

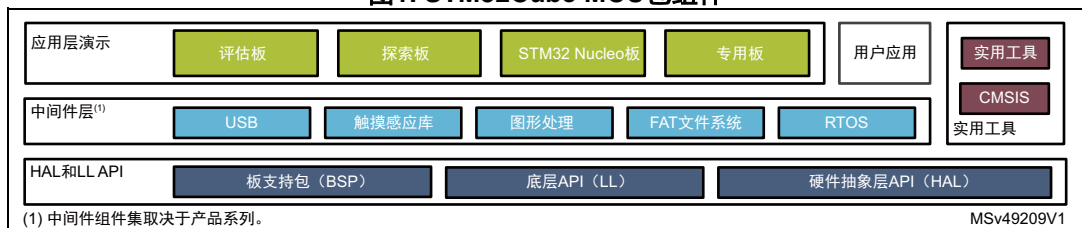
### 引言

STM32Cube™由意法半导体最初发起，通过减少开发工作量、时间和成本，让开发人员的生活更轻松。STM32Cube涵盖整个STM32产品系列。

STM32Cube 包括：

- 图形软件配置工具STM32CubeMX，可通过图形向导生成初始化C代码。
- 每个STM32微控制器系列都提供了全面的STM32Cube MCU封装（例如STM32F4系列的STM32CubeF4），包括：
  - STM32抽象层嵌入式软件STM32Cube HAL，确保用户应用在STM32各个产品之间实现最大限度的可移植性。HAL对所有外设可用
  - 底层API（LL）提供了一个专家级的快速轻量级层，它比HAL更靠近硬件，底层API只存在于一部分外设中。
  - 一套一致的中间件，比如RTOS、USB、TCP/IP和图形
  - 提供了一套完整示例以及嵌入式软件工具

**图1. STM32Cube MCU包组件**



此外，STM32Cube扩展包包含嵌入式软件组件，用于补充STM32Cube MCU包的功能或支持各种应用领域中大量ST器件和最合适的STM32微控制器的使用，或同时具有以上两种作用。

本档描述STM32Cube扩展包开发的兼容性要求，这些要求确保与STM32Cube MCU包和工具的正确匹配和STM32Cube生态系统内部的整体一致性，并能基于经过证明和验证的软件组件快速进行应用开发。

本档的读者必须熟悉STM32Cube架构、HAL和LL API以及编程模型。ST网站 [www.st.com](http://www.st.com) 上的STM32Cube MCU包页面提供了完整的文档集合。



## 目录

<b>1</b>	<b>概述</b> .....	<b>5</b>
<b>2</b>	<b>参考文献和缩略语</b> .....	<b>5</b>
<b>3</b>	<b>STM32Cube MCU包和STM32Cube扩展包</b> .....	<b>6</b>
	3.1 STM32Cube MCU包 .....	6
	3.2 STM32Cube扩展包 .....	7
<b>4</b>	<b>封装要求</b> .....	<b>8</b>
	4.1 使用STM32CubeMX开发示例 .....	8
	4.2 STM32Cube MCU包驱动程序和中间件的扩展 .....	10
<b>5</b>	<b>集成新的中间件</b> .....	<b>12</b>
	5.1 要求 .....	12
	5.2 组织结构 .....	12
	5.3 以目标格式传输 .....	15
<b>6</b>	<b>软件质量要求</b> .....	<b>16</b>
<b>7</b>	<b>版本历史</b> .....	<b>17</b>

## 表格索引

表1.	缩略语列表 .....	5
表2.	文档版本历史 .....	17
表3.	中文文档版本历史 .....	17

# 图片索引

图1.	STM32Cube MCU包组件 .....	1
图2.	STM32Cube MCU包内容 .....	7
图3.	STM32Cube扩展包架构和内容 .....	9
图4.	STM32Cube MCU驱动程序和中间件扩展示例 .....	11
图5.	STM32Cube MCU包中间件的组织结构 .....	13
图6.	取决于STM32核心的中间件CPU接口文件 .....	14
图7.	具有STM32Cube MCU驱动程序的中间件接口文件 .....	14

# 1 概述

STM32Cube MCU包和STM32Cube扩展包在基于Arm® Cortex®-M处理器的STM32 32位微控制器上运行。



# 2 参考文献和缩略语

开发STM32Cube扩展包时使用了以下文档（可在[www.st.com](http://www.st.com)上找到）：

1. STM32Cube扩展包开发清单（UM2312）

表 1给出了相关的缩略语定义，帮助您更好地理解本文档。

表1. 缩略语列表

术语	定义
API	应用编程接口
BSP	板级支持包
CMSIS	Cortex®微控制器系统接口标准
DHCP	动态主机配置协议
FTP	文件传输协议
HAL	硬件抽象层
HTTP	超文本传输协议
LL	底层
TCP/IP	传输控制协议/互联网协议
TLS/SSL	传输层安全/安全套接层

## 3 STM32Cube MCU包和STM32Cube扩展包

STM32Cube解决方案由STM32CubeMX和STM32Cube MCU包组成，前者是工具部分，后者提供利用STM32微控制器功能所需的软件模块。

除了STM32CubeMX和STM32Cube MCU包，STM32Cube扩展包提供的补充性插件进一步丰富了整个STM32Cube生态系统。

### 3.1 STM32Cube MCU包

STM32Cube MCU包（例如用于STM32F4系列微控制器的STM32CubeF4）提供使用STM32微控制器硬件功能所需的所有必要的软件模块。

STM32Cube MCU包主要包含：

- STM32外设驱动程序
  - HAL（硬件抽象层）
  - LL（底层API）
- 涵盖STM32外设集的中间件
  - RTOS、TCP/IP、TLS/SSL、USB、Graphics、File System、JPEG和其他
- STM32板的外部组件驱动程序（BSP）
  - 评估板
  - 探索套件
  - Nucleo板
- 丰富的示例，用于演示STM32硬件和相关嵌入式软件的功能和使用

STM32Cube嵌入式软件按照[www.st.com](http://www.st.com)网站上所述的*Mix Liberty + OSS + 第三方V1*混和许可模型进行分发。

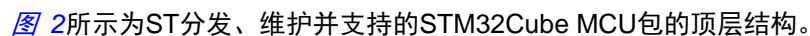
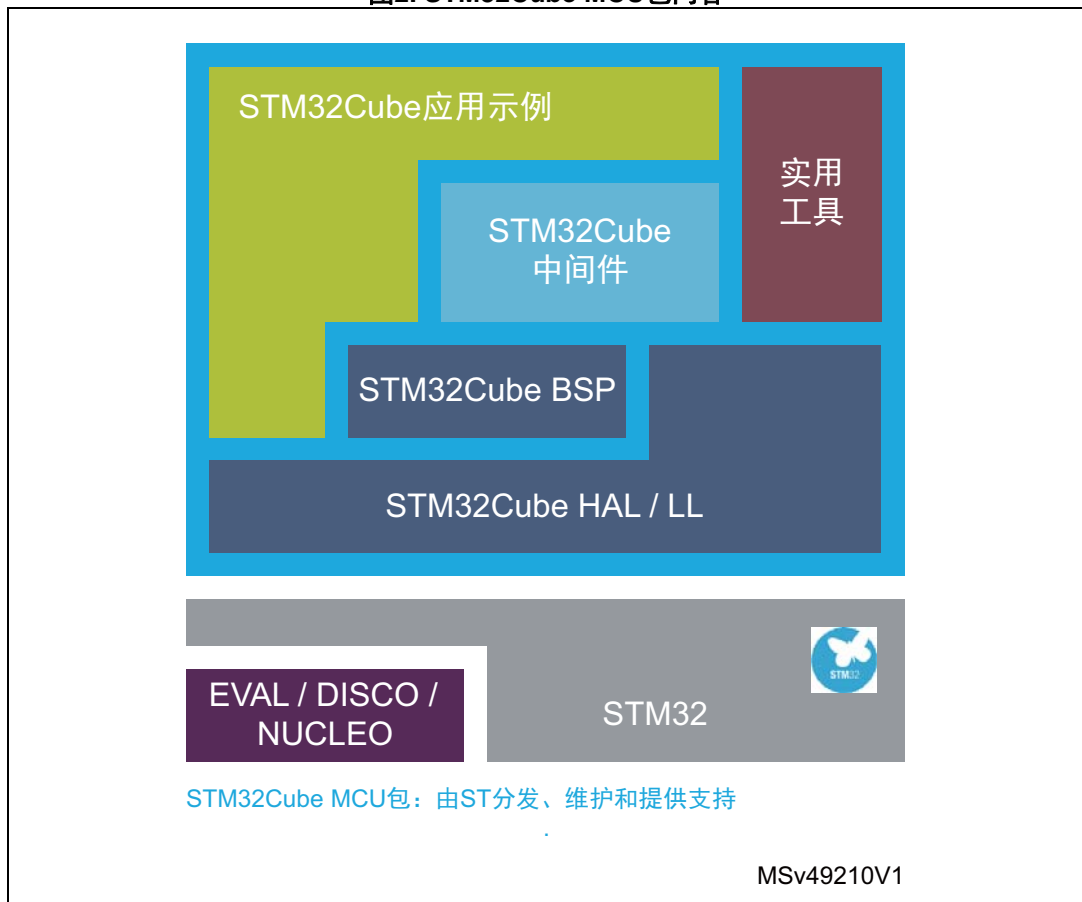
图 2所示为ST分发、维护并支持的STM32Cube MCU包的顶层结构。

图2. STM32Cube MCU包内容



### 3.2 STM32Cube扩展包

STM32Cube扩展包包含插件式的软件组件，补充了STM32Cube MCU包的功能：

- 新的中间件堆栈
- 新的硬件和板支持（BSP）
- 新的示例
- 以上项目中的几个

有两类扩展包：一些STM32Cube扩展包由ST开发、维护并提供支持，其他一些则由不隶属于ST的第三方开发、维护并分发。

## 4 封装要求

STM32CubeMCU包是所有STM32Cube扩展包的主干。因此，务必以不修改原始文件夹结构（如[图 3第 9页](#)所示）的方式管理文件夹和文件结构。

文档*STM32Cube扩展包开发清单*中提供了对STM32Cube扩展包内容的具体要求[1]。

### 4.1 使用STM32CubeMX开发示例

在STM32Cube扩展包中，应使用STM32CubeMX工具开发示例。此要求意味着开发符合以下规则：

- 应使用STM32CubeMX工具配置STM32器件和板，并生成相应的初始化代码
- 在生成的\*.h和\*.c源文件中，用户应在使用/\* USER CODE BEGIN \*/和/\* USER CODE END \*/标记限定的部分中添加应用代码
- 应在示例根目录中提供相关\*.ioc文件
- 如果使用了额外的STM32CubeMX项目设置，应将.extSettings文件保存在\*.ioc文件的同一级

\*.ioc文件的位置和命名规则如[图 3](#)所示。



**图3. STM32Cube扩展包架构和内容**


ST板BSP驱动程序  
重复使用，来自STM32Cube MCU包。  
用户不得修改。

用户示例在\Projects\<USER\_BOARD\_REF>下添加且分类如下：

- 示例（只使用HAL和BSP）
  - 应用（使用中间件）
  - 演示（使用HAL、BSP和中间件组件开发）
- 可以使用ST或用户定制板。

用户示例的文件夹结构如下：

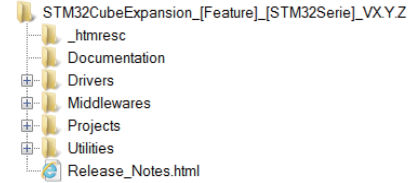
- \Inc为头文件文件夹
- \Src为源文件文件夹
- \<Toolchain-Name>工具链预配置项目。必须删除所有临时文件。
- ApplicationN\_Name.ioc: STM32CubeMX项目文件
- extSettings: STM32CubeMX项目额外设置文件（可选，如果有）
- \Binary: 包含二进制文件，文件名的格式为：“USER\_BOARD\_REF\_ApplicationN\_Name\_VX.Y.Z.bin”
- .mxproject文件：由STM32CubeMX自动生成
- readme.txt: 描述示例行为和示例的运行环境

注释：

- USER\_BOARD\_REF: 表示板名称或参考
- ApplicationCommon\_Name: 表示类的根文件夹名称或应用的全局范围。例如：USB\_PD、Cloud、WiFi或其他
- ApplicationN\_Name: 表示子项目的确切名称。例如：对于WiFi应用，HTTP\_Server和Server\_Client应用就是此类子项目。
- 建议定义便于用户识别的应用名称，避免使用缩略语和意义模糊的简称。

\Media: 包含所有媒体文件（音频、视频、图像及其他文件）。  
必须提供readme文件，它为使用的每个媒体文件提供版权/许可说明。

**折叠的软件包**



STM32 CMSIS、HAL和LL驱动程序  
重复使用，来自STM32Cube MCU包。  
用户不得修改。

ST拥有或许可进行维护的中间件  
重复使用，来自STM32Cube MCU包。  
用户不得修改。

来自第三方和开源社区的中间件

PC\_Software: 包含所有软件工具（任何在PC兼容的平台上运行的应用），但与中间件堆栈相关的软件工具除外（例如用于配置堆栈的工具），它们必须在中间件存储库中提供。

图例： ■ 继承自STM32Cube MCU包的软件组件

■ 来自第三方或专用于已开发的STM32Cube扩展包的软件组件

## 4.2 STM32Cube MCU包驱动程序和中间件的扩展

在某些特殊情况下，STM32Cube MCU包中提供的原生驱动程序（HAL/LL、CMSIS和BSP）或中间件或这两者可能需要由STM32Cube扩展包的开发人员进行更新（例如，为了扩展实现的功能，在ST正式发布前修复问题等）。

这种情况下，用户应执行如下操作：

1. 在\ApplicationN\_Name存储库下创建\Extension文件夹，应将要打补丁的文件复制到该文件夹中。  
此复制操作必须遵循文件夹的层次结构。如果保留默认的文件夹层次结构会导致路径过长，可以任何方式减少子文件夹的数量。
2. 通过添加\_patch后缀重命名要打补丁的文件。
3. 在项目中使用已打补丁的文件而不是原始文件。

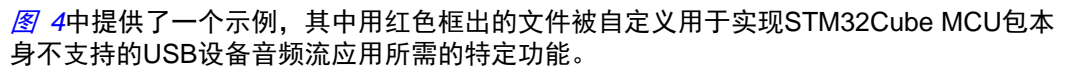
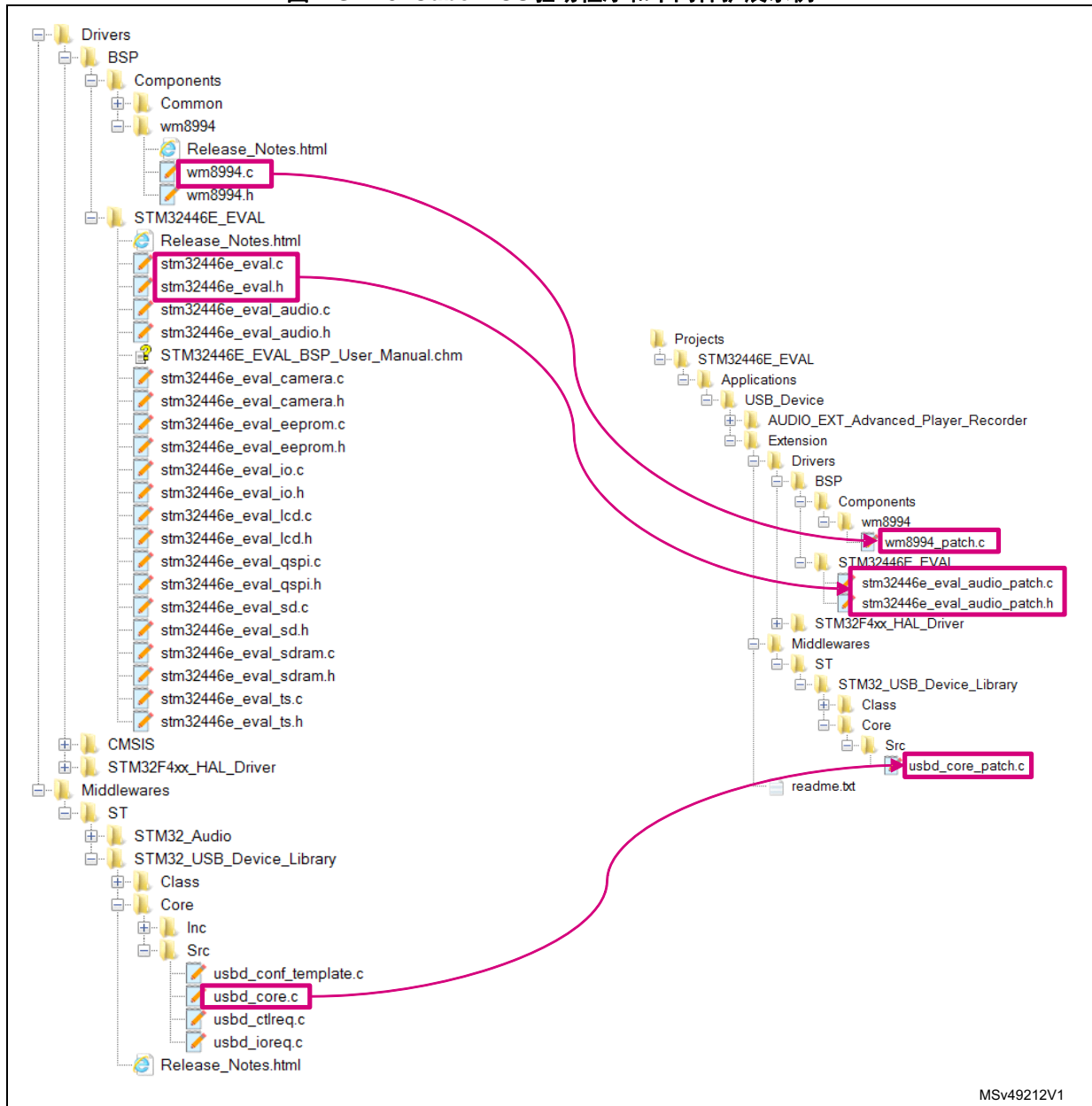
图 4中提供了一个示例，其中用红色框出的文件被自定义用于实现STM32Cube MCU包本身不支持的USB设备音频流应用所需的特定功能。

图4. STM32Cube MCU驱动程序和中间件扩展示例



## 5 集成新的中间件

### 5.1 要求

中间件组件是介于STM32硬件与用户应用之间的固件层。任何新的中间件组件基本上都必须符合以下要求：

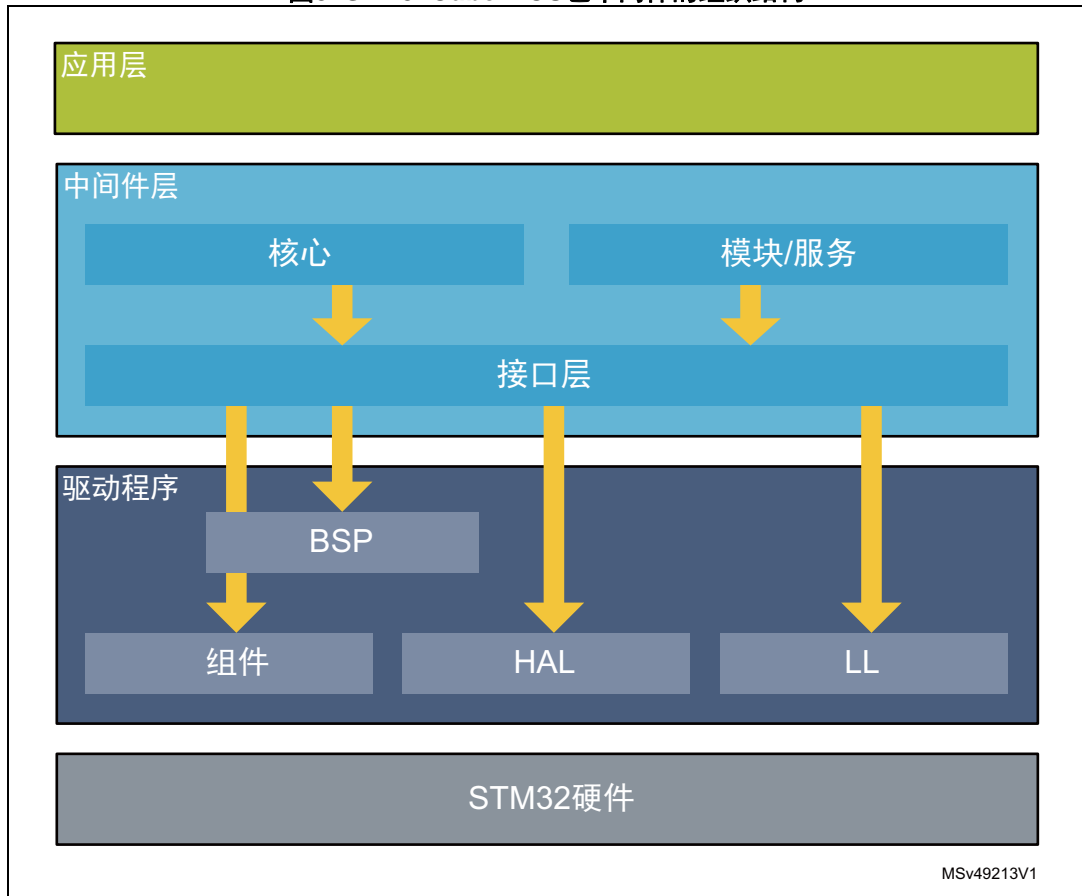
- 必须实现模块化架构，通过分解成较简单的组块或文件来降低复杂性。
- 必须独立于硬件（器件和板）。
- 应通过接口文件实现与硬件驱动程序和其他中间件堆栈的连接。
- 接口文件应作为中间件文件夹中的模板提供，将由用户进行自定义。

### 5.2 组织结构

中间件组件主要包括：

- **模块**：模块是一个具有通用功能的子层，可以单独添加或删除。例如，TCP/IP栈由TCP/IP核心以及DHCP、HTTP和FTP组件构成。每个组件被视为一个模块。可以通过配置文件中的特定定义/宏添加或删除组件，从而将其使能或禁用。
- **核心**：这是组件的核心；它管理主库状态机，并管理各不同模块之间的数据流。
- **接口层**：接口层通常用于连接中间件核心组件与底层（例如HAL、LL和BSP驱动程序）。

图5. STM32Cube MCU包中间件的组织结构



根据中间件组件，接口层可能属于以下类别之一：

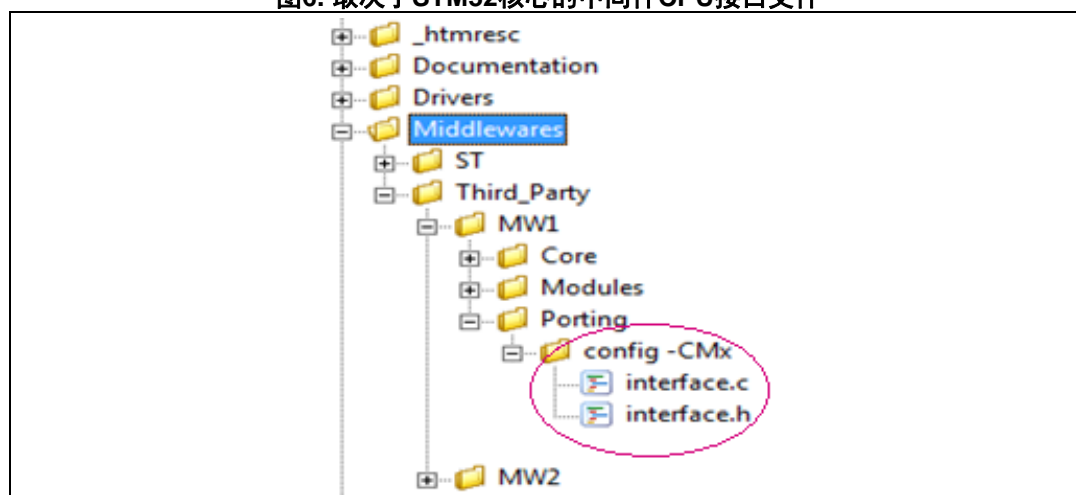
- 只取决于所用STM32核心的CPU接口
- HAL、LL和BSP驱动程序接口

只取决于所用STM32核心的CPU接口：这种情况下，接口文件应位于中间件目录中并被使用同一核心的所有应用使用。

以FreeRTOS™为例，它为每个Arm® Core（例如Cortex®-M3、Cortex®-M4或其他）和每个编译器提供移植文件，因为需要某些汇编器代码段来处理上下文切换和寄存器保存与恢复机制。

中间件结构中接口文件的位置如图 6所示。

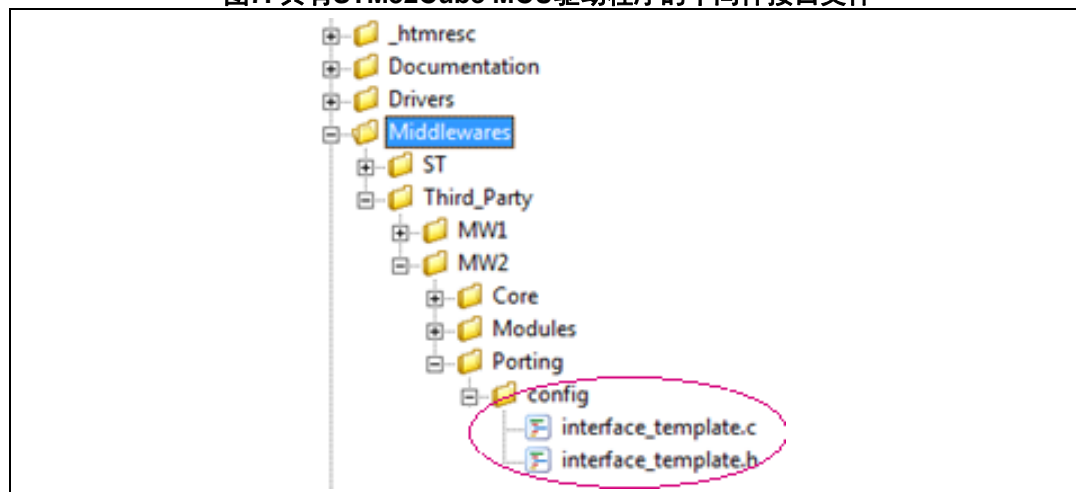
图6. 取决于STM32核心的中间件CPU接口文件



HAL、LL和BSP驱动程序接口：在某些中间件组件中，接口是一组空的或几乎为空的函数，需由用户填充以便将其与HAL和LL层相连接。接口文件通常随中间件组件一起以模板文件的形式提供。必须在应用层对它们进行复制和自定义。

中间件结构中接口文件的位置如图 7所示。

图7. 具有STM32Cube MCU驱动程序的中间件接口文件



每个子文件夹必须包含一个\Inc文件夹和一个\Src文件夹，头文件 (\*.h) 和源文件 (\*.c) 分别位于这两个文件夹中。

如果中间件组件十分简单，则可以省去子文件夹\Core、\Modules和\Porting，而只在中间件组件根目录中提供子文件夹\Inc和\Src。

## 5.3 以目标格式传输

在以二进制或目标格式传输中间件库时，必须至少满足以下要求：

- 应提供头文件，以将库接口API导出到最终应用
- 应提供版本说明
- 应以所有支持的编译器的目标格式提供库。如果库对象与编译器有关，必须在目标文件名称中指明支持的编译器。

例如，**LibraryNameV\_CMx\_C\_O.a**是一个库对象文件名，其中：

- **V**: 模块版本（例如，V=01表示版本为V0.1）
- **x**: CMx核心分类（CM0、CM3、CM4、CM7、CM23和CM33）
- **C**: 编译器（IAR™、Keil®和GCC）
- **O**: 指定编译器优化
- **<empty>**: 大尺寸优化
- **Ot**: 高速优化
- **Otnsc**: 无尺寸限制的高速优化
- **Ob**: 高度平衡优化

## 6 软件质量要求

在STM32Cube扩展包中开发的BSP驱动程序、中间件和项目（对于STM32Cube MCU包，为插件）必须至少满足以下要求：

- 务必在Windows<sup>®</sup> 和Linux<sup>®</sup> 平台上使用所有支持的编译器（EWARM、MDK-ARM和SW4STM32）执行编译，无错误或警告（仅当警告属于不归扩展包开发人员所有的软件组件时，才可以接受）。
- 执行的功能测试无已知问题残留（可以接受次要问题的存在，前提是在组件版本说明中予以记录），有证据报告。

BSP驱动程序和中间件必须符合这些额外要求：

- MISRA C<sup>®</sup>合规和静态代码分析，有证据报告。
- 如不符合MISRA C<sup>®</sup>，应证明偏离规则的合理性。



## 7 版本历史

表2. 文档版本历史

日期	版本	变更
2017年11月14日	1	初始版本。

表3. 中文文档版本历史

日期	版本	变更
2019年4月11日	1	中文初始版本。

**重要通知 - 请仔细阅读**

意法半导体公司及其子公司（“ST”）保留随时对 ST 产品和 / 或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。本文档的中文版本为英文版本的翻译件，仅供参考之用；若中文版本与英文版本有任何冲突或不一致，则以英文版本为准。

© 2019 STMicroelectronics - 保留所有权利