ERTMS/ATO

Interface Specification

Communication Layers for On-board Communication

REF :	SUBSET-143
ISSUE :	1.0.0
DATE :	05-07-2023

1. MODIFICATION HISTORY

Issue Number Date	Section Number	Modification / Description	Author
0.0.1 15-11-2017	All	First issue of the document	UNISIG ATO WP
0.0.1 07-09-2018	5	Transport layer restrictions added.	UNISIG ATO WP
0.0.1 29-03-2019	5, 6	Exported constraints to application layer and security layer restrictions added.	UNISIG ATO WP
0.0.2 18-06-2019	Footer, 4.2, 5.2.1.3, 5.4.1.2	Review comments from X2Rail-1 WP4 suppliers group incorporated	UNISIG ATO WP
0.0.2 17-09-2019	All	Make it more general for all interfaces used onboard Reference to existing standards	UNISIG ATO WP
0.0.3 29-10-2019	Title, §7.1.1.2	Review ATO WP	UNISIG ATO WP
0.0.4 20-11-2019	§8.1	Variable Definition and details added (used on application layer)	UNISIG ATO WP
0.0.5 16-12-2019	§8.1, §8.2.2.1	Requirements for bitset definition added, L_PACKET description modified	UNISIG ATO WP
0.0.6 15-04-2020	§8.1	8.1.1.11 deleted (MSB first) String description updated (no MSB exist)	UNISIG ATO WP
0.0.7 12-11-2020	8.1, 8.2	Incorporation of review feedback from EECT#062+#066+#068)	UNISIG ATO WP
0.0.8 22-06-2021	1	Remove column author (CR S125-192 – EECT#70 Subset 139 review id 5)	UNISIG ATO WP
0.0.9 25-06-2021	5.2	Remove column author (CR S125-192 – EECT#70 Subset 139 review id 5)	UNISIG ATO WP
0.0.10 27-08-2021	1	Restore column author	UNISIG ATO WP
0.0.11 27-01-2022	All	Type: “ATO over ETCS” changed to “ERTMS/ATO”	UNISIG ATO WP
0.1.0 16-03-2022	None	New versioning	UNISIG ATO WP
0.1.1 21-02-2023	7.2	Implement Action 91.07 Removal of watermark	UNISIG ATO WP

0.1.2 27-03-2023	1	Formal remarks from EECT#94 Update Dates in modification history	UNISIG ATO WP
1.0.0 05-07-2023	None	Version for ERTMS/ATO Baseline 1 1^{st} release and ERTMS/ETCS Baseline 4 1^{st} release	UNISIG ATO WP

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

INTRODUCTION

Scope and purpose of the document

This document specifies the communication layers used by ATO-OB for communication to several other on-board subsystems.

Reference documents

Ref. N°	Title	Reference
[Ref 1]	ERTMS/ATO Glossary	13E154
[Ref 2]	ERTMS/ATO System Requirements Specification	SUBSET-125
[Ref 3]	Glossary of Terms and Abbreviations	SUBSET-023
[Ref 4]	Electronic railway equipment – Train communication network (TCN) – Part 3-4: Ethernet Consist Network (ECN)	IEC 61375-3-4:2014
[Ref 5]	User Datagram Protocol	RFC 768
[Ref 6]	Transmission Control Protocol	RFC 793

Table 1 Reference Documents

Abbreviations

1.1.1.1 For ATO related abbreviations see [Ref 1].

1.1.1.2 For ETCS related abbreviations see [Ref 3].

Definitions

1.1.1.3 For ATO related definitions see ERTMS/ATO Glossary [Ref 1].

1.1.1.4 For ETCS related definitions see SUBSET-023 [Ref 3].

GENERAL

This document provides a precise specification of the communication principles used for the communication between ATO-OB and other on-board subsystems. [Ref 2] explains the architecture of ERTMS/ATO and the interfaces to other subsystems.

The communication takes places between two communication parties: ATO-OB and the other on-board subsystem. For each communication party there exist one logical unit (one to one communication). Other constellation may be possible but are outside of this specification's scope, as further functionalities might be needed (e.g. functional addressing).

Usually the interface description on FFFIS layer is divided into two parts: (i) Application layer interface describing all the functionalities and data exchanged between the communication parties and (ii) the lower levels of communication. This document describes part (ii), whereas some constraints and restrictions are addressed towards part (i) (see 0).

The communication between ATO-OB and other on-board subsystems shall be based on Ethernet. According to [Ref 4] each communication party shall be an End Device (ED).

Each ED shall have only one physical network interface attached to the consist network for this communication purpose (to ensure the one to one relation over this consist network).

In case one of the communication parties is designed in a redundant architecture the responsible supplier needs to make sure that there is at maximum one communication channel active at any time. This channel needs to be specified and configured according to this document. As a result, the redundancy including the potential switchover to the mate is fully transparent to the other communication party. This can be realized e.g. by a router.

LAYERS

Introduction

This protocol specification is divided into separate layers. Figure 1 shows the representation of the different layers according to the Open Systems Interconnection (OSI) model.

(7) Application
(6) Presentation
(5) Session
(4) Transport
(3) Network
(2) Data Link
(1) Physical

Figure 1: OSI Layers

Both communication parties the ATO-OB and other on-board subsystems shall be compliant to the End Device Interface characteristics as specified in [Ref 4]. This affects Layer 1 – Layer 7.

Details to be added to [Ref 4]

Data Class Service Parameters

[Ref 4] distinguish between Process Data and Message Data.

The Ethernet Consist Network implemented shall guarantee the service parameters as defined in [Ref 4] Table 6 for the data classes mentioned above.

Transport Layers

In case Process Data is designed on application layer, the Transport Layer shall be designed according to UDP [Ref 5].

In case Message Data is designed on application layer, the Transport Layer shall be designed according to TCP [Ref 6].

The destination ports shall be configured statically for the application data specified in the Application Layer Document.

Whenever the application layer document specifies the use of Message Data, as the default setting ATO-OB serves as the TCP client i.e., the TCP server is hosted by the other on-board subsystem. If justified, the application layer document can deviate from this but must clearly highlight the changed client/server allocation.

Deviations to [Ref 4]

Any deviations to [Ref 4] are listed here explicitly or in the document containing the application data.

Chapter 0 already provides more details on the transport layer for Message Data. Consequently, the limitation on maximum data size for Message Data as given in [Ref 4] is not applicable, whereas the limitation as given by the protocol itself (see 0, L_PACKET) shall be considered.

EXPORTED CONSTRAINTS TO APPLICATION LAYER DOCUMENT

Definition of the Variables

Variables are used to encode single data values. Variables shall not be split into minor units. The whole variable has one type (meaning).

Variables may have special values which are related to the basic meaning of the variable.

Special values which are not spare, shall have always the highest values in a variable (e.g. 32767 = “unknown”).

Spare values shall be located between the normal and special values in the variable range except where justified.

If bitsets are used, the variables shall be defined starting from the lowest bit position, i.e. at offset zero.

Names of variables shall be unique. A variable shall be used in context with the meaning as described in the variable definition. Variables with different meanings have different names.

All variables names shall have one of the following prefixes:

A_	Acceleration
D_	Distance
L_	Length
M_	Miscellaneous
N_	Number
NC_	Class Number
NID_	Identity Number
Q_	Qualifier
T_	Time/Date
V_	Speed
X_	Text

If signed values are used, the negative values shall be encoded as 2’s complement.

One bit variables (Boolean) shall always use 0 for false and 1 for true.

Offsets for numerical values are avoided (0 is used for 0, 1 for 1, etc.) except where justified.

The variable’s length shall be multiple of a byte unless otherwise specified.

If a variable is shorter than a byte in length, the variable shall be part of a bitset whose length is multiple of a byte.

The encoding order shall respect the order of variables listed in the table related to the packet given in the application layer document (from top to bottom).

The following table defines the data types:

Type	Description	Byte Ordering (Endianness)	Range
UINT8	Unsigned 8bits	address n	0 - 255
UINT16	Unsigned 16bits	address n => high byte address n+1 => low byte	0 - 65535
UINT32	Unsigned 32bits	address n => most significant byte ... address n+3 => least significant byte	0 - (2^{32}-1)
INT16	Signed 16bits	address n => high byte address n+1 => low byte	(-32768) - (32767)
INT32	Signed 32bits	address n => most significant byte ... address n+3 => least significant byte	(-2^{31}) - (2^{31}-1)
BITSET8	Bitset 8bits	address n	NA
BITSET16	Bitset 16bits	address n => high byte address n+1 => low byte	NA
BITSET32	Bitset 32bits	address n => most significant byte … address n+3 => least significant byte	NA
STRING16	String 16bytes	address n => first byte of the string … address n+15 => last byte of the string	NA
BCD32	Binary Coded Decimal 32 bits	address n => most significant byte … address n+3 => least significant byte	NA

Table 2 Data types definition

Representation of Bitsets

Chapter 0 already defines the way BITSET Variables are represented. This chapter gives examples how single variables are represented within a BITSET to avoid ambiguity.

Let B7 be the most significant bit (=2^7) and B0 the least significant bit (=2^0). Table 3 shows the specification of Bitset variables in an application layer document, Table 4 gives an overview on the possible binary representation possibilities.

Item	Variable Name		Description	Data Type	Resolution/Formula
	Bit			BITSET16L	
…	…	…	…		…

Item	Variable Name		Description	Data Type	Resolution/Formula
5	1..2	Q_MISC	Qualifier indicating the ….		**Values:** 00 = Value A 01 = Value B 10 = Value C 11 = Value D
6	3…5	M_STATUS	Status of device X.		**Values:** 0 = Value F 1 = Value G 2 = Value H 3 = Value I 4 = Value J 5 - 7 = spare
…	…	…	…	…	…

Table 3: Example specification of variables specified in a Bitset

Bit Address								Meaning
B^7	B^6	B^5	B^4	B^3	B^2	B^1	B^0	
Variable Q_MISC								
x	x	x	x	x	0	0	x	Value A
x	x	x	x	x	0	1	x	Value B
x	x	x	x	x	1	0	x	Value C
x	x	x	x	x	1	1	x	Value D
Variable M_STATUS								
x	x	0	0	0	x	x	x	0 = Value F
x	x	0	0	1	x	x	x	1 = Value G
x	x	0	1	0	x	x	x	2 = Value H
x	x	0	1	1	x	x	x	3 = Value I
x	x	1	0	0	x	x	x	4 = Value J
x	x	1	0	1	x	x	x	5 = spare
x	x	1	1	0	x	x	x	6 = spare
x	x	1	1	1	x	x	x	7 = spare

Table 4: Representation and Meaning of example variables as part of a Bitset

Definition of the Packets

Packets are multiple variables grouped into a single unit, with a defined internal structure.

Data used by the data classes shall not exceed 1500 bytes (see [Ref 4], Table 6). Consequently, the payload data for Process Data shall be limited to 1472 bytes (=1500 bytes MTU ethernet frame – 20 bytes IP header – 8 bytes UDP header).

The optional checksum specified in [Ref 5] and [Ref 6] is deemed as weak. Thus, undetected errors may occur. To overcome this the user protocol on application layer is requested to provide a CRC.

For all Data Classes Message Data and Process Data the same header structure shall be used. They payload data consists of a header, followed by the application specific data and terminated by a CRC. Figure 2 depicts the structure.

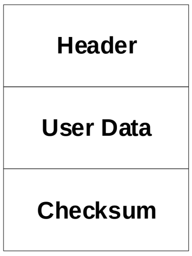

Figure 2: Payload Data Structure

Header Definition

The following table lists the header used on application layer.

Payload Header				
Item	**Variable Name**	**Description**	**Data Type**	**Resolution/Formula**
1	NID_PACKET	ATO Packet number.	UINT8	Numbers 0 .. 30: Slot 1 31 .. 60: Slot 2 61 .. 90: Slot 3 91 .. 120: Slot 4 121 .. 150: Slot 5 151 .. 180: Slot 6 181 .. 210: Slot 7 211 .. 240: Slot 8 241 .. 255: reserved Each application layer interface shall select one slot only. It needs to be made sure that the slots are not reused elsewhere.
2	L_PACKET	Length of Header + User Data length.	UINT16	**Minimum:** 7 **Maximum:** Process Data: 1468 Message Data: 65524 **Resolution:** 1 byte
3	T_TIMESTAMP	Timestamp in milliseconds since start-up. The timestamp will be set at the moment the telegram is issued.	UINT32	**Resolution:** 1 ms

Table 5: Header Structure used on Application Layer

User Data

The definition of User Data is out of scope of this document. It shall be defined in the application layer document.

Checksum

For error detection, the cyclic redundancy check mechanism CRC32, based on the IEEE 802.3 (Ethernet) specification, shall be used with the following parameters:

Parameter	Value
Name	CRC-32/BZIP2
Width	32 bits
Generator-Polynomial	0x{1}04C11DB7
Init	0xFFFFFFFF (-1)
RefIn	False
RefOut	False
Final XOR value (Inversion of CRC) Check value	0xFFFFFFFF (-1) 0xFC891918

Table 6 CRC Parameters

1.1.1.5 The byte order of the CRC is MSB first, LSB last.

1.1.1.6 The CRC value shall be computed over Header and User Data.

SECURITY LAYER

1.1.1.7 The following requirements can be met either by physical protection or cryptographic means upon currently recommended cipher suites, as recommended by ENISA [https://www.enisa.europa.eu/topics/data-protection/security-of-personal-data/cryptographic-protocols-and-tools].

1.1.1.8 The communication parties shall be able to authenticate via cryptographic means or uniquely identify via physical protection their communication party.

1.1.1.9 The communication party shall be able to verify that each received message was not modified.

1.1.1.10 The communication shall be protected against replay attacks.