Prevention of overload using "CPU_RT"

SIMATIC PCS 7 V7.0 and higher

Application Example • November 2011
Industry Automation and Drive Technologies Service & Support Portal

This article is taken from the Service Portal of Siemens AG, Industry Automation and Drive Technologies. The following link takes you directly to the download page of this document:


If you have any questions about this document, please contact us under the following e-mail address:

online-support.industry@siemens.com

For further information on this topic, you may also actively use our Technical Forum in the Service & Support Portal. Add your questions, suggestions and problems and discuss them in our large forum community:

http://www.siemens.com/forum-applications
SIEMENS

SIMATIC PCS 7

CPU_RT

Application Example

Task Description and Solution
Configuration and Settings
Summary
Links & Literature
History
Warranty and Liability

Note

The application examples are not binding and do not claim to be complete with regard to configuration, equipment or any contingencies. The application examples do not represent customer-specific solutions; they are only intended to provide support for typical applications. You are solely responsible for the correct operation of the described products. These application examples do not relieve you of your responsibility to use sound practices in application, installation, operation and maintenance. When using these application examples, you recognize that we will not be liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these application examples at any time without prior notice. If there are any deviations between the recommendations provided in this application example and other Siemens publications (e.g. catalogs), the contents of the other documents shall have priority.

We do not accept any liability for the information contained in this document.

Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act (“Produkthaftungsgesetz”), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or violation of fundamental contractual obligations. Damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change in the burden of proof to your detriment.

It is not permissible to transfer or copy these Application Examples or excerpts thereof without express authorization from Siemens Industry Sector.
Table of Contents

Warranty and Liability ............................................................................................................. 4

1 Task Description and Solution .............................................................................................. 6
  1.1 Function .......................................................................................................................... 6
  1.2 Solution .......................................................................................................................... 6
  1.2.1 Prerequisites and validity ......................................................................................... 6
  1.2.2 Functional scope ....................................................................................................... 8
  1.2.3 Applications .............................................................................................................. 9

2 Configuration and Settings ................................................................................................... 10
  2.1 New controller overload behavior .................................................................................. 10
  2.1.1 Configuration of cycle overload behavior ................................................................. 11
  2.1.2 Signaling behavior with overload .............................................................................. 14
  2.1.3 Cancelation of emergency mode in case of controller overload .............................. 14
  2.2 Cycle Performance Analysis ......................................................................................... 15
  2.2.1 Display of cycle performance data on OS/MS .......................................................... 15
  2.2.2 Scenario 1: High controller cycle load (maintenance demand) .............................. 18
  2.2.3 Scenario 2: OB request error (maintenance request) ............................................... 19
  2.2.4 Scenario 3: Exceedance of the maximum cycle time (maintenance alarm) ............ 20

3 Summary ................................................................................................................................ 21

4 Links & Literature ................................................................................................................. 22

5 History ................................................................................................................................... 22
1 Task Description and Solution

1.1 Function

The controller of an automation system has two primary tasks. On the one hand the controller automates the process and on the other hand the controller has to evaluate and forward any operations and displays.

All tasks are processed sequentially in the controller (one after the other). In order to ensure an appropriate processing time, the controllers of the SIMATIC product line have a basic cycle monitoring (OB1) of 6 seconds maximum. This leads to a time limitation of applications, in addition to the memory-based limitation. If the cycle checkpoint is not reached within 6 seconds, OB80 (time error) will be called and a system message will be issued.

For process automation, SIMATIC PCS 7 provides the option of reacting to temporary and permanent cycle overload through controlled, automatic load shedding. This functionality is realized by the "CPU_RT" function block and will be available as of PCS 7 version 7.0.

1.2 Solution

The required load relieving measures in case of a PCS 7 controller overload are automatically performed by the "CPU_RT" function block. The "CPU_RT" function block combines the following functions:

- Maintaining operability through a new overload behavior by means of multi-stage reduction of the cyclic load.
- Prevention of overload through a cycle performance analysis and visualization of performance data in the Asset Management

1.2.1 Prerequisites and validity

SIMATIC PCS 7 version V7.0 or higher is required for the automation solution.

The function for determining the cycle performance data of the "CPU_RT" function block requires a controller firmware revision level that supports the "SFC 78" system function.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Firmware versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>for single controller</td>
<td>5.0 or higher</td>
</tr>
<tr>
<td>for H controller</td>
<td>4.5 or higher</td>
</tr>
</tbody>
</table>
If you compile CFCs with the option "create module driver", a plan named 
"@CPU_RT" will be created as of PCS 7 V7.0. It will include the "CPU_RT" 
function block, which will be integrated in all relevant execution levels (OB100, 
OB1, all OB 3x, and OB 8x) by the module driver wizard.

Figure 1-1: "CPU_RT" block

Note

The "CPU_RT" block is placed into the system plans by the PCS 7 module driver 
wizard. All controllers will feature the new overload behavior as of PCS 7 V7.0 in 
combination with the V7 Library.

When an older project is migrated without utilization of new functions, the driver 
wizard will not integrate the "CPU_RT" in PCS 7 V7.0, either.

Apart from the "SFC 78" system function the "Asset Management" option package
is required for full utilization of the new controller overload behavior and the cycle 
performance analysis.
1.2.2 Functional scope

Depending on the CPU/FW used and whether or not Asset Management is implemented, the functions of CPU_RT will be displayed and parameterized as follows:

<table>
<thead>
<tr>
<th>Function</th>
<th>Constellation</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycle performance analysis</td>
<td>Controller FW with SFC 78 and option package Asset Management available</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Display of cycle performance data on OS</td>
<td>Controller FW without SFC 78 and option package Asset Management available</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Display of cycle performance data in the CFC</td>
<td></td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Messages</td>
<td></td>
<td>YES</td>
<td>YES*</td>
<td>YES*</td>
</tr>
<tr>
<td>Parameterization via faceplate</td>
<td></td>
<td>YES</td>
<td>YES (via CFC)</td>
<td></td>
</tr>
<tr>
<td>New controller overload behavior</td>
<td></td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>

* No operator message "Net time requirement of all OBs exceeds Max Limit", since performance data cannot be calculated.

Note: The function of the "CPU_RT" block is described in detail in the manual "SIMATIC Process Control System 7 PCS 7 Basis Library V7.1".
1.2.3 Applications

The following applications were taken into account in the development of the "CPU_RT" functional block:

- Applications with little time reserve have the highest risk of overload. Systems with high time load can go into overload quickly through a defective component, typically a defective measuring line that produces an alarm flood (chattering alarm). The "CPU_RT" block enables you to determine the overall runtime (with interruptions by higher-priority process levels) through OBs during commissioning and in normal operation, apart from the net runtime of the organization block (OB). Early load relief of time-critical OBs represents a good protection against controller overload.

- Changes and extensions of the user program with an already high cycle load is another application. Changing or extending the user program causes an overload of the controller after loading into the automation system (AS). The cycle overload of the controller is selectively reduced through the "CPU_RT" block. Process levels or OBs representing a very high load are temporarily reduced (load shedding - emergency operation) until the temporary load has diminished. This ensures permanent operability of the plant even in case of overload. It is possible to reverse changes during operation or to restructure the user program accordingly, in order to return to normal operation. One example for the case described is load reduction by transferring program parts to slower OBs.

- Adding communication lines may lead to a controller overload under certain circumstances. In this case, too, the block allows for cancelation of the change in "emergency" operation.

If an overload occurs during operation for no obvious reason, you can localize the cause by a detailed performance analysis of the "CPU_RT" block. In particular the net times (net OB runtime without interruptions) must include an indication as to where additional time losses have occurred (see chapter 2.2) This can even be done retroactively in normal operation, due to the new overload behavior.
2 Configuration and Settings

2.1 New controller overload behavior

In case of a controller cycle overload you can prevent the controller from being unoperable through controlled load shedding. Load shedding is effected by the "CPU_RT" block and is achieved by reducing cyclical process levels. Load shedding represents an emergency mode, the cause of which needs to be eliminated as soon as possible.

Emergency mode is initiated when the cycle checkpoint monitoring time has been exceeded twice. In this case the cycle checkpoint has not been reached for at least 12 seconds due to a temporary or permanent overload. The process control message "emergency mode, cyclic OBs are reduced" is issued with standard priority "16". Therefore this message is displayed in the list of highest-priority messages.

Note

With the "Asset Management" option package available an additional maintenance alarm will be generated at the Asset faceplate of the CPU concerned.

Figure 2-1: Emergency mode of the controller
2.1 New controller overload behavior

In emergency mode the blocks are skipped x times, independent of the reduction factor, so that you can continue processing all blocks.

The following two escalation levels are available:

- **Level 1:**
  The cycle checkpoint monitoring time has elapsed twice (approx. 12s). The OBs 3x released for reduction are reduced once.

- **Level 2:**
  If time errors continue to occur, all OB 3x are incrementally reduced.

**Note**

The emergency mode is not meant to be a permanent bridging of the "cycle overload" state of the controller, but is rather used for temporarily handling overload situations. The cause for the emergency mode must be eliminated without delay. This is also the reason why the emergency mode will be reported as a maintenance alarm in the Asset Management. In order to change back to normal operation you can, for example, restructure the program or reset / transfer the changes made.

### 2.1.1 Configuration of cycle overload behavior

If the Asset Management option package is available, the required overload behavior is parameterized in the Asset Faceplate of the CPU, by the following procedure:

1. Start the OS runtime.
2. Open the Asset Faceplate of the CPU.
3. Parameterize the desired overload behavior.

In Figure 2-2 below, OB 33 and OB 35 are excluded from the first escalation level and are not reduced there.
2 Configuration and Settings

2.1 New controller overload behavior

Figure 2-2: Parameterizing the block "CPU_RT"

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>OB30 reduction with overload</td>
<td>✔</td>
</tr>
<tr>
<td>OB31 reduction with overload</td>
<td>✔</td>
</tr>
<tr>
<td>OB32 reduction with overload</td>
<td>✔</td>
</tr>
<tr>
<td>OB33 reduction with overload</td>
<td>✔</td>
</tr>
<tr>
<td>OB34 reduction with overload</td>
<td>✔</td>
</tr>
<tr>
<td>OB35 reduction with overload</td>
<td>✔</td>
</tr>
<tr>
<td>OB36 reduction with overload</td>
<td>✔</td>
</tr>
<tr>
<td>OB37 reduction with overload</td>
<td>✔</td>
</tr>
<tr>
<td>OB38 reduction with overload</td>
<td>✔</td>
</tr>
<tr>
<td>Max. number support STOP demanded</td>
<td>10</td>
</tr>
<tr>
<td>Alarm limit capacity</td>
<td>70 %</td>
</tr>
<tr>
<td>Critical reduction ratio</td>
<td>95 %</td>
</tr>
<tr>
<td>Hysteresis alarm limit</td>
<td>5,0 %</td>
</tr>
<tr>
<td>Calculating the CPU load (Display)</td>
<td>25 Cycles</td>
</tr>
<tr>
<td>Calculating the CPU load (Internal)</td>
<td>5 Cycles</td>
</tr>
<tr>
<td>Message as of request error</td>
<td>4</td>
</tr>
<tr>
<td>Increase CPU load to</td>
<td>5 Cycles</td>
</tr>
</tbody>
</table>
Note
In case the Asset Management option package is not available to you, you will have to parameterize the overload behavior directly at the "CPU_RT" block in the CFC plan "@CPU_RT". This procedure is described in following.

The CFC plan "@CPU_RT" is a system plan and must not be deleted.

Reduction factor (max. number of retrigger attempts)

The overload behavior is parameterized at the "MAX_RTG" input. Here you can set the maximum number of retrigger attempts.

Note
If you assign "0" to the "CPU_RT" input, the overload behavior will be as in older SIMATIC PCS 7 versions (< V7.0).

- **MAX_RTRG=1**
  Only the first level of the new overload behavior is initiated.
  If the load shedding through level 1 is not sufficient, the controller will go to STOP at "MAX_RTRG" = 1 after the cycle checkpoint monitoring time (>18s) has been exceeded three times, and level 2 will not be initiated.

- **MAX_RTRG = 2**
  After the cycle checkpoint monitoring time (18s) has been exceeded three times, level 2 will be initiated. All OBs will be reduced once. If this load shedding is not sufficient, either, the controller will go to STOP state after the cycle checkpoint monitoring time (>24) has been exceeded four times.

- **MAX_RTRG = x**
  If you assign a value > "2" to the input, the cycle checkpoint monitoring time can be exceeded x times. The reduction factor will be increased with each exceedance.
2.1 New controller overload behavior

2.1.2 Signaling behavior with overload

Overall load alarm limit (only for controller with SFC 78):

If the overall load (sum of mean values of all OB net runtimes) exceeds the upper limit set at the parameter overall load alarm limit or MAX_VAL (CPU_RT), the message "net time requirement of all OBs exceeds Max Limit" is output.

Note: See chapter 2.2.2 "Scenario 1: High controller cycle load (maintenance demand)"

Max. number of OB request errors

If the number of OB request errors parameterizable at parameter "signaling from request error" or N_REQ_ERR (CPU_RT) is exceeded, the message "OB 3x request error" will be output.

Note: See chapter 2.2.3 "Scenario 2: OB request error (maintenance request)"

2.1.3 Cancelation of emergency mode in case of controller overload

You can automatically end the emergency mode if you reduce the maximum overall load of the controller configured at the parameter "cancel reduction at" or MAX_VAL (CPU_RT) correspondingly and no cycle time exceedance occurs on a permanent basis.

The reductions are also canceled if the slowest OB processes a number of cycles (Parameter: UNDO_CYC) again.

In this case the value of UNDO_CYC must not be too small in order to avoid frequent switching between stop prevention measures and normal operation.
2.2 Cycle Performance Analysis

The "CPU_RT" block determines the runtime of the individual OBs and their participation in the cycle time (cycle performance analysis). This performance data and load information can help you even in the project engineering phase for structuring the user program, allowing you to selectively plan reserves for extensions, utilize the capacity of the individual process level homogeneously, and prevent later controller overload.

Note
The cycle performance analysis is available for controllers with SFC 78 only.

2.2.1 Display of cycle performance data on OS/MS

Preconditions
The cycle performance data is visualized on the OS/MS if the PCS 7 option package "Asset Management" is installed.

Note
The cycle performance data can only be displayed on OS/MS for controllers with SFC 78 and Asset Management.

Configuration
Via the Asset faceplate of the controller you can select the faceplate representing the overall load of the controller (Figure 2-3).

1. Start the OS runtime.
2. Open the faceplate of the corresponding CPU.
3. Select "performance" in the drop-down list.
2 Configuration and Settings
2.2 Cycle Performance Analysis

Figure 2-3: Overall load of the controller

Note
The net runtime is the runtime that was needed for processing the code placed in OB3x, if it is not interrupted by higher-priority OBs.

The overall runtime is the runtime that was actually needed for processing the code placed in OB3x, including the runtimes of higher-priority OBs.

The vertical bar represents the overall load of the controller in % and the sum of all net runtimes (cycle times) of all OBs.

For detail analysis, a further Asset faceplate represents the determined cycle times in the individual process levels (OBs) (Figure 2-4).
2 Configuration and Settings

2.2 Cycle Performance Analysis

Figure 2-4: OB cycle times

![CPU Function Block]

Note

The overall load of a single controller after commissioning should not be higher than 75% (net).

If 20 percent of the cycle time are used for communication and acyclic events, a reserve of 5 to 10% will remain for future program extensions or additional communication orders.

Example

The net runtime of OB38 is 5ms. With a sampling time of 10ms the overall load rate is 50 percent. In this case the overall and net runtime are almost identical for OB38, if it is not interrupted by higher-priority OBs.

The net runtime of OB35 is 45ms. With a sampling time of 100ms the overall load rate is 45 percent. The overall runtime is clearly higher than the net runtime, the reason being that OB35 is interrupted by OB38.

There is a risk of overload with an overall load of 95 percent. The overall runtime of OB35 indicates that this OB is close to an OB request error. In case of acyclic events or high communication load a controller overload might occur in this described example. Due to the overall load and the overall runtime of OB35 the program must be restructured. Program extensions in OB35 or in faster OBs may lead to overload. Here, as a consequence, you should transfer existing or newly added program parts to slow OBs.
### Conclusion

The program logic and processing for slower process signals (e.g., temperatures) are typically placed in OB3x with a call interval of 1 to 2 seconds, and dynamic process values (e.g., print) are processed in faster OBs. This preventive measure is the decisive measure for preventing a cycle overload of the controller.

#### 2.2.2 Scenario 1: High controller cycle load (maintenance demand)

If the overall load (sum of mean values of all net runtimes in % of OB3x, OB8x + OB1) exceeds the overall load alarm limit (Figure 2-5), the message "net time requirement of all OBs exceeds Max Limit" is output and the maintenance status is set to "maintenance demand".

Due to this preventive maintenance message you must take action for cycle relief in the user program in order to avoid a cycle overload of the controller, such as adapting, dividing, or transferring the user program.

![Figure 2-5: Net time requirement of all OBs exceeds Max Limit](image)

**Note**

If "SFC 78" is not supported, the controllers will not signal the exceedance of the net time requirement.
2.2.3 Scenario 2: OB request error (maintenance request)

If the overall load of the controller reaches the limit of 100%, OB request errors will be triggered. In this case cyclic OBs are no longer processed appropriately and called within the specified time frame, since they are, for example, interrupted by acyclic OBs (diagnosis alarms). This lets the program processing time (overall time) be longer than the call interval of the OB (e.g. OB35: 100ms).

If the parameterizable number of request errors is exceeded (Figure 2-6), the message "OB 3x request error" will be issued and the maintenance status is set to "uncertain maintenance request".

When there is no longer an overload situation, the maintenance status will be reset and the process control message will be marked as "going out".

Figure 2-6: OB request error
2.2.4 Scenario 3: Exceedance of the maximum cycle time (maintenance alarm)

If the cycle checkpoint monitoring time set (HW Config > CPU properties) is exceeded (PCS 7 default: 6000ms), the message "cycle time exceeded: 6001ms, OB1" will be output and the maintenance status of the CPU will be set to "bad or maintenance alarm" (Figure 2-7).

The following events can lead to a cycle overload:

- high communication load
- Alarm flooding
- Frequent acyclic fault events (chattering alarms)
- Misdimensioning of cyclic interrupts when adding new program parts

Figure 2-7: Exceedance of cycle time
3 Summary

If you use the new PCS 7 V7.0 Library, the "CPU_RT" block will always be integrated by the driver wizard. This is done individually of the use of Asset Management and the controller firmware.

After an upgrade of the driver components from legacy projects to PCS 7 V7.0 the controller will show a different cycle overload behavior.

In emergency mode of the controller the cyclic OBs are reduced. This behavior affects integral computations, fast switch-off functions, or fast control procedures. In case of overload the STOP controller or the emergency mode controller (permanent operability) can be used.

Therefore you can set the desired behavior for cycle overload either at the Asset faceplate of the CPU or, if the "Asset Management" option package is not available, directly at the "CPU_RT" block.

Note

You can disable the "CPU_RT" block by setting the "MAX_RTRG" parameter to "0".

Since controller overload mostly occurs due to misconfigurations or the cumulation of acyclic events (nuisance/diagnosis alarms OB8x), the permanent operability of the system and the option of searching and eliminating the error with the system running is preferred. For this reason the new overload behavior of the controller is preset as of version PCS 7 V7.0.
4 Links & Literature

The following table gives references to further information.

Table 4-1  Internet links

<table>
<thead>
<tr>
<th>Topic</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>\1\ Reference to this entry</td>
<td><a href="http://support.automation.siemens.com/WW/view/en/55692652">http://support.automation.siemens.com/WW/view/en/55692652</a></td>
</tr>
<tr>
<td>\2\ Siemens Online Support</td>
<td><a href="http://support.automation.siemens.com">http://support.automation.siemens.com</a></td>
</tr>
<tr>
<td>\3\</td>
<td></td>
</tr>
</tbody>
</table>

5 History

Table 5-1  History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Revisions</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.0</td>
<td>12.01.2007</td>
<td>First edition on the DVD PCS 7 V7.0</td>
</tr>
<tr>
<td>V2.0</td>
<td>05.11.2009</td>
<td>Layout adaptation for publishing in the Service &amp; Support Portal under &quot;Applications &amp; Tools&quot; (see link in Table 4-1)</td>
</tr>
<tr>
<td>V3.0</td>
<td>14.11.2011</td>
<td>Überarbeitet; neue Screenshots; Layout angepasst</td>
</tr>
</tbody>
</table>