**Turn 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 7

4.1 Copyright 7

4.2 Version Index 7

# 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 ***Turn*** SWC for the ***SteerTurnIllum*** embedded academy project.

## Overview

The ***Turn*** SWC implements the control of the turn indication LEDs based on the presses of the left turning, right turning and hazard buttons.

# Design Requirements

1. The Turn SWC shall adhere to the structure illustrated in the composite structure diagram from **Figure 1**.

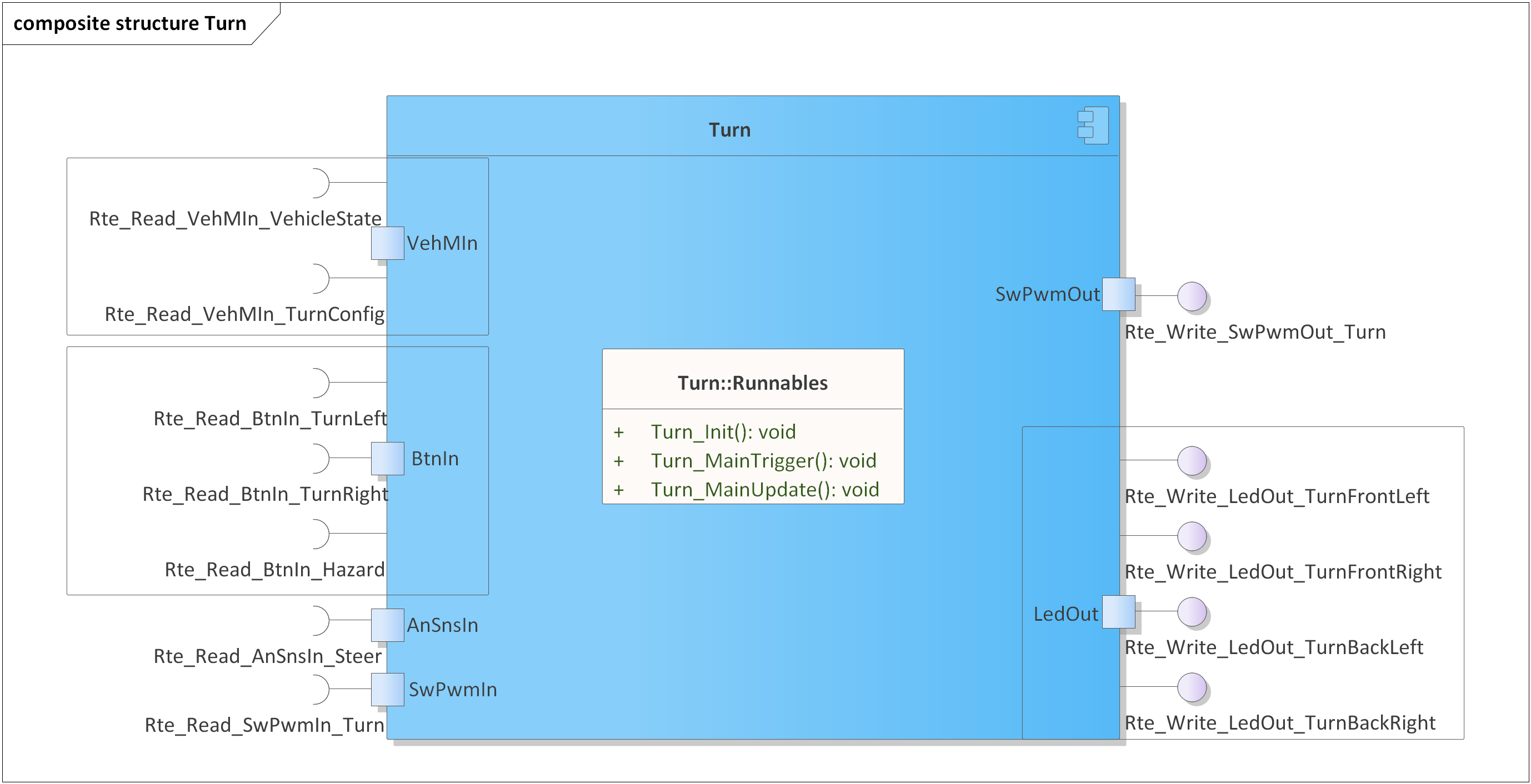


Figure 1 - Turn composite structure diagram.

1. The Turn SWC shall implement the turning requirements as described in chapter 4.4w from the System Requirements.
2. The Turn SWC shall be implemented in two C files: *Turn.h* for exporting all the runnables, and *Turn.c* for implementing the runnables.
3. The Turn SWC shall implement the *void Turn\_Init(void)* runnable for initializing all the internal static and global variables.
4. The Turn SWC shall implement the *void Turn\_MainTrigger(void)* runnable for:

* Reading the press state of the left turning, right turning and hazard buttons through *Rte\_Read\_BtnIn\_\*()*.
* Reading the current turn configuration through *Rte\_Read\_VehMIn\_TurnConfig()*.
* Reading the steering joystick position through *Rte\_Read\_AnSnsIn\_Steer()*.
* Identifying the short and long button triggers as well as the joystick steering triggers for performing transitions between the turning states.
* Starting a new SwPwm job, through *Rte\_Write\_SwPwmOut\_Turn(),* when performing state transitions.

1. The Turn SWC shall implement the *void Turn\_MainUpdate(void)* runnable for writing the indication LEDs control data, through *Rte\_Write\_LedOut\_\*()*, based on the current turning state and on the SwPwm job status, read through *Rte\_Read\_SwPwmIn\_Turn()*.

# 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.