**Pwm 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 |

Table 1 - Glossary.

# Introduction

## Purpose of the Document

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

## Overview

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

* Configuring PWM channels.
* Writing PWM duty cycles.

# Design Requirements

1. The Pwm driver shall be implemented according to AUTOSAR 4.3.1 specifications.
2. The Pwm driver shall implement the following interfaces and all their required data types for configuring PWM channels (timer configurations) and writing PWM duty cycles:

* *Pwm\_Init();*
* *Pwm\_SetDutyCycle();*

1. The Pwm 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 Pwm driver shall consist of the following files:

* *Pwm.c*: Implements all the required interfaces in a generic way.
* *Pwm*.*h*: Exports all the required interfaces, data types and the post-build configuration.
* *Pwm\_PBcfg.*c: Defines the list of registers and values to be written for setting the timers in PWM mode, setting the periods of the PWM channels and the PWM initial duty cycles.
* *Pwm\_Cfg.h:* Defines and exports the following:
  + *PWM\_API\_SET\_DUTY\_CYCLE* macro: defining if the PWM API for setting duty cycles is activated or not (STD\_ON or STD\_OFF).
  + *PWM\_NUMBER\_OF\_CHANNELS* macro: number of configured PWM channels.
  + *PWM\_<Port\_ID>\_<Port\_Pin\_ID>* macros: zero based IDs of all the PWM channels.

# 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.