SW Architecture Specification

Customer

Project

**Revision History**

| **Version** | **Date** | **Change Description / Reason** | **Author** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1. Introduction 4

1.1 Definitions and Glossary 4

2. Task Design 4

2.1 Mapping of modules onto tasks 4

2.2 Event Overview 4

2.2.1 Event/Message/Alarm <Name of Event/Message/Alarm> 4

2.3 <Name of Task 1> 4

2.3.1 Messages/Events/Alarms to Be Sent 4

2.3.2 Messages/Events/Alarms to Be Received 5

3. Modules 5

3.1 <Name of module n> 5

3.1.1 Functional description 5

3.1.2 Resource consumption 6

3.1.3 Interfaces 6

4. Interfaces between modules 6

5. Memory Assignment (Memory Map) 6

5.1 Microcontroller Memory Assignment 6

5.2 EEPROM Memory Assignment 7

6. Communication Interfaces 7

6.1 CAN Communication 7

6.2 LIN Communication 7

7. System Behaviour 7

7.1 Power-up Behaviour 7

7.2 Power-down Behaviour 8

7.3 Stop Mode Behaviour 8

# Introduction

This software architecture design applies to project <project name> and is binding for all phases of the project. The document describes how the software is structured and how the various software modules communicate with each other.

## Definitions and Glossary

Maybe add own, project-specific definitions or abbreviations.

# Task Design

In this section, the task model is described. Briefly describe for each task what job it does and how the communication between the tasks is handled.



## Mapping of modules onto tasks

Show the task design block diagram, meaning a diagram that shows what modules are executed in which tasks. In addition to the task design, a block diagram is to show the software modules and their interconnections. A reference to the drawing suffices.

## Event Overview

In this section, all events/messages/alarms are described which are sent/received in the system. For each event/message/alarm it needs to be described what the event/message/alarm means.



### Event/Message/Alarm <Name of Event/Message/Alarm>

Example

Type: (Event/Alarm/Message)

Description: (e.g. Is set when HPT signal is switched off)

## <Name of Task 1>

Example

***Priority:*** <Number>

***Schedule:*** Preemptive/non-preemptive

***Task:*** Brief description of the task jobs

***Task main function:*** Name of the main function of the task



### Messages/Events/Alarms to Be Sent

|  |  |
| --- | --- |
| **Name** | **Recipient tasks** |
| **Task 1** | **Task 2** | **Task 3** | **Task 4** | **Task 5** | **Task 6** |
| ***HPT\_SIG\_CHANGED*** | X |  |  | X |  |  |
| ***HPT\_KLT\_SET*** |  | X |  |  |  |  |

### Messages/Events/Alarms to Be Received

|  |  |
| --- | --- |
| **Name** | **Sender tasks** |
| **Task 1** | **Task 2** | **Task 3** | **Task 4** | **Task 5** | **Task 6** |
| ***QWE\_DOOR\_OPEN*** |  |  |  |  |  | X |
| ***POI\_ENGINE\_IDLE*** |  |  |  |  | X |  |

# Modules

Describe which modules exist and how they are developed or whether they are externally supplied. A detailed description is provided in form MHE 8364 SW Module and Unit Design where the functions of the module which are called by other modules are described.

Document which modules exist and how they are developed and documented. If a development tool is used (INNOVATOR, model-based), then the modules are documented within the tool environment.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Model-based | MHE standard  module | Module and unit design | Code creation | Module class | Document |
| Module 1 | Innovator |  |  | In-house |  |  |
| Module 2 |  |  |  |  |  | Document name |
|  |  |  |  |  |  |  |
| Module n | Matlab/Simulink |  |  | Generated (Matlab/Simulink) |  |  |
| Operating system OSEK |  |  |  | 3rd Party - 3soft |  |  |
| CANbedded |  |  |  | 3rd Party - Vector |  |  |
| LINbedded |  |  |  | 3rd Party - Vector |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |



## <Name of module n>



### Functional description

Describe shortly the main function of the module. This is also a good place for links to the SWRS. Do not copy and/or reinvent the SWRS here!

### Resource consumption

#### RAM

#### ROM

#### NVRAM

#### CPU Load/Runtime

#### Peripherals

### Interfaces

#### Output

#### Input

# Interfaces between modules

The interfaces between modules are to be described in this section.

# Memory Assignment (Memory Map)



## Microcontroller Memory Assignment

Document the µC (RAM, ROM) memory assignment. If the assignment is documented elsewhere, then just make a reference.

Example

Memory map of RAM



Memory map of Flash



Bootloader: 0000 0000(H) ~ 0001 8000(H) (96KB)

Application: 0001 8207(H) ~ 000B 9D00(H) (646KB)

Calibration: 000B BE10(H) ~ 000B FE00(H) (15KB)

## EEPROM Memory Assignment

Document the EEPROM memory assignment. If the assignment is documented elsewhere, then just make a reference.

Example



DTC: 0000 0000(H) ~ 0000 6FFF(H) (29KB)

EOL: 0000 7000(H) ~ 0000 7FFF(H) (3KB)

# Communication Interfaces



## CAN Communication

Reference to the CAN matrix (eg. .dbc)

It’s best to make a link to the CAN matrix in the PTC, so reference is always made to the latest version.

## LIN Communication

Reference to the LIN matrix (eg. .ldf)

It’s best to make a link to the LIN scheduling in the PTC, so reference is always made to the latest version.

# System Behavior



## Power-up Behavior

To achieve a certain output situation, it is in most cases necessary to power up the tasks in a certain order or to carry out certain initializations. The power-up behavior needs to be described in this section. Diagrams using Message Sequence Charts or phase diagrams are recommended.

Example



## Power-down Behavior

In most cases it is necessary to power down the tasks in a certain order or store certain values before the system is turned off. The system’s power-down behavior needs to be described in this section. Diagrams using Message Sequence Charts or phase diagrams are recommended.

## Stop Mode Behavior

Sometimes it is necessary to run the system in stop mode, e.g. inputs need to be monitored after the engine is stopped. The system’s stop mode needs to be described in this section. Diagrams using Message Sequence Charts or phase diagrams are recommended.