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

## Overview

The ***Btn*** SWC implements the reading, debouncing and active values processing of all the button inputs, proving a simple button pressed or not pressed status.

# Design Requirements

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

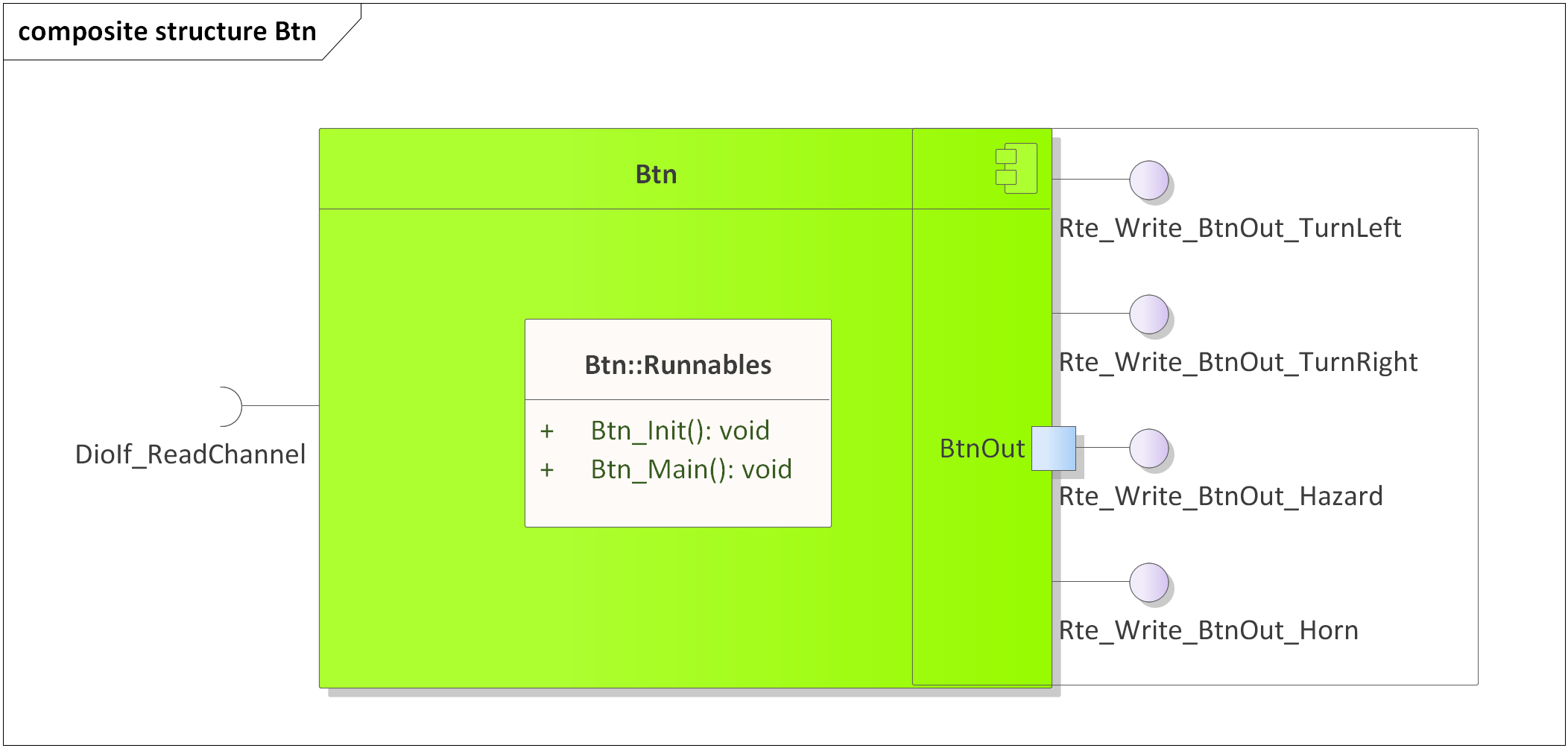


Figure 1 - Btn composite structure diagram.

1. The Btn SWC shall assure a debouncing mechanism according to Chapter 3.2 and [SYRE\_GEN\_11] from the System Requirements.
2. The Btn SWC shall adhere to the SWC file structure template from 4\_Engineering\1\_Software\2\_Development\1\_Sources\8\_Templates\Swc\Code.
3. The Btn SWC shall contain the following configuration parameters:

* In *Btn\_Cfg.h*:
  + *BTN\_MAIN\_PERIODICITY\_US*: macro defining the periodicity of the main function in microseconds.
  + *BTN\_DEBOUNCE\_WINDOW\_US*: macro defining the debounce window for all the buttons, in microseconds.
  + *BTN\_NUMBER\_OF\_INSTANCES*: macro defining the number of button instances to be processed in the main function.
  + *BTN\_\*\_INSTANCE*: zero based macros defining unique IDs of all the button instances, needed for the interaction between the core and RTE.
* In *Btn\_Cfg.c*:
  + *Btn\_gkat\_Config*[BTN\_NUMBER\_OF\_INSTANCES]: internal global constant array defining the active value (STD\_HIGH or STD\_LOW) and the DioIf channel for all the button channels.

1. The Btn SWC shall implement the *void Btn\_Init(void)* runnable for initializing all the internal static and global variables.
2. The Btn SWC shall implement the *void Btn\_Main(void)* runnable for implementing the processing of all the button instances as follows:
   * Reads the button DIO channels through *DioIf\_ReadChannel()*.
   * Calculates the current raw press state depending on the configured active press state in comparison with the actual read DIO level.
   * Applies the debounce algorithm on the current raw press state.
   * Writes the debounced buttons status to the RTE through *Rte\_Write\_BtnOut\_\*()*. RTE\_BTN\_INIT status shall be reported until a full (successful) debounce has been performed since the initialization.
3. The Btn SWC shall include *DioIf.h* and directly use the *DioIf\_ReadChannel()* functionfor reading the button DioIf 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.