

基于AUTOSAR的车载域控制器应用软件开发方法

韩衍东 高泽天 于钊 郭源科 刘兴 吴骄阳

(中国第一汽车股份有限公司研发总院, 长春 130013)

【欢迎引用】韩衍东, 高泽天, 于钊, 等. 基于AUTOSAR的车载域控制器应用软件开发方法[J]. 汽车文摘, 2025(2): 1-5.

【Cite this paper】HAN Y D, GAO Z T, YU Z, et al. Domain Controller Software Development Method Based on AUTOSAR for Vehicular Application[J]. Automotive Digest (Chinese), 2025(2): 1-5.

【摘要】AUTOSAR逐渐成为各整车厂和零部件供应商的主流开发架构并应用于车载域控制器应用软件开发中,但其开发流程还不够完善,缺少通用、全面且高效的应用软件开发方法。为了解决该问题,基于AUTOSAR的基本理论,阐述了AUTOSAR的系统架构及相关接口,提出了从ARXML文件提取数据字典内容、生成接口模型、控制模型搭建、模型代码生成和应用代码释放的全部过程。所提出的车载域控制器应用软件开发方法,每个环节均采用成熟开发工具或编写自动化脚本实现,保证方法通用性的同时,可以保证软件的开发质量,提升软件开发效率,从而降低开发成本。

关键词: AUTOSAR; 汽车电子; 域控制器; 应用软件开发

中图分类号: U469.72 文献标志码: A DOI: 10.19822/j.cnki.1671-6329.20230141

Domain Controller Software Development Method Based on AUTOSAR for Vehicular Application

Han Yandong, Gao Zetian, Yu Zhao, Guo Yuanke, Liu Xing, Wu Jiaoyang
(Global R&D Center, China FAW Corporation Limited, Changchun 130013)

【Abstract】With the development of automotive electronics industry, AUTOSAR has gradually become the mainstream development architecture of Oems and parts suppliers, and is used in the development of vehicle domain controller application software. However, the development process of applying AUTOSAR to vehicle domain controller application software is not perfect, and there is a lack of general, comprehensive and efficient application software development method. To solve this problem, based on the basic theory of AUTOSAR, the system architecture and related interfaces of AUTOSAR are expounded, and the whole process of extracting data dictionary content from ARXML file, generating interface model, building control model, generating model code and releasing application code is proposed. The proposed development method of vehicle domain controller application software adopts mature development tools or automated scripts to realize each link, which ensures the universality of the method, can ensure the quality of software development, improve the efficiency of software development, and thus reduce the development cost.

Key words: AUTOSAR, Automotive electronic, Domain controller, Application software development

0 引言

随着汽车电子产业的持续发展,汽车电控系统的复杂性日益增加。传统嵌入式软件开发方法不支持硬件抽象,导致软件代码量增加、软件重用性降低。同时汽车电子电气架构逐渐从分布式架构向域集中式和中央计算平台方向发展。为了解决汽车控制器软件注入开发周期长、可重用性差的问题^[1],多个汽车制造商联合制定了汽车电控系统基础软件标准,即汽

车开放系统架构(Automotive Open System Architecture, AUTOSAR)。AUTOSAR将汽车系统的基础软件标准化为一个跨整车企业(Original Equipment Manufacturer, OEM)的“标准栈”,通过定义标准接口实现模块化软件在不用硬件平台上的移植,在缩短开发周期的同时提高软件质量。AUTOSAR的主要目标之一是改善产品及开发过程,遏制复杂性和风险,增强集成和传递功能的可扩展性和灵活性^[2]。因此,越来越多汽车整车企业以及零部件供应商采用AUTOSAR进行应用软件开发,基于

AUTOSAR的架构规范已成为行业标准。此外,随着车载功能增多及汽车电子化程度的提升,电子控制单元(Electronic Control Unit, ECU)数量随之增加。大量ECU集成导致整车网络拓扑结构越来越复杂,增加了整车线束布置难度,确保ECU准确无误的协调工作也成为一项技术挑战。域控制器的引入使这一问题得到解决,其将多个功能相近的传统ECU集中到一个算力更高、资源更强的域控制器中。采用域控制器进行开发,可显著减少整车ECU和线束数量,降低整车网络拓扑的复杂性,同时增强通信速率。

目前,已有学者基于AUTOSAR标准进行车载软件开发研究。刘玺斌等^[3]结合AUTOSAR工具,提出了基于AUTOSAR规范的汽车ECU软件开发方法,但该方法仅介绍了应用层软件源代码链接和编译生成目标可执行文件过程,并未阐述应用层软件的开发方法。张丽萍等^[4]以汽车大灯控制模块软件应用层开发举例,描述了一种以MATLAB/Simulink为起点、符合AUTOSAR开发标准的软件应用层开发流程,主要介绍了模型搭建及代码生成的过程。冯江波等^[5]介绍了MATLAB与AUTOSAR标准相兼容的自动代码生成过程,同时介绍了AUTOSAR支持的3种接口类型,但只介绍了代码生成相关内容。潘志前^[6]基于AUTOSAR架构,介绍了电动汽车整车控制器(Vehicle Control Unit, VCU)的软件功能设计方法,以及VCU软件集成和测试方法。虽然目前基于MATLAB与AUTOSAR之间的映射方法和符合AUTOSAR规范的自动代码生成方法均有了一定程度的研究和应用。然而,采用AUTOSAR架构开发应用软件层起步较晚,缺少通用、全面且高效的应用软件开发方法。

文中提出一种基于AUTOSAR的车载域控制器应用软件开发方法,该方法提供了从ARXML文件提取数据字典内容、生成接口模型、控制模型搭建、模型代码生成和应用代码释放的全部过程,在保证软件开发质量的同时提升开发效率。

1 AUTOSAR架构与接口

1.1 系统架构

如图1所示,AUTOSAR的系统架构将运行在微控制器上的软件分为3层^[7],分别为应用软件层(Application Software Layer, ASL)、运行环境层(RunTime Environment, RTE)和基础软件层(Basic SoftWare, BSW)。其中,BSW包括服务层(Service layer)、ECU抽象层(ECU abstraction layer)、微控制器抽象层(Microcon-

troller abstraction layer)和复杂驱动(Complex driver)^[8]。

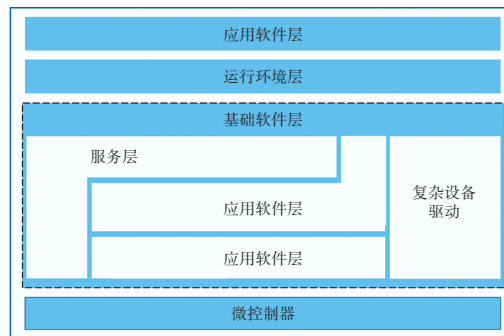


图1 AUTOSAR系统架构

ASL由多个软件组件(SoftWare Component, SWC)组成,由可运行实体(Runnable)进行SWC通信接口配置。其中,可运行实体的触发条件包括定时时间、数据接收事件、异步服务调用返回事件、操作调用事件、数据接收错误事件以及数据发送完成事件等。

RTE位于ASL和BSW之间,是虚拟功能总线接口的实现^[9]。主要负责SWC之间以及SWC与BSW之间的通信。每个ECU的RTE配置因其软件组件具体需求的不同而不同。同时,RTE层还可以防止软件组件直接访问基础软件。在RTE层中,若采用发送者/接受者(Send/Receiver)通信机制,实现的是缓存访问模式。在进入可运行实体之前RTE为数据建立副本,可运行实体在运行过程中,仅对副本进行操作,实际数据不发生改变。在可运行实体运行结束后,RTE将副本中的数据复制至实际数据地址。

在BSW中,微控制器抽象层的目的是实现上层软件与微处理器型号解耦,包含微控制器(MicroController Unit, MCU)中内部外设的驱动和MCU内存映射的外部设备的驱动。ECU抽象层的目的是使上层软件与ECU硬件设计解耦,包含ECU板上外部设备的驱动,以及内部设备与外部设备的接口。服务层包括系统服务、存储器服务和通信服务,目的是为应用层提供可用的服务接口。复杂设备驱动主要提供复杂传感器和执行器的驱动。

1.2 AUTOSAR接口

AUTOSAR的接口共分为3种,分别为标准接口、AUTOSAR接口和标准化的AUTOSAR接口。其中,标准接口和标准化的AUTOSAR接口定义不可更改。如图2所示,在AUTOSAR架构中,ASL与RTE之间采用AUTOSAR接口,服务层与RTE之间采用标准化的AUTOSAR接口,其余采用标准接口。

在数据处理机制中,数据的读写操作可分为显式(Explicit)和隐式(Implicit)2种模式。在执行Runnable

期间,显性接口的数据收发操作由 RTE 直接触发,隐性接口的数据收发不会在 Runnable 执行期间改变数据状态,写入数据将在 Runnable 执行结束后发出。

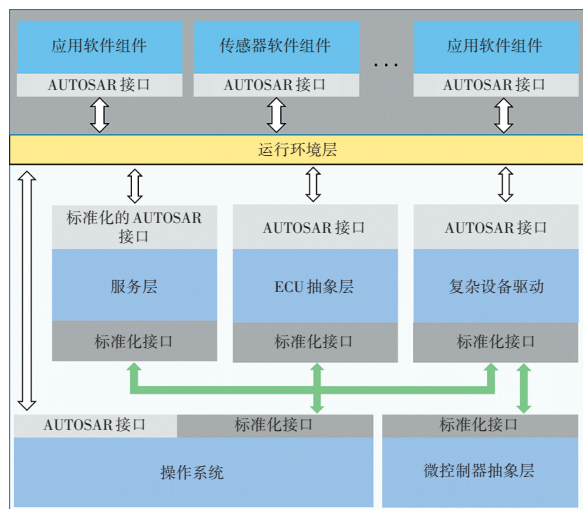


图2 AUTOSAR架构访问接口

2 应用软件开发流程

基于 AUTOSAR 的车载应用软件开发流程如图 3 所示,应用软件完整开发步骤为:(1)将初始 ARXML (AUTOSAR XML)文件导入 AUTOSAR 开发工具,以执行符合性校验。一旦校验通过,将进行配置工作,包括添加初始化触发模块和 Runnable 配置 ARXML。根据项目需求,选择并配置所需的输入和输出接口,同时确定其显式或隐式特征。(2)将配置完成的 ARXML 文件导入信息提取工具,提取出数据字典和类型定义文件。由于 MATLAB/Simulink 是车载应用软件常用的模型开发工具,同时具有 ARXML 文件信息提取及校验功能,因此在该步骤中可作为 AUTOSAR 符合性校验的数据信息提取工具。(3)创建或更新接口模型。判断模型是否为初版,若是初版,则采用 Create 命令创建接口模型,否则采用 update 命令更新接口模型,并进行校验。(4)搭建控制模型,并进行 AUTOSAR 校验。校验通过后,通过 AUTOSAR 选项下的代码生成 (Generate code) 选项进行模型代码生成,并根据运行错误信息修改模型,直至成功生成代码。(5)编写自动化批处理脚本,自动提取应用软件编译对应的代码文件。(6)将生成的应用层软件代码文件与基础软件代码进行编译,若编译过程中报错,则需要根据报错信息,修改模型及代码,直至代码文件编译成功。(7)将编译成功的代码文件释放至下游部门,进行域控制器或中央计算平台程序编译开发。

以整车控制应用层软件开发为例。首先,基于步

骤(1)对整车控制应用层软件所需的各 SWC 初始 ARXML 文件进行校验并完成 Runnable 配置,导出配置完毕的 ARXML。其次,基于步骤(2)提取出每个 SWC 所对应的数据字典和类型定义文件。基于步骤(3)创建或更新应用软件接口模型,搭建应用软件模型,添加项目整车控制所需的功能模块,如高压上下电模块、扭矩控制模块或者换挡处理模块等,并进行模块间的信号传输连线配置,配置完毕后运行模型,直至模型运行通过。基于步骤(4)校验模型并生成代码。最后,将所有的 SWC 均完成代码生成后,基于步骤(5)~(7)进行代码编译和释放。

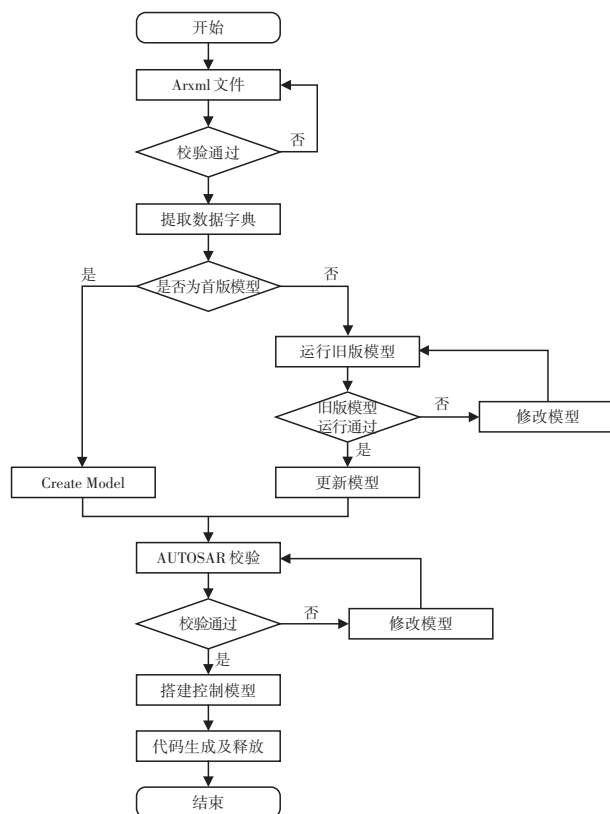


图3 应用软件开发整体流程

上述实例为动力域整车控制应用软件开发流程,还可依据此开发流程进行空调控制应用软件开发、车身域应用软件开发等,该方法适用于所有基于 CP AUTOSAR 开发的车载域控制器应用软件,具有通用性和全面性。同时,由于采用了统一的标准,该方法显著提升了软件开发效率和应用软件的复用性,有效缩短了开发周期,提高了开发效率。

3 接口模型创建及更新

3.1 ARXML 文件校验及配置

在进行应用层软件开发之前,需从上游的整车制造商或研发部门获取符合 AUTOSAR 规范的初始

ARXML文件。以下为在AUTOSAR工具中进行文件处理的配置和步骤:(1)查看ARXML文件属性,解除只读和加密等属性,将初始ARXML文件导入AUTOSAR工具(配置ARXML,进行ARXML内部接口连接及显隐性接口定义等)。(2)对文件进行Validation校验,判断其是否符合AUTOSAR的规范性要求,若不符合要求则通知上游部门修改。(3)根据软件组件SWC具体功能需求添加Runnable模块,如图4所示。需要在初始ARXML文件中添加初始化(Init)模块和周期性可运行实体模块,并配置Runnable属性及与接口的关联关系。配置收发接口时需要根据项目需求选择显隐式。(4)完成配置后再次进行校验,校验通过后导出ARXML文件用于后续开发。

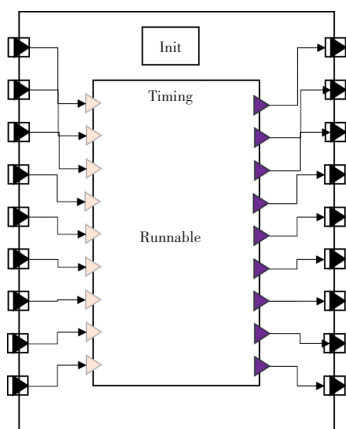


图4 ARXML文件配置Runnable

3.2 数据字典内容提取

数据字典是重要的信息存储工具,其存储了Simulink模型中的接口定义信息和数据信息。在进行代码生成时,Simulink模型与数据字典一般并列存在。因此在代码生成前需从ARXML中提取数据字典信息。

将配置完成的ARXML文件导入MATLAB,提取出运行软件实体、接口和应用软件组件的相关信息,获得所需数据字典mat文件和m文件。其中,m文件为导入ARXML文件过程中自动生成的文件,mat文件是将MATLAB的工作区(Workspace)中自动加载的内容另存为mat文件得到。数据字典mat文件中包含了模型中各信号的定义,如信号的基本类型(Base type)、信号的描述以及进行代码生成时的头文件等。关于ARXML文件导入MATLAB方法详见4.1中Step1描述。

3.3 接口模型生成

在进行接口模型生成之前,首先需判断其是否为首版软件。若为首次建立接口模型,则需要将ARXML文件导入模型开发工具,生成全新的顶层接口模型。具体在MATLAB软件中实现步骤如下。

(1)在ARXML所在目录文件下,在命令行输入如下命令,创建顶层接口模型,其中artest为对应ARXML文件名称:

```
{
Ar = arxml.importer('artest.arxml');
createComponentAsModel(Ar, '/SoftwareTypes/ComponentTypes/artest', 'ModelPeriodicRunnablesAs', 'FunctionCallSubsystem')
}
```

(2)校验模型。在模型界面中,选择APPS(Application Service)目录下的AUTOSAR组件设计器(Component designer),在代码映射(Code mappings)中选择验证代码映射(Validate Code mappings)进行校验,验证AUTOSAR的组件配置。校验结果为成功时可继续进行软件开发,如失败则需要更改ARXML文件后再次执行该过程,直至校验成功后,保存新生成的模型。

若需要基于已有模型进行ARXML文件更新,则在MATLAB中接口模型更新的实现方式为:(1)加载上一版本模型对应的数据字典,打开上版模型并运行,模型运行通过后,对模型进行更;(2)在命令行窗口输入如下语句,更新模型:

```
{%将ARXML文件放在对应的工程目录文件下%
Ar = arxml.importer('artest.arxml');
updateModel(Ar, 'artest');
}
```

(3)更新完成后,根据Simulink生成的更新报告,删除其中需手动处理的接口,并保存模型。

(4)重新加载新的数据字典内容,并按照前文所述步骤,对新生成模型进行AUTOSAR校验,保证接口模型符合AUTOSAR规范要求。

4 控制模型搭建及代码释放

4.1 控制模型搭建及代码生成

根据软件需求,基于MATLAB/Simulink搭建控制模型,包括输入输出接口连线及功能模型搭建,如图5所示。以部分控制模型Simulink模块示例,模型的顶层主要由输入信号、输出信号和主控模块3部分组成。

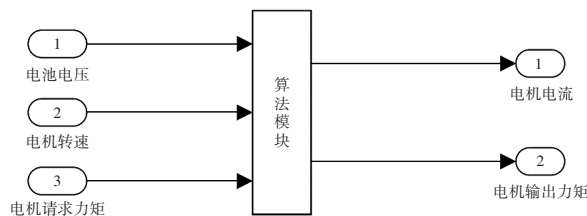


图5 Simulink控制模型示例

为保证生成的代码能够与基础软件匹配并进行编译,且最后生成的可执行文件可以在控制器中正常运行,需在生成代码前对模型生成代码属性进行配置。

(1)调整顶层接口模型的模型设置(Model settings)属性,在Code generation中的系统目标文件(System target file)选项下选择autosar.tlc文件。

(2)修改代码生成模板(Code templates)和数据生成模板(Data template)中的源文件模版(Source file template)配置,确保生成的代码和数据模板满足基础软件对应用软件代码及常量内存分配的需求。

(3)编写控制器代码配置文件,内部包含配置代码文件和数据文件的伪指令,表示生成代码的组件。

(4)配置Model Settings中的其他代码生成相关性,主要包括运行步长、代码生成文件以及警告及错误配置,将配置结果输出为配置参考(Configuration reference)文件,并将其保存至Workspace中。

(5)在顶层接口模型中打开模型浏览器(Model explorer)选项,将属性配置(Configurations)调整为引用(Reference)。并通过传播到引用模型(Propagate to referenced models)选项推送至每个Reference控制模型。

完成搭建控制模型及其配置后,对模型进行编码。在控制模型界面的APPS标签下,选择嵌入式(Embedded)模块,点击“Generate code”进行代码生成。在代码生成过程中,若遇到错误,应根据报错信息对模型进行相应的修正,直至成功生成代码。

4.2 应用代码释放

在模型生成代码的过程中,将生成一些非必要信息,如模型在运行过程中生成的过程文件、接口声明头文件以及一些仿真文件,需在生成的文件中筛选出编译所需的程序文件。人工提取程序效率低,且易发生操作错误。为了高效进行代码编译,本研究采取了一种自动化的方法来提高代码编译的效率。该方法可以实现编写自动化脚本,自动提取编译所需的程序文件,提高工作效率的同时保证软件质量。

批处理(bat)文件是Windows系统内置的脚本,其无需编译即可执行,可以快速、敏捷且灵活的处理文件。无需安装其他软件及环境,在文件处理提取方面存在明显优势。因此本研究选用编写bat脚本的方式进行编译文件提取,并与基础软件一同进行编译,编译成功后将生成的文件释放至下游部门,用于控制器可执行程序文件的生成。具体步骤如下:清除提取文件夹

中的历史文件,删除上一次代码生成时得到的程序文件。提取本次新生成且编译所需的程序文件至提取文件夹中。识别各SWC生成的同名文件,并根据文件内容进行合并或删除。将应用软件与基础软件进行编译,若编译报错,则检查日志(log)文件分析报错原因,并根据分析结果对代码进行相应修改,然后重新执行编译。编译成功后将应用程序代码释放至下游部门,完成应用软件开发。

5 结束语

目前,国内虽然在MATLAB与AUTOSAR间的映射方法和符合AUTOSAR规范的自动代码生成方法方面有了一定的进展,但仍缺少通用、全面且高效的应用软件开发方法。本文在已有研究内容基础上,提供了从ARXML文件提取数据字典内容、生成接口模型、控制模型搭建、模型代码生成和应用代码释放的全过程,在保证软件开发质量的同时提升了开发效率。虽然本文提出的基于AUTOSAR的车载域控制器应用软件开发方法,降低了车载应用软件开发复杂度,缩短了开发时间,提高软件质量,具有通用、全面和高效性,但自动化程度不够高,如何提高自动化程度以便后续实现持续集成工作有待进一步研究。

参考文献

- [1] 李超超,武恪等.基于AUTOSAR的CAN通信栈设计[J].电子测量技术,2021(12):139-145.
- [2] MIRHEIDARI S, FALLAHI A, et al. AUTOSAR Model-Based Software Component Integration of Supplier Software[J]. SAE International, 2015(8): 544-548.
- [3] 刘玺斌,马建,宋青松.基于AUTOSAR规范的汽车ECU软件开发方法[J].长安大学学报,2013(3):76-87.
- [4] 张丽萍,国云飞.基于MATLAB的AUTOSAR自动代码生成技术[J].研究与开发,2016(9):36-40.
- [5] 冯江波,刘亚军.与AUTOSAR兼容的Matlab/Simulink自动代码生成技术[J].佳木斯大学学报,2011(6):833-837.
- [6] 潘志前.基于AUTOSAR架构的电动汽车VCU软件功能设计[J].Defense Manufacturing Technology, 2019(3):36-39.
- [7] 崔淑梅,张玉琦,杜博超,等.一种基于AUTOSAR的电机控制器软件架构设计[J].微特电机,2022(6):1-9.
- [8] 詹克旭.基于AUTOSAR架构的汽车电子软件产品的开发方法[J].汽车电器,2022(11):73-76.
- [9] 张泽莹,颜江,王慧,等.提高AUTOSAR模型开发效率的方法[J].汽车电器,2023(4):35-37.

(责任编辑 梵玲)