

MSP_AUTO_SDK_2.0.0.STS Release Notes

Release Notes

Table of Contents

- Introduction
- Licensing
- Getting Started
- Datasheet
- Documentation
- Dependencies
- What's New
- Device Support
- Tools and Compilers
- Compatibility
- Validation
- Fixed Defects
- Open Defects
- Known Limitations
- Support
- Versioning

Introduction

This is the release notes for **MSP_AUTO_SDK_2.0.0.STS** release, done on 2026-10-21.

The package consists of MCAL Driver & Applications for **MSP family of devices** along with **SDL+MCAL** integrated example using prebuild SDL libs. The MCAL modules are compliant to AUTOSAR specification version **4.3.1**. Except lin slave which is compliant to AUTOSAR version **4.4.0**

Note: Only HSFS device is supported.

Licensing

Refer to MSP manifest at top level for `MSP_AUTO_SDK_2.0.0.STS_manifest.html`

Getting Started

The MSP User Guide (present as part of `mcaldocs`) provides the documentation and references necessary to begin development on TI's platforms using MSPM33C321x/MSPM0G350x/MSPM0G351x.

Datasheet

User Guide provides the Memory footprint details.

Documentation

The MSP User Guide (present as part of `mcal_docs`) provides the documentation and references necessary to begin development on TI's platforms using MSPM33C321x/MSPM0G350x/MSPM0G351x.

This document details about supported driver, installation, dependencies, build instructions, steps to run example applications. This release package also includes module specific User Guides, configurator User Guide and others.

Note: Documentation details regarding integrated SDL+MCAL example is present as part of the readme file present inside the ``Mcal_sdl`` example present as part of ``mcal/example``.

Dependencies

NA

What's New

1. New in this Release:
 - Unified MSP MCAL as part of the installer
 - **MSPM0G350x: Full Functional Testing:** ADC, CAN, CDD_DMA, CDD_I2C, CDD_UART, DIO, FLS, GPT, ICU, LIN, MCU, PORT, PWM, SPI, WDG, FEE
 - **MSPM0G351x: Full Functional Testing:** ADC, CAN, CDD_DMA, CDD_I2C, CDD_UART, DIO, FLS, GPT, ICU, LIN, MCU, PORT, PWM, SPI, WDG, FEE, CRYPTO, CRC, CDD_COMP, WDG_44_IWDG
 - **MSPM33C321x: Full Functional Testing:** ADC, CAN, CDD_DMA, CDD_I2C, CDD_UART, DIO, FLS, GPT, ICU, LIN, MCU, PORT, PWM, SPI, WDG
 - New Module Addition: CRYPTO, CRC, CDD_COMP, WDG_44_IWDG (only for MSPM0G351x)
 - New Feature Addition: CDD_I2C Target for MSPM0G351x
 - Integrated **SDL+MCAL** example is present as part of the `mcal/examples` in order to showcase the compatibility between the two. This includes pre-build `sdl` library
 - Comprehensive User Guide and Datasheet
 - Bug Fixes

Device Support

SoC	HOST (OS)	Target (OS)	Supported CPU	Test Platform
MSPM33C321x	Windows / Linux, EB configurator only on Windows	Baremetal	M33	MSPM33C321x

MSPM0G350x	Windows / Linux, EB configurator only on Windows	Baremetal	M0plus	MSPM0G350x
MSPM0G351x	Windows / Linux, EB configurator only on Windows	Baremetal	M0plus	MSPM0G351x

Tools and Compilers

1. Code Composer Studio: 20.5.0
2. TI ARM CLANG Compiler Version: 5.1.1
3. Elektrobit Tresos Studio: 29.3 (**EB_Tresos_ACG8.8.10_Installer.zip**). Please use link to request access to EB Tresos Studio and License: [Click Here](#)

Compatibility

*Internal Files are organized in V0, V1 and V3 folders. The below table lists the associated V0/V1/V3 files to be used as per device.

#	Modules	MSPM0G350x	MSPM0G351x	MSPM33C321x
1.	MCU	V0	V1	V3
2.	PORT	V0	V0	V0
3.	GPT	V0	V1	V3
4.	PWM	V0	V1	V3
5.	CDD DMA	V0	V0	V0
6.	DIO	V0	V0	V0
7.	SPI	V0	V0	V1
8.	WDG	V0	V0	V0
9.	FLS	V0	V0	V1
10.	ADC	V0	V0	V1
11.	CAN	V0	V0	V0
12.	LIN	V0	V0	V1
13.	CDD_UART	V0	V0	V1
14.	CDD_I2C	V0	V0	V1
15.	ICU	V0	V0	V0

16.	FEE	V0	V0	NA
17.	CRYPTO	NA	V0	NA
18.	CRC	NA	V1	NA
19.	CDD_COMP	NA	V0	NA
20.	WDG_44_IWDG	NA	V0	NA

Validation

#	Module	Validation Scope	Remarks (Refer to <i>Open Defects</i> for details)
1.	ADC	Full Functional Level testing	None
2.	CAN	Full Functional Level testing	None
3.	DIO	Full Functional Level testing	None
4.	CDD_DMA	Full Functional Level testing	None
5.	FLS	Full Functional Level testing	Refer to open defects
6.	GPT	Full Functional Level testing	None
7.	MCU	Full Functional Level testing	Reset Reason tested only for Boot Reset
8.	PORT	Full Functional Level testing	None
9.	PWM	Full Functional Level testing	None
10.	SPI	Full Functional Level testing	None
11.	WDG	Full Functional Level testing	None

12.	LIN	Full Functional Level testing	None
13.	CDD_UART	Full Functional Level testing	None
14.	CDD_I2C	Full Functional Level testing	Refer to open defects
15.	ICU	Full Functional Level testing	Refer Known Limitations
16.	FEE	Full Functional Level testing	None
17.	CRYPTO	Full Functional Level testing	None
18.	CRC	Full Functional Level testing	None
19.	CDD_COMP	Full Functional Level testing	None
20.	WDG_44_IWDG	Full Functional Level testing	None

Fixed Defects

ID	Summary	Platform
MSPMCAL-10719	Review and align MCAL variable placement with AUTOSAR memory mapping guidelines	MSPM0G351x
MSPMCAL-10676	Variable definition of Port_InitDone is not inside a memory section	MSPM0G351x
MSPMCAL-10659	[Port][ARXML] The value of the enumeration parameter /Port/PortGeneral[0:PortDeviceVariant] (value=MSPM0G351X_LQFP_100_PZ100) is not in the list of allowed ones	MSPM0G351x

MSPMCAL-10657	[Mcu] match event channel publisher and subscriber from datasheet	MSPM0G350x /MSPM0G351x/ MSPM33C321x
MSPMCAL-10652	[PWM] PwmDeadbandEnabled parameter missing in the generated ARXML file	MSPM0G351x
MSPMCAL-10651	[LIN] Using Category 2 for ISR in the LIN module leads to error	MSPM0G351x
MSPMCAL-10650	[GPT] GptType parameter missing in the generated ARXML file	MSPM0G351x
MSPMCAL-10649	[CDD_DMA] CddDmaEventSettingRef parameter missing in the generated ARXML file	MSPM0G351x
MSPMCAL-10469	Port Module does not clear Polarity on reinitialisation	MSPM0G350x /MSPM0G351x/ MSPM33C321x
MSPMCAL-10467	Port Module does not disable GPIO interrupts on reinitialisation	MSPM0G351x
MSPMCAL-10326	LIN is having a lot of interrupts - need to optimize the code	MSPM33C321x
MSPMCAL-10043	Watchdog-GPT interrupt race condition causing system resets	MSPM0G351x
MSPMCAL-10022	[Lin]auto baud rate detection can't be disabled	MSPM0G351x
MSPMCAL-10021	Limitation with detecting stop bit errors or framing errors with LIN MCAL	MSPM0G351x
MSPMCAL-9569	Generation issues with Davinci Tool - Gpt Incorrect Enumeration Definitions P1	MSPM0G351x
MSPMCAL-9568	User guide Build instructions Compiler Flags not complete	MSPM0G351x
MSPMCAL-9512	m33: CDD_I2C: Driver is not working fine in Fast+ mode (1MHz freq)	MSPM33C321x
MSPMCAL-9505	CanIf_TriggerTransmit callback violated AUTOSAR Specification of CAN Interface with incorrect pointer semantics	MSPM0G351x

MSPMCAL-9504	CryptoPriv_TurnOffTrng Function not invoked at error state of TRNG	MSPM0G351x
MSPMCAL-9503	Indefinite wait at CryptoPriv_encryptDecryptProcess when invalid input parameter like 0 input length or Null Ptr to inputptr given	MSPM0G351x
MSPMCAL-9420	Variables/constants "eraseLoadFlashSize" and "eraseLoadFlashStart" in the file Fls_Ctrl.c are mapped to a wrong memory section	MSPM0G351x
MSPMCAL-9055	[CDD_UART] Certain UART interrupts are configurable from EB Tresos but are not handled in the ISR handler	MSPM0G351x
MSPMCAL-9016	[Gpt] The GptChannelConfiguration is not generated with symbols	MSPM0G351x
MSPMCAL-8994	[CDD_UART] Data already existing in the RXFIFO is not flushed before a fresh receive operation	MSPM0G351x
MSPMCAL-8987	[CDD_DMA] Remove the runtime DET error report from the Cdd_Dma_disableDMATransfer() function	MSPM0G351x
MSPMCAL-8962	[LIN] The LIN private file for Commander does not supporting wake-up logic correctly	MSPM0G351x
MSPMCAL-8733	[CAN] ECUC Wakeup source ref is incorrectly set	MSPM0G351x
MSPMCAL-6979	[MCU] Incorrect McuClockReferencePointFrequency in Recommended xdm of MCU	MSPM0G351x

Open Defects

ID	Summary	Workaround
MSPMCAL-11210	MSPM0G351x CDD_I2C Target testcase with dma fails to receive 1024 bytes of data	If a DMA receive is ongoing but not yet complete, stop it and mark the transfer as done. Notify the upper layer about the incomplete transfer, unless it was a clean full-buffer overwrite (in which case it was already notified).

MSPMCAL-11190	MSPM33C321x MCU application Sleep does not wait for external GPIO trigger	Customer to not use sleep mode in M33, still other low power modes can be used.
MSPMCAL-11073	MSPM0G351x Blocked SPI Communication Due to Permanently Pending Sequence in Async Mode	
MSPMCAL-11072	MSPM0G351x FEE and FLS status are not aligned with each other causing NvM block Corruption	
MSPMCAL-10687	MSPM0G351x: [Mcu] Low power mode SYSPLL_ERR_01 handling: says syspll should be disabled before going to LPM	Before applying the WFI() in Mcu_LowPowerModeSetting Api, we can check if syspll is enabled or not and if it is enabled we can swith the mclk to sysosc and turn syspll off and then after WFI line we can re-enable syspll again and swith mclk to source from syspll instead of sysosc.
MSPMCAL-10442	[I2c] Function return value need to pass to upper layer	This issue tells that even if transfer is failed, we are making Channel Status is CDD_I2C_CH_RESULT_OK. But In the Driver, all results are based on sequence. so even if channel result is OK, if sequence result is failed, final result is still failed. The bug only is there if the caller relies solely on Cdd_I2c_GetResult(). Anyone using Cdd_I2c_GetSequenceResult() is already safe with no code changes needed.
MSPMCAL-9098	MSPM0G350x/MSPM0G351x: [LIN] Commander mode fails ISO 17987-6 bit error handling conformance test	Driver level changes done, just need to have conformance test done again. Planned post release for conformance test.
MSPMCAL-5909	m33: Fls: Write operation(interrupt mode) on bank1 is not functional	Use bank0 for execution of MCAL and Fls_write() or Use bank1 for execution of MCAL and Fls_write() Cross will not work.

Known Limitations

- All reset reasons can't be tested.
- LFCLK_IN and HFCLK_IN need external signals (not tested).

Support

For technical support and additional assistance, visit [E2E](#) or contact local TI Field Application Engineer.

Versioning

Each package version is composed of 4 period-delimited numbers - represented here by the letters M, m, p and b [**M.m.p.b**]. The table below provides a descriptive reference regarding package version numbering.

Digit	Meaning	Description
1 (M =Major)	Major revision	Incremented when the new version is substantially different from the previous for example, a new module added or an existing modules algorithm significantly altered.
2 (m =minor)	Minor revision	Incremented when the new version has changed but not in a major way. For example, some minor changes in the API or feature set.
3 (p =patch)	Patch number	Incremented for all other source code changes. This includes any packaging support code.
4 (b =build)	Build number	Incremented for each release delivery to CM. Reset for any change to M, m or p

TI is a global semiconductor design and manufacturing company. Innovate with 100,000+ analog ICs and embedded processors, along with software, tools and the industry's largest sales/support staff.

© Copyright 1995-2023, Texas Instruments Incorporated. All rights reserved.
[Trademarks](#) | [Privacy policy](#) | [Terms of use](#) | [Terms of sale](#)