



AC690x_微信硬件_介绍

AC690x_微信硬件_介绍 用户手册

Rev 1.0 —— 2017 年 5 月 26 日 星期五

This translated version is for reference only, and the English version shall prevail in case of any discrepancy between the translated and English versions.

版权所有 2016 杰理科技有限公司未经许可，禁止转载



目录

Chapter1 蓝牙微信硬件开发使用说明.....	4
1.1 编写目的.....	4
1.2 术语和缩写词.....	4
1.3 开发说明.....	5



修改日志

版本	日期	描述
V101	2017/5/25	基本功能完成
V102	2017/5/31	完善功能，编写文档

- 更新：
- 建立初始版本
 - 定义文档格



Chapter1 蓝牙微信硬件开发使用说明

1.1 编写目的

该文档主要描述 AC690x_wechat 开发包的使用方法及开发中注意的一些问题，为用户进行二次开发提供参考。

AC690x_wechat 是基于 AC690x_SDK 基础上进行开发，详细开发说明可参考《AC690x_SDK_介绍.pdf》，下面是针对蓝牙 wechat 开发使用进行说明。

1.2 术语和缩写词

该文档主要描述 AC69_SDK 开发包的使用方法及开发中注意的一些问题，为用户进行二次开发提供参考。

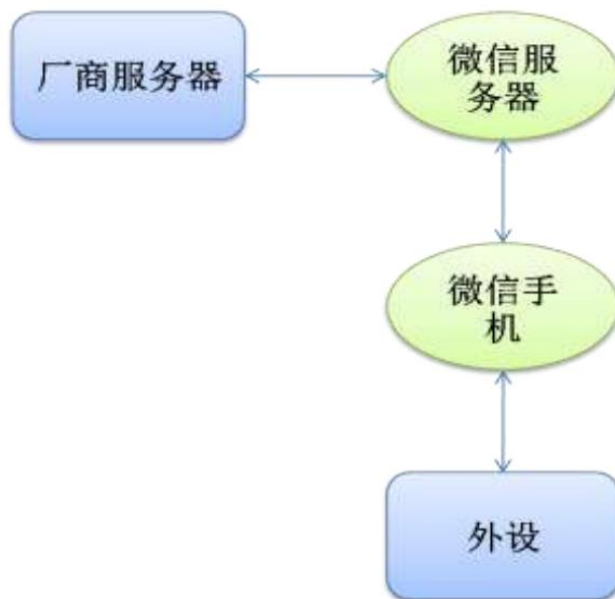
提示：列出本文件中用到专门术语的定义和英文首字母组词的原词组。

缩写和术语	解 释	
AC690x	杰理科技 AC690x 系列芯片	
Ble_wechat	微信硬件	
AuthReq	登录请求	
InitReq	初始化请求	



1.3 微信硬件开发说明

- 整体结构介绍:



注意事项:

- (1) 服务器由客户自己开发完成
- (2) 客户需先开通微信公众平台的硬件号功能。
- (3) 客户需先在公众平台上注册设备（具体 api 见服务器端的微信公众平台的硬件功能文档）

- 工程介绍:

- ❖ Ble_wechat 使能

/// ble 微信硬件功能使能,前提要使能 ble 才能开启

```
#define BT_BLE_WECHAT 1
```

- ❖ 广播包设置。

- 1.根据微信硬件蓝牙协议，广播包需要满足下面条件:



- a. 微信规定的 service uuid。
- b. 厂商自定义字段里，包含 MAC 地址。
- c. 包含指定的 Characteristics。

微信蓝牙硬件的一些规定：

名称	描述
ServiceUUID	0xFEE7（该 uuid 经蓝牙官方授权）
Write Characteristics UUID	0xFEC7
Indicate Characteristics UUID	0xFEC8
Read Characteristics UUID	0xFEC9
Andriod RFCOMM UUID	e5b152ed-6b46-09e9-4678-665e9a972cbc

2.本工程已实现自动将 mac 地址填写到广播包中，开发过程中也可以通过

```
u8 ble_mac_addr[6];
```

```
get_ble_public_addr(ble_mac_addr,1,NULL);
```

获取到本机的 mac 地址。

3.如果调试的时候想使用自己固定的 mac 地址，注释掉

```
set_bt_config_check_callback(bt_info_init);
```

然后设置 test_addr 固定地址即可

```
u8 test_addr[6] = {0x2e, 0x3a, 0xba, 0x98, 0x22, 0x11};
```

```
void adv_reset_deal(void)
```

```
{
```

```
    u8 ble_mac_addr[6];
```

```
    /* get_ble_public_addr(ble_mac_addr,1,NULL); */
```

```
    get_ble_public_addr(ble_mac_addr,0,test_addr);
```

```
    puts("mac_addr\n");
```

```
    put_buf(ble_mac_addr,6);
```

```
    set_addr_into_adv(profile_adv_ind_data,sizeof(profile_adv_ind_data),ble_mac_addr);
```

```
}
```



❖ Ble_wechat 扫描连接

进入特定界面后，微信会开始扫描设备。当接收到符合微信定义的广播包时，微信可以扫描到设备。

❖ Ble_wechat 包结构

由定长包头和变长包体组成。

定长包头：

```
typedef struct
{
    unsigned char bMagicNumber;
    unsigned char bVer;
    unsigned short nLength;
    unsigned short nCmdId;
    unsigned short nSeq;
}__attribute__((packed)) Protobuf_Head;
```

字段	类型	说明
bMagicNumber	unsigned char	填 0xFE
bVer	unsigned char	包格式版本号，填 1
nLength	unsigned short	为包头+包体的长度
nCmdId	unsigned short	命令号
nSeq	unsigned short	递增。 一个 Req 对应一个 Resp，并且它们的 nSeq 相同，并且永不 为 0。 Push 的 nSeq 永远为 0；

变长包体

Protoalbuf 打包的二进制数据



❖ 命令列表

名称	描述
Auth	登录
Init	初始化
SendData	<p>设备发送数据给厂商或微信公众平台或微信客户端。</p> <p>当 type 为空或者等于 0 时，表示发送给厂商服务器。</p> <p>当 type 为 10001 时，表示发送给微信客户端 html5 设备会话界面。</p> <p>当 type 为其他时，表示发送给公众平台服务器。</p> <p>具体的定义请看 附录：微信公众平台 proto 文件。</p> <p>举个例子，type 等于 1 时，表示手环数据。</p>
RecvDataPush	<p>厂商或微信客户端或微信公众平台发送数据给设备</p> <p>当 type 为空或者等于 0 时，表示厂商发送设备。</p> <p>当 type 为 10001 时，表示收到微信客户端 html5 设备会话界面的数据。</p> <p>当 type 为其他时，表示公众平台发送给设备。</p> <p>具体的定义请看 附录：微信公众平台 proto 文件。</p> <p>举个例子，type 等于 1 时，表示手环数据。</p>
SwitchViewPush	微信客户端进入退出界面的通知
SwitchBackgroudPush	微信客户端进入退出后台的通知



❖ 身份验证

微信支持身份验证和加密，本工程使用不加密 MAC 地址认证的身份验证方法，降低接入的难度，加快开发产品的进程，

❖ 登录、初始化、数据发送

设备连上微信之后，需要发送 AuthReq，等收到成功的回包之后，接着还要发送 InitReq，并收到成功的回包之后，才能正常发送数据。

登录：发送 AuthReq

调用 void wechat_auth_fun(void)

```
{  
    u8 ble_txlen;  
  
    ble_txlen = wechat_auth_init_fun(CMD_AUTH,wechat_buf);  
    app_data_send(wechat_buf,ble_txlen);  
}
```

初始化：发送 InitReq

调用 void wechat_init_fun(void)

```
{  
    u8 ble_txlen;  
  
    ble_txlen = wechat_auth_init_fun(CMD_INIT,wechat_buf);  
    app_data_send(wechat_buf,ble_txlen);  
}
```

发送自定义数据：

调用 void wechat_send_fun(void)

```
{  
    u8 ble_txlen;  
  
    ble_txlen = wechat_auth_init_fun(CMD_SENDDAT,wechat_buf);
```



```
app_data_send(wechat_buf,ble_txlen);
```

```
}
```

数据接收端设定:

```
//发送到客户服务器
```

```
/* sendDatReq.has_type = false; */
```

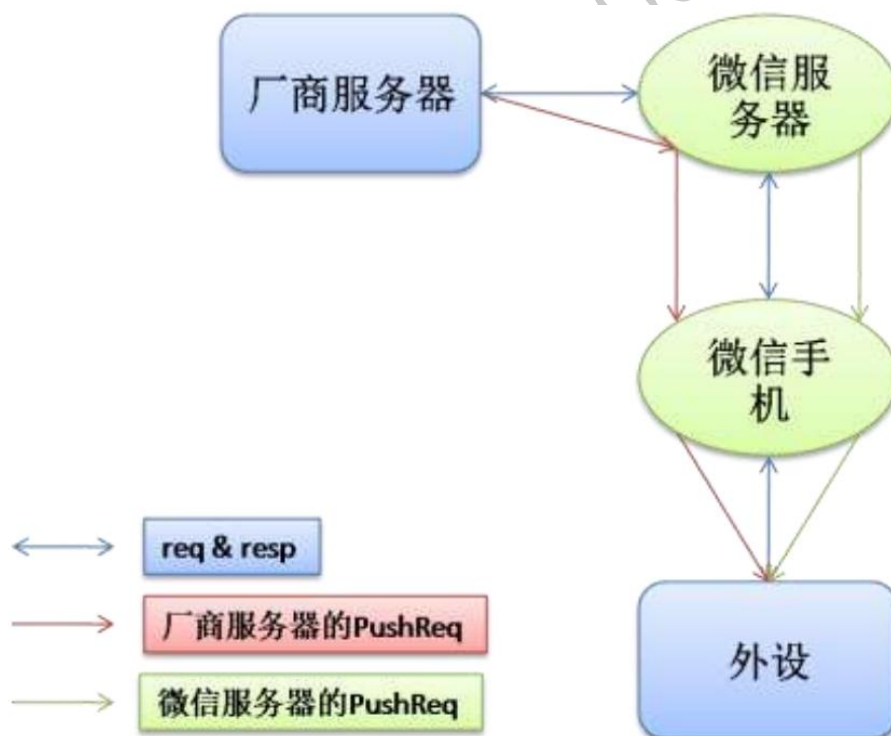
```
/* sendDatReq.type = (EmDeviceDataType)NULL; */
```

```
//发送到微信服务器
```

```
sendDatReq.has_type = true ;
```

```
sendDatReq.type = (EmDeviceDataType)EDDT_wxDeviceHtmlChatView;
```

下图是数据通信图:



接收解析自定义数据:

```
case cmdid_push_rcvData:
```



```
puts("get_data\n");  
recvDataPush=epb_unpack_recv_data_push(buffer+head_len,  
buffer_size-head_len);  
if(!recvDataPush)  
{  
    break;  
}  
BlueDemoHead *bledemohead = (BlueDemoHead*)recvDataPush->data.data;  
put_u16hex(_swap16(bledemohead->m_cmdid));  
/*  
    自行处理数据  
*/  
free(recvDataPush);  
break;
```