

SYDTEK OTA 使用说明

SYDTEK 官方提供一个 OTA 测试使用例程，可以直接使用，APK 如下：

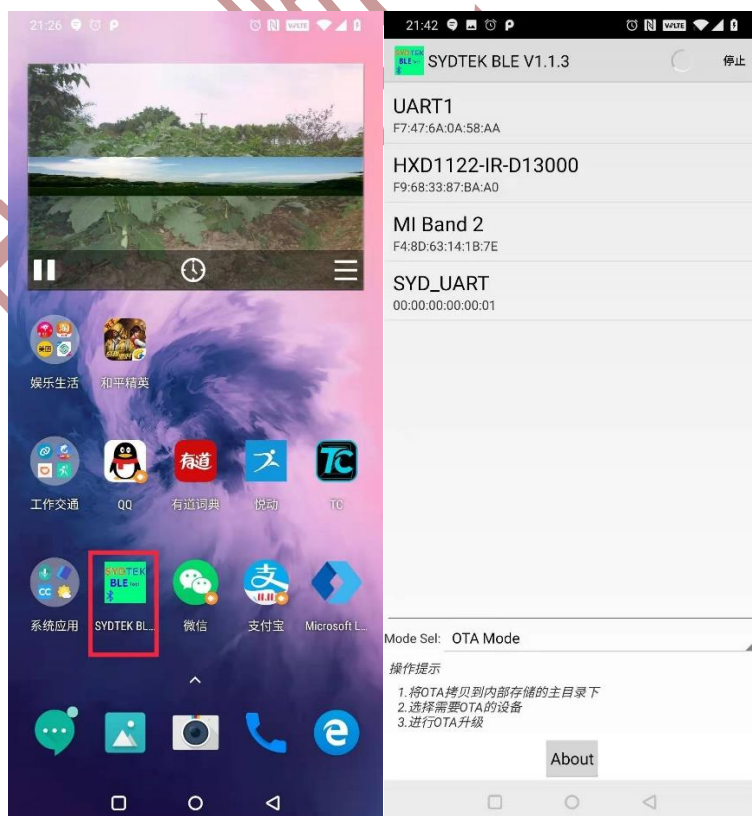


SYDTEK BLE.apk

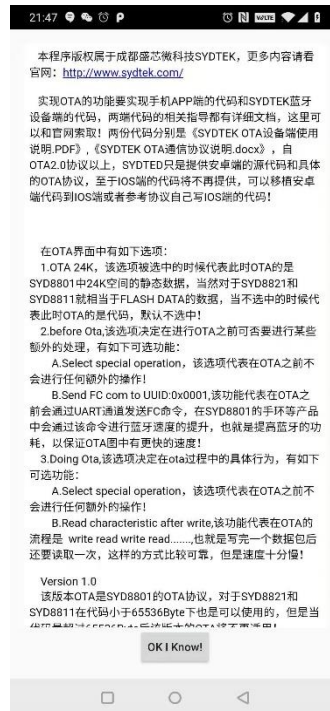
同时也可以直接和原厂索要源代码，然后编译运行，源代码目录如下：

名称	修改日期	类型	大小
.gradle	2019/8/11 23:03	文件夹	
.idea	2019/11/13 21:38	文件夹	
app	2019/11/13 21:38	文件夹	
Documentation	2019/8/11 23:03	文件夹	
gradle	2019/8/11 23:03	文件夹	
projectFilesBackup	2019/8/11 23:03	文件夹	
projectFilesBackup1	2019/8/11 23:03	文件夹	
projectFilesBackup2	2019/11/13 18:03	文件夹	
.gitignore	2016/6/30 10:08	文本文件	1 KB
BLE_OTA.iml	2016/6/30 10:09	IML 文件	1 KB
build.gradle	2019/7/30 16:19	GRADLE 文件	1 KB
gradle.properties	2016/6/30 10:08	PROPERTIES 文件	1 KB
gradlew	2016/6/30 10:08	文件	5 KB
gradlew.bat	2016/6/30 10:08	Windows 批处理...	3 KB
local.properties	2019/11/13 18:03	PROPERTIES 文件	1 KB
settings.gradle	2016/6/30 10:08	GRADLE 文件	1 KB
SYD_OTA_Tool.iml	2019/11/13 18:03	IML 文件	1 KB
SYD_OTA_Tool_v6.1.apk	2019/3/8 17:27	APK 文件	1,847 KB
sydtek_apk_release_key.jks	2019/9/12 11:10	JKS 文件	3 KB

安装 APP 后就可以在手机主页看到 APP(如下左图)，点击打开看到 APP 界面(如下右图)：



这里可以点击下面的“About”按钮查看关于 APP 的相关说明，这里也有 OTA 各个版本的说明：

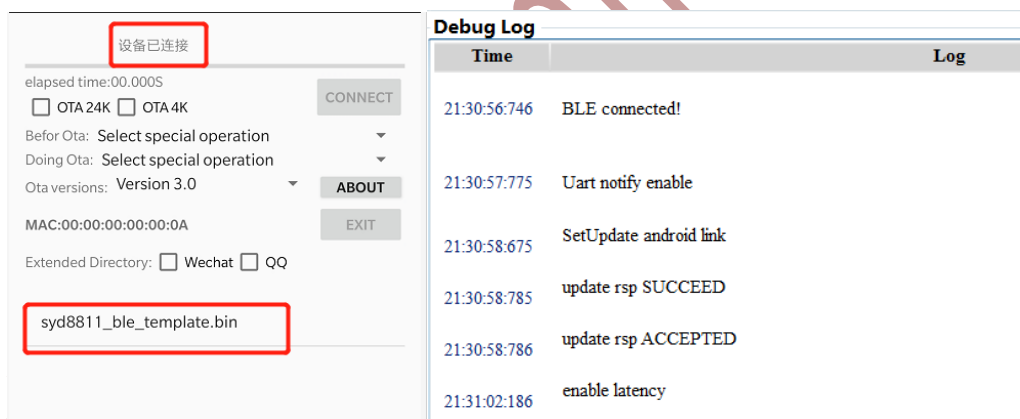


OTA 用户代码:

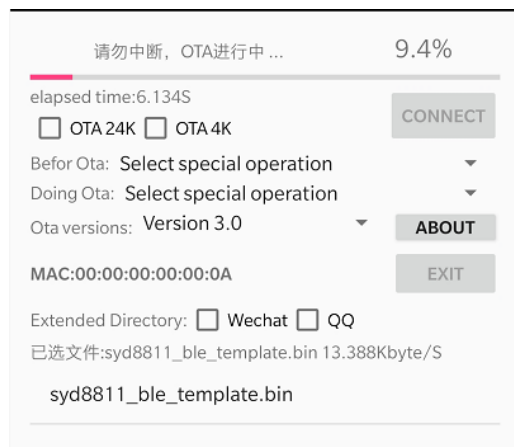
首先，打开官方的 OTA app，如下图所示，选择正确的模式后，点击需要升级的设备。



点击设备后，进入到升级界面，等待 app 提示设备已连接，此时芯片打印的 log 如下。然后，点击需要升级的 bin 文件，注意这个 bin 文件需放在手机的根目录下才能识别。（注意：这里最好等待连接参数更新完成或者等待 10S 后再点击文件开始 OTA）



点击后，耐心等待升级完成。



升级完成后，此时 app 提示复位设备中。OTA 过程的 LOG 如下。

OTA完成, 复位设备中 100%

elapsed time: 16.5855

☐ OTA 24K ☐ OTA 4K

Before Ota: Select special operation

Doing Ota: Select special operation

Ota versions: Version 3.0

MAC:00:00:00:00:00:0A

Extended Directory: ☐ Wechat ☐ QQ

已选文件:syd8811_ble_template.bin 13.388Kbyte/S

syd8811_ble_template.bin

21:32:14:170 SetUpdate ota link

21:32:14:351 update rsp SUCCEEDED

21:32:14:351 update rsp ACCEPTED

21:32:14:708 CMD_FW_ERASE

21:32:16:498 OTA start.

21:32:17:347 CMD_FW_WRITE_START offset:0,size:1400 checksum:e9d2

21:32:21:521 section OK!ota_receive_check:e9d2 ota_section_check:e9d2

21:32:21:577 CMD_FW_WRITE_START offset:1400,size:1400 checksum:630d

21:32:25:875 section OK!ota_receive_check:630d ota_section_check:630d

21:32:25:909 CMD_FW_WRITE_START offset:2800,size:c4c checksum:3eb6

21:32:28:553 section OK!ota_receive_check:3eb6 ota_section_check:3eb6

21:32:28:564 ota sz:344c checksum:8b95

21:32:28:685 CodeUpdatev20=1

这里看到 CodeUpdatev20 返回 1 则代表 OTA 完成！

```

89 static void CmdFwUpgradeV20(struct cmd_fw_upgrade_V20 *p_cmd)
90 {
91     struct ret_fw_erase_cmd Ret;
92     uint8_t temp = 0;
93
94     if(ota_w_idx != 0)
95         CodeWrite(ota_w_cnt*32, ota_w_idx, ota_w_buf);
96
97     #if defined(_DEBUG) || defined(_SYD_RTT_DEBUG)
98     DBGPRINTF("ota sz:%x checksum:%x", p_cmd->sz, p_cmd->checksum);
99     #endif
100     temp = CodeUpdate(NULL, NULL, p_cmd->sz, p_cmd->checksum);
101     #if defined(_DEBUG) || defined(_SYD_RTT_DEBUG)
102     DBGPRINTF("CodeUpdatev20=%d\r\n", temp);
103     #endif
104
105     if(temp==0)
106         Ret.status = ERR_COMMAND_FAILED;
107     else
108         Ret.status = ERR_COMMAND_SUCCESS;
109
110     EvtCommandComplete(CMD_FW_UPGRADE, (uint8_t *)&Ret, sizeof(Ret));
111 }
112
113

```

OTA 完成后底层固件调用复位命令进行复位：

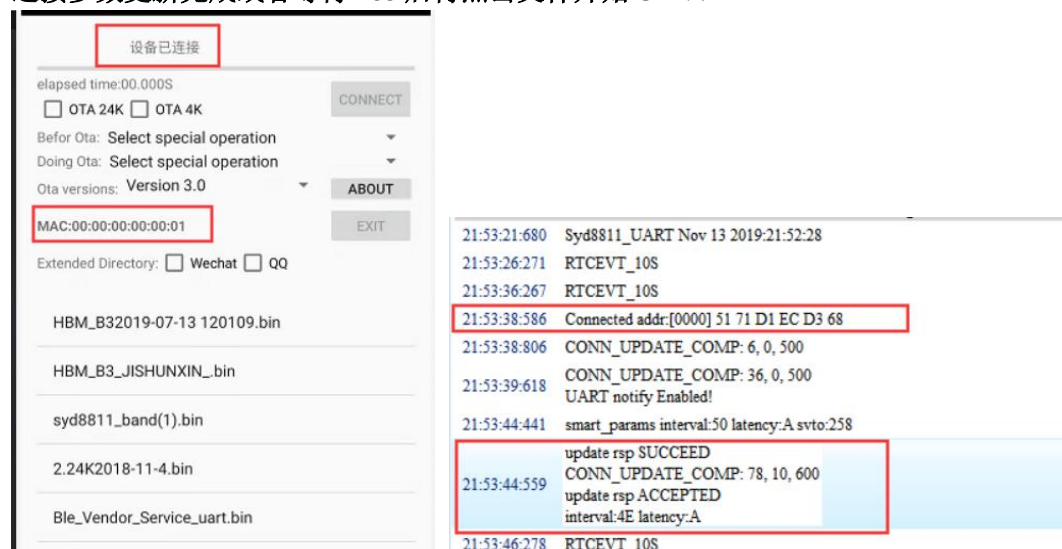
```

777 void ota_manage(void) {
778     #ifdef _OTA
779     if(ota_state) {
780         switch(ota_state) {
781             case 1:
782                 CmdFwErase();
783                 #if defined(_DEBUG) || defined(_SYD_RTT_DEBUG)
784                 ota_state++;
785                 ota_writecnt++;
786                 break;
787             case 2:
788                 ota_writecnt++;
789                 if(ota_writecnt>=200)
790                     break;
791             case 3:
792                 ota_state++;
793                 #if defined(_DEBUG) || defined(_SYD_RTT_DEBUG)
794                 dbg_printf("OTA finish\r\n");
795                 #endif
796                 delay_ms(1000);
797                 #if defined(_DEBUG) || defined(_SYD_RTT_DEBUG)
798                 dbg_printf("Start Reset 1000\r\n");
799                 #endif
800                 SystemReset();
801                 break;
802             case 4:
803                 EraseFlashData(0, 30); //假设FLASHDATA区域大小为120KB,扇区数目为120KB/
804                 #if defined(_DEBUG) || defined(_SYD_RTT_DEBUG)
805                 dbg_printf("OTA start\r\n");
806                 #endif
807                 ota_state++;
808                 ota_writecnt++;
809                 break;
810             default:
811                 break;
812         }
813     }
814     #endif
815 }

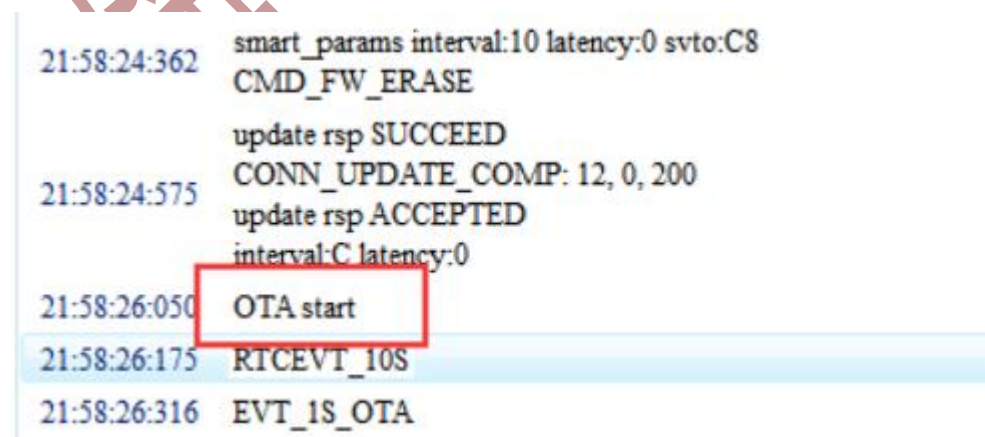
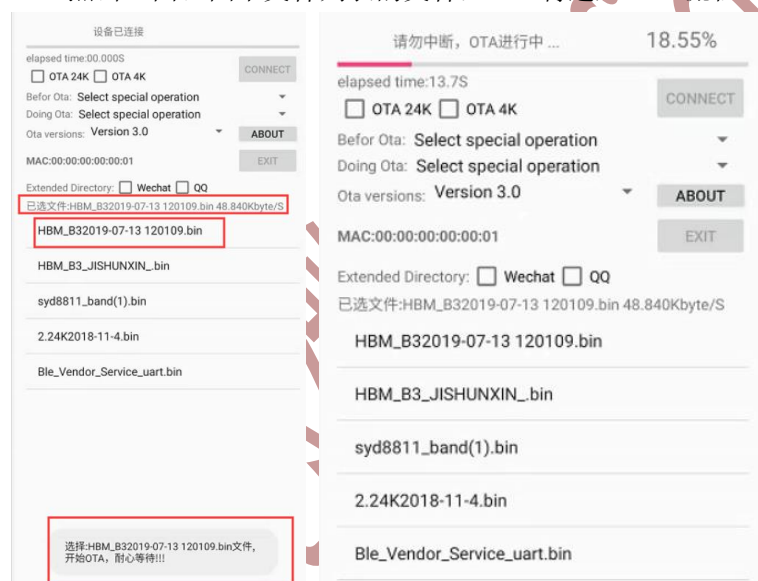
```

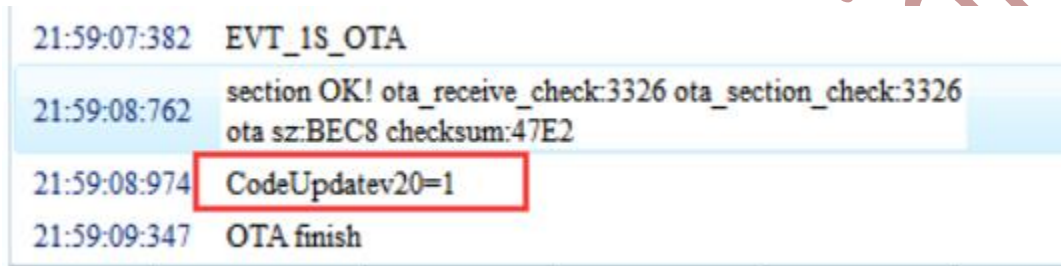
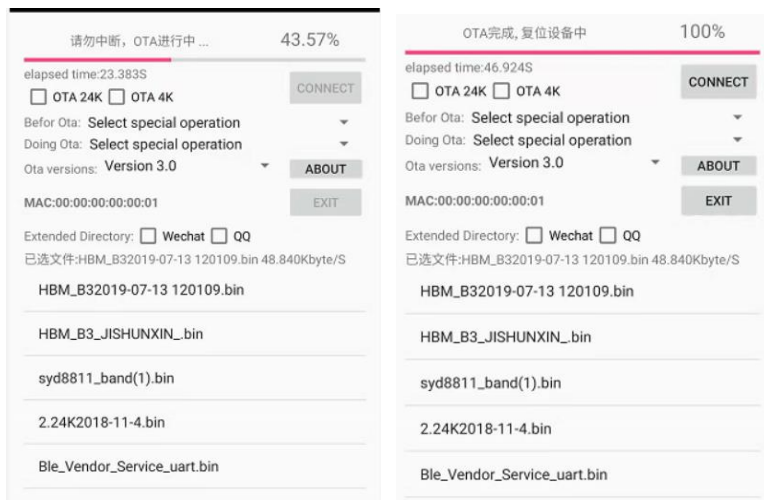
OTA FLASH DATA:

按照上面“OTA 用户代码”的流程，进入 ota 界面正确连接设备（注意：这里最好等待连接参数更新完成或者等待 10S 后再点击文件开始 OTA）：



点击上面左图中文件列表的文件，APP 将进入 OTA 流程：





这里看到 CodeUpdatev20 返回 1 则代表 OTA 完成!

```

113 static void CmdFlashdataUpgradev30(struct cmd_fw_upgrade_V20 *p_cmd)
114 {
115     struct ret_fw_erase_cmd Ret;
116     uint8_t temp = 0;
117
118     #if defined(_DEBUG_) || defined(_SYD_RTT_DEBUG_)
119     DBGPRINTF("ota sz:%x checksum:%x", p_cmd->sz, p_cmd->checksum);
120     #endif
121     temp = FlashDataUpdate(p_cmd->sz, p_cmd->checksum);
122     #if defined(_DEBUG_) || defined(_SYD_RTT_DEBUG_)
123     DBGPRINTF("CodeUpdatev30=%d\r\n", temp);
124     #endif
125
126     if(temp==0)
127         Ret.status = ERR_COMMAND_FAILED;
128     else
129         Ret.status = ERR_COMMAND_SUCCESS;
130
131     EvtCommandComplete(CMD_FW_UPGRADE, (uint8_t *)&Ret, sizeof(Ret));
132 }
133
134

```

OTA 完成后底层固件调用复位命令进行复位:

```

777 void ota_namange(void)
778 {
779     #if defined(_DEBUG_) || defined(_SYD_RTT_DEBUG_)
780     if(ota_state)
781     {
782         switch(ota_state)
783         {
784             case 1:
785                 CmdFwErase();
786                 #if defined(_DEBUG_) || defined(_SYD_RTT_DEBUG_)
787                 ota_writecnt++;
788                 break;
789             case 2:
790                 ota_writecnt++;
791                 if(ota_writecnt>20)
792                 {
793                     break;
794                 }
795             case 3:
796                 ota_state--;
797                 Timer_Brt_Stop(EVT_IS_OTA);
798                 #if defined(_DEBUG_) || defined(_SYD_RTT_DEBUG_)
799                 dbg_printf("OTA finish\r\n");
800                 #endif
801                 delay_ms(1000);
802                 #if defined(_DEBUG_) || defined(_SYD_RTT_DEBUG_)
803                 dbg_printf("Start Reset 1000ms\r\n");
804                 #endif
805                 SystemReset();
806                 ota_state--;
807                 break;
808             case 4:
809                 EraseFlashData(0, 30); //假设FLASHDATA区域大小为120KB扇区数目为120KB/
810                 #if defined(_DEBUG_) || defined(_SYD_RTT_DEBUG_)
811                 dbg_printf("OTA start\r\n");
812                 #endif
813                 ota_state--;
814                 ota_writecnt++;
815                 break;
816             default:
817                 break;
818         }
819     }
820     #endif
821 }
822
823

```

```

ota_receive_check=0;
#if defined(_DEBUG_) || defined(_SYD_RTT_DEBUG_)
dbg_printf("FLASHDATA_WRITE_START offset:%x,size:%x checksum:%x");
#endif
break;
case CMD_FLASHDATA_UPGRADEV30:
CmdFlashdataUpgradev30(&p_cmd->cmdparm.CmdFwUpgradeV20);
ota_state=3;
break;

```

这里的工程目录:

SYD8811_SDK\Source Code\SYD8811_peripheral_misc\SYD8811_BLE_UART_OTA_FLASHDATA

同时请用 SDK 中最新的 lib 进行编译!