HD HYUNDAI ROBOTICS

Warning

All installation works must be performed by a qualified installer and must comply with applicable laws and regulations.

HD
HYUNDAI
ROBOTICS

Hi5aMT230628FMEN2

Hi5a Controller Function Manual

Multitasking

HD HYUNDAI ROBOTICS

HD HYUNDAI ROBOTICS

Printed in Korea – Jun. 2023. 2nd Edition

Table of Contents

1. Overview 1-1

1.1. About the Multitasking Function 1-2
1.2. Description of Terms 1-5

2. Related Functions 2-1

2.1. Statements 2-2
2.1.1. TaskStart 2-2
2.1.2. TaskWait 2-4
2.1.3. TaskSync 2-5
2.2. Monitoring 2-6
2.2.1. Creation of Subtask 2-6
2.2.2. Select the Current Task 2-7
2.2.3. Multitasking Status 2-8
2.3. Creation of Subtask 2-9
2.3.1. Auto Creation 2-9
2.3.2. Manual Creation 2-9
2.4. Termination of Subtask 2-10
2.4.1. Auto Termination 2-10
2.4.2. Manual Termination 2-10
2.5. Task Switching 2-11
2.5.1. Robot Program 2-11
2.5.2. Output Signal Assignment Setting Dialogue Box 2-11
2.6. Program Selection 2-12
2.6.1. Selecting in Main Task 2-12
2.6.2. Selecting in Subtask 2-12
2.7. Step Forward/Backward 2-13
2.8. Handling of Start 2-13
2.9. Handling of Stop 2-13
2.10. Start/Stop Lamp 2-13
2.11. Multitasking Program for Additional Axes 2-15
2.11.1. Overview of the Multitasking Motion 2-15
2.11.2. Example of Teaching the Multitasking Motion 2-16
2.11.3. Cautions in Teaching the Multitasking Motion 2-18

List of Figures

Figure 1.1 Single-tasking structure 1-2
Figure 1.2 Single-tasking structure 1-3
Figure 1.3 Creation of Subtasks in Multitasking Structure 1-4
Figure 2.1 Example of the TaskStart Command Use 2-2
Figure 2.2 Example of creating a subtask and waiting for its termination 2-3
Figure 2.3 Display of the Creation Status of a Subtask 2-6
Figure 2.4 Multitasking Status Monitoring Window 2-8
Figure 2.5 Manual Creation of Subtask 2-9
Figure 2.6 Manual Termination of Multitask 2-10

List of Tables

Table 1-1 Description of the Terms of Multitasking 1-5

1

Overview

1.1. About the Multitasking Function

The Hi5a controller can perform an independent execution of each of eight programs (JOB files) simultaneously. Such multitasking control, performed through this independent operation method, is called the “multitasking function.”

Using the multitasking function makes it possible to execute a program to control another device while executing a robot control program simultaneously. Through this, the control of the robot and other devices can be performed independently of each other. If necessary, cooperative work is possible with the robot and other devices synchronized with each other, providing an advantage in executing complicated and difficult application works.

The figure below shows the existing single-tasking structure (previous Hi5 controllers of version 30.00-00 series). In this structure, only one task exists, making it impossible to perform an independent execution of two or more programs simultaneously. In comparison to the multitasking structure to be described later, you can think that there is only a main task without a subtask. Please refer to Figure 1.1 below.

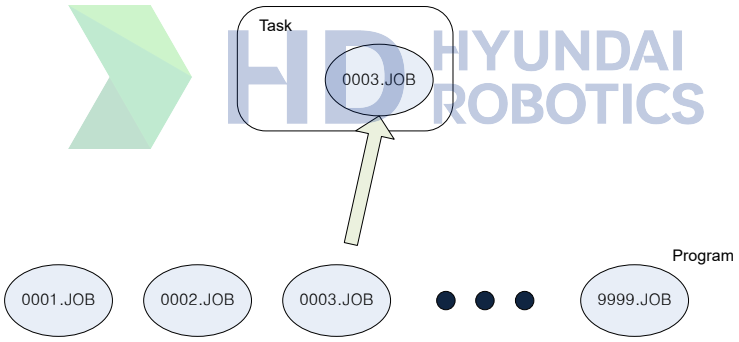


Figure 1.1 Single-tasking structure

Figure 1.2 below shows the multitasking structure supported by Hi5a. As up to eight tasks can be performed simultaneously, it is possible to perform an independent execution of each of eight programs simultaneously by assigning one program (JOB file) to each task. It is possible to perform an independent control of each of multiple devices by performing eight tasks simultaneously. However, when it comes to controlling a robot, only one robot can be controlled in the main task. When performing a synchronized work using multiple robots simultaneously, it is necessary to use the cooperative control function that our company provides.

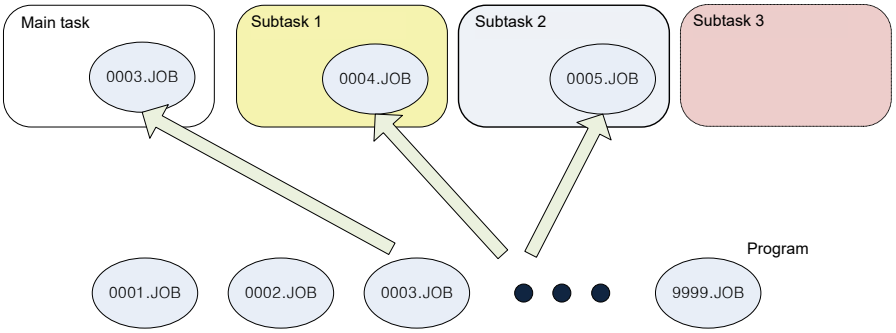


Figure 1.2 Multitasking structure

The names of the tasks executing the programs are as follows.

- Main task
- Subtask 1
- Subtask 2
- Subtask 3
- Subtask 4
- Subtask 5
- Subtask 6
- Subtask 7

The main task will be always created and exist by default to execute a JOB program. Subtask 1, subtask 2, and subtask 7 (hereinafter "subtask 1/2/3/4/5/6/7") can be created and terminated as necessary. A subtask will be created automatically when the TaskStart statement is executed in the main task program or when it is executed in the subtask program created in the main task. In other words, it is possible to give a command to create a subtask even in an operating subtask. Refer to Figure 1.3 for the subtask creation structure. The subtasks will be terminated automatically when the END statement in each subtask program is executed.

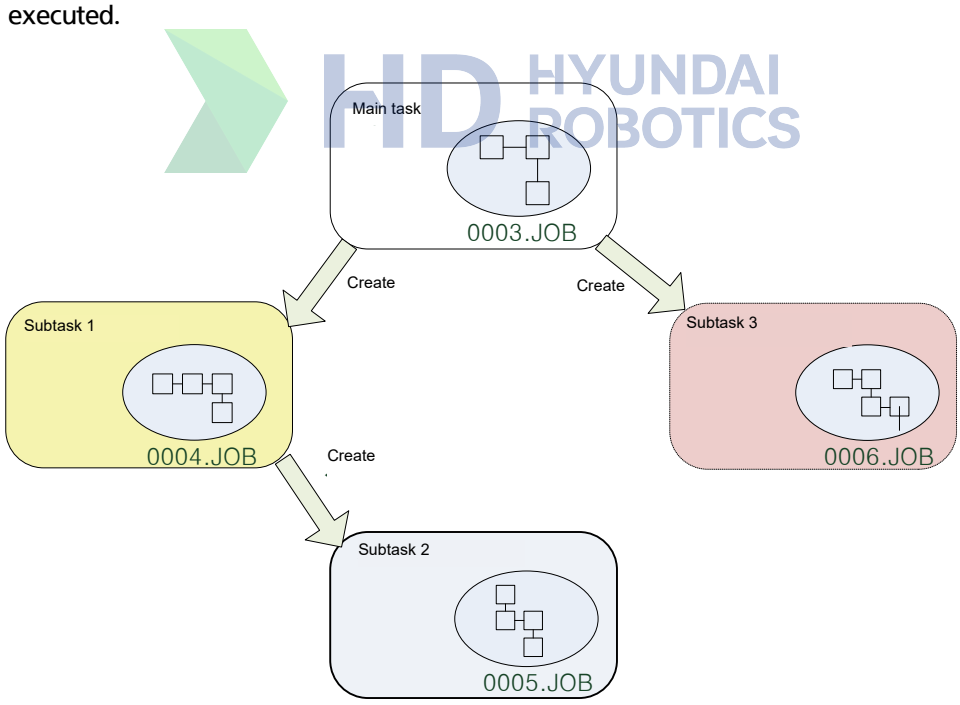


Figure 1.3 Creation of Subtasks in Multitasking Structure

1.2. Description of Terms

The table below describes the terms used in this manual.

Table 1-1 Description of the Terms of Multitasking

Terms	Description
Program (JOB file)	- A work program stored in the storage memory of the controller. (Example: Refers to a JOB file stored in the controller, such as 0001.JOB, 0002.JOB, and 1001.JOB)
Main Task Subtask 1/2/3/4/5/6/7	- A work program executor of the robot controller that can load and execute a work program. - There are eight tasks in total in the robot, and each task can load and execute only one program at a time.
Main task program Subtask 1/2/3/4/5/6/7 program	- Refers to a specific work program assigned to a task for execution. (Example: If 001.JOB is loaded in the main task, the main task program will be 001.JOB.) - A program can be executed only when it is assigned to the main task or subtask.

HD
HYUNDAI
ROBOTICS

HYUNDAI ROBOTICS

2 Related Functions

2.1. Statements

2.1.1. TaskStart

The TaskStart statement creates a subtask, assigns a specific JOB program to the subtask, and starts the subtask program. TaskStart can also create a subtask in the main task or create another subtask in a subtask.

The format of the TaskStart statement is as follows. You can input the TaskStart command by selecting 『[F6]: Command input』 → 『F3: Others』 → 『TaskStart』 in sequence.

TaskStart SUB=〈Subtask Number〉,JOB=〈Program Number〉

Item	Content
Subtask Number	Designate the task number to be created (1–7)
Program Number	Designate the program to be executed in a created subtask (1–9999)
Example of Use	TaskStart SUB=1,JOB=0010 (Execute by assigning 0010.JOB to subtask 1)

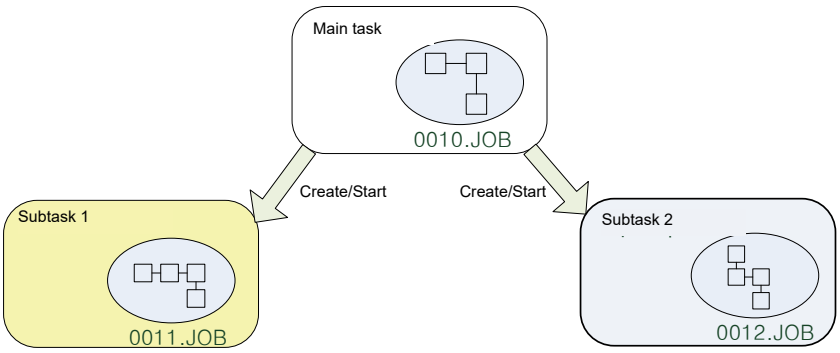


Figure 2.1 Example of the TaskStart Command Use

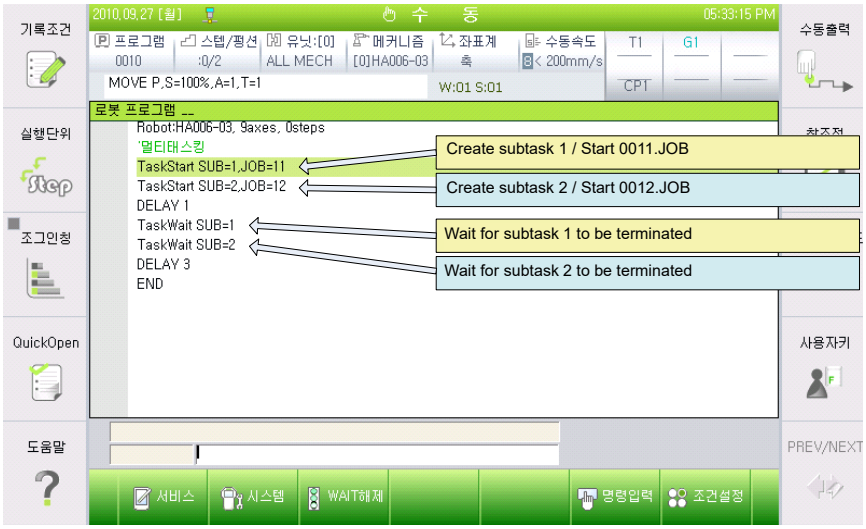


Figure 2.2 Example of creating a subtask and waiting for its termination

What requires attention is that the number of the subtask to be created is a different subtask number and not its own number of the current subtask. In addition, the TaskStart statement is inapplicable in the error situations described below, so it is necessary to take precaution.

Firstly, in a situation where the subtask created using TaskStart has been already created and in the middle of being executed and has not been completed, if the same program or another program is assigned to the same subtask as a subtask program, an error will occur. Please refer to the examples below.

```
TaskStart SUB=1, JOB=0011
PRINT #0,"Main task"
TaskStart SUB=1, JOB=0011
END
```

〈Error because of a re-execution in the middle of a subtask execution〉

```
TaskStart SUB=1, JOB=0011
PRINT #0,"Main task"
TaskStart SUB=1, JOB=0012
END
```

〈Error because of the assignment and execution of another program in the middle of a subtask execution〉

Secondly, an error will also occur if the same program is assigned to multiple subtask programs simultaneously. Please refer to the examples below.

```
TaskStart SUB=1, JOB=0011
PRINT #0,"Main task"
TaskStart SUB=2, JOB=0011
END
```

〈Error because of setting the same program to a different subtask〉

Thirdly, an error will occur when creating the main task program (JOB file currently being executed in the main task) as a subtask program. In other words, it is impossible to assign the same JOB file in the main task and subtask.

```
"An error will occur when the current main task program is 0011
" and TaskStart is executed in the subtask as shown below.
TaskStart SUB=1, JOB=0011
END
```

〈Error because of setting the same program in the main task and subtask〉

2.1.2. TaskWait

The TaskWait statement makes it possible to wait for a subtask termination. Normally, the termination of a subtask will be automatically handled through the END statement execution in that subtask program. This statement can also be used to perform the next action after waiting for another subtask to be completed while a subtask is being performed.
The format of the TaskWait statement is as follows:

```
TaskWait SUB=〈Subtask Number〉
```

Item	Content
Subtask Number	Designate the number of the task to be terminated (1–7)
Example of Use	TaskWait SUB=1 (Wait for subtask 1 to be terminated)

The TaskWait command can be used in the main task to wait for a subtask to be terminated and in a

subtask to wait for another subtask to be terminated.

2.1.3. TaskSync

The TaskSync statement synchronizes between tasks. Generally, synchronization is essential for a cooperative work between two or more robots. In this case, this statement can be conveniently used when matching the synchronization start time between tasks. This statement can be useful when the main task and a subtask are conducting tasks and want to start the tasks simultaneously after waiting at a point of executing a specific statement.
When comparing it with TaskWait, TaskSync is a function used to synchronize between the main task program and a subtask program in the middle of those programs being executed and TaskWait is a function for waiting for a subtask to completely end.

The format of the TaskSync statement is as follows:

TaskSync ID=〈Identifier〉,NO=〈Number of Executions Performed for the Same ID〉

Item	Content
Identifier	Designate the identifier to 1–32. It can designate multiple synchronization points in the program. It is a variable that differentiates a synchronization point.
Number of Tasks to be Synchronized	Designate the number of tasks to be synchronized to 2–8. It should match the number of tasks executed through the input of the TaskSync command.
Example of Use	Main task program TaskStart SUB=1, JOB=0011 PRINT #0,"Main task" TaskSync ID=1,NO=2 PRINT #0,"Task sync" (The task ID is one, and the number of tasks to be synchronized is two)
	Subtask program 1 PRINT #0,"Sub task" DEALY 1 TaskSync ID=1,NO=2 (The task ID is one, and the number of tasks to be synchronized is two)

2.2. Monitoring

2.2.1. Creation of Subtask

The creation status of a subtask can be checked on the title frame. The number of the created subtask is displayed on the title frame. As the main task always exists, it is not displayed separately. In Figure 2.3, the numbers 1 and 2 displayed on the title frame indicate the created subtasks 1 and 2.

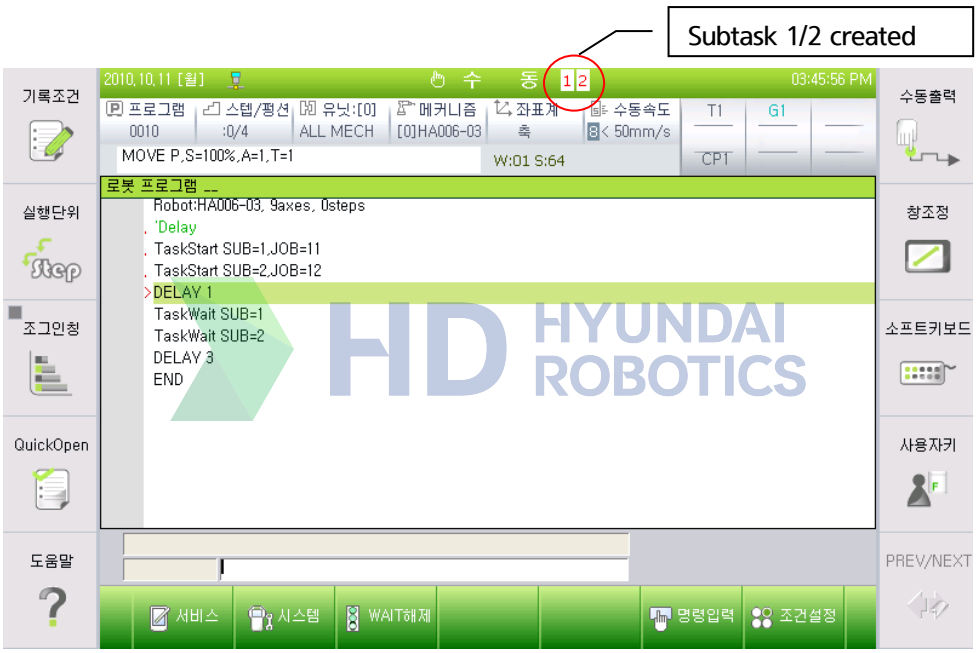
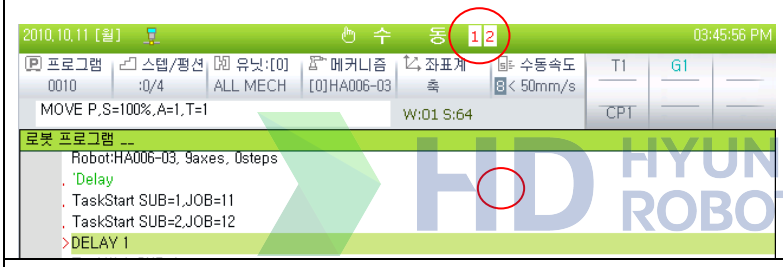
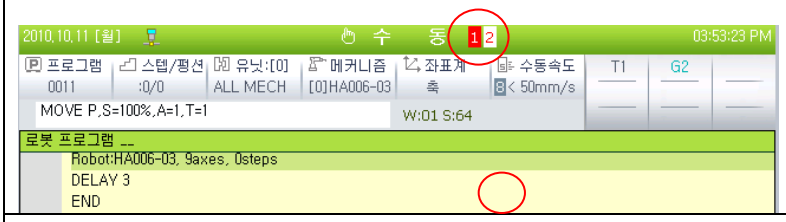
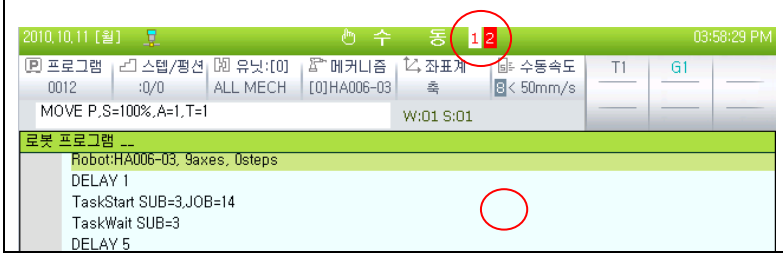


Figure 2.3 Display of the Creation Status of a Subtask

2.2.2. Select the Current Task

The current selection status of a task can be identified from the color on the title frame and the robot program window. Switching between tasks can be performed using [CTRL]+[▶] keys or [CTRL]+[◀] keys. For details on how to switch between tasks, refer to the “Task Switching” chapter in this manual. While a subtask is currently being executed, you can select the subtask to view its progress, but you cannot edit it. To edit a program, it is necessary to stop executing the task and call the program that needs to be edited from the main task window.

Status	Content
Main Task Selected 	– The subtask 1/2/3/4/5/6/7 creation status is set to red text and white background. – The background of the robot program window is white.
Subtask 1 Selected 	– The subtask 1 creation status is changed to white text and red background. – The background of the robot program window is light yellow.
Subtask 2 Selected 	– The subtask 2 creation status is changed to white text and red background. – The background of the robot program window is light blue.

2.2.3. Multitasking Status

In 『[F1]: Service』 → 『1: Monitoring』 → 『18: Multitasking status』, you can check the current program number, the number of the step in progress, the number of the function in progress, the step forward/backward operation status, and the work status of all tasks.
The multitasking status will be displayed in a table classified based on each function and task type. In the monitoring table, the horizontal lines from top to bottom indicate the program assigned to the task, the number of the step being executed in each task, the line number of the JOB program being executed, the step forward/backward operation status, and the current work status. The vertical lines indicate the main task and subtasks 1/2/3/4/5/6/7 from left to right.
The information is configured so the status of each task is reflected in the table in real time according to the step forward and backward operation status, making it possible to easily understand the status of the task.

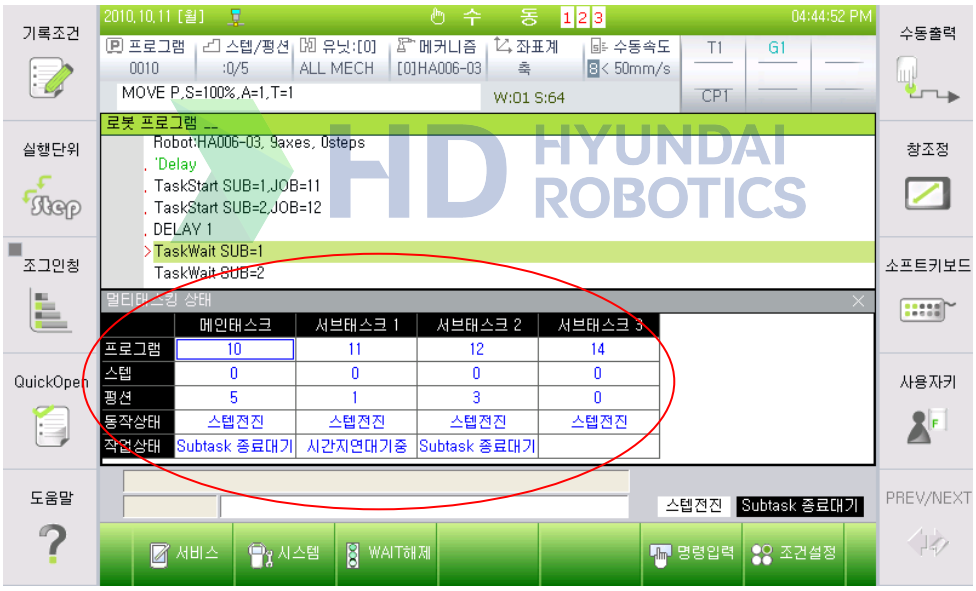


Figure 2.4 Multitasking Status Monitoring Window

2.3. Creation of Subtask

2.3.1. Auto Creation

In the main task program or subtask program, the desired subtask number and subtask program will be assigned through the execution of the TaskStart statement, automatically creating a subtask.

2.3.2. Manual Creation

During the manual creation of a multitask, the user directly selects a subtask by operating the teach pendant without using the TaskStart statement, assigns the desired program as the subtask program, and executes it. The manual creation procedure is as follows.

『[F1]: Service』 → 『1: Monitoring』 → 『18: Multitasking status』 → Status monitoring window opens → Move to and select the desired subtask number → Select the program window → [F5]: Select] → Input the program number

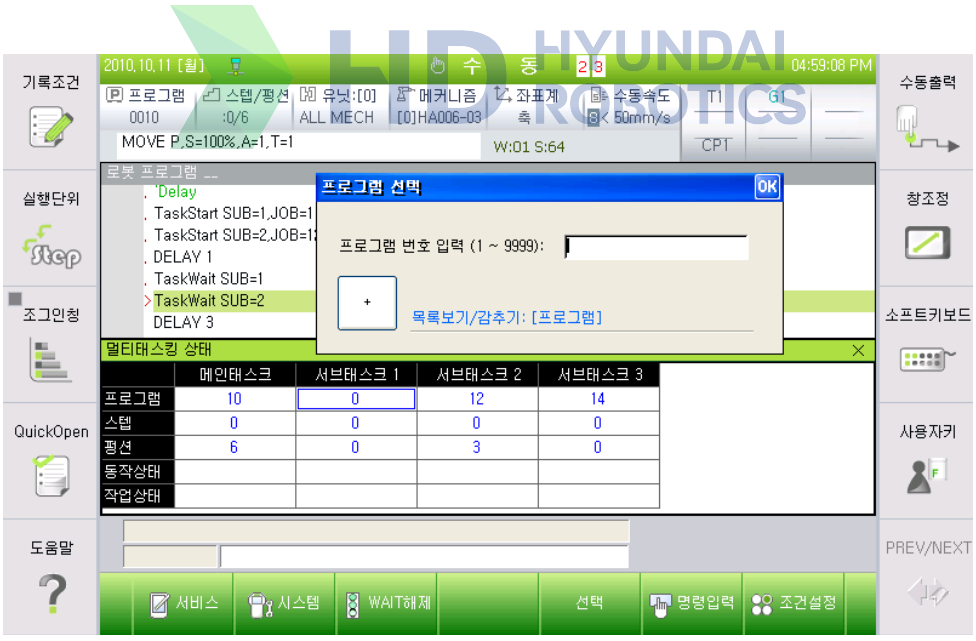


Figure 2.5 Manual Creation of Subtask

2.4. Termination of Subtask

2.4.1. Auto Termination

When the END statement is executed in the subtask program, the subtask will be automatically terminated. Note that even when the main task program encounters the END statement causing the main task to end, if the subtask is being executed and the END statement of the subtask program is not reached yet, the subtask will continue to operate.

2.4.2. Manual Termination

The manual termination and clearing of a multitask can be carried out by assigning 0 to the multitask program number in the monitoring window. The procedure is as follows.

『[F1]: Service』 → 『1: Monitoring』 → 『18: Multitasking status』 → Status monitoring window opens → Move to and select the desired subtask number → Select the program window → [F5]: Select] → Input the program number “0”

In addition, if you change the program of the main task in the manual operation mode, the operation of all subtasks will be stopped and terminated.

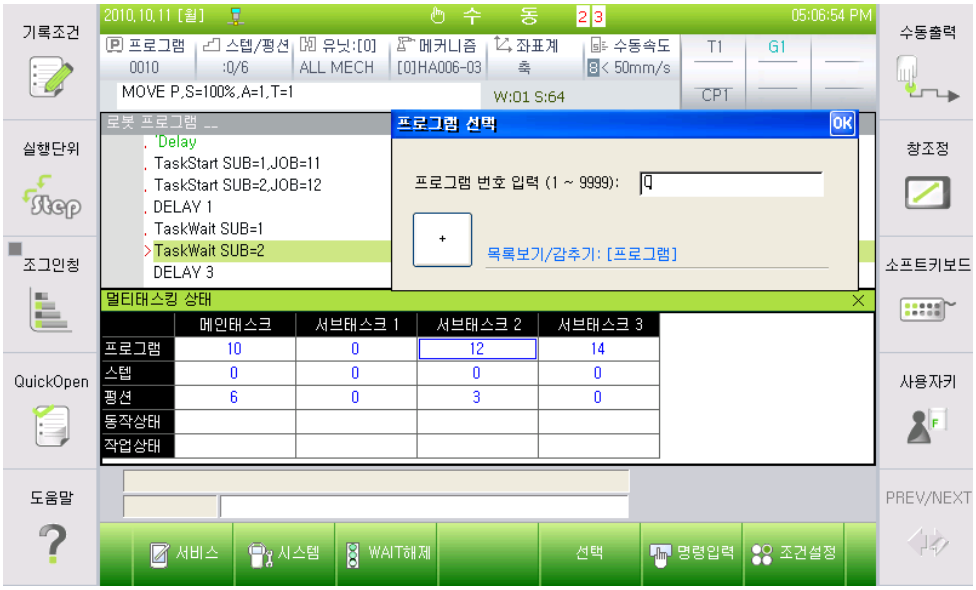


Figure 2.6 Manual Termination of Multitask

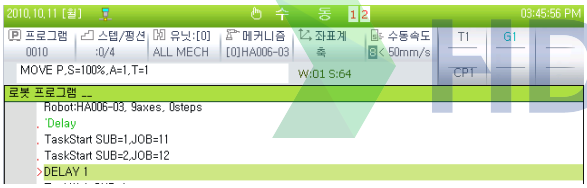
2.5. Task Switching

2.5.1. Robot Program

Task switching can be performed by operating the keys as shown in the table below. Task switching can only be performed between created tasks. Switching to a task that has not been manually or automatically created cannot be performed.

Operation	Content
[CTRL]+[▶] keys	Switch to the next task.
[CTRL]+[◀] keys	Switch to the previous task.

[Main Task]



[Subtask 1]

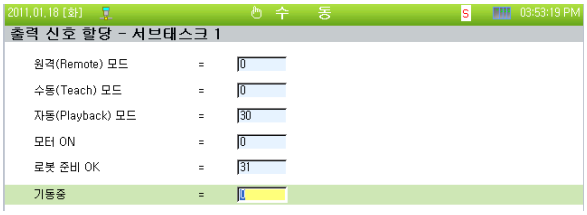
2.5.2. Output Signal Assignment Setting Dialogue Box

The output signal assignment for each task can be performed with the following menu selection procedure. This setting can be carried out even when no subtask is currently assigned.

『F2: System』 -〉 『2: Control Parameter』 -〉 『2: Input/Output Signal Setting』)

[Main Task]

[Subtask 1]



2.6. Program Selection

If a task program is selected by operating the [SHIFT]+[Program] keys, the status and operation of the task will be changed depending on the situation as follows.

2.6.1. Selecting in Main Task

If a program other than the main task program currently being executed in the main task is selected, the operation of all created subtasks will be stopped and terminated.

Type of task	Content of operation
Main task	Change of programs Clearing of step numbers and function numbers
Subtask	Clearing of program numbers Clearing of step numbers and function numbers

2.6.2. Selecting in Subtask

When a program is selected in a subtask, the program assigned to the subtask program will change to a new program.

Type of task	Content of operation
Main task	No change
Subtask	Change of program numbers Clearing of step numbers and function numbers

2.7. Step Forward/Backward

If you want to perform a step forward/backward operation for all created tasks simultaneously or only for the currently selected task, you can use the keys listed in the table below. When the step forward/backward key is selected, the step operations for the main task and subtask will be as shown in the table below.

Operation	Content
[FWD]/[BWD] key	To execute the forward/backward operation for all created tasks simultaneously
[CTRL]+[FWD]/[BWD] key	To execute the forward/backward operation only for the currently selected task

2.8. Handling of Start

To execute a task, activate “MOTOR ON” or “START” in the auto execution mode, or activate “MOTOR ON” in the manual execution mode and then select the [FWD] key. In addition, it is also possible to execute subtasks concerning external signals through an independent execution of the statement.

2.9. Handling of Stop

During the multitask operation, if the T/P maintenance button is pressed or an external stop signal is inputted, all created tasks and the task currently displayed on the screen will be stopped and terminated.

2.10. Start/Stop Lamp

The start/stop lamp of the teach pendant displays the multitask operation status. According to the multitask operation status, the start lamp and stop lamp perform the on/off operation. The on/off operation according to the multitask operation status will be as shown in the table below.

Operation	Content
Start lamp on / Stop lamp off	When at least one task is in operation
Start lamp off / Stop lamp on	When all tasks are stopped

In other words, when all, some, or even one task is in operation, the status will be “Start lamp on / Stop lamp off.” Only when all tasks are stopped will the status be “Start lamp off / Stop lamp on.” The start lamp and stop lamp perform the on/off operation in a mutually exclusive manner when displaying the multitask operation status.

2.11. Multitasking Program for Additional Axes

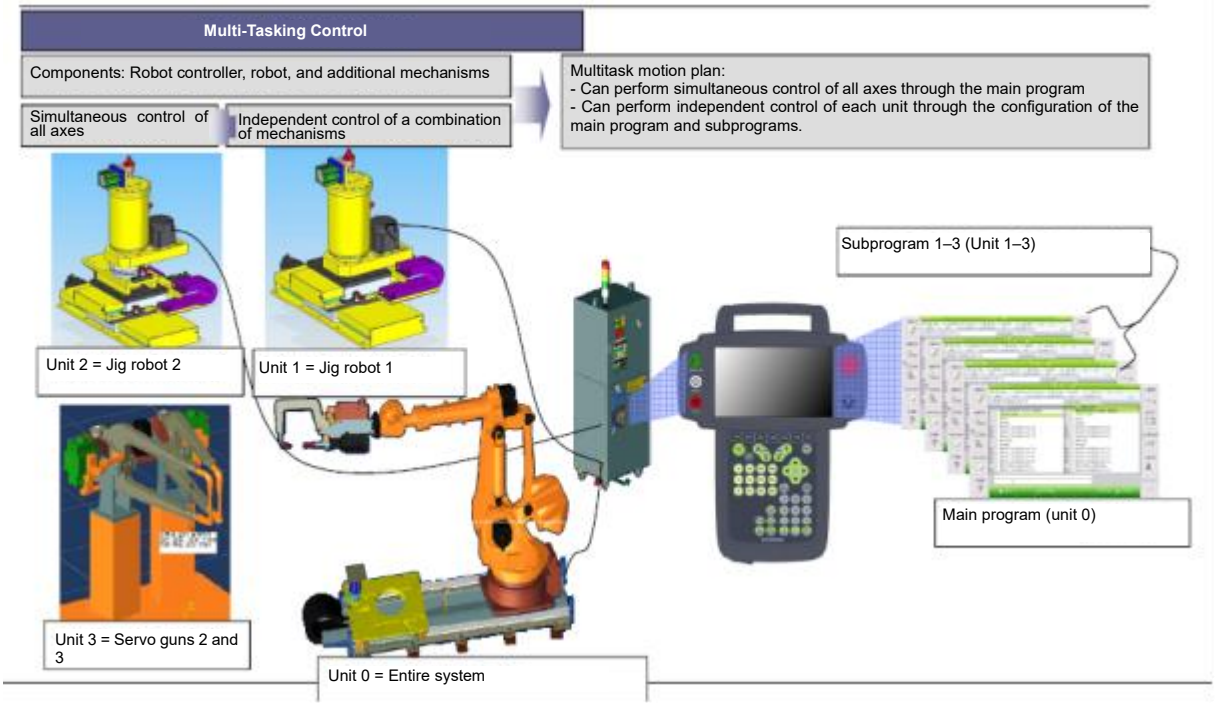
2.11.1. Overview of the Multitasking Motion

If multitasking is applied, it is possible to configure an independent program by assigning the program controlling the additional axis as a subtask program. As the figure below shows, individual subtasks can independently drive additional axes by designating units so they do not overlap with each other. The main task can control both the robot and the assigned additional axes, but the subtask can only control the assigned axes.

To use this function, it is necessary to set the unit and mechanism and use the AXISCTRL command. A brief definition is given below.

- A mechanism is a combination of axes (robot axis, additional axis) configured in one set, and it is possible to select one mechanism with a jog and perform the operation based on each mechanism.
- A unit is a selective combination of mechanisms, and the difference between a unit and a mechanism is that when steps are recorded in a work program, they will be recorded based on each unit.
- AXISCTRL is a statement about the setting of the additional axes control.

For a more detailed explanation and about the methods of setting the mechanisms and units, please refer to the relevant contents of “Hi5a Controller Operation Manual”.

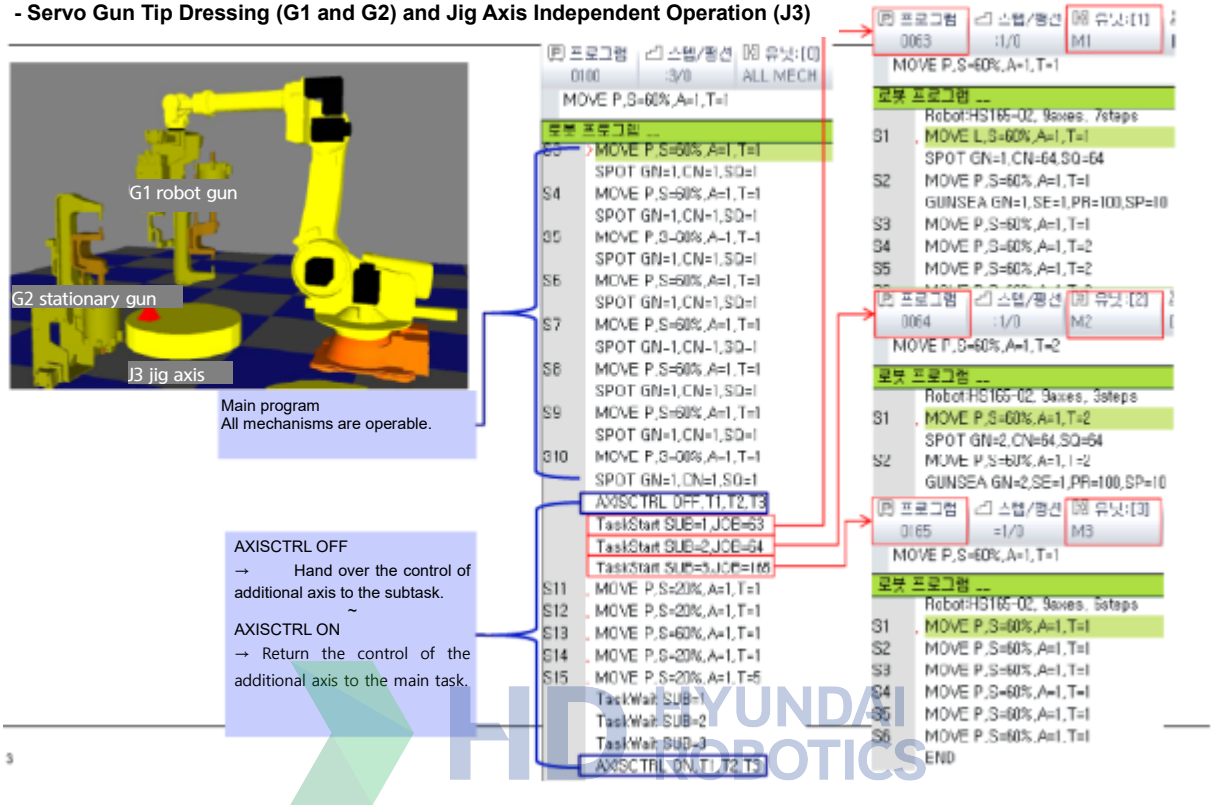


2.11.2. Example of Teaching the Multitasking Motion

As shown in the figure below, here is an explanation about an example of performing a multitasking operation with a system that consists of a robot, a robot gun, a stationary gun, and a jig axis, which totals to one robot axis and three additional axes.
The main task controls the robot axis, and each of the three subtasks controls one additional axis. In the main task, the work for the robot and the creation of subtasks are performed. When created, each subtask performs the work for the unit.

① The main program is program number 100. The main program can be programmed so that all axes can be driven by selecting unit 0. When the main program is executed, the MOVE and SPOT statements will be executed in sequence, and the statements will be executed up to step 10.
② After that, if AXISCTRL OFF T1, T2, and T3 commands are executed in the main program, the setting will be performed so that the designated axis number can be controlled in the subtask. In this way, the T1, T1, T3 additional axes will not move to the positions of T1, T2, and T3 recorded in steps 11–15 when steps 11–15 are performed in the main task.
③ Designate the program to be executed in subtasks 1, 2, and 3 using the TaskStart command. The program assigned to a subtask must be set as an independent unit, and the axis designated as a unit should not be overlapped between subtasks. For example, mechanism M1 = axis T1 should be designated for unit 1, mechanism M2 = axis T2 for unit 2, and mechanism M3 = axis T3 for unit 3 in a mutually exclusive manner.
④ Programs 0063, 0064, and 00165 assigned to subtasks by the TaskStart command will be independently executed and the main program will execute steps 11–15.
⑤ When a TaskWait command is encountered in the main program, waiting will occur until the designated subtask is completed and END is executed. Accordingly, the main program will wait until all sub tasks 1, 2, and 3 are ended.
⑥ For the main program to take back the control of the additional axis, the AXISCTRL ON command should be executed.

- Servo Gun Tip Dressing (G1 and G2) and Jig Axis Independent Operation (J3)



2.11.3. Cautions in Teaching the Multitasking Motion

1. When it comes to a subprogram, it should be registered by **setting a mechanism** for the axes for which AXISCTRL OFF has been performed and then designate them as a unit.
2. The **units** between subprograms should be **set in an exclusive manner** to ensure that mechanisms will not overlap between units.
3. When the motion is recorded as MOVE L and MOVE C in the subprogram, it is necessary to take precautions because the additional axes will operate at the highest speed. Considering that, **the motion should be recorded as MOVE P.**
4. When recording is performed in the subprogram based on the speed unit of mm/s, the additional axis for which the recording is performed may operate at the highest speed, so be careful. Considering that, recording should be performed in the subtask program based on **the speed unit of % or sec**.

● **Daegu Office (Head Office)**

50, Techno sunhwan-ro 3-gil, yuga, Dalseong-gun, Daegu, 43022, Korea

● **GRC**

477, Bundangsuseo-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, Korea

● **ARS : +82-1588-9997 (A/S center)**

● **E-mail : robotics@hyundai-robotics.com**