Document Title	Requirements on I2C Driver
Document Category	Requirement specification (RS)
Document UID	1100
Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R26-11

Document Change History			
Date	**Release**	**Changed by**	**Description**
2026-11-26	R26-11	AUTOSAR Release Management	• No content changes
2025-11-27	R25-11	AUTOSAR Release Management	• No changes
2024-11-27	R24-11	AUTOSAR Release Management	• Initial release

Disclaimer

This work (specification and/or software implementation) and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

1 Scope of Document 4

2 Conventions to be used 5
2.1 Document Conventions . . . 5
2.2 Requirements Guidelines . . . 5

3 Acronyms and abbreviations 6

4 Requirements Specification 7
4.1 Functional Overview . . . 7
4.1.1 I2C Driver, common functionality . . . 7
4.1.2 Asynchronous I2C functionality . . . 7
4.1.3 Synchronous I2C functionality . . . 7
4.2 Functional Requirements . . . 7
4.2.1 External standard . . . 7
4.2.2 Configuration . . . 8
4.2.3 Normal Operation . . . 8

5 References 12

A Change history of AUTOSAR traceable items 13
A.1 Traceable item history of this document according to AUTOSAR Release R26-11 . . . 13
A.1.1 Added Requirements in R26-11 . . . 13
A.1.2 Changed Requirements in R26-11 . . . 13
A.1.3 Deleted Requirements in R26-11 . . . 13
A.2 Traceable item history of this document according to AUTOSAR Release R25-11 . . . 13
A.2.1 Added Requirements in R25-11 . . . 13
A.2.2 Changed Requirements in R25-11 . . . 13
A.2.3 Deleted Requirements in R25-11 . . . 13
A.3 Traceable item history of this document according to AUTOSAR Release R24-11 . . . 14
A.3.1 Added Requirements in R24-11 . . . 14
A.3.2 Changed Requirements in R24-11 . . . 14
A.3.3 Deleted Requirements in R24-11 . . . 14

1 Scope of Document

This document specifies requirements on the I2C Driver module.

2 Conventions to be used

2.1 Document Conventions

The representation of requirements in AUTOSAR documents follows the table specified in [TPS_STDT_00078], see [1, Standardization Template].

The verbal forms for the expression of obligation specified in [TPS_STDT_00053] shall be used to indicate requirements, see [1, Standardization Template].

2.2 Requirements Guidelines

Each module specific chapter contains a short functional description of the Basic Software Module. Requirements of the same kind within each chapter are grouped under the following headlines (where applicable):

Functional Requirements:

- Configuration (which elements of the module need to be configurable)
- Initialization.
- Normal Operation.
- Shutdown Operation.
- Fault Operation.
- ...

Non-Functional Requirements:

- Timing Requirements.
- Resource Usage.
- Usability.
- Output for other WPs (e.g. Description Templates, Tooling, ...).
- ...

3 Acronyms and abbreviations

The glossary below includes acronyms and abbreviations relevant to I2C Driver that are not included in the AUTOSAR Glossary [2].

Abbreviation / Acronym:	Description:
Controller	A device controlling other devices (Targets). Further it initiates a transfer, generates clock signals and terminates a transfer. Formerly known as "Master".
Rx	Reception (in the context of bus communication)
SCL	Serial Clock
SDA	Serial Data
Target	A device being addressed by a Controller device. Formerly known as "Slave".
TX	Transmission (in the context of bus communication)
Sequence	A Sequence is a number of consecutive Jobs to transmit.
Job	A Job is a software exchange medium for data that are defined with the same criteria: Config. Parameters, Number of Data elements with the same size and data pointers.
Request Queue	A request queue queues only the transmission request without the data itself.
active request	An active request is an ongoing transmission.
pending request	A pending request is a queued transmission request.

Table 3.1: Acronyms and abbreviations used in the scope of this Document

4 Requirements Specification

4.1 Functional Overview

4.1.1 I2C Driver, common functionality

An I2C (Inter-Integrated Circuit) bus is a multi-node bus system, where the Controller initiates communication with a Target device. The I2C protocol uses a 2-wire interface for serial communication. Data transmission is facilitated through a Serial Data Line (SDA) and a Serial Clock Line (SCL).

The I2C module provides Job-based read write, and transfer access to various devices on the I2C bus. Jobs can be combined and transmitted in a sequential manner. These Sequences can potentially be sorted by priorities if implemented. They have a static configuration that specifies specific parameters.

In the I2C system, a device is typically identified by the I2C hardware unit being used and the associated device address. The module functions exclusively as an I2C Controller, controlling the data flow and addressing of devices on the bus.

4.1.2 Asynchronous I2C functionality

This part of the I2C Driver provides asynchronous read, write and transfer access to different devices on I2C busses and callback notifications. The access to the different I2C Sequences can be priority controlled.

4.1.3 Synchronous I2C functionality

This part of the I2C Driver provides synchronous read, write and transfer access to different devices on I2C busses.

4.2 Functional Requirements

4.2.1 External standard

[CP_RS_I2C_00016] I2C Driver Compliance with UM10204 I2C-Bus Specification
⌈

Description:	The I2C driver shall be compliant with UM10204 I2C-bus specification and user manual Rev. 7.0 - 1 October 2021.
Rationale:	–
Use Case:	–

▽

△

Dependencies:	–
Supporting Material:	https://www.nxp.com/docs/en/user-guide/UM10204.pdf

⌋

4.2.2 Configuration

[CP_RS_I2C_00002] I2C Driver Transmission Setup ⌈

Description:	The I2C driver shall provide a service to setup transmission relevant parameters such as buffers and the number of data elements.
Rationale:	To prepare a fault free transmission.
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋

4.2.3 Normal Operation

[CP_RS_I2C_00003] I2C Driver Data Transfer Speeds ⌈

Description:	The I2C driver shall support the configuration of the data transfer modes: • Standard-mode (Sm) • Fast-mode (Fm) • Fast-mode Plus (Fm+) • High-speed mode (Hs-mode) • Ultra Fast-mode (UFm) Additional Information: The configuration shall limit the data transfer modes supported by the Hardware.
Rationale:	–
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋

[CP_RS_I2C_00004] I2C Driver Multi Controller Mode ⌈

Description:	The I2C Driver shall support the configuration of Multi Controller Mode.
Rationale:	To ensure that, if more than one controller simultaneously attempts to control the bus, only one of them is allowed to do so, so that no message will be corrupted.
Use Case:	As a clarifying example: Microcontroller A and microcontroller B want to put information on the bus at the same time. A produces a '1' and B produces a '0'. Now A stops sending due to arbitration loss and B can keep going.
Dependencies:	–
Supporting Material:	–

⌋

[CP_RS_I2C_00005] I2C Driver 7-bit and 10-bit Bus Addressing Modes ⌈

Description:	The I2C driver shall support both 7-bit and 10-bit bus addressing modes. Additional Information: Usually, a fully functional header consists of 8 bits. The target address is 7 bits long, followed by one bit for the data direction. Due to 7-bit addressing the number of `Target` devices is limited to 128. By using 10-bit addressing, the number of targets can be increased.
Rationale:	10-bit addressing expands the number of possible addresses.
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋

[CP_RS_I2C_00017] Independent Treatment of Each Hardware Unit ⌈

Description:	The I2C driver shall support independent transmission of `sequence`s per hardware unit.
Rationale:	–
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋

[CP_RS_I2C_00006] I2C Driver Queuing Mechanism for Sequences ⌈

Description:	The I2C driver shall support a queuing mechanism for different `sequence`s Additional Information: The same `sequence` can be queued only once.

▽

△

Rationale:	To guarantee continuous transmissions on the I2C-bus.
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋

[CP_RS_I2C_00007] I2C Driver Support for Asynchronous, Interrupt Driven Transmit/ Read Operations ⌈

Description:	The I2C driver shall support asynchronous, interrupt driven transmit/ read operations. This functionality shall allow reading/ writing data to/from the selected I2C device, giving the following parameters to the driver: • Job • Address of the Job • Pointer to TX transmission data location (equals Null Pointer in case of a read operation) • Pointer to RX transmission data location (equals Null Pointer in case of a write operation) • Length of the transmission This action shall be done asynchronously.
Rationale:	–
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋

[CP_RS_I2C_00008] I2C Driver Support for Synchronous, Non-interrupt Driven Transmit/ Read Operations ⌈

Description:	The I2C driver shall support synchronous, non-interrupt driven transmit/ read operations. This functionality shall allow reading/ writing data to/from the selected I2C device, giving the following parameters to the driver: • Job • Address of the Job • Pointer to TX transmission data location (equals Null Pointer in case of a read operation) • Pointer to RX transmission data location (equals Null Pointer in case of a write operation) • Length of the transmission This action shall be done synchronously with the call of function.

▽

△

Rationale:	–
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋

[CP_RS_I2C_00012] I2C Driver Error Handling ⌈

Description:	The I2C driver shall provide specific error handling.
Rationale:	To help the driver identify and respond appropriately to different errors.
Use Case:	An I2C reset if SCL is stuck or sending 9 clock pulses if SDA is stuck.
Dependencies:	–
Supporting Material:	–

⌋

[CP_RS_I2C_00015] I2C Target device support ⌈

Description:	The I2C driver shall support Target devices
Rationale:	–
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋

5 References

[1] Standardization Template
AUTOSAR_FO_TPS_StandardizationTemplate

[2] Glossary
AUTOSAR_FO_TR_Glossary

A Change history of AUTOSAR traceable items

Please note that the lists in this chapter also include traceable items that have been removed from the specification in a later version. These items do not appear as hyperlinks in the document.

A.1 Traceable item history of this document according to AUTOSAR Release R26-11

A.1.1 Added Requirements in R26-11

none

A.1.2 Changed Requirements in R26-11

none

A.1.3 Deleted Requirements in R26-11

none

A.2 Traceable item history of this document according to AUTOSAR Release R25-11

A.2.1 Added Requirements in R25-11

none

A.2.2 Changed Requirements in R25-11

none

A.2.3 Deleted Requirements in R25-11

none

A.3 Traceable item history of this document according to AUTOSAR Release R24-11

A.3.1 Added Requirements in R24-11

[CP_RS_I2C_00002] [CP_RS_I2C_00003] [CP_RS_I2C_00004] [CP_RS_I2C_00005] [CP_RS_I2C_00006] [CP_RS_I2C_00007] [CP_RS_I2C_00008] [CP_RS_I2C_00012] [CP_RS_I2C_00015] [CP_RS_I2C_00016] [CP_RS_I2C_00017]

A.3.2 Changed Requirements in R24-11

none

A.3.3 Deleted Requirements in R24-11

none