RAINBOW ROBOTICS

RB SERIES
USER MANUAL

ENGLISH
Ver. V6.3

Update : 2024/10

INDEX

PREFACE

Before installing this product, please read the user manual thoroughly. Please follow the instructions in the manual according to the installation procedure. The contents of this manual are based on the version of the manual when it was written, and the information about the product may have been changed without notifying the user in advance.

If you are unsure about the requirements, recommendations or safety procedures described in this manual, please contact Rainbow Robotics. Some illustrations in this manual are intended to help you understand the concepts and installation of the system and may differ from actual products.

CHAPTER 1. INTRODUCTION

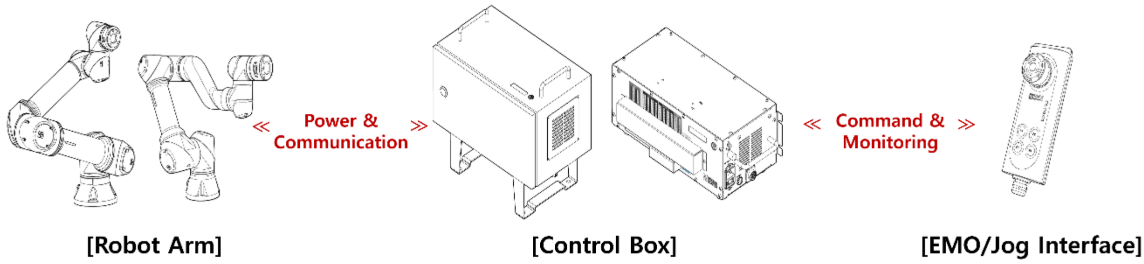
1.1 RAINBOW ROBOTICS' COLLABORATIVE ROBOTS

The RB product line from Rainbow Robotics is a series of collaborative robots. The RB series is designed to be easily accessible and usable to anyone, such as laymen or enthusiasts. The RB series is specialized to perform regular, continuous, and repetitive tasks in small and dense human environments across various fields without any additional safety devices. The RB series provide robotic solutions to increase productivity for your business.

- **Intuitive usability**: It is easy to set up and operate an RB robot unit. Experts and nonexperts alike can use it effectively through the intuitive visual User Interface (UI) configuration.
- **Convenience and safety**: The RB series has external and self-collision detection systems, which minimize accidents and injuries while providing a safe work environment for the operator.
- **Space efficiency**: An RB unit can be applied to all types of production lines regardless of space. It may be used in many different environments due to versatile configurations that allow it to be installed on a variety of surfaces.

1.2 SYSTEM CONFIGURATION

The system configuration of an RB is illustrated in the figure below.



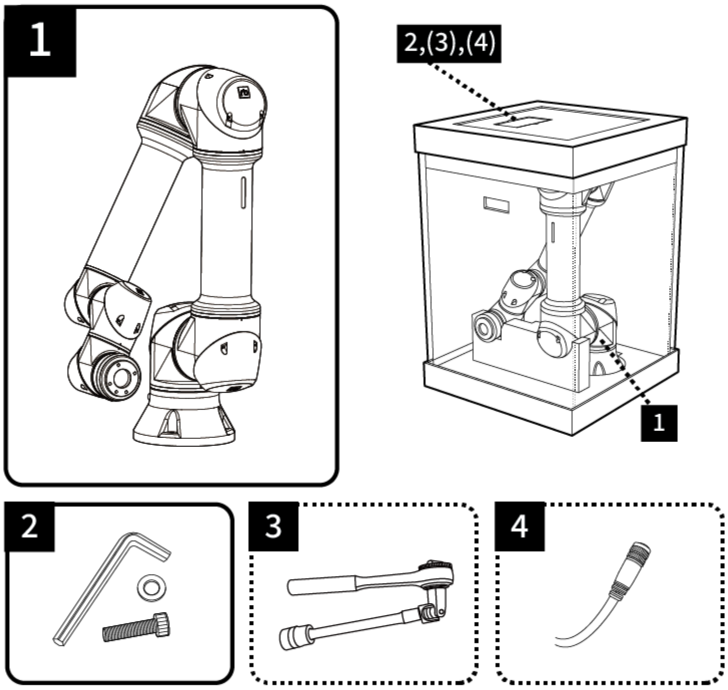
[Stand-type control box system configuration]

※ The product images are for illustrative purposes only, and the appearance may vary depending on the product specifications.

- Robot Arm: The Robot Arm is an industrial collaborative robot that can be used for repetitive routine tasks, carrying small objects, or assembling parts. It can be used with various third-party robotic grippers as well as various types of tools.
- Control Box: The Control Box controls the movement of the robot arm based on user programs contained on the Teach pendant/tablet PC. Digital and analog input/output ports are available for connecting various devices and tools.
- Estop/Jog Interface (for stand-type control box): With the emergency stop switch, the robot operation can be stopped. It comes with simple program flow control buttons such as Play/Stop.
- Teaching Pendant/Tablet PC (optional): The Teaching Pendant/Tablet PC is an external device on which a user can create programs and operate the system. It is used to setup, program, and teach certain postures to the robot arm.

The components of the provided robot system are as follows, and a total of two boxes are included.

- Robot Arm



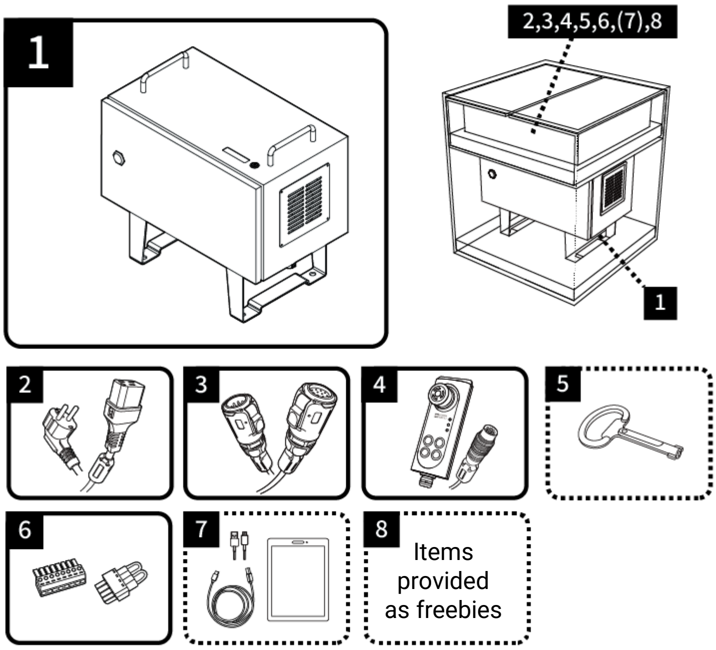
( ) The displayed item is an optional product.

Robot	1	Robot Arm	1 EA
Tool Box	2	Bolt, washer, wrench	1 SET
	(3)	NSF model-specific tool – Base fixing tool	1 EA
	(4)	Tool I/O Cable	1 EA

※ **The NSF model-specific tool applies only to the components of the NSF model.**

※ The components listed in this manual may vary based on the robot specifications, and product images are for illustrative purposes only, so they may differ from the actual product.

■ Control Box



( ) The displayed item is an optional product.

System Components	1	Control Box	1 EA
Accessory Box	2	AC Power Cable	1 SET
	3	Robot - Control Box Connection Cable	1 EA
	4	E-Stop/Jog Interface	1 EA
	5	Control Box Key	1 EA
	6	I/O Terminal Block (spare), Fuse	1 SET
	(7)	Tablet, Short USB Cable, Long USB Cable	1 EA
	8	Service Items Provided (e.g., USB, Velcro for securing external cables, etc.)	1 SET each

※ The appearance of the control box may change depending on the robot specifications.

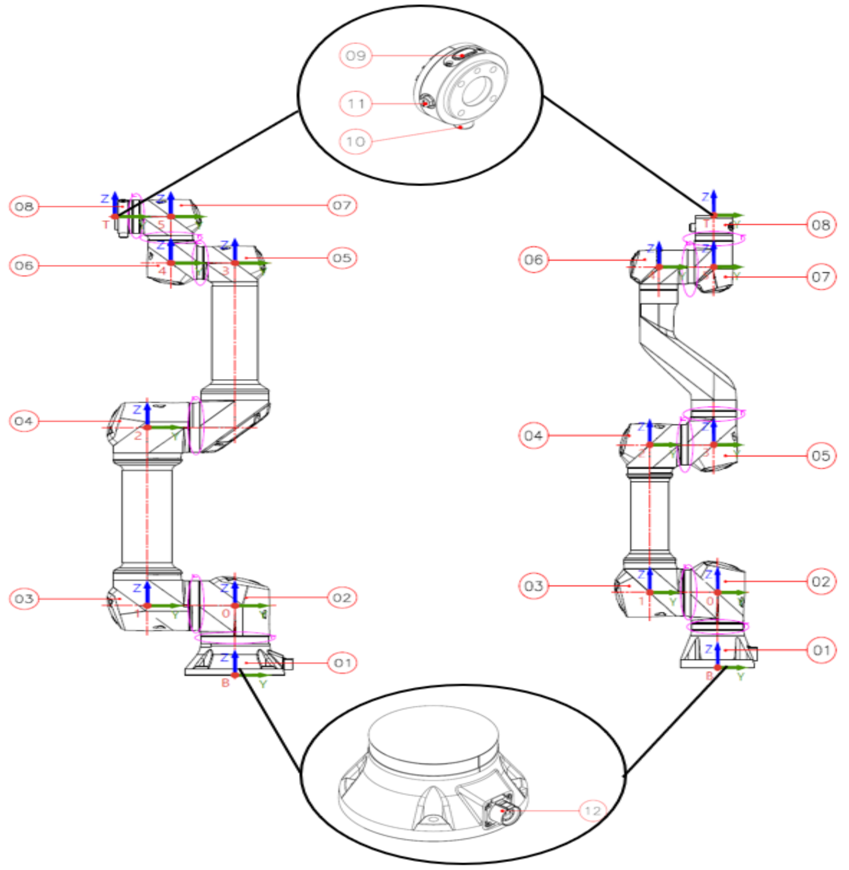
※ It is recommended to use a 5m cable provided by the manufacturer for the AC power cable and the robot arm – control box connection cable, and a cable under 3m provided by the manufacturer for the emergency stop/jog interface cable. For shielded LAN cables, I/O port connection cables, USB cables, and external wire lines for wire-pass models, it is recommended to use cables under 3m.

※ If users are connecting the I/O port of the compact Control Box to the outside, they must install the cable fixing device in the hole provided at the bottom of the protective cover. The manufacturer will not be responsible for any issues arising from improper cable fixing.

※ The items and components provided for service may change without notice.

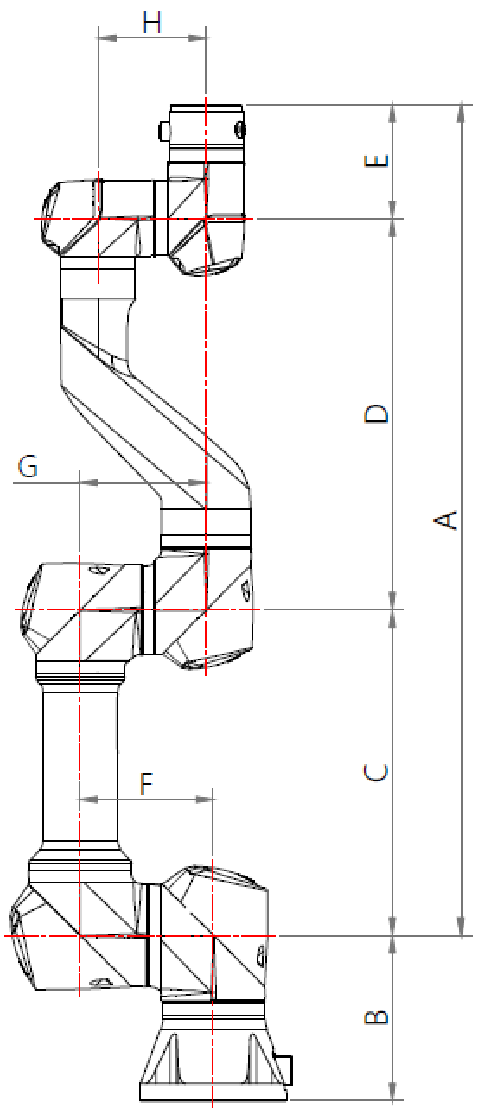
1.3 ROBOT ARM

■ Names of Each Part



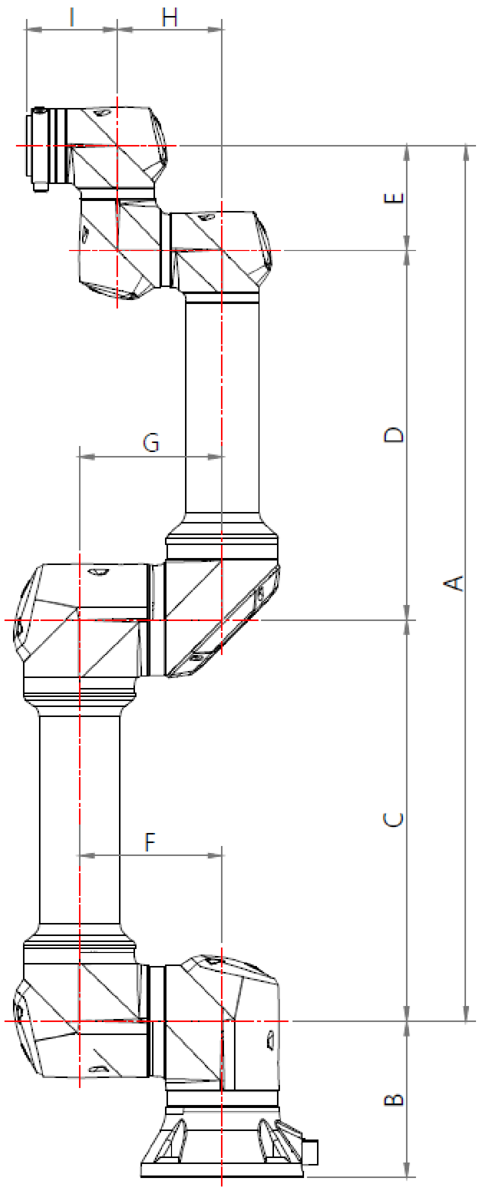
No.	Name	Description
①	Base	Part for securing the robot
②	J0 : Base Joint	
③	J1 : Shoulder Joint	
④	J2 : Elbow Joint	
⑤	J3 : Wrist 1 Joint	
⑥	J4 : Wrist 2 Joint	
⑦	J5 : Wrist 3 Joint	
⑧	Tool Flange	Part for attaching a gripper or tool to the robot
⑨	Teaching Button	Button for direct teaching
⑩, ⑪	I/O Connector A(⑩), B(⑪) (Non E, E-Version : ⑩ / U-Version : ⑩,⑪)	Input/output ports for controlling the gripper or tool
⑫	Robot-Control Box Connector	Cable connection connector between the robot arm and control box

■ Robot Arm Shape and Dimension Information



(Unit : mm)

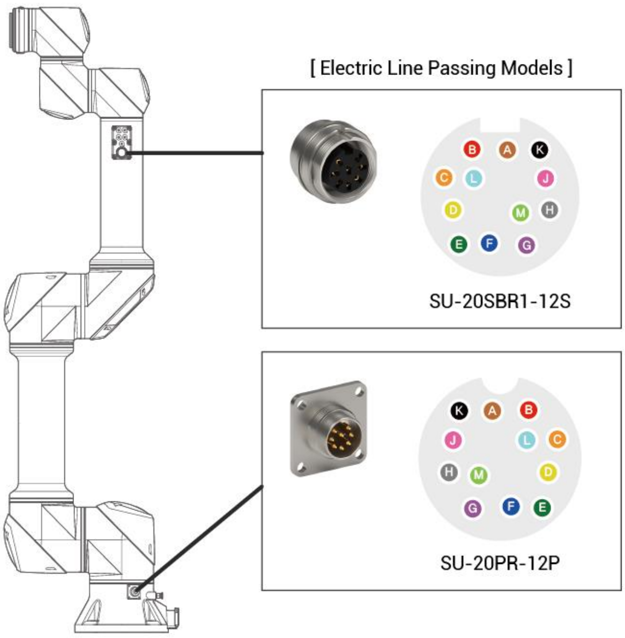
Model	A	B	C	D	E	F	G	H
RB3-730ES Series	730	145.3	286	344	100	117.15	110.7	94.6
RB6-920ES Series	920	165.5	400	423.3	96.7	151.4	129.2	110.7
RB20-1900ES Series	1900	241	885	885	130	254.5	187.5	129.2



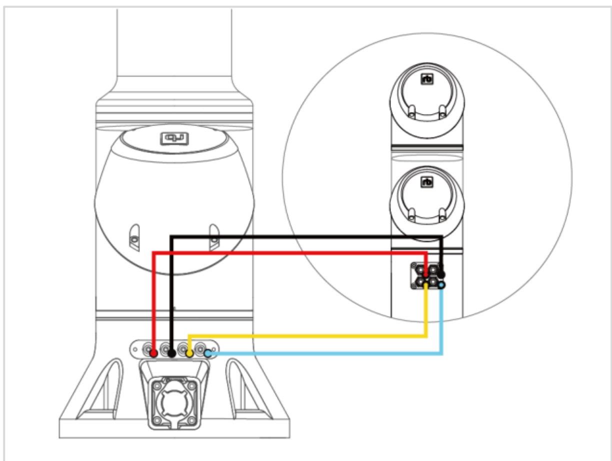
(Unit : mm)

Model	A	B	C	D	E	F	G	H	I
RB3-1200E Series	1200	165.5	566.9	522.4	110.7	151.4	151.4	110.7	96.7
RB5-850E Series	927.7	165.5	425	392	110.7	151.4	151.4	110.7	96.7
RB10-1300E Series	1300	197	612.7	570.15	117.15	187.5	151.4	117.15	115.3
RB16-900E Series	900	197	412.7	370.15	117.15	187.5	151.4	117.15	115.3

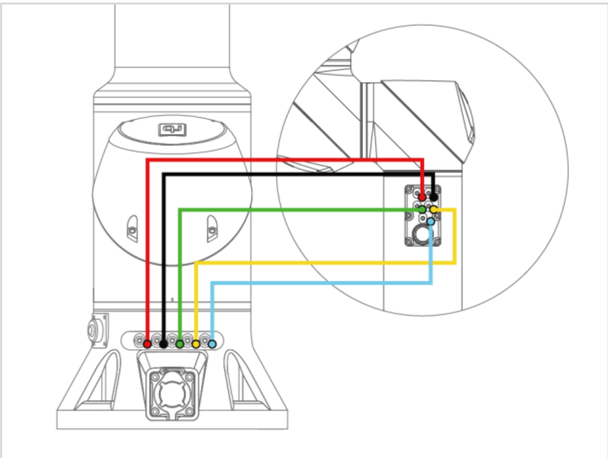
■ Usage of RB5-850EA#, RB3-1200EA#, RB10-1300EA#, RB16-900EA#, RB6-920ESA#, RB20-1900ESA# (Pneumatic/Electric Cable Built-in Models)



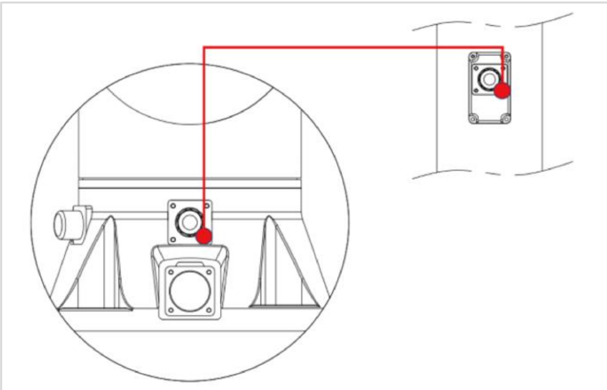
[Internal Pneumatic Tube Connection]



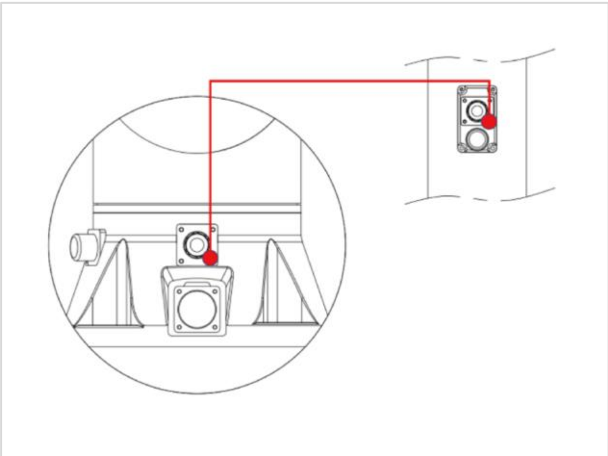
[RB5-850EA1, RB3-1200EA1, RB10-1300EA3, RB16-900EA1, RB6-920ESA1, RB20-1900ESA1]



[RB5-850EA2, RB3-1200EA2]



[RB10-1300EA1, RB16-900EA2, RB20-1900ESA2]



[RB10-1300EA2]

※ The pneumatic and electric lines of models RB5-850EA#, RB3-1200EA#, RB10-1300EA#, RB16-900EA#, RB6-920ESA#, and RB20-1900ESA# are provided as shown in the table below. Please refer to the diagram for usage.

Model	Pneumatic Line	Electric Line
RB5-850EA1	Up to 4 (4ø pneumatic tubes)	None
RB5-850EA2	Up to 5 (4ø pneumatic tubes)	12-pin (AWG28)
RB3-1200EA1	Up to 4 (4ø pneumatic tubes)	None
RB3-1200EA2	Up to 5 (4ø pneumatic tubes)	12-pin (AWG28)
RB10-1300EA1	1 (8ø pneumatic tubes)	None
RB10-1300EA2	1 (8ø pneumatic tubes)	12-pin (AWG28)
RB10-1300EA3	Up to 4 (4ø pneumatic tubes)	None
RB16-900EA1	Up to 4 (4ø pneumatic tubes)	None
RB16-900EA2	1 (8ø pneumatic tubes)	None
RB6-920ESA1	Up to 4 (4ø pneumatic tubes)	None
RB20-1900ESA1	Up to 4 (4ø pneumatic tubes)	None
RB20-1900ESA2	1 (8ø pneumatic tubes)	None

※ The number of pneumatic lines should be adjusted after checking the operating range.

WARNING

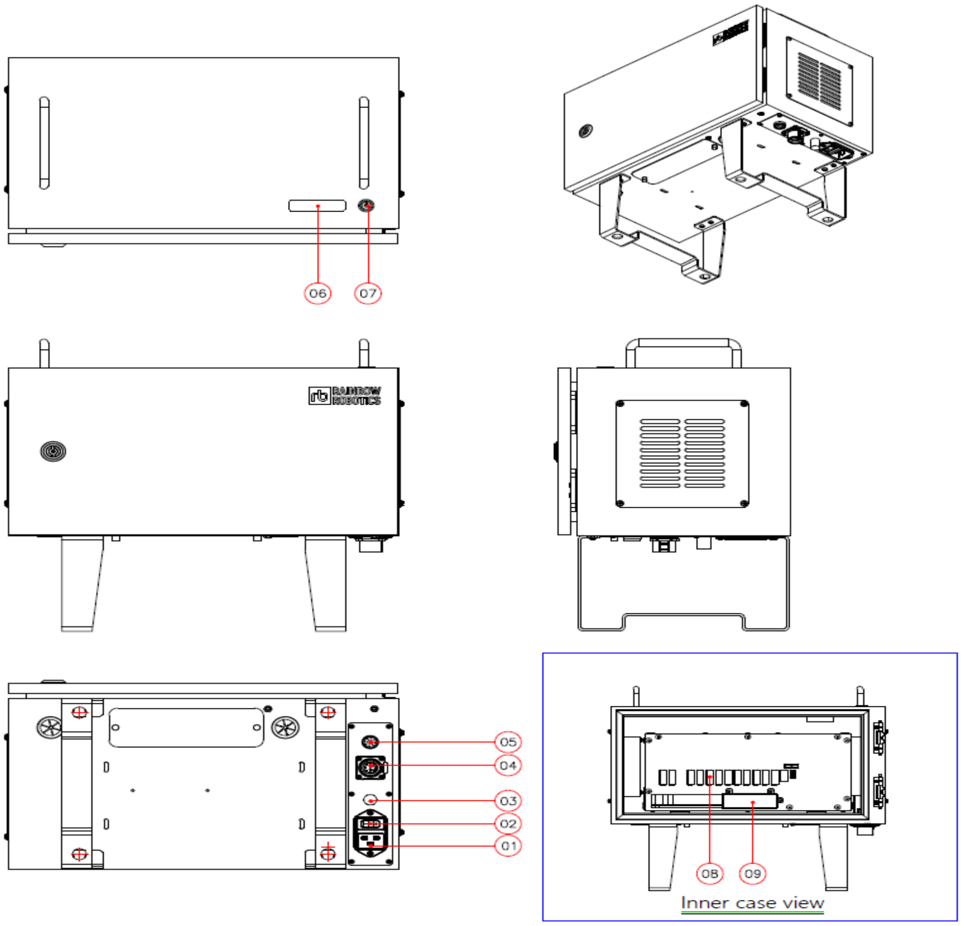
Warning:

1) For pneumatic/electric cable built-in models, passing pneumatic pressure or power beyond the defined specifications may damage the hardware.

1.4 ROBOT CONTROL BOX

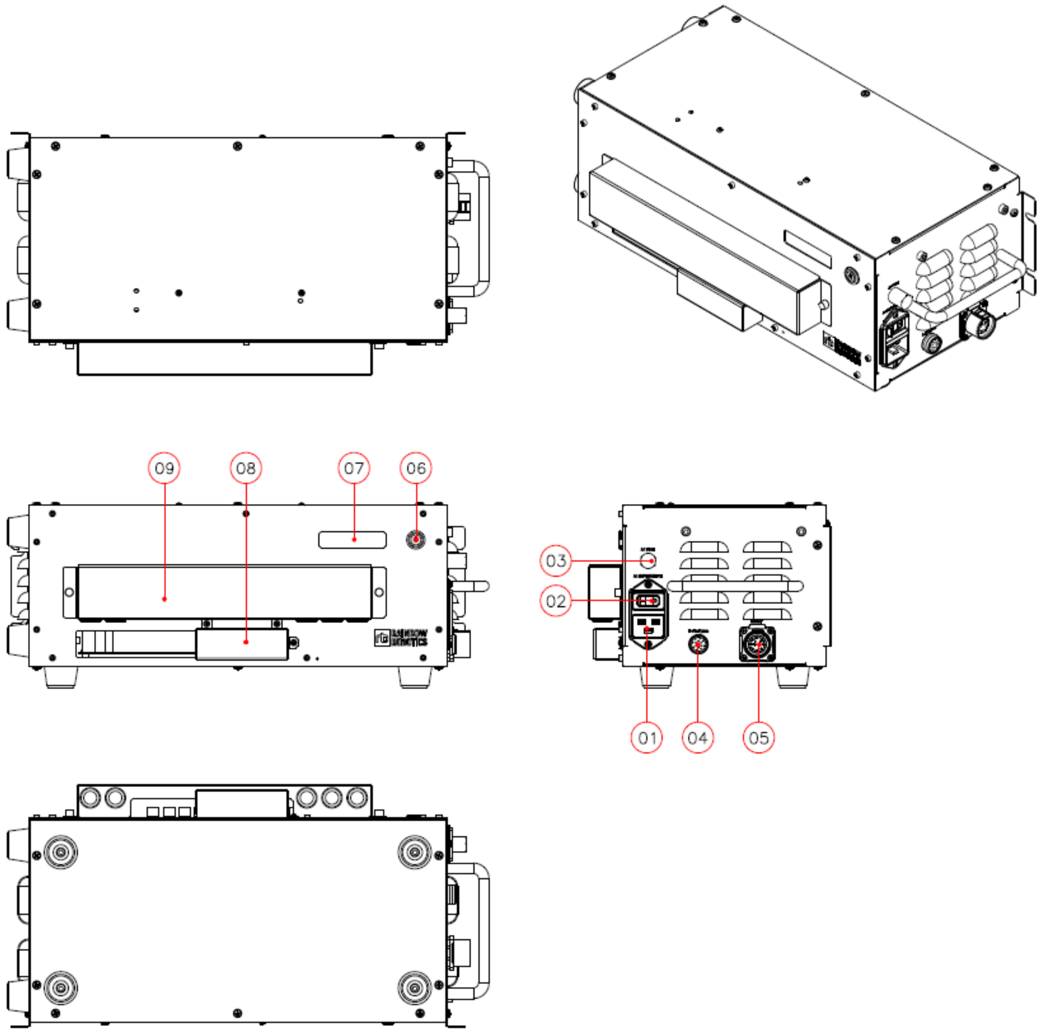
Front, Bottom, and Interior of the Control Box.

- Stand Control Box (CB06, CB06-1)



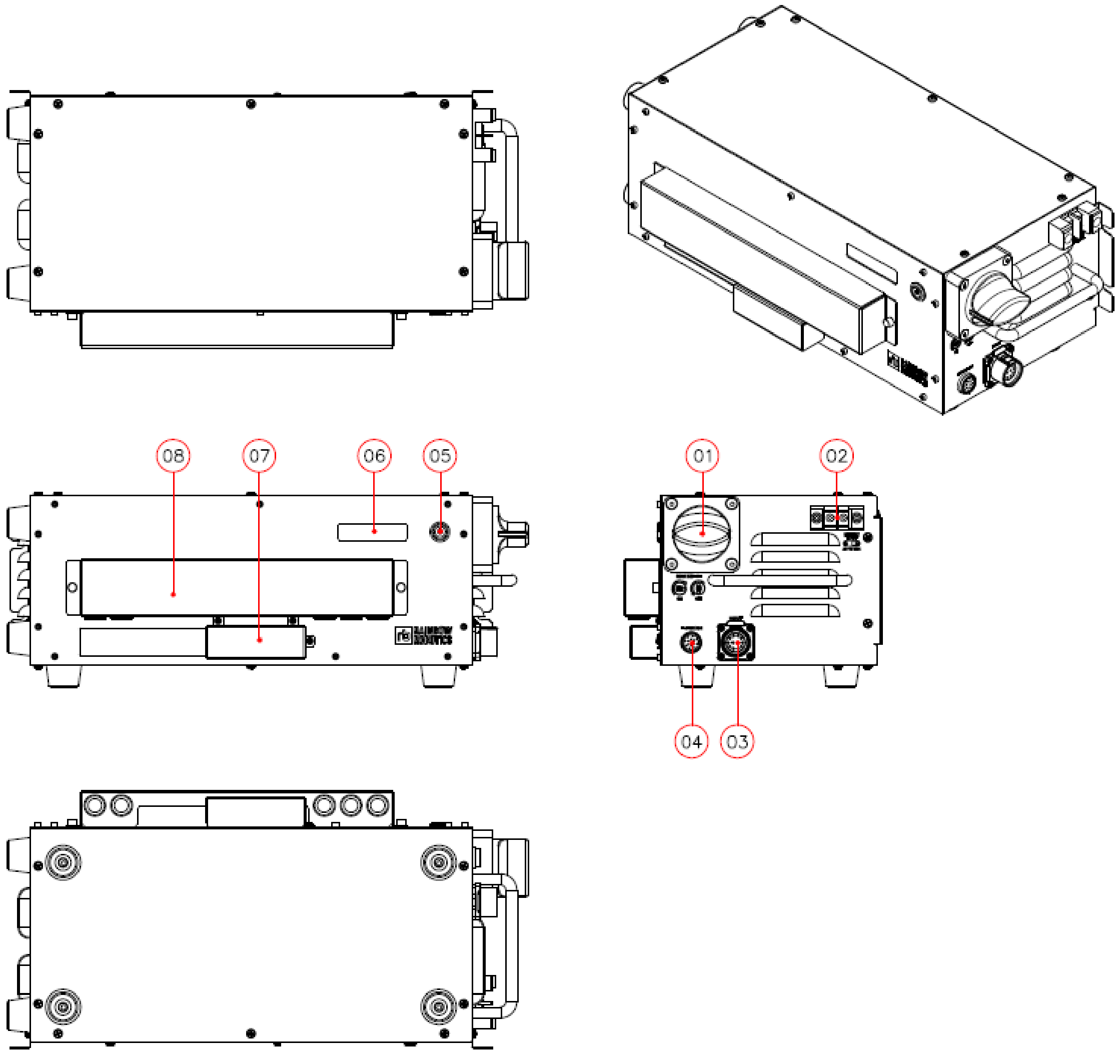
No.	Name
①	AC Input Power Connector
②	AC Main Power Switch
③	AC Fuse Holder
④	Robot Arm Connector (10-pin)
⑤	E-STOP/JOG Connector
⑥	LCD
⑦	Boot-up switch
⑧	I/O port
⑨	USB/LAN connector

■ Compact Control Box (CB07)



No.	Name
①	AC Input power connector
②	AC Main power switch
③	AC fuse holder
④	E-STOP/JOG connector
⑤	Robot Arm Connector (10-pin)
⑥	Boot-up switch
⑦	LCD
⑧	USB/LAN connector
⑨	I/O port

■ DC Control Box (CB08)

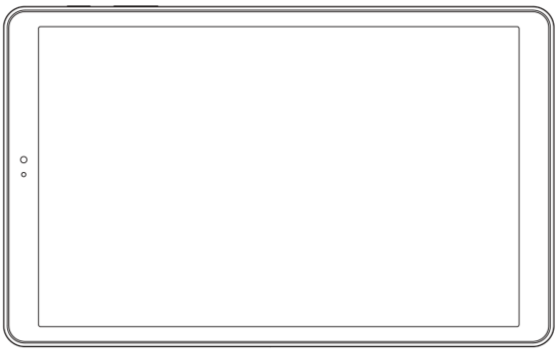


No.	Name
①	DC Main Power Switch (Miniature Circuit Breakers)
②	DC Power Input
③	Robot Arm Connector (12-pin)
④	E-STOP/JOG connector
⑤	Boot-up switch
⑥	LCD
⑦	USB/LAN connector
⑧	I/O port

1.5 TABLET PC (OPTIONAL)

The teaching pendant/tablet PC is an optional accessory.

- For Control Box

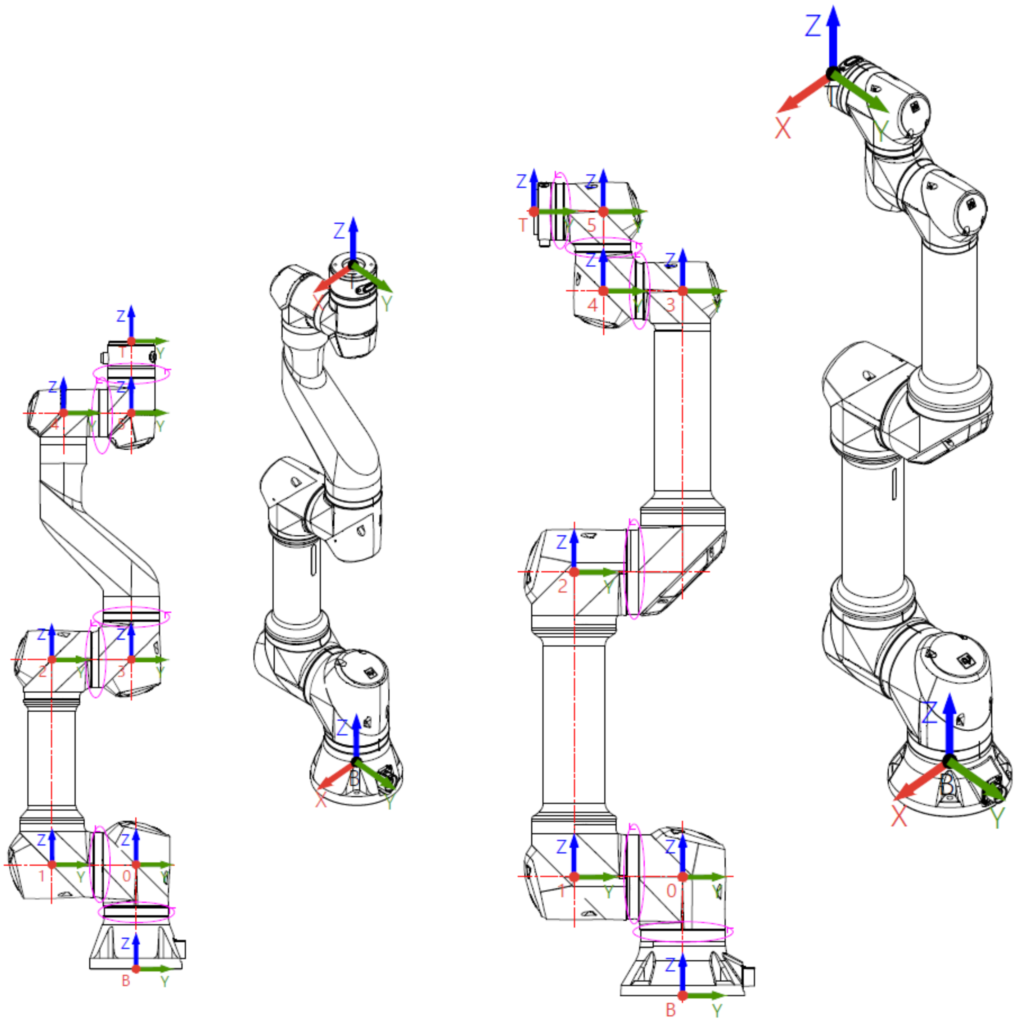


※ If you do not select the tablet PC option, we provide a dedicated application for operating, controlling, and teaching the robot arm. You can download it from our company's website.

※ Tablet setup is required for use with RB products. See the Appendix.

1.6 ROBOT OPERATING RANGE

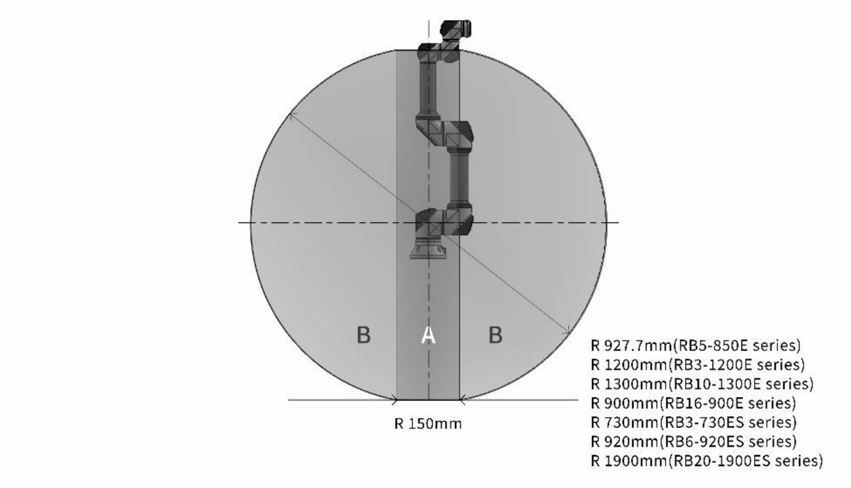
An RB robot consists of six joints. The axes of rotation and joint limits are illustrated in the following section.



Joint	Range
J 1	± 360 °
J 2	± 360 °
J 3	± 165 ° / ± 150 °
J 4	± 360 °
J 5	± 360 °
J 6	± 360 °

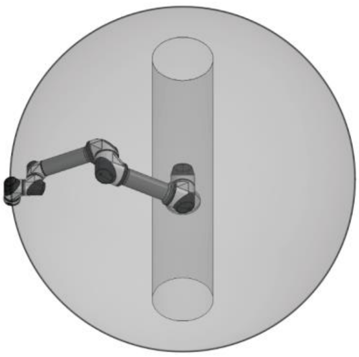
1.7 ROBOT WORKSPACE

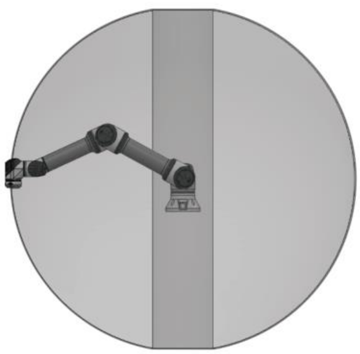
The maximum reach radius when the robot arm is fully extended is as follows:
RB5-850E Series: 927.7mm, RB10-1300E Series: 1300mm, RB3-1200E Series: 1200mm, RB16-900E Series: 900mm, RB3-730ES Series: 730mm, RB6-920ES Series: 920mm, RB20-1900ES Series: 1900mm. The area directly below or above the robot base (the A area and its surroundings in the diagram below) is a restricted work area. There is no issue when using joint coordinate system movements (e.g., Move J), but in movements using the Cartesian coordinate system (e.g., Move L), passing through this area is restricted. This area is a mechanical singularity point, and using the Cartesian coordinate system in this region can generate rapid joint speeds or cause the robot to stop.



A. Singularity point area (a restricted area for movement when using Cartesian coordinates)

B. Robot's operable work area


Tilt View

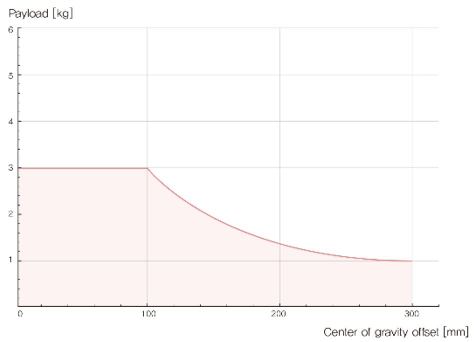

Front View

1.8 ROBOT ARM MAXIMUM LOAD CAPACITY

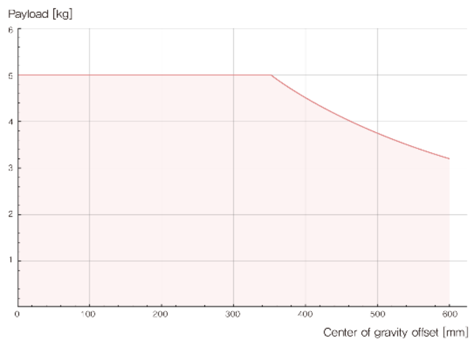
The payload of the robot arm varies depending on the distance between the tool flange and the center of gravity of the payload. The payload capacity by distance is as follows:

※ 상기 내용은 추후 변경될 수 있습니다.

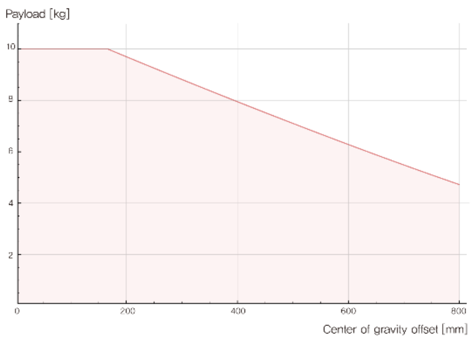
RB3-1200E Series



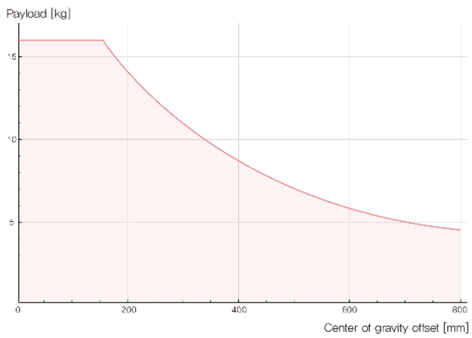
RB5-850E Series



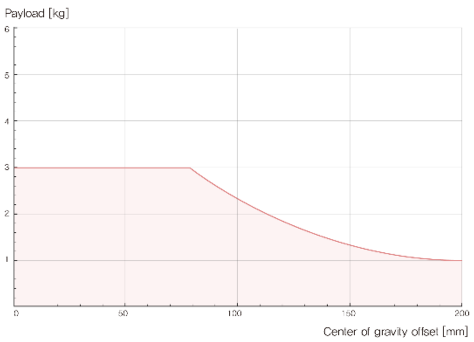
RB10-1300E Series



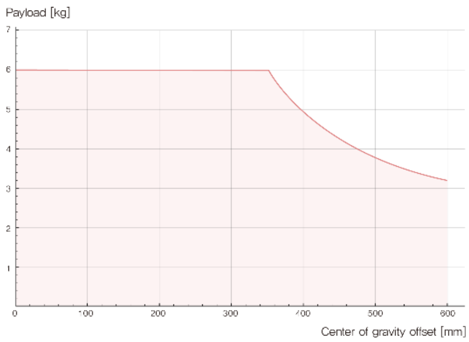
RB16-900E Series

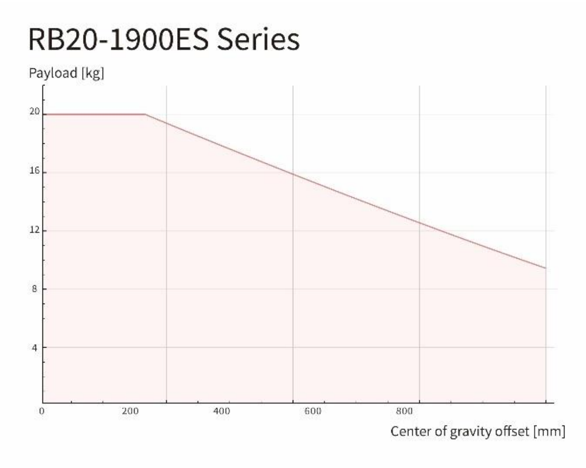


RB3-730ES Series



RB6-920ES Series




RB20-1900ES Series
Payload [kg]
20
16
12
8
4
0
200
400
600
800
Center of gravity offset [mm]

CHAPTER 2. SAFETY & PRECAUTIONS

2.1 SAFETY INDICATIONS

The following safety warnings are used in this manual.

DANGER

Danger:

Failure to follow instructions marked with this symbol may result in severe harm, which could result in serious injury or death.

WARNING

Warning:

Failure to follow the instructions with this symbol may result in an accident, which could result in serious injury to the user.

CAUTION

Caution:

Failure to follow directions marked with this symbol may result in damage to the product or injury to users.

2.2 GENERAL SAFETY WARNING & PRECAUTIONS

This section contains general hazards, warnings, and cautions that will be repeated or further explained elsewhere in this manual.

DANGER

Danger:

1) Robots and electrical equipment must be installed according to the instructions from **Chapter 4. Installation.**

WARNING

Warning:

1) Robot users and robot application system manufacturers should be familiar with the contents of this manual. In addition, they should complete operational training.
2) Make sure enough space is provided for the robot arm to move freely.
3) When using the robot, do not wear loose clothes or jewelry. Long hair should be tied so that it does not get caught in the joints of the robot.
4) Never operate a broken or faulty robot.
5) If a fatal error occurs in the software, immediately hit the emergency switch to stop the robot, and then contact your supplier or Rainbow Robotics.
6) Check that the robot installation angle, tool setting, safety setting, etc. are entered correctly.
7) Do not connect safety equipment to the general use I/O ports in the back of the control box. Safety equipment should only be used with safety-related I/O ports.
8) Watch out for the movement of the robot when using the teaching pendant.
9) During the operation of the robot, do not enter the operating range of the robot, and do not touch the robot while it is operating.

10) Never modify the robot without the support of Rainbow Robotics. Rainbow Robotics (hereinafter "the manufacturer") bears no responsibility/liability for any problems caused by user's modification or modification of the product.
11) Both the robot arm and the control box generate heat when used for a long time. Do not touch the robot after long use. If the user needs to touch the robot, make sure to turn off the controller and allow the robot to cool down before touching.
12) When the robot collides with an external object, a considerable amount of kinetic energy is generated. This kinetic energy is proportional to the speed of the robot and the payload.
13) Confirm that you are using the recommended installation settings for the robot. The teaching or collision detection functions may not work properly if the robot arm's mounting configuration, tool weight, tool center of gravity, length, safety configuration, etc. are not entered correctly.
14) The teaching function should only be used in a safe environment. Do not use this function when there are hazards nearby.
15) Before using the teaching function, input the relevant information (tool length, weight, center of gravity, etc.) accurately. Not entering the accurate relevant specifications will cause malfunctions when using the direct teaching function.
16) If the robot joints move at an unsafe speed when using the direct teaching function, the user can force the robot to stop with the emergency switch for their safety.
17) Robotic arm and control box generate heat during operation. Do not touch the robot arm during operation or immediately after operation as continuous contact with the robot arm may cause it to malfunction. Before manipulating or touching the robot arm, make sure to check the temperature reading on the UI screen or turn off the robot arm. Please wait at least 1 hour to cool it down before touching.

CAUTION

Caution:

1) When using with a machine or another robot that can damage the robot arm, it is recommended to test all functions separately before use. The manufacturer is not liable for any programming errors, damage to the RB, or damage to other machines due to robot malfunctions.
2) Do not expose the robot to strong magnetic fields as the robot may be damaged.

Warning:

1) Attach a warning label to the spot where there is a danger of electric shock from the electric device.
2) Do not tear, damage, or remove the cover. Be careful when handling parts or devices with a label attached, as well as surrounding components.
3) To avoid electric shock, do not touch the internal electric parts.

2.3 USAGE & FUNCTIONALITY

The robot arm is intended to be used for transferring and assembling objects by utilizing tools and should only be operated in the environment specified in the description. It is possible to operate it without a physical protective barrier. However, a safety mechanism should be used after performing the risk evaluation for the whole system. The use of the robot in any of the following applications and environments is considered improper use, and the manufacturer is not liable for any direct or indirect damage to the robot or accidental, consequential damage.

- Use in any potentially explosive environment
- Medical and life-related uses
- Human and animal transport
- Any use without risk assessment
- Any use in places where the performance of the safety function is insufficient
- Any use beyond performance / environmental specifications

※ Improper use is not limited to the above items.

2.4 POTENTIAL SAFETY ISSUES

Additional protective measures must be taken if the final system is deemed unsafe or unable to adequately reduce risk. Users should consider the following potential risks:

- Injury (stenosis), which may occur when a finger is caught between the gears, etc.
- Injury (stabbing, penetrating) by sharp edges or edges of the tool
- Injury (stabbing, penetrating, falling) caused by objects located near the robot
- Injury that can occur when working with toxic and harmful substances (skin damage, dyspnea)
- Injury caused by collision with the robot (stump, fracture)
- Injury that may occur due to not fully fastening objects
- Injury from an object that has detached or dropped from the tool mount

※ Potential risks that may occur depending on the final system may be different.

2.5 LIABILITY LIMITATIONS

This manual does not cover all peripherals that affect safety. The system installer must comply with safety requirements in accordance with national safety regulations and the laws of the country where the robot arm is installed. The robotic arm consists of an end-coupled system of peripherals. This manual also does not cover all peripherals, including the design, installation, operation and safety of the final system. The final system to which the robot arm is applied must be designed and installed to meet safety requirements in accordance with the regulations and laws of the country where the system is installed.

The operator or the installer of the final system containing the robot arm is responsible for:

- Risk assessment of the final system
- Risk assessment of whether to add additional safeguards
- Ensuring that the system is properly designed, configured, and installed
- Definition of usage for the system
- Identification of important markings and contacts for use and safety
- Providing technical documents, such as manuals

※ Not limited to the above items. Complying with the safety instructions in this manual does not imply that you can avoid all risks that may occur.

2.6 SHIPPING & TRANSPORTATION

At least two people are required for transportation. Any damages to the robot incurred during shipment or transportation are excluded from the warranty.

WARNING

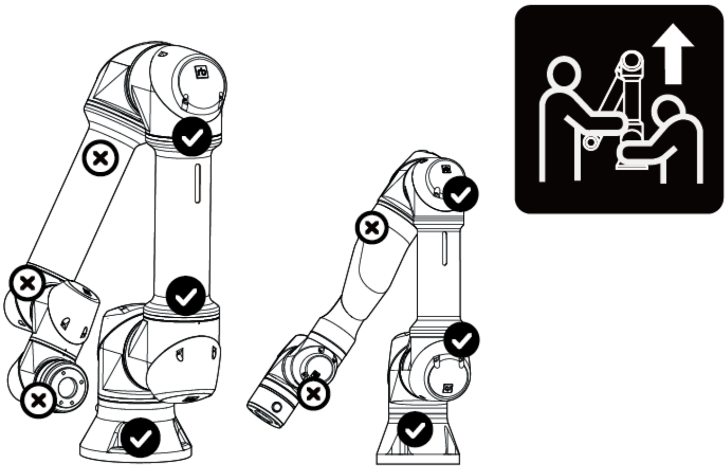
Warning:

1) Be careful not to damage the product during transportation. Damages incurred during transportation will void the warranty.
2) When transporting the robot arm, strong vibration or shock may damage the system. The robot must be transported using the packaging box provided by the manufacturer.

- **Precautions When Transporting**

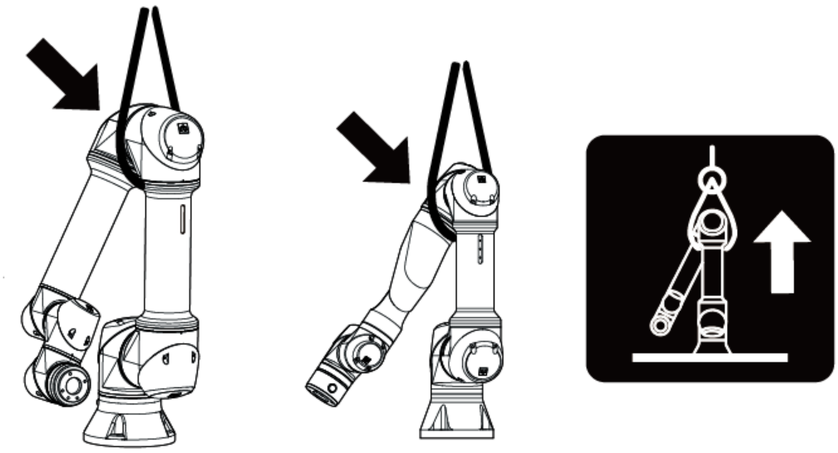
  1. When Transporting by Hand

  When lifting and moving the robot, hold the lower part of the robot, near the base axis. If this is not followed, critical issues may occur with the robot, and the manufacturer will not be responsible for any resulting damage.



2. When Transporting with a Wire Rope

Use a strong wire rope that can withstand the weight of the robot. When securing the rope to the robot, attach it to the areas indicated in the diagram below before transporting.



※ When using lifting equipment to transport the robot, please adhere to the weight and transport standards for the region or country. The manufacturer is not responsible for any damage caused during the transportation of equipment.

2.7 EMERGENCY STOP

The emergency stop button can be used to forcefully stop the robot arm if there is an emergency. By pressing the emergency stop button, the user will stop commands sent from the robot control box and terminate any motion.

The section below describes how the emergency stop button for control box works.

- Emergency Stop

  Users can stop the robot arm immediately by pressing the EMERGENCY STOP button.

- Re-Activating from Emergency Stop

  Turning the EMERGENCY STOP button in clockwise direction.

2.8 USER SAFETY

For the user's safety, please note the following:

- Powerless robot operation

In case of an emergency, or in any situation without power, the user can move the robot arm by forcing the joints into a different configuration (forced back driving). To perform forced back driving, the user must push or pull the robot arm firmly. Please ensure that the robot is not powered-on while performing forced back driving.

CAUTION

Caution:

1) If excessive force is applied to the joints in the non-powered state, please be aware that the driving part may be overloaded. The manufacturer is not responsible for any damage caused by excessive force.

2.9 SAFETY CONTROLLER

The Safety Control System for the Rainbow Robotics robot models listed below meets ISO 13849-1 Cat3. PLd standards:

RB5-850E Series, RB3-1200E Series, RB10-1300E Series, RB16-900E Series, RB3-730ES Series, RB6-920ES Series, RB20-1900ES Series

2.10 RISK ASSESSMENT

Risk assessment is a crucial factor when configuring a system that includes a robotic arm. The safety of the robot installation depends on how the arm is integrated into the overall system, meaning a risk assessment cannot be done based on the robot alone. To conduct a risk assessment for the robot, users who configure systems that include the robot must follow the guidelines of ISO 12100 and ISO 10218-2 when installing and operating the robot. Additionally, refer to technical standard ISO/TS 15066.

The risk assessment must be conducted immediately after the robotic arm is installed. The main focus of the risk assessment is to determine the safety configuration suitable for the surrounding environment and robot settings, as well as the need for additional emergency stop buttons and protective measures.

The safety-related functions of the collaborative robot can be configured in the safety configuration menu. The menu provides the following functions.

- I/O settings: The control box can be set to output safety information through the output terminal.
- Speed control: Allows the user to control the moving speed of the robot arm.
- Collision Detection Sensitivity Adjustment: When the robot collides with a nearby object, it will stop. However, the user can control the sensitivity at which the arm detects the collision.

If the risks are not sufficiently reduced or eliminated by the above safety functions, additional protective measures should be installed to eliminate risks.

The manufacturer is not responsible for accidents that occur due to failure to follow international standards, domestic regulations, or failure to review the above risk assessment during installation.

CHAPTER 3. SAFETY FUNCTIONS

3.1 INTRODUCTION

This chapter contains important safety information, which must be read and understood by the integrator of the RB Series collaborative robots before the robot is powered on for the first time. RB Series can protect users and devices by providing various safety functions and safety device interfaces. Safety functions and interfaces meet Category 3, Performance Level d (PL d) as described in ISO 13849-1 and Hardware Fault Tolerance 1, Safety Integrity Level 2 (SIL 2) as described in IEC 62061.

CAUTION

Caution:

1) Depending on the case of the robot installation, the system integrator must perform a risk assessment, and accordingly, the workspace must be configured using safety monitoring functions and interfaces.
2) If a fault is found in the robot's safety function or interface, Stop Category 0 is initiated.
3) Examples of defects include broken cables in the emergency stop circuit, incorrect wiring of additional safety devices, and non-overlapping wiring of additional safety devices (refer to Section 5 of this chapter).
4) System integrators and operators must be aware that there is a safety monitoring function that the robot performs internally, and the safety detection function is not only the operation of the emergency stop switch, the operation of the protective stop device, but also the position of the robot arm during task execution. The robot can be stopped in the designated stop mode even for movements above the physical limit, such as speed, momentum, etc. (See Section 3 of this chapter for the safety monitoring function).
5) System integrators and operators should consider the time and stopping distance between the robot stopping due to the operation of the error and safety monitoring functions described above. The system integrator must conduct a risk assessment considering the stopping distance and time (see Section 4 of this chapter).

6) System integrators and operators are aware of the fact that there is a safety monitoring function to limit the movement of the robot's joints and the robot/TCP, and must select the range of motion of the robot. TCP refers to the position to which the offset is added from the center point of the end of the robot arm.

DANGER

Danger:

1) The system integrator must conduct a risk assessment before applying power to the robot, and if it is used differently from that determined by the risk assessment or if different parameters are used, a risk that is not sufficiently reduced may occur.
2) When connecting additional safety devices, the power of both the robot and the control panel must be cut off.
3) When installing an additional safety device, measures must be taken to ensure that there is no problem when using it mechanically. For example, when using a light curtain, it must be firmly fixed to the floor/fixture, and movement and vibration must not occur during robot operation.
4) All safety function interfaces are set to 24V. Be careful when connecting devices with different voltages as it may cause equipment damage and fire.
5) The signal from the device mounted on the Tool Flange is not included in the safety function. Do not connect the safety device to the Tool Flange cable.

3.2 STOP CATEGORY

The safety function allows the robot to initiate three types of stop categories defined by IEC 60204-1.

Stop Category	Description
0 [STO]	Immediately, the robot is turned off and stopped. * Joint brake wear & tear may occur, which may shorten the lifespan of the robot. Do not use it unless it is unavoidable. ** Because the power of the robot is shut off, it is necessary to restart it when using it again after removing the danger.
1 [SS1]	All joints of the robot are decelerated to the maximum and stopped, and then the power is shut off to stop. * Since the power of the robot is shut off, it is necessary to restart it when using it again after removing the danger.
2 [SS2]	All the joints of the robot are decelerated to the maximum, stop, and then enter the SOS state. * SOS: Maintains the current position while the robot is powered on and activated, and starts Stop Category 0 when a position change is detected. ** Since the power is not shut off, it can be used immediately after removing the danger.

CAUTION

Caution:

1) According to ISO10218-1 5.5.2 and 5.5.3, a suitable stop category for emergency stop and protective stop should be selected.
2) For an emergency stop, you must select from stop category 0 or 1.
3) For an emergency stop, activation is required.
4) For a protection stop, at least one must be selected from stop categories 0 and 1.
5) For additional protection stops, stop category 2 can be used.

3.3 FUNCTIONAL SAFETY

The manufacturer recommends the following conditions are met for the installation location. The safety functions of the collaborative robot RB Series are used to reduce the risk of the robot system determined by risk assessment.

The parameters of the safety function are set at the factory, and the system integrator can change some items according to the risk assessment. Position and speed items are defined based on the base of the robot.

The following is the safety function specifications provided by RB Series.

	Safety Function	PL & Category
Safety stopping functions	SF.1 STO (Safe Torque Off)	PL d, Category 3
	SF.2 SS1 (Safe Stop 1)	PL d, Category 3
	SF.3 SS2 (Safe Stop 2)	PL d, Category 3
Safety monitoring functions	SF.4 SOS (Safe Operating Stop)	PL d, Category 3
	SF.5 SLP (Safely-Limited Position)	PL d, Category 3
	SF.6 SLS (Safely-Limited Speed)	PL d, Category 3
	SF.7 SLA (Safely-Limited Acceleration)	PL d, Category 3
	SF.8 SLI (Safely-Limited Increment)	PL d, Category 3
	SF.9 SLT (Safely-Limited Torque)	PL d, Category 3
	SF.10 RPL (Robot Position Limit)	PL d, Category 3
	SF.11 TSL (TCP Speed Limit)	PL d, Category 3
	SF.12 CBPL (Control Box Power Limit)	PL d, Category 3
Emergency stop	SF.13 EMS1 (Emergency Stop1)	PL d, Category 3
	SF.14 EMS2 (Emergency Stop2)	PL d, Category 3
Protective stop	SF.15 PRS (Protective Stop)	PL d, Category 3
	SF.16 HSS (Hard Safeguard Stop)	PL d, Category 3
	SF.17 SSS (Soft Safeguard Stop)	PL d, Category 3

- **STO(Safe Torque Off)**: This function prevents force-producing power from being provided to the motor. Power, that can cause rotation, is not applied to the motor. This safety sub-function corresponds to an uncontrolled stop in accordance with stop category 0 of IEC 60204-1.

- **SS1(Safe Stop 1)**: This function is specified as either a) SS1-d (Safe-Stop 1 deceleration controlled) initiates and controls the motor deceleration rate within selected limits to stop the motor and performs the STO function when the motor speed is below a specified limit; or b) SS1-r (Safe-Stop 1 ramp monitored) initiates and monitors the motor deceleration rate within selected limits to stop the motor and performs the STO function when the motor speed is below a specified limit; or c) SS1-t (Safe Stop 1 time controlled) initiates the motor deceleration and performs the STO function after an application specific time delay. This safety sub-function corresponds to a controlled stop in accordance with stop category 1 of IEC 60204-1. Above three candidates, our system uses SS1-t.

- **SS2(Safe Stop 2)**: This function is specified as either SS2-d (Safe Stop 2 deceleration controlled) initiates and controls the motor deceleration rate within selected limits to stop the motor and performs the safe operating stop function when the motor speed is below a specified limit; or b) SS2-r (Safe Stop 2 ramp monitored) initiates and monitors the motor deceleration rate within selected limits to stop the motor and performs the safe operating stop function when the motor speed is below a specified limit; or c) SS2-t (Safe Stop 2 time con-trolled) initiates the motor deceleration and performs the safe operating stop function after an application specific time delay. This safety sub-function SS2 corresponds to a controlled stop in accordance with stop category 2 of IEC 60204-1. Above three candidates, our system uses SS2-t.

- **SOS(Safe Operating Stop)**: This function prevents the motor from deviating more than a defined amount from the stopped position. The PDS (SR) provides energy to the motor to enable it to resist external forces. This description of an operational stop function is based on implementation by means of a PDS (SR) without external (for example mechanical) brakes.

- **SLP(Safely-Limited Position)**: This function prevents the motor shaft (or mover, when a linear motor is used) from exceeding the specified position limit(s).

- **SLS(Safely-Limited Speed)**: This function prevents the motor from exceeding the specified speed limit.

- **SLA(Safely-Limited Acceleration)**: This function prevents the motor from exceeding the specified acceleration and/or deceleration limit.

- **SLI(Safely-Limited Increment)**: This function prevents the motor shaft from exceeding the specified limit of position increment within specified time.

- **SLT(Safely-Limited Torque)**: This function prevents the motor from exceeding the specified torque (or force, when a linear motor is used) limit.

- **RPL(Robot Position Limit)**: This function prevents the robot arm's TCP (tool center point) or body frame exceeding the specified spatial region.

- **TSL(TCP Speed Limit)**: This function prevents the robot arm's TCP speed exceeding the specified speed.

- CBPL(Control Box Power Limit): This function prevents the Control Box's power consumption exceeding the specified limit.

- **EMS1(Emergency Stop1)**: This function activates the stop mode when the emergency stop switch of the Teaching Pendant Unit is activated. The stop mode is the SF.2.

- **EMS2(Emergency Stop2)**: This function activates the stop mode when the special I/O ports of the Control Box are activated. Those ports are provided for users to connect their own switch devices. The stop mode is the SF.2.

- **PRS(Protective Stop)**: This function activates the stop mode when the special I/O ports of the Control Box are activated. Those ports are provided for users to connect their own protective devices. The stop mode is the SF.2.

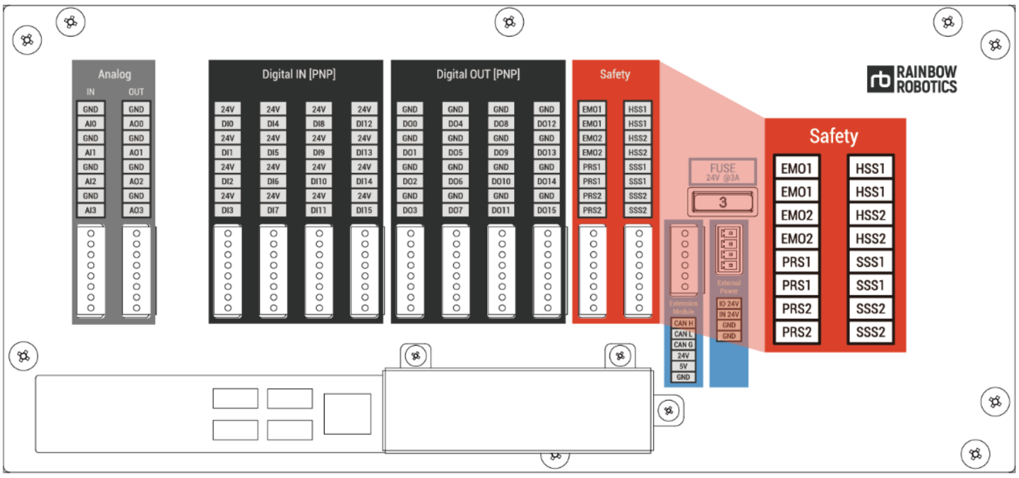
- **HSS(Hard Safeguard Stop)**: This function activates the stop mode when the special I/O ports of the Control Box are activated. Those ports are provided for users to connect their own protective devices. The stop mode is the SF.1.

- **SSS(Soft Safeguard Stop)**: This function activates the stop mode when the special I/O ports of the Control Box are activated. Those ports are provided for users to connect their own protective devices. The stop mode is SF.3.

3.4 SAFETY DEVICE MOUNTING LOCATION

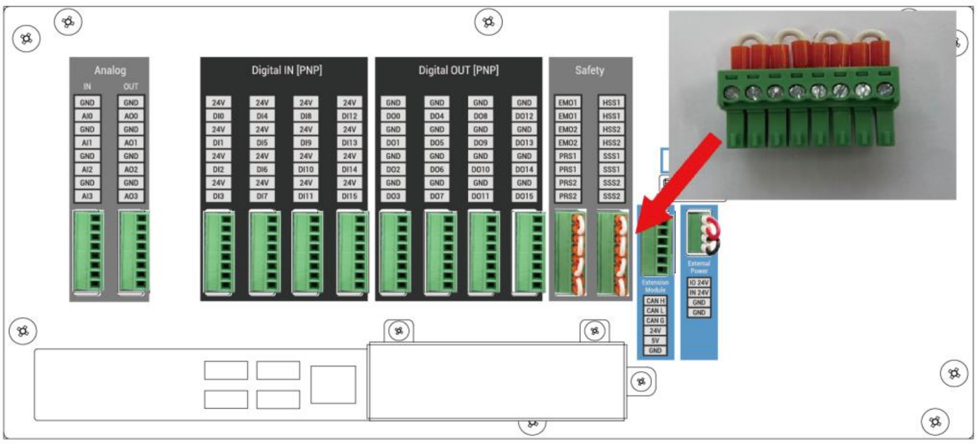
In addition to the basic emergency stop switch, the RB Series can be equipped with additional safety devices required by the system integrator through risk assessment.

The safety-dedicated contact terminal consists of 16 ports. This terminal is a redundant dedicated contact input terminal.



The additional ports can be equipped with 4 equipment. The robot is delivered with a default configuration, which enables operation without any additional safety equipment.

When using without connecting an external safety device, connect and use the basic contact input as shown below.



Safety device port specifications are as follows.

- **EMO**

This port is used when it is necessary to install an extra emergency stop switch through risk assessment.

The emergency stop switch should be used as a product conforming to IEC 60947-5-5.

Emergency stop generated through EMO is designated as stop category 1.

- **PRS**

This port is used to connect one or more protective stop devices through risk assessment.

Protective stop devices must be used in accordance with 5.3.8.3 of ISO 10218-2.

Protection stops occurring through PRS are designated as stop category 1.

- **HSS**

This port is used to connect one or more protective stop devices through risk assessment.

Protective stop devices must be used in accordance with 5.3.8.3 of ISO 10218-2.

Protection stops occurring through HSS are designated as stop category 0.

- **SSS**

This port is used to connect one or more protective stop devices through risk assessment.

Protective stop devices must be used in accordance with 5.3.8.3 of ISO 10218-2.

Protective stops that occur through SSS are designated as stop category 2.

3.5 EMERGENCY STOP SWITCH

The collaborative robot RB Series allows the operator to use the emergency stop switch to stop the robot in preparation for an emergency situation.

In the event of an emergency, you must press the emergency stop switch button to immediately stop the robot.

CAUTION

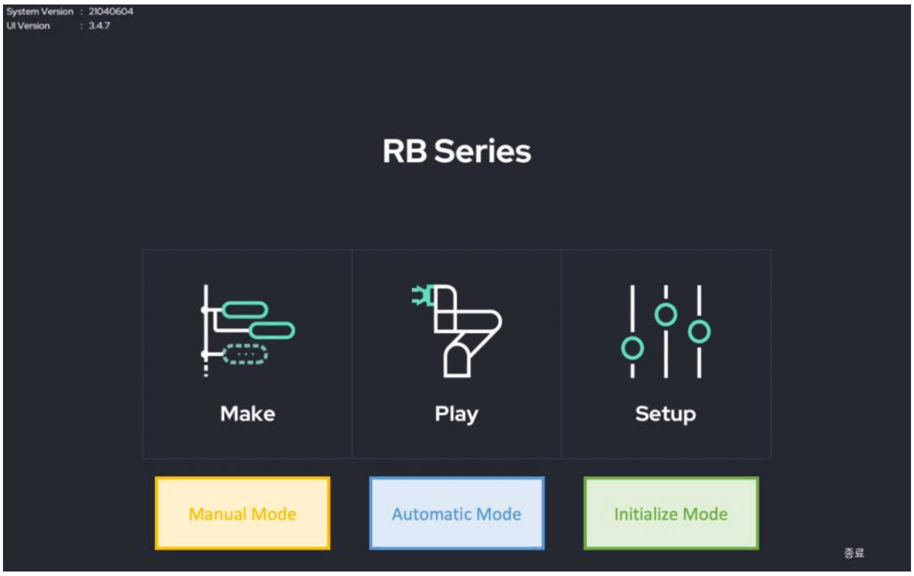
Caution:

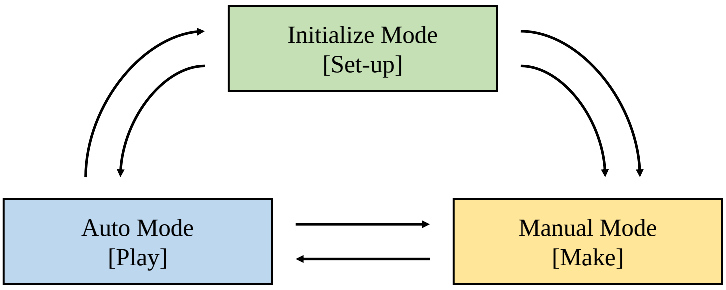
1) The emergency stop switch is designated as stop category 1.
2) You can cancel the emergency stop function by turning the emergency stop switch clockwise.
3) If you need an additional emergency stop switch, you can use it through the control panel.
4) Emergency stop should not be used as a risk reduction method, but should be used as a secondary protective device.

3.6 OPEATION MODE

The operation mode of the collaborative robot RB Series is composed as follows.

When entering the automatic mode, you must access it through a password.





CAUTION

Caution:

1) The password for entering automatic mode is not set at the time of shipment. Set up and use a password so that no one else can access it.
2) Before entering Auto Mode, the user must remove the dangerous situation and check the status of the emergency stop switch and the protective stop device.

3) In any case, the user must correctly grasp the installation state of the robot and complete the setting before operating the robot.

■ **Initialize Mode [Set-up]**

Peripheral device settings or robot status can be set before the robot moves.

Overall settings for robot motion such as workspace, TCP offset, and payload can be made. At this time, power is not supplied to the motor. The motor can be powered through the activation action.

■ **Auto Mode [Play]**

The robot is in a state where only predefined tasks are performed without user intervention. At this time, power is supplied to the motor.

The motion of the robot programmed through the simulation function can be verified through simulation, and the robot can be driven by the verified program by converting it to a real state.

At this time, the robot is performing pre-set safety functions, and the user can monitor the status of the robot and peripheral devices through the Play window.

■ **Manual Mode [Make]**

It is a state in which the robot is operated through direct actions of the user. At this time, power is supplied to the motor. Direct teaching, program creation and modification, and manual operation of peripheral devices can be performed, and the robot can only be operated at the moment the user operates the tablet through the safe speed slide bar.

When you release your hand from the safety slide, the robot will stop moving immediately.

CAUTION

Caution:

1) In case of manual operation, the safety slide function must be set.
2) At initial shipment, the safety slide function is deactivated.
3) In addition, when using a 3-position enabling device, it must be used in accordance with 5.8.3 of ISO 10218-1.

3.7 OPERATING ENVIRONMENT

In order to keep the robot in a safe state for a long time, it must be used in the following environment.

Maximum allowable operating temperature	50˚C
Maximum permissible storage temperature	60˚C
Minimum allowable operating temperature	0˚C
Minimum allowable storage temperature	-5˚C
Maximum permissible humidity	80%
Lowest permissible humidity	20%

3.8 MAINTENANCE OF SAFETY FUNCTIONS

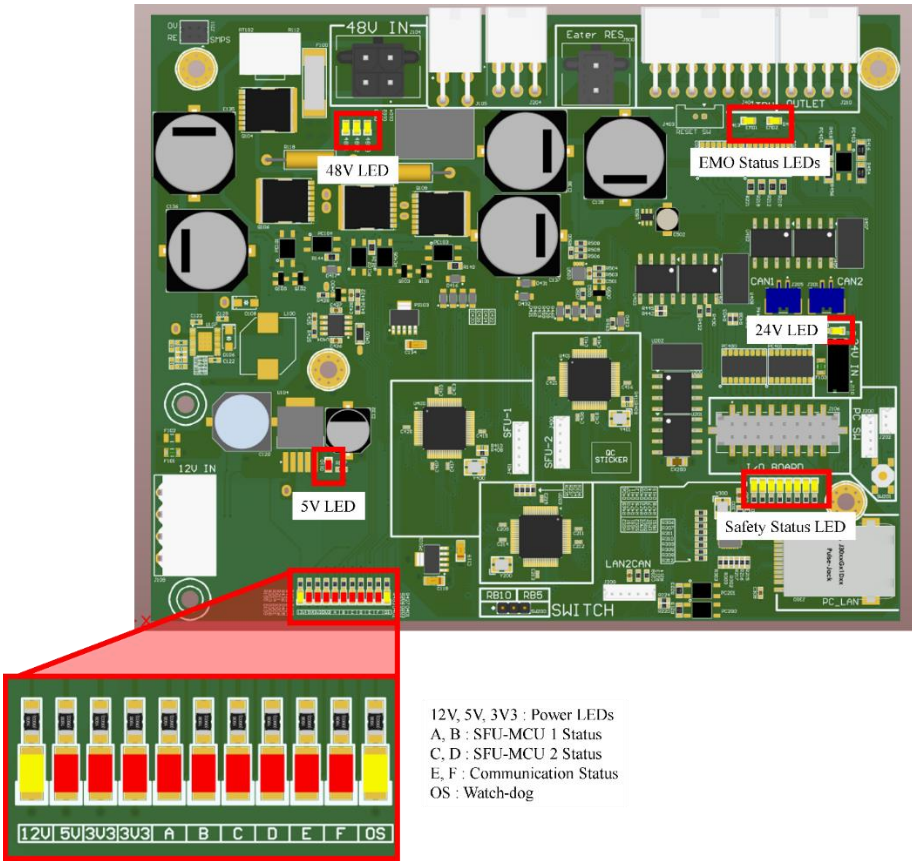
In order to keep the robot in a safe state for a long time, it is necessary to continuously check the safety functions.

Safety Inspection Managers need periodic inspections for the following items. If during the inspection you find a problem that cannot be solved by yourself, contact the manufacturer.

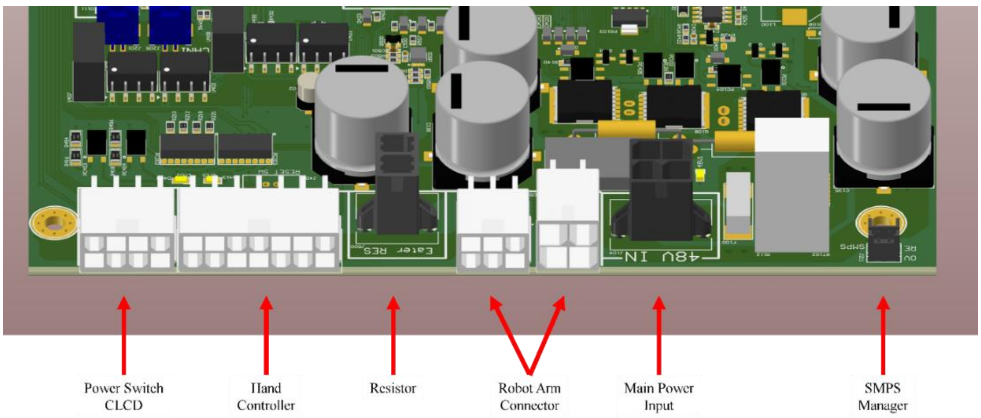
Inspection target		Check List	Period
Pendant	Safety Function	Check whether the emergency stop switch mounted on the pendant is working properly.	1 Month
	Cable	Check the condition of the connection cable between the pendant and the control box.	1 Month
Control Box	Interface	Check whether the EMO port to which the safety device is connected is working properly.	1 Month
		Check whether the PRS port to which the safety device is connected is working properly.	1 Month
		Check whether the HSS port to which the safety device is connected is working properly.	1 Month
		Check whether the SSS port to which the safety device is connected is working properly.	1 Month
	Power	Check the normal output of 24V voltage connecting the safety device.	1 Month
		Check if the 24V fuse is inserted normally.	1 Month
	Cable	Check the condition of the connection cable between the safety device and the control box.	1 Month

■ Safety Function Board Specification

Inside the control box, there is a built-in safety function board to drive the RB Series. The information of the LED indicating the operation status of the board is as follows.



Connector information connected to the board is as follows.



3.9 APPLIED STANDARDS

Standard	Title
IEC 61508-1:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 1: General requirements
IEC 61508-2:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 2: Requirements for electrical/electronic/programmable electronic safetyrelated systems
IEC 61508-3:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 3: Software requirements
IEC 61508-4:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 4: Definitions and abbreviations
IEC 61508-5:2010	Functional safety of electrical/electronic/programmable electronic safety-related system – Part 5: Examples of methods for the determination of safety integrity levels
IEC 61508-6:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 6: Guidelines on the application of IEC 61508-2 and IEC 61508-3
IEC 61508-7:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 7: Overview of techniques and measures
IEC 60204-1:2016	Safety of machinery – Electrical equipment of machines – Part 1: General requirements
IEC 61000-6-1: 2016	Electromagnetic compatibility (EMC) – Part 6-1: Generic standards – Immunity standard for residential, commercial and light-industrial environments

IEC 61000-6-2: 2016	Electromagnetic compatibility (EMC) – Part 6-2: Generic standards – Immunity standard for industrial environments
IEC 61000-6-7: 2014	Electromagnetic compatibility (EMC) – Part 6-7: Generic standards – Immunity requirements for equipment intended to perform functions in a safety-related system (functional safety) in industrial locations
IEC 61326-3-1: 2017	Electrical equipment for measurement, control and laboratory use – EMC requirements – Part 3-1: Immunity requirements for safety-related systems and for equipment intended to perform safety-related functions (functional safety) – General industrial applications
IEC 61800-5-1: 2007	Adjustable speed electrical power drive systems – Part 5-1: Safety requirements –Electrical, thermal and energy
IEC 61800-5-2: 2016	Adjustable speed electrical power drive systems – Part 5-2: Safety requirements – Functional
IEC 62061:2005	Safety of machinery – Functional safety of safety-related electrical, electronic and programmable electronic control systems
ISO/TS 15066: 2016	Robots and robotic devices — Collaborative robots
ISO 10218-1: 2011	Robots and robotic devices — Safety requirements for industrial robots — Part 1: Robots
ISO 10218-2: 2011	Robots and robotic devices — Safety requirements for industrial robots — Part 2: Robot systems and integration
ISO 12100:2010	Safety of machinery — General principles for design — Risk assessment and risk reduction
ISO 13849-1: 2015	Safety of machinery — Safety-related parts of control systems — Part 1: General principles for design
ISO 13849-2: 2012	Safety of machinery — Safety-related parts of control systems — Part 2: Validation

CHAPTER 4. INSTALLATION

4.1 INTALLATION PRECAUTION

Robot installers must install and operate the robots in accordance with the guidelines of ISO 12100 and ISO 10218-2, and installers must comply with the relevant requirements of international standards such as ISO / TS 15066 and national laws.
The manufacturer is not responsible for any accidents caused by risks that do not comply with the relevant requirements provided by international standards, risks that do not comply with the relevant requirements provided by national laws and regulations, or those caused by failure to review the risk assessment in the previous chapter.

4.2 INSTALLATION LOCATION

The manufacturer recommends the following conditions are met for the installation location.

- Building with seismic design
- No leakage
- No flammable or explosive material
- Constant temperature and humidity
- Limited dust inflow

CAUTION

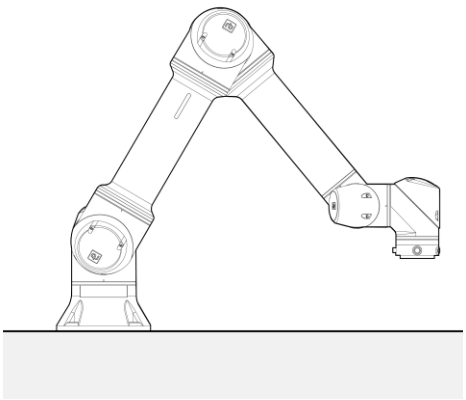
Caution:

1) If the system is not installed in a location that matches the recommendations, the performance and lifespan of the robot may be reduced.

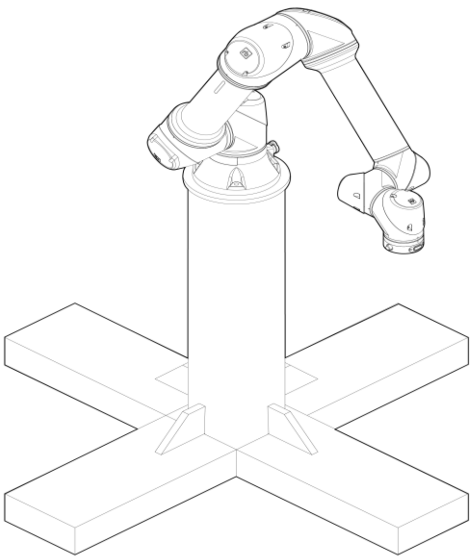
4.3 EXAMPLES OF INSTALLATION

The robot arm can be installed on the floor, wall, or any desired location, and it can also be mounted on a fixed post. The fixed post is not included with the product.

Additionally, it can be installed on walls, ceilings, or at any arbitrary angle. For installations other than flat surfaces, the installation angle must be entered through the system settings.



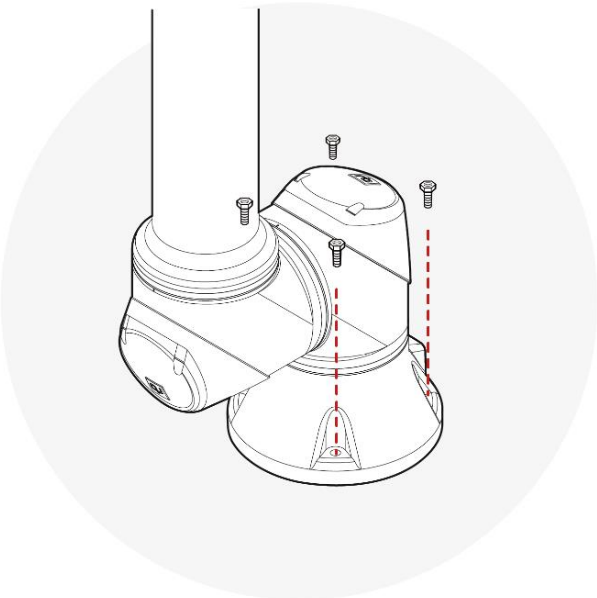
[Installation on the horizontal surface]



[Installation on the fixed post]

4.4 MOUNTING THE ROBOT

It is recommended to install the RB3-730ES Series robot arm using four M6 25mm bolts, while for other robot arms, it is recommended to use four M8 30mm bolts for installation.



WARNING

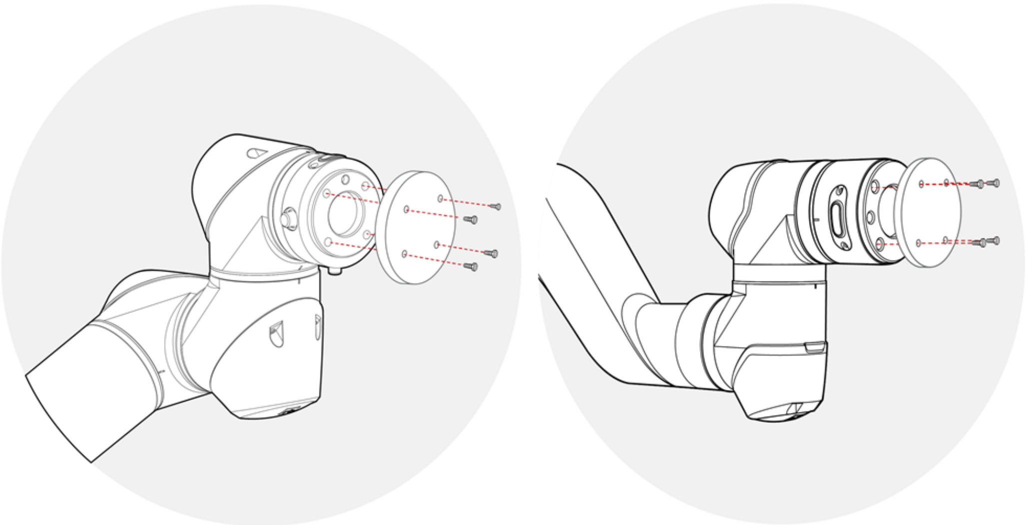
Warning:

1) When attaching the robot, fix it firmly so that the bolts do not come loose.
2) Install the robot on a sturdy surface that can withstand the combined weight of the robot and the load generated by the robot.
3) Please ensure that the mounting surface on the robot arm is completely in contact with the surface that it is mounted upon.
4) Never disassemble the bolts that are assembled in the robot. Ensure that all bolts are securely fastened before operating the robot arm.
5) If the bolts are not fastened properly, or if a bracket etc. is installed incorrectly, the product may become damaged, or the safety of the user may be seriously affected.

4.5 TOOL CONNECTION

Secure the tool to the tool flange using four M6 bolts(Standard: ISO 9409-1-50-4-M6).

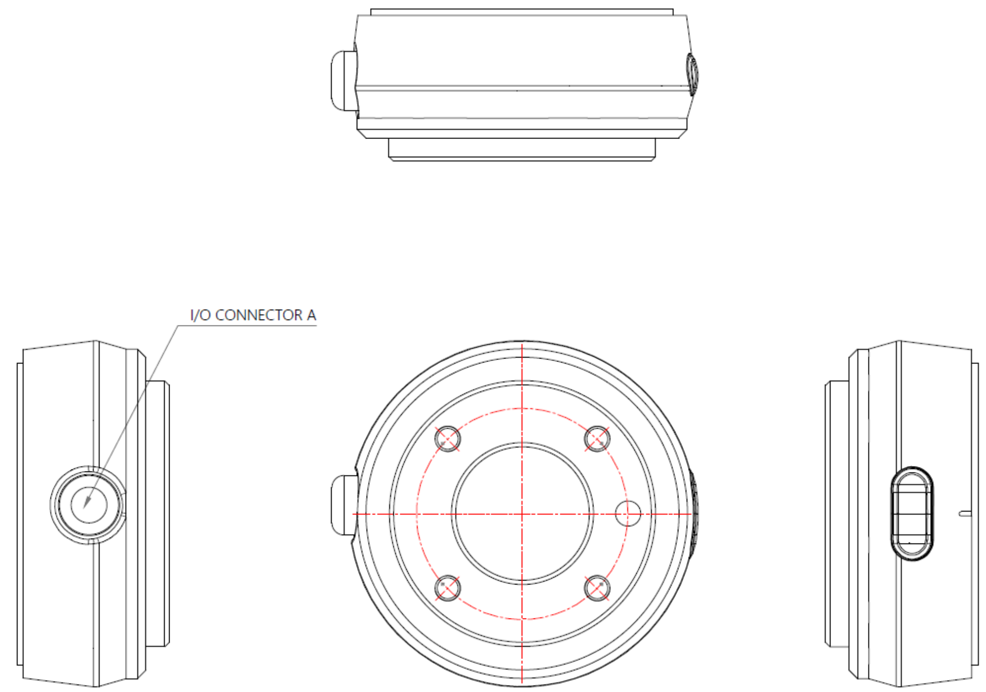
- Tools and M6 bolts are not included in the product.
- The connection methods may be different between tools. Please contact the tool manufacturer for further details.



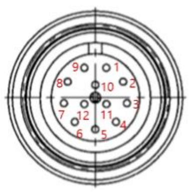
- After fixing the tool to the tool flange, connect the necessary cables to the I/O ports on either the tool I/O or the control box I/O.

- The specifications of the I/O connector located in the tool flange vary depending on the robot model.

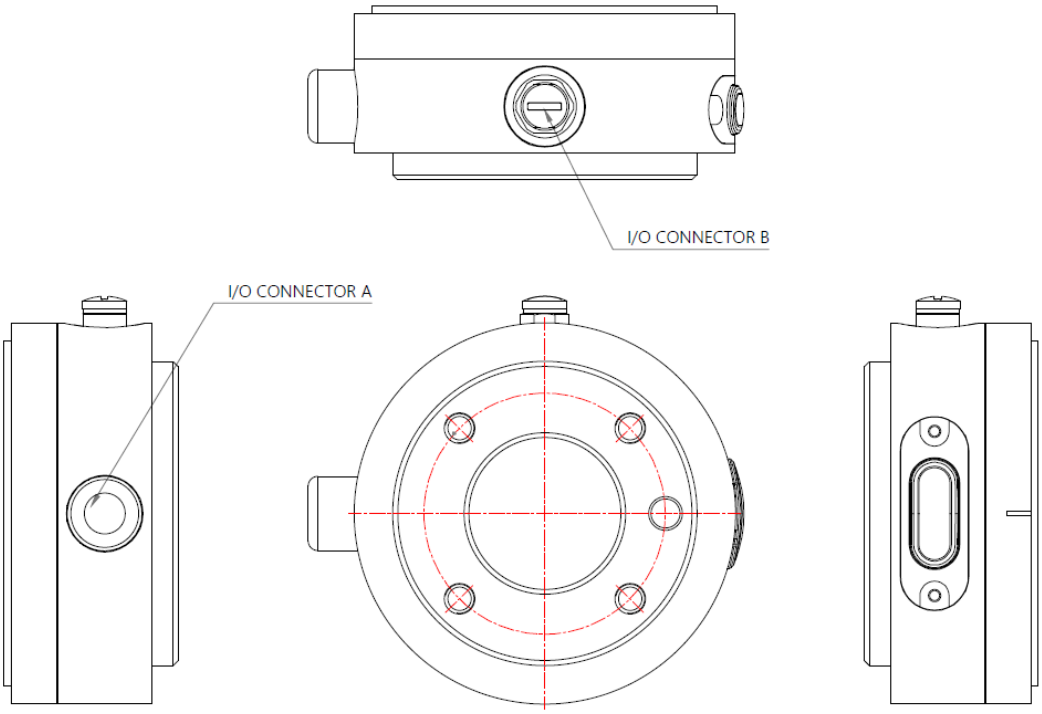
1) Non-E, E Version



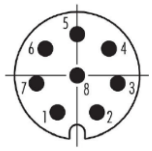
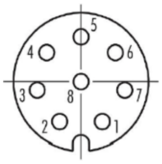
– I/O connector A : Sam Woo Electronics/M12 x P0.5 female circular connector, 12pin/SW-10W-12(P)

Port	Layout (Robot Side)	Pin No.	Signal (Non-E Version)	Color (Thickness)	Signal (E Version)	Color (Thickness)
Tool I/O		1	Digital Output A	Brown (AWG22)	Digital Output A	White (AWG25)
		2	Digital Output B	Blue (AWG22)	Digital Output B	Black (AWG25)
		3	0/12/24 VCC	Red (AWG22)	0/12/24 VCC	Red (AWG25)
		4	Ground	Black (AWG22)	Ground	Green (AWG25)
		5	Digital Input A	White (AWG26)	Digital Input A	Yellow (AWG25)
		6	Digital Input B	Blue (AWG26)	Digital Input B	Brown (AWG25)
		7	Analog Input A	Yellow (AWG26)	Digital Input C	Blue (AWG25)
		8	Analog Input B	Red (AWG26)	Digital Input D	Gray (AWG25)
		9	RS485+	Gray (AWG26)	RS485+	Orange (AWG25)
		10	RS485-	Purple (AWG26)	RS485-	Purple (AWG25)
		11	Common Ground	Black (AWG26)	Digital Input E	Pink (AWG25)
		12	Common Ground	Green(AWG26)	Digital Input F	Natural Color (AWG25)

2) U Version



- I/O connector A : Binder / M8 male panel mount connector, 8pin / 76 6319 1111 00008-0200
- I/O connector B : Binder / M8 female panel mount connector, 8pin / 76 6618 1111 00008-0200

Port	Layout (Robot Side)	Pin No.	U Version A Port	U Version B Port	Color (A=B)
Tool I/O	 A: Robot=Male	1	D+(RS485+)	NC	White
		2	D-(RS485-)	NC	Brown
		3	Digital Input 1 (PNP)	Digital Input 3 (PNP)	Green
		4	Digital Input 0 (PNP)	Digital Input 2 (PNP)	Yellow
	 B: Robot=Female	5	Power(+)	Power(+)	Grey
		6	Digital Output 1 (PNP)	Digital Input 5 (PNP)	Pink
		7	Digital Output 0 (PNP)	Digital Input 4 (PNP)	Blue
		8	Ground (-)	Ground (-)	Red

- Non-E version, E Version, and U Version have different tool flange specifications.
- [Common Content] Internal power supply can be set to 0V, 12V, and 24V on the I/O tab of the GUI.

-	Min	Nominal	Max	Unit
Supply voltage in 24V mode	-	24	-	V
Supply voltage in 12V mode	-	12	-	V
Supply current in both modes	-	-	2000	mA

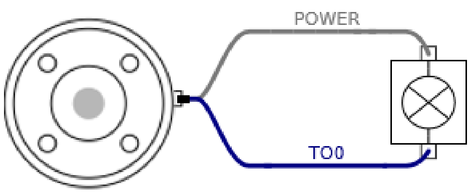
- [Non-E Version] The tool digital output is applied in the NPN method, when the digital output is enabled, the connection of the port leads to a ground (GND). When disabled, the output port is open (open-collector/open-drain), the electrical specifications are as follows.
- [E, U Version] The tool digital output is applied in the NPN method, when the digital output is enabled, the connection of the port leads to the VCC. When disabled, the output port is open (open-collector/open-drain), the electrical specifications are as follows.

Non-E Version	Min	Nominal	Max	Unit
Open Voltage	0	-	24	V
Current via GND	0	-	2000	mA

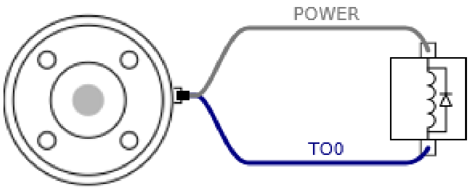
E, U Version	Min	Nominal	Max	Unit
Open Voltage	0	-	24	V
Current via VCC	0	-	50	mA

3) Non-E Version Usage Examples

- The image shown below illustrates how to turn on/off a load with 12V or 24V. The voltage level can be specified in the Tool Out block.



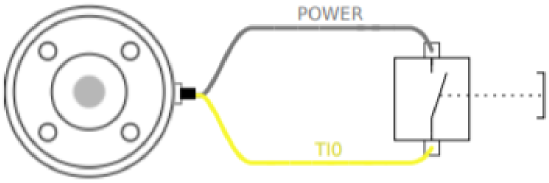
※ It is strongly recommended to use a diode to protect the tool using an inductive load.



- The tools digital inputs use PNP and pull-down resistors. Therefore, when the input port is not connected (floating), the corresponding input port is read as low (0). Electrical specifications are as follows.

-	Min	Nominal	Max	Unit
Input Voltage	0	-	24	V
Logic Low-Voltage	-	-	9	V
Logic High-Voltage	10	-	-	V

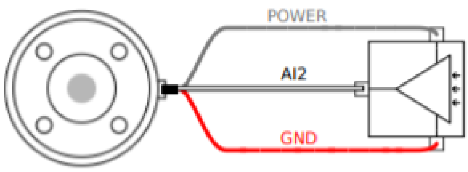
- The figure shown below illustrates how to use the digital input for a simple switch.



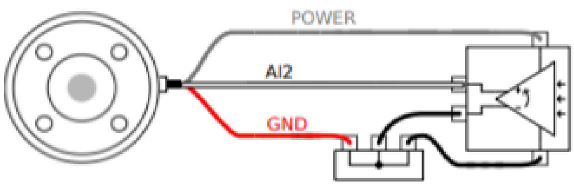
- The tool analog input measures the voltage in a non-differential manner. The measurement categories are as follows.

-	Min	Nominal	Max	Unit
Input Voltage	0	-	10	V
Resolution	-	12	-	bit

- The figure below shows how to connect an analog sensor with non-differential voltage output characteristics to the tool flange.

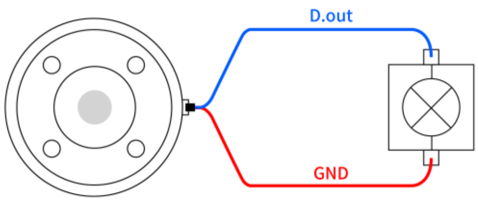


- The figure below shows how to connect an analog sensor with differential voltage output characteristics to the tool flange. Connecting the negative output of the sensor to GND (ground) works the same as the non-differential light sensor.

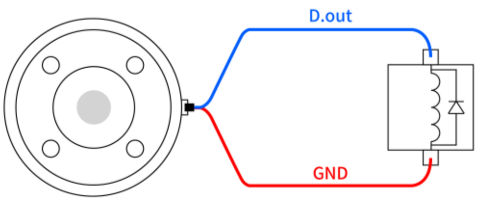


4) E Version, U Version Usage Examples

■ To use the tool digital output, the load is turned on using a 12V or 24V power supply as shown below. You can define the output voltage in the Tool Out block.



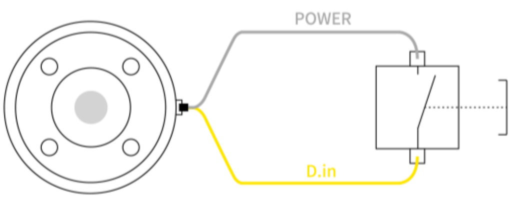
※ It is strongly recommended to use a diode to protect the tool using an inductive load.



■ The tool digital input is applied in the PNP manner. Therefore, if there is no connection to the input terminal (floating), the input port is read as Low, 0). The electrical specifications are as follows.

-	Min	Nominal	Max	Unit
Input Voltage	0	-	24	V
Logic Low-Voltage	-	-	3	V
Logic High-Voltage	4	-	-	V

■ To use the tool digital input, the figure below shows how to connect simple buttons.



■ E Version and U Version do not support tool analog input.

CAUTION

Caution:

1) For further details regarding technical specification and wire connection, please refer to Appendix D.
2) The cross-sectional view related to the tool flange is illustrated in Appendix C.

■ The tool flange supports RS485 serial communication and supports the following serial communication standard.

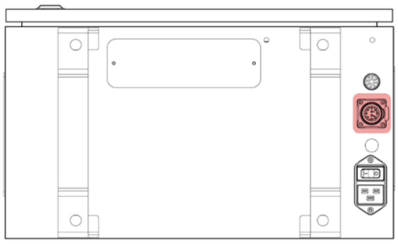
Baud-Rate	9600, 19200, 38400, 57600, 115200, 1M
Stop Bit	1,2
Parity	None, Even, Odd

4.6 CABLE CONNECTION

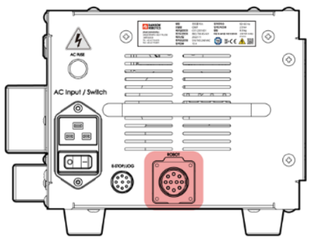
The cable connection for control box describes as follows.

- Connecting the robot arm to the robot control box using the robot arm cable

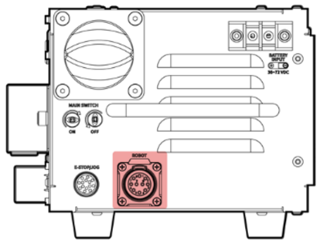
  Please connect the female connector to the robot arm and the male connector to the control box. Please check whether pins in the connector are bent or not.



Stand Control Box



Compact Control Box

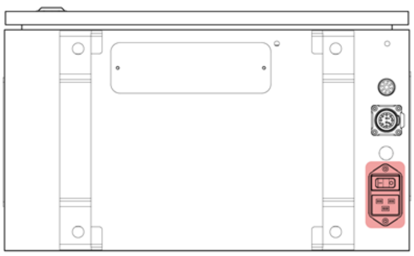


DC Control Box

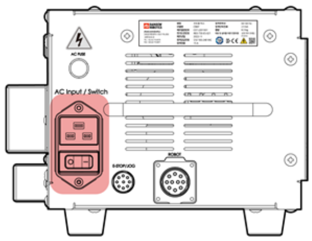
[Connecting Part for Robot Arm cable]

- Connecting the power cable to robot control box

  Connect the power cable to the power terminal as shown in the figure below.

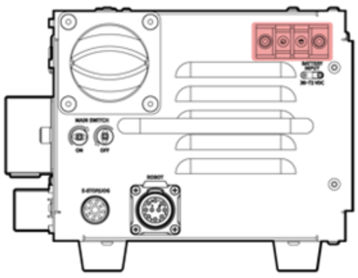


Stand Control Box



Compact Control Box

[Connecting Part for AC power cable]



DC Control Box

[Connecting Part for DC power cable]

The specification of the power system is as follows.

- AC power supply

Input Voltage	100 ~ 240 VAC
Input Frequency	50 ~ 60 Hz

- DC power supply

Input Voltage	48 VDC

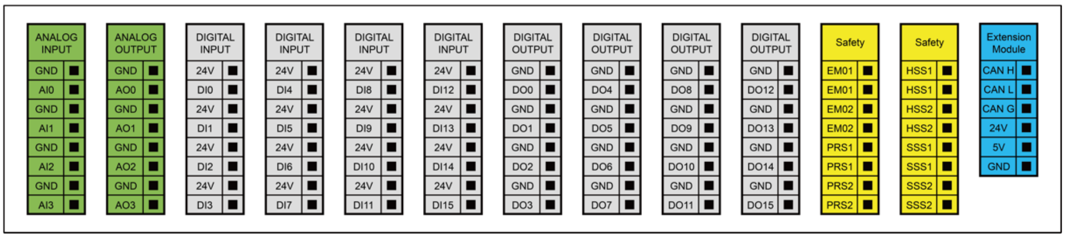
CAUTION

Caution:

1) Do not unplug the robot cable, power cable, or teaching pendant while the robot is turned on.
2) In the use of AC/DC power, the peripherals should share a common ground.

4.7 ROBOT CONTROL BOX I/O OVERVIEW

To connect other external devices to the robot control box, please connect the I/O from the control box to the corresponding device. The I/O of the control box is very flexible, so it can be used to connect with various equipment such as relays, PLCs, and emergency stop buttons. The layout of the electrical interface inside the robot control box is as follows.



The specifications of the power and digital I/O provided by the control box are as follows. All digital I/O is compliant with the IEC 61131-2 standard.

※ IEC 61131-2: IEC standard for programmable controllers

Terminal	Parameter	Min	Type	Max	Unit
Digital Output					
[Dox]	Current	0	-	1	A
[Dox]	Voltage Drop	0	-	0.5	V
[Dox]	Current Leak	0	-	0.1	mA
[Dox]	Type	-	PNP		Type
[Dox]	IEC 61131-2	-	1A	-	Type
Digital Input					
[Dix]	Voltage	-3	-	30	V
[Dix]	OFF Range	-3	-	5	V
[Dix]	ON Range	11	-	30	V
[Dix]	Current (11-30V)	2	-	15	mA
[Dix]	Type		PNP+		Type
[Dix]	IEC 61131-2	-	1	-	Type

CAUTION

Caution:
When tightening the I/O wiring, please turn off the power to the control box in advance. Any damage to the product caused by the user's carelessness (24V power shorts, incorrect wiring, etc.) is not covered by the product's warranty.

4.8 SAFETY INPUT CONFIGURATION

For the safety of users, all safety-related I/O must be configured with multiple backups. Safety devices and equipment must be installed in accordance with the instructions in Chapter 2 Safety and Chapter 3 Installation. The safety input has a safety protection stop. Emergency stop input is for emergency stop of the robot and safety stop input is for all safety level protection of the robot.

DANGER

Danger:

1) Never connect a safety signal to a PLC other than a safety PLC. Failure to follow these warnings could result in unsafe operation, resulting in serious injury or casualty. The safety signal and general I/O signal must be separated.

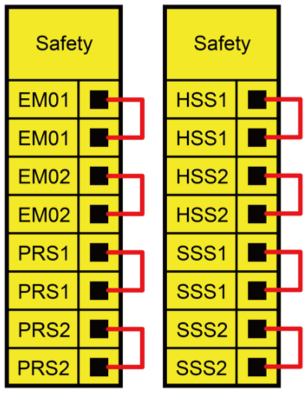
WARNING

Warning:

1) Inputs and outputs of all safety classes are redundant. It is necessary to isolate the channel so that the safety function is not activated due to signal failure. The safety functionality must be confirmed before installing the robot. The safety functionality should also be checked periodically for abnormalities.

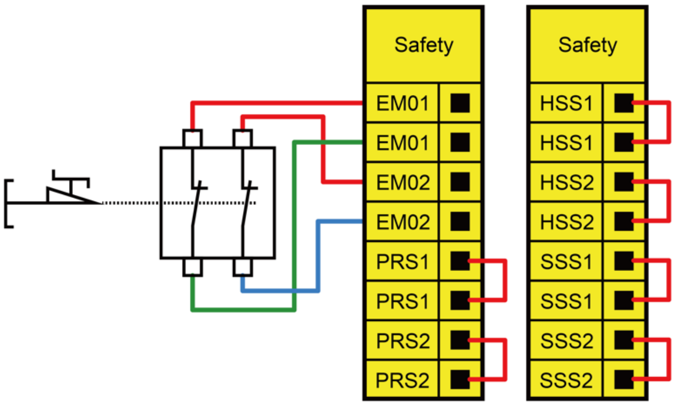
- Initial Safety Configuration

The robot will be shipped with the initial safety configuration set to default, so that the users can use it without further configuration. The initial safety configuration is as follows.



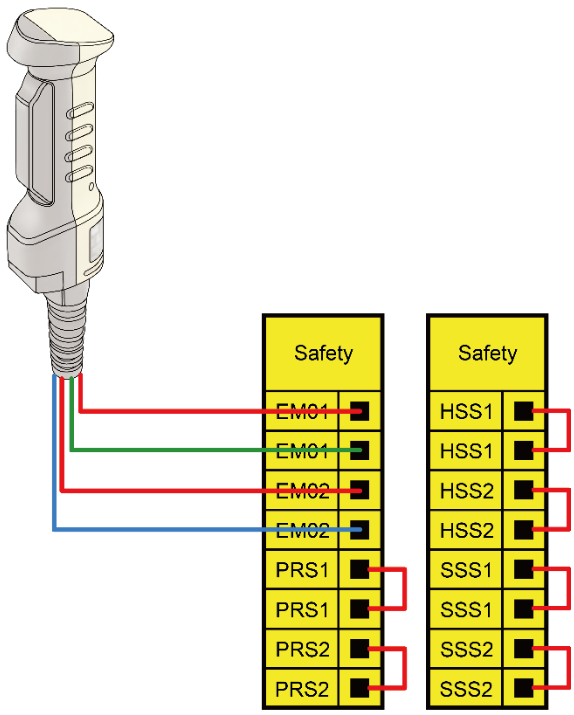
- Safety protection stop and automatic restart

An example of a safety protection device would be a door switch that stops the robot when the door is opened. The figure shows how to configure these features:



■ Enabling Device Input (Option)

Connect the active device input interface using the 3-position switch. When the position activation switch is in the operating position (middle position), the robot starts moving. If the 3-position activation switch is pressed, the switch is in the inoperative position and the robot arm will not move. Rainbow Robotics does not provide an Enabling Device. An Enabling Device is available as an option if the user needs. To configure the feature, refer to the following configuration:

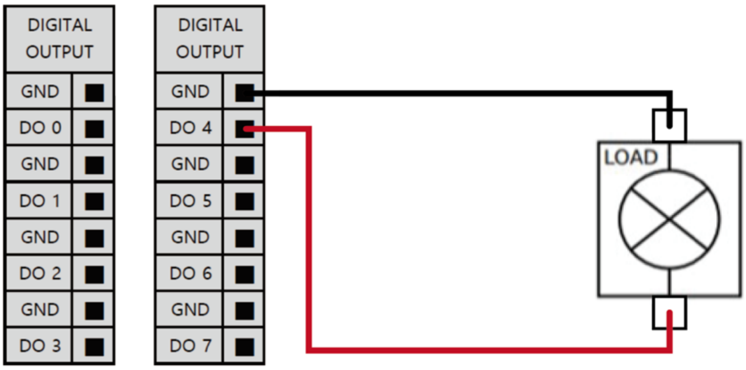


4.9 GENERAL PURPOSE DIGITAL I/O CONFIGURATION

All Digital I/O can be used as general purpose digital I/O. To use other external equipment with the robot, connect the I/O from the robot control box with the corresponding equipment. The universal digital I/O can be used to configure devices such as relays or PLC systems. In this configuration, the output is always LOW unless the program is running. The following subsections are examples.

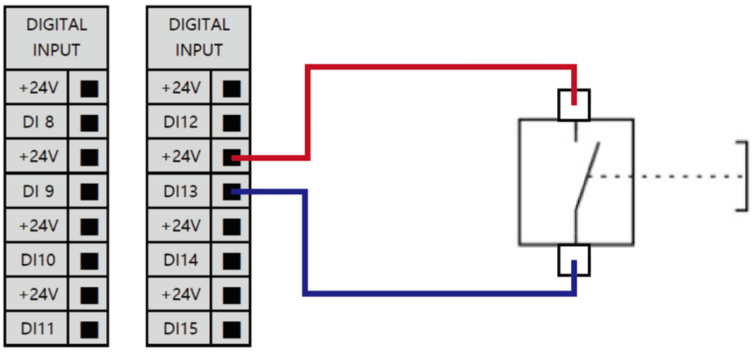
- Electric load control with digital output

The figure below shows a way to control electric load by using the digital output.



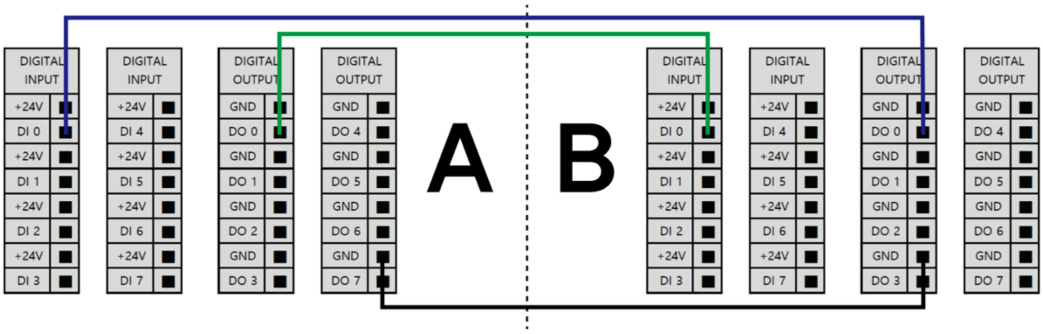
- Control of digital input with a button

The figure below shows a simple way of connecting a button to the digital input.



■ Communication with other system or PLC

If another other system provides PNP and uses a common ground, the digital I/O can be configured to communicate with the other system. Its connection is shown in the figure below.



WARNING

Warning: For the details in the technical specification and wire connection, please refer to Appendix D.

4.10 GENERAL PURPOSE ANALOG I/O CONFIGURATION

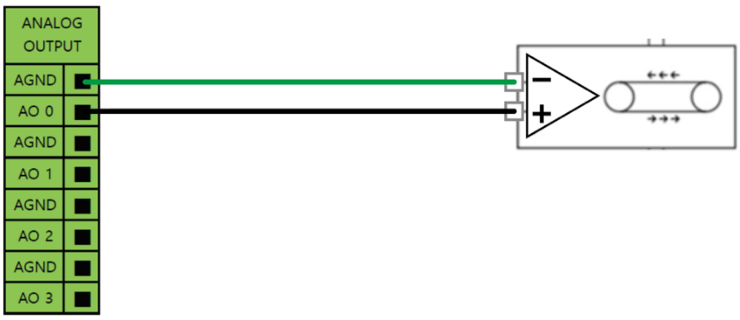
The following methods are recommended for high reliability.

- Use analog GND closest to I/O.
- Equipment and control box use the same GND. Analog I/O is not isolated from the robot control box.
- Use shielded or twisted-pair cable. Connect to the GND shield on the Power (J12) terminal.

Terminal	Parameter	Min	Type	Max	Unit
Voltage mode Input					
AIx - AG	Voltage	0	-	10	V
AIx - AG	Resolution	-	16	-	Bit
Voltage mode Output					
AOx – AG	Voltage	0	-	10	V
AOx – AG	Resolution	-	16	-	Bit

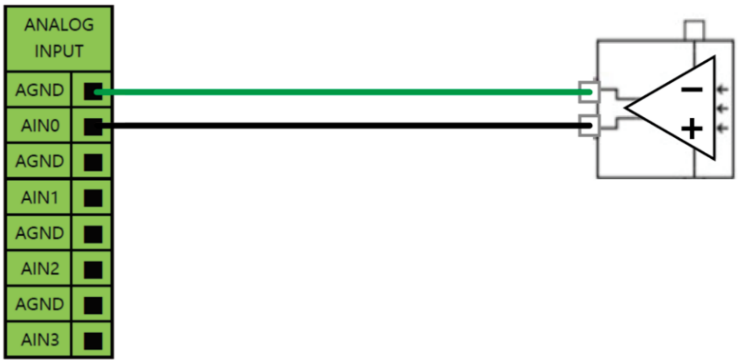
■ Analog output

The analog output can be used to control speed of conveyor. The figure below illustrates a simple demonstration.

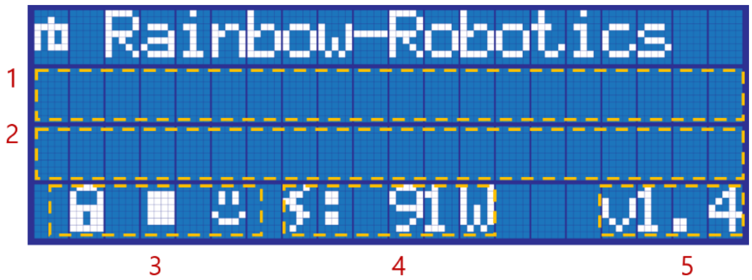


■ Analog input

The output value of the analog sensor can be used by the control box as analog input. The figure shown below illustrates a simple connection to an analog sensor.



■ LCD Status Display



1. Display Box(1): Displays information about system status.

    Please Wait: The main PC in the control box is booting up.

    default: The main PC in the control box is ready.

2. Display Box(2): Displays information about robot operation and status.
3. Action Icon: Definition lock () or release () state, play () or stop () state, crash () or safe() state.
4. Power Consumption: Indicates the total power consumption in watts (W).
5. System Version Information: System version information.

CHAPTER 5. GET STARTED

5.1 ROBOT CONTROL BOX ON/OFF

Procedure for turning control box On/Off is as follows.

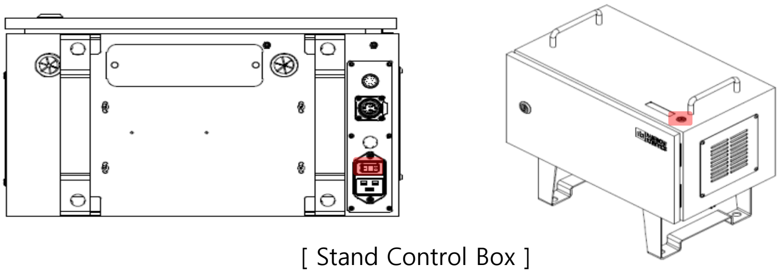
- Control box On/Off

  Press the AC/DC power switch at the bottom of the control box to apply AC/DC power.

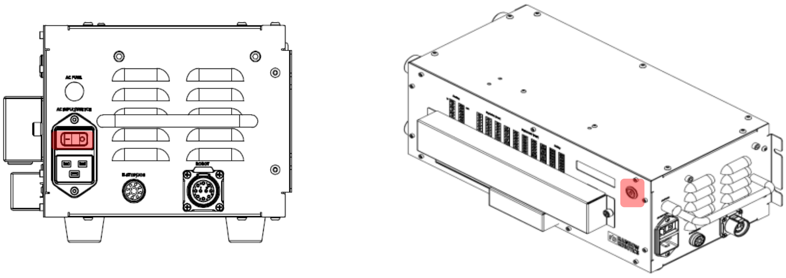
  Press the main power switch at the top of the control box to turn on the main power.

  The message "Please Wait" will appear on the LCD screen of the robot control box. This indicates that the control box is in the initial booting state.

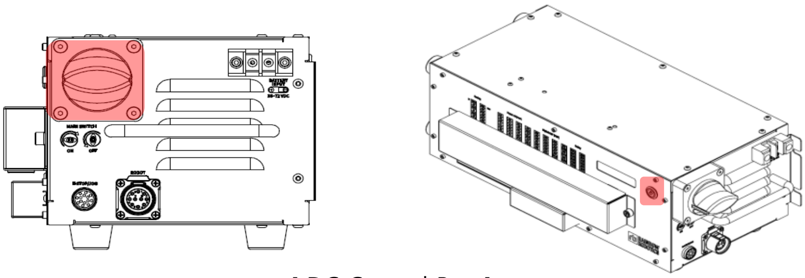
  Once the control box becomes ready for use, the LCD message will change to "default."



[Stand Control Box]



[Compact Control Box]



[DC Control Box]

To turn off the power, press the main power switch during few seconds.

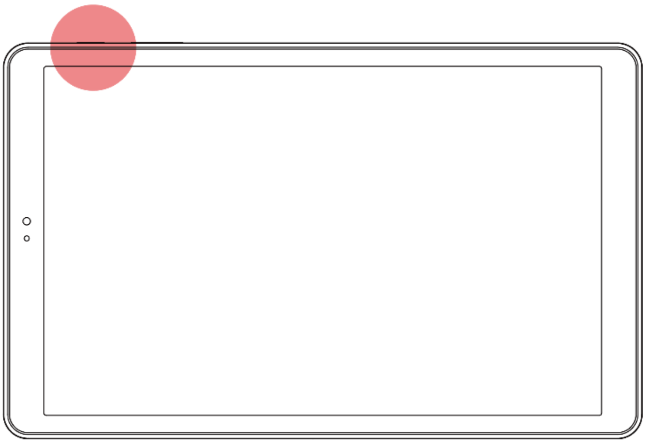
CAUTION

Caution: Our control box uses AC 100-240V single phase (50-60Hz), DC 48V.

5.2 TABLET PC ON/OFF

If you have selected the tablet PC option provided by the manufacturer for the control box, you will receive the tablet PC and cover as shown below. To turn on the tablet PC, press the power button located on the top left of the tablet.

- For Control Box



CAUTION

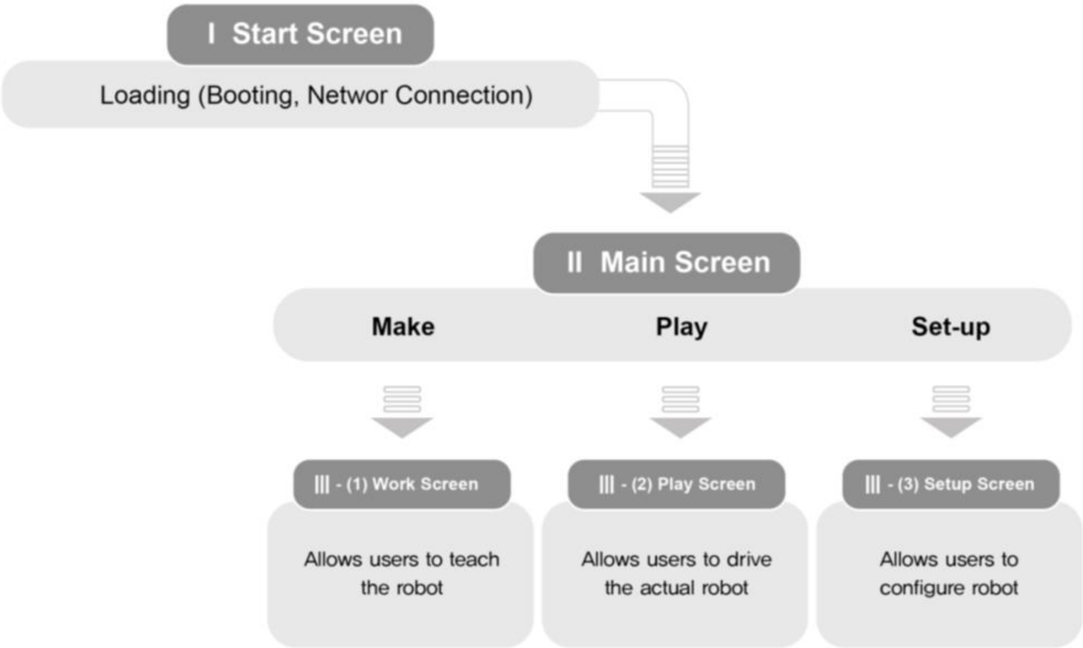
Caution:

1) Please ensure that the tablet PC is connected to the control box and run the application provided by the manufacturer. Do not perform unnecessary operations on the screen while the system is booting. It may cause problems with the system.

CHAPTER 6. SOFTWARE OVERVIEW

6.1 UI STRUCTURE

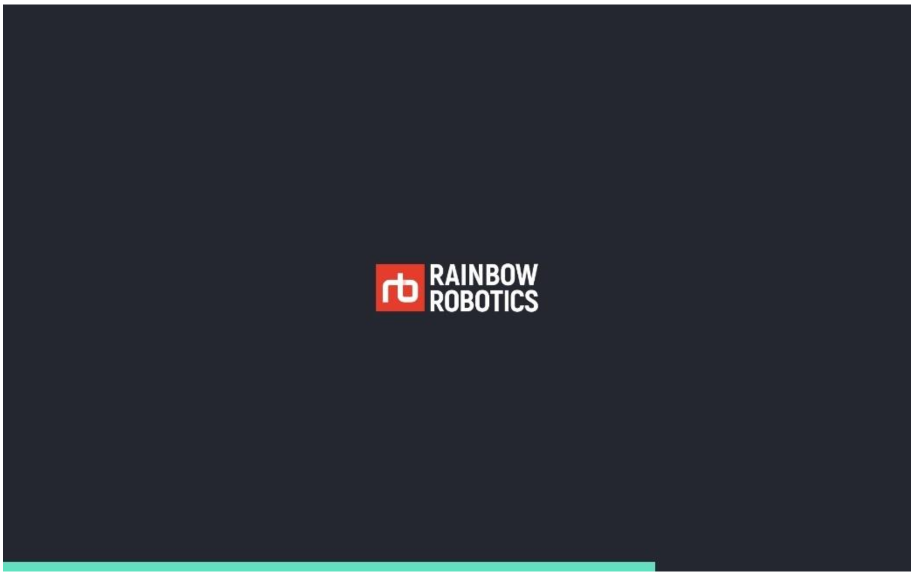
The UI (User Interface) program is divided into three screens as follows. Each section allows the user to enter necessary steps.



6.2 STARTUP SCREEN DISPLAY

- Intro

The start screen will occur while the application is loading its processes.



- Login (Factory-Default login password: 0000)

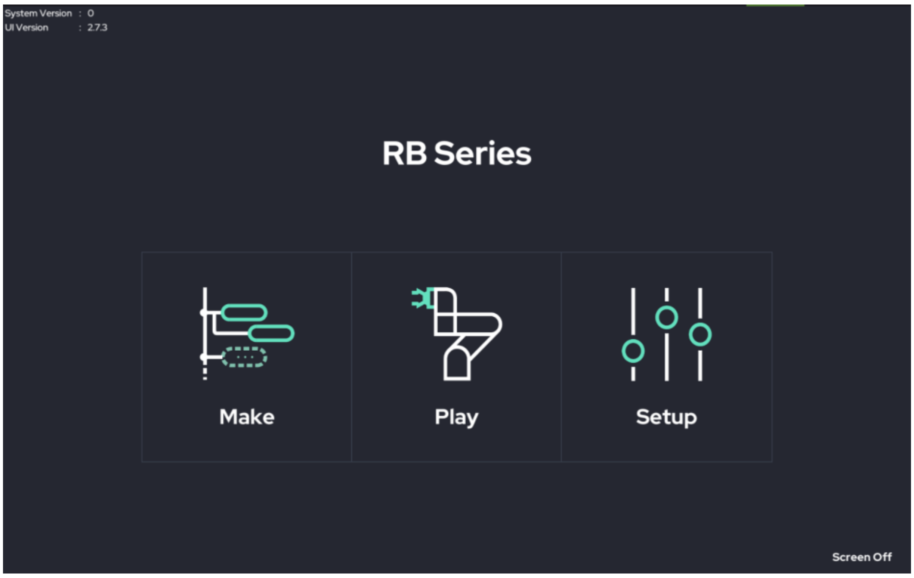
Check login at startup. Password setting and auto login setting will be done at "**Setup > Security**".

6.3 MAIN SCREEN DISPLAY

The UI has three main menus.

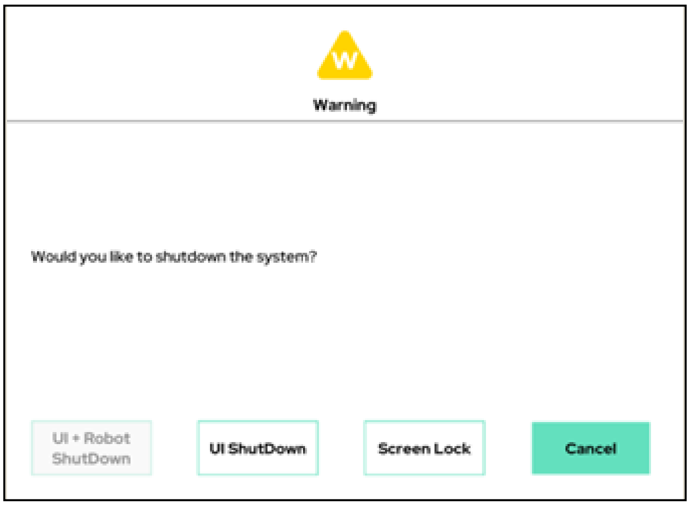
- Make: The area where you actually create programs (task sequences).
- Play: The area where the created program is simply executed.
- Setup: The area where various parameters are configured.

On the main screen, you can enter stages to create the robot's movements (Make), make it move in real-time (Play), or configure the work environment (Setup) through the three menus.



■ Power Off

When you touch the power button at the bottom right, the power-off window will appear. Pressing UI Shutdown will close the program. If the robot is activated and the control box is connected to the UI tablet PC, the robot's power will also shut down together.



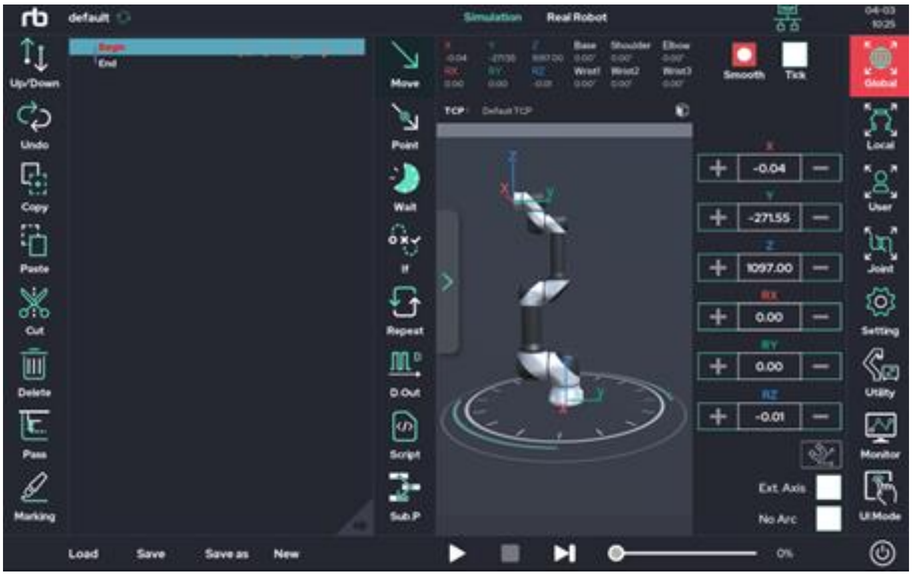
If you press Screen Lock, the tablet PC's screen will become unresponsive to touch. To unlock it, drag the Slide to unlock arrow on the bottom right to the right, and the screen will unlock.

6.4 MAKE

■ Make

This is the screen where you can teach the robot. You can teach the robot arm using the icons at the top of the screen. Use the icons listed on the right side of the screen to move the robot arm and use the icons on the left side to edit the teaching content.

- Left Icons: Functions for editing the program structure, such as copy/paste/save/delete/comment.
- Right Icons: Basic robot jogging (Jog/Jogging) and other settings.
- Middle Icons: Functions for structuring the program are placed here.
- Bottom Icons: Interface elements like program save/load, play, speed control bar, etc.



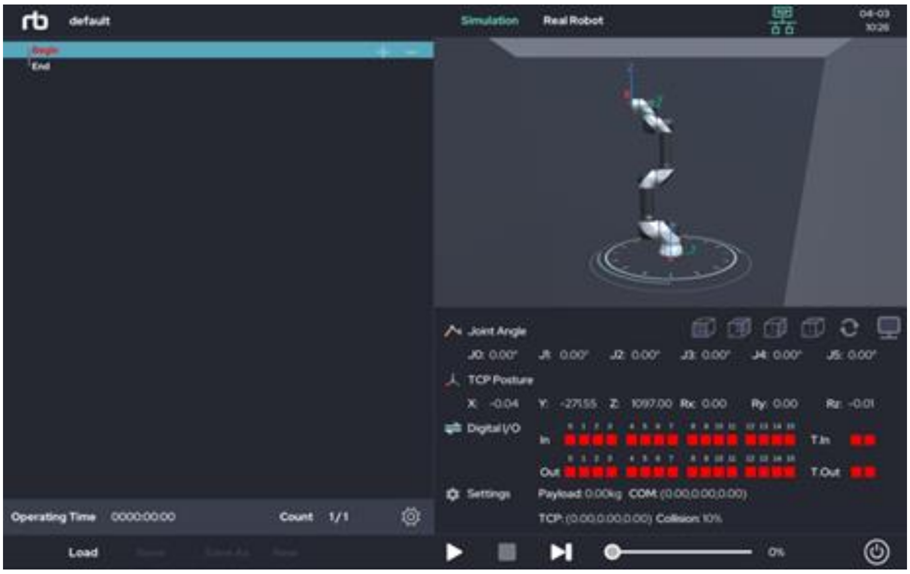
※ For a more detailed explanation of the icons and screen layout, refer to Chapter 7.

※ In the Make menu (Manual mode), the robot will only operate while the movement-related button is being pressed. If the button is not pressed, the robot will not move. This setting can be adjusted in Setup > Interface.

6.5 PLAY

- Play

This is the screen where you actually operate the robot arm. In this screen, you can load and execute the teaching files (programs). The number of play repetitions can be set in Setup > Interface. The total accumulated time after initial playback is displayed at the bottom left.

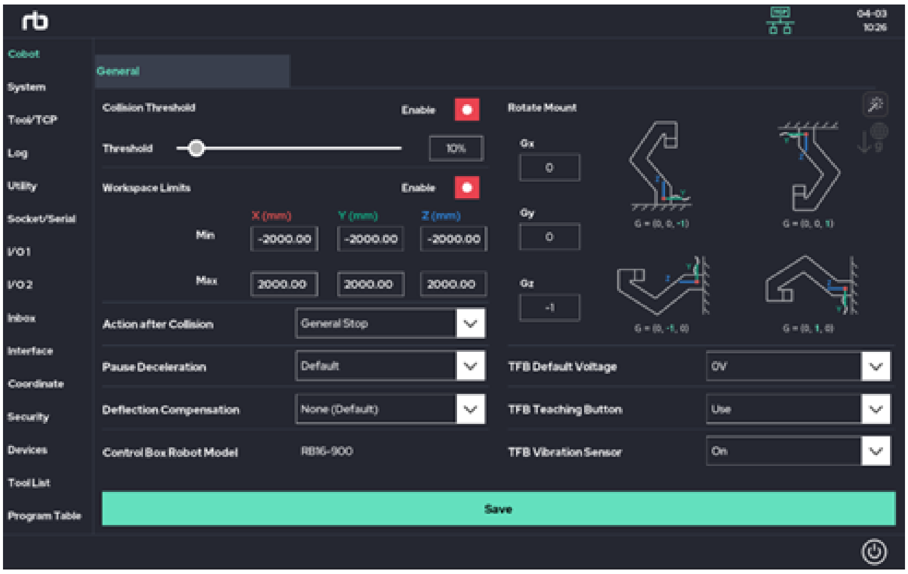


※ For more detailed explanations, refer to Chapter 8.

6.6 SETUP

■ Setup

This is the screen where you can configure various settings to use the robot arm. In the setup menu, you can adjust collision sensitivity, robot installation angle, work area, tool setting parameters, system log view, I/O function settings, coordinate system settings, and more.



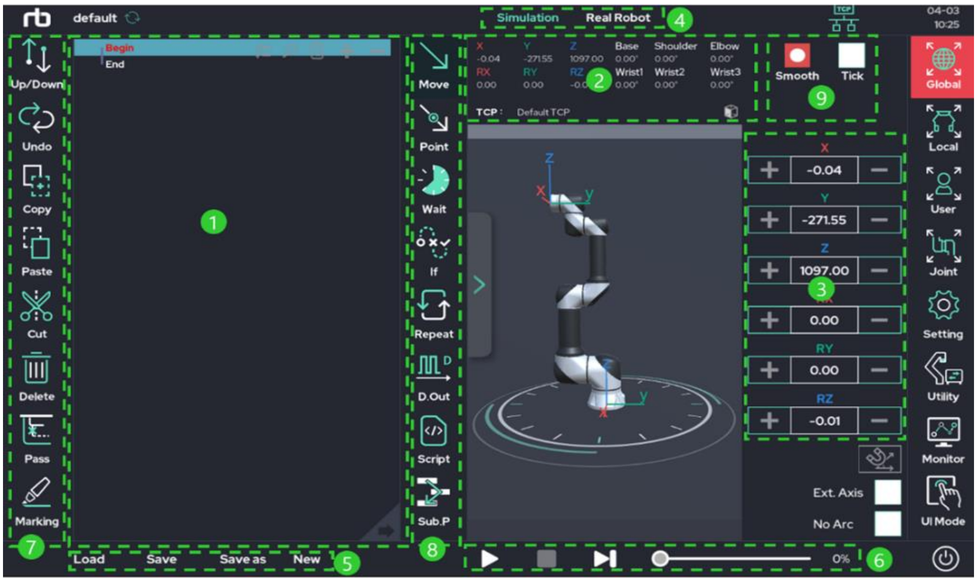
※ For more detailed explanations, refer to Chapter 9.

CHAPTER 7. PROGRAMMING GUIDE

7.1 ICONS AND ACTION SCREEN

- Make

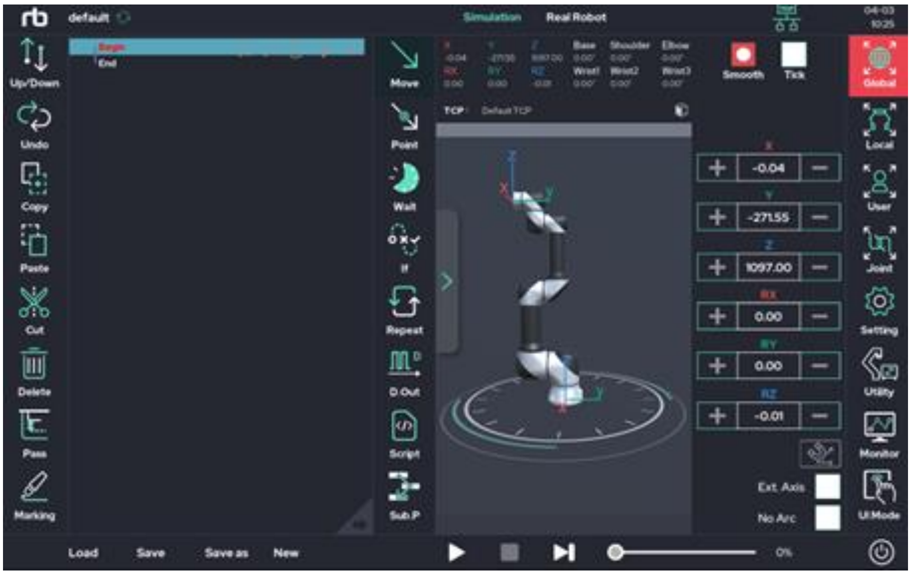
Description of components in Make screen display.



No.	Description
①	Show the program list in tree form.
②	Shows the angle of each joint of the robot arm and the Cartesian coordinate position of the TCP.
③	TCP Jog: can change the Cartesian coordinate position. (base/tool/user defined coordinate system Selectable) Joint: Jog: can change the angle of each joint.
④	Button to switch to Simulation or Real mode. ※Real mode must be selected to drive the real robot arm.
⑤	Can import saved projects and create new projects.
⑥	Starts or stops the program and exists Motion Speed Adjustment Bar.
⑦	Various editing tools are located, such as Copy/Paste/Annotations.
⑧	Various program functions (command/action) are deployed. Click the arrow on the right to include more functions.
⑨	Determines Jog Method – either Smooth Mode or Tick Mode.

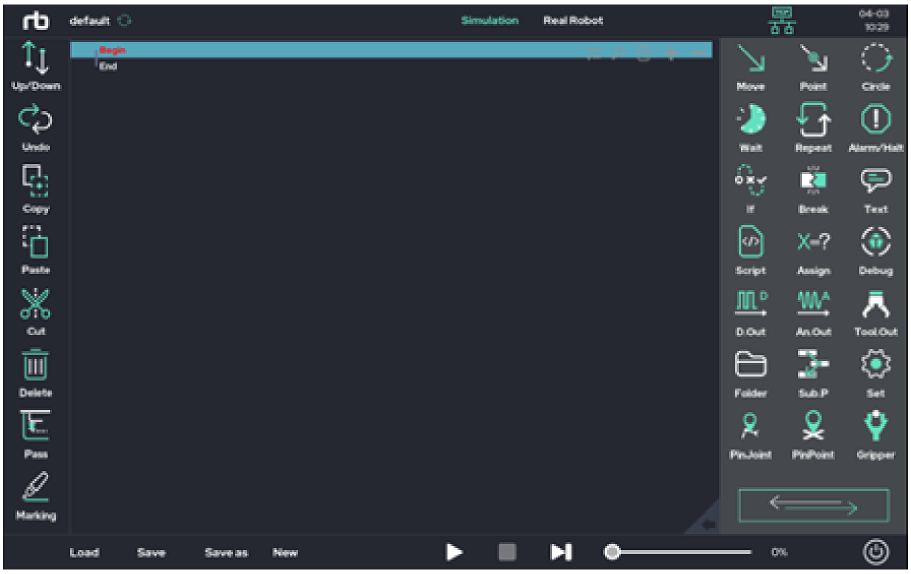
※ Teaching: programming RB's motion by means of moving the robot by hand.

※ TCP (Tool Center Point): The point defined for the tool center point within the robot's base coordinate system. It may also be the origin of the end-effector.



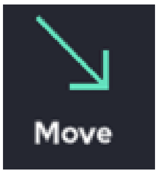
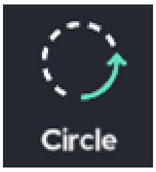
[Basic View Mode]

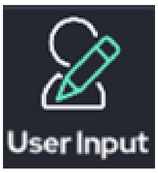
[Icon Extended View Mode]



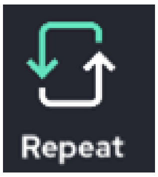
[Program-only Mode]

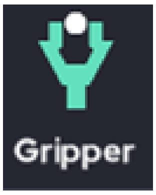
- Teaching Icon (Function Icon, Make screen)

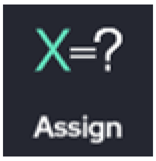
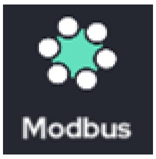
Icon	Description
	You can set the movement methods for the robot arm for teaching. There are two types of movements: Move J, which moves each joint to the specified target joint angles, and Move L, which moves the TCP linearly to the specified target TCP values. Additionally, there are four other movements that can be applied based on these two basic movements: Move JB, Move PB, Move JL, Move ITPL, and Move Pro, making a total of seven built-in movements.
	This feature allows you to input the target values as sub-items of the movements. For Move J and Move JB, the target values are the joint angle values. For Move L, Move PB, Move JL, Move Pro, and Move ITPL, the target values are the orthogonal coordinates of the TCP (x, y, z, rx, ry, rz).
	In addition to the Move, there is a feature specifically for circular motion. It includes a mode for drawing an arc by specifying a starting point and two additional points, forming an arc that passes through all three points. There is also a mode that allows you to draw a circle by setting a center point and a rotation axis.
	This is a function that can be used by storing certain posture/position information as a variable and then referring to (calling) another motion function. However, this function itself does not move to that position.
	This is the pin joint feature used to store and assign specific joint angle values as variables. It does not move to the specified location. Instead, the joint angle values are saved as variables and can be referenced (called) in other motion functions for use.
	This function moves the robot to either the starting position of the program or the zero position of the robot's joints. The movement method can be chosen between Move J and Move L.

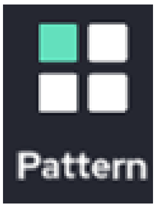
	This function allows the robot to wait for a specified condition or amount of time. There are 5 modes available: waiting for a specified time, waiting while a condition is true, ending the wait when a condition becomes true, waiting while an input condition is true, and ending the wait when an input condition becomes true.
	This feature helps organize program commands into modules by grouping them under sub-items. Each folder can be named differently, making it easier to manage the program.
	This icon allows a user to add a comment or memo to the program.
	This function allows for program termination and the display of an alarm pop-up message. There are four options: terminate the program, terminate only the sub-program, terminate only the folder's sub-programs, and display an alarm pop-up message.
	This is the Debug function for internal value debugging. There are two options: Debug Pop-up and User Log. The Debug Pop-up allows you to check specific variable values or internal parameters by requesting them in a pop-up format. The User Log lets you leave a string or string variable in the log system. However, logging too frequently may affect the system.
	The user input function pauses during program execution to allow users to change the value of a variable/arrangement/point/character/global/ROM by entering it. You can change the value, ignore it, and skip it, depending on your situation.
	This function is for controlling external axis in addition to the robot. Up to 6 can be added.

	This icon is used to create a conditional statement. A user can generate separate motion program branches depending on conditions using If, Else if, and Else.
	This icon is used to create another type of conditional statement. For the Switch statement, a user defines each case.
	This feature includes the Pre Program, which executes the contents only once at the beginning, and the Post Program, which defines functions to be performed after the program ends. In cases where the program runs continuously, such as in Play mode, the Pre Program content is executed only once. For one-time commands like variable declaration or communication setup, you can manage them under the Pre Program. The functions declared under the Post Program are executed sequentially after the program ends. Motion-related commands cannot be used in the Post Program.
	This is the Set function. It allows you to configure various setting parameters that are typically set in the Setup menu directly within the program. You can change specific setting parameters during program execution.
	The TCP setting function allows you to temporarily change the TCP value during program execution by loading a pre-saved TCP value. TCP values can be saved in advance in the Tool List on the Setup page.
	This manual operation feature allows you to pause the program and perform direct teaching during its execution.
	It is a function to operate the robot joint and the external axis at the same time.

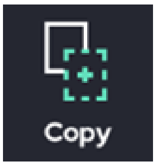
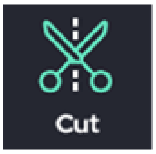
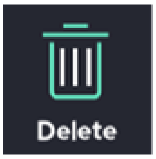
	This is a loop function. It repeats the sub-program based on a specified condition or number of times. There are four modes: repeating a set number of times, repeating while a condition is true, the Do While function (executes at least once and continues while the condition is true), and ending the loop when the condition becomes true.
	This function forcefully breaks out of a loop. It is used under the repeat function, and even if a specific condition is declared for the loop, using the break function allows you to exit the loop early. Additionally, the continue function moves the program flow to the top of the loop, the program logic jump function allows you to jump to a logic point marked with Here, and there is also a function to jump to a folder or the logic jump destination.
	This is the thread function, which includes two options: adding a general/non-stop thread and calling an event thread. There are five thread properties: general thread, non-stop thread, non-stop thread 2, event general thread, and event non-stop thread. The sub-items of the thread function run in parallel with the main program. However, motion-related commands cannot be placed under the thread function.
	The G-code function is available when a G-code file is placed in the specified folder. The robot will execute the path defined by the G-code.
	This is a function to replay a recorded motion using the motion recording feature. The motion recorded via direct teaching can be replayed using either the J or L type with this feature.
	This function adjusts the TCP position based on feedback from the welding current via analog input signals. The user must adjust the current/voltage signal according to the analog input range (0~10V) provided by Rainbow.
	This function automatically generates predefined motions. Multiple predefined motions are available, and you can modify the parameters to create the desired motion.

	This command controls the digital output of the control box. There are seven functions: individual output, bit combination output, signal toggle output, full signal output, unit pulse output, pulse width modulation (PWM), and condition-based output. The selected signal is output through the chosen port among the 16 ports available. Each port can be set to High, Low, or Bypass signals.
	This command controls the analog output of the control box. The selected voltage is output through the chosen port among the four analog ports available. Each port can output a voltage range of 0~10V. There are two functions: individual analog output and speed-based analog output.
	The tool flange has two digital outputs. You can assign signals to these two digital outputs. Additionally, the voltage level (0V, 12V, or 24V) for the output from the tool flange can be adjusted.
	This is the gripper function. Functions for third-party grippers are pre-installed. By selecting the gripper you wish to use and the function provided by that gripper, the system helps you easily operate the gripper automatically.
	This function is available when an I/O expansion module is added. You can configure the digital/analog outputs of the I/O expansion module, which works similarly to the existing digital and analog output functions.
	This is a dedicated macro function for arc welding. It groups functions that can be implemented through common D.out or Wait functions into a macro format for quick use. Settings for the arc welder are performed in Setup > Device.
	This function is for using digital welders. After selecting the brand of digital welder to be used, you can choose the mode and options to easily operate the digital welder.

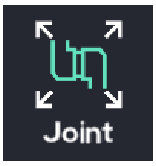
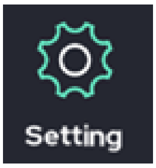
	This is a script function that allows the user to write custom scripts. It can be used for specific calculations, variable substitution, and assignment beyond the basic functions.
	This is the Assign function for declaring variables, with five types available. The types include variable for storing single numbers, array for storing arrays of numbers, point for storing positional information (x, y, z, rx, ry, rz), joint for storing joint angle information, and string for storing text.
	This function declares variables (single variables, arrays, point variables, etc.) for real-time monitoring during program execution. Variables declared in the monitoring function can be observed in real-time during program execution by clicking the monitor icon on the right side of the Make/Play page.
	This function enables RS485/232 output via the tool flange/control box. You can output in ASCII mode or hex mode. The communication protocol is set in the Setup > Socket/Serial menu.
	This is a socket communication function. It allows you to open a socket, connect to a specific server, send request messages, and receive specific data from the server. Up to five different servers can be connected simultaneously.
	This is a Modbus client function. It allows requesting data from a specific IP/address and retrieving the returned data. The request interval and format can be specified. A separate protocol is provided for the Modbus server included with this product.
	This function integrates features for using external products like HMI and PLC easily. You can select the desired product and its specific features for use.
	This function allows you to insert a pre-created program file (teaching file) into the current project. There are three options: sub-program call, copy command content, and switch program. The sub-program call allows viewing the sub-project without modification. If modification is necessary, the sub-project must be opened separately. The copy command content

	option lets you insert another program file into the current document in an editable form. Unlike sub-program call, which simply calls the program, copy command content copies the program in a modifiable format. The switch program option is used when you want to replace the main program with another project.
	This function defines repetitive actions. It defines the space where the repeated actions will be performed and the actions at each location. The same actions will be performed at all defined points. This function is useful for implementing palletizing.
	This is a conveyor tracking function. Once the conveyor speed and direction are set, the sub-actions (L, PB, ITPL, Pro, Circle) of the conveyor function will be added to the conveyor flow.
	This is a force control function. You can select the direction and coordinate system for the desired force. Sub-actions under force control will have this force control feature applied automatically.
	This is a dedicated function for weaving movements. Movements like L-series and Circle under weaving are combined with the set weaving options. The available weaving patterns are Trapezoidal, SineWave, Triangle, C-Wave, and Circle.
	This is a dedicated function for weaving movements. Unlike the existing weaving functions, the reference for weaving is through the TCP coordinate system. Movements like L-series and Circle under weaving are combined with the set weaving options. The available weaving patterns are Trapezoidal, SineWave, Triangle, C-Wave, and Circle.
	This function is used for welding applications. It detects the movement of the base material and reflects the movement and direction to proceed with welding.

■ Editing Icon (Left side in Make screen)

Icon	Description
	You can raise or lower commands one space at a time.
	Revert the operation to a step before, after, up to 50 times.
	It can copy the selected command and you can the copied command to a different location.
	It can paste the copied or cut command into the selected location.
	It can cut the selected command. This command can be pasted to a different location.
	It can delete the selected command.
	This is Annotation function. It prevents the selected command from running. Annotated commands exist in the program but not executed. If you press the disable button on the annotated command, it is activated.
	It can mark the highlight (marking) in the desired program line. Therefore, you can underline important program lines. There are two colors, blue and pink.

- Jog and other utilities (Right side of Make screen)

Icon	Description
	It can move TCP's position relative to a global coordinate system fixed to the base.
	It can move the position of TCP based on the local coordinate system (tool coordinate system) fixed to TCP.
	It can move the position of TCP based on the user-defined coordinate system (User Coordinate).
	It can allow to move each joint of the robot arm separately.
	This is a collection of settings such as User coordinate system settings, automatic TCP find, and other easy-to-use settings with a Jog. These settings can also be set in the Setup menu by default.
	This is a collection of special features which can view status and set-up values such as I/O status information of the system, user-coordinate Setting information and current/temperature information of the robot.
	It is a window for real-time observation of the values selected variables through the Monitor function. In addition to the selected variables, system variables that need to be checked frequently are also displayed.
	This function allows the user to select the UI mode. Users can select UI mode according to their level and environment. There are four mode options: Expert Mode, Beginner Mode, Welding Mode (Analog), and Welding Mode (Digital).

■ System function button

Icon	Description
	This icon is used to move to home screen & another page. (It is located in the top left.).
	This icon is used to power off the UI. When the tablet PC is connected to the robot, the robot will also be turned off. Screen Lock function is included in here. (It is located in the bottom right.).

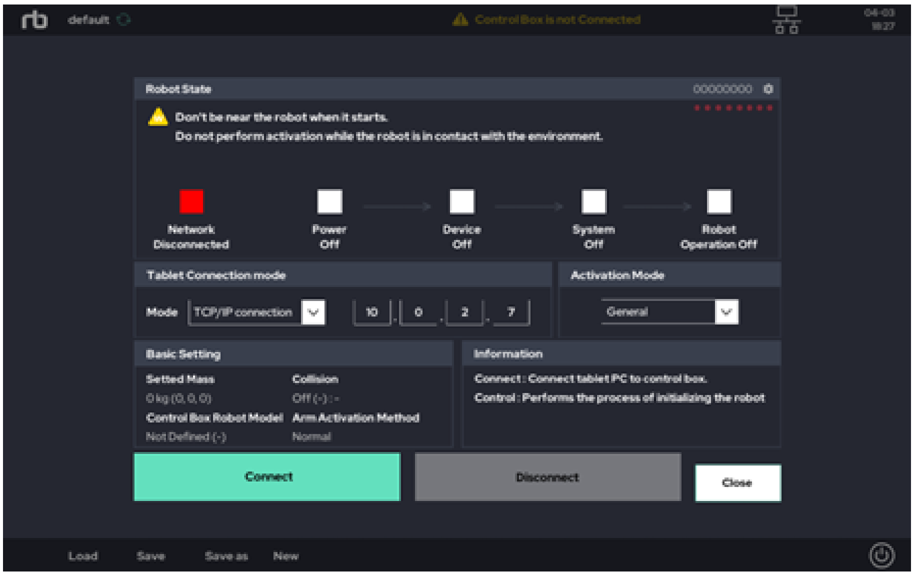
※ A detailed description of each function is described later.

7.2 CREATE TEACHING ENVIRONMENT

Robot teaching (programming) is available only in the Make screen. Please open the Make screen from the Play or Setup screen via the button located at the top of the UI. It is also possible to move to the Make screen from the Home screen.

■ Connect Tablet PC to Control Box

This icon is in the top right of each screen. State The robot control box and tablet PC must be connected before teaching. When this icon is pressed, the following screen is displayed.



Press the 'Connect' button to link the tablet PC to the robot control box.

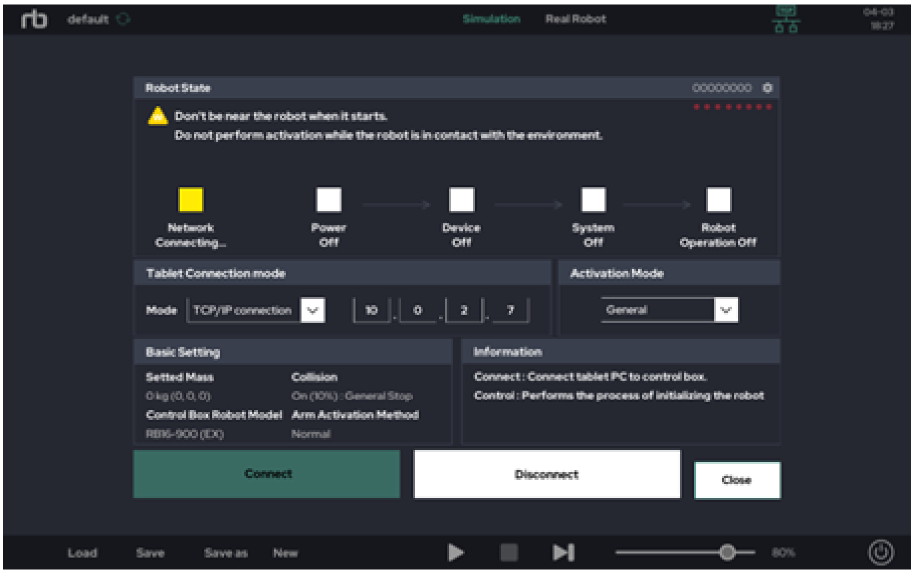
- 'Connect' button: Will connects the tablet PC to the robot control box.

CAUTION

Caution:

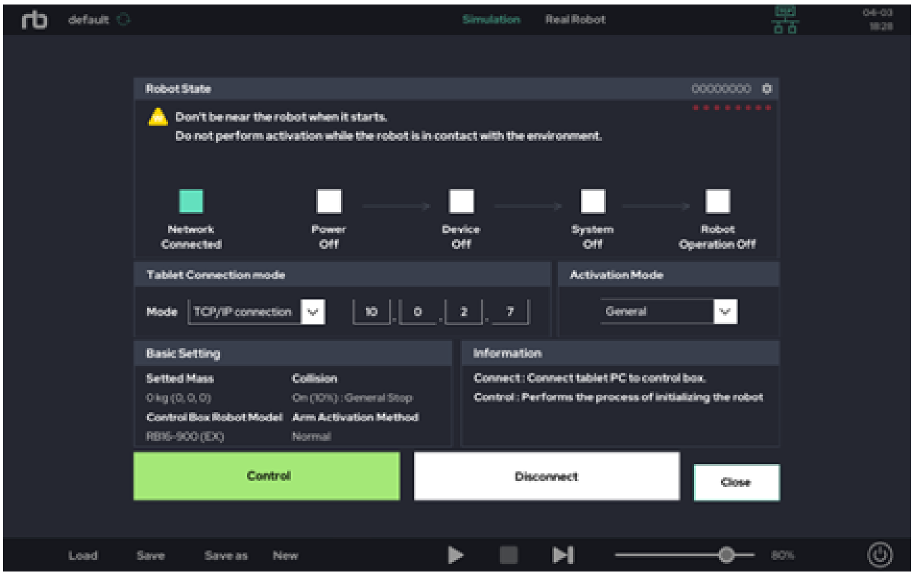
1) Make sure that the control box is turned on and that the emergency stop switch is turned off. If the control box is not on, the light beneath 'Device Off' will turn red.

The figure below shows a display when the tablet PC and control box are being connected.



'Network Connecting' lights yellow when the tablet PC is trying to connect to the control box.

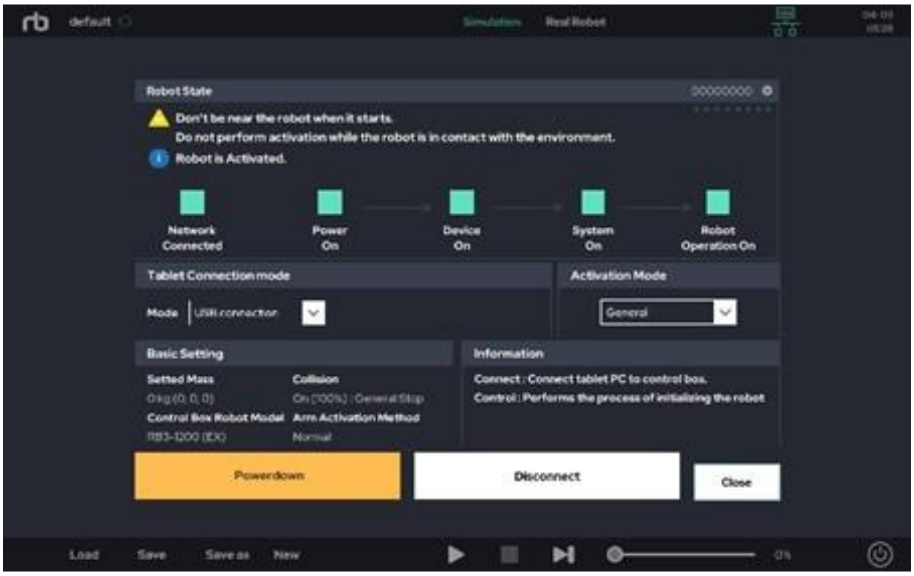
'Network Connected' becomes blue when the table PC and control box are connected properly. The 'Control' button is also activated once more.



After 'Network Connected', press the 'Control' button to activate the robot control system.

- 'Control' button: Will initialize the robot arm for operation.

Pressing the activation button initializes the robot arm, bringing it into a controllable state. During this process, the brakes on each joint will release, producing sequential sounds from the robot arm. Once all systems are powered on and connected, all status indicators will turn blue, as shown on the screen below.



※ If the robot does not show a blue light indicating successful activation and instead displays a red light, please check the emergency stop switch status, tool load settings, and robot installation angle. Follow the instructions in the popup message to review and adjust the settings and operating conditions as necessary.

■ Create New Project

To create a new project, click the 'New' button at the bottom of the Make screen. In the 'Create New Project' screen, select 'Start with an empty project,' and a screen will appear where you can assign a file name.

The default name of a new project is my_project. Please type a name for the new project and press the 'Save' button in the dialog. Note that the new project is not created if the 'Save' button is not clicked.

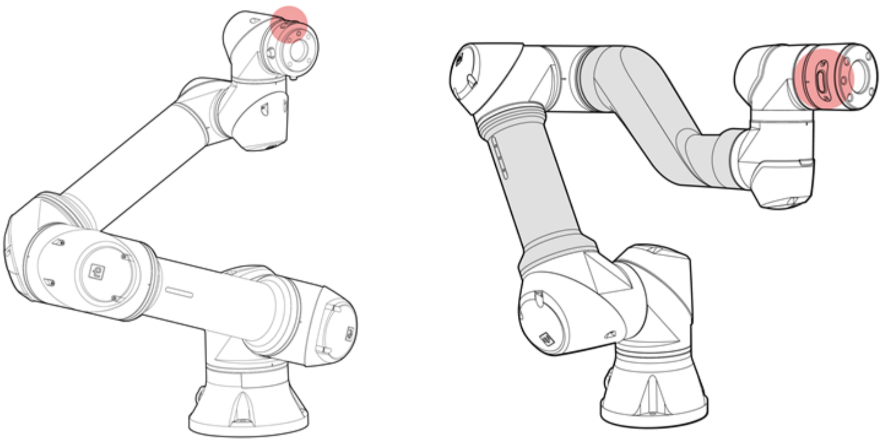
7.3 TEACHING (PROGRAMMING)

■ Ways to Move the Robot

There are two ways to drive the robot.

- Direct-Teaching: When a user manually rotates each joint to change the pose of the robot.
- Jogging: When a user uses the jog buttons in the UI to move the robot.

■ Direct-teaching / Gravity Compensation



For 'Direct-Teaching', a user must press and hold the mechanical button located on the tool flange. Pressing this button allows each joint to move freely.

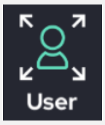
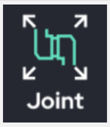
WARNING

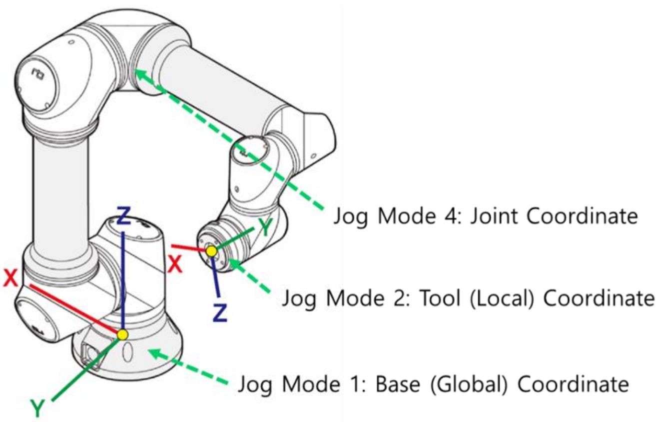
Warning:

1) 'Direct-Teaching' can be used only when the robot is initialized and activated.
2) The load value in 'Setup-Tool' should be set prior to using 'Direct-Teaching' when a tool is installed at the tool flange. Without a proper value of the load, 'Direct-Teaching' may not work properly.
3) In 'Setup-Interface', the sensitivity of joint reaction can be adjusted.
4) Please ensure that the robot is not moving before using 'Direct-Teaching'.

■ Jogging

There are four modes of jogging.

Mode 1		TCP Movement in the Cartesian coordinate system with respect to the base (global) frame.
Mode 2		TCP Movement in the Cartesian coordinate system with respect to the tool (local) frame.
Mode 3		TCP Movement in the Cartesian coordinate system with respect to the user coordinate frame.
Mode 4		Angular joint movement.

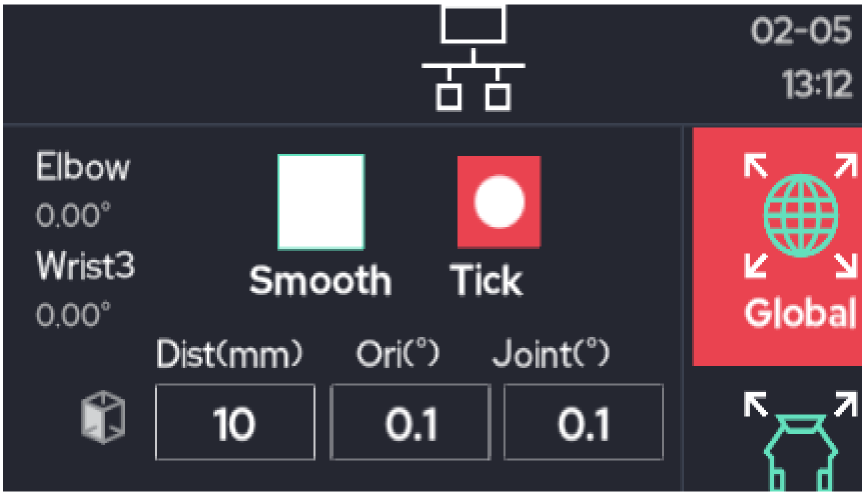


There two ways to control jogging:

- Smooth: Use for continuous motion of the robot. When the '+' or the '-' button is pressed and held, the robot moves continuously until the button is released.
- Tick: Use for discontinuous motion of the robot. The robot will move a specific amount as defined by the user each button click.

※ The control method for jogging can be selected via a toggle button located in the top right in 'Make' screen.

※ In 'Setup-Interface', a user can specify the amount of movement for each 'Tick' button press. Or it can be directly changed in pop-up window as below.



WARNING

Warning:

1) Make sure that there are no obstacles or people in the robot's workspace before the use of jogging.

2) It is highly recommended to use the 'Safety Slider' feature in 'Setup-Interface'. This feature is activated as a factory default.

[Jog Mode 1: TCP jog w/ Global coordinate]

[Jog Mode 2: TCP jog w/ Local coordinate]

[Jog Mode 3: TCP jog w/ User coordinate]

[Jog Mode 4: Joint jog w/ joint coordinates]

■ Real Robot and Simulation Modes

Two Modes are available for testing the robot's movement.

- **Simulation Mode:**

  Allows the user to virtually move the robot arm on the UI screen without moving the actual robot.

  It is recommended to run simulation mode first for safety reasons before teaching a new motion.

- **Real Robot Mode:**

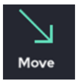
  Drives the real robot as displayed on the UI screen.

WARNING

Warning:

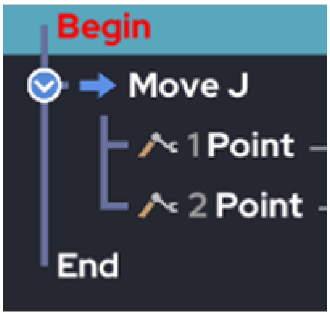
1) Real Robot mode is only available when the robot is connected and activated.
2) Simulation mode only requires the provided tablet and the control box. It does not require the robot arm.
3) When using Real Robot mode, please make sure that the nearby environment is clear & safe before operating, as the robot will move.

Teaching Robot Movement

The basic robot teaching functions are  **Move** and  **Point**. Both icons are on the top bar when using the **Make** screen.

- **Move**: Defines motion property for the robot arm.
- **Point**: A sub-function of **Move**. Defines a destination position for each movement.

After using the Move and Point functions in an empty program, the script field in the UI will look like the following.

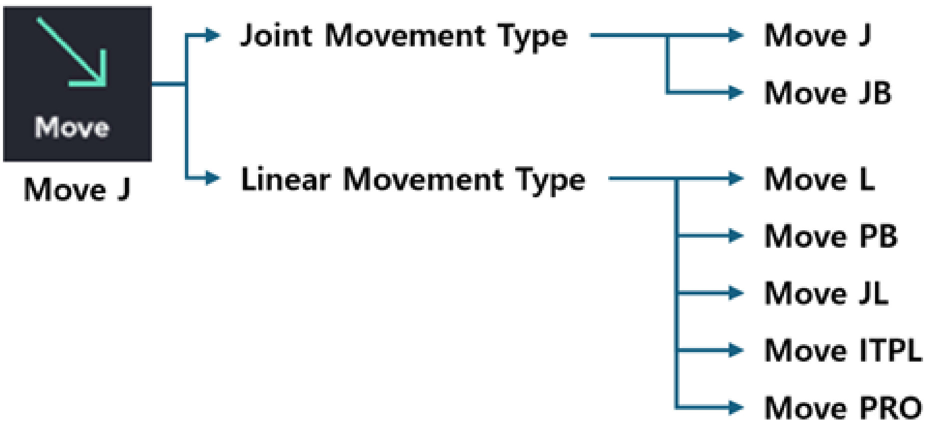


← Move Type

← First Target Point

← Second Target Point

Move Function



Move sets the robot arm's motion properties. The two primary types of movements are **Joint** and **Linear**. These types are further broken down into commands, as shown in the figure above.

① Joint Movement Type

The Joint Movement Commands generate movement by setting the angular value of each individual joint (in degrees). The sub-elements are Move J and Move JB.

▷ Move J (Move Joint):

Sets each joint angle to the values contained within the target Point.
The movement speeds for all joints are slowed relative to the joint that requires the most movement time.

▷ Move JB (Move Joint Blend):

Starting from the initial arm configuration, the arm will move smoothly between each Point without stopping by using the Move J method.
Move JB has two modes: time-based blend and direction-based blend.

② Linear Movement Type

The Linear Movement Commands generate movement by setting the position of the TCP in the Cartesian coordinate system. These commands use Cartesian coordinates (x,y,z coordinate values and rotations) as the target values for the movement. Sub-elements include Move L, Move PB (formerly Move LB), Move JL, Move ITPL, and Move Pro.

▷ Move L (Move Linear):

The TCP moves in a straight line from the starting point (current position) to the target point (set position). The tool's orientation rotates with the least possible angle. For tasks where the TCP movement and orientation change take a significant amount of time, the speed of other tasks is adjusted accordingly.

▷ Move PB (Move Point Blend, Old Move LB):

The TCP moves smoothly from the starting point (current position) between points using the Move L method without stopping.
A blend distance is assigned to each point, cutting and connecting the previous and next paths with an arc, allowing the TCP to follow a smooth trajectory.

If the designated blend distance is larger than half the length of the adjacent paths, it is automatically adjusted to half the length of the shorter path.

The blend ratio at each via point can be set in distance units or as a percentage (%). Additionally, the speed can be configured individually for each via point.

Move PB has three modes:

Constant Mode: The tool’s orientation remains the same as it was at the starting point (current position) during movement.

Intended Mode: The tool’s orientation changes according to the saved orientation at each point.

Smooth Mode: Unlike Intended Mode, where the rate of change in X, Y, Z, and orientation remains constant, Smooth Mode has a lower rate of change in orientation as it approaches the start and end points.

▷ Move JL (Move J with Linear Input):

Like MoveL, the Cartesian value of the target point is used as input. However, instead of going straight to the point, it uses MoveJ's method. When the Cartesian coordinate system input is received, it is converted into the target joint angle through inverse kinematics and inputted again to MoveJ.

▷ Move ITPL (Move Interpolation):

The TCP moves smoothly from the starting point (current position) between points using the Move L method without stopping. While Move PB blends paths without passing through each waypoint exactly, Move ITPL moves along a trajectory that passes through each waypoint precisely, and thus does not require a separate blend setting.

Move ITPL has six modes: Constant Mode: The tool’s orientation remains the same as it was at the starting point (current position) during movement. Intended Mode: The tool’s orientation changes according to the saved orientation at each point. Smooth Mode: Unlike Intended Mode, where the rate of change in X, Y, Z, and orientation remains constant, Smooth Mode has a lower rate of change in orientation as it approaches the start and end points.
Additionally, the speed can be configured individually for each intermediate waypoint.

▷ Move Pro (Move Process):

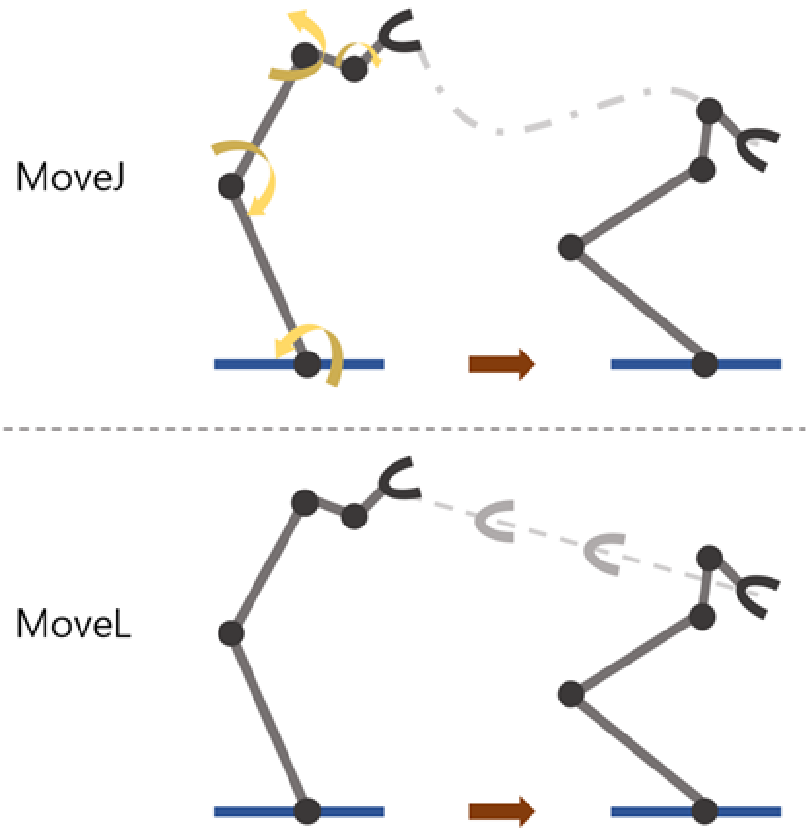
The TCP moves smoothly from the starting point (current position) between points using the Move L method without stopping.

Cartesian coordinate values (X, Y, Z, etc.) are used as input values, and blend radii or blend ratios can be set at intermediate waypoints. The speed can also be configured individually for each waypoint.

By using general points and corner points, the TCP moves smoothly along a trajectory that precisely passes through each waypoint.

Move Pro has three modes, similar to Move PB: Constant, Intended, and Smooth Modes.

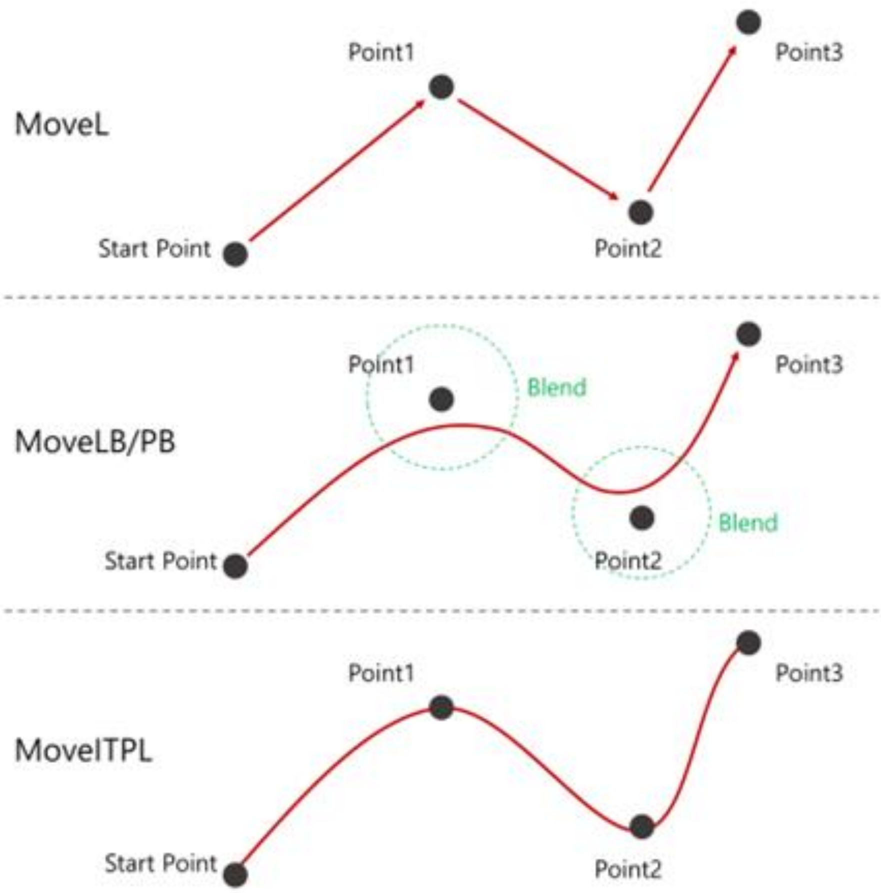
- Difference between Move J and Move L



Move J does not consider the movement trajectory of the terminal (TCP). It is an operation that only uses the joint angle information of the starting point and the joint angle of the target point. The driving speed of other joints are adjusted to the joints that require the most driving time.

Move L is a mode that uses inverse kinematics to move the trajectory of the terminal (TCP) linearly from the starting point to the target point. 6 Cartesian coordinate values (x, y, z, Rx, Ry, Rz) are the inputs for the target point value.

- Difference between Move L, Move PB, and Move ITPL



Move L moves in a straight, linear path between the start and destination points. The arm will arrive at each sequential arrival Point, stop, and then continue to the next Point.

Move PB starts at the initial Point, uses each intermediate Point as a waypoint, then stops at the final Point. The arm will not stop at the specified waypoints. Instead, it will arc around each point according to the blend distance, and then continue without stopping.

Move ITPL, the points other than the arrival point move to the waypoint, creating a trajectory that passes exactly through the waypoint. The trajectory is created without stopping and a separate speed setting is possible for each waypoint.

WARNING

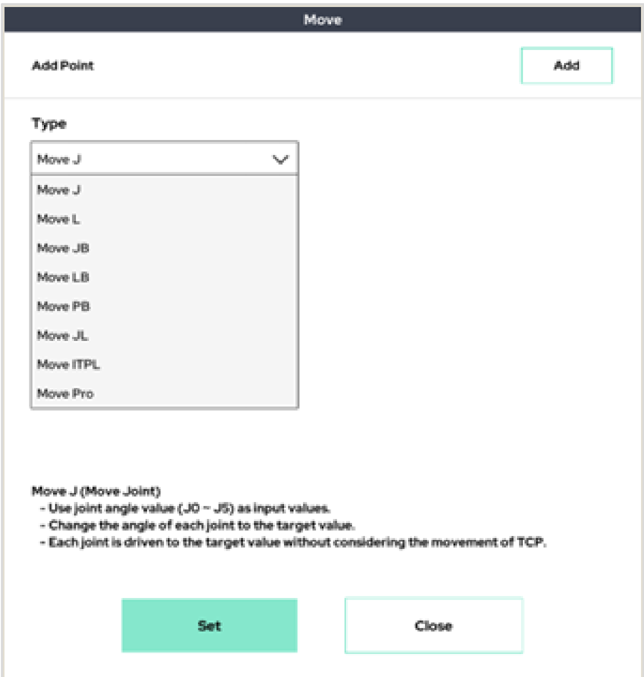
Warning:

1) The five linear motion commands (Move L, Move PB (Formerly Move LB), Move JL, Move ITPL, Move Pro) move the robot using inverse kinematics calculations. Therefore, movement may be limited in singularity positions where inverse kinematics calculations are not possible.
2) Certain joints may move faster or be restricted in motion while in the dead zone of the robot. Further information about dead zones can be found in Section 1.7.

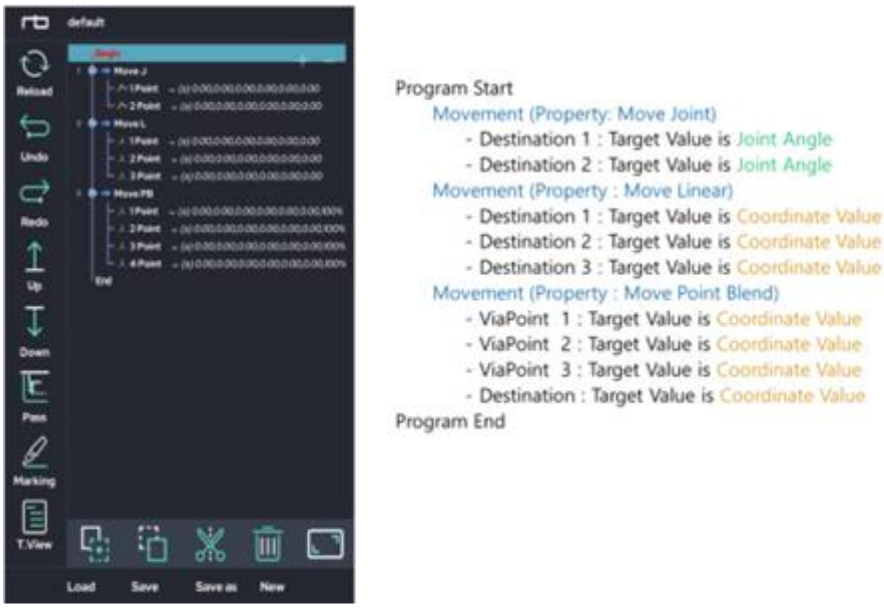
- Changing Move Function Commands

When the Move Function is used for the first time in a program, the program tree will be created as shown below. By default, the Move function is set to Move J.

Click Move J to change the Move command type. A popup will appear as shown below.

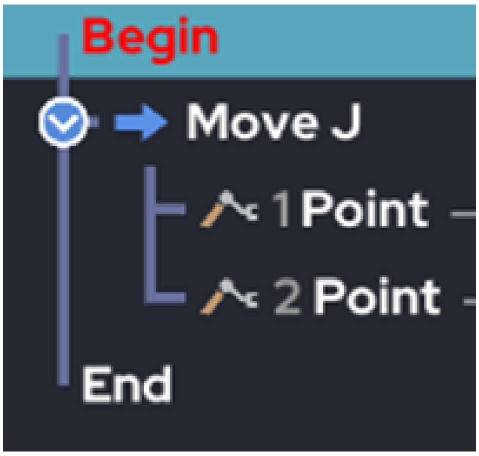


Select the desired movement type and click close to change the movement type. An example of a teaching program is shown below.



Move J, Move JB
The arm moves to the joint angle configuration contained within each Point. Each angle value is relative to the base position. ※ Since the robot arm consists of six joints, the Move J and Move JB functions will move all six joints based on the configuration contained within each Point.
Move L, Move PB (formerly Move LB), Move JL, Move ITPL, Move Pro
The arm moves relative to or directly to a target TCP position contained within each Point. Each Point determines a target location within the Cartesian coordinate system for the TCP to pass through. ※ Since the Cartesian coordinate system consists of six values (x, y, z, Rx, Ry, Rz), all six values will need to be set as subitems of Move L, Move PB, Move JL, Move ITPL and Move Pro.

■ Point Function



← Move Type

← First Target Point

← Second Target Point

As explained earlier, the Point function is a sub-function of the Move function.
Move specifies the properties of the motion, whereas Point is responsible for setting the target position.

In the Point function, the target value will vary depending on the command type of the Move function.

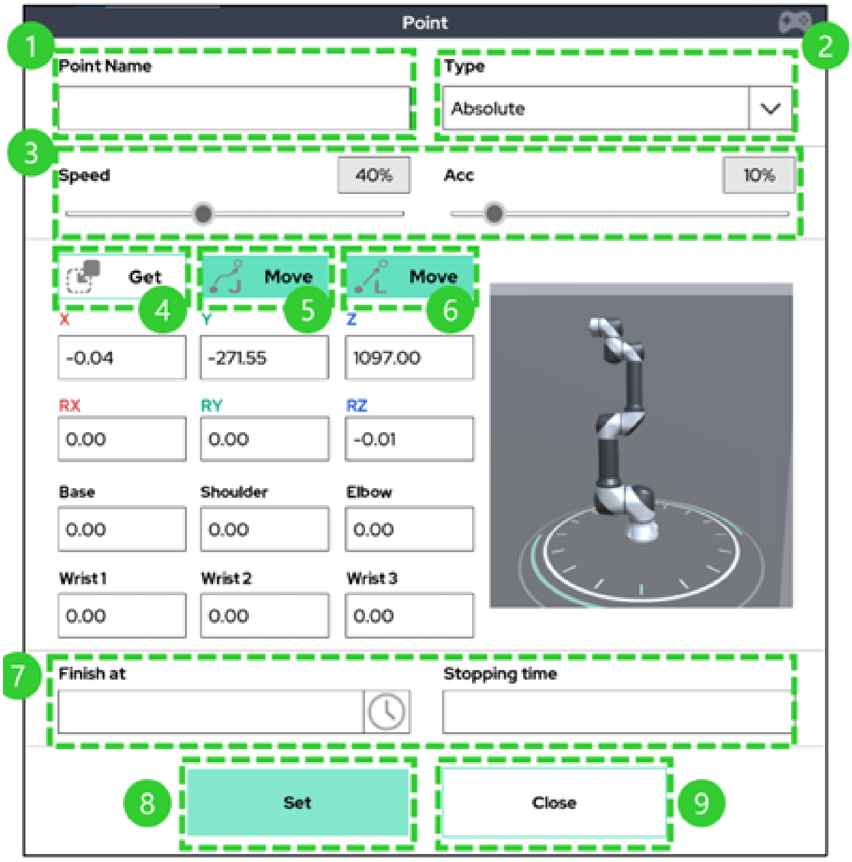
▶ Joint Movement Type(Move J, Move JB) Point:

Contains the target joint angle values (in degrees) for all six joints

▶ Linear Movement Type(Move L, Move PB (formerly Move LB), Move JL, Move ITPL, Move Pro) Point:

Contains the target destination point (in Cartesian coordinates) for the TCP

When a user taps on a Point in the program tree, the Point function popup window will appear. The window contains the following fields:



Each area is described in the table below.

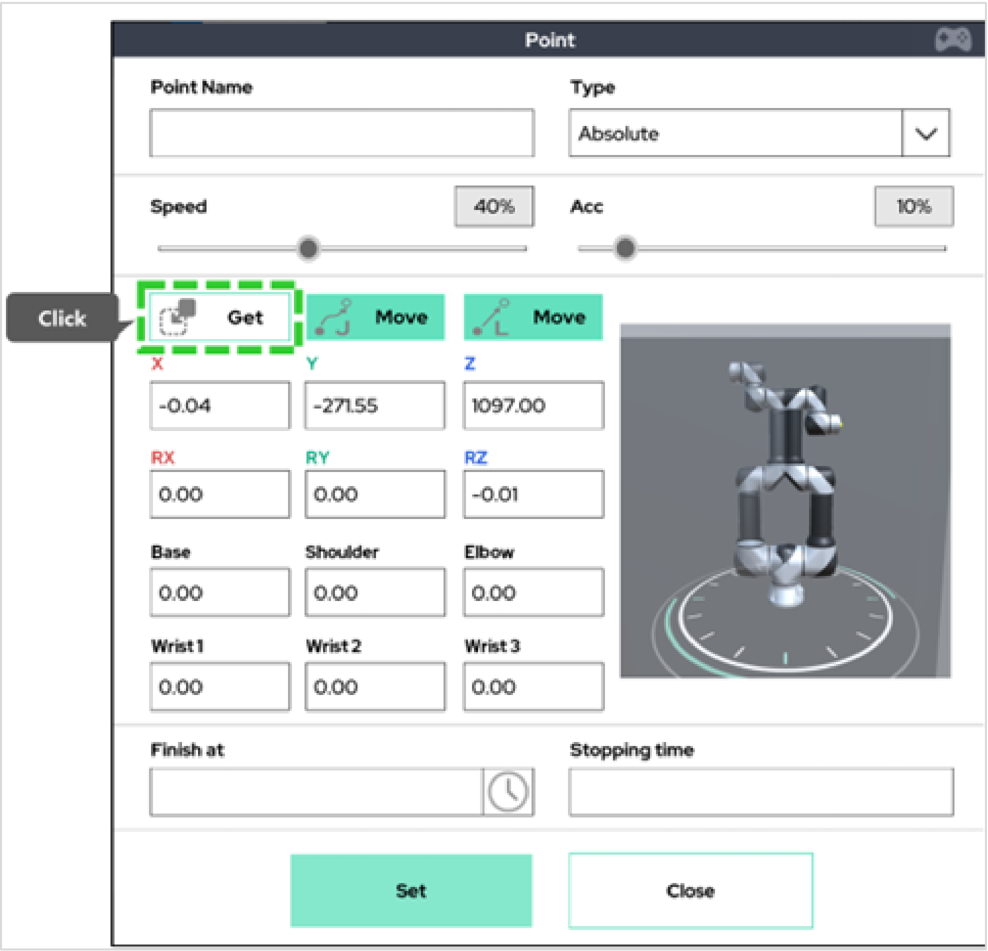
	Description
①	Sets the name of the point (not required). After setting the name, the location information of the point can be used as a variable later.
②	Allows a user to select the setting type of the point function. - The Joint Move has three setting options. - The Linear Move has four setting options. - The default type when creating a Point is 'Absolute'.
③	Sets the speed and acceleration of arm movements to the location.

④	Updates the Point information with the current robot position. - After moving the robot to the desired position/posture, press Get to store the information. - To save the Point at the current position/posture, press the Set button (Section 8). Depending on the Point type (Section 2), the Get button may or may not supported.
⑤	Moves the arm to the specified Point. - Must hold down the button to move the arm to the saved position. - The movement is a joint movement type. When the movement is completed, a pop-up message will be shown. Depending on the Point type (Section 2), button may or may not supported.
⑥	Moves the arm to the specified Point. - Must hold down the button to move the arm to the saved position. - The movement is a linear movement type. When the movement is completed, a pop-up message will be shown. Depending on the Point type (Section 2), button may or may not supported.
⑦	Specify an escape condition (Finish At) and an escape time (Stopping Time) for the action. Not a required input. - If the input is left blank, the operation will end normally after reaching the target point. - Once the escape condition is satisfied, the operation stops according to the escape time and continues to the next action. The minimum escape time is 0 seconds.
⑧	Saves the changed settings.
⑨	Closes the Settings window. Will not save user input without pressing the Set button (Section 8).

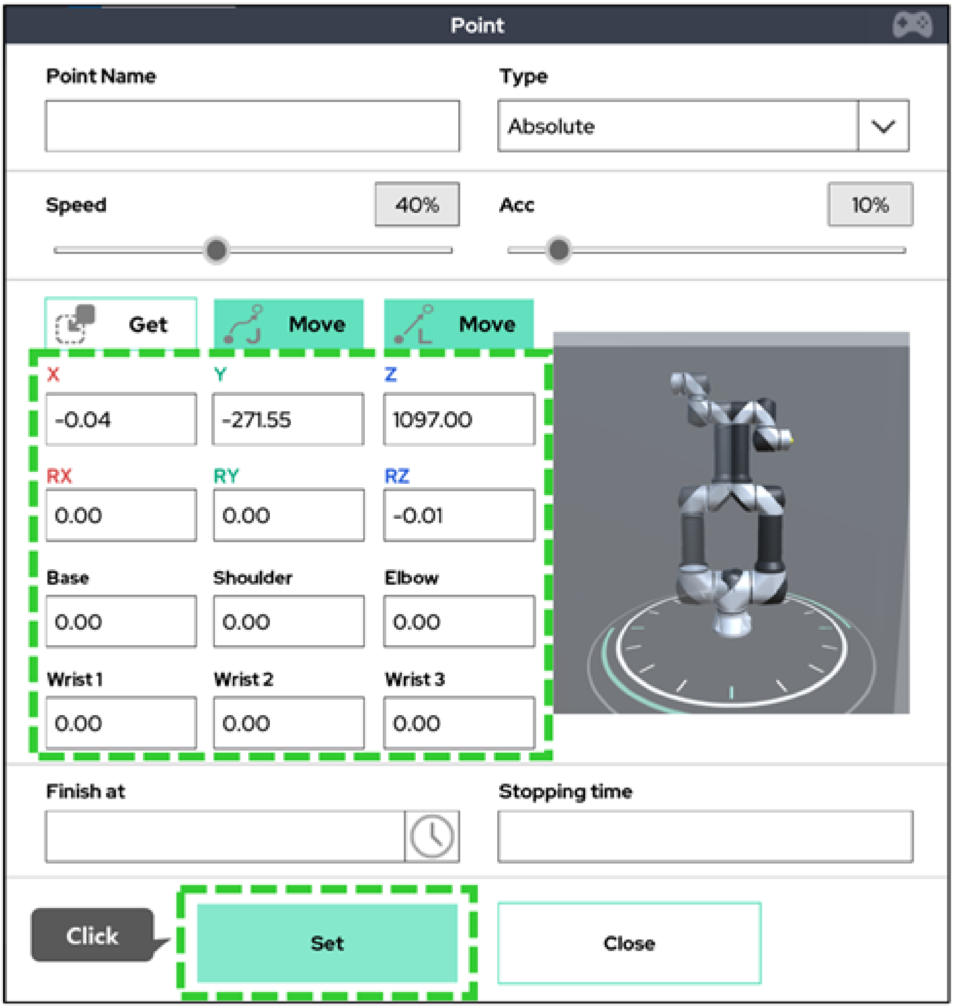
※ An example using the Get function (Section 4) is shown below.

1. Use the jog / direct teach function to move to the desired posture / position

2. Get current posture / location information by pressing Get button



3. Save after confirming reflection

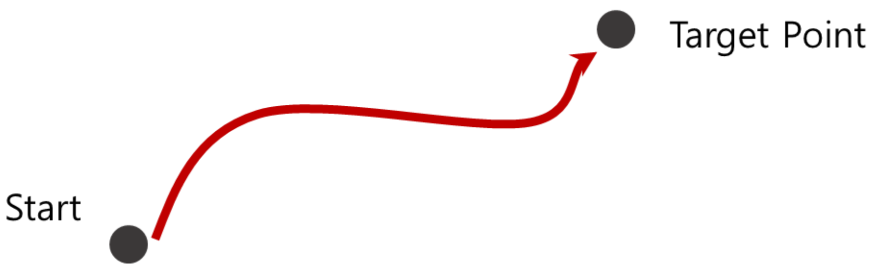


※ An example using the Finish at/Stopping time option (Section 7) is shown below.

■ When not using the Finish at function

(If left blank)

End of motion after arrival to original set target point, execute next command

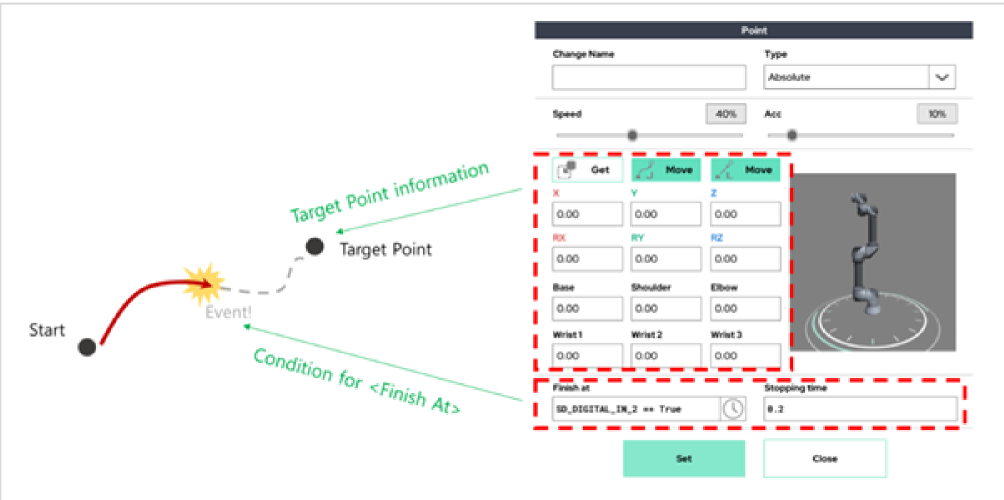


- When using the Finish at function

  (When entering a specific conditional expression)

  Even if the target point is not reached, the operation is terminated when the Finish at condition occurs and the next command is executed.

  If condition does not occur during operation, execute the next command after reaching the target point normally.



The following setting options exist for each type of move function.

Joint Move Type's sub-Point		
Option	Absolute	>Sets the Points for Move J by using fixed, user defined joint angles. >Requires the user to set the desired posture/joint angle configuration through the Get function.
	Variable	>Sets the Points for Move J by using one of several methods. >Allows the user to set the desired posture/joint angle configuration through the Get function. >The user can also change a joint angle by setting it to a variable or a mathematical operation.

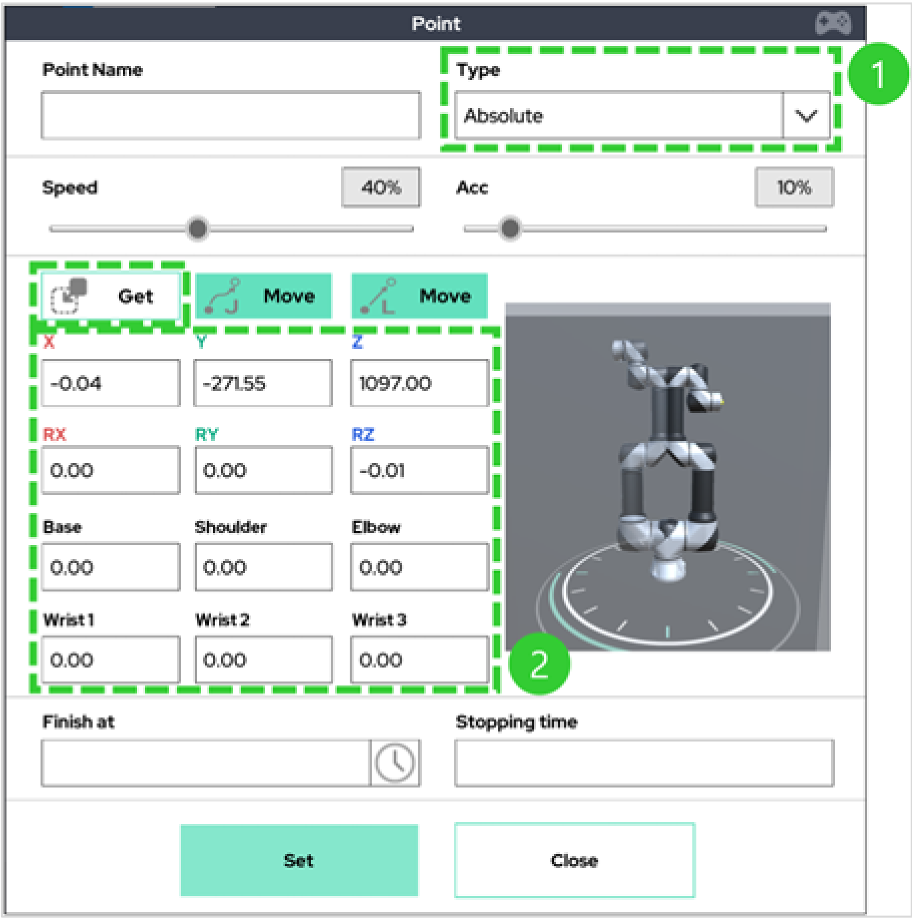
	Relative	>Sets the Points for Move J by changing the joint angles relative to the previous angle position. >If a joint movement is set to zero, then that joint will not move. If all are set to zero, then the robot will not move. >The user can also change a joint angle by entering a variable or mathematical operation

Linear Move Type's sub-Point		
Option	Absolute	>Sets the Points for Move L by using fixed, user defined Cartesian coordinate values. >After moving the robot's TCP, Cartesian coordinate values through the Get function can be set. >The default Cartesian coordinate system for the Absolute Point Type is the base coordinate system of the robot arm (manufacturer's default coordinate system).
	Variable	>Sets the Points for Move L by using one of several methods. target Cartesian coordinate value. >Allows the user to set the Points for Move L by using fixed, user defined Cartesian coordinate values. >The user can also change the TCP Point by setting it to a variable or a mathematical operation.

	Relative	>Sets the Points for Move L by setting the relative distance/offset from the previous Point. >The user can also choose a user defined Reference Point from which to move. The default value is PT_LAST_TCP, which indicates the last arrival point. >In Reference Frame, the user can specify which coordinate system use for relative movement. >The default value is Frame_Base, which represents the base coordinate system of the robot arm. >The user can choose change s to the user coordinate system or the tool's local coordinate system. >In addition, the user can set a point by using a variable or a mathematical operation.
	User Coordinate	>Similar to Variable, but sets a target point based on a user defined coordinate system. >Allows the user to select the user coordinate system as a reference by setting the Reference Frame. >Select the desired reference coordinate system and use the Get function to automatically enter the robot's pose/position information based on the selected coordinate system. >For example, if the user’s coordinate system 0 is selected and 0 is entered in all Cartesian coordinate values, TCP moves to the origin of the user coordinate system. >In addition, the user can set a point by using a variable or a mathematical operation.

The figure below shows each different type of Point as it displays in the UI.

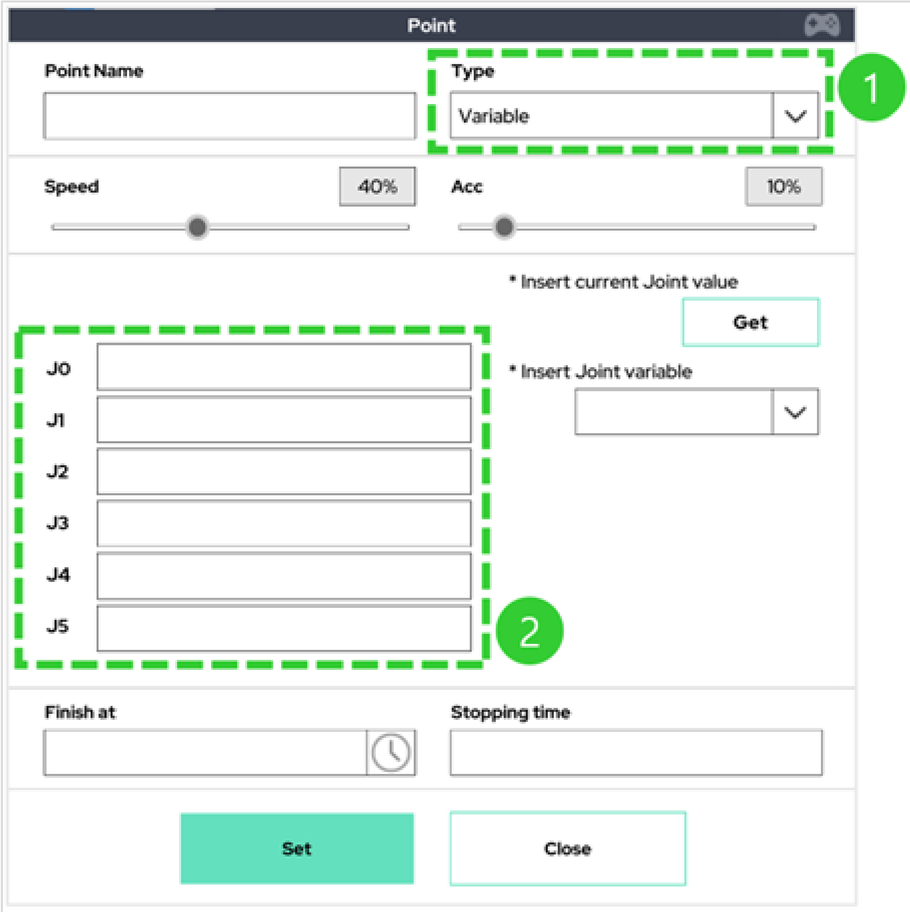
▶ Joint Type - Absolute point



① Absolute Option point.

② The robot's posture/angle value is saved through Get button.

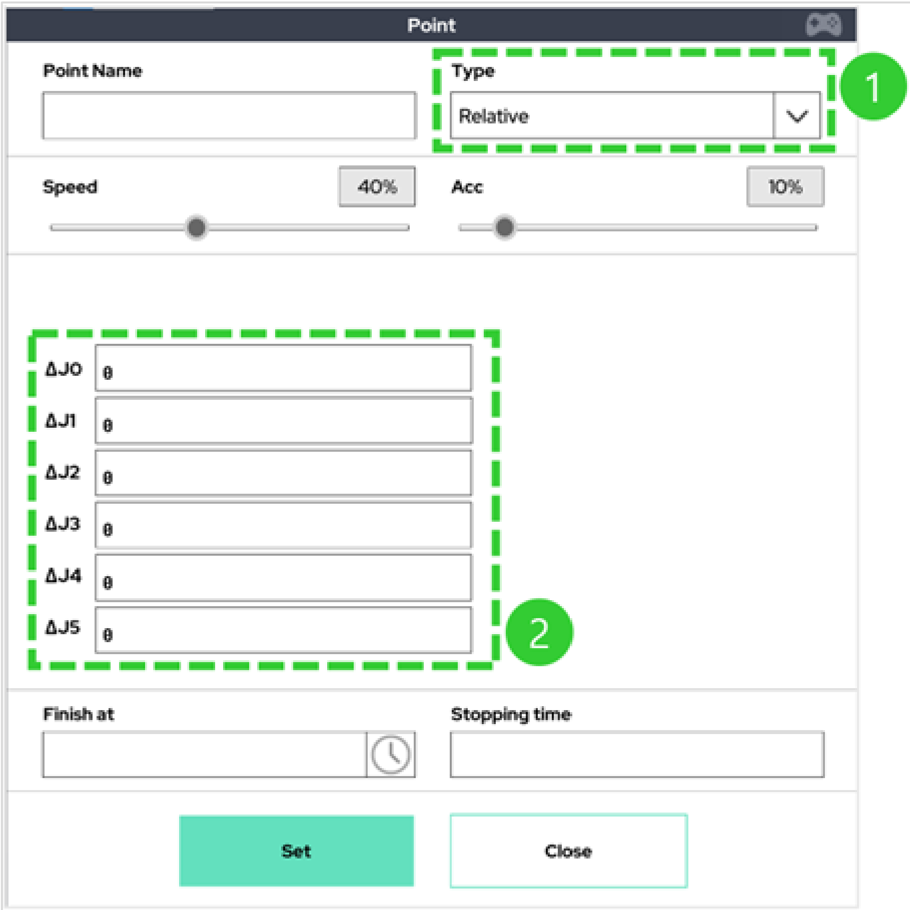
▶ Joint Type - Variable point



① Variable Option point.

② Allows the user to enter the joint angle for the target posture or enter the parameterized information as an equation.

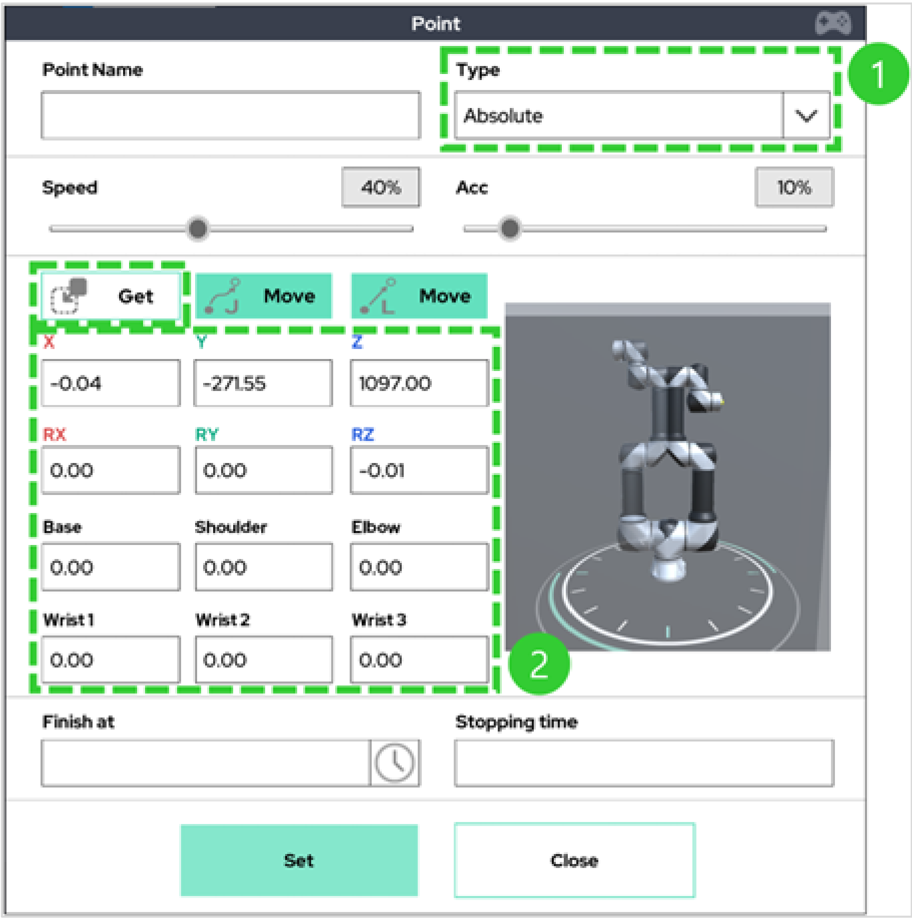
▶ Joint Type - Relative point



① Relative Option point.

② Allows the user to enter how much each joint should move relative to the previous joint angle. All angles are in degrees. In addition, it allows the user to enter parameterized information or formulas.

▶ Linear Type - Absolute point



① Absolute Option point.

② Allows the user to save a posture/position by using the Get/Save button. The reference coordinate system of the Cartesian coordinate system value is the robot base coordinate system.

▶ Linear Type - Variable point

① Variable Option point.

② Allows the user to enter the target Cartesian coordinate values. The user can also enter parameterized information as formulas. The reference coordinate system of the set Cartesian coordinate values is the base coordinate system of the robot arm.

▶ Linear Type - Relative point

① Relative Option point.

② Requires the user to enter the distance/angle offset relative to the reference point. Also allows users to enter variable information.

③ Allows a user to select a user defined point from which to move. The default value is PT_LAST_TCP, which indicates the last arrival point.

④ Chooses a coordinate system to specify relative movement. The default value is Frame Base, which represents the base coordinate system of the robot arm. The user is also able to choose the user coordinate system or the tool's local coordinate system.

▶ Linear Type - User coordinate point

① User Coordinate Option point.

② The User Coordinate Option is similar to Variable, but it allows the user to set the target point based on a previously defined user coordinate system. Users can also enter variable information.

③ Selection box for the user coordinate system that the user would like to use as a reference.

④ The Get button will load in the robot's current posture/position information based on the selected coordinate system.

WARNING

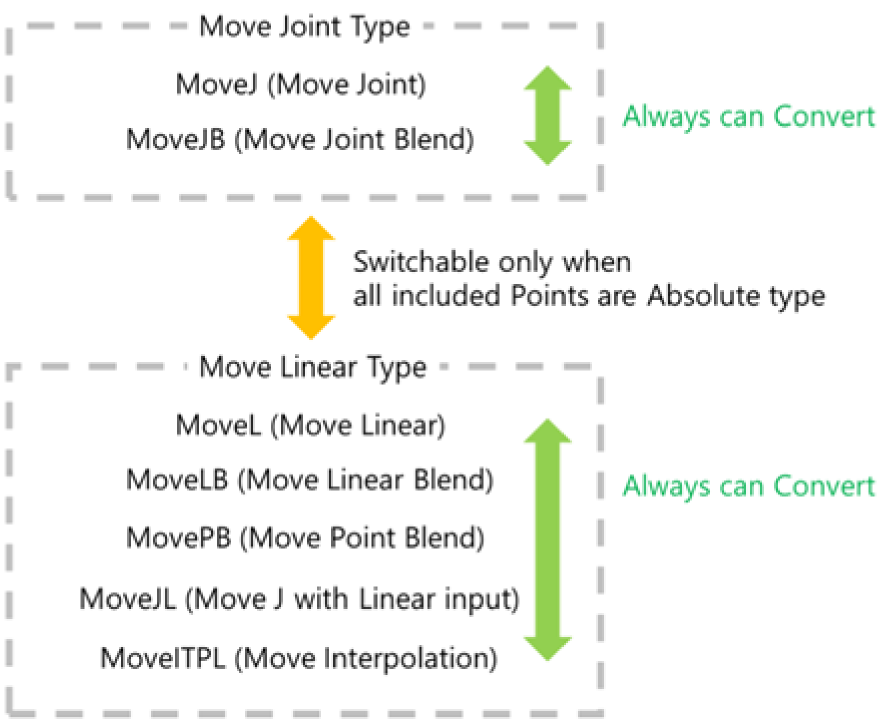
Warning:

1) A user coordinate system can be set through the Coordinate menu in the Setup screen or by using the Setting function in the Make screen.
2) Up to 3 user coordinate systems can be set and used.
3) The factory default user coordinate system is the same coordinate system as the robot base coordinate system.

■ Changing Movement Properties

The following conditions apply when changing the action properties (type of move) of a configured action.

- Switching in the same series can be done without any restrictions.
- Switching to another types (Move Joint types -> Move Linear types / Move Linear types -> Move joint types), can be done only when the type (option) of Point function is used as Absolute.



■ Example of Basic Program Creation

The following is an example of creating and running a simple program based on the above Move and Point functions.

[**Step 1**]

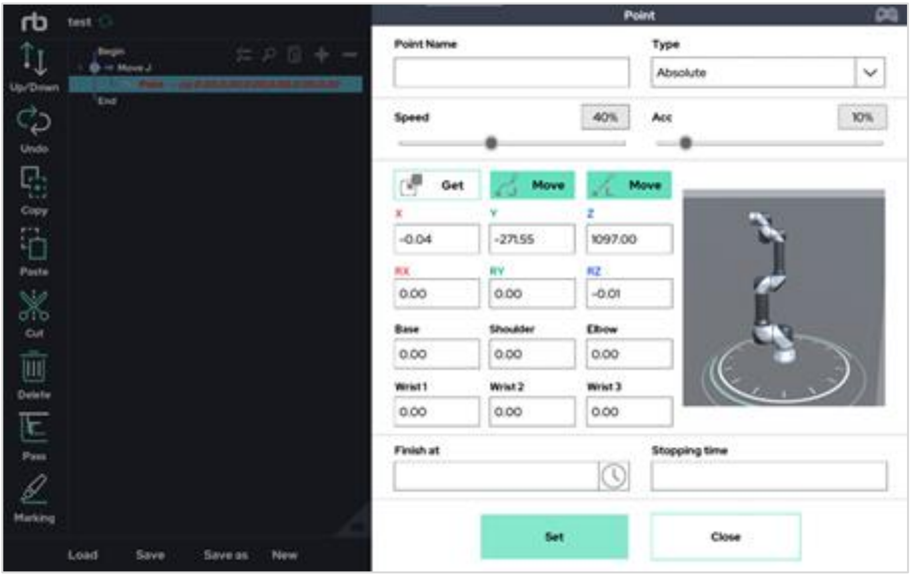
Create a new project.

[**Step 2**]

Click the Move function to add a Move command to the program tree. The default command will be Move J. A Point function will also be added to the tree as shown below.

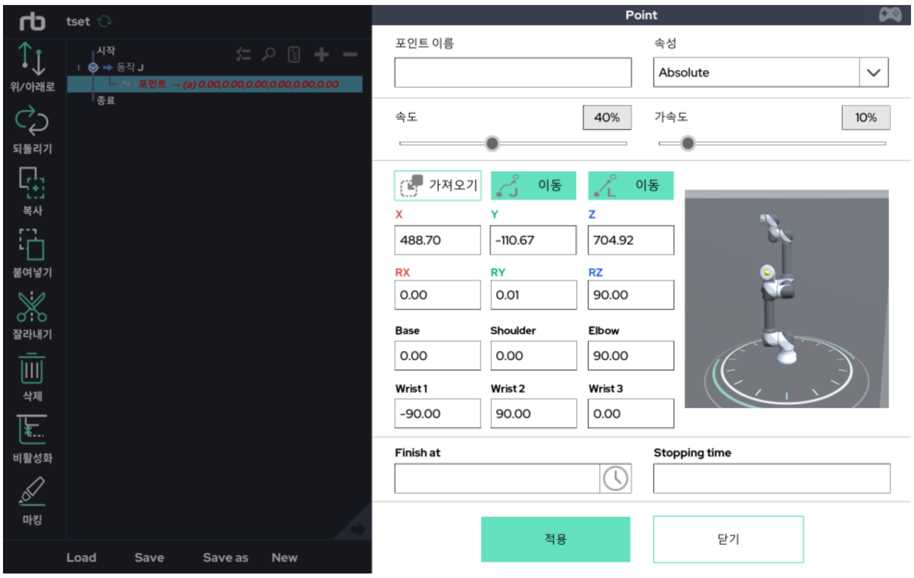
[**Step 3**]

Using the Jog button, move the robot to its intended position. In this example, the robot was moved to the following joint angle: [Base:0', Shoulder:0', Elbow:90', Wrist1:-90', Wrist2:90', Wrist3:0']. Click on a Point in the program tree to display the Point setting popup window as shown below.



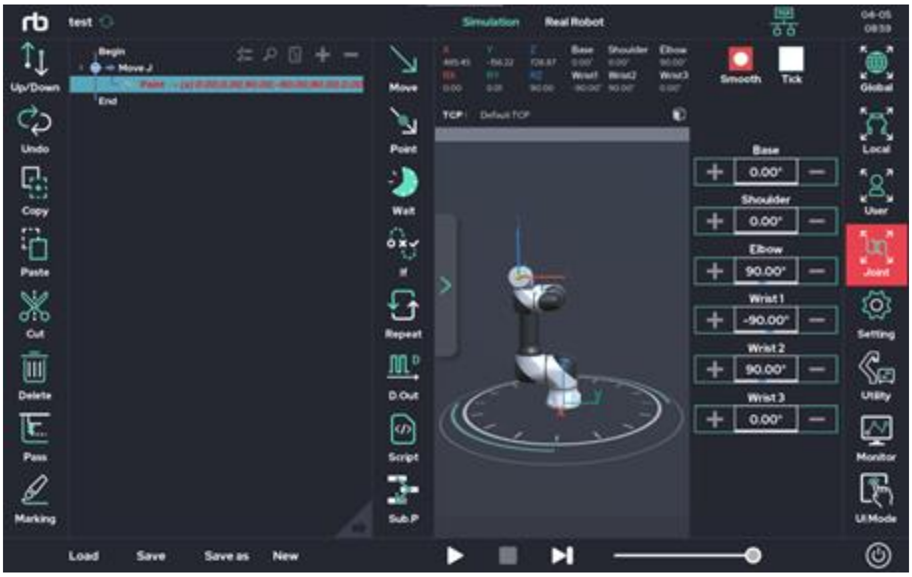
[**Step 4**]

In the Point popup window, click the Get button to update the fields with the current robot posture/angles. The posture angle that was moved at the end is taken. Press Set to save this Point.



[**Step 5**]

After saving the point, the UI will look as follows.



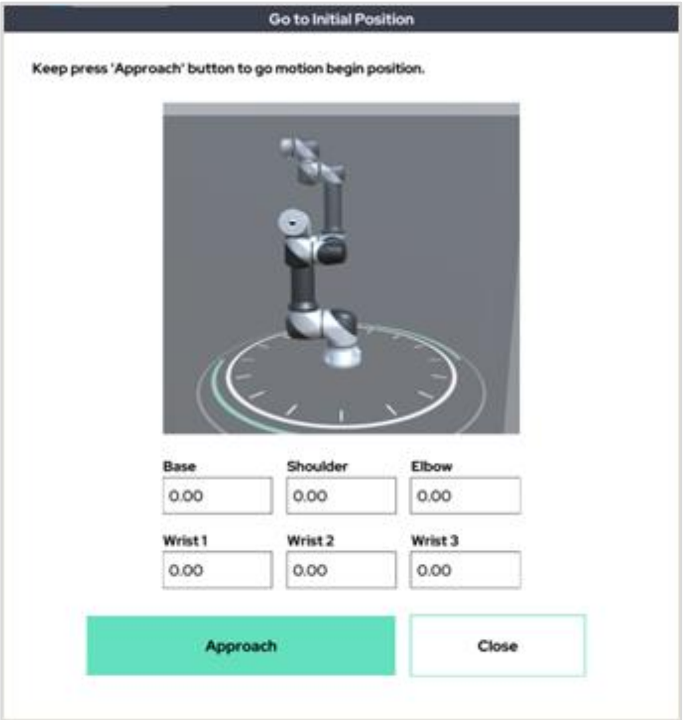
[**Step 6**]

Repeat steps 1~4 several times to teach the robot the desired motion. Our completed example program will look like the following.

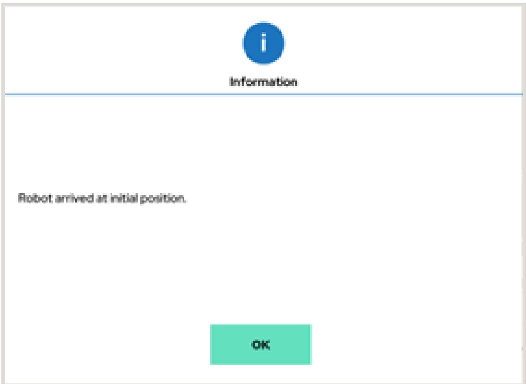
[**Step 7**]

After the program is finished, run it on the work screen by pressing the play (▷) button. To run the movements using the simulation arm, use the Simulation mode. To run the movements using the real robot arm, use Real Robot mode.

After clicking the play button (▷), the robot will move to its initial position as shown below.



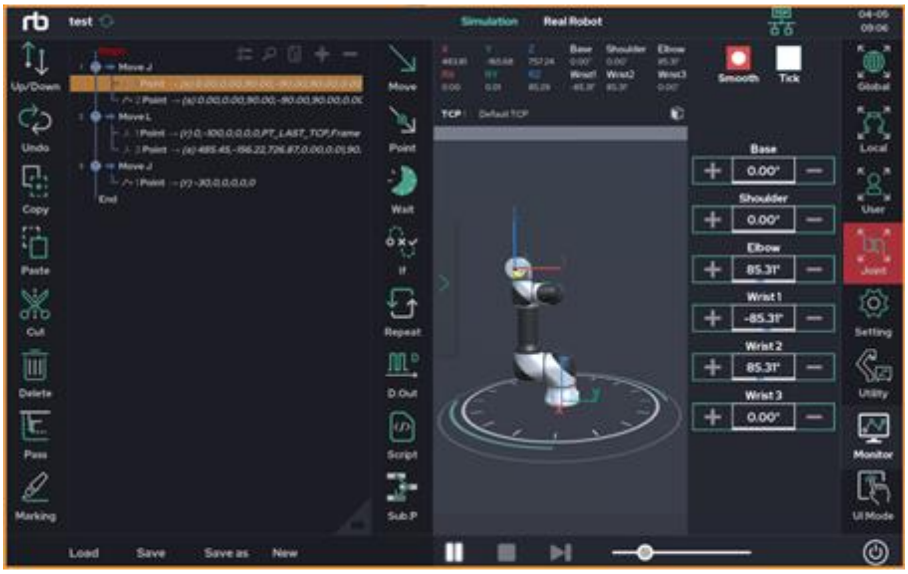
By holding down the Approach button, the robot arm will move to the initial position for the program. Once the robot reaches its starting position, a pop-up message will confirm to the user that the robot has reached its starting position.



[**Step 8**]

After receiving the popup in Step 7, the program is ready to run. Click the play button at the bottom again to run the program.

The image below shows the program running.



WARNING

Warning:

1) The Point that the robot is current moving towards will be displayed as yellow in the program tree.

■ Initial Movement Position

The initial position can be modified in the Begin section of the program.

When running a robot using the tablet UI, before running a program that contains movement, the robot must return to the initial position.

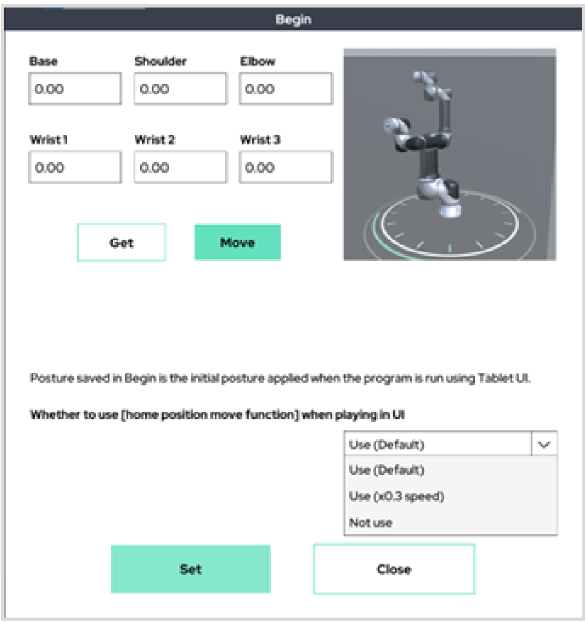
The initial position can be changed by the following way.

1) Move the robot to the desired starting position using either the Jog or teaching button.

2) Click Begin in the Program Tree to open the Begin menu.

3) Click the Get button to record the current posture, then click the Set button to save the position.

4) You can also start the program without moving to the Begin position using the disable function in the lower right corner.



WARNING

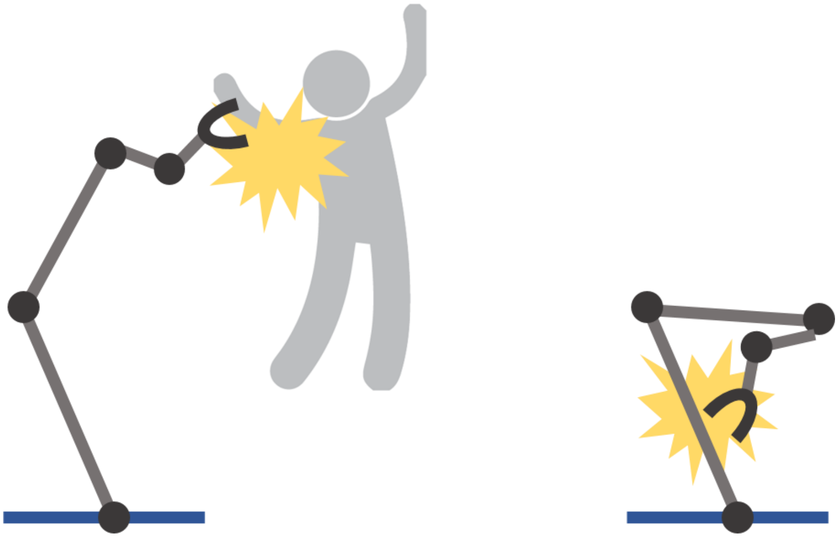
Warning:

1) When the program is first created, the default starting angles will all be set to zero.
2) The Home Position Disable feature should be used with care as it allows you to start the program in any position.

■ Collision detection during operation

The RB Series has two built-in collision detection functions:

- External Collision Detection (Out-Collision Detection)
- Internal Collision Detection (Self-Collision Detection)

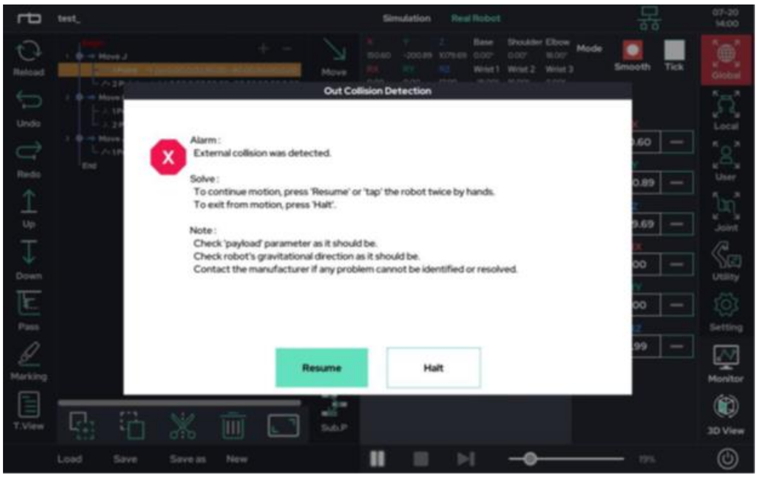


[External Collision]　　　　　　[Internal Collision]

● External Collision (Out Collision Detection)

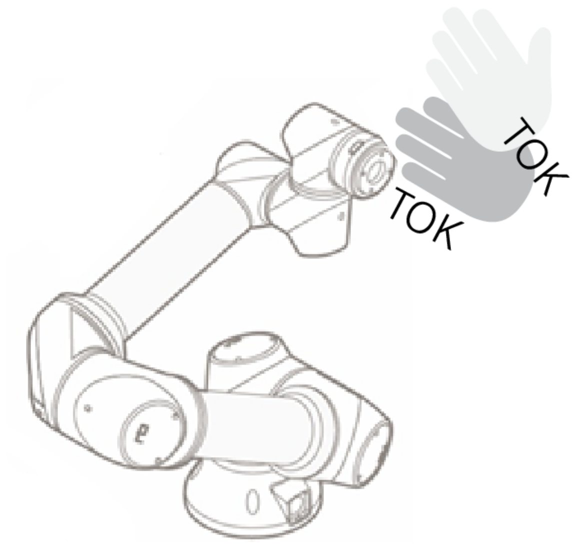
- Detects unplanned external collisions.
- Detects unexpected collisions with the environment, including people.
- Collision sensitivity can be changed in the Setup.
- The user can change the collision sensitivity in real time while the program is running through the Set function.
- For accurate collision detection, the load / center of mass of the tool should be set accurately.
- When operating with high sensitivity collision detection setting, a regular motion could be recognized as a collision due to the sudden acceleration / deceleration of the robot.

If the robot arm detects an external collision while in real mode, the following pop-up will appear.



- Resume: Checks the status and continues robot operation.
- Halt: Exits the program.

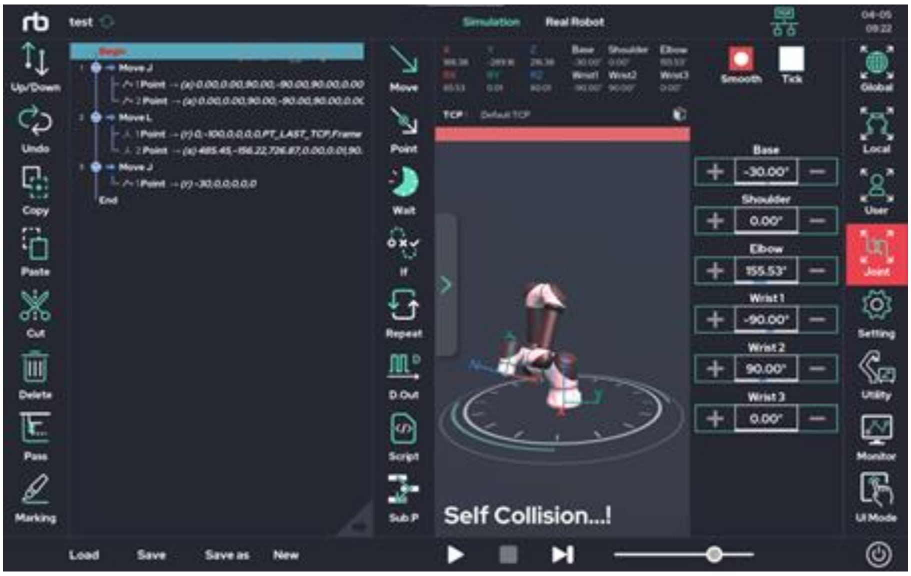
Alternatively, tap (hit) on the robot arm twice to continue the operation. This will perform the same function as the Resume button.



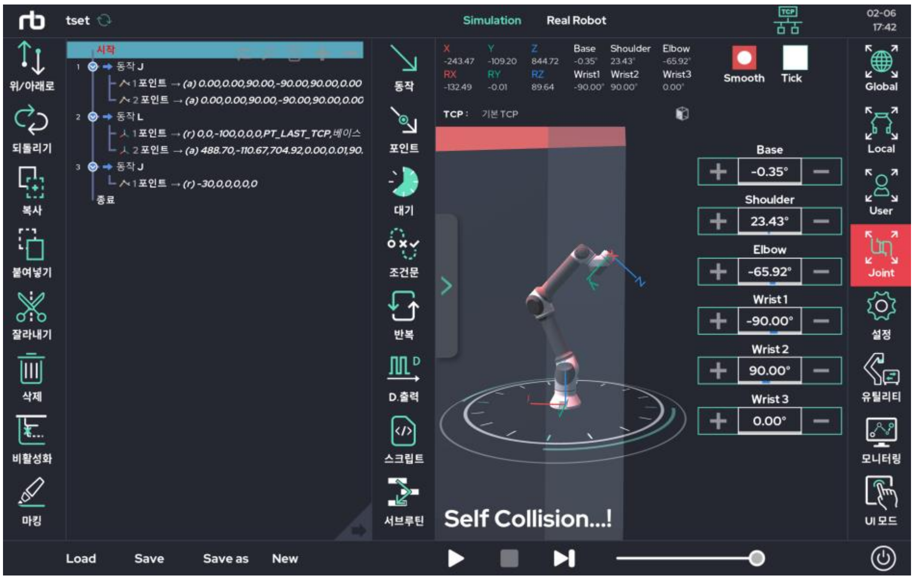
- Internal Collision Detection (Self-Collision Detection)

  - Occurs when the robot predicts that it will collide with itself.
  - If the robot extends beyond the preset Workspace limits, it will stop by itself. The setup for the surrounding environment area is done in the Setup screen.
  - Users can also set a virtual box for collision detection. This will cause the robot to stop itself if either the virtual box is expected to collide with itself or it goes out of the Workspace. The virtual box is configured in the Setup-Tool.

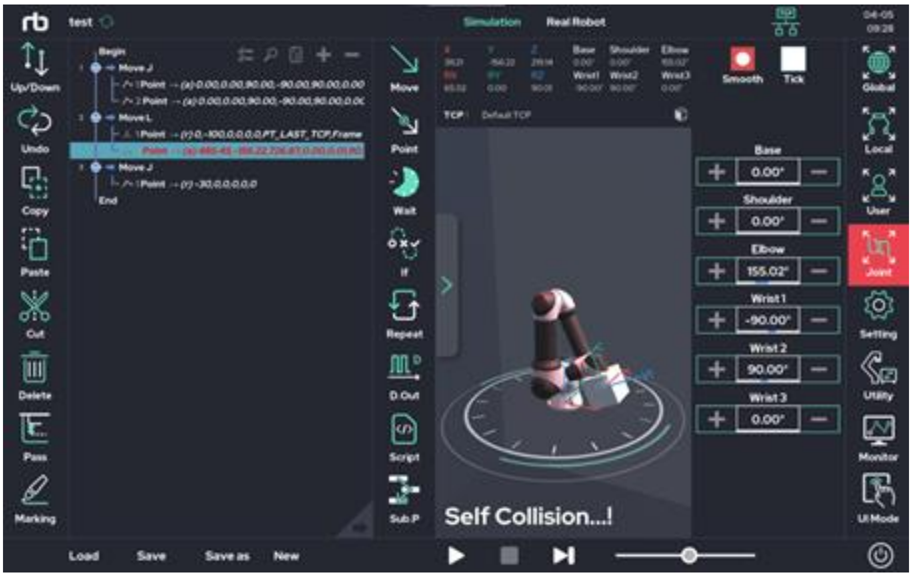
The image below shows a situation where the user caused the robot to crash into itself. Just before colliding into itself, the robot will stop, prompting the UI to display a warning in red.



The image below shows a situation where the robot is about to leave the user-defined Workspace. Just before leaving the Workspace, the robot will stop, prompting the UI to display a warning in red.



The image below shows a situation where the virtual collision box set up by the user detects / predicts a collision. The robot will stop, prompting the UI to display a warning in red.

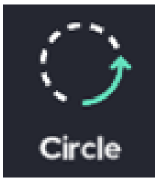


If the robot stops during operation in real mode, please move the robot arm to a safe position before continuing work.

7.4 TEACHING ICONS AND DESCRIPTION

In the previous section 6.3, only the basic teaching functions (Move and Point functions) are described. This section is dedicated to the other teaching functions.

■ Circle Function:

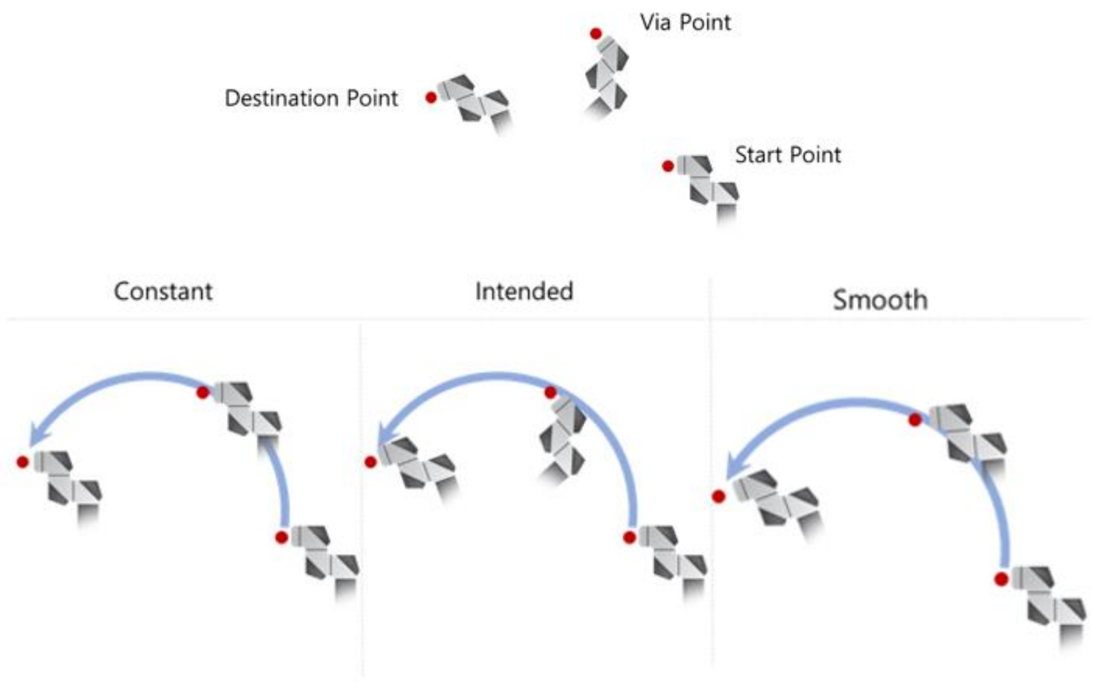


The Circle Function provides a movement method for circular motion.

There is a type that draws an arc that passes three points by taking a starting point and two points, and there is a type that can draw a circle by setting the center point and the axis of rotation.

1) Three Point Circle Type : A mode of drawing an arc with three points of information
2) Axis/Center Circle Type: Mode of drawing an arc/circle with rotation center and rotation axis information

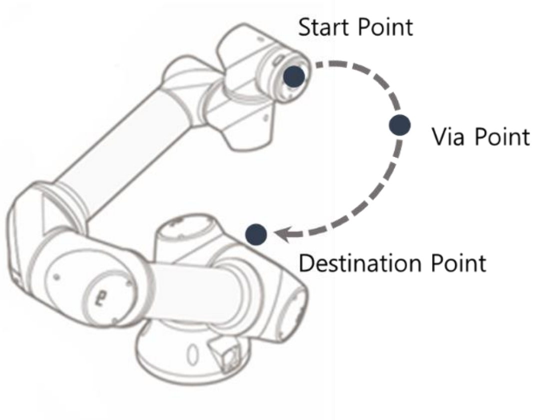
Apart from the type of circle drawing, the original operation feature provides four rotation options as follows.



- Constant: Maintains the initial TCP orientation (Rx, Ry, and Rz) of the TCP through the movements.
- Intended: The TCP rotation set by the user is followed.
- Smooth: The turn changes immediately from the start point to the destination point. The rotation information of the waypoint is ignored.

Three Point Circle Type

The Three Point Circle method draws an arc connecting three points: the starting point, the intermediate waypoint, and the arrival point.



① Circular Motion type selection (3-point setting type)

② Point type (Absolute / Variable / Relative / UserCoord.)

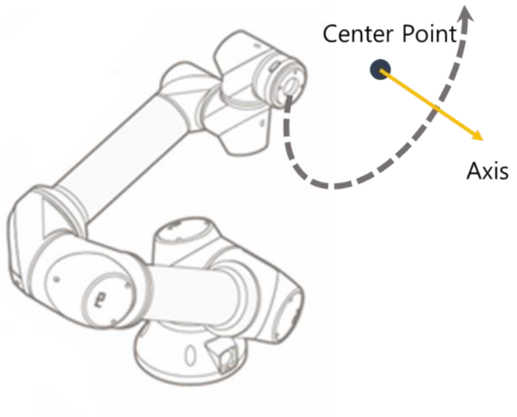
③ Orientation option (Constant / Radial / Intended / Smooth)

④ Via Point information

⑤ Destination Point information

Axis/Center Circle Type

Set the center point for the circular motion, the axis of rotation, and the angle to rotate.



① Circular motion selection (axis / center setting type)

② Point type (Absolute / Variable / Relative / UserCoord.)

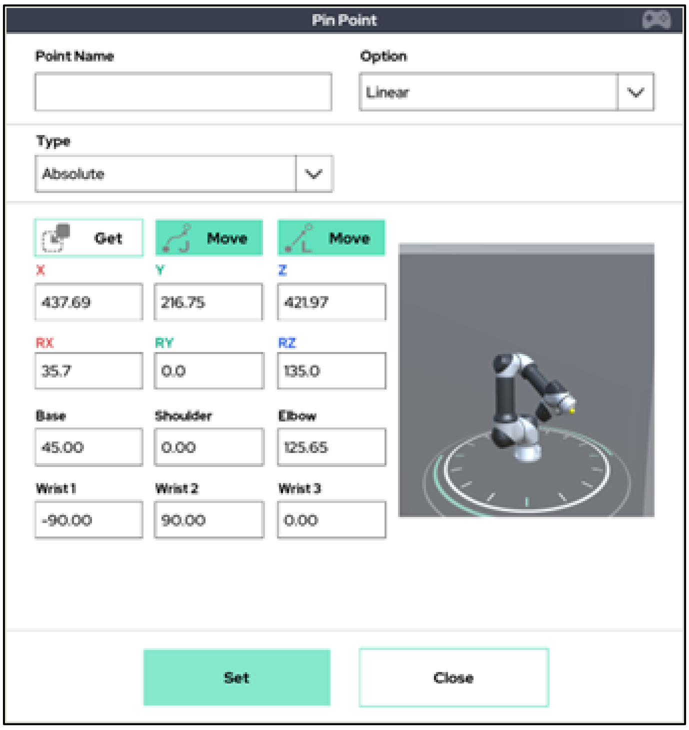
③ Orientation option (Constant / Radial / Intended / Smooth)

④ Center point information

⑤ Axis information

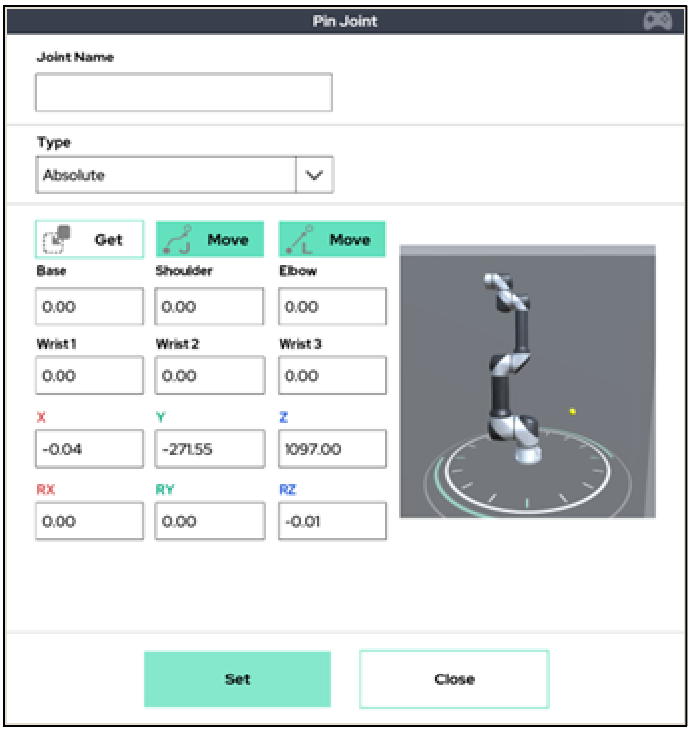
⑥ Rotation angle information

■ PinPoint Function:



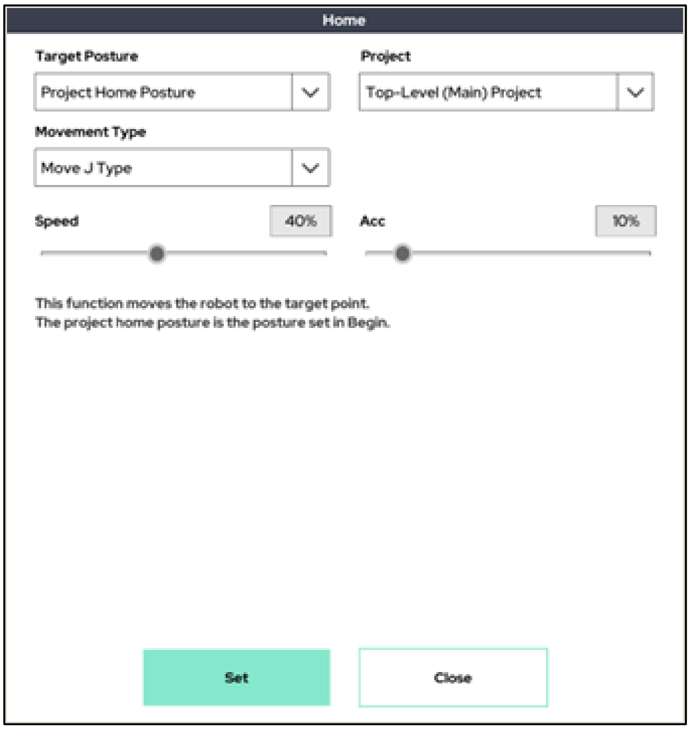
This is a special function for storing posture information only. This is a function to save information of a specific posture/position as a Point variable. If you create a PinPoint while teaching a specific posture and give it a PinPoint name, the posture information is converted into a Point variable. The information saved as Point variable can be used in other operation commands/settings.

■ PinJoint Function:



This is a function to save information of a specific posture/position as a Joint variable. If you create a PinJoint while teaching a specific posture and give it a PinJoint name, the posture information is converted into a Joint variable. The information saved as Joint variable can be used in other operation commands/settings.

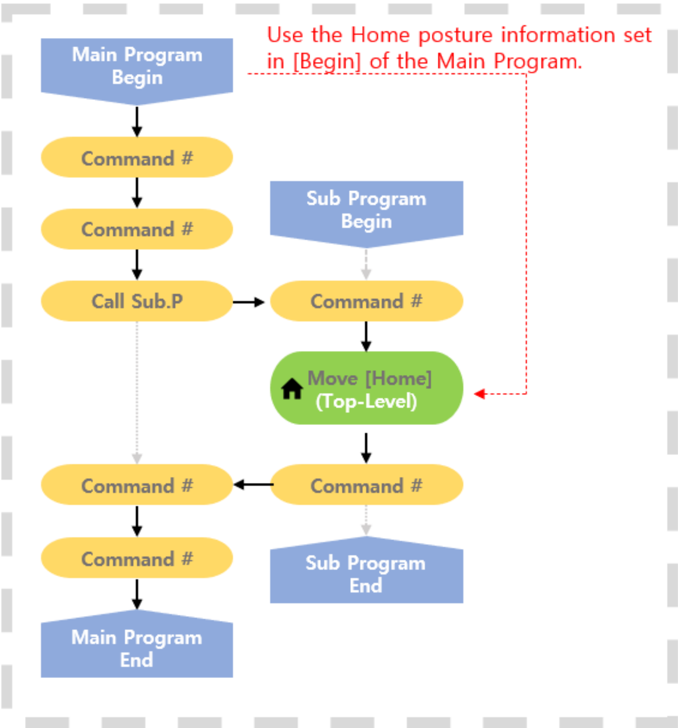
■ Home Function:

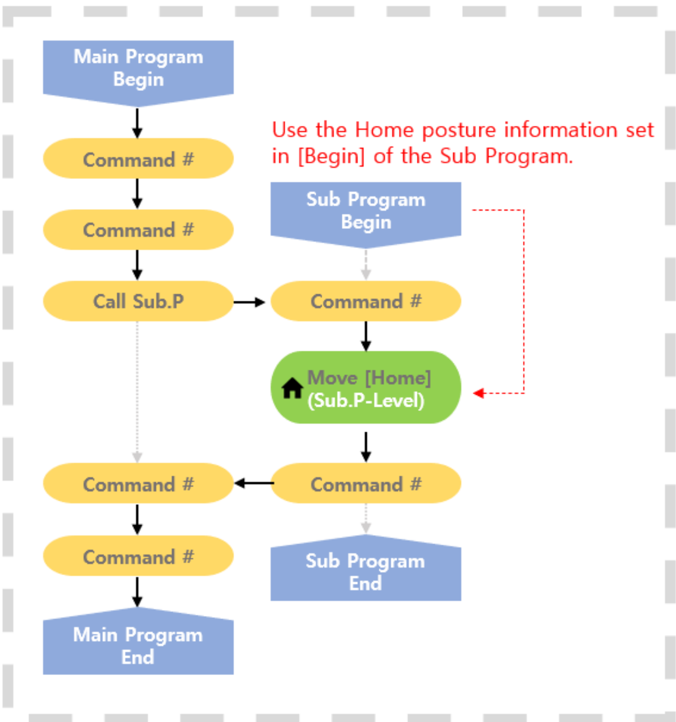


Home function is a function to move the robot with Project Home Posture or Joint Zero Posture. At this time, user can select the movement type.

When going to the Project Home Posture, user can select Project Home Posture of the main program and Project Home Posture of the subprogram.

The diagram below shows the difference between the case of going to the Project Home Posture of the main project and the case of going to the Project Home Posture of the subproject when using the home function within the subprogram.


Main Program Begin
Use the Home posture information set in [Begin] of the Main Program.
Command #
Command #
Sub Program Begin
Call Sub.P
Command #
Move [Home] (Top-Level)
Command #
Command #
Command #
Sub Program End
Main Program End

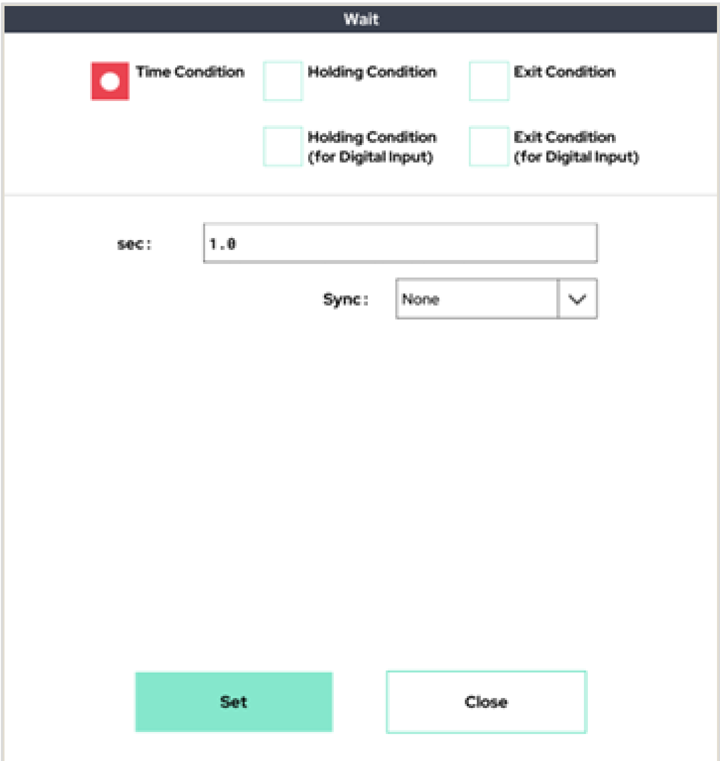

Main Program Begin
Use the Home posture information set in [Begin] of the Sub Program.
Command #
Command #
Sub Program Begin
Call Sub.P
Command #
Move [Home] (Sub.P-Level)
Command #
Command #
Command #
Sub Program End
Main Program End

■ Wait Function:

Waits for either a specified condition or a specific amount of time.

There are five modes: Wait for a specified amount of time, Wait while a condition is true, Wait until the condition evaluates to true, Ability to wait while input conditions are true, The ability to end the wait if the input conditions are true

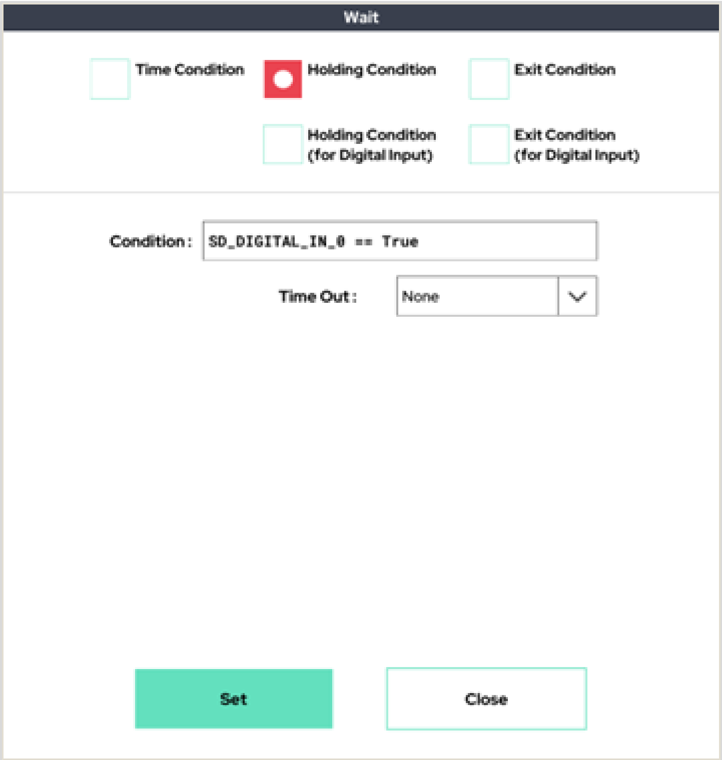
1) Wait for a specified amount of time (Time Condition)



Ex) waits for specified amount of time (i.e. 3.0 seconds), then executes the next command

When using 'sync speed control bar' function in Sync, the waiting time is adjusted in inverse proportion to the speed control bar value.

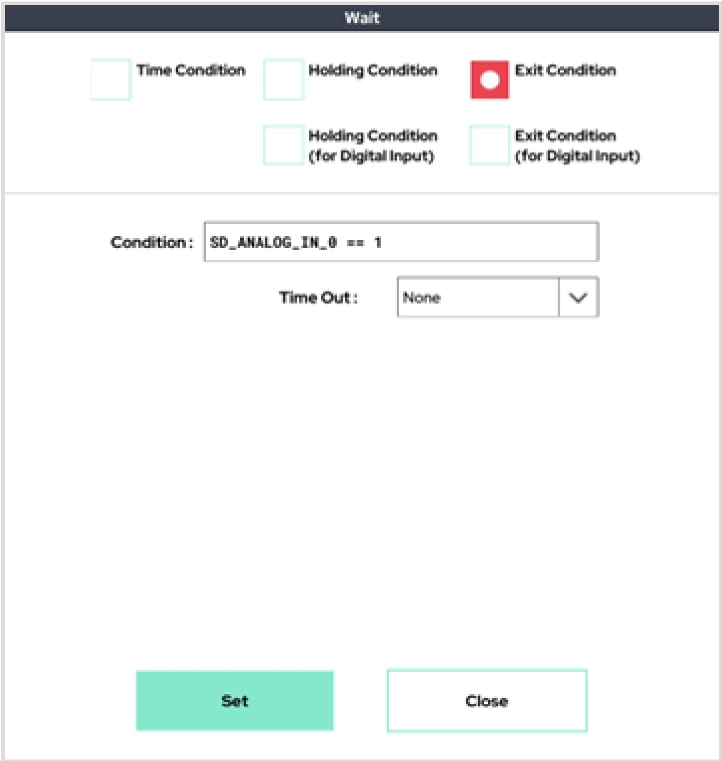
2) Wait while a condition is true (Holding Condition)



Ex) if the condition is true, the function waits indefinitely.

The Time Out function is a function to prevent the condition from continuing to wait until it becomes False in a situation where it cannot be False. Escape the wait after the written time has elapsed.

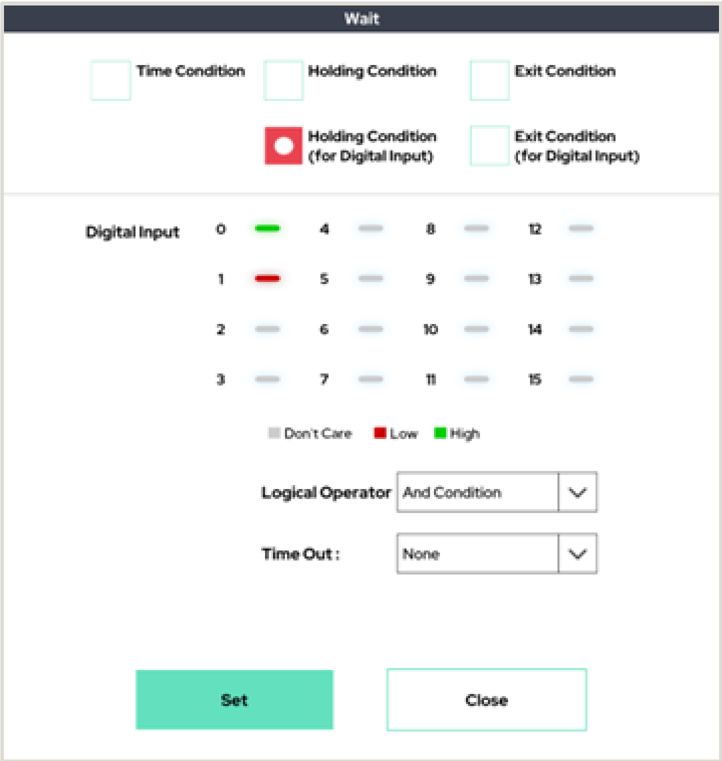
3) Wait until the condition evaluates to true (Exit Condition)



Ex) If the condition is true, the process exits the wait function and then executes the next task.

The Time Out function is a function to prevent the condition from continuing to wait until it becomes True in a situation where it cannot be True. Escape the wait after the written time has elapsed.

4) Ability to wait while input conditions are true (Holding Condition(for Digital Input))

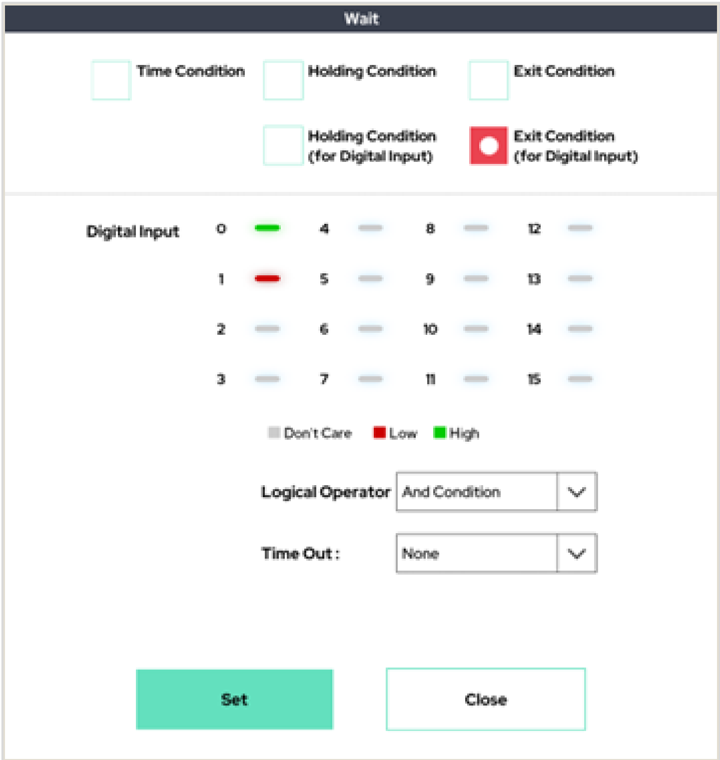


Ex) Continues waiting while the digital input conditions in the digital input window are true.

The Time Out function is used to prevent an indefinite wait if the input conditions can never become false. Once the specified time has elapsed, the wait is exited.

In the logical operator function, you can use both And and Or conditions. (In the situation shown in the image above, since it is an And condition, the system waits only when digital input 0 is in a High state and digital input 1 is in a Low state.)

5) The ability to end the wait if the input conditions are true (Exit Condition(for Digital Input))



Ex) Stops waiting and proceeds to the next command when the digital input conditions in the digital input window become true.

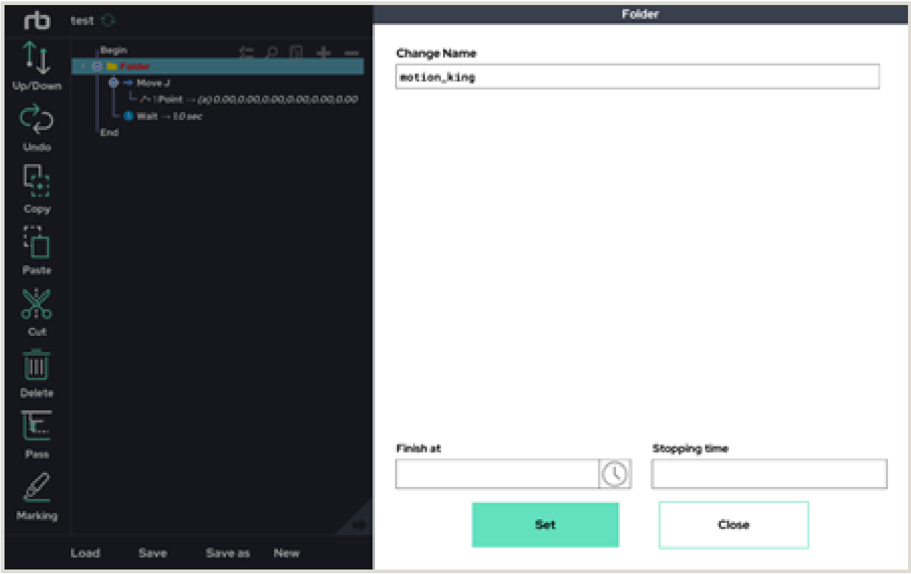
The Time Out function is used to prevent an indefinite wait if the condition can never become true. Once the specified time has elapsed, the wait is exited.

In the logical operator function, you can use both And and Or conditions. (In the situation shown in the image above, since it is an And condition, the system stops waiting and executes the next command when digital input 0 is in a High state and digital input 1 is in a Low state.)

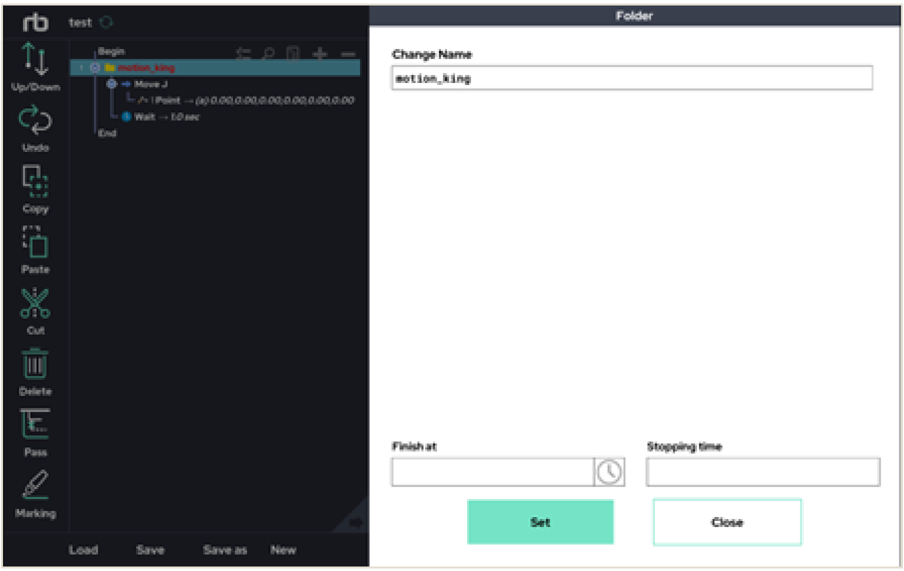
■ Folder Function:

The Folder function helps to organize commands and manage them as modules. Each Folder can contain commands as sub-items, helping with the flow of the program. Each folder can then be renamed to help provide details to the flow of the program. By clicking the Folder icon, it will be added to the program tree. Commands can then be added, as shown below.

To rename the folder, click on the new Folder in the program tree.



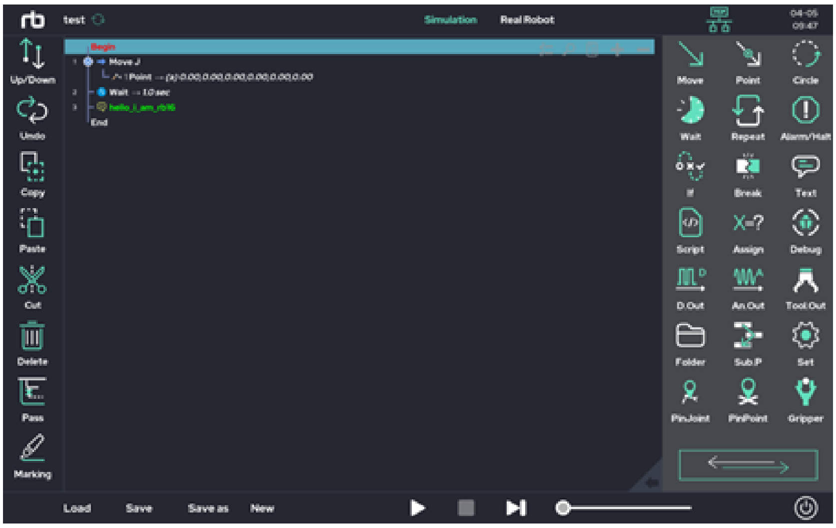
The program tree will now show the folder with its new name.

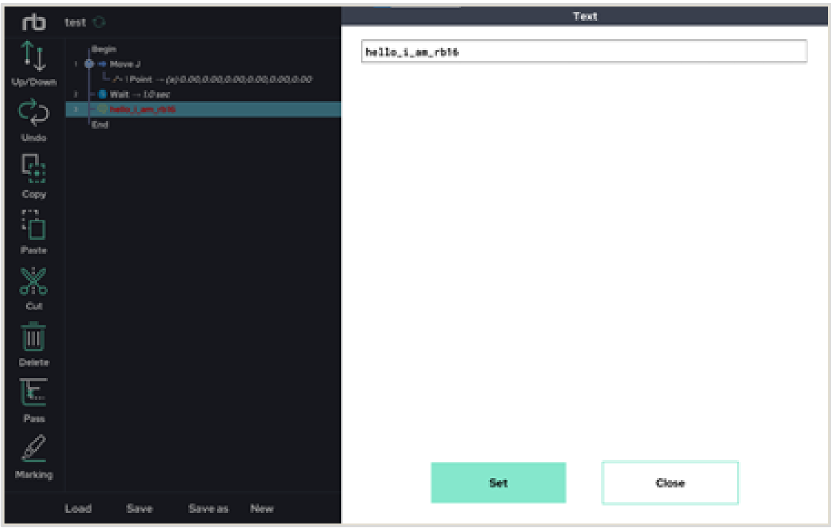


The Finish at function is available for use. It is not a mandatory input, and if you choose not to use the Finish at function, it will not affect the functionality of the program, similar to the Text function. It serves as a tool to help manage modules and groupings of functions during teaching.

■ Text Function:

The Text Function allows users to make notes/comments in the program list tree. The text function is displayed as green text in the program tree and does not affect the functionality of the program. Click the Text icon to add it to the Program Tree.

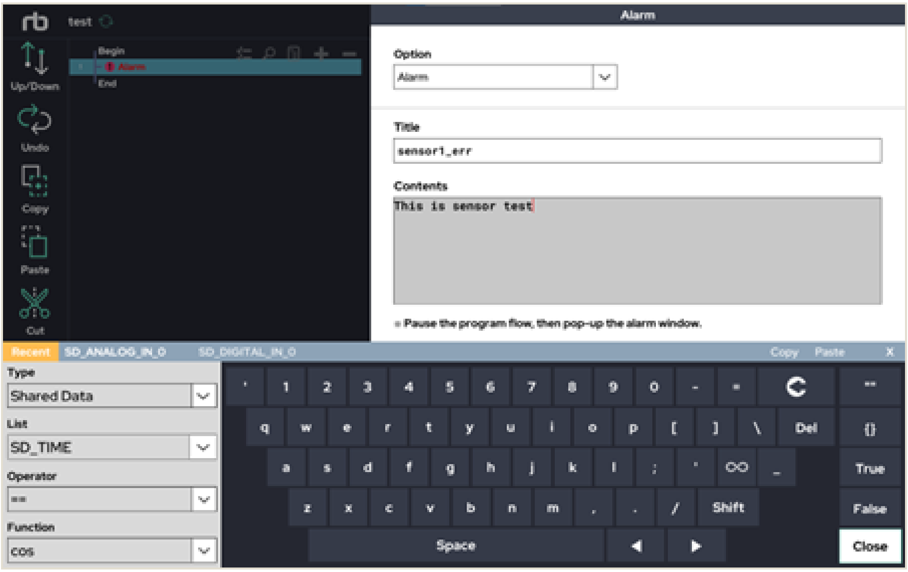




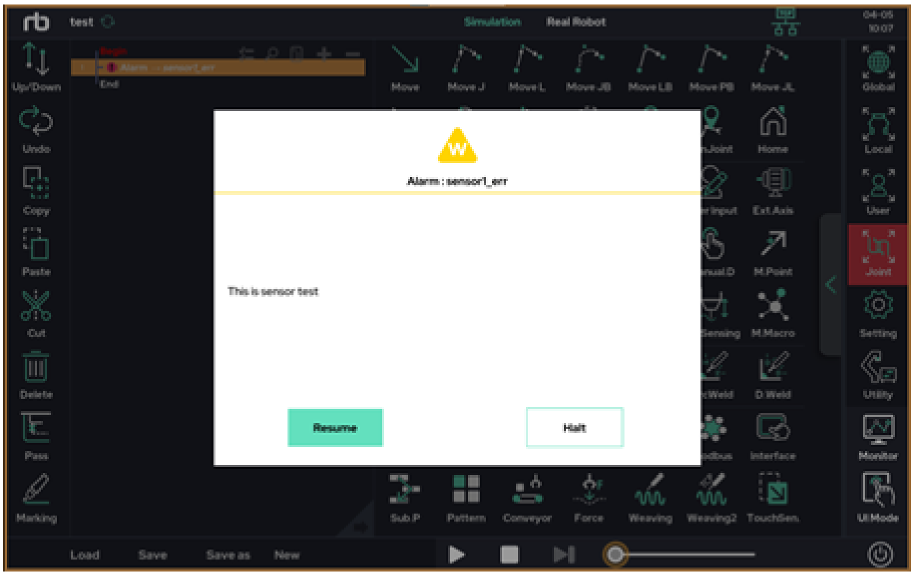
■ Alarm/Halt Function:

The Alarm function allows you to display a popup message. You can insert an alarm message at specific conditions or sections of the program to prompt the user for confirmation.

When you insert the alarm function into the program and click on it, a setup window like the one below will appear.



You can enter the title and content of the alarm message. After setting it up as shown above and running the program, a popup like the one below will appear the moment the alarm command is executed.



- Resume: Continues with the next command.
- Halt: Stops the program at this point.

While the alarm function is active, the program flow is paused. If you click the Resume button in the popup window, the next command will be executed, and if you click Halt, the program will stop running at that point.

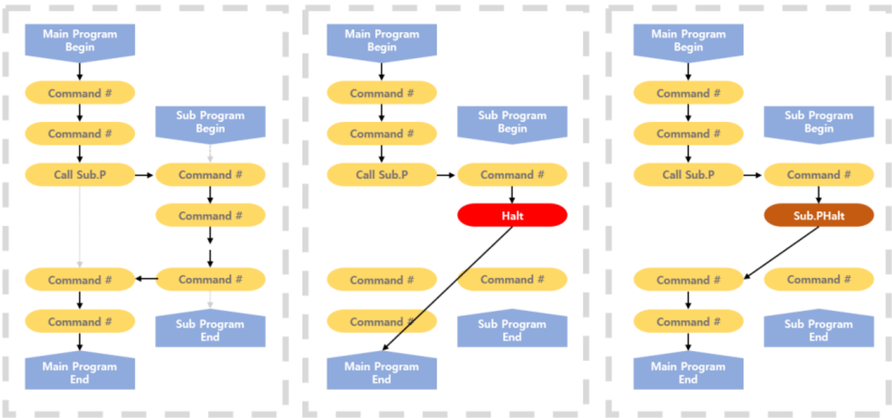
When the alarm function is used, the main program is temporarily paused, and any threads that are running are also paused at the same time.

The Halt function is used to stop the program. You can use this function when you need to terminate the program under specific conditions. The Halt function is divided into Halt, Sub.P Halt (Sub Program Halt), and Folder Halt.

Halt: Terminates the main program, regardless of whether it is executed from the main or sub-program.

Sub.P Halt: Must be used within a sub-program. It terminates the sub-program and returns to the main program.

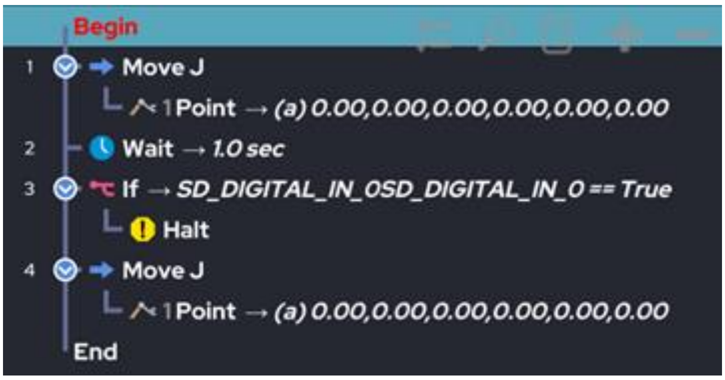
Folder Halt: Must be used within a folder. It terminates the folder and then executes the commands below the folder. Please refer to the diagram below for clarification.



common case | during the subprogram When Halt is called | during the subprogram When Sub.P Halt is called

In the example below, an If condition checks whether a certain condition is true. If the condition is met, the program is set to call the Halt function.

If the condition is true, the program will stop at that point, and the following commands will not be executed.



WARNING

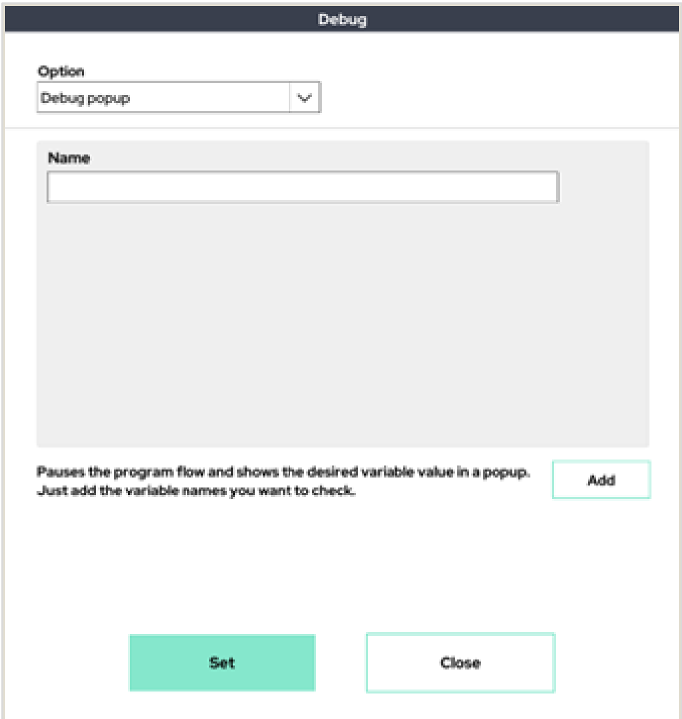
Warning:

1) When the Halt function is executed, the main program will terminate – this includes any additional Thread functions.

■ Debug Function:

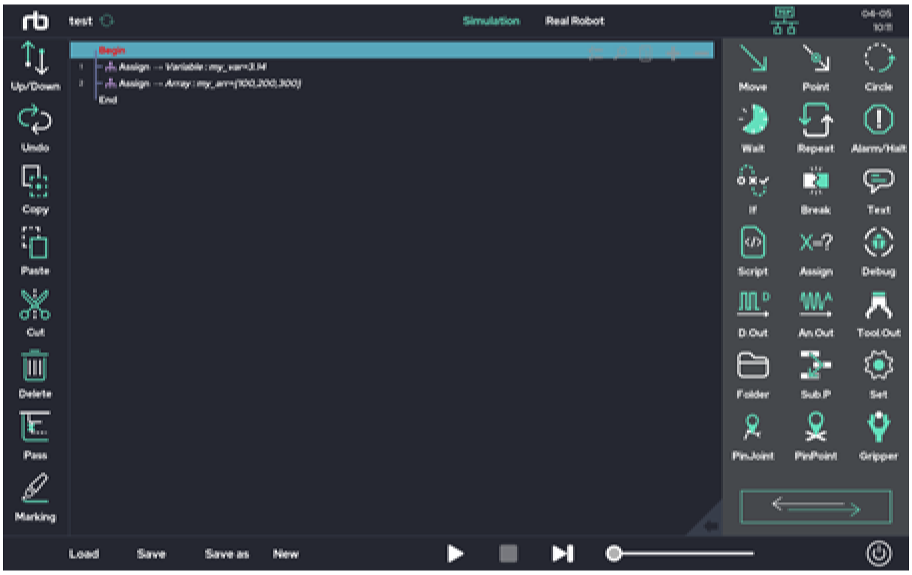
Function for debugging internal values. Users can make a pop-up display the value of a variable or internal parameter, similar to an Alarm.

Debugging is for observing internal variables. It is mainly used to check the value of variables used in the program during program teaching / development.



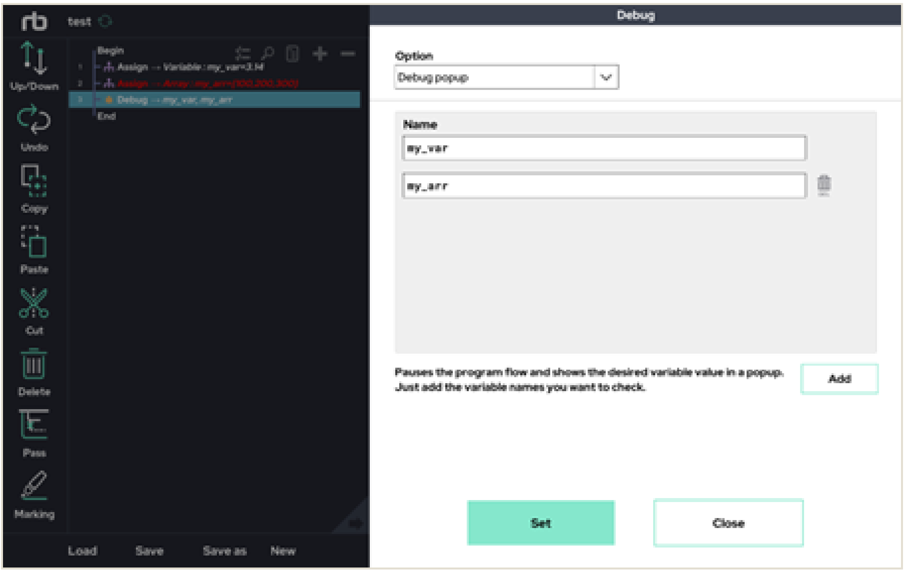
After adding the debug function to the program tree, click Debug to see the popup window as above. Enter the variable name in the Name field to view how variables change. To observe several variables within the same popup, press the (Add) button to add another variable.

The follow is an example on using Debug. Declare one variable type variable (my_var = 3.14) and one array type variable (my_arr = {100,200,300}) using the Assign function as shown below.

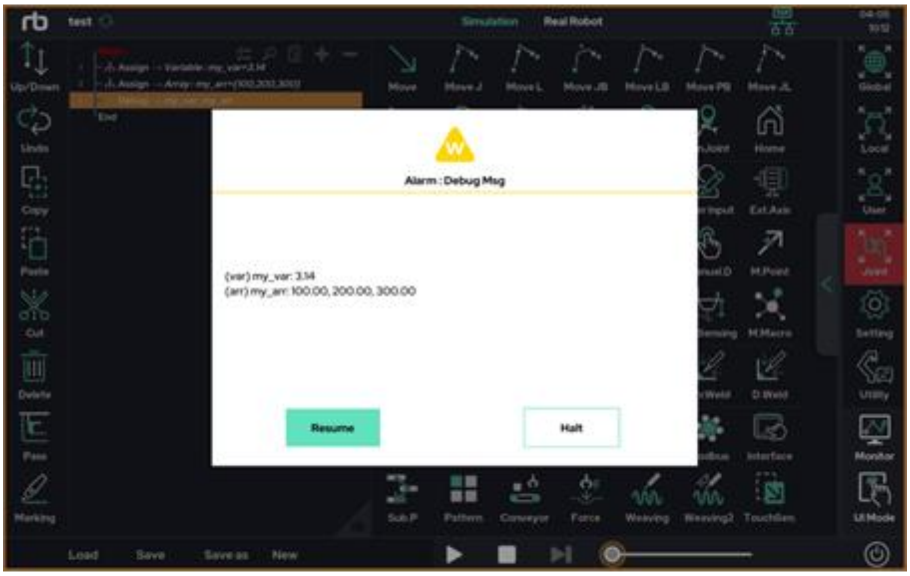


Add a Debug Function below it.

Set the variables in the Debug window to observe the two previously declared variables as shown below.

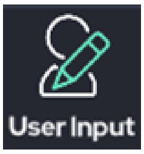


Once the setting is complete, run the program (the tablet PC and the control box must be connected before execution), and the following pop-up window will appear when the Debug command is executed. The pop-up will allow the user to observe the specified variable values.

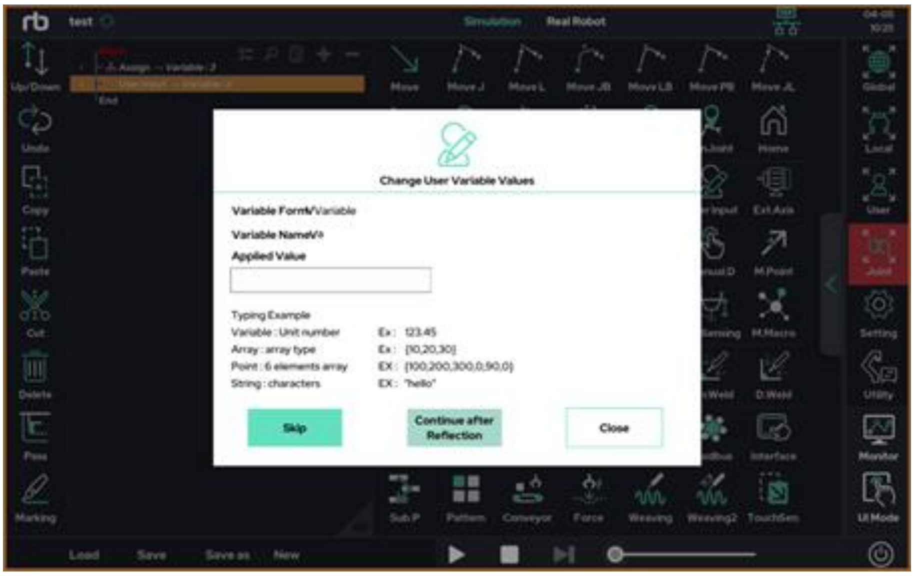


- Resume: Continues to the next command.
- Halt: Terminates the program.

■ User Input Function:



User input is used to change the value of a declared variable during program execution. As shown in the figure below, a variable is declared above the user input, and when user input is executed, a value is entered in the "Applied Value" field. At this point, if "Skip" is selected, the program continues without entering a value. If "Apply and Continue" is selected, the value entered in the "Applied Value" field is reflected in the variable, and the program proceeds.



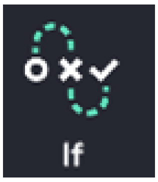
■ External Axis (Ext.Axis) Function:

You can control up to 6 external axes, and the axes can be moved based on the settings configured in the Device section of the Setup. Target values can be set for each axis, and speed and acceleration values should be input between 0 and 1.

For the option "Multi-Axis Operation," you can decide whether to finish all the external axis movements simultaneously or to complete each axis individually when multiple axes are operating.

For the option "Command Completion Condition," you can choose whether to wait for the command to finish executing before continuing or to proceed without waiting for the motion to complete.

■ If Function:



The If Function allows the users to insert a conditional ‘if’ statement. Depending on the conditions, branches can be set up so that the robot can perform different commands. Users can set the If / else if / else statement.

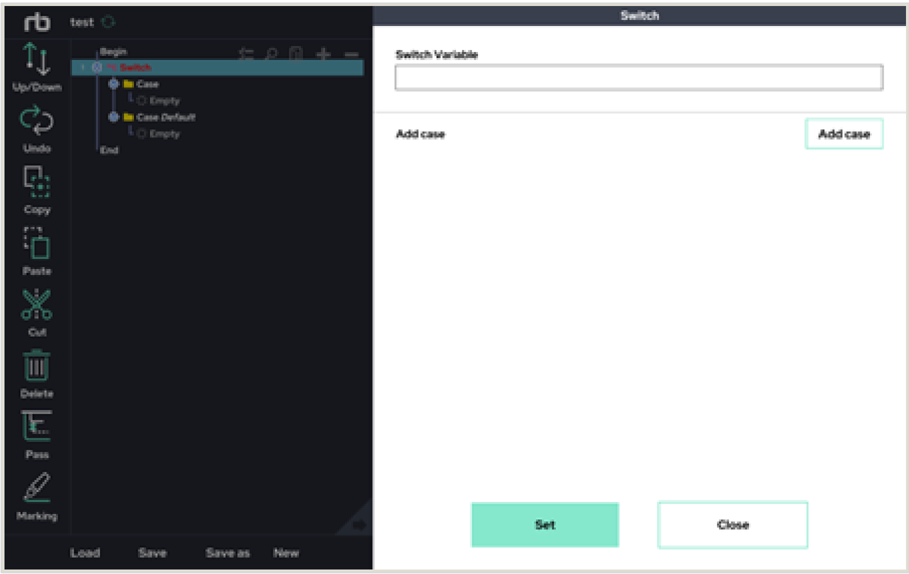
After adding the If function to the program tree and clicking the added If function, the following popup window appears. Users can enter the conditional statement they would like to use in the If statement.

Else if (+ Add else if) or Else functionality (+ Add else) can be created along with branch of conditional statements.

■ Switch Function:

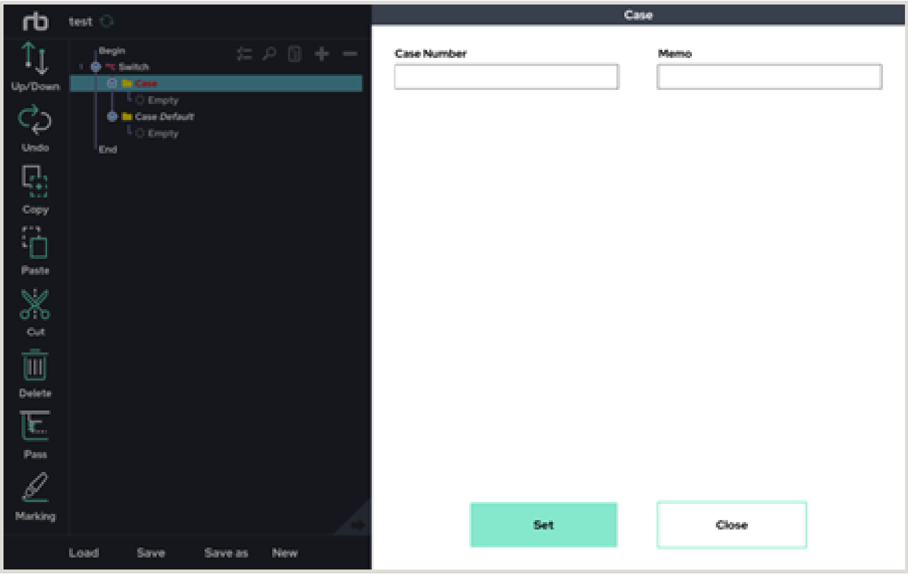
Switch statement. Depending on the conditions, branches can be set up so that the robot can perform different commands. Switch / case statements are available.

The following popup window appears by clicking the added switch function in the program tree. Users than can enter the criteria arguments for the Switch statement to work.



When first creating a Switch statement, 'default' will automatically be created. Additional case statements can then be added using the (+ Add case) button.

After clicking the (+ Add case) button, the following window will appear. Enter the conditional argument in the field, then press the Set button to save.

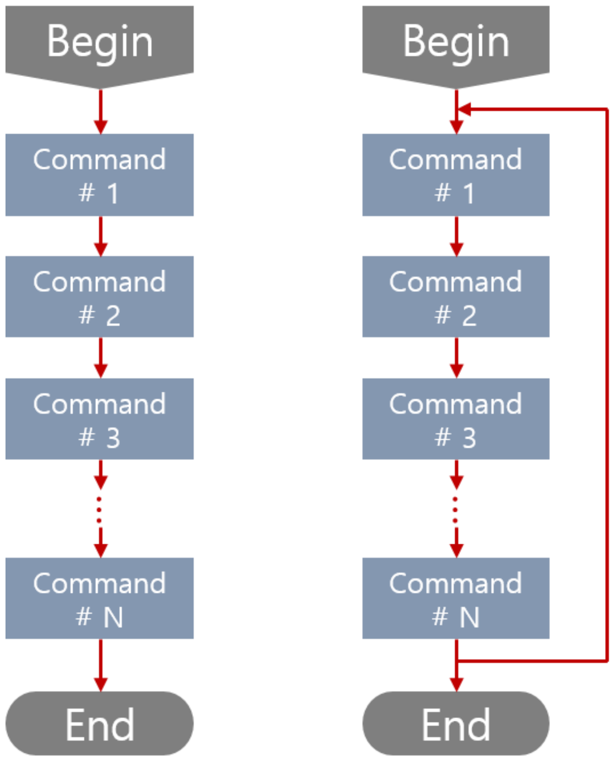

test
Begin
Switch
Case
Empty
Case Default
Empty
End
Up/Down
Undo
Copy
Paste
Cut
Delete
Pass
Marking
Load
Save
Save as
New
Case
Case Number
Memo
Set
Close

■ Pre/Post Program Function:

You can define functions that are executed only once before starting the program (Pre-Program) and functions for wrap-up tasks after the program ends (Post-Program).

When a program continuously runs, like in Play mode, the content declared under the Pre-Program function is executed only once at the beginning. One-time commands, such as variable declarations or communication connections, can be managed under the Pre-Program section.

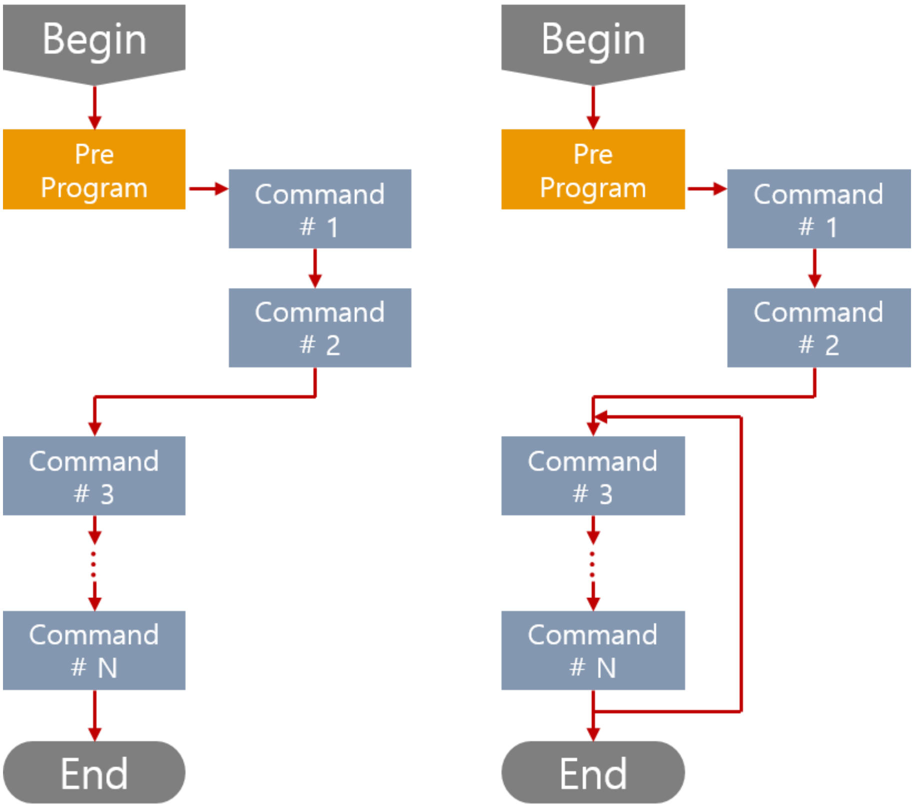
The diagram below shows the normal flow of command execution when the Pre-Program function is not used:



[Make screen] [Play screen]

The left side shows the program execution in the Make screen, while the right side shows it in the Play screen. In the Make screen, the program between Begin and End is executed once. In the Play screen, the program between Begin and End is repeated.

The next diagram shows the flow of command execution when the Pre-Program function is used:



[Make screen] [Play screen]

In the Make screen, the commands between Begin and End are executed sequentially, regardless of whether the Pre-Program function is used. In the Play screen, the program between Begin and End is repeated, but the commands declared under the Pre-Program section are executed only once.

Operations that need to be performed only once, such as variable declarations or communication connections, can be configured using the Pre-Program function.

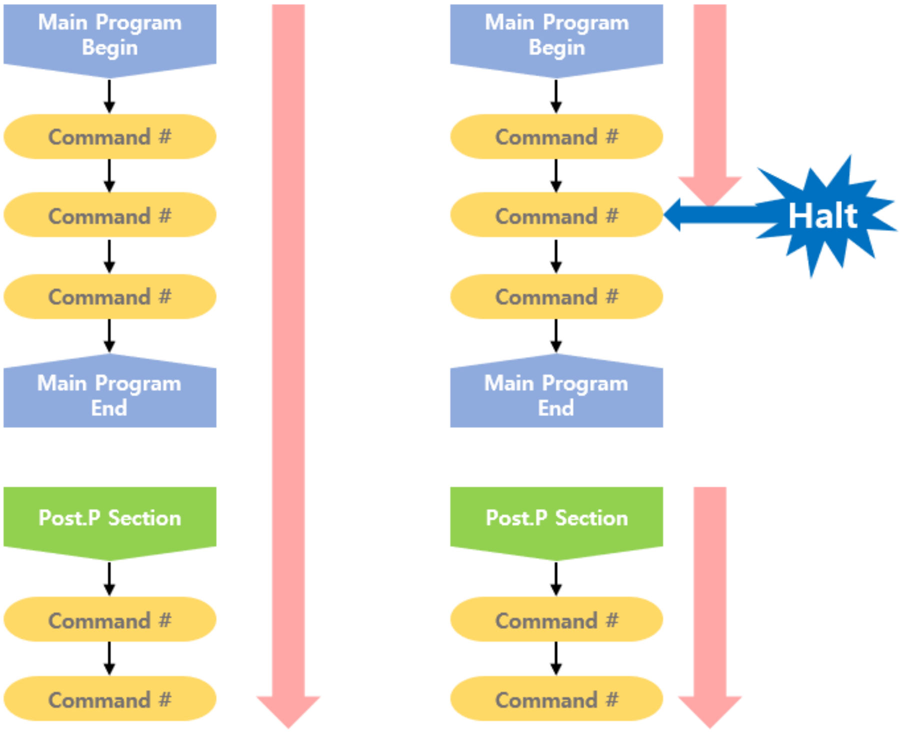
Below is an example of how the Pre-Program function is inserted into an actual project. Due to its nature, the Pre-Program function is only inserted below Begin and cannot be copied and pasted elsewhere.

WARNING

Warning:

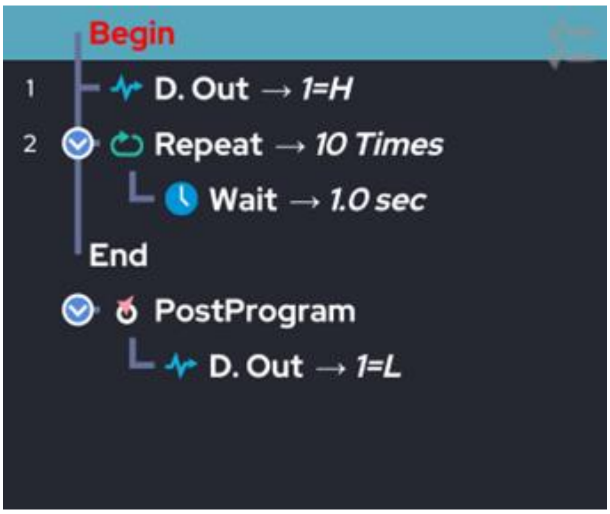
1) If a sub-program called through the Sub.P function uses the Pre-Program function, the Pre-Program commands will only be applied in the main program. The Post-Program function allows you to define commands that will be executed after the program ends. It is intended for wrap-up tasks after the program finishes.

- The commands declared under the Post-Program section are executed sequentially after the program ends.
- The execution of the Post-Program follows the diagram below:



Post-Program Execution Example 1:

The following example shows the use of the Post-Program function. As soon as the program starts (below Begin), a High signal is sent to port 1 using the D.Output function. Even though the D.Output for port 1 to turn off was not added before the end of the program, port 1 automatically sends a Low signal when the program ends, thanks to the use of D.Output 1=L in the Post-Program section.



As shown in the example, the Post-Program function can be used for safety features.

Post-Program Execution Example 2:

This example shows how the Post-Program function is used. When the program ends normally, the beacon light (connected to D.Output 0) does not turn on. However, if the program ends abnormally, the beacon light is turned on.

In this example, the variable SD_IS_INTENDED_STOP is a system internal variable that is always initialized to 0 (false) when the program starts. When the program is stopped intentionally by the user, this variable is set to 1 (true).

If the program stops abnormally for any reason, this variable remains false.

Normal terminations include clicking the UI stop button, I/O stop signals, or other user-triggered stop signals (SD_IS_INTENDED_STOP = true).

Abnormal terminations include stopping due to approaching a singularity or syntax errors (SD_IS_INTENDED_STOP = false).

- Even if the program ends abnormally (e.g., by pressing Halt in an Alarm), the Post-Program functions will still be executed.
- Commands related to robot arm movement, such as Move J or Move L, cannot be used under the Post-Program section.
- The Post-Program function only works in the top-level program. If used in a subroutine, it will not be executed.

■ Set Function:

The Set function allows users to temporarily change parameter settings, regardless of the default values contained within the Setup menu. While the settings in the Setup menu are applied as defaults to all projects, the Set function allows users to temporarily override these parameters.

The various parameters that you change on the Setup screen are applied as default values for all projects that use that control box. If you need to use certain parameters separately for a particular project, you can manage parameter settings by project by adding the Set function to the top of the project (for example, Pre.P. sub).

The Set function is a temporary setting, not a permanent setting. When a new Set function is called for the same parameter setting, the parameter is reflected based on the new Set function.

When the program ends, the parameter settings will return to the default values as defined within the Setup menu.

The parameters that can be changed via the Set function are as follows:

- Time
- Collision Threshold
- Payload
- Linear Move Offset
- Inbox
- TCP Position
- Tool Collision Box
- Global Workspace
- Inbox Size
- Collision On/Off

- Speed Override
- Acceleration Override
- Serial Configuration
- Fixed Vel/Acc
- Spiral Circle Mode
- Speed Bar Control
- Collision Stop Mode
- User Coordinate Shift
- After Collision Detect
- Disable Box D.out
- Move L type XYZ Projection
- Move L type Orientation Align
- User Coordinate Config
- XYZ Shift
- XYZ Shift2
- Vibration sensor On/Off
- Digital Input Simulation
- Program Flow Control
- High acceleration Mode
- Motion Time Constraints
- High Sensitivity Coll.Detect
- Micro offset value
- User Coordinate Shift 6D
- User Coordinate Auto Alignment
- Timer Setting
- No-Arc Move speed
- Manual User Coordinate 6D

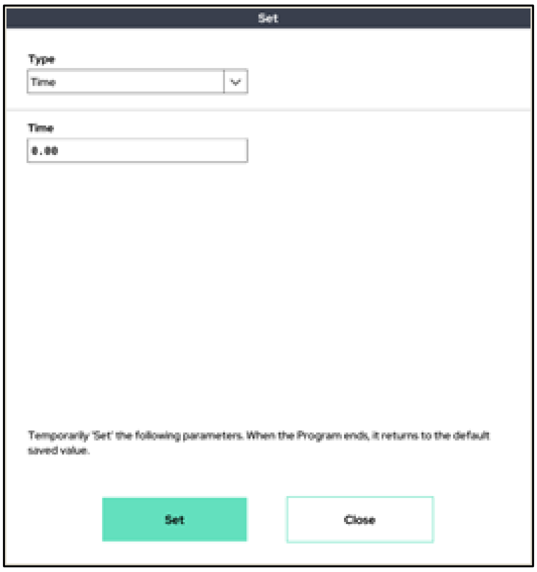
- Jacobian based Speed Control
- External F/T sensor signal
- Joint => Point
- Point => Joint
- TCP Linear Speed Limit
- Force Control Displacement Limit
- Motion Break
- Vibrating Motion
- Move L Deadzone Avoidance
- Axis Aligned Posture Calculation
- Position Control Signal Smoothing
- Disable Control Box Designated Input
- Robot Self Vibration
- PFL Mode
- Gravity Compensation Mode (Current Control)
- Landmark based Fitting

WARNING

Warning:

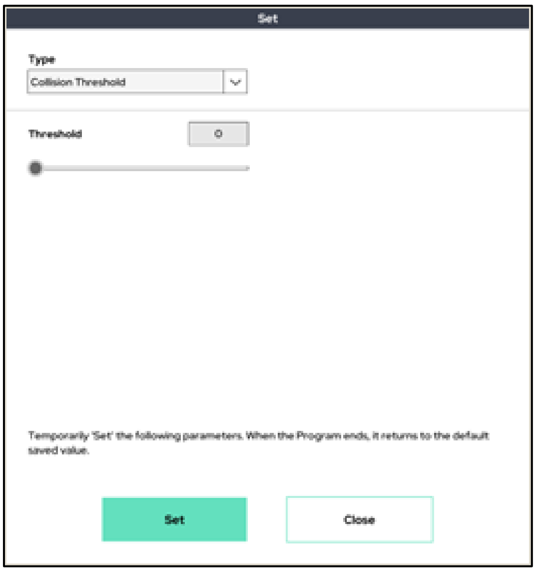
1) The value set in the Set function is a temporary value. When the program exits, it automatically returns to the default values set from the Setup Menu.
2) The functions provided by the Set function allow you to change the setting value to another value in the middle of the program flow. For example, you can use Set's 'Collision On / Off' feature to selectively turn on/off collision detection in the middle of a program flow.

- Set Function: Time



Starts the timer and sets the initial value. Starting with the value entered, the value of the timer increases.

- Set Function: Collision Threshold



Temporarily sets the collision detection sensitivity. The lower the value, the more sensitive the robot is to collision. This has the same functionality as the Collision Threshold option within the Setup-Cobot Menu.

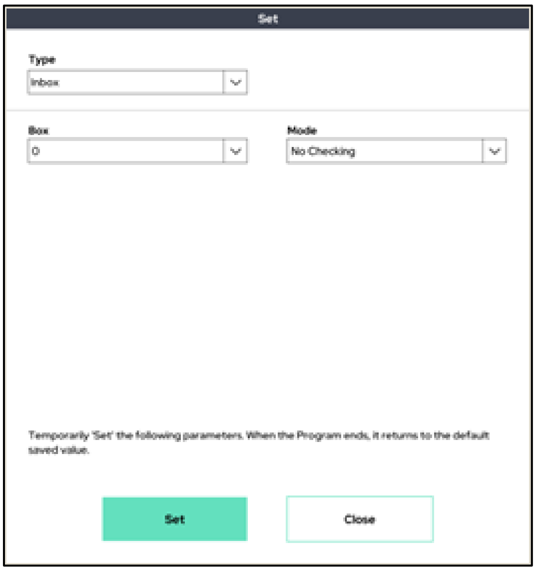
- Set Function: Payload

Temporarily set the tool's weight and center of gravity. This has the same functionality as the Payload option within the Setup Menu.

- Set Function: Linear Move Offset

Gives a slight offset relative to the base coordinate system. This function allows users to temporarily set an offset of up to 20 mm.

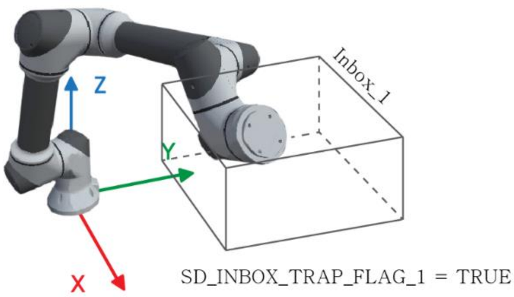
- Set Function: Inbox



Allows the user to enable the Inbox Checking feature. The Inbox Checking feature checks whether a certain part of the robot is in a predefined area (either in the Setup screen or the using the Set function). The parts of the robot that can be checked are as follows.

- Check Tool Flange Center: Verify that the set TCP at the robot arm end has entered a specific area.
- Check Tool Center Point: Determine if the center of the tool flange at the end of the robot arm has entered a specific area.
- Check Tool Box: Verify that the box set virtually on the robot arm end has entered a specific area.
- Check All: All the above.

The size and position of the box can be set in the Inbox screen using Setup mode (or through the Set function). A total of 2 Inboxes can be set, and based on the above settings, whether or not they are in a specific area can be used through SD_INBOX_TRAP_FLAG_0 or SD_INBOX_TRAP_FLAG_1 in the Shared Data type data, respectively.



- Set Function: TCP Position

Temporarily set a relative offset of the tool's TCP position. It has the same functionality as the End Effector menu in Setup-Tool.

- Set Function: Tool Collision Box

Temporarily set the size and position of a virtual box surrounding the gripper for self-collision prevention. It has the same functionality as Tool Setting for Collision Check in Setup-Tool.

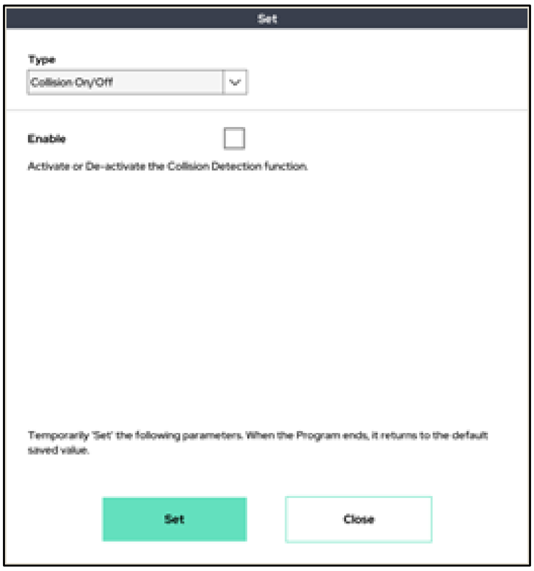
- Set Function: Global Workspace

Temporarily set the limits of the workspace for collision prevention. It has the same functionality as the Workspace Limits menu in Setup-Cobot.

- Set Function: Inbox Size

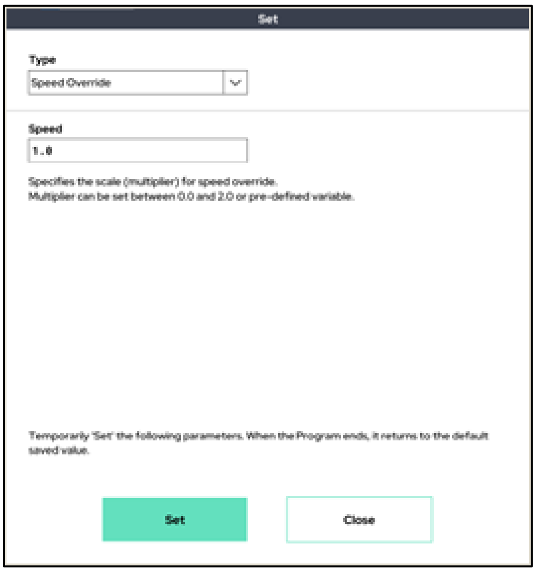
Temporarily set the position and size of the Inbox. It has the same functionality as the Inbox settings in Setup-Inbox.

- Set Function: Collision On/Off

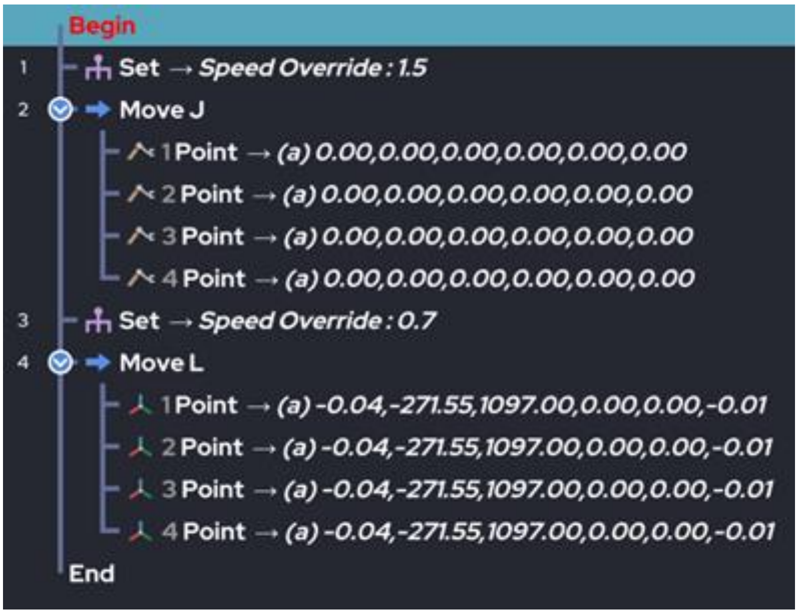


Temporarily sets the use of external collision detection mode. It has the same functionality as the Enable Collision box in Setup-Cobot.

- Set Function: Speed Override

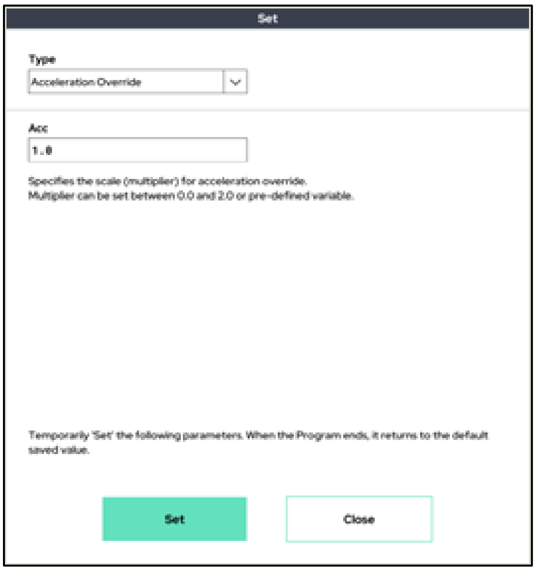


Allows the user to temporarily to change the base scaled speed used by the Move and Point functions. Users can either enter a value between 0 and 2.0, or a predefined variable.

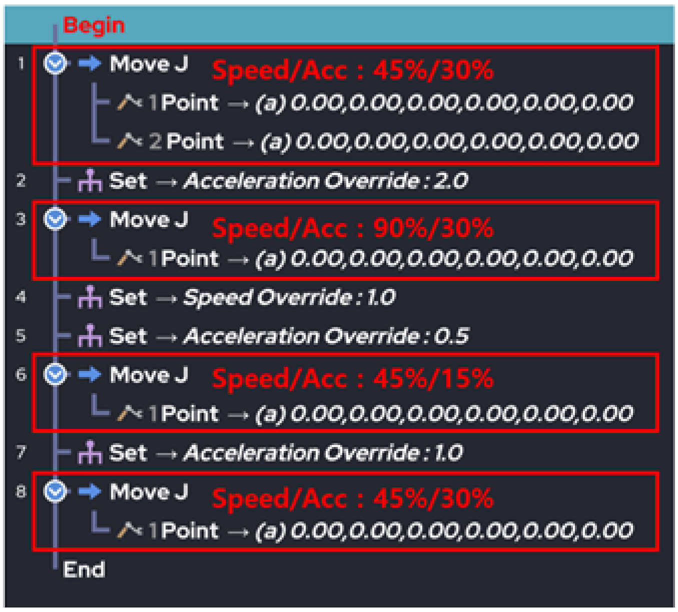


In the example above, Set-Speed Override 1.5 is set in front of Move J. Therefore, the Move J operation has the same effect as if a speed of 1.5 times the set speed was set. Set-Speed Override 0.70 was used between Move J and Move L. Therefore, Move L operates with the same effect as if a speed of 0.70 times the set speed is set.

- Set Function: Acceleration Override

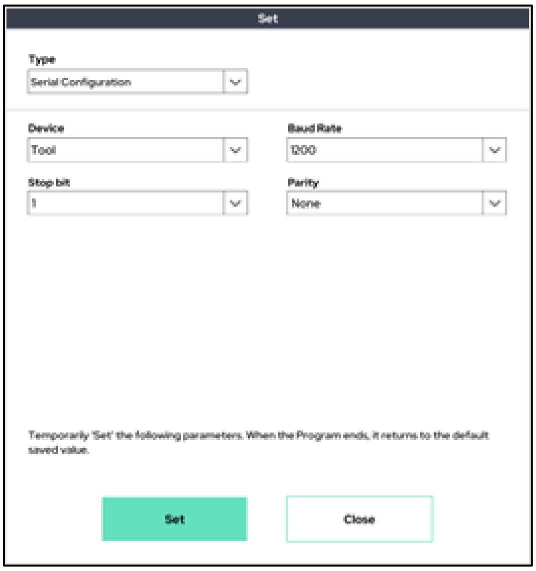


Allows the user to temporarily to change the base scaled acceleration used by the Move and Point functions. Users can either enter a value between 0 and 2.0, or a predefined variable.



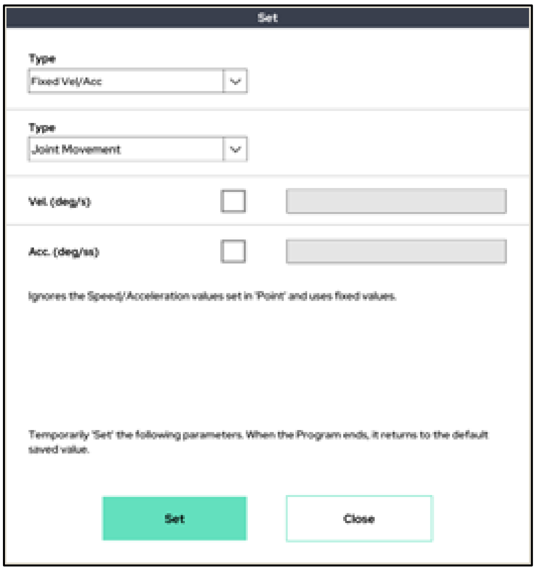
Through the code below, you can see how the speed and acceleration change when Speed Override and Acceleration Override are used. The initial speed and acceleration are set to 45% and 30%, respectively, and change depending on the setting scale.

- Set Function: Serial Configuration



The baud rate and stop bit / parity of the serial communication are temporarily set. It has the same meaning as set in Setup-Serial.

- Set Function: Fixed Vel/Acc



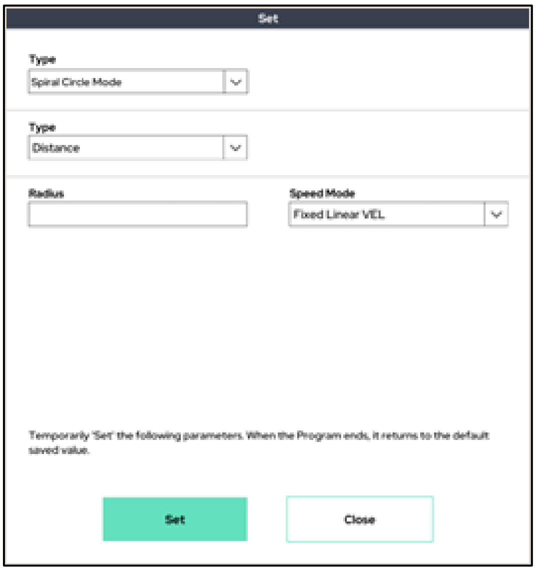
This function is used when you want to use a fixed value, ignoring the set speed / acceleration for each Move point. There are two sub options: Joint Movement and Linear Movement.

The velocity (deg/s) and acceleration (deg/s^2) set in the Joint Movement affect the movement speed and acceleration of the Joint movement types Move J and Move JB.

The velocity (mm/s) and acceleration (mm/s^2) set in Linear Movement affect the movement speed and acceleration of the linear movement types Move L, Move PB (Formerly Move LB), Move JL, Move ITPL, Move Pro and Circle. If you do not want to force speed/acceleration through this function, clear the check box. In this case, it follows the speed/acceleration value set for each point during operation.

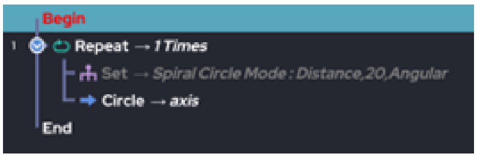
Ex) If you need to keep a certain speed and acceleration during operation, you can use this Set function as in the code below.

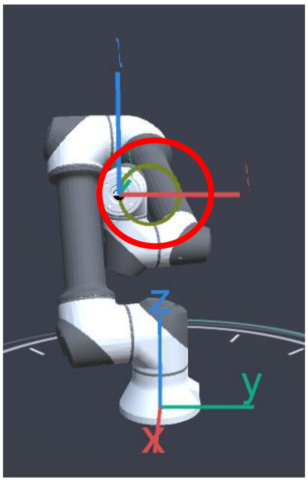
- Set Function: Spiral Circle Mode



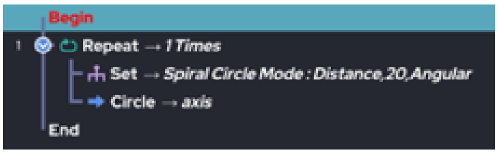
This function is used to change the circular motion into spiral motion. Draw a circle / arc when using the Circle function. If Set-Spiral mode is used over the Circle function, the existing circle / arc will be changed to spiral motion. Therefore, to implement spiral motion, this function should be inserted above the Circle function. In the property, Distance sets how much a helix moves relative to the starting point when implementing a spiral circular motion, and Rate moves by applying a ratio to the radius of the circle or arc. Therefore, 1 is 100% when using Rate.

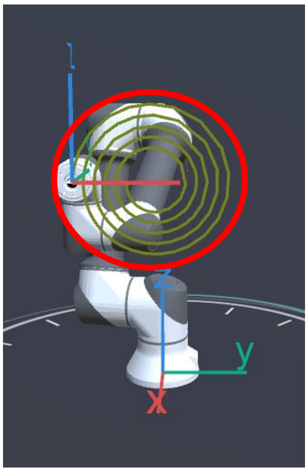
Ex 1) Only Circle is used: Create a general circle / arc trajectory.



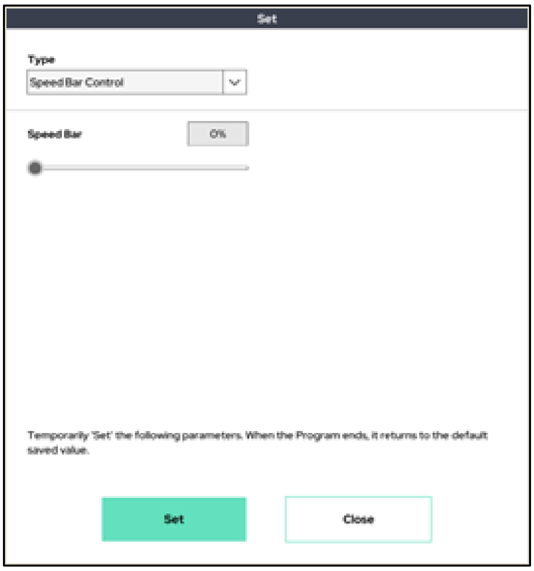


Ex 2) Set-Spiral Mode + Circle: Spiral trajectory



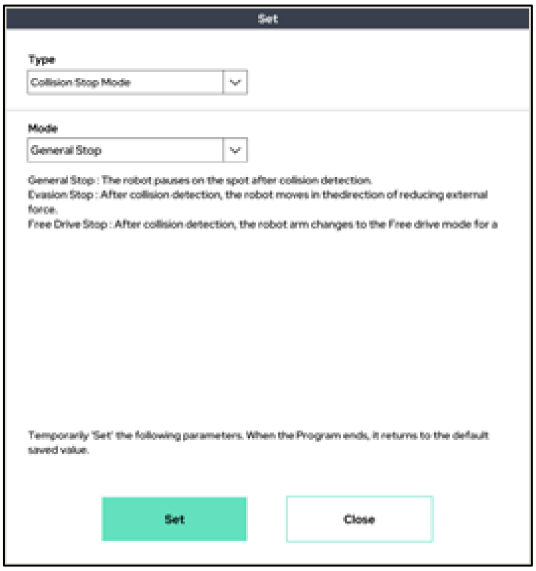


- Set Function: Speed Bar Control



The speed control bar (bottom right) of the UI can be adjusted with the program. You can change the UI speed control bar by using this function in the desired section.

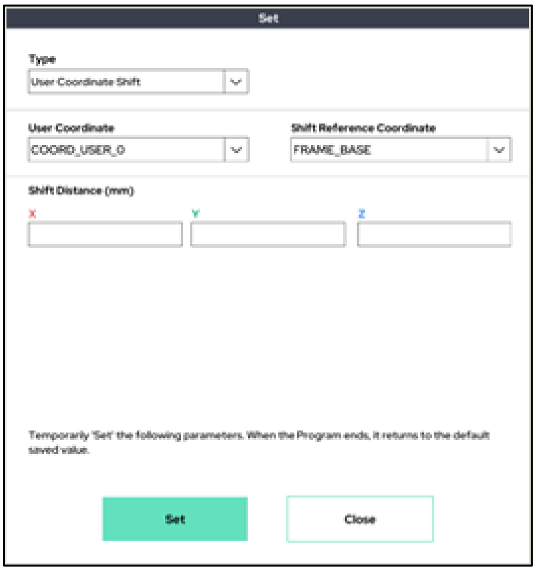
- Set Function: Collision Stop Mode



Select the robot's motion type when after detecting an external collision. There are three options.

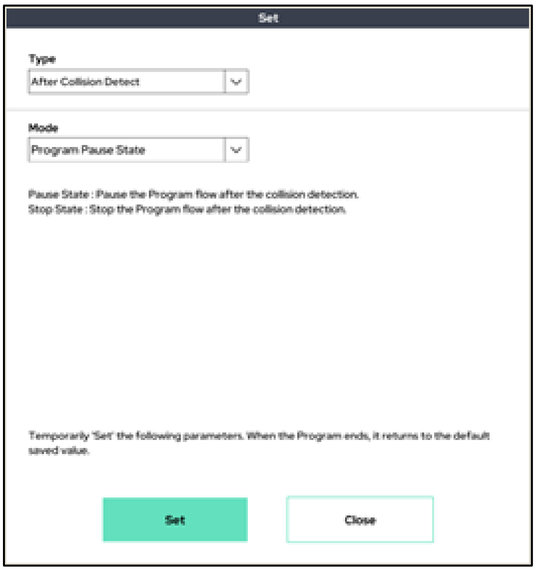
- General Stop: After the collision is detected, the trajectory movement is paused on the spot.
- Evasion Stop: After the collision is detected, the robot moves a small amount away from the external force, then pauses the trajectory movement. The degree of Evasion Stop is divided into Normal, Smallest, Small, Big, and Biggest.
- Free Drive Stop: After detecting the collision, the robot arm switches directly to the teaching state for a certain period of time. The time to stop the direct teaching is divided into 0.8 seconds and 1.6 seconds, respectively. It has the same meaning as Setup > Cobot "Action after Collision".

- Set Function: User Coordinate Shift



This function is to move the origin of user coordinate system temporarily. can set the user coordinate system number and shift distance you want to shift and choose which coordinate system to shift the shift distance.

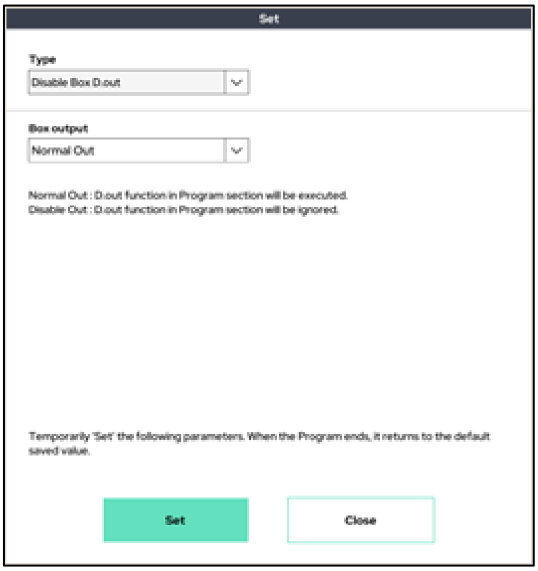
- Set Function: After Collision Detect



The program flow can be selected after external collision detection. Our default setting is to pause the program after detecting an external collision. After detecting a collision, a collision detection alert pops up and the program and threads are paused. If you want to terminate the program after collision detection, you can use this function to select the option as Program Stop state.

- Program Pause State: Program flow is paused after external collision detection.
- Program Stop State: Program flow stops after external collision detection.

- Set Function: Disable Box D.out



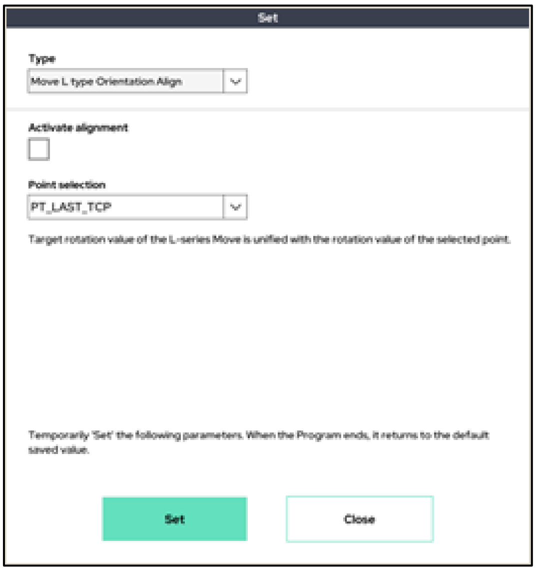
This function temporarily disables the digital output of the control box. Even if the digital output command inserted in the program is not erased, this set command can be used to ignore the digital output command in a specific section. It can be used for development testing, etc., and by selecting an option, the output can be deactivated/activated according to the program section.

- Set Function: Move L type XYZ Projection

This is a function to fix the target position coordinate value of L series movement (eg Move L. Move PB, Circle etc). If you select the value to be fixed and the reference coordinate system, the position coordinate value of the target point or set point is fixed to the value of the selected axis of the selected coordinate system.

For example, if the base coordinate system (Global) is selected as the coordinate system and Z Projection 100mm is selected/written, the Z height of all moving target values/set coordinate values is applied collectively as 100mm.
This function is also a set function, which can be activated/deactivated for each section of the program. If you want to disable it, select None in the coordinate system.

- Set Function: Move L type Orientation Align

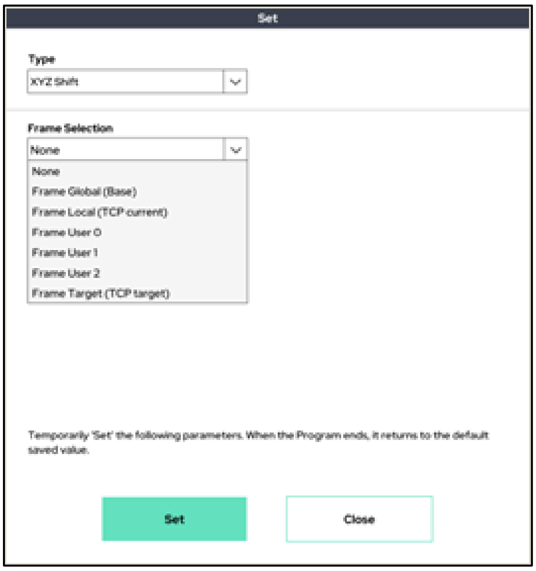


This function is to fix the target rotation coordinate value of L series movement (eg Move L. Move PB, Circle etc). Fix the rotation of L series motions with the rotation value of the selected Point.
As a sub-function of the Set function, this function can be turned on or off depending on the program section. This function can be used when you want to uniformly rotate the TCP rotation at a time.

- Set Function: User Coordinate Config

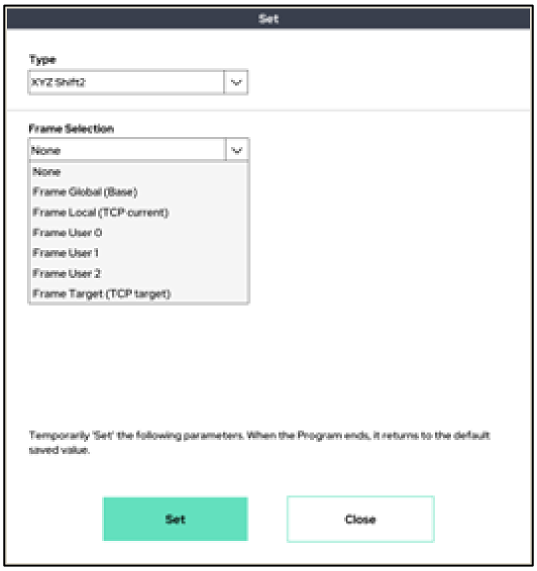
This function allows you to temporarily change the user coordinate system settings. By selecting three points in the middle of the program flow, the user coordinate system setting can be arbitrarily changed in the middle of the program. Because it is a sub-function of Set, the user coordinate system setting returns to the default value when the program ends.

- Set Function: XYZ Shift



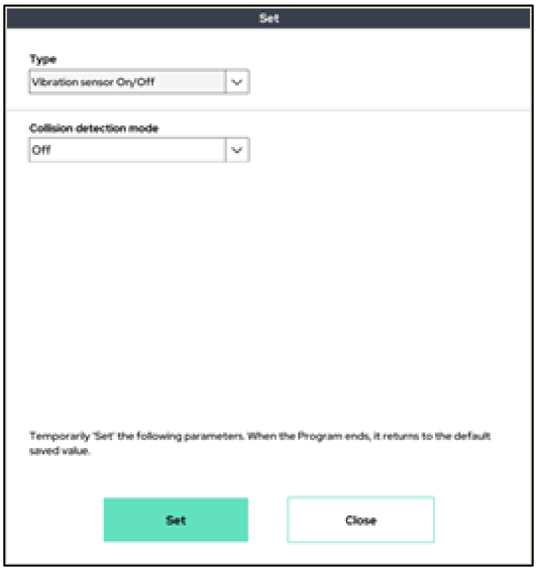
This function allows you to temporarily shift the target point. User can select a base/tool/user coordinate Config and enter shift values from the target point. At this time, select whether to apply this shift only to L type or to both L type and J type. Because it is a sub-function of Set, it returns to the default value when the program ends.

- Set Function: XYZ Shift2



This function allows you to temporarily shift the target point. User can select a base/tool/user coordinate Config and enter shift values from the target point. At this point, this shift is only applicable to L series operation, and both the XYZ position value and the rotation value can be entered. Because it is a sub-function of Set, it returns to the default value when the program ends.

- Set Function: Vibration sensor On/Off

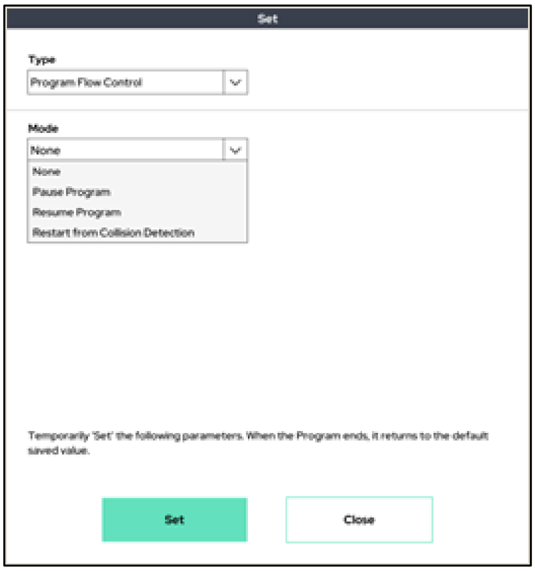


This function allows you to temporarily exclude collision detection by vibration during collision detection. Because it is a sub-function of Set, it returns to the default value when the program ends.

- Set Function: Digital Input Simulation

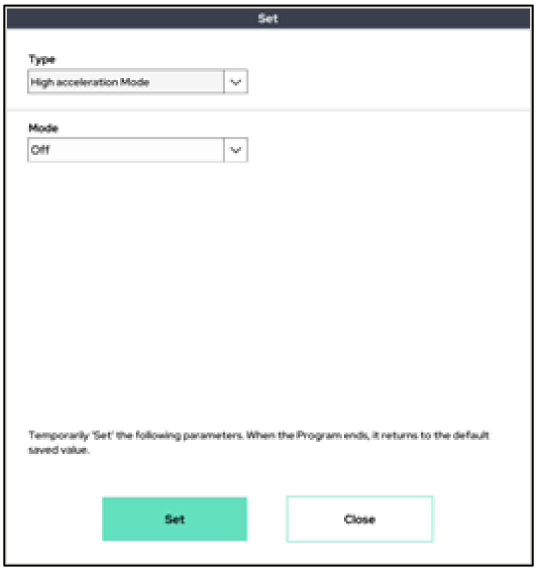
This function allows you to simulate Digital input signal. Create the desired input by setting the state of the port to which you want to input. Because it is a sub-function of Set, it returns to the default value when the program ends.

- Set Function: Program Flow Control



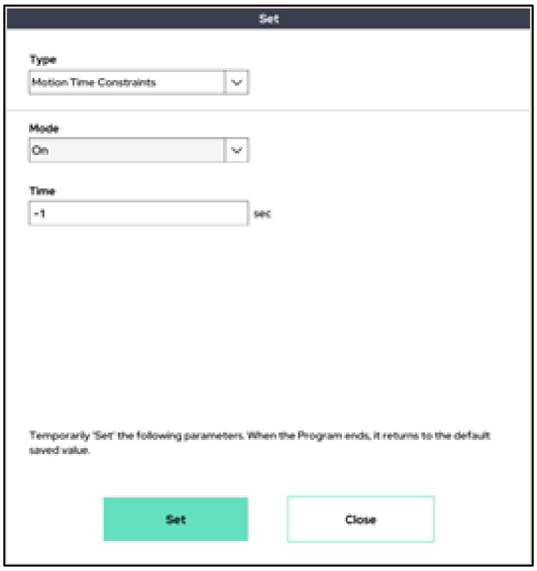
This function allows you to pause and restart without using alarms and I/O when a program is running. Because it is a sub-function of Set, it returns to the default value when the program ends.

- Set Function: High Acceleration Mode



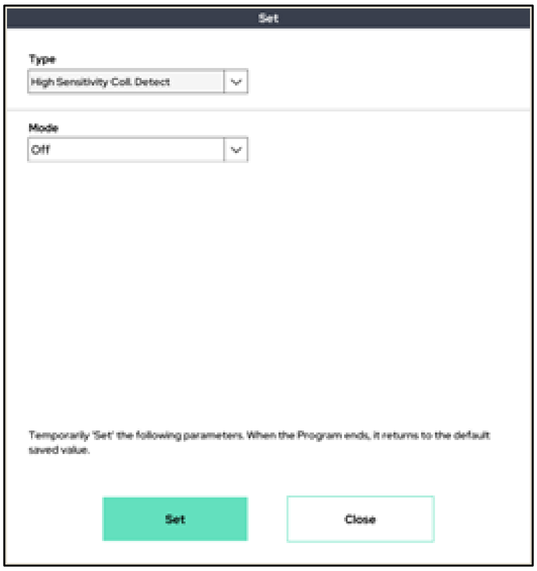
High acceleration mode reduces the time the robot's operating speed reaches the desired operating speed through changes in the reduction/acceleration profile. Because it is a sub-function of Set, it returns to the default value when the program ends.

- Set Function: Motion Time Constraints



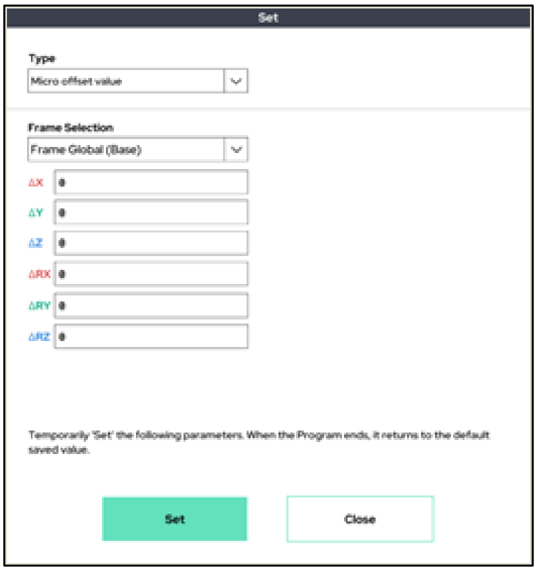
Motion Time Constraint is a function that constrains the time taken to move a point to point by the time entered. At this time, it is possible to increase time but not to reduce it. Time must be a value greater than or equal to zero. Because it is a sub-function of Set, it returns to the default value when the program ends.

- Set Function: High Sensitivity Coll. Detect



High sensitivity Coll.Detect allows the detection of collision to be 30% more sensitive than the existing sensitivity. In Setup, the sensitivity that made collision detection the most sensitive is also 30% more sensitive than 0%. Because it is a sub-function of Set, it returns to the default value when the program ends.

- Set Function: Micro offset value

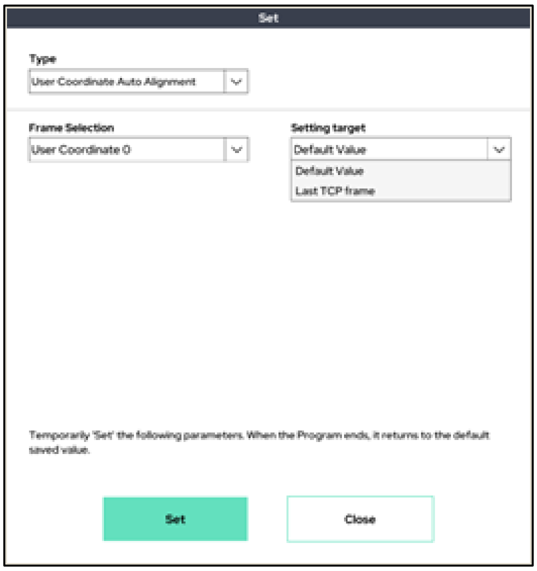


User can give a slight offset based on the desired coordinate system. This function enables temporary offset settings of up to 20mm. Because it is a sub-function of Set, it returns to the default value when the program ends.

- Set Function: User Coordinate Shift 6D

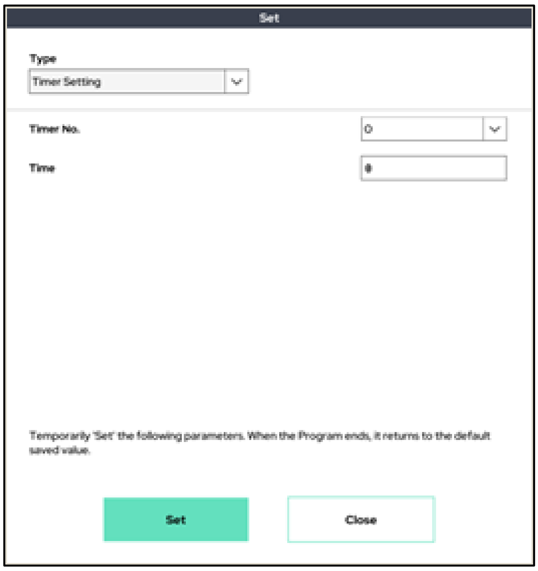
The user can temporarily shift the user coordinate. This function allows the user to temporarily change the position, rotation of the user's coordinate. Because it is a sub-function of Set, it returns to the default value when the program ends.

- Set Function: User Coordinate Auto Alignment



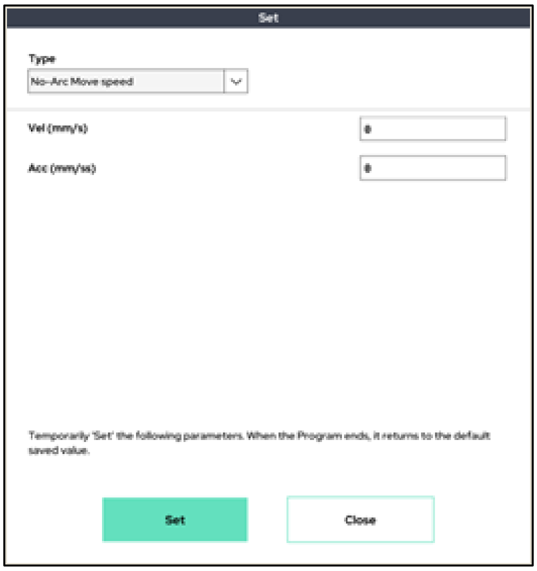
This function allows the user to change the user coordinate to the last TCP frame. It is also possible to return to the default user coordinate. Because it is a sub-function of Set, it returns to the default value when the program ends.

- Set Function: Timer Setting



User can select the timer their want to use and set the initial value of the timer. The timer starts from the initial value set by the user. Because it is a sub-function of Set, it returns to the default value when the program ends.

- Set Function: No-Arc Move speed



Set the move speed of the robot in the no-arc state where welding is not performed. Because it is a sub-function of Set, it returns to the default value when the program ends.

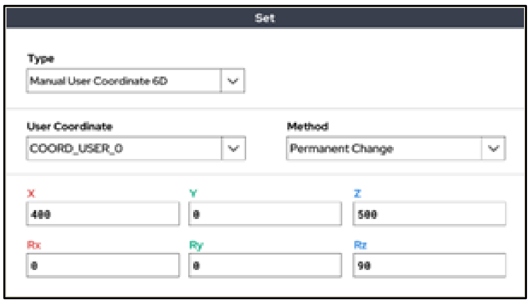
- Set Function: Manual User Coordinate 6D

You can change the user coordinate system during program execution. If you choose temporary change mode, the specified user coordinate system will only be applied when the program is running. If you choose permanent change mode, the value will be applied permanently. As shown in the image below, the current user coordinate system (No. 0) is set with X and Z values of 400mm in Setup-Coordinate. Because it is a sub-function of Set, it returns to the default value when the program ends.

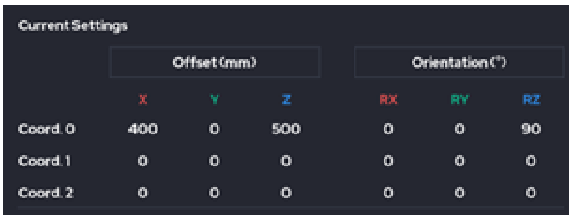
Current Settings

	Offset (mm)			Orientation (°)		
	X	Y	Z	RX	RY	RZ
Coord. 0	0	0	0	0	0	0
Coord. 1	0	0	0	0	0	0
Coord. 2	0	0	0	0	0	0

At this point, the manual user coordinate system 6D is set as shown below, and the program is executed.



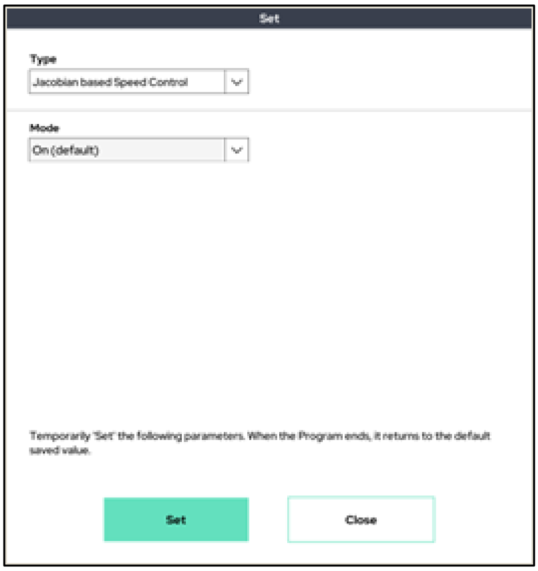
The change can be confirmed as shown below.



Current Settings

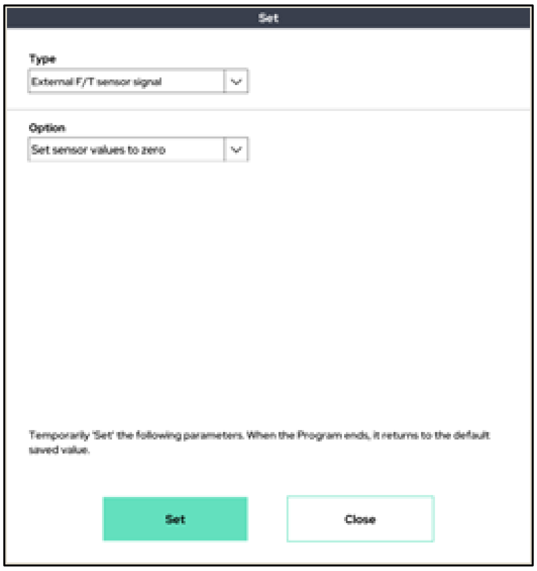
	Offset (mm)			Orientation (°)		
	X	Y	Z	RX	RY	RZ
Coord. 0	400	0	500	0	0	90
Coord. 1	0	0	0	0	0	0
Coord. 2	0	0	0	0	0	0

- Set Function: Jacobian based Speed Control



This feature allows you to turn on or off an algorithm that automatically adjusts the speed when the robot approaches an undesirable posture or configuration.

- Set Function: External F/T sensor signal

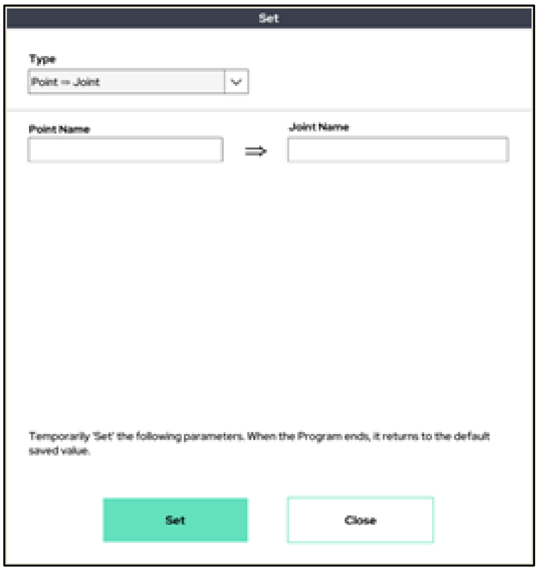


This function resets the external F/T sensor signal to 0 (N). Because it is a sub-function of Set, it returns to the default value when the program ends.

- Set Function: Joint ⇒ Point

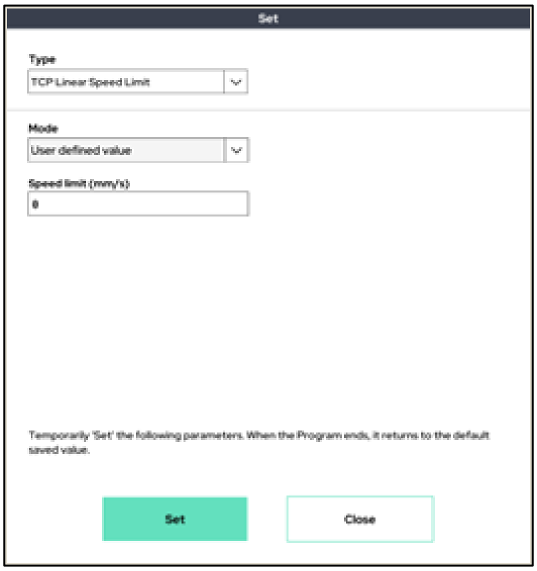
This function allows you to assign the coordinate values corresponding to a joint variable to a point variable. You can use a previously declared joint variable or pin joint and assign it to the joint name, and use a point variable or pin point for the point name. Because it is a sub-function of Set, it returns to the default value when the program ends.

- Set Function: Point ⇒ Joint



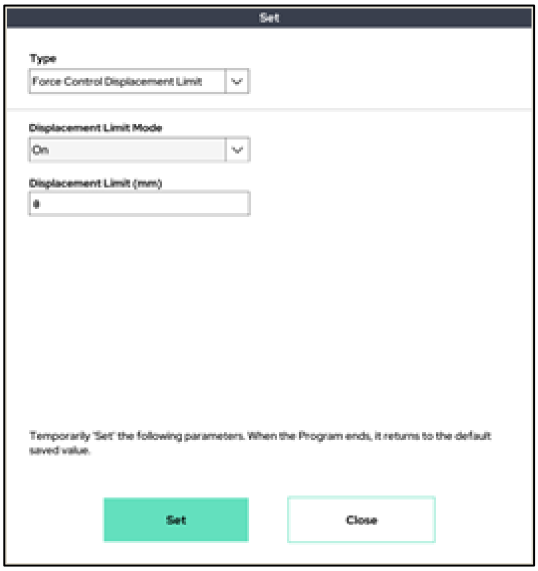
This function allows you to assign the joint angle corresponding to a point variable to a joint variable. You can use a previously declared point variable or pin point for the point name and a joint variable or pin joint for the joint name. Because it is a sub-function of Set, it returns to the default value when the program ends.

- Set Function: TCP Linear Speed Limit



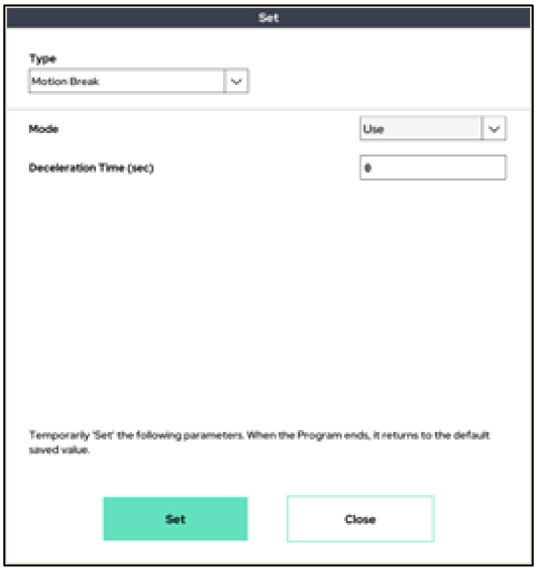
This function allows you to limit the linear speed of the robot's TCP. Because it is a sub-function of Set, it returns to the default value when the program ends.

- Set Function: Force Control Displacement Limit



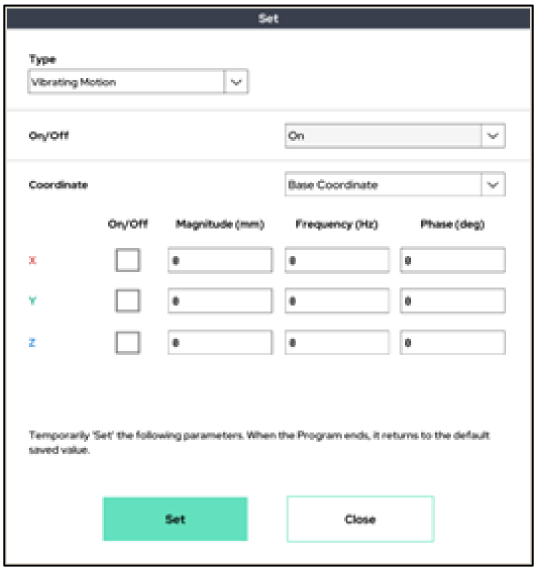
When using force control, the robot moves in a certain direction to measure the force with the sensor. This function limits the displacement during such movement.

- Set Function: Motion Break



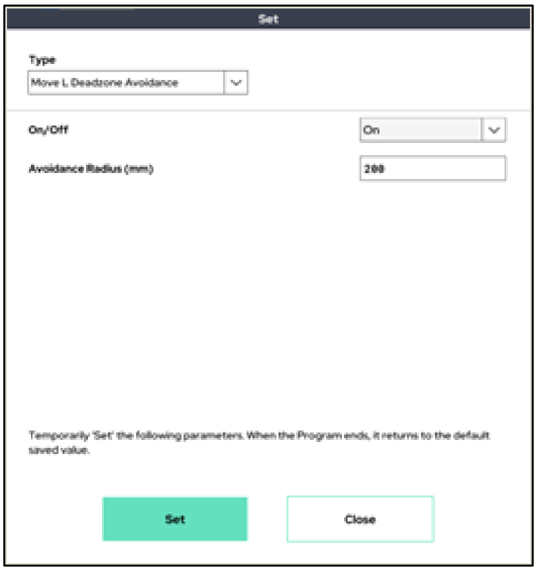
This function is similar to the "Finish At" function but can be used in cases where communication is received from a thread that is not supported by the "Finish At" function to set conditions.

- Set Function: Vibrating Motion



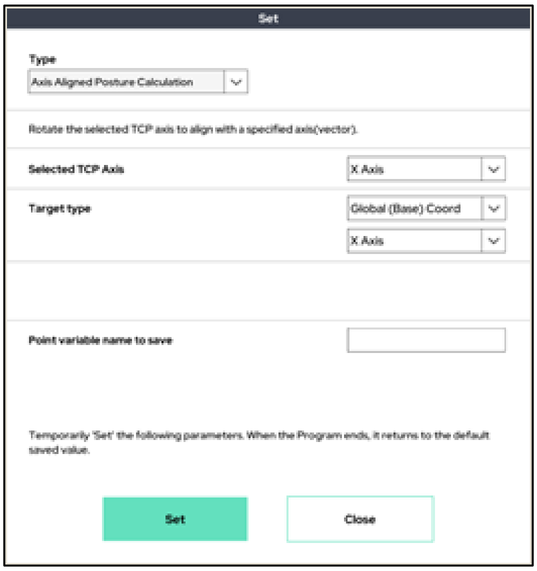
By setting an axis and specifying amplitude and frequency phase, this function makes the robot vibrate in the specified axis direction.

- Set Function: Move L Deadzone Avoidance



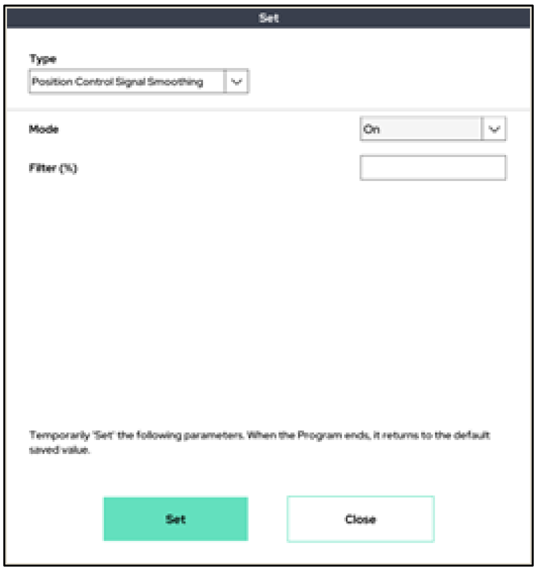
This feature enables the robot to avoid a deadzone with a diameter of 15cm from the center of the robot arm. If the robot arm approaches the deadzone, it will move to avoid it.

- Set Function: Axis Aligned Posture Calculation



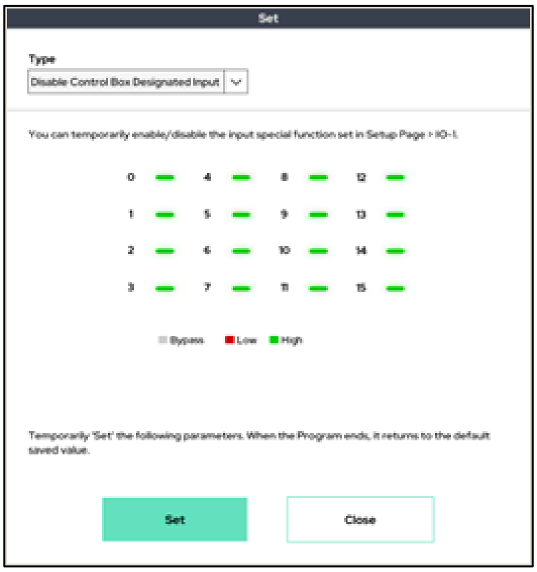
This function aligns the posture in the direction of the specified axis based on the point variable you designate.

- Set Function: Position control Signal Smoothing



This function applies a low-pass filter to the position control signal outputted to the robot's servo motor. Increasing the filter value can make the movement smoother, though it may deviate slightly from the original trajectory.

- Set Function: Disable Control Box Designated Input

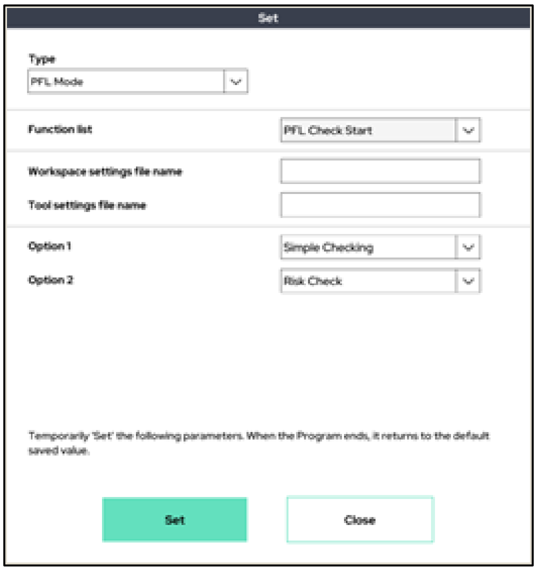


This function enables or disables special functions assigned to digital inputs on the control box under the I/O-1 section of the Setup. Because it is a sub-function of Set, it returns to the default value when the program ends.

- Set Function: Robot Self Vibration

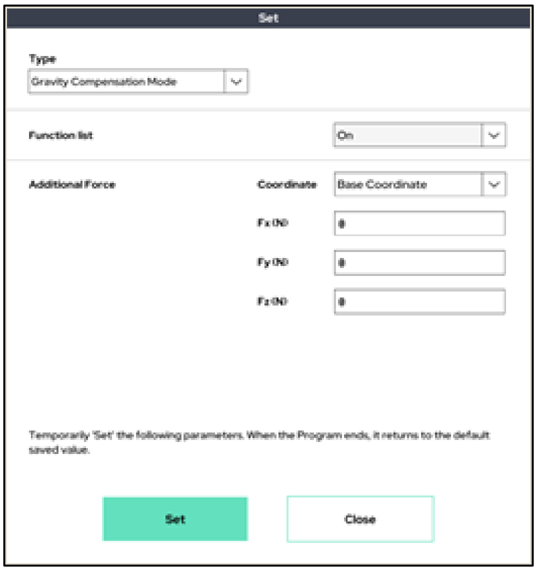
Robot self-vibration implements a motion where the robot vibrates according to the specified time, frequency, and vibration intensity settings.

- Set Function: PFL Mode



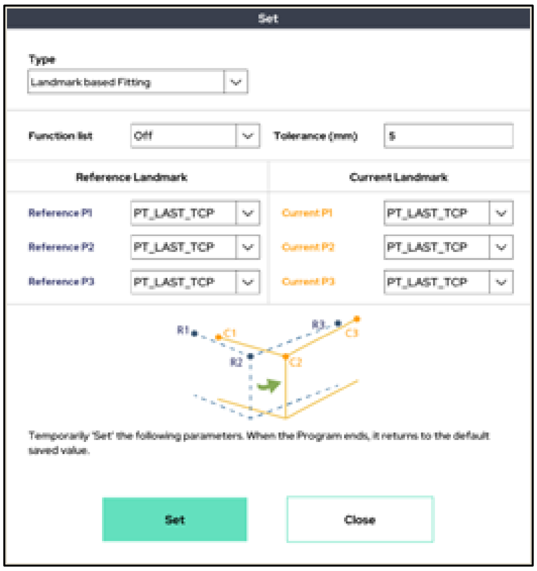
PFL (Power and Force Limit) Mode uses the robot's motion data and tool information to simulate the collision risk at various positions.

- Set Function: Gravity Compensation Mode(Current Control)



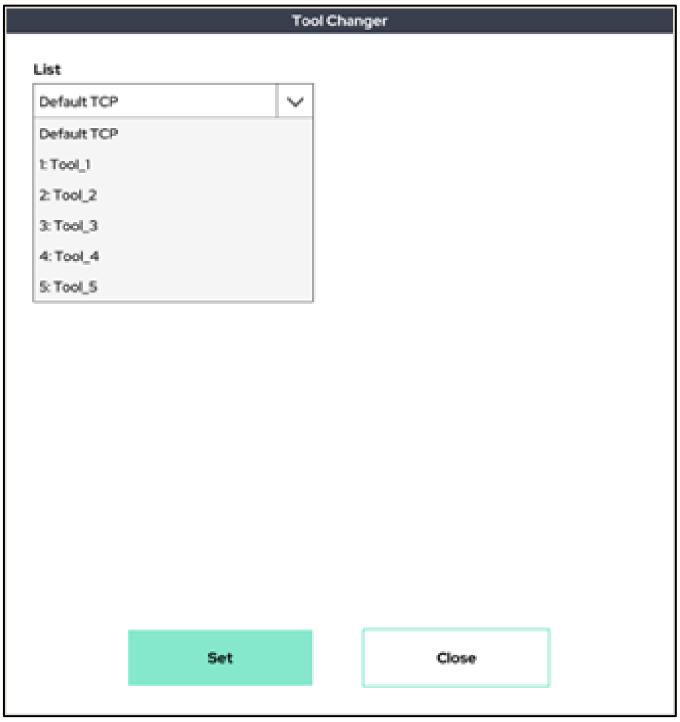
Enabling Gravity Compensation Mode (current control) switches the robot to direct teaching mode. In this mode, the robot generates force in the direction necessary to compensate for gravity according to the amount of force set during direct teaching.

- Set Function: Landmark Based Fitting



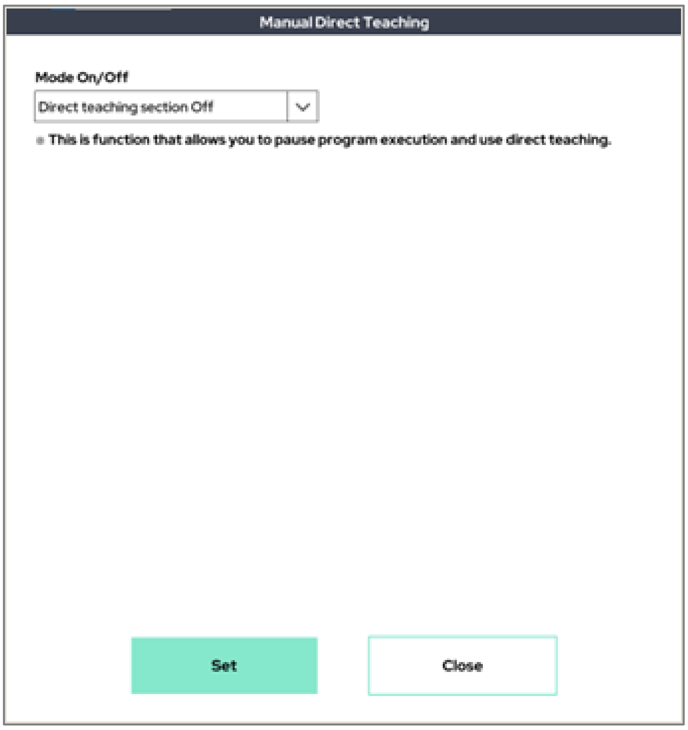
This function automatically calculates and applies an offset to the robot's trajectory based on three user-defined landmarks. For reference points, input the teaching information defined initially during direct teaching. For current landmarks, input the positional information acquired through sensors, such as vision systems.

■ TCP Set Function:

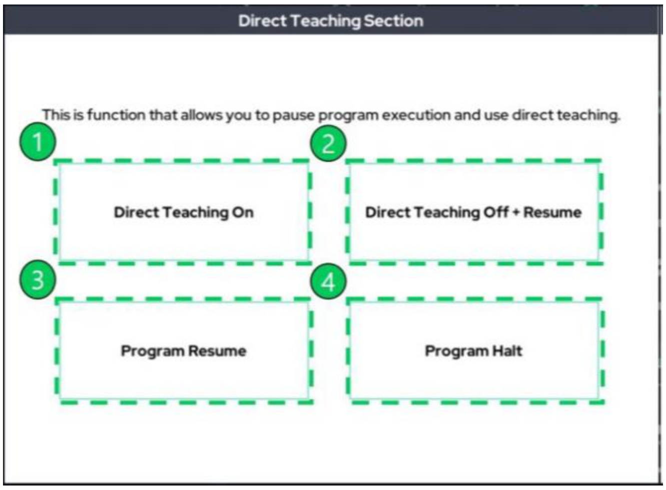


The ability to change the TCP value during program execution with the TCP value pre-saved in Setup-Tool List. A total of five tools can be set to TCP. It does not change again until the TCP value is replaced or the program is shut down.

■ Manual Direct Teaching (Manual.D) Function:



A feature that enables direct teaching during program execution. When mode On, the program pauses when the manual direct teaching command is executed and a pop-up window as shown below appears on the screen.



You can select four features in the pop-up window.

① Use the direct teaching feature while the program is paused.

② If you used the direct teaching feature in ①, turn off the direct contact function and resume the program.

③ Ignore the manual operation and resume the program.

④ Exit the program.

■ Moving Point (M.Point) Function:

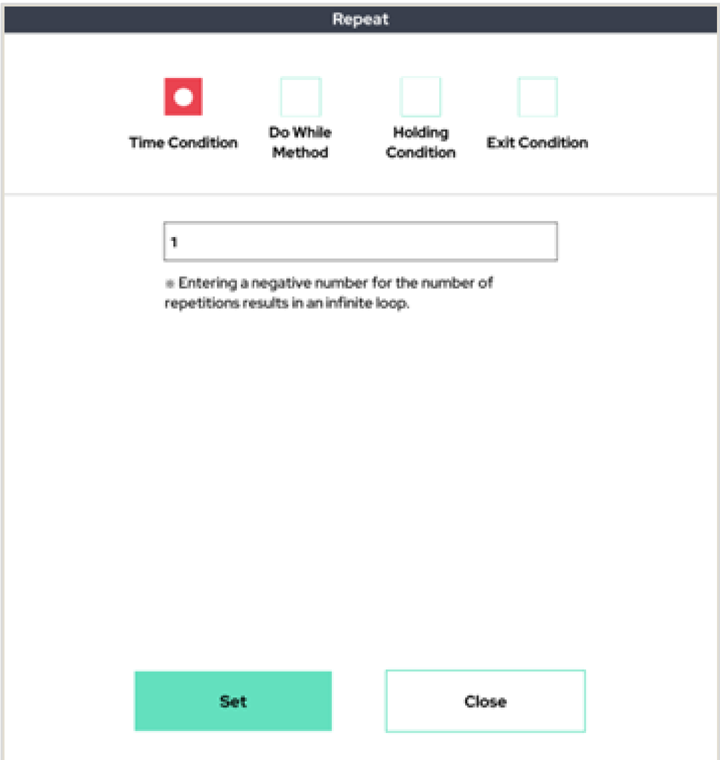
The M.Point function is designed for synchronized control between the robot arm and external axes. If you want synchronized control, you can click the checkbox to the right of the target point for the external axis. If you prefer standard control instead of synchronized control, you can uncheck the box and use the controls in the top-left to operate only the robot arm independently. If a script command is entered in the script field below, the specified script will be executed when the robot reaches that position.

■ Repeat Function:

Repetition statement. Repeat the program below the specified condition or number of times. There are four modes. Repeat a specified number of times, Do While functionality that repeats the specified condition for a true period of time, but runs the first time, even if the condition is not true, Repeat while the specified condition is true, End repeat if the condition is true.

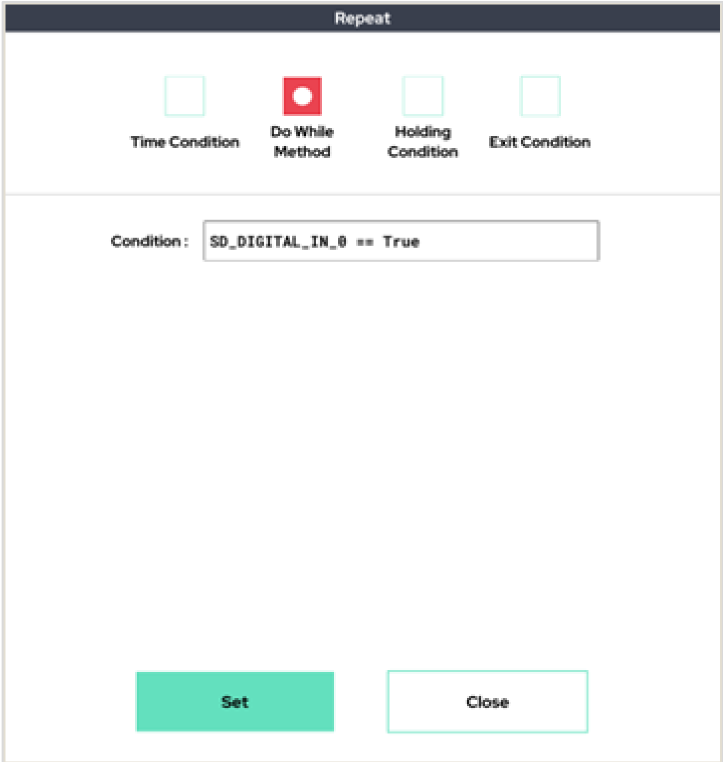
After clicking the Repeat button, a popup menu containing the four modes will appear. Once it opens, select and use the desired function.

1) Repeat a specified number of times (Time Condition)



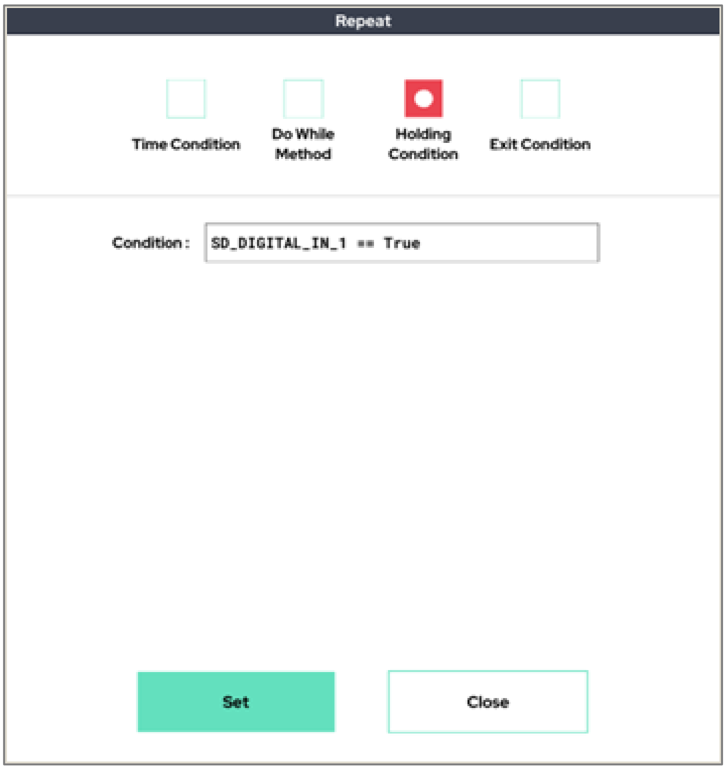
Ex) The above example will repeat a subprogram 1 times.

2) Do While Function



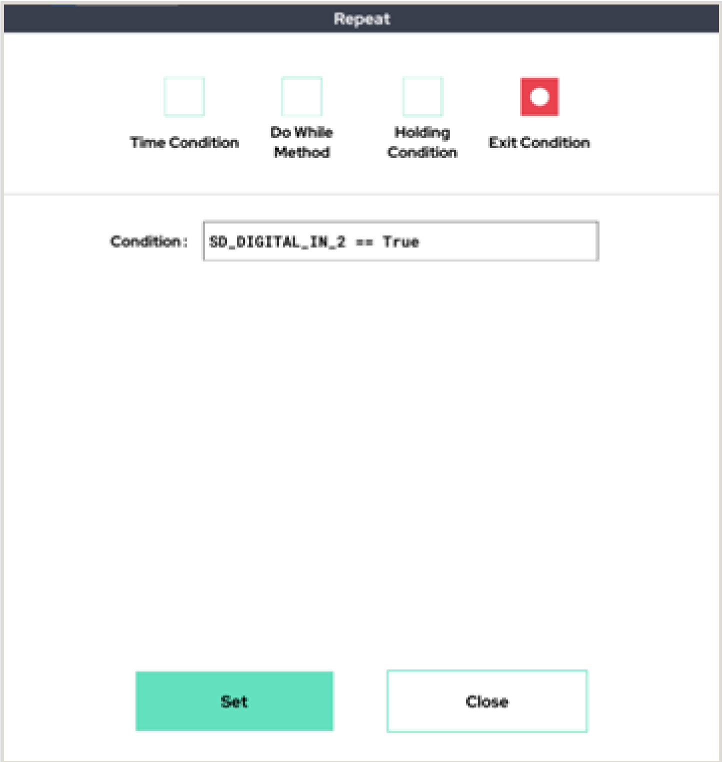
Ex) The conditional statement you entered in Condition repeats for a true period of time, but it repeats the first time, even if it is not true.

3) Repeat while the specified condition is true (Holding Condition)



Ex) If the conditional statement entered in Condition is true, repeat it again and again.

4) End repeat if the condition is true (Exit Condition)



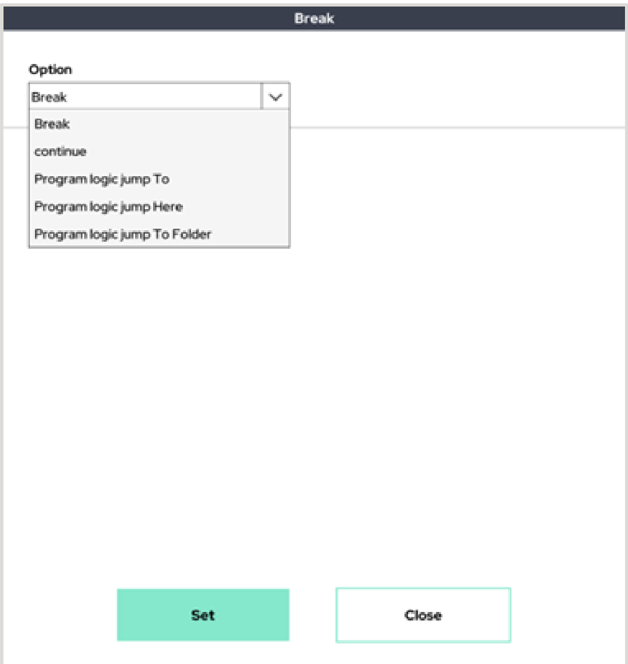
Ex) If the conditional statement you entered in Condition is true, stop repeating and execute the following command.

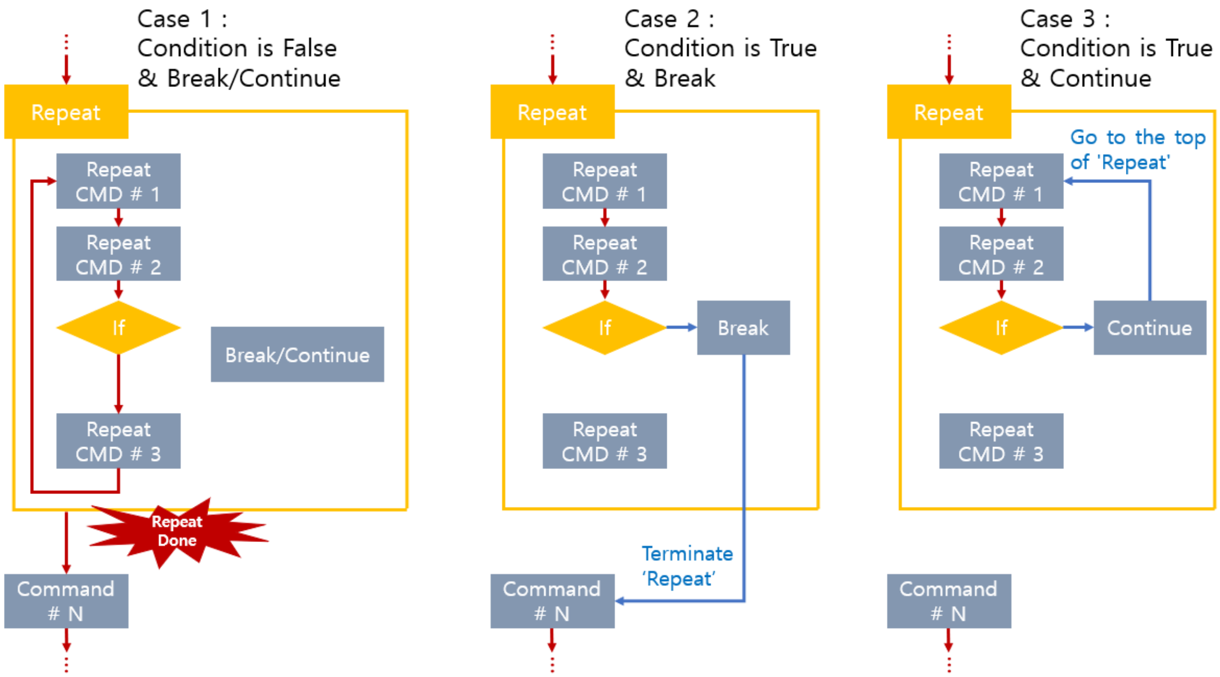
■ Break Function:

This is a function to forcibly terminate the Repeat (break) or move to the top of the Repeat (continue). Even if the Repeat condition determines that the subprogram should continue, the Break function can be used to escape the Repeat. The Continue function is used into the Repeat function, and when used, it moves to the top of the Repeat without executing the subprogram.

In addition to the break function, there are other features for controlling program flow, such as the Continue function, which moves the program flow back to the top of a loop; the Program Logic Jump To function, which allows you to jump to a specific point marked by Program Logic Jump Here; and the Program Logic Jump to Folder function, which allows you to jump to a folder in the program.

It can only be used as a subitem of the Repeat function – it cannot affect any other part of the program.

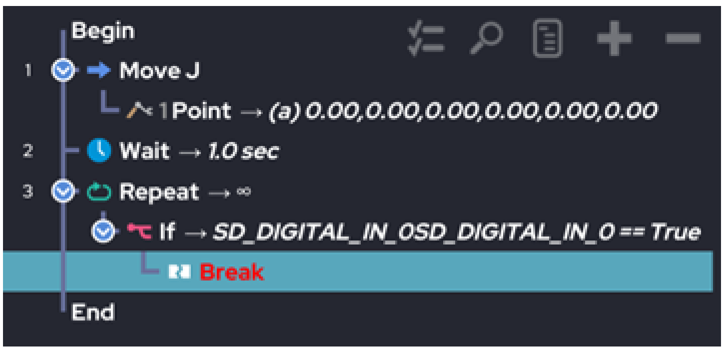




If a repeat break and continue are used, it will behave as shown in the figure above.

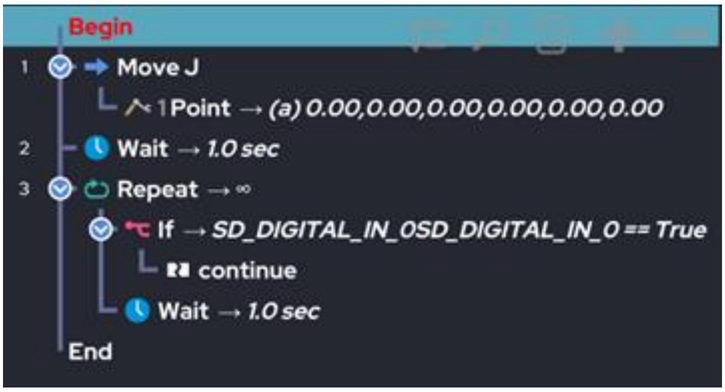
* Example of break function

In the example below, there is an infinite loop, and the program is set to execute the break function based on an If condition. If the condition is not met, the loop will continue indefinitely. However, once the condition is met, the loop will exit, and the next command will be executed.



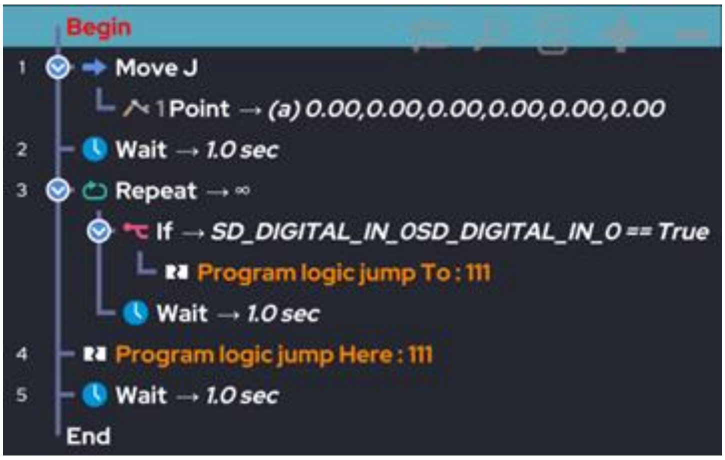
* Example of continue function

In the example below, there is an infinite loop, and the program is set to execute the continue function based on an If condition. If the condition is not met, the loop will repeat all its contents. However, once the condition is met, the program will skip the instructions below the continue function and jump to the instructions above it.



* Example of Program Logic Jump To and Here

In the example below, there is an infinite loop, and the program is set to execute the Program Logic Jump To function based on an If condition. If the condition is not met, the loop will repeat all its contents. However, once the condition is met, the program will skip the instructions below the Program Logic Jump To function and execute the commands after the Program Logic Jump Here function.

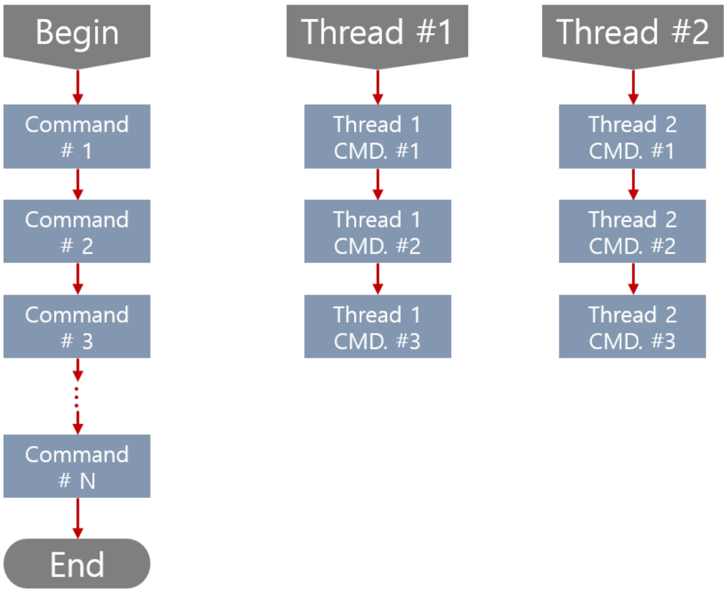


* Example of Program Logic Jump To Folder

In the example below, there is an infinite loop, and the program is set to execute the Program Logic Jump To Folder function based on an If condition. If the condition is not met, the loop will repeat all its contents. However, once the condition is met, the program will skip the instructions below the Program Logic Jump To Folder function and execute the commands inside the folder.

■ Thread Function:

Will create a separate program tree called "Thread." This program will run in parallel (at the same time) with the main program. However, the thread program tree is limited to using functions that do NOT control robot operation. In other words, the user cannot put a Move, Point, or Circle function in the thread program tree.

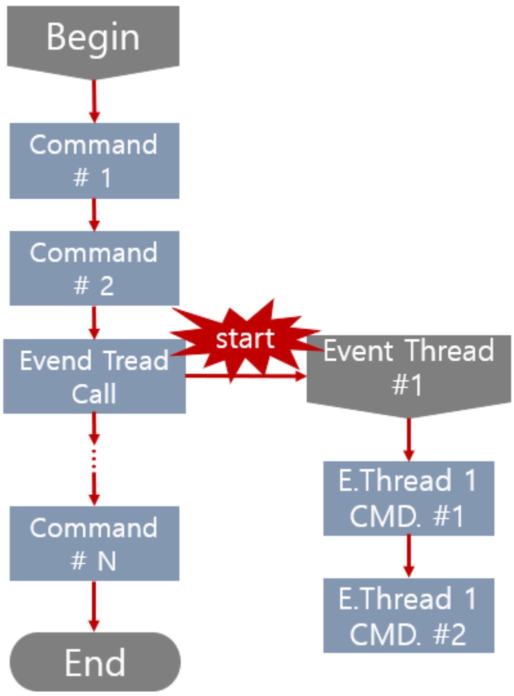


As shown above, the Thread Function is configured in parallel with the main program.

- Threads do not repeat automatically and will end when the main program ends – even if the thread has not completed. To implement a Repeat Function, highlight a command within the Thread program tree and press the Repeat icon. To implement a thread that repeats every second, use the Thread icon, use the Repeat Function within that thread, then place a one second Wait Function within the Repeat.
- The Thread Function will support only up to 3 different threads.
- Thread functionality works only in the current running program. If a subprogram called through a Sub.P function uses a thread, it will not work properly.

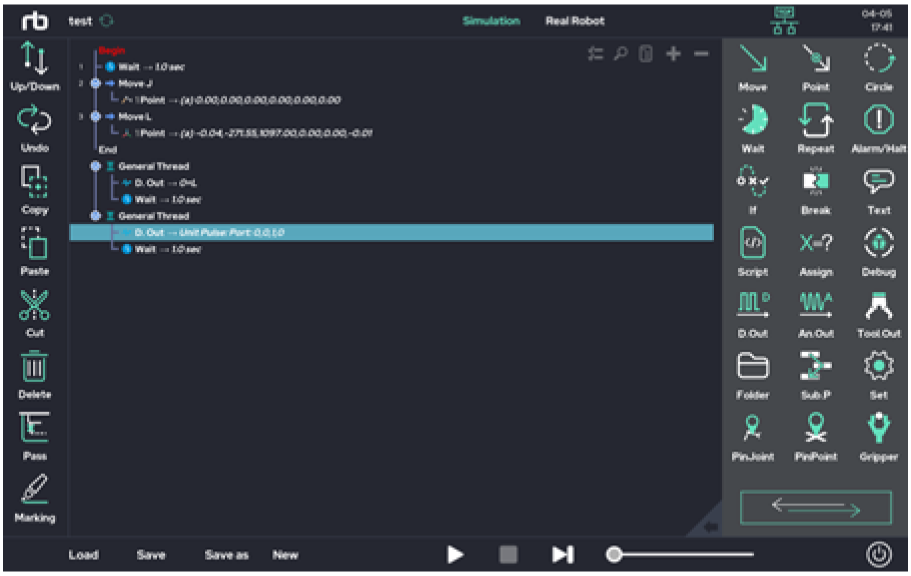
Thread types are as follows:

- General Thread: It stops with the user's intentional pause, alarm, collision detection, etc.
- Non-Stop Thread: It does not stop except for collision detection.
- Non-Stop Thread2: It doesn't stop until the program Halt.
- Event General Thread: This is a General thread executed by the event thread call function in the main program.
- Event Non-Stop Thread: It is a non-stop thread that is executed by the event thread call function in the main program.



As shown in the figure above, the event thread starts running when the event thread call function is used in the main program.

The figure below is an example of how the Thread function can be inserted into an actual project. In the example below, two threads are inserted.

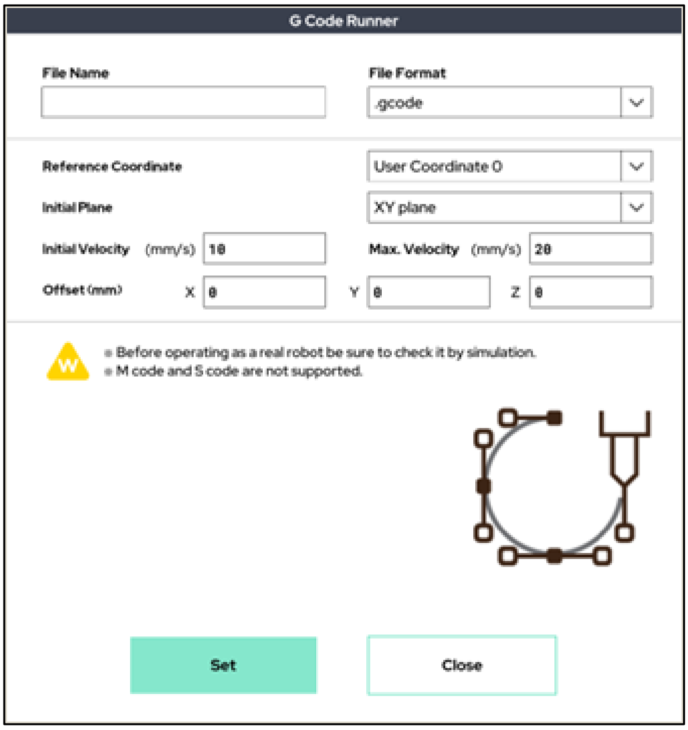


WARNING

Warning:

1) For the stability of the program, the use of threads is not recommended within any program called by Sub.P.
2) Commands such as Move or Circle Functions cannot be placed within a thread.
3) When using Pause or Alarm function, both main program and thread are paused. When the main program exits, the thread will also exit – even if the thread has not yet finished executing.

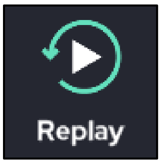
■ G Code Function:

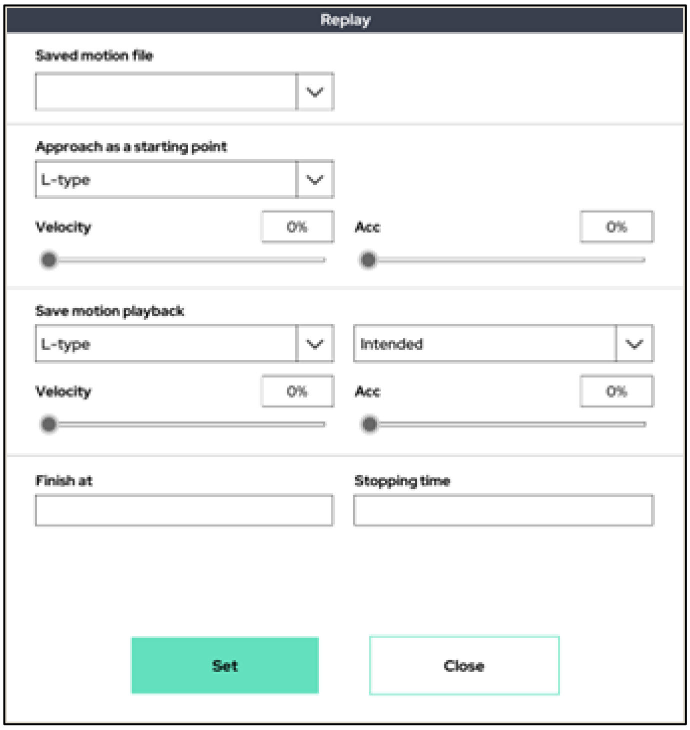


This function allows the robot to move to the path stored in the G code.

The G code file must be stored in a folder at the specified path (₩Tablet₩Android₩data₩com.rainbow.cobot₩files₩work) in advance to be available. Enter the name of the G code file that user saved in File Name. The plane in which the robot moves can then specify the xy, yz, and zx planes of the user-specified coordinate system as the starting planes.

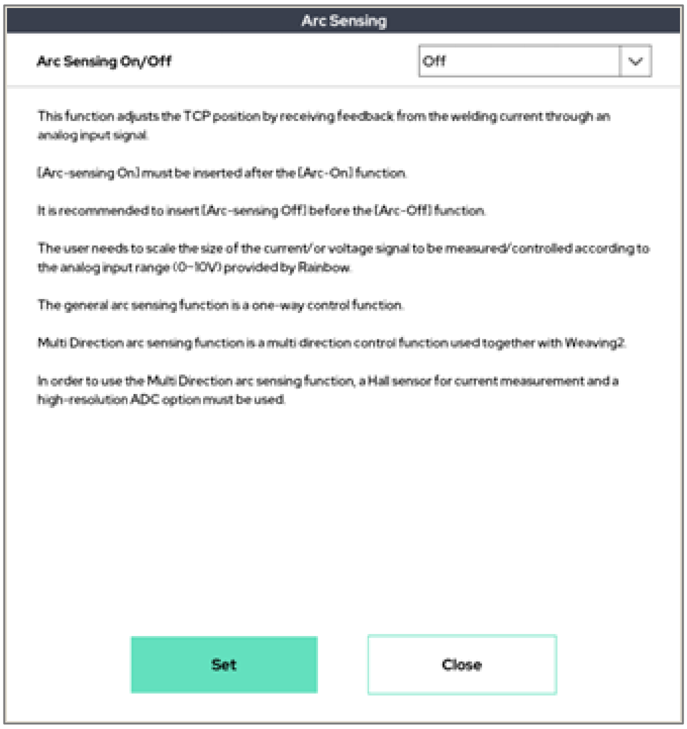
■ Replay Function:



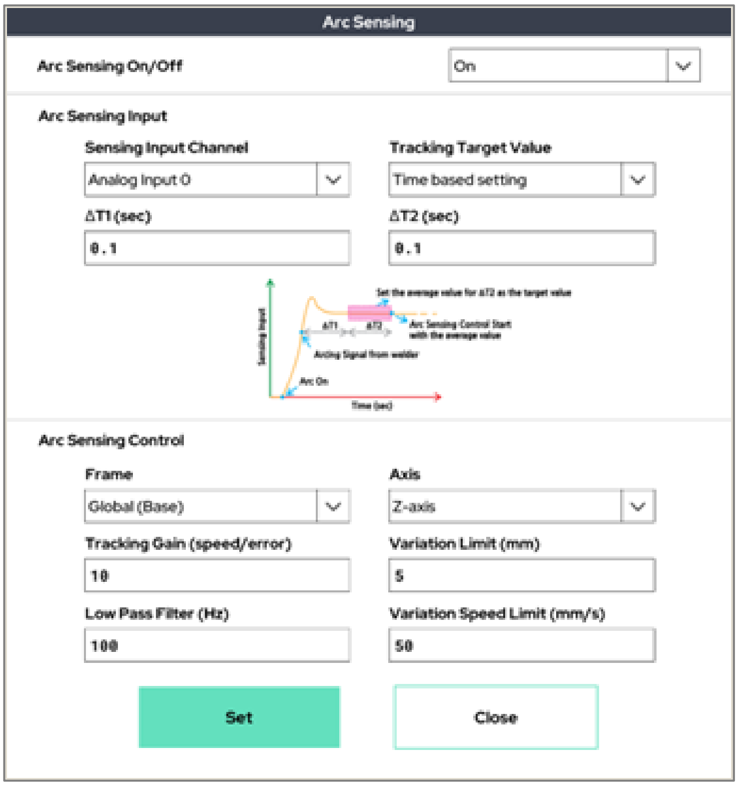


This function is to play the recorded teaching motion. Motion recording is performed in the settings of the Make page. If you select the name and motion speed/property of the recorded motion, the recorded motion is played again.

■ ArcSensing Function:



Arc sensing is a function designed to maintain welding quality by using the current of the welding machine to compensate for issues caused by the misalignment of the workpiece or inaccuracies in teaching during welding. Arc sensing is divided into unidirectional arc sensing and multidirectional arc sensing. For use with weaving, multidirectional arc sensing should be applied.



In unidirectional arc sensing, you can set the average or target value of the input coming from the analog input over a specific time period. Based on the input values, parameters can be adjusted to determine the direction and degree of correction.

In multidirectional arc sensing, the target value is set as the average of the input from the analog input over a specific time period. You can configure how much correction is needed to follow the target value in both the robot's movement direction and the weaving direction. For advanced functions, you can adjust the responsiveness of arc sensing and add intentional offsets by configuring the corresponding settings.

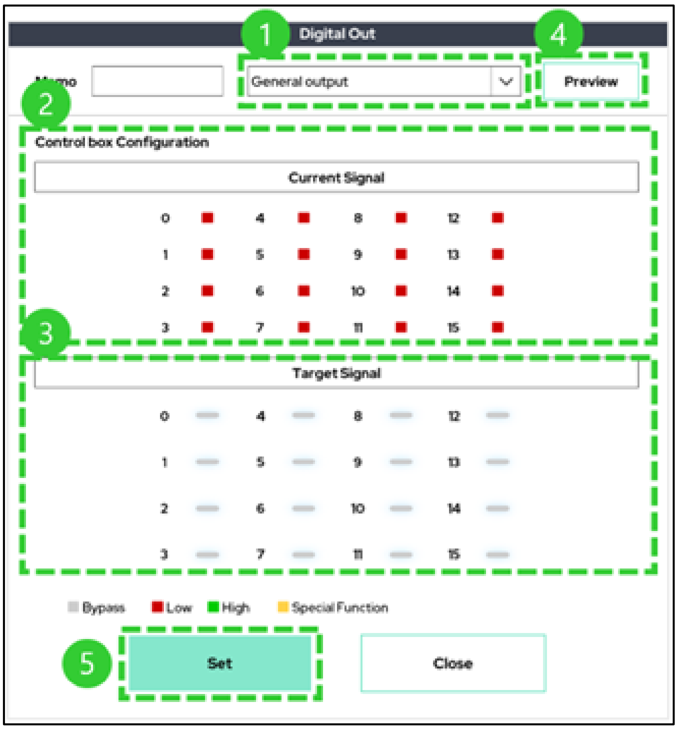
■ Motion Macro (M.Macro) Function:

The system automatically generates actions based on the type of macro. (Flat plane, Rounded Plane(R), Rounded Plane(H), Triangle Plane, 3D Round Plane, Expanding Spiral(M), Expanding Spiral(P), Shrinking Spiral) At this time, teaching must be performed according to each macro, and once the user sets the parameters to match the desired shape, the desired actions can be automatically generated without needing to teach individual points.

■ D.Out (Digital out) Function:

Allows the user to set the digital output of the control box. The user can set the digital output signal of whichever port (0~15) they would like. Each port has three possible settings: high signal, low signal, and bypass.

After adding the D.Out function to the program, click on D.Out in the program tree to have the following pop-up window appear.



① Selection the detailed features available in the D.out function.

② Shows the status of the current Digital Out output from the control box.

③ Allows the user to set their desired setting for a port (0~15). The three setting toggles are Bypass, Low and High.

Bypass: Maintains the previous output signal state (gray).

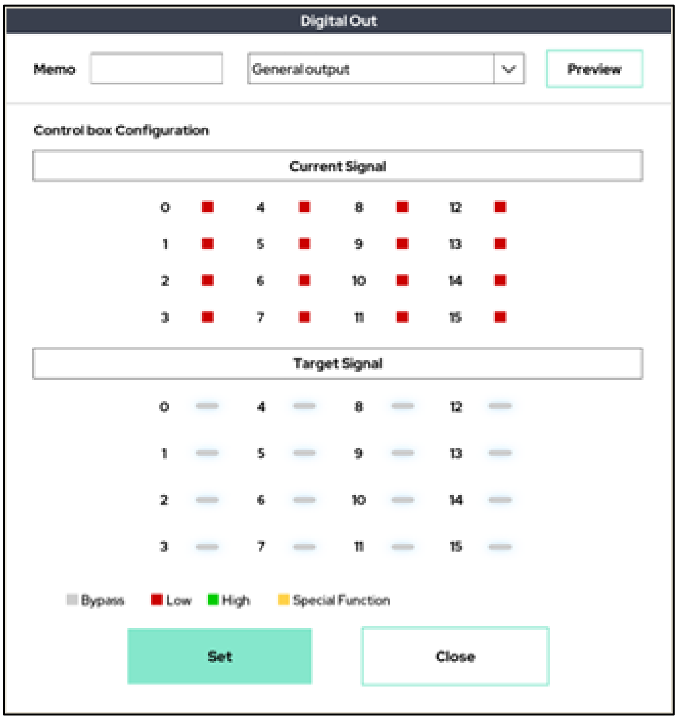
Low: Sets the output signal to the low (0) level (red).

High: Sets the output signal to the high (1) level (green).

④ Allows the user to review the settings selected within the target signal menu. A further explanation is shown below.

⑤ Saves the settings specified within target signal menu.

- D.Out : General output

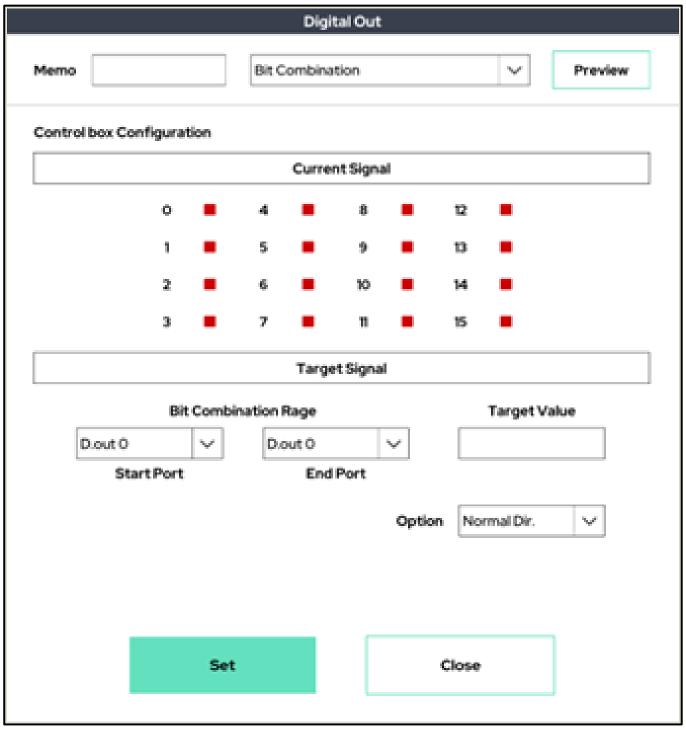


With the control box connected to the teaching pendant, set the Target Signal menu as shown above (to the right). Then, press the Preview button. As shown above, the Current Signal menu will change to match the settings that the user has put in the Target Signal menu.

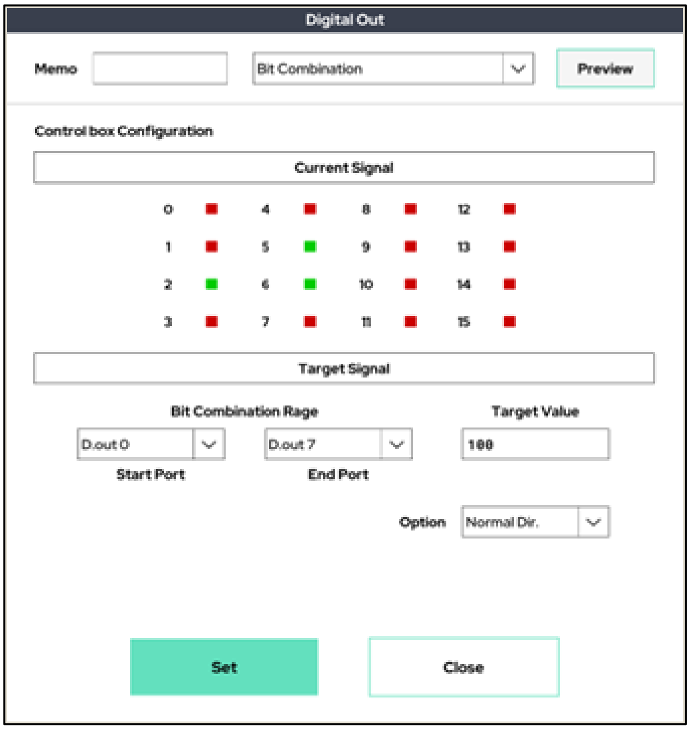

Digital Out
Memo
General output
Preview
Control box Configuration
Current Signal
Target Signal
Bypass
Low
High
Special Function
Set
Close

- D.Out: Bit Combination

You can select the start and end ports, then input the desired value into the target field to output a digital signal using a bit combination. Here, the start and end ports represent the bit positions. For example, if you set the start port to port 0 and the end port to port 3, it corresponds to a 4-bit signal. When you input a decimal value in the target field, it will be converted into its binary equivalent, and the corresponding ports will turn on based on the value.



When the control box is connected to the tablet PC, you can configure the target signal field as shown in the figure below, and by pressing the preview button, the control box will output the digital signal as displayed.



As shown in the above figure, if you choose the "forward" option, port 0 will be the LSB and port 7 will be the MSB. Converting the decimal value 100 to binary gives 0110 0100, which means ports 2, 5, and 6 will turn HIGH. Conversely, if you choose the "reverse" option, port 0 will be the MSB and port 7 will be the LSB, resulting in ports 1, 2, and 5 turning HIGH.

- D.Out: Signal Toggle

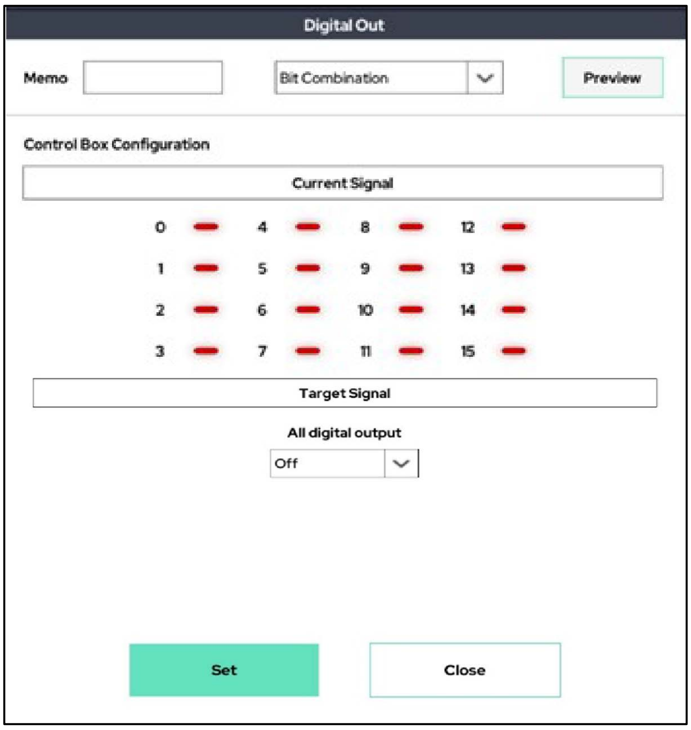
The signal will be toggled, meaning signals that are HIGH will become LOW, and those that are LOW will become HIGH.

With the control box connected to the tablet PC, you can configure the target signal field as shown in the figure above (toggled signals are displayed in blue). By pressing the Preview button, the digital output signal from the control box will be toggled, as shown in the following figure.

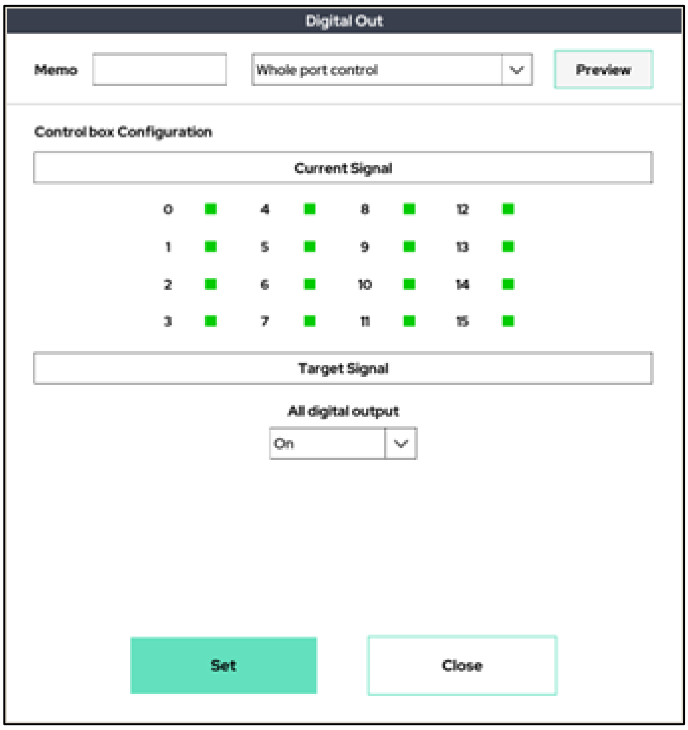

Digital Out
Memo
Signal Toggle
Preview
Control box Configuration
Current Signal
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Target Signal
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Bypass
Signal Toggle
Special Function
Set
Close

- D.Out: Whole port control

You can switch all output ports to HIGH or LOW at once.

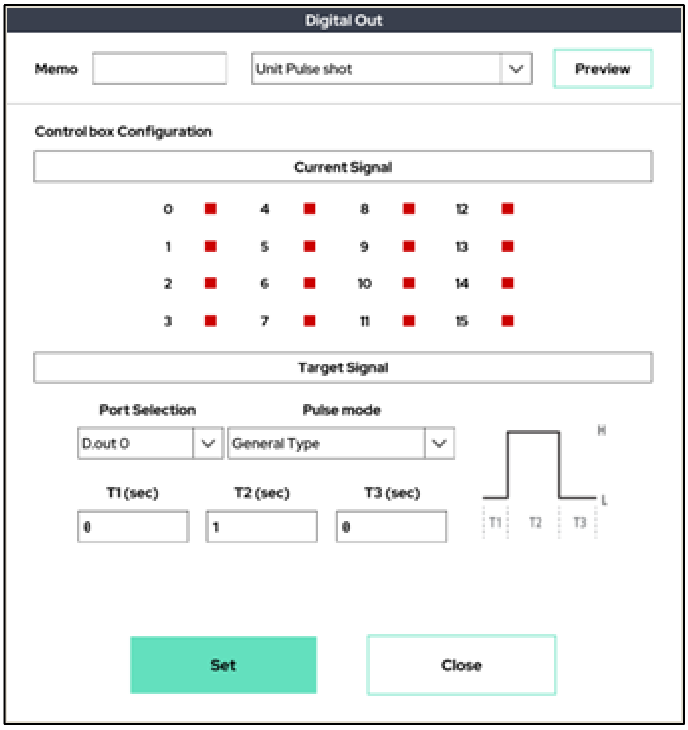


When the control box is connected to the tablet PC, set the target signal field as shown in the figure above, and press the Preview button. The control box will output the digital signal, and you will be able to see that the signal has changed, as shown in the figure below.

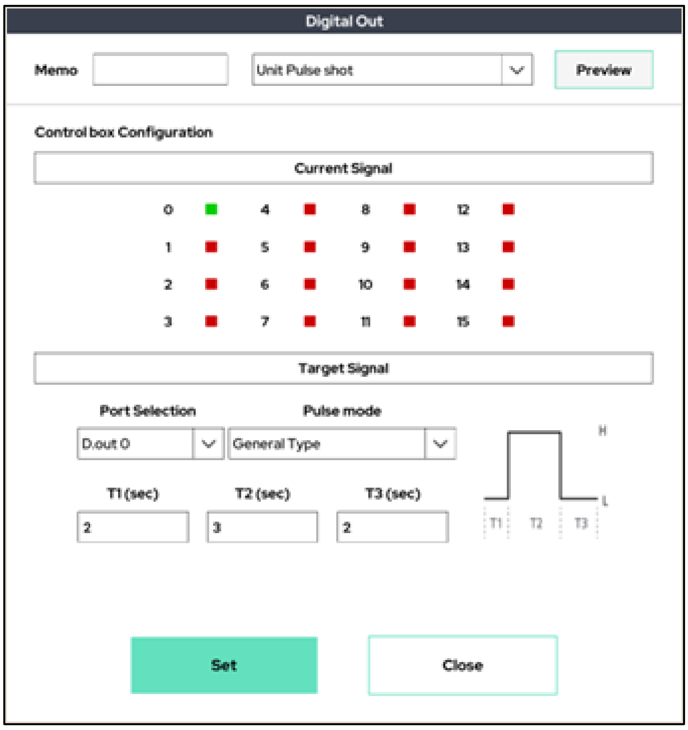

Digital Out
Memo
Whole port control
Preview
Control box Configuration
Current Signal
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Target Signal
All digital output
On
Set
Close

- D.Out: Unit Pulse shot

You can select the port you want to use and input a time between 0 and 3 seconds for T1 to T3. The pulse signal will be output according to the set time.



When the control box is connected to the tablet PC, configure the target signal field as shown in the figure above and press the Preview button. The control box will then output a digital signal.

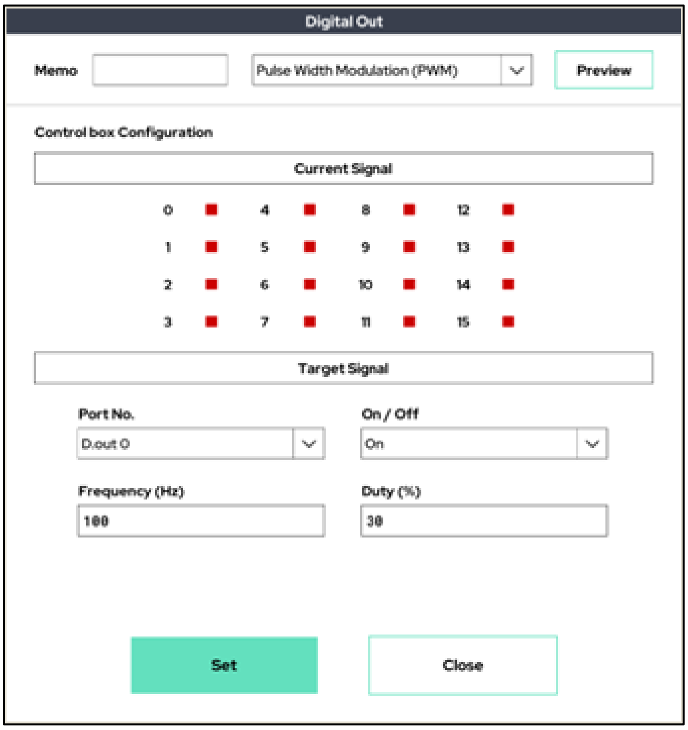


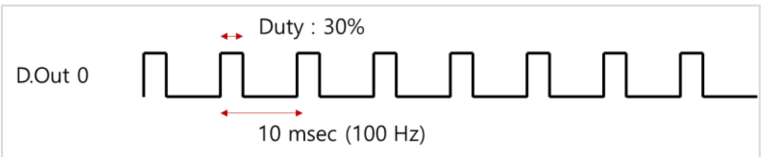
If you set T1 to 2 seconds, T2 to 3 seconds, and T3 to 2 seconds, the pulse will be generated for each set time when previewed. Additionally, if you set the pulse mode to "normal pulse," the LOW-HIGH-LOW signal cycle will be set, as shown on the right. In "inverted pulse" mode, the HIGH-LOW-HIGH signal cycle will be set. Both normal and inverted pulses have a separate Non-Block mode. In Non-Block mode, the next command will be executed in parallel while the pulse is being output. If Non-Block mode is not used, the unit pulse output process will wait for the full cycle to complete before proceeding to the next command.

- D.Out: Pulse Width Modulation(PWM)

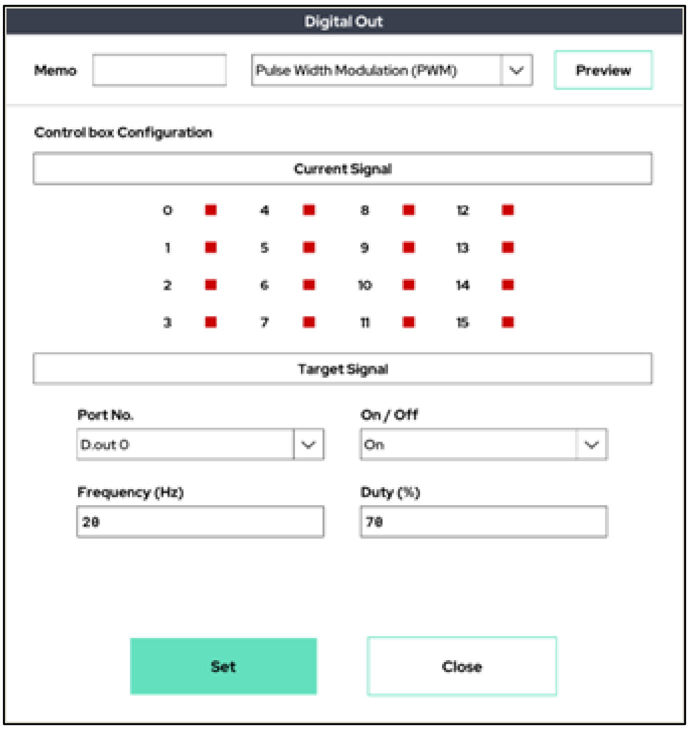
This is the PWM (Pulse Width Modulation) output function. By inputting the frequency and duty cycle of the PWM pulse, you can send a PWM signal through the digital output port.

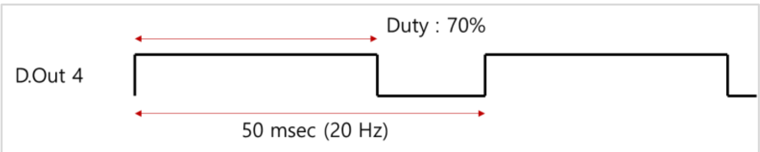
PWM Ex 1)





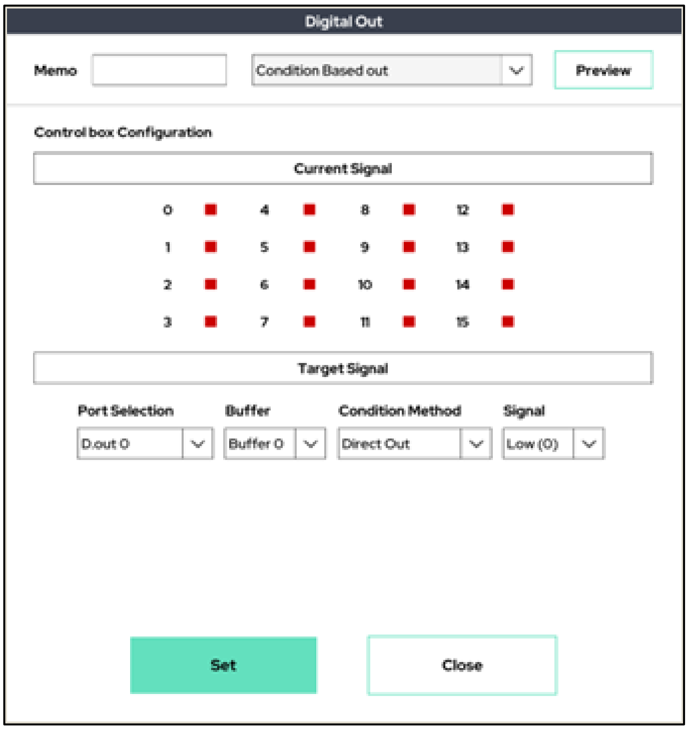
PWM Ex 2)


Digital Out
Memo
Pulse Width Modulation (PWM)
Preview
Control box Configuration
Current Signal
0 4 8 12
1 5 9 13
2 6 10 14
3 7 11 15
Target Signal
Port No.
D.out 0
On / Off
On
Frequency (Hz)
20
Duty (%)
70
Set
Close

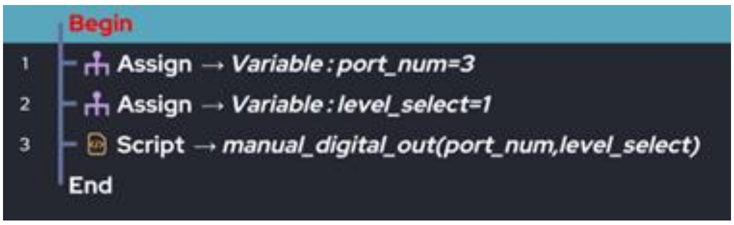

Duty : 70%
D.Out 4
50 msec (20 Hz)

- D.Out: Condition Based out

In addition to using the D.Out function, you can create commands to send digital outputs using the script function, as shown below.



※ Script function: manual_digital_out(port number, output level)

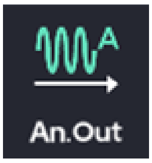


WARNING

Warning:

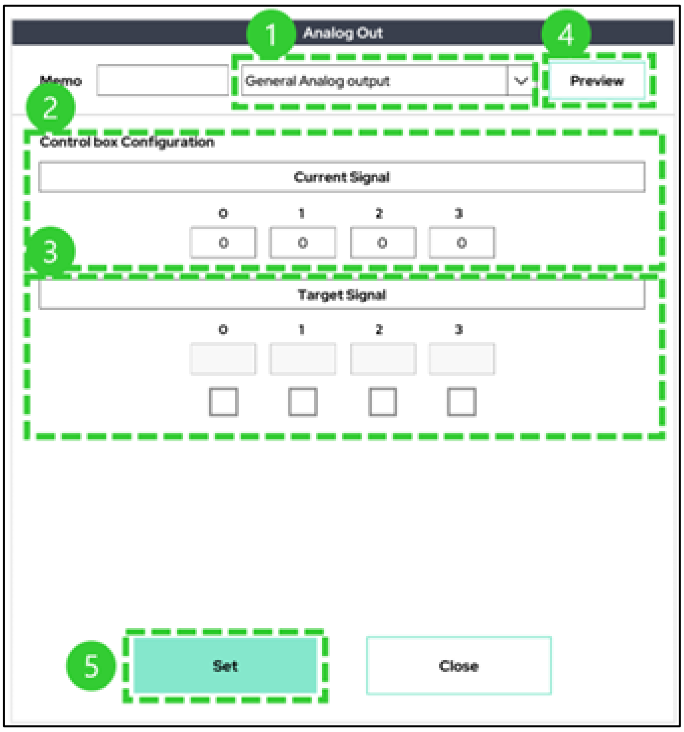
1) If a special function is assigned to a specific digital output port in Setup-I/O, that port is not available through the D.out function.
2) If a special function is assigned to a specific digital output port, it will be indicated in color yellow.
3) If you want to leave a comment about the D.out function you set, you can use the memo function at the top right of the popup window.
4) Before using the digital output, please fully understand the electrical properties of the digital output port provided by the manufacturer.

■ An.out (Analog out) Function:



The Analog Out Function controls the analog output of the control box. Outputs the selected voltage through the target (0~3) analog ports. Each port can output a voltage range of 0~10V.

After adding An.Out to the program, click on An.out in the program tree to open the following popup window.



① You can configure the features provided in An.Out.

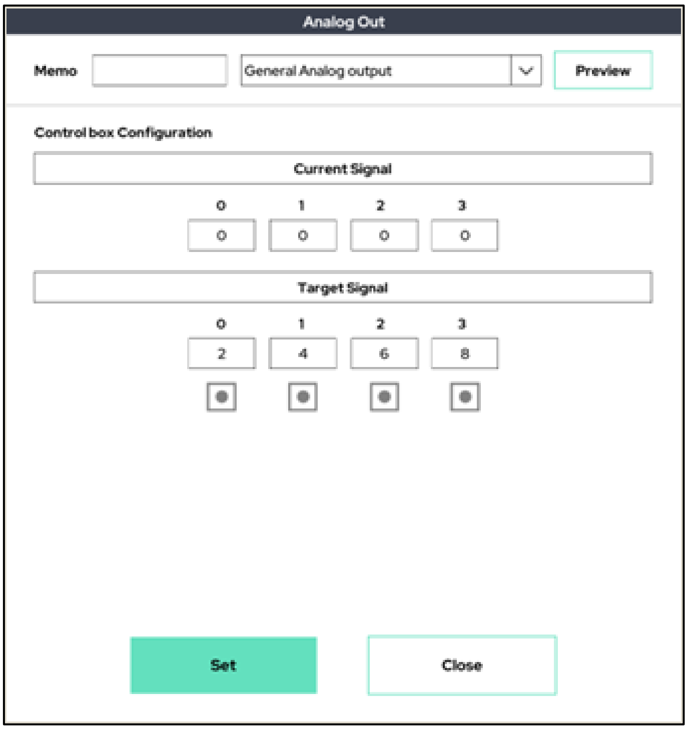
② Displays the current status of the Analog Out being output by the control box.

③ This setting maintains the existing voltage output. Check the checkbox and input the desired voltage (0-10V) to set the voltage.

④ This function allows you to preview the settings selected in item 3.

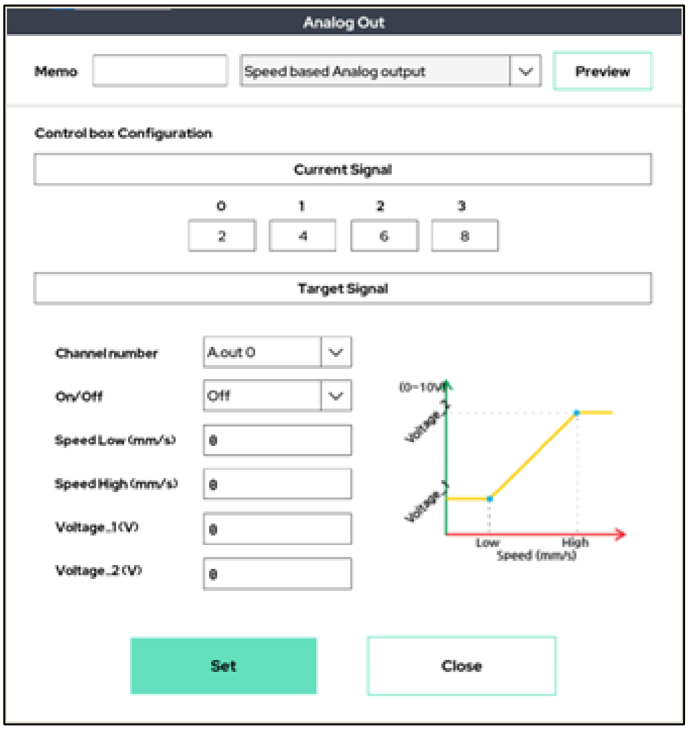
⑤ This configures the An. Output function with the settings selected in item 3.

- An.Out: General Analog output



When the control box is connected to the tablet PC, set the Target Signal field as shown in the figure above, and press the Preview button. The control box will output the analog signal, as shown below. You can see that the Current Signal field on the left side of the image has changed.

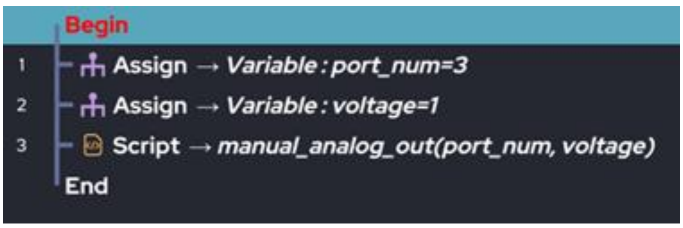
- An.Out: Speed based Analog output



Sets speed-based analog output for the analog output channel corresponding to the channel number. When you set On in the On/Off setting, the output voltage is based on the TCP speed. If the TCP speed is lower than the speed set in Speed Low, the output voltage is set to the value in Voltage_1. If the TCP speed is between Speed Low and Speed High, the output voltage increases proportionally to the TCP speed, up to the value set in Voltage_2. The output voltage, increasing proportionally with TCP speed, is limited by the voltage set in Voltage_2.

In addition to using the An.Out function, you can also create commands to send analog output using the script function, as shown below.

※ **Script** function: manual_analog_out(port number, output voltage)



WARNING

Warning:

1) If you want to leave a comment about the A.out function you set, you can use the memo function at the top right of the popup window.
2) Before using the analog output, please fully understand the electrical properties of the analog output port provided by the manufacturer.

■ Tool Out Function:

The tool flange has two digital outputs. Signals from two digital outputs can be specified. In addition, the level of voltage to be output from the tool flange (0V or 12V or 24V) can be adjusted together. Click the Tool Icon to add it to the program. Click on Tool in the program tree to have the following pop-up window appear.

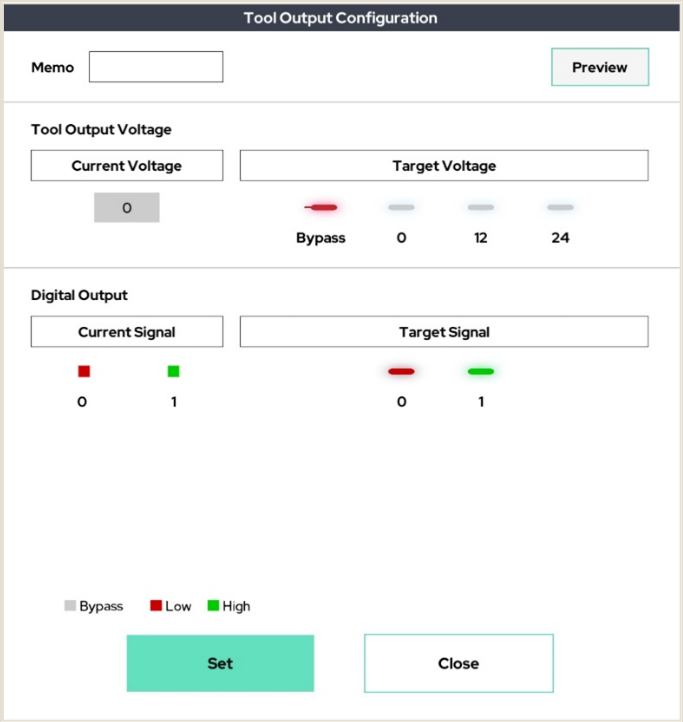
① Shows the current status of the tool flange output at the end of the robot.

② Sets desired voltage and digital output. The output voltage can be selected between 0V, 12V, and 24V. There is also an option to Bypass. The digital output can be toggled between Bypass, Low, and High.

③ Allows the user to preview the settings selected within the target signal menu. A further explanation is shown below.

④ Saves the settings specified within target signal menu.

With the control box connected to the teaching pendant, and after activating the robot, set the Target Signal menu as shown above. Then press the Preview button to preview the tool flange output signal. The Current Signal menu will change to match the settings that the user has put in the Target Signal menu.

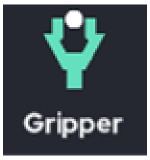


WARNING

Warning:

1) The user can add a comment about the Tool.out function by using the memo function at the top left of the popup window.
2) Before using the tool flange output, please fully understand the electrical properties of the port provided by the manufacturer.

■ Gripper Function:



This is a dedicated function for the gripper dedicated to cooperative robots. It is possible to conveniently test and insert into the program and use of cooperative robot grippers from various companies such as Robotiq's grippers. It is not a simple I/O method, but it is a function that helps users to use a gripper that is cumbersome to write by using serial communication such as RS485 or using CRC.

Add the gripper function to the program tree and click the added Gripper as below.

① Select the gripper product.

② Select gripper connection point(Control Box, Tool Flange).

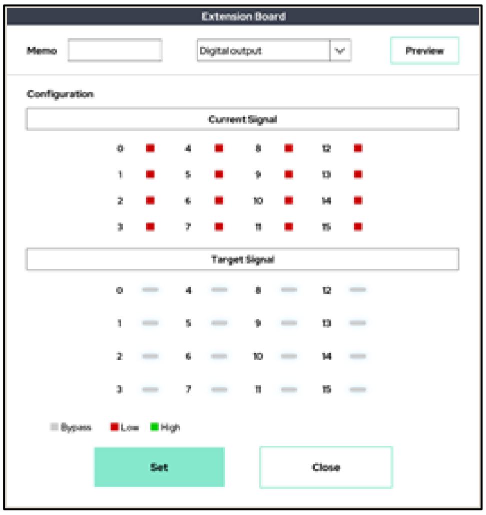
③ Select the function to be used as the gripper.

WARNING

Warning:

1) The product list provided in the gripper function will be updated through user request.

■ Extension Board (I/O Extend) Function:



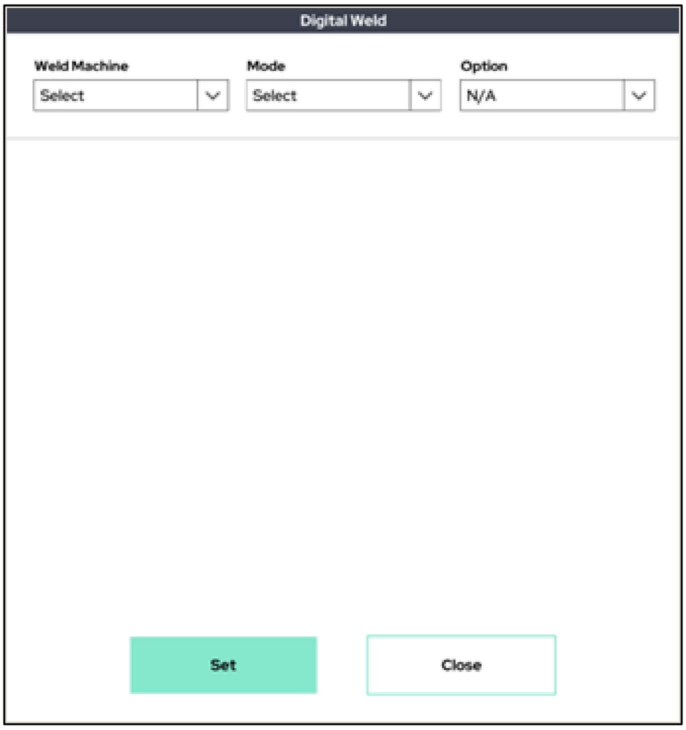
A feature that controls digital/analog output when purchasing and using an extended I/O module. The method of use is the same as the existing D.output and An.output.

■ ArcWeld Function:

This is a special function for arc welding. A special macro function designed to quickly enable implementable functions, such as Wait/D.out. To use this function, the Device field on the Setup page must precede setting the parameters and connection information for the welder.

As illustrated above, this feature allows quick and easy insertion of weld speed/weld current / voltage settings / safety signal processing options into the program to be used for welding.

■ D.Weld Function:

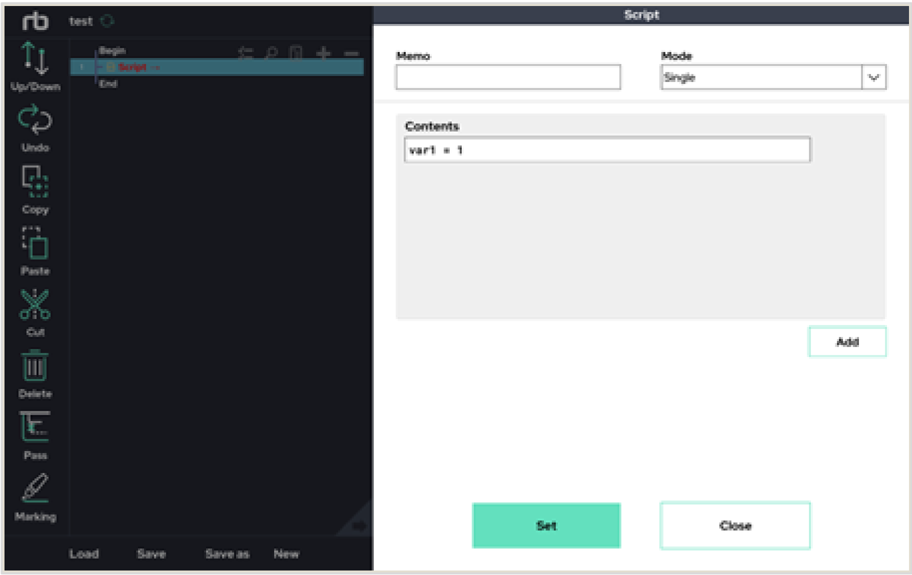


It is a function that can use the digital weld machine. After selecting the weld machine to be used, user can proceed with ‘Weld Start’, ‘Weld Off’, and ‘Weld Setting’.

■ Script Function:

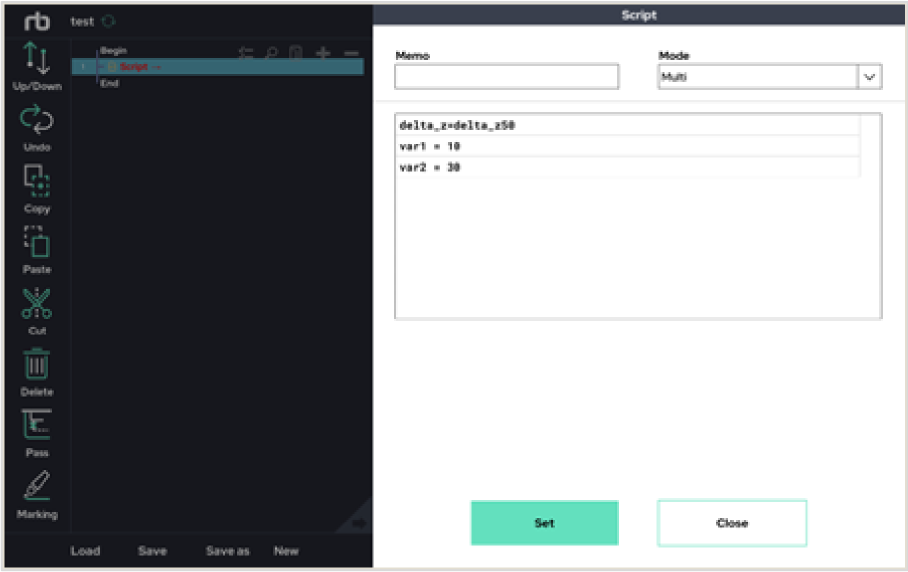
Allows the user to write custom scripts. These scripts allow for custom operations/calculations. The Script Function also allows for functions such as variable substitution and assignment. There are two modes: single-input mode and multi-input mode.

Add the script function to the program tree and click the added script function. The following popup window will appear.



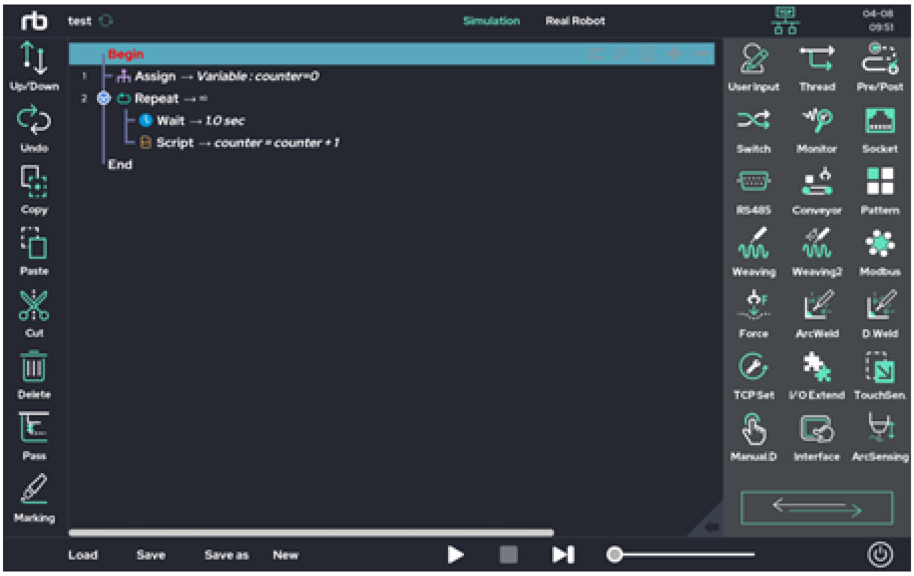
You can freely input the desired script here. Functions such as variable operations, variable replacement, and variable assignment can be freely entered using the keyboard. If you want to use multiple lines of the script function at the same time, you can use the Add button at the bottom of the popup window.

When you select the mode as Multiple Input, a popup window like the one below will appear.



You can use multiple scripts without needing an additional button.

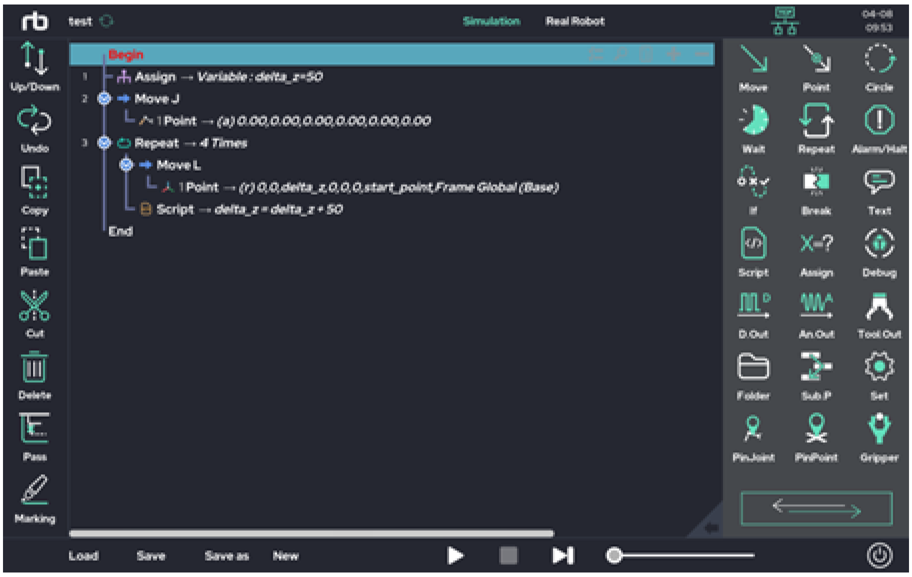
The example below shows a program that creates a loop (Repeat) that runs every 1 second. During each repetition, a variable called counter is incremented by 1 using the Assign function within the script.



In the example below, the robot moves to a specific pose using the Move J function, and the name start_point is assigned to this pose (using the function to name points). Then, a loop (Repeat) is used to move with the Move L function four times.

- Here, the Move L function uses the Relative Point feature to move by delta_z in the z-direction relative to the previously declared start_point (refer to the relative point function of the linear movement family in the point functions).
- At the end of the loop, the delta_z value is increased by 50 using the Script function (the delta_z variable is declared earlier in the program using the Assign function).

In conclusion, the robot moves to the initial position with Move J, saves that position as start_point, and then moves down by 50mm four times using Move L in the loop.

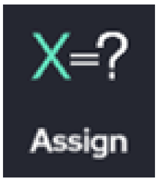


WARNING

Warning:

1) The script function is an area where the user can freely write and execute a script.
2) If the users write a script that doesn't match the syntax, the program may malfunction or stop. Be mindful and use the proper syntax when using this feature.

■ Assign Function:



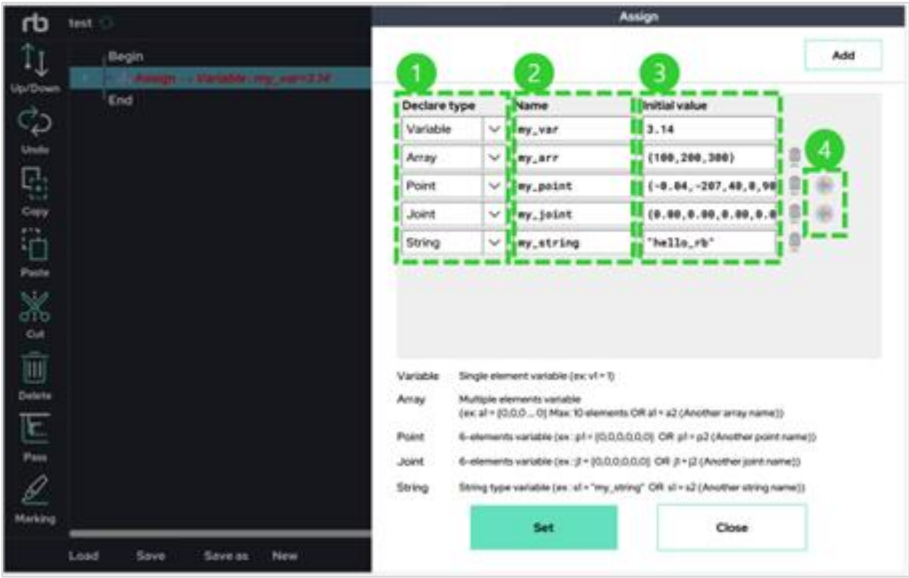
Declare and designate the value of a variable. Variables can be changed through the program to allow for greater flexibility with conditionals.
A variable can be one of the 5 following types:

- Variable Type: Saves a single numerical (float) variable.
- Array Type: Saves multiple values in a list. The maximum length is 10.
- Point Type: Saves position information (saves x, y, z, Rx, Ry, Rz).
- Joint Type: Saves joint angle information (saves J0, J1, J2, J3, J4, J5).
- String Type: Saves a string (alphabetic and numerical characters - e.g. “ASDF1234”).

When the Assign function is added to the program tree, it will look as shown below.

To assign a variable, click on Assign and a popup will appear. Then, the variable can then be assigned within the popup. Multiple variables can be declared by clicking the Add button. To save the variable, click on the Set button. If multiple declarations are made, the program tree will show how many variables of each type were declared.
If a declaration is made, the variable name and initial value will be displayed on the tree as shown below.

An example popup window of the Assign function is shown below.



① Declares the type of variable (Variable, Array, Point, Joint, String).

② Sets the name of the variable.

③ Sets the initial value during the declaration.

For the Variable Type, the initial value is set as a single number (e.g. 1).

For the Array type, place initial values within curly braces (e.g., {100, 200, 300}).

For the Point, Joint type, use curly braces around the initial values, which will be in the form of an array of six lengths, (e.g. {300, 300, 300, 0, 90, 0}).

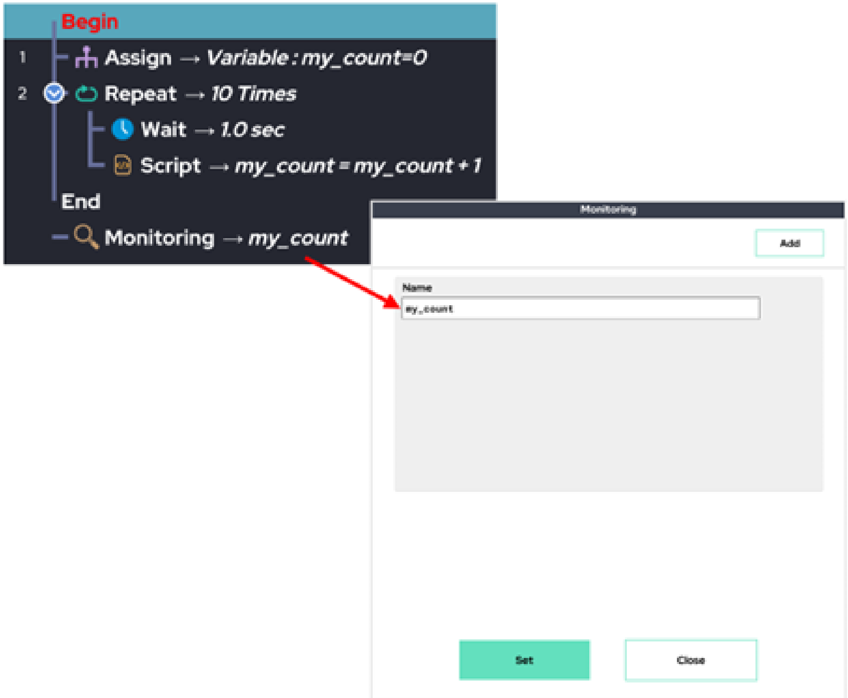
For the String type, put use quotations around the string for the initial value (e.g., “hello_rb5”).

④ Button for the Point/Joint type.

The six coordinates (Point: x, y, z, Rx, Ry, Rz / Joint : J0, J1, J2, J3, J4, J5) of the current robot configuration are imported as initial values.

■ Monitor Function:

This function is used to select variables (single variables, arrays, point variables, etc.) that the user wants to observe in real time while the program is running. Variables declared in the Monitor function can be viewed by clicking the monitor icon on the right side of the Make / Play page.

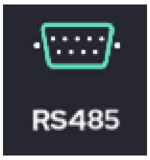


In the program example above, a variable named 'my_count' is declared. The Repeat function increments 'my_count' by 1 every second. By using the Monitor function, the user can select the 'my_count' variable as the object to observe. As shown in the above image, in the Monitoring window, the user can enter the name of the variable to be observed.

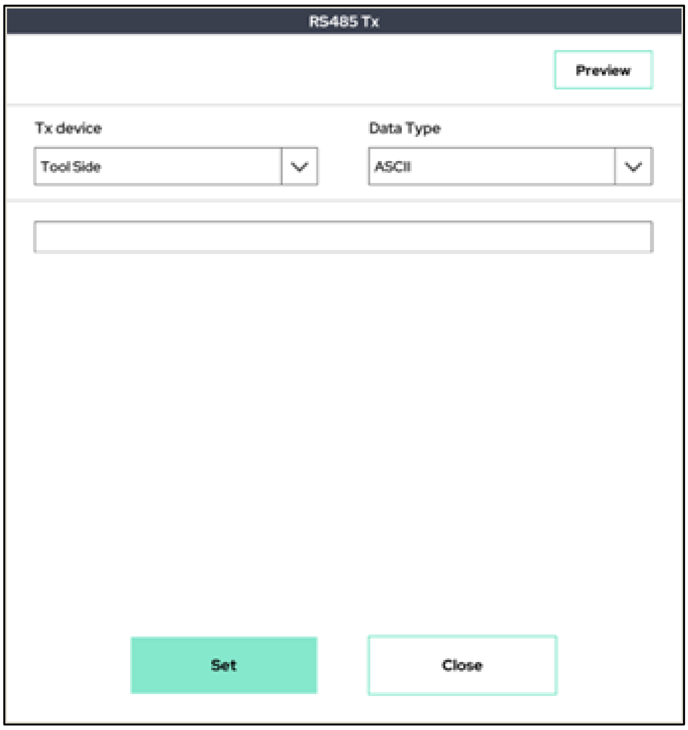
If the user wants to observe the value of the monitored variable, they can click the Monitor icon on the right side of the screen.

After that, if the user presses the play (▷) button, they can observe the value of ‘my_count’ increasing every second.

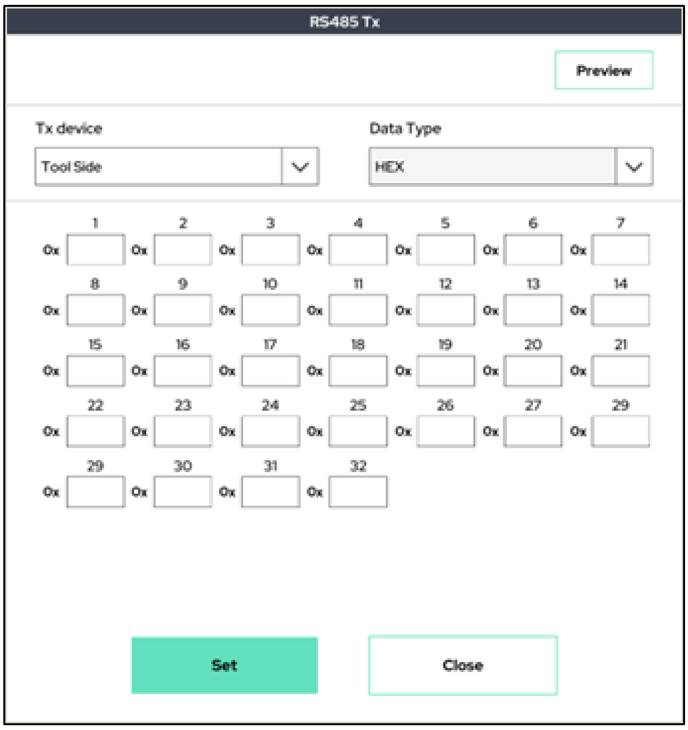
■ RS485 Function:



This function allows the user to set the RS485/232 output for the tool flange or the control Box. Users can output in ASCII mode, or in HEX mode. The UI Tablet (Teaching pendant) only supports UI485 Tx. The configuration can be previewed through the Preview button on the right side of the popup window.



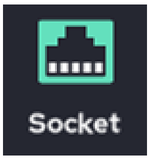
[ASCII mode]



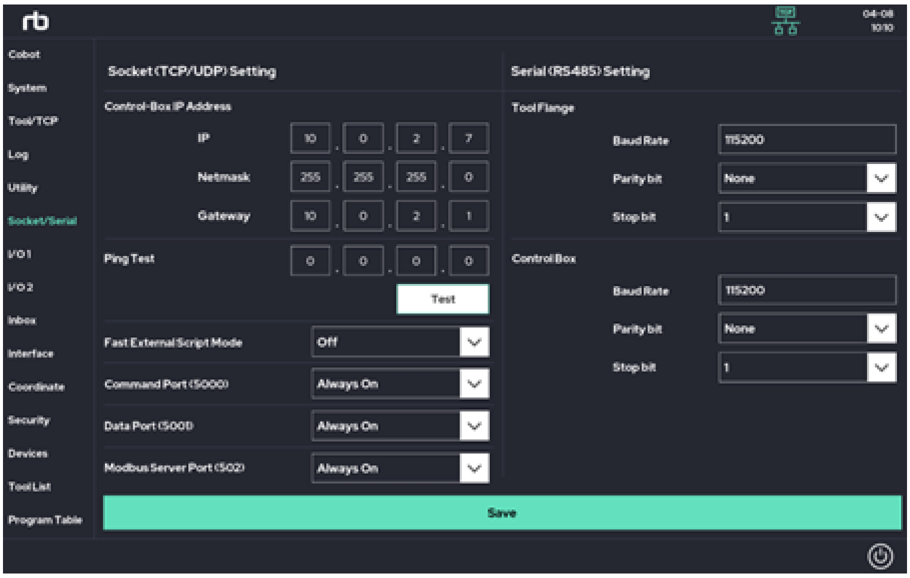
[HEX mode]

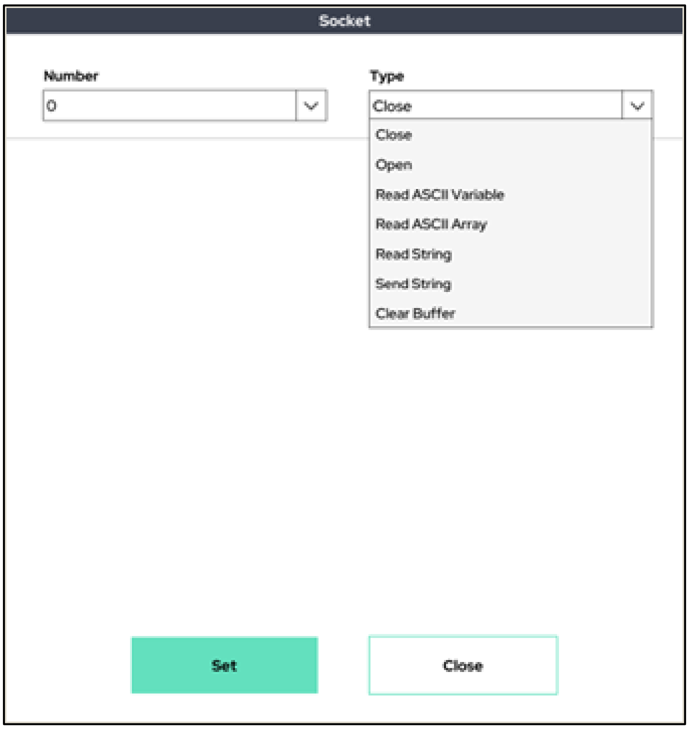
Baud rate and other protocols (Parity bit, Stop bit) for use in Serial-Communication can be set in Setup-Socket/Serial menu. Alternatively, the user can use the Set-Serial Configuration option at the top of the project. To use serial communication on the box side, plug a commercially available USB-Serial (RS232 / 422/485) device into the USB port.

■ Socket Function:



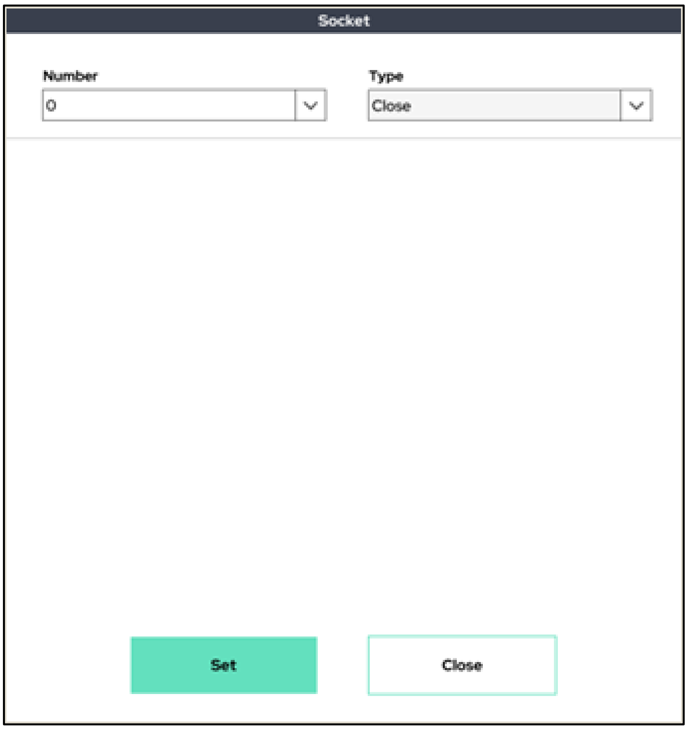
The Socket Function allows for socket communication. It provides the user the ability to open sockets to connect, send request messages, and retrieve data to/from specific server. Socket communication can be connected to at most 5 separate servers. The Socket Function uses the IP settings as defined in the Setup screen. A user that would like to change the IP settings can go to the Setup-Socket/Serial screen. The Socket Function provides six different options as follows.





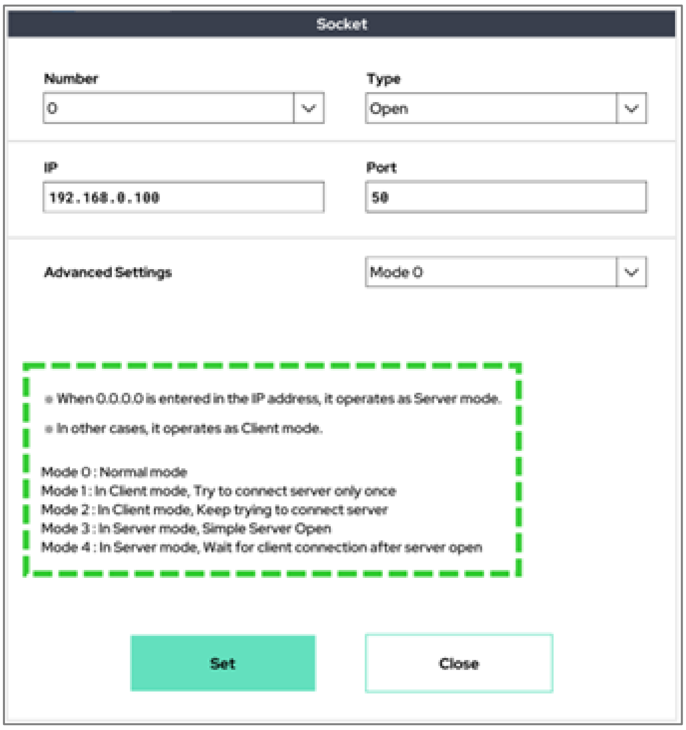
- Close: Closes the socket.
- Open: Opens socket and connects with server.
- Read ASCII Variable: Reads a value sent from the server. The user will need to choose a variable to be overwritten with the received value.
- Read ASCII Array: Reads an array sent from the server and puts it into an array type.
- Read String: Reads a string from the server and puts it into a string type.
- Send String: Send the specified string to the server.
- Clear Buffer: Empty the buffer.

- Socket Function: Close



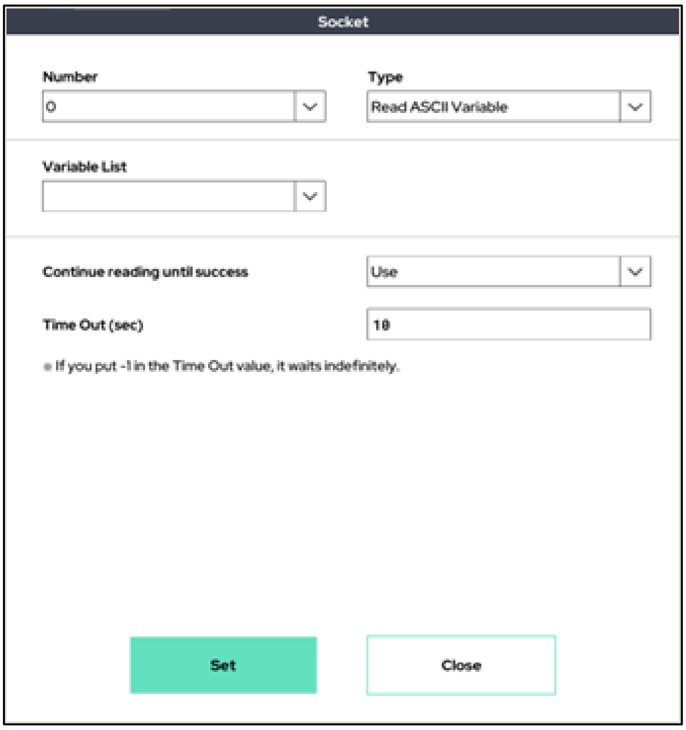
This option closes the selected socket (0~4).

- Socket Function: Open



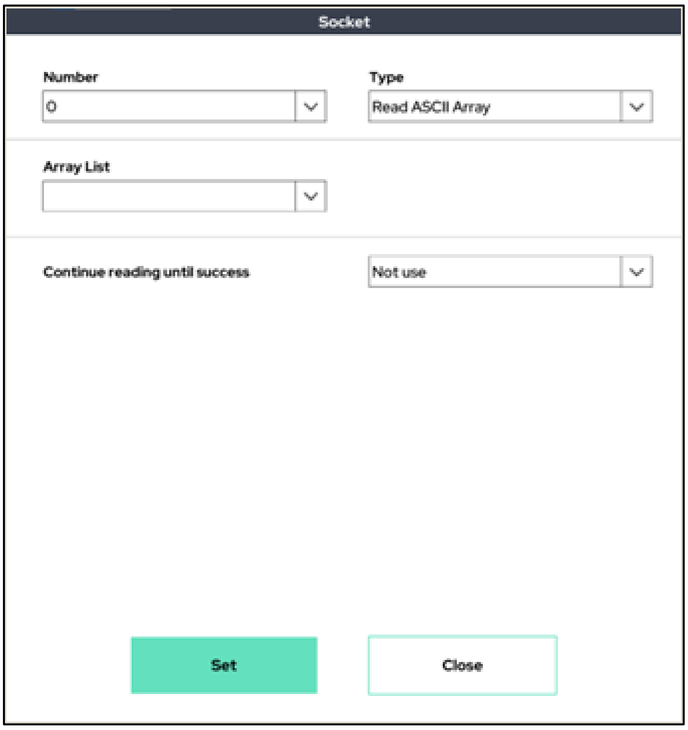
Opens the selected socket (0~4) and connects to the partner server. This option requires the user to set the IP address and port number of the server they would like to connect to.

- Socket Function: Read ASCII Variable



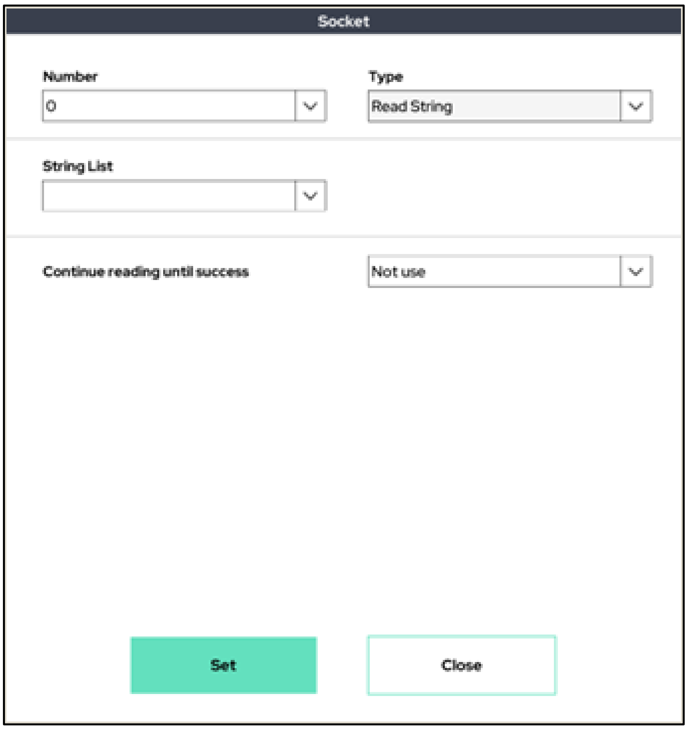
Allows the user to select one predefined variable (from the Assign Function) and overwrite the value of that variable with a value received from the server. If you set Continue Read to Enable until successful, you will set a timeout time, which will stop reading after that time.

- Socket Function: Read ASCII Array



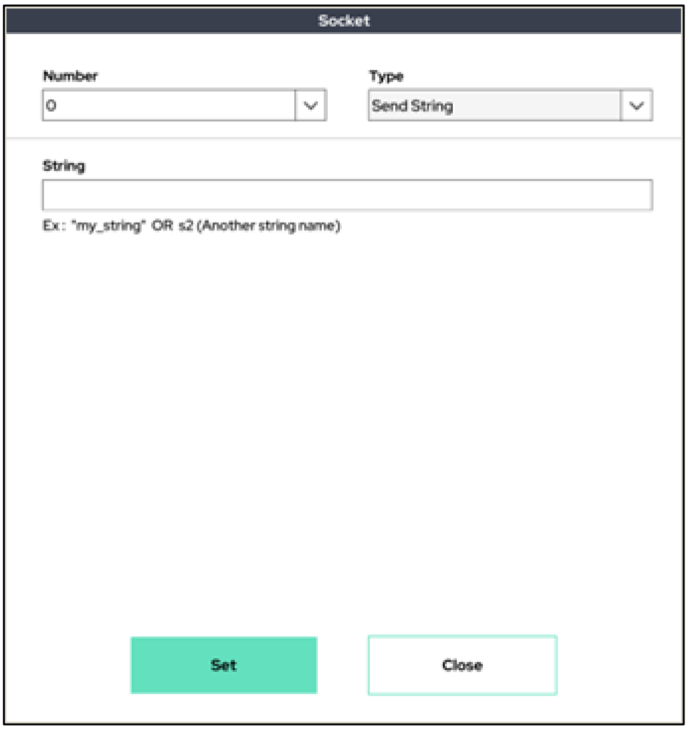
Allows the user to select one predefined array (from the Assign Function) and overwrite the values contained within that array with the values of an array sent by the server.

- Socket Function: Read String



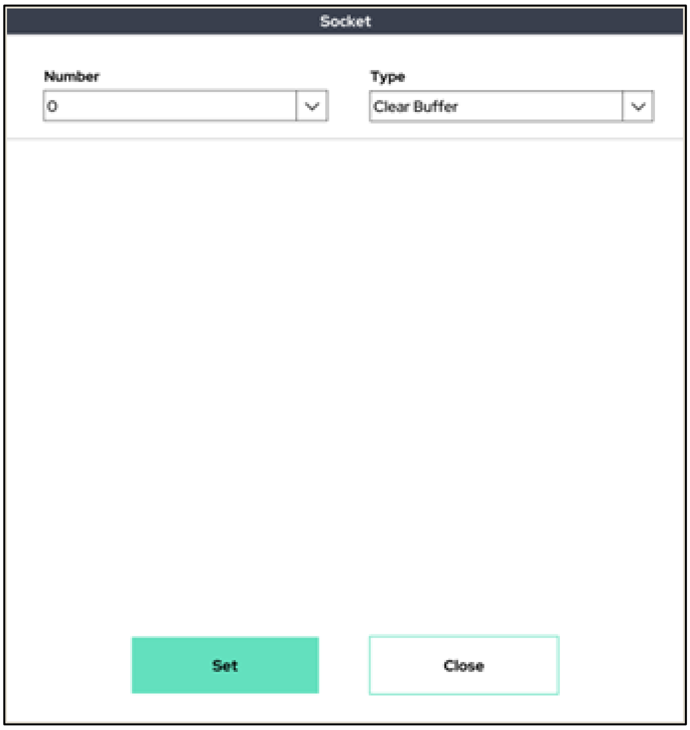
This is the function to put the ASCII string received through Socket communication into the selected string variable.

- Socket Function: Send String



Allows the user to send a specific string to the server. Users can enter a string directly in the field or send a predefined string type variable.

- Socket Function: Clear Buffer



Empty all the contents of the buffer for that number.

WARNING

Warning:

The syntax that needs to be followed:

In order to use the Read ASCII Variable, ASCII Array, and String options provided by the robot manufacturer, the data format received from the server MUST follow the following format. If a special communication grammar/syntax is required, please consult with the manufacturer.

Read ASCII Variable

When receiving a value from the server, the value must be sent as a numerical value. (i.e. the numerical value hasn’t to be contained within quotation marks)

(e.g. 123, 4567)

Read ASCII Array

When the server wants to send an array, it must send the numbers enclosed in curly braces {}. To separate the numbers, a comma (,) must be placed between each number. (i.e. the array must not be sent as a string inside quotes).

(e.g. {100,200,300})

Read String

When the server sends a string, it must be in the format "this_is_string_from_server", including the quotation marks. To check the connection status of socket communication or whether a reply has been received, two internal variables are available as shown below. Both internal variables can be optionally used in the Shared Data type.

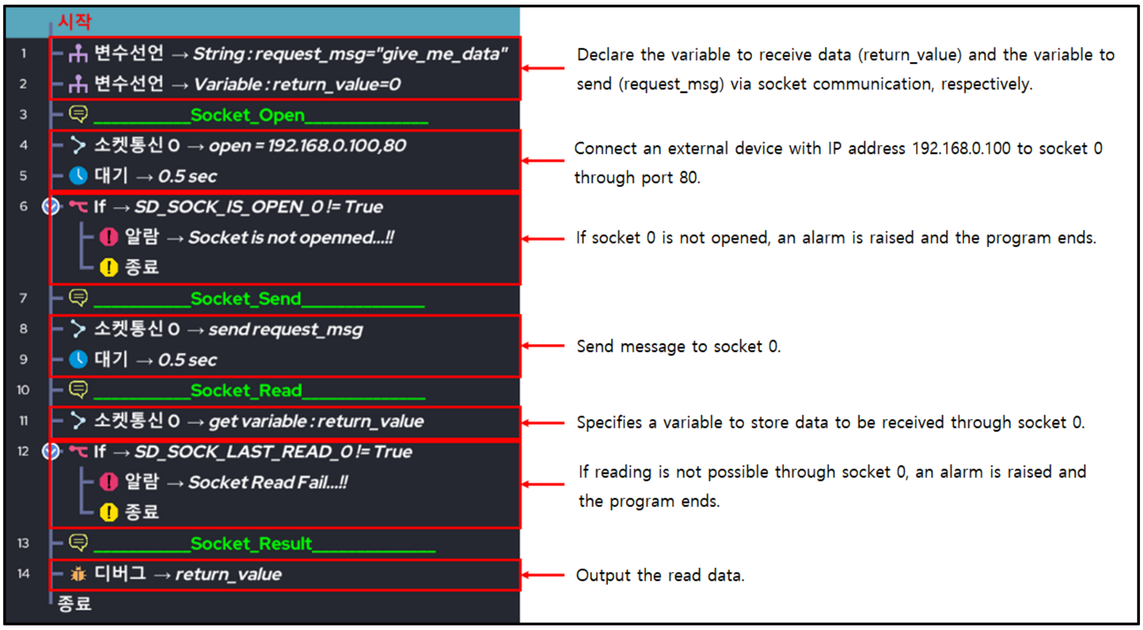
- SD_SOCK_IS_OPEN_# (# represents socket number 0~4)

This variable store whether the socket with the corresponding number is successfully opened and connected to the server. After Socket-Open, you can check if the socket is properly connected using a condition like If(SD_SOCK_IS_OPEN_0).

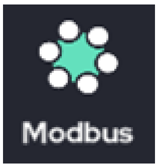
- SD_SOCK_LAST_READ_# (# represents socket number 0~4)

This variable store whether the Read function has been successfully executed for the corresponding socket number. For example, after Socket-ReadAsciiVariable, you can check if the last Read function was successful with a condition like If(SD_SOCK_LAST_READ_0). If no data is received from the server, this variable will have a value of 0.

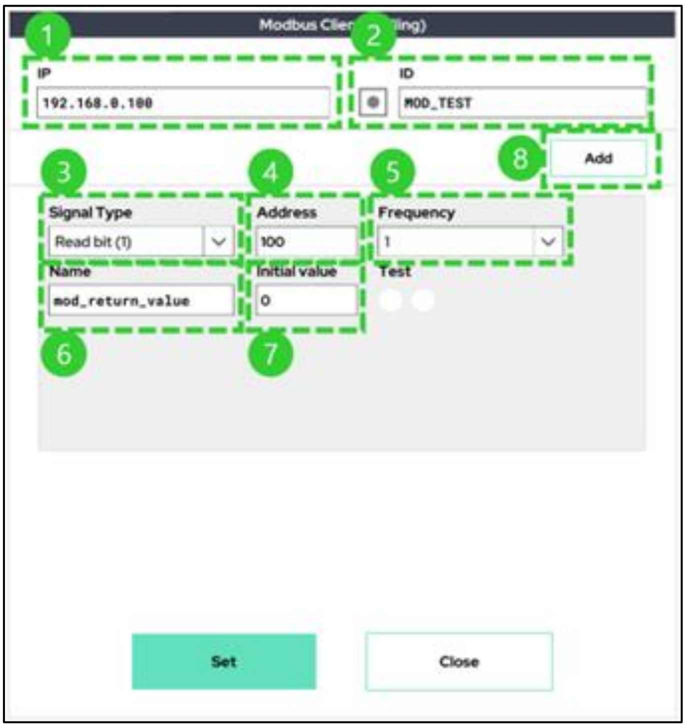
The figure below shows an example of the Socket Function.



■ Modbus TCP (Client) Function:



Provides the ability to request and receive data from a specific IP / address. Data request frequency and format can be specified. The port number for Modbus TCP is fixed at 502 (Modbus standard). The protocols and formats associated with Modbus TCP servers are listed in the Appendix I. The Modbus TCP client function must be added at the top of the program under Pre.P.



① Field for entering the IP address of the server to connect as a Modbus client

② Field for entering an ID for identification

③ Field to select the signal type: Choose one from Read bit (1-bit), Read word (16-bit), Write bit (1-bit), or Write word (16-bit).

④ Field for entering the address value to access on the server

⑤ Field to select the cycle frequency for Read/Write operations (Unit: Hz)

⑥ Field to declare the variable name where the read value will be stored (for Read) or the variable name for output (for Write)

⑦ Initial value of the variable set in step 5

⑧ Button to add the signal to be used

Below is an example of the Modbus Function settings.

Example 1)

The ability to read word type information in address 100 of the server (IP: 192.168.0.100) and put it in a variable called mod_return_value. The reading cycle of information is 1 Hz.

Example 2)

The ability to write the value stored in mod_write_bit to the address 200 of the server (IP: 192.168.0.100). Send the value of mod_write_bit to the server at a cycle of 100 Hz.

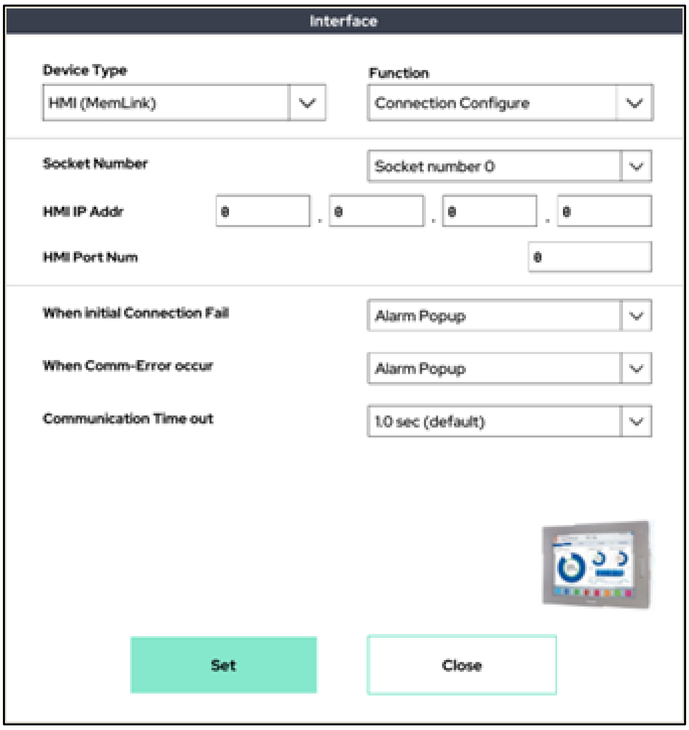
■ Interface Function:

The interface function is for connecting external devices such as PLC, HMI, and PC with the control box. The list of external devices that can be used using the interface is as follows.

- HMI (MemLink) - Proface, TOP
- PLC (MC Protocol) – Mitsubishi PLC
- Music Player
- PLC (XGT Protocol) - LS Electric PLC
- CSV File
- Pickit
- Modbus Client (Interrupt)
- OMRON PLC (FINS Protocol)
- Siemens PLC (S7 Protocol)
- OnRobot Eye

Because each external device has different detailed features available, you should refer to the following information.

- HMI(MemLink)- Connection Configure



A function that connects communication between the HMI and the RB system. User will enter the socket number, IP address, and port. User can also decide whether to turn on or ignore alarm pop-up in the event of a connection failure or communication error and set a communication timeout time.

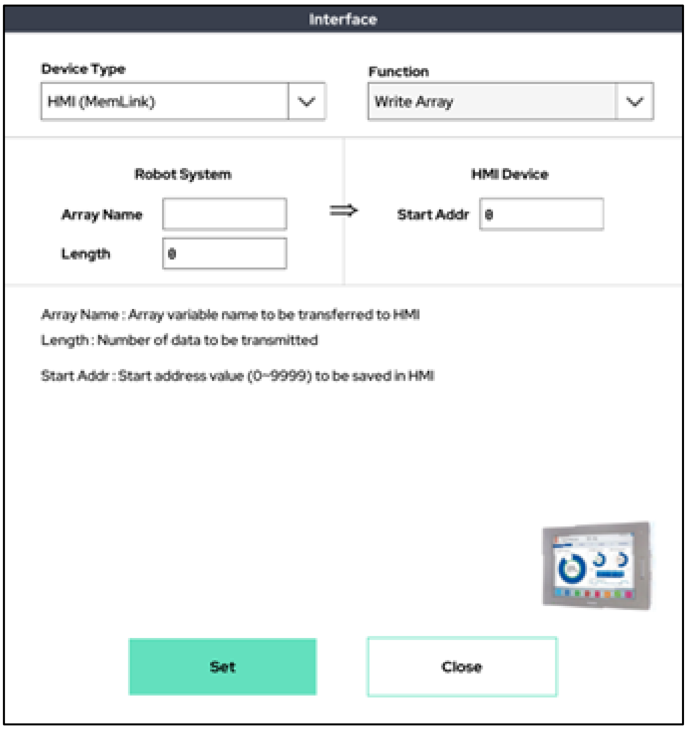
- HMI(MemLink)- Write Single variable

The ability to enter values for one address of HMI. Enter a number or variable name for the transfer value.

- HMI(MemLink)- Read Single variable

The ability to read values from one address in HMI. The read values are stored in the variable you specify (Variable).

- HMI(MemLink)- Write Array

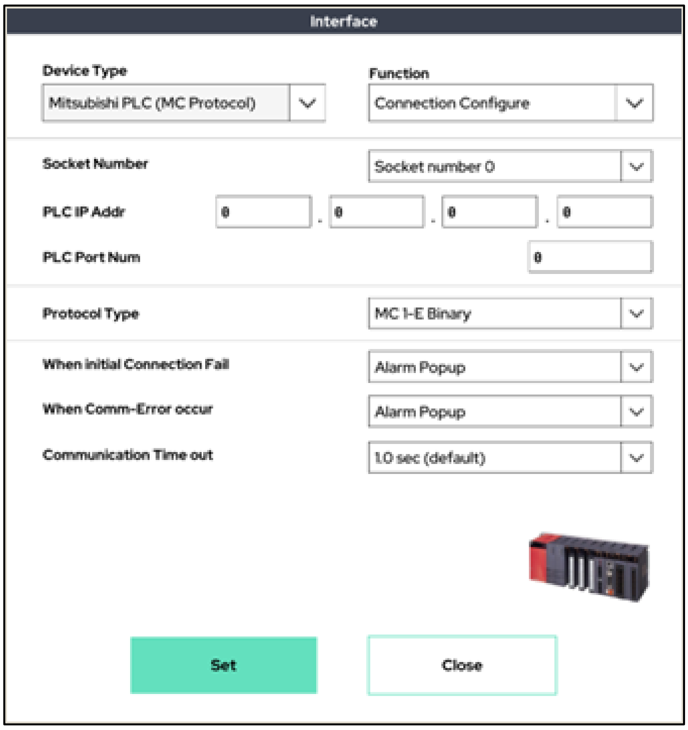


The ability to enter numbers from the starting address of the HMI to the specified number of addresses. The pre-declared array must be written to Array Name and should not exceed the maximum length of the array, 20.

- HMI(MemLink)- Read Array

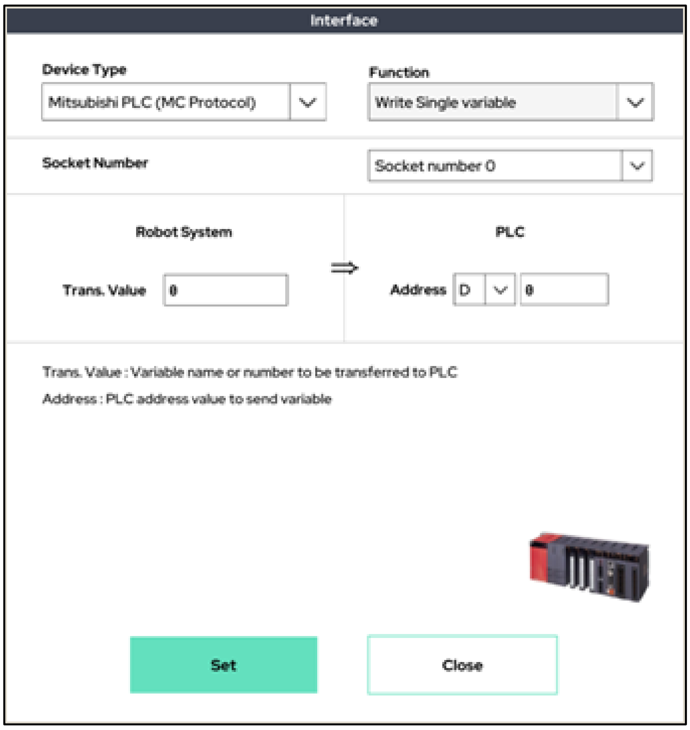
The ability to read data from the starting address of HMI to the specified number of addresses. The pre-declared array must be written to Array Name and should not exceed the maximum length of the array, 20.

- PLC(MC Protocol)- Connection Configure



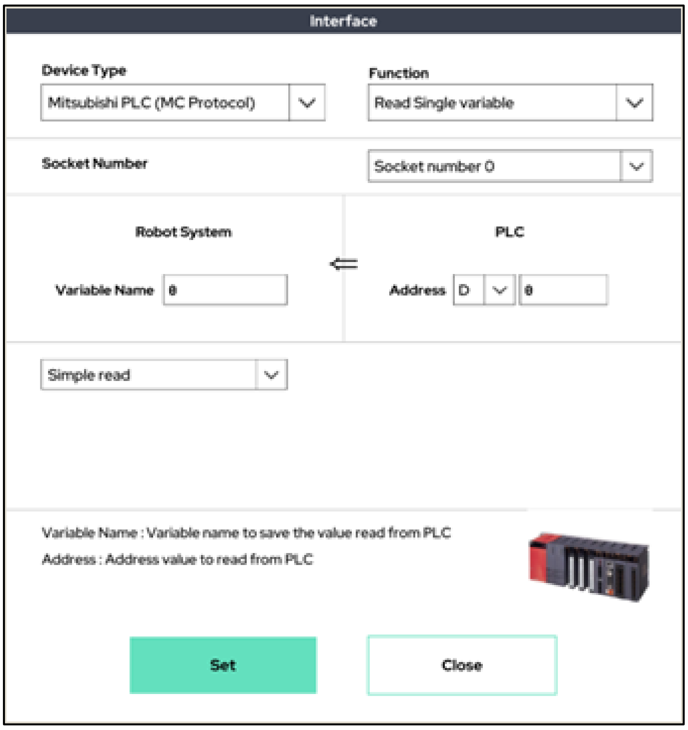
A function that connects communications between the Mitsubishi PLC and the RB system. User will enter the socket number, IP address, and port. User can also decide whether to turn on or ignore alarm pop-up in the event of a connection failure or communication error and set a communication timeout time.

- PLC(MC Protocol)- Write Single variable



The ability to enter values for one address of PLC. Enter a number or variable name for the transfer value.

- PLC(MC Protocol)- Read Single variable



The ability to read values from one address in PLC. The read values are stored in the variable you specify (Variable).

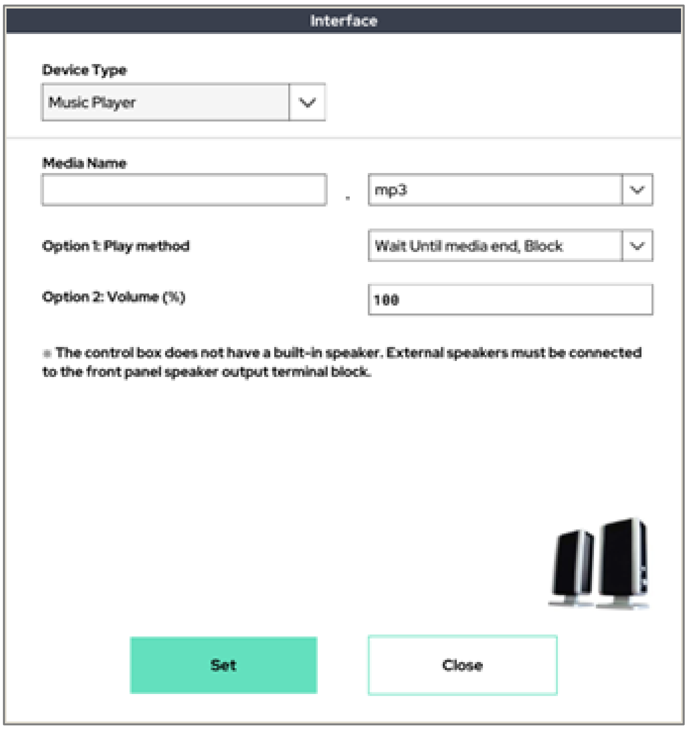
- PLC(MC Protocol)- Write Array

The ability to enter numbers from the starting address of the PLC to the specified number of addresses. The pre-declared array must be written to Array Name and should not exceed the maximum length of the array, 20.

- PLC(MC Protocol)- Read Array

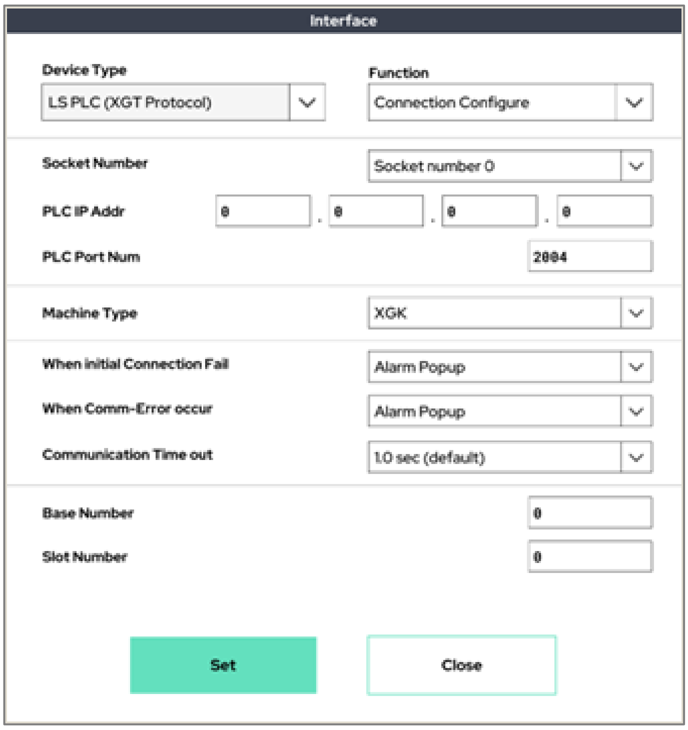
The ability to read data from the starting address of PLC to the specified number of addresses. The pre-declared array must be written to Array Name and should not exceed the maximum length of the array, 20.

- Music Player



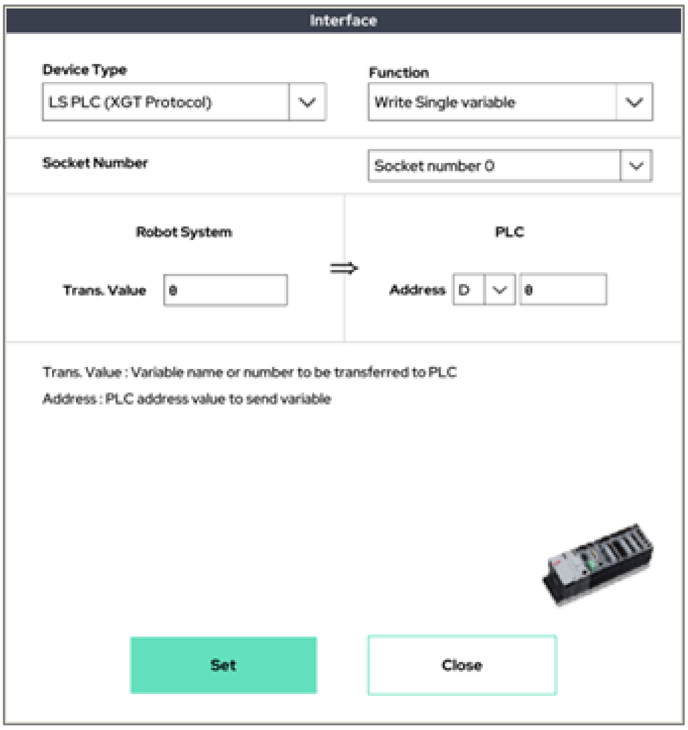
This function plays an mp3 file while the program is running. The Music driver must be installed through the RB Driver, and the mp3 file you want to play must exist in the specified path.

- PLC(XGT Protocol)- Connection Configure



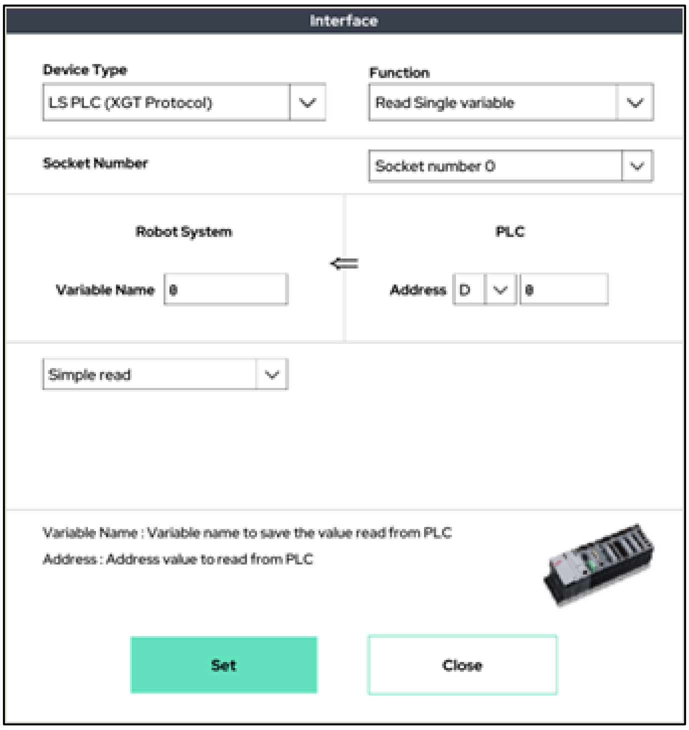
A function that connects communications between the LS Electric PLC and the RB system. User will enter the socket number, IP address, and port. User can also decide whether to turn on or ignore alarm pop-up in the event of a connection failure or communication error and set a communication timeout time.

- PLC(XGT Protocol)- Write Single variable



The ability to enter values for one address of PLC. Enter a number or variable name for the transfer value.

- PLC(XGT Protocol)- Read Single variable

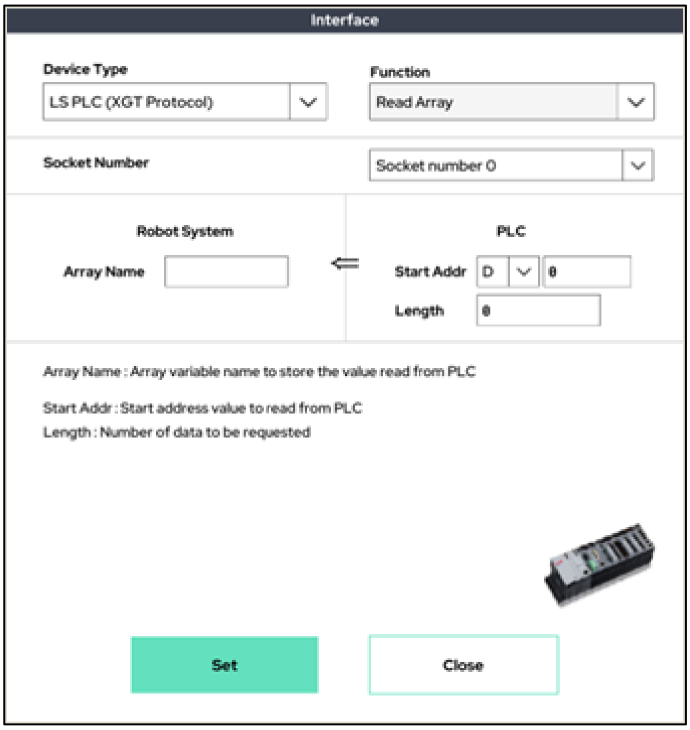


The ability to read values from one address in PLC. The read values are stored in the variable you specify (Variable).

- PLC(XGT Protocol)- Write Array

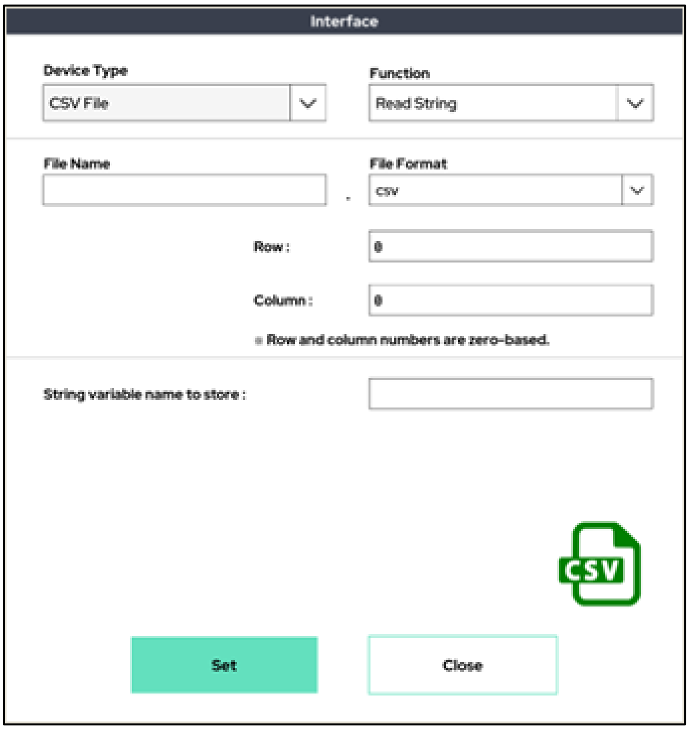
The ability to enter numbers from the starting address of the PLC to the specified number of addresses. The pre-declared array must be written to Array Name and should not exceed the maximum length of the array, 20.

- PLC(XGT Protocol)- Read Array



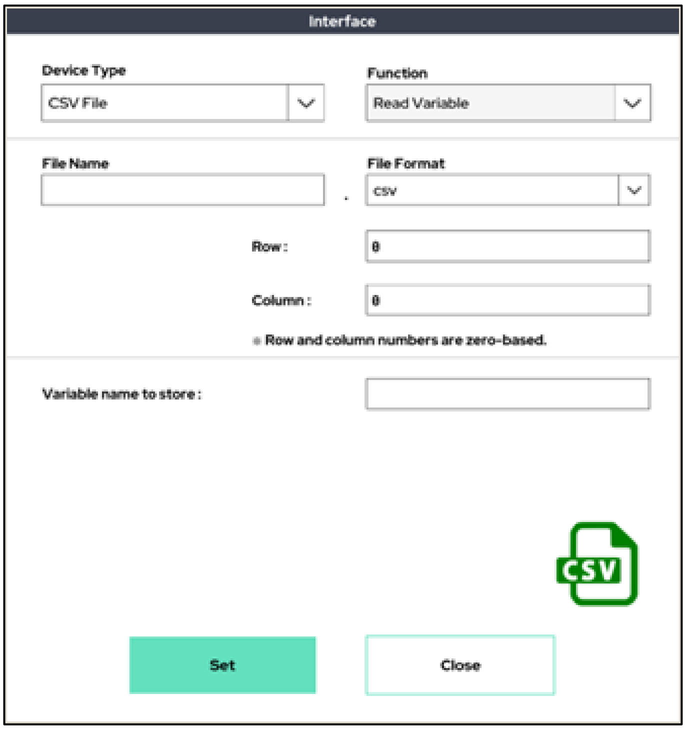
The ability to read data from the starting address of PLC to the specified number of addresses. The pre-declared array must be written to Array Name and should not exceed the maximum length of the array, 20.

- CSV File-Read String



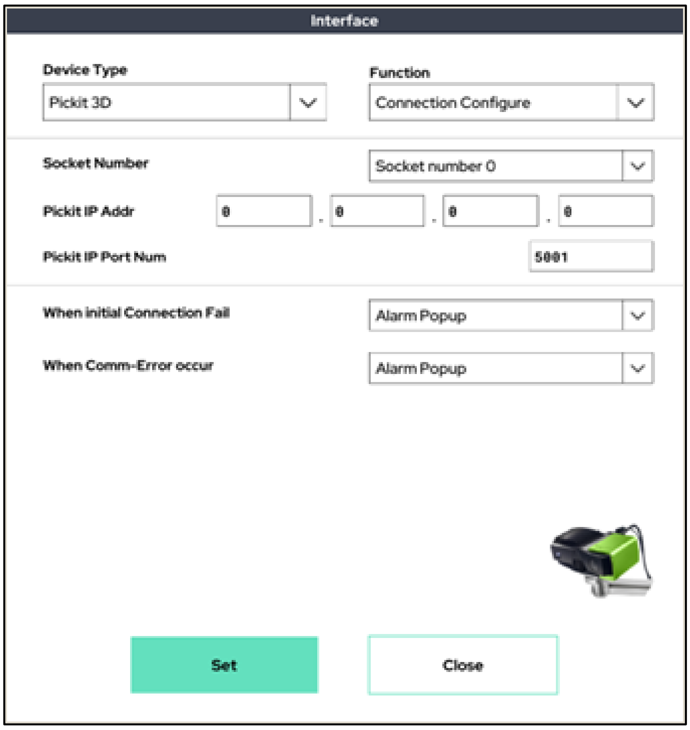
This function reads a string from a CSV file. The CSV file must be saved within the specified path.

- CSV File-Read Variable



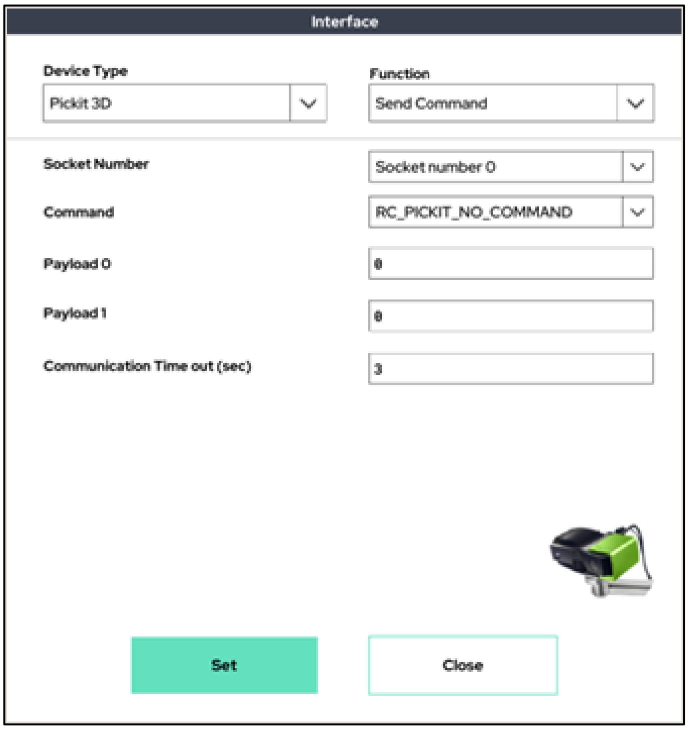
This function reads a single number from a CSV file. The CSV file must be saved within the specified path.

- Pickit-Connection Configure



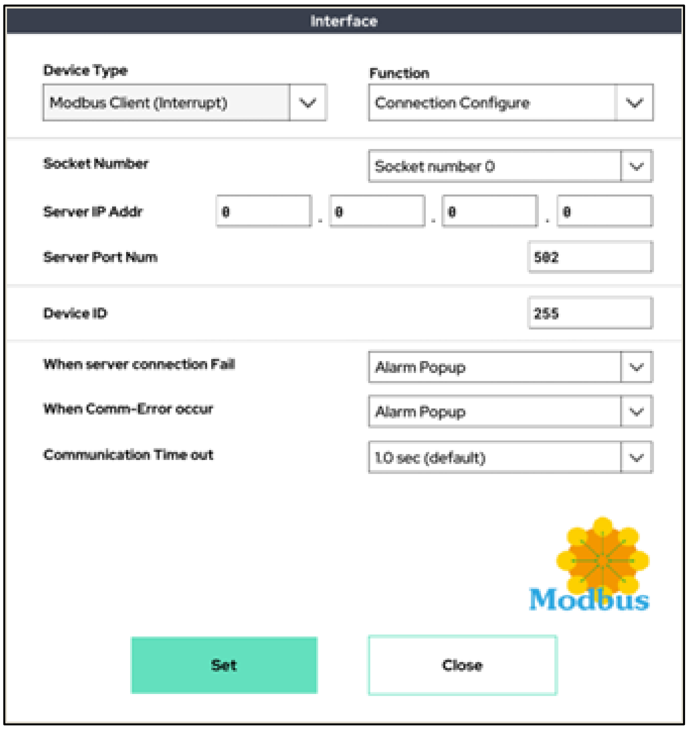
A function that connects communications between the Pickit and the RB system. User will enter the socket number, IP address, and port. User can also decide whether to turn on or ignore alarm pop-up in the event of a connection failure or communication error.

- Pickit -Send Command



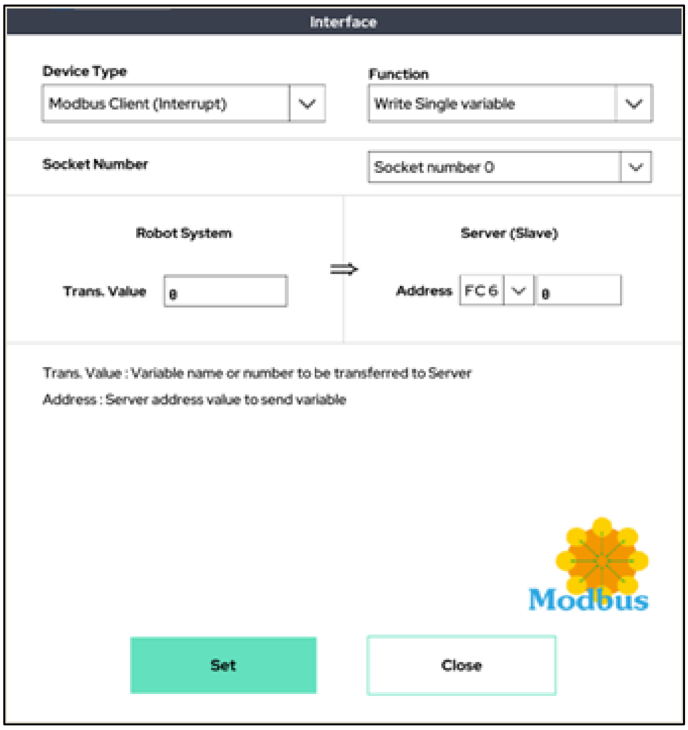
Set the command to be sent to Pickit and the data according to the command.

- Modbus Client(Interrupt) -Connection Configure



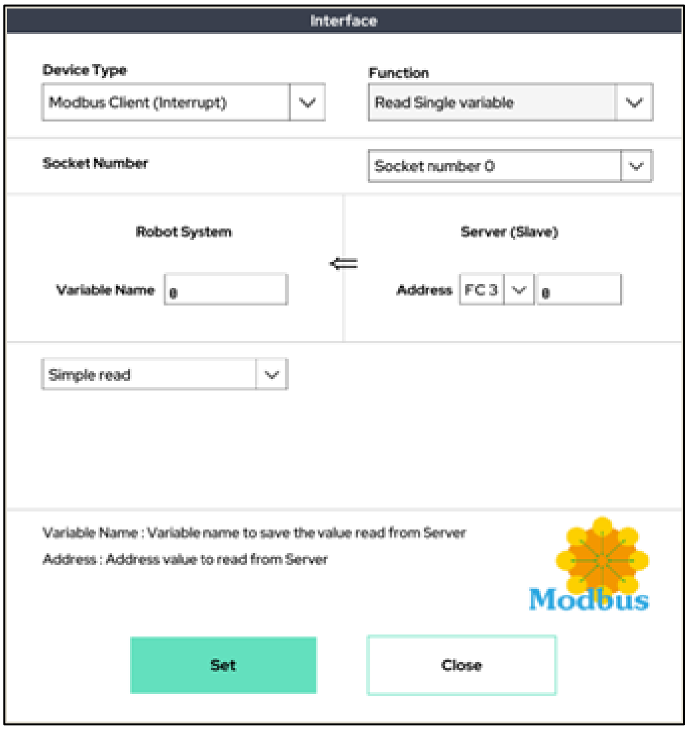
This is a function that connects the RB system as a client in Modbus communication. User will enter the socket number, IP address, and port. User can also decide whether to turn on or ignore alarm pop-up in the event of a connection failure or communication error and set a communication timeout time.

- Modbus Client(Interrupt) -Write Single variable



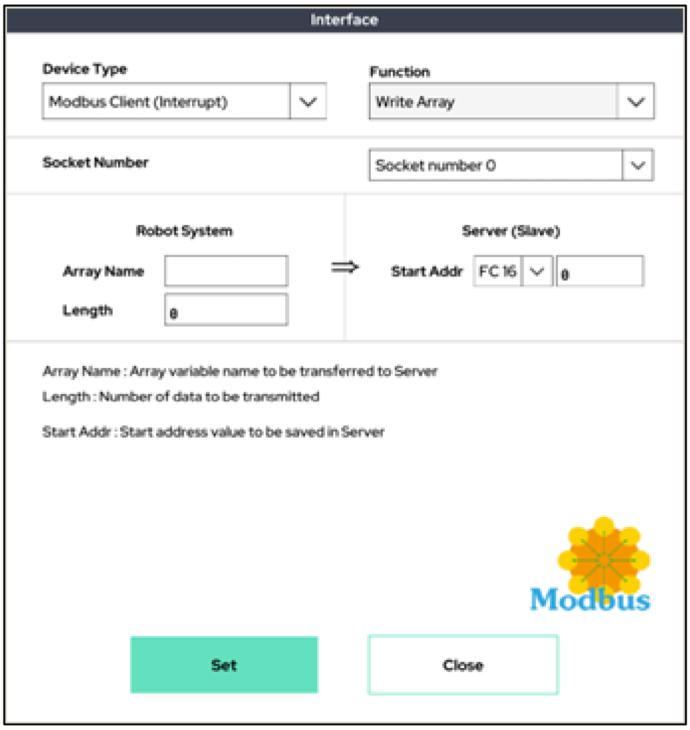
This is a function to input word type data to one address through Modbus communication. At this time, enter the name of a number or variable for the transfer value.

- Modbus Client(Interrupt) -Read Single variable



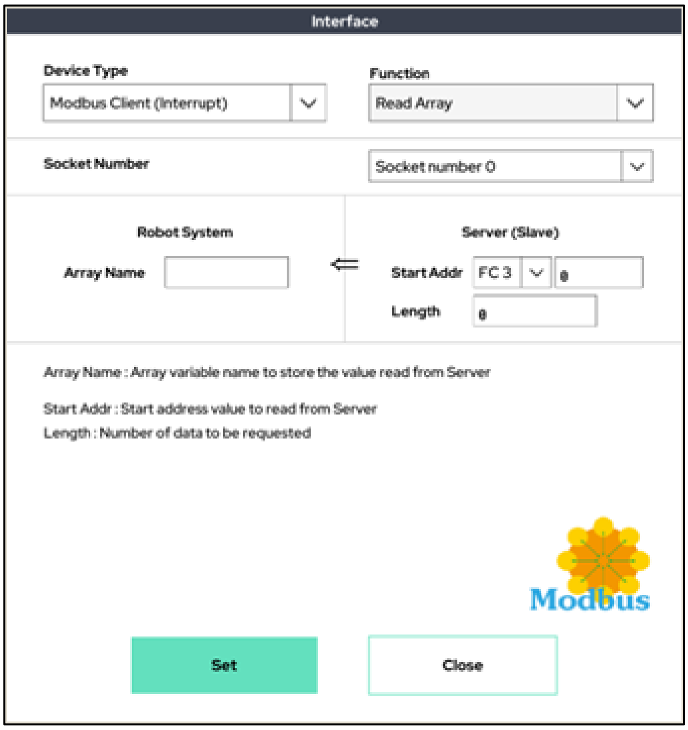
This is a function to read the value of one word type data from the address through Modbus communication. At this time, the read value is saved in the assigned variable.

- Modbus Client(Interrupt) -Write Array



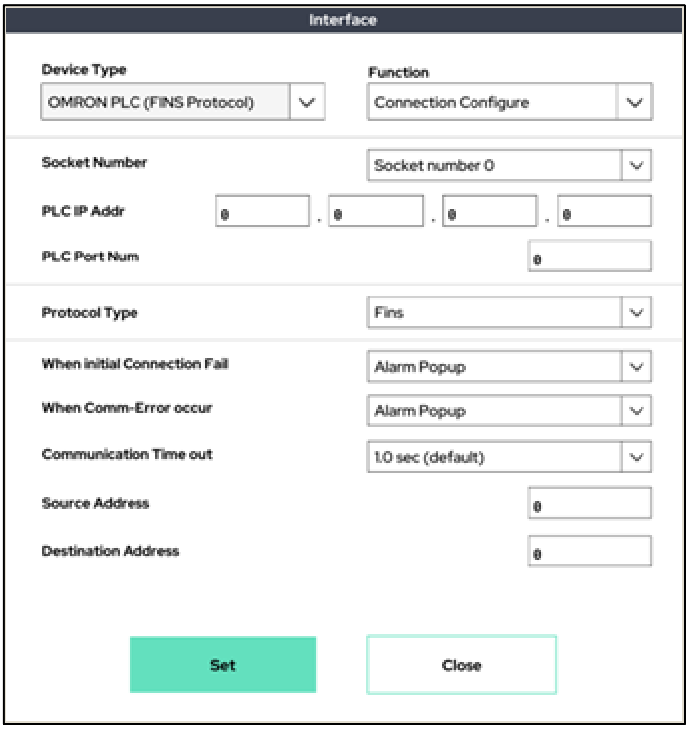
This is a function to input word data from the start address to the specified number of addresses through Modbus communication. At this time, the previously declared array should be written in 'Array Name' and the length should not exceed 20, the maximum length of the array.

- Modbus Client(Interrupt) -Read Array



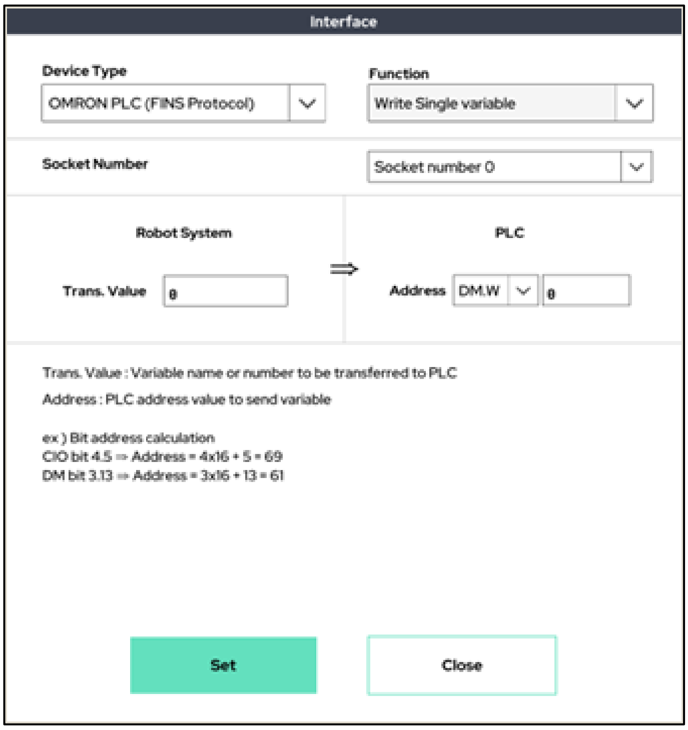
This function reads data from the start address to the specified number of addresses through Modbus communication. At this time, the previously declared array should be written in 'Array Name' and the length should not exceed 20, the maximum length of the array.

- OMRON PLC(FINS Protocol) – Connection Configure



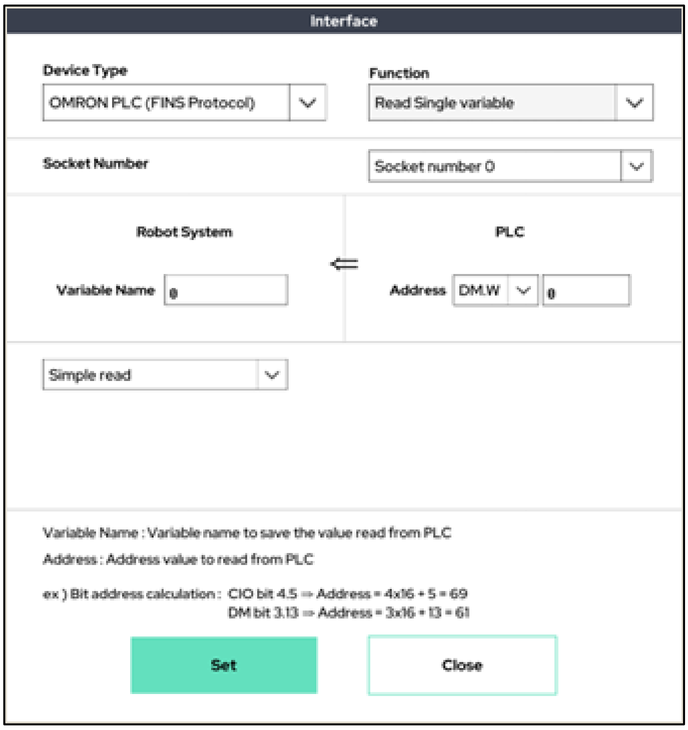
A function that connects communications between the OMRON PLC and the RB system. User will enter the socket number, IP address, and port. User can also decide whether to turn on or ignore alarm pop-up in the event of a connection failure or communication error and set a communication timeout time.

- OMRON PLC(FINS Protocol) – Write Single variable



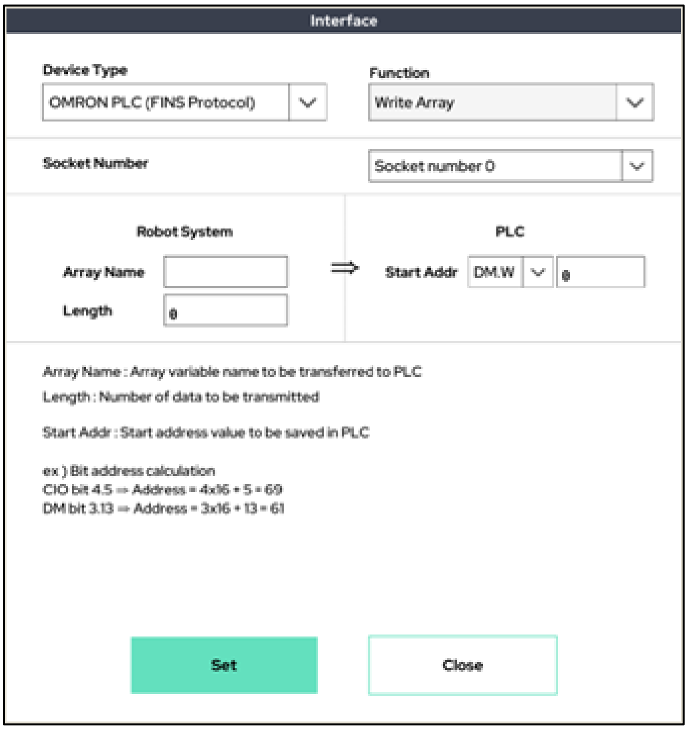
The ability to enter values for one address of PLC. Enter a number or variable name for the transfer value.

- OMRON PLC(FINS Protocol) – Read Single variable



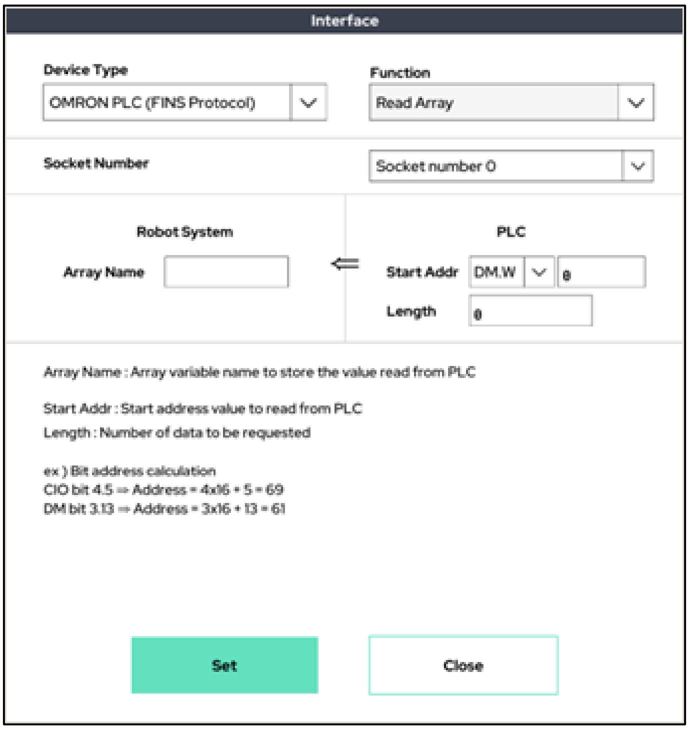
The ability to read values from one address in PLC. The read values are stored in the variable you specify (Variable).

- OMRON PLC(FINS Protocol) – Write Array



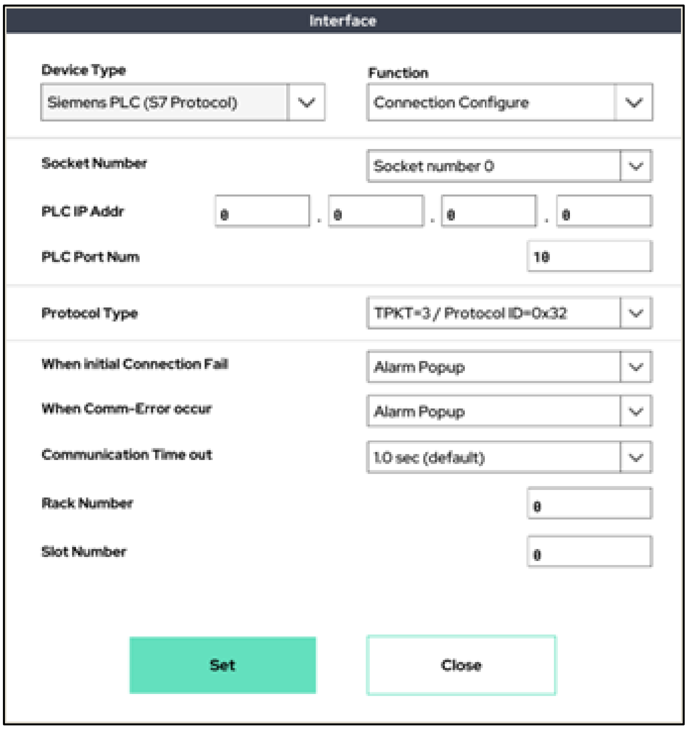
The ability to enter numbers from the starting address of the PLC to the specified number of addresses. The pre-declared array must be written to Array Name and should not exceed the maximum length of the array, 20

- OMRON PLC(FINS Protocol) – Read Array



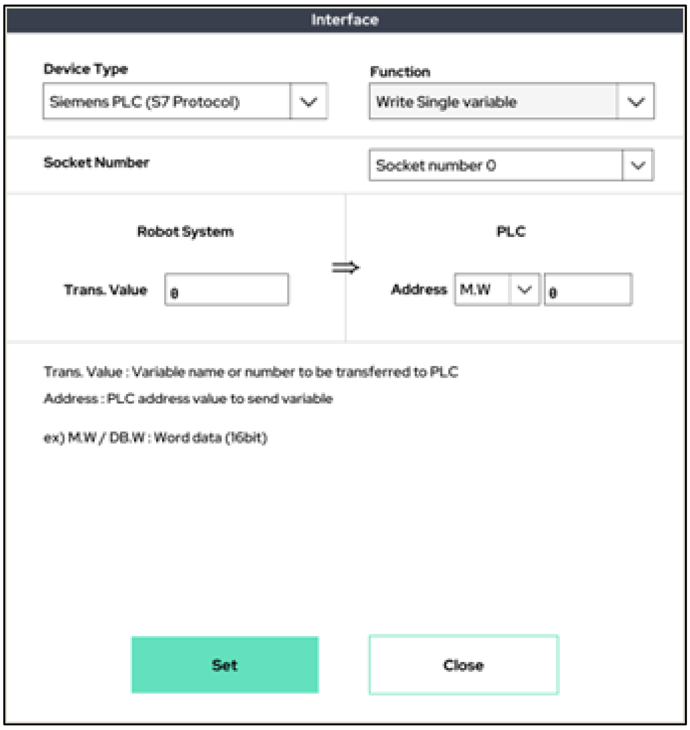
The ability to read data from the starting address of PLC to the specified number of addresses. The pre-declared array must be written to Array Name and should not exceed the maximum length of the array, 20

- Siemens PLC(S7 Protocol) – Connection Configure



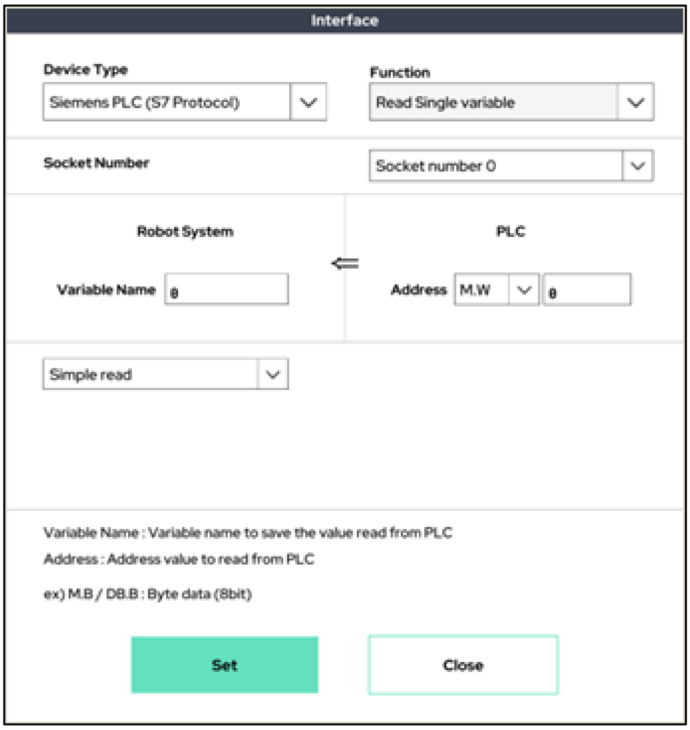
A function that connects communications between the Siemens PLC and the RB system. User will enter the socket number, IP address, and port. User can also decide whether to turn on or ignore alarm pop-up in the event of a connection failure or communication error and set a communication timeout time.

- Siemens PLC(S7 Protocol) – Write Single variable



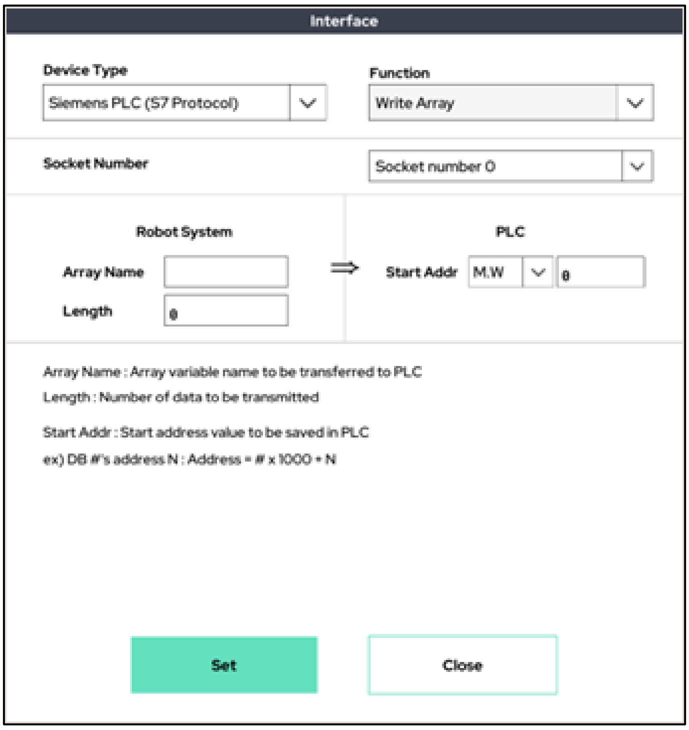
The ability to enter values for one address of PLC. Enter a number or variable name for the transfer value.

- Siemens PLC(S7 Protocol) – Read Single variable



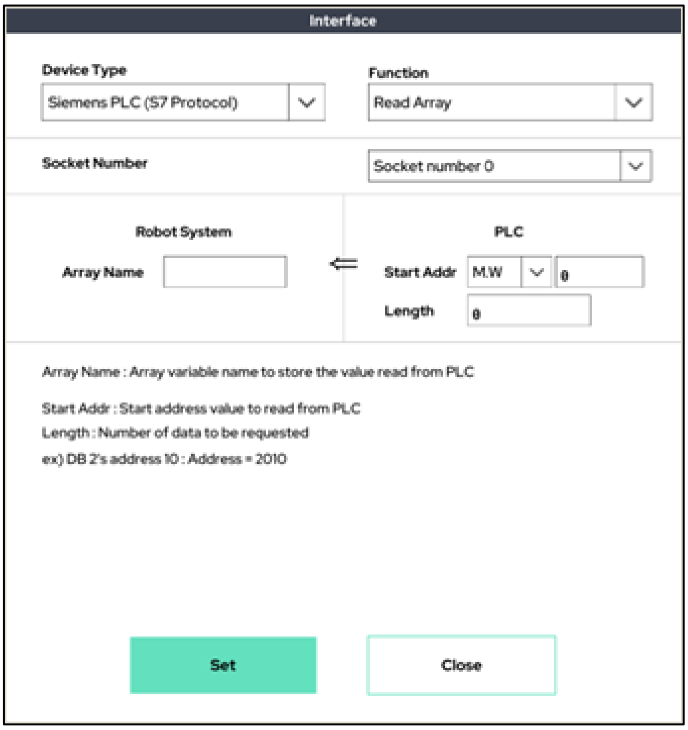
The ability to read values from one address in PLC. The read values are stored in the variable you specify (Variable).

- Siemens PLC(S7 Protocol) – Write Array



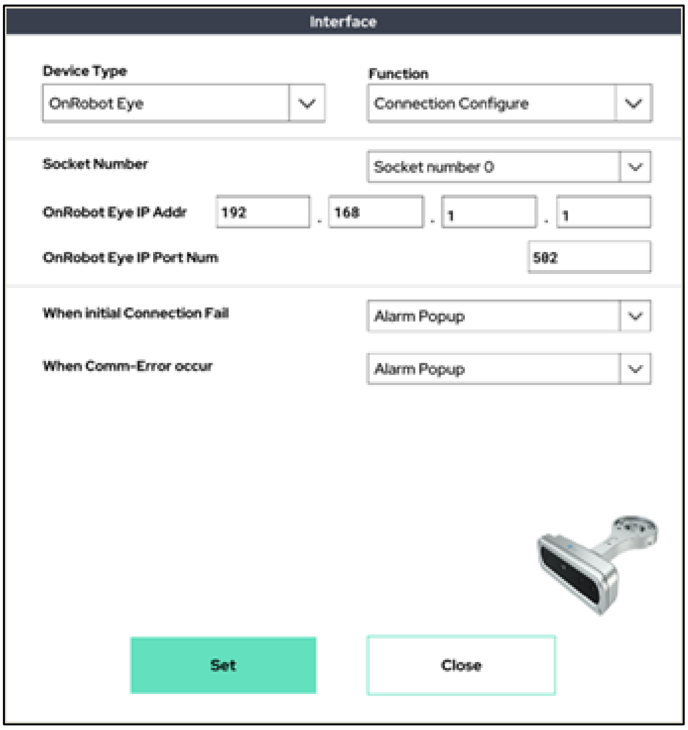
The ability to enter numbers from the starting address of the PLC to the specified number of addresses. The pre-declared array must be written to Array Name and should not exceed the maximum length of the array, 20

- Siemens PLC(S7 Protocol) – Read Array



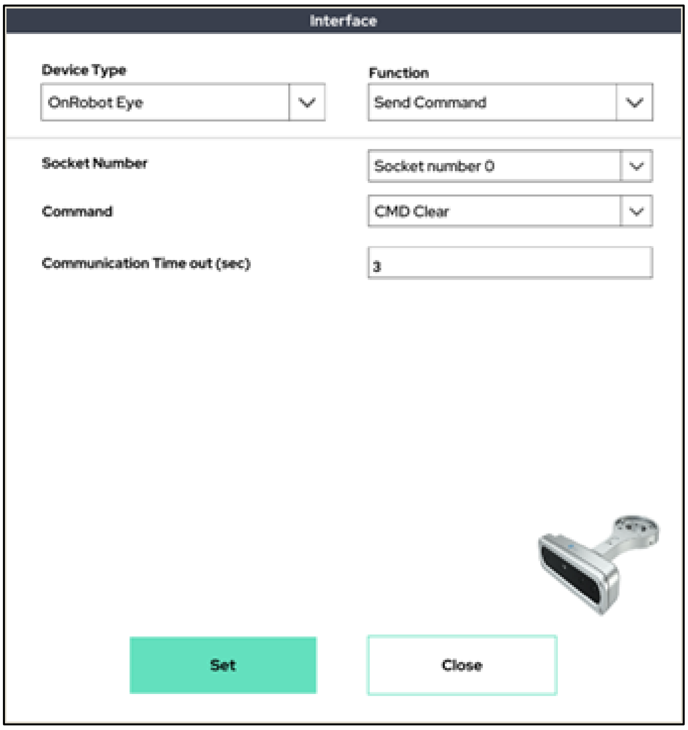
The ability to read data from the starting address of PLC to the specified number of addresses. The pre-declared array must be written to Array Name and should not exceed the maximum length of the array, 20

- OnRobot Eye – Connection Configure



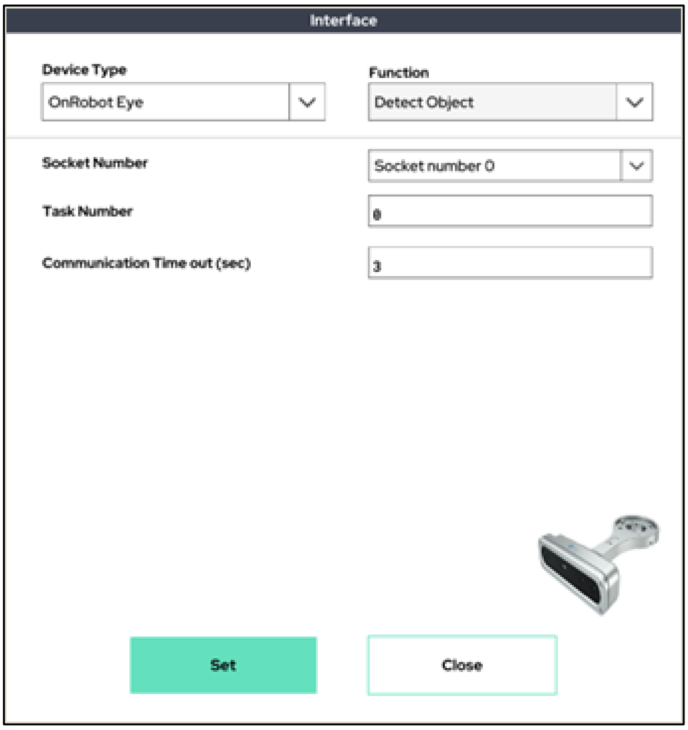
A function that connects communications between the OnRobot Eye and the RB system. User will enter the socket number, IP address, and port. User can also decide whether to turn on or ignore alarm pop-up in the event of a connection failure or communication error and set a communication timeout time.

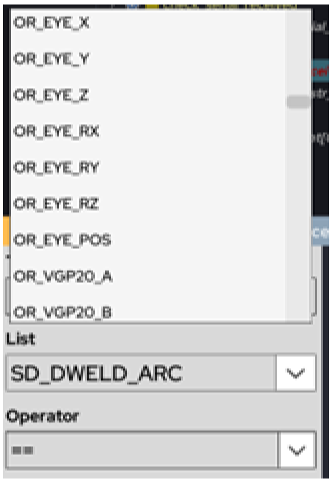
- OnRobot Eye – Send Command



You can send commands to the OnRobot Eye. When you send commands, you can set the timeout by communication timeout.

- OnRobot Eye – Detect Object

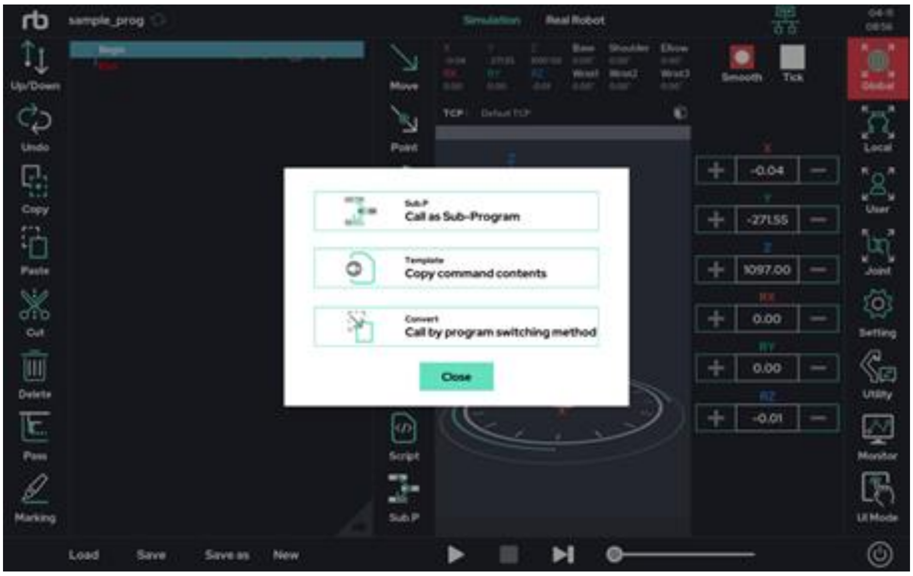




The OnRobot Eye's task number must be pre-set and is intended to execute the set task. For data obtained through the feature, you can select the properties on the keyboard as Devices and use them in the list.

■ Sub.P (Sub Program) Function:

This function allows you to insert a pre-made program file (teaching file) into the current document in an editable format. The "Call as Sub-Program" simply calls another program, while "Copy Command Content" copies the commands from another program and adds them to the main program. The "Call by program switching method" allows the sub-program to switch to the sub-program's window at the moment it is called and execute it.

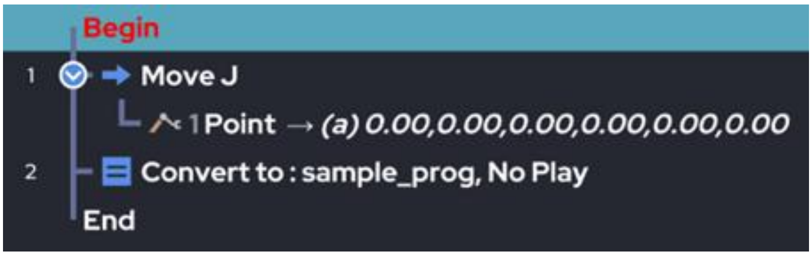


For example, you create a file named "sample_prog" as shown below.

Example 1: Calling sample_prog as a sub-program in a new program.

Example 2: Calling sample_prog using the "Copy Command Content" function in a new program.

Example 3: Calling sample_prog using the "Call as Program Switch" function in a new program.



In Example 1, when sample_prog is "**Called as a sub-program**", it will be executed, but it will not be editable in the main program. Additionally, if the called sub-program is modified, the execution of the main program will also be affected.

In Example 2, when using the "**Copy Command Content**" function, the sub-program is called in an editable format within the main program. Once copied, any changes made to the original sub-program will not affect the copied content in the main program.

In Example 3, using the "**Call as Program Switch**" function, you can specify the number of repetitions for the sub-program's execution, and the UI will switch to the sub-program and display the code being executed.

WARNING

Warning:

1) The contents of a subprogram called by the Sub.P function can be seen by the user, but they cannot be modified. If modifications are required, the project must be opened separately.
2) The Sub.P function can be called up to 10 levels deep. It is not recommended to use recursion with the Sub.P function.

■ Pattern Function:

This function allows the user to define repetitive behavior. By defining information about the operation space, and by defining which actions to be performed at each location, the user can set the robot to perform the same action at every point in space. The user can implement palletizing through this function. There are three sub settings.

- Pattern Property:

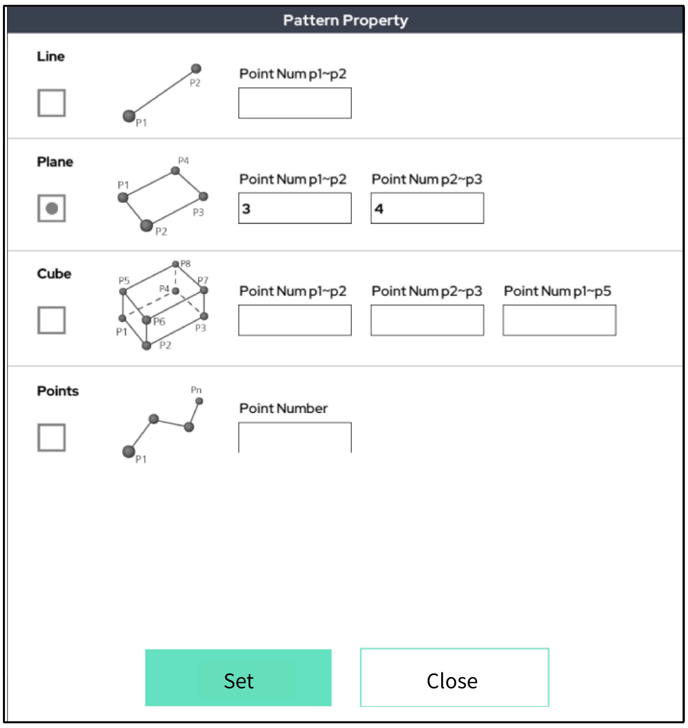
  Define the target space for the repetitive motion.

  The property supports various shapes such as straight line, plane, 3D cube, and arbitrary point.
- Pattern Anchor:

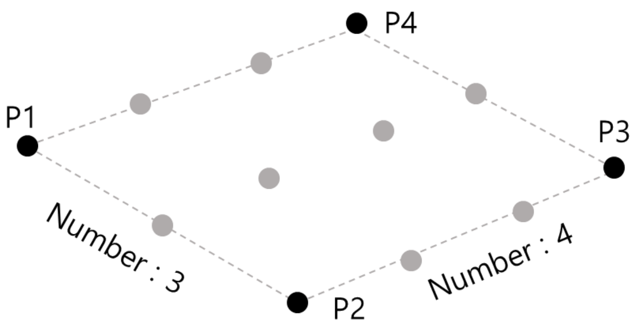
  The Reference point of the action defined in the Pattern Action.
- Pattern Action:

  This setting defines the motion relative to the reference point set in the Pattern Anchor. The defined relative behavior is repeated at every pattern point set in the Pattern Property.

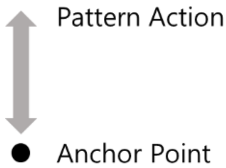
Step 1) Set the Pattern Property as shown below.



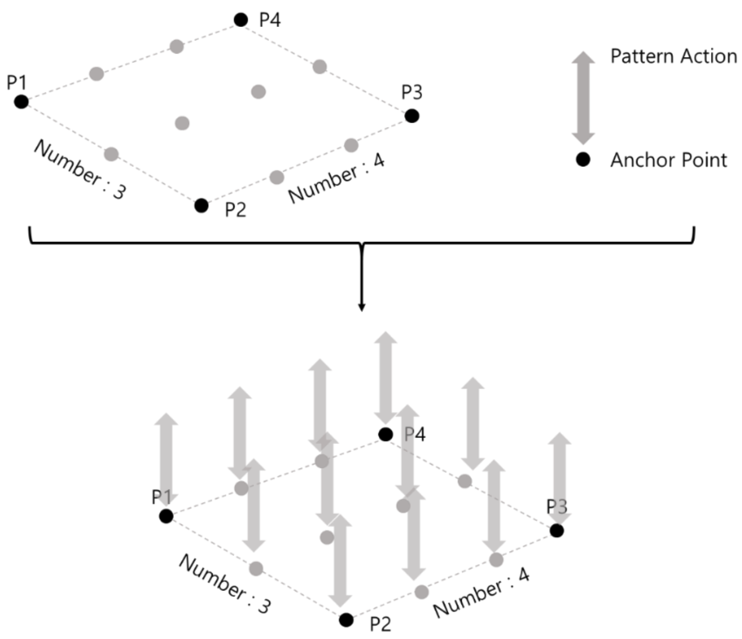
With the above settings, the following repeat points are formed in space.



Step 2) Using the Pattern Anchor and the Pattern Action, define the relative movement as below.



Step 3) Finally, the relative movement set in Step-2 is reflected in all of the pattern points set in Step-1, completing the action of repeating the same action.



■ Conveyor Function:

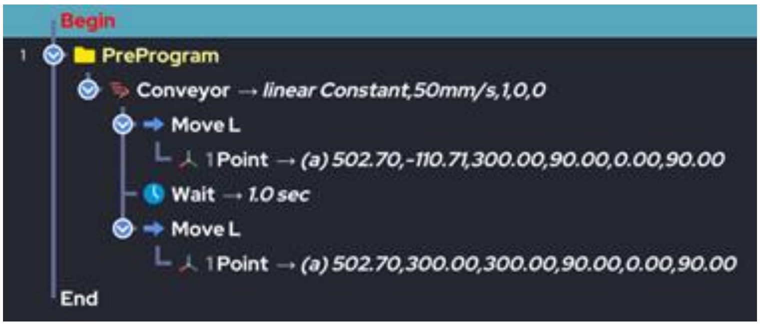
Allows the user to use the robot as a conveyor by generating movement at a consistent speed in a specified direction. The user can also place their own desired movement into the conveyor flow by using the Move L, Move PB, or Circle functions. Joint movement (Move J, Move JB, etc.) cannot be used as a subitem of Conveyor. Move L, Move PB(Formerly Move LB), Move JL, Move ITPL, Circle are supported.

Add the conveyor function to the program tree and click the function to see the options.

① Set the move type and speed of the conveyor.

② Set the direction for the conveyor movement (x, y, z value is based on robot arm base coordinate system).

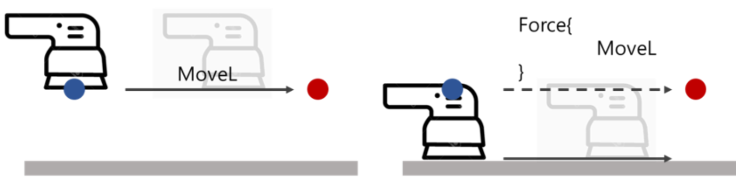
An example program tree using the Conveyor Function will look as follows:



■ Force Function:

This is a function for force control. The movements below the force control function automatically change the trajectory to give the set force.

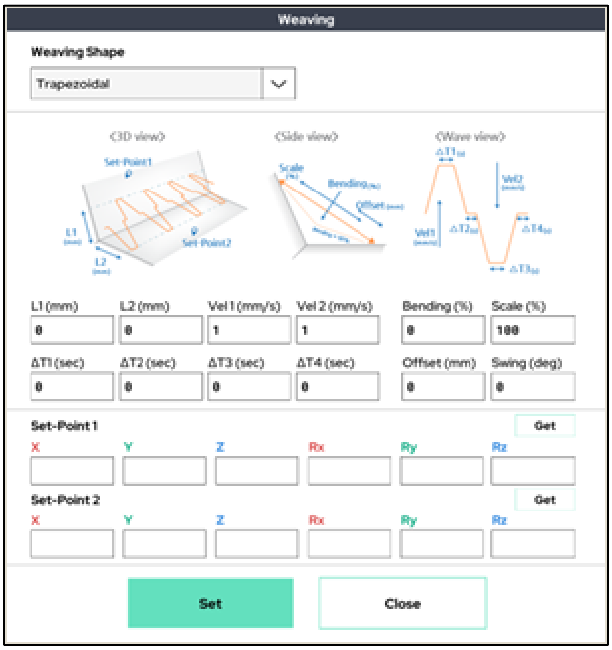
Select and input the desired force control mode, the sensor to be used for force control, and the force control target value. The left side of the figure below is for normal operation only. The motion starts in the air above the plane and ends in the air. If you put this action as a sub-item of force control as it is, it will change to the action of pressing the ground with a certain force (when setting the force control to the ground).



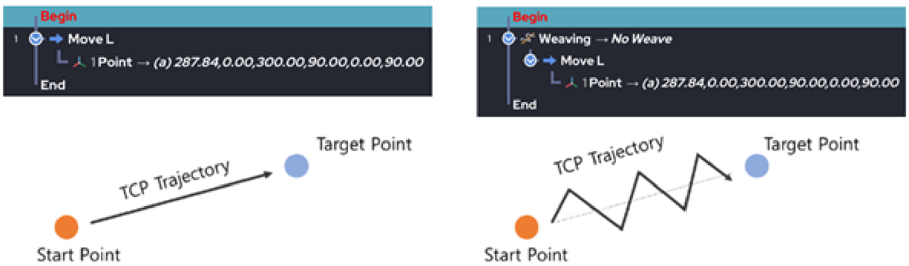
[Before applying force control] [After applying force control]

■ Weaving Function:

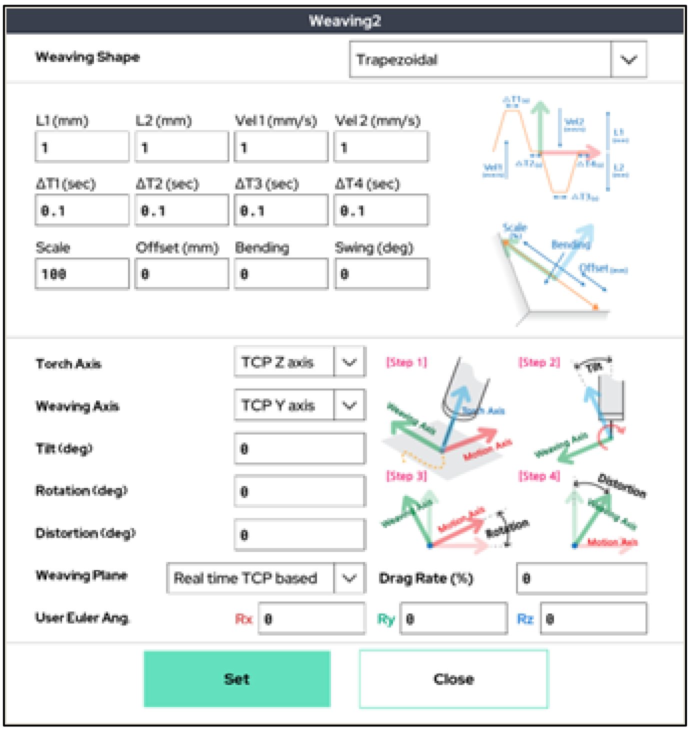
It is a special function for welding weaving. TCP trajectories are automatically changed to set the weaving actions included under the weaving function. Simply select and enter the desired weaving shape and weaving options.



The left side of the figure below is for normal operation only. If this motion is put as a sub-item of weaving, TCP trajectory reflecting the weaving trajectory is drawn (in the example on the right, in the case of triangle wave weaving).



■ Weaving2 Function:



The difference between weaving and weaving 2 is that in the case of weaving, the surface to be welded and the surface to be welded are set as set points, and weaving 2 implements weaving motion based on the frame of the TCP.

■ Touch Sensing (TouchSen.) Function:

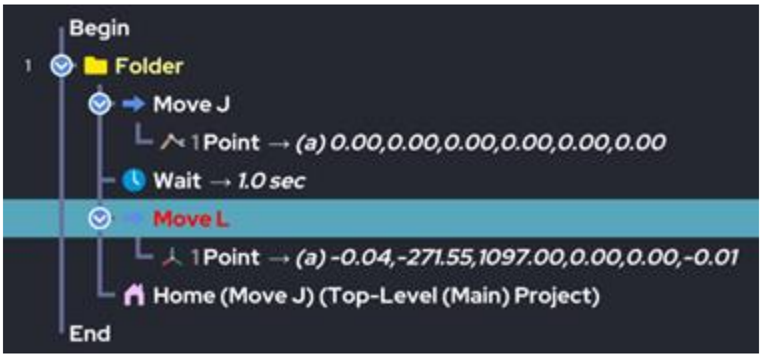
Touch sensing is intended to utilize welding applications. Detects the movement of the base material and reflects the direction of movement of the base material and is used for welding. A detailed description of this feature is provided in a separate manual.

7.5 EDITING THE PROGRAM

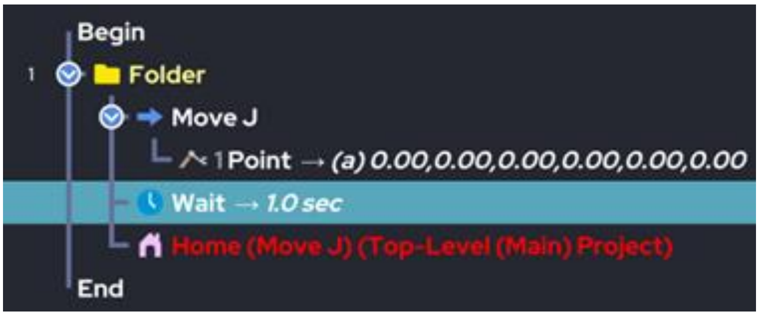
The bar on the left of the screen contains icons that allow a user to change the order or structure of the instructions entered in the program tree.
Please refer to section 6.1 for the description of the edit icon. The example explains how to edit the program.

■ Cut / Paste  

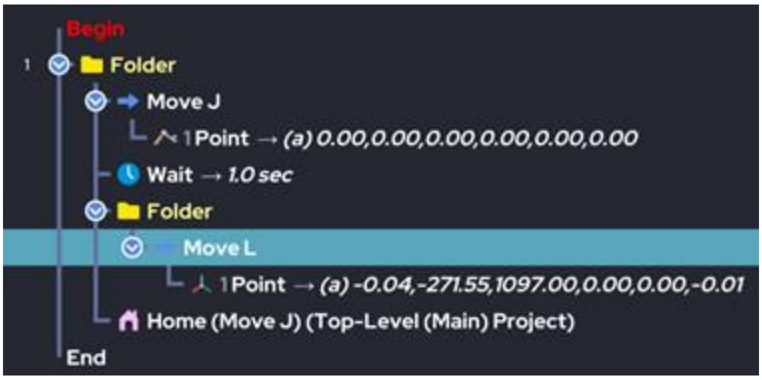
Step1) Select the command to cut. The selected command will be shown in blue. In the example below, the Move L line is selected.



Step2) Press the Cut button. Once Cut is clicked, the line disappears from the program tree.

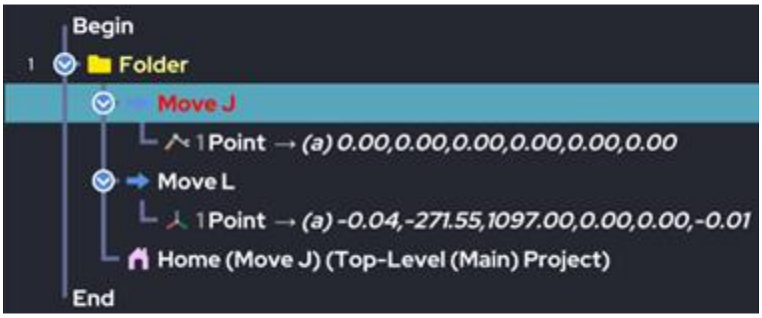


Step3) Click the location to paste and click the Paste button. In the example, the Move L command is pasted inside the Folder.



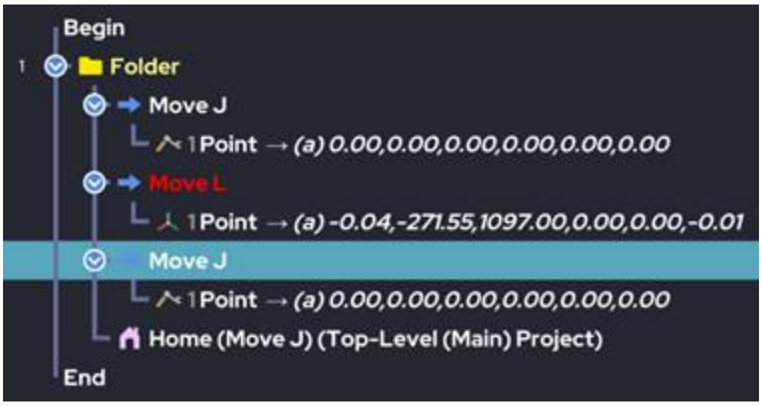
■ Copy / Paste 

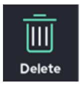
Step1) Select the item to copy. The selected command will be shown in blue. In the below example, the Move J line is selected.



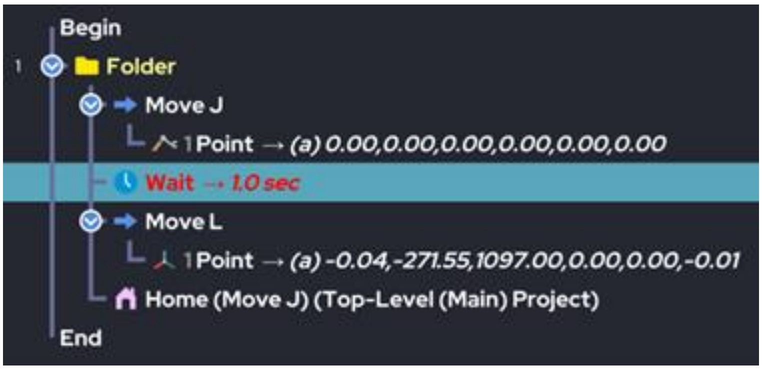
Step2) Press the Copy button.

Step3) Click desired location and click the Paste button. In the example, the Move J command is pasted under the Folder.

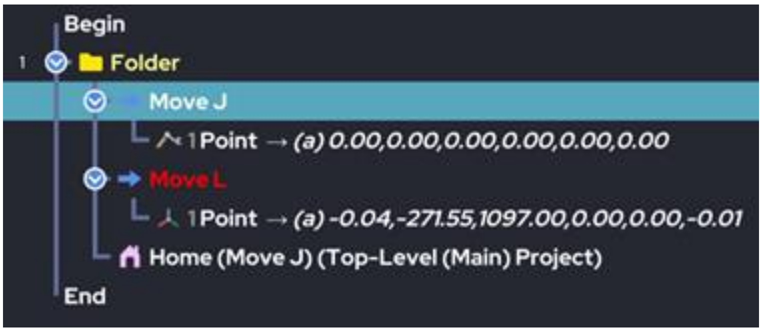


■ Delete 

Step1) Select the command to delete. The selected command will be shown in blue. In this example, the Wait command is selected.

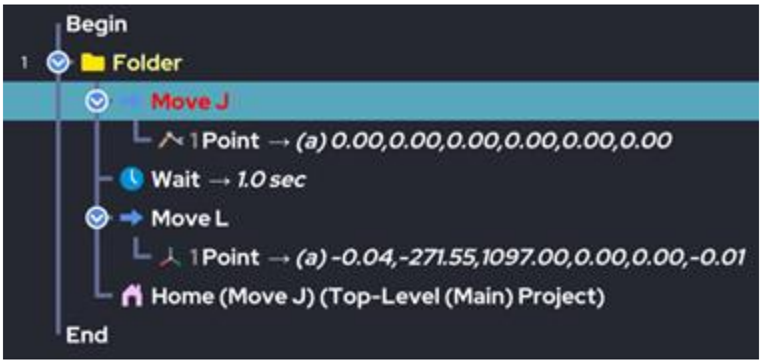


Step2) Click the Del button. The command has been removed as shown below.



■ Move 

Step1) Select the command to move. The selected command is shown in blue. In this example, Move J at the top is selected.

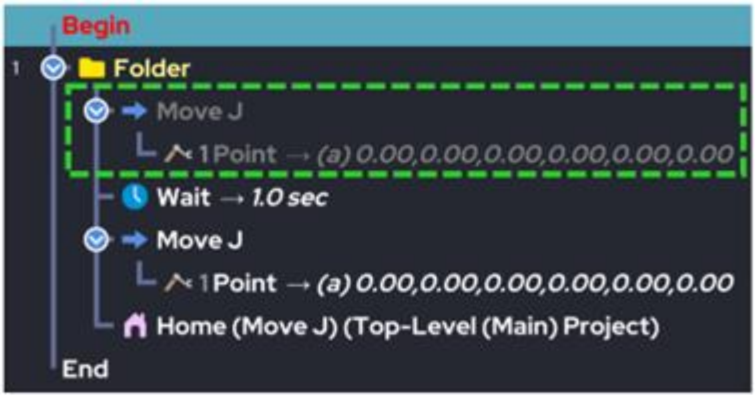


Step2) Click the Down button to move Move J down as shown below.

■ Pass 

Step1) Select a function to temporarily hold / block its execution. The selected command is shown in blue. In this example, the Move J command is selected.

Step2) Click the Pass icon. The command turns dark as shown below and will not execute. To undo, simply select the command again and press the Pass button again.



7.6 PROGRAM MANAGEMENT

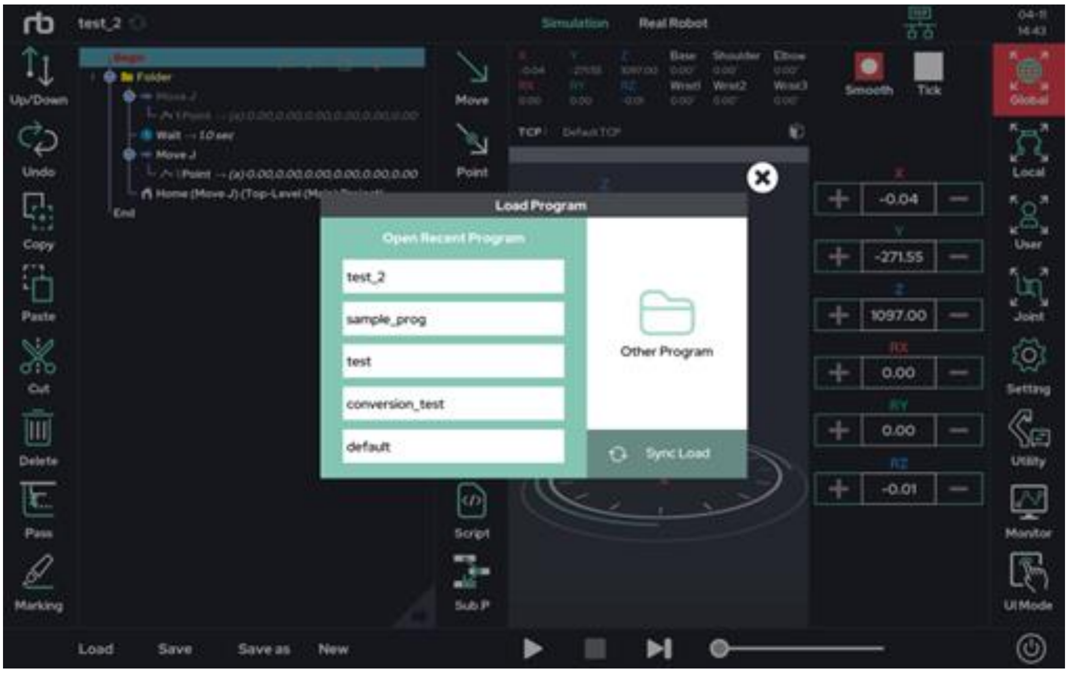
Allows the user to save, load, or create a project.

- Save Project

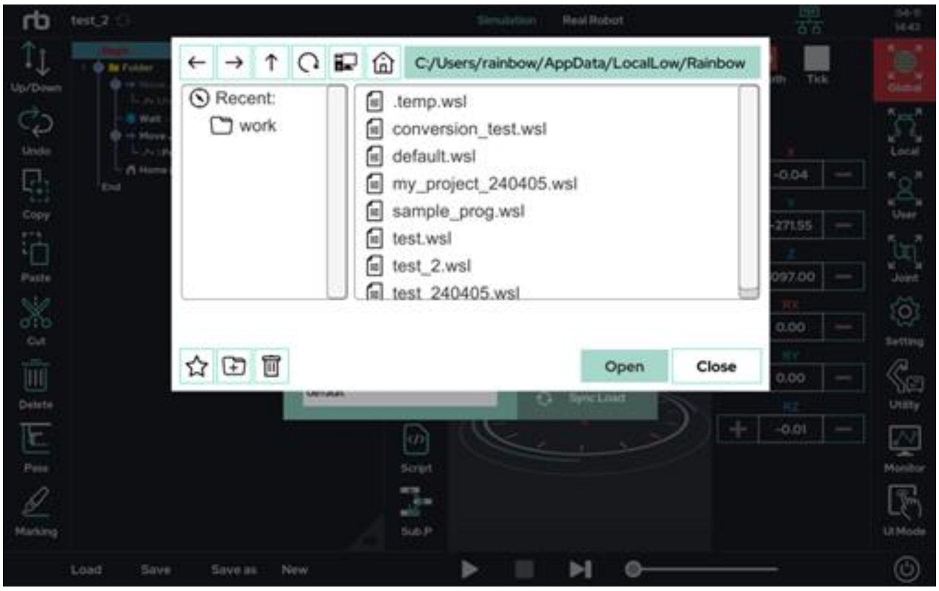
  To save the current project, click the save icon on the bottom left side of the UI work screen. there is no change from the existing saved contents, it is shown as below.

- Load File

To load a saved project, click the Load icon at the bottom of the UI to display a list (shown below). If a user selects a file from the list, it will be loaded in as the current program. If there are unsaved changes to the current project, a prompt will request the user to save. Only recently used files will appear in the list.

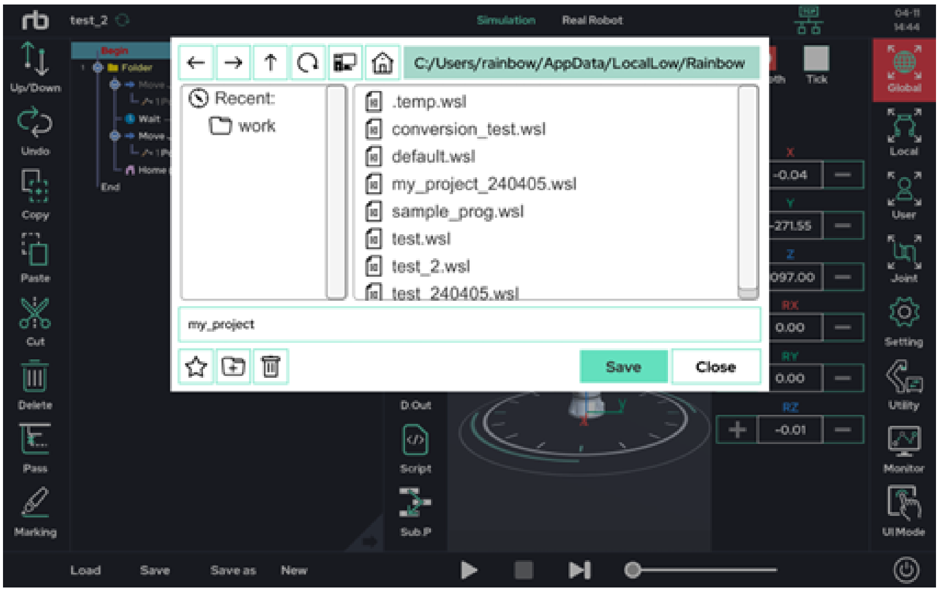


To open a file, users can click the Load option. Clicking Other Program button will open the File Explorer, which allows the user to look through saved files.



■ Save As

To save a program with a different name, click the Save As icon. The following popup window will appear. Using this window, users can save their current file with the desired program name. The program name cannot bet set to "default," as it is already in use by the system.



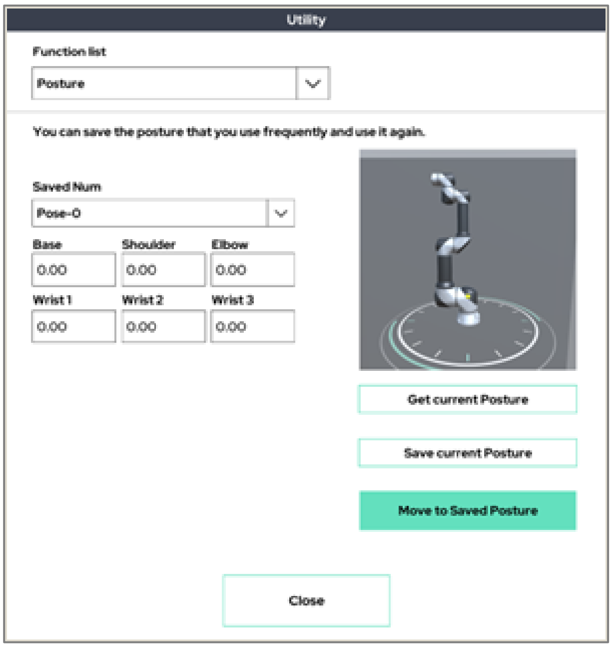
7.7 OPERATION UTILITIES

On the right side of the Make screen, there are other utility functions to help a user operate the system.

1) Setting: Allows the user to use the jog function, as well as other utility functions to help the user's experience.
2) Utility: A collection of additional functions, such as the posture saving function, the system input / output information view function, and the system output test function. These functions are also frequently used.
3) Monitor: Provides a window that allows the user to monitor both system and user variables in real time.
4) UI Mode: UI mode can be selected according to the user's level and the user's purpose.

■ Utility sub-functions

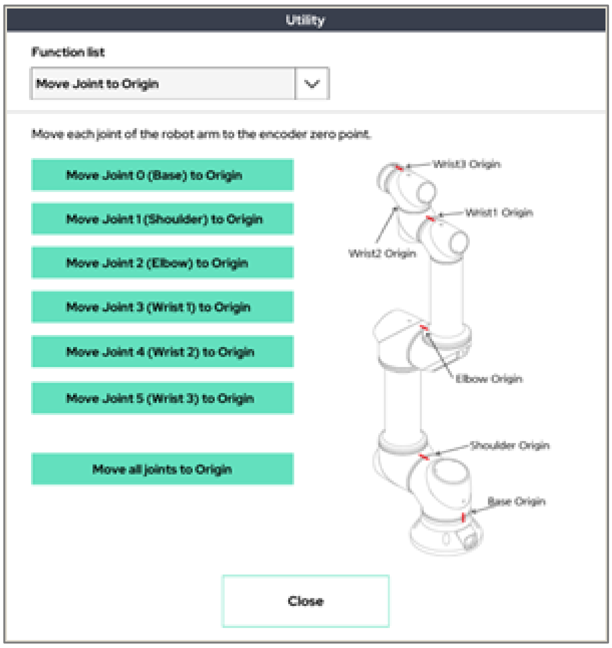
[Utility-Posture]



Up to 20 frequently used postures can be saved and used on the UI tablet. Press the 'Get current Posture' button to get the current position information and press the Set button to save it.

Hold down the 'Move to Saved Posture' button to move to the saved position.

[Utility- Move Joint to Origin]



Move each joint to the mechanical origin marked on the robot arm.

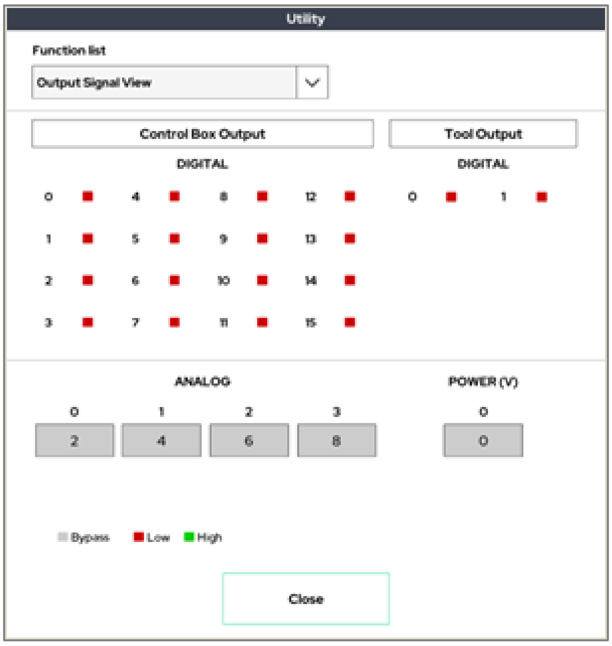
[Utility- Free-Drive with Constrained Orientation]

This mode is only available in Real Robot mode.

[Utility- Input Signal View]

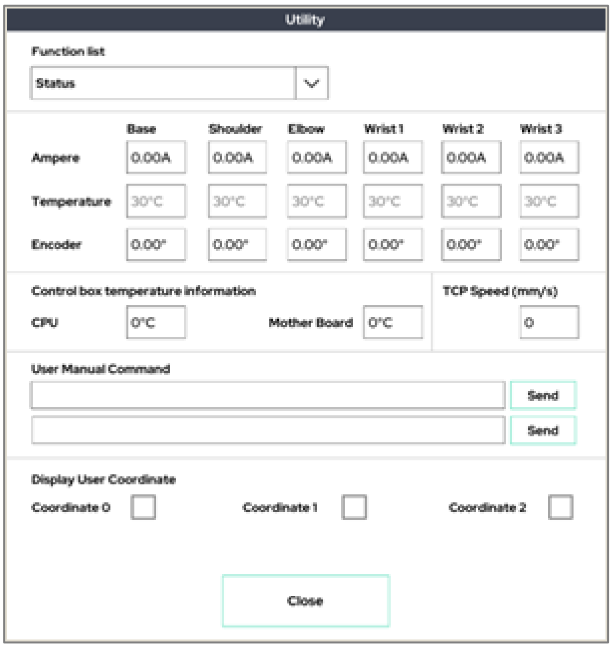
Input signal monitoring window for control box and tool flange.

[Utility- Output Signal View]



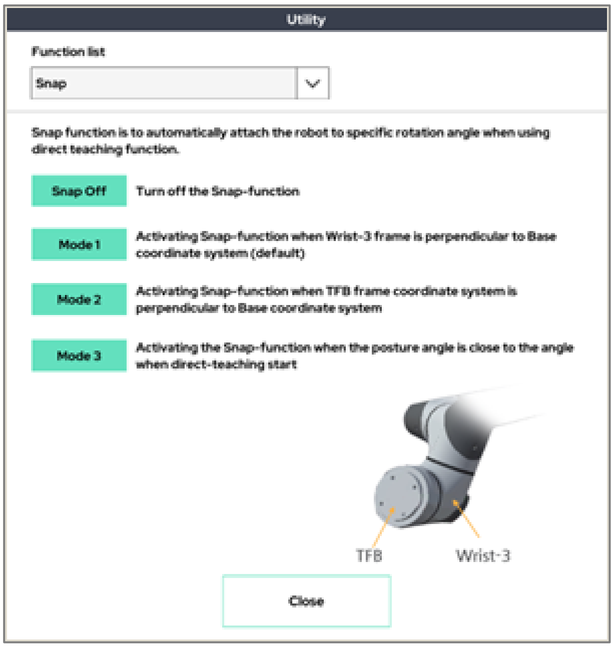
Output signal monitoring window for control box and tool flange.

[Utility- Status]



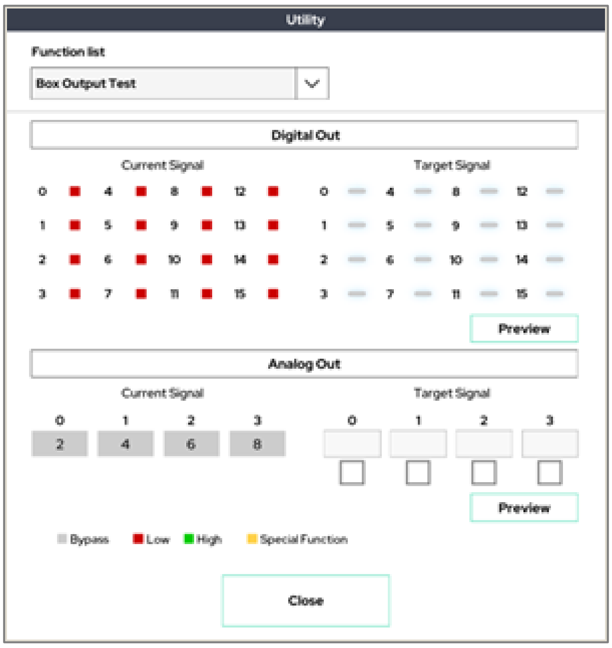
This window allows the user to see the robot arm's current and temperature. It also shows the user coordinate system settings.

[Utility- Snap]



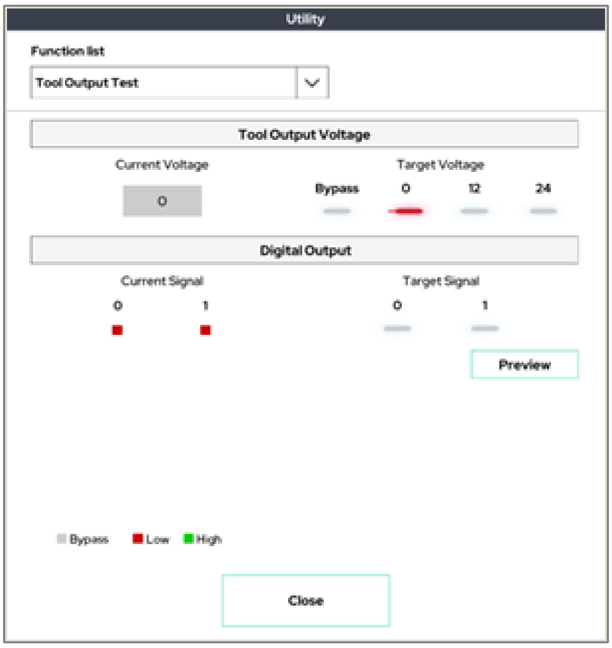
Snap mode selection window to be applied when using direct teaching mode.

[Utility- Box Output Test]



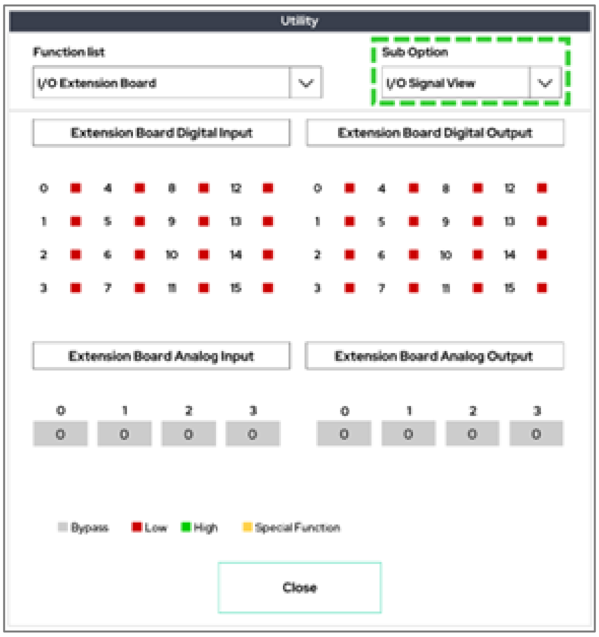
This window allows you to test the output of the control box.

[Utility- Tool Output Test]



This window allows you to test the output of the tool flange.

[Utility- I/O Extension Board]



I/O expansion module's I/O signal monitoring window.

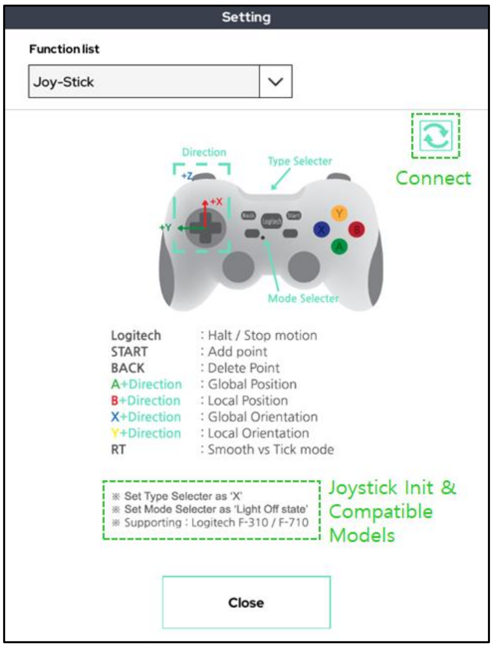
Window for testing the output of the I/O expansion module.

■ Setting sub functions

[Setting- Tool List Select]

There is a Tool List Select setup feature that sets up TCP to use in a pre-saved TCP list. A total of four tool lists can be set.

[Setting- Joystick]

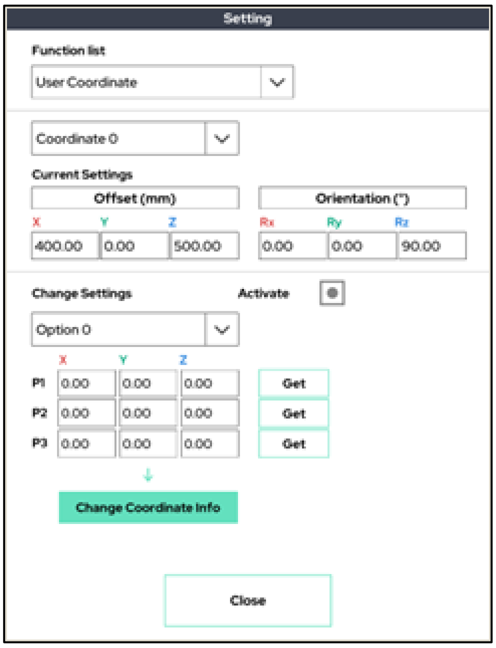


When teaching a robot using a wired/wireless joystick, there is a joystick connection setting function.

The joystick has the compatible model listed below and must be initialized.

When all settings are complete, click the button on the refresh mark at the top right to connect.

[Setting- User Coordinate]

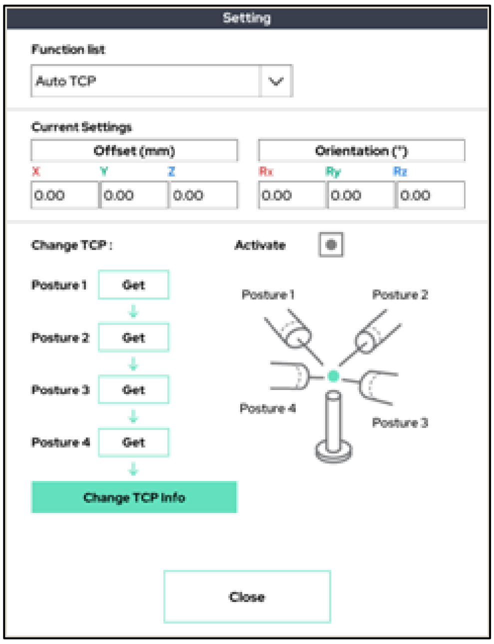


Set the user-defined coordinate system. Press the activation button to display a screen that selects three points. This is the 3-point setup mode (see 7.7 Setup-Coordinate).

[Setting- User Coordinate Center]

Keep the XYZ axis and rotation direction of the previously set user coordinate system, and move only the origin.

[Setting- Auto TCP]

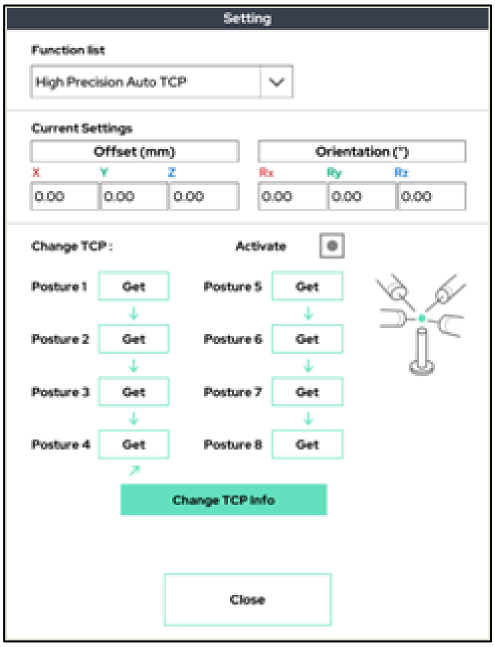


It automatically finds the position of the robot end (TCP).

If you enter four different postures that keep the TCP point you want to set at the same point in a three-dimensional space, it automatically calculates the TCP position value.

Take each position as shown, and press 'Get' on the corresponding number.

[Setting- High Precision Auto TCP]

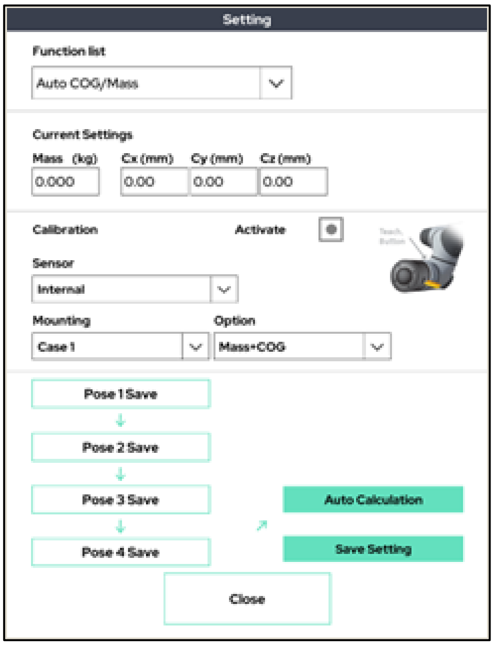


It automatically finds the position of the terminal, just like the robot terminal (TCP). However, it has more posture, so it looks for more precise TCP.

[Setting- External F/T]

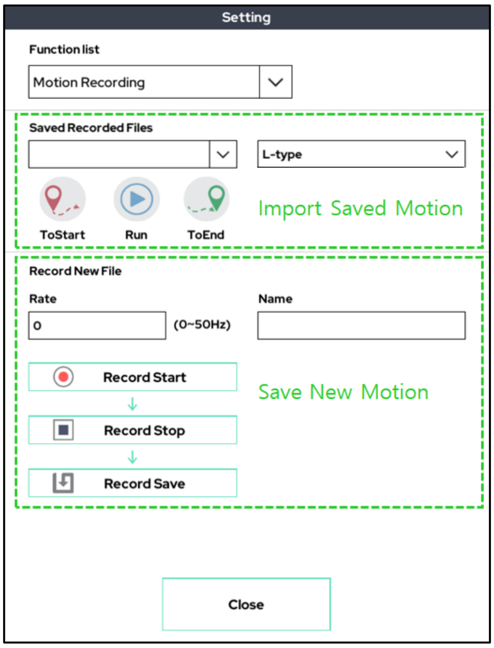
This window allows you to check and calibrate the external F/T sensor (e.g. Robotiq F/T sensor).

[Setting- Auto COG/Mass]



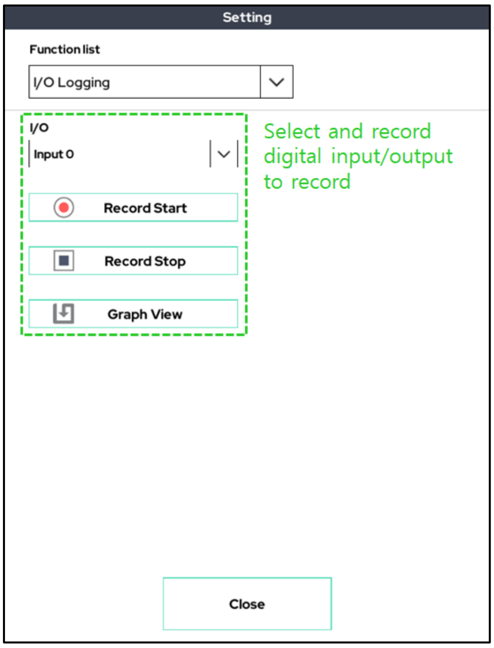
This function finds the weight and center of gravity attached to the tool using the internal / external F/T sensor.

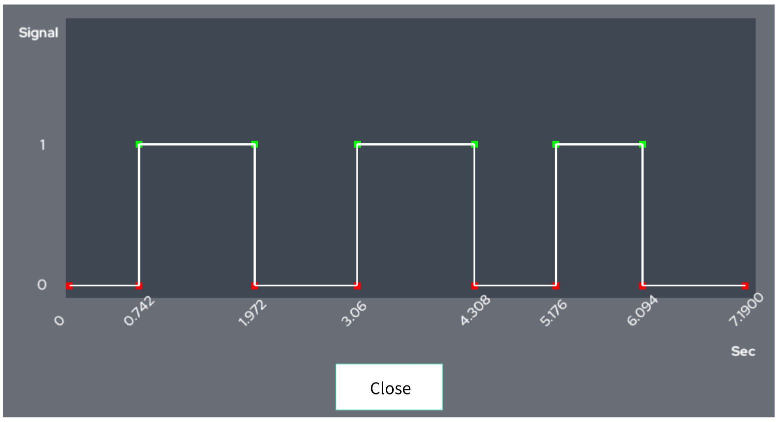
[Setting- Motion Recording]



It is a function that records the motion through the direct-teaching(gravity compensation) function. The recorded action is available in the program via the Replay function.

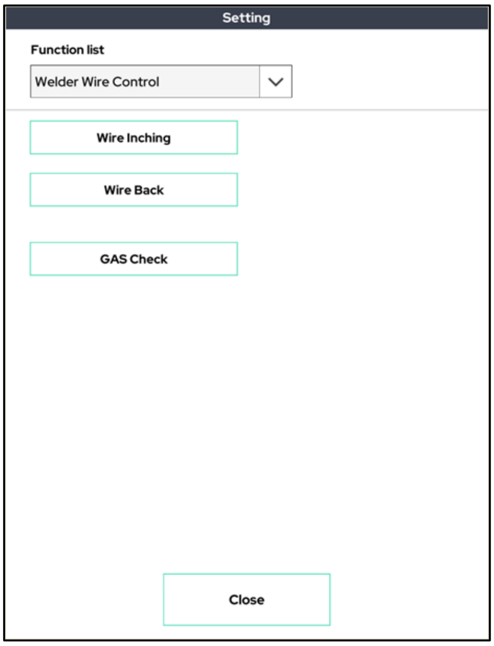
[Setting- I/O Logging]





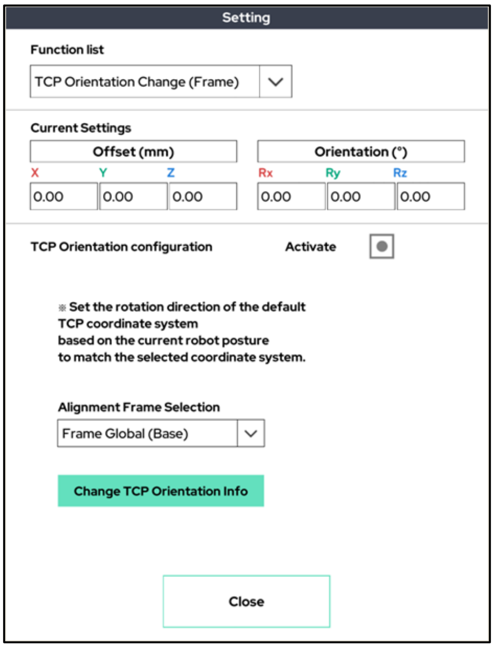
This function sets one digital input/output, records the change in the value of that input/output, and shows it graphically.

[Setting- Welder Wire Control]



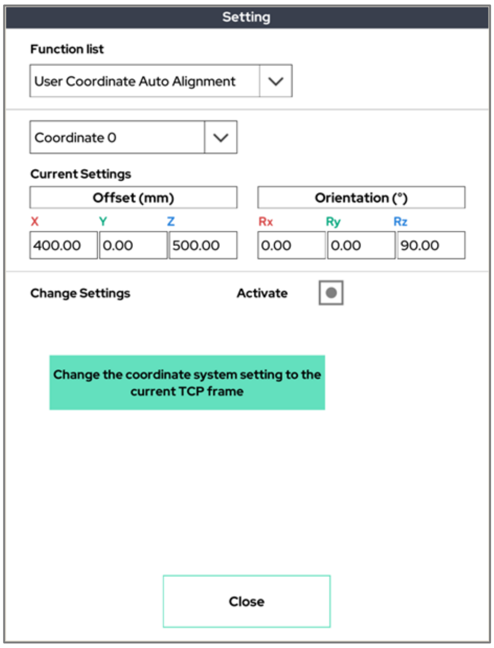
This function can control the welding machine's wire.

[Setting- TCP Orientation Change (Frame)]



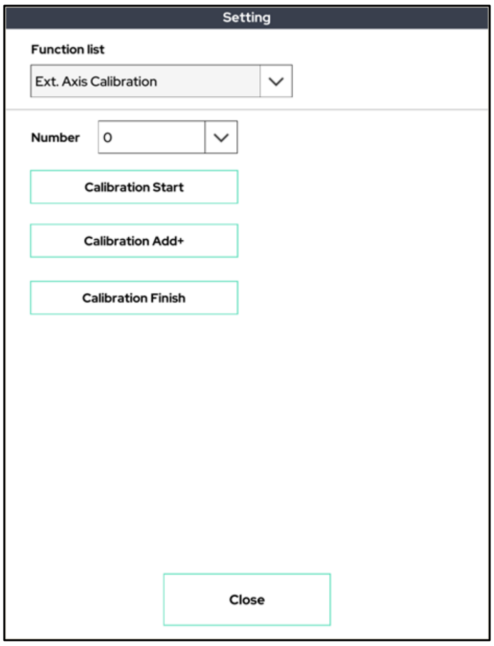
Sets the rotation direction of the default TCP coordinate system based on the current robot pose to match the selected coordinate system.

[Setting- User Coordinate Auto Alignment]



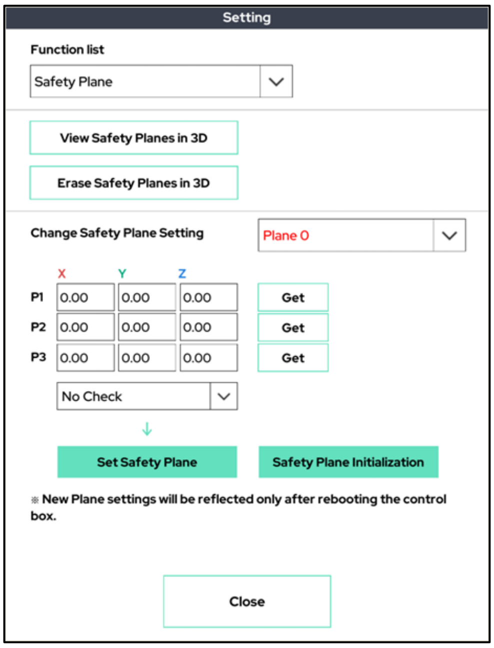
Change the coordinate system setting to the current TCP frame.

[Setting- External Axis Calibration]



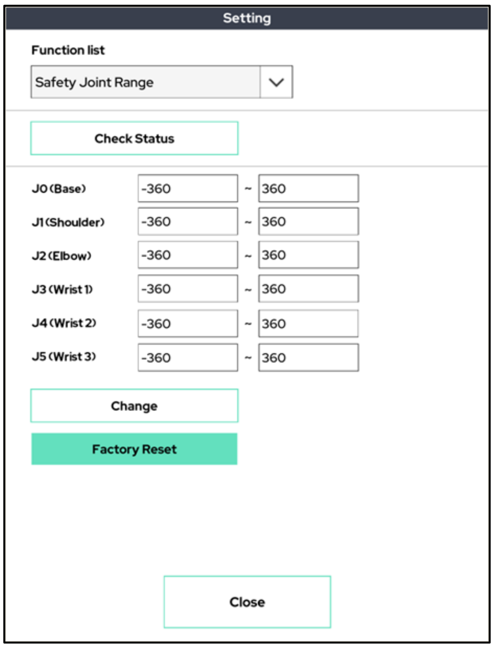
This is a process to inform the robot arm of the external axis of rotation and the radius of rotation. This calibration allows the robot to control the synchronization of the external axis.

[Setting- Safety Plane]



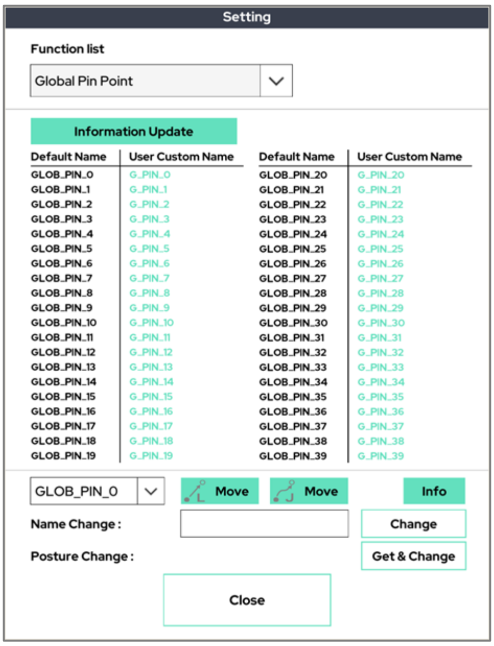
Set the plane that the robot arm cannot cross for safety. If it crosses the set plane, the robot arm will stop.

[Setting- Safety Joint Range]



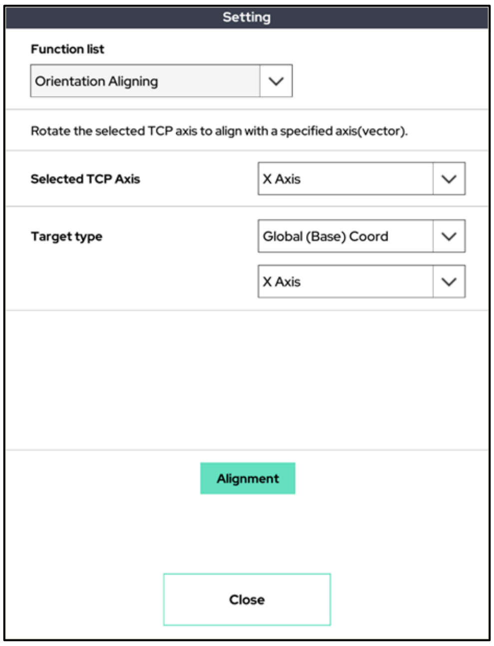
You can specify the range of angles that operate for each joint in the robot arm. An angle greater than or equal to the specified joint angle value will not operate.

[Setting- Global Pin Point]



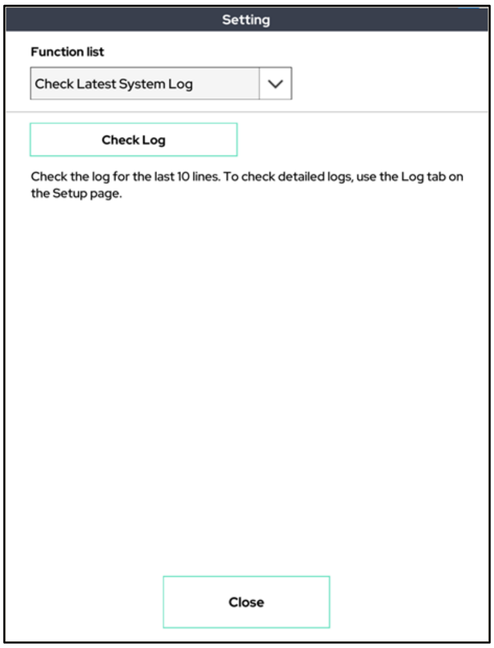
This is the function to set up global pin points. As shown in the above picture, you can set up global pin points to share the variables in multiple programs.

[Setting- Orientation Aligning]



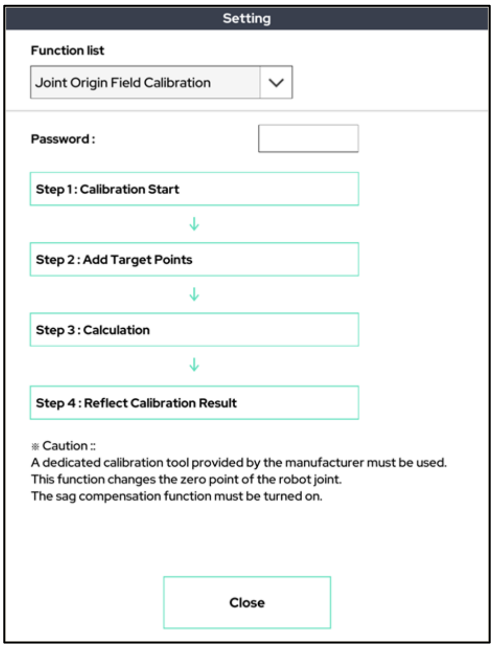
Rotates the axis of the selected TCP of the robot arm to align with the specific axis.

[Setting- Check Lastest System Log]



Displays the contents of 10 lines of the latest log file on the system on the UI.

[Setting- Joint Origin Field Calibration]



This function is to calibrate the joint 0 points on the spot. Please contact the manufacturer for more information.

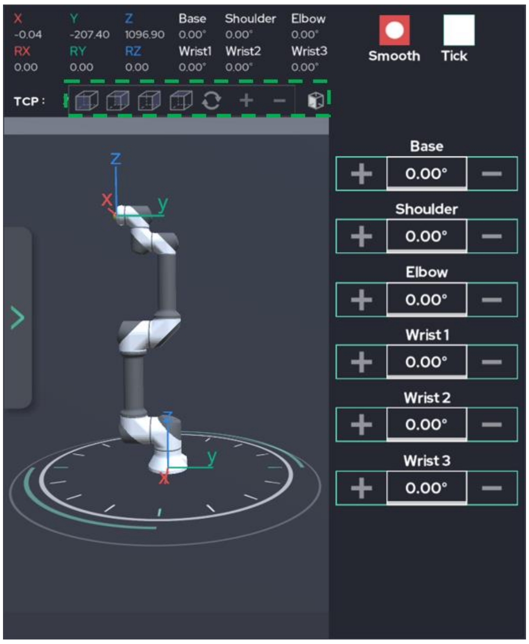
■ Monitor Function

This function is used in conjunction with the Monitor command in Section 6. This window allows the user to observe the system and user variables in real time.

By pressing the recording function on the upper right, the TCP trace of the robot tool is recorded in the 3D viewer in the 3D viewer. (Yellow solid line)

The recording time is up to 30 seconds.

■ 3D View Function



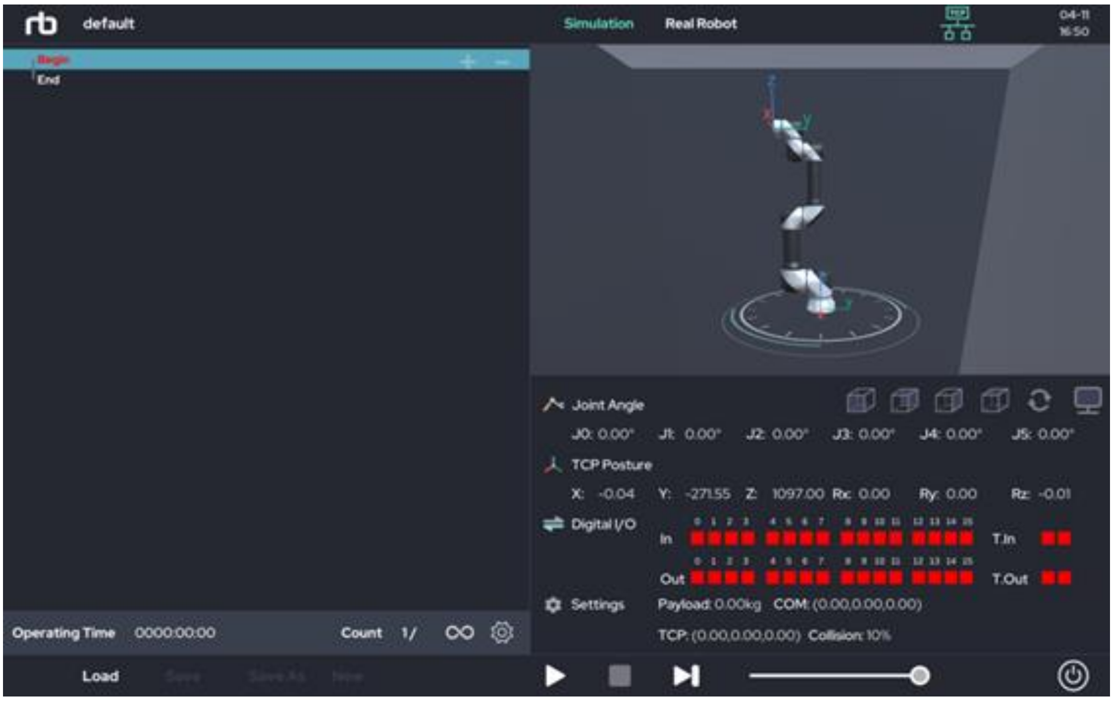
It has a three-dimensional viewing angle that is frequently used. It can be used if you want to quickly change the view of the three-dimensional viewer.

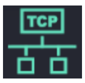
CHAPTER 8. STARTING ROBOT

8.1 ROBOT OPERATION

To drive the robot, go to the Play screen.

- The execution screen is as follows.



- Before using, please check the connection between tablet PC and control box. Check the  Icon to view the connection with the robot. Please refer to Chapter 6.2 for connection.
- Open the desired project. Please refer to Chapter 6.6 for more detail about how to open a project.
- Press the play (▷) button located at the bottom of the screen to run the robot.
- A dialog is pops up when the current robot position is different from the initial position specified. Press and hold the 'approach' button to move the robot to the initial position.
- In 'Play', the program loaded will repeat indefinitely if the 'number of repeat' is not specified. Press 'Count' at the top of the screen to set the 'number of repeat'.
- The motion speed of the robot can be adjusted while the robot is in operation.

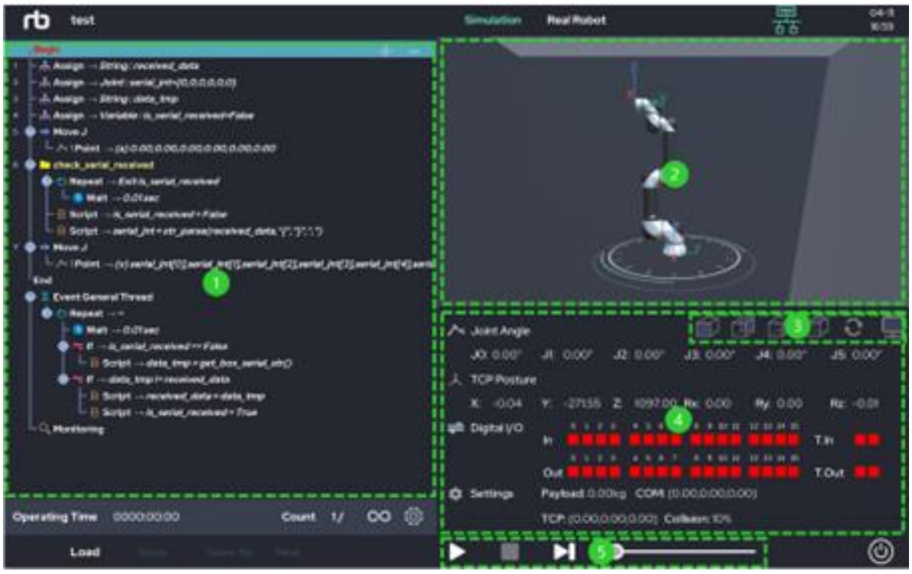
DANGER

Danger:

1) The risk assessment of the robot must be done, and all safety requirements must be satisfied before the robot operation.
2) The initialization of the robot may fail when the robot is not properly installed, the payload is not set accurately, or an issue occurs in the initialization process.
3) In 'Play', the robot physically moves immediately when the 'Play' button is clicked. Please read carefully all sections related to the robot operation.
4) To move to the 'Make' or 'Setup' screen, the program running must be terminated.
5) The USB cable between Tablet PC and control box can be unplugged during the robot operation.

8.2 ROBOT STATUS CHECK

The robot's current status is shown in the 'Play' screen during operation.

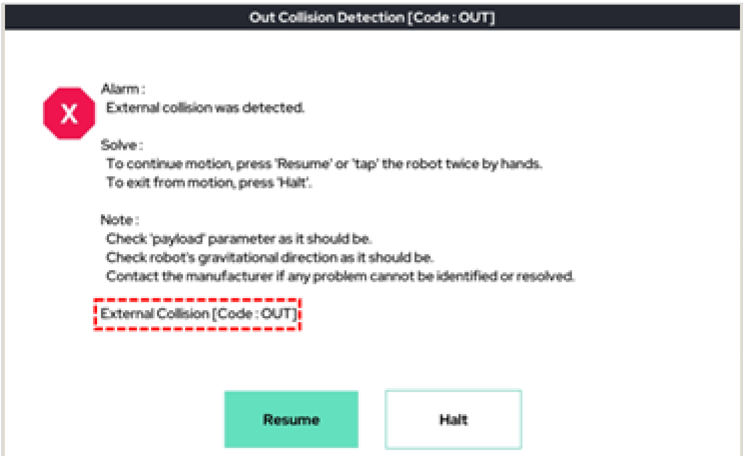


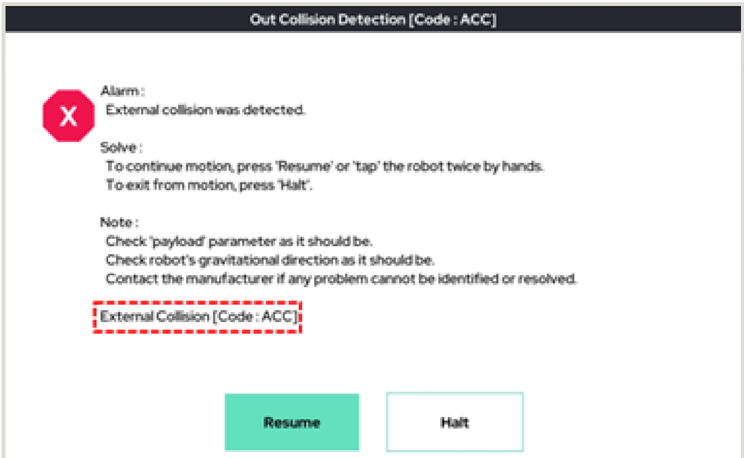
No.	Description
①	Program flow tree
②	3D viewer.
③	3D view angle changer
④	System information, system variable monitor
⑤	Play / Pause / Stop / Velocity slide bar

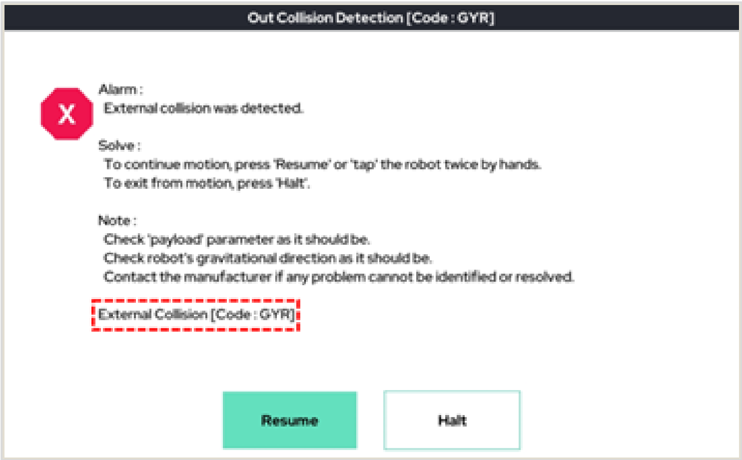
8.3 TROUBLESHOOTING WHILE OPERATING

Various problems can occur while the robot is in operation. Below are some of those problems and ways to troubleshoot.

1. Out Collision







The robot will stop immediately when a collision is detected. Looking at the figures above, there are external collision codes, each meaning the following:

OUT : When an overcurrent due to a collision is detected by the current sensor in each joint.

ACC, GYR : When sudden rapid vibrations caused by a collision are detected by the robot.

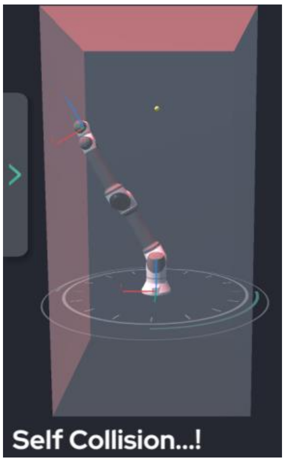
First, remove the external collision factors. Then, there are two options:

To stop the robot's task and proceed with maintenance, press the Halt button to end the task. To continue the robot's task, press the Resume button to resume the task.

TOK TOK Recovery method

When temporarily paused due to an external collision, lightly tapping the end of the robot arm twice will recover it from the paused state (same effect as pressing the Resume button).

2. Self Collision

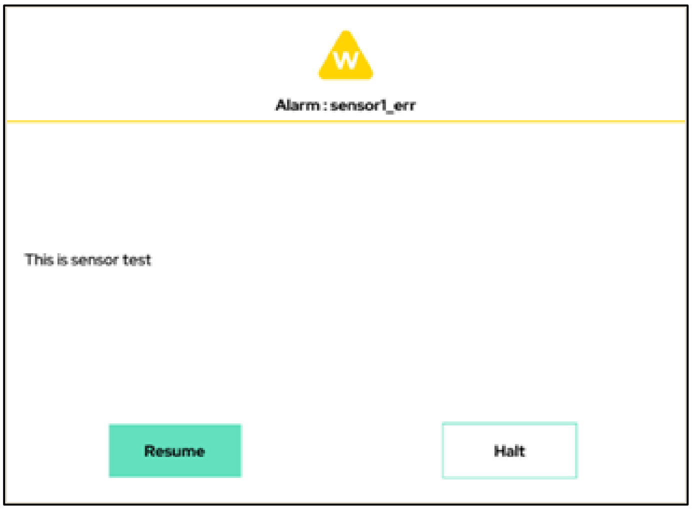


As shown in the first image, when part of the robot moves outside the designated working area, or as shown in the second image, when the robot approaches a motion that poses a collision risk, the robot will stop working.
You must use the teaching button to manually adjust the robot's position to a normal state. Afterward, review the taught motion and the set work range.

If this occurs in Simulation mode on the Make screen, you can recover using the following methods:

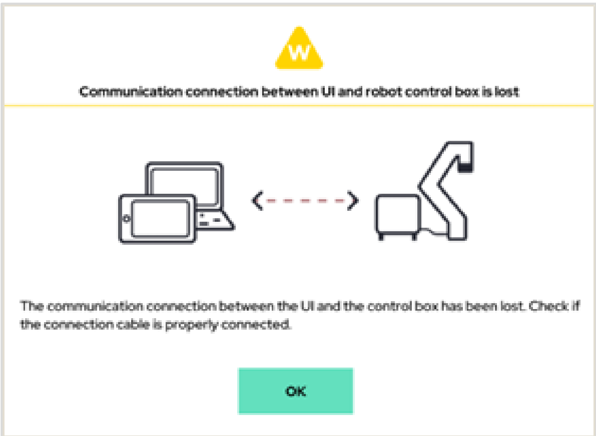
- Use the motion-related buttons to recover the robot from a self-collision situation.
- Temporarily switch to Real mode to receive actual joint information from the robot and recover.
- Press the teaching button to receive actual joint information from the robot and recover.

3. Alarm Message



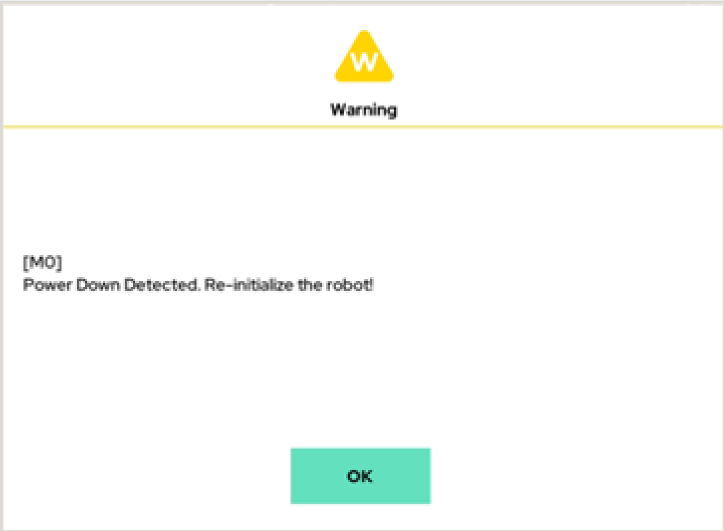
When an 'Alarm' is set in the current program, the robot will pause once the 'Alarm' command is reached. A dialog will then pop up. Press 'Resume' to continue the task or 'Halt' to stop.

4. Tablet PC Disconnection



Occurs when the control box (controller) and the tablet PC are disconnected or forcibly shut off the main power (220V) of the robot control box. If the USB cable is damaged, it needs to be replaced, and if the connection is not consistent, try a different USB port.

5. Interruption of external power supply



Occurs when the power to the robot arm is interrupted. Appears when the emergency stop switch is pressed. After the emergency stop switch is restored normally, initialize the robot again and use it.

6. Joint Controller Errors

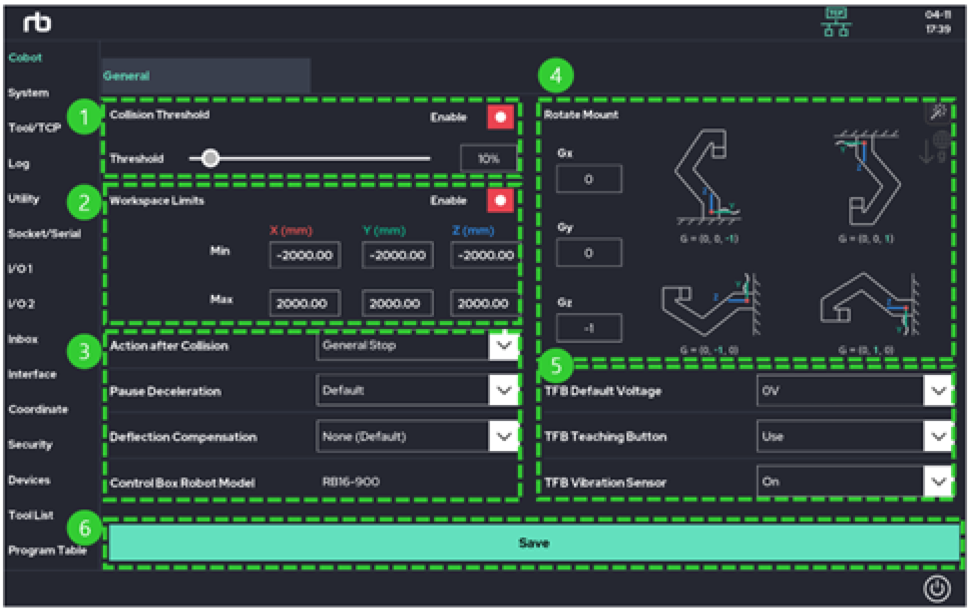
The robot will stop automatically when one of its joint controllers experiences one of the following errors:

BIG Error	The difference between the reference input and encoder value exceeds the factory specified threshold.
JAM Error	The encoder value does not change, but a current is supplied that is over the factory specified threshold.
CUR Error	The current exceeds the maximum current threshold.
Temperature Error	The temperature exceeds the maximum temperature threshold.
Mode Error	The version of software in the main controller is different from the version in the joint controller.

CHAPTER 9. ENVIRONMENT SETTING

9.1 SET-UP(COBOT)

You can set the basic environment for the robot arm.

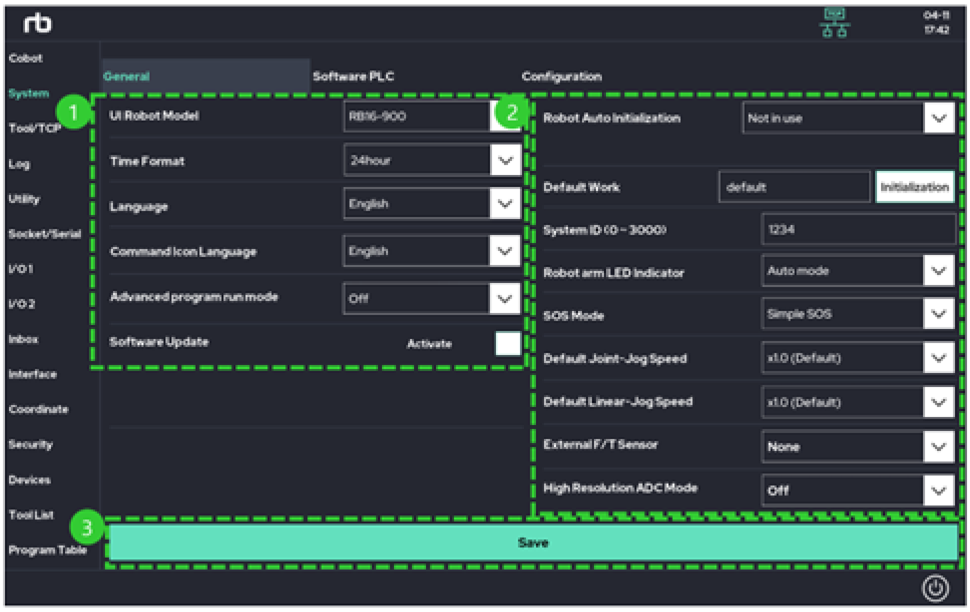


①	Set collision sensitivity. The lower the sensitivity, the more the robot will stop at weak impacts. The collision detection function can be turned on or off.
②	Set robot activity area. When you enable the Workspace, the activity limits will be displayed in 3D. If the robot exceeds the activity area, it will recognize this as a self-collision and stop. This function can also be turned on or off.
③	**Collision Detection Protection Mode**: After detecting a collision, the robot can either stop immediately (General Stop) or move in the opposite direction of the previous motion before stopping (Evasion Stop), which can be set step by step. The robot can also switch to direct teaching mode for a certain period after a collision (Free Drive Stop). **Pause Deceleration Amount**: If a pause is made during program playback, it will stop after a certain period of playback. This time can be set step by step.

	Droop Compensation: If an object is attached to the tool flange and the position value droops downwards, this function compensates for the droop. **Control Box Robot Model**: Displays the currently stored robot model in the control box.
④	Set the installation angle of the robot. If there is a matching installation angle among the example images provided, click on the corresponding image. To enter it manually, input the direction vector of gravitational acceleration based on the Global coordinate system.
⑤	**Tool Flange Default Output Voltage**: Select the default output voltage for the tool flange (can choose from 0V, 12V, or 24V). **Tool Flange Direct Teaching Button**: Choose whether to use the direct teaching button on the tool flange. **Tool Flange Vibration Sensor**: Choose to turn the vibration detection on or off.
⑥	Save the Changed Settings.

9.2 SET-UP(SYSTEM)

You can set display units, date and time, system updates, and more.



①	**UI Robot Model**: Change the robot model displayed on the UI. **Time Format**: Set the time format to either 12-hour or 24-hour format. **Language**: Set the language for the UI. **Command Icon Language**: Set the language for command icons. **Advanced Program Operation Mode**: Specify the starting position for the program from the beginning. **Software Update**: Install the software from the tablet to the control box. This function allows you to upgrade or downgrade the software version of the control box.
②	**Robot Auto Activation Feature**: Choose how to activate the robot. If set to 'Disable,' you will need to press the 'Activate' button each time to activate it. In 'Auto Activate with Key' mode, activation occurs through any digital input. In 'Auto Activate without Key' mode, the robot activates as soon as the power to the control box is turned on, without any operation.

	Default Work File: Set the initial file to be loaded when the control box boots. If no other file is set, it will default to 'default.'. **System ID(0~3000)**: Assign an ID to the system. **Robot Arm LED Indicator**: Enable or disable the LED on the robot arm link. **SOS Mode** : Simple SOS and SF SOS **Default Joint-Jog Speed** : Set the default joint speed. **Default Linear-Jog Speed** : Set the default linear speed. **External F/T Sensor**: Set the model for the external F/T sensor. **High-Resolution ADC Mode**: Enable or disable high-resolution ADC mode.
③	Save the Changed Settings.

①	Set how to use the Software PLC. If you want the settings to be applied regardless of whether the program is running (On (Always)), apply them only when the program is not running (On (Idle)), or apply them only when the program is running (On (Non-Idle)).
②	You can construct logic using various parameters such as input signals, system status information, system parameters, and communication data.
③	You can configure the logic of output signals using input signals.

①	Control Box Time Zone: You can set the current time of the control box.
②	Inversion of Digital Input Signals / Output Signal Inversion: You can invert the signals of individual ports.
③	System Elements: You can set the usage of the tool flange and I/O board.

9.3 SET-UP(TOOL/TCP)

You can configure the settings related to the installed tool.

①	The Tool Center Point (TCP) coordinate system for each robot model is displayed. Enter the values on the right based on this coordinate system.
②	Set the load applied to the robot. The mass should be entered in the 'Mass' field (unit: kg), and the input range varies depending on the robot model's rated load. The distance to the center of gravity is entered in the 'Center of gravity' field (unit: mm).
③	Configure the TCP coordinate system. You can set the linear position (X, Y, Z) and rotation angles (RX, RY, RZ).
④	Set the range of the tool area based on the TCP coordinate system. The area within this range is treated as a self-collision zone.
⑤	Save the changed settings.

9.4 SET-UP(LOG)

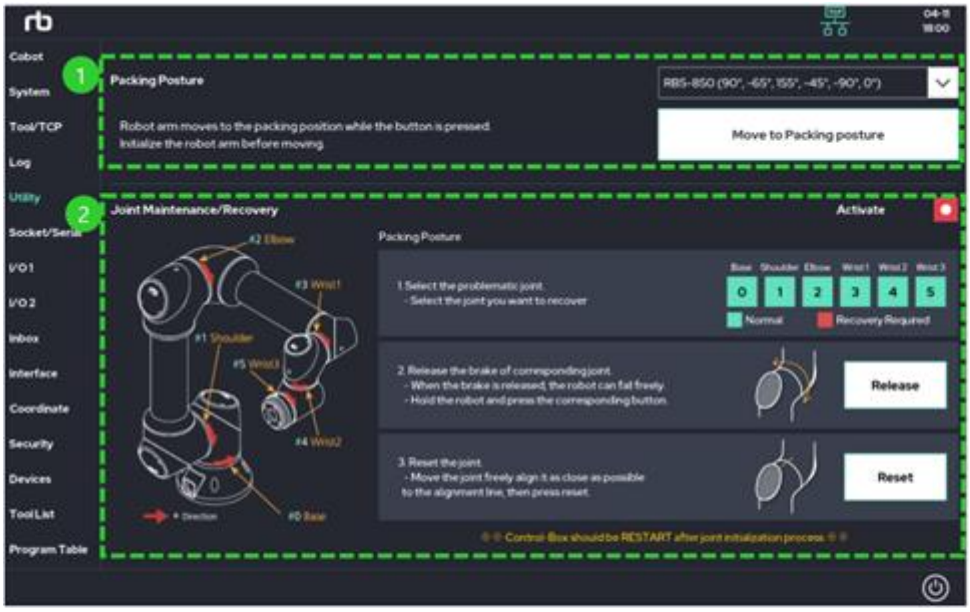
You can check the system log status of the robot arm.

①	Get: Import the LOG file from the control box to the tablet PC. Load: Load the LOG file that was imported from the control box to the tablet PC for output.
②	All: Output all logs. Info: Output only the informational logs. Warn: Output only the warning logs. Error: Output only the error logs. System: Output only the system-related logs. Fatal: Output only the logs related to errors that are directly associated with a failure of the robot. User: Output only the logs related to the logs specified by the user. Mail: Send the logs to Rainbow Robotics' customer support email account. Wi-Fi must be connected for this action.

③	Backup internal files of the control box: Backup the files stored in the control box to the tablet PC. Operating time of the robot arm: Output the time the robot arm has been operated. Copy control box settings: Copy the settings of the control box. Restore control box settings: Restore the previous settings of the control box.
④	Control box time zone: Displays the time zone currently recognized by the control box. System Ver.: Output the current UI version being used by the control box.

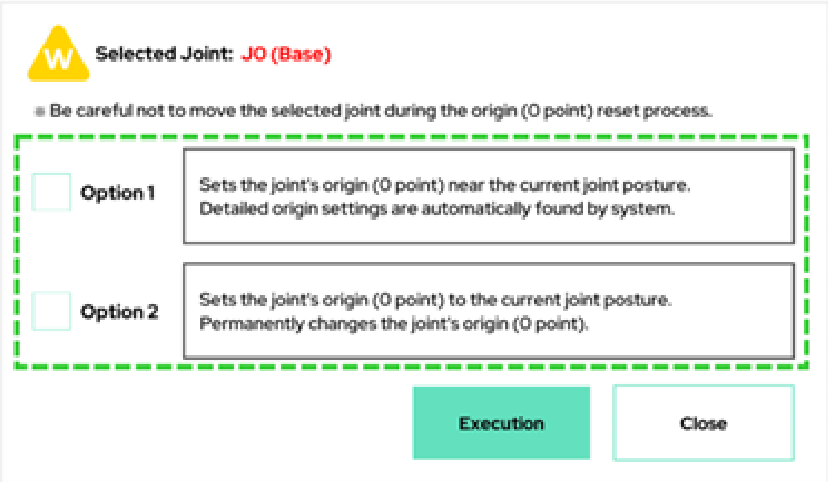
9.5 SET-UP(UTILITY)

Provides features regarding the robot's packaging and emergency recovery.



①	You can create the position for packaging the robot. Before using this function, ensure that the robot arm is activated. Remove all attachments from the robot arm prior to using this function. While holding the button, the robot arm will move to the packaging position. Once the packaging position is reached, an alarm will indicate that the robot has reached the packaging position.
②	You can set the axis reference point to the factory initialization state. This function is intended for recovery in case of abnormal operation, so it should be used with caution. Step1. Select the axis to reset. Step2. Press the Release button to release the brake and make it free. Step3. Align the groove marked on the axis and press the reset button to initialize the axis. At this point, options 1 and 2 will

	appear, as shown in the image below. Option 1 enables the system to correct any slight deviations introduced in step 2, while option 2 sets the origin as adjusted in step 2. Preferably, use option 1, but if the origin does not adjust as expected, use option 2 instead.



CAUTION

Caution:

1) Before using the Emergency Joint Recovery mode, understand the characteristics and operation of this function accurately before using it.
2) When transporting or shipping the robot, it must be placed in the packaging box provided by the manufacturer in the specified orientation.

9.6 SET-UP(SOCKET/SERIAL)

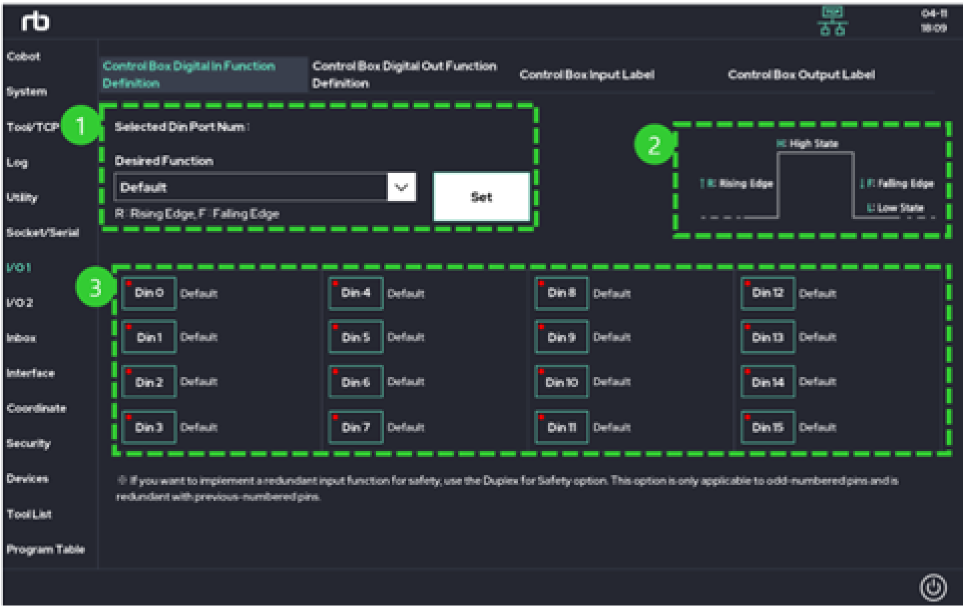
This is the section where you can configure the serial communication between the robot's tool and the control box.

①	You can assign a unique IP address to the control box. The default IP address of the control box is set to 10.0.2.7. When connecting the tablet PC wirelessly to the control box or using a LAN cable to connect a PC to the control box, ensure that the bandwidth of the IP address matches. For the Ping test, enter the IP address of the device connected to the control box and press the test button to check the connection status.
②	**High-speed External Script Mode**: While the usual control cycle for external script control is 10ms, the high-speed external script mode supports 1.8ms. **Command Port(5000)** : This port is used when an external device sends commands to the control box during external script control (refer to Appendix E for more details). **Data Port(5001)** : This port is used when an external device receives data from the control box (refer to the Data Structure for more details).

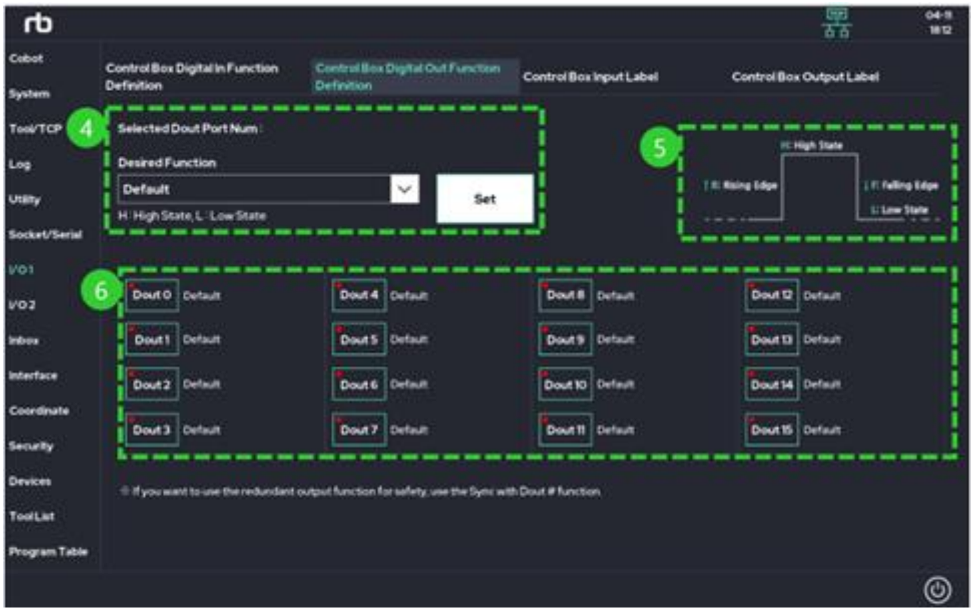
	Modbus Server Port(502) : This port is used for Modbus communication.
	Tool Flange: You can configure the communication speed, parity bit, and other settings for the RS485 on the tool flange side. **Control Box**: You can configure the communication speed, parity bit, and other settings for the RS485 on the control box side.
④	Save the modified settings.

9.7 SET-UP(I/O 1)

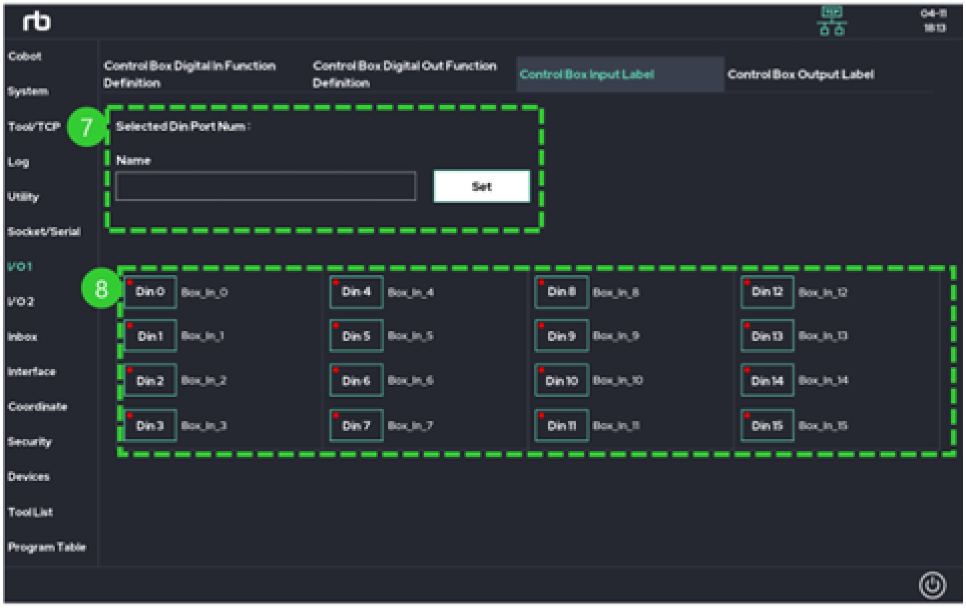
Here is where you can configure the functions of the GPIO ports on the control box.



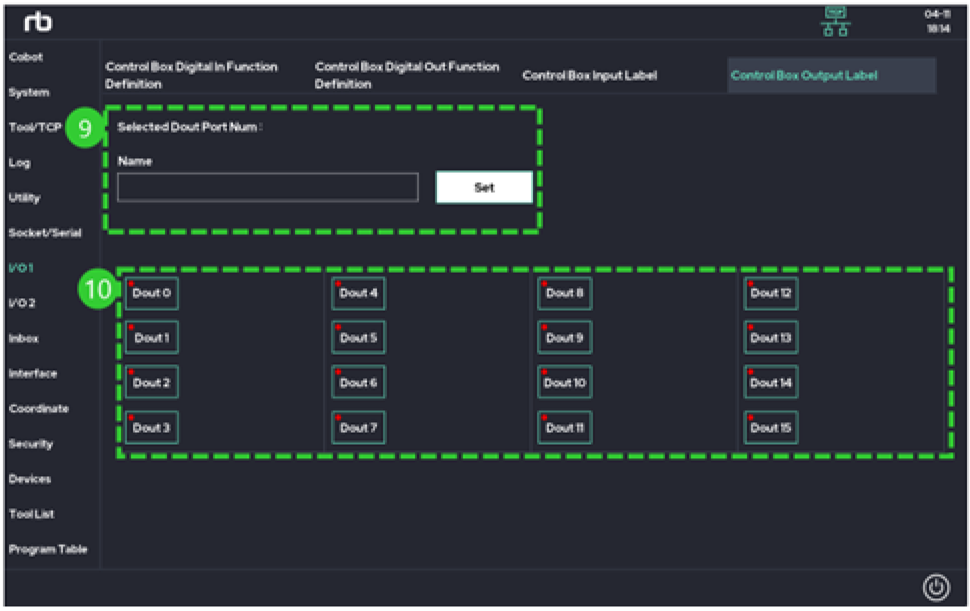
①	Select the input port for which you want to change the function.
②	**R** : The function applied to the selected digital input port is activated when the digital input pulse is at the Rising Edge. **H** : The function applied to the selected digital input port is activated when the digital input pulse is at the High State. **F** : The function applied to the selected digital input port is activated when the digital input pulse is at the Falling Edge. **L** : The function applied to the selected digital input port is activated when the digital input pulse is at the Low State.
③	The functions applied to the 16 digital input ports are displayed.



④	Select the output port for which you want to change the function.
⑤	**R** : The function applied to the selected digital output port is activated when the digital output pulse is at the Rising Edge. **H** : The function applied to the selected digital output port is activated when the digital output pulse is at the High State. **F** : The function applied to the selected digital output port is activated when the digital output pulse is at the Falling Edge. **L** : The function applied to the selected digital output port is activated when the digital output pulse is at the Low State.
⑥	The functions applied to the 16 digital output ports are displayed.



⑦	Enter the desired name.
⑧	Select the input port for which you want to set the name.



⑨	Enter the desired name.
⑩	Select the output port for which you want to set the name.

■ Control box input port functionality

The functions of the input ports (Din 0 to Din 15) in the control box can be set to the following special functions (R = Rising Edge, F = Falling Edge, H = High State).

0. Basic Mode (GPIO)
1. Start Program Once Mode (Rising Edge)
2. Stop Program (Rising Edge)
3. Pause Program (Rising Edge)
4. R = Turn on Direct Teaching Mode / F = Turn off Direct Teaching Mode
5. R = Set Speed to 100% / F = Set Speed to 0%
6. R = Switch to Real Mode / F = Switch to Simulation Mode
7. R = Reset Robot
8. H = Disable Collision Function when in High State
9. H = Auto Initialization Key
10. R = Release Pause due to External Collision
11. R = Add Point Function to UI Program (Rising Edge)
12. R = Start Program in Repeat Mode (Rising Edge)
13. R = Move to Begin Position
14. R = Resume Program
15. H = Quick Freedrive Change
16. R = Pause Program / F = Resume Program
17. F = Pause Program / R = Resume Program
18. H = Speed Slider at 100% / L = Speed Slider at 0%
19. R = Load Default Program
20. F = Turn Off Robot Arm Power
21. R = Touch Sensing
22. F = Touch Sensing
23. H = No Arc
24. H = Program Start Block

25. R = Ext.Joint0 Plus / F = stop
26. R = Ext.Joint0 Minus / F = stop
27. R = Ext.Joint1 Plus / F = stop
28. R = Ext.Joint1 Minus / F = stop
29. R = Ext.Joint2 Plus / F = stop
30. R = Ext.Joint2 Minus / F = stop
31. H = Safety Speed
32. F = UserCoord0 ←TCP frame
33. F = UserCoord1 ←TCP frame
34. F = UserCoord2 ←TCP frame
35. F = Load & Run Program Table
36. R = Start Prog. (Once) / F = Stop Prog.
37. R = Start Prog. (Repeat) / F = Stop Prog.
38. F = Change Global Pinpoint #'s Prog.
39. R = EXT. Joint 0 Slow Plus / F = Stop
40. R = EXT. Joint 0 Slow Minus / F = Stop
41. R = EXT. Joint 1 Slow Plus / F = Stop
42. R = EXT. Joint 1 Slow Minus / F = Stop
43. R = EXT. Joint 2 Slow Plus / F = Stop
44. R = EXT. Joint 2 Slow Minus / F = Stop
45. R = Reset SOS
46. F = Reset SOS
47. R = Delete Point
48. R = Add ARC-ON
49. R = Add ARC-Off
50. H = Speed 75% / L = 0%
51. H = Speed 50% / L = 0%
52. H = Speed 25% / L = 0%
53. H = Speed 10% / L = 0%

54. F = Pause Program
55. H = Block Freedrive Button
56. L = Block Freedrive Button
57. Duplex for Safety
58. F = Stop Program

CAUTION

Caution:

1) Before using Digital Input, be sure to fully understand the electrical characteristics of the digital input port provided by the manufacturer.

■ Control box output port functionality

The functions of the output ports (D.out 0 to D.out 15) of the control box can be set to the following special functions (R = Rising Edge, F = Falling Edge, H = High State).

Output ports set to special features are not available for general use in teaching programs.

0. Basic Mode (GPIO)
1. H = Program/Robot Running / L = Idle State
2. L = Program/Robot Running / H = Idle State
3. H = External Impact Detected
4. H = Direct Teaching Mode Active
5. Transmit the digital input signal with the same number as the selected port
6. Transmit Tool Flange Input Port 0 Signal
7. Transmit Tool Flange Input Port 1 Signal
8. H = Robot Active / L = Not Active
9. H = Real Mode / L = Simulation Mode
10. H = Robot in Operation / L = Idle
11. L = Robot in Operation / H = Idle
12. H = Robot Initialization (Activation) Failed
13. H = Power Supplied to Robot Arm / L = No Power to Arm
14. H = Collision Detection On / L = Collision Detection Off
15. H = Pause State
16. H = Trap Occurred in Inbox 0
17. H = Trap Occurred in Inbox 1
18. Used as PWM Module
19. H = Teaching Pendant Connected
20. H = Program Running in MAKE Screen
21. H = Program Running in PLAY Screen
22. H = Conveyor Mode Active
23. H = Control Box Boot Complete

24. H = Force Control Mode Active
25. PC Alive Pulse
26. H = Speed Slider at 100%
27. H = Last Program Load Successful
28. H = TCP is in InBox 0
29. H = TCP is in InBox 1
30. H = Is Alarm
31. H = Robot posture is Begin posture
32. H = Emergency Teaching Enable
33. H = Prog. Run in Sub.P area
34. H = Normal Program Run / L = Others
35. H = Normal Program Run / H = Others
36. H = Hand Controller F1 pressed
37. H = Hand Controller F2 pressed
38. H = Hand Controller F3 pressed
39. H = Hand Controller F4 pressed
40. L = TCP is in InBox 0
41. L = TCP is in InBox 1
42. Sync with D.out 0
43. Sync with D.out 1
44. Sync with D.out 2
45. Sync with D.out 3
46. Sync with D.out 4
47. Sync with D.out 5
48. Sync with D.out 6
49. Sync with D.out 7
50. Sync with D.out 8
51. Sync with D.out 9
52. Sync with D.out 10

53. Sync with D.out 11
54. Sync with D.out 12
55. Sync with D.out 13
56. Sync with D.out 14
57. Sync with D.out 15
58. H = Is SOS State
59. L = Is SOS State
60. H = EMG Button Released / L = Pressed
61. L = EMG Button Released / H = Pressed
62. L = Robot Speed under 250mm/s
63. H = Robot Speed under 250mm/s
64. H = Near Joint Limit
65. H = Robot Arm under Activating
66. H = Under Program Loading

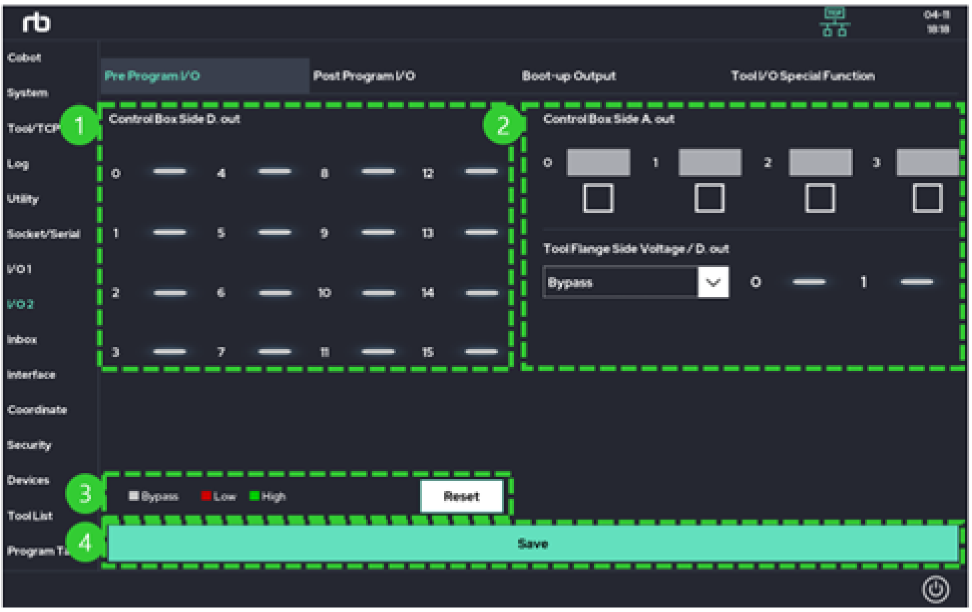
CAUTION

Caution:

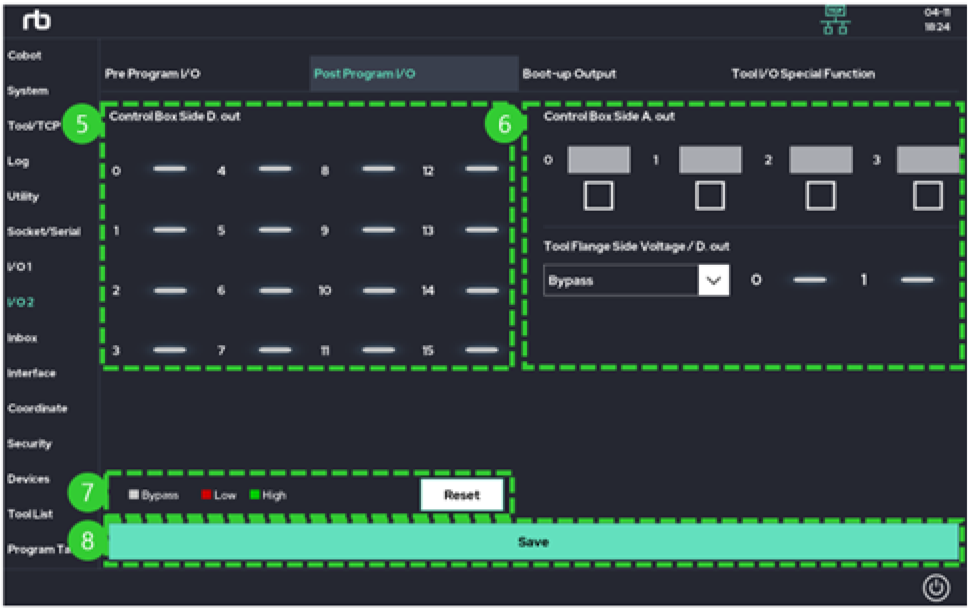
1) Before using the digital output, be sure to fully understand the electrical characteristics of the digital output port provided by the manufacturer.

9.8 SET-UP(I/O 2)

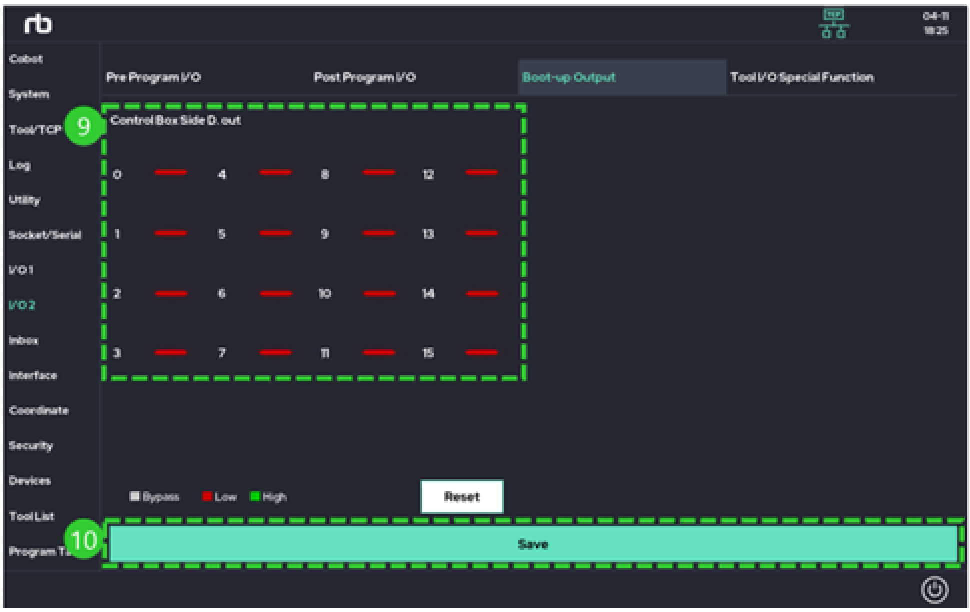
Setting I/O Values to Always Execute Before/After Program Start.



①	Set I/O Outputs to Execute Before Program Start (Playback) The ports set here will output the specified values as soon as the program starts.
②	**Control Box Side An.out**: You can set the voltage for analog output ports 0 to 3. The voltage can be set between 0V to 10V. **Tool Flange Side Voltage / D.out**: You can configure the digital output of the tool flange. Each port can be set to Bypass/0V/12V/24V.
③	Reset All Settings.
④	Save Settings.



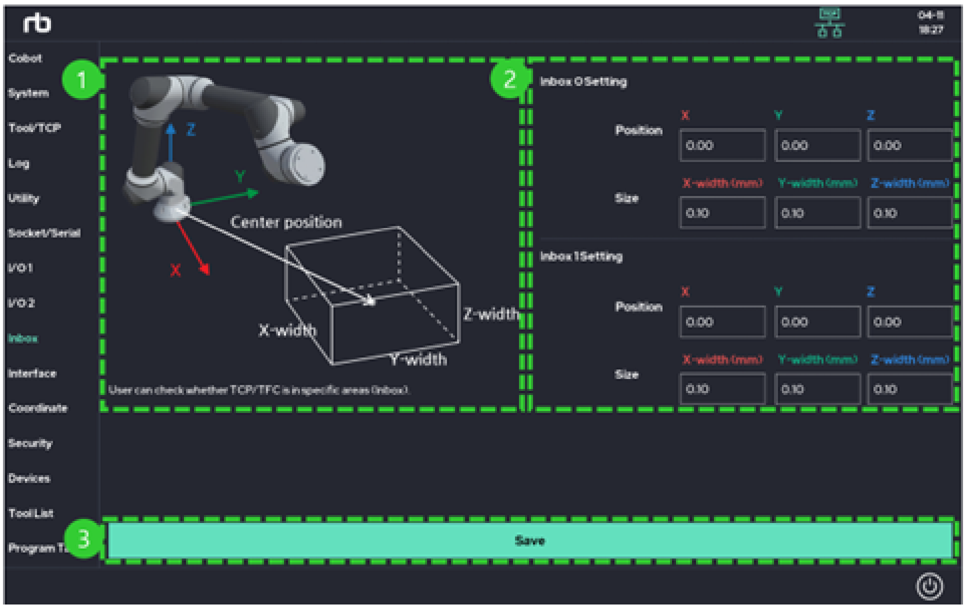
⑤	Set I/O Outputs to Execute After Program End. The ports set here will output the specified values as soon as the program ends.
⑥	**Control Box Side An.out** : You can set the voltage for analog output ports 0 to 3. The voltage can be set between 0V to 10V. **Tool Flange Side Voltage / D.out** : You can configure the digital output of the tool flange. Each port can be set to Bypass/0V/12V/24V.
⑦	Reset All Settings.
⑧	Save Settings.



⑨	Select Digital Output Options Immediately After Control Box Boot-up.
⑩	Save Changes.

9.9 Set-up(Inbox)

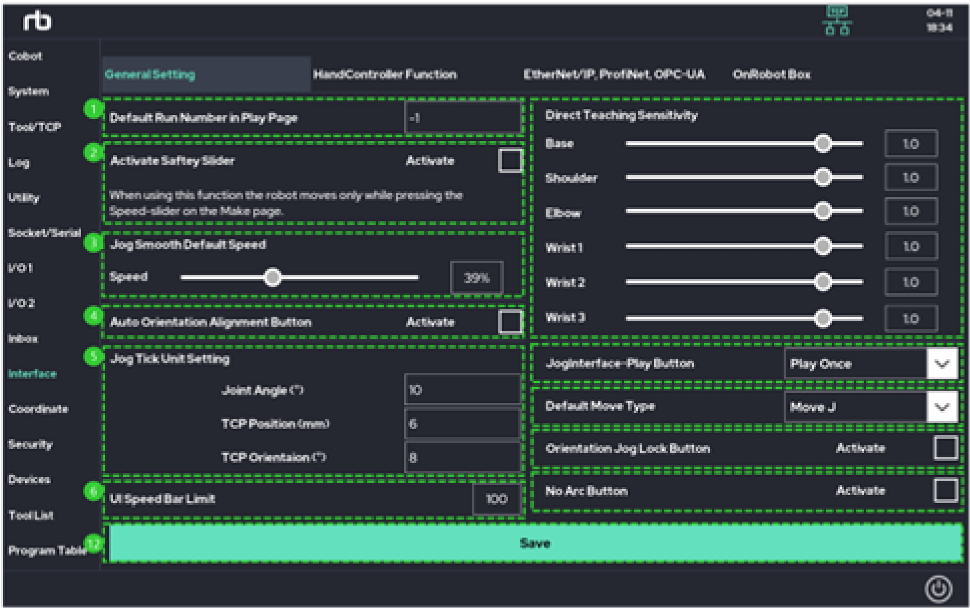
This section is for configuring the size and position of the Inbox for its usage.



①	Displays the Position Information of the Inbox. The position of the Inbox is set based on the robot arm base's coordinate system as the origin.
②	Set the Center Point and Width for Inbox 0 and Inbox 1. The orientation of the Inbox cannot be set. The reference coordinate system for the settings is the manufacturer's base coordinate system.
③	Save the Modified Settings.

9.10 SET-UP(INTERFACE)

Perform the settings required for the user to operate the robot and UI.



①	Enter the number of repetitions on the Play screen.
②	Select whether to use the safety slider function on the Make screen. When the safety slider function is enabled, if the user removes their hand from the screen, the speed adjustment bar will return to 0. When the safety slider function is disabled, the user can set the speed adjustment bar to the desired level, and even after removing their hand from the screen, the speed setting will remain as it is.
③	Set the speed to use when the Smooth option is enabled in the Jog function on the Make screen.
④	Choose whether to visualize the auto-rotation alignment button at the bottom of the Jog function on the Make screen.
⑤	Set the movement amount when the Tick option is enabled in the Jog function on the Make screen.
⑥	Limit the upper value of the speed adjustment bar in the UI.
⑦	Set the direct teaching sensitivity for each joint. The higher the sensitivity, the less force is needed to operate the robot manually.
⑧	Define the role of the play button on the dedicated jog/emergency stop interface device provided by the manufacturer.
⑨	Set the default settings for operations on the Make screen.

⑩	Choose whether to visualize the auto-rotation alignment button at the bottom of the Jog function on the Make screen.
⑪	Enable or disable the No Arc button on the Make screen.
⑫	Save the modified settings.

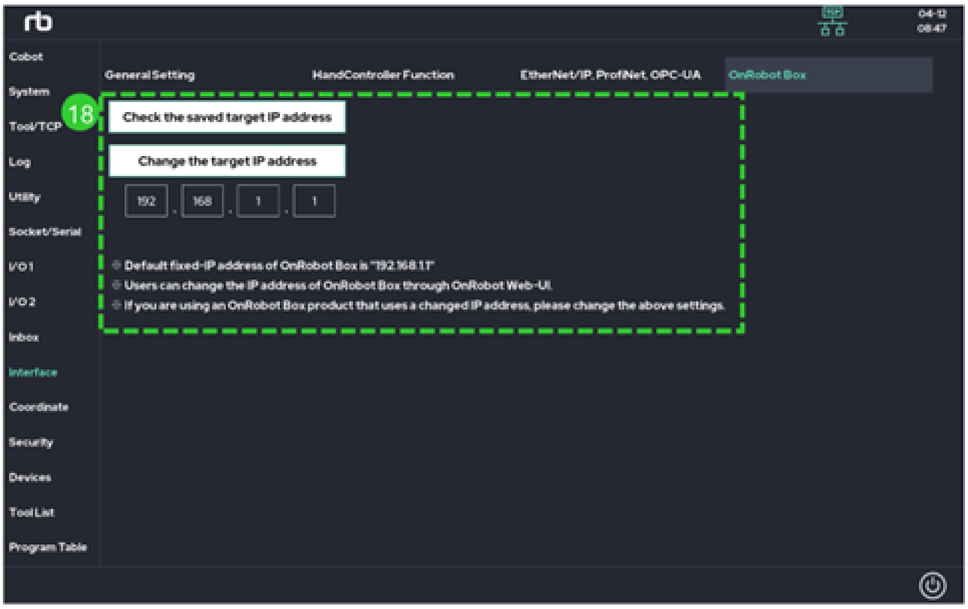
⑬	Check the setting values of the additional function buttons below the emergency stop button. When you press the additional function button, the corresponding button number lights up. This allows you to check if the UI is functioning properly.
⑭	**Change Button Functions**: You can assign functions to each button. **Hand Controller Attachment Status**: When the robot arm is activated, you can configure whether to use the hand controller or not.

CAUTION

Caution:

1) For safety reasons, use the mounted (Connected) state of the hand controller as much as possible.

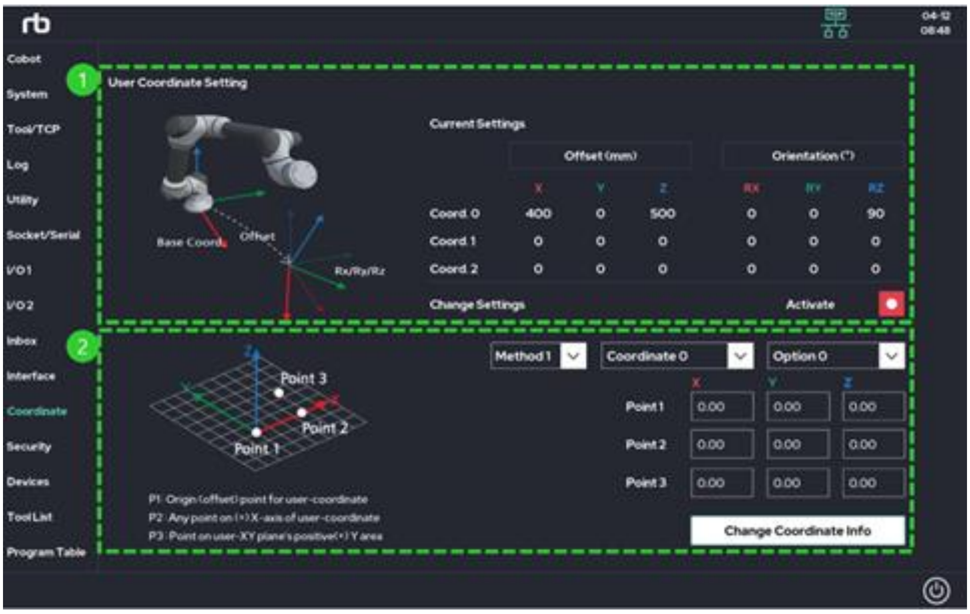
⑮	Enable or disable the Ethernet/IP Adapter function and check its current status.
⑯	Enable or disable the ProfiNet Device function and check its current status.
⑰	Enable or disable the OPC-UA Server function.



⑱	Set the IP address for configuring the On-Robot Box.

9.11 SET-UP(COORDINATE)

This section is for configuring a user-defined coordinate system (User-Coordinate).



①	Displays the Current User-Defined Coordinate System Information. The information is displayed relative to the manufacturer's robot base coordinate system.
②	Modify the Settings for User-Defined Coordinate Systems. You can set up to three user-defined coordinate systems. Each coordinate is defined with the robot arm base as the origin. Set the coordinate system through the 3-point setting. Point 1 (P1): This is the origin of the user-defined coordinate system. Point 2 (P2): This is any point on the X-axis of the user-defined coordinate system. It is recommended to set P2 as far from P1 as possible. Point 3 (P3): This is any point on the XY plane of the user-defined coordinate system. The direction of the Z-axis is determined based on the position of Point 3.

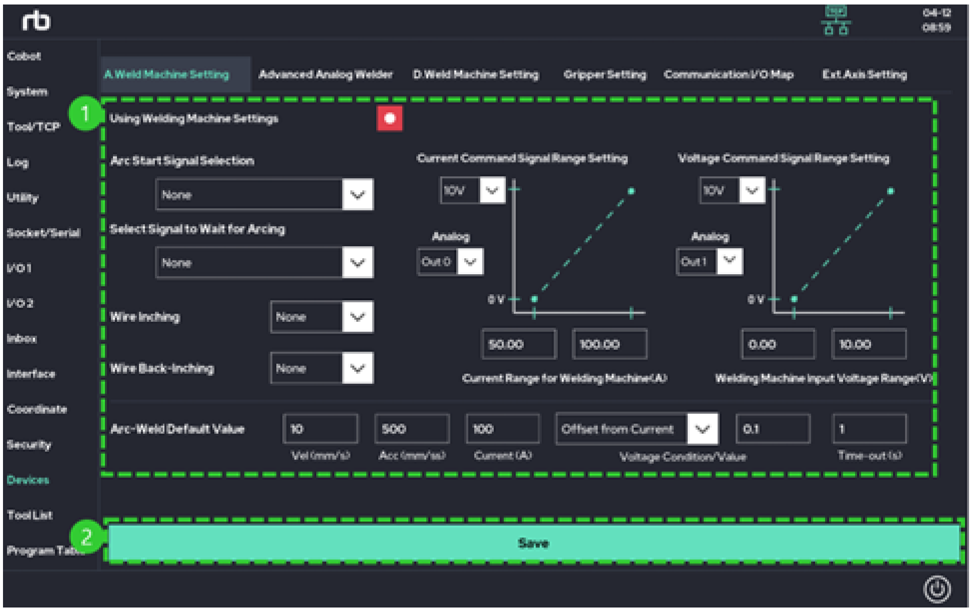
9.12 SET-UP(SECURITY)

This section allows you to set passwords for different menu screens or configure display settings.

①	You can assign a password to each screen handled by the UI.
②	**Always Keep Screen On**: Enables the screen to remain on when using Make and Play modes. (This prevents disconnection when the screen is off, but may drain the battery quickly, so use with caution.) **Gesture Mode**: When enabled, you can navigate the screen using up/down gestures.

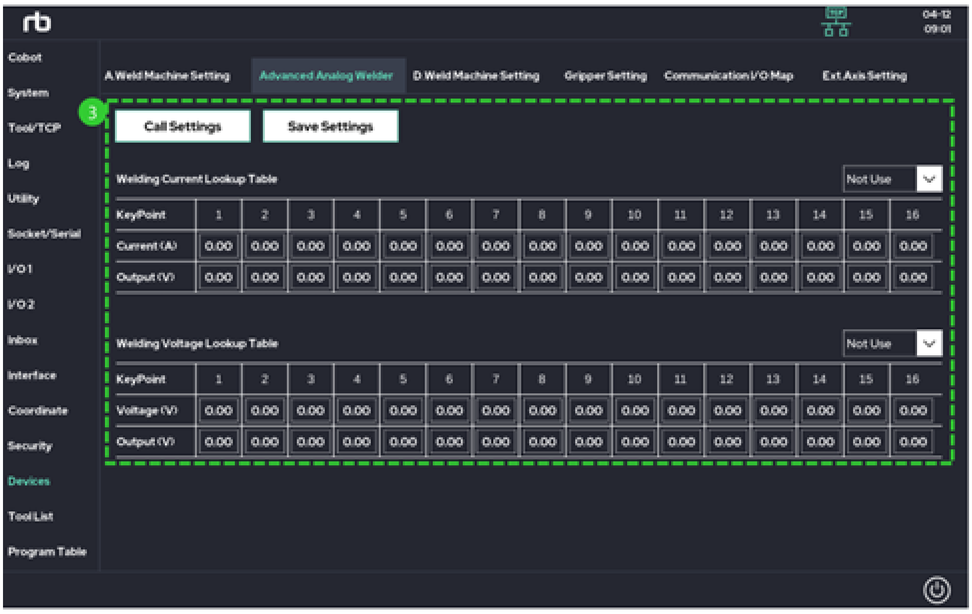
9.13 SET-UP(DEVICES)

Set up additional devices (Devices) connected to the robot.

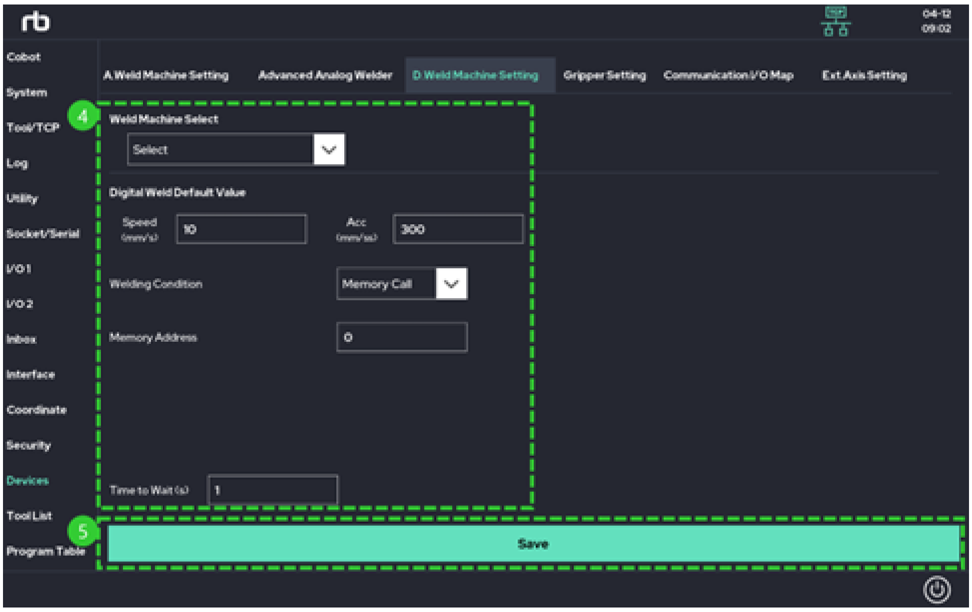


①	Setting can be carried out for the analog welding machine and parameters related to the welding output are set as a linear function.
②	You can save the changed settings.

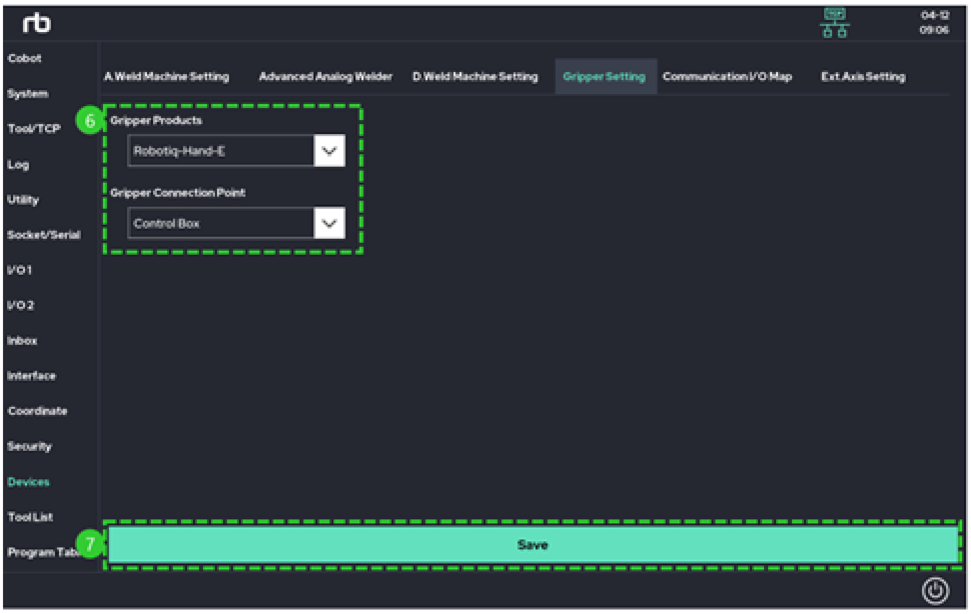
Set up the equipment connected to the robot.



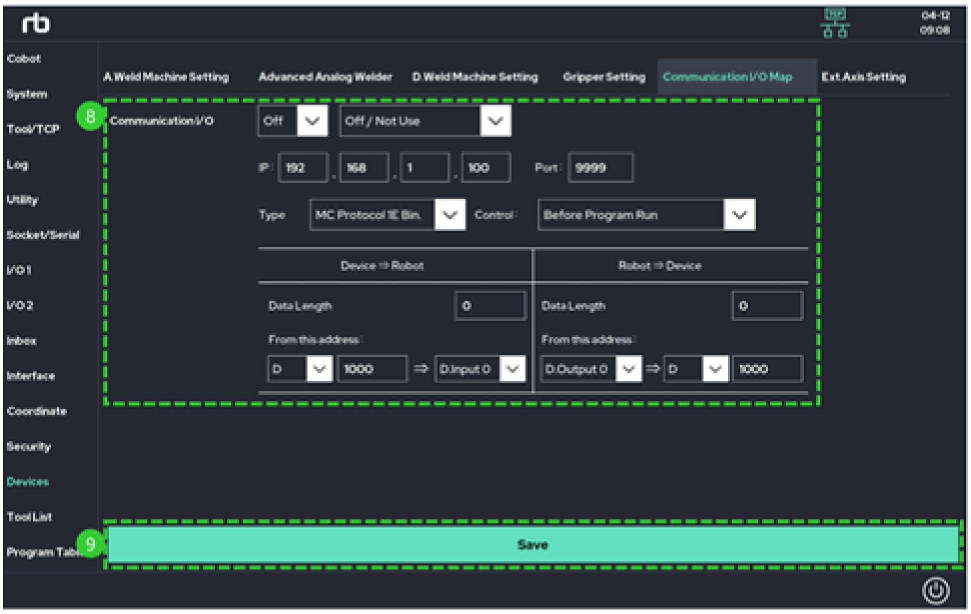
③	The parameters associated with the welding output can be set to the user's discretion.



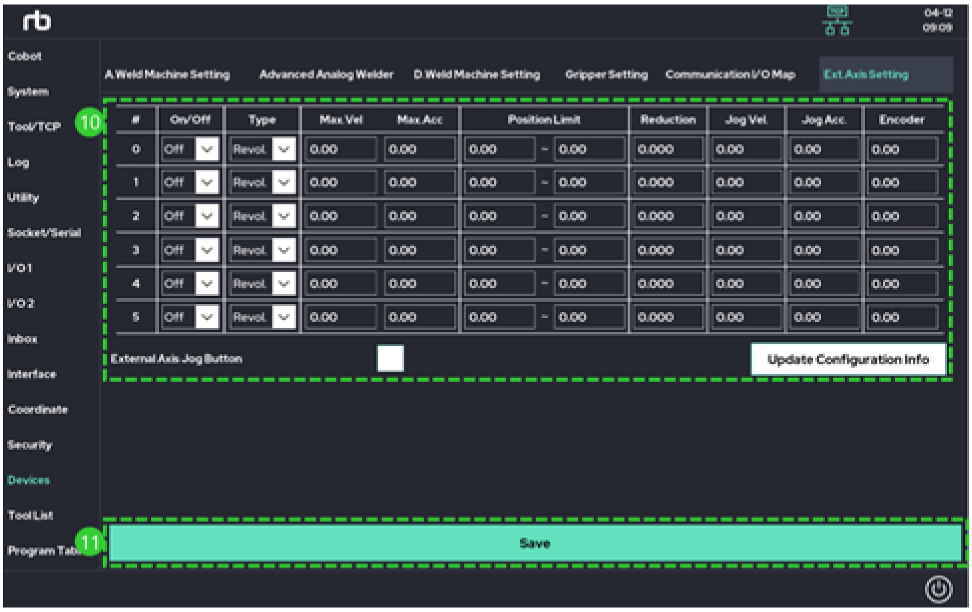
④	If you use the digital welder function in the Make window, you will immediately see what you set in the corresponding compartment.
⑤	You can save the changed settings.



⑥	If you use the gripper function in the Make window, you will see what you set up in the corresponding space immediately.
⑦	You can save the changed settings.



⑧	You can map the device in the PLC to the I/O in the control box, and it can be used even if the program is not running.
⑨	You can save the changed settings.



⑩	Proceed with the settings for each parameter on the external axis.
⑪	You can save the changed settings.

9.14 SET-UP(TOOL LIST)

This section is for configuring the TCP list (Tool List).

①	This shows the current settings for the TCP.
②	Select the TCP Number You Want to Modify. You can configure the name, TCP position, and center of mass, then save the changes.
③	Save the Modified Settings.

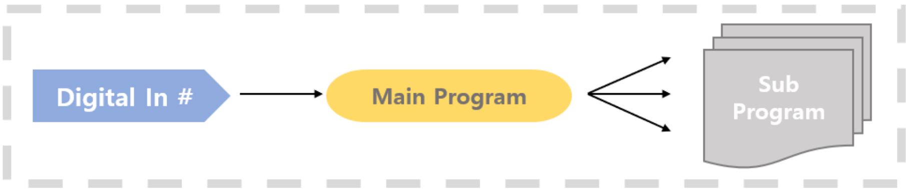
9.15 Set-up(Program Table)

You can set up the programs to be triggered by each digital input.

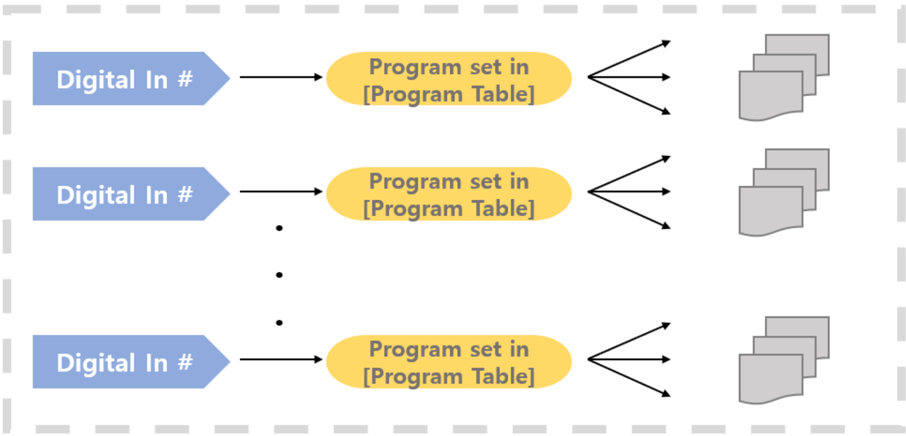
①	Import the existing settings.
②	Select the digital input port to be used.
③	Choose the function to use. The functions are categorized as Load, Load + Play (Once), and Load + Play (Repeat).
④	Select the project to be used via the digital input.
⑤	Save the configured settings.

The existing Start Program function under 'Setup > I/O1' in Control Box Digital Input Function Definition can only utilize one digital input and can only execute the main program currently uploaded to the control box. In contrast, Setup > Program Table allows you to load different programs for different digital inputs, enabling program execution and other functionalities.

You can see the differences between the two functions in the diagram below.



SetUp > I/O1



SetUp > Program Table

CHAPTER 10. MAINTENANCE

10.1 CHECK LIST AND PERIOD

The robot requires regular maintenance to perform in the best condition. As such, a regular maintenance schedule is highly recommended. During maintenance, the following check list has to be done. If you encounter an issue during the inspection that you cannot resolve on your own, please contact the manufacturer.

<table>
<tr><th>Check Item</th><th colspan="2">Check Point</th><th>Period</th></tr>
<tr><td rowspan="5">Robot Arm</td><td rowspan="3">Robot</td><td>Check that the robot moves to the desired location properly.</td><td rowspan="2">Daily</td></tr>
<tr><td>Check that the robot keeps its pose between being turned on and off.</td></tr>
<tr><td>Remove stains, dust, and any contamination.</td><td>Every 3 months</td></tr>
<tr><td>Motor</td><td>Check if a joint becomes irregularly hot or noisy.</td><td>Daily</td></tr>
<tr><td>Screws</td><td>Check that all screws on the body are tightened.</td><td>Every 3 months</td></tr>
<tr><td rowspan="2">Control Box</td><td>Cable</td><td>Check the connection of cables</td><td rowspan="2">Every 6 months</td></tr>
<tr><td>In-Box</td><td>Remove dust in the control box.</td></tr>
</table>

WARNING

Warning:

1) During maintenance, cut off the power to the system (Control Box and Robot Arm) and perform work.
2) For pneumatic/electric line passing models, remove the connected energy source (pneumatic/electric power) and perform the work.

10.2 ROBOT ARM MAINTENANCE

- Maintenance Period

  The robot arm requires an inspection at least per 1 year. Depending on the wear and tear, the maintenance period may differ.

- Maintenance Instruction

  1. Move the robot to the 'Home' position.
  2. Turn off the control box.
  3. Check the following list.
     ① Robot-Control Box Cable: Is it cut or stabbed?
     ② Screws: Are any loose?
     ③ Mechanical Parts (Motor, Brake, Reduction Gear): Are any louder than normal?
  4. Remove stains, dust, and any other contamination.

10.3 ROBOT CONTROL BOX MAINTENANCE

Dust in the control box may cause it to over-heat or generate electrostatic.

These can potentially damage the control box. It is required to regularly clean up dust in the control box.

- Maintenance Period

  The control box requires an inspection and clean-up at least once per 6 months. Depending on the environmental condition around the robot, the period may differ.

- Maintenance Instructions
  1. Turn off the control box.
  2. Remove the cover of the control box.
  3. Check if there is any dust accumulated inside the robot control box.
  4. If dust is present, carefully remove it using a vacuum cleaner or similar tool.
  5. Check that all wires are connected properly.

APPENDIX A. SYSTEM SPECIFICATION

Robot Arm

	Specification
Payload	RB5-850E Series: 5 kg / 11 lbs RB3-1200E Series: 3 kg / 6.6 lbs RB10-1300E Series: 10 kg / 22 lbs RB16-900E Series: 16 kg / 35.2 lbs RB3-730ES Series: 3 kg / 6.6 lbs RB6-920ES Series: 6 kg / 13.2 lbs RB20-1900ES Series: 20 kg / 44.1 lbs
Weight	RB5-850E Series: 22 kg / 48.5 lbs RB3-1200E Series: 22.4 kg / 49.4 lbs RB10-1300E Series: 37.1 kg / 81.8 lbs RB16-900E Series: 32 kg / 70.5 lbs RB3-730ES Series: 11 kg / 24.3 lbs RB6-920ES Series: 21.2 kg / 46.7 lbs RB20-1900ES Series: 76 kg / 167.5 lbs
Arm Reach	RB5-850E Series: 927.7 mm / 36.5 in RB3-1200E Series: 1200 mm / 47.2 in RB10-1300E Series: 1300 mm / 51.1 in RB16-900E Series: 900 mm / 35.4 in RB3-730ES Series: 730 mm / 28.7 in RB6-920ES Series: 920 mm / 36.2 in RB20-1900ES Series: 1900 mm / 74.8 in
Degree of freedom	6 axis
Joint Range	± 360° (Elbow: ± 165°)
Velocity	Joint: Max 180°/s, Tool: 1m/s
Repeatability	±0.05 mm
Foot print	RB5-850E/RB3-1200E/RB6-920ES Series: Φ173 mm RB10-1300E/RB16-900E Series: Φ196 mm RB3-730ES Series: Φ128 mm RB20-1900ES Series: Φ245 mm
Tool (End Effector) Connector Type	NE, E Version: M10-12Pin X 1EA U Version: M8-8Pin X 2EA (depend on option)
Tool I/O Ports	NE Version: Digital In 2, Digital Out 2, Analog In 2 E, U Version: Digital In 6, Digital Out 2
Tool Comm.	RS485
Tool Output Vol.	12V/24V, 2A
IP Rate	IP66
Temperature / Noise	0 ~ 50 ˚C / <65dB
Material	Aluminum, Steel.
Cable Length	Power cable, Robot Arm-Control Box connection cable: 5m Estop/Jog Interface cable: less 3m

Stand-type Control Box

	Specification
Weight	CB04, CB05: 17 kg / 37.5 lbs CB06, CB06-1: 14.9kg / 32.8 lbs
Size (W x H x D)	CB04, CB05: 454 x 240 x 416.2 mm CB06, CB06-1: 443 x 371 x 260 mm
I/O Ports	Digital Input 16, Digital Output 16 Analog Input 4, Analog Output 4
Communication	Ethernet, TCP/IP, Modbus TCP, Ethernet IP, ProfiNet, Siemens S7, OMRON FINS, Mitsubishi MC, LS XGT
Power	100 ~ 240 VAC, 50 ~ 60 Hz
Material	EGI(Electrically galvanized steel sheet)

Compact Control Box

	Specification
Weight	CB07: 8.3 kg / 18.3 lbs
Size (W x H x D)	457 x 173.5 x 232 mm
I/O Ports	Digital Input 16, Digital Output 16 Analog Input 4, Analog Output 4
Communication	Ethernet, TCP/IP, Modbus TCP, Ethernet IP, ProfiNet, Siemens S7, OMRON FINS, Mitsubishi MC, LS XGT
Power	100 ~ 240 VAC, 50 ~ 60 Hz
Material	SUS 304

DC Control Box

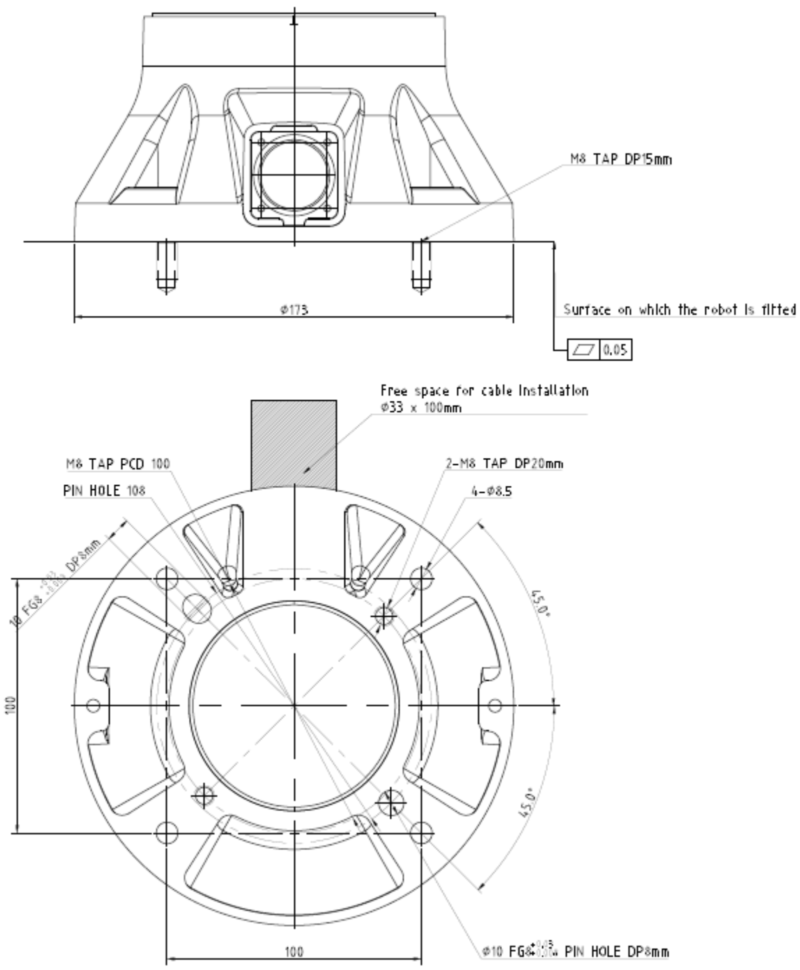
	Specification
Weight	CB08: 8.2 kg / 18.1 lbs
Size (W x H x D)	470 x 173.5 x 232 mm
I/O Ports	Digital Input 16, Digital Output 16 Analog Input 4, Analog Output 4
Communication	Ethernet, TCP/IP, Modbus TCP, Ethernet IP, ProfiNet, Siemens S7, OMRON FINS, Mitsubishi MC, LS XGT
Power	48 VDC
Material	SUS 304

APPENDIX B. FOOT PRINT SCHEMATIC

- RB5-850E / RB3-1200E / RB6-920ES Series

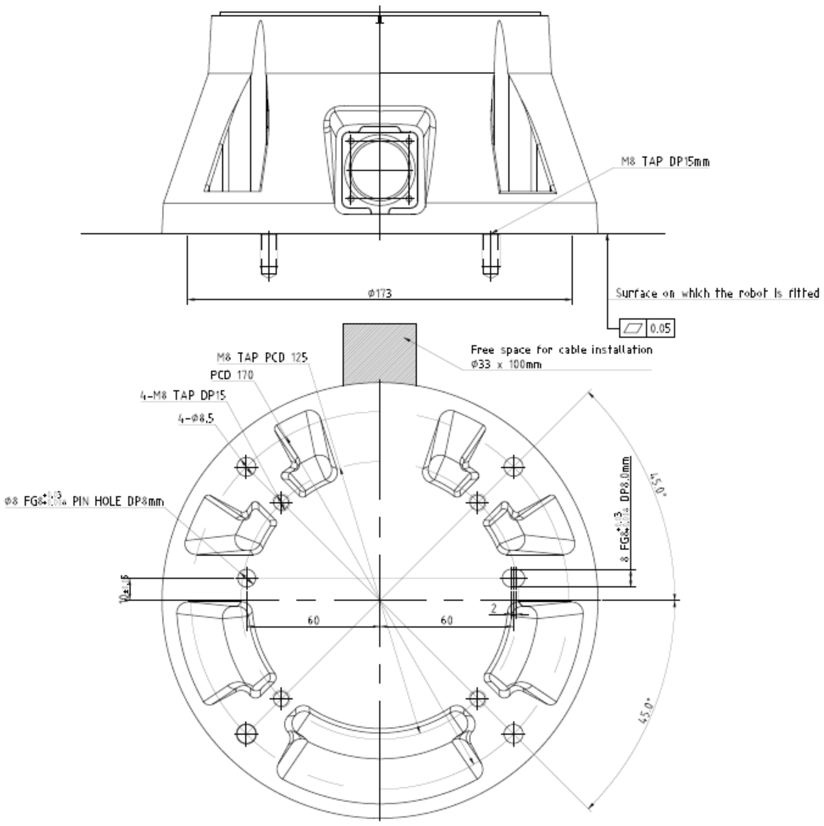
  - P.C.D: Pitch Circle Diameter

  - DP: Depth



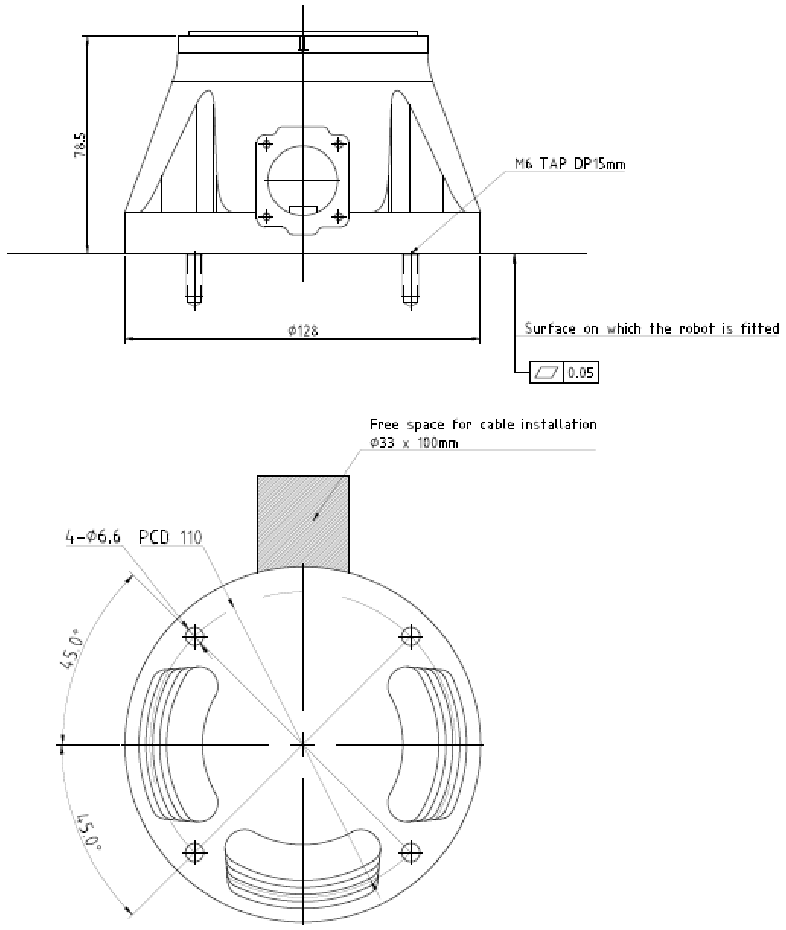
■ RB10-1300E / RB16-900E Series

- P.C.D: Pitch Circle Diameter
- DP: Depth.

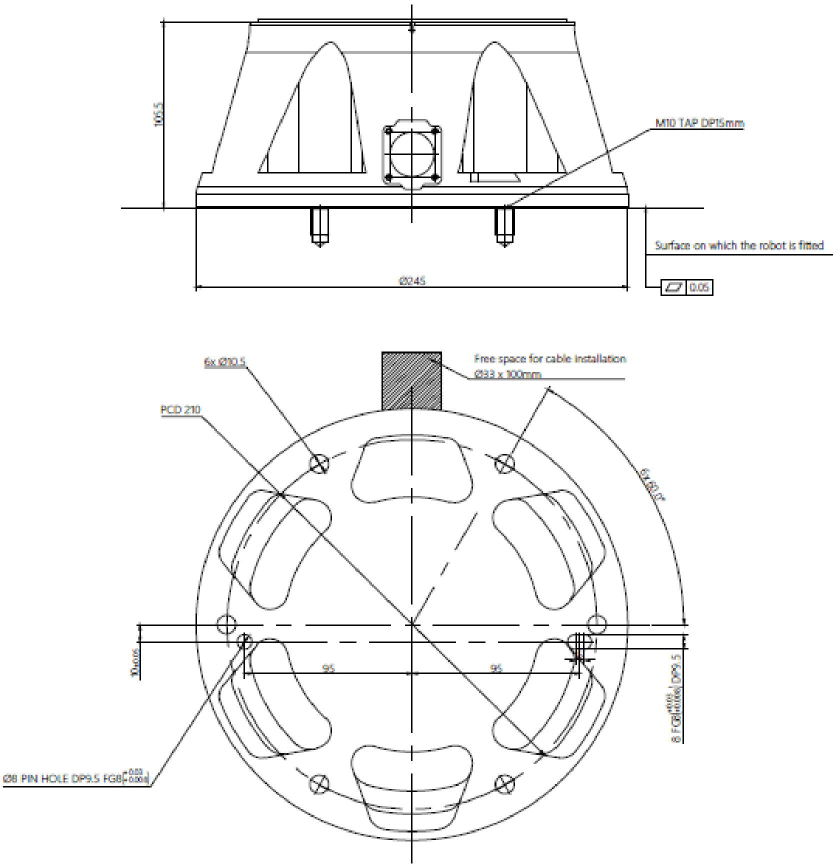


■ RB3-730ES Series

- P.C.D: Pitch Circle Diameter
- DP: Depth.



- RB20-1900ES Series
  - P.C.D: Pitch Circle Diameter
  - DP: Depth.

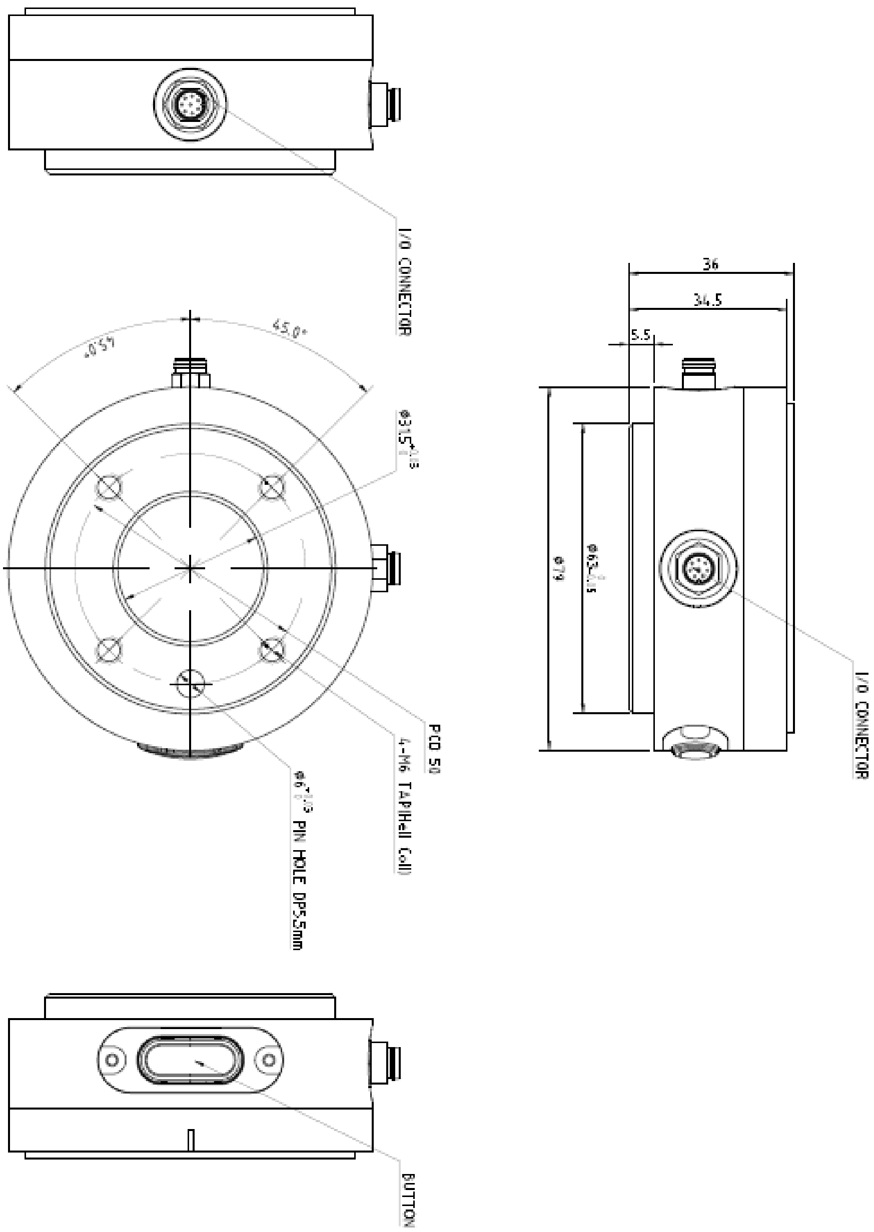


APPENDIX C. TOOL FLANGE SCHEMATIC

- RB5-850E / RB3-1200E / RB6-920ES Series

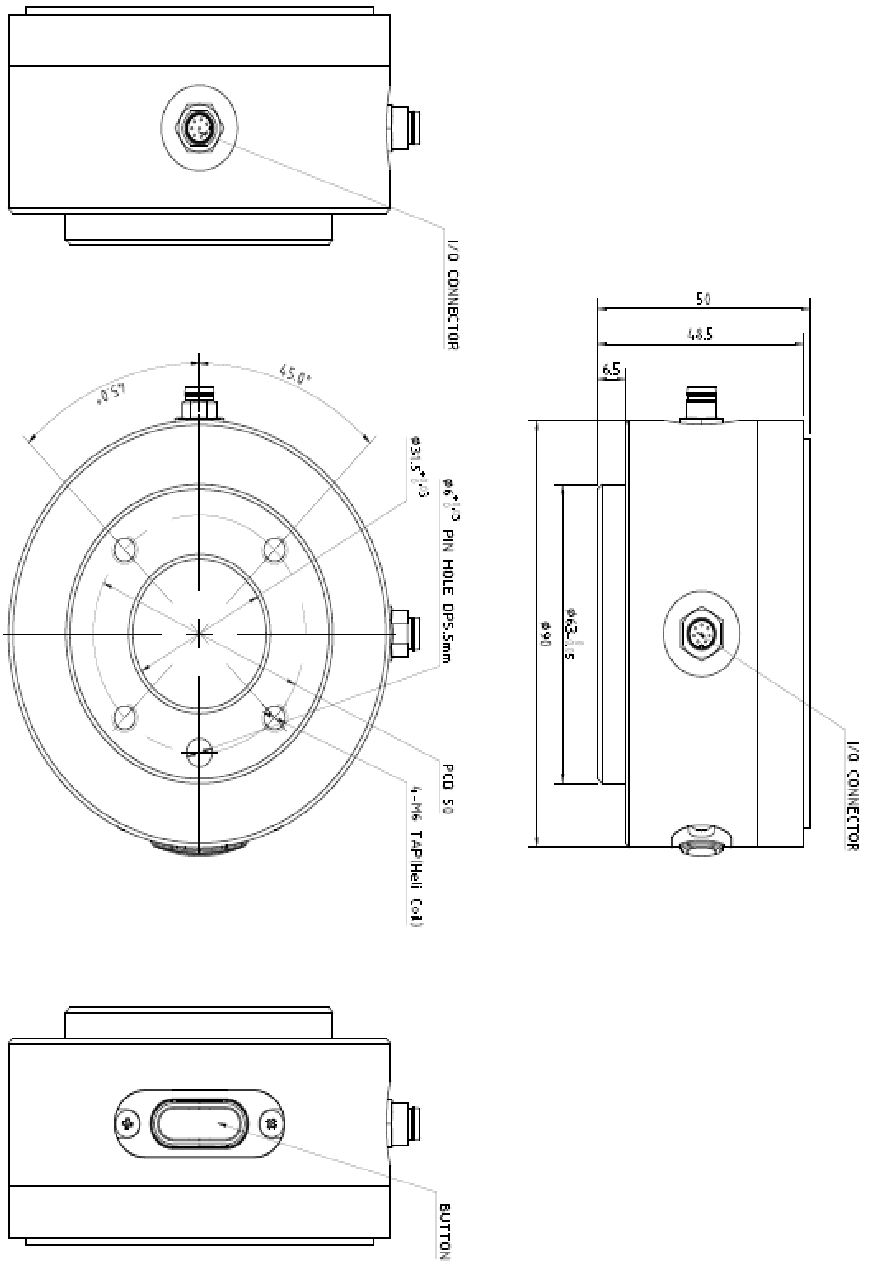
  - P.C.D: Pitch Circle Diameter

  - DP: Depth.



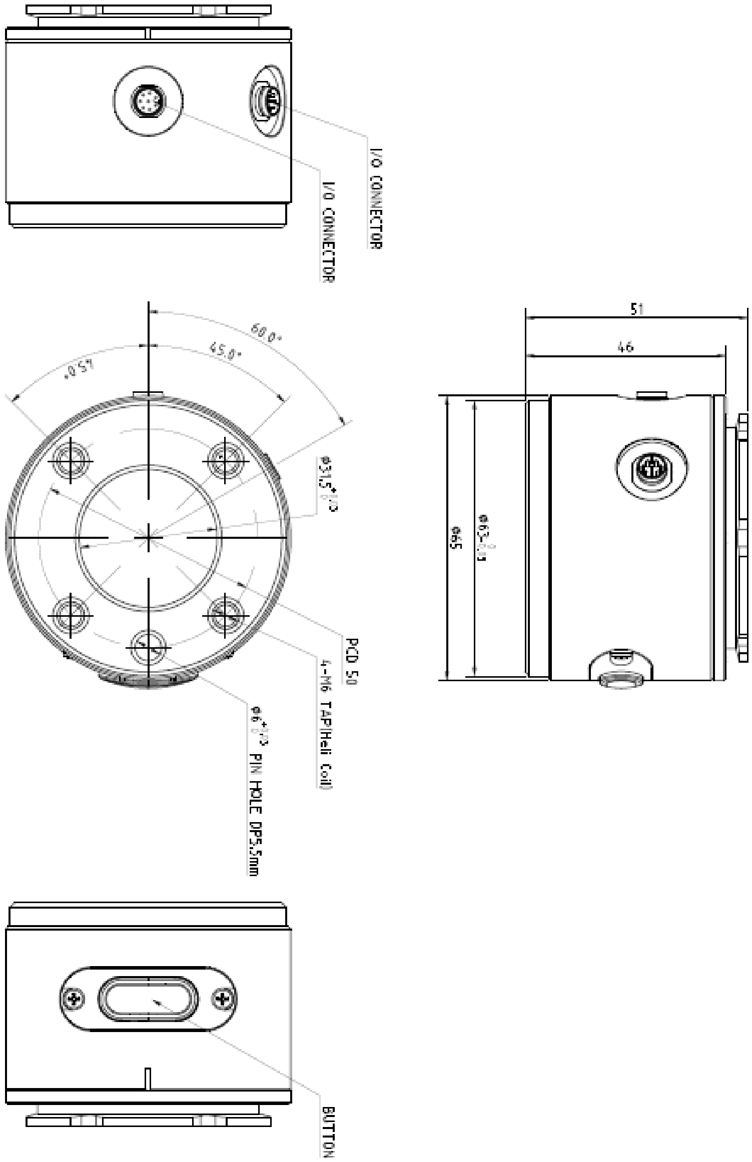
■ RB10-1300E / RB16-900E Series

- P.C.D: Pitch Circle Diameter
- DP: Depth.



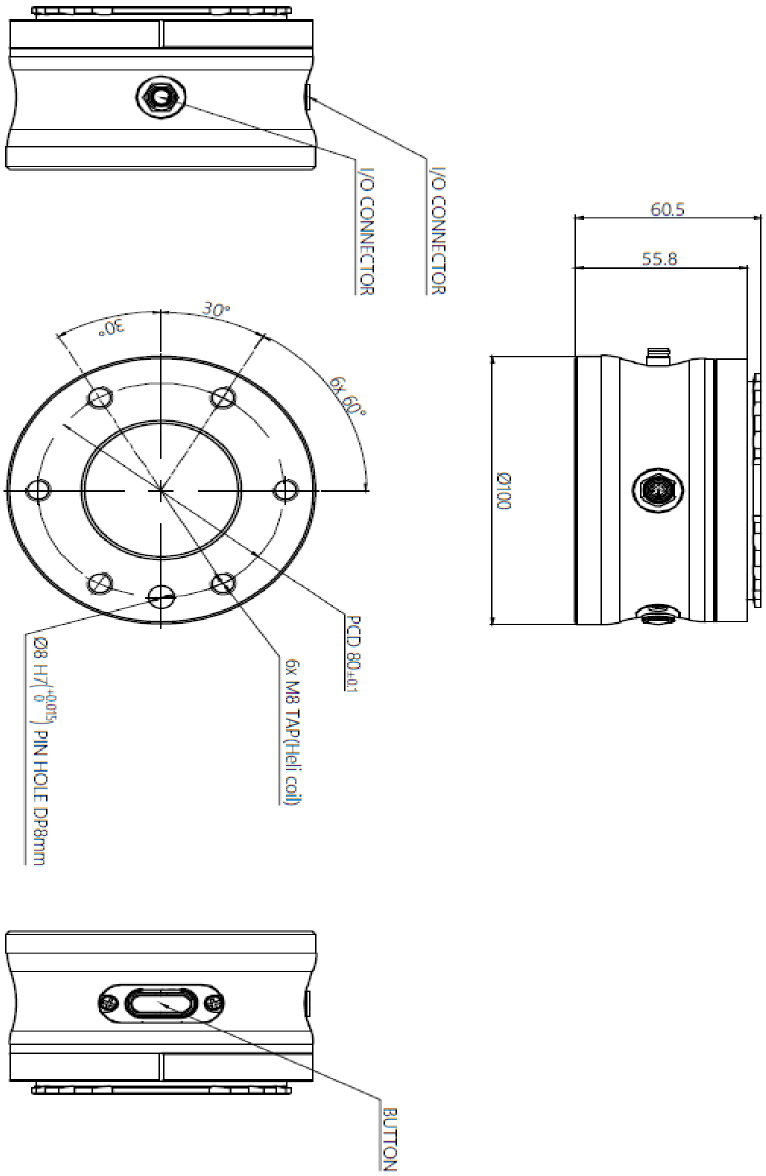
■ RB3-730ES Series

- P.C.D: Pitch Circle Diameter
- DP: Depth.



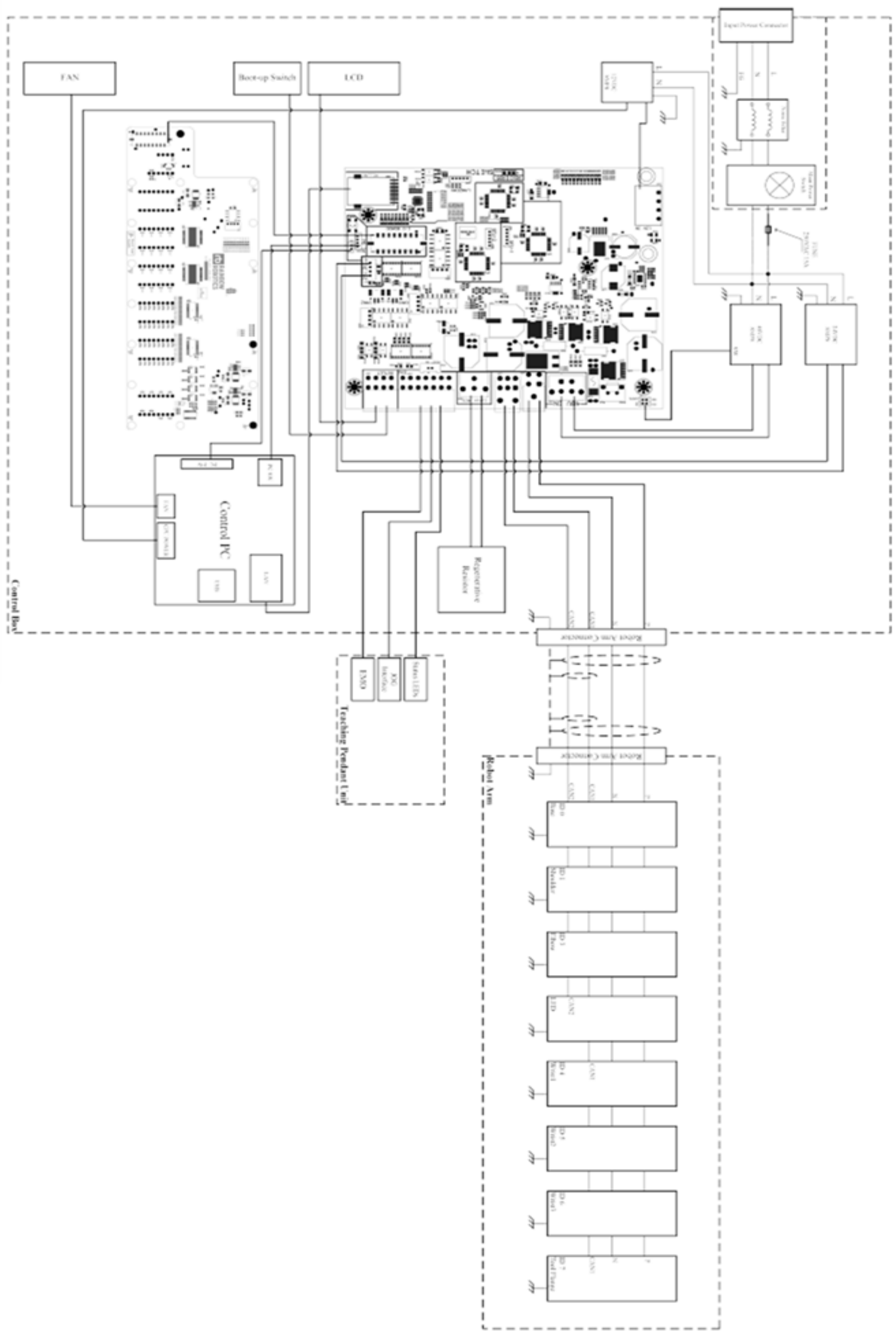
■ RB20-1900ES Series

- P.C.D: Pitch Circle Diameter
- DP: Depth.

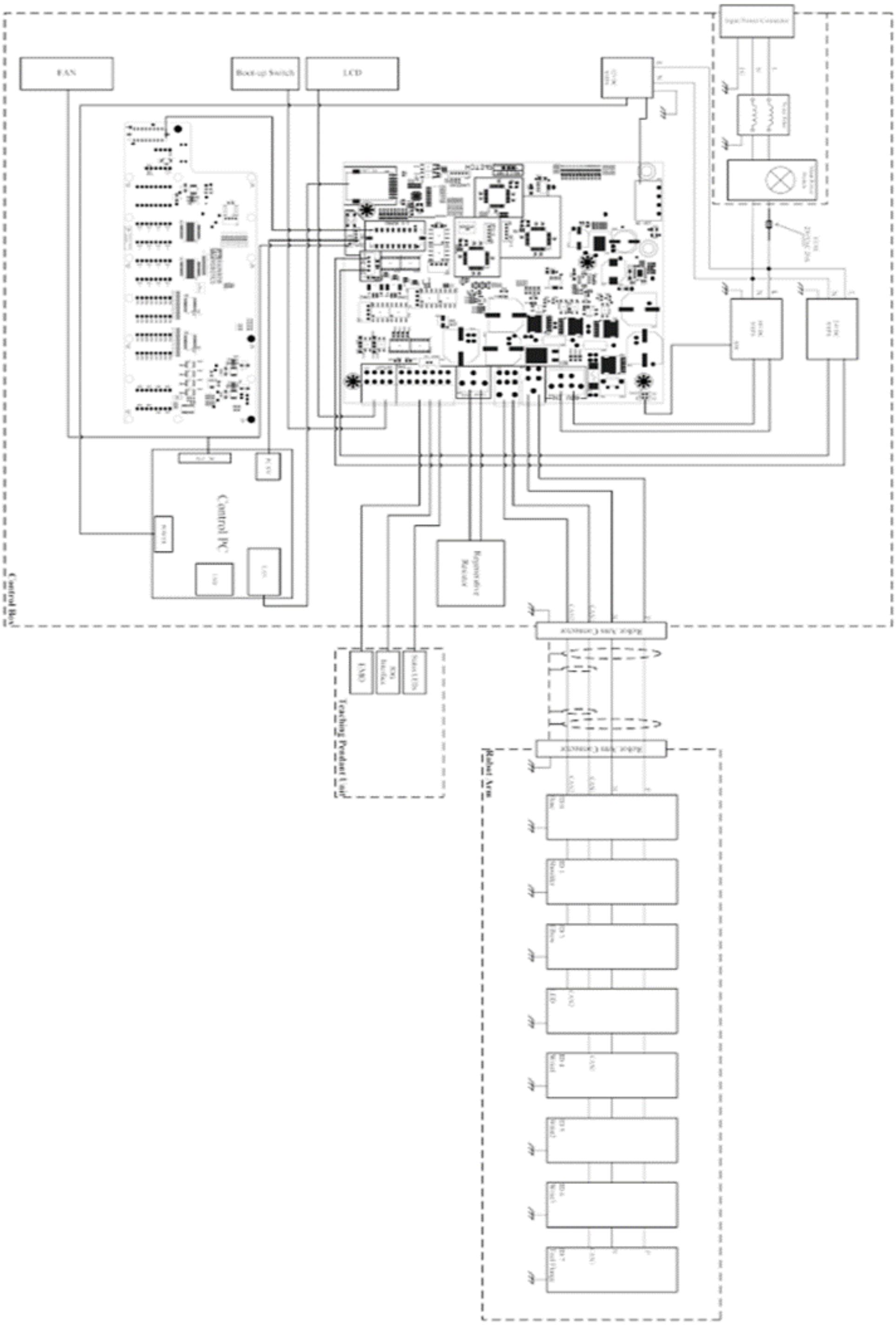


APPENDIX D. CONTROL BOX ELECTRICAL SCHEMATIC

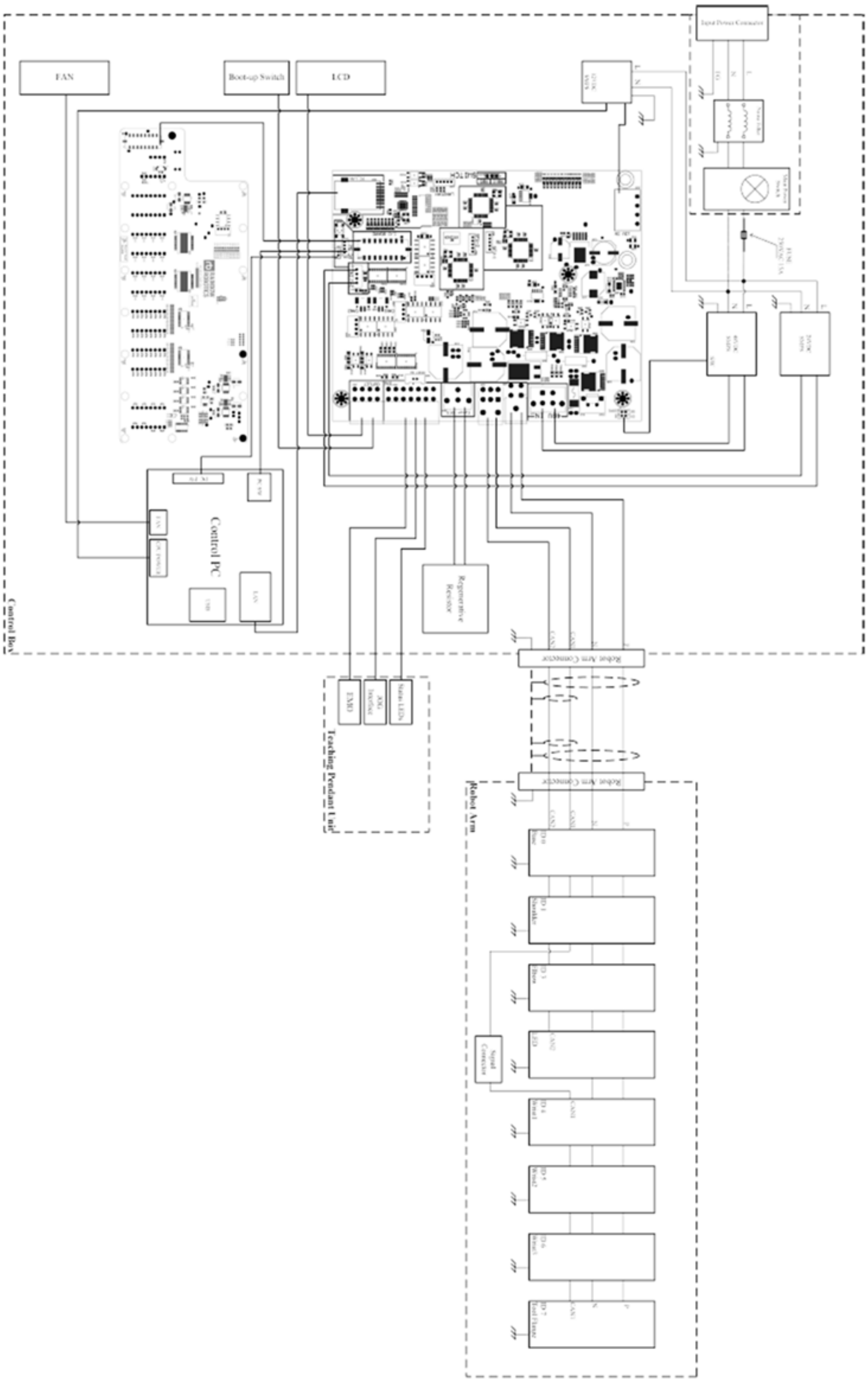
- Stand Control Box(CB06) Electrical Schematic



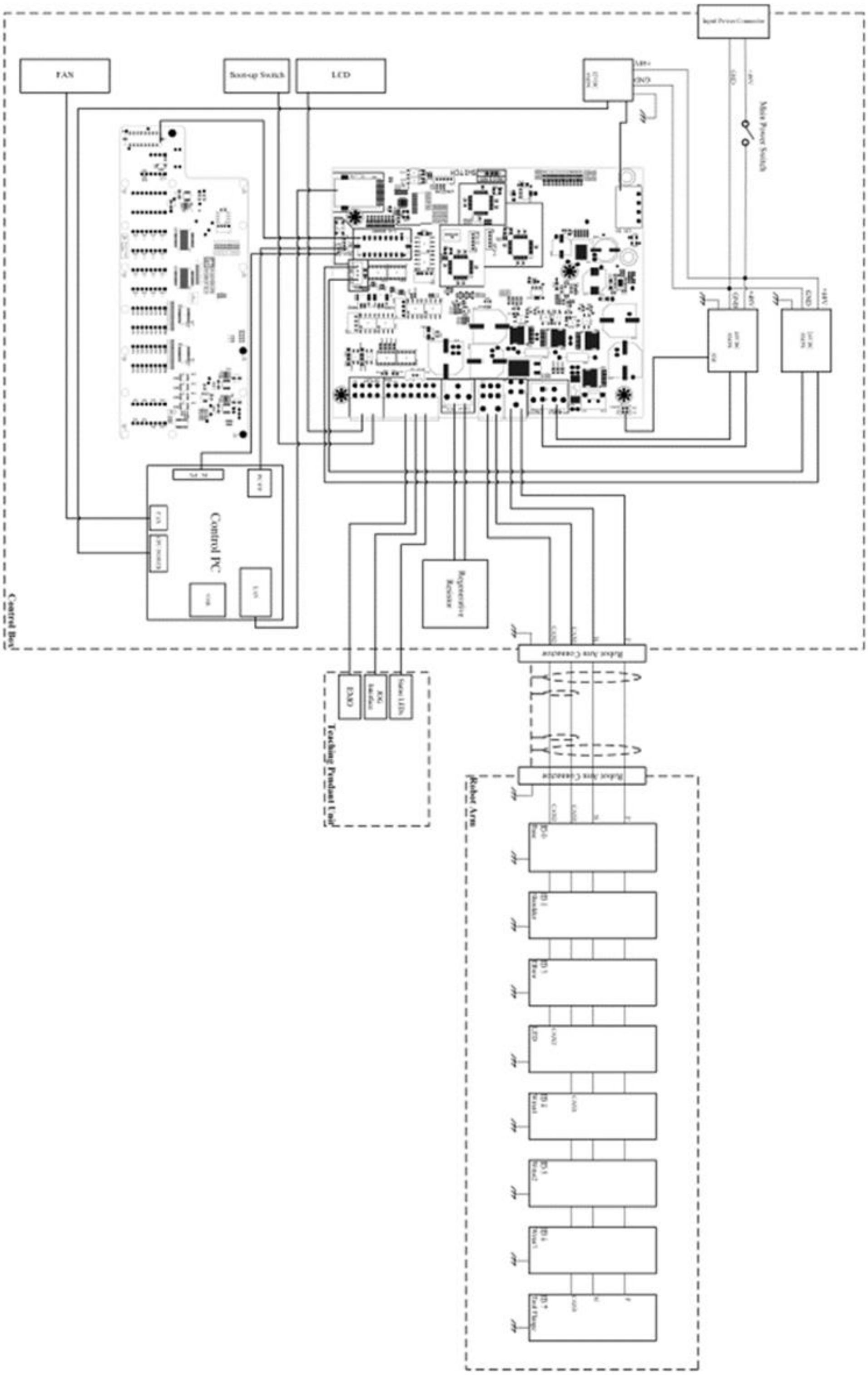
■ Stand-type Control Box(CB06-1) Electrical Schematic



■ Compact Control Box(CB07) Electrical Schematic



■ DC Control Box(CB08) Electrical Schematic

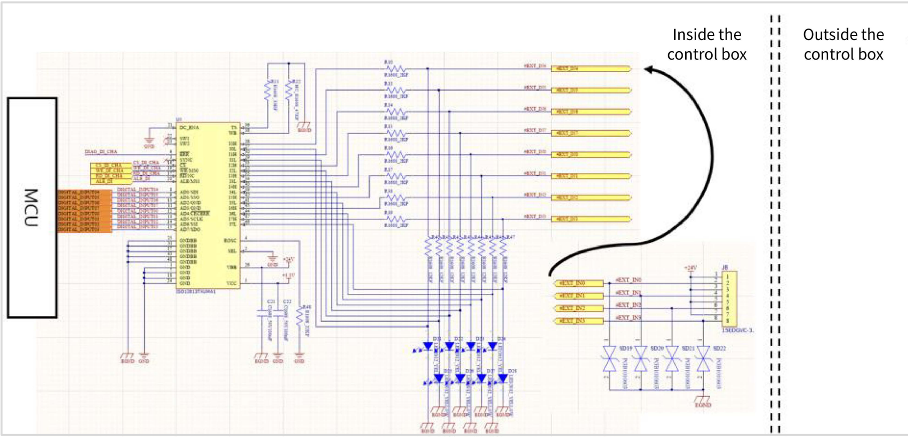


APPENDIX D-1. CONTROL BOX DIGITAL INPUT

- Warning

  Before connecting Control Box Digital input port, the power should be cut off.

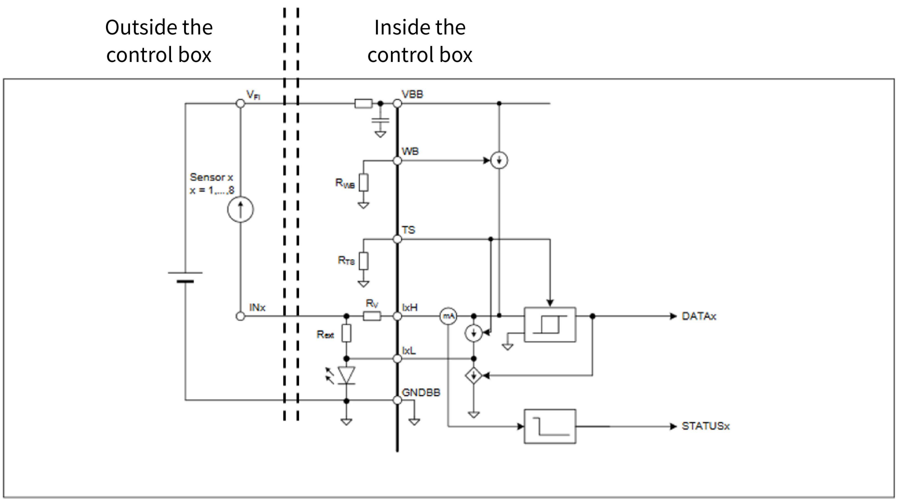
1. Internal Circuit Diagram of Digital Input [DI00 ~ DI15]



Device configuration that receives Control Box Digital input (DI00-DI15).

There is an internal 24V supply terminal. A malfunction will occur if an external 24V is supplied.

2. How to use digital input elements [DI00 ~ DI15]



How to use RB Control Box Digital input device (DI00-DI15).

Functional Description

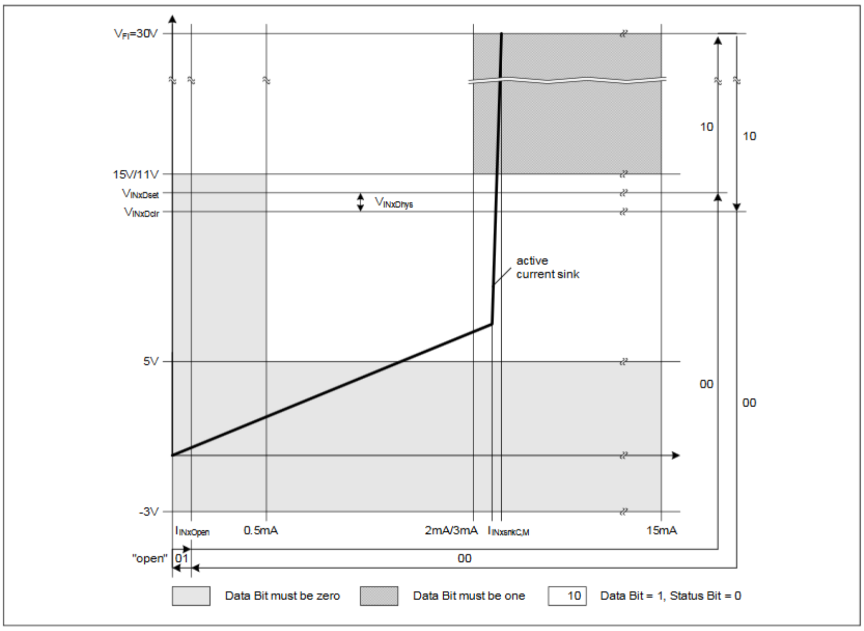


Figure 11 Sensor Input Characteristics

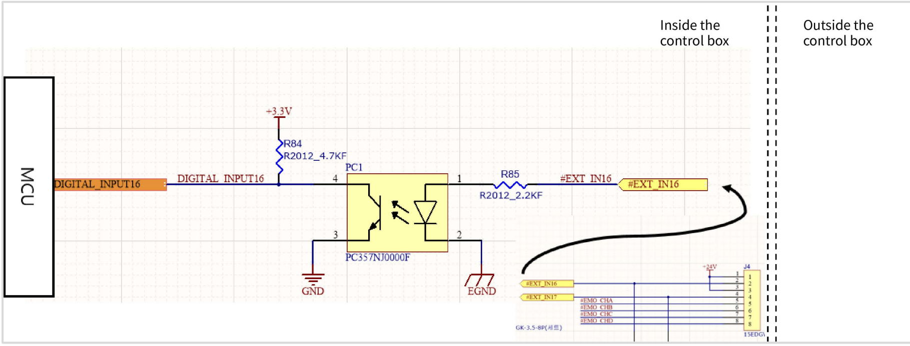
Voltage / current characteristic curve of digital input signal.

3. Digital input characteristics [DI00 ~ DI15]

Terminals	Parameter	Min	Typ	Max	Unit
[DI00 – DI15]	Voltage	-3	-	30	V
[DI00 – DI15]	OFF region	-3	-	5	V
[DI00 – DI15]	ON region	11	-	30	V
[DI00 – DI15]	Current(11-30V)	2	-	15	mA
[DI00 – DI15]	Function	-	PNP+	-	Type
[DI00 – DI15]	IEC 61131-2	-	1	-	Type

<u>This specification applies only to digital input 0 to digital input 15.</u>

4. Internal Circuit Diagram of Digital Input [DI16 - DI17]



Device configuration that receives Control Box Digital input (DI16-DI17).

There is an internal 24V supply terminal. A malfunction occurs when an external 24V is supplied.

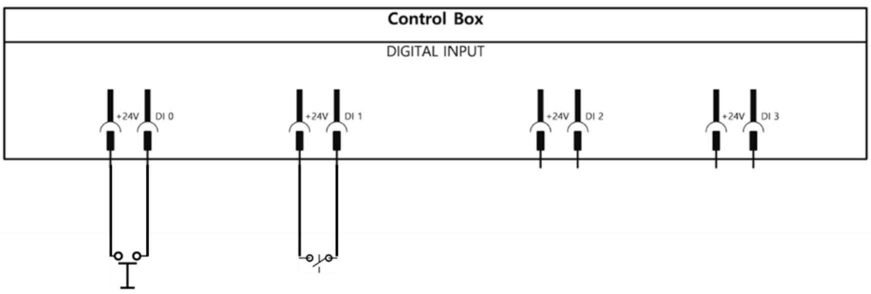
5. Digital input characteristics [DI16 - DI17]

Terminals	Parameter	Min	Typ	Max	Unit
[DI16 – DI17]	Voltage	0	-	25	V
[DI16 – DI17]	OFF region	0	-	7	V
[DI16 – DI17]	ON region	7	-	25	V
[DI16 – DI17]	Function	-	PNP+	-	Type

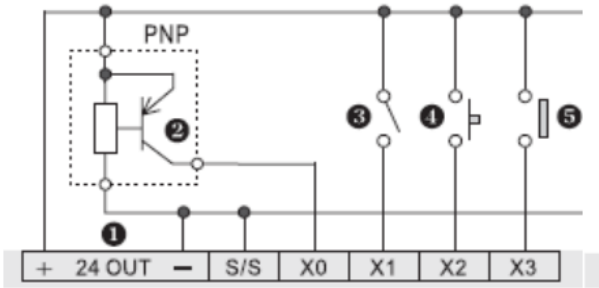
This applies only to digital inputs 16 and 17.

6. Testing environment

Digital input device test was conducted using Toggle switch, and the following configuration was tested.



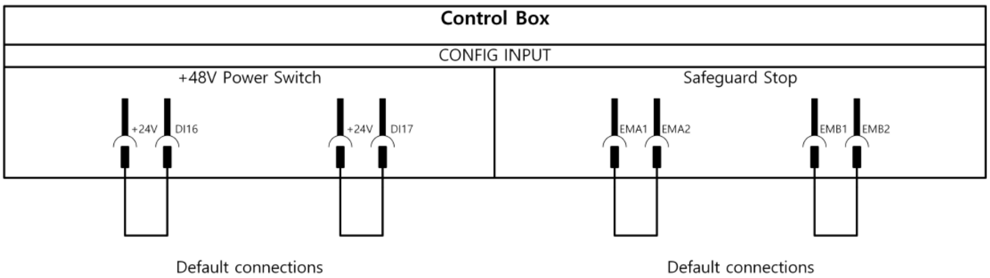
7. How to use PNP sensor



PNP sensor can be used in the same way as above.

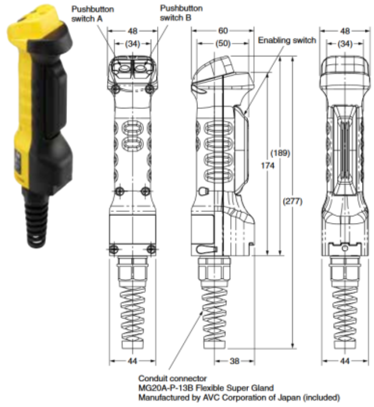
This is a specification that applies to all digital inputs.

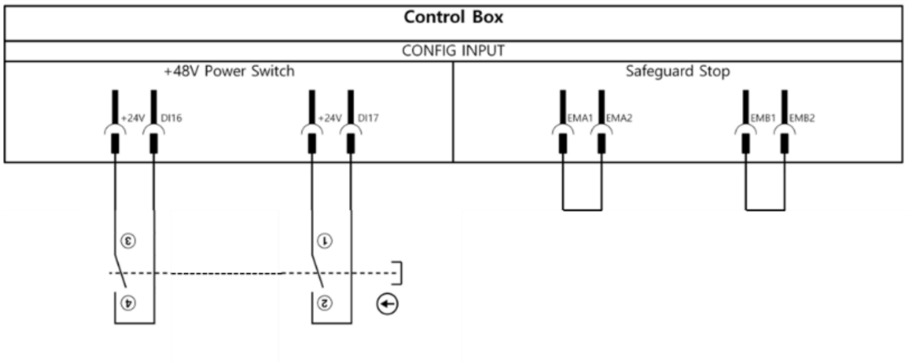
8. How to connect 3-Position Enabling Device

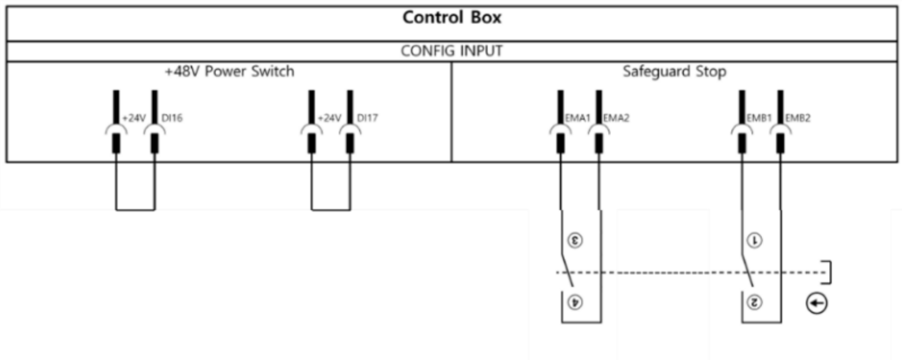


The initial factory condition is as above, and it is possible to install the operation.

Source: https://www.motionsolutions.com







This applies to Enabling Device in accordance with ISO 10218, IEC 60204-1.

9. How to connect safety equipment

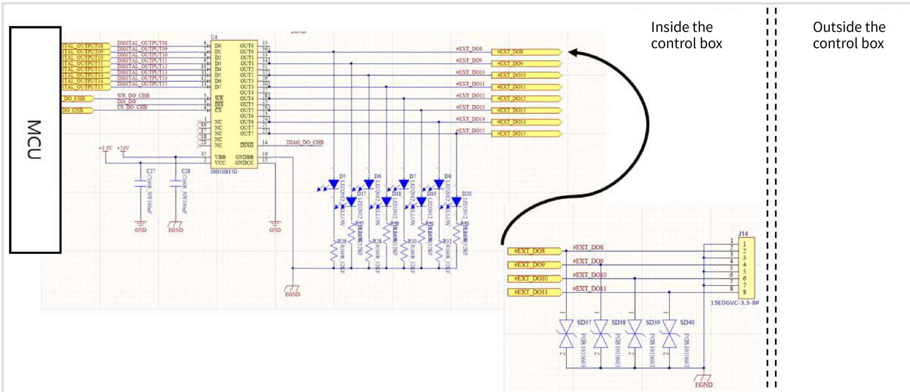
Safety device wiring using PNP type sensor and Enabling Device such as light curtain and safety door sensor is same as above.

APPENDIX D-2. CONTROL BOX DIGITAL OUTPUT

- **Warning**

  Before connecting the Control Box Digital output port, the power should be turned off.

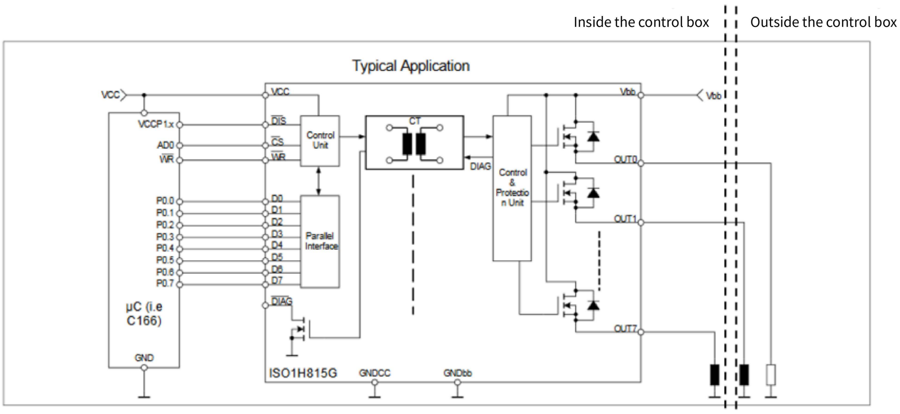
1. Digital output internal circuit diagram [DO00 ~ DO15]



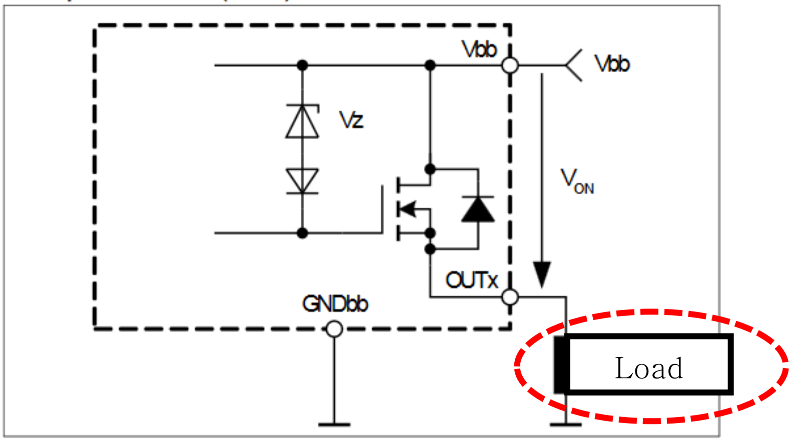
Device configuration that performs Control Box Digital output (DO00-DO15).

There is internal GND terminal, and it should be connected to GND of external sensor and equipment to be connected.

2. Digital output device usage [DO00 ~ DO15]



How to use RB Control Box Digital Output Device (DO00-DO15).



How to use a single digital output.

Vbb power is supplied inside of the control box and its output is the source.

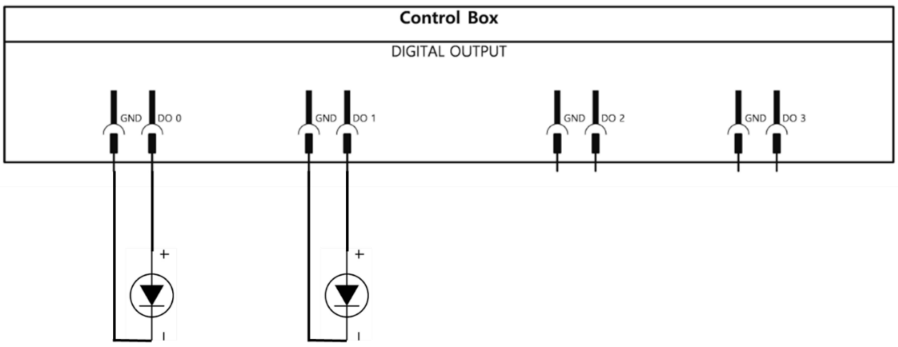
3. Digital output characteristics [DO00 ~ DO15]

Terminals	Parameter	Min	Typ	Max	Unit
[DO00 – DO15]	Voltage	-	24	-	V
[DO00 – DO15]	CURRENT	0	-	1	A
[DO00 – DO15]	Function	-	PNP	-	Type

<u>Single channel 1A is possible, but the total current of all channels must be less than 2A</u>

4. Test environment

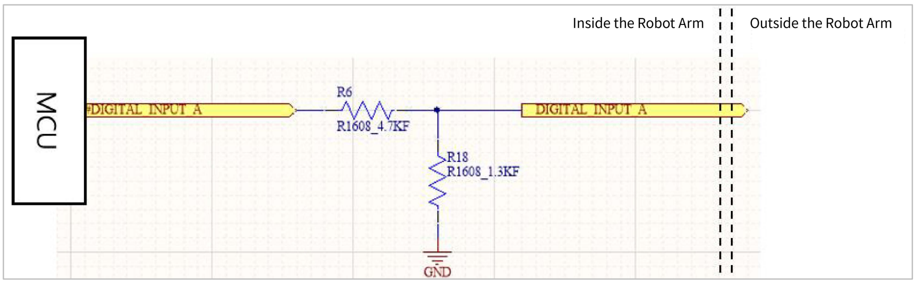
Digital output device test was conducted using 24Vdc LED and the following configuration was tested.



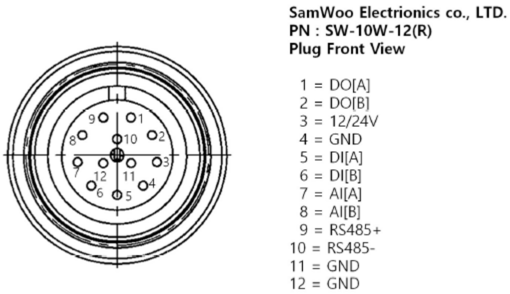
APPENDIX D-3. TOOL FLANGE DIGITAL INPUT

- Warning
  Before connecting RB Tool Flange I / O input port, the power should be cut off.
- The electrical drawing below is for Non-E type only.

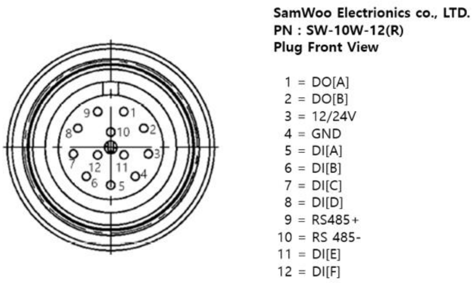
1. Digital input internal circuit diagram



Device configuration for Tool Flange Digital input.



(1) Non-E Version Robot



(2) E Version Robot

Exposed connector wiring diagram. The wiring diagram will be divided as shown above according to the robot version.

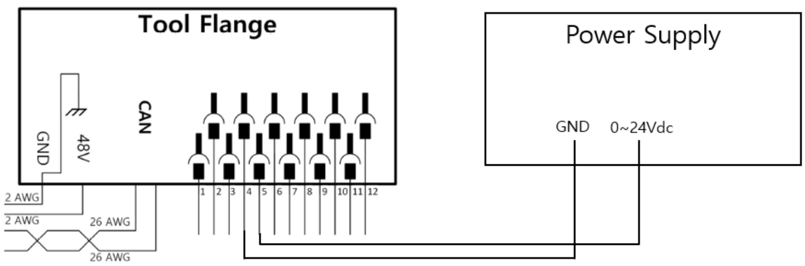
2. Digital input characteristics

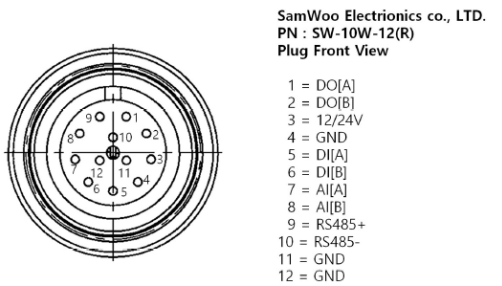
Terminals	Parameter	Min	Typ	Max	Unit
[DIA, ... , DIF]	Voltage	0	-	24	V
[DIA, ... , DIF]	OFF region	0	-	9	V
[DIA, ... , DIF]	ON region	10	-	24	V

<u>This is a specification that applies only to Tool Flange Digital input (At this time, only DIA and DIB for Non-E version Robot are applied.)</u>

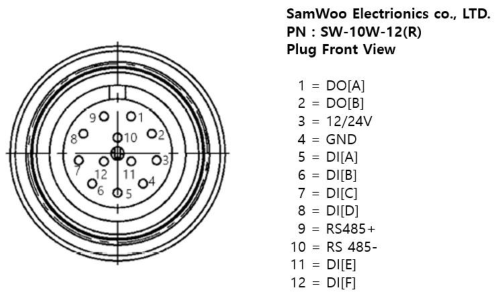
3. Test environment

Digital input device test was conducted using power supply, and the following configuration was tested.





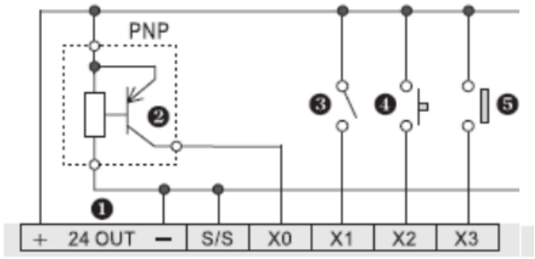
Non-E Version



E Version

4. How to use PNP sensor

Ex source: https://blog.naver.com/mjg5080/97380010



PNP sensor can be used in the same way as the above connection.

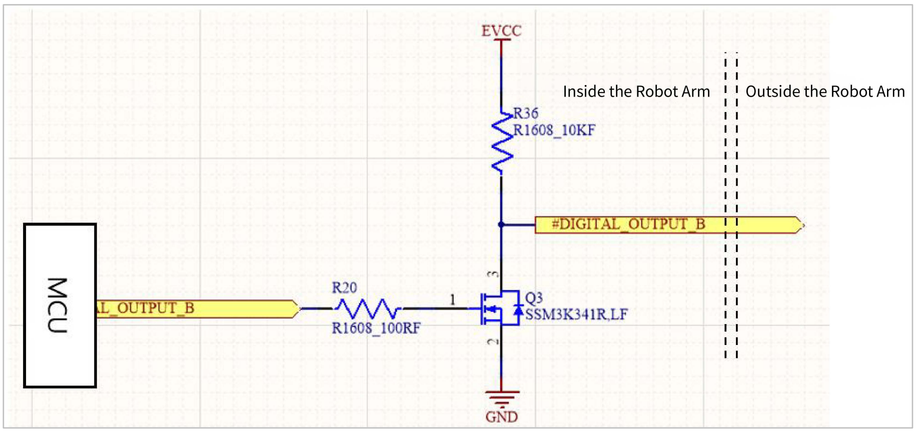
This applies equally to the Control Box Digital input.

APPENDIX D-4. TOOL FLANGE DIGITAL OUTPUT

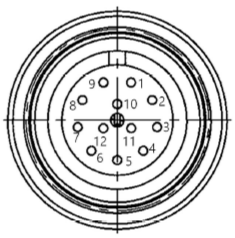
- Warning
  Before connecting the Tool Flange I/O output port, the power should be cut off.

- The electrical drawing below is for Non-E type only.

1. Digital output internal circuit diagram

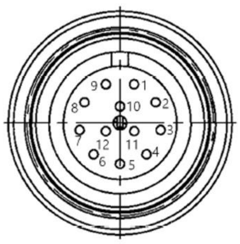


Device composition for Tool Flange Digital output [DOA, DOB].



SamWoo Electrionics co., LTD.
PN : SW-10W-12(R)
Plug Front View

1 = DO[A]
2 = DO[B]
3 = 12/24V
4 = GND
5 = DI[A]
6 = DI[B]
7 = AI[A]
8 = AI[B]
9 = RS485+
10 = RS485-
11 = GND
12 = GND



SamWoo Electrionics co., LTD.
PN : SW-10W-12(R)
Plug Front View

1 = DO[A]
2 = DO[B]
3 = 12/24V
4 = GND
5 = DI[A]
6 = DI[B]
7 = DI[C]
8 = DI[D]
9 = RS485+
10 = RS 485-
11 = DI[E]
12 = DI[F]

(1) Non-E Version Robot

(2) E Version Robot

External connector wiring diagram. The wiring diagram will be divided as shown above according to the robot version.

2. Digital output characteristics

Terminals	Parameter	Min	Typ	Max	Unit
[DOA, DOB]	Voltage	0	12/24	24	V
[DOA, DOB]	CURRENT Ver 1.	0	150	700*	mA
[DOA, DOB]	CURRENT Ver 2.	0	2000	2000	mA

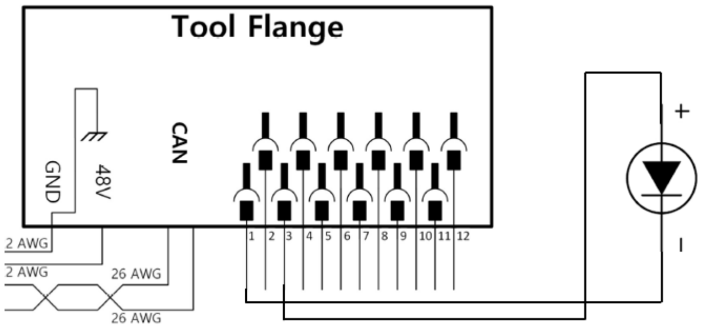
*Tsp=25°C; pulsed; tp≤10μs

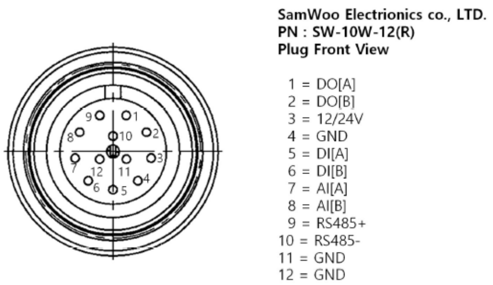
<u>This specification applies only to Tool Flange Digital outputs A and B.</u>

<u>As of July 24, 2019, version of RB5 shipped out is Current Ver 1.</u>

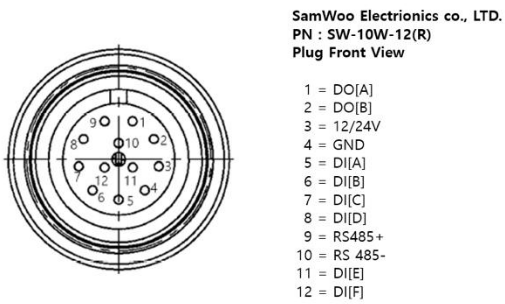
3. Test Environment

Digital output device test was conducted using 24V dc LED and the following configuration was tested.





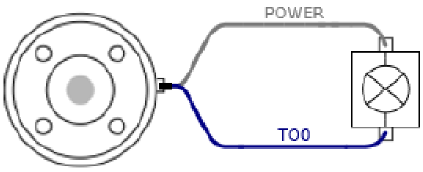
(1) Non-E Version



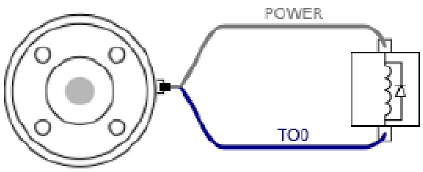
(2) E Version

The following example is shown in this manual.

- The image shown below illustrates how to turn on/off a load with 12V or 24V. The voltage level can be specified in the Tool Out block.



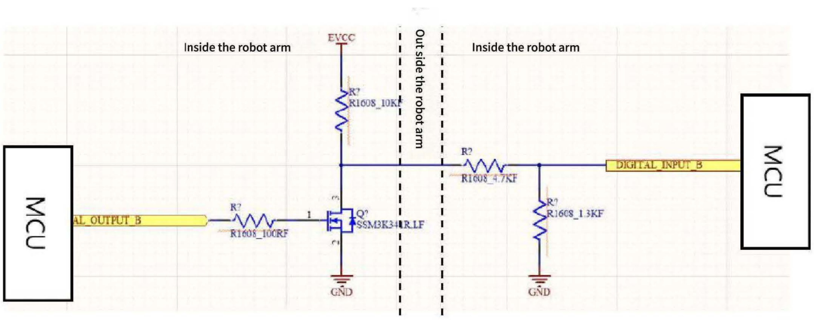
※ It is strongly recommended to use a diode to protect the tool using an inductive load.



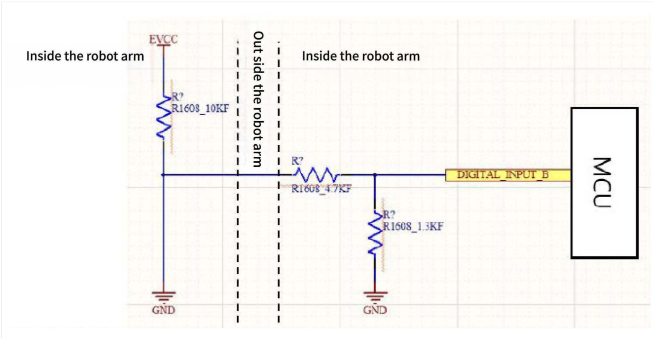
4. Precautions when using

Digital output device is NPN type but has internal 10K pullup resistor.

Most devices (LEDs, solenoid valves, relays) can be used in the test environment No. 3 or with the digital signal application function on commercial grippers. However, they may not work in the environment using the same voltage distribution as Rainbow Robotics' tool flange digital input devices.

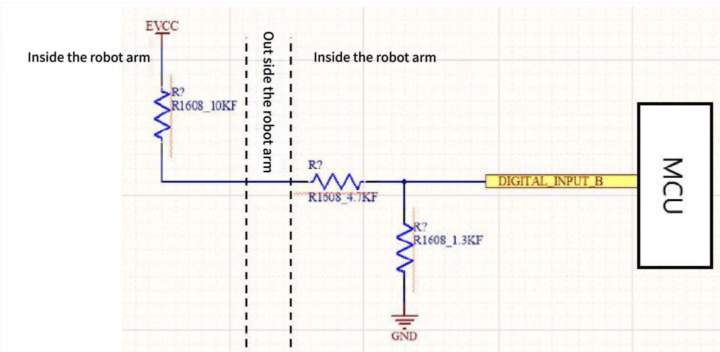


<u>When Rainbow Robotics' digital output is connected to the digital input</u>



Low Digital output

The digital input is input with 0V and output to Low.



Digital output High

Digital input may not be recognized depending on the resistance value.

For the diagram above, if the voltage applied to MCU is EVCC 24Vdc, about 2V is applied to it and is detected as Low.

If users MUST operate as above diagram, digital input stage resistance ratio adjustment is necessary.

APPENDIX E. EXTERNAL SCRIPT CONTROL API

E.0 Concept

The cooperative robot RB series can be operated for various environments and purposes. It can be used in conjunction with multiple RB series or other systems. In conjunction with the vision system, movement coordinates can be changed in real time or used as part of a user's existing system.

Users can control the robot with teaching pendant (tablet UI), but it provides a way to control the robot from any external controller for user's convenience or operation.

The RB series receives script commands by default and executes those commands. The task of writing a motion using the teaching pendant (tablet UI) and executing the script of the file in order is a general operation method. The following method described in this document is an alternative method of receiving a command script from another external device to control a robot of the RB series.

The control syntax provided in the teaching pendant/tablet UI can be implemented by the user directly from the external control device, and the robot operation commands/IO control commands are sent according to the user's use case.

The following document describes an example of driving a robot with the above concepts.

E.1 External Control Script API

The description of the scripts provided in this document looks similar to the scripts in the ".wsl" work document, which is written using a tablet as a dedicated script for external control. Work documents contain statements that control flows such as "repeat", "if-else", and "break", so that the completion of a statement is not directly related to the action, and the parent sentence of that statement must be completed.

For example, suppose there are Point Functions in the Move command Function.

```
1)
move joint {
   point ( ) absolute 0.4, 0.1, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
      point ( ) absolute 0.4, 0.1, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0
}
```

```
2)
move joint {
   point ( ) absolute 0.4, 0.1, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
      point ( ) absolute 0.4, 0.1, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0
```

The difference between 1 and 2 is the presence or absence of "}" at the end. In both cases, the point statement is complete. However, unlike 1, 2 is a syntax that cannot operate because the move statement, which is the parent of point, is not completed, and the parser will wait for the statement to complete.

```
3)
folder( ) {
      move joint {
      point ( ) absolute 0.4, 0.1, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
            point ( ) absolute 0.4, 0.1, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0
      }
```

In the same logic as above, the parser does not run because it waits for the folder statement to complete.

However, the above method is not suitable for external control method. The user expects the robot to operate by parsing the command the moment it sends it through external control. It does not send multiple commands and complete those lines of text like example 3.

So external control must be organized so that each command is sent separately as a string. External control does cannot access any features that control the flow. Commands such as "repeat", "if-else", "break" or "wait" in the work document will not be available externally and must be replaced by the same structure and logic within the external control.

First, the script I will explain is initialization, termination, and operation mode change.

1) mc

Script	mc jall init
Descript.	This command starts initialization process.
Example	"mc jall init"

2) shutdown

Script	shutdown
Descript.	This command terminates the robot operation and turns off the power.
Example	"shutdown"

3) pgmode

Script	pgmode mode_type
Descript.	This command changes the mode between real and simulation modes. The operation mode is selected through mode_type. There are two operation modes: 'real' and 'simulation'. In 'real', the robot physically moves when a motion command is issued. In 'simulation', when a motion command is issued, only the internal reference is changed, and the command does not move the robot. When the robot is first initialized, the default mode is 'simulation' mode.
Example	"pgmode real" "pgmode simulation"

The last command explained is the task script.

task

Script	task load work_file_name
Descript.	This command loads a work file previously programmed. The work_file_name uses the name of the '.wsl' file. At this time, the '.wsl' extension should be omitted, and only the relative path and name of the file should be entered. This file is not one that exists on the pendant, but rather a file that has been loaded or saved to the control panel at least once through the pendant. Therefore, it is possible to load the file even if the pendant is not connected.
Example	“task load test_file”

Script	task play option
Descript.	This command runs the work file loaded. The input value for option is blank or “once”. When option leaves empty, it runs the work file repeatably until the number of repeatation is met. When “once” is set, it runs the work file once.
Example	“task play” “task play once”

Script	task repeat num
Descript.	This command sets the number of repeatation for the work file. The input value for num is the number of repeatation. The number should be an integer. -1 can be used to run the work file unlimitedly. The number of repeatation set by this command is maintained until power off. After rebooting the robot, this value is set by a number in the pendent.
Example	“task repeat 5” “task repeat -1”

Script	task pause
Descript.	This command pauses the robot operation that is currently in progress. It can be used for all work files that are running due to external control commands and the "task play" command. In a paused state, the operation can be resumed using the "task resume_a" command. When the robot is in a paused state, it is not considered as finished, so any other operation commands from external control will be ignored.
Example	"task pause"

Script	task stop
Descript.	This command stops the robot operation that is currently in progress. It can be used for all work files that are running due to external control commands and the "task play" command. Once the operation is stopped, it will not resume with "task resume_a." The operation is completely terminated. When the stop command is issued, the robot may abruptly halt if it was moving quickly. It is recommended to perform the "task stop" command after "task pause."
Example	"task stop"

Script	task resume_a
Descript.	This command resumes the operation of the robot that was paused by the "task pause" command or by an alarm or debug command in the work file.
Example	"task resume_a"

Script	task resume_b
Descript.	This command resumes the operation of the robot that was paused due to external collision detection.
Example	“task resume_b”

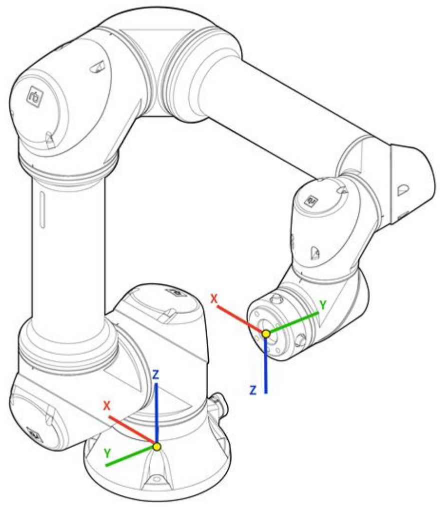
In order to use external control, the external computer must be connected to the control box. The connection uses TCP / IP communication and the corresponding IP address can be set in the pendant. The result is displayed on the screen on control panel. Ports 5000 and 5001 open for external control. Port 5000 is a port for receiving commands, and port 5001 is a port for requesting and sending data indicating robot status. For convenience, port 5000 is called the command port and port 5001 is called the data port.

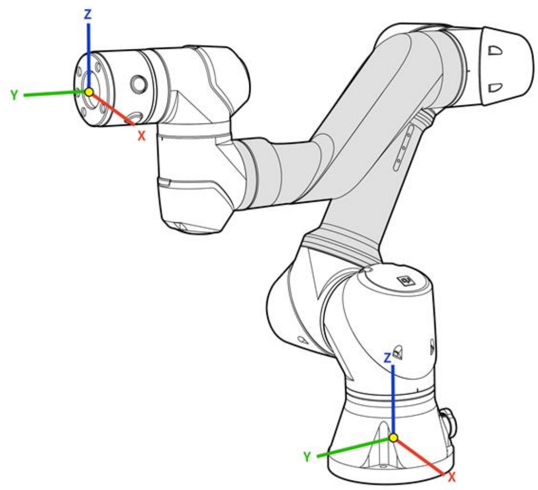
Users can send the script command described above to the command port. The command port has a filter for the first command, so if it does not start with a previously described script command like "mc" or "pgmode," it will respond with the message "The command is not allowed." When a valid command is properly formatted and passed to the parser, the response will be "The command was executed."

Additionally, for more diverse script commands, please refer to [Ui_Script]_6.10_ENG.

For the data port, sending the command "reqdata" will return information about the current robot status through the data port. Please refer to the [reqdata_Structure]_6.10_ENG manual for more details.

APPENDIX F. COORDINATE SYSTEM





- Global Coordinate

  Once the robot is fixed with the coordinate system fixed to the base of the robot, the global coordinate system is also fixed.

  The center of the base surface is the origin. Set the robot direction to the +Z direction from the origin and the connector direction to the +Y direction from the origin.

- Local Coordinate

  Coordinate system fixed to TCP (Tool Center Point) of the robot, the direction of the axis changes in real time by setting or moving the TCP offset.

  Set TCP as the origin and set the robot direction from the origin to the +Y direction and the teach button direction from the origin to the +Z direction.

APPENDIX G. STOPPING TIME/DISTANCE

In the RB Series of collaborative robots, the time and distance between the robots stop and the distance are generated by the safety monitoring function.

The graph below shows the stop time and stop distance for stop category 1 for Joint 0 (Base axis), Joint 1 (Shoulder axis), and Joint 2 (Elbow axis). The measurements were conducted by attaching the payload to each joint.

WARNING

Depending on the situation, the actual stop motion may differ from the results below. Joint 0 is the result of horizontal movement, and Joint 1 and 2 are the result of vertical downward movement. For the length of the arm, the maximum length is applied.

RB5-850E Series Base (Joint 0)		
	Stop Distance (mm)	Stop Time (sec)
Test 1	150.91	0.3
Test 2	149.68	0.28
Test 3	148.5	0.3
Maximum	150.91	0.3
Average	149.7	0.29
Condition	Max. Reach / Max. Velocity / Horizontal Motion	

RB5-850E Series Shoulder (Joint 1)		
	Stop Distance (mm)	Stop Time (sec)
Test 1	193.77	0.28
Test 2	220.46	0.32
Test 3	198.28	0.29
Maximum	220.46	0.32
Average	204.17	0.3
Condition	Max. Reach / Max. Velocity / Vertical Downward Motion	

RB5-850E Series Elbow (Joint 2)		
	Stop Distance (mm)	Stop Time (sec)
Test 1	168.52	0.31
Test 2	165.34	0.3
Test 3	141.02	0.23
Maximum	168.52	0.31
Average	158.29	0.28
Condition	Max. Reach / Max. Velocity / Vertical Downward Motion	

RB3-1200E Series Base (Joint 0)		
	Stop Distance (mm)	Stop Time (sec)
Test 1	253.3	0.33
Test 2	252.63	0.27
Test 3	249.27	0.24
Maximum	252.63	0.33
Average	251.73	0.28
Condition	Max. Reach / Max. Velocity / Horizontal Motion	

RB3-1200E Series Shoulder (Joint 1)		
	Stop Distance (mm)	Stop Time (sec)
Test 1	283.08	0.3
Test 2	285.88	0.29
Test 3	303.28	0.34
Maximum	303.28	0.34
Average	290.75	0.31
Condition	Max. Reach / Max. Velocity / Vertical Downward Motion	

RB3-1200E Series Elbow (Joint 2)		
	Stop Distance (mm)	Stop Time (sec)
Test 1	195.68	0.31
Test 2	200.94	0.31
Test 3	190.95	0.3
Maximum	200.94	0.31
Average	195.86	0.306
Condition	Max. Reach / Max. Velocity / Vertical Downward Motion	

RB10-1300E Series Base (Joint 0)		
	Stop Distance (mm)	Stop Time (sec)
Test 1	132.97	0.3
Test 2	136.53	0.32
Test 3	142.1	0.34
Maximum	142.1	0.34
Average	137.2	0.32
Condition	Max. Reach / Max. Velocity / Horizontal Motion	

RB10-1300E Series Shoulder (Joint 1)		
	Stop Distance (mm)	Stop Time (sec)
Test 1	260.51	0.36
Test 2	246.79	0.35
Test 3	252.54	0.35
Maximum	260.51	0.36
Average	253.28	0.353
Condition	Max. Reach / Max. Velocity / Vertical Downward Motion	

RB10-1300E Series Elbow (Joint 2)		
	Stop Distance (mm)	Stop Time (sec)
Test 1	267.73	0.32
Test 2	269.61	0.32
Test 3	252.86	0.3
Maximum	269.61	0.32
Average	263.4	0.31
Condition	Max. Reach / Max. Velocity / Vertical Downward Motion	

RB16-900E Series Base (Joint 0)		
	Stop Distance (mm)	Stop Time (sec)
Test 1	63.69	0.27
Test 2	67.113	0.29
Test 3	60.864	0.31
Maximum	67.113	0.31
Average	63.89	0.29
Condition	Max. Reach / Max. Velocity / Horizontal Motion	

RB16-900E Series Shoulder (Joint 1)		
	Stop Distance (mm)	Stop Time (sec)
Test 1	140.06	0.33
Test 2	135.55	0.31
Test 3	143.88	0.33
Maximum	143.88	0.33
Average	139.83	0.32
Condition	Max. Reach / Max. Velocity / Vertical Downward Motion	

RB16-900E Series Elbow (Joint 2)		
	Stop Distance (mm)	Stop Time (sec)
Test 1	270.96	0.32
Test 2	266.4	0.32
Test 3	252.93	0.31
Maximum	270.96	0.32
Average	263.43	0.316
Condition	Max. Reach / Max. Velocity / Vertical Downward Motion	

RB3-730ES Series Base (Joint 0)		
	Stop Distance (mm)	Stop Time (sec)
Test 1	122.8	0.28
Test 2	118.11	0.27
Test 3	125.99	0.28
Maximum	125.99	0.28
Average	122.3	0.276
Condition	Max. Reach / Max. Velocity / Horizontal Motion	

RB3-730ES Series Shoulder (Joint 1)		
	Stop Distance (mm)	Stop Time (sec)
Test 1	182.17	0.3
Test 2	190.41	0.35
Test 3	182.64	0.3
Maximum	190.41	0.35
Average	185.07	0.32
Condition	Max. Reach / Max. Velocity / Vertical Downward Motion	

RB3-730ES Series Elbow (Joint 2)		
	Stop Distance (mm)	Stop Time (sec)
Test 1	151.06	0.24
Test 2	147.56	0.23
Test 3	156.75	0.28
Maximum	156.75	0.28
Average	151.79	0.25
Condition	Max. Reach / Max. Velocity / Vertical Downward Motion	

RB6-920ES Series Base (Joint 0)		
	Stop Distance (mm)	Stop Time (sec)
Test 1	171.72	0.24
Test 2	160.14	0.18
Test 3	165.44	0.18
Maximum	171.72	0.24
Average	165.77	0.2
Condition	Max. Reach / Max. Velocity / Horizontal Motion	

RB6-920ES Series Shoulder (Joint 1)		
	Stop Distance (mm)	Stop Time (sec)
Test 1	211.4	0.31
Test 2	211.6	0.3
Test 3	196.8	0.25
Maximum	211.6	0.31
Average	206.6	0.29
Condition	Max. Reach / Max. Velocity / Vertical Downward Motion	

RB6-920ES Series Elbow (Joint 2)		
	Stop Distance (mm)	Stop Time (sec)
Test 1	217.3	0.21
Test 2	211.6	0.21
Test 3	215.6	0.21
Maximum	217.3	0.21
Average	214.83	0.21
Condition	Max. Reach / Max. Velocity / Vertical Downward Motion	

RB20-1900ES Series Base (Joint 0)		
	Stop Distance (mm)	Stop Time (sec)
Test 1	219.67	0.21
Test 2	205.5	0.2
Test 3	217.72	0.2
Maximum	219.67	0.21
Average	214.3	0.2
Condition	Max. Reach / Max. Velocity / Horizontal Motion	

RB20-1900ES Series Shoulder (Joint 1)		
	Stop Distance (mm)	Stop Time (sec)
Test 1	359.34	0.28
Test 2	355.6	0.27
Test 3	363.33	0.29
Maximum	363.33	0.29
Average	359.42	0.28
Condition	Max. Reach / Max. Velocity / Vertical Downward Motion	

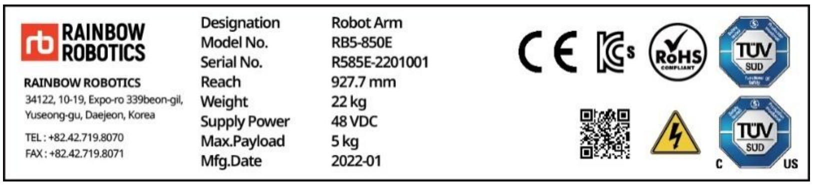
RB20-1900ES Series Elbow (Joint 2)		
	Stop Distance (mm)	Stop Time (sec)
Test 1	205.35	0.24
Test 2	150.78	0.2
Test 3	210.02	0.26
Maximum	210.02	0.26
Average	188.72	0.23
Condition	Max. Reach / Max. Velocity / Vertical Downward Motion	

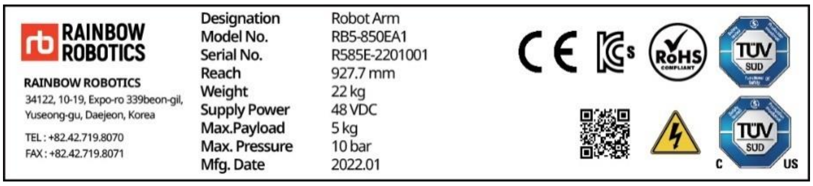
APPENDIX H. NAMEPLATE

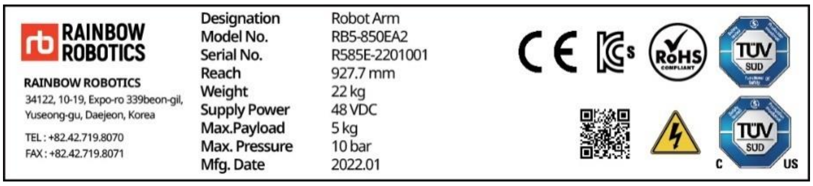
The nameplate of the robot is divided into the robot arm and the control box as shown below.

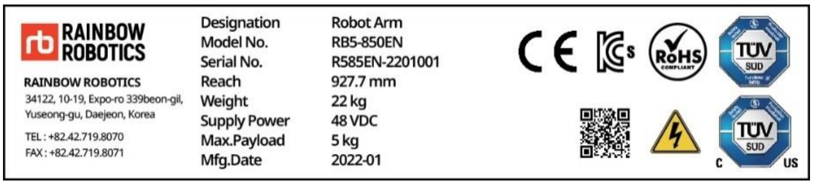
[Robot Arm]

RB5-850E Series

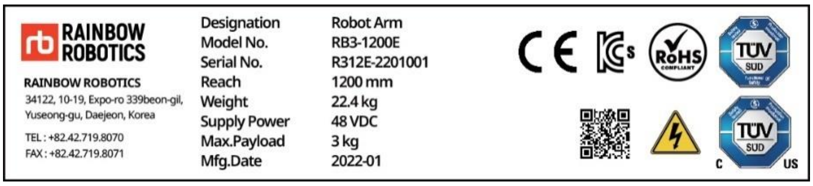


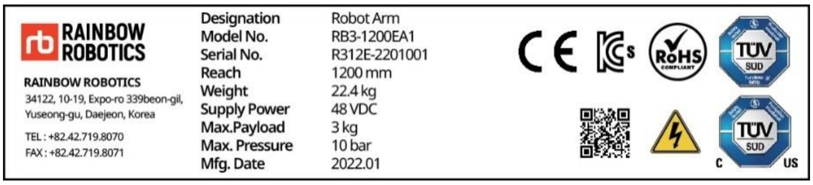


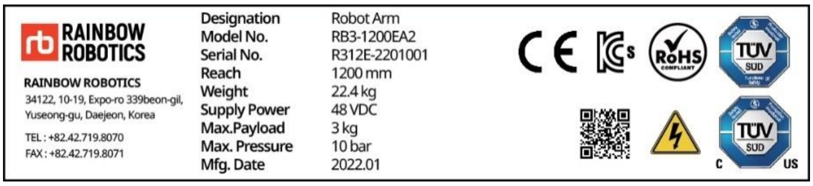


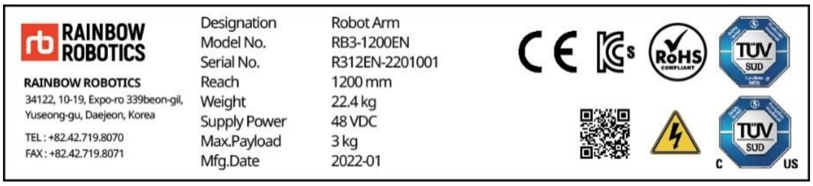


RB3-1200E Series

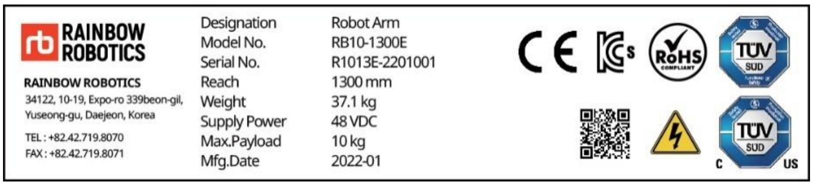


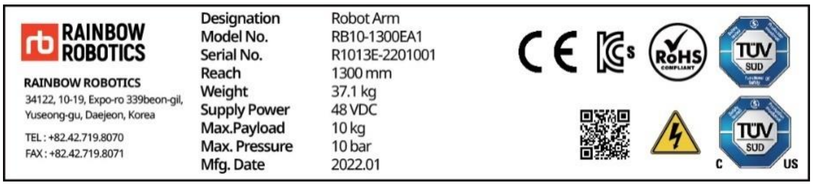


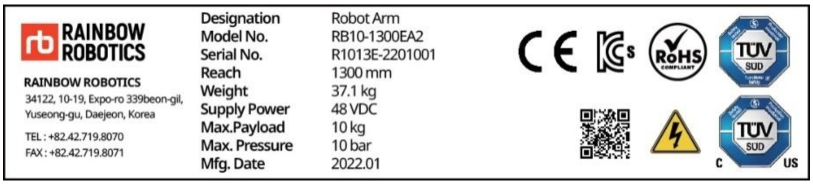


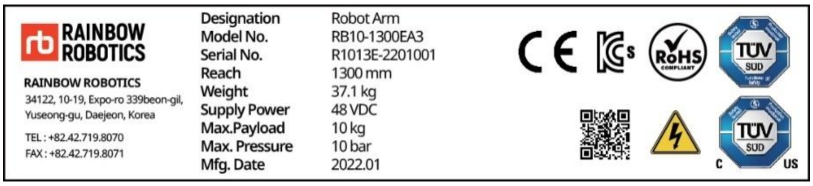


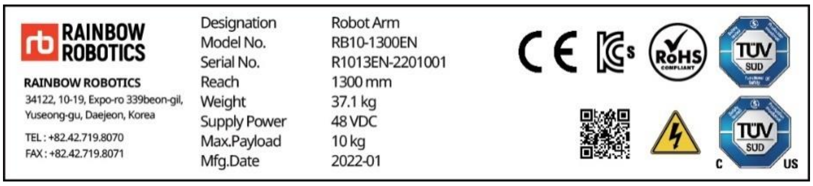
RB10-1300E Series



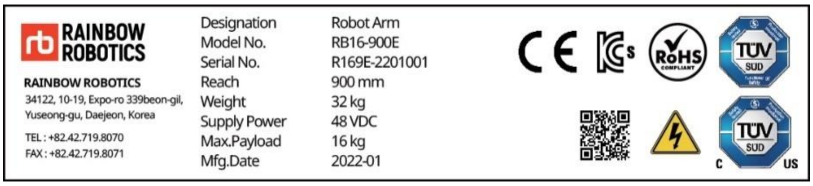


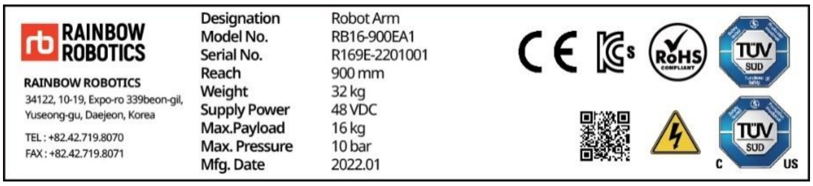


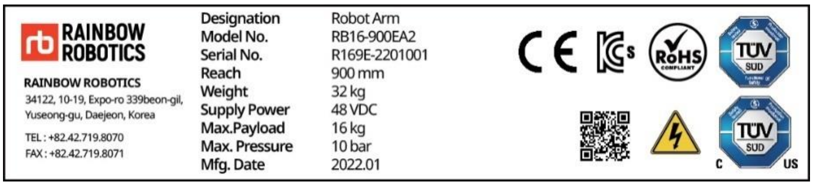


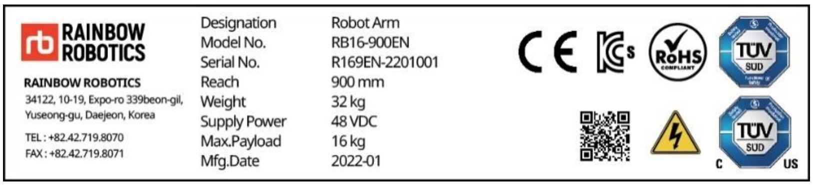


RB16-900E Series

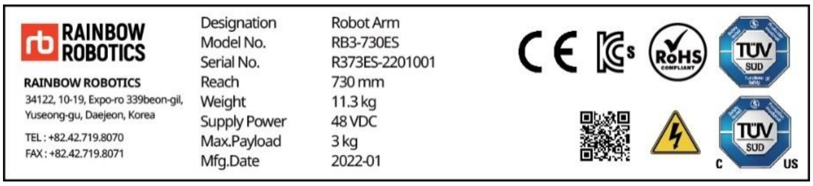


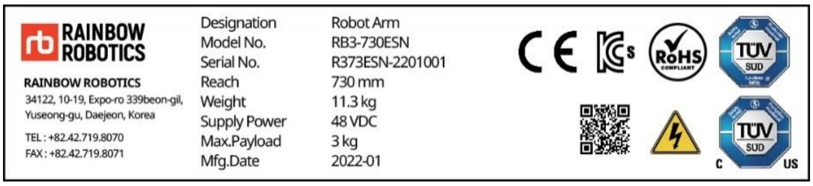




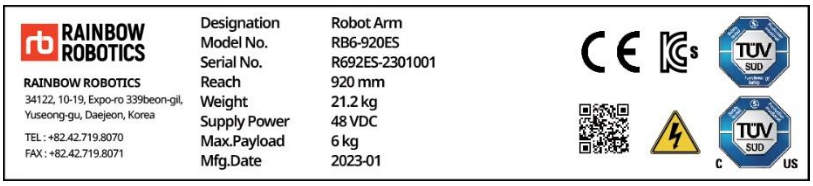


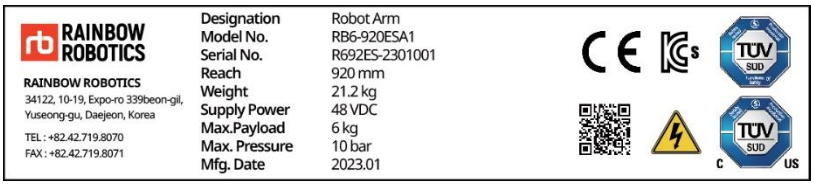
RB3-730ES Series

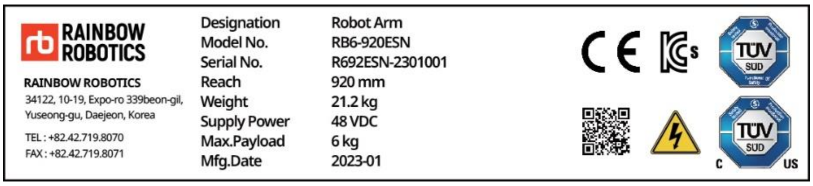




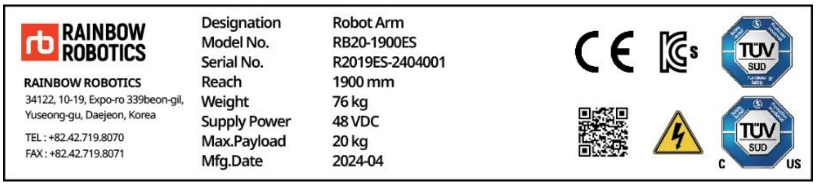
RB6-920ES Series

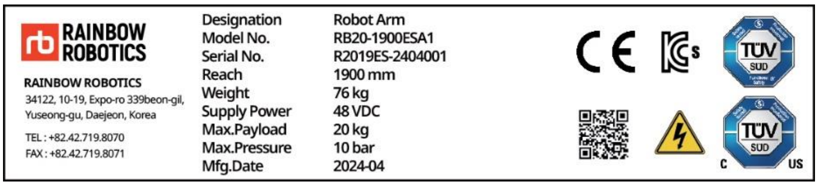


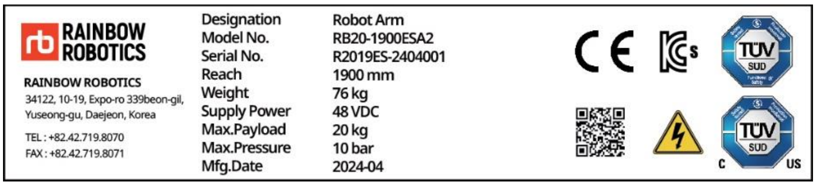


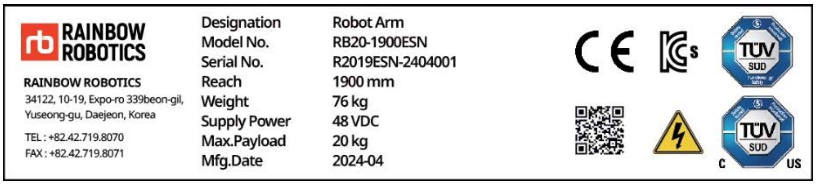


RB20-1900ES Series



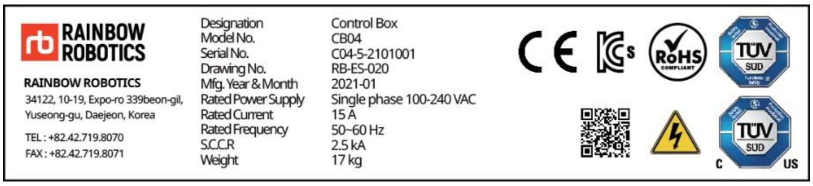




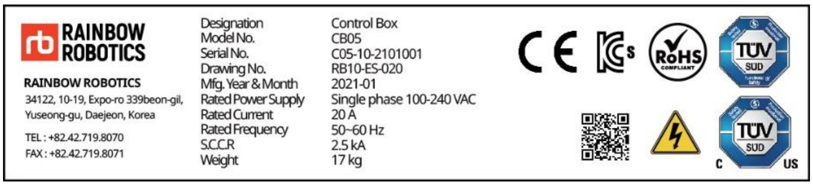


[Control Box]

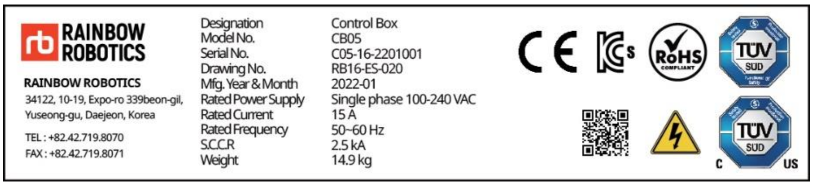
RB5-850E Series, RB3-1200E Series: (CB04)



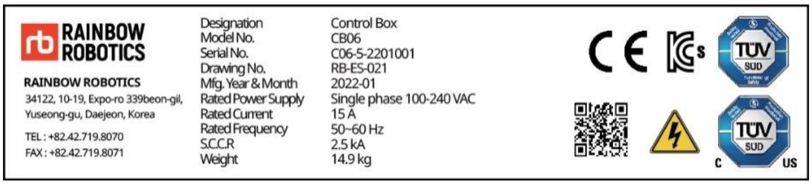
RB10-1300E Series: (CB05)



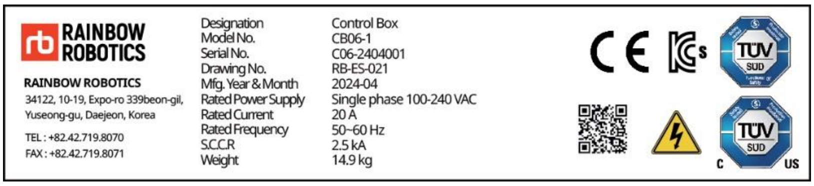
RB16-900E Series: (CB05)



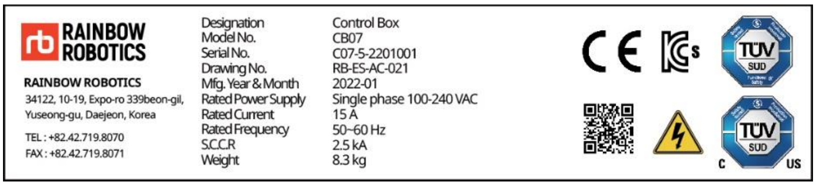
RB5-850E Series, RB3-1200E Series, RB10-1300E Series, RB16-900E Series, RB6-920ES Series: Stand Control Box(CB06)



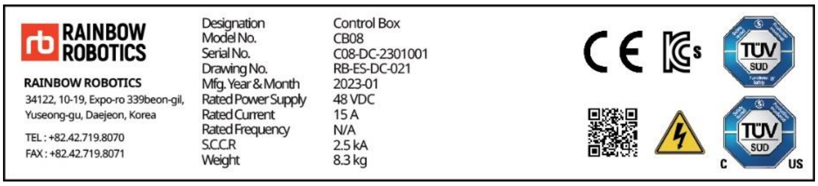
RB20-1900ES Series: Stand Control Box(CB06-1)



RB3-730ES Series, RB6-920ES Series: Compact Control Box(CB07)



RB10-1300E Series: DC Control Box(CB08)



APPENDIX I. MODBUS TCP SERVER

- Warning

  This manual describes the Modbus server (slave controller). See Section 6 for a description of the Modbus client features.

1. Overview

RB's Modbus TCP server (slave controller) is fixed at port number 502. The IP address changes depending on the network settings through the UI. (The initial IP address is 10.0.2.7.).

RB's Modbus server allows the connection of multiple clients and executes the following operation commands.

	Function Code	Function Name
Bit Address	2	Read Discrete Inputs
	1	Read Coils
	5	Write Single Coil
	15	Write Multiple Coils
16-bit (Word) Address	4	Read Input Registers
	3	Read Multiple Holding Registers
	6	Write Single Holding Register
	16	Write Multiple Holding Registers

2. Exception Code

The following error message is returned when accessing the wrong address, incorrect range of values, or invalid command sent.

Exception Code	Exception Name
1	Illegal Function
2	Illegal Data Address
3	Illegal Value

3. Bit Address Map

Bit Address			
Address	Function	Read	Write
0	Box digital input 0	o	x
1	Box digital input 1	o	x
2	Box digital input 2	o	x
3	Box digital input 3	o	x
4	Box digital input 4	o	x
5	Box digital input 5	o	x
6	Box digital input 6	o	x
7	Box digital input 7	o	x
8	Box digital input 8	o	x
9	Box digital input 9	o	x
10	Box digital input 10	o	x
11	Box digital input 11	o	x
12	Box digital input 12	o	x
13	Box digital input 13	o	x
14	Box digital input 14	o	x
15	Box digital input 15	o	x
16	Box digital output 0	o	o
17	Box digital output 1	o	o
18	Box digital output 2	o	o
19	Box digital output 3	o	o
20	Box digital output 4	o	o
21	Box digital output 5	o	o
22	Box digital output 6	o	o
23	Box digital output 7	o	o
24	Box digital output 8	o	o
25	Box digital output 9	o	o
26	Box digital output 10	o	o
27	Box digital output 11	o	o
28	Box digital output 12	o	o
29	Box digital output 13	o	o
30	Box digital output 14	o	o
31	Box digital output 15	o	o
32	Tool digital input 0	o	x
33	Tool digital input 1	o	x
34	Tool digital output 0	o	o
35	Tool digital output 1	o	o

4. Word(16 bit) Address Map

Word Address				
Address	Function	Read	Write	Comments
0	Box digital input 0~15	o	x	[BBBB BBBB BBBB BBBB]
1	Box digital output 0~15	o	o	[BBBB BBBB BBBB BBBB]
2	Box analog input 0	o	x	1mV unit
3	Box analog input 1	o	x	1mV unit
4	Box analog input 2	o	x	1mV unit
5	Box analog input 3	o	x	1mV unit
6	Box analog output 0	o	o	1mV unit
7	Box analog output 1	o	o	1mV unit
8	Box analog output 2	o	o	1mV unit
9	Box analog output 3	o	o	1mV unit
10	Extend digital input 0~15	o	x	[BBBB BBBB BBBB BBBB]
11	Extend digital output 0~15	o	x	[BBBB BBBB BBBB BBBB]
12	Extend analog input 0	o	x	1mV unit
13	Extend analog input 1	o	x	1mV unit
14	Extend analog input 2	o	x	1mV unit
15	Extend analog input 3	o	x	1mV unit
16	Extend analog output 0	o	o	1mV unit
17	Extend analog output 1	o	o	1mV unit
18	Extend analog output 2	o	o	1mV unit
19	Extend analog output 3	o	o	1mV unit
20	Reserved (Box I/O)	x	x	
21	Reserved (Box I/O)	x	x	
22	Reserved (Box I/O)	x	x	
23	Reserved (Box I/O)	x	x	
24	Reserved (Box I/O)	x	x	
25	Reserved (Box I/O)	x	x	
26	Reserved (Box I/O)	x	x	
27	Reserved (Box I/O)	x	x	
28	Reserved (Box I/O)	x	x	
29	Reserved (Box I/O)	x	x	
30	Tool output voltage	o	o	0, 12, 24
31	Tool digital input 0~1 (or 0~5)	o	x	[TTEE EExx xxxx xxxx]
32	Tool digital output 0~1	o	o	[TTxx xxxx xxxx xxxx]
33	Tool analog input 0	o	x	1mV unit
34	Tool analog input 1	o	x	1mV unit
35	Reserved (Tool I/O)	x	x	
36	Reserved (Tool I/O)	x	x	

37	Reserved (Tool I/O)	x	x	
38	Reserved (Tool I/O)	x	x	
39	Reserved (Tool I/O)	x	x	
40	Reserved (Tool I/O)	x	x	
41	Reserved (Tool I/O)	x	x	
42	Reserved (Tool I/O)	x	x	
43	Reserved (Tool I/O)	x	x	
44	Reserved (Tool I/O)	x	x	
45	Reserved (Tool I/O)	x	x	
46	Reserved (Tool I/O)	x	x	
47	Reserved (Tool I/O)	x	x	
48	Reserved (Tool I/O)	x	x	
49	Reserved (Tool I/O)	x	x	
50	Is Robot Activated	o	x	0 or 1
51	Is Real-mode	o	x	0 or 1
52	Is Collision Detected	o	x	0 or 1
53	Is Robot arm power engaged	o	x	0 or 1
54	Is Direct Teaching mode	o	x	0 or 1
55	Is Robot moving	o	x	0 or 1
56	Is Pause state	o	x	0 or 1
57	Is Teaching pendant is connected	o	x	0 or 1
58	Is Program Run	o	x	0 or 1
59	Is No-Arc mode is on	o	x	0 or 1
60	Is EMG button released	o	x	0 or 1
61	Is First Program Run	o	x	0 or 1
62	Is Last Program Load Done	o	x	0 or 1
63	Control Box Amphere	o	x	0.02A unit
64	Is Under Activation	o	x	0 or 1
65	Collision On/Off State	o	x	0 or 1
66	Collision Threshold	o	x	% unit
67	TCP Px	o	x	0.1mm unit / Signed
68	TCP Py	o	x	0.1mm unit / Signed
69	TCP Pz	o	x	0.1mm unit / Signed
70	TCP Rx	o	x	0.02deg unit / Signed
71	TCP Ry	o	x	0.02deg unit / Signed
72	TCP Rz	o	x	0.02deg unit / Signed
73	Mass m	o	x	0.1kg unit / Signed
74	Is System SOS	o	x	Arm SOS Flag
75	Is Home Is Begin	o	x	0 or 1
76	Control Box Heart Beat (0 or 1)	o	x	0 or 1 (Every 1 second)

77	Speed Bar Value	o	x	% unit
78	N/A	o	x	
79	N/A	o	x	
80	RB->MBUS Welder Weld Start Cmd	o	x	0 or 1
81	RB->MBUS Welder Gas on off	o	x	0 or 1
82	RB->MBUS Welder Inching	o	x	0 or 1
83	RB->MBUS Welder Re Inching	o	x	0 or 1
84	RB->MBUS Welder Touch Start Cmd	o	x	0 or 1
85	RB->MBUS Welder Control parameter 1	o	x	-
86	RB->MBUS Welder Control parameter 2	o	x	-
87	RB->MBUS Welder Control parameter 3	o	x	-
88	RB->MBUS Welder Control parameter 4	o	x	-
89	RB->MBUS Welder Control parameter 5	o	x	-
90	Reserved (Future System)	x	x	
91	Reserved (Future System)	x	x	
92	Reserved (Future System)	x	x	
93	Reserved (Future System)	x	x	
94	Reserved (Future System)	x	x	
95	Reserved (Future System)	x	x	
96	Reserved (Future System)	x	x	
97	Reserved (Future System)	x	x	
98	Reserved (Future System)	x	x	
99	Reserved (Future System)	x	x	
100	Command: Start Program Once	o	o	Rising Edge is command
101	Command: Start Program Repeat	o	o	Rising Edge is command
102	Command: Pause Program	o	o	Rising Edge is command
103	Command: Stop Program	o	o	Rising Edge is command
104	Command: Resume from pause	o	o	Rising Edge is command
105	Command: Resume from collision	o	o	Rising Edge is command
106	Command: Load default Program	o	o	Rising Edge is command
107	Command: Robot Arm activation	o	o	Rising Edge is command

108	Command: Change to Real-mode	o	o	Rising Edge is command
109	Command: Power off the robot arm	o	o	Rising Edge is command
110	Command: Load Program Table	o	o	#
111	Command: Reset SOS Flag	o	o	
112	Command: Speed Bar Chage	o	o	% unit
113	N/A	o	o	
114	N/A	o	o	
115	N/A	o	o	
116	N/A	o	o	
117	N/A	o	o	
118	MBUS Welder->RB Info Arc ON	o	o	0 or 1
119	MBUS Welder->RB Welder ERROR	o	o	0 or 1
120	MBUS Welder->RB Touch Sensed	o	o	0 or 1
121	MBUS Welder->RB INFO parameter 1	o	o	-
122	MBUS Welder->RB INFO parameter 2	o	o	-
123	MBUS Welder->RB INFO parameter 3	o	o	-
124	MBUS Welder->RB INFO parameter 4	o	o	-
125	MBUS Welder->RB INFO parameter 5	o	o	-
126	N/A	o	o	
127	N/A	o	o	
128~255	User General Purpose Register (User can read/write this area for program) SD_MOD_USER_#	o	o	User Define Area
256	Joint reference 0	o	x	0.02deg unit / Signed
257	Joint reference 1	o	x	0.02deg unit / Signed
258	Joint reference 2	o	x	0.02deg unit / Signed
259	Joint reference 3	o	x	0.02deg unit / Signed
260	Joint reference 4	o	x	0.02deg unit / Signed
261	Joint reference 5	o	x	0.02deg unit / Signed
262	Joint angle 0	o	x	0.02deg unit / Signed
263	Joint angle 1	o	x	0.02deg unit / Signed
264	Joint angle 2	o	x	0.02deg unit / Signed
265	Joint angle 3	o	x	0.02deg unit / Signed
266	Joint angle 4	o	x	0.02deg unit / Signed

267	Joint angle 5	o	x	0.02deg unit / Signed
268	Joint current 0	o	x	10mA unit / Signed
269	Joint current 1	o	x	10mA unit / Signed
270	Joint current 2	o	x	10mA unit / Signed
271	Joint current 3	o	x	10mA unit / Signed
272	Joint current 4	o	x	10mA unit / Signed
273	Joint current 5	o	x	10mA unit / Signed
274	Joint information 0	o	x	
275	Joint information 1	o	x	
276	Joint information 2	o	x	
277	Joint information 3	o	x	
278	Joint information 4	o	x	
279	Joint information 5	o	x	
280	Joint temperature 0	o	x	celcius unit
281	Joint temperature 1	o	x	celcius unit
282	Joint temperature 2	o	x	celcius unit
283	Joint temperature 3	o	x	celcius unit
284	Joint temperature 4	o	x	celcius unit
285	Joint temperature 5	o	x	celcius unit
286	Joint 0 Estimated Current	o	x	10mA unit / Signed
287	Joint 1 Estimated Current	o	x	10mA unit / Signed
288	Joint 2 Estimated Current	o	x	10mA unit / Signed
289	Joint 3 Estimated Current	o	x	10mA unit / Signed
290	Joint 4 Estimated Current	o	x	10mA unit / Signed
291	Joint 5 Estimated Current	o	x	10mA unit / Signed
292	Joint 0 Gap(Esti.-Meas.) Current	o	x	10mA unit / Signed
293	Joint 1 Gap(Esti.-Meas.) Current	o	x	10mA unit / Signed
294	Joint 2 Gap(Esti.-Meas.) Current	o	x	10mA unit / Signed
295	Joint 3 Gap(Esti.-Meas.) Current	o	x	10mA unit / Signed
296	Joint 4 Gap(Esti.-Meas.) Current	o	x	10mA unit / Signed
297	Joint 5 Gap(Esti.-Meas.) Current	o	x	10mA unit / Signed
298	Joint 0 Gap(Esti.-Meas.) Curr+LPF	o	x	10mA unit / Signed
299	Joint 1 Gap(Esti.-Meas.) Curr+LPF	o	x	10mA unit / Signed
300	Joint 2 Gap(Esti.-Meas.) Curr+LPF	o	x	10mA unit / Signed
301	Joint 3 Gap(Esti.-Meas.) Curr+LPF	o	x	10mA unit / Signed

302	Joint 4 Gap(Esti.-Meas.) Curr+LPF	o	x	10mA unit / Signed
303	Joint 5 Gap(Esti.-Meas.) Curr+LPF	o	x	10mA unit / Signed
304	External FT Sensor Data FX	o	x	0.02 unit / Signed
305	External FT Sensor Data FY	o	x	0.02 unit / Signed
306	External FT Sensor Data FZ	o	x	0.02 unit / Signed
307	External FT Sensor Data MX	o	x	0.02 unit / Signed
308	External FT Sensor Data MY	o	x	0.02 unit / Signed
309	External FT Sensor Data MZ	o	x	0.02 unit / Signed
310	External Axis Reference 0	o	x	0.5 unit / Signed
311	External Axis Reference 1	o	x	0.5 unit / Signed
312	External Axis Reference 2	o	x	0.5 unit / Signed
313	External Axis Reference 3	o	x	0.5 unit / Signed
314	External Axis Reference 4	o	x	0.5 unit / Signed
315	External Axis Reference 5	o	x	0.5 unit / Signed
316	External Axis Encoder 0	o	x	0.5 unit / Signed
317	External Axis Encoder 1	o	x	0.5 unit / Signed
318	External Axis Encoder 2	o	x	0.5 unit / Signed
319	External Axis Encoder 3	o	x	0.5 unit / Signed
320	External Axis Encoder 4	o	x	0.5 unit / Signed
321	External Axis Encoder 5	o	x	0.5 unit / Signed
322	Reserved (Future System)	x	x	
323	Reserved (Future System)	x	x	
324	Reserved (Future System)	x	x	
325	Reserved (Future System)	x	x	
326	Reserved (Future System)	x	x	
327	Reserved (Future System)	x	x	
328	Reserved (Future System)	x	x	
329	Reserved (Future System)	x	x	
330	TCP reference X	o	x	0.1mm unit / Signed
331	TCP reference Y	o	x	0.1mm unit / Signed
332	TCP reference Z	o	x	0.1mm unit / Signed
333	TCP reference RX	o	x	0.02deg unit / Signed
334	TCP reference RY	o	x	0.02deg unit / Signed
335	TCP reference RZ	o	x	0.02deg unit / Signed
336	TCP position X	o	x	0.1mm unit / Signed
337	TCP position Y	o	x	0.1mm unit / Signed
338	TCP position Z	o	x	0.1mm unit / Signed
339	TCP position RX	o	x	0.02deg unit / Signed
340	TCP position RY	o	x	0.02deg unit / Signed

341	TCP position RZ	o	x	0.02deg unit / Signed
342	TCP Speed X	o	x	0.5 unit / Signed
343	TCP Speed Y	o	x	0.5 unit / Signed
344	TCP Speed Z	o	x	0.5 unit / Signed
345	TCP Speed Norm	o	x	0.5 unit / Signed
346	Reserved (Future System)	x	x	
347	Reserved (Future System)	x	x	
348	Reserved (Future System)	x	x	
349	Reserved (Future System)	x	x	
350	Reserved (Future System)	x	x	
351	Reserved (Future System)	x	x	
352	Reserved (Future System)	x	x	
353	Reserved (Future System)	x	x	
354	Reserved (Future System)	x	x	
355	Reserved (Future System)	x	x	
356	Reserved (Future System)	x	x	
357	Reserved (Future System)	x	x	
358	Reserved (Future System)	x	x	
359	Reserved (Future System)	x	x	
360	Reserved (Future System)	x	x	
361	Reserved (Future System)	x	x	
362	Reserved (Future System)	x	x	
363	Reserved (Future System)	x	x	
364	Reserved (Future System)	x	x	
365	Reserved (Future System)	x	x	
366	Reserved (Future System)	x	x	
367	Reserved (Future System)	x	x	
368	Reserved (Future System)	x	x	
369	Reserved (Future System)	x	x	
370	Reserved (Future System)	x	x	
371	Reserved (Future System)	x	x	
372	Reserved (Future System)	x	x	
373	Reserved (Future System)	x	x	
374	Reserved (Future System)	x	x	
375	Motion Generator: Parameter 0	o	o	
376	Motion Generator: Parameter 1	o	o	
377	Motion Generator: Parameter 2	o	o	
378	Motion Generator: Parameter 3	o	o	
379	Motion Generator: Parameter 4	o	o	

380	Motion Generator: Parameter 5	o	o	
381	Motion Generator: Parameter Speed	o	o	
382	Motion Generator: Parameter Accel.	o	o	
383	Motion Generator: -	o	o	
384	Motion Generator: -	o	o	
385	Motion Generator: -	o	o	
386	Motion Generator: -	o	o	
387	Motion Generator: -	o	o	
388	Motion Generator: Motion Command	o	o	
389	Motion Generator: Error Report	o	o	
EOT				

APPENDIX J. SYSTEM UPDATE

- Warning
  It is recommended to back up the program files (.wsl) inside the tablet UI before the system update.

1. Overview

Rainbow Robotics' system update is a two-step process.

UI update through APK install > System software (control box) update

2. Backup Program file

Connect the tablet and personal/business PC and obtain the program file (.wsl) from the path below and back it up.

Tablet → Android → data → com.rainbow.cobot → files → work→ GET .wsl files

※ It is recommended that you back up the acquired files before proceeding to the next step.

3. UI Update

Rainbow Robotics' tablet UI program is distributed in the form of APK.

This is the same installation file as a regular Android application. Therefore, UI program is updated by moving the installation APK file to the tablet and installing it.

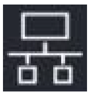
※ Rainbow Robotics recommends installing after deleting an existing application.

※ When deleting an existing application, the program file (.wsl) is deleted together. Back up the program file in step 1 and proceed with this process.

Copy the distributed APK file into Table > APK install

4. Connection between Tablet PC and Control Box

Connect the tablet to the control box and access the UI program. After connecting, connect the control box communication with the tablet.

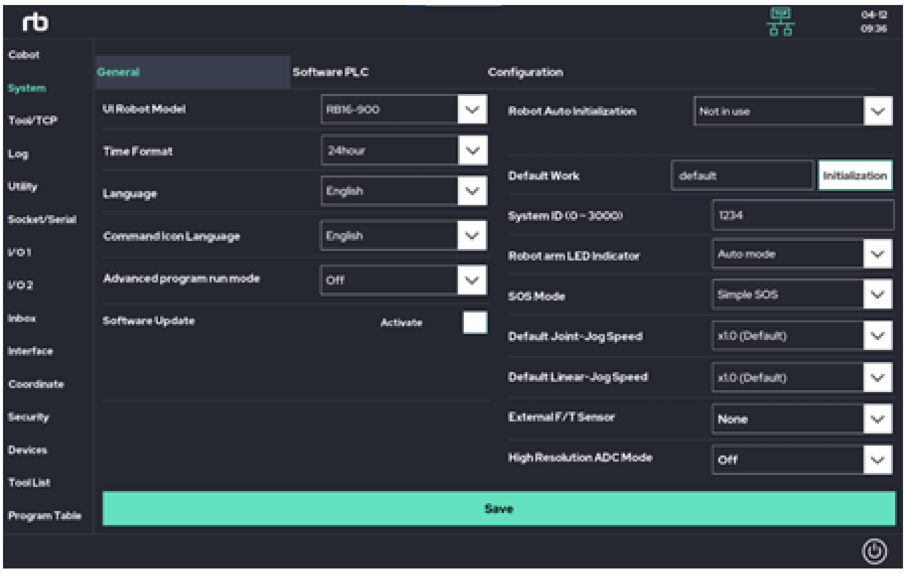
UI Home > Make > Click 'State'  button > Connect

If the communication between the tablet and the control box is normal, the first box will be lit blue. For safety reasons, it is recommended not to initialize the robot.

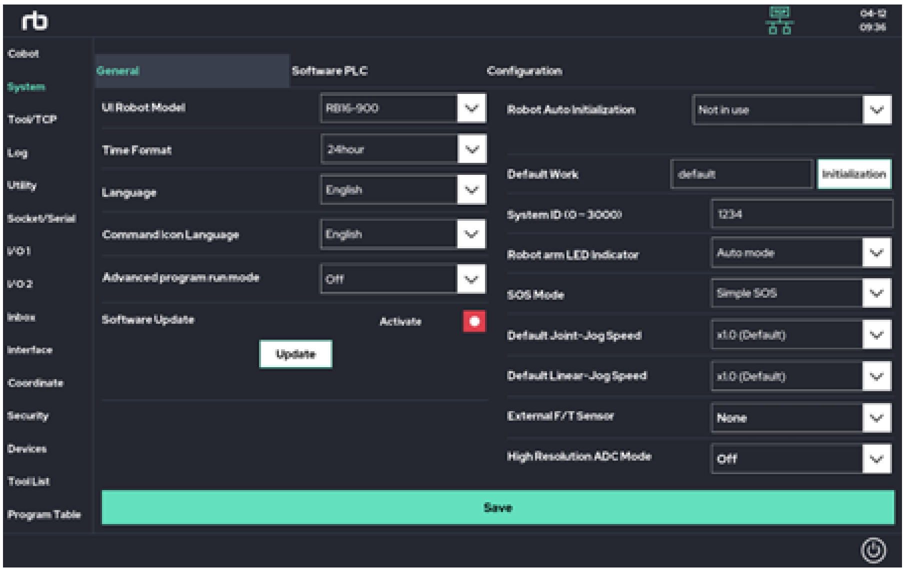
5. Go to and activate the system software update

Navigate to the system software update path as shown below.

UI Home > Setup > System Tab

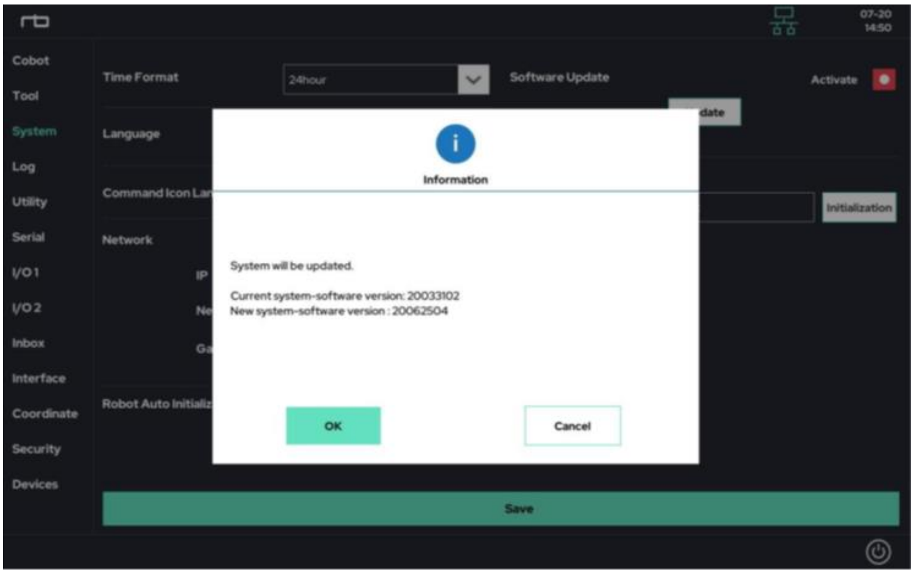


In the “Software Update” section on the right, click the Activate checkbox.



6. Progress System Software Update

The Update button will appear and click this button to open a popup window.



Press "OK" button to update the software.

If the update is completed normally after clicking the OK button, the PC of the control box (controller) will automatically restart within 5~15 seconds.

During the restart process, "Please Wait..." is displayed on the LCD of the control box. Is displayed temporarily. This indicates that the control box is rebooting.

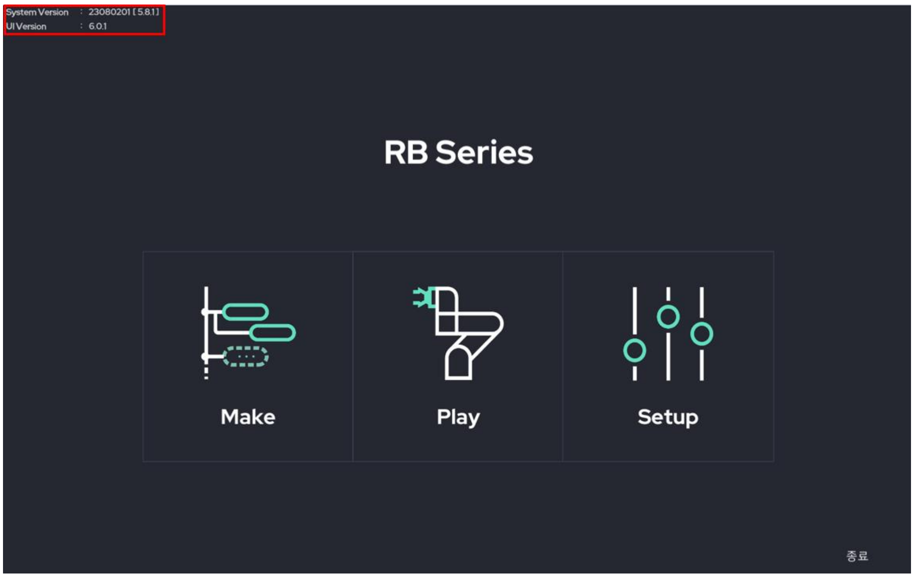
After the reboot is completed, "Normal Operation" is displayed on the LCD of the control box.

7. Check the Update

Reconnect the UI tablet and control box.

UI Home → Top Left

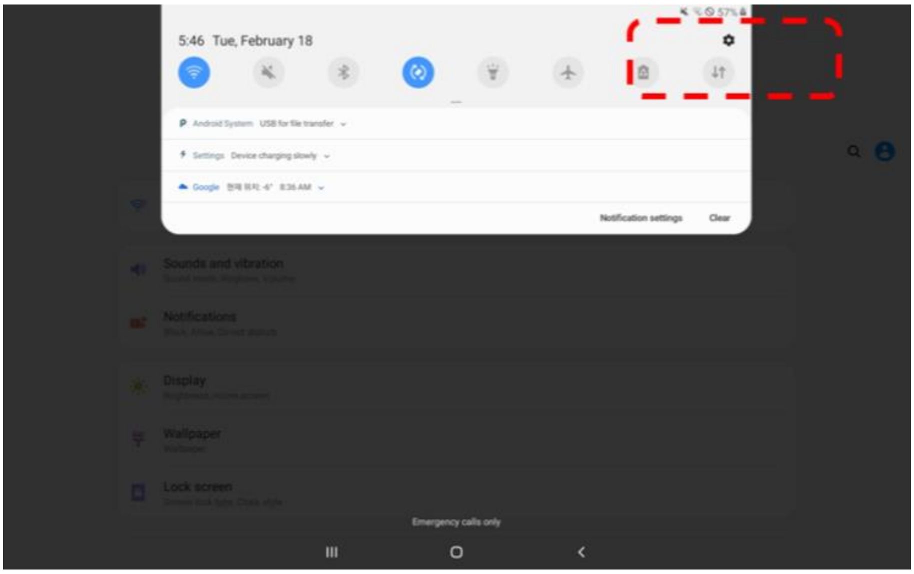
When you go back to the home screen of the UI, the software version is displayed on the upper left. In this context, "System Version" refers to the software version of the control box, while "UI Version" refers to the software version of the tablet PC.



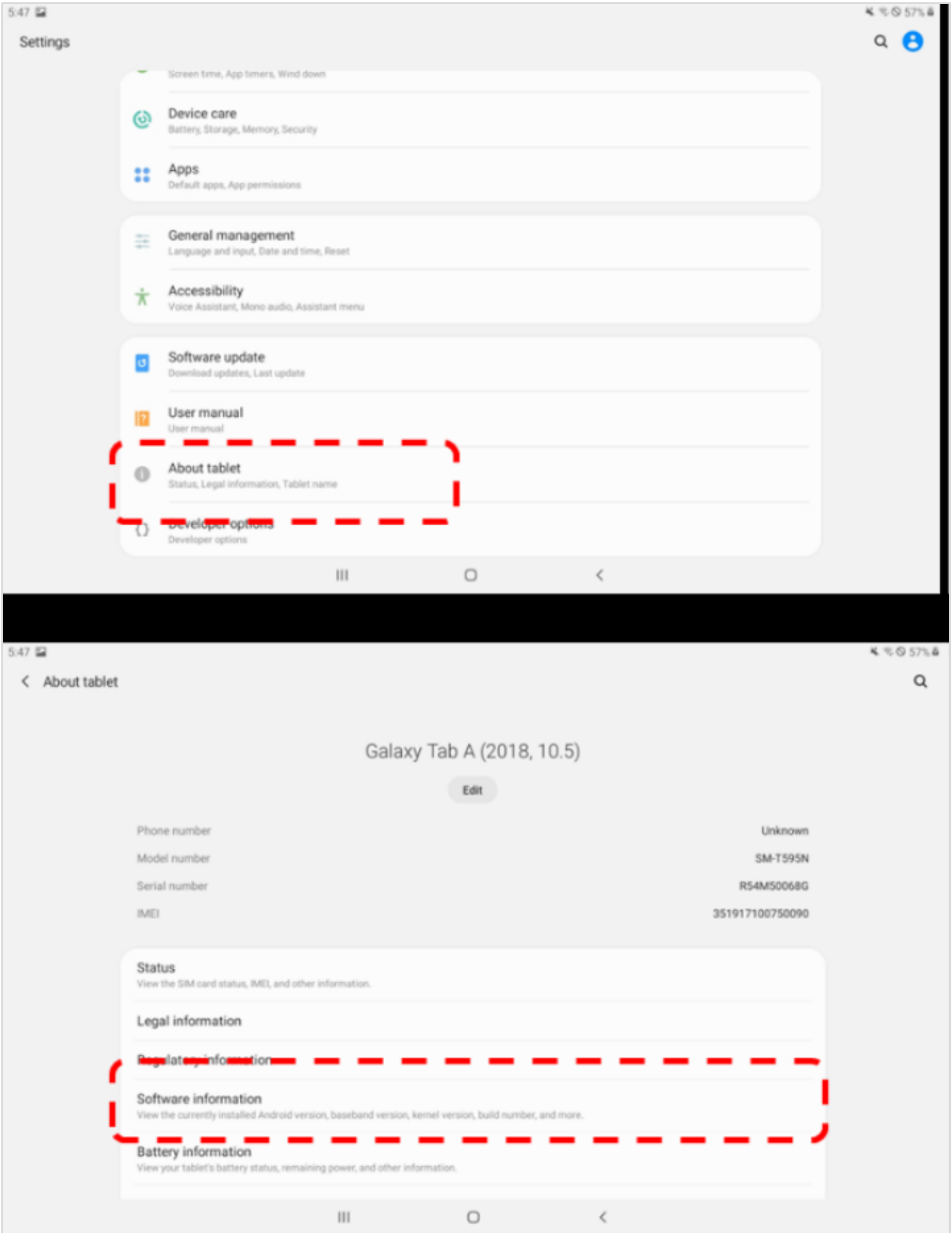
APPENDIX K. ANDROID TABLET CONFIGURATION

Before using the UI program, the following tablet settings are required.

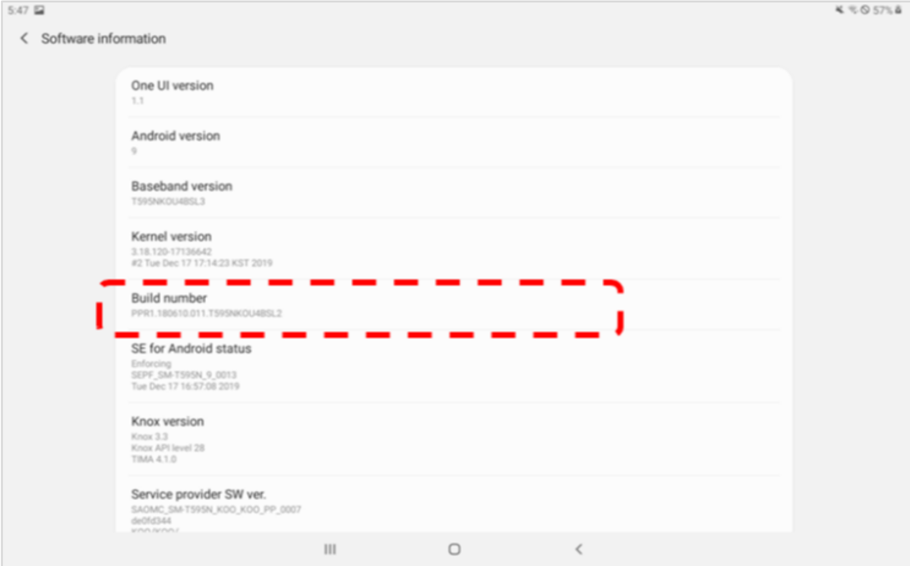
1. Goto Setting section of the Android.



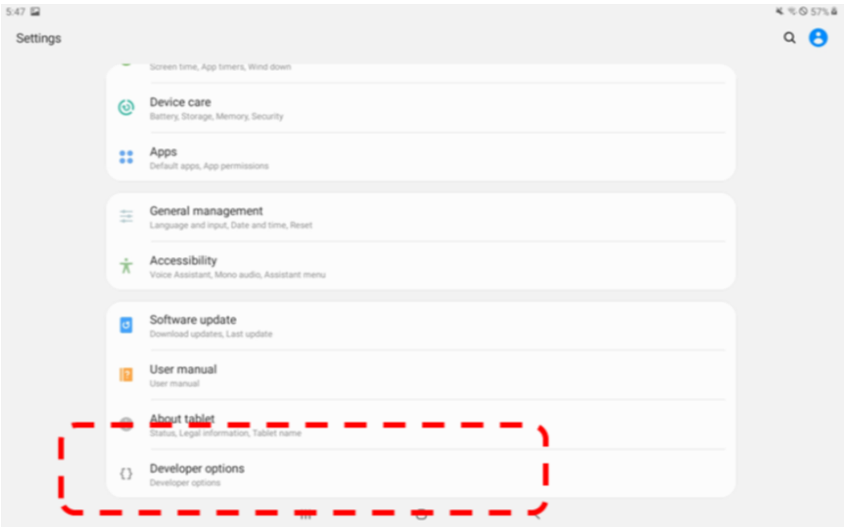
2. Goto "About Tablet" > "Software Information".



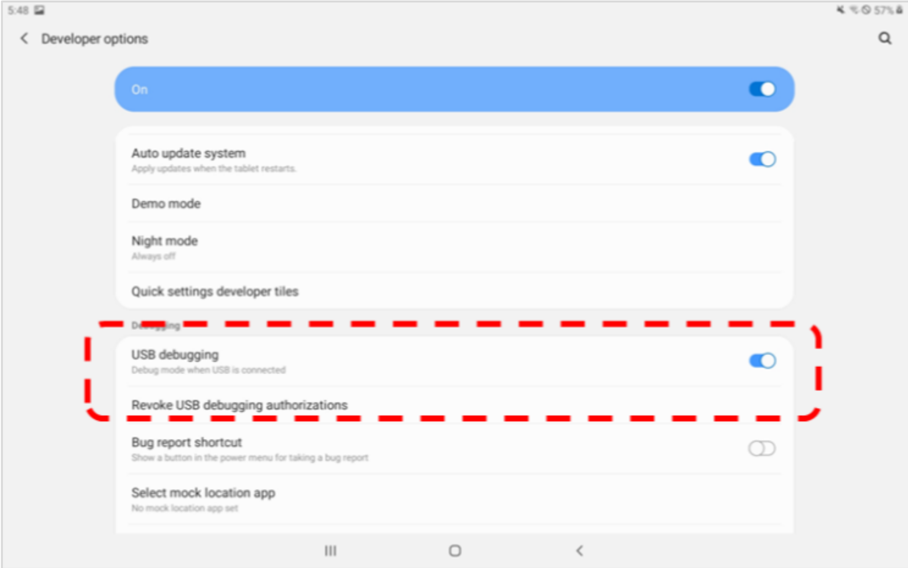
3. Multi-click (7 or more times) “Build Number” of tablet information.



4. A menu called "Developer Options" will appear under "About Tablet" as shown below.



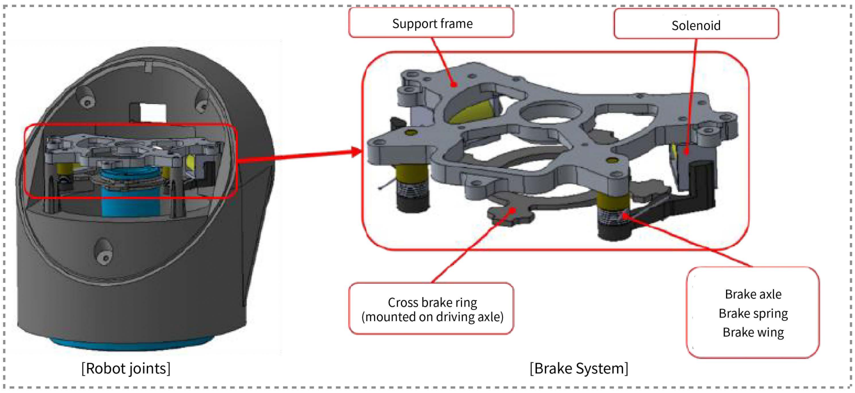
5. Activate "USB Debugging" in "Developer Options".



6. Run the APK distributed by Rainbow Robotics to install the UI program on your tablet.

APPENDIX L. BRAKE SYSTEM

The configuration of the Brake System on each axis of the robot arm consists of a support frame, solenoid, brake ring, brake shaft, brake spring and brake wing, which are installed on the robot joint as shown below.



If the solenoid is on, the physical interference between the turning radius of the brake ring and the brake wing is released, and if the solenoid is off, the physical interference between the end of the brake ring and the brake wing occurs, which stops the rotation of the driveshaft.

When the brake ring rotates and pushes through the brake wing, the wing returns to the spring force, and then a bi-directional brake occurs through physical interference, keeping both bi-directional rotations of the driveshaft stationary.

RAINBOW ROBOTICS

Headquarters and Factory

10-19, Expo-ro 339beon-gil, Yuseong-gu,
Daejeon, Republic of Korea

TEL.	+82-42-719-8070
FAX	+82-42-719-8071
E-mail	rainbow@rainbow-robotics.com
Sales E-mail	sales@rainbow-robotics.com
Homepage	www.rainbow-robotics.com