AutoSAR FEE Parameter Configuration Document

ABSTRACT

AutoSAR Flash EEPROM Emulation (AutoSAR FEE) driver utilizes Code Generation Tool to generate the configuration parameters required for EEPROM emulation. Code Generation Tool is used to configure parameters like which Flash Sectors to use, the number of Blocks, Block Size etc. for EEPROM emulation. Code Generation Tool generates two files (Fee_cfg.h & Fee_cfg.c) depending on the configuration.

This document describes the parameters used by Code Generation Tool to generate the AutoSAR FEE Configuration parameters.

Revision History

Version	Release Date	Author	Comment
[redacted]			
1.1	10/10/2012	Vishwanath Reddy	Add configuration parameter FEE_NUMBER_OF_VIRTUAL_SECTORS_EEP1

FEE_NUMBER_OF_EIGHTBYTEWRITES	New
FEE_CHECK_BANK7_ACCESS	New
FEE_TOTAL_BLOCKS_DATASETS	New
FEE_VIRTUALSECTOR_SIZE	New
FEE_PHYSICALSECTOR_SIZE	New
FEE_GENERATE_DEVICEANDVIRTUALSECTORSTRUC	New

Table of Contents

1 Introduction

The following sections describe each parameter in the Fee_ParamDef.axml file used by Code Generation Tool and the corresponding configuration parameter generated. Code Generation Tool generates two files (Fee_Cfg.c and Fee_cfg.h) depending on the configuration values. This section describes each configuration value in the above two files and their relation to the parameter defined in the Fee_ParamDef.axml file.

1.1 FEE Published information

1.1.1 Block OverHead

Parameter defined in Fee_ParamDef.axml	FeeBlockOverhead
Description	Indicates the number of bytes used for Block Header.
Generated configuration	FEE_BLOCK_OVERHEAD is set to the value assigned to FeeBlockOverhead.
Default Value	0x18
Parameter Range	Fixed to 0x18.
Parameter Type	uint8
Target file	Fee_cfg.h

1.1.2 Maximum Blocking Time

Parameter defined in Fee_ParamDef.axml	FeeMaximumBlockingTime
Description	Indicates the maximum allowed blocking time for any Fee call.
Generated configuration	FEE_MAXIMUM_BLOCKING_TIME is set to the value assigned to FeeMaximumBlockingTime.
Default Value	600.00
Parameter Range	Fixed to 600 µs.
Parameter Type	float
Target file	Fee_cfg.h

1.1.3 Page OverHead

Parameter defined in Fee_ParamDef.axml	FeePageOverhead
Description	Indicates the Page Overhead in bytes.
Generated configuration	FEE_PAGE_OVERHEAD is set to the value assigned to FeePageOverhead. (0x0)
Default Value	0x0
Parameter Range	Fixed to 0x0.
Parameter Type	uint8
Target File	Fee_cfg.h

1.1.4 Sector OverHead

Parameter defined in Fee_ParamDef.axml	FeeVirtualSectorOverhead
Description	Indicates the number of bytes used for Virtual Sector Header.
Generated configuration	FEE_VIRTUAL_SECTOR_OVERHEAD is set to the value assigned to FeeVirtualSectorOverhead (0x10).
Default Value	0x10
Parameter Range	Fixed to 0x10.
Parameter Type	uint8
Target File	Fee_cfg.h

1.2 FEE General Settings

1.2.1 Virtual Page size

Parameter defined in Fee_ParamDef.axml	FeeVirtualPageSize
Description	Indicates the virtual page size in bytes.
Generated configuration	FEE_VIRTUAL_PAGE_SIZE is set to the value assigned to FeeVirtualPageSize. (0x8)
Default Value	0x8
Parameter Range	Fixed to 0x8.
Parameter Type	uint8
Target File	Fee_cfg.h

1.2.2 Driver Index

Parameter defined in Fee_ParamDef.axml	FeeIndex
Description	Instance ID of FEE module. Should always be 0x0.
Generated configuration	FEE_INDEX is set to the value assigned to FeeIndex. (0x0)
Default Value	0x0
Parameter Range	Fixed to 0x0.
Parameter Type	uint8
Target File	Fee_cfg.h

1.2.3 Error Notification

Parameter defined in Fee_ParamDef.axml	FeeNvmJobErrorNotification
Description	Call back function to notify a Job Error.
Generated configuration	FEE_NVM_JOB_ERROR_NOTIFICATION is set to the defined function name. This is relevant only if Polling mode is OFF.
Default Value	NvM_JobErrorNotification
Parameter Range	User defined function name.
Parameter Type	string
Target File	Fee_cfg.h

1.2.4 End Notification

Parameter defined in Fee_ParamDef.axml	FeeNvmJobEndNotification
Description	Call back function to notify end of a Job.
Generated configuration	FEE_NVM_JOB_END_NOTIFICATION is set to the defined function name. This is relevant only if Polling mode is OFF.
Default Value	NvM_JobEndNotification
Parameter Range	User defined function name.
Parameter Type	string
Target File	Fee_cfg.h

1.2.5 Frequency

Parameter defined in Fee_ParamDef.axml	FeeFrequency
Description	Device operating frequency in MHz.
Generated configuration	FEE_OPERATING_FREQUENCY is set to the value assigned to FeeFrequency. FeeFrequency is equivalent to the HCLK frequency in the TMS570 clock tree. It is recommended to copy the value of HCLK obtained by configuring the TMS570 clock tree during MCU configuration to this parameter.
Default Value	160.0
Parameter Range	Device dependent parameter. Refer to the device datasheet to know the range.
Parameter Type	float
Target File	Fee_cfg.h

1.2.6 Enable Polling mode

Parameter defined in Fee_ParamDef.axml	FeePollingMode
Description	Indicates if polling mode is enabled/disabled.
Generated configuration	FEE_POLLING_MODE is set to STD_ON if polling is enabled else it is set to STD_OFF. Currently, this parameter should be always STD_ON.
Default Value	STD_ON
Parameter Range	STD_ON/STD_OFF
Parameter Type	Boolean
Target File	Fee_cfg.h

1.2.7 Enable Error Correction

Parameter defined in Fee_ParamDef.axml	FeeEnableErrorCorrection
Description	Indicates if error correction is enabled.
Generated configuration	FEE_FLASH_ERROR_CORRECTION_ENABLE Is set to STD_ON if Error Correction is enabled else it is set to STD_OFF. This parameter is not used anymore.
Default Value	STD_OFF
Parameter Range	STD_ON/STD_OFF
Parameter Type	Boolean
Target File	Fee_cfg.h

1.2.8 Error Correction Handling

Parameter defined in Fee_ParamDef.axml	FeeFlashErrCorrHandlingType
Description	Indicates desired action to be taken on detection of bit errors.
Generated configuration	FEE_FLASH_ERROR_CORRECTION_HANDLING is set to the value assigned to FeeFlashErrCorrHandlingType. Only Fee_None is supported.
Default Value	Fee_None
Parameter Range	Fee_None or Fee_Fix
Parameter Type	enum {Fee_none, Fee_Fix}
Target File	Fee_cfg.h

1.2.9 Cyclic Redundancy Check

Parameter defined in Fee_ParamDef.axml	FeeCRCEnable
Description	Pre-processor switch to enable the CRC for blocks. STD_ON: CRC for blocks is enabled. STD_OFF:CRC disabled
Generated configuration	FEE_FLASH_CRC_ENABLE is set to STD_ON if CRC check is enabled else it is set to STD_OFF. If enabled, 16 bit CRC of the block is generated.
Default Value	STD_OFF
Parameter Range	STD_ON / STD_OFF
Parameter Type	Boolean
Target File	Fee_cfg.h

1.2.10 Block Write counter save

Parameter defined in Fee_ParamDef.axml	FeeWriteCounterSave
Description	Pre-processor switch to enable the block write counter. STD_ON: Block Write counter is enabled. STD_OFF:Block Write counter is disabled
Generated configuration	FEE_FLASH_WRITECOUNTER_SAVE is set to STD_ON if block write counter save is enabled else it is set to STD_OFF.
Default Value	STD_OFF
Parameter Range	STD_ON / STD_OFF
Parameter Type	Boolean
Target File	Fee_cfg.h

1.2.11 Number of EEPs

Parameter defined in Fee_ParamDef.axml	FeeNumberOfEEPS
Description	Number of EEP's configured. 1 - Only one EEP configured. All Virtual Sectors can be used by this EEP. 2 - Two EEP's configured. Each EEP can use two Virtual Sectors.
Generated configuration	FEE_NUMBER_OF_EEPS is set to 1 if all virtual sectors are used by one EEP. If virtual sectors are shared between two EEPs, it is set to 2.
Default Value	1
Parameter Range	1/2
Parameter Type	Uint8
Target File	Fee_cfg.h

Note: If GUI is calculating this parameter from the configuration, this parameter may not be present in BSWMD file.

1.2.12 Development error Detect

Parameter defined in Fee_ParamDef.axml	FeeDevErrorDetect
Description	Pre-processor switch to enable and disable development error detection. true: Development error detection enabled. false: Development error detection disabled.
Generated configuration	FEE_DEV_ERROR_DETECT Is set to STD_ON if Set Mode supported is required else it is set to STD_OFF.
Default Value	STD_OFF
Parameter Range	STD_ON/STD_OFF
Parameter Type	boolean
Target File	Fee_cfg.h

1.2.13 Non configured blocks to copy

Parameter defined in Fee_ParamDef.axml	FeeNumberOfUnconfiguredBlocksToCopy
Description	Defines the maximum number of non configured blocks to be copied during virtual sector swap. This parameter is used if project configures and writes 10 blocks at start of the project, then reduce the number of blocks to let's say 8 blocks, but still want the other two blocks to be present in Flash. In this case, project should configure this parameter to 2.
Generated configuration	FEE_NUMBER_OF_UNCONFIGUREDBLOCKSTOCOPY is set to defined value.
Default Value	0
Parameter Range	0 -255
Parameter Type	Uint8
Target File	Fee_cfg.h

1.2.14 Number of eight byte writes

Parameter defined in Fee_ParamDef.axml	FeeNumberOfEightByteWrites
Description	Defines the number of 8 byte writes to be done in main function call. If configured to 2, main function writes 16 bytes per call.
Generated configuration	FEE_NUMBER_OF_EIGHTBYTEWRITES is set to defined value.
Default Value	1
Parameter Range	1-255
Parameter Type	Uint8

Target File	Fee_cfg.h

1.2.15 Check BANK7 Address Range

Parameter defined in Fee_ParamDef.axml	FEE_CHECK_BANK7_ACCESS
Description	Pre processor switch to enable EEPROM address range check during read/write.
Generated configuration	FEE_CHECK_BANK7_ACCESS is set to defined value.
Default Value	STD_OFF
Parameter Range	STD_ON/STD_OFF
Parameter Type	boolean
Target File	Fee_cfg.h

1.2.16 Total Blocks and Data Sets

Parameter defined in Fee_ParamDef.axml	FEE_TOTAL_BLOCKS_DATASETS
Description	Macro to indicate total blocks and data sets configured.
Generated configuration	FEE_TOTAL_BLOCKS_DATASETS is set to defined value.
Default Value	1
Parameter Range	1-65536
Parameter Type	Uint16
Target File	Fee_cfg.h

Note: This configuration should calculate total number of blocks and data sets configured in the structure Fee_BlockConfiguration. For example, if 5 blocks are configured with block 1 has 1 data set, block 2 has 4 data sets, block 3 has 2 data sets, block 4 has 5 data sets, block 5 has 3 data sets, then FEE_TOTAL_BLOCKS_DATASETS = [(1)+(4)+(2)+(5)+(3)] = 15(sum of all data sets).
If GUI is calculating this parameter from the configuration, this parameter may not be present in BSWMD file.

1.2.17 Generate Device and Virtual sector structures

Parameter defined in Fee_ParamDef.axml	FEE_GENERATE_DEVICEANDVIRTUALSECTORSTRUC
Description	Pre processor switch to enable/disable generation of Device and Virtual sector structures.
Generated configuration	FEE_GENERATE_DEVICEANDVIRTUALSECTORSTRUC is set to defined value.
Default Value	STD_OFF
Parameter Range	STD_ON/STD_OFF
Parameter Type	boolean
Target File	Fee_cfg.h

Note: When this macro is turned ON, Fee_VirtualSectorConfiguration and Device_FlashDevice are generated during run time.

1.2.18 Required Virtual Sector Size

Parameter defined in Fee_ParamDef.axml	FEE_VIRTUALSECTOR_SIZE
Description	Macro to indicate the required virtual sector size in kilo bytes. This macro is only used when FEE_GENERATE_DEVICEANDVIRTUALSECTORS TRUC is STD_ON. Based on FEE_VIRTUALSECTOR_SIZE and FEE_NUMBER_OF_VIRTUAL_SECTORS, elements of the structure Fee_VirtualSectorConfiguration will be populated during runtime.
Generated configuration	FEE_VIRTUALSECTOR_SIZE is set to defined value.
Default Value	None
Parameter Range	4-32(see below note)
Parameter Type	Uint8
Target File	Fee_cfg.h

Note: Depending on the device, parameter range can be different.
For TMS570LS12xx/11xx family devices, FEE bank is 4*16KB. Macro can take a value of 16 or 32. For TMS570LS09xx, TMS570LS07xx, TMS570LS05xx family devices, FEE bank is 16*4KB. Macro can take a value of 4 or 8 or 12 or 16 or 32.
FEE_VIRTUALSECTOR_SIZE * FEE_NUMBER_OF_VIRTUAL_SECTORS should not exceed the total available FEE bank size on device.

1.2.19 FEE bank Physical Sector Size

Parameter defined in Fee_ParamDef.axml	FEE_PHYSICALSECTOR_SIZE
Description	Macro to indicate the physical sector size on the device. This macro is only used when FEE_GENERATE_DEVICEANDVIRTUALSECTORS TRUC is STD_OFF. This parameter is used to select device specific files.
Generated configuration	FEE_PHYSICALSECTOR_SIZE is set to defined value.
Default Value	None.
Parameter Range	4/16
Parameter Type	Uint8
Target File	Fee_cfg.h

Note: This macro can only have 4/16 as value.
For TMS570LS12xx/11xx family devices, sector size is 16. For TMS570LS09xx, TMS570LS07xx, TMS570LS05xx family devices, sector size is 4.

1.3 Number of Blocks

1.3.1 Blocks

Parameter defined in Fee_ParamDef.axml	FeeNumberOfBlocks
Description	Defines the number of Data Blocks used for EEPROM emulation. This is sum of all the blocks configured on EEP1 and EEP2.
Generated configuration	FEE_NUMBER_OF_BLOCKS is set to the defined value.
Default Value	0x1
Parameter Range	0x1 to 0xFFFE
Parameter Type	uint16
Target File	Fee_cfg.h

Note: If GUI is calculating this parameter from the configuration, this parameter may not be present in BSWMD file.

1.4 Number of Virtual Sectors

1.4.1 Virtual Sectors

Parameter defined in Fee_ParamDef.axml	FeeNumberOfVirtualSectors
Description	Defines the number of Virtual Sectors used for FEE.
Generated configuration	FEE_NUMBER_OF_VIRTUAL_SECTORS is set to the defined value.
Default Value	0x2
Parameter Range	Min :0x2 Max : 0x4,For TMS570LS01227/TMS570LS1113. Min : 0x2 Max : 16, For TMS570LS05xx, TMS570LS07xx, TMS570LS09xx.
Parameter Type	uint16
Target File	Fee_cfg.h

1.4.2 Virtual Sectors for EEP1

Parameter defined in Fee_ParamDef.axml	FeeNumberOfVirtualSectorsEEP1
Description	Defines the number of Virtual Sectors used for EEP1.
Generated configuration	FEE_NUMBER_OF_VIRTUAL_SECTORS_EEP1 is set to the defined value.
Default Value	0x0
Parameter Range	Min : 0x0 Max : (FEE_NUMBER_OF_VIRTUAL_SECTORS-0x02)
Parameter Type	uint16
Target File	Fee_cfg.h

Note: FEE_NUMBER_OF_VIRTUAL_SECTORS_EEP1 should be configured as zero if FEE_NUMBER_OF_EEPS = 1.

1.5 FEE functions

1.5.1 FEE_GetVersionInfo

Parameter defined in Fee_ParamDef.axml	FeeVersionInfoApi
Description	Indicates if the user can use the function Fee_GetVersionInfo().
Generated configuration	FEE_VERSION_INFO_API is set to STD_ON if FeeVersionInfoApi is enabled else it is set to STD_OFF.
Default Value	STD_ON
Parameter Range	STD_ON / STD_OFF
Parameter Type	Boolean
Target File	Fee_cfg.h

1.6 FEE Sector Configuration

Note: The structure definition Fee_VirtualSectorConfiguration in fee_cfg.c should be under the conditional compile check
#if (FEE_GENERATE_DEVICEANDVIRTUALSECTORSTRUC == STD_OFF).
Projects need not configure below parameters when above macro is turned ON.

1.6.1 FEE_VirtualSectorConfiguration

Array Name	FEE_VirtualSectorConfiguration	
Description	Used to define a Virtual Sector	
Array Type	Fee_VirtualSectorConfigType. This is a structure having the following members.	
Members	FeeVirtualSectorNumber	Virtual Sector's Number.
	FeeFlashBank	EEPROM emulation is supported only on Bank 7 for F021 devices..
	FeeStartSector	Starting Sector in the Bank for this Virtual Sector.
	FeeEndSector	Ending Sector in the Bank for this Virtual Sector.

The configurations described in the following sections are repeated for each Virtual Sector.

1.6.1.1 FEE_VirtualSectorNumber

Parameter defined in Fee_ParamDef.axml	FeeSectorNumber
Description	Used to assign a number to the Virtual Sector.
Generated configuration	FeeVirtualSectorNumber is set to the value assigned to the symbolic name for the Virtual Sector.
Default Value	1
Parameter Range	Min : 0x1, Max : 0x4,For TMS570LS01227/TMS570LS1113 Min : 0x1 Max : 16, For TMS570LS05xx, TMS570LS07xx, TMS570LS09xx
Parameter Type	uint16

Target File	Fee_Cfg.c

1.6.1.2 FEE_VirtualSectorBank

Parameter defined in Fee_ParamDef.axml	FeeSectorBank
Description	Indicates the Flash Bank used by the Virtual Sector. All the Virtual Sectors should use the same Flash Bank. EEPROM emulation is supported only on Bank 7 for F021 devices.
Generated configuration	FeeFlashBank is set to the value assigned to FeeSectorBank.
Default Value	0x7 for F021 devices.
Parameter Range	Fixed to 0x7 for F021 devices.
Parameter Type	uint16
Target File	Fee_Cfg.c

1.6.1.3 FEE_VirtualSectorStart

Parameter defined in Fee_ParamDef.axml	FeeSectorStart
Description	Indicates the Flash Sector in the Bank used by the Virtual Sector as the Start sector.
Generated configuration	FeeStartSector is set to the value assigned to FeeSectorStart.
Default Value	0x0
Parameter Range	Device specific, can use any Sector of the selected Flash Bank. Please refer to the device datasheet “Flash Memory Map” for more details.
Parameter Type	uint8
Target File	Fee_Cfg.c

1.6.1.4 FEE_VirtualSectorEnd

Parameter defined in Fee_ParamDef.axml	FeeSectorEnd
Description	Indicates the Flash Sector in the Bank used by the Virtual Sector as the End sector.
Generated configuration	FeeEndSector is set to the value assigned to FeeSectorEnd.
Default Value	0x0
Parameter Range	Device specific, can use any Flash Sector of the selected Flash Bank. It should be greater than the FEE Start Sector. Please refer to the device datasheet “Flash Memory Map” for more details.
Parameter Type	uint8
Target File	Fee_Cfg.c

1.6.2 Example Virtual Sector Configuration

```
#if (FEE_GENERATE_DEVICEANDVIRTUALSECTORSTRUC == STD_OFF)
        const Fee_VirtualSectorConfigType Fee_VirtualSectorConfiguration[] =
        {
          /* Virtual Sector 1 */
          {
            1,    /* Virtual sector number */
            7,    /* Bank                  */
            0,    /* Start Sector          */ /*(0, For TMS570LS05xx, TMS570LS07xx, TMS570LS09xx)*/
            0,    /* End Sector            */ /*(3, For TMS570LS05xx, TMS570LS07xx, TMS570LS09xx)*/
          },
          /* Virtual Sector 2 */
          {
            2,    /* Virtual sector number */
            7,    /* Bank                  */
            1,    /* Start Sector          */ /*(4, For TMS570LS05xx, TMS570LS07xx, TMS570LS09xx)*/
            1,    /* End Sector            */ /*(7, For TMS570LS05xx, TMS570LS07xx, TMS570LS09xx)*/
          },
          /* Virtual Sector 3 */
        {
            3,    /* Virtual sector number */
            7,    /* Bank                  */
            2,    /* Start Sector          */ /*(8, For TMS570LS05xx, TMS570LS07xx, TMS570LS09xx)*/
            2,    /* End Sector            */ /*(11, For TMS570LS05xx, TMS570LS07xx, TMS570LS09xx)*/
          },
          /* Virtual Sector 4 */
          {
            4,    /* Virtual sector number */
            7,    /* Bank                  */
            3,    /* Start Sector          */ /*(12, For TMS570LS05xx, TMS570LS07xx, TMS570LS09xx)*/
            3,    /* End Sector            */ /*(15, For TMS570LS05xx, TMS570LS07xx, TMS570LS09xx)*/
          },
        };
#endif
```

1.7 FEE Block Configuration

1.7.1 FEE Block Configuration

Array Name	Fee_BlockConfiguration	
Description	Used to define a block	
Array Type	Fee_BlockConfigType. This is a structure with the following members.	
Members	FeeBlockNumber	Indicates Block's Number.
	FeeBlockSize	Defines Block's Size in bytes.
	FeeImmediateData	Indicates if the block is used for immediate data.
	FeeNumberOfWriteCycles	Number of write cycles required for this block .
	FeeDeviceIndex	Indicates the device index.
	FeeNumberofDatasets	Indicates the number of Datasets for this Block.
	FeeEEPNumber	Indicates the number of EEP.

The configurations described in the following sections are repeated for each Block.

1.7.1.1 FEE_BlockNumber

Parameter defined in Fee_ParamDef.axml	FeeBlockNumber
Description	Assigns a number for the Block.
Generated configuration	FeeBlockNumber is set to a numeric value. It is equal to the BlockNumber.
Default Value	1
Parameter Range	Min : 0x1 Max : 0xFFFE
Parameter Type	uint16
Target File	Fee_Cfg.c

1.7.1.2 FEE_BlockSize

Parameter defined in Fee_ParamDef.axml	FeeBlockSize
Description	Indicates the size of the Block in bytes.
Generated configuration	FeeBlockSize is set to the value assigned to FeeBlockSize.
Default Value	0x008
Parameter Range	0x1 to 0xFFFE
Parameter Type	uint16
Target File	Fee_Cfg.c

1.7.1.3 FEE_NumberOfWriteCycles

Parameter defined in Fee_ParamDef.axml	FeeNumberOfWriteCycles
Description	Indicates the number of clock cycles required to write to a flash address location.
Generated configuration	FeeNumberOfWriteCycles is set to the value assigned to FeeNumberOfWriteCycles.
Default Value	0x1
Parameter Range	Device or core/flash tech dependent parameter.
Parameter Type	uint32
Target File	Fee_Cfg.c

1.7.1.4 FEE_UseImmediateData

Parameter defined in Fee_ParamDef.axml	FeeImmediateData
Description	Indicates if the block is used for immediate data.
Generated configuration	FeeImmediateData is set to the value assigned to FeeImmediateData.
Default Value	FALSE
Parameter Range	TRUE / FALSE
Parameter Type	Boolean
Target File	Fee_Cfg.c

1.7.1.5 FEE Device Index

Parameter defined in Fee_ParamDef.axml	FeeDeviceIndex
Description	Indicates the device index. This will always be 0.
Generated configuration	FeeDeviceIndex is set to the value 0x0.
Default Value	0x0
Parameter Range	Fixed to 0x0.
Parameter Type	uint8
Target File	Fee_Cfg.c

1.7.1.6 FeeNumberOfDataSets

Parameter defined in Fee_ParamDef.axml	FeeDataset
Description	Indicates the number of Datasets for this particular Block .
Generated configuration	FeeNumberOfDataSets is set to the value assigned to FeeDataset.
Default Value	0x01
Parameter Range	0x1 to 0xFF
Parameter Type	uint8
Target File	Fee_Cfg.c

1.7.1.7 FEE EEP Number

Parameter defined in Fee_ParamDef.axml	FeeEEPNumber
Description	Number indicating into which EEP does the block go. 0 -- Block will be configured on EEP1. 1 -- Block will be configured on EEP2.
Generated configuration	FeeEEPNumber is set to the value assigned.
Default Value	0x0
Parameter Range	0x00/0x01
Parameter Type	uint8
Target File	Fee_Cfg.c

1.7.2 Example Block Configuration

```
/* Block Configuration */
const Fee_BlockConfigType Fee_BlockConfiguration[] =
{
   /* Block 1 */
   {
      0x01,          /* Block number                    */
      0x0010,        /* Block size                      */
      TRUE,          /* Block immediate data used       */
      0x00000064,    /* Block number of write cycles    */
      0,             /* Device Index                    */
      1,             /* Number of DataSets              */
      0              /* EEP number                      */
   },
   /* Block 2 */
   {
      0x02,          /* Block number                    */
      0x000B,        /* Block size                      */
      TRUE,          /* Block immediate data used       */
      0x00000064,    /* Block number of write cycles    */
      0,             /* Device Index                    */
      1,             /* Number of DataSets              */
      1              /* EEP number                      */
   },
   /* Block 3 */
   {
      0x03,          /* Block number                    */
      0x000B,        /* Block size                      */
      TRUE,          /* Block immediate data used       */
      0x00000064,    /* Block number of write cycles    */
      0,             /* Device Index                    */
      1,             /* Number of DataSets              */
      1              /* EEP number                      */
   },

   /* Block 4 */
   {
      0x04,          /* Block number                    */
      0x000B,        /* Block size                      */
      TRUE,          /* Block immediate data used       */
      0x00000064,    /* Block number of write cycles    */
```

```
        0,              /* Device Index                 */
        1,              /* Number of DataSets           */
        0               /* EEP number                   */
    },
};
```

1.8 Header Files

The following header files are included in the files generated by the Code Generation Tool.

1.8.1 Header Files for Fee_cfg.h

The following files are included in Fee_cfg.h

1. MemIf_Types.h

1.8.2 Header Files for Fee_cfg.c

The following files are included in Fee_cfg.c

1. Fee.h
2. Fee_Cbk.h
3. SchM_Fee.h

This file should also include version check as following:

```
#if (FEE_AR_MAJOR_VERSION != 0x03)
  #error Fee_Cfg.c: FEE_AR_MAJOR_VERSION of Fee.h is incompatible.
#endif /* FEE_AR_MAJOR_VERSION */
#if (FEE_AR_MINOR_VERSION != 0x00)
  #error Fee_Cfg.c: FEE_AR_MINOR_VERSION of Fee.h is incompatible.
#endif /* FEE_AR_MINOR_VERSION */
#if (FEE_AR_PATCH_VERSION != 0x01)
  #error Fee_Cfg.c: FEE_AR_PATCH_VERSION of Fee.h is incompatible.
#endif /* FEE_AR_PATCH_VERSION */
#if (FEE_SW_MAJOR_VERSION != 1)
  #error Fee_Cfg.c: FEE_SW_MAJOR_VERSION of Fee.h is incompatible.
#endif /* FEE_SW_MAJOR_VERSION */
#if (FEE_SW_MINOR_VERSION != 20)
  #error Fee_Cfg.c: FEE_SW_MINOR_VERSION of Fee.h is incompatible.
#endif /* FEE_SW_MINOR_VERSION */
#if (FEE_SW_PATCH_VERSION != 0)
  #error Fee_Cfg.c: FEE_SW_PATCH_VERSION of Fee.h is incompatible.
#endif /* FEE_SW_PATCH_VERSION */
```