OTG IP主机Cube库介绍

2018年5月

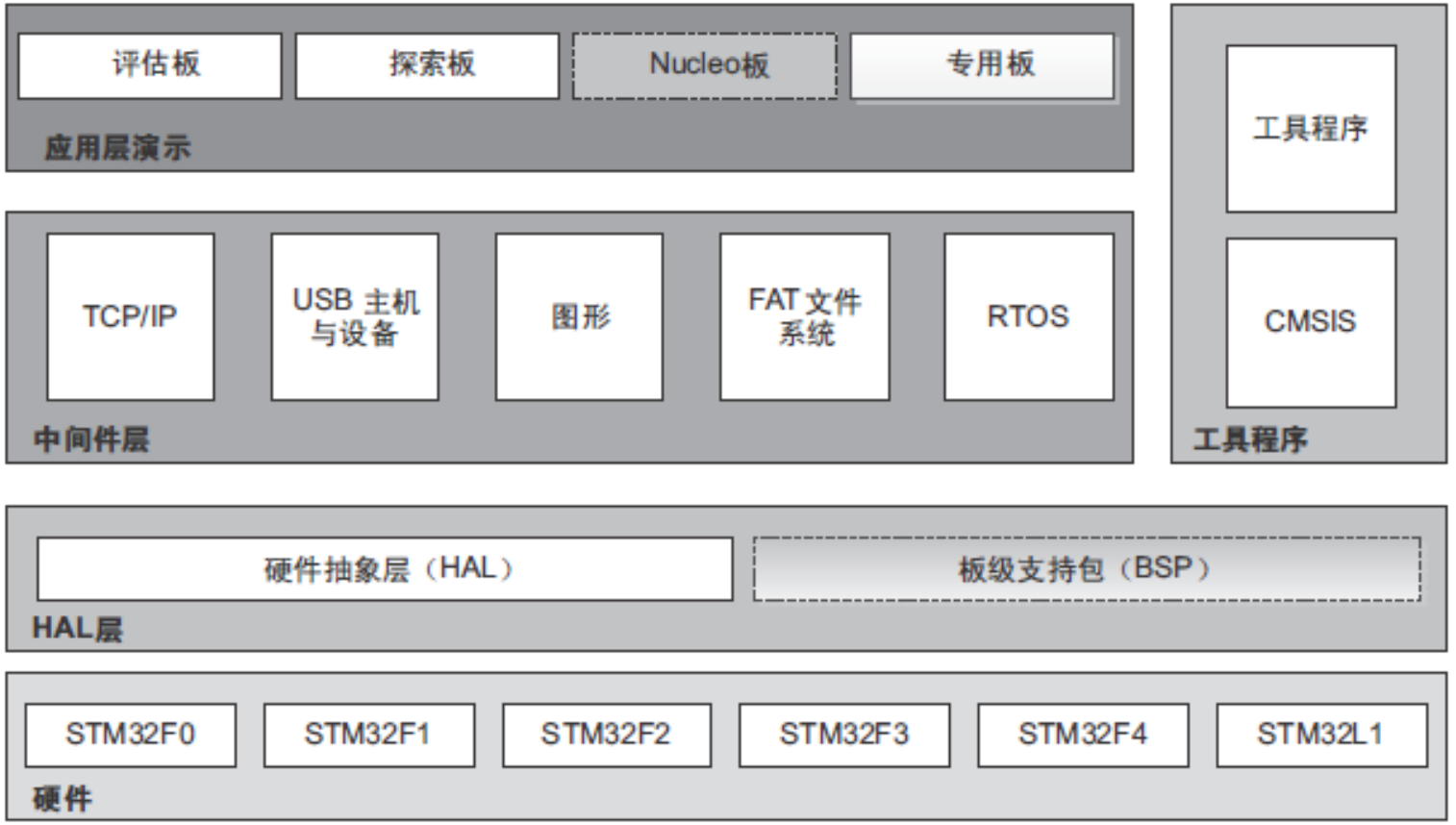
STM32CubeF4 UM1072 FS HID Host

life.augmented

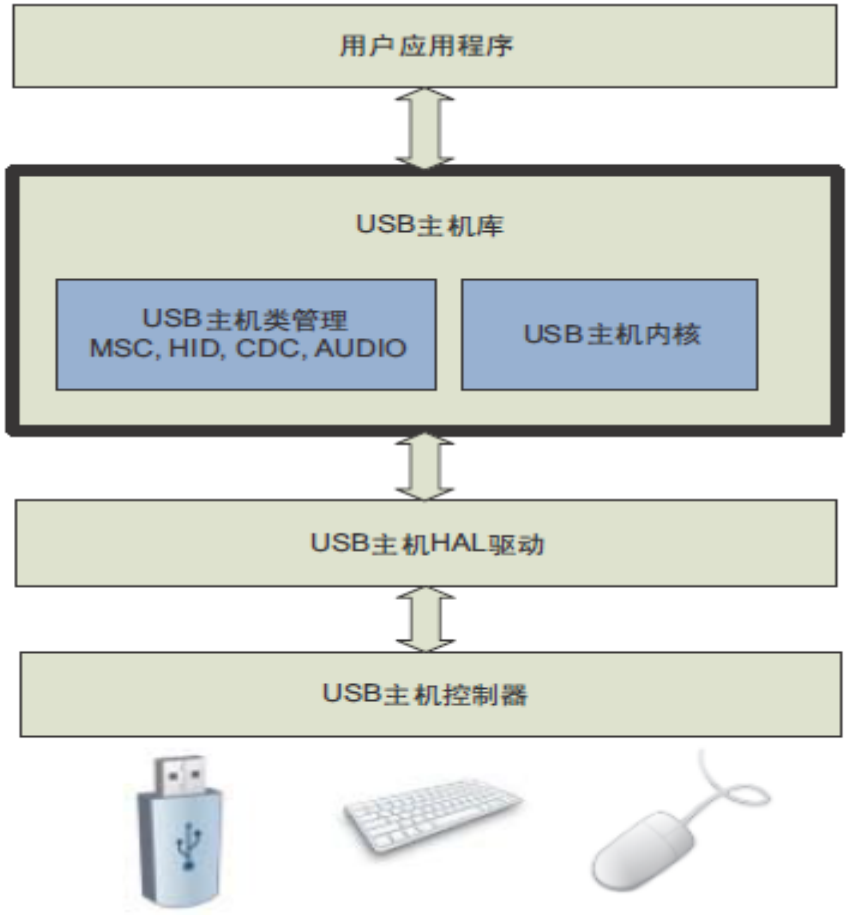
OTG IP主机Cube库讲解要点

- USB主机Cube库架构与文件组织
  - USB主机库架构
  - USB主机库文件组织
    - USB主机内核文件
    - USB主机类文件
- USB主机Cube库内核模块
  - 内核API，用户回调与数据结构
  - 内核状态机概述
  - 具备底层驱动的内核接口
- USB主机Cube库类模块
- Hands ON

STM32Cube™ 概述



STM32Cube USB 主机库



STM32Cube USB主机库概览

- 适用于所有STM32上的USB模块，只是通过HAL层的不同来适用于STM32系列上的不同USB模块
  - 基于STM32Cube USB 主机HAL 驱动之上
  - 提供开发USB主机应用所需的所有API

	F0	F1	F2	F3	F4	F7	L0	L1	L4
USB IP	FS	FS		FS			FS	FS	
OTG IP		FS	FS/ HS		FS/ HS	FS/ HS			FS

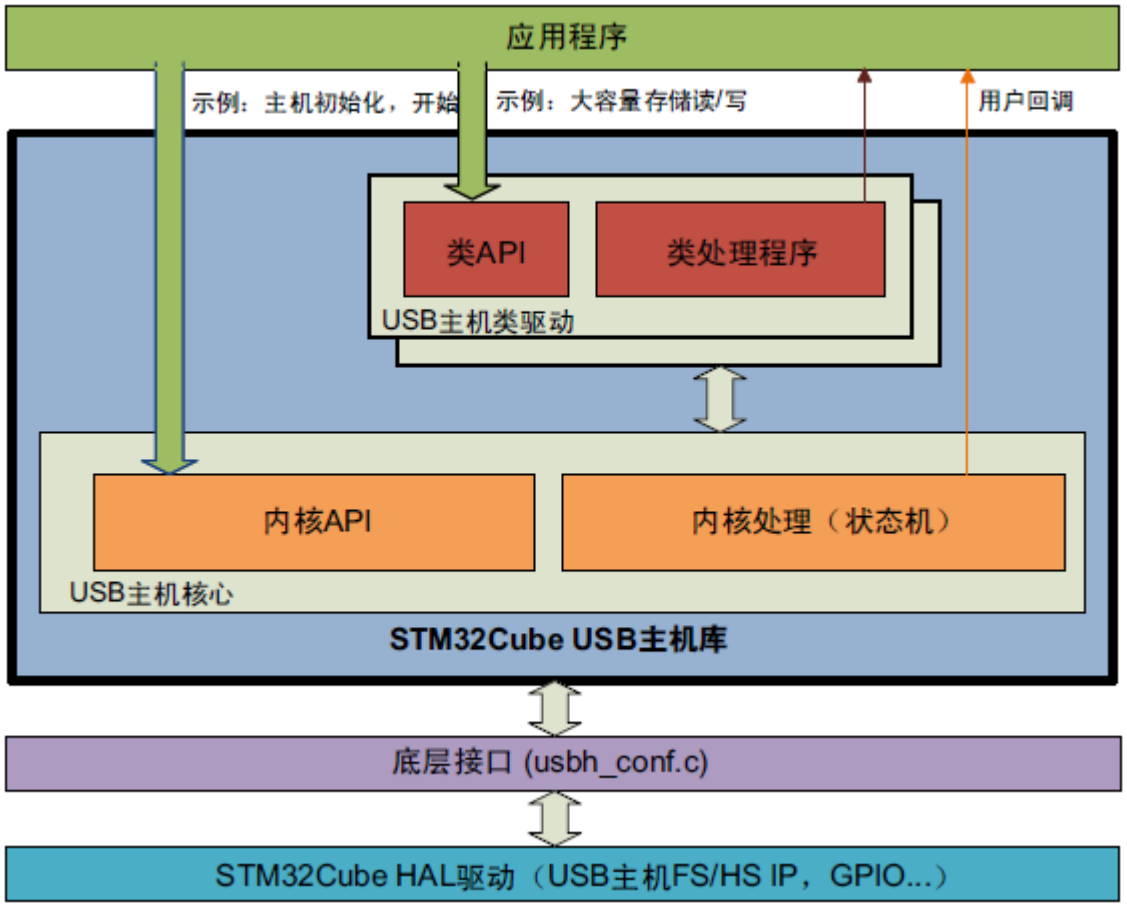
- 为各种传输类型都提供了样例程序

传输类型	项目例程	备注
BULK	MSC、CDC、DFU	Dual Core
INTERRUPT	CustomHID、HID_LPM、HID	
CONTROL	每个项目例程都会用到控制传输	
ISO	AUDIO	

STM32Cube_FW_F4_V1.21.0部分例程

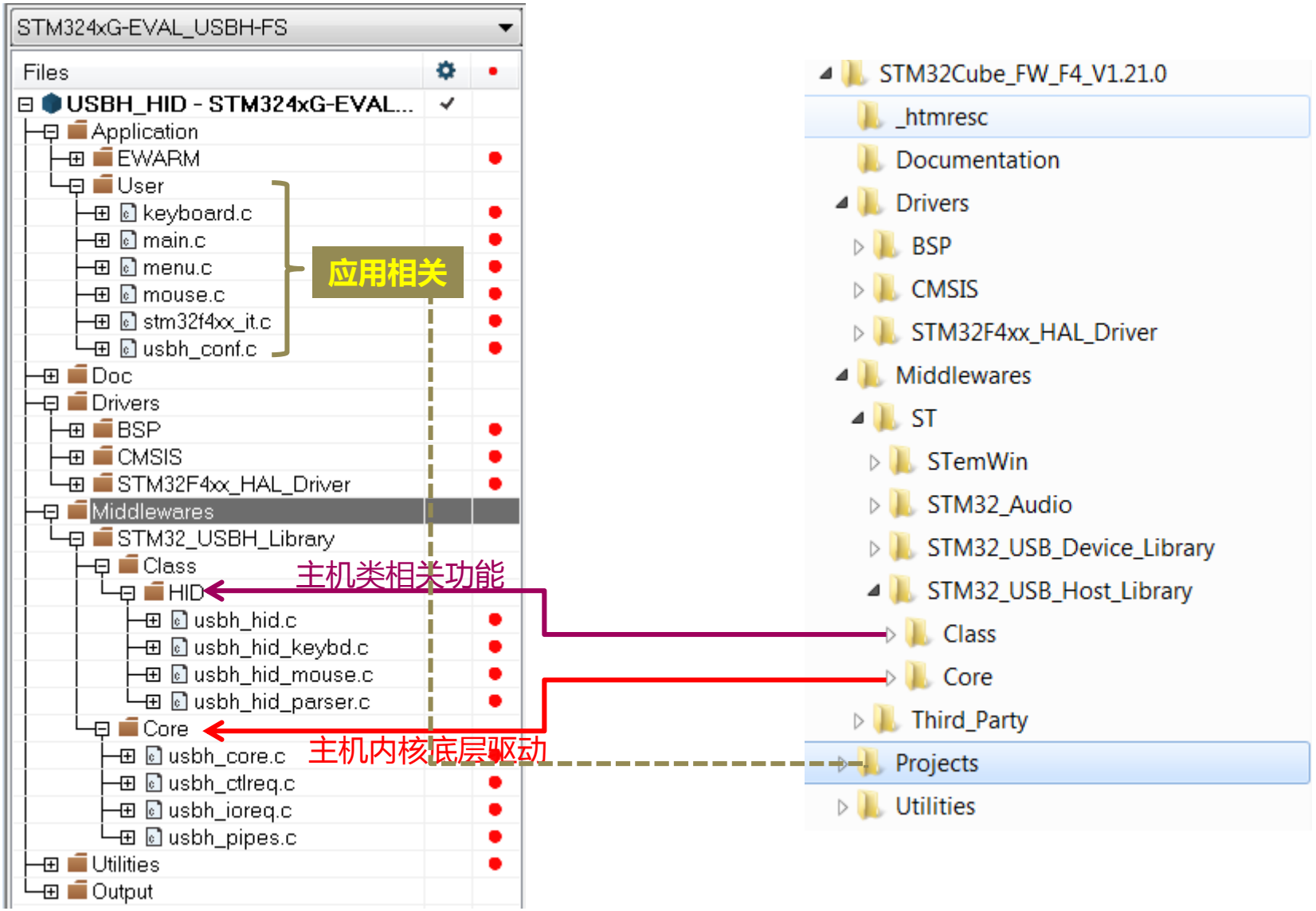
		USB Device		USB Host	
		Standalone	OS.	Standalone	OS.
4xG_EVAL	BULK	CDC/DFU/MSC		CDC/DFU/MSC/ **MTP**	MSC
	INT.	CustomHID/HID		HID	HID
	ISO.	Audio		Audio	
469I_EVAL	BULK	CDC/DFU/MSC		MSC	
	INT.	HID_LPM/HID		HID	HID
	ISO.	---			
4x9I_EVAL	BULK	CDC/DFU/MSC		CDC/DFU/MSC/MTP	MSC
	INT.	CustomHID/HID		HID	HID
	ISO.	Audio		Audio	

USB 主机库架构


应用程序
示例：主机初始化，开始
示例：大容量存储读/写
用户回调
类API
类处理程序
USB主机类驱动
内核API
内核处理（状态机）
USB主机核心
STM32Cube USB主机库
底层接口 (usbh_conf.c)
STM32Cube HAL驱动（USB主机FS/HS IP，GPIO...）

项目结构

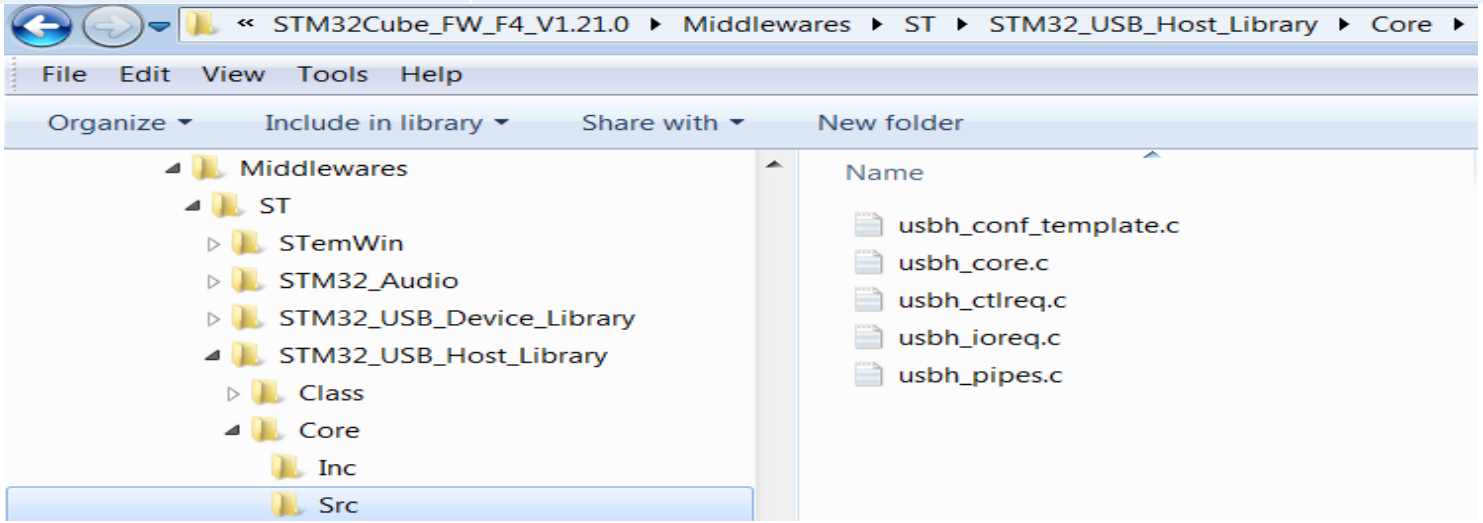
- 运行在STM324xG-EVAL板上的HID **Host**例程



USB OTG主机库文件组织

- USB主机内核文件，位于*Core*文件夹（*STM32_USB_Host_Library\Core*）

文件	说明
Usbh_core.c/h	主要的主机内核文件。 实现主机状态机 管理设备检测与枚举，控制类模块进行类操作。
Usbh_ctlreq.c/h	实现标准控制请求
Usbh_ioreq.c/h	USB 传输管理的 API（控制、批量、中断、同步）。
Usbh_pipes.c/h	管道控制的 API（例如分配、开启、关闭）
Usbh_conf_template.c/h	底层接口文件的模板文件，由**用户定制**，并包含在应用文件中
Usbh_def.h	**通用的**库定义



USB OTG主机库文件组织

- USB主机类文件，位于*Class*文件夹（*TM32_USB_Host_Library\Class*）

USB类	说明	
大容量存储	Usbh_msc.c	大容量存储类处理程序
	Usbh_msc_bot.c	专用批量传输（BOT）协议
	Usb_msc_scsi.c	SCSI指令
HID鼠标与键盘	Usbh_hid.c	HID类程序处理
	Usbh_hid_parser.c	HID描述符解析
	Usbh_hid_mouse.c	HID鼠标子类处理程序
	Usbh_hid_keybd.c	HID键盘子类处理程序
	Usbh_hid_usage.h	HID的通用定义
音频扬声器	Usbh_audio.c	音频类处理程序
CDC虚拟串口通信	Usbh_cdc.c	CDC虚拟通信端口处理程序
MTP类 (Media Transfer Protocol)	Usbh_mtp.c	MTP类处理程序
	Usbh_mtp_ptp	MTP类的PTP规范的实现

USB OTG主机内核模块

- **内核特性**
- 该USB主机内核具备以下主要特性：
- 和具体类无关的设备连接管理与枚举
- 基于状态机的结构，可在后台的主循环中运行或作为 RTOS 任务线程
- 采用用户事件回调，将关于设备连接 / 断连、错误状态等主机事件通知应用层
- 可重定向至任何接口（比如串行端口、 LCD）的用户事件记录
- 错误管理与报告

USB OTG主机内核模块

- 内核API，用户回调函数与数据结构，如下是用户应用程序API

函数	说明
USBH_Init	初始化主机栈和底层。驱动应在应用程序启动时调用
USBH_DeInit	清理主机栈变量，并运行底层清理 （比如关闭所有打开的管道，清除 中断标志）
USBH_RegisterClass	注册所支持的 USB 类处理程序。经过枚举后，主机检查当前的设备类 是否对应于注册的类
USB_ReEnumerate	通过重新初始化主机栈并强制实施 VBUS 的断连 / 连接，对设备进行重 新枚举。
USBH_Start	使能主机端口的 VBUS 电源，然后开始底层操作。
USBH_Stop	关闭主机端口的 VBUS 电源，然后停止底层操作。
USBH_GetActiveClass	设备枚举和类初始化之后，返回当前的活动 USB 类
USBH_Process	以单任务运作模式实现内核状态机的主机过程函数。该函数应在主循环 的后台中调用，以处理主机状态机。

USB OTG主机内核模块

- 主机内核通过调用用户回调函数，将USB事件传递给应用层。当调用 *USBH_Init* API时，该 函数会作为参数进行传递。用户回调函数的原型：void (*pUsrFunc)(USBH_HandleTypeDef * phost, uint8_t event)

内核用户回调事件	说明
HOST_USER_CONNECT	将设备连接的有关信息通知应用程序
HOST_USER_DISCONNECT	将设备断连的有关信息通知应用程序
HOST_USER_CLASS_ACTIVE	将类初始化过程结束的有关信息通知应用程序
HOST_USER_SET_CONFIGURATION	将设备标准枚举结束的有关信息通知应用程序
HOST_USER_CLASS_SELECTED	通知已找到受支持的类

USB OTG主机内核模块

- 类处理程序的内核API，专门用于类处理程序的内核API分为以下几类：
- • I/O 请求 API • 管道控制 API • 标准类控制请求 API • 接口功能 API

函数	类别	说明
USBH_BulkReceiveData	IO请求	接收批量传输的数据
USBH_BulkSendData		发送批量传输的数据
USBH_CtlReceiveData		接收控制传输的数据
USBH_CtlSendData		发送控制传输的数据
USBH_CtlSendSetup		通过控制传输发送命令包
USBH_InterruptReceiveData		接收来自中断管道的数据
USBH_InterruptSendData		将数据发送至中断管道
USBH_IsocReceiveData		接收来自同步管道的数据
USBH_IsocSendData		将数据发送至同步管道

USB OTG主机内核模块

- 类处理程序的内核API

函数	类别	说明
USBH_OpenPipe	管道控制	打开管道
USBH_ClosePipe		关闭管道
USBH_AllocPipe		分配新的管道
USBH_FreePipe		释放已分配的管道
USBH_GetDescriptor	标准控制请求	获取描述符的通用函数
USBH_SetInterface		标准控制请求，用于设置接口的复用设定值
USBH_FindInterface	接口功能	解析配置描述符，找寻与特定的类、子类和协 议相对应的接口描述符
USBH_FindInterfaceIndex		解析配置描述符，根据指定的接口编号和复用 设定值找寻接口描述符的索引序号

USB OTG主机内核模块

主要的主机内核数据结构体与枚举类型定义，在文件*usbh_def.h*中。主要数据结构体包括：

- *USBH_HandleTypedef* 类型的主机内核句柄结构体
- *USBH_DeviceTypedef* 类型的设备句柄结构体，
- *USBH_ClassTypedef* 类型的类句柄结构体。

USB OTG主机内核模块

- 主机库中所采用的主要结构体是USBH_handleTypedef类型的主机句柄

结构体成员	说明
gState	给出主机全局状态机的当前状态
EnumState	给出枚举状态机的当前状态
RequestState	给出控制请求的当前状态（IDLE、SEND 或 WAIT）
Control	保存控制传输管理有关信息的结构体
Device	保存所连接设备有关信息的结构体
pClass[USBH_MAX_NUM_SUPPORTED_CLASS]	指针数组，指针分别指向注册的类句柄结构体
pActiveClass	指向当前活动的类句柄结构体

USB OTG主机内核模块

- 主机库中所采用的主要结构体是USBH_handleTypedef类型的主机句柄

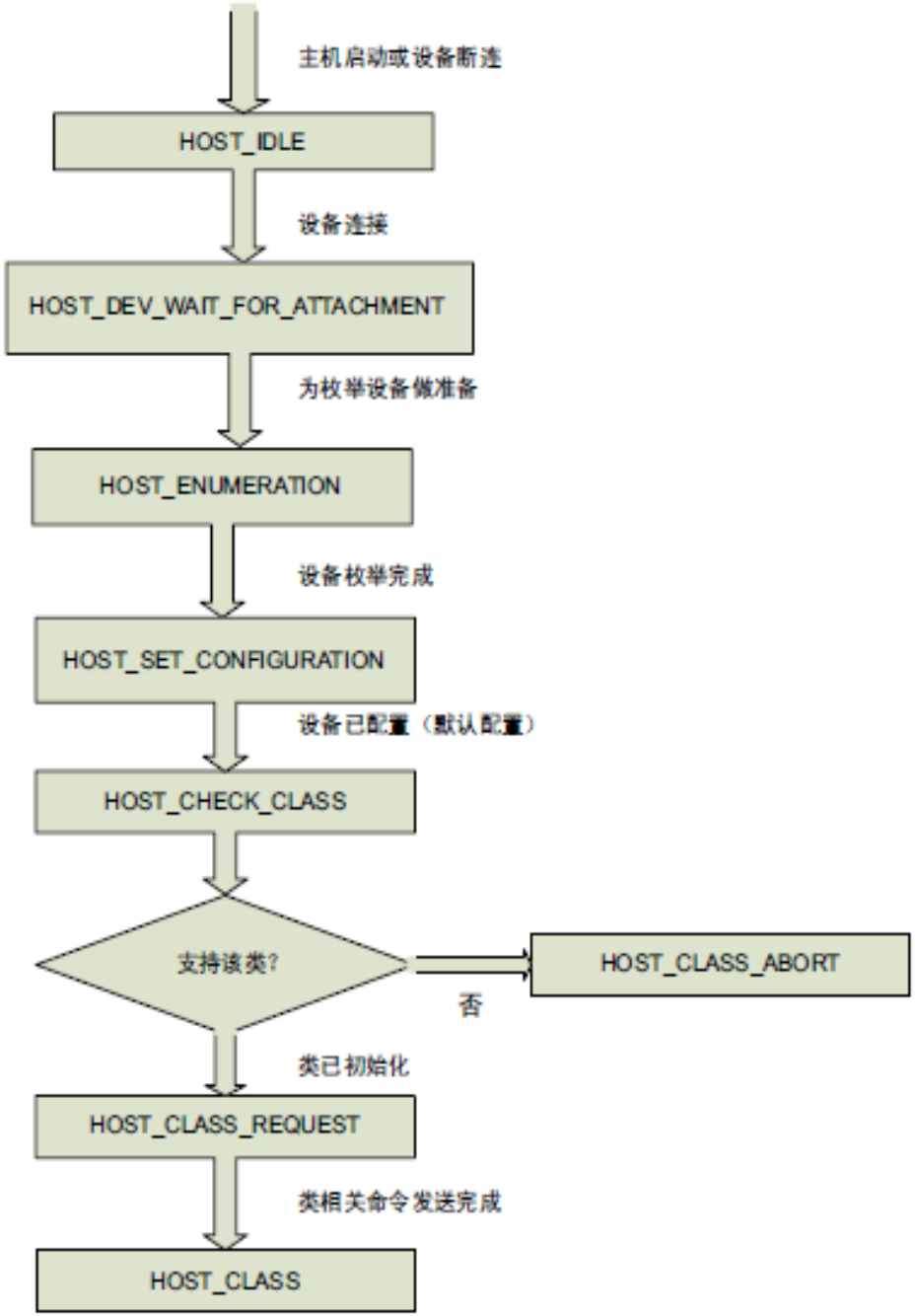
结构体成员	说明
ClassNumber	给出已注册类的总数
Pipes[15]	保存主机各个管道的状态（已分配或已释放）。还表示与该管道通信的设 备的端点号（如果有的话）
Timer	用于时间管理的计数变量 每一帧开始时会自动增加
pData	以底层主机控制器数据结构体（在 *usbh_conf.c* 配置文件中）进行初始化 的指针，可将库与底层驱动相连接
(* pUser)(struct _USBH_HandleTypeDe f *pHandle, uint8_t id);	主机用户事件回调函数

USB OTG主机内核模块

- 主机内核结构体，主机内核设备结构体保存了关于所连接设备的信息。该结构体的类型为*USBH_DeviceTypeDef*

函数	说明
CfgDesc_Raw	配置描述符原始数据
Data	数据缓存
Address	设备所分配的地址
Speed	设备速度
is_connected	设备的连接状态
current_interface	当前所选择的接口
DevDesc	含有设备描述符数据的结构体
CfgDesc	含有配置描述符数据的结构体（仅指配置描述符的前 9 个字节）

内核状态机概述


主机启动或设备断连
HOST_IDLE
设备连接
HOST_DEV_WAIT_FOR_ATTACHMENT
为枚举设备做准备
HOST_ENUMERATION
设备枚举完成
HOST_SET_CONFIGURATION
设备已配置（默认配置）
HOST_CHECK_CLASS
支持该类?
HOST_CLASS_ABORT
否
类已初始化
HOST_CLASS_REQUEST
类相关命令发送完成
HOST_CLASS

USB OTG底层驱动文件.编程模型.主机

Stm32fxxx_hal_hcd.c/.h

- 主机模块初始化
  - **HAL_HCD_Init**(**HCD_HandleTypeDef *hhcd**)
- 主机通道初始化
  - **HAL_HCD_HC_Init**(**HCD_HandleTypeDef *hhcd**, uint8_t ch_num,,,,)
- 启动主机传输
  - **HAL_HCD_HC_SubmitRequest** (**HCD_HandleTypeDef *hhcd** , uint8_t ch_num,,,,,)
  - 之后传输流就由HCD中断处理，用户可通过以下API监测传输状态
  - **HAL_HCD_HC_GetURBState** (**HCD_HandleTypeDef *hhcd** , uint8_t ch_num)
  - **HAL_HCD_GetState** (**HCD_HandleTypeDef *hhcd**)
  - **HAL_HCD_HC_GetXferCount** (**HCD_HandleTypeDef *hhcd**, uint8_t ch_num)
- USB主机中断
  - **HAL_HCD_IRQHandler** (**HCD_HandleTypeDef *hhcd**)

USB OTG主机内核模块

- 内核的底层驱动接口 ,USB主机库利用底层接口层作为与STM32Cube HAL的接口，与 STM32Cube HAL底层驱动交互。

函数	说明
USBH_LL_Init	底层初始化
USBH_LL_DeInit	底层解除初始化
USBH_LL_Start	底层开始
USBH_LL_Stop	底层停止
USBH_LL_GetSpeed	用于获取连接设备所检测到的速度
USBH_LL_ResetPort	发送 USB 复位信号
USBH_LL_GetLastXferSize	获取最近一次完成的传输大小

USB OTG主机内核模块

- 内核的底层驱动接口 ,USB主机库利用底层接口层作为与STM32Cube HAL的接口，与 STM32Cube HAL底层驱动交互。

函数	说明
USBH_LL_DriverVBUS	启用或禁用 VBUS
USBH_LL_OpenPipe	打开管道
USBH_LL_ClosePipe	关闭管道
USBH_LL_SubmitURB	提交主机传输请求
USBH_LL_GetURBState	获取管道状态
USBH_LL_SetToggle	设定传输的初始数据同步位（DATA0 或 DATA1）
USBH_LL_GetToggle	获取数据同步信息

USB OTG主机内核模块

- 底层接口在某些USB事件后所调用的主机库回调函数

回调函数	说明
USBH_LL_SetTimer	在 USB 主机启动过程中应由 USB 主机 HAL 驱动进行调用，以 初始化主机定时器
USBH_LL_IncTimer	每次 SOF 事件均应被调用，使主机定时器变量递增
USBH_LL_SOF	处理需要 SOF 同步的 USB 类过程时，应在 HAL SOF 事件回调 中被调用
USBH_LL_Connect	设备连接时，应在 USB 主机 HAL 事件回调中被调用
USBH_LL_Disconnect	设备断连时，应在 USB 主机 HAL 事件回调中被调用
USBH_LL_NotifyURB Change	采用 RTOS 模式时，当 USB 状态发生变化，该回调函数应在 USB 主机 HAL 事件回调中被调用

USB OTG主机内核模块

- 主机类模块

结构体成员	说明
Name;	类名称
ClassCode	USB 类代码
Init	类接口 Init：为类处理操作初始化所需的管道。在主机内核处理程序的 HOST_CHECK_CLASS 阶段被调用
DeInit	类接口 DeInit：解除接口的初始化。设备断连或当主机停止时被调用
Requests	类控制请求：处理类相关请求的状态机，在主机内核处理程序的 HOST_CLASS_REQUEST 阶段被调用
BgndProcess	类操作的后台处理。在主机内核程序的 HOST_CLASS 阶段被调用。
SOFProcess	SOF 的类处理：需要定期在 SOF 中断处理程序中执行的类相关操作。可用 于安排周期性传输 （ 中断、同步 ）
pData	在类的初始化过程中，用保存有类过程变量的类句柄结构体进行初始化

USB OTG主机内核模块

MSC类句柄通过USBH_ClassTypeDef类型的结构体来实现：

```
USBH_ClassTypeDef USBH_msc =
{
"MSC",
 USB_MSC_CLASS,
USBH_MSC_InterfaceInit,
USBH_MSC_InterfaceDeInit,
USBH_MSC_ClassRequest,
 USBH_MSC_BgndProcess,
NULL,
 USBH_MSC_Handle
 };
```

USB OTG主机内核模块

HID类句柄通过USBH_ClassTypeDef类型的结构体来实现，其定义如下：

```
USBH_ClassTypeDef HID_Class ={
 "HID",
USB_HID_CLASS,
USBH_HID_InterfaceInit,
 USBH_HID_InterfaceDeInit,
USBH_HID_ClassRequest,
USBH_HID_BgndProcess,
USBH_HID_SOF_Process, /* 用于处理中断管道的 SOF 过程 */
USBH_HID_Handle /* 处理程序结构体，用于保存 HID 的后台过程变量
*/

};
```

USB OTG底层驱动文件.编程模型.主机

- **USB** 主机库配置选项

typedef struct _HCD {}	
USBH_MAX_NUM_ENDPOINTS	所支持端点的最大数量
USBH_MAX_NUM_INTERFACES	所支持接口的最大数量
USBH_MAX_NUM_CONFIGURATION	所支持配置的最大数量
USBH_MAX_NUM_SUPPORTED_CLASS	所支持类的最大数量
USBH_KEEP_CFG_DESCRIPTOR	当定义为 1 时，所有配置描述符均保存在内存中
USBH_MAX_SIZE_CONFIGURATION	定义配置描述符的最大大小
USBH_MAX_DATA_BUFFER	定义用于数据传输的最大数据缓冲
USBH_DEBUG_LEVEL	定义日志记录级别： – 0：无日志 – 1：记录用户信息 – 2：记录用户消息与错误消息 – 3：记录用户消息、错误消息与调试消息
USBH_USE_OS	当定义为 1 时，配置主机工作于 OS 模式

HID类

- 支持HID boot鼠标和键盘设备
- 通过中断IN传输来获得HID report

函数	描述
USBH_HID_InterfaceInit	解析接口和端点描述符，配置主机通道以建立获取HID report所需的中断传输IN管道
USBH_HID_InterfaceDeInit	释放之前分配的IN管道
USBH_HID_ClassRequest	HID类相关命令：获得HID report描述符；设置IDLE时间；设置Protocol…
USBH_HID_Handle	HID类核心功能状态机：定时发起中断IN传输
USBH_Get_HID_Descriptor	HID类命令：获得HID类描述符
USBH_Get_HID_ReportDescriptor	HID类命令：获得HID report描述符
USBH_ParseHIDDesc	解析HID report描述符
USBH_Set_Idle	HID类命令：设置IDLE时间
USBH_Set_Report	HID类命令：发送Report OUT数据（demo中未用）
USBH_Set_Protocol	HID类命令：设置HID协议

USB主机库初始化调用

用户应用层

《main.c》

USBD_HandleTypeDef **USBD_Device**;

《usbd_desc.c》

USBD_DescriptorsTypeDef **VCP_Desc**
= { ..... };

《usbd_cdc_if.c》

USBD_CDC_ItfTypeDef **USBD_CDC_fops**
= { ..... };

《main.c》

```
/* 初始化USB设备库 */
USBD_Init                    ( &USBD_Device, &VCP_Desc, 0        );

/* 指定该设备所属USB类别 */
USBD_RegisterClass           ( &USBD_Device, USBD_CDC_CLASS);

/* 指定该应用具体功能 */
USBD_CDC_RegisterInterface   ( &USBD_Device, &USBD_CDC_fops );

/* 启动USB设备*/
USBD_Start                   ( &USBD_Device                      );
```

《usbd_cdc.h》

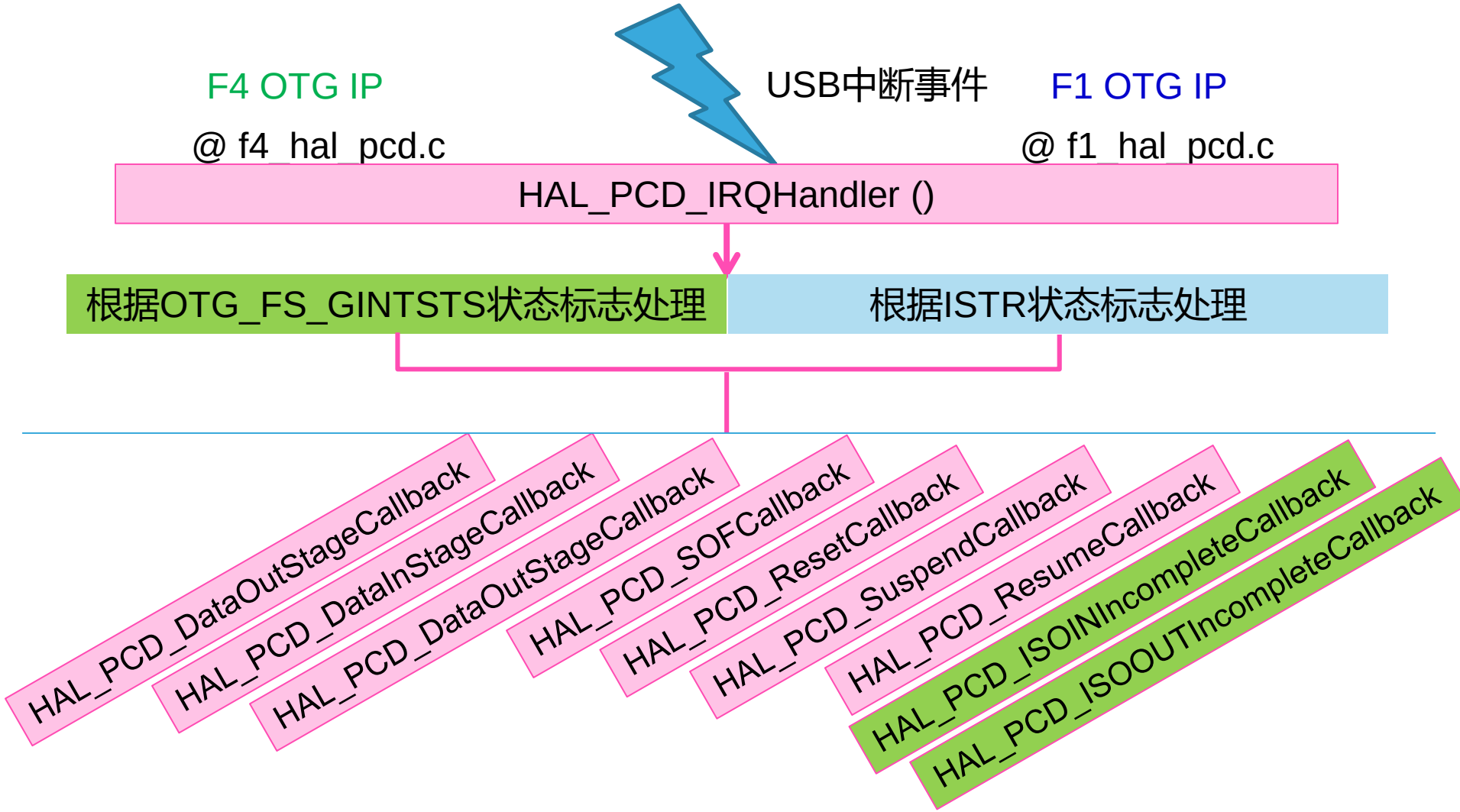
#define **USBD_CDC_CLASS** &USBD_CDC

《usbd_cdc.c》

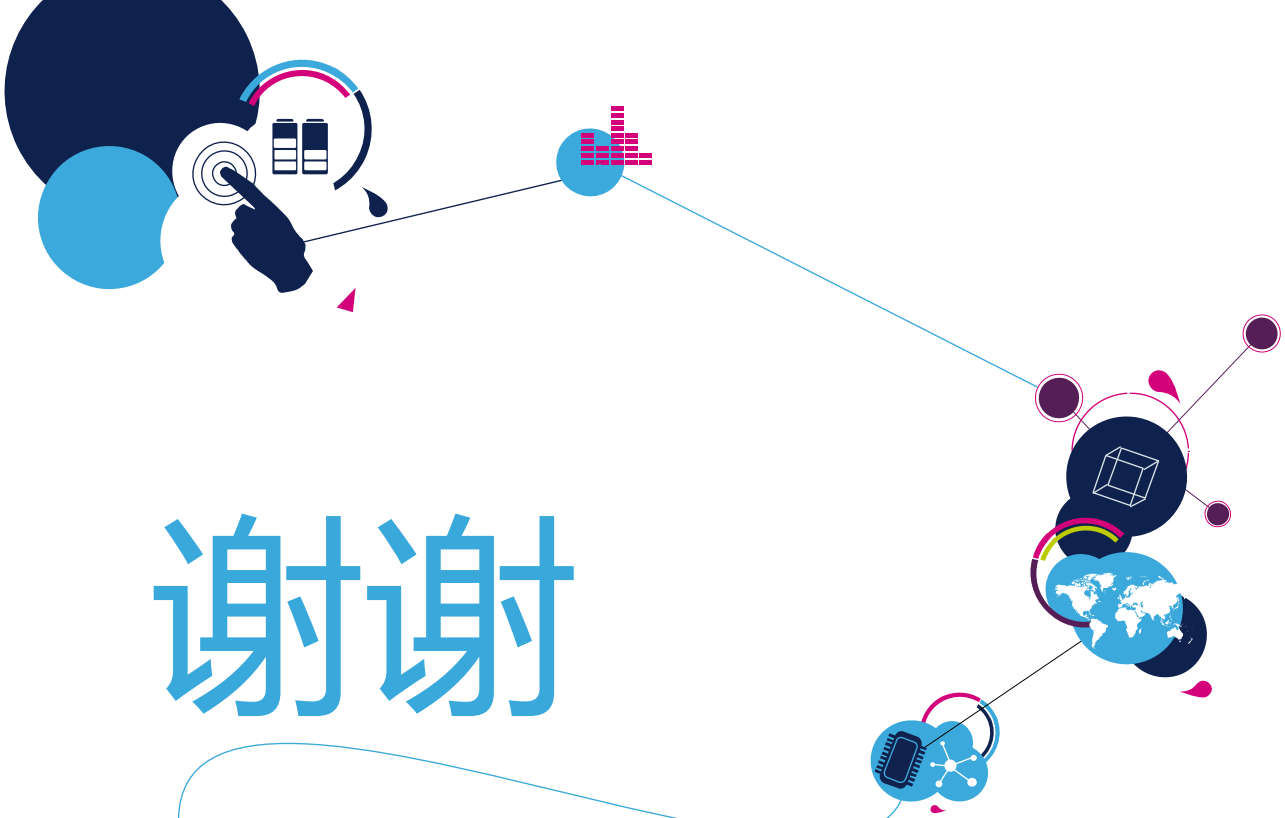
USBD_ClassTypeDef USBD_CDC
= { ..... }

life.augmented

USB主机库的中断响应



在stm32fxx_hal_pcd.c中是弱定义，在usbd_conf.c中重定义，但也只是直接调用usbd_core.c中的对应USBD_LL_XXX ()



谢谢

life.augmented