EtherCAT Specification – Part 1

Overview

Document: ETG.1000.1 S (R) V1.0.4

Nomenclature:

ETG-Number	ETG.1000.1
Type	S (Standard)
State	R (Release)
Version	V1.0.4

Created by: ETG
Contact: info@ethercat.org
Date 15.09.2017

DOCUMENT HISTORY

Version	Comment
1.0.2	Adopted from IEC-Standard 61158-1 Type 12 parts for ETG use only
1.0.3	Update of ETG.1000 series, no changes in this part
1.0.4	Update of ETG.1000 series, no technical changes in this part

CONTENTS

1 Scope

1.1 Scope of this standard and accordance to IEC Standards

The ETG.1000 series specifies the EtherCAT Technology within the EtherCAT Technology Group. It is devided into the following parts:

ETG.1000.1: Overview

ETG.1000.2: Physical Layer service definition and protocol specification

ETG.1000.3: Data Link Layer service definition

ETG.1000.4: Data Link Layer protocol specification

ETG.1000.5: Application Layer service definition

ETG.1000.6: Application Layer protocol specification

These parts are based on the corresponding parts of the IEC 61158 series Type 12. EtherCAT is named Type 12 in IEC 61158 to avoid the usage of brand names.

1.2 Overview

This document presents an overview and guidance for the ETG.1000 series by:

- explaining the structure and content of the ETG.1000 series;
- relating the structure of the ETG.1000 series to the ISO/IEC 7498 OSI Basic Reference Model;
- showing the logical structure of the ETG.1000 series.

2 Normative references

None

3 Abbreviations

For the purposes of this document, the following abbreviations, based partially on the concepts developed in ISO/IEC 7498-1, apply:

AL	Application layer (N = 7)
AR	Application relationship
AREP	Application relationship endpoint
DL-	Data-link layer (as a prefix)
DLL	Data-link layer (N = 2)
IETF	Internet Engineering Task Force
IP	Internet Protocol (see RFC 791)
(*n*)-layer	Layer *n* of the OSI Basic Reference Model
OSI	Open systems interconnection
Ph-	Physical layer (as a prefix)
PhL	Physical layer (N = 1)

4 Concept of the ETG.1000 series

Conceptually, a fieldbus is a digital communication network for integration of industrial control and instrumentation devices into a system. Typical devices are transducers, sensors, actuators and controllers.

The EtherCAT protocol has been engineered to support information processing, monitoring and control systems for any industrial sector and related domains. An example application for high-integrity low-level communication between sensors, actuators and local controllers in a process plant, together with the interconnection of programmable controllers, is shown in Figure 1.

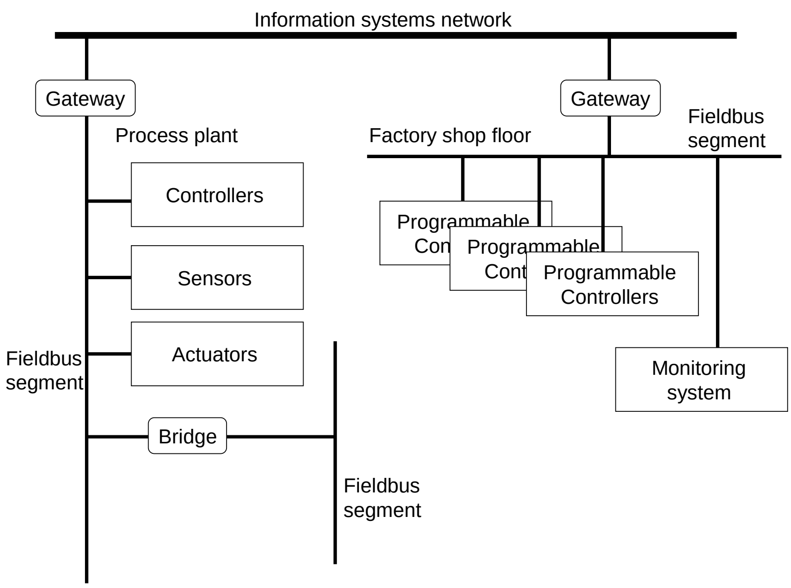

Figure 1 – Generic fieldbus network

Figure 2 illustrates the differences between service and protocol viewpoints of the data-link and application layers. The protocol parts show the layer implementer's view and the service parts show the layer user's oriented view.

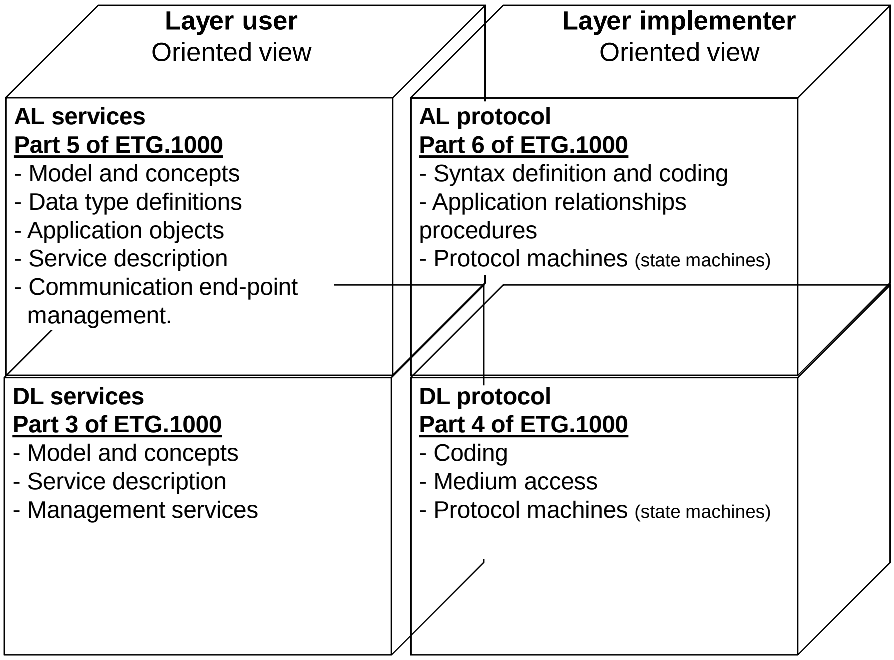

Figure 2 – Concept of DL/AL to separate service and protocol parts

The application layer structure is as follows:

— the "what“ is described by application layer service elements (ASE) in the type-specific parts of ETG.1000-5; and

— the "how“ is described by application layer relationships (AR) in the type-specific parts of ETG.1000-6.

The data-link layer structure is as follows:

— the "what“ is described by data-link layer services and models in the type-specific parts of ETG.1000-3; and

— the "how“ is described by data-link layer protocol machines and medium access principles in the type-specific parts of ETG.1000-5.

The physical layer is structured similarly, but, because its services are readily described, they occur in the same specification (ETG.1000-2) as the definitions of the physical protocols:

— the "what“ is described by physical layer services and models, and

— the "how“ is described by physical layer electromagnetic and mechanical specifications.

5 Mapping onto the OSI Basic Reference Model

5.1 Overview

ETG.1000 protocol types are described using the principles, methodology and model of ISO/IEC 7498. The OSI model provides a layered approach to communications standards, whereby the layers can be developed and modified independently. ETG.1000 specifies functionality from top to bottom of a full OSI stack and, potentially, some functions for the users of the stack. Functions of the intermediate OSI layers, layers 3 through 6, are consolidated into either the ETG.1000 data-link layer or the ETG.1000 application layer, or may be realized by a separate layer. Likewise, some features common to users of the fieldbus application layer may be provided by the ETG.1000 application layer to simplify user operation.

Table 1 shows the OSI layers, their functions, and the equivalent layers in the ETG.1000 basic fieldbus reference model (see Figure 3).

Table 1 – OSI and ETG.1000 layers

OSI layer	Function	ETG.1000 layer
7 **Application**	Translates demands placed on the communications stack into a form understood by the lower layers and vice versa	Application (ETG.1000-5, ETG.1000-6)
6 **Presentation**	Converts data to/from standardized network formats	↑
5 **Session**	Creates and manages dialogue among lower layers	↑
4 **Transport**	Provides transparent reliable data transfer (end-to-end transfer across a network which may include multiple links)	↓ or ↑
3 **Network**	Performs message routing	↓ or ↑
2 **Data-link**	Controls access to the communication medium. Performs error detection, (point-to-point transfer on a link)	Data-link (ETG.1000-3, ETG.1000-4)
1 **Physical**	Encodes/decodes signals for transmission/reception in a form appropriate to the communications medium. Specifies communication media characteristics	Physical (ETG.1000-2)
NOTE ↓ and ↑ indicate that the functionality of this layer, when present, may be included in the fieldbus layer that is nearest in the direction of the arrow. Thus network and transport functionality may be included in either the data-link or application layers, while session and presentation functionality may be included in the application layer but not in the data-link layer.		

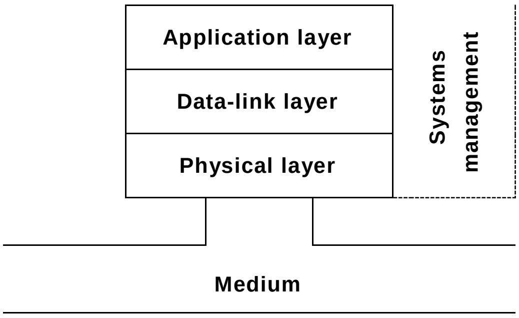

Figure 3 – Basic fieldbus reference model

6 Brief summary of the characteristics of each service and protocol

6.1 Summary of the physical layer service and protocol characteristics

6.1.1 EtherCAT: Wire and optical media

EtherCAT specifies the standard ISO/IEC 8802-3 PhL and the following variant:

— wire medium, 100 Mbit/s, low voltage differential signaling mode (parallel coupling) as specified in ANSI TIA/EIA-644-A.

6.2 Summary of data-link layer service characteristics

EtherCAT supports a DL-service which provides a connectionless subset of those services specified in ISO/IEC 8886.

6.3 Summary of data-link layer protocol characteristics

EtherCAT supports a DL-protocol for the EtherCAT DL-service. The maximum system size is an unlimited number of segments of 2^{16} nodes each. Each node has a maximum of 2^{16} related peer and publisher/subscriber DLCEPs.

6.4 Summary of application layer service characteristics

Fieldbus application layer (FAL) services and protocols are provided by FAL application-entities (AE) contained within the application processes. The FAL AE is composed of a set of object-oriented application service elements (ASEs) and a layer management entity (LME) that manages the AE. The ASEs provide communication services that operate on a set of related application process object (APO) classes. One of the FAL ASEs is a management ASE that provides a common set of services for the management of the instances of FAL classes.

Although these services specify, from the perspective of applications, how request and responses are issued and delivered, they do not include a specification of what the requesting and responding applications are to do with them. That is, the behavioral aspects of the applications are not specified; only a definition of what requests and responses they can send/receive is specified. This permits greater flexibility to the FAL users in standardizing such object behavior.

EtherCAT supports an application service which provides connectionless cyclic exchange of data and for spontaneous communication for different ASEs.

6.5 Summary of application layer protocol characteristics

The FAL is an application layer communication standard designed to support the conveyance of time-critical application requests and responses among devices in an automation environment. The term “time-critical” is used to represent the presence of a time window, within which one or more specified actions are required to be completed with some defined level of certainty. Failure to complete specified actions within the time window risks failure of the applications requesting the actions, with attendant risk to equipment, plant and possibly human life.

EtherCAT supports an application protocol which specifies abstract syntax, coding and behavior of the EtherCAT application service elements.

7 Data type ASE

7.1 Formal definition of data type objects

7.1.1 Data type class

7.1.1.1 Template

The data type class specifies the root of the data type class tree. Its parent class "top" indicates the top of the FAL class tree.

FAL ASE:			**DATA TYPE ASE**
CLASS:			**DATA TYPE**
CLASS ID:			5 (FIXED LENGTH & STRING), 6 (STRUCTURE), 12 (ARRAY)
PARENT CLASS:			TOP
ATTRIBUTES:			
1	(o)	Key attribute:	Data type numeric identifier
2	(o)	Key attribute:	Data type name
3	(m)	Attribute:	Format (FIXED LENGTH, STRING, STRUCTURE, ARRAY)
4	(c)	Constraint:	Format = FIXED LENGTH | STRING
4.1	(m)	Attribute:	Octet length
5	(c)	Constraint:	Format = STRUCTURE
5.1	(m)	Attribute:	Number of fields
5.2	(m)	Attribute:	List of fields
5.2.1	(o)	Attribute:	Field name
5.2.2	(m)	Attribute:	Field data type
6	(c)	Constraint:	Format = ARRAY
6.1	(m)	Attribute:	Number of array elements
6.2	(m)	Attribute :	Array element data type

7.1.1.2 Attributes

Data type numeric identifier

This optional attribute identifies the numeric identifier of the related data type.

Data type name

This optional attribute identifies the name of the related data type.

Format

This mandatory attribute identifies the data type as a fixed-length, string, array, or data structure.

Octet length

This conditional attribute defines the representation of the dimensions of the associated type object. It is present when the value of the format attribute is "FIXED LENGTH" or "STRING". For FIXED LENGTH data types, it represents the length in octets. For STRING data types, it represents the length in octets for a single element of a string.

Number of fields

This conditional attribute defines the number of fields in a structure. It shall be present when the value of the format attribute is "STRUCTURE".

List of fields

This conditional attribute is an ordered list of fields contained in the structure. Each field is specified by its number and its type. Fields shall be numbered sequentially from 0 (zero) in the order in which they occur. Partial access to fields within a structure is supported by identifying the field by number. This attribute shall be present when the value of the format attribute is "STRUCTURE".

Field name

This conditional, optional attribute specifies the name of the field. It may be present when the value of the format attribute is "STRUCTURE".

Field data type

This conditional attribute specifies the data type of the field. It shall be present when the value of the format attribute is "STRUCTURE". This attribute may itself specify a constructed data type either by referencing a constructed data type definition by its numeric id, or by embedding a constructed data type definition here. When embedding a description, the embedded-data-type description shown below shall be used.

Number of array elements

This conditional attribute defines the number of elements for the array type. Array elements shall be indexed starting at “0” through “n-1” where the size of the array is “n” elements. This attribute shall be present when the value of the format attribute is "ARRAY".

Array element data type

This conditional attribute specifies the data type for the elements of an array. All elements of the array shall have the same data type. It shall be present when the value of the format attribute is "ARRAY". This attribute may itself specify a constructed data type either by referencing a constructed data type definition by its numeric id, or by embedding a constructed data type definition here. When embedding a description, the embedded-data-type description shown below shall be used.

Embedded-data-type description

This attribute is used to recursively define embedded data types within a structure or array. The template below defines its contents. The attributes shown in the template are defined above in the data type class, except for the embedded-data-type attribute, which is a recursive reference to this attribute. It is used to define nested elements.

ATTRIBUTES:

1	(m)	Attribute:	Format (FIXED LENGTH, STRING, STRUCTURE, ARRAY)
2	(c)	Constraint:	Format = FIXED LENGTH | STRING
2.1	(m)	Attribute:	Data type numeric ID value
2.2	(m)	Attribute:	Octet length
3	(c)	Constraint:	Format = STRUCTURE
3.1	(m)	Attribute:	Number of fields
3.2	(m)	Attribute:	List of fields
3.2.1	(m)	Attribute:	Embedded data type description
4	(c)	Constraint:	Format = ARRAY
4.1	(m)	Attribute:	Number of array elements
4.2	(m)	Attribute:	Embedded data type description