



ASAM

Association for Standardization of
Automation and Measuring Systems

ASAM XIL

Generic Simulator Interface

Part 1 of 5

Programmers Guide

Version 2.2.0

Date: 2020-03-13

Base Standard

© by ASAM e.V., 2020

Disclaimer

This document is the copyrighted property of ASAM e.V.
Any use is limited to the scope described in the license terms. The license terms can be viewed at www.asam.net/license

Table of Contents

Foreword	9
1 Introduction	10
1.1 Overview	10
1.2 Motivation	10
1.3 What is Hardware in the Loop Simulation	11
1.4 Technical Approach	12
1.5 Technology Independence	12
2 Relations to Other Standards	14
2.1 Backward Compatibility to Earlier Releases	14
2.2 References to Other Standards	15
2.3 Versioning.....	15
3 General Concepts	16
3.1 XIL Test System Architecture.....	16
3.2 Overview Framework	18
3.2.1 General.....	18
3.2.2 Initialization	18
3.2.3 Framework Variables	18
3.2.4 Acquisition and Stimulation	19
3.3 Overview Testbench.....	22
3.3.1 General.....	22
3.3.2 Ports of the XIL Testbench API	22
3.4 ASAM Data Types.....	23
3.5 Instance Creation	23
3.5.1 Implementation Manifest File	24
3.5.1.1 File content.....	24
3.5.1.2 File Naming convention and storage location	28
3.5.2 Framework and Testbench Factory.....	29
4 Framework	31
4.1 Configuration.....	31
4.1.1 Framework Configuration File	31
4.1.2 Configuring the Framework	35
4.1.3 Shutting down the Framework.....	37
4.1.4 Customize Automatic port management of the framework	38
4.1.5 Object model of the framework configuration	39
4.2 Mapping	41
4.2.1 Overview.....	41
4.2.2 The Mapping XML Schema.....	42

4.2.2.1	Identifier Lists	44
4.2.2.2	Framework Label Group.....	47
4.2.2.3	Identifier Mapping	48
4.2.2.4	String Mapping	50
4.2.2.5	Raster Mapping	51
4.2.2.6	Conversion Tables	51
4.2.2.7	Units	52
4.2.2.8	Computation Tables	54
4.2.2.9	Versioning	55
4.2.3	How a Mapping is used by Test Cases	55
4.2.3.1	The Mapping Info API.....	55
4.2.3.2	Identifier Mapping	57
4.2.3.3	String Mapping	57
4.2.3.4	Raster Mapping	57
4.2.4	Tasks of the Framework.....	57
4.2.4.1	Reading mapping XML files	57
4.2.4.2	Consolidating mapping object trees	57
4.2.4.3	Implementing the Mapping Info API	58
4.2.4.4	Resolve the Mapping during Variable Creation.....	60
4.2.4.5	Perform automatic string mappings	60
4.2.4.6	Error handling.....	60
4.2.5	Consistency Rules	60
4.3	Framework Variables	62
4.3.1	What is a Framework Variable	62
4.3.2	Framework variable Classes	63
4.3.2.1	Scalar Variables	66
4.3.2.2	Vector Variables	67
4.3.2.3	Matrix Variables.....	68
4.3.2.4	Curve Variables.....	69
4.3.2.5	Map Variables	71
4.3.3	Quantities.....	73
4.3.3.1	Scalar Quantity Classes	74
4.3.3.2	Complex Quantity Data Classes	77
4.3.3.3	Usage of Quantities	79
4.3.4	Unit and Physical Dimensions	82
4.3.4.1	Unit Class	84
4.3.4.2	Physical Dimension Class	84
4.3.5	MetaData	84
4.3.6	Calculation with quantity objects	85
4.3.6.1	Overview.....	85
4.3.6.2	Addition and Subtraction	86
4.3.6.3	Multiplication and Division	86
4.3.6.4	Computation Table	87
4.3.6.5	Comparison	90
4.3.7	Data Type Conversion	90
4.3.8	Unit Conversion	90
4.3.9	Usage of Mathematical Operations	91
4.3.9.1	ABSOLUTE and RELATIVE	91
4.3.9.2	NEUTRAL UNIT	91
4.3.9.3	Mathematical and Physical Constants	92
4.3.9.4	Example.....	92
4.4	Measuring	93
4.4.1	Motivation	93
4.4.2	Setting up an Acquisition.....	94
4.4.2.1	Introduction.....	94
4.4.2.2	Using the Acquisition Interface.....	94

4.4.2.3	Acquisition States	96
4.4.2.4	Using the AcquisitionConfiguration Interface	96
4.4.2.5	Principle of Triggered Acquisition.....	98
4.4.2.6	Synchronized Data Acquisition	102
4.4.3	Setting up a Recording.....	105
4.4.3.1	Introduction.....	105
4.4.3.2	Configuring an AcquisitionRecorder.....	106
4.4.3.3	Configuring a Recorder	106
4.4.3.4	Using RecorderResult Readers and Writers.....	107
4.4.3.5	Working with the Recorded Results	109
4.4.3.6	Recorder States	111
4.4.4	Sequence of Acquisition and Recorder Commands	112
4.5	Stimulation.....	114
4.5.1	Motivation	114
4.5.2	Setting up a Stimulation	114
4.5.2.1	Introduction.....	114
4.5.2.2	Using the Stimulation Interface	115
4.5.2.3	Stimulation States	115
4.5.3	Setting up a Player.....	116
4.5.3.1	Introduction.....	116
4.5.3.2	Configuring a Player.....	117
4.5.3.3	Using a Player	118
4.5.3.4	Player States	119
5	Testbench	120
5.1	Common Functionalities.....	120
5.1.1	ValueContainer	121
5.1.1.1	Overview.....	121
5.1.1.2	General Value Container Classes.....	121
5.1.1.3	Specific Value Container Classes	123
5.1.2	Document Handling.....	124
5.1.3	Script.....	125
5.1.3.1	Overview.....	125
5.1.3.2	States of Script	126
5.1.3.3	Script Parameters.....	130
5.1.4	TargetScript	131
5.1.4.1	Overview.....	131
5.1.4.2	Parameters	132
5.1.4.3	Custom Properties.....	132
5.1.4.4	Usage of TargetScript	132
5.1.5	Signal Descriptions	134
5.1.5.1	General Remarks about Segment-Based Signals	137
5.1.5.2	Signal Segments	140
5.1.5.3	Converting SignalDescriptions to sample based representations ...	160
5.1.5.4	Using Signal Descriptions	161
5.1.5.5	Signal Description File.....	167
5.1.5.6	Usage of Parameterized SignalDescriptions.....	168
5.1.6	SignalGenerator	171
5.1.6.1	Interpretation of SignalDescriptions by the SignalGenerator	172
5.1.6.2	Parameters	176
5.1.6.3	Custom Properties.....	176
5.1.6.4	Usage of SignalGenerator (Stimulating Model Variables)	176
5.1.7	Document Handling for SignalGenerator and SignalDescriptionSet ...	178
5.1.8	Watcher	181
5.1.8.1	General.....	181
5.1.9	Duration	182

5.1.10	Meta Info.....	183
5.1.10.1	Metadata on Variables	183
5.1.10.2	Metadata on Conversion Methods	185
5.1.10.3	Metadata on Acquisition Rasters / Processor Tasks	187
5.1.11	Variable Ref and Value Representation Mode.....	188
5.1.12	Data Capturing	190
5.1.12.1	Introduction.....	190
5.1.12.2	Capture Configuration and Control	190
5.1.12.3	Untriggered Capturing.....	196
5.1.12.4	Triggered Capturing	197
5.1.12.5	Re-triggered Capturing.....	199
5.1.12.6	Obtain Capture Results from a Capture.....	201
5.1.12.7	Capture Result	202
5.1.12.8	Relation between Triggers, Capture State, Capture Events and Data Frames.....	206
5.1.12.9	Reader and Writer for CaptureResult.....	215
5.1.12.10	Usage examples for Capture and Capture Result	216
5.2	Model access Port.....	227
5.2.1	User Concept	227
5.2.1.1	General.....	227
5.2.1.2	MAPort Interface	227
5.2.1.3	States of the MAPort	229
5.2.2	Usage of MAPort.....	231
5.2.2.1	Creation and Configuration	231
5.2.2.2	Obtaining Information on Available Model Variables, Tasks and their Properties	233
5.2.2.3	Reading & Writing Model Variables	236
5.2.2.4	State dependency of Variables' Writeability and Readability.....	240
5.2.2.5	Relation between MAPort and Capturing / SignalGenerator	241
5.2.2.6	Pausing and stepwise execution of the simulation	242
5.3	Diagnostic Port.....	246
5.3.1	Overview.....	246
5.3.2	API.....	247
5.3.2.1	Communication Modes.....	248
5.3.2.2	ECU.....	248
5.3.2.3	Functional Groups	249
5.3.3	States of the DiagPort	249
5.3.4	Usage of this Port	251
5.3.4.1	Creation and Configuration	251
5.3.4.2	Getting The Ecu Object.....	252
5.3.4.3	Reading And Clearing The Fault Memory.....	253
5.3.4.4	Reading The Variant Coding Data	254
5.3.4.5	Reading Identification Data	254
5.3.4.6	Reading and Writing Values from and to the Eeprom by Alias Names	255
5.3.4.7	Reading from the Eeprom	256
5.3.4.8	Writing to the Eeprom.....	256
5.3.4.9	Implicit and Explicit Communication.....	257
5.3.4.10	Sending Hex Services with Explicit Communication	257
5.3.4.11	Executing Jobs	258
5.3.4.12	Reading Data from a Functional Group	259
5.3.4.13	Using the Basecontroller	260
5.3.5	Special Hints	260
5.3.5.1	Structure of Returned Collections	260
5.3.5.2	States in the Diagnostic Tool	260
5.4	ECUMPort	261

5.4.1	Overview.....	261
5.4.2	States of the ECUMPort.....	264
5.4.3	Usage of ECUMPort.....	266
5.4.3.1	Creation and Configuration	266
5.4.3.2	Getting Lists of Variable and Task Names.....	267
5.4.3.3	Read a Scalar Variable Value and its Properties	268
5.4.3.4	Read an Array Variable Value and its Properties	268
5.4.3.5	Read a Matrix Variable Value and its Properties	269
5.4.3.6	Capturing ECU Variables	270
5.5	ECUCPort.....	274
5.5.1	Overview.....	274
5.5.2	States of the ECUCPort	275
5.5.3	Usage of ECUCPort.....	277
5.5.3.1	Creation and Configuration	277
5.5.3.2	Accessing ECU Parameters.....	278
5.5.3.3	Getting the List of Variables of the ECUCPort	279
5.5.3.4	Manage ECU Memory Pages	280
5.6	EES Port.....	282
5.6.1	User Concept.....	282
5.6.1.1	General.....	282
5.6.1.2	Configuration and Execution of Electrical Errors	284
5.6.1.3	Triggers in EES	286
5.6.1.4	Electrical Errors	286
5.6.1.5	API.....	289
5.6.1.6	States of the EES Port	295
5.6.2	Usage of this Port	298
5.6.2.1	Port Creation and Configuration.....	298
5.6.2.2	Creating Error Configurations	299
5.6.2.3	Error Stimulation.....	302
5.6.2.4	Extension of Error Stimulation.....	302
5.6.2.5	Creating Error Objects.....	304
5.6.2.6	Loading Error Configurations from File	308
5.6.3	Special Hints	309
5.6.3.1	EES Hardware Limitations and Extensions.....	309
5.7	Network Port.....	310
5.7.1	User Concept.....	310
5.7.1.1	General.....	310
5.7.1.2	Network Port.....	311
5.7.1.3	States of the Network Port	312
5.7.2	Usage of the Network Port	314
5.7.2.1	Port Creation and Configuration.....	314
5.7.2.2	Object Model	315
5.7.2.3	Mapping to DBC and FIBEX	317
5.7.2.4	Navigating the Object Model	318
5.7.2.5	Send and Receive Frames.....	319
5.7.2.6	Send and Receive Signals	322
5.7.2.7	Capturing of Bus Signals.....	324
5.7.2.8	Capturing of Bus Frames	327
5.7.2.9	Replay of Bus Frames.....	330
5.7.2.10	Extending the CAN object model	331
6	Symbols and Abbreviated Terms	334
7	Bibliography	335