ESP8266 重启原因以及常见 Fatal Exception 原因



版本 1.0

版权 © 2016

关于本手册

本文介绍了 ESP8266 重启原因以及常见 Fatal exception 原因的判断方法。

本手册结构如下：

章	标题	内容
第 1 章	重启原因	介绍通过 ROM code 和用户程序判断 ESP8266 重启原因的方法。
第 2 章	常见 Fatal Exception 及原因	介绍 ESP8266 常见 Fatal exception 及原因。

发布说明

日期	版本	发布说明
2016.08	V1.0	首次发布。

目录

1. 重启原因 1

1.1. ROM Code 判断重启原因.....1

1.2. 用户程序判断重启原因 1

2. 常见 Fatal Exception 及原因.....3

1. 重启原因

1.1. ROM Code 判断重启原因

ESP8266 每次重启，ROM code 都会将此次重启原因的序号打印出来，如下图。用户可以根据该序号知道此次重启的原因。这种调试方式用于用户程序无法启动，但是需要分析此次重启原因时。

```
ets Jan  8 2013,rst cause:1, boot mode:(3,2)

load 0x40100000, len 31320, room 16
tail 8
chksum 0x69
load 0x3ffe8000, len 2132, room 0
tail 4
chksum 0x44
load 0x3ffe8860, len 7712, room 4
tail 12
chksum 0x11
csum 0x11
rf cal sector: 120
rf[112] : 00
rf[113] : 00
rf[114] : 01
```

ROM code 里面打印的重置原因如下表。

表 1-1. ROM Code 判断重启原因

Rst cause 号	原因
0	没有定义
1	电源重启
2	外部复位或 Deep-sleep 醒来
4	硬件 WDT 重启

> ⚠ **注意：**
> 软件 *WDT* 重启或者软件复位都会维持上次重启状态。比如第一次是电源重启，*rst cause* 为 *1*，软件复位后 *rst cause* 仍然为 *1*。

1.2. 用户程序判断重启原因

用户也可以通过应用层添加程序判断此次重启的原因。这种方式判断比较全面。该方式用于发生崩溃的位置打印乱码，无法调试。

添加代码段如下：

```
struct rst_info *rtc_info = system_get_rst_info();
```

```
os_printf("reset reason: %x\n", rtc_info->reason);
      if (rtc_info->reason == REASON_WDT_RST ||
            rtc_info->reason == REASON_EXCEPTION_RST ||
            rtc_info->reason == REASON_SOFT_WDT_RST) {
            if (rtc_info->reason == REASON_EXCEPTION_RST) {
                  os_printf("Fatal exception (%d):\n", rtc_info->exccause);
            }
            os_printf("epc1=0x%08x, epc2=0x%08x, epc3=0x%08x, excvaddr=0x%08x, depc=0x%08x\n",
                        rtc_info->epc1, rtc_info->epc2, rtc_info->epc3, rtc_info->excvaddr, rtc_info->depc);//此次会打印上次崩溃的地址，多用于崩溃时乱码调试。
      }
```

关于 `system_get_rst_info()` 和相关数据结构，请参考 SDK 的 API 参考：《ESP8266 Non-OS SDK API 参考》和《ESP8266 RTOS SDK API 参考》（链接为：*espressif.com/zh-hans/support/download/documents*）。

通过添加判断重启原因的代码，得到的重启原因如下表。

表 1-2. 用户程序判断重启原因

Rst cause 号	原因	GPIO 状态
0	电源重启	改变
1	硬件看门狗复位	改变
2	Fatal Exception	保持
3	软件看门狗复位	保持
4	软件复位	保持
5	Deep-sleep	改变
6	硬件复位	改变

2. 常见 Fatal Exception 及原因

当程序崩溃时，可以根据 Fatal exception 号协助调试。Fatal exception 号以及描述如下。

表 2-1. 常见 Fatal Exception 及原因

Fatal exception 号	描述	可能原因
0	非法指令	1. BIN 文件被破坏 2. 函数指针野指针
6	除数为 0	除数为 0
9	读/写操作地址非对齐	1. 读/写 Cache 地址不对齐 2. 函数指针野指针
28/29	访问非法地址	1. Cache 被关闭后访问 Cache 2. 函数指针野指针

比如：

```
Fatal exception (28):
epc1=0x4025bfa6, epc2=0x00000000, epc3=0x00000000,
excvaddr=0x0000000f, depc=0x00000000
```

- 如果使用的是 ***user1.1024.new.2.bin***，那么就在 ***user1.1024.new.2.S*** 文件里查证 exception 地址“0x4025bfa6”，然后在用户代码中添加打印进行调试。
- 如果使用的是 ***eagle.irom0text.bin***，就在 ***eagle.S*** 中查找 exception 地址。
- 如果查找不到对应地址，说明程序崩溃问题是在中断中发生或者是 ROM 里面的代码问题，比如：
  - ***4000e190 <memset>***
  - ***4000df48 <memcpy>***
  - ***4000dea8 <memcmp>***
  - ***4000de84 <bzero>***
  - ***4000e1e0 <strstr>***



乐鑫 IOT 团队

www.espressif.com

免责申明和版权公告

本文中的信息，包括供参考的 URL 地址，如有变更，恕不另行通知。

文档“按现状”提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档不负任何责任，包括使用本文档内信息产生的侵犯任何专利权行为的责任。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。

Wi-Fi 联盟成员标志归 Wi-Fi 联盟所有。蓝牙标志是 Bluetooth SIG 的注册商标。

文中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

版权归© 2016 乐鑫所有。保留所有权利。