KSUA MODBUS RTU INTERFACE - Commands and Responses

Document version: 2017-08-08
KSUA Firmware version: 3.0

TABLE OF CONTENTS

==========
FUNCTION/ADDRESS OVERVIEW
==========

Function code 0x01 - "Read Coils":

```
  0x0000          = READ NIGHTMODE FLAG AS COIL
  0x0001          = READ DAMPER TEST FLAG AS COIL
  0x0002          = READ SYSTEM RESET FLAG AS COIL (always returns ”TRUE”)
  0x0003          = READ EVAC FAN TEST FLAG AS COIL
  0x0004          = READ ALARM RESET FLAG AS COIL (always returns ”TRUE”)
```

Function code 0x03 - "Read Holding Registers" and
Function code 0x04 - "Read Input Registers":

(NOTE: Function codes 0x03 and 0x04 works exactly the same way. Any one, or both, can be used)

```
  0x0000 - 0x000F = READ DETECTOR STATUS COMPACT
  0x0100 - 0x0107 = READ DAMPER POSITIONS COMPACT
  0x0200 - 0x0203 = READ DETECTOR GROUP FIRE STATUS COMPACT
  0x0300          = READ INPUTS AND FLAGS COMPACT
  0x0400 - 0x0403 = READ REAL TIME CLOCK
  0x0500 - 0x053F = READ DETECTOR STATUS LONG
  0x0600 - 0x063F = READ DAMPER POSITIONS LONG
  0x0700 - 0x073F = READ DETECTOR GROUP FIRE STATUS LONG
  0x0800 - 0x0809 = READ INPUTS AND FLAGS LONG
  0x0900          = READ KSUC3 ALARM INPUTS COMPACT
  0x0A00 - 0x0A0F = READ KSUC3 ALARM INPUTS LONG
  0x0B00 - 0x0B01 = READ EXTERNAL DETECTOR STATUS COMPACT
  0x0C00 - 0x0C1F = READ EXTERNAL DETECTOR STATUS LONG
  0x0D00          = READ NIGHTMODE FLAG AS REGISTER
  0x0D01          = READ START DAMPER TEST FLAG AS REGISTER (FAKED '1')
  0x0D02          = READ SYSTEM RESET FLAG AS REGISTER (FAKED '1')
  0x0D03          = READ START E-FAN TEST FLAG AS REGISTER (FAKED '1')
  0x0D04          = READ ALARM RESET FLAG AS REGISTER (FAKED '1')
  0x0D05          = READ TIME CHANNEL FLAG REGISTER COMPACT
  0x0D06 - 0x0D0D = READ TIME CHANNEL FLAG REGISTER LONG
  0x0E00 - 0x0E07 = READ DAMPER TEST RESULTS
```

Function code 0x05 - "Write Single Coil":

```
  0x0000          = WRITE NIGHTMODE FLAG AS COIL
  0x0001          = START DAMPER TEST AS COIL
  0x0002          = SYSTEM RESET AS COIL
  0x0003          = START EVAC FAN TEST AS COIL
  0x0004          = ALARM RESET AS COIL
```

(continued on next page)

```
====================================
FUNCTION/ADDRESS OVERVIEW (continued)
====================================


Function code 0x06 - "Write Single Register":

  0x0D00          = WRITE NIGHTMODE FLAG AS REGISTER
  0x0D01          = START DAMPER TEST AS REGISTER
  0x0D02          = SYSTEM RESET AS REGISTER
  0x0D03          = START EVAC FAN TEST AS REGISTER
  0x0D04          = ALARM RESET AS REGISTER
  0x0D05          = WRITE TIME CHANNEL FLAG REGISTER COMPACT
  0x0D06 - 0x0D0D = WRITE TIME CHANNEL FLAG REGISTER ONE-BY-ONE


Function code 0x08 - ”Diagnostics”:
  Subfunction code "Return query" (0x0000) is implemented
  and will echo the incoming telegram exactly as received.


Function code 0x10 - "Write Multiple Registers":

  0x0400 - 0x0403 = WRITE REAL TIME CLOCK
  0x0D05 – 0x0D0D = WRITE TIME CHANNEL FLAG REGISTER LONG
```

READ DETECTOR STATUS (COMPACT)

See also "READ DETECTOR STATUS (LONG)" below.

16 registers (16-bit)
4 detectors per register
* = unused bits, read as zero

16-bit register read:
Function_code: 0x04 ("Read Input Registers") or 0x03
Start_address: Register address, 0x0000..0x000F
Register_count: 1..(16-Start_Address_LSByte)

Reg. Addr.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	< Bit number
0x0000		4	4	4		3	3	3		2	2	2		1	1	1	< Det. number
	*	FAIL	SERV	FIRE	*	FAIL	SERV	FIRE	*	FAIL	SERV	FIRE	*	FAIL	SERV	FIRE	< Det. flags
0x0001		8	8	8		7	7	7		6	6	6		5	5	5	
	*	FAIL	SERV	FIRE	*	FAIL	SERV	FIRE	*	FAIL	SERV	FIRE	*	FAIL	SERV	FIRE	
.....																	
0x000E		60	60	60		59	59	59		58	58	58		57	57	57	
	*	FAIL	SERV	FIRE	*	FAIL	SERV	FIRE	*	FAIL	SERV	FIRE	*	FAIL	SERV	FIRE	
0x000F		64	64	64		63	63	63		62	62	62		61	61	61	
	*	FAIL	SERV	FIRE	*	FAIL	SERV	FIRE	*	FAIL	SERV	FIRE	*	FAIL	SERV	FIRE	

Example MODBUS transaction for "READ DETECTOR STATUS (COMPACT)"
(CRC not shown):

Master request:
Byte 0 = Slave Address = 1..247
Byte 1 = Function Code = 4 or 3
Byte 2 = Starting Address MSB = 0
Byte 3 = Starting Address LSB = 0..15
Byte 4 = Register Count MSB = 0
Byte 5 = Register Count LSB = 1..(16-Start_Address_LSByte)

Slave response:
Byte 0 = Slave Address = 1..247
Byte 1 = Function Code = 4 or 3
Byte 2 = Total Register Bytecount = 2 * (Register Count) = 2..32
Byte 3 = Register MSB from starting address
Byte 4 = Register LSB from starting address
Byte 5 = Register MSB from starting address+1
Byte 6 = Register LSB from starting address+1
...
...
Byte ? = Register MSB from starting address+(Register Count-1)
Byte ? = Register LSB from starting address+(Register Count-1)

READ DAMPER POSITIONS (COMPACT)

See also "READ DAMPER POSITIONS (LONG)" below.

8 registers (16-bit)
8 dampers per register

16-bit register read:
Function_code: 0x04 ("Read Input Registers") or 0x03
Start_address: Register address, 0x0100..0x0107
Register_count: 1..(8-Start_Address_LSByte)

Reg. Addr.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	< Bit number
0x0100	8	8	7	7	6	6	5	5	4	4	3	3	2	2	1	1	< Damper num.
	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	< Pos. flags
0x0101	16	16	15	15	14	14	13	13	12	12	11	11	10	10	9	9	
	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	
.....	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
.....	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
.....	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
0x0106	56	56	55	55	54	54	53	53	52	52	51	51	50	50	49	49	
	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	
0x0107	64	64	63	63	62	62	61	61	60	60	59	59	58	58	57	57	
	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	

Example MODBUS transaction for "READ DAMPER POSITIONS (COMPACT)" (CRC not shown):

Master request:
Byte 0 = Slave Address = 1..247
Byte 1 = Function Code = 4 or 3
Byte 2 = Starting Address MSB = 1
Byte 3 = Starting Address LSB = 0..7
Byte 4 = Register Count MSB = 0
Byte 5 = Register Count LSB = 1..(8-Start_Address_LSByte)

Slave response:
Byte 0 = Slave Address = 1..247
Byte 1 = Function Code = 4 or 3
Byte 2 = Total Register Bytecount = 2 * (Register Count) = 2..16
Byte 3 = Register MSB from starting address
Byte 4 = Register LSB from starting address
Byte 5 = Register MSB from starting address+1
Byte 6 = Register LSB from starting address+1
...
...
Byte ? = Register MSB from starting address+(Register Count-1)
Byte ? = Register LSB from starting address+(Register Count-1)

READ DETECTOR GROUP FIRE STATUS (COMPACT)

See also "READ DETECTOR GROUP FIRE STATUS (LONG)" below.

4 registers (16-bit)
16 detector groups per register

16-bit register read:
Function_code: 0x04 ("Read Input Registers") or 0x03
Start_address: Register address, 0x0200..0x0203
Register_count: 1..(4-Start_Address_LSByte)

Reg. Addr.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	< Bit number
0x0200	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	< Group num.
	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	< Fire flags
0x0201	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	
	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	
0x0202	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	
	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	
0x0203	64	63	62	61	44	43	42	41	40	39	38	37	36	35	34	33	
	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	FIRE	

Example MODBUS transaction for "READ DETECTOR GROUP FIRE STATUS (COMPACT)" (CRC not shown):

Master request:
Byte 0 = Slave Address = 1..247
Byte 1 = Function Code = 4 or 3
Byte 2 = Starting Address MSB = 2
Byte 3 = Starting Address LSB = 0..3
Byte 4 = Register Count MSB = 0
Byte 5 = Register Count LSB = 1..(4-Start_Address_LSByte)

Slave response:
Byte 0 = Slave Address = 1..247
Byte 1 = Function Code = 4 or 3
Byte 2 = Total Register Bytecount = 2 * (Register Count) = 2..8
Byte 3 = Register MSB from starting address
Byte 4 = Register LSB from starting address
Byte 5 = Register MSB from starting address+1
Byte 6 = Register LSB from starting address+1
...
...
Byte ? = Register MSB from starting address+(Register Count-1)
Byte ? = Register LSB from starting address+(Register Count-1)

READ INPUTS AND FLAGS (COMPACT)

See also "READ INPUTS AND FLAGS (LONG)" below.

1 register (16-bit)
* = unused bits, read as zero

16-bit register read:
Function_code: 0x04 ("Read Input Registers") or 0x03
Start_address: Register address, 0x0300
Register_count: 1

Flag names:
"FIRE ALRM" = Fire Alarm Relay, 1 = Activated (Fire)
"SUM ALRM" = Sum Alarm Relay, 1 = Activated (Fault or Fire)
"VENT FAN" = Power Relay 1 (always VENT Fan), 1 = Relay ON
"POWR REL2" = Power Relay 2 (VENT Fan, EVAC Fan, Heater or EXT. FIRE ALARM), 1 = Relay ON
"EXT ALRM" = External Alarm Input on KSUA, 1 = Activated (Fire alarm!)
"FORC OPEN" = Forced Opening Input on KSUA, 1 = Activated (Open!)
"EXT NITE" = External Night Input on KSUA, 1 = Activated (Night!)
"SLAVE DAY" = KSUA forced to daytime mode by request from KSUB slave (if flag = 1)
"NITE FLAG" = Current KSUA mode, 1 = Night, 0 = Day
"DMPR TEST" = Damper test, and possibly also EVAC fan test, is in progress (if flag = 1)

Reg. Addr.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	< Bit number
0x0300	*	*	*	*	*	*	DMPR TEST	NITE FLAG	SLAV DAY	EXT NITE	FORC OPEN	EXT ALRM	POWR REL2	VENT FAN	SUM ALRM	FIRE ALRM	

Example MODBUS transaction for "READ INPUTS AND FLAGS (COMPACT)" (CRC not shown):

Master request:
Byte 0 = Slave Address = 1..247
Byte 1 = Function Code = 4 or 3
Byte 2 = Starting Address MSB = 3
Byte 3 = Starting Address LSB = 0
Byte 4 = Register Count MSB = 0
Byte 5 = Register Count LSB = 1

Slave response:
Byte 0 = Slave Address = 1..247
Byte 1 = Function Code = 4 or 3
Byte 2 = Total Register Bytecount = 2
Byte 3 = Register at 0x0300, MSB
Byte 4 = Register at 0x0300, LSB

READ/WRITE REAL TIME CLOCK

4 registers (16-bit)
* = unused bits, read/write as zero

All values are in binary format (not BCD!):
- SEC5..SEC0 = Seconds, 0..59
- MIN5..MIN0 = Minutes, 0..59
- HOUR4..HOUR0 = Hours, 0..23
- WDAY2..WDAY0 = Weekday, 0..6 where 0 is Sunday
- DATE4..DATE0 = Date, 1..31
- MONT3..MONT0 = Month, 1..12
- YEAR6..YEAR0 = Year, 0..99

16-bit register write:
- Function_code: 0x10 ("Write Multiple Regsisters")
- Start_address: Register address, 0x0400
- Register_count: 4 (must be 4!)

16-bit register read:
- Function_code: 0x04 ("Read Input Registers") or 3
- Start_address: Register address, 0x0400
- Register_count: 4 (must be 4!)

Reg. Addr.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	< Bit number
0x0400	*	*	MIN 5	MIN 4	MIN 3	MIN 2	MIN 1	MIN 0	*	*	SEC 5	SEC 4	SEC 3	SEC 2	SEC 1	SEC 0	
0x0401	*	*	*	*	*	WDAY 2	WDAY 1	WDAY 0	*	*	*	HOUR 4	HOUR 3	HOUR 2	HOUR 1	HOUR 0	
0x0402	*	*	*	*	MONT 3	MONT 2	MONT 1	MONT 0	*	*	*	DATE 4	DATE 3	DATE 2	DATE 1	DATE 0	
0x0403	*	*	*	*	*	*	*	*	YEAR 6	YEAR 5	YEAR 4	YEAR 3	YEAR 2	YEAR 1	YEAR 0		

(continued on next page)

READ/WRITE REAL TIME CLOCK (continued)

Example MODBUS transaction for "READ REAL TIME CLOCK" (CRC not shown):

Master request:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 4 or 3
- Byte 2 = Starting Address MSB = 4
- Byte 3 = Starting Address LSB = 0
- Byte 4 = Register Count MSB = 0
- Byte 5 = Register Count LSB = 4

Slave response:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 4 or 3
- Byte 2 = Total Register Bytecount = 8
- Byte 3 = Register at 0x0400, MSB
- Byte 4 = Register at 0x0400, LSB
- Byte 5 = Register at 0x0401, MSB
- Byte 6 = Register at 0x0401, LSB
- Byte 7 = Register at 0x0402, MSB
- Byte 8 = Register at 0x0402, LSB
- Byte 9 = Register at 0x0403, MSB
- Byte10 = Register at 0x0403, LSB

Example MODBUS transaction for "WRITE REAL TIME CLOCK" (CRC not shown):

Master request:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 16 (0x10)
- Byte 2 = Starting Address MSB = 4
- Byte 3 = Starting Address LSB = 0
- Byte 4 = Register Count MSB = 0
- Byte 5 = Register Count LSB = 4
- Byte 6 = Total Register Bytecount = 8
- Byte 7 = MINUTES (0..59 in binary format)
- Byte 8 = SECONDS (0..59 in binary format)
- Byte 9 = WEEKDAY (0..6 in binary format)
- Byte10 = HOUR (0..23 in binary format)
- Byte11 = MONTH (1..12 in binary format)
- Byte12 = DATE (1..31 in binary format)
- Byte13 = 0 (not used but keep this byte cleared for future compatibility!)
- Byte14 = YEAR (0..99 in binary format)

Slave response:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 16 (0x10)
- Byte 2 = Starting Address MSB = 4
- Byte 3 = Starting Address LSB = 0
- Byte 4 = Register Count MSB = 0
- Byte 5 = Register Count LSB = 4

READ DETECTOR STATUS (LONG)

See also "READ DETECTOR STATUS LONG (COMPACT)" above.

64 registers (16-bit)
1 detector per register
* = unused bits, read as zero
Detector flag polarity: Logic '1' = TRUE

16-bit register read:
Function_code: 0x04 ("Read Input Registers") or 0x03
Start_address: Register address, 0x0500..0x053F
Register_count: 1..(64-Start_Address_LSByte)

Reg. Addr.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	< Bit number
0x0500														1	1	1	< Det. number
	*	*	*	*	*	*	*	*	*	*	*	*	*	FAIL	SERV	FIRE	< Det. flags
0x0501														2	2	2	
	*	*	*	*	*	*	*	*	*	*	*	*	*	FAIL	SERV	FIRE	
.....	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
.....	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
.....	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
0x053E														63	63	63	
	*	*	*	*	*	*	*	*	*	*	*	*	*	FAIL	SERV	FIRE	
0x053F														64	64	64	
	*	*	*	*	*	*	*	*	*	*	*	*	*	FAIL	SERV	FIRE	

Example MODBUS transaction for "READ DETECTOR STATUS (LONG)" (CRC not shown):

Master request:
Byte 0 = Slave Address = 1..247
Byte 1 = Function Code = 4 or 3
Byte 2 = Starting Address MSB = 5
Byte 3 = Starting Address LSB = 0..63
Byte 4 = Register Count MSB = 0
Byte 5 = Register Count LSB = 1..(64-Start_Address_LSByte)

Slave response:
Byte 0 = Slave Address = 1..247
Byte 1 = Function Code = 4 or 3
Byte 2 = Total Register Bytecount = 2 * (Register Count) = 2..128
Byte 3 = Register MSB from starting address
Byte 4 = Register LSB from starting address
Byte 5 = Register MSB from starting address+1
Byte 6 = Register LSB from starting address+1
...
...
Byte ? = Register MSB from starting address+(Register Count-1)
Byte ? = Register LSB from starting address+(Register Count-1)

==========
READ DAMPER POSITIONS (LONG)
==========

See also "READ DAMPER POSITIONS (COMPACT)" above.

64 registers (16-bit)
1 damper per register
* = unused bits, read as zero
Damper flag polarity: Logic '1' = TRUE

16-bit register read:
Function_code: 0x04 ("Read Input Registers") or 0x03
Start_address: Register address, 0x0600..0x063F
Register_count: 1..(64-Start_Address_LSByte)

Reg. Addr.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	< Bit number
0x0600															1	1	< Damper num.
	*	*	*	*	*	*	*	*	*	*	*	*	*	*	ON	OFF	< Pos. flags
0x0601															2	2	
	*	*	*	*	*	*	*	*	*	*	*	*	*	*	ON	OFF	
.....	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
.....	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
.....	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
0x063E															63	63	
	*	*	*	*	*	*	*	*	*	*	*	*	*	*	ON	OFF	
0x063F															64	64	
	*	*	*	*	*	*	*	*	*	*	*	*	*	*	ON	OFF	

Example MODBUS transaction for "READ DAMPER POSITIONS (LONG)" (CRC not shown):

Master request:
Byte 0 = Slave Address = 1..247
Byte 1 = Function Code = 4 or 3
Byte 2 = Starting Address MSB = 6
Byte 3 = Starting Address LSB = 0..63
Byte 4 = Register Count MSB = 0
Byte 5 = Register Count LSB = 1..(64-Start_Address_LSByte)

Slave response:
Byte 0 = Slave Address = 1..247
Byte 1 = Function Code = 4 or 3
Byte 2 = Total Register Bytecount = 2 * (Register Count) = 2..128
Byte 3 = Register MSB from starting address
Byte 4 = Register LSB from starting address
Byte 5 = Register MSB from starting address+1
Byte 6 = Register LSB from starting address+1
...
...
Byte ? = Register MSB from starting address+(Register Count-1)
Byte ? = Register LSB from starting address+(Register Count-1)

READ DETECTOR GROUP FIRE STATUS (LONG)

See also "READ DETECTOR GROUP FIRE STATUS (COMPACT)" above.

64 registers (16-bit)
1 detector group per register
* = unused bits, read as zero

16-bit register read:
Function_code: 0x04 ("Read Input Registers") or 0x03
Start_address: Register address, 0x0700..0x073F
Register_count: 1..(64-Start_Address_LSByte)

Reg. Addr.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	< Bit number
0x0700	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	1	< Group num.
																FIRE	< Fire flag
0x0701	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	2	
																FIRE	
.....	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
0x073E	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	63	
																FIRE	
0x073F	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	64	
																FIRE	

Example MODBUS transaction for "READ DETECTOR GROUP FIRE STATUS (LONG)" (CRC not shown):

Master request:
Byte 0 = Slave Address = 1..247
Byte 1 = Function Code = 4 or 3
Byte 2 = Starting Address MSB = 7
Byte 3 = Starting Address LSB = 0..63
Byte 4 = Register Count MSB = 0
Byte 5 = Register Count LSB = 1..(64-Start_Address_LSByte)

Slave response:
Byte 0 = Slave Address = 1..247
Byte 1 = Function Code = 4 or 3
Byte 2 = Total Register Bytecount = 2 * (Register Count) = 2..128
Byte 3 = Register MSB from starting address
Byte 4 = Register LSB from starting address
Byte 5 = Register MSB from starting address+1
Byte 6 = Register LSB from starting address+1
...
...
Byte ? = Register MSB from starting address+(Register Count-1)
Byte ? = Register LSB from starting address+(Register Count-1)

==========
READ INPUTS AND FLAGS (LONG)
==========

See also "READ INPUTS AND FLAGS (COMPACT)" above.

10 registers (16-bit)
* = unused bits, read as zero

16-bit register read:
Function_code: 0x04 ("Read Input Registers") or 0x03
Start_address: Register address, 0x0800..0x809
Register_count: 1..(10-Start_Address_LSByte)

Flag names:
"FIRE ALRM" = Fire Alarm Relay, 1 = Activated (Fire)
"SUM ALRM" = Sum Alarm Relay, 1 = Activated (Fault or Fire)
"VENT FAN" = Power Relay 1 (always VENT Fan), 1 = Relay ON
"POWR REL2" = Power Relay 2 (VENT Fan, EVAC Fan, Heater or EXT. FIRE ALARM), 1 = Relay ON
"EXT ALRM" = External Alarm Input on KSUA, 1 = Activated (Fire alarm!)
"FORC OPEN" = Forced Opening Input on KSUA, 1 = Activated (Open!)
"EXT NITE" = External Night Input on KSUA, 1 = Activated (Night!)
"SLAVE DAY" = KSUA forced to daytime mode by request from KSUB slave (if flag = 1)
"NITE FLAG" = Current KSUA mode, 1 = Night, 0 = Day
"DMPR TEST" = Damper test, and possibly also EVAC fan test, is in progress (if flag = 1)

Reg. Addr.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	< Bit number
0x0800	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	FIRE ALRM	
0x0801	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	SUM ALRM	
0x0802	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	VENT FAN	
0x0803	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	POWR REL2	
0x0804	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	EXT ALRM	
0x0805	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	FORC OPEN	
0x0806	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	EXT NITE	
0x0807	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	SLAV DAY	
0x0808	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	NITE FLAG	
0x0809	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	DMPR TEST	

(continued on next page)

READ INPUTS AND FLAGS (LONG) (continued)

Example MODBUS transaction for "READ INPUTS AND FLAGS (LONG)" (CRC not shown):

Master request:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 4 or 3
- Byte 2 = Starting Address MSB = 8
- Byte 3 = Starting Address LSB = 0..9
- Byte 4 = Register Count MSB = 0
- Byte 5 = Register Count LSB = 1..(10-Start_Address_LSByte)

Slave response:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 4 or 3
- Byte 2 = Total Register Bytecount = 2 * (Register Count) = 2..20
- Byte 3 = Register MSB from starting address
- Byte 4 = Register LSB from starting address
- Byte 5 = Register MSB from starting address+1
- Byte 6 = Register LSB from starting address+1
- ...
- ...
- Byte ? = Register MSB from starting address+(Register Count-1)
- Byte ? = Register LSB from starting address+(Register Count-1)

==========
READ KSUC3 ALARM INPUTS (COMPACT)
==========

See also "READ KSUC3 ALARM INPUTS (LONG)" below.

1 register (16-bit)

16-bit register read:
Function_code: 0x04 ("Read Input Registers") or 0x03
Start_address: Register address, 0x0900
Register_count: 1

KSUC3 Alarm Flags:
1 = Alarm
0 = No Alarm

Reg. Addr.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	< Bit number
0x0900	FLAG #16	FLAG #15	FLAG #14	FLAG #13	FLAG #12	FLAG #11	FLAG #10	FLAG #9	FLAG #8	FLAG #7	FLAG #6	FLAG #5	FLAG #4	FLAG #3	FLAG #2	FLAG #1	

Example MODBUS transaction for "READ KSUC3 ALARM INPUTS (COMPACT)" (CRC not shown):

Master request:
Byte 0 = Slave Address = 1..247
Byte 1 = Function Code = 4 or 3
Byte 2 = Starting Address MSB = 9
Byte 3 = Starting Address LSB = 0
Byte 4 = Register Count MSB = 0
Byte 5 = Register Count LSB = 1

Slave response:
Byte 0 = Slave Address = 1..247
Byte 1 = Function Code = 4 or 3
Byte 2 = Total Register Bytecount = 2
Byte 3 = Register at 0x0900, MSB
Byte 4 = Register at 0x0900, LSB

==========
READ KSUC3 ALARM INPUTS (LONG)
==========

See also "READ KSUC3 ALARM INPUTS (COMPACT)" above.

16 registers (16-bit)
* = unused bits, read as zero

16-bit register read:
Function_code: 0x04 ("Read Input Registers") or 0x03
Start_address: Register address, 0x0A00..0xA0F
Register_count: 1..(16-Start_Address_LSByte)

KSUC3 Alarm Flags:
1 = Alarm
0 = No Alarm

Reg. Addr.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	< Bit number
0x0A00	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	FLAG #1	
0x0A01	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	FLAG #2	
0x0A02	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	FLAG #3	
0x0A03	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	FLAG #4	
0x0A04	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	FLAG #5	
0x0A05	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	FLAG #6	
0x0A06	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	FLAG #7	
0x0A07	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	FLAG #8	
0x0A08	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	FLAG #9	
0x0A09	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	FLAG #10	
0x0A0A	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	FLAG #11	
0x0A0B	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	FLAG #12	
0x0A0C	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	FLAG #13	
0x0A0D	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	FLAG #14	
0x0A0E	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	FLAG #15	
0x0A0F	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	FLAG #16	

(continued on next page)

READ KSUC3 ALARM INPUTS (LONG) (continued)

Example MODBUS transaction for "READ KSUC3 ALARM INPUTS (LONG)" (CRC not shown):

Master request:

- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 4 or 3
- Byte 2 = Starting Address MSB = 10 (0x0A)
- Byte 3 = Starting Address LSB = 0..15
- Byte 4 = Register Count MSB = 0
- Byte 5 = Register Count LSB = 1..(16-Start_Address_LSByte)

Slave response:

- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 4 or 3
- Byte 2 = Total Register Bytecount = 2 * (Register Count) = 2..32
- Byte 3 = Register MSB from starting address
- Byte 4 = Register LSB from starting address
- Byte 5 = Register MSB from starting address+1
- Byte 6 = Register LSB from starting address+1
- ...
- ...
- Byte ? = Register MSB from starting address+(Register Count-1)
- Byte ? = Register LSB from starting address+(Register Count-1)

READ EXTERNAL DETECTOR STATUS (COMPACT)

See also "READ EXTERNAL DETECTOR STATUS (LONG)" below.

2 registers (16-bit)

16-bit register read:
Function_code: 0x04 ("Read Input Registers") or 0x03
Start_address: Register address, 0x0B00..0x0B01
Register_count: 1..(2-Start_Address_LSByte)

External detector flags:
1 = Alarm
0 = No Alarm

Reg. Addr.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	< Bit number
0x0B00	80e	79e	78e	77e	76e	75e	74e	73e	72e	71e	70e	69e	68e	67e	66e	65e	< From KSUC1
0x0B01	96e	95e	94e	93e	92e	91e	90e	89e	88e	87e	86e	85e	84e	83e	82e	81e	< From KSUC2

Example MODBUS transaction for "READ EXTERNAL DETECTOR STATUS (COMPACT)" (CRC not shown):

Master request:
Byte 0 = Slave Address = 1..247
Byte 1 = Function Code = 4 or 3
Byte 2 = Starting Address MSB = 11 (0x0B)
Byte 3 = Starting Address LSB = 0
Byte 4 = Register Count MSB = 0..1
Byte 5 = Register Count LSB = 1..(2-Start_Address_LSByte)

Slave response:
Byte 0 = Slave Address = 1..247
Byte 1 = Function Code = 4 or 3
Byte 2 = Total Register Bytecount = 2 * (Register Count) = 2..4
Byte 3 = Register MSB from starting address
Byte 4 = Register LSB from starting address
If Register Count = 2:
Byte 5 = Register MSB from starting address+1
Byte 6 = Register LSB from starting address+1

==========
READ EXTERNAL DETECTOR STATUS (LONG)
==========

See also "READ EXTERNAL DETECTOR STATUS (COMPACT)" above.

32 registers (16-bit)
* = unused bits, read as zero

16-bit register read:
Function_code: 0x04 ("Read Input Registers") or 0x03
Start_address: Register address, 0x0C00..0x0C1F
Register_count: 1..(32-Start_Address_LSByte)

External detector (KSUC1, KSUC2) alarm Flags:
1 = Alarm
0 = No Alarm

Reg. Addr.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	< Bit number
0x0C00	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	65e FIRE	< Ext. det. number
0x0C01	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	66e FIRE	
.....	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
.....	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
.....	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
0x0C1E	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	95e FIRE	
0x0C1F	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	96e FIRE	

Example MODBUS transaction for "READ EXTERNAL DETECTOR STATUS (LONG)" (CRC not shown):

Master request:
Byte 0 = Slave Address = 1..247
Byte 1 = Function Code = 4 or 3
Byte 2 = Starting Address MSB = 12 (0x0C)
Byte 3 = Starting Address LSB = 0..31
Byte 4 = Register Count MSB = 0
Byte 5 = Register Count LSB = 1..(32-Start_Address_LSByte)

Slave response:
Byte 0 = Slave Address = 1..247
Byte 1 = Function Code = 4 or 3
Byte 2 = Total Register Bytecount = 2 * (Register Count) = 2..64
Byte 3 = Register MSB from starting address
Byte 4 = Register LSB from starting address
Byte 5 = Register MSB from starting address+1
Byte 6 = Register LSB from starting address+1
...
...
Byte ? = Register MSB from starting address+(Register Count-1)
Byte ? = Register LSB from starting address+(Register Count-1)

==========
READ DAMPER TEST RESULTS
==========

NOTES:
- This readout is only available in "compact" format.
- The "Alarm Reset" command can be used to clear these flags when KSUA is in normal (non-critical) mode.

8 registers (16-bit)
* = unused bits, read as zero

16-bit register read:
Function_code: 0x04 ("Read Input Registers") or 0x03
Start_address: Register address, 0x0E00..0x0E07
Register_count: 1..(8-Start_Address_LSByte)

Damper test failure flags:
1 = Failure
0 = Success
OFF = If 1, the damper failed to reach OFF position during test
ON = If 1, the damper failed to reach ON position during test

Reg. Addr.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	< Bit number
0x0E00	8	8	7	7	6	6	5	5	4	4	3	3	2	2	1	1	< Damper number
	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	< Failure flags
0x0E01	16	16	15	15	14	14	13	13	12	12	11	11	10	10	9	9	
	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	
.....	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
.....	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
.....	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
0x0E06	56	56	55	55	54	54	53	53	52	52	51	51	50	50	49	49	
	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	
0x0E07	64	64	63	63	62	62	61	61	60	60	59	59	58	58	57	57	
	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	

(continued on next page)

READ DAMPER TEST RESULTS (continued)

Example MODBUS transaction for "READ EXTERNAL DETECTOR STATUS (LONG)" (CRC not shown):

Master request:

- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 4 or 3
- Byte 2 = Starting Address MSB = 14 (0x0E)
- Byte 3 = Starting Address LSB = 0..7
- Byte 4 = Register Count MSB = 0
- Byte 5 = Register Count LSB = 1..(8-Start_Address_LSByte)

Slave response:

- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 4 or 3
- Byte 2 = Total Register Bytecount = 2 * (Register Count) = 2..16
- Byte 3 = Register MSB from starting address
- Byte 4 = Register LSB from starting address
- Byte 5 = Register MSB from starting address+1
- Byte 6 = Register LSB from starting address+1
- ...
- ...
- Byte ? = Register MSB from starting address+(Register Count-1)
- Byte ? = Register LSB from starting address+(Register Count-1)

READ/WRITE VARIOUS CONTROL AND STATUS BITS IN COIL MODE

See also READ/WRITE VARIOUS CONTROL AND STATUS BITS IN REGISTER MODE" below.

Single coil write (MODBUS Nightmode Control flag):
Function_code: 0x05 ("Write Single Coil")
Output_address: 0x0000
Output_value: 0xFF00 for NIGHT, 0x0000 for DAY

Single coil write (Start Damper Test):
Function_code: 0x05 ("Write Single Coil")
Output_address: 0x0001
Output_value: 0xFF00

Single coil write (System Reset):
Function_code: 0x05 ("Write Single Coil")
Output_address: 0x0002
Output_value: 0xFF00

Single coil write (Start EVAC fan test):
Function_code: 0x05 ("Write Single Coil")
Output_address: 0x0003
Output_value: 0xFF00

Single coil write (Alarm Reset):
Function_code: 0x05 ("Write Single Coil")
Output_address: 0x0004
Output_value: 0xFF00

Coil read (All 5 flags above, multiple coil read is supported):
Function_code: 0x01 ("Read Coils")
Start_address: 0x0000..0x0004
Coil_count: 1..5 (maximum is 5 - Start_address)

Coil address	Coil function	Read	Write
0x0000	MODBUS Nightmode control flag	YES	YES
0x0001	Start Damper Test	YES	YES
0x0002	System Reset (see notes below)	YES	YES
0x0003	Start EVAC fan test	YES	YES
0x0004	Alarm Reset (see notes below)	YES	YES

Notes:
"Alarm Reset" works only when KSUA is in normal ("non-critical") mode!
In all other cases, the "System Reset" must be used.

A read of the coils "System reset" and "Alarm reset" will always return a faked logic 1 to satisfy ModBus masters that make a verification readback after writing (writing these two coils are actually ”trigger event commands” that have no real readback values).

(continued on next page)

READ/WRITE VARIOUS CONTROL AND STATUS BITS IN COIL MODE (continued)

Example MODBUS transaction for "Write MODBUS nightmode control flag" (CRC not shown):

Master request:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 5
- Byte 2 = Output Address MSB = 0
- Byte 3 = Output Address LSB = 0
- Byte 4 = Output Value MSB = 0 for "DAY", 255 for "NIGHT"
- Byte 5 = Output Value LSB = 0

Slave response:
Exactly the same as the master request.

Example MODBUS transaction for "Start damper test" (CRC not shown):

Master request:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 5
- Byte 2 = Output Address MSB = 0
- Byte 3 = Output Address LSB = 1
- Byte 4 = Output Value MSB = 255 (0 is also a valid value but does *NOT* start the test!)
- Byte 5 = Output Value LSB = 0

Slave response:
Exactly the same as the master request.

Example MODBUS transaction for "System reset" (CRC not shown):

Master request:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 5
- Byte 2 = Output Address MSB = 0
- Byte 3 = Output Address LSB = 2
- Byte 4 = Output Value MSB = 255 (0 is also a valid value but does *NOT* reset the system!)
- Byte 5 = Output Value LSB = 0

Slave response:
Exactly the same as the master request.

(continued on next page)

```
=====================================================================
READ/WRITE VARIOUS CONTROL AND STATUS BITS IN COIL MODE (continued)
=====================================================================

  Example MODBUS transaction for "Start EVAC fan test" (CRC not shown):

  Master request:
    Byte 0 = Slave Address = 1..247
    Byte 1 = Function Code = 5
    Byte 2 = Output Address MSB = 0
    Byte 3 = Output Address LSB = 3
    Byte 4 = Output Value MSB = 255 (0 is also a valid value but
                                     does *NOT* start the test!)
    Byte 5 = Output Value LSB = 0

  Slave response:
    Exactly the same as the master request.


  Example MODBUS transaction for "Alarm reset" (CRC not shown):

  Master request:
    Byte 0 = Slave Address = 1..247
    Byte 1 = Function Code = 5
    Byte 2 = Output Address MSB = 0
    Byte 3 = Output Address LSB = 4
    Byte 4 = Output Value MSB = 255 (0 is also a valid value but
                                     does *NOT* reset the alarms!)
    Byte 5 = Output Value LSB = 0

  Slave response:
    Exactly the same as the master request.

Note:
  "Alarm Reset" works only when KSUA is in normal ("non-critical") mode!
  In all other cases, the "System Reset" command must be used.
```

(continued on next page)

READ/WRITE VARIOUS CONTROL AND STATUS BITS IN COIL MODE (continued)

Example MODBUS transaction for "Read all 5 flags"
(CRC not shown):

Master request:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 1
- Byte 2 = Starting Address MSB = 0
- Byte 3 = Starting Address LSB = 0 (see coil address table on page 22)
- Byte 4 = Number of coils MSB = 0
- Byte 5 = Number of coils LSB = 5

Slave response:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 1
- Byte 2 = Bytecount = 1
- Byte 3 = All 5 flags in bits 0..4, bits 5..7 = 0

When reading more than one coil with a single command, the coil bits are packed into the lsBits of slave response byte 3 with the lowest address coil starting in bit 0 (lsBit):

	Number of coils read:				
	1	**2**	**3**	**4**	**5**
	Bit	**Bit**	**Bit**	**Bit**	**Bit**
Address	**76543210**	**76543210**	**76543210**	**76543210**	**76543210**
0x0000	ooooooooN	ooooooTN	oooooRTN	ooooERTN	oooAERTN
0x0001	ooooooooT	ooooooRT	oooooERT	ooooAERT	xxxxxxxx
0x0002	ooooooooR	ooooooER	oooooAER	xxxxxxxx	xxxxxxxx
0x0003	ooooooooE	ooooooAE	xxxxxxxx	xxxxxxxx	xxxxxxxx
0x0004	ooooooooA	xxxxxxxx	xxxxxxxx	xxxxxxxx	xxxxxxxx

where:
- N = ModBus nightmode control flag (coil address 0x0000)
- T = Damper test in progress flag (coil address 0x0001
- R = System reset (always read as 1) (coil address 0x0002)
- E = EVAC fan test in progress flag (coil address 0x0003)
- A = Alarm reset (always read as 1) (coil address 0x0004)
- xxxxxxxx = invalid combination of starting address and number of coils

Notes:

Once a damper test or EVAC fan test has been started, the corresponding readback flag will stay active (1) until the test has finished.

A read of the ”ModBus nightmode control flag” returns the the last value written to this flag by the ModBus master. The actual day/night state of KSUA may be different as day/night modes can be forced by a number of signals where ModBus control has the lowest priority:

Priority	**Signal**
1	Day mode requested by damper slave
2	Night mode requested by hardware input on KSUA
3	Day mode forced due to KSUA RTC (real time clock) failure
4	RTC controlled day/night schedule is enabled
5	ModBus nightmode control flag

==========
READ/WRITE VARIOUS CONTROL AND STATUS BITS IN REGISTER MODE
==========

See also READ/WRITE VARIOUS CONTROL AND STATUS BITS IN COIL MODE" above.

Read MODBUS nightmode control flag:
1 register (16-bit)

16-bit register read:
Function_code: 0x04 ("Read Input Registers") or 0x03
Start_address: Register address, 0x0D00
Register_count: 1

Write MODBUS nightmode control flag:
1 register (16-bit)

16-bit register write:
Function_code: 0x06 ("Write Single Register")
Start_address: Register address, 0x0D00
Register_data: 0x0000 (day) or 0x0001 (night)

Start damper test:
1 register (16-bit)

16-bit register write:
Function_code: 0x06 ("Write Single Register")
Start_address: Register address, 0x0D01
Register_data: 0x0001

System reset:
1 register (16-bit)

16-bit register write:
Function_code: 0x06 ("Write Single Register")
Start_address: Register address, 0x0D02
Register_data: 0x0001

Start EVAC fan test:
1 register (16-bit)

16-bit register write:
Function_code: 0x06 ("Write Single Register")
Start_address: Register address, 0x0D03
Register_data: 0x0001

Alarm reset:
1 register (16-bit)

16-bit register write:
Function_code: 0x06 ("Write Single Register")
Start_address: Register address, 0x0D04
Register_data: 0x0001

Note:
"Alarm Reset" works only when KSUA is in normal ("non-critical") mode!
In all other cases, the "System Reset" command must be used.

(continued on next page)

```
========================================================================
READ/WRITE VARIOUS CONTROL AND STATUS BITS IN REGISTER MODE (continued)
========================================================================


  Read Time Channel Flag Register (COMPACT):
    1 register (16-bit)

    16-bit register read:
      Function_code:  0x04 ("Read Input Registers") or 0x03
      Start_address:  Register address, 0x0D05
      Register_count: 1

  Read Time Channel Flag Register (LONG):
    1..8 registers (16-bit)

    16-bit register read:
      Function_code:  0x04 ("Read Input Registers") or 0x03
      Start_address:  Register address, 0x0D06..0x0D0D
      Register_count: 1..(0x0D0E-Start_address) (max 8)

  Write Time Channel Flag Register (COMPACT):
    1 register (16-bit)

    16-bit register write:
      Function_code:  0x06 ("Write Single Register")
      Start_address:  Register address, 0x0D05
      Register_data:  0x00tt (tt = time channel flags, bit 0 = TCH1)

  Write Time Channel Flag Register (ONE-BY-ONE):
    1 register (16-bit)

    16-bit register write:
      Function_code:  0x06 ("Write Single Register")
      Start_address:  Register address, 0x0D06..0x0D0D (for TCH1..TCH8)
      Register_data:  0x0000 => Time Channel OFF (normal) or
                      0x0001 => Time Channel ON (close associated dampers)
```

(continued on next page)

READ/WRITE VARIOUS CONTROL AND STATUS BITS IN REGISTER MODE (continued)

Reg. Addr.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	< Bit number
0x0D00	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	NITE MODE	< READ/WRITE
0x0D01	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	DMPR TEST	< READ/WRITE (note 1)
0x0D02	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	SYS. RES.	< READ/WRITE (note 1)
0x0D03	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	EFAN TEST	< READ/WRITE (note 1)
0x0D04	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	ALA. RES.	< READ/WRITE (note 1)
0x0D05	*	*	*	*	*	*	*	*	TCF 8	TCF 7	TCF 6	TCF 5	TCF 4	TCF 3	TCF 2	TCF 1	< R/W TCH COMPACT
0x0D06	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	TCF 1	< READ TCH LONG < WRITE TCH ONE-BY-ONE
0x0D07	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	TCF 2	< READ TCH LONG < WRITE TCH ONE-BY-ONE
0x0D08	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	TCF 3	< READ TCH LONG < WRITE TCH ONE-BY-ONE
0x0D09	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	TCF 4	< READ TCH LONG < WRITE TCH ONE-BY-ONE
0x0D0A	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	TCF 5	< READ TCH LONG < WRITE TCH ONE-BY-ONE
0x0D0B	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	TCF 6	< READ TCH LONG < WRITE TCH ONE-BY-ONE
0x0D0C	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	TCF 7	< READ TCH LONG < WRITE TCH ONE-BY-ONE
0x0D0D	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	TCF 8	< READ TCH LONG < WRITE TCH ONE-BY-ONE

* = unused bits, always write as 0!

Flag names:
- "NITE MODE" = MODBUS nightmode control flag, 1 = Night, 0 = Day (read/write)
- "DMPR TEST" = Start damper test (write only)
- "SYS. RES." = System reset (write only)
- "EFAN TEST" = Start EVAC fan test (write only)
- "ALA. RES." = Alarm reset (write only)
- "TCF #" = Time Channel Flag # (read/write)

Note! KSUA day/night mode is controlled by this "NITE MODE" flag ONLY if the KSUA internal nightmode schedule is disabled! The actual KSUA day/night status can be read via MODBUS with "READ INPUTS AND FLAGS COMPACT" or "READ INPUTS AND FLAGS LONG" (above).

Note! The ”Time Channel Flags” are controlled exclusively by ModBus commands. The flags are cleared (0) at KSUA start-up. The read function is for verification purposes only. A normal ventilation damper will be forced to close if it is associated with a ”Time Channel Flag” that is set (1).

(continued on next page)

READ/WRITE VARIOUS CONTROL AND STATUS BITS IN REGISTER MODE (continued)

Example MODBUS transaction for "Read MODBUS nightmode control flag" (CRC not shown):

Master request:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 4 or 3
- Byte 2 = Starting Address MSB = 13
- Byte 3 = Starting Address LSB = 0
- Byte 4 = Register Count MSB = 0
- Byte 5 = Register Count LSB = 1

Slave response:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 4 or 3
- Byte 2 = Total Register Bytecount = 2
- Byte 3 = Register at 0x0D00, MSB = 0
- Byte 4 = Register at 0x0D00, LSB = 1 if night, else 0

Example MODBUS transaction for "Write MODBUS nightmode control flag" (CRC not shown):

Master request:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 6
- Byte 2 = Starting Address MSB = 13
- Byte 3 = Starting Address LSB = 0
- Byte 4 = Register Data MSB = 0
- Byte 5 = Register Data LSB = 0 for day, 1 for night

Slave response:
- Exactly the same as the master request.

Example MODBUS transaction for "Start damper test" (CRC not shown):

Master request:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 6
- Byte 2 = Starting Address MSB = 13
- Byte 3 = Starting Address LSB = 1
- Byte 4 = Register Data MSB = 0
- Byte 5 = Register Data LSB = 1

Slave response:
- Exactly the same as the master request.

(continued on next page)

READ/WRITE VARIOUS CONTROL AND STATUS BITS IN REGISTER MODE (continued)

Example MODBUS transaction for "System reset" (CRC not shown):

Master request:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 6
- Byte 2 = Starting Address MSB = 13
- Byte 3 = Starting Address LSB = 2
- Byte 4 = Register Data MSB = 0
- Byte 5 = Register Data LSB = 1

Slave response:
Exactly the same as the master request.

Example MODBUS transaction for "Start EVAC fan test" (CRC not shown):

Master request:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 6
- Byte 2 = Starting Address MSB = 13
- Byte 3 = Starting Address LSB = 3
- Byte 4 = Register Data MSB = 0
- Byte 5 = Register Data LSB = 1

Slave response:
Exactly the same as the master request.

Example MODBUS transaction for "Alarm reset" (CRC not shown):

Master request:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 6
- Byte 2 = Starting Address MSB = 13
- Byte 3 = Starting Address LSB = 4
- Byte 4 = Register Data MSB = 0
- Byte 5 = Register Data LSB = 1

Slave response:
Exactly the same as the master request.

Note:
"Alarm Reset" works only when KSUA is in normal ("non-critical") mode!
In all other cases, the "System Reset" command must be used.

Note 1:
The commands "Start damper test", "System reset", "Start EVAC fan test" and "Alarm reset" are trigger event commands that have no corresponding flags to read back. However, read register commands (function code 0x03 or 0x04) have been implemented on addresses 0x0D01..0x0D04. These commands always return a faked register value of 0x0001 = ”TRUE”. This was implemented to satisfy ModBus masters that do verification read operations after every write.

(continued on next page)

READ/WRITE VARIOUS CONTROL AND STATUS BITS IN REGISTER MODE (continued)

Example MODBUS transaction for "Read Time Channel Flag Register (COMPACT)" (CRC not shown):

Master request:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 4 or 3
- Byte 2 = Starting Address MSB = 13
- Byte 3 = Starting Address LSB = 5
- Byte 4 = Register Count MSB = 0
- Byte 5 = Register Count LSB = 1

Slave response:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 4 or 3
- Byte 2 = Total Register Bytecount = 2
- Byte 3 = Register at 0x0D05, MSB = 0
- Byte 4 = Register at 0x0D05, LSB = Time Channel Flags, bit 0 is TCH1

Example MODBUS transaction for "Write Time Channel Flag Register (COMPACT)" (CRC not shown):

Master request:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 6
- Byte 2 = Starting Address MSB = 13
- Byte 3 = Starting Address LSB = 5
- Byte 4 = Register Data MSB = 0
- Byte 5 = Register Data LSB = Time Channel Flags, bit 0 is TCH1

Slave response:
- Exactly the same as the master request.

(continued on next page)

READ/WRITE VARIOUS CONTROL AND STATUS BITS IN REGISTER MODE (continued)

Example MODBUS transaction for "Read Time Channel Flag Register (LONG)" (CRC not shown):

Master request:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 4 or 3
- Byte 2 = Starting Address MSB = 13
- Byte 3 = Starting Address LSB = 6..13
- Byte 4 = Register Count MSB = 0
- Byte 5 = Register Count LSB = 1..(14-Start_Address_LSByte) (max 8)

Slave response:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 4 or 3
- Byte 2 = Total Register Bytecount = 2 * (Register Count) = 2..16
- Byte 3 = Register MSB from starting address
- Byte 4 = Register LSB from starting address
- Byte 5 = Register MSB from starting address+1
- Byte 6 = Register LSB from starting address+1
- ...
- ...
- Byte ? = Register MSB from starting address+(Register Count-1)
- Byte ? = Register LSB from starting address+(Register Count-1)

Example MODBUS transaction for Write Time Channel Flag Register (ONE-BY-ONE) (CRC not shown):

Master request:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 6
- Byte 2 = Starting Address MSB = 13
- Byte 3 = Starting Address LSB = 6..13 for TCH1..TCH8
- Byte 4 = Register Data MSB = 0
- Byte 5 = Register Data LSB = 0 for OFF, 1 for ON

Slave response:
- Exactly the same as the master request.

==========
DIAGNOSTIC FUNCTION
==========

Diagnostic sub-function 0 (Return Query Data = ECHO message)
is available for "pinging" the KSUA unit.

Example MODBUS transaction for "Return Query Data" (CRC not shown):

Master request:
Byte 0 = Slave Address = 1..247
Byte 1 = Function Code = 8
Byte 2 = Sub-function code MSB = 0
Byte 3 = Sub-function code LSB = 0
Byte 4..253 = any number (0..250) of bytes

Slave response:
Exactly the same as the master request.

==========
EXCEPTION CODES
==========

The following exception responses are implemented:

Exception code 1 - Illegal Function Code
Exception code 2 - Illegal Data Address
Exception code 3 - Illegal Data Value

[End of document]