

8762CMF 模组产测流程

V 0.0.3

2019/4/2

修订历史 (Revision History)

日期	版本	修改	作者	Reviewer
2019/03/29	Draft v0.0.1	初稿	黄阳	
2019/04/01	Draft v0.0.2	修改 RF 测试和校准流程 增加 Hardware IO 配置, 写和读命令 增加 sample data 章节	黄阳	
2019/04/02	Draft v0.0.3	修改 RF 测试和校准流程位置 修改 mp command 协议, 增加长度字段 修改 sample data 增加 Hardware IO 配置模式选择 增加 Read public key 命令	黄阳	

Realtek
Confidential for
Xiaomi

目 录

修订历史 (Revision History)	2
目 录	3
表目录	5
图目录	6
词汇表	7
1 背景介绍	8
2 流程图	9
3 通讯方式	11
4 通讯协议	12
4.1 字节序	12
4.2 协议组成	12
5 产测流程和命令	13
5.1 Download firmware	13
5.2 Enter HCI test mode	13
5.2.1 Request	13
5.2.2 Response	13
5.3 RF test	13
5.4 RF calibration	14
5.5 Hardware IO test	14
5.5.1 Hardware IO configure	14
5.5.2 Hardware IO set	15
5.5.3 Hardware IO get	16
5.6 Program MAC	17
5.6.1 Request	17
5.6.2 Response	17
5.7 Restart DUT	18
5.7.1 Request	18
5.7.2 Response	18
5.8 Read and verify MAC	18

5.8.1 Request	18
5.8.2 Response	19
5.9 Create public key	19
5.9.1 Request	19
5.9.2 Response	19
5.10 Read and verify public key	20
5.10.1 Request	20
5.10.2 Response	20
5.11 Get certification.....	20
5.12 Program root certification.....	20
5.12.1 Request	20
5.12.2 Response	21
5.13 Program manufacture server certification.....	21
5.13.1 Request	21
5.13.2 Response	21
5.14 Program device certification	22
5.14.1 Request	22
5.14.2 Response	22
5.15 Verify certification chain.....	22
5.15.1 Request	22
5.15.2 Response	23
5.16 Check device information.....	23
5.16.1 Request	23
5.16.2 Response	23
5.17 操作码汇总	24
6 Sample data	25
6.1 Restart DUT.....	25
附录	26
附录一.....	26

表目录

表 3-1 串口参数.....	11
表 4-1 协议参数.....	12
表 5-27 Enter HCI test mode request.....	13
表 5-28 Enter HCI test mode response	13
表 5-1 Hardware IO configure request	14
表 5-2 Hardware IO configure response	15
表 5-3 Hardware IO set request.....	15
表 5-4 Hardware IO set response	16
表 5-5 Hardware IO get request	16
表 5-6 Hardware IO get response	16
表 5-7 Program MAC request	17
表 5-8 Program MAC response	17
表 5-9 Restart DUT request.....	18
表 5-10 Restart DUT response	18
表 5-11 Read MAC request	18
表 5-12 Read MAC response.....	19
表 5-13 Create public key request.....	19
表 5-14 Create public key response	19
表 5-15 Read public key request	20
表 5-16 Read public key response.....	20
表 5-17 Program root certification request.....	20
表 5-18 Program root certification response	21
表 5-19 Program manufacture server certification request.....	21
表 5-20 Program manufacture server certification response	21
表 5-21 Program device certification request.....	22
表 5-22 Program device certification response	22
表 5-23 Verify certification chain request.....	22
表 5-24 Verify certification chain response	23
表 5-25 Check device information request.....	23
表 5-26 Check device information response	23
表 5-29 操作码汇总	24

图目录

图 2-1 产测流程图	9
-------------------	---

Realtek
Confidential for
Xiaomi

词汇表

缩写	含义

Realtek
Confidential for
Xiaomi

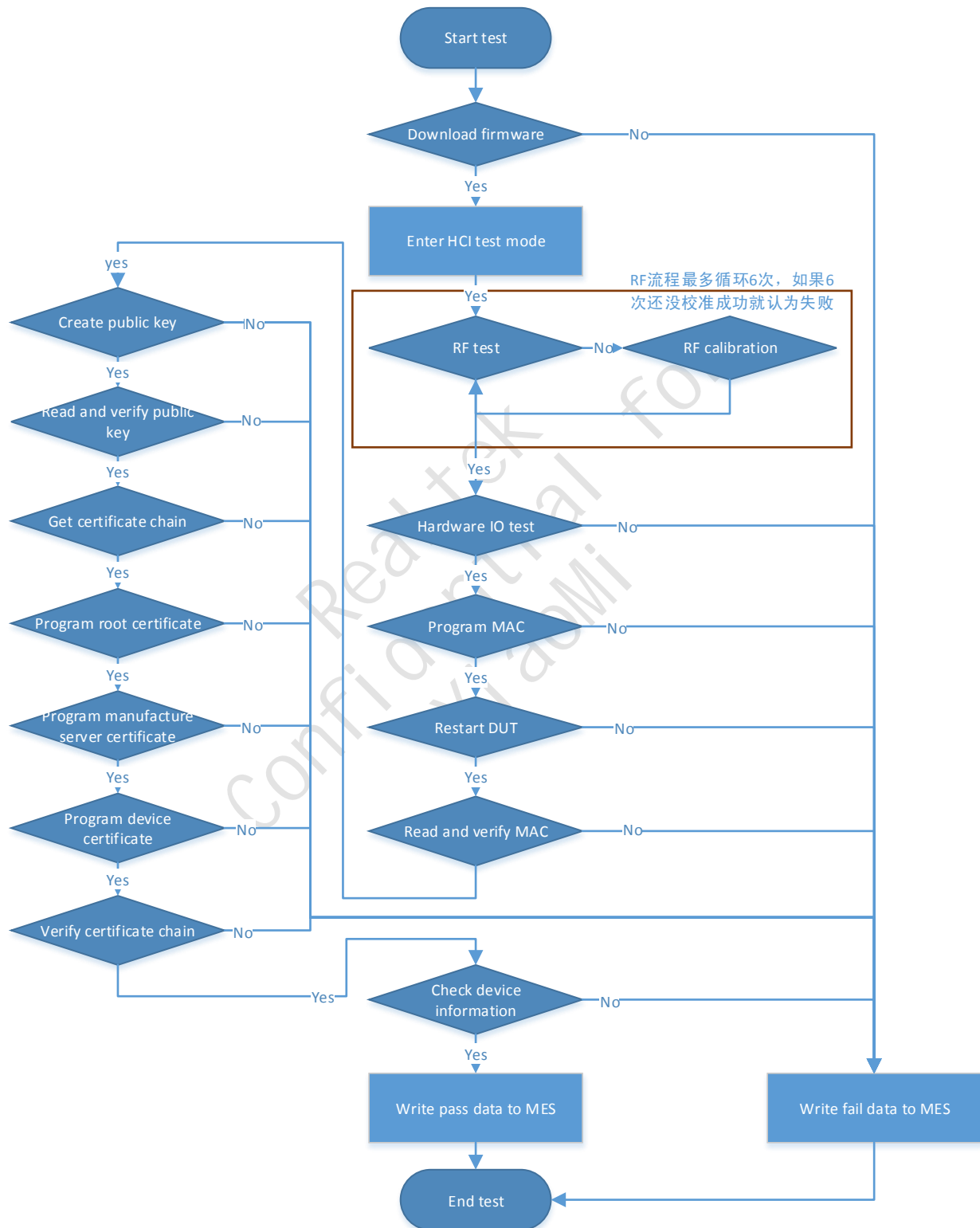
1 背景介绍

本产测流程用于指导小米工厂进行 8762CMF 模组的生产和测试,RF 测试部分需要进入 HCI 测试模式,其他测项串口协议参考[通讯协议](#)。

Realtek
Confidential for
Xiaomi

2 流程图

图 2-1 产测流程图



Realtek
Confidential for
Xiaomi

3 通讯方式

产测工具和 DUT 通过串口连接，参数如表 3-1。

表 3-1 串口参数

波特率	数据位	校验位	停止位	流控
115200	8	无	1	无

Realtek
Confidential for
Xiaomi

4 通讯协议

4.1 字节序

协议多字节字段使用小端模式。

4.2 协议组成

通讯协议由“包头+操作码+有效载荷长度+有效载荷+CRC 校验”5 部分组成，如表 4-1。协议采用 request-response 流程，测试端发起 request，DUT 端回复 response，同一条消息 request 和 response 使用的操作码相同。

表 4-1 协议参数

域	大小(字节)	描述
包头	1	固定为 0x87
操作码	2	具体操作的操作码
有效载荷长度	2	有效载荷长度
有效载荷	0-65535	具体的有效载荷
CRC 校验	2	CRC 校验，校验部分为“包头+操作码+有效载荷长度+有效载荷”

CRC 校验函数参考附录一。

5 产测流程和命令

5.1 Download firmware

使用 MP TOOL 下载固件，下载完毕之后重启 DUT 进行接下来的测试。

5.2 Enter HCI test mode

切换 DUT 进入 HCI 测试模式，发送命令收到回复之后建议等待 1s 以上再进行操作，以确保 DUT 切换完成。

5.2.1 Request

表 5-1 Enter HCI test mode request

域	大小(字节)	值	描述
操作码	2	0x110C	
有效载荷长度	2	0	无有效载荷

5.2.2 Response

表 5-2 Enter HCI test mode response

域	大小(字节)	值	描述
操作码	2	0x110C	
有效载荷长度	2	1	
HCI 切换状态	1	0x00, 0x01	0x00: 准备进入 HCI 测试模式 0x01: 无法进入 HCI 测试模式

5.3 RF test

使用 HCI 命令测试 RF 性能，RF test 失败之后需要进入 RF calibration 的流程。

5.4 RF calibration

当 RF 测试没有通过时，需要进行通过 HCI 命令进行 RF 的校准，RF 校准使用客户自己的校准设备。RF calibration 结束之后需要再进入 RF test 的流程，直到成功或者尝试 6 次还是失败。

5.5 Hardware IO test

用于测试模组的 IO 是否正常。Hardware IO 测试流程为先配置引脚功能，再设置引脚输出值，最后读出引脚的输入值。

5.5.1 Hardware IO configure

5.5.1.1 Request

表 5-3 Hardware IO configure request

域	大小(字节)	值	描述
操作码	2	0x1120	
有效载荷长度	2	6	
模式选择	1	0x00, 0x01	0x00: 第一功能，用于测试 gpio 0x01: 第二功能，用于测试 32Khz, MIC_BIAS 等
P0_0-P0_7 电平状态	1	0x00-0xFF	从最低位到最高位，依次代表 P0_0-P0_7 引脚功能，1 为输出，0 为输入
P1_0-P1_7 电平状态	1	0x00-0xFF	从最低位到最高位，依次代表 P1_0-P1_7 引脚功能，1 为输出，0 为输入
P2_0-P2_7 电平状态	1	0x00-0xFF	从最低位到最高位，依次代表 P2_0-P2_7 引脚功能，1 为输出，0 为输入
P3_0-P3_7 电平状态	1	0x00-0xFF	从最低位到最高位，依次代表 P3_0-P3_7 引脚功能，1 为输出，0 为输入
P4_0-P4_7 电平状态	1	0x00-0xFF	从最低位到最高位，依次代表 P4_0-P4_7 引脚功能，1 为输出，0 为输入

5.5.1.2 Response

表 5-4 Hardware IO configure response

域	大小(字节)	值	描述
操作码	2	0x1120	
有效载荷长度	2	1	
IO 配置状态	1	0x00, 0x01	0x00: IO 配置成功 0x01: IO 配置失败

5.5.2 Hardware IO set

5.5.2.1 Request

表 5-5 Hardware IO set request

域	大小(字节)	值	描述
操作码	2	0x1121	
有效载荷长度	2	5	
P0_0-P0_7 电平状态	1	0x00-0xFF	从最低位到最高位, 依次代表 P0_0-P0_7 输出电平状态, 1 为高电平, 0 为低电平
P1_0-P1_7 电平状态	1	0x00-0xFF	从最低位到最高位, 依次代表 P1_0-P1_7 输出电平状态, 1 为高电平, 0 为低电平
P2_0-P2_7 电平状态	1	0x00-0xFF	从最低位到最高位, 依次代表 P2_0-P2_7 输出电平状态, 1 为高电平, 0 为低电平
P3_0-P3_7 电平状态	1	0x00-0xFF	从最低位到最高位, 依次代表 P3_0-P3_7 输出电平状态, 1 为高电平, 0 为低电平
P4_0-P4_7 电平状态	1	0x00-0xFF	从最低位到最高位, 依次代表 P4_0-P4_7 输出电平状态, 1 为高电平, 0 为低电平

5.5.2.2 Response

表 5-6 Hardware IO set response

域	大小(字节)	值	描述
操作码	2	0x1121	
有效载荷长度	2	1	
IO 设置状态	1	0x00, 0x01	0x00: IO 设置成功 0x01: IO 设置失败

5.5.3 Hardware IO get

5.5.3.1 Request

表 5-7 Hardware IO get request

域	大小(字节)	值	描述
操作码	2	0x1122	
有效载荷长度	2	0	无有效载荷

5.5.3.2 Response

表 5-8 Hardware IO get response

域	大小(字节)	值	描述
操作码	2	0x1122	
有效载荷长度	2	5	
P0_0-P0_7 电平状态	1	0x00-0xFF	从最低位到最高位, 依次代表 P0_0-P0_7 输入电平状态, 1 为高电平, 0 为低电平
P1_0-P1_7 电平状态	1	0x00-0xFF	从最低位到最高位, 依次代表 P1_0-P1_7 输入电平状态, 1 为高电平, 0 为低电平

P2_0-P2_7 电平状态	1	0x00-0xFF	从最低位到最高位, 依次代表 P2_0-P2_7 输入电平状态, 1 为高电平, 0 为低电平
P3_0-P3_7 电平状态	1	0x00-0xFF	从最低位到最高位, 依次代表 P3_0-P3_7 输入电平状态, 1 为高电平, 0 为低电平
P4_0-P4_7 电平状态	1	0x00-0xFF	从最低位到最高位, 依次代表 P4_0-P4_7 输入电平状态, 1 为高电平, 0 为低电平

5.6 Program MAC

写 MAC 地址到 DUT, MAC 地址由小米统一分配。

5.6.1 Request

表 5-9 Program MAC request

域	大小(字节)	值	描述
操作码	1	0x1118	
有效载荷长度	2	6	
MAC 地址	6		写到 DUT 的蓝牙地址

5.6.2 Response

表 5-10 Program MAC response

域	大小(字节)	值	描述
操作码	2	0x1118	
有效载荷长度	2	1	
MAC 更新状态	1	0x00, 0x01	0x00: 更新 MAC 地址成功 0x01: 更新 MAC 地址失败

5.7 Restart DUT

重启 DUT，发送命令收到回复之后建议等待 1s 以上再进行操作，以确保 DUT 重启完成。

5.7.1 Request

表 5-11 Restart DUT request

域	大小(字节)	值	描述
操作码	2	0x1117	
有效载荷长度	2	0	无有效载荷

5.7.2 Response

表 5-12 Restart DUT response

域	大小(字节)	值	描述
操作码	2	0x1117	
有效载荷长度	2	1	
重启 DUT 状态	1	0x00, 0x01	0x00: 准备重启 0x01: 无法重启

5.8 Read and verify MAC

读取和验证 DUT 当前 MAC 地址。

5.8.1 Request

表 5-13 Read MAC request

域	大小(字节)	值	描述
操作码	2	0x1102	
有效载荷长度	2	0	无有效载荷

5.8.2 Response

表 5-14 Read MAC response

域	大小(字节)	值	描述
操作码	2	0x1102	
有效载荷长度	2	6	
MAC 地址	6		DUT 当前 MAC 地址

5.9 Create public key

通知 DUT 生成 public key。

5.9.1 Request

表 5-15 Create public key request

域	大小(字节)	值	描述
操作码	2	0x1420	
有效载荷长度	2	0	无有效载荷

5.9.2 Response

表 5-16 Create public key response

域	大小(字节)	值	描述
操作码	2	0x1420	
有效载荷长度	2	1	
生成状态	1	0x00, 0x01	0x00: 生成 public key 成功 0x01: 生成 public key 失败

5.10 Read and verify public key

对测端上位机读取 DUT 的 public key

5.10.1 Request

表 5-17 Read public key request

域	大小(字节)	值	描述
操作码	2	0x1421	
有效载荷长度	2	0	无有效载荷

5.10.2 Response

表 5-18 Read public key response

域	大小(字节)	值	描述
操作码	2	0x1421	
有效载荷长度	2	64	
Public key	64		DUT 生成的 public key

5.11 Get certification

对测端上位机通过 DUT 发送的 public key 去服务器获取小米的证书链。

5.12 Program root certification

写根证书到 DUT 中。

5.12.1 Request

表 5-19 Program root certification request

域	大小(字节)	值	描述
操作码	2	0x1422	

有效载荷长度	2	变长	根证书长度不固定
根证书	变长		需要写进 DUT 的根证书

5.12.2 Response

表 5-20 Program root certification response

域	大小(字节)	值	描述
操作码	2	0x1422	
有效载荷长度	2	1	
根证书写入状态	1	0x00, 0x01	0x00: 根证书写入成功 0x01: 根证书写入失败

5.13 Program manufacture server certification

写生产服务器证书到 DUT 中。

5.13.1 Request

表 5-21 Program manufacture server certification request

域	大小(字节)	值	描述
操作码	2	0x1423	
有效载荷长度	2	变长	生产服务器证书长度不固定
生产服务器证书	变长		需要写进 DUT 的生产服务器证书

5.13.2 Response

表 5-22 Program manufacture server certification response

域	大小(字节)	值	描述
操作码	2	0x1423	

有效载荷长度	1	1	
生产服务器写入状态	1	0x00, 0x01	0x00: 生产服务器证书写入成功 0x01: 生产服务器证书写入失败

5.14 Program device certification

写设备证书到 DUT 中。

5.14.1 Request

表 5-23 Program device certification request

域	大小(字节)	值	描述
操作码	2	0x1424	
有效载荷长度	2	变长	设备证书长度不固定
设备证书	变长		需要写进 DUT 的设备证书

5.14.2 Response

表 5-24 Program device certification response

域	大小(字节)	值	描述
操作码	2	0x1424	
有效载荷长度	2	1	
设备证书写入状态	1	0x00, 0x01	0x00: 设备证书写入成功 0x01: 设备证书写入失败

5.15 Verify certification chain

DUT 端校验证书链。

5.15.1 Request

表 5-25 Verify certification chain request

域	大小(字节)	值	描述
操作码	2	0x1425	
有效载荷长度	2	0	无有效载荷

5.15.2 Response

表 5-26 Verify certification chain response

域	大小(字节)	值	描述
操作码	2	0x1425	
有效载荷长度	1	1	
证书校验状态	1	0x00, 0x01	0x00: 证书校验成功 0x01: 证书校验失败

5.16 Check device information

检查 DUT 设备信息。

5.16.1 Request

表 5-27 Check device information request

域	大小(字节)	值	描述
操作码	2	0x1426	
有效载荷长度	2	0	无有效载荷

5.16.2 Response

表 5-28 Check device information response

域	大小(字节)	值	描述
操作码	2	0x1426	

有效载荷长度	2	20
产品 ID	2	DUT 的产品 ID
固件版本号	4	DUT 的固件版本号
MAC 地址	6	DUT 的 MAC 地址
设备 ID	8	DUT 的设备 ID

5.17 操作码汇总

表 5-29 操作码汇总

操作码	值
Read MAC	0x1102
Enter HCI test mode	0x110C
Restart DUT	0x1117
Program MAC	0x1118
Hardware IO configure	0x1120
Hardware IO set	0x1121
Hardware IO get	0x1122
Create public key	0x1420
Read public key	0x1421
Program root certification	0x1422
Program manufacture server certification	0x1423
Program device certification	0x1424
Verify certification chain	0x1425
Check device information	0x1426

6 Sample data

6.1 Restart DUT

命令参考[通讯协议](#)和 *Restart DUT*

Request: 87 17 11 00 00 e1 af

Response: 87 17 11 01 00 00 3e 88

Realtek
Confidential for
Xiaomi

附录

附录一

```

/*fcs table*/
static const uint16_t btxfcsTab[256] =
{
    0x0000, 0xc0c1, 0xc181, 0x0140, 0xc301, 0x03c0, 0x0280, 0xc241,
    0xc601, 0x06c0, 0x0780, 0xc741, 0x0500, 0xc5c1, 0xc481, 0x0440,
    0xcc01, 0x0cc0, 0x0d80, 0xcd41, 0x0f00, 0xcfc1, 0xce81, 0x0e40,
    0x0a00, 0xcac1, 0xcb81, 0x0b40, 0xc901, 0x09c0, 0x0880, 0xc841,
    0xd801, 0x18c0, 0x1980, 0xd941, 0x1b00, 0xdbc1, 0xda81, 0x1a40,
    0x1e00, 0xdec1, 0xdf81, 0x1f40, 0xdd01, 0x1dc0, 0x1c80, 0xdc41,
    0x1400, 0xd4c1, 0xd581, 0x1540, 0xd701, 0x17c0, 0x1680, 0xd641,
    0xd201, 0x12c0, 0x1380, 0xd341, 0x1100, 0xd1c1, 0xd081, 0x1040,
    0xf001, 0x30c0, 0x3180, 0xf141, 0x3300, 0xf3c1, 0xf281, 0x3240,
    0x3600, 0xf6c1, 0xf781, 0x3740, 0xf501, 0x35c0, 0x3480, 0xf441,
    0x3c00, 0xfcc1, 0xfd81, 0x3d40, 0xff01, 0x3fc0, 0x3e80, 0xfe41,
    0xfa01, 0x3ac0, 0x3b80, 0xfb41, 0x3900, 0xf9c1, 0xf881, 0x3840,
    0x2800, 0xe8c1, 0xe981, 0x2940, 0xeb01, 0x2bc0, 0x2a80, 0xea41,
    0xee01, 0x2ec0, 0x2f80, 0xef41, 0x2d00, 0xedc1, 0xec81, 0x2c40,
    0xe401, 0x24c0, 0x2580, 0xe541, 0x2700, 0xe7c1, 0xe681, 0x2640,
    0x2200, 0xe2c1, 0xe381, 0x2340, 0xe101, 0x21c0, 0x2080, 0xe041,
    0xa001, 0x60c0, 0x6180, 0xa141, 0x6300, 0xa3c1, 0xa281, 0x6240,
    0x6600, 0xa6c1, 0xa781, 0x6740, 0xa501, 0x65c0, 0x6480, 0xa441,
    0x6c00, 0xacc1, 0xad81, 0x6d40, 0xaf01, 0x6fc0, 0x6e80, 0xae41,
    0xaa01, 0x6ac0, 0x6b80, 0xab41, 0x6900, 0xa9c1, 0xa881, 0x6840,
    0x7800, 0xb8c1, 0xb981, 0x7940, 0xbb01, 0x7bc0, 0x7a80, 0xba41,
    0xbe01, 0x7ec0, 0x7f80, 0xbf41, 0x7d00, 0xbdc1, 0xbc81, 0x7c40,
    0xb401, 0x74c0, 0x7580, 0xb541, 0x7700, 0xb7c1, 0xb681, 0x7640,
    0x7200, 0xb2c1, 0xb381, 0x7340, 0xb101, 0x71c0, 0x7080, 0xb041,
    0x5000, 0x90c1, 0x9181, 0x5140, 0x9301, 0x53c0, 0x5280, 0x9241,
    0x9601, 0x56c0, 0x5780, 0x9741, 0x5500, 0x95c1, 0x9481, 0x5440,
    0x9c01, 0x5cc0, 0x5d80, 0x9d41, 0x5f00, 0x9fc1, 0x9e81, 0x5e40,
    0x5a00, 0x9ac1, 0x9b81, 0x5b40, 0x9901, 0x59c0, 0x5880, 0x9841,
    0x8801, 0x48c0, 0x4980, 0x8941, 0x4b00, 0x8bc1, 0x8a81, 0x4a40,
    0x4e00, 0x8ec1, 0x8f81, 0x4f40, 0x8d01, 0x4dc0, 0x4c80, 0x8c41,
    0x4400, 0x84c1, 0x8581, 0x4540, 0x8701, 0x47c0, 0x4680, 0x8641,
    0x8201, 0x42c0, 0x4380, 0x8341, 0x4100, 0x81c1, 0x8081, 0x4040
};
/**

```

```
* @brief Calculate a new fcs given the current fcs and the new data.
* Polynomial: X**0 + X**2 + X**15 + X16
*
* @param fcs: init or good final
* @param cp: data point
* @param len: length
*
* @return
*
*/
uint16_t btxfcs(uint16_t fcs, uint8_t *cp, uint16_t len)
{
    while (len--)
    {
        fcs = (fcs >> 8) ^ btxfcsTab[(fcs ^ *cp++) & 0xff];
    }

    return fcs;
}
```

Realtek
Confidential
Xiaomi