LOW VOLTAGE SYSTEM

LVS Digital

Modbus RTU/TCP Converter MS572

User Manual

NOTICE

This document contains information about one or more ABB products and may include a description of or a reference to one or more standards that may be generally relevant to the ABB products. The presence of any such description of a standard or reference to a standard is not a representation that all the ABB products referenced in this document support all the features of the described or referenced standard. In order to determine the specific features supported by an ABB product, the reader should consult the product specifications for the ABB product.

ABB may have one or more patents or pending patent applications protecting the intellectual property in the ABB products described in this document.

The information in this document is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this document.

Products described or referenced in this document are designed to be connected and to communicate information and data through network interfaces, which should be connected to a secure network. It is the sole responsibility of the system/product owner to provide and continuously ensure a secure connection between the product and the system network and/or any other networks that may be connected.

The system/product owners must establish and maintain appropriate measures, including, but not limited to, the installation of firewalls, application of authentication measures, encryption of data, installation of antivirus programs, and so on, to protect these products, the network, its system, and interfaces against security breaches, unauthorized access, interference, intrusion, leakage, and/or theft of data or information.

ABB performs functionality testing on the products and updates that we release. However, system/product owners are ultimately responsible for ensuring that any product updates or other major system updates (to include but not limited to code changes, configuration file changes, third-party software updates or patches, hardware change out, and so on) are compatible with the security measures implemented. The system/ product owners must verify that the system and associated products function as expected in the environment in which they are deployed.

In no event shall ABB be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without written permission from ABB, and the contents thereof must not be imparted to a third party nor used for any unauthorized purpose.

The software or hardware described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license. This product meets the requirements specified in EMC Directive 2014/30/EU and in Low Voltage Directive 2014/35/EU.

TRADEMARKS

MNS is a registered trademark of ABB.

NeoGear is a registered trademark of ABB.

Ability is a trademark of ABB.

All rights to copyrights, registered trademarks, and trademarks reside with their respective owners.

Copyright © 2022 ABB.

All rights reserved.

Release: January 2022

Document Number: 1TNC928243

Revision: M0203

Table of Contents

List of Figures

List of Tables

1 General

1.1 Target Group

The manual is primarily intended for those requiring information on the applications of MS572 for the purpose of understanding, engineering, wiring & operating the product

The objective of this manual is to provide the technical functions description of MS572. It is assumed that the user has a basic knowledge of physical and electrical fundamentals, electrical wiring practices and electrical components

1.2 Use of Warning, Caution, Information and Tip icon

This publication includes **Warning, Caution**, and **Information** icons where appropriate to point out safety related or other important information. It also includes **Tip** icons to point out useful hints to the reader. The corresponding symbols should be interpreted as follows:

The electrical warning icon indicates the presence of a hazard that could result in *electrical shock*.

The warning icon indicates the presence of a hazard that could result in *personal injury*

The caution icon indicates important information or warnings related to the concept discussed in the text. It might indicate the presence of a hazard that could result in *corruption of software or damage to equipment/property*

The information icon alerts the reader to pertinent facts and conditions

The tip icon indicates advice on, for example, how to design your project or how to use a certain function

Although **Warning** notices are related to personal injury, and **Caution** notices are associated with equipment or property damage, the operation of damaged equipment could, under certain operational conditions, result in impaired process performance leading to personal injury or death. It is, therefore, imperative that you comply fully with all **Warning** and **Caution** notices.

1.3 Terminology

List of the terms, acronyms, abbreviations, and definitions that the document uses.

Abbreviation	Term	Description
LVS	low-voltage switchgear	A factory-built assembly built to conform with IEC 61439-1
LVS Digital		The digital solution based on low-voltage switchgear
CMES		ABB Ability™ Condition Monitoring for electrical systems
MV570		The HMI used for CN market only
	MODBUS RTU	Fieldbus communication protocol based on serial communication layer
	MODBUS TCP/IP	Fieldbus communication protocol based on Ethernet
	Real time clock	Integrated clock function in devices used to generate time and date information if a remote clock system is not present
	Trip	A consequence of an alarm activated or an external trip command from another device to stop the motor or trip the circuit breaker.
	Alarm	Alarm is defined as status transition from any state to abnormal state. Status transition to abnormal state can be data crossing over the predefined alarm limit.
	Event	An event is a status transition from one state to another. It can be defined as alarm, if the state is defined as abnormal or as warning as a pre-alarm state.
RS485		Communication interface standard from EIA (Electronics Industries Association, USA), operating on voltages between 0V and +5V. RS-485 is more noise resistant than RS-232C, handles data transmission over longer distances, and can drive more receivers.
MRP		Media redundancy protocol

Abbreviation	Term	Description
MRC		Media redundancy client
MRM		Media redundancy manager
Client/ Server		The Modbus TCP messaging service provides a Client/Server communication between devices connected on an Ethernet TCP/IP network. The device initiating the communication (e.g. a PLC) is called the client. The device answering the request is called the server.
MAC		Medium Access Control
MAC Address		Unique address of every Ethernet device. The MAC address of the MS572 is printed on the device directly

1.4 Related Documentation

- 1TGC 908001 ABB Ability Condition Monitoring for Electrical System-CMES User Manual

1.5 Related System Version

The content of this document is related to MS572 products with the following hardware and firmware version release.

	Hardware version(HW)	Firmware version(FW)
MS572	2.0	2.0

Until further notice, this document is also applicable for future firmware versions other than those listed above.

The described functions are designed but may not be fully implemented in all details. Please refer to the release notes regarding possible restrictions.

2 Product Overview

Ethernet converter MS572 provides protocol conversion between Modbus TCP and Modbus RTU.

2.1 Dimension

Dimension W x H x D = 92 mm X 26 mm X 57 mm

Typical installation is DIN rail mounting as per figure 1.

Installation and maintenance must be performed according to the technical rules, codes and relevant standards and carried out by skilled electricians only.

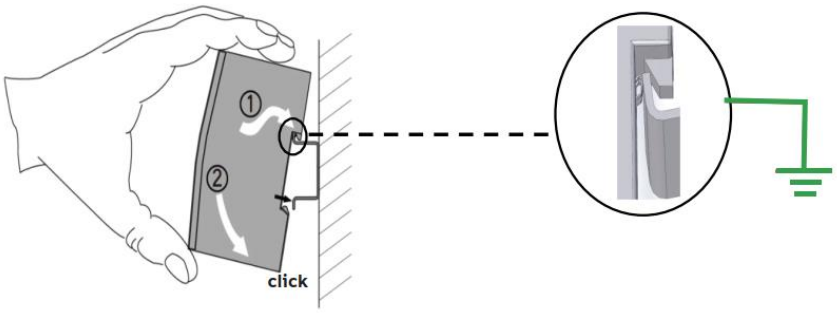

Figure 1: DIN rail installation of MS572

For installation details of MS572, see the related documentation in installation manual.

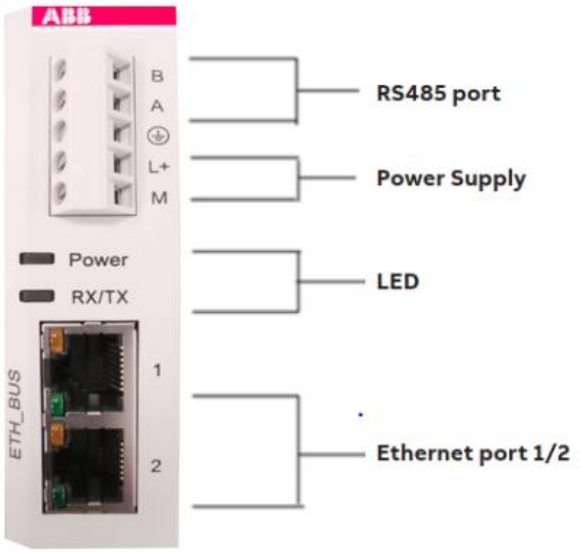

Figure 2: MS572 Terminal designation

2.2 Power supply

Power supply of the device should be always derived from uninterrupted and reliable supply source. (Figure 2)

Terminal No.	Name	Description
L+	24VDC	24VDC+
M	GND	0VDC

2.3 Fieldbus interface

MS572 supports Modbus RTU protocol which is based on physical RS485 layer. (Figure 2)

.

Terminal No.	Name	Description
FE	FE	Function earth for RS485 shielding
A	A	Serial RS485 A (-)
B	B	Serial RS485 B (+)

2.4 Ethernet communication interface

Two identical Ethernet interfaces are provided on MS572. Only one set of IP address is supported. The connection is through RJ45 connector. (Figure 2)

Terminal No.	Name	Description
	ETH1	Ethernet interface 1 with indication
	ETH2	Ethernet interface 2 with indication

Ethernet Indication	Description
Orange	10 Mbps Ethernet connection
Green	100 Mbps Ethernet connection
Off	Ethernet cable is disconnected, or short-circuit

2.5 LED Indicators

(Refer to Figure 2)

ED Indication		Description
PWR	Green	Power on
TX/RX	Orange	Serial port is receiving data
	Green	Serial port is transmitting data
	Off	No data is being transmitted or received through the serial port

3 Ethernet communication

3.1 MODBUS TCP

Modbus TCP is supported by MS572. Upper-level devices such as PLC can read/write directly to MS572's data registers or read/write directly to lower-level RS485 devices' data registers.

Each device includes an embedded 2-port Ethernet switch, which permits flexible, inexpensive network design. The characteristics for connection to the network are as follows:

Designation	Description
Type of cable	Straight or crossed category 5 shielded twisted pair
Maximum cable length (daisy chain or daisy chain loop in ring)	100 m (328 ft)
Maximum number of devices per subnet	160
Maximum number of devices per segment	up to 50 devices in a daisy chain or daisy chain loop (ring)

Table 1 Network characteristics

Up to 4 Modbus TCP clients for Modbus TCP connections with MS572

3.2 Supported network

Possible network topology supported by MS572 include,
• Star topology
• Bus topology
• Ring topology (via a managed switch that sup- ports MRP configurations)
MS572 is compatible with IEEE 802.3 standard.
.

Star topology

In a star topology, each MS572 must connect to a switch via RJ45 as follows:

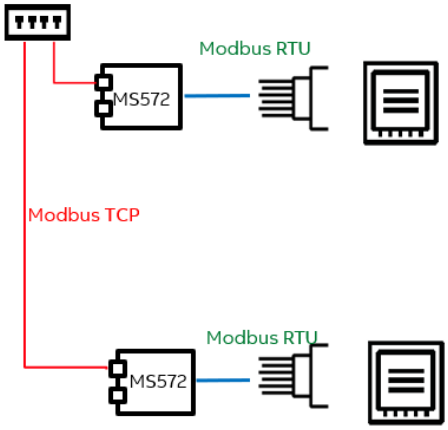

Figure 3: Star connection

Bus topology

Two RJ45 ethernet ports of MS572 are used to connect between MS572 interfaces or between MS572 and a switch. The bus connection is illustrated as below,

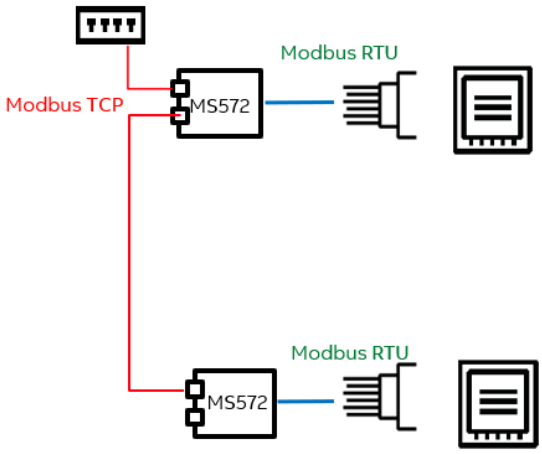

Figure 4: Bus connection

Ring topology with network redundancy

A ring topology is a daisy chain where the last controller in the chain loops back to the managed switch. The ring topology offers redundancy, i.e. if the network fails at any single point, the network on either side of the failure point continue to operate as separate networks.

MS572 supports MRP protocol. MRP standard defines two principal device roles in a MRP network. The MRP manager, which is typically a managed network switch, and MRP clients which are typically automation devices like MS572. The MRP master sends out test telegrams cyclically to check the health status of the network. If everything is ok it blocks telegrams on one side of its internal switch to avoid loops (left side of the next figure). If somewhere in the network a fault is detected the MRP master reorganizes the network and closes its internal switch. So, all network nodes are still accessible.

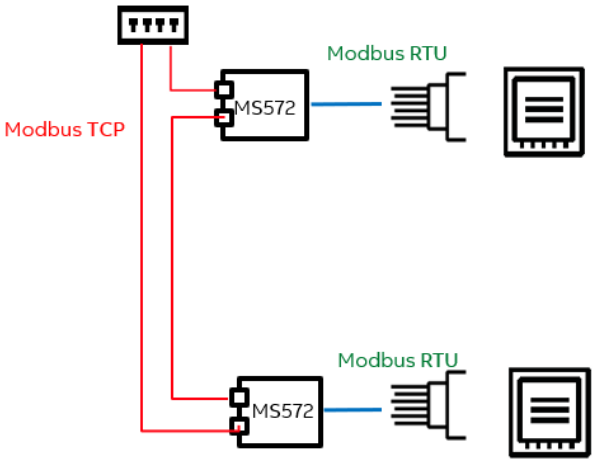

Figure 5: MRP Ring connection

3.3 Protocol implementation

Data Frame Format and Data Rate

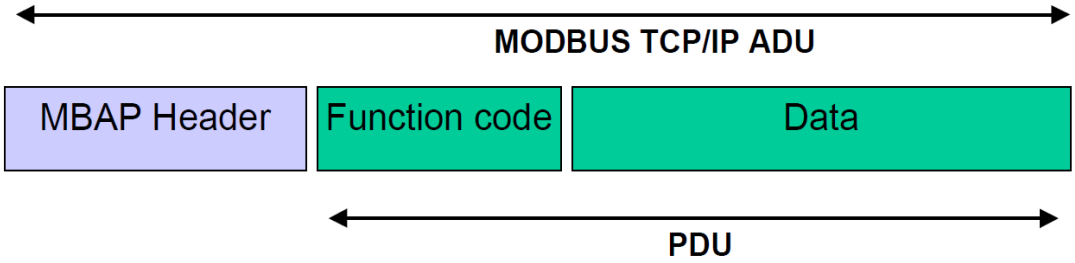

Figure 6: Data Frame Format

Modbus TCP protocol can be implemented at 10/100M communication speed.

A complete request/response sequence consists of the following bytes (transmitted as separate data frames):

Master Request Transmission:
MBAP Header: 7 bytes
Function Code: 1 byte
Data: variable number of bytes depending on function code

Slave Response Transmission:
MBAP Header: 7 bytes
Function Code: 1 byte
Data: variable number of bytes depending on function code

MBAP header:
The MBAP Header is 7 bytes long as follows,

Fields	Length	Description	Client	Server
Transaction Identifier	2 Bytes	Identification of a MODBUS request/ response transaction	Initialized by the client	Recopied by the server from the received request
Protocol Identifier	2 Bytes	MODBUS protocol	Initialized by the client	Recopied by the server from the received request
Length	2 Bytes	Number of following bytes	Number of following bytes	Number of following bytes
Unit Identifier	1 Bytes	Identification of a remote slave connected on a serial line or on other buses	Initialized by the client	Recopied by the server from the received request

Table 2 MBAP header

Transaction identifier - It is used for transaction pairing; the MODBUS server copies in the response the transaction identifier of the request.

Protocol identifier – It is used for intersystem multiplexing. The MODBUS protocol is identified by the value 0.

Length - The length field is a byte count of the following fields, including the Unit Identifier and data fields.

Unit identifier –This field is used for intra-system routing purpose. It is typically used to communicate to a MODBUS or a MODBUS+ serial line slave through a gateway between an Ethernet TCP-IP network and a MODBUS serial line. This field is set by the MODBUS Client in the request and must be returned with the same value in the response by the server. On TCP/IP, the MODBUS server is addressed using its IP address; therefore, the MODBUS Unit Identifier is useless. The value 0xFF must be used.

Modbus TCP default port is port: 502.

Function code: This is the second byte of every transmission. Modbus defines function codes of 1 to 127. MODBUS Gateway support function code 02H, 03H, 04H, 05H, 06H, and 10H. In a master request transmission, the function code tells the slave what action to perform. In a slave response transmission, the function code tells the master what function was performed as requested. If the high order bit of the function code sent from the slave is a 1 (i.e. if the FUNCTION CODE is >127) then the slave did not perform the function as requested and is sending an error or exception response.

Data: This will be a variable number of bytes depending on the function code. This may be Actual Values, Settings, or addresses sent by the master to the slave or by the slave to the master.

Timing: Follow the standard TCP protocol.

Function description

The following functions are supported in MODBUS GW Modbus TCP Implementation.

- FUNCTION CODE 02H - Read Actual Values
- FUNCTION CODE 03H - Read Parameters and Actual Values
- FUNCTION CODE 04H - Read Parameters and Actual Values
- FUNCTION CODE 05H - Execute Operation
- FUNCTION CODE 06H - Store Single parameter
- FUNCTION CODE 10H - Store Multiple parameters

Function code 02H

Standard Modbus Implementation: Read Input Status
MODBUS GW Modbus implementation: Read Actual Values

For MODBUS GW implementation of Modbus, this command can be used to read any actual values ("input registers"). Registers are 16-bit (two byte) values transmitted high order byte first. Thus, all actual values are sent in two bytes. The maximum number of inputs that can be read in one transmission is 2000 (125 registers).

The slave response to this function code is the MBAP Header, function code, a count of the number of data bytes to follow, the data itself. Each data item is sent as a two-byte number with the high order byte sent first.

For example, to request a slave to respond with 1 register starting at address 0x0050. In this example, the value in this register (0x0050) is 0x0001.

MASTER TRANSMISSION	BYTES	EXAMPLE(Hex)	DESCRIPTION
MBAP HEADER	7	00 00 00 00 00 06 FF	Modbus application protocol header
FUNCTON CODE	1	02	Read inputs
DATA STARTING ADDRESS (Starting address must be the multiple of 0x10)	2	05 00	No. 0x0500 input=No.0x0050 register Data starting at 0x0050
NUMBER OF ACTUAL VALUES (Number must be the multiple of 0x10)	2	00 10	0x10 inputs = 2 bytes total

SLAVE RESPONSE	BYTES	EXAMPLE(Hex)	DESCRIPTION
MBAP HEADER	7	00 00 00 00 00 05 FF	Modbus application protocol header
FUNCTON CODE	1	02	Read inputs
BYTE COUNT	1	02	1 registers = 2 bytes total
DATA	2	00 01	Value in address 0x0050

Function code 03H

Standard Modbus implementation: Read Holding Registers
MODBUS GW Modbus implementation: Read Parameters and Actual Values

For MODBUS GW implementation of Modbus, this command can be used to read any parameters ("holding registers") or actual values ("input registers"). Holding and input registers are

16-bit (two byte) values transmitted high order byte first. Thus, all parameters and actual values are sent in two bytes. The maximum number of registers that can be read in one transmission is 125. This function code is identical to function code 04.

The slave response to this function code is the MBAP header, function code, a count of the number of data bytes to follow, the data itself. Each data item is sent as a two-byte number with the high order byte sent first.

For example, to request a slave to respond with 1 register starting at address 0x1021. In this example the value in this register (0x1021) is 0x0023.

MASTER TRANSMISSION	BYTES	EXAMPLE(Hex)	DESCRIPTION
MBAP HEADER	7	00 00 00 00 00 06 FF	Modbus application protocol header
FUNCTON CODE	1	03	Read registers
DATA STARTING ADDRESS	2	10 21	Data starting at 0x1021
NUMBER OF ACTUAL VALUES	2	00 01	1 registers = 2 bytes total

SLAVE RESPONSE	BYTES	EXAMPLE(Hex)	DESCRIPTION
MBAP HEADER	7	00 00 00 00 00 05 FF	Modbus application protocol header
FUNCTON CODE	1	03	Read registers
BYTE COUNT	1	02	1 registers = 2 bytes total
DATA	2	00 23	Value in address 0x1021

Function code 04H

Standard Modbus Implementation: Read Input Registers
MODBUS GW Modbus implementation: Read Parameters and Actual Values

For the MODBUS GW implementation of Modbus, this command can be used to read any Parameters ("holding registers") or actual values ("input registers"). Holding and input registers are 16-bit (two byte) values transmitted high order byte first. Thus, all parameters and actual values are sent in two bytes. The maximum number of registers that can be read in one transmission is 125. This function code is identical to function code 03H.

The slave response to this function code is the MBAP header, function code, a count of the data bytes to follow, the data itself. Each data item is sent as a two bytes number with the high order byte sent first.

For example, consider a request for a slave to respond with 3 registers starting at address 0x0026. In this example the register data in these addresses is shown below:

Address(Hex)	Data(Hex)
0026	00DB
0027	00DA
0028	00DB

The master/slave packet format is shown below:

MASTER TRANSMISSION	BYTES	EXAMPLE(Hex)	DESCRIPTION
MBAP HEADER	7	00 00 00 00 00 06 FF	Modbus application protocol header
FUNCTON CODE	1	04	Read registers for line voltages
DATA STARTING ADDRESS	2	00 26	Data starting at 0x0026
NUMBER OF REGISTERS	2	00 03	3 registers = 6 bytes total

SLAVE RESPONSE	BYTES	EXAMPLE(Hex)	DESCRIPTION
MBAP HEADER	7	00 00 00 00 00 09 FF	Modbus application protocol header
FUNCTON CODE	1	04	Read registers for line voltage
BYTE COUNT	1	06	3 registers = 6 bytes
DATA 1 (see definition above)	2	00 DB	Value in address 0x0026
DATA 2 (see definition above)	2	00 DA	Value in address 0x0027
DATA 3 (see definition above)	2	00 DB	Value in address 0x0028

Function code 05H

Standard Modbus Implementation: Force Single Coil
MODBUS GW Modbus implementation: Execute Operation

This function code allows the master to request a slave device to perform specific command operations. The commands supported are listed in F8.

For example, to request a slave to execute FDR data backup, we have the following master/slave packet format:

MASTER TRANSMISSION	BYTES	EXAMPLE(Hex)	DESCRIPTION
MBAP HEADER	7	00 00 00 00 00 06 FF	Modbus application protocol header
FUNCTON CODE	1	05	Execute operation
MODE CODE	2	00 02	FDR data backup
OPERATION CODE	2	FF 00	Execute start

SLAVE RESPONSE	BYTES	EXAMPLE(Hex)	DESCRIPTION
MBAP HEADER	7	00 00 00 00 00 06 FF	Modbus application protocol header
FUNCTON CODE	1	05	Execute operation
MODE CODE	2	00 02	FDR data backup
OPERATION CODE	2	FF 00	Execute start

Commands can be performed using function code 10H as well as function code 05H. The details refer to the description of function code 10H.

Function code 06H

Standard Modbus Implementation: Preset Single Register
MODBUS GW Modbus implementation: Store Single Parameter

This command allows the master to store a single parameter into the memory. The slave response to this function code is to echo the entire master transmission.

For example, request a slave to store the value 0x017C in parameter address 0x1027. After the transmission in this example is complete, parameter address 0x1027 will contain the value 0x017C.

The master/slave packet format is shown below:

MASTER TRANSMISSION	BYTES	EXAMPLE(Hex)	DESCRIPTION
MBAP HEADER	7	00 00 00 00 00 06 FF	Modbus application protocol header
FUNCTON CODE	1	06	Store single parameter
DATA STARTING ADDRESS	2	10 27	parameter address 0x1027
DATA	2	01 7C	Data stored in parameter address 0x1027

SLAVE RESPONSE	BYTES	EXAMPLE(Hex)	DESCRIPTION
MBAP HEADER	7	00 00 00 00 00 06 FF	Modbus application protocol header
FUNCTON CODE	1	06	Store single parameter
DATA STARTING ADDRESS	2	10 27	Parameter address 0x1027
DATA	2	01 7C	Data stored in parameter address 0x1027

Function code 10H

Standard Modbus Implementation: Preset Multiple Registers
MODBUS GW Implementation: Store Multiple Parameters

This function code allows multiple Parameters to be stored into the memory. Modbus "registers" are 16-bit (two byte) values transmitted high order byte first. Thus, all Parameters are sent as two bytes. The maximum number of Parameters that can be stored in one transmission is dependent on the slave device. Modbus allows up to a maximum of 123 holding registers to be stored. The slave device response to this function code is to echo the MBAP header, function code, starting address, the number of Parameters stored.

For example, consider a request for a slave to store the value 0x0190 to parameter address 0x1036 and the value 0x0005 to parameter address 0x1037. After the transmission in this example is complete, slave will have the following information stored:

Address(Hex)	Data(Hex)
1036	0190
1037	0005

The master/slave packet format is shown below:

MASTER TRANSMISSION	BYTES	EXAMPLE(Hex)	DESCRIPTION
MBAP HEADER	7	00 00 00 00 00 0B FF	Modbus application protocol header
FUNCTON CODE	1	10	Store parameters
DATA STARTING ADDRESS	2	10 36	Parameter address 0x1036
NUMBER OF REGISTER	2	00 02	2 parameters = 4 bytes total
BYTE OF COUNT	1	04	4 bytes of data
DATA 1	2	01 90	Data for parameter address 0x1036
DATA 2	2	00 05	Data for parameter address 0x1037

SLAVE RESPONSE	BYTES	EXAMPLE(Hex)	DESCRIPTION
MBAP HEADER	7	00 00 00 00 00 06 FF	Modbus application protocol header
FUNCTON CODE	1	10	Store parameters
DATA STARTING ADDRESS	2	10 36	register address 0x1036
NUMBER OF REGISTER	2	00 02	2 parameters

Commands can be performed using function code 10H as well as function code 05H. When using function code 10H, the command function code register (address 0x5000) must be written with fixed value of 0x0005 (DATA1). The command operation code register (address 0x5001) must be written with valid data (DATA2), this is dependent upon the command operation. (The commands supported are listed in F8).
For example, to request a slave to execute FDR data backup, we have the following master/slave packet format:

MASTER TRANSMISSION	BYTES	EXAMPLE(Hex)	DESCRIPTION
MBAP HEADER	7	00 00 00 00 00 0B FF	Modbus application protocol header
FUNCTON CODE	1	10	Store parameters
DATA STARTING ADDRESS	2	50 00	Parameter address 0x5000
NUMBER OF REGISTER	2	00 02	2 parameters = 4 bytes total
BYTE OF COUNT	1	04	4 bytes of data
DATA 1	2	00 05	Data for parameter address 0x5000
DATA 2	2	00 02	Data for parameter address 0x5001

SLAVE RESPONSE	BYTES	EXAMPLE(Hex)	DESCRIPTION
MBAP HEADER	7	00 00 00 00 00 06 FF	Modbus application protocol header
FUNCTON CODE	1	10	Store parameters
DATA STARTING ADDRESS	2	50 00	register address 0x5000
NUMBER OF REGISTER	2	00 02	2 parameters

Error responses description

When a MODBUS GW device detects an error other than a CRC error, a response will be sent to the master. The MS Bit of the FUNCTION CODE byte will be set to 1 (i.e. the function code sent from the slave will be equal to the function code sent from the master plus 128). The following byte will be an exception code indicating the type of error that occurred.

The slave response to an error (other than CRC error) will be:
MBAP HEADER: 7 byte
FUNCTION CODE: 1 byte (with MS bit set to 1)
EXCEPTION CODE: 1 byte

The device implements the following exception response codes.

- 01 - ILLEGAL FUNCTION

The function code transmitted is not one of the functions supported.

- 02 - ILLEGAL DATA ADDRESS

The address referenced in the data field transmitted by the master is not listed in the memory map..

- 03 - ILLEGAL DATA VALUE

The value referenced in the data field transmitted by the master is not within range for the selected data address.

- 0B – Gateway problem

The targeted device failed to response. The gateway generates this exception.

4 Modbus RTU network

The MS572 has one RS485 interface to connect to the field devices. Through the RS485 interface, it supports the following functions: parameter setting, control, monitoring and so on.

MS572 implements a subset of Modicon Modbus RTU serial communication standard. Modbus is a single host/multi-slave type protocol for multipoint configuration, supported by the RS485 interface. MS572 is the master of Modbus RTU, and all other devices are connected as slave devices of bus structure.

There is no built-in termination resistor in the Modbus RTU port of MS572. It is necessary to install 120 ohms termination resistor at the beginning and the end of the communication section in project application. At the beginning and end of each Modbus RTU communication section, the cable needs to receive termination resistor.

The max. communication cable length within a section depends on cable type and selected communication speed

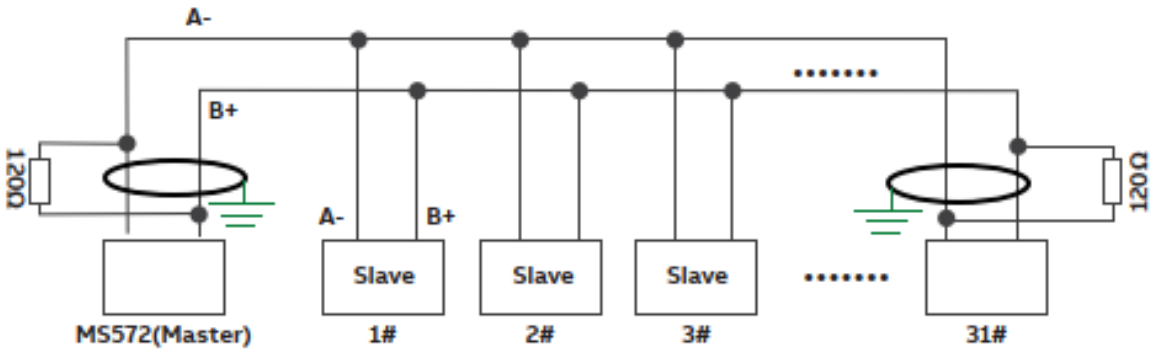

Figure 7: Modbus RTU connection

A shielded cable with twisted pairs shall be used for the installation to reduce the impact of disturbances to a minimum.

For time critical applications, the number of devices connected to each MS572 is recommended not to exceed 10 to ensure fast communication response.

5 Parameterization

In application, it is necessary to set up the communication parameters of MS572 to ensure the normal communication. Users can set parameters for MS572 through Windows® based MConfig-G software

5.1.1 Parameterization via MConfig-G

MConfig-G software is a single device configuration tool which is installed on a laptop. There are 2 ways to access MS572 for parameterization

- Access through a network switch when both MS572 and the laptop with MConfig-G are connected to the same switch.
- Connect the laptop with MConfig-G directly to the Ethernet port of MS572

Both direct and cross cable types are applicable.

5.1.2 Ethernet parameters

Parameters	Setting Range	Factory Setting
Ethernet IP address setting	0.0.0.0...255.255.255.255	Default = 192.168.1.118
Ethernet subnet mask setting	0.0.0.0...255.255.255.255	Default = 255.255.255.0
Ethernet gateway address setting	0.0.0.0...255.255.255.255	Default = 192.168.1.1
Network port fault enable	Enable/Disable	Disable
Network port warning enable	Enable/Disable	Disable
Network port FDR Enable/Disable	Enable/Disable	Disable
Network port FDR auto backup enable	Enable/Disable	Disable
DHCP Function	Disable	Disable
Network port communication loss timeout	0...6553 s in increments of 0.1 s	60 s
Ethernet SNTP Server IP address setting	0.0.0.0...255.255.255.255	default = 0.0.0.0
SNTP cyclic update interval	0…65535 min	30 min
SNTP time shift	-12*60 ~ +14*60 min	480 min
Inter frame delay *1)	0-1000ms	0
Network port 1 warning enable	Enable/Disable	Disable

Parameters	Setting Range	Factory Setting
Network port 2 warning enable	Enable/Disable	Disable
Modbus TCP Keepalive time	0-300s	180s

Table 3 Ethernet parameters

Modbus TCP keepalive time means MS572 will keep Modbus TCP communication channel alive during setting time when communication failure between MS572 and one TCP server.

5.1.3 Modbus RTU parameters

Parameters	Setting Range	Factory Setting
Baud Rate	1200...115200 Baud	115200
Parity	0=None 1=Odd 2=Even	Even
RTU timeout response OFF*2)	Enable Disable	Disable
RTU response timeout type *3)	Default Expert	Default
RTU response timeout* 4)	20-1000ms	-
Inter frame delay	0-500ms	0

Table 4 Modbus RTU parameters

1) Inter frame delay is to define the interval between MS572 receives the response of one RTU slave and sends new RTU request.
2) When select 'Enable' RTU timeout response OFF, no response/message is sent to inform master station via Modbus TCP if any of the RTU devices fail to respond within the timeout.
3) RTU timeout is adjustable only when the 'Expert' mode is selected for RTU timeout type. RTU response timeout is self-adjusted based on baud rate setting when 'Default' type is selected.
4) RTU timeout is adjustable when 'Expert' mode is selected for RTU timeout type.

5.1.4 Ethernet service

Ethernet services are selectable through MConfig-G software. Pull down menu "  " > "Ethernet Service" , following widget for Ethernet Service will appear.

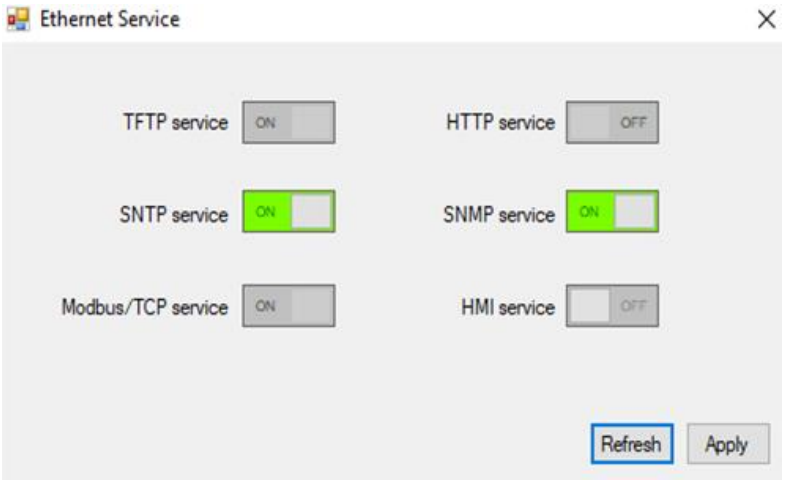

Figure 8: Ethernet service configuration window

TFTP service- Trivial File Transfer Protocol

TFTP is a protocol in TCP/IP protocol family used for simple file transfer between client and server. It provides a simple file transfer service with little complexity and low overhead.

TFTP service is used for MS572 firmware upgrading via ISP hence always switched on.

SNTP service- Simple Network Time Protocol

SNTP is used to synchronize the server clock in the internet. The calibration time is obtained from the SNTP server according to the time set by the SNTP cycle update interval, and then the time is calibrated according to the SNTP time shift.

SNTP service is switched ON by default. If SNTP service is not required in project, SNTP service could be switch OFF.

Modbus/TCP

Modbus/TCP is an industrial ethernet protocol supported by MS572 module. Each MS572 module has two ethernet ports which support redundant communication. The function of parameter setting, control and monitoring can be realized through the ethernet interface.

HTTP service- Hyper Text Transfer Protocol

HTTP service is used for communication between web clients and web servers. One of the important applications of HTTP is WWW(World Wide Web) service.

HTTP service setting is turned OFF in the device and not configurable in current version.

HMI service- A service to make the device visible on Ethernet network in order to build up the connection with an integrated platform, e.g. HMI.

When HMI service is selected, the MS572 starts to send out UDP telegram to the TCP network with the intention to build up the communication with a central integration platform, e.g. the HMI device MV570.

For a network without a connected platform, the HMI service must be switched OFF to avoid excessive telegram generated from the device resulting in network performance degradation.

6 Appendix A: Register map

Register Map (Modbus TCP)

The data stored is grouped into three areas: configurable parameters, monitoring values, command register area. Configurable parameters can be read and written by a Modbus/TCP master. Monitoring values can be read only. Command registers can be written only. Each register is the address of two-byte value.

Configurable Parameters

Group	Register Address (Hex)	Value Name	Data Type/ Format	Step	Unit	Value Range (Default setting)
Communication Option	1000	Parity Check	F5	1		0-2(2)
	1001	Modbus Baud Rate	F6	1		1-8(8)
	1002	…Reserved…				
	1003	…Reserved…				
	1004	…Reserved…				
	1005	RTU timeout mode	F15	1		0-1(0)
	1006	RTU timeout			ms	20-1000
	1007	RTU timeout response OFF	F7	1		0-1(0)
	1008	Inter frame delay		10	ms	0-500(0)
	1009	…Reserved…				
	100A	…Reserved…				
	100B	…Reserved…				
	100C	…Reserved…				
	100D	…Reserved…				
	100E	…Reserved…				
	100F	…Reserved…				
	1011	…Reserved…				
	1012	…Reserved…				
	1013	…Reserved…				

Group	Register Address (Hex)	Value Name	Data Type/ Format	Step	Unit	Value Range (Default setting)
	1014	…Reserved…				
Ethernet Setting	1015	Ethernet IP address setting byte1-2	UWORD			0-65535 (0)
	1016	Ethernet IP address setting byte3-4	UWORD			0-65535 (0)
	1017	Ethernet subnet mask setting byte1-2	UWORD			0-65535 (65535)
	1018	Ethernet subnet mask setting byte3-4	UWORD			0-65535 (65280)
	1019	Ethernet gateway address setting byte1-2	UWORD			0-65535 (0)
	101A	Ethernet gateway address setting byte3-4	UWORD			0-65535 (0)
	101B	…Reserved…				
	101C	…Reserved…				
	101D	…Reserved…				
	101E	…Reserved…				
	101F	…Reserved…				
	1020	…Reserved…				
	1021-1030	…Reserved…				
	1031-1040	…Reserved…				
	1041-1050	…Reserved…				
	1051-1058	…Reserved…				
	1059-1060	…Reserved…				
	1061-1068	…Reserved…				
	1069	Reserved				
	106A	Network port alarm enable	F7	1		0-1 (0)

Group	Register Address (Hex)	Value Name	Data Type/ Format	Step	Unit	Value Range (Default setting)
	106B	Reserved				
	106C	Network port FDR Enable/Disable	F7	1		0-1 (0)
	106D	Network port FDR auto backup enable		1		0=No synchro 1=Auto backup (0)
	106E	DHCP Function	F7	1		0
	106F	Network Port Timeout		1	×0.1s	0-65535 (600)
	1070	Ethernet SNTP Server IP address setting byte1-2	UWORD			0-65535 (0)
	1071	Ethernet SNTP Server IP address setting byte3-4	UWORD			0-65535 (0)
	1072	SNTP cyclic update interval	UWORD	1	min	1-65535 (30)
	1073	SNTP time shift	WORD	1	min	-12*60~ +14*60 (8*60)
	1074	Network port1 alarm enable	F7	1		0-1 (0)
	1075	Network port2 alarm enable	F7	1		0-1 (0)
	1076	Modbus TCP keepalive time	WORD	**1**	**s**	0-300(180)
	1077-107F	…Reserved…				
USER DEF. MEM. MAP DATA	1080	Register address-User Definable Data 0000	UWORD	1		0000-107F
USER DEF. MEM. MAP DATA	1081	Register address-User Definable Data 0001	UWORD	1		0000-107F

Group	Register Address (Hex)	Value Name	Data Type/ Format	Step	Unit	Value Range (Default setting)
	1082	Register address-User Definable Data 0002	UWORD	1		0000-107F
	1083	Register address-User Definable Data 0003	UWORD	1		0000-107F
	↓	↓	↓	↓	↓	↓
	10CF	Register address-User Definable Data 004F	UWORD	1		0000-107F

Monitoring values

Group	Register Address (Hex)	Value Name	Data Type/ Format	Step	Unit	Value Range (Default setting)
Actual Value (Input Registers) Address 0000-0FFF						
PRODUCT ID	0000	Hardware Version	ASCII		See note 1	
	0001	Software Version Code	ASCII		See note 1	
	0002	Serial Number char1 &2	ASCII		See note 1	
	0003	Serial Number char3 &4	ASCII		See note 1	
	0004	Serial Number char5 &6	ASCII		See note 1	
	0005	Serial Number char7 &8	ASCII		See note 1	
	0006	Serial Number char9&10	ASCII		See note 1	
	0007	Serial Number char 11&12	ASCII		See note 1	
	0008	…Reserved…	ASCII			
	0009	…Reserved…				

Group	Register Address (Hex)	Value Name	Data Type/ Format	Step	Unit	Value Range (Default setting)
	000A	Reserved…				
	000B	Reserved…				
	000C	Reserved…				
	000D	Reserved…				
	000E	Reserved…				
	000F	Reserved…				
PRODUCT ID	0010	Reserved…				
	0011	Reserved…				
	0012	Reserved…				
	0013	Reserved…				
	0014	Reserved…				
	0015	Reserved…				
	0016	Module Type	F0			
	0017	Alarm Flag1	F1			
	0018	Alarm Flag2	F2			
	0019	Alarm Flag3	F3			
	001A	Alarm Flag4	F4			
	001B	Set_Para_Status	F11			
	001C	Network port monitoring	F13			
STATUS	001D	Network port FDR status	F14			
	001E	Last alarm	F9	1		0-11
	001F	Last operation	F10	1		1-3
	0020	…Reserved…				
	0021	…Reserved…				
	0022	…Reserved…				
	0023	…Reserved…				
	0024	…Reserved…				

Group	Register Address (Hex)	Value Name	Data Type/ Format	Step	Unit	Value Range (Default setting)
	0025	...Reserved...				
	0026	...Reserved...				
STATUS	0027	...Reserved...				
	0028	...Reserved...				
	0029	...Reserved...				
	002A-003F	...Reserved...				
Monitoring for Communication	0040	Ethernet IP address byte1-2				
	0041	Ethernet IP address byte3-4				
	0042	Ethernet subnet mask byte1-2				
	0043	Ethernet subnet mask byte3-4				
	0044	Ethernet gateway address byte1-2				
	0045	Ethernet gateway address byte3-4				
	0046	Ethernet MAC Address byte1-2				
	0047	Ethernet MAC Address byte3-4				
	0048	Ethernet MAC Address byte5-6				
	0049	...Reserved...				
	004A-004F	...Reserved...				
USER DEFINABLE DATA ↓	0050	User Definable Data 0				
	0051	User Definable Data 1				
	0052	User Definable Data 2				

Group	Register Address (Hex)	Value Name	Data Type/ Format	Step	Unit	Value Range (Default setting)
	0053	User Definable Data 3				
	0054	User Definable Data 4				
USER DEFINABLE DATA	0055	User Definable Data 5				
	0056	User Definable Data 6				
	0057	User Definable Data 7				
	0058	User Definable Data 8				
	↓	↓	↓	↓	↓	↓
	009F	User Definable Data 4F				

Command registers

Group	Register Address (Hex)	Description	Data Type/ Format	Step	Unit	Value Range (Default setting)
Commands (For function code 10H only)	5000	Command Function Code	UWORD	-		5
	5001	Command Operation Code	F8	1		1-3

Notes:

1) How to read version

The version has two chars; the high byte is the main version and the low byte is the sub version.

For example: Hardware version value read as **"31 32"** represents actual version **"1.2".**

2) How to read serial number

The total serial number has 12 bytes and likes following illustration:

	Hardware Version	Manufacture date						Individual identity number part	
		Y	Y	Y	Y	M	M		
	000A	000B		000C		000D		000E	000F

Registers Address (word)	High	Low	High	Low	High	Low	High	Low	High	Low	High	Low
Position	1	2	3	4	5	6	7	8	9	10	11	12
Definition	Main Version	Sub version										

Example:

Position	1	2	3	4	5	6	7	8	9	10	11	12
Value	31	32	32	30	31	38	31	32	31	32	33	34

Hardware version: 1.2

Manufacture date: Year: 2018 month: 12

Identity number: 1234

7 Appendix B: Data Format

FORMAT CODE	DESCRIPTION	VALUE (HEX)
F0	**Module Type**	
	85=GW-Modbus TCP/RTU	0055
F1	**Alarm Status Flag1**	
	Watchdog	0001
	Network port	0002
	Network port 1	0004
	Network port 2	0008
	…Reserved…	0010
	…Reserved…	0020
	Invalid parameter	0040
	MAC Address wrong	0080
	Boot2 CRC error	0100
	Modbus RTU Communication	0200
	Self-check	0400
F2	**Alarm Status Flag2**	
	…Reserved…	
F3	**Alarm Status Flag3**	
	…Reserved…	
F4	**Alarm Status Flag4**	
	…Reserved…	
F5	**Parity check**	FFFF
	0=None	0000
	1=Odd	0001
	2=Even	0002

F6	**MODBUS Baud Rate**	FFFF
	1=1200	0001
	2=2400	0002
	3=4800	0003
	4=9600	0004
	5=19200	0005
	6=38400	0006
	7=57600	0007
	8=115200	0008
F7	**Enable/Disable**	
	0=Disable	0000
	1=Enable	0001
	2=…Reserved…	0002
	3=Alarm only	0003
F8	**Command**	
	1=Reserved	
	2=FDR data backup command	0002
	3=FDR data restore command	0003
F9	**The last alarm**	
	0=No alarm	0000
	1=Watchdog	0001
	2=Network port	0002
	3=Network port1 error	0003
	4=Network port2 error	0004
	…Reserved…	0005
	…Reserved…	0006
	7=Invalid parameter	0007

F9	8=MAC Address wrong	0008
	9= Boot2 CRC error	0009
	10= Modbus RTU Communication	000A
	11=Self check	000B
F10	**The last operation**	
	1=Reserved	
	2=Update parameter	0002
	3=Alarm reset	0003
F11	**Set Parameters Status**	
	1=Setting Parameters	0001
	2=Parameter Setting Done	0002
	3=Parameter Setting Failure	0003
	4=Parameters Not Set	0004
F13	**Network port monitoring**	
	Network port communicating	0001
	Network port connected	0002
	Network port 1 error	0004
	Network port 2 error	0008
	Network port bad config	0010

F14	**Network port FDR status**	
	0=Ready, IP available, no error	0x00
	1=No response from IP server	0x01
	2=No response from parameter server	0x02
	3=No file on parameter server	0x03
	4=Corrupt file on parameter server	0x04
	5=Empty file on parameter server	0x05
	6=Internal communication fault	0x06
	7=Write error copying settings to parameter server	0x07
	8=Invalid settings provided by the controller	0x08
	9=CRC mismatch between parameter server and controller	0x09
F15	**RTU timeout type**	
	0=Default	0x00
	1= Expert	0x01

8 Appendix C: Technical data

Power supply MS572 Interface	
Rated operational voltage (Ue)	24VDC
Voltage operation range	85-110% Ue
Power consumption	
Typical	1.5W
Maximum	2W
Maximum starting current	100mA
Modbus/TCP Communication interface	
Protocol	Modbus TCP
Communication speed	10/100Mbps
Interface	RJ45
EMC	
Electrostatic discharge	IEC61000-4-2 Contact discharge Level 2 Air discharge Level 3
Electromagnetic field immunity	IEC61000-4-3 80-1000MHz Level 3 1.4-6GHz Level 2
Electrical fast transient/burst immunity	IEC61000-4-4 Level 2
Surge immunity	IEC61000-4-5 Level 2
Conducted disturbance immunity	IEC61000-4-6 Level 3
Radiated disturbance	EN55011/CISPR11 Class A
General	
Installation	TS35 DIN rail
Degree of protection	IP20
Storage	-40~+85 ℃
Operation	-10~+60 ℃
Humidity	15% up to 95% without dew
Material	PA6 GF20
	Halogen-free
	Flammability rating UL94V-2
	Color RAL7012
Pollution degree	2

Revision History

Revision	Page(s)	Description of change	Date
M0201		Initial Edition	2019-12
M0202		Format changed, Add in Modbus RTU timeout	2021-08
A		Revise the IP setting range, Add in Ethernet port1/2 alarm	2022-01
B		Add Modbus TCP Keepalive time Remove SNMP & DHCP	2022-12

ABB
Publication Editor:
Electrification-Low Voltage System
ABB (China) Limited
No.885, FangShanXiEr Road, Xiang'an District, Xiamen,
Fujian, 361101 P.R.China

Local Contacts on
http://www.abb.com/mns

© Copyright 2022 ABB. All rights reserved.
Specifications subject to change without notice.

Document Number: 1TNC928243M0203 Rev. B