KSUA MODBUS RTU INTERFACE

The following exception responses are implemented:

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

READ DETECTOR STATUS

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

16-bit register read:
- Function_code: 0x04 ("Read Input Registers")
- Start_address: Register address, 0x0000..0x000F
- Register_count: 1..(0x0010-Start_address)

Register address:	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	<- Detector number
	*	FAIL	SERV	FIRE	*	FAIL	SERV	FIRE	*	FAIL	SERV	FIRE	*	FAIL	SERV	FIRE	<- Detector 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 "Detector status read" (CRC not shown):

Master request:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 4
- 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-Starting Address LSB)

Slave response:
Byte 0 = Slave Address = 1..247
Byte 1 = Function Code = 4
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

8 registers (16-bit)
8 dampers per register

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

Register address:	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 number
	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	<- Position 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 "Damper position read" (CRC not shown):

```
  Master request:
    Byte 0 = Slave Address = 1..247
    Byte 1 = Function Code = 4
    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-Starting Address LSB)

  Slave response:
    Byte 0 = Slave Address = 1..247
    Byte 1 = Function Code = 4
    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
==============================

  5 registers (16-bit)
  16 detector groups per register
  Groups 1..64 consists of detectors on KSUB slaves
  Groups 65e..80e are "External Detector Groups" (XDG)

  16-bit register read:
    Function_code:  0x04 ("Read Input Registers")
    Start_address:  Register address, 0x0200..0x0204
    Register_count: 1..(5-(Start_address-0x0200))

  |----------|
  | Register |----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
  | address: | 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 number
  |          |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|
  |----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
```

0x0204	80e FIRE	79e FIRE	78e FIRE	77e FIRE	76e FIRE	75e FIRE	74e FIRE	73e FIRE	72e FIRE	71e FIRE	70e FIRE	69e FIRE	68e FIRE	67e FIRE	66e FIRE	65e FIRE

Example MODBUS transaction for "Detector group fire flag read" (CRC not shown):

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

Slave response:
- Byte 0 = Slave Address = 1..247
- Byte 1 = Function Code = 4
- Byte 2 = Total Register Bytecount = 2 * (Register Count) = 2..10
- 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

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

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

Register address:	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 "Inputs & flags read" (CRC not shown):

Master request:
- Byte 0 = Slave Address = 1..247

```
    Byte 1 = Function Code = 4
    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
    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")
    Start_address:  Register address, 0x0400
    Register_count: 4 (must be 4!)

  |----------|
  | Register |----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
  | address: | 15 | 14 | 13 | 12 | 11 | 10 |  9 |  8 |  7 |  6 |  5 |  4 |  3 |  2 |  1 |  0 | <- Bit number
  |----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
  |  0x0400  |  * |  * |MIN |MIN |MIN |MIN |MIN |MIN |  * |  * |SEC |SEC |SEC |SEC |SEC |SEC |
  |          |    |    |  5 |  4 |  3 |  2 |  1 |  0 |    |    |  5 |  4 |  3 |  2 |  1 |  0 |
  |----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
  |  0x0401  |  * |  * |  * |  * |  * |WDAY|WDAY|WDAY|  * |  * |  * |HOUR|HOUR|HOUR|HOUR|HOUR|
  |          |    |    |    |    |    |  2 |  1 |  0 |    |    |    |  4 |  3 |  2 |  1 |  0 |
  |----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
  |  0x0402  |  * |  * |  * |  * |MONT|MONT|MONT|MONT|  * |  * |  * |DATE|DATE|DATE|DATE|DATE|
  |          |    |    |    |    |  3 |  2 |  1 |  0 |    |    |    |  4 |  3 |  2 |  1 |  0 |
```

0x0403	*	*	*	*	*	*	*	*	*	YEAR 6	YEAR 5	YEAR 4	YEAR 3	YEAR 2	YEAR 1	YEAR 0

Example MODBUS transaction for "RTC read" (CRC not shown):

Master request:
  Byte 0 = Slave Address = 1..247
  Byte 1 = Function Code = 4
  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
  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 "RTC write" (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/WRITE VARIOUS CONTROL AND STATUS BITS
========================================

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

  Single bit read (MODBUS Nightmode Control only):
    Function_code:  0x01 ("Read Coils")
    Start_address:  0x0000
    Bit_count:      1

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

  Single bit write (Alarm Reset only):
    Function_code:  0x05 ("Write Single Coil")
    Output_address: 0x0002
    Output_value:   0xFF00


    |-------------------------------------------------------|
    | Bit address | Bit function              | Read  | Write |
    |=============|===========================|===============|
    |   0x0000    | MODBUS Nightmode control  |  YES  |  YES  |
    |-------------|---------------------------|---------------|
    |   0x0001    | Start Damper Test         |  NO   |  YES  |
    |-------------|---------------------------|---------------|
    |   0x0002    | Alarm Reset               |  NO   |  YES  |
    |-------------|---------------------------|---------------|

  Example MODBUS transaction for "MODBUS nightmode control write" (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 write" (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 "Alarm reset write" (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 alarm!)
    Byte 5 = Output Value LSB = 0

  Slave response:
    Exactly the same as the master request.


  Example MODBUS transaction for "MODBUS nightmode control read" (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
    Byte 4 = Number of bits MSB = 0
    Byte 5 = Number of bits LSB = 1

  Slave response:
    Byte 0 = Slave Address = 1..247
    Byte 1 = Function Code = 1
    Byte 2 = Bytecount = 1
    Byte 3 = MODBUS Nightmode Flag in bit 0 (remaining bits are zero)


==================
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.

<eof>