

Function blocks - Part 1: Architecture (IEC 61499-1:2012)

This document is a preview generated by EVS

EESTI STANDARDI EESSÕNA

NATIONAL FOREWORD

See Eesti standard EVS-EN 61499-1:2013 sisaldab Euroopa standardi EN 61499-1:2013 ingliskeelset teksti.	This Estonian standard EVS-EN 61499-1:2013 consists of the English text of the European standard EN 61499-1:2013.
Standard on jõustunud sellekohase teate avaldamisega EVS Teatajas.	This standard has been endorsed with a notification published in the official bulletin of the Estonian Centre for Standardisation.
Euroopa standardimisorganisatsioonid on teinud Euroopa standardi rahvuslikele liikmetele kättesaadavaks 15.02.2013.	Date of Availability of the European standard is 15.02.2013.
Standard on kättesaadav Eesti Standardikeskusest.	The standard is available from the Estonian Centre for Standardisation.

Tagasisidet standardi sisu kohta on võimalik edastada, kasutades EVS-i veebilehel asuvat tagasiside vormi või saates e-kirja meiliaadressile standardiosakond@evs.ee.

ICS 25.040, 35.240.50

Standardite reprodutseerimise ja levitamise õigus kuulub Eesti Standardikeskusele

Andmete paljundamine, taastekitamine, kopeerimine, salvestamine elektroonsesse süsteemi või edastamine ükskõik millises vormis või millisel teel ilma Eesti Standardikeskuse kirjaliku loata on keelatud.

Kui Teil on küsimusi standardite autorikaitse kohta, võtke palun ühendust Eesti Standardikeskusega:
Aru 10, 10317 Tallinn, Eesti; www.evs.ee; telefon 605 5050; e-post info@evs.ee

The right to reproduce and distribute standards belongs to the Estonian Centre for Standardisation

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, without a written permission from the Estonian Centre for Standardisation.

If you have any questions about copyright, please contact Estonian Centre for Standardisation:
Aru 10, 10317 Tallinn, Estonia; www.evs.ee; phone 605 5050; e-mail info@evs.ee

English version

**Function blocks -
Part 1: Architecture
(IEC 61499-1:2012)**

Blocs fonctionnels -
Partie 1: Architecture
(CEI 61499-1:2012)

Funktionsbausteine für industrielle
Leitsysteme -
Teil 1: Architektur
(IEC 61499-1:2012)

This European Standard was approved by CENELEC on 2012-12-12. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

Management Centre: Avenue Marnix 17, B - 1000 Brussels

Foreword

The text of document 65B/845/FDIS, future edition 2 of IEC 61499-1, prepared by SC 65B "Measurement and control devices" of IEC/TC 65 "Industrial-process measurement, control and automation" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 61499-1:2013.

The following dates are fixed:

- latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement (dop) 2013-09-12
- latest date by which the national standards conflicting with the document have to be withdrawn (dow) 2015-12-12

This document supersedes EN 61499-1:2005.

EN 61499-1:2013 includes the following significant technical changes with respect to EN 61499-1:2005:

- *Execution control* in basic function blocks (5.2) has been clarified and extended:
 - dynamic and static parts of the EC transition condition are clearly delineated by using the `ec_transition_event[guard_condition]` syntax of the Unified Modeling Language (UML) (5.2.1.3, B.2.1);
 - the terminology "crossing of an EC transition" (3.10) is used preferentially to "clearing" to avoid the misinterpretation that the entire transition condition corresponds to a Boolean variable that can be "cleared.";
 - operation of the ECC state machine in 5.2.2.2 has been clarified and made more rigorous;
 - event and data outputs of adapter instances (plugs and sockets) can be used in EC transition conditions, and event inputs of adapter instances can be used as EC action outputs.
- *Temporary variables* (3.97) can be declared (B.2.1) and used in algorithms of basic function blocks.
- *Service sequences* (6.1.3) can now be defined for basic and composite function block types and adapter types, as well as service interface types.
- *The syntax for mapping* of FB instances from applications to resources has been simplified (Clause B.3).
- Syntax for definition of *segment types* (7.2.3) for network segments of system configurations has been added (Clause B.3).
- Function block types for interoperation with programmable controllers are defined (Clause D.6).
- The READ/WRITE management commands (Table 8) now apply only to *parameters*.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CENELEC [and/or CEN] shall not be held responsible for identifying any or all such patent rights.

Endorsement notice

The text of the International Standard IEC 61499-1:2012 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following notes have to be added for the standards indicated:

IEC 61131-5:2000	NOTE	Harmonised as EN 61131-5:2001 (not modified).
IEC 61499 Series	NOTE	Harmonised as EN 61499 Series (not modified).
IEC 61499-2:2012	NOTE	Harmonised as EN 61499-2:2013 (not modified).
IEC 61499-4	NOTE	Harmonised as EN 61499-4.

This document is a preview generated by EVS

Annex ZA (normative)

Normative references to international publications with their corresponding European publications

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

NOTE When an international publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

<u>Publication</u>	<u>Year</u>	<u>Title</u>	<u>EN/HD</u>	<u>Year</u>
IEC 61131-1	-	Programmable controllers - Part 1: General information	EN 61131-1	-
IEC 61131-3	2003	Programmable controllers - Part 3: Programming languages	EN 61131-3	2003
ISO/IEC 7498-1	1994	Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model	-	-
ISO/IEC 8824-1	2008	Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation	-	-
ISO/IEC 10646	2003	Information technology - Universal multiple- octet coded character set (UCS)	-	-

CONTENTS

FOREWORD.....	5
INTRODUCTION.....	7
1 Scope.....	8
2 Normative references	8
3 Terms and definitions	9
4 Reference models	18
4.1 System model.....	18
4.2 Device model	19
4.3 Resource model	19
4.4 Application model.....	21
4.5 Function block model.....	21
4.5.1 Characteristics of function block instances	21
4.5.2 Function block type specifications	23
4.5.3 Execution model for basic function blocks	23
4.6 Distribution model	25
4.7 Management model.....	25
4.8 Operational state models.....	27
5 Specification of function block, subapplication and adapter interface types.....	27
5.1 Overview	27
5.2 Basic function blocks.....	28
5.2.1 Type declaration	28
5.2.2 Behavior of instances	30
5.3 Composite function blocks.....	33
5.3.1 Type specification.....	33
5.3.2 Behavior of instances	35
5.4 Subapplications	36
5.4.1 Type specification.....	36
5.4.2 Behavior of instances	37
5.5 Adapter interfaces	38
5.5.1 General principles	38
5.5.2 Type specification.....	38
5.5.3 Usage.....	39
5.6 Exception and fault handling.....	41
6 Service interface function blocks	41
6.1 General principles	41
6.1.1 General	41
6.1.2 Type specification.....	42
6.1.3 Behavior of instances	43
6.2 Communication function blocks	45
6.2.1 Type specification.....	45
6.2.2 Behavior of instances	46
6.3 Management function blocks	47
6.3.1 Requirements	47
6.3.2 Type specification.....	47
6.3.3 Behavior of managed function blocks.....	50
7 Configuration of functional units and systems	52

7.1	Principles of configuration	52
7.2	Functional specification of resource, device and segment types	52
7.2.1	Functional specification of resource types	52
7.2.2	Functional specification of device types	53
7.2.3	Functional specification of segment types	53
7.3	Configuration requirements	53
7.3.1	Configuration of systems	53
7.3.2	Specification of applications	54
7.3.3	Configuration of devices and resources	54
7.3.4	Configuration of network segments and links	55
Annex A (normative)	Event function blocks	56
Annex B (normative)	Textual syntax	63
Annex C (informative)	Object models	74
Annex D (informative)	Relationship to IEC 61131-3	82
Annex E (informative)	Information exchange	92
Annex F (normative)	Textual specifications	100
Annex G (informative)	Attributes	113
Bibliography	117
Figure 1 – System model		18
Figure 2 – Device model		19
Figure 3 – Resource model		20
Figure 4 – Application model		21
Figure 5 – Characteristics of function blocks		22
Figure 6 – Execution model		24
Figure 7 – Execution timing		24
Figure 8 – Distribution and management models		26
Figure 9 – Function block and subapplication types		28
Figure 10 – Basic function block type declaration		29
Figure 11 – ECC example		30
Figure 12 – ECC operation state machine		32
Figure 13 – Composite function block <code>PI_REAL</code> example		34
Figure 14 – Basic function block <code>PID_CALC</code> example		35
Figure 15 – Subapplication <code>PI_REAL_APPL</code> example		37
Figure 16 – Adapter interfaces – Conceptual model		38
Figure 17 – Adapter type declaration – graphical example		39
Figure 18 – Illustration of provider and acceptor function block type declarations		40
Figure 19 – Illustration of adapter connections		41
Figure 20 – Example service interface function blocks		43
Figure 21 – Example service sequence diagrams		44
Figure 22 – Generic management function block type		47
Figure 23 – Service primitive sequences for unsuccessful service		48
Figure 24 – Operational state machine of a managed function block		51
Figure A.1 – Event split and merge		62

Figure C.1 – ESS overview	74
Figure C.2 – Library elements	75
Figure C.3 – Declarations	76
Figure C.4 – Function block network declarations	77
Figure C.5 – Function block type declarations	79
Figure C.6 – IPMCS overview	79
Figure C.7 – Function block types and instances	81
Figure D.1 – Example of a “simple” function block type	82
Figure D.2 – Function block type READ	85
Figure D.3 – Function block type UREAD	87
Figure D.4 – Function block type WRITE	88
Figure D.5 – Function block type TASK	90
Figure E.1 – Type specifications for unidirectional transactions	93
Figure E.2 – Connection establishment for unidirectional transactions	93
Figure E.3 – Normal unidirectional data transfer	93
Figure E.4 – Connection release in unidirectional data transfer	94
Figure E.5 – Type specifications for bidirectional transactions	94
Figure E.6 – Connection establishment for bidirectional transaction	95
Figure E.7 – Bidirectional data transfer	95
Figure E.8 – Connection release in bidirectional data transfer	95
Table 1 – States and transitions of ECC operation state machine	32
Table 2 – Standard inputs and outputs for service interface function blocks	42
Table 3 – Service primitive semantics	45
Table 4 – Variable semantics for communication function blocks	46
Table 5 – Service primitive semantics for communication function blocks	46
Table 6 – <i>CMD</i> input values and semantics	48
Table 7 – <i>STATUS</i> output values and semantics	48
Table 8 – Command syntax	49
Table 9 – Semantics of actions in Figure 24	52
Table A.1 – Event function blocks	57
Table C.1 – ESS class descriptions	75
Table C.2 – Syntactic productions for library elements	75
Table C.3 – Syntactic productions for declarations	77
Table C.4 – IPMCS classes	80
Table D.1 – Semantics of <i>STATUS</i> values	83
Table D.2 – Source code of function block type READ	86
Table D.3 – Source code of function block type UREAD	87
Table D.4 – Source code of function block type WRITE	89
Table D.5 – Source code of function block type TASK	90
Table D.6 – IEC 61499 interoperability features	91
Table E.1 – COMPACT encoding of fixed length data types	99
Table G.1 – Elements of attribute definitions	114

INTRODUCTION

IEC 61499 consists of the following parts, under the general title *Function blocks*:

- Part 1 (this document) contains:
 - general requirements, including scope, normative references, definitions, and reference models;
 - rules for the declaration of *function block types*, and rules for the behavior of *instances* of the types so declared;
 - rules for the use of function blocks in the *configuration* of distributed industrial-process measurement and control *systems* (IPMCSs);
 - rules for the use of function blocks in meeting the communication requirements of distributed IPMCSs;
 - rules for the use of function blocks in the management of *applications, resources* and *devices* in distributed IPMCSs.
- Part 2 defines requirements for *software tools* to support the following systems engineering tasks:
 - the specification of *function block types*;
 - the functional specification of *resource types* and *device types*;
 - the specification, analysis, and validation of distributed IPMCSs;
 - the *configuration, implementation, operation, and maintenance* of distributed IPMCSs;
 - the exchange of *information* among *software tools*.
- Part 3 (Tutorial information) has been withdrawn due to the widespread current availability of tutorial and educational materials regarding IEC 61499. However, an updated 2nd Edition of Part 3 may be developed in the future.
- Part 4 defines rules for the development of *compliance profiles* which specify the features of IEC 61499-1 and IEC 61499-2 to be implemented in order to promote the following attributes of IEC 61499-based systems, devices and software tools:
 - interoperability of devices from multiple suppliers;
 - portability of software between software tools of multiple suppliers; and
 - configurability of devices from multiple vendors by software tools of multiple suppliers.

FUNCTION BLOCKS –

Part 1: Architecture

1 Scope

This part of IEC 61499 defines a generic architecture and presents guidelines for the use of *function blocks* in distributed industrial-process measurement and control systems (IPMCSs). This architecture is presented in terms of implementable reference *models*, textual syntax and graphical representations. These models, representations and syntax **can be used for**:

- the specification and standardization of *function block types*;
- the functional specification and standardization of system elements;
- the implementation independent specification, analysis, and validation of distributed IPMCSs;
- the *configuration, implementation, operation, and maintenance* of distributed IPMCSs;
- the exchange of *information* among *software tools* for the performance of the above *functions*.

This part of IEC 61499 does not restrict or specify the functional capabilities of IPMCSs or their system elements, except as such capabilities are represented using the elements defined herein. IEC 61499-4 addresses the extent to which the elements defined in this standard may be restricted by the functional capabilities of compliant systems, subsystems, and devices.

Part of the purpose of this standard is to provide reference models for the use of function blocks in other standards dealing with the support of the system life cycle, including system planning, design, implementation, validation, operation and maintenance. The models given in this standard are intended to be generic, domain independent and extensible to the definition and use of function blocks in other standards or for particular applications or application domains. It is intended that specifications written according to the rules given in this standard be concise, implementable, complete, unambiguous, and consistent.

NOTE 1 The provisions of this standard alone are not sufficient to ensure interoperability among devices of different vendors. Standards complying with this part of IEC 61499 can specify additional provisions to ensure such interoperability.

NOTE 2 Standards complying with this part of IEC 61499 can specify additional provisions to enable the performance of *system, device, resource* and *application* management *functions*.

2 Normative references

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

IEC 61131-1, *Programmable controllers – Part 1: General*

IEC 61131-3:2003, *Programmable controllers – Part 3: Programming languages*

IEC/ISO 7498-1:1994, *Information technology – Open systems interconnection – Basic reference model: The basic model*

ISO/IEC 8824-1:2008, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*

ISO/IEC 10646:2003, *Information technology – Universal Multiple-Octet Coded Character Set (UCS)*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

NOTE Terms defined in Clause 3 are *italicized* where they appear in definitions and Notes to entry of other terms as well as throughout the body of the document.

3.1

acceptor

function block instance which provides a *socket adapter* of a defined *adapter interface type*

3.2

adapter connection

connection from a *plug adapter* to a *socket adapter* of the same *adapter interface type*, which carries the flows of *data* and *events* defined by the *adapter interface type*

3.3

adapter interface type

type which consists of the definition of a set of *event inputs*, *event outputs*, *data inputs*, and *data outputs*, and whose *instances* are *plug adapters* and *socket adapters*

3.4

algorithm

finite set of well-defined rules for the solution of a problem in a finite number of *operations*

3.5

application

software functional unit that is specific to the solution of a problem in industrial-process measurement and control

Note 1 to entry: An application can be distributed among *resources*, and might communicate with other applications.

3.6

attribute

property or characteristic of an *entity*, for instance, the version identifier of a *function block type* specification

3.7

basic function block type

function block type that cannot be decomposed into other function blocks and that utilizes an *execution control chart (ECC)* to control the *execution* of its *algorithms*

3.8

bidirectional transaction

transaction in which a request and possibly *data* are conveyed from an *requester* to a *responder*, and in which a response and possibly *data* are conveyed from the responder back to the requester