SVG 组态 WEB 的使用说明

作者：高强

2017 年 9 月 16 日

目录

一、 QTouch 安装与使用

1．1 QTouch 下载

QTouch 软件的下载可以通过官网地址下载 http://sitcsys.com

下载路径：下载与演示 -> QTouch 组态软件 -> QTouch 通用版（2.2.5）

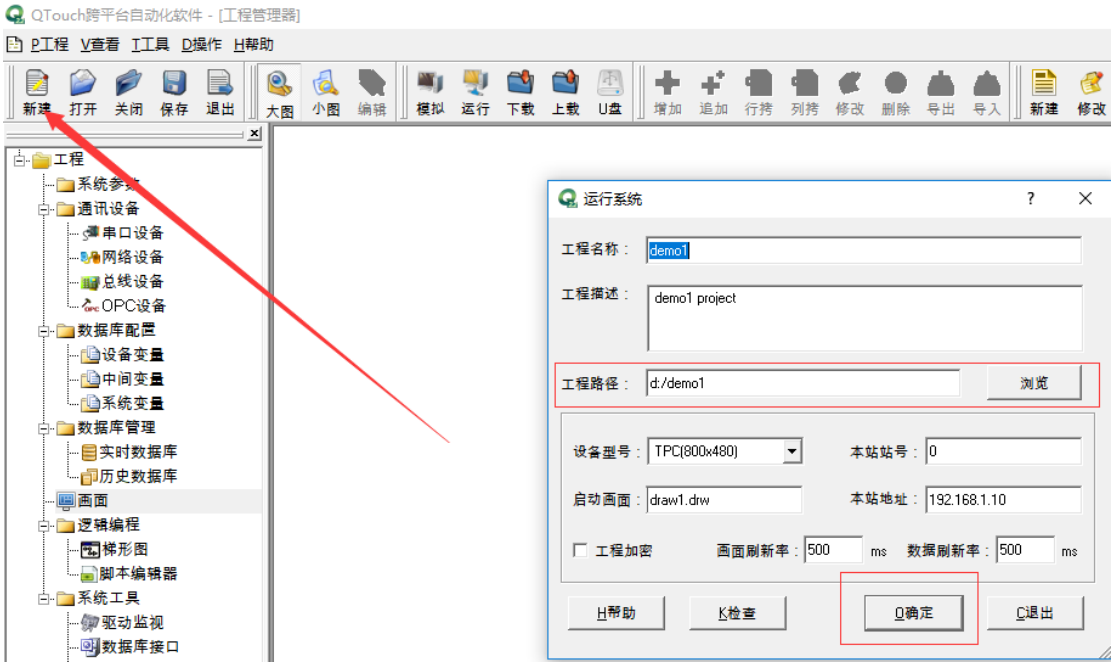
1．2 QTouch 安装

软件下载完成后，运行安装程序包，完成软件的安装。安装过程较简单，具体安装步骤不再此赘述。

1．3 QTouch 绘制系统

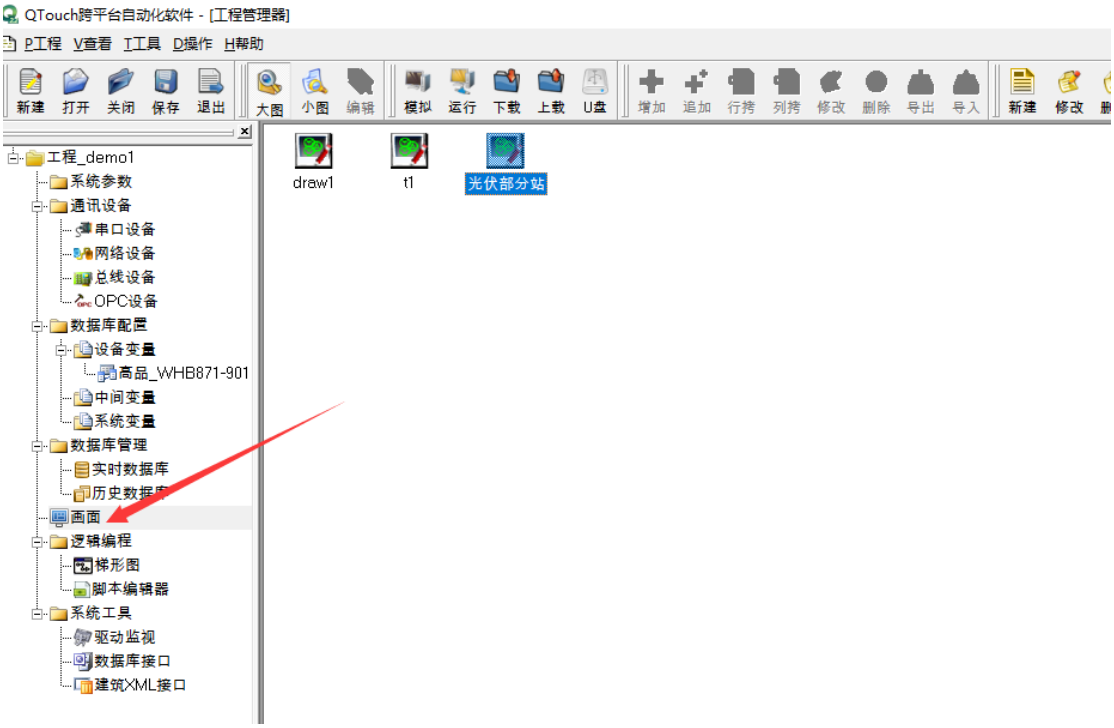
软件安装完成之后，双击桌面快捷方式打开软件，即可进入 QTouch 组态软件。组态软件界面依次分为：菜单栏、工具栏、工程菜单栏（左树形）、信息视图、状态栏。在工程菜单栏中可以找到“画面”，此项即为绘制系统。

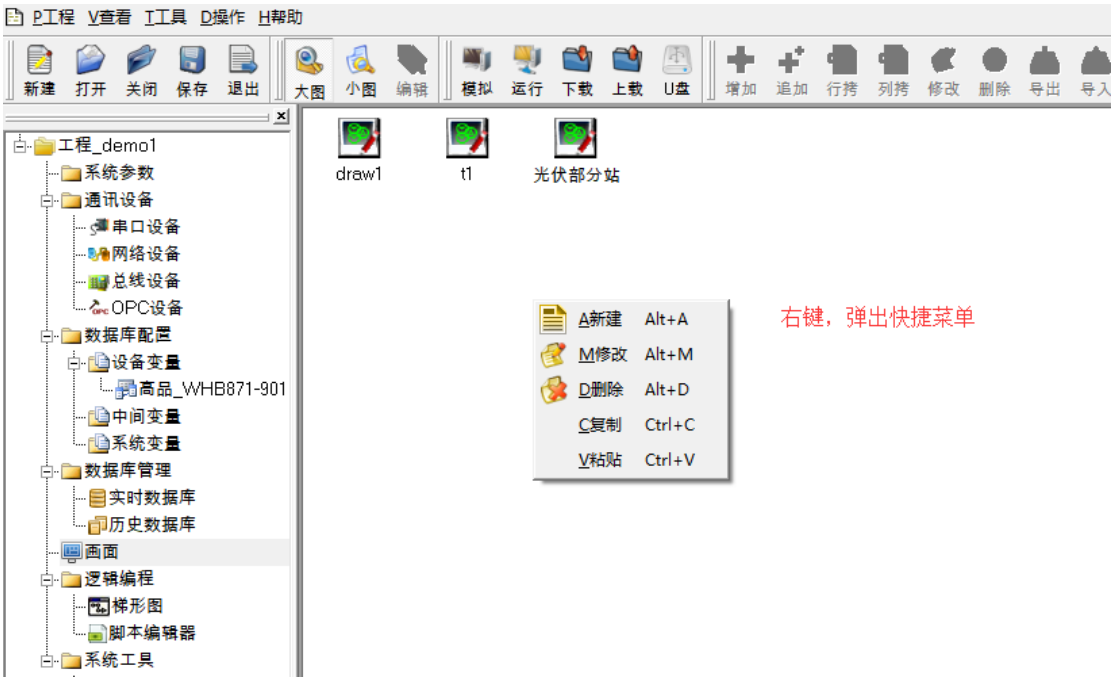
由于 QTouch 项目是以工程为单位进行管理，在进入绘制系统前，需要先创建工程。依次选择执行如下操作：



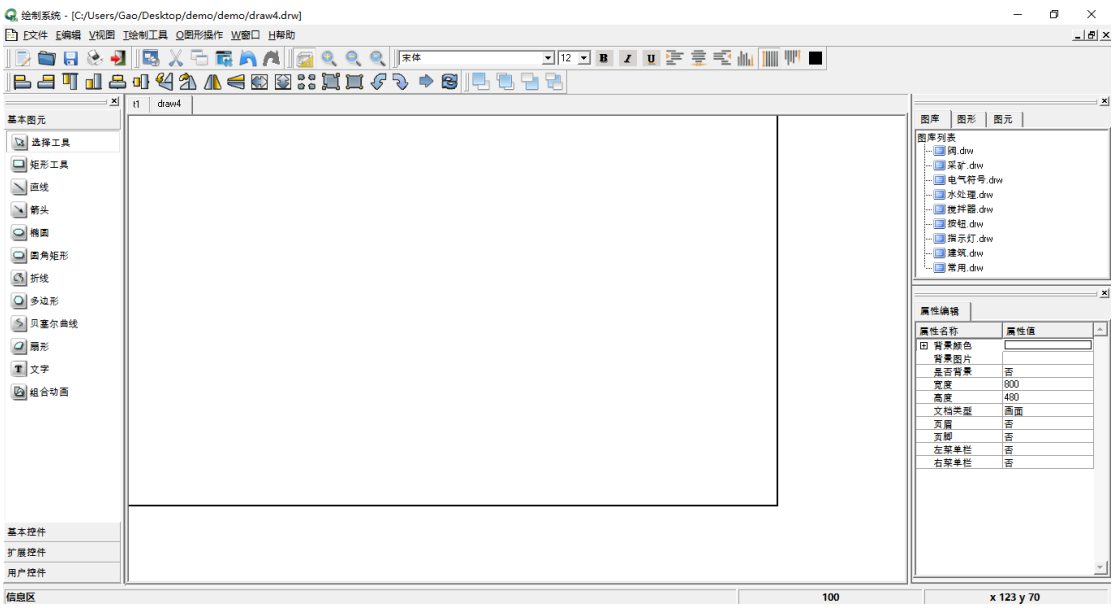
单击工具栏“新建”按钮，在弹出的“运行系统”对话框中配置：“工程路径”，配置完成之后 ，单击“确定”按钮即可。

以上步骤执行完成后，单击“工程菜单栏”中的“画面”按钮，在随后出现的“信息视图”窗口中，右建弹出快捷菜单，如下图所示：





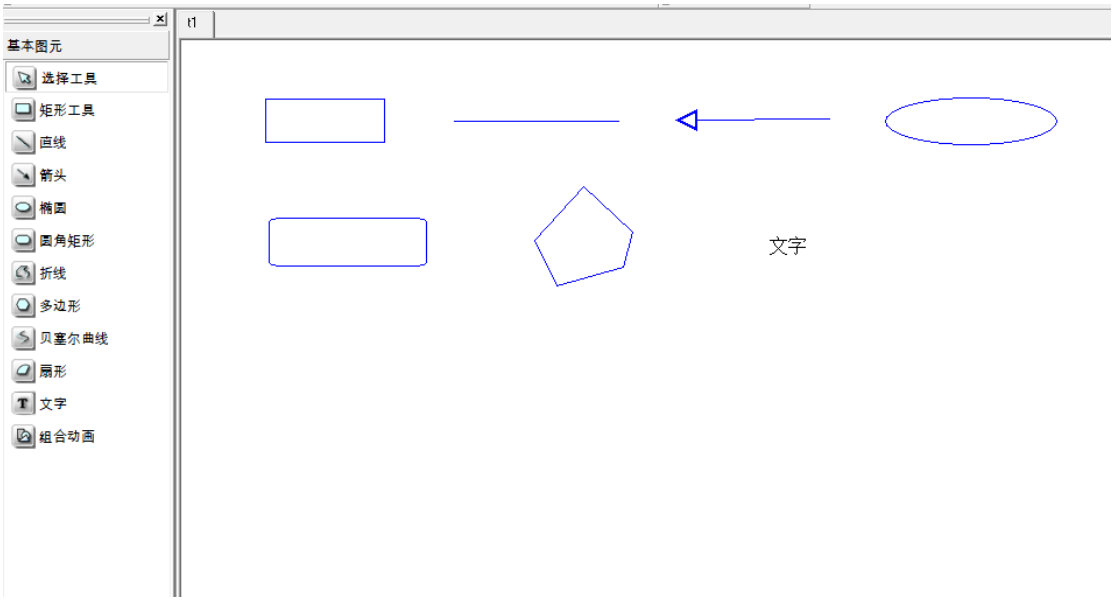
选择“新建”，则会在“信息视图”中新添加一个  ，此即为新增加的页面。选中新增加的页面，并双击图标，即可进入绘制系统界面。绘制系统界面如下图所示：



二、 绘制画面

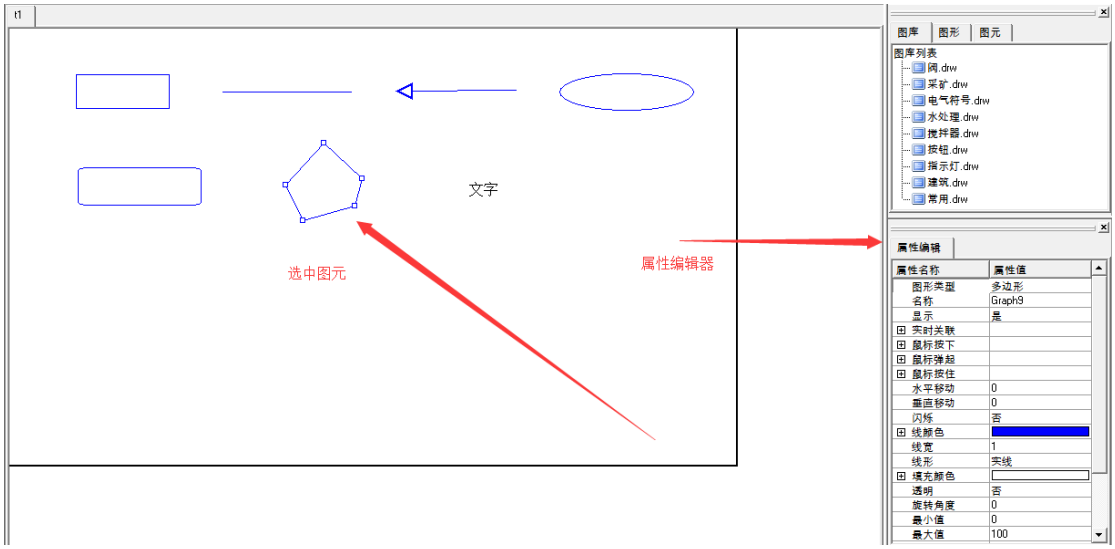
2.1 基本控件

QTouch 绘制系统提供了基本图元、基本控件、扩展控件和用户控件四类。目前在 SVG 转换中，只提供了基本图元。使用基本图元可以满足 80%监控系统的需求。其中基本控件包括：矩形、直线、箭头、椭圆、圆角矩形、折线、多边形、文字和组合动画。如下图所示：



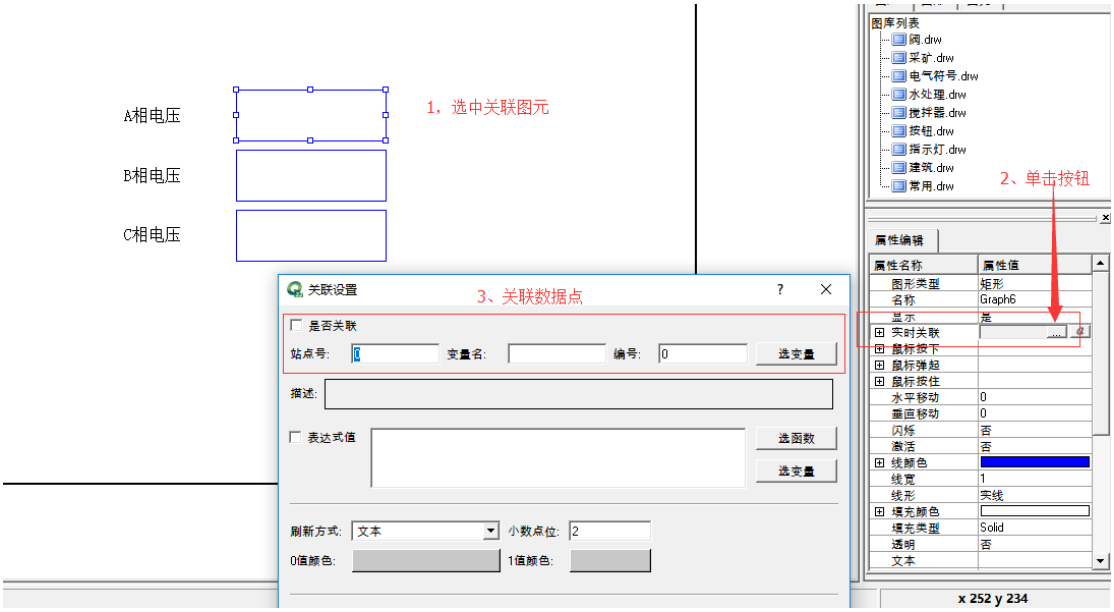
2.2 属性配置

QTouch 绘制系统内的所有图元都提供了相对应的属性设置，包括：背景、大小、鼠标动作、线形、字体、位置、快捷键等。在绘制系统中添加图元后，使用“选择工具”选中图元，就可以看到属性编辑器，如下图所示：



2.3 图元与数据关联

选中图元，在“属性编辑”器中，单击“实时关联”按钮，在弹出的“关联设置”中配置所需关联的数据点即可。注：QTocuh 绘制系统已经开放变量关联数据接口，用户可根据自己数据采集系统中的数据点标识进行数据关联。在经过 QTouch 转为 SVG 时，会在图元附属信息中包含该图标数据点标识字段，为 WEB 端加载和刷新图元关联数据提供对应关系。



三、 WEB 系统加载 SVG

3.1 SVG 加载与刷新

现下所有的浏览器基本都支持 SVG 图元的加载，在 WEB 系统中前端可直接使用 Frame 控件加载 SVG。SVG 加载完成后，用户的 WEB 系统可根据自己的业务逻辑按条件控制 SVG 显示与隐藏，放大与缩小。

SVG 图元数据刷新的方式有多种，为达到实时数据刷新的目的，可以使用 websocket、mqtt 等多种通信方式，具体通信过程由用户系统自行商定。QTouch 提供基于 Mqtt 的完整通信解决方案。

加载 SVG 代码片段如下，用户也可以访问开源中国，搜索 QTouch 开源组态软件获取完整的 Demo 代码，地址:http://120.27.140.246

```
    function getsvg(mqtt_value) {
                        if (!!window.ActiveXObject || "ActiveXObject" in window)
            {
                var arr = JSON.parse(mqtt_value);
                var linedata = arr.lines[0].data;
                var stationid = arr.istation;
            }else{
            var arr = $.parseJSON(mqtt_value);//将 json 数据转换为数据
            var linedata = arr['lines'][0]['data'];
            var stationid = arr['istation'];
            }

            if(sid == stationid)
            {
                //更改 svg 数据
                svg1 = document.getElementById("realView").getSVGDocument();
                svg1.documentElement.style.cursor="pointer";

                //刷值
                var texts = svg1.getElementsByTagName('text');
                $.each(texts,function(i,item) {
                    if(item.childNodes.length > 0)
                    {
                        if (!!window.ActiveXObject || "ActiveXObject" in window)
                        {
                            var                    objectid                    =
item.childNodes[3].childNodes[3].attributes['ObjectID'].value;
                            //获取图元上的 ObjectID
                            $.each(linedata,function(l,ltem) {
                                if(ltem['index'] == objectid && objectid >=0)
                                {
                                    item.childNodes[1].textContent            =
ltem['value']//当收到的 index 等于 ObjectID 的时候给其赋值
                                }
                            });
                        }else
                        {
                            var                    objectid                    =
item.children[1].children[1].attributes['ObjectID'].value;
                            $.each(linedata,function(l,ltem) {
                                if(ltem['index'] == objectid && objectid >=0)
                                {
                                    item.children[0].textContent = ltem['value']
                                }
```

```
                        });
                    }
                }
            });

}
```

3.2 放大缩小控制

为更加丰富 SVG 在 WEB 系统中的体验效果，QTouch 在转为 SVG 时已将放大、缩小等操作内置其中，代码片断如下。完整的代码请访问开源中国，搜索“QTouch 开源组态软件”。

```
var SVGDocument = null;
      var SVGRoot = null;
      var TrueCoords = null;
      var GrabPoint = null;
      var BackDrop = null;
      var DragTarget = null;
      var width = 800;
      var height = 400;
      var gridLength = 20;
      var scale = 1;
      var svgPanel = null;
      function Init(evt){
         SVGDocument = evt.target.ownerDocument;
         SVGRoot = SVGDocument.documentElement;
         TrueCoords = SVGRoot.createSVGPoint();
         GrabPoint = SVGRoot.createSVGPoint();
         BackDrop = SVGDocument.getElementById("BackDrop");
         svgPanel = SVGDocument.getElementById("testdrag");
         if(document.addEventListener){
            document.addEventListener('DOMMouseScroll',scrollZoom,false);
         }
         window.onmousewheel=document.onmousewheel=scrollZoom;
         document.documentElement.style.overflowY = 'hidden'; //???????
      }

      function scrollZoom(e){
          e=e || window.event;

          if(isFirefox=navigator.userAgent.indexOf("Firefox")>0){
                e.wheelDelta>0 || e.detail >0?zoom(0.9,e):zoom(1.1,e);
          }else{
                e.wheelDelta>0 || e.detail >0?zoom(1.1,e):zoom(0.9,e);
          }
      }
```

```
      function zoom(num,e){
        scale *= num;
        var sb = getMousePos(e);
        var sbx = sb.x;
        var sby = sb.y;
        svgPanel.setAttribute("transform-origin","0 0");

svgPanel.setAttribute("transform","scale("+scale+"),translate(-"+sbx+",-"+sby+"
)");
      }

      function Grab(evt){
         var targetElement = svgPanel;
         if (BackDrop != targetElement){
            DragTarget = targetElement;
            DragTarget.parentNode.appendChild(DragTarget);
            DragTarget.setAttributeNS(null, "pointer-events", "none");
            var transMatrix = DragTarget.getCTM();
            GrabPoint.x = TrueCoords.x - Number(transMatrix.e);
            GrabPoint.y = TrueCoords.y - Number(transMatrix.f);
         }
      };

      function getMousePos(event) {
        var e = event || window.event;
        return {'x':e.clientX,'y':e.clientY}
      }

      function Drag(evt){
        GetTrueCoords(evt);
         if (DragTarget){
            var newX = TrueCoords.x - GrabPoint.x;
            var newY = TrueCoords.y - GrabPoint.y;
            DragTarget.setAttributeNS(null, "transform", "translate(" + newX + ","
+ newY + "),scale("+scale+")");
         }
      };

      function Drop(evt){
         if (DragTarget){
            var targetElement = svgPanel;
            DragTarget.setAttributeNS(null, "pointer-events", "all");
            if ("Folder" == targetElement.parentNode.id){
               targetElement.parentNode.appendChild( DragTarget );
```

```
      }
      else{}
      DragTarget = null;
   }
};

function GetTrueCoords(evt){
   var newScale = SVGRoot.currentScale;
   var translation = SVGRoot.currentTranslate;
   TrueCoords.x = (evt.clientX - translation.x)/newScale;
   TrueCoords.y = (evt.clientY - translation.y)/newScale;
};
```