Lecture 5: Introductory concepts of IEC 61499

Valeriy Vyatkin © 2007
IEC 61499:

A New Open Standard of International Electrotechnical Commission

A component-based, open architecture for Distributed Industrial - Process Measurement & Control Systems (IPMCS) which can meet both current and future requirements for intelligent automation.

*architecture*: The structure and relationship among functional units in a system. The architecture may also include the system's interfaces with its environment.

PLC Function Blocks (IEC 61131-3)

DCS Function Blocks (IEC 61804 project)
Structure of IEC61499

Part 1
Architecture
• Structures
• Function blocks
• Service Interface FBs
• Compliance of devices

Part 2
Software Tool Requirements
• Library elements, declarations
• Document Type Definitions
• Graphics Models

Part 3
Application Guidelines
• Engineering models
• Examples

Part 4
Guidelines for Compliance Profiles
• Portability
• Interoperability
• Configurability

Parts 1 and 2, 4 are approved as IEC Standards as of 2005

Has been approved as a Technical Report

IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design, Chapter 5, V. Vyatkin, © 2007
IEC61499: opinions and misunderstandings

Some industry professionals think that:
- IEC61499 is a new programming language
- IEC61499 is a high-level network protocol
- IEC61499 is not needed. We can do all in Visual Basic (Java, C++, C#, etc.)
- The model of IEC61499 is impossible to implement
- Maybe IEC61499 fits well to the control, but visualization surely is better done with WinCC, Wonderware, Intellusion, etc. …
- Compliance with IEC61499 can be reached by means of:
  - new engineering tools
  - the use of the strange FB shape
Elements

- Basic FBT
- Composite FBT
- SIFBT
- SubapplicationTypeDeclaration
- AdapterTypeDeclaration
- DataTypeDeclaration
- LibraryElement
- ResourceTypeDeclaration
- DeviceTypeDeclaration
- SystemConfiguration
Function Block Interface
Event vs Boolean variable
Event-driven Data Exchange

The event output “eo” of the function block FB1 is connected by an event connection with the event input “ei” of the function block FB2. Once the block FB1 emits the event “eo”, it triggers the execution of the block FB2.

The values of input parameters “d” and “e” will be updated before the execution starts because they are associated with the event input “ei” of FB2 only the value of input variable “e” will be actualized as a result of FB1’s execution.
FUNCTION_BLOCK Sample (* Basic Function Block *)

EVENT_INPUT
  INIT WITH QI; (* Initialization Request *)
  REQ WITH PARAM; (* Normal Execution Request *)

END_EVENT

EVENT_OUTPUT
  INITO WITH QO; (* Initialization Confirm *)
  RES1 WITH RESULT;
  RES2 WITH QO,RESULT;

END_EVENT

VAR_INPUT
  QI : BOOL; (* Input event qualifier *)
  PARAM : INT;

END_VAR

VAR_OUTPUT
  QO : BOOL; (* Output event qualifier *)
  RESULT : INT;

END_VAR

END_FUNCTION_BLOCK
Summary

- A function block is an abstraction of a functional component that can be implemented as software or hardware.
- Events are used to initiate function block execution and to define explicitly which data are to be refreshed at the data transfer.
- The interface of function blocks includes: function block type name, input and output events, names and data types of input and output variables, and associations between events and data variables.
- Data of function blocks can be of the same types as in IEC 61131.
- Function block types are developed once and then can be instantiated over and over again in definitions of composite function blocks or in system configurations.