

ProU OSC 曲线控件

简单编程，提升效率



做专业的装备开发平台



目录

1	第一章 控件说明	3
2	第二章 可视化中 OSC 控件的建立与设置	4
3	第三章 编写 C 语言.....	8
4	第四章 编写梯形图与在可视化中加入对应按钮.....	13
5	第五章 写入程序与测试	14
6	第六章 曲线制作与参数设置	18
6.1	曲线的制作	18
6.2	曲线的参数设置	21
6.2.1	坐标轴的固定	21
6.2.2	坐标轴的划分	23
6.2.3	坐标轴标题	24
6.2.4	坐标轴数值显示修改	25
6.2.5	存储的 Excel 文件曲线不显示设置	26





1 第一章 控件说明

OSC 控件与 Excel 控件两者均能够用来生成曲线，但有着些许的区别。

Excel 控件的曲线主要适用在采集大量数据后，**一次生成完整的曲线**。

而 **OSC 控件**则主要针对于**曲线的实时生成**。

Excel 曲线的制作已经有相关的说明手册，在此不再赘述，仅针对 OSC 控件进行说明。

与 OSC 控件相关联的共有四个函数：

1) //曲线复位

```
void GC_Osc_Rst(char *name);
```

2) //曲线行数据上传

```
void GC_Osc_AddRow(char *name, float *buff, int num);
```

3) //曲线列数据上传

```
void GC_Osc_AddCol(char *name, float *buff, int num);
```

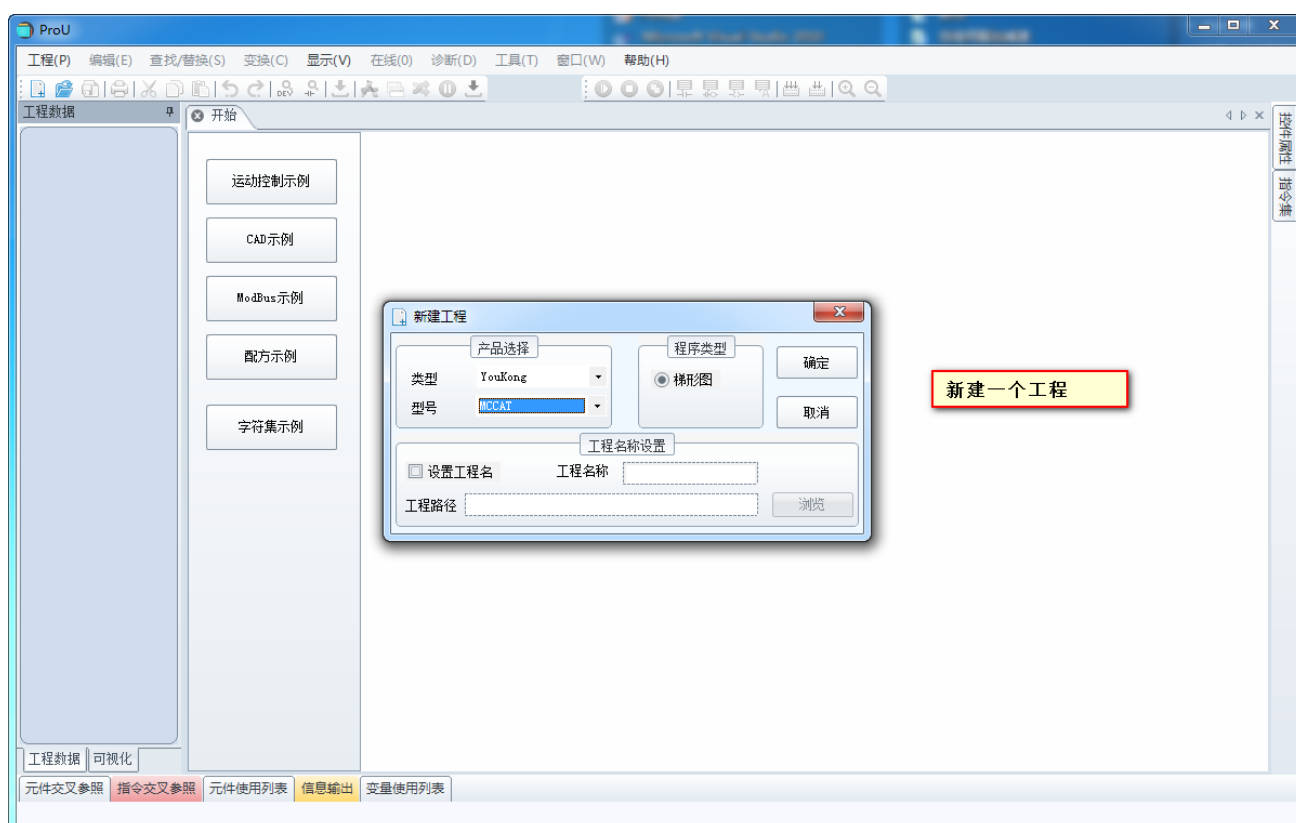
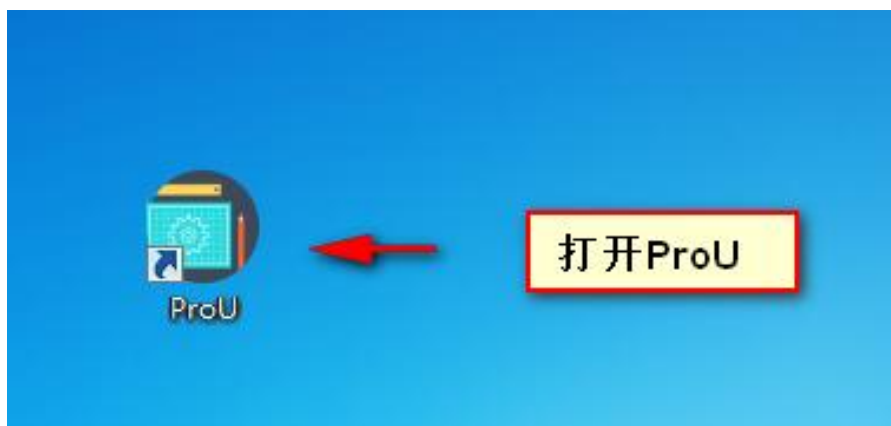
4) //曲线保存 subdir:子目录 format:"bmp" "png" "jpg" "xls"

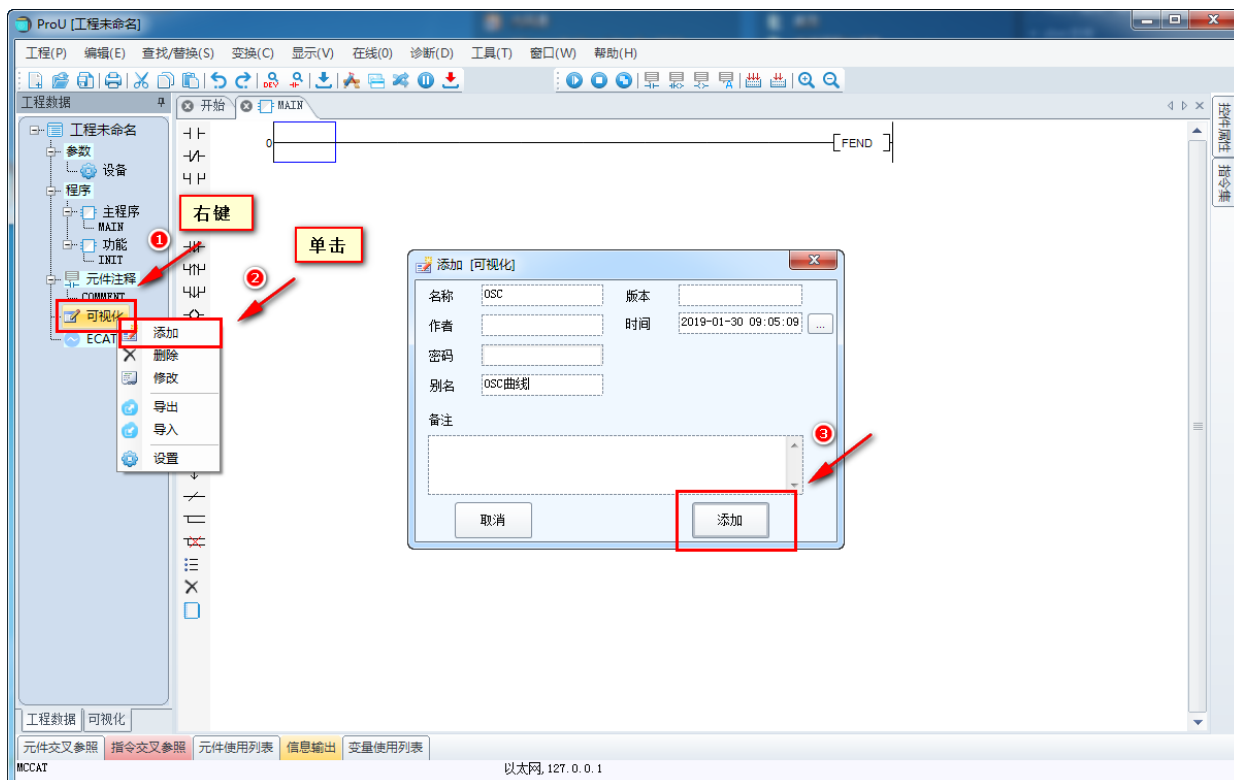
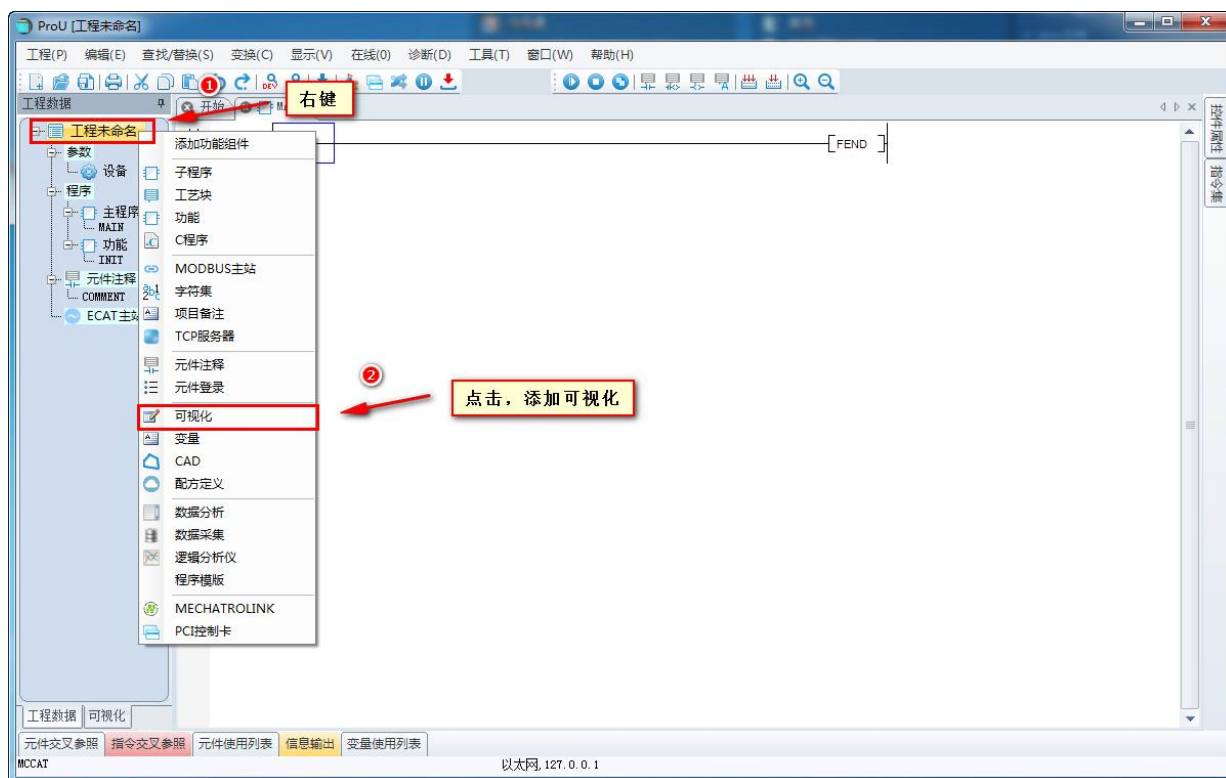
```
void GC_Osc_Save(char *name, char *subdir, char *format);
```

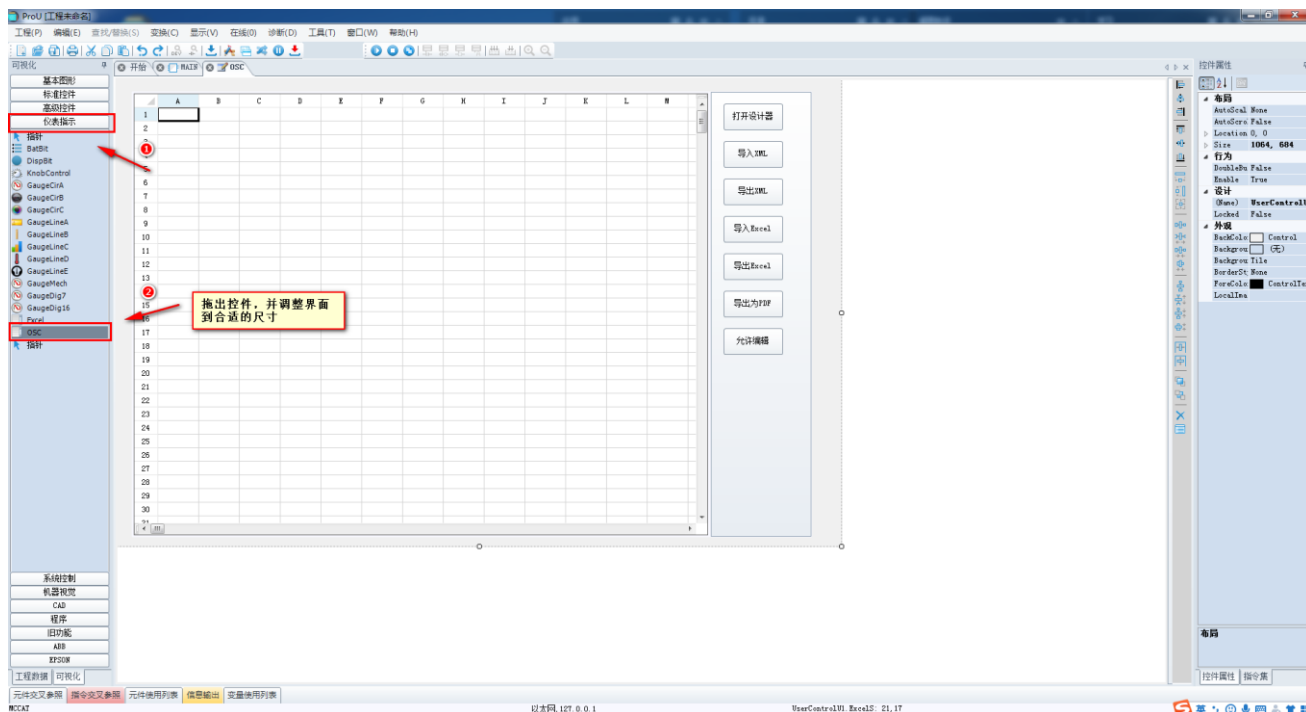
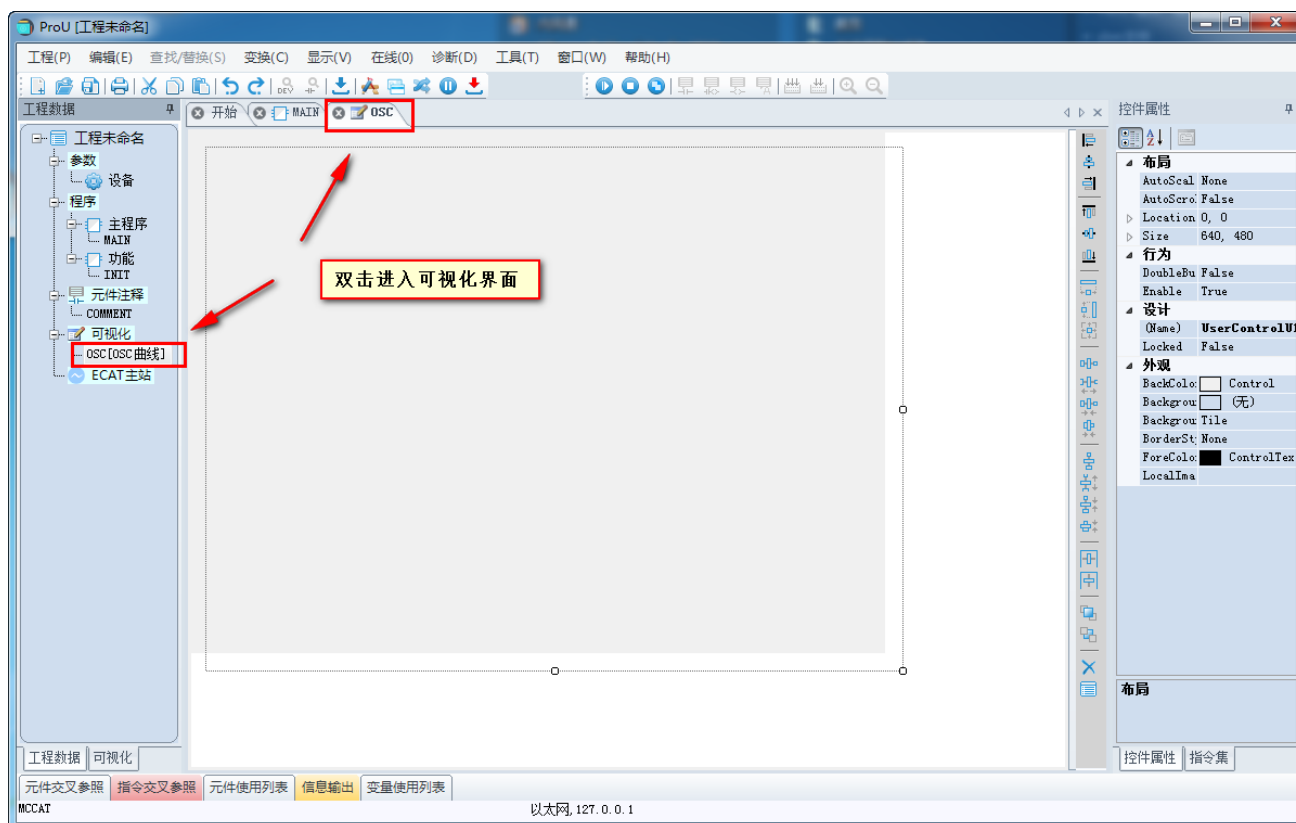
具体使用方法，请见后面章节。



2 第二章 可视化中 OSC 控件的建立与设置









The screenshot displays the ProU OSC software interface, which includes a spreadsheet editor on the left and a properties panel on the right.

Spreadsheet Editor:

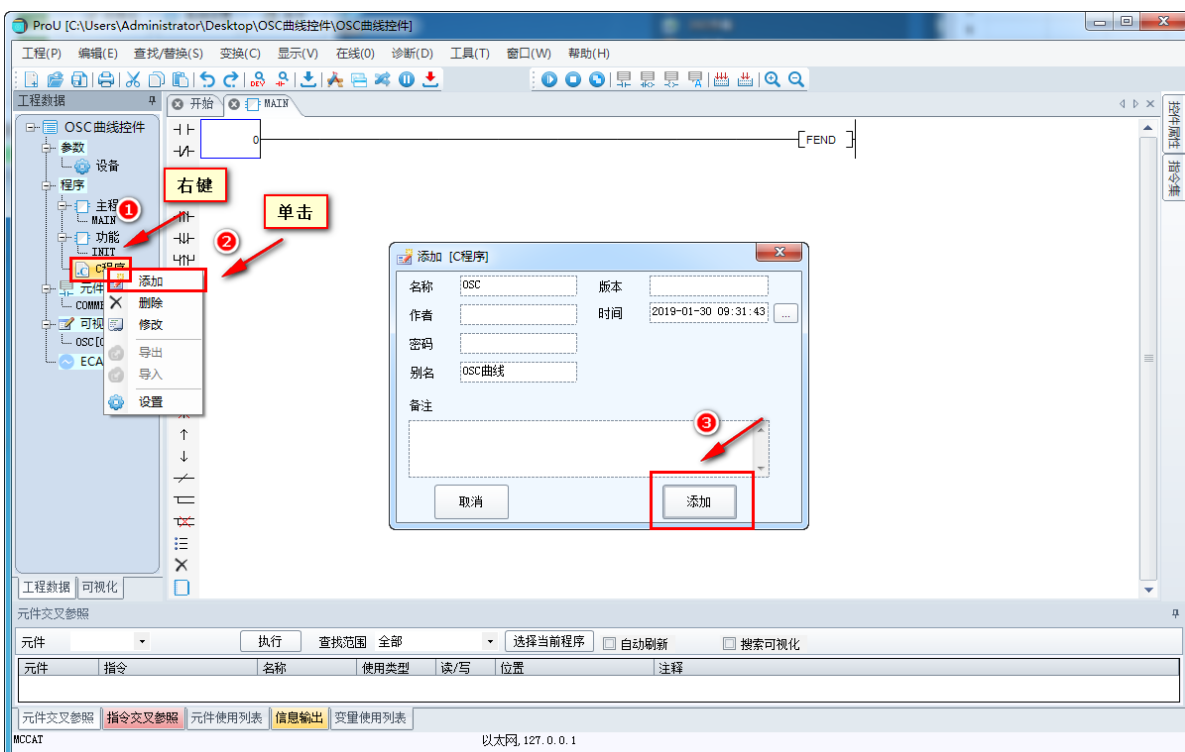
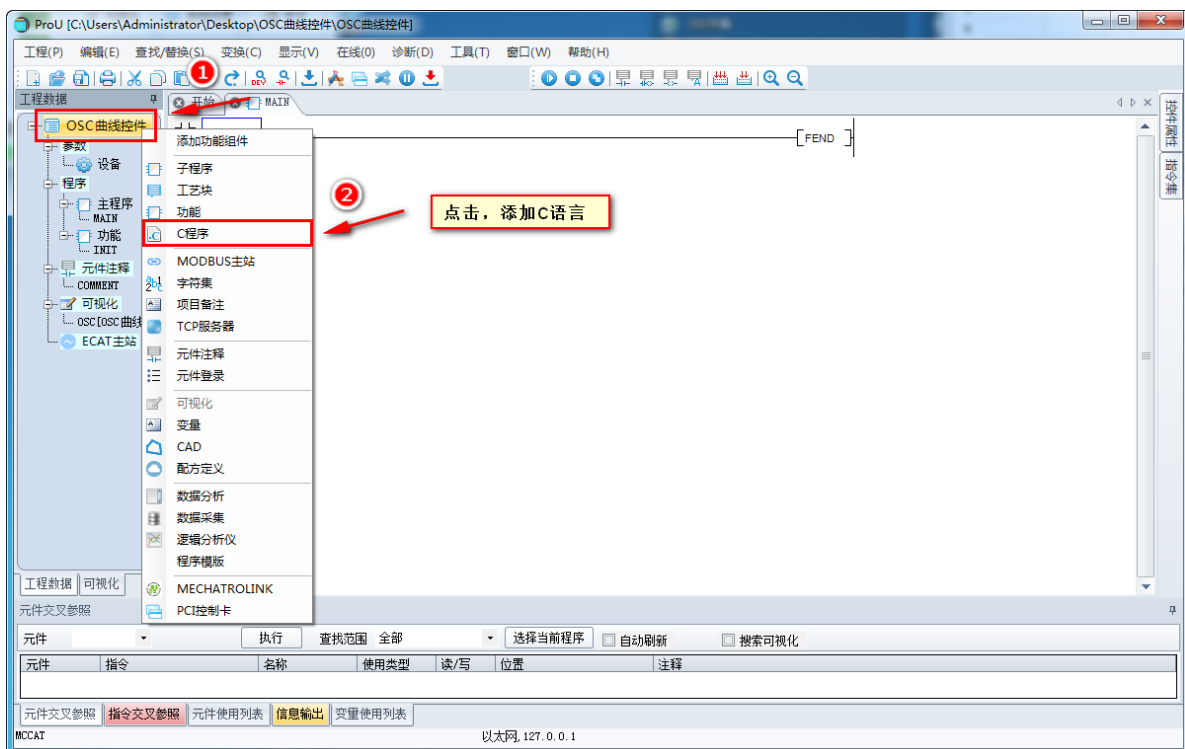
- Row 4: A red box highlights the text "控件的名字, C语言的函数中需要用到" (Control name, needed in C language functions).
- Row 8: A red box highlights the text "DataCol: 数据列数" (DataCol: Number of data columns).
- Row 9: A red box highlights the text "DataRow: 数据行数" (DataRow: Number of data rows).
- Row 10: A red box highlights the text "DpsplayCol: 显示数据开始列数" (DpsplayCol: Starting column for displaying data).
- Row 11: A red box highlights the text "DisplayRow: 显示数据开始行数" (DisplayRow: Starting row for displaying data).
- Row 21: A red box highlights the text "存储图片、Excel表格路径" (Storage path for images and Excel tables).

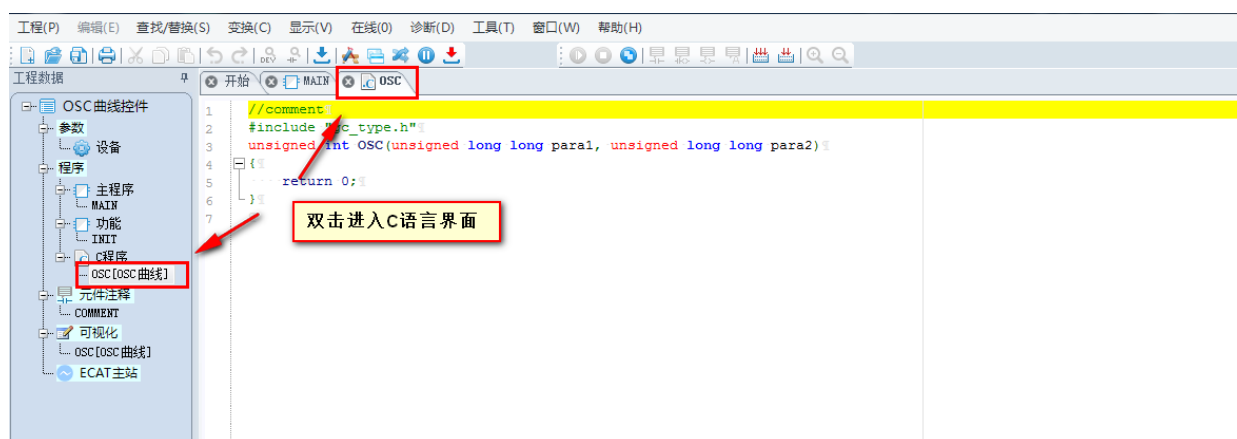
Properties Panel:

- Row 1: A red box highlights the text "控件的名字, C语言的函数中需要用到" (Control name, needed in C language functions).
- Row 2: A red box highlights the text "DataCol: 数据列数" (DataCol: Number of data columns).
- Row 3: A red box highlights the text "DataRow: 数据行数" (DataRow: Number of data rows).
- Row 4: A red box highlights the text "DpsplayCol: 显示数据开始列数" (DpsplayCol: Starting column for displaying data).
- Row 5: A red box highlights the text "DisplayRow: 显示数据开始行数" (DisplayRow: Starting row for displaying data).
- Row 6: A red box highlights the text "存储图片、Excel表格路径" (Storage path for images and Excel tables).



3 第三章 编写 C 语言





```
static char *gra1 = "Press1"; //将OSC空间的name一行的名字定义成字符
float row_data = 0; //定义一个浮点数据，以供测试
float col_data = 0; //定义一个浮点数据，以供测试

unsigned int rest_d(unsigned long long para1, unsigned long long para2)
{
    if (!SM[481])
    {
        return 0;
    }
    GC_Osc_Rst(gra1); //清空osc控件
    row_data = 0; //数据清零
    col_data = 0; //数据清零
    return 0;
}
```

```
void GC_Osc_Rst(char *name);
```

Name 处填入定义好的字符

这里需要注意的是，一定要先将可视化中的OSC控件name处的名字定义成字符，才可以在函数中使用。这个函数的功能是清空OSC控件内的数据。不是清除为0，而是为null。



```
unsigned int get_r(unsigned long long para1, unsigned long long para2){  
{  
    float fv[4]; //定义一个浮点数区间  
    if(!SM[481]){  
        return 0;  
    }  
    fv[0] = row_data; //将数据分别放入对应区间  
    fv[1] = col_data; //将数据分别放入对应区间  
    GC_Osc_AddRow(gra1, fv, 2); //将区间1、2的数据写入osc控件，数据逐行写入  
    col_data += 3; //数据递增  
    row_data += 2; //数据递增  
    return 0;  
}
```

```
void GC_Osc_AddRow(char *name, float *buff, int num);
```

name 处填入定义的字符名

buff 处填入定义的区域

num 处填入写入数据时，每次写入多少个数据的数量

每执行一次这个函数，都会往OSC控件中写入一行数据。本示例就是以两列数据为一行写入OSC控件中。



```
1
unsigned int get_c(unsigned long long para1, unsigned long long para2){
2 {
3     float fv[4];>>>> //定义一个浮点数区间
4     if(!SM[481]){
5         return 0;
6     }
7     fv[0] = 1.1;>>>> //将数据分别放入对应区间
8     fv[1] = 2.2;>>>> //将数据分别放入对应区间
9     fv[2] = 3.3;>>>> //将数据分别放入对应区间
10    >>>>
11    GC_Osc_AddCol(gra1, fv, 3);>>>> //将区间1、2、3的数据写入osc控件，数据逐列写入
12    >>>>
13    return 0;
14 }
15 }
```

```
void GC_Osc_AddCol(char *name, float *buff, int num);
```

name 处填入定义的字符名

buff 处填入定义的区域

num 处填入写入数据时，每次写入多少个数据的数量

每执行一次这个函数，都会往OSC控件中写入一行数据。本示例就是以三行数据为一列写入OSC控件中。





```
1
unsigned int save(unsigned long long para1, unsigned long long para2){
{
> if(!SM[481]){
>     return 0;
> }
> GC_Osc_Save(gra1, "", "bmp");//将osc控件显示的内容，以bmp图片格式存储在存储路径下
> return 0;
}
}
```

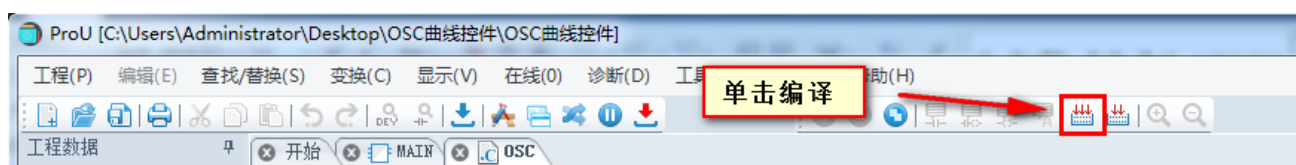
```
//曲线保存 subdir:子目录 format:"bmp" "png" "jpg""xls"
void GC_Osc_Save(char *name, char *subdir, char *format);
```

每执行一次该函数，都会以末尾的格式，保存对应的OSC控件的内容。

Name 处填入定义的字符名

Subdir 处填入保存路径下的子目录。本控件保存是在可视化中定义储存路径，保存的图片/表格以 存储路径 → 日期 → 子目录（即Subdir参数），如果为空，则没有子目录。

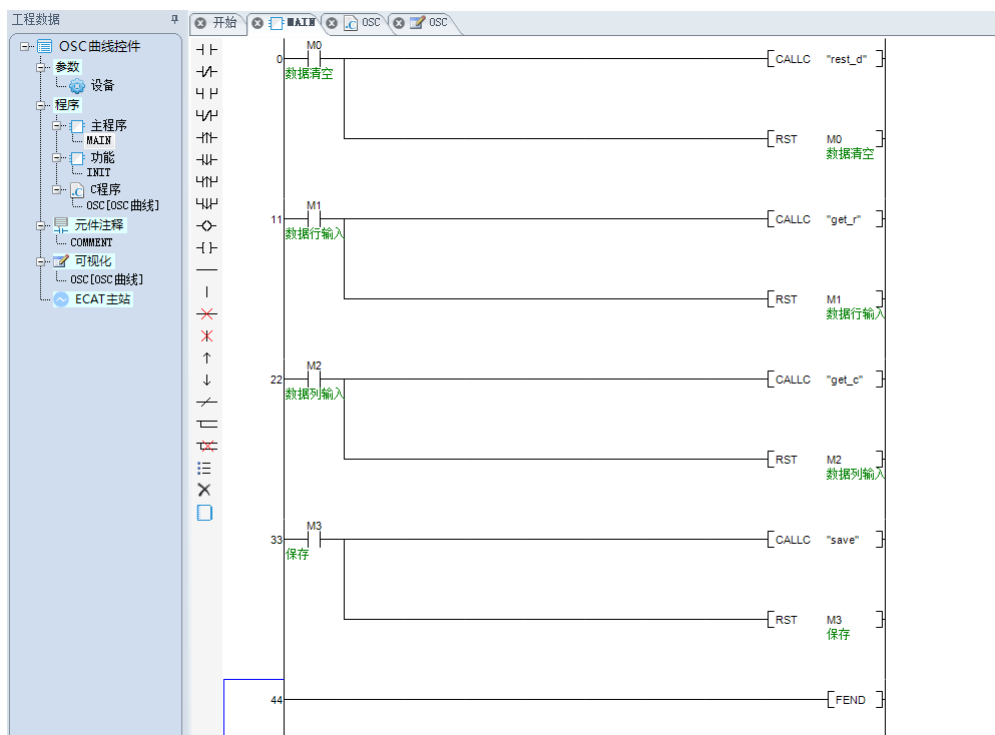
Format 处填入希望保存的格式。支持的图片格式有bmp、png、jpg；支持的excel表格的格式为xls。如果希望同时保存图片与Excel表格，只需将本函数写两次，改变Format参数即可。



编写完C语言后，点击编译

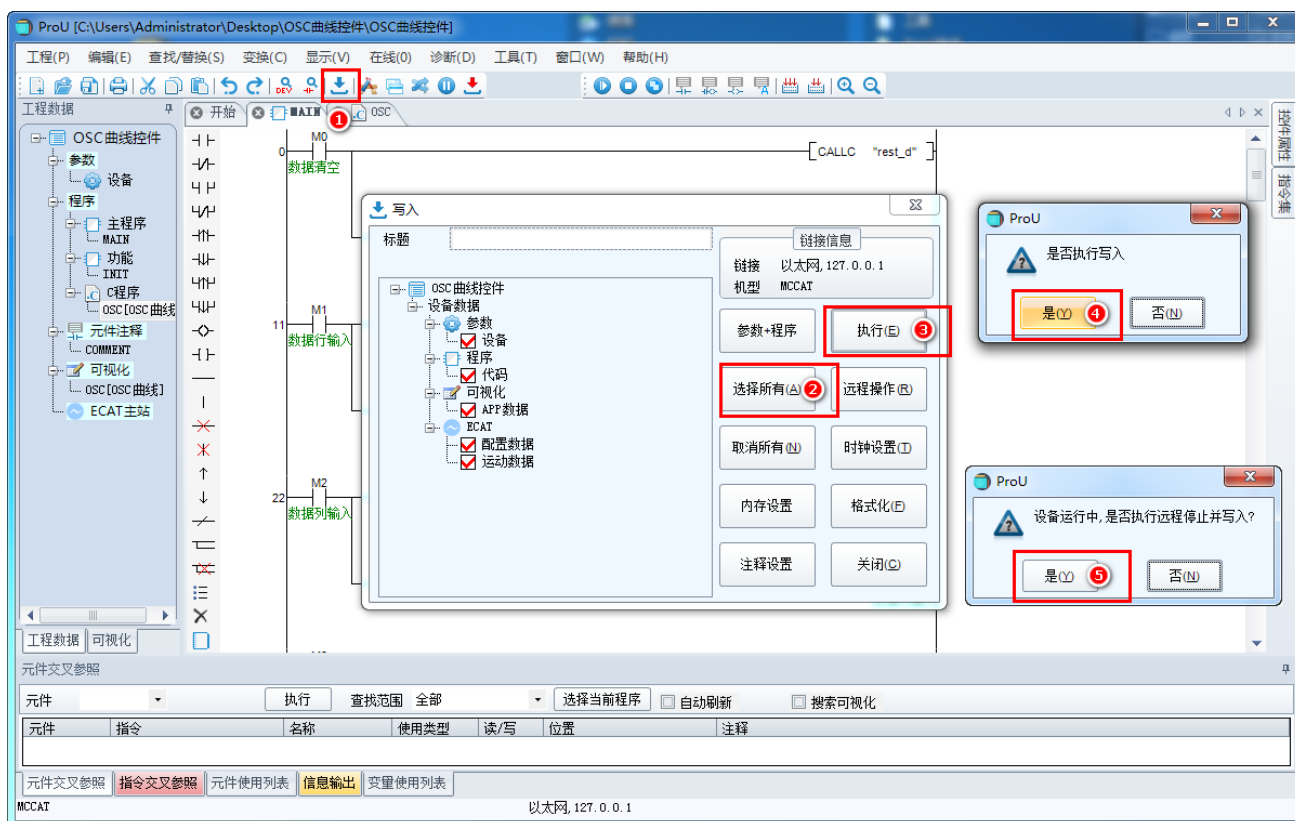


4 第四章 编写梯形图与在可视化中加入对应按钮

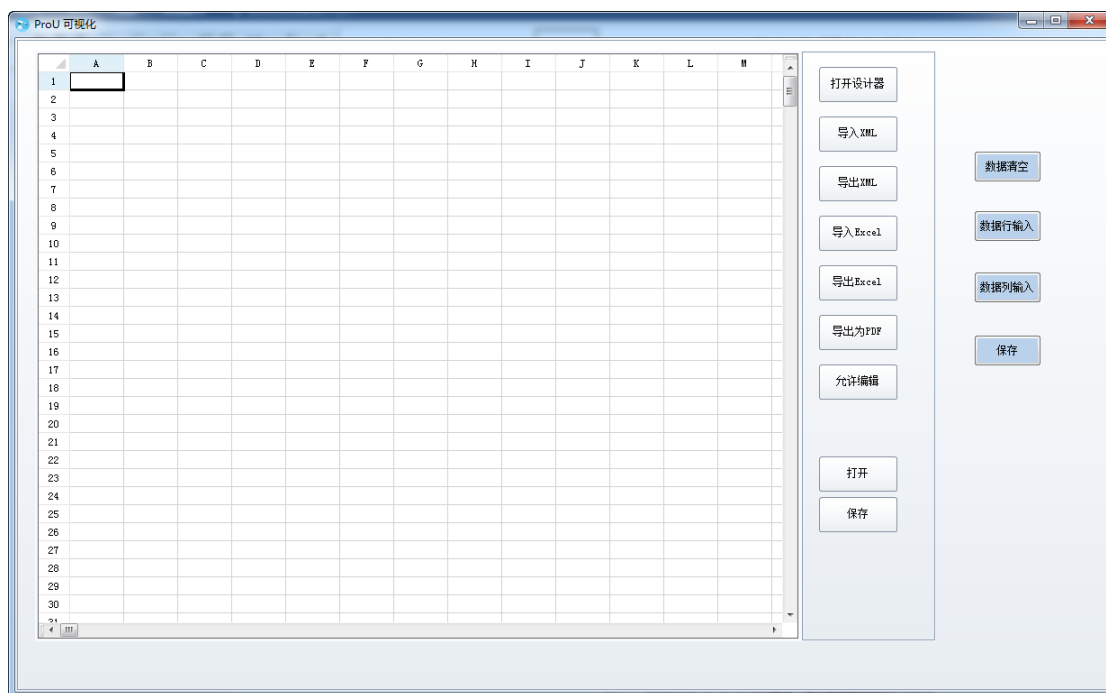




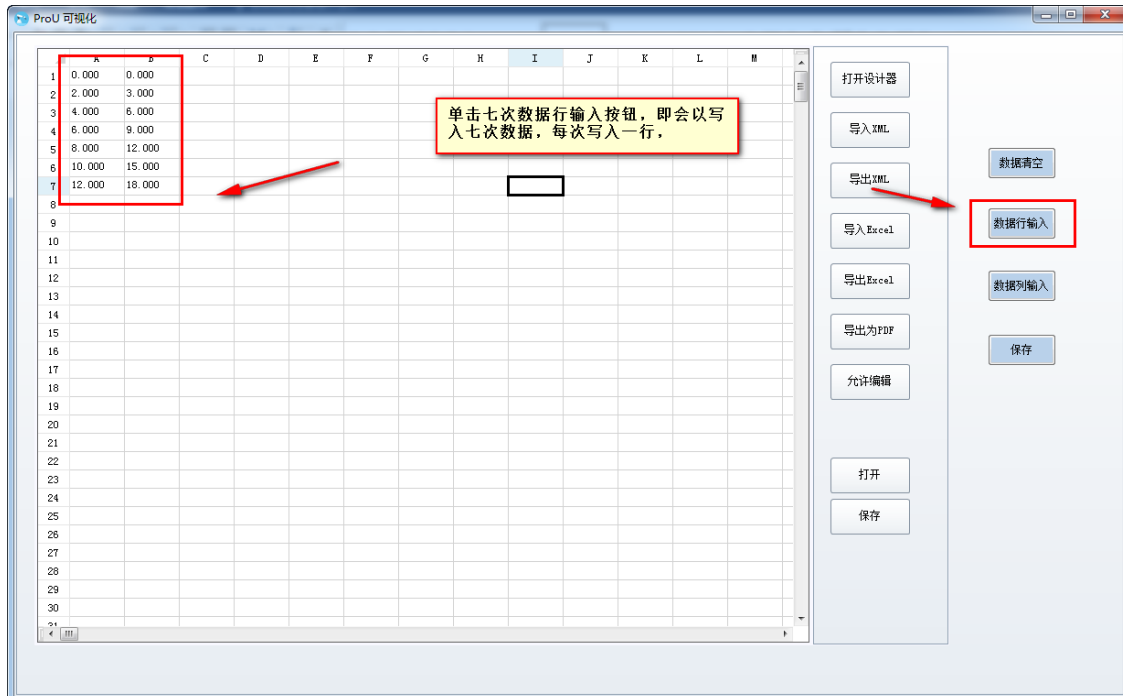
5 第五章 写入程序与测试

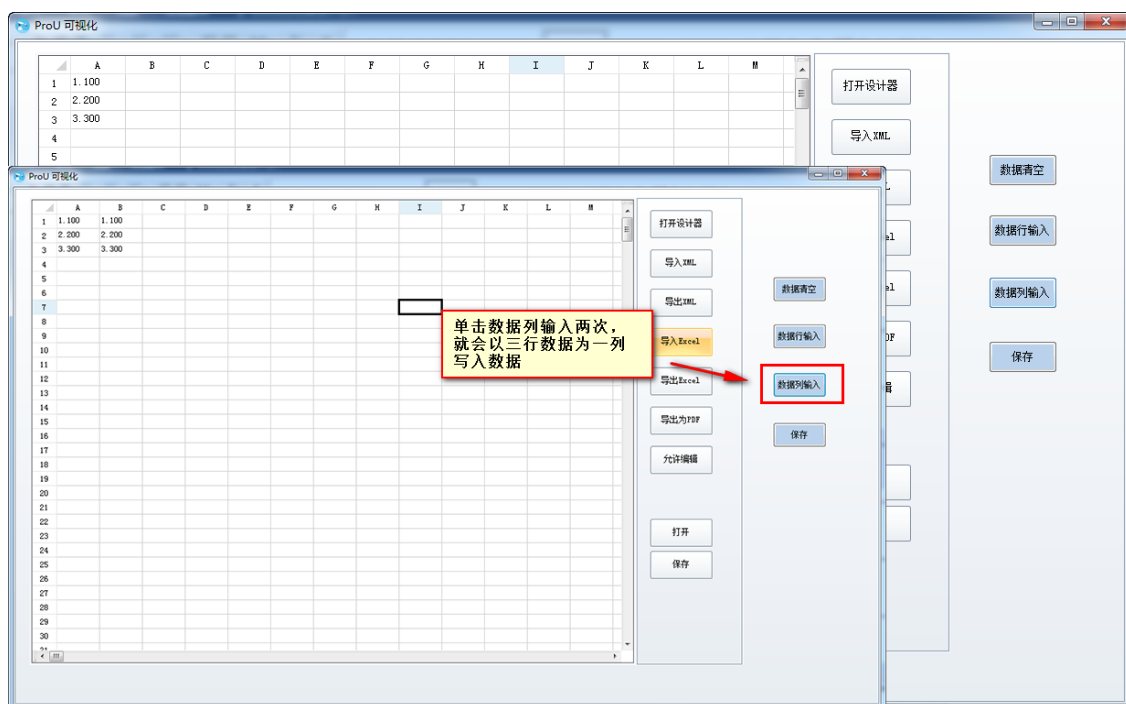
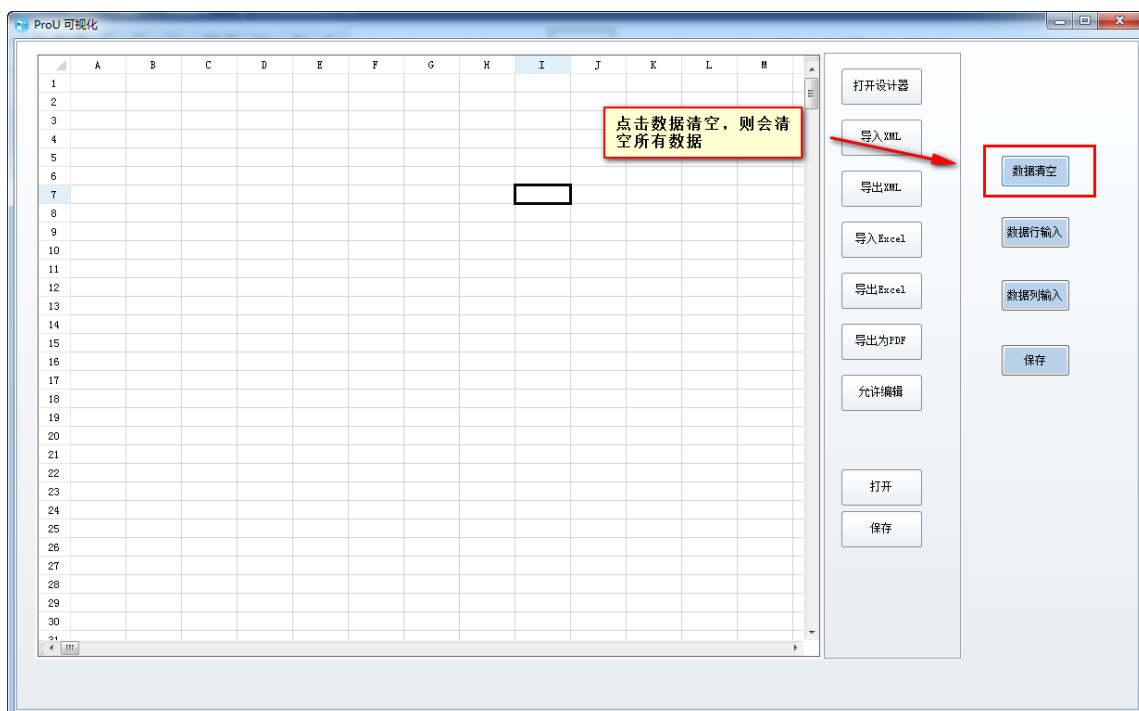


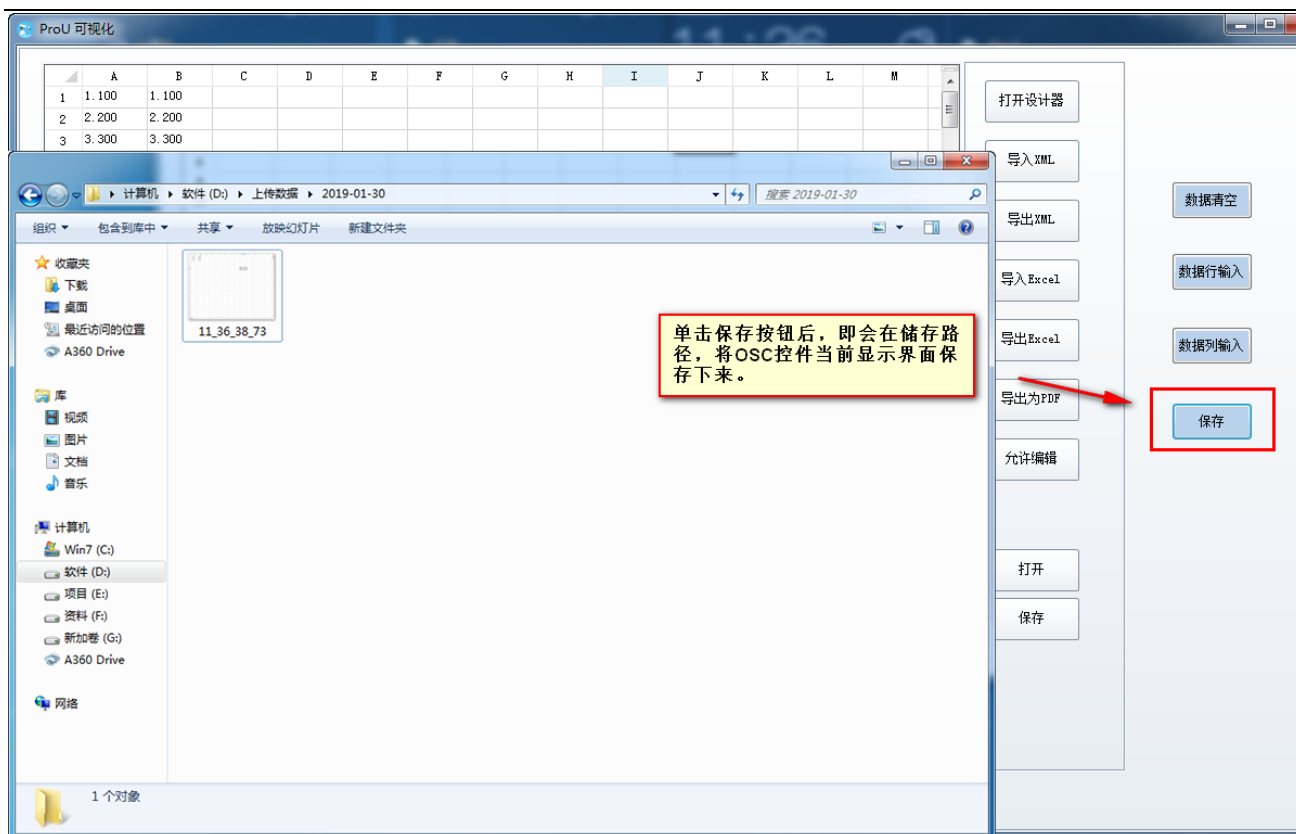
写入完成后，重启 ProH。



重启 ProH 后即会看到设计好的界面。

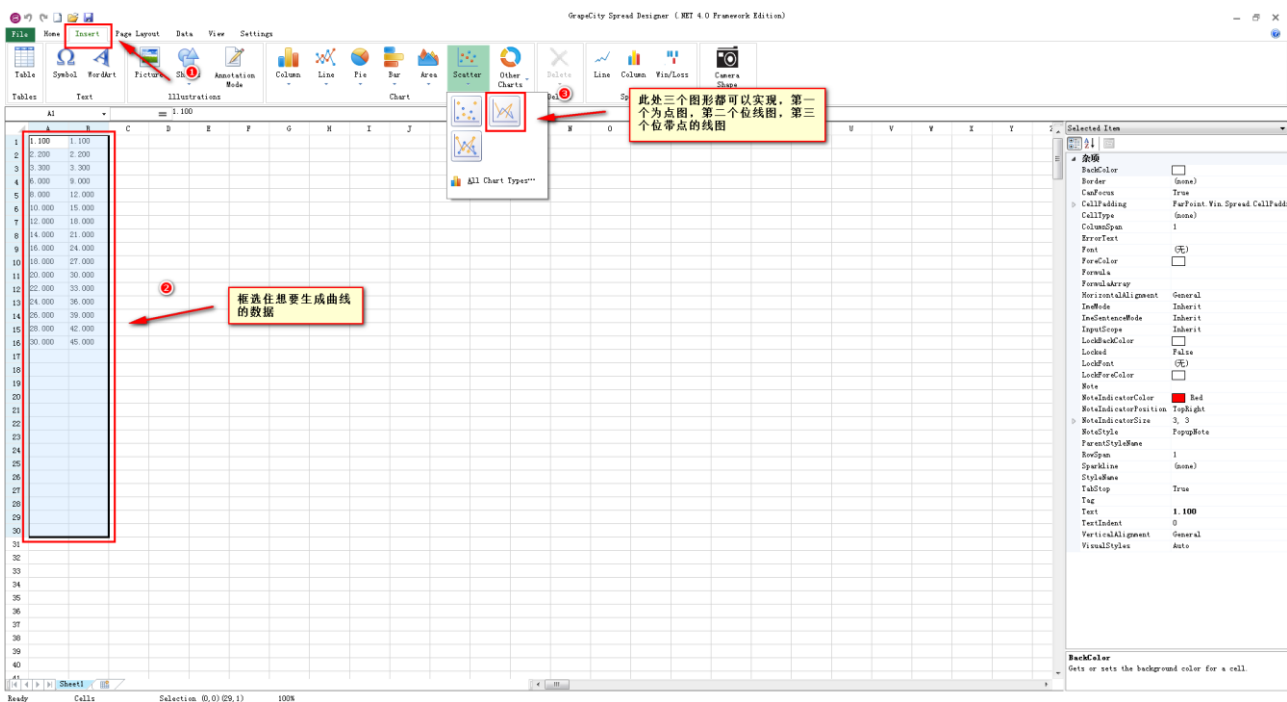
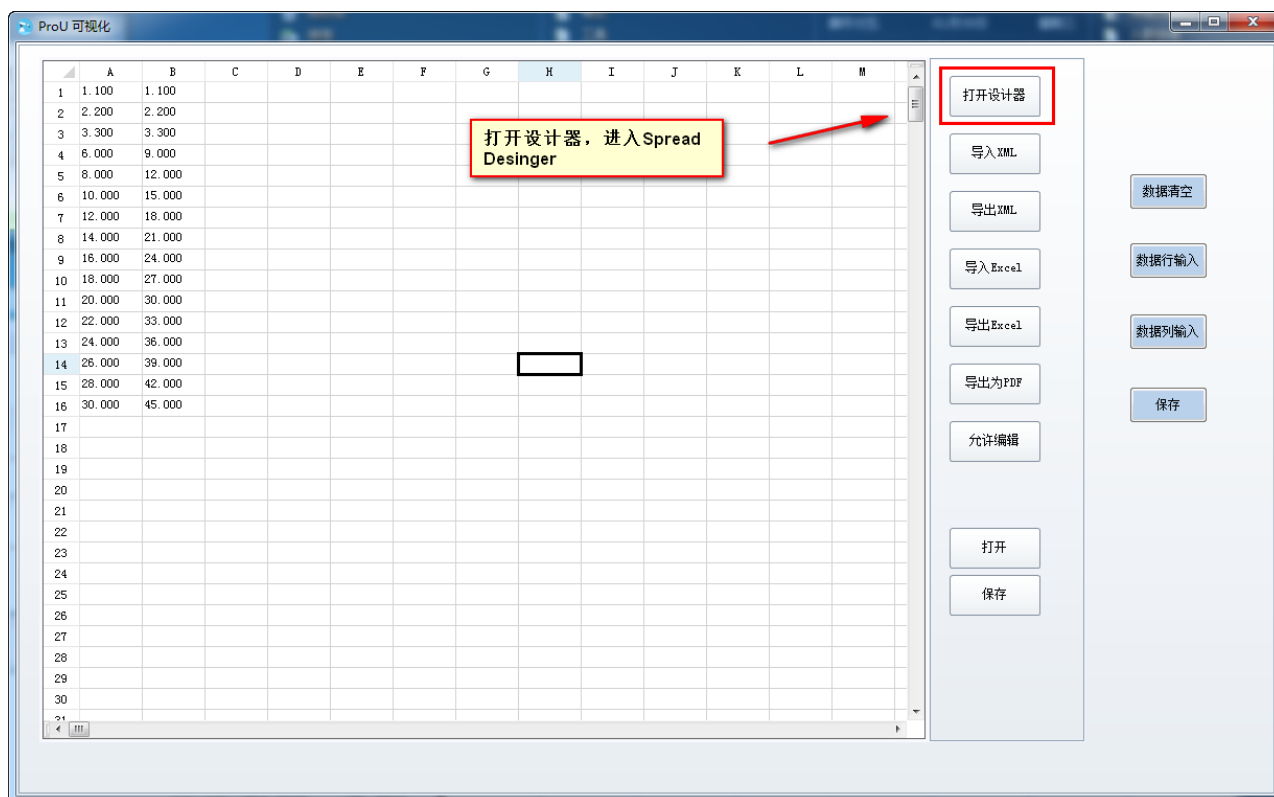


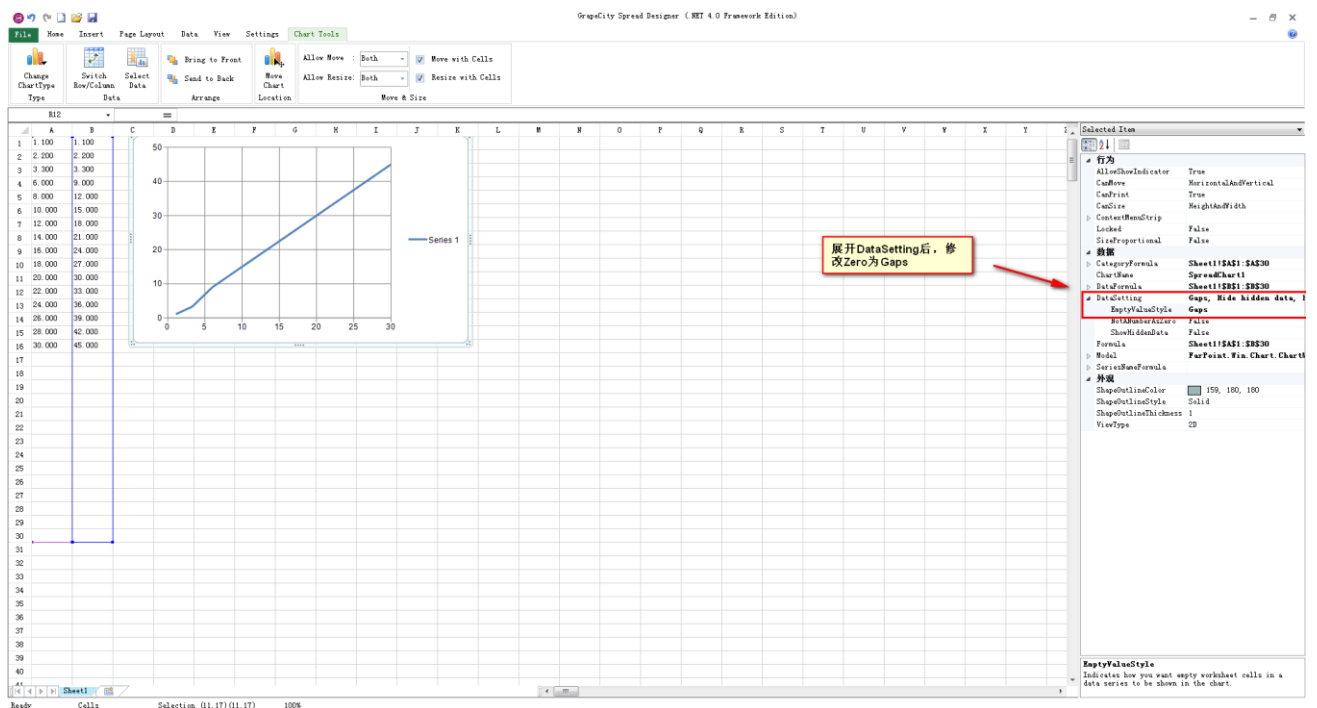
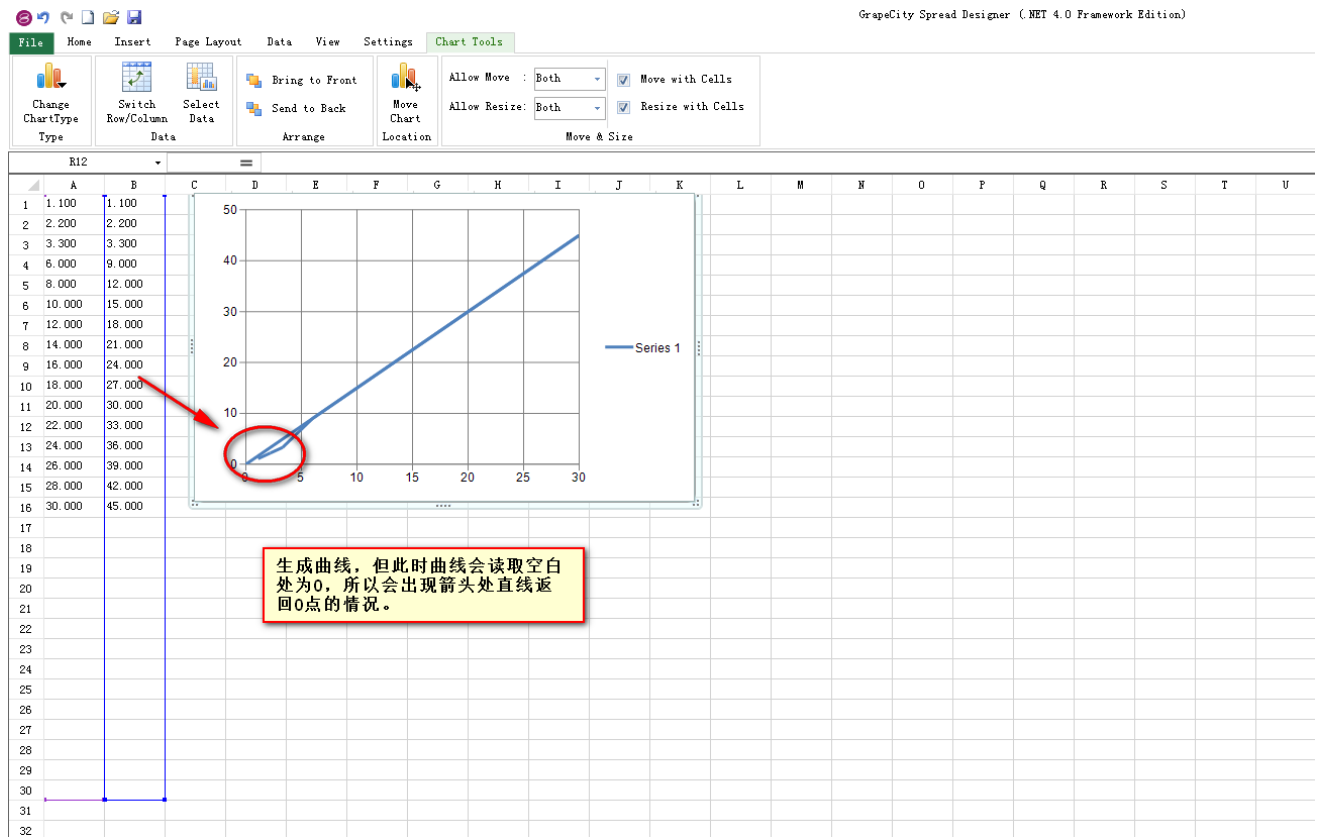


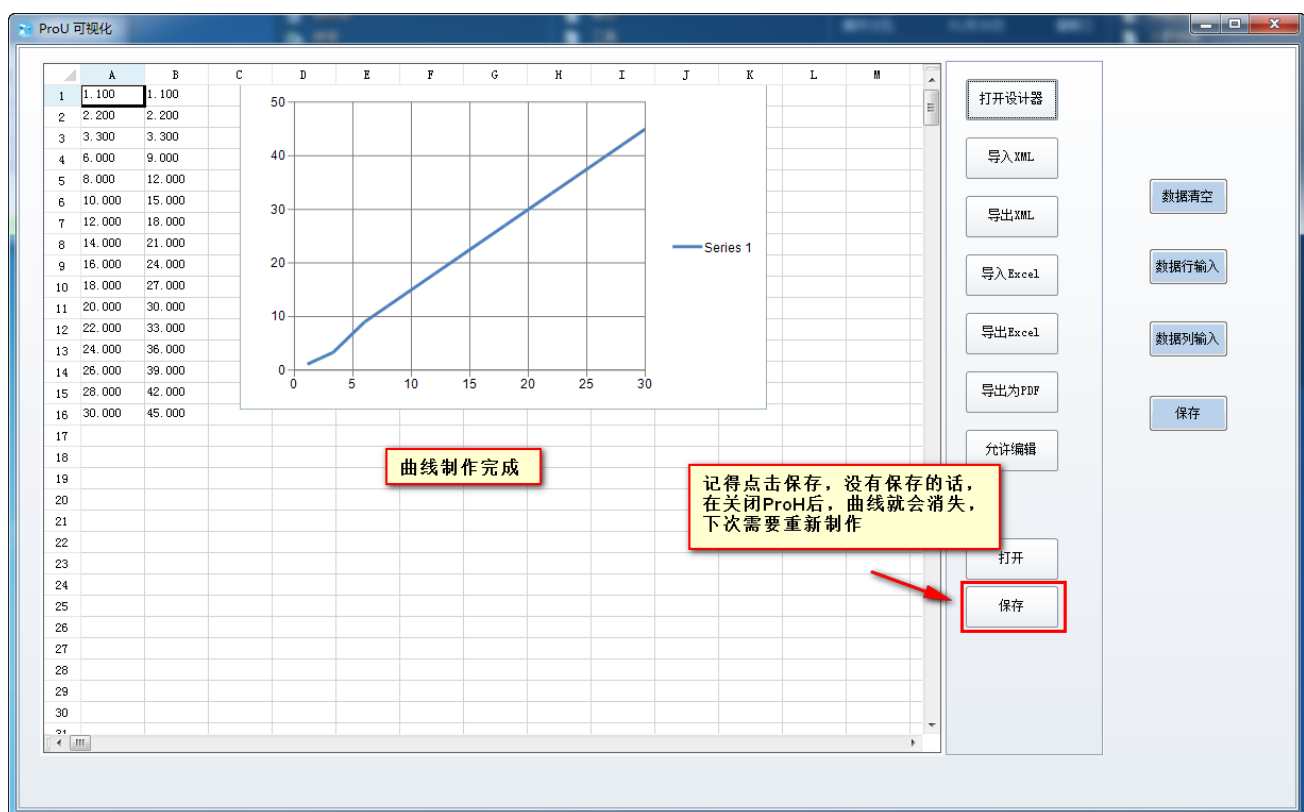
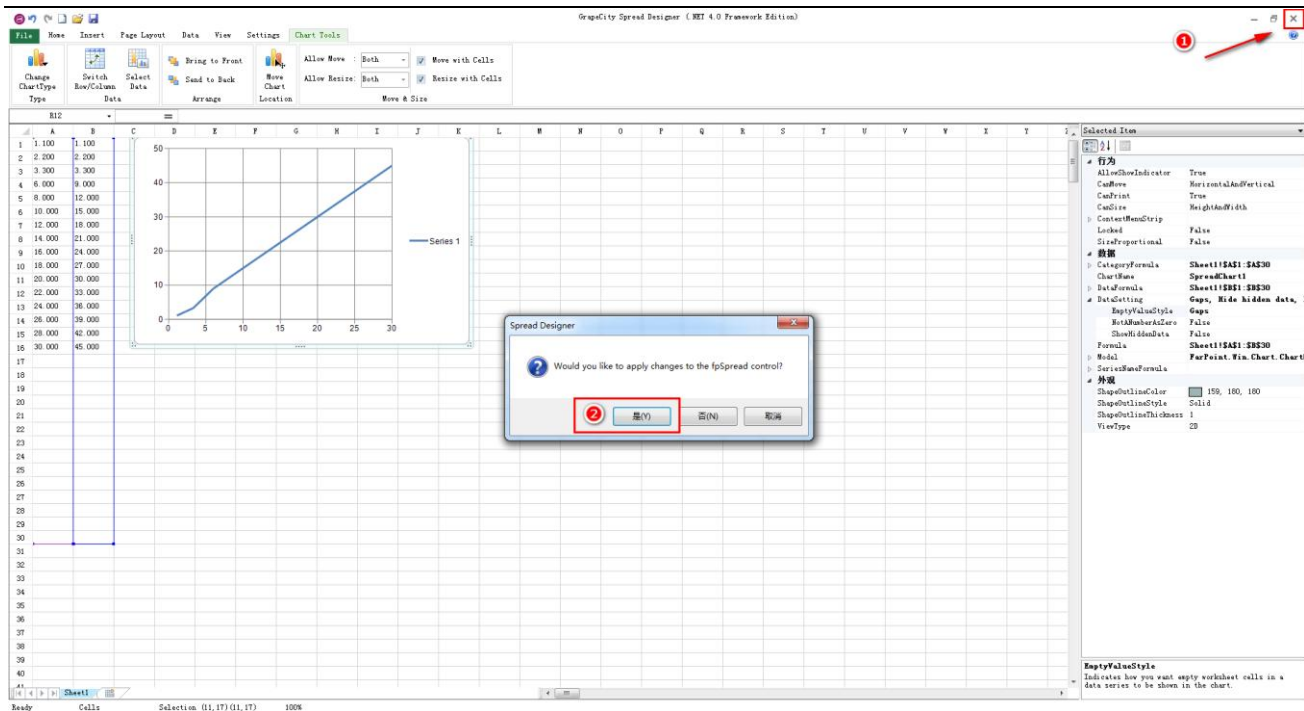


6 第六章 曲线制作与参数设置

6.1 曲线的制作



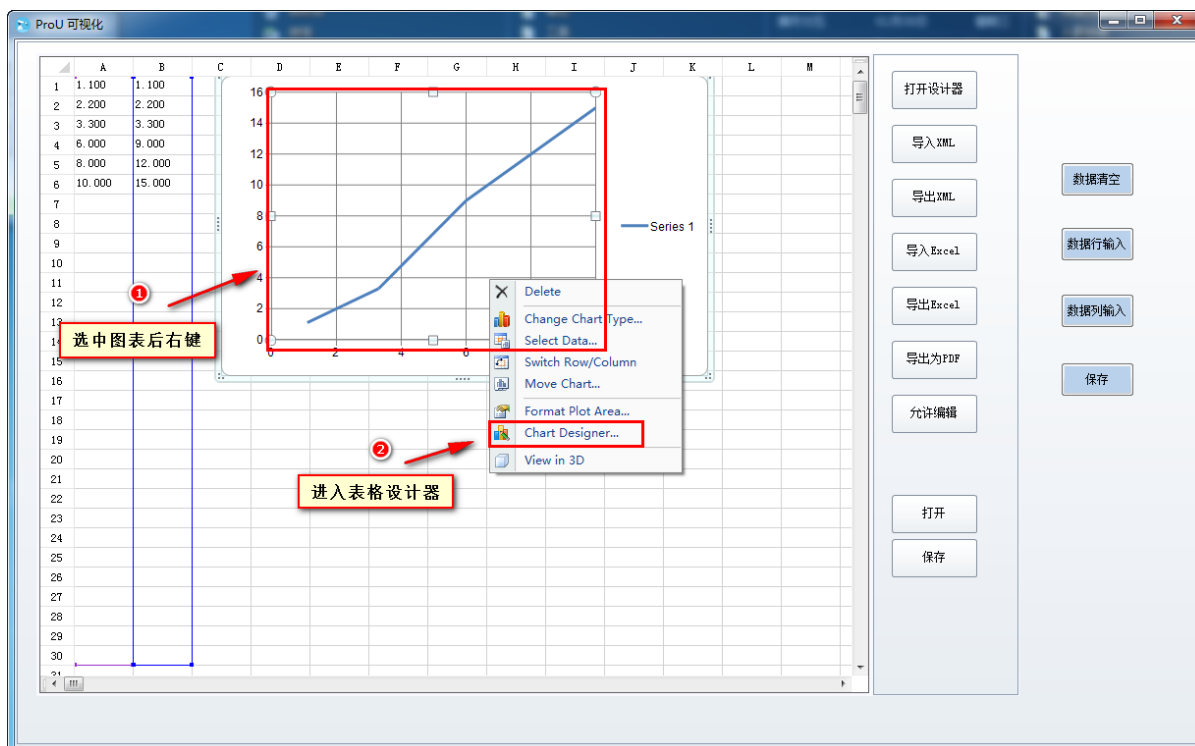
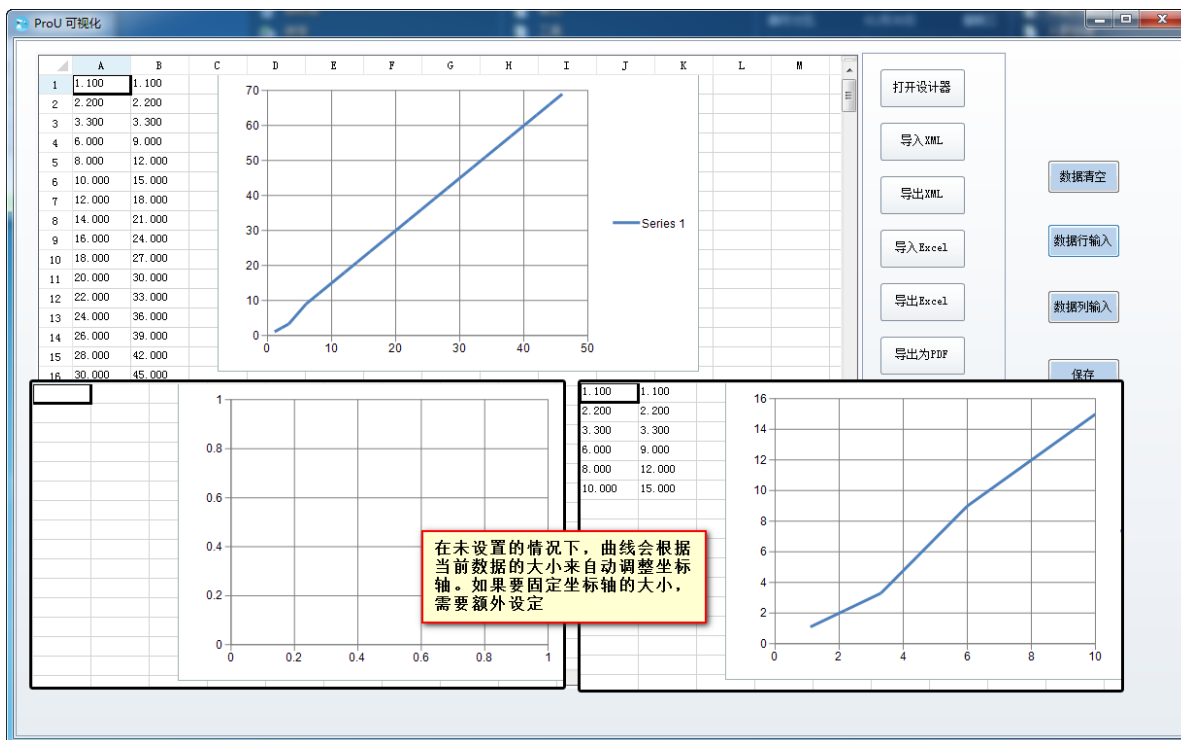


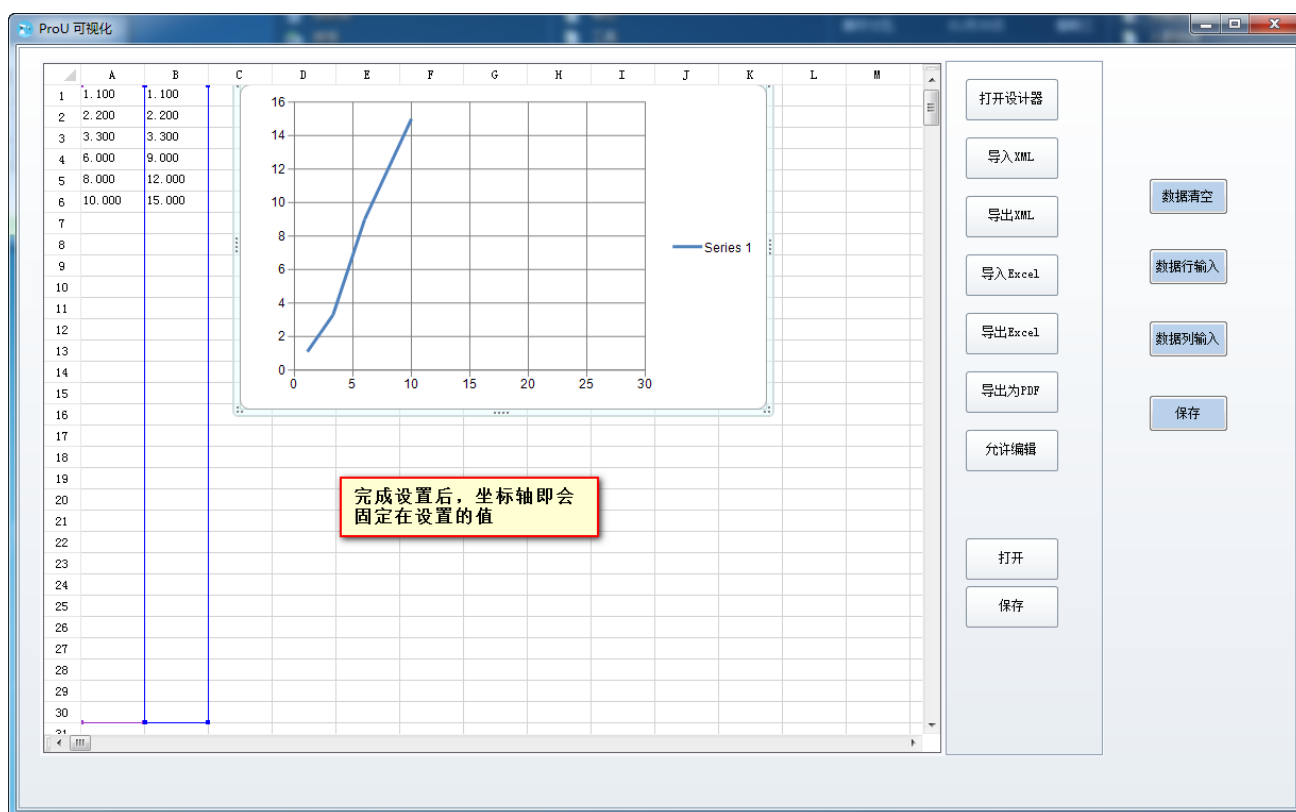
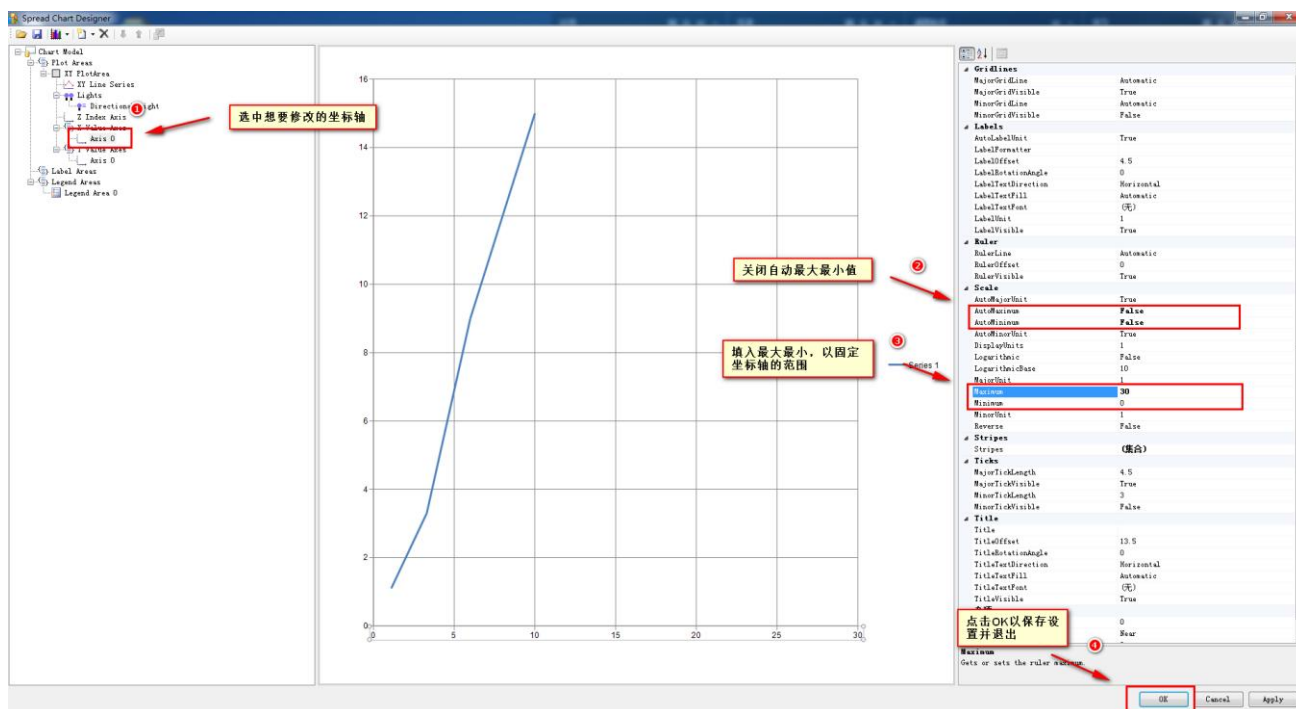




6.2 曲线的参数设置

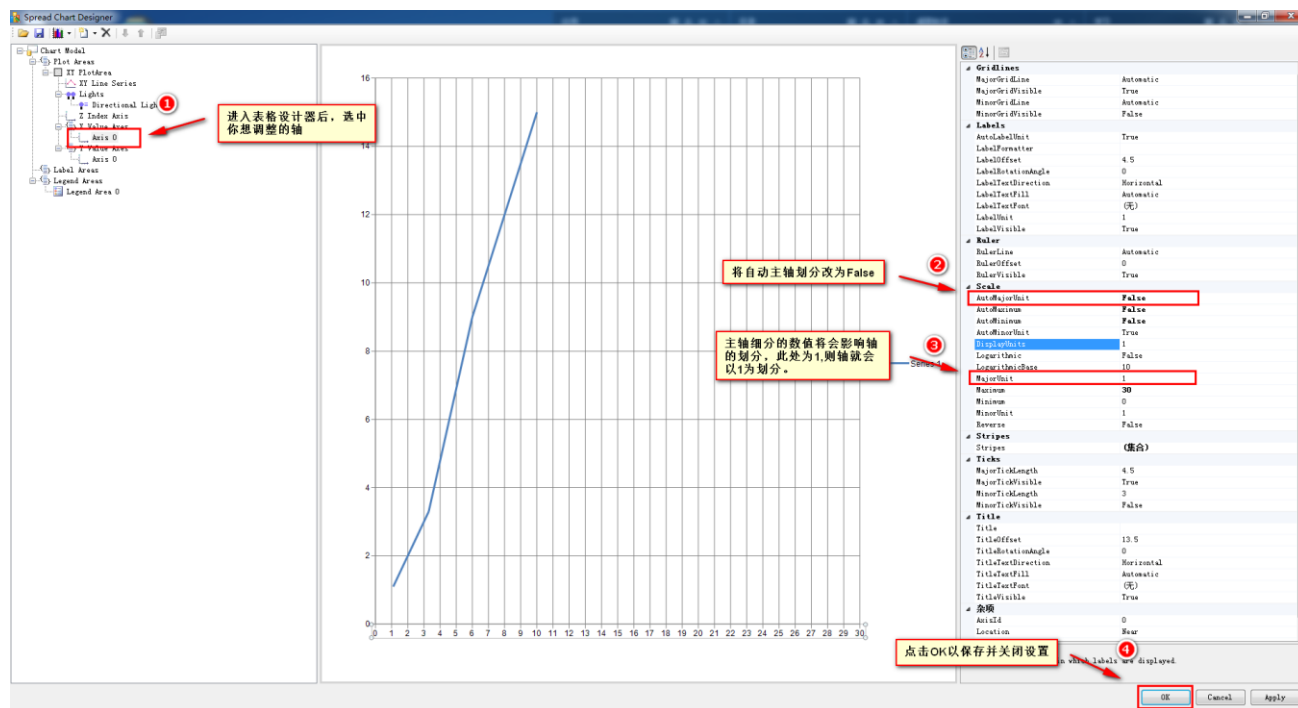
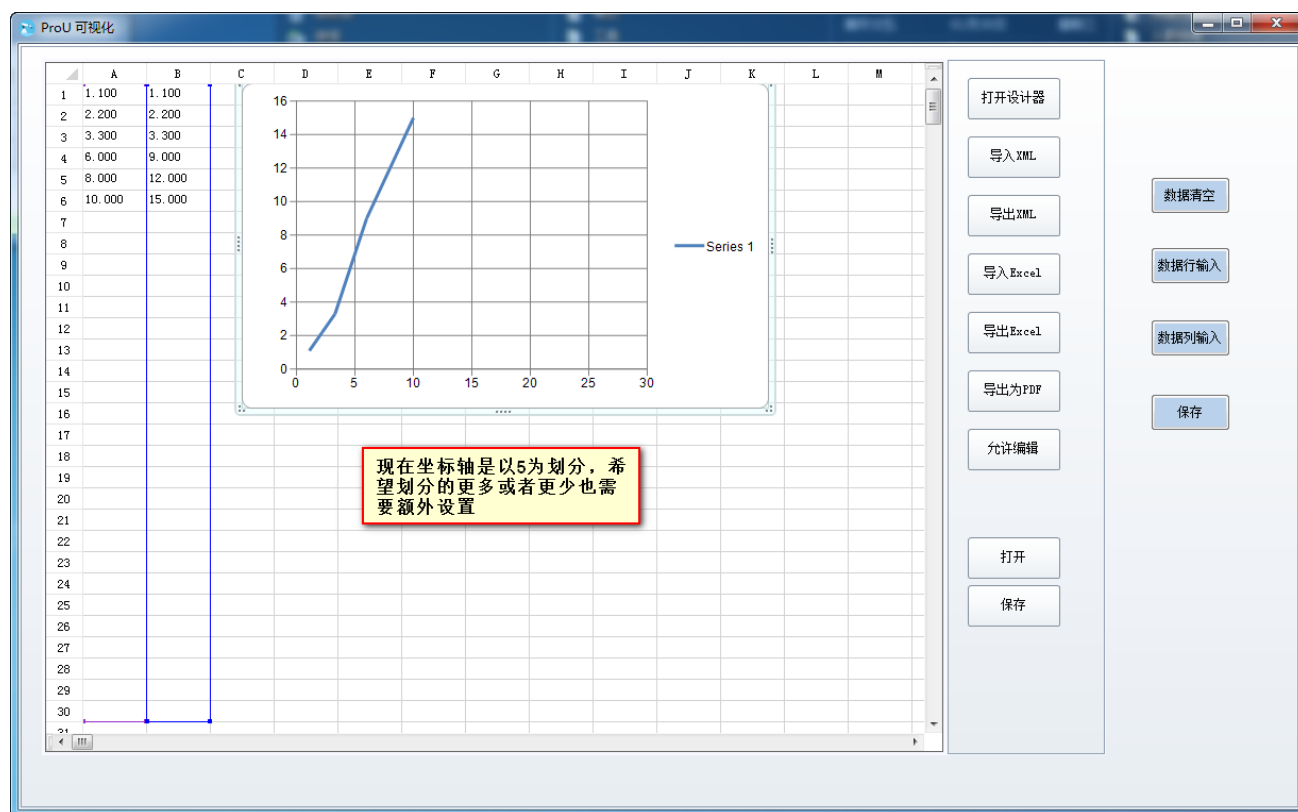
6.2.1 坐标轴的固定







6.2.2 坐标轴的划分





6.2.3 坐标轴标题

Spread Chart Designer

Chart Model

- Plot Area
 - XY PlotArea
 - XY Line Series
 - XY Line Axis
 - Axis 0**
 - Axis 1
 - Axis 2
 - Label Area
 - Legend Area

选中你想修改的坐标轴

如果想为坐标轴添加信息, 诸如时、分、秒、牛、千克等信息也可额外设置

Axis 0

Axis 1

Axis 2

Label Area

Legend Area

Y轴

X轴

Title: 文字内容
TitleOffset: 文字与轴的距离
TitleTextFont: 文字字体、字号的设置

AxisLabelUnit

点击OK以关闭并保存设置

OK Cancel Apply

Gridlines

- MajorGridLine: Automatic
- MajorGridVisible: True
- MinorGridLine: Automatic
- MinorGridVisible: False

Labels

- AxisLabelUnit: True
- LabelFormatter: True
- LabelOffset: 4.5
- LabelRotationAngle: 0
- LabelTextDirection: Horizontal
- LabelTextFill: Automatic
- LabelTextFont: 1
- LabelTextSize: 1
- LabelTextVisible: True

Rules

- RuleLine: Automatic
- RuleOffset: 0
- RuleVisible: True

Scalings

- AutoMajorUnit: False
- AutoMinorUnit: False
- AutoMinorUnit: True
- DisplayUnits: 1
- Logarithmic: False
- LogarithmicBase: 10
- MajorUnit: 3
- Maximum: 30
- Minimum: 0
- MinorUnit: 1
- Reverse: False

Stripes

- Stripes: (空白)

Ticks

- MajorTickLength: 4.5
- MajorTickVisible: True
- MinorTickLength: 3
- MinorTickVisible: False

Title

- Title: X轴
- TitleOffset: 20
- TitleRotationAngle: 0
- TitleTextDirection: Horizontal
- TitleTextFill: Automatic
- TitleTextFont: 宋体, 10pt
- TitleTextSize: 1
- TitleTextVisible: True

AxisLabelUnit

The interval for tick each labels is automatically generated.

ProU 可视化

数据表

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	1.100	1.100											
2	2.200	2.200											
3	3.300	3.300											
4	6.000	9.000											
5	8.000	12.000											
6	10.000	15.000											
7													
8													
9													
10													
11													
12													
13													
14													
15													
16													
17													
18													
19													
20													
21													
22													
23													
24													
25													
26													
27													
28													
29													
30													

Y轴

X轴

打开设计器

导入XML

导出XML

导入Excel

导出Excel

导出为PDF

允许编辑

打开

保存

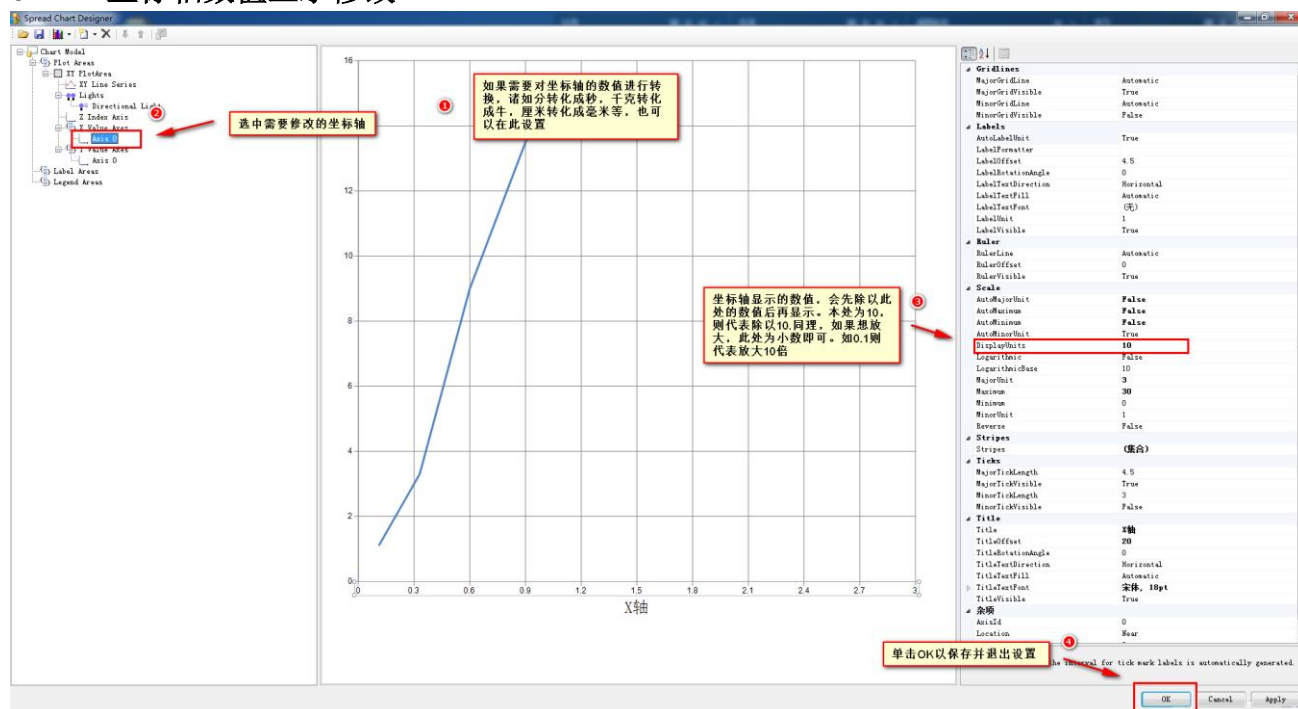
数据清空

数据行输入

数据列输入

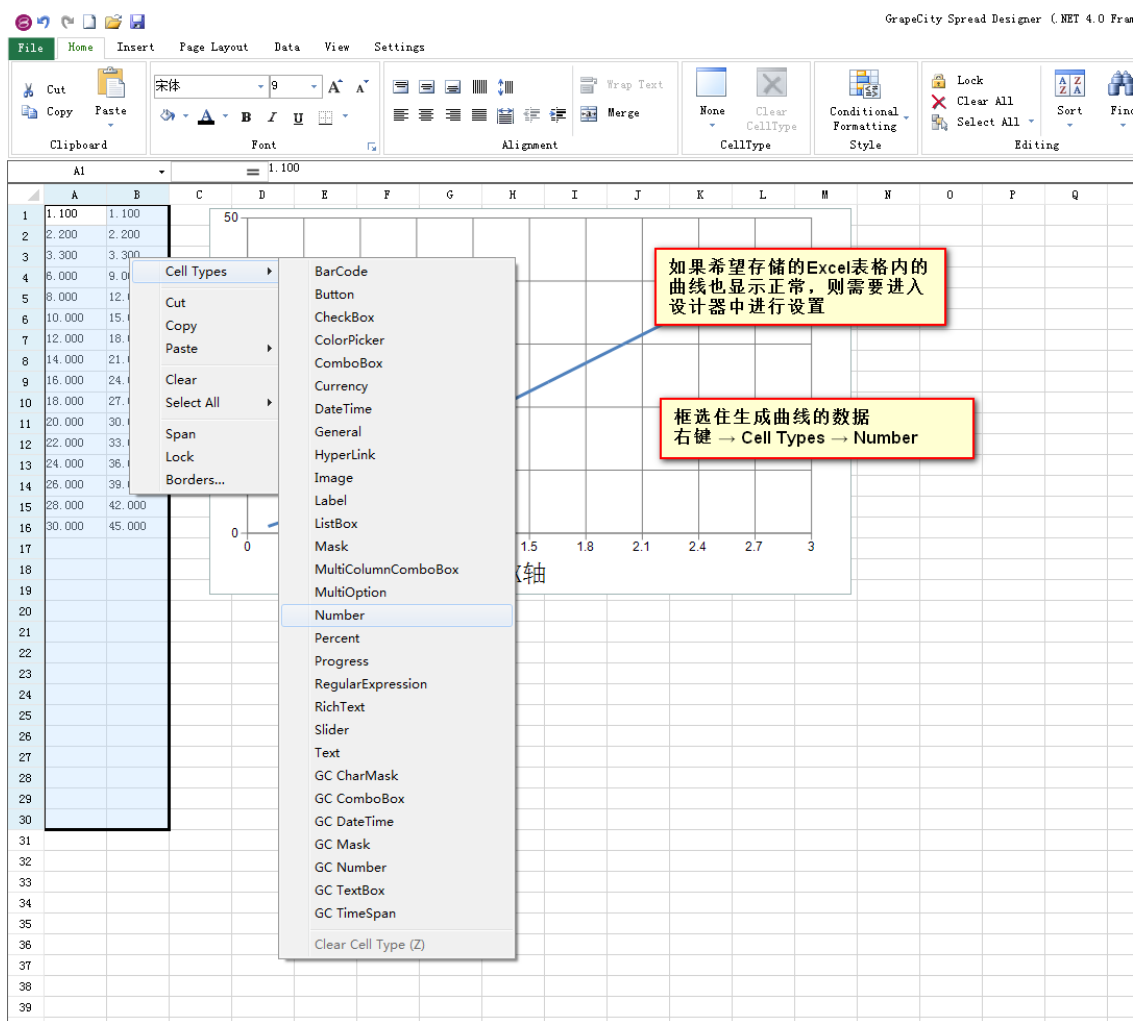
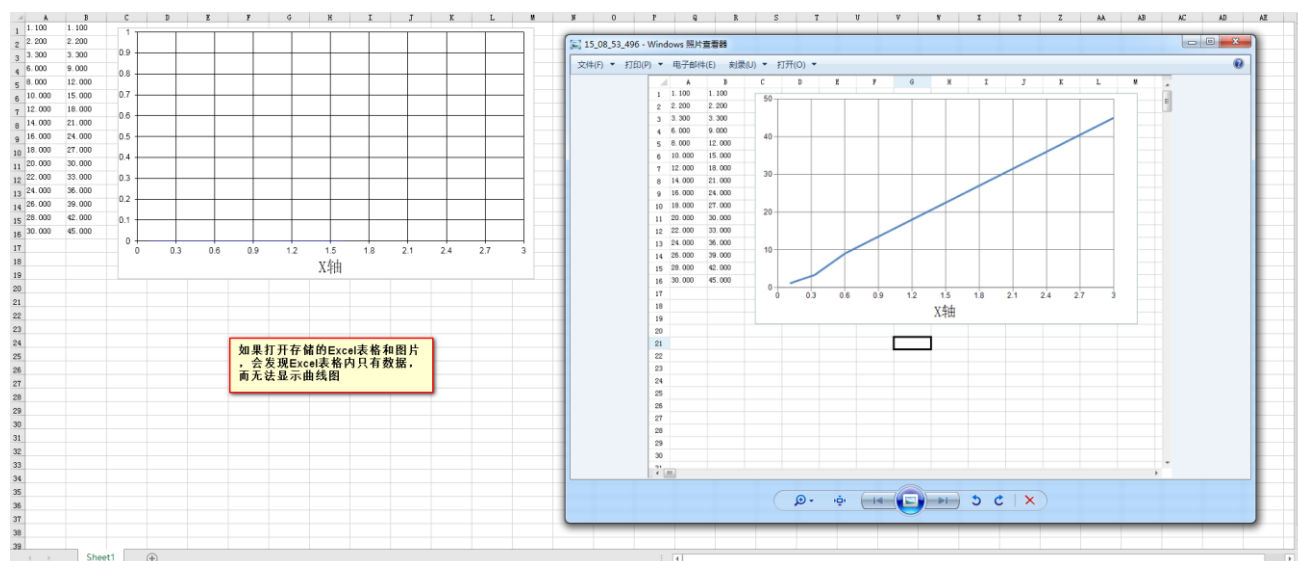
保存

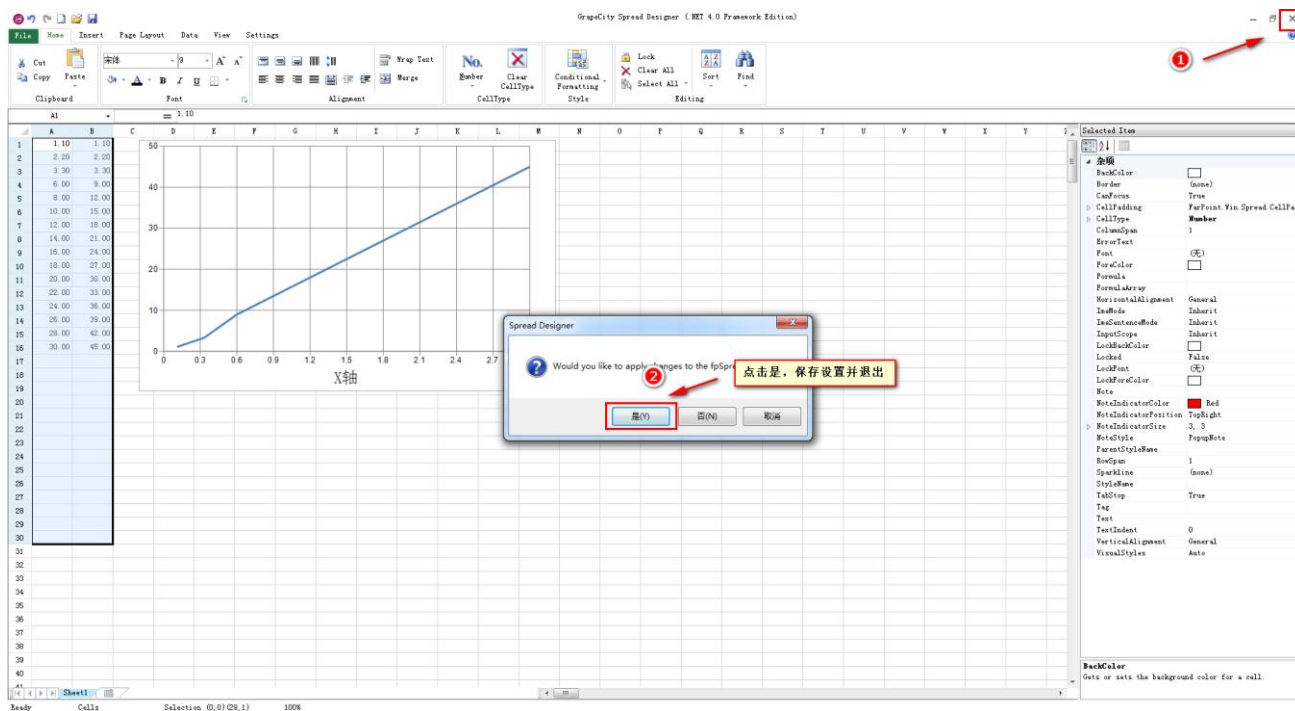
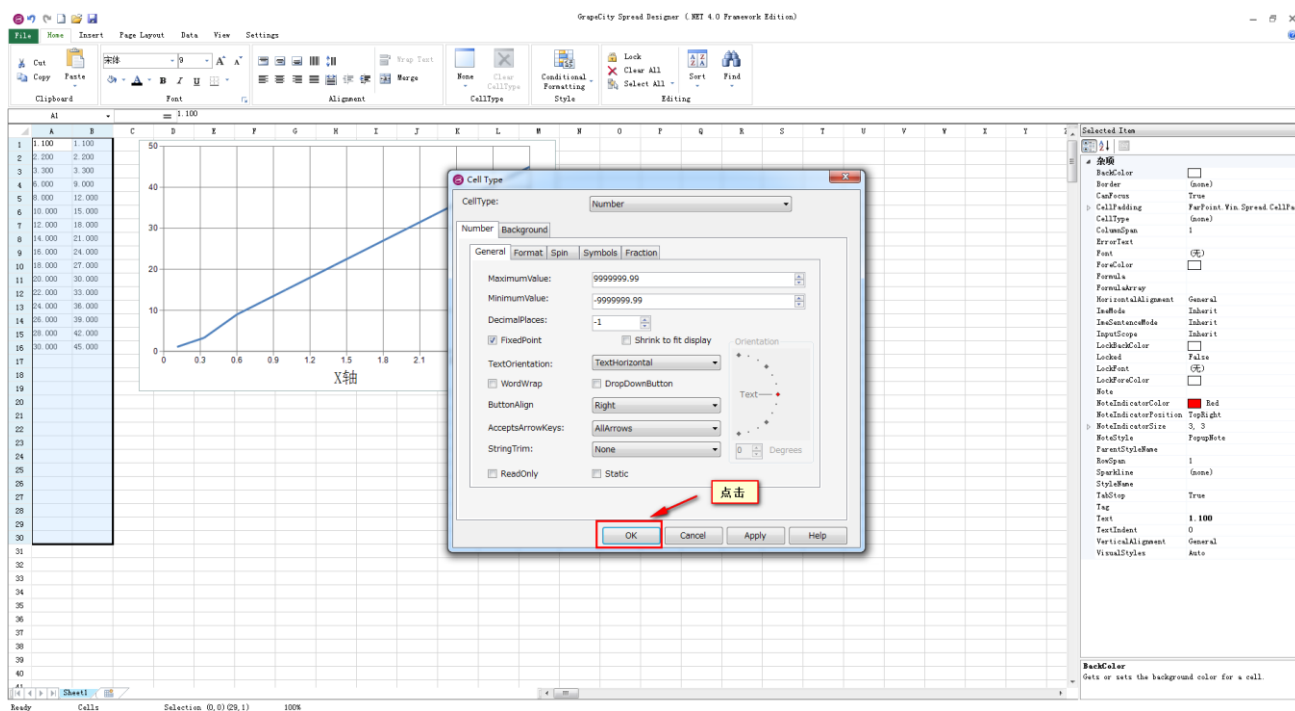
6.2.4 坐标轴数值显示修改

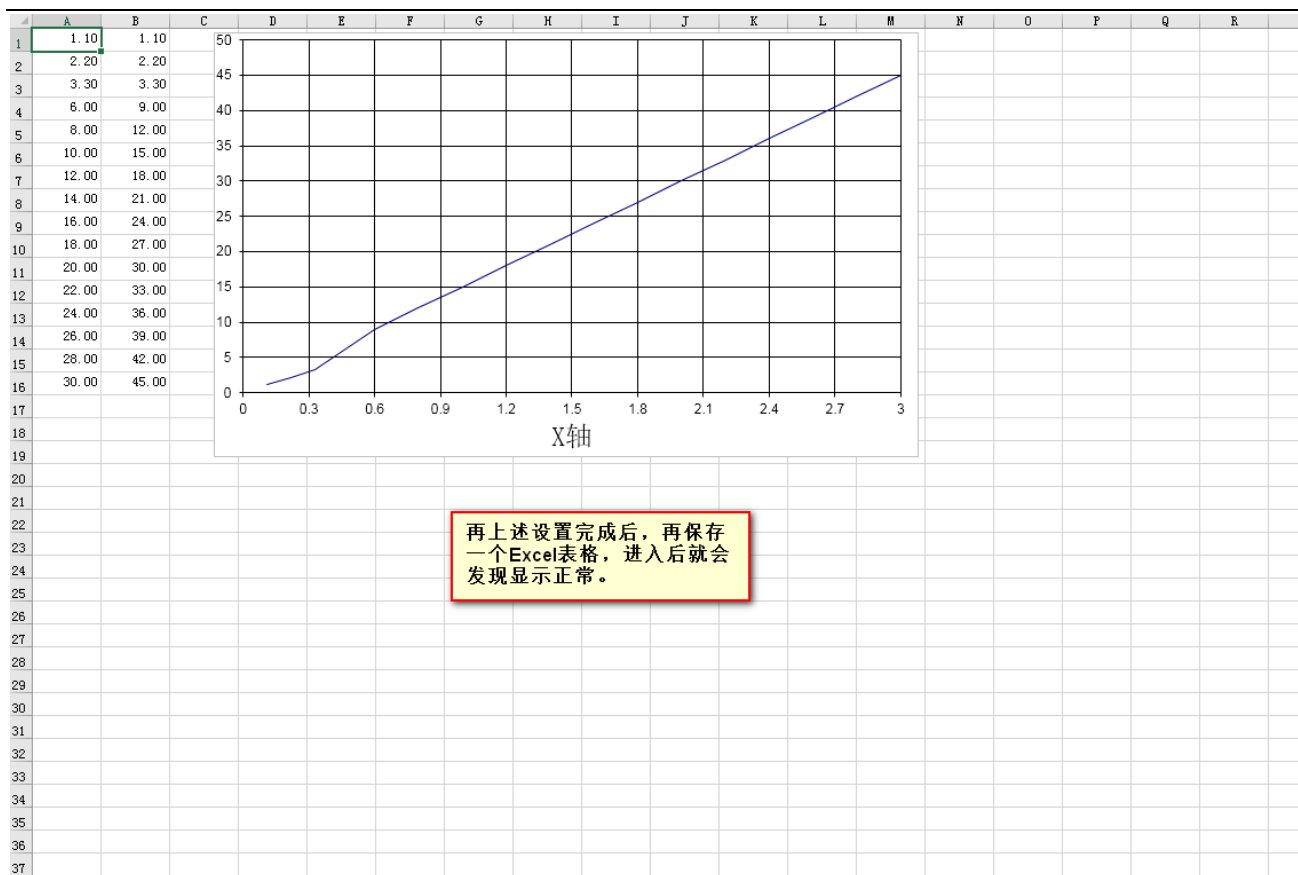




6.2.5 存储的 Excel 文件曲线不显示设置







以上便是 OSC 控件的使用及 Spread Designer 常用的一些设置。

PS: 在所有设置完成之后，记住一定要点击 OSC 控件上的保存按钮