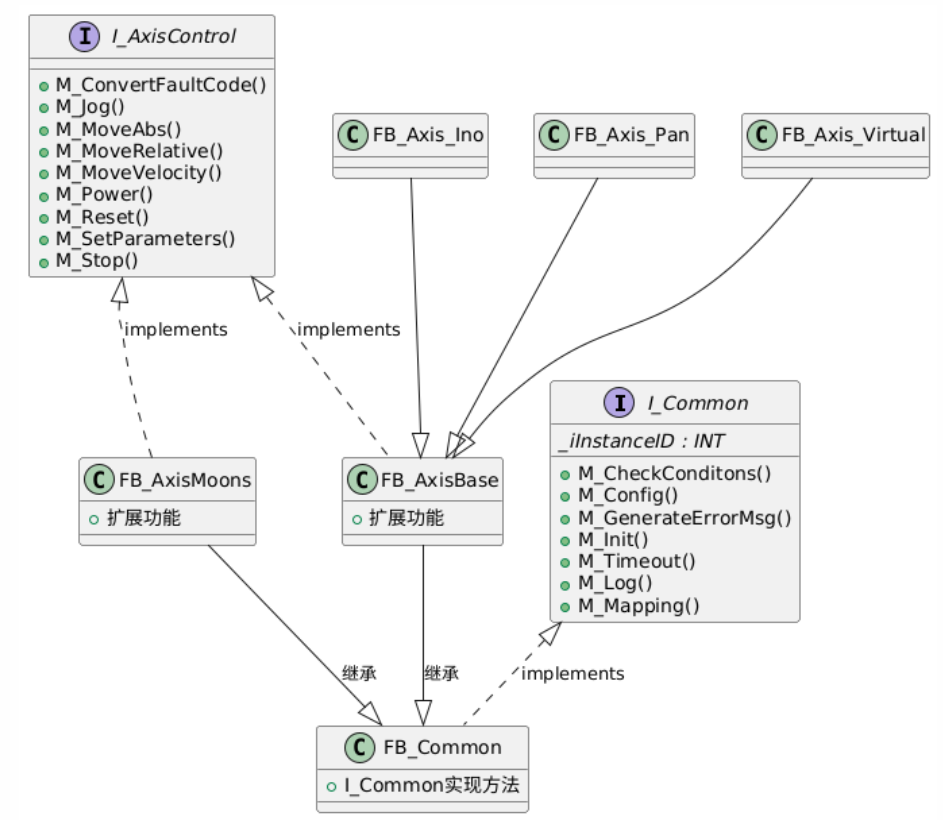
Motion Control开发文档

## 1. 概述

AxisControl模块用于控制轴运动，提供了多种运动控制方法。该模块通过接口和类的继承与实现，遵循面向对象编程（OOP）的基本原则，如封装、继承和多态。



## 2. 面向对象设计

### 2.1 接口定义

#### I\_Common 接口：

系统的通用接口

* + 属性
    - \_iInstanceID : 实例ID，用于标识不同实例，比如有10个伺服，通过instanceID,报警时就能够知道是哪个实例的报警
  + 方法
    - M\_CheckConditons()：用于状态判断跳转，检查条件是否都满足，如果都满足，则返回TRUE
    - M\_Config()：配置相关参数
    - M\_GenerateErrorMsg()：生成错误信息，错误码，描述是什么实例在什么状态报的警
    - M\_Init()：初始化操作，用来写入伺服参数。
    - M\_Timeout()：处理超时情况，判断状态报警或者延时跳转。
    - M\_Log()：记录日志信息。
    - M\_Mapping()：映射相关数据。

#### I\_AxisControl 接口：

基于虚拟轴以及实轴的通用接口

* + 方法
    - M\_ConvertFaultCode(): 转换故障代码。
    - M\_Jog(): 点动操作，使轴以设定的速度和方向进行短距离移动。
    - M\_MoveAbs(): 绝对位置移动，将轴移动到指定的绝对位置。
    - M\_MoveRelative(): 相对位置移动，将轴从当前位置移动指定的距离。
    - M\_MoveVelocity(): 速度模式移动，以设定的速度进行连续运动。
    - M\_Power(): 控制伺服使能，使电机处于激磁状态。
    - M\_Reset(): 复位操作，清除故障状态并恢复正常运行状态。
    - M\_SetParameters(): 设置轴控制参数，如速度、加速度等参数。
    - M\_Stop(): 停止当前运动。

### 2.2.功能块定义

#### FB\_Common

* + 职责：实现了I\_Common接口中的所有方法，提供通用功能支持。
  + 方法： 实现了所有I\_Common接口的方法，包括检查条件、配置参数、生成错误信息、初始化、处理超时、记录日志和数据映射等功能。

具体实现的方法：

##### M\_CheckConditons()：

对所有的状态跳转进行管理，只有所有条件都满足了才进入下一步

##### M\_Log()：

所有的模块都要有能够收集日志信息并生成日志文件并进行日志输出的能力。~~common的暂相当于虚函数，后续整理后可以在这里写一个~~

##### M\_GenerateErrorMsg()：

根据\_instanceID以及当前的状态生成errorID,并根据报警文本策略生成相应的文本用于显示

##### M\_Mapping()：

用于给子类映射相关数据，这里的方法相当于是虚方法

M\_Config()：用于整理生成一些用于日志使用的参数

##### M\_Init()：

#### FB\_AxisBase ()

VAR\_IN\_OUT

stMotionCtrl : ST\_MotionControl;//HMI 调用

AXIS\_REF : AXIS\_REF\_SM3;//轴地址

END\_VAR

* + 职责：扩展了`FB\_Common`类，并增加了一些基于AXIS\_REF\_SM3的轴控制的扩展功能，实现了`I\_AxisControl`接口中的所有方法。
  + 输入输出：stMotionCtrl用于对接HMI，AXIS\_REF接口对接所有基于AXIS\_REF\_SM3的扩展的轴
  + 方法：实现了所有I\_AxisControl接口的方法，包括转换故障代码、点动操作、绝对位置移动、相对位置移动、速度模式移动、电源控制、复位操作、设置参数和停止当前运动等功能。

##### M\_MoveAbs()

根据输入的位置，速度，加速度，减速度进行绝对位置运动，运动完成后返回TRUE

实例调用

##### M\_MoveRelative()

根据输入的位置，速度，加速度，减速度进行相对位置运动，运动完成后返回TRUE

实例调用

##### M\_Jog

根据输入进行点动运动，点动加减速收轴的加减速配置以及速度比例设置影响

当enable=true,并且rVel>0是，往前点动

当enable=true,并且rVel<0是，往后点动

当enable=false,点动停止

实例调用

#### 多态

这两个类分别针对INOVANCE（汇川）与Panasonic（松下）品牌伺服驱动器进行了具体实现。它们继承自FB\_AxisBase类，并根据各自品牌伺服驱动器的特点进行了扩展和优化。

##### FB\_Axis\_Ino 类

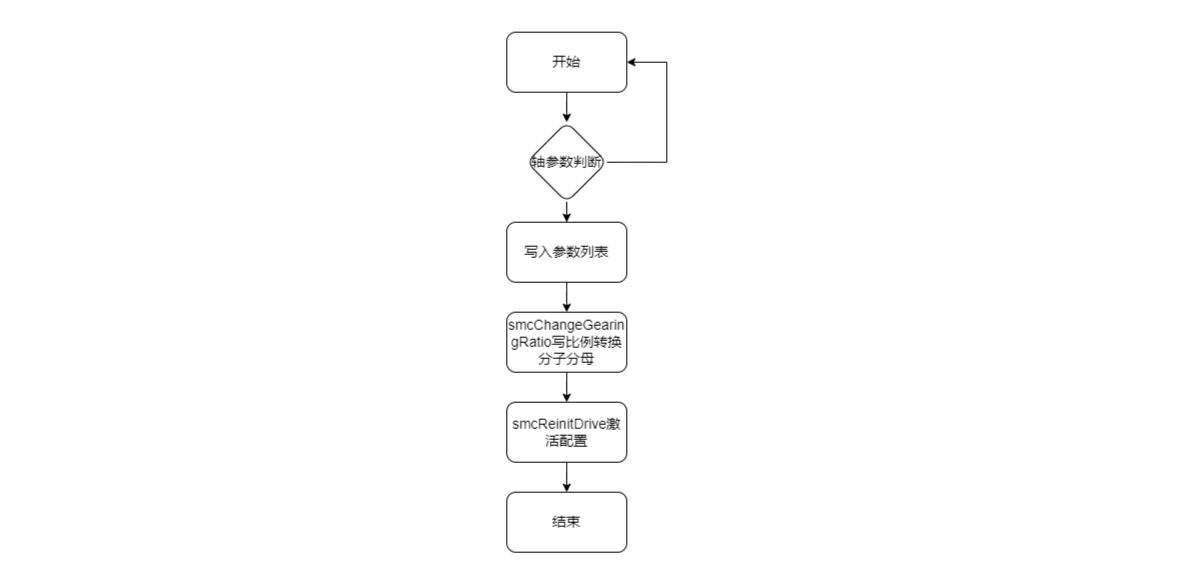
* 职责：针对INOVANCE品牌伺服驱动器进行具体实现，包括特定的通信协议处理及故障代码转换等功能。
* 扩展功能：覆盖或扩展父类中的某些方法，以适应INOVANCE伺服驱动器的特殊需求。例如，可以覆盖M\_ConvertFaultCode（）来处理INOVANCE特有的故障代码格式。

###### M\_ConvertFaultCode()

根据INOVANCE的错误码定义转换为错误信息

###### M\_SetParameters()

根据INOVANCE的通讯手册以及本身的需求确定参数内容，并写入，只需在上电初始化流程里调用一次



##### FB\_Axis\_Pan 类

* 职责：针对Panasonic品牌伺服驱动器进行具体实现，包括特定的通信协议处理及故障代码转换等功能。
* 扩展功能：覆盖或扩展父类中的某些方法，以适应Panasonic伺服驱动器的特殊需求。例如，可以覆盖M\_ConvertFaultCode（）来处理Panasonic特有的故障代码格式。

###### M\_ConvertFaultCode()

根据INOVANCE的错误码定义转换为错误信息

## 程序调用示例

fbapplication里边定义接口：接口映射到通讯内存地址

fbsample定义接口，在fbsample.m\_mapping()接收fbapplication里边接口的值

fbsample根据接口定义实现逻辑

## 附录：

### UML定义

@startuml

' 接口定义

interface I\_Common {

{abstract} \_iInstanceID : INT

+ M\_CheckConditons()

+ M\_Config()

+ M\_GenerateErrorMsg()

+ M\_Init()

+ M\_Timeout()

+ M\_Log()

+ M\_Mapping()

}

interface I\_AxisControl {

+ M\_ConvertFaultCode()

+ M\_Jog()

+ M\_MoveAbs()

+ M\_MoveRelative()

+ M\_MoveVelocity()

+ M\_Power()

+ M\_Reset()

+ M\_SetParameters()

+ M\_Stop()

}

' 类继承结构

class FB\_Common {

+ I\_Common实现方法

}

FB\_Common .up.|> I\_Common : implements

class FB\_AxisBase {

+ 扩展功能

}

FB\_AxisBase --|> FB\_Common : 继承

FB\_AxisBase .up.|> I\_AxisControl : implements

class FB\_AxisMoons {

+ 扩展功能

}

FB\_AxisMoons --|> FB\_Common : 继承

FB\_AxisMoons .up.|> I\_AxisControl : implements

class FB\_Axis\_Ino

class FB\_Axis\_Pan

class FB\_Axis\_Virtual

FB\_Axis\_Ino --|> FB\_AxisBase

FB\_Axis\_Pan --|> FB\_AxisBase

FB\_Axis\_Virtual --|> FB\_AxisBase

@enduml