

Getting started with T2G MCAL using TRAVEO™ T2G MCU

TRAVEO™ 32-bit microcontroller family

About this document

Scope and purpose

This application note provides a brief overview of the tools and configurations necessary to get started with building TRAVEO™ T2G microcontroller abstraction layer (MCAL) code examples.

This application note also introduces you to the software development ecosystem in place for getting started with the TRAVEO™ T2G MCAL and gets you started with a simple blinking LED example using the T2G MCALs.

Intended audience

This document is intended for anyone who wanted to get started with the TRAVEO™ T2G MCAL.

Software/tools versions

- [TRAVEO™ T2G sample driver library](#) (SDL) v8.2.0
- TRAVEO™ T2G MCAL package v1.19.0 or above
- EB Tresos v 26.2.0
- Green Hills/IAR Compiler versions as supported by the T2G MCAL and TRAVEO™ T2G sample driver library

Note: Contact [Infineon support](#) to get access to the TRAVEO™ T2G MCAL package that also contains the EB Tresos v26.2.0 installer.

Table of contents

Table of contents

About this document.....	1
Table of contents.....	2
1 Introduction	3
2 Development ecosystem	4
2.1 Traveo™ T2G family related documents	4
2.2 Application development environment	4
2.2.1 TRAVEO™ sample driver library	4
3 Building a TRAVEO™ T2G MCAL example using SDL environment	8
3.1 Prerequisites.....	8
3.2 Building a T2G MCAL code example	9
3.3 Launching EB Tresos to modify the configurations	11
References.....	12
Revision history.....	13
Disclaimer.....	14

Introduction

1 Introduction

This application note provides information on the prerequisites those are needed to get started with building an MCAL code example using the SDL environment. It also provides an example of how to build the basic code example that is part of the SDL environment, the custom CMake targets to launch the debugger, and launching the EB Tresos GUI to be able to modify the MCAL configurations.

The TRAVEO™ T2G family MCU device is a microcontroller targeted for automotive applications. The TRAVEO™ T2G family CYT4BF MCU integrates the following features on a single chip:

- Up to two 350-MHz 32-bit Arm® Cortex®-M7 CPUs each with the following features:
 - Single-cycle multiply
 - Single-/double-precision floating point unit (FPU)
 - 16-KB data cache, 16-KB instruction cache
 - Memory protection unit (MPU)
 - 16-KB instruction and 16-KB data tightly coupled memory (TCM)
- 100-MHz 32-bit Arm® Cortex® M0+ CPU with single-cycle multiply and memory protection unit (MPU)
- Programmable analog and digital peripherals
- Up to 8384 KB of code flash with an additional up to 256 KB of work flash and an internal SRAM of up to 1024 KB
- TRAVEO™ T2G family CYT4BF MCU is suitable for a variety of power-sensitive applications such as:
 - Body control module (BCM)
 - Gateway module
 - Infotainment
 - Lighting
 - Network sound system

For more details on getting started with the TRAVEO™ T2G family MCU and ModusToolbox™ toolchain, refer to [\[2\]](#).

2 Development ecosystem

2.1 TRAVEO™ T2G family-related documents

The Infineon TRAVEO™ webpage [\[1\]](#) contains documents regarding selection of the right device and helps you in integrating the device in your application design. You can also refer to [\[2\]](#).

2.2 Application development environment

For developing application/example code using the T2G MCALs, you can use the TRAVEO™ sample driver library environment.

2.2.1 TRAVEO™ sample driver library

Infineon sample driver library (SDL) simplifies software development for TRAVEO™ T2G devices by providing software drivers as-is that could be used for evaluation purposes. The SDL consists of the following:

- Drivers for an extensive set of peripherals along with device-specific header files that provide a complete definition of all peripheral registers.
- Arm® Cortex® Microcontroller Software Interface Standard (CMSIS) core header files directly from the CMSIS 5.7 release.
- CMSIS-compliant start-up code to initialize the system after device reset, and transfer the code execution to main().
- Template workspaces, linker files for each supported device, and toolchain (IAR Systems and Green Hills software).
- CMake and Ninja-based build environment to easily getting started by developing your application and generating an image/binary.

The SDL package can be downloaded from the [TRAVEO™ T2G sample driver library](#). In case you do not see the SDL package, contact [Infineon Support](#).

The SDL installer is to be launched and the instruction on the wizard is to be followed to successfully install the package.

Development ecosystem

The SDL folder structure is explained below.

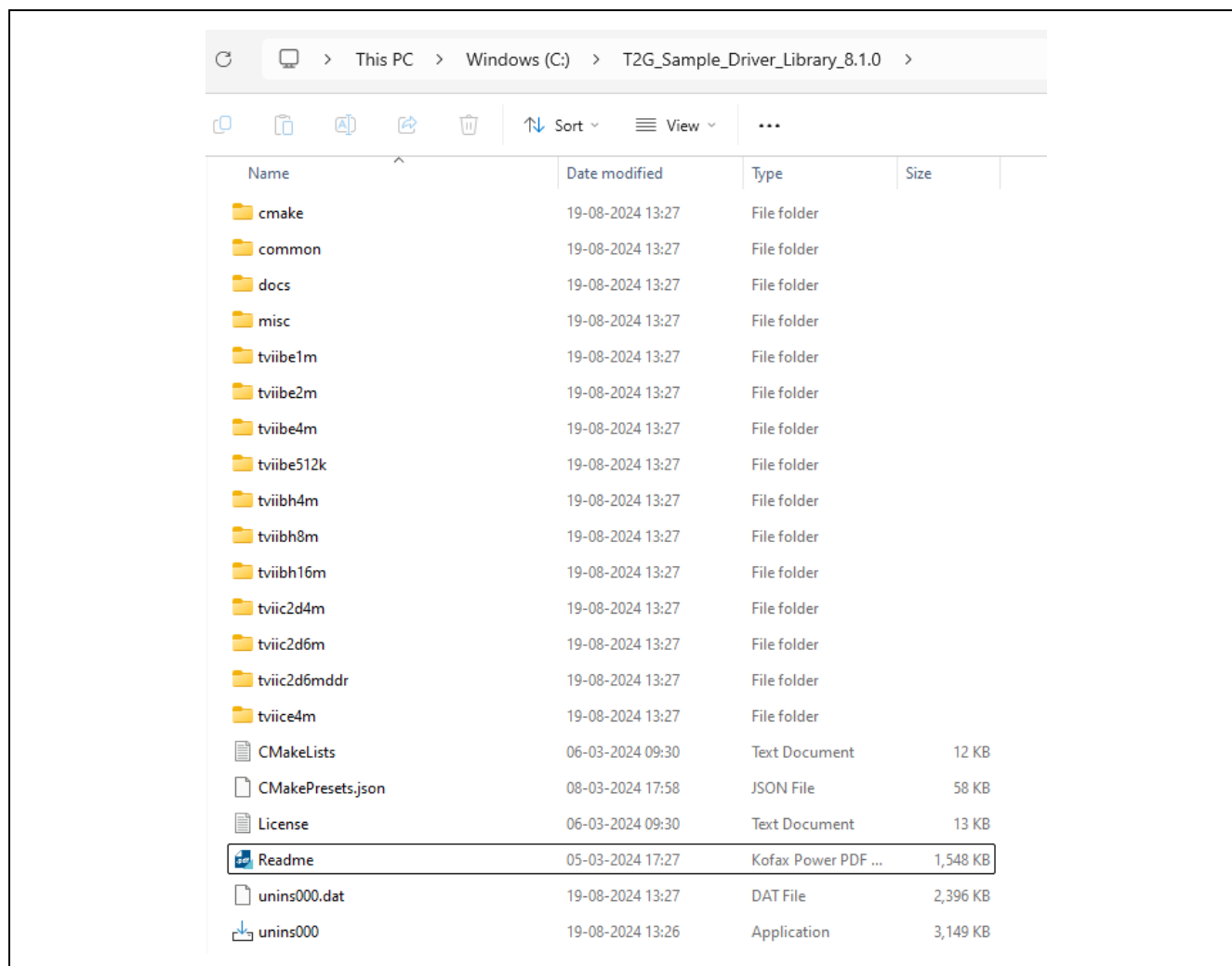


Figure 1 TRAVEO™ T2G sample driver library folder structure

Table 1 TRAVEO™ T2G sample driver library folder structure

Path/folder	Description
common/hdr/cmsis	Common Microcontroller Software Interface Standard (CMSIS) core access headers
common/src/drivers	Drivers common across all the devices
common/src/mw	Middleware common across all the devices
common/src/startup	Tool specific startup code for all the devices
docs	SDL application programming interface (API) documentation, release notes, known issues
misc/tools	GHS/IAR specific flash loaders, system view description (SVD) files

TRAVEO™ T2G body and cluster entry devices (tviibe1m/2m/4m/512k, tviice4m)

Development ecosystem

Path/folder	Description
hdr	Device-specific header files, board support package (BSP) for T2G base board/CPU boards, (general-purpose input/output) GPIO assignments
hdr/ip	Device intellectual property (IP) specific headers
hdr/mcureg	IP specific register addresses
src/drivers	Driver source and respective headers specific to TVIIBE1M/2M/4M/512K, TVIICE4M device
src/examples	Code examples in accordance with TVIIBE1M/2M/4M/512K, TVIICE4M device
src/system	TVIIBE1M/2M/4M/512K, TVIICE4M system specific code and system header for clock configurations
src/interrupts/cy_interrupt_map_cm0plus.h	User interrupt mapping file
src/interrupts/cy_interrupt_map_cm4.h	User interrupt mapping file
src/main_cm0plus.c	Sample main source file for CM0+ core
src/main_cm4.c	Sample main source file for CM4 core
tools/ghs	GHS MULTI workspaces for SRAM/Flash for CM0+/CM4 cores, linker specific files, and GRD files (Register files that is used by the GHS debugger)
tools/iar	IAR workspaces for SRAM/Flash for CM0+/CM4 cores, linker specific files

TRAVEO™ T2G body high and cluster 2D devices (tviibh4m/8m, tviic2d4m/6m/6mddr)

hdr	Device-specific header files, BSP for T2G base board/CPU boards, GPIO assignments
hdr/ip	Device IP specific headers
hdr/mcureg	IP specific register addresses
src/drivers	Driver source and respective headers specific to TVIIBH4M/8M, TVIIC2D4M/6M/6MDDR devices
src/mw/	Middleware support
src/examples	Device specific code examples
src/system	Device system specific code and system header for clock configurations
src/interrupts/cy_interrupt_map_cm0plus.h	User interrupt mapping file
src/interrupts/cy_interrupt_map_cm7_0.h	User interrupt mapping file
src/interrupts/cy_interrupt_map_cm7_1.h	User interrupt mapping file
src/main_cm0plus.c	Sample main source file for CM0+ core
src/main_cm7_0.c	Sample main source file for CM7_0 core
src/main_cm7_1.c	Sample main source file for CM7_1 core
tools/ghs	GHS MULTI workspaces for SRAM/Flash for CM0+/CM7_0/CM7_1 cores, linker specific files, and GRD files

Development ecosystem

Path/folder	Description
tools/iar	IAR workspaces for SRAM/Flash for CM0+/CM7_0/CM7_1 cores, linker specific files

The AUTO TRAINING: TRAVEO™ II Sample driver library document contains information on how to compile and debug SDL based examples using the supported compilers, integrated development environment (IDE) and debuggers template files that are part of the SDL package, which is not covered in this document.

Note: To get access to the TRAVEO™ T2G sample driver library (SDL) training, contact [Infineon Support](#) and get access to AUTO TRAINING: TRAVEO™ II Sample driver library PDF file.

Getting started with T2G MCAL using TRAVEO™ T2G MCU

TRAVEO™ 32-bit microcontroller family

Building a TRAVEO™ T2G MCAL example using the SDL environment

3 Building a TRAVEO™ T2G MCAL example using the SDL environment

3.1 Prerequisites

Before starting, ensure to have the appropriate development kit for the TRAVEO™ T2G family MCU product line, required compilers, debuggers, and IDEs along with necessary licenses and the SDL software package are installed. Once the SDL package is installed, follow the steps mentioned in the AUTO TRAINING: TRAVEO™ II Sample driver library document to patch the compiler-related files to enable support for TRAVEO™ T2G family.

The SDL supports the CPU evaluations boards by default after installation.

The TRAVEO™ T2G MCAL package needs to be installed. Contact [Infineon support](#) to get a license and access to the MCAL package. The MCAL installation guide part of the package can be referred to install EB Tresos toolchain, MCALs, and BSW stubs package.

The SDL supports a combination of CMake and Ninja for compiling the MCAL examples, the Cmake, and Ninja needs to be installed by the user.

Install Cmake from cmake.org.

Install Ninja from ninja-build.org.

After installing the CMake and Ninja, make sure the Windows 'PATH' environment variable is updated with the paths where CMake and Ninja are installed in your machine. See the example below.

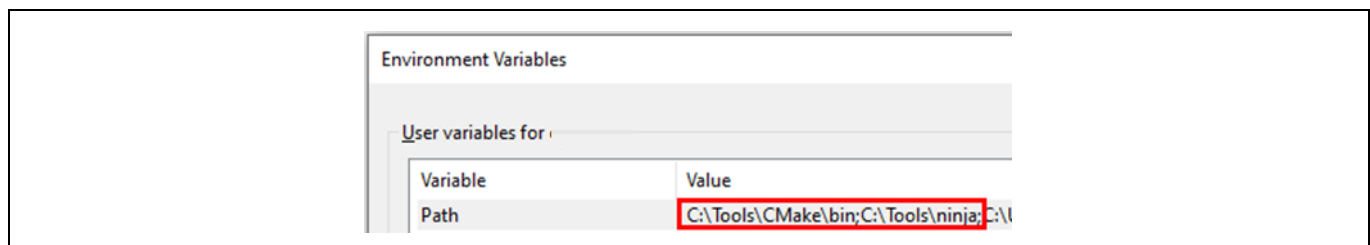


Figure 2 Update PATH environment variable as per where Cmake and Ninja are locally installed

Once the compiler, IDE, and MCAL packages are installed, the paths where these tools are installed in your local machine needs to be configured in the 'tool_config.cmake' located at <Local SDL installation>\cmake\.

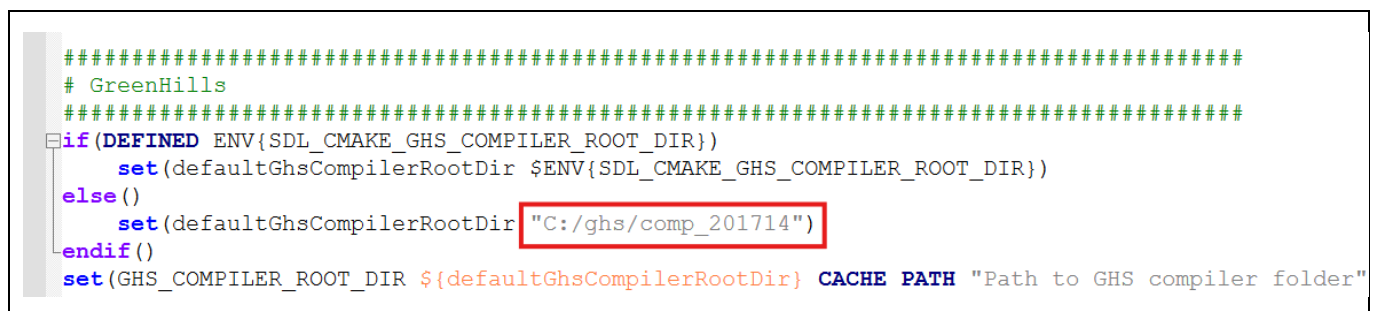


Figure 3 Update the path where the GHS compiler is locally installed

Getting started with T2G MCAL using TRAVEO™ T2G MCU

TRAVEO™ 32-bit microcontroller family

Building a TRAVEO™ T2G MCAL example using the SDL environment

```
#####
# MCAL / Tresos (only relevant if USE_MCAL is true)
#####
# Tresos/MCAL directory (having 'bin', 'plugins', etc. subfolders)
if(DEFINED ENV{SDL_CMAKE_MCAL_VAR_TRESOS_DIR})
    set(defaultMcalVarTresosDir $ENV{SDL_CMAKE_MCAL_VAR_TRESOS_DIR})
else()
    set(defaultMcalVarTresosDir "C:/INFINEON_ESDB/Tresos26_2_0")
endif()
set(MCAL_VAR_TRESOS_DIR ${defaultMcalVarTresosDir} CACHE PATH "Path to Tresos root folder")
```

Figure 4 Update the path where the MCAL package is locally installed

Note: The CMake ReadMe file present at <Local SDL installation>\cmake\CMake_Readme.md provides further information on the supported commands.

3.2 Building a T2G MCAL code example

The SDL along with Cmake environment can be used to build an MCAL code example. The SDL package consists of a basic MCAL code example that can be used to get started, it can be found at <Local SDL installation>\tviibh8m\src\examples\mcaldio_gpt_icu_mcu_port. The basic code example uses the MCU, PORT, digital input/output (DIO), general-purpose timer (GPT), and input capture unit (ICU) MCAL modules.

The code example contains preconfigured EB Tresos XML data-for-mat (XDM) files located at <Local SDL installation>\tviibh8m\src\TVII-B-H-8M_BGA-320 or <Local SDL installation>\tviibh8m\src\TVII-B-H-8M_QFP-176 based on which pin package you are using. The SDL environment by default uses the ball grid array (BGA) 320-pin package.

The MCAL test code is part of the main_cm7_0.c file.

The basic code examples showcase the following functionalities:

- The MCU module is used to initialize the clocks.
- PORT and DIOs modules are used to configure and initialize the onboard USER LEDs and USER SW button.
- GPT module is used to configure timer interrupts for USER LED toggling.
- ICU module is used to detect inputs from the USER SW button.

The commands to be executed to build the basic code example using IAR IDE (compiler and debugger) are shown below:

1. Copy the contents from mcaldio_gpt_icu_mcu_port example folder <Local SDL installation>\tviibh8m\src\examples\mcaldio_gpt_icu_mcu_port into your src folder <Local SDL installation>\tviibh8m\src.

Getting started with T2G MCAL using TRAVEO™ T2G MCU

TRAVEO™ 32-bit microcontroller family

Building a TRAVEO™ T2G MCAL example using the SDL environment



Figure 5 Copy the contents from mcal example to the SRC folder

- Go to the SDL installed folder and open a command line.
- Use the CMake command 'cmake --preset tviih8m-iar -D USE_MCAL=ON' to setup the user settings, command-line parameters, etc. are evaluated and the files for the native build system are generated. Use the respective preset command based on the compiler that you have installed. This command also generates the configuration files from the preconfigured Tresos XDM files. Check for any errors or warnings during EB Tresos configuration generation, if no errors/warnings proceed to the next step.

Note: Refer the CMake ReadMe file present at <Local SDL installation>\cmake\CMake_Readme.md provides further information on the supported presets and commands.

Note: Depending on the compiler installed, use the respective preset.

```
C:\T2G_Sample_Driver_Library_7.9.0>cmake --preset tviih8m-iar -D USE_MCAL=ON
Preset CMake variables:

  CMAKE_TOOLCHAIN_FILE:FILEPATH="C:/T2G_Sample_Driver_Library_7.9.0/cmake/toolchains/toolchain-iar.cmake"
  DIE="tviih8m"
  LINK="flash"
  MCU_REV="latest"

-- Latest MCU revision detected: D
-- The C compiler identification is IAR ARM 9.30.1
-- The ASM compiler identification is IAR ARM 9.30.1
-- Found assembler: C:/Tools/IARSystems/EmbeddedWorkbench91/arm/bin/iasmarm.exe
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: C:/Tools/IARSystems/EmbeddedWorkbench91/arm/bin/iccarm.exe - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- IAR Linker File determined: C:/T2G_Sample_Driver_Library_7.9.0/tviih8m/tools/iar/linker_directives_tviih_rev_c.icf
=====
-- T2G SDL - CMake Configuration
=====
-- Toolchain      = iar
-- DIE            = tviih8m
-- MCU revision   = D
-- Device         = CYT48FCCHE
-- Link location  = flash
-- PSVP           = OFF
-- Board          =
-- Board revision =
-- Use RTOS       = ON
-- Use ETH Stack  = OFF
-- Use MCAL       = ON
-- Tresos         = C:/INFINEON_ESDB_V19/Tresos26_2_0
-- Tresos workspace = C:/T2G_Sample_Driver_Library_7.9.0/tresos_workspace
-- Tresos project  = C:/T2G_Sample_Driver_Library_7.9.0/tviih8m/src/TVII-B-H-8M_BGA-320
=====
```

Figure 6 Run the preset command to setup the environment and generate an EB Tresos configuration

Getting started with T2G MCAL using TRAVEO™ T2G MCU

TRAVEO™ 32-bit microcontroller family

Building a TRAVEO™ T2G MCAL example using the SDL environment

- Execute the build command 'cmake --build --preset tviibh8m-iar' to compile and link all the required files using IAR Compiler. Progress would be shown on the command line and any errors/warnings would also be shown.

```
C:\T2G_Sample_Driver_Library_7.9.0>cmake --build --preset tviibh8m-iar
[250/251] Linking C executable Debug\cm7_1.elf
Warning[Li065]: duplicate file: "C:\T2G_Sample_Driver_Library_7.9.0\build\tviibh8m-iar\Debug\libmcalstubs_cm7.a"
Warning[Li065]: duplicate file: "C:\T2G_Sample_Driver_Library_7.9.0\build\tviibh8m-iar\Debug\libmcal_cm7.a"
[251/251] Linking C executable Debug\cm7_0.elf
Warning[Li065]: duplicate file: "C:\T2G_Sample_Driver_Library_7.9.0\build\tviibh8m-iar\Debug\libmcalstubs_cm7.a"
Warning[Li065]: duplicate file: "C:\T2G_Sample_Driver_Library_7.9.0\build\tviibh8m-iar\Debug\libmcal_cm7.a"
```

Figure 7 Command to build and link all required files

- The SDL CMake environment also supports custom targets to launch the debugger. Execute the command 'cmake --build --preset tviibh8m-iar --target dbg_iar_cm0plus_cm7_0' to launch the debugger template for debugging CM0 and CM7_0 cores. This command launches the IAR debugger for the CM0+ core.

```
C:\T2G_Sample_Driver_Library_7.9.0>cmake --build --preset tviibh8m-iar --target dbg_iar_cm0plus_cm7_0
[0/1] C:\WINDOWS\system32\cmd.exe /C "cd /D C:\T2G_Sample_...uild\tviibh8m-iar\Debug\dbg_iar\cm0plus_cm7_0\cm0plus.eww"
```

Figure 8 Custom target to launch IAR debugger

- Use the debugger of choice to further debug the application by either checking the counter variables or the hardware registers. The basic code example also prints a message on the UART, any serial terminal (For example, TeraTerm) can be used to check the print messages on the terminal.

3.3 Launching EB Tresos to modify the configurations

The CMake also supports a custom target to launch Tresos tool for the users to add/modify the configurations of the MCAL modules. In case of any errors during the Tresos configuration generation this command can be used to launch the EB Tresos tool to fix the errors.

To launch the EB Tresos tool, use the command 'cmake --build --preset tviibh8m-iar --target util_tresos_gui'. This invokes the EB Tresos tool choosing the project files that are part of the basic code example located at <Local SDL Installation>\tviibh8m\src\TVII-B-H-8M_BGA-320 or <Your SDL installed folder>\tviibh8m\src\TVII-B-H-8M_QFP-176 based on which pin package you are using.

```
C:\T2G_Sample_Driver_Library_7.9.0>cmake --build --preset tviibh8m-iar --target util_tresos_gui
[0/1] C:\WINDOWS\system32\cmd.exe /C "cd /D C:\T2G_Sample_...-data C:\T2G_Sample_Driver_Library_7.9.0\tresos_workspace"
```

Figure 9 Command to launch the EB Tresos GUI

References

References

TRAVEO™ T2G webpage

- [1] [32-bit TRAVEO™ T2G Arm® Cortex® microcontroller](#)

Application notes

- [2] Infineon Technologies AG: AN235305 - Getting started with TRAVEO™ T2G family MCUs in ModusToolbox™;
[Available online](#)

For more TRAVEO™ T2G documents, contact [Technical Support](#).

Revision history**Revision history**

Document revision	Date	Description of changes
V 1.0	10-01-2025	Initial release.

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2025-01-10

Published by

Infineon Technologies AG

81726 Munich, Germany

**© 2025 Infineon Technologies AG.
All Rights Reserved.**

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

AN0032

Important notice

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

Warnings

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.