

ARIS Manual

2024. 10. 28.

작성자 : 함종수

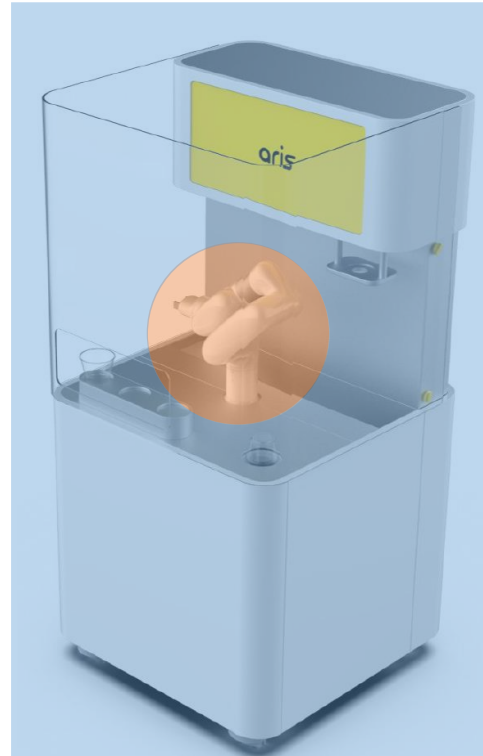
1. ARIS 소개
2. UFACTORY Studio 사용
3. Python SDK



1. ARIS 소개

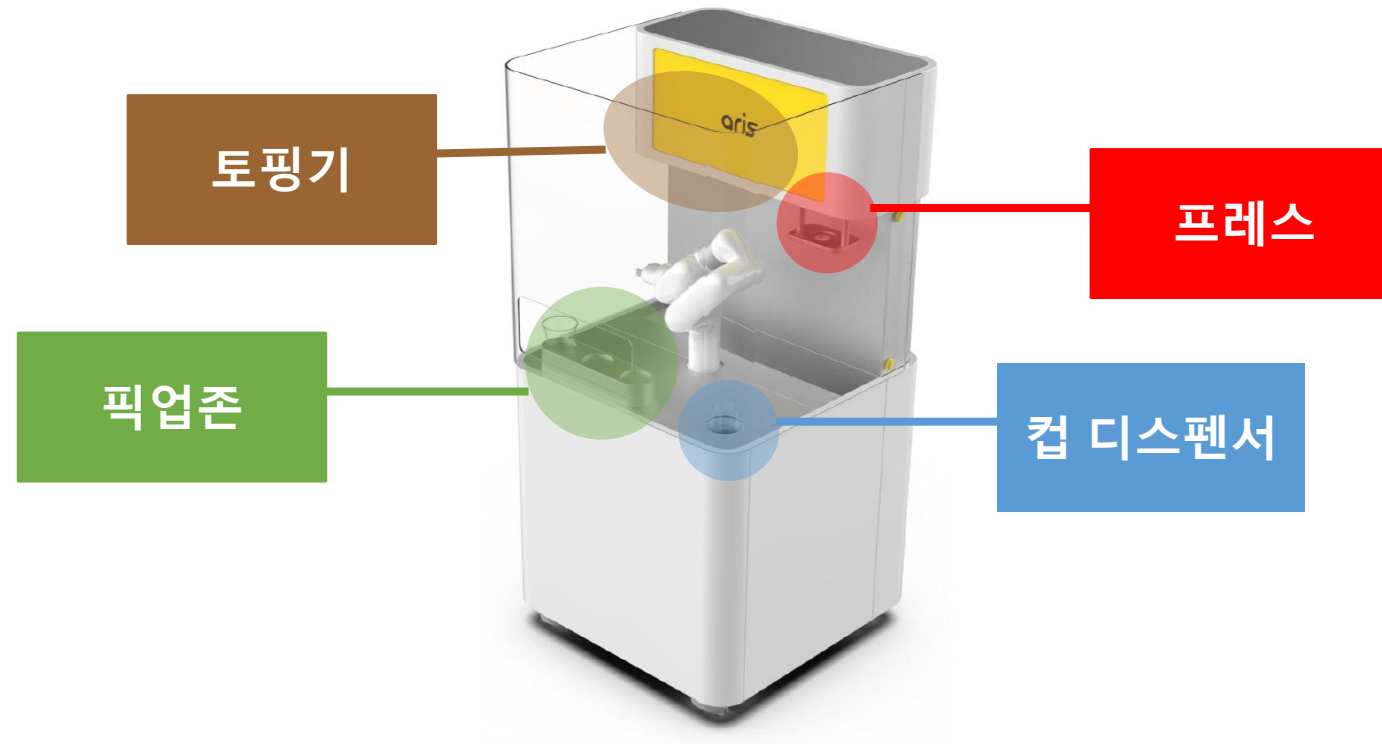
- ARIS 개요

6 DOF 로봇팔 (UFACTORY Lite 6) + 아이스크림 장비 (엑스와의지)



1. ARIS 소개

- 아이스크림 장비 구성



2. UFACTORY Studio 사용

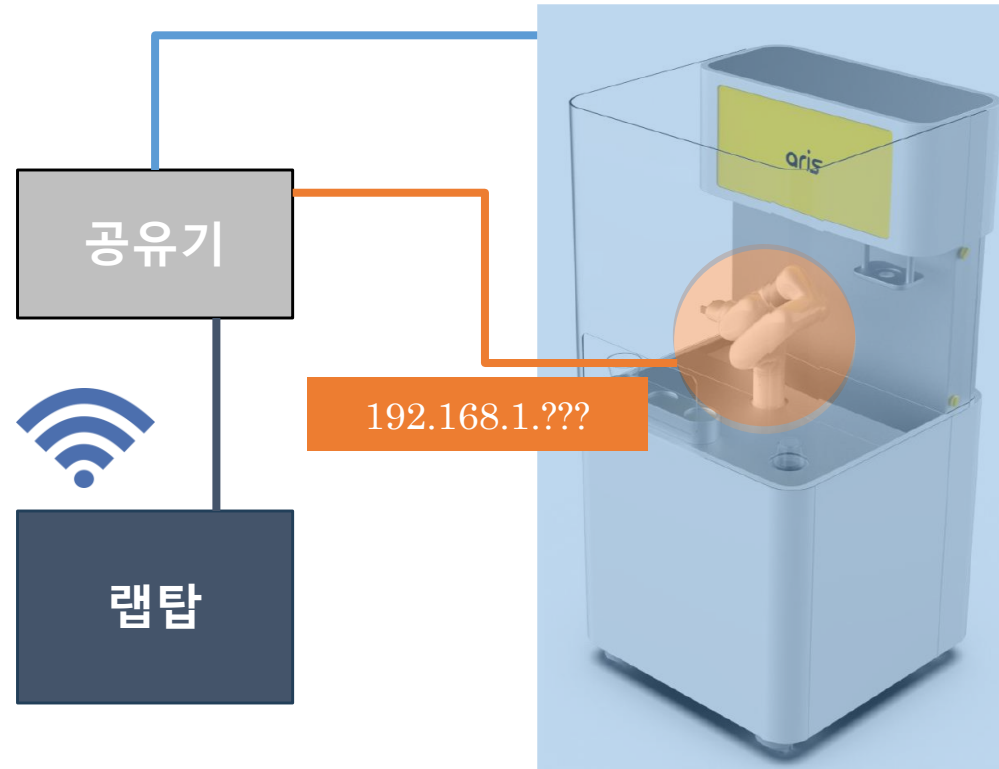
2. UFACTORY Studio 사용

- 연결

6 DOF 로봇팔 (UFACTORY Lite 6) + 아이스크림 장비 (엑스와이지)

Name : S/N

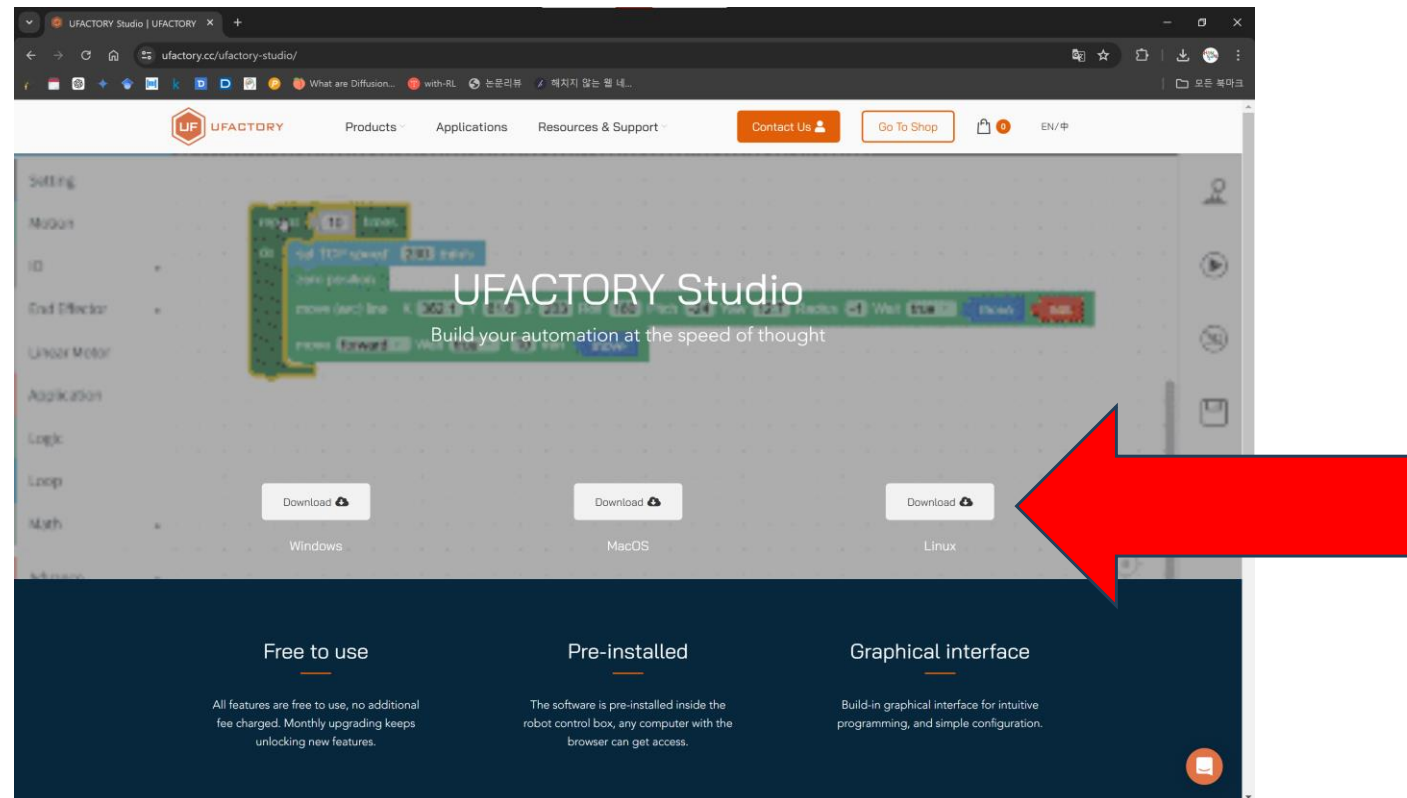
PW : xyz20190529!



2. UFACTORY Studio 사용

- UFACTORY Studio 설치

- <https://www.ufactory.cc/ufactory-studio/>



2. UFACTORY Studio 사용

UFACTORY Studio
Window Language Tool

Live Control

Blockly

Settings

STOP

Gripper

OPEN

CLOSE

OFF

IP: 192.168.1.192

Payload: 0.14KG

Status: Normal

Mode: Position

X: -213.3 mm

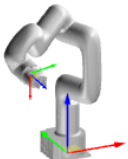
Y: -155.7 mm

Z: 378 mm

R: 146.8 °

P: 76.7 °

Y: 63.9 °



Base 좌표계 표시

Tool 좌표계 표시

Parameters

Enable Robot

Real

Simulated

Position Control

Joint Control

Recording

End Effector

Speed

로봇 동작속도 조절

203.3

50%

Coord

조작시 기준좌표계

Base

Tool

Adjust

기초 포즈 2가지

Initial Position

Align

Control

좌표계를 사용한 로봇 조작

Z-

Z+

Y-

Y+

X+

Y+

XYZ

Y-

X-

P+

R-

RPY

R+

P-

2. UFACTORY Studio 사용

UFACTORY Studio
Window Language Tool

Live Control

Blockly

Settings

STOP

Gripper

OPEN

CLOSE

OFF

IP: 192.168.1.192

Payload: 0.14KG

Status: Normal

Mode: Position

X: -213.3 mm

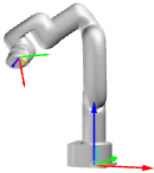
Y: -155.7 mm

Z: 378 mm

R: 146.8 °

P: 76.7 °

Y: 63.9 °



Position Control

Joint Control

Recording

End Effector

Manual Mode

Sensitivity

203.3

1

Manual Mode

수동 조작 모드 on/off

Joint Control

로봇 관절 조작

J1

-

+

203.4 °

J2

-

+

1 °

J3

-

+

57.4 °

J4

-

+

89.2 °

J5

-

+

71 °

J6

-

+

-294.6 °

Parameters

Enable Robot

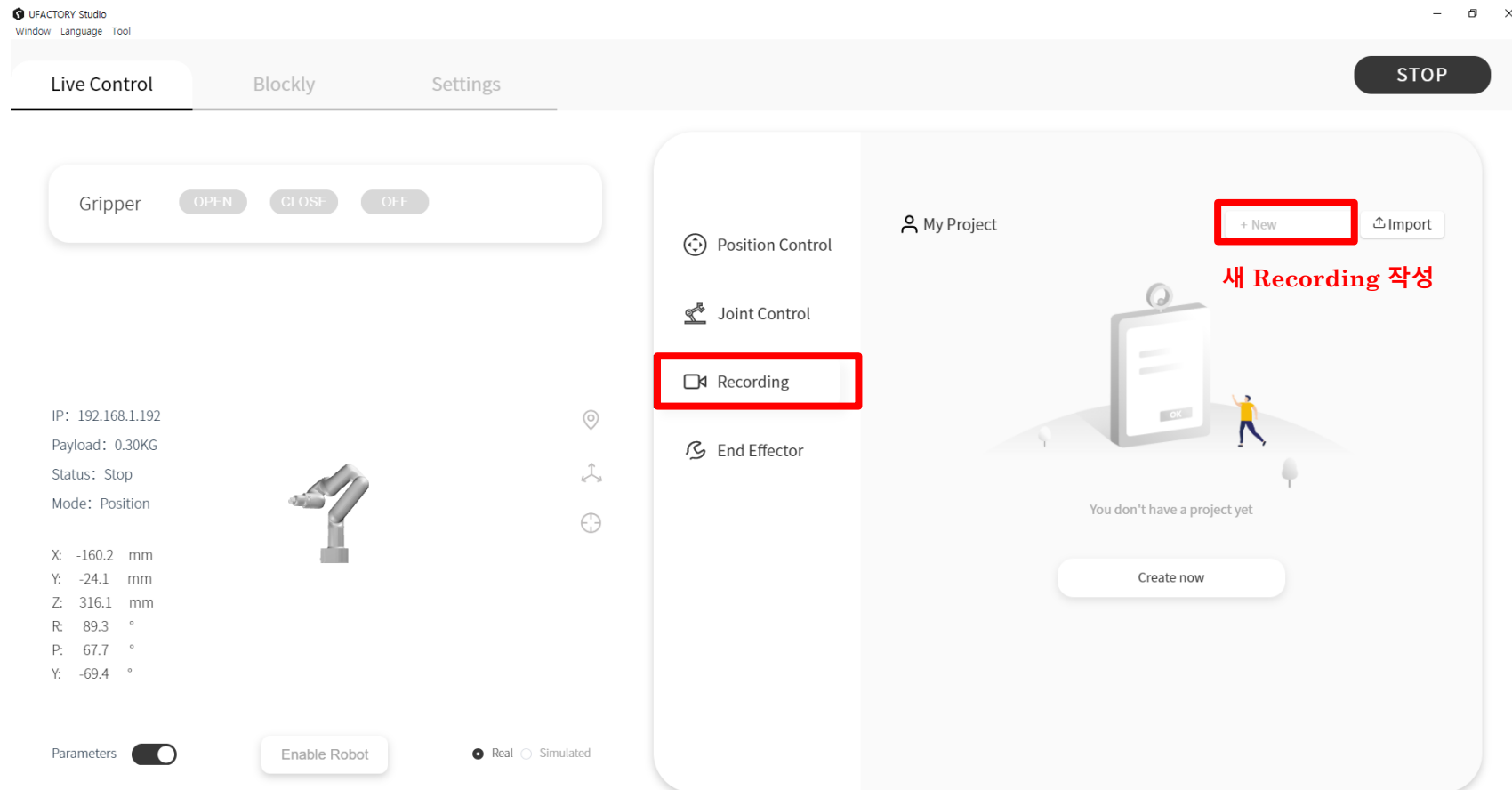
Real

Simulated

2. UFACTORY Studio 사용

- Recording

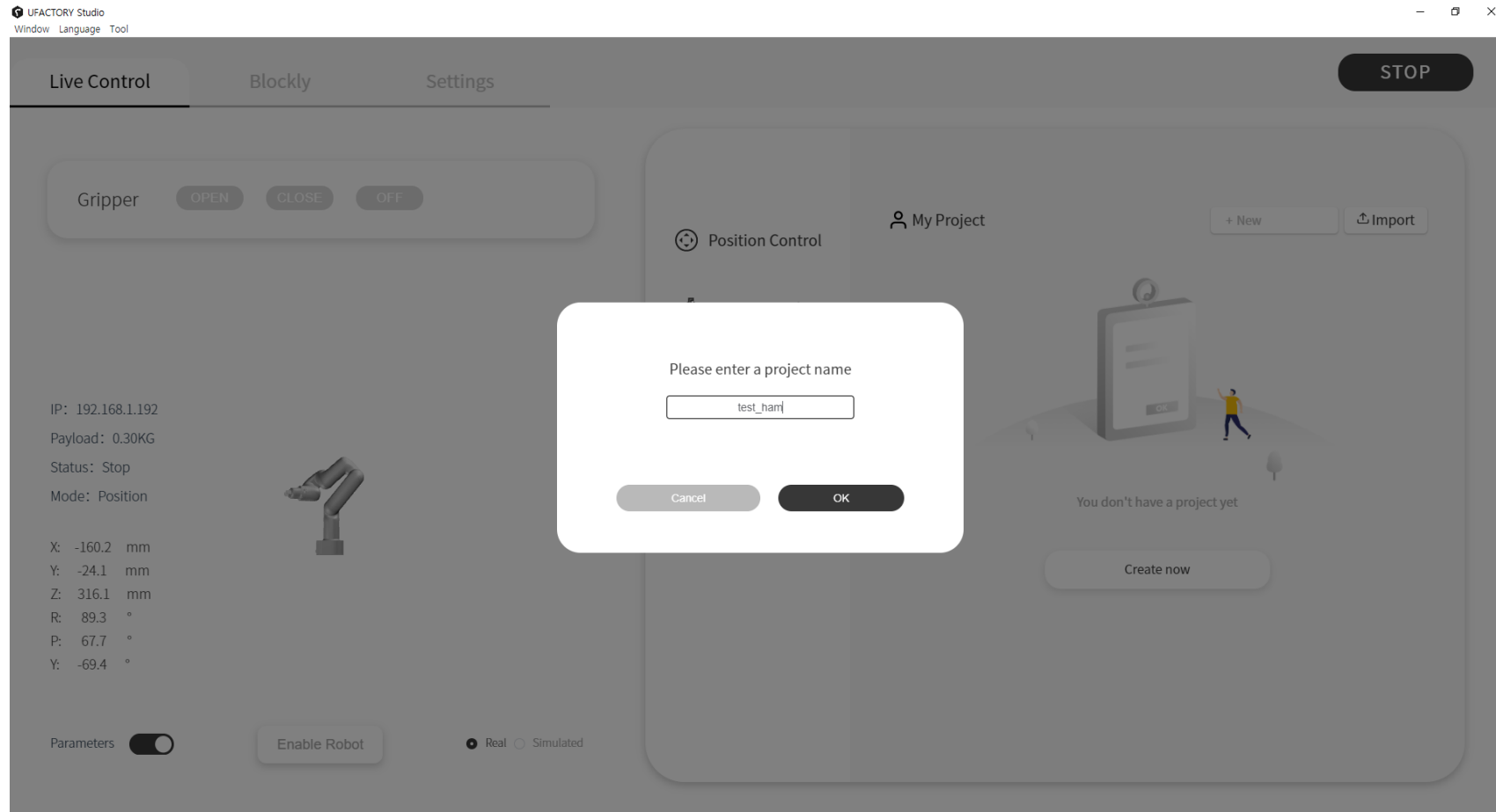
1. +New 버튼으로 새로운 Recording 작성



2. UFACTORY Studio 사용

- **Recording**

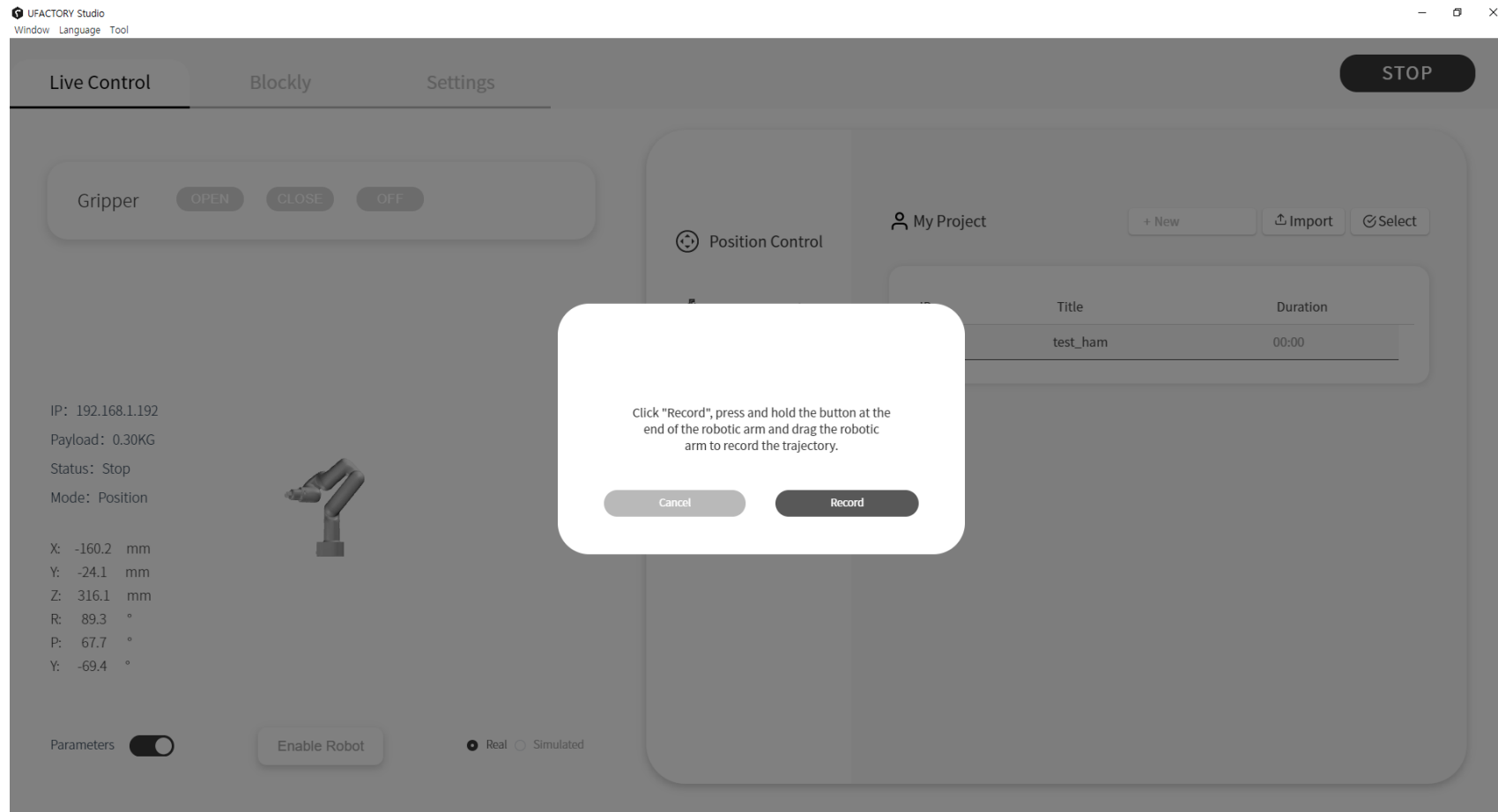
2. Recording 할 로봇 경로명 지정



2. UFACTORY Studio 사용

- **Recording**

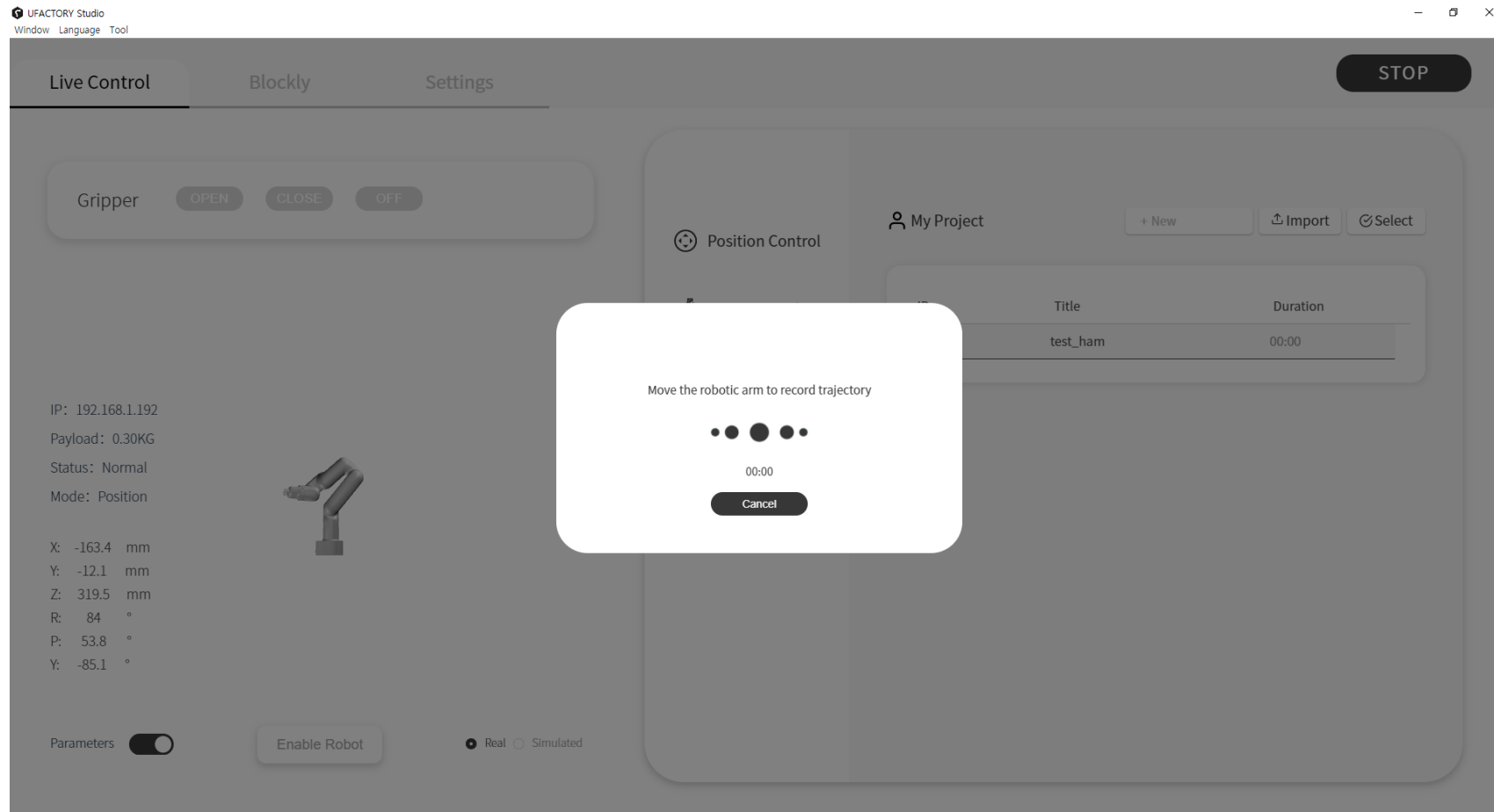
3. 기록할 준비가 끝나면 Record 버튼 눌러서 기록 시작



2. UFACTORY Studio 사용

- **Recording**

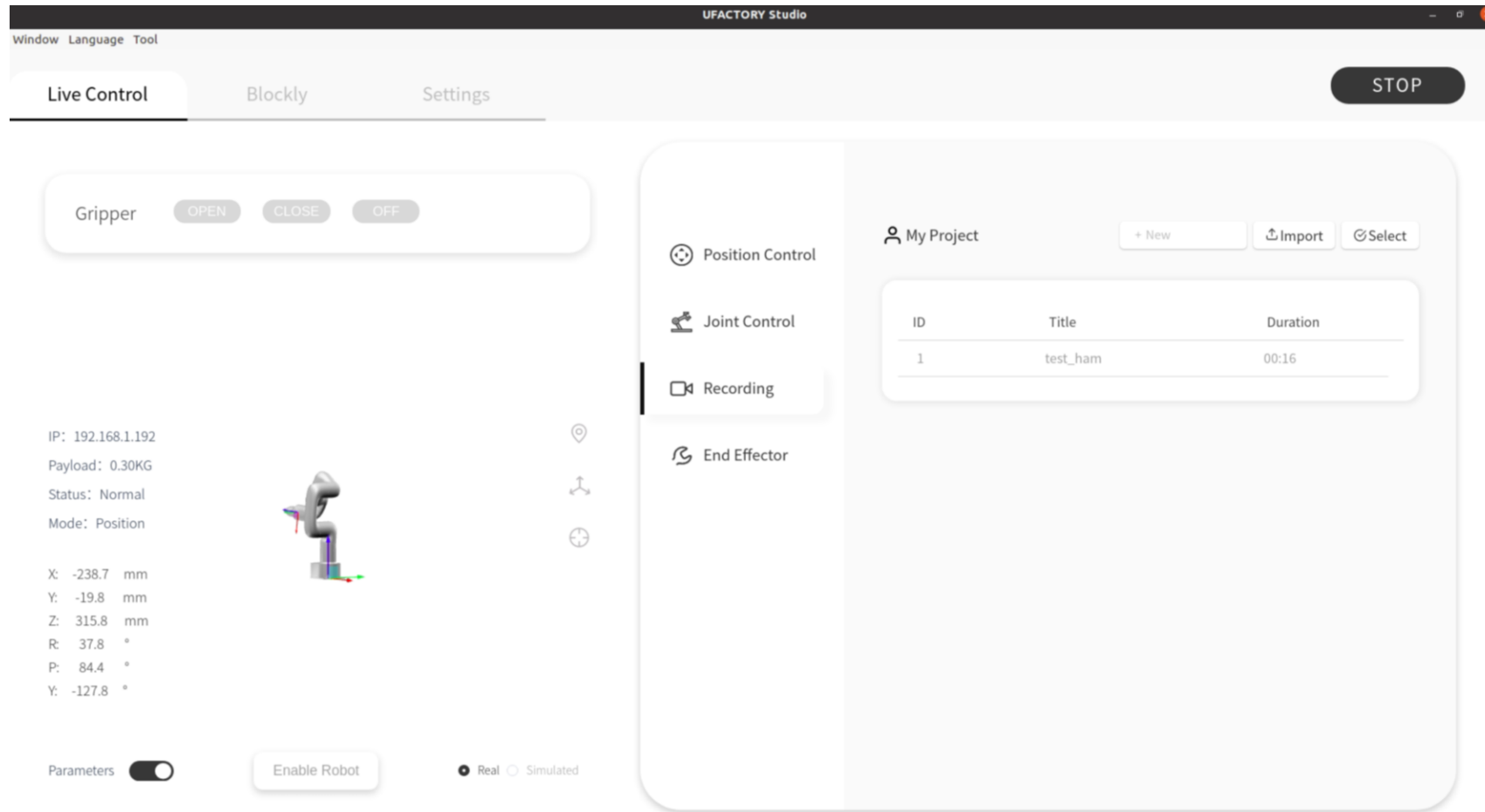
4. 아래와 같은 화면에서 로봇팔을 수동 조작하여 기록 진행



2. UFACTORY Studio 사용

- **Recording**

5. 기록이 완료되면 저장



2. UFACTORY Studio 사용

UFACTORY Studio

Window Language Tool

Live Control

Blockly

Settings

STOP

Gripper

OPEN

CLOSE

OFF

그리퍼 열기

그리퍼 닫기

그리퍼 멈춤
(컴프레서 off)

IP: 192.168.1.192

Payload: 0.30KG

Status: Normal

Mode: Position

X: -238.7 mm

Y: -19.8 mm

Z: 315.8 mm

R: 37.8 °

P: 84.4 °

Y: -127.8 °

Parameters ☒

Enable Robot

☒ Real ☐ Simulated

Position Control

Joint Control

Recording

End Effector

Gripper

No End Effector

Vacuum Gripper

2. UFACTORY Studio 사용

UFACTORY Studio
Window Language Tool

Live Control Blockly **Settings** STOP

Gripper OPEN CLOSE OFF

IP: 192.168.1.190
Payload: 0.30KG
Status: Stop(5)
Mode: Position

X: -278.4 mm
Y: 3.4 mm
Z: 267.1 mm
R: 59.2 °
P: -68.6 °
Y: -52 °

Parameters ☒ Enable Robot ☒ Real ☐ Simulated

Externals

Robot IO
Controller IO
Modbus RTU

Safety

General

My Device

Function IO Debug

- CI0 : Safeguard Reset
- CI1 : Enable Robot
- CI2 : Manual Mode
- CI3 : Offline Task
- CI4 : Stop Moving
- CI5 : General Input
- CI6 : General Input
- CI7 : General Input

- C00 : General Output**
- C01 : General Output**
- C02 : General Output**
- C03 : General Output**
- C04 : Robot Enabled
- C05 : Offline Task Runnii
- C06 : Error
- C07 : Manual Mode

Topping
Press motor

2. UFACTORY Studio 사용

UFACTORY Studio

Window Language Tool

Live Control **Blockly** Settings

STOP

Gripper OPEN CLOSE OFF

IP: 192.168.1.192
Payload: 0.30KG
Status: Normal
Mode: Position

X: -238.7 mm
Y: -19.8 mm
Z: 315.8 mm
R: 37.8 °
P: 84.4 °
Y: -127.8 °

Parameters ☒ Enable Robot ☒ Real ☐ Simulated

Project Program

+New Folder

- UF_APPS
 - [UF]-1001_Joint...
 - [UF]-1002_Conti...
 - [UF]-1003_Linear...
 - [UF]-1004_Conti...
 - [UF]-1005_DrawCr...
 - [UF]-1006_Lite6_G...
 - [UF]-1007_Lite6_V...
 - [UF]-1008_Digital_IO
 - [UF]-1009_Multipl...
- USER_APPS
 - j5
 - test

to motion1

linear motion X 200 Y 200 Z 300 Roll 180 Pitch 0 Yaw 0 Radius 0 Wait true move edit

move up Wait true 50 mm move

move down Wait true 30 mm move

change var_ham by 5

repeat var_ham times

do

to motion2

linear motion X 200 Y 200 Z 300 Roll 180 Pitch 0 Yaw 0 Radius 0 Wait true move edit

move up Wait true 50 mm move

move down Wait true 30 mm move

set TCP speed: 200 mm/s

set TCP acceleration: 2000 mm/s²

linear motion X 200 Y 0 Z 300 Roll 180 Pitch 0 Yaw 0 Radius 0 Wait true move edit

forever

set FLAG to 0

if C10 - C17 are LOW HIGH HIGH HIGH HIGH HIGH HIGH HIGH timeout 0.02

do set FLAG to 1

else if C10 - C17 are HIGH LOW HIGH HIGH HIGH HIGH HIGH HIGH timeout 0.02

do set FLAG to 2

else set FLAG to 0

if FLAG = 1

do motion1

set FLAG to 0

else if FLAG = 2

do motion2

set FLAG to 0

2. UFACTORY Studio 사용

UFACTORY Studio

Window Language Tool

Live Control Blockly Settings

STOP

Project Program

Setting

Motion

IO

End Effector

Application

Logic

Loop

Math

Advance

IP: 192.168.1.192
Payload: 0.30KG
Status: Normal
Mode: Position

X: -189 mm
Y: -69.5 mm
Z: 304.7 mm
R: 33.1 °
P: 82.1 °
Y: -99.6 °

Parameters ☒ Enable Robot ☐ Real ☒ Simulated

Joint motion 등의 블록을 드래그해서 사용하면 현재 포즈 추출 가능

motion resume
emergency stop
zero position
joint motion J1 0 J2 0 J3 0 J4 0 J5 0 J6 0 Radius 0 Wait false move edit
linear motion X 201.5 Y 0 Z 140.5 Roll -180 Pitch 0 Yaw 0 Radius 0 Wait false move edit
move forward Wait false 10 mm move
remark Continuous Joint Motion
move tool line X 0 Y 0 Z 0 Roll 0 Pitch 0 Yaw 0 Wait false
move circle
Pose1 cartesian X 201.5 Y 0 Z 140.5 Roll -180 Pitch 0 Yaw 0 move edit
Pose2 cartesian X 201.5 Y 0 Z 140.5 Roll -180 Pitch 0 Yaw 0 move edit
Center angle(*) 360 -18.4 36.4 69.6 0 33.2 -18.4 Radius 50 Wait false move edit
Wait false move
joint motion [variable] J1 0 J2 0 J3 0 J4 0 J5 0 J6 0 Radius 0 Wait false
linear motion [variable] X 201.5 Y 0 Z 140.5 Roll 180 Pitch 0 Yaw 0 Radius 0 Wait false

[2024-06-29 16:11:47][151] xArm-Python-SDK Version: 1.12.1

Live Control

Blockly

Settings

STOP

B

Gripper

OPEN

CLOSE

OFF

IP: 192.168.1.190

Payload: 0.30KG

Status: Stop(5)

Mode: Position

X: -278.4 mm

Y: 3.4 mm

Z: 267.1 mm

R: 59.2 °

P: -68.6 °

Y: -52 °



Parameters ☒

Enable Robot

☒ Real ☐ Simulated

Project

Program

Options

UF_APPS

[UF]-1001_Joint...

[UF]-1002_Continu...

[UF]-1003_Linear...

[UF]-1004_Continu...

[UF]-1005_DrawCri...

[UF]-1006_Lite6_G...

[UF]-1007_Lite6_V...

[UF]-1008_Digital_IO

[UF]-1009_Multipl...

USER_APPS

test0



2. UFACTORY Studio 사용

The screenshot displays the UFACTORY Studio web interface. At the top, there's a menu bar with 'Window', 'Language', and 'Tool'. Below it, a tab bar shows 'Live Control', 'Blockly' (selected), and 'Settings'. A 'STOP' button is in the top right. On the left, a 'Gripper' control panel has 'OPEN', 'CLOSE', and 'OFF' buttons. Below this, robot status information is shown: IP: 192.168.1.192, Payload: 0.30KG, Status: Normal, Mode: Position, and joint coordinates (X, Y, Z, R, P, Y). A 3D robot arm model is in the center. At the bottom, there's a 'Parameters' toggle, an 'Enable Robot' button, and 'Real'/'Simulated' mode selectors. The main area is divided into a 'Project' pane on the left and a 'Program' editor on the right. The 'Project' pane lists files and folders, with '+New File' and the 'ham' folder highlighted by red boxes. The 'Program' editor is currently empty.

UFACTORY Studio

Window Language Tool

Live Control **Blockly** Settings

STOP

Gripper OPEN CLOSE OFF

IP: 192.168.1.192
Payload: 0.30KG
Status: Normal
Mode: Position

X: -156.7 mm
Y: 6.9 mm
Z: 310.9 mm
R: 77.1 °
P: 86.4 °
Y: -108.1 °

Parameters ☒ Enable Robot ☒ Real ☐ Simulated

Project Program

+New File

- [UF]-1004_Contin...
- [UF]-1005_DrawCr...
- [UF]-1006_Lite6_G...
- [UF]-1007_Lite6_V...
- [UF]-1008_Digital_IO
- [UF]-1009_Multipl...
- USER_APPS
 - j5
 - test
- sksj
 - test_aris
 - ham**
 - test01

폴더를 선택하면 New File 버튼 활성화

2. UFACTORY Studio 사용

The screenshot displays the UFACTORY Studio web interface. At the top, there's a header bar with 'UFACTORY Studio' and window controls. Below it, a navigation bar includes 'Live Control', 'Blockly' (selected), and 'Settings'. A 'STOP' button is on the right. The main area is divided into three sections. On the left, a 'Gripper' control panel has 'OPEN', 'CLOSE', and 'OFF' buttons. Below it, robot status information is shown: IP: 192.168.1.192, Payload: 0.30KG, Status: Normal, Mode: Position, and joint coordinates (X, Y, Z, R, P, Y). A 3D robot arm model is in the center. On the right, a file explorer panel shows a list of files and folders. The '+New Folder' button at the top is highlighted with a red box. The file list includes various project files and folders like 'USER_APPS', 'j5', 'test', 'sksj', 'test_aris', 'ham', and 'test01'. The 'test01' file is also highlighted with a red box. A red text annotation points to the 'test01' file, stating '파일을 선택하면 New Folder 버튼 활성화' (New Folder button is activated when a file is selected).

UFACTORY Studio

Window Language Tool

Live Control Blockly Settings

STOP

Gripper OPEN CLOSE OFF

IP: 192.168.1.192
Payload: 0.30KG
Status: Normal
Mode: Position

X: -156.7 mm
Y: 6.9 mm
Z: 310.9 mm
R: 77.1 °
P: 86.4 °
Y: -108.1 °

Parameters ☒ Enable Robot ☒ Real ☐ Simulated

Project Program

+New Folder

- [UF]-1004_Contin...
- [UF]-1005_DrawCr...
- [UF]-1006_Lite6_G...
- [UF]-1007_Lite6_V...
- [UF]-1008_Digital_IO
- [UF]-1009_Multipl...
- USER_APPS
 - j5
 - test
- sksj
 - test_aris
- ham
 - test01

파일을 선택하면 New Folder 버튼 활성화

2. UFACTORY Studio 사용

The screenshot displays the UFACTORY Studio web interface. At the top, there's a header bar with 'UFACTORY Studio' and window controls. Below it, a navigation bar includes 'Live Control', 'Blockly' (selected), and 'Settings'. A 'STOP' button is on the right. The main area is divided into three sections. On the left, a 'Gripper' control panel has 'OPEN', 'CLOSE', and 'OFF' buttons. Below it, robot status information is shown: IP: 192.168.1.192, Payload: 0.30KG, Status: Normal, Mode: Position, and joint coordinates (X, Y, Z, R, P, Y). A 3D robot arm model is in the center. On the right, a file explorer panel shows a list of files and folders. A red box highlights the upload icon (an arrow pointing up) next to the '+New Folder' button. A red text label 'PC의 블록리 파일을 가져오는 버튼' points to this icon. The file list includes folders like 'USER_APPS' and 'ham', and files like 'test01'.

UFACTORY Studio

Window Language Tool

Live Control Blockly Settings

STOP


Gripper OPEN CLOSE OFF

IP: 192.168.1.192
Payload: 0.30KG
Status: Normal
Mode: Position

X: -156.7 mm
Y: 6.9 mm
Z: 310.9 mm
R: 77.1 °
P: 86.4 °
Y: -108.1 °

Parameters ☒ Enable Robot ☒ Real ☐ Simulated

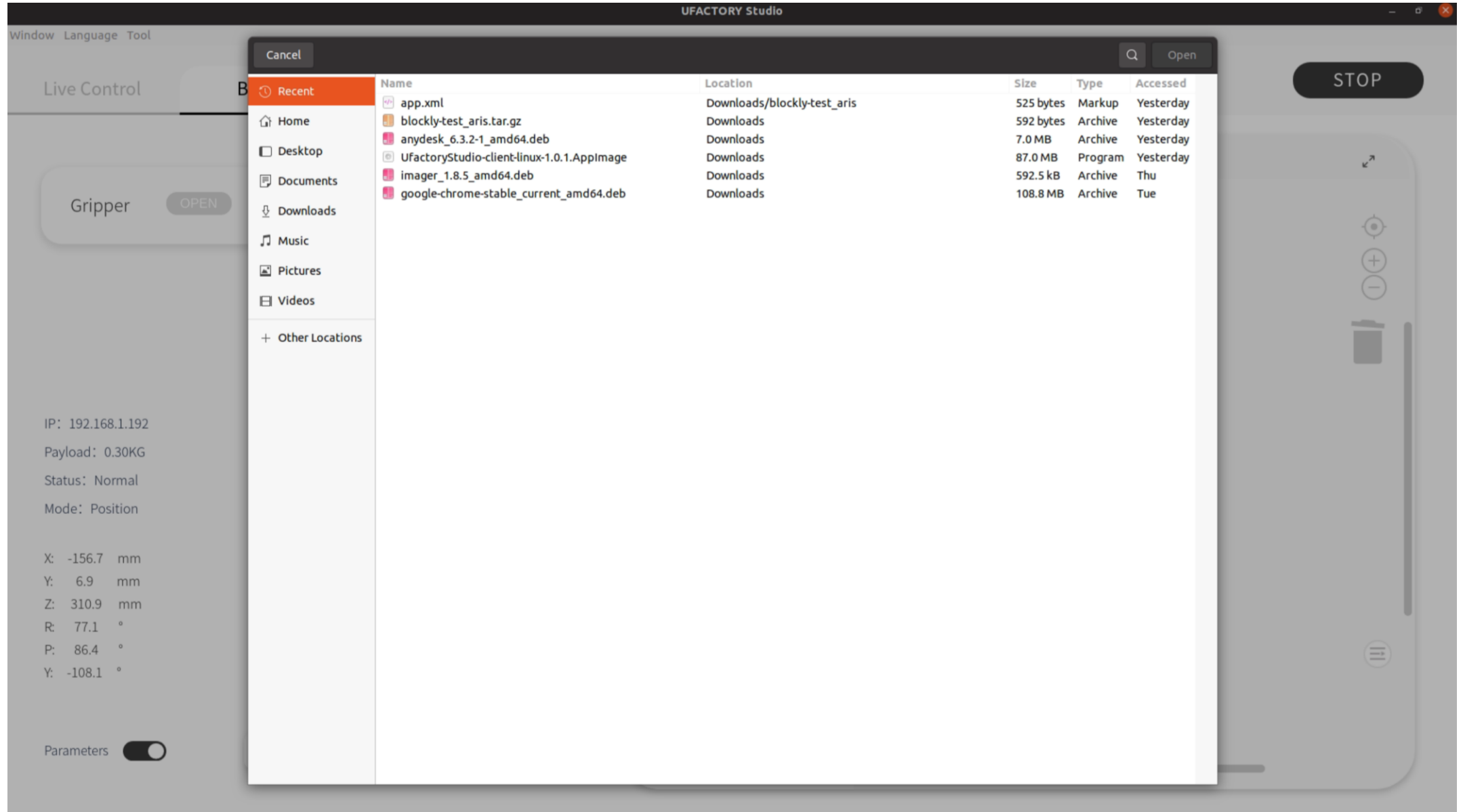
Project Program

+New Folder 

PC의 블록리 파일을 가져오는 버튼

[UF]-1004_Contin...
[UF]-1005_DrawCr...
[UF]-1006_Lite6_G...
[UF]-1007_Lite6_V...
[UF]-1008_Digital_IO
[UF]-1009_Multipl...
USER_APPS
j5
test
sksj
test_aris
ham
test01

2. UFACTORY Studio 사용



2. UFACTORY Studio 사용

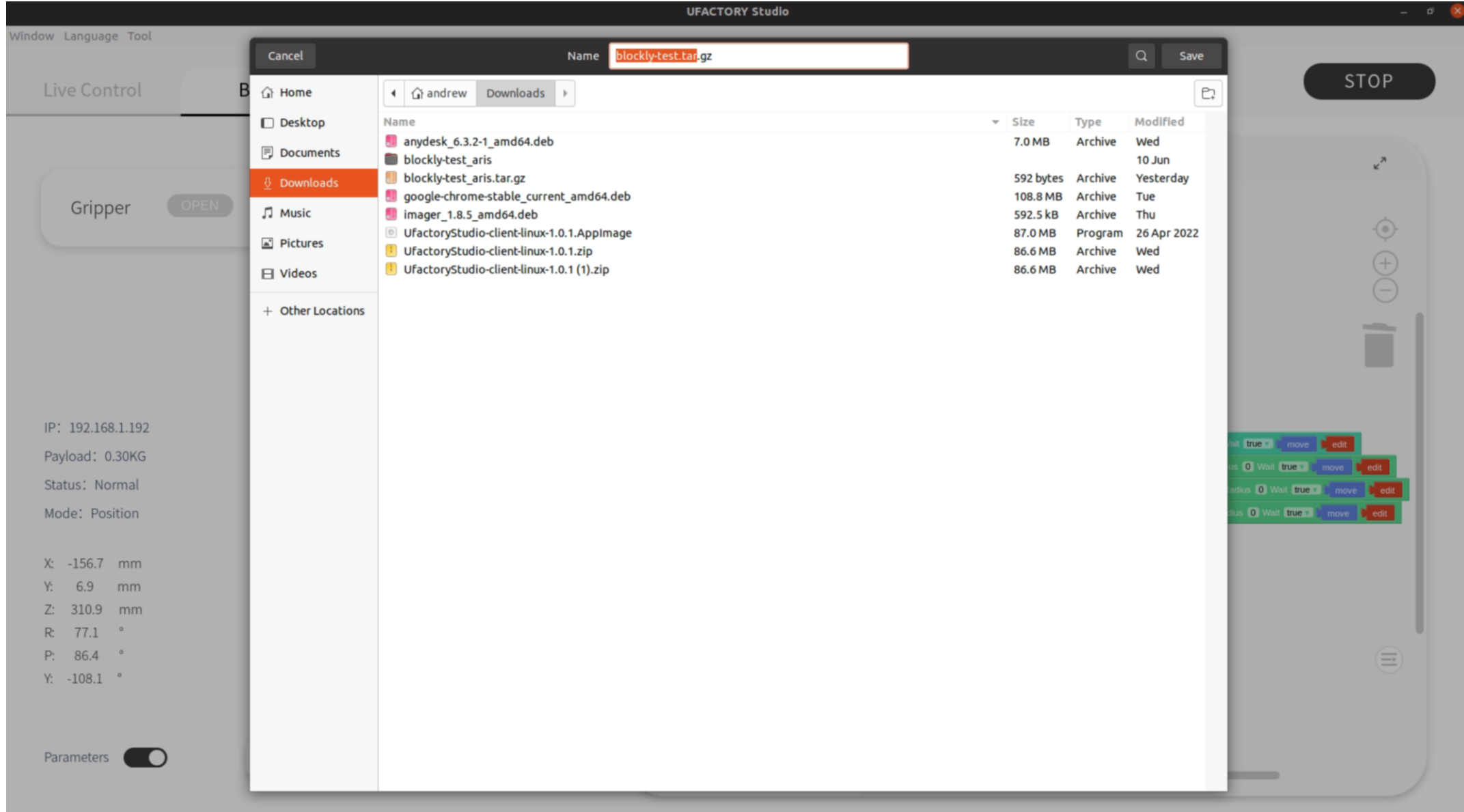
The screenshot displays the UFACTORY Studio web interface. At the top, there are tabs for 'Live Control', 'Blockly' (selected), and 'Settings'. A 'STOP' button is in the top right. On the left, a 'Gripper' control panel has 'OPEN', 'CLOSE', and 'OFF' buttons. Below it, robot status information is shown: IP: 192.168.1.192, Payload: 0.30KG, Status: Normal, Mode: Position, and joint coordinates (X, Y, Z, R, P, Y). A 3D model of a robotic arm is in the center. At the bottom left, there are 'Parameters' and 'Enable Robot' buttons, and a 'Real' vs 'Simulated' toggle.

The right side features a 'Project' and 'Program' tab. The 'Project' tab shows a file tree with folders like 'USER_APPS' and 'test'. A red arrow points to the 'test' folder, which has a context menu open with 'Download', 'Rename', and 'Delete' options. The 'Program' tab shows a Blockly script with blocks for joint motion, linear motion, and wait.

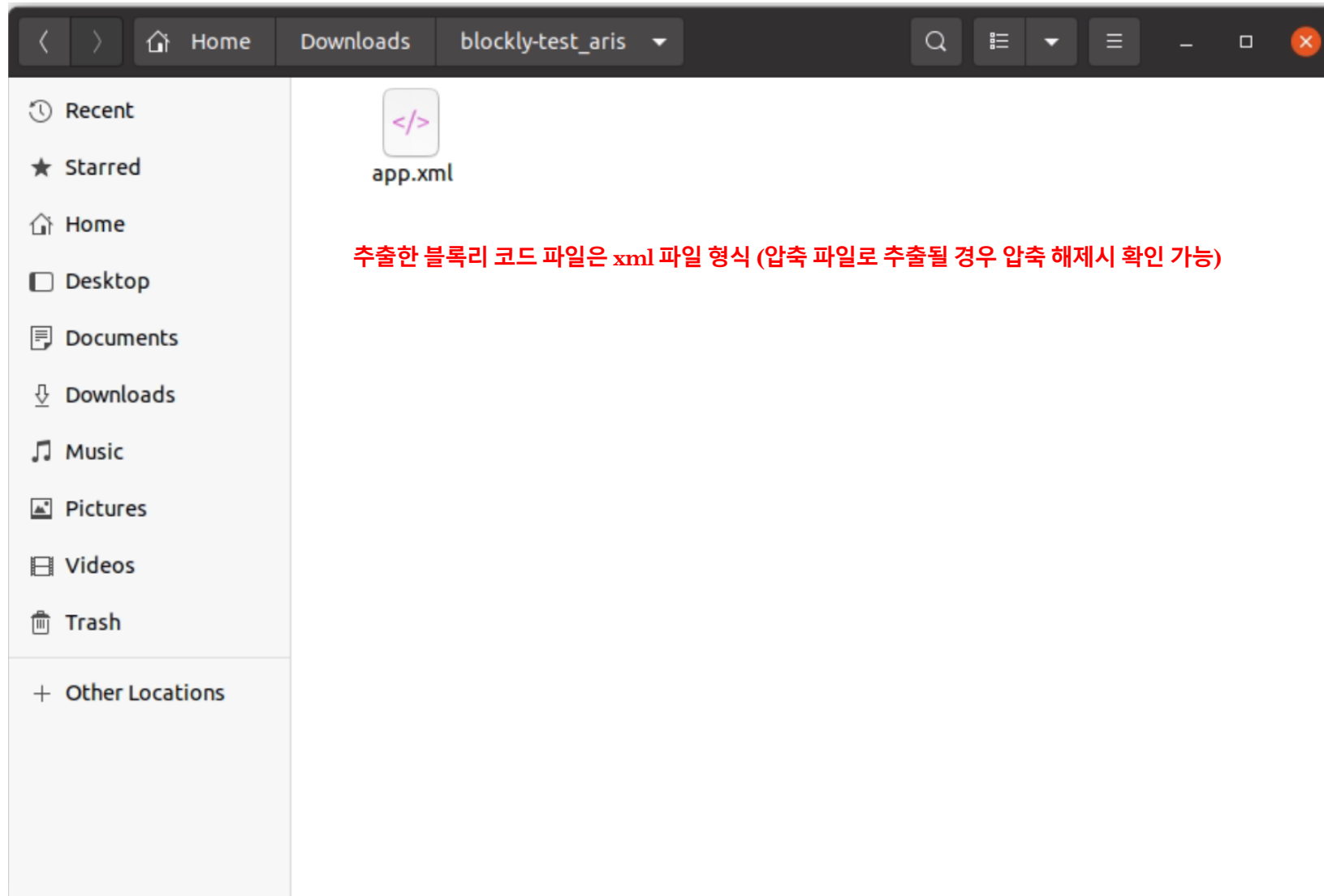
Download
Rename
Delete

- : 작성한 블록리 파일을 PC로 추출
- : 작성한 블록리 파일명 변경
- : 작성한 블록리 파일 삭제

2. UFACTORY Studio 사용



2. UFACTORY Studio 사용



2. UFACTORY Studio 사용

UFACTORY Studio
Window Language Tool

Live ControlBlocklySettings

STOP

GripperOPENCLOSEOFF

IP: 192.168.1.190
Payload: 0.30KG
Status: Stop(5)
Mode: Position
X: -278.4 mm
Y: 3.4 mm
Z: 267.1 mm
R: 59.2 °
P: -68.6 °
Y: -52 °
Parameters ☐ Enable Robot ☒ Real ☐ Simulated

Options

BlocklyToPython

main

main.py

main_v3152.py

main_231208.py

main_240105.py

Python Script Mode

1 # S/N : XYZARIS0V3P2312N03
2 # Robot IP : '192.168.1.190'
3 # script version : v3.1.4
4
5
6
7 #!/usr/bin/env python3
8 # Software License Agreement (BSD License)
9 #
10 # Copyright (c) 2022, UFACTORY, Inc.
11 # All rights reserved.
12 #
13 # Author: Vinman <vinman.wen@ufactory.cc> <vinman.cub@gmail.co
14
15
16 # Notice
17 # 1. Changes to this file on Studio will not be preserved
18 # 2. The next conversion will overwrite the file with the sa
19 #
20 # xArm-Python-SDK: <https://github.com/xArm-Developer/xArm-Pyth>
21 # 1. git clone git@github.com:xArm-Developer/xArm-Python-SDK
22 # 2. cd xArm-Python-SDK
23 # 3. python setup.py install
24
25 import sys
26 import math
27 import time
28 import queue
29 import datetime
30 import random
31 import traceback
32 import threading
33 from xarm import version
34 from xarm.wrapper import XArmAPI
35
36 from threading import Thread, Event
37 import socket
38 import json
39 import os
40
41

2. UFACTORY Studio 사용

로봇팔 조인트 각도로 움직이기

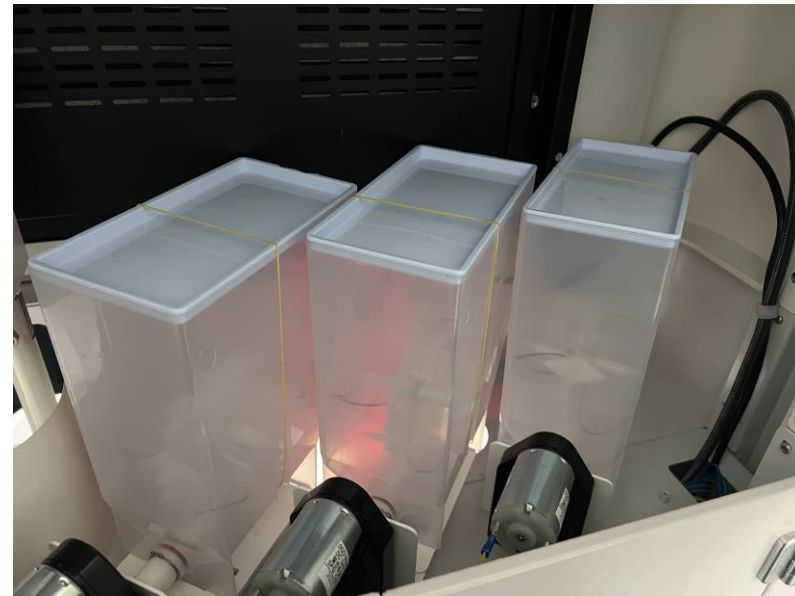
```
code = self._arm.set_servo_angle(angle=[179.0, -17.7, 29.0, 177.8, 43.8, -1.4], speed=self._angle_speed,
                                  mvacc=self._angle_acc, wait=True, radius=0.0)
if not self._check_code(code, 'set_servo_angle'):
    return
```

DIGITAL PIN 사용하기 (토팅 기계 + 프레스)

```
code = self._arm.set_cgpio_digital(Index0, Value0, delay_sec=0)
if not self._check_code(code, 'set_cgpio_digital'):
    return
```



3 (프레스)



2 1 0 (토팅)

2. UFACTORY Studio 사용

ANALOG PIN 사용하기 (컵 디스펜서)

```
code = self._arm.set_cgpio_analog(0, 5)
if not self._check_code(code, 'set_cgpio_analog'):
    return
code = self._arm.set_cgpio_analog(1, 5)
if not self._check_code(code, 'set_cgpio_analog'):
    return
time.sleep(3)
code = self._arm.set_cgpio_analog(0, 0)
if not self._check_code(code, 'set_cgpio_analog'):
    return
time.sleep(3)
code = self._arm.set_cgpio_analog(1, 0)
if not self._check_code(code, 'set_cgpio_analog'):
    return
```

전원 off

Analog Output(C-AO)		
Output	Value (V)	Output
AO0	<input type="text" value="0"/>	<input type="button" value="Set"/>
AO1	<input type="text" value="0"/>	<input type="button" value="Set"/>

좌로 이동

Analog Output(C-AO)		
Output	Value (V)	Output
AO0	<input type="text" value="0"/>	<input type="button" value="Set"/>
AO1	<input type="text" value="5.00"/>	<input type="button" value="Set"/>

우로 이동

Analog Output(C-AO)		
Output	Value (V)	Output
AO0	<input type="text" value="5.00"/>	<input type="button" value="Set"/>
AO1	<input type="text" value="5.00"/>	<input type="button" value="Set"/>

3. Python SDK

3. Python SDK

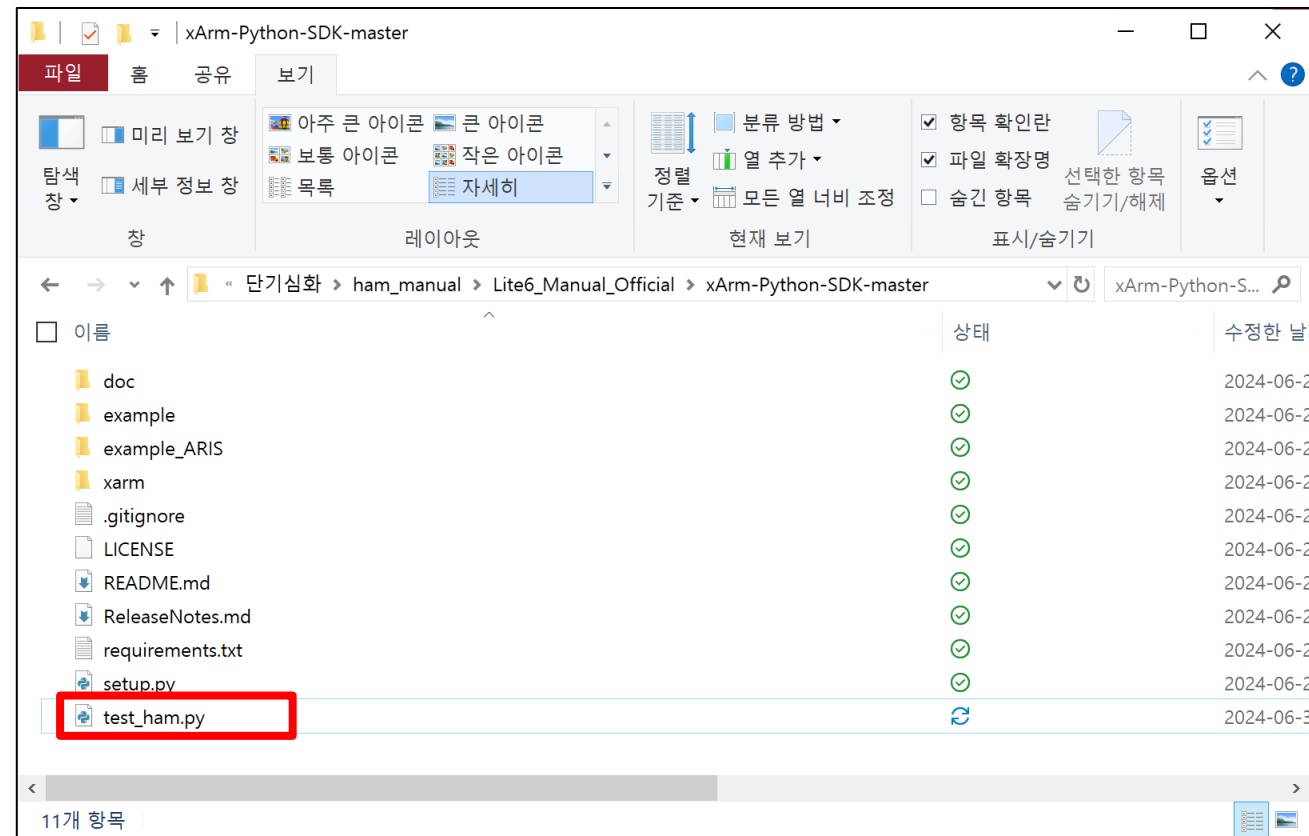
- SDK 사용 환경 구축
 - <https://github.com/xArm-Developer/xArm-Python-SDK>

```
python3 -m venv xarm      가상환경 구축
source xarm/bin/activate  가상환경 활성화
git clone https://github.com/xArm-Developer/xArm-Python-SDK.git
cd xArm-Python-SDK
python setup.py install
```


3. Python SDK

- SDK 기본 사용 방법

- 작성한 로봇팔 시나리오 코드를 sdk 디렉토리에 위치 시킨 후 인터프리터로 실행하면 동작



3. Python SDK

- 코드 작성은 Git 참조 (<https://github.com/xArm-Developer/xArm-Python-SDK>)

- API : https://github.com/xArm-Developer/xArm-Python-SDK/blob/master/doc/api/xarm_api.md

```
code = self._arm.set_servo_angle(angle=[179.0, -17.7, 29.0, 177.8, 43.8, -1.4], speed=self._angle_speed,
                                mvacc=self._angle_acc, wait=True, radius=0.0)
if not self._check_code(code, 'set_servo_angle'):
    return
```

def set_servo_angle(self, servo_id=None, angle=None, speed=None, mvacc=None, mvtime=None, relative=False, is_radian=None, wait=False, timeout=None, radius=None, **kwargs):

Set the servo angle, the API will modify self.last_used_angles value

Note:

1. If the parameter angle you are passing is an radian unit, be sure to set the parameter is_radian to True.
ex: code = arm.set_servo_angle(servo_id=1, angle=1.57, is_radian=True)
2. If you want to wait for the robot to complete this action and then return, please set the parameter wait to True.
ex: code = arm.set_servo_angle(servo_id=1, angle=45, is_radian=False, wait=True)
3. This interface is only used in the base coordinate system.

:param servo_id: 1-(Number of axes), None(8)

1. 1-(Number of axes) indicates the corresponding joint, the parameter angle should be a numeric value
ex: code = arm.set_servo_angle(servo_id=1, angle=45, is_radian=False)
2. None(8) means all joints, default is None, the parameter angle should be a list of values whose length is the number of joints
ex: code = arm.set_servo_angle(angle=[30, -45, 0, 0, 0, 0, 0], is_radian=False)

:param angle: angle or angle list, (unit: rad if is_radian is True else °)

1. If servo_id is 1-(Number of axes), angle should be a numeric value
ex: code = arm.set_servo_angle(servo_id=1, angle=45, is_radian=False)
2. If servo_id is None or 8, angle should be a list of values whose length is the number of joints
like [axis-1, axis-2, axis-3, axis-3, axis-4, axis-5, axis-6, axis-7]
ex: code = arm.set_servo_angle(angle=[30, -45, 0, 0, 0, 0, 0], is_radian=False)

:param speed: move speed (unit: rad/s if is_radian is True else °/s), default is self.last_used_joint_speed

:param mvacc: move acceleration (unit: rad/s^2 if is radian is True else °/s^2), default is self.last used joint acc