

# UM324xF USB 软件移植指南

版本：V1.0



广芯微电子（广州）股份有限公司

<http://www.unicmicro.com>

## 版本修订

版本	日期	描述
V1.0	2022.11.28	初始版

## 目录

1	USB demo 功能介绍 .....	1
2	USB 相关文件介绍 .....	1
3	重要的宏定义.....	2
4	重要的变量/数据结构.....	2
5	协议移植说明.....	3

# 1 USB demo 功能介绍

默认协议：走的是 WINUSB 协议，如果走其他应用协议类，需要客户自己修改相关描述符，并实现相应的应用层协议。

协议栈：只实现了 usb 标准请求的处理响应，对于类请求和厂商请求需要客户根据自己的应用协议在类请求和厂商请求函数接口中进行实现。

## 2 USB 相关文件介绍

- 《usb.c》：usb 驱动函数（初始化，数据收/发，中断响应等），函数接口详见 API 参考手册。
- 《usb.h》：usb.c 驱动文件对应的头文件。
- 《usb\_app.c》：usb 驱动层与 usb 中间件之间的接口，包括 usb 中断回调函数，usb 各个中断或事件的检测，以及事件对应的处理函数的调用接口。
- 《usb\_app.h》：usb\_app.c 文件对应的头文件。
- 《usb\_desc.c》：usb 协议相关的各种描述符的变量定义。
- 《usb\_desc.h》：usb\_desc.c 文件对应的头文件，包含描述符的数据结构。
- 《usb\_protocol\_extend.c》：供客户实现的控制请求接口，包括类请求接口，厂商请求接口，以及特殊描述符请求接口，需要客户根据具体的协议进行添加，文件中给出了 HID，UMS 的示例框架，请客户根据自己协议的需求进行添加实现。
- 《usb\_protocol\_extend.h》：usb\_protocol\_extend.c 文件对应的头文件。
- 《usb\_protocol\_xxx.lib》：usb 协议栈库，主要实现了 usb 标准请求和 winusb 协议的厂商请求，类请求、其他厂商请求以及特殊的描述符请求。已开放给客户进行添加实现（见 usb\_protocol\_extend.c 中 3 个实现接口）。
- 《usb\_protocol.h》：usb\_protocol\_xxx.lib 文件对应的头文件。
- 《winusb.c》：基于 WINUSB 协议下的应用层基本流程的处理示例函数（数据接收、协议解析、应用处理，数据响应）。
- 《winusb.h》：winusb.c 文件对应的头文件。

## 3 重要的宏定义

- USB\_COMM\_SIMPLE (winusb.h)

使能: demo 实现的功能是, 不解析数据帧格式, usb 收到什么数据, 就直接回发什么数据。

不使能: demo 实现的功能是, 解析数据帧格式, 并按照结构体 FRAME\_FORMAT\_T 定义的帧格式进行通信。

- USB\_DEBUG (usb.h)

使能: usb 枚举的过程中会出现过程打印信息

不使能: usb 枚举的过程中无打印信息

- USB\_SINGLE\_DEVICE (usb\_protocol.h)

使能: 单路设备

- USB\_COMPOSITE\_DEVICE (usb\_protocol.h)

使能: 复合设备

- PROTOCOL\_WINUSB (usb\_protocol.h) (其他几个宏类似)

使能: WINUSB 协议

- PROTOCOL\_TYPE (usb\_protocol.h): 当前选择的协议类型 (针对单路 usb 设备)

## 4 重要的变量/数据结构

- rec\_buff (app.h): byte 型接收 buffer 指针

- send\_buff (app.h): byte 型发送 buffer 指针

- BUFF\_SIZE (app.h): 收/发 buffer 总 size (接收、发送各占一半)

- DEVICE\_REQUEST (usb.h): usb 枚举过程中的设备请求结构体

- APP\_PARA\_T (app.h): 应用层数据结构, 包括通信模式, 全局数据 buffer、数据帧格式, usb 参数等

- FRAME\_FORMAT\_T (winusb.h): winusb 协议 demo 下的通信数据帧结构

- 各种描述符结构体和数组: 详见 usb\_desc.c 和 usb\_desc.h

## 5 协议移植说明

1. 根据自己的需求修改 `usb_desc.c` 中各种描述符。
2. 根据自己的需求协议，修改接口描述符中的 3 个 class 定义。
3. 对于应用需要的接口 class 协议，实现 `usb_class_request()` 函数。  
(`usb_vendor_request()` 函数和 `usb_get_description_extend()` 函数，大部分场景下用不上)。
4. 根据自己的需求协议，修改 `usb_app.c` 中的 `xxx_task_proc()` 函数，并在应用中实现 `xxx_task_proc()` 函数，可以参考 `winusb.c` 中的 `winusb_task_proc()` 函数实现流程。
5. 对于单路设备，使能 `USB_SINGLE_DEVICE` (`usb_protocol.h`)。
6. 对于复合设备，使能 `USB_COMPOSITE_DEVICE` (`usb_protocol.h`)。

注：如果想完全跳过协议栈，则需要客户自己完整实现 `usb_control_transfer()` 函数，即完整实现控制传输请求。