 가독성을 위해 세미콜론(;)을 사용할 수 있습니다.

| | |
|---|---|
| a = 1<br>b = a*2 | 가능 |
| a = 1;<br>b = a*2; | 가능 |
| a = 1; b = a*2 | 가능 |
| a = 1 b = a*2 | 가능(가독성 저하) |

**표 2-1 청크 구분 예제**

### 2.1.3.2. PRL의 자료형

PRL은 아래와 같이 8개의 기본 자료형을 가집니다.

| NO | 타입 | 설명 |
|---|---|---|
| 1 | Nil | 의미가 없음을 표기위해 사용됩니다.<br>초기값을 정의하지 않고 변수 생성시 해당 변수는 nil값으로 설정됩니다.<br>변수가 nil 값으로 지정되면, 존재하지 않더라도 소멸할 수 있습니다. |
| 2 | Boolean | Boolean 자료형은 참, 거짓을 구분하기 위한 자료형으로 사용됩니다. |
| 3 | Number | Number 자료형은 부동소수점 실수를 표현하기 위해 사용됩니다. |
| 4 | String | 문자열은 8비트(1 바이트) 문자들의 배열입니다.<br>모션 명령에 옵션을 넣거나, 수신된 통신 데이터를 저장하기 위해 사용됩니다.<br>문자열 선언 방법은 2.6.9 절의 수치 및 문자열 선언을 참조바랍니다. |
| 5 | Table | 배열, 해쉬 테이블과 같습니다.<br>PRL의 가장 강력한 자료형으로 정수로 테이블을 인덱스 할 수 있으며, 어떠한 자료형의 키를 사용하여 테이블로 인덱스 할 수 있습니다. |
| 6 | Function | PRL에서는 함수 또한 하나의 자료형이며, 변수로 지정할 수 있습니다. |
| 7 | UserData | 사용자 데이터 자료형은 PRL 변수가 C/C++ 데이터를 저장할 수 있도록 제공됩니다. |
| 8 | Thread | 별도의 프로그램을 동시에 구동할 수 있도록 제공됩니다. |

**표 2-2 PRL의 기본 자료형**

### 2.1.3.3. 변수명

변수(식별자 또는 변수의 이름)는 자료의 값을 나타내기 위하여 **알파벳, 숫자, 밑줄("_")로 구성**할 수 있습니다.

※ 단, 변수명의 첫 글자는 숫자가 불가합니다.

※ 변수명은 대/소문자를 구분합니다.

※ 변수명으로 예약어를 사용할 경우 에러가 발생합니다.
(대/소문자를 구분하므로 and는 사용 불가하지만, And, AND는 사용 가능합니다.)

| | | | | |
|---|---|---|---|---|
| and | break | do | else | elseif |
| end | false | for | function | if |
| in | local | nil | not | or |
| repeat | return | then | true | until |
| while | | | | |

**표 2-3 PRL 예약어**

PRL은 **동적 자료형(dynamically-typed)** 언어입니다. 따라서 어떠한 자료형으로도 설정될 수 있습니다.

변수를 local 예약어 없이 사용할 경우 명시적으로 전역(global)변수로 선언됩니다.
(단, 함수의 인수나 for의 초기값은 지역(local) 변수입니다.)

### 2.1.3.4. 주석

스크립트 언어에서는 코드를 설명하기 위한 주석을 지원합니다. 주석은 실제 프로그램의 실행 시 제외됩니다.

※ 블록주석의 시작을 "---[["로 변경할 경우 --로 시작하는 한 줄 주석이 되므로 첫 줄과 끝 줄은 서로 관련 없는 한 줄 주석이 됩니다.

| | |
|---|---|
| **한줄주석** | 행의 처음에 "--"을 입력합니다. |
| | -- msg.info(tostring(10)) |
| **블록주석** | 블록의 처음에 "--[[" 블록의 마지막에 "]]--"을 입력합니다. |
| | --[[<br>msg.info(tostring(10))<br>]]-- |

**표 2-4 주석 예제**

| | | |
|---|---|---|
| **잘못된 예 1)** | 스크립트 | -[[<br>msg.info(tostring(10))<br>]]-- |
| | 에러 | unexpected symbol near '-'<br>unexpected symbol near ']' |
| **잘못된 예 2)** | 스크립트 | ---[[<br>msg.info(tostring(10))<br>]]-- |
| | 에러 | unexcepted symbol near ']' |
| **잘못된 예 3)** | 스크립트 | --[[<br>msg.info(tostring(10))<br>]]- |
| | 에러 | unexcepted symbol near '-' |

**표 2-5 블록 주석의 잘못된 예**

### 2.1.3.5. 자료형(테이블)

테이블은 배열, 집합, 레코드 등 다양한 자료구조로 테이블의 인덱스는 nil을 제외한 모든 값을 쓸 수 있는 배열을 말합니다. (PRL에서는 **인덱스가 1부터 시작**합니다.) 스크립트에서는 유일하게 테이블을 이용해 자료구조를 구성할 수 있습니다.
테이블의 크기는 동적으로 늘어나게 설계되어 있으므로 크기를 설정할 수 없으며, 원하는 사이즈만큼 초기화는 할 수 있습니다.

※ 테이블을 초기화 하지 않고 사용할 경우 "attempt to index global 'variableName' (a nil value)" 에러가 발생합니다.

**예 1) 1차원 배열 생성**

```
-- 1 차원 배열을 생성하는 예제입니다.
maxSize = 4

exArray = { }

for i = 1, maxSize do
        exArray[i] = 0
        msg.info("exArray["..i..."] 값: ".. tostring(exArray[i]))
end
```

```
>> exArray[1] 값: 0
>> exArray[2] 값: 0
>> exArray[3] 값: 0
>> exArray[4] 값: 0
```

**예 2) 2차원 배열 생성**

```
-- 2 차원 배열(행렬)을 생성하는 예제입니다.
maxSize = 3

array2D = { }

for i = 1, maxSize, 1 do
    array2D[i] = {}
    for j = 1, maxSize, 1 do
        array2D[i][j] = (i-1)*maxSize + j
      msg.info(("Array["..i..."]["..j..."] = ".. tostring (array2D[i][j]))
    end
end
```

```
>> Array[1][1] = 1
>> Array[1][2] = 2
>> Array[1][3] = 3
>> Array[2][1] = 4
>> Array[2][2] = 5
>> Array[2][3] = 6
>> Array[3][1] = 7
>> Array[3][2] = 8
>> Array[3][3] = 9
```

**예 3) Queue**

```
-- Table 을 이용하여 Queue 를 구현하는 예제입니다. Queue 는 선입선출(FIFO, First in First Out)의 자료 구조로, Add 와 Delete 로 추가 및 출력 합니다.
-- Queue 구현을 위한 head/tail point 지정
function QueueNew()
        return { head = 0, tail = 0 }
end

-- Queue 에 Value 를 삽입하는 함수
function AddQueue(queue, value)
        local tailPoint = queue.tail
        queue[tailPoint] = value
        queue.tail = tailPoint + 1
end

-- Queue 에서 하나의 value 를 빼내는 함수
function DeleteQueue(queue)
        if (queue.head == queue.tail) then
                print("queue is empty")
                return
        end

        local headPoint = queue.head
        local retValue = queue[headPoint]
        queue[headPoint] = nil
        queue.head = headPoint + 1
        return retValue
end

-- Queue 를 하나 선언
local queue = QueueNew()

msg.info(queue.head)
msg.info(queue.tail)

-- value 가 없으므로, empty
DeleteQueue(queue)
```

```
-- Value 를 삽입하는
AddQueue(queue, 10)
AddQueue(queue, 20)
AddQueue(queue, 30)

-- FIFO 에 맞게 value 가 나오는지 확인
msg.info(DeleteQueue(queue))
msg.info(DeleteQueue(queue))
msg.info(DeleteQueue(queue))
```

```
>> 0
>> 0
>> queue is empty
>> 10
>> 20
>> 30
```

**예 4) Stack**

```
-- Table 을 이용한 Stack 구현 예제입니다. Stack 은 후입선출(LIFO, Last in First Out)의
자료 구조로, 입력을 Push, 출력을 Pop 이라고 합니다.
-- Stack 을 생성해 주는 함수
function NewStack()
        return { top = 0 }
end

-- Push 함수
function PushStack(stack, value)
        local newTop = stack.top + 1
        stack[newTop] = value
        stack.top = newTop
        msg.info("Push Stack: " .. stack.top)
end

-- Pop 함수
function PopStack(stack)
        if stack.top == 0 then
                msg.info("Stack is empty")
                return "Stack is empty"
        end

        local oldTop = stack.top
        local retValue = stack[oldTop]
        stack.top = oldTop - 1
        stack[oldTop] = nil

        return retValue
end

-- Stack 을 하나 선언
local stack = NewStack()

-- value 삽입
PushStack(stack, 10)
PushStack(stack, 20)
PushStack(stack, 30)
```

```
-- FIFO 에 맞게 value 가 나오는지 확인
msg.info("Pop Stack : " .. PopStack(stack))
msg.info("Pop Stack : " .. PopStack(stack))
msg.info("Pop Stack : " .. PopStack(stack))

-- Stack 이 empty 일 경우
msg.info("Pop Stack : " .. PopStack(stack))
```

```
>> Push Stack: 1
>> Push Stack: 2
>> Push Stack: 3
>> Pop Stack : 30
>> Pop Stack : 20
>> Pop Stack : 10
>> Stack is empty
>> Pop Stack : Stack is empty
```

**예 5) Linked List**

```
-- Table 을 이용한 Linked List 구현 예제입니다. Linked List 는 리스트의 항목들을 노드에 분산하여 저장하는 리스트로, 데이터의 삽입과 삭제가 용이하며, 연속된 메모리 공간이 필요 없고 크기 제한이 없다는 점이 있습니다.
-- LinkedList 의 first 노드를 만듬
function newLinkedList()
        return { next = nil, data = nil }
end

-- 원하는 노드 뒤에 새로운 값의 노드를 추가
function Add(linkedList, value)
        linkedList.next = { next = linkedList.next, data = value}
end

local linkedList = newLinkedList()

-- head 에 1, 2 값을 입력
Add(linkedList, 1)
Add(linkedList, 2)

-- LinkedList 를 위한 iterator
local iter = linkedList.next
```

```
-- 데이터가 1 인 노드 뒤에 새로운 노드 추가
while iter do
        if iter.data == 1 then
                Add(iter, 3)
                break
        end

        iter = iter.next
end

-- 전체 LinkedList 데이터 출력

iter = linkedList.next

while iter do
        msg.info(tostring(iter.data))
        iter = iter.next
end
```

```
>> 2
>> 1
>> 3
```

**표 2-6 PRL 테이블 사용 예제**

### 2.1.3.6. 수치 및 문자열

1) 수치 (확인 필요)

PRL은 64bit 크기에 해당하는 범위의 **정수 및 실수 값**을 사용할 수 있습니다.

※ 정수: -9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807

※ 실수: 2.225074-308 ~ 1.797693e+308

2) 문자열

- PRL에서 문자열은 바이트의 연속(Sequence)입니다. 바이트를 텍스트로 인코딩 하는 방식에 대하여 특정한 방식을 취하지는 않으나, **UTF-8인코딩을 권장합니다.**

※ 문자열을 만들 때는 **큰 따옴표(" ")와 작은 따옴표(' ')를 모두 사용**할 수 있습니다.

※ 긴 문자열은 **"[[ --내용-- ]]""을 이용해 생성**할 수 있습니다.

※ 문자열 연산: 문자열을 연결할 때는 ".."연산자를 사용할 수 있습니다.

| | |
|---|---|
| 문자열 연결 | stringVal1 = "Hello "<br>stringVal2 = "World!"<br><br>msg.info(stringVal1 .. stringVal2) |
| | >> Hello World! |
| 문자열과 숫자 연결 | stringVal1 = "Hello "<br>numVal1 = **1004**<br><br>msg.info(stringVal1 .. numVal1) |
| | >> Hello 1004 |
| 문자열 숫자와<br>문사열 숫자의 연결 | stringNum1 = "1000"<br>stringNum2 = "4"<br><br>msg.info(stringNum1 .. stringNum2) |
| | >> 10004 |
| 문자열 숫자와<br>문자열 숫자의 연산 | stringNum1 = "1000"<br>stringNum2 = "4"<br><br>msg.info(tostring(stringNum1 + stringNum2)) |
| | >> 1004 |

**표 2-7 문자열 연결의 예**

| 예 1) 문자열 선언 | |
|---|---|
| string = "Hello"<br><br>msg.info(string1) | >> Hello |
| **예 2) 긴 문자열 선언** | |
| string = [[<br>Long string<br><html><br><body><br>Hello<br></body><br>>/html><br>]]<br>msg.info(string) | >> Long string<br>>> <html><br>>> <body><br>>> Hello<br>>> </body><br>>> </html> |
| **예 3) 문자열 연결** | |
| *-- 문자열 연결*<br><br>string1 = "This is"<br>string2 = " a book"<br><br>string3 = string1 .. string2<br><br>msg.info(string3)<br><br>*-- 문사열숫자 연결*<br><br>string3 = "11"<br>string4 = "22"<br><br>msg.info(string3 .. string4)<br><br>*-- 문자열숫자와 숫자 연산*<br><br>msg.info(tostring(string3 + **44**)) | >> This is a book<br>>> 1122<br>>> 55.0 |

**표 2-8 문자열 예제**

### 2.1.3.7. 연산자

연산자의 왼쪽을 좌변, 오른쪽을 우변이라고 하며, 연산순서는 정해진 우선 순위를 따릅니다.

| 연산자 | 내용 | 예제 |
|---|---|---|
| = | 좌변에 우변의 값을 대입 | msg.info(number1)<br>number1 = 100<br>msg.info(tostring(number1))<br><br>>> nil<br>>> 100 |
| + | 좌변에 우변의 값을 더함 | num1 = 1<br>num2 = num1 + 3<br>msg.info(tostring(num2))<br><br>>> 4 |
| - | 좌변에 우변의 값을 빼기 | num1 = 3<br>num2 = num1 - 2<br>msg.info(tostring(num2))<br><br>>> 1 |
| * | 좌변에 우변의 값을 곱함 | num1 = 3<br>num2 = num1 * 3<br>msg.info(tostring(num2))<br><br>>> 9 |
| / | 좌편에 우변의 값을 나눈 몫 | num1 = 5<br>num2 = num1 / 3<br>msg.info(tostring(num2))<br><br>>> 1 |
| % | 좌변에 우변의 값을 나눈 나머지 | num1 = 5<br>num2 = num1 % 3<br>msg.info(tostring(num2))<br><br>>> 2 |
| ^ | 좌변에 우변만큼의 지수승 | num1 = 2<br>num2 = num1^3 |

| | | |
|---|---|---|
| | | msg.info(tostring(num2))<br><br>>> 8 |
| > | 좌변의 값이 우변의 값보다 큼 | num1 = 4<br>num2 = 3<br>msg.info(tostring(num1 > num2))<br><br>>> true |
| < | 좌변의 값이 우변의 값보다 작음 | num1 = 3<br>num2 = 4<br>msg.info(tostring(num1 < num2))<br><br>>> true |
| >= | 좌변의 값이 우변의 값보다 크거나 같음 | num1 = 4<br>num2 = 3<br>msg.info(tostring(num1 >= num2))<br><br>>> true |
| <= | 좌변의 값이 우변의 값보다 작거나 같음 | num1 = 4<br>num2 = 3<br>msg.info(tostring(num1 <= num2))<br><br>>> false |
| == | 좌변의 값과 우변의 값이 같음 | num1 = 4<br>num2 = 4<br>msg.info(tostring(num1 == num2))<br><br>>> true |
| ~= | 좌변의 값과 우변의 값이 같지 않음 | num1 = 4<br>num2 = 3<br>msg.info(tostring(num1 ~= num2))<br><br>>> true |
| A and B | A와 B 모두 참이면 True를 A, B 둘 중 하나라도 거짓이면 거짓을 반환.<br>※ A와 B 모두 참이면 B를 출력함. | msg.info(tostring(4 and 5)) >> 5<br>msg.info(tostring(5 and 3)) >> 3<br>msg.info(tostring(nil and 3)) >> nil<br>msg.info(tostring(false and 5)) >> false |

| | | |
|---|---|---|
| A or B | A와 B 둘 중 하나라도 참이면 참을 반환. | msg.info(tostring(**4** or **5**)) >> 4<br>msg.info(tostring(**5** or **3**)) >> 5<br>msg.info(tostring(**false** or **5**)) >> 5<br>msg.info(tostring(**nil** or **3**)) >> 3<br>msg.info(tostring(**nil** or **false**)) >> false |
| not A | A가 참이면 false, 거짓이면 true를 반환. | msg.info(tostring(not **nil**)) >> true<br>msg.info(tostring(not **false**)) >> true<br>msg.info(tostring(not **0**)) >> false<br>msg.info(tostring(not not nil)) >> false |
| .. | 여러 문자열을 합침 | msg.info( "A" .. **100**) >> A100 |
| # | 문자열 및 테이블(배열)의 길이를 반환 | a = "korea"<br>msg.info(tostring(#a)) >> 5 |

**표 2-9 PRL에서 지원하는 연산자 예**

PRL에서는 **false와 nil만이 거짓을 의미**하며 나머지는 모두 참을 의미합니다. C에서 0은 거짓을 의미하지만 **PRL에서는 0 또한 참을 의미**합니다.

| 우선순위 | 연산자 |
|---|---|
| 1 | ^ |
| 2 | not, # |
| 3 | *, /, % |
| 4 | +, - |
| 5 | .. |
| 6 | >, <, >=, <=, ==, ~= |
| 7 | and |
| 8 | or |

**표 2-10 연산자 우선순위**

### 2.1.3.8. 조건문

조건문은 **주어진 조건에 따라 특정 작업을 수행**하도록 합니다. 주어진 조건은 참 또는 거짓으로 구분할 수 있으며, if문을 이용해 구분할 수 있습니다.

- if ~ then 사이의 조건이 성립할 때, then ~ end 사이의 구문을 실행합니다.
- 여러 개의 조건을 검사하고 싶을 때 elseif를 사용합니다.
- if와 elseif의 조건에 대항하지 않는 구문을 실행하고자 할 경우 else를 사용합니다.

* else 조건에서는 then을 사용하지 않습니다.

* end는 마지막 조건의 끝에 놓입니다.

| **예 1) if 조건문** | |
|---|---|
| `if a == 1 then`<br>`  b = 3`<br>`  msg.info(a + b)`<br>`end` | >> 4 |
| **예 2) if / elseif 조건문** | |
| `a = 1`<br><br>`if a == 0 then`<br>`  msg.info(a)`<br>`elseif a == 1 then`<br>`  msg.info(a * 3)`<br>`end` | >> 3 |
| **예 3) if / elseif / else 조건문** | |
| `a = 5`<br><br>`if a == 0 then`<br>`  msg.info(a)`<br>`elseif a == 1 then`<br>`  msg.info(a*3)`<br>`else`<br>`  msg.info(a*5)`<br>`end` | >> 25 |
| **예 4) 한 개의 if 문에 and 연산자 사용** | |
| `a = 4`<br>`b = 3`<br><br>`if a == 4 and b == 3 then`<br>`  msg.info(a+b)` | >> 7 |

| | |
|---|---|
| **end** | |

**표 2-11 조건문 예제**

### 2.1.3.9. 함수

PRL은 별도의 메인함수가 필요하지 않으며, 사용자가 함수를 정의하여 사용할 수 있습니다. 함수의 이름에 대한 규칙은 **변수명 규칙**을 따릅니다. 함수는 다음과 같은 기본구조를 가지게 됩니다.

- 함수의 시작은 function으로 끝은 end로 구분할 수 있습니다. 따라서 함수를 선언함에 있어, function과 end는 반드시 표기해야 합니다.
- 함수의 입력은 인자의 형태로 입력할 수 있습니다.
- functionName에는 함수의 이름을 설정할 수 있습니다.
- Parameter에는 함수의 입력 인자를 설정할 수 있습니다. 여러 개의 인자가 포함될 경우 ","로 구분할 수 있습니다.
- content에는 함수의 내용을 작성합니다. 함수가 실행되면 content의 내용이 실행됩니다.
- returnValue에는 함수의 반환값을 설정합니다.
  (※ PRL은 자료형이 없으므로 별도의 파라미터 리스트 또는 반환값에 대한 자료형을 명시하지 않습니다.)

**표 2-12 기본 함수 구조**

```
function funcationName(parameter)
    content
    return returnValue
end
```

| 예 1) 한 개의 인자를 받는 함수 | |
|---|---|
| function *sampleFunc*(a)<br>return a+10<br>end<br><br>b = sampleFunc(5)<br><br>msg.info(b) | >> 15 |

| 예 2) 여러 개의 인자를 받는 함수 | |
|---|---|
| function sampleFunc(a, b)<br>x = a + b<br>return x, y<br>end<br><br>rValue1 = sampleFunc(5, 4)<br><br>msg.info(rValue1) | >> 9 |
| **예 3) 여러 개의 인자를 받고 여러 개의 반환 값을 가지는 함수** | |
| function sampleFunc(a, b)<br>x = a + b<br>y = a - b<br>return x, y<br>end<br><br>rValue1, rValue2 = sampleFunc(5, 4)<br><br>msg.info(rValue1)<br>msg.info(rValue2) | >> 9<br>>> 1 |

**표 2-13 PRL 함수 예제**

### 2.1.3.10. 반복문

반복문은 일련의 명령문을 특정 조건에 따른 횟수동안 반복하여 실행하는 명령문입니다.

1) for 반복문

- for문은 for ~ do 사이의 설정된 횟수 동안 do ~ then 사이의 구문을 반복 실행합니다.
- for문은 원하는 횟수만큼 명령어를 반복 수행할 수 있습니다.

| for문 | | 사용법 |
|---|---|---|
| 숫자형 for문 | | for var=initial value, final value, [, increasing value] do<br>content<br>end |
| 제너릭 for문 | pairs(table) | for key[, value] in pairs(table) do<br>content<br>end |

| | | |
|---|---|---|
| | | 테이블에 대하여 iterate할 때, 모든 Key와 Value에 대하여 반복합니다. |
| | ipairs(table) | for index[, value] in inpairs(table) do<br>content<br>end |
| | | 테이블에 대하여 iterate할 때, 올바른 number 형식의 key에 대하여 iterate하며, value가 nil이라면 중지합니다. Indes가 정렬됩니다. |
| | next() | for key[, value] in next, table variable, nil do<br>content<br>end |
| | | Next는 도우미 함수로 pairs에서 사용되는 함수입니다. 입력된 테이블에서 현재 인덱스에서 다음에 오는 키와 값 쌍을 반환합니다. |

**표 2-14 반복문 예제: for**

| **예 1) 숫자형 for 반복문** | |
|---|---|
| `for i = 1, 2 do`<br>`  msg.info(tostring(i))`<br>`end`<br><br>`for i = 1, 4, 2 do`<br>`  msg.info(tostring(i))`<br>`end`<br><br>`msg.info(tostring(i))` *-- i 는 for 문 내부의 local 변수이므로 nil 이 반환됨.* | >> 1<br>>> 2<br>>> 1<br>>> 3<br>>> nil |

| 예 2) 제너릭 for 반복문(pairs) | |
|---|---|
| v = {"a", "b", "c", pi=3.14}<br><br>-- pairs(table)<br>for key, value in pairs(v) do<br><br>  if type(key) == "string" then<br>    msg.info(key, tostring(value))<br>  else<br>    msg.info(tostring(key), value)<br>  end<br>end | >> 1 a<br>>> 2 b<br>>> 3 c<br>>> pi 3.14 |
| **예 3) 제너릭 for 반복문(ipairs)** | |
| v = {"a", "b", "c", pi=3.14}<br><br>-- ipairs(table)<br>for index, value in ipairs(v) do<br>  msg.info(index, tostring(value))<br>end | >> 1 a<br>>> 2 b<br>>> 3 c |
| **예 4) 제너릭 for 반복문(next)** | |
| v = {"a", "b", "c", pi=3.14}<br><br>-- next<br>for key, value in next, v, nil do<br><br>  if type(key) == "string" then<br>    msg.info(key, tostring(value))<br>  else<br>    msg.info(tostring(key), value)<br>  end<br><br>end | >> 1 a<br>>> 2 b<br>>> 3 c<br>>> pi 3.14 |

**표 2-15 PRL 함수 예제**

2) while 반복문과 Repeat 반복문

- while문: while ~ do 사이의 조건 범위 동안 do ~ then 사이의 구문을 반복 실행합니다.
- repeat ~ until 문: until 조건 범위 동안 repeat ~ until 사이의 구문을 반복 실행합니다.

※ while은 비교조건이 실행할 명령문 이전에 나오며, repeat인 경우는 먼저 실행할 명령문을 실행한 후 비교 조건이 나옵니다.

※ while은 비교 조건이 참인 경우 계속 반복되지만, repeat는 비교 조건이 거짓인 경우에만 명령어를 반복 실행합니다.

| 예 1) While 반복문 | |
|---|---|
| a = 0<br>sum = 0<br><br>while a ~= 10 do<br>sum = a + sum<br>a = a + 1<br>end<br>msg.info(tostring(sum)) | >> 45 |
| **예 2) Repeat 반복문** | |
| a = 0<br>sum = 0<br><br>repeat<br>sum = a + sum<br>a = a + 1<br>until a == 10<br><br>msg.info(sum) | >> 45 |

**표 2-16 While / Repeat 반복문 예제**

### 2.1.3.11. break, return문

break및 return문은 블록에서 빠져나가고자 할 때 사용합니다.

- break문은 반복문을 종료할 때 사용합니다.
- break문이 실행되면 프로그램은 중지된 반복문 이후의 시점에서 실행을 계속합니다.
- return문은 함수로부터 결과를 반환할 뿐만 아니라 함수를 종료할 때 사용가능합니다.
- 모든 함수의 끝에는 암시적인 return문이 자연스럽게 있으므로 값을 반환하지 않는 함수의 끝에는 return을 사용할 필요가 없습니다.

(※ 문법적인 이유 때문에 break와 return문은 블록의 마지막에 두어야 합니다. (코드 뭉치의 가장 마지막 문장이거나 end, else, until 바로 앞에 놓여야만 합니다.)

| 예 1) Break 예제 | |
|---|---|
| i = 1<br><br>a = {1, 3, 5, 7, 9}<br><br>while a[i] do<br>  if a[i] == 5 then<br>  break<br>  end<br>  i = i + 1<br>end | |
| **예 2) Return 예제** | |
| function *sampleFunc*(a, b)<br>  return a + b<br>end<br><br>msg.info(sampleFunc(3 ,4)) | >> 7 |
| **예 3) Return 및 Break 복합 예제** | |
| function *sampleFunc2*(a, b)<br>  local s = 0<br>  while a ~= -1 do<br>  s = a + s<br>  a = a + 1<br><br>  if a == b then<br>  return s | >> 45<br>>> 4950 |

| | |
|---|---|
| end<br>end<br>end<br><br>msg.info(tostring(sampleFunc2(1, 10)))<br>msg.info(tostring(sampleFunc2(1, 100))) | |

**표 2-17 break 및 return 예제**

### 2.1.3.12. goto문

goto문은 지정한 이름의 라인으로 이동하여 스크립트를 수행하는 명령어 입니다.

- goto 'label 이름' 명령어를 사용하여 label 라인으로 이동할 수 있습니다.
- label 이름 좌우에 콜론(:) 두개(::)를 입력합니다. (예 : ::tryAgain::)

**예 1) goto 예제**

```
motion.speed = 100

p0 = joint(0, 0, 90, 0, 90, 0)
p1 = joint(0, 0, 90, 45, 90, 0)
pEND = joint(90, 0, 90, 0, 90, 0)

movep(p0)

value = var.i(1)

if value == 1 then
  -- END label 로 이동
  goto END
elseif value == 2 then
  dio.set(2)
end

movep(p1)

-- END label
::END::
movep(pEND)
```

표 2-18 goto문 예제

## 2.2. 스크립트 명령 해설 형식

<table>
<tr><td>■ 기 능</td><td>스크립트 명령어의 기능에 대하여 설명합니다.</td></tr>
<tr><td>■ 형 태</td><td>스크립트 명령어의 형태에 대하여 설명합니다.</td></tr>
<tr><td>■ 인 자</td><td>
<table>
<tr><th>인자명</th><th>자료형</th><th>기본값</th><th>범위</th><th>단위</th></tr>
<tr><td>인자1</td><td>string</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>[인자2]</td><td>int</td><td>-</td><td>-</td><td>-</td></tr>
</table>
※ 스크립트 명령어의 인자에 대하여 설명합니다.<br>
※ 인자의 입력 순서대로 나열하였으나, 입력 순서에 관계없는 경우 이를 별도로 표기하였습니다.<br>
(단, “[]”로 둘러싸인 인자는 옵션사항으로 상황에 따라 생략 또는 수정이 가능합니다.)
</td></tr>
<tr><td>■ 설 명</td><td>스크립트 명령어의 특징 및 응용분야에 대한 자세한 설명을 합니다.</td></tr>
<tr><td>■ 반환 값</td><td>스크립트 실행 결과에 대한 반환 값을 설명합니다.
<table>
<tr><th>구분</th><th>반환 값</th><th>설명</th></tr>
<tr><td>성공</td><td>-</td><td>-</td></tr>
<tr><td>실패</td><td>-</td><td>-</td></tr>
</table>
</td></tr>
<tr><td>■ 예 제</td><td>스크립트 사용법에 대한 예제를 설명합니다.</td></tr>
<tr><td>■ 에러 상황</td><td>에러 상황에 따라 출력되는 메시지의 원인을 설명하고, 간단한 조치 방법을 설명합니다.</td></tr>
<tr><td>■ 관련 함수</td><td>해당 함수와 관련된 함수를 참조합니다.</td></tr>
</table>

## 2.3. 동작 스크립트

모션 스크립트는 로봇 팔의 동작 유형에 따라 직선 이동과 회전 이동을 포함한 혼합 동작(Move), 직선 이동만 하는 병진 동작(Translation), 회전 이동만 하는 회전 동작(Rotation) 명령어로 구분되어 있습니다. 모션 명령어는 경로를 생성하는 방법에 따라 PTP(Point To Point Motion)와 Interpolation 계열로 나뉘어져 있습니다.

### ■ PTP 계열

- PTP 계열의 동작은 로봇 종단(End-Effector)의 경로를 고려하지 않고, 각각의 관절에 대한 경로를 생성하여 목표 위치까지 빠르게 도달하기 때문에 **직선 경로가 보장되지 않습니다**. 따라서 로봇 종단의 이동 경로보다 목표 위치까지 도착 시간이 중요한 경우에 사용됩니다.
- 관절이 2 개 이상 움직이는 경우 최대 회전 변위를 가지는 관절 기준으로 속도 프로파일을 생성하여 로봇 종단을 목표 위치로 이동합니다.
  - **PTP** : 직선 경로를 보장하지 않고 빠른 시간 내에 목표 위치까지 이동합니다.
  - **rotj** : 선택된 관절이 설정된 각도로 회전 이동합니다.

### ■ Interpolation 계열

- Interpolation 계열의 동작은 경로 계획에 따라 생성된 로봇 종단의 경로를 따라 이동합니다. 따라서 로봇 종단이 목표 위치에 도착하는 시간보다 이동 경로가 중요한 경우 사용됩니다.
- 모든 관절이 동시에 목표 위치에 도착하도록 속도 프로파일을 생성하여 로봇 종단을 이동합니다.
  - **Linear** : 목표 위치까지 TCP(Tool Center Point) 좌표계 기준으로 선형 보간 이동을 수행합니다.
  - **Circle** : 현재 위치에서 경유점을 지나 목표 위치까지 TCP 좌표계 기준으로 원호 이동을 수행합니다.
  - **Spline** : 현재 위치에서 경유점들을 거쳐 마지막 경유점까지 연결되는 곡선 이동을 수행합니다.
  - **Base coordinate** : Base 좌표계 기준으로 직선 이동을 수행합니다.
  - **Tool coordinate** : Tool 좌표계 기준으로 직선 이동을 수행합니다.
  - **rot** : 로봇의 종단이 Base 좌표계 기준으로 회전 이동을 수행합니다.

동작속도 옵션에는 로봇의 이동 속도(Speed), 가속도(Acceleration), 감속도(Deceleration)를 설정하는 명령어가 있습니다.

표 2-19 동작 명령어 리스트

| No. | 분류 | 항목 | 설명 | 참조 |
|---|---|---|---|---|
| 1 | 이동 속도옵션 | motion.speed=value | 속도 설정 | 2.3.1 |
| 2 | | motion.accel=value | 가속도 설정 | 2.3.2 |
| 3 | | motion.decel=value | 감속도 설정 | 2.3.3 |
| 4 | 변수 설정 | var_name=joint(J1, J2, J3, J4, J5, J6, [Workframe], [Toolframe]) | 관절 좌표계 위치 값 변수 할당 | 2.3.4 |
| 5 | | var_name=pose(X, Y, Z, Rx, Ry, Rz, [Workframe], [Toolframe]) | 직교 좌표계 위치 값 변수 할당 | 2.3.5 |
| 6 | 위치값 좌표변환 | value.j2p() | 좌표계 위치 변환 (관절 → 직교) | 2.3.6 |
| 7 | 역행렬 좌표계산 | value.inv() | 포즈 변수의 역행렬을 직교 좌표계로 변환 | 2.3.7 |
| 8 | 혼합 이동 | movep(Pos, [Blend], [Speed], [Next]) | PTP 이동 | 2.3.8 |
| 9 | | movel(Pos, [Blend], [Speed], [Next]) | 선형 보간 이동 | 2.3.9 |
| 10 | | movec(Pos1, Pos2, [Blend], [Speed], [Next]) | 원호 보간 이동 | 2.3.10 |
| 11 | | moves(Pos, Indicator, [Speed], [Next]) | 스플라인(Spline) 곡선 이동 | 2.3.11 |
| 12 | 상대 이동 | rmovep(Pos) | PTP 상대 이동 | 2.3.12 |
| 13 | | rmovel(Pos) | 선형 보간 상대 이동 | 2.3.13 |
| 14 | 병진 이동 | trans(X, Y, Z, [Speed]) | X-Y-Z축 기준 병진 이동 | 2.3.14 |
| 15 | | transx(X, [Speed]) | X축 방향 병진 이동 | 2.3.15 |
| 16 | | transy(Y, [Speed]) | Y축 방향 병진 이동 | 2.3.16 |
| 17 | | transz(Z, [Speed]) | Z축 방향 병진 이동 | 2.3.17 |
| 18 | 회전 이동 | rot(Rx, Ry, Rz, [Speed]) | X-Y-Z축 회전 이동 | 2.3.18 |
| 19 | | rotx(Rx, [Speed]) | X축 기준 회전 이동 | 2.3.19 |
| 20 | | roty(Ry, [Speed]) | Y축 기준 회전 이동 | 2.3.20 |
| 21 | | rotz(Rz, [Speed]) | Z축 기준 회전 이동 | 2.3.21 |
| 22 | | rotj1(Deg, [Speed]) | 관절 1축 회전 이동 | 2.3.22 |
| 23 | | rotj2(Deg, [Speed]) | 관절 2축 회전 이동 | 2.3.23 |
| 24 | | rotj3(Deg, [Speed]) | 관절 3축 회전 이동 | 2.3.24 |
| 25 | | rotj4(Deg, [Speed]) | 관절 4축 회전 이동 | 2.3.25 |
| 26 | | rotj5(Deg, [Speed]) | 관절 5축 회전 이동 | 2.3.26 |
| 27 | | rotj6(Deg, [Speed]) | 관절 6축 회전 이동 | 2.3.27 |
| 28 | Tool 좌표계 기준 병진 이동 | tool.trans(X, Y, Z , [Speed]) | Tool 좌표계 기준 병진 이동 | 2.3.28 |
| 29 | | tool.transx(X, [Speed]) | Tool 좌표계 기준 X축 방향 병진 이동 | 2.3.29 |
| 30 | | tool.transy(Y, [Speed]) | Tool 좌표계 기준 Y축 방향 병진 이동 | 2.3.30 |
| 31 | | tool.transz(Z, [Speed]) | Tool 좌표계 기준 Z축 방향 병진 이동 | 2.3.31 |
| 32 | | tool.rot(Rx, Ry, Rz, [Speed]) | Tool 좌표계 기준 회전 이동 | 2.3.32 |
| 33 | | tool.rotx(Rx, [Speed]) | Tool 좌표계 기준 X축 회전 이동 | 2.3.33 |
| 34 | | tool.roty(Ry, [Speed]) | Tool 좌표계 기준 Y축 회전 이동 | 2.3.34 |
| 35 | | tool.rotz(Rz, [Speed]) | Tool 좌표계 기준 Z축 회전 이동 | 2.3.35 |
| 36 | 시간 지연 함수 | delay(Time) | 입력한 시간 만큼 대기 | 2.3.36 |

| | | | | |
|---|---|---|---|---|
| 37 | 좌표계 설정 | workframe(Num) | Work 좌표계 설정 | 2.3.37 |
| 38 | | toolframe(Num) | Tool 좌표계 설정 | 2.3.38 |

표 2-19 는 동작 스크립트에 대해 간략히 설명한 표이며, 보다 안전한 사용을 위해 각 항목의 참조 페이지로 이동 후 자세히 숙지 후 사용하시기 바랍니다.

## 2.3.1. motion.speed : 이동 속도 설정

| | |
|---|---|
| ■ 기 능 | **로봇 이동 속도를 설정합니다.** |
| ■ 형 태 | motion.speed = value |

■ 인 자

| 인자 | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|
| | | | 최소 | 최대 | |
| value | int | 25 | 1 | 100 | % |

* 단위 : % (percent)

■ **세부사항**

➢ **value (이동 속도 설정)**

- 로봇 구동 시 **시스템 속도가 "50%"**, 스크립트의 **속도 명령어가 "10%"** 으로 설정된 경우, 표 2-20의 예제 1과 같이 **최종 이동 속도는** "5%"의 속도가 적용됩니다.

| 항목 | 예제 1 | 예제 2 |
|---|---|---|
| **시스템 속도** | 50 % | 100 % |
| **속도 명령어(motion.speed)** | 10 % | 10 % |
| **최종 이동 속도** | 5 % | 10 % |

**표 2-20 최종 이동 속도**

● **시스템 속도**

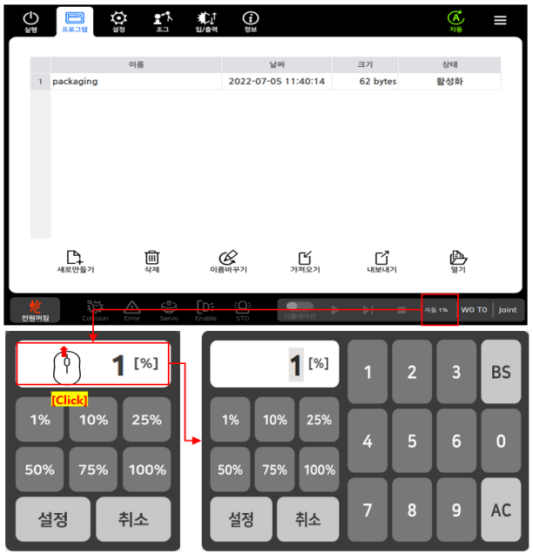



**그림 2-2 시스템 속도**

- 시스템 속도는 그림 2-2과 같이 펜던트 화면의 우측하단에 있습니다.
- 속도 명령어를 사용 시 최종 이동 속도는 시스템 속도와 속도 명령어가 곱해진 값이 적용됩니다.

■ 예 제

**[예제1] 속도 10% 설정**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | motion.speed = 10 | -- 이동 속도 10%로 설정 |
| 2 | p1=pose(700, 408, 82, 90, 1, 35) | -- p1에 위치 데이터 설정 |
| 3 | | |
| 4 | movep(p1) | -- PTP 이동 |

**[예제2] 속도 100% 설정**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | motion.speed = 100 | -- 이동 속도 100%로 설정 |
| 2 | | |
| 3 | movel(lp1) | -- 선형 보간 이동 |

- lp 인자는 로컬 포인트를 의미하며, 사전 티칭 필요

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 타입 | Invalid argument[1] type of number data |
| 실행오류 | 데이터 유효범위 | Value[151] is range over(1~100) |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.2. motion.accel : 가속도 설정

■ 기 능

**로봇 가속도를 설정합니다.**

■ 형 태

motion.accel = value

■ 인 자

| 인자 | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|
| | | | 최소 | 최대 | |
| value | int | 100 | 50 | 150 | % |

* 단위 : % (percent)

■ **세부사항**

➢ **value (가속도 설정)**

- 로봇 전체 이동시간은 인자값이 작으면 증가하고, 크면 감소합니다. (그림 2-3)
- 가속도 변경 후 초기 값으로 되돌리기 위해서는 100으로 설정 바랍니다.

| 인자 | 등속구간 도달 시간 | 로봇 이동 시간 |
|---|---|---|
| **value > 100** | 감소 (▼) | 감소 (▼) |
| **value < 100** | 증가 (▲) | 증가 (▲) |

**표 2-21 가속에 따른 로봇 이동시간**

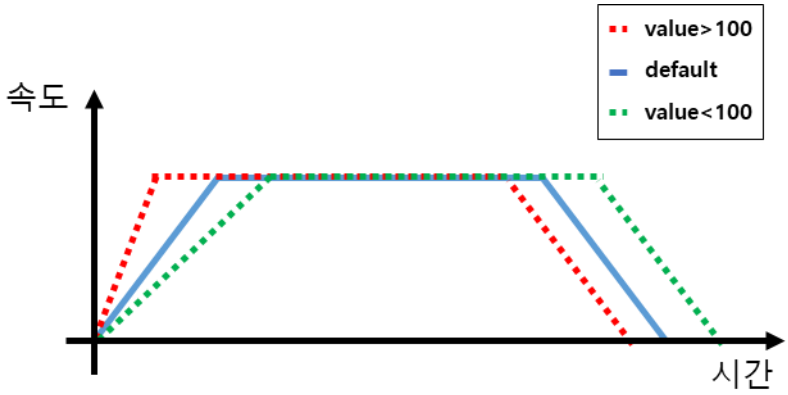



**그림 2-3 가속도에 따른 로봇 이동시간**

■ 예 제

**[예제1] 가속도 70% 설정**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | motion.accel = 70 | -- 가속도 70%로 설정 |
| 2 | | |
| 3 | p1=pose(570, 318, 770, 40, 1, 105) | -- p1에 위치형 데이터 설정 |
| 4 | movep(p1) | -- PTP 이동 |

**[예제2] 가속도 130% 설정**

| 번호 | 스크립트 | 주석 |
|---|---|---|

```
1  motion.accel = 130          -- 가속도 130%로 설정
2
3  movel(lp1)                  -- 선형 보간 이동
```

- lp 인자는 로컬 포인트를 의미하며, 사전 티칭 필요

■ **에러 상황**

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 타입 | Invalid argument[1] type of number data |
| 실행오류 | 데이터 유효범위 | Value[151] is range over(1~100) |

■ **관련 함수**

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.3. motion.decel : 감속도 설정

■ 기 능

**로봇 감속도를 설정합니다.**

■ 형 태

motion.decel = value

■ 인 자

| 인자 | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|
| | | | 최소 | 최대 | |
| value | int | - | 50 | 150 | % |

* 단위 : % (percent)

■ **세부사항**

➢ **value (감속도 설정)**

- 로봇 전체 이동시간은 인자값이 작으면 감소하고, 크면 증가합니다. (그림 2-4)
- 감속도 변경 후 초기 값으로 되돌리기 위해서는 100으로 설정 바랍니다.

| 인자 | 등속구간 도달 시간 | 로봇 이동 시간 |
|---|---|---|
| **value > 100** | 감소 (▼) | 감소 (▼) |
| **value < 100** | 증가 (▲) | 증가 (▲) |

**표 2-22 가속에 따른 로봇 이동시간**

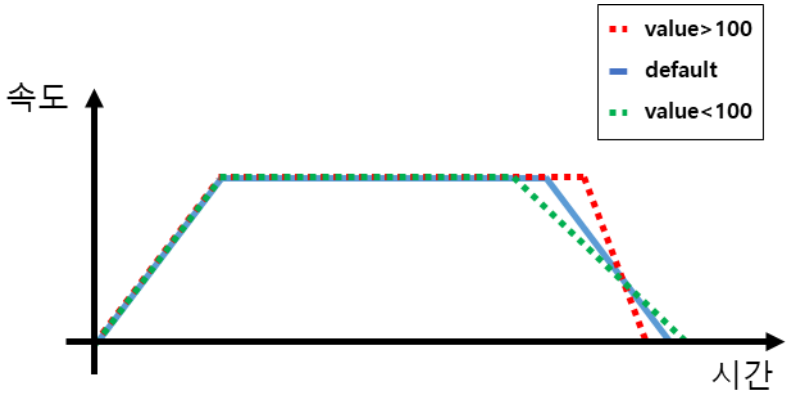



**그림 2-4 감속도에 따른 로봇 이동시간**

■ 예 제

**[예제1] 가속도 70% 설정**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | motion.decel = 70 | -- 감속도 70%로 설정 |
| 2 | | |
| 3 | p1=pose(570, 318, 770, 40, 1, 105) | -- p1에 위치형 데이터 설정 |
| 4 | movep(p1) | -- PTP 이동 |

**[예제2] 가속도 140% 설정**

| 번호 | 스크립트 | 주석 |
|---|---|---|

```
1  motion.decel = 140          -- 이동 속도 140%로 설정
2
3  movel(lp1)                  -- 선형 보간 이동
```

- lp 인자는 로컬 포인트를 의미하며, 사전 티칭 필요

■ **에러 상황**

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | syntax error near <eof> |
| 실행 오류 | 데이터 유효범위 | Value[151] is range over(50~150) |

■ **관련 함수**

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.4. joint : 관절형 위치 변수 초기화

| | |
|---|---|
| ■ 기 능 | **관절 좌표계 위치 값을 변수에 할당합니다.** |
| ■ 형 태 | var_name = joint(J1, J2, J3, J4, J5, J6, [Workframe], [Toolframe]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| Joint | J1 | double | - | -360.0 | 360.0 | deg |
| | J2 | double | - | -360.0 | 360.0 | deg |
| | J3 | double | - | -150.0 | 150.0 | deg |
| | J4 | double | - | -360.0 | 360.0 | deg |
| | J5 | double | - | -360.0 | 360.0 | deg |
| | J6 | double | - | -360.0 | 360.0 | deg |
| [Workframe] | | int | -1 | 0 | 10 | - |
| [Toolframe] | | int | -1 | 0 | 10 | - |

* 단위 : % (percent)

■ **세부사항**

➢ **J1, J2, J3, J4, J5, J6 (관절 좌표계 위치 값 설정)**

- 변수명 규칙에 따라 var_name 대신 사용자 변수명을 사용할 수 있습니다.
- 위치형 변수는 사칙연산을 지원합니다.

➢ **var_name.j* (관절 좌표계 각 축의 위치 값 확인)**

- 관절 좌표계로 할당된 위치형 변수의 각 축 데이터를 확인할 수 있습니다.
- "var_name"은 관절 좌표계로 할당된 위치형 변수입니다.

| 명령어 | 설명 |
|---|---|
| var_name.j1 | 1 번 축 위치형 데이터 |
| var_name.j2 | 2 번 축 위치형 데이터 |
| var_name.j3 | 3 번 축 위치형 데이터 |
| var_name.j4 | 4 번 축 위치형 데이터 |
| var_name.j5 | 5 번 축 위치형 데이터 |
| var_name.j6 | 6 번 축 위치형 데이터 |

**표 2-23 관절 좌표계 각 축의 위치**

➢ **[Workframe] (워크 좌표계 값 설정)**

- 해당 인자는 옵션 인자입니다.
- 워크 좌표계의 값을 설정합니다. 0 ~ 10번까지의 워크 좌표계 설정이 가능합니다.
- 해당 인자값을 설정하지 않았을 경우, 현재 워크 좌표계 값으로 설정됩니다.

➢ **[Toolframe] (툴 좌표계 값 설정)**

- 해당 인자는 옵션 인자입니다.
- 툴 좌표계의 값을 설정합니다. 0 ~ 10번까지의 툴 좌표계 설정이 가능합니다.
- 해당 인자값을 설정하지 않았을 경우, 현재 툴 좌표계 값으로 설정됩니다.

● **사칙연산**

- 위치형 데이터형 사칙연산 결과는 표 2-24와 같이 연산자의 가장 선두에 위치한 데이터의 좌표계로 정의됩니다.

| 사칙연산 | 위치형 데이터형 | | |
|---|---|---|---|
| | **p3 (결과)** | **p1** | **p2** |
| p3 = p1+p2 | **joint** | joint | joint |
| | **joint** | joint | pose |
| | **pose** | pose | joint |
| | **pose** | pose | pose |

**표 2-24 위치형 데이터형 사칙연산**

■ **예 제**

**[예제1] 관절 좌표계 위치 값 할당**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=joint(0, 30, 0, 60, 90, 0) | |
| 2 | p2=joint(30, 0, 30, 0, 0, 0) | |
| 3 | | |
| 4 | p3=p1+p2 | -- p1 + p2 값이 p3에 할당 |
| 5 | p4=p1-p2 | -- p1 - p2 값이 p4에 할당 |
| 6 | | |
| 7 | p5=p4+joint(30, 30, 0, 0, 0, 0) | -- joint 변수를 직접 사용하여 p5 할당 |
| 8 | movep(p5) | -- PTP 이동 |

**[예제2] 관절 좌표계 워크, 툴 좌표계 할당**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=joint(0, 30, 0, 60, 90, 0, 0, 0) | -- 워크 좌표계 0, 툴 좌표계 0인 |
| 2 | | 관절형 위치 변수 선언 |
| 3 | p2=joint(0, 30, 0, 60, 90, 0, 0, 1) | -- 워크 좌표계 0, 툴 좌표계 1인 |
| 4 | | 관절형 위치 변수 선언 |
| 5 | movep(p1) | -- 해당 변수가 가리키는 종단 위치로 |
| 6 | | 이동 |
| 7 | movep(p2) | -- 해당 변수가 가리키는 종단 위치로 |
| 8 | | 이동 |

■ **에러 상황**

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 타입 | Invalid argument[6] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |

■ **관련 함수**

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.5. pose : 직교형 위치 변수 초기화

| | |
|---|---|
| ■ 기 능 | **변수에 직교 좌표계 위치 값을 할당합니다.** |
| ■ 형 태 | var_name = pose(X, Y, Z, Rx, Ry, Rz, [Workframe], [Toolframe]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| XYZ | X | double | - | -1700.0 | 1700.0 | mm |
| | Y | double | - | -1700.0 | 1700.0 | mm |
| | Z | double | - | -1700.0 | 1700.0 | mm |
| | Rx | double | - | -180.0 | 180.0 | deg |
| | Ry | double | - | -180.0 | 180.0 | deg |
| | Rz | double | - | -180.0 | 180.0 | deg |
| [Workframe] | | int | -1 | 0 | 10 | - |
| [Toolframe] | | int | -1 | 0 | 10 | - |

* 단위 : deg (degree) / mm (millimeter)

■ **세부사항**

➢ **X, Y, Z, Rx, Ry, Rz (직교 좌표계 위치 값 설정)**

- 변수명 규칙에 따라 var_name 대신 사용자 변수명을 사용할 수 있습니다.
- 위치형 변수는 사칙연산을 지원합니다.

➢ **var_name.x / y / z / rx / ry / rz (직교 좌표계 각 축의 위치 값 확인)**

- 직교 좌표계로 할당된 위치형 변수의 각 축 데이터를 확인할 수 있습니다.
- "var_name"은 직교 좌표계로 할당된 위치형 변수입니다.

| 명령어 | 설명 |
|---|---|
| var_name.x | X 축 위치형 데이터 |
| var_name.y | Y 축 위치형 데이터 |
| var_name.z | Z 축 위치형 데이터 |
| var_name.rx | Rx 축 위치형 데이터 |
| var_name.ry | Ry 축 위치형 데이터 |
| var_name.rz | Rz 축 위치형 데이터 |

**표 2-25 직교 좌표계 각 축의 위치**

➢ **[Workframe] (워크 좌표계 값 설정)**

- 해당 인자는 옵션 인자입니다.
- 워크 좌표계의 값을 설정합니다. 0 ~ 10번까지의 워크 좌표계 설정이 가능합니다.
- 해당 인자값을 설정하지 않았을 경우, 현재 워크 좌표계 값으로 설정됩니다.

➢ **[Toolframe] (툴 좌표계 값 설정)**

- 해당 인자는 옵션 인자입니다.
- 툴 좌표계의 값을 설정합니다. 0 ~ 10번까지의 툴 좌표계 설정이 가능합니다.
- 해당 인자값을 설정하지 않았을 경우, 현재 툴 좌표계 값으로 설정됩니다.

● **사칙연산**

- 위치형 데이터형 사칙연산 결과는 표 2-26와 같이 연산자의 가장 선두에 위치한 데이터의 좌표계로 정의됩니다.

| 사칙연산 | 위치형 데이터형 | | |
|---|---|---|---|
| | **p3 (결과)** | **p1** | **p2** |
| p3 = p1+p2 | **joint** | joint | joint |
| | **joint** | joint | pose |
| | **pose** | pose | joint |
| | **pose** | pose | pose |

**표 2-26 위치형 데이터형 사칙연산**

■ **예 제**

**[예제1] 직교 좌표계 위치 값 할당**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1 = pose(750,400,800,50,1,135) | |
| 2 | p2 = pose(-90,300,450,180,1,50) | |
| 3 | p3 = pose(100,380,400,0,1,30) | |
| 4 | | |
| 5 | p4 = p1+p2 | -- p1+p2 값이 p4에 할당 |
| 6 | p5 = p4-p3 | -- p4-p3 값이 p5에 할당 |
| 7 | | |
| 8 | movep(p5) | -- PTP 이동 |

**[예제2] 직교 좌표계 워크, 툴 좌표계 할당**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=pose(533.5, 48, 615.7, 180, 0, 180, | -- 워크 좌표계 0, 툴 좌표계 0인 |
| 2 | 0, 0) | 직교형 위치 변수 선언 |
| 3 | | |
| 4 | p2= pose(533.5, 48, 615.7, 180, 0, | -- 워크 좌표계 0, 툴 좌표계 1인 |
| 5 | 180, 0, 1) | 직교형 위치 변수 선언 |
| 6 | | |
| 7 | movep(p1) | -- 해당 변수가 가리키는 종단 위치로 |
| 8 | | 이동 |
| 9 | movep(p2) | -- 해당 변수가 가리키는 종단 위치로 |
| 10 | | 이동 |

■ **에러 상황**

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 타입 | Invalid argument[6] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |

■ **관련 함수**

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.6. j2p : 위치 값 변환(관절에서 직교)

| | |
|---|---|
| ■ 기 능 | **관절 좌표계 위치 값을 직교 좌표계 위치 값으로 변환합니다.** |
| ■ 형 태 | var_name = Joint.j2p() |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| Joint | J1 | double | - | -360.0 | 360.0 | deg |
| | J2 | double | - | -360.0 | 360.0 | deg |
| | J3 | double | - | -150.0 | 150.0 | deg |
| | J4 | double | - | -360.0 | 360.0 | deg |
| | J5 | double | - | -360.0 | 360.0 | deg |
| | J6 | double | - | -360.0 | 360.0 | deg |
| [Workframe] | | int | -1 | 0 | 10 | - |
| [Toolframe] | | int | -1 | 0 | 10 | - |

* 단위 : deg (degree)

■ **세부사항**

➢ **J1, J2, J3, J4, J5, J6 (관절 좌표계 위치 값 설정)**

- "Joint"는 관절 좌표계로 할당된 위치형 변수입니다.
- 변수명 규칙에 따라 var_name 대신 사용자 변수명을 정의 할 수 있습니다.

➢ **[Workframe] (워크 좌표계 값 설정)**

- 해당 인자는 옵션 인자입니다.
- "Joint"는 관절 좌표계로 할당된 위치형 변수입니다.
- 워크 좌표계의 값을 설정합니다. 0 ~ 10번까지의 워크 좌표계 설정이 가능합니다.
- 해당 인자값을 설정하지 않았을 경우, 현재 워크 좌표계 값으로 설정됩니다.

➢ **[Toolframe] (툴 좌표계 값 설정)**

- 해당 인자는 옵션 인자입니다.
- "Joint"는 관절 좌표계로 할당된 위치형 변수입니다.
- 툴 좌표계의 값을 설정합니다. 0 ~ 10번까지의 툴 좌표계 설정이 가능합니다.
- 해당 인자값을 설정하지 않았을 경우, 현재 툴 좌표계 값으로 설정됩니다.

■ 반환값

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| XYZ | X | double | - | -1700.0 | 1700.0 | mm |
| | Y | double | - | -1700.0 | 1700.0 | mm |
| | Z | double | - | -1700.0 | 1700.0 | mm |
| | Rx | double | - | -180.0 | 180.0 | deg |
| | Ry | double | - | -180.0 | 180.0 | deg |
| | Rz | double | - | -180.0 | 180.0 | deg |

■ 예 제

**[예제1] j2p 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1 = joint(60,0,80,50,0,90) | |
| 2 | | |
| 3 | p2 = p1.j2p() | -- 위치형으로 변환 후 p2에 할당 |
| 4 | movep(p2) | -- PTP 이동 |
| 5 | | |
| 6 | p3 = p2 + joint(0,30,10,10,30,0) | |
| 7 | | |
| 8 | movep(p3.j2p()) | -- PTP 이동 |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid argument |
| | 명령어 괄호 | ')' expected near ']' |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.7. inv : 포즈 변수의 역행렬을 직교 좌표계로 변환

| | |
|---|---|
| ■ 기 능 | **포즈 변수의 역행렬 값을 직교 좌표계 값으로 변환합니다.** |
| ■ 형 태 | var_name = Pose.inv() |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| XYZ | X | double | - | -1700.0 | 1700.0 | mm |
| | Y | double | - | -1700.0 | 1700.0 | mm |
| | Z | double | - | -1700.0 | 1700.0 | mm |
| | Rx | double | - | -180.0 | 180.0 | deg |
| | Ry | double | - | -180.0 | 180.0 | deg |
| | Rz | double | - | -180.0 | 180.0 | deg |
| [Workframe] | | int | -1 | 0 | 10 | - |
| [Toolframe] | | int | -1 | 0 | 10 | - |

* 단위 : deg (degree) / mm (millimeter)

■ **세부사항**

➢ **X, Y, Z, Rx, Ry, Rz (직교 좌표계 위치 값 설정)**

- "Pose"는 직교 좌표계로 할당된 위치형 변수입니다.
- 변수명 규칙에 따라 var_name 대신 사용자 변수명을 정의 할 수 있습니다.

➢ **[Workframe] (워크 좌표계 값 설정)**

- 해당 인자는 옵션 인자입니다.
- "Joint"는 관절 좌표계로 할당된 위치형 변수입니다.
- 워크 좌표계의 값을 설정합니다. 0 ~ 10번까지의 워크 좌표계 설정이 가능합니다.
- 해당 인자값을 설정하지 않았을 경우, 현재 워크 좌표계 값으로 설정됩니다.

➢ **[Toolframe] (툴 좌표계 값 설정)**

- 해당 인자는 옵션 인자입니다.
- "Joint"는 관절 좌표계로 할당된 위치형 변수입니다.
- 툴 좌표계의 값을 설정합니다. 0 ~ 10번까지의 툴 좌표계 설정이 가능합니다.
- 해당 인자값을 설정하지 않았을 경우, 현재 툴 좌표계 값으로 설정됩니다.

■ 반환값

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| XYZ | X | double | - | -1700.0 | 1700.0 | mm |
| | Y | double | - | -1700.0 | 1700.0 | mm |
| | Z | double | - | -1700.0 | 1700.0 | mm |
| | Rx | double | - | -180.0 | 180.0 | deg |
| | Ry | double | - | -180.0 | 180.0 | deg |
| | Rz | double | - | -180.0 | 180.0 | deg |

■ 예 제

**[예제1] inv 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1 = joint(60,0,80,50,0,90) | |
| 2 | | |
| 3 | p2 = p1.j2p() | -- 위치형으로 변환 후 p2에 할당 |
| 4 | | ※ 직교 위치 : 303.11, 621, 903.75, |
| 5 | | 74.66, -48.97, 161.69 |
| 6 | p3 = p2.inv() | -- p2 위치를 행렬로 변환 후, 해당 |
| 7 | | 행렬의 역행렬을 구하여 포즈 값으로 |
| 8 | | 변환 |
| 9 | | ※ 직교 위치 : -620.95, -458.37, |
| 10 | | -835.82, 78.49, -29.49, 135.72 |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid argument |
| | 명령어 괄호 | ')' expected near ']' |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.8. movep : 보간이동

| | |
|---|---|
| ■ 기 능 | **직선 경로를 보장하지 않고 빠른 시간 내에 목표 위치까지 이동합니다.** |
| ■ 형 태 | movep(Position, [Blend], [Speed], [Next]) |

■ 인 자

| 인자 | | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|---|
| 좌표계 | | 축 | | | 최소 | 최대 | |
| Position | Joint | J1 | double | - | -360.0 | 360.0 | deg |
| | | J2 | double | - | -360.0 | 360.0 | deg |
| | | J3 | double | - | -150.0 | 150.0 | deg |
| | | J4 | double | - | -360.0 | 360.0 | deg |
| | | J5 | double | - | -360.0 | 360.0 | deg |
| | | J6 | double | - | -360.0 | 360.0 | deg |
| | XYZ | X | double | - | -1700.0 | 1700.0 | mm |
| | | Y | double | - | -1700.0 | 1700.0 | mm |
| | | Z | double | - | -1700.0 | 1700.0 | mm |
| | | Rx | double | - | -180.0 | 180.0 | deg |
| | | Ry | double | - | -180.0 | 180.0 | deg |
| | | Rz | double | - | -180.0 | 180.0 | deg |
| [Blend] | | | string | 0 | 0 | 100 | % |
| [Speed] | | | string | 100 | 0 | 100 | % |
| [Next] | | | string | - | - | - | - |

* 단위 : deg (degree) / mm (millimeter) / % (percent)

■ **세부사항**

➢ **Position (목표점 설정)**

- Position 인자는 Joint(관절 좌표계) 또는 XYZ(직교 좌표계)에 대해 설정합니다.
- Joint인 경우 설정 값을 각도[deg] 단위로 입력합니다
- XYZ인 경우 X, Y, Z축은 위치[mm] 단위로, Rx, Ry, Rz축은 각도[deg] 단위로 입력합니다.

➢ **[Blend] (경유점을 원호로 이은 경로 생성)**

- 해당 인자는 옵션 인자입니다.
- Blend 설정 값은 선행 모션과 후행 모션의 중첩되는 속도의 비율입니다.
- 설정 값이 없는 경우, Blend가 0으로 적용됩니다.
- Blend 모션은 1점 이상의 경유점이 필요합니다.
- 종료 지점에서는 "b0"설정해야하며, "b0" 이외의 설정 값은 Blend 종료 신호로 유효하지 않습니다.

⚠ **Blend의 반지름 길이가 커지면 작업 영역을 벗어나 주변의 물체와 충돌이**

**발생할 수 있습니다.**

**현재 시작위치의 Blend 설정 값만 적용되며, 추후 적용되는 모든 Blend 옵션이 적용되도록 개선 예정입니다.**

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-27을 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ② 속도 명령어 | 50 % | 50 % | |
| ③ 속도 옵션 | - | 10 % | |
| ④ 최종 적용 속도 | 25 % | 5 % | |

**표 2-27 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

**시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.**

➢ **[Next] (다음 명령어 동시 수행)**

- 해당 인자는 옵션 인자입니다.
- 로봇이 생성된 궤적을 이동하며 입출력 설정, 데이터 값 변경, 수학 관련함수를 동시에 수행하는 기능입니다.
- Next 옵션 설정 후 다음 명령어가 모션 명령어인 경우 적용되지 않습니다.

**Blend, Speed, Next 옵션은 순서에 상관없이 사용 가능합니다**

■ **movep 경로**

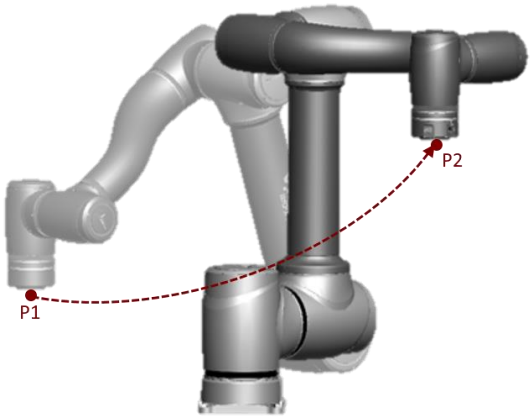



**그림 2-5 movep 경로**

- movep 명령어는 로봇 종단이 P1에서 P2로 그림 2-5와 같이 직선 궤적을 고려하지 않고 경로를 이동합니다.

⚠ **PTP모션에서는 로봇 끝단의 궤적을 고려하지 않으므로, 의도하지 않은 충돌이 발생할 수 있습니다.**

■ **설 명**

■ **Blend**

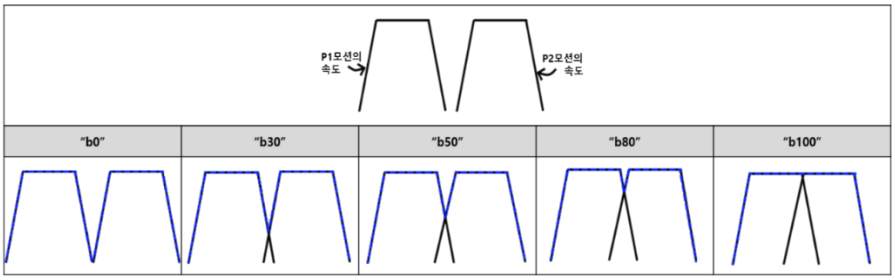



**그림 2-6 movep Blending**

- 그림 2-6의 "b100"의 경우와 같이 모션과 모션간의 중첩되는 속도 비율이 커질수록 목표점까지 이동시간이 감소합니다.

| | |
|---|---|
| **응용분야** | ● **페인팅(Painting)**<br>- 복잡한 곡선 모션을 정밀하고 신속하게 수행하여 디스펜서의 분사물이 일정한 속도로 도포될 수 있도록 블렌딩 기능을 지원하고 있습니다.<br>● **글루잉(Gluing)/본딩(Bonding)**<br>- 직선이나 원형과 같이 모든 형태의 곡선을 손쉽게 정속 주행하여 글루잉/본딩 작업을 수행할 수 있습니다. |

■ 예 제

**[예제1] blend 옵션 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | movep(lp10, "b10") | -- 목표위치 및 블랜딩 설정 |
| 2 | | |
| 3 | movep(lp11, "s20", "b10") | -- 블렌딩 연결 경유점 |
| 4 | | |
| 5 | movep(lp12, "b0") | -- 블렌딩 연결 종료 |

**[예제2] next 옵션 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=joint(30, 30, 60, 30, 0, 0) | -- 목표위치 설정 |
| 2 | | |
| 3 | movep(p1, "s50", "next") | -- next 옵션 사용 |
| 4 | a = math.sin(60) | -- 이동 명령 수행 시, 같이 명령어 |
| 5 | | 실행 |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument type of pose data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 경로 생성 실패 | 경로 생성 실패 |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

**주의(Caution)**

- 로봇은 설정된 관절각으로 이동하기 때문에 Tool, Work 좌표계의 설정과 무관하게 움직일 수 있습니다.

**정보(Notice)**

- 로봇은 각 관절의 각도 값의 정보를 이용하여 움직이기 때문에 중간의 경로를 특정하여 이동할 수 없습니다.

## 2.3.9. movel : 선형이동

| | |
|---|---|
| ■ 기 능 | **로봇 종단이 현재 위치에서 목표 위치까지 TCP 좌표계 기준으로 직선 경로를 생성하여 이동합니다.** |
| ■ 형 태 | movel(Position, [Blend], [Speed], [Next]) |

■ 인 자

| 인자 | | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|---|
| 좌표계 | | 축 | | | 최소 | 최대 | |
| Position | Joint | J1 | double | - | -360.0 | 360.0 | deg |
| | | J2 | double | - | -360.0 | 360.0 | deg |
| | | J3 | double | - | -150.0 | 150.0 | deg |
| | | J4 | double | - | -360.0 | 360.0 | deg |
| | | J5 | double | - | -360.0 | 360.0 | deg |
| | | J6 | double | - | -360.0 | 360.0 | deg |
| | XYZ | X | double | - | -1700.0 | 1700.0 | mm |
| | | Y | double | - | -1700.0 | 1700.0 | mm |
| | | Z | double | - | -1700.0 | 1700.0 | mm |
| | | Rx | double | - | -180.0 | 180.0 | deg |
| | | Ry | double | - | -180.0 | 180.0 | deg |
| | | Rz | double | - | -180.0 | 180.0 | deg |
| [Blend] | | | string | 0 | - | - | mm |
| [Speed] | | | string | 100 | 0 | 100 | % |
| [Next] | | | string | - | - | - | - |

* 단위 : deg (degree) / mm (millimeter) / % (percent)

■ **세부사항**

➢ **Position (목표점 설정)**

- Position 인자는 Joint(관절 좌표계) 또는 XYZ(직교 좌표계)에 대해 설정합니다.
- Joint인 경우 설정 값을 각도[deg] 단위로 입력합니다
- XYZ인 경우 X, Y, Z축은 위치[mm] 단위로, Rx, Ry, Rz축은 각도[deg] 단위로 입력합니다.

➢ **[Blend] (경유점을 원호로 이은 경로 생성)**

- 해당 인자는 옵션 인자입니다.
- 설정 값이 없는 경우, Blend radius가 0으로 적용됩니다.
- Blend 모션은 1점 이상의 경유점이 필요합니다.
- 종료 지점에서는 "b0"을 설정하거나 설정이 없어야 하며, "b0" 이외의 설정 값은 블렌딩 종료 신호로 유효하지 않습니다.

⚠ **Blend의 반지름 길이가 커지면 작업 영역을 벗어나 주변의 물체와 충돌이**

**발생할 수 있습니다.**

ⓘ **현재 시작위치의 Blend 설정 값만 적용되며, 두번째 이후 Blend 값은 무시됩니다.**

➢ **[Speed] (로봇 상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-28 Speed 옵션 사용에 따른 최종 속도(예시)을 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ② 속도 명령어 | 50 % | 50 % | |
| ③ 속도 옵션 | - | 10 % | |
| ④ 최종 적용 속도 | 25 % | 5 % | |

**표 2-28 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

ⓘ **시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.**

➢ **[Next] (다음 명령어 동시 수행)**

- 해당 인자는 옵션 인자입니다.
- 로봇이 생성된 궤적을 이동하며 입출력 설정, 데이터 값 변경, 수학 관련함수를 동시에 수행하는 기능입니다.
- Next 옵션 설정 후 다음 명령어가 모션 명령어인 경우 적용되지 않습니다.

ⓘ **Blend, Speed, Next 옵션은 순서에 상관없이 사용 가능합니다**

■ **movel 경로**

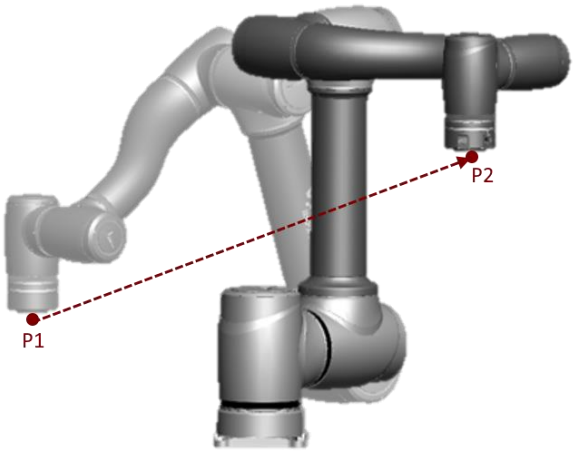



**그림 2-7 movel 경로**

- movel 명령어는 로봇 종단이 P1에서 P2로 그림 2-7과 같이 생성된 직선 경로로 이동합니다.

**Interpolation 계열의 move 명령어는 역기구학 연산을 통해 로봇이 동작합니다. 따라서 역기구학 연산이 불가능한 특이점(Singular) 영역에서는 움직임이 제한됩니다.**

■ **설 명**

■ **Blend**

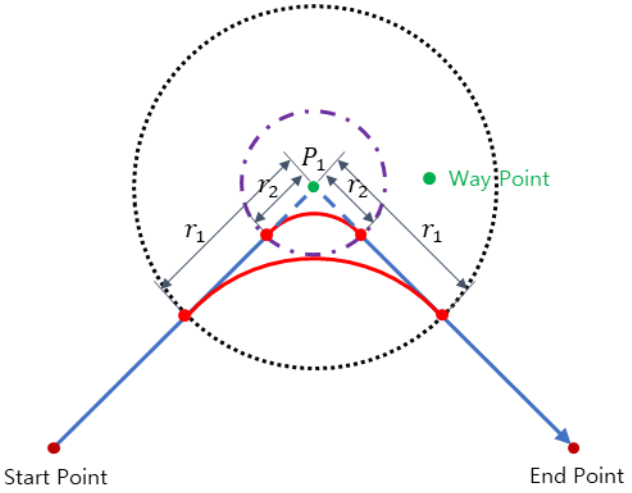



**그림 2-8 movel Blending**

| | |
|---|---|
| **응용분야** | ● **페인팅(Painting)**<br>- 복잡한 곡선 모션을 정밀하고 신속하게 수행하여 디스펜서의 분사물이 일정한 속도로 도포될 수 있도록 블렌딩 기능을 지원하고 있습니다.<br>● **글루잉(Gluing)/본딩(Bonding)**<br>- 직선이나 원형과 같이 모든 형태의 곡선을 손쉽게 정속 주행하여 글루잉/본딩 작업을 수행할 수 있습니다. |

- 시작 위치(Start Point)에서 경유 위치($P_1$)를 거쳐 도착 위치(End Point)까지 생성된 직선 궤적에 접하는 원의 원호를 그림 2-8과 같이 궤적으로 생성합니다.
- Blend 시작 위치에서 경유 위치까지 거리를 설정하여 사용합니다.
- 설정한 거리의 크기에 따라 생성된 궤적과 경유 위치까지 거리 및 목표 위치까지 이동시간은 그림 2-8과 같이 비례하여 증가/감소합니다.
- Blend 크기에 따라 생성된 궤적과 경유 위치까지 거리 및 목표 위치까지 이동시간은 그림 2-8과 같이 비례하여 증가/감소합니다.

**Blend 값의 범위는 시작 위치(Start Point)와 도착 위치(End Point) 중, 거리가 짧은 위치 기준의 1/2 까지 설정이 가능하며, 그 이상 초과하여 입력할 경우, 자동으로 짧은 거리의 1/2 값으로 설정이 됩니다.**

**예) 시작 위치 - 경유 위치 간 거리 : 120mm**
**경유 위치 - 도착 위치 간 거리 : 86mm**
**Blend 적용 최대 값 : 86 / 2 = 43mm**

**Blend 크기를 작게 설정하여 짧은 곡률 궤적으로 로봇이 구동할 경우, 로봇 선속도 초과 에러가 발생할 수 있습니다. Blend의 크기를 알맞게 설정하십시오.**

■ 예 제

**[예제1] 속도 10% 설정**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=joint(30, 30, 60, 30, 0, 0) | -- 목표 위치 설정 |
| 2 | p2=pose(785.70, 413.78, 822.70, 95.50, | -- 관절좌표계 |
| 3 | 1.03, 135.89) | |
| 4 | motion.speed = 20 | -- XYZ 좌표계 |
| 5 | movel(p1) | -- 20% 속도 사용 |
| 6 | movel(p2, "s10") | -- 속도 옵션 사용 (속도 명령어 |
| 7 | | 적용되지 않음) |

**[예제2] blend 설정**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | movel(lp10, *"b10"*) | -- 블렌딩 연결 경유점, "b10"은 |
| 2 | | 10mm를 의미 |
| 3 | movel(lp11, *"b10"*) | -- 블렌딩 연결 경유점 |
| 4 | movel(lp12, *"b0"*) | -- 블렌딩 연결 종료 |

**[예제3] next 설정**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=joint(30, 30, 60, 30, 0, 0) | -- $P_1$ 경유점 |
| 2 | p2=pose(785.70, 413.78, 822.70, 95.50, | -- $P_2$ 목표점 |
| 3 | 1.03, 135.89) | |
| 4 | | |
| 5 | movel(p1, *"s50"*) | |
| 6 | movel(p2, *"s20"*, *"next"*) | -- 모션 종료를 기다리지 않고 다음 |
| 7 | a = math.sin(60) | 명령어까지 수행 |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument type of pose data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 경로 생성 실패 | 경로 생성 실패 |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

**정보(Notice)**

- movec 명령어와 혼합 블렌딩이 가능합니다.

## 2.3.10. movec : 원호이동

| | |
|---|---|
| ■ 기 능 | **로봇 종단이 현재 위치에서 목표 위치까지 TCP 좌표계 기준으로 원호 경로를 생성하여 이동합니다.** |
| ■ 형 태 | movec(Position1, Position2, [Blend], [Speed], [Next]) |

■ 인 자

| 인자 | | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|---|
| 좌표계 | | 축 | | | 최소 | 최대 | |
| Position1 | Joint | J1 | double | - | -360.0 | 360.0 | deg |
| | | J2 | double | - | -360.0 | 360.0 | deg |
| | | J3 | double | - | -150.0 | 150.0 | deg |
| Position2 | | J4 | double | - | -360.0 | 360.0 | deg |
| | | J5 | double | - | -360.0 | 360.0 | deg |
| | | J6 | double | - | -360.0 | 360.0 | deg |
| Position1 | XYZ | X | double | - | -1700.0 | 1700.0 | mm |
| | | Y | double | - | -1700.0 | 1700.0 | mm |
| | | Z | double | - | -1700.0 | 1700.0 | mm |
| Position2 | | Rx | double | - | -180.0 | 180.0 | deg |
| | | Ry | double | - | -180.0 | 180.0 | deg |
| | | Rz | double | - | -180.0 | 180.0 | deg |
| [Blend] | | | string | 0 | 0 | 100 | mm |
| [Speed] | | | string | 100 | 0 | 100 | % |
| [Next] | | | string | - | - | - | - |

* 단위 : deg (degree) / mm (millimeter) / % (percent)

■ **세부사항**

➢ **Position (목표점 설정)**

- Position 인자는 Joint(관절 좌표계) 또는 XYZ(직교 좌표계)에 대해 설정합니다.
- Joint인 경우 설정 값을 각도[deg] 단위로 입력합니다.
- XYZ인 경우 X, Y, Z축은 위치[mm] 단위로, Rx, Ry, Rz축은 각도[deg] 단위로 입력합니다.

➢ **[Blend] (경유점을 원호로 이은 경로 생성)**

- 해당 인자는 옵션 인자입니다.
- 설정 값이 없는 경우, Blend radius가 0으로 적용됩니다.
- Blend 모션은 1점 이상의 경유점이 필요합니다.
- 종료 지점에서는 "b0"을 설정하거나 설정이 없어야 하며, "b0" 이외의 설정 값은 블렌딩 종료 신호로 유효하지 않습니다.
- movel 명령어와 혼합 Blend가 가능합니다. 단, PTP 모션 및 moves 명령어는 지원하지 않습니다.

**Blend의 반지름 길이가 커지면 작업 영역을 벗어나 주변의 물체와 충돌이 발생할 수 있습니다.**

**현재 시작위치의 Blend 설정 값만 적용되며, 추후 적용되는 모든 Blend 옵션이 적용되도록 개선 예정입니다.**

➢ **[Speed] (로봇 상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-29를 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ② 속도 명령어 | 50 % | 50 % | |
| ③ 속도 옵션 | - | 10 % | |
| ④ 최종 적용 속도 | 25 % | 5 % | |

**표 2-29 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

**시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.**

➢ **[Next] (다음 명령어 동시 수행)**

- 해당 인자는 옵션 인자입니다.
- 로봇이 생성된 궤적을 이동하며 입출력 설정, 데이터 값 변경, 수학 관련함수를 동시에 수행하는 기능입니다.
- Next 옵션 설정 후 다음 명령어가 모션 명령어인 경우 적용되지 않습니다.

**Blend, Speed, Next 옵션은 순서에 상관없이 사용 가능합니다**

■ **movec 경로**

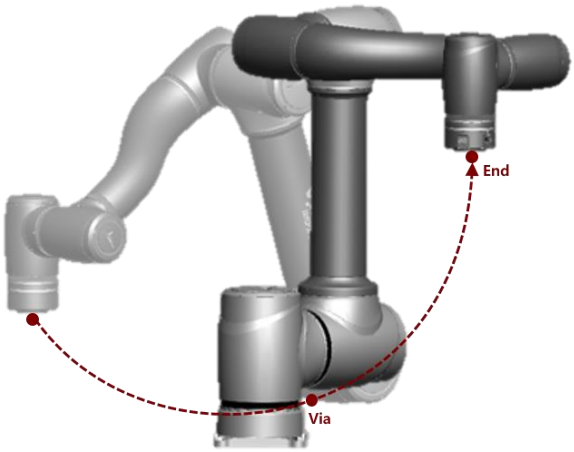



**그림 2-9 movec 경로**

- movec 명령어는 로봇 종단이 그림 2-9와 같이 시작지점에서 Via포인트를 지나 End로 원호를 그리며 이동합니다.

**Interpolation 계열의 move 명령어는 역기구학 연산을 통해 로봇이 동작합니다. 따라서 역기구학 연산이 불가능한 특이점(Singular) 영역에서는 움직임이 제한됩니다.**

■ 설 명

■ **Blend**

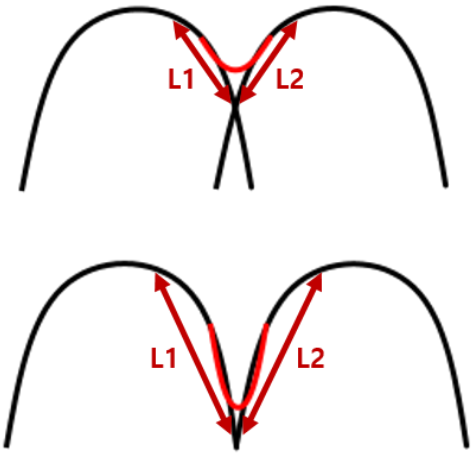



**그림 2-10 movec Blending**

| | |
|---|---|
| **응용분야** | ● **페인팅(Painting)**<br>- 복잡한 곡선 모션을 정밀하고 신속하게 수행하여 디스펜서의 분사물이 일정한 속도로 도포될 수 있도록 블렌딩 기능을 지원하고 있습니다.<br>● **글루잉(Gluing)/본딩(Bonding)**<br>- 직선이나 원형과 같이 모든 형태의 곡선을 손쉽게 정속 주행하여 글루잉/본딩 작업을 수행할 수 있습니다. |

■ 예 제

**[예제1] movec 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | motion.speed=100 | |
| 2 | | |
| 3 | p1=joint(0, 0, 90, 0, 90, 0) | -- $P_1$ 시작 위치 |
| 4 | p2=joint(30, 0, 60, 0, 120, 0) | -- $P_2$ 경유 위치 |
| 5 | p3=joint(60, 20, 50, 0, 110, 0) | -- $P_3$ 목표 위치 |
| 6 | | |
| 7 | movep(p1) | -- 시작 위치로 이동 |
| 8 | movec(p2, p3, *"s20"*, *"next"*) | -- 원호 이동 |
| 9 | dio.set(1) | -- 모션 종료를 기다리지 않고 다음 |
| 10 | | 명령어까지 수행 |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument type of pose data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 경로 생성 실패 | 경로 생성 실패 |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.11. moves : 스플라인 곡선 이동

| | |
|---|---|
| ■ 기능 | **로봇 종단이 현재 위치에서 경유점들을 거쳐 마지막 경유점까지 연결되는 스플라인 곡선경로를 따라 이동합니다.** |
| ■ 형태 | moves(Position, [Blend], [Speed], [Next]) |

■ 인자설명

| 인자 | | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|---|
| 좌표계 | | 축 | | | 최소 | 최대 | |
| Position1 | Joint | J1 | double | - | -360.0 | 360.0 | deg |
| | | J2 | double | - | -360.0 | 360.0 | deg |
| | | J3 | double | - | -150.0 | 150.0 | deg |
| | | J4 | double | - | -360.0 | 360.0 | deg |
| | | J5 | double | - | -360.0 | 360.0 | deg |
| | | J6 | double | - | -360.0 | 360.0 | deg |
| | XYZ | X | double | - | -1700.0 | 1700.0 | mm |
| | | Y | double | - | -1700.0 | 1700.0 | mm |
| | | Z | double | - | -1700.0 | 1700.0 | mm |
| | | Rx | double | - | -180.0 | 180.0 | deg |
| | | Ry | double | - | -180.0 | 180.0 | deg |
| | | Rz | double | - | -180.0 | 180.0 | deg |
| [Blend] | | | string | 0 | - | - | mm |
| [Speed] | | | string | 100 | 0 | 100 | % |
| [Next] | | | string | - | - | - | - |

* 단위 : deg (degree) / mm (millimeter) / % (percent)

■ **세부사항**

➢ **Position (목표점 설정)**

- Position 인자는 Joint(관절 좌표계)에 대해 설정합니다.
- Joint 인 경우 설정 값을 각도[deg] 단위로 입력합니다.
- XYZ인 경우 X, Y, Z축은 위치[mm] 단위로, Rx, Ry, Rz축은 각도[deg] 단위로 입력합니다.

➢ **[Blend] (경유점을 곡선으로 이은 경로 생성)**

- 해당 인자는 옵션 인자입니다.
- moves의 경우 다른 명령어와 다르게 0과 1을 이용하여 경유 위치를 연결지점, 연결 종료 지점을 적용할 수 있습니다.
- 설정 값이 없는 경우, Blend 값은 0으로 적용됩니다.
- Blend 모션은 1점 이상의 경유점이 필요합니다.
- 종료 지점에서는 "b0"을 설정하거나 설정이 없어야 하며, "b0" 이외의 설정 값은 블렌딩 종료 신호로 유효하지 않습니다.

➢ **[Speed] (로봇 상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-30을 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ② 속도 명령어 | 50 % | 50 % | |
| ③ 속도 옵션 | - | 10 % | |
| ④ 최종 적용 속도 | 25 % | 5 % | |

**표 2-30 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

ⓘ **시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.**

ⓘ **moves 명령어의 경우 첫번째 명령어의 속도 옵션을 기준으로 로봇이 동작합니다.**

➢ **[Next] (다음 명령어 동시 수행)**

- 해당 인자는 옵션 인자입니다.
- 로봇이 생성된 궤적을 이동하며 입출력 설정, 데이터 값 변경, 수학 관련함수를 동시에 수행하는 기능입니다.
- Next 옵션 설정 후 다음 명령어가 모션 명령어인 경우 적용되지 않습니다.

ⓘ **Blend, Speed, Next 옵션은 순서에 상관없이 사용 가능합니다**

■ **moves 경로**

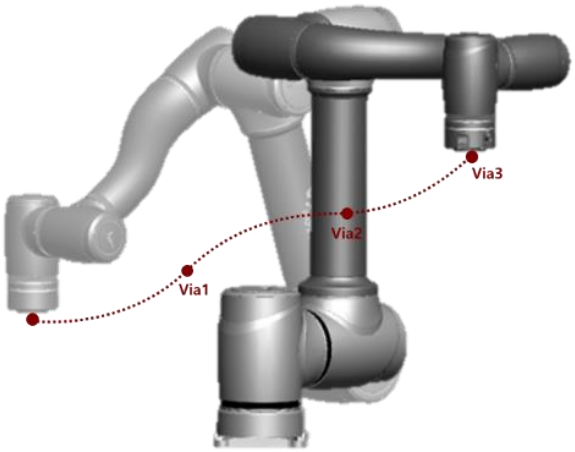



**그림 2-11 moves 경로**

- moves 명령어는 로봇 종단이 시작위치에서 경유점 Via1 ~ 3까지 연속으로 연결되는 곡선 경로를 따라 이동합니다.

■ **예제**

**[예제1] moves 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | motion.speed = 100 | |
| 2 | | |
| 3 | p1=joint(0, 0, 90 ,0 ,90, 0) | -- $P_1$ 시작점 |
| 4 | p2=joint(30, 0, 60, 0, 120, 0) | -- $P_2$ 경우점 |
| 5 | p3=joint(60, 20, 110, 0, 50, 0) | -- $P_3$ 경유점 |
| 6 | p4=joint(80, 40, 40, 0, 100, 0) | -- $P_4$ 목표점 |
| 7 | | |
| 8 | movep(p1) | -- 시작점 이동 |
| 9 | moves(p2, *"b1"*, *"s20"*, *"next"*) | -- 스플라인 경유점 추가 |
| 10 | a = math.sin(60) | -- next 옵션 실행 |
| 11 | moves(p3, *"b1"*) | -- 경유점 추가 |
| 12 | moves(p4, *"b0"*) | -- 종료점 추가 |

■ **에러 상황**

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument type of pose data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 경로 생성 실패 | 경로 생성 실패 |

■ 관련함수

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.12. rmovep : 상대 보간이동

| | |
|---|---|
| ■ 기 능 | **로봇의 현재 위치를 기준으로 입력한 값 만큼 PTP 이동합니다.** |
| ■ 형 태 | rmovep(Position, Speed) |

■ 인 자

| 인자 | | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|---|
| 좌표계 | | 축 | | | 최소 | 최대 | |
| Position | Joint | J1 | double | - | -360.0 | 360.0 | deg |
| | | J2 | double | - | -360.0 | 360.0 | deg |
| | | J3 | double | - | -150.0 | 150.0 | deg |
| | | J4 | double | - | -360.0 | 360.0 | deg |
| | | J5 | double | - | -360.0 | 360.0 | deg |
| | | J6 | double | - | -360.0 | 360.0 | deg |
| | XYZ | X | double | - | -1700.0 | 1700.0 | mm |
| | | Y | double | - | -1700.0 | 1700.0 | mm |
| | | Z | double | - | -1700.0 | 1700.0 | mm |
| | | Rx | double | - | -180.0 | 180.0 | deg |
| | | Ry | double | - | -180.0 | 180.0 | deg |
| | | Rz | double | - | -180.0 | 180.0 | deg |
| [Speed] | | | string | 100 | 0 | 100 | % |

* 단위 : deg (degree) / mm (millimeter) / % (percent)

■ **세부사항**

➢ **Position (목표점 설정)**

- 현재 위치를 기준으로 이동할 각도/거리를 설정합니다.
- Position 인자는 Joint(관절 좌표계) 또는 XYZ(직교 좌표계)에 대해 설정합니다.
- Joint 인 경우 설정 값을 각도[deg] 단위로 입력합니다
- XYZ 인 경우 X, Y, Z축은 위치[mm] 단위로, Rx, Ry, Rz축은 각도 단위로 입력합니다.

➢ **Speed (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-31을 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ② 속도 명령어 | 50 % | 50 % | |
| ③ 속도 옵션 | - | 10 % | |
| ④ 최종 적용 속도 | 25 % | 5 % | |

**표 2-31 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

**시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.**

## ■ rmovep 경로

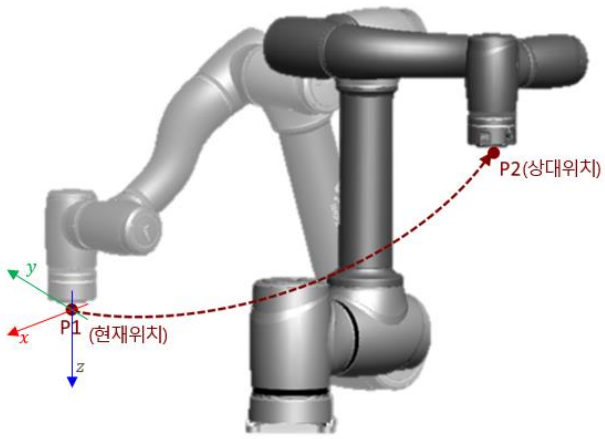



**그림 2-12 rmovep 경로**

- rmovep 명령어는 로봇 종단이 현재 위치 P1에서 상대 위치만큼 이동하여 P2로 그림 2-12와 같이 직선 궤적을 고려하지 않고 경로를 이동합니다.
- PTP모션에서는 로봇 끝단의 궤적을 고려하지 않으므로, 의도하지 않은 충돌이 발생할 수 있습니다.

■ 예 제

**[예제1] rmovep 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=joint(50,30,40,30,0,10) | -- $P_1$ 경유점 |
| 2 | p2=joint(10, 0, 25, 0, 0, 0) | -- $P_2$ 목표점 |
| 3 | | |
| 4 | movep(p1) | -- 경유점 이동 |
| 5 | rmovep(p2) | -- 경유점에서 1, 3축 상대 이동 |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument type of pose data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 경로 생성 실패 | 경로 생성 실패 |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.13. rmovel : 상대 선형이동

| | |
|---|---|
| ■ 기 능 | **로봇의 현재 종단 위치를 기준으로 입력한 위치 값 만큼 직선 이동합니다.** |
| ■ 형 태 | rmovel(Position, Speed) |

■ 인 자

| 인자 | | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|---|
| 좌표계 | | 축 | | | 최소 | 최대 | |
| Position | Joint | J1 | double | - | -360.0 | 360.0 | deg |
| | | J2 | double | - | -360.0 | 360.0 | deg |
| | | J3 | double | - | -150.0 | 150.0 | deg |
| | | J4 | double | - | -360.0 | 360.0 | deg |
| | | J5 | double | - | -360.0 | 360.0 | deg |
| | | J6 | double | - | -360.0 | 360.0 | deg |
| | XYZ | X | double | - | -1700.0 | 1700.0 | mm |
| | | Y | double | - | -1700.0 | 1700.0 | mm |
| | | Z | double | - | -1700.0 | 1700.0 | mm |
| | | Rx | double | - | -180.0 | 180.0 | deg |
| | | Ry | double | - | -180.0 | 180.0 | deg |
| | | Rz | double | - | -180.0 | 180.0 | deg |
| [Speed] | | | string | 100 | 0 | 100 | % |

* 단위 : deg (degree) / mm (millimeter) / % (percent)

■ **세부사항**

➢ **Position (목표점 설정)**

- 현재 위치를 기준으로 이동할 각도/거리를 설정합니다.
- Position 인자는 Joint(관절 좌표계) 또는 XYZ(직교 좌표계)에 대해 설정합니다.
- Joint 인 경우 설정 값을 각도[deg] 단위로 입력합니다
- XYZ 인 경우 X, Y, Z축은 위치[mm] 단위로, Rx, Ry, Rz축은 각도 단위로 입력합니다.

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-32를 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ② 속도 명령어 | 50 % | 50 % | |
| ③ 속도 옵션 | - | 10 % | |
| ④ 최종 적용 속도 | 25 % | 5 % | |

**표 2-32 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

**시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.**

■ **rmovel 경로**

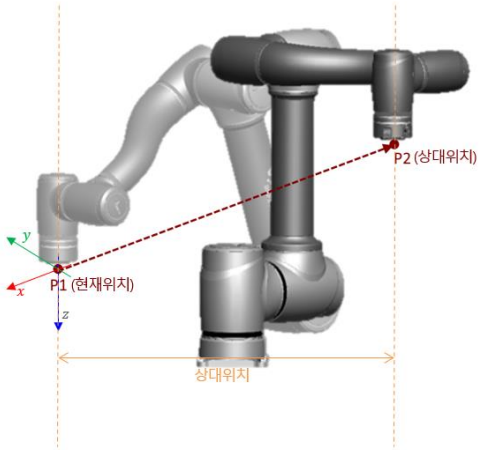



**그림 2-13 rmovel 경로**

- rmovel 명령어는 로봇 종단이 현재 위치 P1에서 상대 위치만큼 이동하여 P2로 그림 2-13과 같이 직선 경로를 이동합니다.

**Interpolation 계열의 move 명령어는 역기구학 연산을 통해 로봇이 동작합니다. 따라서 역기구학 연산이 불가능한 특이점(Singular) 영역에서는 움직임이 제한됩니다.**

■ 예 제

**[예제1] rmovel 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=joint(50,30,40,30,0,10) | -- $P_1$ 경유점 |
| 2 | p2=pose(100, 0, 100, 0, 0, 0) | -- $P_2$ 목표점 |
| 3 | | |
| 4 | movep(p1) | -- 경유점 이동 |
| 5 | rmovel(p2) | -- 경유점에서 x, z축으로 상대 이동 |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument type of pose data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 경로 생성 실패 | 경로 생성 실패 |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.14. trans : 병진이동

| | |
|---|---|
| ■ 기 능 | **현재 위치에서 설정한 변위만큼 Base 좌표계 기준으로 이동합니다.** |
| ■ 형 태 | trans(X, Y, Z, [Speed]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| XYZ | X | double | - | -1700.0 | 1700.0 | mm |
| | Y | double | - | -1700.0 | 1700.0 | mm |
| | Z | double | - | -1700.0 | 1700.0 | mm |
| [Speed] | | string | 100 | 0 | 100 | % |

* 단위 : mm (millimeter) / % (percent)

■ **세부사항**

➢ **X, Y, Z (변위 설정)**

- 현재 위치에서 Base 좌표계 기준으로 이동할 변위를 설정합니다.

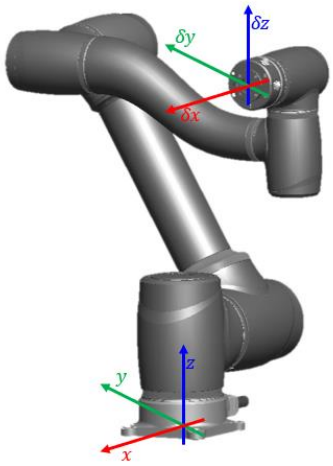



**그림 2-14 trans 이동 방향**

- trans 명령어는 로봇 종단이 현재 위치에서 Base 좌표계 기준으로 설정된 변위만큼 그림 2-14와 같이 이동합니다.

⚠ **Interpolation 계열의 이동 명령어는 역기구학 계산이 불가능한 특이점 영역 (Singular)에서는 움직임이 제한됩니다.**

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.

- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-33을 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ② 속도 명령어 | 50 % | 50 % | |
| ③ 속도 옵션 | - | 10 % | |
| ④ 최종 적용 속도 | 25 % | 5 % | |

**표 2-33 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

**시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.**

■ 예 제

**[예제1] trans 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=pose(500,380,500,0,1,30) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | trans(20,0,10) | -- X축 20mm, Y축 0mm, Z축 10mm 이동 |
| 5 | | |
| 6 | | ※ 최종 위치 : |
| 7 | | pose(520,380,510,0,1,30) |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |
| | 경로 생성 실패 | 경로 생성 실패 |

■ **관련 함수**

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.15. transx : 병진이동 (X축 기준)

| | |
|---|---|
| ■ 기 능 | **현재 위치에서 설정한 X축 변위만큼 Base 좌표계 기준으로 이동합니다.** |
| ■ 형 태 | transx(X, [Speed]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| XYZ | X | double | - | -1700.0 | 1700.0 | mm |
| [Speed] | | string | 100 | 0 | 100 | % |

* 단위 : mm (millimeter) / % (percent)

■ **세부사항**

➢ **X (목표점 설정)**

- 현재 위치에서 Base 좌표계 기준으로 이동할 변위를 설정합니다.

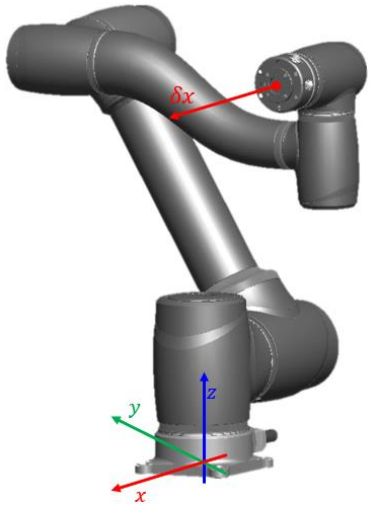



**그림 2-15 transx 이동 방향**

- transx 명령어는 로봇 종단이 현재 위치에서 Base 좌표계 기준으로 설정된 변위만큼 그림 2-15와 같이 이동합니다.

⚠ **Interpolation 계열의 이동 명령어는 역기구학 계산이 불가능한 특이점 영역 (Singular)에서는 움직임이 제한됩니다.**

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.

- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-34를 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ② 속도 명령어 | 50 % | 50 % | |
| ③ 속도 옵션 | - | 10 % | |
| ④ 최종 적용 속도 | 25 % | 5 % | |

**표 2-34 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

**시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.**

■ 예 제

**[예제1] transx 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=pose(500,380,500,0,1,30) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | transx(20) | -- X축 20mm |
| 5 | | ※ 최종 위치 : |
| 6 | | pose(520,380,500,0,1,30) |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |
| | 경로 생성 실패 | 경로 생성 실패 |

■ **관련 함수**

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.16. transy : 병진이동 (Y축 기준)

| | |
|---|---|
| ■ 기 능 | **현재 위치에서 설정한 Y축 변위만큼 Base 좌표계 기준으로 이동합니다.** |
| ■ 형 태 | transy(Y, [Speed]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| XYZ | Y | double | - | -1700.0 | 1700.0 | mm |
| [Speed] | | string | 100 | 0 | 100 | % |

* 단위 : mm (millimeter) / % (percent)

■ **세부사항**

➢ **Y (목표점 설정)**

- 현재 위치에서 Base 좌표계 기준으로 이동할 변위를 설정합니다.

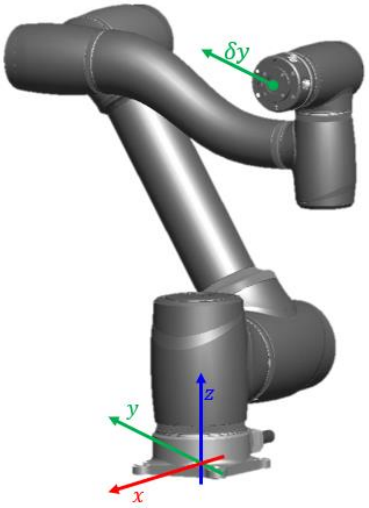



**그림 2-16 transy 이동 방향**

- transy 명령어는 로봇 종단이 현재 위치에서 Base 좌표계 기준으로 설정된 변위만큼 그림 2-16과 같이 이동합니다.

⚠ **Interpolation 계열의 이동 명령어는 역기구학 계산이 불가능한 특이점 영역 (Singular)에서는 움직임이 제한됩니다.**

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는

"5%"가 됩니다.

- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-35를 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ② 속도 명령어 | 50 % | 50 % | |
| ③ 속도 옵션 | - | 10 % | |
| ④ 최종 적용 속도 | 25 % | 5 % | |

**표 2-35 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

**시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.**

■ 예 제

**[예제1] transy 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=pose(500,380,500,0,1,30) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | transy(20) | -- Y축 20mm |
| 5 | | ※ 최종 위치 : |
| 6 | | pose(500,400,500,0,1,30) |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |
| | 경로 생성 실패 | 경로 생성 실패 |

■ **관련 함수**

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.17. transz : 병진이동 (Z축 기준)

| | |
|---|---|
| ■ 기 능 | **현재 위치에서 설정한 Z축 변위만큼 Base 좌표계 기준으로 이동합니다.** |
| ■ 형 태 | transz(Z, [Speed]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| XYZ | Z | double | - | -1700.0 | 1700.0 | mm |
| [Speed] | | string | 100 | 0 | 100 | % |

* 단위 : mm (millimeter) / % (percent)

■ **세부사항**

➢ **Z (목표점 설정)**

- 현재 위치에서 Base 좌표계 기준으로 이동할 변위를 설정합니다.

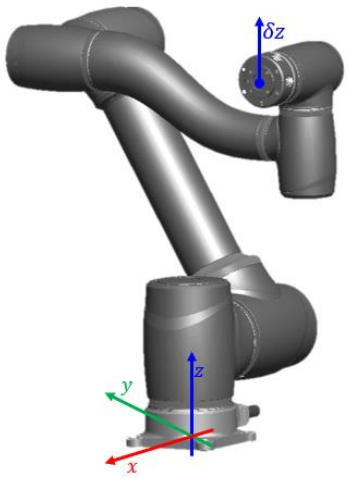



**그림 2-17 transz 경로**

- transz 명령어는 로봇 종단이 현재 위치에서 Base 좌표계 기준으로 설정된 변위만큼 그림 2-17과 같이 이동합니다.

⚠ **Interpolation 계열의 이동 명령어는 역기구학 계산이 불가능한 특이점 영역 (Singular)에서는 움직임이 제한됩니다.**

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만

적용됩니다.

- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-36을 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ② 속도 명령어 | 50 % | 50 % | |
| ③ 속도 옵션 | - | 10 % | |
| ④ 최종 적용 속도 | 25 % | 5 % | |

**표 2-36 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

**시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.**

## ■ 예 제

**[예제1] transz 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=pose(500,380,500,0,1,30) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | transz(20) | -- Z축 20mm |
| 5 | | ※ 최종 위치 : |
| 6 | | pose(500,380,520,0,1,30) |

## ■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |
| | 경로 생성 실패 | 경로 생성 실패 |

■ **관련 함수**

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.18. rot : 회전이동

| | |
|---|---|
| ■ 기 능 | **현재 위치에서 설정한 회전 변위만큼 Base 좌표계 기준으로 회전합니다.** |
| ■ 형 태 | rot(Rx, Ry, Rz, [Speed]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| XYZ | Rx | double | - | -180.0 | 180.0 | deg |
| | Ry | double | - | -180.0 | 180.0 | deg |
| | Rz | double | - | -180.0 | 180.0 | deg |
| [Speed] | | string | 100 | 0 | 100 | % |

* 단위 : deg (degree) / % (percent)

■ **세부사항**

➢ **Rx, Ry, Rz (목표점 설정)**

- 현재 위치에서 Base 좌표계의 축 중심으로 회전 기준할 변위를 설정합니다.

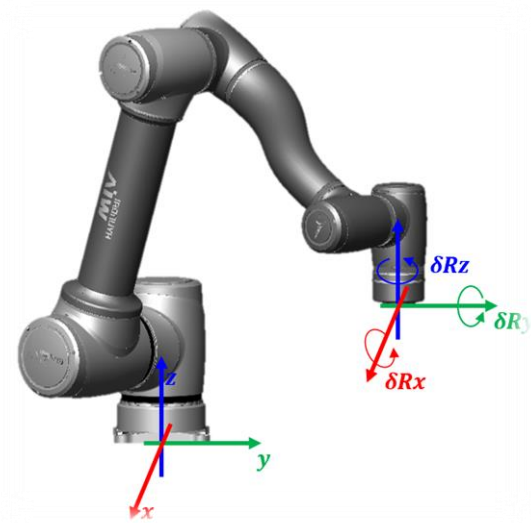



**그림 2-18 rot 회전 방향**

- rot 명령어는 그림 2-18 rot 회전 방향과 같이 로봇 종단이 현재 위치에서 BASE 좌표계의 축 중심으로 **Rx → Ry → Rz** 순서로 설정된 변위만큼 회전 이동합니다.

⚠ **Interpolation 계열의 이동 명령어는 역기구학 계산이 불가능한 특이점 영역 (Singular)에서는 움직임이 제한됩니다.**

■ 설명

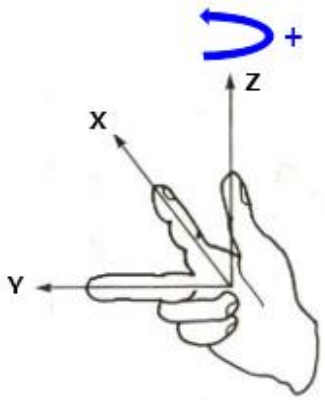



**그림 2-19 오른손 좌표계 (Right Handed Coordinate System)**

- 로봇 좌표계는 그림 2-19과 같이 오른손 법칙을 사용하며 Rx, Ry, Rz회전 방향은 반시계 방향(CCW)이 양수로(+), 시계 방향(CW)이 음수로(-) 표현 됩니다.

  * CCW : Counter-Clockwise, CW : Clockwise

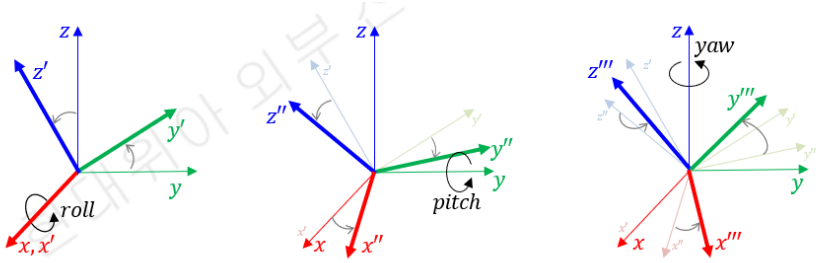



**그림 2-20 고정 좌표계 (Fixed Coordinate)**

- 로봇 좌표계 회전은 그림 2-20과 같이 고정 좌표계로 설정되어 있습니다.

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-37을 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 |

| | | | |
|---|---|---|---|
| ② 속도 명령어 | 50 % | 50 % | 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ③ 속도 옵션 | - | 10 % | |
| ④ 최종 적용 속도 | 25 % | 5 % | |

**표 2-37 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

**시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.**

■ 예 제

**[예제1] rot 회전 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=pose(500,380,500,0,0,30) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | rot(20, 30, 40) | -- Rx 20°, Ry 30°, Rz 40° 회전 이동 |
| 5 | | ※ 최종 위치 : |
| 6 | | pose(520,380,510,31.9,16.55,69.35) |
| 7 | movep(p1) | -- PTP 이동 |
| 8 | rotx(20) | -- Rx 20° 회전 이동 |
| 9 | roty(30) | -- Ry 30° 회전 이동 |
| 10 | rotz(40) | -- Rz 40° 회전 이동 |
| 11 | | ※ 최종 위치 : |
| 12 | | pose(520,380,510,31.9,16.55,69.35) |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |
| | 경로 생성 실패 | 경로 생성 실패 |

■ **관련 함수**

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.19. rotx : 회전이동 (X축 기준)

| | |
|---|---|
| ■ 기 능 | **현재 위치에서 설정한 변위만큼 Base 좌표계 X축 기준으로 회전 이동합니다.** |
| ■ 형 태 | rotx(Rx, [Speed]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| XYZ | Rx | double | - | -180.0 | 180.0 | deg |
| [Speed] | | string | 100 | 0 | 100 | % |

* 단위 : deg (degree) / % (percent)

■ **세부사항**

➢ **Rx (목표점 설정)**

- 현재 위치에서 Base 좌표계의 X축 기준으로 회전 이동할 변위를 설정합니다.

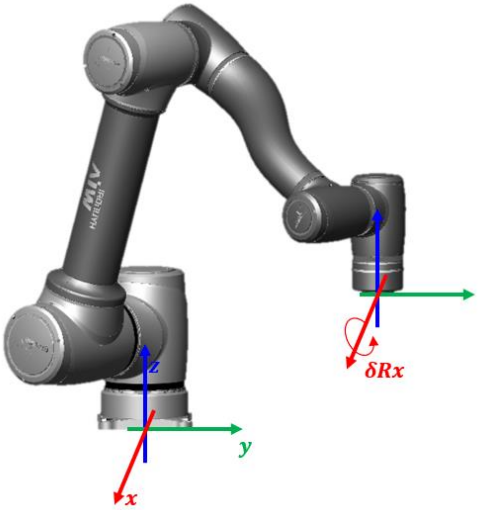



**그림 2-21 rotx 회전 방향**

- rotx 명령어는 그림 2-21와 같이 로봇 종단이 현재 위치에서 BASE 좌표계의 X축 중심으로 설정된 변위만큼 회전 이동합니다

⚠ **Interpolation 계열의 이동 명령어는 역기구학 계산이 불가능한 특이점 영역 (Singular)에서는 움직임이 제한됩니다.**

■ 설명

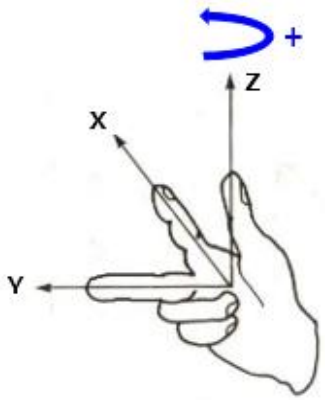



**그림 2-22 오른손 좌표계 (Right Handed Coordinate System)**

- 로봇 좌표계는 그림 2-22과 같이 오른손 법칙을 사용하며 Rx, Ry, Rz회전 방향은 반시계 방향(CCW)이 양수로(+), 시계 방향(CW)이 음수로(-) 표현 됩니다.

  * CCW : Counter-Clockwise, CW : Clockwise

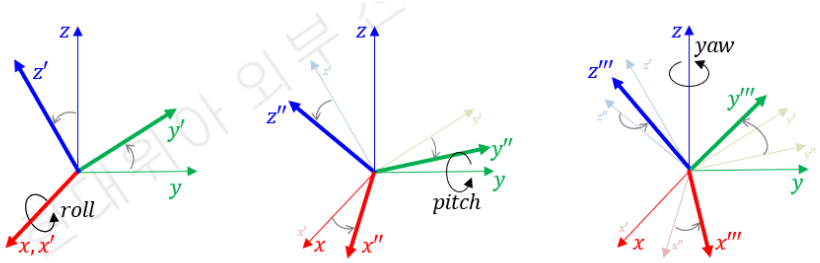



**그림 2-23 고정 좌표계 (Fixed Coordinate)**

- 로봇 좌표계 회전은 그림 2-23와 같이 고정 좌표계로 설정되어 있습니다.

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-38을 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 |

| | | | |
|---|---|---|---|
| ② **속도 명령어** | 50 % | 50 % | 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ③ **속도 옵션** | - | 10 % | |
| ④ **최종 적용 속도** | 25 % | 5 % | |

**표 2-38 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

**시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.**

## ■ 예 제

**[예제1] rotx 회전 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=pose(500,380,500,0,0,30) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | rotx(20) | -- Rx 20° 회전 이동 |
| 5 | roty(30) | -- Ry 30° 회전 이동 |
| 6 | rotz(40) | -- Rz 40° 회전 이동 |
| 7 | | ※ 최종 위치 : |
| 8 | | pose(520,380,510,31.9,16.55,69.35) |

## ■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |
| | 경로 생성 실패 | 경로 생성 실패 |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.20. roty : 회전이동 (Y축 기준)

| | |
|---|---|
| ■ 기 능 | **현재 위치에서 설정한 변위만큼 Base 좌표계 Y축 기준으로 회전 이동합니다.** |
| ■ 형 태 | roty(Ry, [Speed]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| XYZ | Ry | double | - | -180.0 | 180.0 | deg |
| [Speed] | | string | 100 | 0 | 100 | % |

* 단위 : deg (degree) / % (percent)

■ **세부사항**

➢ **Ry (목표점 설정)**

- 현재 위치에서 Base 좌표계의 Y축 기준으로 회전 이동할 변위를 설정합니다.

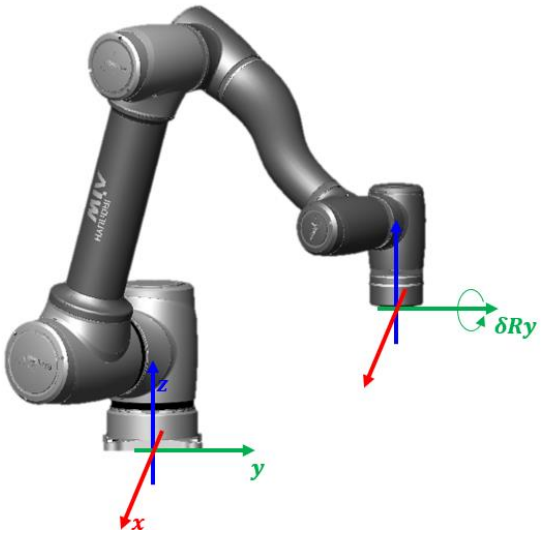



**그림 2-24 roty 회전 방향**

- roty 명령어는 그림 2-24와 같이 로봇 종단이 현재 위치에서 BASE 좌표계의 Y축 중심으로 설정된 변위만큼 회전 이동합니다

⚠ **Interpolation 계열의 이동 명령어는 역기구학 계산이 불가능한 특이점 영역 (Singular)에서는 움직임이 제한됩니다.**

■ 설명

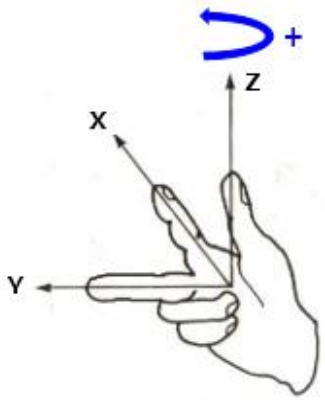



**그림 2-25 오른손 좌표계 (Right Handed Coordinate System)**

- 로봇 좌표계는 그림 2-25과 같이 오른손 법칙을 사용하며 Rx, Ry, Rz회전 방향은 반시계 방향(CCW)이 양수로(+), 시계 방향(CW)이 음수로(-) 표현 됩니다.

  * CCW : Counter-Clockwise, CW : Clockwise

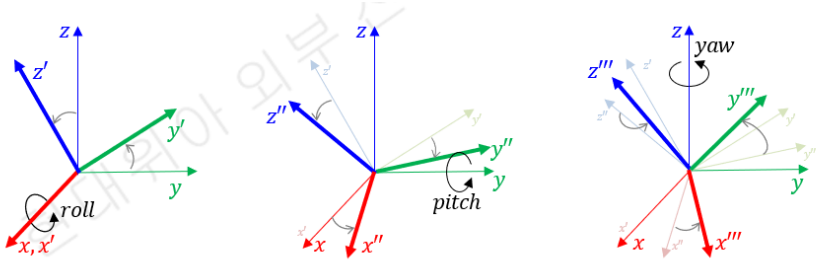



**그림 2-26 고정 좌표계 (Fixed Coordinate)**

- 로봇 좌표계 회전은 그림 2-26과 같이 고정 좌표계로 설정되어 있습니다.

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-39를 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 |

| | | | |
|---|---|---|---|
| ② **속도 명령어** | 50 % | 50 % | 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ③ **속도 옵션** | - | 10 % | |
| ④ **최종 적용 속도** | 25 % | 5 % | |

**표 2-39 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법 : ① × ② = ④ // ① × ③ = ④
* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.
* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

**시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.**

■ **예 제**

**[예제1] roty 회전 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=pose(500,380,500,0,0,30) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | rotx(20) | -- Rx 20° 회전 이동 |
| 5 | roty(30) | -- Ry 30° 회전 이동 |
| 6 | rotz(40) | -- Rz 40° 회전 이동 |
| 7 | | ※ 최종 위치 : |
| 8 | | pose(520,380,510,31.9,16.55,69.35) |

■ **에러 상황**

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |
| | 경로 생성 실패 | 경로 생성 실패 |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.21. rotz : 회전이동 (Z축 기준)

| | |
|---|---|
| ■ 기 능 | **현재 위치에서 설정한 변위만큼 Base 좌표계 Z축 기준으로 회전 이동합니다.** |
| ■ 형 태 | rotz(Rz, [Speed]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| XYZ | Rz | double | - | -180.0 | 180.0 | deg |
| [Speed] | | string | 100 | 0 | 100 | % |

* 단위 : deg (degree) / % (percent)

■ **세부사항**

➢ **Rz (목표점 설정)**

- 현재 위치에서 Base 좌표계의 Z축 기준으로 회전 이동할 변위를 설정합니다.

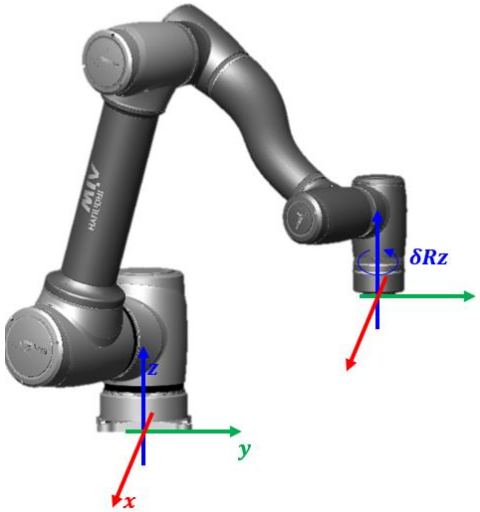



**그림 2-27 rotz 회전 방향**

- rotz 명령어는 그림 2-27과 같이 로봇 종단이 현재 위치에서 BASE 좌표계의 Z축 중심으로 설정된 변위만큼 회전 이동합니다

⚠ **Interpolation 계열의 이동 명령어는 역기구학 계산이 불가능한 특이점 영역 (Singular)에서는 움직임이 제한됩니다.**

■ 설명

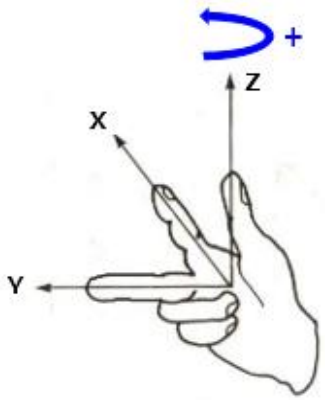



**그림 2-28 오른손 좌표계 (Right Handed Coordinate System)**

- 로봇 좌표계는 그림 2-28와 같이 오른손 법칙을 사용하며 Rx, Ry, Rz회전 방향은 반시계 방향(CCW)이 양수로(+), 시계 방향(CW)이 음수로(-) 표현 됩니다.

  * CCW : Counter-Clockwise, CW : Clockwise

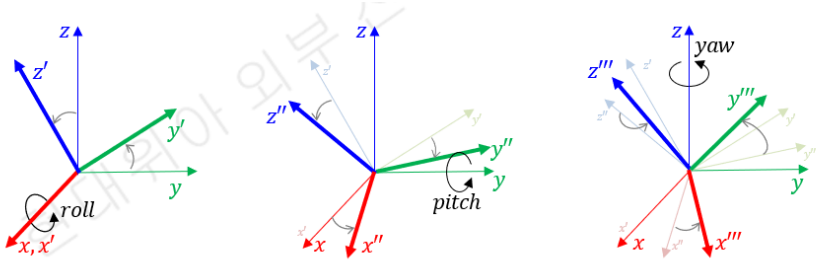



**그림 2-29 고정 좌표계 (Fixed Coordinate)**

- 로봇 좌표계 회전은 그림 2-29과 같이 고정 좌표계로 설정되어 있습니다.

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-40를 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 |

| | | | |
|---|---|---|---|
| ② **속도 명령어** | 50 % | 50 % | 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ③ **속도 옵션** | - | 10 % | |
| ④ **최종 적용 속도** | 25 % | 5 % | |

**표 2-40 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.
* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

**시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.**

## ■ 예 제

**[예제1] rotz 회전 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=pose(500,380,500,0,0,30) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | rotx(20) | -- Rx 20° 회전 이동 |
| 5 | roty(30) | -- Ry 30° 회전 이동 |
| 6 | rotz(40) | -- Rz 40° 회전 이동 |
| 7 | | ※ 최종 위치 : |
| 8 | | pose(520,380,510,31.9,16.55,69.35) |

## ■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |
| | 경로 생성 실패 | 경로 생성 실패 |

■ **관련 함수**

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.22. rotj1 : 회전이동 (관절 1축 기준)

| | |
|---|---|
| ■ 기 능 | **현재 위치에서 설정한 변위만큼 관절 1축 기준으로 회전 이동합니다.** |
| ■ 형 태 | rotj1(J1, [Speed]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| Joint | J1 | double | - | -360.0 | 360.0 | deg |
| [Speed] | | string | 100 | 0 | 100 | % |

* 단위 : deg (degree) / % (percent)

■ **세부사항**

➢ **J1 (각도 설정)**

- 현재 위치에서 관절 좌표계의 1축 기준으로 회전 이동할 변위를 설정합니다.

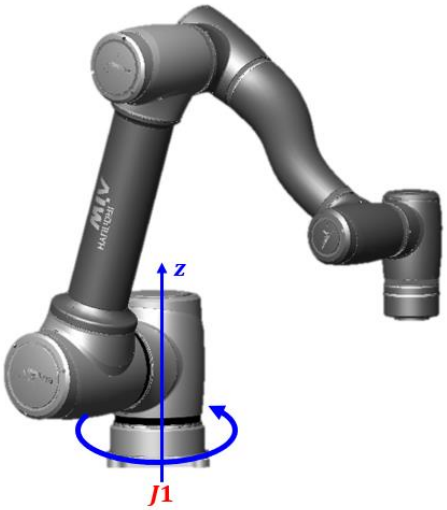



**그림 2-30 rotj1 회전 방향**

- rotj1 명령어는 그림 2-30과 같이 현재 위치에서 관절 좌표계의 1번 축 중심으로 설정된 변위만큼 회전 이동합니다

■ 설명

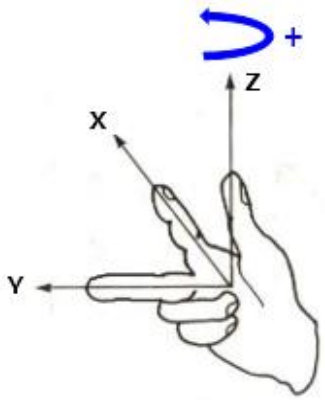



**그림 2-31 오른손 좌표계 (Right Handed Coordinate System)**

- 로봇 좌표계는 그림 2-31와 같이 오른손 법칙을 사용하며 Rx, Ry, Rz회전 방향은 반시계 방향(CCW)이 양수(+) 값으로, 시계 방향(CW)이 음수(-) 값으로 표현 됩니다.

  * CCW : Counter-Clockwise, CW : Clockwise

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-41을 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ② 속도 명령어 | 50 % | 50 % | |
| ③ 속도 옵션 | - | 10 % | |
| ④ 최종 적용 속도 | 25 % | 5 % | |

**표 2-41 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

ⓘ **시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드**

| | |
|---|---|
| | 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다. |

■ 예 제

**[예제1] rotj1 회전 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=joint(30,30,90,60,0,0) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | rotj1(60) | -- J1 60º 회전 이동 |
| 5 | | ※ 최종 위치 : joint(90,30,90,60,0,0) |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |
| | 경로 생성 실패 | 경로 생성 실패 |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.23. rotj2 : 회전이동 (관절 2축 기준)

| | |
|---|---|
| ■ 기 능 | **현재 위치에서 설정한 변위만큼 관절 좌표계 2축 기준으로 회전 이동합니다.** |
| ■ 형 태 | rotj2(J2, [Speed]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| Joint | J2 | double | - | -360.0 | 360.0 | deg |
| [Speed] | | string | 100 | 0 | 100 | % |

* 단위 : deg (degree) / % (percent)

■ **세부사항**

➢ **J2 (각도 설정)**

- 현재 위치에서 관절 좌표계의 2축 기준으로 회전 이동할 변위를 설정합니다.

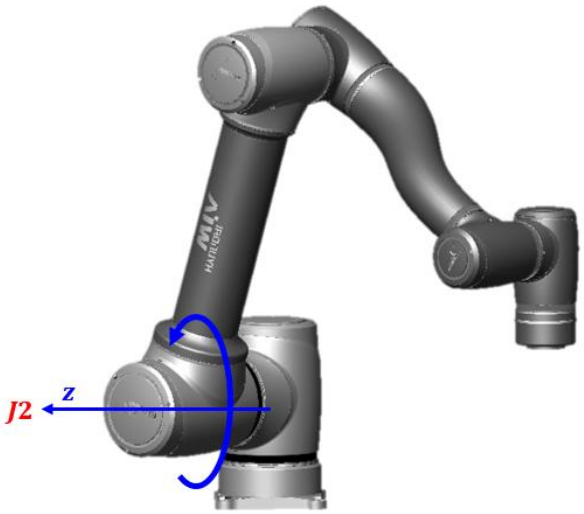



**그림 2-32 rotj2 회전 방향**

- rotj2 명령어는 그림 2-32과 같이 현재 위치에서 관절 좌표계의 2번 축 중심으로 설정된 변위만큼 회전 이동합니다

■ 설명

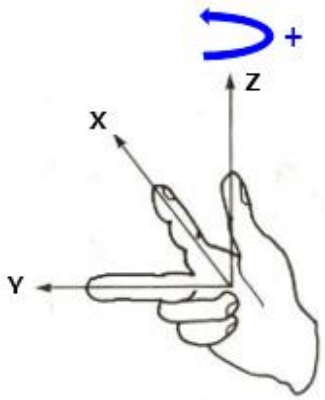



**그림 2-33 오른손 좌표계 (Right Handed Coordinate System)**

- 로봇 좌표계는 그림 2-33와 같이 오른손 법칙을 사용하며 Rx, Ry, Rz회전 방향은 반시계 방향(CCW)이 양수(+) 값으로, 시계 방향(CW)이 음수(-) 값으로 표현 됩니다.

  * CCW : Counter-Clockwise, CW : Clockwise

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-42를 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ② 속도 명령어 | 50 % | 50 % | |
| ③ 속도 옵션 | - | 10 % | |
| ④ 최종 적용 속도 | 25 % | 5 % | |

**표 2-42 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

ⓘ **시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드**

속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.

■ 예 제

**[예제1] rotj2 회전 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=joint(30,30,90,60,0,0) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | rotj2(-10) | -- J2 -10º 회전 이동 |
| 5 | | ※ 최종 위치 : joint(30,20,90,60,0,0) |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |
| | 경로 생성 실패 | 경로 생성 실패 |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.24. rotj3 : 회전이동 (관절 3축 기준)

| | |
|---|---|
| ■ 기 능 | **현재 위치에서 설정한 변위만큼 관절 좌표계 3번 축 중심으로 회전 이동합니다.** |
| ■ 형 태 | rotj3(J3, [Speed]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| Joint | J3 | double | - | -360.0 | 360.0 | deg |
| [Speed] | | string | 100 | 0 | 100 | % |

* 단위 : deg (degree) / % (percent)

■ **세부사항**

➢ **J3 (각도 설정)**

- 현재 위치에서 관절 좌표계의 3번 축 중심으로 회전 이동할 변위를 설정합니다.

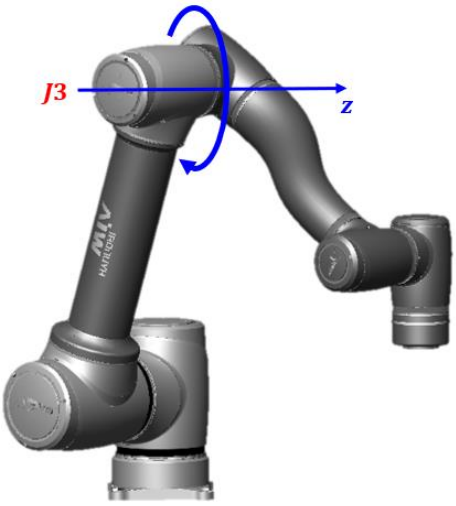



**그림 2-34 rotj3 회전 방향**

- rotj3 명령어는 그림 2-34와 같이 현재 위치에서 관절 좌표계의 3번 축 중심으로 설정된 변위만큼 회전 이동합니다

■ 설명

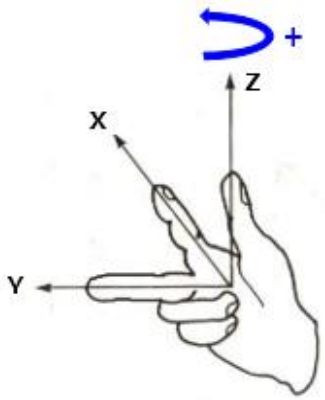



**그림 2-35 오른손 좌표계 (Right Handed Coordinate System)**

- 로봇 좌표계는 그림 2-35과 같이 오른손 법칙을 사용하며 Rx, Ry, Rz회전 방향은 반시계 방향(CCW)이 양수(+) 값으로, 시계 방향(CW)이 음수(-) 값으로 표현 됩니다.

  * CCW : Counter-Clockwise, CW : Clockwise

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-43을 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ② 속도 명령어 | 50 % | 50 % | |
| ③ 속도 옵션 | - | 10 % | |
| ④ 최종 적용 속도 | 25 % | 5 % | |

**표 2-43 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

ⓘ **시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드**

속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.

■ 예 제

**[예제1] rotj3 회전 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=joint(30,30,90,60,0,0) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | rotj3(-30) | -- J3 -30º 회전 이동 |
| 5 | | ※ 최종 위치 : joint(30,30,90,60,0,0) |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |
| | 경로 생성 실패 | 경로 생성 실패 |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.25. rotj4 : 회전이동 (관절 4축 기준)

| | |
|---|---|
| ■ 기 능 | **현재 위치에서 설정한 변위만큼 관절 좌표계 4축 기준으로 회전 이동합니다.** |
| ■ 형 태 | rotj4(J4, [Speed]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| Joint | J4 | double | - | -360.0 | 360.0 | deg |
| [Speed] | | string | 100 | 0 | 100 | % |

* 단위 : deg (degree) / % (percent)

■ **세부사항**

➢ **J4 (각도 설정)**

- 현재 위치에서 관절 좌표계의 4축 기준으로 회전 이동할 변위를 설정합니다.

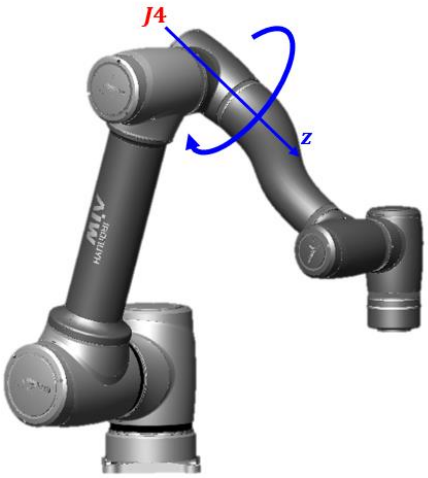



**그림 2-36 rotj4 회전 방향**

- rotj4 명령어는 그림 2-36과 같이 현재 위치에서 관절 좌표계의 4번 축 중심으로 설정된 변위만큼 회전 이동합니다

■ 설명

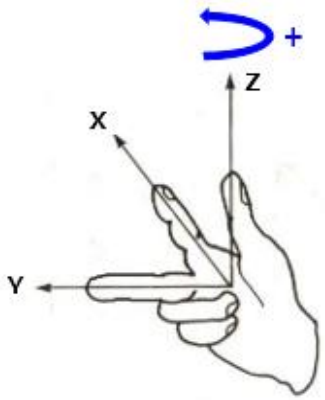



**그림 2-37 오른손 좌표계 (Right Handed Coordinate System)**

- 로봇 좌표계는 그림 2-37과 같이 오른손 법칙을 사용하며 Rx, Ry, Rz회전 방향은 반시계 방향(CCW)이 양수(+) 값으로, 시계 방향(CW)이 음수(-) 값으로 표현 됩니다.

  * CCW : Counter-Clockwise, CW : Clockwise

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-44를 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ② 속도 명령어 | 50 % | 50 % | |
| ③ 속도 옵션 | - | 10 % | |
| ④ 최종 적용 속도 | 25 % | 5 % | |

**표 2-44 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

ⓘ **시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드**

| | |
|---|---|
| | **속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.** |

■ **예 제**

**[예제1] rotj4 회전 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=joint(30,30,90,60,0,0) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | rotj4(30) | -- J4 30º 회전 이동 |
| 5 | | ※ 최종 위치 : joint(30,30,90,90,0,0) |

■ **에러 상황**

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |
| | 경로 생성 실패 | 경로 생성 실패 |

■ **관련 함수**

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.26. rotj5 : 회전이동 (관절 5축 기준)

| | |
|---|---|
| ■ 기 능 | **현재 위치에서 설정한 변위만큼 관절 좌표계 5축 기준으로 회전 이동합니다.** |
| ■ 형 태 | rotj5(J5, [Speed]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| Joint | J5 | double | - | -360.0 | 360.0 | deg |
| [Speed] | | string | 100 | 0 | 100 | % |

* 단위 : deg (degree) / % (percent)

■ **세부사항**

➢ **J5 (각도 설정)**

- 현재 위치에서 관절 좌표계의 5축 기준으로 회전 이동할 변위를 설정합니다.

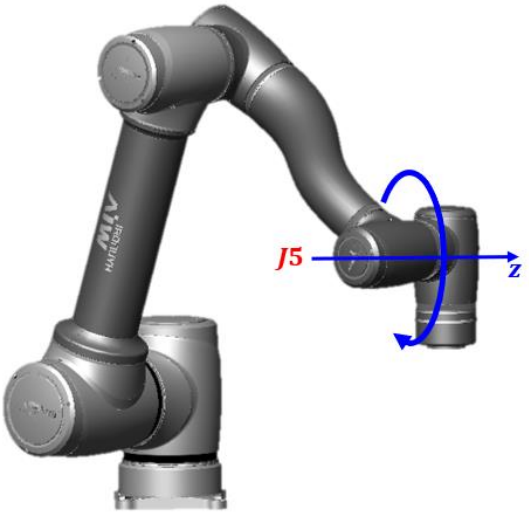



**그림 2-38 rotj5 회전 방향**

- rotj5 명령어는 그림 2-38와 같이 현재 위치에서 관절 좌표계의 5번 축 중심으로 설정된 변위만큼 회전 이동합니다

■ 설명

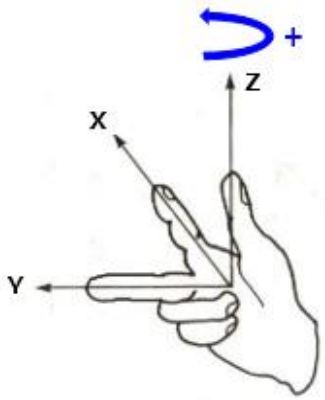



**그림 2-39 오른손 좌표계 (Right Handed Coordinate System)**

- 로봇 좌표계는 그림 2-39과 같이 오른손 법칙을 사용하며 Rx, Ry, Rz회전 방향은 반시계 방향(CCW)이 양수(+) 값으로, 시계 방향(CW)이 음수(-) 값으로 표현 됩니다.

  * CCW : Counter-Clockwise, CW : Clockwise

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-45를 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ② 속도 명령어 | 50 % | 50 % | |
| ③ 속도 옵션 | - | 10 % | |
| ④ 최종 적용 속도 | 25 % | 5 % | |

**표 2-45 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

ⓘ **시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드**

| | |
|---|---|
| | **속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.** |

■ **예 제**

**[예제1] rotj5 회전 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=joint(30,30,90,60,0,0) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | rotj5(60) | -- J5 60º 회전 이동 |
| 5 | | ※ 최종 위치 : joint(30,30,90,60,60,0) |

■ **에러 상황**

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |
| | 경로 생성 실패 | 경로 생성 실패 |

■ **관련 함수**

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.27. rotj6 : 회전이동 (관절 6축 기준)

| | |
|---|---|
| ■ 기 능 | **현재 위치에서 설정한 변위만큼 관절 좌표계 6 축 기준으로 회전 이동합니다.** |
| ■ 형 태 | rotj6(J6, [Speed]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| Joint | J6 | double | - | -360.0 | 360.0 | deg |
| [Speed] | | string | 100 | 0 | 100 | % |

* 단위 : deg (degree) / % (percent)

■ **세부사항**

➢ **J6 (각도 설정)**

- 현재 위치에서 관절 좌표계의 6축 기준으로 회전 이동할 변위를 설정합니다.

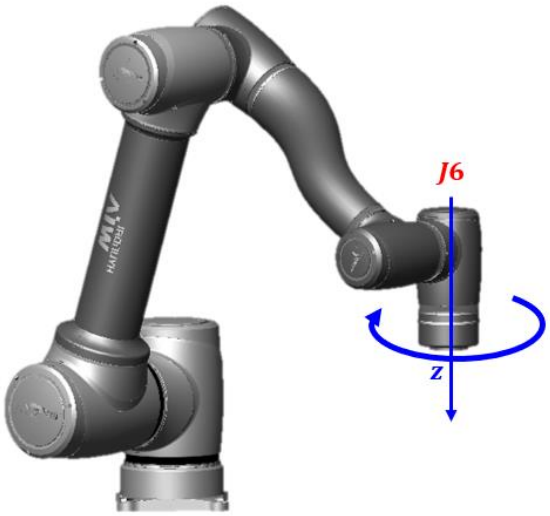



**그림 2-40 rotj6 회전 방향**

- rotj6 명령어는 그림 2-40과 같이 현재 위치에서 관절 좌표계의 6번 축 중심으로 설정된 변위만큼 회전 이동합니다

■ 설명

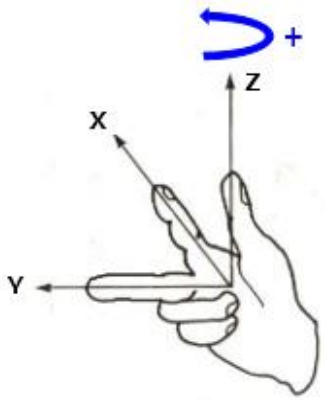



**그림 2-41 오른손 좌표계 (Right Handed Coordinate System)**

- 로봇 좌표계는 그림 2-41와 같이 오른손 법칙을 사용하며 Rx, Ry, Rz회전 방향은 반시계 방향(CCW)이 양수(+) 값으로, 시계 방향(CW)이 음수(-) 값으로 표현 됩니다.

  * CCW : Counter-Clockwise, CW : Clockwise

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-46을 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ② 속도 명령어 | 50 % | 50 % | |
| ③ 속도 옵션 | - | 10 % | |
| ④ 최종 적용 속도 | 25 % | 5 % | |

**표 2-46 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

ⓘ **시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드**

속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.

■ 예 제

**[예제1] rotj6 회전 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=joint(30,30,90,60,0,0) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | rotj6(60) | -- J6 60º 회전 이동 |
| 5 | | ※ 최종 위치 : joint(30,30,90,60,0,60) |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |
| | 경로 생성 실패 | 경로 생성 실패 |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.28. tool.trans : 툴 좌표계 기준 병진이동

| | |
|---|---|
| ■ 기 능 | **현재 Tool 좌표계 기준으로 설정한 변위만큼 이동합니다.** |
| ■ 형 태 | tool.trans(X, Y, Z, [Speed]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| XYZ | X | double | - | -1700.0 | 1700.0 | mm |
| | Y | double | - | -1700.0 | 1700.0 | mm |
| | Z | double | - | -1700.0 | 1700.0 | mm |
| [Speed] | | string | 100 | 0 | 100 | % |

* 단위 : mm (millimeter) / % (percent)

■ **세부사항**

➢ **X, Y, Z (목표점 설정)**

- 현재 Tool 좌표계 기준 이동량 설정 (mm)

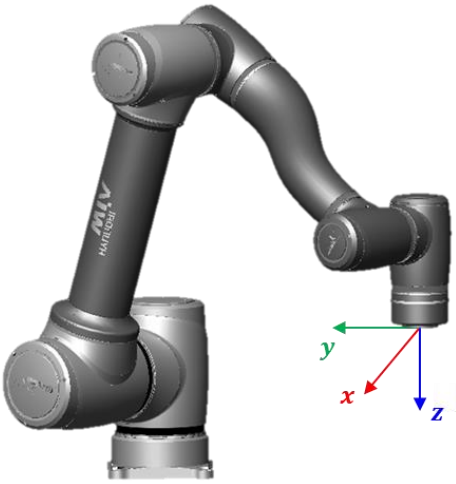



**그림 2-42 tool.trans 이동 방향**

- tool.trans 명령어는 현재 위치에서 설정한 Tool 좌표계 기준으로 설정된 변위만큼 로봇 종단이 그림 2-42과 같이 이동합니다.

⚠ **Interpolation 계열의 이동 명령어는 역기구학 계산이 불가능한 특이점 영역 (Singular)에서는 움직임이 제한됩니다.**

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는

"5%"가 됩니다.

- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-47을 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ② 속도 명령어 | 50 % | 50 % | |
| ③ 속도 옵션 | - | 10 % | |
| ④ 최종 적용 속도 | 25 % | 5 % | |

**표 2-47 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

**시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.**

■ 예 제

**[예제1] tool.trans 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=joint(0,0,90,0,45,45) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | trans(100,50,200) | -- Base 좌표계 기준 병진이동 |
| 5 | | ※ 최종 위치 : |
| 6 | | pose(751.23, 98, 864.47, 144.74, 30, |
| 7 | | 125.26) |
| 8 | | |
| 9 | movep(p1) | -- PTP 이동 |
| 10 | tool.trans(100,50,200) | -- Tool 좌표계 기준 병진이동 |
| 11 | | ※ 최종 위치 : |
| 12 | | pose(767.65, 154.07, 498.05, 144.74, |
| 13 | | 30, 125.26) |

■ **에러 상황**

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |
| | 경로 생성 실패 | 경로 생성 실패 |

■ **관련 함수**

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.29. tool.transx : 툴 좌표계 X축 기준 병진이동

| | |
|---|---|
| ■ 기 능 | **현재 Tool 좌표계의 X축 기준으로 설정한 변위만큼 이동합니다.** |
| ■ 형 태 | tool.transx(X, [Speed]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| XYZ | X | double | - | -1700.0 | 1700.0 | mm |
| [Speed] | | string | 100 | 0 | 100 | % |

* 단위 : mm (millimeter) / % (percent)

■ **세부사항**

➢ **X (목표점 설정)**

- 현재 Tool 좌표계의 X축 기준 이동량 설정 (mm)

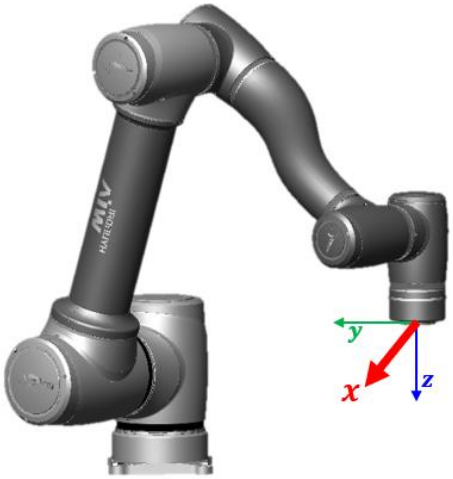



**그림 2-43 tool.transx 이동 방향**

- tool.transx 명령어는 현재 위치에서 설정한 Tool 좌표계 기준으로 x 변위만큼 로봇 종단이 그림 2-43와 같이 이동합니다.

⚠ **Interpolation 계열의 이동 명령어는 역기구학 계산이 불가능한 특이점 영역 (Singular)에서는 움직임이 제한됩니다.**

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.

- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-48을 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ② 속도 명령어 | 50 % | 50 % | |
| ③ 속도 옵션 | - | 10 % | |
| ④ 최종 적용 속도 | 25 % | 5 % | |

**표 2-48 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

**시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.**

■ 예 제

**[예제1] tool.transx 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=joint(0,0,90,0,45,45) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | transx(100) | -- Base 좌표계 X축 기준 병진이동 |
| 5 | | ※ 최종 위치 : |
| 6 | | pose(751.23, 48, 664.47, 144.74, 30, |
| 7 | | 125.26) |
| 8 | | |
| 9 | movep(p1) | -- PTP 이동 |
| 10 | tool.transx(100) | -- Tool 좌표계 X축 기준 병진이동 |
| 11 | | ※ 최종 위치 : |
| 12 | | pose(601.23, 118.71, 614.47, 144.74, |
| 13 | | 30, 125.26) |

■ **에러 상황**

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |
| | 경로 생성 실패 | 경로 생성 실패 |

■ **관련 함수**

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.30. tool.transy : 툴 좌표계 기준 Y축 병진이동

| | |
|---|---|
| ■ 기 능 | **현재 Tool 좌표계의 Y축 기준으로 설정한 변위만큼 이동합니다.** |
| ■ 형 태 | tool.transy(Y, [Speed]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| XYZ | Y | double | - | -1700.0 | 1700.0 | mm |
| [Speed] | | string | 100 | 0 | 100 | % |

* 단위 : mm (millimeter) / % (percent)

■ **세부사항**

➢ **Y (목표점 설정)**

- 현재 Tool 좌표계의 Y축 기준 이동량 설정 (mm)

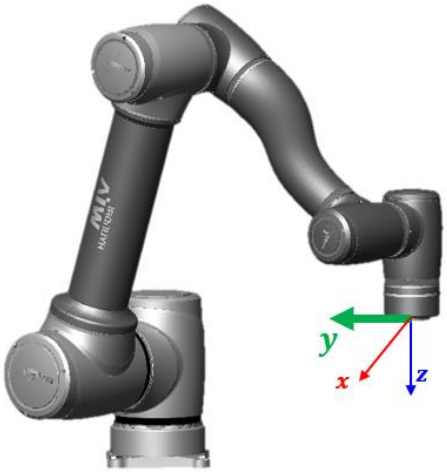



**그림 2-44 tool.transy 이동 방향**

- tool.transy 명령어는 현재 위치에서 설정한 Tool 좌표계 기준으로 y 변위만큼 로봇 종단이 그림 2-44와 같이 이동합니다.

⚠ **Interpolation 계열의 이동 명령어는 역기구학 계산이 불가능한 특이점 영역 (Singular)에서는 움직임이 제한됩니다.**

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만

적용됩니다.

- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-49를 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ② 속도 명령어 | 50 % | 50 % | |
| ③ 속도 옵션 | - | 10 % | |
| ④ 최종 적용 속도 | 25 % | 5 % | |

**표 2-49 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

**시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.**

■ 예 제

**[예제1] tool.transy 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=joint(0,0,90,0,45,45) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | transy(100) | -- Base 좌표계 Y축 기준 병진이동 |
| 5 | | ※ 최종 위치 : |
| 6 | | pose(651.23, 148, 664.47, 144.74, 30, |
| 7 | | 125.26) |
| 8 | | |
| 9 | movep(p1) | -- PTP 이동 |
| 10 | tool.transy(100) | -- Tool 좌표계 Y축 기준 병진이동 |
| 11 | | ※ 최종 위치 : |
| 12 | | pose(701.23, 118.71, 714.47, 144.74, |
| 13 | | 30, 125.26) |

■ **에러 상황**

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |
| | 경로 생성 실패 | 경로 생성 실패 |

■ **관련 함수**

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.31. tool.transz : 툴 좌표계 기준 Z축 병진이동

| | |
|---|---|
| ■ 기 능 | **현재 Tool 좌표계의 Z축 기준으로 설정한 변위만큼 이동합니다.** |
| ■ 형 태 | tool.transz(Z, [Speed]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| XYZ | Z | double | - | -1700.0 | 1700.0 | mm |
| [Speed] | | string | 100 | 0 | 100 | % |

* 단위 : mm (millimeter) / % (percent)

■ **세부사항**

➢ **Z (목표점 설정)**

- 현재 Tool 좌표계의 Z축 기준 이동량 설정 (mm)

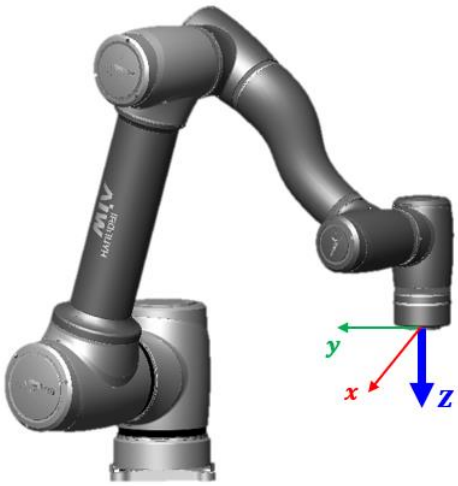



**그림 2-45 tool.transz 이동 방향**

- tool.transz 명령어는 현재 위치에서 설정한 Tool 좌표계 기준으로 z 변위만큼 로봇 종단이 그림 2-45과 같이 이동합니다.

⚠ **Interpolation 계열의 이동 명령어는 역기구학 계산이 불가능한 특이점 영역 (Singular)에서는 움직임이 제한됩니다.**

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.

- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-50을 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ② 속도 명령어 | 50 % | 50 % | |
| ③ 속도 옵션 | - | 10 % | |
| ④ 최종 적용 속도 | 25 % | 5 % | |

**표 2-50 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

**시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.**

■ 예 제

**[예제1] tool.transz 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=joint(0,0,90,0,45,45) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | transz(100) | -- Base 좌표계 Z축 기준 병진이동 |
| 5 | | ※ 최종 위치 : |
| 6 | | pose(651.23, 48, 764.47,, 30, 125.26) |
| 7 | | |
| 8 | movep(p1) | -- PTP 이동 |
| 9 | tool.transz(100) | -- Tool 좌표계 Z축 기준 병진이동 |
| 10 | | ※ 최종 위치 : |
| 11 | | pose(721.94, 48, 593.76, 144.74, 30, |
| 12 | | 125.26) |

■ **에러 상황**

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |
| | 경로 생성 실패 | 경로 생성 실패 |

■ **관련 함수**

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.32. tool.rot : 툴 좌표계 기준 회전이동

| | |
|---|---|
| ■ 기 능 | **현재 위치에서 설정한 회전 변위만큼 Tool 좌표계 기준으로 회전합니다.** |
| ■ 형 태 | tool.rot(Rx, Ry, Rz, [Speed]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| XYZ | Rx | double | - | -180.0 | 180.0 | deg |
| | Ry | double | - | -180.0 | 180.0 | deg |
| | Rz | double | - | -180.0 | 180.0 | deg |
| [Speed] | | string | 100 | 0 | 100 | % |

* 단위 : deg (degree) / % (percent)

■ **세부사항**

➢ **Rx, Ry, Rz (목표점 설정)**

- 현재 위치에서 Tool 좌표계의 축 중심으로 회전 기준할 변위를 설정합니다.

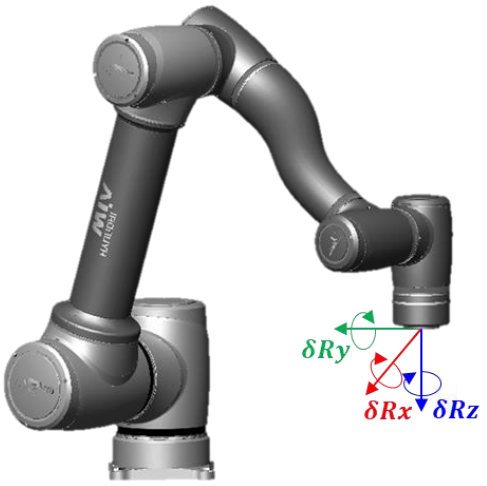



**그림 2-46 tool.rot 회전 방향**

- tool.rot 명령어는 그림 2-46과 같이 로봇 종단이 현재 위치에서 Tool 좌표계의 축 중심으로 **Rx → Ry → Rz** 순서로 설정된 변위만큼 회전 이동합니다.

⚠ **Interpolation 계열의 이동 명령어는 역기구학 계산이 불가능한 특이점 영역 (Singular)에서는 움직임이 제한됩니다.**

■ 설명

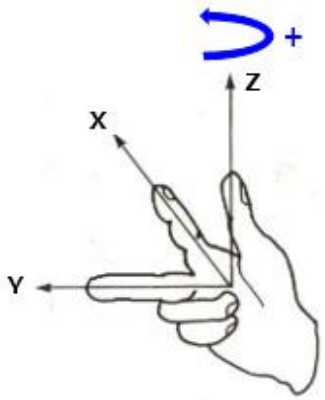



**그림 2-47 오른손 좌표계 (Right Handed Coordinate System)**

- 로봇 좌표계는 그림 2-47과 같이 오른손 법칙을 사용하며 Rx, Ry, Rz회전 방향은 반시계 방향(CCW)이 양수로(+), 시계 방향(CW)이 음수로(-) 표현 됩니다.

  * CCW : Counter-Clockwise, CW : Clockwise

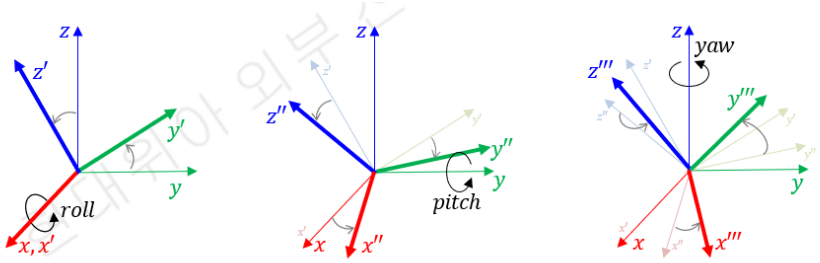



**그림 2-48 고정 좌표계 (Fixed Coordinate)**

- 로봇 좌표계 회전은 그림 2-48와 같이 고정 좌표계로 설정되어 있습니다.

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-51을 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 |

| | | | |
|---|---|---|---|
| ② **속도 명령어** | 50 % | 50 % | 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ③ **속도 옵션** | - | 10 % | |
| ④ **최종 적용 속도** | 25 % | 5 % | |

**표 2-51 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

**시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.**

■ **예 제**

**[예제1] tool.rot 회전 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=joint(0,0,90,0,45,45) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | rot(25, 30, 10) | -- Base 좌표계 기준 회전이동 |
| 5 | | ※ 최종 위치 : |
| 6 | | pose(651.23, 48, 664.47, 156.17, -6.68, |
| 7 | | 130.91) |
| 8 | | |
| 9 | movep(p1) | -- PTP 이동 |
| 10 | tool.rot(25, 30, 10) | -- Tool 좌표계 기준 회전이동 |
| 11 | | ※ 최종 위치 : |
| 12 | | pose(651.23, 48, 664.47, 169.6, -0.13, |
| 13 | | 134.81) |

■ **에러 상황**

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |
| | 경로 생성 실패 | 경로 생성 실패 |

■ **관련 함수**

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.33. tool.rotx : 툴 좌표계 X축 기준 회전이동

| | |
|---|---|
| ■ 기 능 | **현재 위치에서 설정한 변위만큼 Tool 좌표계 X축 기준으로 회전 이동합니다.** |
| ■ 형 태 | tool.rotx(Rx, [Speed]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| XYZ | Rx | double | - | -180.0 | 180.0 | deg |
| [Speed] | | string | 100 | 0 | 100 | % |

* 단위 : deg (degree) / % (percent)

■ **세부사항**

➢ **Rx (목표점 설정)**

- 현재 위치에서 Tool 좌표계의 X축 기준으로 회전 이동할 변위를 설정합니다.

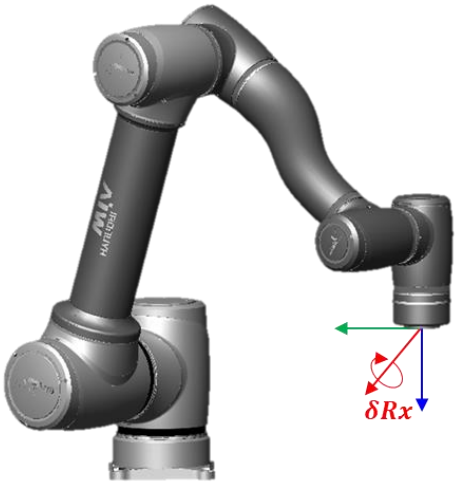



**그림 2-49 tool.rotx 회전 방향**

- tool.rotx 명령어는 그림 2-49과 같이 로봇 종단이 현재 위치에서 Tool 좌표계의 X축 중심으로 설정된 변위만큼 회전 이동합니다

⚠ **Interpolation 계열의 이동 명령어는 역기구학 계산이 불가능한 특이점 영역 (Singular)에서는 움직임이 제한됩니다.**

■ 설명

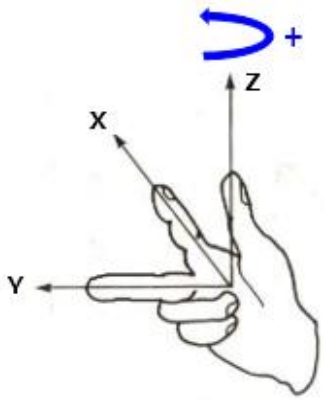



**그림 2-50 오른손 좌표계 (Right Handed Coordinate System)**

- 로봇 좌표계는 그림 2-50과 같이 오른손 법칙을 사용하며 Rx, Ry, Rz회전 방향은 반시계 방향(CCW)이 양수로(+), 시계 방향(CW)이 음수로(-) 표현 됩니다.

  * CCW : Counter-Clockwise, CW : Clockwise

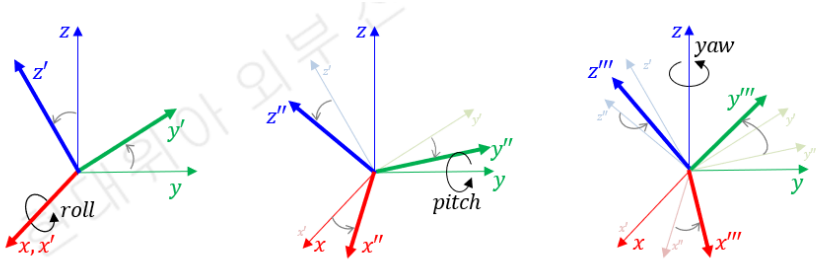



**그림 2-51 고정 좌표계 (Fixed Coordinate)**

- 로봇 좌표계 회전은 그림 2-51와 같이 고정 좌표계로 설정되어 있습니다.

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-52를 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 |

| | | | |
|---|---|---|---|
| ② **속도 명령어** | 50 % | 50 % | 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ③ **속도 옵션** | - | 10 % | |
| ④ **최종 적용 속도** | 25 % | 5 % | |

**표 2-52 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

**시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.**

■ **예 제**

**[예제1] tool.rotx 회전 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=joint(0,0,90,0,45,45) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | rotx(25) | -- Base 좌표계 X축 기준 회전이동 |
| 5 | | ※ 최종 위치 : |
| 6 | | pose(651.23, 48, 664.47, 130.44, 8.88, |
| 7 | | 120.4) |
| 8 | | |
| 9 | movep(p1) | -- PTP 이동 |
| 10 | tool.rotx(25) | -- Tool 좌표계 X축 기준 회전이동 |
| 11 | | ※ 최종 위치 : |
| 12 | | pose(651.23, 48, 664.47, 169.74, 30, |
| 13 | | 125.26) |

■ **에러 상황**

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |
| | 경로 생성 실패 | 경로 생성 실패 |

■ **관련 함수**

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.34. tool.roty : 툴 좌표계 기준 Y축 회전이동

| | |
|---|---|
| ■ 기 능 | **현재 위치에서 설정한 변위만큼 Tool 좌표계 Y축 기준으로 회전 이동합니다.** |
| ■ 형 태 | tool.roty(Ry, [Speed]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| XYZ | Ry | double | - | -180.0 | 180.0 | deg |
| [Speed] | | string | 100 | 0 | 100 | % |

* 단위 : deg (degree) / % (percent)

■ **세부사항**

➢ **Ry (목표점 설정)**

- 현재 위치에서 Tool 좌표계의 Y축 기준으로 회전 이동할 변위를 설정합니다.

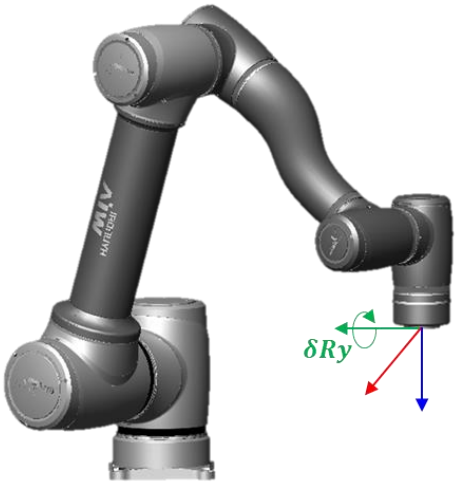



**그림 2-52 tool.roty 회전 방향**

- tool.roty 명령어는 그림 2-52과 같이 로봇 종단이 현재 위치에서 Tool 좌표계의 Y축 중심으로 설정된 변위만큼 회전 이동합니다

⚠ **Interpolation 계열의 이동 명령어는 역기구학 계산이 불가능한 특이점 영역 (Singular)에서는 움직임이 제한됩니다.**

■ 설명

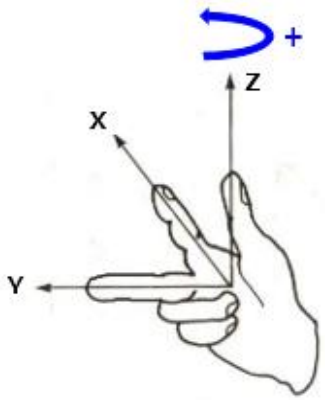



**그림 2-53 오른손 좌표계 (Right Handed Coordinate System)**

- 로봇 좌표계는 그림 2-53와 같이 오른손 법칙을 사용하며 Rx, Ry, Rz회전 방향은 반시계 방향(CCW)이 양수로(+), 시계 방향(CW)이 음수로(-) 표현 됩니다.

  * CCW : Counter-Clockwise, CW : Clockwise

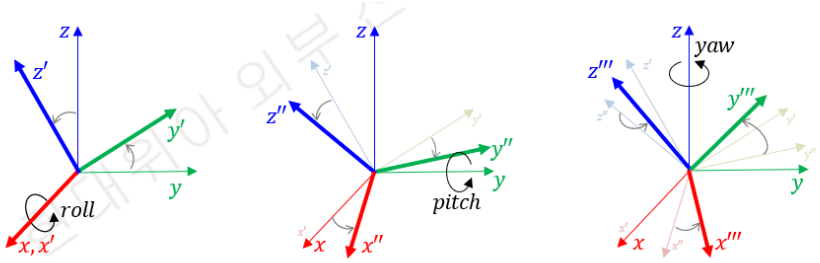



**그림 2-54 고정 좌표계 (Fixed Coordinate)**

- 로봇 좌표계 회전은 그림 2-54와 같이 고정 좌표계로 설정되어 있습니다.

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-53을 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 |

| | | | |
|---|---|---|---|
| ② **속도 명령어** | 50 % | 50 % | 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ③ **속도 옵션** | - | 10 % | |
| ④ **최종 적용 속도** | 25 % | 5 % | |

**표 2-53 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

**시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.**

■ **예 제**

**[예제1] tool.roty 회전 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=joint(0,0,90,0,45,45) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | roty(30) | -- Base 좌표계 Y축 기준 회전이동 |
| 5 | | ※ 최종 위치 : |
| 6 | | pose(651.23, 48, 664.47, 169.27, 10.55, |
| 7 | | 134.01) |
| 8 | | |
| 9 | movep(p1) | -- PTP 이동 |
| 10 | tool.roty(30) | -- Tool 좌표계 Y축 기준 회전이동 |
| 11 | | ※ 최종 위치 : |
| 12 | | pose(651.23, 48, 664.47, 149.9, 4.56, |
| 13 | | 142.1) |

■ **에러 상황**

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |
| | 경로 생성 실패 | 경로 생성 실패 |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.35. tool.rotz : 툴 좌표계 기준 Z축 회전이동

| | |
|---|---|
| ■ 기 능 | **현재 위치에서 설정한 변위만큼 Tool 좌표계 Z축 기준으로 회전 이동합니다.** |
| ■ 형 태 | tool.rotz(Rz, [Speed]) |

■ 인 자

| 인자 | | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|---|
| 좌표계 | 축 | | | 최소 | 최대 | |
| XYZ | Rz | double | - | -180.0 | 180.0 | deg |
| [Speed] | | string | 100 | 0 | 100 | % |

* 단위 : deg (degree) / % (percent)

■ **세부사항**

➢ **Rz (목표점 설정)**

- 현재 위치에서 Tool 좌표계의 Z축 기준으로 회전 이동할 변위를 설정합니다.

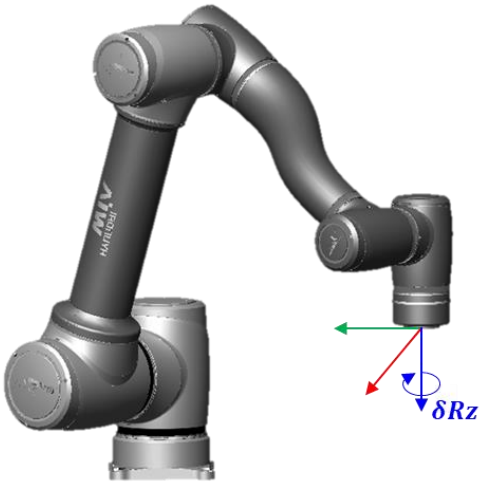



**그림 2-55 tool.rotz 회전 방향**

- tool.rotz 명령어는 그림 2-55과 같이 로봇 종단이 현재 위치에서 Tool 좌표계의 Z축 중심으로 설정된 변위만큼 회전 이동합니다

⚠ **Interpolation 계열의 이동 명령어는 역기구학 계산이 불가능한 특이점 영역 (Singular)에서는 움직임이 제한됩니다.**

■ 설명

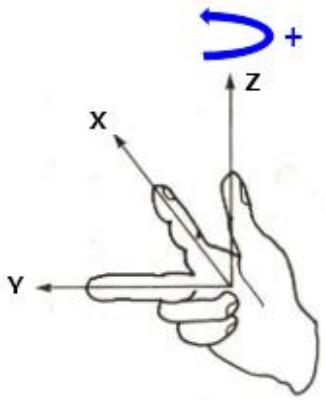



**그림 2-56 오른손 좌표계 (Right Handed Coordinate System)**

- 로봇 좌표계는 그림 2-56과 같이 오른손 법칙을 사용하며 Rx, Ry, Rz회전 방향은 반시계 방향(CCW)이 양수로(+), 시계 방향(CW)이 음수로(-) 표현 됩니다.

  * CCW : Counter-Clockwise, CW : Clockwise

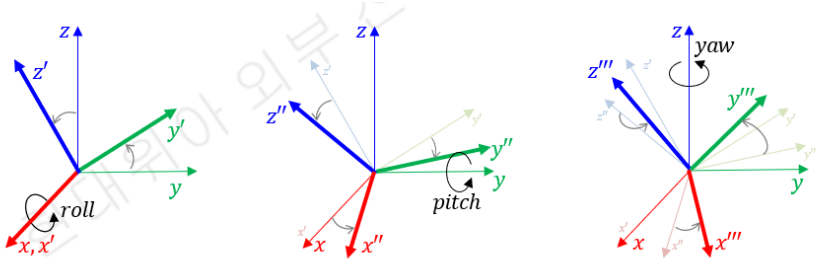



**그림 2-57 고정 좌표계 (Fixed Coordinate)**

- 로봇 좌표계 회전은 그림 2-57과 같이 고정 좌표계로 설정되어 있습니다.

➢ **[Speed] (상대 속도 설정)**

- 해당 인자는 옵션 인자입니다.
- 로딩 작업과 같이 구간마다 로봇의 속도를 조절하는 경우 사용됩니다.
- 시스템 속도 "50%", Speed 옵션이 "s10"으로 설정된 경우, 로봇의 속도는 "5%"가 됩니다.
- "s" 옵션을 사용하지 않은 경우 시스템 속도와 속도 명령어(motion.speed) 만 적용됩니다.
- "s" 옵션과 속도 명령어를 함께 사용하는 경우, 속도 명령어가 반영되지 않습니다. 자세한 사항은 표 2-54를 참고하시기 바랍니다.

| 항목 | 속도 옵션 | | 설명 |
|---|---|---|---|
| | (A) 미사용 | (B) 사용 | |
| ① 시스템 속도 | 50 % | 50 % | "③ 속도 옵션"이 |

| | | | |
|---|---|---|---|
| ② **속도 명령어** | 50 % | 50 % | 설정될 경우, "② 속도 명령어"가 무시됩니다. |
| ③ **속도 옵션** | - | 10 % | |
| ④ **최종 적용 속도** | 25 % | 5 % | |

**표 2-54 Speed 옵션 사용에 따른 최종 속도(예시)**

* 최종 적용 속도 계산법

(A) : ① 시스템 속도 × ② 속도 명령어 = ④ 최종 적용 속도

(B) : ① 시스템 속도 × ③ 속도 옵션 = ④ 최종 적용 속도

* 속도 명령어 (motion.speed)는 모든 이동 명령어(move)에 반영됩니다.

* 속도 옵션 ("s")는 선언된 move 명령어에만 반영됩니다.

**시스템 속도는 펜던트 우측 하단에 표시되며, 자동/수동모드 속도로 구분됩니다. 자동모드 속도의 경우 스크립트 동작 시 적용되며, 수동모드 속도의 경우 사용자 티칭 화면(조그 화면)에서 적용됩니다.**

■ **예 제**

**[예제1] tool.rotz 회전 이동**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=joint(0,0,90,0,45,45) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | rotz(30) | -- Base 좌표계 Z축 기준 회전이동 |
| 5 | | ※ 최종 위치 : |
| 6 | | pose(651.23, 48, 664.47, 153.66, 23.93, |
| 7 | | 129.32) |
| 8 | | |
| 9 | movep(p1) | -- PTP 이동 |
| 10 | tool.rotz(30) | -- Tool 좌표계 Z축 기준 회전이동 |
| 11 | | ※ 최종 위치 : |
| 12 | | pose(651.23, 48, 664.47, 147.45, 21.69, |
| 13 | | 131.46) |

■ **에러 상황**

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[900000.00] is range over (-100000.00 ~ 100000.00) |
| | 경로 생성 실패 | 경로 생성 실패 |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.36. delay : 대기시간

| | |
|---|---|
| ■ 기 능 | **로봇이 입력된 시간만큼 대기합니다.** |
| ■ 형 태 | delay(Time) |

■ 인 자

| 인자 | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|
| | | | 최소 | 최대 | |
| Time | int | - | 0 | 600000 | ms |

* 단위 : ms (milliesecond)

■ **세부사항**

➢ **Time (대기 시간)**

- 대기 시간은 millisecond 단위이므로 1000 입력 시 1초 시간만큼 대기 합니다.

■ 예 제

**[예제1] delay 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | dio.set(1) | -- 디지털 출력 1번 신호 ON |
| 2 | delay(1000) | -- 1000ms 대기 (=1sec) |
| 3 | dio.reset(1) | -- 디지털 출력 1번 신호 OFF |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[600001] is range over (0 ~ 600000) |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.37. workframe : 워크 좌표계 번호 설정

| | |
|---|---|
| ■ 기 능 | **워크(Work) 좌표계 번호를 설정합니다.** |
| ■ 형 태 | workframe(Num) |

■ 인 자

| 인자 | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|
| | | | 최소 | 최대 | |
| Num | int | - | 0 | 9 | - |

■ **세부사항**

➢ **워크 좌표계 정의**

- 워크 좌표계(work frame)에서 설정한 좌표계 번호를 설정하는 기능입니다.
- 워크 좌표계 0번은 로봇 BASE 좌표계 기준으로 설정되었기 때문에 해당 파라미터는 수정이 불가능합니다.
- 이동 명령어의 워크 좌표계 옵션이 사용되는 경우, 이동 명령어의 사용자 좌표계를 우선합니다.
  ※ 워크 좌표계 적용 우선순위 : move(워크 번호) > workframe(워크 번호)
- 좌표 변환 시 위치 및 방향은 설정한 워크 좌표계 파라미터 값을 참조하여 설정 됩니다.

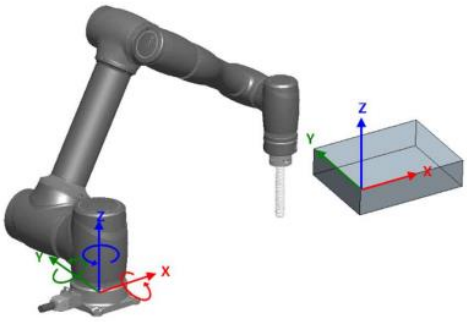
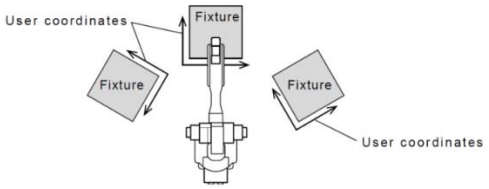



**그림 2-58 워크 좌표계**

■ 설명

■ **워크 좌표계 사용**

- 워크 좌표계는 로봇이 다른 작업 공간에서 동일한 작업을 수행할 때 프로그램에 대한 수정 없이 명령지점에 도달할 수 있도록 사용할 수 있습니다.




위 그림과 같이 작업대의 방향성이 로봇 좌표계와 일치하지 않는 경우 사용자 좌표계를 생성하면 티칭 조작 시 또는 연산식에 의한 위치 생성 프로그래밍에 이점이 있다.

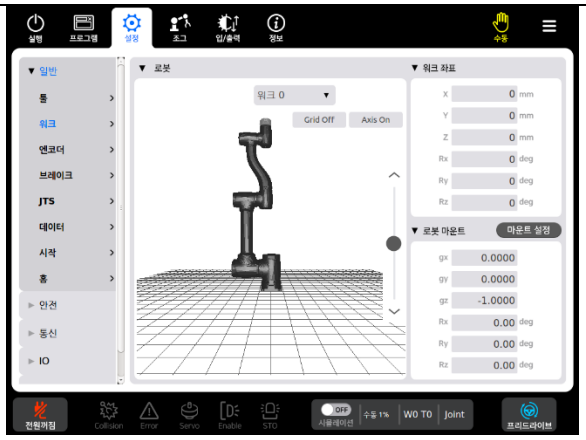



**그림 2-59 워크 좌표계**

- 워크 좌표는 그림과 같이 BASE 좌표계를 기준으로 워크 좌표계의 좌표 값을 설정합니다.
- 로봇 마운트는 설치자세 (천장/벽면/바닥부착) 및 설치 기울기를 입력합니다.
  ※ 로봇 바닥면 부착 시 로봇 마운트 설정 값은 초기 값으로 사용하시기 바랍니다.
- 자세한 워크 좌표계 설정 방법은 사용자 매뉴얼을 참고 해주세요.

■ **예 제**

**[예제1] workframe 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | workframe(1) | -- 워크 좌표계 1 변경 |
| 2 | | |
| 3 | transz(100) | -- Z축 100mm 이동 |
| 4 | transx(-300) | -- X축 -300mm 이동 |
| 5 | | |
| 6 | workframe(0) | -- workframe 0 변경 |

■ **에러 상황**

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[10] is range over (-1 ~ 9) |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

## 2.3.38. toolframe : 툴 좌표계 번호 설정

| | |
|---|---|
| ■ 기 능 | **로봇에 부착된 사용 툴(Tool) 좌표계 번호를 설정합니다.** |
| ■ 형 태 | toolframe(Num) |
| ■ 인 자 | |

| 인자 | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|
| | | | 최소 | 최대 | |
| Num | int | - | 0 | 9 | - |

■ **세부사항**

➢ **툴 좌표계 정의**

- 툴 좌표계(tool frame)에서 설정한 좌표계 번호를 설정하는 기능입니다.
- 툴 좌표계 0번은 로봇의 종단 좌표계로 기준으로 툴이 달려 있지 않은 상태로 설정되었기 때문에 해당 파라미터는 수정이 불가능합니다.
- 이동 명령어의 툴 설정을 사용하는 경우, 이동 명령어의 설정 값이 우선됩니다.
  ※ 툴 좌표계 적용 우선순위 : move(툴 번호) > toolframe(툴 번호)
- 좌표 변환 시 위치 및 방향은 설정한 툴 좌표계 파라미터 값을 참조하여 설정 됩니다.

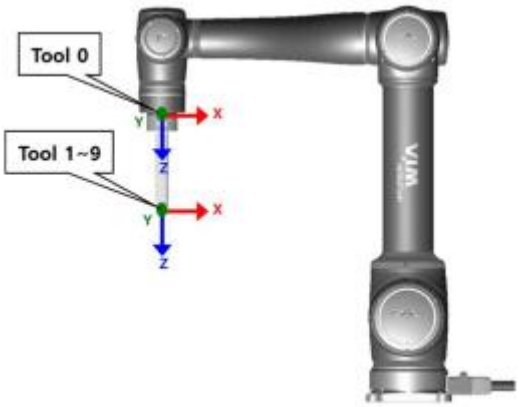



**그림 2-60 툴 좌표계**

■ 툴 좌표계

- 용접, 운반 등 다양한 공정에 적용하기 위한 공구(Tool)를 장착하는 경우, 공구에 형성되는 로봇 종단의 설정된 좌표계입니다.
- 공구에 형성되는 좌표계를 기준으로 로봇 종단의 궤적이 생성됩니다.

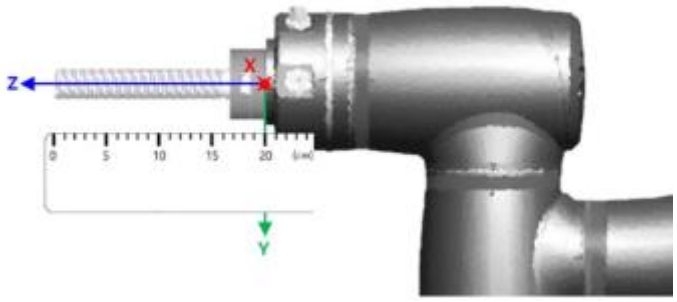



그림 2-61 툴 좌표 정보

■ 설명

- 툴 좌표는 그림 2-61 과 같이 장착된 툴의 끝단에서 로봇 종단까지의 좌표 값을 설정합니다.
- 툴 정보는 장착된 툴의 질량, 축별 질량 중심 및 관성 모멘트를 설정합니다.
- 자세한 툴 좌표계 설정 방법은 사용자 매뉴얼을 참고 해주세요.

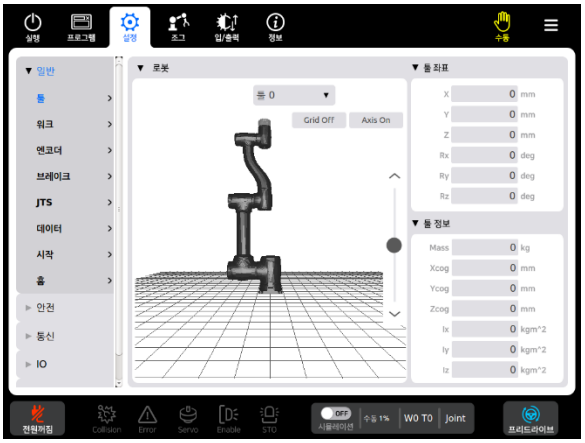



그림 2-62 툴 좌표 화면

■ 예 제

**[예제1] toolframe 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | p1=pose(500,380,500,0,0,0) | -- p1에 위치형 데이터 설정 |
| 2 | | |
| 3 | movep(p1) | -- PTP 이동 |
| 4 | | |
| 5 | toolframe(0) | -- Tool 0번 초기화(표 2-5537 참조) |
| 6 | toolx(100) | -- Tool 좌표계 X축 기준 100mm |
| 7 | | ※ 최종 위치 : |
| 8 | | pose(600,380,500,0,0,0) |
| 9 | | |
| 10 | movep(p1) | -- PTP 이동 |
| 11 | | |
| 12 | toolframe(1) | -- Tool 1번 설정(표 2-5537 참조) |
| 13 | toolx(100) | ※ 최종 위치 : |
| 14 | | pose(620,410,540,0,0,0) |

| | Tool 0 | Tool 1 |
|---|---|---|
| X | 0 | 20 |
| Y | 0 | 30 |
| Z | 0 | 40 |
| Rx | 0 | 0 |
| Ry | 0 | 0 |
| Rz | 0 | 0 |

**표 2-55 Tool 좌표계 설정**

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[10] is range over (-1 ~ 9) |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| motion.speed | movep | trans | rot | delay |
| motion.accel | movel | transx | rotx | workframe |
| motion.decel | movec | transy | roty | toolframe |
| | moves | transz | rotz | |
| joint | | | | |
| pose | rmovep | tool.trans | rotj1 | |
| j2p | rmovel | tool.transx | rotj2 | |
| inv | | tool.transy | rotj3 | |
| | | tool.transz | rotj4 | |
| | | tool.rot | rotj5 | |
| | | tool.rotx | rotj6 | |
| | | tool.roty | | |
| | | tool.rotz | | |

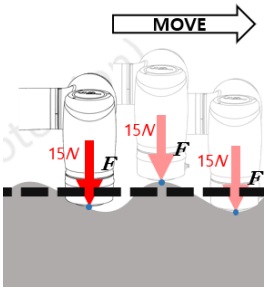
## 2.4. 힘/순응 제어 스크립트

힘 제어(Force control)는 로봇 말단 장치(End-Effector)에 작용하는 힘을 제어하는 방식입니다. 작업 대상물과 접촉하는 공정에서 공간적 한계가 있는 위치제어로는 좋은 결과물을 얻기 어렵기 때문에 사용됩니다. 대표적으로 디버링, 그라인딩, 볼팅, 폴리싱 공정 등에 적용됩니다.

힘 제어의 기능에는 순응 제어(Compliance control) 기능과 충돌 감지(Collision detection) 기능이 포함되어 있습니다.

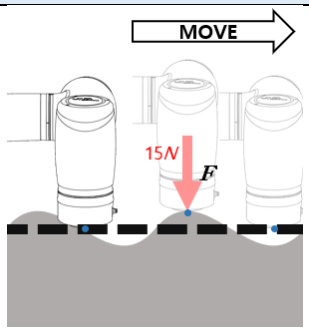
### ■ 힘 제어

- 힘 제어는 표면에 일정한 힘을 가하는 경우 사용되며 사용자가 설정(Desired)한 방향과 힘을 유지하며 이동합니다.

| 힘 제어 예 | 설 명 |
|---|---|
|  | - 힘 제어 사용 시, 사용자가 설정한 15N 힘을 Z 축 방향으로 유지한 상태로 불규칙한 표면을 이동합니다. |

### ■ 순응 제어

- 순응 제어는 울퉁불퉁한 표면을 직선 이동하는 공정에서 로봇과 작업물에 파손 없이 작업해야하는 경우 사용됩니다.

| 순응 제어 예 | 설 명 |
|---|---|
|  | - 순응 제어는 로봇 말단 장치에 외력이 발생한 경우 로봇은 외력에 순응한 상태로 이동하며, 외력이 해제된 경우 원래 위치로 이동합니다. |

### ■ 충돌 감지

- 충돌 감지 기능 사용 시 작업 환경에서 예상치 못한 충돌에서 로봇 및 대상물의 파손을 예방하는데 활용할 수 있습니다.
- 설정된 힘의 크기보다 큰 외부의 힘이 발생하면 동작을 멈춥니다.

**표 2-56 힘/강성 제어 명령어 리스트**

| No. | 분류 | 항목 | 설명 | 참조 |
|---|---|---|---|---|
| 1 | 힘 제어 활성화 유/무 | force.enable() | 힘 제어 활성화 | 2.4.1 |
| 2 | | force.disable() | 힘 제어 비활성화 | 2.4.2 |
| 3 | 힘 제어 설정 | force.set(Fx, Fy, Fz, Tx, Ty, Tz) | 힘 제어할 힘의 방향, 크기 설정 | 2.4.3 |
| 4 | | force.reset() | 힘 제어 방향, 크기 0으로 초기화 | 2.4.4 |
| 5 | | force.bias() | 힘 제어 시작 시점에 측정 힘 0으로 초기화 | 2.4.5 |
| 6 | 순응 제어 설정 | force.set_compliance_x(Value) | X 방향의 순응 제어 값을 설정 | 2.4.6 |
| 7 | | force.set_compliance_y(Value) | Y 방향의 순응 제어 값을 설정 | 2.4.7 |
| 8 | | force.set_compliance_z(Value) | Z 방향의 순응 제어 값을 설정 | 2.4.8 |
| 9 | | force.set_compliance_rx(Value) | Rx 방향의 순응 제어 값을 설정 | 2.4.9 |
| 10 | | force.set_compliance_ry(Value) | Ry 방향의 순응 제어 값을 설정 | 2.4.10 |
| 11 | | force.set_compliance_rz(Value) | Rz 방향의 순응 제어 값을 설정 | 2.4.11 |
| 12 | 충돌감지 설정 | force.collision(Value) | 충돌 감지 설정 | 2.4.12 |
| 13 | | force.set_collision(Value) | 충돌 감지 감도 설정 | 2.4.13 |
| 14 | | force.get_collision(Vlaue) | 현재 충돌 감지 감도 값 읽기 | 2.4.14 |

표 2-56 힘/강성 제어 명령어 리스트는 힘/강성 제어 스크립트에 대해 간략히 설명한 표이며, 보다 안전한 사용을 위해 각 항목의 참조 페이지로 이동 후 자세히 숙지 후 사용하시기 바랍니다.

**주의(Caution)**

- 목표 힘의 크기를 크게 설정하는 경우 강한 힘에 의해 상해를 입을 수 있습니다.
- 힘 제어의 기능을 이용하려면 충돌감지 모드를 끈 후 사용하여야 합니다.
- Tool 을 사용 시 Tool 무게와 TCP 를 정확하게 설정되지 않은 경우, Tool 무게를 외력으로 감지하여 원하지 않은 방향으로 로봇이 이동할 수 있습니다.

**정보(Notice)**

- 현재 로봇 이동 중 힘을 제어하는 기능은 사용할 수 없으며, 추후 개발 예정입니다.

## 2.4.1. force.enable : 힘 제어 활성화

| | |
|---|---|
| ■ 기 능 | **힘 제어 기능을 활성화합니다.** |
| ■ 형 태 | force.enable() |

■ 인 자

| 인자명 | 자료형 | 기본값 | 범위 | 단위 |
|---|---|---|---|---|
| - | - | - | - | - |

■ **세부사항**

➢ **힘 제어 기능을 활성화시키는 명령어로 인자가 필요하지 않습니다.**

➢ **힘 제어 기능**

- 힘 제어 기능을 이용하려면 충돌감지 모드를 비활성 후 사용해야 합니다.
- 설정한 힘과 외력이 힘의 평형을 이룰 때, 로봇 정지 후 다음 단계를 수행합니다.

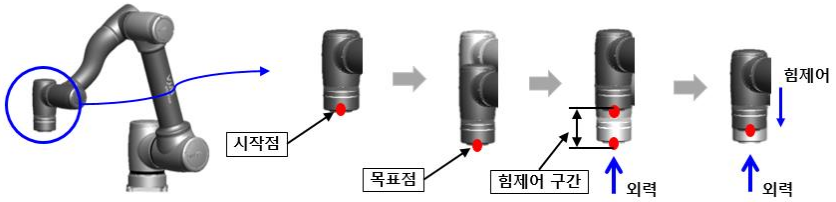



**그림 2-63 힘 제어 작동 원리**

■ 설 명

| | |
|---|---|
| **응용분야** | ● **샌딩(Sanding)/폴리싱(Polishing)**<br>- 일정한 힘을 제공하여 대상물 표면에 파손이나 불균형 없이 연마 작업을 수행할 수 있도록 힘 제어 기능을 지원합니다.<br>● **볼팅(Bolting)**<br>- 설정된 힘을 일정하게 유지하여 각종 스크류, 볼트, 너트 체결 작업을 안정적으로 수행할 수 있습니다. |

■ 예 제

**[예제1] force.enable 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | force.enable() | -- 힘 제어 활성화 |
| 2 | force.bias() | -- FT 센서 측정 값 초기화 |
| 3 | force.reset() | -- 설정 힘의 방향, 크기 초기화 |
| 4 | force.collision(0) | -- 충돌감지 기능 비활성화 |
| 5 | force.set(0,0,30,0,0,0) | -- 힘의 크기 및 방향 설정 |
| 6 | force.disable() | -- 힘 제어 비활성화 |
| 7 | force.collision(1) | -- 충돌감지 기능 활성화 |

■ **에러 상황**

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 명령어 괄호 | ')' expected near ']' |

■ **관련 함수**

| | | | | |
|---|---|---|---|---|
| force.enable | force.set_compliance_x | force.collison | | |
| force.disable | force.set_compliance_x | force.set_collision | | |
| force.set | force.set_compliance_x | force.get_collision | | |
| force.reset | force.set_compliance_x | | | |
| force.bias | force.set_compliance_x | | | |
| | force.set_compliance_x | | | |
| | | | | |
| | | | | |

## 2.4.2. force.disable : 힘 제어 비활성화

| | |
|---|---|
| ■ 기 능 | **힘 제어 기능을 비활성화합니다.** |
| ■ 형 태 | force.disable() |

■ 인 자

| 인자명 | 자료형 | 기본값 | 범위 | 단위 |
|---|---|---|---|---|
| - | - | - | - | - |

■ **세부사항**

➢ **힘 제어 기능을 비활성화시키는 명령어로 인자가 필요하지 않습니다.**

➢ **힘 제어 기능**

- 힘 제어 기능 사용 완료 후, 반드시 비활성화를 하시기 바랍니다.

■ 예 제

**[예제1] force.disable 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | force.enable() | -- 힘 제어 활성화 |
| 2 | force.bias() | -- FT 센서 측정 값 초기화 |
| 3 | force.reset() | -- 설정 힘의 방향, 크기 초기화 |
| 4 | force.collision(0) | -- 충돌감지 기능 비활성화 |
| 5 | force.set(0,0,30,0,0,0) | -- 힘의 크기 및 방향 설정 |
| 6 | force.disable() | -- 힘 제어 비활성화 |
| 7 | force.collision(1) | -- 충돌감지 기능 활성화 |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 명령어 괄호 | ')' expected near ']' |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| force.enable | force.set_compliance_x | force.collison | | |
| force.disable | force.set_compliance_x | force.set_collision | | |
| force.set | force.set_compliance_x | force.get_collision | | |
| force.reset | force.set_compliance_x | | | |
| force.bias | force.set_compliance_x | | | |
| | force.set_compliance_x | | | |
| | | | | |
| | | | | |

## 2.4.3. force.set : 힘 크기 및 방향 설정

| | |
|---|---|
| ■ 기 능 | **힘을 제어할 크기와 방향을 툴(Tool) 좌표계 기준에서 설정합니다.** |
| ■ 형 태 | force.set(Fx, Fy, Fz, Tx, Ty, Tz) |

■ 인 자

| 인자 | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|
| | | | 최소 | 최대 | |
| Fx | double | 0 | -350 | 350 | N |
| Fy | double | 0 | -350 | 350 | N |
| Fz | double | 0 | -350 | 350 | N |
| Tx | double | 0 | -350 | 350 | Nm |
| Ty | double | 0 | -350 | 350 | Nm |
| Tz | double | 0 | -350 | 350 | Nm |

* 단위 : N (Newton) / Nm (Newton meter)

■ **세부사항**

➢ **Fx, Fy, Fz, Tx, Ty, Tz (힘 제어 크기 및 방향 설정)**

- 툴 좌표계의 위치에 따라 목표 힘과 목표 회전력의 크기를 설정합니다.

■ 예 제

**[예제1] force.set 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | force.enable() | -- 힘 제어 활성화 |
| 2 | force.bias() | -- FT 센서 측정 값 초기화 |
| 3 | force.reset() | -- 설정 힘의 방향, 크기 초기화 |
| 4 | force.collision(0) | -- 충돌감지 기능 비활성화 |
| 5 | force.set(0,0,30,0,0,0) | -- 힘의 크기 및 방향 설정 |
| 6 | force.disable() | -- 힘 제어 비활성화 |
| 7 | force.collision(1) | -- 충돌감지 기능 활성화 |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number[6] of argument |
| | 인자 타입 | Invalid argument[2] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[1003] is range over(-1000~1000) |

| | | | | | |
|---|---|---|---|---|---|
| ■ 관련 함수 | force.enable | force.set_compliance_x | force.collison | | |
| | force.disable | force.set_compliance_x | force.set_collision | | |
| | force.set | force.set_compliance_x | force.get_collision | | |
| | force.reset | force.set_compliance_x | | | |
| | force.bias | force.set_compliance_x | | | |
| | | force.set_compliance_x | | | |
| | | | | | |
| | | | | | |

## 경고(Warning)

- 힘 제어 명령어 사용 시 설정 값(크기 및 방향)이 적절하지 않은 경우 제품 손상 및 작업자가 상해를 입을 수 있습니다. 반드시 매뉴얼 숙지 후 사용하시기 바랍니다.

## 2.4.4. force.reset : 힘 제어 설정 초기화

| | |
|---|---|
| ■ 기 능 | **힘 제어 설정 값(크기 및 방향)을 초기화합니다.** |
| ■ 형 태 | force.reset() |
| ■ 인 자 | 인자명 / 자료형 / 기본값 / 범위 / 단위<br>- / - / - / - / -<br><br>■ **세부사항**<br>➢ **힘 제어의 설정 값을 0으로 초기화하는 명령어로 인자가 필요하지 않습니다.** |
| ■ 예 제 | **[예제1] force.reset 사용** |

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | force.enable() | -- 힘 제어 활성화 |
| 2 | force.bias() | -- FT 센서 측정 값 초기화 |
| 3 | force.reset() | -- 설정 힘의 방향, 크기 초기화 |
| 4 | force.collision(0) | -- 충돌감지 기능 비활성화 |
| 5 | force.set(0,0,30,0,0,0) | -- 힘의 크기 및 방향 설정 |
| 6 | force.disable() | -- 힘 제어 비활성화 |
| 7 | force.collision(1) | -- 충돌감지 기능 활성화 |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 명령어 괄호 | ')' expected near ']' |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| force.enable | force.set_compliance_x | force.collison | | |
| force.disable | force.set_compliance_x | force.set_collision | | |
| force.set | force.set_compliance_x | force.get_collision | | |
| force.reset | force.set_compliance_x | | | |
| force.bias | force.set_compliance_x | | | |
| | force.set_compliance_x | | | |
| | | | | |
| | | | | |

## 2.4.5. force.bias : 외력 참조 모드

| | |
|---|---|
| ■ 기 능 | **FT(Force Torque) 센서의 측정 값을 0으로 초기화합니다.** |
| ■ 형 태 | force.bias() |

■ 인 자

| 인자명 | 자료형 | 기본값 | 범위 | 단위 |
|---|---|---|---|---|
| - | - | - | - | - |

■ **세부사항**

➢ **FT 센서의 측정 값을 0으로 초기화하는 명령어로 인자가 필요하지 않습니다.**

■ 예 제

**[예제1] force.bias 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | force.enable() | -- 힘 제어 활성화 |
| 2 | force.bias() | -- FT 센서 측정 값 초기화 |
| 3 | force.reset() | -- 설정 힘의 방향, 크기 초기화 |
| 4 | force.collision(0) | -- 충돌감지 기능 비활성화 |
| 5 | force.set(0,0,30,0,0,0) | -- 힘의 크기 및 방향 설정 |
| 6 | force.disable() | -- 힘 제어 비활성화 |
| 7 | force.collision(1) | -- 충돌감지 기능 활성화 |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number of arguments |
| | 명령어 괄호 | ')' expected near ']' |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| force.enable | force.set_compliance_x | force.collison | | |
| force.disable | force.set_compliance_x | force.set_collision | | |
| force.set | force.set_compliance_x | force.get_collision | | |
| force.reset | force.set_compliance_x | | | |
| force.bias | force.set_compliance_x | | | |
| | force.set_compliance_x | | | |
| | | | | |
| | | | | |

## 2.4.6. force.set_compliance_x : X 방향 순응 제어 설정

| | |
|---|---|
| ■ 기 능 | **X 방향의 순응 제어 값을 설정합니다.** |
| ■ 형 태 | force.set_compliance_x(Value) |

■ 인 자

| 인자 | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|
| | | | 최소 | 최대 | |
| Value | double | - | 0.0001 | 1.0 | - |

■ **세부사항**

➢ **Value (순응 제어 크기)**

- 툴 좌표계의 위치에 따라 X 방향에 대한 순응 제어를 설정합니다.
- 값이 클수록, 외부 힘에 둔감하게 반응하며, 값이 작을수록 외부 힘에 민감하게 작용 합니다..

■ 예 제

**[예제1] force.set_compliance_x 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | force.bias() | -- FT 센서 측정 값 초기화 |
| 2 | force.enable() | -- 힘 제어 활성화 |
| 3 | force.set_compliance_x(0.01) | -- X 방향 순응 제어 값 설정 |
| 4 | force.set_compliance_y(0.01) | -- Y 방향 순응 제어 값 설정 |
| 5 | force.set_compliance_rx(0.01) | -- Rx 방향 순응 제어 값 설정 |
| 6 | force.set_compliance_ry(0.01) | -- Ry 방향 순응 제어 값 설정 |
| 7. | force.set(0, 0, 50, 0, 0, 0) | -- 힘의 크기 및 방향 설정 |
| 8 | delay(1000) | -- 1초간 대기 |
| 9 | force.disable() | -- 힘 제어 비활성화 |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number[1] of argument |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value is range over(0.0001~1.0) |

| | | | | | |
|---|---|---|---|---|---|
| ■ 관련 함수 | force.enable | force.set_compliance_x | force.collison | | |
| | force.disable | force.set_compliance_x | force.set_collision | | |
| | force.set | force.set_compliance_x | force.get_collision | | |
| | force.reset | force.set_compliance_x | | | |
| | force.bias | force.set_compliance_x | | | |
| | | force.set_compliance_x | | | |
| | | | | | |
| | | | | | |

## 2.4.7. force. set_compliance_y : Y 방향 순응 제어 설정

| | |
|---|---|
| ■ 기 능 | **Y 방향의 순응 제어 값을 설정합니다.** |
| ■ 형 태 | force.set_compliance_y(Value) |

■ 인 자

| 인자 | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|
| | | | 최소 | 최대 | |
| Value | double | - | 0.0001 | 1.0 | - |

■ **세부사항**

➢ **Value (순응 제어 크기)**

- 툴 좌표계의 위치에 따라 Y 방향에 대한 순응 제어를 설정합니다.
- 값이 클수록, 외부 힘에 둔감하게 반응하며, 값이 작을수록 외부 힘에 민감하게 작용 합니다..

■ 예 제

**[예제1] force.set_compliance_y 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | force.bias() | -- FT 센서 측정 값 초기화 |
| 2 | force.enable() | -- 힘 제어 활성화 |
| 3 | force.set_compliance_x(0.01) | -- X 방향 순응 제어 값 설정 |
| 4 | force.set_compliance_y(0.01) | -- Y 방향 순응 제어 값 설정 |
| 5 | force.set_compliance_rx(0.01) | -- Rx 방향 순응 제어 값 설정 |
| 6 | force.set_compliance_ry(0.01) | -- Ry 방향 순응 제어 값 설정 |
| 7. | force.set(0, 0, 50, 0, 0, 0) | -- 힘의 크기 및 방향 설정 |
| 8 | delay(1000) | -- 1초간 대기 |
| 9 | force.disable() | -- 힘 제어 비활성화 |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number[1] of argument |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value is range over(0.0001~1.0) |

| | | | | | |
|---|---|---|---|---|---|
| ■ 관련 함수 | force.enable | force.set_compliance_x | force.collison | | |
| | force.disable | force.set_compliance_x | force.set_collision | | |
| | force.set | force.set_compliance_x | force.get_collision | | |
| | force.reset | force.set_compliance_x | | | |
| | force.bias | force.set_compliance_x | | | |
| | | force.set_compliance_x | | | |
| | | | | | |
| | | | | | |

## 2.4.8. force. set_compliance_z : Z 방향 순응 제어 설정

| | |
|---|---|
| ■ 기 능 | **Z 방향의 순응 제어 값을 설정합니다.** |
| ■ 형 태 | force.set_compliance_z(Value) |

■ 인 자

| 인자 | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|
| | | | 최소 | 최대 | |
| Value | double | - | 0.0001 | 1.0 | - |

■ **세부사항**

➢ **Value (순응 제어 크기)**

- 툴 좌표계의 위치에 따라 Z 방향에 대한 순응 제어를 설정합니다.
- 값이 클수록, 외부 힘에 둔감하게 반응하며, 값이 작을수록 외부 힘에 민감하게 작용 합니다..

■ 예 제

**[예제1] force.set_compliance_z 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | force.bias() | -- FT 센서 측정 값 초기화 |
| 2 | force.enable() | -- 힘 제어 활성화 |
| 3 | force.set_compliance_x(0.01) | -- X 방향 순응 제어 값 설정 |
| 4 | force.set_compliance_z(0.01) | -- Z 방향 순응 제어 값 설정 |
| 5 | force.set_compliance_rx(0.01) | -- Rx 방향 순응 제어 값 설정 |
| 6 | force.set_compliance_rz(0.01) | -- Rz 방향 순응 제어 값 설정 |
| 7. | force.set(0, 50, 0, 0, 0, 0) | -- 힘의 크기 및 방향 설정 |
| 8 | delay(1000) | -- 1초간 대기 |
| 9 | force.disable() | -- 힘 제어 비활성화 |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number[1] of argument |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value is range over(0.0001~1.0) |

| | | | | | |
|---|---|---|---|---|---|
| ■ 관련 함수 | force.enable | force.set_compliance_x | force.collison | | |
| | force.disable | force.set_compliance_x | force.set_collision | | |
| | force.set | force.set_compliance_x | force.get_collision | | |
| | force.reset | force.set_compliance_x | | | |
| | force.bias | force.set_compliance_x | | | |
| | | force.set_compliance_x | | | |
| | | | | | |
| | | | | | |

## 2.4.9. force. set_compliance_rx : Rx 방향 순응 제어 설정

| | |
|---|---|
| ■ 기 능 | **Rx 방향의 순응 제어 값을 설정합니다.** |
| ■ 형 태 | force.set_compliance_rx(Value) |

■ 인 자

| 인자 | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|
| | | | 최소 | 최대 | |
| Value | double | - | 0.0001 | 1.0 | - |

■ **세부사항**

➢ **Value (순응 제어 크기)**

- 툴 좌표계의 위치에 따라 Rx 방향에 대한 순응 제어를 설정합니다.
- 값이 클수록, 외부 힘에 둔감하게 반응하며, 값이 작을수록 외부 힘에 민감하게 작용 합니다..

■ 예 제

**[예제1] force.set_compliance_rx 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | force.bias() | -- FT 센서 측정 값 초기화 |
| 2 | force.enable() | -- 힘 제어 활성화 |
| 3 | force.set_compliance_x(0.01) | -- X 방향 순응 제어 값 설정 |
| 4 | force.set_compliance_y(0.01) | -- Y 방향 순응 제어 값 설정 |
| 5 | force.set_compliance_rx(0.01) | -- Rx 방향 순응 제어 값 설정 |
| 6 | force.set_compliance_ry(0.01) | -- Ry 방향 순응 제어 값 설정 |
| 7. | force.set(0, 0, 50, 0, 0, 0) | -- 힘의 크기 및 방향 설정 |
| 8 | delay(1000) | -- 1초간 대기 |
| 9 | force.disable() | -- 힘 제어 비활성화 |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number[1] of argument |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value is range over(0.0001~1.0) |

| | | | | | |
|---|---|---|---|---|---|
| ■ 관련 함수 | force.enable | force.set_compliance_x | force.collison | | |
| | force.disable | force.set_compliance_x | force.set_collision | | |
| | force.set | force.set_compliance_x | force.get_collision | | |
| | force.reset | force.set_compliance_x | | | |
| | force.bias | force.set_compliance_x | | | |
| | | force.set_compliance_x | | | |
| | | | | | |
| | | | | | |

## 2.4.10. force. set_compliance_ry : Ry 방향 순응 제어 설정

| | |
|---|---|
| ■ 기 능 | **Ry 방향의 순응 제어 값을 설정합니다.** |
| ■ 형 태 | force.set_compliance_ry(Value) |

■ 인 자

| 인자 | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|
| | | | 최소 | 최대 | |
| Value | double | - | 0.0001 | 1.0 | - |

■ **세부사항**

➢ **Value (순응 제어 크기)**

- 툴 좌표계의 위치에 따라 Ry 방향에 대한 순응 제어를 설정합니다.
- 값이 클수록, 외부 힘에 둔감하게 반응하며, 값이 작을수록 외부 힘에 민감하게 작용 합니다..

■ 예 제

**[예제1] force.set_compliance_ry 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | force.bias() | -- FT 센서 측정 값 초기화 |
| 2 | force.enable() | -- 힘 제어 활성화 |
| 3 | force.set_compliance_x(0.01) | -- X 방향 순응 제어 값 설정 |
| 4 | force.set_compliance_y(0.01) | -- Z 방향 순응 제어 값 설정 |
| 5 | force.set_compliance_rx(0.01) | -- Rx 방향 순응 제어 값 설정 |
| 6 | force.set_compliance_ry(0.01) | -- Rz 방향 순응 제어 값 설정 |
| 7. | force.set(0, 0, 50, 0, 0, 0) | -- 힘의 크기 및 방향 설정 |
| 8 | delay(1000) | -- 1초간 대기 |
| 9 | force.disable() | -- 힘 제어 비활성화 |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number[1] of argument |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value is range over(0.0001~1.0) |

| | | | | | |
|---|---|---|---|---|---|
| ■ 관련 함수 | force.enable | force.set_compliance_x | force.collison | | |
| | force.disable | force.set_compliance_x | force.set_collision | | |
| | force.set | force.set_compliance_x | force.get_collision | | |
| | force.reset | force.set_compliance_x | | | |
| | force.bias | force.set_compliance_x | | | |
| | | force.set_compliance_x | | | |
| | | | | | |
| | | | | | |

## 2.4.11. force.set_compliance_rz : Rz 방향 순응 제어 설정

| | |
|---|---|
| ■ 기 능 | **Rz 방향의 순응 제어 값을 설정합니다.** |
| ■ 형 태 | force.set_compliance_rz(Value) |

■ 인 자

| 인자 | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|
| | | | 최소 | 최대 | |
| Value | double | - | 0.0001 | 1.0 | - |

■ **세부사항**

➢ **Value (순응 제어 크기)**

- 툴 좌표계의 위치에 따라 Rz 방향에 대한 순응 제어를 설정합니다.
- 값이 클수록, 외부 힘에 둔감하게 반응하며, 값이 작을수록 외부 힘에 민감하게 작용 합니다..

■ 예 제

**[예제1] force.set_compliance_rz 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | force.bias() | -- 힘 제어 활성화 |
| 2 | force.enable() | -- FT 센서 측정 값 초기화 |
| 3 | force.set_compliance_x(0.01) | -- X 방향 순응 제어 값 설정 |
| 4 | force.set_compliance_z(0.01) | -- 설정 힘의 방향, 크기 초기화 |
| 5 | force.set_compliance_rx(0.01) | -- 충돌감지 기능 비활성화 |
| 6 | force.set_compliance_rz(0.01) | -- 힘의 크기 및 방향 설정 |
| 7. | force.set(0, 50, 0, 0, 0, 0) | -- 힘 제어 비활성화 |
| 8 | delay(1000) | -- 충돌감지 기능 활성화 |
| 9 | force.disable() | |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number[1] of argument |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value is range over(0.0001~1.0) |

| | | | | | |
|---|---|---|---|---|---|
| ■ 관련 함수 | force.enable | force.set_compliance_x | force.collison | | |
| | force.disable | force.set_compliance_x | force.set_collision | | |
| | force.set | force.set_compliance_x | force.get_collision | | |
| | force.reset | force.set_compliance_x | | | |
| | force.bias | force.set_compliance_x | | | |
| | | force.set_compliance_x | | | |
| | | | | | |
| | | | | | |

## 2.4.12. force.collision : 충돌 감지 설정

| | |
|---|---|
| ■ 기 능 | 충돌 감지 기능을 활성화 및 비활성화 합니다. |
| ■ 형 태 | force.collision(Value) |

■ 인 자

| 인자 | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|
| | | | 최소 | 최대 | |
| Value | int | - | 0 | 1 | - |

■ 세부사항

➢ **Value**

- 0으로 설정 시, 충돌 감지 기능을 비활성화 합니다.
- 1 로 설정 시, 충돌 감지 기능을 활성화합니다.
- 힘 제어 기능을 사용하는 경우, 충돌 감지 기능을 비활성화해야 합니다.

■ 예 제

**[예제1] 힘 제어 스크립트 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | force.enable() | -- 힘 제어 활성화 |
| 2 | force.bias() | -- FT 센서 측정 값 초기화 |
| 3 | force.reset() | -- 설정 힘의 방향, 크기 초기화 |
| 4 | force.collision(0) | -- 충돌감지 기능 비활성화 |
| 5 | force.set(0,0,30,0,0,0) | -- 힘의 크기 및 방향 설정 |
| 6 | force.disable() | -- 힘 제어 비활성화 |
| 7 | force.collision(1) | -- 충돌감지 기능 활성화 |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number[1] of argument |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[0] is range over(0~1) |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| force.enable | force.set_compliance_x | force.collison | | |
| force.disable | force.set_compliance_x | force.set_collision | | |
| force.set | force.set_compliance_x | force.get_collision | | |
| force.reset | force.set_compliance_x | | | |
| force.bias | force.set_compliance_x | | | |
| | force.set_compliance_x | | | |
| | | | | |
| | | | | |

## 2.4.13. force.set_collision : 충돌 감지 감도 설정

| | |
|---|---|
| ■ 기 능 | 충돌감지의 감도를 설정합니다. |
| ■ 형 태 | force.set_collision(Value) |

■ 인 자

| 인자 | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|
| | | | 최소 | 최대 | |
| Value | int | - | 1 | 100 | - |

■ **세부사항**

➢ **Value**

- Value에 값을 지정하여 충돌 감지의 감도를 설정할 수 있습니다.
- 설정값이 작을수록 둔감하며, 커질수록 적은 힘에도 민감하게 반응합니다.
- 해당 명령어는 메인 프로그램에서만 사용가능하며 스레트 내에서는 사용 불가합니다.

■ 예 제

**[예제1] force.set_collision 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | motion.speed=100 | -- 이동속도 설정 |
| 2 | | |
| 3 | p1=joint(0,0,90,0,90,0) | -- 이동 포인트 설정 |
| 4 | p2=joint(100,30,90,120,0) | -- 이동 포인트 설정 |
| 5 | | |
| 6 | force.set_collision(70) | -- 충돌 감지 감도 70설정 |
| 7 | movep(p1) | -- p1 포인트로 이동 |
| 8 | movep(p2) | -- p2 포인트로 이동 |
| 9 | | |
| 10 | val=force.get_collision() | -- 현재 충돌 감지 감도값 읽기 |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number[1] of argument |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[0] is range over(1~100) |

| | | | | | |
|---|---|---|---|---|---|
| ■ 관련 함수 | force.enable | force.set_compliance_x | force.collison | | |
| | force.disable | force.set_compliance_x | force.set_collision | | |
| | force.set | force.set_compliance_x | force.get_collision | | |
| | force.reset | force.set_compliance_x | | | |
| | force.bias | force.set_compliance_x | | | |
| | | force.set_compliance_x | | | |
| | | | | | |
| | | | | | |

## 2.4.14. force.get_collision : 현재 충돌 감지 감도 값 읽기

| | |
|---|---|
| ■ 기 능 | 현재 설정된 충돌 감지의 감도값을 읽어옵니다. |
| ■ 형 태 | force.get_collision() |
| ■ 인 자 | |

■ 반환값

| 인자 | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|
| | | | 최소 | 최대 | |
| Return | int | - | 1 | 100 | - |

➢ **Return (반환 값)**

- 현재 설정된 출돌 감지의 감도값을 반환합니다.

■ 예 제

**[예제1] 힘 제어 스크립트 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | motion.speed=100 | -- 이동속도 설정 |
| 2 | | |
| 3 | p1=joint(0,0,90,0,90,0) | -- 이동 포인트 설정 |
| 4 | p2=joint(100,30,90,120,0) | -- 이동 포인트 설정 |
| 5 | | |
| 6 | force.set_collision(70) | -- 충돌 감지 감도 70설정 |
| 7 | movep(p1) | -- p1 포인트로 이동 |
| 8 | movep(p2) | -- p2 포인트로 이동 |
| 9 | | |
| 10 | val=force.get_collision() | -- 현재 충돌 감지 감도값 읽기 |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number[0] of argument |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| force.enable | force.set_compliance_x | force.collison | | |
| force.disable | force.set_compliance_x | force.set_collision | | |
| force.set | force.set_compliance_x | force.get_collision | | |
| force.reset | force.set_compliance_x | | | |
| force.bias | force.set_compliance_x | | | |
| | force.set_compliance_x | | | |
| | | | | |
| | | | | |

## 2.5. 입/출력 스크립트

입/출력(In/Out) 스크립트는 로봇과 외부 장치의 입/출력 신호를 연결하여 운용 시 사용합니다. 협동로봇은 디지털(Digital), 아날로그(Analog), 툴(Tool) 입/출력 방식을 지원하고 있습니다.

### ■ 디지털 입/출력

- 그림 2-64과 같이 기본 디지털 입/출력 포트는 16점을 지원합니다.
- 추가 확장 필요 시 외부 모듈을 최대 2개 추가 장착할 수 있으며, 모듈당 16점씩 추가됩니다.
- 디지털 입/출력의 모니터링 주기는 10[ms]이며, 허용 전압은 24[V] 입니다.

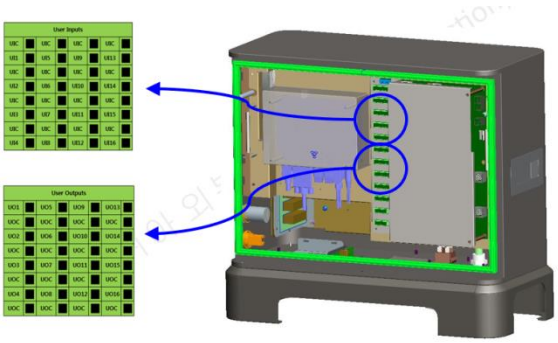



**그림 2-64 디지털 입/출력 위치 및 구성**

### ■ 아날로그 입/출력

- 그림 2-65과 같이 기본 아날로그 입/출력 포트는 2점을 지원합니다.
- 추가 확장 필요 시 외부 모듈을 최대 2개 추가 장착할 수 있으며, 모듈당 4점씩 추가됩니다.
- 아날로그 입/출력의 모니터링 주기는 50[ms]이며, 허용 전압과 전류는 표 2-57과 같습니다.

| | 최소 | 최대 |
|---|---|---|
| **전압 [V]** | 0 | 10 |
| **전류 [mA]** | 4 | 20 |

**표 2-57 허용 전압 및 전류**

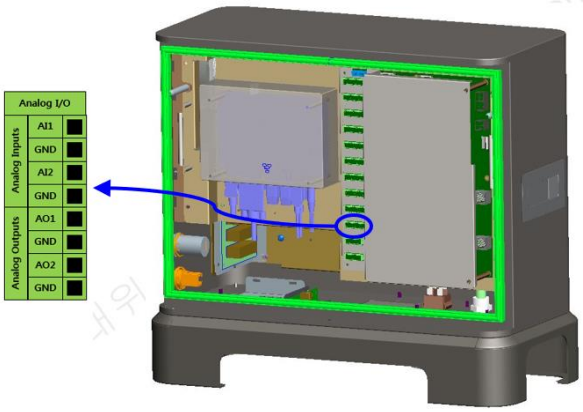


그림 2-65 아날로그 입/출력 위치 및 구성

## ■ 툴 입/출력

- 그림 2-66과 같이 디지털 입/출력 2점, 아날로그 입력 2점을 지원합니다.
- M8 사양의 8핀 커넥터의 사양은 표 2-58과 같습니다.

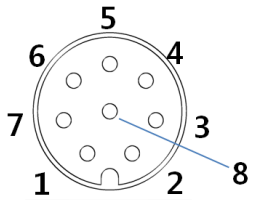
| M8-8pole | 순서 | 신호 | 설명 |
|---|---|---|---|
|  | 1 | GND | Ground |
| | 2 | PWR | Power |
| | 3 | DO1 | Digital Output 1 |
| | 4 | DO2 | Digital Output 2 |
| | 5 | DI1 | Digital Input 1 |
| | 6 | DI2 | Digital Input 2 |
| | 7 | AI2 / RS485+ | AI2 또는 RS485+ |
| | 8 | AI3 / RS485- | AI3 또는 RS485- |

표 2-58 커넥터 사양

- 모니터링 주기는 10[ms]이며, 출력 전원 및 전류에 대한 사양은 표 2-59와 같이 지원합니다.

| 터미널 | 매개변수 | | Spec | 단위 |
|---|---|---|---|---|
| 8pole | 전원 | | 0 / 12 / 24 (GUI에서 선택) | V |
| | 전류 | 단일핀 전원 공급 시 | 1 | A |
| | | 듀얼핀 전원 공급 시 | 1.5 | |

표 2-59 전기적 사양

- 툴 출력의 경우 Sink(NPN) , Source(PNP), POWER모드를 지원하며, 기본 사양은 Source(PNP) 모드 입니다.
- 각 모드에 따른 디지털 입/출력의 전기 사양은 표 2-60과 같습니다.

| 디지털 | 전원 | 매개변수 | Spec | 단위 |
|---|---|---|---|---|
| 출력 | 전압 | Open | 0~24 | V |
| | | Sink(500[mA]) | 0.07 | V |
| | | Source(500[mA]) | 23.93 | V |
| | 전류 | Source/Sink | 0~500 | mA |
| 입력 | 전압 | - | 0 ~ 26V | V |
| | 저항 | - | 2.4 | kΩ |

표 2-60 모드에 따른 허용 전기 사양

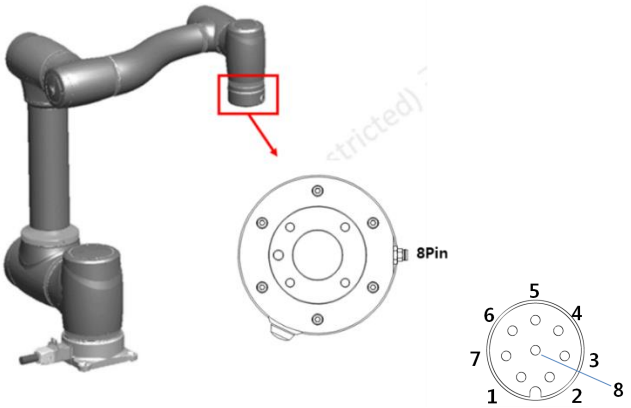



**그림 2-66 툴 입/출력 위치 및 구성**

**표 2-61 입출력(I/O) 스크립트 명령어 리스트**

| No. | 분류 | 항목 | 설명 | 참조 |
|---|---|---|---|---|
| 1 | 디지털 (Digital) | dio.set(Port) | 해당 포트(Port)의 디지털 출력 활성화 | 2.5.1 |
| 2 | | dio.reset(Port) | 해당 포트(Port)의 디지털 출력 비활성화 | 2.5.2 |
| 3 | | dio.set_bits(Port) | 다중 포트 디지털 출력 활성화 | 2.5.3 |
| 4 | | dio.reset_bits(Port) | 다중 포트 디지털 출력 비활성화 | 2.5.4 |
| 5 | | dio.get(Port) | 해당 포트(Port) 디지털 입력 값 가져오기 | 2.5.5 |
| 6 | | dio.in_wait(Port, Value, [Time]) | 입력 포트(Port)가 설정 값(Value)과 동일할 때까지 대기 혹은 지정 시간(Time)까지 대기 | 2.5.6 |
| 7 | | dio.out_wait(Port, Value, [Time]) | 출력 포트(Port)가 설정 값(Value)과 동일할 때까지 대기 혹은 지정 시간(Time)까지 대기 | 2.5.7 |
| 8 | | dio.pulse(Port, Value, [Time]) | 해당 포트(Port)의 출력 값(Value)를 지정 시간(Time)동안 단일 펄스 출력 | 2.5.8 |
| 9 | 아날로그 (Analog) | aio.set(Port, Voltage) | 해당 포트(Port)의 출력 값(Value) 설정 | 2.5.9 |
| 10 | | aio.get(Port) | 해당 포트(Port)의 출력 값(Value) 가져오기 | 2.5.10 |
| 11 | 툴 (Tool) | tool.set(Port) | 해당 포트(Port)의 툴 출력 신호 활성화 | 2.5.11 |
| 12 | | tool.reset(Port) | 해당 포트(Port)의 툴 출력 신호 비활성화 | 2.5.12 |
| 13 | | tool.get(Port) | 해당 포트(Port)의 툴 입력 값 가져오기 | 2.5.13 |
| 14 | | tool.in_wait(Port, Value, [Time]) | 입력 포트(Port)가 설정 값(Value)과 동일할 때 까지 대기 혹은 지정 시간(Time)까지 대기 | 2.5.14 |
| 15 | | tool.out_wait(Port, Value, [Time]) | 출력 포트(Port)가 설정 값(Value)과 동일할 때 까지 대기 혹은 지정 시간(Time)까지 대기 | 2.5.15 |
| 16 | | tool.voltage(Value) | 툴 입/출력의 전원[V] 값 설정 | 2.5.16 |
| 17 | | tool.modbus_write_register(Address, Value) | 해당 주소의 Holding Register에 값을 씁니다. | 2.5.17 |
| 18 | | tool.modbus_read_register(Address) | 해당 주소의 Holding Register 값 1개 읽기 | 2.5.18 |

## 2.5.1. dio.set : 디지털 출력 신호 활성화(ON)

| | |
|---|---|
| ■ 기 능 | **해당 포트(Port)의 디지털 출력 신호를 활성화(ON)합니다.** |
| ■ 형 태 | dio.set(Port) |

■ 인 자

| 인자 | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|
| | | | 최소 | 최대 | |
| Port | int | - | 1 | 48 | - |

■ **세부사항**

➢ **Port (제어기의 I/O 접점 번호)**

- 해당 포트의 신호를 활성화 합니다.
- 1~16번 포트는 기본 제공 포트이며, 확장 IO를 연결 할 경우, 17~48번 포트를 사용 할 수 있습니다.
- 확장 IO는 모듈당 16개의 접점으로 구성되어 있습니다.

ⓘ **해당 명령어는 포트 하나에 대하여 설정 가능합니다.**
**다수의 포트에 대한 설정이 필요 시, dio.set_bits(Value)를 사용하십시오.**

■ 예 제

**[예제1] dio.set 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | dio.set(1) | -- 디지털 출력 1번 신호 활성화 |
| 2 | delay(1000) | -- 1000ms 대기 (=1sec) |
| 3 | dio.reset(1) | -- 디지털 출력 1번 신호 비활성화 |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number[1] of argument |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[50] is range over(1~48) |

| | | | | | |
|---|---|---|---|---|---|
| ■ 관련 함수 | dio.set | aio.set | tool.set | tool.modbus_write_regiseter | |
| | dio.reset | aio.get | tool.reset | tool.modbus_read_regitster | |
| | dio.set_bits | | tool.get | | |
| | dio.reset_bits | | tool.in_wait | | |
| | dio.get | | tool.out_wait | | |
| | dio.in_wait | | tool.voltage | | |
| | dio.out_wait | | | | |
| | dio.pulse | | | | |

## 2.5.2. dio.reset : 디지털 출력 신호 비활성화(OFF)

| | |
|---|---|
| ■ 기 능 | **해당 포트(Port)의 디지털 출력 신호를 비활성화(OFF)합니다.** |
| ■ 형 태 | dio.reset(Port) |

■ 인 자

| 인자 | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|
| | | | 최소 | 최대 | |
| Port | int | - | 1 | 48 | - |

■ **세부사항**

➢ **Port (제어기의 I/O 접점 번호)**

- 해당 포트의 출력 신호를 비활성화합니다.
- 1~16번 포트는 기본 제공 포트이며, 확장 IO를 연결 할 경우, 17~48번 포트를 사용 할 수 있습니다.
- 확장 IO는 모듈당 16개의 접점으로 구성되어 있습니다.

ⓘ **해당 명령어는 포트 하나에 대하여 설정 가능합니다.**
**다수의 포트에 대한 설정이 필요 시, dio.set_bits(Value)를 사용하십시오.**

■ 예 제

**[예제1] dio.reset 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | dio.set(1) | -- 디지털 출력 1번 신호 활성화 |
| 2 | delay(1000) | -- 1000ms 대기 (=1sec) |
| 3 | dio.reset(1) | -- 디지털 출력 1번 신호 비활성화 |

■ 에러 상황

| 발생 시점 | 검사 항목 | 메시지 |
|---|---|---|
| 컴파일 오류 | 인자 개수 | Invalid number[1] of argument |
| | 인자 타입 | Invalid argument[1] type of number data |
| | 명령어 괄호 | ')' expected near ']' |
| 실행 오류 | 데이터 유효범위 | Value[50] is range over(1~48) |

■ 관련 함수

| | | | | |
|---|---|---|---|---|
| dio.set | aio.set | tool.set | tool.modbus_write_regiseter | |
| dio.reset | aio.get | tool.reset | tool.modbus_read_regitster | |
| dio.set_bits | | tool.get | | |
| dio.reset_bits | | tool.in_wait | | |
| dio.get | | tool.out_wait | | |
| dio.in_wait | | tool.voltage | | |
| dio.out_wait | | | | |
| dio.pulse | | | | |

## 2.5.3. dio.set_bits : 다중 디지털 출력 신호 활성화(ON)

| | |
|---|---|
| ■ 기 능 | **다중 포트(Port)의 디지털 출력 신호를 활성화 합니다.** |
| ■ 형 태 | dio.set_bits(Port) |

■ 인 자

| 인자 | 자료형 | 기본값 | 범위 | | 단위 |
|---|---|---|---|---|---|
| | | | 최소 | 최대 | |
| Port | uint32_t | - | 0x0000 | 0xFFFFFFFFFFFF | - |

■ **세부사항**

➢ **Port (복수 개 I/O를 출력하고자 하는 접점 list)**

- 인자의 경우, 16진수로 표현합니다.
- 1~16번 포트는 기본 제공 포트이며, 확장 IO를 연결 할 경우, 17~48번 포트를 사용 할 수 있습니다.
- 확장 IO는 모듈당 16개의 접점으로 구성되어 있습니다.

➢ **16진수 표현 방법**

- 그림 2-67과 같이 16진수의 한자리는 4bit에 해당 됩니다.
- 16진수로 설정한 해당 bit의 값은 1로 설정되며, 해당 값은 디지털 출력 신호를 설정합니다.

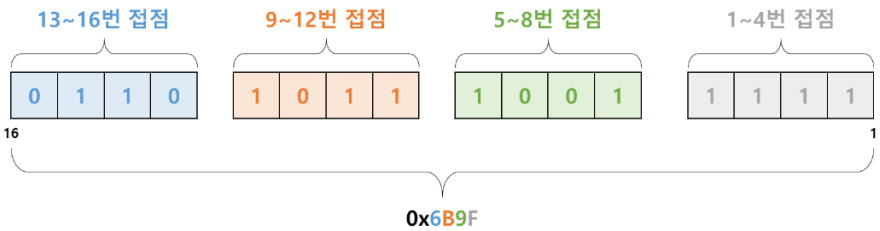



**그림 2-67 16진수와 bits 간의 관계**

■ 예 제

**[예제1] dio.set_bits 사용**

| 번호 | 스크립트 | 주석 |
|---|---|---|
| 1 | dio.set_bits(0x0031) | -- 디지털 출력 1번, 5번, 6번 활성화 |
| 2 | dio.reset_bits(0x0011) | -- 디지털 출력 1번, 5번 비활성화 |
| 3 | dio.set_bits(0x103000200000) | -- 디지털 출력 22번, 37번, 38번 45번 활성화 |
| 4 | | |