Application on Control Technology

applications & TOOLS

Programming Example in Ladder Logic

SIEMENS

Getting the Status for the CPU LED Indicators

Note

The application examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The application examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible in ensuring that the described products are correctly used. These application examples do not relieve you of the responsibility in safely and professionally using, installing, operating and servicing equipment. When using these application examples, you recognize that Siemens cannot be made 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 these application examples and other Siemens publications - e.g. Catalogs - then the contents of the other documents have priority.

Warranty, liability and support

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 breach of a condition which goes to the root of the contract ("wesentliche Vertragspflichten"). However, claims arising from a breach of a condition which goes to the root of the contract shall be limited to the foreseeable damage which is intrinsic to the contract, unless caused by intent or gross negligence or based on mandatory liability for injury of life, body or health The above provisions does not imply a change in the burden of proof to your detriment.

For questions about this document please use the following e-mail address:

mailto:csweb@ad.siemens.de

Table of Contents

Reference to Automation and Drives Service & Support

This entry is from the internet application portal of Automation and Drives Service & Support. The documentation has the entry ID **23330722**. Click the link below to directly display the download page of this document.

http://support.automation.siemens.com/WW/view/en/23330722

All entries referenced in this document are designated by their entry ID and addressed via the above path.

Getting the Status of the LED Indicators

Overview

The LEDs on the front on an S7 CPU display the overall state using a series of LEDs on their front panel. It is often useful to duplicate this display on an HMI system. This applications note shows how you can do that in an S7 program.

There are two system status lists that contain this information: w#16#xx19 and w#16#xx74. System Status List (SSL) w#16#xx19 gets LED status from non-redundant CPU's, while SSL w#16#xx74 gets LED status from redundant CPUs. In order to support the widest range of CPU's, GetLEDState tries SSL w#16#0119 first. If this fails, GetLEDState tries SSL w#16#0174.

Objectives

This example provides sample LAD instructions in an FC that calls SFC51 (RDSYSST) and returns the status for a specific LED.

The GetLEDStatus function was written in STEP 7 5.3.0.1, and tested on WinAC RTX 4.0.1. It should work on any version of STEP 7. It will run on any CPU that supports either SSL_ID W#16#0119 or W#16#0174 and that has at least 18 bytes of temp (L) memory available where GetLEDStatus is called.

Operation of the sample program

The sample program performs the following tasks:

- OB1 calls FC112 ("GetLEDState") with the Index for the LED.
- FC112 calls SFC51 ("RDSYSST") for either SSL_ID W#16#0119 or W#16#0174 (whichever the CPU supports) and returns the following data for the LED:
  - On/Off status: LED_On is true when the led is on, whether solid or flashing.
  - Flashing at the "normal" (fast) rate: Both LED_On and Flash_Fast are true.
  - Flashing at a slow rate: Both LED_On and Flash_Slow are true.

The sample program uses Index W#16#3 to obtain the status of the EXTF (External Fault) LED.

Note FC112 ("GetLEDState") returns the state of a single LED (specified as an input parameter). To read the state of multiple LEDs, use multiple calls to FC112 and vary the input parameter.

1 Using SFC51 ("RDSYSST") to read the state of an LED

OB1 calls FC112 ("GetLEDState") and provides the SSL_Index for a specific LED. SFC112 then calls SFC51 ("RDSYSST") and returns the state of the LED.

There are two system status lists (SSL) that contain this information:

- W#16#xx19 gets LED status from non-redundant CPUs
- W#16#xx74 gets LED status from redundant CPUs

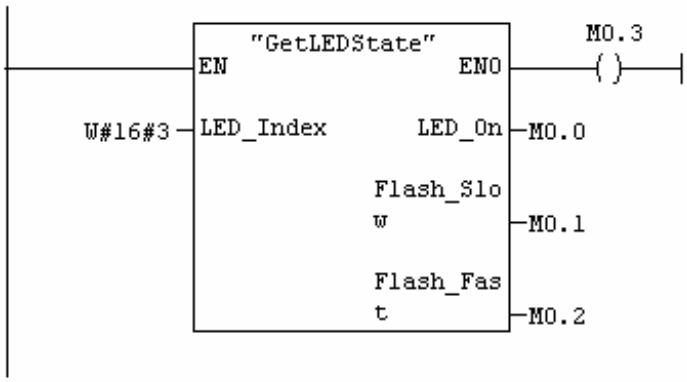
In order to support the widest range of CPUs, GetLEDState tries SSL W#16#0119 first. If this fails, GetLEDState tries SSL w#16#0174.

1.1 OB1

In the sample program, OB1 calls FC112 ("GetLEDState") and provides the value of the EXTF (External Fault) LED. SFC112 returns the state of the LED in three Boolean outputs:

- LED_On: When LED_On = 1 (ON or true), the LED is on (either solid or flashing).
- Flash_Fast: When both LED_On and Flash_Fast = 1 (ON or true), the LED is flashing at the “normal” (fast) rate.
- Flash_Slow: When both LED_On and Flash_Slow = 1 (ON or true), the LED is flashing at a slow rate.


```
OB1 :  "Main Program Sweep (Cycle)"
Network 1: Get state of EXTF LED
```



1.2 FC112 ("GetLEDState")

FC112 ("GetLEDState") calls SFC51 ("RDSYSST") to query the state of an LED for the CPU. For example, the S7 CPU modules CPU416-2F and CPU 416-3 support the following LED_Index values:

- INTF W#16#0002
- EXTF W#16#0003
- RUN W#16#0004
- STOP W#16#0005
- FRCE W#16#0006
- BUS1F W#16#000B
- BUS2F W#16#000C
- IFM1F W#16#0012 (if the CPU supports IF Modules)
- IFM2F W#16#0013 (if the CPU supports IF Modules)

Parameters of FC112

As shown in Table 1-1, FC112 uses one input parameter and three output parameters.

After successfully reading the LED state, FC112 returns ENO = True. If FC112 encounters a problem with either RDSYSST call, then it returns ENO = False.

Table 1-1 Parameters for FC112 ("GetLEDState")

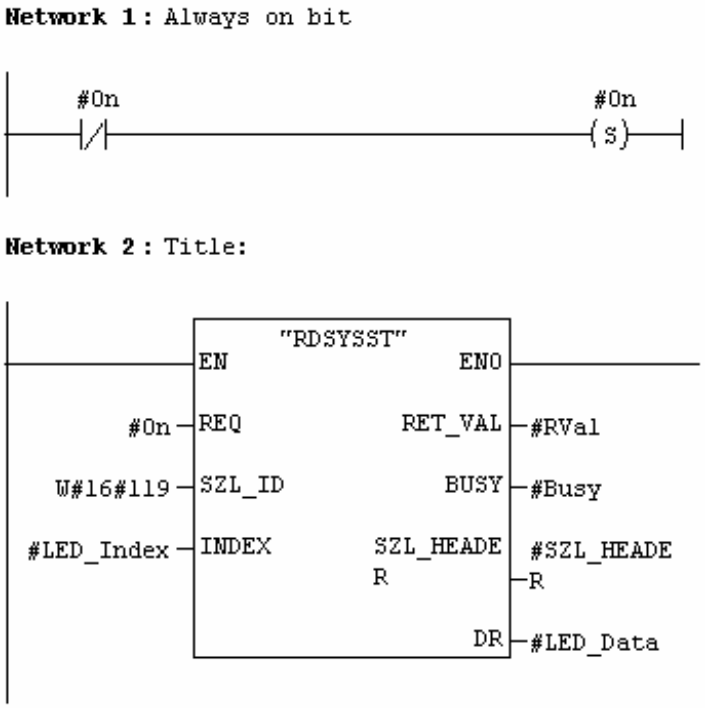
Type	Size	Parameter
Input	WORD	LED_Index: SSL_ID for a specific LED Refer to the documentation for SSL W#16#xx19 (Table 2-3) or W#16#xx74 (Table 2-4).
Output	BOOL	LED_On: On/Off state of the LED (either On or flashing(
	BOOL	Flash_Fast:
	BOOL	Flash_Slow:

Note There is a form of the SSL that returns the state of all LEDs; however, decoding this data to known locations for LEDs is cumbersome. In addition, many of the LEDs do not apply to most installations.

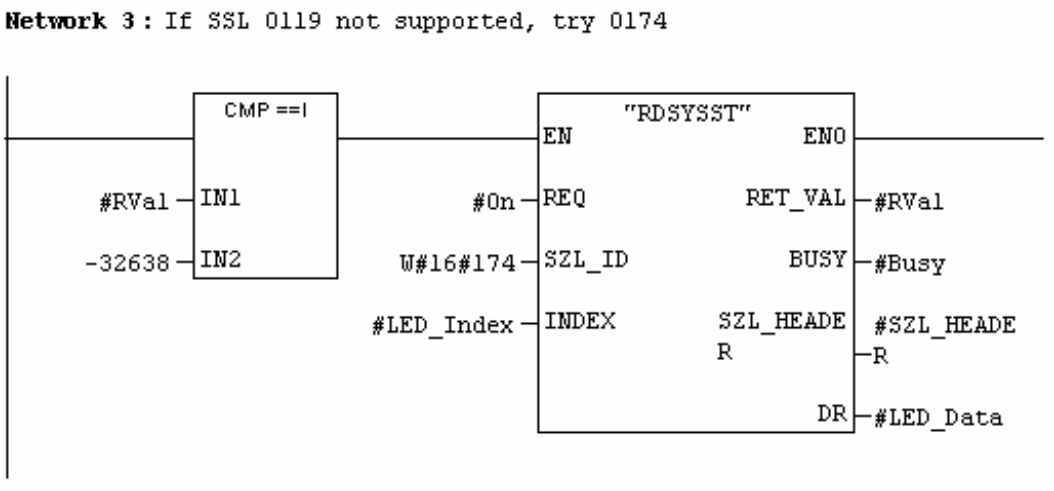
Calling FC112 ("GetLEDState") for each individual LED allows you to easily get the specific information required.

LAD networks for FC112 ("GetLEDState")

SFC112 first calls SFC51 ("RDSYSST") with SSL W#16#0119.



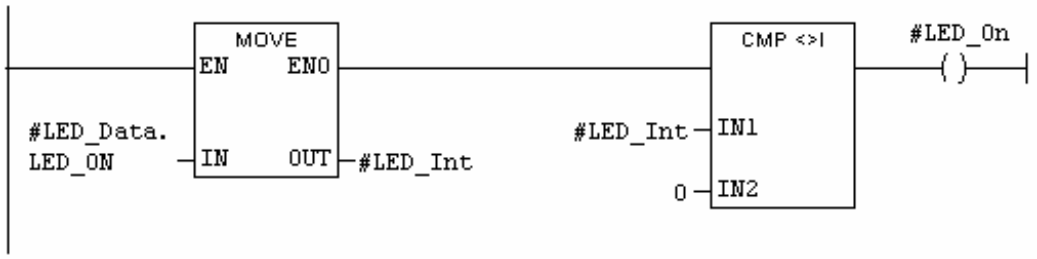
If SFC51 fails with SSL 0119, it returns -32638 (W#16#8082) in the RET_VAL parameter. Network 3 checks the RET_VAL. If W#16#0119 failed to get the LED status, FC112 calls SFC51 with SSL W#16#0174.



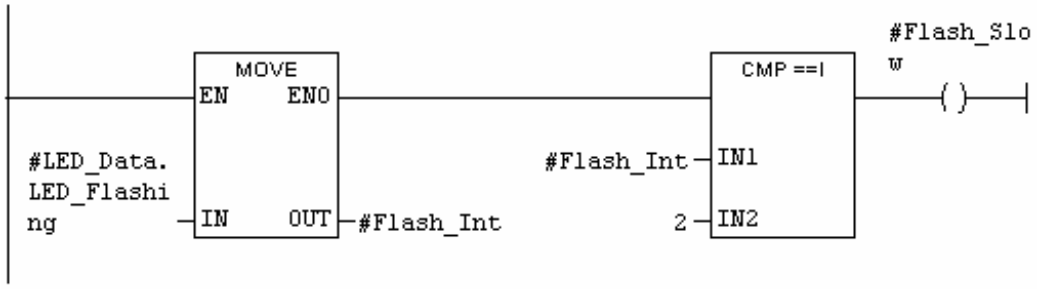
On success, SFC51 RET_VAL = 0 and the number of data records returned (SZL_HEADER.N_DR) = 1.

After getting the LED status, SFC112 calculates the output parameters for LED_On, Flash_Slow, and Flash_Fast.

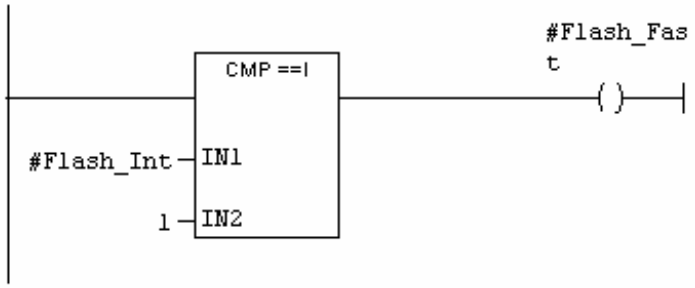
Network 4 : Title:



Network 5 : Title:

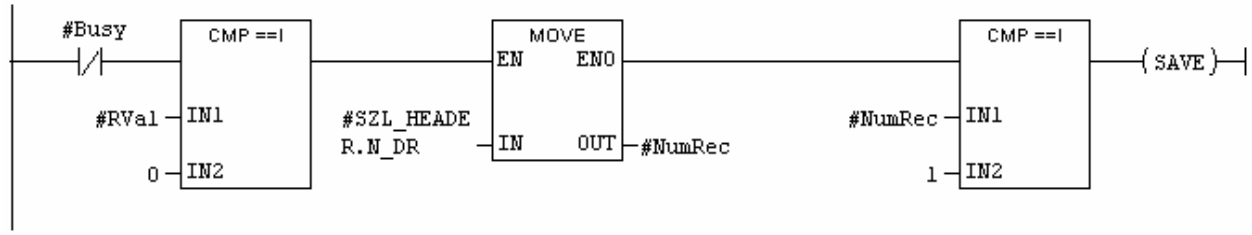


Network 6 : Title:



After setting the output parameters, FC112 sets the ENO state.

Network 7 : Use ENO to return success/fail state



Note SZL_HEADER.N_DR is a WORD value, and Compare instructions use only INT, DINT or REAL. The Move instruction stores the WORD value as an INT in Temp memory (#NumRec).

2 Reference Information

2.1 SFC51 ("RDSYSST")

Functional Description

With system function SFC51 "RDSYSST" (read system status), you read a system status list or a partial system status list.

You start the reading by assigning the value 1 to the input parameter REQ when SFC51 is called.

- If the system status could be read immediately, the SFC returns the value 0 at the BUSY output parameter.
- If BUSY has the value 1, the read function is not yet completed.

Note If you call SFC51 in the diagnostic interrupt OB with the SSL-ID W#16#00B1, W#16#00B2, or W#16#00B3 and access the module that initiated the diagnostic interrupt, the system status is read immediately.

With SFC51, only complete data records are transferred.

System Resources

If you start several asynchronous read functions one after the other at brief intervals, the operating system ensures that all the read jobs are executed and that they do not interfere with each other.

Tasks with the following SSL_IDs perform asynchronous read functions:

- W#16#00B4
- W#16#4C91
- W#16#4092
- W#16#4292
- W#16#4692
- Possibly W#16#00B1 and W#16#00B3

If the limits of the system resources are reached, this is indicated in RET_VAL. You can remedy this temporary error situation by repeating the job.

The maximum number of "simultaneously" active SFC51 jobs depends on the CPU. Refer to the hardware manuals for your S7 CPU.

Input and Output Parameters for SFC51

Table 2-1 Parameters of SFC51 (RDSYSST)

Parameter	Declaration	Data Type	Description
REQ	INPUT	BOOL	REQ=1: Starts processing
SSL_ID	INPUT	WORD	SSL-ID of the system status list or partial list to be read.
INDEX	INPUT	WORD	Type or number of an object in a partial list.
RET_VAL	OUTPUT	INT	If an error occurs while executing the SFC, the RET_VAL parameter contains an error code. Refer to the online help for STEP 7 or to the S7 documentation for information about the RET_VAL.
BUSY	OUTPUT	BOOL	TRUE: Reading not yet completed.
SSL_HEADER	OUTPUT	STRUCT	The SSL_HEADER parameter is a structure defined as follows: SSL_HEADER: STRUCT LENTHDR: WORD N_DR: WORD END_STRUCT LENTHDR is the length of a data record of the SSL list or the SSL partial list. • If you have only read out the header information of an SSL list, N_DR contains the number of data records belonging to it. • Otherwise, N_DR contains the number of data records transferred to the destination area.
DR	OUTPUT	ANY	Destination area of the SSL list read or the SSL partial list read:· • If you have only read out the header information of an SSL list, you must not evaluate DR but only SSL_HEADER. • Otherwise, the product of LENTHDR and N_DR indicates how many bytes were entered in DR.

2.2 SSL-ID W#16#xy19 – Status of the Module LEDS

If you read the partial list with SSL-ID W#16#xy19, you obtain the status of the module LEDs.

Refer to the STEP 7 online help for a full list of Index values, and refer to the documentation for your CPU for a list of the Index values supported by the CPU.

Note If you want to read out the partial list W#16#16#xy19 for an H CPU, remember that this is only possible in the non-redundant H operating modes.

Table 2-2 Structure of the header for SSL_ID W#16#xy19

Contents	Description
SSL-ID	The SSL-ID of the partial system status list • W#16#0019 Status of all LEDs • W#16#0119 Status of one LED
LENTHDR	W#16#0004: one data record is 2 words long (4 bytes)
N_DR	Number of data records

Table 2-3 Data Record for SSL_ID W#16#xy19

Name	Length	Description
Index	1 Word	LED ID (only relevant for SSL-ID W#16#0119) • W#16#0002: INTF (internal error) • W#16#0003: EXTF (external error) • W#16#0004: RUN • W#16#0005: STOP • W#16#0006: FRCE (force) • W#16#0007: CRST (restart) • W#16#0008: BAF (battery fault/overload, short circuit of battery voltage on bus) • W#16#0009: USR (user-defined) • W#16#000A: USR1 (user-defined) • W#16#000B: BUS1F (bus error interface 1) • W#16#000C: BUS2F (bus error interface 2) • W#16#000D: REDF (redundancy error) • W#16#000E: MSTR (master) • W#16#000F: RACK0 (rack number 0) • W#16#0010: RACK1 (rack number 1) • W#16#0011: RACK2 (rack number 2) • W#16#0012: IFM1F (interface error interface module 1) • W#16#0013: IFM2F (interface error interface module 2)
led_on	1 Byte	Status of the LED: • 0: Off • 1: On
led_blink	1 Byte	Flashing status of the LED: • 0: Not flashing • 1: Flashing normally (2 Hz) • 2: Flashing slowly (0.5 Hz)

2.3 SSL-ID W#16#xy74 – Status of the Module LEDS

If you read the partial list SSL-ID W#16#xy74, with standard CPUs (if present) and with the H CPUs, you obtain the status of the module LEDs. If the H CPUs are in a non-redundant H mode, you obtain the LED status of the CPU addressed. If the H CPUs are in the RUN-REDUNDANT mode, the LED status of all redundant H CPUs is returned.

Refer to the STEP 7 online help for a full list of Index values, and refer to the documentation for your CPU for a list of the Index values supported by the CPU.

Table 2-4 Structure of the header for SSL_ID W#16#0074

Contents	Description
SSL-ID	The SSL-ID of the partial list extract • W#16#0174 Status of an LED You select the LED with the INDEX parameter.
INDEX	LED ID (only relevant for SSL-ID W#16#0174) • W#16#0002: INTF (internal error) • W#16#0003: EXTF (external error) • W#16#0004: RUN • W#16#0005: STOP • W#16#0006: FRCE (force) • W#16#0007: CRST (restart) • W#16#0008: BAF (battery fault/overload, short circuit of battery voltage on bus) • W#16#0009: USR (user-defined) • W#16#000A: USR1 (user-defined) • W#16#000B: BUS1F (bus error interface 1) • W#16#000C: BUS2F (bus error interface 2) • W#16#000D: REDF (redundancy error) • W#16#000E: MSTR (master) • W#16#000F: RACK0 (rack number 0) • W#16#0010: RACK1 (rack number 1) • W#16#0011: RACK2 (rack number 2) • W#16#0012: IFM1F (interface error interface module 1) • W#16#0013: IFM2F (interface error interface module 2)
LENTHDR	W#16#0004: one data record is 2 words long (4 bytes)
N_DR	Number of data records

Table 2-5 Data Record for SSL_ID W#16#0074

Name	Length	Description
cpu_led_ID	1 Word	Byte 0: • Standard CPU: B#16#00 • H-CPU: - Bits 0 to 2: rack number - Bit 3: 0=standby CPU, 1=master CPU - Bits 4 to 7: 1111 Byte 1: LED ID
led_on	1 Byte	Status of the LED: • 0: Off • 1: On
led_blink	1 Byte	Flashing status of the LED: • 0: Not flashing • 1: Flashing normally (2 Hz) • 2: Flashing slowly (0.5 Hz)