



Association for Standardisation of
Automation and Measuring Systems

ASAM GDI

Generic Device Interface

Part 1 of 3

Programmers Guide Base

Version 4.5.0

Date: 2011-01-31

Base Standard

© by ASAM e.V., 2011

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

1	Foreword	13
2	Scope	15
3	Terms and definitions	17
4	Symbol and abbreviated terms	25
4.1	Abbreviations	25
4.2	Relations	26
4.3	Legend for used hierarchical graphics	27
5	Compatibility	29
5.1	GDI Components	29
5.2	GDI XSD format	29
6	Principle of GDI	31
6.1	Component Overview	34
6.1.1	Coordinator	34
6.1.2	GDI Device Driver	35
6.1.3	The Device Capability Description	35
6.1.4	Platform Adapter	37
6.1.5	Platform Adapter Extensions	38
6.1.6	The ASAM GDI Device	38
6.2	Life Cycle Mechanism and Possibilities	40
6.2.1	The Parameterization Phase	41
6.2.2	The Setup Phase	41
6.2.3	The Application Phase	42
6.2.4	The Remove Phase	43
7	Platform Adapter and Platform Adapter Extensions	45
7.1	Overview	45
7.1.1	Platform Adapter Functions	45
7.1.2	Platform Adapter Extension Functions	48
7.2	Data Structures of the Platform Adapter	49
7.2.1	Configuration data, IO_CONFDAT	50
7.2.2	Status of periphery interface, IO_STAT	51
7.2.3	Current System Time, OS_UCT	51
7.3	Concurrent Program Threads, Timers and Protective Mechanisms	52
7.3.1	Program Threads	52
7.3.2	Requirements for the Implementation	52
7.3.3	Scheduling and Priorities	52
7.3.4	Identification	52
7.3.5	Timers	52
7.3.6	Semaphores	53
7.3.7	Mutex	53

7.4	Platform Adapter Callback Functions	54
7.4.1	io_complete (Type: PA_CBP_CPL)	54
7.4.2	io_event, (Type: PA_CBP_EVENT)	55
7.5	Platform Adapter API Functions	56
7.5.1	getFuncAddress	56
7.5.2	io_cancel	57
7.5.3	io_clear	58
7.5.4	io_close	59
7.5.5	io_conclude	60
7.5.6	io_config	61
7.5.7	io_execute	62
7.5.8	io_initiate	64
7.5.9	io_open	66
7.5.10	io_read	68
7.5.11	io_stat	70
7.5.12	io_write	71
7.5.13	os_allocate	73
7.5.14	os_clock	74
7.5.15	os_closeDebug	75
7.5.16	os_createMutex	76
7.5.17	os_createSem	77
7.5.18	os_delay	78
7.5.19	os_deleteMutex	79
7.5.20	os_deleteSem	80
7.5.21	os_free	81
7.5.22	os_getLPNumber	82
7.5.23	os_killLPTimer	83
7.5.24	os_killtimer	84
7.5.25	os_openDebug	85
7.5.26	os_reallocate	86
7.5.27	os_releaseMutex	87
7.5.28	os_releaseSem	88
7.5.29	os_setLPTimer	89
7.5.30	os_settimer	90
7.5.31	os_time	91
7.5.32	os_time_a	92
7.5.33	os_waitMutex	93
7.5.34	os_waitSem	94
7.5.35	os_writeDebug	95
7.6	Platform Adapter Extension Callback Functions	96
7.6.1	ext_complete, (Type: PA_CBP_CPL)	96
7.6.2	ext_event, (Type: PA_CBP_EVENT)	97
7.7	Platform Adapter Extension API Functions	98
7.7.1	ext_cancel	98
7.7.2	ext_clear	99
7.7.3	ext_close	100
7.7.4	ext_conclude	101
7.7.5	ext_config	102
7.7.6	ext_execute	103
7.7.7	ext_initiate	105
7.7.8	ext_open	107
7.7.9	ext_read	109
7.7.10	ext_stat	111
7.7.11	ext_write	112

7.8	Return Codes of Platform Adapter Functions	114
8	Device Driver API	119
8.1	Overview DD API Functions	119
8.2	Data Structure of the Device Driver	120
8.2.1	Status of the Communication (Type GDIRESULT)	120
8.2.2	Manufacturer's description of the VD (Type GDIIDENT)	121
8.2.3	Status of a VD (Type GDISTATUS)	122
8.3	Device Driver (Callback Functions)	123
8.3.1	GDI_Complete	124
8.3.2	GDI_Accept	125
8.3.3	GDI_InformationReport	126
8.4	Device Driver API Functions	127
8.4.1	GDI_Abort	128
8.4.2	GDI_Attach	129
8.4.3	GDI_Cancel	130
8.4.4	GDI_Conclude	131
8.4.5	GDI_CreateCommObject	132
8.4.6	GDI_CreateFuncObject	133
8.4.7	GDI_DeleteCommObject	134
8.4.8	GDI_DeleteFuncObject	135
8.4.9	GDI_Execute	136
8.4.10	GDI_Identify	137
8.4.11	GDI_Initiate	138
8.4.12	GDI_Read	139
8.4.13	GDI_Status	140
8.4.14	GDI_Write	141
8.5	Error Messages of the Device Driver	142
8.5.1	Overview	142
8.5.2	Definition of invocation errors	144
8.5.3	Definition of process errors	146
8.5.4	Definition of execution errors	148
8.5.5	Definition of application errors	159
8.5.6	Definition of DIP errors	159
9	IDL based DCD, DIT, DII	161
9.1	Naming Conventions	161
9.1.1	Rules	161
9.1.2	Prefixes for DCD	162
9.2	Structure Overview DCD	163
9.3	Description Elements DCD	164
9.3.1	Type Declarations	164
9.3.2	Constants Declarations	168
9.3.3	Module Declarations	168
9.3.4	Interface Declarations	170
9.4	Predefined Mandatory (Essential) Elements	178
9.4.1	Device Base Function	178
9.4.2	Transition Function	179
9.4.3	Header	181
9.4.4	Description of the "gdi_dcd.h"	182

9.5	Structure Overview DIT / DII	182
9.6	Description Elements DIT / DII	185
9.6.1	Area Declaration (Branch Distributor)	185
9.6.2	Area Definition	186
9.7	Access from DCD to DIT/DII	187
9.7.1	Examples	188
9.8	Backus Naur Form	188
9.8.1	General Definitions	188
9.8.2	Definition of Preprocessor Syntax	189
9.8.3	Definition of DCD Syntax	191
9.8.4	Definition of DIT / DII Syntax	197
9.9	Reference of error messages to the DIT/DII	199
10	XSD based DCD, DIT	201
10.1	XSD DCD	201
10.1.1	Base Classes	201
10.1.2	Description of the Classes Contained in GDICommon	202
10.1.3	General Rules for the Derivation of a GDI – Scheme from Pre-defined Classes	207
10.1.4	GDI_Header	208
10.1.5	DCPT Header	209
10.1.6	Mapping of DCD Specifics to GDI XSDs	211
10.2	XSD DIT	218
10.3	Usage of ASAM ASAM Data Types	218
10.3.1	Differences Between the GDI Presentation and the XSD Presentation	219
11	Modelling	221
11.1	Principle	221
11.2	Guideline	224
11.2.1	Purpose of a DCD	224
11.2.2	Point of view	224
11.2.3	Device Function selection	224
11.2.4	Function identification	226
11.3	Companions and Profiles	227
11.3.1	Usage of Companions	229
11.4	Style Guide	230
11.4.1	Device Profiles	230
11.4.2	Device Independent Profiles	231
11.5	Predefined Service Functions	231
11.5.1	ControlCallback	232
11.5.2	Domain	235
11.5.3	Synchronization	241
11.5.4	ProgramControl	252
11.5.5	Inspection	255
11.5.6	Device Communication Parameters	256
11.5.7	Device Image	258
11.5.8	Device Presence	259
11.6	DIP	260

11.6.1	Introduction	260
11.6.2	DIP and Device Profiles	260
11.6.3	The Channel Principle	262
11.6.4	The DIP State Model	262
11.6.5	Relation between DIP State Model and VD Operating State	264
11.6.6	DIP State Description	265
11.6.7	DIP Channel	266
11.6.8	DIP Group	272
11.6.9	Data transport within DIP	278
11.6.10	DIP ErrorCoding	278
11.7	DCD design by using the DIP	280
11.7.1	DIP within a DCD	280
11.7.2	Purpose	281
11.7.3	Finding the Device capabilities	282
11.7.4	Modelling of Function exceeding Characteristics	283
11.7.5	Assignment of the Defined Device Capabilities	283
12	Coordinator/ Driver Management	285
12.1	Data Types	285
12.2	Calling Conventions of Device Driver API and PA API	287
12.3	Structmember Alignment	287
12.4	Memory Management	287
12.4.1	General Procedure	293
12.4.2	Heap Memory used by the Device Driver	295
12.4.3	Limited and unlimited sequences	295
12.5	Synchronous and Asynchronous Communication	299
12.6	Version Handling	300
12.6.1	DCD Versions	300
12.6.2	Device Driver Versions	300
12.6.3	Platform Adapter Versions	301
12.6.4	Handling Platform Adapter Extension Versions	302
12.6.5	Include hierarchie and naming conventions	304
12.7	Operating state	305
12.8	Application Callback	307
12.8.1	Goal	307
12.8.2	DCD	308
12.8.3	Cooperation between Application and Coordinator for the Registering of the Application Object	309
12.8.4	Cooperation between Coordinator and Device Driver for the Log In of the Application Object	309
12.8.5	Cooperation between Driver and Coordinator for the Execution of an Operation within the Application	310
12.8.6	Cooperation between Coordinator and Application for the Execution of an Operation within the Application	310
12.8.7	Write/Read of COs of a FO Existing Within the Application	310
12.9	Sequence Diagrams	311
12.9.1	Example DCD for Sequence Diagrams	311
12.9.2	Create Instances of Device Driver	313
12.9.3	Destroy Instances of Device Driver	315
12.9.4	Working with Instances of Device Driver	316
12.9.5	GDI_InformationReport and GDI_Accept	317

12.9.6	Dynamic Function Objects	318
12.9.7	Application Call Back	319
12.9.8	Provide Platform Adapter API to Device Driver	320
12.9.9	Connect to Devices	321
12.9.10	Working with Devices	322
12.9.11	Disconnect Devices	324
13	Coordinator Services	325
13.1	Philosophy	325
13.1.1	Objective	325
13.1.2	General	327
13.2	Parameterization	327
13.3	Configuration	328
13.3.1	PID based configuration	329
13.3.2	Service based and mixed configuration	329
13.4	Coordinator Structure	329
13.4.1	Workspace	329
13.4.2	Model Structure	333
13.5	Details	339
13.5.1	Workspace Structure	339
13.5.2	Callback	341
13.5.3	Extend Workspace	342
13.5.4	Workspace Transfer	343
13.5.5	Data Monitoring	344
13.5.6	Datatypes	346
13.5.7	Streaming	348
13.6	Error	352
13.7	Sequence Diagrams	355
13.8	Basis and Profile Coordinators	365
13.8.1	Basis Coordinator	365
13.8.2	Profile Coordinators	365
14	Example	367
14.1	Measuring Device	367
14.2	DIP Example	373
14.2.1	Overview	373
14.2.2	Description of the Automation Task	373
14.2.3	Device Descriptions	374
14.2.4	Error Messages of the Fuel Balance	375
14.2.5	Parameterization	375
14.2.6	Device Capability Descriptions and Device Info Texts	378
14.2.7	Testing Sequence	397
15	Appendix	403
A	Mapping mechanism of platform adapter	403
B	Country dependent Language Coding	405
C	Relation between Coordinator, PID and Interface Type	407
D	Stream Mapping	411

E	Byte Order	415
F	Coordinator Error Codes	417
G	Alignment	419
H	Library Functions restrictions	423
I	Abstract example for memory management of sequences	425
Index		429
Figuredirectory		431
Tabledirectory		435
Books		437

1 Foreword

ASAM GDI makes it possible to connect devices and subsystems simply and smoothly with test automation systems.

GDI uses the following basic structure for documents:

- Base Standard
- Companions (e.g. MDAQ, ChasisDyno, Crash, MCD3)
- Transport Layer Communication Types (e.g. IP, COM, LPT, USB, SOFTSYNC)

The Base standard consist of 2 parts:

- Part 1: Common definition of interfaces and description syntax for all layers
- Part 2: Parameterization Instance Description (PID) [ASAM GDI PID]

The full content of GDI 4.5.0 is available in the ISO 20242 "Industrial automation systems and integration — Service Interface for Testing Applications" with the following parts:

- Part 2: Resource Management Service Interface (content of platform adapter and platform adapter Extension) [ISO 20242-2]
- Part 3: Virtual Device Service Interface (content of Device Driver API) [ISO 20242-3]
- Part 4: Device capability profile template (content of DCD und PID) [ISO 20242-4]
- Part 5: Application Program Service Interface (content of Coordinator Services) [ISO 20242-5]

At the moment the Predefined Service Functions and DIP are not reflected in the ISO standard, as they are specific modeling concepts, which are covered by the creation of ASAM GDI Companions.