

触摸屏控制系统组态软件设计与实现

邹 进 甘永梅 肖煦媛 王兆安 西安交通大学电气学院(710049)

Abstract

The paper introduces a configuration software used in touch panel control system, especially describes the object oriented programming design, structure and implementation of graph configuration system.

Keywords: touch panel, configuration software, object oriented, visual c++

摘要

本文介绍了一种应用于触摸屏控制系统的组态软件,并详细叙述了面向对象程序设计、系统结构和图形组态功能的实现方法等关键技术。

关键词: 触摸屏, 组态软件, 面向对象, Visual C++

触摸屏组态软件为工程技术人员提供了一种采用搭积木的方式制作现场控制过程和控制界面的工具。它将控制系统现场环境中的各种控制对象,抽象为几类基本图形元素和部件,通过组态建立它们之间的控制关系,形成简洁、直观的控制流程图、趋势曲线以及人机交互界面。

1 系统构成

本文所述的组态软件 (TPDesigner) 是基于上海天任公司 TP 系列触摸屏和 PLC 所组成的小型工业控制系统。在此集散控制系统中,触摸屏作为上位机主要完成两方面功能:①监视:以数据、曲线、图形、动画等方式来显示 PLC 的状态和存储器数据,从而直观反映出控制系统的流程和状态。②控制:通过触摸来改变 PLC 内部状态位和存储器数值,从而参与过程控制。

下位机采用可编程控制器,完成工业控制系统的数据采集和控制功能,并通过串口与触摸屏进行通信。

组态人员通过触摸屏组态软件,在计算机上开发出系统监控组态界面后,编译生成二进制代码,并通过串口将生成的监控组态界面下载到触摸屏系统中。系统运行时,由用户在触摸屏上直接进行监视和控制操作。系统构成如图 1 所示。

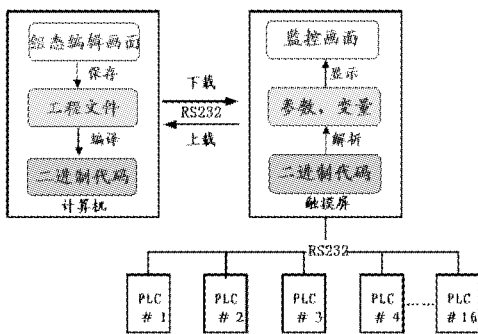


图 1 触摸屏监控系统构成

2 软件的主要功能与特点

TPDesigner 组态软件主要是参照日本 Digital 公司在 1996 年推出的全新触摸屏图形组态软件 GP-PRO/PB3 设计完成的。

2.1 图形编辑

图形编辑系统是触摸屏组态软件的一个重要组成部分,触摸屏运行时的各种显示界面都是在图形编辑开发环境下生成的。通过绘图和编辑操作,用户可以完成矢量图形绘制、位图编辑、趋势曲线绘制、报警信息设定、数据配方设置以及在触摸屏上模拟键盘输入等功能。

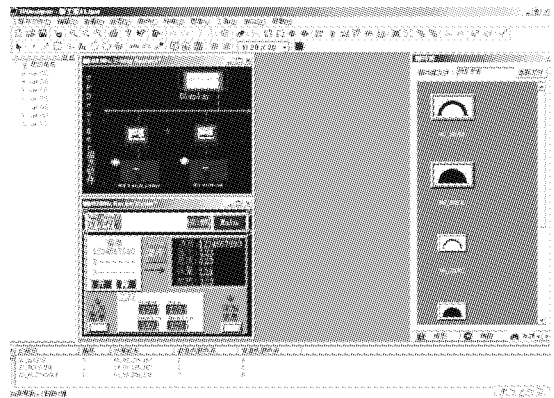


图 2 TPDesigner 软件界面

图形编辑系统还包括了一个丰富的部件库,内含大量的开关、仪表、指示灯等工业控制系统常用设备图符、仪表图符。利用部件库,组态人员可以高效快捷地开发出各种监控流程画面,并可方便地进行编辑。除此之外,图形编辑系统还提供了功能完备的部件编辑工具,使用户可以自行开发出符合特殊需要的部件,以满足不同控制场合的需要。

2.2 动画功能

每种图元对象均有影响其外观的属性,属性可以是静态的,也可以是动态的。设置动态属性的过程也叫做动画连接。选取需要组态的对象后,在动画连接设置对话框中选取对应的功能键并进行简单的参数填写即可完成。

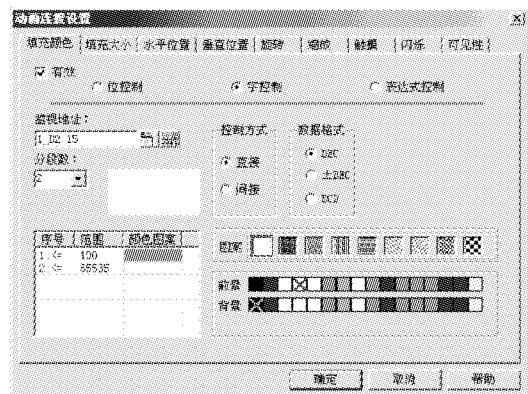


图 3 动画连接设置界面

2.3 模拟功能

触摸屏组态软件还提供了计算机模拟功能。主要是通过计算机模拟生成一系列组态过程中设置的过程变量，并通过地址信息列表界面手动修改这些过程变量的值，从而改变图元对象的动态属性，达到调试系统运行界面的目的。模拟功能可以使组态人员不必将组态界面下载到触摸屏中，直接在计算机上模拟出触摸屏运行时的动态效果。

3 软件的设计与实现

3.1 面向对象程序设计

组态软件要完成大量图形元素的操作和数据管理功能，其结构十分复杂。采用面向对象程序设计(OOD)方法，能够把复杂的系统简单化、直观化，而且易于用面向对象的编程语言实现。

面向对象的设计方法把系统中所有的图元对象和操作都抽象为类和类的操作，从相似的对象中找出它们共同的特点并用一个类来表示，所有的图元对象和操作都从这个类派生出来。这个类就称为“基类”，由它派生出来的类称为“子类”。子类继承基类的所有特性，从而获得了共有的数据和操作，编程人员只需开发属于不同对象特色部分的功能即可。子类不仅可以继承基类的属性和行为，而且也可以通过重载基类的某个行为(虚函数)，实现多态性。利用这一特点，我们可以方便地进行功能的修改：引入基类的一个子类，对要修改的一些行为进行重载，也就是对函数的功能重新进行定义。由于不是在原来的功能中进行的修改，所以彻底解决了软件的可维护性，为程序的修改完善提供了方便。采用面向对象的方法，把不同的功能和属性封装到不同的类中，也减少了系统各部分之间的耦合关系，使得模块之间的界限较为清晰，大大提高了软件的可靠性和健壮性。

我们采用 Microsoft Visual C++ 6.0 集成开发环境，基于 MFC(Microsoft Function Classes)进行开发，实现了触摸屏组态软件的人机界面、图形操作及算法的开发设计。

3.2 基本结构

组态软件中的各类图元对象在程序中统称为实体。实体主要分为三大类，其中一类是基本的矢量图形(基本图元)，如直线、矩形、椭圆、多边形、圆弧、刻度线、文本等。另一类是部件，包括开关部件、图形部件和数据显示部件等。除此之外还包括一些特殊部件，如趋势图、报警信息画面、数据配方画面和键盘画面等。

通过对各种实体进行分析，可以发现它们具有一些相同的属性，如实体的 ID、尺寸和逻辑坐标、实体的可见性和容器等；以及一些相同的操作功能，如子对象调整、编译和拷贝、保存到文件、从文件中读取等。我们将这些实体共同的属性和操作放在一个类中，建立实体类的基类 CTPEntity，它是从 MFC 的 CObject 类继承下来的，支持标准的 MFC 操作例如动态创建对象和文件的序列化等。由 CTPEntity 再派生出三个子类 CTPBasicFigure、CTPCommonParts 和 CTPSpecialParts，分别对应于三类实体对象：基本图元、部件和特殊部件。然后从基本图元类派生如直线、矩形、椭圆、矢量文本等基本图元；从部件类派生出开关部件、图形部件和数据显示部件等部件；从特殊部件类派生出趋势图、报警信息、数据配方、键盘画面等特殊部件。

动画连接作为实体共有的一类特殊属性，由专门的动画连接类 CTPAnimationLink 来实现。它也是从 MFC 的 CObject 类继承下来的，支持序列化的功能。从 CTPAnimationLink 类派生出颜色动画、填充动画、缩放、旋转和触摸等各类具体的动画连接类。类结构体系如图 4 所示。

3.3 组态功能的实现

3.3.1 图形的绘制

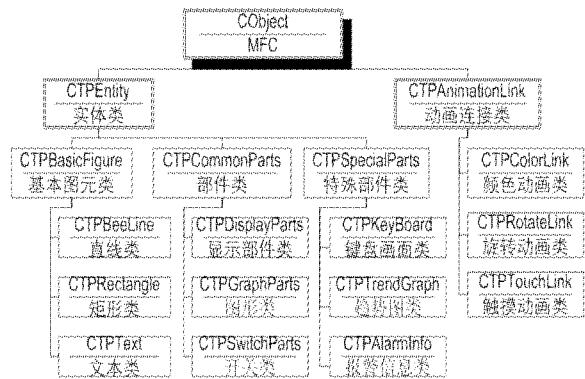


图 4 类衍生结构图

图形的绘制必须结合鼠标和键盘消息完成。当光标在视图窗口内部时，在按下鼠标、键盘或拖动鼠标的过程中，系统会产生标准的 Windows 窗口消息，不同的消息处理函数负责响应这些鼠标和键盘消息。我们定义了绘图工具类基类 TPDDrawTool 以及相关的消息处理虚函数 OnLButtonDown()、和 OnMouseMove()，由 TPDDrawTool 派生出不同实体对象的绘图工具类，每个具体的绘图工具类都重载了以上这些虚函数。程序的视图类 CTPDrawView 接收到鼠标或键盘消息时，CTPDrawView 类中的消息处理函数被调用，它再根据不同的绘图类型，调用具体绘图工具类中已经重载的消息处理函数，来执行绘制工作。

在绘图过程中，当屏幕绘图区域上实体对象非常多的时候，数据量很大。这时对绘图区域进行操作，如添加、删除和调整时，重绘过程比较缓慢，屏幕会出现严重的闪烁现象。为了解决屏闪问题，在绘图过程中采用了双缓存机制。具体的作法是：在屏幕绘图区域中显示我们想要的图形之前，先在内存中创建一个与屏幕绘图区域一致的缓冲区，将图形绘制到这个缓冲区中，再一次性将缓冲区中的图形拷贝到当前屏幕绘图区域上，最后释放内存缓冲区，这样能大大加快绘图的速度，避免出现屏闪。

3.3.2 图形的调整

在组态的过程中，需要对实体对象进行拖曳、移动、旋转和缩放等操作。实现此类操作通常采用的方法是：使用 MFC 提供的 CRectTracker 类，把它作为实体对象的容器，每个对象嵌入到容器中，利用 CRectTracker 类的成员函数实现实体对象的拖曳、移动和调整大小功能。

使用 CRectTracker 类有很大的局限性。它只支持对单个实体对象的整体调整，当需要对实体对象进行局部调整(例如：单独调整多义线对象的其中一个点)时，CRectTracker 无法完成此功能。另外，CRectTracker 类不支持镜像、旋转操作。因此，我们对 CRectTracker 类进行了功能扩展，定义了实体容器类 CTPEntityContainer 以及相关的成员函数。进行实体对象调整时，每个实体对象通过重载实体的 AttachContainer 函数，将自身绑定到指定的容器上，由 CTPEntityContainer 类实现移动、缩放、旋转和镜像等功能。

在调整过程当中，我们采用绝对和相对逻辑坐标的方式来计算实体的坐标。实体相对于屏幕左上角的坐标定义为绝对逻辑坐标。由于实体都存在于一个容器之中，并且容器又可以存在于外层容器(父容器)中，所以在存储它们的坐标值时，应存储相对于父容器原点(矩形区域左上角)的逻辑坐标值，这个坐标称为相对逻辑坐标。CTPEntityContainer 类提供了绝对逻辑坐标和相对逻辑坐标之间的转换函数。进行实体对象调整时，每个实体对象通过转换函数，先将用于描述自身的绝对逻辑坐标转换

(上接第 39 页)

成相对于父容器的相对逻辑坐标,然后进行相应的调整操作。采用这种方式,可以大为缩小实体对象移动和缩放过程中的计算误差,从而避免了调整过程中图形的失真。

3.3.3 动画连接

通过给实体对象添加动态属性的过程,可以为具体的动画连接类实例化一个对象。在实例化这个对象时得到这个动画连接对象的指针,并为每个实体对象建立一个指针链表来管理这些指针。我们采用 MFC 的 CObList 链表类对动画连接对象的指针进行组织和管理。CObList 类能够为 CObject 类和其子类的对象提供类型安全保证,使编译器进行更广泛的类型检查,并减少使用类对象时需要的类型转换,帮助消除由不匹配的指针类型引起的错误。从 CObList 派生出自定义类型的动画连接链

表类 CTPAnimationPtrList,并通过对 CTPAnimationPtrList 类的操作,完成动画连接对象的添加、删除和修改。

4 结束语

触摸屏组态软件 TPDesigner,在天任公司 TP 系列触摸屏上的应用表明,该软件操作简易灵活,用户界面友好,图形组态功能丰富,控制系统中的各种显示仪表、开关、流程图和趋势曲线都可以通过它来实现。

参考文献

- 1 王亚民,陈青.组态软件设计与开发.西安:西安电子科技大学出版社,2003
- 2 H.M.Deitel and P.J.Deitel.C++程序设计教程(薛万鹏等译).北京:机械工业出版社,2000

[收稿日期:2004.10.23]