On Control SUSA v1.15 MODBUS Functions Rev. 2018-11-27

Summary of functions:

Function code 0x03 ”Read holding registers” and
Function code 0x04 ”Read input registers” (both function codes are equivalent in SUSA):

Register address:	**Function:**
0x0000..0x0102 (0..258)	Read damper status
0x0200..0x0281 (512..641)	Read detector status
0x0300..0x048B (768..1163)	Read event log records
0x0500..0x050E (1280..1294)	Read single status flags from separate registers
0x0510 (1296)	Read all status flags as a single register
0x0600..0x0603 (1536..1539)	Read RTC date and time
0x1000..0x1004 (4096..4100)	Readback for register write verification
0x1100..0x111F (4352..4383)	Read external digital input status (e.g. KSUC)

Function code 0x06 ”Write single register” and
Function code 0x10 (16 decimal) ”Write multiple registers”:

Register address:	**Function:**
0x1000 (4096)	Trigger start of damper test
0x1001 (4097)	Trigger start of exhaust fan test
0x1002 (4098)	Alarm reset
0x1003 (4099)	Erase event log
0x1004 (4100)	Force day or night operation
0x1005 (4101)	Force system restart (CPU reset)

Function code 0x10 (16 decimal) ”Write multiple registers” *only*:

Register address:	**Function:**
0x600..0x603 (1536..1539)	Set RTC date and time

Function code 0x08 ”Diagnostics” with **subfunction code 0x0000**
responds with an exact copy of the request message.

Function code 0x2B (43 decimal) ”Encapsulated interface transport” is used to read product information.

Used exception codes:

0x01 = Function code not supported
0x02 = Illegal data address
0x03 = Illegal data value *(but not illegal register data!)*
0x04 = Unable to comply *(e.g. invalid register data)*

Register data mapping

Read damper status

Function code: 0x03 or 0x04

Register adress range: 0x0000..0x0102 (0..258) where

0x0000..0x0002 are local dampers 1..3 and
0x0003..0x0102 are external dampers 1..256

Register bit mapping:

Bit 15 (MSBit):	Always 0
Bit 14:	Always 0
Bit 13:	Always 0
Bit 12:	Always 0
Bit 11:	Always 0
Bit 10:	Always 0
Bit 9:	Always 0
Bit 8:	Always 0
Bit 7:	1 = Damper is busy with travel or damper test
Bit 6:	Always 0, reserved for forced day operation request from damper
Bit 5:	1 = Damper failed to reach OFF position in last damper test
Bit 4:	1 = Damper failed to reach ON position in last damper test
Bit 3:	1 = Damper is currently not in expected OFF position (error)
Bit 2:	1 = Damper is currently not in expected ON position (error)
Bit 1:	1 = Damper is currently in full OFF position (realtime monitoring)
Bit 0 (LSBbit):	1 = Damper is currently in full ON position (realtime monitoring)

Register data mapping (continued)

Read detector status

Function code: 0x03 or 0x04

Register adress range: 0x0200..0x0281 (512..641) where

0x0200..0x0201 are local detectors 1..2 and
0x0202..0x0281 are external detectors 1..128

Register bit mapping:

Bit 15 (MSBit):	MSBit of detector current
Bit 14:	
Bit 13:	(Current format: 8-bit unsigned)
Bit 12:	(Current unit: Milliamperes)
Bit 11:	
Bit 10:	
Bit 9:	
Bit 8:	LSBit of detector current
Bit 7:	Always 0
Bit 6:	Always 0
Bit 5:	Always 0
Bit 4:	Always 0
Bit 3:	Always 0
Bit 2:	1 = Detector service request (excessive idle current for a long time)
Bit 1:	1 = Detector failure (no current at all or current too low)
Bit 0 (LSBit):	1 = Fire alarm

Register data mapping (continued)

Read event log

Function code: 0x03 or 0x04

Register adress range: 0x0300..0x048B (768..1163) where 0x0300 corresponds to the oldest log record.

The log can contain up to 99 records (logged events). The log is circular i.e. a new event will overwrite the oldest record if the log is full.

Empty log records are returned with all 8 bytes cleared (0x00). This can be used by the Modbus master to detect end-of-log when the log is not full. The record is empty if the month (or day) byte = 0x00.

Special requirements:

1. The starting adress ***must*** be a multiple of 4 e.g. 0x0300, 0x0304, 0x0308 etc.
2. The register count ***must*** also be a multiple of 4 e.g. 0x0004, 0x0008, 0x000C etc.

Data mapping of each 4-register log record block:
Register_0_MsByte = Timestamp year, 0..99 (decimal)
Register_0_LsByte = Timestamp month, 1..12 (decimal)
Register_1_MsByte = Timestamp day, 1..31 (decimal)
Register_1_LsByte = Timestamp hour, 0..23 (decimal)
Register_2_MsByte = Timestamp minute, 0..59 (decimal)
Register_2_LsByte = Source identifier, see below
Register_3_MsByte = Event identifier, see below
Register_3_LsByte = Parameter, see below

All values are unsigned 8-bit binary (not BCD).

Source identifiers:
0 = Local dampers
1 = Local detectors
2 = External dampers
3 = External detectors
4 = Bus communication
5 = System

Event identifiers:
If source identifier = 0 (local) or 2 (external) dampers:

0 = Damper (P+1) failed to reach OFF position in damper test
1 = Damper (P+1) failed to reach ON position in damper test
2 = Damper (P+1) failed to reach both OFF and ON position in damper test
3 = Damper (P+1) not in expected OFF position during normal operation
4 = Damper (P+1) not in expected ON position during normal operation
5 = Damper (P+1) appeared to be in both OFF and ON positions simultaneously
Where P is the Parameter byte, for example Parameter=6 means damper number 7.

Register data mapping (continued)

Read event log (continued)

Event identifiers (continued):
If source identifier = 1 (local) or 3 (external) detectors:

0 = Detector (P+1) fire alarm
1 = Detector (P+1) failure
2 = Detector (P+1) servide request
Where P is the Parameter byte, for example Parameter=6 means detector number 7.

If source identifier = 4, bus communication:

0 = Slave (P) communication error e.g. response timeout or bad data
Where P is the Parameter byte containing slave address, 0..31.

If source identifier = 5, system:

0 = External fire alarm input activated
1 = RTC backup battery is low and should be replaced
2 = RTC has stopped
3 = RTC has been set by the user via the LCD menu system
4 = RTC data error detected, the RTC must be set again for correct operation
5 = System started (cold start from CPU reset)
6 = User logged in to the LCD menu system
7 = Damper test completed successfully
8 = Damper test failed
9 = Exhaust fan test completed successfully
10 = Exhaust fan test failed due to lack of pressure rise
11 = Exhaust fan test inhibited due to stuck damper
12 = Reserved code for forced damper opening (not used)
13 = External AUX input fire alarm activated for FG2
The Parameter byte is currently not used for system loggings (always 0x00).

Register data mapping (continued)

Read single status flags from separate registers

Function code: 0x03 or 0x04

Register adress range: 0x0500..0x050E (1280..1294)

Possible register values: 0x0001 if the flag is active, else 0x0000

Register adress:	**Flag:**
0x0500:	Fire alarm relay ouput is activated
0x0501:	Fault alarm relay ouput is activated
0x0502:	Fan 1 relay ouput is activated
0x0503:	Fan 2 relay ouput is activated
0x0504:	Hardware nighttime input is activated
0x0505:	Hardware external fire alarm input is activated
0x0506:	Hardware pressure sensor input is activated
0x0507:	Hardware auxilliary input (AUX) is activated
0x0508:	Communication error on damper bus (latching)
0x0509:	Exhaust fan test is in progress in FG1 (function group 1)
0x050A:	Exhaust fan test is in progress in FG2 (function group 2)
0x050B:	Damper test is in progress in FG1 (function group 1)
0x050C:	Damper test is in progress in FG2 (function group 2)
0x050D:	Exhaust fan test pending but still blocked by active AUX input
0x050E:	Damper test pending but still blocked by active AUX input

Register data mapping (continued)

Read all status flags as a single register

Function code: 0x03 or 0x04

Register adress: 0x0510 (1296)

Register bit:	**Flag:**
Bit 15 (MSBit):	Always 0
Bit 14:	1 = Damper test pending but still blocked by active AUX input
Bit 13:	1 = Exhaust fan test pending but still blocked by active AUX input
Bit 12:	1 = Damper test is in progress in FG2 (function group 2)
Bit 11:	1 = Damper test is in progress in FG1 (function group 1)
Bit 10:	1 = Exhaust fan test is in progress in FG2 (function group 2)
Bit 9:	1 = Exhaust fan test is in progress in FG1 (function group 1)
Bit 8:	1 = Communication error on damper bus (latching)
Bit 7:	1 = Hardware auxilliary input (AUX) is activated
Bit 6:	1 = Hardware pressure sensor input is activated
Bit 5:	1 = Hardware external fire alarm input is activated
Bit 4:	1 = Hardware nighttime input is activated
Bit 3:	1 = Fan 2 relay ouput is activated
Bit 2:	1 = Fan 1 relay ouput is activated
Bit 1:	1 = Fault alarm relay ouput is activated
Bit 0 (LSBbit):	1 = Fire alarm relay ouput is activated

Register data mapping (continued)

Readback of writable registers for verification

Function code: 0x03 or 0x04

Register adress range: 0x1000..0x1004 (4096..4101)

Register adress:	**Data:**
0x1000:	0x0001 = Damper test in progress or pending *(Note 1)*
0x1001:	0x0001 = Exhaust fan test in progress or pending *(Note 1)*
0x1002:	0x0001 = An alarm reset was recently written to this register *(Note 2)*
0x1003:	0x0001 = The event log is empty *(Note 3)*
0x1004:	0x0000..0x0002 = The last data written to this address

Note 1:
These bits indicate that a damper test or exhaust fan test is either already in progress or is pending i.e. waiting to start as soon as the test blocking signal on the hardware AUX input disappears. The registers will be cleared to 0x0000 as soon as the corresponding test is finished. The main purpose of these registers is to provide a feedback to verify that writes to the registers were successful.

Note 2:
Alarm reset is a transient event, not a ”setting”. After a successful write of 0x0001 to this register, it will return 0x0001 for 30 seconds after the write and then fall back to 0x0000.

Note 3:
A read from this register returns the ”log empty” status regardless if the last write to this register was successful or not.

Register data mapping (continued)

Read external digital input status (e.g. KSUC inputs)

Function code: 0x03 or 0x04

Register adress range: 0x1100..0x111F (4352..4383) where

Bits 0..15 at 0x1100 corresponds to external digital inputs 0..15
...
Bits 0..15 at 0x111F corresponds to external digital inputs 496..511

Register data mapping (continued)

Read RTC date and time

Function code: 0x03 or 0x04

Register adress range: 0x0600..0x0603 (1536..1539)

Special requirements:
1. The starting adress ***must*** be 0x0300
2. The register count ***must*** be 0x0004

Data mapping of the 4-register block:
Register_0_MsByte = Year MSByte, year = 2015..4095 (decimal)
Register_0_LsByte = Year LSByte
Register_1_MsByte = Month, 1..12 (decimal)
Register_1_LsByte = Day, 1..31 (decimal)
Register_2_MsByte = Hour, 0..23 (decimal)
Register_2_LsByte = Minute, 0..59 (decimal)
Register_3_MsByte = Second, 0..59 (decimal)
Register_3_LsByte = Day of week, 0..6, 0 = Sunday, 1 = Monday etc.

All values are unsigned binary (not BCD).
Year is a 16-bit binary value, all other values are 8-bit binary.

Set RTC date and time

Function code: 0x10 (16 decimal)

Register adress range: 0x0600..0x0603 (1536..1539)

Special requirements:
1. The starting adress ***must*** be 0x0300
2. The register count ***must*** be 0x0004

Data mapping of the 4-register block:
Register_0_MsByte = Year MSByte, year = 2015..4095 (decimal)
Register_0_LsByte = Year LSByte
Register_1_MsByte = Month, 1..12 (decimal)
Register_1_LsByte = Day, 1..31
Register_2_MsByte = Hour, 0..23 (decimal)
Register_2_LsByte = Minute, 0..59 (decimal)
Register_3_MsByte = Second, 0..59 (decimal)
Register_3_LsByte = Ignored (day-of-week is computed internally)

All values are unsigned binary (not BCD).
Year is a 16-bit binary value, all other values are 8-bit binary.

Register data mapping (continued)

Write triggers and day/night control

Function code: 0x06 or 0x10 (16 decimal)

Register adress range: 0x1000..0x1005 (4096..4101)

Register address:	**Function:**
0x1000 (4096):	Write 0x0001 to trigger start of damper test
0x1001 (4097):	Write 0x0001 to trigger start of exhaust fan test
0x1002 (4098):	Write 0x0001 to reset alarms
0x1003 (4099):	Write 0x0001 to erase the event log
0x1004 (4100):	Force day or night operation: Write 0x0000 for local day/night control via hardware input Write 0x0001 to force night operation Write 0x0002 to force day operation
0x1005 (4101):	Write 0x2BAD to trigger a total system restart (CPU reset)

Note: If function code 0x10 is used to write two or more of theses registers and address or register data error(s) occur with some of the register(s), the returned exception code will be that of the last failing register write i.e. some register writes may execute successfully while other writes fail.

Special functions

Diagnostics

Function code: 0x08

Request:

0x08	Function code
0x00	Sub-function MSByte
0x00	Sub-function LSByte
0x??	Any even number (0, 2, 4..250) of data bytes
0x??	
etc...	

Response:
Exactly the same as the request message.

Special functions (continued)

Read product information

Function code: 0x2B (43 decimal)

Request:

0x2B	Function code
0x0E	MEI type = Read device identification
0x01	Read device ID code = Basic device ID readout with stream access
0x00	Starting object ID, must be 0x00 here (VendorName)

Response:

0x2B	Function code (same as in request)
0x0E	MEI type (same as in request)
0x01	Read device ID code (same as in request)
0x01	Conformity level = Basic ID with stream access only
0x00	More follows = FALSE (all requested data is contained in this message)
0x00	Next object ID = 0x00 because ”More follows” = FALSE above
0x03	Number of objects = 3
0x00	Object ID 0 = VendorName
0x0D	Object length = 13 characters (in this example)
”On Control AB”	VendorName ASCII string, here 13 characters/bytes (in this example)
0x01	Object ID 1 = ProductCode
0x04	Object length = 4 characters (in this example)
”SUSA”	ProductCode ASCII string, here 4 characters/bytes (in this example)
0x02	Object ID 2 = MajorMinorRevision
0x05	Object length = 5 characters (in this example)
”v1.03”	MajorMinorRevision ASCII string here 5 characters/bytes (in this example)