



Harshit Gupta, Nihar Potturu, Tarun Mukesh Puvvada

EP ASM Automotive

摘要

AUTOSAR® 是汽车软件开发的标准化方法。微控制器抽象层 (MCAL) 符合 **AUTOSAR 经典平台** 规范。MCAL 给硬件组件配备了一个接口，通过将多个类别的驱动程序与硬件组件捆绑，管理相应的组件集并提供多种受支持的功能。MCAL 软件包是某一目标软件包所专用的，换言之，MCAL 依赖于硬件，可用于每种受支持的器件系列中，例如 Sitara™ AM263x、AM263Px、AM273x MCU、AM62x、AM62Ax、AM62Px MPU、Jacinto™ DRA8x 和 TDA4x 处理器。

未被 AUTOSAR 标准化的其他器件外设和硬件单元使用复杂驱动器设计 (CDD) 以扩展集成的形式交付。虽然该软件包是以术语 **MCAL** 软件包的名义推广的，但该软件包包括 MCAL 和 CDD 层的驱动程序。软件包以 **BSW** 存根的形式提供上层的任何依赖要求。**BSW** 存根是上层组件的不完全实现，能连接驱动程序层以实现独立的功能包交付。该组件将由主要 AUTOSAR 供应商提供的完整 AUTOSAR 堆栈中的完整组件替代。

内容

1 简介.....	2
1.1 配置程序.....	3
2 软件栈.....	4
2.1 Configurator Plug-in.....	4
2.2 源文件.....	10
2.3 用户指南.....	11
2.4 合规性支持包 (CSP).....	11
3 使用源-构建驱动程序和示例.....	11
3.1 一次性设置.....	11
3.2 构建说明.....	12
3.3 构建命令语法.....	12
4 获取 MCAL 软件包.....	14
5 总结.....	14
6 参考资料.....	14

插图清单

图 1-1. AUTOSAR 4.3.1 架构.....	3
图 3-1. Make 的输出示例.....	13

表格清单

表 1-1. 缩写和首字母缩略词的完整格式扩展.....	2
------------------------------	---

商标

Sitara™ and Jacinto™ are trademarks of Texas Instruments.
 AUTOSAR® is a registered trademark of AUTOSAR GbR.
 Arm® and Cortex® are registered trademarks of Arm Limited.
 tresos® is a registered trademark of Elektrobit Automotive GmbH.
 Microsoft® and Windows® are registered trademarks of Microsoft Corporation.
 Unix® is a registered trademark of The Open Group Limited.
 所有商标均为其各自所有者的财产。

1 简介

表 1-1 显示了首字母缩略词和定义。

表 1-1. 缩写和首字母缩略词的完整格式扩展

首字母缩写词	定义
API	应用程序编程接口
ARXML	AUTOSAR XML
AUTOSAR	汽车开放系统架构
BSW	基础软件
CDD	复杂驱动器设计
CSP	合规性支持包
ECU	电子控制单元
FSQ	功能安全认证
FuSa	功能安全
GUI	图形用户界面
ISO	国际标准化组织
IRQ	中断请求
ISR	中断服务例程
MCAL	微控制器抽象层
MCU	微控制器单元
MPU	微处理器单元 (Microprocessor Unit)
Mip	模块实现前缀
RTE	运行时环境
SIL	器件完整性等级
SDK	软件开发套件
SRS	软件要求规格
SWS	软件规格
TI	Texas Instruments Incorporated
XER5F	在 R5F 内核中可执行

MCAL 与硬件抽象层相同，其负责提供与底层硬件系统（包含处理器、存储器和外设）进行交互的方法，并通过基于归档接口的驱动程序集实现。MCAL 是 AUTOSAR 堆栈下的层之一。

AUTOSAR 是一个开放标准化的汽车开发软件框架，由一群领先的汽车工业家于 2003 年建立。最新版本为 R23-11；成功发布的版本有 R22-11、R21-11、R20-11、R19-11、...、R4.4.0、R4.3.1、R4.3、R4.2、R4.1、R4.0.3、R3.2、R3.1、R3。于 2006 年发布 R2。版本编号为 R<YY>-<MM>，其中 <MM> 为月份，<YY> 为发布年份（两位数），从 R17-03 开始，之前发行的版本在语义上按 v<Major>.<minor>.<Revision> 进行版本控制，直到 v4.4.0 和 4.5.0。这些标准组织提供了经典平台和自适应平台架构下的规范，其中经典平台架构标准包括基出软件 (BSW) 层下的 MCAL。

图 1-1 显示了经典平台标准层架构，它由应用层、运行时环境、BSW 层和下面的微控制器层组成。经典平台架构的范围不仅有应用程序，还有 μC 。

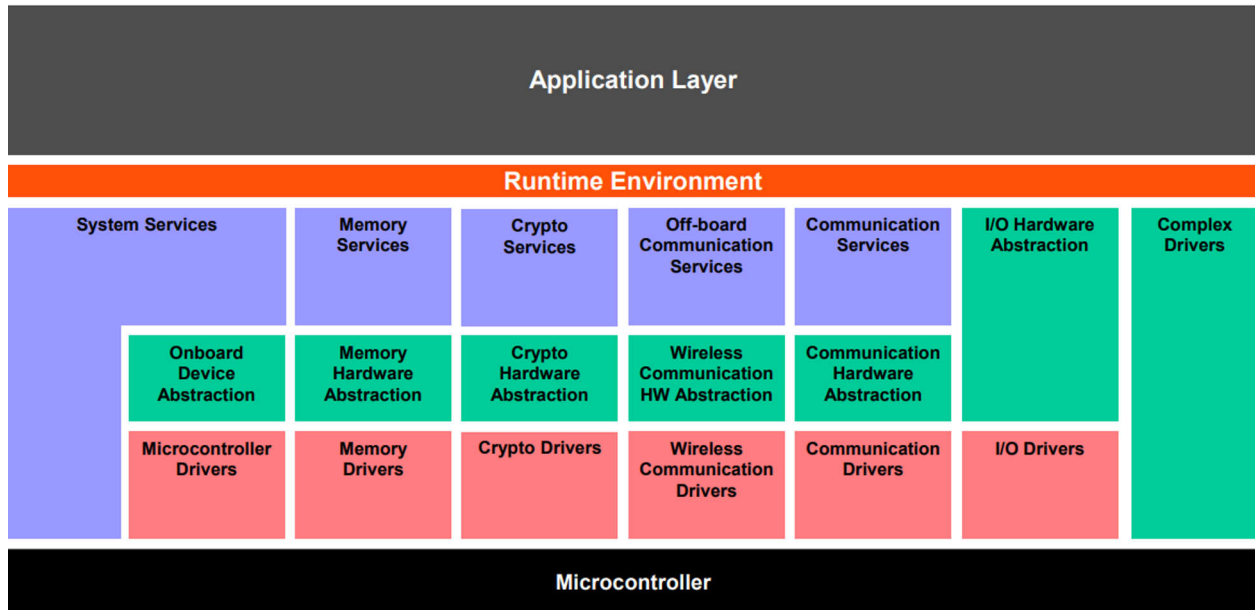


图 1-1. AUTOSAR 4.3.1 架构

夹在运行时环境和微控制器之间的 BSW 层有子层，具体而言是有服务层、ECU 抽象层和 MCAL 层，分别以鲜艳的紫色、绿色和红色标注。虽然复杂驱动程序模块的颜色不明显，但它是 BSW 的另一个子层，完全在微控制器到 RTE 交互的 BSW 内。图 1-1 展示了 MCAL 和复杂驱动程序的细分。

MCAL 软件包旨在德州仪器 (TI) 制造的 Arm® Cortex®-R 内核上多核上运行。在 TI Jacinto™、Sitara™ MCU 和基于 MPU Arm® 的器件系列中，Cortex-R5 内核是支持执行 MCAL 和 CDD 器件驱动程序的目标内核。目前，TI 在这些器件系列上仅提供单核 MCAL 支持。

1.1 配置程序

使用任何符合 AUTOSAR 的配置工具所提供的图形界面，均可轻松完成对任何驱动程序的多项设置。市场上有多种配置程序工具，例如 Elektrobit (EB) tresos® Studio、Vector DaVinci 配置程序和 KPIT K-SAR 等。

TI Sitara MCU、MPU 和 Jacinto 处理器所捆绑的 MCAL 配置插件在 Elektrobit 提供的 EB tresos Studio 配置程序工具上进行开发和测试。Elektrobit tresos Studio 支持配置程序插件的一些附加功能，可在工具上使用，同时不提供直接的跨工具可用性。这包括用于获取节点的、基于模板的代码生成和 XPath (可扩展样式表语言转换路径的缩写) 等功能，这些节点并不是所有符合 AUTOSAR 的 Configurator 工具共用的。

EB tresos Studio 可在 Microsoft® Windows® 的 Desktop GUI 应用程序中获取，也可在基于 Unix® 的系统中作为 CLI 在 <EB_Installation_Path>/bin/tresos_cmd.sh 中使用。

2 软件栈

MCAL 软件包包含一个配置程序插件、驱动程序源文件、每个驱动程序的示例应用程序以及相关文件，具体有用户指南、发行说明以及根据 ISO26262 标准对功能安全合规性进行验证的合规性支持包 (CSP) 文件。

目录结构如下：

- `mcal_config/` - 包含符合 AUTOSAR 配置程序工具标准的配置程序插件，可以测试该插件以加载到 Elektrobit (EB) tresos Studio 配置程序工具中。每个插件对应于每个驱动程序的特定设备变量，并存储在相应的文件夹中。插件可以单独加载以进行数据配置、也可以在单个配置程序项目中使用其他相关插件加载，以便能够引用其他插件中的字段和节点。
- `mcal/` - 包含软件包捆绑的所有驱动程序类别中的所有 MCAL 源文件。此目录通常包含大约 10 到 25 个驱动程序，具体取决于目标器件硬件支持的功能和计划发布的器件驱动程序。
 - `mcal/examples/` - `mcal` 目录中的路径存储每个驱动程序源的示例应用程序，这些驱动程序源演示了初始化流程和驱动程序模块中提供的各种 API 的使用情况。
- `mcal_docs/` - 包含各种文件，具体有发行说明、清单和模块用户指南。发行说明中提到了带有先前版本的 `changelog`，其中包含遇到或修复的新增功能和错误以及已知问题和建议的变通办法。用户指南包含有关驱动程序功能、配置程序插件和 API 的详细信息。
- `build/` - 包括软件包级别的 `makefile` 以及对构建工具进行相关设置的其他相关文件。该路径中的 `make` 命令完成源构建。

具体而言，在 Sitara MPU MCAL 中，除先前引用的 `build/` 之外的文件夹都可以在 `mcal_drv/` 文件夹下找到。若需要 Jacinto MCAL，请从 [PSDK-RTOS-AUTO](#) 提出请求并下载之前介绍的 `mcal_config` 文件夹。

2.1 Configurator Plug-in

Configurator Plug-in 提供模块相关参数的配置选项，可以使用 *Configurator Tool* 编辑和生成这些参数，从而获取可用模块字段的 GUI。任何符合 AUTOSAR 的配置程序工具都可以利用模块插件文件从 ARXML 文件加载可配置字段（具体而言，架构 ARXML 文件加载模型，数据 ARXML 可以为架构中的每个字段提供可加载的预配置值和建议配置值），并生成相应的 *include* 和 *source* 文件。

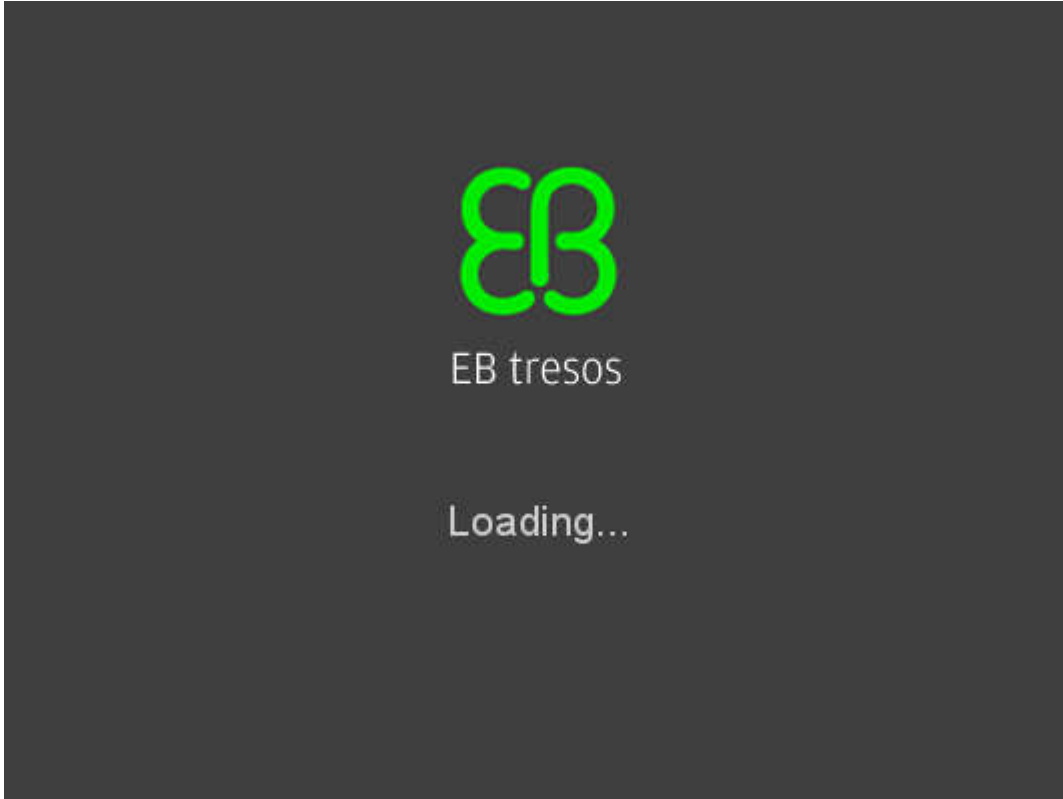
适用于 EB tresos Studio 的 *Configurator Plug-in* 包括以下文件内容：

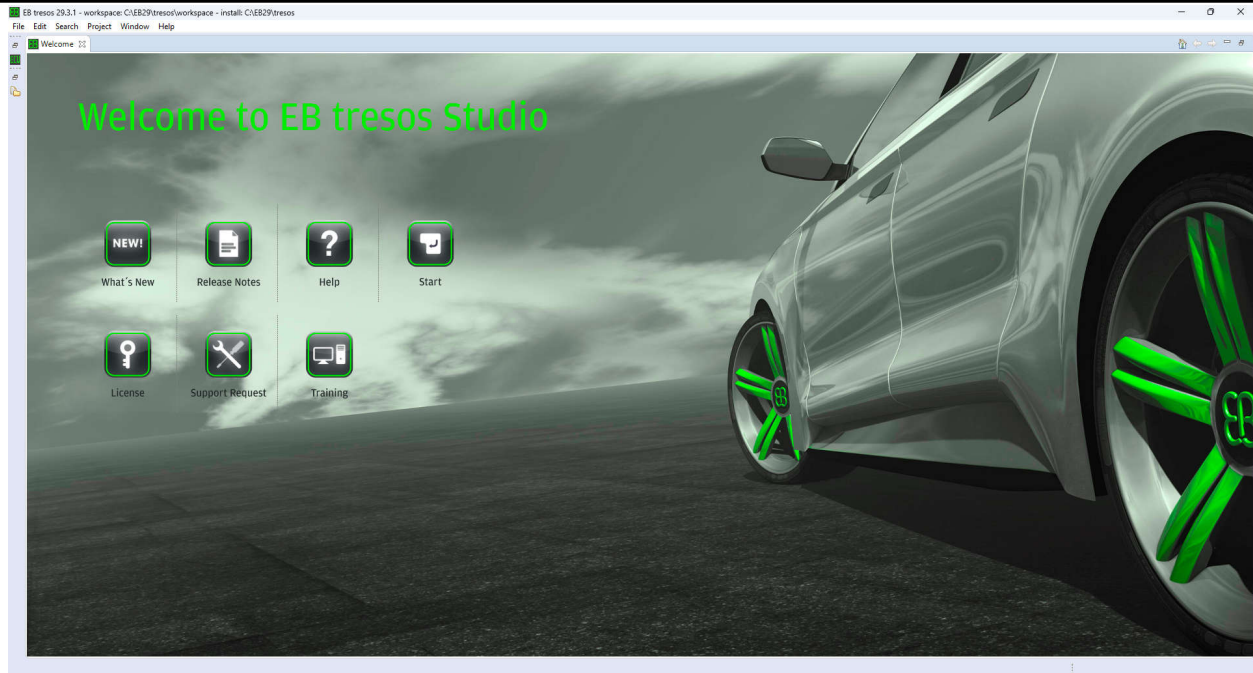
- `config/`
 - `<Module>.xdm` - XML Data Format 文件，它可以使用 EB tresos 支持的 XPath 和表达式类型验证。
 - `<Module>.arxml` - 标准 AUTOSAR XML 文件，可在任何符合 AUTOSAR 的配置程序工具中加载插件架构。
- `config_ext/`
 - `<Module>_Rec.xdm` - 建议的 XDM Data 文件，其可以为 XDM 中的许多可用选项加载建议的字段值。
- `generate/`
 - `include/`
 - `<Module>_Cfg.h` - Module Configuration Header 文件生成模板。
 - `src/`
 - `<Module>_Cfg.c` - Pre-compile 配置变量的 Module Configuration Source 文件生成模板。
 - `<Module>_Lcfg.c` - Link-time 配置变量的 Module Configuration Source 文件生成模板。
 - `<Module>_Pbcfg.c` - Post-Build 配置变量的 Module Configuration Source 文件生成模板。
 - `V<Number>` - 专用于某一版本的文件夹，其中每个版本与驱动程序硬件变量相关。具体包含与硬件变量完全匹配的任何硬件绑定标头文件和源文件。
- `generate_swcd/`
 - `swcd/`
 - `<Module>_Bswmd.arxml` - *plug-in* 的 BSW Module Description 文件。
- `META-INF/`
 - `MANIFEST.MF` - 用于描述插件的 Java Manifest 文件。
 - `CRYPTOMANIFEST.MF` 和 `CRYPTOMANIFESTSIG.MF` - 已签名的 *plug-in* 文件，能对插件的每个文件（这些文件除外）进行签名，并包含与 User License 一起使用的加密插件源信息。

- `plugin.xml` - 适用于多种插件特定设置的 Plug-in XML 文件。

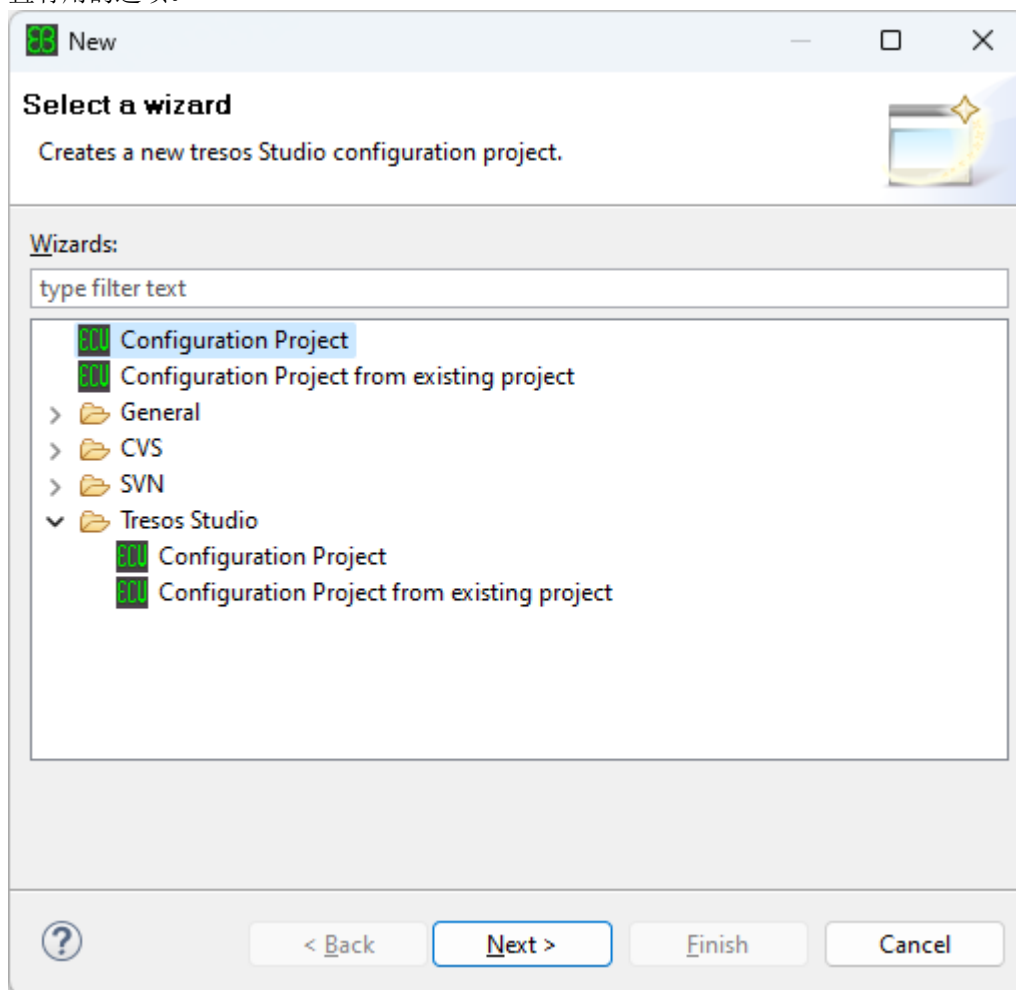
要使用配置程序插件，请执行以下步骤：

1. 在 EB tresos 安装路径中的 `plugins/` 文件夹下粘贴要加载的插件文件夹，通常是 Microsoft® Windows® 系统的 `C:/EB/tresos/`。通过 MCAL 软件文件中所述的机制从 TI 获取特定器件的插件。请勿在复制插件时运行 EB tresos，否则可能无法识别软件，并且必须关闭并重新启动 EB tresos。
2. 启动 EB tresos Studio 应用程序：

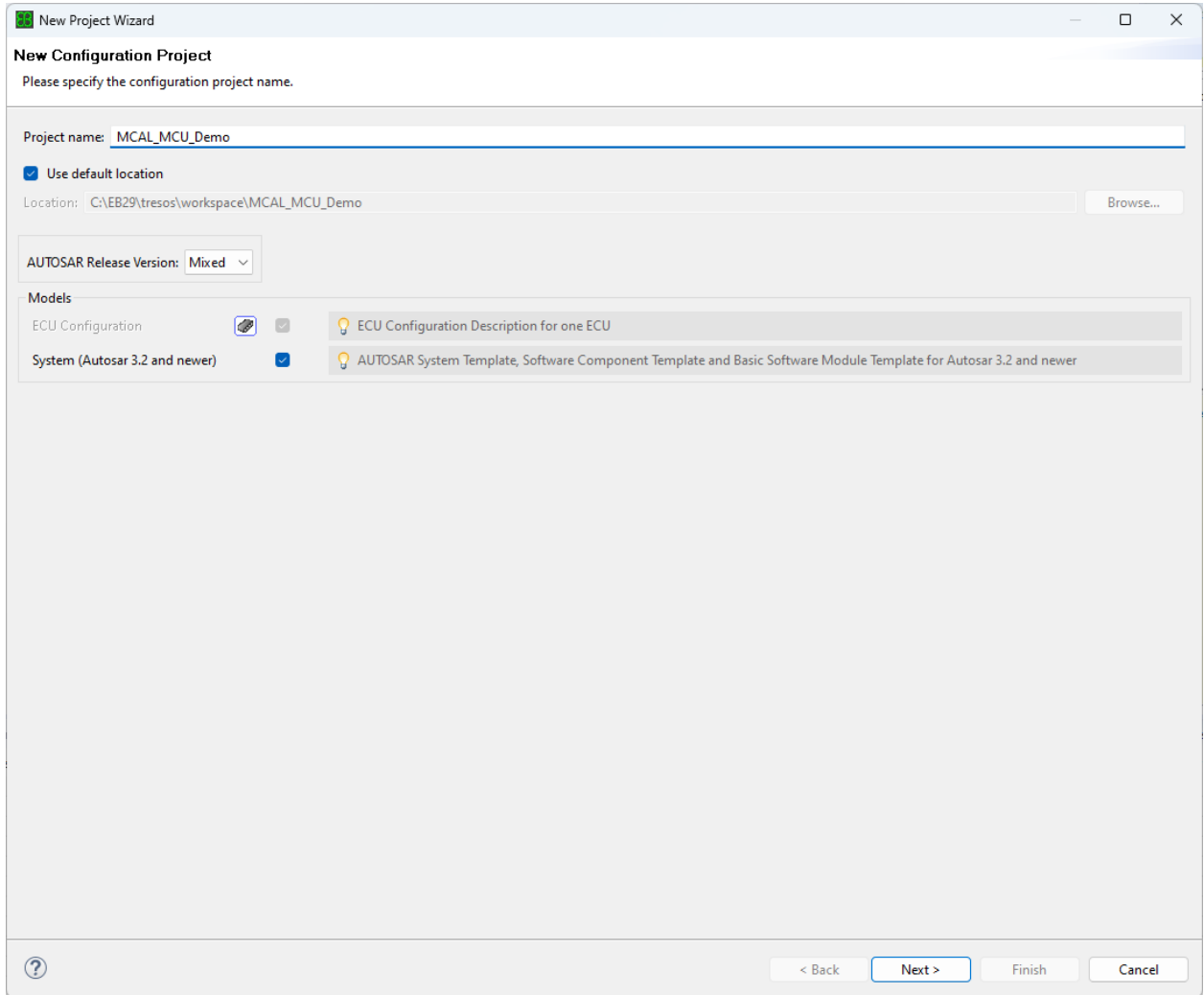




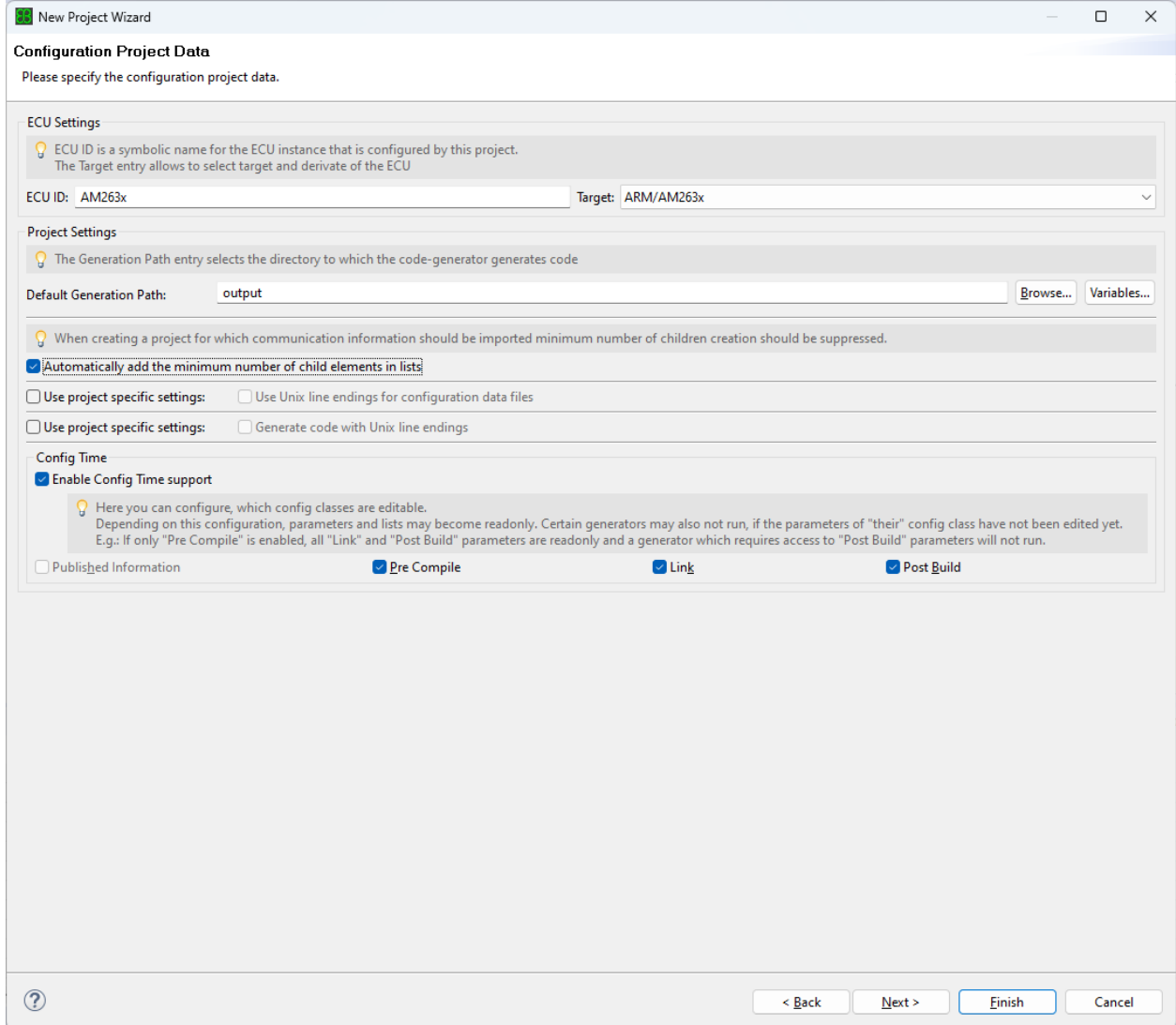
3. 通过 **File** → **New** → **Configuration Project** 创建一个新的 **Configuration Project**。在 **New Project Wizard** 中设置有用的选项。



a. 使用任何适当的项目名称。



- b. 提供相关的 ECU Id (用户选择) , 并将 **Target** 插件正确设置为与 AM2x MCU 系列相匹配的 ARM/AM2xx。一旦看到 **Target** 下拉菜单中填充了器件插件, 则表示 EB tresos 已正确识别该插件。

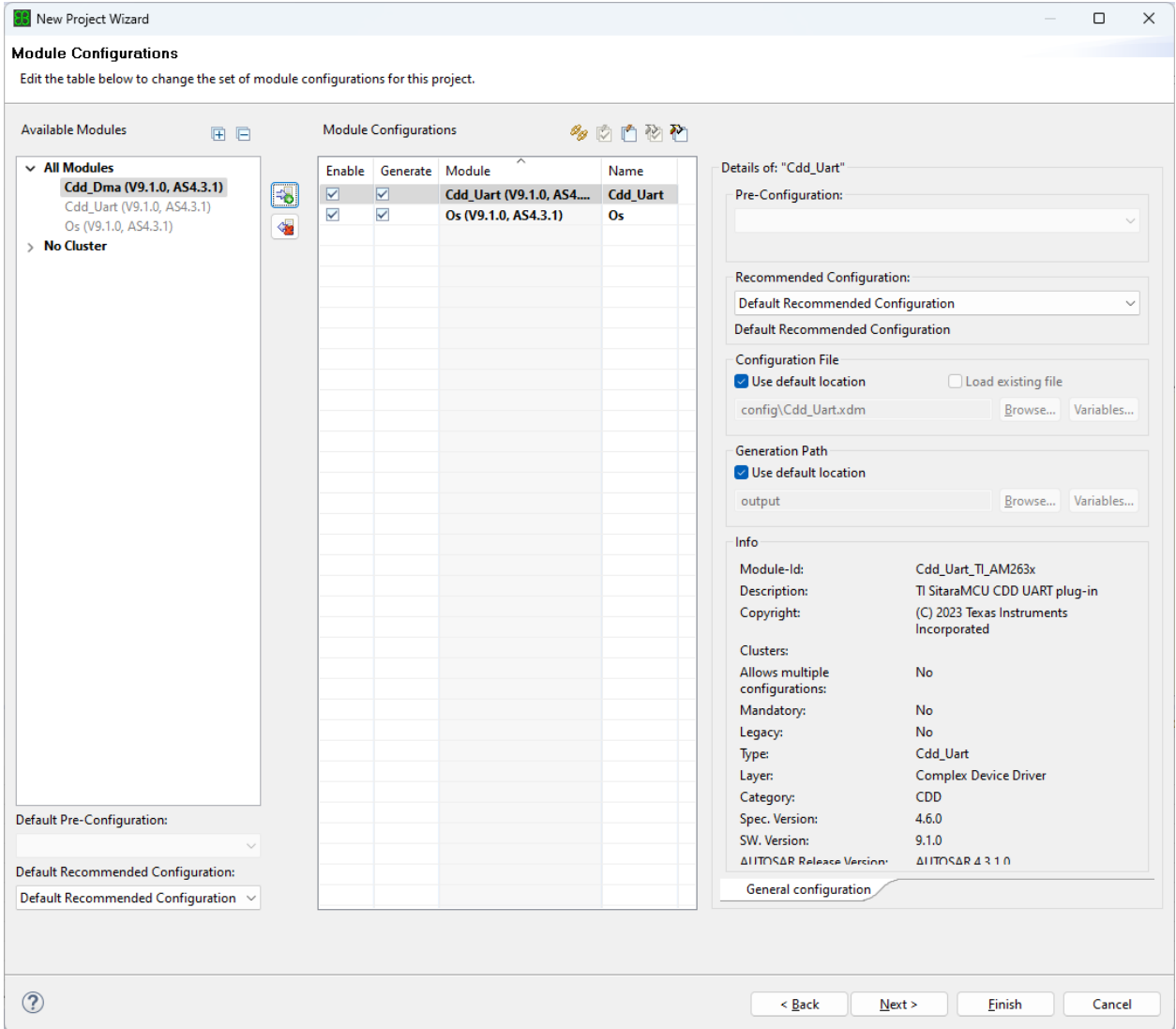


The screenshot shows the "New Project Wizard" dialog box, specifically the "Configuration Project Data" step. The dialog is titled "New Project Wizard" and "Configuration Project Data". It contains the following sections and options:

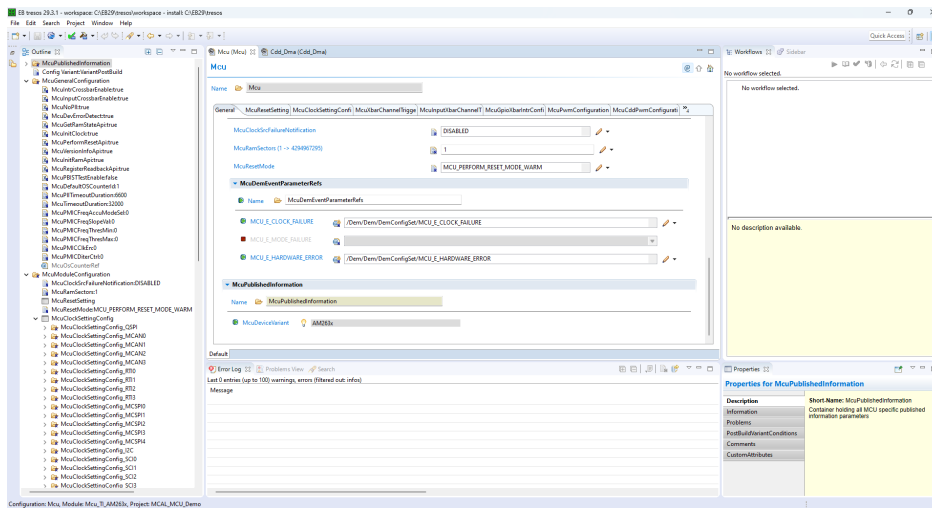
- ECU Settings:** A text box for "ECU ID" is set to "AM263x". A dropdown menu for "Target" is set to "ARM/AM263x". A tooltip explains: "ECU ID is a symbolic name for the ECU instance that is configured by this project. The Target entry allows to select target and derivate of the ECU".
- Project Settings:** A text box for "Default Generation Path" is set to "output". There are "Browse..." and "Variables..." buttons. A tooltip explains: "The Generation Path entry selects the directory to which the code-generator generates code".
- Options:**
 - Automatically add the minimum number of child elements in lists
 - Use project specific settings: Use Unix line endings for configuration data files
 - Use project specific settings: Generate code with Unix line endings
- Config Time:**
 - Enable Config Time support
 - Here you can configure, which config classes are editable. Depending on this configuration, parameters and lists may become readonly. Certain generators may also not run, if the parameters of "their" config class have not been edited yet. E.g.: If only "Pre Compile" is enabled, all "Link" and "Post Build" parameters are readonly and a generator which requires access to "Post Build" parameters will not run.
 - Published Information
 - Pre Compile
 - Link
 - Post Build

At the bottom, there are navigation buttons: "< Back", "Next >", "Finish", and "Cancel". A help icon (?) is also present.

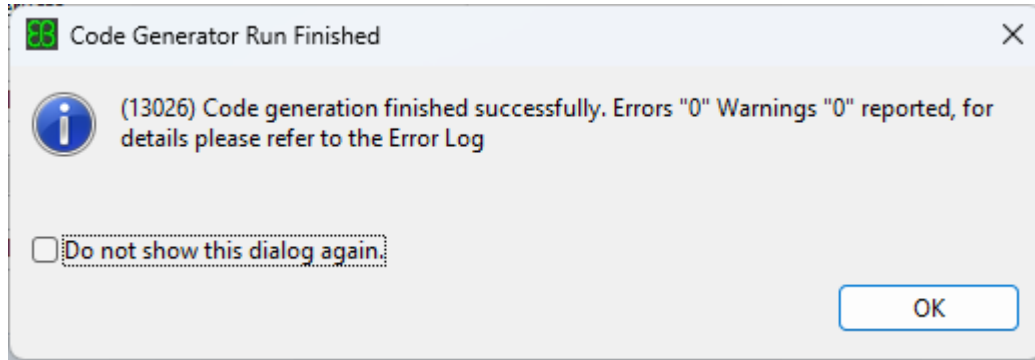
- c. 从可用的集合中添加正确的模块集。如果必须使用任何 *Importers* 或 *Exporters*，点击 *Next* 按钮；否则，直接点击 *Finish* 按钮。



4. 打开每个模块的通用编辑器，并找到可用于设置和更新配置的选项字段。



- 在准备好数值后，在应用程序 *Toolbar* 的 *Context* 菜单或从 *Application Menu Bar*，点击 *Verify* 项目，然后，若无错误，则开始 *Build* 项目。



- 展开在该项目下生成的输出文件夹、然后复制生成的 `include/<Mip>_Cfg.h` 和 `src/<Mip>_Cfg.c` 基于配置变量的配置源文件，以与任何驱动程序的最最终应用一起使用。

在 `<EB_tresos_Installation_Path>/doc/` 文件夹可获取有关安装和使用 EB tresos Studio 的文件。有关使用此工具的更多详细信息，另请参阅 `EB/tresos/doc/2.0_EB_tresos_Studio/2.1_Studio_documentation_users_guide.pdf` 上的 Users Guide。

2.2 源文件

源文件包括驱动程序实现和相关标头文件。其中包括多个文件、这些文件分为面向驱动程序用户的接口和更接近特定目标硬件 MCU 的私有源以及中断源。

特定驱动程序的标头 (.h) 和源 (.c) 文件以 `Pascal_Snake_Case` 中的模块实现前缀 `<Mip>` 开头 (即 `PascalCase` 和 `snake_case` 的组合) 命名，以符合 AUTOSAR 命名和目录结构规范。`<Mip>` 命名文件是具有公共接口内容 (包括可从设备驱动程序中使用的宏、数据类型和 API) 的文件。该源保持通用状态，并符合模块 SRS 和 SWS 规范。针对该层预期的任何给定功能，API 都具有一个固定的且定义明确的接口，从而使 API 可以可靠地与源库一同使用。API 能向后兼容；因此，无需清理或更改版本迁移。

不建议在驱动程序应用中使用名为 `<Mip>_Priv` (包含专用实现) 且硬件高度特定的源。只要迁移硬件设计，就可以更新 `<Mip>_Priv` 标识符和符号，并且由于有任何特性支持和错误修复功能，在现有源实现后不需要提供向后兼容性。

名为 `<Mip>_Irq` 的文件包含 IRQ 和中断处理源，该源用于为中断例程 (ISR) 提供服务。采用 TI 矢量中断模块 (VIM) 实现 ARM 嵌套矢量中断控制器，可设置中断例程。除 `<Mip>` 命名文件以外的任何文件均被视为专用驱动程序源，建议不要将其内容直接视为可从构建驱动程序库使用的接口，即使文件或符号仍然暴露给源或捆绑库的任何部分也是如此。

在通过对软件包中的任何模块进行初始化来设置依赖项时，示例应用程序源可用作参考，甚至可以作为基于驱动程序自定义程序的启动应用程序。示例源可用作示例演示，并保留任何驱动程序库没有应用程序依赖项的情况，但为示例生成的配置文件标头除外，在不使用各种不同的生成式配置标头时，可使用此类配置文件标头构建驱动程序源库。

2.3 用户指南

MCAL 用户文件以根目录 (即 MCAL 源路径下) 中的 *发行说明* 的形式提供。在 Sitara™ MCU 中, 其他用户文件以 PDF 文件 (适用于 v08.06.02 之前的旧 MCAL 版本) 或 HTML 格式网页 (适用于 v09 及更高版本、[AM263 用户指南](#)或 [AM273 用户指南](#)) 的形式提供。用户指南包含有关源特性的详细信息, 这些源特性涵盖每个驱动器提供的可配置参数和 API。

另请参阅 [MCAL 用户指南](#), 了解最新的 Sitara™ MPU 和 Jacinto™ 器件。

2.4 合规性支持包 (CSP)

合规性支持包 (CSP) 是德州仪器 (TI) 提供的一组文件报告, 旨在认证 TI 软件组件符合安全标准。这些报告包括适用于基线质量 (BQ) < 汽车质量 (AQ) < 功能安全质量 (FSQ) 的不同文件集。在汽车中使用, MCAL 的目标是 FSQ。目前, 鉴于在 FuSa 上使用时既未涵盖也未声明其他文件 (例如示例应用程序和 BSW 存根), 因此, 仅在包括配置程序插件在内的驱动程序源中考虑合规性。

甚至会让具有功能安全专业知识的外部组织评估 CSP 文件, 以按照相关安全标准 (例如 IEC61508 和 ISO26262:2018) 进行所需的 SIL 和汽车 SIL (ASIL) 级合规性认证。在进行 CSP 审核之后, 该组织随后可以根据公司的质量阈值审核标准对 TI 器件进行认证。CSP 不是一个通用术语, 而是专门针对 TI 的一系列安全合规性鉴定文件。

在本应用手册发布日之前, TÜV SÜD 授予的、适用于 AM263x Sitara MCU 的 MCAL v09.01.00 和适用于 J7 Jacinto 器件系列的 MCUSW v03.00.00 (基于处理器 SDK RTOS v08.06.00) 均进行了 ASIL-D/SIL-3 认证, 其中, MCAL v09.00.00.02 适用于 AM62x 和 AM62Ax Sitara MPU, 而 v09.01.00 适用于 AM273x Sitara MCU 器件系列。[MCAL 认证文件](#)可从 [TI.com](#) 中获取。

3 使用源-构建驱动程序和示例

驱动程序源可以构建为库, 相关的示例应用程序二进制文件可以构建为可执行的 .xer5f 格式和可擦写的 .appimage 格式。要构建其中的任何一个, makefile 都包含能轻松构建的捆绑包中。安装 Make 客户端 (Microsoft® Windows® 上的 GMake) 作为构建源库和示例的先决条件。如果没有 make, 则必须在命令提示符下以相同的顺序使用 make 命令下给出的所有步骤以构建可执行输出文件。

3.1 一次性设置

构建 MCAL 源的系统中必须安装 [Code Composer Studio™ 集成式开发环境 \(IDE\)](#) 和 [TI Arm® 代码生成工具 \(CGT\) - 编译器 Build Dependency](#) 工具。确保 TI MCAL 软件包下 build/Rules.make 文件中的这些工具路径正确。编辑这些工具目录和其他派生应用程序的相关路径。使用原始 Rules.make 内容中正在使用的确切依赖项版本。也可以从用户指南和发行说明中交叉查看建议的版本, 以确保这些版本与收到的捆绑包版本兼容。

3.2 构建说明

如要构建驱动程序，请按照以下步骤操作：

1. 打开终端并导航到 **MCAL** 源目录和 **build/** 文件夹。
2. 使用适当的构建目标来构建所需的库或可执行二进制文件 (**.xer5f**) / 用户应用程序 (**.appimage**)。

- a. Sitara MCU 的构建命令语法为：

```
make [-s] <build_target> [PLATFORM=<soc_name>] [PROFILE=<release/debug>] ...
```

在该命令中，方括号[]中显示的选项是可选的，在构建时无需提供，除非要从设置默认值进行更改。

- i. **-s** 标志用于抑制其他 **make** 命令日志以简化 **make** 输出。
 - ii. **平台**是要构建的目标器件系列。例如，在 **AM273x Sitara™ MCU** 系列中，**平台**选项可以从 **am273x** 和 **am2732** 中取值
 - iii. **配置文件**能设置版本优化级别。调试版是一种未优化的版本，可以在执行期间轻松分步执行和跟随。为了能围绕已编译源的时序和存储器范围进行高效优化而构建了发行版。
- b. Sitara MPU 和 Jacinto 器件的构建命令语法为：

```
make [-s] <build_target> [BOARD=<device_evm_name>] [SOC=<soc_name>]
[SDK_INSTALL_PATH=<sdk_install_path>] [PROFILE=<release/debug>] ...
```

在该命令中，方括号[]中显示的选项是可选的，在构建时无需提供，除非在 **build/Rules.make** 文件中从设置默认值进行更改。

- i. **-s** 标志用于抑制其他 **make** 命令日志以简化 **make** 输出。
 - ii. **BOARD** 和 **SOC** 是目标器件系列。例如，在 **AM6x Sitara™ MPU** 系列中，**<soc_name>** 选项可以从 **am62x**、**am62ax** 或 **am62px** 中取值。在 **Jacinto** 系列中，**<soc_name>** 选项可以从有效的 **j7xxx** 中取值。
 - iii. **SDK_INSTALL_PATH** 是 **SOC** 系列相应 **MCU+ SDK RTOS** 的安装路径，在安装位置会显示 **SOC** 系列。
 - iv. **配置文件**提供了版本优化级别。**debug** 版是一种未优化的版本，可以在执行期间轻松分步执行和跟随。为了能围绕已编译源的时序和存储器范围进行高效优化而构建了 **release** 版。
3. 在构建完成后，在软件包中的 **binary/** 目录下（在 **build** 文件夹的正上方）找到生成的文件。驱动程序库为 **.aer5f** 格式，示例应用程序是 **.xer5f** 可执行和可擦写的 **.appimage** 文件。

若要执行某一项构建，构建目标名称是 **make build** 命令中唯一所需的项目。可以通过传递属性 **NAME = <value>** 来更新其他可用值；否则，将在构建中使用相应软件包的默认值。构建命令仅适用于独立 **MCAL** 软件包。如果 **MCAL** 是任何一个供应商交付的整个 **AUTOSAR** 堆栈的一部分，而每个供应商都使用唯一的编译设置，则这些命令是不同的。

有关构建相关的详细信息和其他信息，请参阅用户指南内容。

3.3 构建命令语法

make 语法的示例如下所示：

- J7、Sitara MPU：

```
make [-s] <build_target> [BOARD=<device_evm_name>] [SOC=<soc_name>]
[SDK_INSTALL_PATH=<sdk_install_path>] [PROFILE=<release/debug>] ...
```

- Sitara MCU：

```
make [-s] <build_target> [PLATFORM=<soc_name>] [PROFILE=<release/debug>] ...
```

图 3-1 展示了一个 Microsoft Windows 11 示例，其中 `gmake -s adc_app` 命令会构建 ADC 应用程序以及来自 `gmake` 依赖项的驱动程序源。

```

Windows PowerShell
PS C:\MCAL_AM263x_09.02.00\build> gmake -s adc_app
# Making am263:r5f0_0:release:adc...
# Compiling am263:r5f0_0:release:adc: src/Adc.c
# Compiling am263:r5f0_0:release:adc: V0/Adc_Priv.c
# Compiling am263:r5f0_0:release:adc: V0/Adc_Utils.c
# Compiling am263:r5f0_0:release:adc: V0/Adc_Platform.c
# Compiling am263:r5f0_0:release:adc: V0/Adc_Irq.c
#
# Archiving am263:r5f0_0:release:adc
#
# Making am263:r5f0_0:release:dma...
# Compiling am263:r5f0_0:release:dma: src/Cdd_Dma.c
# Compiling am263:r5f0_0:release:dma: src/Cdd_Dma_Irq.c
# Compiling am263:r5f0_0:release:dma: src/Cdd_Dma_Priv.c
# Compiling am263:r5f0_0:release:dma: V0/Cdd_Dma_soc.c
#
# Archiving am263:r5f0_0:release:dma
#
# Making am263:r5f0_0:release:mcu...
# Compiling am263:r5f0_0:release:mcu: src/Mcu.c
# Compiling am263:r5f0_0:release:mcu: V0/Mcu_Priv.c
#
# Archiving am263:r5f0_0:release:mcu
#
# Making am263:r5f0_0:release:port...
# Compiling am263:r5f0_0:release:port: src/Port.c
# Compiling am263:r5f0_0:release:port: V0/Port_Priv.c
# Compiling am263:r5f0_0:release:port: V0/Port_Irq.c
#
# Archiving am263:r5f0_0:release:port
#
# Making am263:r5f0_0:release:app_utils...
# C:/MCAL_AM263x_09.02.00/mcal/binary/examples/Utils/obj/am263/r5f/release/sys_core.oer5f C:/MCAL_AM263x_09.02.00/mcal/binary/examples/Utils/obj/am263/r5f/release/boot_armv7r_asm.oer5f C:/MCAL_AM263x_09.02.00/mcal/binary/examples/Utils/obj/am263/r5f/release/sys_pmu_asm.oer5f Compiling am263:r5f0_0:release:app_utils: ./CLANG/sys_core.asm
# C:/MCAL_AM263x_09.02.00/mcal/binary/examples/Utils/obj/am263/r5f/release/sys_core.oer5f C:/MCAL_AM263x_09.02.00/mcal/binary/examples/Utils/obj/am263/r5f/release/boot_armv7r_asm.oer5f C:/MCAL_AM263x_09.02.00/mcal/binary/examples/Utils/obj/am263/r5f/release/sys_pmu_asm.oer5f Compiling am263:r5f0_0:release:app_utils: ./CLANG/am263/boot_armv7r_asm.asm
# C:/MCAL_AM263x_09.02.00/mcal/binary/examples/Utils/obj/am263/r5f/release/sys_core.oer5f C:/MCAL_AM263x_09.02.00/mcal/binary/examples/Utils/obj/am263/r5f/release/MpuP_armv7r_asm.oer5f C:/MCAL_AM263x_09.02.00/mcal/binary/examples/Utils/obj/am263/r5f/release/sys_pmu_asm.oer5f Compiling am263:r5f0_0:release:app_utils: ./CLANG/am263/MpuP_armv7r_asm.asm
# C:/MCAL_AM263x_09.02.00/mcal/binary/examples/Utils/obj/am263/r5f/release/sys_core.oer5f C:/MCAL_AM263x_09.02.00/mcal/binary/examples/Utils/obj/am263/r5f/release/MpuP_armv7r_asm.oer5f C:/MCAL_AM263x_09.02.00/mcal/binary/examples/Utils/obj/am263/r5f/release/sys_pmu_asm.oer5f Compiling am263:r5f0_0:release:app_utils: ./CLANG/sys_pmu_asm.asm
# Compiling am263:r5f0_0:release:app_utils: ./am263/app_utils.c
# Compiling am263:r5f0_0:release:app_utils: ./am263/app_utils_uart.c
# Compiling am263:r5f0_0:release:app_utils: ./am263/sci.c
# Compiling am263:r5f0_0:release:app_utils: trace.c
# Compiling am263:r5f0_0:release:app_utils: esm.c
# Compiling am263:r5f0_0:release:app_utils: sys_vim.c
# Compiling am263:r5f0_0:release:app_utils: crc.c
# Compiling am263:r5f0_0:release:app_utils: ./am263/0s.c
# Compiling am263:r5f0_0:release:app_utils: ./CLANG/CacheP.c
# Compiling am263:r5f0_0:release:app_utils: Dem.c
# Compiling am263:r5f0_0:release:app_utils: ./CLANG/am263/boot_armv7r.c
# Compiling am263:r5f0_0:release:app_utils: ./CLANG/am263/MpuP_armv7r.c
# Compiling am263:r5f0_0:release:app_utils: sys_pmu.c
# Compiling am263:r5f0_0:release:app_utils: app_utilsEth.c
#
# Archiving am263:r5f0_0:release:app_utils
#
# C:/MCAL_AM263x_09.02.00/mcal/binary/adc_app/obj/am263/r5f0_0/release/sys_intvecs.oer5f C:/MCAL_AM263x_09.02.00/mcal/binary/adc_app/obj/am263/r5f0_0/release/sys_startup.oer5f Compiling am263:r5f0_0:release:adc_app: ./Utils/CLANG/sys_intvecs.asm
# C:/MCAL_AM263x_09.02.00/mcal/binary/adc_app/obj/am263/r5f0_0/release/sys_intvecs.oer5f C:/MCAL_AM263x_09.02.00/mcal/binary/adc_app/obj/am263/r5f0_0/release/sys_startup.oer5f Compiling am263:r5f0_0:release:adc_app: ./Utils/CLANG/am263/sys_startup.asm
# Compiling am263:r5f0_0:release:adc_app: AdcApp.c
# Compiling am263:r5f0_0:release:adc_app: soc/am263/r5f0_0/AdcApp_Startup.c
# Compiling am263:r5f0_0:release:adc_app: C:/MCAL_AM263x_09.02.00/mcal/examples_config/Adc_Demo_Cfg/soc/am263/r5f0_0/src/Adc_PBCfg.c
# Compiling am263:r5f0_0:release:adc_app: C:/MCAL_AM263x_09.02.00/mcal/examples_config/Adc_Demo_Cfg/soc/am263/r5f0_0/src/Adc_Cfg.c
# Compiling am263:r5f0_0:release:adc_app: C:/MCAL_AM263x_09.02.00/mcal/examples_config/Port_Demo_Cfg/soc/am263/r5f0_0/src/Port_PBCfg.c
# Compiling am263:r5f0_0:release:adc_app: C:/MCAL_AM263x_09.02.00/mcal/examples_config/Mcu_Demo_Cfg/soc/am263/r5f0_0/src/Mcu_PBCfg.c
# Compiling am263:r5f0_0:release:adc_app: C:/MCAL_AM263x_09.02.00/mcal/examples_config/Mcu_Demo_Cfg/soc/am263/r5f0_0/src/Mcu_Cfg.c
# Compiling am263:r5f0_0:release:adc_app: C:/MCAL_AM263x_09.02.00/mcal/examples_config/Dma_Demo_Cfg/soc/am263/r5f0_0/src/Cdd_Dma_Cfg.c
# Compiling am263:r5f0_0:release:adc_app: C:/MCAL_AM263x_09.02.00/mcal/examples/Utils/Def.c
# Linking into C:/MCAL_AM263x_09.02.00/mcal/binary/adc_app/bin/am263/adc_app_r5f0_0_release.xer5f...
#
# C:/MCAL_AM263x_09.02.00/mcal/binary/adc_app/bin/am263/adc_app_r5f0_0_release.xer5f created.
#
# Starting Boot image generation: adc_app_r5f0_0_release.appimage ...
# Multicore Boot image generation for: adc_app_r5f0_0_release.appimage Done !!!
  
```

图 3-1. Make 的输出示例

4 获取 MCAL 软件包

德州仪器 (TI) 交付在下述两种器件上可用的 MCAL 软件包：来自多家顶级完整栈供应商（例如 Vector 和 ETAS）的 AUTOSAR 堆栈捆绑包以及独立的 MCAL + CDD 捆绑包。可以让离您最近的客户支持人员和现场工程师提出申请要求使用独立的 MCAL + CDD 捆绑包，他们可以在 TI 官方网站上分享最新 MCAL 版本的 TI 许可证导出表。示例包括：[MCAL AM263x v10.00.00](#)、[MCAL AM263Px v10.00.00](#) 和 [MCAL AM273x v09.01.00](#)。同样，可以让离您最近的客户支持人员或现场工程师提出申请要求使用独立的 MCAL + CDD 捆绑包或者采用 *软件导出* 申请表申请，以将其用于 Sitara MPU 器件和 Jacinto 7 输入器件中。示例包括：[MCAL AM62x v10.00.00](#)、[MCU+SDK AM62x v10.00.00](#) 和 [处理器 SDK RTOS \(MCU+ SDK\) J722S v10.00.00](#)。MCAL 版本与器件 MCU+SDK 发行版本相匹配。剩余 Jacinto 器件的 MCAL 可在 TI 网站 [ti.com](#) 上公开获取，具体位置是 [MCUSW 文件夹](#)，且其是 [RTOS SDK](#) 的一部分（例如：[J721E RTOS SDK](#)），此外，必须单独申请驱动程序的配置程序插件包。

在申请 MCAL 软件包时，也可能会申请 EB tresos Studio 用户许可证以使用 TI 合作伙伴的配置程序工具中的配置程序插件。

MCAL 在 AM263x 和 AM273x Sitara MCU 上启动，2023 年初发布了 v08.06 版本，然后发布了多个版本。v09.02.00 的最新版本是在 2024 年 5 月发布。MCAL 之前已适用于 Sitara MPU 和 Jacinto 系列器件，最新版本为 v10.00。这些发行版本还受制于早期 AUTOSAR 标准版本简介部分所述的语义版本控制，且需要符合 TI 官方网站上为器件提供的器件 SDK ([MCU-PLUS-SDK](#)) 时间轴发行版本。

计划对适用于各种产品的 MCAL 提供许多其他发行版本和长期支持，包括适用于 Jacinto、Sitara MPU、Sitara MCU、C2000、Hercules MCU、雷达、MSP 和任何其他此类 TI 器件的 MCAL 软件包。

5 总结

德州仪器 (TI) 公司提供了 MCAL 软件包、以便在符合功能安全标准的汽车级嵌入式处理器（从 Jacinto 处理器到 Sitara 微处理器和微控制器单元）上实现 AUTOSAR 堆栈。MCAL 软件包提供了一个常规 SDK，其中包含简单的器件驱动程序，可通过软件 API 启用硬件接口。MCAL 软件包还能按照 AUTOSAR 标准对 MCAL 和 CDD 驱动器提出的要求，将驱动程序接口作为标准组件 API 进行维护，同时轻松访问符合 ISO 26262:2018 汽车功能安全标准的抗风险系统。可以请求在 TI 的芯片上独立使用软件包，也可以从整个堆栈内的任何 TI 主要 AUTOSAR 堆栈供应商合作伙伴处获得该软件包。

6 参考资料

1. 德州仪器 (TI)，[平台 MCAL 开发 - AM263 用户指南](#)
2. 德州仪器 (TI)，[平台 MCAL 开发 - AM273 用户指南](#)
3. 德州仪器 (TI)，[Sitara MPU 和 Jacinto 器件用户指南](#)
4. 德州仪器 (TI)，[Jacinto AUTOSAR MCAL 驱动程序证书](#)
5. 德州仪器 (TI)，[\[常见问题解答\] Sitara AM263x/AM273x MCAL - 入门指南 | TI E2E™ 论坛](#)
6. 德州仪器 (TI)，[\[常见问题解答\] 如何获取 MCAL 开发所需的 EB Tresos Studio 和客户许可证？ | TI E2E™ 论坛](#)
7. 德州仪器 (TI)，[功能安全 | TI.com](#)
8. 德州仪器 (TI)，[AM2x 和 Hercules™ 微控制器产品功能安全概述](#)
9. 德州仪器 (TI)，[AutoSAR MCAL — Hercules 安全 MCU 资源指南](#)

重要通知和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
版权所有 © 2025，德州仪器 (TI) 公司