# 基于 MFC 的智能车上位机软件

## 设计说明书

## 一、类设计

**【概述】**

该应用程序采用 MFC 应用程序中的单文档视图结构，这种结构利用文档对象保存用用程序的数据，依靠视图对象控制视图显示数据，文档与视图的关系是一对多的关系，即文档中的数据可以以不同的方式显示。图 1 表示了文档/视图与其他类对象的关系。

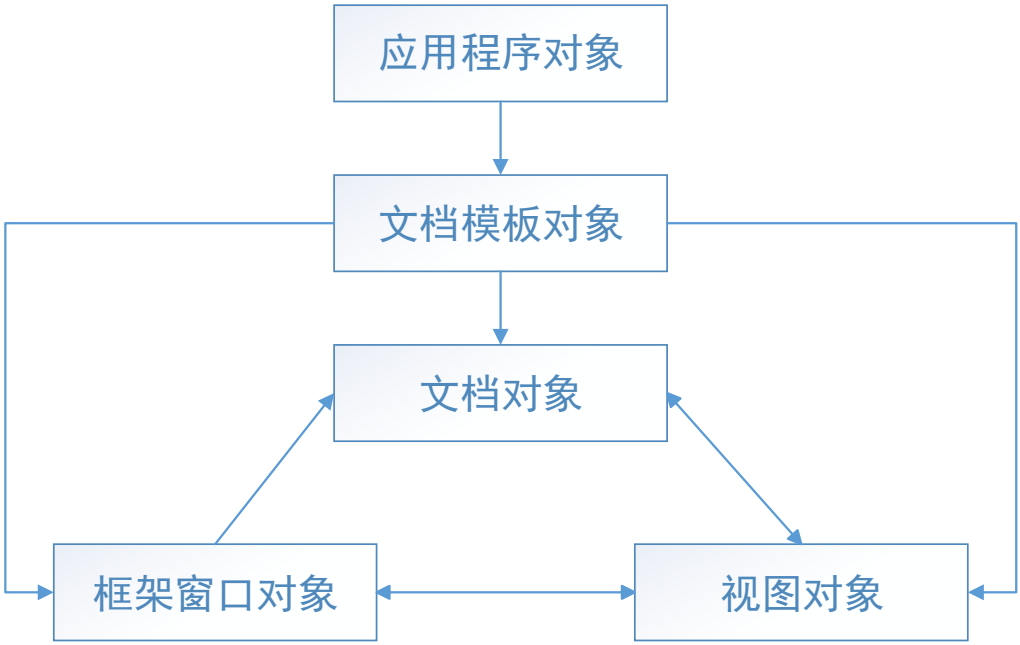


图 1 文档/视图与其他类对象的关系

**【类功能设计】**

基于 MFC 基础的文档视图结构创建的该应用程序主要包括以下几个类：

**应用程序类：**CSmartCarApp，基于 CWinAppEx 类；

**文档类：**CSmartCarDoc，基于 CDocument 类，负责管理文档数据；

**框架窗口类：**CMainFrame，基于 CFrameWndEx 类，负责管理协调各视图之间关系；

**控件窗口类：**CControlView，基于 CFormView 类，负责显示控件窗口，实现与用户之间的信息交互。

**视图类：**

CImageView，基于 CView 类，负责显示图像的原始数据；

CProcessView，基于 CView 类，负责显示处理之后的图像数据；

CPlotView，基于 CView 类，负责以波形的形式显示图像之外的其他参数，以实现虚拟示波器的功能。

**其他类：**

CMSComm 类，基于 CWnd 类，负责串口控件，实现数据的通讯管理；

CHistoGramDlg 类，基于 CDialogEx 对话框类，负责图像数据的灰度直方图显示；

（是 CControlView 的友元函数，方便跟 ControlVIew 控件窗口进行信息交互）

CWavePlotDlg 类，基于 CDialogEx 对话框类，负责虚拟示波器参数配置对话框；

CWavePlot 类，基于 CObject 类，负责虚拟示波器的曲线数据管理，主要为文档类的数据管理服务。

Image_class 类，将图像信息进行封装，用于图像的数据管理，为文档类的数据管理服务。

## 【类图】

文档类 CSmartCarDoc 调用 Image_class 和 WavePlot 类型变量，相互之间的类图关系如图 2 所示：

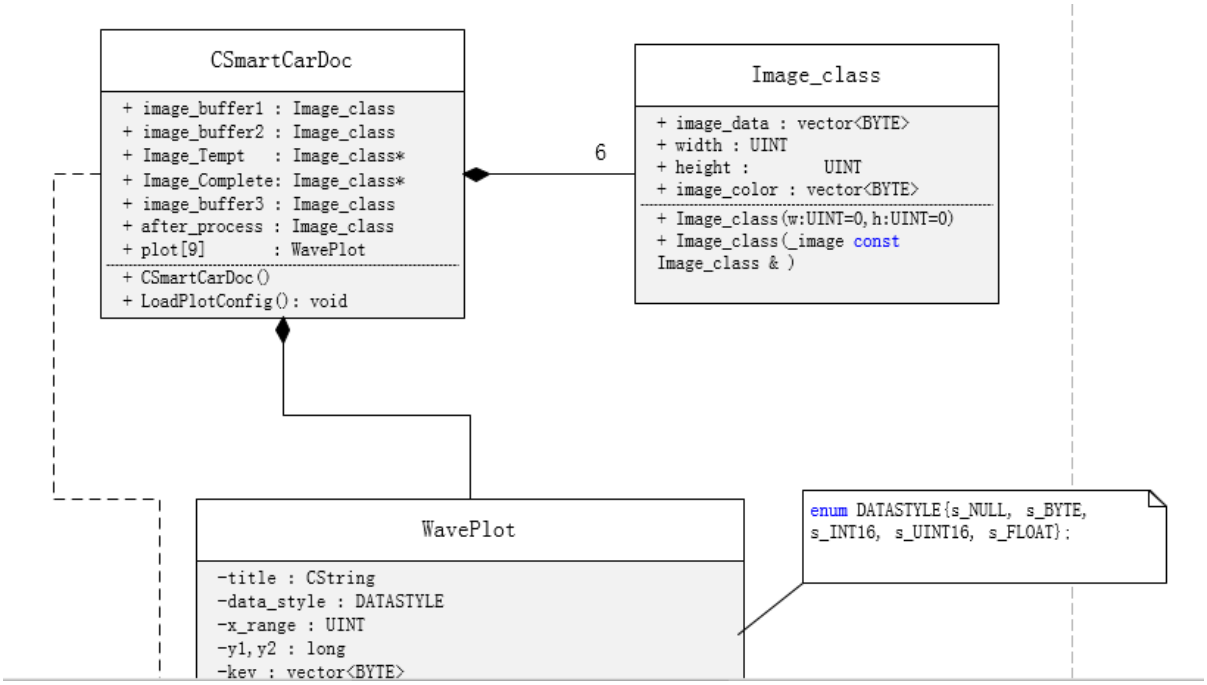


图 2.类图之间的结构关系

WavePlot 类图如图 3 所示：

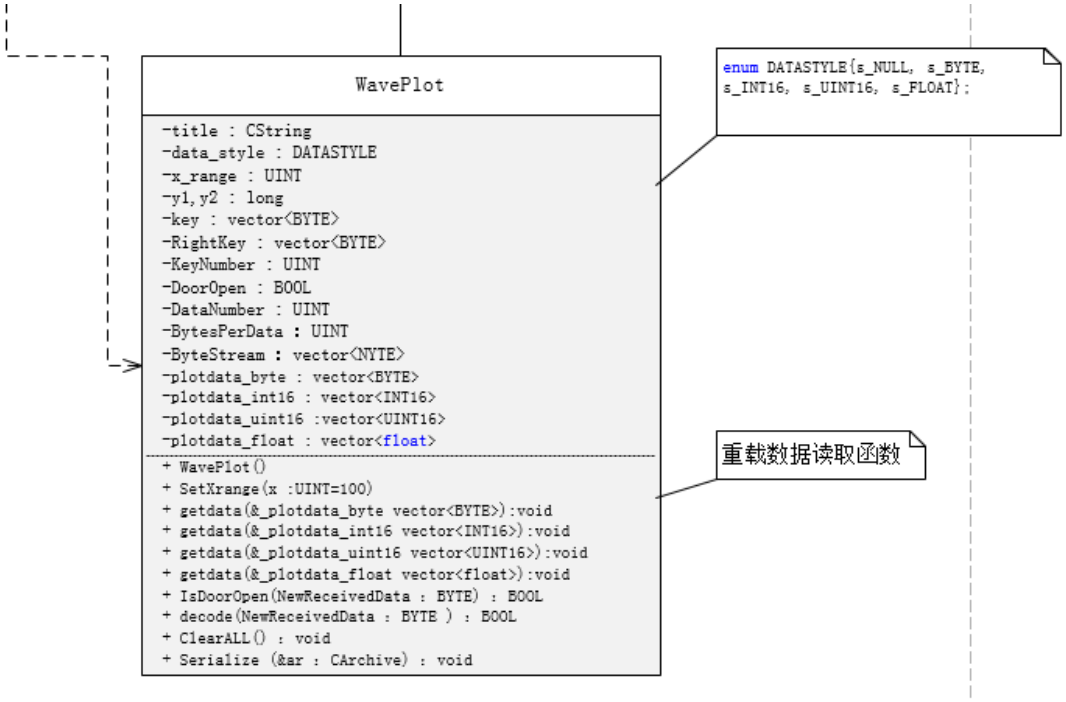


图 3.WavePlot 类图

此处只画出了自己设计的类的类图，其他类主要是添加了一些消息响应函数，由于函数过多就不一一列举，下面将简要介绍关键函数的功能实现。

**【类的关键成员及函数说明】**

（1） CSmartCarDoc 类

数据成员：

```
Image_class image_buffer1;//用于从串口接受图像数据缓冲区，双缓存
Image_class image_buffer2;//用于从串口接受图像数据缓冲区，双缓存
Image_class* Image_Tempt;//用于图像采集过程的指针
Image_class* Image_Complete;//用于指向采集完成的图像指针
Image_class image_buffer3;//从磁盘中调取图像数据
Image_class after_process;//显示处理后的图像，ProcessView 中显示
WavePlot plot[9];//定义 9 条 WavePlot 类的曲线
```

函数成员：

```
virtual void SetTitle(LPCTSTR lpszTitle);//设置应用程序标题
void LoadPlotConfig();//从文件中读取之前的曲线配置信息
```

（2） CImageView 类

函数成员：

virtual void OnDraw(CDC* pDC);　　//绘制原始图像

void Plot_Image(const Image_class _image);//在 ImageView 视图中绘制图像

（3） CProcessView 类

virtual void OnDraw(CDC* pDC); // 绘制处理之后的图像

afx_msg void OnImageprocess();//调用 ImageProcess.dll 动态链接库，显示算法处理之后的图像。

（4） CPlotView 类

virtual void OnDraw(CDC* pDC);　　//绘制 plot[9]中的曲线

（5） CControlView 类

数据成员：（数据成员涉及到控件，控件变量太多，在此不予罗列）

CHistogramDlg *m_pHistogram;//创建非模态灰度直方图对话框指针

CMSComm m_SCIModule;//串口控件变量

CComboBox m_ComNum;//设置串口号

CComboBox m_Baudrate;//配置串口波特率

UINT m_nWidth;//图像宽度

UINT m_nHeight;//图像高度

BYTE m_nThresh;//二值化阈值设置

函数成员：（重点函数成员）

void OnCommMscomm();//串口消息响应处理函数

void SwapImageBuffers();//交换双缓存指针

afx_msg void OnBnClickedBitmapopen();//打开位图文件

afx_msg void OnBnClickedBinaryButton();//图像二值化处理

afx_msg void OnBnClickedHistogramButton();//弹出灰度直方图

afx_msg void OnBnClickedOnalgorithm();//调用动态链接库算法

（6）WavePlot 类（自己设计的曲线类的数据及相关操作）

数据成员：

```
CString title;//曲线名称
DATASTYLE data_style;//曲线的数据类型
UINT x_range;//横坐标显示的数据点数
long y1,y2;//纵坐标的范围为y1~y2;
vector<BYTE> Key;//接收到的下位机端数据起始位，钥匙
UINT KeyNumber;//钥匙数目
BOOL DoorOpen;//起始位是否正确，正确则打开门进行数据接收
vector<BYTE> RightKey;//正确的两把钥匙
UINT DataNumber;//一次接收的数据个数
UINT BytesPerData;//一个数据所占字节数
vector<BYTE> ByteStream;//用于接收数据流管理
//用于接收数据存储
vector<BYTE> plotdata_byte;
vector<INT16> plotdata_int16;
vector<UINT16> plotdata_uint16;
vector<float> plotdata_float;
```

函数成员：

```
WavePlot();//构造函数
void SetXrange(UINT x=100);//设置横坐标范围
UINT GetXrange() //获取横坐标范围
void SetYrange(long _y1=0, long _y2=255); //设置纵坐标范围
//针对不同数据类型，重载获取数据函数
void getdata(vector<BYTE> &_plotdata_byte);
void getdata(vector<INT16> &_plotdata_int16);
void getdata(vector<UINT16> &_plotdata_uint16);
void getdata(vector<float> &_plotdata_float);
//曲线识别
void SetTitle(CString _title);//设置曲线名称
CString GetTitle();//读取曲线名称
void SetDataStyle(DATASTYLE _data_style);//设置数据类型
```

DATASTYLE GetDataStyle();//获取当前曲线的数据类型

void SetDataNumber(UINT _datanumber);//设置曲线每次接受的数据个数

UINT GetDataNumber();//获取应该接收的数据个数，用于串口解帧

void SetRightKey(vector<BYTE> _key);//设置正确的其实标志

vector<BYTE> GetRightKey();//获取正确的其实标志

BOOL IsDoorOpen(BYTE NewReceivedData);//判断是否拿到正确的钥匙

BOOL GetDoorStatus();//返回当前是否获得了正确的起始位

BOOL decode(BYTE NewReceivedData);//开门之后进行数据的解帧操作，将数据读入存储单元

void ClearALL();//数据接受完毕，清空状态标志

void Serialize (CArchive & ar);//序列化存储曲线的参数配置，将曲线参数信息保存成文件格式，应用程序退出后，仍能保存当前配置信息。

# 二、 主要技术难点和算法设计

技术难点：

1. MFC 单文档视图的整体结构把握，类与类之间的函数相互调用以及数据共享有很多的细节问题，要想操作自如需要花很大的功夫；
2. 串口通讯设计，尤其是虚拟示波器模块的“通讯协议”以及类设计；如何接收多条数据类型，起始位，数据数目不定的曲线信息，是程序逻辑设计的难点所在；
3. 位图文件的读取与存储，需要熟悉了解位图文件的结构；
4. 动态链接库的动态加载以及与单片机端 C 语言相协调的函数接口的设计。
5. 虚拟示波器最多支持 9 条曲线，如何保证下次运行应用程序时仍保存上次的配置信息对于实际应用而言具有很大的意义。

算法设计：

1. 图像的获取与显示采用了双缓存技术，实现后台采集与前台显示的独立，保证了数据的稳定与高效；
2. 运用 vector 容器进行数据管理，提高了数据处理的简便性以及数据接收过程中的安全性;
3. 串口通讯协议的设计，实现数据的正确稳定获取;
4. 如何在应用程序关闭之后再打开时保存大量的比较重要的配置信息？程序利用了

序列化信息存储的方法，将基本信息保存为二进制文件，使的程序更能满足用户的实际需求。