EULYNX Initiative

Europe's Rail Joint Undertaking

# Requirements specification for subsystem Point

Document number: Eu.Doc.36
Version: 4.5 (1.A)

# Contents


**1 Introduction 1**
1.1 Release information 1
1.2 Impressum 2
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 3

**2 Conditions of use 3**
2.1 Functional Packages 3

**3 Functional requirements specification 3**
3.1 Subsystem - Point - General Infos and Assumptions 3
3.2 Subsystem - Point - Logical Viewpoint 3
3.2.1 Subsystem - Point - Logical Context 3
3.3 Subsystem - Point - Functional Viewpoint 4
3.3.1 Definition of time values 4
3.3.2 Subsystem - Point - Functional Context 5
3.3.3 Subsystem - Point - Functional Partitioning 73
3.3.4 Subsystem - Point - Functional Architecture 74
3.3.5 Subsystem - Point - Functional Entities 77
3.4 Subsystem - Point - Interfaces 108
3.4.1 SCI-P (Subsystem - Electronic Interlocking) 108
3.4.1.1 SCI-P - Logical Viewpoint 108
3.4.1.1.1 SCI-P - Logical Context 108
3.4.1.2 SCI-P - Information Flows 108
3.4.1.3 SCI-P - Functional Viewpoint 110
3.4.1.3.1 SCI-P - Functional Partitioning 111
3.4.1.3.2 SCI-P - Functional Architecture 111
3.4.1.3.3 SCI-P - Functional Entities 112
3.4.2 SMI-P (Subsystem - Maintenance and Data Management) 127
3.4.3 SDI-P (Subsystem - Maintenance and Data Management) 128
3.4.4 SSI-P (Subsystem - Security Services Platform) 128
3.4.5 P4 (Basic Data Identifier) 128
3.4.6 P1 (Maintainer) 128
3.4.7 P3 (Point machine) 129
3.4.7.1 Interpretation tables of 4-wire patterns 130

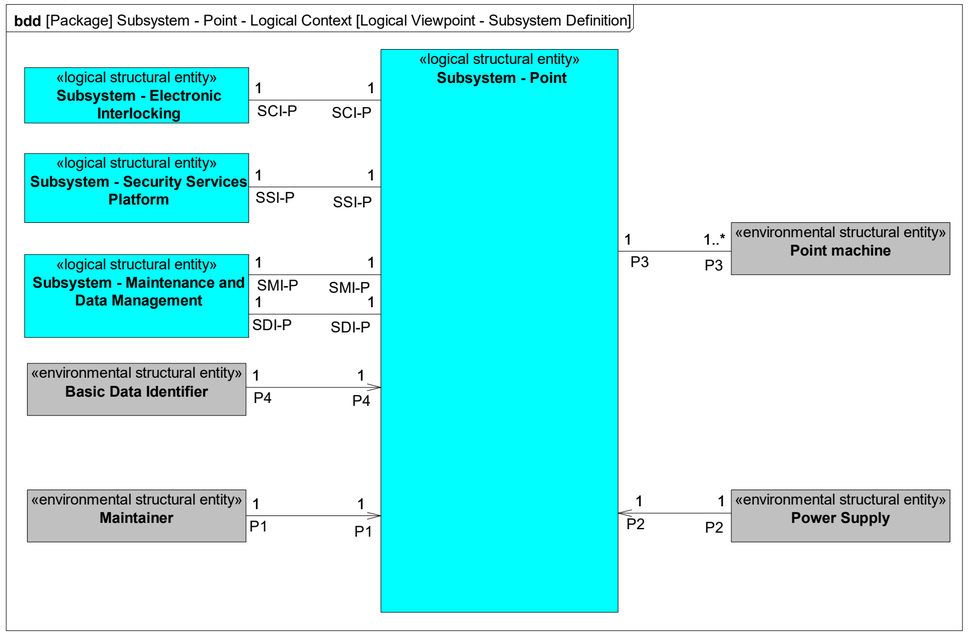
**4 RAMSS requirements 132**

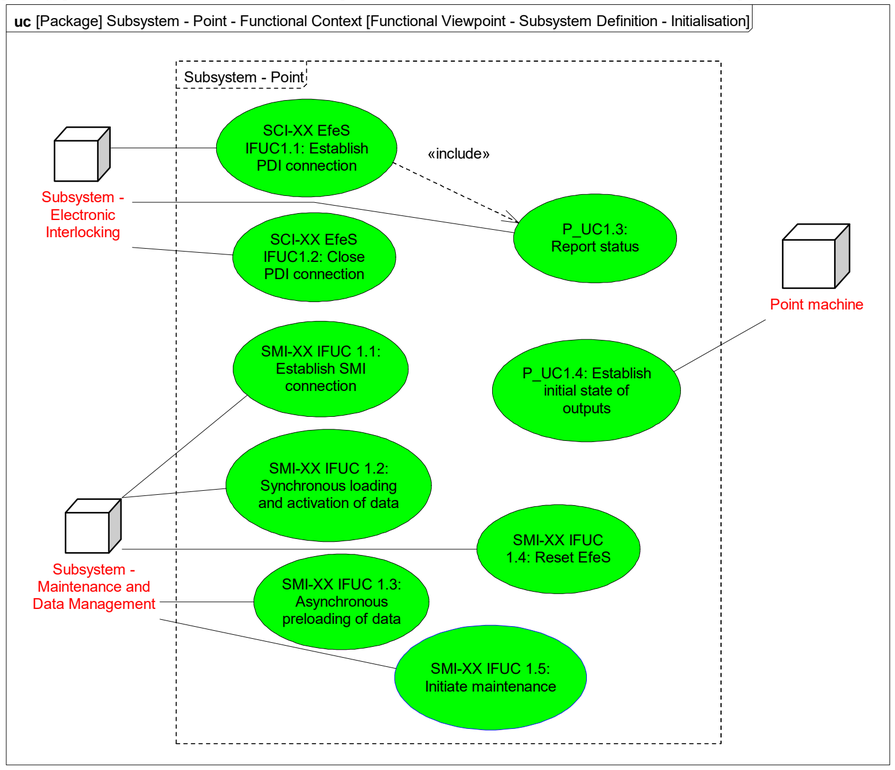
**5 Technical Requirements 132**
5.1 Specific technical interface requirements 132
5.1.1 Interface to the Point of Service Signalling (PoS-Signalling) 132
5.1.2 Interface to the point machine 133
5.2 Time behaviour 133
5.2.1 Response times 134
5.3 Configuration and engineering data 134
5.3.1 Specific data 134
5.3.2 Value configuration 135

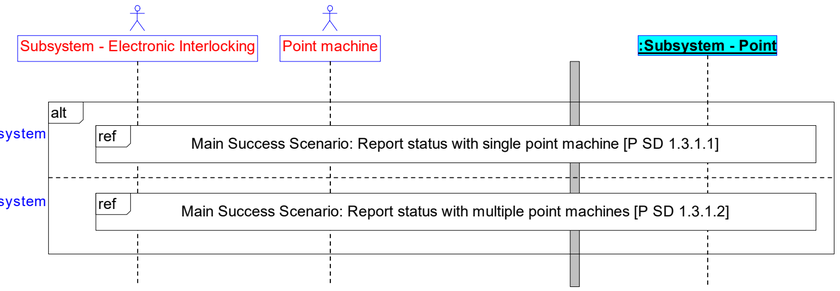
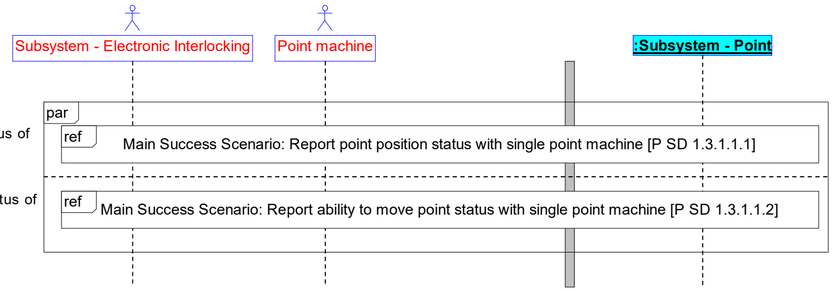
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.1 | Head | **1 Introduction** | | |
| Eu.P.2 | Head | **1.1 Release information** | | |
| Eu.P.3 | Info | [Eu.Doc.36]<br>Requirements specification for subsystem Point<br>CENELEC Phase: 4<br>Version: 4.5 (1.A)<br>Approval date: 02.06.2025 | | |
| Eu.P.3032 | Info | **Version history** | | |
| Eu.P.6591 | Info | version number: 4.0 (0.A)<br>date: 16.05.2022<br>author: Andreas Staudte<br>model version: 18<br>Generic interface and subsystem requirements version: 4.0 (0.A)<br>Generic interface and subsystem requirements for SCI version: 1.0 (0.A)<br>review: CCB<br>changes: EUP-383, EUP-384, EUP-385, EUP-386, EUP-387, EUP-391, EUP-393 | | |
| Eu.P.6701 | Info | version number: 4.1 (0.A)<br>date: 17.04.2023<br>author: Philipp Wolber, Filip Giering<br>model version: 21<br>Generic interface and subsystem requirements version: 4.0 (1.A)<br>Generic interface and subsystem requirements for SCI version: 1.0 (1.A)<br>review:<br>changes: EUP-396, EUP-397, EUP-389, EUP-405, EUP-406, EUP-409, EUP-412, EUP-413, EUP-415, EUP-416, EUP-418, EUP-421, EUP-429, EUP-431, EUP-433, EUP-434, EUP-436, EUP-437, EUP-441, EUP-442, EUP-443, EUP-444, EUP-445, EUP-446, EUP-448, EUP-449, EUP-451, EUP-452 | | |
| Eu.P.7097 | Info | version number: 4.2 (0.A)<br>date: 11.05.2023<br>author: Philipp Wolber, Filip Giering, Dominik Smajgl<br>model version: 22<br>Generic interface and subsystem requirements version: 4.0 (1.A)<br>Generic interface and subsystem requirements for SCI version: 1.0 (1.A)<br>review: cluster<br>changes: EUP-453, EUP-454, EUP-455, EUP-456, EUP-458, EUP-460, EUP-461, EUP-462, EUP-463, EUP-469, EUP-470, EUP-472, EUP-475, EUP-476, EUP-481 | | |
| Eu.P.7153 | Info | version number: 4.2 (1.A)<br>date: 01.06.2023<br>author: Philipp Wolber, Filip Giering, Dominik Smajgl<br>model version: 22<br>Generic interface and subsystem requirements version: 4.0 (1.A)<br>Generic interface and subsystem requirements for SCI version: 1.0 (1.A)<br>review: cluster<br>changes: EUP-457, EUP-459, EUP-467, EUP-468, EUP-471, EUP-473, EUP-478, EUP-479, EUP-482, EUP-483, EUP-488, EUP-489, EUP-491, EUP-493 | | |
| Eu.P.7198 | Info | version number: 4.3 (0.A)<br>date: 28.06.2023<br>author: Philipp Wolber, Filip Giering<br>model version: 22<br>Generic interface and subsystem requirements version: 4.0 (3.A)<br>Generic interface and subsystem requirements for SCI version: 1.0 (3.A)<br>review: TACS Mirror Group<br>changes: EUP-502, EUP-503, EUP-504, EUP-505, EUP-506, EUP-507, EUP-509, EUP-510, EUP-513 | | |
| Eu.P.7241 | Info | version number: 4.3 (1.A)<br>date: 15.12.2023<br>author: Philipp Wolber, Filip Giering<br>model version: 25<br>Generic interface and subsystem requirements version: 4.0 (4.A)<br>Generic interface and subsystem requirements for SCI version: 1.0 (4.A)<br>review: M&T<br>changes: EUP-417, EUP-419, EUP-485, EUP-500, EUP-514, EUP-515, EUP-516, EUP-518, EUP-519, EUP-529, EUP-530, EUP-532, EUP-535, EUP-536, EUP-537, EUP-538, EUP-539, EUP-540, EUP-541, EUP-542, EUP-543, EUP-544 | | |
| Eu.P.7369 | Info | version number: 4.3 (2.A)<br>date: 21.03.2024<br>author: Philipp Wolber, Filip Giering<br>model version: 26<br>Generic interface and subsystem requirements version: 4.0 (4.A)<br>Generic interface and subsystem requirements for SCI version: 1.0 (4.A)<br>review: cluster<br>changes: EUP-486, EUP-496, EUP-499, EUP-501, EUP-531, EUP-533, EUP-545, EUP-546, EUP-547, EUP-548, EUP-550, EUP-551, EUP-552 | | |
| Eu.P.7454 | Info | version number: 4.4 (0.A)<br>date: 18.06.2024<br>author: Philipp Wolber, Filip Giering<br>model version: 26<br>Generic interface and subsystem requirements version: 4.0 (6.A)<br>Generic interface and subsystem requirements for SCI version: 1.1 (0.A)<br>review: TACS Mirror Group<br>changes: EUP-554, EUP-555, EUP-556, EUP-557, EUP-558, EUP-559, EUP-571, EUP-573 | | |

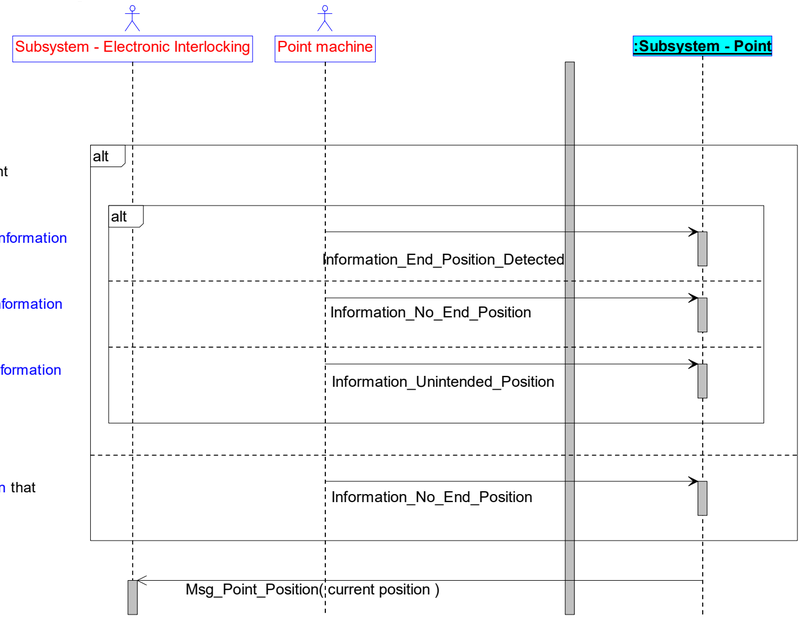
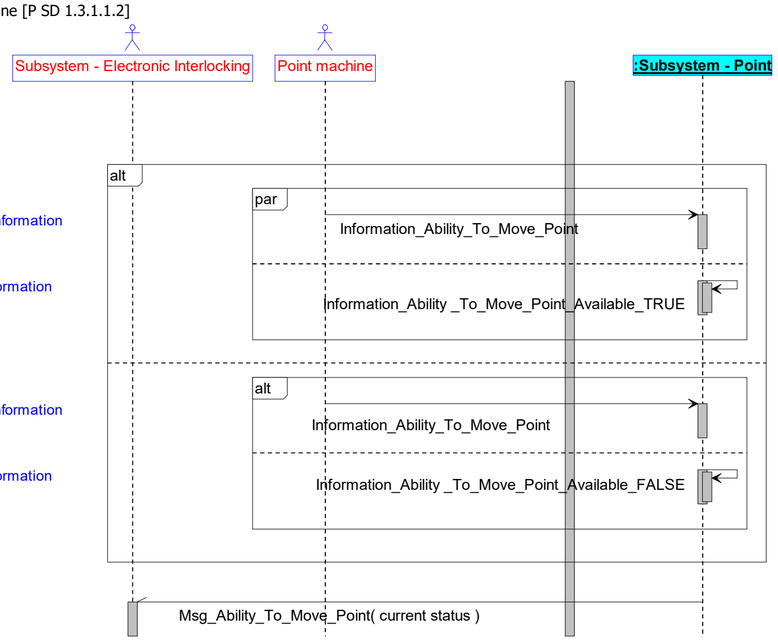
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.7487 | Info | version number: 4.5 (0.A)<br>date: 06.05.2025<br>author: Philipp Wolber<br>model version: 29<br>Generic interface and subsystem requirements version: 4.0 (6.A)<br>Generic interface and subsystem requirements for SCI version: 1.1 (0.A)<br>review:<br>changes: EUP-576, EUP-577, EUP-583 | | |
| Eu.P.7505 | Info | version number: 4.5 (1.A)<br>date: 20.06.2025<br>author: Philipp Wolber<br>model version: 29<br>Generic interface and subsystem requirements version: 4.0 (7.A)<br>Generic interface and subsystem requirements for SCI version: 1.1 (2.A)<br>review: TACS Mirror Group<br>changes: EUP-584, EUP-585 | | |
| Eu.P.7 | Head | **1.2 Impressum** | | |
| Eu.P.8 | Info | Publishers:<br>**Europe's Rail Joint Undertaking**<br>https://rail-research.europa.eu<br><br>**EULYNX Initiative**<br>https://eulynx.eu/ | | |
| Eu.P.9 | Info | Responsible for this document:<br>EU-Rail System Pillar<br>Trackside Assets Control and Supervision domain | | |
| Eu.P.3038 | Info | This document is drafted by and belongs to EU Rail.<br><br>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.<br><br>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].<br><br>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.<br><br>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.P.10 | Head | **1.3 Purpose** | | |
| Eu.P.11 | Info | The purpose of the document is the specification of requirements for the Subsystem - Point. | | |
| Eu.P.12 | Info | This document describes functional, non-functional and technical requirements for the Subsystem - Point and functional requirements for interface SCI-P. | | |
| Eu.P.13 | Info | This document is intended for the following users:<br>• safety authorities<br>• infrastructure managers<br>• safety assessors<br>• signalling system suppliers<br>• validators | | |
| Eu.P.14 | Info | This document is the basis for the implementation by the supplier and for approval by the infrastructure manager. | | |
| Eu.P.7144 | 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.P.15 | Head | **1.4 Applicable standards and regulations** | | |
| Eu.P.314 | Info | The applicable standards and regulations used in EULYNX are listed in the EULYNX Reference Document List [Eu.Doc.12]. | | |
| Eu.P.35 | Head | **1.5 Applicable documents** | | |
| Eu.P.36 | 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.P.51 | Head | **1.6 Terms and abbreviations** | | |
| Eu.P.52 | Info | The terms and abbreviations are listed in the EULYNX Glossary [Eu.Doc.9]. | | |
| Eu.P.1350 | Head | **1.7 Variability management** | | |
| Eu.P.1351 | Info | This document describes harmonised requirements. Variability management is not applicable. | | |
| Eu.P.3024 | Head | **1.8 Definition of object types** | | |
| Eu.P.3025 | Info | The following definition for object types is applied in this document: | | |
| Eu.P.3026 | Info | • "Req" - This denotes a mandatory requirement. | | |
| Eu.P.7246 | Info | • "Def" - This denotes referenceable model elements that are used in the model-based creation of requirements | | |
| Eu.P.3027 | Info | • "Info" - This denotes additional information to help understand the specification. These objects do not specify any additional requirements. | | |

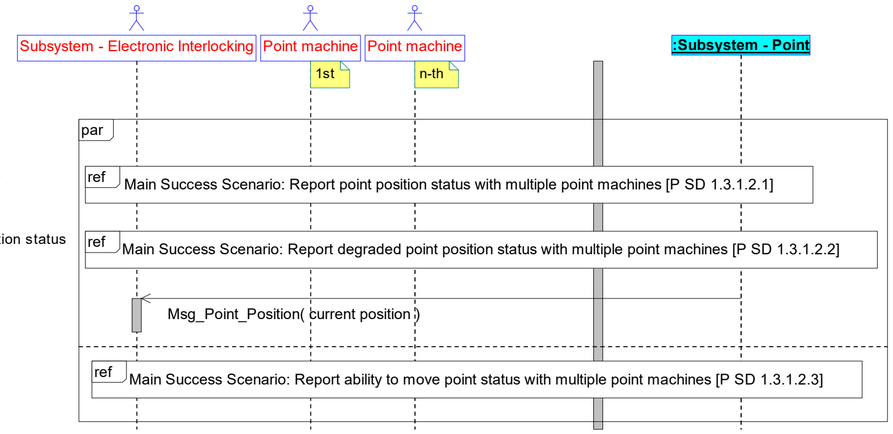
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.3028 | Info | • "Head" - This denotes chapter headings. | | |
| Eu.P.53 | Head | **1.9 Modelling** | | |
| Eu.P.54 | Info | The section "Functional requirements specification" follows a model based systems engineering process using Systems Modelling Language (SysML) and defines the functional system requirements for the Subsystem - Point operational in stimulus-response form. Furthermore the information objects (stimuli and responses) exchanged over the interfaces of the Subsystem - Point are defined. | | |
| Eu.P.55 | Info | The diagrams presented in this document are modelled in SysML [SysML]. | | |
| Eu.P.3050 | Info | The rules for the interpretation of the model based parts of specification are defined in [Eu.Doc.29]. | | |
| Eu.P.3051 | 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.P.3052 | 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.P.3053 | 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.P.3054 | 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.P.7247 | 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.P.57 | Head | **2 Conditions of use** | | |
| Eu.P.4952 | Req | All references to [Eu.Doc.20] refer to version 4.0 of that document. | | |
| Eu.P.6374 | Req | All references to [Eu.Doc.119] refer to version 1.1 of that document. | | |
| Eu.P.6375 | 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.<br>Note: In future phases of the System Pillar, national specifications will be replaced by harmonised specifications. | | |
| Eu.P.58 | Info | The specifications defined in this document shall follow the requirements of the EULYNX System Architecture Specification [Eu.Doc.16]. | | |
| Eu.P.6159 | Head | **2.1 Functional Packages** | | |
| Eu.P.6160 | Info | The specifications in this document are divided into functional packages. There are two types of packages related to the product capabilities. | | |
| Eu.P.6161 | 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.P.6162 | Info | 'Optional package': One or more packages that can be optionally implemented in addition to one or more basic packages. | | |
| Eu.P.6163 | Info | The specifications of the Subsystem – Point are divided into the following functional packages: | | |
| Eu.P.6164 | Info | Subsystem Point for a single point machine 4 wire (basic package) [Basic 4-wire single P]. | | |
| Eu.P.7154 | Info | Subsystem Point for a single point machine non 4 wire (basic package) [Basic non-4-wire single P]. | | |
| Eu.P.6165 | Info | Subsystem Point for multiple point machines 4 wire (basic package) [Basic 4-wire multiple P]. | | |
| Eu.P.7155 | Info | Subsystem Point for multiple point machines non 4 wire (basic package) [Basic non-4-wire multiple P]. | | |
| Eu.P.6166 | Info | Ability to Move functionality (optional package) [Option Able to move].<br>Note: This optional functional package can be combined with any of the 4 basic functional packages. | | |
| Eu.P.7156 | Info | Redrive functionality (optional package) [Option Redrive].<br>Note: This optional functional package can be combined only with the basic packages [Basic non-4-wire single P] and [Basic non-4-wire multiple P] | | |
| Eu.P.7157 | Info | Common Drive functionality (optional package) [Option Common Drive].<br>Note: This optional functional package can be combined only with the basic package [Basic non-4-wire multiple P]. | | |
| Eu.P.884 | Head | **3 Functional requirements specification** | | |
| Eu.P.6039 | Head | **3.1 Subsystem - Point - General Infos and Assumptions** | | |
| Eu.P.6292 | Info | The defined model elements represent the Subsystem - Point in a general way. This refers to:<br>• The functional architectures shown in the internal block diagrams<br>• The defined number of Point machines in the state diagrams<br>• The timing behaviour related to individual Point Machines. | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6158 | Head | **3.2 Subsystem - Point - Logical Viewpoint** | | |
| Eu.P.6282 | Head | **3.2.1 Subsystem - Point - Logical Context** | | |

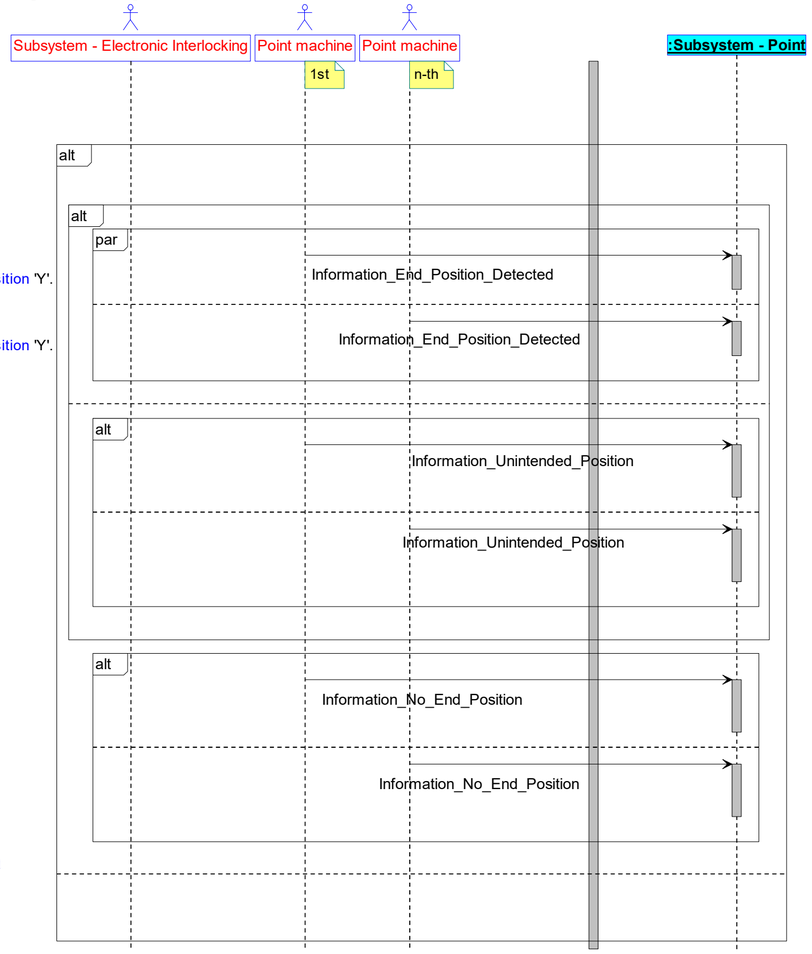
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.950 | Def | [Package] Subsystem - Point - Logical Context [Logical Viewpoint - Subsystem Definition]<br>  <br> | | Basic 4-wire single P<br>Basic 4-wire multiple P<br>Basic non-4-wire single P<br>Basic non-4-wire multiple P |
| Eu.P.7479 | Req | The Subsystem - Point shall provide a logical interface SCI-P to exactly one Subsystem - Electronic Interlocking. | | |
| Eu.P.7482 | Req | The Subsystem - Point shall provide a logical interface SSI-P to exactly one Subsystem - Security Services Platform. | | |
| Eu.P.7480 | Req | The Subsystem - Point shall provide a logical interface SMI-P to exactly one Subsystem - Maintenance and Data Management. | | |
| Eu.P.7481 | Req | The Subsystem - Point shall provide a logical interface SDI-P to exactly one Subsystem - Maintenance and Data Management. | | |
| Eu.P.7483 | Req | The Subsystem - Point shall provide a logical interface P4 to exactly one Basic Data identifier. | | |
| Eu.P.7484 | Req | The Subsystem - Point shall provide a logical interface P1 to exactly one Maintainer. | | |
| Eu.P.7486 | Req | The Subsystem - Point shall provide a logical interface P3 to at least one Point machine. | | |
| Eu.P.7485 | Req | The Subsystem - Point shall provide a logical interface P2 to exactly one Power Supply. | | |
| Eu.P.5780 | Head | **3.3 Subsystem - Point - Functional Viewpoint** | | |
| Eu.P.2286 | Head | **3.3.1 Definition of time values** | | |
| Eu.P.3068 | Info | The generic time values for SCI are specified in [Eu.Doc.119]. | | Basic 4-wire single P<br>Basic 4-wire multiple P<br>Basic non-4-wire single P<br>Basic non-4-wire multiple P |
| Eu.P.6376 | Info | The generic time values for SMI are specified in [Eu.Doc.120]. | | Basic 4-wire single P<br>Basic 4-wire multiple P<br>Basic non-4-wire single P<br>Basic non-4-wire multiple P |

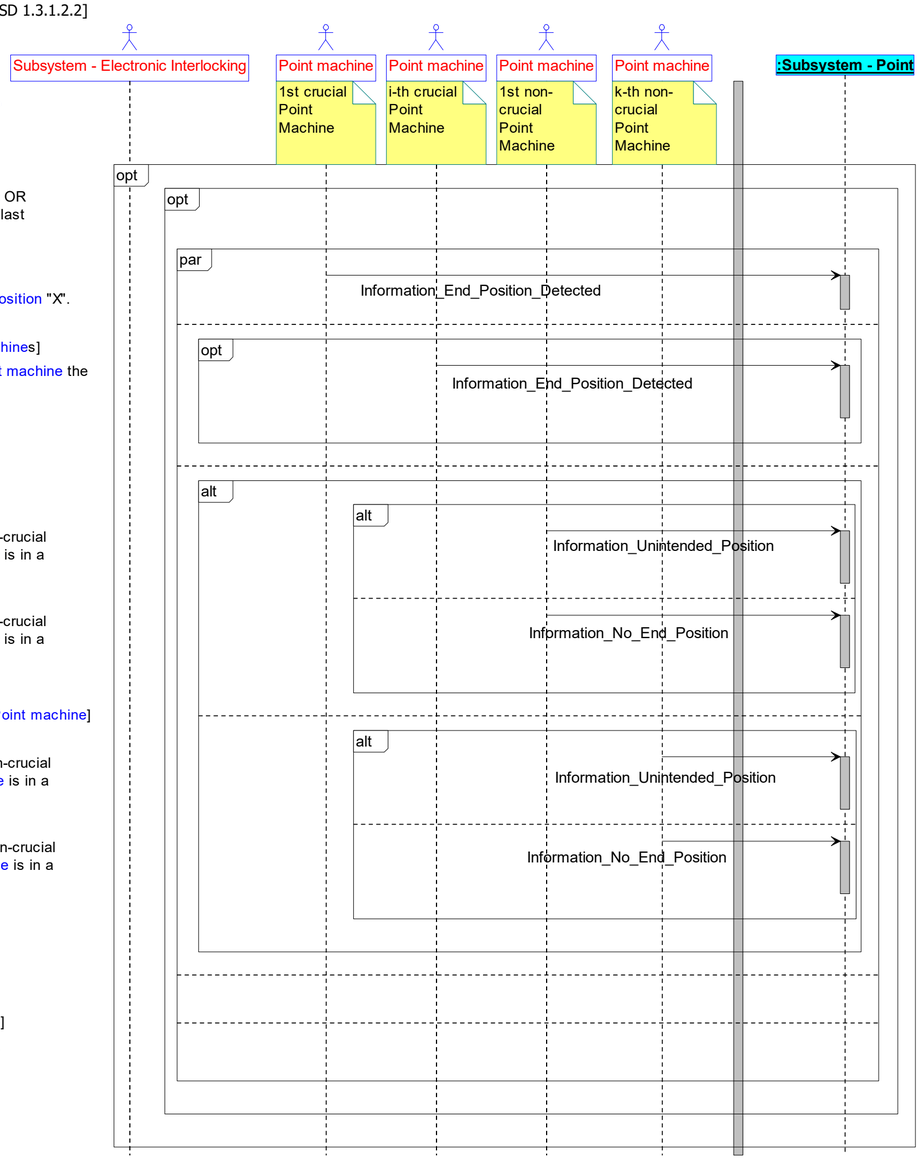
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.2439 | Def | Con_tmax_Point_Operation | Con_tmax_Point_Operation defines the max. time period the Point has to arrive to an End position, starting with the command Moving to the Point machine.<br>After that time period the command Moving to the Point machine is set to Stop moving. | Basic 4-wire single P<br>Basic 4-wire multiple P<br>Basic non-4-wire single P<br>Basic non-4-wire multiple P |
| Eu.P.5782 | Head | **3.3.2 Subsystem - Point - Functional Context** | | |
| Eu.P.1467 | Info | [Package] Subsystem - Point - Functional Context [Functional Viewpoint - Subsystem Definition - Initialisation]<br>  <br> | | Basic 4-wire single P<br>Basic 4-wire multiple P<br>Basic non-4-wire single P<br>Basic non-4-wire multiple P |
| Eu.P.3070 | 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].<br>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 4-wire single P<br>Basic 4-wire multiple P<br>Basic non-4-wire single P<br>Basic non-4-wire multiple P |
| Eu.P.1465 | Info | P_UC1.3: Report status | The Subsystem-UseCase P_UC1.3: Report status defines a scenario about the transmission of status data of Subsystem - Point to Subsystem - Electronic Interlocking, while Process Data Interface protocol connection is establishing. | Basic 4-wire single P<br>Basic 4-wire multiple P<br>Basic non-4-wire single P<br>Basic non-4-wire multiple P |

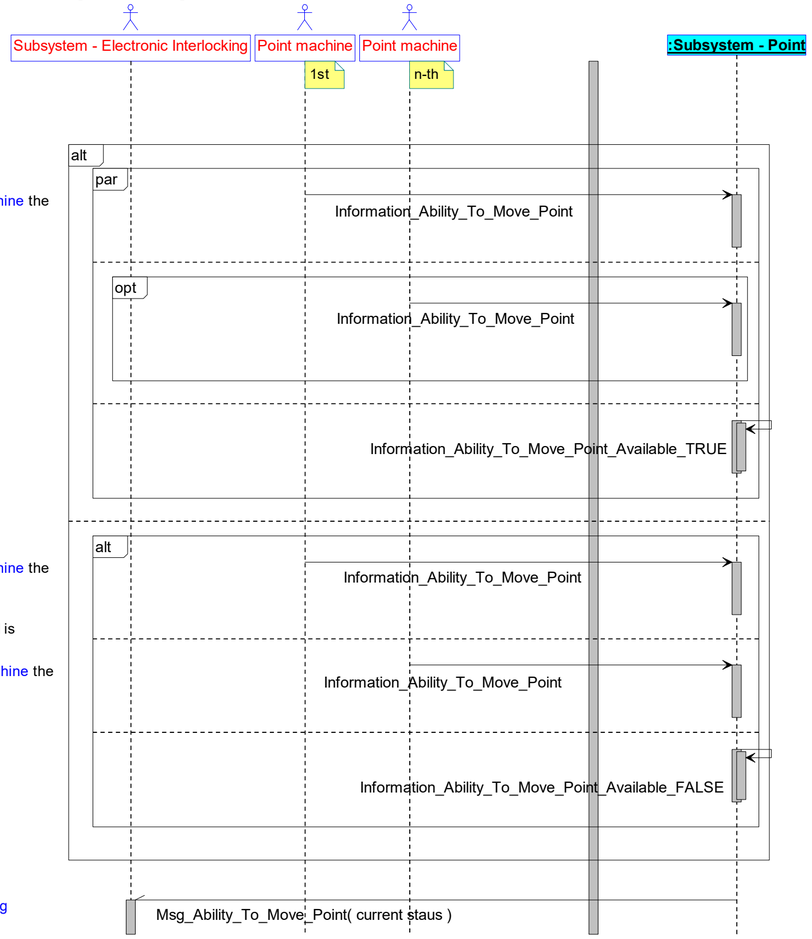
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.5788 | Info | Main Success Scenario: Report status [P SD 1.3.1]<br>**P_UC1.3: Report status**<br>Main Success Scenario: Report status [P SD 1.3.1]<br>**alt** [The Subsystem - Point is configured with one Point machine]<br>**1.a1** The Subsystem - Electronic Interlocking receives the status of the Subsystem - Point<br>**else alt** [The Subsystem - Point is configured with multiple Point machines]<br>**1.b1** The Subsystem - Electronic Interlocking receives the status of the Subsystem - Point<br>**end alt**<br>  <br> | This SD is part of [SCI-XX EfeS IF SD 1.1.1] in [Eu.Doc.119]. | Basic 4-wire single P<br>Basic 4-wire multiple P<br>Basic non-4-wire single P<br>Basic non-4-wire multiple P |
| Eu.P.1110 | Info | Main Success Scenario: Report status with single point machine [P SD 1.3.1.1]<br>**P_UC1.3: Report status**<br>Main Success Scenario: Report status with single point machine [P SD 1.3.1.1]<br>**par**<br>**1.a1** The Subsystem - Electronic Interlocking receives the point position status of the Subsystem - Point<br>**also par**<br>**1.b1** The Subsystem - Electronic Interlocking receives the ability to move status of the Subsystem - Point<br>**end par**<br>  <br> | This SD is part of [P SD 1.3.1]. | Basic 4-wire single P<br>Basic non-4-wire single P |

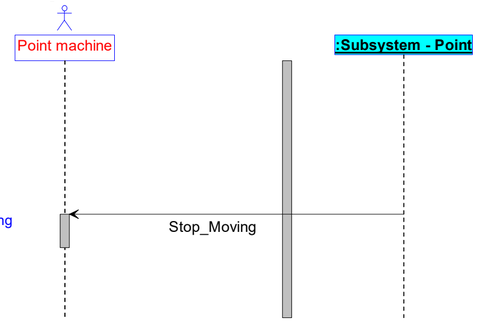
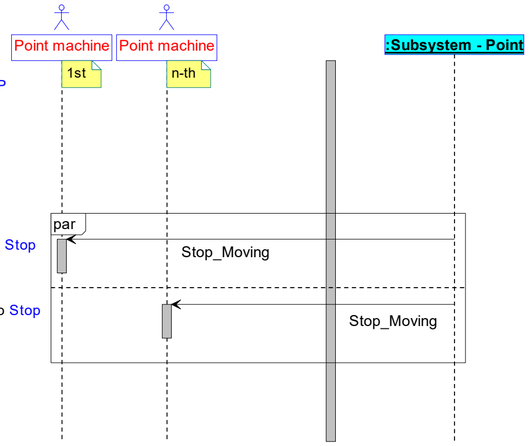
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.5787 | Info | Main Success Scenario: Report point position status with single point machine [P SD 1.3.1.1.1]<br>**P_UC1.3: Report status**<br>Main Success Scenario: Report point position status with single point machine [P SD 1.3.1.1.1]<br>**Interaction 1.3.1.1.1.A**<br>**alt** [The Subsystem - Point is configured with a non-4-wire interface to the point machine OR the Subsystem - Point is configured with a 4-wire interface to the point machine AND the last commanded position is End position "Y"]<br>**alt** [The Point is in an End position "Y"]<br>**1.a1.a1** - The Subsystem - Point receives from the Point machine the Information that the Point is in an End position "Y".<br>**else alt** [The Point is in No end position]<br>**1.a1.b1** - The Subsystem - Point receives from the Point machine the Information that the Point is in No end position.<br>**else alt** [The Point is in a Unintended position]<br>**1.a1.c1** - The Subsystem - Point receives from the Point machine the Information that the Point is in a Unintended position.<br>**end alt**<br>**else alt** [The Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is not available]<br>**1.b1** - The Subsystem - Point receives from the Point machine the Information that the Point is in No end position<br>**end alt**<br>**Interaction 1.3.1.1.1.B**<br>**2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking the Point position.<br>  <br> | If a state change happens while establishing the PDI connection and the status report Msg_Point_Position has already been sent, a new status report Msg_Point_Position has to be sent to Subsystem - Electronic Interlocking immediately after completion of the establishment of the connection.<br><br>This SD is part of [P SD 1.3.1.1]. | Basic 4-wire single P<br>Basic non-4-wire single P |
| Eu.P.5784 | Info | Main Success Scenario: Report ability to move point status with single point machine [P SD 1.3.1.1.2]<br>**P_UC1.3: Report status**<br>Main Success Scenario: Report ability to move point status with single point machine [P SD 1.3.1.1.2]<br>**Interaction 1.3.1.1.2.A**<br>**alt** [The Point is Able to move point]<br>**par**<br>**1.a1.a1** - The Subsystem - Point receives from the Point machine the Information that the Point is Able to move point.<br>**also par**<br>**1.a1.a2** The Subsystem - Point receives from the internal trigger the Information that the Point is Able to move point.<br>**end par**<br>**else alt** [The Point is Unable to move point]<br>**alt** [The Point machine is Unable to move point.]<br>**1.b1.a1** - The Subsystem - Point receives from the Point machine the Information that the Point is Unable to move point.<br>**else alt** [The internal trigger indicates Unable to move point]<br>**1.b1.b1** The Subsystem - Point receives from the internal trigger the Information that the Point is Unable to move point.<br>**end alt**<br>**end alt**<br>**Interaction 1.3.1.1.2 B**<br>**2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking the Ability to move point.<br>  <br> | Only applicable if the package [Option Able to move] is used in combination with [Basic 4-wire single P] or [Basic non-4-wire single P].<br><br>This SD is part of [P SD 1.3.1.1]. | Option Able to move |

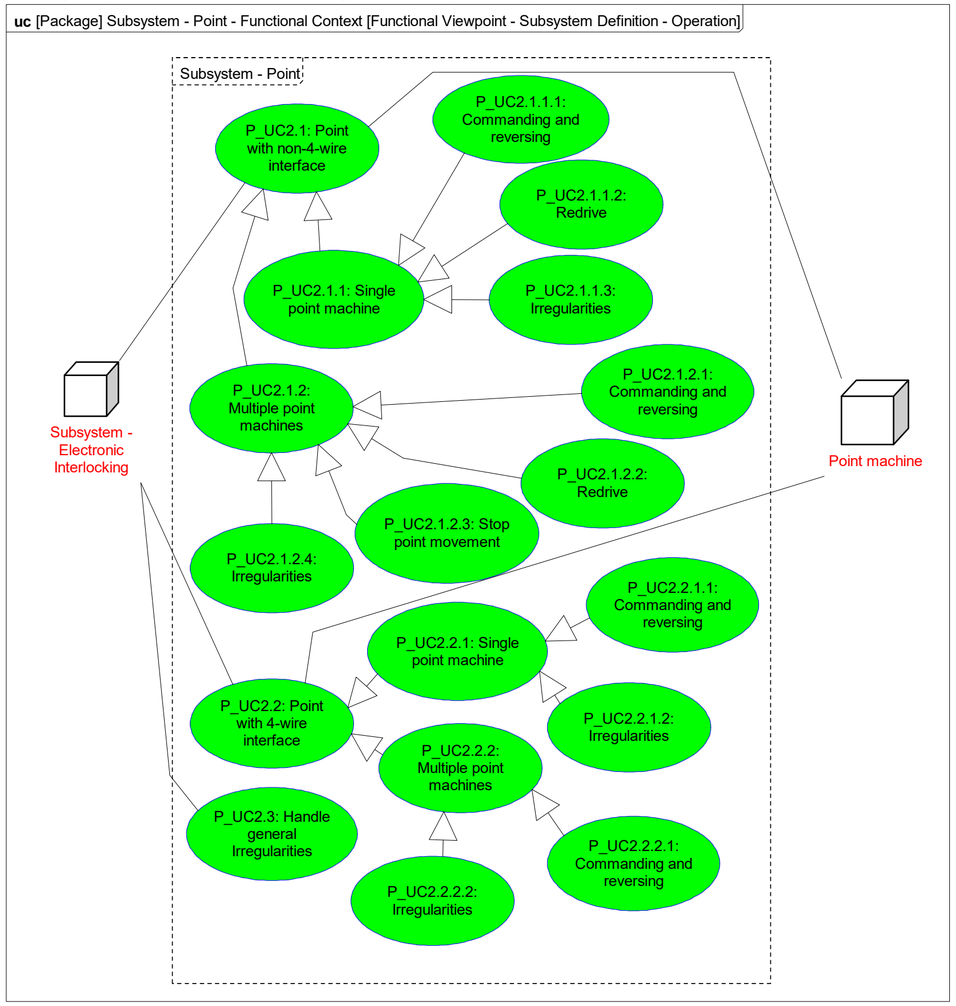
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.5789 | Info | Main Success Scenario: Report status with multiple point machines [P SD 1.3.1.2]<br>**P_UC1.3: Report status**<br>Main Success Scenario: Report status with multiple point machines [P SD 1.3.1.2]<br>**par**<br>**par**<br>**1.a1.a1** The Subsystem - Point receives the point position status from the Point machines.<br>**also par**<br>**1.a1.a2** The Subsystem - Point receives the degraded point position status from the Point machines.<br>**end par**<br>**1.a2** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking the Point position.<br>**also par**<br>**1.b1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking the Ability to move point.<br>**end par**<br>  <br> | If a state change happens while establishing the PDI connection and the status report Msg_Point_Position has already been sent, a new status report Msg_Point_Position has to be sent to Subsystem - Electronic Interlocking immediately after completion of the establishment of the connection.<br><br>This SD is part of [P SD 1.3.1]. | Basic 4-wire multiple P<br>Basic non-4-wire multiple P |

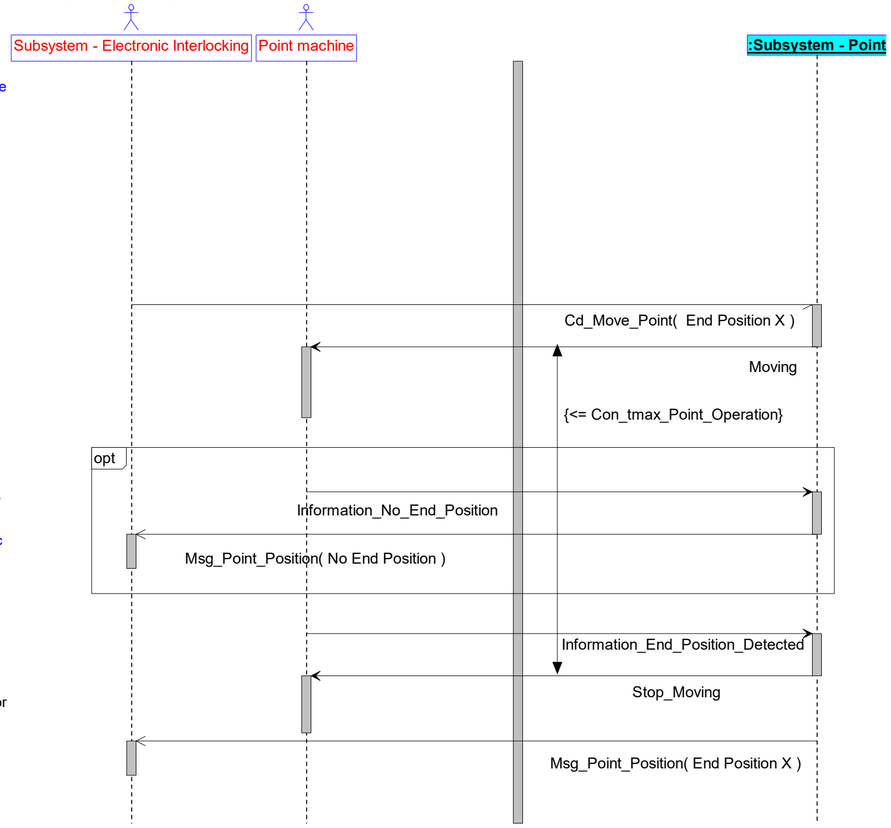
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.5786 | Info | Main Success Scenario: Report point position status with multiple point machines [P SD 1.3.1.2.1]<br>**P_UC1.3: Report status**<br>Main Success Scenario: Report point position status with multiple point machines [P SD 1.3.1.2.1]<br>**Interaction 1.3.1.2.1.A**<br>**alt** [The Subsystem - Point is configured with a non-4-wire interface to the Point machine OR the Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is End position "Y"]<br>**alt** [The Point is in an End position 'Y']<br>**par**<br>**1.a1.a1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point is in an End position 'Y'.<br>**also par**<br>**1.a1.a1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point is in an End position 'Y'.<br>**end par**<br>**else alt** [The Point is in a Unintended position]<br>**alt** [The 1st Point machine is in a Unintended position.]<br>**1.a1.b1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point is in a Unintended position.<br>**else alt** [The n-th Point machine is in a Unintended position.]<br>**1.a1.b1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point is in a Unintended position.<br>**end alt**<br>**end alt**<br>**alt** [The 1st Point machine is in a No end position.]<br>**1.a2.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point is in a No end position.<br>**else alt** [The n-th Point machine is in a No end position.]<br>**1.a2.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point is in a No end position.<br>**end alt**<br>**else alt** [The Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is not available]<br>**1.b1** - The Point is in No end position.<br>**end alt**<br>  <br> | This SD is part of [P SD 1.3.1.2]. | Basic 4-wire multiple P<br>Basic non-4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.5785 | Info | Main Success Scenario: Report degraded point position status with multiple point machines [P SD 1.3.1.2.2]<br>**P_UC1.3: Report status**<br>Main Success Scenario: Report degraded point position status with multiple point machines [P SD 1.3.1.2.2]<br>**Interaction 1.3.1.2.2.A**<br>**opt** [The Subsystem - Point is configured with one or more non-crucial Point machines]<br>**opt** [The Subsystem - Point is configured with a non-4-wire interface to the Point machine OR the Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is End position "X". The Point is in a Degraded point position "X"]<br>**par**<br>**1.a1.a1.a1** - The Subsystem - Point receives from the 1st crucial Point machine the Information that the 1st crucial Point machine is in an End position "X".<br>**also par**<br>**opt** [The Subsystem - Point is configured with more than one crucial Point machines]<br>**1.a1.a1.b1.a1** - The Subsystem - Point receives from the i-th crucial Point machine the Information that the 1st crucial Point machine is in an End position "X".<br>**end opt**<br>**also par**<br>**alt**<br>**alt**<br>**1.a1.a1.c1.a1.a1** - The Subsystem - Point receives from the 1st non-crucial Point machine the Information that the 1st non-crucial Point machine is in a Unintended position.<br>**else alt**<br>**1.a1.a1.c1.a1.b1** - The Subsystem - Point receives from the 1st non-crucial Point machine the Information that the 1st non-crucial Point machine is in a No end position.<br>**end alt**<br>**else alt** [The Subsystem - Point is configured with more than one non-crucial Point machine]<br>**alt**<br>**1.a1.a1.c1.b1.a1** - The Subsystem - Point receives from the k-th non-crucial Point machine the Information that the k-th non-crucial Point machine is in a Unintended position.<br>**else alt**<br>**1.a1.a1.c1.b1.b1** - The Subsystem - Point receives from the k-th non-crucial Point machine the Information that the k-th non-crucial Point machine is in a No end position.<br>**end alt**<br>**end alt**<br>**also par**<br>**1.a1.a1.d1** - The 1st non-crucial Point machine is NOT in an End position "Y".<br>**also par** [The Subsystem - Point is configured with a k-th non-crucial Point machine]<br>**1.a1.a1.e1** - The k-th non-crucial Point machine is NOT in an End position "Y".<br>**end par**<br>**end opt**<br>**end opt**<br>  <br> | This SD is part of [P SD 1.3.1.2]. | Basic 4-wire multiple P<br>Basic non-4-wire multiple P |

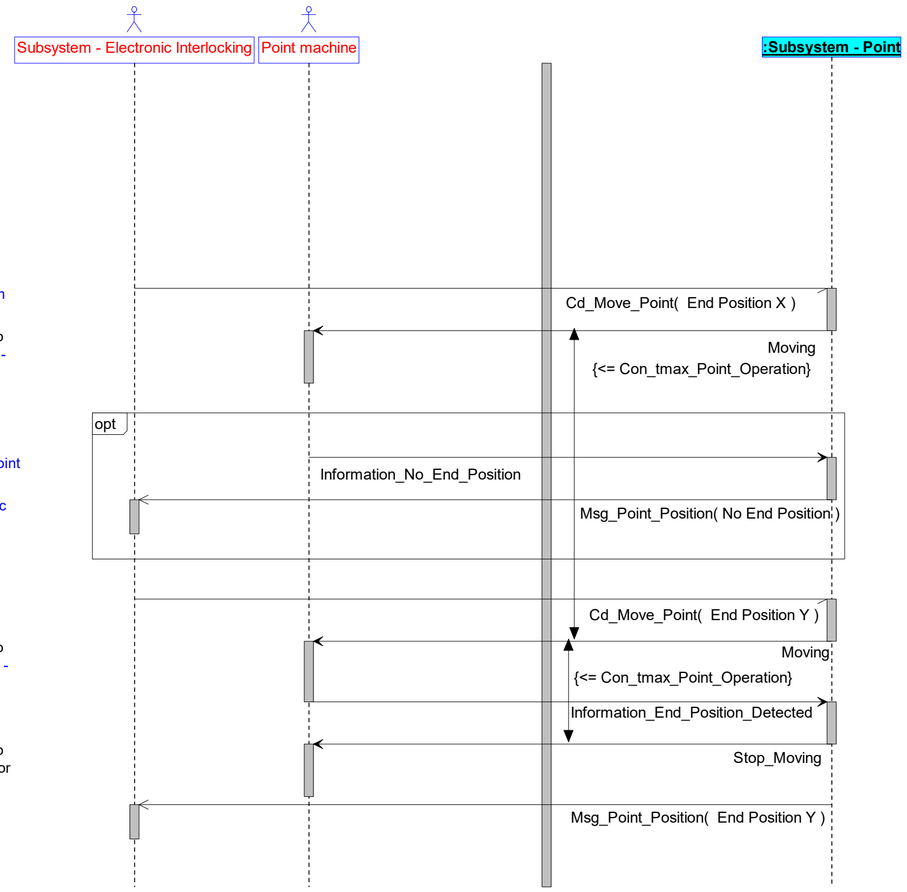
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.5783 | Info | Main Success Scenario: Report ability to move point status with multiple point machines [P SD 1.3.1.2.3]<br><br>**P_UC1.3: Report status**<br><br>Main Success Scenario: Report ability to move point status with multiple point machines [P SD 1.3.1.2.3]<br><br>**Interaction 1.3.1.2.3.A**<br>**alt** [The Point is Able to move point]<br>**par**<br>**1.a1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the 1st Point machine is Able to move point.<br>**also par**<br>**opt** [The n-th Point machine is configured with full functionality]<br>**1.a1.b1.a1** - The Subsystem - Point receives from the n-th Point machine the Information that the n-th Point machine Able to move point.<br>**end opt**<br>**also par**<br>**1.a1.c1** - The Subsystem - Point internal trigger Ability to move point available indicates availability of Ability to move point.<br>**end par**<br>**else alt** [The Point is Unable to move point]<br>**alt** [The 1st Point machine is Unable to move point.]<br>**1.b1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the 1st Point machine Unable to move point.<br>**else alt** [The n-th Point machine is configured with full functionality and is Unable to move point.]<br>**1.b1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the n-th Point machine Unable to move point.<br>**else alt** [The internal trigger indicates Unable to move point]<br>**1.b1.c1** - The Subsystem - Point internal trigger Ability to move point available indicates non-availability of Ability to move point.<br>**end alt**<br>**end alt**<br><br>**Interaction 1.3.1.2.3.B**<br>**2.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking the Ability to move point.<br><br>  <br> | Only applicable if the package [Option Able to move] is used in combination with [Basic 4-wire multiple P] or [Basic non-4-wire multiple P].<br><br>This SD is part of [P SD 1.3.1.2]. | Option Able to move |
| Eu.P.1109 | Info | P_UC1.4: Establish initial state of outputs | The Subsystem-UseCase P_UC1.4: Establish initial state of outputs defines the main success scenario for establishing the initial state of outputs of the Subsystem - Point. | Basic 4-wire single P<br>Basic 4-wire multiple P<br>Basic non-4-wire single P<br>Basic non-4-wire multiple P |

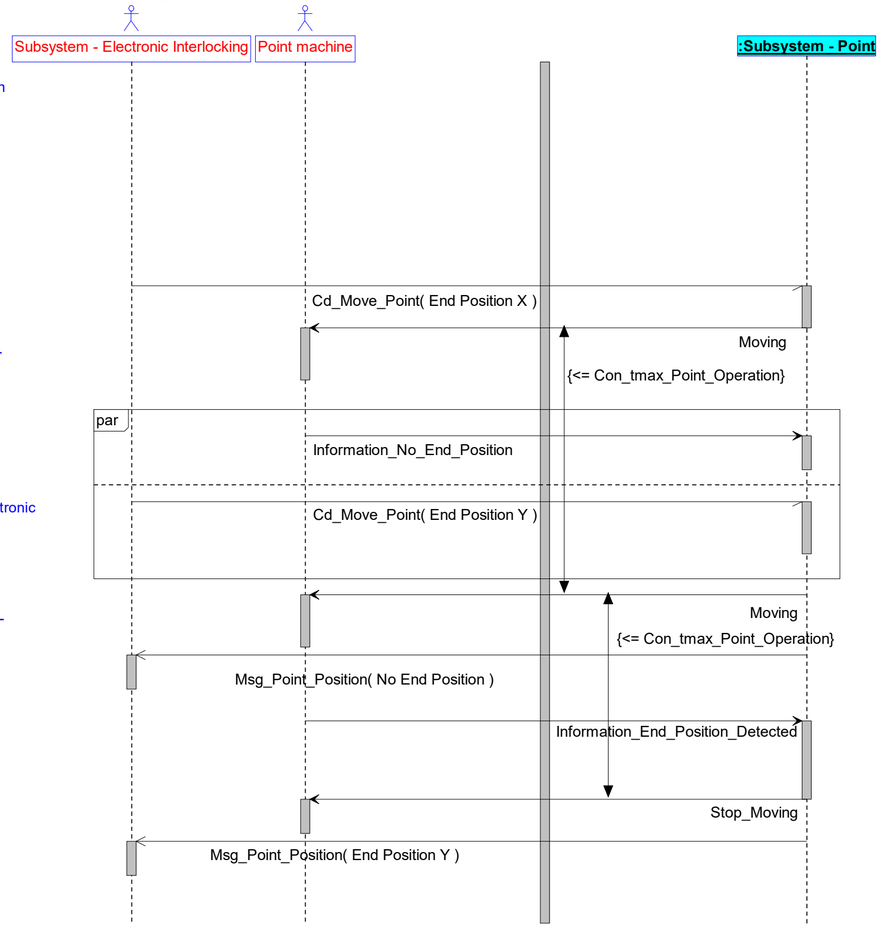
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.1466 | Info | Main Success Scenario: Set Initial State of Outputs with single point machine [P SD 1.4.1]<br>**P_UC1.4: Establish initial state of outputs**<br>Main Success Scenario: Set Initial State of Outputs with single point machine [P SD 1.4.1]<br>**Precondition:**<br>The Subsystem - Point is in the state BOOTING.<br>**Interaction 1.4.1.A:**<br>**1.** - The Subsystem - Point enters the state INITIALISING.<br>**2.** The Subsystem - Point sends the Command to the Point machine to Stop moving *the* Point machine.<br>**Postconditions:**<br>The Subsystem - Point is in the state INITIALISING.<br>The Initial State Of Outputs of the Subsystem - Point has been set.<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic 4-wire single P<br>Basic non-4-wire single P |
| Eu.P.5790 | Info | Main Success Scenario: Set Initial State of Outputs with multiple point machines [P SD 1.4.2]<br>**P_UC1.4: Establish initial state of outputs**<br>Main Success Scenario: Set Initial State of Outputs with multiple point machines [P SD 1.4.2]<br>**Precondition:**<br>The Subsystem - Point is in the state BOOTING.<br>**Interaction 1.4.2.A:**<br>**1.** - The Subsystem - Point enters the state INITIALISING.<br>**par**<br>**2.a1** The Subsystem - Point sends the Command to the 1st Point machine to Stop moving *the* Point machine.<br>**also par**<br>**2.b1** The Subsystem - Point sends the Command to the n-th Point machine to Stop moving *the* Point machine.<br>**end par**<br>**Postconditions:**<br>The Subsystem - Point is in the state INITIALISING.<br>The Initial State Of Outputs of the Subsystem - Point has been set.<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic 4-wire multiple P<br>Basic non-4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6706 | Info | [Package] Subsystem - Point - Functional Context [Functional Viewpoint - Subsystem Definition - Operation]<br>  <br> | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P<br>Option Redrive |
| Eu.P.6732 | Info | P_UC2.1: Point with non-4-wire interface | The Subsystem-UseCase "P_UC2.1: Point with non-4-wire interface" defines the behaviour of the Subsystem - Point which works with a non-4-wire interface.<br>The behaviour will be defined in the following UseCases:<br>P_UC2.1.1: Single point machine<br>P_UC2.1.2: Multiple point machines | Basic non-4-wire single P<br>Basic non-4-wire multiple P |
| Eu.P.6720 | Info | P_UC2.1.1: Single point machine | The Subsystem-UseCase "P_UC2.1.1: Single point machine" defines the behaviour of the Subsystem - Point which works with a single point machine via non-4-wire interface.<br>The behaviour will be defined in the following UseCases:<br>P_UC2.1.1.1: Commanding and reversing<br>P_UC2.1.1.2: Redrive<br>P_UC2.1.1.3: Irregularities | Basic non-4-wire single P |
| Eu.P.6707 | Info | P_UC2.1.1.1: Commanding and reversing | The Subsystem-UseCase "P_UC2.1.1.1: Commanding and reversing" defines the behaviour of commanding and reversing a single point machine via non-4-wire interface. | Basic non-4-wire single P |

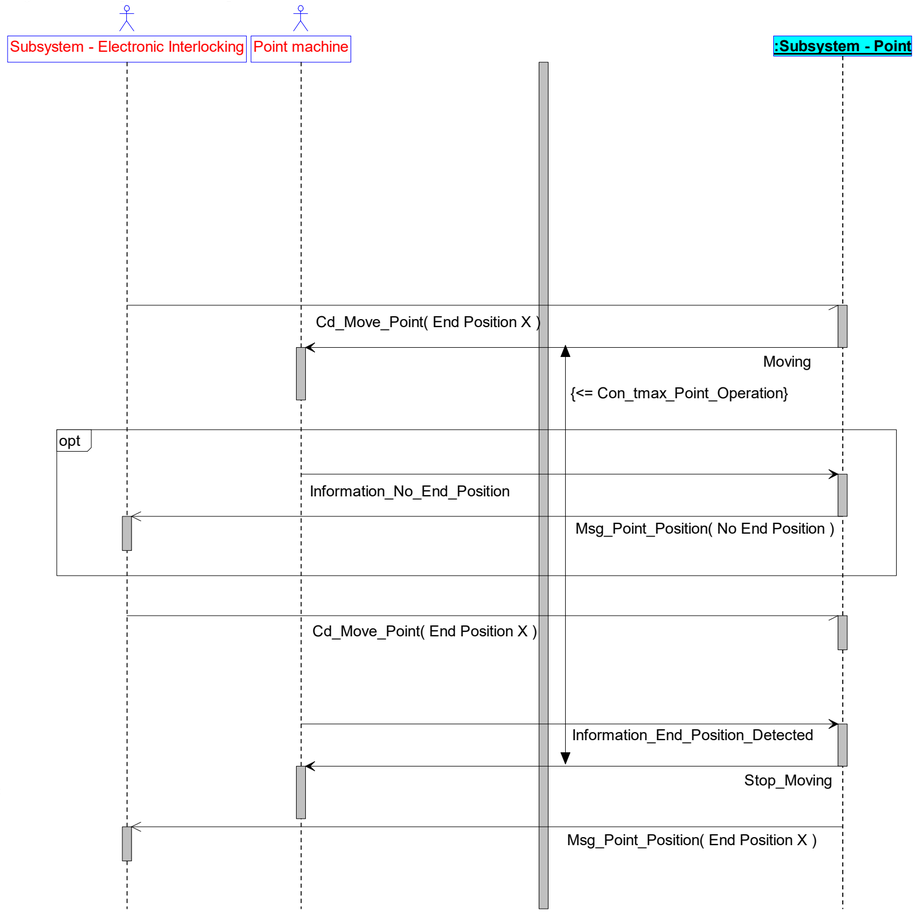
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6714 | Info | Main Success Scenario: Moving of the Point with a single point machine Non 4W [P SD 2.1.1.1.1]<br>**P_UC2.1.1.1: Commanding and reversing**<br>Main Success Scenario: Moving of the Point with a single point machine Non 4W [P SD 2.1.1.1.1]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- No end position, or<br>- an Unintended position.<br>**Interaction 2.1.1.1.1.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**2.** The Subsystem - Point sends the Command to the Point machine to move the Point machine to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**Interaction 2.1.1.1.1.B:**<br>**opt** [The Subsystem - Point was previously in an End position or a Unintended position]<br>**3.a1** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in No end position.<br>**3.b1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**end opt**<br>**Interaction 2.1.1.1.1.C:**<br>**4.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in an End position "X".<br>**5.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine. The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br>**6.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "X".<br>**Postcondition:**<br>The Subsystem - Point is in an End position "X".<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic non-4-wire single P |

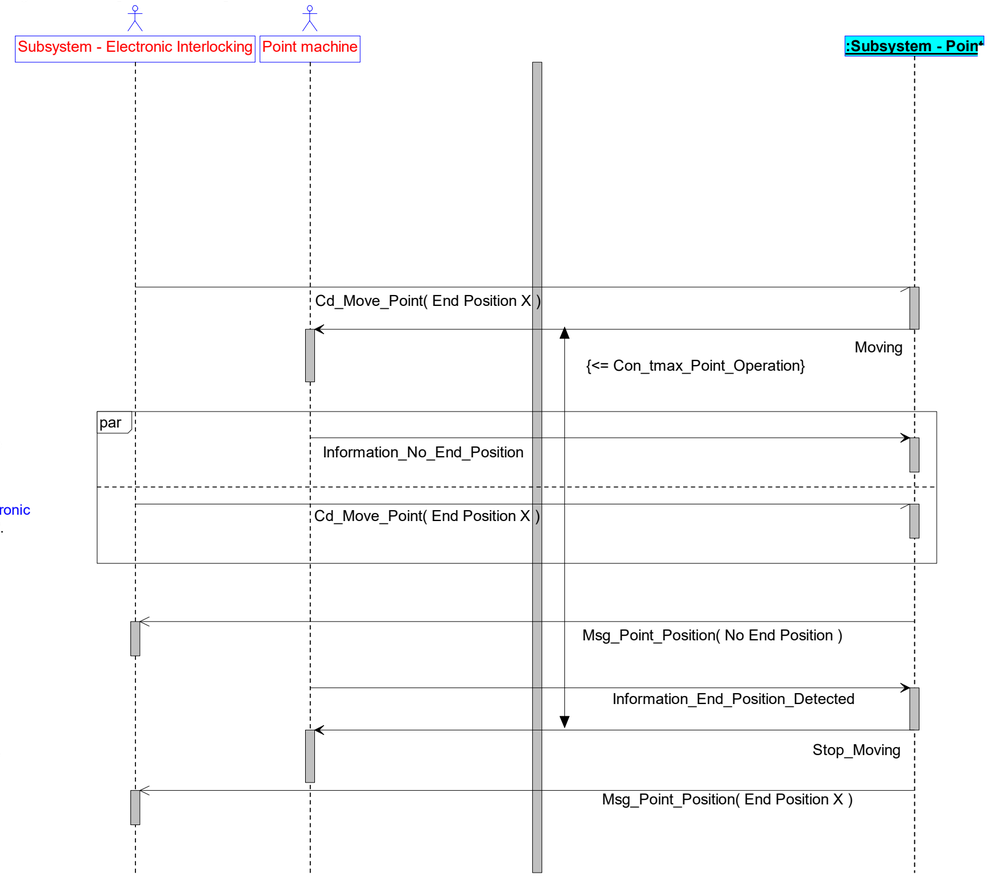
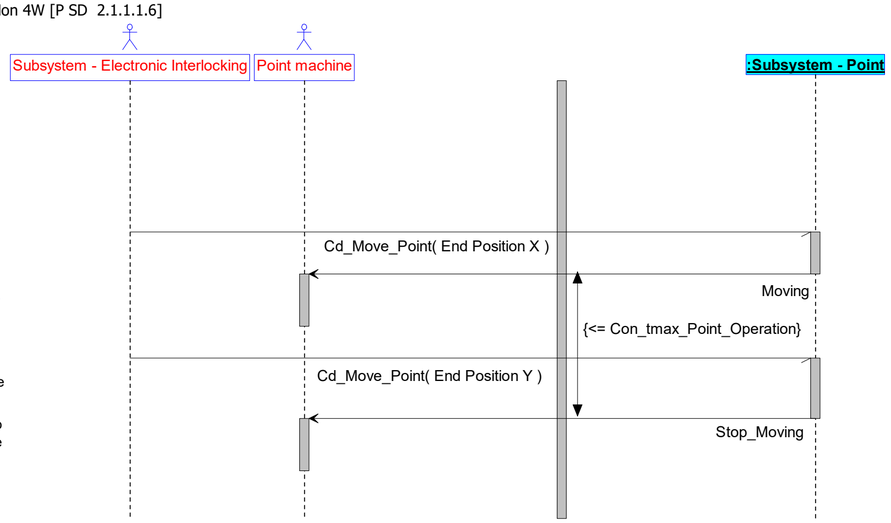
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6713 | Info | Alternative Scenario: Reversing Point Non 4W [P SD 2.1.1.1.2]<br>**P_UC2.1.1.1: Commanding and reversing**<br>Alternative Scenario: Reversing Point Non 4W [P SD 2.1.1.1.2]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- No end position, or<br>- an Unintended position.<br>**Interaction 2.1.1.1.2.A:**<br>**1.** - The Subsystem - Point receives the Command from the Subsystem - Electronic Interlocking to move the Point to an End position "X".<br>**2.** The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**Interaction 2.1.1.1.2.B:**<br>**opt** [The Subsystem - Point was previously in an End position or an Unintended position]<br>**3.a1** - The Subsystem - Point receives the Information from the Point machine that the Point is in No end position.<br>**3.a2** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**end opt**<br>**Interaction 2.1.1.1.2.C:**<br>**4.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "Y".<br>**5.** The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "Y". At this moment the Subsystem - Point restarts to monitor the time period Con_tmax_Point_Operation.<br>**6.** The Subsystem - Point receives from the Point machine the Information that the Point machine is in an End position "Y".<br>**7.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine. The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br>**8.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "Y".<br>**Postcondition:**<br>The Subsystem - Point is in an End position "Y". | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic non-4-wire single P |

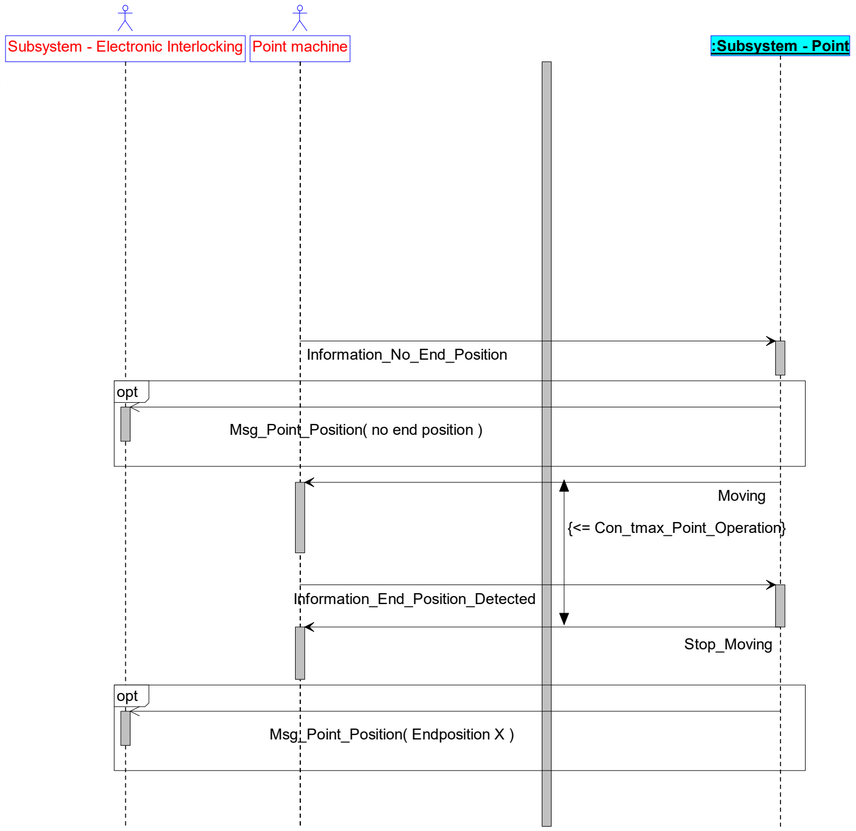


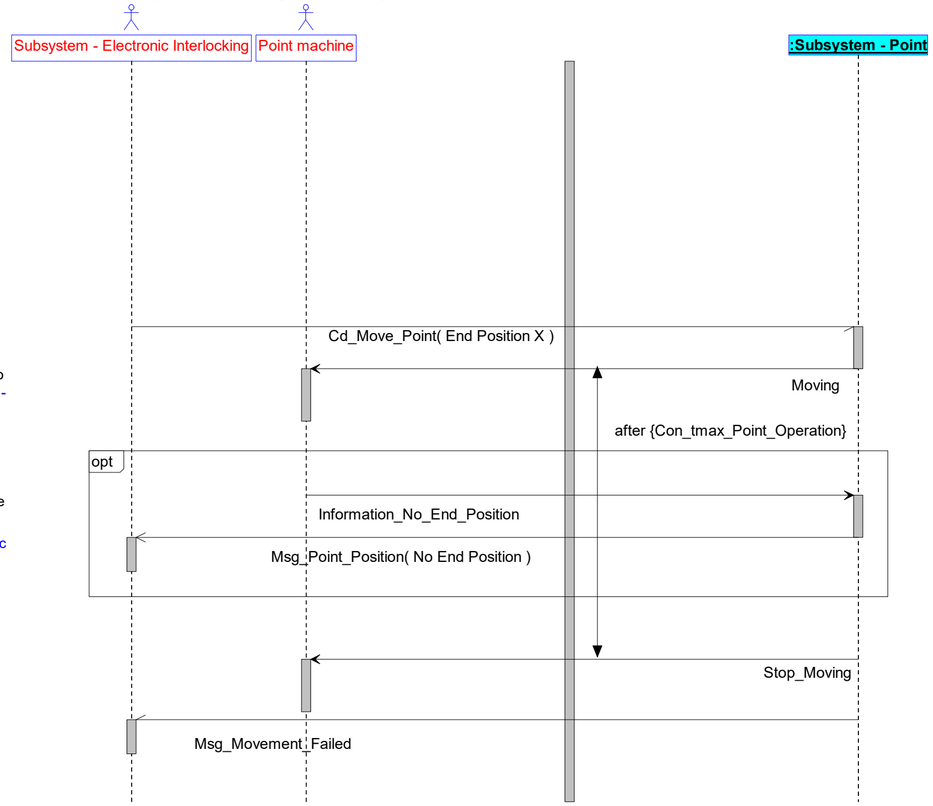
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6711 | Info | Alternative Scenario: Reversing Point directly after the position has been commanded Non 4W [P SD 2.1.1.1.3]<br>**P_UC2.1.1.1: Commanding and reversing**<br>Alternative Scenario: Reversing Point directly after the position has been commanded Non 4W [P SD 2.1.1.1.3]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- an Unintended position.<br>**Interaction 2.1.1.1.3.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**2.** The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**Interaction 2.1.1.1.3.B:**<br>**par**<br>**3.a1** The Subsystem - Point receives from the Point machine the Information that the Point machine is in No end position.<br>**also par**<br>**3.b1** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to the opposite End position "Y".<br>**end par**<br>**4.** The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "Y". At this moment the Subsystem - Point restarts to monitor the time period Con_tmax_Point_Operation.<br>**5.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**Interaction 2.1.1.1.3.C:**<br>**6.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in an End position "Y". The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br>**7.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine.<br>**8.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "Y".<br>**Postcondition:**<br>The Subsystem - Point is in an End position "Y".<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic non-4-wire single P |

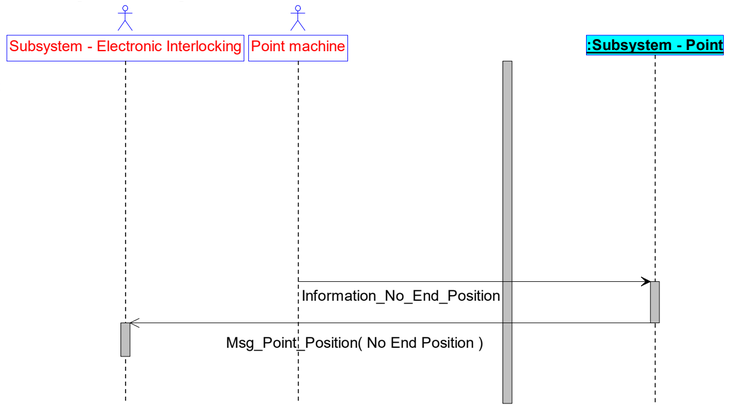
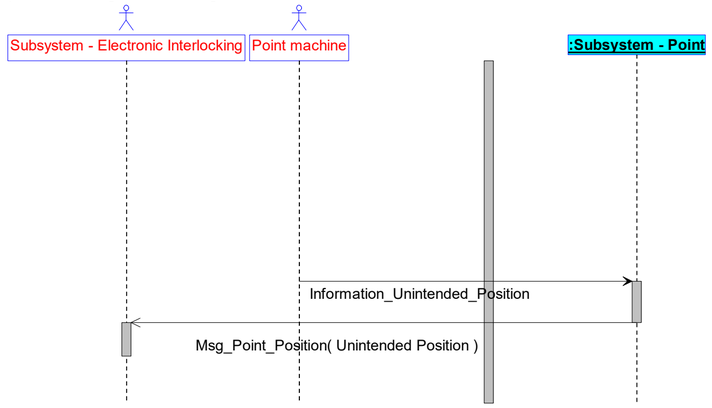
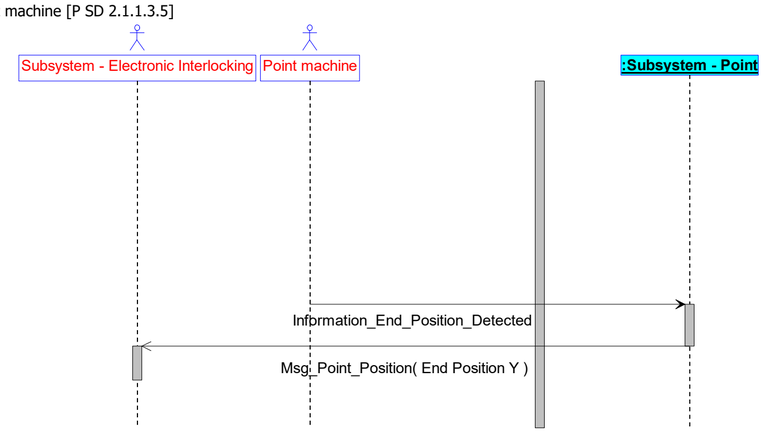
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6708 | Info | Alternative Scenario: Moving of the Point with repeated command of moving #1 Non 4W [P SD 2.1.1.1.4]<br>**P_UC2.1.1.1: Commanding and reversing**<br>Alternative Scenario: Moving of the Point with repeated command of moving #1 Non 4W [P SD 2.1.1.1.4]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- No end position, or<br>- an Unintended position.<br>**Interaction 2.1.1.1.4.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**2.** The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**Interaction 2.1.1.1.4.B:**<br>**opt** [The Subsystem - Point is in an End position or a Unintended position]<br>**3.a1** - The Subsystem - Point receives from the Point machine the Information that the Point is in No end position.<br>**3.a2** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**end opt**<br>**Interaction 2.1.1.1.4.C:**<br>**4.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**5.** The Subsystem - Point ignores the command from the Subsystem - Electronic Interlocking.<br>**Interaction 2.1.1.1.4.D:**<br>**6.** - The Subsystem - Point receives from the Point machine the Information that the Point is in an End position "X".<br>**7.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point. The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br>**8.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "X".<br>**Postcondition:**<br>The Subsystem - Point is in an End position "X". | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic non-4-wire single P |

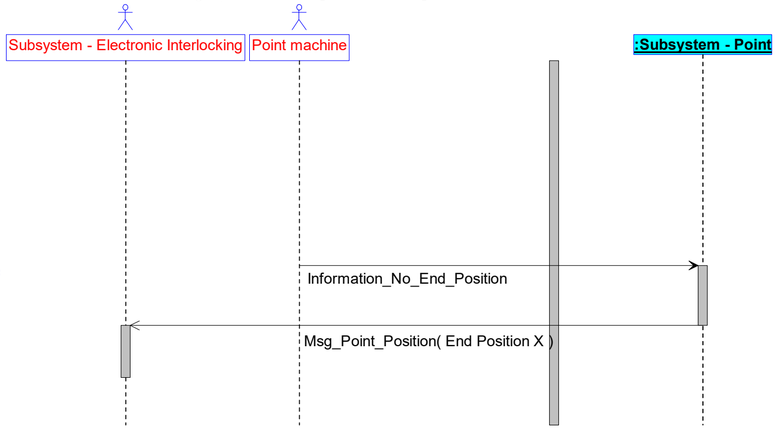
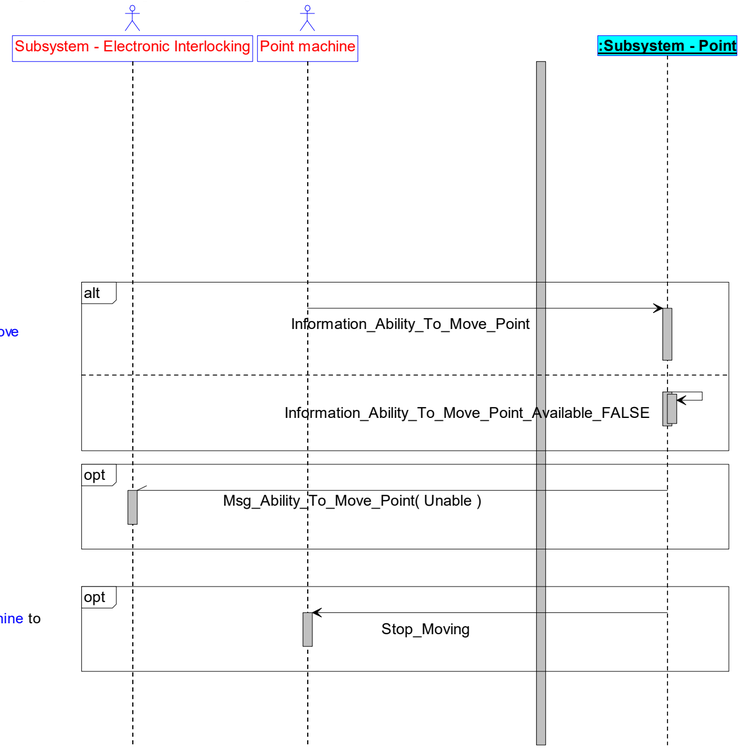


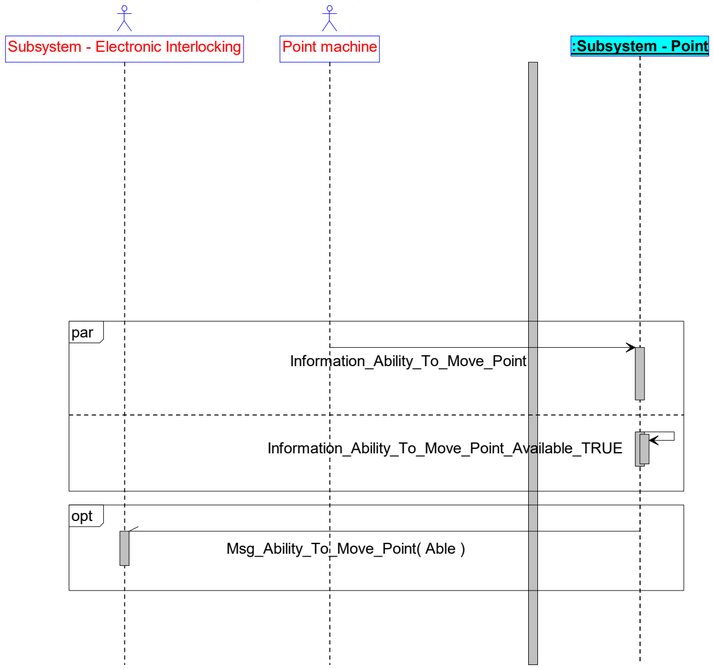
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6709 | Info | Alternative Scenario: Moving of the Point with repeated command of moving #2 Non 4W [P SD 2.1.1.1.5]<br>**P_UC2.1.1.1: Commanding and reversing**<br>Alternative Scenario: Moving of the Point with repeated command of moving #2 Non 4W [P SD 2.1.1.1.5]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y" or<br>- an Unintended position.<br>**Interaction 2.1.1.1.5.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**2.** The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**Interaction 2.1.1.1.5.B:**<br>**par**<br>**3.a1** - The Subsystem - Point receives from the Point machine the Information that the Point is in No end position.<br>**also par**<br>**3.b1** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**end par**<br>**4.** The Subsystem - Point ignores the command from the Subsystem - Electronic Interlocking.<br>**5.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**Interaction 2.1.1.1.5.C:**<br>**6.** - The Subsystem - Point receives from the Point machine the Information that the Point is in an End position "X".<br>**7.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point. The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br>**8.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "X".<br>**Postcondition:**<br>The Subsystem - Point is in an End position "X".<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic non-4-wire single P |
| Eu.P.6712 | Info | Alternative Scenario: Reversing Point directly without position change Non 4W [P SD 2.1.1.1.6]<br>**P_UC2.1.1.1: Commanding and reversing**<br>Alternative Scenario: Reversing Point directly without position change Non 4W [P SD 2.1.1.1.6]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is in an End position "Y".<br>**Interaction 2.1.1.1.6.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**2.** The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**Interaction 2.1.1.1.6.B:**<br>**3.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "Y". The Subsystem - Point is still in an End position "Y".<br>**4.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point. The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br>**Postcondition:**<br>The Subsystem - Point is in an End position "Y".<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic non-4-wire single P |

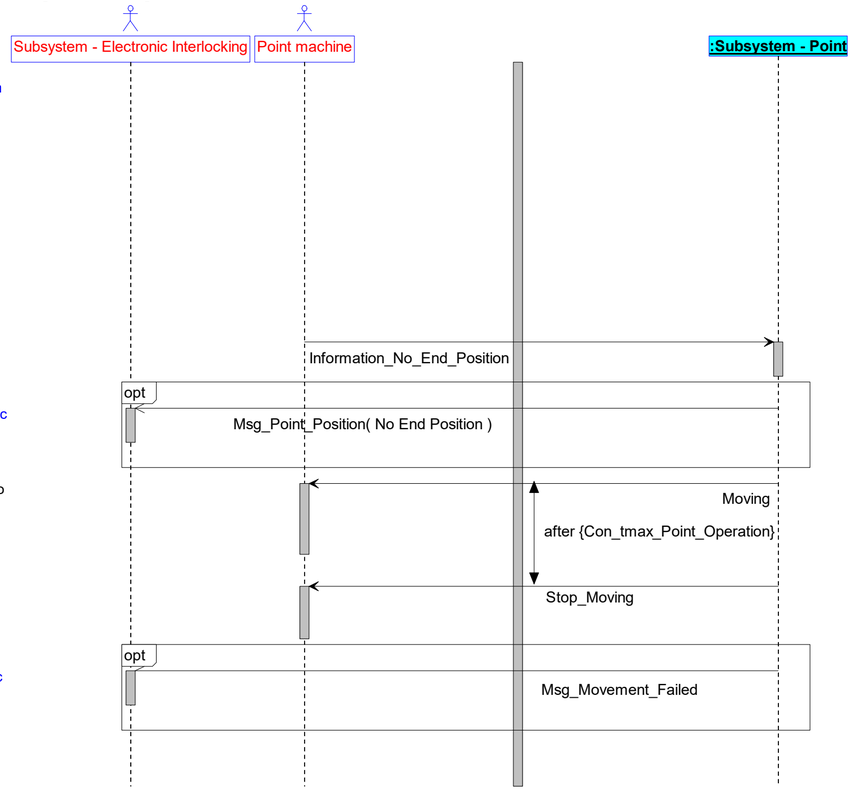
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6715 | Info | P_UC2.1.1.2: Redrive | The Subsystem-UseCase "P_UC2.1.1.2: Redrive" defines the behaviour of redriving a single point machine via non-4-wire interface. Only applicable if the package [Option Redrive] is used in combination with [Basic non-4-wire single P]. | Option Redrive |
| Eu.P.6717 | Info | Main Success Scenario: Redrive Point after lost end position Non 4W [P SD 2.1.1.2.1]<br>**P_UC2.1.1.2: Redrive**<br>Main Success Scenario: Redrive Point after lost end position Non 4W [P SD 2.1.1.2.1]<br>**Precondition:**<br>The Subsystem - Point is in the state INITIALISING or OPERATIONAL.<br>The Subsystem - Point is in an End position "X".<br>The last Cd_Move_Point Command received was also for End position "X".<br>The last Cd_Move_Point Command received occurred since exiting the state BOOTING.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is configured as Redrive point.<br>**Interaction 2.1.1.2.1.A:**<br>**1.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in No end position.<br>**opt** [The Subsystem - Point is in the state OPERATIONAL.]<br>**2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**end opt**<br>**3.** The Subsystem - Point sends the Command to the Point machine to move the Point back to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**Interaction 2.1.1.2.1.B:**<br>**4.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in an End position "X".<br>**5.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine. The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br>**opt** [The Subsystem - Point is in the state OPERATIONAL.]<br>**6.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "X".<br>**end opt**<br>**Postcondition:**<br>The Subsystem - Point is in an End position "X".<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. Only applicable if the package [Option Redrive] is used in combination with [Basic non-4-wire single P]. | Option Redrive |
| Eu.P.6718 | Info | P_UC2.1.1.3: Irregularities | The Subsystem-UseCase "P_UC2.1.1.3: Irregularities" defines the behaviour of the Subsystem - Point which works with a single point machine via non-4-wire interface, when an irregularity occurs. | Basic non-4-wire single P |

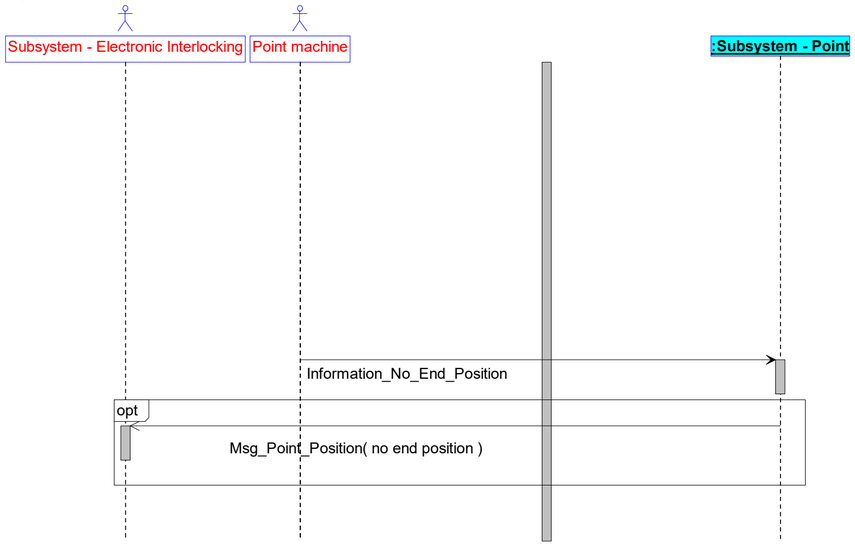
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.1284 | Info | Alternative Scenario: Handle and report Point operation timeout with position change with single point machine [P SD 2.1.1.3.1]<br>**P_UC2.1.1.3: Irregularities**<br>Alternative Scenario: Handle and report Point operation timeout with position change with single point machine [P SD 2.1.1.3.1]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- No end position, or<br>- a Unintended position.<br>**Interaction 2.1.1.3.1.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**2.** The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**Interaction 2.1.1.3.1.B:**<br>**opt** [The Subsystem - Point was previously in an End position or a Unintended position]<br>**3.a1** - The Subsystem - Point receives from the Point machine the Information that the Point is in No end position.<br>**3.a2** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**end opt**<br>**Interaction 2.1.1.3.1.C:**<br>**4.** - The Subsystem - Point detects that the time period Con_tmax_Point_Operation has exceeded and then sends the Command to the Point machine to stop moving the Point machine.<br>**5.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred.<br>**Postcondition:**<br>The Subsystem - Point is in No end position.<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic non-4-wire single P |
| Eu.P.6719 | Info | Alternative Scenario: Handle and report failed movement with single point machine without position change [P SD 2.1.1.3.2]<br>**P_UC2.1.1.3: Irregularities**<br>Alternative Scenario: Handle and report failed movement with single point machine without position change [P SD 2.1.1.3.2]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y",<br>- No end position, or<br>- a Unintended position.<br>**Interaction 2.1.1.3.2.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**2.** The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**Interaction 2.1.1.3.2.B:**<br>**3.** - A failure occurred during the movement resulting in a Failed Movement.<br>The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br>**4.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine.<br>**5.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred.<br>**Postcondition:**<br>--<br> <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic non-4-wire single P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.1474 | Info | Alternative Scenario: Handle and report No end position with single point machine [P SD 2.1.1.3.3]<br>**P_UC2.1.1.3: Irregularities**<br>Alternative Scenario: Handle and report No end position with single point machine [P SD 2.1.1.3.3]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- a Unintended position<br>**Interaction 2.1.1.3.3.A:**<br>**1.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in No end position.<br>**2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**Postcondition:**<br>The Subsystem - Point is in No end position.<br>  <br> | | Basic non-4-wire single P |
| Eu.P.1273 | Info | Alternative Scenario: Handle and report Unintended position with single point machine [P SD 2.1.1.3.4]<br>**P_UC2.1.1.3: Irregularities**<br>Alternative Scenario: Handle and report Unintended position with single point machine [P SD 2.1.1.3.4]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- No end position.<br>**Interaction 2.1.1.3.4.A:**<br>**1.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in a Unintended position.<br>**2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in a Unintended position.<br>**Postcondition:**<br>The Subsystem - Point is in a Unintended position.<br>  <br> | | Basic non-4-wire single P |
| Eu.P.3207 | Info | Alternative Scenario: Handle and report End Position with single point machine [P SD 2.1.1.3.5]<br>**P_UC2.1.1.3: Irregularities**<br>Alternative Scenario: Handle and report End Position with single point machine [P SD 2.1.1.3.5]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- No end position, or<br>- a Unintended position.<br>**Interaction 2.1.1.3.5.A:**<br>**1.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in an End position.<br>**2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "Y".<br>**Postcondition:**<br>The Subsystem - Point is in an End position "Y".<br>  <br> | | Basic non-4-wire single P |

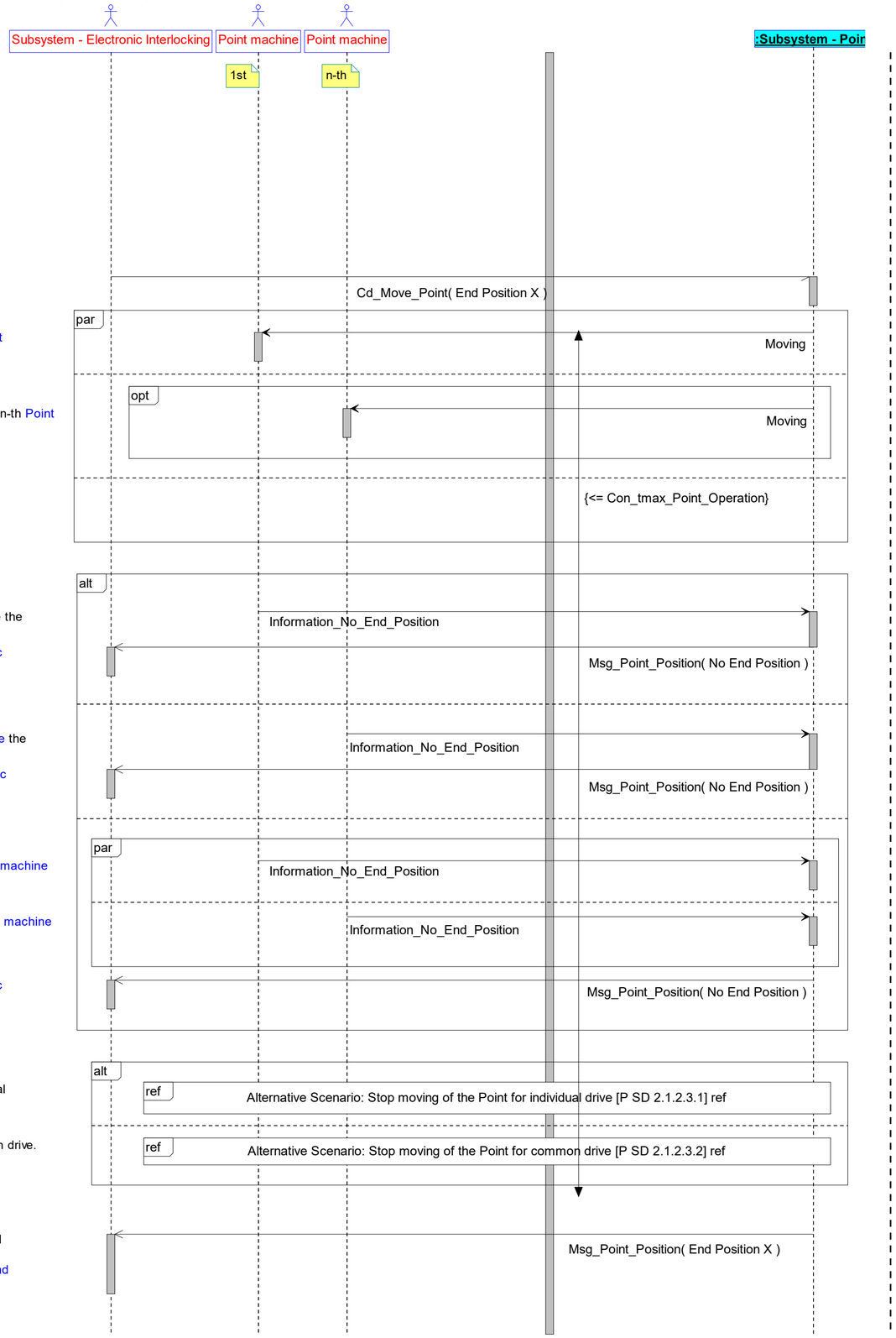
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.4761 | Info | Alternative Scenario: Handle and report End Position out of the other End Position with single point machine [P SD 2.1.1.3.6]<br>**P_UC2.1.1.3: Irregularities**<br>Alternative Scenario: Handle and report End Position out of the other End Position with single point machine [P SD 2.1.1.3.6]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is in an End position "Y".<br>**Interaction 2.1.1.3.6.A:**<br>**1.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in an End position "X".<br>**2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "X".<br>**Postcondition:**<br>The Subsystem - Point is in End position "X".<br>  <br> | | Basic non-4-wire single P |
| Eu.P.5373 | Info | Alternative Scenario: Handle and report loss of ability to move point with single point machine [P SD 2.1.1.3.7]<br>**P_UC2.1.1.3: Irregularities**<br>Alternative Scenario: Handle and report loss of ability to move point with single point machine [P SD 2.1.1.3.7]<br>**Precondition**:<br>The Subsystem - Point is in the state INITIALISING or OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is configured to observe the Ability to move point.<br>The Subsystem - Point is in:<br>- Able to move point<br>**Interaction 2.1.1.3.7.A**:<br>**alt** [The Point machine is Unable to move point]<br>**1.a1** - The Subsystem - Point detects from the switch states of the Interface P3 to the Point machine that the Point has no Ability to move point.<br>**else alt** [The internal trigger indicates Unable to move point]<br>**1.b1** - The Subsystem - Point internal trigger indicates that the Subsystem - Point is Unable to move point<br>**end alt**<br>**opt** [The Subsystem - Point is in the state OPERATIONAL]<br>**2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that it is Unable to move point.<br>**end opt**<br>**Interaction 2.1.1.3.7.B**:<br>**opt** [The point machine is Moving]<br>**3.a1** The Subsystem - Point sends the Command to the Point machine to Stop moving the Point machine.<br>**end opt**<br>**Postcondition:**<br>The Subsystem - Point is in:<br>- Unable to move point<br>  <br> | Only applicable if the package [Option Able to move] is used in combination with [Basic non-4-wire multiple P]. | Option Able to move |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.5797 | Info | Alternative Scenario: Handle and report restoring of Ability to move point with single point machine [P SD 2.1.1.3.8]<br>**P_UC2.1.1.3: Irregularities**<br>Alternative Scenario: Handle and report restoring of Ability to move point with single point machine [P SD 2.1.1.3.8]<br>**Precondition**:<br>The Subsystem - Point is in the state INITIALISING or OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is configured to observe the Ability to move point.<br>The Subsystem - Point is in:<br>- Unable to move point<br>**Interaction 2.1.1.3.8.A**:<br>**par**<br>**1.a1** - The Subsystem - Point detects from the switch states of the Interface P3 to the Point machine that the Point machine has regained Ability to move point.<br>**also par**<br>**1.b1** - The Subsystem - Point internal trigger indicates that the Subsystem - Point is Able to move point.<br>**end par**<br>**opt** [The Subsystem - Point is in state OPERATIONAL]<br>**2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that it is Able to move point.<br>**end opt**<br>**Postcondition:**<br>The Subsystem - Point is in:<br>- Able to move point<br>  <br> | Only applicable if the package [Option Able to move] is used in combination with [Basic non-4-wire single P]. | Option Able to move |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6716 | Info | Alternative Scenario: Handle Point operation timeout during Redrive Non 4W [P SD 2.1.1.3.9]<br>**P_UC2.1.1.3: Irregularities**<br>Alternative Scenario: Handle Point operation timeout during Redrive Non 4W [P SD 2.1.1.3.9]<br>**Precondition:**<br>The Subsystem - Point is in the state INITIALISING or OPERATIONAL.<br>The Subsystem - Point is in an End position "X"<br>The last Cd_Move_Point Command received was also for End position "X"<br>The last Cd_Move_Point Command received occurred since exiting the state BOOTING.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is configured as Redrive point.<br>**Interaction 2.1.1.3.9.A:**<br>**1.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in No end position.<br>**opt** [The Subsystem - Point is in the state OPERATIONAL.]<br>**2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**end opt**<br>**3.** The Subsystem - Point sends the Command to the Point machine to move the Point back to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**Interaction 2.1.1.3.9.B:**<br>**4.** The Subsystem - Point detects that the time period Con_tmax_Point_Operation has exceeded and then sends the Command to the Point machine to stop moving the Point machine.<br>**opt** [The Subsystem - Point is in the state OPERATIONAL.]<br>**5.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking to indicate that a Failed Movement has occurred.<br>**end opt**<br>**Postcondition:**<br>The Subsystem - Point is in No end position.<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. Only applicable if the package [Option Redrive] is used in combination with [Basic non-4-wire single P]. | Option Redrive |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.7370 | Info | Alternative Scenario: Redrive Point trigger unable to move [P SD 2.1.1.3.10]<br>**P_UC2.1.1.3: Irregularities**<br>Alternative Scenario: Redrive Point trigger unable to move [P SD 2.1.1.3.10]<br>**Precondition:**<br>The Subsystem - Point is in the state INITIALISING or OPERATIONAL.<br>The Subsystem - Point is in an End position "X".<br>The last Cd_Move_Point Command received was also for End position "X".<br>The last Cd_Move_Point Command received occurred since exiting the state BOOTING.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is configured as Redrive point.<br>The Subsystem - Point is Unable to move point.<br>**Interaction 2.1.1.3.10.A:**<br>**1.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in No end position.<br>**opt** [The Subsystem - Point is in the state OPERATIONAL.]<br>**2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**end opt**<br>**Postcondition:**<br>The Subsystem - Point is in No end position.<br>  <br> | Only applicable if the package [Option Redrive] is used in combination with [Basic non-4-wire single P]. | Option Redrive |
| Eu.P.6731 | Info | P_UC2.1.2: Multiple point machines | The Subsystem-UseCase "P_UC2.1.2: Multiple point machines" defines the behaviour of the Subsystem - Point which works with multiple point machines via non-4-wire interface.<br>The behaviour will be defined in the following UseCases:<br>P_UC2.1.2.1: Commanding and reversing<br>P_UC2.1.2.2: Redrive<br>P_UC2.1.2.4: Irregularities | Basic non-4-wire multiple P |
| Eu.P.6721 | Info | P_UC2.1.2.1: Commanding and reversing | The Subsystem-UseCase "P_UC2.1.2.1: Commanding and reversing" defines the behaviour of commanding and reversing a multiple point machine via non-4-wire interface. | Basic non-4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6722 | Info | Main Success Scenario: Moving of the Point with multiple Point machines Non 4W [P SD 2.1.2.1.1]<br>**P_UC2.1.2.1: Commanding and reversing**<br>Main Success Scenario: Moving of the Point with multiple Point machines Non 4W [P SD 2.1.2.1.1]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- No end position, or<br>- a Unintended position.<br>**Interaction 2.1.2.1.1.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**par**<br>**2.a1** The Subsystem - Point sends the Command to the 1st Point machine to move the Point to an End position "X".<br>**also par**<br>**opt** [The n-th Point machine has drive capability]<br>**2.b1.a1** The Subsystem - Point sends the Command to the n-th Point machine to move the Point to an End position "X".<br>**end opt**<br>**also par**<br>**2.c1** The Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**end par**<br>**Interaction 2.1.2.1.1.B:**<br>**alt** [The 1st Point machine was previously in an End position or a Unintended position and the n-th is not in a Unintended position]<br>**3.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point is in No end position.<br>**3.a2** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**else alt** [The n-th Point machine was previously in an End position or a Unintended position and the 1st Point machine is not in a Unintended position]<br>**3.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point is in No end position.<br>**3.b2** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**else alt** [The 1st Point machine was previously in a Unintended position and the n-th Point machine was in a Unintended position]<br>**par**<br>**3.c1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point is in No end position.<br>**also par**<br>**3.c1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point is in No end position.<br>**end par**<br>**3.d1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**end alt**<br>**Interaction 2.1.2.1.1.C:**<br>**alt** [The Subsystem - Point is configured for individual drive.]<br>**4.a1** The Subsystem - Point stops moving of the Point for individual drive.<br>**else alt** [The Subsystem - Point is configured for common drive.]<br>**4.b1** The Subsystem - Point stops moving of the Point for common drive.<br>**end alt**<br>**5.** The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br>**6.** When Information_End_Position_Detected has been received from all Point machines, the Subsystem - Point sends a Message to the Subsystem - Electronic Interlocking indicating that the Point is in an End position "X".<br>**Postcondition:**<br>The Subsystem - Point is in an End position "X". | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. The timing related to driving individual (1 to n-th) Point Machines for a single Point is supplier specific and is specified only in a general way. The timing related to individual Point Machines is not specified by EULYNX as part of the application layer, this shall be handled by the physical implementation. | Basic non-4-wire multiple P |



| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6723 | Info | P_UC2.1.2.2: Redrive | The Subsystem-UseCase "P_UC2.1.2.2: Redrive" defines the behaviour of redriving a multiple point machine via non-4-wire interface. Only applicable if the package [Option Redrive] is used in combination with [Basic non-4-wire multiple P]. | Option Redrive |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.7098 | Info | Main Success Scenario: Redrive Point after lost end position [P SD 2.1.2.2.1] | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. Only applicable if the package [Option Redrive] is used in combination with [Basic non-4-wire multiple P]. | Option Redrive |

**P_UC2.1.2.2: Redrive**

Main Success Scenario: Redrive Point after lost end position [P SD 2.1.2.2.1]

**Precondition:**

The Subsystem - Point is in the state INITIALISING or OPERATIONAL.
The Subsystem - Point is in an End position "X".
The last Cd_Move_Point Command received was also for End position "X".
The last Cd_Move_Point Command received occurred since exiting the state BOOTING.
The Subsystem - Point is configured with a non-4-wire interface to the Point machine.
The Subsystem - Point is configured as Redrive point.

**Interaction 2.1.2.2.1.A:**

**alt** [The 1st Point machine was previously in an End position]

**1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point is in No end position.

**opt** [The Subsystem - Point is in the state OPERATIONAL.]

**1.a2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.

**end opt**

**else alt** [The n-th Point machine was previously in an End position]

**1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point is in No end position.

**opt** [The Subsystem - Point is in the state OPERATIONAL.]

**1.b2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.

**end opt**

**end alt**

**Interaction 2.1.2.2.1.B:**

**alt** [The Subsystem - Point is configured for common drive.]

**par**

**2.a1.a1** - The Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.

**also par**

**2.a1.b1** - The Subsystem - Point sends the Command to the 1st Point machine to move the Point to an End position "X".

**also par**

**2.a1.c1** - The Subsystem - Point sends the Command to the n-th Point machine to move the Point to an End position "X".

**end par**

**else alt** [The Subsystem - Point is configured for individual drive.]

**alt** [The 1st Point machine is in No end position]

**2.b1.a1** - The Subsystem - Point sends the Command to the 1st Point machine to move the Point to an End position "X".

**else alt** [The n-th Point machine is in No end position]

**2.b1.b1** - The Subsystem - Point sends the Command to the n-th Point machine to move the Point to an End position "X".

**end alt**

**end alt**

**Interaction 2.1.2.2.1.C:**

**alt** [The Subsystem - Point is configured for individual drive]

**3.a1** - The Subsystem - Point stops moving of the Point for individual drive.

**else alt** [The Subsystem - Point is configured for common drive]

**3.b1** - The Subsystem - Point stops moving of the Point for common drive.

**end alt**

**4.** The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.

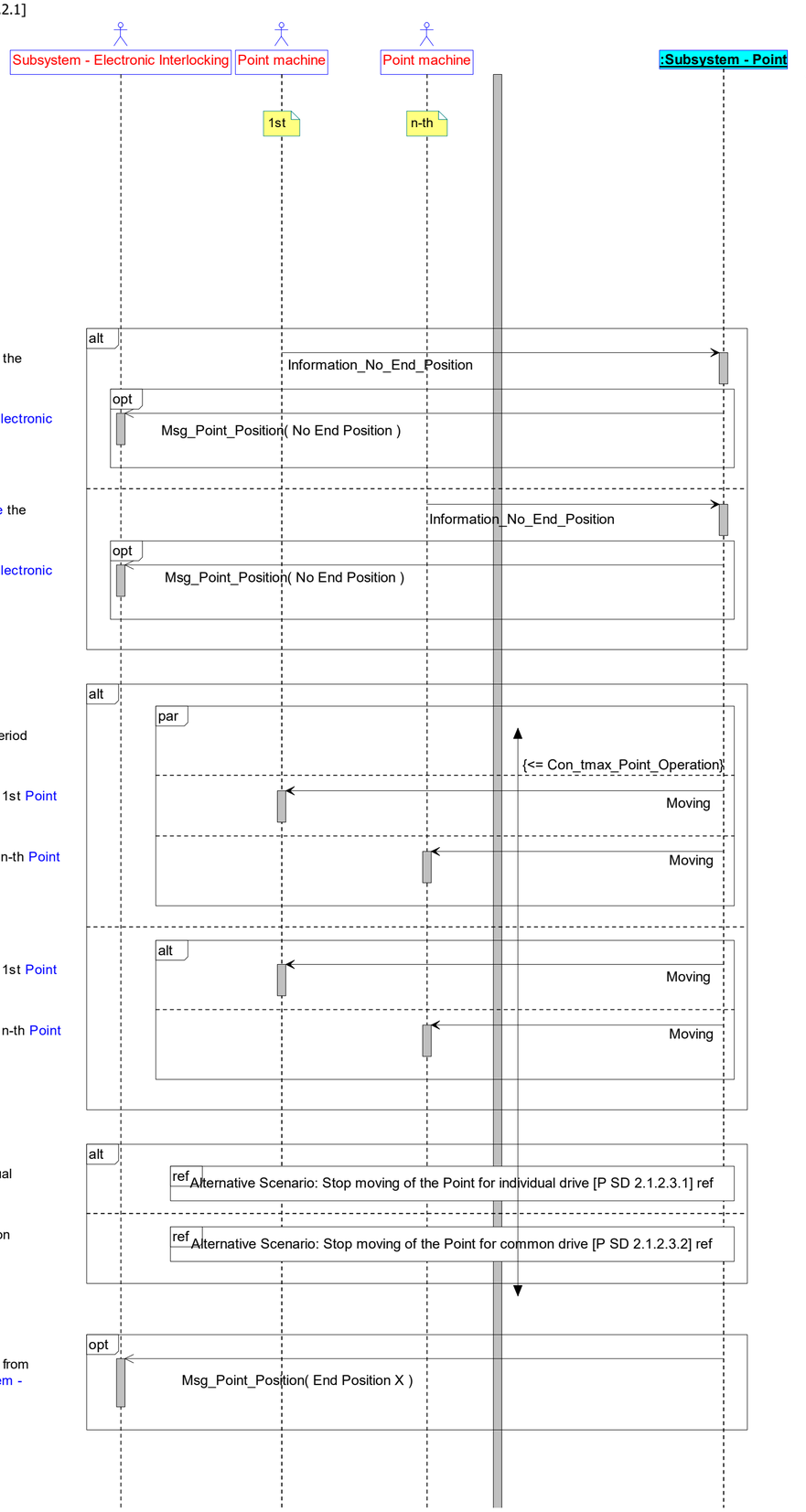
**opt** [The Subsystem - Point is in the state OPERATIONAL.]

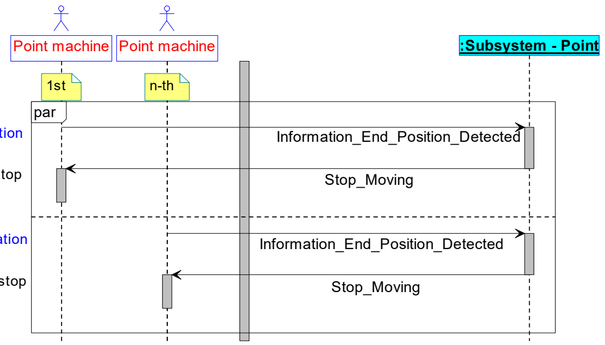
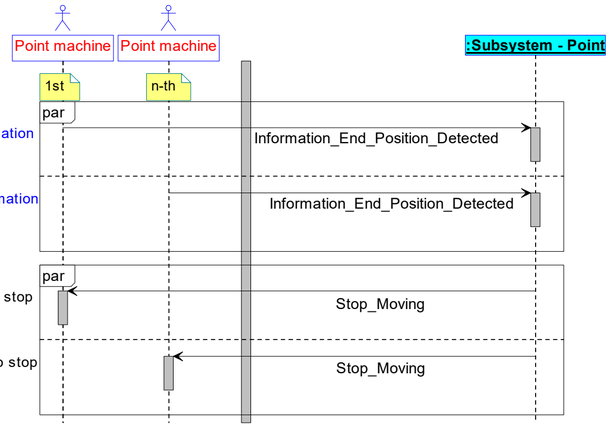
**4.a1** When Information_End_Position_Detected has been received from all Point machines, the Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "X".

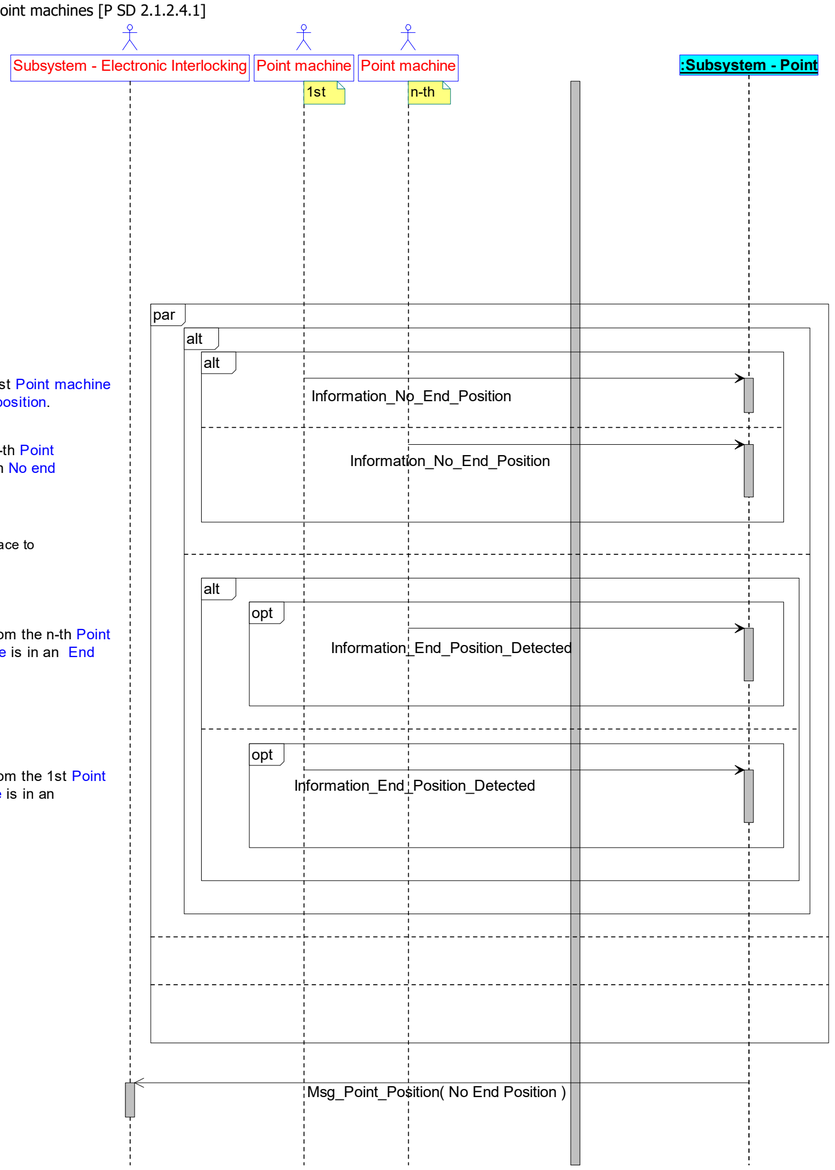
**end opt**

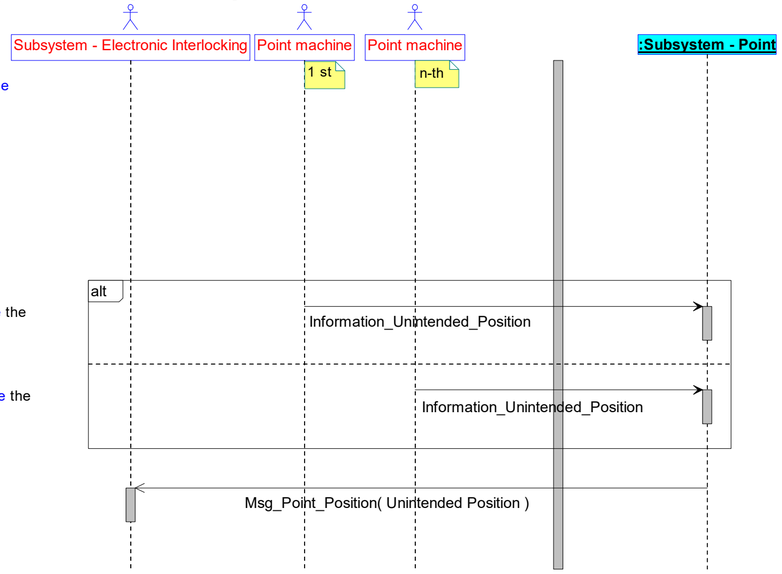
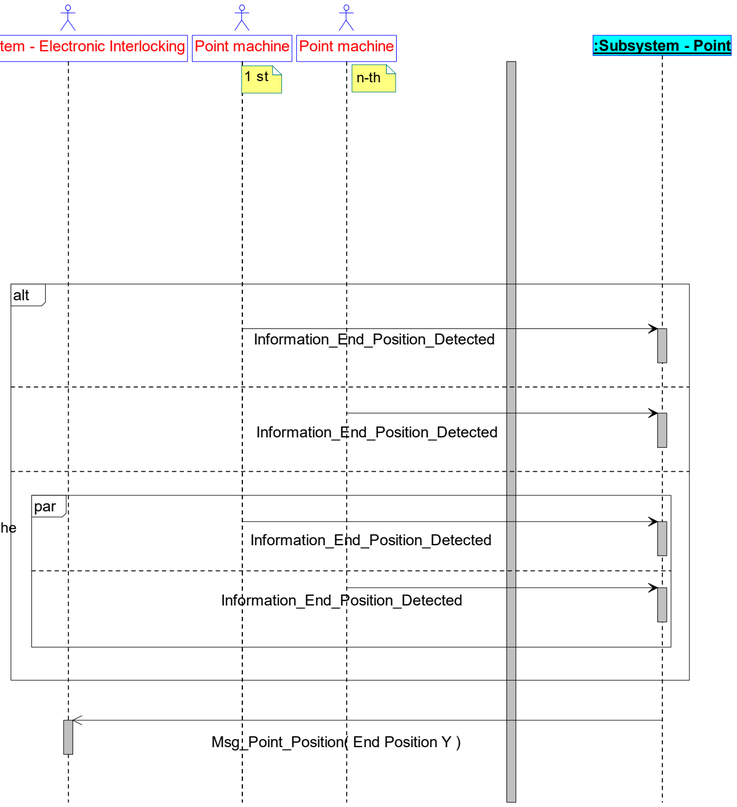
**Postcondition:**

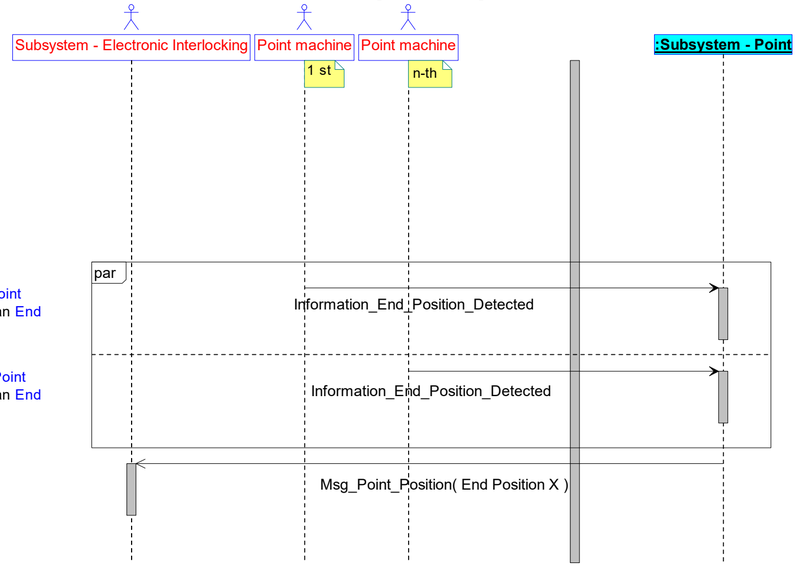
The Subsystem - Point is in an End position "X".

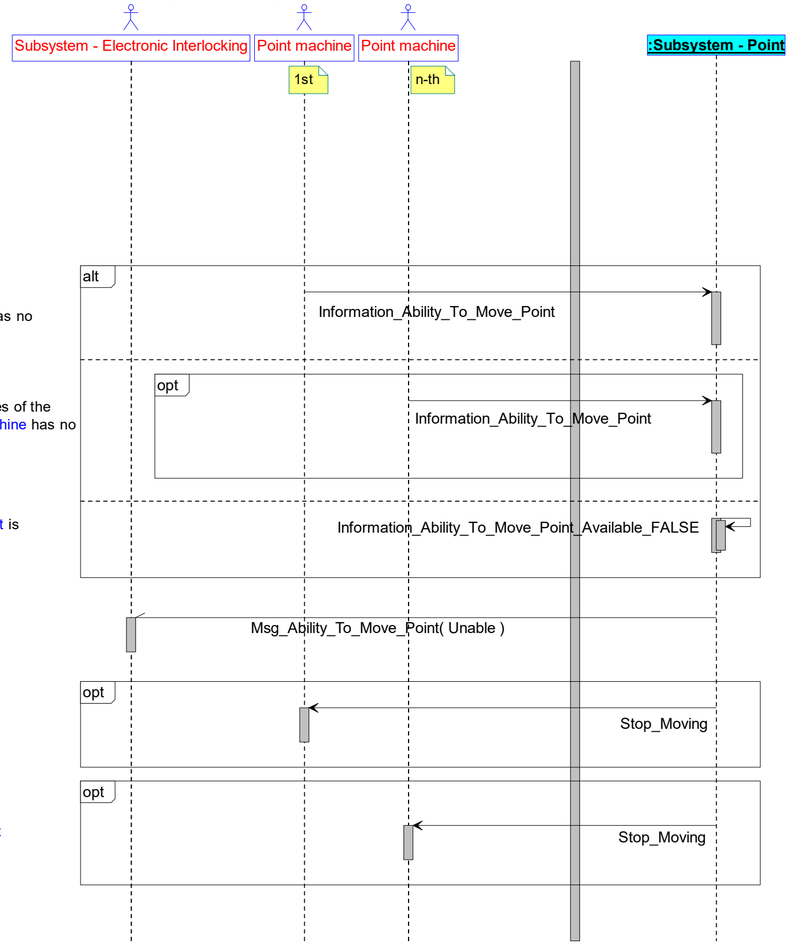


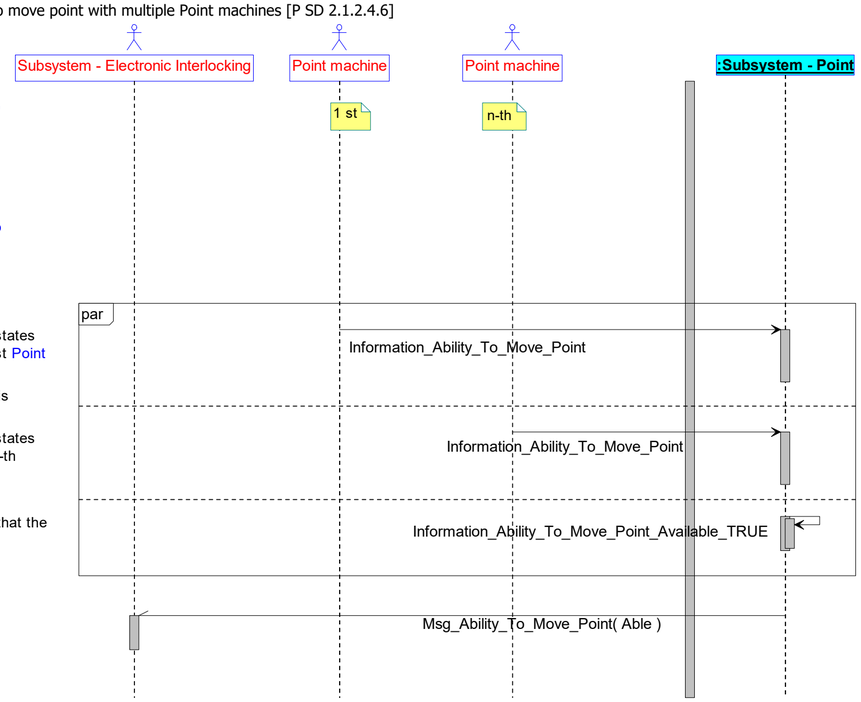
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6726 | Info | P_UC2.1.2.3: Stop point movement | | Basic non-4-wire multiple P |
| Eu.P.6728 | Info | Alternative Scenario: Stop moving of the Point for individual drive [P SD 2.1.2.3.1]<br>**P_UC2.1.2.3: Stop point movement**<br>Alternative Scenario: Stop moving of the Point for individual drive [P SD 2.1.2.3.1]<br>**par**<br>**1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the 1st Point machine is in an End position "X".<br>**1.a2** The Subsystem - Point sends a Command to the 1st Point machine to stop moving the Point.<br>**also par**<br>**1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the n-th Point machine is in an End position "X".<br>**1.b2** The Subsystem - Point sends a Command to the n-th Point machine to stop moving the Point.<br>**end par**<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.<br>The timing related to driving individual (1 to n-th) Point Machines for a single Point is supplier specific and is specified only in a general way. The timing related to individual Point Machines is not specified by EULYNX as part of the application layer, this shall be handled by the physical implementation.<br>This SD is part of [P SD 2.1.2.2.1]. | Basic non-4-wire multiple P |
| Eu.P.6727 | Info | Alternative Scenario: Stop moving of the Point for common drive [P SD 2.1.2.3.2]<br>**P_UC2.1.2.3: Stop point movement**<br>Alternative Scenario: Stop moving of the Point for common drive [P SD 2.1.2.3.2]<br>**par**<br>**1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the 1st Point machine is in an End position "X".<br>**also par**<br>**1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the n-th Point machine is in an End position "X".<br>**end par**<br>**par**<br>**2.a1** The Subsystem - Point sends a Command to the 1st Point machine to stop moving the Point.<br>**also par**<br>**2.b1** The Subsystem - Point sends a Command to the n-th Point machine to stop moving the Point.<br>**end par**<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.<br>The timing related to driving individual (1 to n-th) Point Machines for a single Point is supplier specific and is specified only in a general way. The timing related to individual Point Machines is not specified by EULYNX as part of the application layer, this shall be handled by the physical implementation.<br>This SD is part of [P SD 2.1.2.2.1].<br>Only applicable if the package [Option Common Drive] is used in combination with [Basic non-4-wire multiple P]. | Option Common Drive |
| Eu.P.6729 | Info | P_UC2.1.2.4: Irregularities | The Subsystem-UseCase "P_UC2.1.2.4: Irregularities" defines the behaviour of the Subsystem - Point which works with a multiple point machine via non-4-wire interface, when an irregularity occurs. | Basic non-4-wire multiple P |

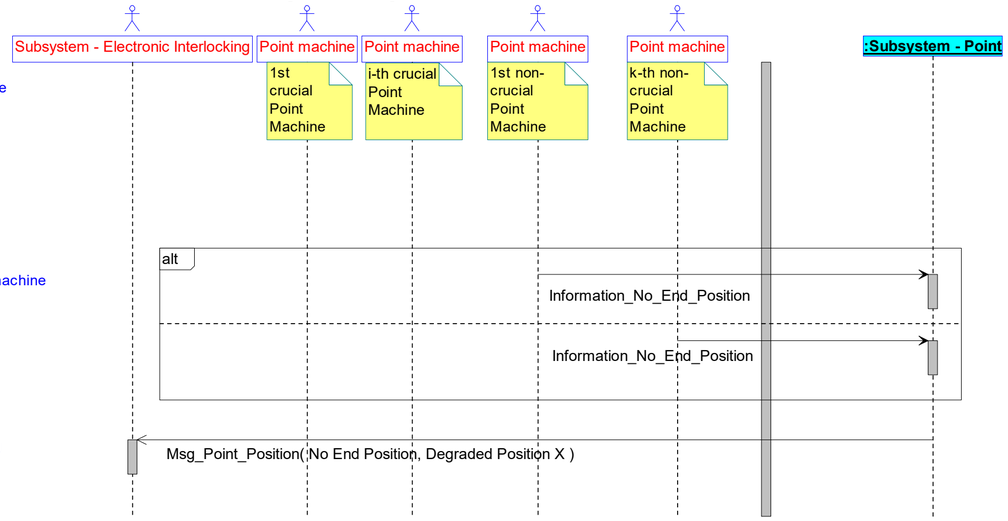
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.5805 | Info | Alternative Scenario: Handle and report No end position with multiple Point machines [P SD 2.1.2.4.1]<br>**P_UC2.1.2.4: Irregularities**<br>Alternative Scenario: Handle and report No end position with multiple Point machines [P SD 2.1.2.4.1]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- a Unintended position<br>**Interaction 2.1.2.4.1.A:**<br>**par**<br>**alt** [Any Point machine is in an End position]<br>**alt** [The 1st Point machine is in No end position]<br>**1.a1.a1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the 1st Point machine is in No end position.<br>**else alt** [The n-th Point machine is in No end position]<br>**1.a1.a1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the n-th Point machine is in No end position.<br>**end alt**<br>**else alt** [The Subsystem - Point is not configured with a 4-wire interface to the Point machine]<br>**alt** [The 1st Point machine is in an End position "Y".]<br>**opt** [The n-th Point machine is in an End position "X".]<br>**1.a1.b1.a1.a1** - The Subsystem - Point receives from the n-th Point machine the Information that the n-th Point machine is in an End position "X".<br>**end opt**<br>**else alt** [The n-th Point machine is in an End position "Y"]<br>**opt** [The 1st Point machine is in an End position "X".]<br>**1.a1.b1.b1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the 1st Point machine is in an End position "X".<br>**end opt**<br>**end alt**<br>**end alt**<br>**also par**<br>**1.b1** - The 1st Point machine is NOT in a Unintended position.<br>**also par**<br>**1.c1** - The n-th Point machine is NOT in a Unintended position.<br>**end par**<br>**Interaction 2.1.2.4.1.B:**<br>**2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**Postcondition:**<br>The Subsystem - Point is in No end position.<br>  <br> | | Basic non-4-wire multiple P |

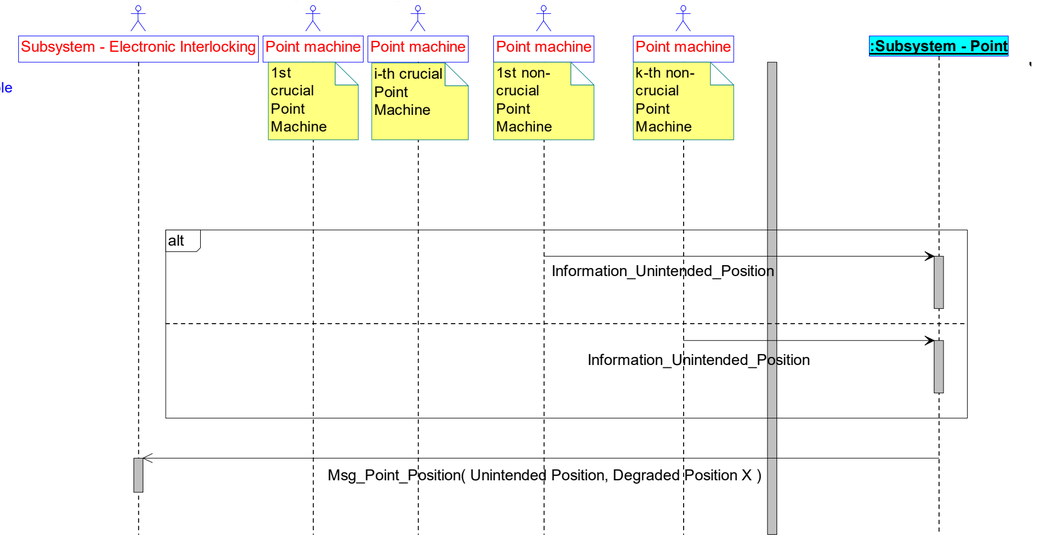
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.5808 | Info | Alternative Scenario: Handle and report Unintended position with multiple Point machines [P SD 2.1.2.4.2]<br>**P_UC2.1.2.4: Irregularities**<br>Alternative Scenario: Handle and report Unintended position with multiple Point machines [P SD 2.1.2.4.2]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- No end position.<br>**Interaction 2.1.2.4.2.A:**<br>**alt** [The 1st Point machine is in a End position "Y" or No end position.]<br>**1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point machine is in a Unintended position.<br>**else alt** [The n-th Point machine is in a End position "Y" or No end position.]<br>**1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point machine is in a Unintended position.<br>**end alt**<br>**Interaction 2.1.2.4.2.B:**<br>**2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in a Unintended position.<br>**Postcondition:**<br>The Subsystem - Point is in a Unintended position.<br>  <br> | | Basic non-4-wire multiple P |
| Eu.P.5803 | Info | Alternative Scenario: Handle and report End Position with multiple Point machines [P SD 2.1.2.4.3]<br>**P_UC2.1.2.4: Irregularities**<br>Alternative Scenario: Handle and report End Position with multiple Point machines [P SD 2.1.2.4.3]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine<br>The Subsystem - Point is in:<br>- a No end position, or<br>- an Unintended position.<br>**Interaction 2.1.2.4.3.A:**<br>**alt** [The 1st Point machine is an No end position or a Unintended position and the n-th Point machine is in an End position "Y".]<br>**1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point machine is in an End position "Y".<br>**else alt** [The n-th Point machine is an No end position or a Unintended position and the 1st Point machine is in an End position "Y".]<br>**1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point machine is in an End position "Y".<br>**else alt** [The n-th Point machine and the 1st Point machine are in an No end position or a Unintended position.]<br>par<br>**1.c1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point machine is in an End position "Y".<br>also par<br>**1.c1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point machine is in an End position "Y".<br>end par<br>**end alt**<br>**Interaction 2.1.2.4.3.B:**<br>**2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "Y".<br>**Postcondition:**<br>The Subsystem - Point is in an End position "Y".<br>  <br> | | Basic non-4-wire multiple P |

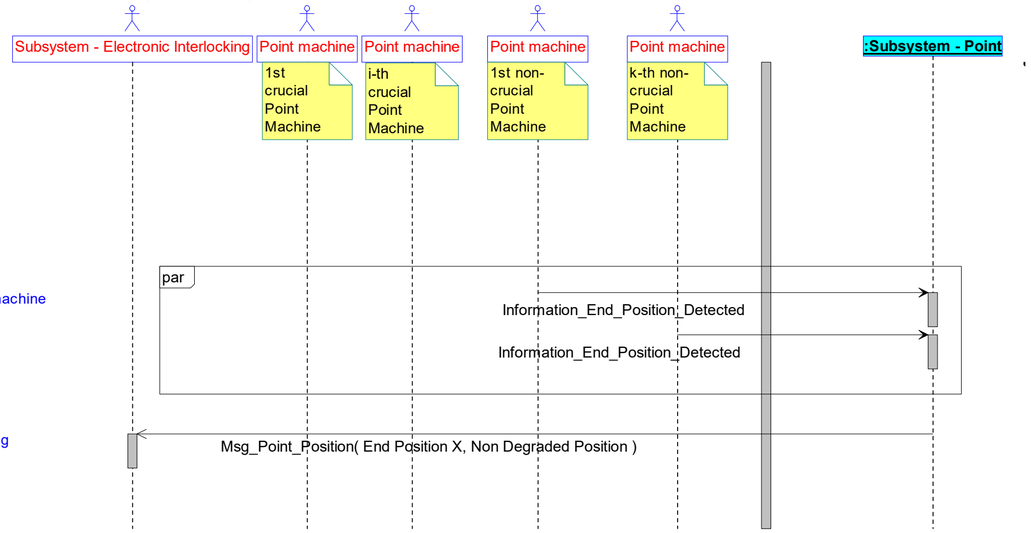
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.5802 | Info | Alternative Scenario: Handle and report opposite End Position with multiple Point machines via non 4-wire interface [P SD 2.1.2.4.4]<br>**P_UC2.1.2.4: Irregularities**<br>Alternative Scenario: Handle and report opposite End Position with multiple Point machines via non 4-wire interface [P SD 2.1.2.4.4]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is in an End position "Y".<br>**Interaction 2.1.2.4.4.A:**<br>**par**<br>**1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point machine is in an End position "X".<br>**also par**<br>**1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point machine is in an End position "X".<br>**end par**<br>**2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "X".<br>**Postcondition:**<br>The Subsystem - Point is in End position "X".<br>  <br> | Whether this SD can occur, depends on the national implementation of P3. | Basic non-4-wire multiple P |

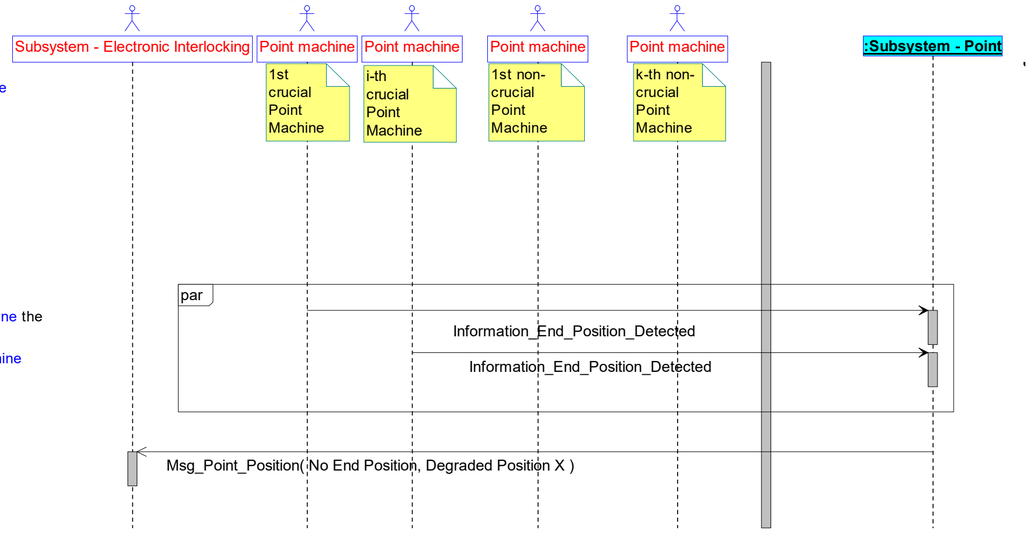
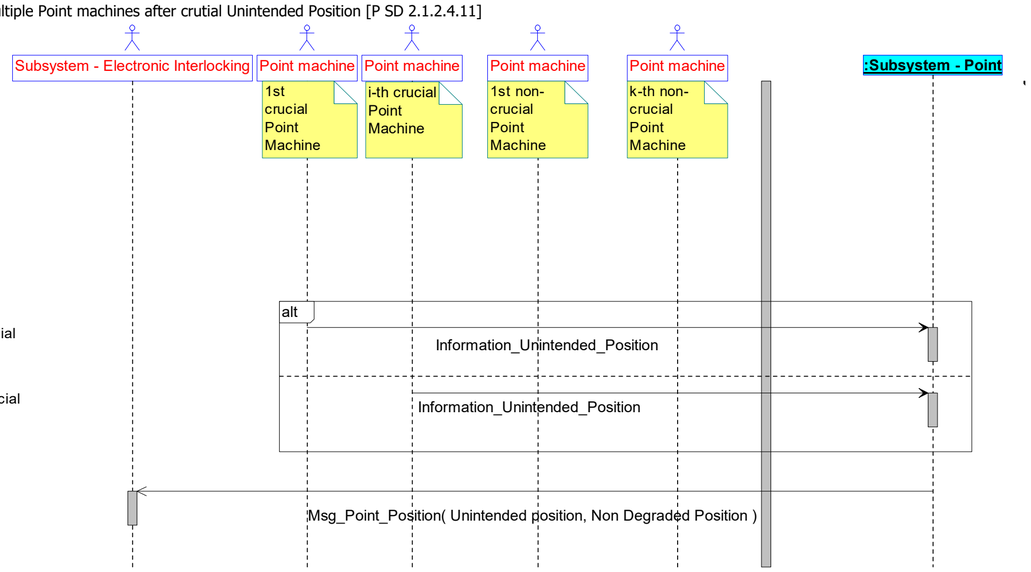
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.5804 | Info | Alternative Scenario: Handle and report loss of ability to move point with multiple Point machines [P SD 2.1.2.4.5]<br>**P_UC2.1.2.4: Irregularities**<br>Alternative Scenario: Handle and report loss of ability to move point with multiple Point machines [P SD 2.1.2.4.5]<br>**Precondition**:<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is configured to observe the Ability to move point.<br>The Subsystem - Point is Able to move point.<br>**Interaction 2.1.2.4.5.A**:<br>**alt** [The 1st Point machine is Unable to move point]<br>**1.a1** - The Subsystem - Point detects from the switch states of the Interface P3 to the 1st Point machine that the 1st Point machine has no Ability to move point.<br>**else alt** [The n-th Point machine is configured with full functionality]<br>**opt** [The n-th Point machine is Unable to move point]<br>**1.b1.a1** - The Subsystem - Point detects from the switch states of the Interface P3 to the n-th Point machine that the n-th Point machine has no Ability to move point.<br>**end opt**<br>**else alt** [internal trigger indicates non-ability to move point]<br>**1.c1** - The Subsystem - Point internal trigger indicates that the Point is Unable to move point.<br>**end alt**<br>**Interaction 2.1.2.4.5.B**:<br>**2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that it is Unable to move point.<br>**Interaction 2.1.2.4.5.C**:<br>**opt** [The1st point machine is in Moving]<br>**3.a1** - The Subsystem - Point sends the Command to the 1st Point machine to Stop moving the 1st Point machine.<br>**end opt**<br>**opt** [The n-th Point machine is configured with full functionality and is in Moving]<br>**4.a1** - The Subsystem - Point sends the Command to the n-th Point machine to Stop moving the n-th Point machine.<br>**end opt**<br>**Postcondition:**<br>The Subsystem - Point is Unable to move point.<br>  <br> | Only applicable if the package [Option Able to move] is used in combination with [Basic non-4-wire multiple P]. | Option Able to move |

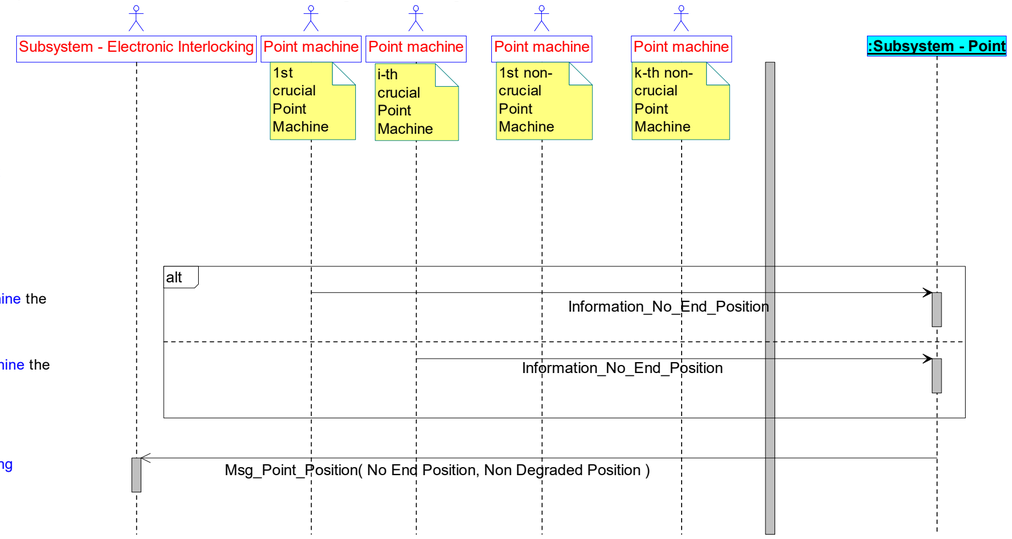
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.5374 | Info | Alternative Scenario: Handle and report restoring of Ability to move point with multiple Point machines [P SD 2.1.2.4.6]<br>**P_UC2.1.2.4: Irregularities**<br>Alternative Scenario: Handle and report restoring of Ability to move point with multiple Point machines [P SD 2.1.2.4.6]<br>**Precondition**:<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is configured to observe the Ability to move point.<br>The Subsystem - Point is Unable to move point.<br>**Interaction 2.1.2.4.6.A**:<br>**par** [The 1st Point machine is Able to move point]<br>**1.a1** - The Subsystem - Point detects from the switch states of the Interface P3 to the 1st Point machine that the 1st Point machine has regained Ability to move point.<br>**also par** [The n-th Point machine is Able to move point and is configured with full functionality]<br>**1.b1** - The Subsystem - Point detects from the switch states of the Interface P3 to the n-th Point machine that the n-th Point machine has regained Ability to move point.<br>**also par** [internal trigger indicates ability to move point]<br>**1.c1** - The Subsystem - Point internal trigger indicates that the Point is Able to move point.<br>**end par**<br>**Interaction 2.1.2.4.6.B**:<br>**2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that it is Able to move point.<br>**Postcondition:**<br>The Subsystem - Point is Able to move point.<br>  <br> | Only applicable if the package [Option Able to move] is used in combination with [Basic non-4-wire multiple P]. | Option Able to move |

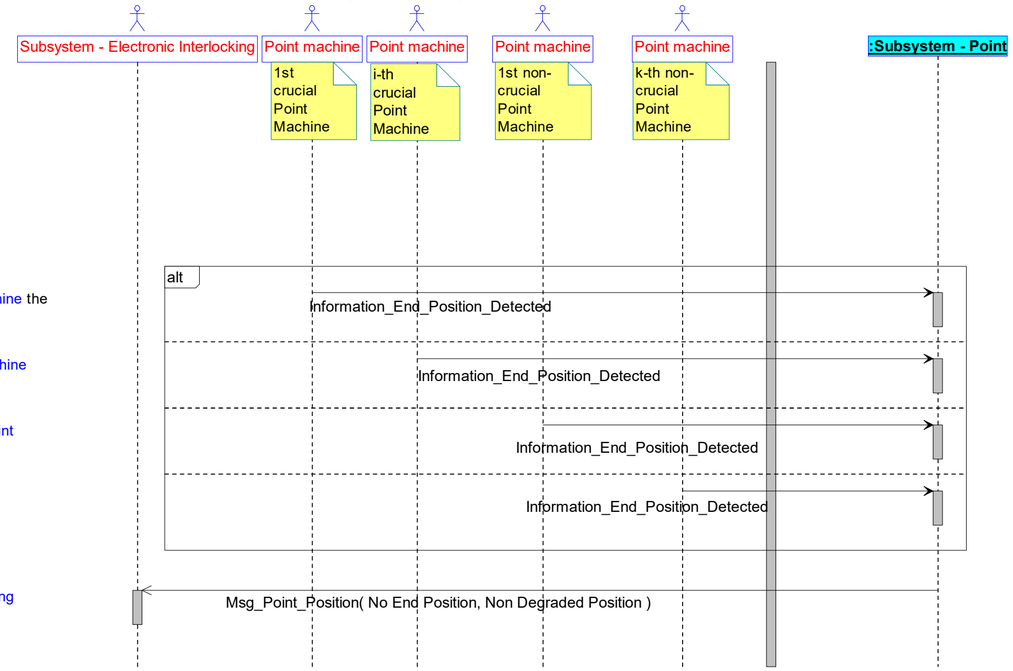
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.7372 | Info | Alternative Scenario: Handle and report Degraded Point Position with multiple Point machines after End Position case 1 [P SD 2.1.2.4.7]<br>**P_UC2.1.2.4: Irregularities**<br>Alternative Scenario: Handle and report Degraded Point Position with multiple Point machines after End Position case 1 [P SD 2.1.2.4.7]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>Subsystem - Point in an End position "X".<br>**Interaction 2.1.2.4.7.A**<br>**alt**<br>**1.a1** - The Subsystem - Point receives from the 1st non-crucial Point machine the Information that the Point machine is in No end position.<br>**else alt**<br>**1.b1** - The Subsystem - Point receives from the k-th non-crucial Point machine the Information that the Point machine is in No end position.<br>**end alt**<br>**Interaction 2.1.2.4.7.B**<br>**2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in a Degraded point position "X"<br>**Postcondition**:<br>The Subsystem - Point is in a Degraded point position "X"<br>  <br> | | Basic non-4-wire multiple P |

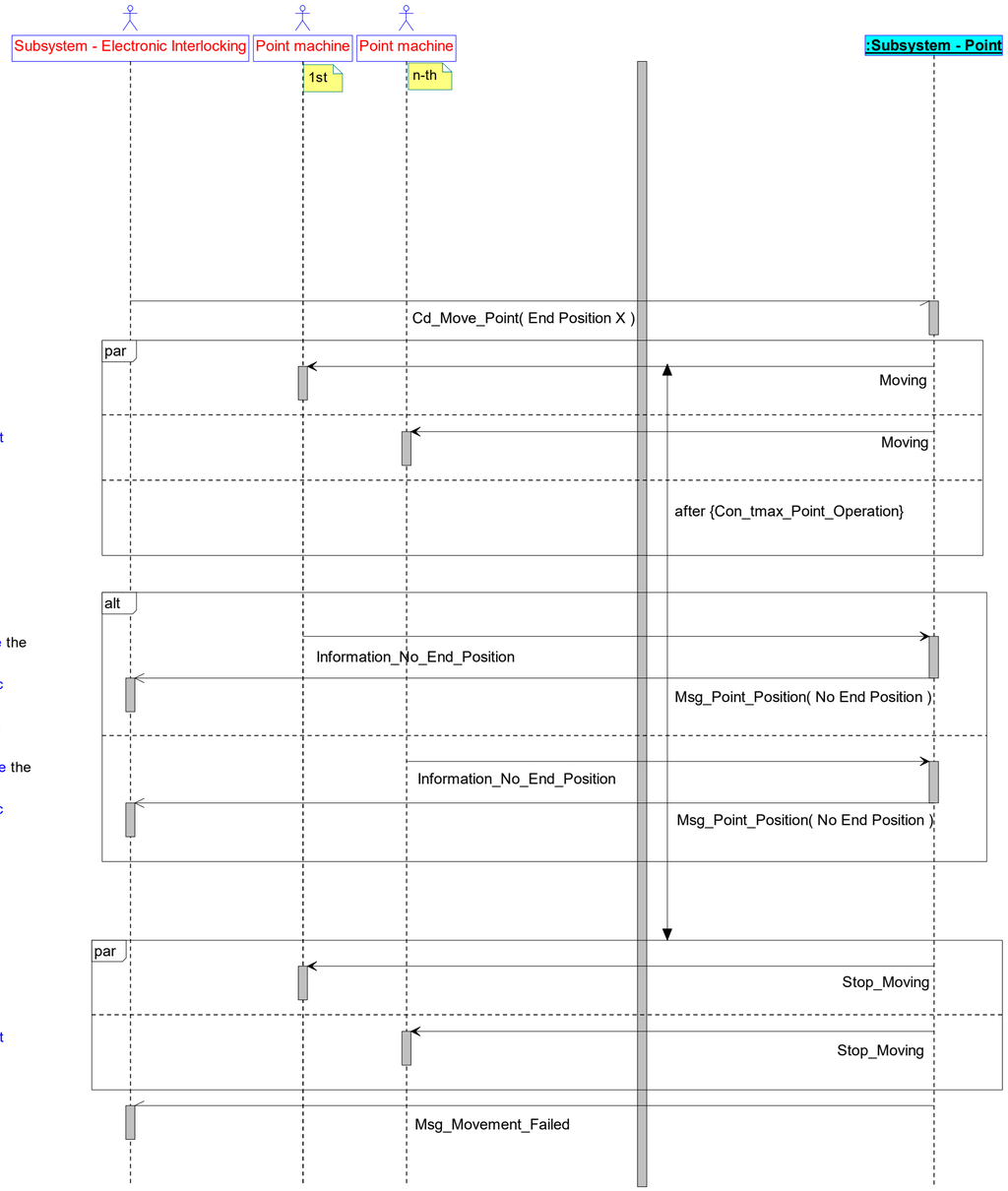
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.7373 | Info | Alternative Scenario: Handle and report Degraded Point Position with multiple Point machines after End Position case 2 [P SD 2.1.2.4.8]<br>**P_UC2.1.2.4: Irregularities**<br>Alternative Scenario: Handle and report Degraded Point Position with multiple Point machines after End Position case 2 [P SD 2.1.2.4.8]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is in a Non degraded position.<br>The Subsystem - Point is in an End position "X".<br>**Interaction 2.1.2.4.8.A**<br>**alt**<br>**1.a1** - The Subsystem - Point receives from the non-crucial 2nd Point machine the Information that the non-crucial 2nd Point machine is in a Unintended position.<br>**else alt**<br>**1.b1** - The Subsystem - Point receives from the non-crucial n-th Point machine the Information that the non-crucial n-th Point machine is in a Unintended position.<br>**end alt**<br>**Interaction 2.1.2.4.8.B**<br>**2.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in a Degraded point position "X"<br>**Postcondition**:<br>The Subsystem - Point is in a Degraded point position "X"<br>  <br> | | Basic non-4-wire multiple P |

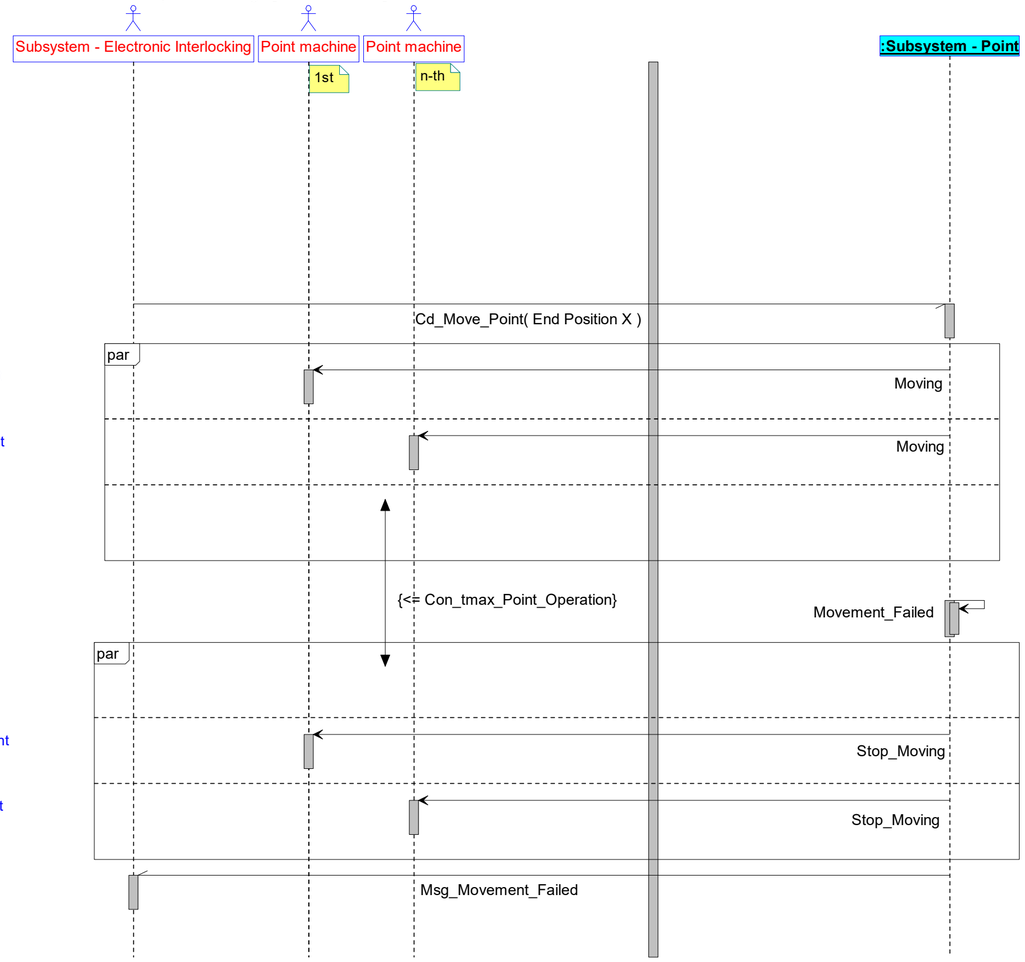
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.7374 | Info | Alternative Scenario: Handle and report non-degraded Point Position with multiple Point machines [P SD 2.1.2.4.9]<br>**P_UC2.1.2.4: Irregularities**<br>Alternative Scenario: Handle and report non-degraded Point Position with multiple Point machines [P SD 2.1.2.4.9]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is in a Degraded point position "X".<br>One or more non-crucial Point machine are in No end position.<br>**Interaction 2.1.2.4.9.A**<br>**par**<br>**1.a1** - The Subsystem - Point receives from the 1st non-crucial Point machine the Information that the Point machine is in End position "X".<br>**1.b1** - The Subsystem - Point receives from the k-th non-crucial Point machine the Information that the Point machine is in End position "X".<br>**end par**<br>**Interaction 2.1.2.4.9.B**<br>**2.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is not in a Degraded point position "X"<br>**Postcondition**:<br>The Subsystem - Point is in an End position "X".<br>The Subsystem - Point is in a Non degraded position.<br>  <br> | | Basic non-4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.7371 | Info | Alternative Scenario: Handle and report Degraded Point Position with multiple Point machines after crucial End Position [P SD 2.1.2.4.10]<br>**P_UC2.1.2.4: Irregularities**<br>Alternative Scenario: Handle and report Degraded Point Position with multiple Point machines after crucial End Position [P SD 2.1.2.4.10]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is in a Non degraded position.<br>The Subsystem - Point is in No end position.<br>One or more non-crucial Point machine are in No end position.<br>**Interaction 2.1.2.4.10.A**<br>**par**<br>**1.a1** - The Subsystem - Point receives from the crucial 1st Point machine the Information that the Point machine is in an End position "X".<br>**1.b2** - The Subsystem - Point receives from the n-th crucial Point machine the Information that the Point machine is in an End position "X".<br>**end par**<br>**Interaction 2.1.2.4.10.B**<br>**2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in a Degraded point position "X"<br>**Postcondition**:<br>The Subsystem - Point is in a Degraded point position "X"<br>  <br> | | Basic non-4-wire multiple P |
| Eu.P.7377 | Info | Alternative Scenario: Handle and report non-degraded Point Position with multiple Point machines after crutial Unintended Position [P SD 2.1.2.4.11]<br>**P_UC2.1.2.4: Irregularities**<br>Alternative Scenario: Handle and report non-degraded Point Position with multiple Point machines after crutial Unintended Position [P SD 2.1.2.4.11]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is in a Degraded point position "X"<br>One or more non-crucial Point machine are in No end position or a Unintended position.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is End position "X".<br>**Interaction 2.1.2.4.11.A**<br>**alt**<br>**1.a1** - The Subsystem - Point receives the Information that the 1st crucial Point machine is in an detected Unintended position.<br>**else alt**<br>**1.b1** - The Subsystem - Point receives the Information that the n-th crucial Point machine is in an detected Unintended position.<br>**end alt**<br>**Interaction 2.1.2.4.11.B**<br>**2.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is not in a Degraded point position "X"<br>**Postcondition**:<br>The Subsystem - Point is in a Non degraded position.<br>  <br> | | Basic non-4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.7375 | Info | Alternative Scenario: Handle and report non-degraded Point Position with multiple Point machines after crucial No End Position [P SD 2.1.2.4.12]<br><br>**P_UC2.1.2.4: Irregularities**<br><br>Alternative Scenario: Handle and report non-degraded Point Position with multiple Point machines after crucial No End Position [P SD 2.1.2.4.12]<br><br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is in a Degraded point position "X".<br>One or more non-crucial Point machine are in No end position.<br><br>**Interaction 2.1.2.4.12.A**<br><br>**alt**<br>**1.b1** - The Subsystem - Point receives from the 1st crucial Point machine the Information that the Point machine is in No end position.<br>**else alt**<br>**1.c1** - The Subsystem - Point receives from the n-th crucial Point machine the Information that the Point machine is in No end position<br>**end alt**<br><br>**Interaction 2.1.2.4.12.B**<br>**2.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is not in a Degraded point position "X".<br><br>**Postcondition**:<br>The Subsystem - Point is in a Non degraded position.<br><br>  <br> | | Basic non-4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.7376 | Info | Alternative Scenario: Handle and report non-degraded Point Position with multiple Point machines after crucial Opposit Position [P SD 2.1.2.4.13]<br>**P_UC2.1.2.4: Irregularities**<br>Alternative Scenario: Handle and report non-degraded Point Position with multiple Point machines after crucial Opposit Position [P SD 2.1.2.4.13]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is in a Degraded point position "X".<br>One or more non-crucial Point machine are in No end position.<br>**Interaction 2.1.2.4.13.A**<br>**alt**<br>**1.a1** - The Subsystem - Point receives from the 1st crucial Point machine the Information that the Point machine is in End position "Y".<br>**else alt**<br>**1.b1** - The Subsystem - Point receives from the n-th crucial Point machine the Information that the Point machine is in End position "Y".<br>**else alt**<br>**1.c1** - The Subsystem - Point receives from the 1st 1st non-crucial Point machine the Information that the Point machine is in End position "Y".<br>**else alt**<br>**1.d1** - The Subsystem - Point receives from the n-th non/crucial Point machine the Information that the Point machine is in End position "Y".<br>**end alt**<br>**Interaction 2.1.2.4.13.B**<br>**2.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is not in a Degraded point position "X"<br>**Postcondition**:<br>The Subsystem - Point is in a Non degraded position.<br>  <br> | | Basic non-4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.1475 | Info | Alternative Scenario: Handle and report Point operation timeout with multiple Point machines with position change [P SD 2.1.2.4.14]<br>**P_UC2.1.2.4: Irregularities**<br>Alternative Scenario: Handle and report Point operation timeout with multiple Point machines with position change [P SD 2.1.2.4.14]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y",<br>- No end position, or<br>- a Unintended position.<br>**Interaction 2.1.2.4.14.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**par**<br>**2.a1** The Subsystem - Point sends the Command to the 1st Point machine to move the 1st Point machine to an End position "X".<br>**also par**<br>**2.b1** The Subsystem - Point sends the Command to the n-th Point machine to move the n-th Point machine to an End position "X".<br>**also par**<br>**2.c1** The Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**end par**<br>**Interaction 2.1.2.4.14.B:**<br>**alt** [The 1st Point machine was previously in an End position or a Unintended position ]<br>**3.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the 1st Point machine is in No end position.<br>**3.a2** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**else alt** [The n-th Point machine was previously in an End position or a Unintended position]<br>**3.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the n-th Point machine is in No end position.<br>**3.b2** The Subsystem - Point reports to the Subsystem - Electronic Interlocking indicating that the Point is in No end position.<br>**end alt**<br>**Interaction 2.1.2.4.14.C:**<br>**4.** - The Subsystem - Point detects that the time period Con_tmax_Point_Operation has exceeded.<br>**par**<br>**5.a1** The Subsystem - Point sends the Command to the 1st Point machine to stop moving the 1st Point machine.<br>**also par**<br>**5.b1** The Subsystem - Point sends the Command to the n-th Point machine to stop moving the n-th Point machine.<br>**end par**<br>**6.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred.<br>**Postcondition:**<br>The Subsystem - Point is in No end position.<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. The timing related to driving individual (1 to n-th) Point Machines for a single Point is supplier specific and is specified only in a general way. The timing related to individual Point Machines is not specified by EULYNX as part of the application layer, this shall be handled by the physical implementation. | Basic non-4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6730 | Info | Alternative Scenario: Handle and report failed movement with multiple Point machines without position change [P SD 2.1.2.4.15]<br>**P_UC2.1.2.4: Irregularities**<br>Alternative Scenario: Handle and report failed movement with multiple Point machines without position change [P SD 2.1.2.4.15]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y",<br>- No end position, or<br>- a Unintended position.<br>**Interaction 2.1.2.4.15.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**par**<br>**2.a1** The Subsystem - Point sends the Command to the 1st Point machine to move the 1st Point machine to an End position "X".<br>**also par**<br>**2.b1** The Subsystem - Point sends the Command to the n-th Point machine to move the n-th Point machine to an End position "X".<br>**also par**<br>**2.c1** The Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**end par**<br>**Interaction 2.1.2.4.15.B:**<br>**3.** - A failure occurred during the movement resulting in a Failed Movement.<br>**par**<br>**4.a1** - The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br>**also par**<br>**4.b1** - The Subsystem - Point sends the Command to the 1st Point machine to stop moving the 1st Point machine.<br>**also par**<br>**4.c1** The Subsystem - Point sends the Command to the n-th Point machine to stop moving the n-th Point machine.<br>**end par**<br>**5.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred.<br>**Postcondition:**<br>---<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. The timing related to driving individual (1 to n-th) Point Machines for a single Point is supplier specific and is specified only in a general way. The timing related to individual Point Machines is not specified by EULYNX as part of the application layer, this shall be handled by the physical implementation. | Basic non-4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.7099 | Info | Alternative Scenario: Handle Point operation timeout during Redrive [P SD 2.1.2.4.16] | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. Only applicable if the package [Option Redrive] is used in combination with [Basic non-4-wire multiple P]. | Option Redrive |

**P_UC2.1.2.4: Irregularities**

Alternative Scenario: Handle Point operation timeout during Redrive [P SD 2.1.2.4.16]

**Precondition:**

The Subsystem - Point is in the state INITIALISING or OPERATIONAL.
The Subsystem - Point is in an End position "X"
The last Cd_Move_Point Command received was also for End position "X"
The last Cd_Move_Point Command received occurred since exiting the state BOOTING.
The Subsystem - Point is configured with a non-4-wire interface to the Point machine.
The Subsystem - Point is configured as Redrive point.

**Interaction 2.1.2.4.16.A:**

**alt** [The 1st Point machine was previously in an End position]

**1.a1** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in No end position.

**opt** [The Subsystem - Point is in the state OPERATIONAL.]

**1.a2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.

**end opt**

**else alt** [The n-th Point machine was previously in an End position]

**1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point is in No end position.

**opt** [The Subsystem - Point is in the state OPERATIONAL.]

**1.b2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.

**end opt**

**end alt**

**Interaction 2.1.2.4.16.B:**

**par** [The Subsystem - Point is configured for common drive.]

**2.a1** - The Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation after a moving has started.

**also par**

**2.b1** - The Subsystem - Point sends the Command to the 1st Point machine to move the Point to an End position "X".

**2.b2** - The Subsystem - Point sends the Command to the n-th Point machine to move the Point to an End position "X".

**also par** [The Subsystem - Point is configured for individual drive.]

**alt** [The 1st Point machine is in No end position]

**2.c1.a1** - The Subsystem - Point sends the Command to the 1st Point machine to move the Point to an End position "X".

**else alt** [The n-th Point machine is in No end position]

**2.c1.a2** - The Subsystem - Point sends the Command to the n-th Point machine to move the Point to an End position "X".

**end alt**

**end par**

**Interaction 2.1.2.4.16.C:**

**3.** - The Subsystem - Point detects that the time period Con_tmax_Point_Operation has exceeded.

**par**

**4.a1** The Subsystem - Point sends the Command to the 1st Point machine to stop moving the 1st Point machine.

**also par**

**4.b1** The Subsystem - Point sends the Command to the n-th Point machine to stop moving the n-th Point machine.

**end par**

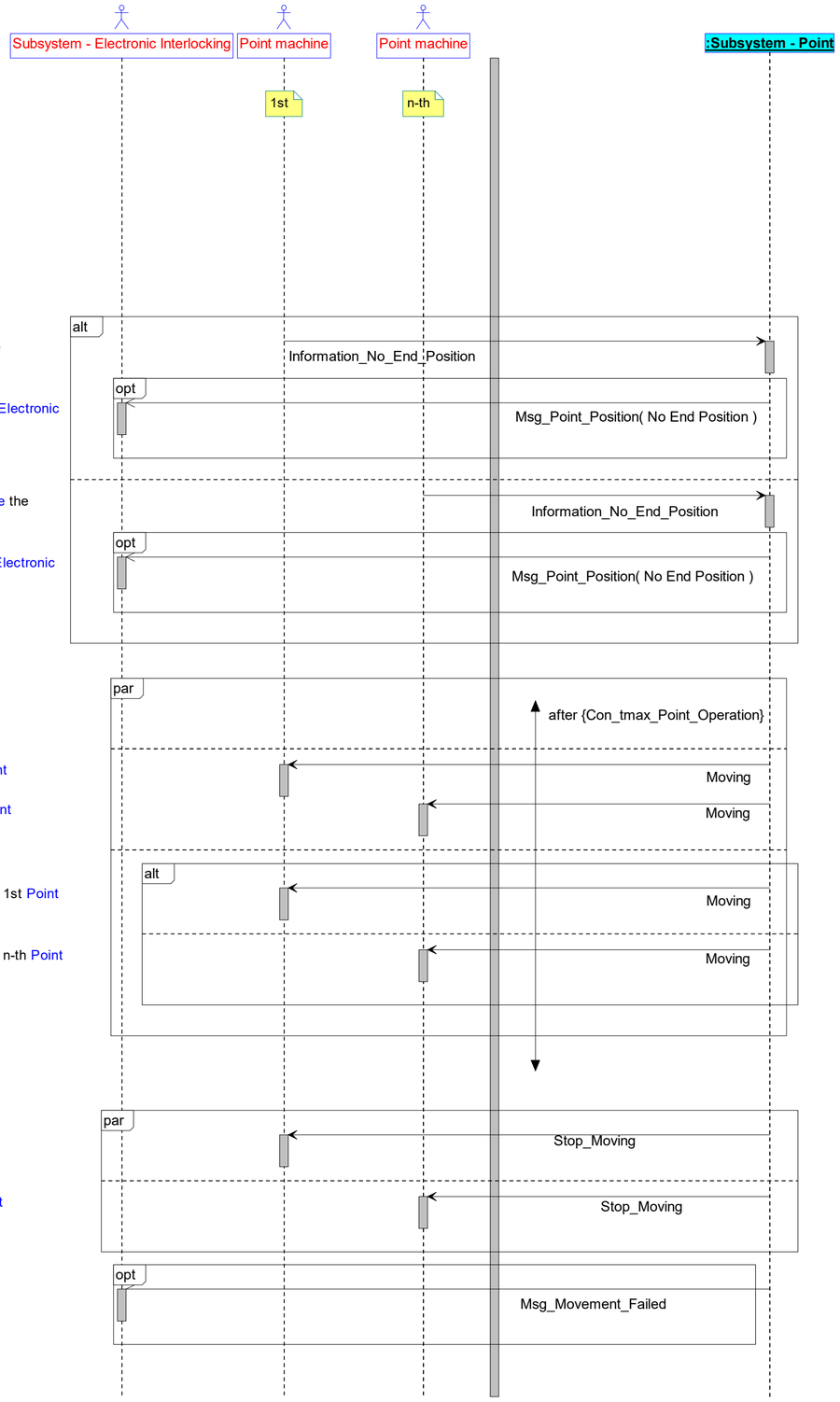
**opt** [The Subsystem - Point is in the state OPERATIONAL.]

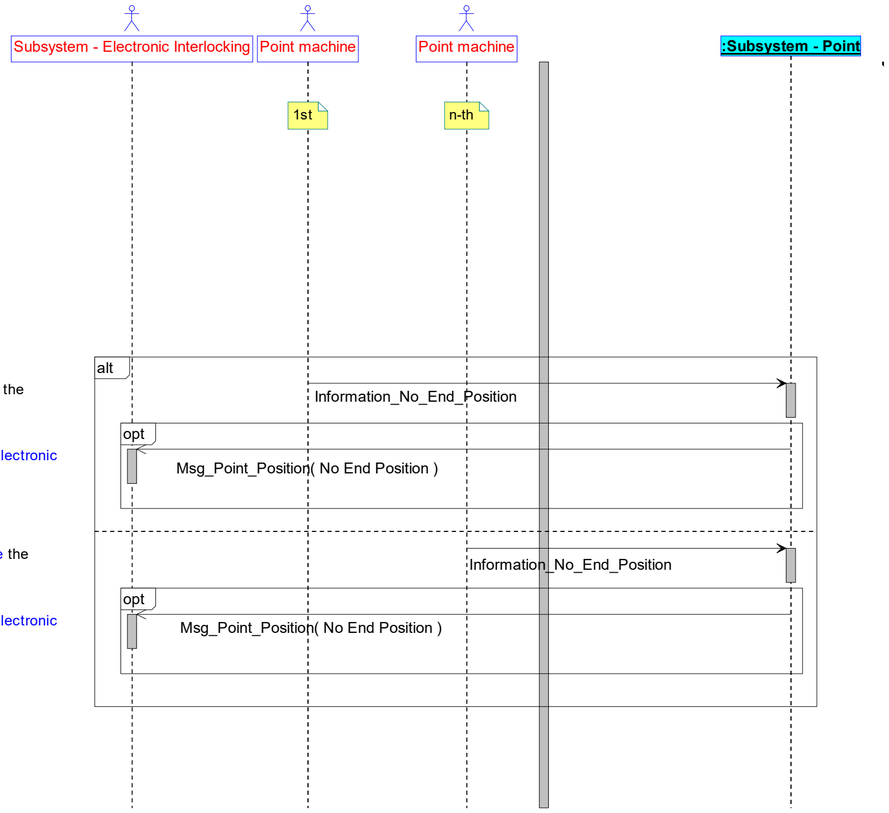
**5.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred.

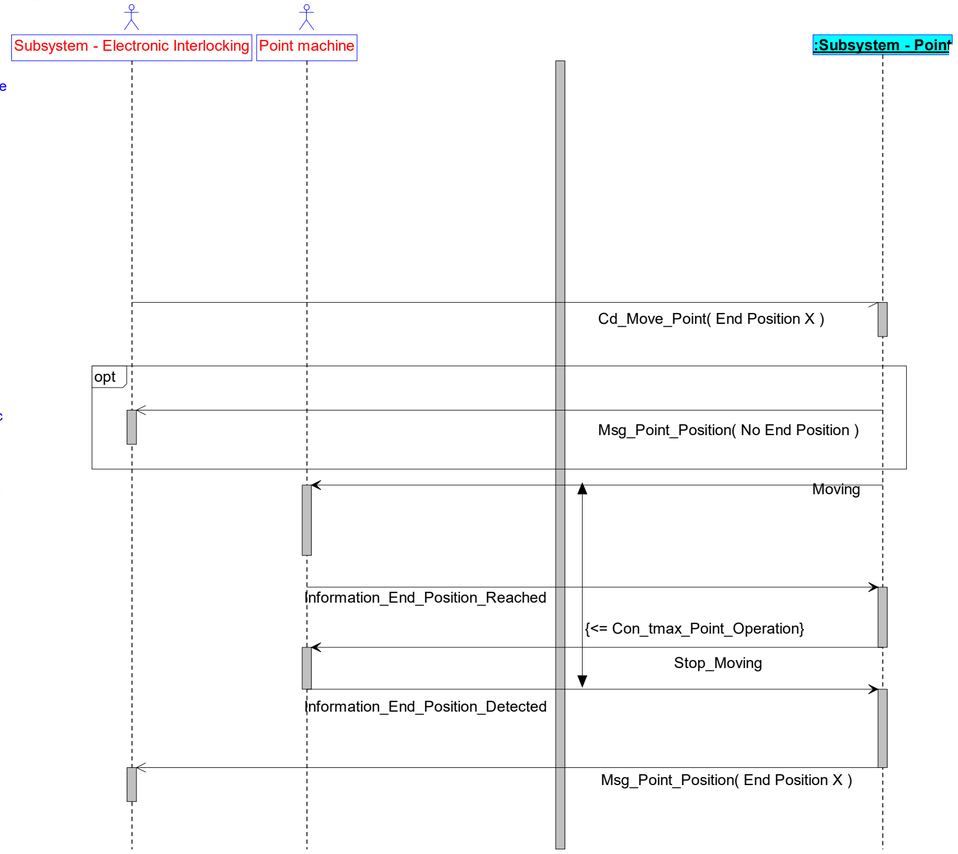
**end opt**

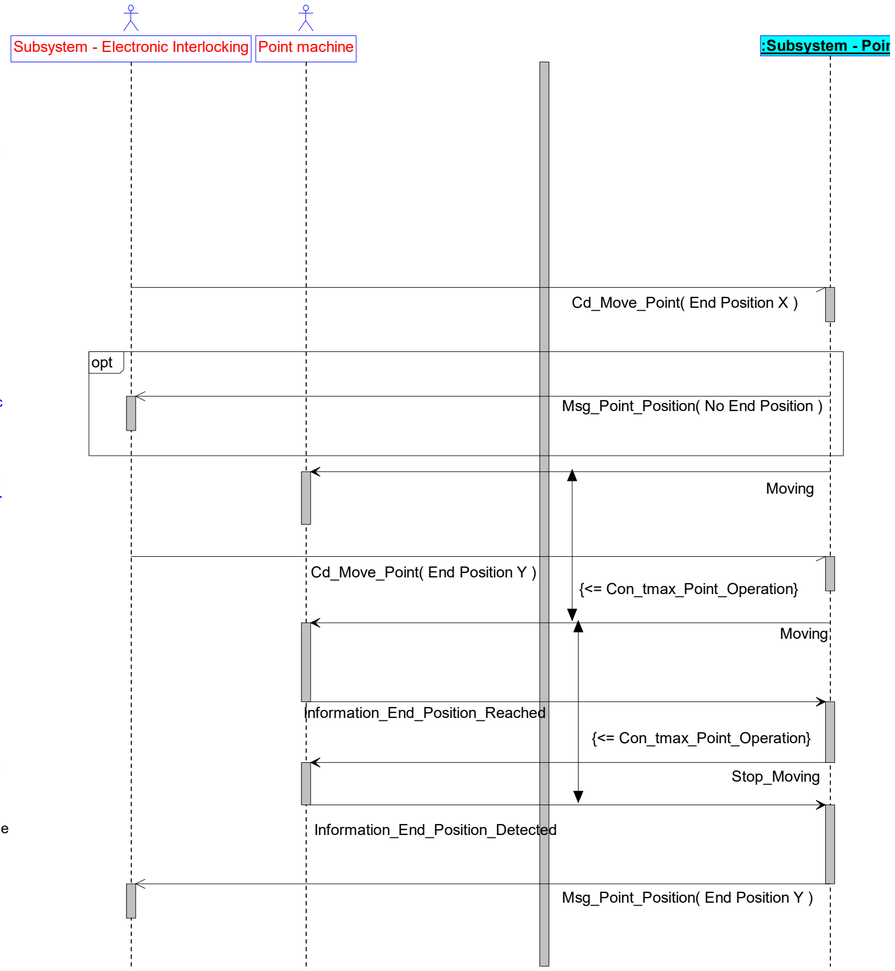
**Postcondition:**

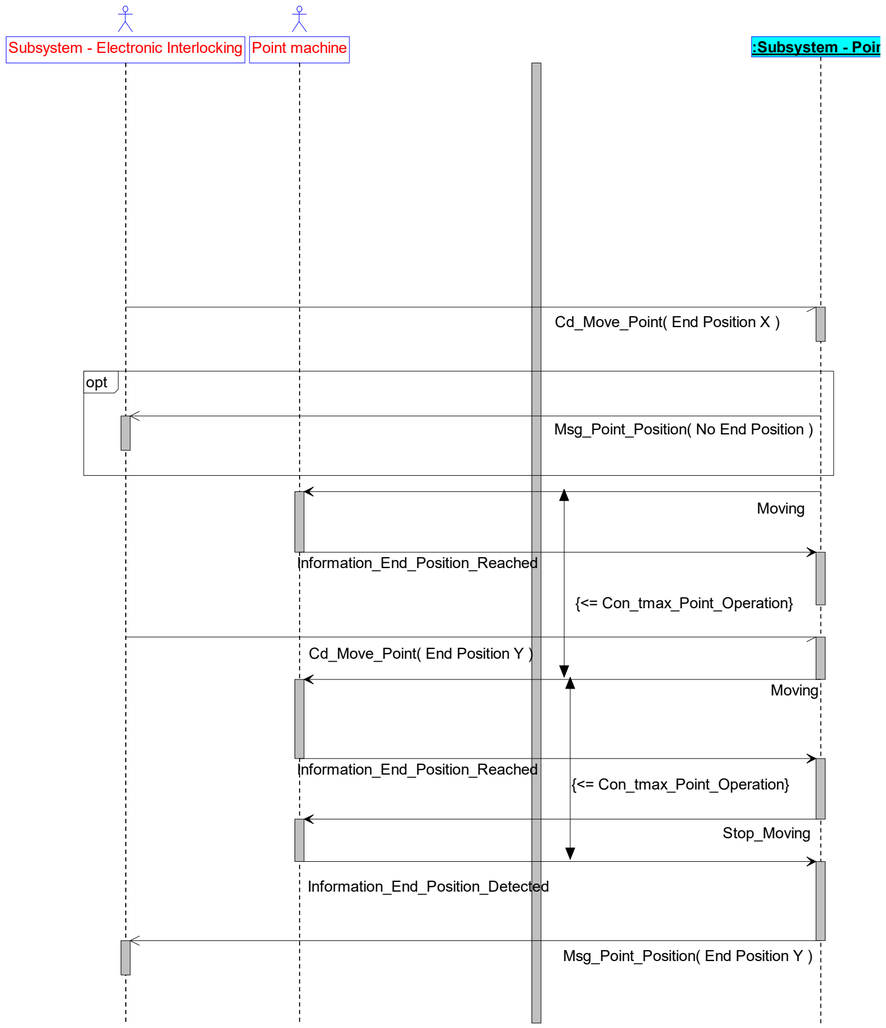
The Subsystem - Point is in No end position.

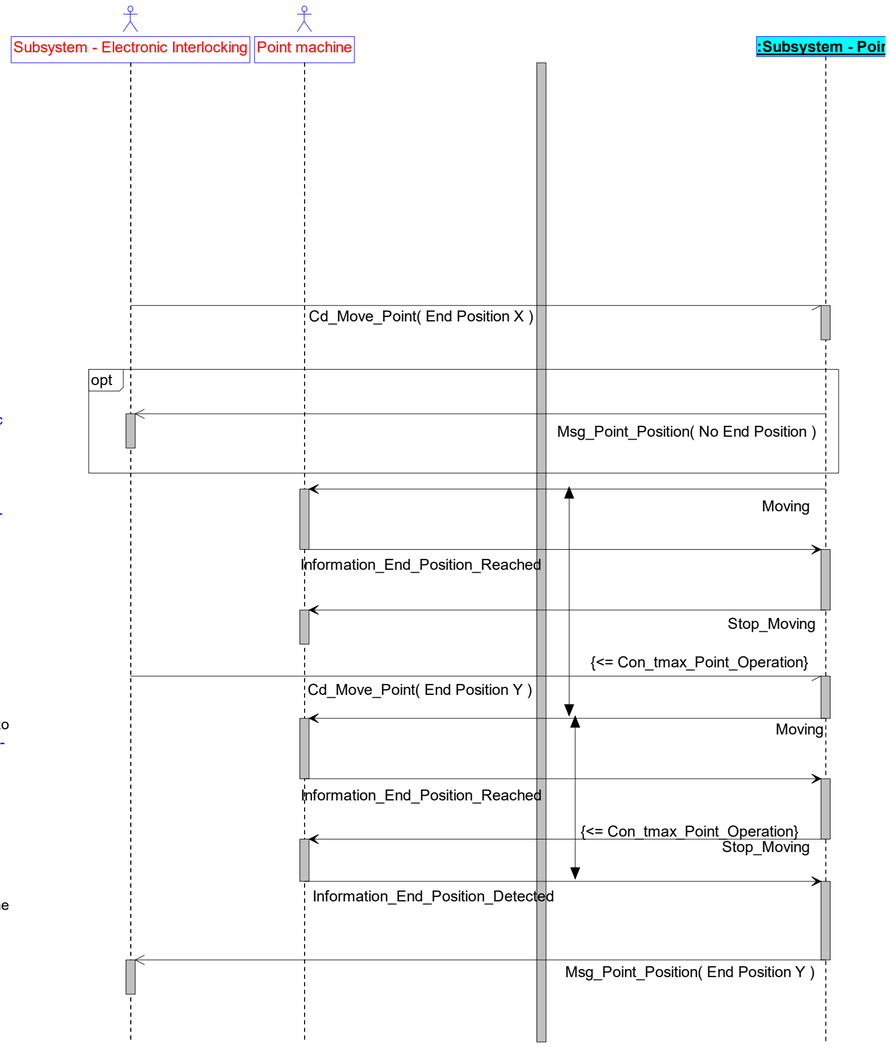


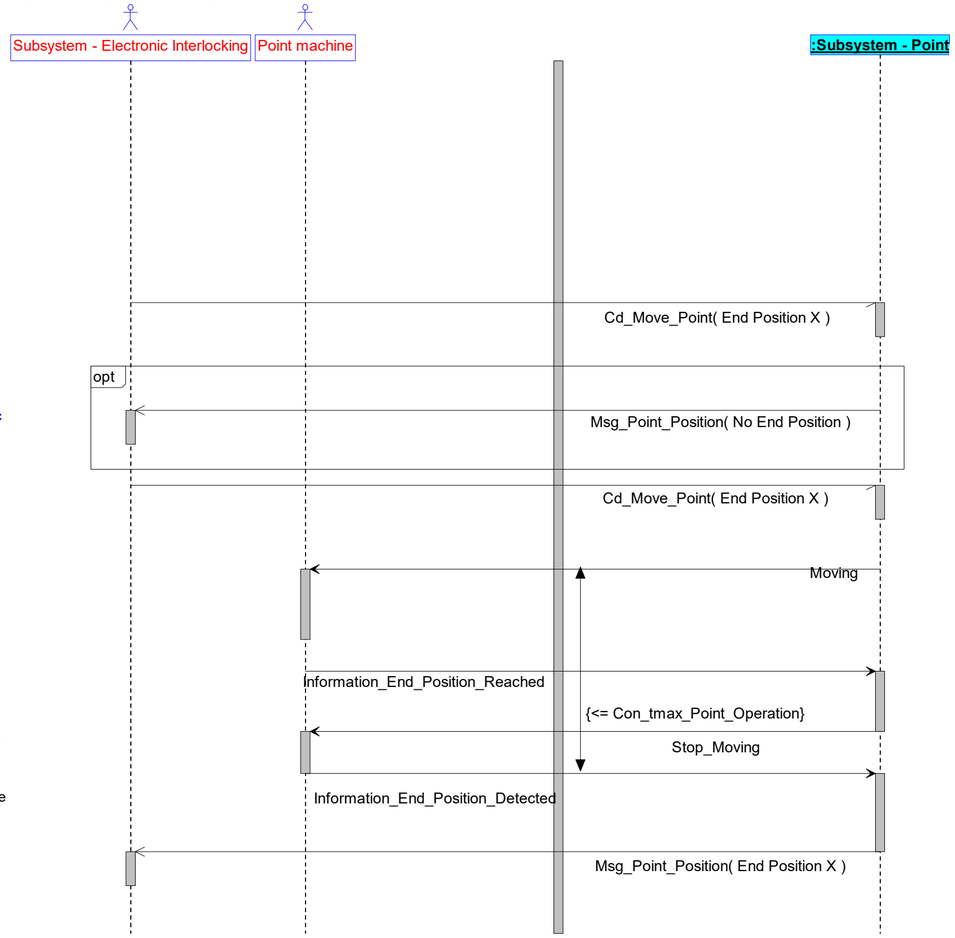
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.7378 | Info | Alternative Scenario: Redrive Point trigger unable to move [P SD 2.1.2.4.17]<br>**P_UC2.1.2.4: Irregularities**<br>Alternative Scenario: Redrive Point trigger unable to move [P SD 2.1.2.4.17]<br>**Precondition:**<br>The Subsystem - Point is in the state INITIALISING or OPERATIONAL.<br>The Subsystem - Point is in an End position "X".<br>The last Cd_Move_Point Command received was also for End position "X".<br>The last Cd_Move_Point Command received occurred since exiting the state BOOTING.<br>The Subsystem - Point is configured with a non-4-wire interface to the Point machine.<br>The Subsystem - Point is configured as Redrive point.<br>The Subsystem - Point is Unable to move point.<br>**Interaction 2.1.2.2.17.A:**<br>**alt** [The 1st Point machine was previously in an End position]<br>**1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point is in No end position.<br>**opt** [The Subsystem - Point is in the state OPERATIONAL.]<br>**1.a2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**end opt**<br>**else alt** [The n-th Point machine was previously in an End position]<br>**1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point is in No end position.<br>**opt** [The Subsystem - Point is in the state OPERATIONAL.]<br>**1.b2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**end opt**<br>**end alt**<br>**Postcondition:**<br>The Subsystem - Point is in No end position.<br>  <br> | Only applicable if the package [Option Redrive] is used in combination with [Basic non-4-wire multiple P]. | Option Redrive |
| Eu.P.6770 | Info | P_UC2.2: Point with 4-wire interface | The Subsystem-UseCase "P_UC2.2: Point with 4-wire interface" defines the behaviour of the Subsystem - Point which works with a 4-wire interface.<br>The behaviour will be defined in the following UseCases:<br>P_UC2.2.1: Single point machine<br>P_UC2.2.2: Multiple point machines | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6752 | Info | P_UC2.2.1: Single point machine | The Subsystem-UseCase "P_UC2.2.1: Single point machine" defines the behaviour of the Subsystem - Point which works with a single point machine via 4-wire interface.<br>The behaviour will be defined in the following UseCases:<br>P_UC2.2.1.1: Commanding and reversing<br>P_UC2.2.1.2: Irregularities | Basic 4-wire single P |
| Eu.P.6733 | Info | P_UC2.2.1.1: Commanding and reversing | The Subsystem-UseCase "P_UC2.2.1.1: Commanding and reversing" defines the behaviour of commanding and reversing a single point machine via 4-wire interface. | Basic 4-wire single P |

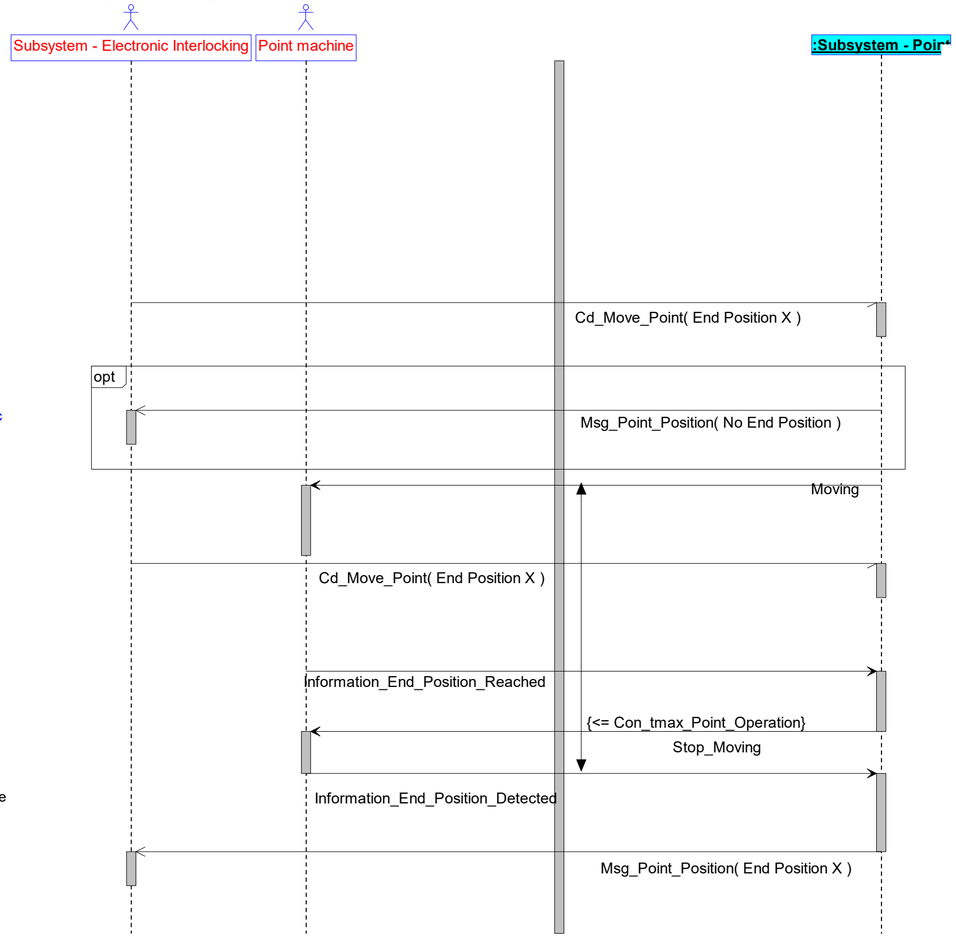
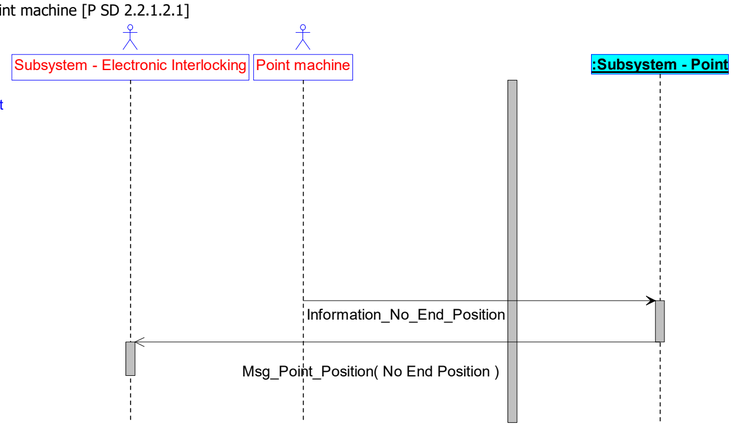
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6740 | Info | Main Success Scenario: Moving of the Point with a single point machine 4W [P SD 2.2.1.1.1]<br>**P_UC2.2.1.1: Commanding and reversing**<br>Main Success Scenario: Moving of the Point with a single point machine 4W [P SD 2.2.1.1.1]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- No end position, or<br>- an Unintended position.<br>**Interaction 2.2.1.1.1.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**Interaction 2.2.1.1.1.B:**<br>**opt** [The Subsystem - Point was previously in an End position or a Unintended position]<br>**2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**end opt**<br>**3.** The Subsystem - Point sends the Command to the Point machine to move the Point machine to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**Interaction 2.2.1.1.1.C:**<br>**4.** - The Subsystem - Point receives from the Point machine the Information that the Point machine has reached End position "Y" as unsafe information.<br>**5.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine.<br>**6.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in a detected End position "X". The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br>**7.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "X".<br>**Postcondition:**<br>The Subsystem - Point is in an End position "X".<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic 4-wire single P |

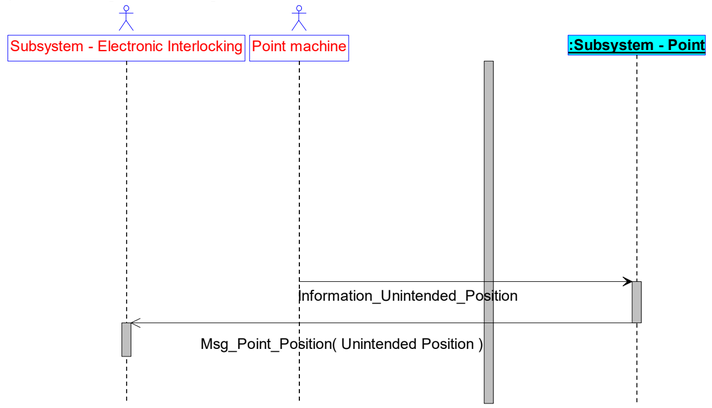
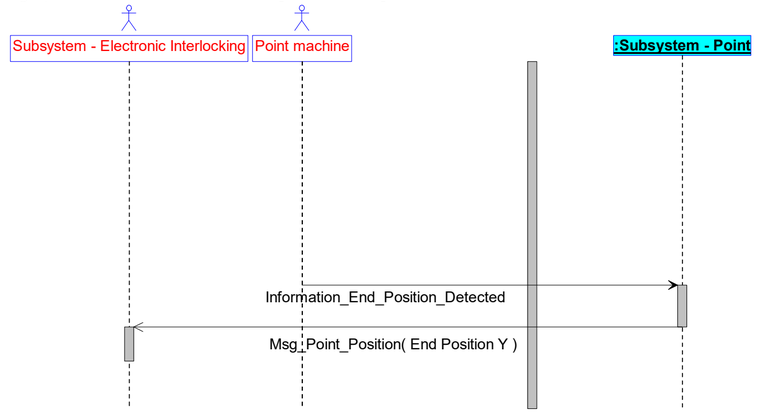
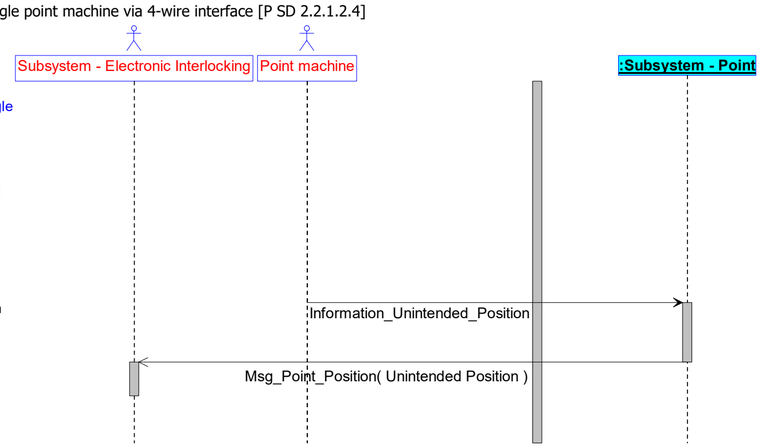
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6736 | Info | Alternative Scenario: Reversing Point 4W [P SD 2.2.1.1.2]<br><br>**P_UC2.2.1.1: Commanding and reversing**<br><br>Alternative Scenario: Reversing Point 4W [P SD 2.2.1.1.2]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- No end position, or<br>- an Unintended position.<br>**Interaction 2.2.1.1.2.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**Interaction 2.2.1.1.2.B:**<br>**opt** [The Subsystem - Point was previously in an End position or a Unintended position]<br>**2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that Point is in No end position.<br>**end opt**<br>**3.** The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**Interaction 2.2.1.1.2.C:**<br>**4.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "Y".<br>**Interaction 2.2.1.1.2.D:**<br>**5.** - The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "Y". At this moment the Subsystem - Point restarts to monitor the time period Con_tmax_Point_Operation.<br>**6.** - The Subsystem - Point receives from the Point machine the Information that the Point machine has reached End position "Y" as unsafe information.<br>**7.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine.<br>**8.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in a detected End position "Y". The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br>**9.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "Y".<br>**Postcondition:**<br>The Subsystem - Point is in an End position "Y".<br><br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic 4-wire single P |

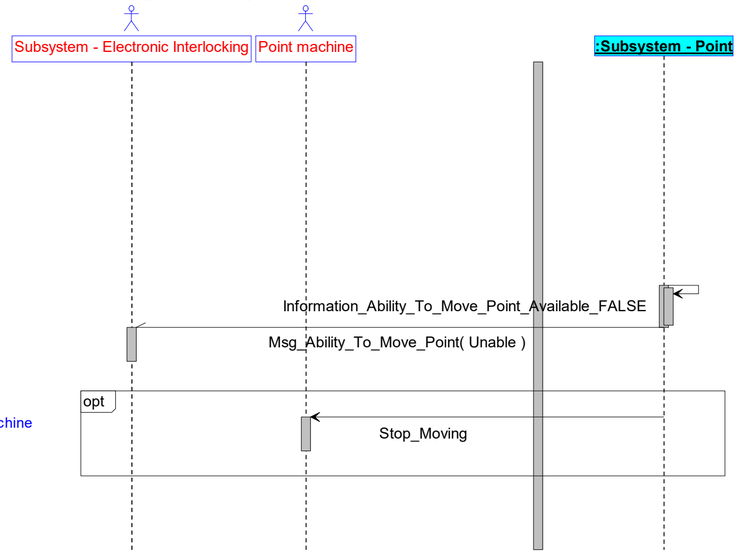
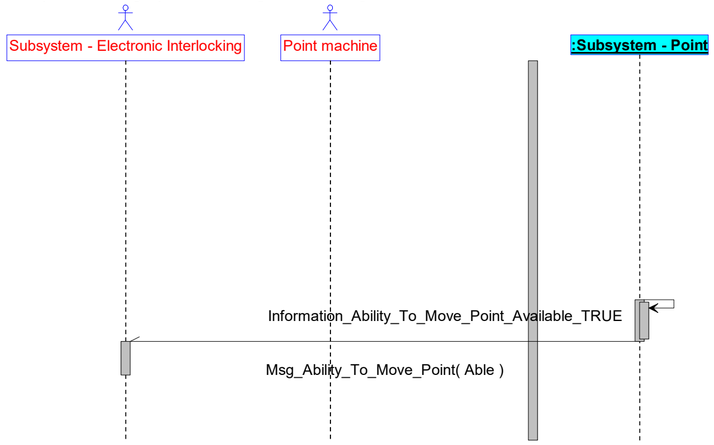
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6737 | Info | Alternative Scenario: Reversing Point 4W after End position reached [P SD 2.2.1.1.3]<br><br>**P_UC2.2.1.1: Commanding and reversing**<br><br>Alternative Scenario: Reversing Point 4W after End position reached [P SD 2.2.1.1.3]<br><br>**Precondition:**<br><br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- No end position, or<br>- an Unintended position.<br><br>**Interaction 2.2.1.1.3.A:**<br><br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br><br>**Interaction 2.2.1.1.3.B:**<br><br>**opt** [The Subsystem - Point was previously in an End position or a Unintended position]<br><br>**2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that Point is in No end position.<br><br>**end opt**<br><br>**3.** The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br><br>**4.** - The Subsystem - Point receives from the Point machine the Information that the Point machine has reached End position "X" as unsafe information.<br><br>**Interaction 2.2.1.1.3.C:**<br><br>**5.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "Y".<br><br>**6.** - The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "Y". At this moment the Subsystem - Point restarts to monitor the time period Con_tmax_Point_Operation.<br><br>**7.** - The Subsystem - Point receives from the Point machine the Information that the Point machine has reached End position "Y" as unsafe information.<br><br>**8.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine.<br><br>**9.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in a detected End position "Y". The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br><br>**10.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "Y".<br><br>**Postcondition:**<br><br>The Subsystem - Point is in an End position "Y".<br><br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic 4-wire single P |

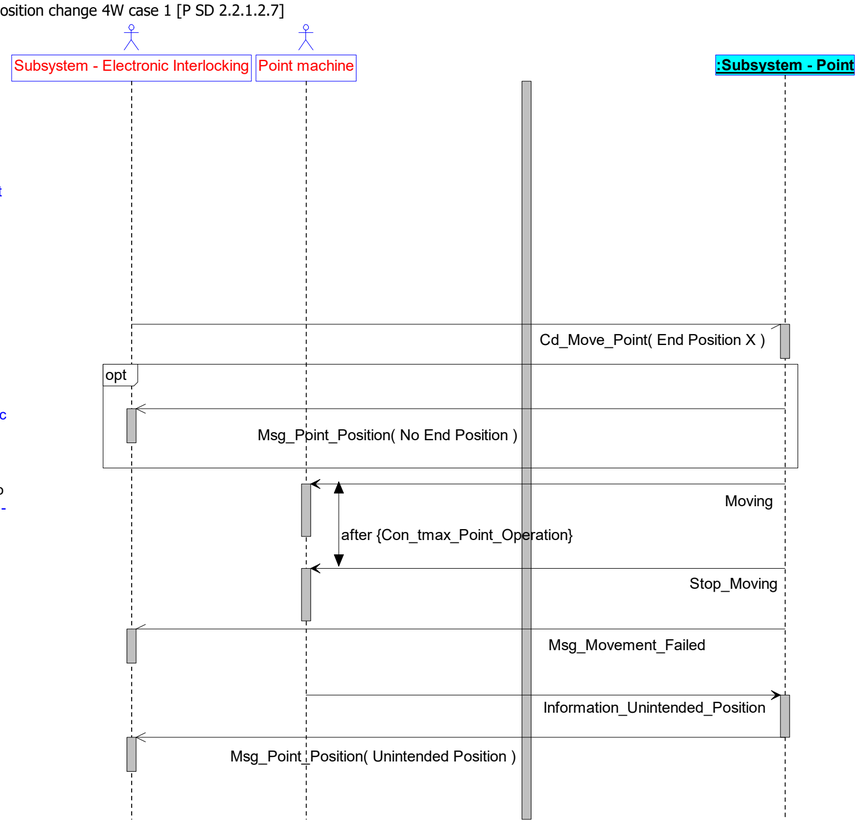
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6738 | Info | Alternative Scenario: Reversing Point 4W before End position is detected[P SD 2.2.1.1.4]<br><br>**P_UC2.2.1.1: Commanding and reversing**<br><br>Alternative Scenario: Reversing Point 4W before End position is detected[P SD 2.2.1.1.4]<br><br>**Precondition:**<br><br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- No end position, or<br>- an Unintended position.<br><br>**Interaction 2.2.1.1.4.A:**<br><br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br><br>**Interaction 2.2.1.1.4.B:**<br><br>**opt** [The Subsystem - Point was previously in an End position or a Unintended position]<br><br>**2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that Point is in No end position.<br><br>**end opt**<br><br>**3.** The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br><br>**4.** - The Subsystem - Point receives from the Point machine the Information that the Point machine has reached End position "X" as unsafe information.<br><br>**5.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine.<br><br>**Interaction 2.2.1.1.4.C:**<br><br>**6.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "Y".<br><br>**7.** - The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "Y". At this moment the Subsystem - Point restarts to monitor the time period Con_tmax_Point_Operation.<br><br>**8.** - The Subsystem - Point receives from the Point machine the Information that the Point machine has reached End position "Y" as unsafe information.<br><br>**9.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine.<br><br>**10.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in a detected End position "Y". The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br><br>**11.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "Y".<br><br>**Postcondition:**<br><br>The Subsystem - Point is in an End position "Y".<br><br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic 4-wire single P |

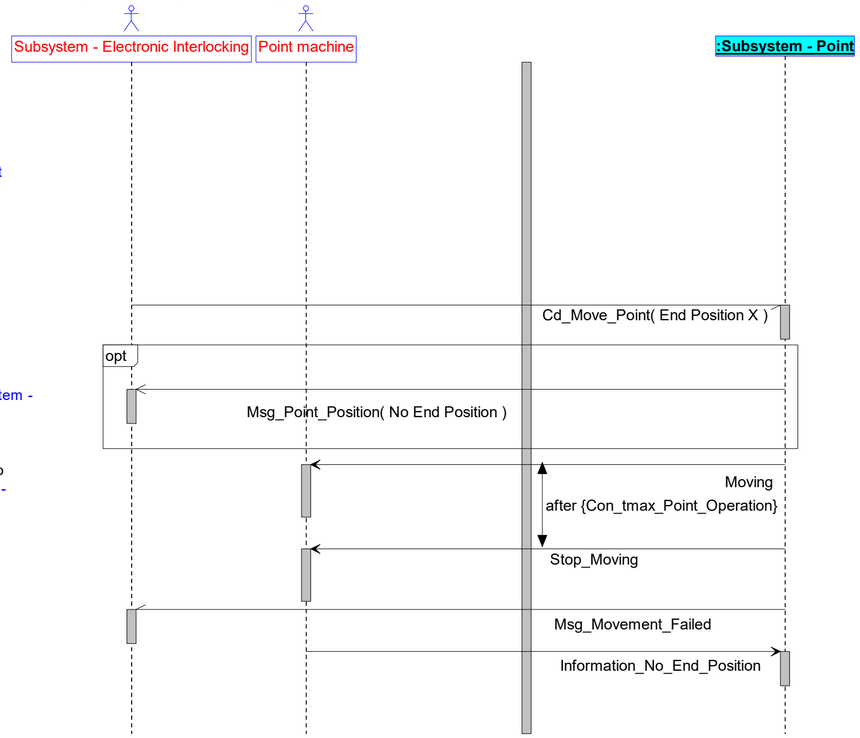
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6734 | Info | Alternative Scenario: Moving of the Point with repeated command of moving 4W case 1[P SD 2.2.1.1.5]<br>**P_UC2.2.1.1: Commanding and reversing**<br>Alternative Scenario: Moving of the Point with repeated command of moving 4W case 1[P SD 2.2.1.1.5]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- No end position, or<br>- an Unintended position.<br>**Interaction 2.2.1.1.5.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**Interaction 2.2.1.1.5.B:**<br>**opt** [The Subsystem - Point was previously in an End position or a Unintended position]<br>**2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**end opt**<br>**3.** The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**4.** The Subsystem - Point ignores the command from the Subsystem - Electronic Interlocking.<br>**5.** The Subsystem - Point sends the Command to the Point machine to move the Point machine to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**Interaction 2.2.1.1.5.C:**<br>**6.** - The Subsystem - Point receives from the Point machine the Information that the Point machine has reached End position "X" as unsafe information.<br>**7.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine.<br>**8.** The Subsystem - Point receives from the Point machine the Information that the Point machine is in a detected End position "X". The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br>**9.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "X".<br>**Postcondition:**<br>The Subsystem - Point is in an End position "X". |  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic 4-wire single P |

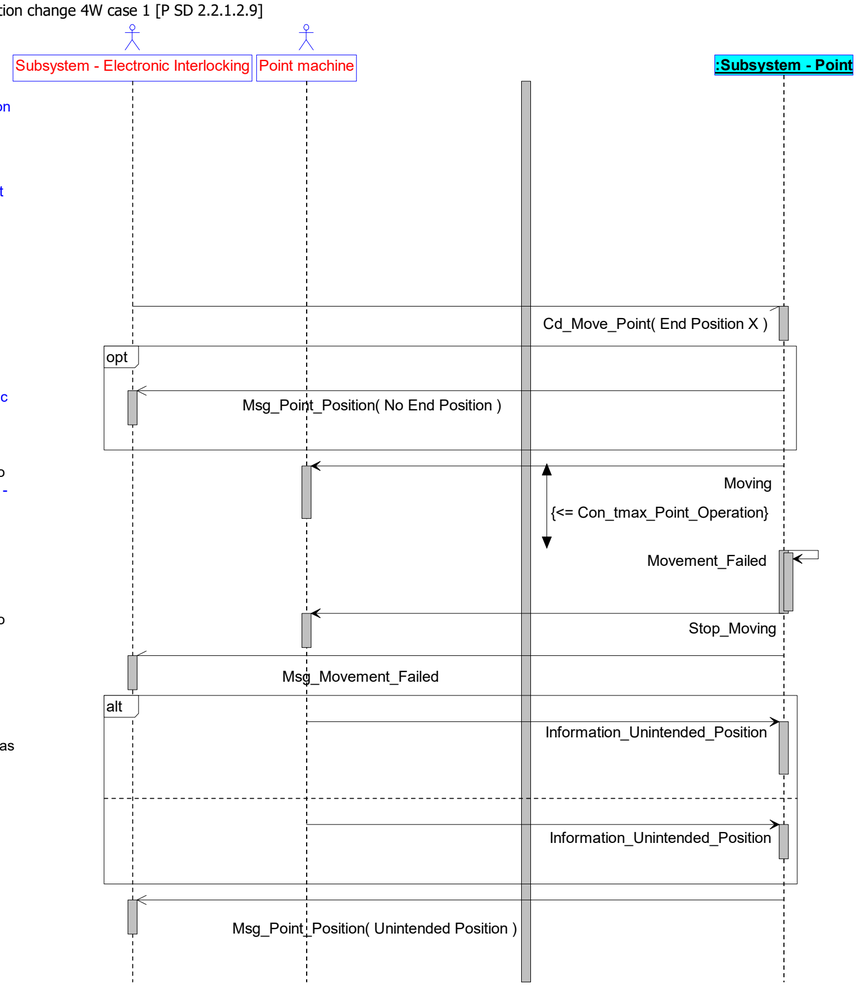
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6735 | Info | Alternative Scenario: Moving of the Point with repeated command of moving 4W case 2[P SD 2.2.1.1.6]<br>**P_UC2.2.1.1: Commanding and reversing**<br>Alternative Scenario: Moving of the Point with repeated command of moving 4W case 2[P SD 2.2.1.1.6]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- No end position, or<br>- an Unintended position.<br>**Interaction 2.2.1.1.6.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**Interaction 2.2.1.1.6.B:**<br>**opt** [The Subsystem - Point was previously in an End position or a Unintended position]<br>**2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**end opt**<br>**3.** The Subsystem - Point sends the Command to the Point machine to move the Point machine to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**4.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**5.** The Subsystem - Point ignores the command from the Subsystem - Electronic Interlocking.<br>**Interaction 2.2.1.1.6.C:**<br>**6.** - The Subsystem - Point receives from the Point machine the Information that the Point machine has reached End position "Y" as unsafe information.<br>**7.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine.<br>**8.** The Subsystem - Point receives from the Point machine the Information that the Point machine is in a detected End position "X". The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br>**9.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "X".<br>**Postcondition:**<br>The Subsystem - Point is in an End position "X".<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic 4-wire single P |
| Eu.P.6741 | Info | P_UC2.2.1.2: Irregularities | The Subsystem-UseCase "P_UC2.2.1.2: Irregularities" defines the behaviour of the Subsystem - Point which works with a single point machine via 4-wire interface, when an irregularity occurs. | Basic 4-wire single P |
| Eu.P.6746 | Info | Alternative Scenario: Handle and report No end position with single point machine [P SD 2.2.1.2.1]<br>**P_UC2.2.1.2: Irregularities**<br>Alternative Scenario: Handle and report No end position with single point machine [P SD 2.2.1.2.1]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- a Unintended position<br>**Interaction 2.2.1.2.1.A:**<br>**1.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in No end position.<br>**2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**Postcondition:**<br>The Subsystem - Point is in No end position.<br>  <br> | | Basic 4-wire single P |

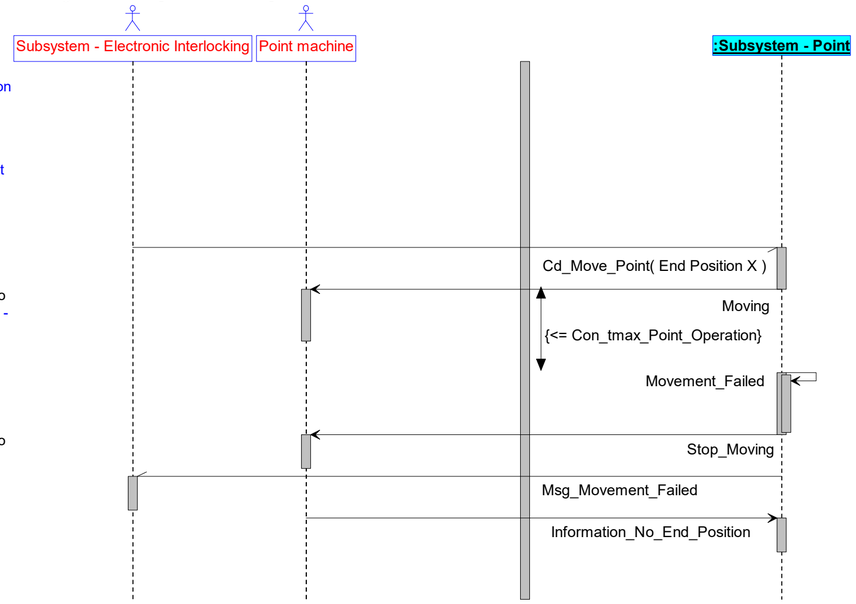
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6751 | Info | Alternative Scenario: Handle and report Unintended position with single point machine [P SD 2.2.1.2.2]<br>**P_UC2.2.1.2: Irregularities**<br>Alternative Scenario: Handle and report Unintended position with single point machine [P SD 2.2.1.2.2]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- No end position.<br>**Interaction 2.2.1.2.2.A:**<br>**1.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in a Unintended position.<br>**2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in a Unintended position.<br>**Postcondition:**<br>The Subsystem - Point is in a Unintended position.<br>  <br> | | Basic 4-wire single P |
| Eu.P.6742 | Info | Alternative Scenario: Handle and report commanded End Position with single point machine via 4-wire interface [P SD 2.2.1.2.3]<br>**P_UC2.2.1.2: Irregularities**<br>Alternative Scenario: Handle and report commanded End Position with single point machine via 4-wire interface [P SD 2.2.1.2.3]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is End position "Y".<br>The Subsystem - Point is in:<br>- No end position, or<br>- a Unintended position.<br>**Interaction 2.2.1.2.3.A:**<br>**1.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in an End position "Y".<br>**2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "Y".<br>**Postcondition:**<br>The Subsystem - Point is in an End position "Y".<br>  <br> | | Basic 4-wire single P |
| Eu.P.6747 | Info | Alternative Scenario: Handle and report opposite End Position with single point machine via 4-wire interface [P SD 2.2.1.2.4]<br>**P_UC2.2.1.2: Irregularities**<br>Alternative Scenario: Handle and report opposite End Position with single point machine via 4-wire interface [P SD 2.2.1.2.4]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is End position "Y".<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- No end position.<br>**Interaction 2.2.1.2.4.A:**<br>**1.** - The Subsystem - Point receives from the Point machine the pattern that represents the opposite End position "X", which is interpreted as Information that the Point machine is in an Unintended position.<br>**2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in a Unintended position.<br>**Postcondition:**<br>The Subsystem - Point is in a Unintended position.<br>  <br> | | Basic 4-wire single P |

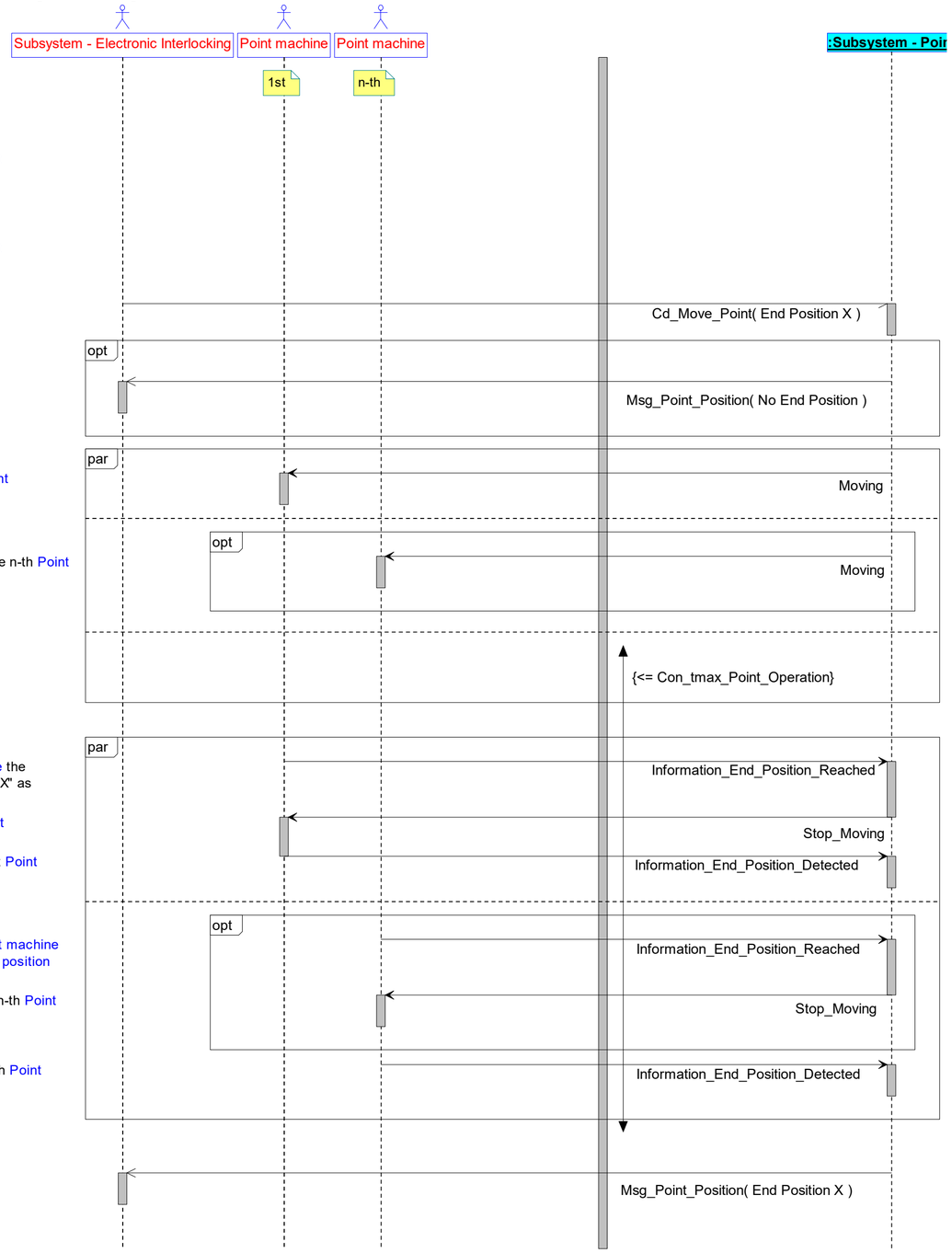
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6745 | Info | Alternative Scenario: Handle and report loss of ability to move point with single point machine [P SD 2.2.1.2.5]<br>**P_UC2.2.1.2: Irregularities**<br>Alternative Scenario: Handle and report loss of ability to move point with single point machine [P SD 2.2.1.2.5]<br>**Precondition**:<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine.<br>The Subsystem - Point is configured to observe the Ability to move point.<br>The Subsystem - Point is in:<br>- Able to move point<br>**Interaction 2.2.1.2.5.A**:<br>**1** - The Subsystem - Point internal trigger indicates that the Subsystem - Point is Unable to move point<br>**2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that it is Unable to move point.<br>**Interaction 2.2.1.2.5.B**:<br>**opt** [The point machine is in Moving]<br>**3.a1** - The Subsystem - Point sends the Command to the Point machine to Stop moving the Point machine.<br>**end opt**<br>**Postcondition:**<br>The Subsystem - Point is in:<br>- Unable to move point<br>  <br> | Only applicable if the package [Option Able to move] is used in combination with [Basic 4-wire single P]. | Option Able to move |
| Eu.P.6750 | Info | Alternative Scenario: Handle and report restoring of Ability to move point with single point machine [P SD 2.2.1.2.6]<br>**P_UC2.2.1.2: Irregularities**<br>Alternative Scenario: Handle and report restoring of Ability to move point with single point machine [P SD 2.2.1.2.6]<br>**Precondition**:<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine.<br>The Subsystem - Point is configured to observe the Ability to move point.<br>The Subsystem - Point is in:<br>- Unable to move point<br>**Interaction 2.2.1.2.6.A**:<br>**1.** - The Subsystem - Point internal trigger indicates that the Subsystem - Point is Able to move point.<br>**2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that it is Able to move point.<br>**Postcondition:**<br>The Subsystem - Point is in:<br>- Able to move point<br>  <br> | Only applicable if the package [Option Able to move] is used in combination with [Basic 4-wire single P]. | Option Able to move |

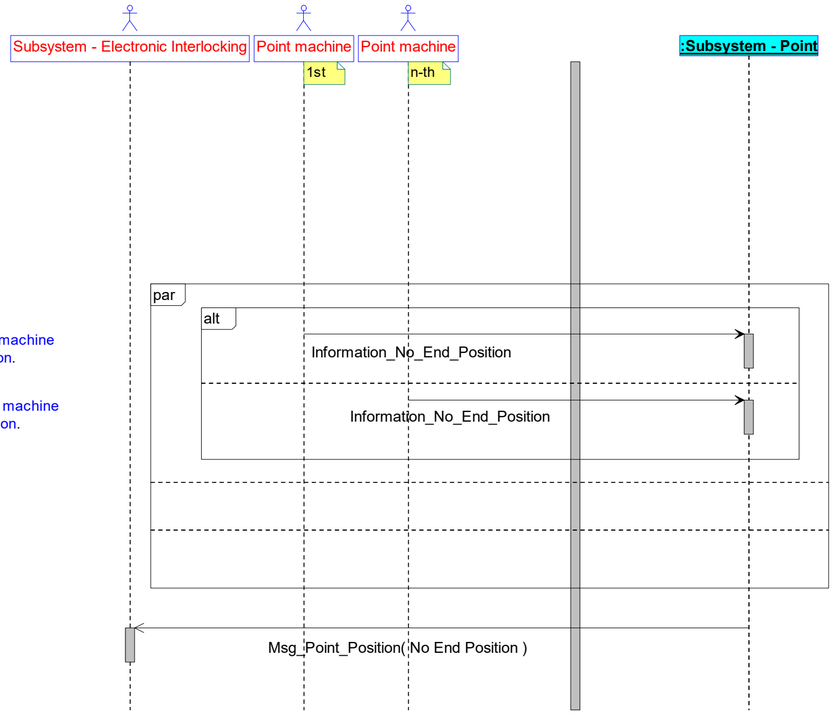
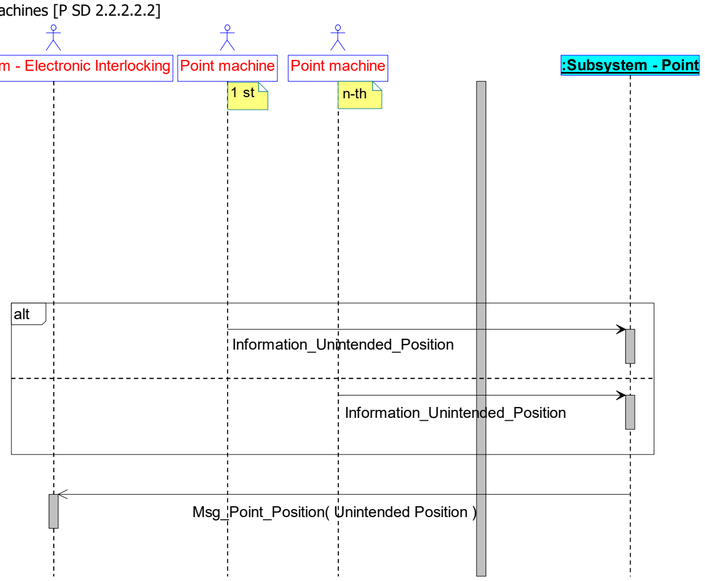
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6748 | Info | Alternative Scenario: Handle and report Point operation timeout with position change 4W case 1 [P SD 2.2.1.2.7]<br>**P_UC2.2.1.2: Irregularities**<br>Alternative Scenario: Handle and report Point operation timeout with position change 4W case 1 [P SD 2.2.1.2.7]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- No end position, or<br>- an Unintended position.<br>**Interaction 2.2.1.2.7.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**opt** [The Subsystem - Point was previously in an End position or a Unintended position]<br>**2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>end opt<br>**3.** The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**Interaction 2.2.1.2.7.B:**<br>**4.** The Subsystem - Point detects that the time period Con_tmax_Point_Operation has exceeded and then sends the Command to the Point machine to stop moving the Point machine.<br>**5.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred.<br>**Interaction 2.2.1.2.7.C:**<br>**6.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in a Unintended position.<br>**7.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in Unintended position.<br>**Postcondition:**<br>The Subsystem - Point is in a Unintended position.<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic 4-wire single P |

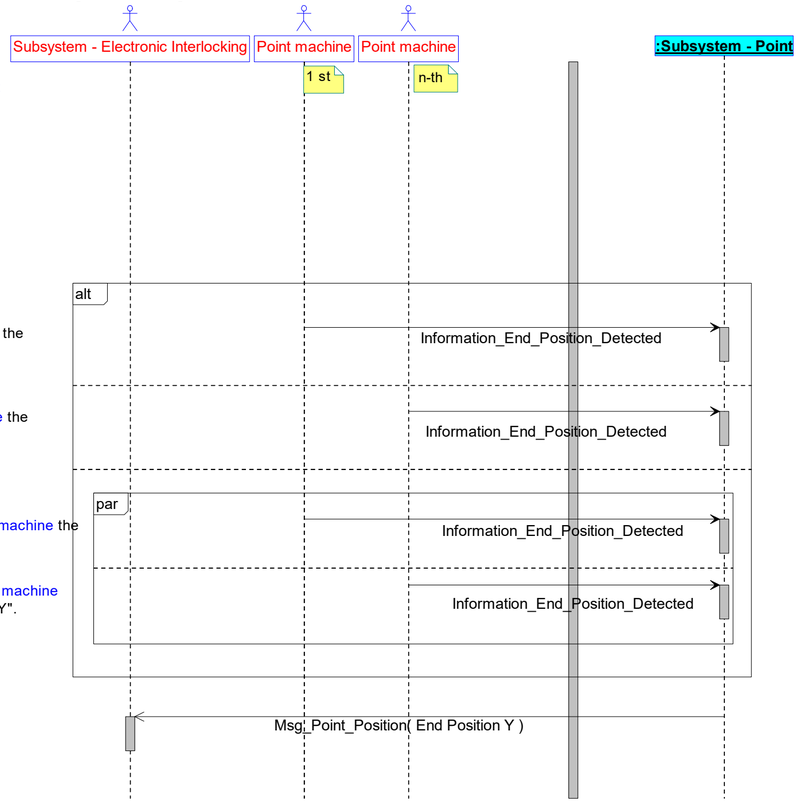
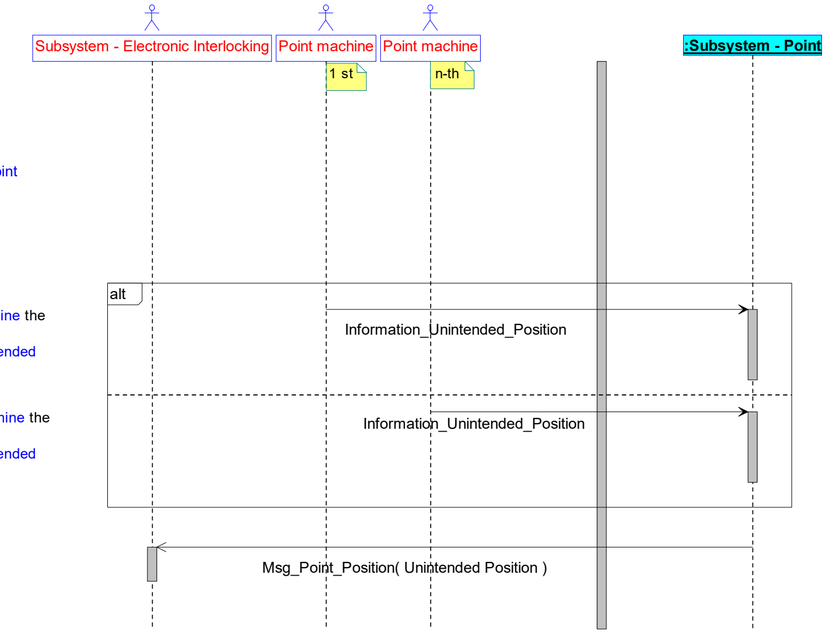
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6749 | Info | Alternative Scenario: Handle and report Point operation timeout with position change 4W case 2 [P SD 2.2.1.2.8]<br>**P_UC2.2.1.2: Irregularities**<br>Alternative Scenario: Handle and report Point operation timeout with position change 4W case 2 [P SD 2.2.1.2.8]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- No end position, or<br>- an Unintended position.<br>**Interaction 2.2.1.2.8.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**opt** [The Subsystem - Point was previously in an End position or a Unintended position]<br>**2.a1** The Subsystem - Point sends the Information to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**end opt**<br>**3.** The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**Interaction 2.2.1.2.8.B:**<br>**4.** The Subsystem - Point detects that the time period Con_tmax_Point_Operation has exceeded and then sends the Command to the Point machine to stop moving the Point machine.<br>**5.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred.<br>**6.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in No end position.<br>**Postcondition:**<br>The Subsystem - Point is in a No end position.<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic 4-wire single P |

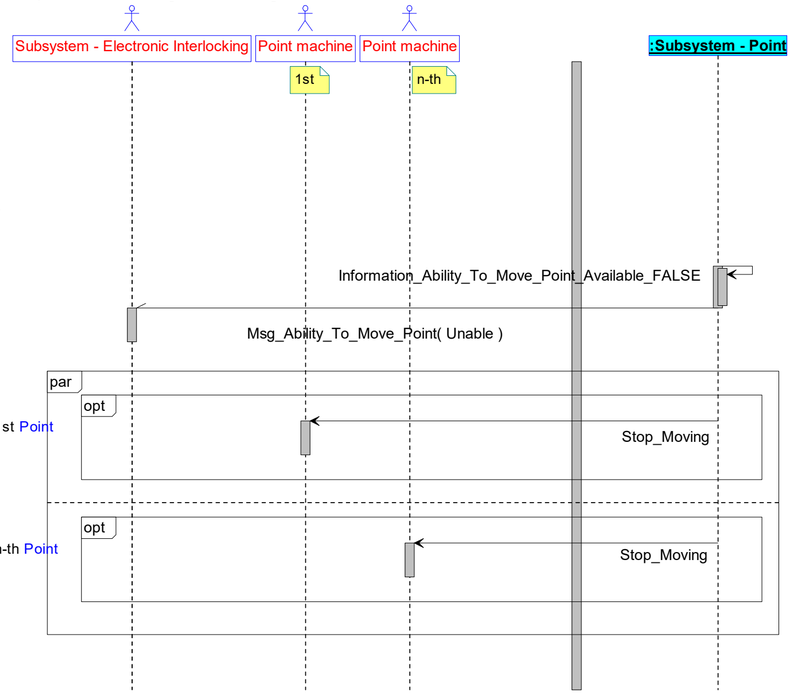
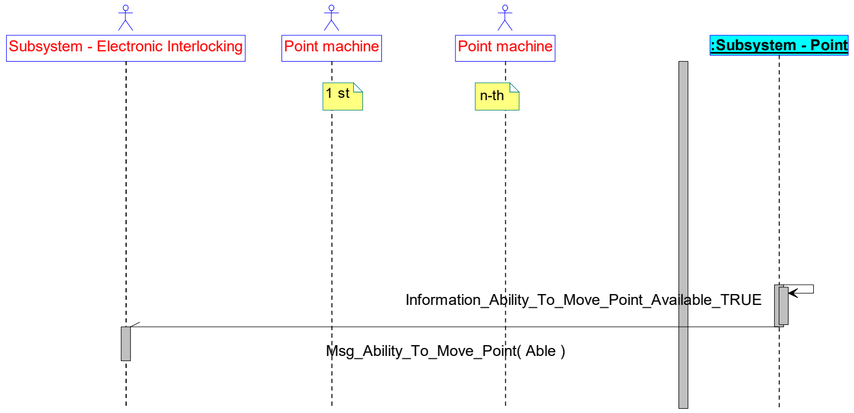
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6743 | Info | Alternative Scenario: Handle and report failed movement without position change 4W case 1 [P SD 2.2.1.2.9]<br>**P_UC2.2.1.2: Irregularities**<br>Alternative Scenario: Handle and report failed movement without position change 4W case 1 [P SD 2.2.1.2.9]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y"or<br>- a Unintended position.<br>**Interaction 2.2.1.2.9.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**opt** [The Subsystem - Point was previously in an End position or a Unintended position]<br>**2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking indicating that the Point is in No end position.<br>**end opt**<br>**3.** The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**Interaction 2.2.1.2.9.B:**<br>**4.** - A failure occurred during the movement resulting in a Failed Movement. The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br>**5.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point.<br>**6.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred.<br>**alt** [The Subsystem - Point was previously in an End position]<br>**7.a1** The Subsystem - Point receives from the Point machine the pattern that represents the End position "Y", which is interpreted as Information that the Point machine is in an Unintended position.<br>**else alt** [The Subsystem - Point was previously in an Unintended position]<br>**7.b1** The Subsystem - Point receives from the Point machine the Information that the Point machine is in a Unintended position.<br>**end alt**<br>**8.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in Unintended position.<br>**Postcondition:**<br>The Subsystem - Point is in a Unintended position.<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic 4-wire single P |

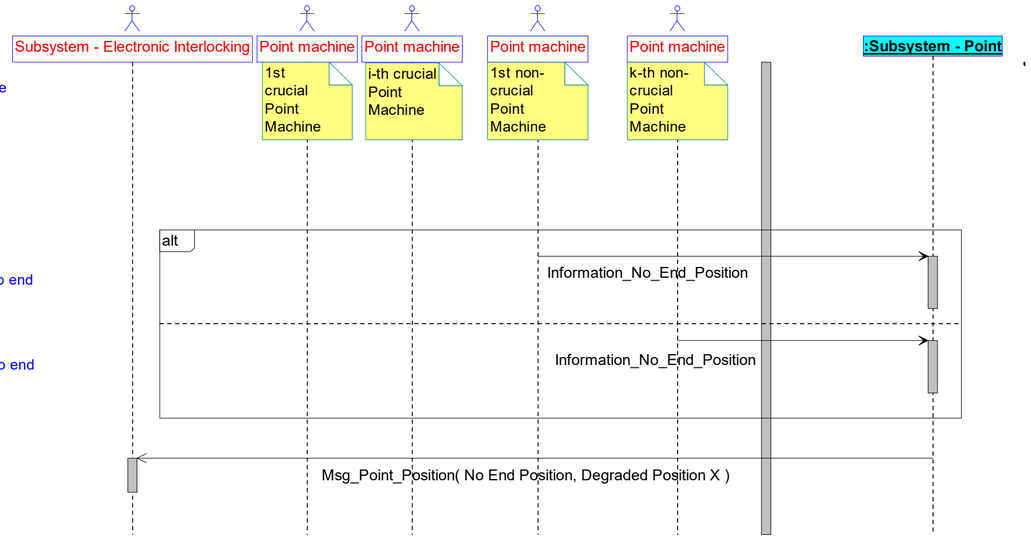
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6744 | Info | Alternative Scenario: Handle and report failed movement without position change 4W case 2 [P SD 2.2.1.2.10]<br>**P_UC2.2.1.2: Irregularities**<br>Alternative Scenario: Handle and report failed movement without position change 4W case 2 [P SD 2.2.1.2.10]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine.<br>The Subsystem - Point is in No end position,<br>**Interaction 2.2.1.2.10.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**2.** The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**Interaction 2.2.1.2.10.B:**<br>**3.** - A failure occurred during the movement resulting in a Failed Movement. The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br>**4.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point.<br>**5.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred.<br>**6.** The Subsystem - Point receives from the Point machine the Information that the Point machine is in No end position.<br>**Postcondition:**<br>The Subsystem - Point is in a No end position.<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic 4-wire single P |
| Eu.P.6769 | Info | P_UC2.2.2: Multiple point machines | The Subsystem-UseCase "P_UC2.2.2: Multiple point machines" defines the behaviour of the Subsystem - Point which works with multiple point machines via 4-wire interface.<br>The behaviour will be defined in the following UseCases:<br>P_UC2.2.2.1: Commanding and reversing<br>P_UC2.2.2.2: Irregularities | Basic 4-wire multiple P |
| Eu.P.6753 | Info | P_UC2.2.2.1: Commanding and reversing | The Subsystem-UseCase "P_UC2.2.2.1: Commanding and reversing" defines the behaviour of commanding and reversing a multiple point machine via 4-wire interface. | Basic 4-wire multiple P |

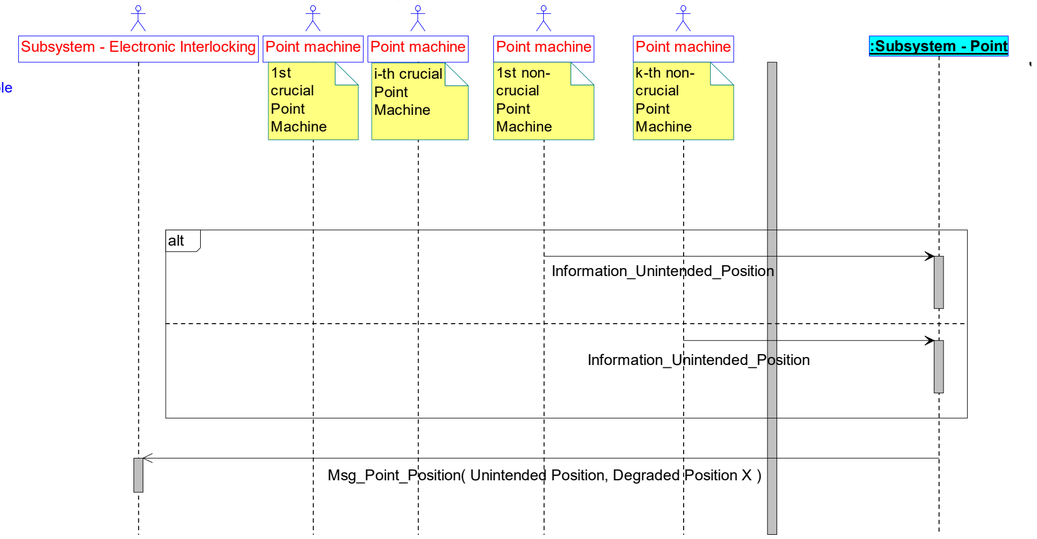
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6754 | Info | Main Success Scenario: Moving of the Point with multiple Point machines 4W [P SD 2.2.2.1.1]<br>**P_UC2.2.2.1: Commanding and reversing**<br>Main Success Scenario: Moving of the Point with multiple Point machines 4W [P SD 2.2.2.1.1]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y",<br>- No end position, or a Unintended position.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine<br>**Interaction 2.2.2.1.1.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**opt** [The Subsystem - Point was previously in an End position or a Unintended position ]<br>**2.a1** The Subsystem - Point reports the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**end opt**<br>**par**<br>**3.a1** - The Subsystem - Point sends the Command to the 1st Point machine to move the Point to an End position "X".<br>**also par**<br>**opt** [The n-th Point machine has drive capability]<br>**3.b1.a1** - The Subsystem - Point sends the Command to the n-th Point machine to move the Point to an End position "X".<br>**end opt**<br>**also par**<br>**3.c1** - The Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**end par**<br>**Interaction 2.2.2.1.1.B:**<br>**par**<br>**4.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the 1st Point machine has reached End position "X" as unsafe information.<br>**4.a2** The Subsystem - Point sends the Command to the 1st Point machine to stop moving the Point.<br>**4.a3** - The Subsystem - Point receives the Information that the 1st Point machine is in an detected End position "X".<br>**also par**<br>**opt** [The n-th Point machine has drive capability]<br>**4.b1.a1** - The Subsystem - Point receives from the n-th Point machine the Information that the n-th Point machine has reached End position "X" as unsafe information.<br>**4.b1.a2** The Subsystem - Point sends the Command to the n-th Point machine to stop moving the Point.<br>**end opt**<br>**4.b2** - The Subsystem - Point receives the Information that the n-th Point machine is in an detected End position "X".<br>**end par**<br>**5.** The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br>**6.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "X".<br>**Postcondition:**<br>The Subsystem - Point is in an End position "X".<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.<br>The timing related to driving individual (1 to n-th) Point Machines for a single Point is supplier specific and is specified only in a general way. The timing related to individual Point Machines is not specified by EULYNX as part of the application layer, this shall be handled by the physical implementation. | Basic 4-wire multiple P |
| Eu.P.6755 | Info | P_UC2.2.2.2: Irregularities | The Subsystem-UseCase "P_UC2.2.2.2: Irregularities" defines the behaviour of the Subsystem - Point which works with a multiple point machine via 4-wire interface, when an irregularity occurs. | Basic 4-wire multiple P |

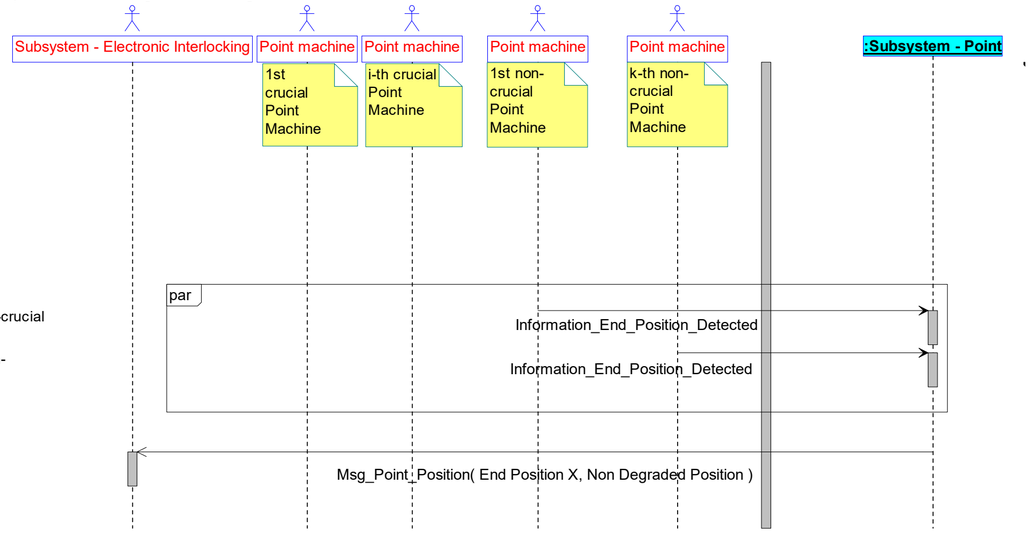
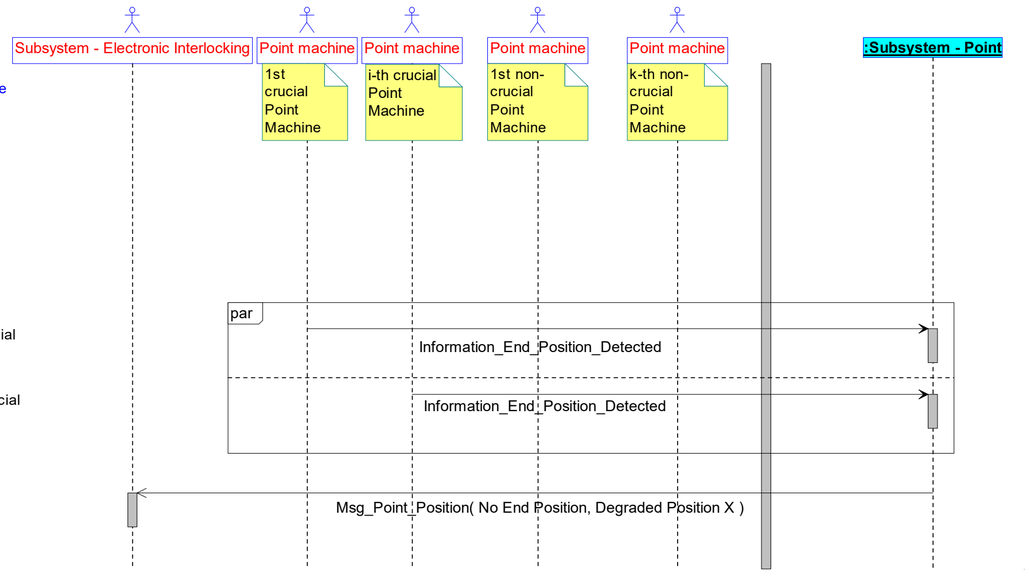
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| | | | Note: For implementations of the Subsystem - Point with more than 2 Point Machine interfaces, there may be more combinations of inputs possible than shown in the SDs in this section. | |
| Eu.P.6762 | Info | Alternative Scenario: Handle and report No end position with multiple Point machines [P SD 2.2.2.2.1]<br>**P_UC2.2.2.2: Irregularities**<br>Alternative Scenario: Handle and report No end position with multiple Point machines [P SD 2.2.2.2.1]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is End position "Y".<br>The Subsystem - Point is in:<br>- an End position "Y",<br>or- a Unintended position.<br>**Interaction 2.2.2.2.1.A:**<br>**par**<br>**alt** [The 1st Point machine is in No end position]<br>**1.a1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the 1st Point machine is in No end position.<br>**else alt** [The n-th Point machine is in No end position]<br>**1.a1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the n-th Point machine is in No end position.<br>**end alt**<br>**also par**<br>**1.b1** - The 1st Point machine is NOT in a Unintended position.<br>**also par**<br>**1.c1** - The n-th Point machine is NOT in a Unintended position.<br>**end par**<br>**Interaction 2.2.2.2.1.B:**<br>**2.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**Postcondition:**<br>The Subsystem - Point is in No end position.<br>  <br> | | Basic 4-wire multiple P |
| Eu.P.6768 | Info | Alternative Scenario: Handle and report Unintended position with multiple Point machines [P SD 2.2.2.2.2]<br>**P_UC2.2.2.2: Irregularities**<br>Alternative Scenario: Handle and report Unintended position with multiple Point machines [P SD 2.2.2.2.2]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- No end position.<br>**Interaction 2.2.2.2.2.A:**<br>**alt** [The 1st Point machine is in an Unintended position.]<br>**1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point machine is in a Unintended position.<br>**else alt** [The n-th Point machine is in a Unintended position.]<br>**1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point machine is in a Unintended position.<br>**end alt**<br>**Interaction 2.2.2.2.2.B:**<br>**2.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking indicating that the Point is in a Unintended position.<br>**Postcondition:**<br>The Subsystem - Point is in a Unintended position.<br>  <br> | | Basic 4-wire multiple P |

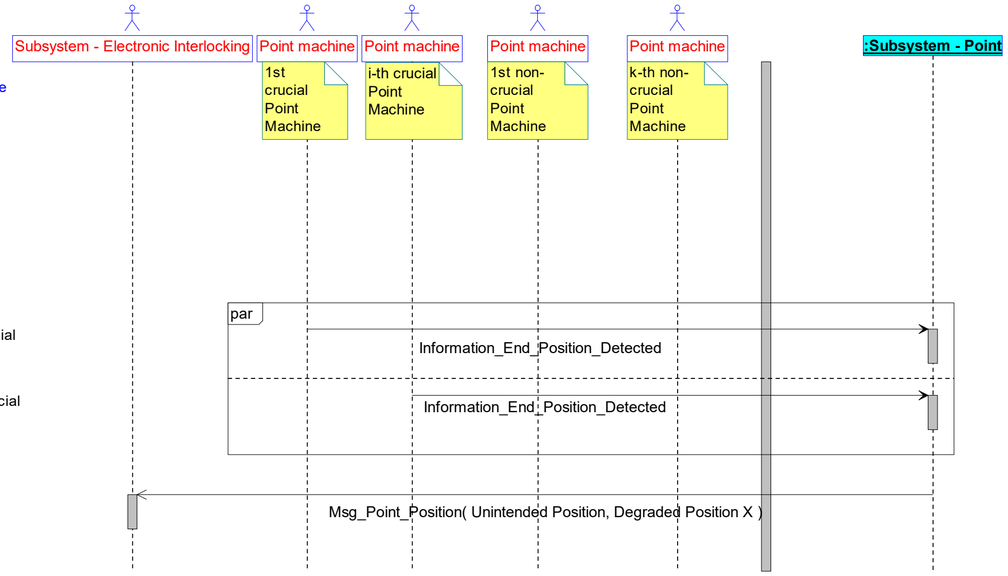
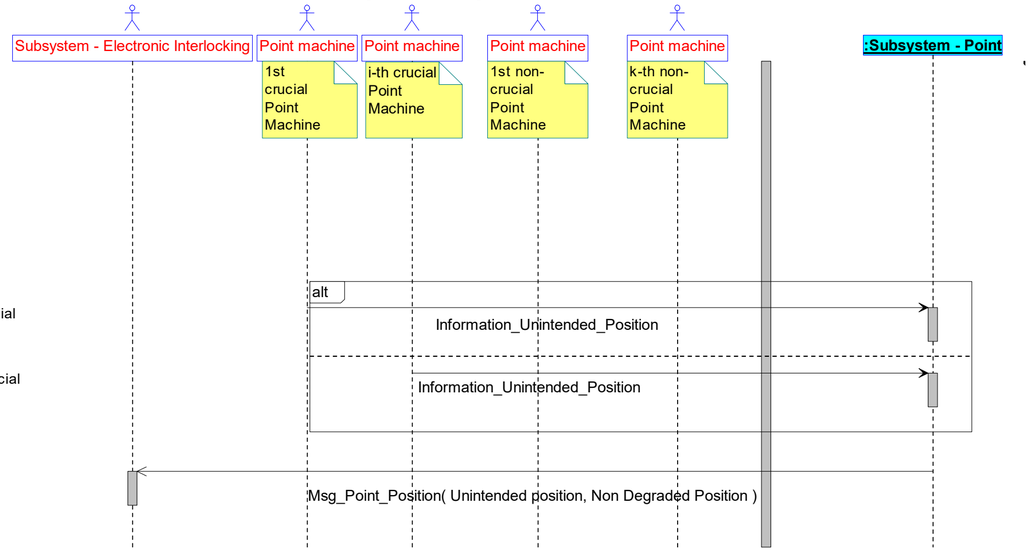
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6758 | Info | Alternative Scenario: Handle and report End Position with multiple Point machines [P SD 2.2.2.2.3]<br>**P_UC2.2.2.2: Irregularities**<br>Alternative Scenario: Handle and report End Position with multiple Point machines [P SD 2.2.2.2.3]<br>**Precondition:**<br>he Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is End position "Y".<br>The Subsystem - Point is in:<br>- an No end position, or<br>- a Unintended position.<br>**Interaction 2.2.2.2.3.A:**<br>**alt** [The 1st Point machine is not in an End position "Y" and the n-th Point machine is in an End position "Y".]<br>**1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point machine is in an End position "Y".<br>**else alt** [The n-th Point machine is not in an End position "Y" and the 1st Point machine is in an End position "Y".]<br>**1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point machine is in an End position "Y".<br>**else alt** [The n-th Point machine is not in an End position "Y" and the 1st Point machine is not in an End position "Y".]<br>**par**<br>**1.c1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point machine is in an End position "Y".<br>**also par**<br>**1.c1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point machine is in an End position "Y".<br>**end par**<br>**end alt**<br>**Interaction 2.2.2.2.3.B:**<br>**2.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "Y".<br>**Postcondition:**<br>The Subsystem - Point is in an End position "Y".<br>  <br> | | Basic 4-wire multiple P |
| Eu.P.6764 | Info | Alternative Scenario: Handle and report opposite End Position with multiple Point machines via 4-wire interface [P SD 2.2.2.2.4]<br>**P_UC2.2.2.2: Irregularities**<br>Alternative Scenario: Handle and report opposite End Position with multiple Point machines via 4-wire interface [P SD 2.2.2.2.4]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is End position "Y".<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- No end position.<br>**Interaction 2.2.2.2.4.A:**<br>**alt** [The 1st Point machine is in End position "X".]<br>**1.a1** - The Subsystem - Point receives from the 1st Point machine the pattern that represents the opposite End position "X", which is interpreted as Information that the Point machine is in an Unintended position.<br>**else alt** [The n-th Point machine is in End position "X".]<br>**1.b1** - The Subsystem - Point receives from the n-th Point machine the pattern that represents the opposite End position "X", which is interpreted as Information that the Point machine is in an Unintended position.<br>**end alt**<br>**Interaction 2.2.2.2.4.B:**<br>**2.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in a Unintended position.<br>**Postcondition:**<br>The Subsystem - Point is in a Unintended position.<br>  <br> | Note:<br>The Subsystem - Point receives from the Point machine the pattern that represents the opposite end position "X", which is interpreted as Information that the Point machine is in an Unintended position. | Basic 4-wire multiple P |

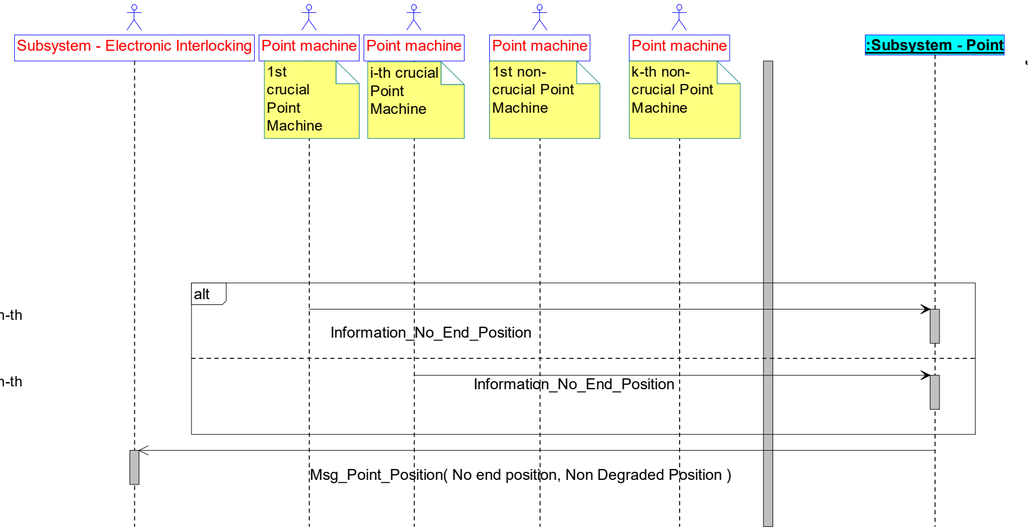
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6761 | Info | Alternative Scenario: Handle and report loss of ability to move point with multiple Point machines [P SD 2.2.2.2.5]<br>**P_UC2.2.2.2: Irregularities**<br>Alternative Scenario: Handle and report loss of ability to move point with multiple Point machines [P SD 2.2.2.2.5]<br>**Precondition**:<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine.<br>The Subsystem - Point is configured to observe the Ability to move point.<br>The Subsystem - Point is Able to move point.<br>**Interaction 2.2.2.2.5.A**:<br>**1.** - The Subsystem - Point internal trigger indicates that the Point is Unable to move point.<br>**2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that it is Unable to move point.<br>**Interaction 2.2.2.2.5.B**:<br>**par**<br>**opt** [The1st point machine is in Moving]<br>**3.a1.a1** - The Subsystem - Point sends the Command to the 1st Point machine to Stop moving the 1st Point machine.<br>**end opt**<br>**also par**<br>**opt** [The n-th Point machine is in Moving and has drive capability]<br>**3.b1.a1** - The Subsystem - Point sends the Command to the n-th Point machine to Stop moving the n-th Point machine.<br>**end opt**<br>**end par**<br>**Postcondition:**<br>The Subsystem - Point is Unable to move point.<br>  <br> | Only applicable if the package [Option Able to move] is used in combination with [Basic 4-wire multiple P]. | Option Able to move |
| Eu.P.6767 | Info | Alternative Scenario: Handle and report restoring of Ability to move point with multiple Point machines [P SD 2.2.2.2.6]<br>**P_UC2.2.2.2: Irregularities**<br>Alternative Scenario: Handle and report restoring of Ability to move point with multiple Point machines [P SD 2.2.2.2.6]<br>**Precondition**:<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine.<br>The Subsystem - Point is configured to observe the Ability to move point.<br>The Subsystem - Point is Unable to move point.<br>**Interaction 2.2.2.2.6.A**:<br>**1.** - The Subsystem - Point internal trigger indicates that the Point is Able to move point.<br>**2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that it is Able to move point.<br>**Postcondition:**<br>The Subsystem - Point is Able to move point.<br>  <br> | Only applicable if the package [Option Able to move] is used in combination with [Basic 4-wire multiple P]. | Option Able to move |

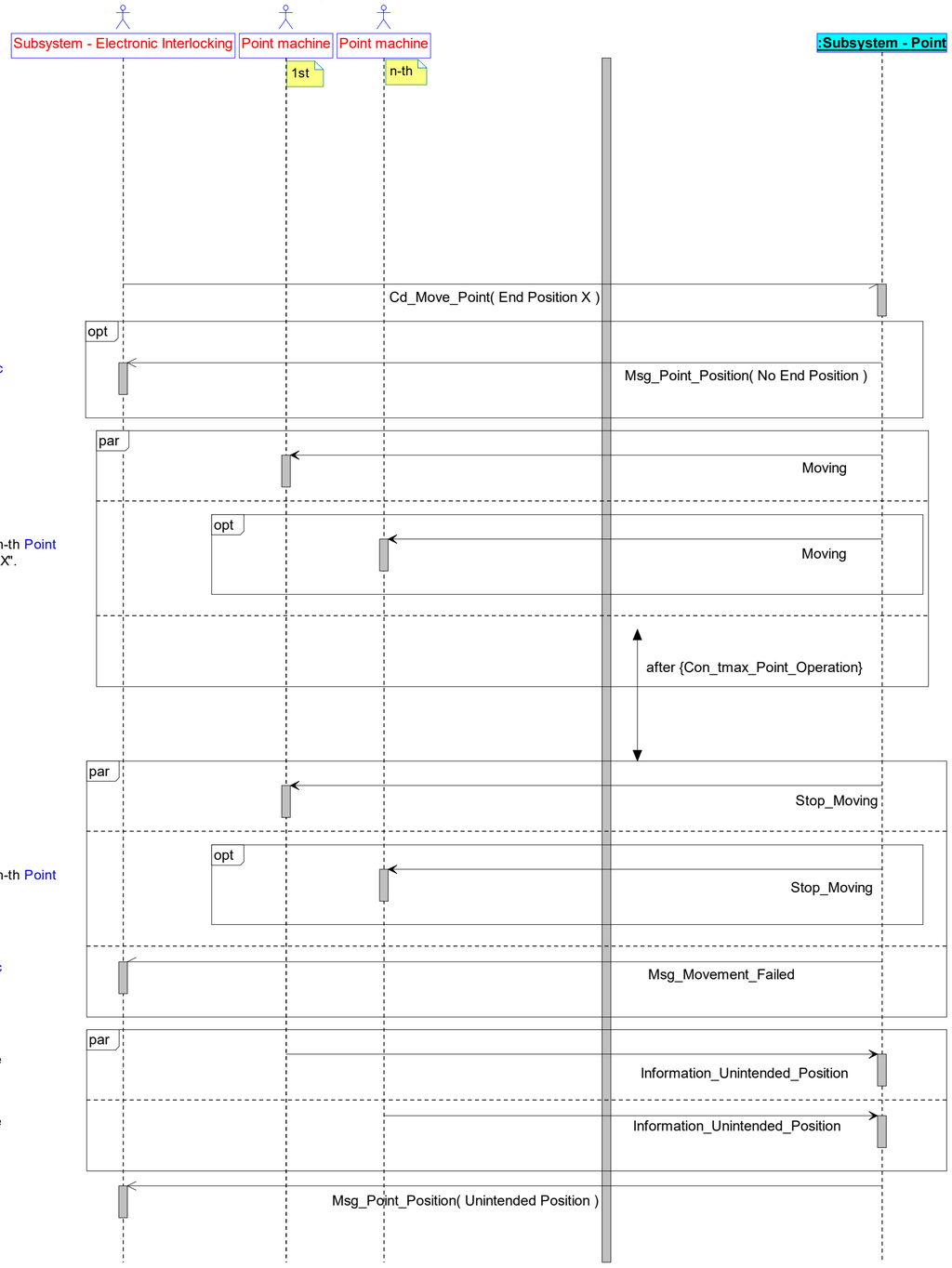
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.7380 | Info | Alternative Scenario: Handle and report Degraded Point Position with multiple Point machines after End Position case 1 [P SD 2.2.2.2.7]<br>**P_UC2.2.2.2: Irregularities**<br>Alternative Scenario: Handle and report Degraded Point Position with multiple Point machines after End Position case 1 [P SD 2.2.2.2.7]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is in a Non degraded position.<br>The Subsystem - Point is in an End position "X".<br>**Interaction 2.2.2.2.7.A**<br>**alt**<br>**1.a1** - The Subsystem - Point receives from the non-crucial 2nd Point machine the Information that the non-crucial 2nd Point machine is in No end position.<br>**else alt**<br>**1.b1** - The Subsystem - Point receives from the non-crucial n-th Point machine the Information that the non-crucial n-th Point machine is in No end position.<br>**end alt**<br>**Interaction 2.2.2.2.7.B**<br>**2.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in a Degraded point position "X"<br>**Postcondition**:<br>The Subsystem - Point is in a Degraded point position "X"<br>  <br> | | Basic 4-wire multiple P |

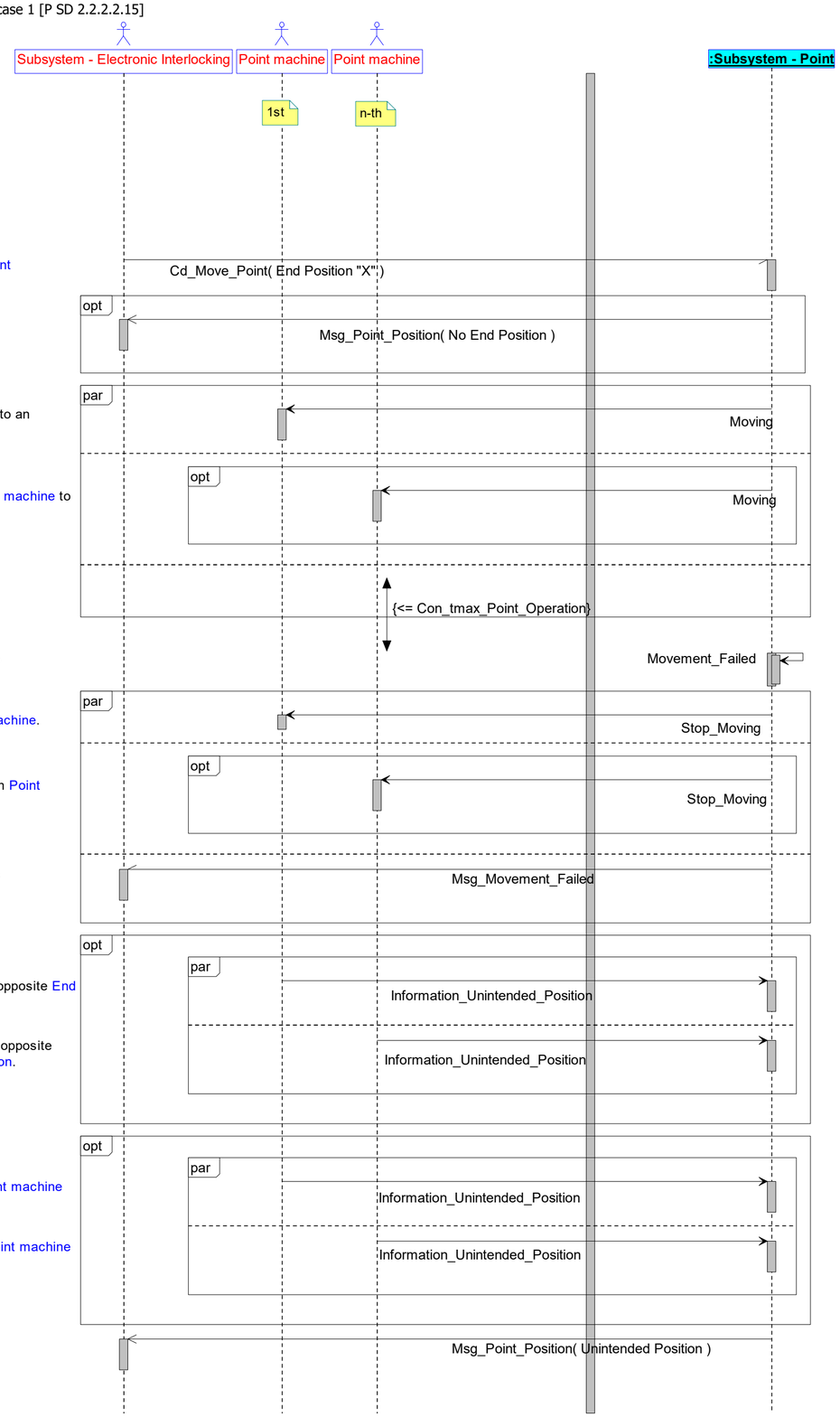
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.7381 | Info | Alternative Scenario: Handle and report Degraded Point Position with multiple Point machines after End Position case 2 [P SD 2.2.2.2.8]<br>**P_UC2.2.2.2: Irregularities**<br>Alternative Scenario: Handle and report Degraded Point Position with multiple Point machines after End Position case 2 [P SD 2.2.2.2.8]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is in a Non degraded position.<br>The Subsystem - Point is in an End position "X".<br>**Interaction 2.2.2.2.8.A**<br>**alt**<br>**1.a1** - The Subsystem - Point receives from the non-crucial 2nd Point machine the Information that the non-crucial 2nd Point machine is in a Unintended position.<br>**else alt**<br>**1.b1** - The Subsystem - Point receives from the non-crucial n-th Point machine the Information that the non-crucial n-th Point machine is in a Unintended position.<br>**end alt**<br>**Interaction 2.2.2.2.8.B**<br>**2.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in a Degraded point position "X"<br>**Postcondition**:<br>The Subsystem - Point is in a Degraded point position "X"<br>  <br> | | Basic 4-wire multiple P |

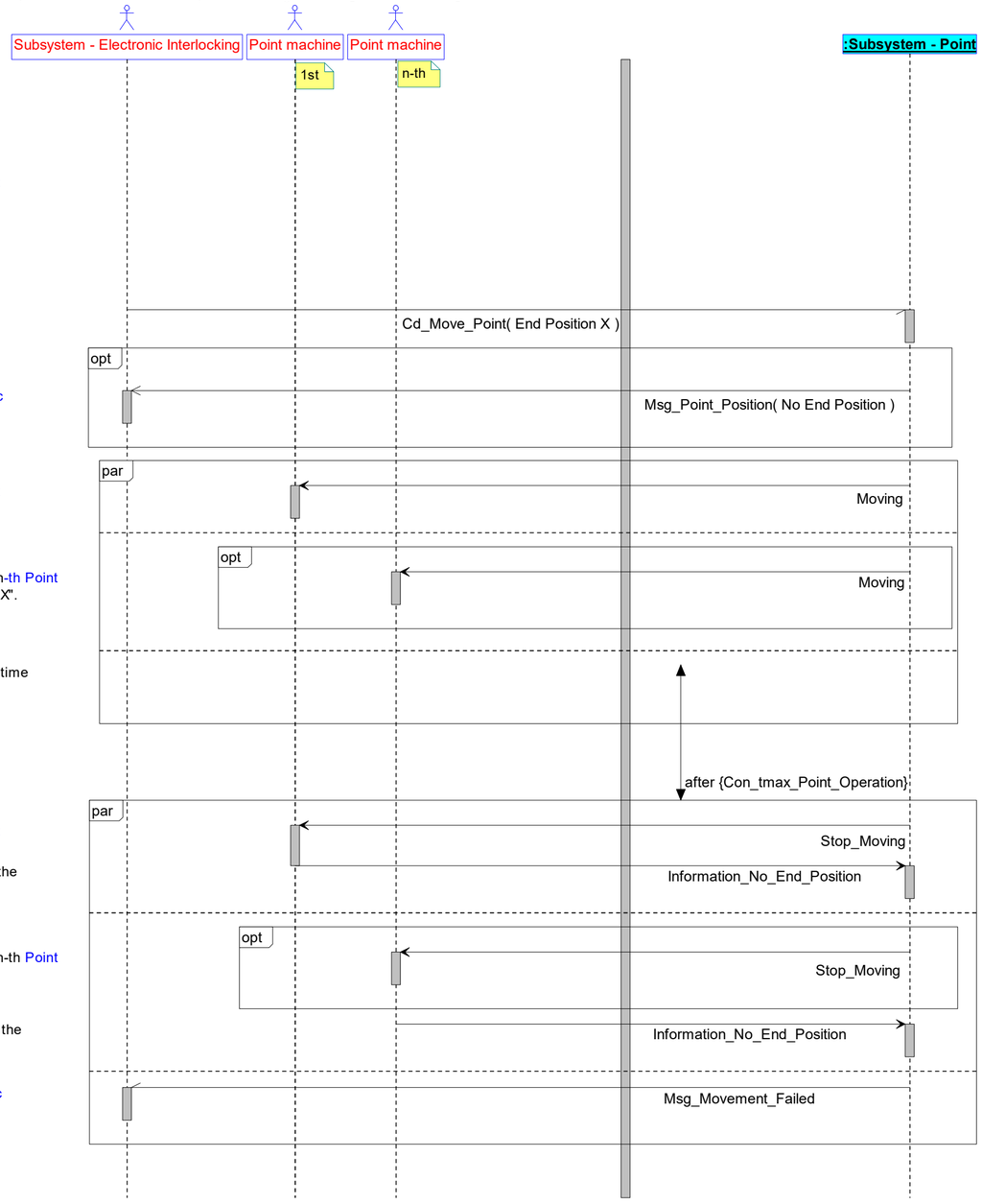
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.7382 | Info | Alternative Scenario: Handle and report non-degraded Point Position with multiple Point machines [P SD 2.2.2.2.9]<br>**P_UC2.2.2.2: Irregularities**<br>Alternative Scenario: Handle and report non-degraded Point Position with multiple Point machines [P SD 2.2.2.2.9]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is End position "X".<br>The Subsystem - Point is in a Degraded point position "X"<br>One or more non-crucial Point machine are in No end position or a Unintended position.<br>**Interaction 2.2.2.2.9.A**<br>**par**<br>**1.a1** - The Subsystem - Point receives the Information that the 1st non-crucial Point machine is in an detected End position "X".<br>**1.b1** - The Subsystem - Point receives the Information that the k-th non-crucial Point machine is in an detected End position "X".<br>**end par**<br>**Interaction 2.2.2.2.9.B**<br>**2.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is not in a Degraded point position "X"<br>**Postcondition**:<br>The Subsystem - Point is in a Non degraded position.<br>  <br> | | Basic 4-wire multiple P |
| Eu.P.7379 | Info | Alternative Scenario: Handle and report Degraded Point Position with multiple Point machines after crucial End Positon - case 1 [P SD 2.2.2.2.10]<br>**P_UC2.2.2.2: Irregularities**<br>Alternative Scenario: Handle and report Degraded Point Position with multiple Point machines after crucial End Positon - case 1 [P SD 2.2.2.2.10]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is End position "X".<br>The Subsystem - Point is in a Non degraded position.<br>One or more crucial Point machine are in No end position or a Unintended position.<br>One or more non-crucial Point machine are in No end position.<br>**Interaction 2.2.2.2.10.A**<br>**par**<br>**1.a1** - The Subsystem - Point receives the Information that the 1st crucial Point machine is in an detected End position "X".<br>**also par**<br>**1.b1** - The Subsystem - Point receives the Information that the n-th crucial Point machine is in an detected End position "X".<br>**end par**<br>**Interaction 2.2.2.2.10.B**<br>**2.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in a Degraded point position "X".<br>**Postcondition**:<br>The Subsystem - Point is in a Degraded point position "X"<br>  <br> | | Basic 4-wire multiple P |

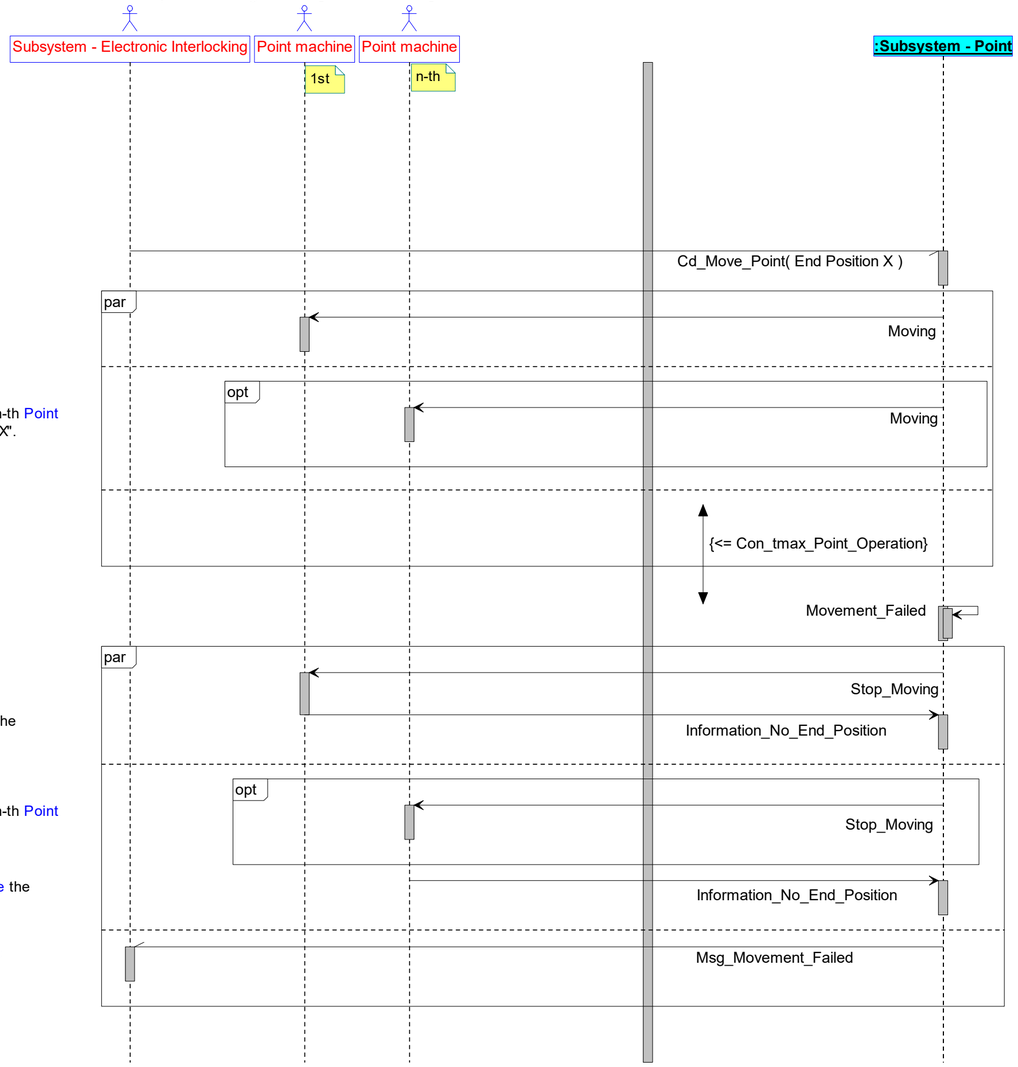
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.7437 | Info | Alternative Scenario: Handle and report Degraded Point Position with multiple Point machines after crucial End Positon - case 2 [P SD 2.2.2.2.11]<br>**P_UC2.2.2.2: Irregularities**<br>Alternative Scenario: Handle and report Degraded Point Position with multiple Point machines after crucial End Positon - case 2 [P SD 2.2.2.2.11]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is End position "X".<br>The Subsystem - Point is in a Non degraded position.<br>One or more crucial Point machine are in No end position or a Unintended position.<br>One or more non-crucial Point machine are in an Unintended position.<br>**Interaction 2.2.2.2.11.A**<br>**par**<br>**1.a1** - The Subsystem - Point receives the Information that the 1st crucial Point machine is in an detected End position "X".<br>**also par**<br>**1.b1** - The Subsystem - Point receives the Information that the n-th crucial Point machine is in an detected End position "X".<br>**end par**<br>**Interaction 2.2.2.2.11.B**<br>**2.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in a Degraded point position "X".<br>**Postcondition**:<br>The Subsystem - Point is in a Degraded point position "X"<br>  <br> | | Basic 4-wire multiple P |
| Eu.P.7384 | Info | Alternative Scenario: Handle and report non-degraded Point Position with multiple Point machines after crutial Unintended Position [P SD 2.2.2.2.12]<br>**P_UC2.2.2.2: Irregularities**<br>Alternative Scenario: Handle and report non-degraded Point Position with multiple Point machines after crutial Unintended Position [P SD 2.2.2.2.12]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is in a Degraded point position "X"<br>One or more non-crucial Point machine are in No end position or a Unintended position.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is End position "X".<br>**Interaction 2.2.2.2.12.A**<br>**alt**<br>**1.a1** - The Subsystem - Point receives the Information that the 1st crucial Point machine is in an detected Unintended position.<br>**else alt**<br>**1.b1** - The Subsystem - Point receives the Information that the n-th crucial Point machine is in an detected Unintended position.<br>**end alt**<br>**Interaction 2.2.2.2.12.B**<br>**2.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is not in a Degraded point position "X"<br>**Postcondition**:<br>The Subsystem - Point is in a Non degraded position.<br>  <br> | | Basic 4-wire multiple P |

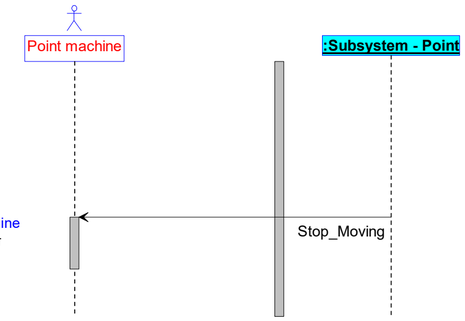
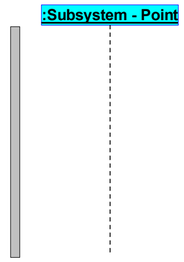
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.7383 | Info | Alternative Scenario: Handle and report non-degraded Point Position with multiple Point machines after crucial No End Position [P SD 2.2.2.2.13]<br>**P_UC2.2.2.2: Irregularities**<br>Alternative Scenario: Handle and report non-degraded Point Position with multiple Point machines after crucial No End Position [P SD 2.2.2.2.13]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is in a Degraded point position "X"<br>One or more non-crucial Point machine are in No end position or a Unintended position.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is End position "X".<br>**Interaction 2.2.2.2.13.A**<br>**alt**<br>**1.a1** - The Subsystem - Point receives the Information that the crucial n-th Point machine is in an detected No end position.<br>**else alt**<br>**1.b1** - The Subsystem - Point receives the Information that the crucial n-th Point machine is in an detected No end position.<br>**end alt**<br>**2**. - The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is not in a Degraded point position "X"<br>**Postcondition**:<br>The Subsystem - Point is in a Non degraded position.<br>  <br> | | Basic 4-wire multiple P |

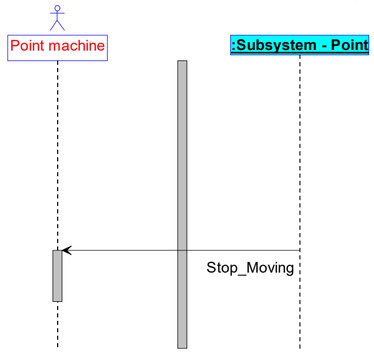
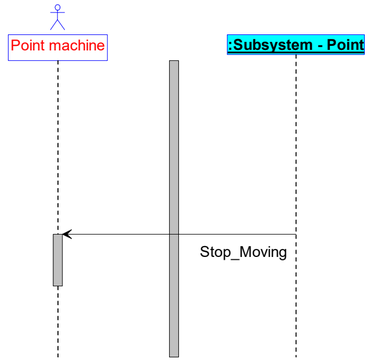
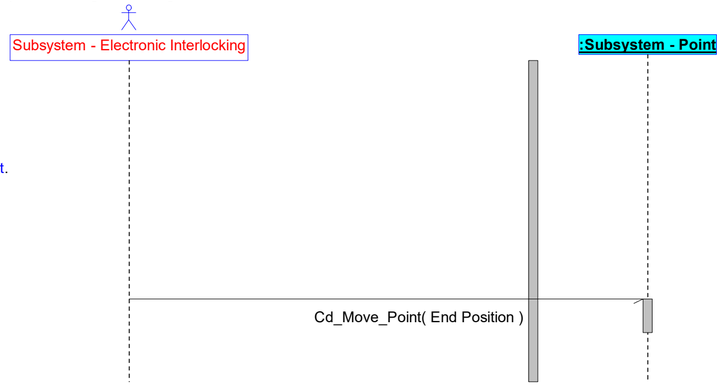
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6765 | Info | Alternative Scenario: Handle and report Point operation timeout with multiple Point machines with position change case 1 [P SD 2.2.2.2.14]<br>**P_UC2.2.2.2: Irregularities**<br>Alternative Scenario: Handle and report Point operation timeout with multiple Point machines with position change case 1 [P SD 2.2.2.2.14]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y",<br>- No end position, or<br>- a Unintended position.<br>**Interaction 2.2.2.2.14.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**opt** [The Subsystem - Point was previously in an End position or a Unintended position ]<br>**2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**end opt**<br>**par**<br>**3.a1** The Subsystem - Point sends the Command to the 1st Point machine to move the 1st Point machine to an End position "X".<br>**also par**<br>**opt** [The n-th Point machine has drive capability]<br>**3.b1.a1** The Subsystem - Point sends the Command to the n-th Point machine to move the n-th Point machine to an End position "X".<br>**end opt**<br>**also par**<br>**3.c1** The Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**end par**<br>**Interaction 2.2.2.2.14.B:**<br>**4.** - The Subsystem - Point detects that the time period Con_tmax_Point_Operation has exceeded.<br>**par**<br>**5.a1** The Subsystem - Point sends the Command to the 1st Point machine to stop moving the 1st Point machine.<br>**also par**<br>**opt** [The n-th Point machine has drive capability]<br>**5.b1.a1** The Subsystem - Point sends the Command to the n-th Point machine to stop moving the n-th Point machine.<br>**end opt**<br>**also par**<br>**5.c1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred.<br>**end par**<br>**par**<br>**6.a1** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in a Unintended position.<br>**also par**<br>**6.b1** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in a Unintended position.<br>**end par**<br>**7.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in Unintended position.<br>**Postcondition:**<br>The Subsystem - Point is in Unintended position.<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. The timing related to driving individual (1 to n-th) Point Machines for a single Point is supplier specific and is specified only in a general way. The timing related to individual Point Machines is not specified by EULYNX as part of the application layer, this shall be handled by the physical implementation. | Basic 4-wire multiple P |

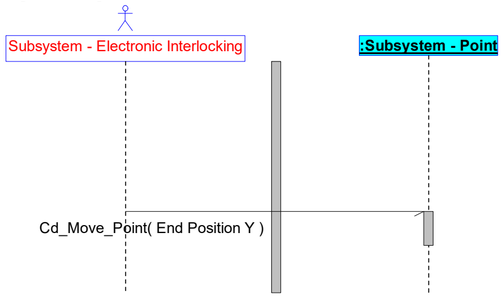
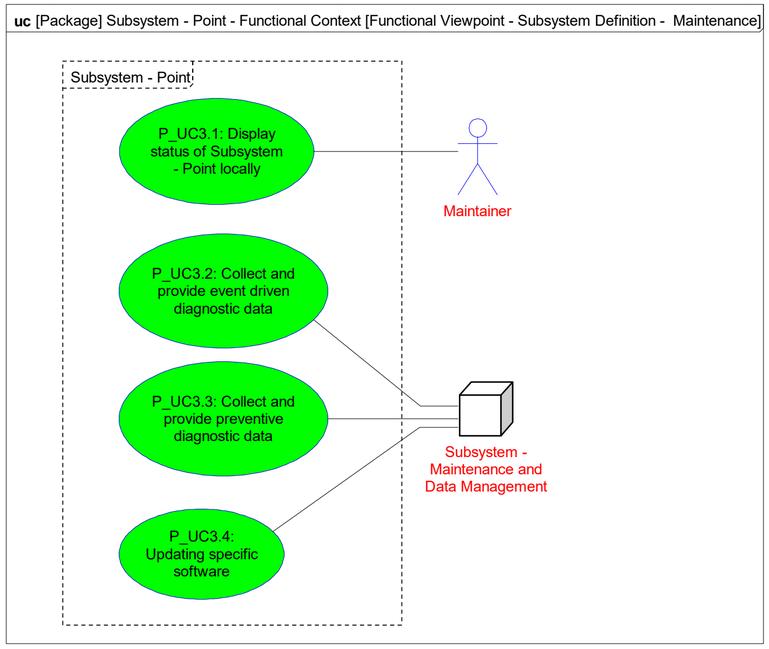
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6759 | Info | Alternative Scenario: Handle and report failed movement with multiple Point machines without position change case 1 [P SD 2.2.2.2.15]<br>**P_UC2.2.2.2: Irregularities**<br>Alternative Scenario: Handle and report failed movement with multiple Point machines without position change case 1 [P SD 2.2.2.2.15]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- an Unintended position.<br>**Interaction 2.2.2.2.15.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**opt** [The Subsystem - Point was previously in an End position or a Unintended position ]<br>**2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**end opt**<br>**par**<br>**3.a1** The Subsystem - Point sends the Command to the 1st Point machine to move the 1st Point machine to an End position "X".<br>**also par**<br>**opt** [The n-th Point machine has drive capability]<br>**3.b1.a1** The Subsystem - Point sends the Command to the n-th Point machine to move the n-th Point machine to an End position "X".<br>**end opt**<br>**also par**<br>**3.c1** The Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**end par**<br>**Interaction 2.2.2.2.15.B:**<br>**4.** - A failure occurred during the movement resulting in a Failed Movement. The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br>**par**<br>**5.a1** The Subsystem - Point sends the Command to the 1st Point machine to stop moving the 1st Point machine.<br>**also par**<br>**opt** [The n-th Point machine has drive capability]<br>**5.b1.a1** The Subsystem - Point sends the Command to the n-th Point machine to stop moving the n-th Point machine.<br>**end opt**<br>**also par**<br>**5.c1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred.<br>**end par**<br>**opt** [The Subsystem - Point was in an End position "Y".]<br>**par**<br>**6.a1.a1** - The Subsystem - Point receives from the 1st Point machine the pattern that represents the opposite End position "Y", which is interpreted as Information that the Point machine is in an Unintended position.<br>**also par**<br>**6.b1.a1** - The Subsystem - Point receives from the n-th Point machine the pattern that represents the opposite End position "Y", which is interpreted as Information that the Point machine is in an Unintended position.<br>**end par**<br>**end opt**<br>**opt** [The Subsystem - Point was in a Unintended position.]<br>**par**<br>**7.a1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the 1st Point machine is in a Unintended position.<br>**also par**<br>**7.b1.a1** - The Subsystem - Point receives from the n-th Point machine the Information that the n-th Point machine is in a Unintended position.<br>**end par**<br>**end opt**<br>**8.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in Unintended position.<br>**Postcondition:**<br>The Subsystem - Point is in Unintended position.<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. The timing related to driving individual (1 to n-th) Point Machines for a single Point is supplier specific and is specified only in a general way. The timing related to individual Point Machines is not specified by EULYNX as part of the application layer, this shall be handled by the physical implementation. | Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6766 | Info | Alternative Scenario: Handle and report Point operation timeout with multiple Point machines without position change case 2 [P SD 2.2.2.2.16]<br>**P_UC2.2.2.2: Irregularities**<br>Alternative Scenario: Handle and report Point operation timeout with multiple Point machines without position change case 2 [P SD 2.2.2.2.16]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine.<br>The Subsystem - Point is in:<br>- an End position "Y", or<br>- No end position, or<br>- an Unintended position.<br>**Interaction 2.2.2.2.16.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**opt** [The Subsystem - Point was previously in an End position or a Unintended position ]<br>**2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.<br>**end opt**<br>**par**<br>**3.a1** The Subsystem - Point sends the Command to the 1st Point machine to move the 1st Point machine to an End position "X".<br>**also par**<br>**opt** [The n-th Point machine has drive capability]<br>**3.b1.a1** The Subsystem - Point sends the Command to the n-th Point machine to move the n-th Point machine to an End position "X".<br>**end opt**<br>**also par**<br>**3.c1** At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**end par**<br>**Interaction 2.2.2.2.16.B:**<br>**4.** - The Subsystem - Point detects that the time period Con_tmax_Point_Operation has exceeded.<br>**par**<br>**5.a1** The Subsystem - Point sends the Command to the 1st Point machine to stop moving the 1st Point machine.<br>**5.a2** The Subsystem - Point receives from the 1st Point machine the Information that the 1st Point machine is in No end position.<br>**also par**<br>**opt** [The n-th Point machine has drive capability]<br>**5.b1.a1** The Subsystem - Point sends the Command to the n-th Point machine to stop moving the n-th Point machine.<br>**end opt**<br>**5.b2** The Subsystem - Point receives from the n-th Point machine the Information that the n-th Point machine is in No end position.<br>**also par**<br>**5.c1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred.<br>**end par**<br>**Postcondition:**<br>The Subsystem - Point is in No end position.<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. The timing related to driving individual (1 to n-th) Point Machines for a single Point is supplier specific and is specified only in a general way. The timing related to individual Point Machines is not specified by EULYNX as part of the application layer, this shall be handled by the physical implementation. | Basic 4-wire multiple P |

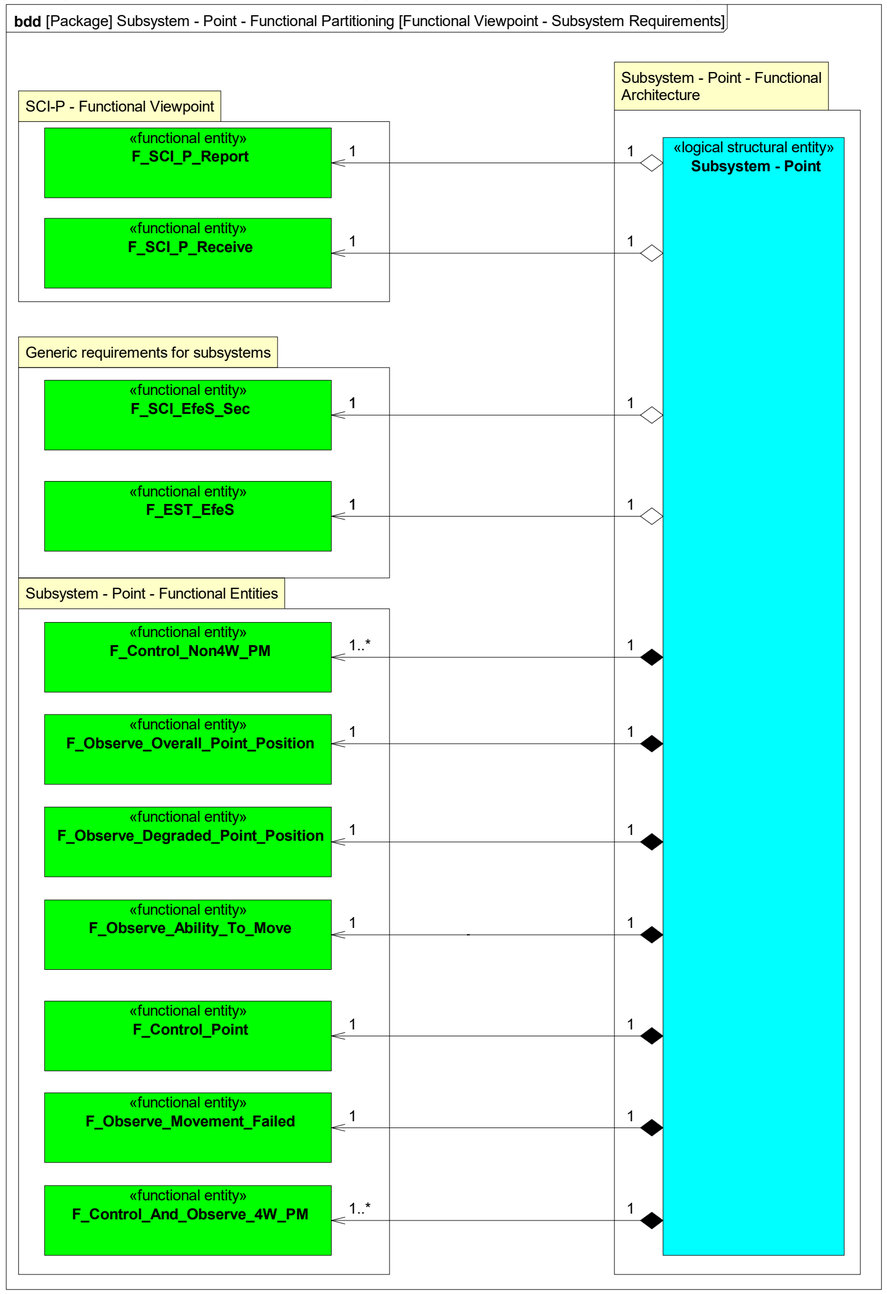
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6760 | Info | Alternative Scenario: Handle and report failed movement with multiple Point machines without position change case 2 [P SD 2.2.2.2.17]<br>**P_UC2.2.2.2: Irregularities**<br>Alternative Scenario: Handle and report failed movement with multiple Point machines without position change case 2 [P SD 2.2.2.2.17]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured with a 4-wire interface to the Point machine.<br>The Subsystem - Point is in No end position.<br>**Interaction 2.2.2.2.17.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".<br>**par**<br>**2.a1** The Subsystem - Point sends the Command to the 1st Point machine to move the 1st Point machine to an End position "X".<br>**also par**<br>**opt** [The n-th Point machine has drive capability]<br>**2.b1.a1** The Subsystem - Point sends the Command to the n-th Point machine to move the n-th Point machine to an End position "X".<br>**end opt**<br>**also par**<br>**2.c1** The Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.<br>**end par**<br>**Interaction 2.2.2.2.17.B:**<br>**3.** The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br>**par**<br>**4.a1** The Subsystem - Point sends the Command to the 1st Point machine to stop moving the 1st Point machine.<br>**4.a2** The Subsystem - Point receives from the 1st Point machine the Information that the 1st Point machine is in No end position.<br>**also par**<br>**opt** [The n-th Point machine has drive capability]<br>**4.b1.a1** The Subsystem - Point sends the Command to the n-th Point machine to stop moving the n-th Point machine.<br>**end opt**<br>**4.b2** - The Subsystem - Point receives from the n-th Point machine the Information that the n-th Point machine is in No end position.<br>**also par**<br>**4.c1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred.<br>**end par**<br>**Postcondition:**<br>The Subsystem - Point is in No end position.<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.<br>The timing related to driving individual (1 to n-th) Point Machines for a single Point is supplier specific and is specified only in a general way. The timing related to individual Point Machines is not specified by EULYNX as part of the application layer, this shall be handled by the physical implementation. | Basic 4-wire multiple P |
| Eu.P.1251 | Info | P_UC2.3: Handle general Irregularities | The Subsystem-UseCase P_UC2.3: Handle general Irregularities defines the behaviour of the Subsystem - Point when an irregularity occurs. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

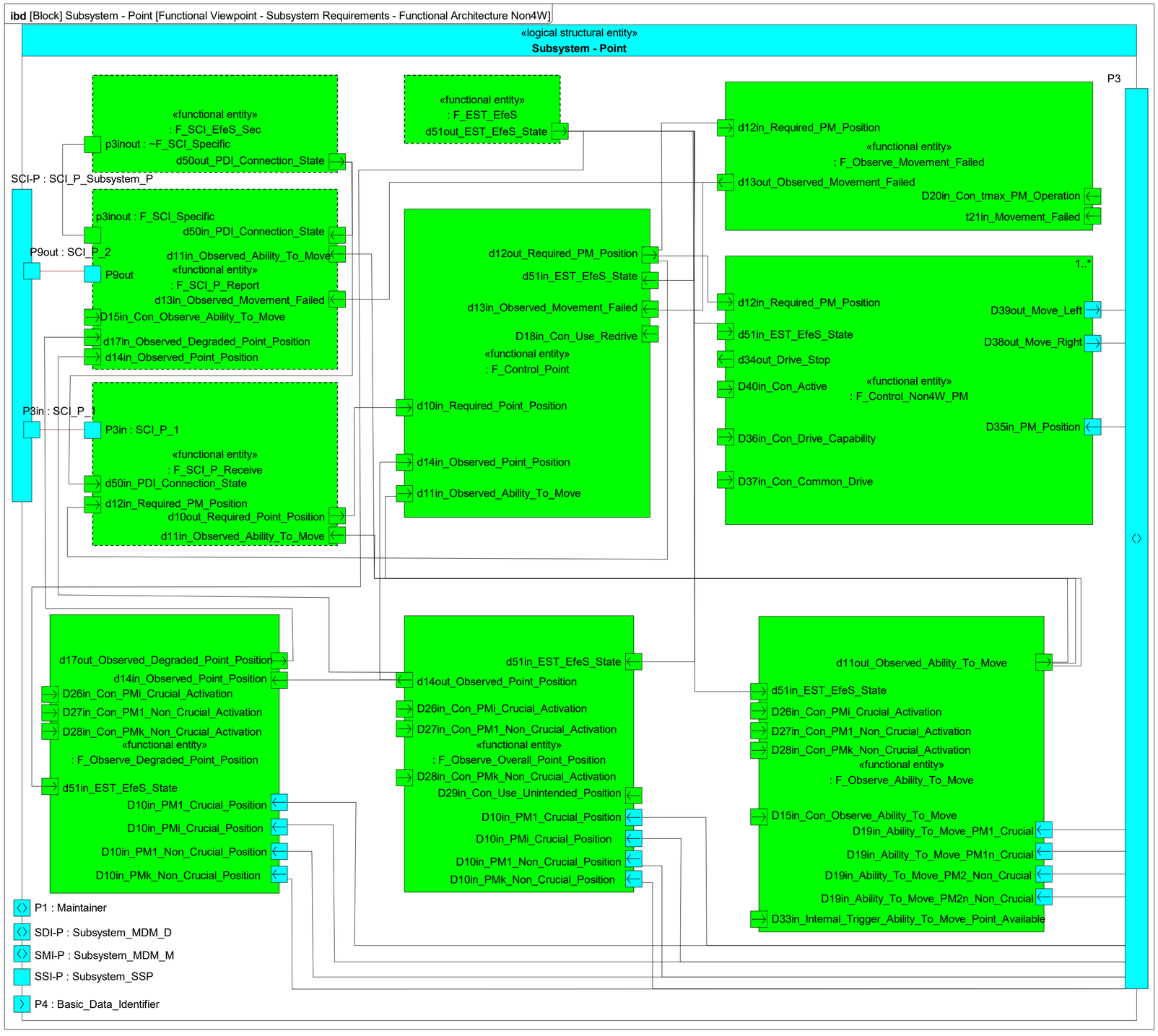
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.1252 | Info | Alternative Scenario: Perform fallback operation [P SD 2.3.1]<br>**P_UC2.3: Handle general Irregularities**<br>Alternative Scenario: Perform fallback operation [P SD 2.3.1]<br>**Precondition:**<br>The Subsystem - Point monitors the Timevalue "Con_tmax_Point_Operation".<br>**Interaction 2.3.1.A:**<br>**1.** - The Subsystem - Point enters the state FALLBACK_MODE.<br>**2.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point. The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br>**Postcondition:**<br>The Subsystem - Point is in the state FALLBACK_MODE.<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.1299 | Info | Alternative Scenario: Handling of interrupted PDI connection [P SD 2.3.2]<br>**P_UC2.3: Handle general Irregularities**<br>Alternative Scenario: Handling of interrupted PDI connection [P SD 2.3.2]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>**Interaction 2.3.2.A:**<br>**1.** - The PDI connection has been terminated.<br>**Postcondition:**<br>The Subsystem - Point is in the state INITIALISING.<br>The Process Data Interface protocol connection is terminated.<br>  <br> | The following functionality remains available within the state INITIALISING after the termination of the PDI connection:<br>• Redrive Point operation (for example as in Main Success Scenario: Redrive Point after lost end position Non 4W [P SD 2.1.1.2.1], Alternative Scenario: Handle Point operation timeout during Redrive Non 4W [P SD 2.1.1.3.9], Main Success Scenario: Redrive Point after lost end position [P SD 2.1.2.2.1], Alternative Scenario: Handle and report non-degraded Point Position with multiple Point machines after crutial Unintended Position [P SD 2.1.2.4.11])<br>• Stop ongoing movement when Ability to move point is lost (for example as in Alternative Scenario: Handle and report loss of ability to move point with single point machine [P SD 2.1.1.3.7], Alternative Scenario: Handle and report non-degraded Point Position with multiple Point machines [P SD 2.1.2.4.9], Alternative Scenario: Handle and report Degraded Point Position with multiple Point machines after crucial End Position [P SD 2.1.2.4.10], Alternative Scenario: Handle and report loss of ability to move point with single point machine [P SD 2.2.1.2.5], Alternative Scenario: Handle and report loss of ability to move point with multiple Point machines [P SD 2.2.2.2.5])<br>• Restoration of Ability to move point (for example as in Alternative Scenario: Handle and report restoring of Ability to move point with single point machine [P SD 2.1.1.3.8], Alternative Scenario: Handle and report restoring of Ability to move point with single point machine [P SD 2.2.1.2.6])<br>• All functionality related to the maintainer interface P1 (for example as in P_UC3.1: Display status of Subsystem - Point locally)<br><br>Note: If the PDI Connection is terminated during ongoing point movement, the movement is continued as normal. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.3093 | Info | Alternative Scenario: Supply voltage of the Subsystem has gone outside of the required range for operation [P SD 2.3.3]<br>**P_UC2.3: Handle general Irregularities**<br>Alternative Scenario: Supply voltage of the Subsystem has gone outside of the required range for operation [P SD 2.3.3]<br>**Precondition:**<br>The Subsystem - Point monitors the Timevalue "Con_tmax_Point_Operation".<br>**Interaction 2.3.3.A:**<br>**1.** - The Subsystem - Point enters the state NO_OPERATING_VOLTAGE.<br>**2.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point. The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br>**Postcondition:**<br>The Subsystem - Point is in the state NO_OPERATING_VOLTAGE.<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.3104 | Info | Alternative Scenario: Reset occurs [P SD 2.3.4]<br>**P_UC2.3: Handle general Irregularities**<br>Alternative Scenario: Reset occurs [P SD 2.3.4]<br>**Precondition:**<br>The Subsystem - Point is in the state INITIALISING or OPERATIONAL.<br>The Subsystem - Point monitors the Timevalue "Con_tmax_Point_Operation".<br>**Interaction 2.3.4.A:**<br>**1.** - A reset has occurred.<br>**2.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point. The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.<br>**3.** The Subsystem - Point enters the state BOOTING.<br>**Postcondition:**<br>The Subsystem - Point is in the state BOOTING.<br>  <br> | Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5371 | Info | Alternative Scenario: Handle received move command if ability to move point is lost [P SD 2.3.5]<br>**P_UC2.3: Handle general Irregularities**<br>Alternative Scenario: Handle received move command if ability to move point is lost [P SD 2.3.5]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is configured to observe the Ability to move point.<br>The Subsystem - Point is Unable to move point.<br>The Subsystem - Point is in:<br>- an End position, or<br>- No end position, or<br>- an Unintended position.<br>**Interaction 2.3.5.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position.<br>**Postcondition:**<br>-- <br>  <br> | | Option Able to move |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.1469 | Info | Alternative Scenario: Commanding of the Point to the current End position [P SD 2.3.6]<br>**P_UC2.3: Handle general Irregularities**<br>Alternative Scenario: Commanding of the Point to the current End position [P SD 2.3.6]<br>**Precondition:**<br>The Subsystem - Point is in the state OPERATIONAL.<br>The Subsystem - Point is in an End position "Y".<br>**Interaction 2.3.6.A:**<br>**1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "Y".<br>**Postcondition:**<br>---<br>  <br> | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.1127 | Info | [Package] Subsystem - Point - Functional Context [Functional Viewpoint - Subsystem Definition - Maintenance]<br>  <br> | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.1124 | Info | P_UC3.1: Display status of Subsystem - Point locally | Information:<br>The Subsystem-UseCase P_UC3.1: Display status of Subsystem - Point locally defines the local display of the EULYNX field element Subsystem.<br>See ID Eu.P.890. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.1125 | Info | P_UC3.2: Collect and provide event driven diagnostic data | Information:<br>The Subsystem-UseCase P_UC3.2: Collect and provide event driven diagnostic data defines the event driven collection and provision of diagnostic data in case of irregularities.<br>See ID Eu.P.925. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.1126 | Info | P_UC3.3: Collect and provide preventive diagnostic data | Information:<br>The Subsystem-UseCase P_UC3.3: Collect and provide preventive diagnostic data defines the continuous collection and provision of diagnostic data for preventive maintenance.<br>See ID Eu.P.925. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.1468 | Info | P_UC3.4: Updating specific software | Information:<br>The Subsystem-UseCase P_UC3.4: Updating specific software defines the process of updating the specific software between Subsystem - Maintenance and Data Management and the Subsystem. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6272 | Head | **3.3.3 Subsystem - Point - Functional Partitioning** | | |

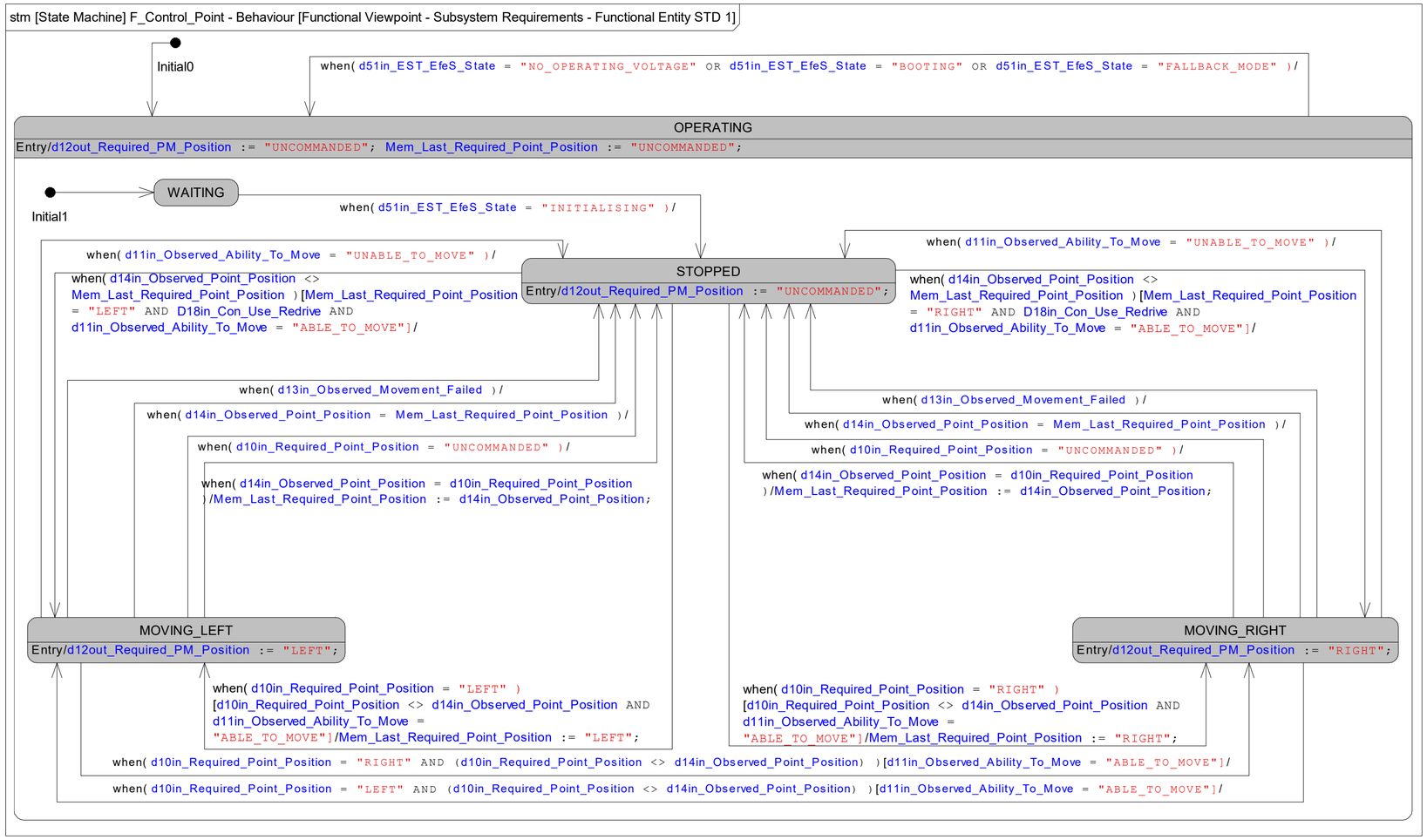
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.3297 | Def | [Package] Subsystem - Point - Functional Partitioning [Functional Viewpoint - Subsystem Requirements]<br>  <br> | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5781 | Head | **3.3.4 Subsystem - Point - Functional Architecture** | | |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.939 | Info | Subsystem - Point | The Subsystem - Point integrates the moveable elements, that may be moved to a different position by a request from the Subsystem - Electronic Interlocking. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6208 | Def | [Block] Subsystem - Point [Functional Viewpoint - Subsystem Requirements - Functional Architecture Non4W]<br>  <br> | The Subsystem - Point shall provide the functional interfaces shown in "[Block] Subsystem - Point [Functional Viewpoint - Subsystem Requirements - Functional Architecture Non4W]", typed by FlowSpecifications. Each FlowSpecification is defined by a set of FlowProperties that specify the possible exchange of information through the particular interface. | Basic non-4-wire single P<br>Basic non-4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6934 | Def | [Block] Subsystem - Point [Functional Viewpoint - Subsystem Requirements - Functional Architecture 4W] | The Subsystem - Point shall provide the functional interfaces shown in "[Block] Subsystem - Point [Functional Viewpoint - Subsystem Requirements - Functional Architecture 4W]", typed by FlowSpecifications. Each FlowSpecification is defined by a set of FlowProperties that specify the possible exchange of information through the particular interface. | Basic 4-wire single P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.942 | Def | P3 | The functional control interface to Point machines for the information flow through the interface, which is defined by the FlowSpecification "Point_machine". | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6378 | Def | P1 | The functional Maintenance/Operation/Display interface to the Maintainer. The InformationFlow through the interface is defined by "Maintainer". | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6379 | Def | P4 | The functional System Data interface to the Basic Data identifier. The InformationFlow through the interface is defined by "Basic_Data_Identifier". | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6380 | Def | SCI-P | The functional Process Data interface to the Subsystem - Electronic Interlocking (SCI: Standard Communication Interface). The InformationFlow through the interface is further defined in SCI_P_Subsystem_EIL. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6381 | Def | SDI-P | 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 non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6382 | Def | SMI-P | 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 non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6383 | Def | SSI-P | The Security Service Interface to the Subsystem - Security Services Platform. The InformationFlow through the interface is further defined in SSI-P (Subsystem - Security Services Platform). | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5809 | Head | **3.3.5 Subsystem - Point - Functional Entities** | | |
| Eu.P.6948 | Info | F_Control_Point | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

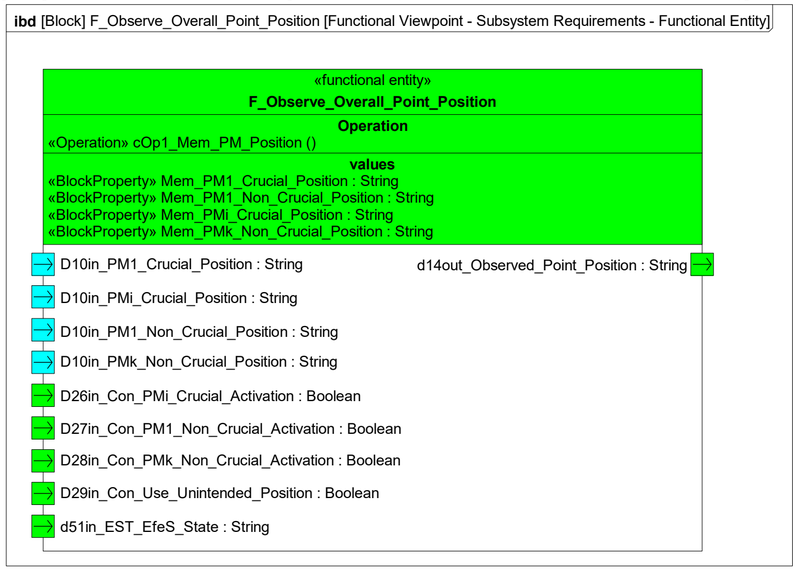
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6949 | Req | [Block] F_Control_Point [Functional Viewpoint - Subsystem Requirements - Functional Entity]<br>**ibd** [Block] F_Control_Point [Functional Viewpoint - Subsystem Requirements - Functional Entity]<br>«functional entity»<br>**F_Control_Point**<br>**values**<br>«BlockProperty» Mem_Last_Required_Point_Position : String<br>d10in_Required_Point_Position : String<br>d11in_Observed_Ability_To_Move : String<br>d13in_Observed_Movement_Failed : Boolean<br>d14in_Observed_Point_Position : String<br>D18in_Con_Use_Redrive : Boolean<br>d51in_EST_EfeS_State : String<br>d12out_Required_PM_Position : String | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6950 | Def | d10in_Required_Point_Position | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6951 | Def | d11in_Observed_Ability_To_Move | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6952 | Def | d13in_Observed_Movement_Failed | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6953 | Def | d14in_Observed_Point_Position | The port d14in_Observed_Point_Position provides the Point machine position to the Subsystem - Point.<br>The port d14in_Observed_Point_Position refines the InformationFlow Information_No_End_Position, Information_End_Position_Reached, Information_End_Position_Detected and Information_Unintended_Position. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6954 | Def | D18in_Con_Use_Redrive | The port D18in_Con_Use_Redrive provides a configuration value to the Point for Redrive point functionality.<br><br>true: Point has Redrive point functionality<br>false: Point has no Redrive point functionality | Option Redrive |
| Eu.P.6956 | Def | d12out_Required_PM_Position | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6957 | Def | d51in_EST_EfeS_State | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6958 | Info | F_Control_Point - Behaviour | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6959 | Req | Functional Viewpoint - Subsystem Requirements - Functional Entity STD 1<br>  <br> | This state machine diagram describes the requirements for the following functionalities:<br><br>- observes and forward the required PM position | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

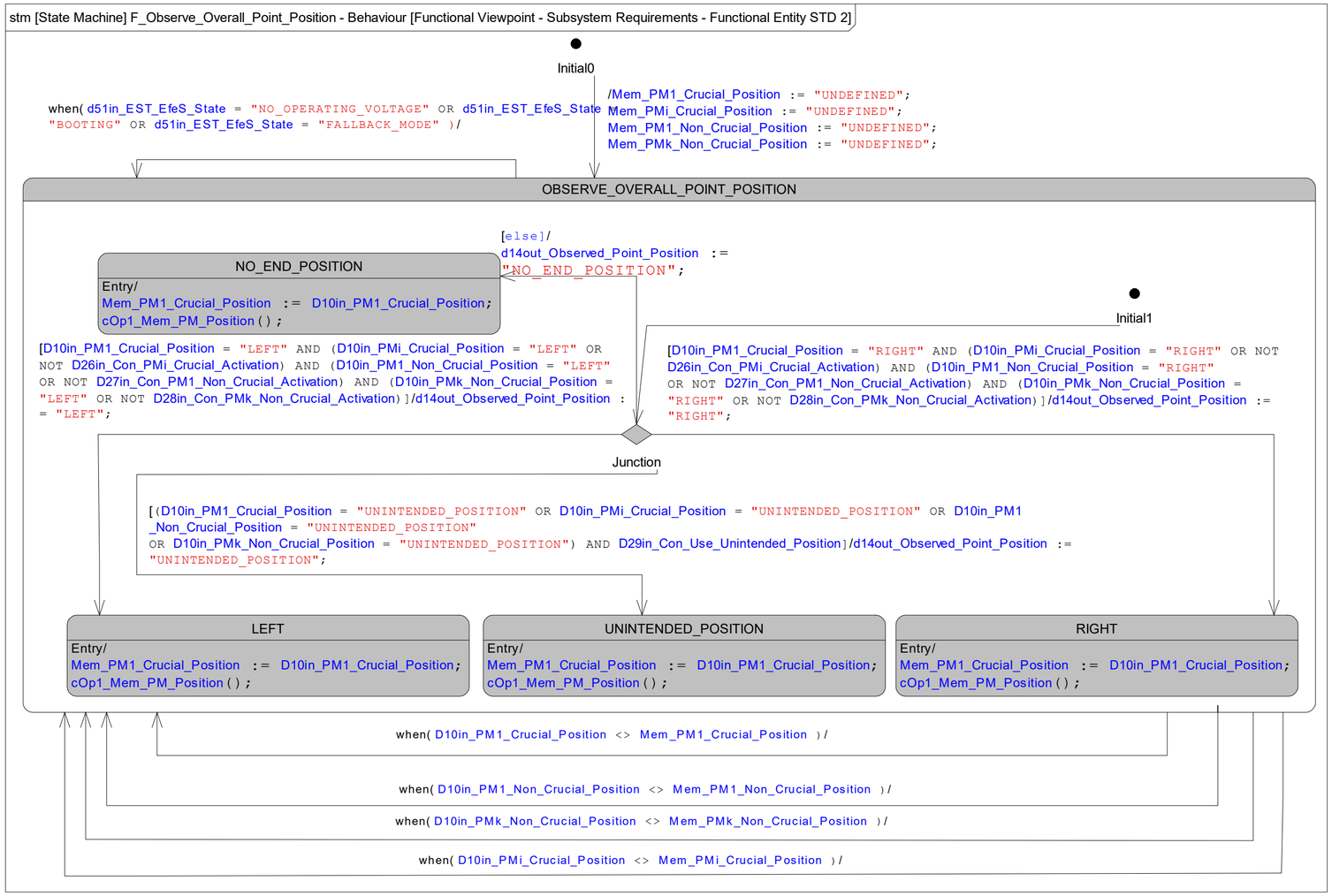
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6960 | Def | Initial0 | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6961 | Def | /{Initial0 - OPERATING} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6962 | Def | OPERATING | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6963 | Def | Initial1 | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6964 | Def | /{Initial1 - WAITING} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6965 | Def | MOVING_LEFT | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6966 | Def | entry/d12out_Required_PM_Position := "LEFT";{State-internal in MOVING_LEFT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6967 | Def | when(d10in_Required_Point_Position = "RIGHT" AND (d10in_Required_Point_Position <> d14in_Observed_Point_Position))[d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE"]/{MOVING_LEFT - MOVING_RIGHT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6968 | Def | when(d10in_Required_Point_Position = "UNCOMMANDED")/{MOVING_LEFT - STOPPED} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6969 | Def | when(d11in_Observed_Ability_To_Move = "UNABLE_TO_MOVE")/{MOVING_LEFT - STOPPED} | | Option Able to move |
| Eu.P.6970 | Def | when(d13in_Observed_Movement_Failed)/{MOVING_LEFT - STOPPED} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6972 | Def | when(d14in_Observed_Point_Position = d10in_Required_Point_Position)/Mem_Last_Required_Point_Position := d14in_Observed_Point_Position;{MOVING_LEFT - STOPPED} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6973 | Def | when(d14in_Observed_Point_Position = Mem_Last_Required_Point_Position)/{MOVING_LEFT - STOPPED} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6974 | Def | MOVING_RIGHT | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6975 | Def | entry/d12out_Required_PM_Position := "RIGHT";{State-internal in MOVING_RIGHT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6976 | Def | when(d10in_Required_Point_Position = "LEFT" AND (d10in_Required_Point_Position <> d14in_Observed_Point_Position))[d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE"]/{MOVING_RIGHT - MOVING_LEFT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6977 | Def | when(d10in_Required_Point_Position = "UNCOMMANDED")/{MOVING_RIGHT - STOPPED} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6978 | Def | when(d11in_Observed_Ability_To_Move = "UNABLE_TO_MOVE")/{MOVING_RIGHT - STOPPED} | | Option Able to move |
| Eu.P.6979 | Def | when(d13in_Observed_Movement_Failed)/{MOVING_RIGHT - STOPPED} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6981 | Def | when(d14in_Observed_Point_Position = d10in_Required_Point_Position)/Mem_Last_Required_Point_Position := d14in_Observed_Point_Position;{MOVING_RIGHT - STOPPED} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| | | | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6982 | Def | when(d14in_Observed_Point_Position = Mem_Last_Required_Point_Position)/{MOVING_RIGHT - STOPPED} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6983 | Def | STOPPED | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6984 | Def | entry/d12out_Required_PM_Position := "UNCOMMANDED";{State-internal in STOPPED} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6985 | Def | when(d10in_Required_Point_Position = "LEFT")[d10in_Required_Point_Position <> d14in_Observed_Point_Position AND d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE"]/Mem_Last_Required_Point_Position := "LEFT";{STOPPED - MOVING_LEFT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6986 | Def | when(d10in_Required_Point_Position = "RIGHT")[d10in_Required_Point_Position <> d14in_Observed_Point_Position AND d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE"]/Mem_Last_Required_Point_Position := "RIGHT";{STOPPED - MOVING_RIGHT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6980 | Def | when(d14in_Observed_Point_Position <> Mem_Last_Required_Point_Position)[Mem_Last_Required_Point_Position = "RIGHT" AND D18in_Con_Use_Redrive AND d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE"]/{STOPPED - MOVING_RIGHT} | | Option Redrive |
| Eu.P.6971 | Def | when(d14in_Observed_Point_Position <> Mem_Last_Required_Point_Position)[Mem_Last_Required_Point_Position = "LEFT" AND D18in_Con_Use_Redrive AND d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE"]/{STOPPED - MOVING_LEFT} | | Option Redrive |
| Eu.P.6987 | Def | entry/d12out_Required_PM_Position := "UNCOMMANDED"; Mem_Last_Required_Point_Position := "UNCOMMANDED";{State-internal in OPERATING} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6988 | Def | WAITING | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6989 | Def | when(d51in_EST_EfeS_State = "INITIALISING")/{WAITING - STOPPED} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

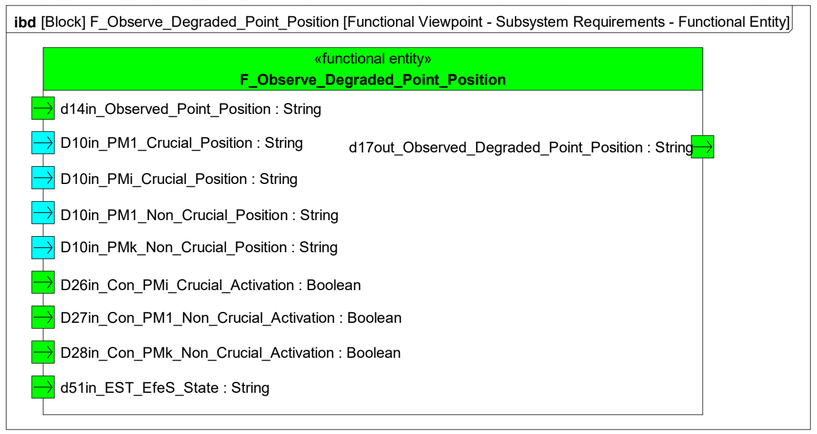
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6990 | Def | when(d51in_EST_EfeS_State = "NO_OPERATING_VOLTAGE" OR d51in_EST_EfeS_State = "BOOTING" OR d51in_EST_EfeS_State = "FALLBACK_MODE")/{OPERATING - OPERATING} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.4548 | Info | F_Observe_Overall_Point_Position | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.4549 | Req | [Block] F_Observe_Overall_Point_Position [Functional Viewpoint - Subsystem Requirements - Functional Entity]<br>  <br> | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6309 | Def | /* cOp1_Mem_PM_Position */<br><br>if D26in_Con_PMi_Crucial_Activation then<br>Mem_PMi_Crucial_Position := D10in_PMi_Crucial_Position;<br>end if<br>if D27in_Con_PM1_Non_Crucial_Activation then<br>Mem_PM1_Non_Crucial_Position := D10in_PM1_Non_Crucial_Position;<br>end if<br>if D28in_Con_PMk_Non_Crucial_Activation then<br>Mem_PMk_Non_Crucial_Position := D10in_PMk_Non_Crucial_Position;<br>end if | cOp1_Mem_PM_Position | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.4566 | Def | d51in_EST_EfeS_State | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.4567 | Def | D10in_PM1_Crucial_Position | The port D10in_PM1_Crucial_Position represents the Point position that was observed at the interface of the first crucial Point machine. The Point position is derived from the InformationFlow Information_No_End_Position, Information_End_Position_Reached, Information_End_Position_Detected and Information_Unintended_Position. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.4568 | Def | D10in_PMi_Crucial_Position | The port D10in_PMi_Crucial_Position represents the Point position that was observed at the interface of the n-th crucial Point machine. The Point position is derived from the InformationFlow Information_No_End_Position and Information_Unintended_Position. | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.4586 | Def | d14out_Observed_Point_Position | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.4979 | Def | D26in_Con_PMi_Crucial_Activation | The port D26in_Con_PMi_Crucial_Activation provides configuration values for the Point machine PM1n crucial and representing a n-th crucial Point machine in configurations with more than one Point machine.<br><br>true: Point machine PM1n Crucial is activated<br>false: Point machine PM1n Crucial is not activated | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6033 | Def | D10in_PM1_Non_Crucial_Position | The port D10in_PM1_Non_Crucial_Position represents the Point position that was observed at the interface of the first non-crucial Point machine. The Point position is derived from the InformationFlowInformation_No_End_Position, and Information_Unintended_Position. | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6034 | Def | D10in_PMk_Non_Crucial_Position | The port D10in_PMk_Non_Crucial_Position represents the Point position that was observed at the interface of the n-th non-crucial Point machine. The Point position is derived from the InformationFlow Information_No_End_Position and Information_Unintended_Position. | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6035 | Def | D27in_Con_PM1_Non_Crucial_Activation | The port D27in_Con_PM1_Non_Crucial_Activation provides configuration values for the Point machine PM2 non crucial, representing the 1st non crucial Point machine in configurations with more than one Point machine.<br><br>true: Point machine PM2 Non Crucial is activated<br>false: Point machine PM2 Non Crucial is not activated | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6036 | Def | D28in_Con_PMk_Non_Crucial_Activation | The port D28in_Con_PMk_Non_Crucial_Activation provides configuration values for the Point machine PM2n non crucial, representing a n-th non crucial Point machine in configurations with more than one Point machine.<br><br>true: Point machine PM2n Non Crucial is activated<br>false: Point machine PM2n Non Crucial is not activated | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6037 | Def | D29in_Con_Use_Unintended_Position | The port D29in_Con_Use_Unintended_Position provides a configuration value to the Subsystem - Point enabling interpretation of Unintended position.<br><br>true: Point is configured for interpretation of Unintended position<br>false: Point is not configured for interpretation of Unintended position | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.4587 | Info | F_Observe_Overall_Point_Position - Behaviour | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

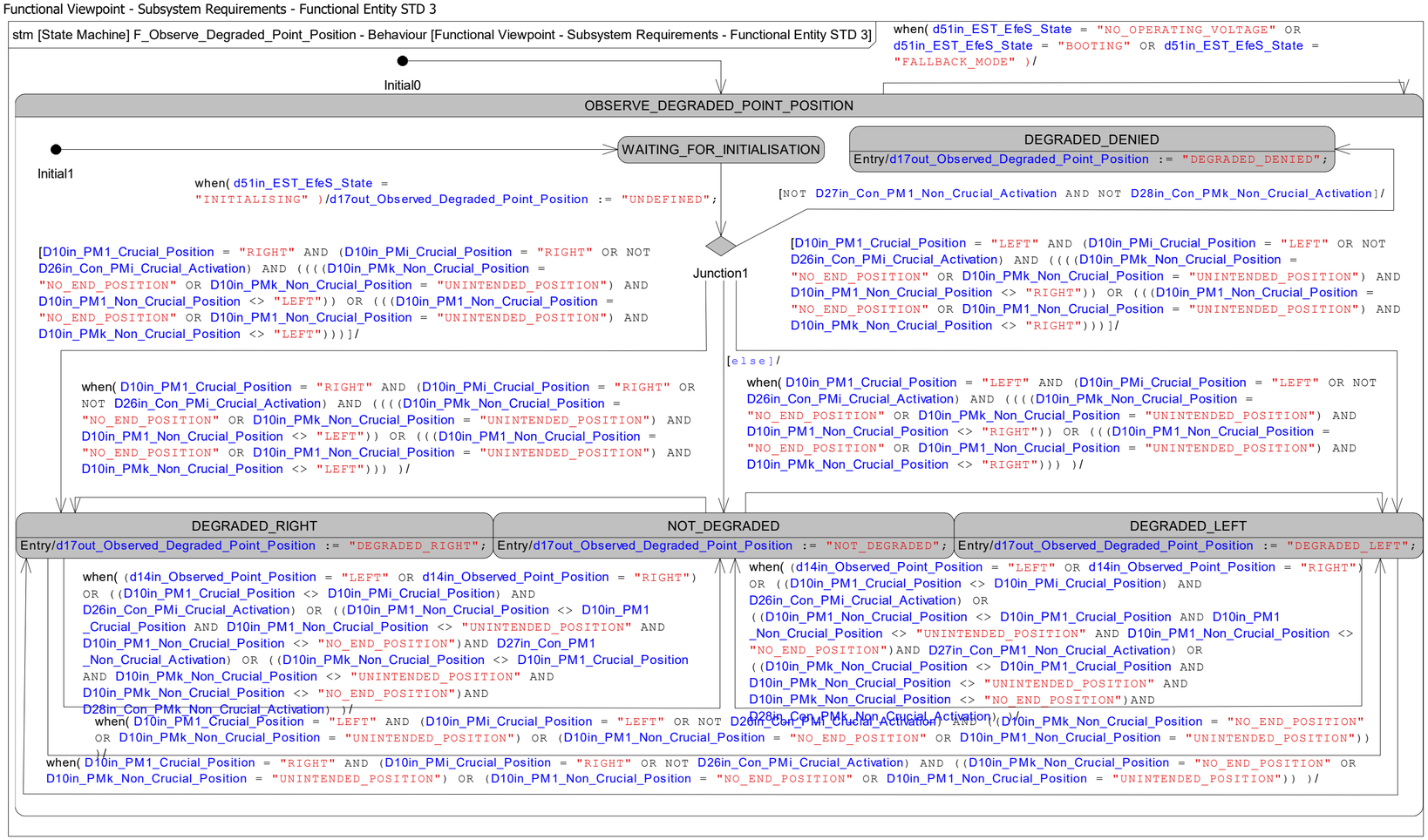
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.4588 | Req | Functional Viewpoint - Subsystem Requirements - Functional Entity STD 2<br>  <br> | This state machine diagram describes the requirements for the following functionalities:<br><br>- observes and forward the overall point position state | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6241 | Def | Initial0 | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6242 | Def | /Mem_PM1_Crucial_Position := "UNDEFINED";<br>Mem_PMi_Crucial_Position := "UNDEFINED";<br>Mem_PM1_Non_Crucial_Position := "UNDEFINED";<br>Mem_PMk_Non_Crucial_Position := "UNDEFINED";{Initial0 - OBSERVE_OVERALL_POINT_POSITION} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6310 | Def | OBSERVE_OVERALL_POINT_POSITION | | Basic non-4-wire single P<br>Basic non-4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| | | | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6311 | Def | Initial1 | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6312 | Def | /{Initial1 - Junction} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6313 | Def | Junction | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6314 | Def | [D10in_PM1_Crucial_Position = "LEFT" AND (D10in_PMi_Crucial_Position = "LEFT" OR NOT D26in_Con_PMi_Crucial_Activation) AND (D10in_PM1_Non_Crucial_Position = "LEFT"<br>OR NOT D27in_Con_PM1_Non_Crucial_Activation) AND (D10in_PMk_Non_Crucial_Position = "LEFT" OR NOT D28in_Con_PMk_Non_Crucial_Activation)]/d14out_Observed_Point_Position := "LEFT";{Junction - LEFT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6315 | Def | [else]/<br>d14out_Observed_Point_Position := "NO_END_POSITION";{Junction - NO_END_POSITION} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6316 | Def | [D10in_PM1_Crucial_Position = "RIGHT" AND (D10in_PMi_Crucial_Position = "RIGHT" OR NOT D26in_Con_PMi_Crucial_Activation) AND (D10in_PM1_Non_Crucial_Position = "RIGHT"<br>OR NOT D27in_Con_PM1_Non_Crucial_Activation) AND (D10in_PMk_Non_Crucial_Position = "RIGHT" OR NOT D28in_Con_PMk_Non_Crucial_Activation)]/d14out_Observed_Point_Position := "RIGHT";{Junction - RIGHT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6317 | Def | [(D10in_PM1_Crucial_Position = "UNINTENDED_POSITION" OR D10in_PMi_Crucial_Position = "UNINTENDED_POSITION" OR D10in_PM1_Non_Crucial_Position = "UNINTENDED_POSITION"<br>OR D10in_PMk_Non_Crucial_Position = "UNINTENDED_POSITION") AND D29in_Con_Use_Unintended_Position]/d14out_Observed_Point_Position := "UNINTENDED_POSITION";{Junction - UNINTENDED_POSITION} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6318 | Def | LEFT | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6408 | Def | entry/<br>Mem_PM1_Crucial_Position := D10in_PM1_Crucial_Position;<br>cOp1_Mem_PM_Position();{State-internal in LEFT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire |

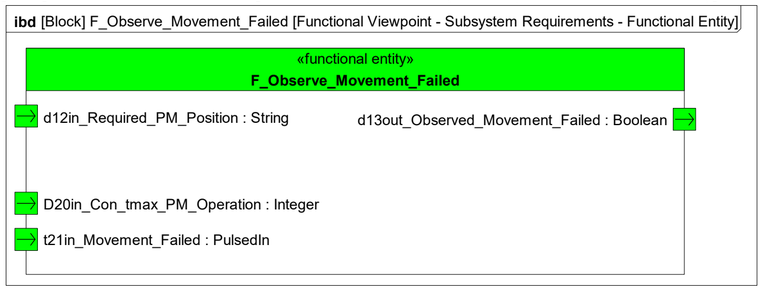
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| | | | | Basic 4-wire multiple P |
| Eu.P.6319 | Def | NO_END_POSITION | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6409 | Def | entry/<br>Mem_PM1_Crucial_Position := D10in_PM1_Crucial_Position;<br>cOp1_Mem_PM_Position();{State-internal in NO_END_POSITION} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6320 | Def | RIGHT | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6410 | Def | entry/<br>Mem_PM1_Crucial_Position := D10in_PM1_Crucial_Position;<br>cOp1_Mem_PM_Position();{State-internal in RIGHT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6321 | Def | UNINTENDED_POSITION | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6411 | Def | entry/<br>Mem_PM1_Crucial_Position := D10in_PM1_Crucial_Position;<br>cOp1_Mem_PM_Position();{State-internal in UNINTENDED_POSITION} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6322 | Def | when(D10in_PM1_Crucial_Position <> Mem_PM1_Crucial_Position)/{OBSERVE_OVERALL_POINT_POSITION - OBSERVE_OVERALL_POINT_POSITION} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6323 | Def | when(D10in_PMi_Crucial_Position <> Mem_PMi_Crucial_Position)/{OBSERVE_OVERALL_POINT_POSITION - OBSERVE_OVERALL_POINT_POSITION} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

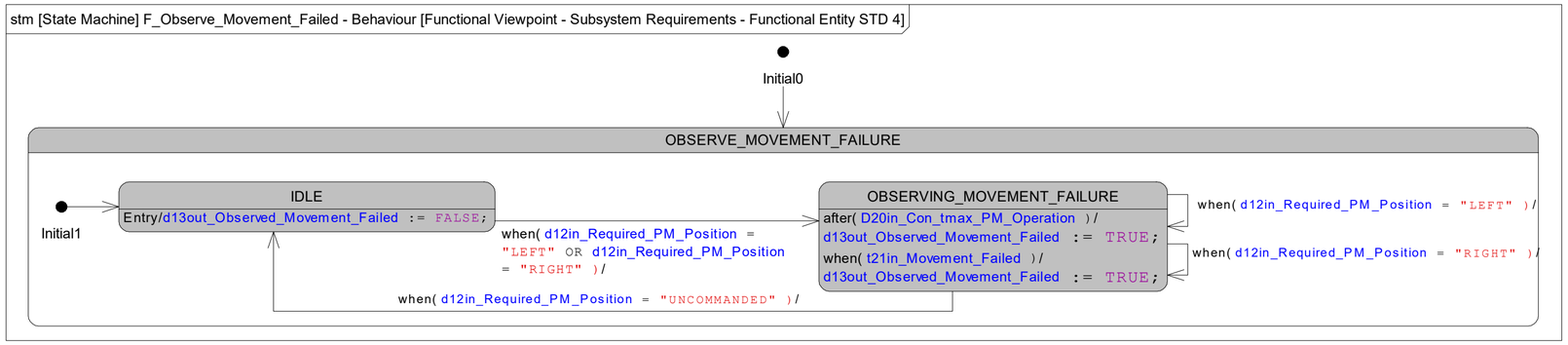
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6324 | Def | when(D10in_PM1_Non_Crucial_Position <> Mem_PM1_Non_Crucial_Position)/{OBSERVE_OVERALL_POINT_POSITION - OBSERVE_OVERALL_POINT_POSITION} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6325 | Def | when(D10in_PMk_Non_Crucial_Position <> Mem_PMk_Non_Crucial_Position)/{OBSERVE_OVERALL_POINT_POSITION - OBSERVE_OVERALL_POINT_POSITION} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6326 | Def | when(d51in_EST_EfeS_State = "NO_OPERATING_VOLTAGE" OR d51in_EST_EfeS_State = "BOOTING" OR d51in_EST_EfeS_State = "FALLBACK_MODE")/{OBSERVE_OVERALL_POINT_POSITION - OBSERVE_OVERALL_POINT_POSITION} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6002 | Info | F_Observe_Degraded_Point_Position | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6003 | Req | [Block] F_Observe_Degraded_Point_Position [Functional Viewpoint - Subsystem Requirements - Functional Entity]<br>  <br> | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6004 | Def | D10in_PM1_Crucial_Position | The port D10in_PM1_Crucial_Position represents the Point position that was observed at the interface of the first crucial Point machine. The Point position is derived from the InformationFlow Information_No_End_Position, Information_End_Position_Reached, Information_End_Position_Detected and Information_Unintended_Position. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6005 | Def | D10in_PMi_Crucial_Position | The port D10in_PMi_Crucial_Position represents the Point position that was observed at the interface of the n-th crucial Point machine. The Point position is derived from the InformationFlow Information_No_End_Position and Information_Unintended_Position. | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6006 | Def | D10in_PM1_Non_Crucial_Position | The port D10in_PM1_Non_Crucial_Position represents the Point position that was observed at the interface of the first non-crucial Point machine. The Point position is derived from the InformationFlow Information_No_End_Position and Information_Unintended_Position. | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6007 | Def | D10in_PMk_Non_Crucial_Position | The port D10in_PMk_Non_Crucial_Position represents the Point position that was observed at the interface of the n-th non-crucial Point machine. The Point position is derived from the InformationFlow Information_No_End_Position and Information_Unintended_Position. | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6008 | Def | d17out_Observed_Degraded_Point_Position | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6009 | Def | d51in_EST_EfeS_State | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6010 | Def | D26in_Con_PMi_Crucial_Activation | The port D26in_Con_PMi_Crucial_Activation provides configuration values for the Point machine PM1n crucial and representing a n-th crucial Point machine in configurations with more than one Point machine.<br><br>true: Point machine PM1n Crucial is activated<br>false: Point machine PM1n Crucial is not activated | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6011 | Def | D27in_Con_PM1_Non_Crucial_Activation | The port D27in_Con_PM1_Non_Crucial_Activation provides configuration values for the Point machine PM2 non crucial, representing the 1st non crucial Point machine in configurations with more than one Point machine.<br><br>true: Point machine PM2 Crucial is activated<br>false: Point machine PM2 Crucial is not activated | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6012 | Def | D28in_Con_PMk_Non_Crucial_Activation | The port D28in_Con_PMk_Non_Crucial_Activation provides configuration values for the Point machine PM2n non crucial, representing a n-th non crucial Point machine in configurations with more than one Point machine.<br><br>true: Point machine PM2n Non Crucial is activated<br>false: Point machine PM2n Non Crucial is not activated | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6013 | Def | d14in_Observed_Point_Position | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6014 | Info | F_Observe_Degraded_Point_Position - Behaviour | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |

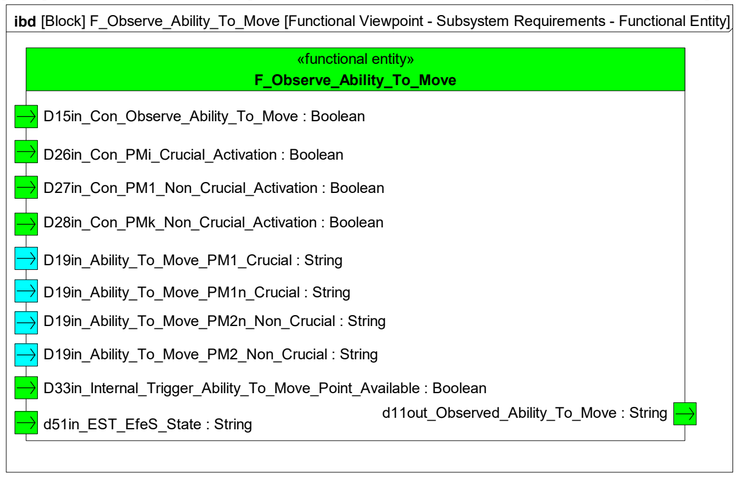
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6015 | Req | Functional Viewpoint - Subsystem Requirements - Functional Entity STD 3<br>  <br> | This state machine diagram describes the requirements for the following functionalities:<br><br>- observes and forward the degraded point position state | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6230 | Def | Initial0 | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6231 | Def | /{Initial0 - OBSERVE_DEGRADED_POINT_POSITION} | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6232 | Def | OBSERVE_DEGRADED_POINT_POSITION | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6016 | Def | DEGRADED_LEFT | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6017 | Def | entry/d17out_Observed_Degraded_Point_Position := "DEGRADED_LEFT";{State-internal in DEGRADED_LEFT} | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6018 | Def | when((d14in_Observed_Point_Position = "LEFT" OR d14in_Observed_Point_Position = "RIGHT") OR ((D10in_PM1_Crucial_Position <> D10in_PMi_Crucial_Position) AND D26in_Con_PMi_Crucial_Activation) OR ((D10in_PM1_Non_Crucial_Position <> D10in_PM1_Crucial_Position AND D10in_PM1_Non_Crucial_Position <> "UNINTENDED_POSITION" AND D10in_PM1_Non_Crucial_Position <> "NO_END_POSITION")AND D27in_Con_PM1_Non_Crucial_Activation) OR ((D10in_PMk_Non_Crucial_Position <> D10in_PM1_Crucial_Position AND D10in_PMk_Non_Crucial_Position <> "UNINTENDED_POSITION" AND D10in_PMk_Non_Crucial_Position <> "NO_END_POSITION")AND D28in_Con_PMk_Non_Crucial_Activation))/{DEGRADED_LEFT - NOT_DEGRADED} | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6019 | Def | when(D10in_PM1_Crucial_Position = "RIGHT" AND (D10in_PMi_Crucial_Position = "RIGHT" OR NOT D26in_Con_PMi_Crucial_Activation) AND ((D10in_PMk_Non_Crucial_Position = "NO_END_POSITION" OR D10in_PMk_Non_Crucial_Position = "UNINTENDED_POSITION") OR (D10in_PM1_Non_Crucial_Position = "NO_END_POSITION" OR D10in_PM1_Non_Crucial_Position = "UNINTENDED_POSITION")))/{DEGRADED_LEFT - DEGRADED_RIGHT} | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6020 | Def | DEGRADED_RIGHT | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6021 | Def | entry/d17out_Observed_Degraded_Point_Position := "DEGRADED_RIGHT";{State-internal in DEGRADED_RIGHT} | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6022 | Def | when((d14in_Observed_Point_Position = "LEFT" OR d14in_Observed_Point_Position = "RIGHT") OR ((D10in_PM1_Crucial_Position <> D10in_PMi_Crucial_Position) AND D26in_Con_PMi_Crucial_Activation) OR ((D10in_PM1_Non_Crucial_Position <> D10in_PM1_Crucial_Position AND D10in_PM1_Non_Crucial_Position <> "UNINTENDED_POSITION" AND D10in_PM1_Non_Crucial_Position <> "NO_END_POSITION")AND D27in_Con_PM1_Non_Crucial_Activation) OR ((D10in_PMk_Non_Crucial_Position <> D10in_PM1_Crucial_Position AND D10in_PMk_Non_Crucial_Position <> "UNINTENDED_POSITION" AND D10in_PMk_Non_Crucial_Position <> "NO_END_POSITION")AND D28in_Con_PMk_Non_Crucial_Activation))/{DEGRADED_RIGHT - NOT_DEGRADED} | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6023 | Def | when(D10in_PM1_Crucial_Position = "LEFT" AND (D10in_PMi_Crucial_Position = "LEFT" OR NOT D26in_Con_PMi_Crucial_Activation) AND ((D10in_PMk_Non_Crucial_Position = "NO_END_POSITION" OR D10in_PMk_Non_Crucial_Position = "UNINTENDED_POSITION") OR (D10in_PM1_Non_Crucial_Position = "NO_END_POSITION" OR D10in_PM1_Non_Crucial_Position = "UNINTENDED_POSITION")))/{DEGRADED_RIGHT - DEGRADED_LEFT} | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6024 | Def | Initial1 | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6025 | Def | /{Initial1 - WAITING_FOR_INITIALISATION} | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6233 | Def | Junction1 | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6234 | Def | [D10in_PM1_Crucial_Position = "LEFT" AND (D10in_PMi_Crucial_Position = "LEFT" OR NOT D26in_Con_PMi_Crucial_Activation) AND ((((D10in_PMk_Non_Crucial_Position = "NO_END_POSITION" OR D10in_PMk_Non_Crucial_Position = "UNINTENDED_POSITION") AND D10in_PM1_Non_Crucial_Position <> "RIGHT")) OR (((D10in_PM1_Non_Crucial_Position = "NO_END_POSITION" OR D10in_PM1_Non_Crucial_Position = "UNINTENDED_POSITION") AND D10in_PMk_Non_Crucial_Position <> "RIGHT")))]/{Junction1 - DEGRADED_LEFT} | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6235 | Def | [D10in_PM1_Crucial_Position = "RIGHT" AND (D10in_PMi_Crucial_Position = "RIGHT" OR NOT D26in_Con_PMi_Crucial_Activation) AND ((((D10in_PMk_Non_Crucial_Position = "NO_END_POSITION" OR D10in_PMk_Non_Crucial_Position = "UNINTENDED_POSITION") AND D10in_PM1_Non_Crucial_Position <> "LEFT")) OR (((D10in_PM1_Non_Crucial_Position = "NO_END_POSITION" OR D10in_PM1_Non_Crucial_Position = "UNINTENDED_POSITION") AND D10in_PMk_Non_Crucial_Position <> "LEFT")))]/{Junction1 - DEGRADED_RIGHT} | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6236 | Def | [else]/{Junction1 - NOT_DEGRADED} | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6237 | Def | [NOT D27in_Con_PM1_Non_Crucial_Activation AND NOT D28in_Con_PMk_Non_Crucial_Activation]/{Junction1 - DEGRADED_DENIED} | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6026 | Def | NOT_DEGRADED | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6027 | Def | entry/d17out_Observed_Degraded_Point_Position := "NOT_DEGRADED";{State-internal in NOT_DEGRADED} | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |

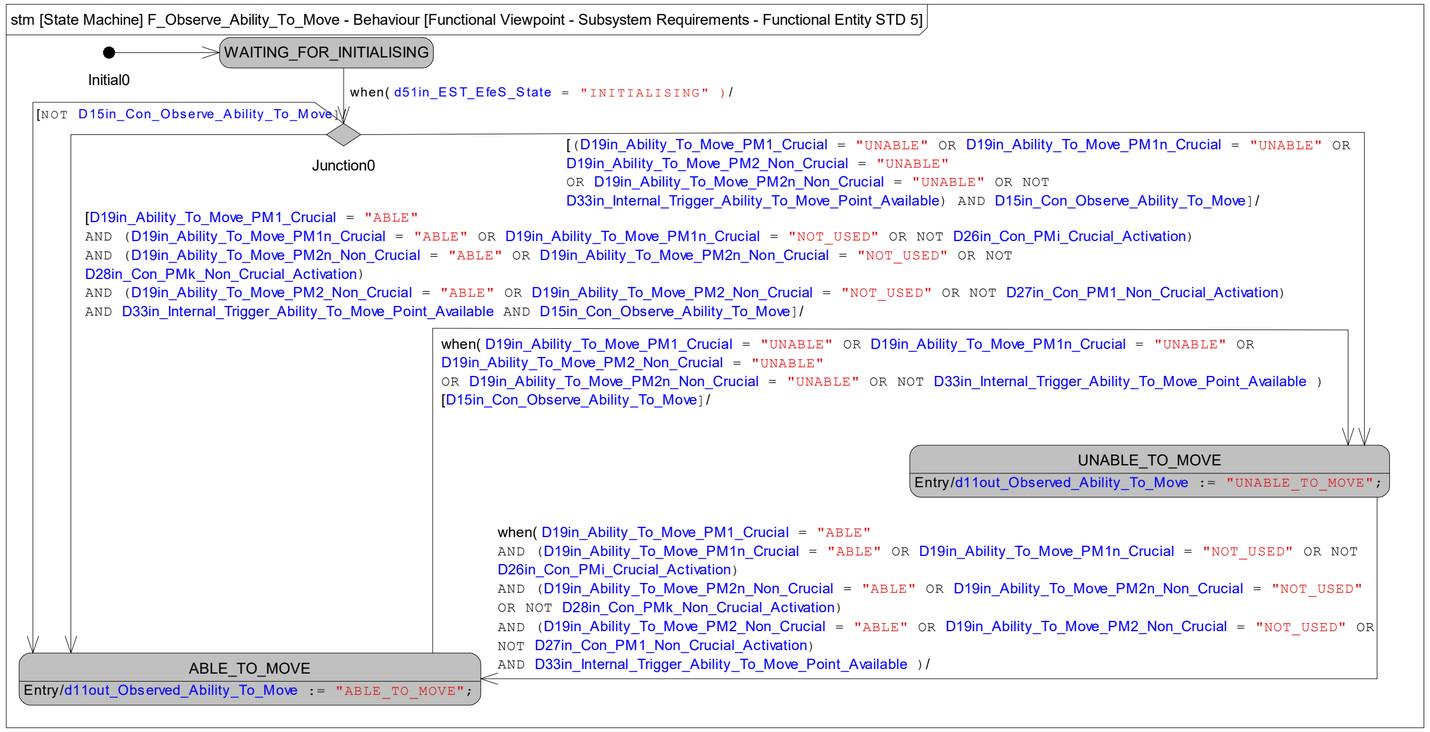
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6028 | Def | when(D10in_PM1_Crucial_Position = "LEFT" AND (D10in_PMi_Crucial_Position = "LEFT" OR NOT D26in_Con_PMi_Crucial_Activation) AND ((((D10in_PMk_Non_Crucial_Position = "NO_END_POSITION" OR D10in_PMk_Non_Crucial_Position = "UNINTENDED_POSITION") AND D10in_PM1_Non_Crucial_Position <> "RIGHT")) OR (((D10in_PM1_Non_Crucial_Position = "NO_END_POSITION" OR D10in_PM1_Non_Crucial_Position = "UNINTENDED_POSITION") AND D10in_PMk_Non_Crucial_Position <> "RIGHT"))))/{NOT_DEGRADED - DEGRADED_LEFT} | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6029 | Def | when(D10in_PM1_Crucial_Position = "RIGHT" AND (D10in_PMi_Crucial_Position = "RIGHT" OR NOT D26in_Con_PMi_Crucial_Activation) AND ((((D10in_PMk_Non_Crucial_Position = "NO_END_POSITION" OR D10in_PMk_Non_Crucial_Position = "UNINTENDED_POSITION") AND D10in_PM1_Non_Crucial_Position <> "LEFT")) OR (((D10in_PM1_Non_Crucial_Position = "NO_END_POSITION" OR D10in_PM1_Non_Crucial_Position = "UNINTENDED_POSITION") AND D10in_PMk_Non_Crucial_Position <> "LEFT"))))/{NOT_DEGRADED - DEGRADED_RIGHT} | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6238 | Def | WAITING_FOR_INITIALISATION | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6239 | Def | when(d51in_EST_EfeS_State = "INITIALISING")/d17out_Observed_Degraded_Point_Position := "UNDEFINED";{WAITING_FOR_INITIALISATION - Junction1} | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6240 | Def | when(d51in_EST_EfeS_State = "NO_OPERATING_VOLTAGE" OR d51in_EST_EfeS_State = "BOOTING" OR d51in_EST_EfeS_State = "FALLBACK_MODE")/{OBSERVE_DEGRADED_POINT_POSITION - OBSERVE_DEGRADED_POINT_POSITION} | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6406 | Def | DEGRADED_DENIED | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6407 | Def | entry/d17out_Observed_Degraded_Point_Position := "DEGRADED_DENIED";{State-internal in DEGRADED_DENIED} | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6993 | Info | F_Observe_Movement_Failed | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6994 | Req | [Block] F_Observe_Movement_Failed [Functional Viewpoint - Subsystem Requirements - Functional Entity]<br>  <br> | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6995 | Def | d12in_Required_PM_Position | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6996 | Def | d13out_Observed_Movement_Failed | | Basic non-4-wire single P<br>Basic non-4-wire multiple |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| | | | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6998 | Def | D20in_Con_tmax_PM_Operation | The port D20in_Con_tmax_PM_Operation refines the time value Con_tmax_Point_Operation. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7015 | Def | t21in_Movement_Failed | The port t21in_Movement_Failed represents all internal occurrences of conditions to the Subsystem - Point | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6999 | Info | F_Observe_Movement_Failed - Behaviour | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7000 | Req | Functional Viewpoint - Subsystem Requirements - Functional Entity STD 4<br>  <br> | This state machine diagram describes the requirements for the following functionalities:<br><br>- monitors the Con_tmax_PM_Operation and forwards an overrun as a Movent Failure<br>- observes and forward a Movement failure | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7001 | Def | Initial0 | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7002 | Def | /{Initial0 - OBSERVE_MOVEMENT_FAILURE} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

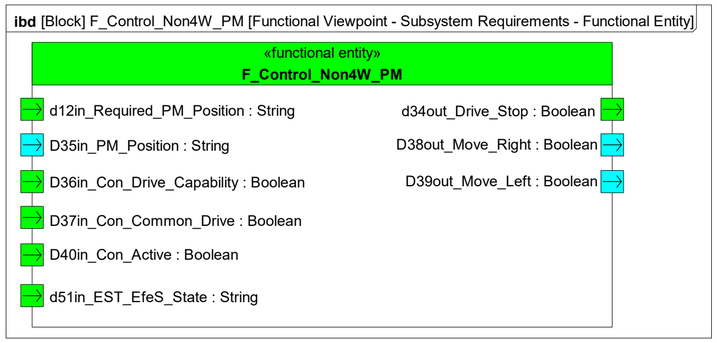
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.7003 | Def | OBSERVE_MOVEMENT_FAILURE | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7004 | Def | IDLE | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7005 | Def | entry/d13out_Observed_Movement_Failed := FALSE;{State-internal in IDLE} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7006 | Def | when(d12in_Required_PM_Position = "LEFT" OR d12in_Required_PM_Position = "RIGHT")/{IDLE - OBSERVING_MOVEMENT_FAILURE} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7007 | Def | Initial1 | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7008 | Def | /{Initial1 - IDLE} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7009 | Def | OBSERVING_MOVEMENT_FAILURE | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7010 | Def | after(D20in_Con_tmax_PM_Operation)/<br>d13out_Observed_Movement_Failed := TRUE;{State-internal in OBSERVING_MOVEMENT_FAILURE} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7011 | Def | when(t21in_Movement_Failed)/<br>d13out_Observed_Movement_Failed := TRUE;{State-internal in OBSERVING_MOVEMENT_FAILURE} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.7012 | Def | when(d12in_Required_PM_Position = "LEFT")/{OBSERVING_MOVEMENT_FAILURE - OBSERVING_MOVEMENT_FAILURE} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7013 | Def | when(d12in_Required_PM_Position = "RIGHT")/{OBSERVING_MOVEMENT_FAILURE - OBSERVING_MOVEMENT_FAILURE} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7014 | Def | when(d12in_Required_PM_Position = "UNCOMMANDED")/{OBSERVING_MOVEMENT_FAILURE - IDLE} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5980 | Info | F_Observe_Ability_To_Move | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5981 | Req | [Block] F_Observe_Ability_To_Move [Functional Viewpoint - Subsystem Requirements - Functional Entity]<br>  <br> | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5982 | Def | D19in_Ability_To_Move_PM1_Crucial | The port D19in_Ability_To_Move_PM1_Crucial refines the InformationFlow Information_Ability_To_Move_Point from the first Point machine to Subsystem - Point.<br>Note - Drive voltage levels monitored shall be defined by national specifications. In future phases of the System Pillar, national specifications will be replaced by harmonised specifications. | Option Able to move |

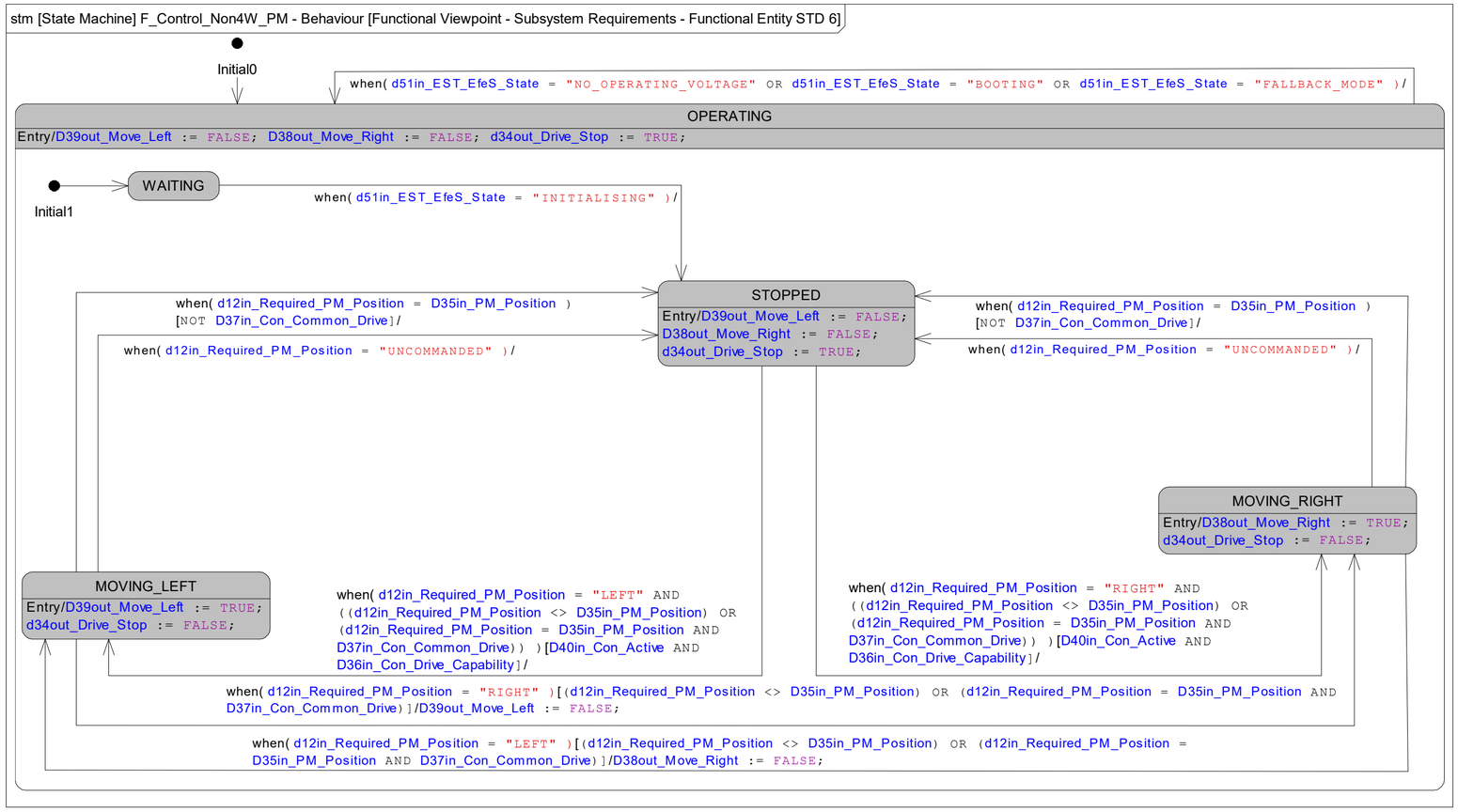
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.5983 | Def | D19in_Ability_To_Move_PM1n_Crucial | The port D19in_Ability_To_Move_PM1n_Crucial refines the InformationFlow Information_Ability_To_Move_Point from the n-th Point machine to Subsystem - Point.<br>Note - Drive voltage levels monitored shall be defined by national specifications. In future phases of the System Pillar, national specifications will be replaced by harmonised specifications.<br><br>Only applicable if the package [Option Able to move] is used in combination with a [Basic non-4-wire multiple P] or [Basic 4-wire multiple P]. | Option Able to move |
| Eu.P.5984 | Def | D19in_Ability_To_Move_PM2_Non_Crucial | The port D19in_Ability_To_Move_PM2 _Non_Crucial refines the InformationFlow Information_Ability_To_Move_Point from the n-th Point machine to Subsystem - Point.<br>Note - Drive voltage levels monitored shall be defined by national specifications. In future phases of the System Pillar, national specifications will be replaced by harmonised specifications.<br><br>Only applicable if the package [Option Able to move] is used in combination with a [Basic non-4-wire multiple P] or [Basic 4-wire multiple P]. | Option Able to move |
| Eu.P.5985 | Def | D19in_Ability_To_Move_PM2n_Non_Crucial | The port D19in_Ability_To_Move_PM2n_Non_Crucial refines the InformationFlow Information_Ability_To_Move_Point from the n-th Point machine to Subsystem - Point.<br>Note - Drive voltage levels monitored shall be defined by national specifications. In future phases of the System Pillar, national specifications will be replaced by harmonised specifications.<br><br>Only applicable if the package [Option Able to move] is used in combination with a [Basic non-4-wire multiple P] or [Basic 4-wire multiple P]. | Option Able to move |
| Eu.P.5986 | Def | D33in_Internal_Trigger_Ability_To_Move_Point_Available | The port D33in_Internal_Trigger_Ability_To_Move_Point_Available represents all internal occurrences of conditions to the Subsystem - Point, leading to unability to move. | Option Able to move |
| Eu.P.5987 | Def | d11out_Observed_Ability_To_Move | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5988 | Def | d51in_EST_EfeS_State | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6224 | Def | D26in_Con_PMi_Crucial_Activation | The port D26in_Con_PMi_Crucial_Activation provides configuration values for the Point machine PM1n crucial and representing a n-th crucial Point machine in configurations with more than one Point machine.<br><br>true: Point machine PM1n Crucial is activated<br>false: Point machine PM1n Crucial is not activated | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6225 | Def | D27in_Con_PM1_Non_Crucial_Activation | The port D27in_Con_PM1_Non_Crucial_Activation provides configuration values for the Point machine PM2 non crucial, representing the 1st non crucial Point machine in configurations with more than one Point machine.<br><br>true: Point machine PM2 Non Crucial is activated<br>false: Point machine PM2 Non Crucial is not activated | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6226 | Def | D28in_Con_PMk_Non_Crucial_Activation | The port D28in_Con_PMk_Non_Crucial_Activation provides configuration values for the Point machine PM2n non crucial, representing a n-th non crucial Point machine in configurations with more than one Point machine. | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| | | | true: Point machine PM2n Non Crucial is activated<br>false: Point machine PM2n Non Crucial is not activated | |
| Eu.P.6991 | Def | D15in_Con_Observe_Ability_To_Move | | Option Able to move |
| Eu.P.5992 | Info | F_Observe_Ability_To_Move - Behaviour | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5993 | Req | Functional Viewpoint - Subsystem Requirements - Functional Entity STD 5<br>  <br> | This state machine diagram describes the requirements for the following functionalities:<br><br>- observes and forward the Ability to move state | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6227 | Def | Initial0 | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6228 | Def | /{Initial0 - WAITING_FOR_INITIALISING} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6229 | Def | WAITING_FOR_INITIALISING | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6405 | Def | when(d51in_EST_EfeS_State = "INITIALISING")/{WAITING_FOR_INITIALISING - Junction0} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6396 | Def | ABLE_TO_MOVE | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6397 | Def | entry/d11out_Observed_Ability_To_Move := "ABLE_TO_MOVE";{State-internal in ABLE_TO_MOVE} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6398 | Def | when(D19in_Ability_To_Move_PM1_Crucial = "UNABLE" OR D19in_Ability_To_Move_PM1n_Crucial = "UNABLE" OR D19in_Ability_To_Move_PM2_Non_Crucial = "UNABLE"<br>OR D19in_Ability_To_Move_PM2n_Non_Crucial = "UNABLE" OR NOT D33in_Internal_Trigger_Ability_To_Move_Point_Available)[D15in_Con_Observe_Ability_To_Move]/{ABLE_TO_MOVE - UNABLE_TO_MOVE} | | Option Able to move |
| Eu.P.6399 | Def | Junction0 | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6400 | Def | [D19in_Ability_To_Move_PM1_Crucial = "ABLE"<br>AND (D19in_Ability_To_Move_PM1n_Crucial = "ABLE" OR D19in_Ability_To_Move_PM1n_Crucial = "NOT_USED" OR NOT D26in_Con_PMi_Crucial_Activation)<br>AND (D19in_Ability_To_Move_PM2n_Non_Crucial = "ABLE" OR D19in_Ability_To_Move_PM2n_Non_Crucial = "NOT_USED" OR NOT D28in_Con_PMk_Non_Crucial_Activation)<br>AND (D19in_Ability_To_Move_PM2_Non_Crucial = "ABLE" OR D19in_Ability_To_Move_PM2_Non_Crucial = "NOT_USED" OR NOT D27in_Con_PM1_Non_Crucial_Activation)<br>AND D33in_Internal_Trigger_Ability_To_Move_Point_Available AND D15in_Con_Observe_Ability_To_Move]/{Junction0 - ABLE_TO_MOVE} | | Option Able to move |
| Eu.P.6401 | Def | [(D19in_Ability_To_Move_PM1_Crucial = "UNABLE" OR D19in_Ability_To_Move_PM1n_Crucial = "UNABLE" OR D19in_Ability_To_Move_PM2_Non_Crucial = "UNABLE"<br>OR D19in_Ability_To_Move_PM2n_Non_Crucial = "UNABLE" OR NOT D33in_Internal_Trigger_Ability_To_Move_Point_Available) AND D15in_Con_Observe_Ability_To_Move]/{Junction0 - UNABLE_TO_MOVE} | | Option Able to move |
| Eu.P.6992 | Def | [NOT D15in_Con_Observe_Ability_To_Move]/{Junction0 - ABLE_TO_MOVE} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6402 | Def | UNABLE_TO_MOVE | | Option Able to move |
| Eu.P.6403 | Def | entry/d11out_Observed_Ability_To_Move := "UNABLE_TO_MOVE";{State-internal in UNABLE_TO_MOVE} | | Option Able to move |
| Eu.P.6404 | Def | when(D19in_Ability_To_Move_PM1_Crucial = "ABLE"<br>AND (D19in_Ability_To_Move_PM1n_Crucial = "ABLE" OR D19in_Ability_To_Move_PM1n_Crucial = "NOT_USED" OR NOT D26in_Con_PMi_Crucial_Activation)<br>AND (D19in_Ability_To_Move_PM2n_Non_Crucial = "ABLE" OR D19in_Ability_To_Move_PM2n_Non_Crucial = "NOT_USED" OR NOT D28in_Con_PMk_Non_Crucial_Activation)<br>AND (D19in_Ability_To_Move_PM2_Non_Crucial = "ABLE" OR D19in_Ability_To_Move_PM2_Non_Crucial = "NOT_USED" OR NOT D27in_Con_PM1_Non_Crucial_Activation)<br>AND D33in_Internal_Trigger_Ability_To_Move_Point_Available)/{UNABLE_TO_MOVE - ABLE_TO_MOVE} | | Option Able to move |

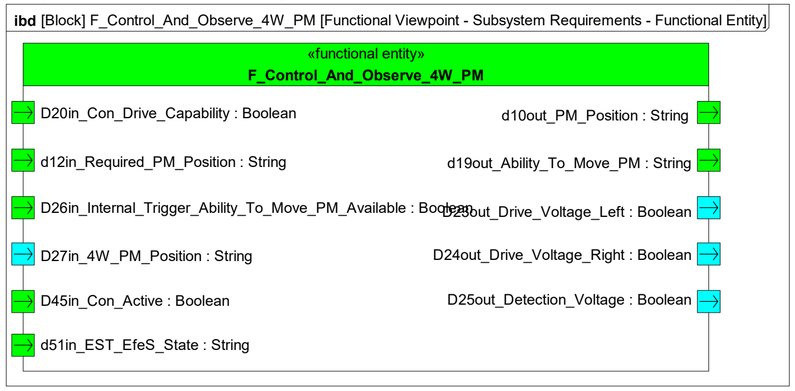
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.5925 | Info | F_Control_Non4W_PM | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5926 | Req | [Block] F_Control_Non4W_PM [Functional Viewpoint - Subsystem Requirements - Functional Entity]<br>  <br> | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5927 | Def | D35in_PM_Position | The port D35in_PM_Position provides the Point machine position to the Subsystem - Point.<br>The port D35in_PM_Position refines the InformationFlow Information_No_End_Position and Information_Unintended_Position. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5928 | Def | D39out_Move_Left | The port D39out_Move_Left refines the Informationflow Moving and contains the information if driving left was started.<br>The following values are valid: "True" (driving left), "False" (not driving left) | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5929 | Def | D38out_Move_Right | The port D38out_Move_Right provides the command  Moving right to the point machine and refines the InformationFlow Moving and Stop moving.<br>The following values are valid: "True" (driving right), "False" (not driving right, Stop moving). | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5930 | Def | d12in_Required_PM_Position | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5932 | Def | d34out_Drive_Stop | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.5934 | Def | d51in_EST_EfeS_State | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5936 | Def | D37in_Con_Common_Drive | The port D37in_Con_Common_Drive provides a configuration value to the Point for Redrive point functionality.<br><br>true: Point has Common Drive functionality<br>false: Point has Individual Drive functionality | Option Common Drive |
| Eu.P.6222 | Def | D40in_Con_Active | The port D40in_Con_Active provides configuration values for the Point machine. The Point machine in single Point machine configurations and the first crucial Point machine in multiple Point machine configurations is always activated.<br><br>true: Point machine is activated<br>false: Point machine is not activated | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6223 | Def | D36in_Con_Drive_Capability | The port D36in_Con_Drive_Capability provides a configuration value for the Point machine telling the Subsystem - Point whether the Point machine is configured with full drive capability or as Point detector only.<br><br>true: full drive capability<br>false: Point detector only | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5938 | Info | F_Control_Non4W_PM - Behaviour | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

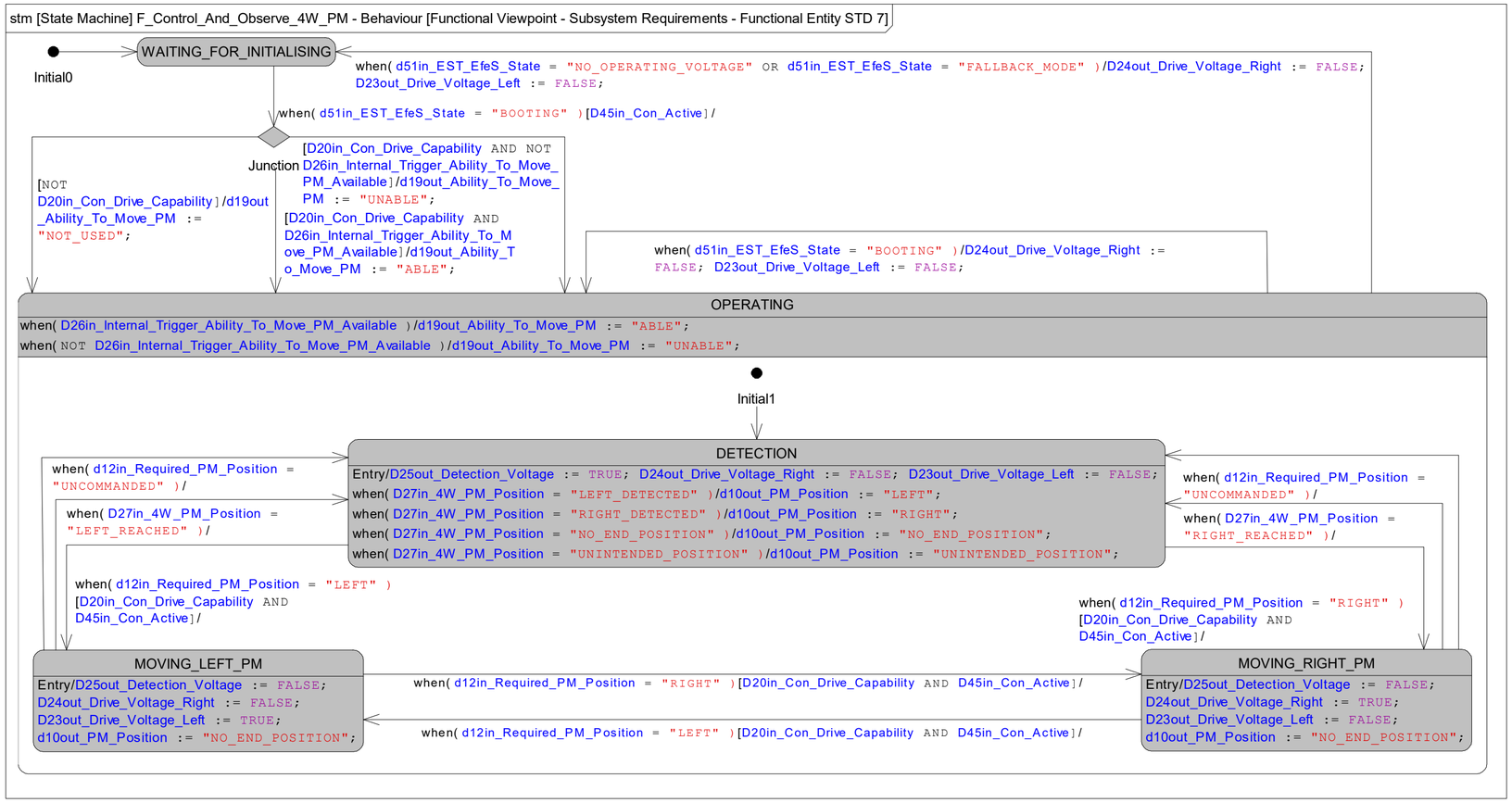
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.5939 | Req | Functional Viewpoint - Subsystem Requirements - Functional Entity STD 6<br>  <br> | This state machine diagram describes the requirements for the following functionalities:<br><br>- observes and forward the Moving state for an non 4W PM<br>- observes and forward the Drive stop state for an non 4W PM | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5940 | Def | Initial0 | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5941 | Def | /{Initial0 - OPERATING} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5942 | Def | OPERATING | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5947 | Def | Initial1 | | Basic non-4-wire single P<br>Basic non-4- |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| | | | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5948 | Def | /{Initial1 - WAITING} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5955 | Def | MOVING_LEFT | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5957 | Def | entry/D39out_Move_Left := TRUE;<br>d34out_Drive_Stop := FALSE;{State-internal in MOVING_LEFT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5960 | Def | when(d12in_Required_PM_Position = "RIGHT")[(d12in_Required_PM_Position <> D35in_PM_Position) OR (d12in_Required_PM_Position = D35in_PM_Position AND D37in_Con_Common_Drive)]/D39out_Move_Left := FALSE;{MOVING_LEFT - MOVING_RIGHT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5961 | Def | when(d12in_Required_PM_Position = "UNCOMMANDED")/{MOVING_LEFT - STOPPED} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6301 | Def | when(d12in_Required_PM_Position = D35in_PM_Position)[NOT D37in_Con_Common_Drive]/{MOVING_LEFT - STOPPED} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5965 | Def | MOVING_RIGHT | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5967 | Def | entry/D38out_Move_Right := TRUE;<br>d34out_Drive_Stop := FALSE;{State-internal in MOVING_RIGHT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5970 | Def | when(d12in_Required_PM_Position = "LEFT")[(d12in_Required_PM_Position <> D35in_PM_Position) OR (d12in_Required_PM_Position = D35in_PM_Position AND D37in_Con_Common_Drive)]/D38out_Move_Right := FALSE;{MOVING_RIGHT - MOVING_LEFT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire |

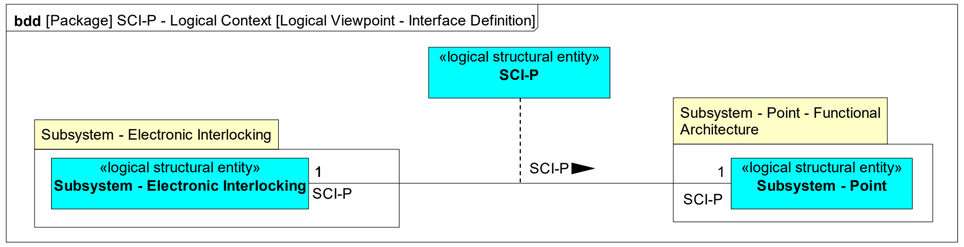
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| | | | | Basic 4-wire multiple P |
| Eu.P.5971 | Def | when(d12in_Required_PM_Position = "UNCOMMANDED")/{MOVING_RIGHT - STOPPED} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6303 | Def | when(d12in_Required_PM_Position = D35in_PM_Position)[NOT D37in_Con_Common_Drive]/{MOVING_RIGHT - STOPPED} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5975 | Def | STOPPED | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5976 | Def | when(d12in_Required_PM_Position = "LEFT" AND ((d12in_Required_PM_Position <> D35in_PM_Position) OR (d12in_Required_PM_Position = D35in_PM_Position AND D37in_Con_Common_Drive)))[D40in_Con_Active AND D36in_Con_Drive_Capability]/{STOPPED - MOVING_LEFT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5977 | Def | when(d12in_Required_PM_Position = "RIGHT" AND ((d12in_Required_PM_Position <> D35in_PM_Position) OR (d12in_Required_PM_Position = D35in_PM_Position AND D37in_Con_Common_Drive)))[D40in_Con_Active AND D36in_Con_Drive_Capability]/{STOPPED - MOVING_RIGHT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6304 | Def | entry/D39out_Move_Left := FALSE;<br>D38out_Move_Right := FALSE;<br>d34out_Drive_Stop := TRUE;{State-internal in STOPPED} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6392 | Def | entry/D39out_Move_Left := FALSE; D38out_Move_Right := FALSE; d34out_Drive_Stop := TRUE;{State-internal in OPERATING} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6393 | Def | WAITING | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6394 | Def | when(d51in_EST_EfeS_State = "INITIALISING")/{WAITING - STOPPED} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6395 | Def | when(d51in_EST_EfeS_State = "NO_OPERATING_VOLTAGE" OR d51in_EST_EfeS_State = "BOOTING" OR d51in_EST_EfeS_State = "FALLBACK_MODE")/{OPERATING - OPERATING} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5810 | Info | F_Control_And_Observe_4W_PM | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5811 | Req | [Block] F_Control_And_Observe_4W_PM [Functional Viewpoint - Subsystem Requirements - Functional Entity]<br>  <br> | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5812 | Def | D20in_Con_Drive_Capability | The port D20in_Con_Drive_Capability provides configuration values for the Point machine.<br><br>true: Point machine has drive capability<br>false: Point machine has no drive capability | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5813 | Def | d19out_Ability_To_Move_PM | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5814 | Def | d10out_PM_Position | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5817 | Def | D27in_4W_PM_Position | The port D27in_4W_PM_Position refines the Informationflow Information_No_End_Position, Information_End_Position_Reached, Information_End_Position_Detected and Information_Unintended_Position and is a representative for the position of an undefined number of Point machines, in case they are implemented with a 4-wire interface.<br><br>Note: The same 4-wire input pattern doesn't always represent the same Informationflow. The interpretation of all patterns is fully defined in Eu.P.6797. | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5818 | Def | D26in_Internal_Trigger_Ability_To_Move_PM_Available | The port D26in_Internal_Trigger_Ability_To_Move_PM_Available represents all internal occurrences of conditions concerning the Point machine, leading to unability to move.<br>true: Point machine is able to move<br>false: Point machine is unable to move | Basic 4-wire single P<br>Basic 4-wire multiple P |

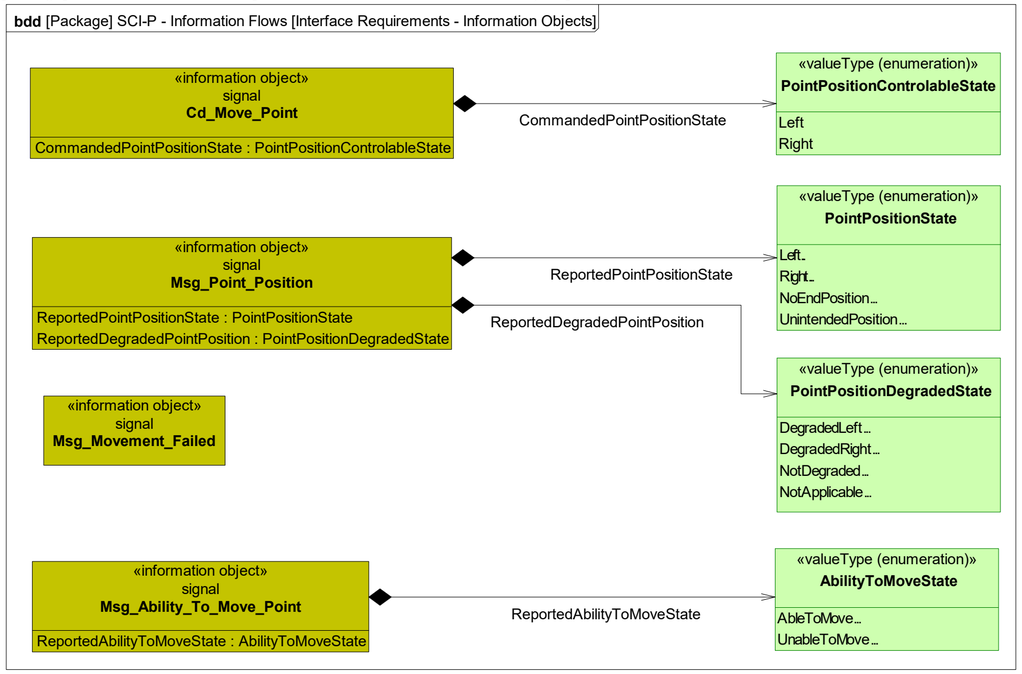
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.5819 | Def | D25out_Detection_Voltage | The port D25out_Detection_Voltage refines the Informationflow Detection_Voltage and contains the information if Detection voltage is provided to the Point machine via the 4-wire interface.<br>The following values are valid: "True" (detection), "False" (no detection) | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5820 | Def | D24out_Drive_Voltage_Right | The port D24out_Drive_Voltage_Right refines the Informationflow Moving and contains the information if Drive voltage for driving right is provided to the Point machine via the 4-wire interface.<br>The following values are valid: "True" (driving right), "False" (not driving right) | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5821 | Def | D23out_Drive_Voltage_Left | The port D23out_Drive_Voltage_Left refines the Informationflow Moving and contains the information if Drive voltage for driving left is provided to the Point machine via the 4-wire interface.<br>The following values are valid: "True" (driving left), "False" (not driving left) | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6209 | Def | d12in_Required_PM_Position | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6211 | Def | D45in_Con_Active | The port D45in_Con_Active provides configuration values for the Point machine.<br><br>true: Point machine is activated<br>false: Point machine is not activated | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6212 | Def | d51in_EST_EfeS_State | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5822 | Info | F_Control_And_Observe_4W_PM - Behaviour | | Basic 4-wire single P<br>Basic 4-wire multiple P |

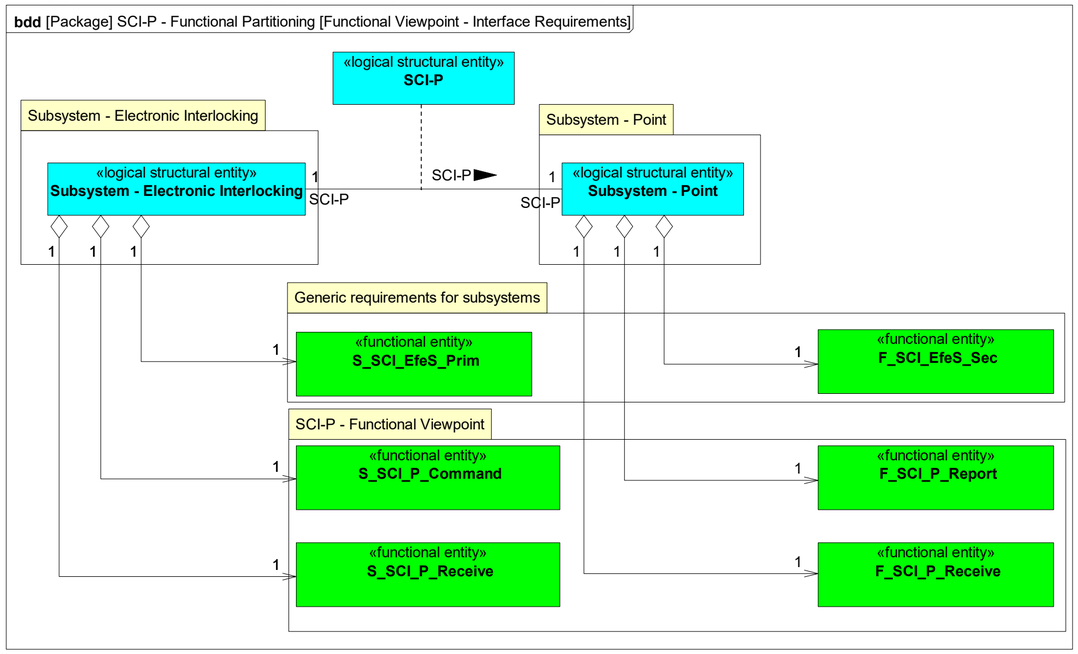
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.5823 | Req | Functional Viewpoint - Subsystem Requirements - Functional Entity STD 7<br>  <br> | This state machine diagram describes the requirements for the following functionalities:<br><br>- observes and forward the Moving state for an 4W PM<br>- observes and forward the Detection state for an 4W PM | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5826 | Def | OPERATING | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6221 | Def | when(d51in_EST_EfeS_State = "NO_OPERATING_VOLTAGE" OR d51in_EST_EfeS_State = "FALLBACK_MODE")/D24out_Drive_Voltage_Right := FALSE; D23out_Drive_Voltage_Left := FALSE;{OPERATING - WAITING_FOR_INITIALISING} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7078 | Def | when(d51in_EST_EfeS_State = "BOOTING")/D24out_Drive_Voltage_Right := FALSE; D23out_Drive_Voltage_Left := FALSE;{OPERATING - OPERATING} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5915 | Def | DETECTION | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5918 | Def | entry/D25out_Detection_Voltage := TRUE; D24out_Drive_Voltage_Right := FALSE; D23out_Drive_Voltage_Left := FALSE;{State-internal in DETECTION} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6935 | Def | when(D27in_4W_PM_Position = "LEFT_DETECTED")/d10out_PM_Position := "LEFT";{State-internal in DETECTION} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6936 | Def | when(D27in_4W_PM_Position = "NO_END_POSITION")/d10out_PM_Position := "NO_END_POSITION";{State-internal in DETECTION} | | Basic 4-wire single P<br>Basic 4-wire multiple P |

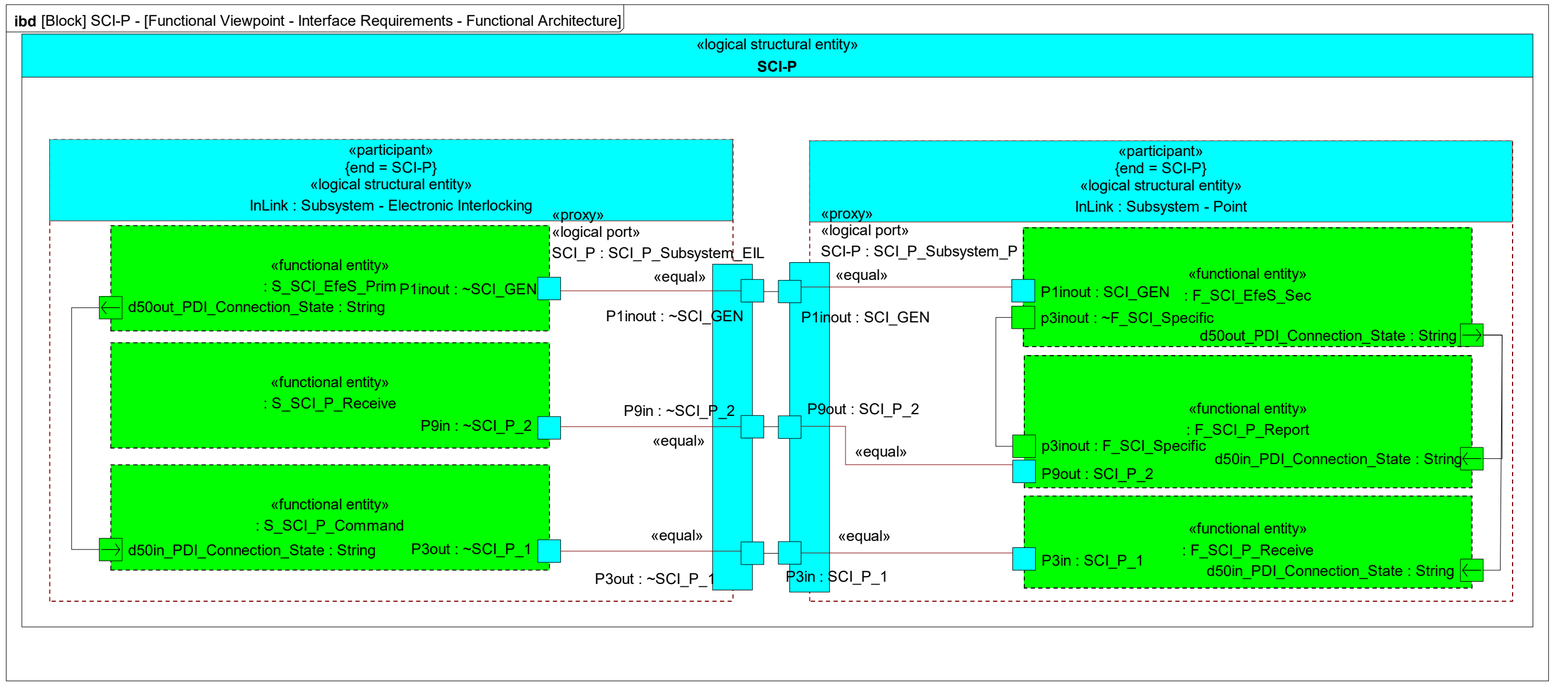
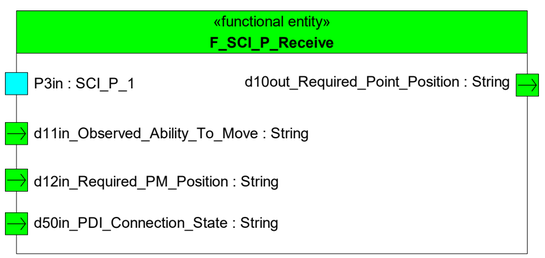
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6937 | Def | when(D27in_4W_PM_Position = "RIGHT_DETECTED")/d10out_PM_Position := "RIGHT";{State-internal in DETECTION} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6938 | Def | when(D27in_4W_PM_Position = "UNINTENDED_POSITION")/d10out_PM_Position := "UNINTENDED_POSITION";{State-internal in DETECTION} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6939 | Def | when(d12in_Required_PM_Position = "LEFT")[D20in_Con_Drive_Capability AND D45in_Con_Active]/{DETECTION - MOVING_LEFT_PM} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6940 | Def | when(d12in_Required_PM_Position = "RIGHT")[D20in_Con_Drive_Capability AND D45in_Con_Active]/{DETECTION - MOVING_RIGHT_PM} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6941 | Def | Initial1 | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6942 | Def | /{Initial1 - DETECTION} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5888 | Def | MOVING_LEFT_PM | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5890 | Def | entry/D25out_Detection_Voltage := FALSE;<br>D24out_Drive_Voltage_Right := FALSE;<br>D23out_Drive_Voltage_Left := TRUE;<br>d10out_PM_Position := "NO_END_POSITION";{State-internal in MOVING_LEFT_PM} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5877 | Def | when(d12in_Required_PM_Position = "RIGHT")[D20in_Con_Drive_Capability AND D45in_Con_Active]/{MOVING_LEFT_PM - MOVING_RIGHT_PM} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6943 | Def | when(D27in_4W_PM_Position = "LEFT_REACHED")/{MOVING_LEFT_PM - DETECTION} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6944 | Def | when(d12in_Required_PM_Position = "UNCOMMANDED")/{MOVING_LEFT_PM - DETECTION} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5905 | Def | MOVING_RIGHT_PM | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5907 | Def | entry/D25out_Detection_Voltage := FALSE;<br>D24out_Drive_Voltage_Right := TRUE;<br>D23out_Drive_Voltage_Left := FALSE;<br>d10out_PM_Position := "NO_END_POSITION";{State-internal in MOVING_RIGHT_PM} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5894 | Def | when(d12in_Required_PM_Position = "LEFT")[D20in_Con_Drive_Capability AND D45in_Con_Active]/{MOVING_RIGHT_PM - MOVING_LEFT_PM} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6945 | Def | when(D27in_4W_PM_Position = "RIGHT_REACHED")/{MOVING_RIGHT_PM - DETECTION} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6946 | Def | when(d12in_Required_PM_Position = "UNCOMMANDED")/{MOVING_RIGHT_PM - DETECTION} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7134 | Def | when(D26in_Internal_Trigger_Ability_To_Move_PM_Available)/d19out_Ability_To_Move_PM := "ABLE";{State-internal in OPERATING} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7135 | Def | when(NOT D26in_Internal_Trigger_Ability_To_Move_PM_Available)/d19out_Ability_To_Move_PM := "UNABLE";{State-internal in OPERATING} | | Basic 4-wire single P<br>Basic 4-wire multiple P |

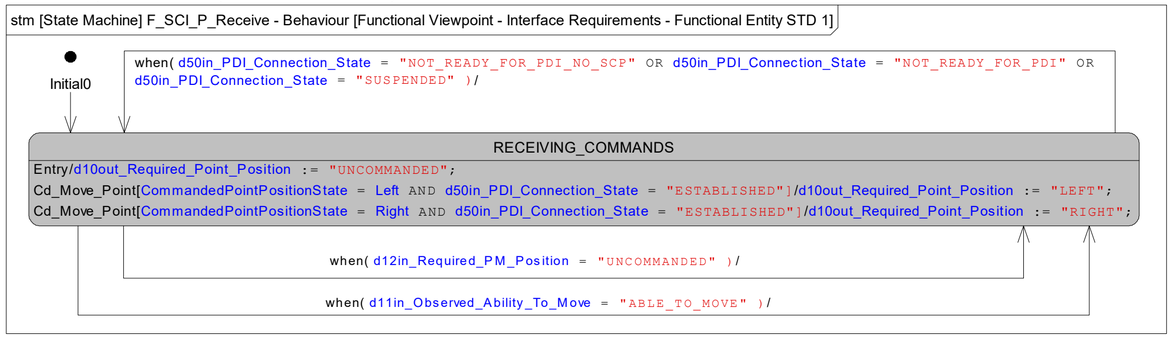
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.5909 | Def | Initial0 | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5910 | Def | /{Initial0 - WAITING_FOR_INITIALISING} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5911 | Def | WAITING_FOR_INITIALISING | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6220 | Def | when(d51in_EST_EfeS_State = "BOOTING")[D45in_Con_Active]/{WAITING_FOR_INITIALISING - Junction} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6487 | Def | Junction | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6488 | Def | [D20in_Con_Drive_Capability AND D26in_Internal_Trigger_Ability_To_Move_PM_Available]/d19out_Ability_To_Move_PM := "ABLE";{Junction - OPERATING} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6489 | Def | [NOT D20in_Con_Drive_Capability]/d19out_Ability_To_Move_PM := "NOT_USED";{Junction - OPERATING} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6490 | Def | [D20in_Con_Drive_Capability AND NOT D26in_Internal_Trigger_Ability_To_Move_PM_Available]/d19out_Ability_To_Move_PM := "UNABLE";{Junction - OPERATING} | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.889 | Head | **3.4 Subsystem - Point - Interfaces** | | |
| Eu.P.913 | Head | **3.4.1 SCI-P (Subsystem - Electronic Interlocking)** | | |
| Eu.P.6154 | Head | **3.4.1.1 SCI-P - Logical Viewpoint** | | |
| Eu.P.6281 | Head | **3.4.1.1.1 SCI-P - Logical Context** | | |
| Eu.P.3287 | Def | [Package] SCI-P - Logical Context [Logical Viewpoint - Interface Definition]<br>  <br> | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6041 | Head | **3.4.1.2 SCI-P - Information Flows** | | |
| Eu.P.3064 | Info | The generic commands and messages through the SCI_P_Subsystem_EIL are specified in [Eu.Doc.119]. | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6042 | Def | [Package] SCI-P - Information Flows [Interface Requirements - Direction of Information Objects]<br>**bdd** [Package] SCI-P - Information Flows [Interface Requirements - Direction of Information Objects]<br>«information flow»<br>**SCI_P_Subsystem_EIL**<br>**proxyPorts**<br>«ProxyPort» P1inout : SCI_GEN<br>«ProxyPort» P3out : SCI_P_1<br>«ProxyPort» P9in : SCI_P_2<br>«information flow»<br>**SCI_P_Subsystem_P**<br>**proxyPorts**<br>«ProxyPort» P1inout : SCI_GEN<br>«ProxyPort» P3in : SCI_P_1<br>«ProxyPort» P9out : SCI_P_2<br>«information flow»<br>**SCI_P_1**<br>prov «signal» Cd_Move_Point<br>«information flow»<br>**SCI_P_2**<br>reqd «signal» Msg_Point_Position<br>reqd «signal» Msg_Ability_To_Move_Point<br>«signal» Msg_Movement_Failed<br>«information flow»<br>**SCI_GEN**<br>prov «signal» : Cd_PDI_Version_Check<br>reqd «signal» : Msg_PDI_Version_Check<br>prov «signal» : Cd_Close_PDI<br>prov «signal» : Cd_Initialisation_Request<br>reqd «signal» : Msg_Start_Initialisation<br>reqd «signal» : Msg_Initialisation_Completed<br>prov «signal» : Cd_Release_PDI_for_Maintenance<br>reqd «signal» : Msg_PDI_Available<br>reqd «signal» : Msg_PDI_Not_Available<br>reqd «signal» : Msg_Reset_PDI | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

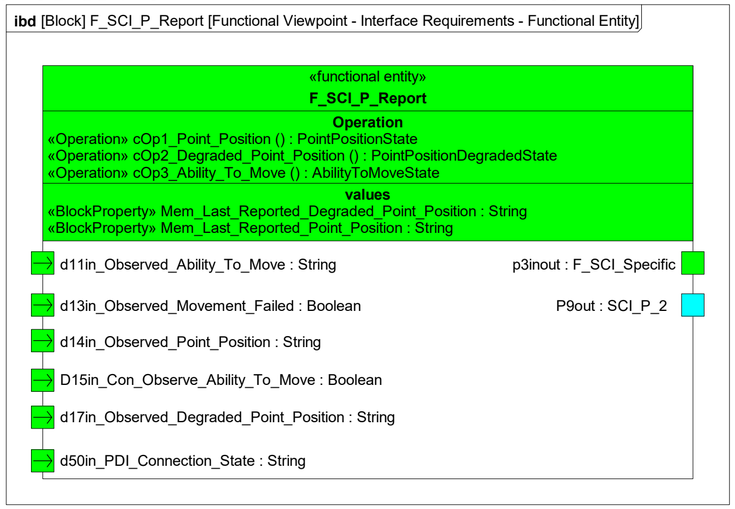
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6044 | Def | [Package] SCI-P - Information Flows [Interface Requirements - Information Objects]<br>  <br> | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6183 | Def | Cd_Move_Point | Command (Cd) from Subsystem - Electronic Interlocking to Subsystem - Point to move the Point into the commanded position. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6185 | Def | Msg_Ability_To_Move_Point | Message (Msg) from Subsystem - Point to Subsystem - Electronic Interlocking about the current Ability to move point. | Option Able to move |
| Eu.P.6187 | Def | Msg_Point_Position | Message (Msg) from Subsystem - Point to Subsystem - Electronic Interlocking about the current Point position. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6190 | Def | Msg_Movement_Failed | Message (Msg) from Subsystem - Point to Subsystem - Electronic Interlocking that the ongoing moving failed. The Subsystem - Point has abandoned the moving for one of the following reasons:<br>- The time period Con_tmax_Point_Operation has expired.<br>- The point movement has failed.<br><br>Note: The conditions defining a failed moment can be supplier and IM specific. They may include, for example, a failure of the star point buildup or a situation in which only non-driven point machines have not yet reached the commanded end position. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6045 | Head | **3.4.1.3 SCI-P - Functional Viewpoint** | | |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6280 | Head | **3.4.1.3.1 SCI-P - Functional Partitioning** | | |
| Eu.P.6155 | Def | [Package] SCI-P - Functional Partitioning [Functional Viewpoint - Interface Requirements]<br>  <br> | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6244 | Head | **3.4.1.3.2 SCI-P - Functional Architecture** | | |
| Eu.P.3279 | Info | SCI-P | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

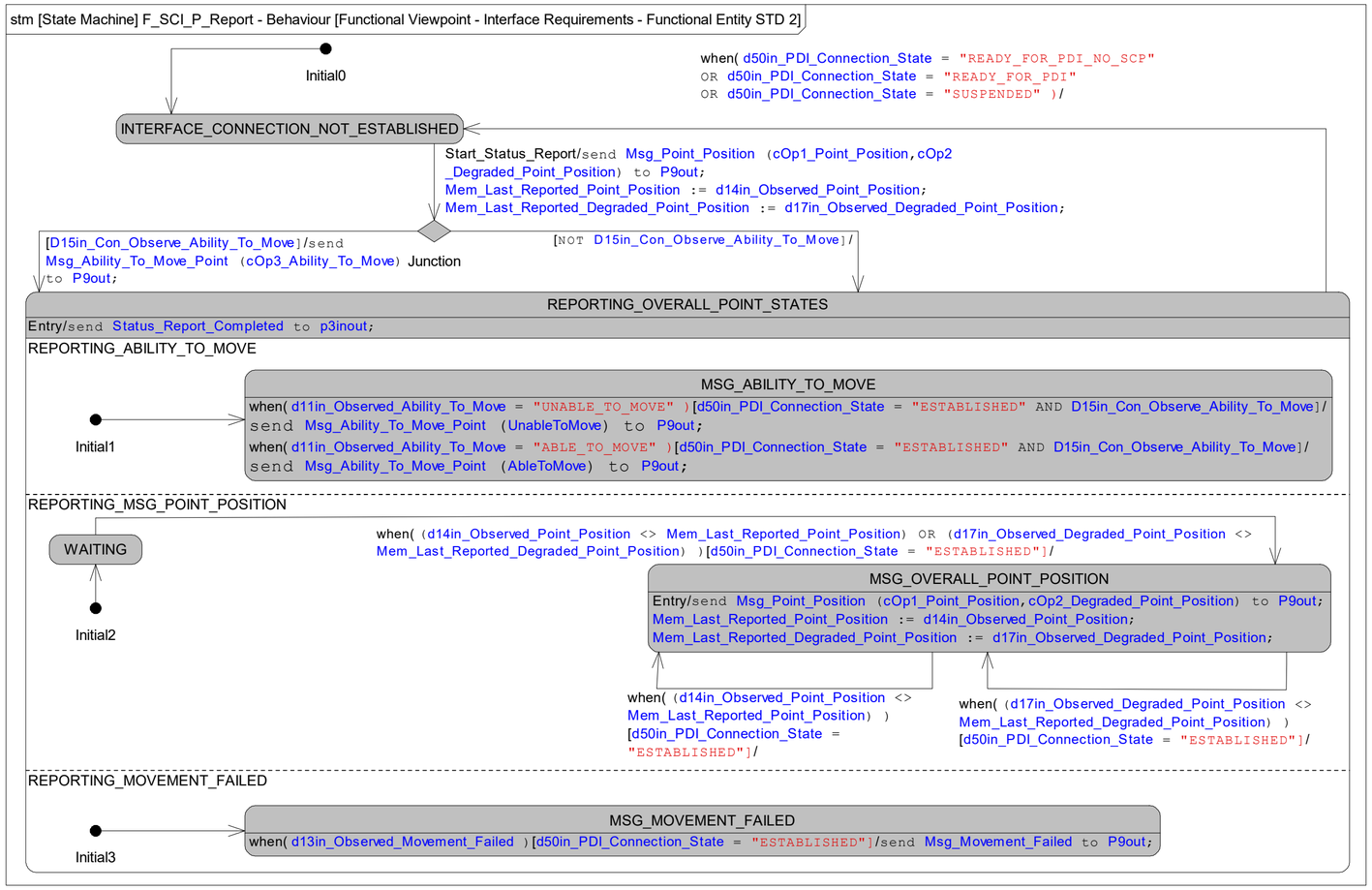
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.3286 | Def | [Block] SCI-P - [Functional Viewpoint - Interface Requirements - Functional Architecture]<br>**ibd** [Block] SCI-P - [Functional Viewpoint - Interface Requirements - Functional Architecture]<br>  <br> | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6245 | Head | **3.4.1.3.3 SCI-P - Functional Entities** | | |
| Eu.P.6412 | Info | F_SCI_P_Receive | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6413 | Req | [Block] F_SCI_P_Receive [Functional Viewpoint - Interface Requirements - Functional Entity]<br>**ibd** [Block] F_SCI_P_Receive [Functional Viewpoint - Interface Requirements - Functional Entity]<br>  <br> | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6415 | Def | d10out_Required_Point_Position | The port d10out_Required_Point_Positiont contains the Point Target Position. The following values are valid: "LEFT", "RIGHT". | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6429 | Def | d50in_PDI_Connection_State | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6430 | Def | d12in_Required_PM_Position | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7018 | Def | d11in_Observed_Ability_To_Move | | Option Able to move |
| Eu.P.6470 | Def | P3in | The port P3in exchanges information objects according to SCI_P_1. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6431 | Info | F_SCI_P_Receive - Behaviour | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6432 | Req | Functional Viewpoint - Interface Requirements - Functional Entity STD 1<br>  <br> | This state machine diagram describes the requirements for the following functionalities:<br><br>- receives the commanded point position state and forwards it to the internal logic of the Subsystem - Point | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6433 | Def | Initial0 | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6434 | Def | /{Initial0 - RECEIVING_COMMANDS} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6435 | Def | RECEIVING_COMMANDS | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6469 | Def | when(d50in_PDI_Connection_State = "NOT_READY_FOR_PDI_NO_SCP" OR d50in_PDI_Connection_State = "NOT_READY_FOR_PDI" OR d50in_PDI_Connection_State = "SUSPENDED")/{RECEIVING_COMMANDS - RECEIVING_COMMANDS} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7019 | Def | Cd_Move_Point[CommandedPointPositionState = Left AND d50in_PDI_Connection_State = "ESTABLISHED"]/d10out_Required_Point_Position := "LEFT";{State-internal in RECEIVING_COMMANDS} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7020 | Def | Cd_Move_Point[CommandedPointPositionState = Right AND d50in_PDI_Connection_State = "ESTABLISHED"]/d10out_Required_Point_Position := "RIGHT";{State-internal in RECEIVING_COMMANDS} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7021 | Def | entry/d10out_Required_Point_Position := "UNCOMMANDED";{State-internal in RECEIVING_COMMANDS} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7022 | Def | when(d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE")/{RECEIVING_COMMANDS - RECEIVING_COMMANDS} | | Option Able to move |
| Eu.P.7023 | Def | when(d12in_Required_PM_Position = "UNCOMMANDED")/{RECEIVING_COMMANDS - RECEIVING_COMMANDS} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6522 | Info | F_SCI_P_Report | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

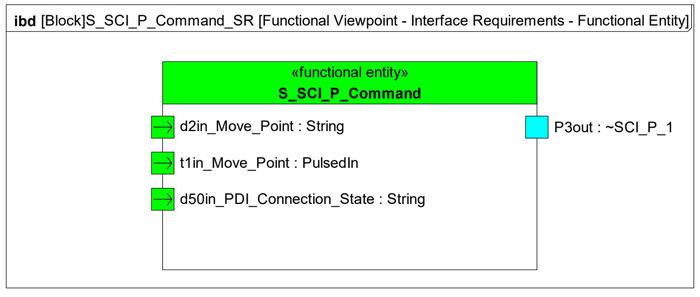
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6523 | Req | [Block] F_SCI_P_Report [Functional Viewpoint - Interface Requirements - Functional Entity]<br>  <br> | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6524 | Def | /* cOp1_Point_Position */<br><br>if (d14in_Observed_Point_Position = "LEFT") then<br>return PointPositionState.Left;<br>elseif (d14in_Observed_Point_Position = "RIGHT") then<br>return PointPositionState.Right;<br>elseif (d14in_Observed_Point_Position = "NO_END_POSITION") then<br>return PointPositionState.NoEndPosition;<br>elseif (d14in_Observed_Point_Position = "UNINTENDED_POSITION") then<br>return PointPositionState.UnintendedPosition;<br>end if | cOp1_Point_Position | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6525 | Def | /* cOp2_Degraded_Point_Position */<br><br>if (d17in_Observed_Degraded_Point_Position = "DEGRADED_LEFT") then<br>return PointPositionDegradedState.DegradedLeft;<br>elseif (d17in_Observed_Degraded_Point_Position = "DEGRADED_RIGHT") then<br>return PointPositionDegradedState.DegradedRight;<br>elseif (d17in_Observed_Degraded_Point_Position = "DEGRADED_DENIED") then<br>return PointPositionDegradedState.NotApplicable;<br>else<br>return PointPositionDegradedState.NotDegraded;<br>end if | cOp2_Degraded_Point_Position | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6526 | Def | /* cOp3_Ability_To_Move */<br><br>if (d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE") then<br>return AbilityToMoveState.AbleToMove;<br>elseif (d11in_Observed_Ability_To_Move = "UNABLE_TO_MOVE") then<br>return AbilityToMoveState.UnableToMove;<br>end if | cOp3_Ability_To_Move | Option Able to move |
| Eu.P.6527 | Def | d17in_Observed_Degraded_Point_Position | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6528 | Def | d50in_PDI_Connection_State | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

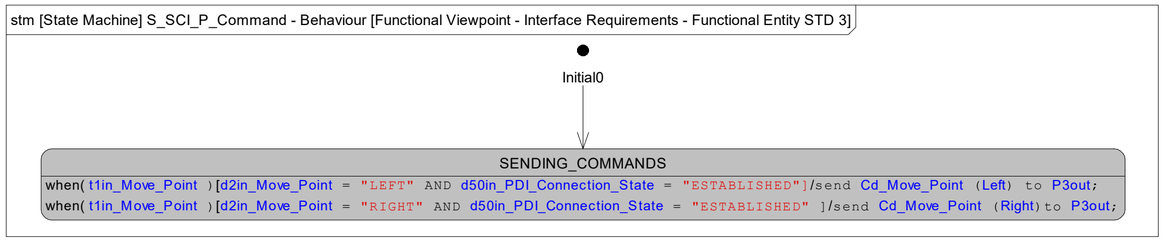
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6529 | Def | d14in_Observed_Point_Position | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6530 | Def | d11in_Observed_Ability_To_Move | | Option Able to move |
| Eu.P.6556 | Def | P9out | The port P9out exchanges information objects according to SCI_P_2. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6557 | Def | p3inout | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7024 | Def | d13in_Observed_Movement_Failed | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7025 | Def | D15in_Con_Observe_Ability_To_Move | | Option Able to move |
| Eu.P.6531 | Info | F_SCI_P_Report - Behaviour | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

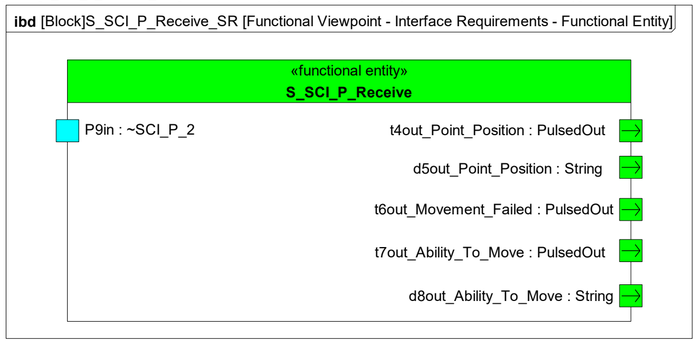
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6532 | Req | Functional Viewpoint - Interface Requirements - Functional Entity STD 2<br>  <br> | This state machine diagram describes the requirements for the following functionalities:<br><br>- receives the observed ability to move status from internal logic and reports this to the Subsystem - Electronic Interlocking<br>- receives the observed Point Position status from internal logic and reports this to the Subsystem - Electronic Interlocking<br>- receives the observed Movement Failed status from internal logic and reports this to the Subsystem - Electronic Interlocking | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6533 | Def | Initial0 | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6534 | Def | /{Initial0 - INTERFACE_CONNECTION_NOT_ESTABLISHED} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6535 | Def | INTERFACE_CONNECTION_NOT_ESTABLISHED | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6536 | Def | Start_Status_Report/send Msg_Point_Position (cOp1_Point_Position,cOp2_Degraded_Point_Position) to P9out;<br>Mem_Last_Reported_Point_Position := d14in_Observed_Point_Position;<br>Mem_Last_Reported_Degraded_Point_Position := d17in_Observed_Degraded_Point_Position;{INTERFACE_CONNECTION_NOT_ESTABLISHED - Junction} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6537 | Def | REPORTING_OVERALL_POINT_STATES | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6538 | Def | REPORTING_ABILITY_TO_MOVE | | Option Able to move |
| Eu.P.6539 | Def | Initial1 | | Option Able to move |
| Eu.P.6540 | Def | /{Initial1 - MSG_ABILITY_TO_MOVE} | | Option Able to move |
| Eu.P.6541 | Def | MSG_ABILITY_TO_MOVE | | Option Able to move |
| Eu.P.6542 | Def | when(d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE")[d50in_PDI_Connection_State = "ESTABLISHED" AND D15in_Con_Observe_Ability_To_Move]/<br>send Msg_Ability_To_Move_Point (AbleToMove) to P9out;{State-internal in MSG_ABILITY_TO_MOVE} | | Option Able to move |
| Eu.P.6543 | Def | when(d11in_Observed_Ability_To_Move = "UNABLE_TO_MOVE")[d50in_PDI_Connection_State = "ESTABLISHED" AND D15in_Con_Observe_Ability_To_Move]/<br>send Msg_Ability_To_Move_Point (UnableToMove) to P9out;{State-internal in MSG_ABILITY_TO_MOVE} | | Option Able to move |
| Eu.P.6544 | Def | REPORTING_MSG_POINT_POSITION | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6545 | Def | Initial2 | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6546 | Def | /{Initial2 - WAITING} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6547 | Def | MSG_OVERALL_POINT_POSITION | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6548 | Def | entry/send Msg_Point_Position (cOp1_Point_Position,cOp2_Degraded_Point_Position) to P9out;<br>Mem_Last_Reported_Point_Position := d14in_Observed_Point_Position;<br>Mem_Last_Reported_Degraded_Point_Position := d17in_Observed_Degraded_Point_Position;{State-internal in MSG_OVERALL_POINT_POSITION} | | Basic non-4-wire single P<br>Basic non-4-wire multiple |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| | | | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6549 | Def | when((d17in_Observed_Degraded_Point_Position <> Mem_Last_Reported_Degraded_Point_Position))[d50in_PDI_Connection_State = "ESTABLISHED"]/{MSG_OVERALL_POINT_POSITION - MSG_OVERALL_POINT_POSITION} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6550 | Def | when((d14in_Observed_Point_Position <> Mem_Last_Reported_Point_Position))[d50in_PDI_Connection_State = "ESTABLISHED"]/{MSG_OVERALL_POINT_POSITION - MSG_OVERALL_POINT_POSITION} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6551 | Def | WAITING | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6553 | Def | when((d14in_Observed_Point_Position <> Mem_Last_Reported_Point_Position) OR (d17in_Observed_Degraded_Point_Position <> Mem_Last_Reported_Degraded_Point_Position))[d50in_PDI_Connection_State = "ESTABLISHED"]/{WAITING - MSG_OVERALL_POINT_POSITION} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6554 | Def | entry/send Status_Report_Completed to p3inout;{State-internal in REPORTING_OVERALL_POINT_STATES} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6555 | Def | when(d50in_PDI_Connection_State = "READY_FOR_PDI_NO_SCP"<br>OR d50in_PDI_Connection_State = "READY_FOR_PDI"<br>OR d50in_PDI_Connection_State = "SUSPENDED")/{REPORTING_OVERALL_POINT_STATES - INTERFACE_CONNECTION_NOT_ESTABLISHED} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7029 | Def | REPORTING_MOVEMENT_FAILED | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7030 | Def | Initial3 | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7031 | Def | /{Initial3 - MSG_MOVEMENT_FAILED} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| | | | | Basic 4-wire multiple P |
| Eu.P.7032 | Def | MSG_MOVEMENT_FAILED | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7033 | Def | when(d13in_Observed_Movement_Failed)[d50in_PDI_Connection_State = "ESTABLISHED"]/send Msg_Movement_Failed to P9out;{State-internal in MSG_MOVEMENT_FAILED} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7026 | Def | Junction | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7027 | Def | [NOT D15in_Con_Observe_Ability_To_Move]/{Junction - REPORTING_OVERALL_POINT_STATES} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7028 | Def | [D15in_Con_Observe_Ability_To_Move]/send Msg_Ability_To_Move_Point (cOp3_Ability_To_Move) to P9out;{Junction - REPORTING_OVERALL_POINT_STATES} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.4704 | Info | S_SCI_P_Command | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.4711 | Req | [Block]S_SCI_P_Command_SR [Functional Viewpoint - Interface Requirements - Functional Entity]<br>  <br> | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.4707 | Def | d2in_Move_Point | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.4706 | Def | d50in_PDI_Connection_State | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6153 | Def | t1in_Move_Point | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6477 | Def | P3out | The port P3out exchanges information objects according to SCI_P_1. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.4712 | Info | S_SCI_P_Command - Behaviour | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.4729 | Req | Functional Viewpoint - Interface Requirements - Functional Entity STD 3<br>  <br> | This state machine diagram describes the requirements for the following functionalities:<br><br>- receives the Moving Command to be set from internal logic and commands this to the Subsystem - Point | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.4713 | Def | Initial0 | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.4714 | Def | /{Initial0 - SENDING_COMMANDS} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6131 | Def | SENDING_COMMANDS | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6132 | Def | when(t1in_Move_Point)[d2in_Move_Point = "RIGHT" AND d50in_PDI_Connection_State = "ESTABLISHED" ]/send Cd_Move_Point (Right)to P3out;{State-internal in SENDING_COMMANDS} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6133 | Def | when(t1in_Move_Point)[d2in_Move_Point = "LEFT" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Move_Point (Left) to P3out;{State-internal in SENDING_COMMANDS} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7034 | Info | S_SCI_P_Receive | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7035 | Req | [Block]S_SCI_P_Receive_SR [Functional Viewpoint - Interface Requirements - Functional Entity]<br>  <br> | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7037 | Def | d5out_Point_Position | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7038 | Def | d8out_Ability_To_Move | | Option Able to move |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.7039 | Def | P9in | The port P9in exchanges information objects according to SCI_P_2. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7072 | Def | t4out_Point_Position | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7073 | Def | t6out_Movement_Failed | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7074 | Def | t7out_Ability_To_Move | | Option Able to move |
| Eu.P.7040 | Info | S_SCI_P_Receive - Behaviour | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.7041 | Req | Functional Viewpoint - Interface Requirements - Functional Entity STD 4<br> <br> | This state machine diagram describes the requirements for the following functionalities:<br><br>- receives the reported point position status and forwards it to the internal logic<br>- receives the reported ability to move status and forwards it to the internal logic<br>- receives the reported movement failed report and forwards it to the internal logic | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7042 | Def | Initial0 | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7043 | Def | /{Initial0 - OPERATING} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.7044 | Def | OPERATING | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7045 | Def | ABILITY_TO_MOVE_OBSERVING | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7046 | Def | Initial2 | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7047 | Def | /{Initial2 - RECEIVE_ABILITY_TO_MOVE_REPORT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7048 | Def | RECEIVE_ABILITY_TO_MOVE_REPORT | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7049 | Def | Msg_Ability_To_Move_Point[ReportedAbilityToMoveState = UnableToMove]/d8out_Ability_To_Move := "UNABLE_TO_MOVE";<br>t7out_Ability_To_Move := TRUE;{State-internal in RECEIVE_ABILITY_TO_MOVE_REPORT} | | Option Able to move |
| Eu.P.7050 | Def | Msg_Ability_To_Move_Point[ReportedAbilityToMoveState = AbleToMove]/d8out_Ability_To_Move := "ABLE_TO_MOVE";<br>t7out_Ability_To_Move := TRUE;{State-internal in RECEIVE_ABILITY_TO_MOVE_REPORT} | | Option Able to move |
| Eu.P.7051 | Def | MOVEMENT_FAILED_OBSERVING | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7052 | Def | Initial3 | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7053 | Def | /{Initial3 - RECEIVE_MOVEMENT_FAILED_REPORT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7054 | Def | RECEIVE_MOVEMENT_FAILED_REPORT | | Basic non-4-wire single P<br>Basic non-4-wire multiple P |

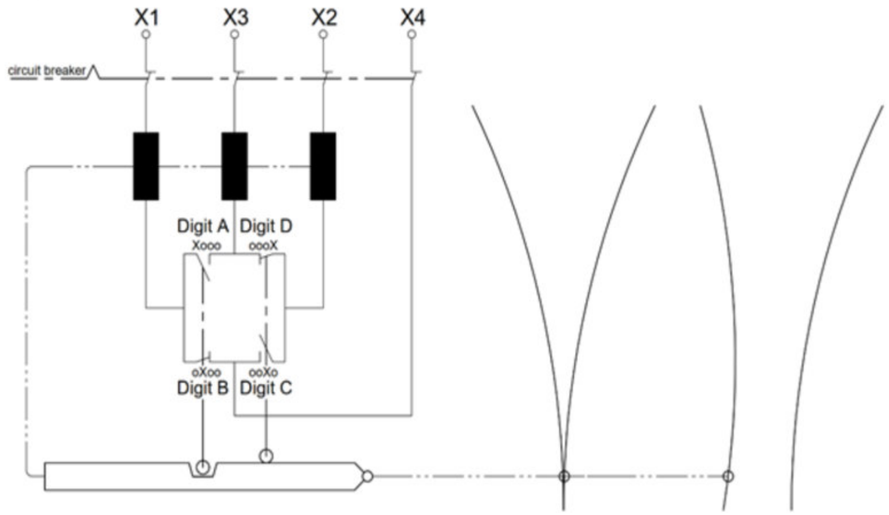
| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| | | | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7055 | Def | Msg_Movement_Failed/t6out_Movement_Failed := TRUE;{State-internal in RECEIVE_MOVEMENT_FAILED_REPORT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7056 | Def | POINT_POSITION_OBSERVING | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7057 | Def | Initial1 | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7058 | Def | /{Initial1 - RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7059 | Def | RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7060 | Def | Msg_Point_Position[ReportedPointPositionState = Left AND ReportedDegradedPointPosition = NotApplicable]/d5out_Point_Position := "LEFT, NOT_APPLICABLE";<br>t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7061 | Def | Msg_Point_Position[ReportedPointPositionState = UnintendedPosition AND ReportedDegradedPointPosition = DegradedRight]/d5out_Point_Position := "UNINTENDED_POSITION, DEGRADED_RIGHT";<br>t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7062 | Def | Msg_Point_Position[ReportedPointPositionState = UnintendedPosition AND ReportedDegradedPointPosition = NotDegraded]/d5out_Point_Position := "UNINTENDED_POSITION, NOT_DEGRADED";<br>t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7063 | Def | Msg_Point_Position[ReportedPointPositionState = UnintendedPosition AND ReportedDegradedPointPosition = NotApplicable]/d5out_Point_Position := "UNINTENDED_POSITION, NOT_APPLICABLE";<br>t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| | | | | Basic 4-wire multiple P |
| Eu.P.7064 | Def | Msg_Point_Position[ReportedPointPositionState = Left AND ReportedDegradedPointPosition = NotDegraded]/d5out_Point_Position := "LEFT, NOT_DEGRADED";<br>t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7065 | Def | Msg_Point_Position[ReportedPointPositionState = Right AND ReportedDegradedPointPosition = NotApplicable]/d5out_Point_Position := "RIGHT, NOT_APPLICABLE";<br>t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7066 | Def | Msg_Point_Position[ReportedPointPositionState = Right AND ReportedDegradedPointPosition = NotDegraded]/d5out_Point_Position := "RIGHT, NOT_DEGRADED";<br>t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7067 | Def | Msg_Point_Position[ReportedPointPositionState = NoEndPosition AND ReportedDegradedPointPosition = DegradedLeft]/d5out_Point_Position := "NO_END_POSITION, DEGRADED_LEFT";<br>t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7068 | Def | Msg_Point_Position[ReportedPointPositionState = NoEndPosition AND ReportedDegradedPointPosition = DegradedRight]/d5out_Point_Position := "NO_END_POSITION, DEGRADED_RIGHT";<br>t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7069 | Def | Msg_Point_Position[ReportedPointPositionState = NoEndPosition AND ReportedDegradedPointPosition = NotDegraded]/d5out_Point_Position := "NO_END_POSITION, NOT_DEGRADED";<br>t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7070 | Def | Msg_Point_Position[ReportedPointPositionState = NoEndPosition AND ReportedDegradedPointPosition = NotApplicable]/d5out_Point_Position := "NO_END_POSITION, NOT_APPLICABLE";<br>t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7071 | Def | Msg_Point_Position[ReportedPointPositionState = UnintendedPosition AND ReportedDegradedPointPosition = DegradedLeft]/d5out_Point_Position := "UNINTENDED_POSITION, DEGRADED_LEFT";<br>t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT} | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.929 | Head | **3.4.2 SMI-P (Subsystem - Maintenance and Data Management)** | | |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.3066 | Info | The generic FlowSpecification and the related FlowProperties through SMI-P are specified in [Eu.Doc.120]. | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.925 | Head | **3.4.3 SDI-P (Subsystem - Maintenance and Data Management)** | | |
| Eu.P.3065 | Info | The generic data points through the SDI-P are specified in [Eu.Doc.94].<br>The specific data points through the SDI-P are specified in [Eu.Doc.80]. | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6156 | Head | **3.4.4 SSI-P (Subsystem - Security Services Platform)** | | |
| Eu.P.6157 | Info | The generic content through SSI-P is specified in [SP-SEC-ServSpec]. | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.910 | Head | **3.4.5 P4 (Basic Data Identifier)** | | |
| Eu.P.3063 | Info | The generic FlowSpecification and the related FlowProperties through P4 are specified in [Eu.Doc.20]. | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.890 | Head | **3.4.6 P1 (Maintainer)** | | |
| Eu.P.3173 | Info | The generic FlowProperties through  P1 are specified in [Eu.Doc.20]. | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6206 | Info | The defined FlowProperties through P1 are mandatory only when the physical interfaces related to the specific maintainer information are available on the Subsystem – Point.<br>Example: The FlowProperty Point_Unintended_Position is only mandatory when the Subsystem – Point is equipped with a point machine interface that can detect unintended position. | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.891 | Info | Maintainer | Definition of the InformationFlow (by FlowSpecification) for Maintenance/Operation/Display P1 (Maintainer). | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.896 | Def | Point_Moving | Displays the moving of the point at the local status display. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.1377 | Def | End_Position_R | Displays the status of the detection of point end position on the right hand. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.894 | Def | End_Position_L | Displays the status of the detection of point end position on the left hand. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.3037 | Def | Point_Unintended_Position | Displays the unintended position of the point at the local status display (in an unintended position or not). | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.4949 | Def | P3_Moving_N | Displays the state for each respective physical Point Machine Output (N > 0). (Moving, Stopped). | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.4950 | Def | P3_Status_N | Displays the state for each respective physical Point Machine Input (N > 0). (End_Position, No_End_Position, Unintended_Position). | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.902 | Head | **3.4.7 P3 (Point machine)** | | |
| Eu.P.903 | Info | Point_machine | Definition of the InformationFlow (by FlowSpecification) for the Control Interfaces P3 (Point machine).<br><br>Note:<br>The behaviour of the interfaces P3 is described generically.<br>The Subsystem - Point needs to be able to write and to read the generic information objects of the statuses from the Point machine. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.905 | Def | Information_No_End_Position | Information object from Point machine to Subsystem - Point that the Point machine has No end position.<br>Note: This is expressed as a 4-wire pattern in case the Point machine is implemented with a 4-wire interface.<br>The interpretation of the 4-wire patterns depends on the driving/detection state and the last commanded position. The interpretation tables can be found in Eu.P.6797. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.906 | Def | Information_Unintended_Position | Information object from Point machine to Subsystem - Point that the Point machine has an Unintended position position.<br>Note 1: This is expressed as a 4-wire pattern in case the Point machine is implemented with a 4-wire interface. The interpretation of the 4-wire patterns depends on the driving/detection state and the last commanded position. The interpretation tables can be found in Eu.P.6797.<br>Note 2: This information may not be available in case the Point machine is implemented with a non-4-wire interface. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.907 | Def | Moving | Information object from Subsystem - Point to Point machine to move the Point machine to an end position (left hand position or right hand position). | Basic non-4-wire single P<br>Basic non-4-wire multiple |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| | | | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.909 | Def | Stop_Moving | Information object from Subsystem - Point to Point machine to stop Moving the Point machine. | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.5364 | Def | Information_Ability_To_Move_Point | Information object from Point machine to Subsystem - Point that the Point machine has Drive voltage (able) or has no Drive voltage (unable).<br>Note: The reason for reporting ability to move depends on national specification and also Drive voltage levels monitored shall be defined by national specifications. In future phases of the System Pillar, national specifications will be replaced by harmonised specifications. | Option Able to move |
| Eu.P.6274 | Def | Detection_Voltage | Information object from Subsystem - Point to Point machine to switch the Point machine to Detection Mode (on or off).<br>Note: This information is only available in case the Point machine is implemented with a 4-wire interface. | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7016 | Def | Information_End_Position_Detected | Information object from Point machine to Subsystem - Point that the Point machine has reliably reached an End position (left hand position or right hand position).<br>Note 1: This is expressed as a 4-wire pattern, in case the Point machine is implemented with a 4-wire interface.<br>LEFT: 1010<br>RIGHT: 0101 | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7017 | Def | Information_End_Position_Reached | Information object from Point machine to Subsystem - Point that the Point machine has reached an End position (left hand position or right hand position), but the position has not yet been reliably detected.<br>Note 1: This information is only available in case the Point machine is implemented with a 4-wire interface. This is expressed as a 4-wire pattern.<br>LEFT: 1010<br>RIGHT: 0101 | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6797 | Head | **3.4.7.1 Interpretation tables of 4-wire patterns** | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6801 | Info | For the 4-wire implementation, the input information is represented as 4-wire patterns. | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6806 | Info | The 4-wire pattern consists of four digits, each being in a state of „1" or „0". The state of „1" represents a closed contact in the 4-wire circuit while „0" represents an open contact in the 4-wire circuit. | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6805 | Info | There are four contact pairs, where each pair is represented by a specific digit in the 4-wire pattern (ABCD):<br>Contact 1+3 -> Digit A<br>Contact 1+4 -> Digit B<br>Contact 2+4 -> Digit C<br>Contact 2+3 -> Digit D | | Basic 4-wire single P<br>Basic 4-wire multiple P |

<table>
<tr><th>ID</th><th>Type</th><th>Requirement Part 1</th><th>Requirement Part 2</th><th>Func. Pkg.</th></tr>
<tr><td>Eu.P.6807</td><td>Info</td><td>The figure shows a schematic representation of the 4-wire circuit<br> 
</td><td></td><td>Basic 4-wire single P<br>Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.6804</td><td>Req</td><td>The interpretation of the 4-wire patterns is state-dependent. There are 5 interpretations states. The interpretation state is determined by two factors.<br>- Whether the Subsystem Point has been commanded to move the point or the movement has stopped (moving to left, moving to right, no movement/detection)<br>- The last commanded position of the point machine (left, right, not commanded)</td><td></td><td>Basic 4-wire single P<br>Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.6803</td><td>Req</td><td>The implementation of the Subsystem Point and/or the Point machine shall be such that the last commanded position is always retained, including in case of loss of communication and in case of rebooting.<br>Note: The only case in which the state ‘not commanded’ occurs is after the initial start-up.</td><td></td><td>Basic 4-wire single P<br>Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.6703</td><td>Req</td><td>Interpretation of 4-wire patterns without command to move the point and last commanded position "Right"
<table>
<tr><th>4-wire pattern</th><th>Interpretation of information on interface P3</th></tr>
<tr><td>0000</td><td>Information_No_End_Position</td></tr>
<tr><td>0001</td><td>Information_No_End_Position</td></tr>
<tr><td>0010</td><td>Supplier specific</td></tr>
<tr><td>0011</td><td>Supplier specific</td></tr>
<tr><td>0100</td><td>Information_No_End_Position</td></tr>
<tr><td>0101</td><td>Information_End_Position_Detected (Right)</td></tr>
<tr><td>0110</td><td>Supplier specific</td></tr>
<tr><td>0111</td><td>Supplier specific (not Information_No_End_Position)</td></tr>
<tr><td>1000</td><td>Supplier specific</td></tr>
<tr><td>1001</td><td>Information_Unintended_Position</td></tr>
<tr><td>1010</td><td>Information_Unintended_Position</td></tr>
<tr><td>1011</td><td>Supplier specific (not Information_No_End_Position)</td></tr>
<tr><td>1100</td><td>Supplier specific</td></tr>
<tr><td>1101</td><td>Supplier specific (not Information_No_End_Position)</td></tr>
<tr><td>1110</td><td>Supplier specific (not Information_No_End_Position)</td></tr>
<tr><td>1111</td><td>Supplier specific (not Information_No_End_Position)</td></tr>
</table></td><td></td><td>Basic 4-wire single P<br>Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.6704</td><td>Req</td><td>Interpretation of 4-wire patterns without command to move the point and last commanded position "Left"
<table>
<tr><th>4-wire pattern</th><th>Interpretation of information on interface P3</th></tr>
<tr><td>0000</td><td>Information_No_End_Position</td></tr>
<tr><td>0001</td><td>Supplier specific</td></tr>
<tr><td>0010</td><td>Information_No_End_Position</td></tr>
<tr><td>0011</td><td>Supplier specific</td></tr>
<tr><td>0100</td><td>Supplier specific</td></tr>
<tr><td>0101</td><td>Information_Unintended_Position</td></tr>
<tr><td>0110</td><td>Supplier specific</td></tr>
<tr><td>0111</td><td>Supplier specific (not Information_No_End_Position)</td></tr>
<tr><td>1000</td><td>Information_No_End_Position</td></tr>
<tr><td>1001</td><td>Information_Unintended_Position</td></tr>
<tr><td>1010</td><td>Information_End_Position_Detected (Left)</td></tr>
<tr><td>1011</td><td>Supplier specific (not Information_No_End_Position)</td></tr>
<tr><td>1100</td><td>Supplier specific</td></tr>
<tr><td>1101</td><td>Supplier specific (not Information_No_End_Position)</td></tr>
<tr><td>1110</td><td>Supplier specific (not Information_No_End_Position)</td></tr>
<tr><td>1111</td><td>Supplier specific (not Information_No_End_Position)</td></tr>
</table></td><td></td><td>Basic 4-wire single P<br>Basic 4-wire multiple P</td></tr>
</table>

<table>
<tr><th>ID</th><th>Type</th><th>Requirement Part 1</th><th>Requirement Part 2</th><th>Func. Pkg.</th></tr>
<tr><td>Eu.P.6798</td><td>Req</td><td>Interpretation of 4-wire patterns without command to move the point and no last commanded position<br><br><table><tr><th>4-wire pattern</th><th>Interpretation of information on interface P3</th></tr><tr><td>0000</td><td>Information_No_End_Position</td></tr><tr><td>0001</td><td>Information_No_End_Position</td></tr><tr><td>0010</td><td>Information_No_End_Position</td></tr><tr><td>0011</td><td>Supplier specific</td></tr><tr><td>0100</td><td>Information_No_End_Position</td></tr><tr><td>0101</td><td>Information_End_Position_Detected (Right)</td></tr><tr><td>0110</td><td>Supplier specific</td></tr><tr><td>0111</td><td>Supplier specific</td></tr><tr><td>1000</td><td>Information_No_End_Position</td></tr><tr><td>1001</td><td>Information_Unintended_Position</td></tr><tr><td>1010</td><td>Information_End_Position_Detected (Left)</td></tr><tr><td>1011</td><td>Supplier specific</td></tr><tr><td>1100</td><td>Supplier specific</td></tr><tr><td>1101</td><td>Supplier specific</td></tr><tr><td>1110</td><td>Supplier specific</td></tr><tr><td>1111</td><td>Supplier specific</td></tr></table></td><td></td><td>Basic 4-wire single P<br>Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.6799</td><td>Req</td><td>Interpretation of 4-wire patterns when commanded to move the point to "Right", until the Subsystem Point has stopped all point machines with drive capability.<br><br><table><tr><th>4-wire pattern</th><th>Interpretation of information on interface P3</th></tr><tr><td>0101</td><td>Information_End_Position_Reached (Right)</td></tr><tr><td>All other</td><td>Information_No_End_Position</td></tr></table></td><td></td><td>Basic 4-wire single P<br>Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.6800</td><td>Req</td><td>Interpretation of 4-wire patterns when commanded to move the point to "Left", until the Subsystem Point has stopped all point machines with drive capability.<br><br><table><tr><th>4-wire pattern</th><th>Interpretation of information on interface P3</th></tr><tr><td>1010</td><td>Information_End_Position_Reached (Left)</td></tr><tr><td>All other</td><td>Information_No_End_Position</td></tr></table></td><td></td><td>Basic 4-wire single P<br>Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.6802</td><td>Req</td><td>For certain 4-wire patterns, a supplier specific interpretation is allowed. When one of these patterns is detected on the 4-wire interface to the Point Machine, the Subsystem Point may react to it in three ways:<br>- interpret the abstract state ‘Information_No_End_Position’.<br>- interpret the abstract state ‘Information_Unintended_Position’<br>- detect a fatal error (T5in_SIL_Not_Fulfilled in generic EfeS state machine)<br><br>The supplier may choose which of the 3 system reactions is most appropriate. The reaction may also be time dependent.<br>For the 4-wire patterns 0111, 1011, 1101, 1110 and 1111, it is not allowed to have a persistent interpretation as the abstract state ‘Information_No_End_Position’ if the commanded position is 'Right' or 'Left'.</td><td></td><td>Basic 4-wire single P<br>Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.233</td><td>Head</td><td>4 RAMSS requirements</td><td></td><td></td></tr>
<tr><td>Eu.P.2987</td><td>Info</td><td>The requirements for reliability, availability, maintainability, safety and security are specified in [Eu.Doc.20]</td><td></td><td>Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.3244</td><td>Head</td><td>5 Technical Requirements</td><td></td><td></td></tr>
<tr><td>Eu.P.3245</td><td>Info</td><td>The generic technical requirements are specified in [Eu.Doc.20]</td><td></td><td>Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.3246</td><td>Head</td><td>5.1 Specific technical interface requirements</td><td></td><td></td></tr>
<tr><td>Eu.P.3247</td><td>Head</td><td>5.1.1 Interface to the Point of Service Signalling (PoS-Signalling)</td><td></td><td></td></tr>
<tr><td>Eu.P.3248</td><td>Req</td><td>Via the technical interface PoS-Signalling, the data of the functional interface "SCI-P" shall be exchanged with the Subsystem - Electronic Interlocking as specified in [Eu.Doc.92].</td><td></td><td>Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.3249</td><td>Req</td><td>Via the technical interface PoS-Signalling, the data of the functional interface "SMI-P" shall be exchanged with the Subsystem - Maintenance and Data Management as specified in [Eu.Doc.76].</td><td></td><td>Basic non-4-wire single P<br>Basic non-4-</td></tr>
</table>

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| | | | | Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.3250 | Req | Via the technical interface **PoS-Signalling**, the data of the functional interface "SDI-P" shall be exchanged with the Subsystem - Maintenance and Data Management as specified in [Eu.Doc.77]. | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6207 | Req | Via the technical interface **PoS-Signalling**, the data of the functional interface "SSI-P" shall be exchanged with the Subsystem - Security Services Platform as specified in [SP-SEC-ServSpec]. | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.3251 | Head | **5.1.2 Interface to the point machine** | | |
| Eu.P.6271 | Req | The interface to the point machine can be implemented as one of 2 variants:<br>- non-4-wire<br>- 4-wire | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.3252 | Req | The technical requirements for both implementation variants shall be defined by national specifications.<br>Note: In future phases of the System Pillar, national specifications will be replaced by harmonised specifications. | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.4946 | Info | Power property assumptions<br>• The P3 interface may be implemented with separate power supplies for driving and detection.<br>• The power supply for detection is provided by the Subsystem - Point.<br>• A loss of the power supply for detection will report as “Information_No_End_Position” on P3.<br>• When a crank handle is inserted to the Point Machine, typically its power supply for driving is disconnected.<br>• If the point machine monitors its power supply for driving, a loss of the power supply for driving will report as "Information_Ability_To_Move_Point" on P3. (Note: A point machine may not be implemented with the function to monitor its power supply for driving). | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.4941 | Info | P3 is defined as a functional interface, physical properties are not currently defined. This specification is based upon the following assumptions on the properties of the P3 interface. | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.4942 | Req | On a Point machine with detection functionality only, the information objects Moving and Stop_Moving are not available<br>on P3. | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.4943 | Req | The timing behaviour related to multiple Point Machines for a single Point is supplier specific and is not specified by EULYNX as part of the application layer, this shall be handled by the physical implementation. | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.3253 | Head | **5.2 Time behaviour** | | |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.3254 | Req | The time values defined in the chapter Functional requirements specification (Eu.P.2286) shall be configured for the operation of the Subsystem - Point. | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.3262 | Head | **5.2.1 Response times** | | |
| Eu.P.3263 | Req | The Subsystem - Point shall send the corresponding message telegram to the Subsystem - Electronic Interlocking within 250 ms after successful change of state, according to specific use cases. | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.3264 | Req | The Subsystem - Point shall start the reversal operation within 500 ms after receiving a command telegram. | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.3265 | Req | The Subsystem - Point shall start the redrive operation within 500 ms after detecting No end position. | | Option Redrive |
| Eu.P.3255 | Head | **5.3 Configuration and engineering data** | | |
| Eu.P.3256 | Head | **5.3.1 Specific data** | | |
| Eu.P.3257 | Req | The engineering and configuration data for the Subsystem - Point shall include as a minimum the following information: | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.3258 | Req | • The applicable time values defined in chapter Definition of time values (Eu.P.2286). | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.4947 | Req | • The number of implemented Point Machines | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.4948 | Req | • For each configured Point Machine, whether it is implemented with full functionality or as point detector only. The 1st PM must be configured with drive capability. | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.6293 | Req | • Whether the Subsystem - Point is configured to perform redrive | | Option Redrive |
| Eu.P.6294 | Req | • Whether the Subsystem - Point is configured to detect and report unintended position | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |

| ID | Type | Requirement Part 1 | Requirement Part 2 | Func. Pkg. |
|---|---|---|---|---|
| Eu.P.6295 | Req | • For each configured Point Machine, whether it is crucial or non-crucial | | Basic non-4-wire multiple P<br>Basic 4-wire multiple P |
| Eu.P.6636 | Req | • The implementation variant of the interface to the point machine (non-4-wire/4-wire) | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7108 | Req | • Whether the Subsystem - Point is configured to detect and report Ability to Move | | Option Able to move |
| Eu.P.7109 | Req | • Whether the Subsystem - Point is configured to stop driving Point Machines individually or according to Common Drive. | | Option Common Drive |
| Eu.P.3259 | 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 non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.3260 | Req | • configuration data, such as the IP addresses of the Subsystem - Electronic Interlocking , the value of the diagnostic data points with attribute type 'configuration', is not safety-relevant. This data shall be used to calculate the CSNS. | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.3261 | Req | • The remaining configuration data is currently categorised as safety-relevant. This data shall be used to calculate the CSS. | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.4546 | Req | • The engineering data is safety-relevant. This data shall be used to calculate the CSS. | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |
| Eu.P.7244 | Head | **5.3.2 Value configuration** | | |
| Eu.P.7245 | Req | **Con_tmax_Point_Operation**<br><br>The time value shall be configured in accordance with:<br><br>Configurable resolution: steps of 100 ms.<br>Configurable range: from 100 ms up to 30 s<br><br>Con_tmax_Point_Operation is defined in Eu.P.2439. | | Basic non-4-wire single P<br>Basic non-4-wire multiple P<br>Basic 4-wire single P<br>Basic 4-wire multiple P |