

**产品型号**

**版本号V1.0.0**

**下位机模拟程序介绍**

**深圳市银河风云网络系统股份有限公司**

**Shenzhen GALAXYWIND Network Systems Co., Ltd.**

地址：深圳市高新技术产业园区北区新西路5号银河风云大厦

Address: GALAXYWIND building, No.5 Xinxi road, Shenzhen High-Tech Industry Park, NanShan district, China

邮编 P.C.: 518055

电话 Tel: +86-755-83400088

传真 FAX: +86-755-86139063

网站 Web: http://[www.galaxywind.com](http://www.galaxywind.com)

客服 Hotline: 400 000 9879

企业QQ: 4000009879

微信公众号 WeChat:



**版权声明**

版权所有©深圳市银河风云网络系统股份有限公司2015。深圳市银河风云网络系统股份有限公司（以下简称"银河风云"）对本资料进行版权声明，未经银河风云书面许可，任何单位及个人不得以任何方式或理由对本资料的任何内容进行复制、修改、抄录、传播。

**商标声明**

"银河风云"，"GALAXYWIND"等是深圳市风云实业有限公司的注册商标，本文档提及的其他商标、服务标志、注册商标及注册服务标志均为其所有者拥有。

**免责声明**

本资料会进行不定期更新，本资料中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目录

[1 设计概述 3](#_Toc435016429)

[1.1 设计目的 3](#_Toc435016430)

[1.2 通信框架 3](#_Toc435016431)

[1.3 实现方法 3](#_Toc435016432)

前言

**概述**

**产品版本**

与本文档相对应的产品版本如下所示：

|  |  |
| --- | --- |
| **产品名称** | **产品版本** |
|  |  |

**阅读对象**

本文档主要适用于以下用户：

结构设计工程师【√】

硬件开发工程师【√】

软件开发工程师【√】

系统规划工程师【√】

系统维护工程师【√】

**约定**

**符号约定**

文档中可能出现的符号定义如下所示：

|  |  |
| --- | --- |
| **符号** | **说明** |
| 警告警告 | 表示有中度或低度潜在危险，如果不能避免，可能导致较严重结果。 |
| 注意注意 | 表示有潜在风险，如果忽视这些文本，可能导致不可预知的结果。 |
| 备注备注 | 表示对正文的强调和补充。 |

**通用格式约定**

|  |  |
| --- | --- |
| **格式** | **说明** |
| 黑体 | 中文：1 -6级标题，图表/表格标题采用黑体 |
| Arial | 英文：1 -6级标题，图表/表格标题采用Arial |
| 宋体 | 中文：正文采用宋体 |
| Times New Roman | 英文：正文采用Times New Roman |

**修改记录**

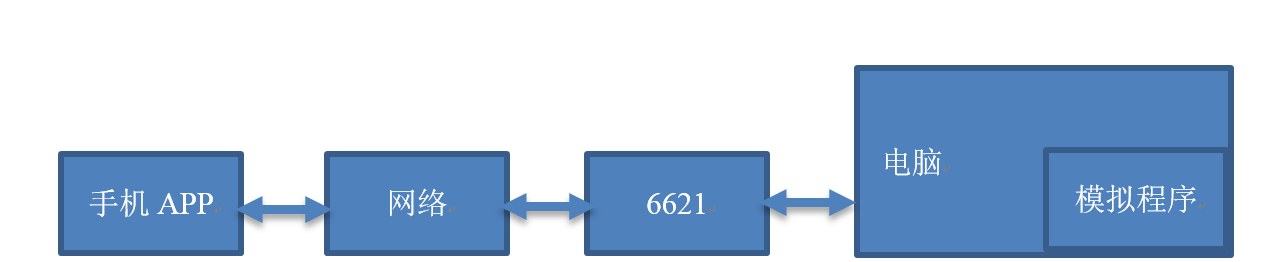
|  |  |  |  |
| --- | --- | --- | --- |
| **修订时间** | **修订人** | **审核人** | **修订版本** |
| 2015-09-22 | 冯胜 |  | V001R001 |

# 设计概述

## 设计目的

目前项目开发过程中，有些时候客户的样机来得比较慢，而且时间要求比较紧。虽然可以用串口助手等工具调试串口IO，但此类工具不能解析协议，不方便进行整个通信流程的调试。为了在设备端、sdk、app编码完成之后尽快进行联调，可用电脑来模拟下位机的行为，从而处理设备端下发的串口数据，解析协议，完成整个通信流程，提前进行联调。

## 通信框架



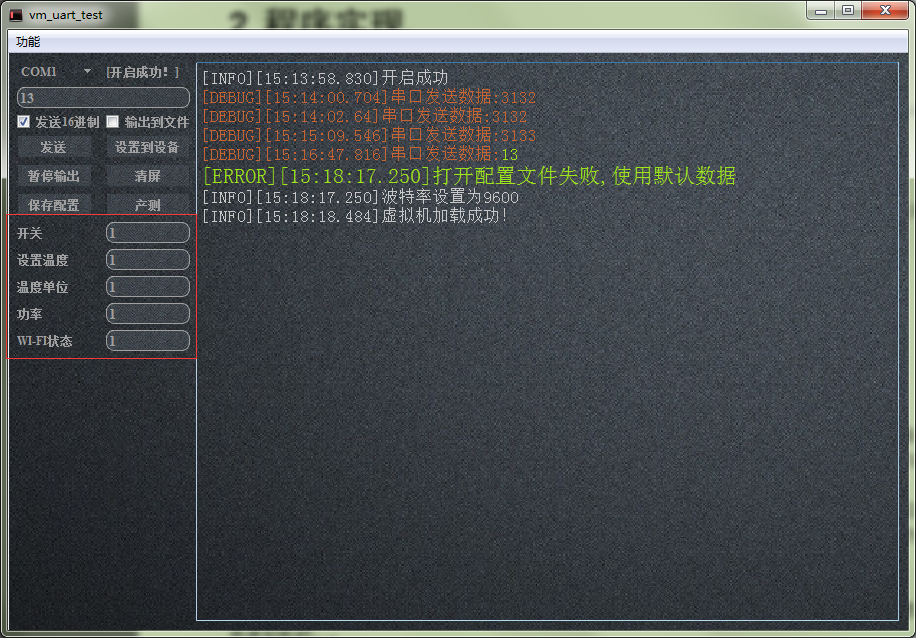
## 程序简介

程序使用Qt5.5结合EmbedVM开发，主要是由Qt提供界面相关、电脑端串口收发等API，EmbedVM进行数据处理并更新界面。开发新项目时仅需按照EmbedVM语法，结合提供的API框架，即可完成协议解析，实现简单的下位机模拟程序。

# 程序实现

## 程序显示框架

通常，下位机要处理的数据都是一些功能开关状态、功能参数值等，如开关状态，定时小时数等。每一个要显示的值，本程序使用一个label加一个输入框的结合体来显示，如图下红框部分：



Label用来表示这个值对应的意义，输入框用来显示值，每个值对应一个字节的内存，显示多个下位机的状态参数时，为了方便批量处理，使用一个状态数组来表示：

array8u shared\_uart\_stat[256];

每个要显示的控件在虚拟机中使用以下数据结构来定义：

$struct{

$member global8u type = 3;

$member global8u index = 1;

$member array8u name[40] = "设置温度";

}temp;

Type表示控件类型，3就表示label和输入框的结合体控件；index表示该结合体控件的值对应的内存在数组中的索引号，每个具体的项目开发时手动指定；name表示结合体控件中label显示的内容，对应上图的“开关”。

程序框架目前还支持按钮控件，定义如下：

$struct{

$member global8u type = 1; $member global8u call\_id = 10; $member array8u name[40] = "产测";

}pt\_botton;

Type为1表示这是一个按钮，call\_id是一个调用id，当这个按钮按下时会以这个id为参数调用虚拟机，虚拟机中只要在main函数中定义好处理这个id的代码即可。

这些控件在虚拟机中定义，初始化虚拟机时传入主程序，Qt主程序中维护所有的控件的数据结构。这样当要改变界面显示的值时，只需更改其对应的内存，然后调用界面刷新接口，该接口遍历所有控件并将每个输入框置为其数组索引对应在数组中的值；例如将状态数组memset为1，调用界面刷新接口后所有的输入框都显示1。反之，通过手动修改输入框的值，也可同步到状态内存中。

## 程序执行流程

Qt与EmbedVM的结合使用过程中，Qt占主动，在需要的时候才会调用虚拟机进行处理，虚拟机处理后将结果传回主程序。第一次调用虚拟机时，虚拟机要将其一个字节的内存地址和状态数组的起视地址回传给主程序。这个字节是用来存放call\_id的，Qt主程序可以直接赋值后调用虚拟机，即可实现带参数调用虚拟机的目的。状态数组的起视地址是用来处理显示值的，主程序和虚拟机使用同一内存空间。

虚拟机每次执行时main函数中会根据call\_id值来决定进行什么操作。第一次调用虚拟机时，Call\_id为0，这时会将call\_id的地址状态数组的地址传到Qt，以后Qt调用虚拟机就可以设置call\_id了。按钮的回调就是根据call\_id来实现的。目前定义call\_id有以下：

enum{

CALL\_VM\_ID\_UI\_INIT = 1,//初始化虚拟机定义的界面

CALL\_VM\_ID\_UART\_READ = 2,//当Qt主程序捕捉到串口数据时调用虚拟机来解析

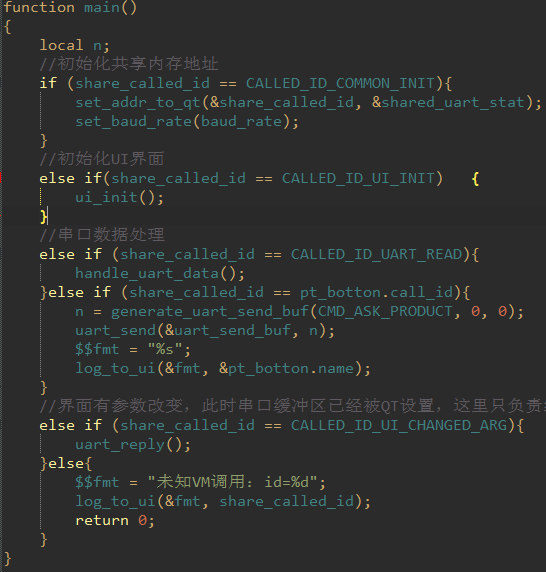
CALL\_VM\_ID\_UI\_CHANGED = 3,//Qt程序界面输入框的值改变后，点击“设置到设备”按钮时调用虚拟机处理，通常需要上报串口数据

};

此外每个自定义的按钮也对应一个call\_id，不能和上面重复。

程序的主要流程如图：

主程序启动，初始化基础控件



串口、UI输入、按钮按下等

事件循环

以id=CALL\_VM\_ID\_UI\_INIT调用虚拟机，初始化自定义控件

加载虚拟机，传出两个地址

虚拟机

Qt主程序

## 虚拟机代码框架

虚拟机代码中，main函数前面必须是下面的数组：

array8u before\_main\_flag[50] = "THIS\_IS\_USED\_TO\_CALC\_BEGIN\_ADDR\_OF\_MAIN\_FUNCTION!";

这个用来在加载虚拟机时寻找main函数地址。此外虚拟机中必须实现的操作：

1. 定义波特率，状态数组，call\_id（默认必须为0，用来传出地址）；
2. 实现以下3个call\_id的回调

enum{

CALL\_VM\_ID\_UI\_INIT = 1,//初始化虚拟机定义的界面

CALL\_VM\_ID\_UART\_READ = 2,//当Qt主程序捕捉到串口数据时调用虚拟机来解析

CALL\_VM\_ID\_UI\_CHANGED = 3,//Qt程序界面输入框的值改变后，点击“设置到设备”按钮时调用虚拟机处理，通常需要上报串口数据

};

另外当call\_id为0时需调用接口传出call\_id与状态数组的地址，第一次运行虚拟机时就会执行这个操作。

1. 每个自定义按钮实现其call\_id的回调；

具体的编写与所有的API请参考示例项目。

