**CH573 BLE 无线更新**

# 1 User程序更新流程

准备两块CH573，一块作为“主机”，一块作为“从机”。使用WCHISPStudio将BIOS-RTOS-CH573-V1.11-20241104.hex烧写进芯片后，使用AHL-GEC-EXTERN-AUTO(V1.8)-20241105进行BLE-RF无线更新测试，更新的User程序为“AHL-User-RF-CH573-20241104”。

## 1.1 软硬件地址以及主从机设置

（1）打开“AHL-GEC-EXTERN\_AUTO(V1.8)”，点击左上角“用户功能内的“设备信息更新”，进入设备软硬件地址及主从机设置界面。

|  |
| --- |
| 图1 设备信息更新入口 |

（2）选择需要设置的设备后点击“连接GEC”连接上该设备，修改好下方硬件地址、软件地址、主从机类型后，点击右侧“更改地址”即可完成对相关信息的设置。更新过程中至少需要一个主机和一个从机，同时保证更新的主从机处于同一个硬件地址。

此处设置主机硬件地址为：0x12345678，软件地址为01（主机的软件地址实际不起效，在更新过程中视作00）；从机硬件地址为：0x12345678，软件地址为：01。

【注】执行该功能时建议手动选择需要连接的GEC，减少自动搜索占用的时间。

|  |
| --- |
| 图2 设置主机信息 |

## 1.2 User程序无限更新

（1）点击左上角“用户功能”内的“单个设备程序更新”，进入程序更新界面。

|  |
| --- |
| 图3 单个设备程序更新入口 |

（2）在左上角“1.从机信息”框内可以选择指定从机进行连接，如图所示，点击“连接GEC”后即会连接硬件地址为0x12345678，软件地址为01的COM29所在设备。再次点击“等待更新”按钮后会进入BLE-RF等待更新状态，等待来自主机的User程序更新信息。

【注】1.建议手动选择连接串口，减少遍历串口寻找对应设备的时间；2.如果从机已经运行了User程序，且该程序由文件夹的模板程序“AHL-User-RF-CH573-20241104”修改而来则具备被动的等待更新功能，无需手动“连接GEC”以及点击“等待更新”按钮，在文本框内确定好目标设备的软硬件地址，即可在收到主机更新信息的情况下自动跳转至RF等待更新状态；3.设备进入等待更新状态后会有2min的等待更新时限，如果在这段时间内没有收到更新信息会自动退出更新状态设备重启。

|  |
| --- |
| 图4 目标从机配置 |

（3）点击左下方“2.主机连接”框内的“连接GEC”即可让主机连接“1.从机信息”文本框内设置的软硬件地址所属从机。

【注】1.建议手动选择连接串口，减少遍历串口寻找对应设备的时间；2.如果信号较差无法连接上从机可以尝试在GEC40外接一根杜邦线、靠近主从机距离、多次尝试连接等方式。

|  |
| --- |
| 图5 主从机申请建立连接 |

（4）等待右下方“4.更新提示”内显示主从机成功建立连接后即可在右上方“3.导入hex文件”框内点击“选择文件”选择“AHL-User-RF-CH573-20241104”程序地hex文件导入，然后点击“一键更新”开始更新。

【注】1.更新分为两部分，一是主机解码并存储更新程序，而是主机向从机发送更新数据，请耐心等待；2.显示“程序更新已完成”后，从机会等待30s确认主机以完成全部更新，然后完成自身的程序替换更新。

|  |
| --- |
| 图6 主从机成功建立连接 |
| 图7 开始更新 |
| 图8 更新成功 |

# 2 更新逻辑优化

本节会简单介绍一下之前设计的BLE-RF的User程序更新流程，分析其缺陷与不足，然后介绍重新设计的程序更新流程及其对原先缺陷的优化与补足。

## 2.1 原先的更新逻辑

（1）上位机发给主机的就是User更新时的那一套东西，一帧长度 484B；

（2）主机并不和User更新一样先把收到的一帧消息解码处理一下，就直接往从机发，可以说，主机在“从机”和“上位机”之间只起到了一个数据中转的作用；

（3）发的时候因为CH573 BLE一次发送最大的数据长度是256字节，无法一次性发送484字节，所以将其分为两段分别发送，以下称为“帧段落1”和“帧段落2”；

（4）主机串口接收到上位机的更新帧之后，立刻向从机发送“帧段落1”和“帧段落2”，**只发一次**；

（5）在SysTick\_Handler中断里面差不多每隔100ms再向从机发送一个“结束帧”，这个结束帧就是告诉你“帧段落1”和“帧段落2”已经全部发送结束了；

（6）直到主机收到从机的回发消息（或是太久没收到回发消息）才会进行下一步处理，要么是发送下一帧，要么是重新发送这一帧；

**存在问题：**

（1）在中断里面直接调用了 tmos\_set\_event() 这个按照官方的说法来说不太好，中断里面不建议设置tmos事件；

（2）从机自主机收到的，要往flash里面写入的数据对齐不了扇区边界，到时候会有跨扇区写入的情况出现，在当前RAM资源紧张的情况下，存在隐患；

（3）主机通过结束帧确认“帧段落1”和“帧段落2”的发送情况，从结论上来说相当于用1个消息确认了3个消息的发送情况，从机对结束帧的回发实际上既确认了“帧段落1”、“帧段落2”也确认了自己，在信号差、消息发送频繁的情况下这种处理方式无疑是不合理的；

（4）还有就是代码为了实现以上的"分段发送"以及确认机制使用了大量的标志位用于判断，可读性差。

## 2.2 重构后的更新逻辑

（1）首先就是主机，不再只是作中转用途，主机完全有能力对上位机发送下来的数据进行解析存储，毕竟走的就是User更新那一套，到时候直接把数据处理好放在代码User更新段（代码B段）；

（2）然后到时候无线更新发送的数据直接从Host的B段去读，去截取，这样省了不少事；

（3）BLE一次发送的数据也只选择发送128字节，保证字节对齐避免Flash操作时出现跨扇区的情况；

（4）新的数据帧相当于就是代码本体，从机不用再做解析，直接往里往Flash内存就行；

（5）这样处理之后发送和接收可以自然地采取发一帧确认一帧的形式，避免上面发一帧确认三帧；

（6）新的帧除了帧头帧尾、帧长度、地址、编号之外还应该加上一个crc，确保其有自检能力；

（7）这么搞了之后好处就是可以把tmos里面个事件的检测相关的东西都放到主循环里面，不再去跑中断，感觉能避免很多隐性的问题。

|  |
| --- |
| 图9 主从机更新逻辑 |

# 3 注意事项

如果出现更新信号差，长时间无法推进更新进度的情况，可以尝试在GEC40号口上外插一根杜邦线作天线用，同时拉近两块板子的距离。