RCX3 series

RCX320 2 axes

RCX340 3 to 4 axes

[Multi-axis robot controller]

RCX320

Number of axes		2 axes
Operation method		Program, Remote command Online command
Number of points		30000 points
Input power	Control power	Single phase 200 to 230V AC +/-10% maximum
	Main power	Single phase 200 to 230V AC +/-10% maximum
Origin search method		Absolute, Incremental Semi-absolute

RCX340

Number of axes		3 to 4 axes
Operation method		Program, Remote command Online command
Number of points		30000 points
Input power	Control power	Single phase 200 to 230V AC +/-10% maximum
	Main power	Single phase 200 to 230V AC +/-10% maximum
Origin search method		Absolute, Incremental Semi-absolute

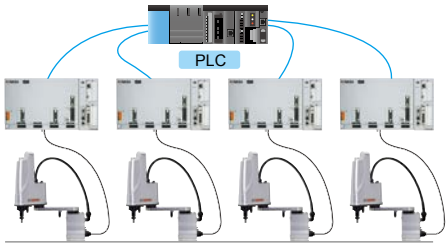
Advanced functionality allowing construction of high-level equipment

Multiple robots can be operated synchronously through the high-speed communication. Use of linking among controllers makes it possible to store programs into only one controller. Use of a newly developed algorithm achieves shortening of the positioning time and improvement of the tracking accuracy.

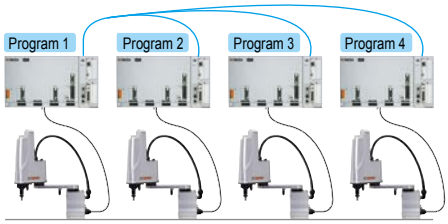
The control of multiple robots can be managed using one master controller

The RCX340 controller allows high-speed communication among the controllers. As the operation command can be sent to the controller of each slave from the master controller, the programs or points can be managed only using the host master controller. Additionally, since the controller flexibly supports multitasking, interactions using PLCs can be simplified, making it easier to build systems at lower costs.

Conventional method



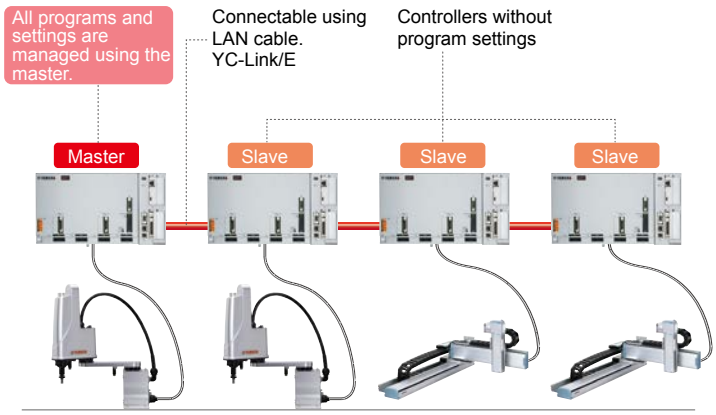
All controllers are controlled using one PLC.



Programs are set for each controller. Operation is performed while synchronizing the robots through I/O.

RCX340

- Up to four robots or 16 axes can be managed with one master RCX340.
- Simultaneous start and arrival of each robot can be controlled freely.

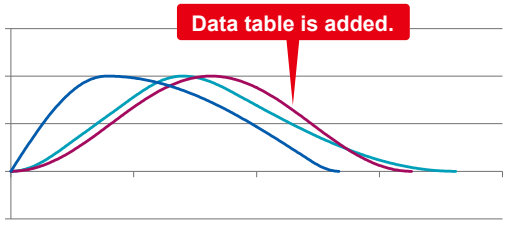


Motion optimization

The optimization of the motion to meet the operation pattern is further strengthened to bring out the robot performance at its maximum level. Higher quality robot operations, such as shortening of the operation time and suppression of vibrations during stopping are achieved.

Optimal acceleration/deceleration motion

Acceleration/deceleration motion is generated that can perform the high-speed operation while suppressing vibrations.



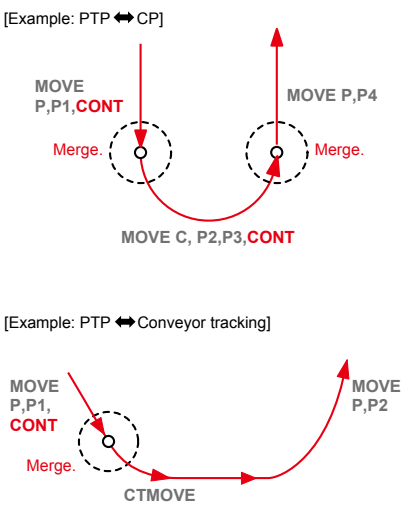
Smooth movement is achieved by greatly improving motion functions

As a new servo motion engine is incorporated, various operations can be merged. Use of a newly developed algorithm achieves shortening of the positioning time and improvement of the tracking accuracy.

Expansion of CONT option function

Different type operations, such as PTP, interpolation operation, and conveyor tracking, etc. are merged to improve the speed.

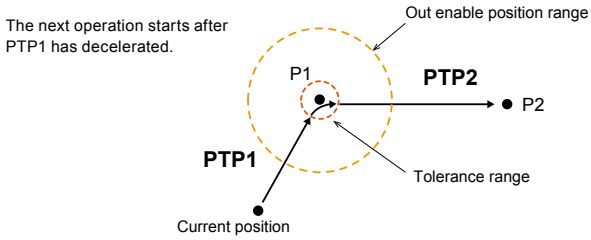
Different type operations can be merged.



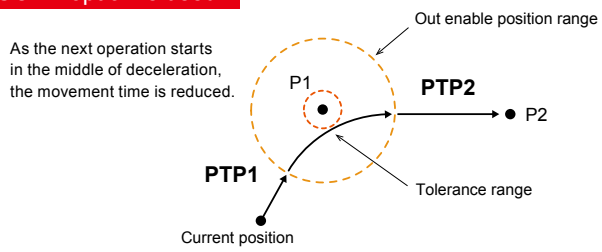
Improvement of continuous operation

By using the CONT option, such as when passing through a relay point in the middle of an operation to avoid an obstacle, it is possible to smoothly merge operations without decelerating and stopping for each operation. Regardless of the type of operation (PTP, interpolation operation), operations can be merged.

Normal movement



CONT option is used.

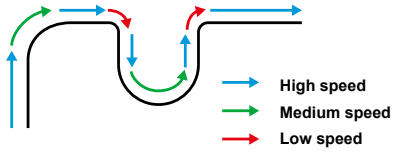


Proper use according to application Note

In merge PTP, priority is given to the movement time, and the movement speed is changed between the straight line section and circular interpolation section. In PATH, by registering paths in advance, it is possible to operate at a constant speed even on complex paths, and tracking accuracy is further improved. This is ideal for applications such as sealing.

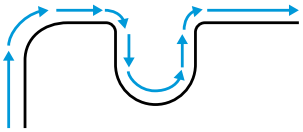
Merge PTP

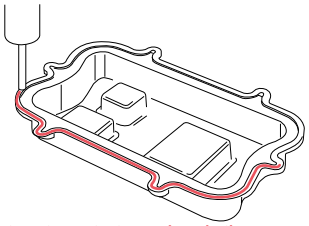
The operation speed is different between the straight line section and circular arc section.



PATH

Moves at constant speed along all paths.





Operates at constant speed on both straight line and circular arc sections.

PBX with USB port for backup

Simple and easy operation for adding function or editing work.

Storing backup data is a simple task.

The operation menu supports Japanese, English, and Chinese.

Convenient LED Display for Error Status.

The operation status is displayed on the “7-segment LED display” located on the front panel of the controller.

If an error occurs, the relevant error message is displayed. The error status can visibly recognized without connecting the programming box.

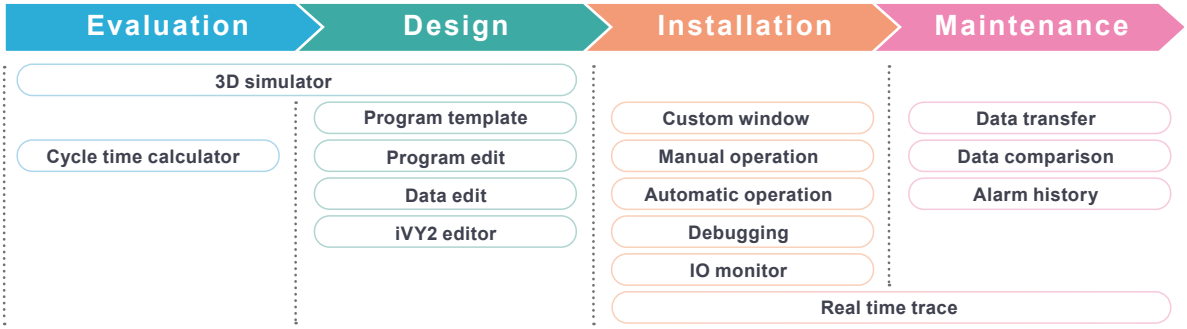
▲ 7-segment LED display

Built-in regenerative unit RCX340

As the regenerative unit (equivalent to RGU3) is built-in, no additional regenerative unit is needed when connecting to the existing robot.

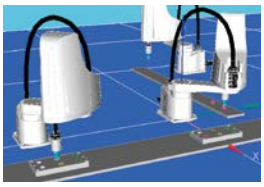
PC Programming Software “RCX-Studio 2020”

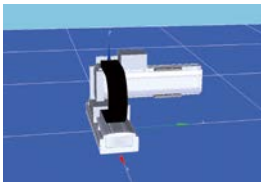
New functions such as 3D simulator function and program template (program template automatic creation function) are added for ease of user operation.



3D simulator

Layout can be verified beforehand without connecting robot.





Robots and peripheral devices are displayed in 3D, and the robot operation is simulated on PC. (This function supports SCARA and Cartesian robots.)

- Robot layout, teaching, and debugging can be performed.
- Physical interference between the robot and peripheral device can be checked before operation is started.

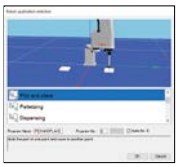
Program template (Program template automatic creation function)

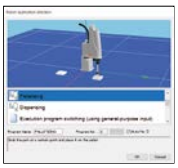
Program creation time can be shortened greatly.

Program templates for 10 types of applications are incorporated. Just following the steps to perform the operation creates a program template automatically.

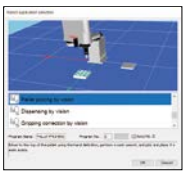
Supported applications

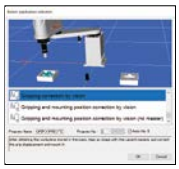
- Pick & place
- Palletizing
- Dispensing work
- Execution program switching
- Conveyor tracking
- Pallet picking using vision
- Dispensing with vision
- Gripping deviation correction using vision
- Parts orientation adjustment on the fly with vision
- Parts orientation adjustment on the fly with vision (without master)


Pick & place


Palletizing


Conveyor tracking


Pallet picking using vision

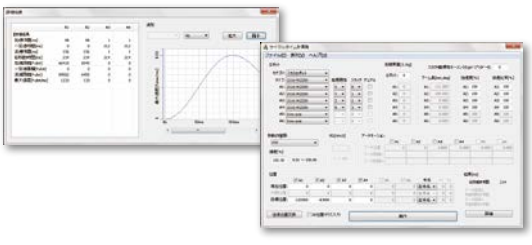

Parts orientation adjustment on the fly with vision


Switching execution program

Program automatic conversion function

Controller program for RCX240 and earlier is converted to that for RCX3 series.

Other functions



All useful features from RCX-Studio Pro are succeeded to help supporting from startup to maintenance.

Cycle time calculator	Real time trace
Data comparison	Custom window creation function

Enhanced expandability

RS-232C and Ethernet ports are provided as standard equipment. A wide variety of high-speed and large capacity field networks, such as CC-Link, DeviceNet™, EtherNet/IP™, and EtherCAT are supported as options. Connections with general-purpose servo amplifier or other company's VISION are easy. So, the RCX320 and RCX340 is called "connectable controller".

Communication between controllers

YC-Link/E

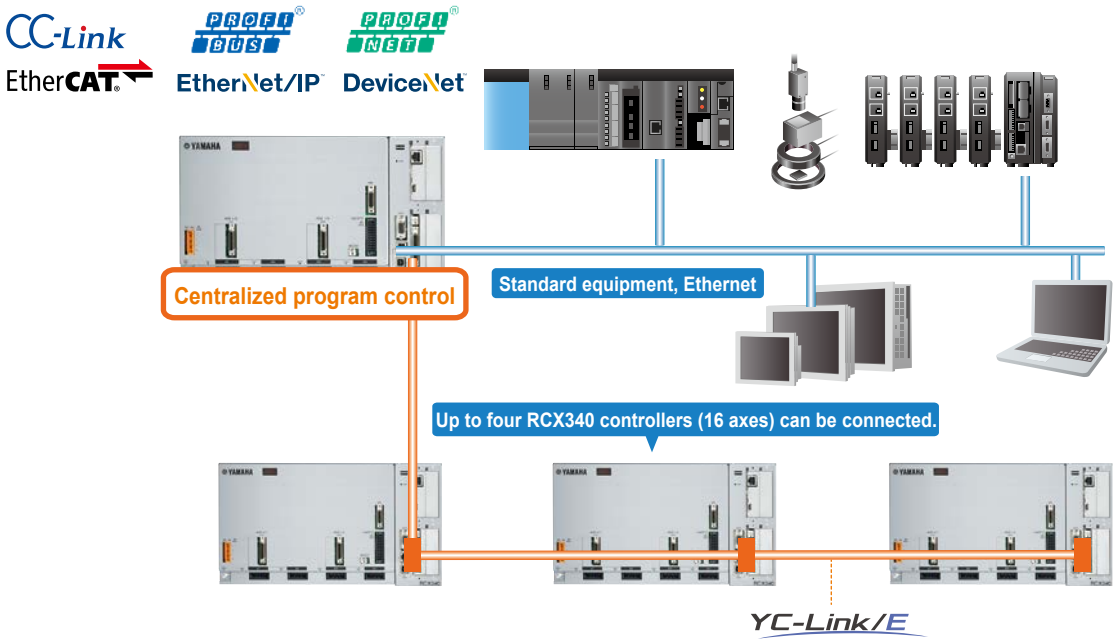
Up to four RCX320 and RCX340 controllers (up to 16 controllable axes) can be connected.

- More flexible robot configuration
- Easy programming
- Centralized control of multiple robots
- Cost reduction

Applicable to various field buses/centralized control of robots through connections of up to four controllers

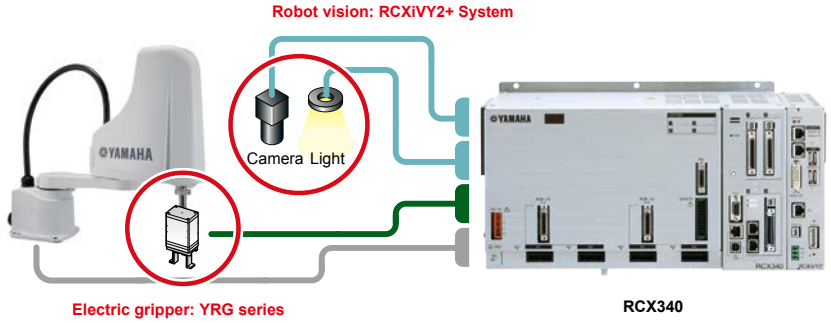
RS-232C and Ethernet ports are provided as standard equipment. Additionally, fulfilling field buses, such as CC-Link, EtherNet/IP™, DeviceNet™, PROFIBUS, PROFINET Note 1, and EtherCAT can be supported to connect and control a wide variety of devices. For 5 or more axes, use of YC-Link/E makes it possible to connect up to four RCX340 controllers so as to perform the centralized control of multiple robots.
Additionally, when using YC-Link/E Note 2, multiple robots can be handled as if they are operated using one controller. This ensures very easy robot programming and management.
Therefore, this robot controller contributes to reduction of unseen costs, such as labor cost necessary for the setup work.

Note 1. Supports PROFINET Ver. 2.2
Note 2. When ordering YC-Link/E, please specify what robot is connected to what number controller.



Applicable to robot vision and electric gripper

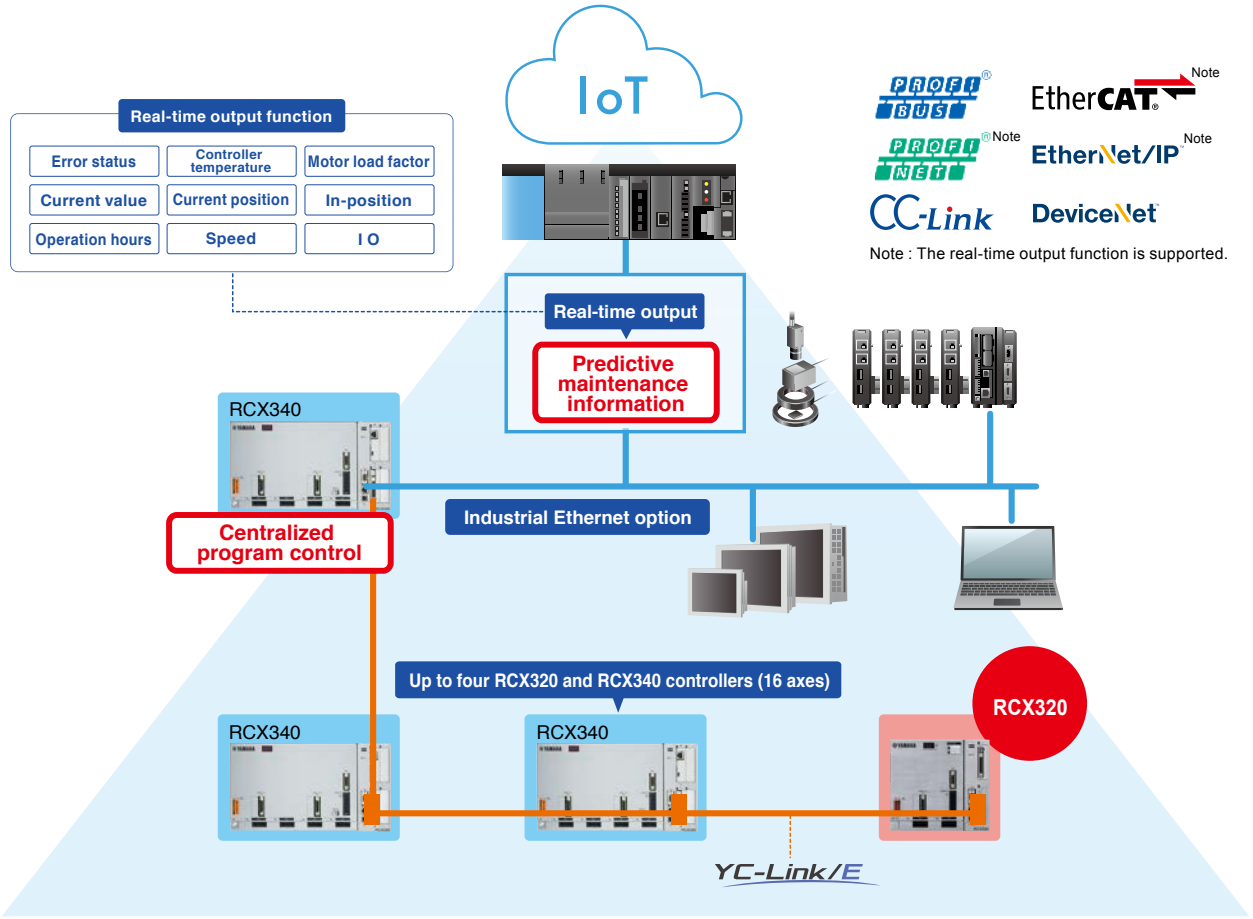
Robot integrated vision "RCXiVY2+" and electric gripper "YRG series" are supported. All control is possible with one robot controller. Data exchanging with the host unit, such as PLC is not needed. The setup or startup is very easy.



Real-Time output function for Preventive Maintenance.

Industrial Ethernet option Real-Time output function

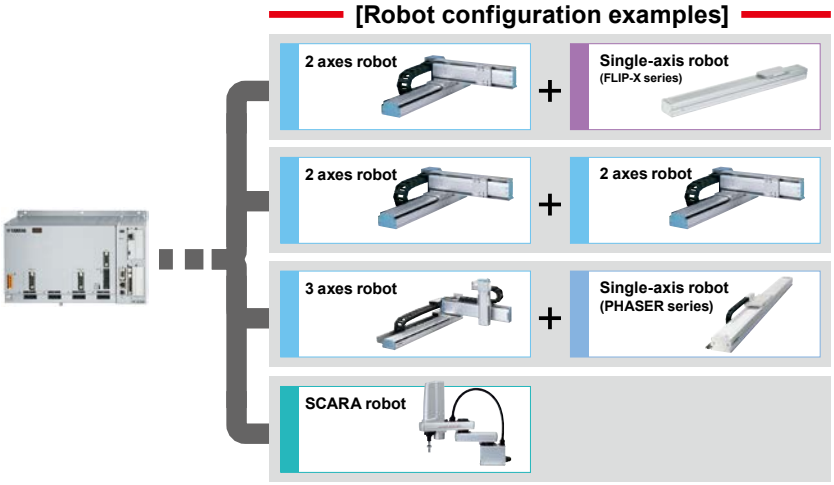
When the industrial Ethernet option (EtherNet/IP, EtherCAT, or Profinet) is selected, the information necessary for the predictive maintenance such as error status, current position, current value, motor load factor, operation hours, and others can be output in real-time to contribute to achievement of the “non-stop production line”.



RCX340 are applicable to all single-axis, Cartesian, SCARA, and P&P robots Note

The 4-axis robot controller RCX340 are applicable to all robot models including single-axis, Cartesian, SCARA, and Pick & Place robots.
As the mixed control of the ball screw type FLIP-X series and linear motor type PHASER series can be performed, the robots can be combined freely according to the applications. Additionally, when preparing the robot controllers for the maintenance work of multiple robots, it is enough to prepare only one robot controller. This robot controller can be used for any model only by changing the setting.

Note. Except for 24 V specification models.

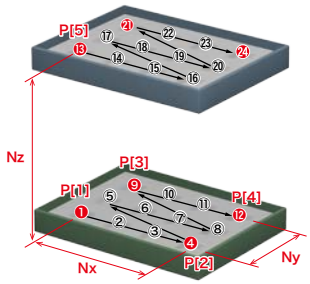


Major features and functions of RCX controller

To palletize.

Function: Palletize

By entering the coordinate values of the four corners on the palette and specifying the number of palettes in the vertical and horizontal directions, the coordinate values of each point are automatically generated. By specifying the coordinate values and the number of palettes in the height direction, a three-dimensional palette is also supported.
The maximum number of pallets that can be defined is 40, but the coordinate values of the four corners and the number of pallets in each direction can be changed by program, so virtually any number of pallets can be supported.



- **Number of pallets that can be used at the same time: 40**
- **2D/3D pallets are supported.**

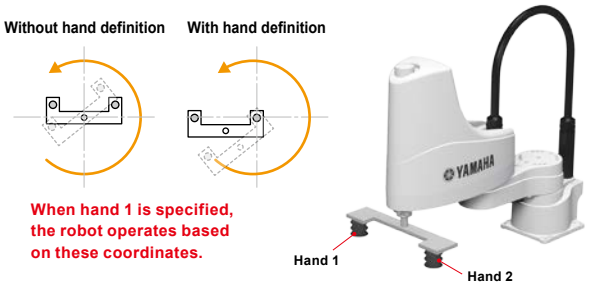
Sample program

PDEF(1)=3,4,2,P3991	···	Defines pallet definition 1 to Nx : 4, Ny : 3, and Nz : 2 using P3991 to P3995.
PMOVE(1,16),S=50	···	Moves the robot to the point at position number 16 of palette number 1 at 50% speed.

To use the tool offset from the tip of the robot.

Function: Hand definition

This function is used to operate the robot based on the coordinates of the offset tool tip when a tool is attached to the tip axis of the robot in an offset state. This function is especially effective when there are multiple hands or when a SCARA robot or a robot with rotation axis rotates around the tool.



- **Number of hands that can be registered: 32**
- **How to specify when there is R axis: 1) Angle based on +X direction**
  2) Hand length
  3) Z-axis offset amount

Sample program

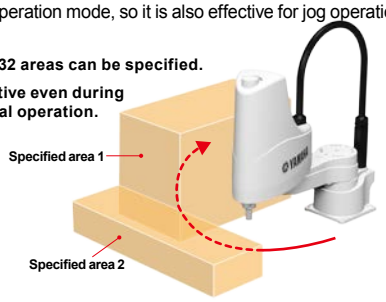
```
HAND H1= 0.000 150.000 0.000 R
HAND H2= -90.000 100.000 0.000 R
P1= 150.000 300.000 0.000 0.000 0.000 0.000
CHANGE H1   ··· Changes the hand data of robot 1 to hand 1.
MOVE P,P1   ··· Moves the tip of hand 1 of robot 1 to P1.
CHANGE H2   ··· Changes the hand data of robot 1 to hand 2.
MOVE P,P1   ··· Moves the tip of hand 2 of robot 1 to P1.
HALT
```

To prevent interference with peripheral devices.

Function: Area judgement output

When the robot enters the pre-registered range, a signal is output to the specified port. This function is useful when there are interfering objects in the equipment to limit the robot operation range or when multiple robots are used in a layout where they interfere with each other. This function operates regardless of the automatic or manual operation mode, so it is also effective for jog operation during teaching.

- **Max: 32 areas can be specified.**
- **Effective even during manual operation.**

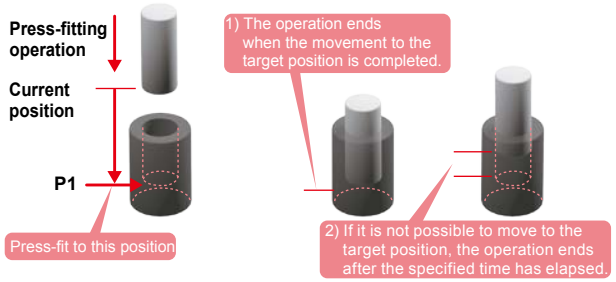


- **Number of areas that can be registered: 32**
- **Functions not only during automatic operation, but also during manual operation.**

To push the workpiece lightly.

Function: Torque limit (PUSH)

It is possible to operate by limiting the motor torque and movement speed when press-fitting a workpiece. If the movement to the target position is not completed even after the specified pressing time has elapsed, the operation stops.



- **Specified by axis.**
- **Pressing force designation: Specified by % to rated thrust.**
- **Pressing time value: 1 to 32767 msec**
- **Pressing speed designation: 1 to 100%**
- **STOPON condition designation: Movement stops when the conditions are met.**

Sample program

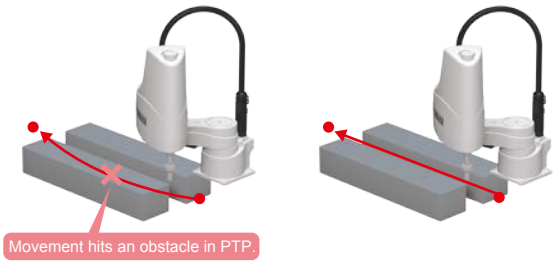
```
PUSH(3,P1),F=20,TIM=5000,S=10
        ··· Moves the 3rd axis to the position specified by P0 under the following
            conditions.
            Pressing force: 20% of rated thrust, Pressing time: 5 sec, Speed: 10%
            * The command ends when the pressing force reaches 20% for 5
              seconds or more.
```

To move along a specified path.

Function: Linear interpolation and circular interpolation (2D/3D)

2D/3D linear and circular interpolation control is possible. This function is effective for sealing work and when you want to specify a path to avoid obstacles.

For PTP movement **For linear interpolation**



- **Linear interpolation and circular interpolation are supported.**

<Option>

- **SPEED: Relative speed designation**
- **DSPEED: Absolute speed designation**
- **VEL: Linear speed designation (Specified in mm/s)**
- **STOPON condition designation: Deceleration stops when the conditions are met.**
- **CONT designation: Connects with next movement command.**
- **Acceleration/deceleration designation**
- **Port output designation: Outputs a signal after moving a specified distance.**

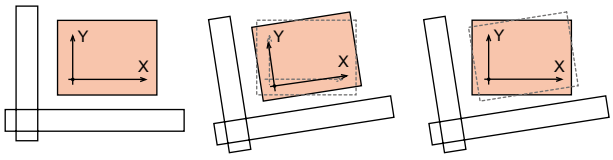
Sample program

```
MOVE L,P20          ··· Linear interpolation movement from the current position to P20
MOVE C,P21,P22,P23,P20  ··· Circular interpolation movement consisting of P21, P22, P23, and P20
MOVE L,P24          ··· Linear interpolation movement to P24
```

To remove the robot, but not to reteach it.

Function: Shift coordinates

A deviation may occur in the coordinate system when re-installing or replacing the robot during maintenance work. In this case, the coordinate system can be corrected using the shift coordinate function. So, the point data can be used as it is. No re-teaching is needed.



- **Number of shifts that can be defined: 40**

Sample program

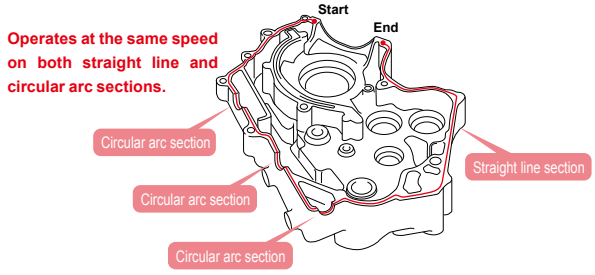
```
S0= 0.000 0.000 0.000 0.000      ··· Defines the shift coordinates of S0.
S1= 100.000 200.000 50.000 90.000  ··· Defines the shift coordinates of S1.
P3= 100.000                      ··· Defines the point data of P3.
SHIFT S0                         ··· Changes the shift coordinates to S0.
MOVE P,P3                        ··· PTP movement to P3.
SHIFT S1                         ··· Changes the shift coordinates to S1.
MOVE P,P3                        ··· PTP movement to P3.
HALT
```

To perform sealing at constant speed.

Function: PATH statement

Sealing requires the path accuracy and constant movement speed. PATH is a function that moves at a specified speed on a path consisting of straight lines and circular arcs, and is suitable for sealing applications because there is little speed fluctuation during movement.

It is possible to change the speed only for a part of the path or output a signal to a specified port at an arbitrary section during movement.



- **Moves at a "constant speed" along a specified path.**
- **After specifying the path in advance with "PATH SET, PATH, PATH END", start the movement with "PATH START".**
- **Up to 1000 points can be specified.**

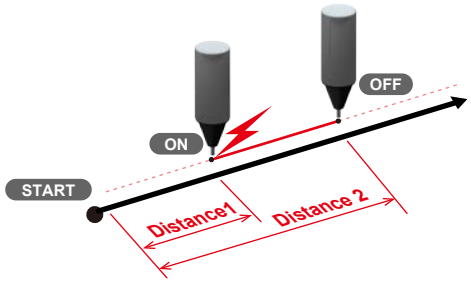
Sample program

```
PATH SET
PATH L,P1,DO(20)=1@10     ··· While moving from the current position to P1 by linear interpolation, set to output "1" to DO(20) at a 10 mm radius position from the start position.
PATH L,P2,DO(21)=1@12.5   ··· While moving to P2 by linear interpolation, set to output "1" to DO(21) at a 12.5 mm radius position from P1.
PATH END
PATH START
```

To output a signal during sealing movement.

Function: Passing point output

For applications such as turning discharge ON/OFF during sealing, general-purpose outputs can be controlled ON/OFF at a specified position without stopping the axis operation during interpolation operation. This function can be used with either the MOVE or PATH command.



- **Up to 3 decimal places can be specified (mm).**
- **Up to two times can be specified in one MOVE statement.**

Sample program

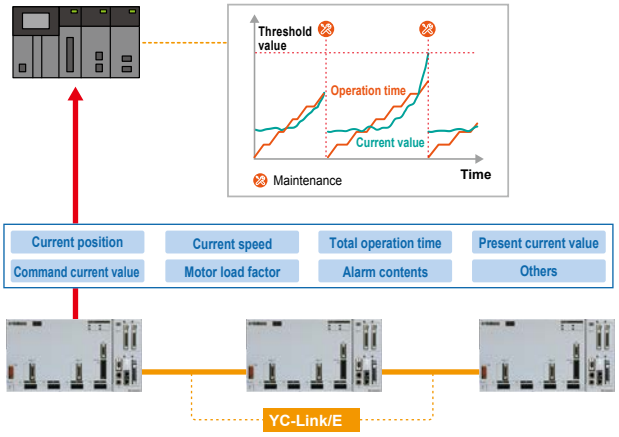
```
A!=10
B!=20
MOVE L,P1,DO(20)=1@A!,DO(20)=0@B!···After starting to move to P1, DO (20) is turned ON at the timing of 10 mm away and DO (20) is turned OFF at the timing of 20 mm away.
```

To output information necessary for predictive maintenance.

Function: Real-time output

Information necessary for predictive maintenance, such as error status, current position, current value, motor load factor, and operation time, can be output in real time.

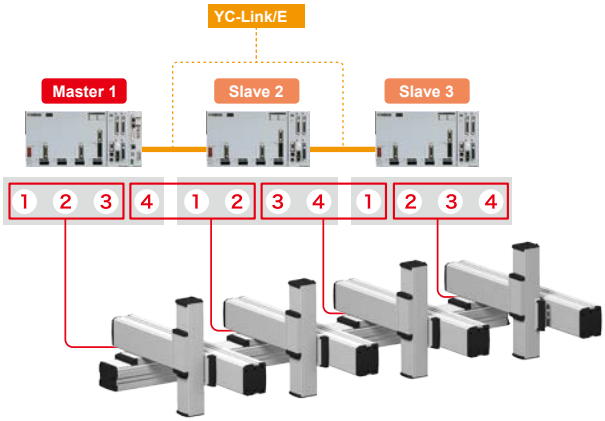
* Industrial Ethernet options (EtherNet/IP, EtherCAT, Profinet) are supported.



To control multiple robots with one controller.

Function: YC-Link/E

Multiple RCX controllers can be linked and controlled by one master controller. Single-axis, Cartesian, and SCARA robots can be mixed, and all network boards and vision units are mounted only on the master controller. Therefore, information on one camera can be shared by multiple robots.



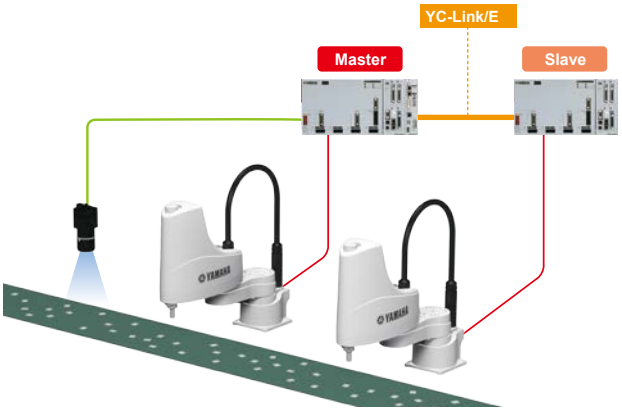
- **Up to 4 controllers can be connected.**
- **When the RCX340 is used, up to 16 axes are supported.**

To operate two robots efficiently.

Function: Multi-task

Multiple tasks (up to 16 tasks) such as robots and peripheral devices can be executed in parallel at the same time. Each task can be prioritized, and the priority can be changed while the task is running.

This is effective for applications such as simultaneously executing vision and robot operations in different tasks during conveyor tracking, and constantly monitoring the workpiece even during robot operation.



- **Number of tasks that can be executed at the same time: 16**
- **Priority: 1 to 64 (high to low)**

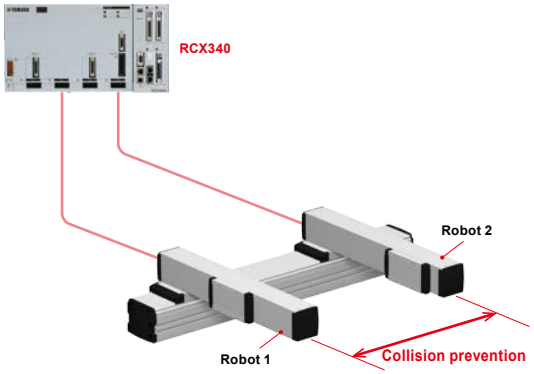
Sample program

```
Program name <TRACK_MAIN>
START<CONV_SCAN>,T2          ···Starts the search task.
*CONVEYOR:
WHILE CCHEKQUE(1)=-1         ···Repeats until no workpiece passes through
                                the work area.
CRMVQUE(1)                   ···Deletes workpiece elements that have
WEND                            passed through the area.
IF CCHKQUE(1)>0 THEN         ···Starts the work when workpiece enters
                                the work area.
(Robot operation routine)

ENDIF
GOTO *CONVEYOR               ···Repeats the routine.

Program name <CONV_SUB>
CTVISION ON(1)               ···Switches to vision use on conveyor 1.
*SCAN:
VSEARCH 1,2,0                ···Performs the search.
IF VGENCNT>0 THEN            ···Process when workpiece is detected.
FOR I%=0 TO VGETCNT-1        ···Adds search results to the position
                                monitoring array.
CADDQUEV 1,VGETPOS(I%),TG=I% ···Adds to the position monitoring queue.
NEXT I
ENDIF
GOTO *SCAN                   ···Repeats the search.
```

To control multiple robots with one controller.

Function: Multiple-robot setting

Each axis of one controller can be distributed and set to multiple robots.
The RCX320 supports up to 2 axes and the RCX340 supports up to 4 axes.
Furthermore, by connecting multiple controllers via YC-LINK/E, up to 4 robots and 16 axes can be set.

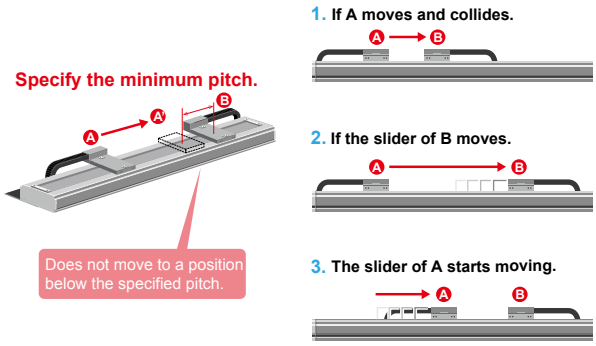


- Each robot can be operated using MOVE [1] to MOVE [4] commands.
- Using multi-task also allows smooth coordination of each robot.

To prevent pallet interference with the double carrier robot.

Function: Collision prevention function

With the double-carrier robot, collision of both carriers is prevented by control in the controller. No zone control or external sensor installation is required.
When a pallet larger than the carrier is mounted, the minimum distance between the carriers can be set using parameters.

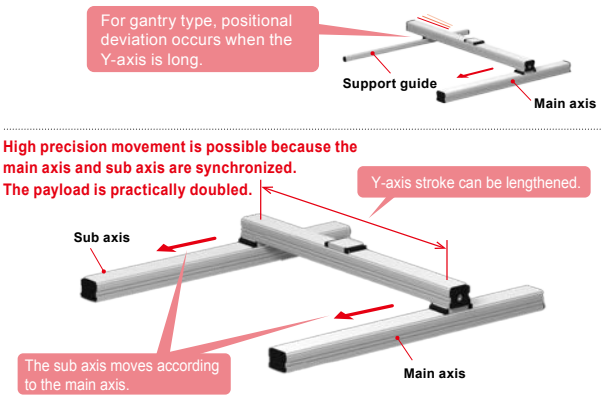


- Support for operating double-carrier robot with RCX (N15, N18, and PHASER series)

To lengthen the Y-axis stroke of the Cartesian robot.

Function: Dual drive

This function synchronously controls two robots of the same type. When the main axis is moved, the sub axis follows in accordance with the movement of the main axis.
This function is effective for transferring heavy objects and supporting the long Y-axis stroke of the Cartesian robot. It is also possible to synchronize two sliders with a double-carrier robot such as a linear motor.

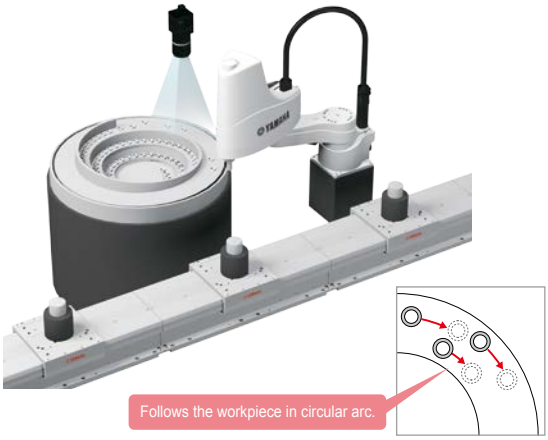


- Rigid dual: The main axis and sub axis are connected with high rigidity.
- Flexible dual: The main axis and sub axis do not have any force interference or are not connected.
- Tandem dual: Two sliders on the same axis are synchronized.

To pick up a workpiece while following a moving object.

Function: Conveyor tracking

Picking can be made by following the movement of the workpiece moving on the conveyor.
Straight line and circular arc tracking is supported. Since the follow-up operation is performed based on the encoder input signal, the follow-up operation is possible even when the conveyor speed fluctuates.
This function supports not only workpieces searched by robot vision, but also tracking by sensor signal input.



- Vision tracking and sensor tracking are supported.
- Number of encoders connected: 2
- Target encoder: Line driver equivalent to 26LS31/26C31
- Maximum response frequency: 2 MHz

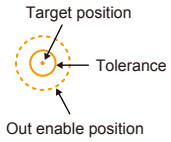
To increase the tact.

Function: Payload setting, arch motion, out enable position

Arch motion is effective for increasing the tact such as pick and place of workpieces. By specifying the linear movement distance when the Z-axis moves up or down, the operation can be performed with the optimal movement pattern.

In addition, increasing the value of the out enable position speeds up the timing for executing the next operation, which has the effect of reducing operation time.

* The robot is automatically set to the optimum acceleration when the payload is set. (Moment of inertia can also be set for SCARA robots.)



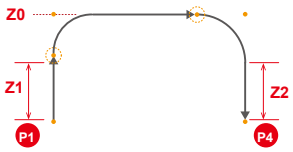
Out enable position:
When the axis tip enters this range, the next operation starts. When passing through relay points to avoid obstacles, etc., the operation time can be shortened by increasing this value.
* The value can be changed using the program.

Normal movement

Normally, P1 to P4 are specified. Each operation starts the next operation when it enters the out enable position range.

```
MOVE P,P2,CONT  ··· Moves from the current position to P2.
MOVE P,P3,CONT  ··· Moves to P3 without stopping when the out enable position is entered.
MOVE P,P4       ··· Moves to P4 without stopping when the out enable position is entered.
```

Arch motion is used.



When the arch motion is operated,
- Only P1 and P4 are specified.
- Z-axis height during movement is specified. (Z0)
- The linear movement distances when ascending and descending are specified. (Z1, Z2)

```
A%=OUTPOS(3)          ··· Assigns the parameter at the out enable position to A%.
OUTPOS(3)=2000        ··· Changes the parameter at the out enable position to 2000.
MOVE P,P4,A3=0.00{50.00,70.00}
                      ··· The A3-axis moves up to 0.00 mm when moving to P4.
                          The A3 axis moves linearly 50.00 mm when ascending
                          and 70.00 mm when descending.
OUTPOS(3)=A           ··· Returns the parameter at the out enable position to the
                          original value.
```

To improve the accuracy.

Function: WAIT ARM, tolerance setting, acceleration setting

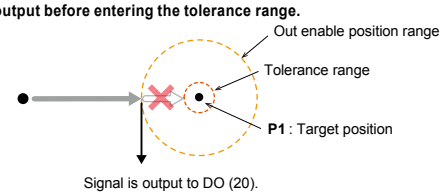
In a normal movement command, the next command is executed when the out enable position is entered. If positioning accuracy during operation is required, use “WAIT ARM” to execute the command after waiting for the position to fall within the tolerance range.

Additionally, since the tolerance range can be changed using the program, it is possible to move with different tolerance for each movement command.

- **WAIT ARM**
  Executes the next command after entering the tolerance range.
- **TOLE**
  Sets/acquires the tolerance parameter.

Normal movement

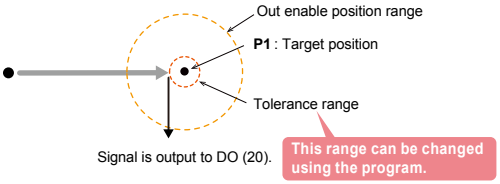
Signal is output before entering the tolerance range.




```
MOVE P,P1       ··· Moves to P1.
DO(20)=1        ··· “1” is output to DO20 when the out enable position is entered.
```

WAIT ARM is used.

Signal is output after entering the tolerance range.




```
MOVE P,P1       ··· Moves to P1.
WAIT ARM        ··· Continues to move until entering the tolerance.
DO(20)=1        ··· “1” is output to DO20 when entering the tolerance range.
```

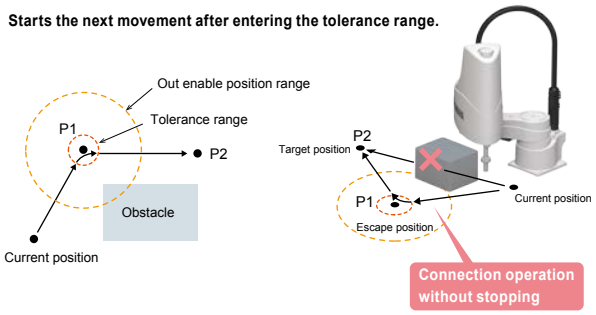
To operate without stopping at the avoidance point

Function: CONT option

When there is an obstacle on the robot movement path and an escape point is set to avoid it, use the CONT option in the movement command to enable smoother movement.
The normal MOVE command performs the positioning at each point, but when the CONT option is used, each movement is linked so that the movement continues without stopping in the middle.

Normal movement

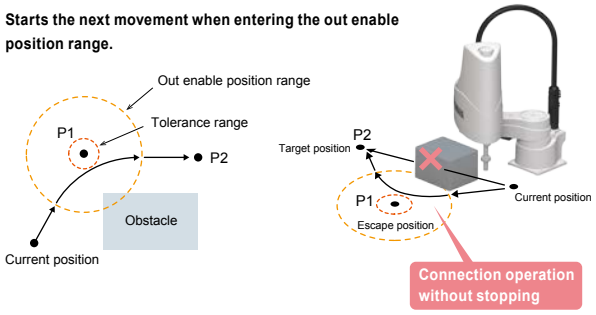
Starts the next movement after entering the tolerance range.



MOVE P,P1 ··· Moves to P1. When the movement axis enters the tolerance range, the movement to P2 starts.
MOVE P,P2

CONT option is used.

Starts the next movement when entering the out enable position range.



For out enable position

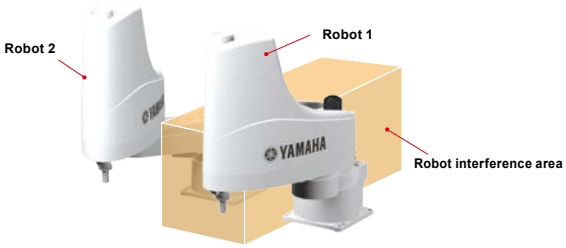
OUTPOS 10000 ··· Changes the OUTPOS parameters of all axes to 10000.
MOVE P,P1,CONT ··· Moves to P1. When the movement axis enters the out enable position range, the movement to P2 starts.
MOVE P,P2

To increase the tact using two robots.

Function: Area judgement output, internal output variable

When two robots are used to transfer a workpiece for tact-up purposes, the area judgment output can be used to ensure that the robots do not interfere with each other. In this case, by using the internal output variables (MI, MO), it is possible to exchange signals at high speed without using the host PLC.

Area judgement output setting



MO(20) ··· ON when robot 1 enters the area.
MO(40) ··· ON when robot 2 enters the area.

Program example

Program name <ROB1_MAIN>

```
START <ROB2_SUB>,T2      ··· Starts the sub task.
MOVE[1] P,P1,A3=0.00     ··· Moves to the standby position.
*LOOP1:
WAIT MO(50,40)=&B10      ··· Waits until robot 2 moves out of area.
MO(30)=0                 ··· Operating flag is OFF.
MOVE[1] P,P3             ··· Moves to the place position.
WAIT ARM[1]
MO(30)=1                 ··· Operating flag is ON.
MOVE[1] P,P2             ··· Moves to the pick position.
WAIT ARM[1]
GOTO *LOOP1
```

Program name <ROB2_SUB>

```
MOVE[2] P,P11,A3=0.00    ··· Moves robot 2 to the standby position.
*LOOP2:
MO(50)=1                 ··· Operating flag is ON.
MOVE[2] P,P12            ··· Moves to the pick position.
WAIT ARM[2]
WAIT MO(30,20)=&B10      ··· Waits until robot 1 moves out of area.
MO(50)=0                 ··· Operating flag is OFF.
MOVE[2] P,P13            ··· Moves to the place position.
WAIT ARM[2]
GOTO *LOOP2
```