**版次： A**

**密级：** **内部** **代号：** **18S2.480.00318XS**

**CJSJ1006**

**发动机电子控制器软件设计说明**

**共 52 页**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CJSJ1006 | | | 发动机电子控制器软件设计说明 | | | | | 18S2.480.00318XS | | |
| 文 件 签 署 | | | | | | | | | | |
| 批准： | | | | | |  | | | | |
| 签 署 | | 日 期 | | 签 署 | | 日 期 | 签署 | | | 日期 |
| 编写 |  |  | | 会签 |  |  | 会签 | |  |  |
| 校对 |  |  | |  |  |  |  | |  |  |
| 审核 |  |  | |  |  |  |  | |  |  |
| 标审 |  |  | |  |  |  |  | |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 发 送 单 位 | | | |
| 单 位 | 单 位 | 单 位 | 单 位 |
| 十八室 | 东风技术中心 |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**文档更改记录**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 版次 | 更改单号 | 更改内容 | 更改日期 | 更改人 | 备注 |
| 1 | A |  | 新建 |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

目 次

1 范围 1

1.1 标识 1

1.2 系统概述 1

1.3 文档概述 4

2 引用文档 4

2.1 国家军用标准 5

2.2 所级标准 5

2.3 项目文件 5

3 CSCI级设计决策 5

4 CSCI体系结构设计 5

5 CSCI详细设计 7

5.1 PWM输出驱动 7

5.1.1 e200z4端业务逻辑 7

5.1.2 eTPU端业务逻辑 10

5.1.3 应用软件接口 13

5.2 发动机转速输出驱动 13

5.2.1 e200z4端业务逻辑 14

5.2.2 eTPU端业务逻辑 14

5.2.3 应用软件接口 14

5.3 频率量输入驱动 14

5.3.1 e200z4端业务逻辑 14

5.3.2 eTPU端业务逻辑 16

5.3.3 应用软件接口 18

5.4 燃油喷射驱动 18

5.4.1 e200z4端业务逻辑 18

5.4.2 eTPU业务逻辑 21

5.4.3 应用软件接口 28

5.5 曲轴凸轮采集与同步驱动 29

5.5.1 e200z4端业务逻辑 29

5.5.2 eTPU业务逻辑 34

5.5.3 应用软件接口 50

6 需求可追踪性 52

7 注释 52

1. 范围
   1. 标识

本文档适用的软件系统开发代号和名称为：18S2.480.00318 发动机电子控制器软件

软件缩略名为：ECU\_CDD

本软件系统适用的系统名称为：发动机电子控制器（以下简称ECU）

文档代号：18S2.480.00318XS

文档名称：发动机电子控制器软件设计说明

文档版本：A

* 1. 系统概述
     1. 系统用途

ECU系统作为高压共轨柴油发动机燃料供给系统的组成部分，负责通过各种传感器采集发动机在不同工况下的运行信息，执行控制解算，对发动机在不同车速、负荷、道路坡度、气压和温度等条件下的喷油脉宽、喷油时刻及轨压进行精确控制，使车辆达到最优的动力性、经济性和排放要求。

ECU系统功能由ECU平台及其基础上的上层应用程序协同工作、配合实现。ECU平台负责提供并管理全部硬件设备资源的功能及工作状态，并向上层应用程序提供访问各种资源的对应接口。如图1所示，ECU平台的处理器MPC5644A内部包含两个处理单元，作为主处理器的e200z4处理单元及作为协处理器的eTPU处理单元，处理单元之间通过共享存储器（以下简称SCM）进行数据交互，分别从各个外部传感器获得信息，经过处理及通信，控制各个外部作动器的动作，实现系统功能。



图1　ECU的处理单元构成

* + 1. 软件用途

ECU平台作为智能电子装置，其软件系统由bootloader软件、基础平台软件、复杂驱动软件及应用软件构成。各个组件通过集成协作，在不同层面对控制器内部的各种硬件资源进行管理和驱动，实现ECU系统功能。

各个软件模块在系统中的位置见图2。



图2　系统包图

* + 1. 一般特性

本软件系统作为车载设备所属的嵌入式实时软件，具备以下一般特性：

1. 实时性：指本系统对于外界激励的响应必须满足时间约束的特性。其正确性不仅由系统的功能和行为特性决定，还依赖于系统的时间特性。
2. 嵌入式：指本系统内置于一个更大的上层系统之中，作为上层系统不可分割的一部分存在。嵌入式软件专为特定系统设计，与硬件紧密结合，共同完成系统功能，软件脱离特定系统无法运行。同时，本软件系统作为嵌入式软件，其运行环境（目标环境）与开发环境（宿主环境）存在差异，在软件测试时需着重考虑其造成的影响。
3. 高可靠性和安全性：本软件系统作为车载软件，其运行异常可能会带来灾难性的后果，因此其对可靠性和安全性的要求较高。
   * 1. 开发历史

2017年11月1日，作为ECU项目的子系统，开始开发本软件系统的首个版本。

* + 1. 相关机构

东风汽车集团股份有限公司技术中心作为本项目的需方，对本项目的开发过程和交付产品负有批准责任。

东风汽车集团股份有限公司技术中心作为本软件系统的软件用户及开发方，承担平台软件组件及应用软件组件全生命周期内的开发及维护活动，负责软件系统的整体集成及验证活动，并使用集成后的完整软件系统提供的功能。

航空工业计算所十八研究室硬件组作为本软件系统所属的硬件平台的开发方，负责软件开发及验证平台的组建活动，确定及维护软硬件接口，配合软件开发方的软件验证及确认活动。

航空工业计算所十八研究室软件组作为本软件系统中复杂驱动软件组件及bootloader软件组件的开发方，响应需方的要求，策划和执行项目开发过程，向需方交付软件产品，配合需方执行软件验证及确认工作。

航空工业计算所十八研究室软件组作为本软件系统中复杂驱动软件组件及bootloader软件组件的保障机构，在需方对交付的软件产品做出认可后，向联合开发方及软件用户提供复杂驱动软件组件及bootloader软件组件的技术支持和更新维护服务。

* + 1. 运行现场

本软件系统作为嵌入式软件，其开发过程中的运行环境和装机后的运行环境存在差异。

1. 开发过程中的运行环境

本软件系统在开发过程中的运行环境如图3所示。软件代码在宿主机中的CodeWarrior Development Studio软件集成开发环境中编译完毕后，通过编程器下载到目标机的处理器中运行。



图3　开发过程中的运行环境

1. 装机后的运行环境

本软件系统在装机之后的运行环境如图4所示。编译生成的可执行文件烧录在MPC5644A处理器的片内FLASH存储器中，上电后处理器自动跳转到相应地址执行。



图4　装机后的运行环境

* 1. 文档概述

本文档依据GJB 438B-2009《军用软件开发文档通用要求》进行编制，参考Q/18S 611A-2012《嵌入式软件设计标准》的要求，包含了体系结构设计与详细设计两部分。体系结构设计描述了本CSCI的整体结构设计、模块功能划分及接口函数定义、数据定义及流向；详细设计描述了各模块函数的具体实现。本文档可作为软件需求分析人员、软件设计人员以及用户之间相互沟通的基础，软件实现人员进行编码的依据及软件测试人员进行测试的依据。

本文档不涉及国家秘密，但涉及航空工业计算所的知识产权及商业秘密。本文档仅限于项目内部使用，使用者或机构应对其受众范围进行控制，对由于版权问题引起的疑问和纠纷，航空工业计算所具有最终解释权。

1. 引用文档

本章节所列文件中的条款通过本文档的引用而成为本文档的条款。凡是注明版本和日期的引用文件，其随后所有的修改单（不包括勘误的内容）或修订版均不适用于本文档。凡是未注明版本和日期的引用文件，其最新版本适用于本文档。

* 1. 国家军用标准

GJB 2786A-2009 军用软件开发通用要求 中国人民解放军总装备部 2009年

GJB 438B-2009 军用软件开发文档通用要求 中国人民解放军总装备部 2009年

GJB/Z 141-2004 军用软件测试指南 中国人民解放军总装备部 2004年

* 1. 所级标准

Q/18S 611A-2012 嵌入式软件设计标准 航空工业计算所

* 1. 项目文件

18S2.480.00318JG 发动机电子控制器软件接口需求规格说明 航空工业计算所

1. CSCI级设计决策

本CSCI采用如图5所示的软件架构方案。针对各类需求，软件共设计3类软件单元：

1. e200z4端业务逻辑：部署于e200z4处理单元，负责在SCM数据及应用软件要求的数据之间进行格式转换及异常检测，并通过HSR与eTPU进行交互。
2. eTPU端业务逻辑：部署于eTPU处理单元，由各个eTPU线程组成，实现要求的各种跳变检测及波形输出。
3. 应用软件接口：部署于e200z4处理单元，通过全局变量与周期任务函数实现与应用软件的直接数据交互。



图5　本CSCI软件架构方案

本CSCI包含的详细软件单元划分参见4　节。

1. CSCI体系结构设计

本CSCI的软件单元划分见图6、图7及图8。各个软件单元的详细设计参见5　节。



图6　本CSCI软件单元划分（总图）



图7　本CSCI软件单元划分（燃油喷射部分）



图8　本CSCI软件单元划分（曲轴凸轮采集与同步部分）

1. CSCI详细设计
   1. PWM输出驱动

PWM输出驱动软件单元用于向应用软件提供为燃油计量阀（以下简称IMV阀）、温控阀（以下简称UTHCV阀）及风扇等外部作动器输出控制信号的能力。该驱动通过接收不同的频率与占空比，控制eTPU输出对应调理电流，向应用软件提供通过IMV阀控制共轨压力腔压力，通过UTHCV阀控制尿素箱温度，及控制风扇转速的接口。

* + 1. e200z4端业务逻辑
       1. PWM输出初始化

**SDD\_ECU\_CDD\_PWM\_LOG.1**

函数原型：

int32\_t EtpuPwmInit(uint8\_t channel,

uint8\_t priority,

uint32\_t freq,

uint16\_t duty,

uint8\_t polarity,

uint8\_t timeBase,

uint32\_t timeBaseFreq)

参数说明：

channel——ETPU通道索引号，标记0~31通道。

priority——通道优先级，包括\_HIGH、MIDDLE及LOW。

freq——输出PWM波形的频率。

duty——占空比，范围为[0, 100]%，精度为0.01%。

polarity——输出PWM波形极性，包括ACTIVEHIGH 及ACTIVELOW。

timeBase——使用的时基，可选TCR1或TCR2。

timeBaseFreq——时基的频率值。

处理流程：

该软件单元用于初始化eTPU的PWM输出功能，具体处理流程详见图9。



图9　PWM输出初始化

*Traceability：IRS\_ECU\_CDD\_PWM.1，IRS\_ECU\_CDD\_PWM.2*

* + - 1. PWM输出立即更新

**SDD\_ECU\_CDD\_PWM\_LOG.2**

函数原型：

int32\_t EtpuPwmUpdateImmed(uint8\_t channel,

uint32\_t freq,

uint16\_t duty,

uint32\_t timeBaseFreq)

参数说明：

channel——ETPU通道索引号，标记0~31通道。

freq——输出PWM波形的频率。

duty——占空比，范围为[0, 100]%，精度为0.01%。

timeBaseFreq——时基的频率值。

处理流程：

该软件单元用于立刻更新eTPU的PWM输出波形，具体处理流程详见图10。



图10　PWM输出立即更新

*Traceability：IRS\_ECU\_CDD\_PWM.1，IRS\_ECU\_CDD\_PWM.2*

* + - 1. PWM输出下次更新

**SDD\_ECU\_CDD\_PWM\_LOG.3**

函数原型：

int32\_t EtpuPwmUpdateCohere(uint8\_t channel,

uint32\_t freq,

uint16\_t duty,

uint32\_t timeBaseFreq)

参数说明：

channel——ETPU通道索引号，标记0~31通道。

freq——输出PWM波形的频率。

duty——占空比，范围为[0, 100]%，精度为0.01%。

timeBaseFreq——时基的频率值。

处理流程：

该软件单元用于在下一周期更新eTPU的PWM输出波形，具体处理流程详见



图11　PWM输出下次更新

*Traceability：IRS\_ECU\_CDD\_PWM.1，IRS\_ECU\_CDD\_PWM.2*

* + 1. eTPU端业务逻辑

**SDD\_ECU\_CDD\_PWM\_LOG.4**

函数原型：

void PWM(unsigned int8 Flag,

unsigned int24 Period,

unsigned int24 ActiveTime,

unsigned int24 Coherent\_Period,

unsigned int24 Coherent\_ActiveTime)

参数说明：

Flag——内部参数，标记flag0，用于限制当前的匹配，为1时表示即将处理有效沿或正在处理无效沿，为0时表示即将处理无效沿或正在处理有效沿。

Period——PWM的周期，以TCRx的个数为单位。

ActiveTime——PWM有效电平的持续时间，以TCRx的个数为单位。

Coherent\_Period——用于下周期更新时指定周期数。

Coherent\_ActiveTime——用于下周期更新时指定有效时间。

处理流程：

PWM输出驱动软件单元的eTPU业务逻辑共包含5个处理线程，如图12所示。



图12　PWM输出驱动eTPU线程划分

其中，HSR处理线程用于处理来自e200z4处理单元服务请求，包括初始化、立即更新及下次更新。具体处理流程见图13及图14。



图13　HSR处理线程



图14　HSR为初始化时的处理流程

有效沿匹配事件处理线程及无效匹配事件处理线程用于当匹配事件发生时设置对应的输出引脚状态，产生PWM波形。其处理流程见图15及图16。



图15　有效沿匹配事件处理线程



图16　无效沿匹配事件处理线程

其他通道请求处理线程用于关闭对其他通道请求的响应。异常处理线程用于记录发生异常的通道的标识。

*Traceability：IRS\_ECU\_CDD\_PWM.1，IRS\_ECU\_CDD\_PWM.2*

* + 1. 应用软件接口

PWM输出驱动包含的应用软件接口如表1所示。

表1　PWM输出驱动接口

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 定义 | 名称 | 备注 |
| 函数 | ETPU\_PWM\_Init(void) | PWM输出初始化 |  |
| 全局变量 | PWM\_m\_i\_InitIndex | 通道索引 | PWM\_INITINDEX\_IMV：IMV阀  PWM\_INITINDEX\_UTHCV：UTHCV阀  PWM\_INITINDEX\_FAN：风扇 |
| PWM\_m\_ti\_InitCycle | 输出周期 | 值域为[2.5，62.5]ms |
| PWM\_m\_pc\_InitDuty | 输出占空比 | 值域为[0，100]% |
| 函数 | ETPU\_PWM\_Update(void) | PWM输出更新 |  |
| 全局变量 | PWM\_m\_i\_UpdateIndex | 通道索引 | PWM\_INITINDEX\_IMV：IMV阀  PWM\_INITINDEX\_UTHCV：UTHCV阀  PWM\_INITINDEX\_FAN：风扇 |
| PWM\_m\_ti\_UpdateCycle | 输出周期 | 值域为[2.5，62.5]ms |
| PWM\_m\_pc\_UpdateDuty | 输出占空比 | 值域为[0，100]% |
| PWM\_m\_pc\_UpdateMode | 更新模式 | PWM\_UPDATEMODE\_IMME：立即更新  PWM\_UPDATEMODE\_NEXT：下次更新 |

* 1. 发动机转速输出驱动

发动机转速输出驱动用于接收发动机转速信息，并通过特定PWM输出通道输出至外部系统。

* + 1. e200z4端业务逻辑

使用5.1.1　节描述的e200z4端业务逻辑，输出波形的占空比固定为50%。

* + 1. eTPU端业务逻辑

使用5.1.2　节描述的eTPU端业务逻辑，输出波形的占空比固定为50%。

* + 1. 应用软件接口

发动机转速输出驱动包含的应用软件接口如表2所示。

表2　发动机转速输出驱动接口

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 定义 | 名称 | 备注 |
| 函数 | ETPU\_SPD\_Init(void) | 发动机转速输出初始化 |  |
| 全局变量 | ENG\_m\_n\_InitSpd | 初始化输出转速 | 值域为[50，4000]rpm |
| 函数 | ETPU\_SPD\_Init\_exception(void) | 发动机转速输出更新 | 立刻更新 |
| 全局变量 | ENG\_m\_n\_UpdateSpd | 更新输出转速 | 值域为[50，4000]rpm |

* 1. 频率量输入驱动

频率量输入驱动用于从外部传感器获得应用软件指定的频率量信号状态，并将其反馈至应用软件。

* + 1. e200z4端业务逻辑
       1. 频率量输入初始化

**SDD\_ECU\_CDD\_FRQ\_LOG.1**

函数原型：

int32\_t EtpuFinInit(uint8\_t channel,

uint8\_t priority,

uint32\_t periodCnt,

uint8\_t timeBase,

uint8\_t fallingEdge)

参数说明：

channel——ETPU通道索引号，标记0~31通道。

priority——通道优先级，包括\_HIGH、MIDDLE及LOW。

periodCnt——输入信号的周期数。

timeBase——使用的时基，可选TCR1或TCR2。

fallingEdge——检测模式，为1时表示使用下降沿检测，为0时表示使用上升沿检测。

处理流程：

该软件单元用于初始化eTPU的频率量输入功能，具体处理流程详见图17。



图17　频率量输入初始化

*Traceability：IRS\_ECU\_CDD\_FREQ.1*

* + - 1. 输入频率计算

**SDD\_ECU\_CDD\_FRQ\_LOG.2**

函数原型：

uint32\_t EtpuFinGetResult(uint8\_t channel, uint32\_t timeBaseFreq)

参数说明：

channel——ETPU通道索引号，标记0~31通道。

timeBaseFreq——频率计算使用的时基频率。

处理流程：

该软件单元用于根据输入信号状态计算对应频率值，具体处理流程见图18。



图18　输入频率计算

*Traceability：IRS\_ECU\_CDD\_FREQ.1*

* + 1. eTPU端业务逻辑

**SDD\_ECU\_CDD\_FRQ\_LOG.3**

函数原型：

void FreqIn(unsigned int24 Result,

unsigned int24 PeriodCnt,

unsigned int24 PeriodCntUpdate)

参数说明：

Result——输入信号在一个周期包含的时基数均值。

PeroidCnt——输入信号的周期数。

PeriodCntUpdate——需要更新的输入信号周期数。

处理流程：

频率量输入驱动软件单元的eTPU业务逻辑共包含4个处理线程，如图19所示。



图19　频率量输入驱动eTPU线程划分

其中，初始化线程用于设置对应eTPU输入通道的工作模式及检测条件，处理流程见图20。



图20　初始化线程

跳变检测线程用于检测输入频率信号的波形状态，统计并计算频率值，处理流程见图21。



图21　跳变检测线程

更新计数周期线程在计数周期无效时，清零更新标志。异常处理线程用于记录发生异常的通道的标识。

*Traceability：IRS\_ECU\_CDD\_FREQ.1*

* + 1. 应用软件接口

频率量输入驱动包含的应用软件接口如表3所示。

表3　频率量输入驱动接口

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 定义 | 名称 | 备注 |
| 函数 | ETPU\_FRQ\_Get(void) | 频率量输入 |  |
| 全局变量 | FRQ\_m\_i\_Index | 通道号 | FRQ\_INDEX\_FAN：风扇  FRQ\_INDEX\_CARSPD：车速  FRQ\_INDEX\_ENGSPD：发动机转速表 |
| FRQ\_m\_u\_Result | 频率值 | 异常时为0 |

* 1. 燃油喷射驱动

燃油喷射驱动依据输入数据控制eTPU的对应输出通道，驱动各缸喷油器喷射燃油。

* + 1. e200z4端业务逻辑
       1. 喷油初始化

**SDD\_ECU\_CDD\_INJ\_LOG.1**

函数原型：

int32\_t EtpuFuelInit2(uint8\_t chn,

uint8\_t priority,

uint8\_t cynNum,

uint8\_t railChn,

uint32\_t \*pInjTimeArr,

uint32\_t railAngle,

uint32\_t exprEdgeAngle,

uint32\_t anglePerCycle)

参数说明：

chn——ETPU通道索引号，标记0~31通道。

priority——通道优先级，包括\_HIGH、MIDDLE及LOW。

cynNum——缸编号，从1开始到6结束。

railChn——轨压采集通道号。

pInjTimeArr——1~6缸的喷油参数数组。

railAngle——轨压中断提前角。

exprEdgeAngle——凸轮上升沿对应的TCR2计数值。

anglePerCycle——发动机每圈的角度计数。

处理流程：

该软件单元用于初始化各个喷油气缸对应的eTPU通道，具体处理流程详见图22。



图22　喷油初始化

*Traceability：IRS\_ECU\_CDD\_INJ.1*

* + - 1. 喷油时序设置

**SDD\_ECU\_CDD\_INJ\_LOG.2**

函数原型：

void EtpuFuelSetInjTime(uint8\_t chn, uint8\_t cynNum, uint32\_t \*pInjTimeArr)

参数说明：

chn——ETPU通道索引号，标记0~31通道。

cynNum——缸编号，从1开始到6结束。

pInjTimeArr——保存喷油起始角度计数和持续时间计数的数组。

处理流程：

该软件单元用于设置各个气缸的喷油时序，具体处理流程见图23。



图23　喷油时序设置

*Traceability：IRS\_ECU\_CDD\_INJ.1*

* + 1. eTPU业务逻辑

**SDD\_ECU\_CDD\_INJ\_LOG.1**

void FuelMain(unsigned int24 angleOffset,

unsigned int24 startAngle,

unsigned int24 \*pInjTimeArr,

unsigned int24 deadAngle,

unsigned int24 exprEdgeAngle,

unsigned int24 angleTickPerEngCycle)

参数说明：

angleOffset——本缸角度偏移，以TCR2个数为单位。

startAngle——喷油提前角，以TCR2个数为单位，正值表示比上止点提前，负值表示比上止点滞后

pInjTimeArr——保存喷油持续时间和相邻喷油间隔时间的数组，数组应包含9个 元素，其中元素0和1分别表示预喷2的持续时间，预喷2到预喷1的间隔时间；元素2和3分别表示预喷1的持续时间，预喷1到主喷的间隔时间；元素4和5分别表示主喷的持续时间，主喷到后喷1的间隔时间；元素6和7分别表示后喷1的持续时间，后喷1到后喷2的间隔时间；元素8表示后喷2的持续时间。

deadAngle——喷油结束角度，超过这个角度后，本缸不允许再次喷油。

exprEdgeAngle——凸轮标记齿对应的角度计数。

angleTickPerEngCycle——引擎每圈的角度计数值，以TCR2计数为单位。

处理流程：

燃油喷射驱动软件单元的eTPU业务逻辑共包含6个处理线程，如图24所示。喷油序列更新线程、喷由时序调度线程及喷油停止角处理线程包含触发时喷油有效或喷油无效的不同处理流程。



图24　燃油喷射驱动eTPU线程划分

其中，初始化线程用于配置eTPU通道的工作模式并计算对应喷油参数的初始值。具体处理流程见图25。



图25　初始化处理线程

喷油序列更新线程用于在喷油过程中处理来自应用软件的喷油参数，形成更新后的喷油序列，调整喷射气缸的燃油喷射状态。根据该线程触发时喷油状态的不同（喷油有效或喷油无效），其包含两种不同的处理过程，详见图26及图27。



图26　喷油无效时的喷油序列更新线程



图27　喷油有效时的喷油序列更新线程

喷油时序调度线程用于根据喷油序列计算各次喷射的起始时间及停止时间，控制eTPU通道输出对应波形。根据该线程触发时喷油状态的不同（喷油有效或喷油无效），其包含两种不同的处理过程，详见图28及图29。



图28　喷油无效时的喷油时序调度线程



图29　喷油有效时的喷油时序调度线程

喷油停止角处理线程在到达喷油停止角时根据喷油状态控制eTPU输出波形保护外部作动器，并设置之后的匹配事件。根据该线程触发时喷油状态的不同（喷油有效或喷油无效），其包含两种不同的处理过程，详见图30及图31。



图30　喷油无效时的喷油停止角处理线程



图31　喷油有效时的喷油停止角处理线程

角度计算线程用于根据喷油序列计算各次喷射对应的起始角度和结束角度。具体处理流程见图32。



图32　角度计算线程

异常处理线程用于记录发生异常的通道的标识。

*Traceability：IRS\_ECU\_CDD\_INJ.1*

* + 1. 应用软件接口

燃油喷射驱动包含的应用软件接口如表4所示。

表4　燃油喷射驱动接口

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 定义 | 名称 | 备注 |
| 函数 | ETPU\_Fuel\_Set(void) | 燃油喷射 |  |
| 全局变量 | INJ\_m\_i\_Index | 喷射气缸索引号 | INJ\_INDEX\_1~ INJ\_INDEX\_6 |
| INJ\_m\_phi\_Parameter[10] | 喷射时序数组 | INJ\_PRE\_2\_ANGLE：预喷2起始角  INJ\_PRE\_2\_DURATION：预喷2持续时间  INJ\_PRE\_1\_ANGLE：预喷1起始角  INJ\_PRE\_1\_DURATION：预喷1持续时间  INJ\_MAIN\_ANGLE：主喷起始角  NJ\_MAIN\_DURATION：主喷持续时间  INJ\_RE\_1\_ANGLE：后喷1起始角  INJ\_RE\_1\_DURATION：后喷1持续时间  INJ\_RE\_2\_ANGLE：后喷2起始角  INJ\_RE\_2\_DURATION：后喷2持续时间 |

* 1. 曲轴凸轮采集与同步驱动

曲轴凸轮采集与同步驱动根据曲轴位置传感器及凸轮位置传感器的状态计算与曲轴及凸轮相关的同步状态信息，并以此进行故障检测，判断并反馈发动机的工作状态信息。

* + 1. e200z4端业务逻辑
       1. 同步初始化

**SDD\_ECU\_CDD\_SYN\_LOG.1**

函数原型：

int32\_t EtpuEngPosInit2(uint8\_t camChannel,

uint8\_t camPrio,

uint32\_t camFirstTimeout,

ufract24\_t camWinRatioNormal,

ufract24\_t camWinRatioACMark,

ufract24\_t camWinRatioAFMark,

ufract24\_t camMarkRatio,

uint8\_t crankChannel,

uint8\_t crankPrio,

uint8\_t crankTeethPerRev,

uint8\_t crankMissingTeeth,

uint32\_t crankBlankTime,

uint8\_t crankBlankTeeth,

uint32\_t crankFirstTimeout,

ufract24\_t crankWinRatioNormal,

ufract24\_t crankWinRatioACGap,

ufract24\_t crankWinRatioAFGap,

ufract24\_t crankWinRatioTimeout,

ufract24\_t crankGapRatio,

uint32\_t linkGroup1,

uint32\_t linkGroup2)

参数说明：

camChannel——凸轮检测通道号。

camPrio——凸轮通道优先级，包括\_HIGH、MIDDLE及LOW。

camAngleWinStart——凸轮检测时角度窗口的起始，取值范围为0~71999，精度为0.01°。

camAngleWinWidth——凸轮检测时角度窗口的宽度，取值范围为0~71999，精度为0.01°。

crankChannel——曲轴检测通道号。

crankPrio——曲轴通道优先级，包括\_HIGH、MIDDLE及LOW。

crankTeethPerRev——曲轴每圈的齿数，应包含缺齿。

crankMissingTeeth——曲轴缺齿数。

crankWinRatioNormal——正常齿时的窗口系数，取值为0~0xFFFFFF，代表0~1。

crankWinRatioACGap——缺齿后第一个齿的窗口系数，取值同上。

crankWinRatioAFGap——缺齿后第二个齿的窗口系数，取值同上。

crankWinRatioTimeout——超时后的齿的窗口系数，取值同上。

crankGapRatio——识别缺齿时的缺齿系数，取值同上。

处理流程：

该软件单元用于初始化曲轴凸轮同步功能，具体处理流程见。



图33　同步初始化

*Traceability：IRS\_ECU\_CDD\_SYN.1*

* + - 1. 凸轮瞬时转速计算

**SDD\_ECU\_CDD\_SYN\_LOG.2**

函数原型：

uint32\_t EtpuEngPosGetCamSpeed(uint8\_t camChannel, uint32\_t tcr1Freq)

参数说明：

camChannel——凸轮检测通道号。

tcr1Freq——TCR1的频率。

处理流程：

该软件单元用于计算凸轮瞬时转速，具体处理流程见图34。



图34　凸轮瞬时转速计算

*Traceability：IRS\_ECU\_CDD\_SYN.5*

* + - 1. 曲轴瞬时转速计算

**SDD\_ECU\_CDD\_SYN\_LOG.3**

函数原型：

uint32\_t EtpuEngPosGetCrankSpeed(uint8\_t crankChannel, uint32\_t tcr1Freq)

参数说明：

crankChannel——曲轴检测通道号。

tcr1Freq——TCR1的频率。

处理流程：

该软件单元用于计算曲轴瞬时转速，具体处理流程见图35。



图35　曲轴瞬时转速计算

*Traceability：IRS\_ECU\_CDD\_SYN.4*

* + - 1. 曲轴凸轮平均转速计算

**SDD\_ECU\_CDD\_SYN\_LOG.4**

函数原型：

void aveSpeedCompt(void)

处理流程：

该软件单元用于根据曲轴转速及凸轮转速的瞬时值计算平均值，具体处理流程见图36。



图36　曲轴凸轮平均转速计算

*Traceability：IRS\_ECU\_CDD\_SYN.6~7*

* + 1. eTPU业务逻辑
       1. 曲轴相关处理线程

**SDD\_ECU\_CDD\_SYN\_LOG.5**

函数原型：

void CrankDetect(unsigned int8 errorStatus,

unsigned int8 camChannel,

unsigned int24 crankTeethRev,

unsigned int24 crankNumMissing,

unsigned int24 blankTime,

unsigned int8 blankTeeth,

unsigned int24 firstTimeout,

unsigned fract24 winRatioNormal,

unsigned fract24 winRatioACGap,

unsigned fract24 winRatioAFGap,

unsigned fract24 winRatioTimeout,

unsigned fract24 gapRatio,

unsigned int32 linkGroup1,

unsigned int32 linkGroup2,

unsigned int24 hostAssertTeeth,

signed int24 angleAdjust,

unsigned int8 crankState,

unsigned int24 toothCount,

unsigned int24 toothPeriodA)

参数说明：

errorStatus——错误状态。

camChanne——凸轮通道索引。

crankTeethRev——曲轴每圈齿数（包含缺齿）。

crankNumMissing——曲轴缺齿数。

blankTime——初始化后的空白时间，以TCR1计数为单位。

blankTeeth——空白时间后的空白齿个数。

winRatioNormal——下一个齿为正常齿时的判断窗口系数。

winRatioACGap——缺齿后第一个正常齿的判断窗口系数。

winRatioAFGap——缺齿后第二个正常齿的判断窗口系数。

winRatioTimeout——超时的判断窗口系数。

gapRatio——缺齿的ABa检测系数。

linkGroup1——曲轴停车后需要发出链接请求的通道索引首部。

linkGroup2——曲轴停车后需要发出链接请求的通道索引尾部。

hostAssertTeeth——主处理器插入的齿数计数。

angleAdjust——用于调整TCR2的齿数计数。

crankState——曲轴状态。

toothCount——用来计算曲轴转速的齿计数。

toothPeriodA——用来计算曲轴转速的齿周期。

处理流程：

曲轴凸轮采集与同步驱动软件单元的曲轴相关eTPU业务逻辑共包含9个处理线程，如图37所示。



图37　曲轴相关eTPU线程划分

其中，初始化线程用于配置eTPU通道的工作模式、配置各个标志的初始状态并算曲轴齿相关的参数。具体处理流程见图38。



图38　初始化线程

角度计数调整线程用于响应主处理器端直接调整角度计数的服务请求。具体处理流程见图39。



图39　角度计数调整线程

主处理器插入齿处理线程用于响应主处理器端要求插入齿计数时的服务请求。具体处理流程见图40。



图40　主处理器插入齿处理线程

正常齿处理线程用于在正常齿计数过程中计算并校正窗口大小，及当窗口超时时判断超时原因。具体处理流程见图41。



图41　正常齿处理线程

缺齿前齿处理线程用于当缺齿前的最后一齿到来时计算接纳缺齿的窗口。具体处理流程见图42。



图42　缺齿前齿处理线程

缺齿后齿处理线程用于计算当缺齿周期到来后的下一齿检测窗口，对缺齿周期进行验证。具体处理流程见图43。



图43　缺齿后齿处理线程

其他齿处理线程用于在初始化之后直到检测到首个缺齿之间的时间段内的跳变沿处理。具体处理流程见图44、图45、图46及图47。



图44　其他齿处理线程（总图）



图45　其他齿处理线程（子图1）



图46　其他齿处理线程（子图2）



图47　其他齿处理线程（子图3）

发动机同步状态更新线程用于根据当前的曲轴及凸轮状态更新发动机同步状态。具体处理流程见图48。



图48　发动机同步状态更新线程

故障处理线程用于根据当前故障标志判断曲轴状态，并进行对应处理。具体处理流程见图49。



图49　故障处理线程

* + - 1. 凸轮相关处理线程

**SDD\_ECU\_CDD\_SYN\_LOG.2**

函数原型：

void CamDetect(unsigned int8 errorStatus,

unsigned int8 crankChannel,

unsigned int24 firstTimeout,

unsigned fract24 winRatioNormal,

unsigned fract24 winRatioACMark,

unsigned fract24 winRatioAFMark,

unsigned fract24 markRatio,

unsigned int24 camPeriodA,

unsigned int24 edgeAngle)

参数说明：

errorStatus——凸轮检测错误状态。

crankChannel——曲轴通道索引。

firstTimeout——首次的超时时间设置值。

winRatioNormal——正常齿的窗口系数。

winRatioACMark——下一个齿是标记齿的窗口计算系数。

winRatioAFMark——下一个齿是标记齿后的齿的窗口计算系数。

markRatio——标记齿周期判断系数。

camPeriodA——凸轮齿的周期。

edgeAngle——标记齿上升沿对应的角度计数。

处理流程：

曲轴凸轮采集与同步驱动软件单元的凸轮相关eTPU业务逻辑共包含6个处理线程，如图50所示。



图50　凸轮相关eTPU线程划分

其中，初始化线程用于配置eTPU通道的工作模式、配置各个标志的初始状态凸轮检测特性。具体处理流程见图51。



图51　初始化线程

正常齿处理线程用于在正常齿计数过程中计算凸轮周期及窗口大小。具体处理流程见图52。



图52　正常齿处理线程

标志齿前齿处理线程用于计算并设置凸轮标志齿的验证窗口大小。具体处理流程见图53。



图53　标志齿前齿处理线程

标志齿后齿处理线程用于对凸轮标志齿验证窗口，并验证标志齿周期。具体处理流程见图54。



图54　标志齿后齿处理线程

其他齿处理线程用于在初始化之后直到进入正常齿检测之间的时间段内的跳变沿处理。具体处理流程见图55及图56。



图55　其他齿处理线程（总图）



图56　其他齿处理线程（子图）

故障处理线程用于根据当前故障标志判断凸轮状态，并进行对应处理。具体处理流程见图57。



图57　故障处理线程

*Traceability：IRS\_ECU\_CDD\_SYN.1~9*

* + 1. 应用软件接口

曲轴凸轮采集与同步驱动包含的应用软件接口如表5所示。

表5　曲轴凸轮采集与同步驱动接口

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 定义 | 名称 | 备注 |
| 函数 | EPS\_GetCrankSta(void) | 获得曲轴状态 |  |
| EPS\_GetCamSta(void) | 获得凸轮状态 |  |
| EPS\_GetCrankSpd(void) | 获取曲轴瞬时转速 |  |
| EPS\_GetCamSpd(void) | 获得凸轮瞬时转速 |  |
| EPS\_GetCrankAndCamAvgSpd(void) | 获得凸轮及曲轴平均转速 |  |
| EPS\_GetEngSpdAndAvgSpd(void) | 获得发动机转速及平均转速 |  |
| EPS\_GetEngSta(void) | 获得发动机同步状态 |  |
| EPS\_GetSenSta(void) | 获得传感器状态 |  |
| EPS\_Bit (void) | 获得同步故障检测状态 |  |

续表5

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 定义 | 名称 | 备注 |
| 全局变量 | EPS\_m\_st\_CrkSync | 曲轴状态 | FS\_ETPU\_CRANK\_SEEK：初始状态  FS\_ETPU\_CRANK\_BLANK\_TIME：初始化后空白时间  FS\_ETPU\_CRANK\_BLANK\_TEETH：空白齿检测  FS\_ETPU\_CRANK\_FIRST\_EDGE：检测到首个齿  FS\_ETPU\_CRANK\_SECOND\_EDGE：检测到第二齿  FS\_ETPU\_CRANK\_TEST\_POSSIBLE\_GAP：检测到缺齿  FS\_ETPU\_CRANK\_VERIFY\_GAP：验证缺齿  FS\_ETPU\_CRANK\_GAP\_VERIFIED：缺齿已验证  FS\_ETPU\_CRANK\_COUNTING：正常运行  FS\_ETPU\_CRANK\_TOOTH\_BEFORE\_GAP：当前齿为缺齿前齿  FS\_ETPU\_CRANK\_TOOTH\_AFTER\_GAP：当前齿为缺齿后齿 |
| EPS\_m\_st\_CamSync | 凸轮状态 | FS\_ETPU\_CAM\_FIRST\_EDGE：检测到首个齿  FS\_ETPU\_CAM\_SECOND\_EDGE：检测到第二齿  FS\_ETPU\_CAM\_TEST\_POSSIBLE\_MARK：检测到标志齿  FS\_ETPU\_CAM\_VERIFY\_NARK：验证标志齿  FS\_ETPU\_CAM\_MARK\_VERIFIED：标志齿已验证  FS\_ETPU\_CAM\_COUNTING：正常运行  FS\_ETPU\_CAM\_TOOTH\_BEFORE\_MARK：当前齿为标志齿前齿  FS\_ETPU\_CAM\_TOOTH\_AFTER\_GAP：当前齿为标志齿后齿 |
| EPS\_m\_n\_Crank | 曲轴瞬时转速 |  |
| EPS\_m\_n\_Cam | 凸轮瞬时转速 |  |
| EPS\_m\_n\_CrankAvg | 曲轴平均转速 |  |
| EPS\_m\_n\_CamAvg | 凸轮平均转速 |  |
| EPS\_m\_n\_Engine | 发动机瞬时转速 | 当前为曲轴瞬时转速 |
| EPS\_m\_n\_EngineAvg | 发动机平均转速 | 当前为曲轴平均转速 |

续表5

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 定义 | 名称 | 备注 |
|  | EPS\_m\_st\_Status | 发动机同步状态 | FS\_ETPU\_ENG\_POS\_SEEK：同步初始化  FS\_ETPU\_ENG\_POS\_FIRST\_HALF\_SYNC：首次半同步（已验证缺齿，但无凸轮信号）  FS\_ETPU\_ENG\_POS\_PRE\_FULL\_SYNC：发动机预同步（在当前曲轴周期中检测到凸轮信号）  FS\_ETPU\_ENG\_POS\_HALF\_SYNC：发动机半同步（正常运行后未在期望的位置检测到凸轮信号）  FS\_ETPU\_ENG\_POS\_FULL\_SYNC：发动机同步 |
| EPS\_m\_st\_SenSta | 传感器状态 | EPS\_SEN\_STA\_CRANK：曲轴传感器  EPS\_SEN\_STA\_CAM：凸轮传感器  EPS\_SEN\_STA\_ERR：传感器故障 |

1. 需求可追踪性

本文档包含的所有软件单元的追踪性描述详见对应软件单元的*Traceability*属性。

1. 注释

DSPI Deserial Serial Peripheral Interface

ECU Engine Control Unit

JTAG Joint Test Action Group

HSR Host Service Request

N/A Not Applicable

PWM Pulse Width Modulation

SCM Shared Code Memory