**VehM 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 ***VehM*** SWC for the ***SteerTurnIllum*** embedded academy project.

## Overview

The ***VehM*** SWC implements the control of the vehicle states (NORMAL or CONFIGURATION) and the control of the configuration menu while in the CONFIGURATION state.

# Design Requirements

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

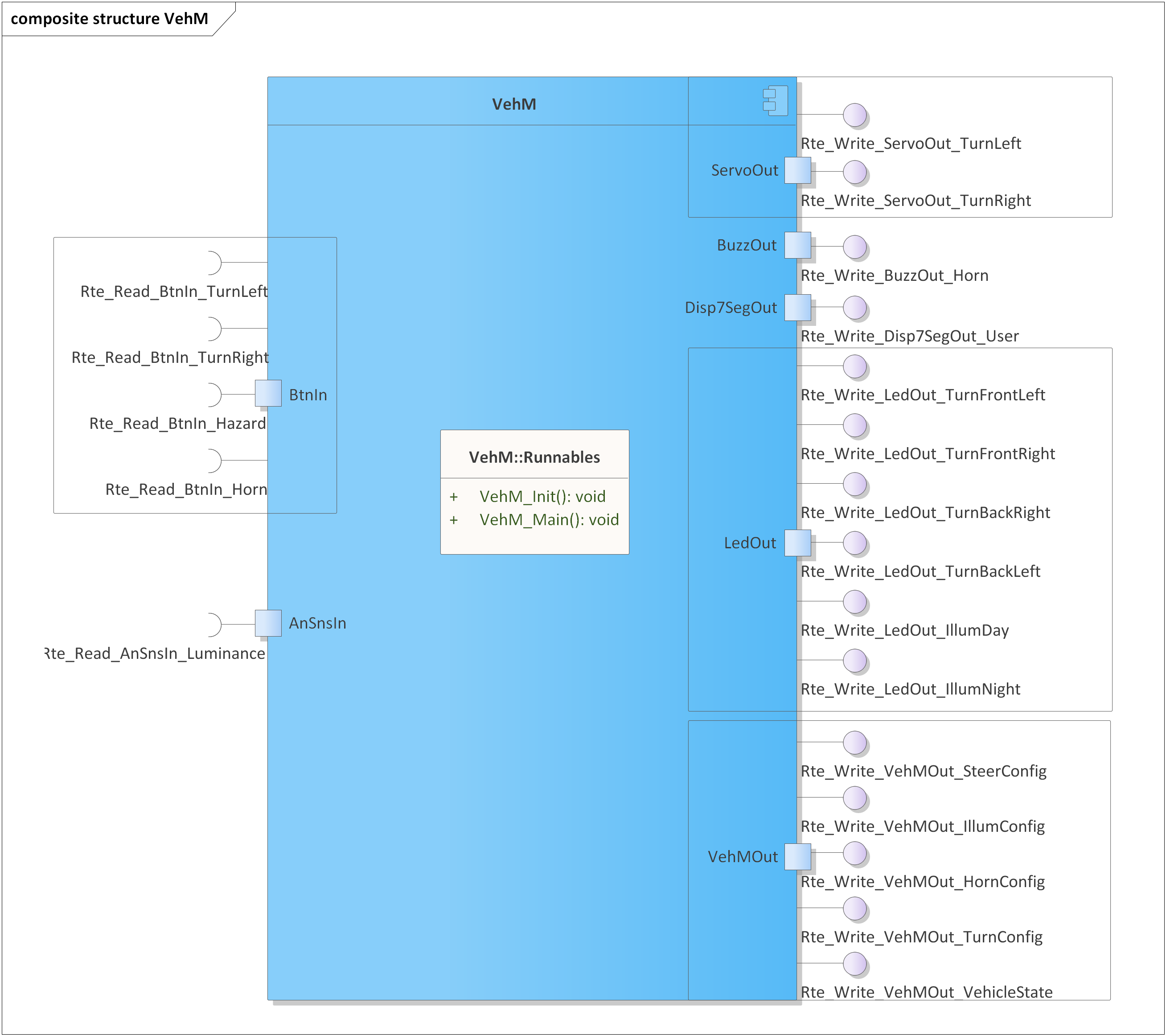


Figure 1 - VehM composite structure diagram.

1. The VehM SWC shall implement the vehicle management requirements as described in chapter 4.2 from the System Requirements.
2. The VehM SWC shall be implemented in two C files: *VehM.h* for exporting all the runnables, and *VehM.c* for implementing the runnables.
3. The VehM SWC shall implement the *void VehM\_Init(void)* runnable for initializing all the internal static and global variables.
4. The VehM SWC shall implement the *void VehM\_Main(void)* runnable for:

* Reading the press state of all the buttons, through *Rte\_Read\_BtnIn\_\*()*, and the luminance potentiometer, through *Rte\_Read\_AnSnsIn\_Luminance()*.
* Checking the trigger for performing the transition between the RTE\_VEHICLE\_STATE\_NORMAL and RTE\_VEHICLE\_STATE\_CONFIG states and writing the current vehicle state through *Rte\_Write\_VehMOut\_VehicleState()*.
* If in the RTE\_VEHICLE\_STATE\_CONFIG state:
  + Setting default control values for the servomotors and buzzer through *Rte\_Write\_ServoOut\_\*()* and *Rte\_Write\_BuzzOut\_Horn()*.
  + Identifying the triggers of left turning, right turning and hazard buttons and perform the transitions between the CONFIGURATION states.
  + Display the current state on the indication LEDs through *Rte\_Write\_LedOut\_Turn\*().*
  + Display the current substate on the illumination LEDs through *Rte\_Write\_LedOut\_Illum\*()*.
  + Display the current configuration value for the current configuration state and substate on the 7 segments display through *Rte\_Write\_Disp7SegOut\_User()*.
  + Save the current configuration value for the current configuration state and substate through *Rte\_Write\_VehMOut\_\*Config()* when the horn button is pressed.

# 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.