Integration Manual – ApXcp

Table of Contents

[1 Dependencies 2](#_Toc357692818)

[1.1 SWCs 2](#_Toc357692819)

[1.2 Functions to be provided to Integration Project 2](#_Toc357692820)

[2 Configuration 3](#_Toc357692821)

[2.1 Build Time Config 3](#_Toc357692822)

[2.2 Configuration Files to be provided by Integration Project 3](#_Toc357692823)

[2.2.1 Da Vinci Config generation 3](#_Toc357692824)

[2.2.2 Manual Configuration Changes 3](#_Toc357692825)

[3 Integration 4](#_Toc357692826)

[3.1 Required Global Data Inputs 4](#_Toc357692827)

[3.2 Optional Global Data Inputs 4](#_Toc357692828)

[3.3 Specific Include Path present 4](#_Toc357692829)

[4 Runnable Scheduling 5](#_Toc357692830)

[5 Memory Mapping 6](#_Toc357692831)

[5.1 Mapping 6](#_Toc357692832)

[5.2 Usage 6](#_Toc357692833)

[5.3 NvM Blocks 6](#_Toc357692834)

[6 Compiler Settings 6](#_Toc357692835)

[6.1 Preprocessor MACRO 6](#_Toc357692836)

[6.2 Optimization Settings 6](#_Toc357692837)

[7 Revision Control Log 7](#_Toc357692838)

# Dependencies

## GetSetIndexes() and GetPersIndexes()

### Overview

The integration specific functions below, GetSetIndexes and GetPersIndexes, are used to return an array of tuning identifiers to the tune-on-the-fly (or online calibration as defined in the XCP specs) function. They are used to copy the active sets or personalities into RAM. The functions should be implemented in an integration specific component that determines the desired personality and desired set identifiers sent to tuning select authority SWC.

If tune-on-the-fly functionality is disabled, the functions below do not need to be defined.

If tune-on-the-fly is enabled and the lookup functions are disabled, the functions below do not need to be defined. When a copy cal page XCP command is called, the personalities are copied from 0 to a max limit based on the active tuning set that is currently being used. In the case of the tuning sets being copied, only the active tuning set is copied in to the RAM.

|  |  |
| --- | --- |
| **Module** | **Required API** |
| **<Integration Specific Module>** | Function outline defined below: |

### Function Prototypes

The actual implementation of the function will vary between programs. However, the function should be structured so the passed arguments are of the same time to provide a common interface to the XCP functions.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | GetSetIndexes or GetPersndexes | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | NumOfSets\_Cnt\_T\_u8 or NumOfPers\_Cnt\_T\_u8 | uint8 | I | 0 | 255 |  |
|  | data | uint8\* | I/O | 0 | 255 |  |
| **Return Value** | N/A |  |  |  |  |  |

### Example 1 -- BMW

The following example illustrates a similar situation found in the BMW program. Active personalities are determined by coding bits. Using the GetPersIndexes function, the function will loop through the personalities to find the index of the personality that contains the matching coding ID. The indexes are returned and the active personalities are copied into RAM.



## ProcessXCPPID()

### Overview

This function must be available to this module to call by the integration manual. This should be provided by the CMS component which does the processing on XCP PIDs. This function is required to allow the CMS component to figure out when an XCP PID is finished being read or written.

|  |  |
| --- | --- |
| **Module** | **Required API** |
| **CMS Common** | Function outline defined below: |

### Function Prototypes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | ProcessXCPPID | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | Size\_Cnt\_T\_u08 | uint8 | I | 1 | 8 |  |
| **Return Value** | N/A |  |  |  |  |  |

# Configuration

## Build Time Config

|  |  |  |
| --- | --- | --- |
| Modules | Notes |  |
| None |  |  |

## Configuration Files to be provided by Integration Project

### Da Vinci Parameter Configuration Changes

|  |  |  |
| --- | --- | --- |
| Parameter | Notes | SWC |
| D\_NUMOFVLDEEMEMRGNS\_CNT\_U08 | Generated in Ap\_ApXcp.Cfg.h if external EEPROM access is set to STD\_ON in Configurator. Value will be determined by the EEPROM memory access defined in the ECUC file. | ApXcp |
| BC\_XCP\_EXTEEPACCESS | Set to STD\_ON in Configurator if XCP access to external EEPROM is required. | ApXcp |
| BC\_XCP\_TUNEONFLY | Set to STD\_ON in Configuartor if tune-on-the-fly support is required. | ApXcp |
| BC\_XCP\_PERSINDEXLOOKUP | Set to STD\_ON in Configurator if a function call to GetPersIndexes is required to return the indexes of active personalities. Note: BC\_XCP\_TUNEONFLY must be set to STD\_ON | ApXcp |
| BC\_XCP\_SETINDEXLOOKUP | Set to STD\_ON in Configurator if a function call to GetSetIndexes is required to return the indexes of active sets. Note: BC\_XCP\_TUNEONFLY must be set to STD\_ON | ApXcp |
| <Non Trusted Function Stack Size> | This needs to be configured (to a non-zero value) for the task or ISR that handles the XCP commands. This is required to support application switching for XCP writes. Please note this is only required if a program has non-trusted applications configured. | Os |
| <Trusted/Non Trusted functions for XCP writes> | If a program has multiple applications configured, non-trusted or trusted functions must be configured for each application. This is required to allow the XCP write commands to switch context before performing the write. These functions should end up calling ApXcpWriteCommon() to perform the write.  Ex:  NtWrapS\_XcpWriteAp8()  NtWrapS\_XcpWriteAp9()  TWrapS\_XcpWriteAp0 | Os |
| BC\_XCP\_TOTFRAMSIZE\_UNT\_32 | This should be set to the same size as the amount of RAM allocated to TOTF. Also used as the size of the buffer if Segment16 is enabled. | ApXcp |
| BC\_XCP\_SEG[x]ENABLE | This should be set to STD\_ON to enable the number segment, where [x] = Segment number. | ApXcp |
| BC\_XCP\_MAXPERCOPY\_CNT\_U8 | Used with segment 0, the number represents the number of tuning personalities to copy into RAM. Must be set to at least 1. | ApXcp |
| CalConstNumOfPerSymbol | Set to the number of tuning personalties constant. Typically this is defined in CalConstants.h with the name D\_NUMOFPERS\_CNT\_U16. | ApXcp |
| BC\_XCP\_MAXSETCOPY\_CNT\_U8 | Used with segment 1, the number represents the number of tuning sets to copy into RAM. Must be set to at least 1. | ApXcp |
| CalConstNumOfSetSymbol | Set to the number of tuning sets constant. Typically this is defined in CalConstants.h with the name D\_NUMOFTUNSETS\_CNT\_U16. | ApAcp |
| BC\_XCP\_SEG16LNKSYMB | Used with segment 16. The linker symbol represents the start address of the custom memory range to copy into RAM. | ApXcp |
| BC\_XCP\_SEG16LNKSYMBLEN\_CNT\_U8 | Used with segment 16. The linker symbol represents the size of the custom memory range to copy into RAM. | ApXcp |

### DaVinci Interrupt Configuration Changes

|  |  |  |  |
| --- | --- | --- | --- |
| ISR Name | VIM # | Priority Dependency | Notes |
|  |  |  |  |

### Manual Configuration Changes

|  |  |  |
| --- | --- | --- |
| Constant | Notes | SWC |
|  |  |  |

### GENy Configuration Changes

|  |  |  |
| --- | --- | --- |
| Parameter | Notes | SWC |
| User configuration file | ApXcp component functionality relies on a callout function for XCP memory writing from the XCP module as delivered by vector. To enable this callout function, a user configuration file typically needs to be included in the GENy configuration to enable the appropriate build constant. See attached file below | GENy XCP configuration |
| General Settings->Enable Calibration | This should be enabled to allow download commands | GENy XCP configuration |
| General Settings->Memory Write Protection | This should be enabled to allow checking of XCP access on the addresses requested for writing. | GENy XCP configuration |
| General Settings->Memory Read Protection | This should be enabled to allow checking of XCP access on the addresses requested for reading. | GENy XCP configuration |
| EEPROM Access->Read Access | This needs to be enabled if external EEPROM is available and needs to be read through XCP | GENy XCP configuration |
| EEPROM Access->Write Access | This needs to be enabled if external EEPROM is available and needs to be written through XCP | GENy XCP configuration |
| Standard Commands->User Defined Command | This needs to be enabled for CMS XCP support | GENy XCP configuration |
| Block Transfer->Block Upload | Needs to be enabled to support large XCP PID reads | GENy XCP configuration |
| Block Transfer->Block Download | Needs to be enabled to support large XCP PID writes | GENy XCP configuration |
| Page Switching->Page Switching | Needs to be enabled only if Tune On the Fly Support is turned on | GENy XCP configuration |
| Page Switching->General Paging Info | Needs to be enabled only if Tune On the Fly Support is turned on | GENy XCP configuration |
| Page Switching->Copy Page | Needs to be enabled only if Tune On the Fly Support is turned on | GENy XCP configuration |
|  |  |  |

# Integration

## Required Global Data Inputs

<Mention any global variable that this component requires for other components>

## Required Global Data Outputs

<Mention any global variable that this component requires for other components>

## Specific Include Path present

Yes

# Runnable Scheduling

This section specifies the required runnable scheduling.

|  |  |  |
| --- | --- | --- |
| Init | Scheduling Requirements | Trigger |
| ApXcp\_Init | Should be called prior to the start of the O/S (typically EcuStartup\_Init1) | Once At Init |

|  |  |  |
| --- | --- | --- |
| Runnable | Scheduling Requirements | Trigger |
| ApXcp\_Per1 | None | RTE (10ms) |
| DAQ\_2msTL |  | RTE (2ms) |
| DAQ\_1msTL | This function should only be scheduled if 1ms DAQ support is required. If not, it needs to be deleted from the integration project in the ApXcp Component. | RTE (1ms) |

**.**

# Memory Mapping

## Mapping

|  |  |  |
| --- | --- | --- |
| Memory Section | Contents | Notes |
| < Memory mapping Info> |  |  |
|  |  |  |

\* Each …START\_SEC… constant is terminated by a …STOP\_SEC… constant as specified in the AUTOSAR Memory Mapping requirements.

## Usage

|  |  |  |
| --- | --- | --- |
| Feature | RAM | ROM |
| <Memmap usuage info> |  |  |

Table 1: ARM Cortex R4 Memory Usage

## Non RTE NvM Blocks

|  |
| --- |
| Block Name |
| <NVM block used Non RTE functions > |

Note : Size of the NVM block if configured in developer

## RTE NvM Blocks

|  |
| --- |
| Block Name |
| <NVM block used in RTE functions > |

Note : Size of the NVM block if configured in developer

# Compiler Settings

## Preprocessor MACRO

<Define all the preprocessor Macros needed and conditions when needed>.

## Optimization Settings

<Define Optimization levels that are needed and conditions when needed>.

# Revision Control Log

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev #** | **Change Description** | **Date** | **Author** |
| 1 | Initial version | 30-July-13 | KJS |
| 2 | Updates for better description of configuration requirements | 29-Aug-13 | LWW |
| 3 | Added updates for new configuration constants generated with tune on the fly. | 10-Oct-13 | KJS |