

一步一步学习 CC3200 的 WIFI 音频对讲

前言

CC3200 器件是一款集成了高性能 ARM Cortex-M4 MCU 的无线 MCU，包含多种外设，其中包括一个快速并行摄像头接口，I2S，SD/MMC，UART，SPI，I²C 和四通道模数转换器 (ADC)，而 Wi-Fi 网络处理器子系统特有一个 Wi-Fi 片上互联网并且包含一个额外的专用 ARM MCU，此 MCU 可完全免除应用 MCU 的处理负担，因此，外设可以做很多与 WIFI 密切相关的应用。此次以外设 I2S 做基于 WIFI 的音频对讲，教大家一步一步学习 CC3200 的 WIFI 音频对讲

1. 准备材料

- 1.1. 软件：CC3200SDK_1.1.0 版本、CCS6.0.1、CCS UniFlash、wifi starter (app)
- 1.2. 硬件：RF-DK-3200B1 开发板，CC3200AUDBOOST 各两块、3.5 耳机公对公插头一条、一条耳机或者一个音箱。

2. 软件部分

使用 CCS 导入 MQTT 工程，路径为：C:\ti\CC3200SDK_1.1.0\cc3200-sdk\example\wifi_audio_app；如图 1

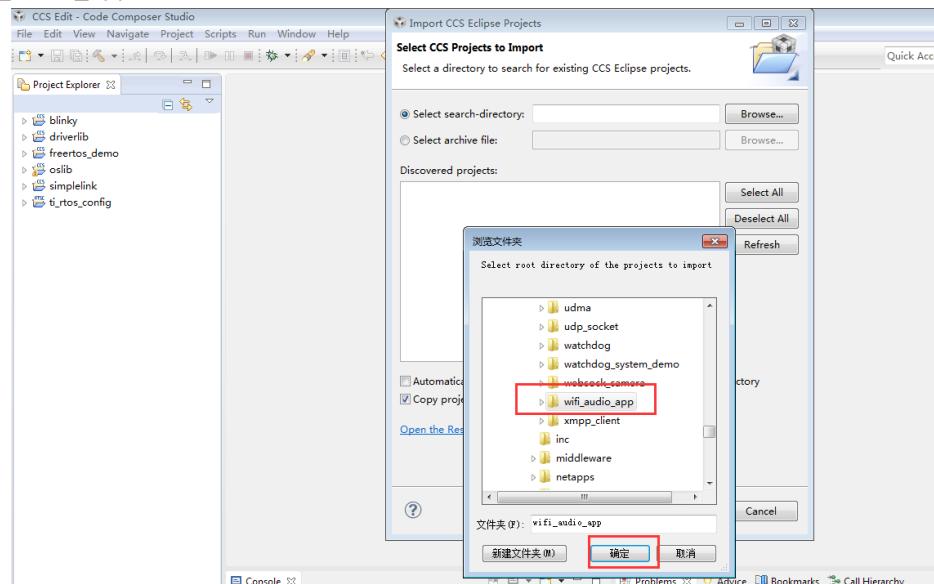


图 1

点击 Finish 完成导入，如图 2

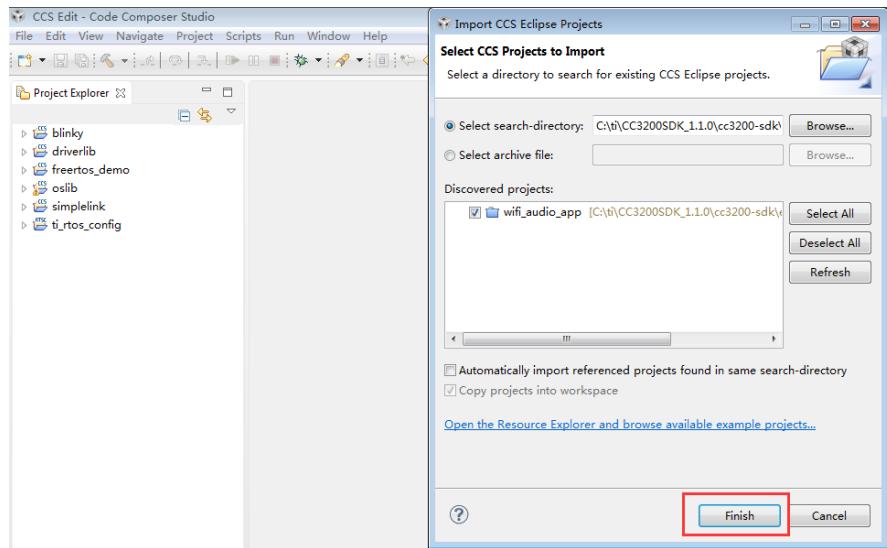


图 2

导入后如图，如图 3

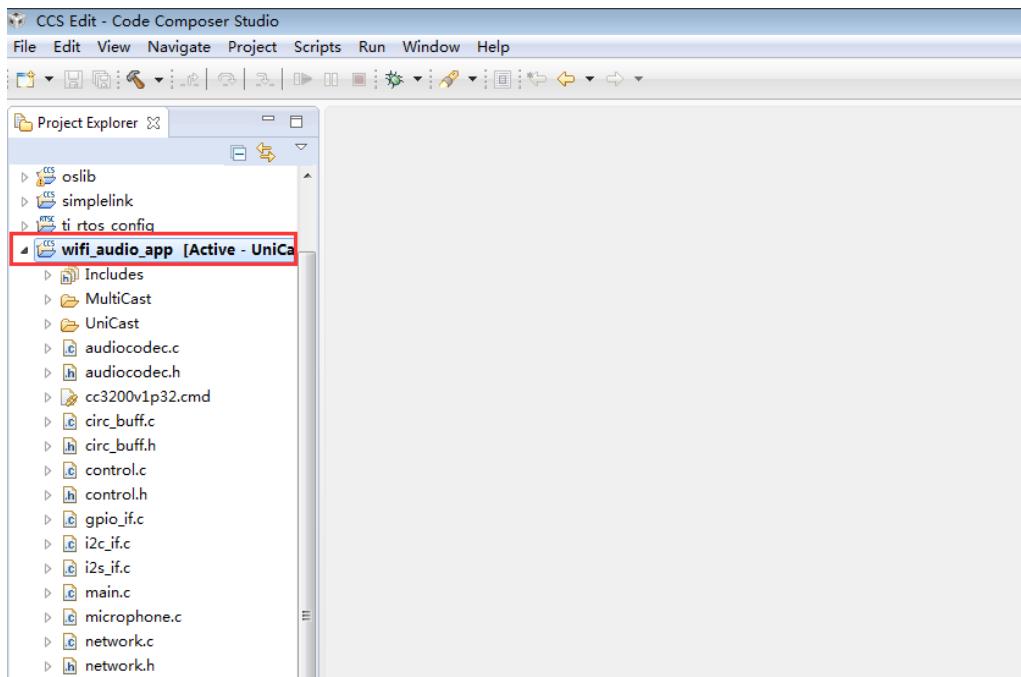
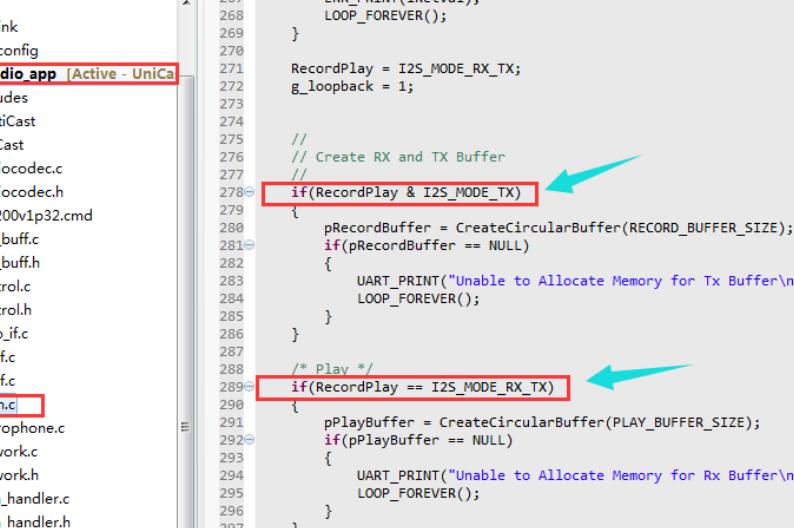


图 3

由于此程序有 bug，需要我们根据 TI CC3200 Issues 来修改

在 `wifi_audio_app` 工程的 `main.c` 文件中，修改位置为 278 行和 289 行；如图 4 中箭头所示，注释 “`if(RecordPlay & I2S_MODE_TX)`”，加上 “`if(RecordPlay == I2S_MODE_RX_TX)`”
 注释 “`if(RecordPlay == I2S_MODE_RX_TX)`”，加上 “`if(RecordPlay & I2S_MODE_TX)`”



Project Explorer

- oslib
- simplelink
- ti_rtos_config
- wifi_audio_app [Active - UniCast]
- Includes
- MultiCast
- UniCast
- audiocodec.c
- audiocodec.h
- cc3200v1p32.cmd
- circ_buf.c
- circ_buf.h
- control.c
- control.h
- gpio_if.c
- i2c_if.c
- i2s_if.c
- main.c
- microphone.c
- network.c
- network.h
- pcm_handler.c
- pcm_handler.h
- pinmux.c
- pinmux.h
- smartconfig.c

main.c

```
266
267
268
269
270
271
272
273
274
275
276
277
278 if(RecordPlay & I2S_MODE_RX_TX)
279 {
280     pRecordBuffer = CreateCircularBuffer(RECORD_BUFFER_SIZE);
281     if(pRecordBuffer == NULL)
282     {
283         UART_PRINT("Unable to Allocate Memory for Tx Buffer\n\r");
284         LOOP_FOREVER();
285     }
286 }
287
288 /* Play */
289 if(RecordPlay == I2S_MODE_RX_TX)
290 {
291     pPlayBuffer = CreateCircularBuffer(PLAY_BUFFER_SIZE);
292     if(pPlayBuffer == NULL)
293     {
294         UART_PRINT("Unable to Allocate Memory for Rx Buffer\n\r");
295         LOOP_FOREVER();
296     }
297 }
298
299
300
301 // Configure Audio Codec
```

图 4

修改后效果如图 5

```
274
275     //
276     // Create RX and TX Buffer
277     //
278     //if(RecordPlay & I2S_MODE_TX)
279     if(RecordPlay == I2S_MODE_RX_TX)
280     {
281         pRecordBuffer = CreateCircularBuffer(RECORD_BUFFER_SIZE);
282         if(pRecordBuffer == NULL)
283         {
284             UART_PRINT("Unable to Allocate Memory for Tx Buffer\n\r");
285             LOOP_FOREVER();
286         }
287     }
288
289     /* Play */
290     //if(RecordPlay == I2S_MODE_RX_TX)
291     if(RecordPlay & I2S_MODE_TX)
292     {
293         pPlayBuffer = CreateCircularBuffer(PLAY_BUFFER_SIZE);
294         if(pPlayBuffer == NULL)
295         {
296             UART_PRINT("Unable to Allocate Memory for Rx Buffer\n\r");
297             LOOP_FOREVER();
298         }
299     }
300 }
```

图 5

在 main 文件中，修改位置为 322 行和 328 行，如图 6 中箭头所示，
取消注释 “MAP_PinTypeGPIO(PIN_01, PIN_MODE_0, false);”
取消注释 “MAP_GPIODirModeSet(GPIOA1_BASE, 0x4, GPIO_DIR_MODE_OUT);”
取消注释 “MAP_PinTypeGPIO(PIN_02, PIN_MODE_0, false);”
取消注释 “MAP_GPIODirModeSet(GPIOA1_BASE, 0x8, GPIO_DIR_MODE_OUT);”

```

305     AudioCodecReset(AUDIO_CODEC_TI_3254, NULL);
306     AudioCodecConfig(AUDIO_CODEC_TI_3254, AUDIO_CODEC_16_BIT, 16000,
307                         AUDIO_CODEC_STEREO, AUDIO_CODEC_SPEAKER_ALL,
308                         AUDIO_CODEC_MIC_ALL);
309
310     AudioCodecSpeakerVolCtrl(AUDIO_CODEC_TI_3254, AUDIO_CODEC_SPEAKER_ALL, 50);
311     AudioCodecMicVolCtrl(AUDIO_CODEC_TI_3254, AUDIO_CODEC_SPEAKER_ALL, 50);
312
313
314     GPIO_IF_LedConfigure(LED2|LED3);
315
316     GPIO_IF_LedOff(MCU_RED_LED_GPIO);
317     GPIO_IF_LedOff(MCU_GREEN_LED_GPIO);
318
319     // Configure PIN_01 for GPIO0Output
320
321     //MAP_PinTypeGPIO(PIN_01, PIN_MODE_0, false);
322     //MAP_GPIODirModeSet(GPIOA1_BASE, 0x4, GPIO_DIR_MODE_OUT);
323
324
325     // Configure PIN_02 for GPIO0Output
326
327     //MAP_PinTypeGPIO(PIN_02, PIN_MODE_0, false);
328     //MAP_GPIODirModeSet(GPIOA1_BASE, 0x8, GPIO_DIR_MODE_OUT);
329
330
331
332     //Turning off Green,Orange LED after i2c writes completed - First Time
333     GPIO_IF_LedOff(MCU_GREEN_LED_GPIO);
334     GPIO_IF_LedOff(MCU_ORANGE_LED_GPIO);
335
336

```

图 6

修改后效果如图 7

```

305     AudioCodecReset(AUDIO_CODEC_TI_3254, NULL);
306     AudioCodecConfig(AUDIO_CODEC_TI_3254, AUDIO_CODEC_16_BIT, 16000,
307                         AUDIO_CODEC_STEREO, AUDIO_CODEC_SPEAKER_ALL,
308                         AUDIO_CODEC_MIC_ALL);
309
310     AudioCodecSpeakerVolCtrl(AUDIO_CODEC_TI_3254, AUDIO_CODEC_SPEAKER_ALL, 50);
311     AudioCodecMicVolCtrl(AUDIO_CODEC_TI_3254, AUDIO_CODEC_SPEAKER_ALL, 50);
312
313
314     GPIO_IF_LedConfigure(LED2|LED3);
315
316     GPIO_IF_LedOff(MCU_RED_LED_GPIO);
317     GPIO_IF_LedOff(MCU_GREEN_LED_GPIO);
318
319     // Configure PIN_01 for GPIO0Output
320
321     //MAP_PinTypeGPIO(PIN_01, PIN_MODE_0, false);
322     //MAP_GPIODirModeSet(GPIOA1_BASE, 0x4, GPIO_DIR_MODE_OUT);
323
324
325     // Configure PIN_02 for GPIO0Output
326
327     //MAP_PinTypeGPIO(PIN_02, PIN_MODE_0, false);
328     //MAP_GPIODirModeSet(GPIOA1_BASE, 0x8, GPIO_DIR_MODE_OUT);
329
330
331
332     //Turning off Green,Orange LED after i2c writes completed - First Time
333     GPIO_IF_LedOff(MCU_GREEN_LED_GPIO);
334

```

图 7

在 main 文件中，修改位置为 346 行和 351 行，如图 8 中箭头所示，

注释 “if (RecordPlay == I2S_MODE_RX_TX)”，加上 “if(RecordPlay & I2S_MODE_TX)”
注释 “if(RecordPlay & I2S_MODE_TX)”，加上 “if(RecordPlay == I2S_MODE_RX_TX)”

```

339  // Initialize the Audio(I2S) Module
340  // 
341  // 
342  // Initialize the DMA Module
343  // 
344  // 
345  UDMAInit();
346  ⊞ if(RecordPlay == I2S_MODE_RX_TX)
347  {
348      UDMAChannelSelect(UDMA_CH5_I2S_TX, NULL);
349      SetupPingPongDMATransferRx(pPlayBuffer);
350  }
351  ⊞ if(RecordPlay & I2S_MODE_TX)
352  {
353      UDMAChannelSelect(UDMA_CH4_I2S_RX, NULL);
354      SetupPingPongDMATransferTx(pRecordBuffer);
355  }
356  // 
357  // Setup the Audio In/Out
358  // 
359  lRetVal = AudioSetupDMAMode(DMAPingPongCompleteAppCB_opt, \
360                               CB_EVENT_CONFIG_SZ, RecordPlay);
361  ⊞ if(lRetVal < 0)
362  {
363      ERR_PRINT(lRetVal);
364      LOOP_FOREVER();
365

```

图 8

修改后效果如图 9

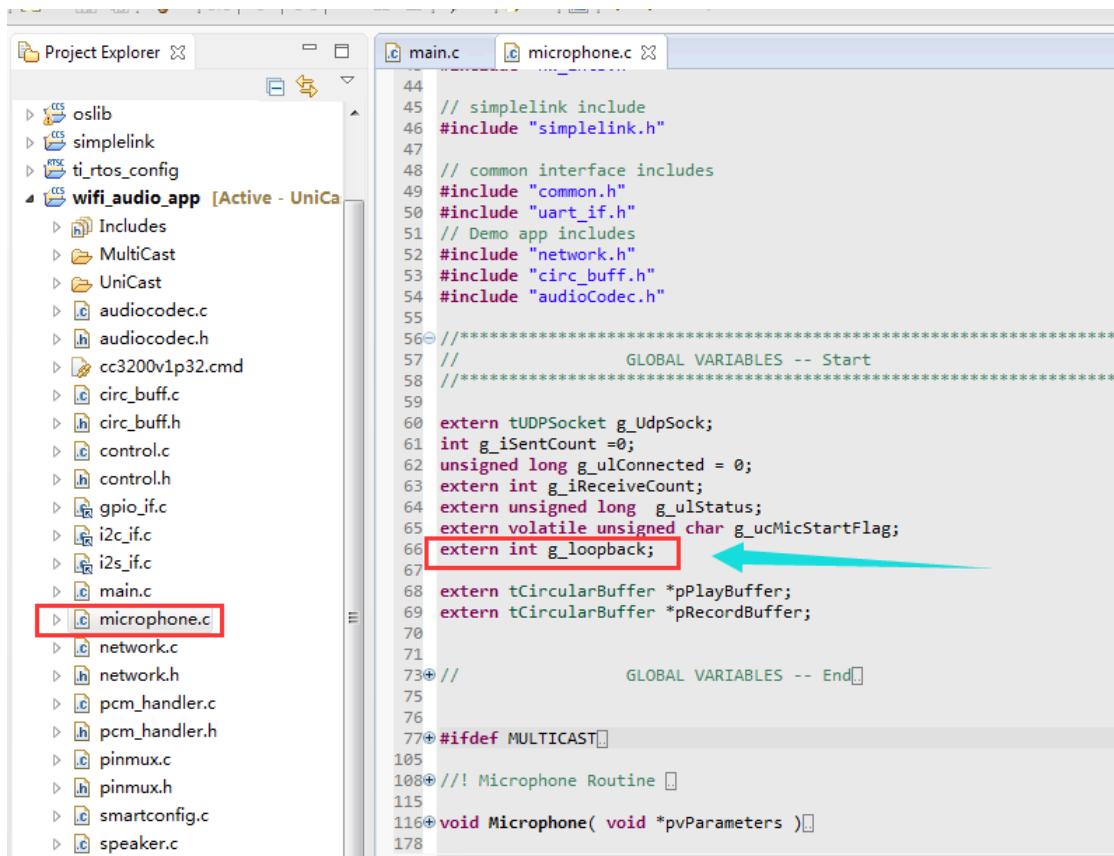
```

339  // Initialize the Audio(I2S) Module
340  // 
341  // 
342  // Initialize the DMA Module
343  // 
344  // 
345  UDMAInit();
346  // if(RecordPlay == I2S_MODE_RX_TX)
347  ⊞ if(RecordPlay & I2S_MODE_TX)
348  {
349      UDMAChannelSelect(UDMA_CH5_I2S_TX, NULL);
350      SetupPingPongDMATransferRx(pPlayBuffer);
351  }
352  // if(RecordPlay & I2S_MODE_TX)
353  ⊞ if(RecordPlay == I2S_MODE_RX_TX)
354  {
355      UDMAChannelSelect(UDMA_CH4_I2S_RX, NULL);
356      SetupPingPongDMATransferTx(pRecordBuffer);
357  }
358  // 
359  // Setup the Audio In/Out
360  // 
361  lRetVal = AudioSetupDMAMode(DMAPingPongCompleteAppCB_opt, \
362                               CB_EVENT_CONFIG_SZ, RecordPlay);
363  ⊞ if(lRetVal < 0)
364  {

```

图 9

在 wifi_audio_app 工程的 microphone.c 文件中，修改位置为 66 行；如图 10 中箭头所示，注释 “extern int g_loopback” ，加上 “extern char g_loopback”



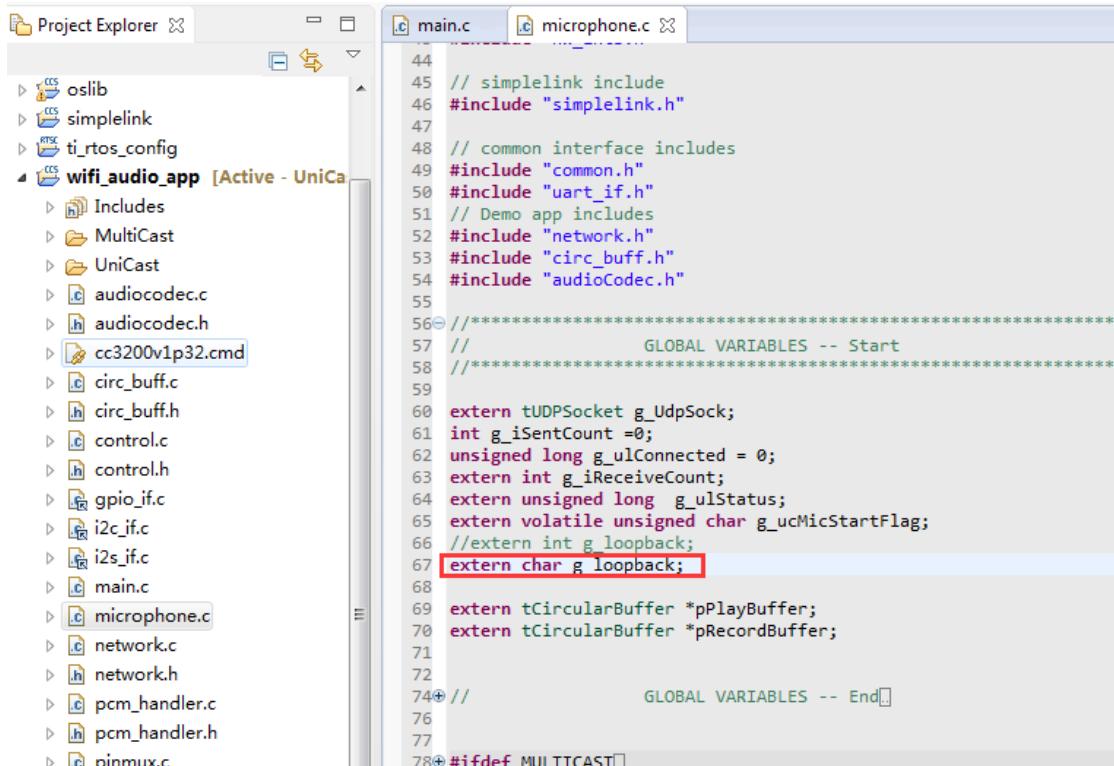
```

44 // simplelink include
45 #include "simplelink.h"
46
47 // common interface includes
48 #include "common.h"
49 #include "uart_if.h"
50 // Demo app includes
51 #include "network.h"
52 #include "circ_buff.h"
53 #include "audioCodec.h"
54
55
56 //*****GLOBAL VARIABLES -- Start*****
57 //*****GLOBAL VARIABLES -- End*****
58
59
60 extern tUDPSocket g_UdpSock;
61 int g_iSentCount =0;
62 unsigned long g_ulConnected = 0;
63 extern int g_iReceiveCount;
64 extern unsigned long g_ulStatus;
65 extern volatile unsigned char g_ucMicStartFlag;
66 extern int g_loopback; ←
67
68 extern tcircularBuffer *pPlayBuffer;
69 extern tcircularBuffer *pRecordBuffer;
70
71
72 //*****GLOBAL VARIABLES -- End*****
73 //*****GLOBAL VARIABLES -- Start*****
74
75
76
77 #ifdef MULTICAST
78 // Microphone Routine
79
80 void Microphone( void *pvParameters )
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178

```

图 10

修改后效果如图 11



```

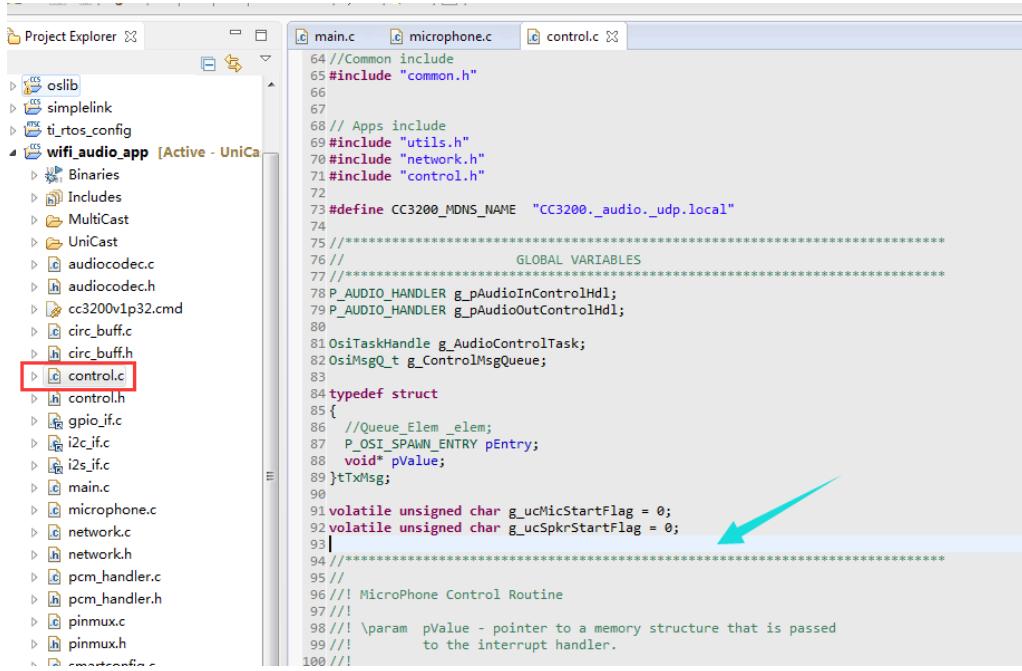
44 // simplelink include
45 #include "simplelink.h"
46
47 // common interface includes
48 #include "common.h"
49 #include "uart_if.h"
50 // Demo app includes
51 #include "network.h"
52 #include "circ_buff.h"
53 #include "audioCodec.h"
54
55
56 //*****GLOBAL VARIABLES -- Start*****
57 //*****GLOBAL VARIABLES -- End*****
58
59
60 extern tUDPSocket g_UdpSock;
61 int g_iSentCount =0;
62 unsigned long g_ulConnected = 0;
63 extern int g_iReceiveCount;
64 extern unsigned long g_ulStatus;
65 extern volatile unsigned char g_ucMicStartFlag;
66 extern char g_loopback;
67
68 extern tcircularBuffer *pPlayBuffer;
69 extern tcircularBuffer *pRecordBuffer;
70
71
72 //*****GLOBAL VARIABLES -- End*****
73 //*****GLOBAL VARIABLES -- Start*****
74
75
76
77 #ifdef MULTICAST
78 // Microphone Routine
79
80 void Microphone( void *pvParameters )
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
110
111
112
113
114
115
116
117
118
119
11
120
121
122
123
124
125
126
127
128
129
12
130
131
132
133
134
135
136
137
138
139
13
140
141
142
143
144
145
146
147
148
149
14
150
151
152
153
154
155
156
157
158
159
15
160
161
162
163
164
165
166
167
168
169
16
170
171
172
173
174
175
176
177
178

```

图 11

在 wifi_audio_app 工程的 control.c 文件中，修改位置为 93 行；如图 12 中箭头所示，加

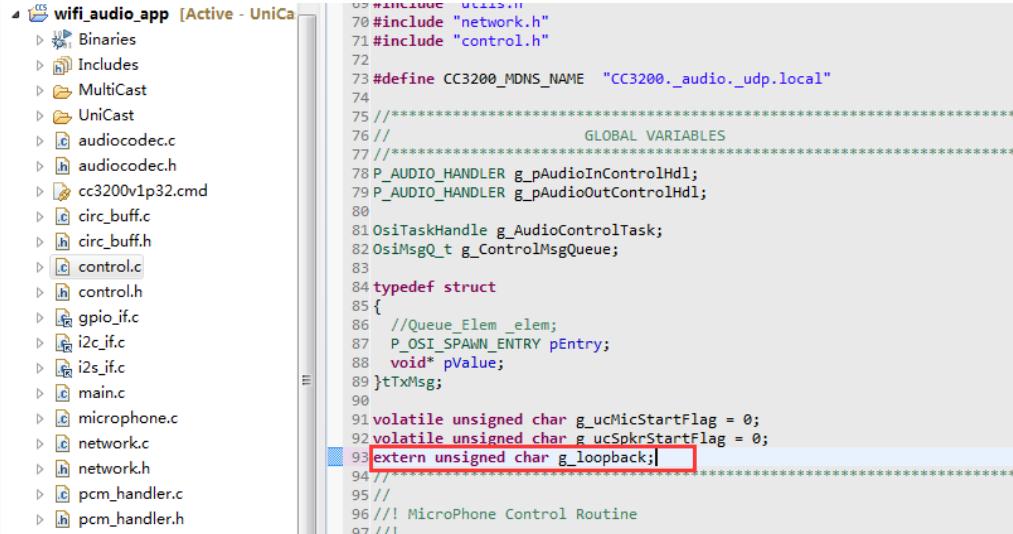
上 “extern unsigned char g_loopback;”



```
64 //Common include
65 #include "common.h"
66
67
68 // Apps include
69 #include "utils.h"
70 #include "network.h"
71 #include "control.h"
72
73 #define CC3200_MDNS_NAME "CC3200._audio._udp.local"
74
75 //*****GLOBAL VARIABLES*****
76 //*****GLOBAL VARIABLES*****
77
78 P_AUDIO_HANDLER g_pAudioInControlHdl;
79 P_AUDIO_HANDLER g_pAudioOutControlHdl;
80
81 OsiTaskHandle g_AudioControlTask;
82 OsiMsgQ_t g_ControlMsgQueue;
83
84 typedef struct
85 {
86     //Queue_Elem _elem;
87     P_OSI_SPAWN_ENTRY pEntry;
88     void* pValue;
89 }tTxMsg;
89
90
91 volatile unsigned char g_ucMicStartFlag = 0;
92 volatile unsigned char g_ucSpkrStartFlag = 0;
93
94 //*****MicroPhone Control Routine
95 //!
96 //! MicroPhone Control Routine
97 //!
98 //! \param pValue - pointer to a memory structure that is passed
99 //!           to the interrupt handler.
100 //!
```

图 12

修改后效果如图 13

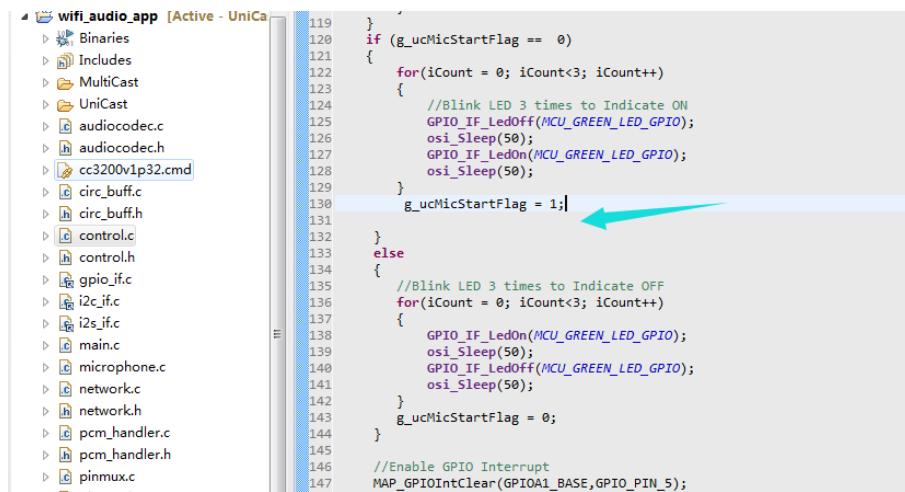


```
64 //INCLUDES
65 #include "network.h"
66 #include "control.h"
67
68
69 #define CC3200_MDNS_NAME "CC3200._audio._udp.local"
70
71
72 //*****GLOBAL VARIABLES*****
73 //*****GLOBAL VARIABLES*****
74
75
76 P_AUDIO_HANDLER g_pAudioInControlHdl;
77 P_AUDIO_HANDLER g_pAudioOutControlHdl;
78
79 OsiTaskHandle g_AudioControlTask;
80 OsiMsgQ_t g_ControlMsgQueue;
81
82
83
84 typedef struct
85 {
86     //Queue_Elem _elem;
87     P_OSI_SPAWN_ENTRY pEntry;
88     void* pValue;
89 }tTxMsg;
90
91 volatile unsigned char g_ucMicStartFlag = 0;
92 volatile unsigned char g_ucSpkrStartFlag = 0;
93
94 extern unsigned char g_loopback;
95
96 //! MicroPhone Control Routine
97 //!
```

图 13

在 wifi_audio_app 工程的 control.c 文件中，修改位置为 131，如图 14 中箭头所示，加上

```
“ #ifdef MULTICAST
    g_loopback = 0;
#endif”
```



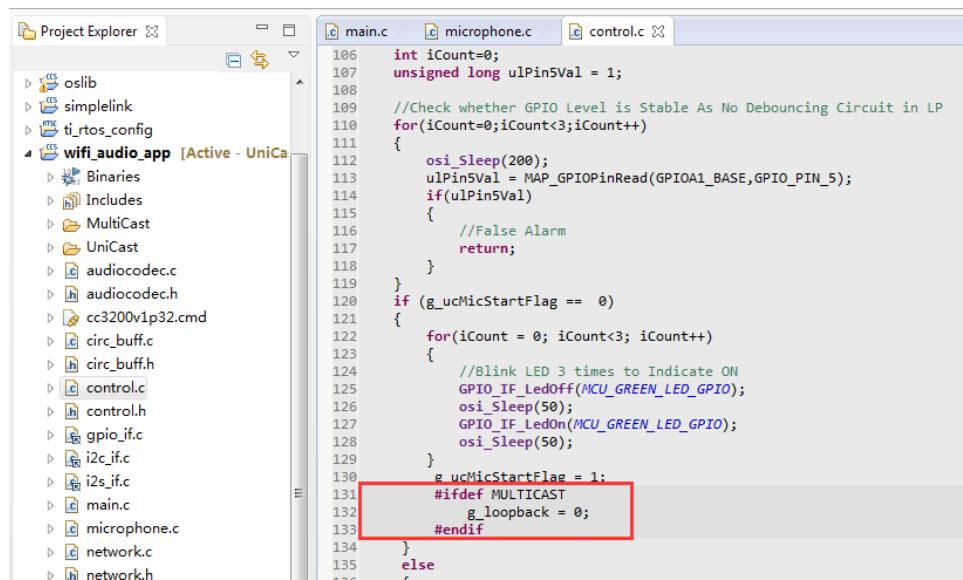
```

119     }
120     if (g_ucMicStartFlag == 0)
121     {
122         for(iCount = 0; iCount<3; iCount++)
123         {
124             //Blink LED 3 times to Indicate ON
125             GPIO_IF_LedOff(MCU_GREEN_LED_GPIO);
126             osi_Sleep(50);
127             GPIO_IF_LedOn(MCU_GREEN_LED_GPIO);
128             osi_Sleep(50);
129         }
130         g_ucMicStartFlag = 1;
131     }
132     else
133     {
134         //Blink LED 3 times to Indicate OFF
135         for(iCount = 0; iCount<3; iCount++)
136         {
137             GPIO_IF_LedOn(MCU_GREEN_LED_GPIO);
138             osi_Sleep(50);
139             GPIO_IF_LedOff(MCU_GREEN_LED_GPIO);
140             osi_Sleep(50);
141         }
142         g_ucMicStartFlag = 0;
143     }
144
145     //Enable GPIO Interrupt
146     MAP_GPIOIntClear(GPIOA1_BASE,GPIO_PIN_5);

```

图 14

修改后效果如图 15



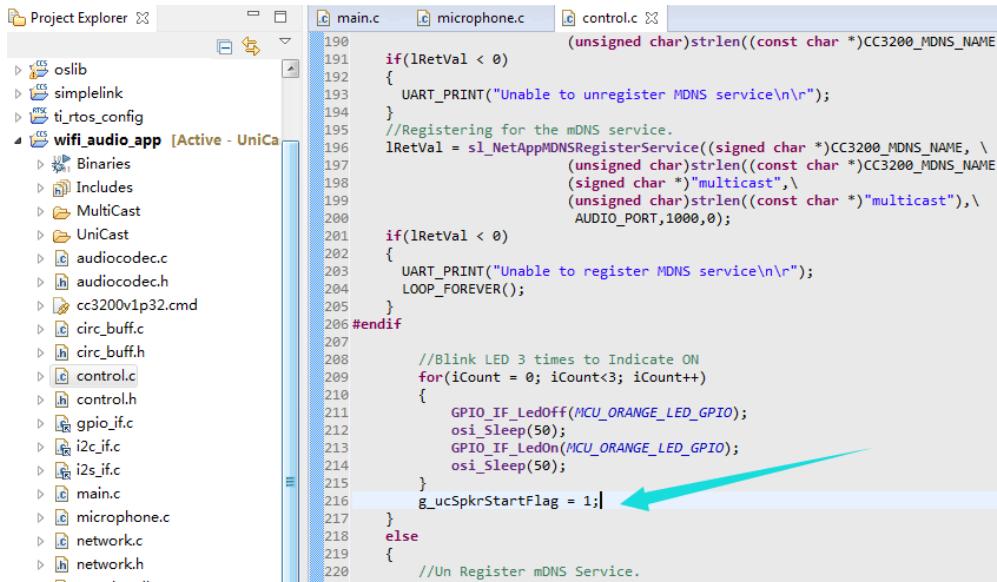
```

106     int iCount=0;
107     unsigned long ulPin5Val = 1;
108
109     //Check whether GPIO Level is Stable As No Debouncing Circuit in LP
110     for(iCount=0;iCount<3;iCount++)
111     {
112         osi_Sleep(200);
113         ulPin5Val = MAP_GPIOPinRead(GPIOA1_BASE,GPIO_PIN_5);
114         if(ulPin5Val)
115         {
116             //False Alarm
117             return;
118         }
119     }
120     if (g_ucMicStartFlag == 0)
121     {
122         for(iCount = 0; iCount<3; iCount++)
123         {
124             //Blink LED 3 times to Indicate ON
125             GPIO_IF_LedOff(MCU_GREEN_LED_GPIO);
126             osi_Sleep(50);
127             GPIO_IF_LedOn(MCU_GREEN_LED_GPIO);
128             osi_Sleep(50);
129         }
130         g_ucMicStartFlag = 1;
131         #ifdef MULTICAST
132         g_loopback = 0;
133         #endif
134     }
135     else

```

图 15

在 wifi_audio_app 工程的 control.c 文件中，修改位置为 216 行；如图 16 中箭头所示，加上
“ g_loopback = 0; ”



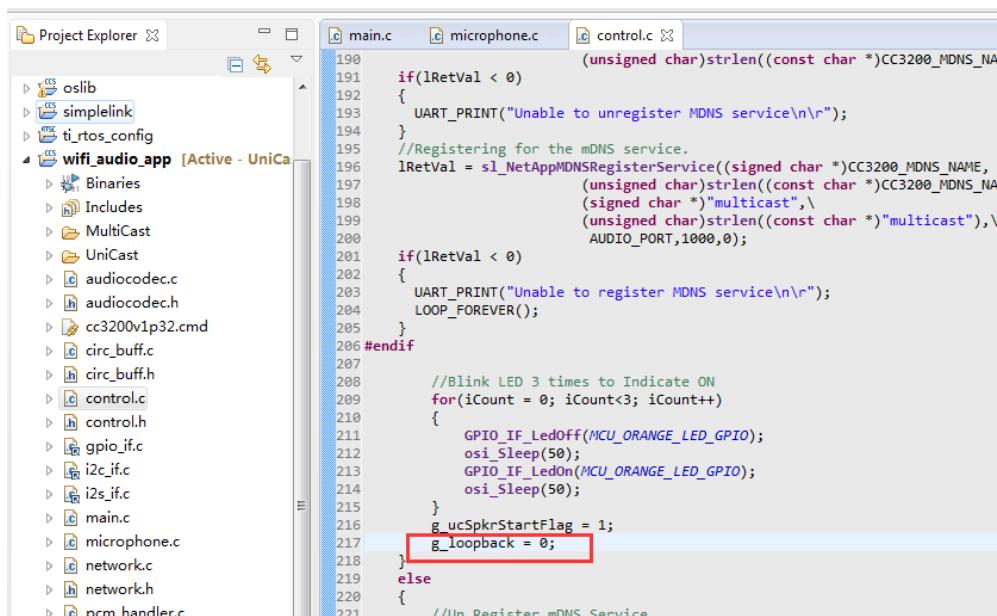
```

190     if(lRetVal < 0)
191     {
192         UART_PRINT("Unable to unregister MDNS service\n\r");
193     }
194     //Registering for the mDNS service.
195     lRetVal = sl_NetAppMDNSRegisterService((signed char *)CC3200_MDNS_NAME,\n
196                                         (unsigned char)strlen((const char *)CC3200_MDNS_NAME,\n
197                                         (unsigned char)strlen((const char *)"multicast",\n
198                                         (unsigned char)strlen((const char *)"multicast"),\n
199                                         AUDIO_PORT,1000,0);
200     if(lRetVal < 0)
201     {
202         UART_PRINT("Unable to register MDNS service\n\r");
203         LOOP_FOREVER();
204     }
205 }
206 #endif
207
208     //Blink LED 3 times to Indicate ON
209     for(iCount = 0; iCount<3; iCount++)
210     {
211         GPIO_IF_LedOff(MCU_ORANGE_LED_GPIO);
212         osi_Sleep(50);
213         GPIO_IF_LedOn(MCU_ORANGE_LED_GPIO);
214         osi_Sleep(50);
215     }
216     g_ucSpkrStartFlag = 1;
217 }
218 else
219 {
220     //Un Register mDNS Service.

```

图 16

修改后效果如图 17



```

190     if(lRetVal < 0)
191     {
192         UART_PRINT("Unable to unregister MDNS service\n\r");
193     }
194     //Registering for the mDNS service.
195     lRetVal = sl_NetAppMDNSRegisterService((signed char *)CC3200_MDNS_NAME,\n
196                                         (unsigned char)strlen((const char *)CC3200_MDNS_NAME,\n
197                                         (unsigned char)strlen((const char *)"multicast",\n
198                                         (unsigned char)strlen((const char *)"multicast"),\n
199                                         AUDIO_PORT,1000,0);
200     if(lRetVal < 0)
201     {
202         UART_PRINT("Unable to register MDNS service\n\r");
203         LOOP_FOREVER();
204     }
205 }
206 #endif
207
208     //Blink LED 3 times to Indicate ON
209     for(iCount = 0; iCount<3; iCount++)
210     {
211         GPIO_IF_LedOff(MCU_ORANGE_LED_GPIO);
212         osi_Sleep(50);
213         GPIO_IF_LedOn(MCU_ORANGE_LED_GPIO);
214         osi_Sleep(50);
215     }
216     g_ucSpkrStartFlag = 1;
217     g_loopback = 0;
218 }
219 else
220 {
221     //Un Register mDNS Service.

```

图 17

把程序保存，编译，并使用 CCS UniFlash 下载到 CC3200 的开发板上。如图 18，下载路径为“C:\Users\Administrator\workspace_v6_0\wifi_audio_app\UniCast\wifi_audio_app.bin”。还要下载 servicepack_1.0.0.10.0.bin。

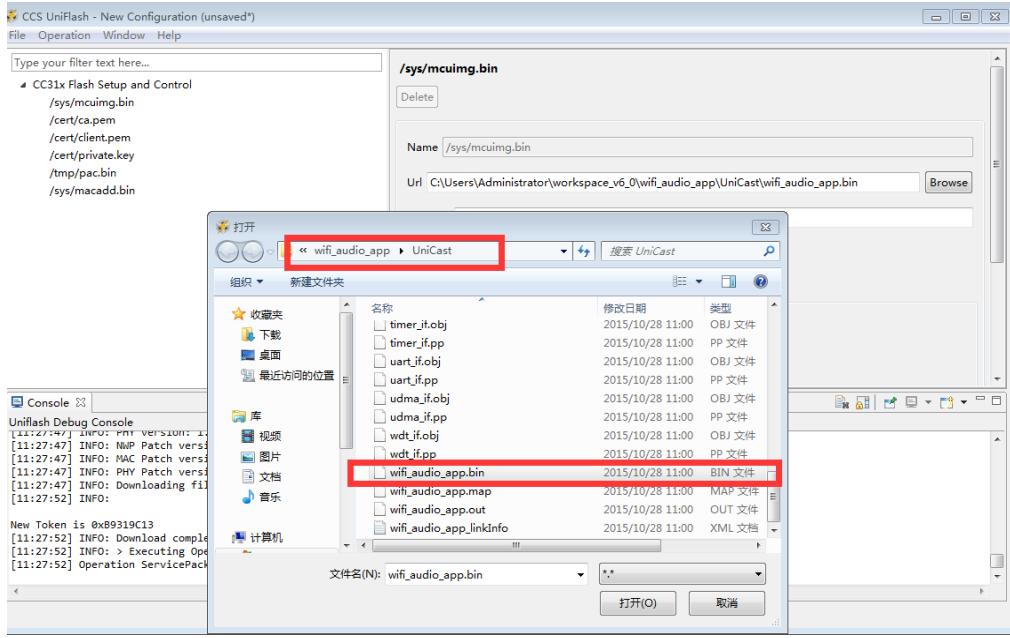


图 18

图 18

3. 硬件部分

在插上 CC3200AUDBOOST 板前, 请确保两块 RF-DK-3200B1 板如图 19 中所示, J1 (I2C 接口) 的两个跳线帽闭合, J2 的跳线帽为打开的



图 19

把 CC3200AUDBOOST 板插上去, 插好了如图 20, 注意方向

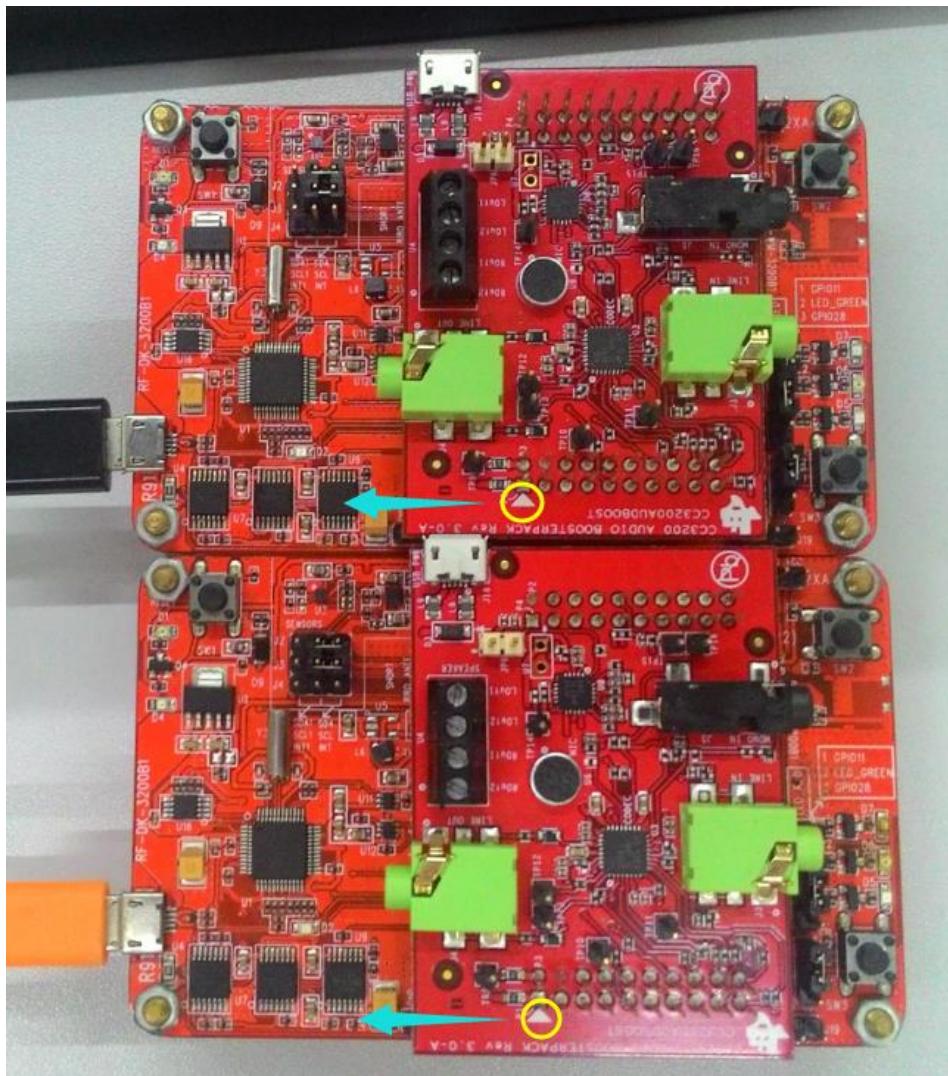


图 20

4. 测试

按下复位键 (SW1)，开发板绿灯会闪烁几秒钟，表示正在连接 AP，如果绿灯长亮，表示连接不上 AP，这时需要使用安卓或者 IOS 设备运行 SmartConfig 程序

在运行 SmartConfig 程序后，如果开发板成功连接上 AP 之后，绿灯会熄灭，红灯长亮。

在实验前，在其中一块开发板插上耳机，当接收板，另外一块开发板插上 3.5 耳机公对公插头，并连接到电脑或者手机（播放音频文件）当音频发射板。如图 21

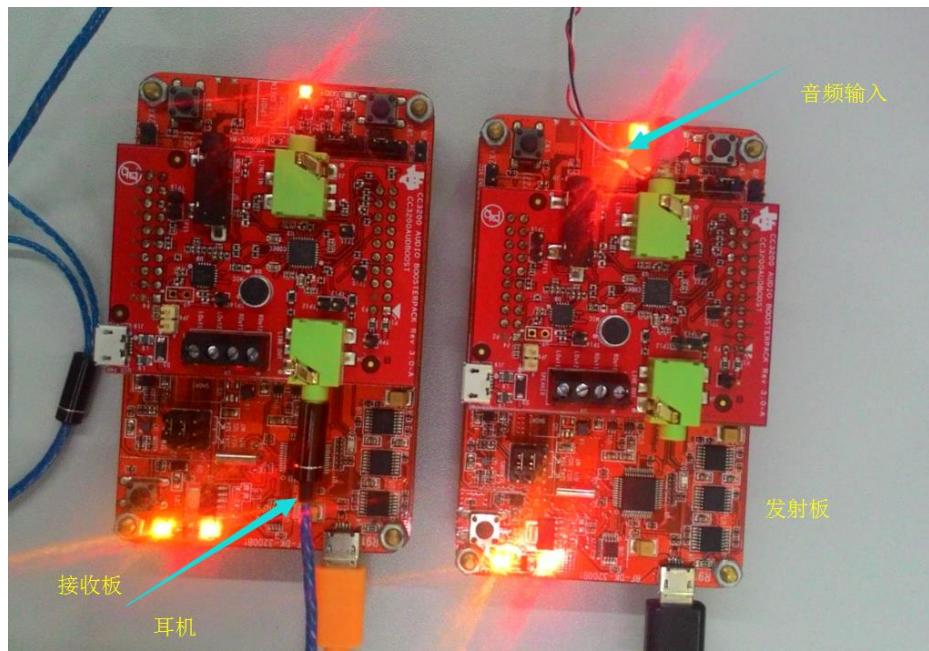


图 21

按下发射板的 SW3 键，绿灯长亮，再按下接收板的 SW2 键，黄灯长亮，红灯也长亮，而发射板的红灯熄灭，表示发射板正在传输音频，接收板正在接收音频，这时耳机就能听到传输过来的音频了。如图 22

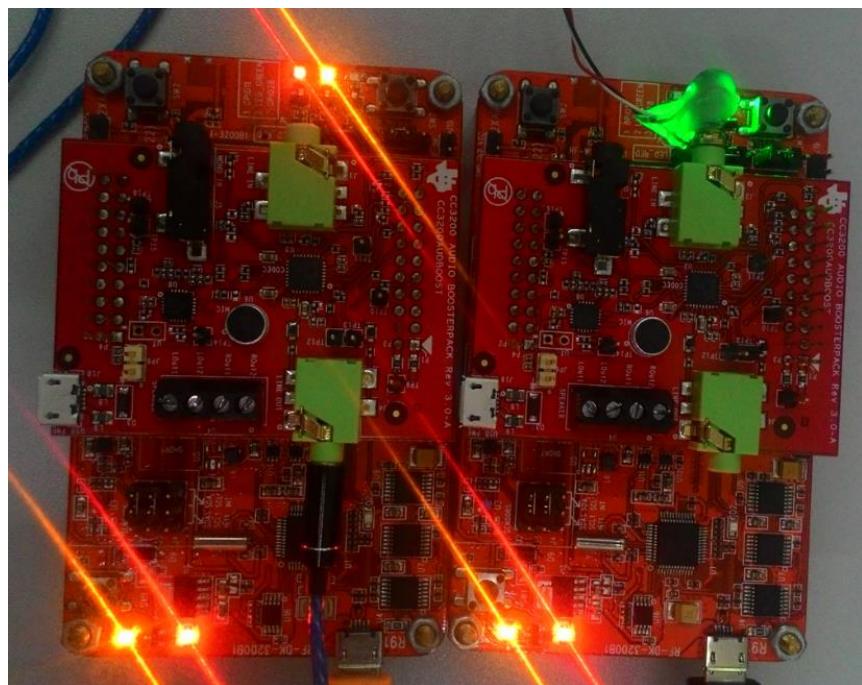


图 22