

# emTMF多音多频Wifi配置软件 用户手册

北京宇音天下科技有限公司

地址：北京市海淀区上地信息路7号数字传媒大厦503室



010-62986600



010-62988818



[www.tts168.com.cn](http://www.tts168.com.cn)



宇音天下官方订阅号

## 重要声明

### 版权声明

版权归北京宇音天下科技有限公司所有，保留所有权利。

### 商标声明

北京宇音天下科技有限公司的产品是北京宇音天下科技有限公司专有。在提及其他公司及其产品时将使用各自公司所拥有的商标，这种使用的目的仅限于引用。本文档可能涉及北京宇音天下科技有限公司的专利（或正在申请的专利）、商标、版权或其他知识产权，除非得到北京宇音天下科技有限公司的明确书面许可协议，本文档不授予使用这些专利（或正在申请的专利）、商标、版权或其他知识产权的任何许可协议。

### 不作保证声明

北京宇音天下科技有限公司不对此文档中的任何内容作任何明示或暗示的陈述或保证，而且不对特定目的的适销性及适用性或者任何间接、特殊或连带的损失承担任何责任。本手册内容若有变动，恕不另行通知。本手册例子中所用的公司、人名和数据若非特别声明，均属虚构。未得到北京宇音天下科技有限公司明确的书面许可，不得为任何目的、以任何形式或手段（电子的或机械的）复制或传播手册的任何部分。

### 保密声明

本文档（包括任何附件）包含的信息是保密信息。接收人了解其获得的本文档是保密的，除用于规定的目的外不得用于任何目的，也不得将本文档泄露给任何第三方。

本软件产品受最终用户许可协议（EULA）中所述条款和条件的约束，该协议位于产品文档和/或软件产品的联机文档中，使用本产品，表明您已阅读并接受了EULA的条款。

版权所有：北京宇音天下科技有限公司

## 目录

<b>第一章 引言 .....</b>	<b>4</b>
1.1 系统概述 .....	4
1.2 术语定义 .....	4
<b>第二章 主要应用领域 .....</b>	<b>4</b>
<b>第三章 EMTMF 简介 .....</b>	<b>4</b>
3.1 EMTMF 系统构成 .....	4
3.2 EMTMF 系统使用说明 .....	5
<b>第四章 EMTMF SDK 软件简介 .....</b>	<b>5</b>
4.1 EMTMF 功能介绍 .....	5
4.2 EMTMF SDK 组成 .....	6
4.3 EMTMF SDK 业务流程 .....	6
4.4 运行环境 .....	7
4.4.1 软件环境 .....	7
4.4.2 硬件环境 .....	7
<b>第五章 EMTMF 终端 SDK 集成调用接口 .....</b>	<b>8</b>
5.1 <i>emTMF</i> 音频解码接口 .....	8
5.2 <i>emTMF</i> 终端与 APP 之间通过 TCP 通信上报 wifi 配置结果 .....	12
5.3 测试用例 .....	15
<b>第六章 EMTMF APP 端 SDK 集成调用接口 .....</b>	<b>15</b>
<b>第七章 常用 WIFI 配置应用指标对比表 .....</b>	<b>15</b>
7.1 对比表 .....	15
<b>第八章 软件配置效果 .....</b>	<b>16</b>
8.1 配置效果 .....	16
<b>第九章 合作&amp;联系方式 .....</b>	<b>16</b>

## 第一章 引言

### 1.1 系统概述

emTMF 是一套基于声波的多音多频智能硬件终端 Wifi 配置软件，emTMF 核心技术采用宇音天下自主研发的 MTMF (Multi Tone Multi Frequency) 编码技术，MTMF 技术可以把任意的字符串编码成不同频率组合的声波信号，接收端在接收到此声波信号后，通过 MTMF 的解码规则可以获取输入的字符串信息。

emTMF SDK 是宇音天下提供给智能终端厂家的 emTMF 开发工具包，用户集成 emTMF SDK 后可在自己的产品上实现通过 emTMF 配置终端上网参数功能。

### 1.2 术语定义

- **MTMF:** 多音多频，指由多种不同的频率组合构成多种不同音频信号的技术。
- **智能设备:** 包括智能手机，平板电脑等可以发送声波信号的设备。
- **智能终端:** 待接入 WIFI 网络的智能硬件终端。
- **编码:** 根据预先规定的方法，将字符串信息转变成声波信号。
- **解码:** 根据预先规定的方法，将声波信号转变为字符串信息。
- **配置距离:** 智能设备扬声器与智能硬件终端麦克风的直线距离。

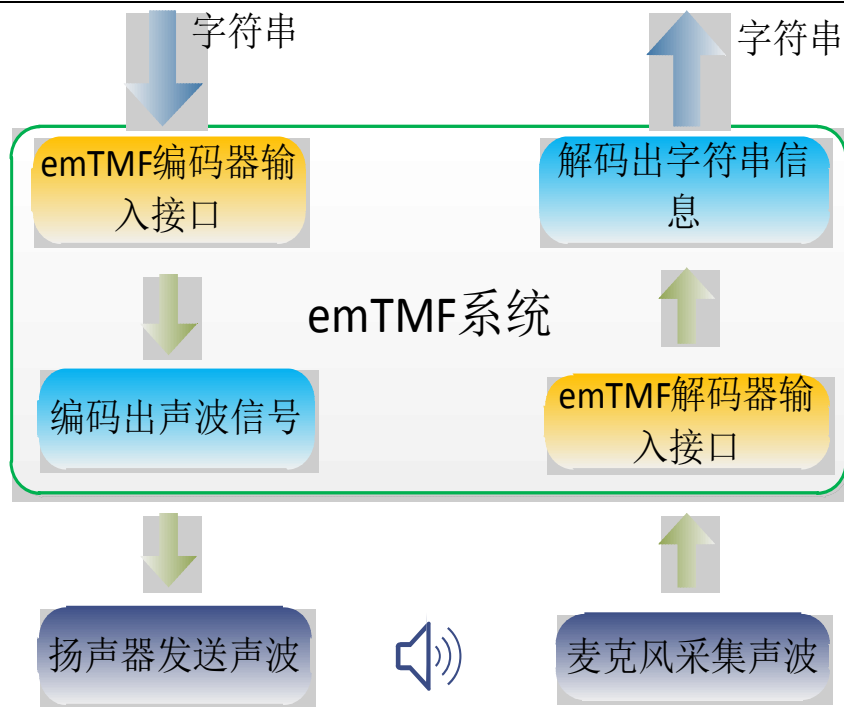
## 第二章 主要应用领域

- WIFI 信息传输
- 数据查询系统（身份证查询，快递单查询，图书馆藏信息查询）
- 考勤，会议签到
- 门禁系统
- 小额支付系统
- 电子票据
- 资料分享

## 第三章 emTMF 简介

### 3.1 emTMF 系统构成

emTMF 系统主要是由智能设备端的编码部分和智能硬件终端的解码部分组成，以下是 emTMF 的系统构成框图：



### 3.2 emTMF 系统使用说明

- 1) 系统理论上支持最大 50cm 的配置距离，必须满足智能设备音量最大的条件。目前，配置距离在 20-30cm 左右最优，智能设备的音量在 60%-80%最优。
- 2) 根据配置 WIFI 网络的 SSID 和密码长度的不同，生成的声波信号持续时间不同，目前，配置 WIFI 网络的声波信号持续时间在 3s-5s 左右，此处不包含声波信号持续时间只是音频声音的长度，不包括智能硬件终端配置成功后连接 WIFI 网络的时间。
- 3) 系统可实现一键配置，即用户只需在智能设备 APP 上点击一次发送声波，便可实现整个配置操作，但是智能硬件终端在上电后或无法连接 WIFI 时需打开录音接口，保证在配置 WIFI 网络时，智能硬件终端保持在录音状态。
- 4) 系统目前只支持录音采样率为 16KHZ, 因为不同的频率组合产生的声波信号是不同的，对于解码的效果也是有一定的影响，目前最优的采样为 16KHZ，不支持其他采样频率。

## 第四章 emTMF SDK 软件简介

### 4.1 emTMF 功能介绍

emTMF 主要功能包括：

- **任意字符输入：** 声波信号可携带任意字符信息，帮助用户实现各种功能。
- **配置反馈提示：** 在配置出现问题时，智能终端会解析出相应的错误编码，例如距离太远，或者是噪音太大等。
- **一键配置：** 智能硬件终端在上电后开启录音，只要智能设备（如手机 APP）发送声波，智能终端可

实现启动检测，自动进入配置程序，用户无需按键打开智能终端。

## 4.2 emTMF SDK 组成

宇音天下提供的 emTMF SDK 软件包由设备端（如 APP）SDK（分 IOS 和 Android）和终端 SDK 两部分组成，方便用户将 emTMF 功能集成到应用中。

app 端 SDK 功能：

- 1) 启动时向宇音天下后台进行授权认证，并根据后台返回的认证决定 sdk 是否可用。
- 2) 供用户 app 调用，传入 wifi 上网配置参数，SDK 将上网配置参数调制成声波发送。
- 3) 声波发送后启动监听服务接收终端上报的配置结果。

终端端 SDK 功能：

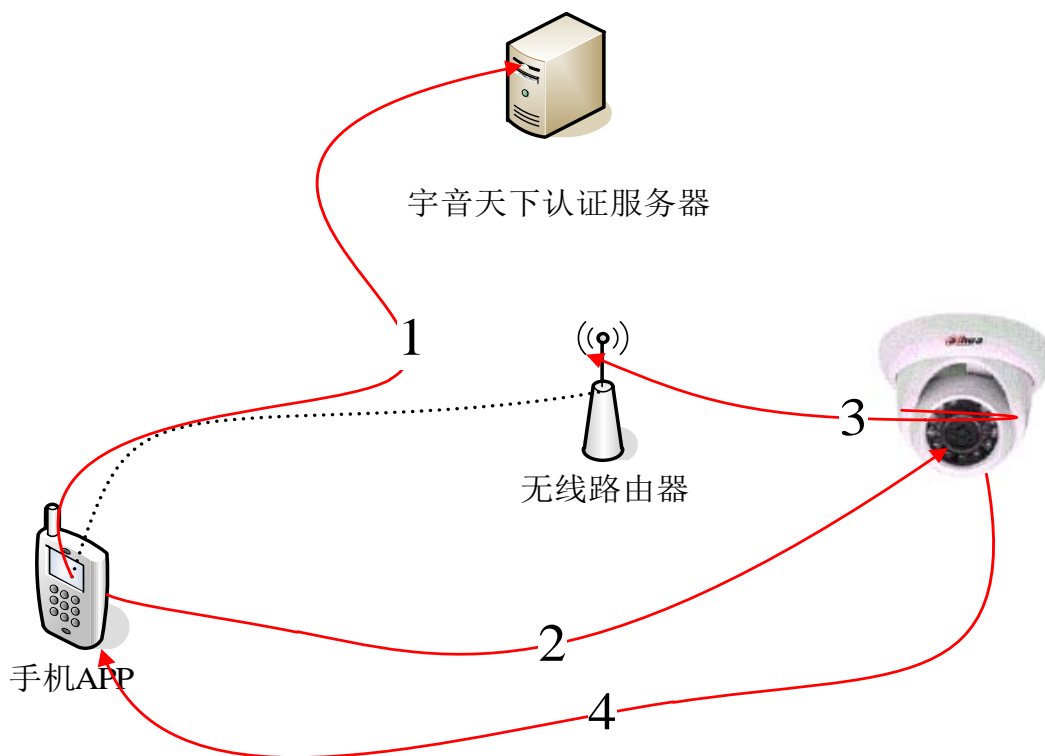
- 1) 解析声波得到上网配置参数返回给上层应用，由上层应用配置网络
- 2) 提供配置结果上报给 app 功能，由上层应用调用通知 app 配置结果。

## 4.3 emTMF SDK 业务流程

为了保证 emTMF SDK 软件包被合法授权使用，用户在集成 emTMF SDK 开发包时必须按照宇音天下软件授权认证流程对 SDK 进行认证，emTMF 配置业务流程如下：

- 1) emTMF APP SDK 初始化连接认证服务器验证 SDK 使用合法性
- 2) emTMF APP 客户端的音频编码及发送（播放）；
- 3) emTMF 模组 SDK 的音频接收（录音）及解码；  
将解码得到的上网配置帐号提交给上层应用，由上层应用实现 wifi 上网配置。
- 4) 模组用 TCP 方式给 APP 端发送 wifi 配置结果；

如下图所示：



## 4.4 运行环境

### 4.4.1 软件环境

智能设备的 APP 开发程序支持 Android, IOS, WINDOWS 等操作系统。

智能硬件终端支持主流的嵌入式操作系统, 如 Linux, Android, IOS, RTOS , Vxworks 等。

### 4.4.2 硬件环境

运行本系统所需要的硬件配置包括:

- 智能设备 (如手机等) 必须具备扬声器;
- 智能硬件终端必须具备麦克风, 灵敏度要求在 $-33\text{dB}$  至 $-40\text{dB}$ , 频响在  $60\text{dB}$ , 硬件处理器内存占用  $20\text{K}$  字节, 主频大约在  $100\text{MHZ}$  即可, 麦克风录音采样率支持  $16000\text{HZ}$ 。

## 第五章 emTMF 终端 sdk 集成调用接口

emTMF 终端 SDK 包集成调用接口分为 emTMF 音频解码和终端与 APP 之间通过 TCP 通信认证合法性两大部分。

### 5.1 emTMF 音频解码接口

接口定义参照终端演示代码头文件#include "emTMF\_Api.h"中，如下

#### ■ emTMF 调用接口

```
/*  
// Description : emTMFSDKInit 解码程序初始化  
// Param1:IN - 音频回调接口指针  
// Param2:IN - 厂商名称  
// Param3:IN - 客户名称  
// Param4:IN - 产品型号  
// Param5:IN - license  
// return: emTMF_InitRET_E  
// Global resource dependence :  
// Author :  
// Date: 2016-02-25  
// Note : 厂商名称、客户名称、产品型号、license 是软件授权时，用户提供的，若传入的参数与授权时用户提供的不一致则初始化失败，以后的功能都不能使用。  
emTMF_InitRET_E emTMFSDKInit (emTMF_Audio_Source *Audio_T,unsigned char*Manufacturer,unsigned char*Client,unsigned char*ProductModel,unsigned char*License); //初始化 emTMF  
  
/*  
// Description : emTMFSDKExit 退出  
// return:  
// Global resource dependence :  
// Author :  
// Date: 2016-02-25  
// Note :  
void emTMFSDKExit(void); //
```



```
// Param1: IN 给 emTMF 程序分配的内存空间首地址
// Param2: IN 给 emTMF 程序分配的内存空间大小
// Param3: OUT 输出解码结果
// return: 返回解码状态, = 0 配置成功
// Global resource dependence :
// Author :
// Date: 2016-02-25
// Note :
/*****/
emTMF_RET_E emTMFParseWifi( uint8 *Memory, //开辟的内存地址
                           uint32 MemoryLen, // 开 辟 的 内 存 总 长 度 ,
EMTMF_MEMORY_ALL
                           emTMF_Out_T *OutTxt); //输出文本 (wifi 用户名及密码)
```

■ emTMF 调用接口用到的结构定义

```
typedef enum
{
    EMTMF_STATE_OK, //成功
    EMTMF_STATE_AUDIO_NULL, //音频回调指针为空
    EMTMF_STATE_LICENSE_INVALID //license 非法
}emTMF_InitRET_E;
```

```
typedef enum
{
    EMTMF_STATE_ZERO, //解码成功
    EMTMF_STATE_ONE, //解码错, 重新配置
    EMTMF_STATE_TWO, //能量低, 靠近一些
    EMTMF_STATE_THREE, //有噪音, 保持安静
    EMTMF_STATE_FOUR, //破音, 离远一些
    EMTMF_STATE_FIVE, //发送超时或未发送
    EMTMF_STATE_SIX, //录音失败 (打开音频设备失败, 或获取音频数据失败)
    EMTMF_STATE_SEVEN, //取消录音
    EMTMF_STATE_EIGHT, //客户标识或 ID 与 APP 端不匹配
    EMTMF_STATE_NINE, //emTMF 初始化未通过
    EMTMF_STATE_ERROR=255 // 其他错误
}emTMF_RET_E;
```

```
typedef struct
{
    uint8 ssid[32+4]; //SSID, 最大 32 字节
```

```
uint8 password[64+4]; //密码, 最大 64 字节  
}emTMF_Out_T;
```

■ emTMF 系统回调接口---音频回调接口

typedef struct

```
{  
    /*****/  
    // Description : 打开音频设备, 单声道, 采样率:16000,采样位数:16bit  
    // Param1:  
    // return: 设备打开成功返回 TRUE(1),设备打开失败返回 FALSE(0)  
    // Global resource dependence :  
    // Author :  
    // Date: 2016-02-25  
    // Note :  
    /*****/  
    char (*startRecord) (void); //打开录音  
    /*****/  
    // Description : 获取录音数据  
    // Param1: (out)录音数据输出缓冲区地址  
    // Param2: (in) 想获取音频数据的长度, (EMTMF_AUDIOLEN),  
    // return: 实际获取到的音频数据的长度, 此值必须小于 Param2 输入的值, 若此值<0, 表示录音失败,  
    若此值等于 0 表示主动取消录音  
    // Global resource dependence :  
    // Author :  
    // Date: 2016-02-25  
    // Note :  
    /*****/  
    int32 (*getRecordData) (uint8* audioData, uint32 sizeInBytes);//读取音频数据 如 fread  
    /*****/  
    // Description : 关闭录音  
    // Param1:  
    // return:  
    // Global resource dependence :  
    // Author : xuchuanjun  
    // Date: 2016-02-25  
    // Note :  
    /*****/  
    void (*stopRecord) (void); //关闭录音  
    //关闭录音 如 fclose  
} emTMF_Audio_Source;
```

■ emTMF 测试例程

```
/*
// Description : 测试例程
// Param1:
// return:
// Global resource dependence :
// Author :
// Date: 2016-02-25
// Note :
void emTMF_Test(void)
{
    uint8 *MemBuf=NULL;
    emTMF_Out_T OutTxt={0};
    emTMF_RET_E Fsk_T;
    emTMF_InitRET_E InitRet = EMTMF_STATE_OK;

    emTMF_Audio_Source Audio_T={0};

    MemBuf = (uint8 *)malloc(sizeof(uint8)*EMTMF_MEMORY_ALL);

    Audio_T.startRecord = _emTMF_AudioRecord_Start;
    Audio_T.stopRecord = _emTMF_Audio_Stop;
    Audio_T.getRecordData= _emTMF_Audio_GetInputBuffer;

    InitRet =
emTMFSDKInit(&Audio_T,EM_SYN_MANUFACTURER,EM_SYN_CLIENT_NAME,EM_SYN_PRODUCT_MODEL,EM_SY
N_DES_LICENSE);

    Fsk_T = emTMFParseWifi(MemBuf,EMTMF_MEMORY_ALL,&OutTxt);

    if(Fsk_T == EMTMF_STATE_ZERO)
    {
        SYN_Printf("FSK #####s\r\n",OutTxt.ssid);
        SYN_Printf("FSK #####s\r\n",OutTxt.password);
    }
    else if(Fsk_T == EMTMF_STATE_ONE)
    {
        SYN_Printf("解码错误");
    }
    else if(Fsk_T == EMTMF_STATE_TWO)
    {

```

```
        SYN_Printf("能量低");
    }
    else if(Fsk_T == EMTMF_STATE_THREE)
    {
        SYN_Printf("有噪音, 请保持安静");
    }
    else if(Fsk_T == EMTMF_STATE_FOUR)
    {
        SYN_Printf("破音, 请离远一点");
    }
    else if(Fsk_T == EMTMF_STATE_FIVE)
    {
        SYN_Printf("有噪音, 请保持安静");
    }
    else if(Fsk_T == EMTMF_STATE_SIX)
    {
        SYN_Printf("录音失败");
    }
    else if(Fsk_T == EMTMF_STATE_SEVEN)
    {
        SYN_Printf("主动取消录音, 退出配置");
    }
    else if(Fsk_T == EMTMF_STATE_EIGHT)
    {
        SYN_Printf("客户标识或 ID 与 APP 端不匹配");
    }
    else if(Fsk_T == EMTMF_STATE_NINE)
    {
        SYN_Printf("emTMF 初始化未通过");
    }
    else
    {
        SYN_Printf("未知错误");
    }
    emTMFSDKExit();
}
```

## 5.2 emTMF 终端与 APP 之间通过 TCP 通信上报 wifi 配置结果

此接口同时具有验证终端 emTMF SDK 授权合法性, 若用户恶意不调用此接口, 多次使用后会致 APP 端 emTMF SDK 授权失效, 会导致当前所有使用该版本的 emTMF sdk 用户 emTMF 功能失效。合法购买授权的用户必须按照接口规范调用 report 接口向 app 通报 wifi 配置结果。

接口都在头文件#include "SYN\_Register\_API.h"中, 如下

■ 设置终端 emTMF 上报接口

```
/*  
// Description : 设置 emTMF 发送数据及打印 log 数据回调接口  
// Param1:IN 数据发送回调接口  
// return: >0 初始化成功, <0 初始化失败  
// Global resource dependence :  
// Author :  
// Date: 2016-02-25  
// Note :  
*/  
int emSDKSetReportCallback(emSend_Callback*Syn_Send_handle);  
  
/*  
// Description : emTMF 软件包发送配置结果给 APP  
// Param1:IN - 设备 ID 字符串格式, 最大 32 字节  
// Param2:IN - MAC 地址, 字符串格式, 12 字节  
// return: emTMF_Reg_E  
// Global resource dependence :  
// Author :  
// Date: 2016-02-25  
// Note : 若函数返回的不是 EMTMF_ERR_OK, 则需要特别注意。若多次使用 emTMF 配置, 此函数都不能正  
正确返回 EMTMF_ERR_OK, 则会影响 APP SDK 的正常使用。  
*/  
emTMF_Reg_E emTMFreportToApp(unsigned char*DeviceId,unsigned char*MAC);
```

■ emTMF 发送数据及打印 log 数据回调接口

```
typedef struct  
{  
    /**  
    * Do printf log  
    */  
    int (*printfLog) (const char* format, ...);  
  
    int (*reportData)( /* 0 表示发送--接收成功, 其他值表示发送--接收失败*/  
    unsigned int IpAddr, /* IN - ip address 网络二进制地址,like 0x623e397b 表示 123.57.62.98*/  
    unsigned short Port, /* IN - port number 网络二进制模式,like 0xFD1F 表示 8189*/  
    unsigned char*SendBuf,/* IN - 要发送的数据首地址*/  
    unsigned int SendLen, /* IN - 要发送的数据长度*/  
    unsigned char*RecvBuf,/* OUT - 接收到的数据首地址*/  
    unsigned int *RecvLen /* OUT - 接收到的数据长度*/  
);
```

```
}emSend_Callback;
```

■ emTMF 认证接口用到的结构定义

```
typedef enum
{
    EMTMF_ERR_OK,           //正确
    EMTMF_ERR_PARAM,       //参数传递错误
    EMTMF_ERR_MAKE,        //构建数据错误
    EMTMF_ERR_DATA,        //发送的数据非法
    EMTMF_ERR_FORM,        //发送的数据格式不对
    EMTMF_ERR_OTHER,       //App 返回的其他错误
    EMTMF_ERR_SEND_RECV    //3 次发送或接收数据错误
}emTMF_Reg_E;
```

■ emTMF 认证接口调用示例

```
int SYN_emTMF_RegisterTest(void)
{
    emTMF_Reg_E Ret=1;
    emSend_Callback SocketTcpSend;
    unsigned char MacAddr[16];
    unsigned char DeviceID[24];

    SYN_Printf("%s,%d", __FILE__, __LINE__);

    SocketTcpSend.reportData = SYN_emTMFSendRegisterToAPP;
    SocketTcpSend.printfLog = SYN_Printf;

    emSDKSetReportCallback(&SocketTcpSend);
    SYN_Memset(DeviceID, 0, 24);
    SYN_Memset(MacAddr, 0, 16);

    my_do_getDeviceId(DeviceID);
    my_do_getMAC(MacAddr);

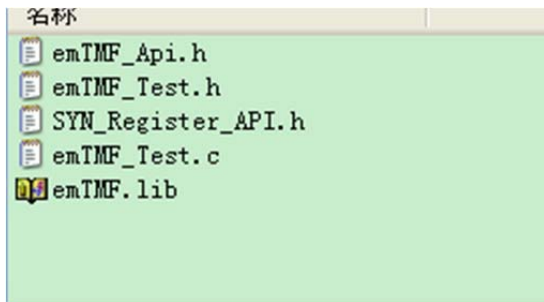
    SYN_Printf("%s,%d, DeviceID=%s, MacAddr=%s", __FILE__, __LINE__, DeviceID, MacAddr);

    Ret = emTMFreportToApp(DeviceID, MacAddr);
    SYN_Printf("%s ,Ret=%d\r\n", __FILE__, Ret);

    return Ret;
}
```

## 5.3 测试用例

终端设备 SDK 包包含以下文件



其中文件 emTMF\_Test.c 中有详细的调用示例，请参考 SDK 包技术资料

```
int main(void)
{
    //调用 emTMF 音频解码
    emTMF_Test();
    //连网成功后，调用此接口发注册数据给 APP, 若不调用此接口，使用多次以后 APP 将失效
    SYN_emTMF_RegisterTest();
    printf("sdf\r\n");
}
```

## 第六章 emTMF APP 端 SDK 集成调用接口

参考《emTMF 多音多频 Wifi 配置软件\_开发指南\_Android SDK.doc》

## 第七章 常用 wifi 配置应用指标对比表

### 7.1 对比表

由于目前家居环境下 WIFI 路由器越来越多，电磁环境越来越复杂，导致 Smartlink\airkiss\easylink 等市场上主流的配置方式，一次性配置成功的概率越来越小，实际配置时间越来越长；而其它非主流的技术，也由于效率差、可操作性、用户操作体验差等原因，正逐渐退出主流市场，越来越变成辅助性操作。下表是各类 WIFI 配置方式的全方位对比：

应用指标 配置方式	易操作性	配置时长	配置成功率	用户直观性	一次性配置多个设备
emTMF	容易	短	高	简单明了	支持一配多
Smartlink	容易	长	差	隐蔽	支持一配多
Airkiss	容易	长	差	隐蔽	支持一配多
Easylink	容易	长	差	隐蔽	支持一配多
Soft AP	难	长	高	隐蔽	只能一对一
QSS/WPS	一般	短	高	简单明了	支持一配多

## 第八章 软件配置效果

### 8.1 配置效果

- 请联系我司购买演示板。
- 请访问乐视网 <http://www.letv.com/ptv/vplay/24258324.html>，观看多音多频配置的演示视频。

## 第九章 合作&联系方式

emTMF 多音多频智能硬件 Wifi 配置软件针对用户的智能平台进行定制，销售方式为：技术授权

### 联系方式：

北京宇音天下科技有限公司（总部）	张红英	010-62986600	15321801789
珠海宇音天下科技有限公司（子公司）	何海滨	0756-6918288	18575636388
深圳办事处		0755-22197482	