

## 调试记录

运行环境：Windows7，64 位

软件版本：Microsoft Visual Studio 2010

硬件环境：2 个 Zigbee 模块，一个发送写好的数据（包），另一个接收数据，以接收数据的节点来模拟 25 号节点（这个过程称为[虚拟发送](#)），PC 端运行 C 语言接口程序，来接收该节点通过串口发送过来的数据并写入文件。

Bug:

A.接收两个半[数据包](#)后会被迫中止；

B.每个数据包的第二行格式不对（该行的数据个数超过了 1+24+2+4 个，以后的每个包都重复），而且第二行以后的数据格式都乱了。

情况分析 1：如果不接收数据，程序运行不会出现中断，所以 bug 处在处理接收数据上，有可能与文件操作不合理有关。

情况分析 2：格式不对可能与终止字符设置有关。

Debug:

修改 1：修改了 66Dlg.cpp 文件（见本文件夹中）中默认路径和默认串口号，省去每次运行程序都要改动这些项，详情见 Line 71&73，可根据不同电脑做相应修改；

修改 2：修改终止字符到 52，因为肖宇的 Zigbee 数据包每行最后一个数据为发送该数据包的节点 ID，即十进制数 52（二号节点），倒数第二个数据没有改原程序的，为十进制数 101（0x65），而接口程序原来的的终止字符为十进制数 101（0x65），而现在的终止字符应该是十进制数 52，但是修改后数据还是有小问题，详细数据见 Bug.B.csv 文件，问题还是出在第二行上，但是除该行以外其他行数据正常，这是因为程序遇到第二行数据第一个数据，即节点编号时，进行终结字符判断且判断成立，输出了时间然后另起一行了。

修改 3：鉴于修改 2 仍不能满足需要，现在我们不判断终止字符了，而改成通过数据长度（每行 27 个）将数据分成 24 行，详情见 66Dlg.cpp 的 line 228，至此 Bug.B 解决。

修改 4：经过反复分析程序，发现 bug.a 的原因：接口程序设置了接收最大时间 60 秒（详情见 66Dlg.cpp 的 line 274: if(current\_time - start\_time > 60)）和最大文件大小 8KB（详情见 66Dlg.cpp 的 line 258），一旦达到其中一个指标程序就会中止。我们将最大文件大小限制注释掉，最大时间限制暂时不动，在具体情况时做相应更改，至此 Bug.A 解决。

[修改 5](#)：发现[修改 2](#)在数据包长度较大情况会产生混乱，数据结构会错。现回归到最初的方式，仍采用终止字符方式，该方式的好处在于[一旦发生丢失数据包，接收数据只会乱几行，但若采用按长度分行，一旦丢包，其一个包的数据都会乱。](#)[虚拟发送](#)一组（27\*1 个）数据，发现能持续无误接收 9 分钟以上没问题，但如

果数据大小变为 27\*24，则持续无误接收时间只有 18 秒，以后会隔一段时间出现 3 个左右数据乱行，有待确定究竟是 C 语言接口程序出了问题还是数据传输过程中丢包之类的错误。