Operating Guide

iC7 Series Modbus RTU

Modbus RTU OS7MR

Explore our solutions
drives.danfoss.com

Contents

1 Introduction and Safety

1.1 Purpose of the Operating Guide

This operating guide provides information about configuring the system, controlling the drive or power converter, accessing parameters, programming, troubleshooting, and some typical application examples.

The operating guide is intended for use by qualified personnel, who are familiar with the iC7 drives and power converters, Modbus RTU technology, and the PC or PLC that is used as a master in the system.

Read the instructions before configuring Modbus RTU, and follow the procedures in this guide.

1.2 Additional Resources

Additional resources are available to help understand the features, and safely install and operate the iC7 products:

- Safety guides, which provide important safety information related to installing iC7 drives.
- Installation guides, which cover the mechanical and electrical installation of drives, or functional extension options.
- Operating guides, which include instructions for control options, and other components for the drive.
- Application guides, which provide instructions on setting up the drive for a specific end use. Application guides for application software packages also provide an overview of the parameters and value ranges for operating the drives, configuration examples with recommended parameter settings, and troubleshooting steps.
- Facts Worth Knowing, available for download on www.danfoss.com.
- Other supplemental publications, drawings, and guides are available at www.danfoss.com.

Latest versions of Danfoss product guides are available for download at https://www.danfoss.com/en/service-and-support/documentation/.

1.3 Safety Symbols

The following symbols are used in Danfoss documentation and products.

⚠ DANGER

Indicates a hazardous situation which, if not avoided, will result in death or serious injury.

⚠ WARNING

Indicates a hazardous situation which, if not avoided, could result in death or serious injury.

⚠ CAUTION

Indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.

NOTICE

Indicates information considered important, but not hazard-related (for example, messages relating to property damage).

	ISO warning symbol for general warnings
	ISO warning symbol for hot surfaces and burn hazard

	ISO warning symbol for high voltage and electric shock
	Symbol for indicating the required discharge time of the capacitors in the product.
	ISO action symbol for referring to the instructions

1.4 Qualified Personnel

Correct and reliable transport, storage, installation, operation, and maintenance are required for the trouble-free and safe operation of the product. Only qualified personnel are allowed to install and operate this equipment.

Qualified personnel are defined as trained staff, who are authorized to install, commission, and maintain equipment, systems, and circuits in accordance with pertinent laws and regulations. Also, the qualified personnel must be familiar with the instructions and safety measures described in this guide.

1.5 Safety Precautions

⚠ WARNING

HIGH VOLTAGE

Drives and power converters contain high voltage when connected to AC mains input, DC supply, or load sharing. Failure to perform installation, startup, and maintenance by qualified personnel can result in death or serious injury.

- Only qualified personnel are allowed to perform installation, startup, and maintenance.

⚠ WARNING

UNINTENDED START

When the drive or power converter is connected to the AC mains or connected on the DC terminals, the motor may start at any time, causing risk of death, serious injury, and equipment or property damage.

- Stop the drive or power converter before configuring parameters.
- Make sure that the drive or power converter cannot be started by an external switch, a fieldbus command, an input reference signal from the control panel, or after a cleared fault condition.
- Disconnect the drive or power converter from the mains whenever safety considerations make it necessary to avoid an unintended motor start.
- Check that the drive or power converter and any driven equipment are in operational readiness.

⚠ WARNING

DISCHARGE TIME

The drive or power converter contains DC-link capacitors, which can remain charged even when the drive or power converter is not powered. High voltage can be present even when the warning indicator lights are off. Failure to wait the specified time after power has been removed before performing service or repair work can result in death or serious injury.

- Stop the motor.
- Disconnect all power sources, including permanent magnet type motors.
- Wait for capacitors to discharge fully. The discharge time is specified on the drive or power converter product label.
- Measure the voltage level to verify full discharge.

⚠ WARNING

LEAKAGE CURRENT HAZARD

Leakage currents exceed 3.5 mA. Failure to ground the drive or power converter properly can result in death or serious injury.

- Ensure that the minimum size of the ground conductor complies with the local safety regulations for high touch current equipment.

⚠ WARNING

EQUIPMENT HAZARD

Contact with rotating shafts and electrical equipment can result in death or serious injury.

- Ensure that only trained and qualified personnel perform installation, start-up, and maintenance.
- Ensure that electrical work conforms to national and local electrical regulations.
- Follow the procedures in this guide.

⚠ CAUTION

INTERNAL FAILURE HAZARD

An internal failure in the drive or power converter can result in serious injury when the drive or power converter is not properly closed.

- Ensure that all safety covers are in place and securely fastened before applying power.

1.6 Abbreviations

Table 1: Abbreviations

Abbreviation	Definition
BOOL	Boolean
CTW	Control word
EMC	Electromagnetic compatibility
I/O	Input/Output
IP	Internet protocol
CP	Control panel

Table 1: Abbreviations - (continued)

Abbreviation	Definition
LED	Light-emitting diode
LSB	Least significant bit
MAV	Main actual value
MEI	Modbus encapsulated interface
MRV	Main reference value
MSB	Most significant bit
PC	Personal computer
PCD	Process channel data
PLC	Programmable logic controller
PNU	Parameter number
PPO	Process parameter object
REF	Reference
RTU	Remote Terminal Unit
STW	Status word

1.7 Trademarks

MODBUS® is a registered trademark of Schneider Electric USA, Inc.

1.8 Version History

This guide is regularly reviewed and updated. All suggestions for improvement are welcome.

The original language of this guide is English.

Table 2: Version History

Version	Remarks
AQ497328073284, version 0101	First release.

2 Product Overview

2.1 Modbus RTU Overview

Modbus RTU is a widely used communication protocol that defines how controllers (devices) communicate over a physical network, regardless of the type of network used. The communication process includes how the Modbus RTU responds to requests from another device, and how errors are detected and reported. It also establishes a common format for the layout and contents of telegram fields.

During communications over a Modbus RTU network, the protocol:

- determines how each controller learns its device address.
- recognizes a telegram addressed to it.
- determines which actions to take.
- extracts any data or other information contained in the telegram.

Modbus RTU follows a master-follower communication model. Only the master device can initiate communication, called a query. The follower devices respond to the request by either sending the required data or performing the requested action.

If a reply is required, the controller constructs the reply telegram and sends it. Controllers communicate using a master/follower technique in which only the master can initiate transactions (called queries). Followers respond by supplying the requested data to the master, or by acting as requested in the query. The master can address individual followers, or initiate a broadcast telegram to all followers. Followers return a response to queries that are addressed to them individually. No responses are returned to broadcast queries from the master.

The Modbus RTU protocol establishes the format for the master query by providing the following information:

- The device (or broadcast) address.
- A function code defining the requested action.
- Any data to be sent.
- An error-checking field.

Each device on the network has a unique address. The master includes this address in its message to ensure that only the intended follower responds.

When receiving a request from the master, the follower:

- detects if the message is addressed to it.
- understands the requested action.
- processes any data included in the message.
- builds and sends a reply message to the master, if needed.

The response telegram of the follower device is also constructed using the Modbus protocol. It contains fields confirming the action taken, any data to be returned, and an error-checking field. If an error occurs in receipt of the telegram, or if the follower is unable to perform the requested action, the follower constructs and sends an error message. Alternatively, a timeout occurs.

Fieldbus options for iC7 are integrated in the control board. Modbus RTU can be selected in the configurator when ordering a drive, or alternatively, it can be activated later.

Table 3: Modbus Model Codes

Model code	Description
+BAMR	Modbus RTU

Table 4: Modbus RTU Features

Feature	Technical data
Cyclic response	5 ms response time per variable (read/write non-persistent storage)
Supported Modbus objects	Coils (1 bit): Read-write
	Input register (16 bits): Read-only
	Holding register (16 bits): Read-write
Configuration	MyDrive® Insight

2.2 Communication Profiles and Objects

2.2.1 Overview

Communication profiles are selected in parameter ***Fieldbus Profile*** (parameter number ***1301***).

For Modbus, the fieldbus profile can also be selected via holding register location 1.

Table 5: Communication Profiles and Supported Applications

Product	Application software	Profiles
iC7-Aqua	Aqua	• iC Speed Profile • PROFIdrive
iC7-HVACR	HVAC	• iC Speed Profile • PROFIdrive

2.2.2 iC Speed Profile

2.2.2.1 Overview

The iC Speed profile is used with the iC7 series motor applications. The iC Speed profile differs from the PROFIdrive profile, because it does not have a state machine. It is only controlled by the actual state 1/0 of the control bits, not the sequence in which they are manipulated.

2.2.2.2 Control Word

Table 6: iC Speed Profile Control Word Bits

Bit number	Name	Description
0+1	Preset reference selector	00: Preset reference 1 01: Preset reference 2 10: Preset reference 3 11: Preset reference 4
2	Reserved	Reserved for future use. Any control words sent to the device should keep this bit at 0 to ensure compatibility with future extensions of the control word.
3	No coast/Coast	0: Causes the drive to immediately coast the motor. 1: No function.
4	No quick stop/Quick stop	0: Quick stops the drive and ramps down the motor speed to stop as defined with the quick stop ramp parameter. 1: No function.

Table 6: iC Speed Profile Control Word Bits - (continued)

Bit number	Name	Description
5	No hold/Hold output frequency	0: Hold the present output frequency (in Hz). 1: No function.
6	Start/No start	0: Stops the drive and ramps down the motor speed as defined with the ramp-down parameter. 1: If the other starting conditions are fulfilled, this selection allows the drive to start the motor.
7	Reset	0: No function. 0 ⇒ 1: Reset faults.[1]
8	Jog/No jog	0: No function. 1: Sets the output frequency to the jog speed defined with the jog speed parameter.
9	Ramp select	0: Ramp 1 is active. 1: Ramp 2 is active.
10	Data valid	0: Ignore the current process data. This is linked to the submodule where the CTW is present. If signals are to be covered, the CTW/STW profile (for example, the iC Speed profile) must be part of the signals list. 1: Use process data (controlled by PLC). Use the previously processed data when the data valid bit was true (no control by PLC).
11	Reserved	Reserved for future use.
12	User-defined	These bits are reserved for application-specific advanced control. Select the value ***CTW bit x***for any input parameter to use this signal for the activation of a selected function. For more information, refer to the *Parameter Descriptions* chapter in the application guide.
13	User-defined	
14	User-defined	
15	User-defined	

1) Edge-triggered from 0 to 1 to reset the fault.

2.2.2.3 Status Word

Table 7: iC Speed Profile Status Word Bits

Bit number	Name	Description
0	Control ready	0 = The device controls are not ready and do not react to process data. 1 = The device controls are ready and react to process data.
1	Frequency converter ready	0 = The frequency converter is not ready for operation. This status does not involve faults and warnings as they are indicated in their respective bits elsewhere. 1 = The frequency converter is ready for operation.
2	Coast	0 = The frequency converter has an active coast signal and has released the motor. 1 = There are no active coast signals, and the motor can start when a start signal is given.
3	Fault	0 = There are no faults. 1 = A fault has occurred, and an acknowledge signal is required to re-establish operation.

Table 7: iC Speed Profile Status Word Bits - (continued)

Bit number	Name	Description
4	Reserved	Reserved.
5	Reserved	Reserved.
6	Reserved	Reserved.
7	Warning	0 = There are no warnings. 1 = A warning is active.
8	Speed=reference	0 = The motor runs, but the current speed is different from the current speed reference, for example, while the speed ramps up or down during start or stop. 1 = The current motor speed matches the current speed reference within a given tolerance. The tolerance is product-specific.
9	Bus control/Local operation	0 = The device does not react on commands from the fieldbus, for 1 of the following reasons: • CTW bit 10 = 0. • HMI is in local mode. • MyDrive® Insight has taken control. • Control places do not include fieldbus. 1 = The device is controlled and reacting to I/O and process data.
10	Frequency limit	0 = The output frequency has exceeded the defined motor limits. 1 = The output frequency is within the defined motor limits. The speed limits are set with the parameters in parameter group ***5.8.3 Speed Limits and Monitors***.
11	Operation	0 = There are no active start requests, and the process does not run. The motor is coasted and is not started. 1 = The process is running, and the motor can be running or start at any time.
12	Reserved	Reserved.
13	Reserved	Reserved.
14	User-defined	These bits are reserved for application-specific advanced control. For more information, refer to the *Parameter Descriptions* chapter in the application guide.
15	User-defined	

2.2.3 PROFIdrive - Standard Telegram 1

2.2.3.1 Overview

Standard telegram 1 is implemented according to PROFIdrive Application Class 1 profile as defined in the PROFIdrive standard and state machine diagram. It can be used with iC7 series motor applications.

2.2.3.2 Control Word

Table 8: Control Word Bits in PROFIdrive Standard Telegram 1

Bit number	Name	Description
0	On-Off	0: Off. 1: On.
1	Coast stop	0: Coast stop. 1: No coast stop.
2	Quick stop	0: Quick stop. 1: No quick stop.
3	Operation	0: Disable operation. 1: Enable operation.
4	Ramp generator	0: Reset ramp generator. The output of the RFG is set to 0. The drive decelerates along the current limit or along the voltage limit of the DC link. 1: Enable ramp generator (RFG).
5	Freeze	0: Freeze ramp generator. Freezes the present output frequency (in Hz). 1: Unfreeze ramp generator.
6	Enable setpoint	0: Disable setpoint. 1: Enable setpoint.
7	Fault acknowledge	0: No function. 0 ⇒ 1: Acknowledge faults.[1]
8	Jog 1	0: Jog 1 off. 1: Jog 1 on. Operation is enabled, drive is at standstill, and STW1 bit 4, 5, 6: 0. The drive runs up along the ramp to jogging setpoint 1.
9	Jog 2	0: Jog 2 off. 1: Jog 2 on. Operation is enabled, drive is at standstill, and STW1 bit 4, 5, 6: 0. The drive runs up along the ramp to jogging setpoint 2.
10	Control by PLC	0: Ignores the current process data. This is linked to a submodule where the CTW is present. If signals are to be covered, the CTW/STW profile (for example, the iC Speed Profile) must be part of the signals list. 1: Uses process data (controlled by PLC).
11	–	Reserved
12	User-defined	These bits are reserved for application-specific advanced control. For more information, refer to the *Parameter Descriptions* chapter in the application guide.
13	User-defined	
14	User-defined	
15	User-defined	

1) Acknowledging is edge-triggered, when changing from logic 0 to logic 1.

2.2.3.3 Status Word

Table 9: Status Word Bits in PROFIdrive Standard Telegram 1

Bit number	Name	Description
0	Ready to switch on	0 = Not ready to switch on. 1 = Ready to switch on.
1	Ready to operate	0 = Not ready to operate. 1 = Ready to operate.
2	Operation enabled	0 = Operation disabled. 1 = Operation enabled.
3	Operation fault	0 = No fault. 1 = Fault present.
4	Coast stop	0 = Coast stop activated (OFF2). 1 = Coast stop not activated (No OFF2).
5	Quick stop	0 = Quick stop activated (OFF3). 1 = Quick stop not activated (No OFF3).
6	Switching on inhibited	0 = Switching on not inhibited. 1 = Switching on inhibited.
7	Warning	0 = There are no warnings. 1 = A warning has occurred.
8	Speed=reference/Speed<>reference	0 = The motor runs, but the current speed is different from the current speed reference. This can happen, for example, when the speed ramps up or down during start or stop. 1 = The current motor speed matches the current speed reference within a given tolerance. The tolerance is product specific.
9	Bus control/Local operation	0 = The device does not react on commands from fieldbus, because of one of the following reasons: • CTW bit 10 = 0. • HMI is in local mode. • MyDrive® Insight has taken control. • Control places do not include fieldbus. 1 = The device is controlled and reacting to I/O and process data.
10	Frequency limit ok/Out of frequency limit	0 = The output frequency has exceeded the defined motor limits given by parameters. 1 = The output frequency is within the defined motor limits. The speed limits are set with the parameters in parameter group ***5.8.3 Speed Limits and Monitors***.
11	User-defined	These bits enable the mapping of application functionality of the drive to the status word. Mapping is done through parameters. For more information, refer to the *Parameter Descriptions* chapter in the application guide.
12	User-defined	
13	User-defined	
14	User-defined	
15	User-defined	

2.2.3.4 PROFIdrive State Machine

In the PROFIdrive control profile, the control bits perform different functions:

- 0–3 perform the basic startup and power-down functions.
- 4–10 perform application-oriented control.
- 12–15 can be configured for different purposes.

See Figure 1 for the basic state transition diagram, where control bits 0–3 control the transitions and the corresponding status bit indicates the actual state. The black dots indicate the priority of the control signals. Fewer dots indicate lower priority, and more dots indicate higher priority.

The general state diagram is defined in the PROFIdrive standard.

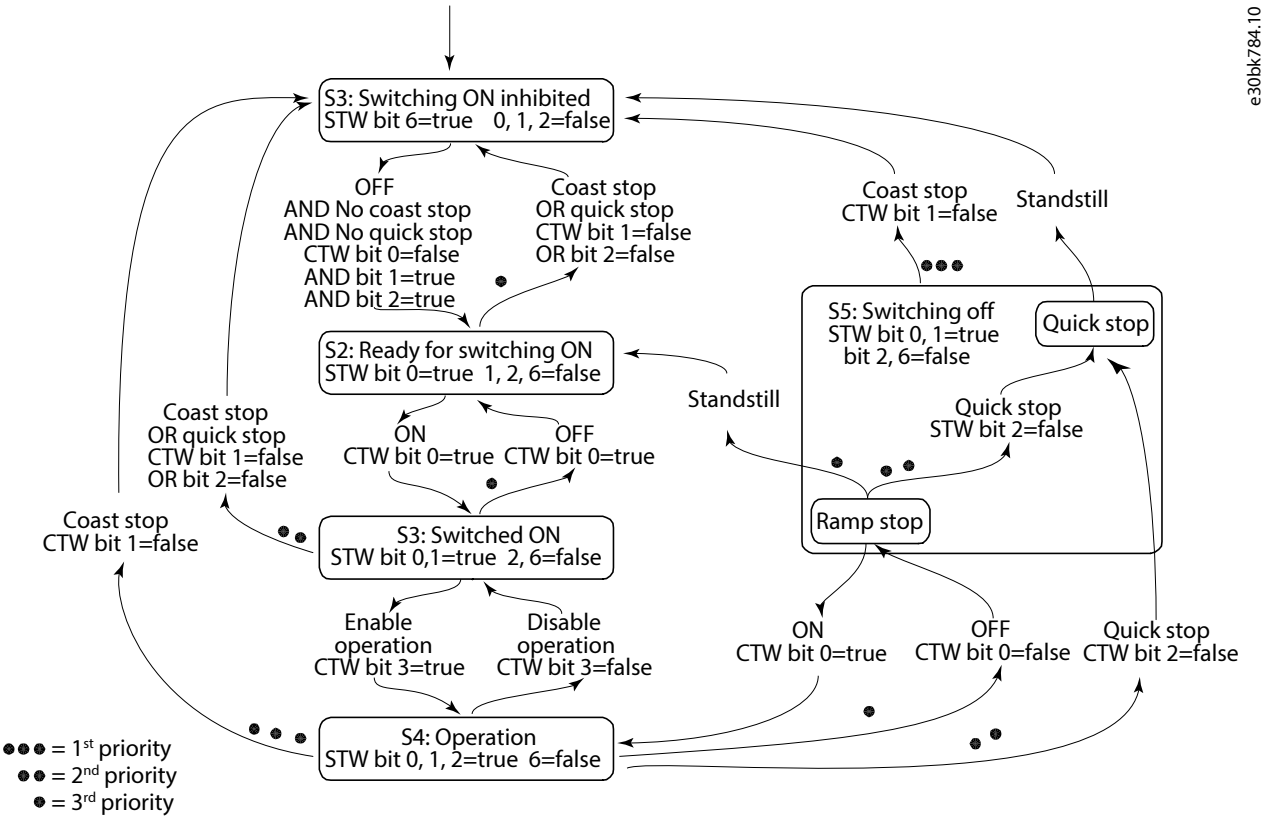


Figure 1: General State Diagram

2.3 Network Topology

Communication interface X4 is used for the RS-485 fieldbus connection.

RS-485 is a 2-wire bus interface compatible with multi-drop network topology, that is, nodes can be connected as a bus, or via drop cables from a common trunk line.

A total of 32 nodes can be connected to 1 network segment. Repeaters divide network segments.

> **IMPORTANT:** Each repeater functions as a node within the segment in which it is installed. Each node connected within a given network must have a unique node address across all segments.

NOTICE

Installing drives or power converters of different current ratings in line topology may result in unwanted power-off behavior.

- Mount the drives or power converters with the longest discharge time first in the line topology. In normal operation, the drives or power converters with bigger current ratings have a longer discharge time.

2.4 Modbus RTU Message Framing Structure

2.4.1 Modbus RTU Telegram Structure

The transmitting device places a Modbus RTU telegram into a frame with a known beginning and ending point. This allows receiving devices to begin at the start of the telegram, read the address portion, determine which device is addressed (or all devices, if the telegram is broadcast), and to recognize when the telegram is completed. Partial telegrams are detected and errors set as a result. Characters for transmission must be in hexadecimal 00–FF format in each field. The drive continuously monitors the network bus, also during silent intervals. When the 1st field (the address field) is received, each drive or device decodes it to determine which device is being addressed. Modbus RTU telegrams addressed to 0 are broadcast telegrams. No response is allowed for broadcast telegrams. A typical telegram frame is shown in the following table.

Table 10: Typical Modbus RTU Telegram Structure

Start	Address	Function	Data	CRC check	End
T1-T2-T3- T4	8 bits	8 bits	N x 8 bits	16 bits	T1-T2-T3- T4

2.4.2 Start/Stop Field

Telegrams start with a silent period of at least 3.5 character intervals. The silent period is implemented as a multiple of character intervals at the selected network baud rate (shown as Start T1-T2-T3-T4). The 1st field to be transmitted is the device address. Following the last transmitted character, a similar period of at least 3.5 character intervals marks the end of the telegram. A new telegram can begin after this period.

Transmit the entire telegram frame as a continuous stream. If a silent period of more than 1.5 character intervals occurs before completion of the frame, the receiving device flushes the incomplete telegram and assumes that the next byte is the address field of a new telegram. Similarly, if a new telegram begins before 3.5 character intervals after a previous telegram, the receiving device considers it a continuation of the previous telegram. This behavior causes a timeout (no response from the follower), since the value in the final CRC field is not valid for the combined telegrams.

2.4.3 Address Field

The address field of a telegram frame contains 8 bits. Valid follower device addresses are in the range of 0–247 decimal. The individual follower devices are assigned addresses in the range of 1–247. 0 is reserved for broadcast mode, which all followers recognize. A master addresses a follower by placing the follower address in the address field of the telegram. When the follower sends its response, it places its own address in this address field to let the master know which follower is responding.

2.4.4 Function Field

The function field of a telegram frame contains 8 bits. Valid codes are in the range of 1–FF. Function fields are used to send telegrams between master and follower. When a telegram is sent from a master to a follower device, the function code field tells the follower what kind of action to perform. When the follower responds to the master, it uses the function code field to indicate either a normal (error free) response, or that some kind of error occurred (called an exception response).

For a normal response, the follower simply echoes the original function code. For an exception response, the follower returns a code that is equivalent to the original function code with its most significant bit set to logic 1. In addition, the follower places a unique code into the data field of the response telegram. This code tells the master what kind of error occurred, or the reason for the exception. Also refer to 2.5.3.1 Supported Modbus RTU Function Codes and 2.5.6.1 Typical Response Codes.

2.4.5 Data Field

The data field is constructed using sets of 2 hexadecimal digits, in the range of 00–FF hexadecimal. These digits are made up of 1 RTU character. The data field of telegrams sent from a master to a follower device contains additional information which the follower must use to perform accordingly.

The information can include items such as:

- Coil or register addresses.
- The quantity of items to be handled.
- The count of actual data bytes in the field.

2.4.6 CRC Check Field

Telegrams include an error-checking field, operating based on a cyclic redundancy check (CRC) method. The CRC field checks the contents of the entire telegram. It is applied regardless of any parity check method used for the individual characters of the telegram. The transmitting device calculates the CRC value and appends the CRC as the last field in the telegram. The receiving device recalculates a CRC during receipt of the telegram and compares the calculated value to the actual value received in the CRC field. 2 unequal values result in bus timeout. The error-checking field contains a 16-bit binary value implemented as 2 8-bit bytes. After the implementation, the low-order byte of the field is appended first, followed by the high-order byte. The CRC high-order byte is the last byte sent in the telegram.

2.5 Modbus Features

2.5.1 Data Objects

Table 11: Supported Object Types

Object type	Access	Size
Coil	Read/write	1 bit
Input register	Read only	16 bits
Holding register	Read/write	16 bits

Table 12: Modbus RTU Frame Components

Component	Size (bytes)	Description
Follower Address	1	Identifies the target device.
Function Code	1	Specifies the operation, for example, read/write.
Data Field	0–252	Variable length depending on function and data.
CRC	2	Cyclic Redundancy Check for error detection.

- Minimum frame size: 4 bytes, for example, a simple read with no data.
- Maximum frame size: 256 bytes (1 + 1 + 252 + 2)

2.5.2 Follower Address

FollowerAddress is the address of the device. It can take a value from 0 to 247; addresses from 248 to 255 are reserved.

2.5.3 Function Codes

2.5.3.1 Supported Modbus RTU Function Codes

The function code of a message frame contains 8 bits. Valid codes are in the range of 1–FF. Use function codes to send messages between client and server. When a message is sent from a client to a server device, the function code tells the server which action to perform. When the server responds to the client, it uses the function code to indicate either a normal (error-free) response, or that some error occurred (called an exception response). For a normal response, the server echoes the original function code. For an exception response, the server returns a code that is equivalent to the original function code with its most significant bit set to logic 1. Furthermore, the server places a unique code into the data field of the response message. It tells the client which error occurred, or the reason for the exception.

Table 13: Supported Function Codes

Function code		Object type	Access type	Address range
Dec	Hex			
1	1	Read coils	Discrete (1 bit)	00001–09999
3	3	Read multiple holding registers	Register (16 bit)	40001–49999
4	4	Read input registers	Register (16 bit)	30001–39999
5	5	Write single coils	Discrete (1 bit)	00001–09999
6	6	Write single holding register	Register (16 bit)	40001–49999
15	F	Write multiple coils	Discrete (1 bit)	00001–09999
16	10	Write multiple holding registers	Register (16 bit)	40001–49999
23	17	Read/write multiple registers	Register (16 bit)	40001–49999
43	28	Read device ident	MEI	–

2.5.3.2 Read Coil Status (01 hex)

Description

This function reads the ON/OFF status of discrete outputs (coils) in the drive. Broadcast is never supported for reads.

Query

The query telegram specifies the starting coil and quantity of coils to be read. Coil addresses start at 0, that is, coil 33 is addressed as 32.

Example: request to read coils 33–48 (status word) from follower device 01.

Table 14: Query

Field name	Example (hex)
Follower address	01 (drive address)
Function	01 (read coils)
Starting address HI	00
Starting address LO	20 (32 decimals) coil 33
Number of points HI	00
Number of points LO	10 (16 decimals)
Error check (CRC)	–

Response

The coil status in the response telegram is packed as 1 coil per bit of the data field. Status is indicated as: 1 = ON; 0 = OFF. The lsb of the 1st data byte contains the coil addressed in the query. The other coils follow toward the high-order end of this byte, and from low order to high order in subsequent bytes.

If the returned coil quantity is not a multiple of 8, the remaining bits in the final data byte are padded with values 0 (toward the high-order end of the byte). The byte count field specifies the number of complete bytes of data.

Table 15: Response

Field name	Example (hex)
Follower address	01 (drive address)
Function	01 (read coils)
Byte count	02 (2 bytes of data)
Data (coils 40–33)	07
Data (coils 48–41)	06 (STW = 0607hex)
Error check (CRC)	–

NOTE: Coils and registers are addressed explicitly with an offset of -1 in Modbus. For example, coil 33 is addressed as coil 32.

2.5.3.3 Read Holding Registers (03 hex)

Description

This function reads the contents of holding registers in the follower.

Query

The query telegram specifies the starting register and quantity of registers to be read. Register addresses start at 0, that is, registers 1–4 are addressed as 0–3.

Example: read parameter ***5.8.3.3 Maximum Speed Limit*** (parameter number 1722), mapped as 2 consecutive floating points to holding registers 3029 and 3030.

Table 16: Query

Field name	Example (hex)
Follower address	01
Function	03 (Read holding registers)
Starting address HI	0B (Register address 3029)
Starting address LO	D5 (Register address 3029)
Number of points HI	00
Number of points LO	02 – (***5.8.3.3 Maximum Speed Limit*** is 32 bits long, that is, 2 registers.)
Error check (CRC)	–

Response

The register data in the response telegram is packed as 2 bytes per register, with the binary contents right justified within each byte. For each register, the 1st byte contains the high-order bits, and the 2nd contains the low-order bits.

Example: 00004248 = 50 Hz as 32-bit floating point with little-endian byte swap.

Table 17: Response

Field name	Example (hex)
Follower address	01
Function	03
Byte count	04
Data HI (register 3030)	00
Data LO (register 3030)	00
Data HI (register 3031)	42
Data LO (register 3031)	48
Error check (CRC)	–

2.5.3.4 Read Input Registers (04 hex)

Description

This function reads the binary contents of input registers in the follower.

Query

The query message specifies the starting register and the quantity of registers to be read. Register addresses start at 0, that is, registers 1–4 are addressed as 0–3.

Example: read parameter ***5.8.3.3 Maximum Speed Limit*** (parameter number 1722), mapped as 2 consecutive floating points to holding registers 20 and 21.

Table 18: Query

Field name	Example (hex)
Follower address	1
Function	04 (read input registers)
Starting address HI	0B (register address 3029)
Starting address LO	D5 (register address 3029)
Number of points HI	0
Number of points LO	Parameter ***5.8.3.3 Maximum Speed Limit*** (parameter number 1722), mapped as 2 consecutive floating points to holding registers 20 and 21.
Error check (CRC)	-–

Response

The register data in the response message are packed as 2 bytes per register, with the binary contents right justified within each byte. For each register, the 1st byte contains the high order bits and the 2nd contains the low-order bits.

Example: hex 00004248 = 50 Hz as 32-bit floating point with little-endian byte swap.

Table 19: Response

Field name	Example (hex)
Follower address	1
Function	4
Byte count	4

Table 19: Response - (continued)

Field name	Example (hex)
Data HI (register 3030)	0
Data LO (register 3030)	0
Data HI (register 3031)	42
Data LO (register 3031)	48
Error check (CRC)	–

2.5.3.5 Write Single Coil (05 hex)

Description

This function writes the coil to either ON or OFF. When broadcast, the function writes the same coil references in all attached followers.

Query

The query telegram specifies the coil 65 (parameter write control) to be written. Coil addresses start at 0, that is, coil 65 is addressed as 64. Write data = 00 00 hex (OFF) or FF 00 hex (ON).

Table 20: Query

Field name	Example (hex)
Follower address	01 (drive address)
Function	05 (write single coil)
Coil address HI	00
Coil address LO	40 (64 decimal) Coil 65
Write data HI	FF
Write data LO	00 (FF 00 = ON)
Error check (CRC)	–

Response

The normal response is an echo of the query, returned after the coil state has been written.

Table 21: Response

Field name	Example (hex)
Follower address	01
Function	05
Write data HI	FF
Write data LO	00
Quantity of coils HI	00
Quantity of coils LO	01
Error check (CRC)	–

2.5.3.6 Write Single Holding Register (06 hex)

Description

This function presets a value into a single holding register.

Query

The query telegram specifies the register reference to be preset. Register addresses start at 0, that is, register 1 is addressed as 0.

Example: set parameter ***5.4.2.16 Operation Mode*** (parameter number 2500), mapped as Single Unsigned to holding register 1000 to Process Control (value 21 decimal).

Table 22: Query

Field name	Example (hex)
Follower address	01
Function	06
Starting address HI	03 (register address 999)
Starting address LO	E7 (register address 999)
Preset data HI	00
Preset data LO	15
Error check (CRC)	–

Response

The normal response is an echo of the query, returned after the register contents have been passed.

Table 23: Response

Field name	Example (hex)
Follower address	01
Function	06
Register address HI	03
Register address LO	E7
Preset data HI	00
Preset data LO	15
Error check (CRC)	–

2.5.3.7 Write Multiple Coils (0F hex)

Description

This function writes each coil in a sequence of coils to either on or off. When broadcasting, the function writes the same coil references in all attached followers.

Query

The query telegram specifies the coils 17–32 (speed setpoint) to be written.

NOTE: Coil addresses start at 0, that is, coil 17 is addressed as 16.

Table 24: Query

Field name	Example (hex)
Follower address	01 (drive address)
Function	0F (write multiple coils)
Coil address HI	00

Table 24: Query - (continued)

Field name	Example (hex)
Coil address LO	10 (coil address 17)
Quantity of coils HI	00
Quantity of coils LO	10 (16 coils)
Byte count	02
Write data HI (Coils 8–1)	20
Write data LO (Coils 16–9)	00 (reference = 2000 hex)
Error check (CRC)	–

Response

The normal response returns the follower address, function code, starting address, and quantity of coils written.

Table 25: Response

Field name	Example (hex)
Follower address	01 (drive address)
Function	0F (write multiple coils)
Coil address HI	00
Coil address LO	10 (coil address 17)
Quantity of coils HI	00
Quantity of coils LO	10 (16 coils)
Error check (CRC)	–

2.5.3.8 Write Multiple Holding Registers (10 hex)

Description

This function presets values into a sequence of holding registers.

Query

The query telegram specifies the register references to be written.

Example: set parameter ***4.2.2.3 Nominal Current*** (parameter number 400), mapped as 2 consecutive floating points to holding register 1500 to 7.38 A.

Table 26: Query

Field name	Example (hex)
Follower address	01
Function	10
Starting address HI	05
Starting address LO	DB
Number of registers HI	00
Number of registers LO	02
Byte count	04
Write data HI (Register 4: 1049)	28

Table 26: Query - (continued)

Field name	Example (hex)
Write data LO (Register 4: 1049)	F6
Write data HI (Register 4: 1050)	40
Write data LO (Register 4: 1050)	40
Error check (CRC)	–

Response

The normal response returns the follower address, function code, starting address, and quantity of registers preset.

Table 27: Response

Field name	Example (hex)
Follower address	01
Function	10
Starting address HI	05
Starting address LO	DB
Number of registers HI	00
Number of registers LO	02
Error check (CRC)	–

2.5.3.9 Read/Write Multiple Registers (17 hex)

Description

This function code performs a combination of 1 read operation and 1 write operation of holding registers in a single Modbus transaction. The write operation is performed before the read operation.

Query

The query message specifies the starting address and number of holding registers to be read as well as the starting address, number of holding registers, and the data to be written. Holding registers are addressed starting at 0.

Example: request to set parameter ***4.2.2.2 Motor Current*** to 738 (7.38 A) and read parameter ***5.8.3.1 Maximum Speed Limit*** which has the value 50000 (50 Hz). Both parameters are mapped as Two Consecutive Float, little-endian byte swap.

Table 28: Query

Field name	Example (hex)
Follower address	01
Function	17
Reading starting address HI	0B (Register 3030)
Reading starting address LO	D5 (Register 3030)
Quantity to read HI	00
Quantity to read LO	02 (Parameter ***5.8.3.1 Maximum Speed Limit*** is 32 bits long, that is, 2 registers.)
Write starting address HI	05 (Register 1500)
Write starting address LO	DB (Regster1500)
Quantity to write HI	00

Table 28: Query - (continued)

Field name	Example (hex)
Quantity to write LO	02
Write byte count	04
Write registers value HI	04
Write registers value LO	28
Write registers value HI	F6
Write registers value LO	40
Error check (CRC)	–

Response

The normal response contains the data from the group of registers that were read. The byte count field specifies the quantity of bytes to follow in the read data field.

Table 29: Response

Field name	Example (hex)
Follower address	01
Function	17
Byte count	04
Read registers value HI	00
Read registers value LO	00
Read registers value HI	42
Read registers value LO	48
Error check (CRC)	–

2.5.3.10 Read Device Identity (2B HEX)

The Read Device Identity function code is for tunneling service requests through Modbus PNUs. iC7 series drives support Modbus Encapsulated Interface (MEI) type 14 for transmitting device identification information. All information used to generate the reply is read from related manifest items.

Table 30: Request Structure

Function code	0x2B
MEI type	0x0E
MEI type specific data	Object ID See Table 31.

Table 31: Object IDs

Object ID	Object name/description	Type	M/O	Category
0x00	VendorName	ASCII String	Mandatory	Basic
0x01	ProductCode	ASCII String	Mandatory	
0x02	MajorMinorRevision	ASCII String	Mandatory	

Table 31: Object IDs - (continued)

Object ID	Object name/description	Type	M/O	Category
0x03	VendoreUrl	ASCII String	Optional	Regular
0x04	ProductName	ASCII String	Optional	
0x05	ModelName	ASCII String	Optional	
0x06	UserApplicationName	ASCII String	Optional	
0x07–0x7F	Reserved	–	Optional	Extended
0x80–0xFF	Private objects can be defined optionally. The range 0x80–0xFF is product-dependent.	Device-dependent	Optional	

2.5.4 Data Field

The data field is a crucial part of the message frame. It contains the actual information being transmitted between devices. Its structure and content vary depending on the function code used in the message.

- Read requests include starting address and quantity.
- Write requests include starting address, quantity, byte count, and values.
- Can be up to 252 bytes.

2.5.5 Cyclic Redundancy Check (CRC)

The Cyclic Redundancy Check (CRC) in Modbus RTU is a 16-bit checksum used to detect errors in transmitted messages. It ensures that the data received by a follower or master device is exactly what was sent.

- 16-bit checksum (2 bytes).
- Ensures message integrity.
- Calculated using a standard Modbus polynomial.

2.5.6 Error Handling

2.5.6.1 Typical Response Codes

In Modbus RTU communication, the follower device sends responses to the master device requests. Typical response codes include:

- Success response: The follower device successfully processes the request and returns the requested data or confirmation. This includes the function code and the relevant data or status.
- Error response: If the follower device encounters an error, it sends an exception response. The function code in the response is the original function code plus 0x80, followed by an exception code indicating the error type. Common exception codes include:

  01: Illegal Function: The function code is not supported.

  02: Illegal Data Address: The data address is not valid.

  03: Illegal Data Value: The data value is not allowable.

  04: Follower Device Failure: An unrecoverable error has occurred.

  For more information about the exception codes, see Table 32.

Table 32: Modbus Exception Codes

Code	Name	Meaning
01	Illegal function	The function code received in the query is not an allowable action for the server (or follower). This may be because the function code is only applicable to newer devices and was not implemented in the unit selected. It could also indicate that the server (or follower) is in the wrong state to process a request of this type, for example, because it is not configured and is being asked to return register values.
02	Illegal data address	The data address received in the query is not an allowable address for the server (or follower). More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers, a request with offset 96 and length 4 succeeds, while a request with offset 96 and length 5 generates exception 02.
03	Illegal data value	A value contained in the query data field is not an allowable value for the server (or follower). This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It does not mean that a data item submitted for storage in a register has a value outside the expectation of the application program, since the Modbus protocol is unaware of the significance of any value of any register.
04	Follower device failure	An unrecoverable error occurred while the server (or follower) was attempting to perform the requested action. If the code occurs while accessing the parameter values in the drive, detailed information about the latest exception can be read from Holding Register 0007. This register may contain 1 of the detailed error codes regarding the latest Modbus exception.

2.5.6.2 Error Handling and CRC Calculation

Each frame includes a 2-byte CRC value for error detection, computed using a polynomial algorithm to ensure data integrity during communication. Proper CRC handling is crucial for reliable Modbus operations, preventing data corruption and ensuring accurate device control.

3 Modbus Data Mapping

3.1 Mapping Modbus Coils

A Modbus coil is a single-bit binary data item which can be both read and written. Coils can be mapped to specific bits in a parameter by using the customization feature in MyDrive® Insight.

Only bit-type parameters can be mapped to coils. The source type must be INT, UINT, USINT DINT, UDINT, WORD, or BOOL.

1. In MyDrive® Insight, go to *Customization* > *Live* > *MODBUS* > *Coils*.
   - To add a single coil, click the + *Add Item* icon.
   - To map multiple coils to a single parameter, click the – *Add Multiple Items* icon.

Table 33: Coil Mapping Settings

Column	Description
Location	Refers to the coil number. Coils are mapped from address 00000–0FFFF.
Reference type	• ***Terminal***: Mapping a terminal on an option to a coil. • ***Parameter***: Mapping a specific bit of a parameter to a coil. • ***Feature***: Functions such as store and restore.
Reference	The name of the mapped parameter, feature, or event.
Source type	Data type of mapped parameter
Index	Index pointer for array parameters
Bit number	Mapped bit number of the selected parameter. 0 refers to the 1st bit.
Permissions	Setting Read/Write permissions for each coil.
Watchdog	***Watchdog1***: Process data timeout watchdog. Timeout time is configured in parameter ***Process Data Timeout Time (1340)***. The Watchdog timer is (re)triggered when writing to the mapped coil or register. ***Watchdog2***: Not used

3.2 Mapping Modbus Holding Registers

A Modbus holding register is a bit binary data item which can be both read and written. Registers can be mapped to a specific parameter by using the customization feature in MyDrive® Insight. The first 19 holding registers are pre-mapped or reserved and cannot be changed. From location 20 onwards, registers can be mapped freely.

IMPORTANT: The parameter mapped to 1 register can only be remapped to another register.

IMPORTANT: A mismatch between the datatype and register type issues a warning of possible data loss.

Table 34: Modbus Holding Registers

Location	Reference type	Reference (parameter number)
1	Parameter	***Fieldbus Profile (1301)***
2	Parameter	***Fieldbus Control Word (1335)***
3	Parameter	***Fieldbus Main Reference (1339)***
4	Parameter	Reserved

Table 34: Modbus Holding Registers - (continued)

Location	Reference type	Reference (parameter number)
5	Parameter	Reserved
6	Parameter	Reserved
7	Parameter	Reserved
8	Parameter	Reserved
9	Parameter	Reserved
10	Parameter	***Fieldbus Status Word (1307)***
11	Parameter	***Fieldbus Main Actual Value (1308)***
12	Parameter	Reserved
13	Parameter	Reserved
14	Parameter	Reserved
15	Parameter	Reserved
16	Parameter	Reserved
17	Parameter	Reserved
18	Parameter	Reserved
19	Parameter	Reserved
20	...	...
…	…	...

1. To map a parameter, go to *Customization* > *Live* > *MODBUS* > *Holding Registers* in MyDrive® Insight.
   - To add a single register, click the + *Add Item* icon.
   - To add multiple registers, click the – *Add Multiple Items* icon.

Table 35: Holding Register Settings

Column	Description
Location	Refers to the register number. Coils are mapped from address 40000–4FFFF.
Reference type	• ***Terminal***: Mapping a terminal on an option to a register. • ***Parameter***: Mapping a parameter to a register. • ***Constant***: Mapping a constant value to a register. • ***Feature***: Mapping a register to a feature, for example, Factory restore. • ***Event***: Mapping events to registers.
Reference	The name of the mapped parameter, feature, or event. Value for ***Constant*** type.
Source type	Data type of mapped parameter.
Index	Index pointer for array parameters.
Length	Number of bytes for string-type parameters.
Register type	Data type of the mapped parameter. By default, the Customizer selects a register type that matches the data type of the selected parameter.
Scaling	Scales the parameter value on the Modbus interface by dividing it by the scaling value entered.

Table 35: Holding Register Settings - (continued)

Column	Description
Unit	Unit of the mapped parameter
Permissions	Setting Read/Write permissions for each register.
Watchdog	***Watchdog1***: Process data timeout watchdog. Timeout time is configured in parameter ***Process Data Timeout Time (1340)***. The Watchdog timer is triggered when writing to the mapped coil or register. ***Watchdog2***: Not used

Mapping parameter of the REAL 32-bit datatype results in 2 consecutive register mappings.

Table 36: Example of a Holding Register Mapping

Location	Reference type	Reference (parameter number)	Source type	Register type
20	Parameter	Motor Current (9000)	REAL	Two Consecutive Floating point
22	Parameter	Heat Sink Temperature (2950)	REAL	Two Consecutive Floating point
24	...	...	...	...

Events from the device event log can be mapped into holding registers, starting from the most recent event. Each Event results in 6 consecutive register mappings.

Table 37: Example of an Event Register Mapping

Register	Description	Format
n	Timestamp	Seconds/milliseconds [SS.SSS](1)
n+1		Hours/Minutes [HHMM](1)
n+2		Month/Day [MMDD](1)
n+3		Year [YYYY](1)
n+4	Event type	MSB: 0 = Inactive event. MSB: 1 = Active event. LSB: 010 = Info. LSB: 011 = Warning. LSB: 100 = Fault.
n+5	Event code	See the relevant application guide.

1) Readable in hexadecimal format

2. To map a feature, go to *Customization* > *Live* > *MODBUS* > *Holding Registers* and click Add (+) in live mode.

It is also possible to map features to registers in MyDrive® Insight.

Table 38: Example of a Feature Register Mapping

Location	Reference type	Reference (feature)	Source type	Register type
20	Feature	Device Reset Operations	UINT	Single Unsigned

3.3 Mapping Modbus Input Registers

A Modbus input register is a 16-bit read-only value.

Input registers can be mapped to specific parameters by using the customization feature in MyDrive® Insight.

1. In MyDrive® Insight, go to *Customization* > *Live* > *MODBUS* > *Coils*.
   - To add a single register, click the + *Add Item* icon.
   - To add multiple registers, click the – *Add Multiple Items* icon.

Table 39: Coil Mapping Settings

Column	Description
Location	Refers to the coil number. Coils are mapped from address 30000–39999.
Reference type	• Parameter: Mapping a register to a parameter. • Constant: Mapping a register to a constant value. • Event: Mapping events to registers.
Reference	The name of the mapped parameter, feature, or event. Value for ***Constant*** type.
Source type	Data type of mapped parameter
Index	Index pointer for array parameters.
Length	Length as number of bytes for string-type parameters.
Register type	Data type of the mapped parameter. By default, the Customizer selects a register type that matches the data type of the selected parameter.
Scaling	Scales the parameter value on the Modbus interface by dividing it by the scaling value entered.
Unit	Unit of the mapped parameter.

3.4 Setting Permissions

Permissions can be set either in the *Permissions* column in the mapping view, or in the *Permissions* view in MyDrive® Insight.

1. In MyDrive® Insight, go to *Customization* > *Live* > *MODBUS* > *Permissions*.
2. Set the Read/Write permission for coils and registers:
   - By using the *Permissions* column in the mapping view.
   - By using the *Permissions* view in MyDrive® Insight.

> **!** IMPORTANT: Individual settings for each coil or register overwrite the default selection ***ALL***. If the permission is not set for a specific coil, or register, the selection ***ALL*** is predominant.

3.5 Saving Modbus Configuration to a Device

After the Modbus mapping is finalized, the configuration can be saved as a new project or added to an existing MyDrive® Insight project.

1. To save a configuration, click the *Save* icon.
2. To deploy a configuration directly into a device, click the *Download* icon to create and save an export package (*.vpkg).
3. After exporting the configuration, update and power cycle the device with the exported software package.

Table 40: Example of an Export Package

Package name (Default: Fully Qualified Domain Name)	###
Package version (Default: 1.0.0)	###

4 Fieldbus Cable Connections

4.1 Prerequisites for Installation

Communication interfaces are integrated in the control board in iC7 drives and power converters.

The position of the connections differs based on the control board concept and frame, for example. For more information on the location of the connections, cabling, and shielding, refer to the product-specific design guide.

4.2 Installation

One or more frequency converters can be connected to a control (or master) using the RS-485 interface. X4 Terminal 65 is connected to the A+ signal, while terminal 67 is connected to the B- signal. If more than 1 frequency converter is connected to a master, use parallel connections.

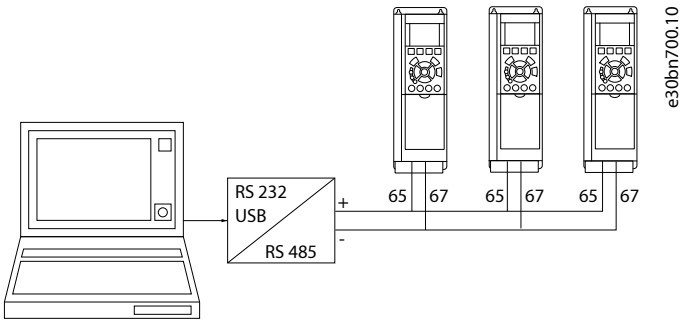


Figure 2: RS-485 Network Connection

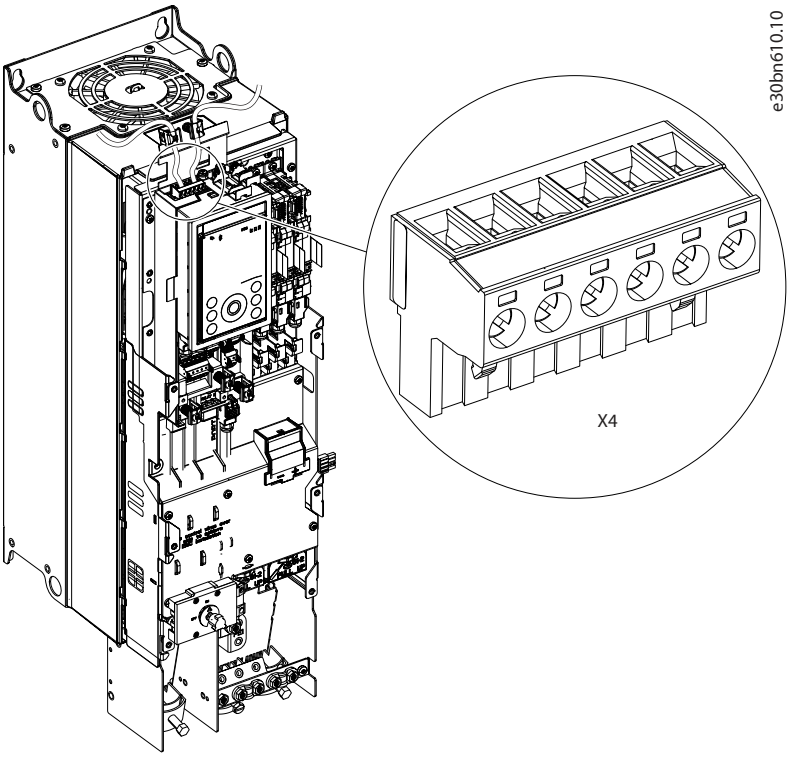


Figure 3: Location of the X4 Terminal in Frames FA02–FA12

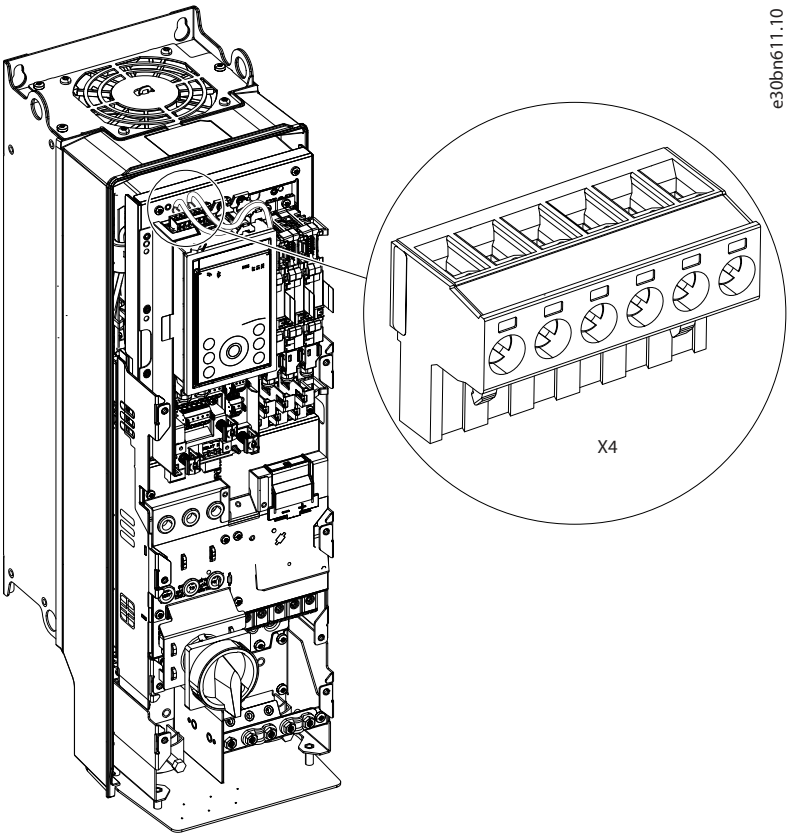


Figure 4: Location of the X4 Terminal in Frames FK03, FK05–FK12, FB03, and FB05–FB12

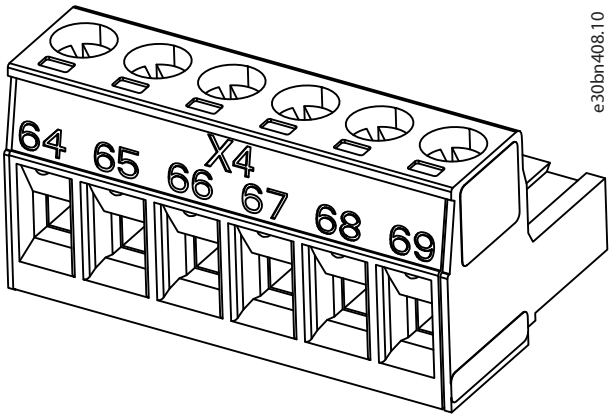


Figure 5: X4 Connector

Table 41: X4 Terminal Functions

Terminal number	Name	Function
64	GND	The GND terminal is the isolated ground. This terminal can be used, for example, for adding external bias resistors if required.
65	A+	The positive or data+ of the RS-485 bus.
66	Shield	The shield terminal has an RC network connected internally to the ground terminal.
67	B-	The negative or data- of the RS-485 bus.

Table 41: X4 Terminal Functions - (continued)

Terminal number	Name	Function
68	5 V out	5 V out is the output to the isolated 5 V fieldbus supply. This terminal can be used, for example, for adding external bias resistors if required.
69	Dir	Dir is the direction pin. It is driven to +5 V whenever data is transmitted out.

Observe the following guidelines:

- Terminate each segment at both ends, using either the termination switch (S801) of the drives or a biased termination resistor network. The termination switch is located on the top of the control unit below the X4 socket. The control panel and its cradle must be removed to access the switch.
- Always use shielded twisted pair (STP) cable for bus cabling and follow good common installation practice.
- Low-impedance ground connection of the shield at every node is important, including at high frequencies. Sometimes, it is necessary to apply potential-equalizing cables to maintain the same ground potential throughout the network, particularly in installations with long cables.
- To prevent impedance mismatch, use the same type of cable throughout the entire network. When connecting a motor to the drive, always use a shielded motor cable.

Table 42: Technical Specifications

Function	Data
Cable type	Shielded twisted pair Cu, 75 °C (167 °F)
Flexible/rigid wire without cable end sleeves	0.2–1.5 mm^2 (24–16 AWG)
Flexible wire without cable end sleeves	0.2–1.5 mm^2 (24–16 AWG)
Supported baud rates	9600, 19200, 38400, 57600, 76800, 115200
Supported Modbus function codes	See 2.5.3.1 Supported Modbus RTU Function Codes.

The recommended maximum Modbus cable length between the device and the controller should not exceed 1200 meters (4000 ft). While the RS-485 specification does not specify cabling, the recommendation is 24 AWG shielded twisted-pair cable with a shunt capacitance of 16 pF/ft and 100 Ω impedance. Another choice is the same cable commonly used in the twisted pair Ethernet cabling. This cable is commonly referred to as a Category 5/5e cable. The cable has a maximum capacitance of 17 pF/ft (14.5 pF/ft typical) and characteristic impedance of 100 Ω.

To avoid potential equalizing currents in the shield, terminate the cable shield via the terminal which is connected to the frame via an RC link.

5 Configuration

5.1 Selecting the Fieldbus Profile

For more information about the available profiles for the application in use, see Communication Profiles and Objects.

1. Navigate to *Parameters* > *Control Places* > *Fieldbus Control* > *Fieldbus Profile*.
2. Select the fieldbus profile.

The selection affects the interpretation of the control word and status word. Selecting a profile may also automatically set specific values for certain parameters or affect control schemes. For details, refer to the application guide.

5.2 General Connectivity Settings

General connectivity settings are in parameter group *10 Connectivity* > *Integrated Communication* > *Protocols* > *General Settings*.

Table 43: General Connectivity Parameters

Parameter	Parameter number	Value	Description
Fieldbus Fault Response	1303	• Info (default) • Warning • Fault, ramp to coast • Fault, coast See Table 44 for descriptions of the events.	Select the behavior when a fieldbus fault, for example, loss of I/O connection occurs.
No Fieldbus Connection Response	1327	• Info (default) • Warning • Fault, ramp to coast • Fault, coast See Table 44 for descriptions of the events.	Select the response if there is no fieldbus connection.
Process Data Timeout Time	1340	0.05–18000 s (Default value: 1.00 s)	Set the timeout time. If process data is not received within the time set, a process data timeout is triggered.
Process Data Timeout Response	1341	• Info • Warning • Warning – Change Control Place • Warning – Change Control Place – Persistent • Fault, ramp to coast • Fault (default) See Table 45 for descriptions of the selections.	Select the response in case there is no fieldbus connection.
Process Data Timeout Control Place	112	• Local control (default) • Fieldbus control • I/O control • Advanced control See table Table 45 for descriptions of the control selections.	Set the control place dependency for Fieldbus Fault, Process Data Timeout. When enabled, these functions are active only in the fieldbus control place. When disabled, these functions are active regardless of the control place.

Table 44: Event Descriptions

Value	Description
Info	The event is logged in the event log.
Warning	The drive issues a warning.
Fault, ramp to coast	The drive issues a fault, ramps down, and coasts.
Fault	The drive issues a fault and coasts the motor.
Warning – Change Control Place	The drive issues a warning. The control place changes to the selected alternative while the timeout warning is active. The control place changes back to the original control place when fieldbus process data returns.
Warning – Change Control Place – Persistent	The drive issues a warning. The control place changes to the selected alternative if the timeout warning is active. The control place requires a reset command to change back to the original control place after fieldbus process data returns.

Table 45: Control Place Descriptions

Value	Description
Local control	The drive is controlled by a connected control panel.
Fieldbus control	The drive is controlled via a fieldbus.
I/O control	The drive is controlled via I/O.
Advanced control	The drive is controlled via a combination of I/O and fieldbus.

5.3 Reference Handling

The speed reference is scaled as a normalized relative value in percent (N2). The value is transmitted in hexadecimal:

- 0% = 0 hex
- 100% = 4000 hex
- -100% = C000 hex

The drive can take negative reference contributions but is limited to output only positive references as these applications typically run in a forward direction. Reverse operation can only be achieved through coil or CTW. For more information about reference handling, refer to the application guide of the application in use.

5.4 Configuring the RS-485 Interface

1. Configure the RS-485 settings.
   - In MyDrive® Insight, go to *Setup and Service* > *Interface configuration* > *RS485 Port X4* > *RS485 Settings*.
   - In the control panel, navigate to parameter group ***10.2 Communication Interfaces***.

Table 46: RS-485 Parameters

Parameter name	Value	Description
Port Address	00:247	The address of the device.
Baud Rate	9600 bps 19200 bps 38400 bps 57600 bps 76800 bps 115200 bps	Baud rate setting for the interface.
Data Frame Settings	8 data bits, no parity, 1 stop bit 8 data bits, no parity, 2 stop bits 8 data bits, odd parity, 1 stop bit 8 data bits, even parity, 2 stop bits	Select the desired frame settings, including number of data bits, parity, and number of stop bits.

5.5 Configuring Modbus Data Settings

Use MyDrive® Insight or the control panel to select persistent storage, and the byte and word order for Modbus communication.

1. Using MyDrive® Insight or the control panel, navigate to parameter group ***10.3.3.2 Configuration***.

Table 47: Parameters for Modbus Data Settings

Parameter name	Parameter number	Value	Description
Persistent Storage	7061	• Disable • Enable	Persistent Storage enables storing parameters written via Modbus in non-volatile memory. IMPORTANT: Enabling ***Persistent Storage*** causes a decrease in Modbus communication performance.
Byte Order	7062	• Little Endian • Big Endian (default)	Byte order of holding register.
Word Order	7063	• Little Endian (default) • Big Endian	Word order when mapping parameters (for example, 32-bit REAL) in multiple registers.

5.6 Mapping Device Reset Operations and Store Settings Operation

Mapping the factory restore and store operations is done in MyDrive® Insight.

Both the *Device Reset Operations* and *Store settings operation* can be mapped with a coil or a holding register and are write-only functions. Attempting to read from them results in error code 11.

The operations have the following differences in the scope and trigger values:

- *Device Reset Operations*: This operation applies to all settings except for communication settings. It is triggered by a value of 1 when mapped with a coil, or 3 when mapped with a holding register.
- *Store settings operations*: This operation applies to all settings. It is triggered by a value of 1 when mapped with either a coil or a holding register.

NOTE: Software feature does not support Bitoffet, scaling and unit conversion.

1. In MyDrive® Insight, go to *Device* > *Customization* > *Modbus* > *Holding Register* and click Add (+) in live mode.

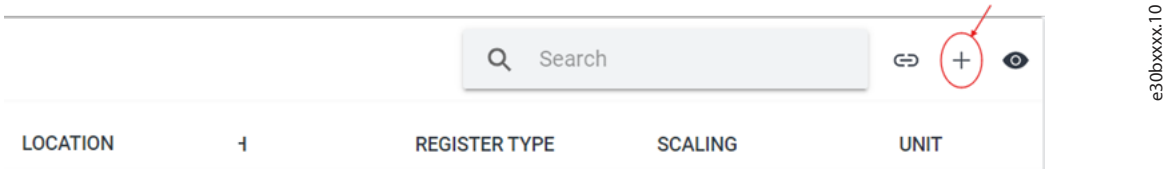


Figure 6: Adding a Holding Register

2. Select from the following options.
   - To configure the device reset operation, select the following options:
     - Reference Type: *Feature*
     - Reference: *Device Reset Operations*
   - To configure the store settings operation, select the following options:
     - Reference Type: *Feature*
     - Reference: *Store settings operation*

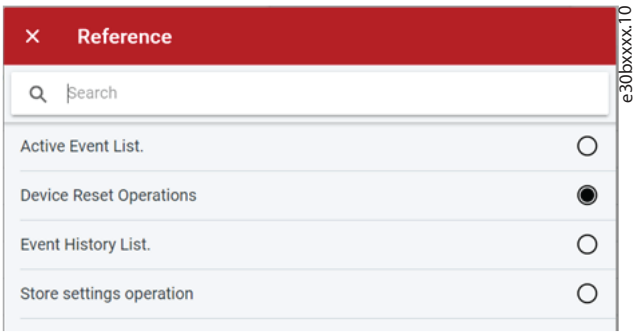


Figure 7: Selecting the Feature

5.7 Event Log

The Event Log feature can be used to read the latest event list through Modbus. Each event includes the following information:

- TimeStamp provides the relative time stamp of the time when the logger is triggered.
- Type specifies if the event is active/inactive and if the event type is Fault, Info, or Warning.
  - If the event is active, the most significant bit is set to 1. If the most significant bit is set to 0, the event is inactive.
- Code provides the code of the event occurred.

Each event consists of the TimeStamp, type, and code. See Table 48.

Table 48: Event Information

Name	Size (bits)
TimeStamp	64 bits
Type	16 bits The most significant bit: 15th bit 1: Active event 0: Inactive event The least significant bits (its 0–3) specify the event type: 100: Fault 011: Warning 010: Info The remaining bits are reserved.
Code	16 bits

5.8 Parameter Channel

5.8.1 Overview

In the Modbus parameter channel concept, each channel consists of a configuration section (including slot, parameter number, index) and a value section (the actual parameter value). These 2 parts are paired using an index number, which acts as a unique identifier for each channel instance. This pairing is managed and configured using the Customizer in MyDrive® Insight.

The parameter channel index number ensures that the configuration and value registers are logically linked, even if they are physically separated in the Modbus address space. This allows flexible grouping and organization of parameter channels based on system requirements.

The following function codes are supported by the parameter channel configuration:

1. Read multiple holding registers (03).
2. Write single holding register (06).
3. Write multiple holding registers (16).
4. Read/write multiple registers (23).

Table 49: Parameter Channel Configuration

Field name	Description	Number of registers
Slot number	Identifies the slot number of the drive or module. This information is useful for systems with multiple slots or with a modular configuration. In drives with multiple option slots, slot number 101 refers to the Basic I/O and 201 refers to functional extension options.	1
Parameter number	Specifies the unique identifier for the parameter to be accessed. This maps to the internal parameter of the drive.	1
Index	Indicates the array index for parameters that support array instances.	1
Parameter channel value	The parameter value field holds the actual data associated with the parameter defined in the setup (that is, slot, parameter number, and index). It is the source value that users read from or write to the drive.	2

There are 2 main ways to group parameter configurations and values:

- Grouping based on parameter channels: In this method each channel consumes 3 consecutive registers for configuration and 2 consecutive registers for value in a split manner. For details, see Table 50.
- Grouping based on parameter values: In this method each channel consumes 3 consecutive registers for configuration and 2 consecutive registers for value in a subsequent manner. For details, see Table 51.

Table 50: Parameter Channel Grouping

Channel number	Register address	Description	Value
Ch-01	40000	Slot number	101
	40001	Parameter number	405
	40002	Index	0
Ch-02	40003	Slot number	101
	40004	Parameter number	400
	40005	Index	0
Ch-03	40006	Slot number	101
	40007	Parameter number	402
	40008	Index	0
Ch-01	40009	Value	Value in MyDrive® Insight parameter 405
	40010		
Ch-02	40011	Value	Value in MyDrive® Insight parameter 400
	40012		
Ch-03	40013	Value	Value in MyDrive® Insight parameter 402
	40014		

Table 51: Parameter Value Grouping

Channel number	Register address	Description	Value
Ch-01	40000	Slot number	101
	40001	Parameter number	405
	40002	Index	0
	40003	Value	Value in MyDrive® Insight parameter 405
	40004		
Ch-02	40005	Slot number	101
	40006	Parameter number	400
	40007	Index	0
	40008	Value	Value in MyDrive® Insight parameter 400
	40009		

Table 51: Parameter Value Grouping - (continued)

Channel number	Register address	Description	Value
Ch-03	40010	Slot number	101
	40011	Parameter number	402
	40012	Index	0
	40013	Value	Value in MyDrive® Insight parameter 402
	40014		

5.8.2 Configuring the Parameter Channel

1. Power up the drive.
2. Launch MyDrive® Insight.

> NOTE: Configuring the parameter channel requires MyDrive® Insight version 2.20 or higher.

3. Configure the RS-485 interface. See 5.4 Configuring the RS-485 Interface.
4. Connect the drive in MyDrive® Insight and go to *Customization* > *Live* > *Modbus* > *Holding Registers* > *Add Register* > *Reference Type* > *Select Parameter Channel*.

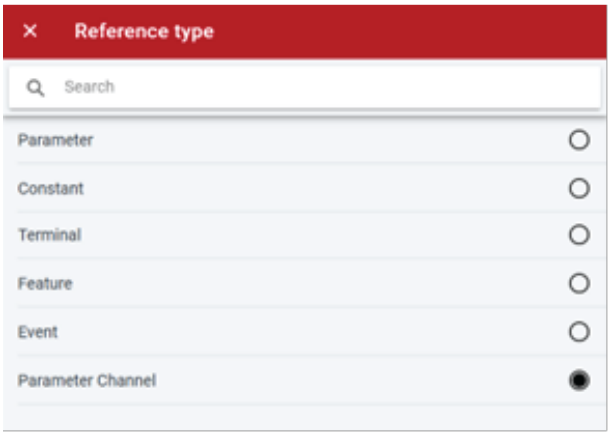


Figure 8: Selecting the Parameter Channel

5. Select *Configuration* for the channel configuration.

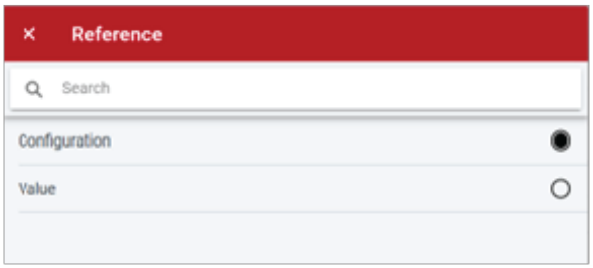


Figure 9: Selecting the Configuration

6. Select *Value* for the channel value.

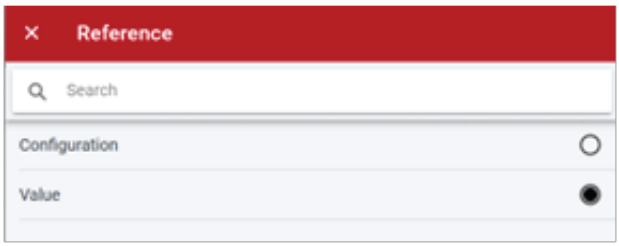


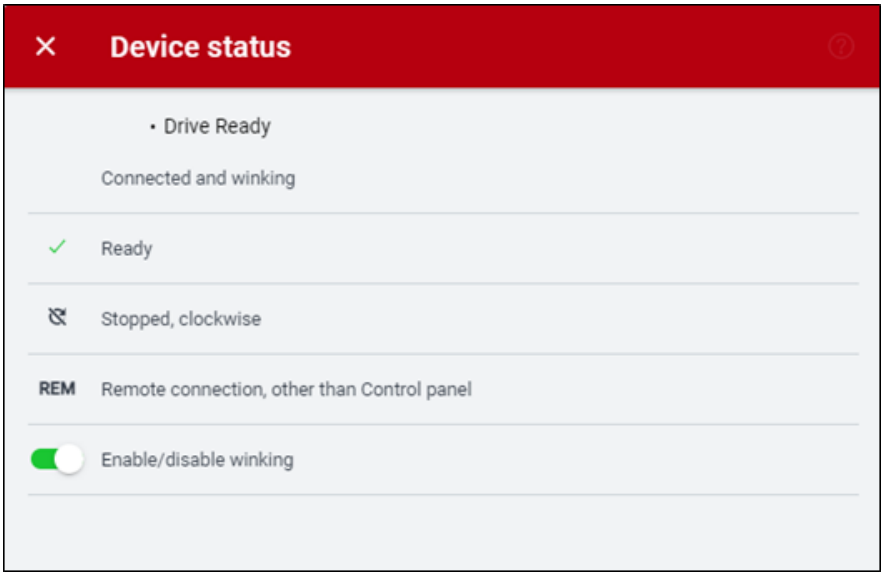
Figure 10: Selecting the Value

6 Troubleshooting

6.1 Identifying a Unit

The winking function makes the fieldbus indicator LEDs flash yellow to make it easy to identify a unit. The function is enabled in MyDrive® Insight.

1. In MyDrive® Insight, click the device name in live mode.
2. Select *Device Status*.
3. To activate or deactivate the feature, click the toggle switch.

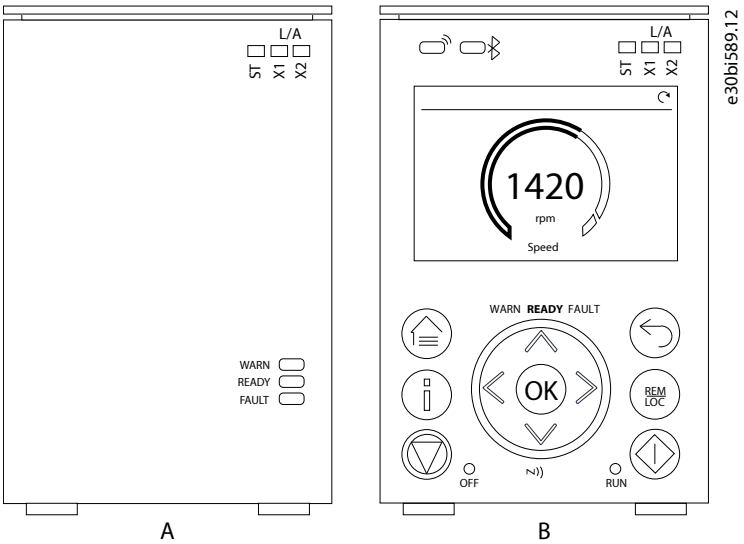


e30bk763.10

See 6.2 Fieldbus Indicator LEDs for more information on interpreting the LED signals and where the LEDs are located.

6.2 Fieldbus Indicator LEDs

There are 3 LEDs (X1, X2, and ST) on the top right corner of the control panel. The LED labeled X1 shows the network status on the RS-485 port X4 and the LED labeled ST shows the module status.



e30bi589.12

Figure 11: Fieldbus Indicator LEDs on Control Panel

Table 52: Fieldbus Indicator LED Functions

LED label	Status	LED pattern	Description
X1	Infrequent good communication	Flashing green	No active RS-485 connections.
	Communication is good	Solid green	RS-485 connection active.
	No communication	Off	There is no communication, or no serial communication on the UART port.
	Intermittent problems	Flashing red	Received more error bytes compared to valid bytes.
	Wrong configuration	Steady red	Wrong serial configuration, for example, because of a baud rate mismatch or parity mismatch.
ST	No communication	Off	There is no communication, for example, because of a broken cable.
	Infrequent valid Modbus RTU frames	Green blink	Infrequent valid Modbus RTU frames received.
	Frequent valid Modbus RTU frames.	Green	Frequent valid Modbus RTU frames received.
	Invalid Modbus RTU request	Flashing red	Invalid Modbus request, for example, because the entity does not exist, the function code is not supported, or access is denied.
	Dropped Modbus RTU frames	Steady red	Dropped Modbus frames, for example, because of a CRC error, invalid length, or an RTU framing error.

6.3 Diagnostic Counters

The Modbus RTU diagnostic counters are used to monitor and diagnose RTU communication. The limit of each counter is 4.294.967.295 (maximum value of data type uint32_t). If the counter exceeds the limit, it restarts from zero. No information about the counter completing a cycle is recorded and no notifications are sent. The counters are reset when the drive restarts, or a warm start or cold start is executed.

The counters monitor the status on the UART level and not in the module level.

Table 53: UART Diagnostic Counters

Counter name	Description
RxByteCounter	Total count of all received bytes
TxByteCounter	Total count of all transmitted bytes.
RxDataFormatErrorCounter	Total count of all received data format error bytes.
RxFifoOverflowCounter	Total count of all received overflow bytes.

Danfoss A/S
Ulsnaes 1
DK-6300 Graasten
drives.danfoss.com

M00458
AQ497328073284en-010101
Danfoss A/S © 2026.04

136R0367