BEVENT RASCH

RCBK4-MOD / RCMU8-MOD

MODBUS RTU INTERFACED RCBK4 / RCMU8

TECHNICAL DESCRIPTION

Project:	604
Journal:	RCxxxMOD_V101_20120705
Date:	2012-07-05
Document version:	1.15.604.1
Pages:	29

CONTENTS

1. INFORMATION

1.1 PRODUCT

RCBK4-MOD / RCMU8-MOD.

Modbus RTU Interface Module for RCBK4 / RCMU8.

Copyright © 2008-2012 by Puls Design A/S.

1.2 CONTACT INFORMATION

Contact information:

Puls Design A/S
Telefon +45 36 49 37 54
Telefax +45 36 49 37 55

Lars Pehrson (LUP)
lup@puls-design.dk
Tel. +45 28 10 08 65 (direct)

1.3 NEWEST REVISION

The newest revision of Modbus-module firmware and documentation can always be obtained at web site www.puls-design.eu under *Support*.

2. REVISION

Rev.	Date	Init.	Description
1.00.604.1	2007-01-26	LUP	First draft.
1.01.604.1	2007-03-29	LUP	Translated to English and added a few topics. Changed to LONWORKS variables. Added Modbus reference.
1.02.604.1	2008-05-19	JPH	Modifications and updates.
1.03.604.1	2008-05-29	JPH	Major Modbus profile modifications.
1.04.604.1	2008-07-01	JPH	All structure descriptions are documented.
1.05.604.1	2008-07-04	JPH	Sub final prototype document (T0.90).
1.06.604.1	2008-07-08	JPH	Final prototype document (T0.90).
1.07.604.1	2009-10-06	JPH	Test release document (T0.95).
1.08.604.1	2009-10-29	JPH	Test release document (T0.97), error corrections.
1.09.604.1	2009-11-11	JPH	Test release document (T.098), telegram example added, RCBK4-MOD added.
1.10.604.1	2010-02-13	JPH	Test release document (T0.99), evolution.
1.11.604.1	2010-04-21	JPH	Final test release candidate (T1.00), functional test interval can be read/written at network and section level.
1.12.604.1	2010-04-29	JPH	Final test release candidate (T1.01) for the both RCMU8-MOD and RCBK4-MOD. LED function description added for hardware version 1 and 2.
1.14.604.1	2010-05-21	JPH	Picture of RCBK4-MOD. Minor corrections.
			V1.01 document. Small corrections. Billede af RCBK4-MOD, - billede af DIP-switch og LED’s tæt på - Regin-rettelser - styr på figur-numre
1.15.604.1	2012-07-05	JPH	Final V1.01 release document. Minor grammatic corrections. Picture of LED’s and DIP-switch. Correction of register address to poll group 0 status level (272 decimal, but 270 in earlier texts).

3. INTRODUCTION

The control units RCBK4 and RCMU8 are used for interfacing to and control of fire dampers and fire sensors. As a standard feature these control units can be linked together in a special network system that enables simple control and monitoring of the units and their connected equipment. As an option the built in network system can be expanded with a Modbus RTU interface module for networking in an open system.

Please read the documentation on RCBK4 and RCMU8 found on the Bevent Rasch website www.bevent-rasch.com for more information on how the device works.

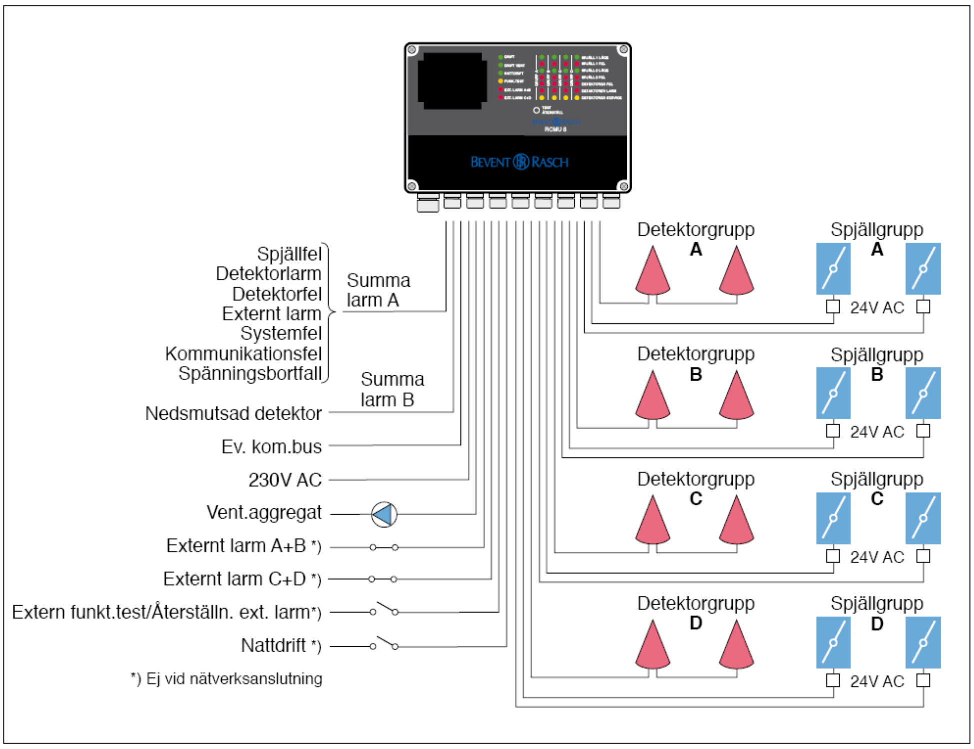


Figure 2, RCMU8 block diagram example.

3.1 INTERNAL FUNCTIONALITY

Internally the RCBK4-MOD and RCMU8-MOD contain an extension module for enabling the standard RCBK4/RCMU8 to Modbus RTU. This module – called the *gateway* module - contains some 'intelligence' of its own which works in parallel with the RCBK4/RCMU8. This means the RCBK4/RCMU8 will in most regards work as it does in standard mode – still monitoring the status of the fire detectors and dampers connected to it. If for example the fire detector of one group has detected smoke, the device will go into alarm mode and close the dampers for that particular group. This way the RCBK4/RCMU8 still works autonomously even when connected to the Modbus gateway module. The big difference is the ability to monitor the status via Modbus and we can force the RCBK4/RCMU8 into other states. We can monitor this via Modbus and then force dampers of all other or selected groups on that Modbus network. This functionality is the responsibility of the Modbus integrator.

3.1.1 Gateway module

The gateway module handles all Modbus related communication and communication with the RCBK4-MOD or RCMU8-MOD device. The gateway module has recently been revised into hardware revision 2 which amongst other things includes the addition of a red error LED and faster internal processing. The hardware revision 2 is found in following devices:

Device	Hardware revision	Software revision	Serial number
RCMU8-MOD	1	T0.xx	<= 16397[1]
RCMU8-MOD	1	V1.00	<= 16400
RCMU8-MOD	2	V1.01	> 16400
RCBK4-MOD	2	V1.01	All

Version V1.00 and V1.01 are functionally identical, but differs on the gateway modules hardware revision.

3.1.2 Module DIP-switch

The 10-switch-DIP on the front of the RCBK4-MOD/RCMU8-MOD is for configuring the Modbus-interface address and baudrate; see section 7.2.

3.1.3 Internal communication

The gateway module is constantly communicating with the RCBK4/RCMU8 to read out status and to set the RCBK4/RCMU8 into states dictated by the Modbus master. If the communication between the gateway module and the RCBK4/RCMU8 for some reason is broken, the internal communication is said to fail.

3.1.4 External alarm

On RCBK4-MOD/RCMU8-MOD the electrical input(s) for activating the external alarm is enabled though a few seconds cycle delay must be expected. Once activated the external alarm can only be deactivated by power cycle, the reset button or by transmitting a deactivation command via the Modbus status register; see section 5.6.1. Using this command scheme external alarm can also be activated.

3.1.5 Night Mode

The night mode function cannot be activated by the electrical input on the RCBK4/RCMU8 main board as is the case in standard RCBK4/RCMU8 mode, but the Modbus status register command scheme can be used to activate or deactivate night mode; see section 5.6.1.

3.1.6 Functional test

A standard RCBK4/RCMU8 does a timed functional test for every 48 hours – read more about this in the RCBK4/RCMU8 documentation. On an RCBK4-MOD/RCMU8-MOD this timed functional test is trigged by the gateway module. By a default the cycle time is 48 hours although this can be altered by writing to the appropriate register at network level or at section level. Writing a new cycle time on

[1] Exceptions are devices sent for upgrade after production.

either of these levels will affect an RCBK4-MOD or an RCMU8-MOD at unit level; see section 5.7.

It is also possible to trig the functional test via Modbus – writing to the status register of the Modbus profile; see section 5.6.1.

The functional test cannot be trigged by the electrical input on the RCBK4/RCMU8 main board as is the case in standard RCBK4/RCMU8 mode.

3.1.7 Groups

The RCBK4 has two groups while the RCMU8 has four groups. They are by default all separated so if the detector of group B detects smoke, it will close damper B1 and B2 (two dampers per group). The other groups (A for RCBK4 and A, C, D for RCMU8) will work undisturbed by this only closing their dampers if their particular fire detector finds smoke.

The Modbus gateway makes it possible to read out specific status on all these four groups (and each four of the detectors and eight dampers). Overruling the states of the dampers and controlling other features is done on an entire unit basis.

3.1.8 Control functions

The Modbus gateway enables you to control the following functions of an entire unit basis (all groups at the same time):

- Forcing the dampers in closed position,
- Forcing the dampers in open position,
- Starting functional test of dampers and
- Setting the device in night mode.

See RCBK4/RCMU8 description by Bevent Rasch for details about these concepts.

3.1.9 Baseboard DIP-switch

The 3 DIP-switches on the RCBK4 main printed circuit board and the 5 DIP-switches on the RCMU8 main printed circuit board are ignored in the MOD-versions and they are therefore irrelevant in this case. All references to a certain DIP-switch in this document, is the 10-switch DIP on the front of the RCBK4-MOD/RCMU8-MOD.

4. BEVENT RASCH STANDARD PROFILE

The Modbus-compatible MRB-products from Bevent Rasch are all designed to fit into a general profile description as possible. This profile description not only includes descriptions of single devices equipped with a Modbus-interface, but also multiple devices in a sub network controlled by a Modbus-device or gateway. This allows a reusable model of devices regardless of being in a network or not.

4.1 TERMS

The profile description defines the following terms in a hierarchical manner: *Network*, *Section*, *Unit* and *Group*. The *Network* term describes the whole profile including one or more *Sections*. In each *Section* one or more units are included. *Units* correspond normally to physical devices and they can each include a number of groups. One *Group* could for example be a detector and a damper set.

4.2 CONCEPT

To grasp all of Bevent Rasch MRB-devices produces now and in the future, a general network scheme as shown on figure 3 has been developed:

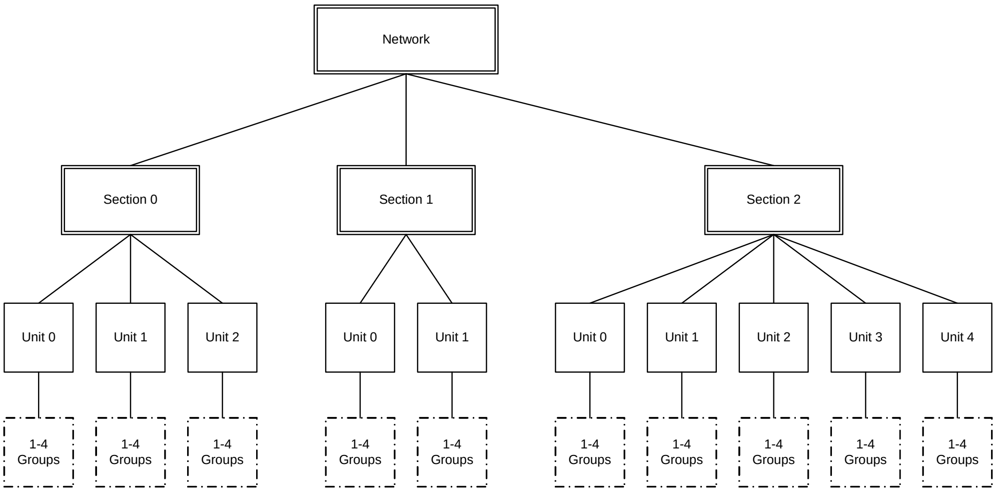


Figure 3, Network hierarchy of Bevent Rasch Modbus profiles

Within one Modbus Gateway the following number of logical entities can be administered:

Level type	Maximum count	Remarks
Network	1	Only one network per gateway
Sections	Maximum 16	Sections administer units
Units	Maximum 120 per Section	Units correspond to devices and administer one or more groups
Groups	Maximum 4 per Unit	One group can have one detector

4.3 KEEPING IT SIMPLE

Understanding the concept for RCBK4 or RCMU8 and the concept of adapting it to a Modbus compatible network might be quite a lot to deal with at once. This is why we recommend you look at the concept in a simple way unless you really need to explore all the possibilities with this device.

The RCBK4 and RCMU8 is not the only Bevent Rasch product which will be Modbus enabled. To be able to keep control of the devices as easy as possible you are encouraged to look at the devices at the Network, Section level or the Unit level. From here the profiles will seem quite alike from product to product. In the case of the RCBK4-MOD and RCMU8-MOD you will get the exact same behaviour on the Network level, Section level as on the Unit level.

4.3.1 Future devices

The benefits you gain from using the Network level or Section level is if you one day replace your RCBK4-MOD or RCMU8-MOD with a similar Bevent Rasch product. One day Bevent Rasch also might offer a Modbus enabled master device which can control a number of slaves. In that case all devices can be monitored at Network and Section level.

4.4 REGISTER SPACE AND LAYOUT

The standard profile for all Bevent Rasch modules is laid out as in figure 4. All current devices have a specific profile which fits inside this standard profile:

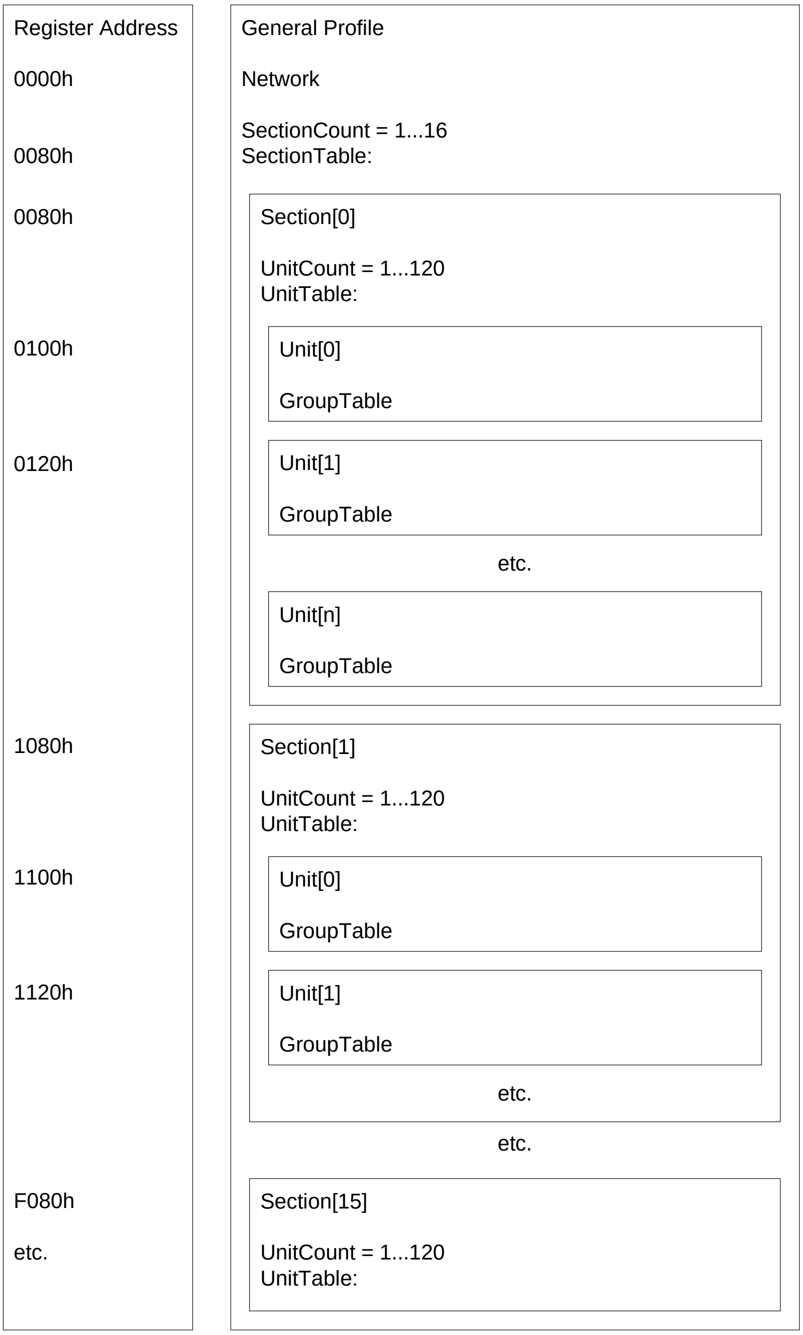


Figure 4, General Profile Description.

4.5 RCBK4-MOD PROFILE

Following is a description of the profile specific for RCBK4-MOD. Note that this device has only one single section, one unit and two groups defined:

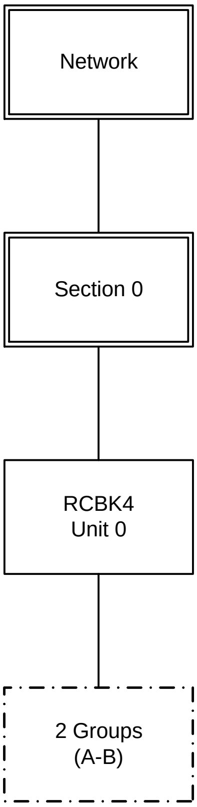


Figure 5, Network hierarchy of RCBK4-MOD

4.6 RCMU8-MOD PROFILE

Following is a description of the profile specific for RCMU8-MOD. Note that this device has only one single section, one unit and four groups defined:

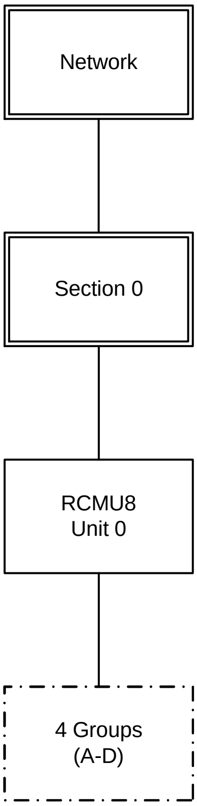
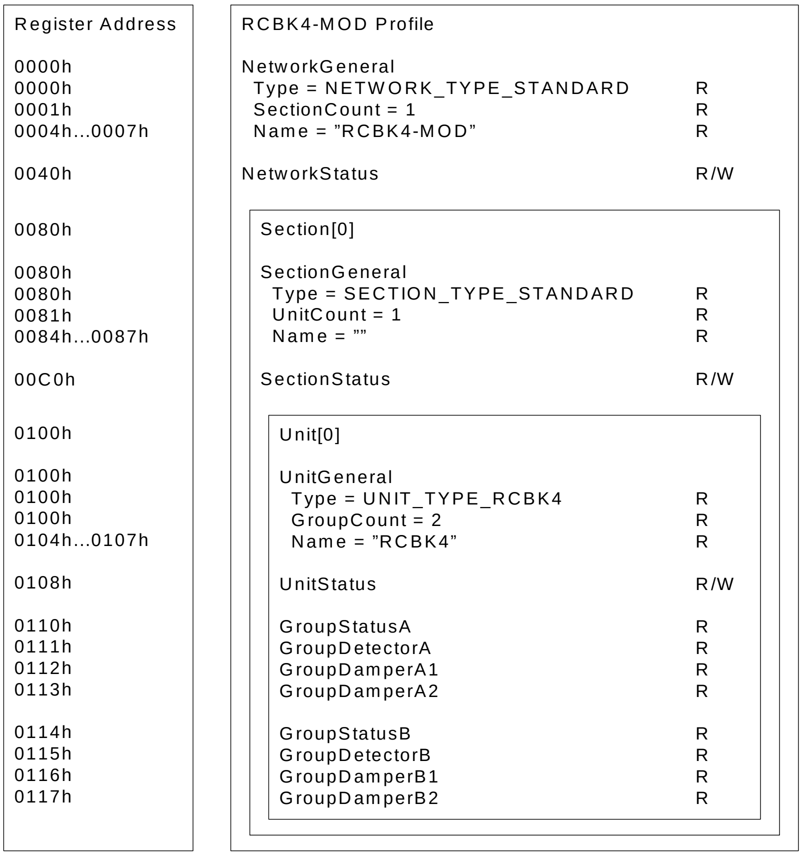


Figure 6, Network hierarchy of RCMU8-MOD

4.6.1 RCBK4-MOD structure layout

Figure 7 shows the detailed structure layout of the RCBK4-MOD:



Register Address	RCBK4-MOD Profile	
0000h	NetworkGeneral	
0000h	Type = NETWORK_TYPE_STANDARD	R
0001h	SectionCount = 1	R
0004h...0007h	Name = ”RCBK4-MOD”	R
0040h	NetworkStatus	R/W
0080h	Section[0]	
0080h	SectionGeneral	
0080h	Type = SECTION_TYPE_STANDARD	R
0081h	UnitCount = 1	R
0084h...0087h	Name = ””	R
00C0h	SectionStatus	R/W
0100h	Unit[0]	
0100h	UnitGeneral	
0100h	Type = UNIT_TYPE_RCBK4	R
0100h	GroupCount = 2	R
0104h...0107h	Name = ”RCBK4”	R
0108h	UnitStatus	R/W
0110h	GroupStatusA	R
0111h	GroupDetectorA	R
0112h	GroupDamperA1	R
0113h	GroupDamperA2	R
0114h	GroupStatusB	R
0115h	GroupDetectorB	R
0116h	GroupDamperB1	R
0117h	GroupDamperB2	R

Figure 7, RCBK4-MOD profile description.

4.6.2 RCMU8-MOD structure layout

Figure 8 shows the detailed structure layout of the RCMU8-MOD:

Register Address	RCMU8-MOD Profile	
0000h	NetworkGeneral	
0000h	Type = NETWORK_TYPE_STANDARD	R
0001h	SectionCount = 1	R
0004h...0007h	Name = ”RCMU8-MOD”	R
0040h	NetworkStatus	R/W
0080h	Section[0]	
0080h	SectionGeneral	
0080h	Type = SECTION_TYPE_STANDARD	R
0081h	UnitCount = 1	R
0084h...0087h	Name = ””	R
00C0h	SectionStatus	R/W
0100h	Unit[0]	
0100h	UnitGeneral	
0100h	Type = UNIT_TYPE_RCMU8	R
0100h	GroupCount = 4	R
0104h...0107h	Name = ”RCMU8”	R
0108h	UnitStatus	R/W
0110h	GroupStatusA	R
0111h	GroupDetectorA	R
0112h	GroupDamperA1	R
0113h	GroupDamperA2	R
0114h	GroupStatusB	R
0115h	GroupDetectorB	R
0116h	GroupDamperB1	R
0117h	GroupDamperB2	R
0118h	GroupStatusC	R
0119h	GroupDetectorC	R
011Ah	GroupDamperC1	R
011Bh	GroupDamperC2	R
011Ch	GroupStatusD	R
011Dh	GroupDetectorD	R
011Eh	GroupDamperD1	R
011Fh	GroupDamperD2	R

Figure 8, RCMU8-MOD profile description.

5. REGISTER DEFINITIONS

The following pages define all registers of interest to you as a Modbus-integrator. This may be extended in the future.

Most registers are read-only (marked **R**) and exceptions will be returned when you try to write to those registers. Only a few writeable registers exist (marked **R/W**). Care should be taken to write to these registers in the documented manner.

Warning: The register definitions found in this document are the only supported definitions for RCBK4/RCMU-MOD. One might find more definitions in the include files and when comparing with future Modbus enabled Bevent Rasch products, but those definitions are not supported and can be changed without warning.

5.1.1 Hexadecimal values

In some parts of this description the term *hexadecimal* is used – with a radix or base of 16 (contrary 10 in the decimal system). This is a notation different from our normal decimal notation. For instance the number 23 can be written as 17 hexadecimal in that 1 x 16 + 7 = 23. In this document you will see numbers written as 23 (decimal) or *17h* (hexadecimal).

5.2 NETWORK REGISTERS

Each Modbus-enabled Bevent Rasch device will define the Network Registers as the first registers. In these registers you can read out the network type, the network name and most importantly the number of sections. In this case we have only one section with only one unit.

5.2.1 NetworkGeneral

The general description of the network most importantly defines the type and the number of sections:

Address space	Bit	Field name	R/W	Description
0x0000h	0…7	ucID[1]	R	Identification field which contains the characters “NW” (NetWork).
	8…15	ucID[0]	R	
0x0001h	0…7	ucMasterType	R	Section master type (= NETWORK_ TYPE_STANDARD = 10h)
	8…15	ucFormatVersion	R	Structure format (= 1)
0x0002h	0…15	uiSectionCount	R	Number of units in section (= 1)
0x0003h	0…15	uiFTInterval	R/W	Funtional Test Interval (minutes) Default 2880 min. (=48 hours), 0 = OFF
0x0004h…0x0007h	0…15	-	R	Reserved for future use (= 0)
0x0008h…0x000Fh	0…7 x 16	tczName[16]	R	Contains the section name = “RCMU8-MOD”

5.2.2 NetworkStatus

The status of the Network presents a status overview of the entire Network, all Sections and all Units in those sections. The *NetworkStatus* register is a calculated version of all *SectionStatus* registers OR’ed together:

Address space	Bit	Field name	R/W	Description
0x0040h	15	bNormal	R	(0) Alarm condition (1) Normal condition
	14	bFireAlarm	R	(0) No fire alarm condition (1) Alarm condition trigged from detector
	13	bExternalAlarm	R/W	(0) No external Alarm condition (1) External Alarm condition
	12	-	-	Reserved for future use
	11	bDetectorError	R	(0) No fire detector in error condition (1) Fire detector(s) error
	10	bDetectorService	R	(0) No fire detector in service condition (1) Fire detector(s) service
	9	bDamperError	R	(0) No damper(s) in error condition (1) Damper(s) in error condition
	8	bCommunicationError	R	(0) No communication problems to unit (1) Communication problems with unit
	7	bDampersForcedOpen	R/W	(0) Damper(s) not forced into open position (1) Damper(s) forced into open position
	6	bDampersForcedClosed	R/W	(0) Damper(s) not forced into closed position (1) Damper(s) forced into closed position
	5	bNightMode	R/W	(0) No groups in night mode condition (1) Group(s) in night mode condition
	4	bFunctionalTest	R/W	(0) No groups in functional test (1) Group(s) in functional test
	3…0	-	-	Reserved for future use

5.3 SECTION REGISTERS

One section can contain one or more units. A section is a local set of units meaning the units can easily coexist in the same physical network with units from other sections, but they are logically related to units only in the same section. In this case we have only one section with only one unit.

5.3.1 SectionGeneral

The general description of the section most importantly defines the ID and the number of units:

Address space	Bit	Field name	R/W	Description
0x0080h	0…7	ucID[1]	R	Identification field which contains the characters “SE” (SEction).
	8…15	ucID[0]	R	
0x0081h	0…7	ucMasterType	R	Section master type (= SECTION_ TYPE_STANDARD = 10h)
	8…15	ucFormatVersion	R	Structure format (= 1)
0x0082h	0…15	uiUnitCount	R	Number of units in section (= 1)

0x0083h	0…15	uiFTInterval	R/W	Funtional Test Interval (minutes) Default 2880 min. (=48 hours), 0 = OFF
0x0084h…0x0087h	0…15	-	R	Reserved for future use (= 0)
0x0088h…0x008Fh	0…7 x16	tczName[16]	R	Contains the section name = “”

5.3.2 SectionStatus

The status of the section presents a status overview of the section and the units in the section. The *SectionStatus* register is a calculated version of all the *UnitStatus* registers OR’ed together:

Address space	**Bit**	**Field name**	**R/W**	**Description**
0x00C0h	15	bNormal	R	(0) Alarm condition (1) Normal condition
	14	bFireAlarm	R	(0) No fire alarm condition (1) Alarm condition trigged from detector
	13	bExternalAlarm	R/W	(0) No external Alarm condition (1) External Alarm condition
	12	-	-	Reserved for future use
	11	bDetectorError	R	(0) No fire detector in error condition (1) Fire detector(s) error
	10	bDetectorService	R	(0) No fire detector in service condition (1) Fire detector(s) service
	9	bDamperError	R	(0) No damper(s) in error condition (1) Damper(s) in error condition
	8	bCommunicationError	R	(0) No communication problems to unit (1) Communication problems with unit
	7	bDampersForcedOpen	R/W	(0) Damper(s) not forced into open position (1) Damper(s) forced into open position
	6	bDampersForcedClosed	R/W	(0) Damper(s) not forced into closed position (1) Damper(s) forced into closed position
	5	bNightMode	R/W	(0) No groups in night mode condition (1) Group(s) in night mode condition
	4	bFunctionalTest	R/W	(0) No groups in functional test (1) Group(s) in functional test
	3…0	-	-	Reserved for future use

5.4 UNIT REGISTERS

One unit can manage one or more groups. An RCBK4 has two groups and an RCMU8 has four groups which are all separated when the unit is Modbus-enabled. So if one group is in alarm mode doesn't mean all groups are in alarm mode.

5.4.1 UnitGeneral

The general description of the unit most importantly defines the type, the unit identification and the number of groups:

Address space	Bit	Field name	R/W	Description
0x0100h	0…7	ucType	R	Type (e.g. UNIT_TYPE_RCMU8 = 14h)
	8…10	uiGroupCount	R	Number of groups (e.g. 4 for RCMU8)
	11	-	R	Reserved for future use
	12…15	uiHardwareVersion	R	Hardware version of unit
0x0101h	0…7	ucFirmwareRevision	R	Firmware revision of unit
	8…15	ucFirmwareVersion	R	Firmware version of unit
0x0102h	0…15	uiUnitID (bit 0-15)	R	32 bit unit identification number. A combination of type and serial number
0x0103h	0…15	uiUnitID (bit 16-31)	R	
0x0104h… 0x0107h	0…7 x 8	tczName[8]	R	Contains unit name e.g. "RCMU8"

5.4.2 UnitStatus

The status of the unit presents a status overview of the unit and the groups in the unit. The *UnitStatus* register is a calculated version of all the UnitStatus registers OR'ed together. The state of the internal communication between the Modbus gateway module and the actual RCBK4 or RCMU8 is reflected in the flag *bCommunicationError*.

Address space	Bit	Field name	R/W	Description
0x0108h	15	bNormal	R	(0) Alarm condition (1) Normal condition
	14	bFireAlarm	R	(0) No fire alarm condition (1) Alarm condition trigged from detector
	13	bExternalAlarm	R/W	(0) No external Alarm condition (1) External Alarm condition
	12	-	-	Reserved for future use
	11	bDetectorError	R	(0) No fire detector in error condition (1) Fire detector(s) error
	10	bDetectorService	R	(0) No fire detector in service condition (1) Fire detector(s) service
	9	bDamperError	R	(0) No damper(s) in error condition (1) Damper(s) in error condition
	8	bCommunicationError	R	(0) No communication problems in unit

				(1) Communication problems in unit
	7	bDampersForcedOpen	R/W	(0) Damper(s) not forced into open position (1) Damper(s) forced into open position
	6	bDampersForcedClosed	R/W	(0) Damper(s) not forced into closed position (1) Damper(s) forced into closed position
	5	bNightMode	R/W	(0) No groups in night mode condition (1) Group(s) in night mode condition
	4	bFunctionalTest (*)	R/W	(0) No groups in functional test (1) Group(s) in functional test
	3… 0	-	-	Reserved for future use

(*) Activating functional test will lead to closing and reopening of the dampers connected to the RCMU8. First group A and B will be tested and next group C and D will be tested. This will be reflected on the group status flags and the lamps on the device.

5.5 GROUP REGISTERS

Status can be monitored on group basis, but control is *not* offered at this level.

5.5.1 GroupStatus

Address space	**Bit**	**Field name**	**R/W**	**Description**
0x0110h (A) / 0x0114h (B) / 0x0118h (C) / 0x011Ch (D) *Note:* *Groups C and D are valid for RCMU8 only.*	15	bNormal	R	(0) Alarm condition (1) Normal condition
	14	bFireAlarm	R	(0) No fire alarm condition (1) Alarm condition trigged from detector
	13	bExternalAlarm	R	(0) No external Alarm condition (1) External Alarm condition
	12	-	-	Reserved for future use
	11	bDetectorError	R	(0) No fire detector in error condition (1) Fire detector(s) error
	10	bDetectorService	R	(0) No fire detector in service condition (1) Fire detector(s) service
	9	bDamperError	R	(0) No damper(s) in error condition (1) Damper(s) in error condition
	8	-	-	Reserved for future use
	7	bDampersForcedOpen	R	(0) Damper(s) not forced into open position (1) Damper(s) forced into open position
	6	bDampersForcedClosed	R	(0) Damper(s) not forced into closed position (1) Damper(s) forced into closed position
	5	bNightMode	R	(0) No groups in night mode condition (1) Group(s) in night mode condition
	4	bFunctionalTest (see 5.4.2 and (*) comments)	R	(0) No groups in functional test (1) Group(s) in functional test
	3… 0	-	-	Reserved for future use

5.5.2 GroupDetector

Fire detector status of group A to D:

Address space	Bit	Field name	R/W	Description
0x0111h (A) / 0x0115h (B) / 0x0119h (C) / 0x011Dh (D) *Note:* *Groups C and D are valid for RCMU8 only.*	15	bConnected (*)	R	(0) No connection to fire detector (1) Normal connection to fire detector
	14	bAlarm	R	(0) No fire alarm condition (1) Fire alarm trigged from fire detector
	13	bService	R	(0) No service condition (1) Fire detector service condition
	12	-	-	Reserved for future use
	11	bUnstable	R	(0) No unstable error condition (1) Fire detector unstable error condition
	10	bShortCurcuit (*)	R	(0) No short circuit error condition (1) Fire detector short circuit condition
	9…8	-	-	Reserved for future use
	7	bUnknownError	R	(0) No unknown error condition (1) Fire detector unknown error condition
	6…0	-	-	Reserved for future use

(*) Short circuit of a detector will lead to no connection of the other groups' detectors.

5.5.3 GroupDamper

Damper status of group A to D for set 1 and 2:

Address space	Bit	Field name	R/W	Description
0x0112h (A1) / 0x0113h (A2) / 0x0116h (B1) / 0x0117h (B2) / 0x011Ah (C1) / 0x011Bh (C2) / 0x011Eh (D1) / 0x011Fh (D2) *Note:* *Groups C1, C2, D1 and D2 are valid for RCMU8 only.*	15	bConnected (*)	R	(0) No connection to damper (1) Normal connection to damper
	14	bUnstable	R	(0) No unstable error condition (1) Damper unstable error condition
	13	bShortCurcuit (*)	R	(0) No short circuit error condition (1) Damper short circuit condition
	12	bVolapyk	R	(0) No volapyk error condition (1) Damper volapyk error condition
	11	bTimoutToActive	R	(0) No switch error. (1) Error when moving to active position.
	10	bTimoutToPassive	R	(0) No switch error. (1) Error when moving to passive position.
	9	bNoSignalAtActive	R	(0) No switch error. (1) Missing signal at active position.
	8	bNoSignalAtPassive	R	(0) No switch error. (1) Missing signal at passive position.
	7	bUnknown	R	(0) Known state or error. (1) Unknown state or error.

	6…0	iPosition	R	Position of damper in degrees [0…90]: 0 – Closed damper 45 – Traveling between open and closed (*2) 90 – Open damper

(*) Short circuit of a damper motor will lead to the setting of the *bDamperShortCuircuit* error condition for all groups as well as the connected flag will be true even if no damper was connected to that particular group.

(*2) In some devices it is be possible to read out the exact damper angle. In that case traveling from for example open to closed will be read out as values continuously being altered from 89 down to 1 until the damper is actually closed at 0 degrees. So a general way to test if a damper is traveling would be for example: IF((iPosition > 0) AND (iPosition < 90)) THEN traveling.

5.6 STATUS LEVEL

As already mentioned the level of monitoring is greater than the level of control. In the case of RCBK4-MOD and RCMU8-MOD control at the group level is not possible while monitoring is:

Level	Can read status	Can write commands	Status register address
Network	Yes	Yes	0040h (64 dec)
Section	Yes	Yes	00C0h (192 dec)
Unit	Yes	Yes	0108h (264 dec)
Group	Yes	No	0110h… (272 dec)…

5.6.1 Writing commands

Following commands are supported in the RCMU4-MOD and RCMU8-MOD at Network, Section and Unit levels:

Name	Value	Description
COMMAND_EXTERNALALARM_ACTIVATE	A000h	Activated external alarm mode
COMMAND_EXTERNALALARM_DEACTIVATE	2000h	Deactivated external alarm mode
COMMAND_FORCEDAMPER_CLOSE	8040h	Forces dampers into closed position
COMMAND_FORCEDAMPER_OPEN	8080h	Forces dampers into open position
COMMAND_FORCEDAMPER_DEACTIVATE	00C0h	Deactivates any forced damper position
COMMAND_FUNCTIONALTEST_ACTIVATE	8010h	Trig functional test (cannot be cancelled, but will complete after 5-10 minutes test cycle)
COMMAND_NIGHTMODE_ACTIVATE	8020h	Activate night mode
COMMAND_NIGHTMODE_DEACTIVATE	0020h	Deactivate night mode

5.7 FUNCTIONAL TEST CYCLE TIME

The functional test cycle time can be read and written at two levels: Network level and section level. In RCBK4-MOD and RCMU8-MOD the values on these to levels are directed to the same internal value. Changing the value in one level will change the value at the other level as well.

Reading the default value of the functional test cycle time in the registers at section or network level, will return the value 2880 minutes which corresponds to 2880 / 60 = 48 hours.

The value can be set freely to between 0 and 65535. When setting the value to 0, times functional test is disabled.

Level	Reading value possible	Writing value possible	Register address
Network	Yes	Yes	0003h (3 dec)
Section	Yes	Yes	0083h (131 dec)

5.8 C INCLUDE FILES

Users writing their own applications in the C/C++ language will benefit from a package of header files for the MODBUS-enabled Bevent Rasch-products. This package can be downloadet at www.puls-design.eu under *Support, Modbus*.

5.9 C TIPS

When programming Modbus-integration in the language C you should also be aware of the structure definitions for RCBK4-MOD and RCMU8-MOD:

```
stNetworkRCBK4_t
stNetworkRCMU8_t
```

These structures define the entire register layout for the devices. It includes network, section, unit and group definitions. These definitions are hierarchical so you can use subsets if needed.

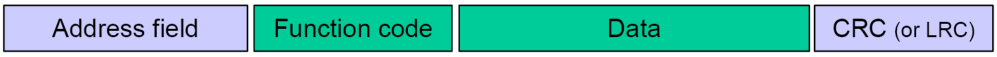
5.9.1 Created with Visual Studio 2005

The include file(s) are created with Microsoft® Visual Studio 2005. All bits defined should correspond exactly to those described in this document when using the Visual Studio 2005 compiler. Using another compiler – for example a compiler for an embedded OS – you should take great care about where the bits are actually mapped (you could define your own include files to match the compiler you use).

6. PROTOCOL

6.1 PROTOCOL

The transmission mode used is RTU mode.



Address field	Function code	Data	CRC (or LRC)

Figure 9, Modbus RTU frame.

6.1.1 ADDRESS FIELD

The device can be addressed using addresses from 0 to 247 (requests at address 0 are broadcast requests and will not be replied to).

6.1.2 FUNCTION CODES AND DATA

6.1.2.1 Function code 03 (0x03) Read Holding Registers

This function code allows values to be read one or more registers in the RCBK4-MOD/RCMU8-MOD device (all valid registers can be read, hence marked R on figure 6).

6.1.2.2 Function code 06 (0x06) Write Single Register

This function code allows values to be set in writable registers (marked R/W).

6.1.2.3 Function code 43 (0x2B) Read Device Identification

This function is a standard Modbus device identification. So far only mandatory objects have been implemented:

VendorName	= “Bevent Rasch”
ProductCode	= “604”
MajorMinorRevision	= “T1.00”, “T1.01” or “V1.01”

6.1.3 CHECKSUM

The protocol checksum is standard CRC16.

7. RTU INTERFACE

7.1 REFERENCE DOCUMENTS

Please refer to Modbus IDA at www.modbus.org in the *Technical Resources* link and following documents:

A. MODBUS Application Protocol Specification, V1.1b (or later) and
B. MODBUS over Serial Line, Specification and Implementation Guide, V1.02 (or later).

Further more RS485 information can be found in:

C. ANSI/TIA/EIA-485-A-1998 (US) or ISO/IEC-8482-1993 (“RS485” standard specifications).

7.2 NETWORK TOPOLOGY

The Modbus RS485-variant is working as a master-slave system. The RCBK4-MOD/RCMU8-MOD device always functions as a slave.

7.3 ELECTRICAL INTERFACE

The electrical interface is a two wire RS485 interface supporting baud rates of 9600 or 19200, 8 data bits, even parity and one stop bit.

7.3.1 Terminals

Terminal 27 and 28 on the RCBK4-MOD are used for Modbus:

RCBK4-MOD Terminal	RS485	2W-MODBUS	MDW-45	Termination
28	A/A’	D0	T+ (3)	Pull down
27	B/B’	D1	T- (4)	Pull up to 5V

Terminal 49 and 50 on the RCMU8-MOD are used for Modbus:

RCMU8-MOD Terminal	RS485	2W-MODBUS	MDW-45	Termination
50	A/A’	D0	T+ (3)	Pull down
49	B/B’	D1	T- (4)	Pull up to 5V

Shown here are also the connection terminals to a Westermo MDW-45 RS232 to RS422/RS485 converter for test purposes (please go to www.westermo.se for more information on these devices).

7.3.2 BUS-termination

The Modbus device can be configured to bus termination of 120Ω in series with a 1nF capacitor. This termination is activated when DIP switch 10 is ON (default).

7.4 BAUD RATE

The baud rate is selected with DIP switch 9.

DIP switch 9	Baud rate	Remarks
OFF	19200	8 data bits, even parity, one stop bit
ON	9600	8 data bits, even parity, one stop bit (default)

Note: This configuration is only read during power up of the entire device or after resetting the Modbus gateway module lying inside the device.

7.5 MODULE ADDRESS

The Modbus address is set with DIP switches 1-8 on the device. All addresses in the range of 1 to 247 are supported. This address can be changed during operation.

Address	**DIP switch**								**Description**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	
0	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	Broadcast address
1	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	Valid address (default)
2	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	Valid address
3	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	Valid address
4	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	Valid address
5…246	…	…	…	…	…	…	…	…	Valid addresses
247	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	Valid address
248	ON	ON	ON	ON	ON	OFF	OFF	OFF	Invalid address
…	…	…	…	…	…	…	…	…	Invalid addresses
255	ON	ON	ON	ON	ON	ON	ON	ON	Invalid address (standalone)

7.5.1 Standalone

If the Modbus address is set to 255 at power on, the device enters *standalone* mode. In this case the internal Modbus gateway module is not communicating with the RCMU8 or via Modbus. This means the RCMU8 runs stand alone and corresponds to normal RCMU8 mode without any network connection.

If the RCMU8-MOD has been configured for a valid address at an earlier point and one or more dampers is connected to RCMU8-MOD, the RCMU8 itself will enter fault mode; the DRIFT lamp will blink fast. To get the RCMU8 out of fault mode, disconnect the dampers, press the reset button and after start up, the RCMU8 is in normal standalone mode; DRIFT will be constantly lit. Then power down the RCMU8-MOD, reconnect the dampers and power up again.

7.6 LEDS

In hardware revision 1 gateway modules only the yellow communication LED is present. In hardware revision 2 gateway modules the red error LED was added; se hardware revision section 3.1.1. You will find the LED's to the left of the 10 switch DIP (revision 2 showed):

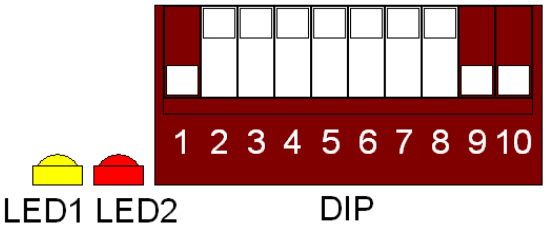


Figure 10, Communication and error LED

7.6.1 Communication LED

The communication LED1 hashes during reception and transmission of packages. It is placed on the left side next to the DIP-switch. Reception is defined as reception of broadcast packages, packages for this device and packages for other devices. Transmission of packages is defined as being packages sent only from this device.

7.6.2 Error LED

The error LED2 is placed next to the communication LED in hardware revision 2 modules.

7.6.3 Translation table

Following table is valid for understanding both hardware revision 1 and 2 modules:

Yellow LED	**Red LED**	**Status**	**Description**
Lit constantly	-	OK	Standalone mode (DIP address 255).
-	Lit constantly	Error	Internal Modbus module doesn't recognize the client or can't communicate with the client during power startup.
One blink	No blink	OK	Message received; no reply. Address doesn't match or broadcast address.
One blink	One blink	Error	Message received, but format is unknown or wrong.
Two blinks	No blink	OK	Message received; reply transmitted. Read or write accepted.
Two blinks	Two blink	Error	Message received, but message exception detected. More information on *exceptions* in official Modbus documentation.

8. EXAMPLE

This section provides an example of the communication telegrams to and from an RCBK4/RCMU8-MOD device.

8.1 CONFIGURATION

The Modbus address of the device is set to 1. This means DIP switch 1 is ON and DIP switches 2-8 are OFF. The communication speed is set to 9600 baud which means DIP switch 9 is set to ON.

Imagine the host to be a standard pc with an RS485 port. The port is configured at 9600,e,1 and communicates without an automatic handshake scheme.

8.2 READING STATUS

When settings the *Modbus address* on the DIP-switches to 1, we are required to communicate using this Modbus address. Reading the status at the network level, we need to read the register at *register address* 0040h or 64 decimal; see section 5.2.2 and section 5.6. It should be noted that in the case of all Modbus enabled *Bevent Rasch modules*, when we write register address 0040h, this is the exact address we communicate to the device. Using address 0, the address communicated is 0. Using address 1, the address communicated is 1 and so fourth.

8.2.1 Request

When reading one or more registers in a Modbus device, you can use the function Read Holding Registers (03)[2]; see section 6.1.2.1. This is a standard Modbus function supported by virtually all Modbus devices. The RFU-layout for this function request is:

Character	RTU level	PDU level	Example	Remarks
0	Modbus address		01h = 1	Address at DIP-switch
1	Modbus PDU	Function code	03h = 3	Read Holding Registers
2		Register address, Hi	00h = 0	Address 0040h = 64 dec
3		Register address, Lo	40h = 64	
4		Quantity of registers, Hi	00h = 0	0001h = 1 registers
5		Quantity of registers, Lo	01h = 1	
6	CRC, Lo		85h	CRC = DE85h = 56965 dec
7	CRC, Hi		DEh	

The CRC[3] value is a check value based on the other telegram values. If one telegram value is altered, the CRC value will also be altered. This means a single bit error introduced while a telegram is communicated, will introduce a mismatch to the CRC value and we can detect a communication error.

[2] See document A (in section 6.3 page 15) specified in section 7.1.

[3] See document B (in section 6.2.2 page 39) specified in section 7.1.

As stated in the table above, we will be sending a telegram with the 8 raw values in left to right order:

```
01h, 03h, 00h, 40h, 00h, 01h, 85h, DEh
```

If this function is successfully communicated to the device, the device will respond with the telegram described in the next section.

8.2.2 Response

Keep in mind all registers are 16 bit equal to 2 bytes wide. So when asking for one register, two bytes or 16 bits, will be returned to us. This will be a telegram with the following 7 characters (with CRC values as explained before) packages as:

```
01h, 03h, 02h, 80h, 00h, D9h, 84h
```

The characters have the following meaning:

Character	RTU level	PDU level	Example	Remarks
0	Modbus address		01h = 1	Address at DIP-switch
1	Modbus PDU	Function code	03h = 3	Read Holding Registers
2		Byte count	02h = 2	2 bytes (16 bits) returned
3		Registers value, Hi	80h = 128	Register 0064h content:
4		Registers value, Lo	00h = 0	8000h = 32768 = “normal”
5	CRC, Lo		D9h	CRC = 84D9h =
6	CRC, Hi		84h	34009 dec

All this means the value of register 0040h was returned. This is, as mentioned, the status of the device at network level. This register content was 8000h which means bit 15 is one, but bits 0 to 14 are all zero. Looking at the table in section 5.2.2, this corresponds to *normal condition*.

The status at section level, at device level or at group level can be read just as easily – just alter the register address according to the table in section 5.6.