



Elektrobit

# MCAL Wrapper (McalExt) documentation

for TRICORE TC39XX

product release 8.8.7



Elektrobit Automotive GmbH  
Am Wolfsmantel 46  
91058 Erlangen, Germany  
Phone: +49 9131 7701 0  
Fax: +49 9131 7701 6333  
Email: [info.automotive@elektrobit.com](mailto:info.automotive@elektrobit.com)

## Technical support

<https://www.elektrobit.com/support>

## Legal disclaimer

Confidential information.

ALL RIGHTS RESERVED. No part of this publication may be copied in any form, by photocopy, microfilm, retrieval system, or by any other means now known or hereafter invented without the prior written permission of Elektrobit Automotive GmbH.

All brand names, trademarks, and registered trademarks are property of their rightful owners and are used only for description.

Copyright 2023, Elektrobit Automotive GmbH.

# Table of Contents

1. Overview of MCAL Integration release notes .....	4
1.1. Location of MCAL documentation .....	4
2. Supported toolchain .....	5
2.1. Toolchain options .....	5
3. Scope of this release .....	6
3.1. Platforms Module .....	6
3.2. Third-party MCAL version .....	6
3.3. Third-party MCAL modules .....	6
3.4. Third-party MCAL AUTOSAR APIs tested by Elektrobit Automotive GmbH .....	8
3.5. Third-party MCAL CDD APIs tested by Elektrobit Automotive GmbH .....	19
3.6. Third-party MCAL patches by Elektrobit Automotive GmbH .....	24
3.6.1. Use of original or patched version in third-party MCAL modules .....	24
4. Overview of McalExt documentation .....	26
4.1. Background information .....	26
5. Using the McalExt module .....	27
5.1. Add McalExt and MCAL modules to project .....	27
5.2. EB build environment .....	28
5.3. User build environment .....	28
6. McalExt module description .....	29
6.1. Vendor delivery package .....	29
6.2. Connection with vendor delivery package .....	29
6.2.1. <code>plugin.xml</code> connection description .....	29
6.2.2. Makefile connection description .....	30
6.2.3. <code>anchors.xml</code> connection description .....	31
6.3. McalExt file description .....	32
6.4. Use Spi template for asynchronous mode .....	34
6.5. MCAL version update .....	35

# 1. Overview of MCAL Integration release notes

Welcome to the MCAL release notes. These release notes are target-specific and derivative-specific.

[Chapter 2, “Supported toolchain”](#) provides information about the supported toolchain.

[Chapter 3, “Scope of this release”](#) provides specific information about the hardware-dependent third-party modules contained in this EB tresos AutoCore release:

- ▶ AUTOSAR version and revision of your hardware-dependent modules
- ▶ SWS version and revision
- ▶ Module version
- ▶ Supplier of your hardware-dependent modules

## 1.1. Location of MCAL documentation

Depending on the platform release that you purchased, some of the modules may be supplied to Elektrobit Automotive GmbH by third-parties. All MCAL modules are documented outside of these release notes. This documentation contains additional information about the third-party MCAL modules and the patches that were made by Elektrobit Automotive GmbH.

You can find the MCAL module documentation in the following locations:

- ▶ `$TRESOS_BASE/doc/5.0_MCAL_modules/MCAL_Wrapper_documentation.pdf`
- ▶ `$TRESOS_BASE/plugins/<McalExt plugin>/doc`
- ▶ In the online help of EB tresos Studio.

For information about the online help in EB tresos Studio, see the EB tresos Studio user documentation.

## 2. Supported toolchain

This release of EB tresos AutoCore supports Tasking C Compiler v6.3r1p2

### 2.1. Toolchain options

The toolchain options summarize under which conditions this release needs to be built. The release is tested using these toolchain options. If you change the compiler options, consider this release *untested*.

Compiler	Options
	<code>--integer-enumeration --fp-model=1 --tradeoff=2 --switch=auto --global-type-checking --core=tc1.6.2 -O2 -AGKpvX --iso=99 -- default-near-size=0 --eabi=BCFHNSW</code>
Assembler	Options
	<code>--core=tc1.6.2 --list-format=1 --optimize=gs</code>
Linker	Options
	<code>--user-provided-initialization-code --global-type-checking -- core=mpe:vtc -OcLtXY</code>

## 3. Scope of this release

### 3.1. Platforms Module

This release of the `Platforms` module contains the mandatory and derivative-specific implementation part of the `Base` module.

This `Platforms` module shall be used only for TC39XX derivatives.

This module is tested only on hardware with the same sub-derivative as the third-party MCAL version. Other sub-derivatives are not tested.

### 3.2. Third-party MCAL version

This release contains the MCAL release

- ▶ MC-ISAR\_AS440\_TC3xx\_BASIC\_2.10.0
- ▶ MC-ISAR\_AS440\_TC3xx\_COM-E\_2.10.0
- ▶ MC-ISAR\_AS440\_TC3xx\_CD\_2.10.0
- ▶ MC-ISAR\_AS440\_TC3xx\_Demo\_2.10.0

from Infineon.

This release of EB tresos AutoCore is tested only with sub-derivative TC397.

### 3.3. Third-party MCAL modules

This release includes the hardware-dependent third-party MCAL modules listed in the table below.

Module name	AUTOSAR version and revision	SWS version and revision	Module version	Supplier
Adc	4.4.0	4.4.0	20.10.0	Infineon
Bfx	4.4.0	4.4.0	20.0.1	Infineon

Module name	AUTOSAR version and revision	SWS version and revision	Module version	Supplier
Can_17_McmCan	4.4.0	4.4.0	20.10.0	Infineon
CanTrcv_17_V9251	4.4.0	4.4.0	20.10.0	Infineon
CanTrcv_17_W9255	4.4.0	4.4.0	20.10.0	Infineon
Crc	4.4.0	4.4.0	20.0.2	Infineon
Dio	4.4.0	4.4.0	20.0.1	Infineon
Dma	4.4.0	4.4.0	20.10.0	Infineon
Dsadc	4.4.0	4.4.0	20.10.0	Infineon
Eth_17_GEthMacV2	4.4.0	4.4.0	20.0.2	Infineon
Fee	4.4.0	4.4.0	20.10.0	Infineon
Fls_17_Dmu	4.4.0	4.4.0	20.10.0	Infineon
FlsLoader	4.4.0	4.4.0	20.0.0	Infineon
Fr_17_Eray	4.4.0	4.4.0	20.10.0	Infineon
Gpt	4.4.0	4.4.0	20.10.0	Infineon
Hssl	4.4.0	4.4.0	20.0.1	Infineon
I2c	4.4.0	4.4.0	20.10.0	Infineon
Icu_17_TimerIp	4.4.0	4.4.0	20.10.0	Infineon
Iom	4.4.0	4.4.0	20.0.0	Infineon
Lin_17_AscLin	4.4.0	4.4.0	20.10.0	Infineon
McalLib	4.4.0	4.4.0	20.10.0	Infineon
Mcu	4.4.0	4.4.0	20.10.0	Infineon
Ocu	4.4.0	4.4.0	20.10.0	Infineon
Port	4.4.0	4.4.0	20.10.0	Infineon
Pwm_17_GtmCcu6	4.4.0	4.4.0	20.10.0	Infineon
Sent	4.4.0	4.4.0	20.10.0	Infineon
Smu	4.4.0	4.4.0	20.0.2	Infineon
Spi	4.4.0	4.4.0	20.10.0	Infineon
Stm	4.4.0	4.4.0	20.0.0	Infineon
Uart	4.4.0	4.4.0	20.10.0	Infineon
Wdg_17_Scu	4.4.0	4.4.0	20.10.0	Infineon
ResourceM	4.4.0	4.4.0	20.10.0	Infineon

Table 3.1. Third-party hardware-dependent modules

## 3.4. Third-party MCAL AUTOSAR APIs tested by Elektrobit Automotive GmbH

The following third-party MCAL AUTOSAR APIs are verified against the AUTOSAR standard. The scope of Elektrobit Automotive GmbH testing of the MCAL AUTOSAR APIs is limited to generation and compilation.

Vendor AUTOSAR MCAL Module	API Name
Adc	<ul style="list-style-type: none"> <li>▶ Adc_Init</li> <li>▶ Adc_SetupResultBuffer</li> <li>▶ Adc_DeInit</li> <li>▶ Adc_StartGroupConversion</li> <li>▶ Adc_StopGroupConversion</li> <li>▶ Adc_ReadGroup</li> <li>▶ Adc_EnableHardwareTrigger</li> <li>▶ Adc_DisableHardwareTrigger</li> <li>▶ Adc_EnableGroupNotification</li> <li>▶ Adc_DisableGroupNotification</li> <li>▶ Adc_GetGroupStatus</li> <li>▶ Adc_GetStreamLastPointer</li> <li>▶ Adc_GetVersionInfo</li> <li>▶ Adc_SetPowerState</li> <li>▶ Adc_GetCurrentPowerState</li> <li>▶ Adc_GetTargetPowerState</li> <li>▶ Adc_PreparePowerState</li> <li>▶ Adc_Main_PowerTransitionManager</li> </ul>
Bfx	<ul style="list-style-type: none"> <li>▶ Bfx_SetBit_u8u8</li> <li>▶ Bfx_SetBit_u16u8</li> <li>▶ Bfx_SetBit_u32u8</li> <li>▶ Bfx_SetBit_u64u8</li> <li>▶ Bfx_ClrBit_u8u8</li> <li>▶ Bfx_ClrBit_u16u8</li> <li>▶ Bfx_ClrBit_u32u8</li> </ul>



Vendor AUTOSAR MCAL Module	API Name
	<ul style="list-style-type: none"> <li>▶ Bfx_ClrBit_u64u8</li> <li>▶ Bfx_GetBit_u8u8_u8</li> <li>▶ Bfx_GetBit_u16u8_u8</li> <li>▶ Bfx_GetBit_u32u8_u8</li> <li>▶ Bfx_GetBit_u64u8_u8</li> <li>▶ Bfx_SetBits_u8u8u8u8</li> <li>▶ Bfx_SetBits_u16u8u8u8</li> <li>▶ Bfx_SetBits_u32u8u8u8</li> <li>▶ Bfx_SetBits_u64u8u8u8</li> <li>▶ Bfx_GetBits_u8u8u8_u8</li> <li>▶ Bfx_GetBits_u16u8u8_u16</li> <li>▶ Bfx_GetBits_u32u8u8_u32</li> <li>▶ Bfx_GetBits_u64u8u8_u64</li> <li>▶ Bfx_SetBitMask_u8u8</li> <li>▶ Bfx_SetBitMask_u16u16</li> <li>▶ Bfx_SetBitMask_u32u32</li> <li>▶ Bfx_SetBitMask_u64u64</li> <li>▶ Bfx_ClrBitMask_u8u8</li> <li>▶ Bfx_ClrBitMask_u16u16</li> <li>▶ Bfx_ClrBitMask_u32u32</li> <li>▶ Bfx_ClrBitMask_u64u64</li> <li>▶ Bfx_TstBitMask_u8u8_u8</li> <li>▶ Bfx_TstBitMask_u16u16_u8</li> <li>▶ Bfx_TstBitMask_u32u32_u8</li> <li>▶ Bfx_TstBitMask_u64u64_u8</li> <li>▶ Bfx_TstBitLnMask_u8u8_u8</li> <li>▶ Bfx_TstBitLnMask_u16u16_u8</li> <li>▶ Bfx_TstBitLnMask_u32u32_u8</li> <li>▶ Bfx_TstBitLnMask_u64u64_u8</li> <li>▶ Bfx_TstParityEven_u8_u8</li> </ul>

Vendor AUTOSAR MCAL Module	API Name
	<ul style="list-style-type: none"> <li>▶ Bfx_TstParityEven_u16_u8</li> <li>▶ Bfx_TstParityEven_u32_u8</li> <li>▶ Bfx_TstParityEven_u64_u8</li> <li>▶ Bfx_ToggleBits_u8</li> <li>▶ Bfx_ToggleBits_u16</li> <li>▶ Bfx_ToggleBits_u32</li> <li>▶ Bfx_ToggleBits_u64</li> <li>▶ Bfx_ToggleBitMask_u8u8</li> <li>▶ Bfx_ToggleBitMask_u16u16</li> <li>▶ Bfx_ToggleBitMask_u32u32</li> <li>▶ Bfx_ToggleBitMask_u64u64</li> <li>▶ Bfx_ShiftBitRt_u8u8</li> <li>▶ Bfx_ShiftBitRt_u16u8</li> <li>▶ Bfx_ShiftBitRt_u32u8</li> <li>▶ Bfx_ShiftBitRt_u64u8</li> <li>▶ Bfx_ShiftBitLt_u8u8</li> <li>▶ Bfx_ShiftBitLt_u16u8</li> <li>▶ Bfx_ShiftBitLt_u32u8</li> <li>▶ Bfx_ShiftBitLt_u64u8</li> <li>▶ Bfx_RotBitRt_u8u8</li> <li>▶ Bfx_RotBitRt_u16u8</li> <li>▶ Bfx_RotBitRt_u32u8</li> <li>▶ Bfx_RotBitRt_u64u8</li> <li>▶ Bfx_RotBitLt_u8u8</li> <li>▶ Bfx_RotBitLt_u16u8</li> <li>▶ Bfx_RotBitLt_u32u8</li> <li>▶ Bfx_RotBitLt_u64u8</li> <li>▶ Bfx_CopyBit_u8u8u8u8</li> <li>▶ Bfx_CopyBit_u16u8u16u8</li> <li>▶ Bfx_CopyBit_u32u8u32u8</li> </ul>

Vendor AUTOSAR MCAL Module	API Name
	<ul style="list-style-type: none"> <li>▶ Bfx_CopyBit_u64u8u64u8</li> <li>▶ Bfx_PutBits_u8u8u8u8</li> <li>▶ Bfx_PutBits_u16u8u8u16</li> <li>▶ Bfx_PutBits_u32u8u8u32</li> <li>▶ Bfx_PutBits_u64u8u8u64</li> <li>▶ Bfx_PutBitsMask_u8u8u8</li> <li>▶ Bfx_PutBitsMask_u16u16u16</li> <li>▶ Bfx_PutBitsMask_u32u32u32</li> <li>▶ Bfx_PutBitsMask_u64u64u64</li> <li>▶ Bfx_PutBit_u8u8u8</li> <li>▶ Bfx_PutBit_u16u8u8</li> <li>▶ Bfx_PutBit_u32u8u8</li> <li>▶ Bfx_PutBit_u64u8u8</li> <li>▶ Bfx_GetVersionInfo</li> </ul>
Can_17_McmCan	<ul style="list-style-type: none"> <li>▶ Can_17_McmCan_Init</li> <li>▶ Can_17_McmCan_GetVersionInfo</li> <li>▶ Can_17_McmCan_DeInit</li> <li>▶ Can_17_McmCan_SetBaudrate</li> <li>▶ Can_17_McmCan_SetControllerMode</li> <li>▶ Can_17_McmCan_DisableControllerInterrupts</li> <li>▶ Can_17_McmCan_EnableControllerInterrupts</li> <li>▶ Can_17_McmCan_CheckWakeup</li> <li>▶ Can_17_McmCan_GetControllerErrorState</li> <li>▶ Can_17_McmCan_GetControllerMode</li> <li>▶ Can_17_McmCan_GetControllerRxErrorCounter</li> <li>▶ Can_17_McmCan_GetControllerTxErrorCounter</li> <li>▶ Can_17_McmCan_Write</li> <li>▶ Can_17_McmCan_MainFunction_Write</li> <li>▶ Can_17_McmCan_MainFunction_Read</li> <li>▶ Can_17_McmCan_MainFunction_BusOff</li> </ul>

Vendor AUTOSAR MCAL Module	API Name
	<ul style="list-style-type: none"> <li>▶ Can_17_McmCan_MainFunction_Wakeup</li> <li>▶ Can_17_McmCan_MainFunction_Mode</li> </ul>
Crc	<ul style="list-style-type: none"> <li>▶ Crc_CalculateCRC8</li> <li>▶ Crc_CalculateCRC8H2F</li> <li>▶ Crc_CalculateCRC16</li> <li>▶ Crc_CalculateCRC16ARC</li> <li>▶ Crc_CalculateCRC32</li> <li>▶ Crc_CalculateCRC32P4</li> <li>▶ Crc_CalculateCRC64</li> <li>▶ Crc_GetVersionInfo</li> </ul>
Dio	<ul style="list-style-type: none"> <li>▶ Dio_ReadChannel</li> <li>▶ Dio_WriteChannel</li> <li>▶ Dio_ReadPort</li> <li>▶ Dio_WritePort</li> <li>▶ Dio_ReadChannelGroup</li> <li>▶ Dio_WriteChannelGroup</li> <li>▶ Dio_GetVersionInfo</li> <li>▶ Dio_FlipChannel</li> </ul>
Eth_17_GEthMacV2	<ul style="list-style-type: none"> <li>▶ Eth_17_GEthMacV2_Init</li> <li>▶ Eth_17_GEthMacV2_SetControllerMode</li> <li>▶ Eth_17_GEthMacV2_GetControllerMode</li> <li>▶ Eth_17_GEthMacV2_GetPhysAddr</li> <li>▶ Eth_17_GEthMacV2_SetPhysAddr</li> <li>▶ Eth_17_GEthMacV2_UpdatePhysAddrFilter</li> <li>▶ Eth_17_GEthMacV2_WriteMii</li> <li>▶ Eth_17_GEthMacV2_ReadMii</li> <li>▶ Eth_17_GEthMacV2_GetCounterValues</li> <li>▶ Eth_17_GEthMacV2_GetRxStats</li> <li>▶ Eth_17_GEthMacV2_GetTxStats</li> <li>▶ Eth_17_GEthMacV2_GetTxErrorCounterValues</li> </ul>

Vendor AUTOSAR MCAL Module	API Name
	<ul style="list-style-type: none"> <li>▶ Eth_17_GEthMacV2_GetCurrentTime</li> <li>▶ Eth_17_GEthMacV2_EnableEgressTimeStamp</li> <li>▶ Eth_17_GEthMacV2_GetEgressTimeStamp</li> <li>▶ Eth_17_GEthMacV2_GetIngressTimeStamp</li> <li>▶ Eth_17_GEthMacV2_ProvideTxBuffer</li> <li>▶ Eth_17_GEthMacV2_Transmit</li> <li>▶ Eth_17_GEthMacV2_Receive</li> <li>▶ Eth_17_GEthMacV2_TxConfirmation</li> <li>▶ Eth_17_GEthMacV2_GetVersionInfo</li> <li>▶ Eth_17_GEthMacV2_MainFunction</li> </ul>
Fee	<ul style="list-style-type: none"> <li>▶ Fee_Init</li> <li>▶ Fee_SetMode</li> <li>▶ Fee_Read</li> <li>▶ Fee_Write</li> <li>▶ Fee_Cancel</li> <li>▶ Fee_GetStatus</li> <li>▶ Fee_GetJobResult</li> <li>▶ Fee_InvalidateBlock</li> <li>▶ Fee_GetVersionInfo</li> <li>▶ Fee_EraseImmediateBlock</li> <li>▶ Fee_JobEndNotification</li> <li>▶ Fee_JobErrorNotification</li> <li>▶ Fee_MainFunction</li> </ul>
Fls_17_Dmu	<ul style="list-style-type: none"> <li>▶ Fls_17_Dmu_Init</li> <li>▶ Fls_17_Dmu_Erase</li> <li>▶ Fls_17_Dmu_Write</li> <li>▶ Fls_17_Dmu_Cancel</li> <li>▶ Fls_17_Dmu_GetStatus</li> <li>▶ Fls_17_Dmu_GetJobResult</li> <li>▶ Fls_17_Dmu_Read</li> </ul>

Vendor AUTOSAR MCAL Module	API Name
	<ul style="list-style-type: none"> <li>▶ Fls_17_Dmu_Compare</li> <li>▶ Fls_17_Dmu_SetMode</li> <li>▶ Fls_17_Dmu_GetVersionInfo</li> <li>▶ Fls_17_Dmu_BlankCheck</li> <li>▶ Fls_17_Dmu_MainFunction</li> </ul>
Fr_17_Eray	<ul style="list-style-type: none"> <li>▶ Fr_17_Eray_Init</li> <li>▶ Fr_17_Eray_ControllerInit</li> <li>▶ Fr_17_Eray_StartCommunication</li> <li>▶ Fr_17_Eray_AllowColdstart</li> <li>▶ Fr_17_Eray_AllSlots</li> <li>▶ Fr_17_Eray_HaltCommunication</li> <li>▶ Fr_17_Eray_AbortCommunication</li> <li>▶ Fr_17_Eray_SendWUP</li> <li>▶ Fr_17_Eray_SetWakeupChannel</li> <li>▶ Fr_17_Eray_GetPOCStatus</li> <li>▶ Fr_17_Eray_TransmitTxLPdu</li> <li>▶ Fr_17_Eray_CancelTxLPdu</li> <li>▶ Fr_17_Eray_ReceiveRxLPdu</li> <li>▶ Fr_17_Eray_CheckTxLPduStatus</li> <li>▶ Fr_17_Eray_PrepareLPdu</li> <li>▶ Fr_17_Eray_ReconfigLPdu</li> <li>▶ Fr_17_Eray_DisableLPdu</li> <li>▶ Fr_17_Eray_GetGlobalTime</li> <li>▶ Fr_17_Eray_GetNmVector</li> <li>▶ Fr_17_Eray_GetNumOfStartupFrames</li> <li>▶ Fr_17_Eray_GetChannelStatus</li> <li>▶ Fr_17_Eray_GetClockCorrection</li> <li>▶ Fr_17_Eray_GetSyncFrameList</li> <li>▶ Fr_17_Eray_GetWakeupRxStatus</li> <li>▶ Fr_17_Eray_SetAbsoluteTimer</li> </ul>

Vendor AUTOSAR MCAL Module	API Name
	<ul style="list-style-type: none"> <li>▶ Fr_17_Eray_CancelAbsoluteTimer</li> <li>▶ Fr_17_Eray_EnableAbsoluteTimerIRQ</li> <li>▶ Fr_17_Eray_AckAbsoluteTimerIRQ</li> <li>▶ Fr_17_Eray_DisableAbsoluteTimerIRQ</li> <li>▶ Fr_17_Eray_GetAbsoluteTimerIRQStatus</li> <li>▶ Fr_17_Eray_GetVersionInfo</li> <li>▶ Fr_17_Eray_ReadCCConfig</li> </ul>
Gpt	<ul style="list-style-type: none"> <li>▶ Gpt_GetVersionInfo</li> <li>▶ Gpt_Init</li> <li>▶ Gpt_DeInit</li> <li>▶ Gpt_GetTimeElapsed</li> <li>▶ Gpt_GetTimeRemaining</li> <li>▶ Gpt_StartTimer</li> <li>▶ Gpt_StopTimer</li> <li>▶ Gpt_EnableNotification</li> <li>▶ Gpt_DisableNotification</li> <li>▶ Gpt_SetMode</li> <li>▶ Gpt_DisableWakeup</li> <li>▶ Gpt_EnableWakeup</li> <li>▶ Gpt_CheckWakeup</li> <li>▶ Gpt_GetPredefTimerValue</li> </ul>
Icu_17_Timerlp	<ul style="list-style-type: none"> <li>▶ Icu_17_Timerlp_Init</li> <li>▶ Icu_17_Timerlp_DeInit</li> <li>▶ Icu_17_Timerlp_SetMode</li> <li>▶ Icu_17_Timerlp_DisableWakeup</li> <li>▶ Icu_17_Timerlp_EnableWakeup</li> <li>▶ Icu_17_Timerlp_CheckWakeup</li> <li>▶ Icu_17_Timerlp_SetActivationCondition</li> <li>▶ Icu_17_Timerlp_DisableNotification</li> <li>▶ Icu_17_Timerlp_EnableNotification</li> </ul>

Vendor AUTOSAR MCAL Module	API Name
	<ul style="list-style-type: none"> <li>▶ Icu_17_TimerIp_GetInputState</li> <li>▶ Icu_17_TimerIp_StartTimestamp</li> <li>▶ Icu_17_TimerIp_StopTimestamp</li> <li>▶ Icu_17_TimerIp_GetTimestampIndex</li> <li>▶ Icu_17_TimerIp_ResetEdgeCount</li> <li>▶ Icu_17_TimerIp_EnableEdgeCount</li> <li>▶ Icu_17_TimerIp_EnableEdgeDetection</li> <li>▶ Icu_17_TimerIp_DisableEdgeDetection</li> <li>▶ Icu_17_TimerIp_DisableEdgeCount</li> <li>▶ Icu_17_TimerIp_GetEdgeNumbers</li> <li>▶ Icu_17_TimerIp_StartSignalMeasurement</li> <li>▶ Icu_17_TimerIp_StopSignalMeasurement</li> <li>▶ Icu_17_TimerIp_GetTimeElapsed</li> <li>▶ Icu_17_TimerIp_GetDutyCycleValues</li> <li>▶ Icu_17_TimerIp_GetVersionInfo</li> </ul>
Lin_17_AscLin	<ul style="list-style-type: none"> <li>▶ Lin_17_AscLin_Init</li> <li>▶ Lin_17_AscLin_CheckWakeup</li> <li>▶ Lin_17_AscLin_GetVersionInfo</li> <li>▶ Lin_17_AscLin_SendFrame</li> <li>▶ Lin_17_AscLin_GoToSleep</li> <li>▶ Lin_17_AscLin_GoToSleepInternal</li> <li>▶ Lin_17_AscLin_Wakeup</li> <li>▶ Lin_17_AscLin_WakeupInternal</li> <li>▶ Lin_17_AscLin_GetStatus</li> </ul>
Mcu	<ul style="list-style-type: none"> <li>▶ Mcu_Init</li> <li>▶ Mcu_InitRamSection</li> <li>▶ Mcu_InitClock</li> <li>▶ Mcu_DistributePllClock</li> <li>▶ Mcu_GetPllStatus</li> <li>▶ Mcu_GetResetReason</li> </ul>



Vendor AUTOSAR MCAL Module	API Name
	<ul style="list-style-type: none"> <li>▶ Mcu_GetResetRawValue</li> <li>▶ Mcu_PerformReset</li> <li>▶ Mcu_SetMode</li> <li>▶ Mcu_GetVersionInfo</li> <li>▶ Mcu_GetRamState</li> </ul>
Ocu	<ul style="list-style-type: none"> <li>▶ Ocu_Init</li> <li>▶ Ocu_DeInit</li> <li>▶ Ocu_StartChannel</li> <li>▶ Ocu_StopChannel</li> <li>▶ Ocu_SetPinState</li> <li>▶ Ocu_SetPinAction</li> <li>▶ Ocu_GetCounter</li> <li>▶ Ocu_SetAbsoluteThreshold</li> <li>▶ Ocu_SetRelativeThreshold</li> <li>▶ Ocu_DisableNotification</li> <li>▶ Ocu_EnableNotification</li> <li>▶ Ocu_GetVersionInfo</li> </ul>
Port	<ul style="list-style-type: none"> <li>▶ Port_Init</li> <li>▶ Port_SetPinDirection</li> <li>▶ Port_RefreshPortDirection</li> <li>▶ Port_GetVersionInfo</li> <li>▶ Port_SetPinMode</li> </ul>
Pwm_17_GtmCcu6	<ul style="list-style-type: none"> <li>▶ Pwm_17_GtmCcu6_Init</li> <li>▶ Pwm_17_GtmCcu6_DeInit</li> <li>▶ Pwm_17_GtmCcu6_SetDutyCycle</li> <li>▶ Pwm_17_GtmCcu6_SetPeriodAndDuty</li> <li>▶ Pwm_17_GtmCcu6_SetOutputToIdle</li> <li>▶ Pwm_17_GtmCcu6_GetOutputState</li> <li>▶ Pwm_17_GtmCcu6_DisableNotification</li> <li>▶ Pwm_17_GtmCcu6_EnableNotification</li> </ul>

Vendor AUTOSAR MCAL Module	API Name
	<ul style="list-style-type: none"> <li>▶ Pwm_17_GtmCcu6_SetPowerState</li> <li>▶ Pwm_17_GtmCcu6_GetCurrentPowerState</li> <li>▶ Pwm_17_GtmCcu6_GetTargetPowerState</li> <li>▶ Pwm_17_GtmCcu6_PreparePowerState</li> <li>▶ Pwm_17_GtmCcu6_GetVersionInfo</li> <li>▶ Pwm_17_GtmCcu6_Main_PowerTransitionManager</li> </ul>
Spi	<ul style="list-style-type: none"> <li>▶ Spi_Init</li> <li>▶ Spi_DeInit</li> <li>▶ Spi_WriteIB</li> <li>▶ Spi_AsyncTransmit</li> <li>▶ Spi_ReadIB</li> <li>▶ Spi_SetupEB</li> <li>▶ Spi_GetStatus</li> <li>▶ Spi_GetJobResult</li> <li>▶ Spi_GetSequenceResult</li> <li>▶ Spi_GetVersionInfo</li> <li>▶ Spi_SyncTransmit</li> <li>▶ Spi_GetHWUnitStatus</li> <li>▶ Spi_Cancel</li> <li>▶ Spi_SetAsyncMode</li> <li>▶ Spi_MainFunction_Handling</li> </ul>
Wdg_17_Scu	<ul style="list-style-type: none"> <li>▶ Wdg_17_Scu_Init</li> <li>▶ Wdg_17_Scu_SetMode</li> <li>▶ Wdg_17_Scu_SetTriggerCondition</li> <li>▶ Wdg_17_Scu_GetVersionInfo</li> </ul>

Table 3.2. Third-party MCAL AUTOSAR APIs tested by Elektrobit Automotive GmbH

## 3.5. Third-party MCAL CDD APIs tested by Elektrobit Automotive GmbH

The following third-party MCAL CDD APIs are verified against the modules implementation. The scope of Elektrobit Automotive GmbH testing of the MCAL CDD APIs is limited to generation and compilation.

Vendor CDD MCAL Module	API Name
Dma	<ul style="list-style-type: none"> <li>▶ Dma_DeInit</li> <li>▶ Dma_Init</li> <li>▶ Dma_IsInitDone</li> <li>▶ Dma_ChInit</li> <li>▶ Dma_ChUpdate</li> <li>▶ Dma_ChDeInit</li> <li>▶ Dma_ChTransferFreeze</li> <li>▶ Dma_ChTransferResume</li> <li>▶ Dma_ChEnableHardwareTrigger</li> <li>▶ Dma_ChDisableHardwareTrigger</li> <li>▶ Dma_ChStartTransfer</li> <li>▶ Dma_ChStopTransfer</li> <li>▶ Dma_ChGetRemainingData</li> <li>▶ Dma_ChSwitchBuffer</li> <li>▶ Dma_GetEvents</li> <li>▶ Dma_ChStatusClear</li> <li>▶ Dma_ChInterruptEnable</li> <li>▶ Dma_ChInterruptDisable</li> <li>▶ Dma_GetVersionInfo</li> <li>▶ Dma_MEStatusClear</li> <li>▶ Dma_InitCheck</li> <li>▶ Dma_GetCrcValue</li> <li>▶ Dma_GetCurrentTimeStamp</li> <li>▶ Dma_IsChannelInitDone</li> <li>▶ Dma_SetPattern</li> </ul>

Vendor CDD MCAL Module	API Name
Dsadc	<ul style="list-style-type: none"> <li>▶ Dsadc_Init</li> <li>▶ Dsadc_DeInit</li> <li>▶ Dsadc_StartModulation</li> <li>▶ Dsadc_StopModulation</li> <li>▶ Dsadc_ReadStreamResults</li> <li>▶ Dsadc_ReadResult</li> <li>▶ Dsadc_GetStatus</li> <li>▶ Dsadc_SetupResultBuffer</li> <li>▶ Dsadc_StartCarrierSignal</li> <li>▶ Dsadc_StopCarrierSignal</li> <li>▶ Dsadc_EnableNotifications</li> <li>▶ Dsadc_DisableNotifications</li> <li>▶ Dsadc_GetTimestamp</li> <li>▶ Dsadc_StartCalibration</li> <li>▶ Dsadc_GetCalibrationStatus</li> <li>▶ Dsadc_InitCheck</li> <li>▶ Dsadc_GetVersionInfo</li> <li>▶ Dsadc_RestartDemodulator</li> <li>▶ Dsadc_GetIrmsValue</li> <li>▶ Dsadc_SetGainCorrRegValue</li> <li>▶ Dsadc_RestartIntegrator</li> <li>▶ Dsadc_GetSdcapValue</li> </ul>
FIsLoader	<ul style="list-style-type: none"> <li>▶ FIsLoader_DeInit</li> <li>▶ FIsLoader_Erase</li> <li>▶ FIsLoader_GetVersionInfo</li> <li>▶ FIsLoader_Init</li> <li>▶ FIsLoader_Lock</li> <li>▶ FIsLoader_UnLock</li> <li>▶ FIsLoader_Write</li> </ul>
Hssl	<ul style="list-style-type: none"> <li>▶ Hssl_Init</li> </ul>

Vendor CDD MCAL Module	API Name
	<ul style="list-style-type: none"> <li>▶ Hssl_InitChannel</li> <li>▶ Hssl_SetMode</li> <li>▶ Hssl_Reset</li> <li>▶ Hssl_Write</li> <li>▶ Hssl_WriteAck</li> <li>▶ Hssl_Read</li> <li>▶ Hssl_ReadRply</li> <li>▶ Hssl_Id</li> <li>▶ Hssl_Trigger</li> <li>▶ Hssl_StartStream</li> <li>▶ Hssl_StopStream</li> <li>▶ Hssl_MultiWrite</li> <li>▶ Hssl_MultiRead</li> <li>▶ Hssl_ActivateSlave</li> <li>▶ Hssl_DeactivateSlave</li> <li>▶ Hssl_SelectSlave</li> <li>▶ Hssl_GetGlobalError</li> <li>▶ Hssl_GetChannelError</li> <li>▶ Hssl_GetVersionInfo</li> <li>▶ Hssl_DmaCallout</li> <li>▶ Hssl_DmaErrCallout</li> </ul>
I2c	<ul style="list-style-type: none"> <li>▶ I2c_Init</li> <li>▶ I2c_DeInit</li> <li>▶ I2c_GetStatus</li> <li>▶ I2c_SyncWrite</li> <li>▶ I2c_SyncRead</li> <li>▶ I2c_AsyncWrite</li> <li>▶ I2c_AsyncRead</li> <li>▶ I2c_CancelOperation</li> <li>▶ I2c_GetVersionInfo</li> </ul>

Vendor CDD MCAL Module	API Name
Iom	<ul style="list-style-type: none"> <li>▶ Iom_Init</li> <li>▶ Iom_DeInit</li> <li>▶ Iom_ResetKernel</li> <li>▶ Iom_GetResetStatus</li> <li>▶ Iom_ClrResetStatus</li> <li>▶ Iom_ClrFpcEdgeStatus</li> <li>▶ Iom_GetFpcEdgeStatus</li> <li>▶ Iom_SetFpcCompare</li> <li>▶ Iom_GetFpcCompare</li> <li>▶ Iom_SetLamConfig</li> <li>▶ Iom_GetLamConfig</li> <li>▶ Iom_SetLamThreshold</li> <li>▶ Iom_GetLamThreshold</li> <li>▶ Iom_GetLamEntWinCount</li> <li>▶ Iom_SetEcmGlobalEveSel</li> <li>▶ Iom_GetEcmGlobalEveSel</li> <li>▶ Iom_SetEcmThresVal</li> <li>▶ Iom_GetEcmThresVal</li> <li>▶ Iom_GetEcmEveTrigHis</li> <li>▶ Iom_ClrEcmStatusHistory</li> <li>▶ Iom_GetVersionInfo</li> </ul>
Sent	<ul style="list-style-type: none"> <li>▶ Sent_Init</li> <li>▶ Sent_SetChannel</li> <li>▶ Sent_ReadData</li> <li>▶ Sent_ReadSerialData</li> <li>▶ Sent_ReadChannelStatus</li> <li>▶ Sent_SpcGenPulse</li> <li>▶ Sent_SetWdgTimer</li> <li>▶ Sent_GetVersionInfo</li> <li>▶ Sent_DeInit</li> </ul>

Vendor CDD MCAL Module	API Name
	<ul style="list-style-type: none"> <li>▶ Sent_ReadGlitchFilterStatus</li> <li>▶ Sent_ResetGlitchFilterStatus</li> <li>▶ Sent_FDFLParameters</li> </ul>
Smu	<ul style="list-style-type: none"> <li>▶ Smu_Init</li> <li>▶ Smu_DeInit</li> <li>▶ Smu_GetAlarmAction</li> <li>▶ Smu_SetAlarmAction</li> <li>▶ Smu_ClearAlarmStatus</li> <li>▶ Smu_GetAlarmStatus</li> <li>▶ Smu_SetAlarmStatus</li> <li>▶ Smu_GetAlarmDebugStatus</li> <li>▶ Smu_LockConfigRegs</li> <li>▶ Smu_ReleaseFSP</li> <li>▶ Smu_ActivateFSP</li> <li>▶ Smu_SetupErrorPin</li> <li>▶ Smu_ReleaseErrorPin</li> <li>▶ Smu_RTStop</li> <li>▶ Smu_GetRTMissedEvent</li> <li>▶ Smu_ActivatePES</li> <li>▶ Smu_RegisterMonitor</li> <li>▶ Smu_GetSmuState</li> <li>▶ Smu_ActivateRunState</li> <li>▶ Smu_GetVersionInfo</li> <li>▶ Smu_CoreAliveTest</li> <li>▶ Smu_InitCheck</li> <li>▶ Smu_GetAlarmExecutionStatus</li> <li>▶ Smu_ClearAlarmExecutionStatus</li> </ul>
Stm	<ul style="list-style-type: none"> <li>▶ Stm_EnableModule</li> <li>▶ Stm_EnableAlarm</li> <li>▶ Stm_DisableAlarm</li> </ul>

Vendor CDD MCAL Module	API Name
	<ul style="list-style-type: none"><li>▶ Stm_SetCompareMatchControl</li><li>▶ Stm_ReadTimerValue</li><li>▶ Stm_ReadTotalTimerValue</li><li>▶ Stm_SleepModeHandle</li><li>▶ Stm_GetVersionInfo</li></ul>
Uart	<ul style="list-style-type: none"><li>▶ Uart_InitCheck</li><li>▶ Uart_Init</li><li>▶ Uart_Read</li><li>▶ Uart_Write</li><li>▶ Uart_AbortRead</li><li>▶ Uart_AbortWrite</li><li>▶ Uart_GetStatus</li><li>▶ Uart_DeInit</li><li>▶ Uart_GetVersionInfo</li><li>▶ Uart_MainFunction_Read</li><li>▶ Uart_MainFunction_Write</li></ul>

Table 3.3. Third-party MCAL CDD APIs verified by Elektrobit Automotive GmbH

## 3.6. Third-party MCAL patches by Elektrobit Automotive GmbH

### 3.6.1. Use of original or patched version in third-party MCAL modules

The EB tresos Installer is the installation tool for the third-party MCAL modules. For more information on the EB tresos Installer, see 1.1\_EB\_tresos\_installation\_guide.pdf. In the EB tresos Installer, you can choose one of the following options:

- ▶ During installation it is possible to disable the update package. Afterwards, only original MCAL modules are part of your installation. For more information, see [Chapter 6, “McalExt module description”](#).



- ▶ If you also install the update package, all changes by Elektrobit Automotive GmbH that are described in [Section 6.3, “McalExt file description”](#) are part of your EB tresos AutoCore installation.

Additionally, with this update package you can switch between the original version and the patched version of each MCAL module.

The update package includes the `perform_MCAL_change.bat` batch file in the `McalExt` module. Use this batch file to switch from one version to another.

In the `perform_MCAL_change.bat`, you need to specify one of the following parameters:

- ▶ `EB_update`: Update the files in the module with content of Elektrobit Automotive GmbH.
- ▶ `origin`: Reset the files in the module with vendor content.

---

#### **WARNING**



#### **Changes due to execution of the batch file affect all projects**

The changes affect all projects that use the changed module. Therefore, execute the batch-file before generating code for your project.

---

## 4. Overview of McalExt documentation

This documentation is target-specific and derivative-specific.

This user guide describes the concepts and the configuration of the module:

► McalExt

### 4.1. Background information

McalExt is a wrapper module that connects the vendor MCAL delivery, EB tresos Studio, and the Elektrobit Automotive GmbH (EB) build environment. It allows you to use the vendor MCAL with EB tresos Studio with as few modifications as possible.

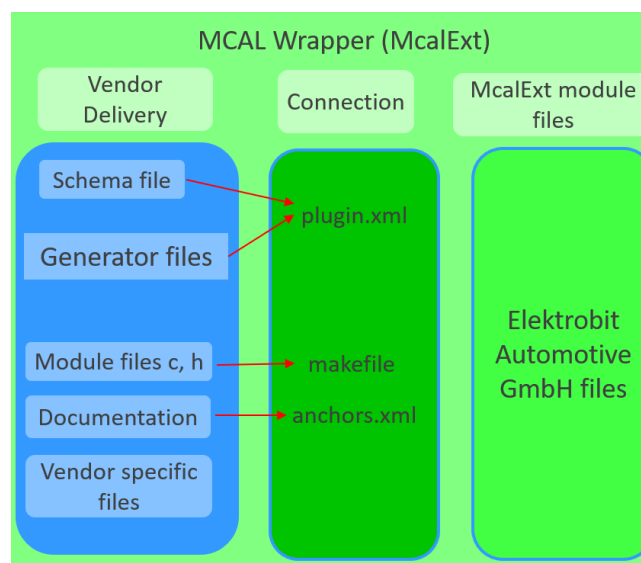


Figure 4.1. MCAL Wrapper (McalExt) overview

The MCAL Wrapper (McalExt) documentation also:

- provide information about the vendor delivery package, see [Section 6.1, “Vendor delivery package”](#)
- provide information about the connection with the vendor delivery package, see [Section 6.2, “Connection with vendor delivery package”](#)
- provide information about Elektrobit Automotive GmbH files, see [Section 6.3, “McalExt file description”](#)

## 5. Using the McalExt module

### 5.1. Add McalExt and MCAL modules to project

- Add the `McalExt` module to the project configuration.

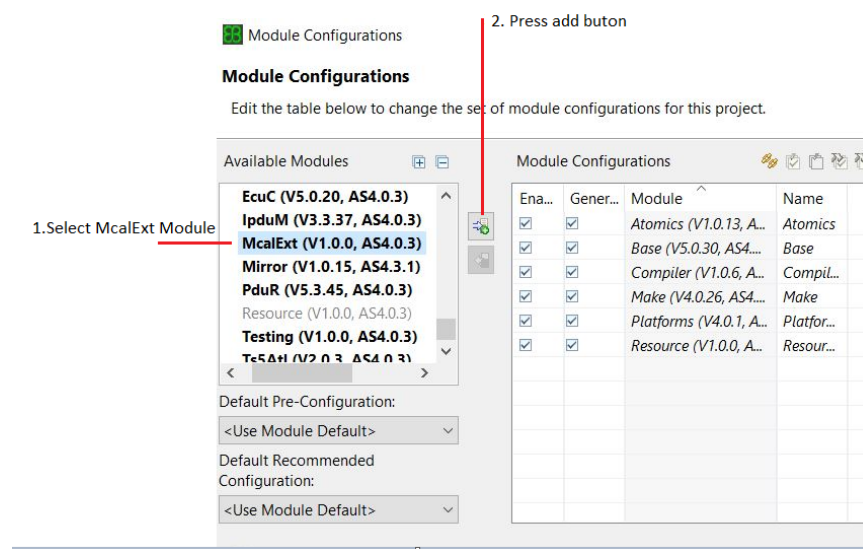


Figure 5.1. Add McalExt Module

- Add the needed MCAL modules to the project configuration

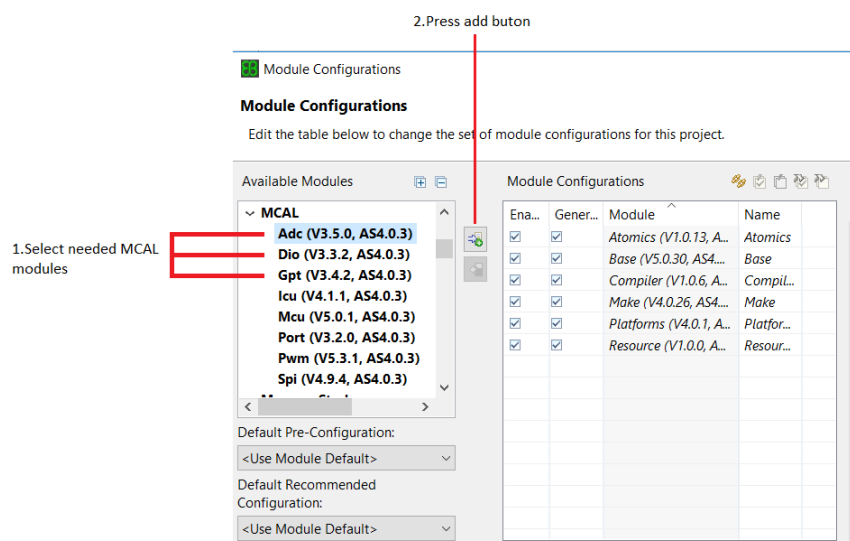


Figure 5.2. Add Mcal Module

## 5.2. EB build environment

In order to use the `McalExt` module in a project that is using the EB build environment, you must do the following:

- ▶ Add the `McalExt` module to the project configuration. see [Figure 5.1, “Add McalExt Module”](#)
- ▶ Add the needed MCAL modules to the project configuration. see [Figure 5.2, “Add Mcal Module”](#)

## 5.3. User build environment

When a different build environment (other than the EB build environment) is used in the project, you must do the following:

- ▶ Add the `McalExt` module to the project configuration. see [Figure 5.1, “Add McalExt Module”](#)
- ▶ Add the needed MCAL modules to the project configuration. see [Figure 5.2, “Add Mcal Module”](#)
- ▶ Add all the files that are configured in the module makefiles and all files configured in the `IncludePaths` and `FilesToBuild` from `McalExt` module configuration, see `IncludePaths` and `FilesToBuild` description in [Section 6.3, “McalExt file description”](#)

## 6. McalExt module description

### 6.1. Vendor delivery package

This represents the package that is delivered by different MCAL vendors (e.g. Infineon, Renesas, NXP, etc.) and contains different MCAL modules. For the MCAL modules and version that are integrated in this release please see [Section 3.3, “Third-party MCAL modules”](#).

### 6.2. Connection with vendor delivery package

The vendor delivery has a structure that cannot be used directly in the EB build environment. The `McalExt` module is introduced to make the connection between the EB build environment and the vendor delivery. The connection is made in the following files of the `McalExt` module:

`plugin.xml` connection, see [Section 6.2.1, “plugin.xml connection description”](#)

`makefile` connection, see [Section 6.2.2, “Makefile connection description”](#)

`anchors.xml` connection, see [Section 6.2.3, “anchors.xml connection description”](#)

#### 6.2.1. plugin.xml connection description

- Schema file, e.g.:

```
<schema>
<manager class="dreisoft.tresos.autosar2.resourcehandling.AutosarSchemaManager"/>
<!-- Define the file(s) from which to load the schemas -->
<resource value="MCAL_Delivery/PathToSchemaFile/ModuleName.xdm" type="xdm"/>
</schema>
```

**PathToSchemaFile** - represents the path to where the module schema file is located in the MCAL vendor delivery.

**ModuleName** - represents the name of the schema file that should be used.

- Code Generator e.g.:

```
<!-- common template path parameters -->
```

```
<parameter name="templates"
mode="generate,verify" value="MCAL_Delivery/PathToGenerator"/>
```

**PathToGenerator** - represents the path where the generator is located in the vendor delivery.

- ▶ Ant code generator, e.g.:

```
<generator moduleId="ModuleId"
class="dreisoft.tresos.generator.ng.api.NGGenerator"
id="ModuleId_UniqueNGGeneratorId"
step="post"> <!-- run after code-generation -->
<parameter name="buildfile" value="MCAL_Delivery/PathToAntGenerator/AntGeneratorFile.xml"/>
</generator>
```

**PathToAntGenerator** - represents the path where the ant generator file is located in the vendor delivery.

**AntGeneratorFile** - represents the name of the ant generator file delivered by the vendor.

---

**NOTE**

This is applicable only if vendor provided the `AntGeneratorFile.xml` file.



## 6.2.2. Makefile connection description

For each MCAL module that is integrated in the `McalExt` a `make` folder exists that contains the makefiles for the respective MCAL module:

- ▶ `Module_defs.mak` file - registers the file(s) that are present in vendor delivery and the files that are generated for this module.

```
McalExt_GEN_FILES += $(McalExt_OUTPUT_PATH)\inc\ModuleName_Cfg.h
McalExt_GEN_FILES += $(McalExt_OUTPUT_PATH)\inc\ModuleName_PBcfg.h
McalExt_GEN_FILES += $(McalExt_OUTPUT_PATH)\src\ModuleName_PBcfg.c
CC_INCLUDE_PATH   += $(McalExt_CORE_PATH)\MCAL_Delivery\PathToHeaderFiles
```

**ModuleName** - represents the name of the header files that are generated.

**PathToHeaderFiles** - represents the path from the vendor delivery where the static header files are located.

- ▶ `Module_rules.mak` file - registers the specific module file(s) that are needed for compilation.

```
Module_src_FILES += $(McalExt_CORE_PATH)\MCAL_Delivery\PathToSourceFile\Module_Name.c
Module_src_FILES += $(McalExt_OUTPUT_PATH)\src\ModuleName_PBcfg.c
```

**ModuleName** - represents the name of the C files that are generated or are present in the vendor delivery.

**PathToSourceFile** - represents the path from the vendor delivery where the C static files are located.

All the MCAL modules makefiles will be included in `McalExt_defs.mak` and `McalExt_rules.mak` files only if the respective module is used in the EB tresos Studio project:

► **McalExt\_defs.mak file**

```
ifeq ($(McalExt_Can_USED),true)
ifeq ($(Can_VARIANT),ModuleNameVariant)
include $(McalExt_CORE_PATH)\make\make_Can\Can_defs.mak
endif
endif
```

► **McalExt\_rules.mak file**

```
ifeq ($(McalExt_Can_USED),true)
ifeq ($(Can_VARIANT),ModuleNameVariant)
LIBRARIES_TO_BUILD += Can_src
include $(McalExt_CORE_PATH)\make\make_Can\Can_rules.mak
endif
endif
```

**ModuleNameVariant** - represents the MCAL module name that is used in the EB tresos Studio project.

### 6.2.3. anchors.xml connection description

► **anchors.xml file registers the MCAL documentation that is shown in the EB tresos Studio help window, e.g:**

```
<topic label="DocName" href="PathToDoc/DocName"/>
<topic label="DocName" href="PathToDoc/DocName"/>
<topic label="DocName" href="PathToDoc/DocName"/>
```

**PathToDoc** - represents the path where the MCAL documents are located.

**DocName** - represents the name of the MCAL documents.

## 6.3. McalExt file description

Some patches made by Elektrobit Automotive GmbH are due to missing or incomplete files in the MCAL vendor delivery. Additionally, some EB tresos Studio features are enabled. These patches are separated from the original installation files.

- **config** - In this folder there is located the `McalExt` wrapper schema file `McalExt.xdm` that allows you to configure the following parameters that can be used in the project:
- **PlatformModuleDefine** – you can configure defines that will be generated in `Platforms_Modules.h` file and that can be used in the project, e.g.: configure an `Mcu` configuration pointer that will be used in the `Mcu_Init()` function.

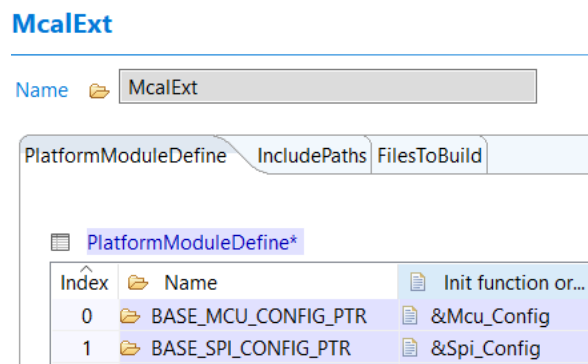


Figure 6.1. Platform Module Define parameter configuration

Generated file content example:

```
#define BASE_MCU_CONFIG_PTR          &Mcu_Config
#define BASE_SPI_CONFIG_PTR          &Spi_Config
```

Usage of defines (`EcuM_DriverInitListOne()`), e.g.:

```
/* *** Call service Init of module Mcu *** */
Mcu_Init(BASE_MCU_CONFIG_PTR);
```

- **IncludePaths** – allows you to configure different paths that need to be included by the build environment. This will be generated in the `McalExtWrapper.mak` file:



## McalExt

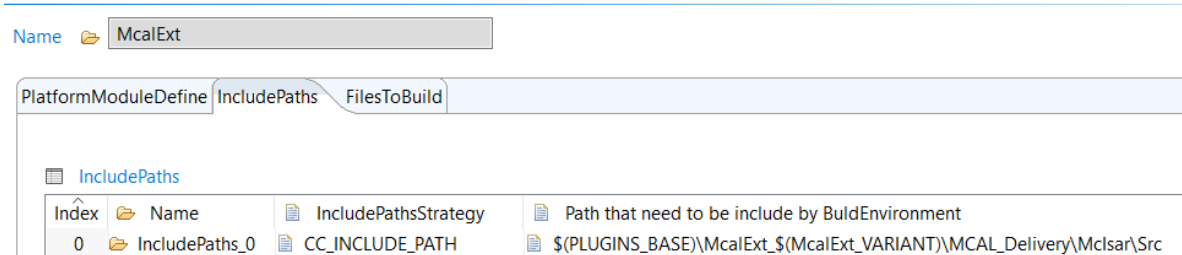


Figure 6.2. Path that needs to be included by BuildEnvironment

Generated content in `McalExtWrapper.mak` file:

```
CC_INCLUDE_PATH += $(PLUGINS_BASE)\McalExt_$(McalExt_VARIANT)\MCAL_Delivery\McIsar\Src
```

- **FilesToBuild** - allows you to configure different files that need to be compiled. This will be generated in the `McalExtWrapper.mak` file:

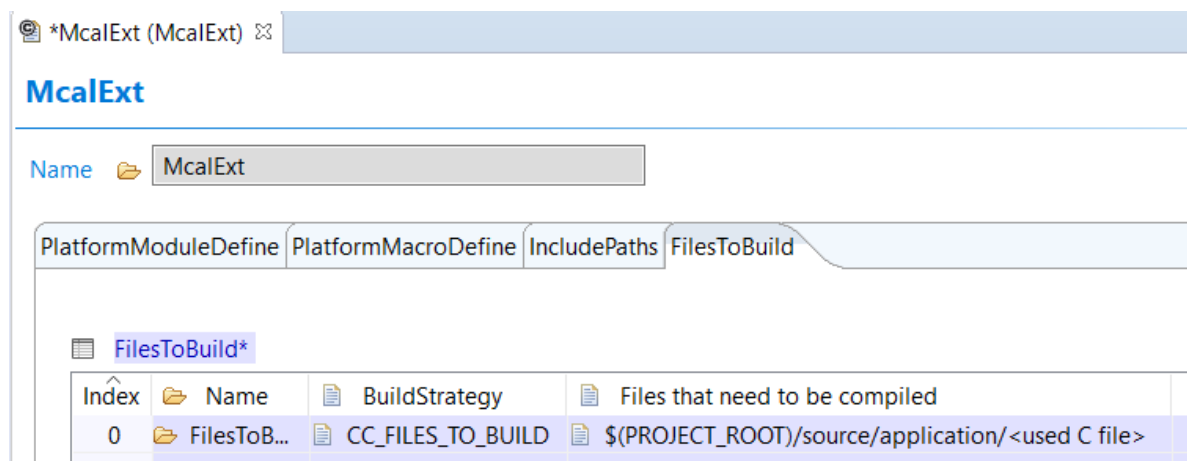


Figure 6.3. Files that need to be compiled

Generated content in `McalExtWrapper.mak` file:

```
CC_FILES_TO_BUILD += $(PROJECT_ROOT)\source\application\Eb_Intgr_BswM_UserCallouts.c
```

- **config\_ext** - In this folder there are present `Mcu` preconfiguration and recommended configurations:
  - Preconfiguration files that contain the parameter configuration value that should not be modified e.g. `McuResetReasonConf`.
  - Recommended configuration – the configuration that was validated by Elektrobit Automotive GmbH while performing IP3/QP2 (if this was ordered), e.g.: `Mcu` recommended configuration that contains

the clock configuration and other `Mcu` related parameter configuration. You can decide if the recommended configuration is used or not.

- ▶ `doc` - In this folder there are added MCAL module documentations if applicable.
- ▶ `generate` - Contains the files that will be generated by `McalExt` wrapper plugin.
- ▶ `include` - Contains the header file(s) that are created/patched by Elektrobit Automotive GmbH.
- ▶ `make` - Contains the makefiles for integrated MCAL module, see [Section 6.2.2, “Makefile connection description”](#)
- ▶ `MCAL_Delivery` - Contains the vendor delivery files.
- ▶ `resources` - Includes several XML-based service needs assistant or properties files that are provided by Elektrobit Automotive GmbH. These files support you to faster complete a valid configuration.
  - ▶ `Dem_Events.xml` – Dem event generation in the EB tresos AutoCore Generic `Dem` module.
  - ▶ EcuM initialization in the EB tresos AutoCore Generic `EcuM` module.
  - ▶ SchM Main function handling in the EB tresos AutoCore Generic `Rte` module.
- ▶ `swcd` - Includes the BSWMD files that are mandatory since AUTOSAR 4.0. Those files are used by BSW modules and EB tresos Studio wizards provided by Elektrobit Automotive GmbH.
  - ▶ Generation of exclusive areas in the EB tresos AutoCore Generic `Rte` module.
  - ▶ Mapping of the Main function in the EB tresos AutoCore Generic `Rte` module.
  - ▶ Generation of MemMap header file(s) in EB tresos AutoCore Generic `MemMap` module.
- ▶ `src` - Contains source file(s) that should be compiled, created by Elektrobit Automotive GmbH.

## 6.4. Use Spi template for asynchronous mode

The SPI uses the service of the DMA for data transmission in asynchronous mode. The Dma Channels used by Spi are configurable and the allocation of the same is done in MCU. Each channel can work in software triggered mode or hardware triggered mode. For SPI to use DMA channel, the channel should be configured as hardware trigger type.

The SPI uses the service of the DMA for data transmission in asynchronous mode. The Dma Channels used by Spi are configurable and the allocation of the same is done in MCU. Each channel can work in software triggered mode or hardware triggered mode. For SPI to use DMA channel, the channel should be configured as hardware trigger type.

In this implementation the Combined Software/Hardware Controlled Mode will be used. Hence the first DMA transfer will be triggered by software when setting the corresponding request channel bit in the Software Transaction Request Register, while the hardware requests are still disabled. The subsequent DMA transfers are triggered by the corresponding hardware request, this happens after the hardware requests have been enabled.

The QSPI Peripheral's RX or TX Interrupt Line is used to trigger the DMA transactions. Each QSPI Peripheral uses two DMA channels to trigger reception and transmission respectively.

The DMA Mechanism is used for the asynchronous transmission mode. It is not used during synchronous transmission mode. The SPI handler driver initializes the DMA channels used.

The DMA channel number used by SPI will be used as the priority number as per HW recommendation.

QSPI RX and TX have to be mapped to DMA - not to interrupts. Infineon IRQ module is capable to map interrupt nodes to CPU0/CPU1/CPU2 interrupts or to DMA. This functionality have to be implemented in the user application as part of integration code with EB's Autocore OS or Safety OS and with AutoCore Generic.

Please copy following files

- ▶ `plugins\McalExt_TS_*.templates\Spi_UserCode.h`
- ▶ `plugins\McalExt_TS_*.templates\Spi_UserCode.c`

into your user application and add them to your build environment.

Please change following macros in `Spi_UserCode.h` to your needs

- ▶ `#define QSPI_SCR_RX_PRIO (4u)`
- ▶ `#define QSPI_SCR_TX_PRIO (5u)`

This example will map QSPI2RX to DMA channel 4 and QSPI2TX to DMA channel 5.

In order to map QSPI RX and TX the template function `Spi_InitDMAUsr` should be executed after `Spi_Init` function. Therefore, please add include `Spi_UserCode.h` in `BswM_UserCallouts.c` after include of `Spi.h`. Additionally add function call `Spi_InitDMAUsr` after `Spi_Init`, e.g. `SPI_INIT_FUNC` function macro.

## 6.5. MCAL version update

When the MCAL version is updated, you must verify the following:

- ▶ Check if all the used files are at the same location. If the vendor does not modify the MCAL installed structure, then the used path should be the same.
- ▶ During integration, Elektrobit Automotive GmbH applies patches on the vendor files due to some bugs or due to some incompatibility. These patches are located in the files with the `.EB_update` extension. When you install the new MCAL version, you must to verify if those patches need to be applied on the new MCAL version files.