EULYNX Initiative

Europe's Rail Joint Undertaking

Requirements specification for subsystem Generic IO

Document number: Eu.Doc.45
Version: 4.3 (1.A)

Contents

1 Introduction 1
1.1 Release information 1
1.2 Impressum 1
1.3 Purpose 2
1.4 Applicable standards and regulations 2
1.5 Applicable documents 2
1.6 Terms and abbreviations 2
1.7 Variability management 2
1.8 Definition of object types 2
1.9 Modelling 2

2 Conditions of use 2
2.1 Functional packages 3

3 Functional requirements specification 3
3.1 Subsystem - Generic IO - General Infos and Assumptions 3
3.2 Subsystem - Generic IO - Logical Viewpoint 3
3.2.1 Subsystem - Generic IO - Logical Context 3
3.3 Subsystem - Generic IO - Functional Viewpoint 4
3.3.1 Definition of time values 4
3.3.2 Subsystem - Generic IO - Functional Context 5
3.3.3 Subsystem - Generic IO - Functional Partitioning 19
3.3.4 Subsystem - Generic IO - Functional Architecture 19
3.3.5 Subsystem - Generic IO - Functional Entities 21
3.4 Subsystem - Generic IO - Interfaces 33
3.4.1 SCI-IO (Subsystem - Electronic Interlocking) 33
3.4.1.1 SCI-IO - Logical Viewpoint 33
3.4.1.1.1 SCI-IO - Logical Context 33
3.4.1.2 SCI-IO - Information Flows 34
3.4.1.3 SCI-IO - Functional Viewpoint 35
3.4.1.3.1 SCI-IO - Functional Partitioning 35
3.4.1.3.2 SCI-IO - Functional Architecture 35
3.4.1.3.3 SCI-IO - Functional Entities 36
3.4.2 SMI-IO (Subsystem - Maintenance and Data Management) 45
3.4.3 SDI-IO (Subsystem - Maintenance and Data Management) 45
3.4.4 SSI-IO (Subsystem - Security Services Platform) 45
3.4.5 IO1 (Basic Data identifier) 45
3.4.6 IO5 (Maintainer) 45
3.4.7 IO2 (Adjacent IO Systems) 45
3.4.8 IO3 (Adjacent IO Systems) 46

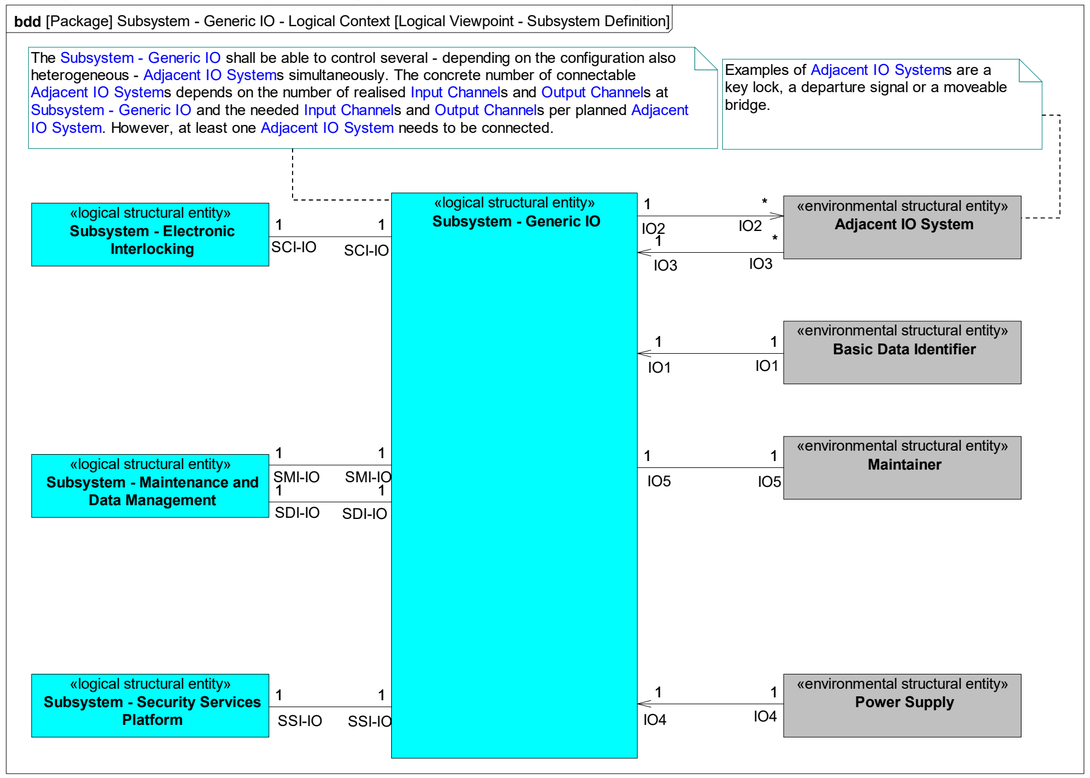
4 RAMSS Requirements 46

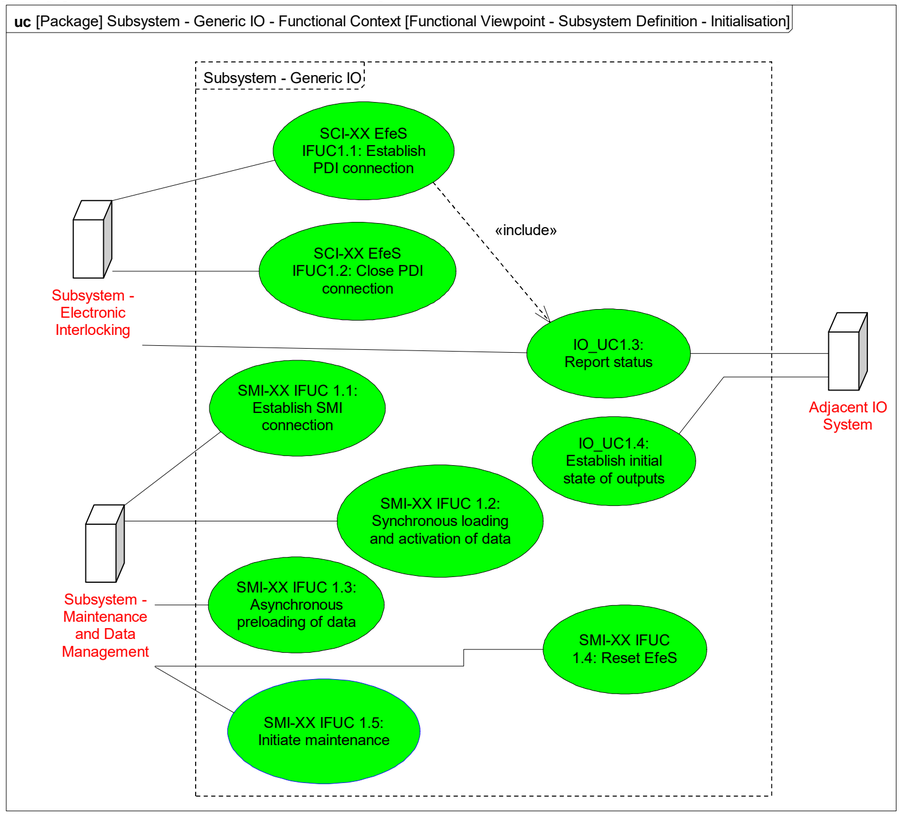
5 Technical requirements 46
5.1 Specific technical interface requirements 46
5.1.1 Interface to the Point of Service Signalling (PoS-Signalling) 46
5.1.2 Interfaces to Adjacent IO System 46
5.1.3 Examples for wiring schemes of Adjacent IO Systems 46
5.2 Time behaviour 46
5.2.1 Response times 46
5.3 Configuration and engineering data 46
5.3.1 Specific data 46
5.3.2 Value configuration 47

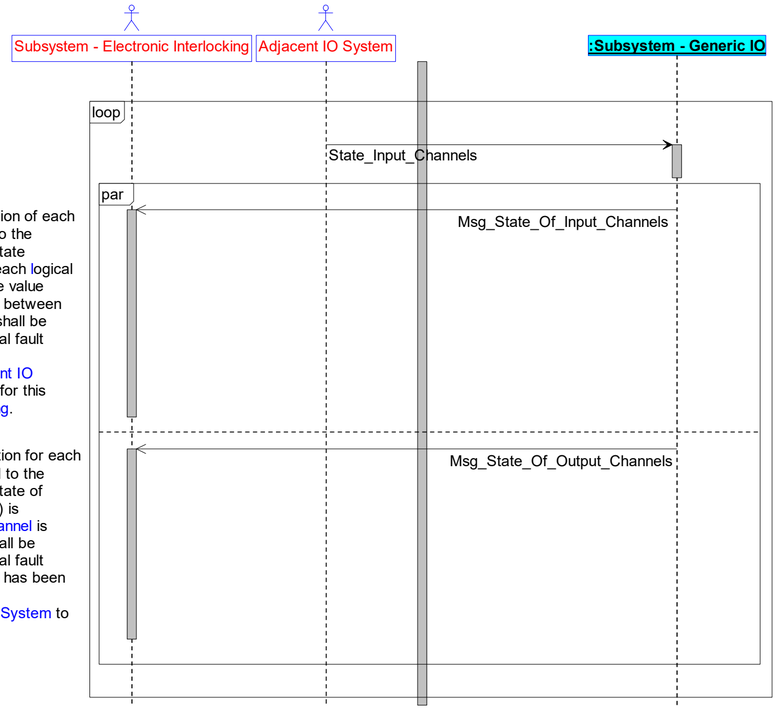
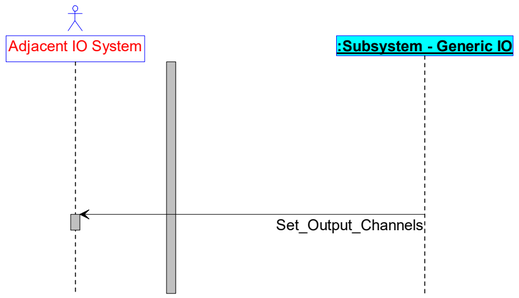
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.1	Head	**1 Introduction**		
Eu.IO.2	Head	**1.1 Release information**		
Eu.IO.3	Info	[Eu.Doc.45] Requirements specification for subsystem Generic IO CENELEC Phase: 4 Version: 4.3 (1.A) Approval date: 02.06.2025		
Eu.IO.1639	Info	**Version history**		
Eu.IO.7995	Info	version number: 4.0 (0.A) date: 16.05.2022 author: Jorge Block model version: 18 Generic interface and subsystem requirements version: 4.0 (0.A) Generic interface and subsystem requirements for SCI version: 1.0 (0.A) review: CCB changes: EUIO-368, EUIO-369		
Eu.IO.8028	Info	version number: 4.1 (0.A) date: 06.04.2023 author: Jorge Block model version: 21 Generic interface and subsystem requirements version: 4.0 (0.A) Generic interface and subsystem requirements for SCI version: 1.0 (0.A) review: Cluster changes: EUIO-376, EUIO-377		
Eu.IO.8037	Info	version number: 4.2 (0.A) date: 27.06.2023 author: Jorge Block model version: 22 Generic interface and subsystem requirements version: 4.0 (3.A) Generic interface and subsystem requirements for SCI version: 1.0 (3.A) review: TACS Mirror Group changes: EUIO-378, EUIO-380, EUIO-381, EUIO-385, EUIO-386, EUIO-387, EUIO-389, EUIO-390, EUIO-392, EUIO-394, EUIO-395		
Eu.IO.8060	Info	version number: 4.2 (1.A) date: 15.12.2023 author: Jorge Block, Philipp Wolber model version: 25 Generic interface and subsystem requirements version: 4.0 (4.A) Generic interface and subsystem requirements for SCI version: 1.0 (4.A) review: M&T changes: EUIO-375, EUIO-403, EUIO-404, EUIO-405, EUIO-406, EUIO-407, EUIO-408, EUIO-409, EUIO-410, EUIO-411, EUIO-412		
Eu.IO.8107	Info	version number: 4.2 (2.A) date: 21.03.2024 author: Jorge Block, Ricky Holz model version: 26 Generic interface and subsystem requirements version: 4.0 (4.A) Generic interface and subsystem requirements for SCI version: 1.0 (4.A) review: cluster changes: EUIO-382, EUIO-414, EUIO-415, EUIO-416, EUIO-417, EUIO-418, EUIO-419, EUIO-420, EUIO-421		
Eu.IO.8116	Info	version number: 4.3 (0.A) date: 18.06.2024 author: Jorge Block, Ricky Holz model version: 26 Generic interface and subsystem requirements version: 4.0 (6.A) Generic interface and subsystem requirements for SCI version: 1.1 (0.A) review: TACS Mirror Group changes: EUIO-428, EUIO-429, EUIO-430, EUIO-431, EUIO-432, EUIO-433, EUIO-434, EUIO-436		
Eu.IO.8167	Info	version number: 4.3 (1.A) date: 20.06.2025 author: Jorge Block, Ricky Holz model version: 29 Generic interface and subsystem requirements version: 4.0 (7.A) Generic interface and subsystem requirements for SCI version: 1.1 (2.A) review: TACS Mirror Group changes: EUIO-440, EUIO-443		
Eu.IO.7	Head	**1.2 Impressum**		
Eu.IO.8	Info	Publishers: **Europe's Rail Joint Undertaking** https://rail-research.europa.eu/ **EULYNX Initiative** https://eulynx.eu/		
Eu.IO.9	Info	Responsible for this document: EU-Rail System Pillar Trackside Assets Control and Supervision domain		

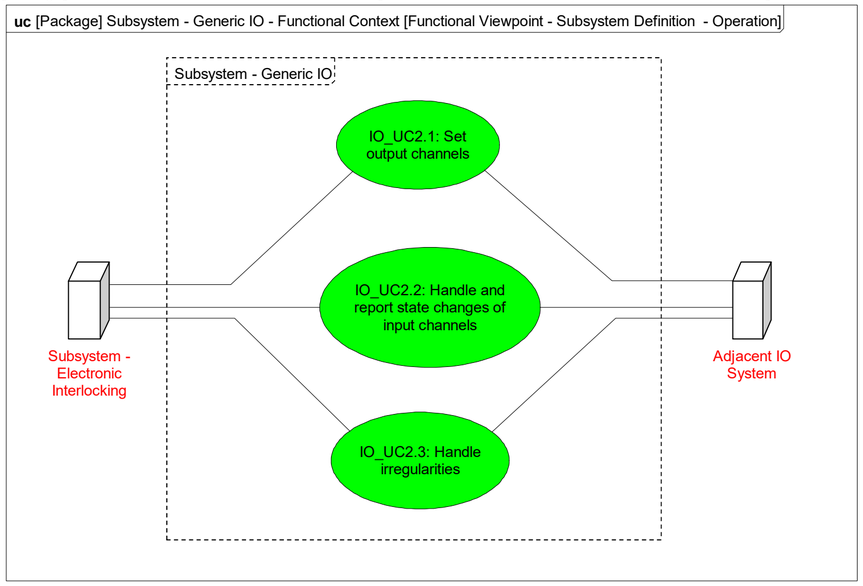
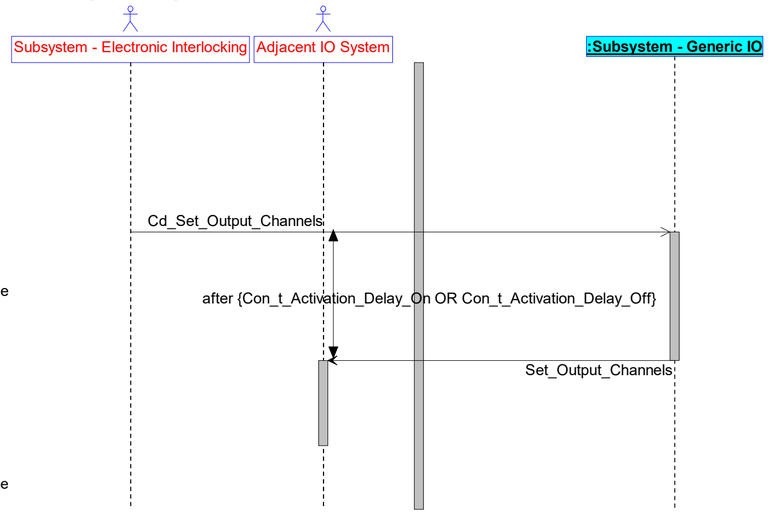
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.1643	Info	This document is drafted by and belongs to EU Rail. EU Rail encourages the distribution and re-use of this document, the technical specifications and the information it contains. EU Rail holds several intellectual property rights, such as copyright and trade mark rights, which need to be considered when this document is used. EU Rail authorizes you to re-publish, re-use, copy and store this document without changing it, provided that you indicate its source and include the following mention [EU Rail trade mark, title of the document, year of publication, version of document]. EU Rail makes no representation or warranty as to the accuracy or completeness of the information contained within these documents. EU Rail shall have no liability to any party as a result of the use of the information contained herein. EU Rail will have no liability whatsoever for any indirect or consequential loss or damage, and any such liability is expressly excluded. You may study, research, implement, adapt, improve and otherwise use the information, the content and the models in this document for your own purposes. If you decide to publish or disclose any adapted, modified or improved version of this document, any amended implementation or derivative work, then you must indicate that you have modified this document, with a reference to the document name and the terms of use of this document. You may not use EU Rail's trade marks or name in any way that may state or suggest, directly or indirectly, that EU Rail is the author of your adaptations. EU Rail cannot be held responsible for your product, even if you have used this document and its content. It is your responsibility to verify the quality, completeness and the accuracy of the information you use, for your own purposes.		
Eu.IO.10	Head	**1.3 Purpose**		
Eu.IO.11	Info	The purpose of the document is the specification of requirements for the Subsystem - Generic IO.		
Eu.IO.12	Info	This document describes functional, non-functional and technical requirements for the Subsystem - Generic IO and functional requirements for interface SCI-IO.		
Eu.IO.13	Info	This document is intended for the following users: • safety authorities • infrastructure managers • safety assessors • signalling system suppliers • validators		
Eu.IO.14	Info	This document is the basis for the implementation by the supplier and for approval by the infrastructure manager.		
Eu.IO.8059	Info	This document is applicable for both the EU-Rail System Pillar target architecture and the EULYNX architecture. The document is delivered as a single specification fitting both the System Pillar documentation sets and the EULYNX documentation sets. EU-Rail System Pillar is the technical authority for this document.		
Eu.IO.16	Head	**1.4 Applicable standards and regulations**		
Eu.IO.404	Info	A list of applicable standards and regulations used in EULYNX is listed in the EULYNX Reference Document List [Eu.Doc.12].		
Eu.IO.34	Head	**1.5 Applicable documents**		
Eu.IO.35	Info	The current versions of documents used as input or related to this document are listed in the EULYNX Documentation Plan [Eu.Doc.11]. The relationships between the documents are displayed in the Appendix A1 Documentation plan and structure [Eu.Doc.11_A1].		
Eu.IO.46	Head	**1.6 Terms and abbreviations**		
Eu.IO.47	Info	The terms and abbreviations are listed in the EULYNX Glossary [Eu.Doc.9].		
Eu.IO.703	Head	**1.7 Variability management**		
Eu.IO.704	Info	This document describes harmonised requirements. Variability management is not applicable.		
Eu.IO.1644	Head	**1.8 Definition of object types**		
Eu.IO.1645	Info	The following definition for object types is applied in this document:		
Eu.IO.1646	Info	• "Req" - This denotes a mandatory requirement.		
Eu.IO.8063	Info	• "Def" - This denotes referenceable model elements that are used in the model-based creation of requirements		
Eu.IO.1649	Info	• "Info" - This denotes additional information to help understand the specification. These objects do not specify any additional requirements.		
Eu.IO.1650	Info	• "Head" - This denotes chapter headings.		
Eu.IO.48	Head	**1.9 Modelling**		
Eu.IO.49	Info	The section "Functional requirements specification" follows a model based systems engineering process using Systems Modelling Language (SysML) and defines the functional system requirements for the Subsystem - Generic IO operational in stimulus-response form. Furthermore the information objects (stimuli and responses) exchanged over the interfaces of the Subsystem - Generic IO are defined.		
Eu.IO.50	Info	The diagrams presented in this document are modelled in SysML [SysML].		
Eu.IO.1688	Info	The rules for the interpretation of the model based parts of specification are defined in [Eu.Doc.29].		
Eu.IO.1665	Info	In chapter 3 "Functional requirements specification" the functional system requirements, defined in the form of a SysML model in the PTC Integrity Modeler are depicted as a surrogate of this model in the form of DOORS-objects.		
Eu.IO.1666	Info	A requirement thereby consists of the respective SysML model element, for instance a SysML diagram, and if necessary an additional extension of the requirement.		
Eu.IO.1667	Info	In the column "Requirement Part 1" the particular SysML model element is depicted and in the column "Requirement Part 2" the corresponding extension of the definition is given. The stated object type normally applies both to "Requirement Part 1" and to "Requirement Part 2".		
Eu.IO.1689	Info	There are requirements with type "Req" given, where the column "Requirement Part 2" or a part of it is provided with the heading "Information". In this case, the defined type only applies to the column "Requirement Part 1" and the part of "Requirement Part 2", which is not labelled as "Information".		
Eu.IO.8064	Info	State machines or several state machines linked together in a Functional Architecture define the totality of all functional requirements of an SUS or an SIUS in a coherent and consistent manner. State diagrams of a corresponding state machine are marked with the object type "Req". For the later design and implementation, it is not the description language SysML that is binding, but the domain-specific meaning expressed by it. The specified behaviour can be converted into a vendor specific language but must retain the domain specific meaning describing the functional requirements. The specific model elements are additionally specified and defined by object type "Def" to allow for traceability to supplier designs or test cases. The compliance of products to the specifications must be demonstrated by testing against EULYNX test cases, which are derived from the functionality specified by the models.		
Eu.IO.51	Head	**2 Conditions of use**		
Eu.IO.7628	Req	All references to [Eu.Doc.20] refer to version 4.0 of that document.		

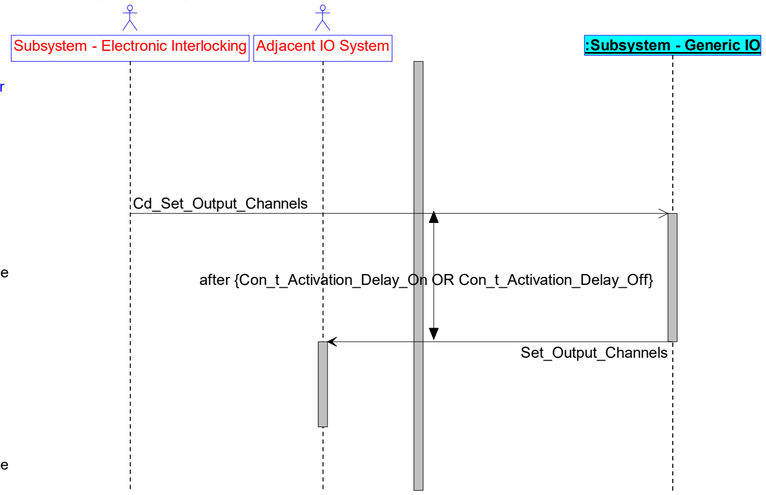
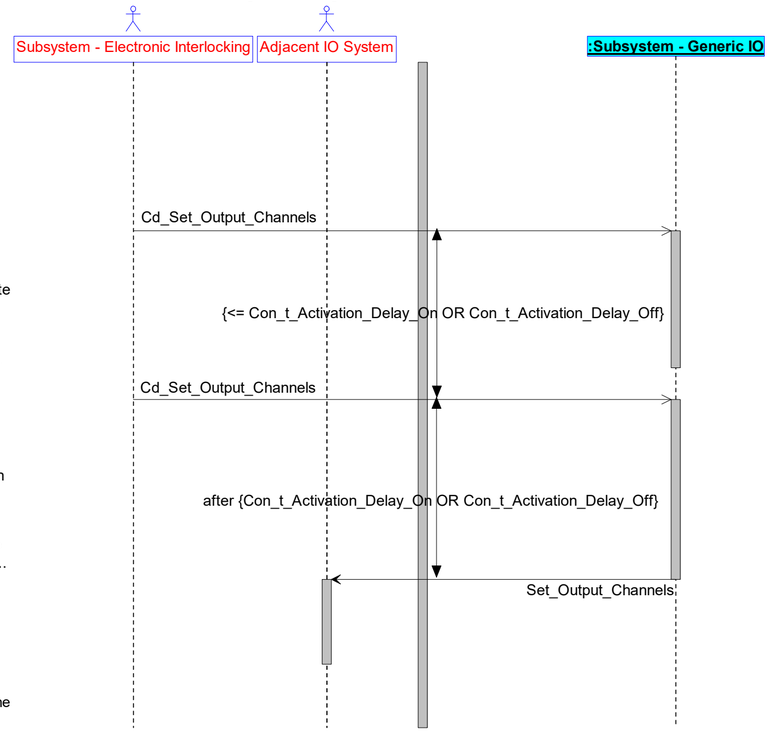
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7988	Req	All references to [Eu.Doc.119] refer to version 1.1 of that document.		
Eu.IO.7989	Info	References to [Eu.Doc.120] do not refer to a concrete version of that document. The applicable version shall be defined by national specifications. Note: In future phases of the System Pillar, national specifications will be replaced by harmonised specifications.		
Eu.IO.405	Info	The specifications defined in this document shall follow the requirements of the EULYNX System Architecture Specification [Eu.Doc.16].		
Eu.IO.7953	Head	**2.1 Functional packages**		
Eu.IO.7954	Info	The specifications in this document are divided into functional packages. There are two types of packages related to the product capabilities.		
Eu.IO.7955	Info	'Basic packages': One or more packages, at least one of them must be implemented. It is allowed to combine and implement more than one 'basic package' in a product.		
Eu.IO.7956	Info	'Optional package': One or more packages that can be optionally implemented in addition to one or more basic packages.		
Eu.IO.7957	Info	The specifications of the Subsystem – Generic IO are divided into the following functional packages:		
Eu.IO.7959	Info	Basic Generic IO functionality (basic package) [Basic IO]		
Eu.IO.7958	Info	Flashing IO output (optional package) [Option flashing]		
Eu.IO.406	Head	**3 Functional requirements specification**		
Eu.IO.407	Head	**3.1 Subsystem - Generic IO - General Infos and Assumptions**		
Eu.IO.1785	Info	Purpose of the Subsystem - Generic IO is to connect one or more elements to the Subsystem - Electronic Interlocking. These elements are identified as Adjacent IO Systems. The commands from Subsystem - Electronic Interlocking are processed and the current statuses are transmitted to the Subsystem - Electronic Interlocking, with the use of binary input and output information. The task of Subsystem - Generic IO is to ensure the controlling and reporting the status of the Adjacent IO Systems by Subsystem - Electronic Interlocking.		Basic IO
Eu.IO.1786	Info	For controlling, incoming commands from Subsystem - Electronic Interlocking are passed to the Adjacent IO System. For reporting, states of the Adjacent IO Systems are passed to the Subsystem - Electronic Interlocking. The commands and statuses are related each to each individual Adjacent IO System.		Basic IO
Eu.IO.1787	Info	For supporting the usual concept of safe transmission via two separated channels, known from relay technology, a "logical" channel can be assigned to two "physical" channels by configuration. In this case, it can be determined, if the second physical channel is supposed to behave Antivalent or Equivalent to the first physical channel. The Subsystem - Electronic Interlocking is sending commands and receiving messages related to the logical channel. If this is assigned to two physical channels (and if Antivalent or Equivalent) is not known to Subsystem - Electronic Interlocking. See EULYNX Domain Knowledge [Eu.Doc.10] for further information.		Basic IO
Eu.IO.7625	Info	The defined model elements represent the Subsystem - Generic IO in a general way. This refers to: - The functional architectures shown in the diagrams. - The defined number of functional entities which represents the connection to the Adjacent IO System. - The defined number of Input Channels and Output Channels in the state diagrams and internal block diagrams. For complete implementation the requirements from chapter "Interfaces to Adjacent IO System" (see Eu.IO.1463) shall be taken into account.		Basic IO
Eu.IO.7285	Head	**3.2 Subsystem - Generic IO - Logical Viewpoint**		
Eu.IO.7978	Head	**3.2.1 Subsystem - Generic IO - Logical Context**		

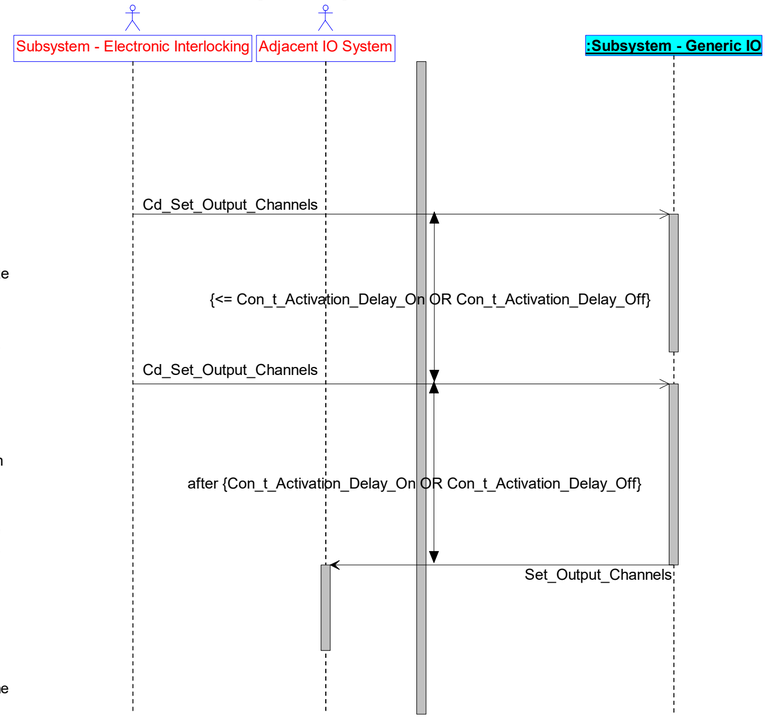
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.447	Def	[Package] Subsystem - Generic IO - Logical Context [Logical Viewpoint - Subsystem Definition]  		Basic IO
Eu.IO.8142	Req	The Subsystem - Generic IO shall provide a logical interface SCI-IO to exactly one Subsystem - Electronic Interlocking.		Basic IO
Eu.IO.8143	Req	The Subsystem - Generic IO shall provide a logical interface SMI-IO to exactly one Subsystem - Maintenance and Data Management.		Basic IO
Eu.IO.8144	Req	The Subsystem - Generic IO shall provide a logical interface SDI-IO to exactly one Subsystem - Maintenance and Data Management.		Basic IO
Eu.IO.8145	Req	The Subsystem - Generic IO shall provide a logical interface SSI-IO to exactly one Subsystem - Security Services Platform.		Basic IO
Eu.IO.8146	Req	The Subsystem - Generic IO shall provide a logical interface IO2 to each Adjacent IO System.		Basic IO
Eu.IO.8147	Req	The Subsystem - Generic IO shall provide a logical interface IO3 to each Adjacent IO System.		Basic IO
Eu.IO.8148	Req	The Subsystem - Generic IO shall provide a logical interface IO1 to exactly one Basic Data identifier.		Basic IO
Eu.IO.8149	Req	The Subsystem - Generic IO shall provide a logical interface IO5 to exactly one Maintainer.		Basic IO
Eu.IO.8150	Req	The Subsystem - Generic IO shall provide a logical interface IO4 to exactly one Power supply.		Basic IO
Eu.IO.455	Head	**3.3 Subsystem - Generic IO - Functional Viewpoint**		
Eu.IO.1279	Head	**3.3.1 Definition of time values**		
Eu.IO.1697	Info	The generic time values for SCI are specified in [Eu.Doc.119].		Basic IO
Eu.IO.7990	Info	The generic time values for SMI are specified in [Eu.Doc.120].		Basic IO
Eu.IO.1281	Def	Con_t_Activation_Delay_On	"Con_t_Activation_Delay_On" is a configurable activation delay period for each Output Channel for switching-on, for special applications.	Basic IO
Eu.IO.8038	Def	Con_t_Activation_Delay_Off	"Con_t_Activation_Delay_Off" is a configurable activation delay period for each Output Channel for switching-off, for special applications.	Basic IO
Eu.IO.1282	Def	Con_t_Message_Delay_Time_On	"Con_t_Message_Delay_Time_On" is a configurable time value for each Input Channel for report of switching-on, for special applications. The Message delay time is not relevant in case of disturbances or revocation of disturbances.	Basic IO

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.8039	Def	Con_t_Message_Delay_Time_Off	"Con_t_Message_Delay_Time_Off" is a configurable time value for each Input Channel for report of switching-off, for special applications. The Message delay time is not relevant in case of disturbances or revocation of disturbances.	Basic IO
Eu.IO.1287	Def	Con_tmax_Switching_Period	"Con_tmax_Switching_Period" is an individually configurable period for each Input Channel, which is monitored Antivalent or Equivalent. Within this period, the violation of antivalence or equivalence condition is tolerated.	Basic IO
Eu.IO.7683	Def	Con_Flash_Duty_Cycle	"Con_Flash_Duty_Cycle" is a configurable time value for each Output Channel.	Option flashing
Eu.IO.7684	Def	Con_t_Flash_Period	"Con_t_Flash_Period" is a centrally configurable time value for one Subsystem - Generic IO.	Option flashing
Eu.IO.1288	Def	Con_t_Debouncing_Time	"Con_t_Debouncing_Time" is a configurable time value followed individually for each action of both, switching-on and switching-off, of an Input Channel.	Basic IO
Eu.IO.7687	Head	**3.3.2 Subsystem - Generic IO - Functional Context**		
Eu.IO.498	Info	[Package] Subsystem - Generic IO - Functional Context [Functional Viewpoint - Subsystem Definition - Initialisation]  		Basic IO
Eu.IO.1699	Info	The generic UseCases SCI-XX EfeS IFUC1.1: Establish PDI connection and SCI-XX EfeS IFUC1.2: Close PDI connection are specified in [Eu.Doc.119]. The generic UseCases SMI-XX IFUC 1.1: Establish SMI connection, SMI-XX IFUC 1.2: Synchronous loading and activation of data, SMI-XX IFUC 1.3: Asynchronous preloading of data, SMI-XX IFUC 1.4: Reset EfeS and SMI-XX IFUC 1.5: Initiate maintenance are specified in [Eu.Doc.120].		Basic IO
Eu.IO.494	Info	IO_UC1.3: Report status	The Subsystem-UseCase "IO_UC1.3: Report status" defines a scenario about the transmission of status data of the Subsystem - Generic IO to the Subsystem - Electronic Interlocking, while Process Data Interface protocol connection is establishing.	Basic IO

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.495	Info	Main Success Scenario: Report status [IO SD 1.3.1] **IO_UC1.3: Report status** Main Success Scenario: Report status [IO SD 1.3.1] **loop** [across all Adjacent IO Systems connected to a Subsystem - Generic IO] **1.** The Subsystem - Generic IO detects the current state for each existing channel of the Adjacent IO System considered. **par** **2.a1** The Subsystem - Generic IO reports the status information of each logical Input Channel of the Adjacent IO System considered to the Subsystem - Electronic Interlocking. To this end the current state (Switched Off, Switched On or Disturbed) is transmitted for each logical Input Channel. For Antivalent or Equivalent configured RIC the value Disturbed shall be transmitted if the corresponding conditions between the RIC and VIC are violated. Moreover the value Disturbed shall be transmitted if the Subsystem - Generic IO detected a technical fault internally for a physical Input Channel. If no Input Channel has been assigned to the affected Adjacent IO System, no message Msg_State_Of_Input_Channels is sent for this Adjacent IO System to the Subsystem - Electronic Interlocking. **also par** **2.b1** The Subsystem - Generic IO reports the status information for each logical Output Channel of the Adjacent IO System considered to the Subsystem - Electronic Interlocking. To this end the current state of disturbance (Not Physically Disturbed or Physically Disturbed) is transmitted for each logical Output Channel. If the Output Channel is configured to be monitored, the value Physically Disturbed shall be transmitted if the Subsystem - Generic IO detected a technical fault internally for a physical Output Channel. If no Output Channel has been assigned to the affected Adjacent IO System, no message Msg_State_Of_Output_Channels is sent for this Adjacent IO System to the Subsystem - Electronic Interlocking. **end par** **end loop**  	The state of each logical channel is reported exactly once while establishing the PDI connection. All changes, detected afterwards, but before Msg_Initialisation_Completed are reported immediately after the completion of Initialisation. This SD is part of [SCI-XX EfeS IF SD 1.1.1] in [Eu.Doc.119].	Basic IO
Eu.IO.496	Info	IO_UC1.4: Establish initial state of outputs	The Subsystem-UseCase "IO_UC1.4: Establish initial state of outputs" defines the main success scenario for establishing the initial state of outputs of the Subsystem - Generic IO. While initialising, each physical Output Channel gets Switched Off by Subsystem - Generic IO.	Basic IO
Eu.IO.497	Info	Main Success Scenario: Establish initial state of outputs [IO SD 1.4.1] **IO_UC1.4: Establish initial state of outputs** Main Success Scenario: Establish initial state of outputs [IO SD 1.4.1] **Precondition:** The Subsystem - Generic IO is in the state BOOTING or INITIALISING. The Initial State Of Outputs has not been established. **Interaction 1.4.1.A:** **1.** - The Subsystem - Generic IO detects the readiness for establishing the Initial State Of Outputs. **2.** The Subsystem - Generic IO switches off each physical Output Channel. **Postcondition:** Each physical Output Channel is Switched Off. Initial State Of Outputs established.  		Basic IO

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.523	Info	[Package] Subsystem - Generic IO - Functional Context [Functional Viewpoint - Subsystem Definition - Operation]  		Basic IO
Eu.IO.519	Info	IO_UC2.1: Set output channels	The Subsystem-UseCase "IO_UC2.1: Set output channels" defines the process of commanding switching actions by Subsystem - Electronic Interlocking to Output Channels of Subsystem - Generic IO.	Basic IO
Eu.IO.520	Info	Alternative Scenario: Set dual-channel output channel to switched on or switched off [IO SD 2.1.1] **IO_UC2.1: Set output channels** Alternative Scenario: Set dual-channel output channel to switched on or switched off [IO SD 2.1.1] **Precondition:** The Subsystem - Generic IO is in the state OPERATIONAL. The Output Channel of interest for this scenario is configured as Equivalent OR Antivalent. **Interaction 2.1.1.A:** **1.** - The Subsystem - Generic IO receives from the Subsystem - Electronic Interlocking the switching command for each logical Output Channel of an Adjacent IO System. At least the Output Channel of interest is affected. The commanded state for the channel of interest is Switched On or Switched Off. At this Moment the Subsystem - Generic IO starts to monitor the time period "Con_t_Activation_Delay_On" OR "Con_t_Activation_Delay_Off". **2.** The Subsystem - Generic IO detects that "Con_t_Activation_Delay_On" OR "Con_t_Activation_Delay_Off" has expired and sets the new target state for each ROC and the corresponding VOC in which the current state differs from the target state. **Postcondition:** The physical Output Channels have been switched in accordance with the commanded logical state.  		Basic IO

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.522	Info	Alternative Scenario: Set single-channel output channel to switched on or switched off [IO SD 2.1.2] **IO_UC2.1: Set output channels** Alternative Scenario: Set single-channel output channel to switched on or switched off [IO SD 2.1.2] **Precondition:** The Subsystem - Generic IO is in the state OPERATIONAL. The Output Channel of interest for this scenario is configured as single. **Interaction 2.1.2.A:** **1.** - The Subsystem - Generic IO receives from the Subsystem - Electronic Interlocking the switching command for each logical Output Channel of an Adjacent IO System. At least the Output Channel of interest is affected. The commanded state for the channel of interest is Switched On or Switched Off. At this Moment the Subsystem - Generic IO starts to monitor the time period "Con_t_Activation_Delay_On" OR "Con_t_Activation_Delay_Off". **2.** The Subsystem - Generic IO detects that "Con_t_Activation_Delay_On" OR "Con_t_Activation_Delay_Off" has expired and sets the new target state for each ROC and the corresponding VOC in which the current state differs from the target state. **Postcondition:** The physical Output Channels have been switched in accordance with the commanded logical state.  		Basic IO
Eu.IO.1718	Info	Alternative Scenario: Set dual-channel output channel - quick change between switched on and switched off [IO SD 2.1.3] **IO_UC2.1: Set output channels** Alternative Scenario: Set dual-channel output channel - quick change between switched on and switched off [IO SD 2.1.3] **Precondition:** The Subsystem - Generic IO is in the state OPERATIONAL. The Output Channel of interest for this scenario is configured as Equivalent OR Antivalent. **Interaction 2.1.3.A:** **1.** - The Subsystem - Generic IO receives from the Subsystem - Electronic Interlocking the switching command for each logical Output Channel of an Adjacent IO System. At least the Output Channel of interest is affected. The commanded state for the channel of interest is Switched On or Switched Off (opposite of current state). At this moment the Subsystem - Generic IO starts to monitor the time period "Con_t_Activation_Delay_On" OR "Con_t_Activation_Delay_Off". **Interaction 2.1.3.B:** **2.** - The Subsystem - Generic IO detects that "Con_t_Activation_Delay_On" OR "Con_t_Activation_Delay_Off" has exceeded and receives from the Subsystem - Electronic Interlocking a second switching command for each logical Output Channel of an Adjacent IO System (prior to the expiry of the channel-specific activation delay of the channel of interest (start in step 1). The commanded state for the channel of interest is Switched On or Switched Off (opposite of what was commanded in step 1). At this moment the Subsystem - Generic IO restarts to monitor the time period "Con_t_Activation_Delay_On" OR "Con_t_Activation_Delay_Off".. **3.** The Subsystem - Generic IO detects that "Con_t_Activation_Delay_On" OR "Con_t_Activation_Delay_Off" has expired and sets the new target state for each ROC and the corresponding VOC in which the current state differs from the target state. No state change for the channel of interest. **Postcondition:** The physical Output Channels have been switched in accordance with the commanded logical state.  		Basic IO

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.1730	Info	Alternative Scenario: Set single-channel output channel - quick change between switched on and switched off [IO SD 2.1.4] **IO_UC2.1: Set output channels** Alternative Scenario: Set single-channel output channel - quick change between switched on and switched off [IO SD 2.1.4] **Precondition:** The Subsystem - Generic IO is in the state OPERATIONAL. The Output Channel of interest for this scenario is configured as single. **Interaction 2.1.4.A:** **1.** - The Subsystem - Generic IO receives from the Subsystem - Electronic Interlocking the switching command for each logical Output Channel of an Adjacent IO System. At least the Output Channel of interest is affected. The commanded state for the channel of interest is Switched On or Switched Off (opposite of current state). At this moment the Subsystem - Generic IO starts to monitor the time period "Con_t_Activation_Delay_On" OR "Con_t_Activation_Delay_Off". **Interaction 2.1.4.B:** **2.** - The Subsystem - Generic IO detects that "Con_t_Activation_Delay_On" OR "Con_t_Activation_Delay_Off" has exceeded and receives from the Subsystem - Electronic Interlocking a second switching command for each logical Output Channel of an Adjacent IO System (prior to the expiry of the channel-specific activation delay of the channel of interest (start in step 1). The commanded state for the channel of interest is Switched On or Switched Off (opposite of what was commanded in step 1). At this moment the Subsystem - Generic IO restarts to monitor the time period "Con_t_Activation_Delay_On" OR "Con_t_Activation_Delay_Off". **3.** The Subsystem - Generic IO detects that "Con_t_Activation_Delay_On" OR "Con_t_Activation_Delay_Off" has expired and sets the new target state for each ROC and the corresponding VOC in which the current state differs from the target state. No state change for the channel of interest. **Postcondition:** The physical Output Channels have been switched in accordance with the commanded logical state.  		Basic IO

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7688	Info	Alternative Scenario: Set dual-channel output channel to flashing [IO SD 2.1.5]		Option flashing

IO_UC2.1: Set output channels

Alternative Scenario: Set dual-channel output channel to flashing [IO SD 2.1.5]

Precondition:

The Subsystem - Generic IO is in the state OPERATIONAL.

The Output Channel of interest for this scenario is configured as Equivalent OR Antivalent.

The flashing cycles are calculated based on a channel-specific Con_Flash_Duty_Cycle and a Subsystem - Generic IO-specific Con_t_Flash_Period.

Interaction 2.1.5.A:

1. - The Subsystem - Generic IO receives from the Subsystem - Electronic Interlocking the switching command for each logical Output Channel of an Adjacent IO System. At least the Output Channel of interest is affected. The commanded state for the channel of interest is Flashing.

alt [Configured Flashing cycle is currently off]

2.a1 The Subsystem - Generic IO sets the new target state for each ROC and the corresponding VOC in which the current state differs from the target state.

loop [Required logical state is Flashing for channel of interest]

2.a2 The Subsystem - Generic IO sets the new target state for ROC of interest and the corresponding VOC and the Flashing cycle changes to on.
At this moment the Subsystem - Generic IO starts to monitor the time period "bc1_Flashing_On_Cycle".

2.a3 The Subsystem - Generic IO detects that "bc1_Flashing_On_Cycle" has expired and the Flashing cycle changes to off and the Subsystem - Generic IO sets the new target state for ROC of interest and the corresponding VOC.
At this moment the Subsystem - Generic IO starts to monitor the time period "bc2_Flashing_Off_Cycle".

end loop

else alt [Configured Flashing cycle is currently on]

2.b1 The Subsystem - Generic IO sets the new target state for each ROC and the corresponding VOC in which the current state differs from the target state.

loop [Required logical state is Flashing for channel of interest]

2.b2 The Subsystem - Generic IO sets the new target state for ROC of interest and the corresponding VOC and the Flashing cycle changes to on.
At this moment the Subsystem - Generic IO starts to monitor the time period "bc1_Flashing_On_Cycle".

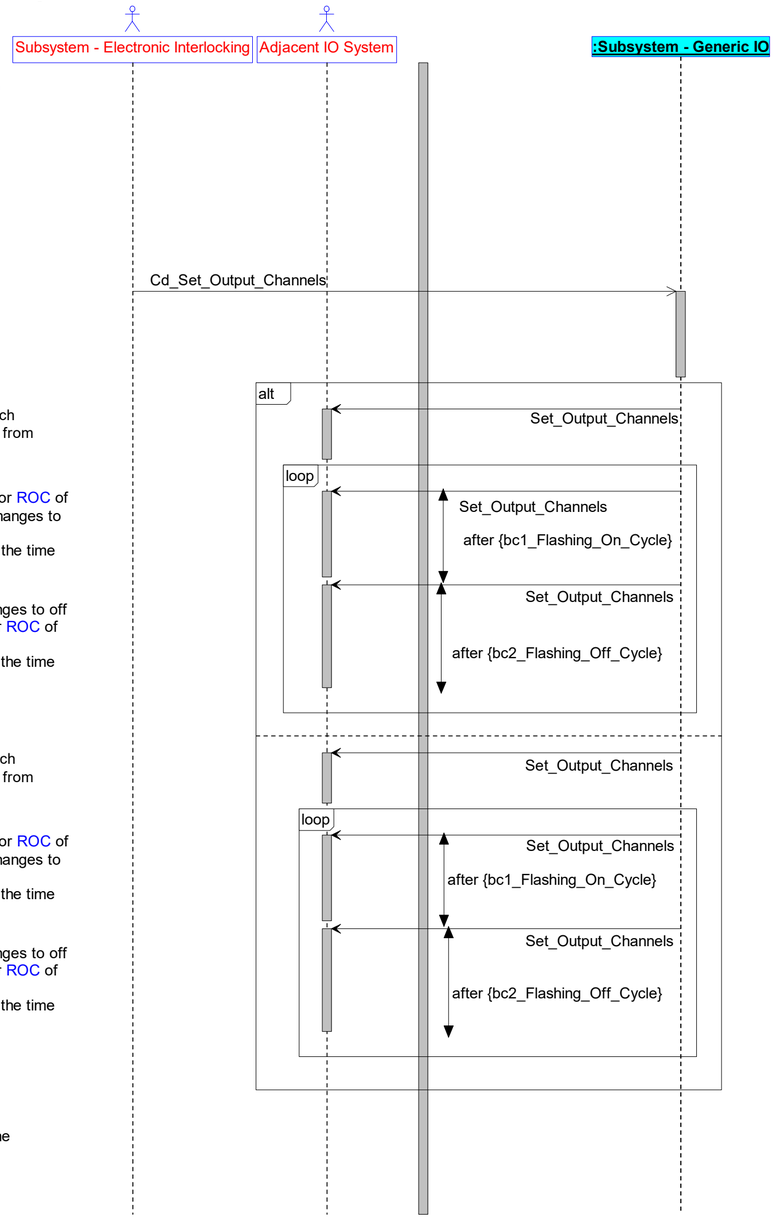
2.b3 The Subsystem - Generic IO detects that "bc1_Flashing_On_Cycle" has expired and the Flashing cycle changes to off and the Subsystem - Generic IO sets the new target state for ROC of interest and the corresponding VOC.
At this moment the Subsystem - Generic IO starts to monitor the time period "bc2_Flashing_Off_Cycle".

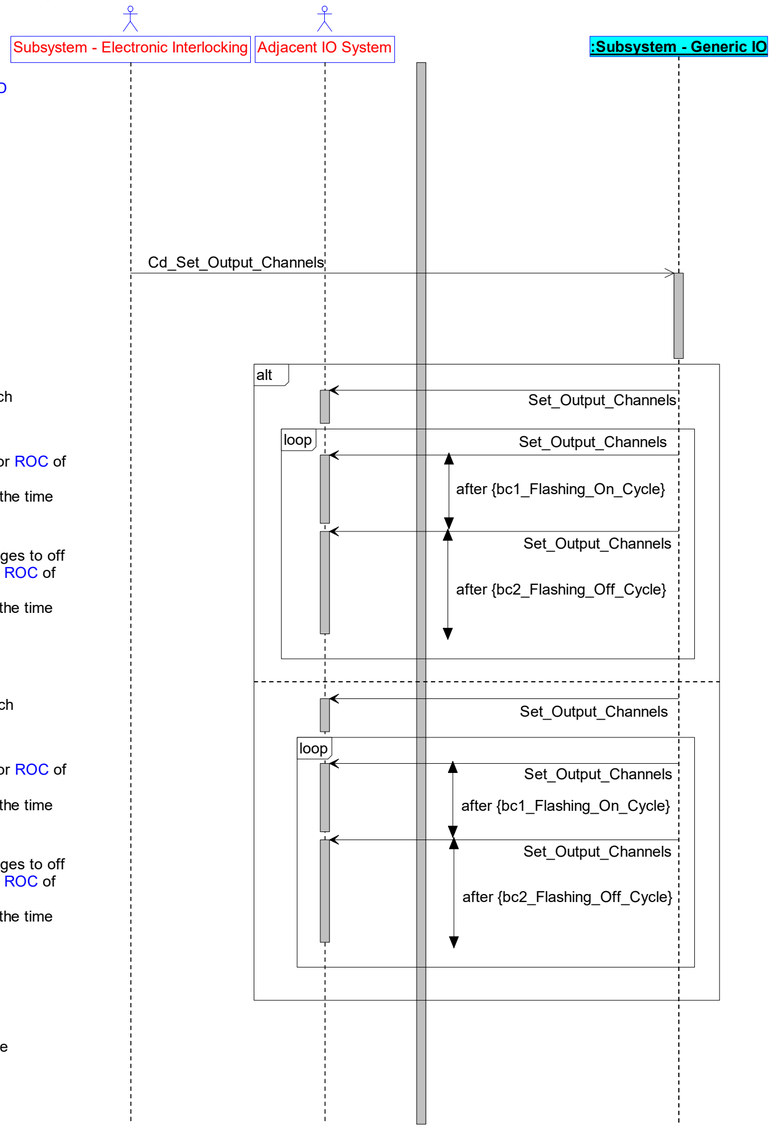
end loop

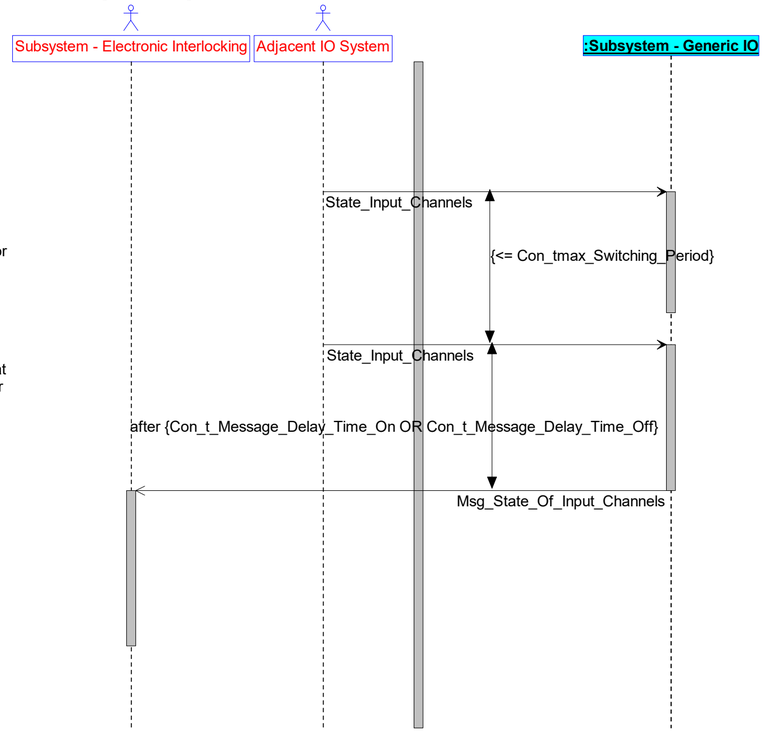
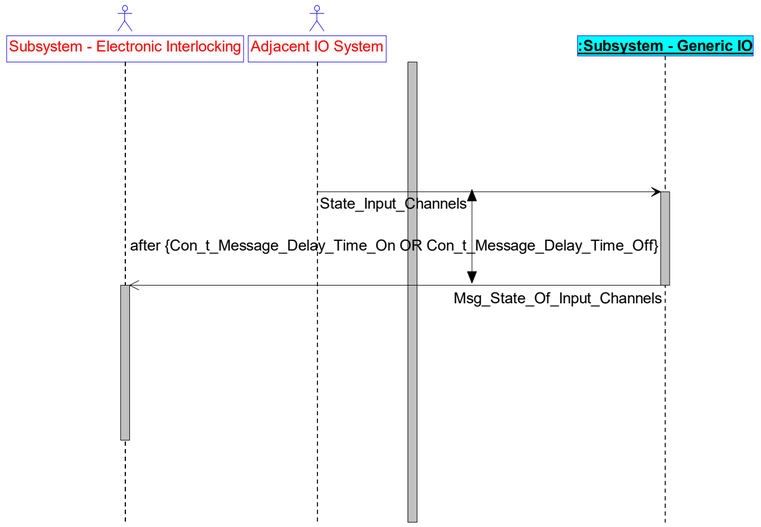
end alt

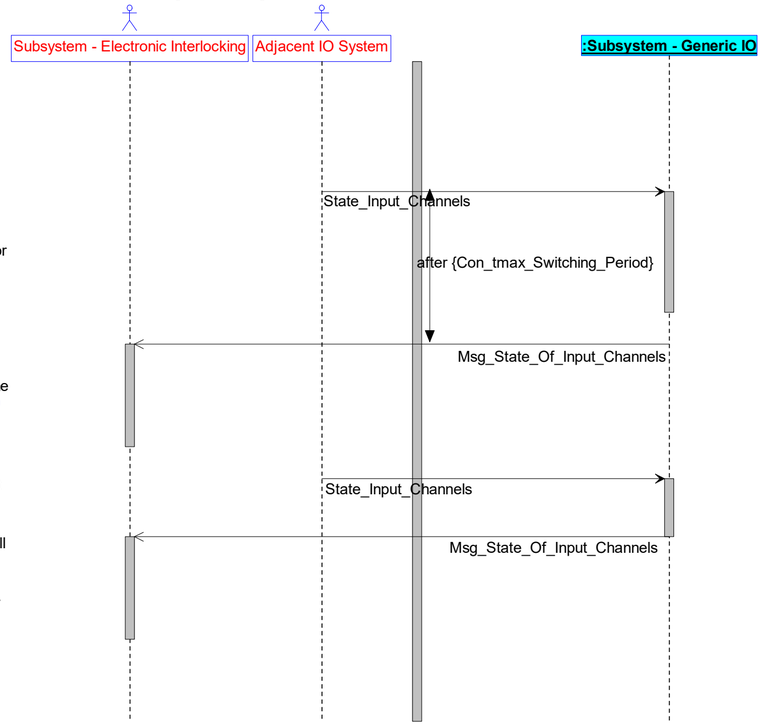
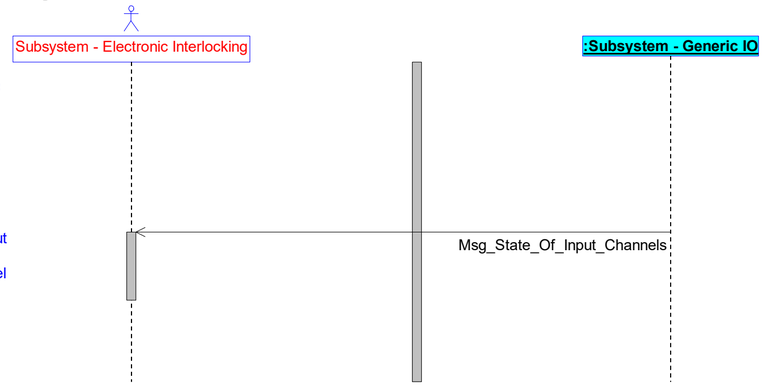
Postcondition:

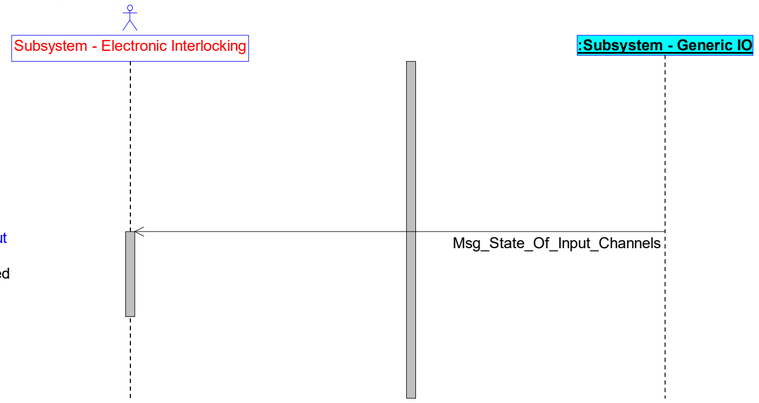
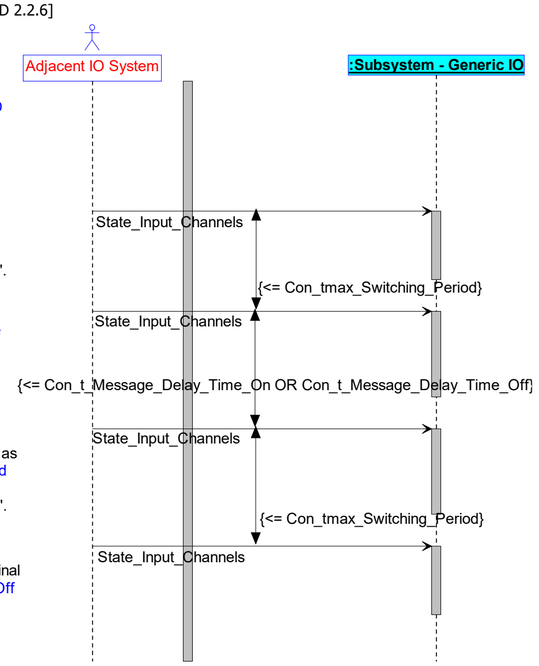
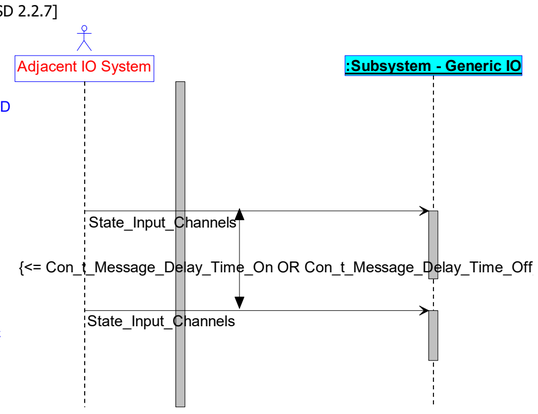
The physical Output Channels have been switched in accordance with the commanded logical state.
The channel of interest is Flashing between state Switched On and Switched Off with the channel-specific Con_Flash_Duty_Cycle and a Subsystem - Generic IO-specific Con_t_Flash_Period.

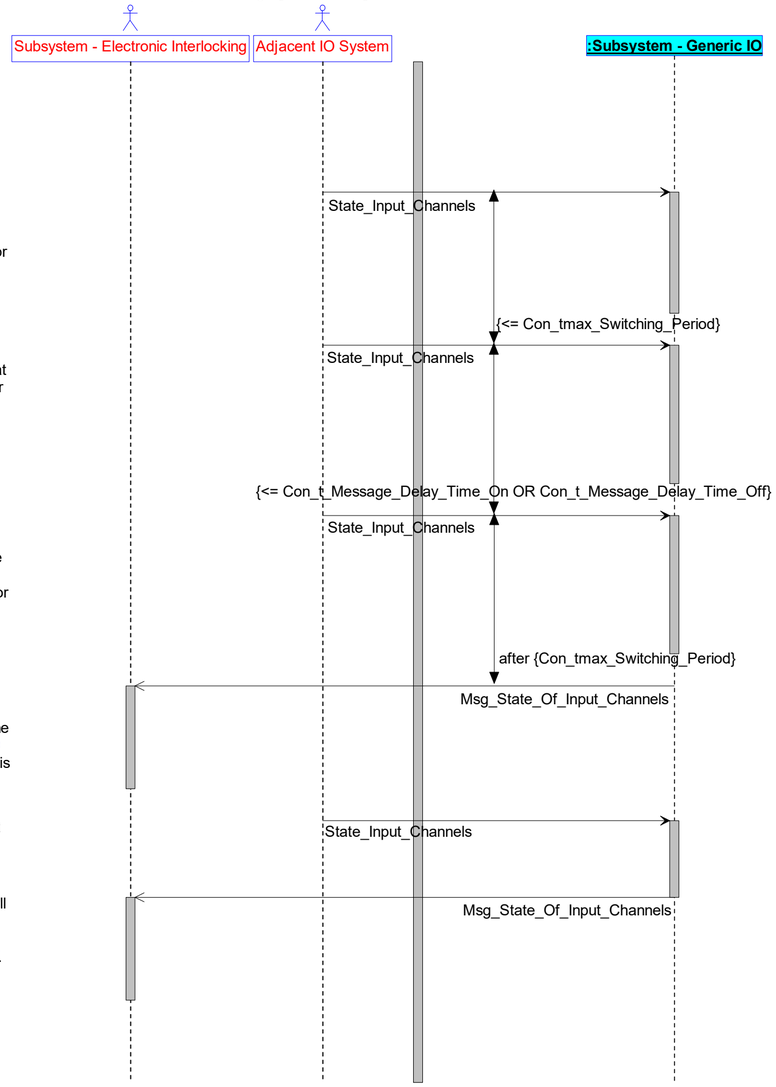


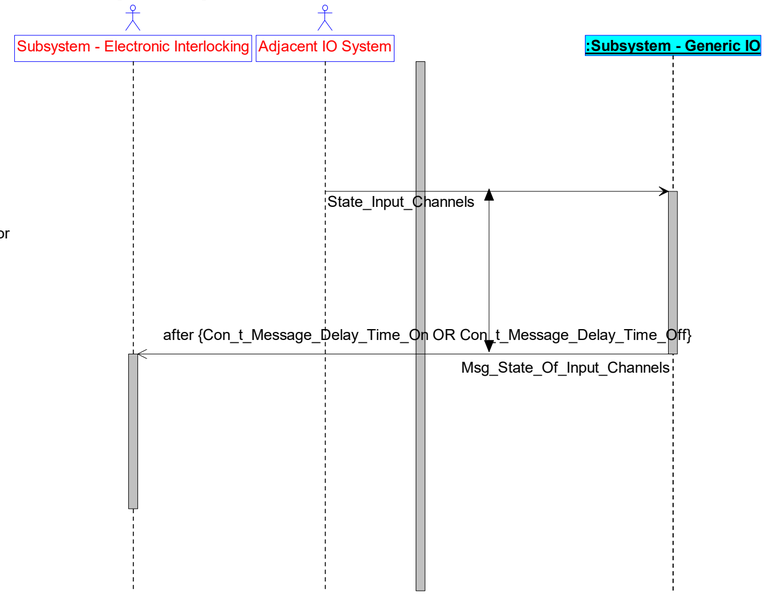
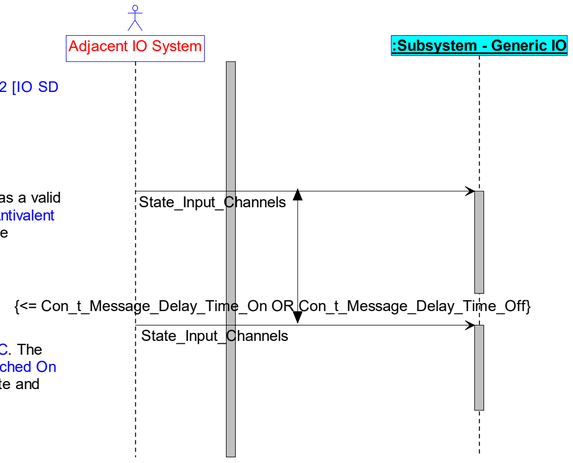
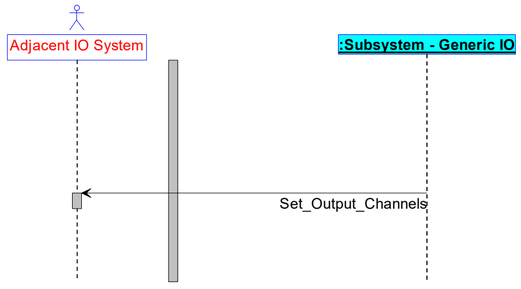
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7689	Info	Alternative Scenario: Set single-channel output channel to flashing [IO SD 2.1.6] **IO_UC2.1: Set output channels** Alternative Scenario: Set single-channel output channel to flashing [IO SD 2.1.6] **Precondition:** The Subsystem - Generic IO is in the state OPERATIONAL. The Output Channel of interest for this scenario is configured as single. The Flashing cycles are calculated based on a channel-specific Con_Flash_Duty_Cycle and a Subsystem - Generic IO-specific Con_t_Flash_Period. **Interaction 2.1.6.A:** **1.** - The Subsystem - Generic IO receives from the Subsystem - Electronic Interlocking the switching command for each logical Output Channel of an Adjacent IO System. At least the Output Channel of interest is affected. The commanded state for the channel of interest is Flashing. **alt** [Configured Flashing cycle is currently off] **2.a1** The Subsystem - Generic IO sets the new target state for each ROC in which the current state differs from the target state. **loop** [Required logical state is Flashing for channel of interest] **2.a2** The Subsystem - Generic IO sets the new target state for ROC of interest and the Flashing cycle changes to on. At this moment the Subsystem - Generic IO starts to monitor the time period "bc1_Flashing_On_Cycle". **2.a3** The Subsystem - Generic IO detects that "bc1_Flashing_On_Cycle" has expired and the Flashing cycle changes to off and the Subsystem - Generic IO sets the new target state for ROC of interest. At this moment the Subsystem - Generic IO starts to monitor the time period "bc2_Flashing_Off_Cycle". **end loop** **else alt** [Configured Flashing cycle is currently on] **2.b1** The Subsystem - Generic IO sets the new target state for each ROC in which the current state differs from the target state. **loop** [Required logical state is Flashing for channel of interest] **2.b2** The Subsystem - Generic IO sets the new target state for ROC of interest and the Flashing cycle changes to on. At this moment the Subsystem - Generic IO starts to monitor the time period "bc1_Flashing_On_Cycle". **2.b3** The Subsystem - Generic IO detects that "bc1_Flashing_On_Cycle" has expired and the Flashing cycle changes to off and the Subsystem - Generic IO sets the new target state for ROC of interest. At this moment the Subsystem - Generic IO starts to monitor the time period "bc2_Flashing_Off_Cycle". **end loop** **end alt** **Postcondition:** The physical Output Channels have been switched in accordance with the commanded logical state. The channel of interest is Flashing between state Switched On and Switched Off with the channel-specific Con_Flash_Duty_Cycle and a Subsystem - Generic IO-specific Con_t_Flash_Period. 		Option flashing
Eu.IO.511	Info	IO_UC2.2: Handle and report state changes of input channels	The Subsystem-UseCase "IO_UC2.2: Handle and report state changes of input channels" describes the process of recognising states at physical Input Channels of Adjacent IO Systems connected to Subsystem - Generic IO, as well as the report of those states to Subsystem - Electronic Interlocking. It´s distinguished, if an Input Channel is a single-channel or dual-channel (Antivalent or Equivalent). Debouncing is not described here. It´s expected, that debouncing time has expired, before the transmission of the depicted message Msg_State_Of_Input_Channels happens.	Basic IO

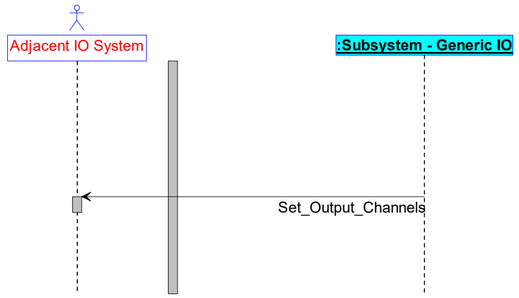
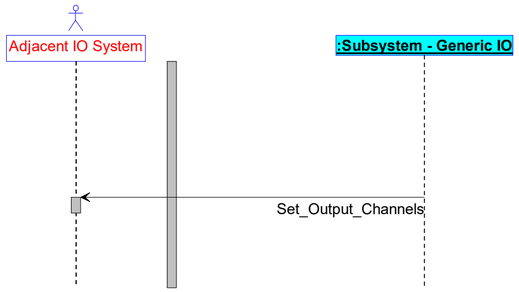
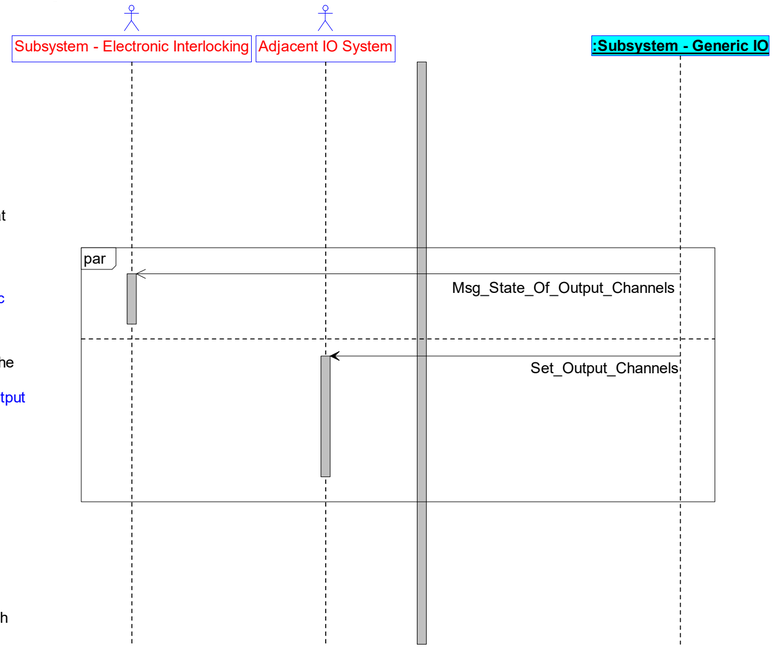
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.516	Info	Alternative Scenario: Handle and report state changes of dual-channel input channels [IO SD 2.2.1] **IO_UC2.2: Handle and report state changes of input channels** Alternative Scenario: Handle and report state changes of dual-channel input channels [IO SD 2.2.1] **Precondition:** The Subsystem - Generic IO is in the state OPERATIONAL. **Interaction 2.2.1.A:** **1.** - The Subsystem - Generic IO detects a switching action from the Adjacent IO System at the RIC (configured as dual-channel) and at the VIC as invalid switching state (e.g. RIC Switched On and VIC Switched On for Antivalent channels or RIC Switched On and VIC Switched Off for Equivalent channels). At this moment the Subsystem - Generic IO starts to monitor the time period "Con_tmax_Switching_Period". **Interaction 2.2.1.B:** **2.** - The Subsystem - Generic IO detects that permitted switching "Con_tmax_Switching_Period" has exceeded and detects a valid state at the RIC and at the VIC (e.g. RIC Switched On and VIC Switched Off for Antivalent channels or RIC Switched Off and VIC Switched Off for Equivalent channels). At this moment the Subsystem - Generic IO starts to monitor the time period "Con_t_Message_Delay_Time_On" OR "Con_t_Message_Delay_Time_Off". **3.** The Subsystem - Generic IO detects that "Con_t_Message_Delay_Time_On" OR "Con_t_Message_Delay_Time_Off" has expired and reports the new status information for each logical Input Channel of the Adjacent IO System to the Subsystem - Electronic Interlocking. The newly detected state (Switched On or Switched Off) is transmitted for the affected logical Input Channel. Note: Given the channel-specific message delay time it is possible that several telegrams are sent for one Adjacent IO System. **Postcondition:** The state of the logical Input Channels of the affected Adjacent IO System has been reported in accordance with the current state of the physical Input Channels.  		Basic IO
Eu.IO.517	Info	Alternative Scenario: Handle and report state changes of single-channel input channels [IO SD 2.2.2] **IO_UC2.2: Handle and report state changes of input channels** Alternative Scenario: Handle and report state changes of single-channel input channels [IO SD 2.2.2] **Precondition:** The Subsystem - Generic IO is in the state OPERATIONAL. **Interaction 2.2.2.A:** **1.** - The Subsystem - Generic IO detects a switching action from the Adjacent IO System at the RIC (configured as single-channel). At this moment the Subsystem - Generic IO starts to monitor the time period "Con_t_Message_Delay_Time_On" OR "Con_t_Message_Delay_Time_Off". **2.** The Subsystem - Generic IO detects that "Con_t_Message_Delay_Time_On" OR "Con_t_Message_Delay_Time_Off" has expired and reports the new status information for each logical Input Channel of the Adjacent IO System to the Subsystem - Electronic Interlocking. The newly detected state (Switched On or Switched Off) is transmitted for the affected logical Input Channel. Note: Given the channel-specific message delay time it is possible that several telegrams are sent for one Adjacent IO System. **Postcondition:** The state of the logical Input Channels of the affected Adjacent IO System has been reported in accordance with the current state of the physical Input Channels.  		Basic IO

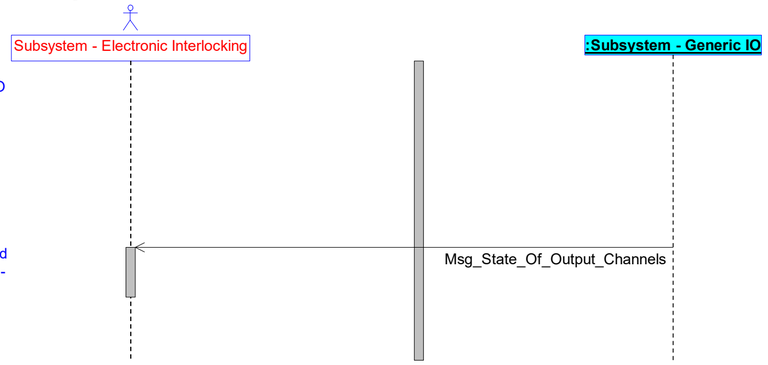
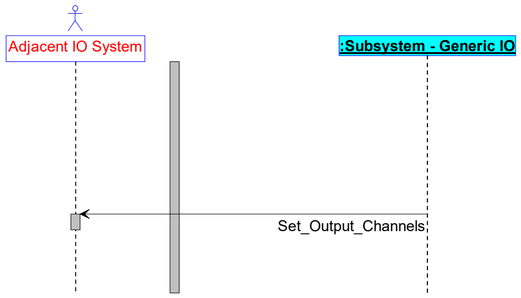
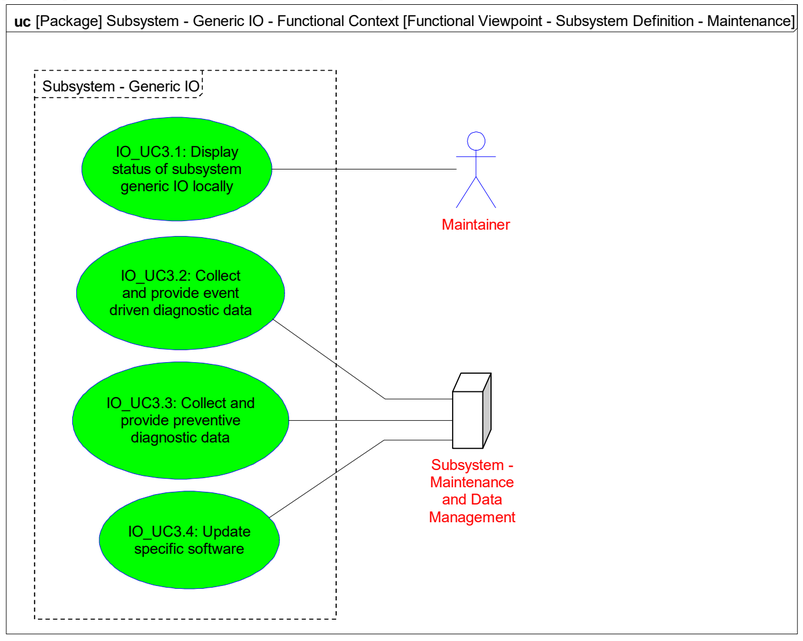
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.513	Info	Alternative Scenario: Handle and report incomplete state changes of dual-channel input channels [IO SD 2.2.3] **IO_UC2.2: Handle and report state changes of input channels** Alternative Scenario: Handle and report incomplete state changes of dual-channel input channels [IO SD 2.2.3] **Precondition:** The Subsystem - Generic IO is in the state OPERATIONAL. **Interaction 2.2.3.A:** **1.** - The Subsystem - Generic IO detects a switching action from the Adjacent IO System at the RIC (configured as dual-channel) and at the VIC as invalid switching state (e.g. RIC Switched On and VIC Switched On for Antivalent channels or RIC Switched On and VIC Switched Off for Equivalent channels). At this moment the Subsystem - Generic IO starts to monitor the time period "Con_tmax_Switching_Period". **Interaction 2.2.3.B:** **2.** - The Subsystem - Generic IO detects that permitted switching "Con_tmax_Switching_Period" has expired and did not detect a valid state at the RIC and at the VIC. The Subsystem - Generic IO reports the new status information for each logical Input Channel of the Adjacent IO System to the Subsystem - Electronic Interlocking. The state Disturbed is transmitted for the affected logical Input Channel. **Interaction 2.2.3.C:** **3.** - The Subsystem - Generic IO detects a valid state at the RIC and at the VIC (e.g. RIC Switched On and VIC Switched Off for Antivalent channels or RIC and VIC Switched Off for Equivalent channels). **4.** The Subsystem - Generic IO reports the new status information for all logical Input Channels of the Adjacent IO System to the Subsystem - Electronic Interlocking without waiting for the message delay time Con_t_Message_Delay_Time_On or Con_t_Message_Delay_Time_Off. The newly detected state (Switched On or Switched Off) is transmitted for the affected logical Input Channel. **Postcondition:** The state of the logical Input Channels of the affected Adjacent IO System has been reported in accordance with the current state of the physical Input Channels.  	Note: Some safety hazards may be related to the reporting of a valid state of a dual channel logical Input Channel after detecting an invalid state and reporting Disturbed, depending on the characteristics of the Adjacent IO System. National specifications or the supplier implementation may contain mitigations for these hazards.	Basic IO
Eu.IO.512	Info	Alternative Scenario: Handle and report disturbed input channels [IO SD 2.2.4] **IO_UC2.2: Handle and report state changes of input channels** Alternative Scenario: Handle and report disturbed input channels [IO SD 2.2.4] **Precondition:** The Subsystem - Generic IO is in the state OPERATIONAL. **Interaction 2.2.4.A:** **1.** - The Subsystem - Generic IO detects that a disturbance in the functionality is present at a physical Input Channel. **2.** The Subsystem - Generic IO reports the new states of all logical Input Channels of the Adjacent IO System to the Subsystem - Electronic Interlocking. The value Disturbed is reported for the logical Input Channel belonging to the physical Input Channel considered here. **Postcondition:** The state of the logical Input Channels of the affected Adjacent IO System has been reported in accordance with the current state of the physical Input Channels.  		Basic IO

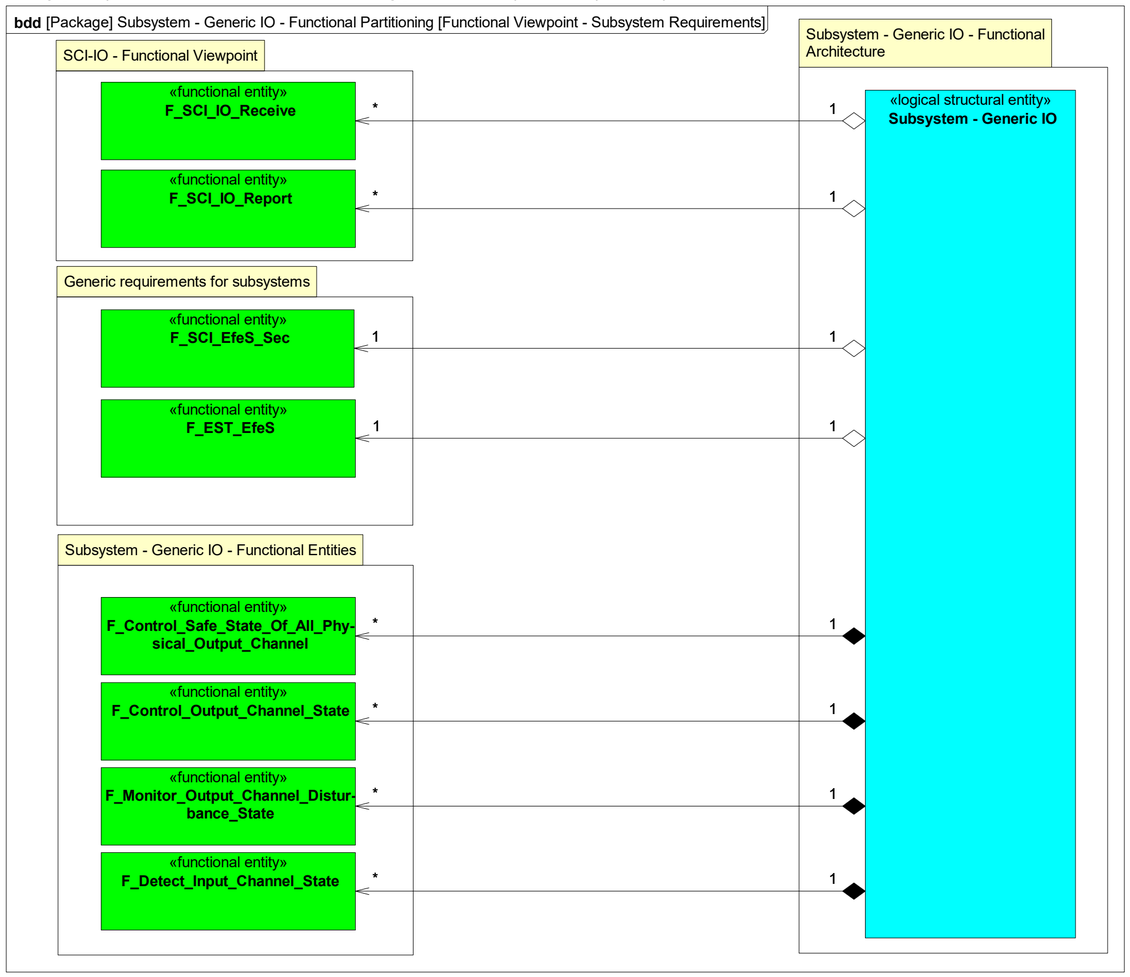
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.515	Info	Alternative Scenario: Handle and report revoked disturbances of input channels [IO SD 2.2.5] **IO_UC2.2: Handle and report state changes of input channels** Alternative Scenario: Handle and report revoked disturbances of input channels [IO SD 2.2.5] **Precondition:** The Subsystem - Generic IO is in the state OPERATIONAL. **Interaction 2.2.5.A:** **1.** - The Subsystem - Generic IO detects that a disturbance has been revoked at a physical Input Channel. **2.** The Subsystem - Generic IO reports the new states of all logical Input Channels of the Adjacent IO System to the Subsystem - Electronic Interlocking. The now valid state (Switched On, Switched Off) is reported for the logical Input Channel belonging to the physical Input Channel considered here. **Postcondition:** The state of the logical Input Channels of the affected Adjacent IO System has been reported in accordance with the current state of the physical Input Channels.  		Basic IO
Eu.IO.1758	Info	Alternative Scenario: Handle and report state changes of dual-channel input channels - quick change [IO SD 2.2.6] **IO_UC2.2: Handle and report state changes of input channels** Alternative Scenario: Handle and report state changes of dual-channel input channels - quick change [IO SD 2.2.6] **Precondition:** The Subsystem - Generic IO is in the state OPERATIONAL. **Interaction 2.2.6.A:** **1.** - The Subsystem - Generic IO detects a switching action from the Adjacent IO System at the RIC (configured as dual-channel) and at the VIC as invalid switching state (e.g. RIC Switched On and VIC Switched On for Antivalent channels or RIC Switched On and VIC Switched Off for Equivalent channels). At this moment the Subsystem - Generic IO starts to monitor the time period "Con_tmax_Switching_Period". **Interaction 2.2.6.B:** **2.** - The Subsystem - Generic IO detects that permitted switching "Con_tmax_Switching_Period" has exceeded and detects a valid state at the RIC and at the VIC (e.g. RIC Switched On and VIC Switched Off for Antivalent channels or RIC and VIC Switched On for Equivalent channels). At this moment the Subsystem - Generic IO starts to monitor the time period "Con_t_Message_Delay_Time_On" OR "Con_t_Message_Delay_Time_Off". **Interaction 2.2.6.C:** **3.** - The Subsystem - Generic IO detects that "Con_t_Message_Delay_Time_On" OR "Con_t_Message_Delay_Time_Off" has exceeded and a switching action occurs at the RIC and at the VIC as invalid switching state (e.g. RIC Switched Off and VIC Switched Off for Antivalent channels or RIC Switched Off and VIC Switched On for Equivalent channels). At this moment the Subsystem - Generic IO starts to monitor the time period "Con_tmax_Switching_Period". **Interaction 2.2.6.D:** **4.** - The Subsystem - Generic IO detects that permitted switching "Con_tmax_Switching_Period" has exceeded and detects a valid state at the RIC and at the VIC. The RIC and the VIC turned back to the original state before step 1 (e.g. RIC Switched Off and VIC Switched On for Antivalent channels or RIC Switched Off and VIC Switched Off for Equivalent channels). **Postcondition:** ---  		Basic IO
Eu.IO.1772	Info	Alternative Scenario: Handle and report state changes of single-channel input channels - quick change [IO SD 2.2.7] **IO_UC2.2: Handle and report state changes of input channels** Alternative Scenario: Handle and report state changes of single-channel input channels - quick change [IO SD 2.2.7] **Precondition:** The Subsystem - Generic IO is in the state OPERATIONAL. **Interaction 2.2.7.A:** **1.** - The Subsystem - Generic IO detects a switching action from the Adjacent IO System at the RIC (configured as single-channel) (e.g. RIC is switching from the state Switched Off to Switched On). At this moment the Subsystem - Generic IO starts to monitor the time period "Con_t_Activation_Delay_On" OR "Con_t_Activation_Delay_Off" **Interaction 2.2.7.B:** **2.** - The Subsystem - Generic IO detects that "Con_t_Activation_Delay_On" OR "Con_t_Activation_Delay_Off" has exceeded and detects the RIC turned back to the original state (e.g. RIC Switched Off). **Postcondition:** ---  		Basic IO

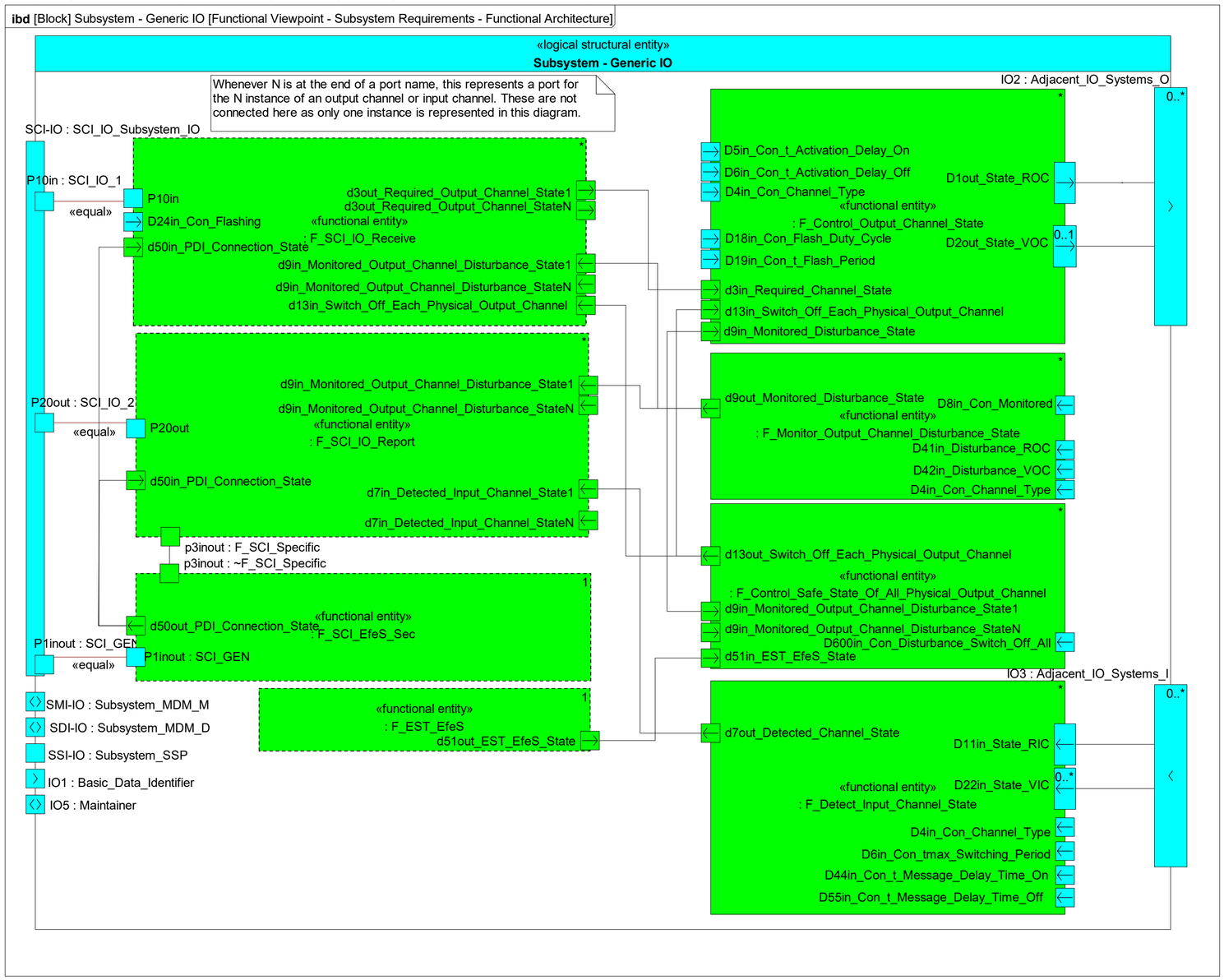
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.1741	Info	Alternative Scenario: Handle and report incomplete state changes of dual-channel input channels - quick change [IO SD 2.2.8] **IO_UC2.2: Handle and report state changes of input channels** Alternative Scenario: Handle and report incomplete state changes of dual-channel input channels - quick change [IO SD 2.2.8] **Precondition:** The Subsystem - Generic IO is in the state OPERATIONAL. **Interaction 2.2.8.A:** **1.** - The Subsystem - Generic IO detects a switching action from the Adjacent IO System at the RIC (configured as dual-channel) and at the VIC as invalid switching state (e.g. RIC Switched On and VIC Switched On for Antivalent channels or RIC Switched On and VIC Switched Off for Equivalent channels). At this moment the Subsystem - Generic IO starts to monitor the time period "Con_tmax_Switching_Period". **Interaction 2.2.8.B:** **2.** - The Subsystem - Generic IO detects that permitted switching "Con_tmax_Switching_Period" has exceeded and detects a valid state at the RIC and at the VIC (e.g. RIC Switched On and VIC Switched Off for Antivalent channels or RIC Switched On and VIC Switched On for Equivalent channels). At this moment the Subsystem - Generic IO starts to monitor the time period "Con_t_Message_Delay_Time_On" OR "Con_t_Message_Delay_Time_Off". **Interaction 2.2.8.C:** **3.** - The Subsystem - Generic IO detects that "Con_t_Message_Delay_Time_On" OR "Con_t_Message_Delay_Time_Off" has exceeded at the RIC and at the VIC as invalid switching state (e.g. RIC Switched Off and VIC Switched Off for Antivalent channels or RIC Switched Off and VIC Switched On for Equivalent channels). At this moment the Subsystem - Generic IO starts to monitor the time period "Con_tmax_Switching_Period". **Interaction 2.2.8.D:** **4.** - The Subsystem - Generic IO detects that permitted switching "Con_tmax_Switching_Period" has expired and did not detect a valid state at the RIC and at the VIC. The Subsystem - Generic IO reports the new status information for each logical Input Channel of the Adjacent IO System to the Subsystem - Electronic Interlocking. The state Disturbed is transmitted for the affected logical Input Channel. **Interaction 2.2.8.E:** **5.** - The Subsystem - Generic IO detects a valid state at the RIC and at the VIC (e.g. RIC Switched On and VIC Switched Off for Antivalent channels or RIC Switched Off and VIC Switched Off for Equivalent channels). **6.** The Subsystem - Generic IO reports the new status information for all logical Input Channels of the Adjacent IO System to the Subsystem - Electronic Interlocking without waiting for the message delay time Con_t_Message_Delay_Time_On or Con_t_Message_Delay_Time_Off. The newly detected state (Switched On or Switched Off) is transmitted for the affected logical Input Channel. **Postcondition:** The state of the logical Input Channels of the affected Adjacent IO System has been reported in accordance with the current state of the physical Input Channels.  	Note: Some safety hazards may be related to the reporting of a valid state of a dual channel logical Input Channel after detecting an invalid state and reporting Disturbed, depending on the characteristics of the Adjacent IO System. National specifications or the supplier implementation may contain mitigations for these hazards.	Basic IO

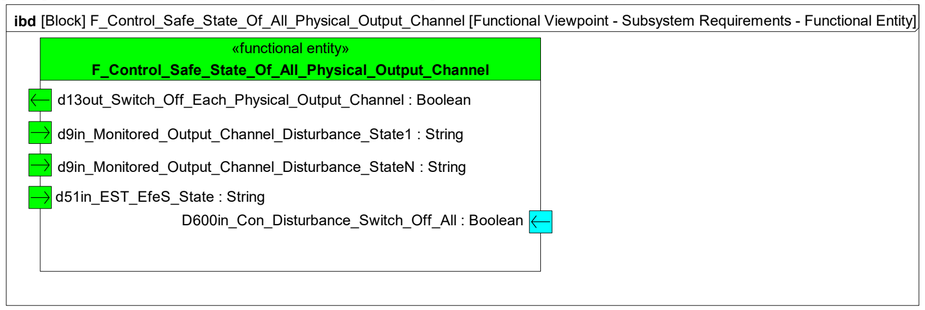
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7672	Info	Alternative Scenario: Handle and report state changes of dual-channel input channels 2 [IO SD 2.2.9] **IO_UC2.2: Handle and report state changes of input channels** Alternative Scenario: Handle and report state changes of dual-channel input channels 2 [IO SD 2.2.9] **Precondition:** The Subsystem - Generic IO is in the state OPERATIONAL. **Interaction 2.2.9.A:** **1.** - The Subsystem - Generic IO detects a switching action from the Adjacent IO System at the RIC as a valid state (configured as dual-channel) and at the VIC (e.g. RIC Switched On and VIC Switched Off for Antivalent channels or RIC Switched On and VIC Switched On for Equivalent channels) (RIC state and VIC state switched at the same time). At this moment the Subsystem - Generic IO starts to monitor the time period "Con_t_Message_Delay_Time_On" OR "Con_t_Message_Delay_Time_Off". **2.** The Subsystem - Generic IO detects that "Con_t_Message_Delay_Time_On" OR "Con_t_Message_Delay_Time_Off" has expired and reports the new status information for each logical Input Channel of the Adjacent IO System to the Subsystem - Electronic Interlocking. The newly detected state (Switched On or Switched Off) is transmitted for the affected logical Input Channel. Note: Given the channel-specific message delay time it is possible that several telegrams are sent for one Adjacent IO System. **Postcondition:** The state of the logical Input Channels of the affected Adjacent IO System has been reported in accordance with the current state of the physical Input Channels.  		Basic IO
Eu.IO.7671	Info	Alternative Scenario: Handle and report state changes of dual-channel input channels - quick change 2 [IO SD 2.2.10] **IO_UC2.2: Handle and report state changes of input channels** Alternative Scenario: Handle and report state changes of dual-channel input channels - quick change 2 [IO SD 2.2.10] **Precondition:** The Subsystem - Generic IO is in the state OPERATIONAL. **Interaction 2.2.10.A:** **1.** -The Subsystem - Generic IO detects a switching action from the Adjacent IO System at the RIC as a valid state (configured as dual-channel) and at the VIC (e.g. RIC Switched On and VIC Switched Off for Antivalent channels or RIC Switched On and VIC Switched On for Equivalent channels) (RIC state and VIC state switched at the same time). At this moment the Subsystem - Generic IO starts to monitor the time period "Con_t_Message_Delay_Time_On" OR "Con_t_Message_Delay_Time_Off". **Interaction 2.2.10.B:** **2.** - The Subsystem - Generic IO detects that "Con_t_Message_Delay_Time_On" OR "Con_t_Message_Delay_Time_Off" has exceeded and detects a valid state at the RIC and at the VIC. The RIC and the VIC turned back to the original state before step 1 (e.g. RIC Switched Off and VIC Switched On for Antivalent channels or RIC Switched Off and VIC Switched Off for Equivalent channels) (RIC state and VIC state switched at the same time). **Postcondition:** ---  		Basic IO
Eu.IO.505	Info	IO_UC2.3: Handle irregularities	The Subsystem-UseCase "IO_UC2.3: Handle irregularities" defines the behaviour of the Subsystem - Generic IO when an irregularity occurs.	Basic IO
Eu.IO.506	Info	Alternative Scenario: Execution of safety switch-off [IO SD 2.3.1] **IO_UC2.3: Handle irregularities** Alternative Scenario: Execution of safety switch-off [IO SD 2.3.1] **Precondition:** --- **Interaction 2.3.1.A:** **1.** - The Subsystem - Generic IO enters the state FALLBACK_MODE. **2.** The Subsystem - Generic IO switches off each physical Output Channel. **Postcondition:** The Subsystem - Generic IO is in the state FALLBACK_MODE. Each physical Output Channel is Switched Off.  		Basic IO

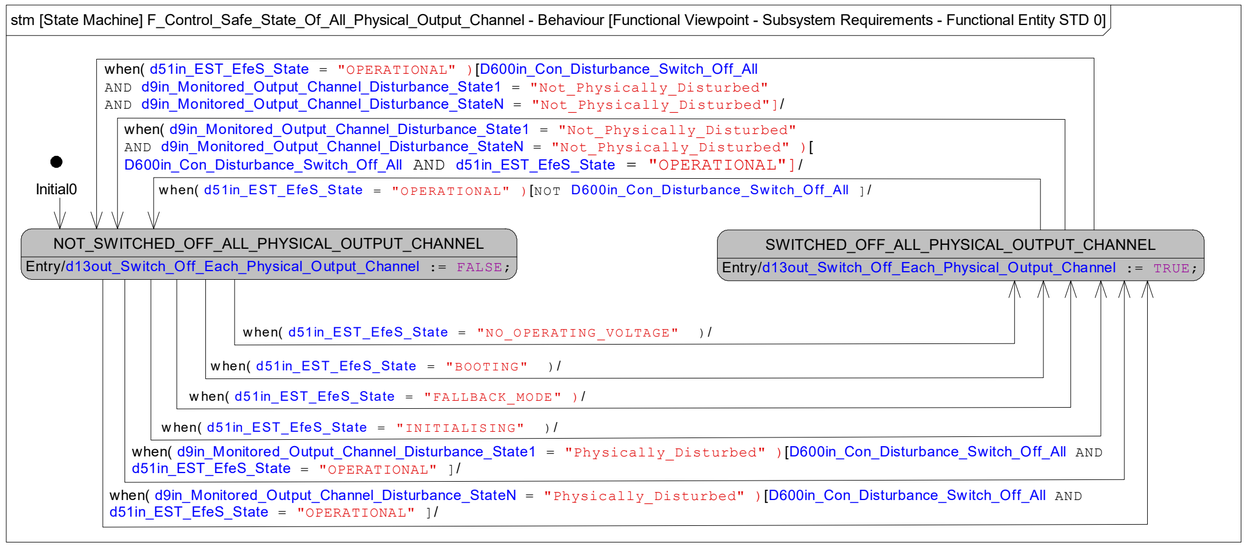
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.510	Info	Alternative Scenario: Handling of interrupted PDI connection [IO SD 2.3.2] **IO_UC2.3: Handle irregularities** Alternative Scenario: Handling of interrupted PDI connection [IO SD 2.3.2] **Precondition:** The Subsystem - Generic IO is in the state OPERATIONAL. **Interaction 2.3.2.A:** **1.** - The PDI connection has been terminated. **2.** The Subsystem - Generic IO switches off each physical Output Channel. **Postcondition:** The Subsystem - Generic IO is in the state INITIALISING. The Process Data Interface protocol connection is terminated. Each physical Output Channel is Switched Off.  	The following functionality remains available within the state INITIALISING after the termination of the PDI connection: • All functionality related to the maintainer interface IO5 (for example as in IO_UC3.1: Display status of subsystem generic IO locally)	Basic IO
Eu.IO.1788	Info	Alternative Scenario: Reset occurs [IO SD 2.3.3] **IO_UC2.3: Handle irregularities** Alternative Scenario: Reset occurs [IO SD 2.3.3] **Precondition:** The Subsystem - Generic IO is in the state INITIALISING or OPERATIONAL. **Interaction 2.3.3.A:** **1.** - A reset has been occurred. **2.** The Subsystem - Generic IO switches off each physical Output Channel. **Postcondition:** The Subsystem - Generic IO is in the state BOOTING. Each physical Output Channel is Switched Off.  		Basic IO
Eu.IO.508	Info	Alternative Scenario: Handle and report a disturbed output channel [IO SD 2.3.4] **IO_UC2.3: Handle irregularities** Alternative Scenario: Handle and report a disturbed output channel [IO SD 2.3.4] **Precondition:** The Subsystem - Generic IO is in the state OPERATIONAL. The logical Output Channel is configured to be monitored. **Interaction 2.3.4.A:** **1.** - The Subsystem - Generic IO detects that a disturbance is present at a physical Output Channel. **par** **2.a1** The Subsystem - Generic IO reports the disturbance of the corresponding logical Output Channel to the Subsystem - Electronic Interlocking. **also par** **2.b1** The Subsystem - Generic IO shall - as far as possible given the disturbance - either - switch off the physical Output Channels of the affected logical Output Channel or - all physical Output Channels of the affected Adjacent IO System (dependent on engineering data, see ID Eu.IO.629). **end par** **Postcondition:** The disturbance of the corresponding logical Output Channel has been reported to the Subsystem - Electronic Interlocking. As far as possible either the physical Output Channels of the affected logical Output Channel or all of the physical Output Channels of the affected Adjacent IO System have been Switched Off in accordance with engineering data.  		Basic IO

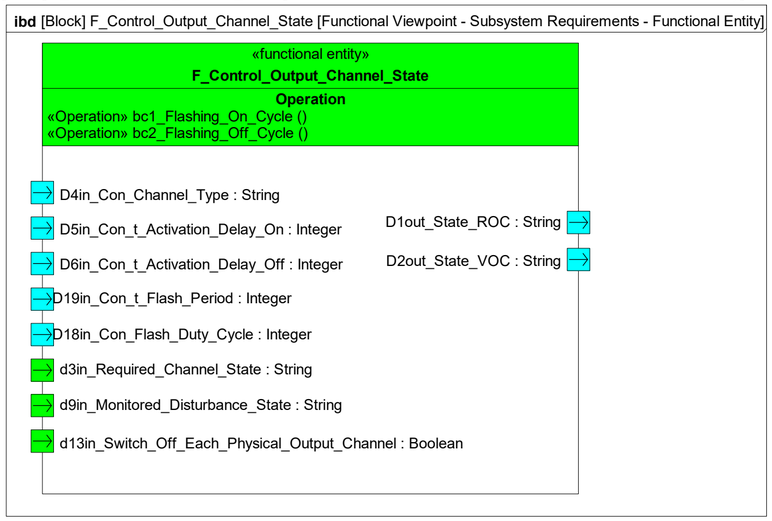
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.509	Info	Alternative Scenario: Handle revocation of a disturbed output channel [IO SD 2.3.5] **IO_UC2.3: Handle irregularities** Alternative Scenario: Handle revocation of a disturbed output channel [IO SD 2.3.5] **Precondition:** The Subsystem - Generic IO is in the state OPERATIONAL. The logical Output Channel is configured to be monitored. **Interaction 2.3.5.A:** **1.** - The Subsystem - Generic IO detects that a disturbance has been revoked at a physical Output Channel. **2.** The Subsystem - Generic IO reports, that it is not physically Disturbed anymore at the corresponding logical Output Channel to the Subsystem - Electronic Interlocking. **Postcondition:** The revocation of the disturbance of the corresponding logical Output Channel has been reported to the Subsystem - Electronic Interlocking.  		Basic IO
Eu.IO.1611	Info	Alternative Scenario: Supply voltage of the subsystem has gone outside the required range for operation [IO SD 2.3.6] **IO_UC2.3: Handle irregularities** Alternative Scenario: Supply voltage of the subsystem has gone outside the required range for operation [IO SD 2.3.6] **Precondition:** --- **Interaction 2.3.6.A:** **1.** - The Subsystem - Generic IO enters the state NO_OPERATING_VOLTAGE. **2.** The Subsystem - Generic IO ensures, that each physical Output Channel is Switched Off. **Postcondition:** The Subsystem - Generic IO is in the state NO_OPERATING_VOLTAGE. All physical Output Channels of Subsystem - Generic IO are Switched Off.  		Basic IO
Eu.IO.503	Info	[Package] Subsystem - Generic IO - Functional Context [Functional Viewpoint - Subsystem Definition - Maintenance]  		Basic IO
Eu.IO.500	Info	IO_UC3.1: Display status of subsystem generic IO locally	Information: The Subsystem-UseCase "IO_UC3.1: Display status of subsystem generic IO locally" defines the local display of the EULYNX field element Subsystem. See ID Eu.IO.416.	Basic IO

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.501	Info	IO_UC3.2: Collect and provide event driven diagnostic data	Information: The Subsystem-UseCase "IO_UC3.2: Collect and provide event driven diagnostic data" defines the event driven collection and provision of diagnostic data in case of irregularities. See ID Eu.IO.433.	Basic IO
Eu.IO.502	Info	IO_UC3.3: Collect and provide preventive diagnostic data	Information: The Subsystem-UseCase "IO_UC3.3: Collect and provide preventive diagnostic data" defines the continuous collection and provision of diagnostic data for preventive maintenance. See ID Eu.IO.433.	Basic IO
Eu.IO.1374	Info	IO_UC3.4: Update specific software	Information: The Subsystem-UseCase "IO_UC3.4: Update specific software" defines the process of updating the specific software between Subsystem - Maintenance and Data Management and the Subsystem.	Basic IO
Eu.IO.7972	Head	**3.3.3 Subsystem - Generic IO - Functional Partitioning**		
Eu.IO.7291	Def	[Package] Subsystem - Generic IO - Functional Partitioning [Functional Viewpoint - Subsystem Requirements]  		Basic IO
Eu.IO.7685	Head	**3.3.4 Subsystem - Generic IO - Functional Architecture**		
Eu.IO.437	Info	Subsystem - Generic IO	The Subsystem - Generic IO is used for integrating signalling components, particularly in the track and platform area, which are controlled or monitored with input and output information.	Basic IO

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.445	Def	[Block] Subsystem - Generic IO [Functional Viewpoint - Subsystem Requirements - Functional Architecture]  	The Subsystem - Generic IO shall provide the functional interfaces as shown in "[Block] Subsystem - Generic IO [Functional Viewpoint - Subsystem Requirements - Functional Architecture]", typed by InformationFlows. Each InformationFlows is defined by a set of FlowProperties that specify the possible exchange of information through the particular interface.	Basic IO
Eu.IO.7686	Def	SCI-IO	The functional Process Data interface to the Subsystem - Electronic Interlocking (SCI: Standard Communication Interface). The InformationFlow through the interface is further defined in SCI-IO (Subsystem - Electronic Interlocking).	Basic IO
Eu.IO.444	Def	SMI-IO	The functional Maintenance Interface to the Subsystem - Maintenance and Data Management for the InformationFlow through the interface, which is defined by "Subsystem_MDM_M".	Basic IO
Eu.IO.443	Def	SDI-IO	The functional Diagnostic interface to the Subsystem - Maintenance and Data Management for the InformationFlow through the interface, which is defined by "Subsystem_MDM_D".	Basic IO

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7987	Def	SSI-IO	The Security Service Interface to the Subsystem - Security Services Platform. The InformationFlow through the interface is further defined in SSI-IO (Subsystem - Security Services Platform).	Basic IO
Eu.IO.438	Def	IO1	The functional System Data interface to the Basic Data identifier. The InformationFlow through the interface is defined by "Basic_Data_Identifier".	Basic IO
Eu.IO.1268	Def	IO5	The functional Local Control and Display interface to the Maintainer. The InformationFlow through the interface is defined by "Maintainer".	Basic IO
Eu.IO.439	Def	IO2	The functional Control interface for the Output Channels (Reference Output Channel ROC and possibly Validation Output Channel VOC) to the Adjacent IO System. The InformationFlow through the interface is defined by "Adjacent_IO_Systems_O".	Basic IO
Eu.IO.440	Def	IO3	The functional Control interface for the Input Channels (Reference Input Channel RIC and possibly Validation Output Channel VIC) to the Adjacent IO System. The InformationFlow through the interface is defined by "Adjacent_IO_Systems_I".	Basic IO
Eu.IO.7690	Head	**3.3.5 Subsystem - Generic IO - Functional Entities**		
Eu.IO.1698	Info	The abstract essential subsystem states F_EST_EfeS of the Subsystem - Generic IO are specified in [Eu.Doc.20].		Basic IO
Eu.IO.7737	Info	F_Control_Safe_State_Of_All_Physical_Output_Channel		Basic IO
Eu.IO.7738	Req	[Block] F_Control_Safe_State_Of_All_Physical_Output_Channel [Functional Viewpoint - Subsystem Requirements - Functional Entity]  		Basic IO
Eu.IO.7740	Def	d51in_EST_EfeS_State	The port d51in_EST_EfeS_State provides information about the state of the generic state machine.	Basic IO
Eu.IO.7741	Def	D600in_Con_Disturbance_Switch_Off_All	The port D600in_Con_Disturbance_Switch_Off_All provides configuration values for the behaviour in case of disturbance depending on engineering data, see ID Eu.IO.629.	Basic IO
Eu.IO.7742	Def	d9in_Monitored_Output_Channel_Disturbance_State1	The port d9in_Monitored_Output_Channel_Disturbance_State1 provides the monitored state of Output Channel 1. The following values are permitted: - Physically Disturbed - Not Physically Disturbed	Basic IO
Eu.IO.7743	Def	d9in_Monitored_Output_Channel_Disturbance_StateN	The port d9in_Monitored_Output_Channel_Disturbance_StateN provides the monitored state of Output Channel N. The following values are permitted: - Physically Disturbed - Not Physically Disturbed	Basic IO
Eu.IO.7739	Def	d13out_Switch_Off_Each_Physical_Output_Channel		Basic IO
Eu.IO.7744	Info	F_Control_Safe_State_Of_All_Physical_Output_Channel - Behaviour		Basic IO

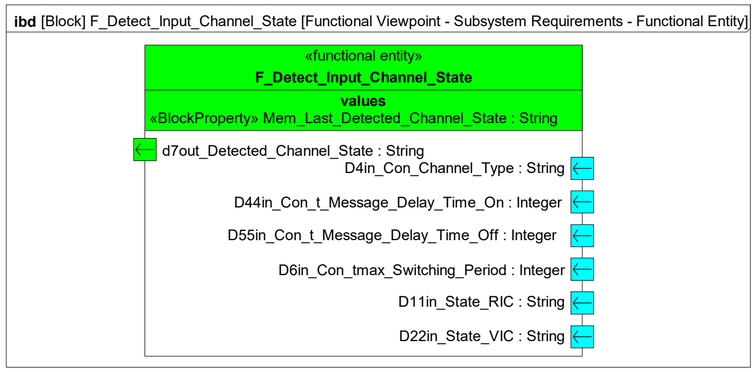
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7745	Req	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 0  	This state machine diagram describes the requirements for the following functionalities: - observe the status of the physical output channel	Basic IO
Eu.IO.7746	Def	Initial0		Basic IO
Eu.IO.7747	Def	/{Initial0 - NOT_SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL}		Basic IO
Eu.IO.7748	Def	NOT_SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL		Basic IO
Eu.IO.7749	Def	entry/d13out_Switch_Off_Each_Physical_Output_Channel := FALSE;{State-internal in NOT_SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL}		Basic IO
Eu.IO.7753	Def	when(d51in_EST_EfeS_State = "NO_OPERATING_VOLTAGE")/{NOT_SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL - SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL}		Basic IO
Eu.IO.7750	Def	when(d51in_EST_EfeS_State = "BOOTING")/{NOT_SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL - SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL}		Basic IO
Eu.IO.7752	Def	when(d51in_EST_EfeS_State = "INITIALISING")/{NOT_SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL - SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL}		Basic IO
Eu.IO.7751	Def	when(d51in_EST_EfeS_State = "FALLBACK_MODE")/{NOT_SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL - SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL}		Basic IO
Eu.IO.7754	Def	when(d9in_Monitored_Output_Channel_Disturbance_State1 = "Physically_Disturbed")[D600in_Con_Disturbance_Switch_Off_All AND d51in_EST_EfeS_State = "OPERATIONAL"]/{NOT_SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL - SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL}		Basic IO
Eu.IO.7755	Def	when(d9in_Monitored_Output_Channel_Disturbance_StateN = "Physically_Disturbed")[D600in_Con_Disturbance_Switch_Off_All AND d51in_EST_EfeS_State = "OPERATIONAL"]/{NOT_SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL - SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL}		Basic IO
Eu.IO.7756	Def	SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL		Basic IO
Eu.IO.7757	Def	entry/d13out_Switch_Off_Each_Physical_Output_Channel := TRUE;{State-internal in SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL}		Basic IO
Eu.IO.7758	Def	when(d51in_EST_EfeS_State = "OPERATIONAL")[D600in_Con_Disturbance_Switch_Off_All AND d9in_Monitored_Output_Channel_Disturbance_State1 = "Not_Physically_Disturbed" AND d9in_Monitored_Output_Channel_Disturbance_StateN = "Not_Physically_Disturbed"]/{SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL - NOT_SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL}		Basic IO
Eu.IO.7759	Def	when(d51in_EST_EfeS_State = "OPERATIONAL")[NOT D600in_Con_Disturbance_Switch_Off_All]/{SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL - NOT_SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL}		Basic IO
Eu.IO.7760	Def	when(d9in_Monitored_Output_Channel_Disturbance_State1 = "Not_Physically_Disturbed" AND d9in_Monitored_Output_Channel_Disturbance_StateN = "Not_Physically_Disturbed")[D600in_Con_Disturbance_Switch_Off_All AND d51in_EST_EfeS_State = "OPERATIONAL"]/{SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL - NOT_SWITCHED_OFF_ALL_PHYSICAL_OUTPUT_CHANNEL}		Basic IO
Eu.IO.7399	Info	F_Control_Output_Channel_State		Basic IO

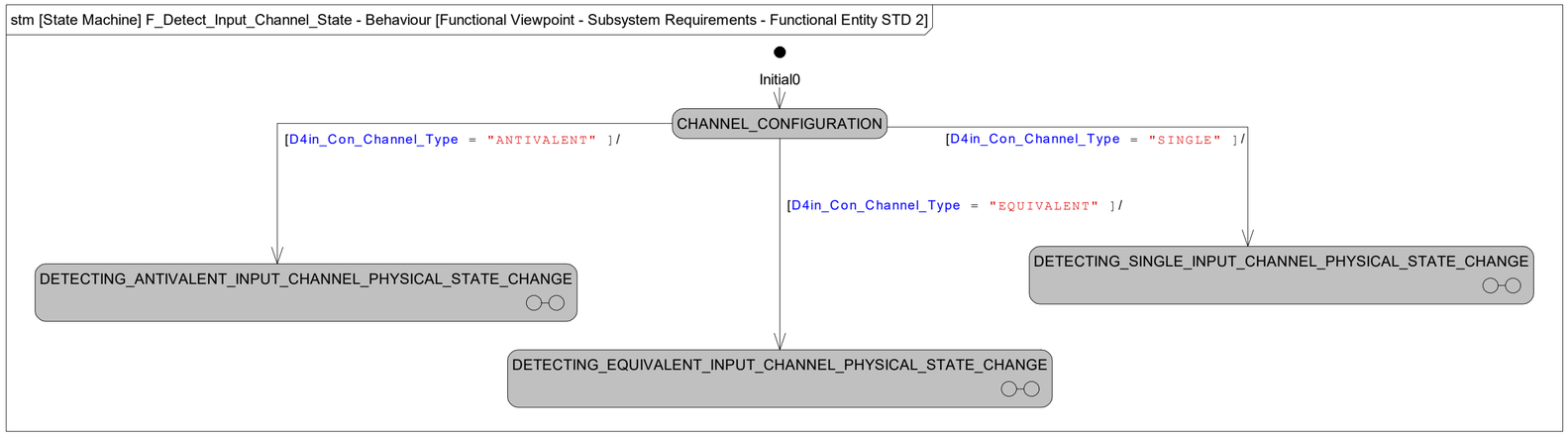
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7482	Req	[Block] F_Control_Output_Channel_State [Functional Viewpoint - Subsystem Requirements - Functional Entity]  		Basic IO
Eu.IO.7400	Def	//Flashing cycle changing to on	bc1_Flashing_On_Cycle	Option flashing
Eu.IO.7401	Def	//Flashing cycle changing to off	bc2_Flashing_Off_Cycle	Option flashing
Eu.IO.7406	Def	D1out_State_ROC	The port D1out_State_ROC refines the FlowProperty Set_Output_Channels and represents the ROC with the permanent available information at the interface IO2. A state change of D1out_State_ROC is always a try to switch the Adjacent IO System which is connected behind the interface IO2. If D41in_Disturbance_ROC is true the Adjacent IO System do not switched to the permanent available target state from D1out_State_ROC. The following values are permitted: - Switched On - Switched Off	Basic IO
Eu.IO.7407	Def	D2out_State_VOC	The port D2out_State_VOC refines the FlowProperty Set_Output_Channels and represents the VOC with the permanent available information at the interface IO2. A state change of D2out_State_VOC is always a try to switch the Adjacent IO System which is connected behind the interface IO2. If D42in_Disturbance_VOC is true the Adjacent IO System do not switched to the permanent available target state from D2out_State_VOC. The following values are permitted: - Switched On - Switched Off	Basic IO
Eu.IO.7413	Def	d3in_Required_Channel_State		Basic IO
Eu.IO.7409	Def	D4in_Con_Channel_Type	The port D4in_Con_Channel_Type provides configuration values for the implementation of the logical Output Channel. The following values are permitted: - Single (assigned to one physical channel) - Antivalent (assigned to two physical channels) - Equivalent (assigned to two physical channels)	Basic IO
Eu.IO.7411	Def	D5in_Con_t_Activation_Delay_On	The port D5in_Con_t_Activation_Delay_On represents the time value for Con_t_Activation_Delay_On. This is the delay for switching-on the Output Channel.	Basic IO
Eu.IO.7412	Def	D6in_Con_t_Activation_Delay_Off	The port D6in_Con_t_Activation_Delay_Off represents the time value for Con_t_Activation_Delay_Off. This is the delay for switching-off the Output Channel.	Basic IO
Eu.IO.7484	Def	d13in_Switch_Off_Each_Physical_Output_Channel		Basic IO
Eu.IO.7694	Def	D19in_Con_t_Flash_Period	The port D19in_Con_t_Flash_Period represents the time value for Con_t_Flash_Period.	Option flashing
Eu.IO.7693	Def	D18in_Con_Flash_Duty_Cycle	The port D18in_Con_Flash_Duty_Cycle represents the value for Con_Flash_Duty_Cycle.	Option flashing

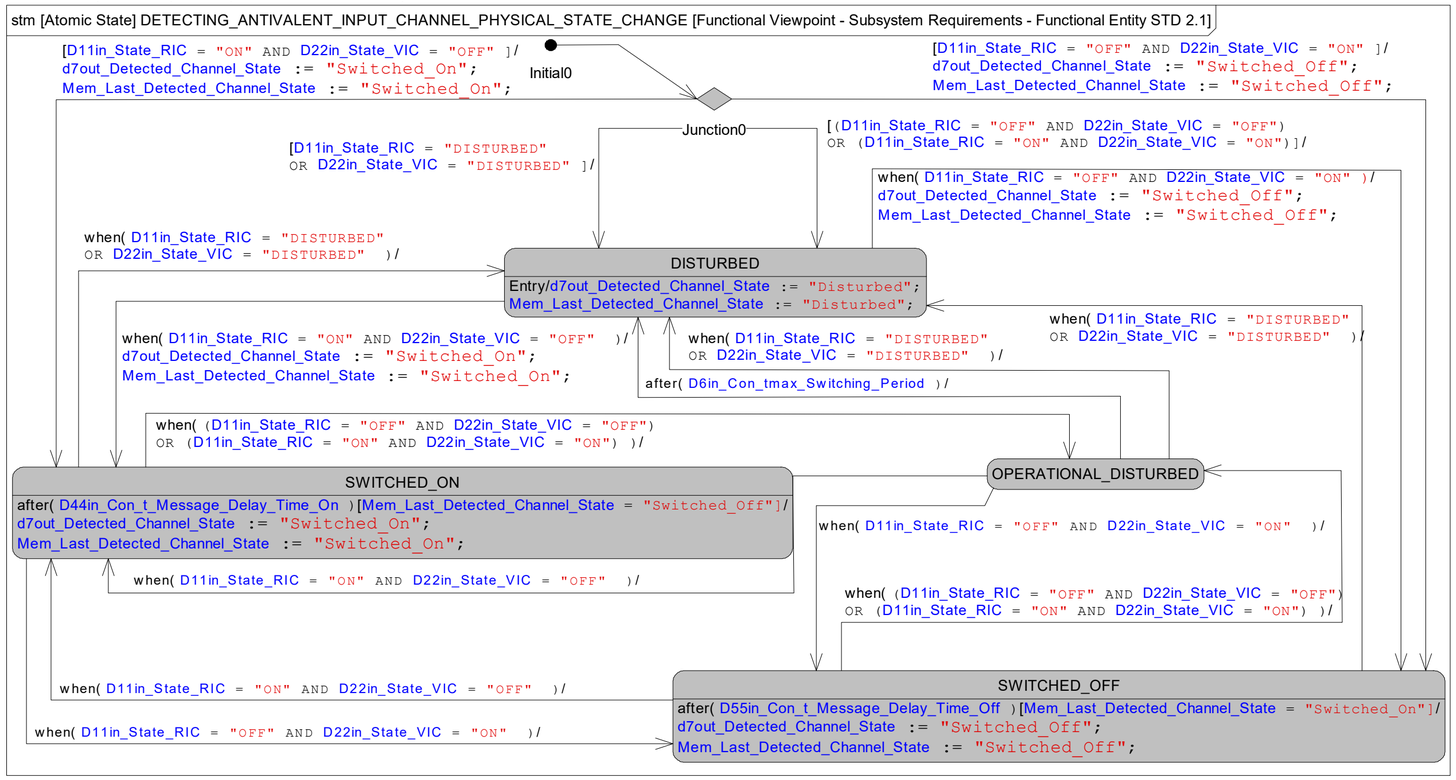
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7695	Def	d9in_Monitored_Disturbance_State	The port d9in_Monitored_Disturbance_State provides the monitored state of one Output Channel. The following values are permitted: - Physically Disturbed - Not Physically Disturbed	Basic IO
Eu.IO.7415	Info	F_Control_Output_Channel_State - Behaviour		Basic IO
Eu.IO.7418	Req	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 1 	This state machine diagram describes the requirements for the following functionalities: - observe the status of the flashing cycle - monitors the Switching of VOC	Basic IO
Eu.IO.7416	Def	Initial0		Basic IO
Eu.IO.7417	Def	/{Initial0 - SETTING_PHYSICAL_OUTPUT_CHANNEL_STATES}		Basic IO

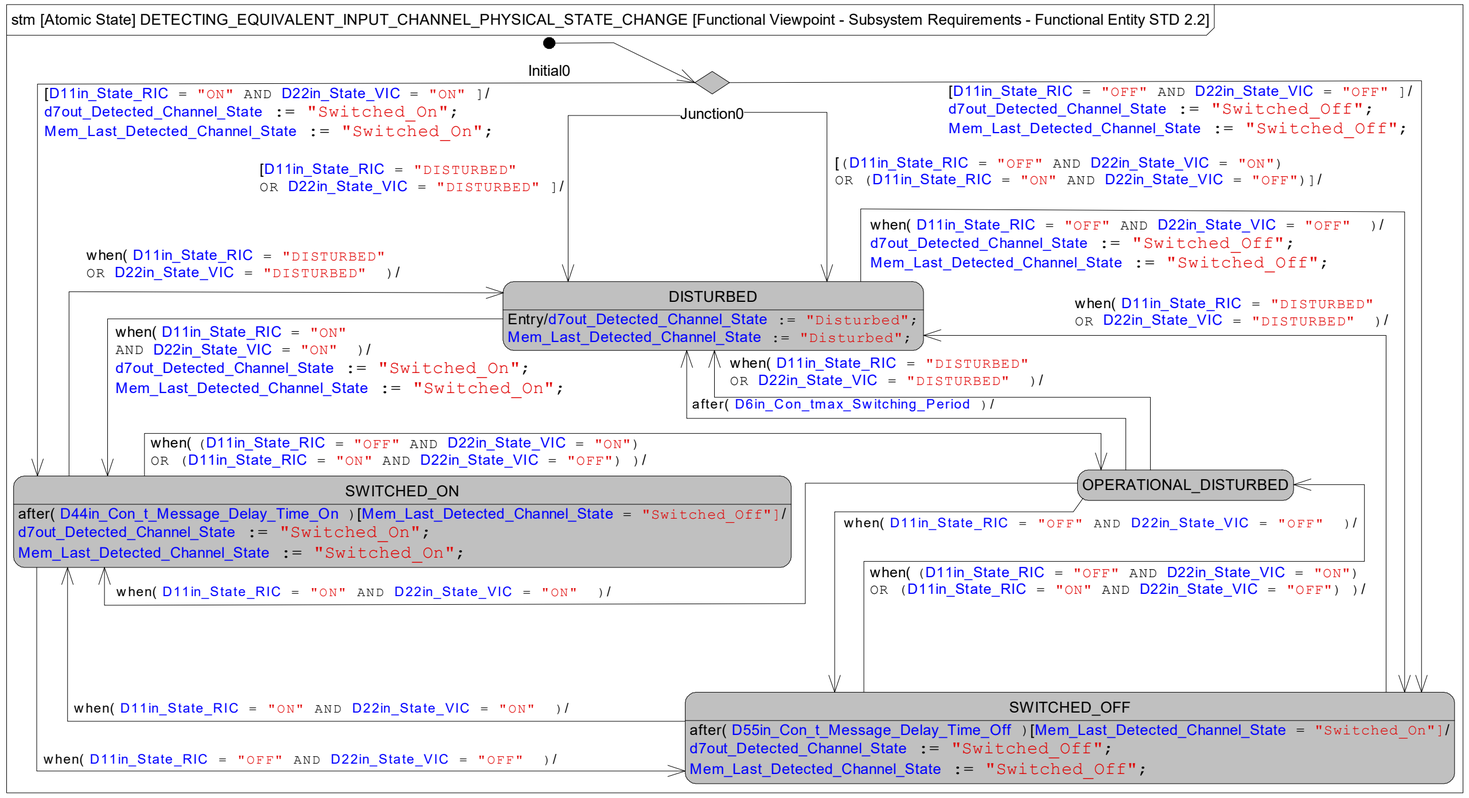
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7419	Def	SETTING_PHYSICAL_OUTPUT_CHANNEL_STATES		Basic IO
Eu.IO.7420	Def	CONTROLLING_FLASHING_CYCLE		Option flashing
Eu.IO.7421	Def	Initial1		Option flashing
Eu.IO.7696	Def	/{Initial1 - OFF_CYCLE}		Option flashing
Eu.IO.7697	Def	OFF_CYCLE		Option flashing
Eu.IO.7698	Def	after(D19in_Con_t_Flash_Period*(1-(D18in_Con_Flash_Duty_Cycle*0.01)))/bc1_Flashing_On_Cycle();{OFF_CYCLE - ON_CYCLE}		Option flashing
Eu.IO.7980	Def	when(d3in_Required_Channel_State = "Flashing")/bc2_Flashing_Off_Cycle();{State-internal in OFF_CYCLE}		Option flashing
Eu.IO.7699	Def	ON_CYCLE		Option flashing
Eu.IO.7700	Def	after(D19in_Con_t_Flash_Period*(D18in_Con_Flash_Duty_Cycle*0.01))/bc2_Flashing_Off_Cycle();{ON_CYCLE - OFF_CYCLE}		Option flashing
Eu.IO.7981	Def	when(d3in_Required_Channel_State = "Flashing")/bc1_Flashing_On_Cycle();{State-internal in ON_CYCLE}		Option flashing
Eu.IO.7449	Def	SWITCHING_ROC		Basic IO
Eu.IO.7450	Def	Initial2		Basic IO
Eu.IO.7451	Def	/{Initial2 - ROC_SWITCHED_OFF}		Basic IO
Eu.IO.7452	Def	ROC_SWITCHED_OFF		Basic IO
Eu.IO.7454	Def	entry/D1out_State_ROC := "Switched_Off";{State-internal in ROC_SWITCHED_OFF}		Basic IO
Eu.IO.7453	Def	when(d3in_Required_Channel_State = "Switched_On")[NOT d13in_Switch_Off_Each_Physical_Output_Channel AND d9in_Monitored_Disturbance_State = "Not_Physically_Disturbed"]/{ROC_SWITCHED_OFF - WAITING_TO_SWITCH_ON_AFTER_DELAY}		Basic IO
Eu.IO.7701	Def	bc1_Flashing_On_Cycle[d3in_Required_Channel_State = "Flashing" AND NOT d13in_Switch_Off_Each_Physical_Output_Channel AND d9in_Monitored_Disturbance_State = "Not_Physically_Disturbed"]/{ROC_SWITCHED_OFF - ROC_SWITCHED_ON}		Option flashing
Eu.IO.7707	Def	WAITING_TO_SWITCH_ON_AFTER_DELAY		Basic IO
Eu.IO.7708	Def	after(D5in_Con_t_Activation_Delay_On)/{WAITING_TO_SWITCH_ON_AFTER_DELAY - ROC_SWITCHED_ON}		Basic IO
Eu.IO.7709	Def	when(d13in_Switch_Off_Each_Physical_Output_Channel OR d9in_Monitored_Disturbance_State = "Physically_Disturbed")/{WAITING_TO_SWITCH_ON_AFTER_DELAY - ROC_SWITCHED_OFF}		Basic IO
Eu.IO.7710	Def	when(d3in_Required_Channel_State = "Switched_Off")/{WAITING_TO_SWITCH_ON_AFTER_DELAY - ROC_SWITCHED_OFF}		Basic IO
Eu.IO.7455	Def	ROC_SWITCHED_ON		Basic IO
Eu.IO.7458	Def	entry/D1out_State_ROC := "Switched_On";{State-internal in ROC_SWITCHED_ON}		Basic IO
Eu.IO.7456	Def	when(d3in_Required_Channel_State = "Switched_Off")/{ROC_SWITCHED_ON - WAITING_TO_SWITCH_OFF_AFTER_DELAY}		Basic IO
Eu.IO.7457	Def	when(d13in_Switch_Off_Each_Physical_Output_Channel OR d9in_Monitored_Disturbance_State = "Physically_Disturbed")/{ROC_SWITCHED_ON - ROC_SWITCHED_OFF}		Basic IO
Eu.IO.7702	Def	bc2_Flashing_Off_Cycle[d3in_Required_Channel_State = "Flashing"]/{ROC_SWITCHED_ON - ROC_SWITCHED_OFF}		Option flashing
Eu.IO.7703	Def	WAITING_TO_SWITCH_OFF_AFTER_DELAY		Basic IO
Eu.IO.7704	Def	after(D6in_Con_t_Activation_Delay_Off)/{WAITING_TO_SWITCH_OFF_AFTER_DELAY - ROC_SWITCHED_OFF}		Basic IO
Eu.IO.7705	Def	when(d13in_Switch_Off_Each_Physical_Output_Channel OR d9in_Monitored_Disturbance_State = "Physically_Disturbed")/{WAITING_TO_SWITCH_OFF_AFTER_DELAY - ROC_SWITCHED_OFF}		Basic IO
Eu.IO.7706	Def	when(d3in_Required_Channel_State = "Switched_On")/{WAITING_TO_SWITCH_OFF_AFTER_DELAY - ROC_SWITCHED_ON}		Basic IO
Eu.IO.7461	Def	SWITCHING_VOC		Basic IO
Eu.IO.7462	Def	Initial3		Basic IO
Eu.IO.7463	Def	/{Initial3 - CHANNEL_CONFIGURATION}		Basic IO
Eu.IO.7474	Def	CHANNEL_CONFIGURATION		Basic IO
Eu.IO.7475	Def	[D4in_Con_Channel_Type = "SINGLE"]/D2out_State_VOC := "Not_Configured";{CHANNEL_CONFIGURATION - VOC_NOT_CONFIGURED}		Basic IO
Eu.IO.7476	Def	[D4in_Con_Channel_Type = "EQUIVALENT"]/{CHANNEL_CONFIGURATION - VOC_SWITCHED_OFF_EQUIVALENT}		Basic IO
Eu.IO.7711	Def	[D4in_Con_Channel_Type = "ANTIVALENT"]/{CHANNEL_CONFIGURATION - VOC_SWITCHED_OFF_ANTIVALENT}		Basic IO
Eu.IO.7464	Def	VOC_NOT_CONFIGURED		Basic IO
Eu.IO.7465	Def	VOC_SWITCHED_OFF_EQUIVALENT		Basic IO
Eu.IO.7468	Def	entry/D2out_State_VOC := "Switched_Off";{State-internal in VOC_SWITCHED_OFF_EQUIVALENT}		Basic IO

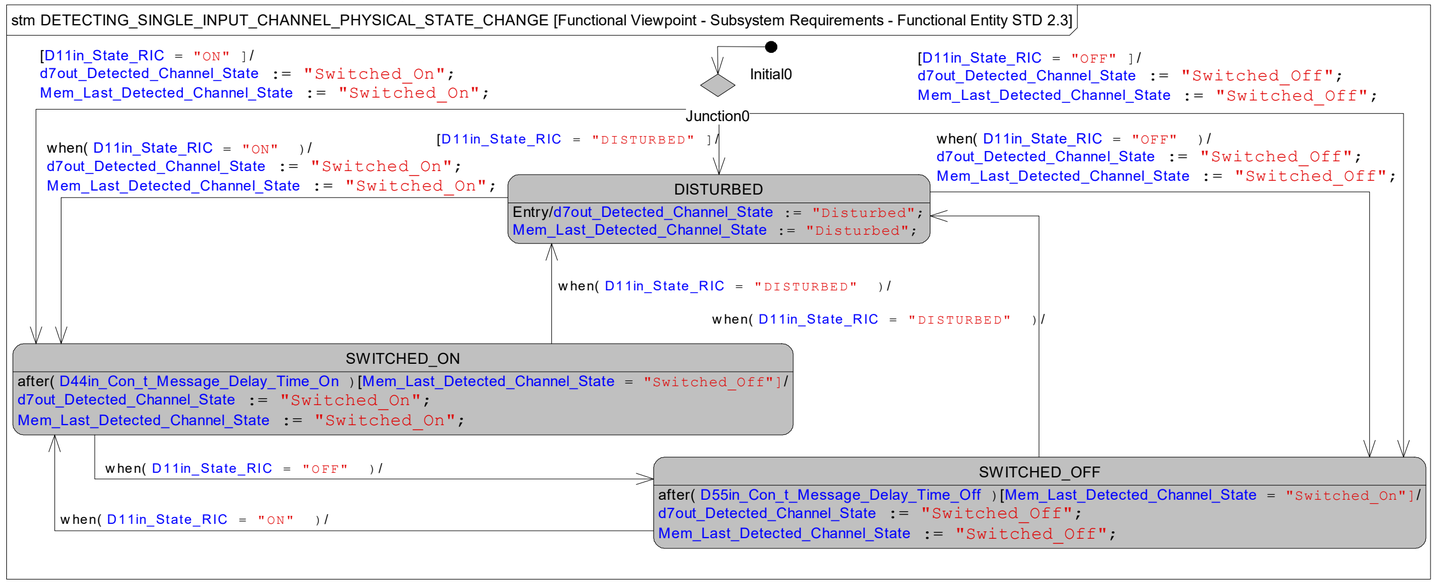
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7466	Def	when(d3in_Required_Channel_State = "Switched_On")[NOT d13in_Switch_Off_Each_Physical_Output_Channel AND d9in_Monitored_Disturbance_State = "Not_Physically_Disturbed"]/{VOC_SWITCHED_OFF_EQUIVALENT - WAITING_TO_SWITCH_ON_AFTER_DELAY_E}		Basic IO
Eu.IO.7715	Def	bc1_Flashing_On_Cycle[d3in_Required_Channel_State = "Flashing" AND NOT d13in_Switch_Off_Each_Physical_Output_Channel AND d9in_Monitored_Disturbance_State = "Not_Physically_Disturbed"]/{VOC_SWITCHED_OFF_EQUIVALENT - VOC_SWITCHED_ON_EQUIVALENT}		Option flashing
Eu.IO.7733	Def	WAITING_TO_SWITCH_ON_AFTER_DELAY_E		Basic IO
Eu.IO.7734	Def	after(D5in_Con_t_Activation_Delay_On)/{WAITING_TO_SWITCH_ON_AFTER_DELAY_E - VOC_SWITCHED_ON_EQUIVALENT}		Basic IO
Eu.IO.7735	Def	when(d13in_Switch_Off_Each_Physical_Output_Channel OR d9in_Monitored_Disturbance_State = "Physically_Disturbed")/{WAITING_TO_SWITCH_ON_AFTER_DELAY_E - VOC_SWITCHED_OFF_EQUIVALENT}		Basic IO
Eu.IO.7736	Def	when(d3in_Required_Channel_State = "Switched_Off")/{WAITING_TO_SWITCH_ON_AFTER_DELAY_E - VOC_SWITCHED_OFF_EQUIVALENT}		Basic IO
Eu.IO.7469	Def	VOC_SWITCHED_ON_EQUIVALENT		Basic IO
Eu.IO.7473	Def	entry/D2out_State_VOC := "Switched_On";{State-internal in VOC_SWITCHED_ON_EQUIVALENT}		Basic IO
Eu.IO.7471	Def	when(d3in_Required_Channel_State = "Switched_Off")/{VOC_SWITCHED_ON_EQUIVALENT - WAITING_TO_SWITCH_OFF_AFTER_DELAY_E}		Basic IO
Eu.IO.7472	Def	when(d13in_Switch_Off_Each_Physical_Output_Channel OR d9in_Monitored_Disturbance_State = "Physically_Disturbed")/{VOC_SWITCHED_ON_EQUIVALENT - VOC_SWITCHED_OFF_EQUIVALENT}		Basic IO
Eu.IO.7720	Def	bc2_Flashing_Off_Cycle[d3in_Required_Channel_State = "Flashing"]/{VOC_SWITCHED_ON_EQUIVALENT - VOC_SWITCHED_OFF_EQUIVALENT}		Option flashing
Eu.IO.7725	Def	WAITING_TO_SWITCH_OFF_AFTER_DELAY_E		Basic IO
Eu.IO.7726	Def	after(D6in_Con_t_Activation_Delay_Off)/{WAITING_TO_SWITCH_OFF_AFTER_DELAY_E - VOC_SWITCHED_OFF_EQUIVALENT}		Basic IO
Eu.IO.7727	Def	when(d13in_Switch_Off_Each_Physical_Output_Channel OR d9in_Monitored_Disturbance_State = "Physically_Disturbed")/{WAITING_TO_SWITCH_OFF_AFTER_DELAY_E - VOC_SWITCHED_OFF_EQUIVALENT}		Basic IO
Eu.IO.7728	Def	when(d3in_Required_Channel_State = "Switched_On")/{WAITING_TO_SWITCH_OFF_AFTER_DELAY_E - VOC_SWITCHED_ON_EQUIVALENT}		Basic IO
Eu.IO.7712	Def	VOC_SWITCHED_OFF_ANTIVALENT		Basic IO
Eu.IO.7713	Def	bc2_Flashing_Off_Cycle[d3in_Required_Channel_State = "Flashing" AND NOT d13in_Switch_Off_Each_Physical_Output_Channel AND d9in_Monitored_Disturbance_State = "Not_Physically_Disturbed"]/{VOC_SWITCHED_OFF_ANTIVALENT - VOC_SWITCHED_ON_ANTIVALENT}		Option flashing
Eu.IO.7714	Def	entry/D2out_State_VOC := "Switched_Off";{State-internal in VOC_SWITCHED_OFF_ANTIVALENT}		Basic IO
Eu.IO.7467	Def	when(d3in_Required_Channel_State = "Switched_Off")[NOT d13in_Switch_Off_Each_Physical_Output_Channel AND d9in_Monitored_Disturbance_State = "Not_Physically_Disturbed"]/{VOC_SWITCHED_OFF_ANTIVALENT - WAITING_TO_SWITCH_OFF_AFTER_DELAY_A}		Basic IO
Eu.IO.7716	Def	VOC_SWITCHED_ON_ANTIVALENT		Basic IO
Eu.IO.7717	Def	bc1_Flashing_On_Cycle[d3in_Required_Channel_State = "Flashing"]/{VOC_SWITCHED_ON_ANTIVALENT - VOC_SWITCHED_OFF_ANTIVALENT}		Option flashing
Eu.IO.7718	Def	entry/D2out_State_VOC := "Switched_On";{State-internal in VOC_SWITCHED_ON_ANTIVALENT}		Basic IO
Eu.IO.7719	Def	when(d13in_Switch_Off_Each_Physical_Output_Channel OR d9in_Monitored_Disturbance_State = "Physically_Disturbed")/{VOC_SWITCHED_ON_ANTIVALENT - VOC_SWITCHED_OFF_ANTIVALENT}		Basic IO
Eu.IO.7470	Def	when(d3in_Required_Channel_State = "Switched_On")/{VOC_SWITCHED_ON_ANTIVALENT - WAITING_TO_SWITCH_ON_AFTER_DELAY_A}		Basic IO
Eu.IO.7721	Def	WAITING_TO_SWITCH_ON_AFTER_DELAY_A		Basic IO
Eu.IO.7722	Def	after(D5in_Con_t_Activation_Delay_On)/{WAITING_TO_SWITCH_ON_AFTER_DELAY_A - VOC_SWITCHED_OFF_ANTIVALENT}		Basic IO
Eu.IO.7723	Def	when(d13in_Switch_Off_Each_Physical_Output_Channel OR d9in_Monitored_Disturbance_State = "Physically_Disturbed")/{WAITING_TO_SWITCH_ON_AFTER_DELAY_A - VOC_SWITCHED_OFF_ANTIVALENT}		Basic IO
Eu.IO.7724	Def	when(d3in_Required_Channel_State = "Switched_Off")/{WAITING_TO_SWITCH_ON_AFTER_DELAY_A - VOC_SWITCHED_ON_ANTIVALENT}		Basic IO
Eu.IO.7729	Def	WAITING_TO_SWITCH_OFF_AFTER_DELAY_A		Basic IO
Eu.IO.7730	Def	after(D6in_Con_t_Activation_Delay_Off)/{WAITING_TO_SWITCH_OFF_AFTER_DELAY_A - VOC_SWITCHED_ON_ANTIVALENT}		Basic IO
Eu.IO.7731	Def	when(d13in_Switch_Off_Each_Physical_Output_Channel OR d9in_Monitored_Disturbance_State = "Physically_Disturbed")/{WAITING_TO_SWITCH_OFF_AFTER_DELAY_A - VOC_SWITCHED_OFF_ANTIVALENT}		Basic IO
Eu.IO.7732	Def	when(d3in_Required_Channel_State = "Switched_On")/{WAITING_TO_SWITCH_OFF_AFTER_DELAY_A - VOC_SWITCHED_OFF_ANTIVALENT}		Basic IO
Eu.IO.7293	Info	F_Detect_Input_Channel_State		Basic IO

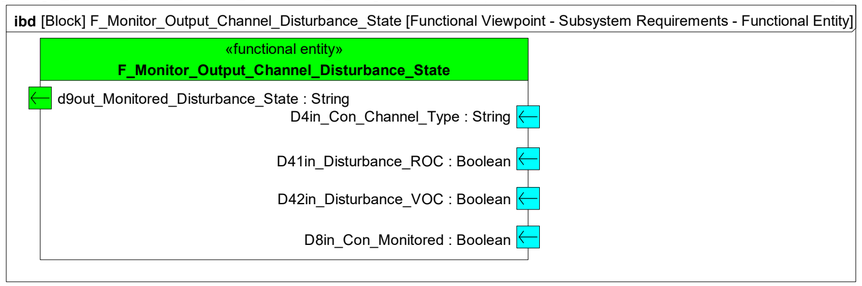
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7397	Req	[Block] F_Detect_Input_Channel_State [Functional Viewpoint - Subsystem Requirements - Functional Entity]  		Basic IO
Eu.IO.7297	Def	D4in_Con_Channel_Type	The port D4in_Con_Channel_Type provides configuration values for the implementation of the Input Channel. The following values are permitted: - Single (assigned to one physical channel) - Antivalent (assigned to two physical channels) - Equivalent (assigned to two physical channels)	Basic IO
Eu.IO.7295	Def	D11in_State_RIC	The port D11in_State_RIC refines the FlowProperty State_Input_Channels and represents the RIC with the permanent available information at the interface IO3. The Con_t_Debouncing_Time needs to be expired if there is a state change at the RIC, this shall be handled by the physical implementation. The following values are permitted: - Switched On - Switched Off - Disturbed	Basic IO
Eu.IO.7296	Def	D22in_State_VIC	The port D22in_State_VIC refines the FlowProperty State_Input_Channels and represents the VIC with the permanent available information at the interface IO3. The Con_t_Debouncing_Time needs to be expired if there is a state change at the VIC, this shall be handled by the physical implementation. The following values are permitted: - Switched On - Switched Off - Disturbed	Basic IO
Eu.IO.7761	Def	D44in_Con_t_Message_Delay_Time_On	The port D44in_Con_t_Message_Delay_Time_On represents the time value for Con_t_Message_Delay_Time_On. This is the delay time to detect the valid logical state Switched On.	Basic IO
Eu.IO.7762	Def	D55in_Con_t_Message_Delay_Time_Off	The port D55in_Con_t_Message_Delay_Time_Off represents the time value for Con_t_Message_Delay_Time_Off. This is the delay time to detect the valid logical state Switched Off.	Basic IO
Eu.IO.7300	Def	D6in_Con_tmax_Switching_Period	The port D6in_Con_tmax_Switching_Period represents the time value for Con_tmax_Switching_Period.	Basic IO
Eu.IO.7301	Def	d7out_Detected_Channel_State		Basic IO
Eu.IO.7302	Info	F_Detect_Input_Channel_State - Behaviour		Basic IO

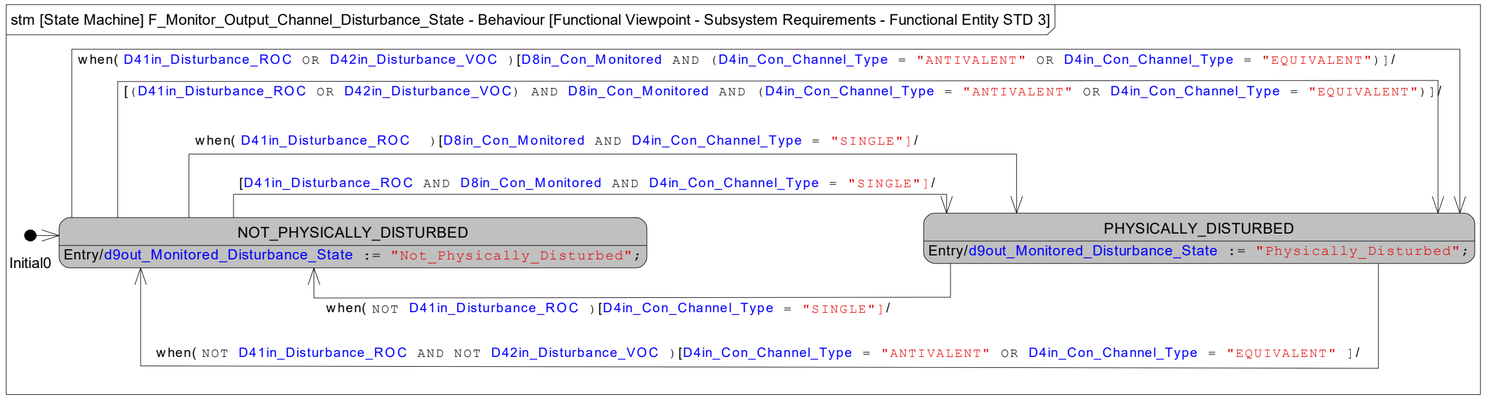
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7305	Req	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 2  	This state machine diagram describes the requirements for the following functionalities: - receives the status of the type of input on Channel	Basic IO
Eu.IO.7303	Def	Initial0		Basic IO
Eu.IO.7304	Def	/{Initial0 - CHANNEL_CONFIGURATION}		Basic IO
Eu.IO.7327	Def	CHANNEL_CONFIGURATION		Basic IO
Eu.IO.7328	Def	[D4in_Con_Channel_Type = "ANTIVALENT"]/{CHANNEL_CONFIGURATION - DETECTING_ANTIVALENT_INPUT_CHANNEL_PHYSICAL_STATE_CHANGE}		Basic IO
Eu.IO.7329	Def	[D4in_Con_Channel_Type = "EQUIVALENT"]/{CHANNEL_CONFIGURATION - DETECTING_EQUIVALENT_INPUT_CHANNEL_PHYSICAL_STATE_CHANGE}		Basic IO
Eu.IO.7330	Def	[D4in_Con_Channel_Type = "SINGLE"]/{CHANNEL_CONFIGURATION - DETECTING_SINGLE_INPUT_CHANNEL_PHYSICAL_STATE_CHANGE}		Basic IO
Eu.IO.7308	Info	DETECTING_ANTIVALENT_INPUT_CHANNEL_PHYSICAL_STATE_CHANGE		Basic IO

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7315	Req	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 2.1  	This state machine diagram describes the requirements for the following functionalities: - observe the status of a antivalent input channel pyhsical state	Basic IO
Eu.IO.7313	Def	Initial0		Basic IO
Eu.IO.7314	Def	/{Initial0 - Junction0}		Basic IO
Eu.IO.7316	Def	Junction0		Basic IO
Eu.IO.7320	Def	[D11in_State_RIC = "ON" AND D22in_State_VIC = "OFF"]/ d7out_Detected_Channel_State := "Switched_On"; Mem_Last_Detected_Channel_State := "Switched_On";{Junction0 - SWITCHED_ON}		Basic IO
Eu.IO.7318	Def	[D11in_State_RIC = "DISTURBED" OR D22in_State_VIC = "DISTURBED"]/{Junction0 - DISTURBED}		Basic IO
Eu.IO.7317	Def	[(D11in_State_RIC = "OFF" AND D22in_State_VIC = "OFF") OR (D11in_State_RIC = "ON" AND D22in_State_VIC = "ON")]/{Junction0 - DISTURBED}		Basic IO
Eu.IO.7319	Def	[D11in_State_RIC = "OFF" AND D22in_State_VIC = "ON"]/ d7out_Detected_Channel_State := "Switched_Off"; Mem_Last_Detected_Channel_State := "Switched_Off";{Junction0 - SWITCHED_OFF}		Basic IO
Eu.IO.7324	Def	SWITCHED_ON		Basic IO
Eu.IO.7325	Def	when((D11in_State_RIC = "OFF" AND D22in_State_VIC = "OFF") OR (D11in_State_RIC = "ON" AND D22in_State_VIC = "ON"))/{SWITCHED_ON - OPERATIONAL_DISTURBED}		Basic IO
Eu.IO.7326	Def	when(D11in_State_RIC = "DISTURBED" OR D22in_State_VIC = "DISTURBED")/{SWITCHED_ON - DISTURBED}		Basic IO
Eu.IO.7674	Def	when(D11in_State_RIC = "OFF" AND D22in_State_VIC = "ON")/{SWITCHED_ON - SWITCHED_OFF}		Basic IO
Eu.IO.7769	Def	after(D44in_Con_t_Message_Delay_Time_On)[Mem_Last_Detected_Channel_State = "Switched_Off"]/ d7out_Detected_Channel_State := "Switched_On"; Mem_Last_Detected_Channel_State := "Switched_On";{State-internal in SWITCHED_ON}		Basic IO
Eu.IO.7309	Def	OPERATIONAL_DISTURBED		Basic IO
Eu.IO.7310	Def	after(D6in_Con_tmax_Switching_Period)/{OPERATIONAL_DISTURBED - DISTURBED}		Basic IO
Eu.IO.7311	Def	when(D11in_State_RIC = "OFF" AND D22in_State_VIC = "ON")/{OPERATIONAL_DISTURBED - SWITCHED_OFF}		Basic IO
Eu.IO.7312	Def	when(D11in_State_RIC = "ON" AND D22in_State_VIC = "OFF")/{OPERATIONAL_DISTURBED - SWITCHED_ON}		Basic IO
Eu.IO.7767	Def	when(D11in_State_RIC = "DISTURBED" OR D22in_State_VIC = "DISTURBED")/{OPERATIONAL_DISTURBED - DISTURBED}		Basic IO

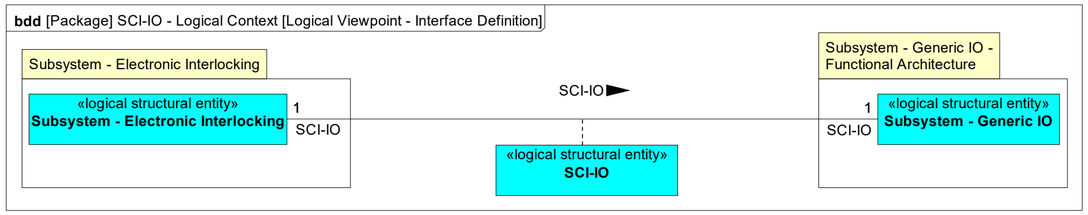
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7321	Def	SWITCHED_OFF		Basic IO
Eu.IO.7322	Def	when((D11in_State_RIC = "OFF" AND D22in_State_VIC = "OFF") OR (D11in_State_RIC = "ON" AND D22in_State_VIC = "ON"))/{SWITCHED_OFF - OPERATIONAL_DISTURBED}		Basic IO
Eu.IO.7323	Def	when(D11in_State_RIC = "DISTURBED" OR D22in_State_VIC = "DISTURBED")/{SWITCHED_OFF - DISTURBED}		Basic IO
Eu.IO.7673	Def	when(D11in_State_RIC = "ON" AND D22in_State_VIC = "OFF")/{SWITCHED_OFF - SWITCHED_ON}		Basic IO
Eu.IO.7768	Def	after(D55in_Con_t_Message_Delay_Time_Off)[Mem_Last_Detected_Channel_State = "Switched_On"]/ d7out_Detected_Channel_State := "Switched_Off"; Mem_Last_Detected_Channel_State := "Switched_Off";{State-internal in SWITCHED_OFF}		Basic IO
Eu.IO.7763	Def	DISTURBED		Basic IO
Eu.IO.7764	Def	entry/d7out_Detected_Channel_State := "Disturbed"; Mem_Last_Detected_Channel_State := "Disturbed";{State-internal in DISTURBED}		Basic IO
Eu.IO.7765	Def	when(D11in_State_RIC = "OFF" AND D22in_State_VIC = "ON")/ d7out_Detected_Channel_State := "Switched_Off"; Mem_Last_Detected_Channel_State := "Switched_Off";{DISTURBED - SWITCHED_OFF}		Basic IO
Eu.IO.7766	Def	when(D11in_State_RIC = "ON" AND D22in_State_VIC = "OFF")/ d7out_Detected_Channel_State := "Switched_On"; Mem_Last_Detected_Channel_State := "Switched_On";{DISTURBED - SWITCHED_ON}		Basic IO
Eu.IO.7331	Info	DETECTING_EQUIVALENT_INPUT_CHANNEL_PHYSICAL_STATE_CHANGE		Basic IO
Eu.IO.7338	Req	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 2.2  	This state machine diagram describes the requirements for the following functionalities: - observe the status of a equivalent input channel pyhsical state	Basic IO
Eu.IO.7336	Def	Initial0		Basic IO
Eu.IO.7337	Def	/{Initial0 - Junction0}		Basic IO
Eu.IO.7339	Def	Junction0		Basic IO
Eu.IO.7343	Def	[D11in_State_RIC = "ON" AND D22in_State_VIC = "ON"]/ d7out_Detected_Channel_State := "Switched_On"; Mem_Last_Detected_Channel_State := "Switched_On";{Junction0 - SWITCHED_ON}		Basic IO
Eu.IO.7340	Def	[D11in_State_RIC = "DISTURBED" OR D22in_State_VIC = "DISTURBED"]/{Junction0 - DISTURBED}		Basic IO

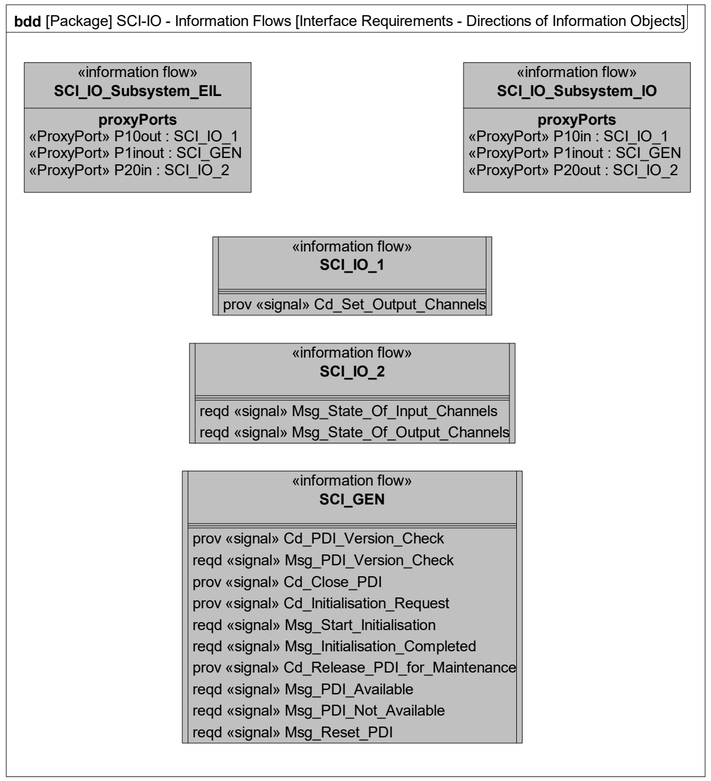
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7341	Def	[(D11in_State_RIC = "OFF" AND D22in_State_VIC = "ON") OR (D11in_State_RIC = "ON" AND D22in_State_VIC = "OFF")]/{Junction0 - DISTURBED}		Basic IO
Eu.IO.7342	Def	[D11in_State_RIC = "OFF" AND D22in_State_VIC = "OFF"]/ d7out_Detected_Channel_State := "Switched_Off"; Mem_Last_Detected_Channel_State := "Switched_Off";{Junction0 - SWITCHED_OFF}		Basic IO
Eu.IO.7347	Def	SWITCHED_ON		Basic IO
Eu.IO.7348	Def	when((D11in_State_RIC = "OFF" AND D22in_State_VIC = "ON") OR (D11in_State_RIC = "ON" AND D22in_State_VIC = "OFF"))/{SWITCHED_ON - OPERATIONAL_DISTURBED}		Basic IO
Eu.IO.7349	Def	when(D11in_State_RIC = "DISTURBED" OR D22in_State_VIC = "DISTURBED")/{SWITCHED_ON - DISTURBED}		Basic IO
Eu.IO.7676	Def	when(D11in_State_RIC = "OFF" AND D22in_State_VIC = "OFF")/{SWITCHED_ON - SWITCHED_OFF}		Basic IO
Eu.IO.7776	Def	after(D44in_Con_t_Message_Delay_Time_On)[Mem_Last_Detected_Channel_State = "Switched_Off"]/ d7out_Detected_Channel_State := "Switched_On"; Mem_Last_Detected_Channel_State := "Switched_On";{State-internal in SWITCHED_ON}		Basic IO
Eu.IO.7332	Def	OPERATIONAL_DISTURBED		Basic IO
Eu.IO.7333	Def	after(D6in_Con_tmax_Switching_Period)/{OPERATIONAL_DISTURBED - DISTURBED}		Basic IO
Eu.IO.7334	Def	when(D11in_State_RIC = "OFF" AND D22in_State_VIC = "OFF")/{OPERATIONAL_DISTURBED - SWITCHED_OFF}		Basic IO
Eu.IO.7335	Def	when(D11in_State_RIC = "ON" AND D22in_State_VIC = "ON")/{OPERATIONAL_DISTURBED - SWITCHED_ON}		Basic IO
Eu.IO.7774	Def	when(D11in_State_RIC = "DISTURBED" OR D22in_State_VIC = "DISTURBED")/{OPERATIONAL_DISTURBED - DISTURBED}		Basic IO
Eu.IO.7344	Def	SWITCHED_OFF		Basic IO
Eu.IO.7345	Def	when((D11in_State_RIC = "OFF" AND D22in_State_VIC = "ON") OR (D11in_State_RIC = "ON" AND D22in_State_VIC = "OFF"))/{SWITCHED_OFF - OPERATIONAL_DISTURBED}		Basic IO
Eu.IO.7346	Def	when(D11in_State_RIC = "DISTURBED" OR D22in_State_VIC = "DISTURBED")/{SWITCHED_OFF - DISTURBED}		Basic IO
Eu.IO.7675	Def	when(D11in_State_RIC = "ON" AND D22in_State_VIC = "ON")/{SWITCHED_OFF - SWITCHED_ON}		Basic IO
Eu.IO.7775	Def	after(D55in_Con_t_Message_Delay_Time_Off)[Mem_Last_Detected_Channel_State = "Switched_On"]/ d7out_Detected_Channel_State := "Switched_Off"; Mem_Last_Detected_Channel_State := "Switched_Off";{State-internal in SWITCHED_OFF}		Basic IO
Eu.IO.7770	Def	DISTURBED		Basic IO
Eu.IO.7771	Def	entry/d7out_Detected_Channel_State := "Disturbed"; Mem_Last_Detected_Channel_State := "Disturbed";{State-internal in DISTURBED}		Basic IO
Eu.IO.7772	Def	when(D11in_State_RIC = "OFF" AND D22in_State_VIC = "OFF")/ d7out_Detected_Channel_State := "Switched_Off"; Mem_Last_Detected_Channel_State := "Switched_Off";{DISTURBED - SWITCHED_OFF}		Basic IO
Eu.IO.7773	Def	when(D11in_State_RIC = "ON" AND D22in_State_VIC = "ON")/ d7out_Detected_Channel_State := "Switched_On"; Mem_Last_Detected_Channel_State := "Switched_On";{DISTURBED - SWITCHED_ON}		Basic IO
Eu.IO.7352	Info	DETECTING_SINGLE_INPUT_CHANNEL_PHYSICAL_STATE_CHANGE		Basic IO
Eu.IO.7358	Req	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 2.3  	This state machine diagram describes the requirements for the following functionalities: - observe the status of a single input channel pyhsical state	Basic IO

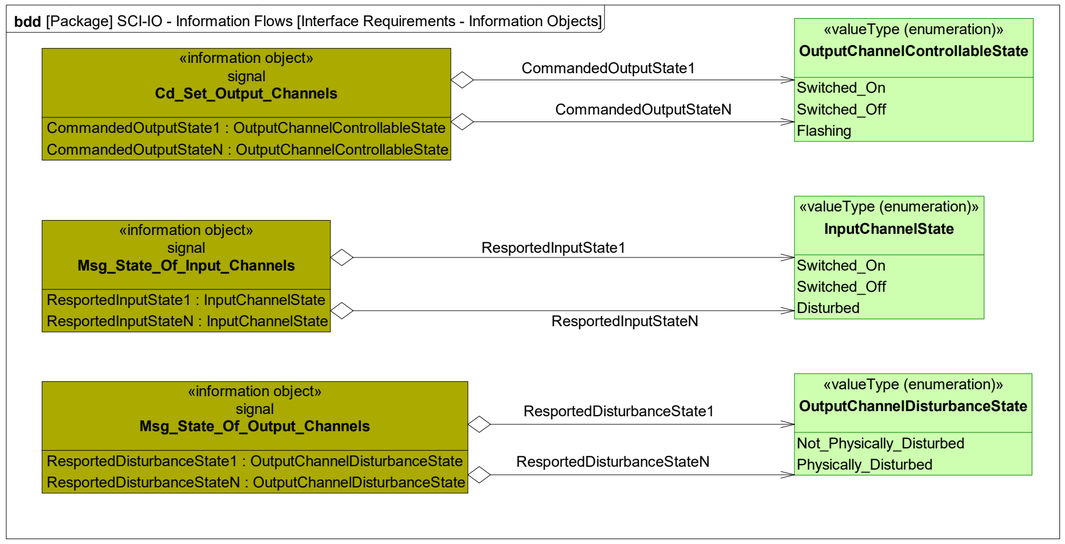
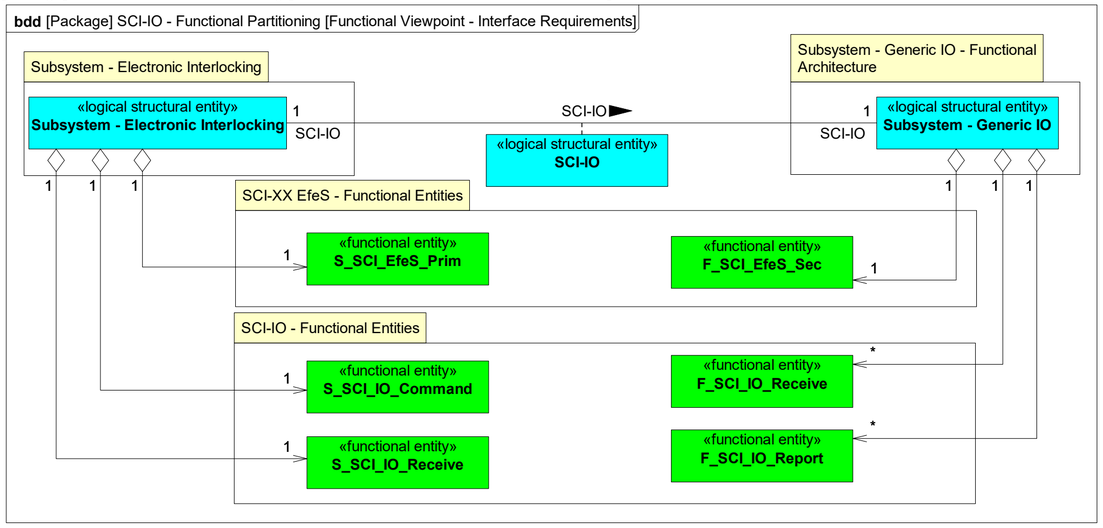
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7356	Def	Initial0		Basic IO
Eu.IO.7357	Def	/{Initial0 - Junction0}		Basic IO
Eu.IO.7359	Def	Junction0		Basic IO
Eu.IO.7362	Def	[D11in_State_RIC = "ON"]/ d7out_Detected_Channel_State := "Switched_On"; Mem_Last_Detected_Channel_State := "Switched_On";{Junction0 - SWITCHED_ON}		Basic IO
Eu.IO.7360	Def	[D11in_State_RIC = "DISTURBED"]/{Junction0 - DISTURBED}		Basic IO
Eu.IO.7361	Def	[D11in_State_RIC = "OFF"]/ d7out_Detected_Channel_State := "Switched_Off"; Mem_Last_Detected_Channel_State := "Switched_Off";{Junction0 - SWITCHED_OFF}		Basic IO
Eu.IO.7366	Def	SWITCHED_ON		Basic IO
Eu.IO.7367	Def	when(D11in_State_RIC = "DISTURBED")/{SWITCHED_ON - DISTURBED}		Basic IO
Eu.IO.7368	Def	when(D11in_State_RIC = "OFF")/{SWITCHED_ON - SWITCHED_OFF}		Basic IO
Eu.IO.7779	Def	after(D44in_Con_t_Message_Delay_Time_On)[Mem_Last_Detected_Channel_State = "Switched_Off"]/ d7out_Detected_Channel_State := "Switched_On"; Mem_Last_Detected_Channel_State := "Switched_On";{State-internal in SWITCHED_ON}		Basic IO
Eu.IO.7353	Def	DISTURBED		Basic IO
Eu.IO.7354	Def	when(D11in_State_RIC = "OFF")/ d7out_Detected_Channel_State := "Switched_Off"; Mem_Last_Detected_Channel_State := "Switched_Off";{DISTURBED - SWITCHED_OFF}		Basic IO
Eu.IO.7355	Def	when(D11in_State_RIC = "ON")/ d7out_Detected_Channel_State := "Switched_On"; Mem_Last_Detected_Channel_State := "Switched_On";{DISTURBED - SWITCHED_ON}		Basic IO
Eu.IO.7777	Def	entry/d7out_Detected_Channel_State := "Disturbed"; Mem_Last_Detected_Channel_State := "Disturbed";{State-internal in DISTURBED}		Basic IO
Eu.IO.7363	Def	SWITCHED_OFF		Basic IO
Eu.IO.7364	Def	when(D11in_State_RIC = "DISTURBED")/{SWITCHED_OFF - DISTURBED}		Basic IO
Eu.IO.7365	Def	when(D11in_State_RIC = "ON")/{SWITCHED_OFF - SWITCHED_ON}		Basic IO
Eu.IO.7778	Def	after(D55in_Con_t_Message_Delay_Time_Off)[Mem_Last_Detected_Channel_State = "Switched_On"]/ d7out_Detected_Channel_State := "Switched_Off"; Mem_Last_Detected_Channel_State := "Switched_Off";{State-internal in SWITCHED_OFF}		Basic IO
Eu.IO.7780	Info	F_Monitor_Output_Channel_Disturbance_State		Basic IO
Eu.IO.7781	Req	[Block] F_Monitor_Output_Channel_Disturbance_State [Functional Viewpoint - Subsystem Requirements - Functional Entity]  		Basic IO
Eu.IO.7782	Def	D41in_Disturbance_ROC	The port D41in_Disturbance_ROC represents the physically disturbance at the ROC. If D41in_Disturbance_ROC is true the Adjacent IO System do not switch to the permanent available target state from D1out_State_ROC. It is expected that it turns to false if a command was executed. The following values are permitted: - True: Physically Disturbed - False: Not Physically Disturbed	Basic IO
Eu.IO.7783	Def	D42in_Disturbance_VOC	The port D42in_Disturbance_VOC represents the physically disturbance at the VOC. If D42in_Disturbance_VOC is true the Adjacent IO System do not switch to the permanent available target state from D2out_State_VOC. It is expected that it turns to false if a command was executed. The following values are permitted: - True: Physically Disturbed - False: Not Physically Disturbed	Basic IO

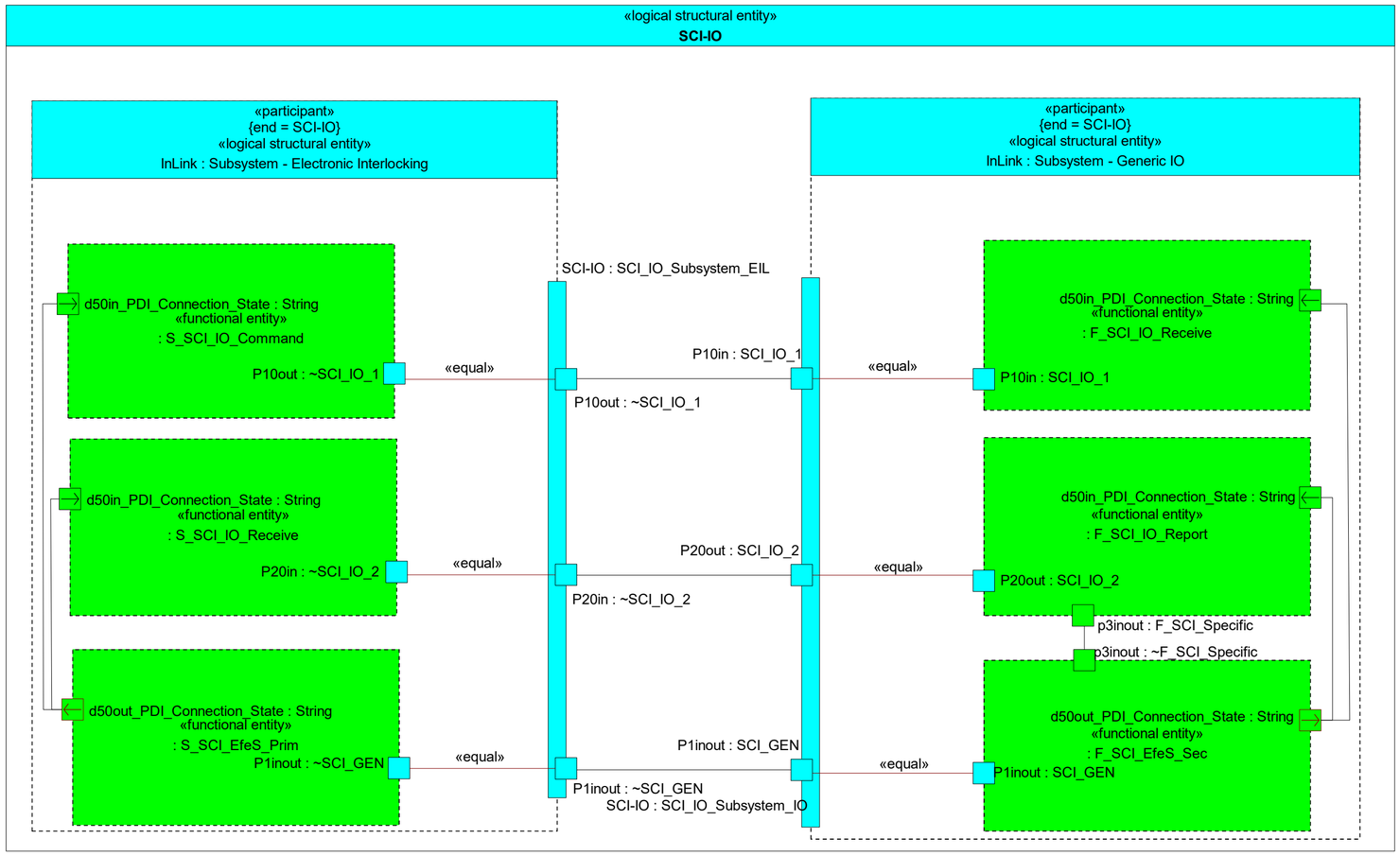
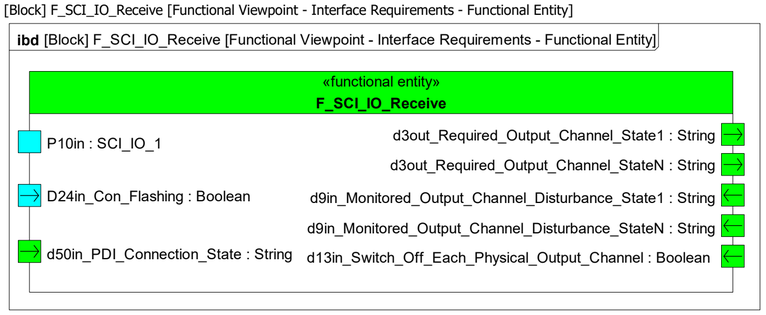
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7784	Def	D4in_Con_Channel_Type	The port D4in_Con_Channel_Type provides configuration values for the implementation of the logical Output Channel. The following values are permitted: - Single (assigned to one physical channel) - Antivalent (assigned to two physical channels) - Equivalent (assigned to two physical channels)	Basic IO
Eu.IO.7785	Def	D8in_Con_Monitored	The port D8in_Con_Monitored provides configuration values for the logical Output Channel. The following values are permitted: - True: Output Channel is monitored - False: Output Channel is not monitored	Basic IO
Eu.IO.7786	Def	d9out_Monitored_Disturbance_State	The port d9out_Monitored_Disturbance_State provides the monitored state of one Output Channel. The following values are permitted: - Physically Disturbed - Not Physically Disturbed	Basic IO
Eu.IO.7787	Info	F_Monitor_Output_Channel_Disturbance_State - Behaviour		Basic IO
Eu.IO.7788	Req	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 3  	This state machine diagram describes the requirements for the following functionalities: - monitors the physical output channel disturbance state	Basic IO
Eu.IO.7789	Def	Initial0		Basic IO
Eu.IO.7790	Def	/{Initial0 - NOT_PHYSICALLY_DISTURBED}		Basic IO
Eu.IO.7791	Def	NOT_PHYSICALLY_DISTURBED		Basic IO
Eu.IO.7792	Def	[(D41in_Disturbance_ROC OR D42in_Disturbance_VOC) AND D8in_Con_Monitored AND (D4in_Con_Channel_Type = "ANTIVALENT" OR D4in_Con_Channel_Type = "EQUIVALENT")]/{NOT_PHYSICALLY_DISTURBED - PHYSICALLY_DISTURBED}		Basic IO
Eu.IO.7793	Def	[D41in_Disturbance_ROC AND D8in_Con_Monitored AND D4in_Con_Channel_Type = "SINGLE"]/{NOT_PHYSICALLY_DISTURBED - PHYSICALLY_DISTURBED}		Basic IO
Eu.IO.7794	Def	entry/d9out_Monitored_Disturbance_State := "Not_Physically_Disturbed";{State-internal in NOT_PHYSICALLY_DISTURBED}		Basic IO
Eu.IO.7795	Def	when(D41in_Disturbance_ROC)[D8in_Con_Monitored AND D4in_Con_Channel_Type = "SINGLE"]/{NOT_PHYSICALLY_DISTURBED - PHYSICALLY_DISTURBED}		Basic IO
Eu.IO.7796	Def	when(D41in_Disturbance_ROC OR D42in_Disturbance_VOC)[D8in_Con_Monitored AND (D4in_Con_Channel_Type = "ANTIVALENT" OR D4in_Con_Channel_Type = "EQUIVALENT")]/{NOT_PHYSICALLY_DISTURBED - PHYSICALLY_DISTURBED}		Basic IO
Eu.IO.7797	Def	PHYSICALLY_DISTURBED		Basic IO
Eu.IO.7798	Def	entry/d9out_Monitored_Disturbance_State := "Physically_Disturbed";{State-internal in PHYSICALLY_DISTURBED}		Basic IO
Eu.IO.7799	Def	when(NOT D41in_Disturbance_ROC)[D4in_Con_Channel_Type = "SINGLE"]/{PHYSICALLY_DISTURBED - NOT_PHYSICALLY_DISTURBED}		Basic IO
Eu.IO.7800	Def	when(NOT D41in_Disturbance_ROC AND NOT D42in_Disturbance_VOC)[D4in_Con_Channel_Type = "ANTIVALENT" OR D4in_Con_Channel_Type = "EQUIVALENT"]/{PHYSICALLY_DISTURBED - NOT_PHYSICALLY_DISTURBED}		Basic IO
Eu.IO.408	Head	**3.4 Subsystem - Generic IO - Interfaces**		
Eu.IO.422	Head	**3.4.1 SCI-IO (Subsystem - Electronic Interlocking)**		
Eu.IO.7279	Head	**3.4.1.1 SCI-IO - Logical Viewpoint**		
Eu.IO.7977	Head	**3.4.1.1.1 SCI-IO - Logical Context**		

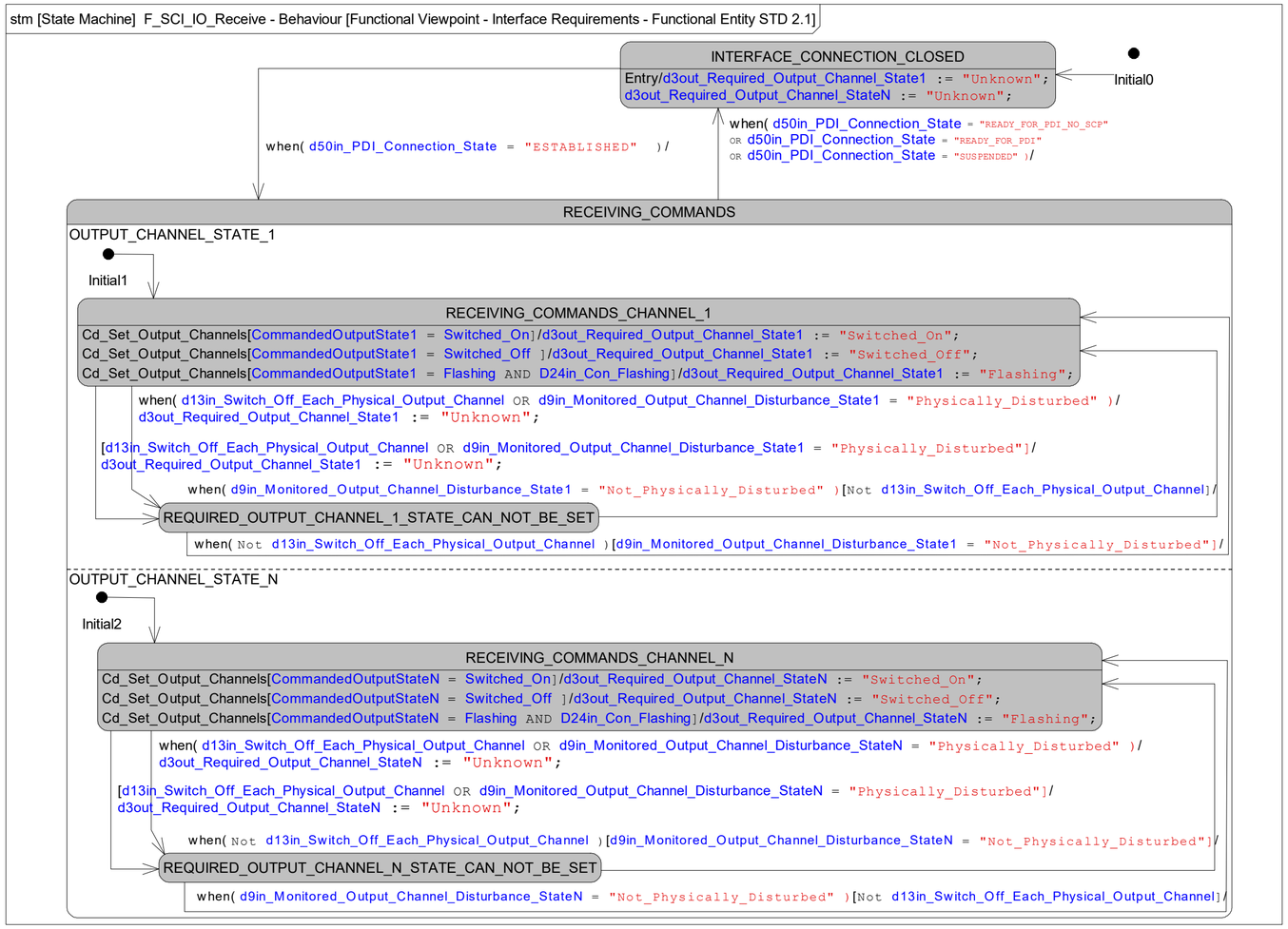
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7947	Def	[Package] SCI-IO - Logical Context [Logical Viewpoint - Interface Definition]		Basic IO
Eu.IO.7801	Head	**3.4.1.2 SCI-IO - Information Flows**		
Eu.IO.7803	Def	[Package] SCI-IO - Information Flows [Interface Requirements - Directions of Information Objects]		Basic IO

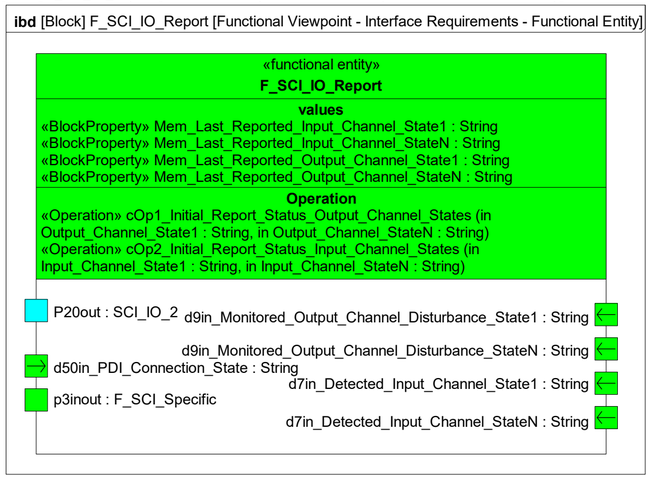




ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7816	Def	[Package] SCI-IO - Information Flows [Interface Requirements - Information Objects]  		Basic IO
Eu.IO.1693	Info	The generic commands and messages through the SCI-IO are specified in [Eu.Doc.119].		Basic IO
Eu.IO.7962	Def	Cd_Set_Output_Channels	Command (Cd) from Subsystem - Electronic Interlocking to Subsystem - Generic IO, to set all the states, transmitted as parameter, at all the logical Output Channels of the transmitted Adjacent IO System.	Basic IO
Eu.IO.7963	Def	Msg_State_Of_Input_Channels	Message (Msg) from Subsystem - Generic IO to Subsystem - Electronic Interlocking to report the status of the current state of the logical Input Channels of the transmitted Adjacent IO System.	Basic IO
Eu.IO.7964	Def	Msg_State_Of_Output_Channels	Message (Msg) from Subsystem - Generic IO to Subsystem - Electronic Interlocking to report the status related to disturbance of the logical Output Channels of the transmitted Adjacent IO System.	Basic IO
Eu.IO.7817	Head	**3.4.1.3 SCI-IO - Functional Viewpoint**		
Eu.IO.7974	Head	**3.4.1.3.1 SCI-IO - Functional Partitioning**		
Eu.IO.7284	Def	[Package] SCI-IO - Functional Partitioning [Functional Viewpoint - Interface Requirements]  		Basic IO
Eu.IO.7960	Head	**3.4.1.3.2 SCI-IO - Functional Architecture**		
Eu.IO.7280	Info	SCI-IO		Basic IO

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7281	Def	[Block] SCI-IO [Functional Viewpoint - Interface Requirements - Functional Architecture] **ibd** [Block] SCI-IO [Functional Viewpoint - Interface Requirements - Functional Architecture]  		Basic IO
Eu.IO.7961	Head	**3.4.1.3.3 SCI-IO - Functional Entities**		
Eu.IO.7819	Info	F_SCI_IO_Receive		Basic IO
Eu.IO.7851	Req	[Block] F_SCI_IO_Receive [Functional Viewpoint - Interface Requirements - Functional Entity] **ibd** [Block] F_SCI_IO_Receive [Functional Viewpoint - Interface Requirements - Functional Entity]  		Basic IO
Eu.IO.7856	Def	d50in_PDI_Connection_State		Basic IO
Eu.IO.7852	Def	P10in	The port P10in exchanges information objects according to SCI_IO_1.	Basic IO
Eu.IO.7853	Def	d13in_Switch_Off_Each_Physical_Output_Channel		Basic IO
Eu.IO.7854	Def	d3out_Required_Output_Channel_State1		Basic IO
Eu.IO.7855	Def	d3out_Required_Output_Channel_StateN		Basic IO
Eu.IO.7857	Def	d9in_Monitored_Output_Channel_Disturbance_State1		Basic IO

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7858	Def	d9in_Monitored_Output_Channel_Disturbance_StateN		Basic IO
Eu.IO.7982	Def	D24in_Con_Flashing	The port D24in_Con_Flashing provides configuration values for the Subsystem - Generic IO. The following values are permitted: - True: Option flashing is configured - False: Option flashing is not configured	Option flashing
Eu.IO.7820	Info	F_SCI_IO_Receive - Behaviour		Basic IO
Eu.IO.7821	Req	Functional Viewpoint - Interface Requirements - Functional Entity STD 2.1  	This state machine diagram describes the requirements for the following functionalities: - receives the commands for Channel 1 and forwards it to the internal logic - receives the commands for Channel N and forwards it to the internal logic	Basic IO
Eu.IO.7822	Def	Initial0		Basic IO
Eu.IO.7823	Def	/{Initial0 - INTERFACE_CONNECTION_CLOSED}		Basic IO
Eu.IO.7824	Def	INTERFACE_CONNECTION_CLOSED		Basic IO
Eu.IO.7825	Def	entry/d3out_Required_Output_Channel_State1 := "Unknown"; d3out_Required_Output_Channel_StateN := "Unknown";{State-internal in INTERFACE_CONNECTION_CLOSED}		Basic IO
Eu.IO.7826	Def	when(d50in_PDI_Connection_State = "ESTABLISHED")/{INTERFACE_CONNECTION_CLOSED - RECEIVING_COMMANDS}		Basic IO
Eu.IO.7827	Def	RECEIVING_COMMANDS		Basic IO
Eu.IO.7828	Def	OUTPUT_CHANNEL_STATE_1		Basic IO
Eu.IO.7829	Def	Initial1		Basic IO
Eu.IO.7830	Def	/{Initial1 - RECEIVING_COMMANDS_CHANNEL_1}		Basic IO

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7831	Def	RECEIVING_COMMANDS_CHANNEL_1		Basic IO
Eu.IO.7832	Def	Cd_Set_Output_Channels[CommandedOutputState1 = Switched_Off]/d3out_Required_Output_Channel_State1 := "Switched_Off";{State-internal in RECEIVING_COMMANDS_CHANNEL_1}		Basic IO
Eu.IO.7833	Def	Cd_Set_Output_Channels[CommandedOutputState1 = Flashing AND D24in_Con_Flashing]/d3out_Required_Output_Channel_State1 := "Flashing";{State-internal in RECEIVING_COMMANDS_CHANNEL_1}		Option flashing
Eu.IO.7834	Def	Cd_Set_Output_Channels[CommandedOutputState1 = Switched_On]/d3out_Required_Output_Channel_State1 := "Switched_On";{State-internal in RECEIVING_COMMANDS_CHANNEL_1}		Basic IO
Eu.IO.7835	Def	when(d13in_Switch_Off_Each_Physical_Output_Channel OR d9in_Monitored_Output_Channel_Disturbance_State1 = "Physically_Disturbed")/ d3out_Required_Output_Channel_State1 := "Unknown";{RECEIVING_COMMANDS_CHANNEL_1 - REQUIRED_OUTPUT_CHANNEL_1_STATE_CAN_NOT_BE_SET}		Basic IO
Eu.IO.7983	Def	[d13in_Switch_Off_Each_Physical_Output_Channel OR d9in_Monitored_Output_Channel_Disturbance_State1 = "Physically_Disturbed"]/ d3out_Required_Output_Channel_State1 := "Unknown";{RECEIVING_COMMANDS_CHANNEL_1 - REQUIRED_OUTPUT_CHANNEL_1_STATE_CAN_NOT_BE_SET}		Basic IO
Eu.IO.7836	Def	REQUIRED_OUTPUT_CHANNEL_1_STATE_CAN_NOT_BE_SET		Basic IO
Eu.IO.7837	Def	when(d9in_Monitored_Output_Channel_Disturbance_State1 = "Not_Physically_Disturbed")[Not d13in_Switch_Off_Each_Physical_Output_Channel]/{REQUIRED_OUTPUT_CHANNEL_1_STATE_CAN_NOT_BE_SET - RECEIVING_COMMANDS_CHANNEL_1}		Basic IO
Eu.IO.7838	Def	when(Not d13in_Switch_Off_Each_Physical_Output_Channel)[d9in_Monitored_Output_Channel_Disturbance_State1 = "Not_Physically_Disturbed"]/{REQUIRED_OUTPUT_CHANNEL_1_STATE_CAN_NOT_BE_SET - RECEIVING_COMMANDS_CHANNEL_1}		Basic IO
Eu.IO.7839	Def	OUTPUT_CHANNEL_STATE_N		Basic IO
Eu.IO.7840	Def	RECEIVING_COMMANDS_CHANNEL_N		Basic IO
Eu.IO.7841	Def	Cd_Set_Output_Channels[CommandedOutputStateN = Switched_Off]/d3out_Required_Output_Channel_StateN := "Switched_Off";{State-internal in RECEIVING_COMMANDS_CHANNEL_N}		Basic IO
Eu.IO.7842	Def	Cd_Set_Output_Channels[CommandedOutputStateN = Flashing AND D24in_Con_Flashing]/d3out_Required_Output_Channel_StateN := "Flashing";{State-internal in RECEIVING_COMMANDS_CHANNEL_N}		Option flashing
Eu.IO.7843	Def	Cd_Set_Output_Channels[CommandedOutputStateN = Switched_On]/d3out_Required_Output_Channel_StateN := "Switched_On";{State-internal in RECEIVING_COMMANDS_CHANNEL_N}		Basic IO
Eu.IO.7844	Def	when(d13in_Switch_Off_Each_Physical_Output_Channel OR d9in_Monitored_Output_Channel_Disturbance_StateN = "Physically_Disturbed")/ d3out_Required_Output_Channel_StateN := "Unknown";{RECEIVING_COMMANDS_CHANNEL_N - REQUIRED_OUTPUT_CHANNEL_N_STATE_CAN_NOT_BE_SET}		Basic IO
Eu.IO.7984	Def	[d13in_Switch_Off_Each_Physical_Output_Channel OR d9in_Monitored_Output_Channel_Disturbance_StateN = "Physically_Disturbed"]/ d3out_Required_Output_Channel_StateN := "Unknown";{RECEIVING_COMMANDS_CHANNEL_N - REQUIRED_OUTPUT_CHANNEL_N_STATE_CAN_NOT_BE_SET}		Basic IO
Eu.IO.7845	Def	Initial2		Basic IO
Eu.IO.7846	Def	/{Initial2 - RECEIVING_COMMANDS_CHANNEL_N}		Basic IO
Eu.IO.7847	Def	REQUIRED_OUTPUT_CHANNEL_N_STATE_CAN_NOT_BE_SET		Basic IO
Eu.IO.7848	Def	when(d9in_Monitored_Output_Channel_Disturbance_StateN = "Not_Physically_Disturbed")[Not d13in_Switch_Off_Each_Physical_Output_Channel]/{REQUIRED_OUTPUT_CHANNEL_N_STATE_CAN_NOT_BE_SET - RECEIVING_COMMANDS_CHANNEL_N}		Basic IO
Eu.IO.7849	Def	when(Not d13in_Switch_Off_Each_Physical_Output_Channel)[d9in_Monitored_Output_Channel_Disturbance_StateN = "Not_Physically_Disturbed"]/{REQUIRED_OUTPUT_CHANNEL_N_STATE_CAN_NOT_BE_SET - RECEIVING_COMMANDS_CHANNEL_N}		Basic IO
Eu.IO.7850	Def	when(d50in_PDI_Connection_State = "READY_FOR_PDI_NO_SCP" OR d50in_PDI_Connection_State = "READY_FOR_PDI" OR d50in_PDI_Connection_State = "SUSPENDED")/{RECEIVING_COMMANDS - INTERFACE_CONNECTION_CLOSED}		Basic IO
Eu.IO.7486	Info	F_SCI_IO_Report		Basic IO
Eu.IO.7549	Req	[Block] F_SCI_IO_Report [Functional Viewpoint - Interface Requirements - Functional Entity]  		Basic IO
Eu.IO.7540	Def	d50in_PDI_Connection_State		Basic IO
Eu.IO.7535	Def	d7in_Detected_Input_Channel_State1		Basic IO
Eu.IO.7895	Def	d7in_Detected_Input_Channel_StateN		Basic IO
Eu.IO.7539	Def	d9in_Monitored_Output_Channel_Disturbance_State1		Basic IO
Eu.IO.7896	Def	d9in_Monitored_Output_Channel_Disturbance_StateN		Basic IO
Eu.IO.7894	Def	P20out	The port P20out exchanges information objects according to SCI_IO_2.	Basic IO

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7973	Def	p3inout		Basic IO

Eu.IO.7890	Def	/* cOp1_Initial_Report_Status_Output_Channel_States*/
		
		if Output_Channel_State1 = "Not_Physically_Disturbed" AND Output_Channel_StateN = "Not_Physically_Disturbed" then
		send Msg_State_Of_Output_Channels(OutputChannelDisturbanceState.Not_Physically_Disturbed,OutputChannelDisturbanceState.Not_Physically_Disturbed) to P20out; Mem_Last_Reported_Output_Channel_State1 := "Not_Physically_Disturbed";
		Mem_Last_Reported_Output_Channel_StateN := "Not_Physically_Disturbed";
		elseif Output_Channel_State1 = "Physically_Disturbed" AND Output_Channel_StateN = "Physically_Disturbed" then
		send Msg_State_Of_Output_Channels(OutputChannelDisturbanceState.Physically_Disturbed,OutputChannelDisturbanceState.Physically_Disturbed) to P20out; Mem_Last_Reported_Output_Channel_State1 := "Physically_Disturbed";
		Mem_Last_Reported_Output_Channel_StateN := "Physically_Disturbed";
		elseif Output_Channel_State1 = "Physically_Disturbed" AND Output_Channel_StateN = "Not_Physically_Disturbed" then
		send Msg_State_Of_Output_Channels(OutputChannelDisturbanceState.Physically_Disturbed,OutputChannelDisturbanceState.Not_Physically_Disturbed) to P20out; Mem_Last_Reported_Output_Channel_State1 := "Physically_Disturbed";
		Mem_Last_Reported_Output_Channel_StateN := "Not_Physically_Disturbed";
		elseif Output_Channel_State1 = "Not_Physically_Disturbed" AND Output_Channel_StateN = "Physically_Disturbed" then
		send Msg_State_Of_Output_Channels(OutputChannelDisturbanceState.Not_Physically_Disturbed,OutputChannelDisturbanceState.Physically_Disturbed) to P20out; Mem_Last_Reported_Output_Channel_State1 := "Not_Physically_Disturbed";
		Mem_Last_Reported_Output_Channel_StateN := "Physically_Disturbed";
		end if
			cOp1_Initial_Report_Status_Output_Channel_States	Basic IO

Eu.IO.7892	Def	/* cOp2_Initial_Report_Status_Input_Channel_States*/
		
		if Input_Channel_State1 = "Switched_On" AND Input_Channel_StateN = "Switched_On" then
		send Msg_State_Of_Input_Channels(InputChannelState.Switched_On,InputChannelState.Switched_On) to P20out;
		Mem_Last_Reported_Input_Channel_State1 := "Switched_On"; Mem_Last_Reported_Input_Channel_StateN := "Switched_On";
		elseif Input_Channel_State1 = "Switched_Off" AND Input_Channel_StateN = "Switched_Off" then
		send Msg_State_Of_Input_Channels(InputChannelState.Switched_Off,InputChannelState.Switched_Off) to P20out;
		Mem_Last_Reported_Input_Channel_State1 := "Switched_Off"; Mem_Last_Reported_Input_Channel_StateN := "Switched_Off";
		elseif Input_Channel_State1 = "Disturbed" AND Input_Channel_StateN = "Disturbed" then
		send Msg_State_Of_Input_Channels(InputChannelState.Disturbed,InputChannelState.Disturbed) to P20out;
		Mem_Last_Reported_Input_Channel_State1 := "Disturbed"; Mem_Last_Reported_Input_Channel_StateN := "Disturbed";
		elseif Input_Channel_State1 = "Switched_Off" AND Input_Channel_StateN = "Switched_On" then
		send Msg_State_Of_Input_Channels(InputChannelState.Switched_Off,InputChannelState.Switched_On) to P20out;
		Mem_Last_Reported_Input_Channel_State1 := "Switched_Off"; Mem_Last_Reported_Input_Channel_StateN := "Switched_On";
		elseif Input_Channel_State1 = "Switched_On" AND Input_Channel_StateN = "Switched_Off" then
		send Msg_State_Of_Input_Channels(InputChannelState.Switched_On,InputChannelState.Switched_Off) to P20out;
		Mem_Last_Reported_Input_Channel_State1 := "Switched_On"; Mem_Last_Reported_Input_Channel_StateN := "Switched_Off";
		elseif Input_Channel_State1 = "Switched_Off" AND Input_Channel_StateN = "Disturbed" then
		send Msg_State_Of_Input_Channels(InputChannelState.Switched_Off,InputChannelState.Disturbed) to P20out;
		Mem_Last_Reported_Input_Channel_State1 := "Switched_Off"; Mem_Last_Reported_Input_Channel_StateN := "Disturbed";
		elseif Input_Channel_State1 = "Switched_On" AND Input_Channel_StateN = "Disturbed" then
		send Msg_State_Of_Input_Channels(InputChannelState.Switched_On,InputChannelState.Disturbed) to P20out;
		Mem_Last_Reported_Input_Channel_State1 := "Switched_On"; Mem_Last_Reported_Input_Channel_StateN := "Disturbed";
		elseif Input_Channel_State1 = "Disturbed" AND Input_Channel_StateN = "Switched_Off" then
		send Msg_State_Of_Input_Channels(InputChannelState.Disturbed,InputChannelState.Switched_Off) to P20out;
		Mem_Last_Reported_Input_Channel_State1 := "Disturbed"; Mem_Last_Reported_Input_Channel_StateN := "Switched_Off";
		elseif Input_Channel_State1 = "Disturbed" AND Input_Channel_StateN = "Switched_On" then
		send Msg_State_Of_Input_Channels(InputChannelState.Disturbed,InputChannelState.Switched_On) to P20out;
		Mem_Last_Reported_Input_Channel_State1 := "Disturbed"; Mem_Last_Reported_Input_Channel_StateN := "Switched_On";
		end if
			cOp2_Initial_Report_Status_Input_Channel_States	Basic IO

Eu.IO.7487	Info	F_SCI_IO_Report - Behaviour		Basic IO

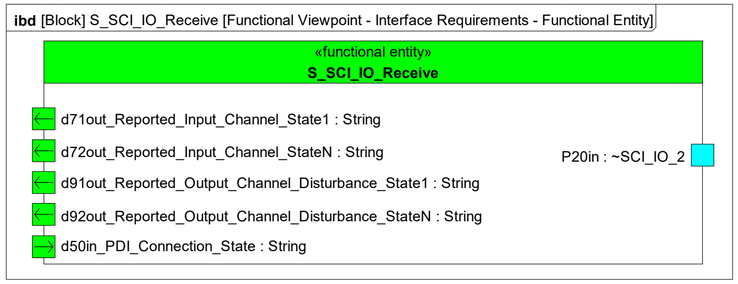
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7531	Req	Functional Viewpoint - Interface Requirements - Functional Entity 2.2 stm [State Machine] F_SCI_IO_Report - Behaviour [Functional Viewpoint - Interface Requirements - Functional Entity 2.2] when(d50in_PDI_Connection_State = "READY_FOR_PDI_NO_SCP" OR d50in_PDI_Connection_State = "READY_FOR_PDI" OR d50in_PDI_Connection_State = "SUSPENDED")/ INTERFACE_CONNECTION_NOT_ESTABLISHED Initial0 Start_Status_Report/ cOp1_Initial_Report_Status_Output_Channel_States (d9in_Monitored_Output_Channel_Disturbance_State1, d9in_Monitored_Output_Channel_Disturbance_StateN) ; cOp2_Initial_Report_Status_Input_Channel_States (d7in_Detected_Input_Channel_State1, d7in_Detected_Input_Channel_StateN) ; REPORTING_LOGICAL_CHANNEL_STATES Entry/send Status_Report_Completed to p3inout; REPORTING_LOGICAL_INPUT_CHANNEL_STATES Initial1 SENDING_INPUT_CHANNEL_REPORTS when(d50in_PDI_Connection_State = "ESTABLISHED")[d7in_Detected_Input_Channel_State1 <> Mem_Last_Reported_Input_Channel_State1 OR d7in_Detected_Input_Channel_StateN <> Mem_Last_Reported_Input_Channel_StateN]/ cOp2_Initial_Report_Status_Input_Channel_States (d7in_Detected_Input_Channel_State1, d7in_Detected_Input_Channel_StateN) ; when(d7in_Detected_Input_Channel_State1 = "Switched_On" AND d7in_Detected_Input_Channel_StateN = "Switched_On")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Input_Channels (Switched_On, Switched_On) to P20out; when(d7in_Detected_Input_Channel_State1 = "Switched_Off" AND d7in_Detected_Input_Channel_StateN = "Switched_Off")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Input_Channels (Switched_Off, Switched_Off) to P20out; when(d7in_Detected_Input_Channel_State1 = "Disturbed" AND d7in_Detected_Input_Channel_StateN = "Disturbed")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Input_Channels (Disturbed, Disturbed) to P20out; when(d7in_Detected_Input_Channel_State1 = "Switched_Off" AND d7in_Detected_Input_Channel_StateN = "Switched_On")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Input_Channels (Switched_Off, Switched_On) to P20out; when(d7in_Detected_Input_Channel_State1 = "Switched_On" AND d7in_Detected_Input_Channel_StateN = "Switched_Off")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Input_Channels (Switched_On, Switched_Off) to P20out; when(d7in_Detected_Input_Channel_State1 = "Switched_Off" AND d7in_Detected_Input_Channel_StateN = "Disturbed")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Input_Channels (Switched_Off, Disturbed) to P20out; when(d7in_Detected_Input_Channel_State1 = "Switched_On" AND d7in_Detected_Input_Channel_StateN = "Disturbed")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Input_Channels (Switched_On, Disturbed) to P20out; when(d7in_Detected_Input_Channel_State1 = "Disturbed" AND d7in_Detected_Input_Channel_StateN = "Switched_Off")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Input_Channels (Disturbed, Switched_Off) to P20out; when(d7in_Detected_Input_Channel_State1 = "Disturbed" AND d7in_Detected_Input_Channel_StateN = "Switched_On")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Input_Channels (Disturbed, Switched_On) to P20out; REPORTING_LOGICAL_OUTPUT_CHANNEL_STATES Initial2 SENDING_OUTPUT_CHANNEL_REPORTS when(d50in_PDI_Connection_State = "ESTABLISHED")[d9in_Monitored_Output_Channel_Disturbance_State1 <> Mem_Last_Reported_Output_Channel_State1 OR d9in_Monitored_Output_Channel_Disturbance_StateN <> Mem_Last_Reported_Output_Channel_StateN]/ cOp1_Initial_Report_Status_Output_Channel_States (d9in_Monitored_Output_Channel_Disturbance_State1, d9in_Monitored_Output_Channel_Disturbance_StateN) ; when(d9in_Monitored_Output_Channel_Disturbance_State1 = "Not_Physically_Disturbed" AND d9in_Monitored_Output_Channel_Disturbance_StateN = "Not_Physically_Disturbed")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Output_Channels (Not_Physically_Disturbed, Not_Physically_Disturbed) to P20out; when(d9in_Monitored_Output_Channel_Disturbance_State1 = "Physically_Disturbed" AND d9in_Monitored_Output_Channel_Disturbance_StateN = "Not_Physically_Disturbed")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Output_Channels (Physically_Disturbed, Not_Physically_Disturbed) to P20out; when(d9in_Monitored_Output_Channel_Disturbance_State1 = "Not_Physically_Disturbed" AND d9in_Monitored_Output_Channel_Disturbance_StateN = "Physically_Disturbed")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Output_Channels (Not_Physically_Disturbed, Physically_Disturbed) to P20out; when(d9in_Monitored_Output_Channel_Disturbance_State1 = "Physically_Disturbed" AND d9in_Monitored_Output_Channel_Disturbance_StateN = "Physically_Disturbed")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Output_Channels (Physically_Disturbed, Physically_Disturbed) to P20out;	This state machine diagram describes the requirements for the following functionalities: - receives the input channel reports and forwards it to the internal logic - receives the output channel reports and forwards it to the internal logic	Basic IO
Eu.IO.7516	Def	Initial0		Basic IO
Eu.IO.7517	Def	/{Initial0 - INTERFACE_CONNECTION_NOT_ESTABLISHED}		Basic IO
Eu.IO.7859	Def	INTERFACE_CONNECTION_NOT_ESTABLISHED		Basic IO

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7860	Def	Start_Status_Report/ cOp1_Initial_Report_Status_Output_Channel_States(d9in_Monitored_Output_Channel_Disturbance_State1,d9in_Monitored_Output_Channel_Disturbance_StateN); cOp2_Initial_Report_Status_Input_Channel_States(d7in_Detected_Input_Channel_State1,d7in_Detected_Input_Channel_StateN);{INTERFACE_CONNECTION_NOT_ESTABLISHED - REPORTING_LOGICAL_CHANNEL_STATES}		Basic IO
Eu.IO.7861	Def	REPORTING_LOGICAL_CHANNEL_STATES		Basic IO
Eu.IO.7862	Def	REPORTING_LOGICAL_INPUT_CHANNEL_STATES		Basic IO
Eu.IO.7866	Def	Initial1		Basic IO
Eu.IO.7867	Def	/{Initial1 - SENDING_INPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7868	Def	SENDING_INPUT_CHANNEL_REPORTS		Basic IO
Eu.IO.7869	Def	when(d50in_PDI_Connection_State = "ESTABLISHED")[d7in_Detected_Input_Channel_State1 <> Mem_Last_Reported_Input_Channel_State1 OR d7in_Detected_Input_Channel_StateN <> Mem_Last_Reported_Input_Channel_StateN]/ cOp2_Initial_Report_Status_Input_Channel_States(d7in_Detected_Input_Channel_State1,d7in_Detected_Input_Channel_StateN);{State-internal in SENDING_INPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7870	Def	when(d7in_Detected_Input_Channel_State1 = "Switched_Off" AND d7in_Detected_Input_Channel_StateN = "Switched_On")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Input_Channels(Switched_Off,Switched_On) to P20out;{State-internal in SENDING_INPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7871	Def	when(d7in_Detected_Input_Channel_State1 = "Disturbed" AND d7in_Detected_Input_Channel_StateN = "Switched_Off")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Input_Channels(Disturbed,Switched_Off) to P20out;{State-internal in SENDING_INPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7872	Def	when(d7in_Detected_Input_Channel_State1 = "Disturbed" AND d7in_Detected_Input_Channel_StateN = "Switched_On")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Input_Channels(Disturbed,Switched_On) to P20out;{State-internal in SENDING_INPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7873	Def	when(d7in_Detected_Input_Channel_State1 = "Disturbed" AND d7in_Detected_Input_Channel_StateN = "Disturbed")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Input_Channels(Disturbed,Disturbed) to P20out;{State-internal in SENDING_INPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7874	Def	when(d7in_Detected_Input_Channel_State1 = "Switched_Off" AND d7in_Detected_Input_Channel_StateN = "Switched_Off")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Input_Channels(Switched_Off,Switched_Off) to P20out;{State-internal in SENDING_INPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7875	Def	when(d7in_Detected_Input_Channel_State1 = "Switched_Off" AND d7in_Detected_Input_Channel_StateN = "Disturbed")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Input_Channels(Switched_Off,Disturbed) to P20out;{State-internal in SENDING_INPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7876	Def	when(d7in_Detected_Input_Channel_State1 = "Switched_On" AND d7in_Detected_Input_Channel_StateN = "Disturbed")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Input_Channels(Switched_On,Disturbed) to P20out;{State-internal in SENDING_INPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7877	Def	when(d7in_Detected_Input_Channel_State1 = "Switched_On" AND d7in_Detected_Input_Channel_StateN = "Switched_On")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Input_Channels(Switched_On,Switched_On) to P20out;{State-internal in SENDING_INPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7878	Def	when(d7in_Detected_Input_Channel_State1 = "Switched_On" AND d7in_Detected_Input_Channel_StateN = "Switched_Off")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Input_Channels(Switched_On,Switched_Off) to P20out;{State-internal in SENDING_INPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7879	Def	REPORTING_LOGICAL_OUTPUT_CHANNEL_STATES		Basic IO
Eu.IO.7880	Def	Initial2		Basic IO
Eu.IO.7881	Def	/{Initial2 - SENDING_OUTPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7882	Def	SENDING_OUTPUT_CHANNEL_REPORTS		Basic IO
Eu.IO.7883	Def	when(d50in_PDI_Connection_State = "ESTABLISHED")[d9in_Monitored_Output_Channel_Disturbance_State1 <> Mem_Last_Reported_Output_Channel_State1 OR d9in_Monitored_Output_Channel_Disturbance_StateN <> Mem_Last_Reported_Output_Channel_StateN]/ cOp1_Initial_Report_Status_Output_Channel_States(d9in_Monitored_Output_Channel_Disturbance_State1,d9in_Monitored_Output_Channel_Disturbance_StateN);{State-internal in SENDING_OUTPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7884	Def	when(d9in_Monitored_Output_Channel_Disturbance_State1 = "Not_Physically_Disturbed" AND d9in_Monitored_Output_Channel_Disturbance_StateN = "Physically_Disturbed")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Output_Channels(Not_Physically_Disturbed,Physically_Disturbed) to P20out;{State-internal in SENDING_OUTPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7885	Def	when(d9in_Monitored_Output_Channel_Disturbance_State1 = "Not_Physically_Disturbed" AND d9in_Monitored_Output_Channel_Disturbance_StateN = "Not_Physically_Disturbed")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Output_Channels(Not_Physically_Disturbed,Not_Physically_Disturbed) to P20out;{State-internal in SENDING_OUTPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7886	Def	when(d9in_Monitored_Output_Channel_Disturbance_State1 = "Physically_Disturbed" AND d9in_Monitored_Output_Channel_Disturbance_StateN = "Physically_Disturbed")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Output_Channels(Physically_Disturbed,Physically_Disturbed) to P20out;{State-internal in SENDING_OUTPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7887	Def	when(d9in_Monitored_Output_Channel_Disturbance_State1 = "Physically_Disturbed" AND d9in_Monitored_Output_Channel_Disturbance_StateN = "Not_Physically_Disturbed")[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Msg_State_Of_Output_Channels(Physically_Disturbed,Not_Physically_Disturbed) to P20out;{State-internal in SENDING_OUTPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7889	Def	when(d50in_PDI_Connection_State = "READY_FOR_PDI_NO_SCP" OR d50in_PDI_Connection_State = "READY_FOR_PDI" OR d50in_PDI_Connection_State = "SUSPENDED")/{REPORTING_LOGICAL_CHANNEL_STATES - INTERFACE_CONNECTION_NOT_ESTABLISHED}		Basic IO
Eu.IO.7888	Def	entry/send Status_Report_Completed to p3inout;{State-internal in REPORTING_LOGICAL_CHANNEL_STATES}		Basic IO
Eu.IO.7897	Info	S_SCI_IO_Command		Basic IO

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7898	Req	[Block] S_SCI_IO_Command [Functional Viewpoint - Interface Requirements - Functional Entity] **ibd** [Block] S_SCI_IO_Command [Functional Viewpoint - Interface Requirements - Functional Entity] «functional entity» **S_SCI_IO_Command** t30in_New_Output_Channel_State_Required : PulsedIn d31in_Required_Channel_State1 : String d32in_Required_Channel_StateN : String d50in_PDI_Connection_State : String P10out : ~SCI_IO_1		Basic IO
Eu.IO.7902	Def	d50in_PDI_Connection_State		Basic IO
Eu.IO.7917	Def	t30in_New_Output_Channel_State_Required		Basic IO
Eu.IO.7900	Def	d31in_Required_Channel_State1		Basic IO
Eu.IO.7901	Def	d32in_Required_Channel_StateN		Basic IO
Eu.IO.7899	Def	P10out	The port P10out exchanges information objects according to SCI_IO_1.	Basic IO
Eu.IO.7903	Info	S_SCI_IO_Command - Behaviour		Basic IO

Eu.IO.7904	Req	Functional Viewpoint - Interface Requirements - Functional Entity STD 1.1
		stm [State Machine] S_SCI_IO_Command - Behaviour [Functional Viewpoint - Interface Requirements - Functional Entity STD 1.1]
		Initial0
		SENDING_OUTPUT_CHANNEL_COMMAND
		when(t30in_New_Output_Channel_State_Required)[d31in_Required_Channel_State1 = "Switched_Off" AND d32in_Required_Channel_StateN = "Switched_Off" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Set_Output_Channels(Switched_Off,Switched_Off) to P10out;
		when(t30in_New_Output_Channel_State_Required)[d31in_Required_Channel_State1 = "Switched_Off" AND d32in_Required_Channel_StateN = "Switched_On" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Set_Output_Channels(Switched_Off,Switched_On) to P10out;
		when(t30in_New_Output_Channel_State_Required)[d31in_Required_Channel_State1 = "Switched_Off" AND d32in_Required_Channel_StateN = "Flashing" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Set_Output_Channels(Switched_Off,Flashing) to P10out;
		when(t30in_New_Output_Channel_State_Required)[d31in_Required_Channel_State1 = "Switched_On" AND d32in_Required_Channel_StateN = "Switched_On" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Set_Output_Channels(Switched_On,Switched_On) to P10out;
		when(t30in_New_Output_Channel_State_Required)[d31in_Required_Channel_State1 = "Switched_On" AND d32in_Required_Channel_StateN = "Switched_Off" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Set_Output_Channels(Switched_On,Switched_Off) to P10out;
		when(t30in_New_Output_Channel_State_Required)[d31in_Required_Channel_State1 = "Switched_On" AND d32in_Required_Channel_StateN = "Flashing" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Set_Output_Channels(Switched_On,Flashing) to P10out;
		when(t30in_New_Output_Channel_State_Required)[d31in_Required_Channel_State1 = "Flashing" AND d32in_Required_Channel_StateN = "Flashing" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Set_Output_Channels(Flashing,Flashing) to P10out;
		when(t30in_New_Output_Channel_State_Required)[d31in_Required_Channel_State1 = "Flashing" AND d32in_Required_Channel_StateN = "Switched_On" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Set_Output_Channels(Flashing,Switched_On) to P10out;
		when(t30in_New_Output_Channel_State_Required)[d31in_Required_Channel_State1 = "Flashing" AND d32in_Required_Channel_StateN = "Switched_Off" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Set_Output_Channels(Flashing,Switched_Off) to P10out;
			This state machine diagram describes the requirements for the following functionalities:
			
			- receives the Output Channel states to be set from internal logic and commands this to the Subsystem - Generic IO
				Basic IO

Eu.IO.7905	Def	Initial0		Basic IO
Eu.IO.7906	Def	/{Initial0 - SENDING_OUTPUT_CHANNEL_COMMAND}		Basic IO
Eu.IO.7907	Def	SENDING_OUTPUT_CHANNEL_COMMAND		Basic IO
Eu.IO.7908	Def	when(t30in_New_Output_Channel_State_Required)[d31in_Required_Channel_State1 = "Switched_Off" AND d32in_Required_Channel_StateN = "Switched_Off" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Set_Output_Channels(Switched_Off,Switched_Off) to P10out;{State-internal in SENDING_OUTPUT_CHANNEL_COMMAND}		Basic IO
Eu.IO.7909	Def	when(t30in_New_Output_Channel_State_Required)[d31in_Required_Channel_State1 = "Switched_Off" AND d32in_Required_Channel_StateN = "Switched_On" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Set_Output_Channels(Switched_Off,Switched_On) to P10out; {State-internal in SENDING_OUTPUT_CHANNEL_COMMAND}		Basic IO
Eu.IO.7910	Def	when(t30in_New_Output_Channel_State_Required)[d31in_Required_Channel_State1 = "Switched_Off" AND d32in_Required_Channel_StateN = "Flashing" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Set_Output_Channels(Switched_Off,Flashing) to P10out; {State-internal in SENDING_OUTPUT_CHANNEL_COMMAND}		Option flashing
Eu.IO.7911	Def	when(t30in_New_Output_Channel_State_Required)[d31in_Required_Channel_State1 = "Switched_On" AND d32in_Required_Channel_StateN = "Switched_On" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Set_Output_Channels(Switched_On,Switched_On) to P10out; {State-internal in SENDING_OUTPUT_CHANNEL_COMMAND}		Basic IO
Eu.IO.7912	Def	when(t30in_New_Output_Channel_State_Required)[d31in_Required_Channel_State1 = "Switched_On" AND d32in_Required_Channel_StateN = "Switched_Off" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Set_Output_Channels(Switched_On,Switched_Off) to P10out; {State-internal in SENDING_OUTPUT_CHANNEL_COMMAND}		Basic IO
Eu.IO.7913	Def	when(t30in_New_Output_Channel_State_Required)[d31in_Required_Channel_State1 = "Switched_On" AND d32in_Required_Channel_StateN = "Flashing" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Set_Output_Channels(Switched_On,Flashing) to P10out; {State-internal in SENDING_OUTPUT_CHANNEL_COMMAND}		Option flashing
Eu.IO.7914	Def	when(t30in_New_Output_Channel_State_Required)[d31in_Required_Channel_State1 = "Flashing" AND d32in_Required_Channel_StateN = "Flashing" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Set_Output_Channels(Flashing,Flashing) to P10out; {State-internal in SENDING_OUTPUT_CHANNEL_COMMAND}		Option flashing

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7915	Def	when(t30in_New_Output_Channel_State_Required)[d31in_Required_Channel_State1 = "Flashing" AND d32in_Required_Channel_StateN = "Switched_On" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Set_Output_Channels(Flashing,Switched_On) to P10out; {State-internal in SENDING_OUTPUT_CHANNEL_COMMAND}		Option flashing
Eu.IO.7916	Def	when(t30in_New_Output_Channel_State_Required)[d31in_Required_Channel_State1 = "Flashing" AND d32in_Required_Channel_StateN = "Switched_Off" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Set_Output_Channels(Flashing,Switched_Off) to P10out; {State-internal in SENDING_OUTPUT_CHANNEL_COMMAND}		Option flashing
Eu.IO.7561	Info	S_SCI_IO_Receive		Basic IO
Eu.IO.7599	Req	[Block] S_SCI_IO_Receive [Functional Viewpoint - Interface Requirements - Functional Entity]  		Basic IO
Eu.IO.7563	Def	d50in_PDI_Connection_State		Basic IO
Eu.IO.7570	Def	d91out_Reported_Output_Channel_Disturbance_State1		Basic IO
Eu.IO.7920	Def	d92out_Reported_Output_Channel_Disturbance_StateN		Basic IO
Eu.IO.7566	Def	d71out_Reported_Input_Channel_State1		Basic IO
Eu.IO.7919	Def	d72out_Reported_Input_Channel_StateN		Basic IO
Eu.IO.7918	Def	P20in	The port P20in exchanges information objects according to SCI_IO_2.	Basic IO
Eu.IO.7576	Info	S_SCI_IO_Receive - Behaviour		Basic IO

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7584	Req	Functional Viewpoint - Interface Requirements - Functional Entity STD 1.2 stm [State Machine] S_SCI_IO_Receive - Behaviour [Functional Viewpoint - Interface Requirements - Functional Entity STD 1.2] Initial0 when(d50in_PDI_Connection_State = "DISCONNECTED" OR d50in_PDI_Connection_State = "IMPERMISSIBLE" OR d50in_PDI_Connection_State = "SUSPENDED" OR d50in_PDI_Connection_State = "REQUESTED_NO_SCP")/ RECEIVING_INPUT_AND_OUTPUT_CHANNEL_STATES RECEIVING_INPUT_CHANNEL_STATES Initial1 RECEIVING_INPUT_CHANNEL_REPORTS Entry/d71out_Reported_Input_Channel_State1 := "Unknown"; d72out_Reported_Input_Channel_StateN := "Unknown"; Msg_State_Of_Input_Channels[ResportedInputState1 = Switched_On AND ResportedInputStateN = Switched_On]/ d71out_Reported_Input_Channel_State1 := "Switched_On"; d72out_Reported_Input_Channel_StateN := "Switched_On"; Msg_State_Of_Input_Channels[ResportedInputState1 = Switched_Off AND ResportedInputStateN = Switched_Off]/ d71out_Reported_Input_Channel_State1 := "Switched_Off"; d72out_Reported_Input_Channel_StateN := "Switched_Off"; Msg_State_Of_Input_Channels[ResportedInputState1 = Disturbed AND ResportedInputStateN = Disturbed]/ d71out_Reported_Input_Channel_State1 := "Disturbed"; d72out_Reported_Input_Channel_StateN := "Disturbed"; Msg_State_Of_Input_Channels[ResportedInputState1 = Switched_Off AND ResportedInputStateN = Switched_On]/ d71out_Reported_Input_Channel_State1 := "Switched_Off"; d72out_Reported_Input_Channel_StateN := "Switched_On"; Msg_State_Of_Input_Channels[ResportedInputState1 = Switched_On AND ResportedInputStateN = Switched_Off]/ d71out_Reported_Input_Channel_State1 := "Switched_On"; d72out_Reported_Input_Channel_StateN := "Switched_Off"; Msg_State_Of_Input_Channels[ResportedInputState1 = Switched_Off AND ResportedInputStateN = Disturbed]/ d71out_Reported_Input_Channel_State1 := "Switched_Off"; d72out_Reported_Input_Channel_StateN := "Disturbed"; Msg_State_Of_Input_Channels[ResportedInputState1 = Switched_On AND ResportedInputStateN = Disturbed]/ d71out_Reported_Input_Channel_State1 := "Switched_On"; d72out_Reported_Input_Channel_StateN := "Disturbed"; Msg_State_Of_Input_Channels[ResportedInputState1 = Disturbed AND ResportedInputStateN = Switched_Off]/ d71out_Reported_Input_Channel_State1 := "Disturbed"; d72out_Reported_Input_Channel_StateN := "Switched_Off"; Msg_State_Of_Input_Channels[ResportedInputState1 = Disturbed AND ResportedInputStateN = Switched_On]/ d71out_Reported_Input_Channel_State1 := "Disturbed"; d72out_Reported_Input_Channel_StateN := "Switched_On"; RECEIVING_OUTPUT_CHANNEL_STATES Initial2 RECEIVING_OUTPUT_CHANNEL_REPORTS Entry/d91out_Reported_Output_Channel_Disturbance_State1 := "Unknown"; d92out_Reported_Output_Channel_Disturbance_StateN := "Unknown"; Msg_State_Of_Output_Channels[ResportedDisturbanceState1 = Not_Physically_Disturbed AND ResportedDisturbanceStateN = Not_Physically_Disturbed]/ d91out_Reported_Output_Channel_Disturbance_State1 := "Not_Physically_Disturbed"; d92out_Reported_Output_Channel_Disturbance_StateN := "Not_Physically_Disturbed"; Msg_State_Of_Output_Channels[ResportedDisturbanceState1 = Physically_Disturbed AND ResportedDisturbanceStateN = Physically_Disturbed]/ d91out_Reported_Output_Channel_Disturbance_State1 := "Physically_Disturbed"; d92out_Reported_Output_Channel_Disturbance_StateN := "Physically_Disturbed"; Msg_State_Of_Output_Channels[ResportedDisturbanceState1 = Not_Physically_Disturbed AND ResportedDisturbanceStateN = Physically_Disturbed]/ d91out_Reported_Output_Channel_Disturbance_State1 := "Not_Physically_Disturbed"; d92out_Reported_Output_Channel_Disturbance_StateN := "Physically_Disturbed"; Msg_State_Of_Output_Channels[ResportedDisturbanceState1 = Physically_Disturbed AND ResportedDisturbanceStateN = Not_Physically_Disturbed]/ d91out_Reported_Output_Channel_Disturbance_State1 := "Physically_Disturbed"; d92out_Reported_Output_Channel_Disturbance_StateN := "Not_Physically_Disturbed";	This state machine diagram describes the requirements for the following functionalities: - receives the Input Channel Reports and forwards it to the internal logic - receives the Output Channel Reports and forwards it to the internal logic	Basic IO
Eu.IO.7577	Def	Initial0		Basic IO
Eu.IO.7578	Def	/{Initial0 - RECEIVING_INPUT_AND_OUTPUT_CHANNEL_STATES}		Basic IO
Eu.IO.7585	Def	RECEIVING_INPUT_AND_OUTPUT_CHANNEL_STATES		Basic IO
Eu.IO.7592	Def	RECEIVING_INPUT_CHANNEL_STATES		Basic IO
Eu.IO.7593	Def	Initial1		Basic IO
Eu.IO.7924	Def	/{Initial1 - RECEIVING_INPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7925	Def	RECEIVING_INPUT_CHANNEL_REPORTS		Basic IO
Eu.IO.7926	Def	entry/d71out_Reported_Input_Channel_State1 := "Unknown"; d72out_Reported_Input_Channel_StateN := "Unknown";{State-internal in RECEIVING_INPUT_CHANNEL_REPORTS}		Basic IO

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.7927	Def	Msg_State_Of_Input_Channels[ResportedInputState1 = Disturbed AND ResportedInputStateN = Switched_On]/ d71out_Reported_Input_Channel_State1 := "Disturbed"; d72out_Reported_Input_Channel_StateN := "Switched_On";{State-internal in RECEIVING_INPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7928	Def	Msg_State_Of_Input_Channels[ResportedInputState1 = Switched_On AND ResportedInputStateN = Switched_On]/ d71out_Reported_Input_Channel_State1 := "Switched_On"; d72out_Reported_Input_Channel_StateN := "Switched_On";{State-internal in RECEIVING_INPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7929	Def	Msg_State_Of_Input_Channels[ResportedInputState1 = Switched_Off AND ResportedInputStateN = Switched_Off]/ d71out_Reported_Input_Channel_State1 := "Switched_Off"; d72out_Reported_Input_Channel_StateN := "Switched_Off";{State-internal in RECEIVING_INPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7930	Def	Msg_State_Of_Input_Channels[ResportedInputState1 = Disturbed AND ResportedInputStateN = Disturbed]/ d71out_Reported_Input_Channel_State1 := "Disturbed"; d72out_Reported_Input_Channel_StateN := "Disturbed";{State-internal in RECEIVING_INPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7931	Def	Msg_State_Of_Input_Channels[ResportedInputState1 = Switched_Off AND ResportedInputStateN = Switched_On]/ d71out_Reported_Input_Channel_State1 := "Switched_Off"; d72out_Reported_Input_Channel_StateN := "Switched_On";{State-internal in RECEIVING_INPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7932	Def	Msg_State_Of_Input_Channels[ResportedInputState1 = Switched_On AND ResportedInputStateN = Switched_Off]/ d71out_Reported_Input_Channel_State1 := "Switched_On"; d72out_Reported_Input_Channel_StateN := "Switched_Off";{State-internal in RECEIVING_INPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7933	Def	Msg_State_Of_Input_Channels[ResportedInputState1 = Switched_Off AND ResportedInputStateN = Disturbed]/ d71out_Reported_Input_Channel_State1 := "Switched_Off"; d72out_Reported_Input_Channel_StateN := "Disturbed";{State-internal in RECEIVING_INPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7934	Def	Msg_State_Of_Input_Channels[ResportedInputState1 = Switched_On AND ResportedInputStateN = Disturbed]/ d71out_Reported_Input_Channel_State1 := "Switched_On"; d72out_Reported_Input_Channel_StateN := "Disturbed";{State-internal in RECEIVING_INPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7935	Def	Msg_State_Of_Input_Channels[ResportedInputState1 = Disturbed AND ResportedInputStateN = Switched_Off]/ d71out_Reported_Input_Channel_State1 := "Disturbed"; d72out_Reported_Input_Channel_StateN := "Switched_Off";{State-internal in RECEIVING_INPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7936	Def	RECEIVING_OUTPUT_CHANNEL_STATES		Basic IO
Eu.IO.7937	Def	Initial2		Basic IO
Eu.IO.7938	Def	/{Initial2 - RECEIVING_OUTPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7939	Def	RECEIVING_OUTPUT_CHANNEL_REPORTS		Basic IO
Eu.IO.7940	Def	entry/d91out_Reported_Output_Channel_Disturbance_State1 := "Unknown"; d92out_Reported_Output_Channel_Disturbance_StateN := "Unknown";{State-internal in RECEIVING_OUTPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7941	Def	Msg_State_Of_Output_Channels[ResportedDisturbanceState1 = Physically_Disturbed AND ResportedDisturbanceStateN = Physically_Disturbed]/ d91out_Reported_Output_Channel_Disturbance_State1 := "Physically_Disturbed"; d92out_Reported_Output_Channel_Disturbance_StateN := "Physically_Disturbed";{State-internal in RECEIVING_OUTPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7942	Def	Msg_State_Of_Output_Channels[ResportedDisturbanceState1 = Not_Physically_Disturbed AND ResportedDisturbanceStateN = Physically_Disturbed]/ d91out_Reported_Output_Channel_Disturbance_State1 := "Not_Physically_Disturbed"; d92out_Reported_Output_Channel_Disturbance_StateN := "Physically_Disturbed";{State-internal in RECEIVING_OUTPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7943	Def	Msg_State_Of_Output_Channels[ResportedDisturbanceState1 = Physically_Disturbed AND ResportedDisturbanceStateN = Not_Physically_Disturbed]/ d91out_Reported_Output_Channel_Disturbance_State1 := "Physically_Disturbed"; d92out_Reported_Output_Channel_Disturbance_StateN := "Not_Physically_Disturbed";{State-internal in RECEIVING_OUTPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7944	Def	Msg_State_Of_Output_Channels[ResportedDisturbanceState1 = Not_Physically_Disturbed AND ResportedDisturbanceStateN = Not_Physically_Disturbed]/ d91out_Reported_Output_Channel_Disturbance_State1 := "Not_Physically_Disturbed"; d92out_Reported_Output_Channel_Disturbance_StateN := "Not_Physically_Disturbed";{State-internal in RECEIVING_OUTPUT_CHANNEL_REPORTS}		Basic IO
Eu.IO.7946	Def	when(d50in_PDI_Connection_State = "DISCONNECTED" OR d50in_PDI_Connection_State = "IMPERMISSIBLE" OR d50in_PDI_Connection_State = "SUSPENDED" OR d50in_PDI_Connection_State = "REQUESTED_NO_SCP")/{RECEIVING_INPUT_AND_OUTPUT_CHANNEL_STATES - RECEIVING_INPUT_AND_OUTPUT_CHANNEL_STATES}		Basic IO
Eu.IO.434	Head	**3.4.2 SMI-IO (Subsystem - Maintenance and Data Management)**		
Eu.IO.1695	Info	The generic InformationFlow and the related FlowProperties through the SMI-IO are specified in [Eu.Doc.120].		Basic IO
Eu.IO.433	Head	**3.4.3 SDI-IO (Subsystem - Maintenance and Data Management)**		
Eu.IO.1694	Info	The generic data points through the SDI-IO are specified in [Eu.Doc.94]. The specific data points through the SDI-IO are specified in [Eu.Doc.82].		Basic IO
Eu.IO.7948	Head	**3.4.4 SSI-IO (Subsystem - Security Services Platform)**		
Eu.IO.7949	Info	The generic content through SSI-LS is specified in [SP-SEC-ServSpec].		Basic IO
Eu.IO.409	Head	**3.4.5 IO1 (Basic Data identifier)**		
Eu.IO.1692	Info	The generic InformationFlow and the related FlowProperties through IO1 are specified in [Eu.Doc.20].		Basic IO
Eu.IO.416	Head	**3.4.6 IO5 (Maintainer)**		
Eu.IO.1784	Info	The generic FlowProperties through IO5 are specified in [Eu.Doc.20].		Basic IO
Eu.IO.1205	Info	Maintainer	Definition of the InformationFlow for Maintenance/Operation/Display IO5 (Maintainer).	Basic IO
Eu.IO.1208	Def	Input_ChannelX	Displays the state for each respective physical Input Channel (X > 0).	Basic IO
Eu.IO.1210	Def	Output_ChannelX	Displays the state for each respective physical Output Channel (X > 0).	Basic IO
Eu.IO.410	Head	**3.4.7 IO2 (Adjacent IO Systems)**		

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.411	Info	Adjacent_IO_Systems_O	Definition of the InformationFlow for Control Interface IO2 (Adjacent IO System) (Output Channels).	Basic IO
Eu.IO.412	Def	Set_Output_Channels	Currently set state of the physical Output Channels (Switched On, Switched Off) in Subsystem - Generic IO as a permanent available information.	Basic IO
Eu.IO.413	Head	**3.4.8 IO3 (Adjacent IO Systems)**		
Eu.IO.414	Info	Adjacent_IO_Systems_I	Definition of the InformationFlow for Control Interface IO3 (Adjacent IO System) (Input Channels).	Basic IO
Eu.IO.415	Def	State_Input_Channels	Current state of the physical Input Channels (Switched On, Switched Off, Disturbed) in Adjacent IO System as a permanent available information.	Basic IO
Eu.IO.185	Head	**4 RAMSS Requirements**		
Eu.IO.1606	Info	The requirements for reliability, availability, maintainability, safety and security are specified in [Eu.Doc.20].		Basic IO
Eu.IO.199	Head	**5 Technical requirements**		
Eu.IO.1533	Info	The generic technical requirements are specified in [Eu.Doc.20].		Basic IO
Eu.IO.1422	Head	**5.1 Specific technical interface requirements**		
Eu.IO.1423	Head	**5.1.1 Interface to the Point of Service Signalling (PoS-Signalling)**		
Eu.IO.1424	Req	Via the technical interface **PoS-Signalling** the data of the functional interface "SCI-IO" shall be exchanged with the Subsystem - Electronic Interlocking as specified in [Eu.Doc.92].		Basic IO
Eu.IO.1425	Req	Via the technical interface **PoS-Signalling** the data of the functional interface "SMI-IO" shall be exchanged with the Subsystem - Maintenance and Data Management as specified in [Eu.Doc.76].16		Basic IO
Eu.IO.1426	Req	Via the technical interface **PoS-Signalling** the data of the functional interface "SDI-IO" shall be exchanged with the Subsystem - Maintenance and Data Management as specified in [Eu.Doc.77].		Basic IO
Eu.IO.7986	Req	Via the technical interface **PoS-Signalling** the data of the functional interface "SSI-IO" shall be exchanged with the Subsystem - Security Services Platform as specified in [SP-SEC-ServSpec].		Basic IO
Eu.IO.1463	Head	**5.1.2 Interfaces to Adjacent IO System**		
Eu.IO.1621	Req	The Adjacent IO System is connected to the Subsystem - Generic IO via the technical interfaces IO2 and IO3.		Basic IO
Eu.IO.1622	Req	If the Subsystem - Generic IO provides Input Channels, at least 8 physical Input Channels shall be provided by it.		Basic IO
Eu.IO.1623	Req	These 8 physical Input Channels shall be configurable into 4 or more logical Input Channels.		Basic IO
Eu.IO.1624	Req	If the Subsystem - Generic IO provides Output Channels, at least 8 physical Output Channels shall be provided by it.		Basic IO
Eu.IO.1625	Req	These 8 physical Output Channels shall be configurable into 4 or more logical Output Channels.		Basic IO
Eu.IO.1626	Req	The input types "Antivalent input" and "Equivalent input" shall be implemented with two physical Input Channels.		Basic IO
Eu.IO.1627	Req	The output types "Antivalent output" and "Equivalent output" shall be implemented with two physical Output Channels.		Basic IO
Eu.IO.1628	Req	During the evaluation of an Input Channel, the configured so-called individual "Debouncing time" (see ID Eu.IO.1288) shall be waited separately for switching on and switching off before the monitored state change is evaluated as valid and then reported.		Basic IO
Eu.IO.1629	Req	For Input Channels the message to the Subsystem - Electronic Interlocking about the current state shall be delayed for a configurable time duration ("message delay time", see Eu.IO.1282) calculated from the time of the change and separate for the messages "Switched On" and "Switched Off". Note: Disturbed-messages always shall be sent without waiting for the message delay time.		Basic IO
Eu.IO.1630	Req	For Output Channels the switching commanded by the Subsystem - Electronic Interlocking shall be delayed for a configurable time duration ("activation delay time", see Eu.IO.1281) calculated from the time of receiving the command and separate for the commands "Switched On" and "Switched Off". Note: It is advised to not set an "activation delay time" (value 0 ms) for channels that can be commanded to the state Flashing.		Basic IO
Eu.IO.1631	Info	It shall be possible to connect an Adjacent IO System to different Subsystem - Generic IO, if the needed amount of Input Channels and Output Channels cannot be offered by a single Subsystem - Generic IO.		Basic IO
Eu.IO.1632	Req	Output Channels dependent in time shall always be connected to the same Subsystem - Generic IO.		Basic IO
Eu.IO.1633	Req	Input Channels dependent in time shall always be connected to the same Subsystem - Generic IO.		Basic IO
Eu.IO.1534	Head	**5.1.3 Examples for wiring schemes of Adjacent IO Systems**		
Eu.IO.1535	Info	These examples shall be defined by national specifications. Note: In future phases of the System Pillar, national specifications will be replaced by harmonised specifications.		Basic IO
Eu.IO.1545	Head	**5.2 Time behaviour**		
Eu.IO.1546	Req	The time values defined in the chapter Functional requirements specification (ID Eu.IO.1279) shall be configured of the operation for the Subsystem - Generic IO.		Basic IO
Eu.IO.1547	Head	**5.2.1 Response times**		
Eu.IO.1548	Req	The Subsystem – Generic IO shall send the corresponding message telegram to the Subsystem - Electronic Interlocking within 500 ms after successful change of state, according to the specified UseCases.		Basic IO
Eu.IO.1549	Req	The Subsystem – Generic IO shall switch the output at the IO2 interface within 300 ms after receiving a command telegram and expiry of all delay times.		Basic IO
Eu.IO.1550	Head	**5.3 Configuration and engineering data**		
Eu.IO.1565	Head	**5.3.1 Specific data**		
Eu.IO.7608	Info	The specific configuration and engineering data for the Subsystem – Generic IO shall include as a minimum the following information: Note: This data applies to all Adjacent IO Systems (i.e. all applications).		Basic IO

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.1567	Req	• The connected Adjacent IO Systems which are connected to a Subsystem - Generic IO.		Basic IO
Eu.IO.7609	Req	• For each Input Channel the assigned Adjacent IO System and the assigned type (Antivalent, Equivalent, Single).		Basic IO
Eu.IO.7610	Req	• For each Output Channel the assigned Adjacent IO System and the assigned type (Antivalent, Equivalent, Single).		Basic IO
Eu.IO.1569	Req	• For each logical Output Channel, whether its physical output states shall be monitored. Note: If a logical output channel is not monitored, the fail-safe supervision may be provided by using an Input Channel for confirming the activation of the output.		Basic IO
Eu.IO.8048	Req	• For each Output Channel, the Con_t_Activation_Delay_On and Con_t_Activation_Delay_Off.		Basic IO
Eu.IO.7612	Req	• For each Input Channel the Con_t_Message_Delay_Time_On and Con_t_Message_Delay_Time_Off.		Basic IO
Eu.IO.8049	Req	• For each Input Channel the Con_tmax_Switching_Period.		Basic IO
Eu.IO.8050	Req	• For each Input Channel the Con_t_Debouncing_Time.		Basic IO
Eu.IO.7979	Req	• For each Output Channel, whether it can be commanded to the state Flashing the applicable duty cycle for Flashing.		Option flashing
Eu.IO.7950	Req	• For each Subsystem – Generic IO, the applicable Flashing period.		Option flashing
Eu.IO.629	Req	In the engineering data it shall be defined for the Subsystem - Generic IO whether during the disturbance of an Output Channel either only the physical Output Channels of the affected logical Output Channel or all physical Output Channels of the Adjacent IO System concerned (as far as possible given the disturbance) are Switched Off.		Basic IO
Eu.IO.7613	Info	Two different data sections can be loaded which are the safety-relevant data and the non safety-relevant data. The following definitions apply to the assignment of the sections:		Basic IO
Eu.IO.1560	Req	• The configuration data, such as the IP addresses of the Subsystem - Electronic Interlocking, the value of the diagnostic data points with attribute type 'configuration', is non safety-relevant. This data shall be used to calculate the CSNS.		Basic IO
Eu.IO.1561	Req	• The remaining configuration data is currently categorised as safety-relevant. This data shall be used to calculate the CSS.		Basic IO
Eu.IO.1562	Req	• The engineering data is safety-relevant. This data shall be used to calculate the CSS.		Basic IO
Eu.IO.8065	Head	**5.3.2 Value configuration**		
Eu.IO.8066	Req	**Con_t_Activation_Delay_On** The time value shall be configured in accordance with: Configurable resolution: steps of 50 ms Configurable range: from 0 up to 2000 ms Con_t_Activation_Delay_On is defined in Eu.IO.1281.		Basic IO
Eu.IO.8067	Req	**Con_t_Activation_Delay_Off** The time value shall be configured in accordance with: Configurable resolution: steps of 50 ms Configurable range: from 0 up to 2000 ms Con_t_Activation_Delay_Off is defined in Eu.IO.8038.		Basic IO
Eu.IO.8068	Req	**Con_t_Message_Delay_Time_On** The time value shall be configured in accordance with: Configurable resolution: steps of 50 ms Configurable range: from 0 up to 2000 ms Con_t_Message_Delay_Time_On is defined in Eu.IO.1282.		Basic IO
Eu.IO.8069	Req	**Con_t_Message_Delay_Time_Off** The time value shall be configured in accordance with: Configurable resolution: steps of 50 ms Configurable range: from 0 up to 2000 ms Con_t_Message_Delay_Time_Off is defined in Eu.IO.8039.		Basic IO
Eu.IO.8070	Req	**Con_tmax_Switching_Period** The time value shall be configured in accordance with: Configurable resolution: steps of 50 ms Configurable range: from 0 up to 250 ms Con_tmax_Switching_Period is defined in Eu.IO.1287.		Basic IO

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.IO.8071	Req	**Con_Flash_Duty_Cycle** The time value shall be configured in accordance with: Configurable duty cycles: 1. 75% on, 25% off 2. 50% on, 50% off 3. 25% on, 75% off Con_Flash_Duty_Cycle is defined in Eu.IO.7683.		Option flashing
Eu.IO.8072	Req	**Con_t_Flash_Period** The time value shall be configured in accordance with: Configurable periods (frequencies): 1. 2000 ms (0,5 hertz) 2. 1333 ms (0,75 hertz) 3. 1000 ms (1 hertz) 4. 800 ms (1,25 hertz) The configured frequency shall be valid for all Flashing channels in one Subsystem - Generic IO. Their phases are synchronised to this central frequency. Con_t_Flash_Period is defined in Eu.IO.7684.		Option flashing
Eu.IO.8073	Req	**Con_t_Debouncing_Time** The time value shall be configured in accordance with: Configurable resolution: steps of 50 ms Configurable range: from 0 up to 1000 ms Con_t_Debouncing_Time is defined in Eu.IO.1288.	Note: The resolution of the configurable time value is taken as 10% of the response time of the Subsystem - Generic IO to a change of state of input channels as specified in Eu.IO.1548.	Basic IO