

CH37xLib 库接口函数说明

版本: 1B

<http://wch.cn>

一、简介

CH37xLib是macOS下USB设备通讯接口库, 提供USB全速/高速/超高速传输所需的API, 使用前需要安装CH37xUSBDriver (驱动安装, 请参考驱动安装文档), 该库支持macOS11.0及以上的系统版本, 适用于CH372/CH374/CH375/CH376/CH378等USB接口芯片。

二、使用说明

将 CH37xLib.h 和 CH37xLib.m 添加到应用中, 创建 CH37xLib 对象后, 调用对象提供的 API 接口完成与 USB 设备的通信。可参考如下步骤调用对应接口函数, 插入设备后, 主动扫描设备 (或使用热插拔检测) → 打开设备 → 枚举设备端点信息 → 通过端点接口函数进行通讯 → 关闭设备。

三、函数接口

CH37xDeviceConnect: 检测设备连接的代理方法

函数原型: - (void)CH37xDeviceConnect: (NSString *) locationID;

locationID: 设备的 locationID 的值

CH37xDeviceDisconnect: 检测设备断开的代理方法

函数原型: - (void)CH37xDeviceDisconnect: (NSString *) locationID;

locationID: 设备的 locationID 的值

CH37xScanDevices: 枚举连接的设备

函数原型: - (NSInteger)CH37xScanDevices;

返回 枚举到 CH37x 设备, 若无设备返回 0

CH37xGetAllDevices: 获取所有设备

函数原型: - (NSArray *)CH37xGetAllDevices;

返回 CH37x 设备的数组

CH37xGetLocationID: 获取设备的 locationID

函数原型: - (NSString *)CH37xGetLocationID: (NSString *)device;

device: 代表设备的字符串

返回 CH37x 设备的 LocationID

CH37xOpenDevice: 打开选择的设备

函数原型: - (IOReturn)CH37xOpenDevice: (NSInteger) index;

index: 设备序号

成功返回 IORetrunSuccess

CH37XCloseDevice: 关闭选择的设备

函数原型: - (IOReturn) CH37xCloseDevice: (NSInteger) index;

index: 设备序号

成功返回 IOReturnSuccess

CH37xSyncReadData: 同步读取

函数原型: - (IOReturn) CH37xSyncReadData: (NSInteger) index

 bEndpointAddress: (uint8_t) bEndpointAddress buffer: (void *) buffer
 size: (size_t) size;

index: 设备序号

bEndPointAddress: 端点地址

buffer: 存放读取数据的缓冲区

size: 读取的数据长度

成功返回 IOReturnSuccess

CH37xInterruptRead: 中断读取

函数原型: - (IOReturn) CH37xInterruptRead: (NSInteger) index

 bEndpointAddress: (uint8_t) bEndpointAddress buffer: (void *) buffer
 size: (size_t) size;

index: 设备序号

bEndPointAddress: 端点地址

buffer: 用于接收数据的buffer

size: 读取的数据长度

成功返回 IOReturnSuccess

CH37xSetBufUpload: 设置上传缓冲区大小，驱动内部默认100M

函数原型: - (IOReturn) CH37xSetBufUpload: (NSInteger) index

 bufferSize: (size_t) bufferSize;

index: 设备序号

bufferSize: 缓冲区大小

成功返回 IOReturnSuccess

CH37xStartBufUpload: 开始缓冲区上传，按照指定的包大小存放在缓冲区

函数原型: - (IOReturn) CH37xStartBufUpload: (NSInteger) index

 bEndpointAddress: (uint8_t) bEndpointAddress
 packageSize: (size_t) packageSize;

index: 设备序号

bEndPointAddress: 端点地址

packageSize: 每包的数据大小, 从缓冲区读取时需要按照这个大小进行读, 建议数据包大于等于1M

成功返回 IOReturnSuccess

CH37xReadEndP: 上传缓冲区读取数据包

函数原型: - (IOReturn) CH37xReadEndP: (NSInteger) index
 bEndpointAddress: (uint8_t) bEndpointAddress buffer: (uint8_t *) buffer
 size: (size_t) size;
index: 设备序号
bEndPointAddress: 端点地址
buffer: 接收数据包的缓冲区
size: 数据包大小, 必须与上传时数据包大小一致, 调用者可再次分包
成功返回 IOReturnSuccess

CH37xStopBufUpload: 停止缓冲区上传

函数原型: - (IOReturn) CH37xStopBufUpload: (NSInteger) index
 bEndpointAddress: (uint8_t) bEndpointAddress;
index: 设备序号
bEndPointAddress: 端点地址
成功返回 IOReturnSuccess

CH37xQueryBuf: 查询指定端点缓冲区数据包的个数

函数原型: - (NSInteger) CH37xQueryBuf: (NSInteger) index
 bEndpointAddress: (uint8_t) bEndpointAddress;
index: 设备序号
bEndPointAddress: 端点地址
返回 查询端点缓冲区包的个数

CH37xSyncWriteData: 同步写入

函数原型: - (IOReturn) CH37xSyncWriteData: (NSInteger) index
 bEndpointAddress: (uint8_t) bEndpointAddress buffer: (void *) buffer
 size: (size_t) size;
index: 设备序号
bEndPointAddress: 端点地址
buffer: 存放写入数据的缓冲区
size: 写入数据长度
成功返回 IOReturnSuccess

CH37xSetBufDownload: 设置下传缓冲区大小, 驱动内部默认100M

函数原型: - (IOReturn) CH37xSetBufDownload: (NSInteger) index bufferSize:
 (size_t) bufferSize;
index: 设备序号
bufferSize: 缓冲区大小
成功返回 IOReturnSuccess

CH37xStartBufDownload: 开始缓冲区下传

函数原型: - (IOReturn) CH37xStartBufDownload: (NSInteger) index
 bEndpointAddress: (uint8_t) bEndpointAddress buffer: (void *) buffer

size: (size_t) size needRead: (BOOL) isRead;
index: 设备序号
bEndPointAddress: 端点地址
buffer: 存放数据包的缓冲区
isRead: 是否在下传数据时会进行读的操作
成功返回 IOReturnSuccess

CH37xResetWrite: 重置选择的下传端点的状态

函数原型: - (IOReturn) CH37xResetWrite: (NSInteger) index
bEndpointAddress: (uint8_t) bEndpointAddress;
index: 设备序号
bEndPointAddress: 端点地址
成功返回 IOReturnSuccess

CH37xSetTimeout: 设置读写超时

函数原型: - (IOReturn) CH37xSetTimeout: (NSInteger) index readTimeout:
(uint64_t) readTimeout writeTimeout: (uint64_t) writeTimeout;
index: 设备序号
readTimeout: 读端点的超时时间, 单位ms
writeTimeout: 写端点的超时时间, 单位ms
成功返回 IOReturnSuccess

CH37xGetVersion: 获取驱动版本信息

函数原型: - (IOReturn) CH37xGetVersion: (NSInteger) index buffer: (void *) buffer
size: (size_t) size;
index: 设备序号
buffer: 存放版本信息的缓冲区, 大小为64
size: 缓冲区大小64
成功返回 IOReturnSuccess

CH37xGetDeviceID: 获取设备的PID和VID

函数原型: - (IOReturn) CH37xGetDeviceID: (NSInteger) index buffer: (uint64_t
) *buffer size: (uint32_t) size;
index: 设备序号
buffer: 存放设备ID的数组
size: 缓冲区大小2
成功返回 IOReturnSuccess, buffer[0] 为PID, buffer[1] 为VID

CH37xGetDeviceMsg: 获取设备的全部端点

函数原型: - (NSArray *) CH37xGetDeviceMsg: (NSInteger) index;
index: 设备序号
返回值: 存放IOUSBEndpointDescriptor结构体的数组, 通过
enumerateObjectsUsingBlock进行枚举

CH37xGetBulkInEndPoint: 获取设备的读(上传)端点

函数原型: - (NSArray *)CH37xGetBulkInEndPoint: (NSInteger) index;

index: 设备序号

返回值: 存放IOUSBEndpointDescriptor结构体的数组, 通过
 enumerateObjectsUsingBlock进行枚举

CH37xGetBulkOutEndPoint: 获取设备的写(下传)端点

函数原型: - (NSArray *)CH37xGetBulkOutEndPoint: (NSInteger) index;

index: 设备序号

返回值: 存放IOUSBEndpointDescriptor结构体的数组, 通过
 enumerateObjectsUsingBlock进行枚举

CH37xGetInterruptEndPoint: 获取设备的中断端点

函数原型: - (NSArray *)CH37xGetInterruptEndPoint: (NSInteger) index;

index: 设备序号

返回值: 存放IOUSBEndpointDescriptor结构体的数组, 通过
 enumerateObjectsUsingBlock进行枚举

CH37xGetDeviceDesc: 获取设备的描述

函数原型: - (IOReturn)CH37xGetDeviceDesc: (NSInteger) index buffer: (uint64_t
 *)buffer;

index: 设备序号

buffer: 存放描述信息的数组, 长度为12

成功返回 IOReturnSuccess, buffer取值如下

```
buffer [0] = bLength;  
buffer [1] = bDescriptorType;  
buffer [2] = bcdUSB;  
buffer [3] = bDeviceClass;  
buffer [4] = bDeviceSubClass;  
buffer [5] = bDeviceProtocol;  
buffer [6] = bMaxPacketSize0;  
buffer [7] = bcdDevice;  
buffer [8] = iManufacturer;  
buffer [9] = iProduct;  
buffer [10] = iSerialNumber;  
buffer [11] = bNumConfigurations;
```