Kilsen

Modbus Protocol Guide for KFP-A Series Control Panels

P/N 00-3243-505-1603-02 • ISS 08JUL21

Trademarks and patents

The KFP-A name and logo are trademarks of Carrier.

Other trade names used in this document may be trademarks or registered trademarks of the manufacturers or vendors of the respective products.

Manufacturer

Carrier Manufacturing Poland Spòlka Z o.o.,
Ul. Kolejowa 24, 39-100 Ropczyce, Poland.

Authorized EU manufacturing representative:
Carrier Fire & Security B.V., Kelvinstraat 7, 6003 DH Weert, Netherlands.

Version REV 02

Certification CE

Contact information and product documentation

For contact information or to download the latest product documentation, visit firesecurityproducts.com.

Content

Important information ii
About this guide ii
Limitation of liability ii
Advisory messages iii

About Modbus 1
Introduction 1
Modbus TCP/IP 2
Modbus PDU 4
Registers 5
Function codes 6
Exception responses 9
Implementation 10

Read holding registers 11
Zone/Point mode 11
Zone mode 21

Write holding registers 31
Sending commands 31
Execute Reset 32
Execute Panel Silence 32
Execute Fire Protection Override Delay 32
Execute Fire Routing Override Delay 32
Set Sounders Start/Stop 32
Set Sounder Delay On/Off 33
Set Fire Protection Delay On/Off 33
Set Fire Routing Delay On/Off 33
Heartbeat 33

Other settings 34
Maximum Polling Frequency 34
Maximum Quantity of Registers 34

Examples 35
Zone/Point mode 35
Zone mode 38

Important information

About this guide

The purpose of this guide is to describe the Modbus function codes used for KFP-A Series control panel applications.

The specification referenced to create the control panel Modbus implementation is Modbus Application Protocol Specification V1.1b.

Caution: Read this guide, all related control panel documentation, and all related Modbus protocol standards and specifications entirely before creating Modbus applications.

Limitation of liability

To the maximum extent permitted by applicable law, in no event will Carrier be liable for any lost profits or business opportunities, loss of use, business interruption, loss of data, or any other indirect, special, incidental, or consequential damages under any theory of liability, whether based in contract, tort, negligence, product liability, or otherwise. Because some jurisdictions do not allow the exclusion or limitation of liability for consequential or incidental damages the preceding limitation may not apply to you. In any event the total liability of Carrier shall not exceed the purchase price of the product. The foregoing limitation will apply to the maximum extent permitted by applicable law, regardless of whether Carrier has been advised of the possibility of such damages and regardless of whether any remedy fails of its essential purpose.

Installation in accordance with this manual, applicable codes, and the instructions of the authority having jurisdiction is mandatory.

While every precaution has been taken during the preparation of this manual to ensure the accuracy of its contents, Carrier assumes no responsibility for errors or omissions.

Product warnings and disclaimers

THESE PRODUCTS ARE INTENDED FOR SALE TO AND INSTALLATION BY QUALIFIED PROFESSIONALS. CARRIER FIRE & SECURITY B.V. CANNOT PROVIDE ANY ASSURANCE THAT ANY PERSON OR ENTITY BUYING ITS PRODUCTS, INCLUDING ANY “AUTHORIZED DEALER” OR “AUTHORIZED RESELLER”, IS PROPERLY TRAINED OR EXPERIENCED TO CORRECTLY INSTALL FIRE AND SECURITY RELATED PRODUCTS.

For more information on warranty disclaimers and product safety information, please check https://firesecurityproducts.com/policy/product-warning/ or scan the QR code:

Advisory messages

Advisory messages alert you to conditions or practices that can cause unwanted results. The advisory messages used in this document are shown and described below.

WARNING: Warning messages advise you of hazards that could result in injury or loss of life. They tell you which actions to take or to avoid in order to prevent the injury or loss of life.

Caution: Caution messages advise you of possible equipment damage. They tell you which actions to take or to avoid in order to prevent the damage.

Note: Note messages advise you of the possible loss of time or effort. They describe how to avoid the loss. Notes are also used to point out important information that you should read.

About Modbus

Introduction

Modbus is an application layer messaging protocol, positioned at level 7 of the OSI model that provides client/server communication between devices connected on different types of buses or networks.

The industry's serial de facto standard since 1979, Modbus continues to enable millions of automation devices to communicate. Today, support for the simple and elegant structure of Modbus continues to grow. The Internet community can access Modbus at a reserved system port 502 on the TCP/IP stack.

Modbus is a request/reply protocol and offers services specified by function codes.

The Modbus protocol was incorporated, for industrial automation systems and Modicon programmable controllers. It has since become an industry standard method for the transfer of discrete/analog I/O information and register data between industrial control and monitoring devices. Modbus is now a widely accepted, open, public domain protocol that requires a license, but does not require royalty payment to its owner.

Modbus devices communicate using a master/slave (client/server) technique in which only one device (the master/client) can initiate transactions (queries). The other devices (slaves/servers) respond by supplying the requested data to the master, or by taking the action requested in the query. A slave is any peripheral device (I/O transducer, valve, network drive, or other measuring device) which processes information and sends its output to the master using Modbus. A typical master device is a host computer running appropriate application software.

Modbus TCP/IP

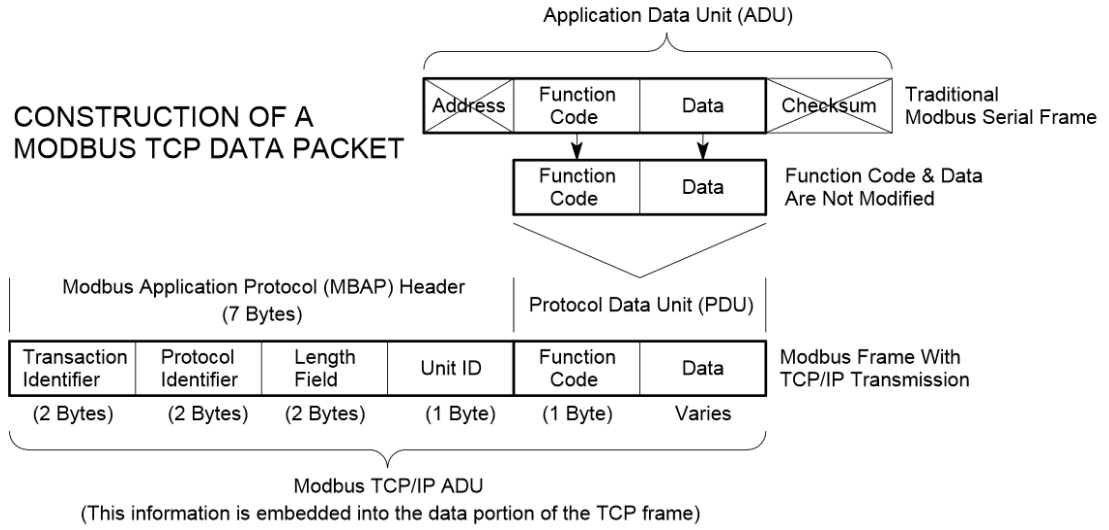
Modbus TCP/IP (or Modbus TCP) is the Modbus RTU protocol with a TCP interface that runs on Ethernet.

TCP/IP refers to the Transmission Control Protocol and Internet Protocol, which provides the transmission medium for Modbus TCP/IP messaging.

The primary function of TCP is to ensure that all packets of data are received correctly, while IP makes sure that messages are correctly addressed and routed. The Modbus TCP/IP uses TCP/IP and Ethernet to carry the data of the Modbus message structure between compatible devices. That is, Modbus TCP/IP combines a physical network (Ethernet), with a networking standard (TCP/IP), and a standard method of representing data (Modbus as the application protocol). Essentially, the Modbus TCP/IP message is simply a Modbus communication encapsulated in an Ethernet TCP/IP wrapper.

In practice, Modbus TCP embeds a standard Modbus data frame into a TCP frame, without the Modbus checksum, as shown below.

Figure 1: General Modbus TCP/IP ADU frame.



The Modbus commands and user data are themselves encapsulated into the data container of a TCP/IP telegram without being modified in any way. However, the Modbus error checking field (checksum) is not used, as the standard Ethernet TCP/IP link layer checksum methods are instead used to guaranty data integrity. Further, the Modbus frame address field is supplanted by the unit identifier in Modbus TCP/IP, and becomes part of the Modbus Application Protocol (MBAP) header.

The Protocol Data Unit (PDU), the function code, and data fields are absorbed in their original form. Thus, a Modbus TCP/IP Application Data Unit (ADU) takes the form of a 7-byte header (transaction identifier + protocol identifier + length field + unit identifier), and the Protocol Data Unit (function code + data).

The MBAP header is 7-bytes long and includes the following fields:

- **Transaction/invocation Identifier (2 bytes):** this identification field is used for transaction pairing when multiple messages are sent along the same TCP connection by a client without waiting for a prior response.
- **Protocol Identifier (2 bytes):** This field is always 0 for Modbus services and other values are reserved for future extensions.
- **Length (2 bytes):** this field is a byte count of the remaining fields and includes the unit identifier byte, function code byte, and the data fields.
- **Unit Identifier (1 byte):** this field is used to identify a remote server located on a non TCP/IP network (for serial bridging). In a typical Modbus TCP/IP server application, the unit ID is set to 00 or FF, ignored by the server, and simply echoed back in the response.

Figure 2: Modbus request ADU example header

MBAP Header Fields	Example Decimal (Hexadecimal)
Transaction ID High Order	0 (00) *Client sets, unique value.*
Transaction ID Low Order	1 (01) *Client sets, unique value.*
Protocol Identifier High Order	0 (00) *Specifies Modbus service.*
Protocol Identifier Low Order	0 (00) *Specifies Modbus service.*
Length High Order	0 (00) *Client calculates.*
Length Low Order	6 (06) *Client calculates.*
Unit Identifier	255 (FF) or 0 (00) *Do not bridge.*

Figure 3: Modbus response ADU example header

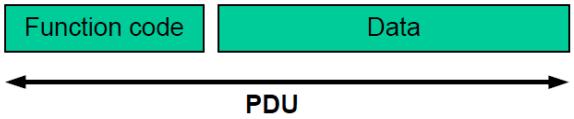
MBAP Header Fields	Example Decimal (Hexadecimal)
Transaction ID High Order	0 (00) *Echoed back, no change*
Transaction ID Low Order	1 (01) *Echoed back, no change*
Protocol Identifier High Order	0 (00) *Echoed back, no change*
Protocol Identifier Low Order	0 (00) *Echoed back, no change*
Length High Order	0 (00) *Server calculates*
Length Low Order	4 (04) *Server calculates.*
Unit Identifier	255 (FF) or 0 (00) *No change*

The complete Modbus TCP/IP Application Data Unit is embedded into the data field of a standard TCP frame and sent via TCP to port 502, which is specifically reserved for Modbus applications. Modbus TCP/IP clients and servers listen and receive Modbus data via port 502.

Modbus PDU

The Modbus protocol defines a simple Protocol Data Unit (PDU) independent of the underlying communication layers. The service request (Modbus Protocol Data Unit) is comprised of a function code and a number of additional data bytes, depending on the function.

Figure 4: General Modbus PDU frame

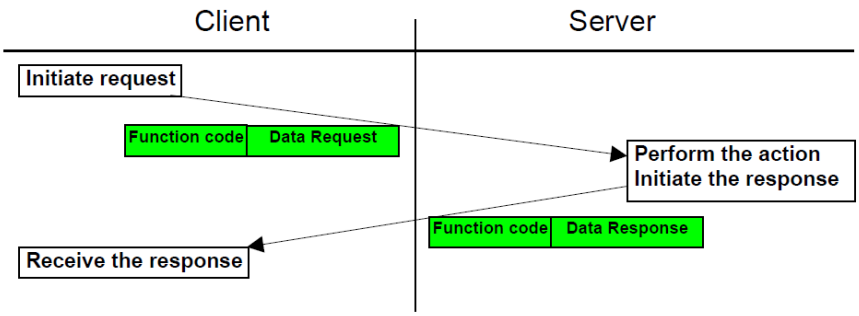


The Modbus application data unit is built by the client that initiates a Modbus transaction. The function indicates to the server what kind of action to perform. The Modbus application protocol establishes the format of a request initiated by a client.

The function code field of a Modbus data unit is coded in one byte. Valid codes are in the range of 1-255 decimal (with 128-255 reserved for exception responses). When a message is sent from a client to a server device the function code field tells the server what kind of action to perform. Function code "0" is not valid.

The data field of messages sent from a client to server devices contains additional information that the server uses to take the action defined by the function code. This can include items like discrete and register addresses, the quantity of items to be handled, and the count of actual data bytes in the field.

Figure 5: Modbus transaction



The size of the Modbus PDU is limited by the size constraint inherited from the first Modbus implementation on Serial Line network (max. RS-485 ADU = 256 bytes).

The control panel supports the functions described in this document (see “Function codes” on page 6). Use of any other functions will generate an ILLEGAL FUNCTION exception.

Registers

Modbus uses 16-bit registers. These registers represent a contiguous address mapping of control panel, zone, and device status.

Zone/Point mode

Start address	End address	Access	Use
0x0001	0x0001	Write (W)	Execute Reset
0x0002	0x0002	Write (W)	Execute Panel Silence
0x0003	0x0003	Write (W)	Set Sounders Start/Stop
0x0004	0x0004	Write (W)	Set Sounders Delay On/Off
0x0005	0x0005	Write (W)	Set Fire Protection Delay On/Off
0x0006	0x0006	Write (W)	Execute Fire Protection Override Delay
0x0007	0x0007	Write (W)	Set Fire Routing Delay On/Off
0x0008	0x0008	Write (W)	Execute Fire Routing Override Delay
0x1001	0x1002	Read (R)	Control Panel Global Status Flags (32 Nodes).
0x2001	0x2080	Read (R)	Control Panel Status Flags (32 Nodes)
0x3001	0x7000	Read (R)	Control Panel Zone Status (32 Nodes, 512 Zones)
0x7001	0xF000	Read (R)	Control Panel Device Status (32 Nodes, 4 Loops, 256 Devices)

Zone mode

Start address	End address	Access	Use
0x0001	0x0001	Write (W)	Execute Reset
0x0002	0x0002	Write (W)	Execute Panel Silence
0x0003	0x0003	Write (W)	Set Sounders Start/Stop
0x0004	0x0004	Write (W)	Set Sounders Delay On/Off
0x0005	0x0005	Write (W)	Set Fire Protection Delay On/Off
0x0006	0x0006	Write (W)	Execute Fire Protection Override Delay
0x0007	0x0007	Write (W)	Set Fire Routing Delay On/Off
0x0008	0x0008	Write (W)	Execute Fire Routing Override Delay
0x1001	0x1002	Read (R)	Control Panel Global Status Flags (128 Nodes)
0x2001	0x2200	Read (R)	Control Panel Status Flags (128 Nodes)
0x3001	0xB000	Read (R)	Control Panel Zone Status (128 Nodes, 512 Zones)

Heartbeat

Start address	End address	Access	Use
0xFFFF	0xFFFF	Write (W)	Resets the inactivity time counter to prevent the TCP/IP connection from closing automatically

Function codes

The following Modbus function codes are supported:

- 03 Read holding registers
- 06 Write single register

03 read holding registers

This function code is used to read the contents of a contiguous block of holding registers in a remote device.

The Request PDU specifies the starting register address and the number of registers. In the PDU registers are addressed starting at zero (registers 1-16 are addressed 0-15).

The register data in the response message are packed as two bytes per register, with the binary contents right justified within each byte. For each register, the first byte contains the high order bits and the second contains the low order bits.

Figure 6: Read holding registers definition

Request

Function code	1 Byte	**0x03**
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Registers	2 Bytes	1 to 125 (0x7D)

Response

Function code	1 Byte	**0x03**
Byte count	1 Byte	2 x **N***
Register value	**N*** x 2 Bytes	

***N** = Quantity of Registers

Error

Error code	1 Byte	**0x83**
Exception code	1 Byte	01 or 02 or 03 or 04

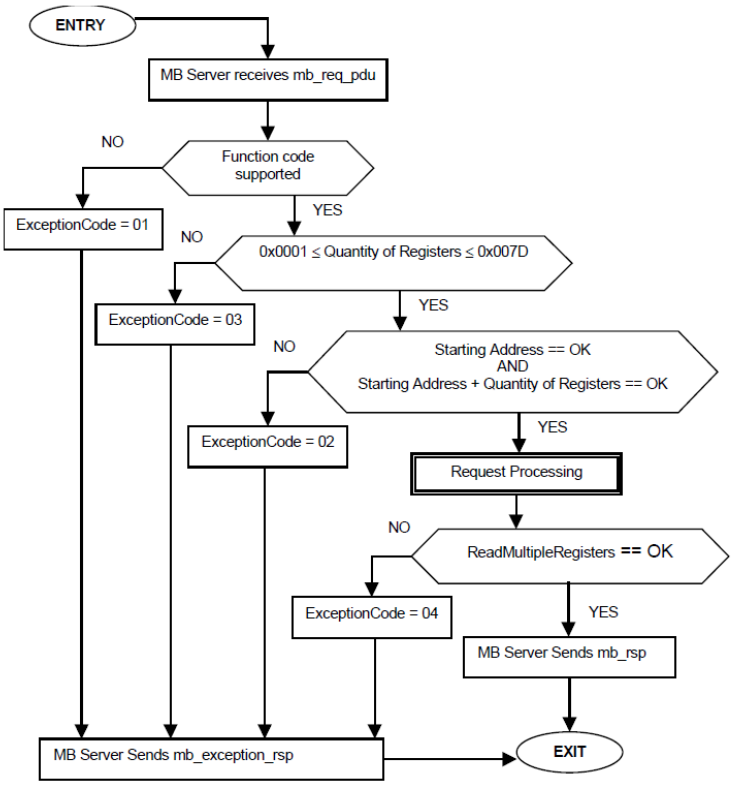
Below is an example of a request to read registers 108-110.

Figure 7: Read holding registers request/response example

Request		**Response**	
Field Name	*(Hex)*	*Field Name*	*(Hex)*
Function	**03**	Function	**03**
Starting Address Hi	**00**	Byte Count	**06**
Starting Address Lo	**6B**	Register value Hi (108)	**02**
No. of Registers Hi	**00**	Register value Lo (108)	**2B**
No. of Registers Lo	**03**	Register value Hi (109)	**00**
		Register value Lo (109)	**00**
		Register value Hi (110)	**00**
		Register value Lo (110)	**64**

The contents of register 108 are shown as the two-byte values of 02 2B hex or 555 decimal. The contents of registers 109-110 are 00 00 and 00 64 hex, or 0 and 100 decimal, respectively.

Figure 8: Read holding registers state diagram



06 write single register

This function code is used to write a single holding register to a remote device.

The Request PDU specifies the address of the register to be written. Registers are addressed starting at zero (register 1 is addressed 0).

The normal response is an echo of the request, returned after the register content is written.

Figure 9: Write single register definition

Request

Function code	1 Byte	**0x06**
Register Address	2 Bytes	0x0000 to 0xFFFF
Register Value	2 Bytes	0x0000 or 0xFFFF

Response

Function code	1 Byte	**0x06**
Register Address	2 Bytes	0x0000 to 0xFFFF
Register Value	2 Bytes	0x0000 or 0xFFFF

Error

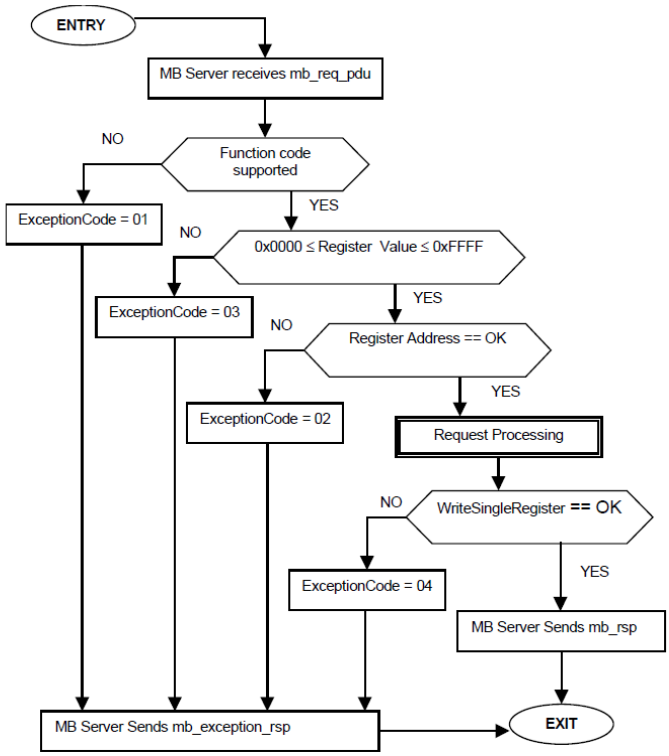
Error code	1 Byte	**0x86**
Exception code	1 Byte	01 or 02 or 03 or 04

Below is an example of a request to write register 2 to 00 03 hex.

Figure 10: Write single register request/response example

Request		Response	
Field Name	*(Hex)*	*Field Name*	*(Hex)*
Function	**06**	Function	**06**
Register Address Hi	**00**	Register Address Hi	**00**
Register Address Lo	**01**	Register Address Lo	**01**
Register Value Hi	**00**	Register Value Hi	**00**
Register Value Lo	**03**	Register Value Lo	**03**

Figure 11: Write single register state diagram



Exception responses

The exception response message has two fields that differentiate it from a normal response. They are:

- **Function Code Field:** in a normal response, the server echoes the function code of the original request in the function code field of the response. All function codes have a most–significant bit (MSB) of 0 (their values are all below 80 hexadecimal). In an exception response, the server sets the MSB of the function code to 1. This makes the function code value in an exception response exactly 80 hexadecimal higher than the value would be for a normal response. With the function code's MSB set, the client's application program can recognize the exception response and can examine the data field for the exception code.
- **Data Field:** in a normal response, the server may return data or statistics in the data field (any information that was requested in the request). In an exception response, the server returns an exception code in the data field. This defines the server condition that caused the exception.

Modbus exception codes

Code	Name	Description
01	ILLEGAL FUNCTION	The function code received in the query is not an allowable action for the server (or slave). This may be because the function code is only applicable to newer devices, and was not implemented in the unit selected. It could also indicate that the server (or slave) is in the wrong state to process a request of this type, for example because it is not configured and is being asked to return register values.
02	ILLEGAL DATA ADDRESS	The data address received in the query is not an allowable address for the server (or slave). More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers, a request with offset 96 and length 4 would succeed, a request with offset 96 and length 5 will generate exception 02.
03	ILLEGAL DATA VALUE	A value contained in the query data field is not an allowable value for server (or slave). This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It specifically does NOT mean that a data item submitted for storage in a register has a value outside the expectation of the application program, since the Modbus protocol is unaware of the significance of any particular value of any particular register.
04	SLAVE DEVICE FAILURE	An unrecoverable error occurred while the server (or slave) was attempting to perform the requested action.
05	ACKNOWLEDGE	Specialized use in conjunction with programming commands. The server (or slave) has accepted the request and is processing it, but a long duration of time will be required to do so. This response is returned to prevent a timeout error from occurring in the client (or master). The client (or master) can next issue a Poll Program Complete message to determine if processing is completed.
06	SLAVE DEVICE BUSY	Specialized use in conjunction with programming commands. The server (or slave) is engaged in processing a long–duration program command. The client (or master) should retransmit the message later when the server (or slave) is free.

Code	Name	Description
08	MEMORY PARITY ERROR	Specialized use in conjunction with function codes 20 and 21 and reference type 6, to indicate that the extended file area failed to pass a consistency check. The server (or slave) attempted to read record file, but detected a parity error in the memory. The client (or master) can retry the request, but service may be required on the server (or slave) device.
0A	GATEWAY PATH UNAVAILABLE	Specialized use in conjunction with gateways, indicates that the gateway was unable to allocate an internal communication path from the input port to the output port for processing the request. Modbus Application Protocol Specification V1.1a Modbus-IDA Usually means that the gateway is incorrectly configured or overloaded.
0B	GATEWAY TARGET DEVICE FAILED TO RESPOND	Specialized use in conjunction with gateways, indicates that no response was obtained from the target device. Usually means that the device is not present on the network.

Implementation

The Modbus implementation for control panels allows:

- Monitoring the status for all devices in the control panel network
- Executing selected control panel operations

The following Modbus function codes are supported:

- 03 Read holding registers
- 06 Write single register

Note: Multiple register writes is not implemented as normal operation consists of monitoring (with only occasional operation commands).

Separate registers are used to hold the status of control panels, zones, and devices in the network (see “Registers” on page 5).

Read holding registers

Zone/Point mode

Control Panel Global Status

This status consists of two read-only holding registers. These registers represent the global status for all control panels in the network. Each register is divided into two bytes.

Start address	End address	Access	Use
0x1001	0x1001	R	GLOBAL_NODE_STATUS1 (1 to 32 Nodes)
0x1002	0x1002	R	GLOBAL_NODE_STATUS2 (1 to 32 Nodes)

Status 1

High byte								Low byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Sounder status							Reserved		Functional conditions					

Bit	High byte	Bit	Low byte
8	Sounder delay enabled	0	Alarm functional condition
9	Sounder activation delay in progress	1	Fault functional condition
10	Sounder outputs activated	2	Disable functional condition
11	Sounder outputs silenced	3	Test functional condition
12	Sounders disabled	4	Day/Night mode (1=nightmode/0=daymode)
13	Sounder override time elapsed	5	MCP alarm
14	Sounders in Test	6	Reserved
15	Reserved	7	Reserved

- **Alarm:** If bit 0 is 1, then one or more control panels are in alarm status.
- **Fault:** If bit 1 is 1, then one or more control panels are in fault status.
- **Disabled:** If bit 2 is 1, then one or more control panels are disabled.
- **Test:** If bit 3 is 1, then one or more control panels are in test status.

Status 2

High byte								Low byte							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		Fire protection status						Res.	Fire routing status						

Bit	High byte	Bit	Low byte
8	Fire Protection: Delay enabled	0	Fire Routing: Delay enabled
9	Fire Protection: Activation delay in progress	1	Fire Routing: Activation delay in progress
10	Fire Protection: Outputs activated	2	Fire Routing: Outputs activated
11	Fire Protection: Acknowledged	3	Fire Routing: Acknowledged
12	Fire Protection: Disabled	4	Fire Routing: Disabled
13	Fire Protection: Test ON	5	Fire Routing: Test ON
14	Reserved	6	Fire Routing: Ext. activation delay in progress
15	Reserved	7	Reserved

Control Panel Status

This status consists of 128 read-only holding registers (32 nodes, 4 registers per node). These registers represent all status options for a single node (control panel). Each register is divided into two bytes.

Start address	End address	Access	Use
0x2001	0x2080	R	NODE1_STATUS1 - NODE_32_STATUS4

Addr.	Value	Addr.	Value	Addr.	Value	Addr.	Value	Addr.	Value
0x2001	NODE1_ST1	0x201D	NODE8_ST1	0x2039	NODE15_ST1	0x2055	NODE22_ST1	0x2071	NODE29_ST1
0x2002	NODE1_ST2	0x201E	NODE8_ST2	0x203A	NODE15_ST2	0x2056	NODE22_ST2	0x2072	NODE29_ST2
0x2003	NODE1_ST3	0x201F	NODE8_ST3	0x203B	NODE15_ST3	0x2057	NODE22_ST3	0x2073	NODE29_ST3
0x2004	NODE1_ST4	0x2020	NODE8_ST4	0x203C	NODE15_ST4	0x2058	NODE22_ST4	0x2074	NODE29_ST4
0x2005	NODE2_ST1	0x2021	NODE9_ST1	0x203D	NODE16_ST1	0x2059	NODE23_ST1	0x2075	NODE30_ST1
0x2006	NODE2_ST2	0x2022	NODE9_ST2	0x203E	NODE16_ST2	0x205A	NODE23_ST2	0x2076	NODE30_ST2
0x2007	NODE2_ST3	0x2023	NODE9_ST3	0x203F	NODE16_ST3	0x205B	NODE23_ST3	0x2077	NODE30_ST3
0x2008	NODE2_ST4	0x2024	NODE9_ST4	0x2040	NODE16_ST4	0x205C	NODE23_ST4	0x2078	NODE30_ST4
0x2009	NODE3_ST1	0x2025	NODE10_ST1	0x2041	NODE17_ST1	0x205D	NODE24_ST1	0x2079	NODE31_ST1
0x200A	NODE3_ST2	0x2026	NODE10_ST2	0x2042	NODE17_ST2	0x205E	NODE24_ST2	0x207A	NODE31_ST2
0x200B	NODE3_ST3	0x2027	NODE10_ST3	0x2043	NODE17_ST3	0x205F	NODE24_ST3	0x207B	NODE31_ST3
0x200C	NODE3_ST4	0x2028	NODE10_ST4	0x2044	NODE17_ST4	0x2060	NODE24_ST4	0x207C	NODE31_ST4
0x200D	NODE4_ST1	0x2029	NODE11_ST1	0x2045	NODE18_ST1	0x2061	NODE25_ST1	0x207D	NODE32_ST1
0x200E	NODE4_ST2	0x202A	NODE11_ST2	0x2046	NODE18_ST2	0x2062	NODE25_ST2	0x207E	NODE32_ST2
0x200F	NODE4_ST3	0x202B	NODE11_ST3	0x2047	NODE18_ST3	0x2063	NODE25_ST3	0x207F	NODE32_ST3
0x2010	NODE4_ST4	0x202C	NODE11_ST4	0x2048	NODE18_ST4	0x2064	NODE25_ST4	0x2080	NODE32_ST4
0x2011	NODE5_ST1	0x202D	NODE12_ST1	0x2049	NODE19_ST1	0x2065	NODE26_ST1		
0x2012	NODE5 _ST2	0x202E	NODE12_ST2	0x204A	NODE19_ST2	0x2066	NODE26_ST2		
0x2013	NODE5 _ST3	0x202F	NODE12_ST3	0x204B	NODE19_ST3	0x2067	NODE26_ST3		
0x2014	NODE5 _ST4	0x2030	NODE12_ST4	0x204C	NODE19_ST4	0x2068	NODE26_ST4		
0x2015	NODE6_ST1	0x2031	NODE13_ST1	0x204D	NODE20_ST1	0x2069	NODE27_ST1		
0x2016	NODE6 _ST2	0x2032	NODE13_ST2	0x204E	NODE20_ST2	0x206A	NODE27_ST2		
0x2017	NODE6 _ST3	0x2033	NODE13_ST3	0x204F	NODE20_ST3	0x206B	NODE27_ST3		
0x2018	NODE6 _ST4	0x2034	NODE13_ST4	0x2050	NODE20_ST4	0x206C	NODE27_ST4		
0x2019	NODE7 _ST1	0x2035	NODE14_ST1	0x2051	NODE21_ST1	0x206D	NODE28_ST1		
0x201A	NODE7 _ST2	0x2036	NODE14_ST2	0x2052	NODE21_ST2	0x206E	NODE28_ST2		
0x201B	NODE7 _ST3	0x2037	NODE14_ST3	0x2053	NODE21_ST3	0x206F	NODE28_ST3		
0x201C	NODE7 _ST4	0x2038	NODE14_ST3	0x2054	NODE21_ST4	0x2070	NODE28_ST4		

Status 1

High byte								Low byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Sounder status							Reserved		Functional conditions					

Bit	High byte	Bit	Low byte
8	Sounder delay enabled	0	Alarm functional condition
9	Sounder activation delay in progress	1	Fault functional condition
10	Sounder outputs activated	2	Disable functional condition
11	Sounder outputs silenced	3	Test functional condition
12	Sounders disabled	4	Day/Night mode (1=nightmode/0=daymode)
13	Sounder override time elapsed	5	MCP alarm
14	Sounders in Test	6	Reserved
15	Reserved	7	Reserved

Status 2

High byte								Low byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		Fire protection status						Res.	Fire routing status						

Bit	High byte	Bit	Low byte
8	Fire Protection: Delay enabled	0	Fire Routing: Delay enabled
9	Fire Protection: Activation delay in progress	1	Fire Routing: Activation delay in progress
10	Fire Protection: Outputs activated	2	Fire Routing: Outputs activated
11	Fire Protection: Acknowledged	3	Fire Routing: Acknowledged
12	Fire Protection: Disabled	4	Fire Routing: Disabled
13	Fire Protection: Test ON	5	Fire Routing: Test ON
14	Reserved	6	Fire Routing: Ext. activation delay in progress
15	Reserved	7	Reserved

Status 3

High byte								Low byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								Reserved							

Status 4

High byte								Low byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								Reserved							

Control Panel Zone Status

This status consists of 16384 read-only holding registers (32 nodes, 512 zones per node). Each register represents one zone status per node (control panel). Each register is divided into two bytes.

Start address	End address	Access	Node	Zone
0x3001	0x3200	R	NODE1	1-512
0x3201	0x3400	R	NODE2	1-512
0x3401	0x3600	R	NODE3	1-512
0x3601	0x3800	R	NODE4	1-512
0x3801	0x3A00	R	NODE5	1-512
0x3A01	0x3C00	R	NODE6	1-512
0x3C01	0x3E00	R	NODE7	1-512
0x3E01	0x4001	R	NODE8	1-512
0x4001	0x4200	R	NODE9	1-512
0x4201	0x4400	R	NODE10	1-512
0x4401	0x4600	R	NODE11	1-512
0x4601	0x4800	R	NODE12	1-512
0x4801	0x4A00	R	NODE13	1-512
0x4A01	0x4C00	R	NODE14	1-512
0x4C01	0x4E00	R	NODE15	1-512
0x4E01	0x5000	R	NODE16	1-512
0x5001	0x5200	R	NODE17	1-512
0x5201	0x5400	R	NODE18	1-512
0x5401	0x5600	R	NODE19	1-512
0x5601	0x5800	R	NODE20	1-512
0x5801	0x5A00	R	NODE21	1-512
0x5A01	0x5C00	R	NODE22	1-512
0x5C01	0x5E00	R	NODE23	1-512
0x5E01	0x6000	R	NODE24	1-512
0x6001	0x6200	R	NODE25	1-512
0x6201	0x6400	R	NODE26	1-512
0x6401	0x6600	R	NODE27	1-512
0x6601	0x6800	R	NODE28	1-512

Start address	End address	Access	Node	Zone
0x6801	0x6A00	R	NODE29	1-512
0x6A01	0x6C00	R	NODE30	1-512
0x6C01	0x6E00	R	NODE31	1-512
0x6E01	0x7000	R	NODE32	1-512

High byte								Low byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								Reserved			Dis	Test	Fault	Alarm	Alert

- **Alert:** If bit 0 is 1, then one or more zone devices are in prealarm status.
- **Alarm:** If bit 1 is 1, then one or more zone devices are in alarm status.
- **Fault:** If bit 2 is 1, then one or more zone devices are in fault status.
- **Test:** If bit 3 is 1, then the zone is in test.
- **Dis:** If bit 4 is 1, then the zone is disabled.

Control Panel Device Status

This status consists of 32768 read-only holding registers (32 nodes, 4 loops per node, 256 devices per loop). Each register represents the status for a single device. Each register is divided into two bytes.

Start address	End address	Access	Node	Use
0x7001	0x7100	R	NODE1	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x7101	0x7200	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0x7201	0x7300	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x7301	0x7400	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0x7401	0x7500	R	NODE2	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x7501	0x7600	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0x7601	0x7700	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x7701	0x7800	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0x7801	0x7900	R	NODE3	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x7901	0x7A00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0x7A01	0x7B00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x7B01	0x7C00	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0x7C01	0x7D00	R	NODE4	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x7D01	0x7E00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0x7E01	0x7F00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x7F01	0x8000	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0x8001	0x8100	R	NODE5	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x8101	0x8200	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS

Start address	End address	Access	Node	Use
0x8201	0x8300	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x8301	0x8400	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0x8401	0x8500	R	NODE6	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x8501	0x8600	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0x8601	0x8700	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x8701	0x8800	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0x8801	0x8900	R	NODE7	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x8901	0x8A00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0x8A01	0x8B00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x8B01	0x8C00	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0x8C01	0x8D00	R	NODE8	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x8D01	0x8E00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0x8E01	0x8F00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x8F01	0x9000	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0x9001	0x9100	R	NODE9	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x9101	0x9200	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0x9201	0x9300	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x9301	0x9400	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0x9401	0x9500	R	NODE10	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x9501	0x9600	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0x9601	0x9700	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x9701	0x9800	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0x9801	0x9900	R	NODE11	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x9901	0x9A00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0x9A01	0x9B00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x9B01	0x9C00	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0x9C01	0x9D00	R	NODE12	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x9D01	0x9E00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0x9E01	0x9F00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x9F01	0xA000	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xA001	0xA100	R	NODE13	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xA101	0xA200	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xA201	0xA300	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xA301	0xA400	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xA401	0xA500	R	NODE14	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xA501	0xA600	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xA601	0xA700	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xA701	0xA800	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS

Start address	End address	Access	Node	Use
0xA801	0xA900	R	NODE15	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xA901	0xAA00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xAA01	0xAB00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xAB01	0xAC00	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xAC01	0xAD00	R	NODE16	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xAD01	0xAE00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xAE01	0xAF00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xAF01	0xB000	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xB001	0xB100	R	NODE17	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xB101	0xB200	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xB201	0xB300	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xB301	0xB400	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xB401	0xB500	R	NODE18	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xB501	0xB600	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xB601	0xB700	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xB701	0xB800	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xB801	0xB900	R	NODE19	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xB901	0xBA00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xBA01	0xBB00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xBB01	0xBC00	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xBC01	0xBD00	R	NODE20	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xBD01	0xBE00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xBE01	0xBF00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xBF01	0xC000	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xC001	0xC100	R	NODE21	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xC101	0xC200	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xC201	0xC300	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xC301	0xC400	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xC401	0xC500	R	NODE22	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xC501	0xC600	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xC601	0xC700	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xC701	0xC800	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xC801	0xC900	R	NODE23	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xC901	0xCA00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xCA01	0xCB00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xCB01	0xCC00	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xCC01	0xCD00	R	NODE24	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xCD01	0xCE00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS

Start address	End address	Access	Node	Use
0xCE01	0xCF00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xCF01	0xD000	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xD000	0xD100	R	NODE25	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xD101	0xD200	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xD201	0xD300	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xD301	0xD400	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xD401	0xD500	R	NODE26	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xD501	0xD600	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xD601	0xD700	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xD701	0xD800	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xD801	0xD900	R	NODE27	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xD901	0xDA00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xDA01	0xDB00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xDB01	0xDC00	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xDC01	0xDD00	R	NODE28	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xDD01	0xDE00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xDE01	0xDF00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xDF01	0xE000	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xE001	0xE100	R	NODE29	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xE101	0xE200	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xE201	0xE300	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xE301	0xE400	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xE401	0xE500	R	NODE30	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xE501	0xE600	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xE601	0xE700	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xE701	0xE800	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xE801	0xE900	R	NODE31	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xE901	0xEA00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xEA01	0xEB00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xEB01	0xEC00	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xEC01	0xED00	R	NODE32	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xED01	0xEE00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xEE01	0xEF00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xEF01	0xF000	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS

High byte								Low byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								Reserved			Dis	Test	Fault	Alarm	Pre

- **Pre:** If bit 0 is 1, then the device is in prealarm status.
- **Alarm:** If bit 1 is 1, then the device is in alarm status.
- **Fault:** If bit 2 is 1, then the device is in fault status.
- **Test:** If bit 3 is 1, then the device is in test.
- **Dis:** If bit 4 is 1, then the device is disabled.

Zone mode

Control Panel Global Status

This status consists of two read-only holding registers. These registers represent the global status for all control panels in the network. Each register is divided into two bytes.

Start address	End address	Access	Use
0x1001	0x1001	R	GLOBAL_NODE_STATUS1 (1 to 128 Nodes)
0x1002	0x1002	R	GLOBAL_NODE_STATUS2 (1 to 128 Nodes)

Status 1

High byte								Low byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Sounder status							Reserved		Functional conditions					

Bit	High byte	Bit	Low byte
8	Sounder delay enabled	0	Alarm functional condition
9	Sounder activation delay in progress	1	Fault functional condition
10	Sounder outputs activated	2	Disable functional condition
11	Sounder outputs silenced	3	Test functional condition
12	Sounders disabled	4	Day/Night mode (1=nightmode/0=daymode)
13	Sounder override time elapsed	5	MCP alarm
14	Sounders in Test	6	Reserved
15	Reserved	7	Reserved

- **Alarm:** If bit 0 is 1, then one or more control panels are in alarm status.
- **Fault:** If bit 1 is 1, then one or more control panels are in fault status.
- **Disabled:** If bit 2 is 1, then one or more control panels are disabled.
- **Test:** If bit 3 is 1, then one or more control panels are in test status.

Status 2

High byte								Low byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		Fire protection status						Res.	Fire routing status						

Bit	High byte	Bit	Low byte
8	Fire Protection: Delay enabled	0	Fire Routing: Delay enabled
9	Fire Protection: Activation delay in progress	1	Fire Routing: Activation delay in progress
10	Fire Protection: Outputs activated	2	Fire Routing: Outputs activated
11	Fire Protection: Acknowledged	3	Fire Routing: Acknowledged
12	Fire Protection: Disabled	4	Fire Routing: Disabled
13	Fire Protection: Test ON	5	Fire Routing: Test ON
14	Reserved	6	Fire Routing: Ext. activation delay in progress
15	Reserved	7	Reserved

Control Panel Status

This status consists of 512 read-only holding registers (128 nodes, 4 registers per node). These registers represent all status options for a single node (control panel). Each register is divided into two bytes.

Start address	End address	Access	Use
0x2001	0x2200	R	NODE1_STATUS1 - NODE_128_STATUS4

Addr.	Value	Addr.	Value	Addr.	Value	Addr.	Value	Addr.	Value
0x2001	NODE1_ST1	0x2069	NODE27_ST1	0x20D1	NODE53_ST1	0x2139	NODE79_ST1	0x21A1	NODE105_ST1
0x2002	NODE1_ST2	0x206A	NODE27_ST2	0x20D2	NODE53_ST2	0x213A	NODE79_ST2	0x21A2	NODE105_ST2
0x2003	NODE1_ST3	0x206B	NODE27_ST3	0x20D3	NODE53_ST3	0x213B	NODE79_ST3	0x21A3	NODE105_ST3
0x2004	NODE1_ST4	0x206C	NODE27_ST4	0x20D4	NODE53_ST4	0x213C	NODE79_ST4	0x21A4	NODE105_ST4
0x2005	NODE2_ST1	0x206D	NODE28_ST1	0x20D5	NODE54_ST1	0x213D	NODE80_ST1	0x21A5	NODE106_ST1
0x2006	NODE2_ST2	0x206E	NODE28_ST2	0x20D6	NODE54_ST2	0x213E	NODE80_ST2	0x21A6	NODE106_ST2
0x2007	NODE2_ST3	0x206F	NODE28_ST3	0x20D7	NODE54_ST3	0x213F	NODE80_ST3	0x21A7	NODE106_ST3
0x2008	NODE2_ST4	0x2070	NODE28_ST4	0x20D8	NODE54_ST4	0x2140	NODE80_ST4	0x21A8	NODE106_ST4
0x2009	NODE3_ST1	0x2071	NODE29_ST1	0x20D9	NODE55_ST1	0x2141	NODE81_ST1	0x21A9	NODE107_ST1
0x200A	NODE3_ST2	0x2072	NODE29_ST2	0x20DA	NODE55_ST2	0x2142	NODE81_ST2	0x21AA	NODE107_ST2
0x200B	NODE3_ST3	0x2073	NODE29_ST3	0x20DB	NODE55_ST3	0x2143	NODE81_ST3	0x21AB	NODE107_ST3
0x200C	NODE3_ST4	0x2074	NODE29_ST4	0x20DC	NODE55_ST4	0x2144	NODE81_ST4	0x21AC	NODE107_ST4
0x200D	NODE4_ST1	0x2075	NODE30_ST1	0x20DD	NODE56_ST1	0x2145	NODE82_ST1	0x21AD	NODE108_ST1
0x200E	NODE4_ST2	0x2076	NODE30_ST2	0x20DE	NODE56_ST2	0x2146	NODE82_ST2	0x21AE	NODE108_ST2
0x200F	NODE4_ST3	0x2077	NODE30_ST3	0x20DF	NODE56_ST3	0x2147	NODE82_ST3	0x21AF	NODE108_ST3
0x2010	NODE4_ST4	0x2078	NODE30_ST4	0x20E0	NODE56_ST4	0x2148	NODE82_ST4	0x21B0	NODE108_ST4

Addr.	Value	Addr.	Value	Addr.	Value	Addr.	Value	Addr.	Value
0x2011	NODE5_ST1	0x2079	NODE31_ST1	0x20E1	NODE57_ST1	0x2149	NODE83_ST1	0x21B1	NODE109_ST1
0x2012	NODE5 _ST2	0x207A	NODE31_ST2	0x20E2	NODE57_ST2	0x214A	NODE83_ST2	0x21B2	NODE109_ST2
0x2013	NODE5 _ST3	0x207B	NODE31_ST3	0x20E3	NODE57_ST3	0x214B	NODE83_ST3	0x21B3	NODE109_ST3
0x2014	NODE5 _ST4	0x207C	NODE31_ST4	0x20E4	NODE57_ST4	0x214C	NODE83_ST4	0x21B4	NODE109_ST4
0x2015	NODE6_ST1	0x207D	NODE32_ST1	0x20E5	NODE58_ST1	0x214D	NODE84_ST1	0x21B5	NODE110_ST1
0x2016	NODE6 _ST2	0x207E	NODE32_ST2	0x20E6	NODE58_ST2	0x214E	NODE84_ST2	0x21B6	NODE110_ST2
0x2017	NODE6 _ST3	0x207F	NODE32_ST3	0x20E7	NODE58_ST3	0x214F	NODE84_ST3	0x21B7	NODE110_ST3
0x2018	NODE6 _ST4	0x2080	NODE32_ST4	0x20E8	NODE58_ST4	0x2150	NODE84_ST4	0x21B8	NODE110_ST4
0x2019	NODE7 _ST1	0x2081	NODE33_ST1	0x20E9	NODE59_ST1	0x2151	NODE85_ST1	0x21B9	NODE111_ST1
0x201A	NODE7 _ST2	0x2082	NODE33_ST2	0x20EA	NODE59_ST2	0x2152	NODE85_ST2	0x21BA	NODE1011_ST2
0x201B	NODE7 _ST3	0x2083	NODE33_ST3	0x20EB	NODE59_ST3	0x213	NODE85_ST3	0x21BB	NODE111_ST3
0x201C	NODE7 _ST4	0x2084	NODE33_ST4	0x20EC	NODE59_ST4	0x2154	NODE5 _ST4	0x21BC	NODE111_ST4
0x201D	NODE8_ST1	0x2085	NODE34_ST1	0x20ED	NODE60_ST1	0x2155	NODE86_ST1	0x21BD	NODE112_ST1
0x201E	NODE8_ST2	0x2086	NODE34_ST2	0x20EE	NODE60_ST2	0x2156	NODE86_ST2	0x21BE	NODE112_ST2
0x201F	NODE8_ST3	0x2087	NODE34_ST3	0x20EF	NODE60_ST3	0x2157	NODE86_ST3	0x21BF	NODE112_ST3
0x2020	NODE8_ST4	0x2088	NODE34_ST4	0x20F0	NODE60_ST4	0x2158	NODE86_ST4	0x21C0	NODE112_ST4
0x2021	NODE9_ST1	0x2089	NODE35_ST1	0x20F1	NODE61_ST1	0x2159	NODE87_ST1	0x21C1	NODE113_ST1
0x2022	NODE9_ST2	0x208A	NODE35_ST2	0x20F2	NODE61_ST2	0x215A	NODE87_ST2	0x21C2	NODE113_ST2
0x2023	NODE9_ST3	0x208B	NODE35_ST3	0x20F3	NODE61_ST3	0x215B	NODE87_ST3	0x21C3	NODE113_ST3
0x2024	NODE9_ST4	0x208C	NODE35_ST4	0x20F4	NODE61_ST4	0x215C	NODE87_ST4	0x21C4	NODE113_ST4
0x2025	NODE10_ST1	0x208D	NODE36_ST1	0x20F5	NODE62_ST1	0x215D	NODE88_ST1	0x21C5	NODE114_ST1
0x2026	NODE10_ST2	0x208E	NODE36_ST2	0x20F6	NODE62_ST2	0x215E	NODE88_ST2	0x21C6	NODE114_ST2
0x2027	NODE10_ST3	0x208F	NODE36_ST3	0x20F7	NODE62_ST3	0x215F	NODE88_ST3	0x21C7	NODE114_ST3
0x2028	NODE10_ST4	0x2090	NODE36_ST4	0x20F8	NODE62_ST4	0x2160	NODE88_ST4	0x21C8	NODE114_ST4
0x2029	NODE11_ST1	0x2091	NODE37_ST1	0x20F9	NODE63_ST1	0x2161	NODE89_ST1	0x21C9	NODE115_ST1
0x202A	NODE11_ST2	0x2092	NODE37_ST2	0x20FA	NODE63_ST2	0x2162	NODE89_ST2	0x21CA	NODE115_ST2
0x202B	NODE11_ST3	0x2093	NODE37_ST3	0x20FB	NODE63_ST3	0x2163	NODE89_ST3	0x21CB	NODE115_ST3
0x202C	NODE11_ST4	0x2094	NODE37_ST4	0x20FC	NODE63_ST4	0x2164	NODE89_ST4	0x21CC	NODE115_ST4
0x202D	NODE12_ST1	0x2095	NODE38_ST1	0x20FD	NODE64_ST1	0x2165	NODE90_ST1	0x21CD	NODE116_ST1
0x202E	NODE12_ST2	0x2096	NODE38_ST2	0x20FE	NODE64_ST2	0x2166	NODE90_ST2	0x21CE	NODE116_ST2
0x202F	NODE12_ST3	0x2097	NODE38_ST3	0x20FF	NODE64_ST3	0x2157	NODE90_ST3	0x21CF	NODE116_ST3
0x2030	NODE12_ST4	0x2098	NODE38_ST4	0x2100	NODE64_ST4	0x2158	NODE90_ST4	0x21D0	NODE116_ST4
0x2031	NODE13_ST1	0x2099	NODE39_ST1	0x2101	NODE65_ST1	0x2159	NODE91_ST1	0x21D1	NODE117_ST1
0x2032	NODE13_ST2	0x209A	NODE39_ST2	0x2102	NODE65_ST2	0x215A	NODE91_ST2	0x21D2	NODE117_ST2
0x2033	NODE13_ST3	0x209B	NODE39_ST3	0x2103	NODE65_ST3	0x215B	NODE91_ST3	0x21D3	NODE117_ST3
0x2034	NODE13_ST4	0x209C	NODE39_ST4	0x2104	NODE65_ST4	0x215C	NODE91_ST4	0x21D4	NODE117_ST4
0x2035	NODE14_ST1	0x209D	NODE40_ST1	0x2105	NODE66_ST1	0x215D	NODE92_ST1	0x21D5	NODE118_ST1
0x2036	NODE14_ST2	0x209E	NODE40_ST2	0x2106	NODE66_ST2	0x215E	NODE92_ST2	0x21D6	NODE118_ST2

Addr.	Value	Addr.	Value	Addr.	Value	Addr.	Value	Addr.	Value
0x2037	NODE14_ST3	0x209F	NODE40_ST3	0x2107	NODE66_ST3	0x215F	NODE92_ST3	0x21D7	NODE118_ST3
0x2038	NODE14_ST4	0x20A0	NODE40_ST4	0x2108	NODE66_ST4	0x2170	NODE92_ST4	0x21D8	NODE118_ST4
0x2039	NODE15_ST1	0x20A1	NODE41_ST1	0x2109	NODE67_ST1	0x2171	NODE93_ST1	0x21D9	NODE119_ST1
0x203A	NODE15_ST2	0x20A2	NODE41_ST2	0x210A	NODE67_ST2	0x2172	NODE93_ST2	0x21DA	NODE119_ST2
0x203B	NODE15_ST3	0x20A3	NODE41_ST3	0x210B	NODE67_ST3	0x2173	NODE93_ST3	0x21DB	NODE119_ST3
0x203C	NODE15_ST4	0x20A4	NODE41_ST4	0x210C	NODE67_ST4	0x2174	NODE93_ST4	0x21DC	NODE119_ST4
0x203D	NODE16_ST1	0x20A5	NODE42_ST1	0x210D	NODE68_ST1	0x2175	NODE94_ST1	0x21DD	NODE120_ST1
0x203E	NODE16_ST2	0x20A6	NODE42_ST2	0x210E	NODE68_ST2	0x2176	NODE94_ST2	0x21DE	NODE120_ST2
0x203F	NODE16_ST3	0x20A7	NODE42_ST3	0x210F	NODE68_ST3	0x2177	NODE94_ST3	0x21DF	NODE120_ST3
0x2040	NODE16_ST4	0x20A8	NODE42_ST4	0x2110	NODE68_ST4	0x2178	NODE94_ST4	0x21E0	NODE120_ST4
0x2041	NODE17_ST1	0x20A9	NODE43_ST1	0x2111	NODE69_ST1	0x2179	NODE95_ST1	0x21E1	NODE121_ST1
0x2042	NODE17_ST2	0x20AA	NODE43_ST2	0x2112	NODE69_ST2	0x217A	NODE95_ST2	0x21E2	NODE121_ST2
0x2043	NODE17_ST3	0x20AB	NODE43_ST3	0x2113	NODE69_ST3	0x217B	NODE95_ST3	0x21E3	NODE121_ST3
0x2044	NODE17_ST4	0x20AC	NODE43_ST4	0x2114	NODE69_ST4	0x217C	NODE95_ST4	0x21E4	NODE121_ST4
0x2045	NODE18_ST1	0x20AD	NODE44_ST1	0x2115	NODE70_ST1	0x217D	NODE96_ST1	0x21E5	NODE122_ST1
0x2046	NODE18_ST2	0x20AE	NODE44_ST2	0x2116	NODE70_ST2	0x217E	NODE96_ST2	0x21E6	NODE122_ST2
0x2047	NODE18_ST3	0x20AF	NODE44_ST3	0x2117	NODE70_ST3	0x217F	NODE96_ST3	0x21E7	NODE122_ST3
0x2048	NODE18_ST4	0x20B0	NODE44_ST4	0x2118	NODE70_ST4	0x2180	NODE96_ST4	0x21E8	NODE122_ST4
0x2049	NODE19_ST1	0x20B1	NODE45_ST1	0x2119	NODE71_ST1	0x2181	NODE97_ST1	0x21E9	NODE123_ST1
0x204A	NODE19_ST2	0x20B2	NODE45_ST2	0x211A	NODE71_ST2	0x2182	NODE97_ST2	0x21EA	NODE123_ST2
0x204B	NODE19_ST3	0x20B3	NODE45_ST3	0x211B	NODE71_ST3	0x2183	NODE97_ST3	0x21EB	NODE123_ST3
0x204C	NODE19_ST4	0x20B4	NODE45_ST4	0x211C	NODE71_ST4	0x2184	NODE97_ST4	0x21EC	NODE123_ST4
0x204D	NODE20_ST1	0x20B5	NODE46_ST1	0x211D	NODE72_ST1	0x2185	NODE98_ST1	0x21ED	NODE124_ST1
0x204E	NODE20_ST2	0x20B6	NODE46_ST2	0x211E	NODE72_ST2	0x2186	NODE98_ST2	0x21EE	NODE124_ST2
0x204F	NODE20_ST3	0x20B7	NODE46_ST3	0x211F	NODE72_ST3	0x2187	NODE98_ST3	0x21EF	NODE124_ST3
0x2050	NODE20_ST4	0x20B8	NODE46_ST4	0x2120	NODE72_ST4	0x2188	NODE98_ST4	0x21F0	NODE124_ST4
0x2051	NODE21_ST1	0x20B9	NODE47_ST1	0x2121	NODE73_ST1	0x2189	NODE99_ST1	0x21F1	NODE125_ST1
0x2052	NODE21_ST2	0x20BA	NODE47_ST2	0x2122	NODE73_ST2	0x218A	NODE99_ST2	0x21F2	NODE125_ST2
0x2053	NODE21_ST3	0x20BB	NODE47_ST3	0x2123	NODE73_ST3	0x218B	NODE99_ST3	0x21F3	NODE125_ST3
0x2054	NODE21_ST4	0x20BC	NODE47_ST4	0x2124	NODE73_ST4	0x218C	NODE99_ST4	0x21F4	NODE125_ST4
0x2055	NODE22_ST1	0x20BD	NODE48_ST1	0x2125	NODE74_ST1	0x218D	NODE100_ST1	0x21F5	NODE126_ST1
0x2056	NODE22_ST2	0x20BE	NODE48_ST2	0x2126	NODE74_ST2	0x218E	NODE100_ST2	0x21F6	NODE126_ST2
0x2057	NODE22_ST3	0x20BF	NODE48_ST3	0x2127	NODE74_ST3	0x218F	NODE100_ST3	0x21F7	NODE126_ST3
0x2058	NODE22_ST4	0x20C0	NODE48_ST4	0x2128	NODE74_ST4	0x2190	NODE100_ST4	0x21F8	NODE126_ST4
0x2059	NODE23_ST1	0x20C1	NODE49_ST1	0x2129	NODE75_ST1	0x2191	NODE101_ST1	0x21F9	NODE127_ST1
0x205A	NODE23_ST2	0x20C2	NODE49_ST2	0x212A	NODE75_ST2	0x2192	NODE101_ST2	0x21FA	NODE127_ST2
0x205B	NODE23_ST3	0x20C3	NODE49_ST3	0x212B	NODE75_ST3	0x2193	NODE101_ST3	0x21FB	NODE127_ST3
0x205C	NODE23_ST4	0x20C4	NODE49_ST4	0x212C	NODE75_ST4	0x2194	NODE101_ST4	0x21FC	NODE127_ST4

Addr.	Value	Addr.	Value	Addr.	Value	Addr.	Value	Addr.	Value
0x205D	NODE24_ST1	0x20C5	NODE50_ST1	0x212D	NODE76_ST1	0x2195	NODE102_ST1	0x21FD	NODE128_ST1
0x205E	NODE24_ST2	0x20C6	NODE50_ST2	0x212E	NODE76_ST2	0x2196	NODE102_ST2	0x21FE	NODE128_ST2
0x205F	NODE24_ST3	0x20C7	NODE50_ST3	0x212F	NODE76_ST3	0x2197	NODE102_ST3	0x21FF	NODE128_ST3
0x2060	NODE24_ST4	0x20C8	NODE50_ST4	0x2130	NODE76_ST4	0x2198	NODE102_ST4	0x2200	NODE128_ST4
0x2061	NODE25_ST1	0x20C9	NODE51_ST1	0x2131	NODE77_ST1	0x2199	NODE103_ST1		
0x2062	NODE25_ST2	0x20CA	NODE51_ST2	0x2132	NODE77_ST2	0x219A	NODE103_ST2		
0x2063	NODE25_ST3	0x20CB	NODE51_ST3	0x2133	NODE77_ST3	0x219B	NODE103_ST3		
0x2064	NODE25_ST4	0x20CC	NODE51_ST4	0x2134	NODE77_ST4	0x219C	NODE103_ST4		
0x2065	NODE26_ST1	0x20CD	NODE52_ST1	0x2135	NODE78_ST1	0x219D	NODE104_ST1		
0x2066	NODE26_ST2	0x20CE	NODE52_ST2	0x2136	NODE78_ST2	0x219E	NODE104_ST2		
0x2067	NODE26_ST3	0x20CF	NODE52_ST3	0x2137	NODE78_ST3	0x219F	NODE104_ST3		
0x2068	NODE26_ST4	0x20D0	NODE52_ST4	0x2138	NODE78_ST4	0x21A0	NODE104_ST4		

Status 1

High byte								Low byte							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Sounder status							Reserved		Functional conditions					

Bit	High byte	Bit	Low byte
8	Sounder delay enabled	0	Alarm functional condition
9	Sounder activation delay in progress	1	Fault functional condition
10	Sounder outputs activated	2	Disable functional condition
11	Sounder outputs silenced	3	Test functional condition
12	Sounders disabled	4	Day/Night mode (1=nightmode/0=daymode)
13	Sounder override time elapsed	5	MCP alarm
14	Sounders in Test	6	Reserved
15	Reserved	7	Reserved

Status 2

High byte								Low byte							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		Fire protection status						Res.	Fire routing status						

Bit	High byte	Bit	Low byte
8	Fire Protection: Delay enabled	0	Fire Routing: Delay enabled
9	Fire Protection: Activation delay in progress	1	Fire Routing: Activation delay in progress
10	Fire Protection: Outputs activated	2	Fire Routing: Outputs activated
11	Fire Protection: Acknowledged	3	Fire Routing: Acknowledged
12	Fire Protection: Disabled	4	Fire Routing: Disabled
13	Fire Protection: Test ON	5	Fire Routing: Test ON
14	Reserved	6	Fire Routing: Ext. activation delay in progress
15	Reserved	7	Reserved

Status 3

High byte								Low byte							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved								Reserved							

Status 4

High byte								Low byte							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved								Reserved							

Control Panel Zone Status

This status consists of 32767 read-only holding registers (128 nodes, 512 zones per node, 2 zones per register). Each register represents the status for two zones per node (control panel). Each register is divided into two bytes.

Start address	End address	Access	Node	Use
0x3001	0x3100	R	NODE1	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x3101	0x3200	R	NODE2	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x3201	0x3300	R	NODE3	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x3301	0x3400	R	NODE4	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x3401	0x3500	R	NODE5	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x3501	0x3600	R	NODE6	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x3601	0x3700	R	NODE7	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x3701	0x3800	R	NODE8	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x3801	0x3900	R	NODE9	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x3901	0x3A00	R	NODE10	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x3A01	0x3B00	R	NODE11	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x3B01	0x3C00	R	NODE12	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x3C01	0x3D00	R	NODE13	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x3D01	0x3E00	R	NODE14	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x3E01	0x3F00	R	NODE15	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x3F01	0x4000	R	NODE16	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x4001	0x4100	R	NODE17	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x4101	0x4200	R	NODE18	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x4201	0x4300	R	NODE19	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x4301	0x4400	R	NODE20	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x4401	0x4500	R	NODE21	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x4501	0x4600	R	NODE22	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x4601	0x4700	R	NODE23	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x4701	0x4800	R	NODE24	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x4801	0x4900	R	NODE25	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x4901	0x4A00	R	NODE26	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x4A01	0x4B00	R	NODE27	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x4B01	0x4C00	R	NODE28	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x4C01	0x4D00	R	NODE29	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x4D01	0x4E00	R	NODE30	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x4E01	0x4F00	R	NODE31	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x4F01	0x5000	R	NODE32	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x5001	0x5100	R	NODE33	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x5101	0x5200	R	NODE34	ZONE_1&2 STATUS to ZONE_511&512 STATUS

Start address	End address	Access	Node	Use
0x5201	0x5300	R	NODE35	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x5301	0x5400	R	NODE36	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x5401	0x5500	R	NODE37	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x5501	0x5600	R	NODE38	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x5601	0x5700	R	NODE39	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x5701	0x5800	R	NODE40	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x5801	0x5900	R	NODE41	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x5901	0x5A00	R	NODE42	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x5A01	0x5B00	R	NODE43	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x5B01	0x5C00	R	NODE44	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x5C01	0x5D00	R	NODE45	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x5D01	0x5E00	R	NODE46	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x5E01	0x5F00	R	NODE47	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x5F01	0x6000	R	NODE48	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x6001	0x6100	R	NODE49	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x6101	0x6200	R	NODE50	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x6201	0x6300	R	NODE51	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x6301	0x6400	R	NODE52	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x6401	0x6500	R	NODE53	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x6501	0x6600	R	NODE54	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x6601	0x6700	R	NODE55	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x6701	0x6800	R	NODE56	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x6801	0x6900	R	NODE57	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x6901	0x6A00	R	NODE58	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x6A01	0x6B00	R	NODE59	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x6B01	0x6C00	R	NODE60	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x6C01	0x6D00	R	NODE61	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x6D01	0x6E00	R	NODE62	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x6E01	0x6F00	R	NODE63_	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x6F01	0x7000	R	NODE64	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x7001	0x7100	R	NODE65	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x7101	0x7200	R	NODE66	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x7201	0x7300	R	NODE67	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x7301	0x7400	R	NODE68	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x7401	0x7500	R	NODE69	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x7501	0x7600	R	NODE70	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x7601	0x7700	R	NODE71	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x7701	0x7800	R	NODE72	ZONE_1&2 STATUS to ZONE_511&512 STATUS

Start address	End address	Access	Node	Use
0x7801	0x7900	R	NODE73	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x7901	0x7A00	R	NODE74	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x7A01	0x7B00	R	NODE75	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x7B01	0x7C00	R	NODE76	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x7C01	0x7D00	R	NODE77	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x7D01	0x7E00	R	NODE78	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x7E01	0x7F00	R	NODE79	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x7F01	0x8000	R	NODE80	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x8001	0x8100	R	NODE81	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x8101	0x8200	R	NODE82	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x8201	0x8300	R	NODE83	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x8301	0x8400	R	NODE84	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x8401	0x8500	R	NODE85	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x8501	0x8600	R	NODE86	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x8601	0x8700	R	NODE87	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x8701	0x8800	R	NODE88	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x8801	0x8900	R	NODE89	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x8901	0x8A00	R	NODE90	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x8A01	0x8B00	R	NODE91	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x8B01	0x8C00	R	NODE92	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x8C01	0x8D00	R	NODE93	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x8D01	0x8E00	R	NODE94	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x8E01	0x8F00	R	NODE95	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x8F01	0x9000	R	NODE96	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x9001	0x9100	R	NODE97	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x9101	0x9200	R	NODE98	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x9201	0x9300	R	NODE99	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x9301	0x9400	R	NODE100	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x9401	0x9500	R	NODE101	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x9501	0x9600	R	NODE102	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x9601	0x9700	R	NODE103	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x9701	0x9800	R	NODE104	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x9801	0x9900	R	NODE105	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x9901	0x9A00	R	NODE106	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x9A01	0x9B00	R	NODE107	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x9B01	0x9C00	R	NODE108	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x9C01	0x9D00	R	NODE109	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x9D01	0x9E00	R	NODE110	ZONE_1&2 STATUS to ZONE_511&512 STATUS

Start address	End address	Access	Node	Use
0x9E01	0x9F00	R	NODE111	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0x9F01	0xA000	R	NODE112	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0xA001	0xA100	R	NODE113	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0xA101	0xA200	R	NODE114	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0xA201	0xA300	R	NODE115	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0xA301	0xA400	R	NODE116	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0xA401	0xA500	R	NODE117	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0xA501	0xA600	R	NODE118	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0xA601	0xA700	R	NODE119	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0xA701	0xA800	R	NODE120	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0xA801	0xA900	R	NODE121	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0xA901	0xAA00	R	NODE122	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0xAA01	0xAB00	R	NODE123	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0xAB01	0xAC00	R	NODE124	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0xAC01	0xAD00	R	NODE125	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0xAD01	0xAE00	R	NODE126	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0xAE01	0xAF00	R	NODE127	ZONE_1&2 STATUS to ZONE_511&512 STATUS
0xAF01	0xB000	R	NODE128	ZONE_1&2 STATUS to ZONE_511&512 STATUS

High byte								Low byte							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			Dis	Test	Fault	Alarm	Alert	Reserved			Dis	Test	Fault	Alarm	Alert

- **Alert:** If bit 0 is 1, then one or more zone devices are in prealarm status.
- **Alarm:** If bit 1 is 1, then one or more zone devices are in alarm status.
- **Fault:** If bit 2 is 1, then one or more zone devices are in fault status.
- **Test:** If bit 3 is 1, then the zone is in test.
- **Dis:** If bit 4 is 1, then the zone is disabled.

Write holding registers

This command consists of 8 write holding registers. Each register is divided into two bytes.

Start address	End address	Access	Use
0x0001	0x0001	Write (W)	Execute Reset
0x0002	0x0002	Write (W)	Execute Panel Silence
0x0003	0x0003	Write (W)	Set Sounders Start/Stop
0x0004	0x0004	Write (W)	Set Sounders Delay On/Off
0x0005	0x0005	Write (W)	Set Fire Protection Delay On/Off
0x0006	0x0006	Write (W)	Execute Fire Protection Override Delay
0x0007	0x0007	Write (W)	Set Fire Routing On/Off
0x0008	0x0008	Write (W)	Execute Fire Routing Override Delay
0xFFFF	0xFFFF	Write (W)	Used to reset the inactivity time counter (prevents the TCP/IP connection from closing automatically)

Sending commands

Use the following for sending all commands listed in this section.

Item	Description
Panel ID	The address of the control panel where the command is to be executed
0	Executes the command at the local control panel
1-128	Executes the command at the address specified
65535	Broadcasts the command to all panels

Execute Reset

Resets the control panel.

High byte								Low byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Panel ID															

Execute Panel Silence

Silences the control panel buzzer.

High byte								Low byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Panel ID															

Execute Fire Protection Override Delay

Executes the fire protection override delay.

High byte								Low byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Panel ID															

Execute Fire Routing Override Delay

Executes the fire routing override delay.

High byte								Low byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Panel ID															

Set Sounders Start/Stop

Activates the sounders.

High byte								Low byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Panel ID								Reserved							Start/ Stop

Start/Stop: if bit 0 is 1, then sounders are activated (sounding), if 0 they are silenced.

Set Sounder Delay On/Off

Enables or disables a configured sounder delay.

High byte								Low byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Panel ID								Reserved							On/ Off

On/Off: If bit 0 is 1, then the sounder delay is enabled, if 0 it is disabled.

Set Fire Protection Delay On/Off

Enables or disables a configured fire protection delay.

High byte								Low byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Panel ID								Reserved							On/ Off

On/Off: If bit 0 is 1, then the fire protection delay is enabled, if 0 it is disabled.

Set Fire Routing Delay On/Off

Enables or disables a configured fire routing delay.

High byte								Low byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Panel ID								Reserved							On/ Off

On/Off: If bit 0 is 1, then the fire routing delay is enabled, if 0 it is disabled.

Heartbeat

The control panel may automatically close the TCP/IP connection after 1 minute of inactivity (no requests received from the Modbus client). This write holding register request (Heartbeat) can be used to reset the inactivity counter and keep the TCP/IP connection open.

High byte								Low byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Other settings

Maximum Polling Frequency

This setting defines the minimum interval of time between two Modbus TCP/IP client requests. The resolution of this setting is approximately 1000 ms.

After sending a request to the control panel, the Modbus TCP/IP client shall wait 1000 ms before sending the next request.

Maximum Quantity of Registers

This setting defines the maximum quantity of registers in the Read Holding Registers function. The value of this field for control panel Modbus applications is four registers.

Examples

Zone/Point mode

BMS Configuration:

- Protocol MODBUS
- Initial Panel 001
- Mode ZONEPOINT

Execute Broadcast Reset

In the request, the request register address is set to 00 00 hex (request to write the register 00 01 hex) and the register value is set to FF FF hex.

The response is an echo of the request, returned if contents of the register have been successfully written.

	Sequence (HEX)
Request	01 00 00 00 00 06 00 06 00 00 FF FF
Response	01 00 00 00 00 06 00 06 00 00 FF FF

Execute Start Sounders in the Panel ID 2

In the request, the request register address is set to 00 02 hex (request to write register 00 03 ex) and the register value is set to 02 01 hex.

The response is an echo of the request, returned if contents of the register have been successfully written.

	Sequence (HEX)
Request	01 00 00 00 00 06 00 06 00 02 02 01
Response	01 00 00 00 00 06 00 06 00 02 02 01

Read Panel ID 1 Status

In the request, the starting register address is set to 20 00 hex (request to read register 20 01 hex) and the quantity is set to 00 04 hex.

The response contains 8 bytes with values of registers 20 01 – 20 04 hex.

	Sequence (HEX)
Request	01 00 00 00 00 06 00 03 20 00 00 04
Response	01 00 00 00 00 0B 00 03 08 00 12 00 00 03 00 00 01

The table below shows contents of registers:

Register	Description	Register Value (HEX)	Node Status
0x2001	NODE1_ST1	00 12	Fault functional condition Night Mode
0x2002	NODE1_ST2	00 00	-
0x2003	NODE1_ST3	03 00	Fault Counter = 3 Alarm Counter = 0
0x2004	NODE1_ST4	00 01	Condition Counter = 1

Read Panel ID 2, Loop 1, Device 6-7 Status

In the request, the starting register address is set to 74 05 hex (request to read register 74 06 hex) and the quantity is set to 00 04 hex.

The response contains 8 bytes with values of registers 74 06 – 74 09 hex.

	Sequence (HEX)
Request	01 00 00 00 00 06 00 03 74 05 00 04
Response	01 00 00 00 00 0B 00 03 08 00 00 00 02 00 00 00 00

The table below shows contents of registers:

Register	Description	Register Value (HEX)	Device Status
0x7406	NODE2_LOOP1_ DEVICE 6 STATUS	00 00	-
0x7407	NODE2_LOOP1_ DEVICE 7 STATUS	00 02	Device 7 in alarm
0x7408	NODE2_LOOP1_ DEVICE 8 STATUS	00 00	-
0x7409	NODE2_LOOP1_ DEVICE 9 STATUS	00 00	-

BMS Configuration:

- Protocol MODBUS
- Initial Panel 002
- Mode ZONEPOINT

Read Panel ID 2, Zone 1-2

In the request, the starting register address is set to 30 00 hex (request to read register 30 01 hex) and the quantity is set to 00 02 hex.

The response contains 4 bytes with values of registers 30 01 – 30 02 hex.

	Sequence (HEX)
Request	01 00 00 00 00 06 00 03 30 00 00 02
Response	01 00 00 00 00 0B 00 03 04 00 02 00 04

The table below shows contents of registers:

Register	Description	Register value (HEX)	Zone status
0x3001	NODE1_ZONE 1 STATUS	00 02	Zone 1 in Alarm
0x3002	NODE1_ZONE 2 STATUS	00 04	Zone 2 in Fault

Zone mode

BMS Configuration:

- Protocol MODBUS
- Initial Panel 001
- Mode ZONE

Read Panel ID 2, Zone 1-8 Status

In the request, the starting register address is set to 31 00 hex (request to read register 31 01 hex) and the quantity is set to 00 04 hex.

The response contains 8 bytes with values of registers 31 01 – 31 04 hex.

	Sequence (HEX)
Request	01 00 00 00 00 06 00 03 31 00 00 04
Response	01 00 00 00 00 0B 00 03 08 00 02 00 00 06 10 00 00

The table below shows contents of registers:

Register	Description	Register value (HEX)	Zone status
0x3101	NODE2_ZONE 1 & 2 STATUS	00 02	Zone 1 Status = 02 (in Alarm) Zone 2 Status = 00
0x3102	NODE2_ZONE 3 & 4 STATUS	00 00	Zone 3 Status = 00 Zone 4 Status = 00
0x3103	NODE2_ZONE 5 & 6 STATUS	06 10	Zone 5 Status = 10 (Disabled) Zone 6 Status = 06 (in Alarm and Fault)
0x3104	NODE2_ZONE 7 & 8 STATUS	00 00	Zone 7 Status = 00 Zone 8 Status = 00