# 系统详细设计

## 需求介绍

前期由于项目需求实现与客户最终满意期望不一致，导致本次需要重新设计软件。根据客户需求，界面上需要实现如下功能。

### 发送消息模块

1. 支持串口设置（串口范围COM1-COM15）,串口打开和关闭；
2. 支持1-20个频道选择，同时一个通道界面要求能选择2个频道。；
3. 支持输入消息（消息长度暂定512字符），支持发送消息；
4. 支持发送消息的历史记录的显示；
5. 支持任意一条历史消息重发；
6. 支持清楚历史记录功能；
7. 界面显示三个通道，以便与分别发送不同类型的数据（例如发送字符串、文件、视频），目前三个通道均发送字符串。

### 接收机参数设置模块

1. 支持2个频道选择（频道范围为1-20）；支持2个COM口设置（串口范围COM1-COM15）；支持串口打开和关闭；
2. 支持送“连接接收机”命令；

该功能是分别对已打开的串口发送“频道”数据；

1. 支持发送“清楚数据”命令；

该功能是分别对已经打开的2个串口发送“清楚数据命令”；

1. 支持发送“锁定接收机”命令

该功能是分别对已经打开的2个串口发送锁定接收机命令。发送消息必须包括接收机ID的有效取值范围（有效范围不超过 [1,1024]区间）或者某个具体接收机ID值（该值也应该在[1,1024]范围内）。

### 其他要求

要求系统初始化时，“发送消息模块”中的“频道1”默认为“10”，“频道2”默认为“11”；

### 界面大致要求

界面图如下：



## 需求确认

### 消息发送模块

1. 发送消息后，收到响应的消息是否显示？已经确认需要显示响应的消息，当2个频道均有消息时，默认显示频道1消息。
2. 消息的发送与历史记录消息重发功能，涉及下发的消息类型是否一样，亦单片机是否区分重发消息与发送的消息？不一样

### 接收机参数设置模块

无

## 需求分析

### 用列设计

 

### 功能分析后重新设计界面

以客户界面为准暂不调整。

### 系统设计通信协议设计：

#### 通信协议结构TLV类型。

TLV结构，即type-length-vlaue结构。在通信过程中，以TLV类型的结构作为通信过程中数据传输的基本结构。

这样设计优点：

1. 通用性强，易于扩展。若后期有新消息类型，仅需要新增命令字，消息结构。然后在发送端和接收端做对应的新类型处理即可。

消息类型T:消息类型的命令字。本次需求命令字包括连接命令、锁定命令、清楚数据命令、发送消息命令、接收消息命令、重发命令、应答命令；

消息内容的长度L:消息命令的具体内容。如：锁定命令内容包括“频道数+接收机ID”。

消息内容V: 消息内容的起始地址。该地址为其实地址的连续Length空间为具体的消息内容。

#### 关于应答命令目前三种方案：

1. 每个命令的响应的应答命令均相同。

缺点：不能区分应答命令是属于哪一个命令的应答命令。例如：发送连接命令后，仅接着在发锁定命令，此时收到单片机的应答命令。此时，客户端软件无法区分是锁定命令的应答还是连接命令的应答。

优点：应答命令统一，命令类型少。

1. 每个发送命令对应不同的应答命令。即，锁定命令对应锁定应答命令，连接命令对应连接应答命令。

优点：应答比较可靠

缺点：命令类型多。

3、每个命令的应答命令类型均相同，增加V值不同。其中V值包含待发送的命令字。

考虑到后期每个发送命令可能均会有非确认帧的响应消息情况，在这种情况下每个响应消息类型均不同，即一个发送命令对应一个响应消息。若选择2方案设计确认帧，确认帧的实现与响应帧机制基本相同，这样导致消息类型数量随发送命令多少成2~3倍增加，这样对后期维护不是一个好的选择。

#### 最终选定方案：目前选定方案3；通信协议的重传机制：

为保证串口通信数据传输可靠，参考UDP传输协议，选择了采用“发送-确认”方式确保可靠的传输。即，上位机每发送一个消息到单片机，单片机收到消息后，需要先应答确认帧到上位机，之后再进行其他逻辑业务处理。

消息重传机制方案，考虑如下方案：

1. 不重传，仅仅给客户呈现发送失败的错误消息。

发送消息后，启动定时器（定时周期暂定5s），若在这段时间内未收到确认帧。则表明发送消息失败，给客户呈现失败消息。若收到确认帧，则做其他处理。

缺点：未收到确认帧不重传，可靠行稍低

优点：实现较易

1. 重传3次，3次重传失败后，给客户呈现发送失败消息。

发送消息后，启动定时器（定时周期为5s）。定时器处理操作，先检测是否已经接收确认帧，若已经接收确认帧，则停止定时器。若未接收并且重传次数少3次，则重传消息（该重传处理内不加入启动定时器操作）。若大于等于3次没有接收到确认帧，则停止定时器，给客户呈现发送失败消息。

优点：有重传，消息发送比较可靠

缺点：实现稍复杂。对于如下场景，正在考虑一种解决办法。当程序正处于重传期间，客户再次触发发送事件，这种情况下，会导致新发送数据与老数据冲突。

对于这种场景，目前想到一个解决办法是。发送事件触发后，需要先检测一下当前是否在重发数据，若在重发或者等待接收确认帧（定时器已启动则表明处于重发或者等待确认帧阶段），则暂时不能发送数据。上报“正在发送数据，请稍后发送”。

目前暂定选择方案2。

### 命令类型介绍

1. 连接命令

频道1；

频道2；

1. 锁定命令

频道1+接收机ID；

频道2+接收机ID;

1. 清楚数据命令

频道1；

频道2

1. 发送消息命令

频道1+消息内容；

频道2+消息内容；

1. 接收消息命令

消息内容

1. 重发命令

频道1+消息内容；

频道2+消息内容；

1. 应答命令

命令类型

通用消息接口：

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 名字 | 含义 | 备注 |
| UINT8 | ucHead | 消息头 | 0xAF |
| UINT8 | ucCmdType | 命令类型 | 当前为如上描述的几种命令类型 |
| UINT16 | usReserved | 保留字 |  |
| UINT16 | usLength | 数据长度 |  |
| UINT8 [2] | aucValue | 数据起始地址 |  |

消息头不含数据起始地址aucValue，大小为6个字节。

各个命令子消息内容：

连接命令：

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 名字 | 含义 | 备注 |
| UINT16 | usChannel | 频道 |  |

锁定命令：

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 名字 | 含义 | 备注 |
| UINT16 | usChannel | 频道 |  |
| UINT16 | usStartReciverId | 起始接收机ID | 1-65535 |
| UINT16 | usEndRecverId | 结束接收机ID | 1-65535 |

清楚数据命令：

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 名字 | 含义 | 备注 |
| UINT16 | usChannel | 频道 |  |
| UINT16 | usStartReciverId | 起始接收机ID | 1-65535 |
| UINT16 | usEndRecverId | 结束接收机ID | 1-65535 |

发送消息命令：

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 名字 | 含义 | 备注 |
| UINT16 | usChannel | 频道 |  |
| UINT16 | usMsgNum | 消息的序号 | 该消息来源与历史记录数+1。初始为1，后面发送消息该需要根据历史记录。 |
| UINT8 | aucMsg[512] | 发送消息内容 | 中文字符2个字节，最大可容256-1个中文字符，512-1个英文字符 |

接收消息命令：

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 名字 | 含义 | 备注 |
| UINT8 | aucMsg[512] | 发送消息内容 | 中文字符2个字节，最大可容256-1个中文字符，512-1个英文字符 |

重发消息命令：

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 名字 | 含义 | 备注 |
| UINT16 | usChannel | 频道 |  |
| UINT8 | usMsgNum | 重传消息的序号 | 发送的第一条消息，该值为1，后续发送消息依次递增。 |
| UINT8 | aucMsg[512] | 发送消息内容 | 中文字符2个字节，最大可容256-1个中文字符，512-1个英文字符 |

应答命令：

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 名字 | 含义 | 备注 |
| UINT16 | usAckCmdType | 需要确认的命令类型 | 不包括应答命令类型 |

约定各个命令类型与其对应值

|  |  |
| --- | --- |
| 命令类型 | 命令字的值 |
| 应答命令 | 0x01 |
| 连接命令 | 0x02 |
| 锁定命令 | 0x03 |
| 清楚数据命令 | 0x04 |
| 发送消息命令 | 0x05 |
| 接收消息命令 | 0x06 |
| 重发消息命令 | 0x07 |
| 无效命令 | >=0x07无效 |

以发送锁定命令为例子，说明发送以接收数据：

上位机发送的数据：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 说明 | 类型 | 字段名 | 值 | 备注 |
| 通用消息结构头 | UINT8 | ucHead | 0Xaf | 消息头 |
| UINT8 | ucCmdType | 0x03 | 锁定命令字 |
| UINT16 | usReserved | 0XFFFF | 默认值 |
| UINT16 | usLength | 4 | 命令内容字节数 |
| 命令内容 | UINT16 | usChannel | 10 |  |
| UINT16 | usStartReciverId | 1 |  |
| UINT16 | usEndRecverId | 20 |  |

单片机返回确认帧消息：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 说明 | 类型 | 字段名 | 值 | 备注 |
| 通用消息结构头 | UINT8 | ucHead | 0Xaf | 消息头 |
| UINT8 | ucCmdType | 0x01 | 锁定命令字 |
| UINT16 | usReserved | 0XFFFF | 默认值 |
| UINT16 | usLength | 0x0002 | 命令内容字节数 |
| 命令内容 | UINT16 | usAckCmdType | 0x0003 | 锁定命令 |

### 模块设计

一、发送消息实现方案（单串口）

1. 发送消息的主流程
2. 获取界面数据
3. 构造消息
4. 调用通用接口发送消息
5. 启动定时器
6. 定时器处理流程
7. 若消息发送成功，关闭定时器，计数器清零。
8. 若失败且计数器<3，计数器+1，并且调用接口发送消息。
9. 若失败且计数器>=3，关闭定时器，计数器清零。呈现发送消息失败消息。
10. 考虑出问题场景

当串口处于发送消息状态时，界面再次触发发送消息事件，系统如何处理？若系统继续发送消息，则系统发送消息可能会冲突，造成信息不可知结果。

系统应当返回提示信息，说明当前不可发送消息。对于这种情况解决办法是在主流程中新增一个判断是否是该场景，若是则直接返回的处理。

如何解决这个场景出现的问题，首先需要确定这种场景出现时，系统是个什么状态。即，串口处于发送状态，在这种状态下系统正在首次发送消息，或者通过定时器在重发消息。

当前想到方案：

新增“串口状态”标志，串口状态分为“空闲状态”，“发送状态”，其初始状态为“空闲状态”。目前没有将“接收数据状态”考虑在内，因为这个状态对，解决这个问题无关。

使用该状态时，应该在主流程里面调用发送消息接口前，设置状态为“发送状态”，然后在发送成功或者3次重传失败时候，再设置器状态为“空闲状态”。 该状态标记使用，跟重传计数器类似。也需要在主流程里面调用发送消息前，清零计数器。

1. 技术问题

VC6.0定时器如何使用？定时器打开和关闭

通用发送消息接口如何设计更好？

考虑：参数设计需要包括串口对象（发送消息时需要）、定时器ID（关闭定时器需要）、对应重传计数器地址（计数器清理和增加需要）、串口状态地址（设置串口状态需要）。这样通用性强，多个串口均可以直接调用。

二、发送消息实现方案（2个及2个以上串口）

1. 发送消息主流程

1) 判断当前系统状态是否符合响应发送事件的条件

2) 获取界面数据

3) 构造待发送的数据。由于两个串口发送数据不同，需要携带频道号。故需要分别构造2个串口数据。

4) 若串口1打开且串口1状态为空闲状态，则调用通用接口给串口1发送消息，启动定时器1；

5) 若串口2打开且串口2状态为空闲状态，则调用通用接口给串口2发送消息发送数据，启动定时器2；

1. 定时器处理流程

每个定时器的处理流程跟单串口下定时处理流程一样。区别在与此处多了一个定时器而已。

1. 考虑出问题场景

1)两个串口同时打开，一个发送成功，一个发送失败场景，是否会出现发送成功的串口重发数据的情况？ 不会，数据重传，在定时器里面处理，若成功定时器将会立即关闭，不会再次重传。

2)两个串口同时打开，均发送失败的情况。会先后呈现两个消息表征消息发送失败。暂无解决办法。

3)两个串口同时打开，均发送成功情况。无会问题情况。

4)一个串口打开，一个关闭情况。按照单串口处理一样。这里不考虑。

5)考虑符合响应发送的事情的条件有哪些?

！(串口打开状态&&发送状态（串口2）||串口1关闭&&串口2也关闭)

1. 相关状态标记考虑

当前需要用到状态包括“串口状态”+“串口关闭状态”+“计数器”

三、接收消息实现方案（单串口）

1. 接收消息主流程

1)读取串口数据

2)解析数据

3)调用通用发送消息接口响应ACK消息。（仅仅发送一次无管是否发送成功）

4)显示消息

1. 显示消息流程

1 )拷贝消息到待显示数据存储区

2)调用接口显示数据

四、接收消息实现方案（2个及2个以上串口）

1. 接收消息主流程

与单串口流程相同

1. 显示消息流程

考虑：两个串口同时显示消息情况，是否会出现“临界资源”同时访问的情况。需要考虑是否通过信号量或者互斥机制控制。

五、历史记录功能

1、在发送消息命令发送成功情况下，才历史记录里面添加记录。历史记录数目前暂定上限20条。超过20条数据自动从第一条覆盖。