

Information on controlling the Logamatic 5000/Control 8000 control device series via Modbus TCP/IP

This document will help you to implement the connection and control of the Logamatic 5000/Control 8000 control unit series via Modbus TCP/IP with your building management system.



Table of contents

1.	Interface description BCT 531/BCT 831	4
2.	Control of the control device via Modbus TCP/IP	5
2.1	Supported Modbus function codes	5
2.2	Supported Modbus register types	6
2.3	Determine relevant Modbus TCP/IP data points	6
2.4	Checking the current firmware of the control device	7
2.5	Necessary settings on the controller	8
2.6	Network behavior and diagnostic options of the controller	10
2.7	data types and register width	11
2.8	Control of the control device via Modbus TCP/IP from version 1.3 onwards.....	12
2.8.1	Temperature-controlled heat request via Modbus TCP/IP from and including version 1.3.....	14
2.8.2	Power-based heat request via Modbus TCP/IP from and including version 1.3.....	15
2.8.3	Information on parameters 'Heat request via Modbus' and 'Internal heat request '.....	16
2.8.3.1	Heat request exclusive mode	16
2.8.3.2	Parallel mode of heat demand.....	17
2.9	Control via Modbus TCP/IP in a network	17
2.9.1	Control via Modbus TCP/IP in the controller network.....	17
2.9.2	Control via Modbus TCP/IP in combination with Logaflow HSM plus ...	19
2.9.2.1	Access to substation.....	20
2.9.2.2	Access via segment substation to substation	21
3.	General information	22
3.1	Message concept of the control device.....	22
3.1.1	General information about the HMI status of the controller	23
3.1.2	General information on boiler malfunctions	24
3.1.3	General information on evaluating current messages on the control device	24
3.1.4	Example for reading a message key	25
3.2	Modbus TCP/IP Heart Beat	26
3.2.1	Variant 1: With Heart Beat.....	26
3.2.2	Variant 2: Without heart beat.....	26

3.3	Conversion of the Modbus address scheme.....	27
3.4	Setting or reading individual bits in a Modbus register	27
3.4.1	Setting individual bits in the Modbus register	28
3.4.2	Reading individual bits from Modbus registers.....	29
3.5	Converting values to ASCII characters	29
4.	Outdated software versions and their special features	31
4.1	Message behavior up to controller version 1.8	31
4.2	Control of the control device via Modbus TCP/IP up to and including version 1.2.7.....	32
4.2.1	Temperature-controlled heat request via Modbus TCP/IP up to and including version 1.2.7.....	33
4.2.2	Power-based heat request via Modbus TCP/IP up to and including version 1.2.7	34
4.2.3	Information on the 'Heat request only via Modbus ' parameter	34

1. Interface description BCT 531/BCT 831

If you would like to control the control device from a building management system (BMS) via Modbus TCP/IP, use a network cable to connect network interface 1 (LAN 1) of the control device (see Figure 1-1: Module BCT 531/BCT 831 – representation of the available interfaces) with your Machine network in which the BMS is also located. You can then put the control device into operation properly (see documentation “Service instructions for the specialist – Logamatic 5311/Control 8311” or “Service instructions for the specialist – Logamatic 5313/Control 8313”).

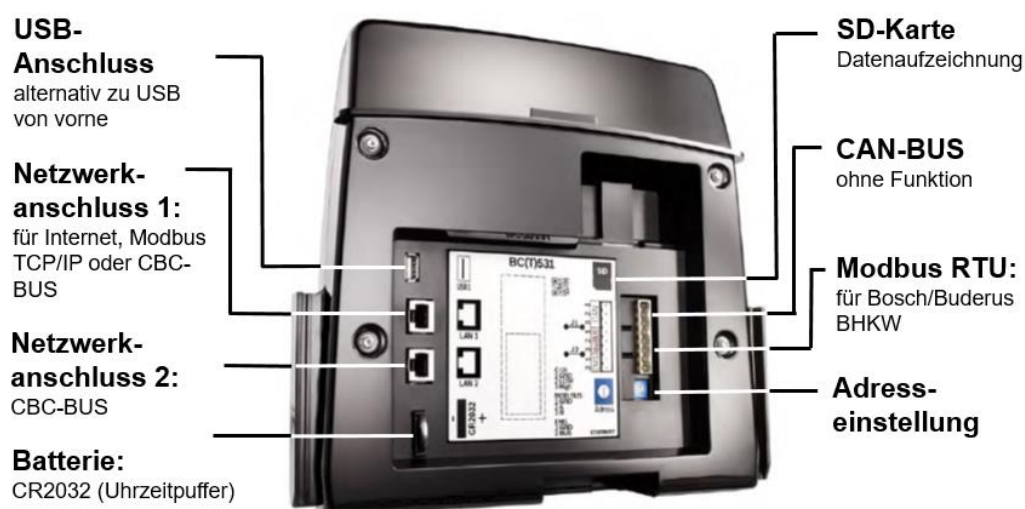


Figure 1-1: Module BCT 531/BCT 831 – representation of the available interfaces

2. Control of the control device via Modbus TCP/IP

In the following section you will find an excerpt of the cover sheet of the controller's data point list, which is relevant for setting up and using the Modbus TCP/IP communication. You can download the complete data point list from the Buderus website ¹.

The details of the Modbus addresses are listed in decimal format. To convert the addresses to the Modicon scheme, please continue reading in the chapter "Conversion of the Modbus address scheme". When using the data points, please ensure that you select the correct register types and carry out any necessary conversions of the received values. The conversions are necessary to transfer decimal places.

Notes :

- Starting with firmware version 1.3, the control, parameterization, and some Modbus addresses of the control device have changed via Modbus TCP/IP. Therefore, during commissioning, check which firmware your control device has (see chapter "Checking the current firmware of the control device") and, if necessary, carry out a firmware update first ². If you are already controlling a control device with a version below 1.3 using Modbus TCP/IP, you can activate the use of old Modbus addresses in your control device (see chapter "Necessary settings on the controller"). If you have activated the Modbus compatibility mode, you do not have to make any changes to your current solution for control via Modbus TCP/IP (see chapter "Outdated software versions and their special features"). Updating the firmware to the latest version is recommended.
- The decision as to whether the decimal or Modicon scheme must be used depends on the software used and the configuration of the BMS. Please speak to the manufacturer of your BMS for more information on the correct schematic.

2.1 Supported Modbus function codes

The Modbus TCP/IP server of the controller supports the following Modbus function codes:

¹ <https://www.buderus.de/de/5000-modbus>

²Software updates for the firmware of the Logamatic 5313 control unit and the associated instructions can be found on the Internet at <https://www.buderus.de/de/5000-software>

Function name	function code
<i>read coils</i>	01
<i>Read Discrete Inputs</i>	02
Read Multiple Holding Registers	03
Read input registers	04
<i>Write Single Coil</i>	05
Write single holding register	06
<i>Write Multiple Coils</i>	15
Write Multiple Holding Registers	16
Read/Write Multiple Registers	23

Notes :

- Function codes 01, 02, 05 and 15 are currently not used by the controller.

2.2 Supported Modbus register types

The Modbus TCP/IP server of the controller supports the following Modbus register types:

data type	Description
<i>coils</i>	<i>Readable and writable, 1 bit</i>
<i>discrete inputs</i>	<i>Readable, 1 bit</i>
input registers	Readable, 16 bits
holding registers	Readable and writable, 16 bits

Notes :

- Currently, the Coils and Discrete Inputs register types are not used by the control device.

2.3 Determine relevant Modbus TCP/IP data points

The data points that the control device provides or are specified in the data point list vary depending on the installed hardware modules and their configuration. Since the control device keeps all data points in the list, not all data points are relevant to you.

For example, if you have not connected a boiler to the control device and want to read out the relevant data points from the boiler, you will only receive invalid values (e.g. flow temperature from the boiler = 0 °C). When writing data points that are not

© All rights reserved by Robert Bosch GmbH, also in the case of intellectual property rights registrations

available in the control system, you can set the values, but these have no effect on the behavior of the controller.

The data point list is structured according to components. For example, if you don't use a boiler, you can ignore any data points that refer to the boiler component. Therefore, before using the data point list, check which data points are suitable for your application. Your Buderus/Bosch contact can help you select suitable data points for your system.

If you use heating circuit modules, you also need to pay attention to the slot. The slot determines the number of the heating circuit. You should also make sure that you have configured (e.g. activated) the components (heating circuits, boilers, etc.) in the control device accordingly.

2.4 Checking the current firmware of the control device

You can check the firmware used in the controller as follows:

First go to the information menu. Here click on the "Version" menu item. The current firmware can be found in the "Operating system" parameter. If the displayed firmware is lower than 1.3 or if you have activated the "Modbus compatibility for firmware lower than 1.3.6" mode, you must use the old Modbus data point list and control (see chapter "Outdated software versions and their special features").



Figure 2-1: Information menu

If, on the other hand, you are shown firmware higher than or equal to 1.3, please use the Modbus data point list that matches the firmware currently in use with the new control (see chapter "Control device control Control of the control device via Modbus TCP/IP from version 1.3 onwards").

2.5 Necessary settings on the controller

First switch on the controller and wait until it is fully operational. You can then open the service menu on the user interface by long-clicking on the "Overview of fault messages" icon at the bottom left.



Figure 2-2: Overview with button for the settings menu

Here you switch to the "Connectivity" menu.



Figure 2-3: Settings menu

You must then make some settings for the Modbus TCP/IP communication on the controller. In particular, pay attention to the settings shown in bold, which may differ from the factory settings in your controller:

Surname	Parameter to choose	Meaning
Local Area Network	Modbus TCP/IP (factory setting: Internet)	The controller activates the Modbus TCP/IP functionality, which is necessary for data transmission.
Communication Modbus	With Heart Beat / Without Heart Beat (Factory setting: No)	The controller uses a Modbus TCP/IP connection with/without heartbeat. For more information, please refer to the " Modbus TCP/IP Heart Beat" chapter. Note : If "No" is selected, communication via Modbus is deactivated and not possible!
timed out	120 to 600 seconds (factory setting: 180 seconds)	Sets the Modbus TCP/IP heartbeat timeout. If the time is exceeded before a new counter value is received, an error message is generated.
Modbus Unit ID	0 to 255 (factory setting: 255)	Specifies the Modbus device identifier of the controller. This setting must be coordinated with the BMS to enable communication.
Allow write access Up to firmware 1.5, the parameter is called " Write access via Modbus TCP/IP" Up to firmware 1.2.7 the parameter is called "Allow write access"	On (factory setting: On)	Enables write access via Modbus TCP/IP and is therefore a prerequisite for external setpoint specification via Modbus TCP/IP.
Provide extended monitor data of the plant Up to firmware 1.5, the parameter is called " Preparing the data via Modbus TCP/IP" Up to firmware 1.2.7, the parameter is called "Providing the data from the substations"	On (Factory setting: Off)	Enables the provision of data from internal components of the control as well as from substations connected via Modbus TCP/IP. Note : For communication via Modbus TCP/IP, this parameter must always be activated!
Heat request only via Modbus (Only valid up to firmware 1.2.7)	On (factory setting: off)	If activated, a heat request can only be received via Modbus TCP/IP. Heat requirements that may be generated by built-in function modules are not taken into account. For more information, please refer to the " Information on parameters 'Heat request via Modbus' and 'Internal heat request '" chapter. Note : This parameter is only valid up to firmware 1.2.7.

Surname	Parameter to choose	Meaning
address mapping	Static / DHCP	<p>If you do not have a DHCP server in the machine network, select static for the address assignment and then set the IP address, the network mask and the gateway - otherwise you can select DHCP.</p> <p>Note : Restriction on the IP address is that you must not use the IP address range 172.31.42.X as it is used for internal purposes of the regulation.</p>
Enable Modbus compatibility for firmware lower than 1.3.6	Off (Factory setting: Off)	<p>Enables or disables Modbus compatibility mode. Please only use this mode if you are already controlling a controller up to version 1.2.7 via Modbus TCP/IP through your BMS. If the Modbus compatibility mode has been activated, you can continue to use the old Modbus data points / control without making any changes to your existing solution.</p>

Once you have made all the settings regarding Modbus TCP/IP communication, you can apply the settings using the “Save” button. Any changes made are discarded using the “Cancel” button. Once you have saved the changed connectivity settings, the controller is ready for communication via Modbus TCP/IP.

Notes :

- To determine which components can generate a heat request on the control device, please continue reading for the necessary configuration in the chapter " Information on parameters 'Heat request via Modbus' and 'Internal heat request'".
- Up to version 1.2.7, this setting was controlled via the "Heat request only via Modbus" parameter (see chapter " Information on the 'Heat request only via Modbus' parameter and chapter " Outdated software versions and their special features").

2.6 Network behavior and diagnostic options of the controller

When using Modbus TCP/IP, the Modbus Device ID (Unit Identifier) of the control device is factory set to 255. This can be changed in the control device from version 1.5.13 (see chapter “ Necessary settings on the controller”). In addition, when using Modbus TCP/IP, any values are transmitted in the byte order Big-Endian (Most Significant Bit (MSB) First) also known as "order3210" or "orderDCBA" (see chapter " Data types and register width " data types and register width.

Up to software version 1.5, it was not possible to address the controller with the network diagnosis tool "Ping". This has been changed from software version 1.5: Activate this function by setting the "Local Area Network" parameter to "Modbus TCP/IP" (see chapter " Necessary settings on the controller"). The control device

thus responds to the ping request and you can see that the control device can be reached in the network.

2.7 data types and register width

The controller supports the following data types, which must be used via Modbus TCP:

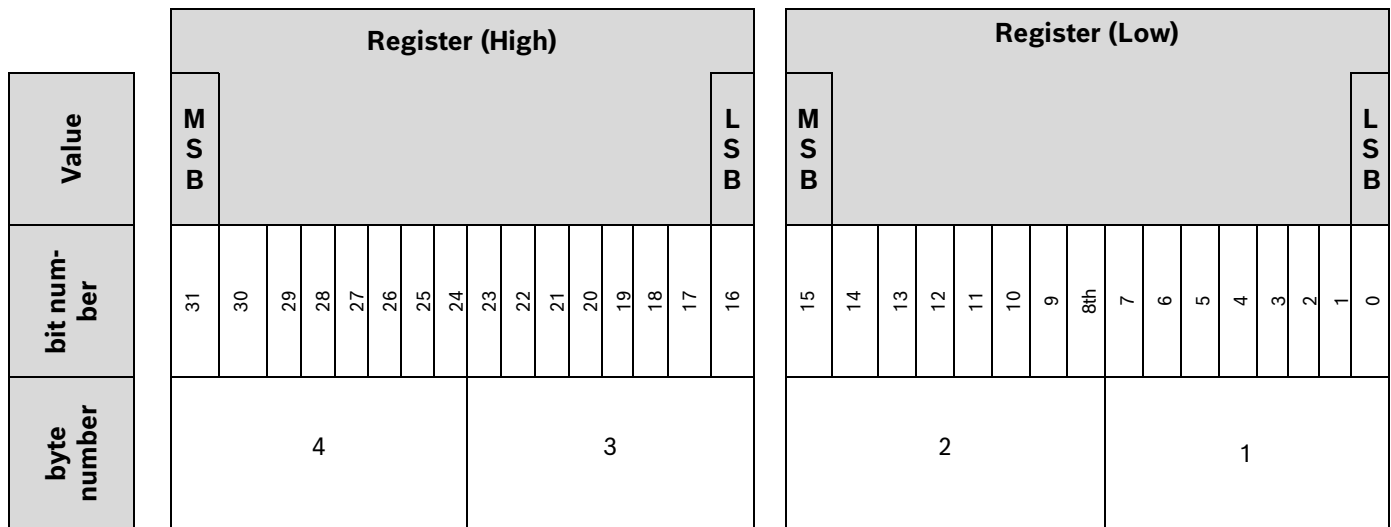
data type	Bits	bytes	range of values
bit (bit)	1	0	0 to 1
Integer (int)	16	2	0 to 65535
Signed Integer (signed int)	16	2	-32768 to +32767
Signed Long (long)	32	4	-2147483648 to +2147483647
Unsigned Long (unsigned long)	32	4	0 to +4294967295

The Modbus registers each have a width of 16 bits. The registers are structured as follows:

	register															
Value	MSB														LSB	
Bit number	15	14	13	12	11	10	9	8th	7	6	5	4	3	2	1	0
byte number	2								1							

Notes :

- If you read out a register with the "Long" data type, you must ensure that the value is divided between two consecutive registers, since the memory of just one register would not be sufficient to map such a large number. In this case, the register to be read looks like this:



2.8 Control of the control device via Modbus TCP/IP from version 1.3 onwards

Please use the following control of the control device from firmware “1.3”.

register type	address	bit	Data point name	data type	Value range	Meaning
Holding registers	0		Heart Beat (in)	internal	0 to 65535	If the Modbus connection is to be monitored for connection aborts or timeouts (Modbus communication with heartbeat), the counter must be written to this register, which is unequal to the old value that can be read via the heartbeat (out) register.
holding register	1		Heartbeat (out)	internal	0 to 65535	If this register is read, the current counter value of the heartbeat (in) can be retrieved (to verify the sent counter value on the GLT side).
holding register	400		Setpoint system flow temperature in °C	internal	0 to 120	Target value of the currently required system temperature. Note : The maximum system temperature depends on the boiler types installed and their performance.
holding register	401		Setpoint system performance in %	internal	0 to 100	Setpoint of the currently required system performance.
Holding registers	402		Operating mode	internal	0 (Off) or 1 (operation)	Operating mode of the heat generators. Off = Heat generators are off, consumers continue to be controlled Operation = Normal operation via heat request.

register type	address	bit	Data point name	data type	Value range	Meaning
Holding registers	405		Bit block for cascade control	bit		See table entries below
holding register	405	0	lead boiler	bit	0 (inactive) or 1 (active)	From software version 1.3.3 or higher, this setting is mandatory. In a system with only one boiler, this is identified as the lead boiler. In a system with FM-CM, the lead boiler function must be enabled. This depends on the sequence changeover set in the control system. In the case of a boiler that is active as a lead boiler, the run-on times of the boiler circuit pump are adjusted, among other things.
holding register	405	1	prioritization	bit	0 (inactive) or 1 (active)	Allows prioritization to be assigned to the heat generators. An attempt is made to reach the desired setpoint as quickly as possible. This is used, for example, for faster hot water demand. Note : This functionality is available from software version 1.3.3 and higher .
holding register	405	2	Temperature-controlled control	bit	0 (inactive) or 1 (active)	Activates or deactivates the temperature-controlled control of the heat generators.
holding register	405	3	Performance-based control	bit	0 (inactive) or 1 (active)	Activates or deactivates the power-controlled control of the heat generators.

Notes :

- Data type integer (int) corresponds to 16 bits (bit₀ to bit₁₅), see chapter " data types and register width".
- The addresses mentioned here refer to the fact that the controller is configured as a master. If you are using the control device as a slave, please continue reading in the chapter " Control via Modbus TCP/IP in the controller network".
- To set individual bits in a Modbus register (as required for holding register 405), please continue reading in the chapter " Setting or reading individual bits in a Modbus register".
- The data point "Setpoint system flow temperature in °C" affects the strategy in the controller. The specific structure of the system or module configuration in the control unit decides which heat generators are requested by the control unit.

- Power-based control can only be used if one or more boilers of the same type are connected to the control unit. This function is not possible with several different heat generators.

2.8.1 Temperature-controlled heat request via Modbus TCP/IP from and including version 1.3

In order to be able to send a temperature-controlled heat request via Modbus, you must first configure the controller accordingly. Please note the information in the "Information on parameters 'Heat request via Modbus' and 'Internal heat request'" section if you want to control the heat request exclusively via the GLT.

If you now want to generate a temperature-controlled heat request, you must write to the following registers:

register	value for registers
Operating mode (holding register 402)	1
Flow temperature setpoint (holding register 400)	X in °C
Strategy bit block holding register 405	7

The sent value "7" in "Strategy Bit Block" corresponds to the following bit settings:

register	value for registers
Lead boiler (holding register 405; bit ₀)	1
Prioritization (holding register 405; bit ₁)	1
Temperature-controlled regulation active (holding register 405; bit ₂)	1
Power-based control active (holding register 405; bit ₃)	0

Hints:

- Using the values from the example above, a temperature-controlled heat request with X °C is placed on a boiler, which is to work as a lead boiler. In a system with FM-CM, the lead boiler function must be activated.
- X in °C corresponds to your desired flow temperature.
- You can check the successful transfer of the registers in the controller. To do this, navigate to the " *Service menu > Monitor data (camera symbol) > Heat generation > Strategy data > Request* " menu. Here you will find the values

of the registers and can check whether the data has been transferred accordingly. The setpoints and control mode will only display the sent value if the operating mode and heat request via bus are set.

- You can read about setting individual bits as required for "Bit block strategy" in the "Setting individual bits in the Modbus register" chapter.

2.8.2 Power-based heat request via Modbus TCP/IP from and including version 1.3

In order to be able to send a power request via Modbus, you must first configure the control device accordingly (see chapter "Information on parameters 'Heat request via Modbus' and 'Internal heat request'").

You can then describe the following Modbus registers:

register	Value for register
Operating mode (Holding Register 402)	1
Setpoint power (Holding Register 401)	X in %
Strategy Bitblock Holding Register 405	9

The sent value "9" in "Strategy Bit Block" corresponds to the following bit settings:

register	value for registers
Lead boiler (holding register 405; bit 0)	1
Prioritization (holding register 405; bit 1)	0
Temperature-controlled control (holding register 405; bit 2)	0
Power-based control (holding register 405; bit 3)	1

Note e :

- With the values from the above example, an output-based heat request with X% is sent to a boiler that is to work as a lead boiler. In a system with FM-CM, the lead boiler function must be activated.
- You can only use the power-based control if you have deactivated the internal heat request.
- X in % corresponds to your desired performance.
- You can check the successful transfer of the registers in the controller. To do this, navigate to the "Service menu > Monitor data (camera symbol) > Heat

generation > Strategy data > Request " menu. Here you will find the values of the registers and can check whether the data has been transferred accordingly. The setpoints and control mode will only display the sent value if the operating mode and heat request via bus are set.

- You can read about setting individual bits, as required for "Bit block strategy", in the "Setting individual bits in the Modbus register" chapter.

2.8.3 Information on parameters 'Heat request via Modbus' and 'Internal heat request'

You can use the following parameters of the control device to determine how a heat or power request can be generated on the control device. The parameters for configuring the behavior can be found in the service menu > heat generation > strategy data > basic settings. Depending on the parameterization, this can have an impact on the control.

If you have additional function modules (e.g. for a heating circuit) or substations connected to the control, you can activate the heat request for these function modules using the "Internal heat request" parameter.

If you want to generate a heat request via Modbus, you must activate the "Heat request via Modbus" parameter. Depending on which heat requirements you have activated, the behavior of the control is influenced as follows:

"Internal heat requirement" parameter	"Heat request via Modbus" parameter	impact
Out of	Out of	No heat requirements are taken into account by the control.
At	Out of	Only heat requirements from internal functional modules are taken into account.
Out of	At	Only heat requests via Modbus are taken into account (exclusive mode).
At	At	Internal and Modbus heat requirements are taken into account (parallel mode).

2.8.3.1 Heat request exclusive mode

In heat request exclusive mode, any heat requests from internal function modules are ignored. This means that you have full control of any heat requirements generated by the control system. In this mode you can use a heat request via a temperature-controlled as well as a power-controlled heat request.

2.8.3.2 Parallel mode of heat demand

In the parallel mode of the heat request, both heat requests from internal function modules and heat requests via Modbus are taken into account. A heat request via Modbus, which comes from the BMS, can only add a heat request. This does not affect existing heat requirements from other modules. The setpoint is generally the highest heat requirement. In this mode you cannot use power controlled heat demand.

2.9 Control via Modbus TCP/IP in a network

With the control units of the Logamatic 5000/Control 8000 control unit series, it is possible to operate additional control units or components from the Buderus Logaflow HSM plus modular system in a network. A hierarchical structure is created in which there can be a head station (master) and several substations (slaves). In addition, with Logaflow HSM plus it is possible to create so-called segment substations. A segment substation is a substation that can itself include other substations in the network.

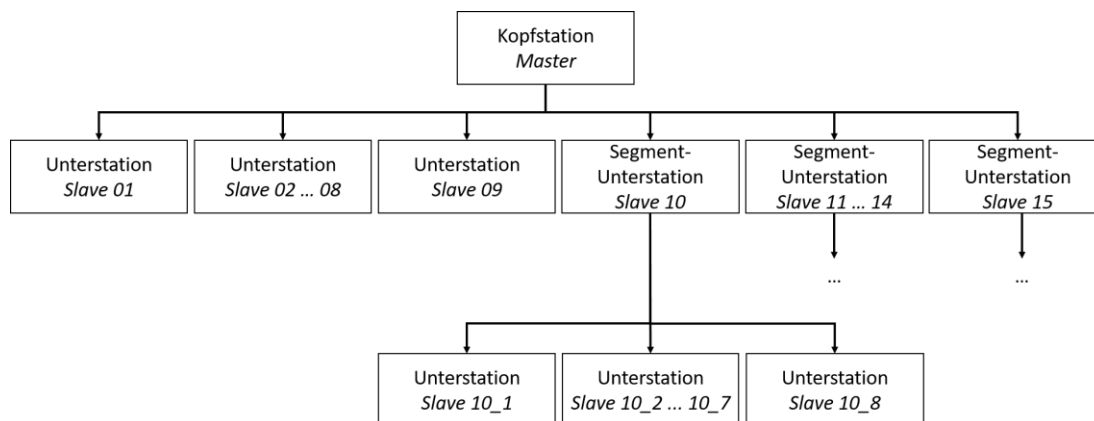


Figure 2-4: Functional principle

2.9.1 Control via Modbus TCP/IP in the controller network

If you operate the control device in a control device network, i.e. you have a master control device and at least one slave control device connected to each other, you can access the data points of the slave control device via the master control device. The behavior of the control device, whether it should function as master or slave, is set using the rotary coding switch on the control device (see documentation “Service instructions for the specialist – Logamatic 5311/Control 8311” or “Service instructions for the specialist – Logamatic 5313/Control 8311”).

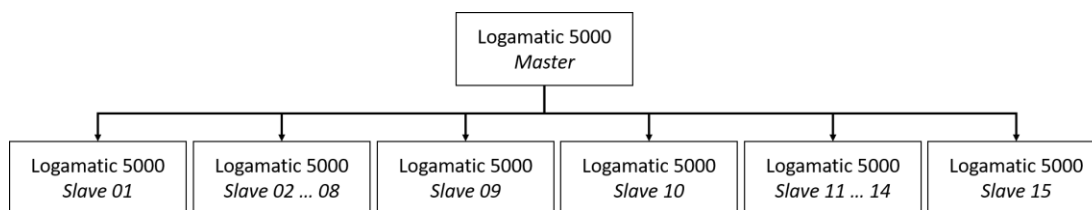


Figure 2-5: Overview of control device network with Logamatic 5000/Control 8000

The data points between a master and a slave controller only differ in their Modbus address for addresses less than 8000. As soon as the controller acts as a slave, you can access the data points via the master controller using the following formula:

$$\text{ModbusAdresseSlave} = \text{ModbusAdresseMaster} + (\text{Drehcodierschalter} * 500)$$

Notes :

- The addresses already calculated can be found in the data point list (sheets Slave01 – Slave15).
- The formula can only be used for data points with addresses less than 8000. Data points from address 8000 onwards are only available in the master.

Example:

Assuming you have two controllers (1x master and 1x slave) in use and set up a controller network. The rotary coding switch on the master controller is set to position 0 (= master) and on the slave controller to position 1 (= slave 01). You have installed and configured a heating circuit module in slot 1 on the slave controller. For example, if you now want to access the current flow temperature of heating circuit 1 from the slave controller, you can calculate the address as follows:

$$\text{ModbusAdresseSlave} = 102 + (1 * 500) = 602$$

- The Modbus address for the current flow temperature on heating circuit 1 is on the master controller on input register 102.
- The position of the rotary coding switch on the slave controller is set to 1. The position is multiplied by 500 and added to the master Modbus address of the corresponding data point.
- The address calculated for querying the current flow temperature of heating circuit 1 from the slave controller via the master controller is on input register 602.

If you now request the calculated address (input register 602) on the master control device, you will receive the current flow temperature of heating circuit 1 from the slave control device.

Notes :

- You will also find the calculated addresses of all positions of the rotary coding switch (slave 01-15 corresponds to the position of rotary coding switch 1-15) in the Modbus TCP/IP data point list.
- If you have connected one boiler to the master and one to the slave controller, you need a cascade module on the master controller to control both boilers via Modbus TCP/IP. If you have not installed a cascade module on the

master controller and set up a controller network, the boiler on the slave controller will be blocked by the master controller. If you want to do without the cascade module, you must implement the cascade functionality in your building management system. In this case, you must use both controllers as masters (no controller network may be created).

2.9.2 Control via Modbus TCP/IP in combination with Logaflow HSM plus

You can operate the control unit in combination with the Buderus Logaflow HSM plus modular system. For this you need a connection between the master controller and at least one HSM plus device. You can then access the data points of the HSM plus via the master controller. The behavior of the controller, whether it is to act as a master or slave, is set using the rotary coding switch on the controller. Further information on this can be found in the documentation "Service instructions for professionals - Logamatic 5311/Control 8311" or "Service instructions for professionals - Logamatic 5313/Control 8313".

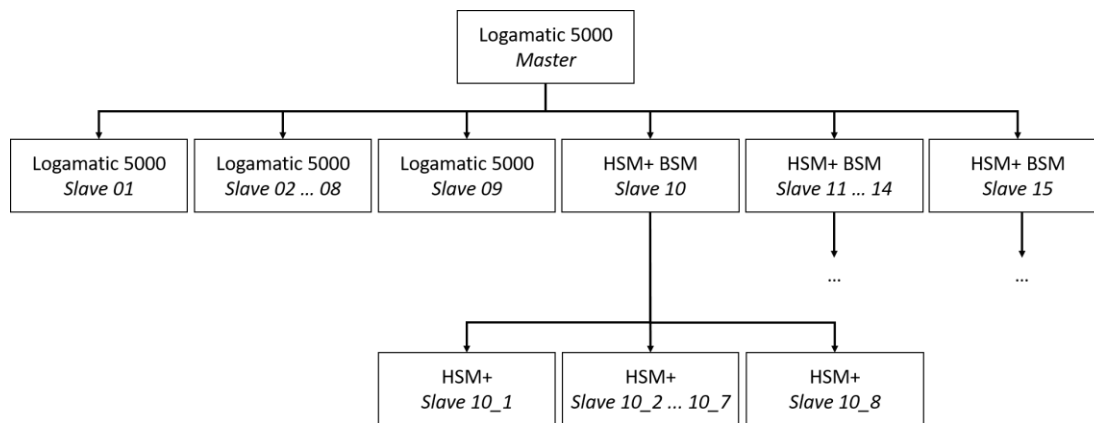


Figure 2-6: Overview of connection with Logamatic 5000/Control 8000 and Logaflow HSM plus

Positions Slave10 to Slave 15 are provided for the connection between Logamatic 5000/Control 8000 and Logaflow HSM plus. Up to six HSM plus BSM can be connected to a Logamatic 5000/Control 8000 (master). Each HSM plus BSM can be used as a segment substation. This means that up to eight additional HSM plus devices can be connected to each HSM plus BSM.

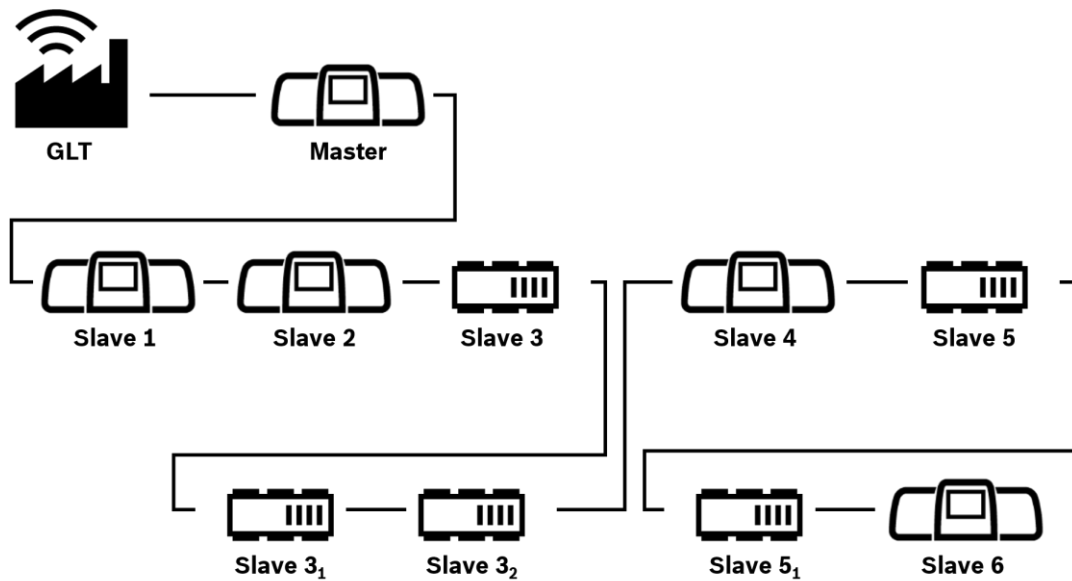


Figure 2-7: Exemplary installation of a network

Hints:

- Another controller from the Logamatic 5000/Control 8000 controller series is always integrated as a substation (slave). It cannot be used as a segment substation (see Figure 2-7).

Example:

If you connect an HSM plus BSM to a Logamatic 5000/Control 8000, it will automatically be integrated into the Logamatic 5000/Control 8000 as Slave10. A possible second, third, ... HSM plus BSM is integrated accordingly as Slave11, Slave12,

2.9.2.1 Access to substation

All data points can be accessed via the master via Modbus TCP. Since the addresses of the data points can differ depending on the system structure, you can use the following formulas to calculate the corresponding addresses:

First you have to calculate the required offset for the respective slave (10 – 15):

$$\text{OffsetSlaveX} = X * 500$$

You can now add this offset to the addresses of the HSM plus BSM. The result is the address of the data point with which you can access the respective BSM via the master.

$$\text{AdresseViaMaster} = \text{OffsetSlaveX} + \text{Adresse}$$

Example:

You want to get the address “Input Register 1 – Own Flow Temperature” from Slave11 via Master.

$$\begin{aligned} \text{OffsetSlave11} &= 11 * 500 = 5500 \\ \text{AdresseViaMaster} &= 5500 + 1 = 5501 \end{aligned}$$

2.9.2.2 Access via segment substation to substation

You can also access the substations of an HSM plus BSM (Slave10 – Slave15) via the master. For this you must also first calculate the offset and then add this to the address.

$$\begin{aligned} \text{OffsetSlaveX} &= (X - 10) * 8 * 500 \\ \text{OffsetSubSlaveY} &= (Y - 1) * 500 \\ \text{OffsetSubSlaveX}_Y &= 10000 + \text{OffsetSlaveX} + \text{OffsetSubSlaveY} \end{aligned}$$

You can now also add this offset to the addresses of the respective HSM plus device. The result is the address of the data point with which you can access the respective HSM plus device via the master.

$$\text{AdresseViaMaster} = \text{OffsetSubSlaveX}_Y + \text{Adresse}$$

Example:

You want to call up the address "Input Register 34 - HMI State" via Slave10 of substation 1 via the master:

$$\begin{aligned} \text{OffsetSlave10} &= (10 - 10) * 8 * 500 = 0 \\ \text{OffsetSubSlave1} &= (1 - 1) * 500 = 0 \\ \text{OffsetSubSlave11}_5 &= 10000 + 0 + 0 = 10000 \\ \text{AdresseViaMaster} &= 10000 + 34 = 10034 \end{aligned}$$

You can access the HMI state of substation 1 of segment substation 10 via the head station via the address "Input Register 10034".

Example:

You want to call up the address "Input Register 34 - HMI State" from substation 8 via Slave12 via the master:

$$\begin{aligned} \text{OffsetSlave12} &= (12 - 10) * 8 * 500 = 8000 \\ \text{OffsetSubSlave8} &= (8 - 1) * 500 = 3500 \\ \text{OffsetSubSlave12}_8 &= 10000 + 8000 + 3500 = 21500 \\ \text{AdresseViaMaster} &= 21500 + 34 = 21534 \end{aligned}$$

You can access the HMI state of substation 8 of segment substation 12 via the head station via the address "Input Register 21534".

3. General information

The following chapters contain general information on handling the controller. In addition to the message concept and an explanation for the use of the heartbeat, the use of the Modbus address scheme and the evaluation of bits and ASCII characters are also discussed.

3.1 Message concept of the control device

The control device is designed to display pending messages (notices or faults) to the user. Particularly when using the control device with a BMS, it is important to correctly interpret pending messages and, if necessary, to be able to initiate necessary measures.

To simplify the display of the messages, so-called “HMI statuses” were added to all subcomponents of the control device. With the help of these HMI statuses, a possible cause of the error can be narrowed down and the error weighting can be identified. Please continue reading the chapter “General information about the HMI status of the controller” for a detailed explanation of HMI status. An HMI status can also be used to identify any boiler malfunctions (see chapter “General information on boiler malfunctions”).

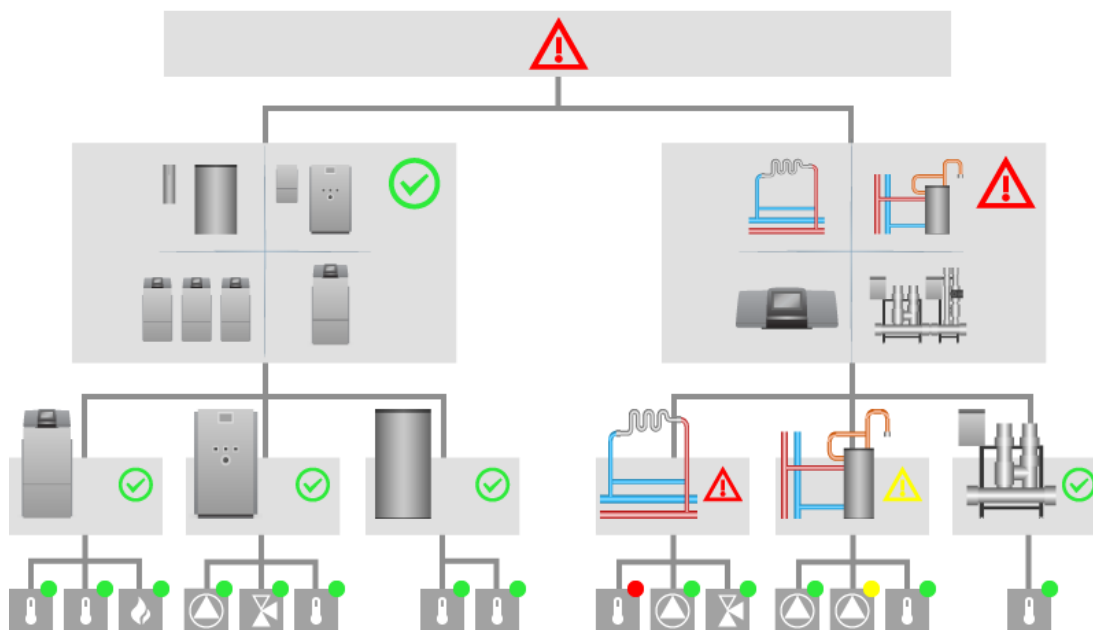


Figure 3-1: Message concept of the control device

Since the HMI status can only be used for a rough assessment of the error and more detailed information is often necessary to initiate an appropriate measure, there are additional Modbus TCP registers that make the pending message identifiable via a message key. To do this, please continue reading the chapter “General information on evaluating current messages on the control device”.

Note e :











© All rights reserved by Robert Bosch GmbH, also in the case of intellectual property rights registrations

- If the connected boiler is in a fault or in manual mode, no heat requests are passed on from the control device to the boiler. As soon as the boiler is ready for operation again, the boiler is controlled according to the specifications.

3.1.1 General information about the HMI status of the controller

An HMI status can be used to determine the current status of the controller and its components. Each software component in the controller has such an HMI status. This shows the current status of the respective component via five possible states. As soon as one of the software components detects an irregularity in regular operation, the associated HMI status is set. The higher-level software component recognizes the HMI status, supplements it if necessary and also forwards it to its higher-level software component until the respective status has arrived at the highest software component level. The principle of passing on the HMI status allows the software component that originally influenced the HMI status to be reliably identified and the exact cause of the error to be analyzed.

There are five HMI statuses available within the control device, which use color assignment:







HMI Status	Color	Meaning
OK 	Green Blue 	The system/component is working properly.
warning 	Yellow 	A deviation from the normal state of the system/component was detected, e.g. manual operation was activated or the boiler has an excessively high flue gas temperature.
Critical 	Orange 	The system/component can currently still be operated, but immediate intervention is required.
Mistake 	Red 	The system / component has a fault. An intervention is required.
Unknown 	Gray 	The status of the system/component cannot be determined.

Note e :

- Only the colors green/blue, yellow and red are displayed on the hardware LED on the controller.

3.1.2 General information on boiler malfunctions

In addition to the HMI status of individual software components of the control device, the heat generator offers additional information on operating messages. The message classes of the boiler affect the HMI status of the associated components in the controller. There are three different message classes:

boiler error	HMI Status	Meaning
maintenance notice 	Warning (Yellow) 	Signaling of an upcoming maintenance measure.
Blocking disorder 	Warning (Yellow) 	Fault that leads to a temporary shutdown of a system (e.g. a heating system), which restarts automatically as soon as the fault no longer exists.
Locking fault 	Error (Red) 	Malfunction that leads to a system (e.g. a heating system) being switched off. The error must be acknowledged on the boiler before the system can resume work.

3.1.3 General information on evaluating current messages on the control device

For a quick overview of the system, it is recommended to use the HMI status of the individual areas. If more precise message information is required, you can read the message registers.

Some reports contain additional information, which is only necessary in individual cases. If you need this additional information for a detailed error description, you can get in touch with your Buderus/Bosch contact.

The values can be interpreted using the document “Message key list of the Logamatic 5000/Control 8000 control device series”. The “Key” field can be used to receive the corresponding error message. This document is in “.csv” file format. This makes it possible to automatically read out or process the message code. You can obtain the document from your Buderus contact person.

The following registers contain the message keys:

register type	address	Area	Surname	data type
input	476-477	system	Error register 1	Unsigned Long
input	478 - 479	system	error register 2	Unsigned Long
input	480-481	system	Error register 3	Unsigned Long
input	482-483	system	Error register 4	Unsigned Long
input	266-267	system	Error register 5	Unsigned Long
input	268-269	system	Error register 6	Unsigned Long
input	270-271	system	Error register 7	Unsigned Long
input	272-273	system	Error register 8	Unsigned Long
input	274-275	system	Error register 9	Unsigned Long
input	276-277	system	Error register 10	Unsigned Long

A message remains in the register until the message is resolved in the controller.

You can find an example of the procedure in the chapter “ Example for reading a message key”.

Notes :

- Since this is the “Unsigned Long” data type, two Modbus registers must be queried in order to obtain a message key.

3.1.4 Example for reading a message key

There is a message in the control device. You now want to read out error register 1. To do this, proceed as follows:

1. Read register 476 using Modbus function code 04 (Read Input Register) of the type “unsigned long” . In this example, you get the decimal number 2953134087.
2. Look for the relevant translation in the "Message key list for the Logamatic 5000/Control 8000 control unit series" file. You will receive the message key "Standard heat generator - external fault input pump boiler".

Depending on your building management system, an "unsigned long" can be misinterpreted as a "signed long" (you get a negative number from the register). In this case you can do the following:

1. Read register 476 using Modbus function code 04 (Read Input Register) of type “unsigned int” . In this example, you get the decimal number 45061. Then convert the value to binary. You get the binary number 1011 0000 0000 0101b .
2. Read register 477 using Modbus function code 04 (Read Input Register) of type “unsigned int”. In this example, you get the decimal number 16391.

Then convert the value to binary. You get the binary number 0100 0000 0000 0111 b.

3. Now put both binary values together (bit shift left and logical or) and convert them to decimal system:
 $(1011\ 0000\ 0000\ 0101b \ll 16) \vee 0100\ 0000\ 0000\ 0111b =$
 $1011\ 0000\ 0000\ 0101\ 0100\ 0000\ 0000\ 0111b =$
 2953134087
4. Find the corresponding translation for the message code 2953134087 in the file "Message key list of the Logamatic 5000/Control 8000 control device series". You will receive the message code "Standard heat generator – external fault input for boiler pump".

3.2 Modbus TCP/IP Heart Beat

Communication is monitored via the heart beat using a counter. If the counter is not changed by the communication partner (in this case the BMS) within a certain period of time, an interruption in communication is detected by the control system. In this case, the error message "Connection to building management system faulty" is displayed in the "Overview of fault messages" menu.

3.2.1 Variant 1: With Heart Beat

If you operate the "Communication Modbus" parameter with the setting "with Heart Beat", the GLT must monitor the communication via a counter. In doing so, the BMS must write this counter reading to the Modbus register "Heart Beat (in)" of the control device.

To do this, the BMS must send a value to the "Heart Beat (in)" register. The "Heart Beat (out)" register always reflects the current value from the "Heart Beat (in)" register. The value can then be read again from the BMS on the "Heart Beat (Out)" register. The GLT then increments (increases by 1) the value and checks whether this changed value is also available on the "Heart Beat (Out)" tab. If not, the connection between the building management system and the controller is faulty. This process is carried out continuously and the connection is therefore monitored. The value in the "Heart Beat (in)" register should be reset in good time before an overflow occurs.

As soon as the "Heart Beat (In)" register is written, the timeout of the heart beat in the controller is reset. The timeout can be set via the control's HMI.

3.2.2 Variant 2: Without heart beat

If you do not have the option to implement the heart beat on the GLT side, you can also use the "no heart beat" setting. The control device assumes that the connection to the GLT is constant. In this case, detecting a communication breakdown is not available.

3.3 Conversion of the Modbus address scheme

If you would like to use the Modbus Modicon address scheme instead of the decimal address scheme, you can use the following formula to convert the addresses:

$$\text{ModiconAdresse} = \text{RegisterTyp} + \text{DezimalAdresse} + 1$$

For the “Holding Register” register type, use the number 40000. For the “Input Register” register type, use the number 30000.

Example :

For example, you want to address the holding register data point with decimal address 400 “Setpoint system flow temperature in °C” in the Modicon scheme:

1. You determine the value of the register type:

$$\text{RegisterTyp} = \text{Holding – Register} = 40000$$

2. You calculate the address for the Modicon scheme:

$$\text{ModiconAdresse} = 40000 + 400 + 1 = 40401$$

So you can use address 40401 if you want to use the Modicon schema to describe the “ Setpoint system flow temperature in °C ” data point.

Notes :

- The decision as to whether the decimal or Modicon scheme must be used depends on the software used and the configuration of the BMS. Please speak to your BMS manufacturer for further information on the correct scheme.

3.4 Setting or reading individual bits in a Modbus register

If you are unable to set individual bits in a register, you can alternatively enter an integer that corresponds to the desired bit pattern in the Modbus register. You can also read out a complete register as an integer and interpret it as a bit pattern.

A Modbus register (2 bytes) consists of 16 bits (bit₀ to bit₁₅). Big Endian (Most Significant Bit First (MSB)) is used as the byte order. If you now want to set individual bits in a Modbus register, you can determine the required integer (≙ bit pattern) using the following table:

bit number	bit 15	bit x ...	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Value	32768	...	128	64	32	16	8th	4	2	1
Bit to be set										

Notes :

- For reasons of presentation, the values of all bits were omitted. The value of a bit can be determined using the following formula:

$$\text{Wertigkeit} = 2^{\text{Bitnummer}}$$

Example of calculating the weight

You want to determine the significance of bit₁₅:

$$\text{Wertigkeit Bit}_{15} = 2^{15} = 32768$$

- The following examples use the Modbus data point list from version 1.3. However, the procedure described can be transferred to any Modbus register and thus also to the old data point list.
- The conversion from integers to bit patterns and vice versa can also be calculated using a pocket calculator with programming mode.
- Bits can also be set and read using corresponding bit operations.
- Numbers with a _b are in binary format and numbers with a _d are in decimal format.

3.4.1 Setting individual bits in the Modbus register

You want to use temperature-controlled heat request via Modbus TCP/IP. To do this, you must activate bit₂ in holding register 405. In this case the table would look like this:

Bit number	Bit ₇	Bit ₆	Bit ₅	Bit ₄	Bit ₃	Bit ₂	Bit ₁	Bit ₀
Value	128	64	32	16	8th	4	2	1
Bit to be set						X		

Calculated integer or bit pattern:

$$\begin{aligned}\text{Ganzzahl} &= \text{Wertigkeit Bit}_2 = 4 \\ \text{Bitmuster} &= 0000000000000100_b\end{aligned}$$

However, if you would like to set several bits (e.g. temperature-controlled [Bit₂], priority [Bit₁] and lead boiler [Bit₀]), the table and the corresponding calculation would look like this:

bit number	bit ₇	Bit ₆	Bit ₅	Bit ₄	Bit ₃	Bit ₂	Bit ₁	Bit ₀
Value	128	64	32	16	8th	4	2	1
Bit to be set						X	X	X

Calculated integer or bit pattern:

$$\begin{aligned}\text{Ganzzahl} &= \text{Wertigkeit Bit}_2 + \text{Wertigkeit Bit}_1 + \text{Wertigkeit Bit}_0 = 4 + 2 + 1 = 7 \\ \text{Bitmuster} &= 0000000000000111_b\end{aligned}$$

3.4.2 Reading individual bits from Modbus registers

If you would like to read out individual bits from a Modbus register, for example to read out the currently pending errors in the safety chain, proceed as follows:

First you read out the contents of the register as an integer. You must then transfer the read integer to a bit pattern. To transfer the integer to a bit pattern, please proceed as follows:

Now divide the read integer by the value, starting with the highest bit (division with remainder). If the quotient is zero, you can divide the integer by the significance of the next smaller bit. As soon as the quotient is not equal to zero, the current bit is set and you can use the remainder of the division as a dividend for the calculation with the significance of the next smaller bit. You apply this scheme until you have divided the integer or the remainder by all the bits. The result gives the corresponding bit pattern.

You can use the following formula for this (division with remainder):

$$Bit_{NummerX} = \frac{[Ganzzahl\ oder\ Rest]}{Wertigkeit\ Bit_{NummerX}}$$

example :

For example, if you want to determine the current errors in the safety chain, you would use input register 225 for this. Suppose you read the integer 35 from the register. For this example, the remainder division would look like this:

Bit ₁₅	= 35 / Value Bit ₁₅	= 35 / 32768	= 0
Bit ₁₄	= ...		
...	= ...		
Bit ₅	= 35 / value bit ₅	= 35 / 32	= 1 remainder 3
Bit ₄	= 3 / Value Bit ₄	= 3 / 16	= 0
Bit ₃	= 3 / Value Bit ₃	= 3 / 8	= 0
Bit ₂	= 3 / Value Bit ₂	= 3 / 4	= 0
Bit ₁	= 3 / Value Bit ₁	= 3 / 2	= 1 Rest 1
Bit ₀	= 1 / Value Bit ₀	= 1 / 1	= 1

The result of the integer 35 is the following bit pattern: 0000000000100011_b

3.5 Converting values to ASCII characters

if you would like to convert the error display code in the display (= display code) into ASCII characters, please proceed as follows:

1. Read the integer value from the register.
2. Convert the integer into binary format as described in the chapter “Setting or reading individual bits in a Modbus register”.

3. Now split the binary format into the individual bytes (= 8 bits). Each byte corresponds to a letter.
4. Now convert each byte back to an integer.
5. Look in an ASCII table to see ³which letter is represented by each integer.

example :

You want to know the display code of boiler 1. To do this, read the integer from the input register 8019. In the following example the number 12880 is read.

1. Convert the integer to binary:
0011001001010000_b
2. Now divide the binary number into bytes:
00110010_b, 01010000_b
3. Now convert the individual bytes into decimal:
00110010_b = 50_d
01010000_b = 80_d
4. According to the ASCII table, the numbers can now be converted into characters:
50_d = "2"
80_d = "P"

If you read a 12880 from input register 8019, this corresponds to the display code "2P".

Hints:

- If you also read a 564 from the input register 8018 (error code), for example, this corresponds to the error message "Boiler sensor temperature rises too quickly (>70K/min)". You can obtain additional information about the error and display codes from your Buderus/Bosch contact person.
- In addition to faults, operating codes are also displayed in the "Error Code" register (input register 8018).

³ https://de.wikipedia.org/wiki/American_Standard_Code_for_Information_Interchange#ASCII_table

4. Outdated software versions and their special features

The control device is gradually being further developed. Here, functions are added or changed in order to be able to provide more security, comfort and expanded options in future versions. It is therefore always advisable to update your controller to the currently available version (see chapter "Checking the current firmware of the control device").

If, however, an older software version is available on your controller, you will find information about the special features to be observed in the following chapters.

4.1 Message behavior up to controller version 1.8

Up to and including control device version 1.8, up to four currently pending messages from the control device can be read out via Modbus TCP/IP. To evaluate and retrieve these messages, the following registers have been added to the data point list:

Register type	Addresses in the master ⁴	Area	Designation	data type
input	476-477	system	Error register 1	Unsigned Long
input	478-478	system	error register 2	Unsigned Long
input	480-481	system	Error register 3	Unsigned Long
input	482-483	system	Error register 4	Unsigned Long

Please note that the latest message is always written to error register 1. Existing messages are then moved to the next error register.

You can find an example of the procedure in the chapter "Example for reading a message key".

Notes :

- Since this is the "Unsigned Long" data type, two Modbus registers must be queried in order to obtain a message key.

⁴The addresses of the possible additional control devices can be found in the data point list

4.2 Control of the control device via Modbus TCP/IP up to and including version 1.2.7

Please use the following activation of the controller up to software version "1.2.7".

Register type	address	Data point name	data type	Value range	Meaning
Holding registers	0	Heart Beat (in)	internal	0 to 65535	If the Modbus connection is to be monitored for connection aborts or timeouts (Modbus communication with heartbeat), the counter must be written to this register, which is unequal to the old value that can be read via the heartbeat (out) register.
holding register	1	Heartbeat (out)	internal	0 to 65535	If this register is read, the current counter value of the heartbeat (in) can be retrieved (to verify the sent counter value on the GLT side).
holding register	35	Operating mode	internal	0 = off, 1 = keep warm, 2 = run	Boiler request mode. 0 = Off = Boiler is off, consumers continue to be controlled 1 = warming = warming function for industrial cascade applications with separate warming pump (currently not relevant for Logamatic 5000/Control 8000) 2 = operation = normal operation via boiler heat request
holding register	2	System temperature Temperature specification in °C	internal	0 to 100 (the maximum temperature depends on the system)	Setpoint of the system flow temperature. This setpoint only applies explicitly if the "Heat request only via Modbus" mode is deactivated.
holding register	36	Setpoint flow temperature in °C	internal	0 to 100 (the maximum temperature depends on the boiler type)	Setpoint of the current boiler flow temperature. This setpoint is considered the exclusive setpoint when the mode "Request heat only via Modbus" is activated.
Holding registers	37	Setpoint power in %	internal	0 to 100	Target value of the current output of the boiler.

© All rights reserved by Robert Bosch GmbH, also in the case of intellectual property rights registrations

Notes :

- We recommend updating the firmware of the control device and using control from version 1.3 (see chapter "Control of the control device via Modbus TCP/IP").
- If you carry out a firmware update on your control device that you are already using, you can continue to use the subsequent control via Modbus TCP/IP by activating the use of old Modbus addresses (see chapter "Required settings on the control Necessary settings on the controller").
- Data type integer (int) corresponds to 16 bits (bit₀ to bit₁₅), see chapter "data types and register width".
- The addresses mentioned here refer to the fact that the controller is configured as a master. If you are using the control device as a slave, please continue reading in the chapter "Control via Modbus TCP/IP in the controller network".

4.2.1 Temperature-controlled heat request via Modbus TCP/IP up to and including version 1.2.7

In order to be able to send a heat request via Modbus, you must first configure the controller accordingly.

Depending on which mode you have selected for the "Heat request only via Modbus" parameter, you must send a different address for the flow temperature setpoint.

Variant "Heat request only via Modbus" deactivated:

register	value for registers
System temperature Temperature default (Holding Register 2)	X in °C

"Heat request only via Modbus" variant activated:

register	value for registers
Operating mode (Holding Register 35)	2
Setpoint Power (Holding Register 37)	100%
Flow temperature setpoint (holding register 36)	X in °C

Note e :

- The "System temperature specification" tab is only relevant if the "Heat request only via Modbus" mode is deactivated. If the "Heat request only via Modbus" mode is activated, the "Setpoint flow temperature" register must be used for the temperature specification.
- X in °C corresponds to your specification.

4.2.2 Power-based heat request via Modbus TCP/IP up to and including version 1.2.7

In order to be able to send a power request via Modbus, you must first configure the controller accordingly. You can then transfer the following values to the following Modbus registers:

register	value for registers
Operating mode (holding register 35)	2
Setpoint Power (Holding Register 37)	X in %
Flow temperature setpoint (holding register 36)	100 in °C

Note e :

- X in % corresponds to your specification.

4.2.3 Information on the 'Heat request only via Modbus ' parameter

If you have connected additional function modules (e.g. for a heating circuit) or substations to the controller, these may generate a heat request. The handling of the heat requests in connection with Modbus TCP/IP can be defined via the parameter "Heat request only via Modbus".

If you have deactivated the "Heat request only via Modbus" parameter, the connected function modules/substations can issue a heat request. A heat request via Modbus, which comes from the BMS, can only add a heat request. This does not affect existing heat requirements from other modules. The setpoint is generally the highest heat requirement.

If the "Heat request only via Modbus" parameter is activated, additionally connected modules/substations cannot issue a heat request. With this setting, heat requests are only carried out by the BMS. The heat requirement is therefore available exclusively to the BMS, giving it full control of any heat requirements.

In contrast to the heat requirement, a power request is always available exclusively for the BMS. Therefore, you cannot carry out an output-based heat request if you have deactivated the "Heat request only via Modbus" parameter.