

ICS 27.020

CCS J 92

# 团 体 标 准

T/CSICE 033.2-2025

## 柴油发动机电控单元 第2部分：应用软件开发

Diesel engine electronic control system—Part 2: Application software development

2025-12-26 发布

2025-12-26 实施

中国内燃机学会 发布



## 目 次

前言 .....	II
引言 .....	III
1 范围 .....	1
2 规范性引用文件 .....	1
3 术语和定义 .....	1
4 符号和缩略语 .....	2
5 技术指标 .....	2
6 开发过程 .....	2
6.1 应用软件需求分析 .....	2
6.1.1 基本要求 .....	2
6.1.2 技术要求 .....	2
6.1.3 实施步骤 .....	2
6.2 应用软件架构设计 .....	2
6.2.1 技术要求 .....	2
6.2.2 实施步骤 .....	3
6.3 应用软件单元开发流程 .....	3
6.3.1 开发流程 .....	3
6.3.2 设计与实现步骤 .....	3
6.3.3 测试实施步骤 .....	4

## 前　　言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

T/CSICE 033-2025《柴油发动机电控单元》已经或计划发布7个部分。

- 第1部分：需求开发。
- 第2部分：应用软件开发。
- 第3部分：基础软件开发。
- 第4部分：硬件设计。
- 第5部分：软件集成。
- 第6部分：软件测试。
- 第7部分：硬件验证。

本文件为T/CSICE 033-2025的第2部分。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由国内燃机学会标准管理部提出。

本文件由国内燃机学会归口。

本文件起草单位：潍柴动力股份有限公司、南岳电控（衡阳）工业技术股份有限公司、天津大学、潍柴新能源商用车有限公司、山东大学、潍柴雷沃（潍坊）农业装备有限公司。

本文件主要起草人：高登峰、刘兴义、常秀书、王加林、孟卫东、宋国梁、陈韬、孙波、余浪、白书战、王健、杨涛、吕宪勇。

本文件于2025年首次发布。

## 引　　言

T/CSICE 033-2025《柴油发动机电控单元》是指导柴油发动机电控单元如何自上而下进行正向开发。它涉及柴油发动机电控单元开发的原则、流程、设计、集成和验证的行业标准，拟由7个部分组成。

- 第1部分：需求开发。
- 第2部分：应用软件开发。
- 第3部分：基础软件开发。
- 第4部分：硬件设计。
- 第5部分：软件集成。
- 第6部分：软件测试。
- 第7部分：硬件验证。

T/CSICE 033-2025的第1部分规定了柴油发动机电控单元需求开发的术语和定义、一般要求和需求开发过程要求。第2部分规定了柴油发动机电控系统中基于模型的应用软件开发流程的要求。第3部分规定了柴油发动机电控单元中基础软件的技术指标及开发过程。第4部分规定了柴油发动机电控单元中硬件需求说明、架构设计、元器件及芯片选型、电路原理图设计和印刷电路板（PCB）设计需要遵循的基本指导方法和规则。第5部分规定了柴油发动机电控单元中软件集成的技术要求及实施步骤。第6部分规定了柴油发动机电控单元系统测试过程的要求。第7部分规定了柴油发动机电控单元硬件验证过程中试验条件、试验工况、试验方法及判定依据的要求。

# 柴油发动机电控单元 第2部分：应用软件开发

## 1 范围

本文件规定了柴油发动机电控单元中基于模型的应用软件开发流程的要求。

本文件适用于柴油发动机电控单元的应用软件开发，包括功能安全相关内容和非功能安全相关内容，其他电控单元可参照本文件执行。

## 2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

ISO 26262-1:2018 道路车辆 功能安全 第1部分：术语 (Road vehicles—Functional safety—Part 1: Vocabulary)

T/CSICE 033.1-2025 柴油发动机电控单元 第1部分：需求开发

## 3 术语和定义

下列术语和定义适用于本文件。

### 3.1

**应用软件 application software**

实现电控系统控制策略的软件集合。

### 3.2

**浮点模型 floating point model**

利用图形化建模软件搭建的变量和模块的数据类型都采取默认数据类型的模型。

### 3.3

**定标 define data types**

指定义模块中变量的数据类型的行为过程。

### 3.4

**测试用例 test case**

模型测试中，结合发动机的实际工况或者为满足模型测试覆盖度，而给出的相应的输入信号和输出信号。

## 4 符号和缩略语

下列缩略语适用于本文件。

MIL: 模型在环测试 (model in the loop)

SIL: 软件在环测试 (software in the loop)

PIL: 编译在环测试 (processor in the loop)

## 5 技术指标

基于模型的应用软件开发的一般要求如下：

- a) 变量命名应符合编程规范, 具备一定可读性;
- b) 通过应用软件搭建应具备层次性、模块化、标准接口, 实现应用软件的可移植性;
- c) 应用软件建模应具备软件版本记录。

## 6 开发过程

### 6.1 应用软件需求分析

#### 6.1.1 基本要求

应用软件需求应满足T/CSICE 033. 1-2025要求。

#### 6.1.2 技术要求

应用软件需求应实现条目化, 每条需求应包含需求编号、需求名称、需求描述、需求优先级及验证方式。需求描述应具备完整性、正确性、可行性、必要性及可验证性, 具体包括以下几方面。

- a) 功能定义。明确该需求实现的具体功能。
- b) 约束。根据需求的具体内容, 确认实现该需求要遵循的约束条件。
- c) 外部接口。软件模块与其他软件模块之前的接口, 包括接口的用途、格式等。
- d) 异常处理要求。故障或异常情况下的软件处理策略。
- e) 性能要求, 如控制精度、动态响应性等。

#### 6.1.3 实施步骤

应用软件需求分析按照以下步骤进行:

- a) 根据系统需求分析与应用软件层面的需求, 完成应用软件需求规格书;
- b) 从需求的完备性、准确性、可行性、必要性及可验证性等方面对应用软件需求规格书进行评审并出具评审报告。

### 6.2 应用软件架构设计

#### 6.2.1 技术要求

应用软件架构设计是应用软件开发的重要环节, 应满足以下技术要求:

- a) 应用软件架构应满足应用软件需求, 需编写详尽的应用软件架构规格说明书;
- b) 应用软件架构设计应确定应用软件模块之间的结构关系和数据流, 清晰地定义系统的组件及组件之间的关系;

c) 应用软件架构设计应可以灵活调整以适应需求的变化。

## 6.2.2 实施步骤

应用软件架构设计应按照以下步骤进行：

- 根据应用软件需求规格书, 遵照 6.2.1 中的技术要求进行应用软件架构设计, 完成应用软件架构设计说明书;
- 对应用软件架构设计说明书进行评审并出具评审报告。

## 6.3 应用软件单元开发流程

### 6.3.1 开发流程

应用软件单元开发流程应按照图1步骤进行。



### 6.3.2 设计与实现步骤

应用软件单元设计与实现实施步骤具体如下。

- 模型搭建。通过可视化建模工具, 根据应用软件需求, 完成模型搭建。模型搭建过程中, 应选择简单、高效的建模方法。模型搭建应遵循从左到右, 自上而下的原则, 应力求美观、清晰、易读。
- 模型定标。为浮点数据类型的变量选择合适的数值精度及合适的数值范围。合适的数值精度是指数值的最小变化值所代表的物理意义要满足控制系统的需求, 需要分析变量的精细程度、传感器测试精度、执行器的控制精度等方面来确定。合适的数值范围是指变量定标后能表示的最大值和最小值符合实际物理范围。
- 模型需求关联。通过需求开发管理工具, 建立的需求和模型之间应可双向跟踪。
- 定浮点对比。模型定标完成后, 需要对测量定点数据类型模型和浮点数据类型模型的输出结果进行对比, 校验模型定标是否满足要求。根据应用需求设置可接受的计算偏差。
- 建模规范性检查。即模型静态检查, 对搭建好的模型进行规范性检查, 及时修改检查出的问题, 建模规范性检查可结合企业自身情况购买成熟的商业查核软件, 或根据开发经验自定义开发具有针对性的查核项。
- 变动对比。针对应用软件模型修改前、后的模型进行对比, 任何修改内容都应在报告中明确指出, 开发人员需要确认报告中的修改前、后对比内容是否正确。
- 编制设计说明文档。模型搭建并测试完成后, 编制对应的应用软件设计说明文档。
- 有效性查核。在进行模型功能等测试之前, 需要确认软件开发交付物的完整性及改动的正确

性。查核确认时，需在有效性核查单中勾选相应的检查项，核查结果将记录在最终的应用软件验证检查单中。

### 6.3.3 测试实施步骤

应用软件测试实施步骤具体如下。

a) 测试用例设计。在进行模型功能测试时，应给出明确的输入、执行条件和预期结果，用于验证特定功能是否符合要求。测试用例设计应对应条目化的需求，每条需求应至少对应一条测试用例。测试用例应覆盖正常、异常、边界场景，确保每条测试用例独立且可执行。

b) 测试用例评审。通过针对测试用例与需求的对应关系以及测试用例与需求条目的对应性等方面进行评审，保证测试用例对需求的覆盖程度（测试覆盖度）。

c) 单元 MIL 测试。该项为必做项。测试用例的设计可根据改动范围或是否为新建模型，选择针对整个模型进行测试或针对改动涉及的子系统进行测试。模型输出与预期结果误差应在允许的范围内。决策覆盖度应达到 100%，条件覆盖度应达到 90% 以上。每个测试用例需关联需求文档的具体条目。

d) 单元 SIL 测试。该项为选做项。将测试模型生成代码在 PC 中运行，相同代码生成配置和输入激励下，记录输出结果，与 MIL 的运行结果进行对比。通过代码运行结果与模型仿真运行结果对比，从而判断生成代码后的逻辑与模型逻辑效果的一致性。当生成代码执行结果与模型仿真结果偏差超过合理范围，则应反馈开发人员修改逻辑后重新进行测试。

e) 单元 PIL 测试。该项为选做项。将测试模型生成代码在单片机中运行，将实际控制器中的代码运行结果与模型仿真运行结果对比，从而判断二者的一致性。在相同的代码生成配置、编译环境下，基于实际项目相同的单片机进行仿真，从而评估编译环境有效性，同时，单元 PIL 测试也要关注所测试的逻辑的运行时间及负荷和代码空间占用情况，判断是否有代码优化的空间。当发现生成代码执行结果与模型仿真结果偏差超过合理范围，或发现代码空间占用超限或运行负荷超限，则应反馈开发人员修改优化逻辑后重新进行测试。

f) 多模型检查。每次版本升级前，需要做多模型检查，进行模型和模型之间、模型和硬件底层接口之间、模型、诊断管理系统等相关配置列表信息之间的数据信息交互检查。保证开发内容修改无问题，应用软件开发内容可顺利集成。

T/CSICE 033. 2-2025