ASIMP-TRDP

ASIMP-TRDP protocol description

Programmer's manual

Version 1.03

Producer: AMiT, spol. s r. o.
Naskové 1100/3, 150 00 Praha
amit-transportation.com

Technical support: support@amit.cz

Contents

History of revisions

Document name: asimp-trdp_ms_en_103.pdf

Version	Date	Author of change	Changes
100	20. 07. 2018	Humpál D.	New document.
101	06. 11. 2018	Humpál D.	Implementation requirements changes, PD.putData function description added, MD.indicate second occurrence deleted the necessary functionality is covered by the remaining MD.indicate, The indicate functions no longer expect a response from host
102	07. 11. 2018	Humpál D.	Added TRDP module control functions. Enhanced data structure descriptions. Minor aesthetic changes
103	08. 04 2019	Bělohlávek A.	Added netmask and gateway to trdp module settings. Added description of ASIM/IP identification and configuration frames. Added description of firmware upload and service mode.

Related documentation

1. **IEC 61375-2-3 ed1.0:2005**

1 Abbreviations

ASIMP Amit Simple Ip Multicast Protocol

MD Message Data

PD Process Data

TRDP Train Real Time Data Protocol

2 Scope

Document includes description of RM-TRDP maintenances and control over ASIMP/IP protocol. ASIMP protocol is used for communication between TRDP module and host system. Host accesses all services using this protocol, except firmware update which is handled over FTP. TRDP services is accessed with ASIMP-TRDP protocol based on ASIMP.

3 ASIMP/IP protocol description

The protocol is based on UDP. Both broadcast and unicast are supported. The protocol is request reply oriented.

RM-TRDP acts as ASIMP/IP sever listening on port 75. Client (host CPU) is connected from any port supplied by its OS. RM-TRDP send its replies to this port.

Unsolicited messages form RM-TRDP are send to its default gateway (usually host CPU address) and port 75.

3.1 Default setting

By default, RM-TRDP host side ethernet is configured as follows:

- Address 172.0.0.1
- Netmask 255.255.255.0
- Gateway 172.0.0.2

3.2 Data types

Data type	Length in bytes
Byte	1
Word	2
DWord	4
Byte[N]	N*1
Word[N]	N*2
DWord[N]	N*4
String[N]	N*1

All numeric data types are little-endian encoded. Little-endian coding is related to protocol fields only. Array data types (Byte[N], Word[N], DWord[N], String[N]) will use little endian encoding for the individual values. The array however shall use natural flow of data.
EXAMPLE String[5](null terminated) shall be represented as: ‘a’ ’b’ ‘c’ ‘\0’ ‘\0’

EXAMPLE Word[2](13, 26) shall be represented as: 0d 00 1a 00

Application data shall be coded in accordance with IEC 61375-2-3 ed1.0:2015 standard.

3.3 Protocol structure

ASIMP/IP protocol consists of common ASIMP header, data part and optional checksum.

3.3.1 ASIMP header

ASIMP header has the following format:

Packet offset	Local offset	Field	Type	Description
+0	+0	sessionId	Word	session identifier
+2	+2	comType	Byte	type of communication
+3	+3	frameParam	Byte	parameters of used communication frame
+4	+4	address	Byte	device address
+5	+5	frameType	Byte	type of used communication frame
+6	+6	length	Word	byte length of dataset
+8	+8	dataset	Byte[N]	user data set
+8+N	+8+N	checksum	Word	Checksum (optional see below)

ASIMP header is coded as follows:

0 … 15		16 … 31	
sessionId		comType	frameParam
address	frameType	Length	
Dataset			
checksum			

Detailed description of fields:

Field	Description	Value
sessionId	The session identifier • Shall be generated when creating a request to a server • Shall be returned with a reply	Computed Received
comType	Communication type specifies the direction and type. The value is one of the preset constants 0x00 = Addressed reply from server 0x01 = Broadcasted reply from server 0x10 = request requiring addressed reply 0x20 = request requiring a broadcasted reply 0xFF = unsolicited message without required reply For comType == 0xFF the sessionId field shall be set to 0	0x00 0x01 0x10 0x20 0xFF
frameParam	Further parameters of the used communication frame. bit mask: 0 – 0 – short form of ASIMP frame; 1 – extended ASIMP frame type 1 To fully utilize the IP protocol, it is recommended to set the value to 1 1 – Not used 2 – Not used 3 – Not used 4 – Not used	A bit mask; computed

<table>
<tr><td></td><td><table><tr><td>5</td><td>Not used</td></tr><tr><td>6</td><td>Not used</td></tr><tr><td>7</td><td>0 – with checksum; 1 – without checksum
Due to the verification aspects of IP protocol it is recommended to set the value to 1</td></tr></table></td><td></td></tr>
<tr><td>address</td><td colspan="2">Used for non-IP transmission protocols. Otherwise shall be set to 0</td><td>Fixed</td></tr>
<tr><td>frameType</td><td colspan="2">Type of used communication frame
00h – Non-valid frame
02h – NAK reply
88h – Host side ethernet configuration
89h – Reboot request
8fh – Device identification
0Ch – RM-TRDP call</td><td>computed</td></tr>
<tr><td>length</td><td><table><tr><td>Extended ASIMP frame type 1</td><td>Length of the dataset in bytes in little endian
1. octet has the least significant byte
34 12 -> 1234 octets</td></tr><tr><td>Short form of ASIMP frame</td><td>The length of the dataset, number of octets in dataset, twice.

Dataset has 120 octets -> 0x7878</td></tr></table></td><td></td><td>computed</td></tr>
<tr><td>dataset</td><td colspan="2">The user dataset. Further specification is based on the used frameType</td><td>variable</td></tr>
<tr><td>checksum</td><td colspan="2">Negatively taken checksum of frame starting at address (offset 4) and ending at the end of datapart. E.g. Sum of bytes from address to the end of data plus checksum should be zero. Checksum could be disabled by setting bit 7 of frameParam.</td><td>computed</td></tr>
</table>

4 Supported ASIMP/IP frames

4.1 Device identification (frameType 8fh)

Device identification request is usually sent as broadcast message requesting broadcast reply (comType = 0x20). This approach allows to identify devices which is configured with different subnet settings.

4.1.1 Request payload description

Request data has the following format:

Packet offset	Local offset	Field	Type	Description
+8	+0	request	Word	Request

Payload coded as follows:

0	15
request	

Detailed description of fields:

Field	Description	Value
request	For identification always set to 0	0x00

4.1.2 Reply payload description

Reply data has the following format:

Packet offset	Local offset	Field		Type	Description
+8	+0	OK/KO		Byte	0x00 – OK 0x80 – Data read error
+9	+1	Flags		Byte	Flags
+10	+2	MAC		Byte[6]	Mac address of host side ethernet
+16	+8	APLID	HWID	Word	AMiT internal hardware identification number
			PID	Word	Program identification number. (0 for generic application)
+20	+12	CFGID		Dword	Modification identification number (0 for generic application)
+24	+16	S/N		String[8]	Serial number
+32	+24	HW_STR		String[32]	Hardware name

+64	+56	CFG_STR	String[32]	Configuration name
+96	+88	USER_STR	String[32]	User supplied name
+128	+120	ACT_IP	Dword	Host side IP address
+132	+124	ACT_NMSK	Dword	Host side netmask
+136	+128	ACT_GTW	Dword	Host side default gateway
+140	+132	SERVICES	N x Dword	List of supported ASIMP services 0-30 elements. Elements contain Word of service identificatory and Word of service version

Payload coded as follows:

0 … 15		16 … 31
OK/OK	FLAGS	MAC[0-1]
MAC[2-5]		
HWID		PID
CFGID		
S/N[0-3]		
S/N[4-5]		
HW_STR 8x DWORD		
CFG _STR 8x DWORD		
USER _STR 8x DWORD		
ACT_IP		
ACT_NMSK		
ACT_GTW		
Service[0] code		Service[0] version
Service[n+1] code		Service[n+1] version

Detailed description of fields:

Field	Description	Value
OK/KO	Operation result code	0 success, 0x80 error
Flags	bit mask: 0 – 0 - dhcp client disabled, 1 - enabled. Always 0 for RM-TRDP 1 – 0 – application is running, 1 – running safe application (service mode) 2-7 – Not used	0x00 for application 0x02 for safe application
MAC	MAC address of host side ethernet interface	Factory data

HWID	Hardware identification number	Not used
PID	Project identification number	Factory data
CFGID	Configuration identification number	Factory data
S/N	Serial number of device	Factory data
HW_STR	Name of device	Factory data
CFG_STR	Application name and version	“TRDP X.Y”
USER_STR	User supplied string	Not used
ACT_IP	Actual IP address of host side ethernet	Default 172.0.0.1, otherwise configured by user
ACT_NMSK	Actual netmask of host side ethernet	Default 255.255.255.0, otherwise configured by user
ACT_GTW	Actual default gateway of host side ethernet	Default 172.0.0.2, otherwise configured by user
SERVICES	List of supported services if form of identifier (Word), version pair (Word)	0x0000,0x007E 0x000D,0x0067 ADD version 126 TRDP version 103

4.2 Host side ethernet setting (frameType 88h)

Device ethernet setting request is usually sent as broadcast message requesting broadcast reply (comType = 0x20). This approach allows to reconfigure devices which is configured with different subnet settings.

Ethernet settings is stored persistently.

4.2.1 Request payload description

Request data has the following format:

Packet offset	Local offset	Field	Type	Description
+8	+0	RESET	Byte	Set ethernet immediately, or after reboot
+9	+1	FLAGS	Word	Bitmask which configuration set
+11	+3	MAC	Byte[6]	Mac address of configured device
+17	+9	NIP	Byte[4]	New IP address, MSB first
+17	+9	NMKS	Byte[4]	New netmask, MSB first
+17	+9	NGTW	Byte[4]	New default gateway, MSB first

Payload coded as follows:

0	15 | 16	31
RESET	FLAGS	MAC[0]
MAC[1-4]		

MAC[5]	NIP[0-2]
NIP [3]	NMKS [0-2]
NMKS[3]	NGTW[0-2]
NGTW[0-3]	

Detailed description of fields:

Field	Description	Value
RESET	Set network immediately 0xff, or after reboot 0x00	Supplied by user, immediate set recommended (0xff)
FLAGS	0 – 0 – read DHCP config, 1 – set DHCP config (not supported) 1 – Reserved, must be 0 2 – 0 – read IP, 1 – set IP 3 – 0 – read mask, 1 – set mask 4 – 0 – read getaway, 1 – set getaway 5 – 0 – DHCP disable (not supported) 1 – DHCP enable (not supported) 6-15 – Not used	Supplied by user
NIP	New host side IP address	Supplied by user
NMKS	New host side IP network mask	Supplied by user
NGTW	New host side IP default gateway	Supplied by user

4.2.2 Reply payload description

Reply data has the following format:

Packet offset	Local offset	Field	Type	Description
+8	+0	OK/KO	Byte	0x00 – OK 0x02 – Setting error
+9	+1	Flags	Byte	Flags describing error
+11	+3	MAC	Byte[6]	Mac address of configured device
+17	+9	IP	Byte[4]	Actual IP address, MSB first
+17	+9	MKS	Byte[4]	Actual netmask, MSB first
+17	+9	GTW	Byte[4]	Actual default gateway, MSB first

Payload coded as follows:

0	15	16	31
RESET	FLAGS		MAC[0]
MAC[1-4]			
MAC[5]	NIP[0-2]		
NIP [3]	NMKS [0-2]		
NMKS[3]	NGTW[0-2]		
NGTW[0-3]			

Detailed description of fields:

Field	Description		Value
OK/OK	Operation result code		0x00 on success, 0x02 on error, detail encoded in flags
FLAGS	0	0 – set DHCP OK (not supported) 1 – set DHCP failure	Result of operation and flags
	1	Reserved, must be 0	
	2	0 – set IP OK, 1 – set IP failure	
	3	0 – read mask OK, 1 – set mask failure	
	4	0 – read getaway OK, 1 – set getaway failure	
	5	0 – DHCP is disabled 1 – DHCP id enabled (not supported)	
	6	0 – Device doesn't have ethernet 1 - Device has ethernet (always 1)	
	7	0 – Device doesn't support DHCP 1 - Device supports DHCP (always 0)	
	8-15	Not used	
NIP	Actual host side IP address		Runtime value
NMKS	Actual host side IP network mask		Runtime value
NGTW	Actual host side IP default gateway		Runtime value

4.3 Reboot request (frameType 89h)

Device reboot request is usually sent as unicast message requesting unicast reply (comType = 0x10).

Ethernet settings is stored persistently.

4.3.1 Request payload description

Request data has the following format:

Packet offset	Local offset	Field	Type	Description

+8	+0	REQ	Byte	0x1 – request reboot
+9	+1	TEST1	Byte	Fixed value 0x12
+10	+2	TEST2	Byte	Fixed value 0x34
+11	+3	TEST3	Byte	Fixed value 0x56

Payload coded as follows:

0 … 15		16 … 31	
REQ	TEST1	TEST2	TEST3

Detailed description of fields:

Field	Description	Value
REQ	Always 0x1	Fixed
TEST1	Always 0x12	Fixed
TEST2	Always 0x34	Fixed
TEST3	Always 0x56	Fixed

4.3.2 Reply payload description

Reply data has the following format:

Packet offset	Local offset	Field	Type	Description
+8	+0	OK/KO	Byte	0x00 – request accepted
+9	+1	REQ	Byte	Copy of REQ from request
+10	+2	TEST1	Byte	Fixed value 0x78
+11	+3	TEST2	Byte	Fixed value 0x90

Payload coded as follows:

0 … 15		16 … 31	
OK/KO	REQ	TEST1	TEST2

Detailed description of fields:

Field	Description	Value
OK/KO	Always 0x00	Fixed
REQ	Always 0x01	Fixed
TEST1	Always 0x78	Fixed
TEST2	Always 0x90	Fixed

4.4 ASIMP-TRDP frames (frameType 0ch)

TRDP stack is controlled via unicast ASIMP messages with frame type 0x0c and additional ASIMP-TRDP header and TRDP function specific payloads. Unsolicited messages (indication) from TRDP stack are send to host side default gateway, port 75.

4.4.1 ASIMP-TRDP header

ASIMP-TRDP header has the following format:

Packet offset	Local offset	Field	Type	Description
+8	+0	identification	String[4]	ASIMP-TRDP protocol identification
+12	+4	funcId	Byte	function identifier
+13	+5	frameVersion	Byte	reserved for future use
+14	+6	protocolVersion	Word	used version of the protocol
+16	+8	resultCode	DWord	result code of called function
+20	+12	dataset	Byte[N]	data part of the frame

ASIMP-TRDP header is coded as follows:

0 … 15		16 … 31
funcId	frameVersion	protocolVersion
dataset		

Detailed description of fields:

Parameter	Description	Value
identification	ASIMP-TRDP protocol identification, ASCII encoded	‘astr’
funcId	Identifies the function of TRDP stack to call.	Function identifier
frameVersion	Used version of the frame specification	set by user
protocolVersion	Used version of the protocol, encoded in little endian First the lower significant octet i.e., the minor version Second the higher significant octet i.e. the major version EXAMPLE 0x0301 – protocol version 1.3	Fixed
resultCode	The result code of TRDP module service. Shall be set to 0 in request frame from host	Result code
dataset	The user dataset. Further specification is based on the used function	variable

4.5 Function identifiers

Function identifiers are used to distinguish TRDP service.

funcId	Function	Direction
0x00	PD.publish	Host -> TRDP module
0x01	PD.unPublish	Host -> TRDP module
0x02	PD.putData	Host -> TRDP module
0x03	PD.activateRed	Host -> TRDP module
0x04	PD.deactivateRed	Host -> TRDP module
0x05	PD.request	Host -> TRDP module
0x06	PD.subscribe	Host -> TRDP module
0x07	PD.unsubscribe	Host -> TRDP module
0x08	PD.indicate	TRDP module -> Host
0x09	- function has been removed	-
0x80	MD.request	Host -> TRDP module
0x81	MD.indicate	TRDP module -> Host
0x82	MD.confirm	Host -> TRDP module
0x83	MD.abort	Host -> TRDP module
0x84	MD.addListener	Host -> TRDP module
0x85	- function has been removed	-
0x86	MD.remListener	Host -> TRDP module
0x87	- function has been removed	-
0x88	MD.reply	Host -> TRDP module
0x89	MD.release	Host -> TRDP module
0xff	TRDP Start	Host -> TRDP module
0xfe	TRDP Restart	Host -> TRDP module
0xfd	TRDP ChangeAddress	Host -> TRDP module
0xfc	TRDP Stop	Host -> TRDP module

4.6 Result codes

Code	Name	Description
0x00	trdp_ok	Success
0x01	trdp_fail	Unspecified failure
0x02	trdp_srv_not_reg	Service not registered
0x03	trdp_srv_alr_reg	Service already registered
0x04	trdp_fat_err	Fatal internal error
0x05	trdp_srv_not_exe	Service not executed
0x06	trdp_com_err	Communication error
0x07	trdp_off	TRDP session is not running

4.7 ASIMP-TRDP module management functions

Each TRDP function has its specific payload. Most of functions are prepared according IEC 61375-2-3 ed1.0:2005, but few more functions for device configuration and initialization were added. Description follows.

4.7.1 TRDP Start function

Request Host → TRDP module

Initializes TRDP stack and configures TRDP side IP settings.

TRDP Start function has the following format:

Packet offset	Local offset	Field	Type	Description
+20	+0	Address	DWord	The new address of the module on the TRDP layer
+24	+4	Netmask	DWord	The new netmask of the module on the TRDP layer
+28	+8	Gateway	DWord	The new default gateway of the module on the TRDP layer

TRDP Start function is coded as follows:

0 … 15	16 … 31
Address	
Netmask	
Gateway	

Detailed description of fields:

Parameter	Description	Value
Address	IP Address of the device in the TRDP network encoded as little endian	set by user
Netmask	Network mask of the device in the TRDP network encoded as little endian	set by user
Gateway	Default gateway of the device in the TRDP network encoded as little endian	set by user

Response TRDP module → Host
No parameters

4.7.2 TRDP Restart function

Request Host → TRDP module

No parameters

Response TRDP module → Host
No parameters

Reinitializes TRDP stack.

4.7.3 TRDP ChangeAddress function

Request Host → TRDP module

Reinitializes TRDP stack and also reconfigures TRDP side IP settings.

TRDP ChangeAddress function has the following format:

Packet offset	Local offset	Field	Type	Description
+20	+0	Address	DWord	The new address of the module on the TRDP layer
+24	+4	Netmask	DWord	The new netmask of the module on the TRDP layer
+28	+8	Gateway	DWord	The new default gateway of the module on the TRDP layer

TRDP ChangeAddress function is coded as follows:

0 … 15	16 … 31
Address	
Netmask	
Gateway	

Detailed description of fields:

Parameter	Description	Value
Address	IP Address of the device in the TRDP network encoded as little endian	set by user
Netmask	Network mask of the device in the TRDP network encoded as little endian	set by user
Gateway	Default gateway of the device in the TRDP network encoded as little endian	set by user

Response TRDP module → Host
No parameters

4.7.4 TRDP Stop function

Request Host → TRDP module

No parameters

Response TRDP module → Host
No parameters

Stops and cleanups TRDP stack. Subsequent call to Start function is possible.

4.8 ASIMP-TRDP functions according IEC 61375-2-3 ed1.0:2005

4.8.1 TRDP PD.publish function

Request Host → TRDP module

TRDP PD.publish function has the following format:

Packet offset	Local offset	Field	Type	Description
+20	+0	etbTopoCnt	DWord	etbTopoCnt value
+24	+4	opTrnTopoCnt	DWord	opTrnTopoCnt value
+28	+8	comId	DWord	communication identifier
+32	+12	datasetLength (M)	DWord	Length of the user dataset
+36	+16	cycleTime	DWord	cycle time for sending PD
+40	+20	redundancy	Byte	control whether the PD is redundant
+41	+21	reserved01	Byte	reserved for future use
+42	+22	reserved02	Byte	reserved for future use
+43	+23	reserved03	Byte	reserved for future use
+44	+24	destinationIpAddress	DWord	The destination Ip address
+48	+28	dataset	Byte[M]	the user dataset

TRDP PD.publish function is coded as follows:

0 … 15		16 … 31	
etbTopoCnt			
opTrnTopoCnt			
comId			
datasetLength			
cycleTime			
redundancy	reserved01	reserved02	reserved03
destinationIpAddress			
dataset			

Detailed description of fields:

Parameter	Description	Value
etbTopoCnt	The ETB topography counter: • shall be used (train addressing) as defined in IEC 61375-2-5 (parameter 'etbTopoCnt') • shall be set by the user • shall be set for all communication over the ETB	$0..2^{32-1}$ set by user

	• shall be set if a valid opTrnTopoCnt is set • optional in all other cases. Shall be set to 0 (= invalid) if not used.	
opTrnTopoCnt	The operational train topography counter: • shall be used as defined in 5.3.3 • shall be set by the user • shall be set for communication between functions which use information from the operational train directory (e.g. side selective operations) • shall be set when the source device used the operational train directory to retrieve the destination IP address (e.g. resolving of URI 'vcu.leadVeh.anyCst.anyClTrn.lTrn') • optional in all other cases. Shall be set to 0 (= invalid) if not used	
comId	Identifier of the user dataset.	set by user
datasetLength	The dataset length: • shall be the length of the user data set in number of octets without padding octets. • shall be the primary information about the user data size NOTE In case of fixed size data sets this is redundant information because the dataset length is already defined by the ComId.	Computed
cycleTime	Cycle time (txTime) for sending PD. Unit: µs = 10^{-6} s	set by user
redundancy	PD set as not redundant are published immediately. PD set as redundant are published if redundant ComIds are activated (function activateRed)	0x00 Not redundant 0x01 Redundant
reserved01	Reserved for future use. • Shall be set to 0	Fixed
reserved02	Reserved for future use. • Shall be set to 0	Fixed
reserved03	Reserved for future use. • Shall be set to 0	Fixed
destinationIpAddress	A destination IP address encoded in little endian. Address of subscriber, or multicast group. EXAMPLE 192.168.0.1 0x0100A8C0	set by user
dataset	The user dataset	

Response TRDP module → Host

TRDP PD.publish function has the following format:

Packet offset	Local offset	Field	Type	Description
+20	+0	Handle	DWord	handle for the publishing

TRDP PD.publish function is coded as follows:

0 … 15	16 … 31
Handle	

Detailed description of fields:

Parameter	Description	Value
Handle	Handle returned by the TRDP layer.	set by user

4.8.2 TRDP PD.unPublish function

Request Host → TRDP module

TRDP PD.unPublish function has the following format:

Packet offset	Local offset	Field	Type	Description
+20	+0	Handle	DWord	handle for the publishing

TRDP PD.unPublish function is coded as follows:

0 … 15	16 … 31
Handle	

Detailed description of fields:

Parameter	Description	Value
Handle	Handle returned by the TRDP layer.	set by user

Response TRDP module → Host
No parameters

4.8.3 TRDP PD.putData function

Request Host → TRDP module

TRDP PD.putData function has the following format

Packet offset	Local offset	Field	Type	Description
+20	+0	Handle	DWord	handle for the publishing
+24	+4	datasetLength(N)	DWord	Length of the user dataset
+28	+8	Dataset	Byte[N]	The user dataset

Response TRDP module → Host
No parameters

4.8.4 TRDP PD.activateRed function

Request Host → TRDP module

No parameters

Response TRDP module → Host
No parameters

4.8.5 TRDP PD.deactivateRed function

Request Host → TRDP module

No parameters

Response TRDP module → Host
No parameters

4.8.6 TRDP PD.request function

Request Host → TRDP module

TRDP PD.request function has the following format:

Packet offset	Local offset	Field	Type	Description
+20	+0	subHandle	DWord	Subscription handle
+24	+4	etbTopoCnt	DWord	etbTopoCnt value
+28	+8	opTrnTopoCnt	DWord	opTrnTopoCnt value
+32	+12	comId	DWord	communication identifier
+36	+16	datasetLength (M)	DWord	Length of the user dataset
+40	+20	cycleTime	DWord	cycle time for sending PD
+44	+24	redundancy	Byte	control whether the PD is redundant
+45	+25	reserved01	Byte	reserved for future use
+46	+26	reserved02	Byte	reserved for future use
+47	+27	reserved03	Byte	reserved for future use
+48	+28	destinationIpAddress	DWord	The destination Ip

				address
+52	+32	replyComId	DWord	communication identifier of the reply
+56	+36	replyIpAddress	DWord	the destination IP of the reply
+60	+40	dataset	Byte[M]	the user dataset

TRDP PD.request function is coded as follows:

0 … 15		16 … 31	
subHandle			
etbTopoCnt			
opTrnTopoCnt			
comId			
datasetLength			
cycleTime			
redundancy	reserved01	reserved02	reserved03
destinationIpAddress			
replyComId			
replyIpAddress			
dataset			

Detailed description of fields:

Parameter	Description	Value
subHandle	Handle from corresponding subscription	set by user
etbTopoCnt	The ETB topography counter: • shall be used (train addressing) as defined in IEC 61375-2-5 (parameter 'etbTopoCnt') • shall be set by the user • shall be set for all communication over the ETB • shall be set if a valid opTrnTopoCnt is set • optional in all other cases. Shall be set to 0 (= invalid) if not used.	$0..2^{32-1}$ set by user
opTrnTopoCnt	The operational train topography counter: • shall be used as defined in 5.3.3 • shall be set by the user • shall be set for communication between functions which use information from the operational train directory (e.g. side selective operations) • shall be set when the source device	

	used the operational train directory to retrieve the destination IP address (e.g. resolving of URI 'vcu.leadVeh.anyCst.anyClTrn.lTrn') • optional in all other cases. Shall be set to 0 (= invalid) if not used	
comId	Identifier of the user dataset.	set by user
datasetLength	The dataset length: • shall be the length of the user data set in number of octets without padding octets. • shall be the primary information about the user data size NOTE In case of fixed size data sets this is redundant information because the dataset length is already defined by the ComId.	Computed
cycleTime	Cycle time (txTime) for sending PD. Unit: µs = 10^{-6} s	set by user
redundancy	PD set as not redundant are published immediately. PD set as redundant are published if redundant ComIds are activated (function activateRed)	0x00 Not redundant 0x01 Redundant
reserved01	Reserved for future use. • Shall be set to 0	Fixed
reserved02	Reserved for future use. • Shall be set to 0	Fixed
reserved03	Reserved for future use. • Shall be set to 0	Fixed
destinationIpAddress	A destination IP address encoded in little endian. EXAMPLE 192.168.0.1 0x0100A8C0	set by user
replyComId	The requested ComId: • shall be used only in a PD request • shall be used as ComId in the reply • If set to 0, the ComId of the request shall be used for the reply • If set to 0, and the ComId of the request is also set to 0, the reply shall be sent as an unspecified PDU.	set by user
replyIpAddress	A destination IP address of the response. • if set to 0 the source IP address shall be used	set by user
dataset	The user dataset	

Response TRDP module → Host

No parameters

4.8.7 TRDP PD.subscribe function

Request Host → TRDP module

TRDP PD.subscribe function has the following format

Packet offset	Local offset	Field	Type	Description
+20	+0	etbTopoCnt	DWord	etbTopoCnt value
+24	+4	opTrnTopoCnt	DWord	opTrnTopoCnt value
+28	+8	comId	DWord	communication identifier
+32	+12	datasetLength (N)	DWord	Length of the user dataset
+36	+16	cycleTime	DWord	cycle time for sending PD
+40	+20	redundancy	Byte	control whether the PD is redundant
+41	+21	reserved01	Byte	reserved for future use
+42	+22	reserved02	Byte	reserved for future use
+43	+23	reserved03	Byte	reserved for future use
+44	+24	sourceIpAddress	DWord	Source IP address or lower IP address in case of range
+48	+28	destinationIpAddress	DWord	Destination IP address
+52	+42	timeout	DWord	timeout for the PD to be received
+56	+46	dataset	Byte[N]	the user dataset

TRDP PD.subscribe function is coded as follows:

0 … 15		16 … 31	
etbTopoCnt			
opTrnTopoCnt			
comId			
datasetLength			
cycleTime			
redundancy	reserved01	reserved02	reserved03
sourceIpAddress			
destinationIpAddress			
timeout			
dataset			

Detailed description of fields:

Parameter	Description	Value
etbTopoCnt	The ETB topography counter: • shall be used (train addressing) as defined in IEC 61375-2-5 (parameter 'etbTopoCnt') • shall be set by the user • shall be set for all communication	$0..2^{32-1}$ set by user

	over the ETB • shall be set if a valid opTrnTopoCnt is set • optional in all other cases. Shall be set to 0 (= invalid) if not used.	
opTrnTopoCnt	The operational train topography counter: • shall be used as defined in 5.3.3 • shall be set by the user • shall be set for communication between functions which use information from the operational train directory (e.g. side selective operations) • shall be set when the source device used the operational train directory to retrieve the destination IP address (e.g. resolving of URI 'vcu.leadVeh.anyCst.anyClTrn.lTrn') • optional in all other cases. Shall be set to 0 (= invalid) if not used	
comId	Identifier of the user dataset.	set by user
datasetLength	The dataset length: • shall be the length of the user data set in number of octets without padding octets. • shall be the primary information about the user data size NOTE In case of fixed size data sets this is redundant information because the dataset length is already defined by the ComId.	Computed
cycleTime	Cycle time (txTime) for sending PD. Unit: µs = 10^{-6} s	set by user
redundancy	PD set as not redundant are published immediately. PD set as redundant are published if redundant ComIds are activated (function activateRed)	0x00 Not redundant 0x01 Redundant
reserved01	Reserved for future use. • Shall be set to 0	Fixed
reserved02	Reserved for future use. • Shall be set to 0	Fixed
reserved03	Reserved for future use. • Shall be set to 0	Fixed
sourceIpAddress	IP source address (address of publisher, which will send the date), generated out of respective URI's using DNS. Defines the lower IP address in case of an IP address range	set by user

destinationIpAddress	Defines the destination IP address (address of publisher which is being asked for data), generated out of respective URI's using DNS.	set by user
timeout	Timeout time (rxTime) for receiving PD. Unit: µs = 10^{-6} s	set by user
dataset	The user dataset	

Response TRDP module → Host

TRDP PD.subscribe function has the following format:

Packet offset	Local offset	Field	Type	Description
+20	+0	Handle	DWord	handle for the subscription

TRDP PD.subscription function is coded as follows:

0 … 15	16 … 31
Handle	

Detailed description of fields:

Parameter	Description	Value
Handle	Handle returned by the TRDP layer.	set by user

4.8.8 TRDP PD.unsubscribe function

Request Host → TRDP module

TRDP PD.unsubscribe function has the following format:

Packet offset	Local offset	Field	Type	Description
+20	+0	Handle	DWord	handle for the subscription

TRDP PD.unsubscribe function is coded as follows:

0 … 15	16 … 31
Handle	

Detailed description of fields:

Parameter	Description	Value
Handle	Handle returned by the TRDP layer.	set by user

Response TRDP module → Host
No parameters

4.8.9 TRDP PD.indicate function

Request TRDP module → Host

TRDP PD.indicate function has the following format:

Packet offset	Local offset	Field	Type	Description
+20	+0	Handle	DWord	handle for the subscription
+24	+4	indType	String[2]	type of indication
+26	+6	status	Byte	status of the TRDP layer
+27	+7	reserved01	Byte	reserved for future use
+28	+8	etbTopoCnt	DWord	etbTopoCnt value
+32	+12	opTrnTopoCnt	DWord	opTrnTopoCnt value
+36	+16	comId	DWord	communication identifier
+40	+20	sequenceCounter	DWord	message sequence counter
+44	+24	sourceIpAddress	DWord	Source IP address
+48	+28	destinationIpAddress	DWord	Destination IP address
+52	+32	replyComId	DWord	communication identifier of the reply
+56	+36	replyIpAddress	DWord	the destination IP of the reply
+60	+40	datasetLength	DWord	Length of the user dataset
+64	+44	dataset	Byte[N]	the user dataset

TRDP PD.indicate function is coded as follows:

0 … 15	16 … 31	
Handle		
indType	status	reserved01
etbTopoCnt		
opTrnTopoCnt		
comId		
sequenceCounter		
sourceIpAddress		
destinationIpAddress		
replyComId		
replyIpAddress		
datasetLength		
dataset		

Detailed description of fields:

Parameter	Description	Value
Handle	Handle returned by the TRDP layer.	set by user
indType	The type of indication.	• process data ('Pd') • process data reply ('Pp') • process data request ('Pr') • process data error ('Pe')
status	Provides additional information if the indType reports an error. • Shall be set to 0, if no error is present 0x01 timeout 0x06 no subscription 0x05 no memory 0x07 data invalid	0 OK <0 not OK
reserved01	Reserved for future use. • Shall be set to 0	Fixed
etbTopoCnt	The ETB topography counter: • shall be used (train addressing) as defined in IEC 61375-2-5 (parameter 'etbTopoCnt') • shall be set by the user • shall be set for all communication over the ETB • shall be set if a valid opTrnTopoCnt is set • optional in all other cases. Shall be set to 0 (= invalid) if not used.	$0..2^{32-1}$ set by user
opTrnTopoCnt	The operational train topography counter: • shall be used as defined in 5.3.3 • shall be set by the user • shall be set for communication between functions which use information from the operational train directory (e.g. side selective operations) • shall be set when the source device used the operational train directory to retrieve the destination IP address (e.g. resolving of URI 'vcu.leadVeh.anyCst.anyClTrn.lTrn') • optional in all other cases. Shall be set to 0 (= invalid) if not used	
comId	Identifier of the user dataset.	set by user

sequenceCounter	A sequence counter incremented on the TRDP layer • Shall be a copied if a poll function is called as a response.	set by user
sourceIpAddress	IP address of the source of the PD, encoded in little endian EXAMPLE 192.168.0.1 - 0x0100A8C0	set by user
destinationIpAddress	Destination IP address of the PD encoded in little endian EXAMPLE 192.168.0.1 - 0x0100A8C0	set by user
replyComId	The requested ComId: • shall be used only in a PD request • shall be used as ComId in the reply • If set to 0, the ComId of the request shall be used for the reply • If set to 0, and the ComId of the request is also set to 0, the reply shall be sent as an unspecified PDU.	set by user
replyIpAddress	A destination IP address of the response. • if set to 0 the source IP address shall be used	set by user
datasetLength	The dataset length: • shall be the length of the user data set in number of octets without padding octets. • shall be the primary information about the user data size NOTE In case of fixed size data sets this is redundant information because the dataset length is already defined by the ComId.	Computed
dataset	The user dataset	

Response Not expected

4.8.10 TRDP MD.request function

Request Host → TRDP module

TRDP MD.request function has the following format:

Packet offset	Local offset	Field	Type	Description
+20	+0	msgType	String[2]	type of message
+22	+2	transProtocol	Byte	type of transport protocol
+23	+3	numRepliers	Byte	amount of repliers
+24	+4	userIdentifier	DWord[2]	user identification

+32	+12	etbTopoCnt	DWord	etbTopoCnt value
+36	+16	opTrnTopoCnt	DWord	opTrnTopoCnt value
+40	+20	comId	DWord	communication identifier
+44	+24	sourceUri	String[32]	source URI (user part)
+76	+56	destinationUri	String[32]	destination URI (user part)
+108	+88	destinationIpAddress	DWord	Destination IP address
+112	+92	replyTimeOut	DWord	reply timeout
+116	+96	maxNumRetries	Byte	Maximum number of retries in case num Repliers == 1
+117	+97	reserved01	Byte	reserved for future use
+118	+98	reserved02	Byte	reserved for future use
+119	+99	reserved03	Byte	reserved for future use
+120	+100	datasetLength	DWord	Length of the user dataset
+124	+124	dataset	Byte[N]	the user dataset

TRDP MD.request function is coded as follows:

0 … 15	16 … 31	
msgType	transProtocol	numRepliers
userIdentifier		
etbTopoCnt		
opTrnTopoCnt		
comId		
sourceUri		
destinationUri		
destinationIpAddress		
replyTimeOut		
maxNumRetries	reserved01	reserved02	reserved03
datasetLength		
dataset		

Detailed description of fields:

Parameter	Description	Value
msgType	Type of the telegram coded in ASCII ‘Mn’ = Notification (Request without reply) ‘Mr’ = MD Request with reply ‘Mp’ = MD Reply without confirmation ‘Mq’ = MD Reply with confirmation ‘Mc’ = MD Confirm ‘Me’= MD error	‘4D6E‘H (‘Mn’) ‘4D72‘H (‘Mr’) ‘4D70‘H (‘Mp’) ‘4D71‘H (‘Mq’) ‘4D63‘H (‘Mc’) ‘4D65‘H (‘Me’)
transProtocol	The transport protocol to be used for the MD 0x00 UDP 0x01 TCP	set by user
numRepliers	Amount of expected replies to the MD. • If no reply is expected shall be set to 0, e.g. a notification • If an unknown number of replies is expected shall be set to 0	0..255 set by user
userIdentifier	An identifier that is further returned by the indicate function. It is to associate the indication with the request function call	set by user
etbTopoCnt	The ETB topography counter: • shall be used (train addressing) as defined in IEC 61375-2-5 (parameter ‘etbTopoCnt’) • shall be set by the user • shall be set for all communication over the ETB • shall be set if a valid opTrnTopoCnt is set • optional in all other cases. Shall be set to 0 (= invalid) if not used.	$0..2^{32-1}$ set by user
opTrnTopoCnt	The operational train topography counter: • shall be used as defined in 5.3.3 • shall be set by the user • shall be set for communication between functions which use information from the operational train directory (e.g. side selective operations) • shall be set when the source device used the operational train directory to retrieve the destination IP address (e.g. resolving of URI ‘vcu.leadVeh.anyCst.anyClTrn.lTrn	

	’) • optional in all other cases. Shall be set to 0 (= invalid) if not used	
comId	Identifier of the user dataset.	set by user
sourceUri	The URI of the source • shall be the used for functional addressing • shall be a null terminated string • filling bytes at the end shall be set to 0 • shall contain only the “user part” without “host part” and “@” • may be an empty string (all 0)	set by user
destinationUri	The URI of the destination • shall be the used for functional addressing • shall be a null terminated string • filling bytes at the end shall be set to 0 • shall contain only the “user part” without “host part” and “@” • may be an empty string (all 0)	set by user
destinationIpAddress	Destination IP address of the PD encoded in little endian EXAMPLE 192.168.0.1 - 0x0100A8C0	set by user
replyTimeOut	The reply timeout value shall be set to define the expected reply time in a request / reply session. • Shall be set to 0 for msgTypes without a reply • If infinite type is required shall be set to 0 Unit: µs = 10^{-6} s	0 1..(2^{32}-1)
maxNumRetries	The maximum amount of retries if numRepliers == 1. • if numRepliers != 1 shall be set to 0	0..2 set by user
reserved01	Reserved for future use. • Shall be set to 0	Fixed
reserved02	Reserved for future use. • Shall be set to 0	Fixed
reserved03	Reserved for future use. • Shall be set to 0	Fixed
datasetLength	The dataset length: • shall be the length of the user data set in number of octets without padding octets. • shall be the primary information	Computed

	about the user data size NOTE In case of fixed size data sets this is redundant information because the dataset length is already defined by the ComId.	
dataset	The user dataset	

Response TRDP module → Host

TRDP MD.request function has the following format:

Packet offset	Local offset	Field	Type	Description
+20	+0	sessionId	DWord[4]	session identifier returned by the TRDP layer

TRDP MD.request function is coded as follows:

0 … 15	16 … 31
sessionId	

Detailed description of fields:

Parameter	Description	Value
sessionId	A session identifier generated by the TRDP layer	set by user

4.8.11 TRDP MD.indicate function

Request TRDP module → Host

TRDP MD.indicate function has the following format:

Packet offset	Local offset	Field	Type	Description
+20	+0	indType	String[2]	type of indication
+22	+2	NoOfRepl	Byte	number of received replies
+23	+3	reserved01	Byte	reserved for future use
+24	+4	userIdentifier	DWord[2]	user identification
+32	+12	sessionId	DWord[4]	session identifier
+48	+28	etbTopoCnt	DWord	etbTopoCnt value
+52	+32	opTrnTopoCnt	DWord	opTrnTopoCnt value
+56	+36	comId	DWord	communication identifier
+60	+40	sourceUri	String[32]	source URI (user part)
+92	+72	destinationUri	String[32]	destination URI (user part)
+124	+104	sourceIpAddress	DWord	source IP address
+128	+108	destinationIpAddress	DWord	Destination IP address

+132	+112	replyTimeOut	DWord	reply timeout
+136	+116	replyStatus	DWord	reply status
+140	+120	datasetLength	DWord	Length of the user dataset
+144	+124	dataset	Byte[N]	the user dataset

TRDP MD.indicate function is coded as follows:

0 … 15	16 … 31	
indType	NoOfRepl	reserved01
userIdentifier		
sessionId		
etbTopoCnt		
opTrnTopoCnt		
comId		
sourceUri		
destinationUri		
sourceIpAddress		
destinationIpAddress		
replyTimeOut		
replyStatus		
datasetLength		
dataset		

Detailed description of fields:

Parameter	Description	Value
indType	Type of indication	• reception of a reply message (‘Mp’) • reception of a

		query message ('Mq') • TRDP layer error ('Me')
NoOfRepl	Amount of received replies	set by user
reserved01	Reserved for future use. • Shall be set to 0	Fixed
userIdentifier	An identifier that is further returned by the indicate function. It is to associate the indication with the request function call	set by user
sessionId	A session identifier generated by the TRDP layer	set by user
etbTopoCnt	The ETB topography counter: • shall be used (train addressing) as defined in IEC 61375-2-5 (parameter 'etbTopoCnt') • shall be set by the user • shall be set for all communication over the ETB • shall be set if a valid opTrnTopoCnt is set • optional in all other cases. Shall be set to 0 (= invalid) if not used.	$0..2^{32-1}$ set by user
opTrnTopoCnt	The operational train topography counter: • shall be used as defined in 5.3.3 • shall be set by the user • shall be set for communication between functions which use information from the operational train directory (e.g. side selective operations) • shall be set when the source device used the operational train directory to retrieve the destination IP address (e.g. resolving of URI 'vcu.leadVeh.anyCst.anyClTrn.lTrn') • optional in all other cases. Shall be set to 0 (= invalid) if not used	
comId	Identifier of the user dataset.	set by user
sourceUri	The URI of the source • shall be the used for functional addressing • shall be a null terminated string • filling bytes at the end shall be set to 0 • shall contain only the "user part" without "host part" and "@" • may be an empty string (all 0)	set by user
destinationUri	The URI of the destination	set by user

	• shall be the used for functional addressing • shall be a null terminated string • filling bytes at the end shall be set to 0 • shall contain only the “user part” without “host part” and “@” • may be an empty string (all 0)	
sourceIpAddress	IP address of the source of the PD, encoded in little endian EXAMPLE 192.168.0.1 - 0x0100A8C0	set by user
destinationIpAddress	Destination IP address of the PD encoded in little endian EXAMPLE 192.168.0.1 - 0x0100A8C0	set by user
replyTimeOut	The reply timeout value shall be set to define the expected reply time in a request / reply session. • Shall be set to 0 for msgTypes without a reply • If infinite type is required shall be set to 0 Unit: µs = 10^{-6} s	0 1..(2^{32}-1)
replyStatus	The status value shall be set by the replier to report the execution result of a request message or by the caller sending a confirmation. The execution result is supplied by the replying application and transmitted to the requesting application in addition to the reply message itself. In case of a TRDP error reply (‘Me’) the value is supplied by TRDP. –1 reserved –2 session abort –3 no replier instance (at replier side) –4 no memory (at replier side) –5 no memory (local) –6 no reply –7 not all replies –8 no confirm –9 reserved –10 sending failed	<0 Not OK 0 OK >0 user reply status
datasetLength	The dataset length: • shall be the length of the user data set in number of octets without padding octets. • shall be the primary information	Computed

	about the user data size NOTE In case of fixed size data sets this is redundant information because the dataset length is already defined by the ComId.	
dataset	The user dataset	

Response Not expected

4.8.12 TRDP MD.confirm function

Request Host → TRDP module

TRDP MD.confirm function has the following format:

Packet offset	Local offset	Field	Type	Description
+20	+0	sessionId	DWord[4]	session identifier
+36	+16	replyStatus	Dword	reply status

TRDP MD.confirm function is coded as follows:

0 … 15	16 … 31
sessionId	
replyStatus	

Detailed description of fields:

Parameter	Description	Value
sessionId	A session identifier generated by the TRDP layer	set by user
replyStatus	The status value shall be set by the replier to report the execution result of a request message or by the caller sending a confirmation. The execution result is supplied by the replying application and transmitted to the requesting application in addition to the reply message itself. In case of a TRDP error reply ('Me') the value is supplied by TRDP. –1 reserved –2 session abort –3 no replier instance (at replier side) –4 no memory (at replier side) –5 no memory (local) –6 no reply	<0 Not OK 0 OK >0 user reply status

	–7 not all replies –8 no confirm –9 reserved –10 sending failed	

Response TRDP module → Host
No parameters

4.8.13 TRDP MD.abort function

Request Host → TRDP module

TRDP MD.abort function has the following format:

Packet offset	Local offset	Field	Type	Description
+20	+0	sessionId	DWord[4]	session identifier returned by the TRDP layer

TRDP MD.abort function is coded as follows:

0 … 15	16 … 31
sessionId	

Detailed description of fields:

Parameter	Description	Value
sessionId	A session identifier generated by the TRDP layer	set by user

Response TRDP module → Host
No parameters

4.8.14 TRDP MD.addListener function

Request Host → TRDP module

TRDP MD.addListener function has the following format:

Packet offset	Local offset	Field	Type	Description
+20	+0	comId	DWord	communication identifier
+24	+4	etbTopoCnt	DWord	etbTopoCnt value
+28	+8	opTrnTopoCnt	DWord	opTrnTopoCnt value
+32	+12	destinationUri	String[32]	destination URI (user part)

+64	+44	destinationIpAddress	DWord	Destination IP address
+68	+48	userIdentifier	DWord[2]	user identification

TRDP MD.addListener function is coded as follows:

0 … 15	16 … 31
comId	
etbTopoCnt	
opTrnTopoCnt	
destinationUri	
destinationIpAddress	
userIdentifier	

Detailed description of fields:

Parameter	Description	Value
comId	Identifier of the user dataset.	set by user
etbTopoCnt	The ETB topography counter: • shall be used (train addressing) as defined in IEC 61375-2-5 (parameter 'etbTopoCnt') • shall be set by the user • shall be set for all communication over the ETB • shall be set if a valid opTrnTopoCnt is set • optional in all other cases. Shall be set to 0 (= invalid) if not used.	$0..2^{32-1}$ set by user
opTrnTopoCnt	The operational train topography counter: • shall be used as defined in 5.3.3 • shall be set by the user • shall be set for communication between functions which use information from the operational train directory (e.g. side selective operations) • shall be set when the source device used the operational train directory to retrieve the destination IP address	

	(e.g. resolving of URI 'vcu.leadVeh.anyCst.anyClTrn.lTrn') • optional in all other cases. Shall be set to 0 (= invalid) if not used	
destinationUri	The URI of the destination • shall be the used for functional addressing • shall be a null terminated string • filling bytes at the end shall be set to 0 • shall contain only the "user part" without "host part" and "@" • may be an empty string (all 0)	set by user
destinationIpAddress	Defines the destination IP address, generated out of respective URI's using DNS. • if no filetering is requested shall be set to 0	set by user
userIdentifier	An identifier that is further returned by the indicate function. It is to associate the indication with the request function call	set by user

Response TRDP module → Host

TRDP MD.addListener function has the following format:

Packet offset	Local offset	Field	Type	Description
+20	+0	Handle	DWord	handle for the subscription

TRDP MD.addListener function is coded as follows:

0 … 15	16 … 31
Handle	

Detailed description of fields:

Parameter	Description	Value
Handle	Handle returned by the TRDP layer.	set by user

4.8.15 TRDP MD.remListener function

Request Host → TRDP module

TRDP MD.remListener function has the following format:

Packet offset	Local offset	Field	Type	Description
+20	+0	Handle	DWord	handle for the subscription

TRDP MD.remListener function is coded as follows:

0 … 15	16 … 31
Handle	

Detailed description of fields:

Parameter	Description	Value
Handle	Handle returned by the TRDP layer.	set by user

Response TRDP module → Host
No parameters

4.8.16 TRDP MD.reply function

Request Host → TRDP module

TRDP MD.reply function has the following format:

Packet offset	Local offset	Field	Types	Description
+20	+0	msgType	String[2]	type of message
+22	+2	reserved01	Byte	reserved for future use
+23	+3	reserved02	Byte	reserved for future use
+24	+4	sessionId	DWord[4]	session identifier
+40	+20	comId	DWord	communication identifier
+44	+24	sourceUri	String[32]	source URI (user part)
+76	+56	sourceIpAddress	DWord	source IP address
+80	+60	replyTimeOut	DWord	reply timeout
+84	+64	replyStatus	DWord	reply status
+88	+68	datasetLength	DWord	Length of the user dataset
+92	+72	dataset	Byte[N]	the user dataset

TRDP MD.reply function is coded as follows:

0 … 15	16 … 31	
msgType	reserved01	reserved02
sessionId		
comId		

sourceUri
sourceIpAddress
replyTimeOut
replyStatus
datasetLength
dataset

Detailed description of fields:

Parameter	Description	Value
msgType	Type of the telegram coded in ASCII ‘Mn’ = Notification (Request without reply) ‘Mr’ = MD Request with reply ‘Mp’ = MD Reply without confirmation ‘Mq’ = MD Reply with confirmation ‘Mc’ = MD Confirm ‘Me’= MD error	‘4D6E‘H (‘Mn’) ‘4D72‘H (‘Mr’) ‘4D70‘H (‘Mp’) ‘4D71‘H (‘Mq’) ‘4D63‘H (‘Mc’) ‘4D65‘H (‘Me’)
reserved01	Reserved for future use. • Shall be set to 0	Fixed
reserved02	Reserved for future use. • Shall be set to 0	Fixed
sessionId	A session identifier generated by the TRDP layer	set by user
comId	Identifier of the user dataset.	set by user
sourceUri	The URI of the source • shall be the used for functional addressing • shall be a null terminated string • filling bytes at the end shall be set to 0 • shall contain only the “user part” without “host part” and “@” • may be an empty string (all 0)	set by user
sourceIpAddress	IP address of the source of the PD, encoded in little endian EXAMPLE 192.168.0.1 - 0x0100A8C0	set by user
replyTimeOut	The reply timeout value shall be set to define the expected	0 1..(2^{32}-1)

	reply time in a request / reply session. • Shall be set to 0 for msgTypes without a reply • If infinite type is required shall be set to 0 Unit: µs = 10^{-6} s	
replyStatus	The status value shall be set by the replier to report the execution result of a request message or by the caller sending a confirmation. The execution result is supplied by the replying application and transmitted to the requesting application in addition to the reply message itself. In case of a TRDP error reply ('Me') the value is supplied by TRDP. –1 reserved –2 session abort –3 no replier instance (at replier side) –4 no memory (at replier side) –5 no memory (local) –6 no reply –7 not all replies –8 no confirm –9 reserved –10 sending failed	<0 Not OK 0 OK >0 user reply status
datasetLength	The dataset length: • shall be the length of the user data set in number of octets without padding octets. • shall be the primary information about the user data size NOTE In case of fixed size data sets this is redundant information because the dataset length is already defined by the ComId.	Computed
dataset	The user dataset	

Response TRDP module → Host
No parameters

4.8.17 TRDP MD.release function

Request TRDP module → Host

TRDP MD.release function has the following format:

Packet offset	Local offset	Field	Type	Description
+20	+0	msgType	String[2]	type of message
+22	+2	reserved01	Byte	reserved for future use
+23	+3	reserved02	Byte	reserved for future use
+24	+4	sessionId	DWord[4]	session identifier
+40	+20	etbTopoCnt	DWord	etbTopoCnt value
+44	+24	opTrnTopoCnt	DWord	opTrnTopoCnt value
+48	+28	comId	DWord	communication identifier
+52	+32	sourceUri	String[32]	source URI (user part)
+84	+64	sourceIpAddress	DWord	source IP address
+88	+68	replyStatus	DWord	reply status

TRDP PD.request function is coded as follows:

0 … 15	16 … 31	
msgType	reserved01	reserved02
sessionId		
etbTopoCnt		
opTrnTopoCnt		
comId		
sourceUri		
sourceIpAddress		
replyStatus		

Detailed description of fields:

Parameter	Description	Value
msgType	Type of the telegram coded in ASCII ‘Mn’ = Notification (Request without reply) ‘Mr’ = MD Request with reply ‘Mp’ = MD Reply without confirmation ‘Mq’ = MD Reply with confirmation ‘Mc’ = MD Confirm ‘Me’= MD error	‘4D6E‘H (‘Mn’) ‘4D72‘H (‘Mr’) ‘4D70‘H (‘Mp’) ‘4D71‘H (‘Mq’) ‘4D63‘H (‘Mc’) ‘4D65‘H (‘Me’)
reserved01	Reserved for future use.	Fixed

	• Shall be set to 0	
reserved02	Reserved for future use. • Shall be set to 0	Fixed
sessionId	A session identifier generated by the TRDP layer	set by user
etbTopoCnt	The ETB topography counter: • shall be used (train addressing) as defined in IEC 61375-2-5 (parameter 'etbTopoCnt') • shall be set by the user • shall be set for all communication over the ETB • shall be set if a valid opTrnTopoCnt is set • optional in all other cases. Shall be set to 0 (= invalid) if not used.	$0..2^{32-1}$ set by user
opTrnTopoCnt	The operational train topography counter: • shall be used as defined in 5.3.3 • shall be set by the user • shall be set for communication between functions which use information from the operational train directory (e.g. side selective operations) • shall be set when the source device used the operational train directory to retrieve the destination IP address (e.g. resolving of URI 'vcu.leadVeh.anyCst.anyClTrn.lTrn') • optional in all other cases. Shall be set to 0 (= invalid) if not used	
comId	Identifier of the user dataset.	set by user
sourceUri	The URI of the source • shall be the used for functional addressing • shall be a null terminated string • filling bytes at the end shall be set to 0 • shall contain only the "user part" without "host part" and "@" • may be an empty string (all 0)	set by user
sourceIpAddress	IP address of the source of the PD, encoded in little endian EXAMPLE 192.168.0.1 - 0x0100A8C0	set by user
replyStatus	The status value shall be set by the replier to report the execution result of a request message or by the caller sending a confirmation. The	<0 Not OK 0 OK >0 user reply status

	execution result is supplied by the replying application and transmitted to the requesting application in addition to the reply message itself. In case of a TRDP error reply (‘Me’) the value is supplied by TRDP. –1 reserved –2 session abort –3 no replier instance (at replier side) –4 no memory (at replier side) –5 no memory (local) –6 no reply –7 not all replies –8 no confirm –9 reserved –10 sending failed	

Response Host → TRDP module
No parameters

5 Workflow

Host side ethernet setting should be done during module integration. This setting is permanently stored and usually there are no needs to change it later although it is possible.

TRDP stack is initialized by sending request with TRDP Start function. Any other TRDP functions are not available before start call. TRDP start also configures IP setting of TRDP side of device. TRDP restart function reinitializes TRDP stack, but keeps network setting.

TRDP changeAddress both reinitializes stack and change network settings and thus is equivalent to TRDP stop and subsequent start call.

To restart whole device use ASIMPIP frame 0x89 (reboot request) or perform hard power cycle.

6 Firmware actualization

Actual application of TRDP module is uploaded over FTP, which is running on host side network interface. Current version is readable using ASIMP/IP frame 8fh. The credentials for FTP access are:

User: service
Password: remote

On FTP there are two files app.tar.gz containing applications and app.conf, a configuration file for host side network, this file is generated automatically, use ASIMP/IP frame 88h for change this setting.

It the application file is removed or it is corrupted, a built-in safe application is started. This application supports frames 88h (network settings), 89h (reboot), and 8fh (identification). It identifies itself as “TRDP SAFE APP 1.0”.

6.1 Service mode

It may be possible that malfunctional application is uploaded. In such case a network connection with module could be lost and therefore an upload of correct application would be impossible.

To handle this problem a special procedure could be issued to start safe application.

One second after device boot but before application start, device is listening for special ASIMP/IP frames. These frames have frameType 8fh (device identification) and special payload. These frames are always broadcast and are using long frame format (frameParam 0x1). They are:

6.1.1 Request negotiation

Send as frameType 8fh. Request data has the following format:

Packet offset	Local offset	Field	Type	Description
+8	+0	WE	Word	Always 0x67A3
+10	+2	TEST	STRING[24]	Always “AMIT REQUEST NEGOTIATION”

Payload coded as follows:

0	15
WE	TEST[0-1]
TEST[2-23]	

Detailed description of fields:

Field	Description	Value
WE	Always 0x67A3	Fixed
TEST	Always “AMIT REQUEST NEGOTIATION”	Fixed

6.1.2 Ack negotiation

Reply for request negotiation. Reply data has the following format:

Packet offset	Local offset	Field	Type	Description
+8	+0	OK/KO	Byte	0x00 – Ok 0xFE – Malformed request
+9	+1	IDENT	Byte	Always 0x80
+10	+2	MAC	Byte[6]	MAC address of module
+16	+8	S/N	String[8]	Serial number of device
+24	+16	TEST	String[20]	Always „AMIT ACK NEGOTIATION“

Payload coded as follows:

0	15	
OK/OK	INDENT	MAC[0-1]
MAC[2-5]		
S/N[0-3]		
S/N[4-5]		
TEST[0-19]		

Detailed description of fields:

Field	Description	Value
OK/OK	Request result code	0x00 – Ok 0xFE – Malformed request
IDENT	Indentation	0x80 – fixed
MAC	MAC address of module	Factory data
S/N	Serial number of device	Factory data
TEST	Always „AMIT ACK NEGOTIATION“	Fixed

6.1.3 Request RunLoader

Send as frameType 8fh. Request data has the following format:

Packet offset	Local offset	Field	Type	Description
+8	+0	WE	Word	Always 0xA6B7
+10	+2	MAC	Byte[6]	MAC address of module
+16	+8	S/N	String[8]	Serial number of device

+24	+16	TEST	STRING[23]	Always „AMIT REQUEST RUN LOADER“

Payload coded as follows:

0	15
WE	MAC [0-1]
MAC[2-5]	
S/N[0-3]	
S/N[4-5]	
TEST[0-22]	

Detailed description of fields:

Field	Description	Value
WE	Always 0xA6B7	Fixed
MAC	MAC of target device	User supplied
S/N	Serial number of target device	User supplied
TEST	Always “AMIT REQUEST RUN LOADER“	Fixed

6.1.4 Ack RunLoader

Reply for request RunLoader. Reply data has the following format:

Packet offset	Local offset	Field	Type	Description
+8	+0	OK/KO	Byte	0x00 – Ok 0xFE – Malformed request
+9	+1	IDENT	Byte	Always 0x81
+10	+2	MAC	Byte[6]	MAC address of module
+16	+8	S/N	String[8]	Serial number of device
+24	+16	TEST	String[19]	Always „AMIT ACK NEGOTIATION“

Payload coded as follows:

0		15
OK/OK	INDENT	MAC[0-1]
MAC[2-5]		
S/N[0-3]		
S/N[4-5]		
TEST[0-18]		

Detailed description of fields:

Field	Description	Value
OK/OK	Request result code	0x00 – Ok 0xFE – Malformed request
IDENT	Indentation	0x81 – fixed
MAC	MAC address of module	Factory data
S/N	Serial number of device	Factory data
TEST	Always „AMIT ACK RUN LOADER“	Fixed

6.1.5 Starting the service mode

Hard reset device and start periodically sending broadcast negotiation request. Since there is one second window between boot and application start, the period should be no longer than 100ms.

After device reply with Ack negotiation, usually few second after powerup. Use reported MAC address and serial number to build and send run loader request. One the run loader request is acked, the safe application is started and it is possible to configure host side network of device and access FTP remove malfunctional application.