CH9140lib 库接口函数说明

版本：1A

http://wch.cn

一、概述

CH9140lib是Linux系统下的BLE串口接口库，用于CH914X低功耗蓝牙转串口的芯片模组，提供BLE串口设备的自动搜索、连接、打开关闭、串口参数设置、MODEM操作、串口读写、蓝牙控制器版本号查询等操作的API函数。

二、库函数介绍

2.1. 查询蓝牙控制器版本

int CH9140BleGetBluetoothVer();

函数正确返回值对应下表：

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

返回值为-1 表示设备查询超时，返回值为-2 表示系统蓝牙未打开。

2.2. 扫描设备

int CH9140Ble_BLE_Scan(int ScanTime, FunDiscoverDeviceInfo Ble_AdvertisingDevice_Info);

该函数会在规定时间内扫描附近 CH914X 设备，通过扫描回调函数返回扫描结果（设备地址、设备名称、信号强度 RSSI 和芯片版本号）。

参数 ScanTime 是指定的扫描时间，以秒为单位；

参数 Ble_AdvertisingDevice_Info 为扫描结果的回调函数（设备地址、设备名称、信号强度 RSSI 和芯片版本号）；

函数返回值为 1 表示系统蓝牙未打开；

返回值为 0 表示扫描成功。

2.3. 连接设备

CH9140HANDLE *CH9140Ble_Connect(const char *mac, FunConnectionStateCallback connectionstate);

该函数通过 mac 地址连接设备。

函数返回值为 CH9140HANDLE，此句柄在后续函数调用时将作为参数传递；

参数 mac 是设备地址，可通过扫描设备函数获取；

参数 connectionstate 是连接回调函数，上传连接状态。

2.4. 断开连接

Void CH9140Ble_Disconnect(CH9140HANDLE *connection);

该函数将断开已连接的设备；

参数 connection 为设备连接句柄。

2.5. 注册连接断开回调

void CH9140Ble_register_on_disconnect(CH9140HANDLE *connection, FunDisconnectionStateCallback disconnection_state);

该函数将注册设备连接断开事件。

参数 connection 为设备连接句柄；

参数 disconnection_state 为连接断开状态回调。

2.6. 获取设备服务

int CH9140Ble_discover_primary(CH9140HANDLE *connection, gatt_primary_service *services, int *services_count);

该函数将获取连接设备的所有服务。

参数 connection 为设备连接句柄；

参数 services 获取设备的服务；

参数 services_count 获取设备服务的个数；

函数返回值为 0 表示获取服务成功；

函数返回值为-1 表示获取服务失败。

2.7. 获取设备特征

int CH9140Ble_discover_characteristics(CH9140HANDLE *connection, gatt_characteristic *characteristics, int *characteristics_count);

该函数将获取连接设备的所有特征。

参数 connection 为设备连接句柄；

参数 characteristics 为获取设备特征的结构体数组，结构体定义如下：

```
typedef struct {
    uint16_t  handle;
    uint8_t   properties;
    uuid_t    uuid;
} gatt_characteristic;
```

其中 handle 为特征句柄，uuid 为特征标识；

Properties 为特征的特性值，具体含义如下：

bit0(0x01)为 1 支持广播（broadcast）操作，为 0 表示不支持广播模式；

bit1(0x02)为 1 表示支持读操作，为 0 则表示不支持读操作；

bit2(0x04)为 1 表示支持无应答写操作（write without response），为 0 则表示不支持无应答写操作；

bit3(0x08)为 1 表示支持有应答写操作（write with response），为 0 则表示不支持有

应答写操作；

bit4(0x10)为 1 表示支持通知操作（notification），为 0 则表示不支持通知操作。

参数 characteristics_count 获取特征的个数；

函数返回值为 0 表示获取特征成功；

函数返回值为-1 表示获取特征失败。

2.8. 读取特征值

int CH9140Ble_Read_Char_by_Handle(CH9140HANDLE *connection, uint16_t handle, char *buffer, size_t *length);

该函数用于读取特征值。

参数 connection 为设备连接句柄；

参数 handle 为特征句柄；

参数 Buffer 为存放读取结果的字符串；

参数 length 为期望读取的字符串长度；

函数返回值为 0 表示读取特征成功；

其他返回值表示读取特征失败。

2.9. 写入特征值

int CH9140Ble_Write_Characteristic(CH9140HANDLE *connection, const char *CharacteristicUUID, bool WriteWithResponse, const char *buffer, size_t buffer_length);

该函数用于写入特征值。

参数 connection 为设备连接句柄；

参数 CharacteristicUUID 为特征的 UUID（其特性值需可写）；

参数 WriteWithResponse 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 为代写入的字符串；

参数 length 为代写入字符串的长度；

函数返回值为 0 成功；

函数返回值为 1 表示写入失败；

2.10. 注册通知

void CH9140Ble_register_notification(CH9140HANDLE *connection, FunRegisterNotifyCallback notification_handler);

该函数为注册通知，通过通知回调函数接收数据。

参数 connection 为设备连接句柄；

参数 notification_handler 为通知回调函数。

2.11. 打开通知

int CH9140Ble_Open_Notification(CH9140HANDLE *connection, const char *CharacteristicUUID);

该函数用于打开通知。

参数 connection 为设备连接句柄；

参数 CharacteristicUUID 为特征标识；

函数返回值为 0 表示打开通知成功；

函数返回值为-1 表示打开通知失败。

2.12. 关闭通知

int CH9140Ble_Close_Notification(CH9140HANDLE *connection, const char *CharacteristicUUID);

该函数用于关闭通知。

参数 connection 为设备连接句柄；

参数 CharacteristicUUID 为特征标识；

函数返回值为 0 表示关闭通知成功；

函数返回值为-1 表示关闭通知失败。

2.13. 获取设备 MTU

int CH9140Ble_Get_MTU(CH9140HANDLE *connection, uint16_t *mtu);

该函数用于获取当前连接设备的 MTU 值。

参数 connection 为设备连接句柄；

参数 mtu 为存放获取结果的变量；

返回值为 0 表示获取 MTU 值成功，其它返回值表示失败。

2.14. 发送串口数据

int CH9140BleUart_Write_Buffer(CH9140HANDLE *connection, char *buffer, int buffer_length, uint16_t mtu)

该函数用于发送串口数据。

参数 connection 为设备连接句柄；

参数 buffer 为代写入的字符串；

参数 length 为代写入字符串的长度；

函数返回值大于 0 时，表示实际发送的数据数；

返回值为-1，表示 buffer 长度超过 4096。

2.15. 设置串口参数

int CH9140Ble_Set_SerialBaud(CH9140HANDLE *connection, int BaudRate, int DataBit, int StopBit, int Parity);

该函数将设置串口波特率、数据位、停止位和校验位参数。

参数 connection 为设备连接句柄；

参数 BaudRate，波特率。支持常规的波特率设置，例如 9600,115200 等；

参数 DataBit，数据位，5-8；

参数 StopBit，数据位，1-2；

参数 Parity，0：无校验，1：奇校验，2：偶校验，3：标志位，4：空白位；

函数返回值为 0 表示串口参数设置成功；

函数返回值为-1 表示串口参数设置失败。

2.16. 设置串口流控与 Modem 状态

int CH9140Ble_Set_Modem(CH9140HANDLE *connection, int flow, int DTR, int RTS);

该函数用于设置流控与 Modem 状态。

参数 connection 为设备连接句柄；

参数 flow 为硬件自动流控开关，1：开启，0：关闭；

参数 DTR，1 有效、低电平，0：无效、高电平；

参数 RTS，1 有效、低电平，0：无效、高电平；

函数返回值为 0 表示成功；
函数返回值为-1 表示失败。

2.17. 显示串口 Modem 状态

void CH9140Ble_Get_Modem_state(const uint8_t *data, FunGetModemStateCallback getmodemstate);

该函数用来显示串口的 Modem 状态。
参数 data，接收通知回调的数据,用来显示 Modem 状态；
参数 getmodemstate，获取 Modem 状态回调函数。

三、回调函数介绍

3.1. 设备扫描回调

typedef void (*FunDiscoverDeviceInfo)(const char *addr, const char *name, int8_t rssi, uint8_t *ChipVer);

该函数用于返回扫描到的 CH9140X 设备信息。
参数 addr，扫描到的设备地址；
参数 name，扫描到的设备名称；
参数 rssi，扫描到的设备信号强度；
参数 ChipVer 为芯片版本号。

3.2. 连接状态回调

typedef void(*FunConnectionStateCallback)(CH9140HANDLE *connection, int state);

该函数用于返回设备连接时的状态，在连接状态发生改变时会上报当前连接状态。
参数 connection 为设备连接句柄；
参数 state，1 表示连接成功，0 表示连接失败。

3.3. 断开连接状态回调

typedef void(*FunDisconnectionStateCallback)(bool state);

该函数用于上报设备连接断开事件。

3.4. 通知回调

typedef void (*FunRegisterNotifyCallback)(const uuid_t *uuid, const uint8_t *data, size_t data_length);

该函数用于接收串口发来的数据；
参数 uuid 为特征标识；
参数 data，接收的数据；
参数 data_length，接收数据的长度。

3.5. Modem 状态回调

typedef void(*FunGetModemStateCallback)(bool DCD, bool RI, bool DSR, bool CTS);

该函数用于返回串口 Modem 状态值。
参数 DCD 为串口的 DCD 输入状态；
参数 RI 为串口 RI 输入状态；
参数 DSR 为串口 DSR 输入状态；

参数 CTS 为串口 CTS 输入状态。

四、接口调用顺序介绍

4.1. 调用顺序一

1. 扫描设备 CH9140Ble_BLE_Scan
2. 连接设备 CH9140Ble_Connect
3. 注册断开连接回调 CH9140Ble_register_on_disconnect
4. 获取设备 MTU 值 CH9140Ble_Get_MTU
5. 获取设备服务 CH9140Ble_discover_primary
6. 获取设备特征 CH9140Ble_discover_characteristics
7. 打开通知 CH9140Ble_Open_Notification
8. 设置波特率 CH9140Ble_Set_SerialBaud
9. 设置流控 CH9140Ble_Set_Modem
10. 读取特征值 CH9140Ble_Read_Char_by_Handle
11. 写入特征值 CH9140Ble_Write_Characteristic
12. 关闭通知 CH9140Ble_Close_Notification
13. 断开 BLE 设备连接 CH9140Ble_Disconnect

4.2. 调用顺序二

1. 扫描设备 CH9140Ble_BLE_Scan
2. 连接设备 CH9140Ble_Connect
3. 注册断开连接回调 CH9140Ble_register_on_disconnect
4. 获取设备 MTU 值 CH9140Ble_Get_MTU
5. 打开通知 CH9140Ble_Open_Notification
6. 设置波特率 CH9140Ble_Set_SerialBaud
7. 设置流控 CH9140Ble_Set_Modem
8. 发送数据 CH9140BleUart_Write_Buffer
9. 断开 BLE 设备连接 CH9140Ble_Disconnect