|  |
| --- |
| **上海赫千电子科技有限公司** |
| **CM设计文档** |
| |  |  |  |  | | --- | --- | --- | --- | | 版本 | 发布日期 | 描述 | 发布人 | | 1.0 |  |  | Hinge | |  |  |  |  | |  |  |  |  | |  |  |  |  | |
|  |

|  |
| --- |
|  |

[1. 目的 2](#_Toc500251972)

[2. 功能概述 2](#_Toc500251973)

[3. 定义与缩写 2](#_Toc500251974)

[4. 系统架构 2](#_Toc500251975)

[4.1. CM模块的组成 2](#_Toc500251980)

[4.2. CM模块的部署 2](#_Toc500251981)

[4.3. CM模块的API 3](#_Toc500251982)

[4.3.1. ServicveSkeleton类的接口 3](#_Toc500251983)

[4.3.2. ServiceProxy类的接口 3](#_Toc500251984)

[4.3.3. Method类的接口 3](#_Toc500251985)

[4.3.4. PublishEvent类的接口 3](#_Toc500251986)

[4.3.5. SubcribeEvent类的接口 3](#_Toc500251987)

[5. 系统逻辑 4](#_Toc500251988)

[5.1. 时序图 4](#_Toc500251990)

[5.1.1. Event publish 4](#_Toc500251991)

[5.1.2. Event Subscribe 5](#_Toc500251992)

[5.1.3. Event Update 6](#_Toc500251993)

[5.1.4. Method Request And Response 7](#_Toc500251994)

[5.1.5. Proxy Create 8](#_Toc500251995)

[5.1.6. Skeleton Create 9](#_Toc500251996)

[6. 函数及参数介绍 9](#_Toc500251998)

[6.1. ServiceSkeleton类 9](#_Toc500251999)

[6.2. ServiceProxy类 10](#_Toc500252000)

[6.3. Method类 10](#_Toc500252001)

[6.4. PublishEvent类 11](#_Toc500252002)

[6.5. SubscribeEvent类 11](#_Toc500252003)

[7. 库/数据库文件描述 12](#_Toc500252004)

[8. 消息机制 12](#_Toc500252005)

[9. 代码文件 12](#_Toc500252006)

# 目的

本文档是CM功能群集的详细设计说明文档，描述了该功能的详细设计方案。CM实现面向服务的通信，适用于Adaptive AUTOSAR Platform应用程序的各级通信。本文以AUTOSAR的Adaptive Platform 17-03标准文档为基础，是一个CM提供者的实现方案和应用开发者的使用说明。

# 功能概述

主要功能描述，或模块系统架构图。

CM是Adaptive AUTOSAR Platform标准的一个功能群集，包括CM提供的API的实现、

SOME/IP协议的使用、SOME/IP-SD协议的使用和进程间通信IPC方式的使用。

CM提供的API有服务提供与发现接口、Method的调用接口、Event的订阅与发布接口。

# 定义与缩写

关键词汇，定义及缩写词语描述。

API：Application Programming Interface

ARXML：AUTOSAR XML

CM：Communication Management

SOME/IP：Scalable service-Oriented MiddlewarE over IP protocol

SOME/IP-SD：SOME/IP Service Discovery protocol

# 系统架构



## CM模块的组成

CM模块由SOME/IP协议库、SOME/IP-SD协议的Server进程，SOME/IP-SD协议的API库，CM的API库和IPC库组成。

## CM模块的部署

1. 一个ECU上只运行一个SOME/IP-SD协议的Server进程。
2. 基于CM模块的每个应用程序，编译需要链接CM的API库、SOME/IP-SD协议API库、SOME/IP协议库和IPC库。

## CM模块的API

CM模块的API以C++类的形式提供，包括Skeleton类、Proxy类、Method类和Event类。

### ServicveSkeleton类的接口

1. ServiceSkeleton
2. OfferService
3. StopOfferService
4. Init

### ServiceProxy类的接口

1. ServiceProxy
2. FindService
3. StartFindService
4. StopFindService
5. Init

### Method类的接口

1. operator()

### PublishEvent类的接口

1. Send
2. Allocate

### SubcribeEvent类的接口

1. Subscribe
2. Unsubscribe
3. Update
4. GetCacheSamples
5. Cleanup
6. SetReceiveHandler
7. UnsetReceiveHandler

## service开发流程

1. 定义Service Interface

定义新增的数据类型，Events，Methods，Service

生成Service Interface的描述arxml文件

1. 生成Service Interface头文件

通过代码生成工具和Service Interface的描述arxml文件，生成Service Interface头文件

1. 配置序列化的属性

配置Service Interface所用数据的序列化的属性，生成arxml文件。

1. 生成序列化代码

通过代码生成工具和序列化的属性arxml文件，生成序列化代码。

1. 配置部署Service
2. 生成Service部署arxml文件

实现Service Proxy和Skeleton代码

使用Service Interface头文件，序列化代码和Service部署arxml文件，实现Service Proxy和Skeleton代码

1. 配置Service Instance

生成Service Instance的Manifest文件

# 系统逻辑

## 类图



## 流程图

### client流程图



图5-1 Client流程图

### Server流程图



图5-2 Server流程图

## 时序图

### Event publish



图5-3 Event publish

### Event Subscribe



图5-4 Event Subscribe

### Event Update



图5-5 Event Update

### Method Request And Response



图5-6 Method Request And Response

### Proxy Create



图5-7 Proxy Create

### Skeleton Create



图5-8 Skeleton Create



# 函数及参数介绍

## ServiceSkeleton类

1. ServiceSkeleton(InstanceIdentifier instance, MethodCallProcessingMode mode = MethodCallProcessingMode::kEvent)

功能：ServiceSkeleton的构造函数。

参数：

instance 服务的特定实例的标识符。

mode 指定服务方法的处理模式，默认是MethodCallProcessingMode::kEvent。

返回值：无。

1. void OfferService()

功能：提供服务的方法，CM要检测是否存在相同的service ID和instance ID,检测提供服务的协议。

参数：无。

返回值：无。

1. void StopOfferService()

功能：停止提供服务的方法。

参数：无。

返回值：无。

4）bool Init(Configuration\* conf)

功能：初始化。

参数：

conf service instance manifest。

返回值：true初始化成功, false初始化失败。

## ServiceProxy类

1. ServiceProxy(HandleType handle)

功能：ServiceProxy类的构造函数。

参数：

handle FindService方法的返回值。

返回值：无。

1. static ServiceHandleContainer<HandleType> FindService(InstanceIdentifier instance = InstanceIdentifier::Any)

功能：查找服务的方法。

参数：指定要查找的服务的实例的标识符。

返回值：包含所有匹配服务实例的句柄的容器。

1. static FindServiceHandle StartFindService(FindServiceHandler<HandleType> handler, InstanceIdentifier instance)

功能：注册处理回调的查找方法。

参数：

handler 处理回调函数。

instance 指定要查找的服务的实例的标识符。

返回值：用于停止查找服务。

1. static void StopFindService(FindServiceHandle handle)

功能：停止查找服务。

参数：

handle StartFindService的返回值。

返回值：无。

5） bool Init(Configuration\* conf)

功能：初始化。

参数：

conf service instance manifest。

返回值：true初始化成功, false初始化失败。

## Method类

1. Future<Output> operator()(TypeInputParameter1 input1, TypeInputParameter1 input2…)

功能：方法类的操作符()，用于方法的调用。

参数：作为输入参数，操作符()应取所提供方法的相应输入参数。

返回值：包装输出参数的对象。

## PublishEvent类

PublishEvent是模板类，类型T是模板类型

1. void Send(T& data)

功能：发送应用程序负责数据的事件。

参数：

data 要发送的事件数据

返回值：无。

1. void Send(SampleAllocateePtr<T> data)

功能：发送通信管理负责数据的事件。

参数：

data 要发送的事件数据。

返回值：无。

1. SampleAllocateePtr<T> Allocate()

功能：创建数据内存的方法。

参数：无。

返回值：数据内存。

## SubscribeEvent类

SubscribeEvent类是模板类，类型T是模板类型

1. void Subscribe(EventCacheUpdatePolicy policy, length\_t cacheSize)

功能：订阅服务事件的方法。

参数：

policy 缓存更新的政策

cacheSize 缓存事件数据的数量

返回值：无。

1. void Unsubscribe()

功能：取消订阅服务事件。

参数：无。

返回值：无。

1. void Update(FilterFunction<T> filter)

功能：更新接收的事件到事件缓存。

参数：

filter 决定是否存储或删除事件。

返回值：无。

1. SampleContainer<SamplePtr<const T>>& GetCachedSamples()

功能：获取当前缓存的事件数据。

参数：无。

返回值：包含事件缓存中存储的事件的容器。

1. void Cleanup()

功能：从事件缓存中删除所有事件。

参数：无。

返回值：无。

1. void SetReceiveHandler(EventReceiveHandler handler)

功能：使应用程序在收到事件时触发处理回调。

参数：

handler 处理回调函数。

返回值：无

1. void UnsetReceiveHandler()

功能：在接收事件时禁用应用程序的触发处理回调。

参数：无。

返回值：无。

# 库/数据库文件描述

介绍模块使用的库/数据库等文件及调用关系。

本模块使用SOME/IP、SOME/IP-SD相关的动态库。

# 消息机制

CM功能模块实现基于SOME/IP、SOME/IP-SD协议，其中SOME/IP协议是基于UDP和TCP传输协议，SOME/IP-SD是基于UDP传输协议。CM在本地进程之间还支持IPC的通信方式，Unix domain socket作为本地进程的IPC通信方式。

# 代码文件

头文件：

types.h

ServiceSkeleton.h

ServiceProxy.h

Method.h

PublishEvent.h

SubscribeEvent.h

ServiceProvider.h

ServiceRequester.h

BaseNetworkBinding.h

SomeIpBinding.h

IpcBinding.h

SomeIpSd.h

ManagementFactory.h

Configuration.h

Future.h

Promise.h

C++文件：

ServiceSkeleton.cpp

ServiceProxy.cpp

Method.cpp

PublishEvent.cpp

SubscribeEvent.cpp

ServiceProvider.cpp

ServiceRequester.cpp

BaseNetworkBinding.cpp

SomeIpBinding.cpp

IpcBinding.cpp

SomeIpSd.cpp

ManagementFactory.cpp

Configuration.cpp

Future.cpp

Promise.cpp