

# 代码使用说明

## 代码使用说明

代码介绍

0.技术框图

1.\_bluez\_part

2.\_edp\_part

2.1.主线程

2.1.副线程

## 代码介绍

名称	修改日期	类型	大小
 _bluez_part	2023/4/21 23:25	文件夹	
 _edp_part	2023/4/26 10:01	文件夹	
 _fork_edp_and_bluez	2023/4/21 23:25	文件夹	
 _uniapp_part	2023/4/26 22:28	文件夹	
 _zigbee_part	2023/4/26 22:27	文件夹	
 readme.md	2023/4/21 23:37	Markdown File	1 KB
 技术框图.png	2023/4/21 22:41	PNG 图片文件	596 KB

一共有5部分代码，其中

**\_bluez\_part**为bluz代码，可执行文件名为"**bt\_app**"

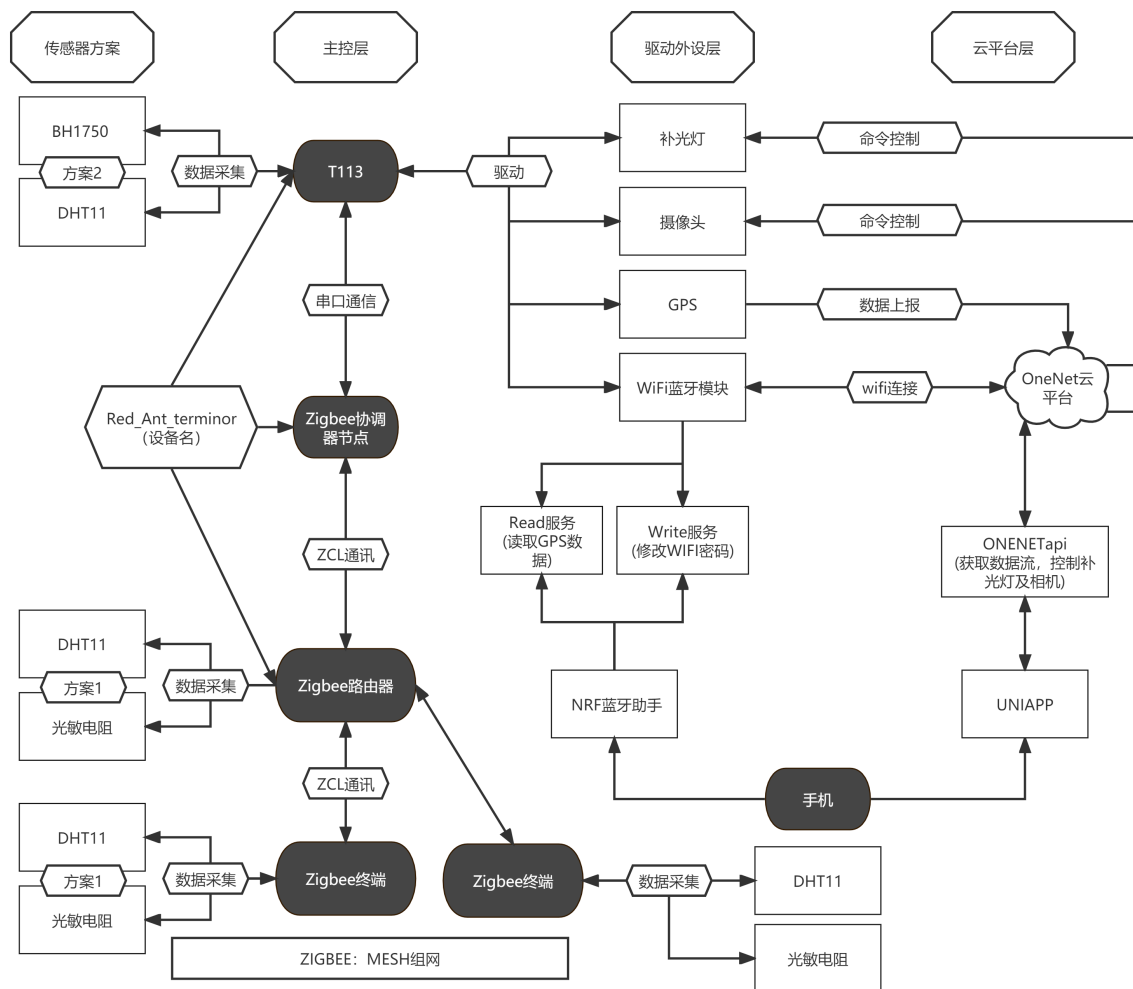
**\_edp\_part**为T113连接onenet的应用部分代码，可执行文件名为"**edp**"

**\_fork\_edp\_and\_bluez**为进程创建代码，可执行文件名为"**main**"

**\_uniapp\_part**为uniapp开发的安卓APP，目前功能有登录界面(账号密码都为1)，下拉onenet的api控制设备，用户退出界面等。

**\_zigbee\_part**为zigbee代码，这份代码包含协调器，路由器，终端设备的代码，用于路由器或终端设备采集温湿度光照数据发送回协调器，协调器再通过串口发送到T113开发板。

## 0.技术框图

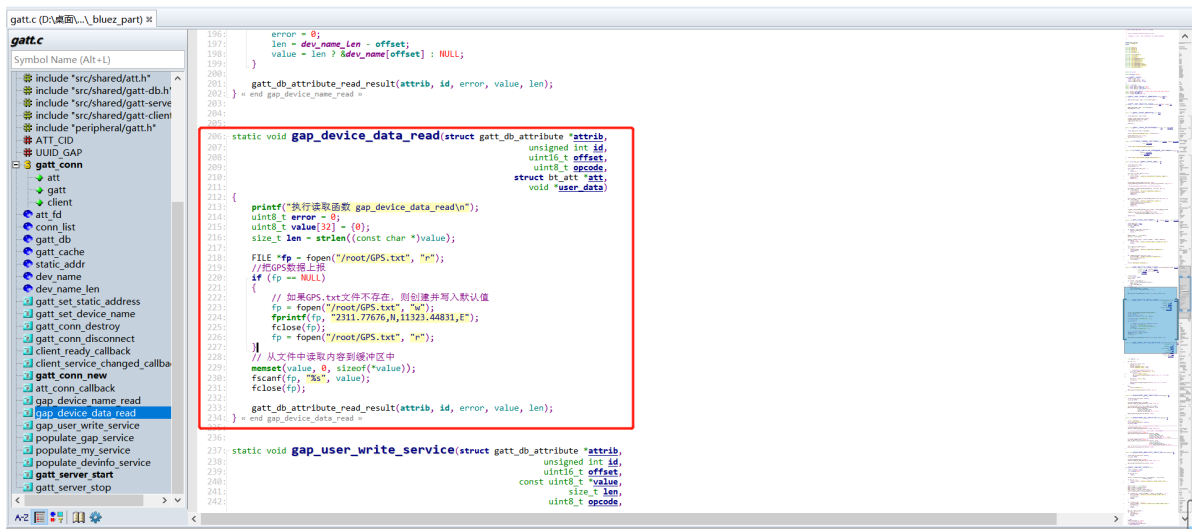


## 1.\_bluez\_part

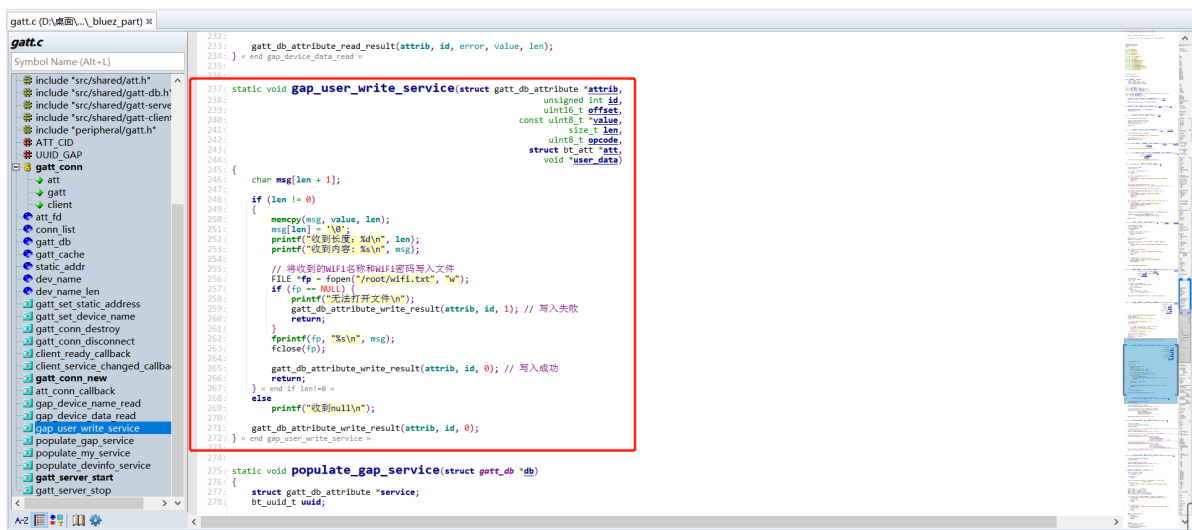
名称	修改日期	类型	大小
_bluez_part	2023/4/29 23:47	文件夹	
_edp_part	2023/5/5 14:41	文件夹	
_fork_edp_and_bluez	2023/4/26 22:56	文件夹	
_uniapp_part	2023/4/26 22:56	文件夹	
_zigbee_part	2023/4/29 15:27	文件夹	
readme.md	2023/4/26 22:53	Markdown File	1 KB

这部分代码，主要是实现了蓝牙模块的驱动，蓝牙模块主要实现了如下功能：

1.读服务，我是直接读取/root/GPS.txt的内容，后续可以想一下如何改进



1.写服务，更改WiFi密码的格式为"WiFi名称,WiFi密码"，例如:"Ace2,123780807"，Ace2为WiFi名称，123780807为WiFi密码



## 2.\_edp\_part

名称	修改日期	类型	大小
_bluez_part	2023/5/5 15:17	文件夹	
<b>_edp_part</b>	2023/5/5 15:16	文件夹	
_fork_edp_and_bluez	2023/4/26 22:56	文件夹	
_uniapp_part	2023/4/26 22:56	文件夹	
_zigbee_part	2023/4/29 15:27	文件夹	
readme.md	2023/4/26 22:53	Markdown File	1 KB

这部分代码，主要是设备与云平台(OneNet)的交互中间层代码，主要实现了如下功能：

## 2.1.主线程

1.串口接收ZIGBEE协调器设备串口数据，协调器设备数据主要包含路由器设备及终端设备通过Mesh网传输过来的**温湿度、光照强度**数据

```
229: /*3.把串口数据处理后放到buf数据缓冲区中上上传到onenet*/
230: /*3.1.准备要传到onenet上的数据*/
231: /*3.1.1.准备串口0获得的温度（temp）、湿度（humi）、光照强度数据（light）*/
232: onenet_value.temp = ((uart_buf0[0] - '0') * 10) + (uart_buf0[1] - '0');
233: onenet_value.humi = ((uart_buf0[2] - '0') * 10) + (uart_buf0[3] - '0');
234: if (!isdigit(uart_buf0[5])) { //如果第6位数据不为数字
235:     onenet_value.light = (uart_buf0[4] - '0');
236: }else{
237:     onenet_value.light = ((uart_buf0[4] - '0') * 10) + (uart_buf0[5] - '0');
238: }
239: }
```

2.串口接收GPS数据，GPS数据将被解析并存放至**gngga\_buf**结构体中，结构体成员详见**gps\_analyse.c**

```
266: /*3.1.2.准备串口1获得的GPS经度、维度数据*/
267: memset(&gngga_buf, 0, sizeof(gngga_buf)); //清空gngga_buf结构体
268: gps_analyse(uart_buf1, &gngga_buf); //将数据放进gngga_buf结构体，这个结构里暂时存储了解析后的GPS数据
269:
270: //2311.77676,N = 23+(11.77676/60) = 23.1962793°N
271: // (float)(2311/100) = 23.00000 (float)(2311%100) = 11.00000 ((float)77676)/10000 = 0.77676
272: onenet_value.latitude = (float)(gngga_buf.latitude/100) + (((float)(gngga_buf.latitude%100) + ((float)gngga_buf.latitude_x)/10000))/60;
273: onenet_value.longitude = (float)(gngga_buf.longitude/100) + (((float)(gngga_buf.longitude%100) + ((float)gngga_buf.longitude_x)/10000))/60;
274: }
```

3.根据时间及光照强度，在17:00开始进行5分钟的拍照

```
240: /*定时拍照功能*/
241: timestamp = time(NULL);
242: /*将时间戳转换为本地时间*/
243: struct tm *local_time = localtime(&timestamp);
244: /*判断是否是下午五点钟*/
245: if ((local_time->tm_hour == 17 && local_time->tm_min >= 0 && local_time->tm_min <= 5 && local_time->tm_sec >= 0)) { //下午五点钟开始拍照，拍个5分钟
246:     if (onenet_value.light < 50)
247:     {
248:         /* 点灯 */
249:         sprintf(command, "echo 1 > /sys/class/leds/red/brightness");
250:         ret = system(command);
251:         /* 拍照 */
252:         sprintf(command, "ffmpeg -y -i /dev/video0 -vframes 1 -s 1920x1080 -q:v 0 -f image2 ./test_photo.jpg");
253:         ret = system(command);
254:         sleep(1);
255:         /* 关灯 */
256:         sprintf(command, "echo 0 > /sys/class/leds/red/brightness");
257:         ret = system(command);
258:     }else{
259:         /* 拍照 */
260:         sprintf(command, "ffmpeg -y -i /dev/video0 -vframes 1 -s 1920x1080 -q:v 0 -f image2 ./test_photo.jpg");
261:         ret = system(command);
262:         sleep(1);
263:     }
264: } « end if (local_time->tm_hour=... » |
265: }
```

4.将温度、湿度、光照强度、经纬度数据上传至OneNet

```
275: /*3.2.清空原本的onenet数据流缓冲区，把新数据写进去，发送到onenet平台*/
276: switch (onenet_send_state)
277: {
278: case 0: //状态0发送TEMP数据
279:     memset(buf, 0, sizeof(buf));
280:     OneNet_SendData(buf, "Temp", onenet_value.temp);
281:     sleep(1);
282:     printf("Send Temp: %d \n", onenet_value.temp); // 打印
283:     onenet_send_state = 1;
284: case 1: //状态1发送HUMI数据
285:     memset(buf, 0, sizeof(buf));
286:     OneNet_SendData(buf, "Humi", onenet_value.humi);
287:     sleep(1);
288:     printf("Send Humi: %d \n", onenet_value.humi); // 打印
289:     onenet_send_state = 2;
290: case 2: //状态2发送LIGH数据
291:     memset(buf, 0, sizeof(buf));
292:     OneNet_SendData(buf, "ligh", onenet_value.light);
293:     sleep(1);
294:     printf("Send LIGH: %d \n", onenet_value.light); // 打印
295:     onenet_send_state = 3;
296: case 3: //状态3发送维度数据
297:     OneNet_SendData_float(1, onenet_value.latitude);
298:     sleep(1);
299:     printf("latitude: %f \n", onenet_value.latitude); // 打印
300:     onenet_send_state = 4;
301: case 4: //状态4发送经度数据
302:     OneNet_SendData_float(2, onenet_value.longitude);
303:     sleep(1);
304:     printf("longitude: %f \n", onenet_value.longitude); // 打印
305:     onenet_send_state = 0;
306:     printf("*****Send UART End*****\n"); // 打印
307:     printf("*****Send UART End*****\n"); // 打印
308: } « end switch onenet_send_state »
309: }
```

## 2.1.副线程

1.根据OneNet云平台下发的数据，实现补光灯的远程控制，命令格式为{"LEDSET": "1"}/{**"LEDSET": "0"**}

```

77: } else {
78:     /* 如果没有找到{"CAMERA\":"字符串，则继续查找{"LEDSET\":"字符串 */
79:     cmdStart = strstr(RecvBuffer, "{"LEDSET\":"");
80:     if (cmdStart == NULL) {
81:         /* 如果命令部分不存在，则执行默认操作 */
82:         continue;
83:     }
84:
85:     cmdStart += strlen("{LEDSET\":"");
86:     int cmdLen = strcspn(cmdStart, "\n");
87:
88:     /* 复制命令部分到一个新的缓冲区 */
89:     char cmdBuffer[20]; // 假设命令部分的最大长度为20
90:     strncpy(cmdBuffer, cmdStart, cmdLen);
91:     cmdBuffer[cmdLen] = '\0';
92:
93:     /* 根据提取出的命令执行相应的操作 */
94:     if (strcmp(cmdBuffer, "0") == 0) {
95:         /* 2.灭灯 */
96:         sprintf(command, "echo 0 > /sys/class/leds/red/brightness");
97:         ret = system(command);
98:     } else if (strcmp(cmdBuffer, "1") == 0) {
99:         /* 1.点灯 */
100:         sprintf(command, "echo 1 > /sys/class/leds/red/brightness");
101:         ret = system(command);
102:     }
103: } « end else »

```

2.根据OneNet云平台下发的数据，实现相机的远程控制，命令格式为  
{"CAMERA":"1"}/ {"CAMERA":"0"}

```

59: /* 查找命令部分的起始位置和长度 */
60: char* cmdStart = strstr(RecvBuffer, "{"CAMERA\":"");
61: if (cmdStart != NULL) {
62:     /* 如果找到了{"CAMERA\":"字符串，则执行相应的操作 */
63:     cmdStart += strlen("{CAMERA\":"");
64:     int cmdLen = strcspn(cmdStart, "\n");
65:
66:     /* 复制命令部分到一个新的缓冲区 */
67:     char cmdBuffer[20]; // 假设命令部分的最大长度为20
68:     strncpy(cmdBuffer, cmdStart, cmdLen);
69:     cmdBuffer[cmdLen] = '\0';
70:
71:     /* 执行相应的操作 */
72:     if (strcmp(cmdBuffer, "1") == 0) {
73:         /* 3.拍照 */
74:         sprintf(command, "ffmpeg -y -i /dev/video0 -vframes 1 -s 1920x1080 -q:v 0 -f image2 ./test_photo.jpg");
75:         ret = system(command);
76:     }
77: } else {

```