



CORE_A_PCBA+WLT7150

使用说明

最新版本|V1.1

维霖通（上海）通信技术有限公司

COPYRIGHT © 2011-2024

关于本手册

本文档对CORE_A_PCBA+WLT7150的使用做了简要说明。

修订历史

版本信息管理

版本号	时间	更新记录	编辑者
V1.0	2024.02.18	初版	刘祺
V1.1	2024.03.05	1. 添加NFC读取功能 2. 优化BLE日志功能 3. 变更部分接口命名	刘祺

目录

关于本手册

修订历史

目录

1.概述

1.1.功能特点

2. 硬件接线

2.1. WLT7150接口

2.2. CORE_A_PCBA与WLT7150接线

3. 功能说明

3.1. 通过BLE连接苹果手表进行数据交互

3.1.1. 初始化

3.1.2. 更新上传到手表的FTMS数据

3.1.3. 更新手表下发来的FTMS数据

3.2. 通过IIC实现WLT7150固件升级

3.3. 通过BLE的FFF0服务做日志打印

3.4. 读取NFC信息

关于我们

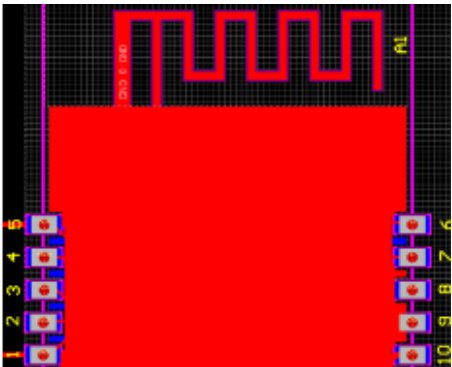
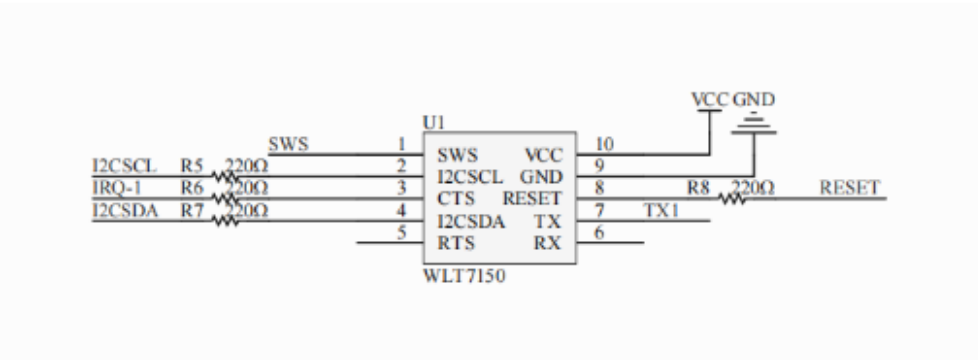
1.概述

1.1.功能特点

- 支持A2DP播歌
- 支持通过BLE连接苹果手表（GymKit）
- 支持通过IIC升级WLT7150
- 支持通过BLE输出调试日志
- 支持读取NFC信息

2. 硬件接线

2.1. WLT7150接口



2.2. CORE_A_PCBA与WLT7150接线

功能	CORE_A_PCBA	WLT7150	备注
IIC_SCL	GPIO41, IN、OUT	I2CSCL, IN、OUT	
IIC_SDA	GPIO42, IN、OUT	I2CSDA, IN、OUT	
IIC_INFORM	GPIO40, IN	CTS, OUT	中断
WLT7150_RESET	GPIO39, OUT	RESET, IN	上电时序同步、异常复位
WLT7150_VCC	-	VCC	3.3V
WLT7150_SWS	-	SWS	Telink烧录器烧录口

3. 功能说明

3.1. 通过BLE连接苹果手表进行数据交互

苹果手表轻点NFC天线后，点击“室内跑步”，稍等后自动转入运动数据显示界面：

1. 时间、距离相关数据，是从板子上传到手表的
2. 心率、卡路里相关数据，是从手表下发到板子的



3.1.1. 初始化

```
1  /**
2   * @brief Gymkit 初始化
3   *
4   * @return 0: 成功
5   *
6   * @note 此时会通过WLT7150_RESET管脚同步时序
7   */
8  int wlt_gymkit_init(void);
```

```
1  #include "LibGymkit.h"
2
3  static void BT_GAP_Event_CB(BT_GAP_Event_Type_t BT_GAP_Event_Type, void *EventData)
4  {
5      ...
6
7      case GAP_BT_START:
8      ...
9
10         wlt_gymkit_init();
11         break;
12
13     ...
14 }
```

3.1.2. 更新上传到手表的FTMS数据

1. 正常连接到手表后，会自动开启定时器：当前定时1s上传FTSM数据到手表上
2. 与手表断开连接后，会自动关闭定时器

```
1 Treadmill_Data_t TreadmillObj;  
2  
3 void ftms_update_test(void) // 通过1s定时器调用  
4 {  
5     ...  
6  
7     // 更新数值开始  
8     TreadmillObj.Elapsed_Time++; // 更新运动时间  
9     TreadmillObj.Total_Distance[0]++; // 更新运动距离  
10    // 更新数值结束  
11  
12    // 通过FTMS上传到手表显示  
13    Ftms_Machine_Data_Update();  
14  
15    ...  
16 }
```


3.1.3. 更新手表下发来的FTMS数据

参考 EUR-CommunicationSpecification 文档中的 Eurotas Data Update (0x0008) 部分:

Eurotas Data Update (0x0008)

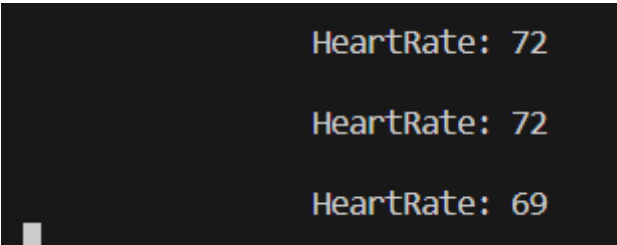
Data (feature values) sent from a connected device. These values are calculated on the connected device and passed to the console for displaying.

BLE Device => Console

Field	Size	Notes
0x55AA	2	Sync word
0x0008	2	Command
0x0000	2	Not used
0x????	2	Payload length (variable)
	?	Raw payload from BLE device (see Eurotas Service Definition Document)
0x??	1	CRC8 of all bytes up to this field

- 1. buffer_size 变量: 对应指令文档中的 Payload length 部分
- 2. buffer 变量: 对应指令中的 Payload 部分

```
1 static int ble_gatt_write_callback(...)
2 {
3   ...
4
5   /* Gymkit Write */
6   case ATT_CHARACTERISTIC_2ADB_01_VALUE_HANDLE:
7
8     log_info("Gymkit Write: %u\r\n", buffer_size);
9
10    /* Eurotas Data Update (0x0008) */
11    /* buffer = Raw payload from BLE device (see Eurotas Service Definition Document) */
12    printf("\r\n\t\tHeartRate: %d\r\n", buffer[4]);
13
14    break;
15
16    ...
17 }
```



3.2. 通过IIC实现WLT7150固件升级

例程中，把module.txt固件文件include成固件数组，然后调用 wlt_gymkit_set_nfc_fw_addr() 函数修改来源，其他方式同理。

```
1  /**
2   * @brief 手动触发OTA
3   *
4   * @return 0: 成功
5   *
6   * @note 从 Gymkit_Init() 开始的上电流程会调用
7   */
8  int wlt_gymkit_update_nfcfw(void);
9
10 /**
11  * @brief 设置固件存储区中的固件版本
12  *
13  * @param Ver
14  *
15  * @return nfc固件版本
16  *
17  * @note 应该在 Gymkit_Init() 之前调用
18  */
19  uint32_t wlt_gymkit_set_nfc_fw_ver(uint32_t version);
20
21 /**
22  * @brief 设置固件存储区
23  *
24  * @param addr 固件存储区地址。注意，内部直接使用，不复制
25  * @param len 固件存储区大小
26  *
27  * @return 0: 成功
28  * @return 1: addr为NULL 或 len为0
29  *
30  * @note 应该在 Gymkit_Init() 之前调用
31  */
32  int wlt_gymkit_set_nfc_fw_addr(uint32_t *addr, uint32_t len);
```

```
1 #include "LibGymkit.h"
2
3 static const unsigned char Update_Img[] =
4 {
5 #include "module.txt"
6 };
7 static void BT_GAP_Event_CB(BT_GAP_Event_Type_t BT_GAP_Event_Type, void *EventData)
8 {
9 ...
10
11 case GAP_BT_START:
12 ...
13
14     wlt_gymkit_set_nfc_fw_ver(20);
15     wlt_gymkit_set_nfc_fw_addr(&Update_Img, sizeof(Update_Img));
16     wlt_gymkit_init();
17     break;
18
19 ...
20 }
```

以下图片对应WLT7150的日志，方便调试对照：

```
[20:37:46.936]收←◆]tart WLT7150
CompileTime:17:44:29 Feb 22 2024
SV:V1.0.1
[20:37:47.464]收←◆Imge Size:26260
Imgee Cro:77
[20:37:47.640]收←◆NRF Task Running
[20:37:48.072]收←◆
WAITING FOR NFC CARD DISCOVERY:1
[20:37:52.503]收←◆Cro Succeed
[20:37:52.551]收←◆Cro Succeed
[20:37:52.601]收←◆Cro Succeed
Size:26260 Pos:128
[20:38:15.206]收←◆Cro Succeed
[20:38:15.254]收←◆Cro Succeed
Size:26260 Pos:25984
[20:38:15.302]收←◆Cro Succeed
[20:38:15.351]收←◆Cro Succeed
[20:38:15.398]收←◆Cro Succeed
Size:26260 Pos:26112
[20:38:15.446]收←◆Cro Succeed
[20:38:15.494]收←◆Cro Succeed
Size:26260 Pos:26240
Cro Succeed
[20:38:16.054]收←◆Size:26260 Pos:26260
[20:38:16.294]收←◆ota cro:77, flash cro:77
ota img cro pass
[20:38:16.615]收←◆]tart WLT7150
CompileTime:17:23:09 Dec 13 2022
SV:V1.0.20
[20:38:17.318]收←◆NRF Task Running
[20:38:17.750]收←◆
WAITING FOR NFC CARD DISCOVERY:1
```

升级开始

固件包传输中

固件包校验完成

升级完成

3.3. 通过BLE的FFF0服务做日志打印

默认关闭此功能，上电后会有 "CORE_A_PCBA_BLE" 广播，连接后开启 0xFFFF0 Service 中的 0xFFFF1 Characteristic 的 Notification 可开启此功能。

```
1  /**
2   * @brief 初始化 BLE 打印日志功能
3   *
4   * @return 0: 成功
5   * @return !=0: 失败
6   */
7  int wlt_ble_log_init(void);
8
9  /**
10   * @brief 通过 BLE 打印日志
11   *
12   * @param string 日志字符串
13   * @param length 日志字符串长度
14   *
15   * @return >0: 打印日志的长度
16   * @return -1: 未使能
17   * @return -2: 无BLE连接
18   *
19   * @note Notify 到 0xFFFF0 Service 中的 0xFFFF1 Characteristic
20   */
21  int wlt_ble_log_write(uint8_t *string, uint16_t length);
22
23  /**
24   * @brief 获取 BLE 打印日志功能开关
25   *
26   * @return 0: 关闭
27   * @return 1: 开启
28   */
29  uint8_t wlt_ble_log_get_enable(void);
30  /**
31   * @brief 设置 BLE 打印日志功能开关
32   *
33   * @param enable 使能状态
34   *
35   * @return 0: 关闭
36   * @return 1: 开启
37   */
38  uint8_t wlt_ble_log_set_enable(uint8_t enable);
```

```

1 #include "LibLE.h"
2
3 void MainThread(void)
4 {
5 ...
6
7     wlt_ble_log_init();
8 }
9
10
11 static int ble_gatt_write_callback(...)
12 {
13 ...
14
15     case ATT_CHARACTERISTIC_FFF1_01_CLIENT_CONFIGURATION_HANDLE:
16 ...
17
18     wlt_ble_log_set_enable(buffer[0]);
19
20 ...
21     break;
22
23 ...
24 }

```

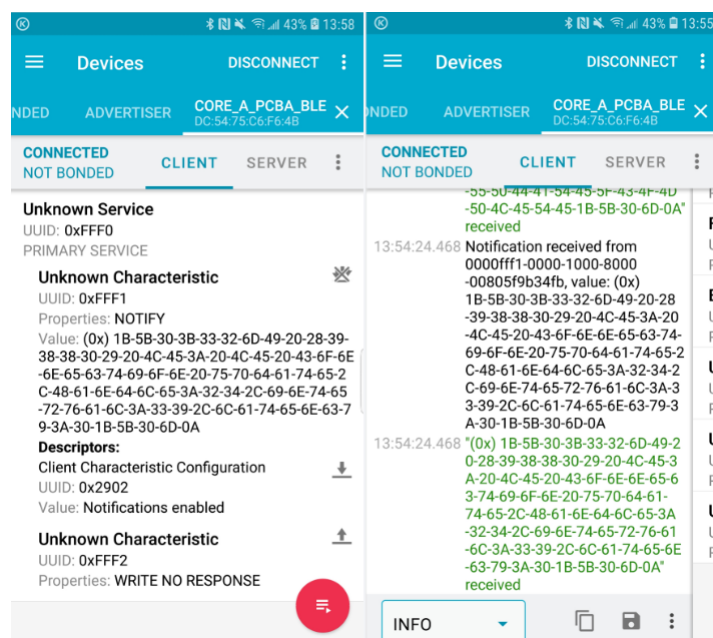
使用如下接口，会同时往串口、BLE发送（功能开启状态）：

```

1 ESP_LOGI("LE", "LE Connection update,Handle:%d,interval:%d,latency:%d", wlt_le_connection_updata_complete_data->connection_handle, wlt_le_connection_updata_complete_data->conn_interval, wlt_le_connection_updata_complete_data->conn_latency);

```

使用nrfconnect连接后即可接收到：



3.4. 读取NFC信息

参考 EUR-CommunicationSpecification 文档中的 NFC Card Info (0x02FD) 部分:

NFC Card Info(0x02FD)

Module => Console

Field	Size	Notes
0x55AA	2	Sync word
0x02FD	2	Command
0x0000	2	Result Code (Success)
0x????	2	Payload Length
0x??	1	Mode Tech : 0x1:TECH_PASSIVE_15693(ISO15693) 0x2:TECH_PASSIVE_NFCA(ISO14443A) 0x3:TECH_PASSIVE_NFCB(ISO14443B) 0x5:TECH_PASSIVE_NFCF(SONY FeliCa)
0x??	1	Protocol: 0x0:PROT_UNDETERMINED 0x1:PROT_T1T 0x2:PROT_T2T 0x3:PROT_T3T 0x4:PROT_ISODEP 0x5:PROT_NFCDEP 0x6:PROT_ISO15693 0x80:PROT_MIFARE
0x??	1	UID Length
	UID	??..??
	2	NFC Info Length: 0x0000 if no NFC payload data
Ndef Payload Recd(s)	??...??(all raw data)	@ NFC P2P NDEF Basics <ul style="list-style-type: none"> NFC Data Exchange Format Each message consists of one or more records, which are made up of a header and the record data. <ul style="list-style-type: none"> 12 (1) Check Flag: "1" for Chartered payloads 16 (1) Short Record: "1" for the size of the payload length is one byte, "0" for otherwise it is a four byte 8 (1) 0 Length field: "1" for this field is present; • M1 (1) Message Length: "1" for first record • M0 (1) Message End: "1" for the last record
0x??	1	CRC8 of all bytes up to this field

1. event 入参：在此样例中为 CMD_NFC_CARD_INFO
2. data 入参：对应指令文档中的 Payload 部分
3. len 入参：对应指令中的 Payload Length 部分

```
1  #define CMD_NFC_CARD_INFO          0x02FD
2
3  /**
4   * @brief Gymkit 命令事件回调函数
5   *
6   * @param event 事件类型
7   * @param data 事件数据
8   * @param len 事件数据长度
9   */
10 typedef void (*wlt_gymkit_event_callback_t)(uint32_t event, void *data, uint32_t len);
11
12 /**
13  * @brief 设置 Gymkit 事件触发时使用的回调函数
14  *
15  * @param callback
16  *
17  * @return 0: 成功
18  * @return 1: 回调地址为 NULL
19  */
20 int wlt_gymkit_set_event_callback(wlt_gymkit_event_callback_t callback);
```



```

1 #include "LibGymkit.h"
2
3 void user_gymkit_event_handler(uint32_t event, void *data, uint32_t len)
4 {
5     switch (event)
6     {
7     case CMD_NFC_CARD_INFO:
8         printf("CMD_NFC_CARD_INFO:\n");
9         for(uint8_t i=0;i<len;i++)
10         {
11             printf("%02X,",((char *)data)[i]);
12         }
13         printf("\n");
14         break;
15     default:
16         break;
17     }
18 }
19
20 static void BT_GAP_Event_CB(BT_GAP_Event_Type_t BT_GAP_Event_Type, void *EventData)
21 {
22     ...
23
24     case GAP_BT_START:
25     ...
26
27         wlt_gymkit_set_event_callback(user_gymkit_event_handler);
28         wlt_gymkit_init();
29         break;
30
31     ...
32 }

```

CMD_NFC_CARD_INFO:

02,02,07,04,A5,86,CD,70,00,00,10,00,D1,01,0C,54,02,65,6E,39,38,37,36,35,34,33,32,31,

关于我们

维霖通（上海）通信技术有限公司成立于2011年，是一家快速成长中的物联网无线通信技术公司，位于上海浦东张江高新技术开发区核心地带。公司专注于提供全球领先的物联网无线连接方案（WiFi/蓝牙/BLE/Lora/NB-IOT等），包括自主研发和自主品牌的通信芯片、通信模块、通信板卡、通信协议软件、手机APP、云计算等部分。主要服务于工业物联网、汽车、医疗与健身、金融支付与安全、高端消费电子、专业乐器、办公设备等行业的大中型客户，目前已经包括中国、美国、欧洲、韩国等地区的40多家深度定制的全球行业龙头客户，以及200多家紧密合作的大中型客户。

公司秉承以人为本、诚信、责任和创新的理念，致力于成为以人才和技术为核心竞争力、可持续发展的高科技公司。公司的核心团队具备10余年500强美资高科技企业管理和技术经验，强调与客户的可持续共赢合作，将公司的无线连接和Cloud技术与客户行业应用深度定制结合起来，为物联网时代下大中型客户持续的产品创新和服务创新，提供可靠的无线连接技术支撑。

Web: <http://www.wi-linktech.com>

Tel: (+86)21-20255077

E-mail: contact@wi-linktech.com

Add: 上海市浦东新区盛荣路88弄3号楼602室