

MICREX-SX How-to Guide

Issue No.	GE0186b	Issue date	June 2018	Distribution class	General
-----------	---------	------------	-----------	--------------------	---------

Fuji Electric Co., Ltd.

How to Create Multi-CPU Project with SX-Programmer Standard

1. Overview

This guide describes how to create a project of multi-CPU configuration using SX-Programmer Standard.

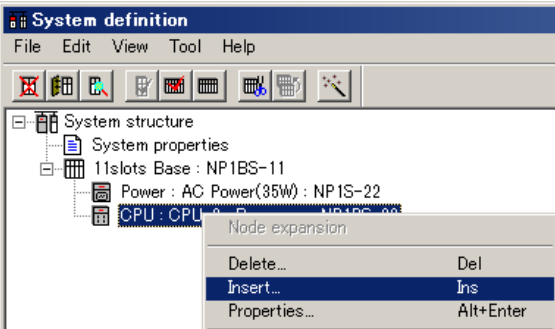
In SX-Programmer Standard, one project controls one resource. Therefore, for a multi-CPU system, you need to create projects as many as the CPUs to be used.

2. When creating a new project

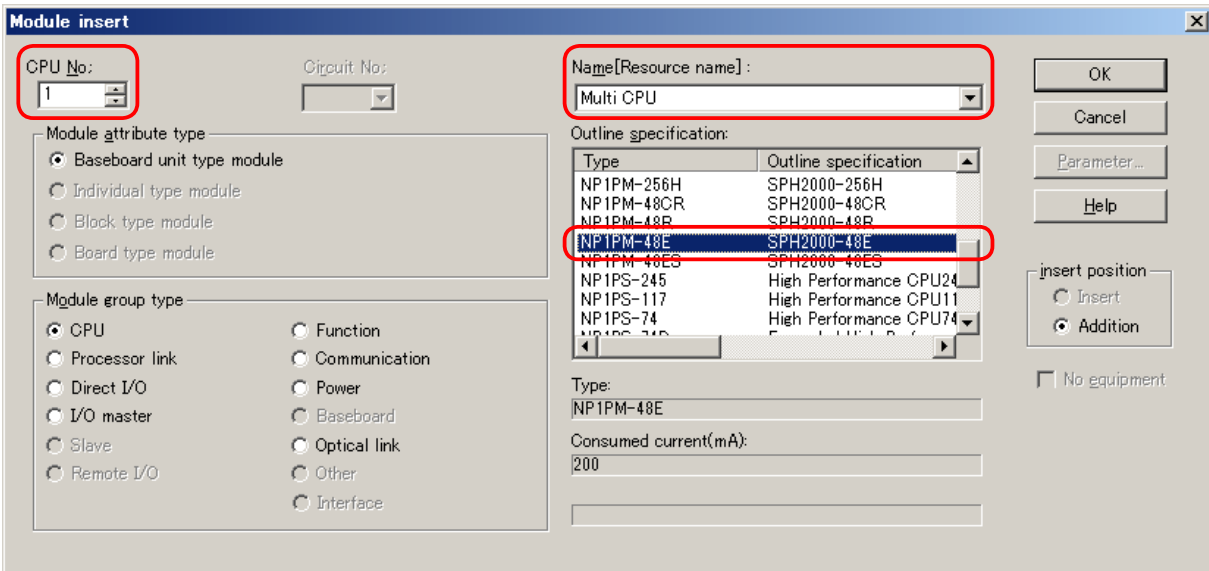
This section describes how to create a new project using a multi-CPU system with CPU 0 (NP1PS-32R) and CPU 1 (NP1PM-48E) as an example.

2.1 Creating a template project for CPU 0

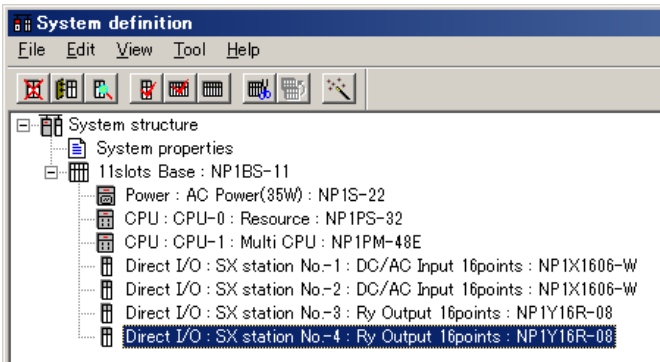
Execute the “New Project” command and select “SPH300-32.” Then, open the system definition screen and add CPU 1.



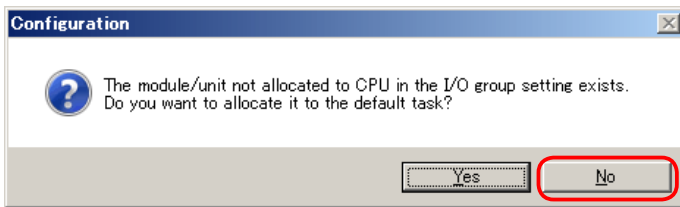
On the “Module insert” dialog, set the CPU No. to “1”, and select “Multi CPU” for the name and “NP1PM-48E” for the outline specification.



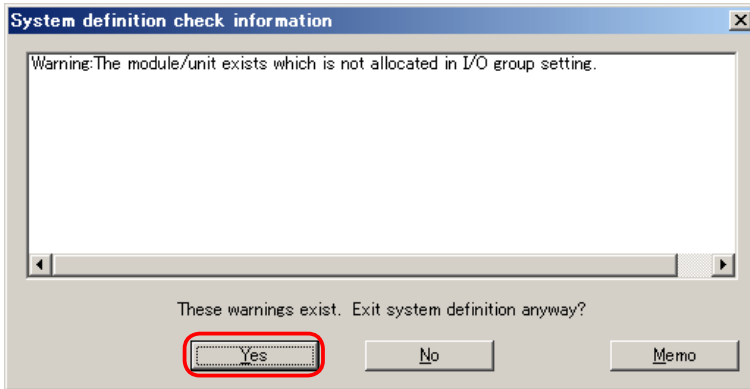
Register the other modules.



Close the system definition screen. The following dialog appears. Click the [No] button.



The following dialog appears. Click the [Yes] button.

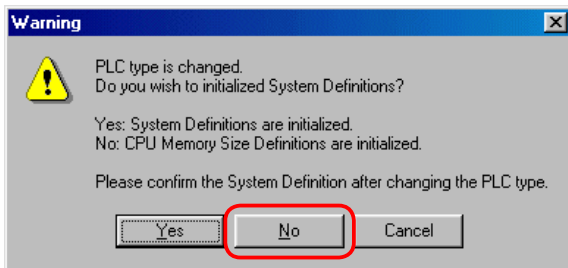
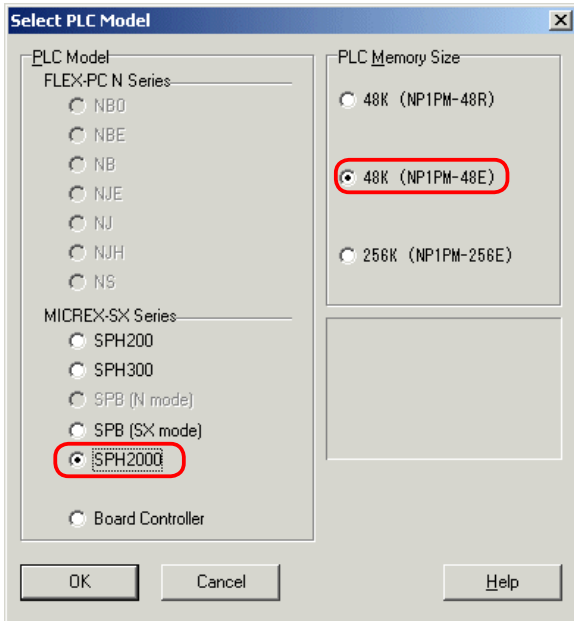


Execute the "Save As" command to save the project with a new name. The template project for CPU 0 has been created.

2.2 Creating a template project for CPU 1

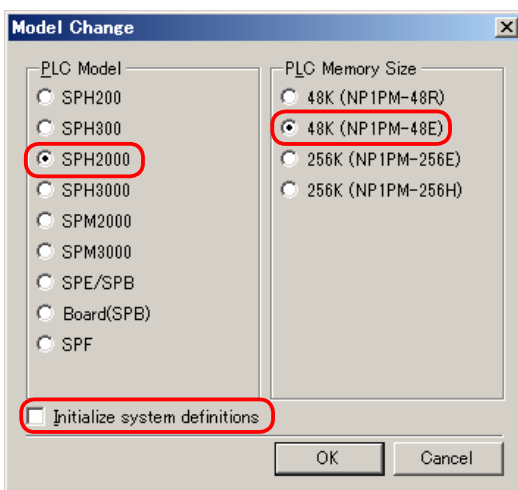
Then, save the same project with a different name. This will become the template project for CPU 1.
Next, change the PLC model to 48E of SPH2000.

<StandardV2>



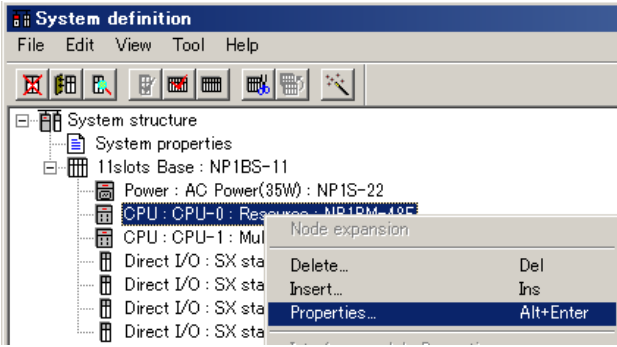
Click the [No] button on this dialog.

<StandardV3>

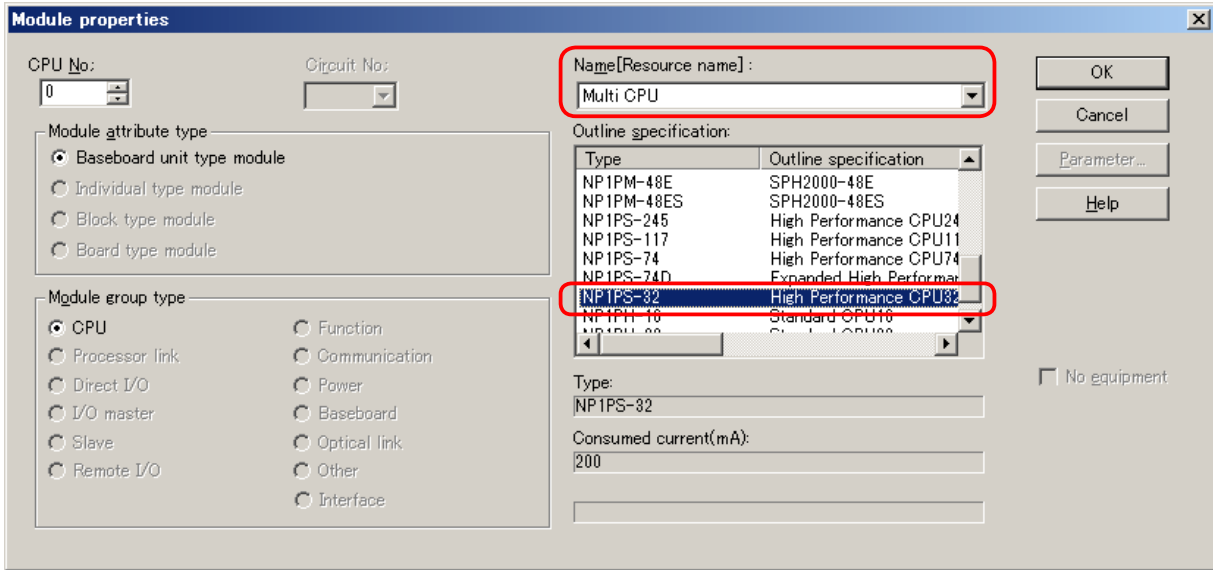


Do not check "Initialize system definitions."

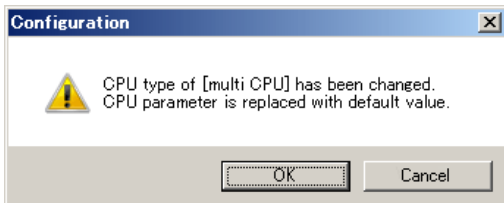
Then, change CPU 0 to "Multi CPU." Right-click CPU 0 and open the "Module properties" dialog.



Change the name to "Multi CPU" and the outline specification to "NP1PS-32."

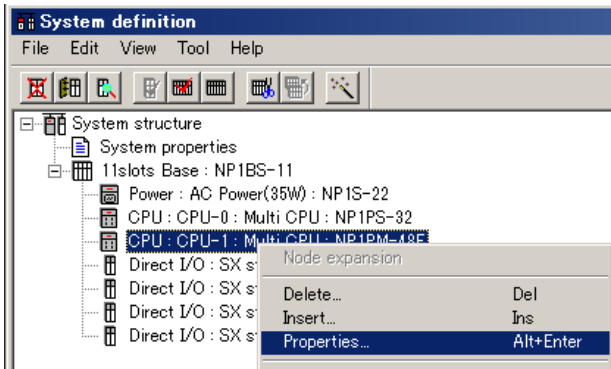


On the following dialog, click the [OK] button.

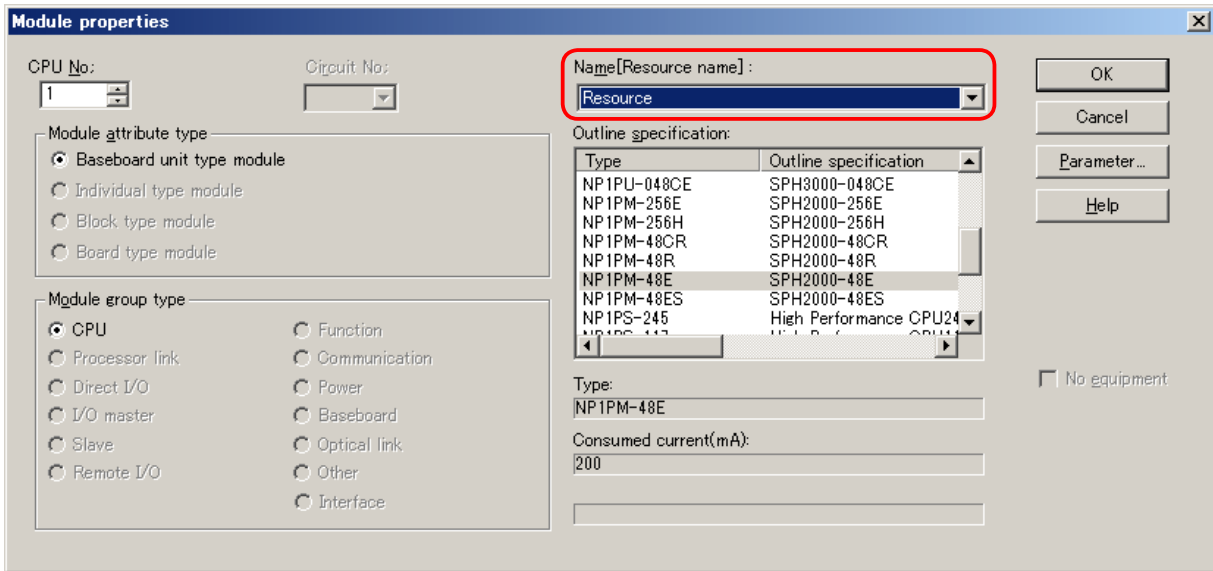


CPU 0 has been changed to "Multi CPU."

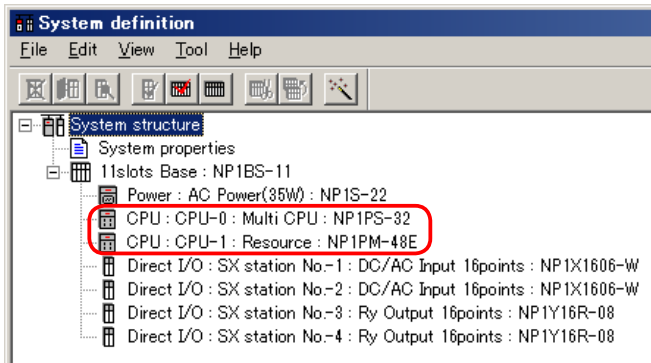
Then, change CPU 1 to "Resource." Right-click CPU 1 and open the "Module properties" dialog.



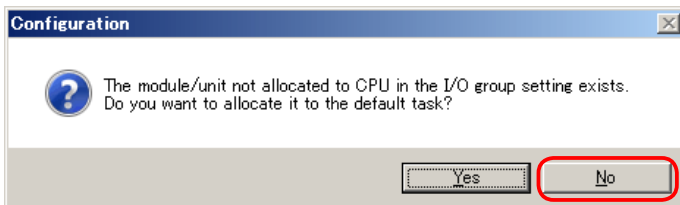
Change the name to "Resource."



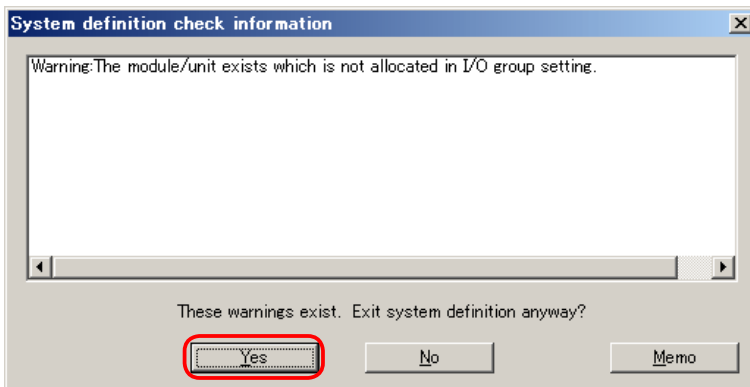
CPU 1 has been changed to "Resource."



Close the system definition screen. The following dialog appears. Click the [No] button.



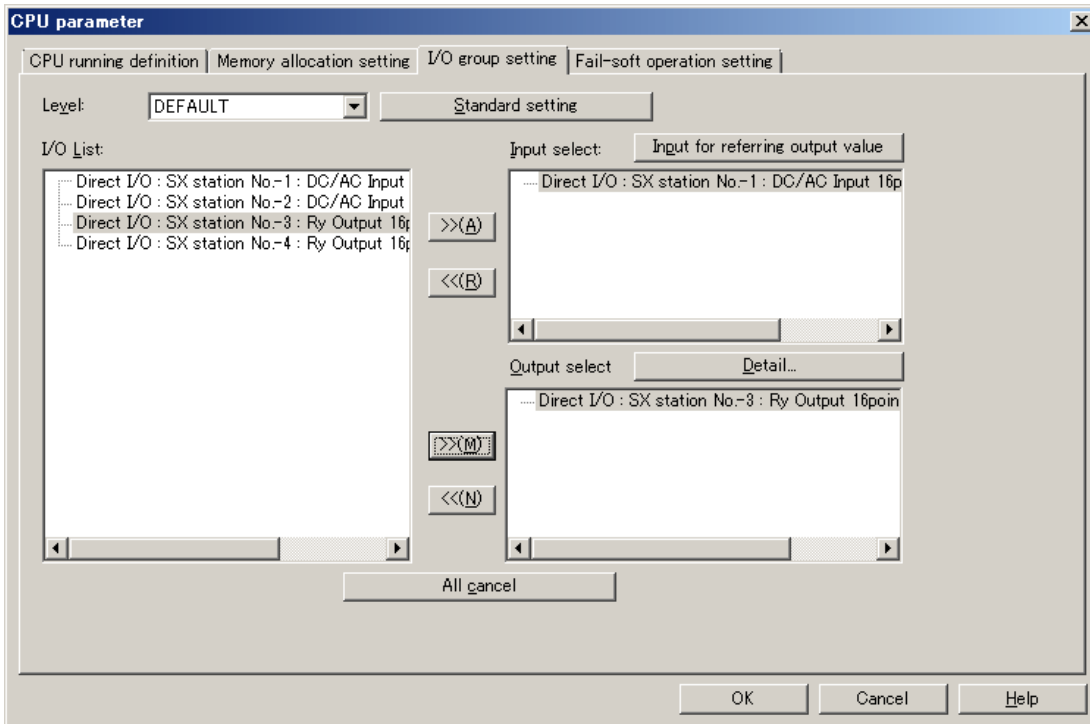
The following dialog appears. Click the [Yes] button.



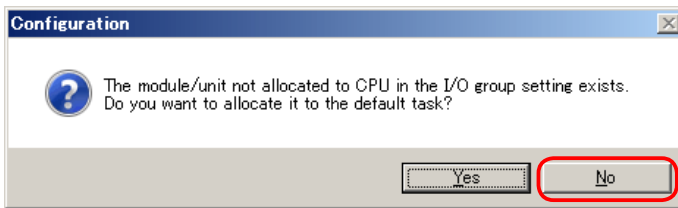
The template project for CPU 1 has been created.

2.3 Setting I/O group of CPU 0

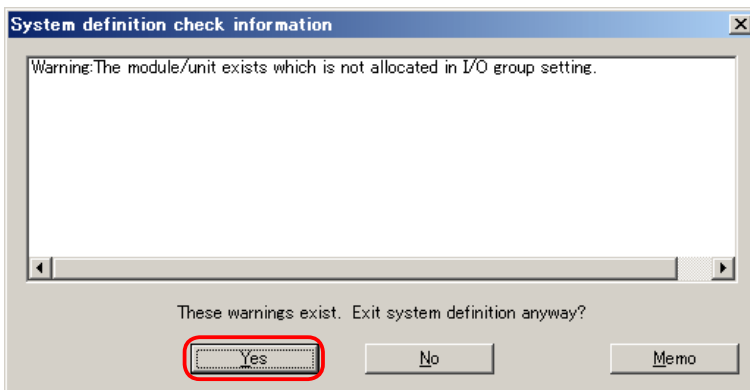
Open the project for CPU 0, and then open the “CPU parameter” dialog in the system definition. Register I/O modules to assign to CPU 0 in Input select / Output select.



Close the system definition screen. The following dialog appears. Click the [No] button.



The following dialog appears. Click the [Yes] button.



2.4 Setting I/O group of CPU 1

In the same manner, open the project for CPU 1, and then open the “CPU parameter” dialog in the system definition. Register I/Os to assign to CPU 1 in Input select / Output select.

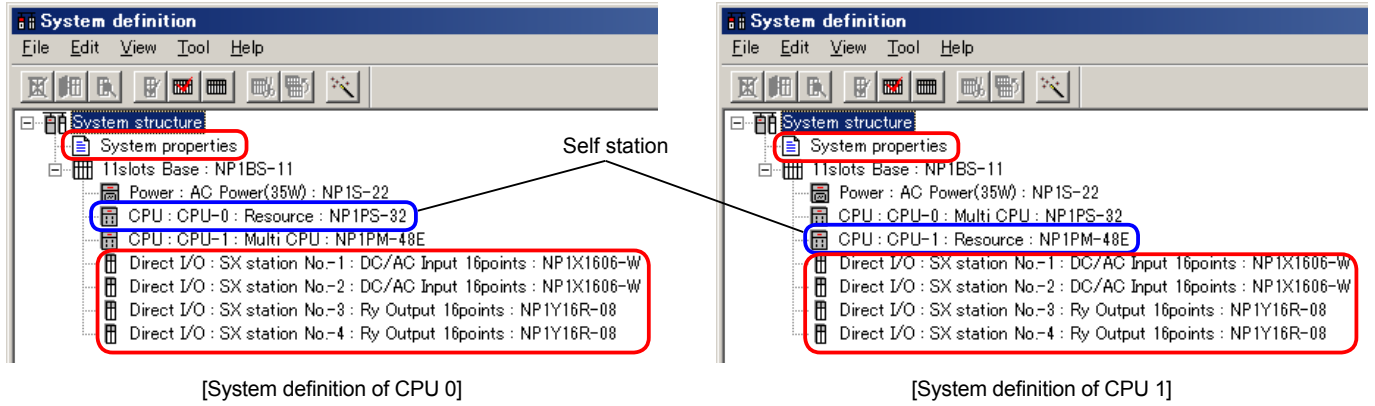
3. Special notes

Basically, the system definition settings such as the system property settings and module registration must be the same in both projects for CPU 0 and CPU 1.

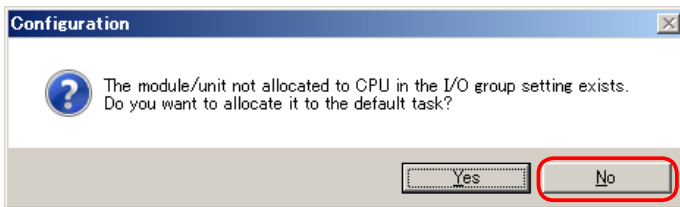
For the name of the CPU, CPU 0 is called "Resource" and CPU 1 is called "Multi CPU" in the project for CPU 0 and CPU 1 is called "Resource" and CPU 0 is called "Multi CPU" in the project for CPU 1.

(Self station = "Resource," The other station = "Multi CPU")

<System definition setting>



The following dialog appears every time you close the system definition screen. Be sure to click **[No]** button. If you click [Yes] button, all the I/O modules including the ones that are to be controlled by another CPU are assigned to the self CPU.



When you change the memory allocation, make the same change to both projects for CPU 0 and CPU 1.

<Memory allocation setting>
 [Project for CPU 0]

CPU 0 Memory allocation setting

CPU parameter

Memory allocation setting

Range of

Non retain memory: 8.0 KW WM0000000

Retain memory: 4.0 KW WL0000000

User FB memory: 4.0 KW

System FB memory: 16.0 KW

Initial data: 3200 Default

Detail of system FB memory

Edge detection	1024	Point x 2W	2048	W
Counter	256	Point x 4W	1024	W
Additional timer	128	Point x 8W	1024	W
Timer	512	Point x 8W	4096	W
Other system FB area			8192	W

CPU 1 Memory allocation setting

CPU parameter

Memory allocation setting

Total Size: 96.0 KW

Memory Size: 64.0 KW

Non retain memory: 64.0 KW

Memory Size: 32.0 KW Auto

Multi CPU non retain memory: 0.0 KW (B)

Retain memory: 8.0 KW (F)

Multi CPU retain memory: 0.0 KW (G)

User FB memory: 8.0 KW (J)

System FB memory: 16.0 KW (K)

Initial data: 12800

Detail of system FB memory

Edge detection	1024	Point x 2W	2048	W
Counter	256	Point x 4W	1024	W
Additional timer	128	Point x 8W	1024	W
Timer	512	Point x 8W	4096	W
Other system FB area			8192	W

Same setting

Same setting

[Project for CPU 1]

CPU 0 Memory allocation setting

CPU parameter

Memory allocation setting

Range of

Non retain memory: 8.0 KW WM0000000

Retain memory: 4.0 KW WL0000000

User FB memory: 4.0 KW

System FB memory: 16.0 KW

Initial data: 3200 Default

Detail of system FB memory

Edge detection	1024	Point x 2W	2048	W
Counter	256	Point x 4W	1024	W
Additional timer	128	Point x 8W	1024	W
Timer	512	Point x 8W	4096	W
Other system FB area			8192	W

OK Cancel

CPU 1 Memory allocation setting

CPU parameter

Memory allocation setting

Total Size: 96.0 KW

Memory Size: 64.0 KW

Non retain memory: 64.0 KW

Memory Size: 32.0 KW Auto

Multi CPU non retain memory: 0.0 KW (B)

Retain memory: 8.0 KW (F)

Multi CPU retain memory: 0.0 KW (G)

User FB memory: 8.0 KW (J)

System FB memory: 16.0 KW (K)

Initial data: 12800

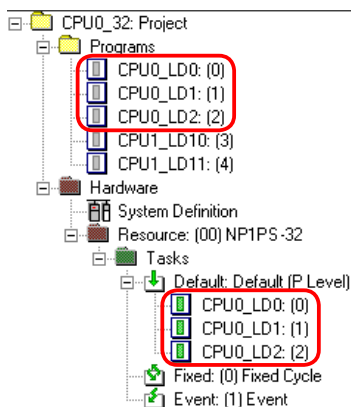
Detail of system FB memory

Edge detection	1024	Point x 2W	2048	W
Counter	256	Point x 4W	1024	W
Additional timer	128	Point x 8W	1024	W
Timer	512	Point x 8W	4096	W
Other system FB area			8192	W

Assign programs to a task in the project for CPU 0 and in the project for CPU 1 respectively.

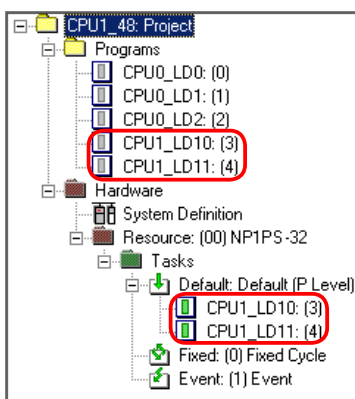
<StandardV2>

[Project for CPU 0]



Assign the programs for CPU 0 to a task.

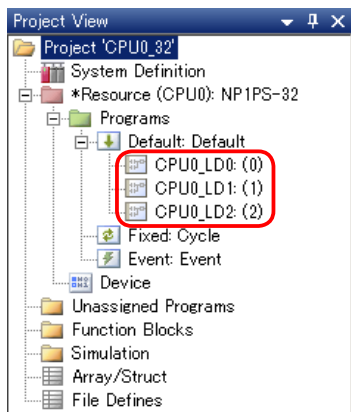
[Project for CPU 1]



Assign the programs for CPU 1 to a task.

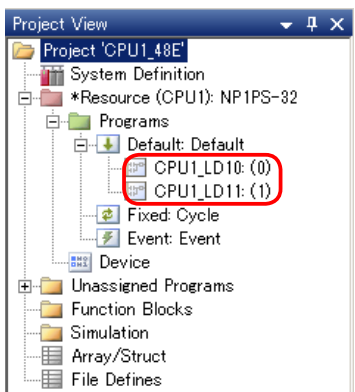
<StandardV3>

[Project for CPU 0]



Assign the programs for CPU 0 to a task.

[Project for CPU 1]



Assign the programs for CPU 1 to a task.

4. When changing the project

When changing system definition or memory allocation, the projects of all the CPUs must be changed in the same way. After changing the projects, download the program and system definition to all the CPUs and reset the entire system.

When changing only a program without changing system definition or memory allocation, you do not need to change the projects of all the CPUs. Change only the program of the target CPU.

When changing memory allocation by uploading the project from a CPU, read the following instructions.

4.1 Correct procedure

- 1) Upload the projects from each CPU and save them individually in a file with a new name for each CPU.
- 2) Close the online screen and open the project file saved in the step 1) in offline mode.
- 3) Change the memory allocation. Make the same change to all the project files.
- 4) Stop all the CPUs and download the changed project files individually to each CPU. After the download is completed, reset the entire system.

4.2 Incorrect procedure

- 1) The project is uploaded from CPU 0. After memory allocation is changed, the project is downloaded to CPU 0.
- 2) The project is uploaded from CPU 1. After memory allocation is changed, the project is downloaded to CPU 1.
- 3) The entire system is reset.

* Do not execute "upload" operation and "project change" operation alternately as follows: "Upload" → "Project change" → "Upload" → "Project change"

Upload the projects from all the CPUs first, change them in offline mode, and then download them individually to each CPU.

If you follow the incorrect procedure, an address that is different from the one specified in the program may be accessed during data communication between the CPUs.