INDUCTION TRAINING EXERCISE MODE COMPONENTS DESIGN DESCRIPTION TEMPLATE

Template ID	VIAS_T01_00_004
Reference	
Version	1.0
Status	Approved
Author	Mohamed Saad
Date	17/05/2016

Name	Department / Function	Title	Date
Approved by:			
Ghada ELRamly	SEPG	SEPG Leader	02/03/2010

Reviewed by:			
Ghada ELRamly	SEPG	SEPG Leader	02/03/2010

Document History

Revision	Status	Date	Author	Changes
1.0	Proposed	21/05/2012	Mohamed Saad	Initial Version

Table of Contents

List of Figures

List of Tables

1 - Introduction

1.1 - Document Scope

This document is the Components Design of the MODE component at the signs Induction ECU Project.

1.2 - Components Description

The MODE manager is a central component present at the application layer and communicates both vertically and horizontally in the architecture with the TIF, HIF and PBD components to: get the state of the push buttons “PBD_enuGetButtonState”, set the mode of the TIF “TIF_vidSetMode” and set the mode of the HIF “HIF_vidSetMode”.

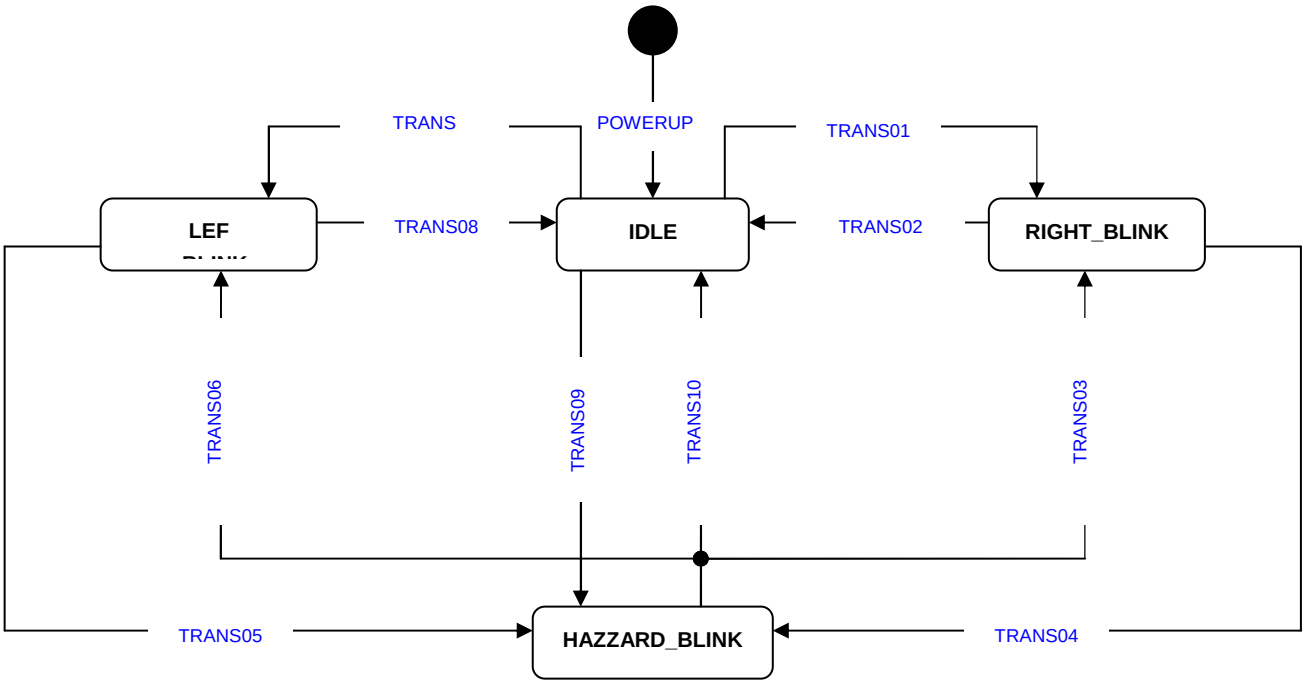

Figure 1: Context Diagram

1.3 - Software Context

Figure 1 shows the context diagram of the MODE Component.

Functionality	Description of Functionality	Data Exchanged
Input push button detection	Communication with PBD to get the status of all the push buttons	
Mode calculation	Receiving required Mode from MODE component	
Mode communication	Mode communication to HIF and TIF component to manage output	

Table 1: HIF component Functionality Description

1.4 - Design Standards & Guidelines

Introduce here the standards, guidelines and processes followed during the design activity.

#	Document Name	Reference	Version
1	Coding Rules	GEI-RD-H01-0000-095	1.10
2	Source Code Naming Rules Guidelines	GEI-RD-H01-0000-094	1.5
3	SW Design Guide	GEI-RD-H01-0000-081	G
4	Source Code Appearance Rules	GEI-RD-H01-0000-096	1.0

1.5 - Reference Documents

	Document Name	Reference	Version
1	Software Requirements Specification	SRS Induction.doc	2.0
2	Software High Level Design document of the Induction ECU	HLD_Induction.doc	1.0

2 - Terms and abbreviations

Abbreviation	Description
VIAS	Valeo Inter-branch Automotive Software
CEE	Center for Electronics Excellence
HW	Hardware
SW	Software
CMP	Component Name
ISR	Interrupt Service Routine

3 - Design Constraints and Requirements Description

3.1 - External Interfaces

3.1.1 - Std

{CDD-MODE-0001(0)}	{Covers:}	
{Assignation: }	{Test Method: Static Test}	{Justification: }

Availability:	Reused
Description:	The standard types and symbols component.
Caveats:	The header file “Std_Types.h” should be included
Configuration:	N/A

The following tables list all interfaces used from the “Std” component.

Interface Used	Type	Description	Caveats
boolean	Typedef	Is used only in conjunction with standard TRUE and FALSE symbols.	No arithmetic or logical operators are permitted with this type.
uint8	Typedef	8-bit unsigned integer data type	N/A
uint16	Typedef	16-bit unsigned integer data type	N/A
uint32	Typedef	32-bit unsigned integer data type	N/A
sint8	Typedef	8-bit signed integer data type	N/A
sint16	Typedef	16-bit signed integer data type	N/A
sint32	Typedef	32-bit signed integer data type	N/A
uint8_least	Typedef	Optimized 8-bit unsigned integer data type	N/A
uint16_least	Typedef	Optimized 16-bit unsigned integer data type	N/A
uint32_least	Typedef	Optimized 32-bit unsigned integer data type	N/A
sint8_least	Typedef	Optimized 8-bit signed integer data type	N/A
sint16_least	Typedef	Optimized 16-bit signed integer data type	N/A
sint32_least	Typedef	Optimized 32-bit signed integer data type	N/A
float32	Typedef	Single precision (32-bit) floating point number data type	N/A
float64	Typedef	Double precision (64-bit) floating point number data type	N/A

Interface Used	Type	Description	Caveats
Std_ReturnType	Typedef	Is used as the standard API return type which is shared between the RTE and the BSW modules	N/A
StatusType	Typedef	Similar to Std_ReturnType for OSEK compliance.	N/A
TRUE, FALSE	Symbol	The possible values for the boolean data type.	N/A
E_OK, E_NOT_OK	Symbol	Symbol Definitions for Std_ReturnType.	N/A
STD_HIGH	Symbol	Symbol Definitions for High state (Physical State 5 Volts or 3.3 Volts).	N/A
STD_LOW	Symbol	Symbol Definitions for Low state (Physical State 0 Volts).	N/A
STD_ACTIVE	Symbol	Logical State active	N/A
STD_IDLE	Symbol	Logical State idle	N/A
STD_ON	Symbol	Logical State On	N/A
STD_OFF	Symbol	Logical State Off	N/A

3.1.2 - PBD

{CDD-MODE-0005(0)}	{Covers: Covered Requirements Tags}	
{Assignation: }	{Test Method: Static Test}	{Justification: }

Availability:	Stubbed
Description:	MODE use PBD As an abstraction layer to get different push buttons input status
Caveats:	N/A
Configuration:	N/A

The following tables list all interfaces used from the PBD component.

Interface Used	Type	Description	Caveats
PBD_enuGetButtonState()	Service	MODE use PBD As an abstraction layer to get different push buttons input status	N/A

3.1.3 - HIF

{CDD-MODE-0005(0)}	{Covers: Covered Requirements Tags}	
{Assignation: }	{Test Method: Static Test}	{Justification: }

Availability:	Stubbed
Description:	MODE Shall define the mode for the HIF component.
Caveats:	N/A
Configuration:	N/A

The following tables list all interfaces used from the PBD component.

Interface Used	Type	Description	Caveats
HIF_vidSetMode()	Service	MODE Shall define the mode for the HIF component.	N/A

3.1.4 - TIF

{CDD-MODE-0005(0)}	{Covers: Covered Requirements Tags}	
{Assignation: }	{Test Method: Static Test}	{Justification: }

Availability:	Stubbed
Description:	MODE Shall define the mode for the TIF component.
Caveats:	N/A
Configuration:	N/A

The following tables list all interfaces used from the PBD component.

Interface Used	Type	Description	Caveats
TIF_vidSetMode()	Service	MODE Shall define the mode for the HIF component.	N/A

3.2 - Design Constraints

N/A

3.3 - Dimensioning & Performance Constraints

N/A

4 - Design Choices and Alternatives

None

5 - Components Design

5.1 - MODE

5.1.1 - Dynamic Architecture

5.1.1.1 - Mode Management

5.1.1.1.1 - Standard Mode

The following figure shows the state machine controlling the <Name of Managed Mode>.

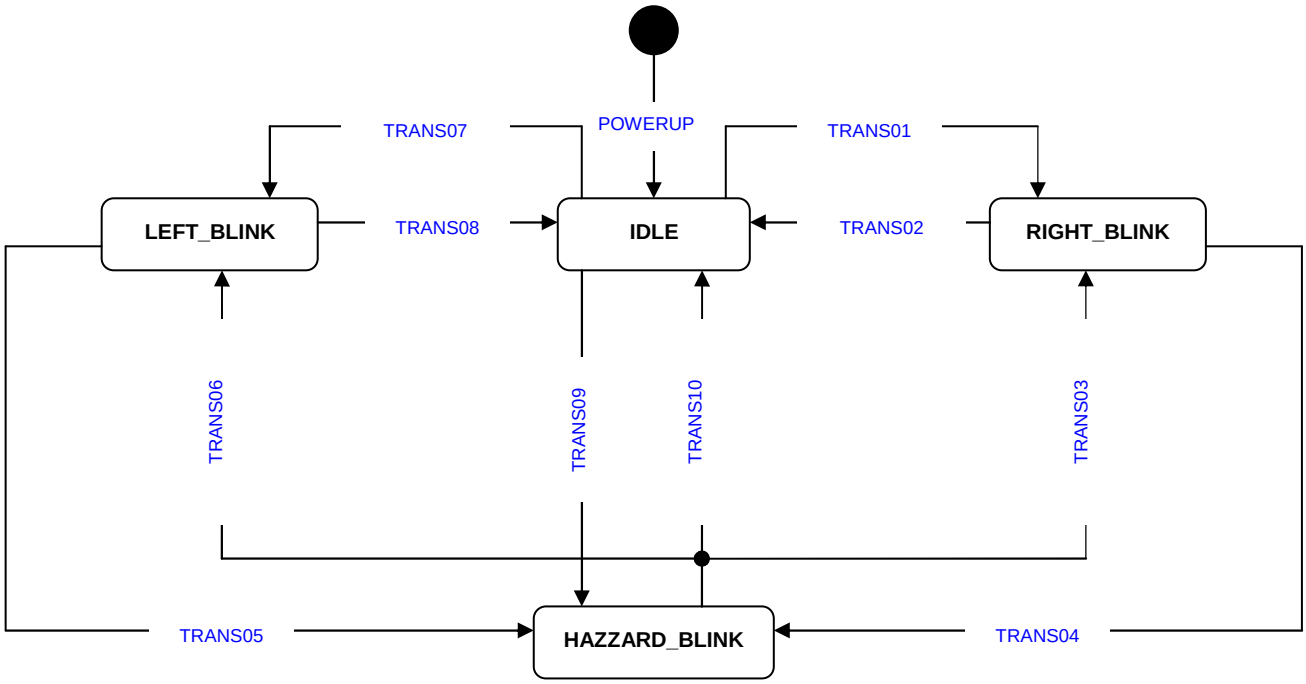

Figure 2: State Machine

5.1.1.2 - Sequence Management

5.1.2 - Static Architecture

5.1.2.1 - Sub-components

N/A

5.1.2.1.1 - Sub-components Interfaces

N/A

5.1.2.2 - API Description

5.1.2.2.1 - Public Symbols

N/A

5.1.2.2.2 - Public Types

{CDD-MODE-0002(0)}	{Covers: }	
{Assignation: }	{Test Method: Static Test}	{Justification: }

Name:	**MODE_enutModeState**
Type:	Enumeration
Range:	IDLE, RIGHT_BLINK, LEFT_BLINK, HAZZARD_BLINK
Description:	Component State Type

5.1.2.2.3 - Public Constants

N/A

5.1.2.2.4 - Exported Data

N/A

5.1.2.2.5 - Offered Services

5.1.2.2.5.1 - MODE_vidInit

{CDD-MODE-0023(0)}	{Covers: {HLD-0114(0)}}	
{Assignation: MODE}	{Test Method: Component Test}	{Justification: }

Syntax:	`void MODE_vidInit(void)`	
Description:	Initializes the MODE component	
Sync/Async:	Synchronous	
Reentrancy:	non reentrant	
Parameters (in):	None	None
Parameters (out):	None	None
Return value:	None	None
Caveats:	None	
Configuration:	None	
MemSegment:	Default compiler memories will be used	
MemClass:	Default compiler memories will be used	

{CDD-MODE-0026(0)} {Covers: {HLD-0117(0)}}
{Assignation: MODE} {Test Method: Component Test} {Justification: }

Upon startup, this service should put the system in IDLE state.

{CDD-MODE-0024(0)} {Covers: {HLD-0118(0)}}
{Assignation: MODE} {Test Method: Component Test} {Justification: }

Upon startup, both LEFT and RIGHT SSDs should be switched off. HIF and TIF modes should be Undefined.

5.1.2.2.6 - Tasks

5.1.2.2.6.1 - MODE_vidTask

{CDD-MODE-0022(0)}	{Covers: {HLD-0115(0)}}	
{Assignation: MODE}	{Test Method: Component Test}	{Justification: }

Syntax:	`void MODE_vidTask(void)`

Description:	Task of the MODE component	
Sync/Async:	Synchronous	
Reentrancy:	non reentrant	
Parameters (in):	None	None
Parameters (out):	None	None
Return value:	None	None
Caveats:	None	
Configuration:	None	
MemSegment:	Default compiler memories will be used	
MemClass:	Default compiler memories will be used	

{CDD-MODE-0027(0)} {Covers: {HLD-0116(0)}}
{Assignation: MODE} {Test Method: Component Test} {Justification: }

This task shall grab the state of different push buttons and switch between states accordingly by calling PBD_enuGetButtonState() from PBD component.

{CDD-MODE-0037(0)} {Covers: {HLD-0119(0)}}
{Assignation: MODE} {Test Method: Component Test} {Justification: }

If the system is at the IDLE state and a valid press was detected on STEP_DOWN_PB, the system shall go to the LEFT_BLINK state.

{CDD-MODE-0036(0)} {Covers: {HLD-0120(0)}}
{Assignation: MODE} {Test Method: Component Test} {Justification: }

If the system is at the IDLE state and a valid press was detected on STEP_UP_PB, the system shall go to the RIGHT_BLINK state.

{CDD-MODE-0035(0)} {Covers: {HLD-0121(0)}}
{Assignation: MODE} {Test Method: Component Test} {Justification: }

If the system is at the IDLE state and a valid press was detected on HAZZARD_PB, the system shall go to the HAZZARD_BLINK state.

{CDD-MODE-0034(0)} {Covers: {HLD-0122(0)}}
{Assignation: MODE} {Test Method: Component Test} {Justification: }

If the system is at the LEFT_BLINK state and a valid press was detected on STEP_UP_PB, the system shall go to the IDLE state.

{CDD-MODE-0044(0)} {Covers: {HLD-0123(0)}}
{Assignation: MODE} {Test Method: Component Test} {Justification: }

If the system is at the LEFT_BLINK state and a valid press was detected on STEP_DOWN_PB, the system shall remain in the LEFT_BLINK state.

{CDD-MODE-0033(0)} {Covers: {HLD-0124(0)}}
{Assignation: MODE} {Test Method: Component Test} {Justification: }

If the system is at the LEFT_BLINK state and a valid press was detected on HAZZARD_PB, the system shall go to the HAZARD_BLINK state.

{CDD-MODE-0032(0)} {Covers: {HLD-0125(0)}}

{Assignation: MODE} {Test Method: Component Test} {Justification: }

If the system is at the RIGHT_BLINK state and a valid press was detected on STEP_DOWN_PB, the system shall go to the IDLE state.

{CDD-MODE-0031(0)} {Covers: {HLD-0126(0)}}
{Assignation: MODE} {Test Method: Component Test} {Justification: }

If the system is at the RIGHT_BLINK state and a valid press was detected on STEP_UP_PB, the system shall remain in the RIGHT_BLINK state.

{CDD-MODE-0030(0)} {Covers: {HLD-0127(0)}}
{Assignation: MODE} {Test Method: Component Test} {Justification: }

If the system is at the RIGHT_BLINK state and a valid press was detected on HAZZARD_PB, the system shall go to the HAZZARD_BLINK state.

{CDD-MODE-0029(0)} {Covers: {HLD-0128(0)}}
{Assignation: MODE} {Test Method: Component Test} {Justification: }

If the system is at the HAZZARD_BLINK state and a valid press was detected on STEP_UP_PB or STEP_DOWN_PB, the system shall remain in the HAZZARD_BLINK state.

{CDD-MODE-0028(0)} {Covers: {HLD-0129(0)}}
{Assignation: MODE} {Test Method: Component Test} {Justification: }

If the system is at the HAZZARD_BLINK state and a valid press was detected on HAZZARD_PB, the system shall go back to its previous state before going to HAZZARD_BLINK state. A local static variable should save the previous component state before HAZZARD_BLINK.

{CDD-MODE-0038(0)} {Covers: }
{Assignation: } {Test Method: Component Test} {Justification: }

When Mode Component is at IDLE state MODE should call HIF and TIF set mode API with the following mode values:

HIF_vidSetMode(HIF_OFF_MODE)

TIF_vidSetMode(TIF_OFF_MODE)

{CDD-MODE-0040(0)} {Covers: }
{Assignation: } {Test Method: Component Test} {Justification: }

When Mode Component is at RIGHT_BLINK state MODE should call HIF and TIF set mode API with the following mode values:

HIF_vidSetMode(HIF_OFF_MODE)

TIF_vidSetMode(TIF_ RIGHT_BLINK _MODE).

{CDD-MODE-0041(0)} {Covers: }
{Assignation: } {Test Method: Component Test} {Justification: }

When Mode Component is at LEFT_BLINK state MODE should call HIF and TIF set mode API with the following mode values:

HIF_vidSetMode(HIF_OFF_MODE)

TIF_vidSetMode(TIF_ LEFTT_BLINK _MODE).

{CDD-MODE-0042(0)} {Covers: }
{Assignation: } {Test Method: Component Test} {Justification: }

When Mode Component is at HAZZARD_BLINK state MODE should call HIF and TIF set mode API with the following mode values:

HIF_vidSetMode(HIF_ON_MODE)

TIF_vidSetMode(TIF_OFF_MODE).

5.1.2.2.7 - Provided Callbacks

N/A

5.1.2.2.8 - Interrupt Service Routines

N/A

5.1.3 - Hardware Resources

N/A

5.1.4 - Software Resources

5.1.4.1 - SW Resources Description

N/A

5.1.4.2 - Shared SW Resources Protection Mechanism

N/A

5.1.5 - Component Configuration

N/A