



# AicVision 快速入门

北京艾克信控科技有限公司

目 录

1	AicVision简介 .....	3
2	利用AicVision构建一个简单的设备监控工程.....	3
2.1	前提.....	3
2.2	在AicDatahub中建立模型和对象.....	3
2.2.1	登录AicDatahub .....	3
2.2.2	建立罐体的对象类型 .....	5
2.2.3	建立罐体的对象实例 .....	7
2.3	在AicIOserver中建立IO链接 .....	8
2.3.1	新建IO工程 .....	8
2.3.2	通道/设备/变量.....	9
2.4	使用AicStudio创建桌面监控画面 .....	12
2.4.1	绝对绑定 .....	14
2.4.2	相对绑定 .....	18
2.4.3	利用脚本切换相对绑定对象.....	20
2.5	使用AicStudio创建Web监控画面.....	24
2.6	在AicStudio中创建一个动画 .....	26
2.7	为Tank建立液位报警 .....	31
2.7.1	报警和报警区 .....	31
2.7.2	实时报警显示 .....	35
2.7.3	历史报警显示 .....	37
2.8	趋势曲线.....	38
2.8.1	实时趋势曲线.....	38
2.8.2	实时库配置历史记录.....	41
2.8.3	历史趋势曲线.....	42

## 1 AicVision简介

AicVision 是一款全新的基于实时数据库的 SCADA 套件，它主要包括 AicDatahub（工业实时库）、AicStudio（图形开发工具）和 AicIOserver（IO 服务器）三个组件。它全面采用 OPC UA 这一最新的工业通信国际标准，实现基于 Internet/Intranet 的远程数据采集；数据服务和画面彻底分离，为构建大型的分布式监控系统提供了强有力的支持；创造性地引入对象模型理念，实现面向对象的数据处理和画面组态；在一套服务端，同时支持桌面、浏览器和移动平台多种客户端访问。

## 2 利用AicVision构建一个简单的设备监控工程

本文档将结合一个简单发酵罐监控的项目，说明如何利用 AicVision 构建一个监控工程。

### 2.1 前提

在根据本文档操作之前，请确保已经正确安装以下软件：

- AicDatahub：实时数据库
- AicIOserver：IO 服务器
- AicStudio：客户端编程工具

AicDatahub 可以安装在本地或局域网的其它计算机上，并确保其防火墙已经开放了 4520（opc ua 的默认端口）端口。

### 2.2在AicDatahub中建立模型和对象

#### 2.2.1 登录AicDatahub

运行 AicDataHub Configurator 后，首先进入的是登陆界面（如图 2.1 所示）。用户必须正确的填写服务器的地址(Server Uri)、填写正确的用户名(默认 sa)及密码(默认 sa)，然后点击“确定（OK）”按钮方可登陆成功进入主界面。否则将会弹出错误信息，提醒用户输入错误。点击“Cancel”按钮则会直接退出系统。

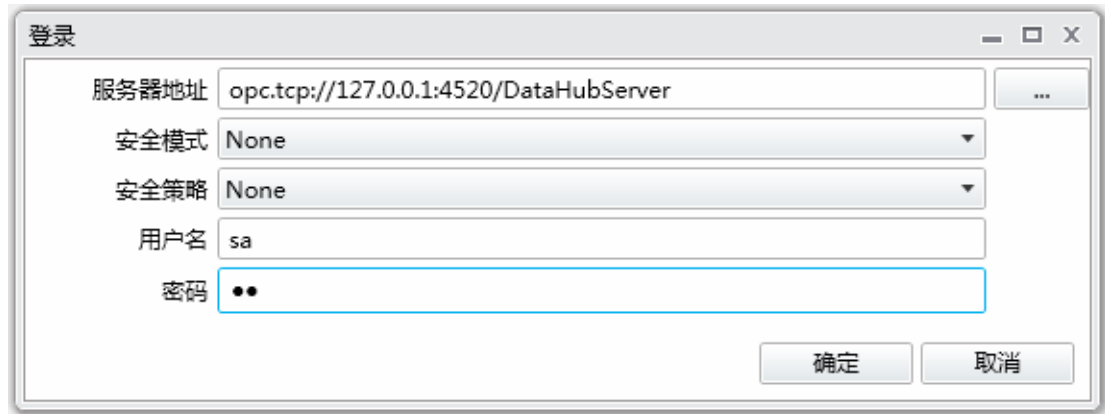


图 2-1 登录 Datahub

如果不确定服务器的 Uri，可以点击登录界面右侧的  按钮，打开服务器地址选择窗口（如图 2.2 所示）。在 Host Name 中输入服务器 IP 地址或主机名，然后点击右侧的  按钮，则会在终端列表中展示所有通信协议和安全模式的终端信息。

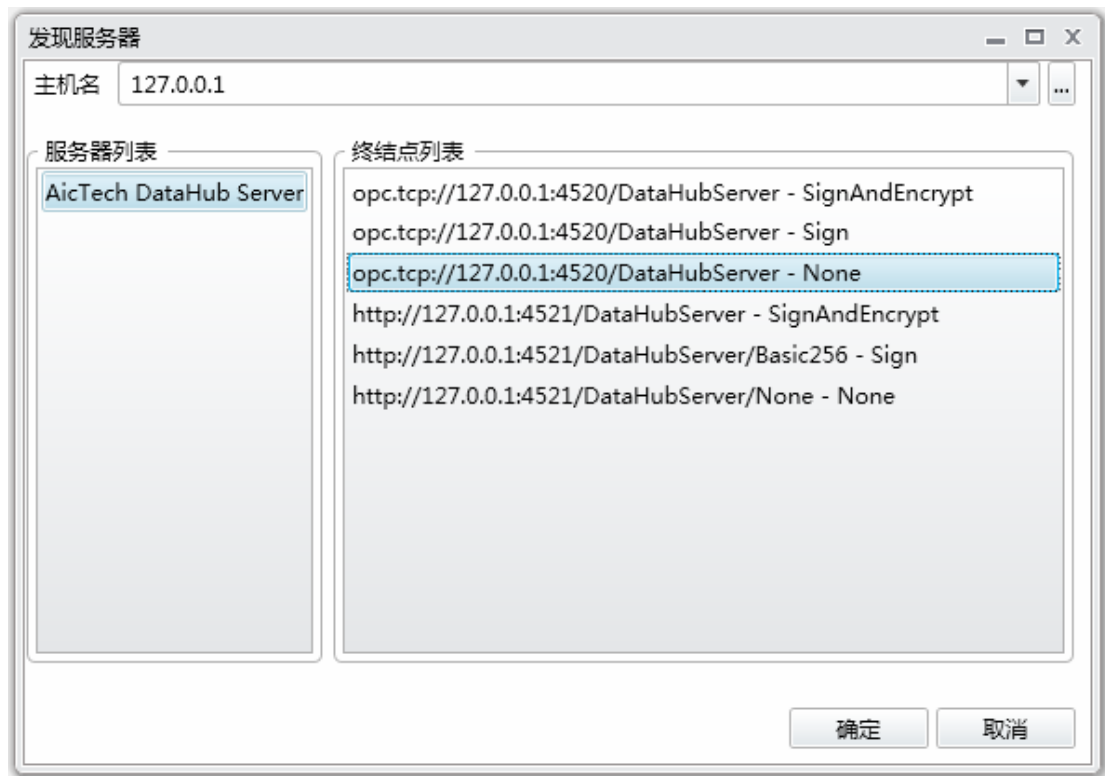


图 2-2 选择服务器地址

其中以“opc.tcp”开头的表示底层使用 TCP 通信协议，以“http”开头的表示使用的是 http 协议。

登录后，显示该实时库的内容。

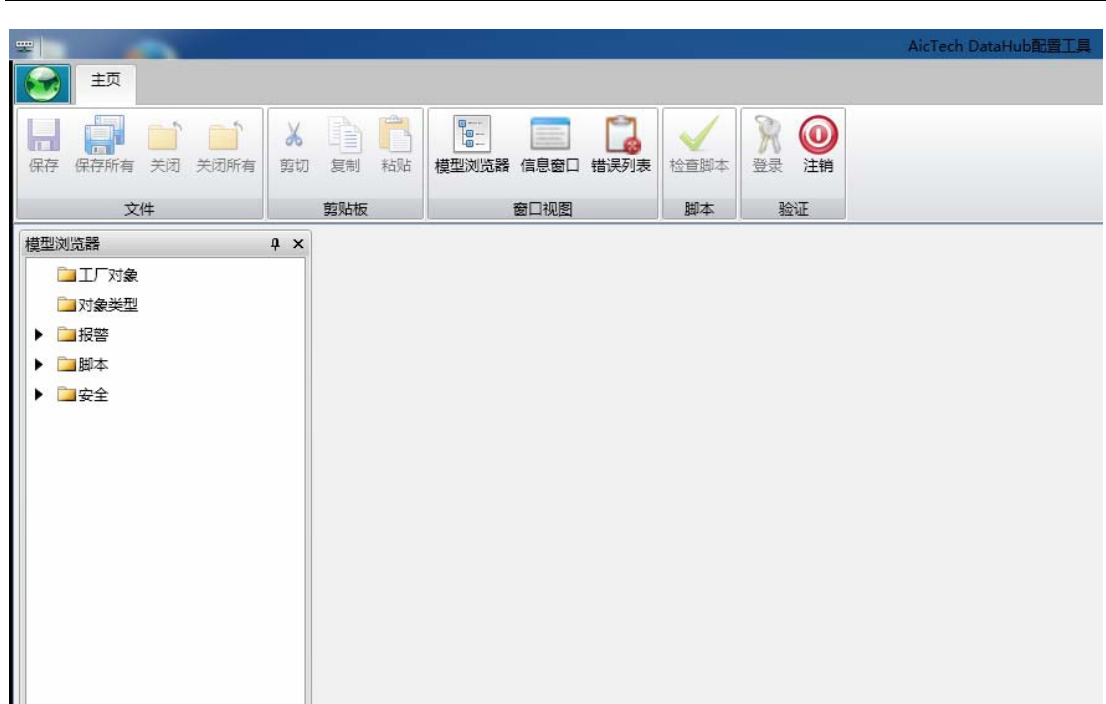


图 2-3 AicTech Datahub 配置工具

## 2.2.2 建立罐体的对象类型

AicDatahub 中的对象类型是指从多个相同或类似的实例中抽象出来的一个基础类型，这个模型具有这些实例所具有的所有共同特性。基于此类型可以构建无数个与此类型相同的对象实例。利用对象类型会极大地简化构建对象实例的过程。

假设需要集中监控 10 个罐体，为了确保罐体始终处于安全状态，需要时刻监测罐体的关键参数，主要包括：液位（Level）、温度（Temperature）和压力（Pressure）。

(1) 在 DataHub 的 ObjectTypes 中建立一个液位罐类型

右击“对象类型（ObjectTypes）”选择“新建对象类型（NewObject）”，然后弹出如下新建对象类型窗口，填入“浏览名称（BrowseName）”为“Tank”，“描述（Description）”为“罐体”，然后点击“确定（OK）”即可。如下图所示：

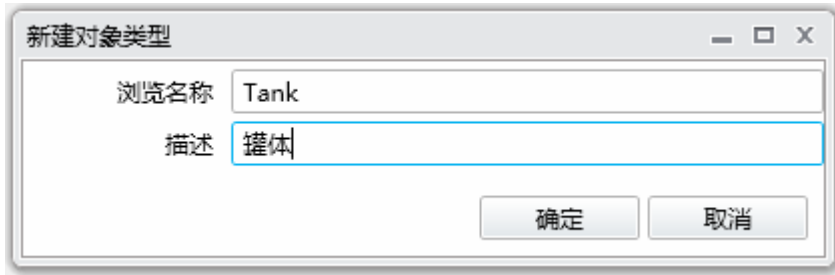


图 2-4 新建罐体对象类型

(2) 在新建的罐体对象模型中添加需要监控的参数

依次添加 Level(double 类型)、Temperature(double 类型)和 Pressure(double 类型)这三个监测参数。

右击“Tank”选择“新建变量(NewVariable)”弹出“新建数据变量(New Data Variable)”对话框，“浏览名称(BrowseName)”填写“Level”，“Description(描述)”填写“液位”，“数据类型(DataType)”选择“Double”，“数值等级(ValueRank)”选择“标量(Scalar)”，“类型定义(Type Definition)”选择“数据项目类型(DataItemType)”，“建模规则(Modelling Rule)”选择“强制(Mandatory)”，然后点击“OK”按钮即可，如下图所示：



图 2-5 新建变量

温度(Temperature)和压力(Pressure)建立与液位(Level)的建立过程完全类似。当这三个变量建立好之后对象类型中液位罐(Tank)模型的效果如下：

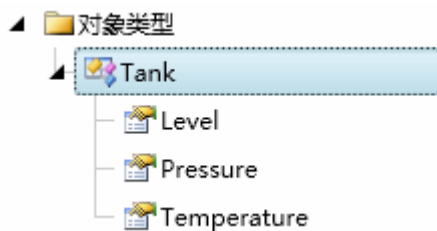


图 2-6 建立好的罐体模型

至此一个简单的罐体的对象模型已经建立完毕，可以基于此模型来创建三个液位罐的对象实例来分别表示工厂内的三个液位罐。

### 2.2.3 建立罐体的对象实例

在“工厂对象 (PlantObject)”上右击选择“新建对象 (New Object)”，弹出“新建对象 (New Object)”对话框，在“浏览名称 (Browse Name)”中填入“Tank1”，“描述 (Description)”中填入“罐 1”，“类型定义 (Type Definition)”中选择“Tank”，然后单击“确定”按钮即可。如下图所示：



图 2-7 创建液位罐实例对话框

建立好第一个罐的对象实例后对象实例文件夹中如下图所示：



图 2-8 建立好的第一个罐对象实例

建立好第一个液位罐对象实例之后其余的两个液位罐对象实例的建立都以类似的步骤来建立，建成之后如下图所示：



图 2-9 建立好的 3 个液位罐对象实例

至此罐体监控的对象模型和实例均已建立完成。

## 2.3 在AicIOServer中建立IO链接

AicIOServer 负责从现场设备中采集数据，并主动或被动传递给 AicDatahub。在本节中将建立为罐 1 建立 IO 链接。

### 2.3.1 新建IO工程

打开 AicIOServer 配置工具，点击工具栏中的新建工程选项，弹出新建工程对话框，如下图所示：

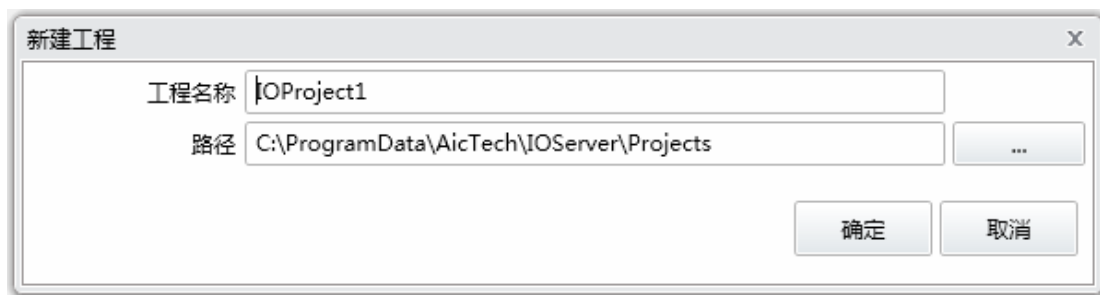


图 2-10 新建 IO 工程

工程名称就选默认的“IOProject1”，然后点击“确定(OK)”按钮出现 IOProject1 的编辑窗口，如下图所示：

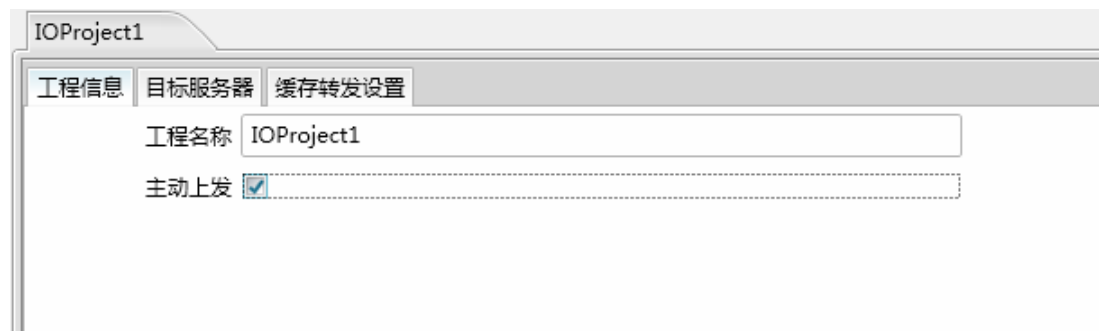


图 2-11 主动上发到实时库

勾选“主动上发 (Unsolicited Mode)”，然后切换到“目标服务器 (Target server)”，由于上面的 AicDatahub 是建立在本机上的，所以“服务器地址”选择



本机，“用户名”填入“sa”，“密码”填入“sa”，如下图所示：



图 2-12 实时库服务器地址

### 2.3.2 通道/设备/变量

然后右击“IOProject”选择“新建通道”，出现新建采集通道窗口，在“通道名称”中填入“Channel1”，“通道类型”选择“Simulation”，然后点击“确定（OK）”按钮即可，如下图所示：

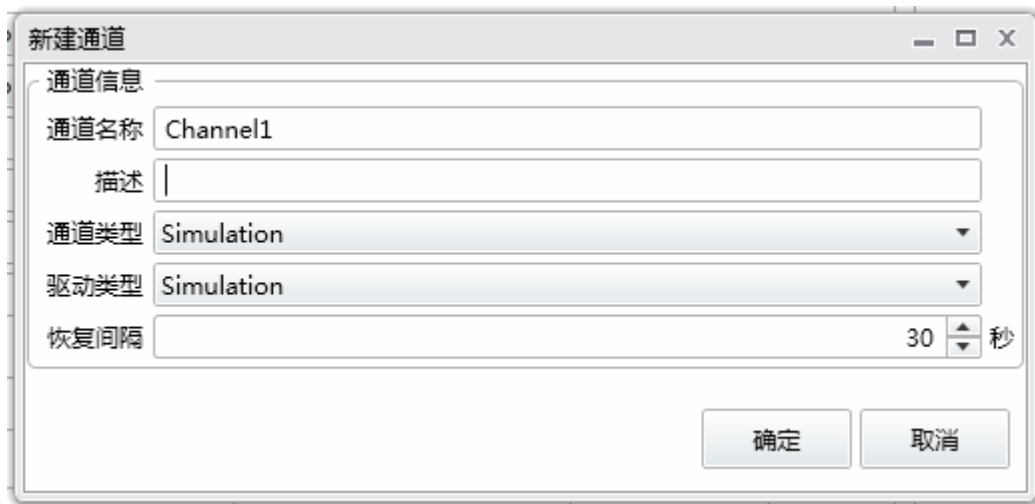


图 2-13 新建通道

之后在“Channel1”通道上右击选择“新建设备”即弹出新建设备窗口，在“设备名称”中输入“Tank1”，然后单击“确定（OK）”按钮即可。

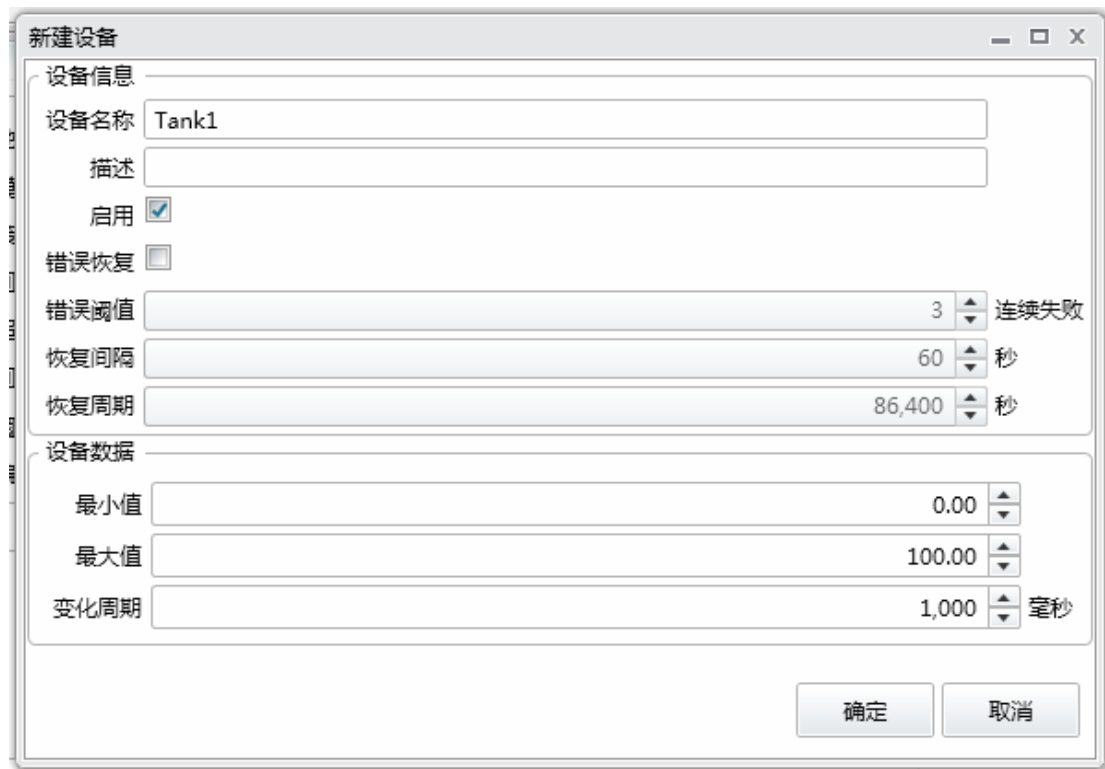


图 2-14 新建设备

建好“Tank1”设备之后在“Tank1”设备上右击选择“新建变量组”，弹出新建数据分类窗口，在“Group Name”中填入“Group1”，然后单击“确定（OK）”按钮即可。

建好“Group1”之后就可以从 AicDatahub 中将 Tank1 的所有监控参数一次性导入到 Group1 当中。右击“Group1”选择“从服务器导入”，弹出“选择项”窗口，依次展开“System”、“PlantObjects”，选择“Tank1”，然后单击“确定（OK）”按钮即可将 Tank1 中的所有监控变量导入到 Group1 当中，如下图所示：

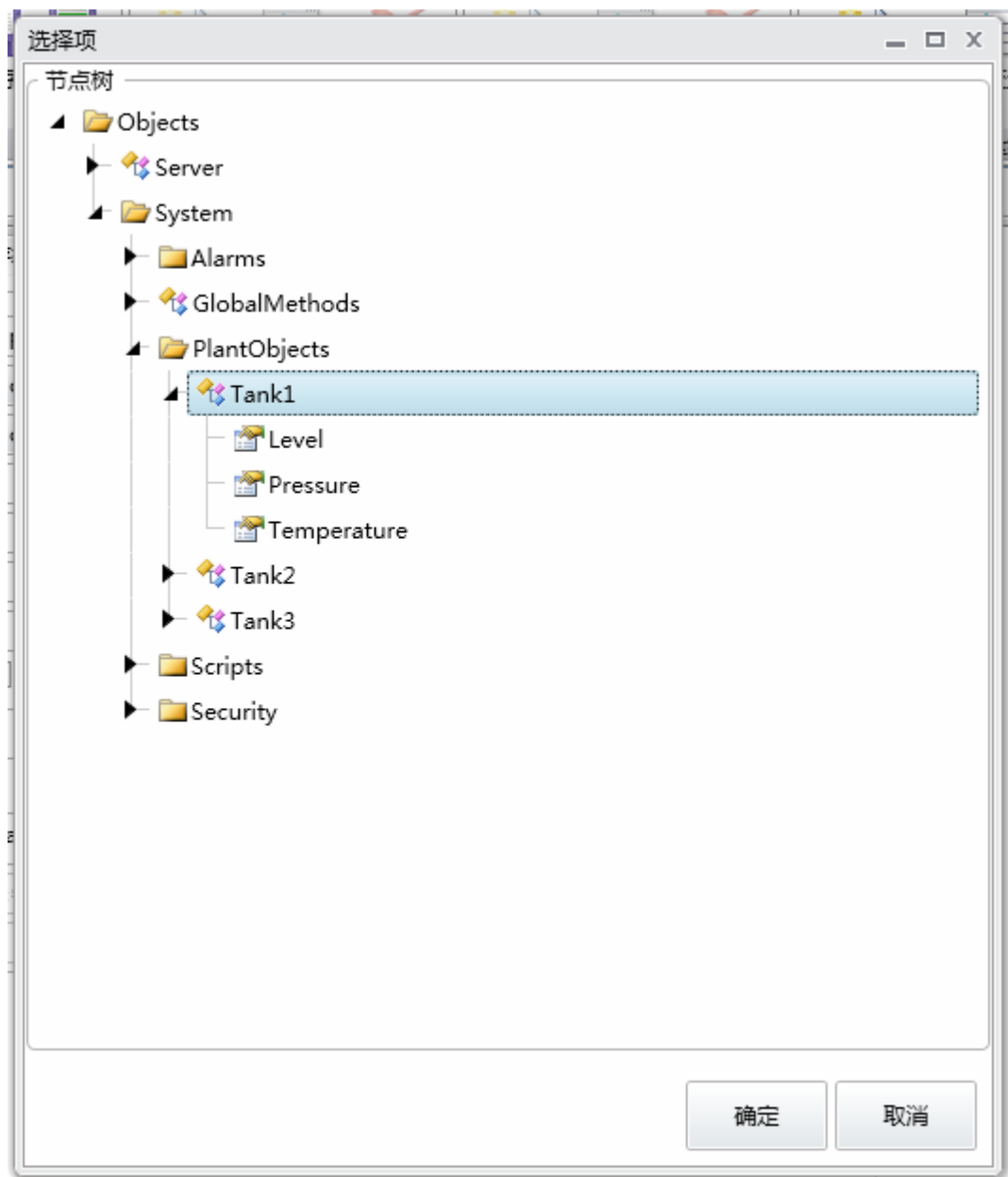


图 2-15 从实时库服务器导入变量

导入后 Group1 结构如下图所示：

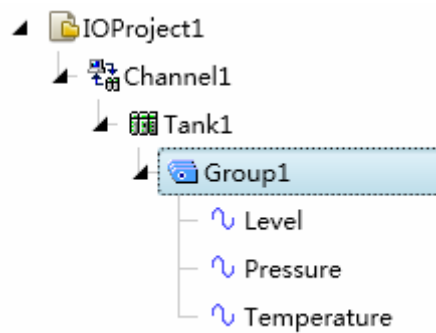


图 2-16 导入后的 IO 配置

导入的变量已经和实时库的变量关联。双击变量“Level”，弹出 IO 变量设置窗口，

变量信息	
变量名称	Level
描述	液位
激活	<input checked="" type="checkbox"/>
数据类型	Double <input type="checkbox"/> 数组
读/写	ReadOnly
采集方式	Unsolicited
数据变化触发器	StatusValue
死区	0
最大超时	0 毫秒
转换模式	None ...
采样间隔	1,000 毫秒
输出节点	ns=3;i=6 ...
输出索引范围	
输入节点	ns=3;i=6 ...
输入索引范围	

变量数据	
寄存器名称	Triangle
寄存器编号	1

图 2-17 变量配置

在第二个组合框“变量数据”中，把“寄存器名称”设置为三角波形式“Triangle”，这样变量的值会从最小值以  $1/s$  的变化频率增加到最大值之后再从最大值均匀的减小到最小值，并不断重复。其它变量的设置和“Level”类似。

同样的，添加设备 Tank2 和 Tank3，并分别设定通道对应各自的变量。

当设置完成之后需要右击“IOProject1”，选择“设为启动工程(本机)”，将其设置为启动工程，并且在服务中重启“AicTech IOServer”服务，此时第一个罐体的变量就开始从仿真 PLC 上采集数据。

## 2.4 使用AicStudio创建桌面监控画面

打开 AicStudio 新建一个 WPF 工程，取名为 TankMonitor，如下图所示：

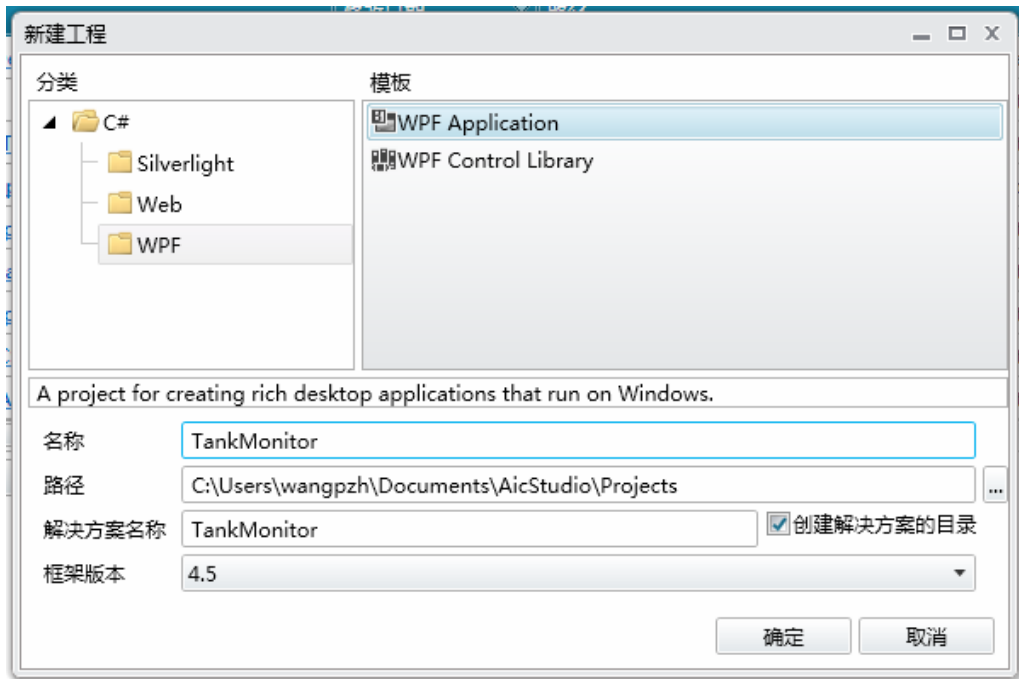


图 2-18 新建监控工程

TankMonitor 工程建好之后打开其中的 MainPage.xaml，然后在左侧的“工具箱→艾克信控”中拖入一个罐类控件“IndicatorTank”和两个仪表类控件“DualGauge”，分别命名为 tank1, gauge1, gauge2。并将两个 DualGauge 的 Caption 属性分别设置为“温度”和“压力”，其中 IndicatorTank 用于显示液体罐中的液位，两个 DualGauge 分别用于显示温度和压力。效果如下图所示：



图 2-19 监控主画面

为了让监控的液体罐的参数能在图上只管的显示出来，需要将液罐的“level”属性绑定到 AicDatahub 中的 Tank 中的变量“level”上，绑定方式有绝对绑定和相对绑定两种方式。

#### 2.4.1 绝对绑定

选中工程中的 IndicatorTank 控件之后在属性窗口中点击“Value”属性右侧的小白方块，弹出选择对话框，选择“创建数据绑定”，弹出“创建数据绑定”对话框，如下图所示：

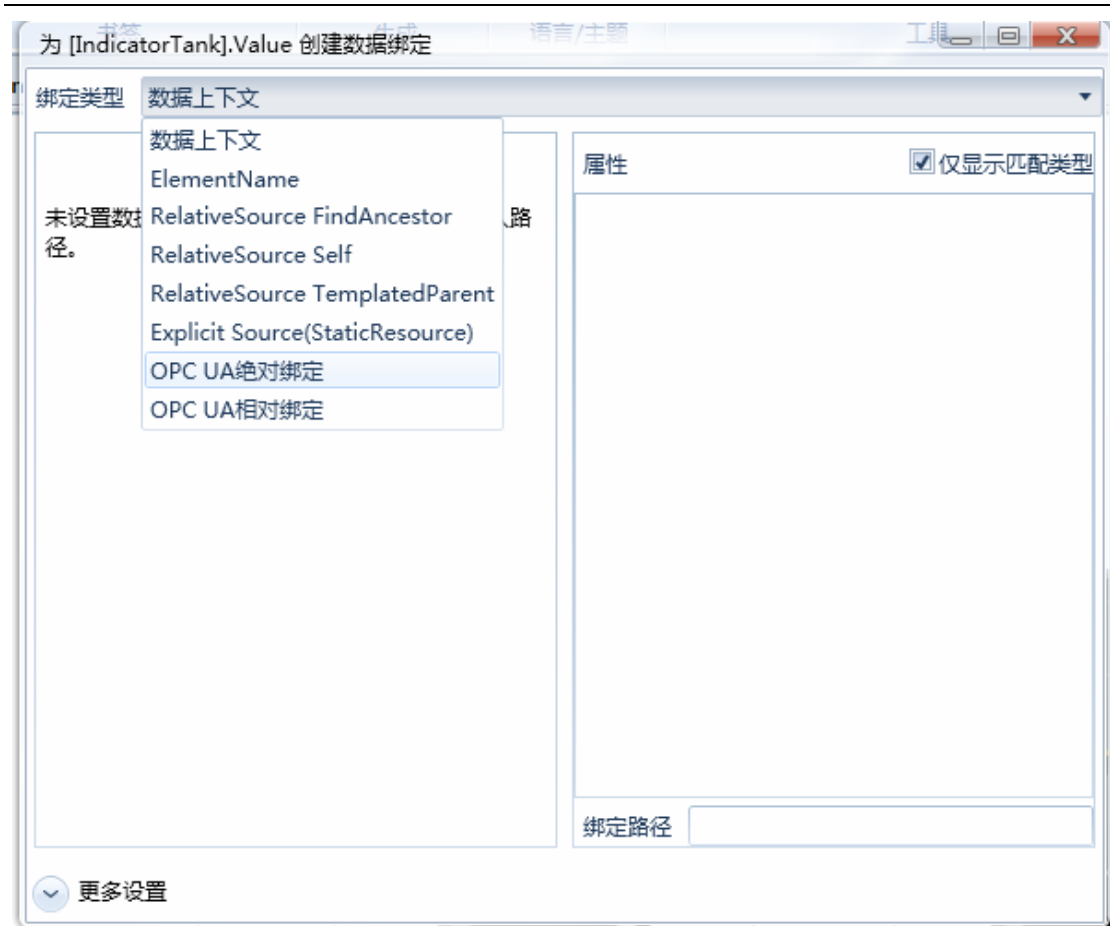


图 2-20 绝对绑定

对话框中“绑定类型”选择“OPC UA 绝对绑定”，在“连接名称”一栏选择“新建”按钮，弹出“OPC UA 连接配置”对话框，在此选择本机作为演示。配置结果如下图所示：

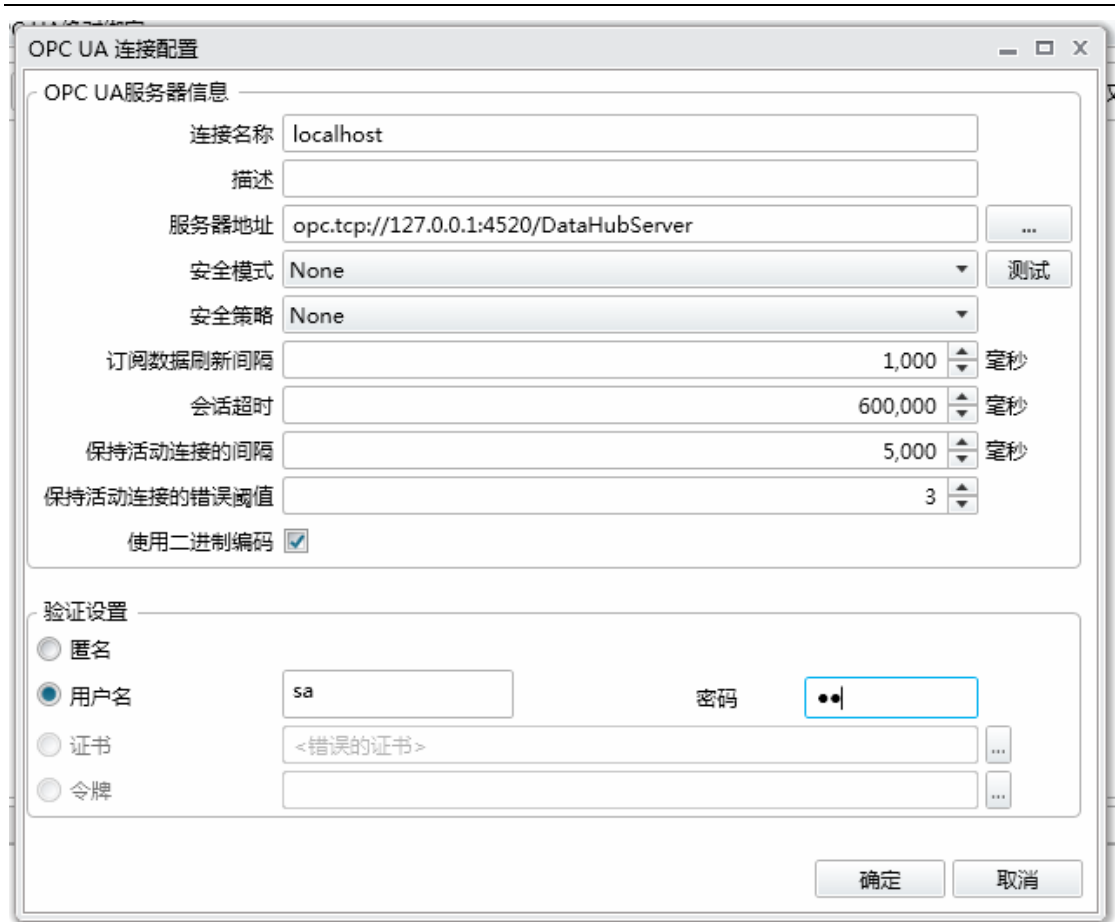


图 2-21 配置实时库服务器

配置好之后单击确定按钮，回到“创建数据绑定”对话框。然后在“连接名称”一栏中最左侧的下拉框中选择“本机”，之后点击中间的“刷新”按钮会展示出本机 AicDatahub 中的项目列表，依次展开列表之后选择 Tank1 中的 Level 变量，然后单击确定按钮即可。如下图所示：



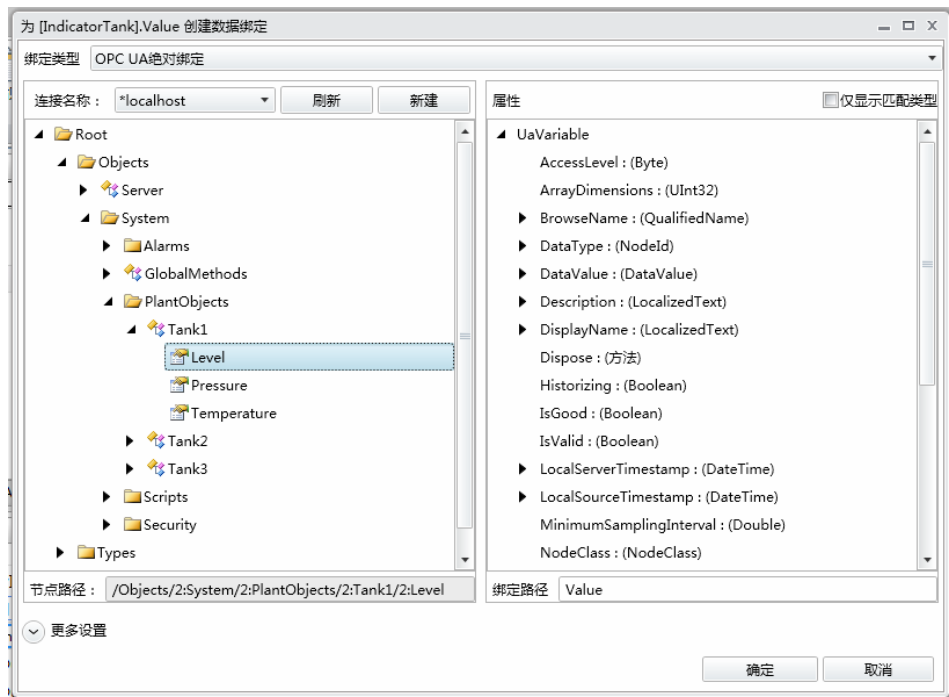


图 2-22 选择绑定变量

至此 Tank1 的 Level 变量也就绑定到了 IndicatorTank 控件的 Value 属性上，剩余“Pressure”和“Temperature”属性绑定到 DualGauge 控件的 Value 属性的过程完全类似。这样 Tank1 中三个变量的变化的情况就可以直观的在图形界面上显示出来。

配置好之后单击 AicStudio 中工具栏上的“生成解决方案”按钮，编译无错误之后点击“运行”按钮即可查看 Tank1 的监控画面。如下图所示：

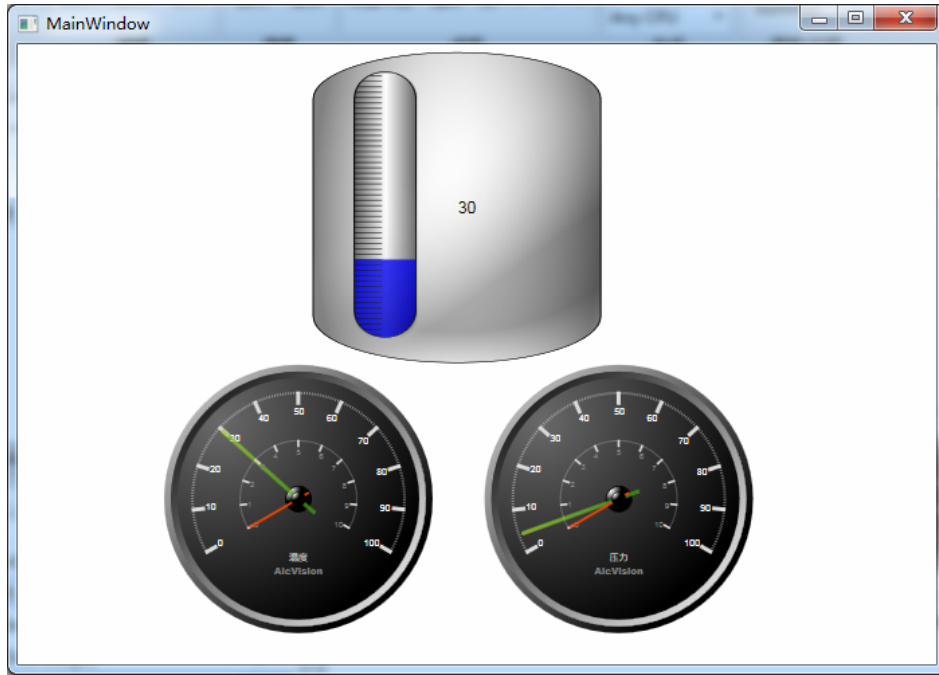


图 2-23 运行效果

### 2.4.2 相对绑定

相对绑定是指绑定到 AicDatahub 中某个类型，而不是某个具体的实例。相对绑定可以实现用同一套画面监视一类对象，从而减轻工作量。

本节中演示如何使用相对绑定实现在 Tank1、Tank2 和 Tank3 之间切换。

首先给 tank1, gauge1 和 gauge2 所在的容器 Grid 命名，比如 grid1，如下图：

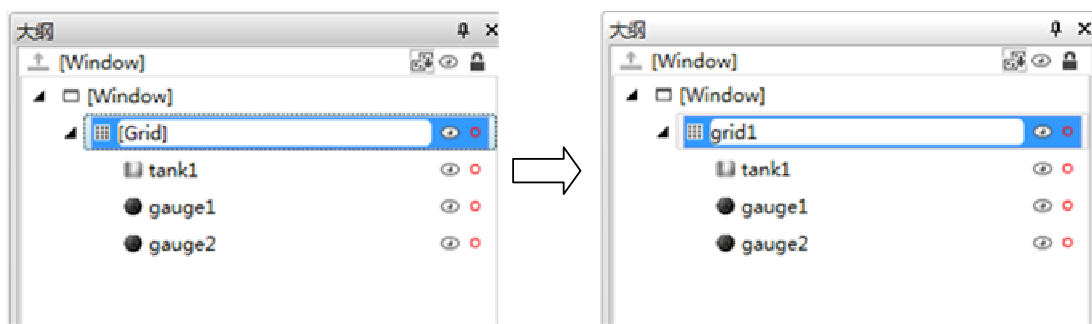


图 2-24 给所在容器控件命名

重新选中 tank1 点击其属性 “Value” 右侧的小方框，在弹出的 “创建数据绑定” 对话框中选择 “OPC UA 相对绑定”，然后点击 “设置 BindingContext”，目标元素选择 grid1，类型选择 Tank，如下图所示：

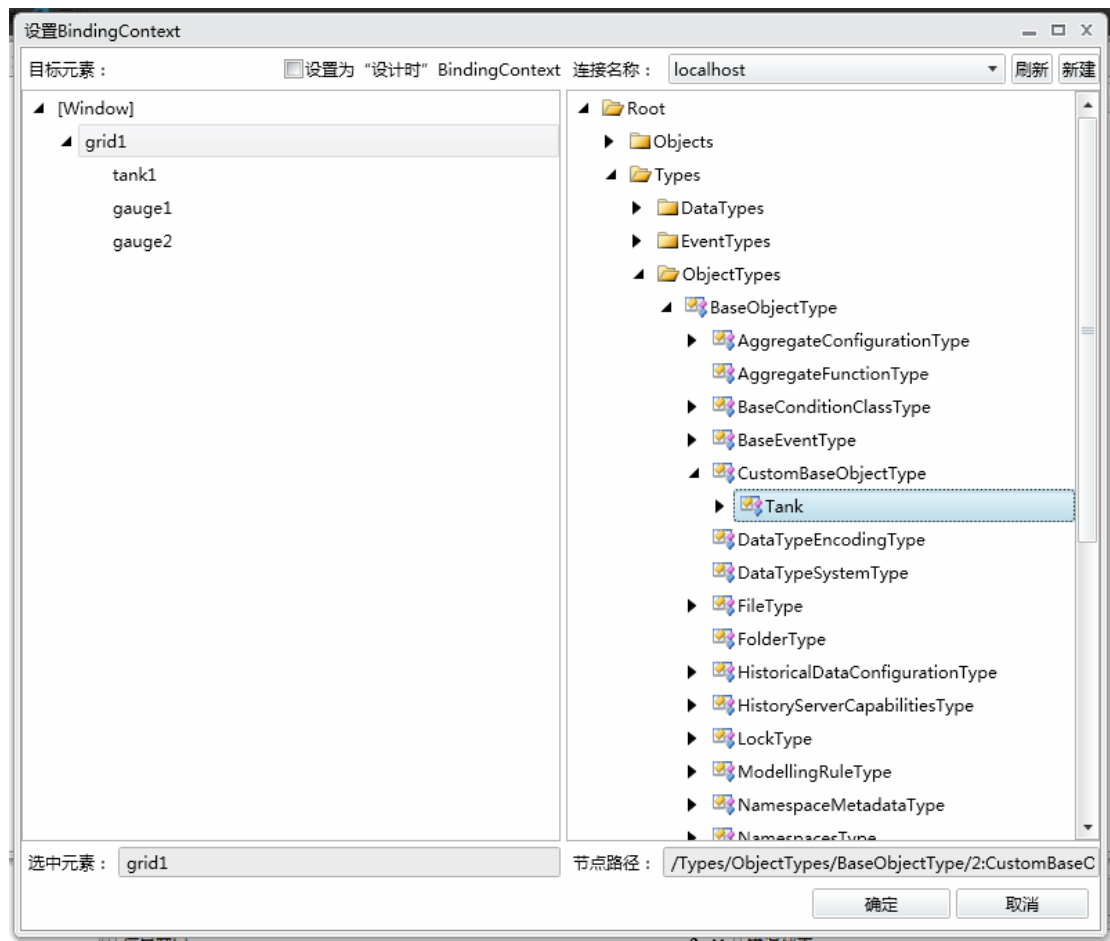


图 2-25 设置绑定上下文

点击“确定后”，在模型中选择 Tank/Level1，就把 tank1 控件的 value 相对绑定到了类型 tank 上：

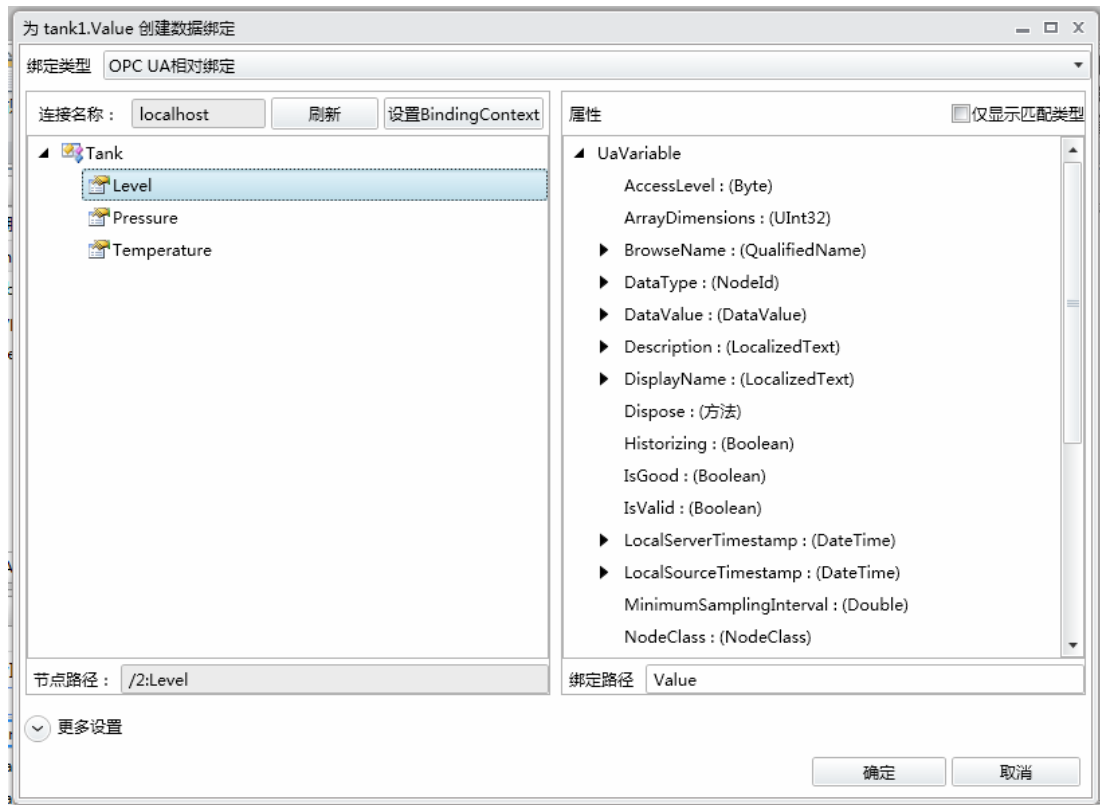


图 2-26 相对绑定到类型

类似地，把 gauge1 和 gauge2 的 value 分别相对绑定到 Tank/Temperature 和 Tank/Pressure 上。

### 2.4.3 利用脚本切换相对绑定对象

配置相对绑定后，从工具箱中往界面上添加三个按钮 button1, button2 和 button3，实现三个罐体对象之间的切换。



图 2-27 添加切换按钮

选择 button1，在属性窗，点击事件按钮，



图 2-28 给按钮添加事件

双击 Click，会在 MainWindow.xaml.cs 中添加 button1\_Click 事件函数，  
右键点击代码空行，选择插入代码片段/设置相对绑定对象，



图 2-29 脚本向导

在弹出的对象选择界面选择 Tank1

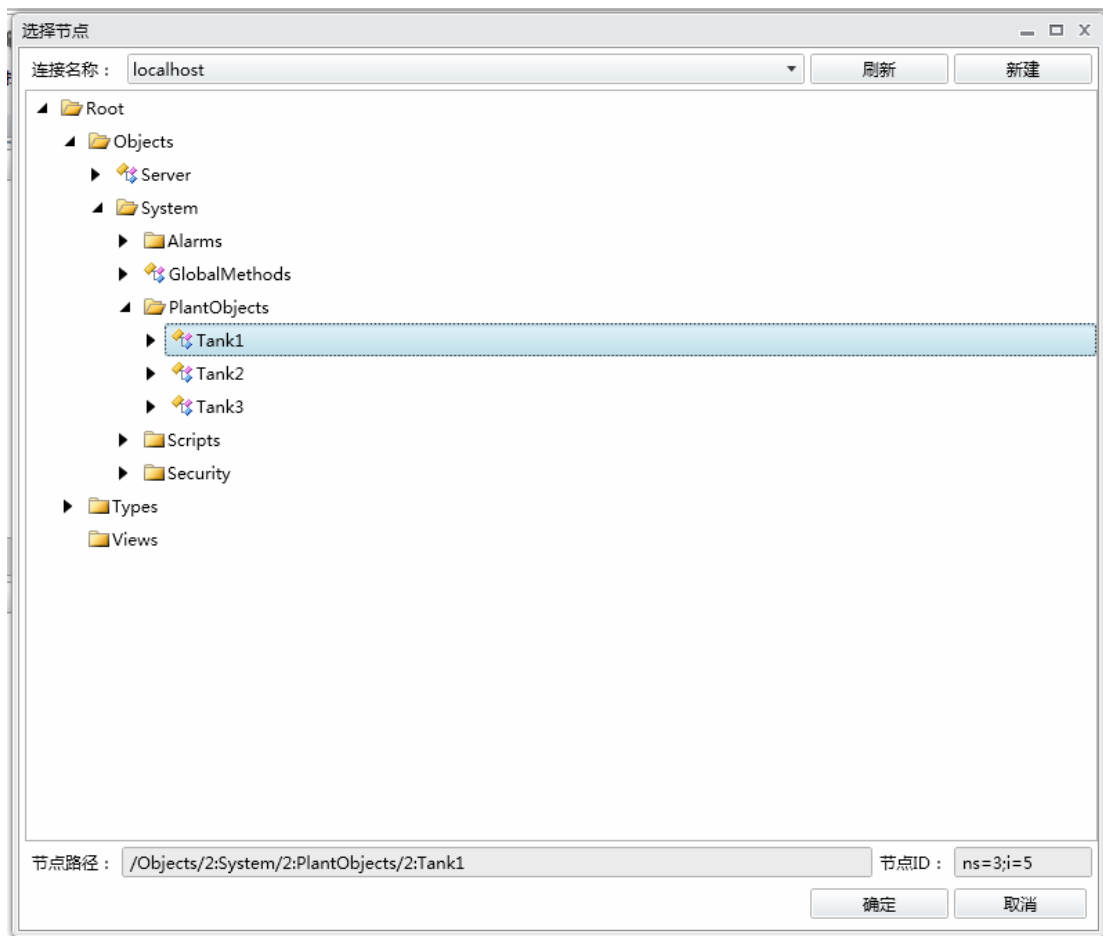


图 2-30 脚本向导选择对象实例

脚本向导就会插入如下代码：

```
33 private void button1_Click(object sender, System.Windows.RoutedEventArgs e)
34 {
35     // TODO: Add event handler implementation here.
36     var configuration = Application.Current.Resources["localhost"] as IUaConnectionConfiguration;
37     if (configuration == null || !configuration.IsValid) {
38         throw new ServiceResultException(StatusCodes.BadConfigurationError);
39     }
40
41     UaBindingManager.SetBindingContext(
42         this, // TODO: replace with binding target element (normally use root UI element)
43         new UaNodeReference(configuration, new UaNodePath("/Objects/2:System/2:PlantObjects/2:Tank1")));
44
45 }
46
```

图 2-31 脚本向导插入的代码

把 this 替换成 this.grid1，就把界面中的上下文切换到 Tank1，如下图所示：

示：

```
33 private void button1_Click(object sender, System.Windows.RoutedEventArgs e)
34 {
35     // TODO: Add event handler implementation here.
36     var configuration = Application.Current.Resources["localhost"] as IUaConnectionConfiguration;
37     if (configuration == null || !configuration.IsValid) {
38         throw new ServiceResultException(StatusCodes.BadConfigurationError);
39     }
40
41     UaBindingManager.SetBindingContext(
42         this.grid1, // TODO: replace with binding target element (normally use root UI element)
43         new UaNodeReference(configuration, new UaNodePath("/Objects/2:System/2:PlantObjects/2:Tank1")));
44
45 }
46
47
```

图 2-32 修改绑定上下文对应容器

类似地，完成 button2 和 button3 的代码，实现对 Tank2 和 Tank3 的相对绑定切换。

至此，一个相对绑定的示例就已构建完成，当工程运行的时候，在没点击按钮之前罐和仪表是没有任何显示的。当点击 Tank1 的切换按钮之后，将参考节点的路径设置为对象实例的 Tank1，点击“切换到罐 2”和“切换到罐 3”将显示 Tank2 和 Tank3 数据。

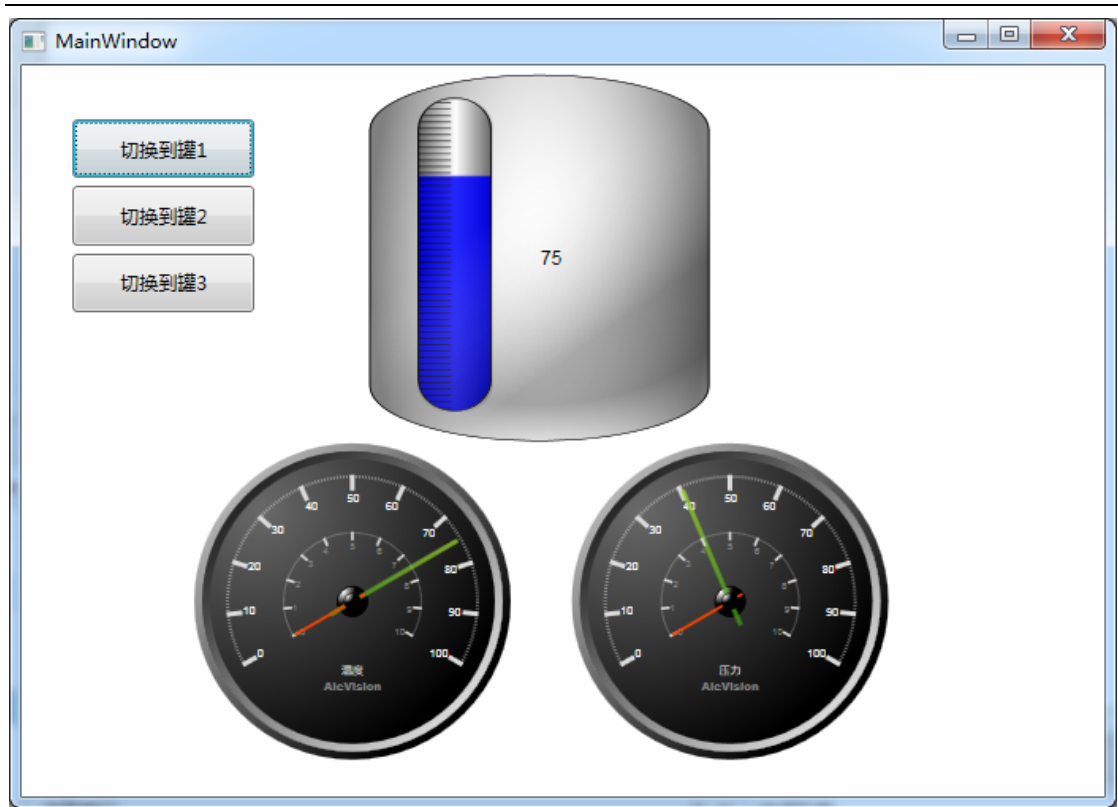


图 2-33 相对绑定运行结果

## 2.5 使用AicStudio创建Web监控画面

使用 AicStudio 创建 Web 工程和桌面工程几乎相同, 首先打开 AicStudio 之后新建一个 Silverlight 工程, 取名为 TankMonitorWeb, 如下图所示:



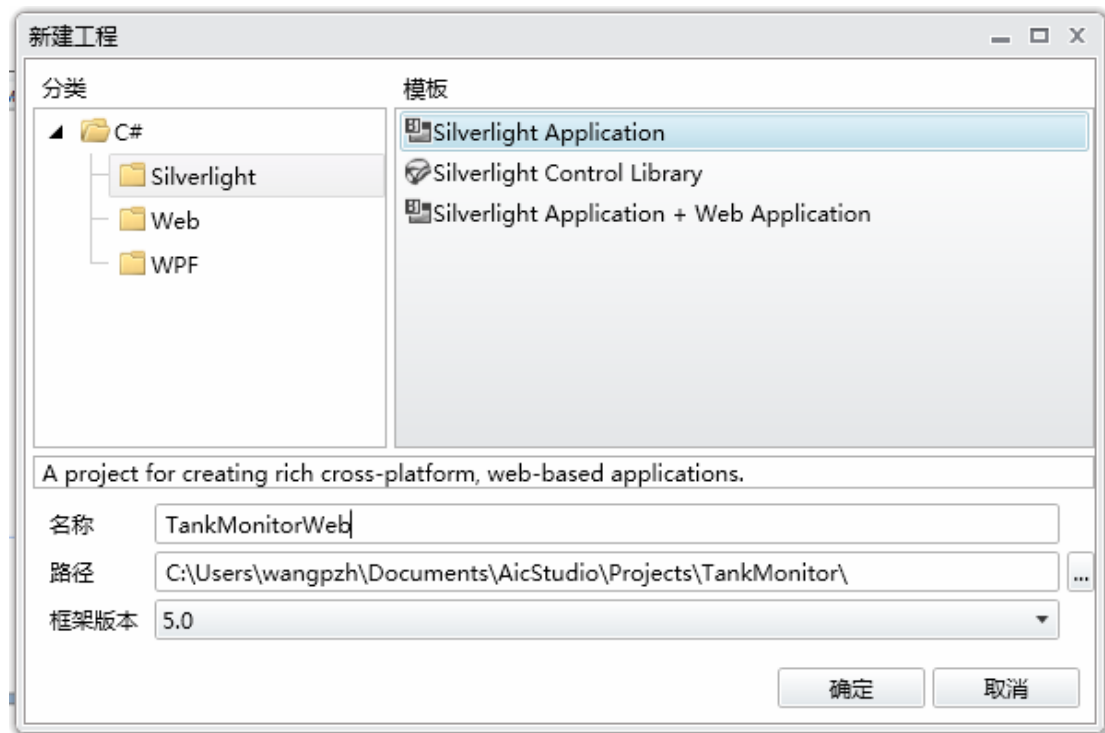


图 2-34 创建 Sliverlight 工程

工程建好之后打开其中的 MainPage.xaml，和前面例子相同，添加 tank1，gauge1 和 gauge2，创建绝对绑定，编译后，点击“运行”，弹出基于浏览器的监控界面：

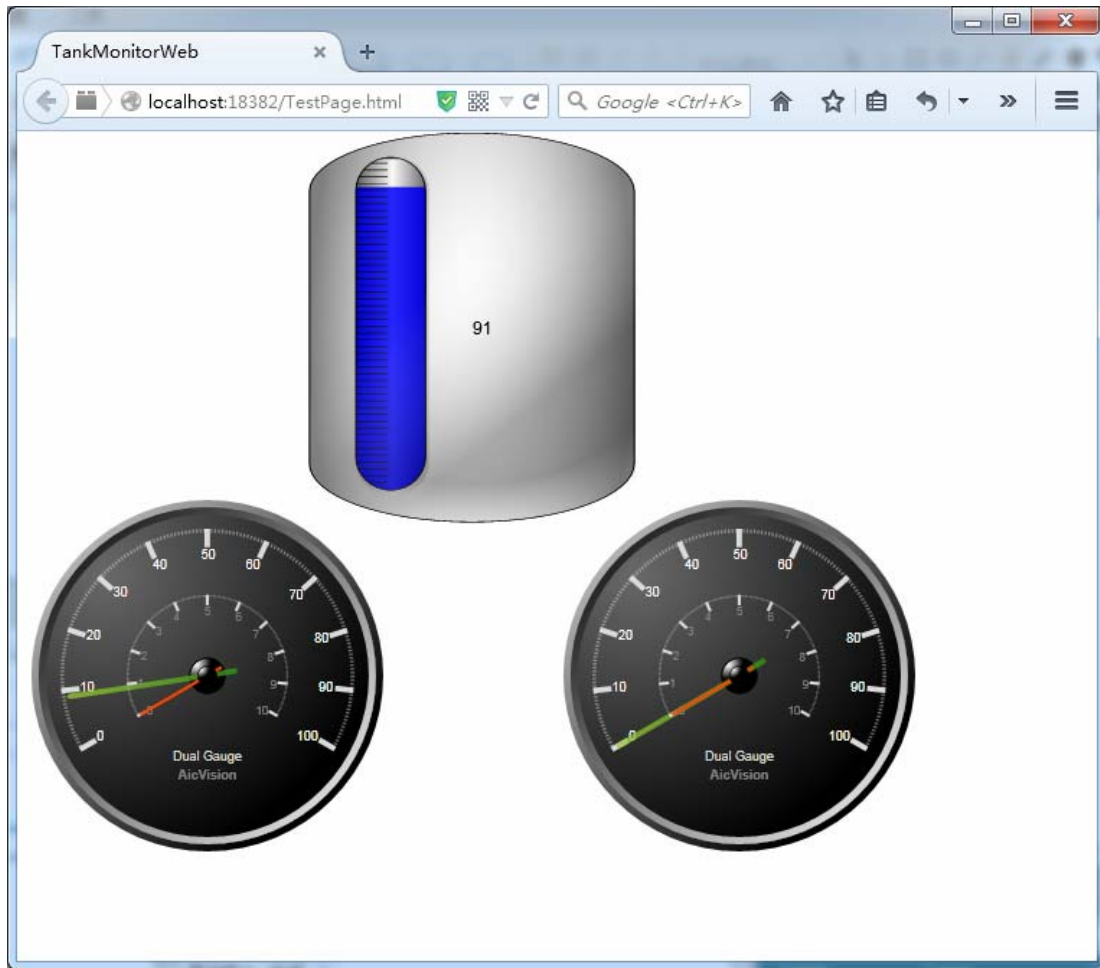


图 2-35 Silverlight 运行结果

正式使用时，需要把 Web 工程发布到 Web 服务器上，具体步骤请参考艾克信控的相关文档。

## 2.6 在 AicStudio 中创建一个动画

AicStudio 支持丰富的动画功能，本节将创建一个旋转动画，并和 Tank1 的 Level 变量的值绑定。

打开前面建立的 TankMonitor 工程，在“解决方案浏览器”中选中 TankMonitor，

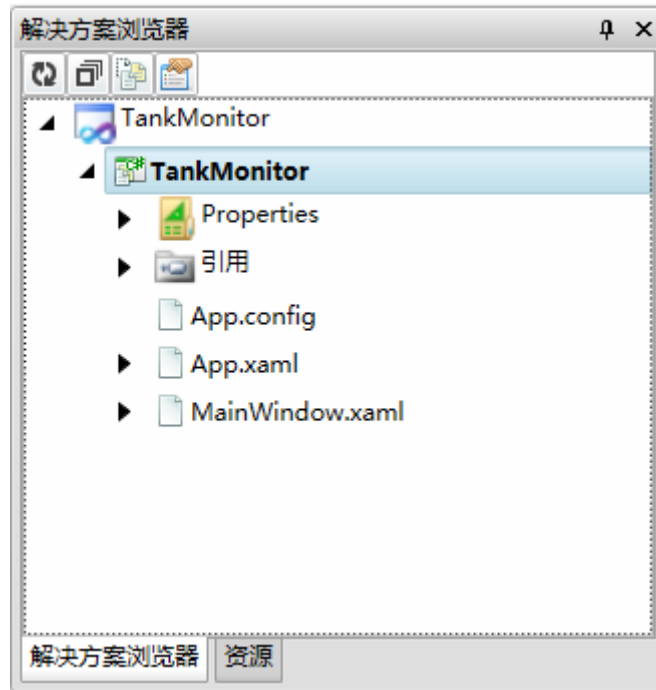


图 2-36 解决方案浏览器

右键点击并选择“新建项”，弹出新建项对话框，选择 WPF，使用 Window 模板，建立一个名字为“Window1.xaml”的窗体，如下图所示：

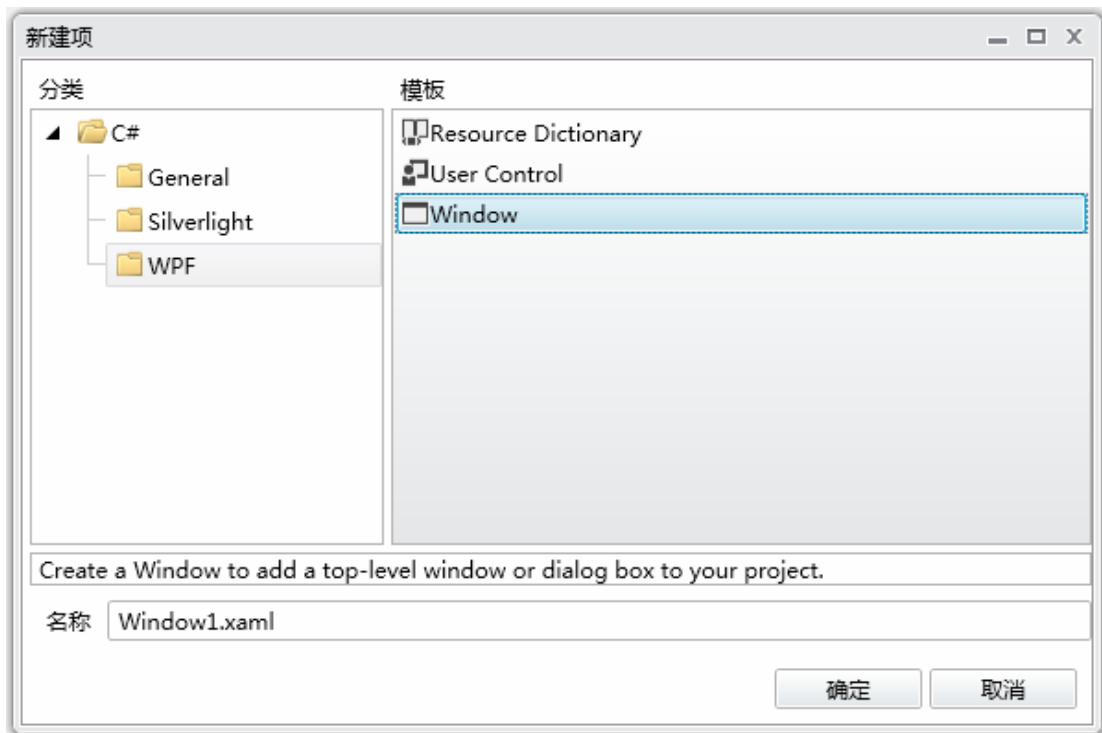


图 2-37 添加 WPF 的 Window

新建窗体将在解决方案浏览器中增加两个文件：Window1.xaml 和 Window1.xaml.cs，前者是界面描述文件（依据微软的 xaml 文件标准），后者是

C#脚本文件。

从工具箱的通用控件中拖入一个矩形控件（这个矩形控件不太清楚），命名为“rect1”，并将其背景属性设置为红色。然后从工具箱中“行为”选项的子目录中选中“LinearRotateAction”，

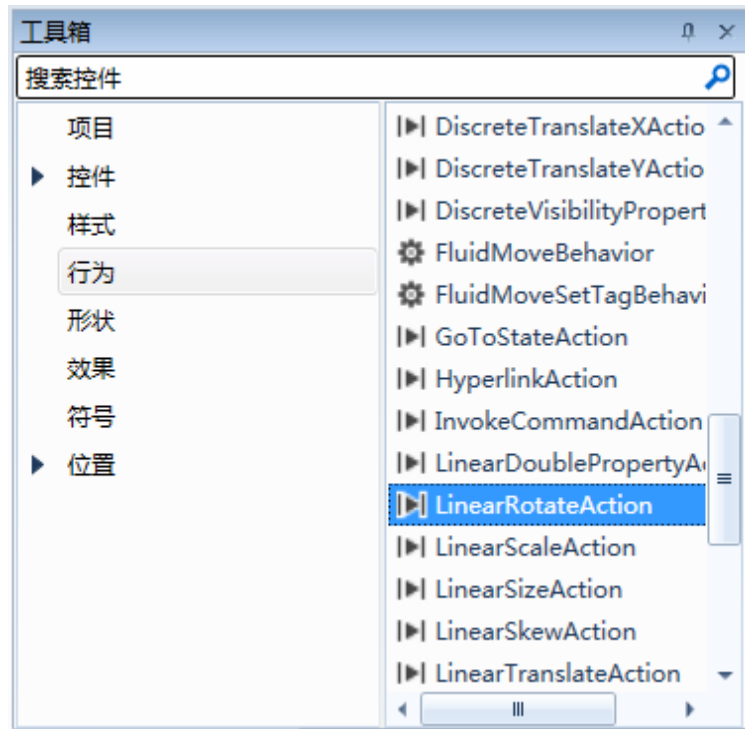


图 2-38 选择行为

鼠标左键选中之后拖动到设计框中的矩形上，完成后在大纲的树形目录中“rectangle1”的下面会出现“LinearRotateAction”项。如下图所示：

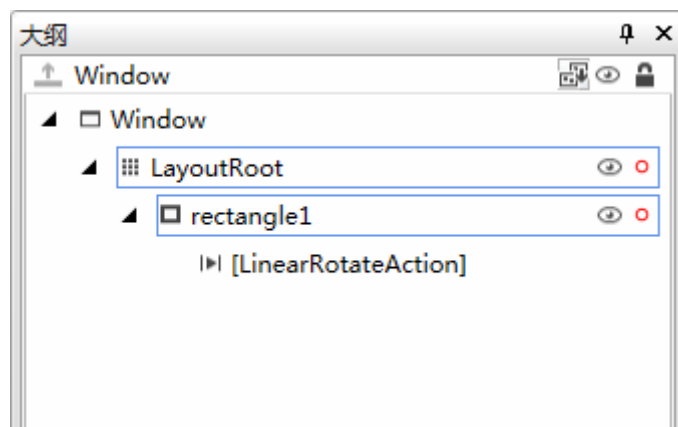


图 2-39 在大纲中查看添加的行为

选中树形目录中 rectangle1 下的“LinearRotateAction”，界面右侧会出现其属性窗口，动画的属性窗口共分为四部分，如下图所示：

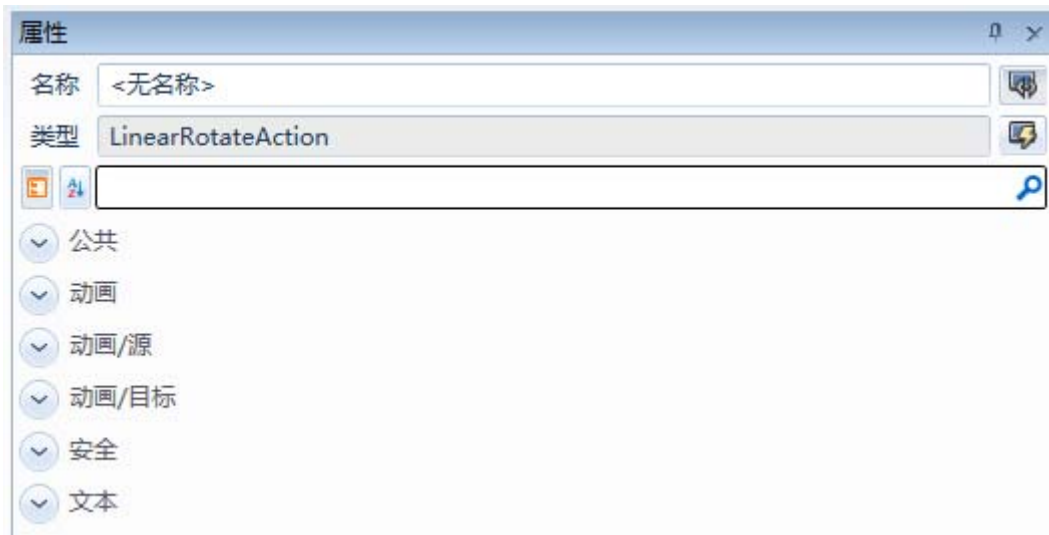


图 2-40 行为的属性

其中“公共”、“动画”、“安全”和“文本”这四个选项选择默认设置就可以。展开“动画/源”，其中的“MaxValue”设置为 100（因为在 IOserver 中设置的 Level 的模拟采集最大值为 100），“MinValue”设置为 0（因为在 IOserver 中设置的 Level 的模拟采集最小值为 0）。然后单击 Value 右侧的白色方框，在弹出的窗口中选择“创建数据绑定”，类似上一节中讲到的绑定过程一样将 Value 绑定到本机的 AicDatahub 中实例 Tank1 的变量 Level 上。

展开“动画/目标”，其中“AngleMax”设置为 90，“AngleMin”设置为 0，表示该矩形实际旋转的角度范围是 0 度到 90 度。默认的旋转中心为 (0,0)，即矩形的左上角。

然后在 MainWindow 中建立一个按钮 button4，切换到 Window1 窗体。

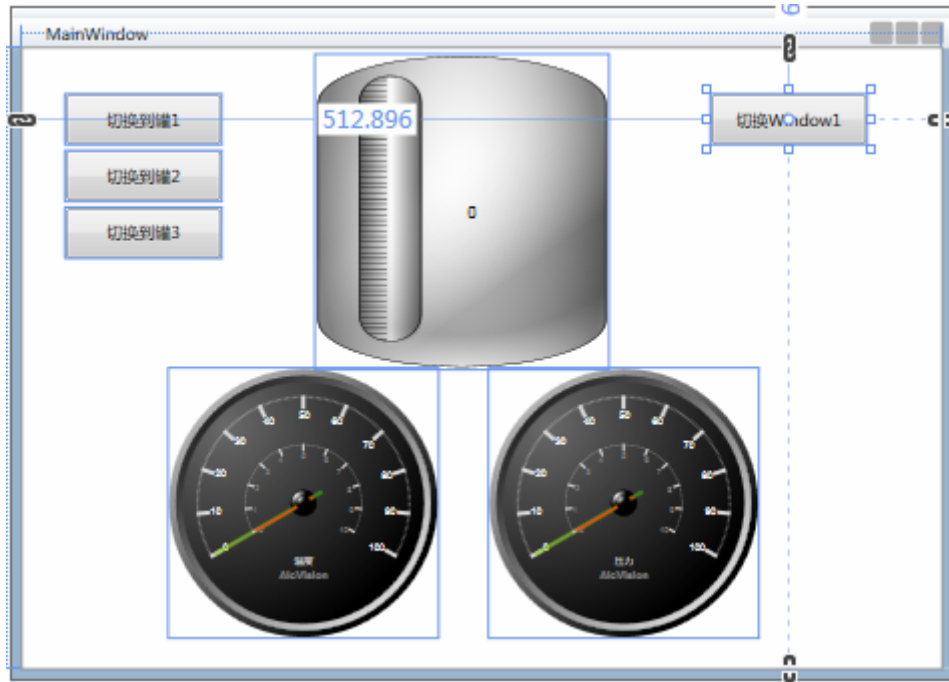


图 2-41 添加切换画面按钮

在对应的脚本中，添加跳转脚本如下：

```
private void button4_Click(object sender, System.Windows.RoutedEventArgs e)
{
    // TODO: Add event handler implementation here.
    var window = new Window1();
    window.Show();
}
```

图 2-42 在按钮脚本中实现画面切换

编译后，点击“运行”，在主窗体中，点击“切换 Window1”按钮，弹出 Window1，即可看到该矩形跟随 Tank1 的 Level 变量的旋转变化情况。如下图所示：

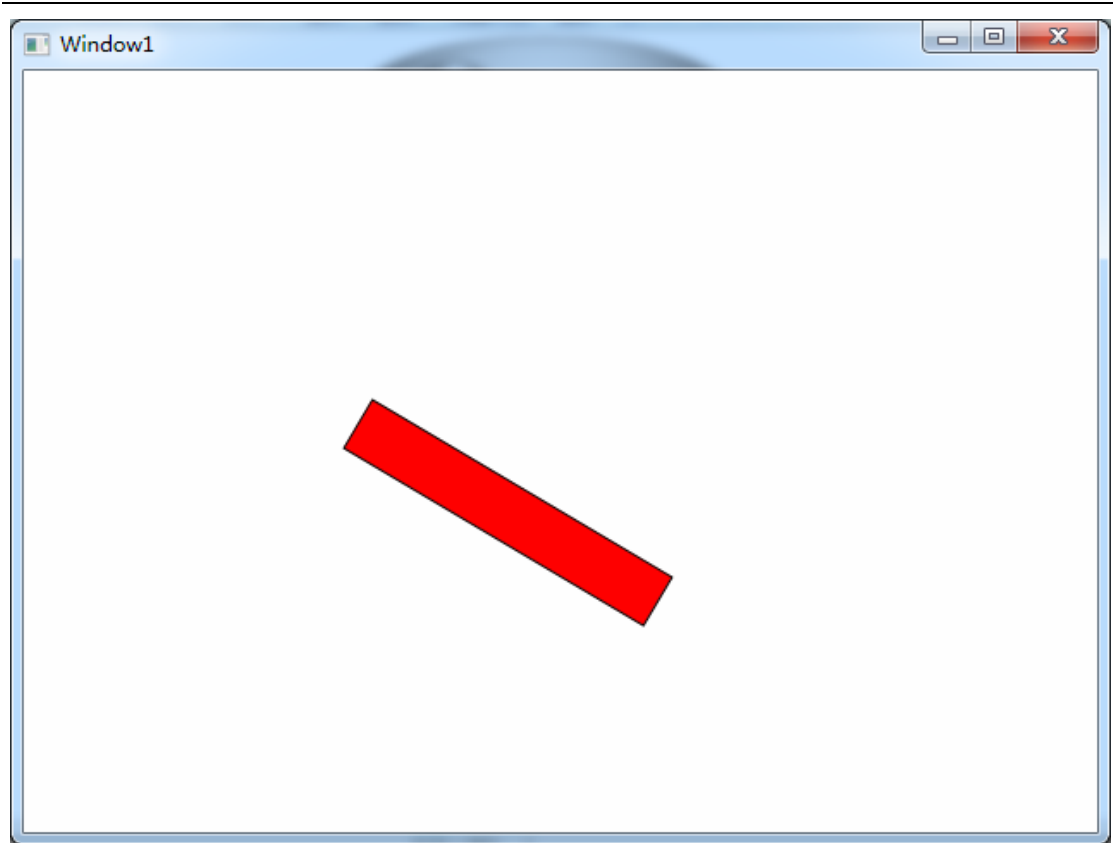


图 2-43 动画效果

## 2.7 为Tank建立液位报警

本节中将通过示例简述在类型中增加报警，并应用到各个对象实例的过程。

### 2.7.1 报警和报警区

回到 AidDatahub 的配置工具中，在“对象类型”中右击类型 Tank，选择“新建报警”，弹出新建报警窗口，在报警窗口“浏览名称”中填写“LevelAlarm”，“描述”中填入“液位报警”，“类型定义”选择“ExclusiveLevelAlarmType（单一水平报警）”，其余项默认即可，配置完后如下图所示：

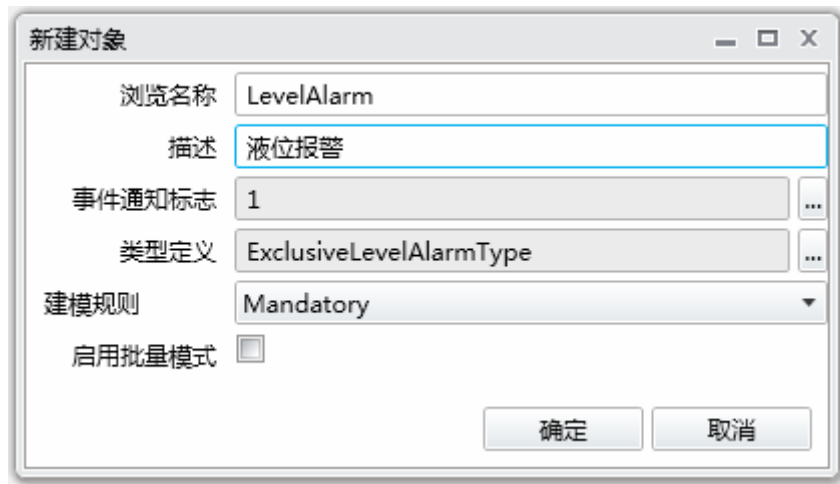


图 2-44 新建报警对象

配置好之后点击“确定（OK）”按钮即打开新建报警的配置窗口，点击源对应的选择按钮，在“相对节点”中将源设置为 Level 变量，如下图所示：

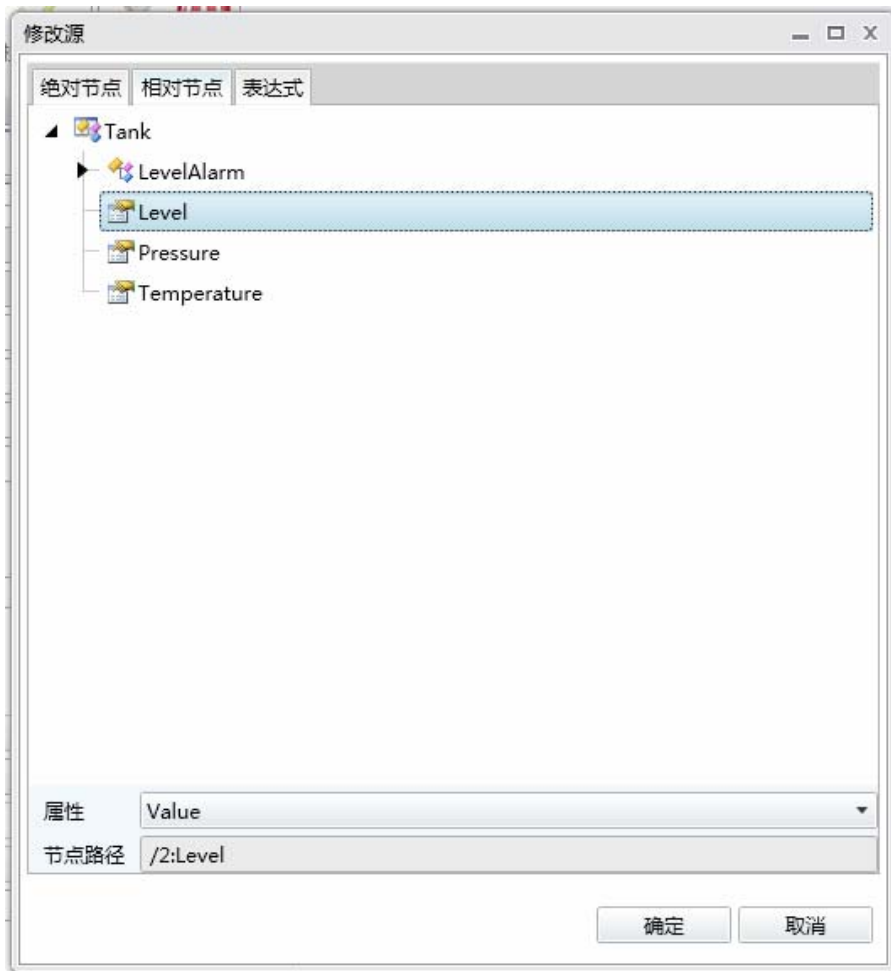


图 2-45 修改报警源

然后，设置报警接触消息为“液位恢复正常”，启用报警以及 4 个报警范围



高高、高、低和低低。并设置高高，高，低，低低的限值分别为：90，80，20，10。

报警信息

名称	LevelAlarm	
源	/2:Level	...
报警激活延时	0	毫秒
报警解除延时	0	毫秒
报警解除消息		...
死区	0.00	
启用	<input checked="" type="checkbox"/>	
报警解除要求确认	<input type="checkbox"/>	

高高

启用	<input checked="" type="checkbox"/>	
要求确认	<input type="checkbox"/>	
比较运算符	>	
限值	90.00	
报警激活消息	液位极高	...
严重性	高	

高	
启用	<input checked="" type="checkbox"/>
要求确认	<input type="checkbox"/>
比较运算符	>
限值	80.00
报警激活消息	液位高
严重性	中等

低	
启用	<input checked="" type="checkbox"/>
要求确认	<input type="checkbox"/>
比较运算符	<
限值	20.00
报警激活消息	液位低
严重性	中等

低低	
启用	<input checked="" type="checkbox"/>
要求确认	<input type="checkbox"/>
比较运算符	<
限值	10.00
报警激活消息	液位极低
严重性	高

图 2-46 报警限值设定

类型配置后，在“工厂对象”选中 Tank1，右键点击“应用对象类型更改到对象实例”，



图 2-47 应用对象类型更改到对象实例

Tank1 中就会出现对象类型配置过的 LevelAlarm 报警项，



图 2-48 应用后的结果

类似地，应用到 Tank2 和 Tank3 对象。

配置报警后，需要把相应的对象添加到报警区中，本例中把 Tank1，Tank2，Tank3 添加到默认报警区 RootArea 下，如下图所示：（此处需不需要说明添加过程）

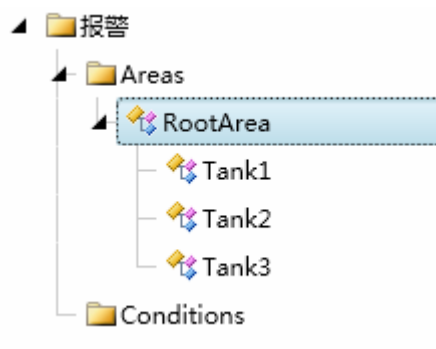


图 2-49 把对象添加到默认报警区

### 2.7.2 实时报警显示

打开 AicStudio 的 TankMonitor 工程，在“解决方案浏览器”中增加 WPF 的一个新窗体 Window2，向界面中拖入一个“艾克信控”的“LiveAlarmControl”控件，命名为 LiveAlarmControl1。然后将其“AreaObjectNode”属性绝对绑定到 RootArea，如下图所示：

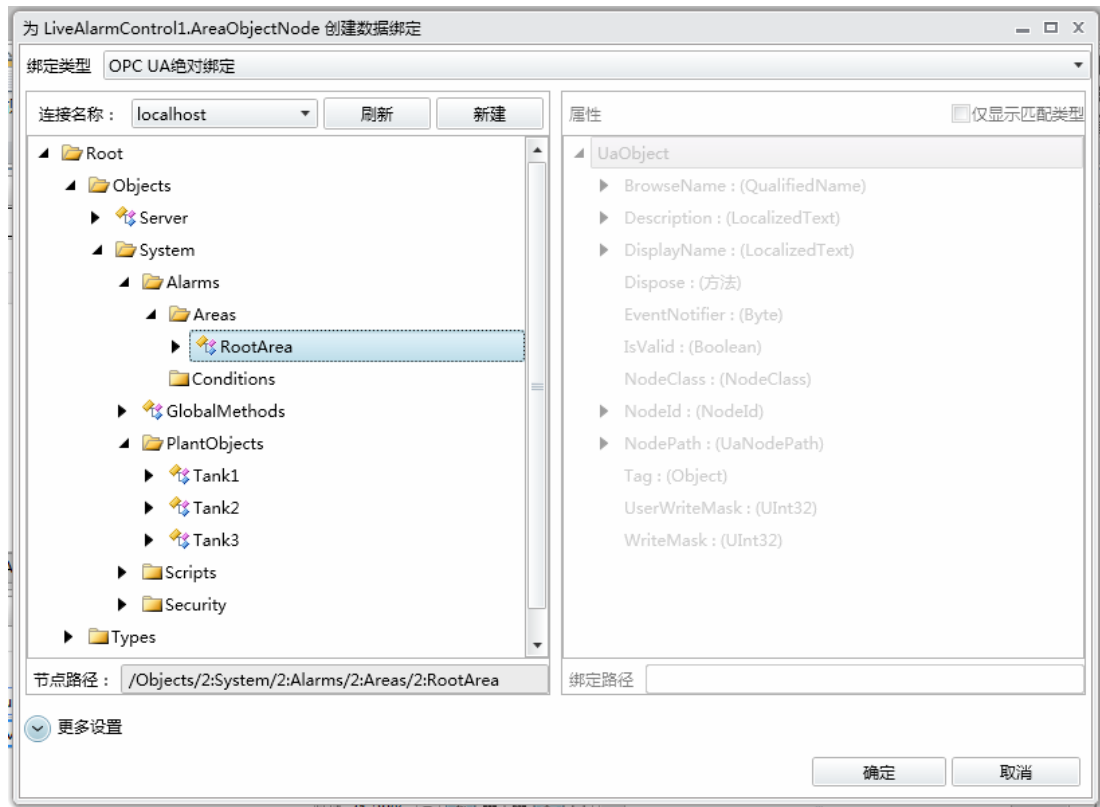


图 2-50 报警控件绑定到报警区

这样该控件就可以实时监视 3 个罐的液位报警了。在 MainWindow 中添加一个按钮 button5，内容为“切换到 Window2”，运行并切换到 Window2，如下图所示：

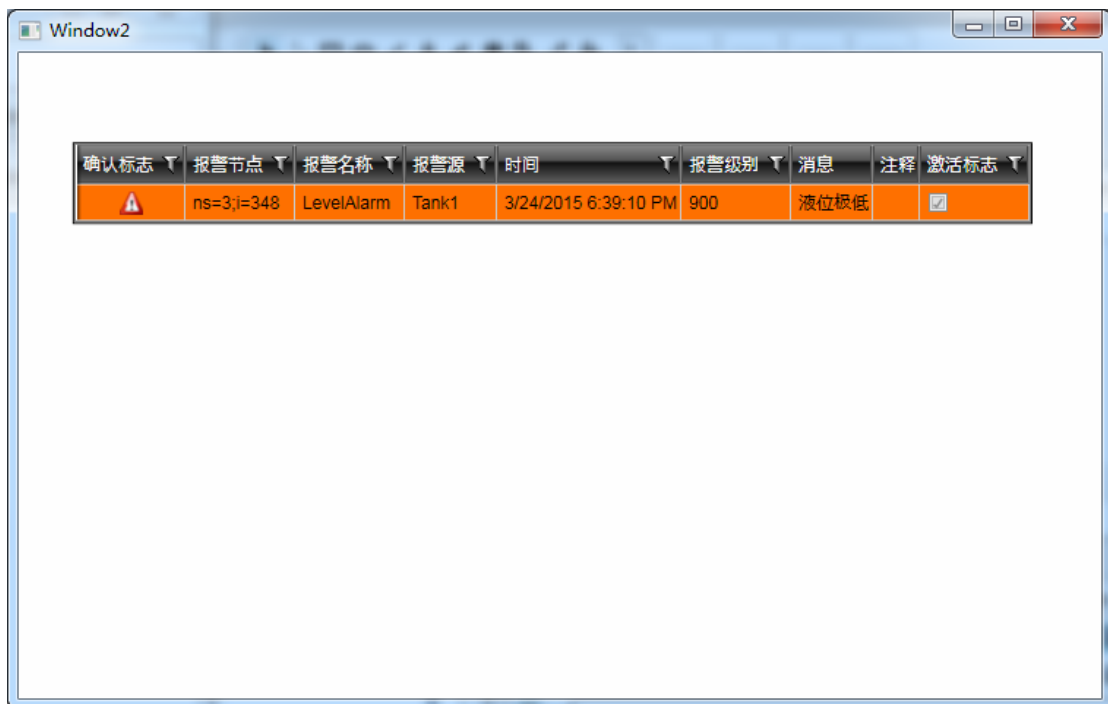


图 2-51 报警运行结果

### 2.7.3 历史报警显示

在刚才建立的 Window2 中，从工具中添加 HistoryAlarmControl，命名为“HistoryAlarmControl1”。点击 DataConnection 属性右侧的小白块，选择“本地资源”，选择“localhost”，就把此控件的报警源关联到了本地的实时库，运行效果如下图所示：

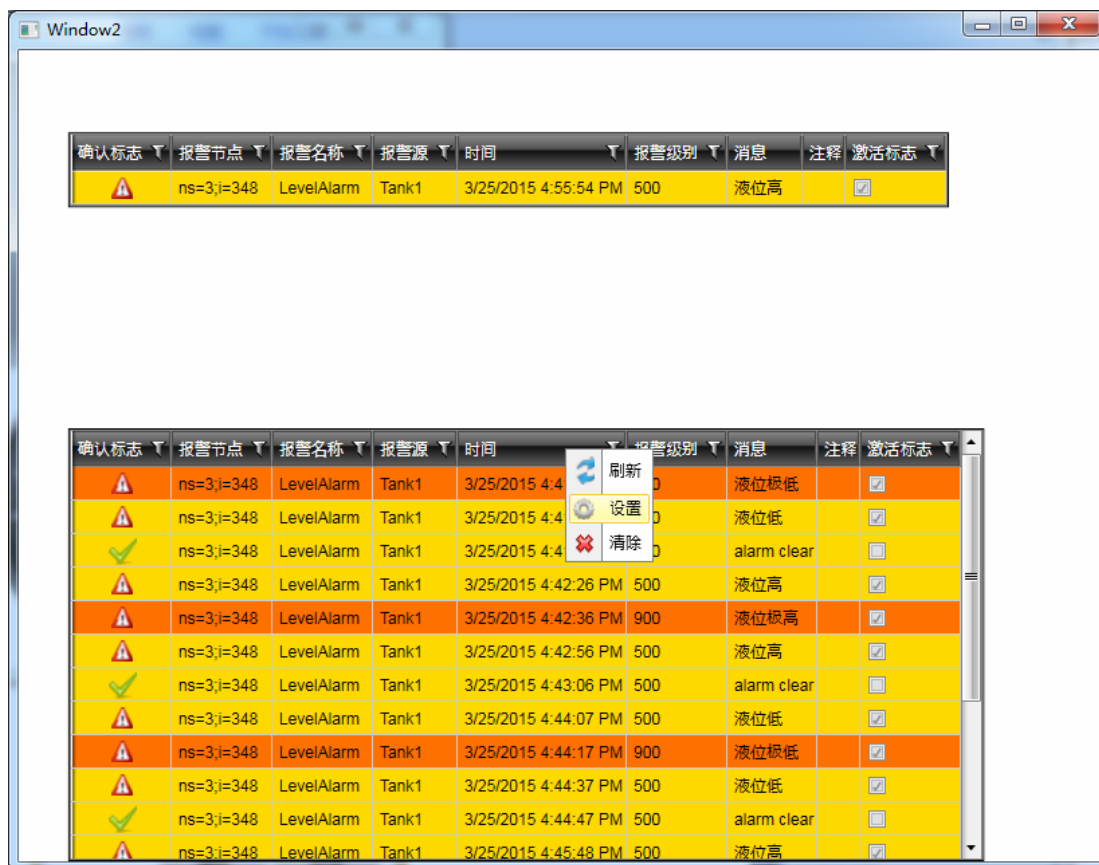


图 2-52 历史报警显示

右键点击此控件，选择设置，将弹出如下的设置窗口，

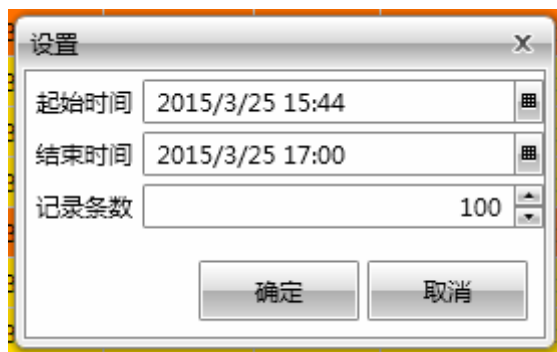


图 2-53 设置历史报警查询条件

设置起始，结束时间后，右键控件，点击“刷新”，就会显示历史报警。点

击“清除”，将清除所有历史记录。

## 2.8 趋势曲线

本节演示如何使用趋势控件，显示变量的实时和历史曲线。

### 2.8.1 实时趋势曲线

在 AicStudio 中新建一个 WPF 窗体，命名 Window3，从工具箱中选择 TrendGraph，添加到画面中，命名为“trend1”，

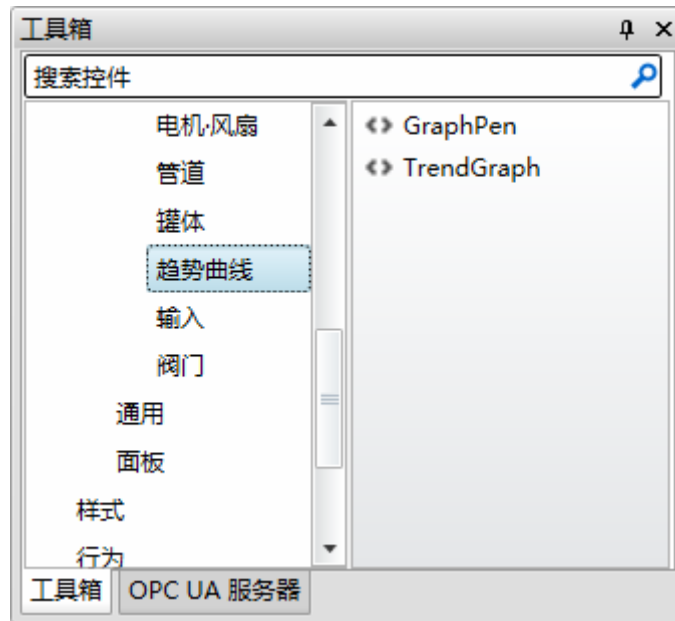


图 2-54 添加趋势曲线

宽度设置 600，高度设置 400，如下图所示：

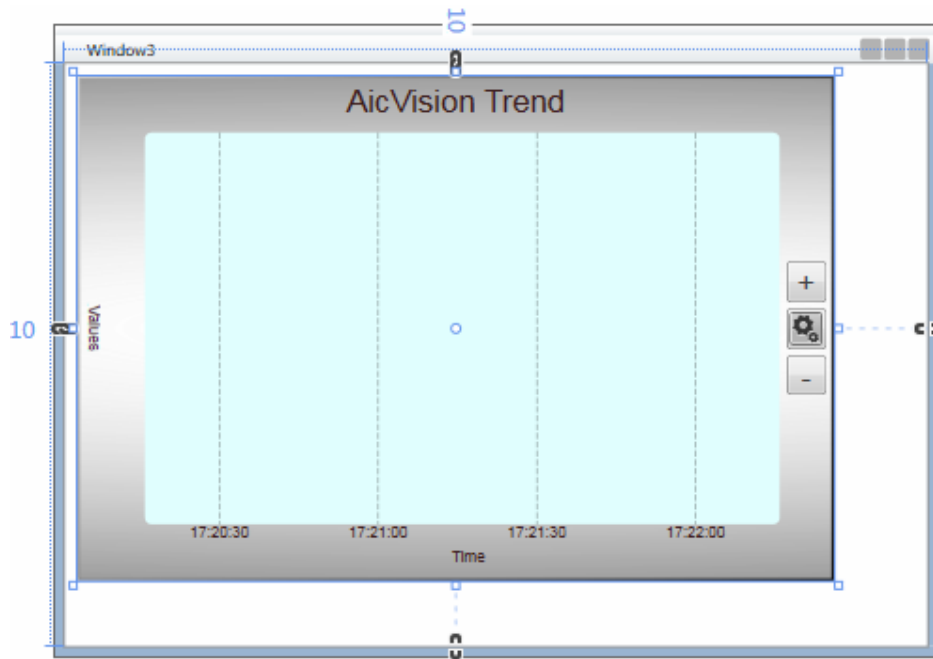


图 2-55 趋势曲线初始画面

然后从工具箱中拖入一个 GraphPen 到 trend1 上，命名为 pen1，在大纲窗口选中“pen1”，设置“Title”为“液位”，“UaDataValue”属性绑定到 Tank1 的 Level 变量上，注意绑定路径需要设置成“DataValue”，如下图所示：

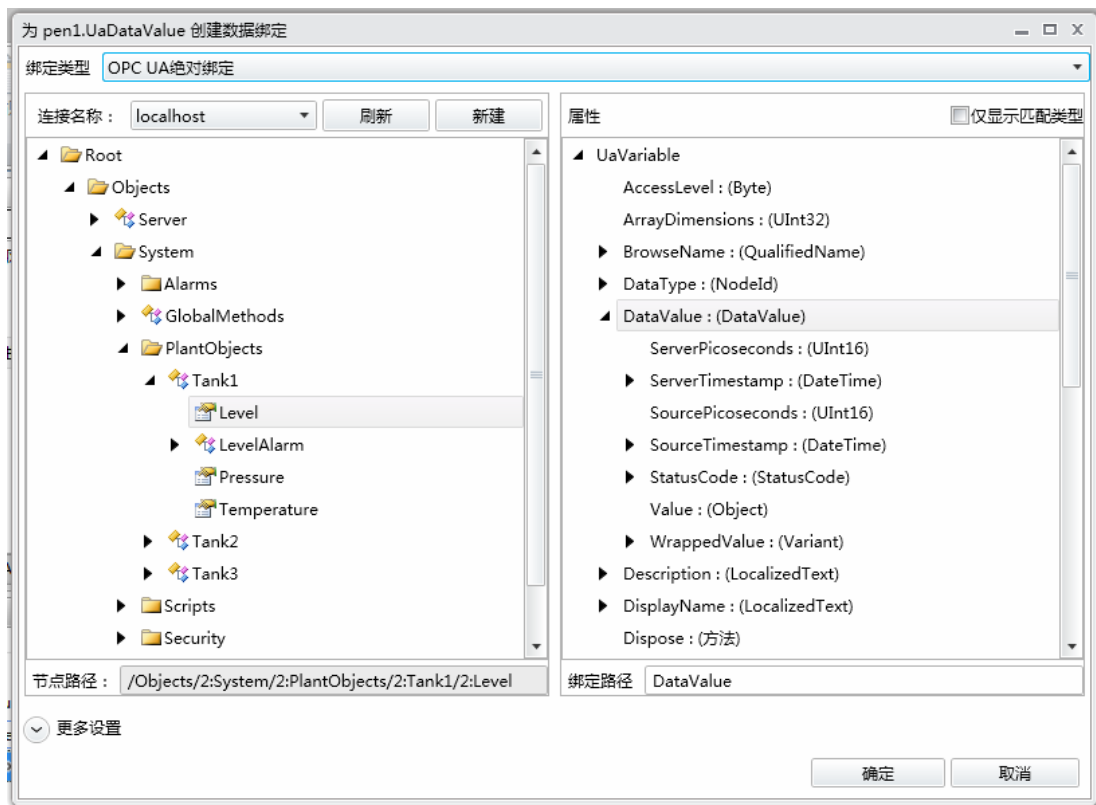


图 2-56 画笔的“UaDataValue”绑定变量

在主画面中添加一个按钮 `button6`，并添加脚本切换到 `Window3`，运行效果如下：

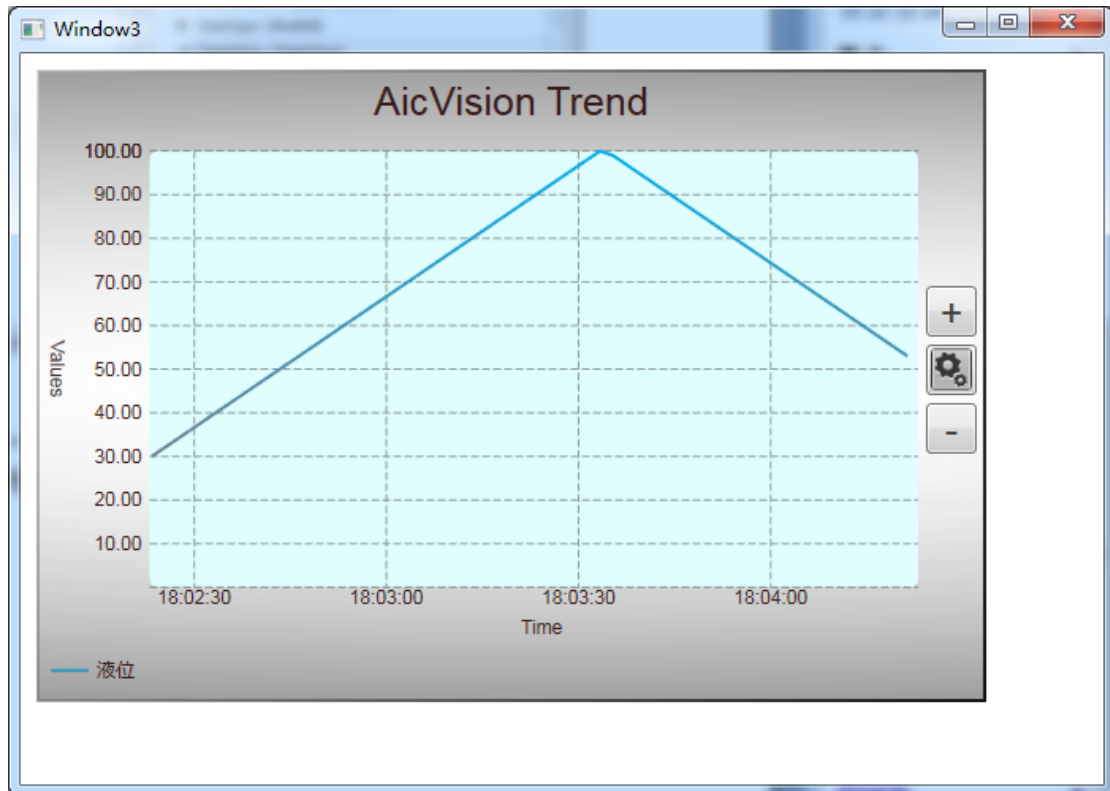


图 2-57 液位趋势曲线

类似地，增加画笔 `pen2`，`Title` 设置成“温度”，绑定到 `Tank1/Temperature`，“`Stroke`”（画笔颜色）设置成红色；增加画笔 `pen3`，`Title` 设置成“压力”，绑定到 `Tank1/Pressure`，画笔设置成绿色。修改后的运行效果如下：



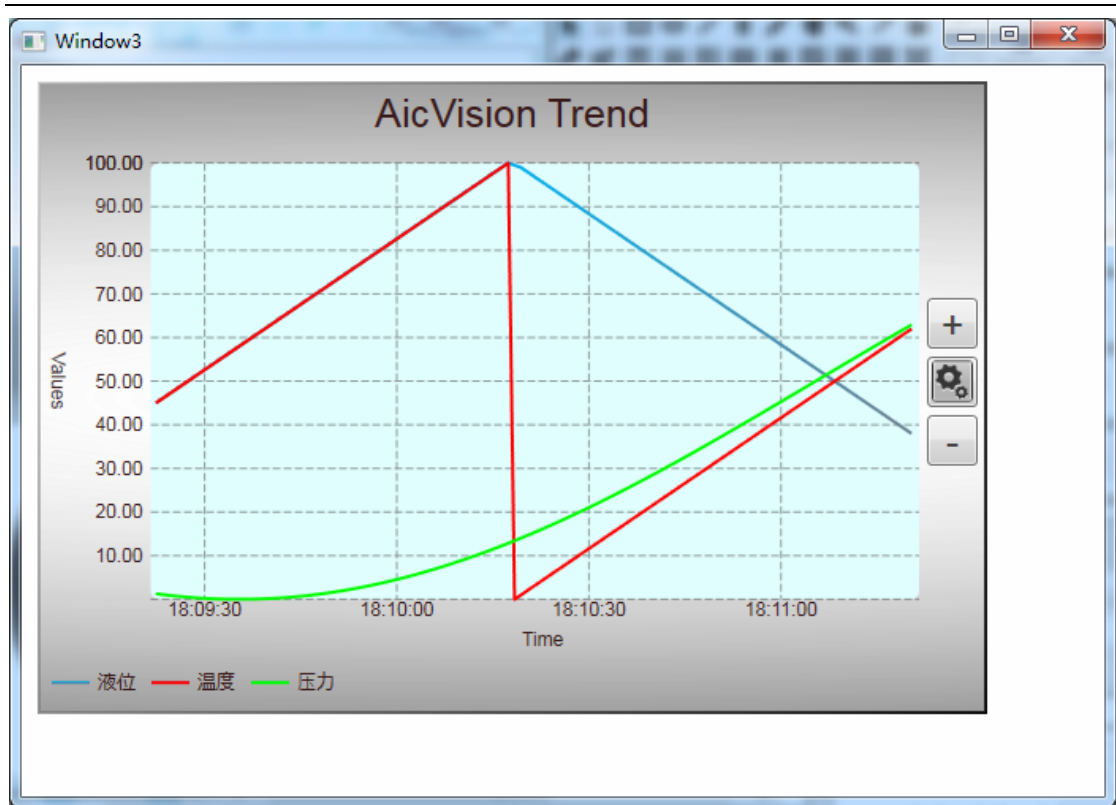


图 2-58 液位、温度和压力趋势曲线

### 2.8.2 实时库配置历史记录

查看历史曲线首先要在实时库中配置历史记录，打开 AicDatahub 配置工具，在工厂对象中选择 Tank1 的 Level 变量，打勾“保存历史数据”。同样地，对 Temperature 和 Pressure 变量进行配置。

---

父对象信息

父节点: 2:Tank1

引用类型: HasComponent

节点属性

节点分类: Variable

节点ID: ns=3;i=347

浏览名称: 2:Level

显示名称: Level

描述: 液位

可写标志: 1577587

数据类型: Double

维度: Scalar

值: 43

保存历史数据:

访问级别: 15

实例声明

类型定义: DataItemType

建模规则: None

图 2-59 设置保存历史记录

需要注意，历史记录功能仅在有正式版授权时才起作用，演示版中即使做如上设置，也不会记录。

### 2.8.3 历史趋势曲线

返回 AicStudio 的 TankMonitor 工程，在大纲中找到 pen1，设置其 HistoricalDataSource 属性，绝对绑定到 Tank1/Level 变量

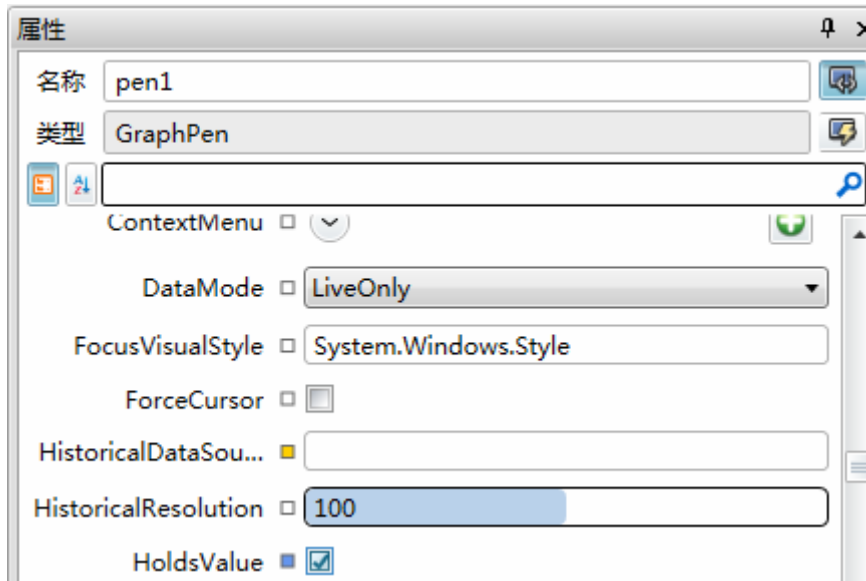



图 2-60 绑定画笔的 HistoricalDataSource 属性

同样地，绑定 pen2、pen3 的 HistoricalDataSource 属性到 Tank1/Temperature 和 Tank1/Pressure。运行程序，切换到 Window3，点击趋势曲线右侧的配置按钮 ，选择 Historical，并设置查询的起始/终止时间，就会显示相应的历史曲线。

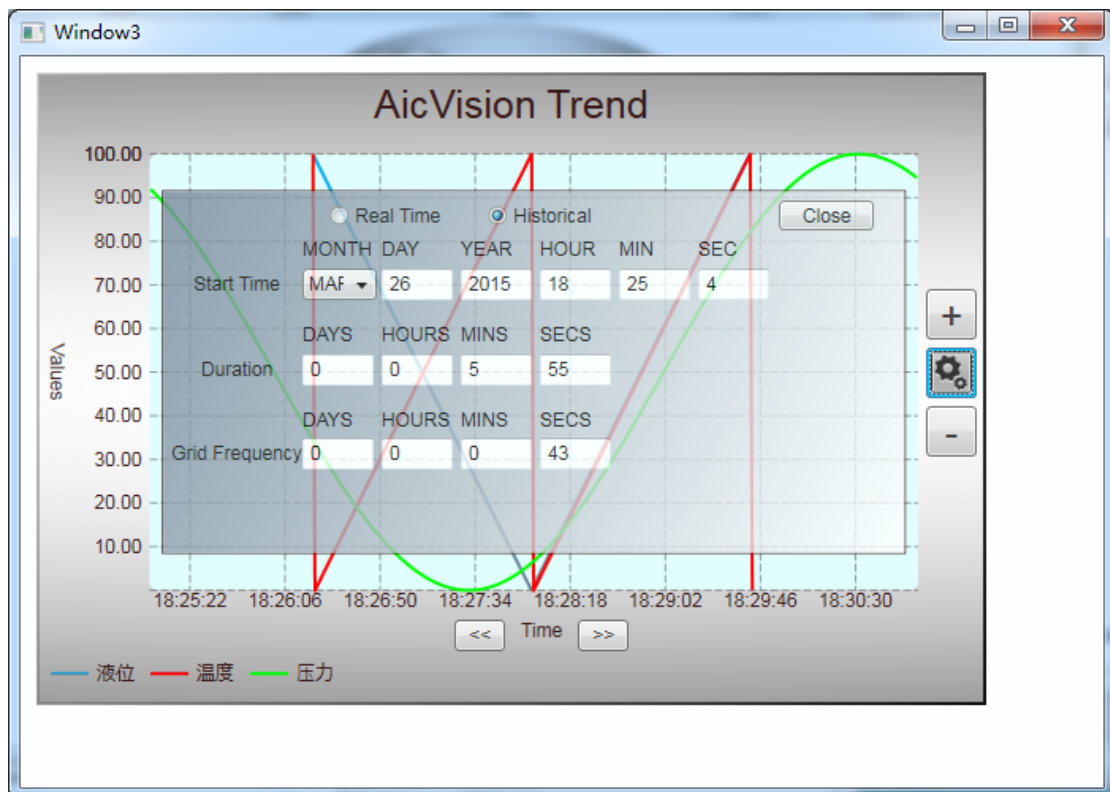


图 2-61 历史趋势曲线