FAQ

YASKAWA

# Title: Why is the CPU load high when there is minimal logic executing?

| Product(s): | Doc. No. |
|---|---|
| MP2300Siec, MP2310iec, MP2600iec, MP3200iec, MP3300iec, Sigma-7Siec, MotionWorks IEC | CNT-DIR2IJ |

One common cause of the CPU load being higher than expected is caused by using a custom Function Block that uses a large structure connected as a VAR_INPUT instead of a VAR_IN_OUT.

When a variable is declared as a VAR_INPUT, the MPiec Controller copies the entire contents of the structure into the Function Block every scan. Other programming languages refer to this technique as “Passing by Value.” In this situation, the Function Block does not have the ability to alter the contents of the original data. When declaring a variable as VAR_IN_OUT, the MPiec Controller passes a pointer to the original location of the variable rather than a copy of the data. Other programming languages refer to this technique as “Passing by Reference.” The pointer is an internal 32 bit value, which is significantly more efficient than copying the entire contents every scan. This method also gives the Function Block the ability to alter the contents of the structure variable.

To show this example in practice, the images on the next page show two scenarios. The first scenario scans a Function Block with a large structure as a VAR_INPUT and results in a high CPU load. The other scenario shows the same structure as a VAR_IN_OUT and has no impact on the CPU load.

The large structure is an array of an array creating 500,000 LREAL’s. There is nothing else in the program, the custom Function Blocks have no code.

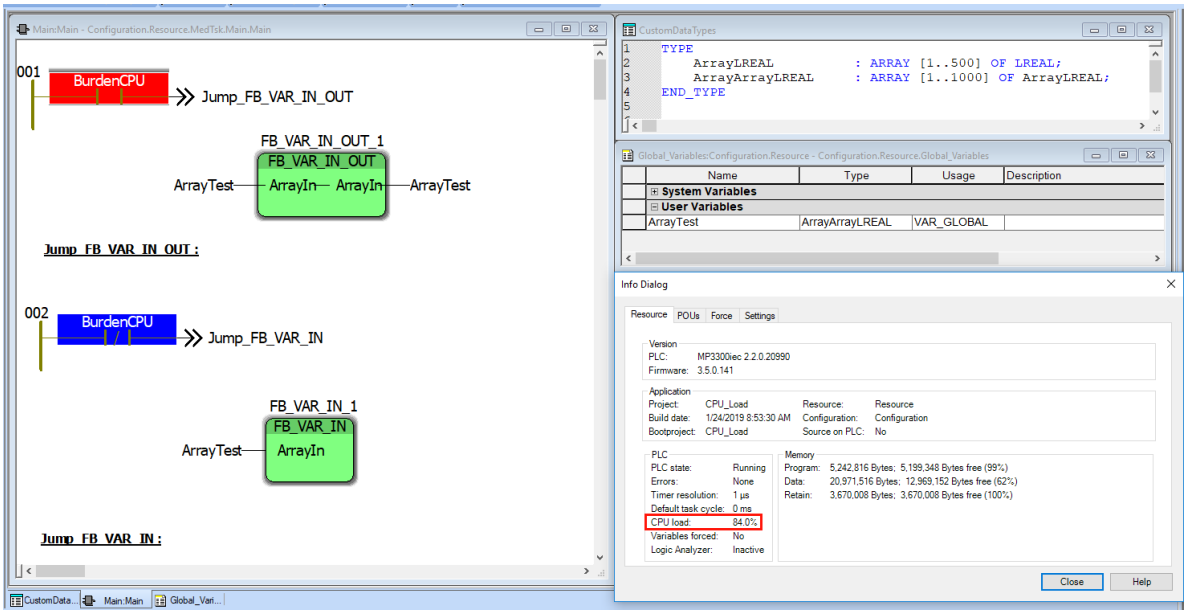
# FAQ

YASKAWA

**Title:** Why is the CPU load high when there is minimal logic executing?

| **Product(s):** MP2300Siec, MP2310iec, MP2600iec, MP3200iec, MP3300iec, Sigma-7Siec, MotionWorks IEC | **Doc. No.** CNT-DIR2IJ |
|---|---|

VAR_IN = 84.0% CPU Load




VAR_IN_OUT = 0.1% CPU Load