***Project Title***

**Software Architecture Design Specification - Application**

**Ver. XX**

**Revision History**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | Revision  Date | Contents | Responsibility | Reviewed By | Approval |
| Date | Date | Date |
|  |  |  |  |  |  |
|  |  |  |
|  |  |  |  |  |  |
|  |  |  |
|  |  |  |  |  |  |
|  |  |  |
|  |  |  |  |  |  |
|  |  |  |
|  |  |  |  |  |  |
|  |  |  |
|  |  |  |  |  |  |
|  |  |  |
|  |  |  |  |  |  |
|  |  |  |

**Table of Contents**

[1. Introduction 4](#_Toc14177375)

[1.1. Objective of This Document 4](#_Toc14177376)

[1.2. Terms, definitions and abbreviated terms 4](#_Toc14177377)

[1.3. Reference 4](#_Toc14177378)

[2. Static Architecture 5](#_Toc14177379)

[2.1. Static View of The Architecture 5](#_Toc14177380)

[2.2. SW Component Specification 6](#_Toc14177381)

[2.2.1. ADC\_Health\_Check.c 6](#_Toc14177382)

[3. SW Dynamic Behavior 9](#_Toc14177383)

[3.1. SW Function List 9](#_Toc14177384)

[3.2. SW Function Behavior 9](#_Toc14177385)

[3.2.1. SW Function Specification: Function Name 9](#_Toc14177386)

[3.2.2. SW Function Specification: Function Name 11](#_Toc14177387)

[3.3. SW Task Structure 12](#_Toc14177388)

[4. Global Signal(variable) Definition 13](#_Toc14177389)

[5. Calibration / Configuration Specification 14](#_Toc14177390)

# Introduction

## Objective of This Document

This document specifies software architecture design of the application function for the [product name] to achieve followings

* Explain the SW functions and related design properties at the technical aspect of the system
* Provide comprehensible understanding of the software architecture design
* Specify the structure and the design properties of the implemented software
* Provide verification criteria for each software functions and design properties

## Terms, definitions and abbreviated terms

|  |  |  |
| --- | --- | --- |
| Term | Description | Note |
| ALS | Auto Light Sensor |  |
| SAS | Sun Angle Sensor |  |
| System | Set of elements that relates at least a sensor, a controller and an actuator with one another |  |
|  |  |  |

## Reference

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Document Name | Version | Release Date | Author | Note |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |
| --- |
| Specification Guideline |
| Any documents that contribute to specify this document should be identified. Generally following items can be considered.   * Software requirements specification * HW-SW interface specification * Analysis report to define the SW functionality (e.g. algorithms) at software architecture development level * Engineering standards (ex. VDA, SAE standards) * Supplier’s own SW development standards |

# Static Architecture

## Static View of The Architecture

*Context type: Informative*

|  |
| --- |
|  |

|  |
| --- |
| Specification Guideline |
| Static Architecture should have below properties   * High level components or sub systems * Relationship among components or sub systems * Structural, hierarchical * Component or sub system ID * Other principles of its design |

## SW Component Classification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Component ID | Component Name | Layer | Type | ASIL | Component Brief Description |
| SWE01  (REU-03) | ADC\_Health\_Check.c | Device driver | Re-use | ASIL B | - |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Note) Use the SW component “management ID”, if “re-use” is applied.

|  |
| --- |
| Specification Guideline |
| 1. Component ID: unique ID of the software component (the ID shall be unique in the development project). The management ID shall be specified when the SW component in the “SW component list” is used as “re-use”. 2. Component name: C file name 3. Type:    * New: New development in the project    * Re-use: Using the legacy SW component that was registered in the “Re-usable SW Component List” (SW component qualification activity shall be initiated.    * COTs with QM: A software component that is developed by 3rd-party without ISO 26262 application    * COTs with ASIL (SEooC): A software component that is developed by 3rd-party with ISO 26262 application (i.e. safety requirements and ASIL) 4. Layer: Application, basic software, device driver, etc |

## SW Component Specification

### ADC\_Health\_Check.c

*Context type: Specification*

*Assumption: SSR-13 / ASIL D / Health status of the ADC shall be verified during the boot-up sequence (The SW component is dedicated only for the safety mechanism)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Component ID / C file Name | SWE01 / ADC\_Health\_Check.c | | | |
| SWR/SSR ID | SWR01-1, SSR-13 | | | |
| SW Function | ECU Initialization | | ASIL | ASIL D |
| Assigned SW Task | TASK4 | | Description | Detection of the failure (offset, stuck at) of the ADC peripheral in the MCU |
| Inputs | Name | Input Type | Data Type | Description |
| GV\_IGN\_STATUS | Global Variable | U8 | Vehicle ignition status (Refer to global variable specification for details\_ |
| Outputs | Name | Output Type | Data Type | Specification |
| GV\_ADC\_STATUS | Global Variable | U8 | - |
| GV\_SR\_EX\_FLAG\_T4 | Global Variable | U16 | - |
| REG\_ADC01 | Register | U16 | Refer to HSI specification for details |
| REG\_RST01 | Register | U16 | Refer to HSI specification for details |
| Functionality | **ㆍBehavior sequence**   1. Configure the ADC HW register to check the band-gap voltage 2. Measure the voltage successive 3 times with 1ms interval 3. Check the fault status of the ADC 4. Perform COLD RESET when fault is detected   **ㆍPseudo code**  void **eFunc\_ADC\_BG\_TEST**(void) // Entry function  {  Clear (GV\_SR\_EX\_FLAG\_T4)  if (GV\_IGN\_STATUS == running)  {  ***iFunc\_ADC\_REG\_Configuration()***  ***iFunc\_ADC\_Read()*** / 3times  Diagnosis & Update (GV\_ADC\_STATUS)  If(GV\_ADC\_STATUS == offset or stuck)  {  Perform HW reset  }  }  Else  {  Update (GV\_ADC\_STATUS)  }  Set (GV\_SR\_EX\_FLAG\_T4)  Return  } | | | |
| HW Resource Usage | MCU ADC Channel 01  HW reset | | | |
| Target Execution Time Duration | Under 200us | | | |
| C Function List | | | | |
| Function Prototype | | Usage | Description | |
| void Entry\_Func\_ADC\_BG\_TEST(void) | | Entry | - | |
| U8 iFunc\_ADC\_REG\_Configuration(U8 i\_channel, U8 i\_mode) | | Internal | - | |
| U8 cFunc\_ADC\_Read(U8 i\_channel) | | External | - | |

ㆍSpecification Guideline

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Component ID / C file Name | A C file should be regarded as a software component except for specific case (e.g. common library C file)  A software component should have certain level of cohesion and should be distinguishable with other software components in the view point of code decoupling. | | | |
| SWR/SSR ID | Identification of software requirements that are allocated to this SW component | | | |
| SW Function | SW function name that is defined in the SW requirements specification | | ASIL | ASIL of the SW component |
| Assigned SW Task | Identification of SW Task that is to contain this software component | | Description | Brief description of the functionality of this SW component |
| Inputs | Name | Input Type | Data Type | Description |
| Variable name. Followings shall be considered.   * Input arguments of the entry function * Global variables that are read by this SW component * HW register that is read by this SW component | Global variable, input argument of the entry function, HW register ID | Data type (ex. U8, unsigned int) | Description for the input |
| Outputs | Name | Output Type | Data Type | Specification |
| Variable name. Followings shall be considered   * Return * Global variables that are read by this SW component * HW register that is to write in this SW component | Global variable, return, HW register ID | Data type (ex. U8, unsigned int) | Description for the output |
| Functionality | Specification of this software component. Followings can be considered to specify the functionality of this SW component   * Execution sequence of the functionality * Internal function call sequence * Data flow in this SW component level   Notation of truth table, flow chart or pseudo code is preferred depends on the characteristics of the SW component. | | | |
| HW Resource Usage | Following can be identified in this specification depends on needs   * Limit of code size, data size * Usage of peripherals of the MCU (incl. cold reset, SW reset) * Usage of communication channel (e.g. CAN, LIN, SPI) | | | |
| Target Execution Time Duration | Max. allowed execution time interval (this value depends on the SW requirements or system level functionality, not depends on the performance of the MCU) | | | |
| C Function List | | | | |
| Function Prototype | | Usage | Description | |
| A C function can be classified by followings   * Entry: Main function of the SW component * Internal: sub-function that is called by the entry * External: this sub-function is designed to be called by the entry function of other internal function but for external function call (e.g. library function) | | Entry / Internal / External | Minimum contents   * Function call argument specification * Return value specification | |

### ADC\_Health\_Check.c

*Context type: Specification*

*Assumption: SSR-13 / ASIL D / Cold reset shall be initiated when ADC fault is occurred. (The SW component consist of the intended functionality and the safety mechanism)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Component ID / C file Name | SWE01 / ADC\_Health\_Check.c | | | |
| SWR/SSR ID | SWR01-1, SSR-13 | | | |
| SW Function | ECU Initialization | | ASIL | ASIL D |
| Assigned SW Task | TASK4 | | Description | Detection of the failure (offset, stuck at) of the ADC peripheral in the MCU |
| Inputs | Name | Input Type | Data Type | Description |
| GV\_IGN\_STATUS | Global Variable | U8 | Vehicle ignition status (Refer to global variable specification for details\_ |
| Outputs | Name | Output Type | Data Type | Specification |
| GV\_ADC\_STATUS | Global Variable | U8 | - |
| GV\_SR\_EX\_FLAG\_T4 | Global Variable | U16 | - |
| REG\_ADC01 | Register | U16 | Refer to HSI specification for details |
| REG\_RST01 | Register | U16 | Refer to HSI specification for details |
| Functionality | **ㆍBehavior sequence**   1. Configure the ADC HW register to check the band-gap voltage 2. Measure the voltage successive 3 times with 1ms interval 3. Check the fault status of the ADC 4. Perform COLD RESET when fault is detected (ASIL D / SSR-13)   **ㆍPseudo code**  void **eFunc\_ADC\_BG\_TEST**(void) // Entry function  {  Clear (GV\_SR\_EX\_FLAG\_T4)  if (GV\_IGN\_STATUS == running)  {  ***iFunc\_ADC\_REG\_Configuration()***  ***iFunc\_ADC\_Read()*** / 3times  Diagnosis & Update (GV\_ADC\_STATUS)  If(GV\_ADC\_STATUS == offset or stuck)  {  Perform HW reset  }  }  Else  {  Update (GV\_ADC\_STATUS)  }  Set (GV\_SR\_EX\_FLAG\_T4)  Return  } | | | |
| HW Resource Usage | MCU ADC Channel 01  HW reset | | | |
| Target Execution Time Duration | Under 200us | | | |
| C Function List | | | | |
| Function Prototype | | Usage | Description | |
| void Entry\_Func\_ADC\_BG\_TEST(void) | | Entry | - | |
| U8 iFunc\_ADC\_REG\_Configuration(U8 i\_channel, U8 i\_mode) | | Internal | - | |
| U8 cFunc\_ADC\_Read(U8 i\_channel) | | External | - | |

# SW Dynamic Behavior

## SW Function List

*Context type: Informative*

|  |  |  |  |
| --- | --- | --- | --- |
| System Level Function | SW Function Name  (Top Level SW Function) | Description | ASIL |
| Cell voltage measurement and CAN communication | Cell voltage measurement |  |  |
| CAN management |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| Specification Guideline |
| To design the top-level software functions, required software components, sub-function in the software components, software function deployment into the lowest function level (ex. C-function) can be performed. In this chapter, the identified top-level software function is to be specified to provide the understanding of the software function to meet the system level functionality. |

## SW Function Behavior

### SW Function Specification: Function Name

*Context type: Specification*

#### Input Definition

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Input Type | Input Description | ASIL |
| Cell voltages | data | Measurement data of the individual cell voltages |  |
| Measurement enable | control | Control signal to enable the cell voltage measurement functionality |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| Specification Guideline |
| The inputs of the top-level software function can be identified in the information level   * The signal name of the “static view of the architecture” is preferred. * It is not recommended to use the global variables, function calls to represent the inputs of the top-level software function. (In this design stage, all global variables, function calls is not identified) Appropriate level of abstraction is required. The purpose of this abstraction is to provide better understanding of the top-level software function. * The inputs can be classified into two groups (data and control). “data” means that it is to be processed into other forms or used to execute a specific function. “control” means that it is not contain a specific data to be processed but, used to control the execution sequence or execution itself (i.e. execution or not) |

#### Function Details

|  |
| --- |
|  |

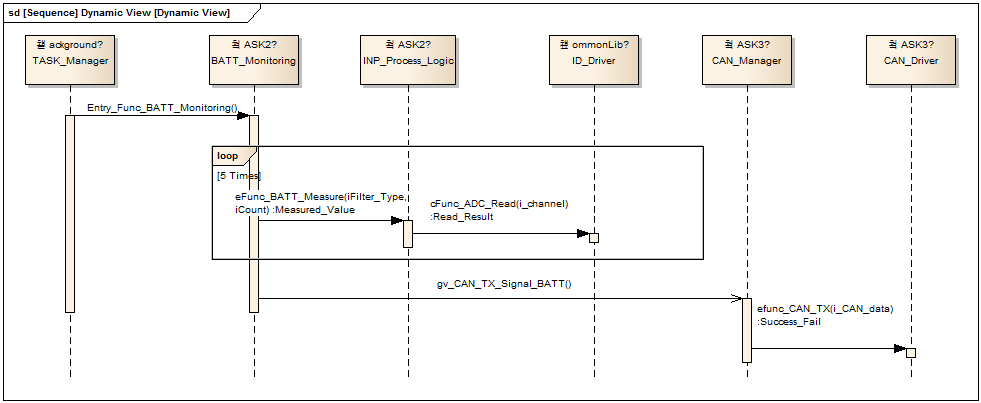
|  |
| --- |
| Specification Guideline |
| The functionality of the top-level software function can be described by using the appropriate description notation depends on the its functionality.   * State diagram, flow chart, pseudo code can be used. * The purpose of this “function details” is to provide the abstracted operation of the software function, not to specify the real interactions between the already defined software components. (Real execution of the software function in the static architecture can be specified by using the sequence diagram.) |

#### Output Definition

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Output Type | Output Description | ASIL |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| Specification Guideline |
| Refer to the “specification guideline” of the “input definition” |

#### Sequence Diagram

****

|  |
| --- |
| Specification Guideline |
| The purpose of the sequence diagram is to specify the real (non-abstracted) interaction between the software components of the top-level software function to specify the software design and to show the ‘as-is’ implementation of the software in the temporal manner. |

*■ Design Reference*

|  |  |  |
| --- | --- | --- |
|  | Reference ID | Description |
| SYSR, TSR |  |  |
| External Sources |  |  |

### SW Function Specification: Function Name

*Context type: Specification*

#### Input Definition

#### Function Details

#### Output Definition

#### Sequence Diagram

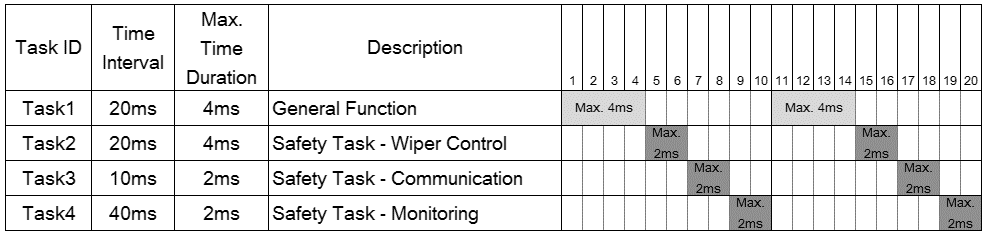
*■ Design Reference*

|  |  |  |
| --- | --- | --- |
|  | Reference ID | Description |
| SYSR, TSR |  |  |
| External Sources |  |  |

## SW Task Structure

*Context type: Specification*

* Description task scheduling strategies. ( NOT implement issues)

****

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Task ID | ASIL | Components List / Task Internal Sequence |
| 1 | TASK1 |  | General Function (세부 내용 생략) |
| 2 | TASK2 |  | 1. SWC07 / INP\_Process\_Logic  2. SWC19 / Wiper\_Motor\_Parking\_Monitoring  3. SWC02 / BATT\_Monitoring  4. SWC10 / MF\_Switch\_Monitoring  5. SWC15 / Safety\_Wiper\_Auto\_Logic  6. SWC13 / Relay\_Operating  7. SWC14 / Relay\_Test\_Logic |
| 3 | TASK3 |  | 1. SWC04 / CAN\_Manager  2. SWC09 / LIN\_Manager  3. SWC17 / Vehicle\_Speed\_Process\_Logic  4. SWC12 / Rain\_Sensing\_Process\_Logic |
| 4 | TASK4 |  | 1. SWC06 / ECC\_Overflow\_Monitoring  2. SWC01 / ADC\_Health\_Test  3. SWC18 / Watchdog\_Alive\_Test\_Logic  4. SWC11 / Program\_Flow\_Monitoring |
| - | Background |  | Scheduler, IO\_Driver, SPI\_Driver, CAN\_Driver, LIN\_Dirver |

# Global Signal(variable) Definition

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Signal name | Description | Created by | Used by | ASIL |
| 1 | GV\_IGN\_STATUS | IGN 상태 정보 값  - 0x1A: Engine off  - 0x2B: During cranking  - 0x3C: Engine is running | SWE01 |  |  |
| 2 | GV\_ADC\_STATUS | ADC failure 판정 값  - 0x0F: ADC ok  - 0xF1: ADC offset 또는 stuck  - 0xF4: Not available |  | SWE01 |  |
| 3 | GV\_SR\_EX\_FLAG\_T4 | Bit Configuration  [0] SWE11 / Program\_Flow\_Monitoring  [1] SWE06 / ECC\_Overflow\_Monitoring  [2] SWE01 / ADC\_Health\_Test  [3] SWE18 / Watchdog\_Alive\_Test\_Logic  [4 – 15] Reserved | SWE11 | SWE01  SWE11  SWE06  SWE18 |  |
| 4 | GV\_SR\_EX\_FLAG\_T2 | Bit Configuration  [0] SWE12 / Rain\_Sensing\_Process\_Logic  [1] SWE17 / Vehicle\_Speed\_Process\_Logic  [2] SWE04 / CAN\_Manager  [3] SWE09 / LIN\_Manager  [4 – 15] Reserved | SWE11 | SWE12  SWE17  SWE04  SWE09 |  |

|  |
| --- |
| Specification Guideline |
| Global signal list to be identified in case of when there is a risk of confusion as below   * Memory block, function call sequences etc. |

# Calibration / Configuration Specification

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Calibration | Description | Min | max | Res-  olution | Created by  (SW Component Name) | ASIL |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Configuration | Description | Value | Used by  (SW Component Name) | ASIL |
| CFG\_CAN\_GEN | CAN 사양 분리  1세대 CAN 사양일 떄 1,  2세대 CAN 사양일 때 2 | 1 | C11\_ExeSing1 |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |
| --- |
| Specification Guideline |
| Calibration specification   * See specification guideline of 2.2.1.1 C1\_ExeSig1   Configuration specification   * Configuration: All used configuration list * Description: configuration description and specification * Value: current set value for this SW * Used by: subsystems in which configuration is used |

<End of document>