FDT 2.1 for Modbus
Protocol Annex

Technical Specification

Version 1.01.00

FDT Group – Document No. <19-0005>

Publisher:

FDT Group AISBL
Industrieweg 5 • 3001 Heverlee • Belgium
242 East Washington Street • Marshfield MO 65706 • USA

www.fdtgroup.org
info@fdtgroup.org

History

Rev.	Author	Change Description	Date
1.01.00	Wagner	Initial release of FDT2.1 Annex: Datatype references updated to FDT 2.1 Core specification Editorial corrections, FDT Group address and icon updated Table 59: ConformityLevel code clarification (hex notation) for conformance with [2] Table 69: Protocol specific names corrected (spaces inserted) in document to ensure consistency with data type source code	2020-08-06
1.00	PG Modbus	FDT2 Annex release version 1.00	2014-03-01

CONTENTS

Table of Figures

Table of Tables

1 Scope

1.1 General

This document provides information for integrating the Modbus protocol into the Field Device Tool (FDT) specification version 2.1.

This document neither contains the FDT2.1 specification nor modifies it. The FDT2.1 specification is available from the homepage of the FDT Group (www.fdtgroup.org).

This document also neither contains the complete specification for MODBUS networks nor modifies them.

1.2 Intended audience

The intended audiences of this document are persons who are going to develop FDT2.1 products for MODBUS.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61158:2008 (all parts): *Industrial communication networks – Fieldbus specifications*

IEC 61784-1: *Industrial communication networks - Profiles - Part 1: Fieldbus profiles*

IEC 62453-1:2009: *Field Device Tool (FDT) interface specification – Part 1: Overview and guidance*

IEC 62453-2:2009: *Field Device Tool (FDT) interface specification – Part 2: Concepts and detailed description*

IEC 62453-41:2009: *Field Device Tool (FDT) interface specification – Part 41: Object model integration profile – Common object model*

FDT2.1, Technical Specification, Version 1.01.00, FDT Group, AISBL; March 2018

3 Terms, definitions, symbols, abbreviated terms and conventions

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in FDT2.1, IEC 62453-1, IEC 62453-2, IEC 62453-41, MSDN® apply.

3.2 Abbreviations

For the purpose of this document, the abbreviations given in IEC 62453-1, IEC 62453-2 and the following apply.

DTM	Device Type Manager
IP	Internet Protocol
TCP	Transmission Control Protocol
UML	Unified Modeling Language
IO	Input / Output

3.3 Conventions

3.3.1 General convention

The conventions for naming and referencing of data types are explained in FDT2.1 Specification.

3.3.2 Vocabulary for requirements

The following expressions are used when specifying requirements.

Usage of “shall” or “Mandatory”	No exceptions allowed.
Usage of “should” or “recommended”	Strong recommendation. It may make sense in special exceptional cases to differ from the described behavior.
Usage of “can" or “optional"	Function or behavior may be provided, depending on defined conditions.

3.3.3 Use of UML

Figures in this document are using UML notation as defined in Annex I of the FDT2.1 specification.

4 Bus category

The MODBUS protocol over Serial Line and over TCP is identified by the following unique identifiers in the ProtocolId property of class BusCategory:

Table 1 – Definition of BusCategory

ProtocolId value	ProtocolName value	Description
59629a40-285f-11db-a98b-0800200c9a66	Modbus Serial	Support of Modbus over Serial Line
59629a41-285f-11db-a98b-0800200c9a66	Modbus TCP	Support of Modbus over TCP

Table 2 defines which PhysicalLayer can be used together with the BusCategory defined in Table 1 for the Modbus Serial protocol.

Table 2 – Physical Layer Identifiers Modbus Serial

Identifier value	Description
C0458028-F240-45A5-8664-70DC84FDC6FE	RS-232
3BF008DC-5A44-4220-8C3E-3C46A589A0B4	RS-422
036D1591-387B-11D4-86E1-00E0987270B9	RS-485

Table 3 defines which PhysicalLayer can be used together with the BusCategory defined in Table 1 for the Modbus TCP protocol.

Table 3 – Physical Layer Identifiers Modbus TCP

Identifier value	Description
307DD810-C010-11DB-90E7-0002B3ECDCBE	100BaseTXFD (default for Media Type Copper)
307DD812-C010-11DB-90E7-0002B3ECDCBE	100BaseFXFD (default for Media Type Fiber Optic)
307DD813-C010-11DB-90E7-0002B3ECDCBE	100BaseLX10
307DD816-C010-11DB-90E7-0002B3ECDCBE	1000BaseXFD
307DD818-C010-11DB-90E7-0002B3ECDCBE	1000BaseLXFD
307DD81A-C010-11DB-90E7-0002B3ECDCBE	1000BaseSXFD
307DD81C-C010-11DB-90E7-0002B3ECDCBE	1000BaseTFD
307DD81D-C010-11DB-90E7-0002B3ECDCBE	10GbaseFX
307DD81E-C010-11DB-90E7-0002B3ECDCBE	10GbaseLX4
307DD81F-C010-11DB-90E7-0002B3ECDCBE	10GBaseR
307DD820-C010-11DB-90E7-0002B3ECDCBE	10GbaseER
307DD821-C010-11DB-90E7-0002B3ECDCBE	10GbaseLR
307DD822-C010-11DB-90E7-0002B3ECDCBE	10GbaseSR
307DD823-C010-11DB-90E7-0002B3ECDCBE	10GbaseW
307DD824-C010-11DB-90E7-0002B3ECDCBE	10GbaseEW
307DD824-C010-11DB-90E7-0002B3ECDCBE	10GbaseLW
307DD825-C010-11DB-90E7-0002B3ECDCBE	10GbaseSW
307DD826-C010-11DB-90E7-0002B3ECDCBE	10GbaseCX4
307DD827-C010-11DB-90E7-0002B3ECDCBE	2BaseTL
307DD828-C010-11DB-90E7-0002B3ECDCBE	10PassTS
307DD829-C010-11DB-90E7-0002B3ECDCBE	100BaseBX10D
307DD82A-C010-11DB-90E7-0002B3ECDCBE	100BaseBX10U
307DD82B-C010-11DB-90E7-0002B3ECDCBE	100BaseLX10
307DD82C-C010-11DB-90E7-0002B3ECDCBE	1000BaseBX10D
307DD82D-C010-11DB-90E7-0002B3ECDCBE	1000BaseBX10U
307DD82F-C010-11DB-90E7-0002B3ECDCBE	1000BaseLX10

Identifier value	Description
307DD830-C010-11DB-90E7-0002B3ECDCBE	1000BasePX10D
307DD831-C010-11DB-90E7-0002B3ECDCBE	1000BasePX10U
307DD832-C010-11DB-90E7-0002B3ECDCBE	1000BasePX20D
307DD833-C010-11DB-90E7-0002B3ECDCBE	1000BasePX20U
307DD834-C010-11DB-90E7-0002B3ECDCBE	10GBaseT or 100BasePXFD
307DD835-C010-11DB-90E7-0002B3ECDCBE	10GBaseLRM
307DD836-C010-11DB-90E7-0002B3ECDCBE	1000BaseKX
307DD837-C010-11DB-90E7-0002B3ECDCBE	1000BaseKX4
307DD838-C010-11DB-90E7-0002B3ECDCBE	1000BaseKR
307DD839-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePRXD1
307DD83A-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePXRD2
307DD83B-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePXRD3
307DD83C-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePRXU1
307DD83D-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePXRU2
307DD83E-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePXRU3
307DD83F-C010-11DB-90E7-0002B3ECDCBE	10GBasePRD1
307DD840-C010-11DB-90E7-0002B3ECDCBE	10GBasePRD2
307DD841-C010-11DB-90E7-0002B3ECDCBE	10GBasePRD3
307DD842-C010-11DB-90E7-0002B3ECDCBE	10GBasePRU1
307DD843-C010-11DB-90E7-0002B3ECDCBE	10GBasePRU3
307DD844-C010-11DB-90E7-0002B3ECDCBE	40GbaseKR4
307DD845-C010-11DB-90E7-0002B3ECDCBE	40GbaseCR4
307DD845-C010-11DB-90E7-0002B3ECDCBE	40GbaseSR4
307DD846-C010-11DB-90E7-0002B3ECDCBE	40GbaseFR
307DD847-C010-11DB-90E7-0002B3ECDCBE	40GbaseLR4
307DD848-C010-11DB-90E7-0002B3ECDCBE	100GbaseCR10
307DD849-C010-11DB-90E7-0002B3ECDCBE	100GbaseSR10
307DD84A-C010-11DB-90E7-0002B3ECDCBE	100GbaseLR4
307DD84B-C010-11DB-90E7-0002B3ECDCBE	100GbaseER4
307DD84C-C010-11DB-90E7-0002B3ECDCBE	100BasePXFD
307DD84D-C010-11DB-90E7-0002B3ECDCBE	Radio Communication
307DD84E-C010-11DB-90E7-0002B3ECDCBE	Speed of 100 Mbit/s (and more) and full duplexity

The DataLinkLayer property is not applicable for Modbus and has to be set to null.

5 Access to instance, device and process data

5.1 General

The minimum set of provided data shall be:

- All process values available for the device shall be modeled as ProcessData including the ranges and scaling if applicable
- All network configuration related parameters shall be exposed in NetworkData (see section 8)

5.2 IO signals provided by DTM

A DTM shall provide IO signal information for the device using the IProcessData interface. The IO signals describe data type and address parameters of process data as detailed in chapter 10.2.

5.3 Data interfaces

For Modbus no minimum set of parameters or common data set is defined which must be provided by a DTM. Exposure of device parameters is at the vendor's discretion. If a DTM provides process values the process variables shall be modeled as process data objects.

5.3.1 Mapping of Modbus data types to FDT data types

Modbus uses data types as specified in [2] for the transmission on the fieldbus. The FDT interfaces IDeviceData and IInstanceData use .NET data types, while PLC applications use data types defined in IEC 61131-3. This chapter defines the mapping of parameter data types, whereas mapping of process data types is defined in Chapter 11.3.

The mapping of parameter data types is described in Table 4.

Table 4 – Mapping of data types

Modbus data type	FDT data type	IEC 61131 data type
Discrete Inputs	BitArray	ARRAY [] OF BOOL
Coil	bool	BOOL
Coils	BitArray	ARRAY [] OF BOOL
Input Registers	ushort[]	ARRAY [] OF WORD
Holding Register	ushort	WORD
Holding Registers	ushort[]	ARRAY [] OF WORD

The FDT data types do not change any byte or bit order of the data. The data interpretation, e.g. if a float value is transferred in a holding register, is under the responsibility of the DTM respectively the user of the data. The user of the data has to know how the data is represented by the hardware device.

5.3.2 SemanticInfo

The identifier in SemanticId shall be unique and always reference the same element. This means the semantic information shall be the same whenever the same data is referenced. By using this attribute, e.g. a Frame Application is able to get the information regarding the meaning and usage of a single data structure.

Table 5 – Usage of general data types

Attribute	Description for use in Modbus
SemanticInfo.ParameterReadAddress SemanticInfo.ParameterWriteAddress	The value of ParameterReadAddress and ParameterWriteAdress is a string conforming to a pattern as follows: FunctionCode**.**StartAddress**.**BitOffset**.**BitLength where FunctionCode = Modbus function code to access the data StartAddress = starting address BitOffset = start bit within the starting address BitLength = length in bits **All values are unsigned integers!**
SemanticInfo.ApplicationDomain	The value of ApplicationDomain is as follows: MODBUS
SemanticInfo.SemanticId	The value of SemanticId is vendor specific.

6 Protocol specific behavior

6.1 Modbus data and addressing model

The Modbus data model is very simple and described in [2]. Figure 1 shows an extract of the data model description.

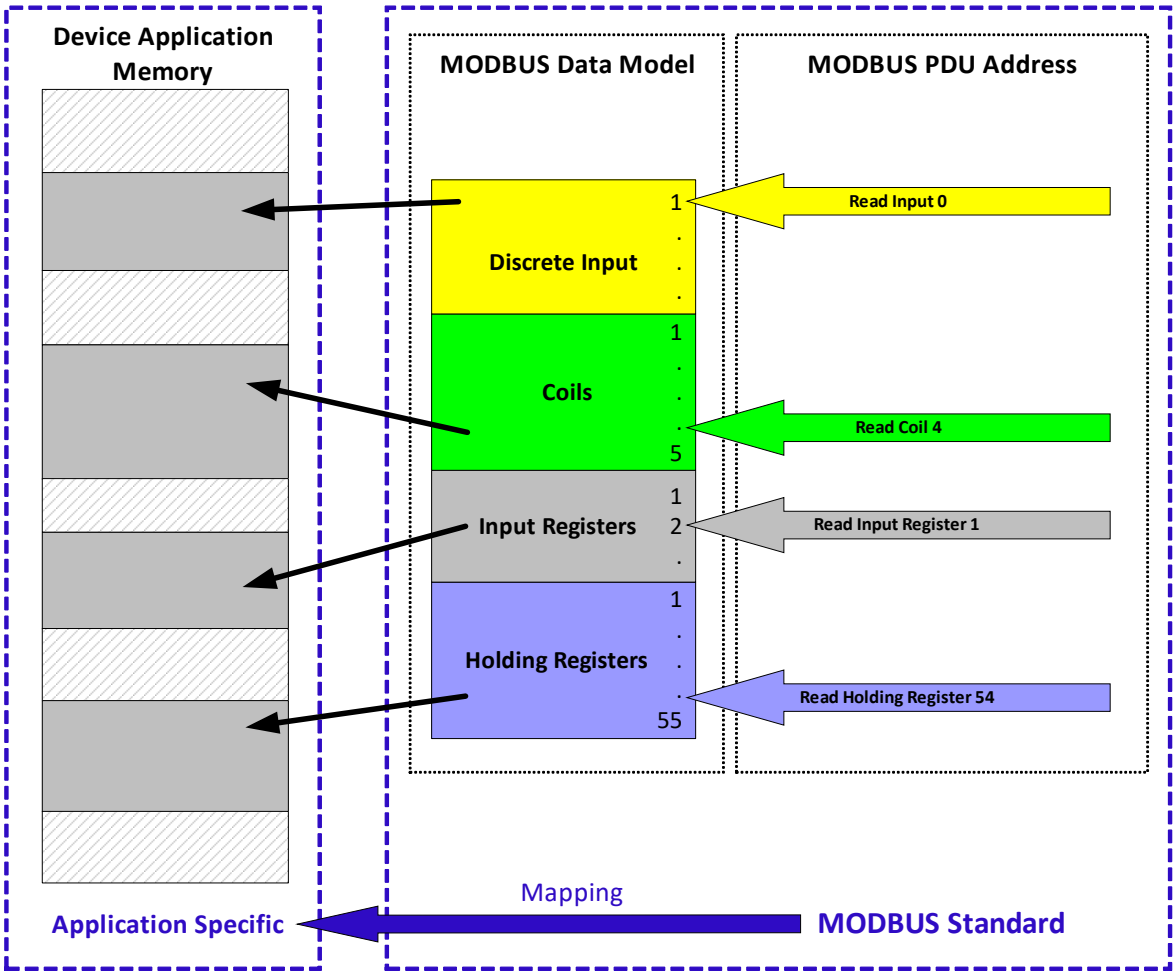


Figure 1 – Modbus Data and Addressing Model

6.2 Modbus-related information of a device DTM

The information used by a Modbus client (IO scanner) to set up the Modbus network properly and allow cyclic communication between control system and Modbus server devices is provided by a DTM in Process data items.

A DTM of a Modbus device must deliver process data items related information to get integrated into an FDT-based engineering system.

This specification makes no assumptions whether a modular Modbus device is modeled in a Device DTM, Composite DTM or Gateway DTM. All types of DTMs have to provide all mandatory information defined in the following chapters.

7 Protocol specific usage of general FDT2.1 data types

The FDT2.1 specification already defines a set of data types that can be used to identify a device and to provide device information. This chapter describes how these data types are used with Modbus.

Table 6 – Usage of general data types

Attribute	Description for use in Modbus
ProtocolId	See chapter 4
PhysicalLayer	Not applicable
ApplicationDomain / SemanticId	See chapter 5.3.2
Address	See chapter 8.1
ManufacturerId	Modbus Vendor Name
DeviceTypeId	Modbus Product Code
HardwareRevision	Not applicable
SoftwareRevision	Modbus Major Minor Revision
ProtocolIdentificationProfile	Not applicable

7.1 Protocol specific behavior

7.1.1 Broadcasting

In broadcast mode a DTM can send a Modbus request to all devices connected to the bus. This mode is only supported for devices which are connected via Modbus Serial Line. The connection can either be a direct connection or a connection via a gateway. The broadcast mode must be initiated by a ConnectRequest with the slave address of the target device set to 0 (SlaveAddress=0).

Because in broadcast mode no response will be returned by the device, the broadcast mode shall be only used with the transaction requests listed in the table below.

Table 7 – Usage of broadcasts in transaction requests

Broadcast Transaction Request	Restrictions
ModbusWriteSingleRequest	None
ModbusWriteSingleRegisterRequest	None
ModbusDiagnosticsRequest	This transaction request shall be used in broadcast mode only with the following sub-functions: 0x01: Restart Communication Option 0x03: Change ASCII Input Delimiter 0x04: Force Listen Only Mode 0x0A: Clear Counters and Diagnostic Register 0x14: Clear Overrun Counter and Flag
ModbusWriteMultipleCoilsRequest	None
ModbusWriteMultipleRegisterRequest	None

Broadcast Transaction Request	Restrictions
ModbusWriteFileRecordRequest	None
ModbusMaskWriteRegisterRequest	None
ModbusPrivateRequest	Shall be only used with private services if no response is required from the device

Although in broadcast mode no response will be returned by the target devices, a transaction response which corresponds to the transaction request shall be generated. This transaction response shall be generated in order to inform the DTM that the broadcast request was sent on the bus. If a Modbus Communication DTM is used to establish the communication, it has to provide this generated transaction response. If no Modbus Communication DTM is used to establish the communication, the Modbus Gateway DTM, which provides the link to the higher communication level different from Modbus, has to provide this transaction response.

7.1.2 Unconfirmed private Modbus request

This transaction request can be used to send any unconfirmed private Modbus request. Although no response will be returned by the target device, a transaction response, which corresponds to the transaction request, shall be generated. This transaction response shall be generated in order to inform the DTM that the unconfirmed request was sent on the bus. If a Modbus Communication DTM is used to establish the communication, it has to provide this generated transaction response. If no Modbus Communication DTM is used to establish the communication, the Modbus Gateway DTM, which provides the link to the higher communication level different from Modbus, has to provide this transaction response.

8 Protocol specific common data types

8.1 ModbusDeviceAddress data type

The protocol specific device address relates to network management and communication.

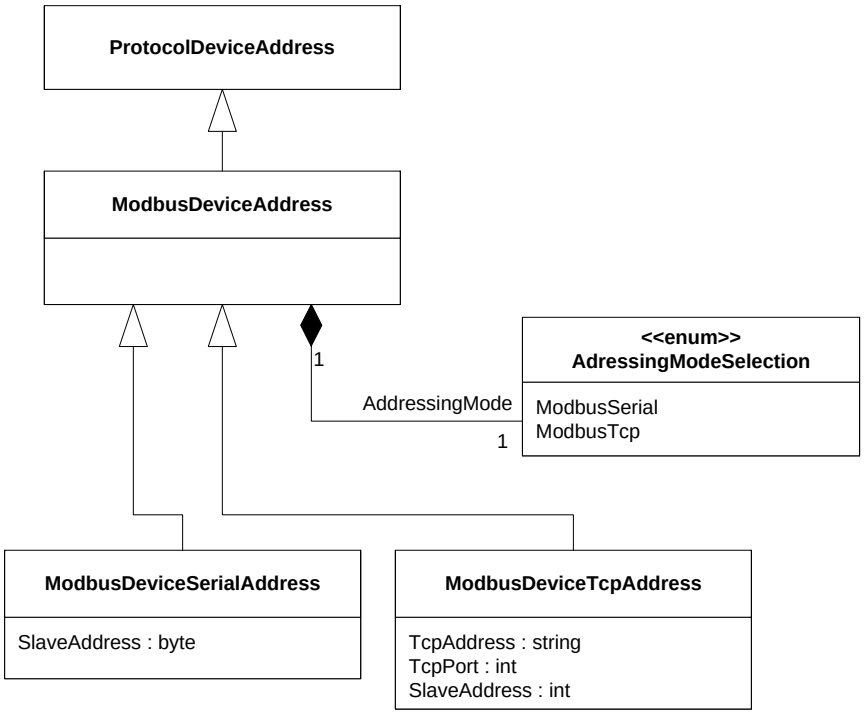


Figure 2 – ModbusDeviceAddress

Table 8 – ModbusDeviceAddress

Member Name	Type	Description
AddressingModeSelection	enum	Specifies whether a Modbus Serial or a Modbus TCP address is used

Table 9 – ModbusDeviceSerialAddress

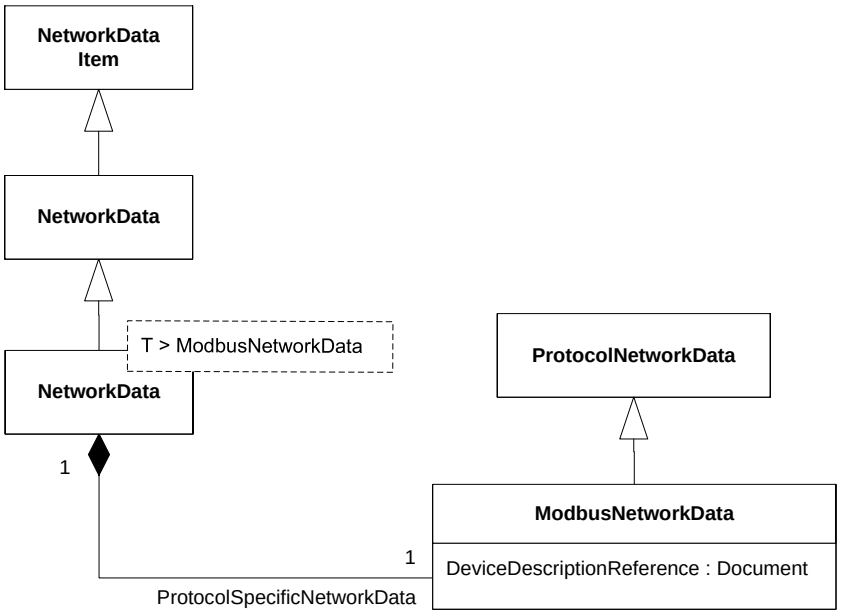
Member Name	Type	Description
SlaveAddress	byte	Address of the Modbus Serial device

Table 10 – ModbusDeviceTcpAddress

Member Name	Type	Description
TcpAddress	string	String representation of the IP address of a Modbus TCP device
TcpPort	int	Port of the Modbus TCP connection
SlaveAddress	int	Slave address of a (virtual) Modbus device behind a Modbus TCP/ Modbus Serial Line gateway

9 Network management

The data needed for management of the network are exposed by the Device DTM in the INetworkData interface (see Figure 3).



Used in INetworkData.GetNetworkDataInfo()

Figure 3 – ModbusNetworkData

The properties of ModbusNetworkData are described in the following table.

Table 11 – Modbus Network Data

Member Name	Type	Description
DeviceDescriptionReference	Document	Path to file which contains the Modbus device description if available

9.1 General

The data types specified in this sub clause are used in the following services:

- GetNetworkDataInfo service;
- SetNetworkData service.

The data type ModbusDeviceSerialAddress respectively the data type ModbusDeviceTcpAddress is used for defining the network address of a device.

9.2 Configuration

Modbus networks do not require a special configuration. The focus for Modbus DTMs is on the process data items and parameterization of devices. See chapters 9.4 and 10.2.

9.3 Process Data Items

For a detailed description about Process Data Items, please refer to chapter 10.2.

9.4 Parameterization

For the parameterization of a Modbus device the DTM must use the communication services and communication data types described in chapter 10.

Because Modbus does not define a specific or common way to parameterize a device, the parameterization is vendor specific.

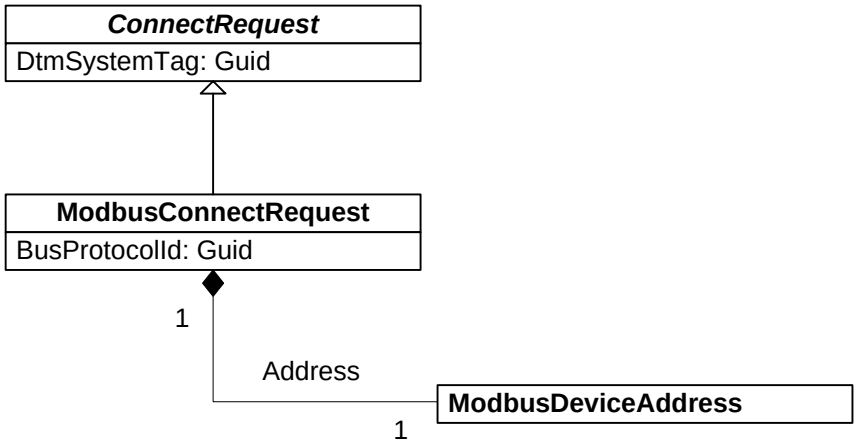
10 Communication data types

10.1 Introduction

The data types contain the address information and the communication data required to execute the respective request or to transport the response information.

10.2 ModbusConnectRequest

This is the Modbus specific implementation of the abstract class ConnectRequest (see Figure 4).



Used in ICommunication.BeginConnect()

Figure 4 – ModbusConnectRequest

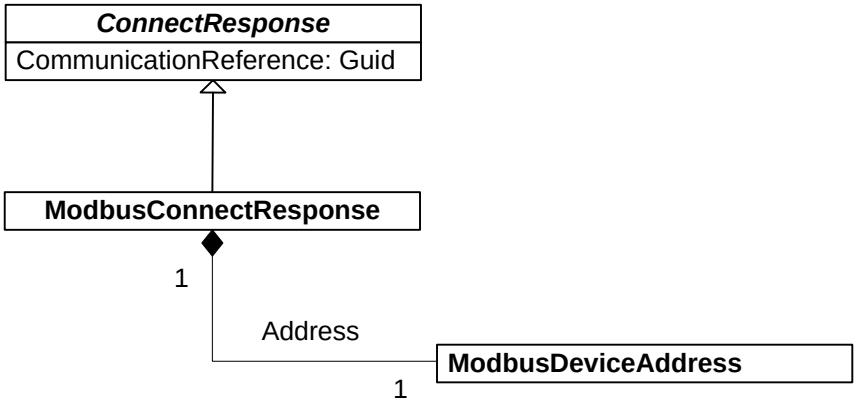
The properties of the ModbusConnectRequest data type are described in Table 12.

Table 12 – ModbusConnectRequest data type

Property	Description
Address	Address information of the Modbus device
BusProtocolId	Protocol identifier of the used protocol (Serial or TCP) of the connect request
DtmSystemTag	Unique identification of the DTM in the Frame Application.

10.3 ModbusConnectResponse

This is the Modbus specific implementation of the abstract class ConnectResponse (see Figure 5).



Used in ICommunication.EndConnect()

Figure 5 – ModbusConnectResponse

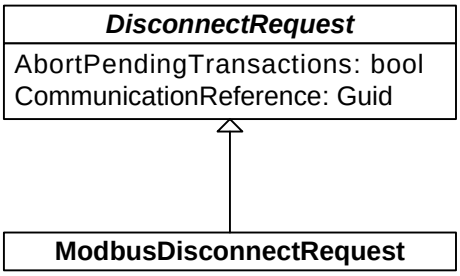
The properties of the ModbusConnectResponse data type are described in Table 13.

Table 13 – ModbusConnectResponse data type

Property	Description
Address	Address information of the Modbus device
CommunicationReference	Identifier for a communication link to a device.

10.4 ModbusDisconnectRequest

This is the Modbus specific implementation of the abstract class DisconnectRequest (see Figure 6).



Used in ICommunication.BeginDisconnect()

Figure 6 – ModbusDisconnectRequest

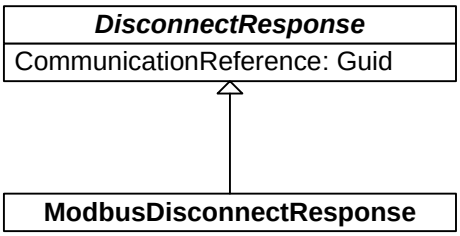
The properties of the ModbusDisconnectRequest data type are described in Table 14.

Table 14 – ModbusDisconnectRequest data type

Property	Description
AbortPendingTransactions	Indicates whether pending transactions shall be aborted.
CommunicationReference	Identifier for a communication link to a device.

10.5 ModbusDisconnectResponse

This is the Modbus specific implementation of the abstract class DisconnectResponse (see Figure 7).



Used in ICommunication.EndDisconnect()

Figure 7 – ModbusDisconnectResponse

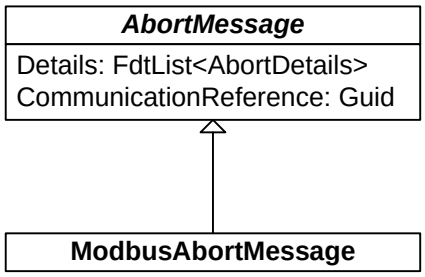
The properties of the ModbusDisconnectRequest data type are described in Table 15.

Table 15 – ModbusDisconnectResponse data type

Property	Description
CommunicationReference	Identifier for a communication link to a device.

10.6 ModbusAbortMessage

This is the Modbus specific implementation of the abstract AbortMessage class (see Figure 8).



Used in ICommunication.EndDisconnect()

Figure 8 – ModbusAbortMessage

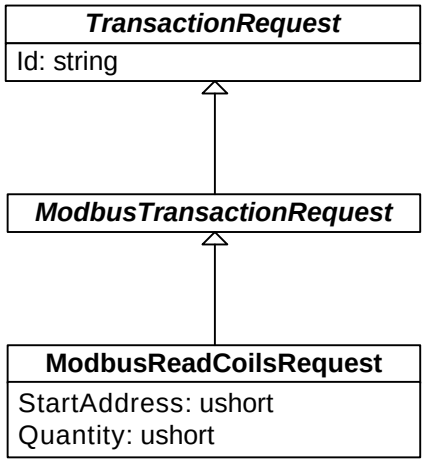
The properties of the ModbusAbortMessage data type are described in Table 16.

Table 16 – ModbusAbortMessage data type

Property	Description
CommunicationReference	Identifier for a communication link to a device.
Details	Details about the cause and source of the Abort.

10.7 ModbusReadCoilsRequest

This chapter describes the request for the Modbus service Read Coils (see Figure 9).



Used in ICommunication.BeginCommunicationRequest()

Figure 9 – ModbusReadCoilsRequest

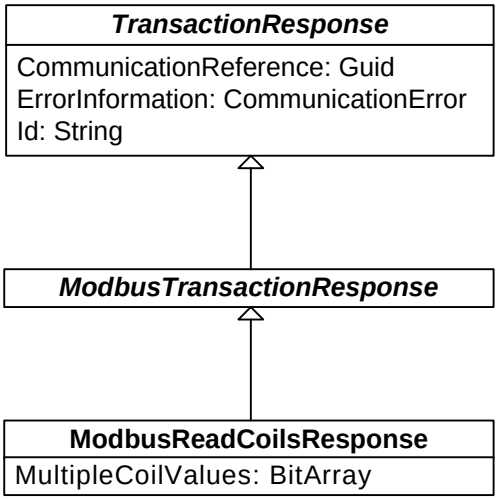
The properties of the ModbusReadCoilsRequest data type are described in Table 17.

Table 17 – ModbusReadCoilsRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	n/a
Quantity	Number of coils to be read	Quantity of coils
StartAddress	Address of the first coil to be read	Address of first coil

10.8 ModbusReadCoilsResponse

This chapter describes the response for the Modbus service Read Coils (see Figure 10).



Used in ICommunication.EndCommunicationRequest()

Figure 10 – ModbusReadCoilsResponse

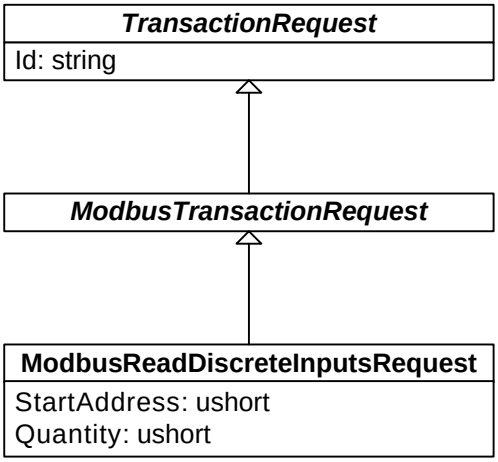
The properties of the ModbusReadCoilsResponse data type are described in Table 18.

Table 18 – ModbusReadCoilsResponse data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
Id	[Optional] Identifier for a single Transaction Request.	n/a
MultipleCoilValues	Bit array with each coil state coded in one character: “0” = FALSE or “OFF” “1” = TRUE or “ON”	Data

10.9 ModbusReadDiscreteInputsRequest

This chapter describes the request for the Modbus service Read Discrete Inputs (see Figure 11).



Used in ICommunication.BeginCommunicationRequest()

Figure 11 – ModbusReadDiscreteInputsRequest

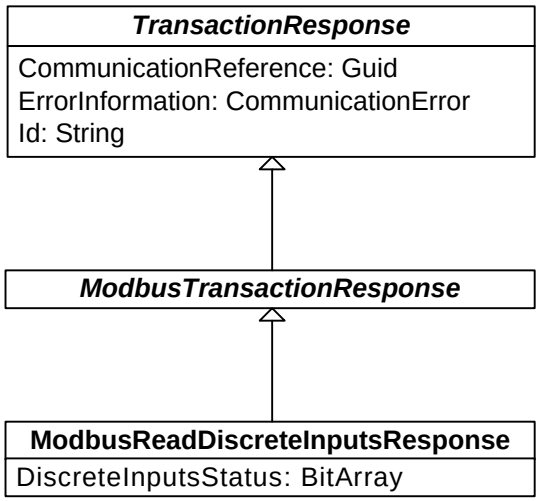
The properties of the ModbusReadDiscreteInputsRequest data type are described in Table 19.

Table 19 – ModbusReadDiscreteInputsRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	n/a
Quantity	Number of discrete inputs to be read	Quantity of discretes
StartAddress	Address of the first discrete input to be read	Address of first discrete

10.10 ModbusReadDiscreteInputsResponse

This chapter describes the response for the Modbus service Read Discrete Inputs (see Figure 12).



Used in ICommunication.EndCommunicationRequest()

Figure 12 – ModbusReadDiscreteInputsResponse

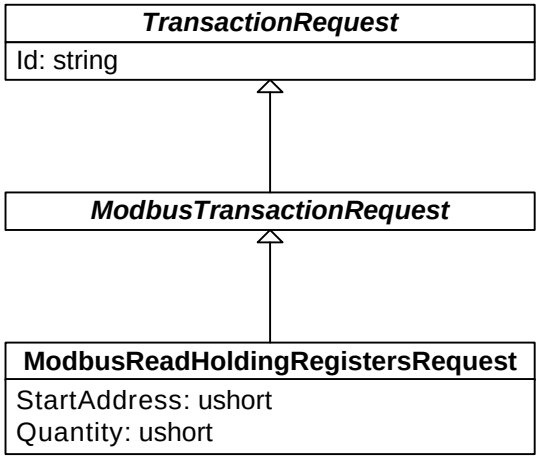
The properties of the ModbusDiscreteInputsResponse data type are described in Table 20.

Table 20 – ModbusReadDiscreteInputsResponse data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
DiscreteInputsStatus	Bit-array with each coil state coded in one character: "0" = FALSE or "OFF" "1" = TRUE or "ON"	Data
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
Id	[Optional] Identifier for a single Transaction Request.	n/a

10.11 ModbusReadHoldingRegistersRequest

This chapter describes the request for the Modbus service Read Holding Registers (see Figure 13).



Used in ICommunication.BeginCommunicationRequest()

Figure 13 – ModbusReadHoldingRegistersRequest

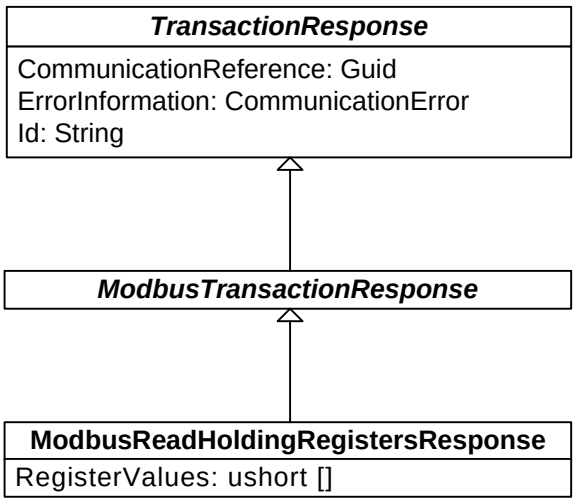
The properties of the ModbusReadHoldingRegistersRequest data type are described in Table 21.

Table 21 – ModbusReadHoldingRegistersRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	n/a
Quantity	Number of holding registers to be read	Quantity of holding registers to read
StartAddress	Address of the first holding register to be read	Address of first holding register to read

10.12 ModbusReadHoldingRegistersResponse

This chapter describes the response for the Modbus service Read Holding Registers (see Figure 14).



Used in ICommunication.EndCommunicationRequest()

Figure 14 – ModbusReadHoldingRegistersResponse

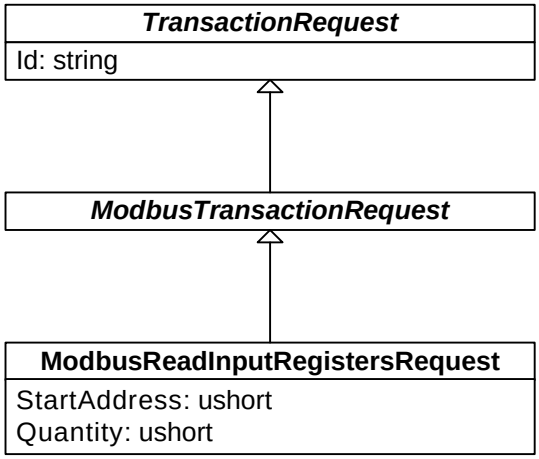
The properties of the ModbusReadHoldingRegistersResponse data type are described in Table 22.

Table 22 – ModbusReadHoldingRegistersResponse data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
Id	[Optional] Identifier for a single Transaction Request.	n/a
RegisterValues	Read holding register values	Data

10.13 ModbusReadInputRegistersRequest

This chapter describes the request for the Modbus service Read Input Registers (see Figure 15).



Used in ICommunication.BeginCommunicationRequest()

Figure 15 – ModbusReadInputRegistersRequest

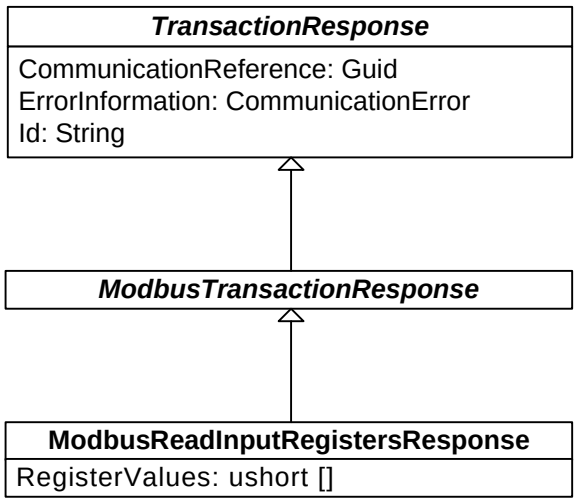
The properties of the ModbusReadInputRegistersRequest data type are described in Table 23.

Table 23 – ModbusReadInputRegistersRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	n/a
Quantity	Number of input registers to be read	Quantity of input registers
StartAddress	Address of the first input register to be read	Address of first input register

10.14 ModbusReadInputRegistersResponse

This chapter describes the response for the Modbus service Read Input Registers (see Figure 16).



Used in ICommunication.EndCommunicationRequest()

Figure 16 – ModbusReadInputRegistersResponse

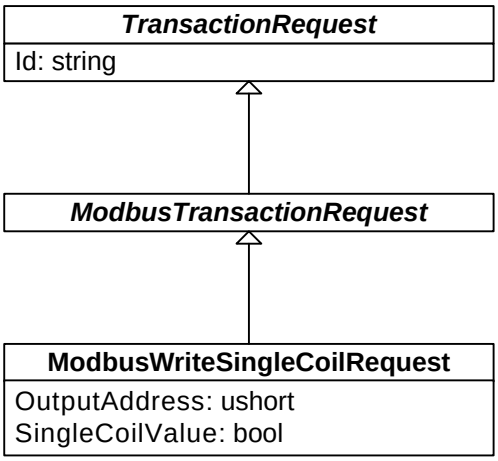
The properties of the ModbusReadInputRegistersResponse data type are described in Table 24.

Table 24 – ModbusReadInputRegistersResponse data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
Id	[Optional] Identifier for a single Transaction Request.	n/a
RegisterValues	Read input register values	Data

10.15 ModbusWriteSingleCoilRequest

This chapter describes the request for the Modbus service Write Single Coil (see Figure 17).



Used in ICommunication.BeginCommunicationRequest()

Figure 17 – ModbusWriteSingleCoilRequest

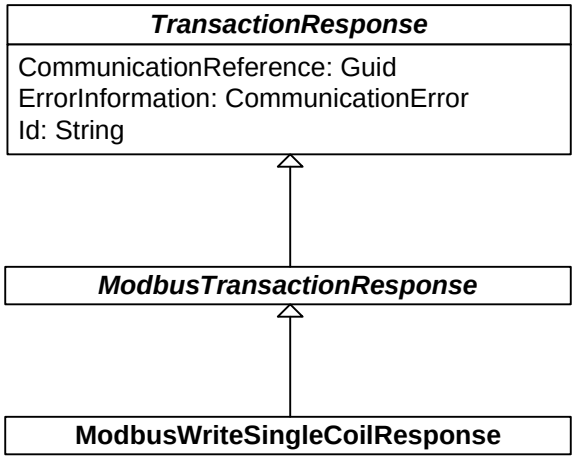
The properties of the ModbusWriteSingleCoilRequest data type are described in Table 25.

Table 25 – ModbusWriteSingleCoilRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	n/a
OutputAddress	Address of the coil to be forced	Address of first coil
SingleCoilValue	Coil state to be forced, with: "0" = FALSE or "OFF" "1" = TRUE or "ON"	Data single coil

10.16 ModbusWriteSingleCoilResponse

This chapter describes the response for the Modbus service Write Single Coil (see Figure 18).



Used in ICommunication.EndCommunicationRequest()

Figure 18 – ModbusWriteSingleCoilResponse

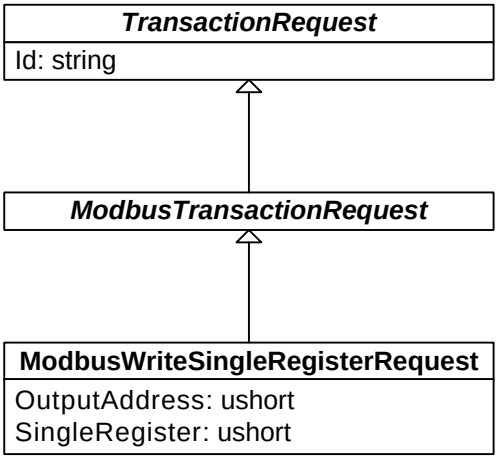
The properties of the ModbusWriteSingleCoilResponse data type are described in Table 26.

Table 26 – ModbusWriteSingleCoilResponse data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
Id	[Optional] Identifier for a single Transaction Request.	n/a

10.17 ModbusWriteSingleRegisterRequest

This chapter describes the request for the Modbus service Write Single Register (see Figure 19).



Used in ICommunication.BeginCommunicationRequest()

Figure 19 – ModbusWriteSingleRegisterRequest

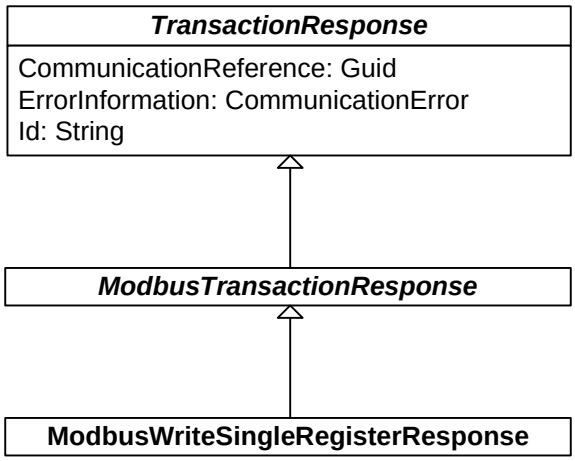
The properties of the ModbusWriteSingleRegisterRequest data type are described in Table 27.

Table 27 – ModbusWriteSingleRegisterRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	n/a
OutputAddress	Address of the holding register to be written	Address of first holding register to write
SingleRegister	Value to be written to the register	Data

10.18 ModbusWriteSingleRegisterResponse

This chapter describes the response for the Modbus service Write Single Register (see Figure 20).



Used in ICommunication.EndCommunicationRequest()

Figure 20 – ModbusWriteSingleRegisterResponse

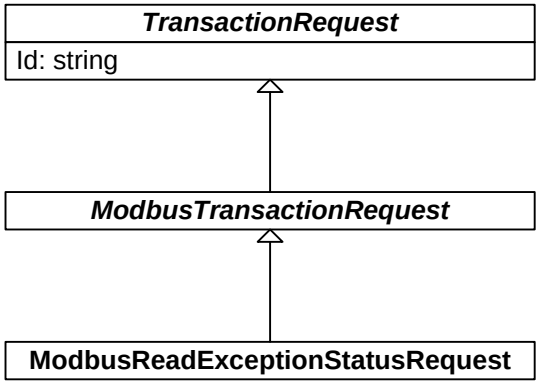
The properties of the ModbusWriteSingleRegisterResponse data type are described in Table 28.

Table 28 – ModbusWriteSingleRegisterResponse data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
Id	[Optional] Identifier for a single Transaction Request.	n/a

10.19 ModbusReadExceptionStatusRequest

This chapter describes the request for the Modbus service Read Exception Status (see Figure 21).



Used in ICommunication.BeginCommunicationRequest()

Figure 21 – ModbusReadExceptionStatusRequest

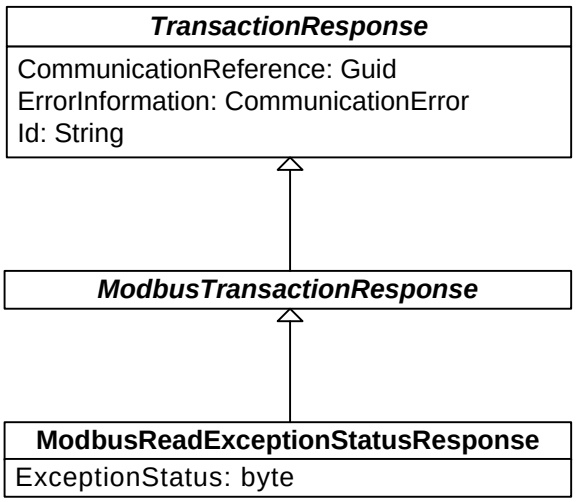
The properties of the ModbusReadExceptionStatusRequest data type are described in Table 29.

Table 29 – ModbusReadExceptionStatusRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	n/a

10.20 ModbusReadExceptionStatusResponse

This chapter describes the response for the Modbus service Read Exception Status (see Figure 22).



Used in ICommunication.EndCommunicationRequest()

Figure 22 – ModbusReadExceptionStatusResponse

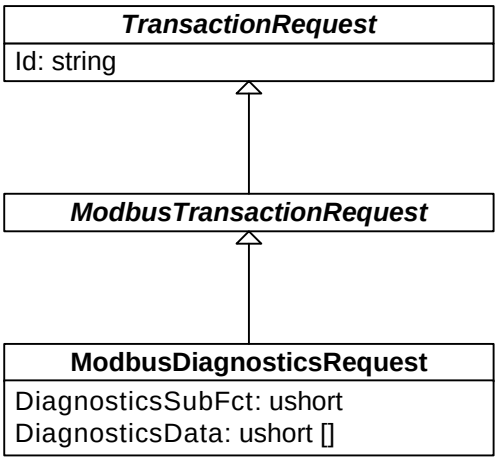
The properties of the ModbusReadExceptionStatusResponse data type are described in Table 30.

Table 30 – ModbusReadExceptionStatusResponse data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
ExceptionStatus	Exception status of Modbus Serial Line device	Output Data
Id	[Optional] Identifier for a single Transaction Request.	n/a

10.21 ModbusDiagnosticsRequest

This chapter describes the request for the Modbus service Diagnostics (see Figure 23).



Used in ICommunication.BeginCommunicationRequest()

Figure 23 – ModbusDiagnosticsRequest

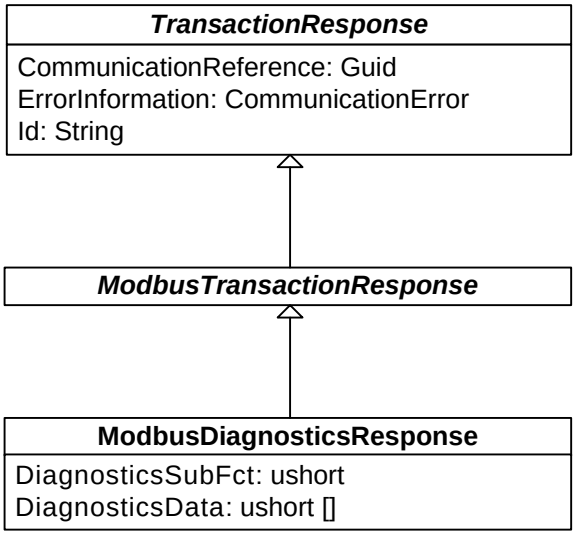
The properties of the ModbusDiagnosticsRequest data type are described in Table 31.

Table 31 – ModbusDiagnosticsRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
DiagnosticsData	Required data, depending on the sub-function type	Data
DiagnosticsSubFct	Sub-function code	Sub-function
Id	[Optional] Identifier for a single Transaction Request.	n/a

10.22 ModbusDiagnosticsResponse

This chapter describes the response for the Modbus service Diagnostics (see Figure 24).



Used in ICommunication.EndCommunicationRequest()

Figure 24 – ModbusDiagnosticsResponse

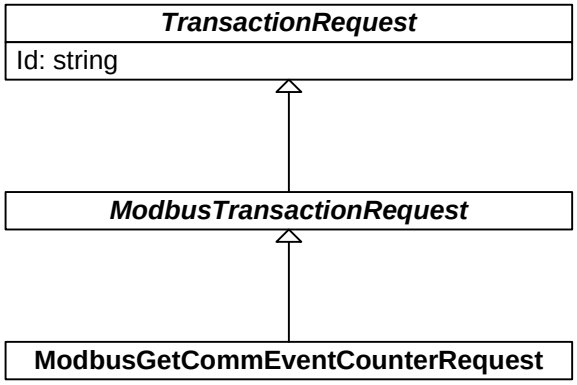
The properties of the ModbusDiagnosticsResponse data type are described in Table 32.

Table 32 – ModbusDiagnosticsResponse data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
DiagnosticsData	Required data, depending on the sub-function type	Data
DiagnosticsSubFct	Sub-function code	Sub-function
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
Id	[Optional] Identifier for a single Transaction Request.	n/a

10.23 ModbusGetCommEventCounterRequest

This chapter describes the request for the Modbus service Get Comm Event Counter (see Figure 25).



Used in ICommunication.BeginCommunicationRequest()

Figure 25 – ModbusGetCommEventCounterRequest

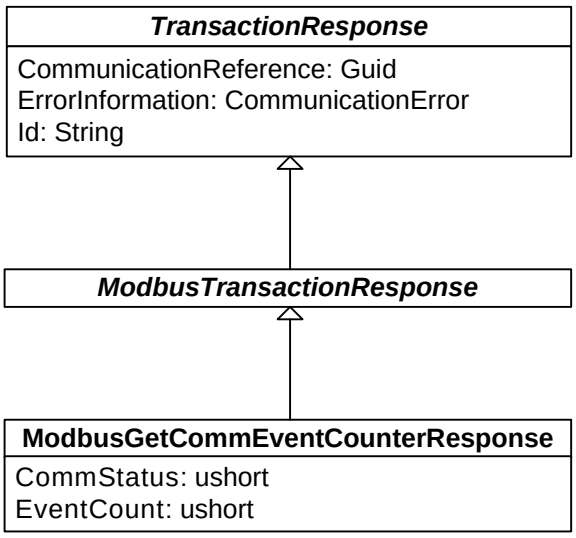
The properties of the ModbusGetCommEventCounterRequest data type are described in Table 33.

Table 33 – ModbusGetCommEventCounterRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	n/a

10.24 ModbusGetCommEventCounterResponse

This chapter describes the response for the Modbus service Get Comm Event Counter (see Figure 26).



Used in ICommunication.EndCommunicationRequest()

Figure 26 – ModbusGetCommEventCounterResponse

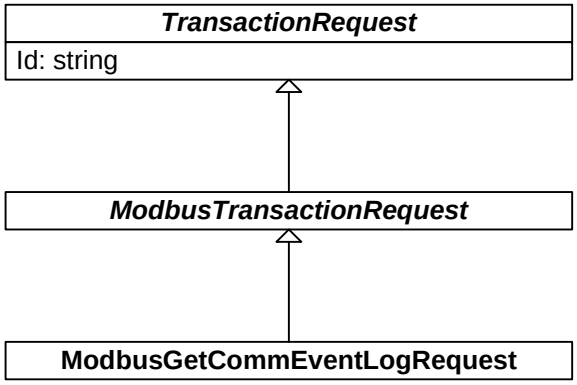
The properties of the ModbusGetCommEventCounterResponse data type are described in Table 34.

Table 34 – ModbusDiagnosticsResponse data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
CommStatus	Two-byte status information. The status information will be 0xFFFF if a previously issued program command is still being processed by the remote device (busy condition), otherwise it will be 0x0000.	Status
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
EventCount	Number of events counted by the device	Event Count
Id	[Optional] Identifier for a single Transaction Request.	n/a

10.25 ModbusGetCommEventLogRequest

This chapter describes the request for the Modbus service Get Comm Event Log (see Figure 27).



Used in ICommunication.BeginCommunicationRequest()

Figure 27 – ModbusGetCommEventLogRequest

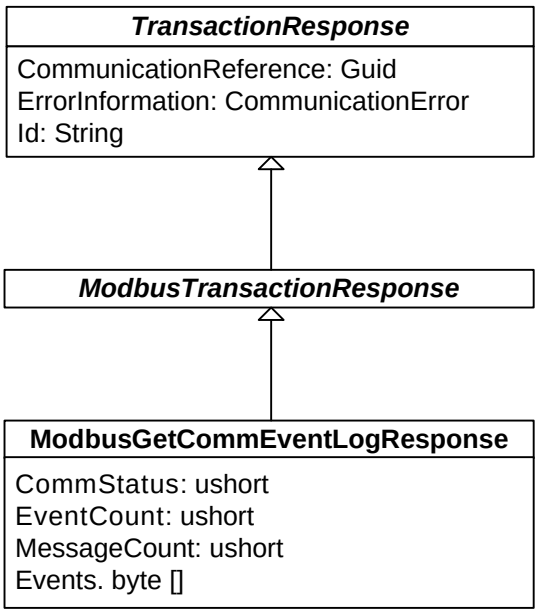
The properties of the ModbusGetCommEventLogRequest data type are described in Table 35.

Table 35 – ModbusGetCommEventLogRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	n/a

10.26 ModbusGetCommEventLogResponse

This chapter describes the response for the Modbus service Get Comm Event Log (see Figure 28).



Used in ICommunication.EndCommunicationRequest()

Figure 28 – ModbusGetCommEventLogResponse

The properties of the ModbusGetCommEventLogResponse data type are described in Table 36.

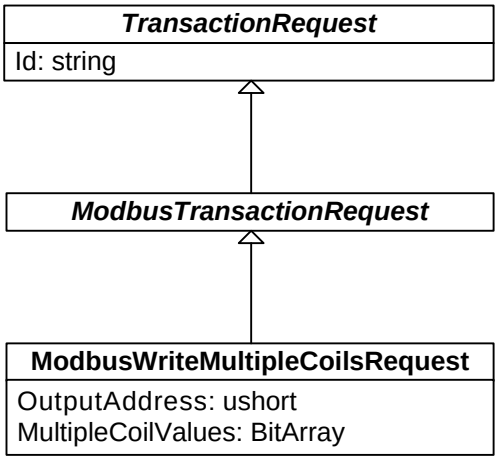
Table 36 – ModbusGetCommEventLogResponse data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
CommStatus	Two-byte status information. The status information will be 0xFFFF if a previously issued program command is still being processed by the remote device (busy condition), otherwise it will be 0x0000.	Status
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
Events	Each byte corresponds to the status of one MODBUS send or receive operation for the remote device	Events
EventCount	Number of events counted by the device	Event Count
Id	[Optional] Identifier for a single Transaction Request.	n/a

Property	Description	Equivalent IEC 61158-5-15 service parameter
MessageCount	Quantity of messages processed by the remote device since its last restart, clear counters operation, or power–up.	Message Count

10.27 ModbusWriteMultipleCoilsRequest

This chapter describes the request for the Modbus service Write Multiple Coils (see Figure 29).



Used in ICommunication.BeginCommunicationRequest()

Figure 29 – ModbusWriteMultipleCoilsRequest

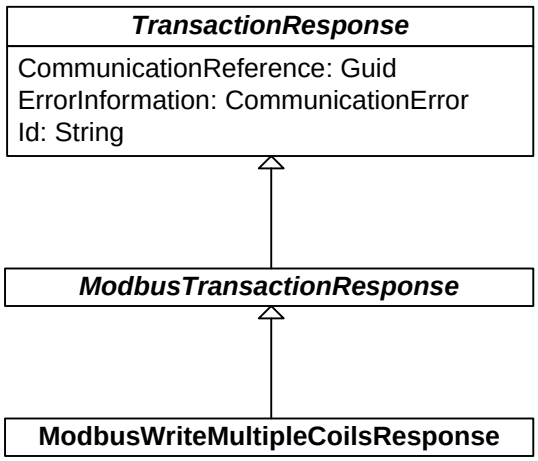
The properties of the ModbusWriteMultipleCoilsRequest data type are described in Table 37.

Table 37 – ModbusWriteMultipleCoilsRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	n/a
MultipleCoilValues	Bit-array with each coil state coded in one bit, where the first bit in the array represents the first coil to be written: “0” = FALSE or “OFF” “1” = TRUE or “ON”	Data
OutputAddress	Address of the first coil to be forced	Address of first coil

10.28 ModbusWriteMultipleCoilsResponse

This chapter describes the response for the Modbus service Write Multiple Coils (see Figure 30).



Used in ICommunication.EndCommunicationRequest()

Figure 30 – ModbusWriteMultipleCoilsResponse

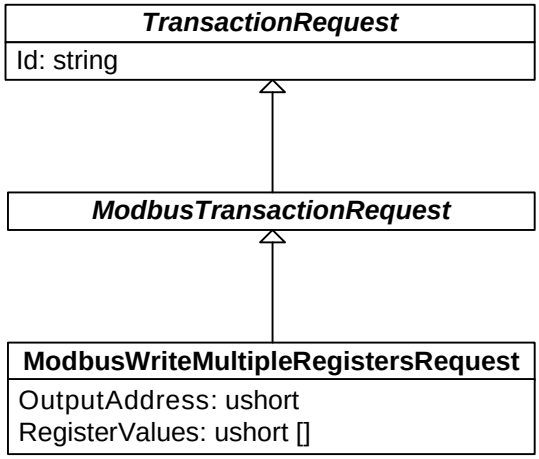
The properties of the ModbusWriteMultipleCoilsResponse data type are described in Table 38.

Table 38 – ModbusWriteMultipleCoilsResponse data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
Id	[Optional] Identifier for a single Transaction Request.	n/a

10.29 ModbusWriteMultipleRegistersRequest

This chapter describes the request for the Modbus service Write Multiple Registers (see Figure 31).



Used in ICommunication.BeginCommunicationRequest()

Figure 31 – ModbusWriteMultipleRegistersRequest

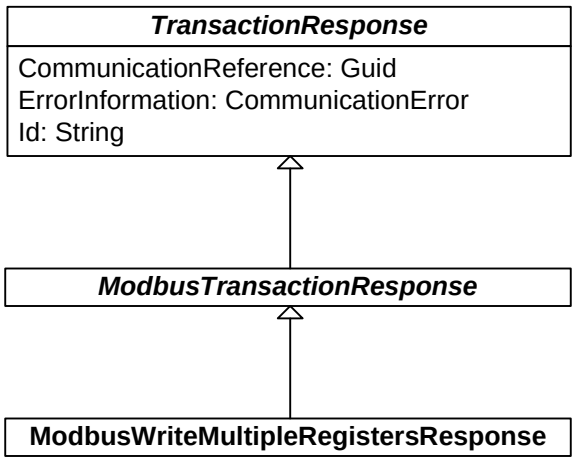
The properties of the ModbusWriteMultipleRegistersRequest data type are described in Table 39.

Table 39 – ModbusWriteMultipleRegistersRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	n/a
OutputAddress	Address of the first coil to be forced	Address of first coil
RegisterValues	Register values to be written	Data

10.30 ModbusWriteMultipleRegistersResponse

This chapter describes the response for the Modbus service Write Multiple Registers (see Figure 30).



Used in ICommunication.EndCommunicationRequest()

Figure 32 – ModbusWriteMultipleRegistersResponse

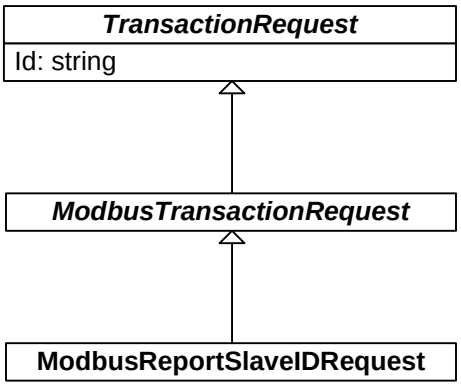
The properties of the ModbusWriteMultipleRegistersResponse data type are described in Table 38.

Table 40 – ModbusWriteMultipleRegistersResponse

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
Id	[Optional] Identifier for a single Transaction Request.	n/a

10.31 ModbusReportSlaveIDRequest

This chapter describes the request for the Modbus service Report Slave ID (see Figure 33).



Used in ICommunication.BeginCommunicationRequest()

Figure 33 – ModbusReportSlaveIDRequest

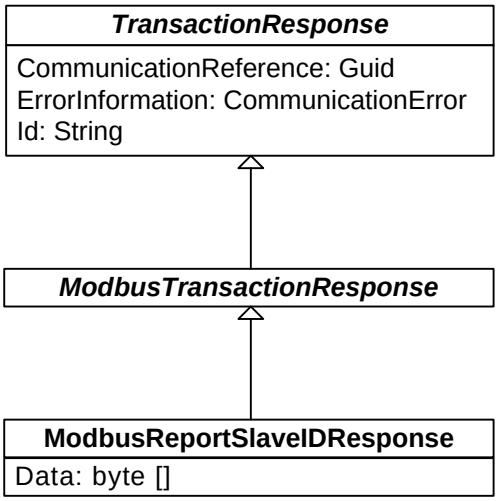
The properties of the ModbusReportSlaveIDRequest data type are described in Table 41.

Table 41 – ModbusReportSlaveIDRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	n/a

10.32 ModbusReportSlaveIDResponse

This chapter describes the response for the Modbus service Report Slave ID (see Figure 34).



Used in ICommunication.EndCommunicationRequest()

Figure 34 – ModbusReportSlaveIDResponse

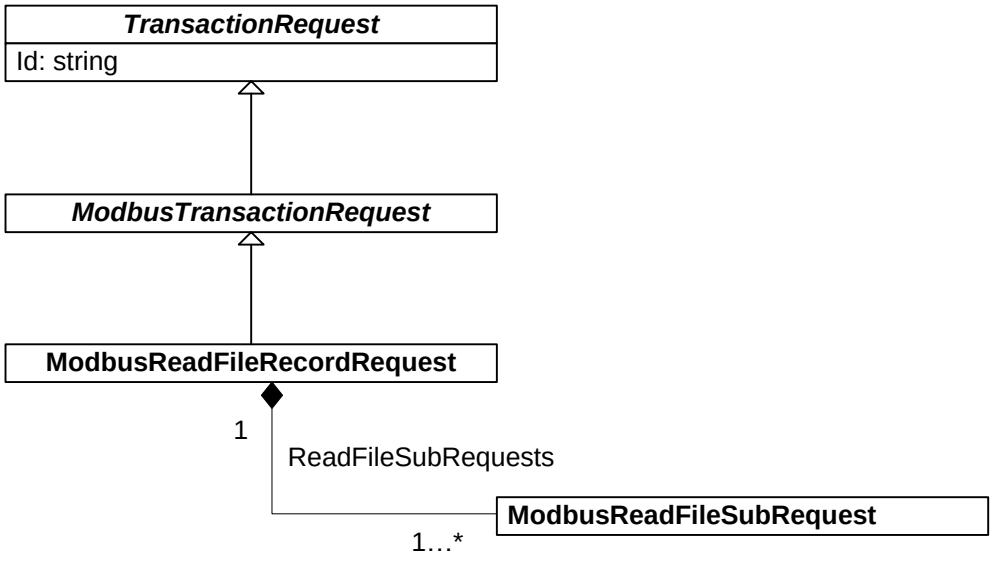
The properties of the ModbusReportSlaveIDResponse data type are described in Table 42.

Table 42 – ModbusReportSlaveIDResponse

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
Data	This attribute contains the: • Slave ID, • the Run Indicator Status (0x00 or 0xFF) • and the additional device specific data in the same format and order as defined in the MODBUS Application Protocol Specification	Slave ID, Run Indicator Status Additional data
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
Id	[Optional] Identifier for a single Transaction Request.	n/a

10.33 ModbusReadFileRecordRequest

This chapter describes the request for the Modbus service Read File Record (see Figure 35).



Used in ICommunication.BeginCommunicationRequest()

Figure 35 – ModbusReadFileRecordRequest

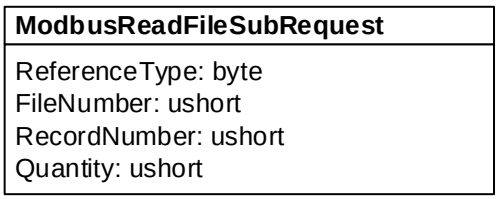
The properties of the ModbusReadFileRecordRequest data type are described in Table 43.

Table 43 – ModbusReadFileRecordRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	n/a
ModbusReadFileSubRequest	Sub-request equivalent to the Read File Record sub-request element defined in IEC 61158-5-15	n/a

10.33.1 ModbusReadFileSubRequest

This chapter describes the sub-request for the Modbus service Read File Record (see Figure 36).



Used in ICommunication.BeginCommunicationRequest()

Figure 36 – ModbusReadFileSubRequest

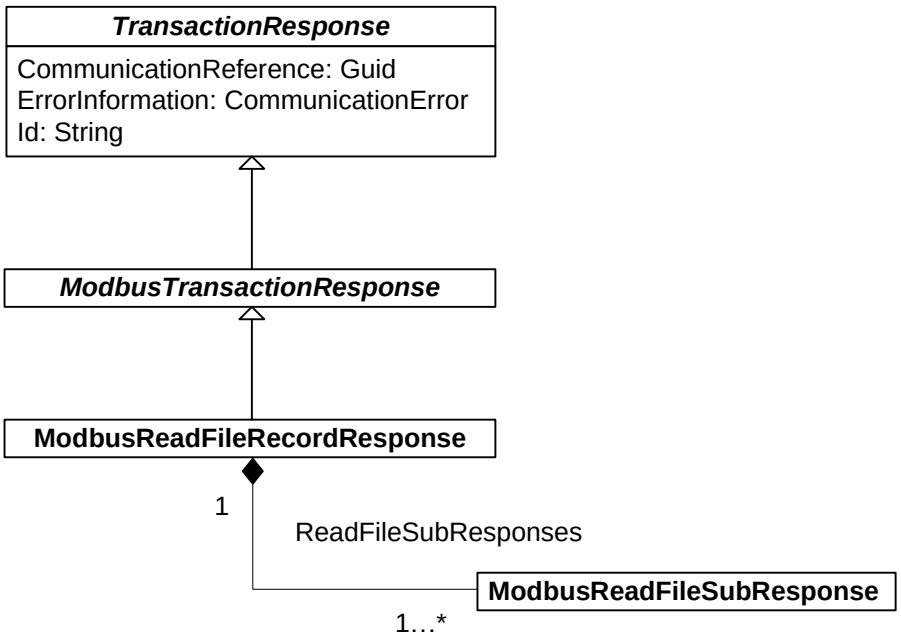
The properties of the ModbusReadFileSubRequest data type are described in Table 44.

Table 44 – ModbusReadFileSubRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
FileNumber	File number	File number
Id	[Optional] Identifier for a single Transaction Request.	n/a
Quantity	Length of the record to be read as quantity of registers	Record length
RecordNumber	Starting record number within the file	Record number
ReferenceType	Reference type	Reference type

10.34 ModbusReadFileRecordResponse

This chapter describes the response for the Modbus service Read File Record (see Figure 37).



Used in ICommunication.EndCommunicationRequest()

Figure 37 – ModbusReadFileRecordResponse

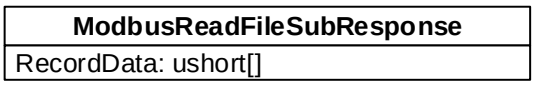
The properties of the ModbusReadFileRecordResponse data type are described in Table 45.

Table 45 – ModbusReadFileRecordResponse

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
Id	[Optional] Identifier for a single Transaction Request.	n/a
ReadFileSubResponse	Sub-response equivalent to the Read File Record sub-response element defined in IEC 61158-5-15	n/a

10.34.1 ModbusReadFileSubResponse

This chapter describes the sub-response for the Modbus service Read File Record (see Figure 38).



Used in ICommunication.EndCommunicationRequest()

Figure 38 – ModbusReadFileSubResponse

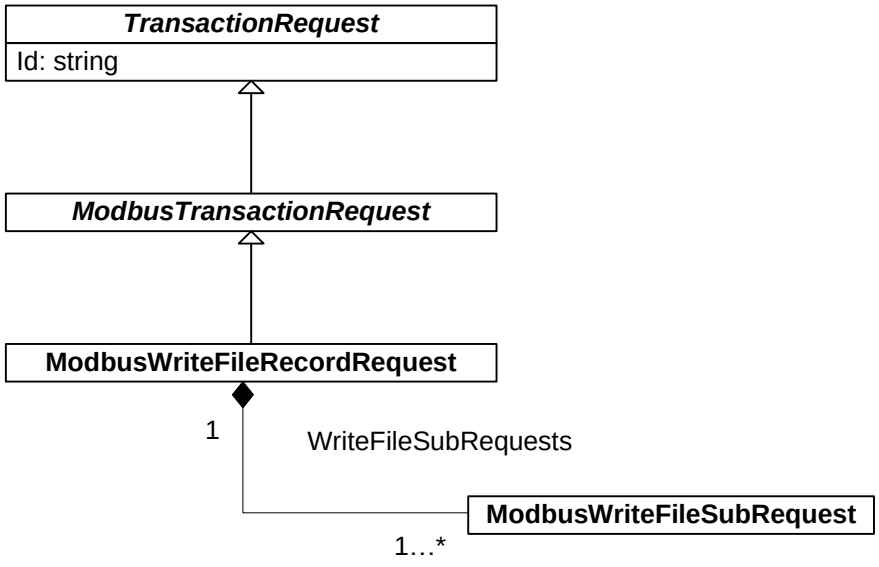
The properties of the ModbusReadFileSubResponse data type are described in Table 46.

Table 46 – ModbusReadFileSubResponse

Property	Description	Equivalent IEC 61158-5-15 service parameter
RecordData	Record data	Record data array

10.35 ModbusWriteFileRecordRequest

This chapter describes the request for the Modbus service Write File Record (see Figure 39).



Used in ICommunication.BeginCommunicationRequest()

Figure 39 – ModbusWriteFileRecordRequest

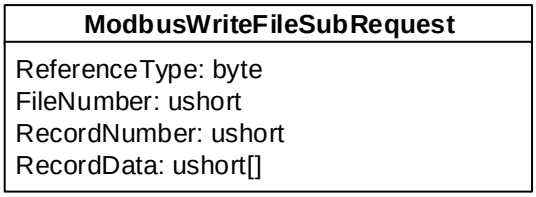
The properties of the ModbusWriteFileRecordRequest data type are described in Table 47.

Table 47 – ModbusWriteFileRecordRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	n/a
ModbusWriteFileSubRequest	Reference type	Sub-request equivalent to the Write File Record sub-request element defined in IEC 61158-5-15

10.35.1 ModbusWriteFileSubRequest

This chapter describes the sub-request for the Modbus service Write File Record (see Figure 40).



Used in ICommunication.BeginCommunicationRequest()

Figure 40 – ModbusWriteFileSubRequest

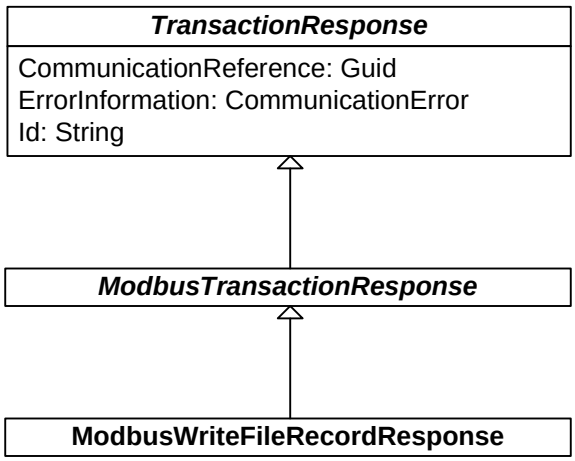
The properties of the ModbusWriteFileSubRequest data type are described in Table 47.

Table 48 – ModbusWriteFileSubRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
FileNumber	File number	File number
RecordData	Record data	Record data array
RecordNumber	Starting record number within the file	Record number
ReferenceType	Reference type	Reference type

10.36 ModbusWriteFileRecordResponse

This chapter describes the response for the Modbus service Write File Record (see Figure 41).



Used in ICommunication.EndCommunicationRequest()

Figure 41 – ModbusWriteFileRecordResponse

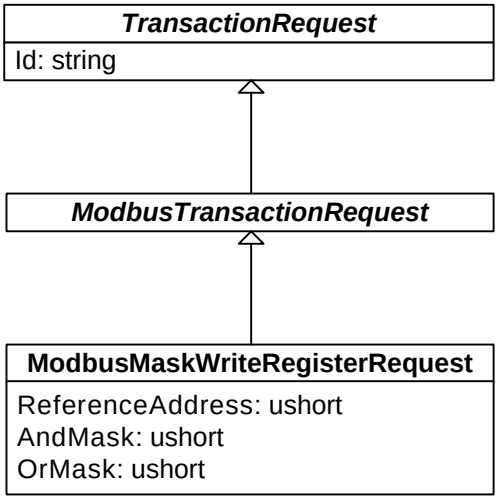
The properties of the ModbusWriteFileRecordResponse data type are described in Table 49.

Table 49 – ModbusWriteFileRecordResponse

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
Id	[Optional] Identifier for a single Transaction Request.	n/a

10.37 ModbusMaskWriteRegisterRequest

This chapter describes the request for the Modbus service Mask Write Register (see Figure 42).



Used in ICommunication.BeginCommunicationRequest()

Figure 42 – ModbusMaskWriteRegisterRequest

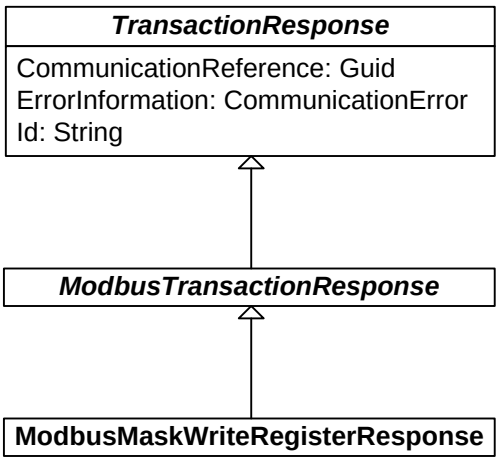
The properties of the ModbusMaskWriteRegisterRequest data type are described in Table 50.

Table 50 – ModbusMaskWriteRegisterRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
AndMask	AND mask	AND Mask
Id	[Optional] Identifier for a single Transaction Request.	n/a
OrMask	OR mask	OR Mask
ReferenceAddress	Address of the holding register the mask shall be applied to	Address of first holding register to write

10.38 ModbusMaskWriteRegisterResponse

This chapter describes the response for the Modbus service Mask Write Register (see Figure 43).



Used in ICommunication.EndCommunicationRequest()

Figure 43 – ModbusMaskWriteRegisterResponse

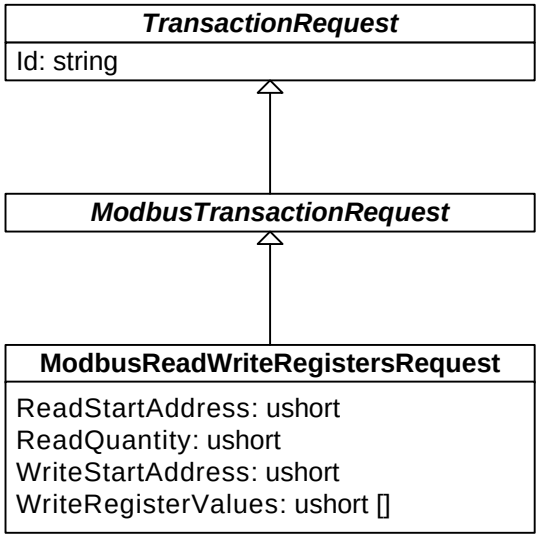
The properties of the ModbusMaskWriteRegisterResponse data type are described in Table 51.

Table 51 – ModbusMaskWriteRegisterResponse

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
Id	[Optional] Identifier for a single Transaction Request.	n/a

10.39 ModbusReadWriteRegistersRequest

This chapter describes the request for the Modbus service Read/Write Multiple Registers (see Figure 44).



Used in ICommunication.BeginCommunicationRequest()

Figure 44 – ModbusReadWriteRegistersRequest

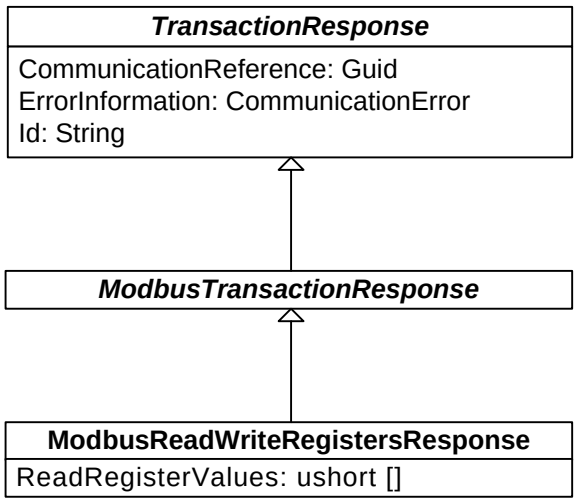
The properties of the ModbusReadWriteRegistersRequest data type are described in Table 52.

Table 52 – ModbusReadWriteRegistersRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	n/a
ReadQuantity	Number of holding registers to be read	Quantity of holding registers to read
ReadStartAddress	Address of the first holding register to be read	Address of first holding register to read
WriteRegisterValues	Register values to be written	Data
WriteStartAddress	Address of the first holding register to be written	Address of first holding register to write

10.40 ModbusReadWriteRegistersResponse

This chapter describes the response for the Modbus service Read/Write Multiple Registers (see Figure 45).



Used in ICommunication.EndCommunicationRequest()

Figure 45 – ModbusReadWriteRegistersResponse

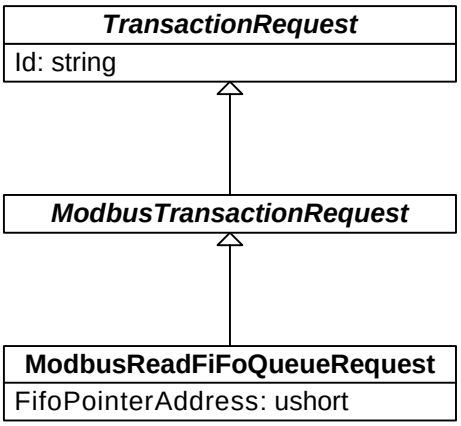
The properties of the ModbusReadWriteRegistersResponse data type are described in Table 53.

Table 53 – ModbusReadWriteRegistersResponse

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
Id	[Optional] Identifier for a single Transaction Request.	n/a
ReadRegisterValues	Read holding register values	Data

10.41 ModbusReadFiFoQueueRequest

This chapter describes the request for the Modbus service Read FiFo Queue (see Figure 46).



Used in ICommunication.BeginCommunicationRequest()

Figure 46 – ModbusReadFiFoQueueRequest

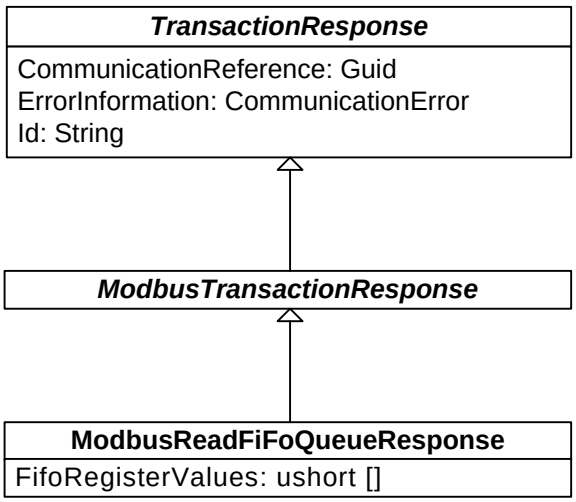
The properties of the ModbusReadFiFoQueueRequest data type are described in Table 54.

Table 54 – ModbusReadFiFoQueueRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	n/a
FifoPointerAddress	Address of the first register to be read in a First-In-First-Out (FIFO) queue of registers.	Address of FIFO queue

10.42 ModbusReadFiFoQueueResponse

This chapter describes the response for the Modbus service Read FIFO Queue (see Figure 47).



Used in ICommunication.EndCommunicationRequest()

Figure 47 – ModbusReadFiFoQueueResponse

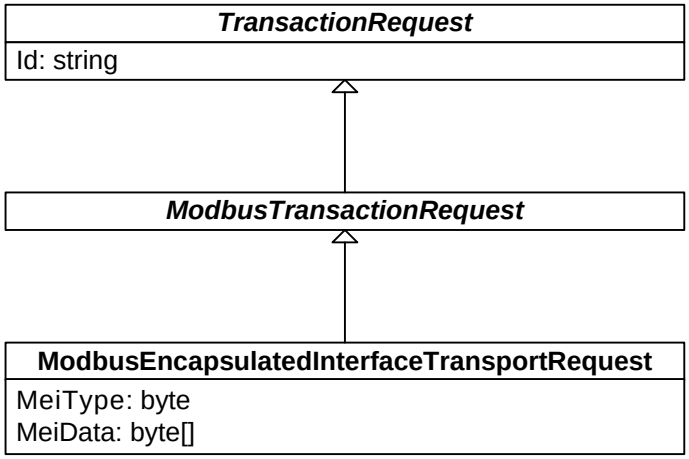
The properties of the ModbusReadFiFoQueueResponse data type are described in Table 55.

Table 55 – ModbusReadFiFoQueueResponse

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
Id	[Optional] Identifier for a single Transaction Request.	n/a
FifoRegisterValues	Register values read from the FIFO queue	Data

10.43 ModbusEncapsulatedInterfaceTransportRequest

This chapter describes the request for the Modbus service Encapsulated Interface Transport (see Figure 48).



Used in ICommunication.BeginCommunicationRequest()

Figure 48 – ModbusEncapsulatedInterfaceTransportRequest

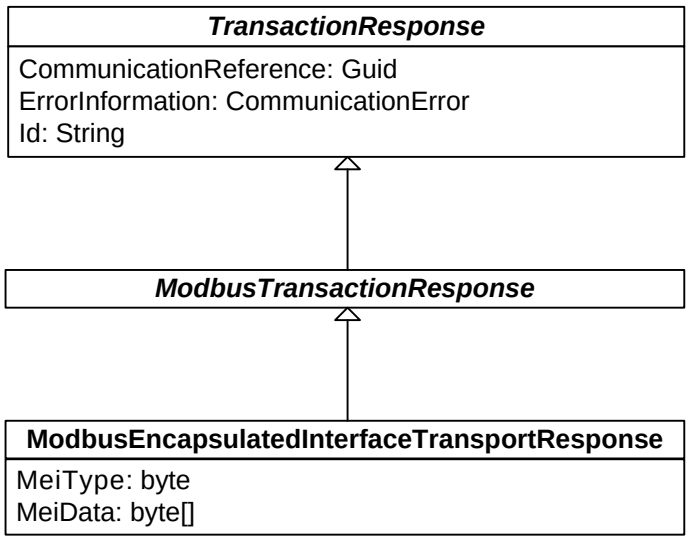
The properties of the ModbusEncapsulatedInterfaceTransportRequest data type are described in Table 56.

Table 56 – ModbusEncapsulatedInterfaceTransportRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	n/a
MeiData	Modbus encapsulated interface type specific data	MEI type specific data
MeiType	Modbus encapsulated interface type (MEI type)	MEI type

10.44 ModbusEncapsulatedInterfaceTransportResponse

This chapter describes the response for the Modbus service Encapsulated Interface Transport (see Figure 49).



Used in ICommunication.EndCommunicationRequest()

Figure 49 – ModbusEncapsulatedInterfaceTransportResponse

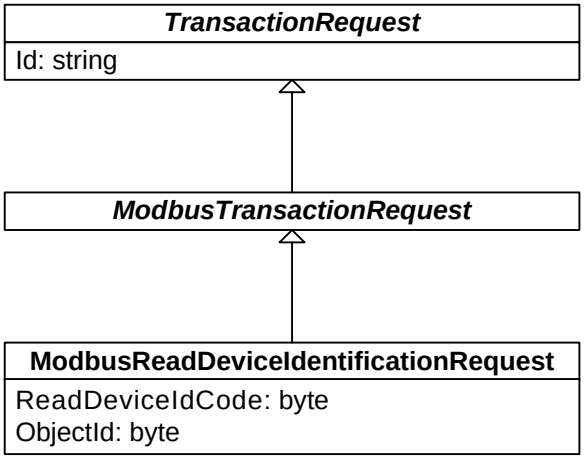
The properties of the ModbusEncapsulatedInterfaceTransportResponse data type are described in Table 57.

Table 57 – ModbusEncapsulatedInterfaceTransportResponse

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
Id	[Optional] Identifier for a single Transaction Request.	n/a
MeiData	Modbus encapsulated interface type specific data	MEI type specific data
MeiType	Modbus encapsulated interface type (MEI type)	MEI type

10.45 ModbusReadDeviceIdentificationRequest

This chapter describes the request for the Modbus service Read Device Identification (see Figure 50).



Used in ICommunication.BeginCommunicationRequest()

Figure 50 – ModbusReadDeviceIdentificationRequest

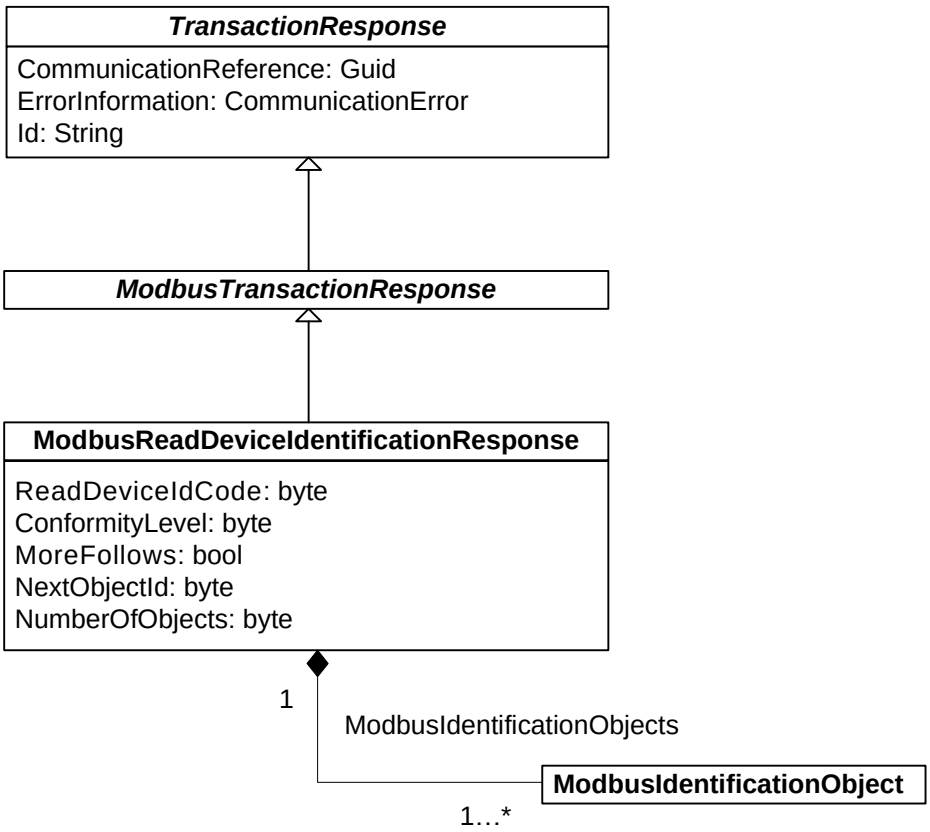
The properties of the ModbusReadDeviceIdentificationRequest data type are described in Table 58.

Table 58 – ModbusReadDeviceIdentificationRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	n/a
ReadDeviceIDCode	The parameter "Read Device ID code" allows to define four access types: 01: request to get the basic device identification (stream access) 02: request to get the regular device identification (stream access) 03: request to get the extended device identification (stream access) 04: request to get one specific identification object (individual access)	Read device ID code
ObjectId	Identification of the first object to obtain	Requested object ID

10.46 ModbusReadDeviceIdentificationResponse

This chapter describes the response for the Modbus service Read Device Identification (see Figure 51).



Used in ICommunication.EndCommunicationRequest()

Figure 51 – ModbusReadDeviceIdentificationResponse

The properties of the ModbusReadDeviceIdentificationResponse data type are described in Table 59.

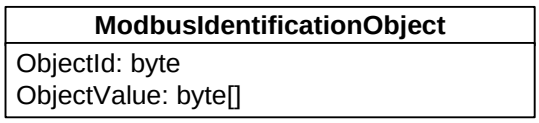


Figure 52 – ModbusIdentificationObject

The properties of the ModbusIdentificationObject data type are described in Table 60.

Table 59 – ModbusReadDeviceIdentificationResponse

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ConformityLevel	Identification conformity level of the device and type of supported access 0x01: basic identification (stream access only) 0x02: regular identification (stream access only) 0x03: extended identification (stream access only) 0x81: basic identification (stream access and individual access) 0x82: regular identification (stream access and individual access) 0x83: extended identification (stream access and individual access)	Conformity level
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
Id	[Optional] Identifier for a single Transaction Request.	n/a
ModbusIdentificationObject	Identification objects of the Read Device Identification service response provided by the Encapsulated Interface defined in IEC 61158-5-15	n/a
MoreFollows	In case of ReadDeviceIdCode 01, 02 or 03 (stream access), if the identification data does not fit into a single response and several request/response transactions may be required: “0” : no more objects are available “1” : other identification objects are available and further Modbus transactions are required In case of ReadDeviceIdCode 04 (individual access), this field shall be set to “0”.	More-available flag
NextObjectid	“0”, if no more identification objects are available (moreFollows=”0”) Identification of the next object to be obtained, if more identification objects are available (moreFollows=”1”)	Next object ID
NumberOfObjects	Number of identification objects returned in this response (for an individual access, numberOfObjects=”1”)	Number of objects

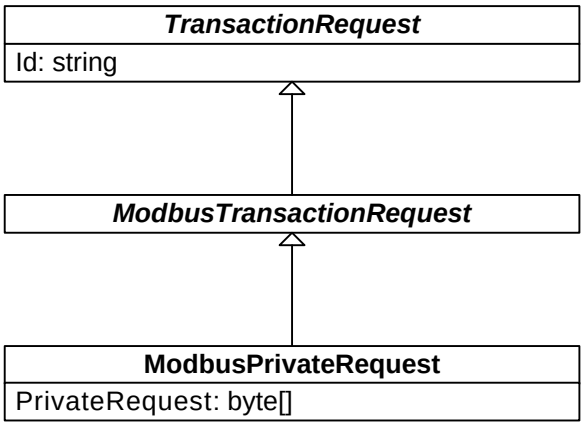
Property	Description	Equivalent IEC 61158-5-15 service parameter
ReadDeviceIdCode	The parameter "Read Device ID code" allows to define four access types: 01: request to get the basic device identification (stream access) 02: request to get the regular device identification (stream access) 03: request to get the extended device identification (stream access) 04: request to get one specific identification object (individual access)	Read device ID code
ModbusIdentificationObject	Identification conformity level of the device and type of supported access 0x01: basic identification (stream access only) 0x02: regular identification (stream access only) 0x03: extended identification (stream access only) 0x81: basic identification (stream access and individual access) 0x82: regular identification (stream access and individual access) 0x83: extended identification (stream access and individual access)	Conformity level

Table 60 – ModbusIdentificationObject

Property	Description	Equivalent IEC 61158-5-15 service parameter
ObjectId	Identification of the returned object	Returned object ID
ObjectValue	Object value	Object value

10.47 ModbusPrivateRequest

This chapter describes the request for a private Modbus service (see Figure 53).



Used in ICommunication.BeginCommunicationRequest()

Figure 53 – ModbusPrivateRequest

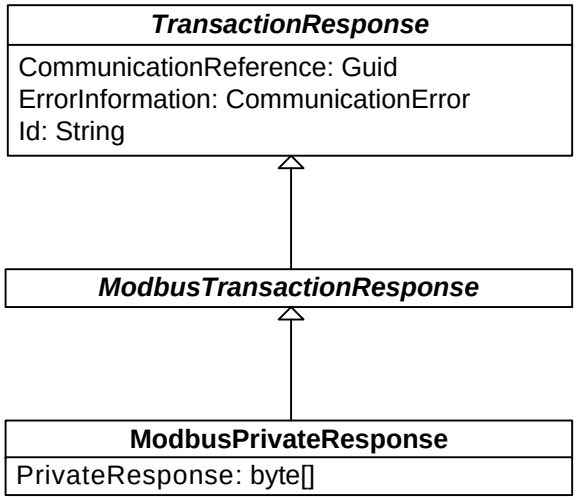
The properties of the ModbusPrivateRequest data type are described in Table 61.

Table 61 – ModbusPrivateRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	n/a
PrivateRequest	Sequence of hexadecimal digits representing the private Modbus request	n/a

10.48 ModbusPrivateResponse

This chapter describes the response for a private Modbus service (see Figure 54).



Used in ICommunication.EndCommunicationRequest()

Figure 54 – ModbusPrivateResponse

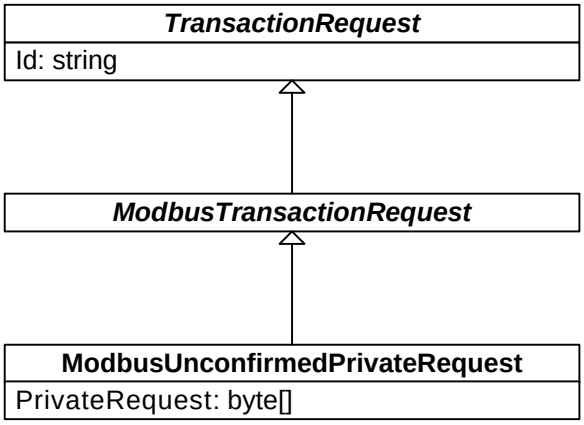
The properties of the ModbusPrivateResponse data type are described in Table 62.

Table 62 – ModbusPrivateResponse

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
Id	[Optional] Identifier for a single Transaction Request.	n/a
PrivateResponse	Sequence of hexadecimal digits representing the private Modbus response	n/a

10.49 ModbusUnconfirmedPrivateRequest

This chapter describes the request for an unconfirmed private Modbus service (see Figure 55).



Used in ICommunication.BeginCommunicationRequest()

Figure 55 – ModbusUnconfirmedPrivateRequest

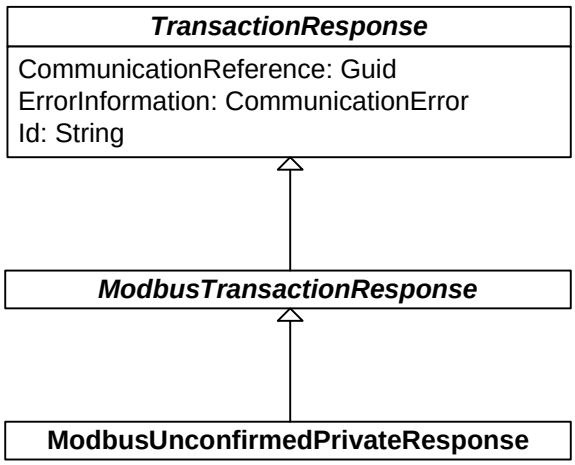
The properties of the ModbusUnconfirmedPrivateRequest data type are described in Table 63.

Table 63 – ModbusUnconfirmedPrivateRequest data type

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	n/a
PrivateRequest	Sequence of hexadecimal digits representing the private Modbus request	n/a

10.50 ModbusUnconfirmedPrivateResponse

This chapter describes the response for an unconfirmed private Modbus service (see Figure 56).



Used in ICommunication.EndCommunicationRequest()

Figure 56 – ModbusUnconfirmedPrivateResponse

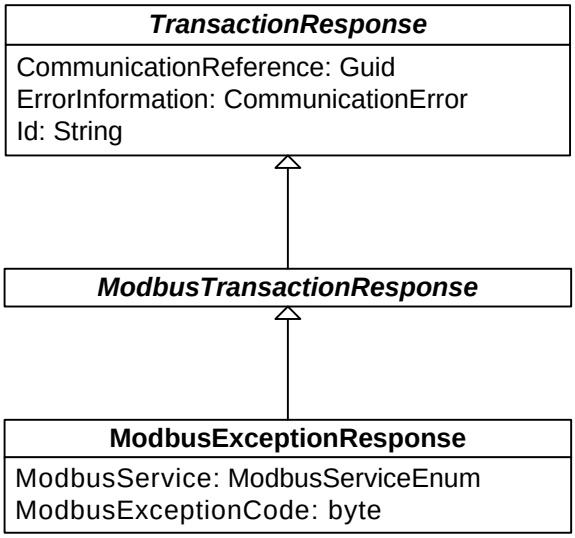
The properties of the ModbusUnconfirmedPrivateResponse data type are described in Table 64.

Table 64 – ModbusUnconfirmedPrivateResponse

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
Id	[Optional] Identifier for a single Transaction Request.	n/a

10.51 ModbusExceptionResponse

This chapter describes the Modbus Exception Response (see Figure 57).



Used in ICommunication.EndCommunicationRequest()

Figure 57 – ModbusExceptionResponse

The properties of the ModbusExceptionResponse data type are described in Table 65.

Table 65 – ModbusExceptionResponse

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	n/a
Id	[Optional] Identifier for a single Transaction Request.	n/a
ModbusService	Enumeration of Modbus service names, used to identify the Modbus service on which the exception occurred.	n/a
ModbusExceptionCode	Modbus Exception Code	Exception code

11 Data types for process data information

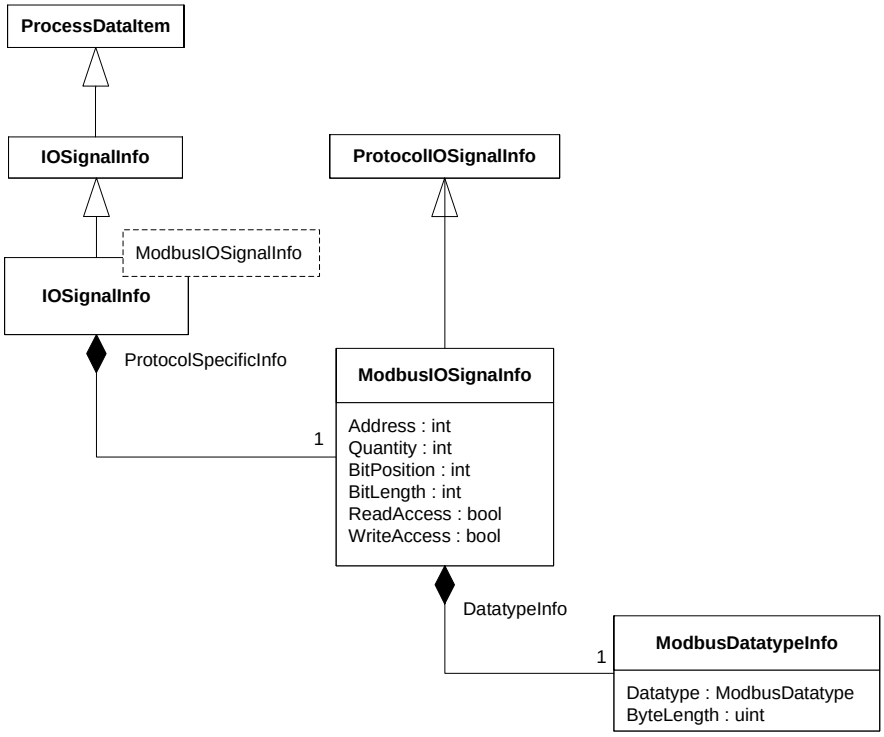
11.1 General

The process data information of a DTM represents the “Device Variables”, available on that device. A Process Control System (i.e. some external system that monitors values on a device) can query the DTM's process data information via the IProcessData interface. The process data describes the process values such that an external system can use the

information to access and interpret the values from the device during normal device runtime. The external system might not use FDT to access the values.

11.2 ModbusIOSignalInfo

This is the Modbus specific implementation of the abstract class ProtocolIOSignalInfo (see Figure 58).



Used in: IProcessData.<ProcessData>()
IProcessData.SetIOSignalInfo()

Figure 58 – ModbusIOSignalInfo

The properties of the ModbusIOSignalInfo data type are described in Table 66

Table 66 – ModbusIOSignalInfo data type

Property	Description
Address	Address of the discrete input, coil or register which shall be accessed. In case that a range of these Modbus data items shall be accessed, the address data type contains the address of the first Modbus data item within this range.
BitLength	Number of bits
BitPosition	If the access to single bits or some collections of bits is needed, it can be defined the BitPosition and BitLength property
ModbusDatatypeInfo	The data type of the IO signal.
Quantity	Number of discrete inputs, coils or registers which represent the channel object in the device
ReadAccess	Defines whether the IOSignal can be read or not
WriteAccess	Defines whether the IOSignal can be written or not

The Modbus data model defines four primary data types:

- Discrete Inputs
- Coils
- Input Registers
- Holding Registers

The bit access is defined for Discrete Inputs and Coils. Nevertheless, most devices use Holding Register for input and output data, also with bit access.

If the access to single bits or some collections of bits is needed, it can be defined inside the ModbusIOSignalInfo data type with the BitPosition and BitLength properties. It is also feasible to define the bits of a Holding Register and additionally the whole word of the same Holding Register as an IO signal. In other words, you can define 17 process data objects for a single holding register. On the other hand, it is not mandatory to describe all bits of a register word.

11.3 Mapping of Modbus data types to FDT data types

The data type mapping defines how Modbus IO Signals are mapped to PLC applications using data types defined in IEC 61131-3.

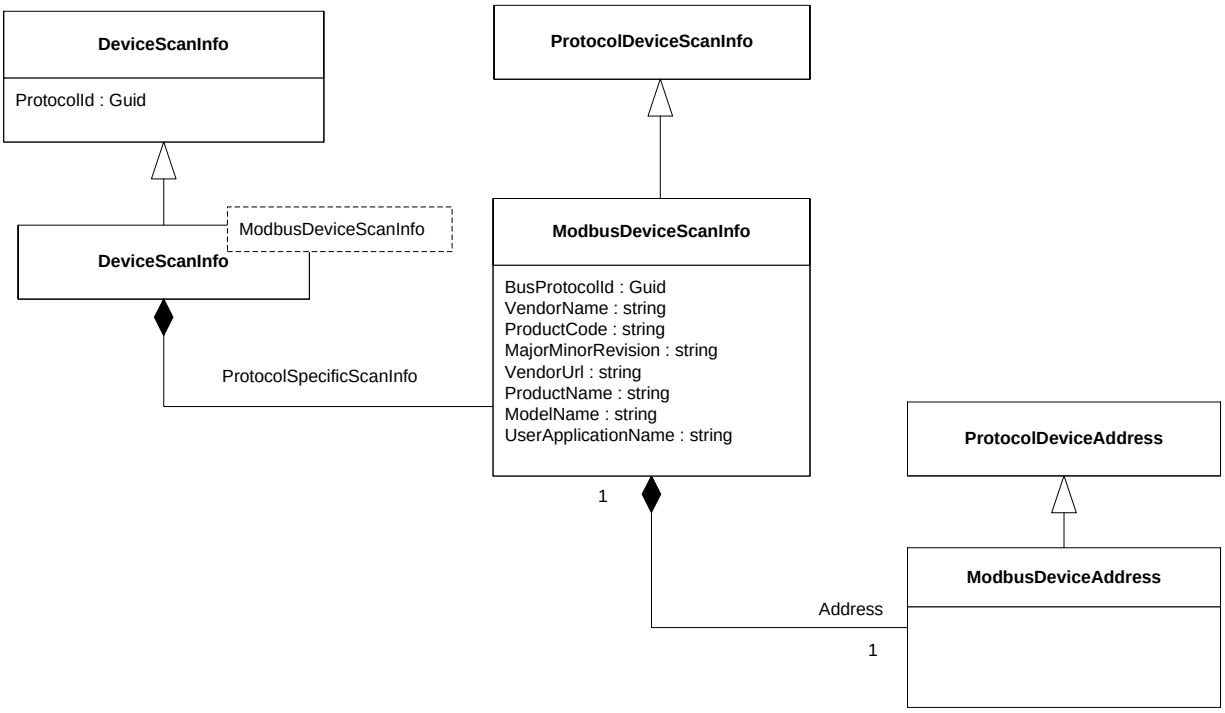
See Table 4 for a mapping of data types.

12 Device identification

This chapter defines identification relevant protocol specific data types.

12.1 ModbusDeviceScanInfo data type

This is the Modbus specific implementation of the abstract class ProtocolDeviceScanInfo (see Figure 59).



Used in IDtmScanning.EndScanRequest()

Figure 59 – ModbusDeviceScanInfo

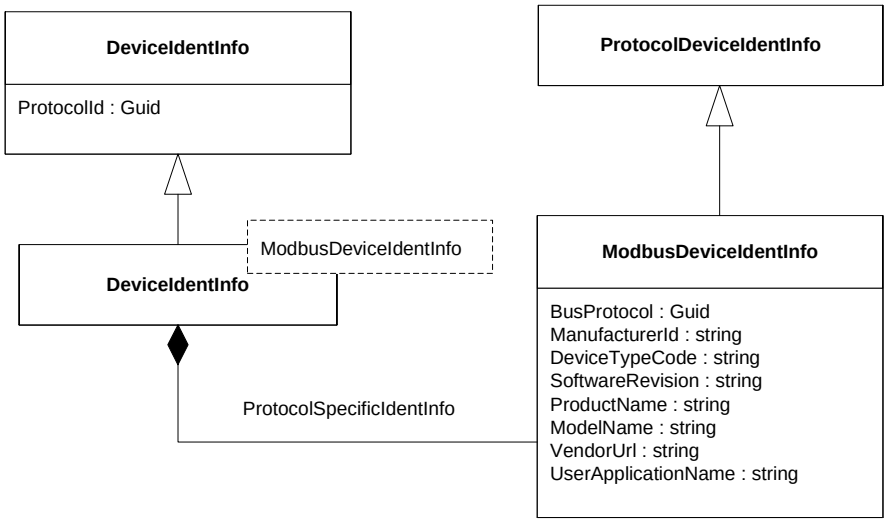
The properties of the ModbusDeviceScanInfo data type are described in Table 67.

Table 67 – ModbusDeviceScanInfo data type

Property	Description
Address	The bus address of the device.
BusProtocolId	Can be set to either Modbus Serial or Modbus TCP. This information is provided by the Communication Channel (based on the ScanRequest)
MajorMinorRevision	Information of the Modbus protocol specific identification object MajorMinorRevision
ModelName	Information of the Modbus protocol specific identification object ModelName
ProductCode	Information of the Modbus protocol specific identification object ProductCode
ProductName	Information of the Modbus protocol specific identification object ProductName
UserApplicationName	Information of the Modbus protocol specific identification object UserApplicationName
VendorName	Information of the Modbus protocol specific identification object VendorName
VendorUrl	Information of the Modbus protocol specific identification object VendorUrl

12.2 ModbusDeviceIdentInfo data type

Modbus DTMs that may connect to a Modbus Communication Channel (e.g. Device DTMs and Gateway DTMs) shall provide information, which may be used to identify the corresponding devices on the fieldbus. This section describes the offline information (see Figure 60). For DTM assigned after Fieldbus-scanning, the frame application can check in a protocol independent way if the identification of a scanned device type (DeviceScanInfo) matches the supported DeviceInfo.



Used in IDtmInformation.GetDeviceIdentInfo()

Figure 60 – ModbusDeviceIdentInfo

The properties of the ModbusDeviceIdentInfo data type are described in Table 68.

Table 68 – ModbusDeviceIdentInfo data type

Property	Description
BusProtocol	The unique identifier of either Modbus Serial or Modbus TCP
DeviceTypeCode	Modbus product code
ManufacturerId	Modbus vendor name
ModelName	Modbus model name
ProductName	Modbus product name
SoftwareRevision	Modbus Major Minor Version
UserApplicationName	Modbus user application name
VendorUrl	Modbus Vendor URL

The information described here will be used to match the information retrieved from Communication Channels via the method ICommunication.<ScanRequest()>. This match is executed by device independent software. That is why it is important to provide in ModbusDeviceIdentInfo information that can be matched with the ModbusDeviceScanInfo information. Developers of DTMs need to consider which information the devices will provide (see 12.3).

12.3 Mapping of Information Source

The following Table 69 – Protocol specific mapping of scan information defines the semantics of ModbusDeviceScanInfo properties and how this information is mapped to predefined properties of DeviceScanInfo.

The Communication channel shall read these values from the device (if implemented in the device) and write them into the properties of ModbusDeviceScanInfo.

Table 69 – Protocol specific mapping of scan information

ModbusDeviceScanInfo Property Name	Mapped DeviceScanInfo Property Name	Data Request in physical device	Protocol Specific Name (␣ = space)	Modbus Data Format	Specific Reference
-	ProtocolIdentificationProfile	-	-	-	-
Address	Address.BusAddress	Modbus device address	-	-	-
BusProtocolId	ProtocolId	Set by CommunicationChannel	-	-	-
-	PhysicalLayer	-	-	-	-
VendorName	ManufacturerId	Function Code 43 (0x2B) / MEI 14 (0x0E), Object Id 0x00	Vendor␣Name	String	[2] Section 6.21
ProductCode	DeviceTypeId	Function Code 43 (0x2B) / MEI 14 (0x0E), Object Id 0x01	Product␣Code	String	[2] Section 6.21
MajorMinorRevision	SoftwareRevision	Function Code 43 (0x2B) / MEI 14 (0x0E), Object Id 0x02	Major␣Minor␣Revision	String	[2] Section 6.21
Protocol Specific Properties:					
VendorUrl	ProtocolSpecificProperty	Function code 43 (0x2B) / MEI 14 (0x0E), Object Id 0x03	Vendor␣URL	String	[2] Section 6.21
ProductName	ProtocolSpecificProperty	Function code 43 (0x2B) / MEI 14 (0x0E), Object Id 0x04	Product␣Name	String	[2] Section 6.21
ModelName	ProtocolSpecificProperty	Function code 43 (0x2B) / MEI 14 (0x0E), Object Id 0x05	Model␣Name	String	[2] Section 6.21
UserApplicationName	ProtocolSpecificProperty	Function code 43 (0x2B) / MEI 14 (0x0E), Object Id 0x06	User␣Application␣Name	String	[2] Section 6.21

The following Table 70 – Profile specific mapping of identification information defines the semantics of ModbusDeviceIdentInfo properties and how this information is mapped to predefined properties of DeviceIdentInfo.

Table 70 – Profile specific mapping of identification information

ModbusDeviceIdentInfo Property Name	Mapped DeviceIdentInfo Property Name	Data Request in physical device	Protocol Specific Name (␣ = space)	Modbus Data Format	Specific Reference
BusProtocol	ProtocolId	Set by CommunicationChannel	-	-	-
-	ProtocolIdentificationProfile	-	-	-	-
ManufacturerId	ManufacturerId	Function Code 43 (0x2B) / MEI 14 (0x0E), Object Id 0x00	Vendor␣Name	String	[2] Section 6.21
DeviceTypeCode	DeviceTypeId	Function Code 43 (0x2B) / MEI 14 (0x0E), Object Id 0x01	Product␣Code	String	[2] Section 6.21
SoftwareRevision	SoftwareRevision	Function Code 43 (0x2B) / MEI 14 (0x0E), Object Id 0x02	Major␣Minor␣Revision	String	[2] Section 6.21
Protocol Specific Properties:					
VendorUrl	ProtocolSpecificProperty	Function Code 43 (0x2B) / MEI 14 (0x0E), Object Id 0x03	Vendor␣URL	String	[2] Section 6.21
ProductName	ProtocolSpecificProperty	Function Code 43 (0x2B) / MEI 14 (0x0E), Object Id 0x04	Product␣Name	String	[2] Section 6.21
ModelName	ProtocolSpecificProperty	Function Code 43 (0x2B) / MEI 14 (0x0E), Object Id 0x05	Model␣Name	String	[2] Section 6.21
UserApplicationName	ProtocolSpecificProperty	Function Code 43 (0x2B) / MEI 14 (0x0E), Object Id 0x06	User␣Application␣Name	String	[2] Section 6.21

The ModbusDeviceIdentInfo properties may have either a single value which must exactly match the supported device, or a range of matching values may be defined in regular expressions.

13 References

[1] FDT 2.1 Specification V1.01, March 2018, Order No. of FDT Group: 0001-0008-001

[2] MODBUS Application Protocol Specification V1.1b3, Modbus.org, April 26, 2012