**RESTRICTED ACCESS**

|  |  |  |  |
| --- | --- | --- | --- |
| **DISTRIBUTION :** | **Firm** | **To** | **Copies** |
|  | AEE-C |  |  |
|  |  |  |  |

**PP PLATFORM SOFTWARE :**

**DESIGN OF BFS COMPONENT**

OBJECT: This document is the main document of the software design for *BFS*

SUMMARY: The aim of the BFS module (Belt Function Selection) is to provide the belt function identifier that shall be played to the BFE component.

CONCLUSION:

EVOLUTION OF THE DOCUMENT

|  |  |  |  |
| --- | --- | --- | --- |
| **Issue** | **Date** | **Author** | **Motive and nature of the modifications** |
| 000 | 29.05.12 | CSA | First release (Draft release : no traceability information yet) |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

This document contains **15** pages.

**CONTENTS**

1. Documentation 4

1.1. Upper level relevant documents 4

1.2. Design Specification documents 4

1.3. Other documents 4

1.4. Glossary and definition 4

2. overview 5

2.1. Description of the component 5

2.2. Context diagram 5

3. software interface 6

3.1. Diagrams 6

3.2. Exported functions 7

3.2.1. Function *BFS\_AC\_BeltFunctionSelection\_BFS\_runBeltFunctionSelection* 7

3.3. Exported variables 8

3.4. Exported constants 9

3.5. Exported types 10

3.6. Exported macros 11

4. Software description 12

4.1. Main data 12

4.2. Functioning overview 12

5. Post Build parameters 14

6. Compilation options 15

# Documentation

## Upper level relevant documents

This section presents all the documents needed to write the design document.

|  |  |  |  |
| --- | --- | --- | --- |
| Nb | **Document** | **Reference** | **Company** |
| [A1] | Drive the Motor requirements | TF\_G\_Drive\_the\_Motor | AEE-C |
|  |  |  |  |
|  |  |  |  |

## Design Specification documents

This section presents all the documents that complete this design document.

|  |  |  |  |
| --- | --- | --- | --- |
| Nb | **Document** | **Reference** | **Company** |
| [B1] | BFE design document | BFE - SW Module Design Document .docx | AEE-C |
|  |  |  |  |
|  |  |  |  |

## Other documents

This section presents all the documents that also have been needed to write this design document.

|  |  |  |  |
| --- | --- | --- | --- |
| Nb | **Document** | **Reference** | **Company** |
| [C1] |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Glossary and definition

This section presents all the definitions and/or abbreviations used in this design document.

BFD Belt Function Decision algorithms

BFE Belt Function Execution

BFS Belt Function Selection

ECU Electronic Controller Unit

NVM Non Volatile Memory

PAL Power Abstraction Layer

RTE Real Time Environment

SW Software

# overview

The purpose of this section is to present and define all the services realized by BFS.

## Description of the component

The BFS component is an applicative software module, that shall provide to the BFE the belt function identifier that shall be executed.

Depending of the current executed cycle (provided by the BFE), and the triggering/interrupt requests (provided by the BFD), it will assign a weight to each requested cycle. And finally the belt function with the higher weight will be selected.

This module is a MBD software component developed in an Autosar context.

So the file organization and the code generated will depend of these 2 constraints.

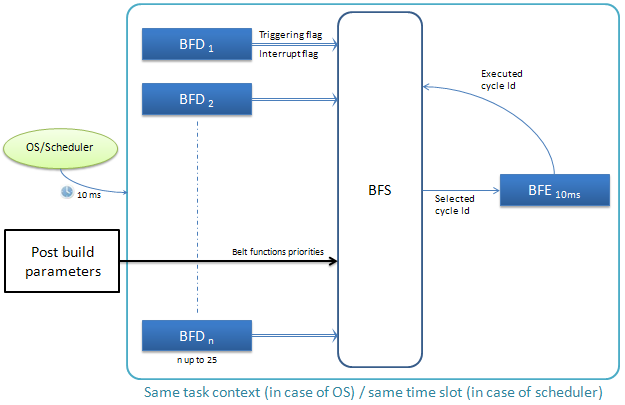
If this component shall be integrated in a Non-Autosar project context, a wrapper will be used to perform the integration with non-Autosar modules, and a mini RTE will be integrated to perform the communication with other SW-Modules developed in Autosar context (i.e. BFS/BFE/PAL)

## Context diagram

This diagram describes the main flows between BFS modules and other modules.

The BFS SW module can manage up to 25 different belt functions identifiers. For each of these belt functions, a BFD module shall provide to the BFS the triggering flag state and the interrupt flag state periodically.

In the same time the belt function currently executed shall be monitored.

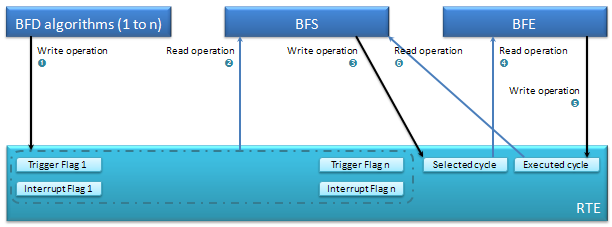


# software interface

## Diagrams

This module was developed in an Autosar project context. So the BFS uses RTE interfaces. These RTE interfaces shall be handled by the RTE layer in case of an integration in a full autosar project or by a Wrapper extern module in case of an integration in a non-autosar project.

For the Wrapper module see **TBD.**



Interface ➋ :

To get the Trigger flags and Interrupt flags status for all belt functions supported. These flags are handled by the RTE layer.

Void Rte\_Read\_prrAlgo*XX*\_Flags\_b8InterruptFlag ( b8AlgorithmFlagsStatusType \* b8AlgoFlag )

Void Rte\_Read\_prrAlgo*XX*\_Flags\_b8TriggerFlag ( b8AlgorithmFlagsStatusType \* b8AlgoFlag )

*XX can take values from 01 to 25*

Interface ➌ :

The BFS shall write the Selected cycle in the RTE layer to provide this information to the BFE component.

Void Rte\_Write\_psrSelectedCycle\_u8CycleNumber ( uint8 u8SelectedCycle )

Interface ➍ :

The BFE shall be able to access to the Selected cycle information. This connection shall be ensured by the RTE layer and the SW architecture.

Interface ➎ :

The BFE shall refresh periodically the belt function executed, and provide this information to all SW modules that require this information. Since the BFE was also designed in an Autosar context, this connection shall be ensured by the RTE layer and the SW architecture.

Interface ➏ :

The BFS shall be able to access to the current executed cycle information returned by the BFE at the previous time slot. This connection shall be ensured by the RTE layer and the SW architecture.

Uint8 Rte\_IRead\_BFS\_runBeltFunctionSelection\_prrExecutedCycle\_u8CycleNumber (void)

## Exported functions

All the following functions are exported by the component.

### Function *BFS\_AC\_BeltFunctionSelection\_BFS\_runBeltFunctionSelection*

* Object :

DSG\_BFS\_00004:

The aim of this function is to output the belt function identifier that shall be executed by lower applicative layers (BFE).

[COVERS: ]

* Entering and outgoing parameters:

None.

* Dynamic aspect :

This function shall be called every 10ms in the same time slot than BFD algorithms and BFE.

* Who :

This function shall be called in the OS task context or directly by a scheduler state machine.

* External items used :

None.

* Returned values :

None.

* Constraints :

*(Architecture constraint)*

This function shall be called every 10 ms after the refreshment of all triggering and interrupt flags by the BFD algorithms and before the execution of the BFE process in charge of the steps/cycles scheduling.

## Exported variables

All the following variables are exported by the component.

|  |  |  |
| --- | --- | --- |
| **Variable name** | | |
|  | Type |  |
|  | Object |  |
|  | Unit |  |
|  | Range |  |
|  | Safety |  |
|  | Constraint |  |

## Exported constants

All the following constants are exported by the component.

|  |  |  |
| --- | --- | --- |
| **KB8\_ALGO\_FLAG\_ON** | | |
|  | Type | Unsigned char 8 bits |
|  | Object | Value that shall be used as argument by BFD algorithms when they request the triggering or the interrupt of their belt function |
|  | Safety | None |
|  | Constraint | This constant shall be defined in Rte\_Type header file |
|  | Value | 0xAA |

|  |  |  |
| --- | --- | --- |
| **KB8\_ALGO\_FLAG\_OFF** | | |
|  | Type | Unsigned char 8 bits |
|  | Object | Value that shall be used as argument by BFD algorithms when they don’t request the triggering or the interrupt of their belt function |
|  | Safety | None |
|  | Constraint | This constant shall be defined in Rte\_Type header file |
|  | Value | 0x55 |

|  |  |  |
| --- | --- | --- |
| **KU8\_NO\_CYCLE** | | |
|  | Type | Unsigned char 8 bits |
|  | Object | Value taken by the Selected Cycle and Executed Cycle signals when no belt function is requested or executed. |
|  | Safety | None |
|  | Constraint | This constant shall be defined in Rte\_Type header file |
|  | Value | 0xFF |

|  |  |  |
| --- | --- | --- |
| **KU8\_NB\_MAX\_CYCLES** | | |
|  | Type | Unsigned char 8 bits |
|  | Object | Upper limit of belt function supported by the ECU. |
|  | Safety | This value will determine the size of many array on the software. |
|  | Constraint | This constant shall be defined in Rte\_Type header file |
|  | Value | 25 by default |

## Exported types

All the following types are exported by the component.

|  |  |  |
| --- | --- | --- |
| **b8AlgorithmFlagsStatusType** | | |
|  | Object | Data type to set or clear flags in triggering and interrupt services that shall be called by BFD |
|  | Safety | None |
|  | Constraint | Valid range : KB8\_ALGO\_FLAG\_OFF / KB8\_ALGO\_FLAG\_ON  This data type shall be defined in Rte\_Type header file. |

## Exported macros

All the following macros are exported by the component.

|  |  |  |
| --- | --- | --- |
| **Macro name** | | |
|  | Type |  |
|  | Object |  |
|  | Unit |  |
|  | Range |  |
|  | Safety |  |
|  | Constraint |  |

# Software description

## Main data

When entering the BFS main function, a snapshot of all interrupt & triggering flags is done. The status of these flags are stored in 2 arrays. This will be more useful to manage them in the FOR loop of the BFS module.

DSG\_BFS\_00001:

|  |  |  |
| --- | --- | --- |
| **au8IntFlags** | | |
|  | Type | Array of unsigned char |
|  | Object | Local array containing the interrupt flag states for the 25 supported BFD algorithms. |
|  | Unit | None |
|  | Range | KB8\_ALGO\_FLAG\_OFF / KB8\_ALGO\_FLAG\_ON |
|  | Safety | None |
|  | Constraint | All tests done on the flags shall be done only with the value KB8\_ALGO\_FLAG\_ON to manage the possible corruption of RAM data. |

[COVERS: ]

DSG\_BFS\_00002:

|  |  |  |
| --- | --- | --- |
| **au8TrigFlags** | | |
|  | Type | Array of unsigned char |
|  | Object | Local array containing the triggering flag states for the 25 supported BFD algorithms. |
|  | Unit | None |
|  | Range | KB8\_ALGO\_FLAG\_OFF / KB8\_ALGO\_FLAG\_ON |
|  | Safety | None |
|  | Constraint | All tests done on the flags shall be done only with the value KB8\_ALGO\_FLAG\_ON to manage the possible corruption of RAM data. |

[COVERS: ]

DSG\_BFS\_00003:

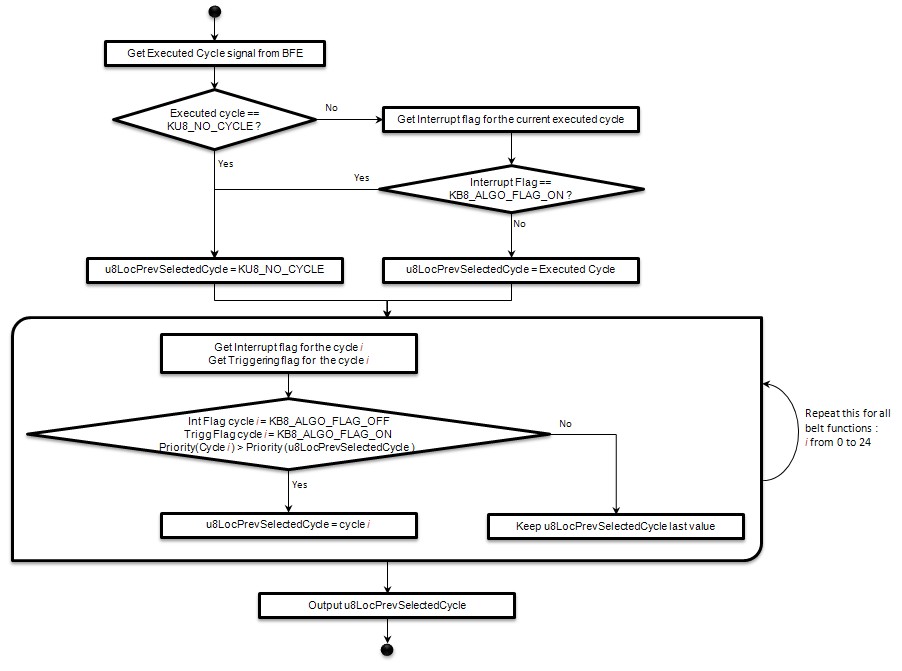
|  |  |  |
| --- | --- | --- |
| **u8LocPrevSelectedCycle** | | |
|  | Type | Unsigned char |
|  | Object | Local data used to store the belt function Identifier during the BFS loop processing. |
|  | Unit | None |
|  | Range | [0;KU8\_NB\_MAX\_CYCLES-1] ∪ KU8\_NO\_CYCLE |
|  | Safety | None |
|  | Constraint | This data shall be initialized with the BFE executed cycle value at the first iteration of the FOR loop. |

[COVERS: ]

## Functioning overview

The complete behavior of the BFS main function is explained on the following diagram.

.

. 

# Post Build parameters

The BFS component will use an array containing the belt functions priorities to affect a weight to each triggering requests send by BFD algorithms.

So the width of this array will be equal to the max number of belt function supported by the BFE (i.e. 25).

And unsigned 8 bits data type will be enough to manage the priority for these 25 belt functions in worst case.

# Compilation options

There are no compilation options for the BFS component.