|  |  |
| --- | --- |
| **名称:** | **UART软件架构及概要设计说明书** |

|  |  |
| --- | --- |
| **项目:** | 高合汽车Figure项目 |

|  |  |  |
| --- | --- | --- |
| 作者 | | |
| **角色** | **姓名** | **日期和签名** |
| 软件工程师 | 黄鹏飞 | 2022-08-12 |
| 审核 | | |
| **角色** | **姓名** | **日期和签名** |
| 软件架构师 |  |  |
| 发布 | | |
| **角色** | **姓名** | **日期和签名** |
| 项目经理 |  |  |

|  |
| --- |
| **目的:** |
| 本文档描述UART软件架构与概要设计。 |

|  |
| --- |
| **范围 （有效区域）:** |
| 本文档适用于UART开发人员。 |

|  |
| --- |
| **读者:** |
| 本文的目标读者是软件开发人员。 |

**变更履历**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本 | 状态 | 日期 | 姓名 | 部门 | 变更 |
| a01 | 初稿 | 2022-08-12 | 黄鹏飞 | 智能汽车事业部 | 初版 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目录

[1 概述 4](#_Toc16025)

[1.1 文档概述 4](#_Toc8108)

[1.2 术语和定义 4](#_Toc5362)

[1.3 名词解释 4](#_Toc30867)

[2 软件架构描述 5](#_Toc7297)

[2.1 整体架构 5](#_Toc14512)

[2.2 软件功能描述 6](#_Toc12008)

[3 UART 6](#_Toc993)

[3.1 Uart Stack模块说明 6](#_Toc22087)

[3.1.1 模块描述 6](#_Toc17048)

[3.1.2 物理层 6](#_Toc3276)

[3.1.3 应用层/数据帧格式 7](#_Toc5032)

[3.1.4 应用层/应答帧格式 8](#_Toc21980)

[3.1.5 发送应答流程 8](#_Toc13656)

[3.2 UART Native 模块 9](#_Toc25929)

[3.2.1 模块描述 9](#_Toc26674)

[3.2.2 接口描述 9](#_Toc28021)

[3.3 UART Framework模块 10](#_Toc8488)

[3.3.1 模块描述 10](#_Toc13136)

[3.3.2 接口描述 10](#_Toc19970)

[4 流程或场景描述 11](#_Toc16280)

[4.1 UART Native Service启动流程 11](#_Toc17982)

[4.2 注册回调流程 11](#_Toc32453)

[4.3 消息发送与接收流程 13](#_Toc5049)

# 概述

## 文档概述

本文档为UART的软件架构、概要设计说明书。本软件按照功能划分，分为以下几大模块：

* + UART-Service模块，实现加载UART协议栈，开机自启动，提供Framework调用接口。
  + UART-Stack 模块，实现与MCU之间的通信，以及对数据帧的封装和解析。
  + UART-framwork模块，实现获取UART-Servic服务，同时提供上层应用进行调用的接口。

## 术语和定义

图表 1 术语和定义

|  |  |
| --- | --- |
| **术语** | **定义** |
| UART | Universal Asynchronous Receiver/Transmitter  通用异步收发器 |
| HAL | Hardware Abstraction Layer  硬件抽象层 |

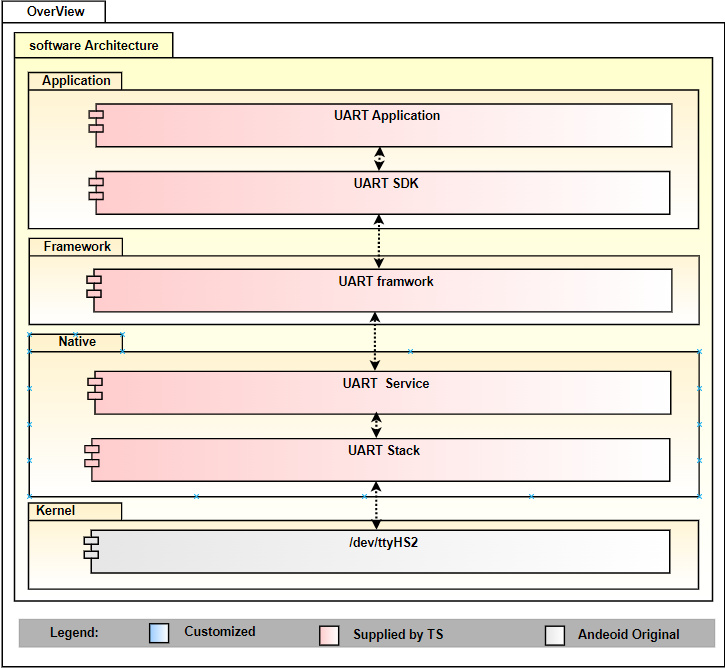
## 名词解释

图表 2 名词解释

|  |  |
| --- | --- |
| **名词** | **解释** |
| UART Service | 基于andorid架构的hal服务，在系统启动时自启动 |

# 软件架构描述

## 整体架构



图表 3 软件架构图

## 软件功能描述

图表 4 软件模块说明

|  |  |
| --- | --- |
| 软件单元 | 功能描述 |
| Application | 应用层程序,调用 SDK 接口完成定义的业务 |
| UART SDK | SDK 接口文件 |
| UART Framework | 负责封装 C++ Middleware 层提供的 UART基础服务 |
| UART Native Service | UART 管理模块, 是一个守护进程. 为上层 APP 提供服务, 同时负责维护 UART 协议栈连接. |
| UART Stack | UART协议栈，提供数据封装，解析，传输 |

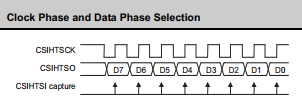
# UART

## Uart Stack模块说明

### 模块描述

Uart Stack模块负责与MCU进行基于串口的通信，针对数据帧进行封装及解析。

### 物理层



图表 5 物理层报文格式图

* 起始位：

开始进行数据传输时发送方要先发出一个低电平’0’来表示传输字符的开始。因为空闲位一直是高电平所以开始第一次通讯时先发送一个明显区别于空闲状态的信号即为低电平。

* 数据位：

起始位之后就是要传输的数据，数据是8位。先发送最低位最后发送最高位。

* 奇偶校验位：

偶校验（even parity）：如果数据为中’1’的数目是偶数，则校验位为’0’，如果为奇数，校验位为’1’。

* 停止位：

数据结束标志，1位的高电平。

### 应用层/数据帧格式

图表 6 数据帧格式

|  |  |  |  |
| --- | --- | --- | --- |
| **字节偏移** | **值** | **内容** | **描述** |
| 0 | 0x3C | 帧同步 | 表示一帧的起始 |
| 1 | 0x04~0xFF | 帧长度LenF | 整个通讯帧的长度,两字节，最大1033字节（为了固件升级时固件长度为 1024 字节），先低字节后高字节。例如，长度为 258 (0x0102)，先发送 0x02，后发送 0x01 |
| 2 | 0x00~0x02 | 帧类型 | 见下方 |
| 0x10~0x12 |
| 3 | 0xMM | 应用ID | 应用层ID |
| 4 ~ LenF-2 | 0xNN | 应用Data | 应用层命令参数或数据内容 |
| LenF-1 | 0xZZ | 帧校验 | 帧数据校验 = 帧同步^帧长度^帧类型^应用ID^应用Data |

图表 7 帧类型

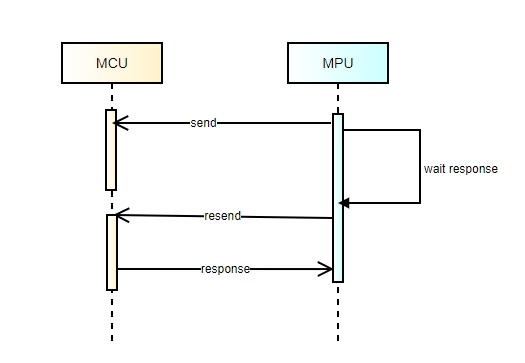
|  |  |
| --- | --- |
| 帧类型 | |
| **值** | **内容** |
| 0x00 | MCU到SOC的信息返回 |
| 0x01 | MCU到SOC的信息读 |
| 0x02 | MCU到SOC的命令写 |
| 0x10 | SOC到MCU的信息返回 |
| 0x11 | SOC到MCU的信息读 |
| 0x12 | SOC到MCU的命令写 |

### 应用层/应答帧格式

|  |  |  |  |
| --- | --- | --- | --- |
| **字节偏移** | **值** | **内容** | **描述** |
| 0 | 0x3E/0x3F | 帧应答 | 0x3E:表示一帧正常接收，0x3F:校验失败，需要重传,接收方应答帧的响应时间：<10ms |

应答帧是收到数据之后单独回一帧数据，保证相关数据可以收到，否则发送方应该重传，重传三次之后默认发送成功。

### 发送应答流程



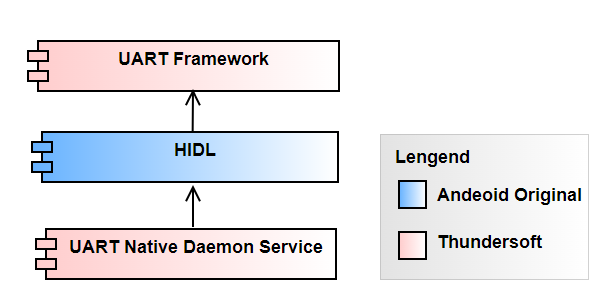
图表 8 发送应答流程

发送方只有在收到应答后才会发送下一个数据帧；如果发送方在发送完成后10ms内未接收到应答，则重新发送，共三次失败后认为发送完成。对于发送方，如果发送前有需要应答的接收，则先发送应答信息，然后再发送自己的报文。

## UART Native 模块

### 模块描述

UART Service的功能使用C++实现，为了让Framework层可以使用协议栈的接口，封装了一个Native层的服务。



图表 9 UART Native Service模块图

### 接口描述

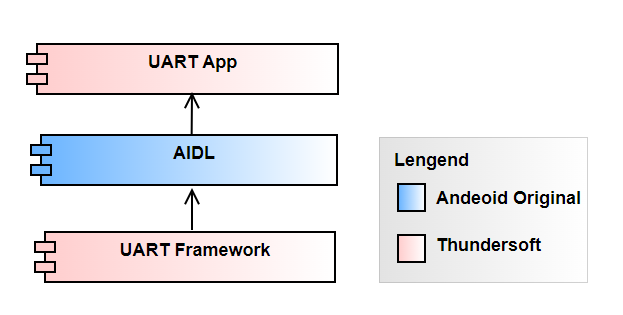
图表 10 UART Native Service 接口

|  |  |  |
| --- | --- | --- |
| **接口名称** | **接口参数** | **说明** |
| registerUartCallback | (app\_id id, string tag, IUartCallback callback) | 注册回调函数，用于MCU消息通知到上层应用 |
| sendMessage | (handle\_t hdl, UartMessage msg) | 用于上层应用主动向MCU端发起请求 |
| onUartMessage | (UartMessage msg) | 用于回调的接口 |

## UART Framework模块

### 模块描述

UART Framework的功能使用Java实现，负责封装 c++ Middleware 层提供的 UART基础服务。



图表 11 UART Framework模块图

### 接口描述

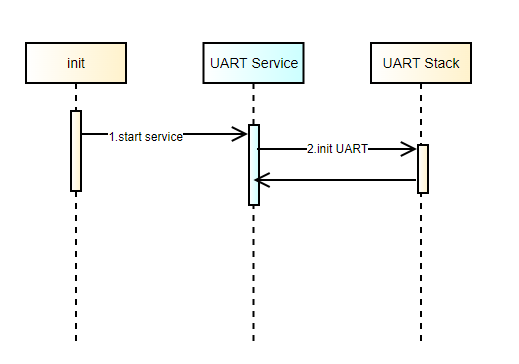
图表 12 UART Framework 接口

|  |  |  |
| --- | --- | --- |
| **接口名称** | **接口参数** | **说明** |
| registerUartCallback | (byte id, String tag, IUartServiceCallback callback) | 注册回调函数用于底层消息通知到上层应用 |
| sendMessage | (int hdl, byte type,in byte[] data) | 用于上层应用主动向MCU端发起请求 |
| onRegisterUartFinished | (boolean isregister, int sid) | 用于回调通知服务是否ready |
| onUartMessage | (byte type,in byte[] data) | 用于回调函数的接口 |

# 流程或场景描述

本章节我们会更具需求描述一些主要的流程和场景。

## UART Native Service启动流程



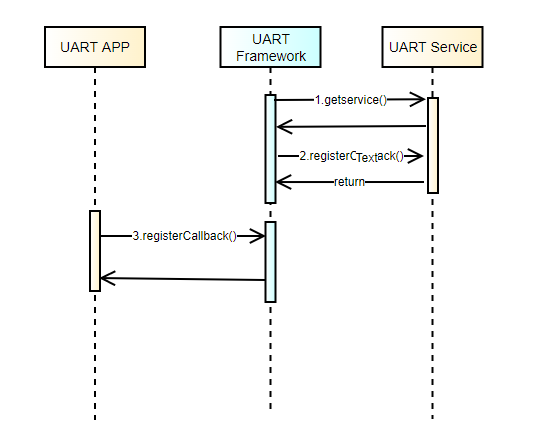
图表 13 UART Native Service 启动流程

流程说明：

1.系统的 Init 进程根据 rc 文件中的配置启动 UART Service服务

2.UART Service会初始化UART Stack，整个服务启动完成

## 注册回调流程



图表 14 注册回调流程

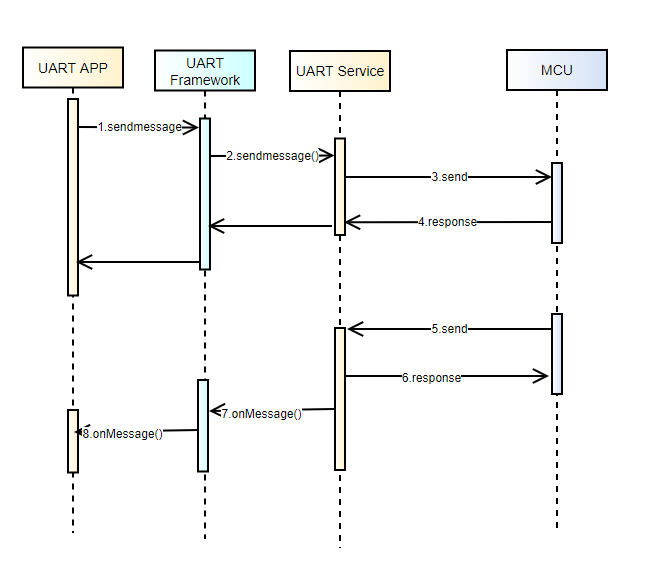
流程说明：

1.UART Framework层启动时会等待去获取Native的服务

2.获取成功之后进行相关的注册

3.上层APP通过SDK接口进行注册回调接口

## 消息发送与接收流程



流程说明：

1.APP发起消息通过UART Framework层进行透传到UART Service层，通过串口传输给MCU，MCU进行相关的数据校验之后返回应答

2.MCU发送消息到MCU，UART Stack先进行数据的校验，校验之后立马响应，然后回调通知上层应用