

WCHBLEDLL 库接口函数说明

版本：1A

<http://wch.cn>

一、概述

WCHBLEDLL是一款用于Windows系统下进行低功耗蓝牙应用开发的WIN32标准动态库，无需UWP环境，支持C/C++/C#/Java/Python等各编程语言调用。

WCHBLEDLL提供BLE设备的枚举、扫描、连接和检测，蓝牙服务和特征的枚举、读写、订阅，控制器版本查询等API函数。

支持Windows8.1及以上系统，蓝牙控制器4.0及以上版本。

二、库函数介绍

2.1. 库初始化

```
void WINAPI WCHBLEInit();
```

该函数会初始化库运行前的准备，需要在所有库函数调用之前先调用，并且只能调用一次。

2.2. 查询系统蓝牙是否打开

```
BOOL WINAPI WCHBLEIsBluetoothOpened();
```

函数返回值为 FALSE 为未打开；

返回值为 TRUE 为已打开。

2.3. 查询蓝牙控制器版本

```
UCHAR WINAPI WCHBLEGetBluetoothVer();
```

函数返回值对应下表：

返回值	版本号
0	V1.0
1	V1.1
2	V1.2
3	V2.0
4	V2.1
5	V3.0
6	V4.0
7	V4.1
8	V4.2
9	V5.0
10	V5.1
11	V5.2

2.4. 查询控制器是否支持低功耗蓝牙

```
BOOL WINAPI WCHBLEIsLowEnergySupported();
```

函数返回值为 0 不支持；
返回值为 1 为支持。

2.5. 查询控制器是否可以作为从机

```
BOOL WINAPI WCHBLEIsPeripheralRoleSupported();
```

函数返回值为 0 不可以；
返回值为 1 为可以。

2.6. 查询控制器是否可以作为主机

```
BOOL WINAPI WCHBLEIsCentralRoleSupported();
```

函数返回值为 0 不可以；
返回值为 1 为可以。

2.7. 查询控制器是否支持低功耗安全连接

```
BOOL WINAPI WCHBLEAreLowEnergySecureConnectionsSupported();
```

函数返回值为 0 不支持；
返回值为 1 为支持。

2.8. 查询控制器是否支持加载广播包

```
BOOL WINAPI WCHBLEIsAdvertisementOffloadSupported();
```

函数返回值为 0 不支持；
返回值为 1 为支持。

2.9. 设备枚举

```
void WINAPI WCHBLEEnumDevice(ULONG scanTimes, PCHAR DevIDFilter, BLENameDevID* pBLENameDevIDArray, PULONG pNum);
```

该函数会在规定时间扫描附近设备并通过结构体数组返回扫描结果。

参数 scanTimes 是指定扫描时间，以 ms 为单位；

参数 DevIDFilter 是过滤名称字符串，如果为空则返回扫描到的所有设备；

参数 pBLENameDevIDArray 为扫描结果结构体数组，结构体定义如下

```
struct BLENameDevID
{
    UCHAR Name[MAX_PATH];
    UCHAR DevID[MAX_PATH];
    int Rssi;
};
```

其中 Name[MAX_PATH] 是设备名；

DevID[MAX_PATH] 是设备路径；

Rssi 为扫描到设备时的 Rssi（信号强度）值；

参数 pNum 是指定扫描设备数量，并且在函数执行完会返回真是扫描到的设备数量。

2.10. CH914X 设备枚举

```
void WINAPI WCHBLEEnumCH914X(ULONG scanTimes, BLENameDevID* pBLENameDevIDArray, PULONG pNum);
```

该函数会在规定时间扫描附近 CH914X 设备并通过结构体数组返回扫描结果。

参数 scanTimes 是指定扫描时间，以 ms 为单位；

参数 pBLENameDevIDArray 为扫描结果结构体数组，结构体定义如下

```
struct BLENameDevID
{
    UCHAR Name[MAX_PATH];
    UCHAR DevID[MAX_PATH];
    int Rssi;
};
```

其中 Name[MAX_PATH] 是设备名；

DevID[MAX_PATH] 是设备路径；

Rssi 为扫描到设备时的 Rssi（信号强度）值；

参数 pNum 是指定扫描设备数量，并且在函数执行完会返回真是扫描到的设备数量。

2.11. 打开设备

```
WCHBLEHANDLE WINAPI WCHBLEOpenDevice(PCHAR deviceId,
pFunDevConnChangeCallBack pFunDevConnChange);
```

该函数会将指定路径的设备打开。

函数返回值为 typedef void* WCHBLEHANDLE 类型的句柄，此句柄在后续调用函数时将作为参数传递；

参数 deviceId 是设备路径，可通过扫描设备获取；

参数 pFunDevConnChange 为上传设备连接状态的回调函数。

2.12. 关闭设备

```
void WINAPI WCHBLECloseDevice(WCHBLEHANDLE pDev);
```

该函数将关闭已打开的设备。

参数 pDev 为设备句柄。

2.13. 获取设备所有服务

```
UCHAR WINAPI WCHBLEGetAllServicesUUID(WCHBLEHANDLE pDev, PUSHORT pUUIDArray,
PUSHORT pUUIDArrayLen);
```

该函数将获取设备的所有服务以及数量。

函数返回值为 0 成功；

返回值为 1 设备句柄无效；

返回值为其他表示获取失败。

参数 pDev 为设备句柄；

参数 pUUIDArray 为获取服务存放数组；

参数 pUUIDArrayLen 为获取服务个数。

2.14. 获取对应服务的特征

```
UCHAR WINAPI WCHBLEGetCharacteristicByUUID(WCHBLEHANDLE pDev, USHORT
ServiceUUID, PUSHORT pUUIDArray, PUSHORT pUUIDArrayLen);
```

该函数将获取服务的所有特征值以及特征数量。

函数返回值为 0 成功；

返回值为 1 设备句柄无效；

返回值为其他表示获取失败。

参数 pDev 为设备句柄；

参数 ServiceUUID 为服务标识;

参数 pUUIDArray 为获取特征值存放数组;

参数 pUUIDArrayLen 为获取特征值个数。

2. 15. 获取特征支持操作

UCHAR WINAPI WCHBLEGetCharacteristicAction(WCHBLEHANDLE pDev, USHORT ServiceUUID, USHORT CharacteristicUUID, PULONG pAction);

该函数将返回特征支持的操作。

函数返回值为 0 成功;

返回值为 1 设备句柄无效;

返回值为 2 特征句柄无效;

返回值其他表示获取失败。

参数 pDev 为设备句柄;

参数 ServiceUUID 为服务标识;

参数 CharacteristicUUID 为特征标识;

参数 pAction 为获取结果,

bit0 为 1 表示支持读操作, 为 0 则表示不支持读操作;

bit1 为 1 表示支持写操作, 为 0 则表示不支持写操作;

bit2 为 1 表示支持订阅操作, 为 0 则表示不支持订阅操作。

2. 16. 写入特征值

UCHAR WINAPI WCHBLEWriteCharacteristic(WCHBLEHANDLE pDev, USHORT ServiceUUID, USHORT CharacteristicUUID, BOOL bWriteWithResponse, PCHAR buffer, UINT length);

该函数用于写入特征值。

函数返回值为 0 成功;

返回值为 1 设备句柄无效;

返回值为 2 特征句柄无效;

返回值其他表示写入失败。

参数 pDev 为设备句柄;

参数 ServiceUUID 为服务标识;

参数 CharacteristicUUID 为特征标识;

参数 bWriteWithResponse 为传输模式, 为 0 表示无应答传输, 为 1 表示有应答传输;

参数 buffer 为待写入的字符串;

参数 length 为待写入的字符串长度。

2. 17. 读取特征值

UCHAR WINAPI WCHBLEReadCharacteristic(WCHBLEHANDLE pDev, USHORT ServiceUUID, USHORT CharacteristicUUID, PCHAR buffer, PUINT length);

该函数用于读取特征值。

函数返回值为 0 成功;

返回值为 1 设备句柄无效;

返回值为 2 特征句柄无效;

返回值其他表示读取失败。

参数 pDev 为设备句柄;

参数 ServiceUUID 为服务标识;

参数 CharacteristicUUID 为特征标识;

参数 buffer 为存放读取结果字符串;

参数 Length 为期望读取的字符串长度 (最大不超过 512), 函数执行成功后为实际读取长度。

2. 18. 开启/关闭读订阅

UCHAR WINAPI WCHBLERegisterReadNotify(WCHBLEHANDLE pDev, USHORT ServiceUUID, USHORT CharacteristicUUID, pFunReadCallback pFun, void* paramInf);

该函数用于开启读订阅, 当特征值发生改变时会通过回调函数返回特征值。

函数返回值为 0 成功;

返回值为 1 设备句柄无效;

返回值为 2 特征句柄无效;

返回值其他表示开启订阅失败。

参数 pDev 为设备句柄;

参数 ServiceUUID 为服务标识;

参数 CharacteristicUUID 为特征标识;

参数 pFun 为读订阅的回调函数;

参数 paramInf 为可包含设备句柄、服务标识和特征标识的任意设备信息结构体。

当 pFun 参数传 NULL 值时代表关闭读订阅。

2. 19. 开启/关闭 RSSI 订阅

UCHAR WINAPI WCHBLERegisterRSSINotify(pFunRSSICallback pFun);

该函数用于开启 RSSI 订阅, 当系统检测到 RSSI 时会通过回调函数上报检测结果。

函数返回值为 0 成功;

返回值其他表示开启 RSSI 订阅失败。

当 pFun 参数传 NULL 值时代表关闭 RSSI 订阅。

2. 20. 获取设备 MTU

UCHAR WINAPI WCHBLEGetMtu(WCHBLEHANDLE pDev, PUSHORT pMTU);

该函数用于获取当前连接设备的 MTU 获取。

函数返回值为 0 成功;

返回值为 1 设备句柄无效;

返回值其他表示获取 MTU 失败。

参数 pDev 为设备句柄;

参数 pMTU 为存放获取结果变量。

三、回调函数介绍

回调函数分别有三种: 读取特征值回调、连接状态回调、RSSI 回调。

3. 1. 读取特征值回调

typedef void (CALLBACK* pFunReadCallback)(void* ParamInf, PCHAR ReadBuf, ULONG ReadBufLen);

该函数用于在设备接收到数据时, 在回调函数中给出接收到的数据。

参数 ParamInf 为可包含设备句柄和特征句柄的任意设备信息结构体, 在开启订阅是将此结构体传给库, 在收到数据时可通过此结构体里的信息判断是哪一个设备的哪一个特

征收到的数据；

参数 ReadBuf 为接收到的数据；

参数 ReadBufLen 为接收到的数据长度。

3.2. 连接状态回调

```
typedef void (CALLBACK* pFunDevConnChangeCallBack)(void* hDev, UCHAR  
ConnectStatus);
```

该函数用于返回连接状态，在连接状态发生改变时会上报当前状态。

参数 hDev 为设备句柄；

参数 ConnectStatus 为当前的连接状态，0 为断开，1 为连接。

3.3. RSSI 回调

```
typedef void (CALLBACK* pFunRSSICallBack)(PCHAR pMAC, int rssi);
```

该函数用于返回系统扫描到的设备 RSSI 值。

参数 pMAC 为扫描到的设备的 MAC 地址的字符串；

参数 rssi 为扫描时该设备的 RSSI 值。

四、接口调用顺序介绍

1. 初始化库 WCHBLEInit
2. 枚举设备 WCHBLEEnumDevice
3. 打开设备 WCHBLEOpenDevice
4. 获取设备服务标识 WCHBLEGetAllServicesUUID
5. 获取设备 MTU 值 WCHBLEGetMtu
6. 获取设备特征标识 WCHBLEGetCharacteristicByUUID
7. 获取各特征句柄对应支持的操作 WCHBLEGetCharacteristicAction
8. 读取特征值 WCHBLEReadCharacteristic
9. 写入特征值 WCHBLEWriteCharacteristic
10. 开启 ValueChange 事件订阅 WCHBLERegisterReadNotify
11. 关闭 ValueChange 事件订阅 WCHBLERegisterReadNotify
12. 开启 RSSI 事件订阅 WCHBLERegisterRSSINotify
13. 关闭 RSSI 事件订阅 WCHBLERegisterRSSINotify
14. 断开 BLE 设备连接 WCHBLECloseDevice