**Spi Software Design**

***SteerTurnIllum***



|  |  |  |
| --- | --- | --- |
| Author: |  | Nicolae-Bogdan Bacrău |
| Version: |  | 1.0 |

Index

1 Glossary 3

2 Introduction 4

2.1 Purpose of the Document 4

2.2 Overview 4

3 Design Requirements 5

4 Information about this Document 6

4.1 Copyright 6

4.2 Version Index 6

# Glossary

This section contains a glossary of all the important terms and acronyms used inside the document.

|  |  |
| --- | --- |
| **Term / Acronym** | **Description** |
| AUTOSAR | AUTomotive Open System ARchitecture |
| VFB | Virtual Functional Bus |
| SWC | Software Component |
| RTE | Runtime Environment |
| BSW | Basic Software |
| OS | Operating System |
| S/R | Sender / Receiver |
| C/S | Client / Server |
| ECU | Electronic Control Unit |
| uC | Microcontroller |
| ADC | Analog Digital Converter |
| DIO | Digital Input / Output |
| PWM | Pulse Width Modulation |
| SPI | Serial Peripheral Interface |

Table 1 - Glossary.

# Introduction

## Purpose of the Document

The purpose of the document is to define the software design of the ***Spi*** driver for the ***SteerTurnIllum*** embedded academy project.

## Overview

While the AUTOSAR ***Spi*** module is complex and is usually configured via generators, in this project it is manually implemented and its features are limited to:

* Only supports IB, one channel, one job, one sequence

# Design Requirements

1. The Spi driver shall be implemented according to AUTOSAR 4.3.1 specifications.
2. The Spi driver shall implement the following interfaces and all their required data types for a one Channel, one Job, one Sequence operation, using an internal buffer and Synchronuous Transmit mode (see, “9.3 Write/AsyncTransmit/Read (IB)” and “9.6.1 Write/SyncTransmit/Read (IB): Many Channels, many Jobs and one Sequence” from the AUTOSAR SPI specifications). This corresponds to the LEVEL 0 – simple Synchronous behavior.

* *Spi\_Init();*
* *Spi\_WriteIB();*
* *Spi\_ReadIB();*
* *Spi\_SyncTransmit();*

1. The Spi driver shall include *Init.h* and directly use the *Init\_gv\_Full\*Bits()* and / or *Init\_gv\_Masked\*Bits()* functions for loading registers.
2. The Spi driver shall consist of the following files:

* *Spi.c*: Implements all the required interfaces in a generic way.
* *Spi*.*h*: Exports all the required interfaces, data types and the post-build configuration.
* *Spi\_PBcfg.*c: Defines the list of registers and values to be written for setting the SPI HW
* *Spi\_Cfg.h:* Defines and exports the number of configured SPI channels and sequences, through the *SPI\_NUMBER\_OF\_CHANNELS and SPI\_NUMBER\_OF\_SEQUENCES* macros, and the zero based IDs of all the channels and sequences through the *SPI\_CHANNEL\_<ID> and SPI\_SEQUENCE\_<ID>* macros.

# Information about this Document

## Copyright

All rights, including translation rights, are reserved. Under no circumstances shall any fragment of this document be reproduced without written authorization from NTT DATA Romania S.A, including copying, photographing or replicating through other methods.

Copyright (c) NTT DATA Romania S.A.

## Version Index

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Chapter** | **Modification description** |
| 1.0 | 10.07.2021 | Nicolae-Bogdan Bacrău | All | Created. |

Table 2 - Version Index.