

Device Driver Skeleton

Benutzerdokumentation



**rd electronic GmbH
Zweigstelle Dresden
Bernhardstraße 70
01187 Dresden**

**Tel. +49 351-6563-56-4
Fax +49 351-6563-56-5**

Internet: <http://www.rd-electronic.de>

Dokumentnummer	:	DD-Skeleton-UserMan
Ausgabe	:	1.0.0.0
Status	:	Release
Letzte Änderung	:	30.03.2021
Erstellung	:	27.04.2018
Autoren	:	JWo, SBa

Rechtliche Hinweise

DIESES DOKUMENT ENTHÄLT INFORMATIONEN, WELCHE DURCH DAS URHEBERRECHT GESCHÜTZT SIND. KEIN TEIL DIESES DOKUMENTES DARF OHNE SCHRIFTLICHE GENEHMIGUNG DES AUTORS IN IRGEND EINER FORM, ALS FOTOKOPIE, MIKROFILM ODER MIT EINEM ANDEREN VERFAHREN, AUCH NICHT FÜR ZWECKE DER UNTERRICHTSGESTALTUNG, REPRODUZIERT ODER UNTER VERWENDUNG ELEKTRONISCHER SYSTEME VERARBEITET, VERVIELFÄLTIGT ODER VERBREITET WERDEN.

DER AUTOR BEHÄLT SICH DAS RECHT VOR, DIE IN DIESEM DOKUMENT ENTHALTENEN INFORMATIONEN OHNE ANKÜNDIGUNG ZU ÄNDERN. DER ANWENDER IST VERPFLICHTET, SICH ÜBER DEN GÜLTIGEN STAND ZU INFORMIEREN.

OBWOHL BEI DER ERSTELLUNG DIESES DOKUMENTES MIT GRÖSSTER SORGFALT VORGEGANGEN WURDE, SIND FEHLER DENNOCH NICHT GANZ AUSZUSCHLIESSEN. AUS DIESEM GRUND ÜBERNIMMT DER AUTOR KEINE HAFTUNG FÜR FEHLER, DIE IN DIESEM DOKUMENT ENTHALTEN SIND ODER FÜR ZUFÄLLIGE ODER FOLGESCHÄDEN IM ZUSAMMENHANG MIT DER ANWENDUNG DIESES DOKUMENTES.

Änderungsübersicht

Version	Datum	Autor	Änderung
0.0.0.0	27.04.2018	SBa	<ul style="list-style-type: none">• Neuerstellung
1.0.0.0	30.03.2021	SBa	<ul style="list-style-type: none">• Überarbeitung, Release

Inhaltsverzeichnis

RECHTLICHE HINWEISE	1
ÄNDERUNGSÜBERSICHT	2
INHALTSVERZEICHNIS	3
ABBILDUNGSVERZEICHNIS	4
TABELLENVERZEICHNIS	5
1 ALLGEMEINES	6
1.1 IDENTIFIKATION	6
1.2 ZIEL	6
1.3 GINA2010-DOKUMENTE	6
1.4 REFERENZIERTER DOKUMENTE	7
1.5 DEFINITIONEN UND ABKÜRZUNGEN	9
2 EINLEITUNG	12
2.1 WAS IST EIN DEVICE DRIVER GERÜST	12
2.2 BESCHAFFUNG UND LIZENZEN	12
2.3 ASAM-GDI VORKENNTNISSE	12
3 VERSCHIEDENE BETRIEBSSYSTEME	13
3.1 WINDOWS	13
3.1.1 Alignment an der Platform Adapter API	13
3.1.2 Alignment für Strukturen aus Communication Types	13
3.1.3 Alignment an der Device Driver API	14
3.2 LINUX	14
3.2.1 Alignment an der Platform Adapter API	14
3.2.2 Alignment für Strukturen aus Communication Types	15
3.2.3 Alignment an der Device Driver API	15
4 DEVICE DRIVER SKELETON	16
4.1 GEMEINSAME DATEIEN	16
4.1.1 Dateien des ASAM GDI Standards	16
4.1.2 Dateien des RDE Platform Adapter 4.5	17
4.1.3 Dateien des RDE Trace Moduls	18
4.1.4 Dateien für Device Drivers 4.4	18
4.2 GENERIERTE DATEIEN	18

Abbildungsverzeichnis

Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.



Tabellenverzeichnis

TABELLE 1: BENÖTIGTE HEADER-DATEIEN	17
---	----

1 Allgemeines

1.1 Identifikation

Dieses Dokument beschreibt die Benutzung von mittels [GINA2010](#) [RDE-Doc1] generierten [Device Driver Skeletons](#) der Firma [RDE](#).

1.2 Ziel

Das Dokument gibt dem Entwickler eines [Device Drivers](#) Hilfestellung bei der Befüllung eines mittels [Skeleton Generator](#) der [RDE](#) generierten [Device Driver Skeleton](#). Der Entwickler soll so in die Lage versetzt werden, ein [Device Driver Skeleton](#) zu einem vollfunktionalen [Device Driver](#) auszubauen, welcher die Kommunikation mit seiner Anlage/Gerät umsetzt.

1.3 GINA2010-Dokumente

[RDE-Doc1] **"GINA2010 - Benutzerdokumentation "**

Datei : [GINA2010-UserMan.pdf](#)

Version : 2.0.0.1

Datum : 27.04.2018

[RDE-Doc2] **"Abspeicherung von IR Daten"**

Datei : [GINA2010-UserMan-IrData.pdf](#)

[RDE-Doc3] **"GINA2010 - Reporterstellung"**

Datei : [GINA2010-UserMan-Report.pdf](#)

Version : 1.1.0.0

Datum : 03.12.2014

[RDE-Doc4] **"Device Driver Skeleton - Benutzerdokumentation"**

Datei : [GINA2010-UserMan-Skeleton.pdf](#)

Version : 1.0.0.0

Datum : 30.03.2021

[RDE-Doc5] "GINA2010 - Makrosprache"Datei : [GINA2010-UserMan-Macro.pdf](#)

Version : 1.2.0.0

Datum : 05.03.2014

1.4 Referenzierte Dokumente

[GDI-Doc1] "Specification of the Device Capability Description of an ASAM-Device Driver"Datei : [Gdi-Part_A_Ver420_en.pdf](#)

Titel : ASAM-GDI Part A

Version : 4.20

Datum : 01.02.2001

[GDI-Doc2] "Specification of the Interface of an ASAM-Device Driver"Datei : [Gdi-Part_B_Ver420_en.pdf](#)

Titel : ASAM-GDI Part B

Version : 4.2

Datum : 02.02.2001

[GDI-Doc3] "Generic Device Interface Specification - Version 4.3.2 - Specification"Datei : [ASAM_GDI_4_3_2_Specification.pdf](#)

Version : 4.3.2 (Maintenance Release)

Datum : 15.02.2005

[GDI-Doc4] "ASAM GDI - Version 4.4.0 - Specification"Datei : [ASAM_GDI_V4.4.0_Specification.pdf](#)

Version : 4.4.0 (Release)

Datum : 31.01.2008

[GDI-Doc5] "ASAM GDI – Generic Device Interface – Part 1 of 3"Datei : [ASAM_CAT_GDI_BS_1_3_Programmers-Guide-Base_V4-5-0.pdf](#)

Version : 4.5.0

Datum : 31.01.2011

[GDI-Doc6] "ASAM GDI – Transport Layer Communication Type COM"Datei : [ASAM_GDI_Transport-Layer-Communication-Types-COM_V3-0-0.pdf](#)

Version : 3.0.0

Datum : 08.08.2008

[GDI-Doc7] "ASAM GDI – Transport Layer Communication Type IP4"Datei : [ASAM_GDI_Transport-Layer-Communication-Types-IP4_V3-0-0.pdf](#)

Version : 3.0.0

Datum : 08.08.2008

1.5 Definitionen und Abkürzungen

Abstract Interface	Nicht instanzitierbares vererbbares Interface [GDI-Doc3]
Accept	Callback-Mechanismus auf Communication Objects zur asynchronen, durch einen Device Driver angeforderten, Übertragung von Daten aus der Anwendung an den Device Driver
API	Application Programmers Interface, Programmierschnittstelle einer Softwarekomponente
ASAM	Association of Standardisation of Automation and Measuring Systems
Attribute	DCD -Typ für Communication Objects zum Datenaustausch in Function Objects
Betriebsmodus	Der Modus enthält die zwei Ausprägungen „Automatisch“ zur automatischen Bedienung der Transition Functions durch die Coordinator Engine und „Manuell“ zur Bedienung der Transition Functions durch den Benutzer
Boolean	ASAM GDI Datentyp [GDI-Doc3]
Char	ASAM GDI Datentyp [GDI-Doc3]
Create Parameter	Spezieller Parameter (Communication Object) zum einmaligen Parametrieren von Function Objects
Communication Object	DCD -Typ bzw. Instanz eines solchen nach ASAM GDI zum Austausch von Informationen zwischen Anwendung und Device Driver [GDI-Doc3] Auch als Attribute bezeichnet
Communication Type	Transport Layer Communication Type (COM [GDI-Doc6], IP4 [GDI-Doc7])
Companion	Spezifikation eines Geräteprofils nach ASAM GDI [GDI-Doc3]
Coordinator	ASAM GDI Coordinator [GDI-Doc3] der RDE mit spezieller Anwendungsschnittstelle für die Volkswagen AG
Coordinator Engine	In GINA2010 integrierte Kernkomponente zur Ansteuerung von Device Drivers nach dem ASAM GDI Standard Die Coordinator Engine findet auch im Coordinator der RDE Einsatz.
Control VD	Virtual Device zur Steuerung der GDI Phasen [GDI-Doc3]
Double	ASAM GDI Datentyp [GDI-Doc3]
DCD	Device Capability Description, Datei, die die Fähigkeiten eines Gerätes abstrahiert beschreibt. Enthält Informationen über die instanzitierbaren Objekte und definierte Datentypen sowie Konstanten [GDI-Doc3]
Device Base Function	Function Object des Control VD mit optionalen Operationen

Device Driver	ASAM GDI Gerätetreiber (Device Driver) [GDI-Doc3]
Device Function	Alternative Bezeichnung für Interface
DIT	Device Info Text, Datei, die verschiedene, beschreibende Strings enthält. Sie ist eine Ergänzung zur DCD und kann in verschiedenen Sprachen vorliegen [GDI-Doc3]
DLL	Dynamic Linked Library, Bezeichnung einer dynamischen Bibliothek unter Windows mit der Dateiendung *.dll (analog DSO unter Linux)
DSO	Dynamic Shared Object, Bezeichnung einer dynamischen Bibliothek unter Linux/Unix mit der Dateiendung *.so (analog DLL unter Windows)
Enumeration	ASAM GDI Datentyp [GDI-Doc3]
Enumerator	Spezielle Ausprägung innerhalb einer Enumeration [GDI-Doc3]
Float	ASAM GDI Datentyp [GDI-Doc3]
Function Reference	DCD-Typ zur Referenzierung eines Function Object [GDI-Doc3]
Function Object	Instanz des DCD-Typs Interface [GDI-Doc3]
GDI	Generic Device Interface, ASAM GDI Standard [GDI-Doc1], [GDI-Doc2], [GDI-Doc3], [GDI-Doc4], [GDI-Doc5]
GDI Phase	Zustand nach ASAM GDI Phasenmodell [GDI-Doc3]
Information Report	Callback-Mechanismus auf Communication Objects zur asynchronen Übertragung von Daten vom Device Driver zur Anwendung
Initiate Parameter	Spezieller Parameter (Communication Object) zum initialen Parametrieren von Function Objects
Interface	DCD-Typ nach ASAM GDI als Container für Communication Objects und Operations [GDI-Doc3] Auch als Device Function bezeichnet
LicManClient	Lizenzierungstool der RDE zur Aktivierung von Einzelplatzlizenzen
Limited Sequence	In ihrer Länge begrenzte Sequence [GDI-Doc3]
Long	ASAM GDI Datentyp [GDI-Doc3]
Module	DCD-Typ nach ASAM GDI als Container für Interfaces [GDI-Doc3]
Octet	ASAM GDI Datentyp [GDI-Doc3]
Operation	DCD-Typ nach ASAM GDI zum Ausführen von Aktionen an Anlagen und Geräten
PA	Platform Adapter
Parameter	DCD-Typ für Communication Objects zur Parametrierung von Function Objects

Platform Adapter	ASAM GDI Bibliothek zur Nutzung für plattformunabhängige Device Driver . Besitzt eine standardisierte Schnittstelle und abstrahiert systemspezifische Funktionalitäten [GDI-Doc3]
Platform Adapter Extension	ASAM GDI Platform Adapter Extension [GDI-Doc3]
RDE	Firma rd electronic GmbH
Selector	Bestandteil des ASAM GDI Datentyps Union [GDI-Doc3]
Sequence	ASAM GDI Datentyp [GDI-Doc3]
Short	ASAM GDI Datentyp [GDI-Doc3]
Skeleton Generator	In GINA2010 integrierte Komponente zur Erstellung von Device Driver Gerüsten nach dem ASAM GDI Standard (zusätzliche Lizenz erforderlich)
Standard Extension	Platform Adapter Extension mit Communication Types for Ethernet und serielle Verbindungen der Firma RDE
Struktur	ASAM GDI Datentyp [GDI-Doc3]
Transition	Umschaltung in eine andere GDI Phase
Transition Function	Function Object des Control VD mit Operationen zur Schaltung der GDI Phasen
ULong	ASAM GDI Datentyp [GDI-Doc3]
Union	ASAM GDI Datentyp [GDI-Doc3]
Unlimited Sequence	In ihrer Länge unbegrenzte Sequence [GDI-Doc3]
UShort	ASAM GDI Datentyp [GDI-Doc3]
VD	Virtual Device
Virtual Device	Instanz des DCD -Typs Module . Repräsentiert einen bestimmten Aspekt eines Gerätes, das über einen Device Driver angesprochen wird [GDI-Doc3]

2 Einleitung

2.1 Was ist ein Device Driver Gerüst

Ein **Device Driver Skeleton** ist ein mittels **GINA2010** über eine **DCD** generiertes C++ -Projektset für Microsoft Visual Studio 6.0 und UNIX-Systeme, welches den **ASAM GDI** Standard für **Device Drivers** implementiert. **Device Driver Skeletons** sind für den Anlagen-/Gerätespezifischen Ausbau durch Anlagenhersteller zu vollwertigen **Device Drivers** konzipiert.

2.2 Beschaffung und Lizenzen

RDE generiert generell **Device Driver Skeletons** auf Anfrage und bietet zu deren Befüllung Unterstützungsleistungen für einen reibungsarmen und schnellen Projektablauf.

Darüber hinaus bietet **RDE** den Erwerb von **GINA2010** mit integriertem **Skeleton Generator**, um Anlagenhersteller zu befähigen, **Device Driver Skeletons** selbst zu erstellen.

Der Erwerb von **GINA2010** mit oder ohne integriertem **Skeleton Generator** kann dabei in verschiedenen Lizenzformen erfolgen.

Üblicherweise werden hier Einzelplatz-Lizenzen angeboten, welche an spezielle Rechner gebunden und bedarfsgerecht zur Kostenreduktion zeitlich beschränkt werden.

Umfassende Informationen zu Lizenzen und Preislisten erhalten Sie auf eine Anfrage an license.support@rd-electronic.de.

2.3 ASAM-GDI Vorkenntnisse

Es ist notwendig, sich vor der Entwicklung eines **Device Driver** mit den Konzepten und Möglichkeiten von **ASAM GDI** vertraut zu machen. Als Referenz sollte die Spezifikation zu **ASAM GDI 4.3.2** [GDI-Doc3] herangezogen werden. Für ältere **Device Driver** kann es auch sinnvoll sein, die **ASAM GDI 4.2** Spezifikationen [GDI-Doc1] und [GDI-Doc2], für neuere **Device Driver** die Spezifikationen zu **ASAM GDI 4.4** [GDI-Doc4] und zu **ASAM GDI 4.5** [GDI-Doc5] heranzuziehen.

3 Verschiedene Betriebssysteme

Device Drivers sind entsprechend **ASAM GDI** Standard plattformunabhängig zu implementieren. Dennoch gibt es bezüglich der Projekt-Dateien und der einstellbaren Alignments ein paar Beachtungen.

3.1 Windows

Für das Betriebssystem Windows wird eine Projektdatei (*.dsp) für Microsoft Visual Studio 6.0 generiert. Diese wird entsprechend im Verzeichnispfad „\Driver\Project_Win32\“ relativ zum Verzeichnispfad der **Device Driver Skeleton** - Konfigurationsdatei (*.cfg) abgelegt.

Die MS-VS-Projektdatei kann nach Generierung auf höhere MS Studio Versionen gehoben werden.

Device Driver Skeletons sind derzeit für die Windows-Systeme NT4, 2000, XP, 2003, 7, 8 und 10 erstellbar. Ältere NT-Systeme sowie Windows 95, 98, ME werden nicht unterstützt.

3.1.1 Alignment an der Platform Adapter API

Das Alignment für Strukturen, welche an der **Platform Adapter** API durch Einbindung der Header-Datei `gdi_pa.h` und der damit implizit eingebundenen Header-Dateien festgelegt werden, wird mittels der Präprozessor-Direktive `PA_API_ALIGNMENT` in `Driver/Source/Settings.h` eingestellt.

Die API eines **Platform Adapter** der **RDE** ist immer mit einem Alignment von 8 übersetzt.

3.1.2 Alignment für Strukturen aus Communication Types

Communication Type Strukturen sind Strukturen aus **Platform Adapter Extensions**, welche weitere Arten von Kommunikations-Kanälen oder anderweitigen zusätzlichen Systemressourcen verfügbar machen. Sie werden an den API-Funktionen von **Platform Adapters** und **Platform Adapter Extensions** durch das generische Typkonstrukt `void*` beschrieben.

Das Alignment dieser Strukturen wird vom Hersteller der **Platform Adapter Extension** vorgegeben und ist für den Hersteller des **Device Driver** maßgebend.

Derartige Strukturen sind bei Verwendung von **Platform Adapter Extensions** der **RDE** immer mit einem Alignment von 8 übersetzt.

3.1.3 Alignment an der Device Driver API

Das Alignment für Strukturen der **Device Driver** API und deren übertragene **DCD**-Strukturen wird vom Hersteller des **Device Driver** festgelegt und über die Präprozessor-Direktive `DD_API_ALIGNMENT` in `Driver/Source/Settings.h` eingestellt.

Der **Skeleton Generator** verwendet das entsprechend im seinem Dialog eingestellte Alignment.

DCD-Strukturen sind Strukturen, welche sich erst aus den Inhalten der **DCD**-Dateien ergeben und vom **Skeleton Generator** generiert werden. Sie werden an den API-Funktionen von **Device Drivers** durch das generische Typkonstrukt `void*` beschrieben.

3.2 Linux

Für UNIX-Betriebssysteme wird das Makefile „`\\Driver\\Project_Lin32\\Makefile`“ generiert. Eine Integration der Quellen in eine eigene Entwicklungs-Software ist üblicherweise vom Hersteller des **Device Driver** selbst vorzunehmen.

3.2.1 Alignment an der Platform Adapter API

Das Alignment für Strukturen, welche an der **Platform Adapter** API durch Einbindung der Header-Datei `gdi_pa.h` und der damit implizit eingebundenen Header-Dateien festgelegt werden, ist nicht einstellbar.

Das heißt, dass der Compiler über das Alignment entscheidet. Bei Verwendung einer IA32-Architektur ist üblicherweise ein Alignment von 4 vorzufinden.



Auf Unix-Systemen wird generell die Erstellung aller Komponenten direkt auf dem Zielsystem empfohlen, um Alignment-Konflikten vorzubeugen.

3.2.2 Alignment für Strukturen aus Communication Types

Communication Type Strukturen sind Strukturen aus **Platform Adapter Extensions**, welche weitere Arten von Kommunikations-Kanälen oder anderweitigen zusätzlichen Systemressourcen verfügbar machen. Sie werden an den API-Funktionen von **Platform Adapters** und **Platform Adapter Extensions** durch das generische Typkonstrukt `void*` beschrieben.

Das Alignment dieser Strukturen wird vom Hersteller der **Platform Adapter Extension** vorgegeben und ist für den Hersteller des **Device Driver** maßgebend.

Derartige Strukturen sind bei Verwendung von **Platform Adapter Extensions** der **RDE** immer mit einem Alignment vom Compiler gewählten Alignment übersetzt. Damit ist das Alignment bei Verwendung einer IA32-Architektur üblicherweise 4.



Auf Unix-Systemen wird generell die Erstellung aller Komponenten direkt auf dem Zielsystem empfohlen, um Alignment-Konflikten vorzubeugen.

3.2.3 Alignment an der Device Driver API

Das Alignment für Strukturen der **Device Driver** API und deren übertragene **DCD**-Strukturen wird vom Hersteller des **Device Driver** festgelegt und über die Präprozessor-Direktive `DD_API_ALIGNMENT` in `Driver/Source/Settings.h`) eingestellt.

Unterstützt wird dabei das Alignment 1 bzw. die Verwendung eines vom Compiler gewählten Alignment, welches bei Verwendung einer IA32-Architektur üblicherweise 4 ist.

DCD-Strukturen sind Strukturen, welche sich erst aus den Inhalten der **DCD**-Dateien ergeben und vom **Skeleton Generator** generiert werden. Sie werden an den API-Funktionen von **Device Drivers** durch das generische Typkonstrukt `void*` beschrieben.



Auf Unix-Systemen wird generell die Erstellung aller Komponenten direkt auf dem Zielsystem empfohlen, um Alignment-Konflikten vorzubeugen.

4 Device Driver Skeleton

Ein **Skeleton Generator** kopiert Dateien, welche von allen **Device Driver Skeletons** genutzt werden [4.1] und nicht durch den Hersteller des **Device Driver** geändert werden dürfen.

Aus verschiedenen **DCD**-Inhalten ergeben sich weiterhin entsprechend speziell generierte Dateien [4.2], welche für die Modifikation durch den Hersteller des **Device Driver** vorbereitet sind.

4.1 Gemeinsame Dateien

Gemeinsame Dateien sind im Verzeichnis **CommonFiles/** zu finden.

Diese Dateien werden nicht generiert, sondern vielmehr durch RDE beigestellt.



Diese Dateien dürfen durch den Hersteller des **Device Driver** nicht modifiziert werden.

4.1.1 Dateien des ASAM GDI Standards

ASAM GDI Standarddateien sind im Verzeichnis **CommonFiles/Libs/GDIPlatform45** abgelegt. Enthalten sind dabei alle **GDI**-Versionen bis 4.5.

Diese Dateien sind innerhalb des **ASAM GDI** Standards festgelegt. Sie sind öffentlich erreichbar und dienen als Schnittstellen-Definitionen zwischen **ASAM GDI** Komponenten.

Sie werden bereits weitestgehend durch den **Device Driver Skeleton** eingebunden, wobei dieser bereits Implementierungen entsprechend dem Standard enthält.

Eine Ausnahme bilden Definitionen zu **Communication Types**, welche in den Header-Dateien enthalten sind, welche mit „gdi_paextct_“ beginnen.

Tabelle 1 listet die wichtigsten Schnittstellen-Dateien im Überblick.

API-Komponente	Header	Inhalt	Verwendung
Device Driver API	gdi_dd.h	DD API	GDI_API4.cpp
Platform Adapter API	gdi_pa.h	PA API	PA_API4.cpp
Platform Adapter Extension API	gdi_paext.h	EXT API	
Serielle Verbindung	gdi_paextct_com	Serielle Verbindung	Direkt PA_API4.hpp +
TCP/IP, UDP/IP	gdi_paextct_ip4		Direkt PA_API4.hpp +
Parallele Verbindung	gdi_paextct_lpt		Direkt PA_API4.hpp +

Tabelle 1: Benötigte Header-Dateien



Diese Dateien dürfen durch den Hersteller des **Device Driver** nicht modifiziert werden.

4.1.2 Dateien des RDE Platform Adapter 4.5



Diese Dateien dürfen durch den Hersteller des **Device Driver** nicht modifiziert werden.

4.1.3 Dateien des RDE Trace Moduls



Diese Dateien dürfen durch den Hersteller des **Device Driver** nicht modifiziert werden.

4.1.4 Dateien für Device Drivers 4.4



Diese Dateien dürfen durch den Hersteller des **Device Driver** nicht modifiziert werden.

4.2 Generierte Dateien

Der Skeleton Generator generiert für jedes Device Driver Skeleton einen speziellen Satz an Dateien, deren Inhalte sich an der DCD ausrichten.