

---

# RDA5981 SDK

## 开发手册

说明：版本为本文档的版本，SDK 版本为 SDK 的版本。

更新记录：

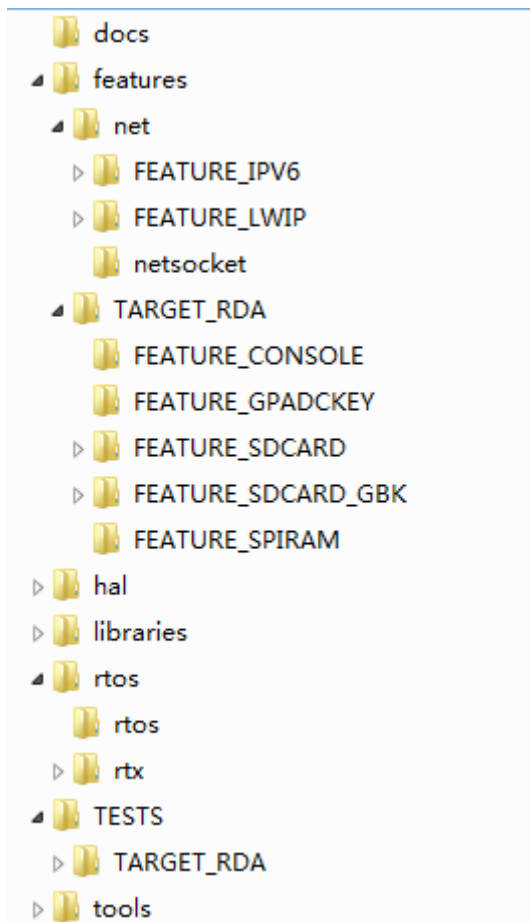
日期	版本	SDK 版本	摘要
2017-02-20	1.00	1.10	创建
2017-05-21	1.10	1.20	增加 features、test case、资料和 DEMO 板介绍。
2017-06-02	1.11	1.21	增加 features、test case 介绍。

## 目 录

1	SDK 介绍.....	1
2	Windows 开发环境.....	2
3	features 介绍.....	5
4	test case 介绍.....	9
5	资料介绍.....	11
6	DEMO 板介绍.....	12

# 1 SDK 介绍

RDA5981 SDK 目录:



## 1、WiFi Stack

目录 features/FEATURE\_LWIP 为 WiFi Stack + LWIP, 实现 WLAN 和 TCP/IP 联网功能。

## 2、HAL

目录 hal/targets/TARGET\_RDA 为硬件虚拟层, 其中包括各外设底层驱动。

## 3、RTX

目录 rtos 是 ARM 公司提供的 CMSIS-RTOS RTX 源码, 是一个基于时间片和优先级的实时 OS。

## 4、features

目录 features/TARGET\_RDA 下提供 RDA5981 的一些

## 5、Test Case

目录 TESTS/TARGET\_RDA 包含了若干针对 RDA5981 的测试用例, 可以对外设和协议栈分别进行测试。

## 2 Windows 开发环境

Windows 开发环境需要预先安装以下软件：

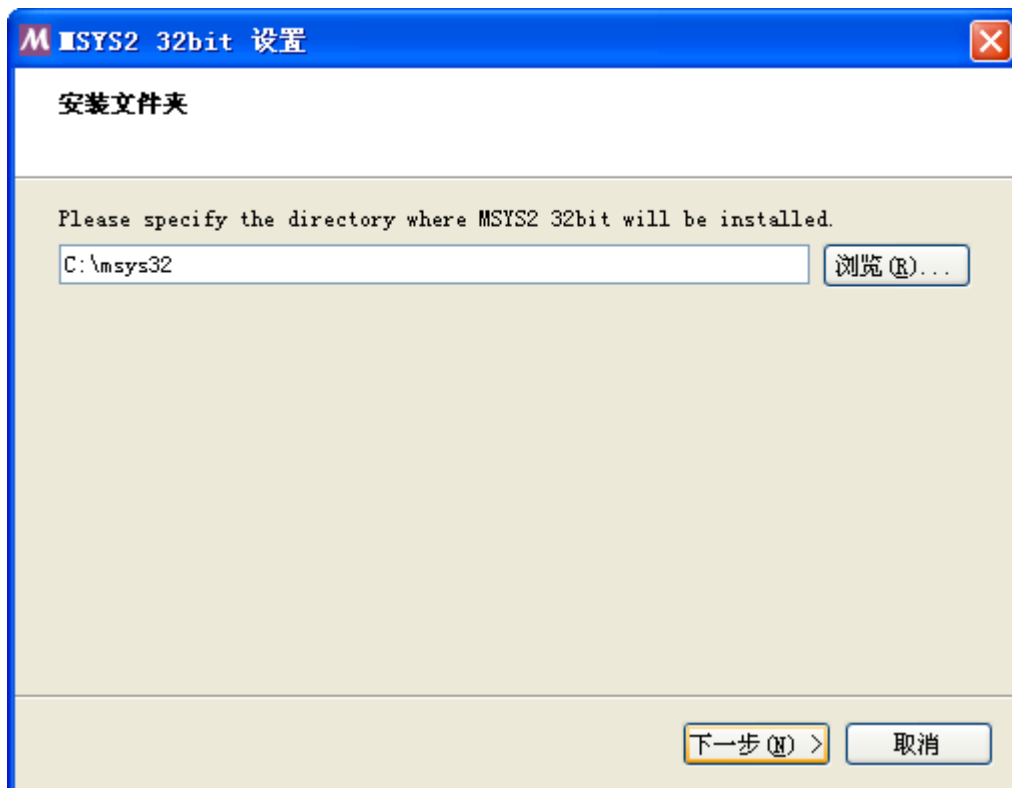
- Keil MDK (>= 5.17)
- Python (2.7.10)

除此之外，建议使用命令行工具 Msys2，更方便地进行编译。下面将详细介绍如何配置 mbed 编译环境。

### 1、下载安装 Msys2:

官网：[www.msys2.org](http://www.msys2.org)

根据需要下载相应版本并安装。



### 2、安装配置 mbed-cli 命令行工具

假设 Keil MDK 和 Python 安装目录分别为：C:/Keil\_v5 与 C:/Python27

- 将路径 C:/Python27 和 C:/Python27/Scripts 添加到系统环境变量 PATH
- 打开 msys2 软件，执行 `python --version` 和 `pip --version` 保证 python 工具可用
- 执行 `pip install mbed-cli` 安装 mbed-cli 工具，安装完成后执行 `mbed --version` 保证工具可用

- 执行 `pip install -r [SDK 根目录]/requirements.txt` 安装其他编译相关的工具
- 执行 `mbed config --global ARM_PATH C:/Keil_v5/ARM/ARMCC` 设置编译器路径

### 3、编译 SDK

SDK 中包含若干 test case，以 timer 为例，需要在 SDK 根目录执行编译命令：

```
mbed compile -m UNO_91H -t ARM --source TESTS/TARGET_RDA/timer/ --source ./
```

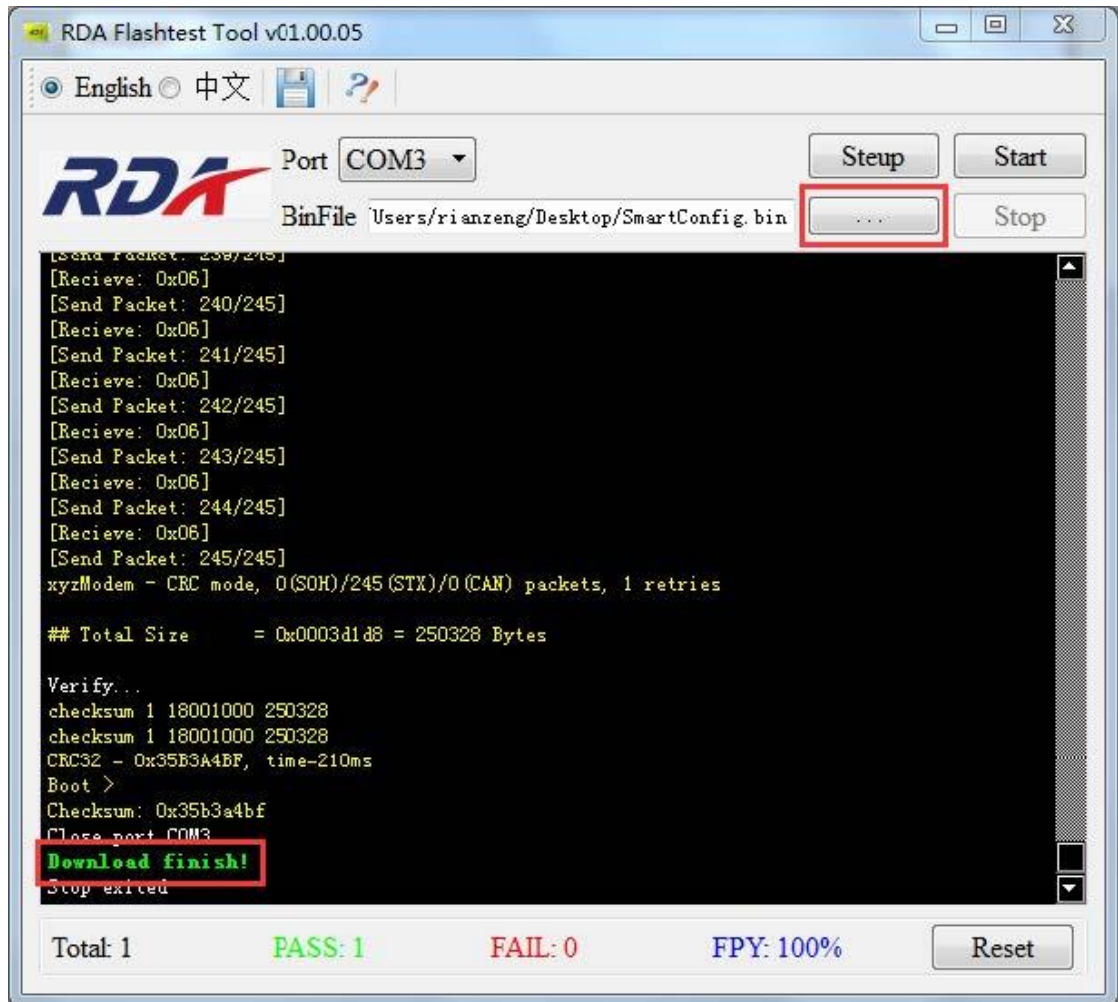
其中各参数选项含义：

- `-m` 指定 MCU Target 名称
- `-t` 指定编译器工具链
- `--source` 指定参与编译的源码目录
- `-c` 编译前 clean BUILD 目录

编译完成后将产生文件：`.\BUILD\UNO_91H\ARM\timer.bin`，可以将其烧录到硬件板卡 Flash 中运行。

### 4、烧录 bin 文件

- 首先在我司 bbs 上下载“RDA5981 单口下载工具”，地址为：  
<http://bbs.rdamicro.com/forum.php?mod=viewthread&tid=108&extra=page%3D1>
- 将 DEMO 板通过 USB 转串口数据线连接至 PC，并记录下对应的 COM 编号。连接时只需要连接上 DEMO 板上的 TXD、RXD 和 GND 三个引脚，不同 DEMO 板的引脚位置不同，将在后文中说明。
- 将下载下来的“RDA5981 单口下载工具”解压，运行其中的 `flashtest.exe`，如下图所示。在“Port”处选择上一步中对应 COM 编号，在“BinFile”处选择要烧录的 bin 文件。点击下载工具右上角的“Start”按钮，DEMO 板上电后按一下 RST 按键后开始下载，下载成功后会显示“Download finish”。



##### 5、查看 trace

我们推荐使用 SecureCRT、Xshell 等终端查看 trace。目前默认的输出波特率为 921600bps，如果要改成其它的波特率，请修改 SDK 根目录下 mbed\_lib.json 文件中“stdio-baud-rate”项的值，如下图所示。该值是全局有效的，请使用 921600、460800、230400、115200 等常见值。

```
"target_overrides": {
  "UNO_91H": {
    "stdio-convert-newlines": true,
    "stdio-baud-rate": 921600
  }
}
```

## 3 features 介绍

### 1、features 简介

mbed 将部分功能以 feature 的形式提供出来，用户可以自由选择使用所需要的功能，而不必将不需要的功能编译也一起编译。目前主要有以下几个 feature：

FEATURE\_LWIP: WiFi Stack + LWIP，实现 WLAN 和 TCP/IP 联网功能。

FEATURE\_CONSOLE: 控制台功能，可以执行用户通过终端输入的指令，用户也可以以此为基础扩展自己的指令。

FEATURE\_GPADCKEY: GPADC KEY 功能，目前我们的 DEMO 板上提供了 6 个 KEY 供用户使用。

FEATURE\_SDCARD: SD Card + FAT32 fs，只支持英文。

FEATURE\_SDCARD\_GBK: 同上，支持中文和英文。

FEATURE\_SDCARD\_EXFAT: SD Card + exFAT fs，只支持英文。

FEATURE\_SPIRAM: PSRAM 功能，采用 SPI 接口。

### 2、添加 features 至工程

要使用某个 feature，需要将该 feature 的关键字加入到工程对应的 mbed\_app.json 文件中。以 FEATURE\_SDCARD 为例，TESTS/TARGET\_RDA/sdcard\_sdmmc 这个 test case 是一个简易的读写 SD Card 的 case，打开这个 case 下的 mbed\_app.json 文件，可以看到如下内容：

```
{
  "target_overrides": {
    "UNO_91H": {
      "target.features_add": ["SDCARD"]
    }
  }
}
```

如果这个 case 同时要使用 ADC KEY 的功能，那么只需要将"target.features\_add": ["SDCARD"] 修改为"target.features\_add": ["SDCARD", "GPADCKEY"]，增加其它的 feature 方法相同。

### 3、FEATURE\_LWIP 的使用

FEATURE\_LWIP 包括我们的 Wifi Stack 和 LWIP 协议栈，实现 WLAN 和 TCP/IP 联网功能。请

用户在阅读此处的同时参考 wifistack 这个 case。

- 首先要#include "WiFiStackInterface.h"这个头文件。
- 声明类 WiFiStackInterface 的对象, WiFiStackInterface 为提供了一套 C++ API, 以支持 WIFI 设备连接 Internet。

```
WiFiStackInterface wifi;
```

- 连接 AP, 假设 SSID 为"a", PWD 为"qqqqqqqq"。

```
wifi.connect("a", "qqqqqqqq", NSAPI_SECURITY_NONE);
```

- 其他操作, 如获取 IP 地址, MAC 地址, 断开连接等操作, 请参阅 mbed-os APIreference。

#### 4、FEATURE\_CONSOLE 的使用

我们为用户提供了一个基本的控制台功能, 请用户阅读此处的同时参考 console 这个 case。

- 首先要#include "console.h"这个头文件。
- 初始化一个 cmd\_tbl\_t 类型的变量, cmd\_tbl\_t 是一个结构体, 起定义如下:

```
typedef struct cmd_tbl_s {  
    char    *name;        /* Command Name */  
  
    int     maxargs;      /* maximum number of arguments */  
  
    int     (*cmd)(struct cmd_tbl_s *, int, char *[]);  
  
    char    *usage;       /* Usage message(short)*/  
  
} cmd_tbl_t;
```

下面我们定义了一个 cmd\_tbl\_t 类型的变量 echo\_cmd, 其命令的名称为"echo", 即用户在终端输入的命令。该命令的回调函数为 uart\_echo\_func, 入参数量为 3, 命令的说明为"echo len - echo uart data with len\r\n"。

```
cmd_tbl_t echo_cmd = {  
  
    "echo", 3, uart_echo_func,  
  
    "echo len - echo uart data with len\r\n"  
  
};
```

- 初始化 console。
- ```
console_init();
```
- 将命令添加至 console, 返回值为 0 代表添加成功。

```
console_cmd_add(&echo_cmd);
```



这时候新添加的命令"echo"就可以使用了。

## 5、FEATURE\_GPADCKEY 的使用

目前我们的 DEMO 板上提供了 6 个 KEY 供用户使用，分别为 KEY\_A0 至 KEY\_A5，请用户阅读此处介绍的同时参考 gpadc\_key 这个 case。

- 首先要#include "gpadckey.h"这个头文件。
- 声明你要使用的 KEY，例如你要使用 KEY\_A0，自定义的名称为 ps\_key，则需要在程序中声明 GpadcKey ps\_key(KEY\_A0);
- 编写中断服务函数，KEY 中断只支持下降沿触发和上升沿触发，故一个 KEY 只能同时绑定两个中断服务函数。
- 注册中断服务函数，假设 KEY 的上升沿和下降沿中断服务函数分别为 void FallFlip()和 void RiseFlip()，则执行以下两句代码即完成了注册。

```
ps_key.rise(&RiseFlip);
```

```
ps_key.fall(&FallFlip);
```

## 6、FEATURE\_SDCARD、GBK、EXFAT 的使用

FEATURE\_SDCARD\_GBK、FEATURE\_SDCARD\_EXFAT 的使用方法与 FEATURE\_SDCARD 相同，GBK 增加了对中文的支持，EXFAT 则是增加对 exFAT fs 的支持。请用户在阅读此处介绍的同时参考 sdcard\_sdmmc 和 sdcard\_sdmmc\_exfat 这两个 case。

- 首先要#include "SDMMCFFileSystem.h"这个头文件。
  - 声明你要挂载的卷的名称，加入卷名为"sdc":
- ```
SDMMCFFileSystem sdc(PB_9, PB_0, PB_3, PB_7, PC_0, PC_1, "sdc");
```
- 接下来可以使用 fopen、fread、fwrite、fclose 等标准文件操作函数了，如以只读方式打开 sd 卡根目录下的一个文件 a.mp3:

```
FILE *fp = fopen("/sdc/a.mp3", "r");
```

## 7、FEATURE\_SPIRAM 的使用

我们目前支持 SPI 接口的 PSRAM，目前最大 32Mbit。请用户在阅读此处时参考 psram\_spi 这个 case。

- 首先要#include "SpiRAM.h"这个头文件。
- 声明类 SpiRAM 的对象。

```
SpiRAM sRam(PD_2, PD_3, PD_0, PD_1);
```

- 调用 `SpiRAM::setMode(SPIRAM_MODE_T mode)` 设置读写模式，分为 Byte 模式和 Word 模式，默认使用 Byte 模式，使用 Word 模式时必须保证读写时的地址是 4 字节对齐的。

`sRam.setMode(SPIRAM_MODE_BYTE);`//设置为 Byte 模式

`sRam.setMode(SPIRAM_MODE_WORD);`//设置为 Word 模式

- 调用 `SpiRAM::readBuffer(uint32_t address, void *buffer, uint32_t length)` 写数据至 PSRAM，其中 address 为 PSRAM 地址，buffer 为数据指针，length 为数据长度。
- 调用 `SpiRAM::writeBuffer(uint32_t address, void *buffer, uint32_t length)` 读 PSRAM 的数据，其中 address 为 PSRAM 地址，buffer 为数据指针，length 为数据长度。

## 4 test case 介绍

RDA5981 的 test case 全部放在 TESTS/TARGET\_RDA 下，目前所有的 case 如下所示：

analog\_in: ADC 测试 case，采集 PB\_6 引脚的电平并打印出量化之后的数值。

airkiss: 与 smartconfig 功能类似，需要使用到微信 airkiss app。详细使用请见本 case 目录下 readme.txt。

console: 控制台测试 case，使用到了 FEATURE\_CONSOLE。case 中演示了如何增加”test”和”echo”这两个指令到控制台中。

gpadc\_key: GPADC KEY 测试 case，使用到了 FEATURE\_GPADCKEY。case 中演示了如何使用 KEY。

i2c: I2C 测试 case。

i2s\_in: I2S 作为 Slave 时的输入测试 case。

i2s\_out: I2S 作为 Master 时的输出测试 case。

psram\_spi: SPI 接口 PSRAM 测试 case，用到了 FEATURE\_SPIRAM。

pwm: pwm 输出测试 case。

rda5856\_player: RDA5856 音频播放测试 case，RDA5856 为我司的一款 BT(支持 BLE)+CODEC 芯片，使用方法请参考 readme.txt。

rtc: RTC timer 测试 case。

sdcard\_sdmmc: SD Card + FAT32 fs 测试 case，使用到了 FEATURE\_SDCARD。

sdcard\_sdmmc\_exfat: SD Card + exFAT fs 测试 case，使用到了 FEATURE\_SDCARD\_EXFAT。

serial: UART 测试 case，此 case 中使用的是 UART1，也就是输出 trace 和烧录 bin 使用的 UART。如果要使用另外一路 UART，需要将程序中的 UART1 更改为 UART2。

SmartConfig: SmartConfig 测试 case，使用到了 FEATURE\_LWIP 和 FEATURE\_GPADCKEY。用户在自己的工程中集成此功能后，只需要在手机上安装了我司对应的 APP，就可以使用手机给 DEMO 板配网，而不需要 DEMO 板的任何操作。详细使用请见本 case 目录下 readme.txt。

spi: SPI 测试 case。

timer: timer 测试 case。

tls\_client: tls 加密测试 case。

uartwifi: 通过 AT 测试 WIFI 功能的 case。使用方法请见本 case 目录下 readme.txt。

wifistack: lwip 协议栈测试 case，使用到了 FEATURE\_LWIP。

## 5 资料介绍

我们会定期将 RDA5981 的资料更新到我司的 bbs 上，bbs 地址为：<http://bbs.rdamicro.com>。

RDA5981 资料汇总：

<http://bbs.rdamicro.com/forum.php?mod=viewthread&tid=74&extra=page%3D1>

RD5981 SDK released (for NDA customer):

<http://bbs.rdamicro.com/forum.php?mod=viewthread&tid=132&extra=page%3D1>

RD5981 HDK released (for NDA customer):

<http://bbs.rdamicro.com/forum.php?mod=viewthread&tid=145&extra=page%3D1>

RDA5981 单口下载工具：

<http://bbs.rdamicro.com/forum.php?mod=viewthread&tid=108&extra=page%3D1>

RDA5981 SmartConfig apk(android):

<http://bbs.rdamicro.com/forum.php?mod=viewthread&tid=166&extra=page%3D1>

RDA5981&RDA5856 软件版本更新：

<http://bbs.rdamicro.com/forum.php?mod=viewthread&tid=151&extra=page%3D1>

mbed OS API 介绍：

<https://docs.mbed.com/docs/mbed-os-api-reference/en/latest/>

## 6 DEMO 板介绍

RDA5981A/C\_HDK\_V1.1 正面如图 6-1 所示，右下角为 RDA5981 的串口，用于烧录 RDA5981 的 bin 文件和查看 trace。中间为 RDA5856 的串口，用于烧录 RDA5856 的 lod 文件。左下角为 6 个 KEY。左上方为 3.5mm 音频接口和 MIC。

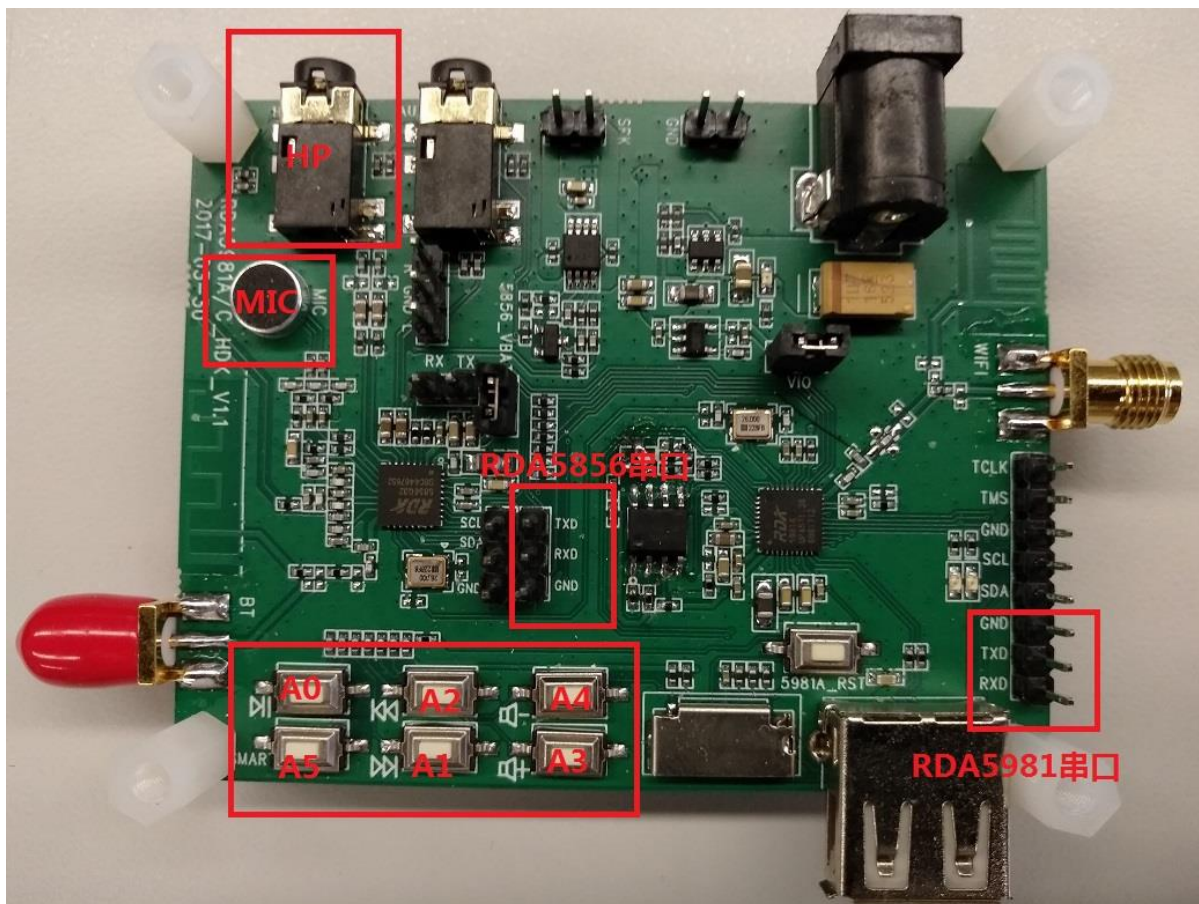


图 6-1 RDA5981A/C\_HDK\_V1.1