

GRAPH 控件

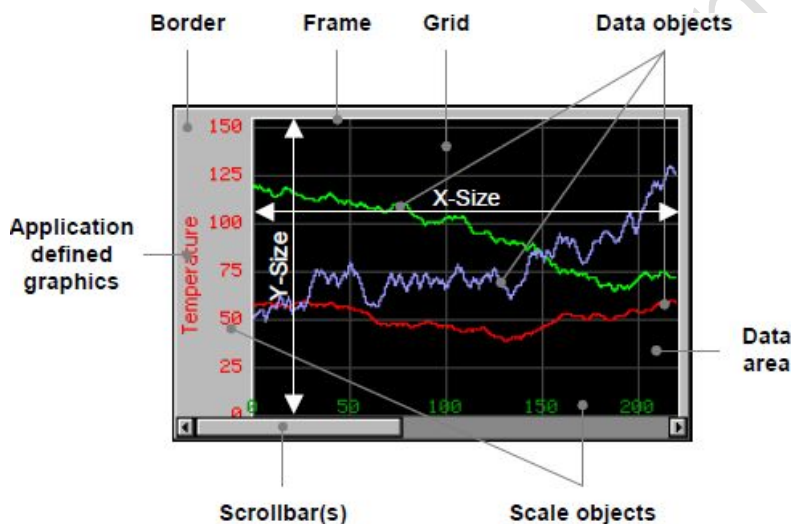
GRAPH 控件可以用来显示图形化的数据。它的典型应用是显示的测量值或曲线图。可以同时显示多条曲线。可用水平和垂直刻度可以用来标记曲线。在背景上可以显示一个具有不同的水平和垂直网格间距的网格，如果数组值超出了控件的可见范围之内，该控件可以自动显示滚动条，通过滚动条可以查看到更多的数据。

1. GRAPH 控件的结构

一个 GRAPH 控件包含不同类型的对象

- 控件可附加自身的数据对象和比例尺对象
- 可选一个或多个数据对象
- 可选一个或多个比例尺对象

下图显示了 GRAPH 控件的详细结构



下表是对上图的详细说明

构件	描述
Border	可选的边框是 GRAPH 控件的一部分
Frame	环绕数据区的细线，GRAPH 控件的一部分
Grid	显示在数据区的背景，GRAPH 控件的一部分
Data area	一个区域，网格和数据对象可以显示在里面
Data object(s)	对于每一条曲线，可以给控件增加一个数据对象
Application defined graphic	一个定义回调函数的应用可以用来绘制任何应用程序定义的文本和/或图形。
Scrollbar(s)	当数据对象的范围超出了 GRAPH 控件的数据区域，控件可以自动显示水平和/或垂直的滚动条
Scale object(s)	可以附加在 GRAPH 控件的水平和垂直刻度
X-Size	数据区域的水平范围
Y-Size	数据区域的垂直范围

2. 创建和删除一个 GRAPH 控件

GRAPH 控件的创建应按如下步骤

- 创建 GRAPH 控件，并设置所需的属性
- 创建数据对象
- 将数据对象关联到 GRAPH 控件
- 创建可选的刻度尺对象
- 将刻度尺对象关联到 GRAPH 控件

数据对象和刻度尺对象一旦关联到 GRAPH 控件，不需要在应用中删除他们，GRAPH 控件会自动完成。

举例

下面的程序显示了如何创建和删除一个 GRAPH 控件

```
GRAPH_DATA_Handle hData;
GRAPH_SCALE_Handle hScale;
WM_HWIN hGraph;
hGraph = GRAPH_CreateEx(10, 10, 216, 106, WM_HBKWIN, WM_CF_SHOW, 0, GUI_ID_GRAPH0);
hData = GRAPH_DATA_YT_Create(GUI_DARKGREEN, NumDataItems, aData0, MaxNumDataItems);
GRAPH_AttachData(hGraph, hData);
hScale = GRAPH_SCALE_Create(28, GUI_TA_RIGHT, GRAPH_SCALE_CF_VERTICAL, 20);
GRAPH_AttachScale(hGraph, hScale);
/*
 Do somthing with the widget...
*/
WM_DeleteWindow(hGraph);
```

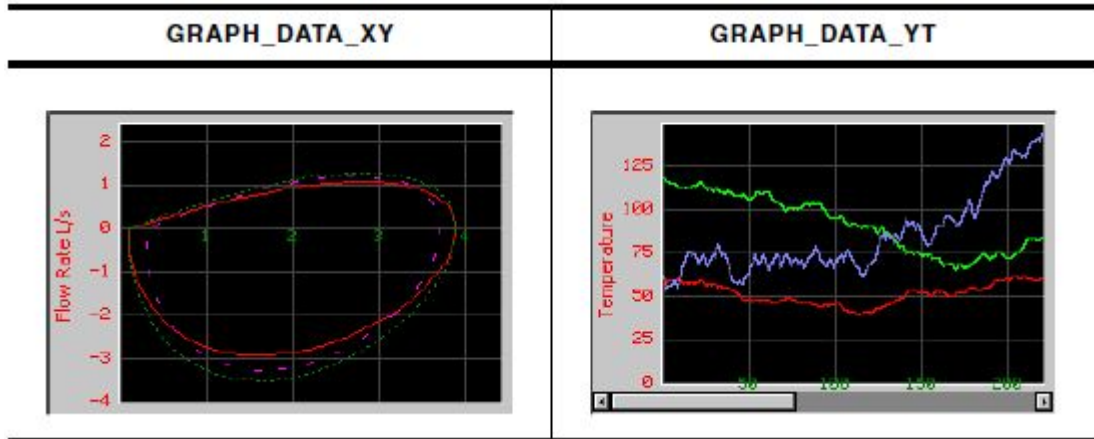
3. 绘图过程

如上所述 GRAPH 控件包含不同的对象，下面解释该控件绘图的过程

- 使用背景色填充背景
- 调用可选的回调函数，这使它可以绘制诸如用户定义的网格
- 绘制网格（如果使能）
- 绘制数据对象和边框区域
- 绘制刻度尺对象
- 调用可选的回调函数，这使它可以绘制诸如用户定义的刻度或一些附加文字和/或图形

4. 波形支持的类型

对于显示连续更新的测量值的要求，不同于显示一个带有 X/Y 坐标的函数波形的要求，因此该控件目前支持两种类型的数据对象，如下图



GRAPH_DATA_XY

数据对象用来显示包含一系列点组成的曲线，目标数据被绘制成一个多线段，一个典型应用是绘制函数波形。

GRAPH_DATA_YT

数据对象用来显示对于每一个 X 位置都带有一个 Y 值的波形，使用该数据对象的典型应用绘制连续更新的测量值波形。

5. 配置选项

5.1 GRAPH 控件

类型	宏	缺省值	说明
N	GRAPH_BKCOLOR_DEFAULT	GUI_BLACK	数据区域的缺省背景色
N	GRAPH_BORDERCOLOR_DEFAULT	0xC0C0C0	边框的缺省背景色
N	GRAPH_FRAMECOLOR_DEFAULT	GUI_WHITE	框体线的缺省颜色
N	GRAPH_GRIDCOLOR_DEFAULT	GUI_DARKGRAY	网格的缺省颜色
N	GRAPH_GRIDSPACING_X_DEFAULT	50	网格的水平缺省间隔
N	GRAPH_GRIDSPACING_Y_DEFAULT	50	网格的垂直缺省间隔
N	GRAPH_BORDER_L_DEFAULT	0	左边框的缺省大小
N	GRAPH_BORDER_T_DEFAULT	0	上边框的缺省大小
N	GRAPH_BORDER_R_DEFAULT	0	右边框的缺省大小
N	GRAPH_BORDER_B_DEFAULT	0	下边框的缺省大小

5.2 刻度尺对象

类型	宏	缺省值	说明
----	---	-----	----

N	GRAPH_SCALE_TEXTCOLOR_DEFAULT	GUI_WHITE	缺省文字颜色
S	GRAPH_SCALE_FONT_DEFAULT	&GUI_Font6x8	缺省字体

6. 按键响应

该控件不能获得输入的焦点，也不能对按键输入做出响应

7. GRAPH API

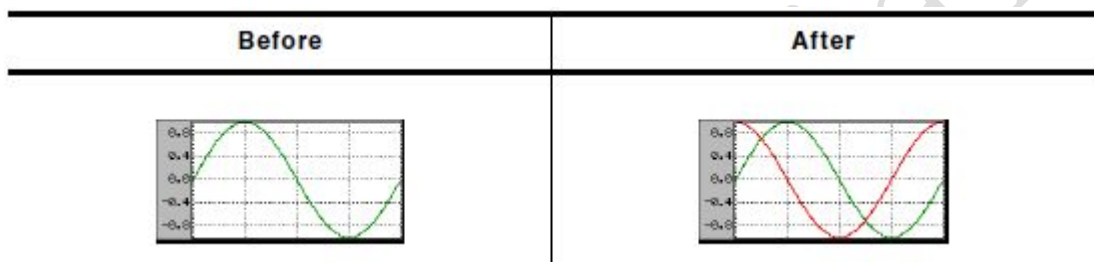
下表按字母顺序列出可用的 uC/GUI 相关的函数，详细描述如下

函数	说明
公共函数	
GRAPH_AttachData()	将数据对象附加到一个现有的 GRAPH 控件
GRAPH_AttachScale()	将刻度尺对象附加到一个现有的 GRAPH 控件
GRAPH_CreateEx()	创建一个 GRAPH 控件
GRAPH_SetBorder()	设置边框（上、下、左、右）的尺寸
GRAPH_SetColor()	设置 GRAPH 控件的颜色
GRAPH_SetGridDistX()	设置水平网格间距
GRAPH_SetGridDistY()	设置垂直网格间距
GRAPH_SetGridFixedX()	在 X 轴上对网格修正
GRAPH_SetGridVis()	使能网格绘制
GRAPH_SetLineStyleH()	设置水平网格线的线型
GRAPH_SetLineStyleV()	设置垂直网格线的线型
GRAPH_SetVSizeX()	设置 GRAPH 控件的水平范围
GRAPH_SetVSizeY()	设置 GRAPH 控件的垂直范围
GRAPH_SetUserDraw()	设置用户回调函数
GRAPH_DATA_YT 相关函数	
GRAPH_DATA_YT_AddValue()	向数据对象增加一个数据项
GRAPH_DATA_YT_Create()	创建一个 GRAPH_DATA_YT 对象
GRAPH_DATA_YT_SetOffy()	设置数据绘制的垂直偏移量
GRAPH_DATA_XY 相关函数	
GRAPH_DATA_XY_AddPoint()	增加一个数据对象指针
GRAPH_DATA_XY_Create()	创建一个 GRAPH_DATA_XY 对象
GRAPH_DATA_XY_SetLineStyle()	设置用来绘制数据的线型
GRAPH_DATA_XY_SetOffx()	设置绘制数据的水平偏移量
GRAPH_DATA_XY_SetOffy()	设置绘制数据的垂直偏移量
GRAPH_DATA_XY_SetPenSize()	设置绘制数据的笔的粗细
刻度尺相关函数	
GRAPH_SCALE_Create()	创建一个刻度尺对象
GRAPH_SCALE_SetFactor()	设置一个计算因子，用于像素转换到所需单位的计算
GRAPH_SCALE_SetFixed()	避免滚动刻度
GRAPH_SCALE_SetFont()	设置显示数字的字体

GRAPH_SCALE_SetNumDecs()	设置小数部分的位数
GRAPH_SCALE_SetOfff()	设置对数量增加的一个可选的偏移量
GRAPH_SCALE_SetPos()	设置刻度水平或垂直的位置
GRAPH_SCALE_SetTextColor()	设置刻度的文字颜色
GRAPH_SCALE_SetTickDist()	设置刻度之间的距离（像素）

7.1 公共函数

GRAPH_AttachData()



描述

向现有的一个 GRAPH 控件附加一个数据对象

原型

```
void GRAPH_AddGraph(GRAPH_Handle hObj, GRAPH_DATA_Handle hData);
```

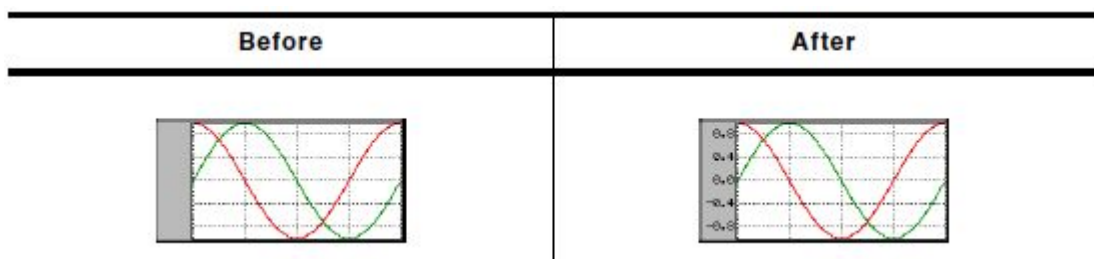
参数	含义
hObj	控件的句柄
hData	要增加到控件的数据对象句柄，数据对象应使用 GRAPH_DATA_YT_Create() 或 GRAPH_DATA_XY_Create() 函数创建

附加信息

一旦附加到一个 GRAPH 控件，应用程序不需要破坏数据对象，GRAPH 控件在自身被删除的时候会删除所有的附加数据对象，

关于如何创建数据对象的细节，请参考函数 GRAPH_DATA_YT_Create() 和 GRAPH_DATA_XY_Create()。

GRAPH_AttchScale()



描述

向现有的一个 GRAPH 控件附加一个刻度尺对象

原型

```
void GRAPH_AttchScale(GRAPH_Handle hObj, GRAPH_SCALE_Handle hScale);
```

参数	含义
hObj	控件的句柄
hScale	要增加的刻度尺句柄

附加信息

一旦附加到一个 GRAPH 控件，应用程序不需要破坏刻度尺对象，GRAPH 控件在自己被删除的时候会删除所有的附加数据对象，

关于如何创建刻度尺对象的细节，请参考函数 GRAPH_SCALE_Create()。

GRAPH_CreateEx()**描述**

创建一个新的 GRAPH 控件

原型

```
GRAPH_Handle GRAPH_CreateEx(int x0, int y0, int xsize, int ysize, WM_HWIN hParent, int WinFlags, int ExFlags, int Id);
```

参数	含义
X0	控件最左边的像素（相对父坐标）
Y0	控件最上边的像素（相对父坐标）
xsize	控件的水平宽度（像素）
ysize	控件的垂直高度（像素）
hParent	父窗口的句柄，如果为 0，则作为桌面（最顶层窗口）的子窗口
winFlags	窗口创建标志，典型值为 WM_CF_SHOW 使得控件立即可见，取值范围可参考窗口管理相关内容
ExFlags	见下面的表格
Id	控件的窗口 Id

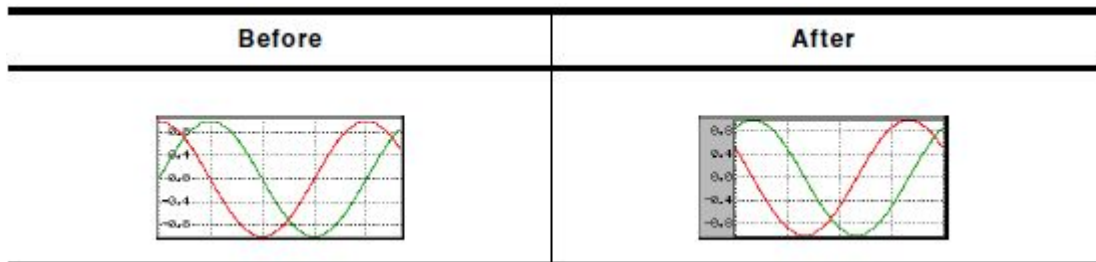
参数 x 允许的取值

GRAPH_CF_GRID_FIXED_X	在 x 轴上修复网格，即如果使用的水平滚动条，网格会保持在它的位置
-----------------------	-----------------------------------

返回值

所创建控件的句柄，如果创建失败，返回 0

GRAPH_SetBorder()



描述

设置给定的 GRAPH 控件的左、右、上、下边框

原型

```
void GRAPH_SetBorder(GRAPH_Handle hObj, unsigned BorderL, unsigned BorderT, unsigned BorderR, unsigned BorderB);
```

参数	含义
hObj	控件的句柄
BorderL	左边框大小像素值
BorderT	上边框大小像素值
BorderR	右边框大小像素值
BorderB	下边框大小像素值

附加信息

边框的大小是控件和数据区之间的像素值，如果边框大小至少有一个像素值，围绕着数据区的框架、细线是唯一可见的，有关如何设置边框颜色和瘦框架参考函数 GRAPH_SetColor()。

GRAPH_SetColor()



描述

给 GRAPH 控件设置所需的颜色

原型

```
GUI_COLOR GRAPH_SetColor(GRAPH_Handle hObj, GUI_COLOR Color, unsigned Index);
```

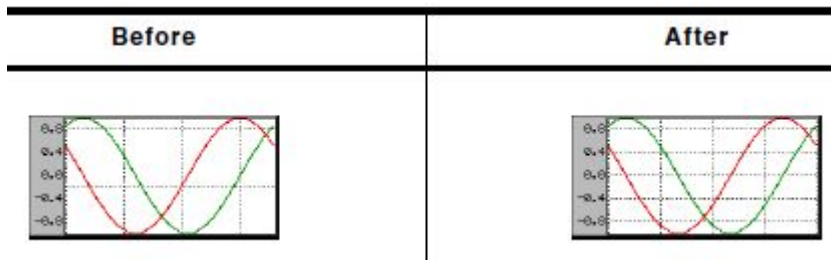
参数	含义
hObj	控件的句柄
Color	控件所需的颜色
Index	见下表

参数 Index 允许的取值	
GRAPH_CI_BK	设置背景色
GRAPH_CI_BOERDER	设置边框区域的颜色
GRAPH_CI_FRAME	设置细框架线的颜色
GRAPH_CI_GRID	设置网格的颜色

返回值

返回所需条目之前的颜色

GRAPH_SetGridDistX(),GRAPH_SetGridDistY()



描述

设置网格线之间的距离

原型

unsigned GRAPH_SetGridDistX(GRAPH_Handle hObj, unsigned Value);

unsigned GRAPH_SetGridDistY(GRAPH_Handle hObj, unsigned Value)

参数	含义
hObj	控件的句柄
Value	网格线之间的距离，缺省为 50 像素

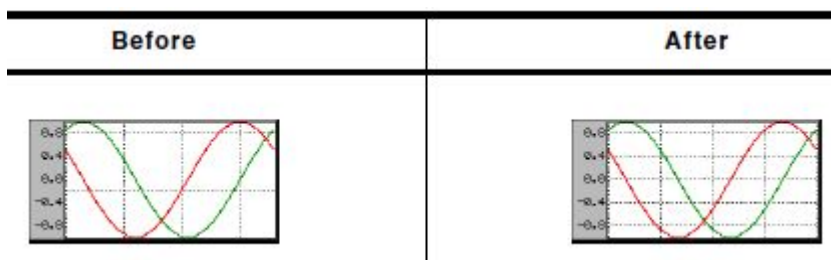
返回值

返回之前的线间距

附加信息

第一条竖线画在数据区最左边的位置，水平线画在数据区最底部的位置，除非使用了偏移量

GRAPH_SetGridFixedX ()



描述

<以上所有信息均为中兴通讯股份有限公司所有，不得外传>

All Rights reserved, No Spreading abroad without Permission of ZTE

对 X 轴网格进行修正

原型

```
unsigned GRAPH_SetGridFixedX(GRAPH_Handle hObj, unsigned OnOff);
```

参数	含义
hObj	控件的句柄
onoff	1=在 X 轴上对网格修正, 0=不处理 (缺省)

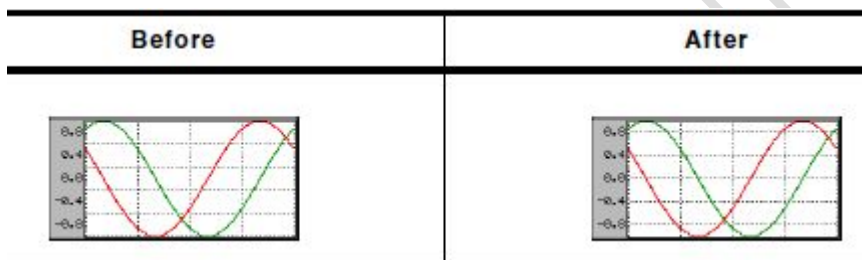
返回值

之前使用的值

附加信息

在某些情况下, 在 X 坐标上修正网格是很有用的, 一个典型的应用实在 YT-图形, 即连续的新值不断增加需要水平滚动条, 这种情况下可以通过修正背景里的网格来解决, 有关如何激活滚动条, 请参考函数 GRAPH_SetVSizeX() and GRAPH_SetVSizeY()。

GRAPH_SetGridOffY ()



描述

增加一个偏移量用于显示水平方向的网格线

原型

```
unsigned GRAPH_SetGridOffY(GRAPH_Handle hObj, unsigned Value);
```

参数	含义
hObj	控件的句柄
Value	偏移量

返回值

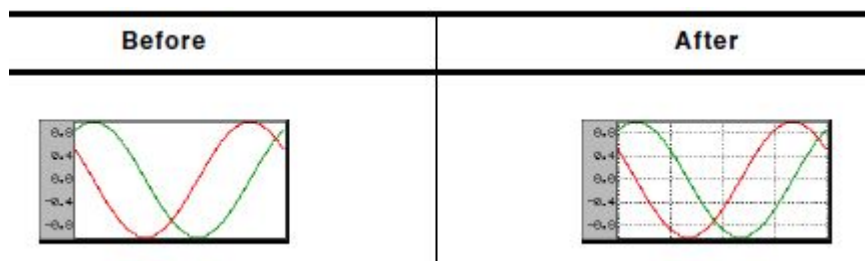
返回之前的偏移量

附加信息

渲染网格时, 控件从数据区底部开始使用当前的间距画网格线, 当纵坐标的零点位于 Y 轴的中间时, 可能会出现中间没有网格线的情况, 此时可以用此函数增加一个偏移量来进行 Y 轴上的转移, 正值向下转移, 负值向上转移。

有关网格间距的设置可参考函数 GRAPH_SetGridSpacingX() 和 GRAPH_SetGridSpacingY。

GRAPH_SetGridVis ()



描述

设置网格线为可见

原型

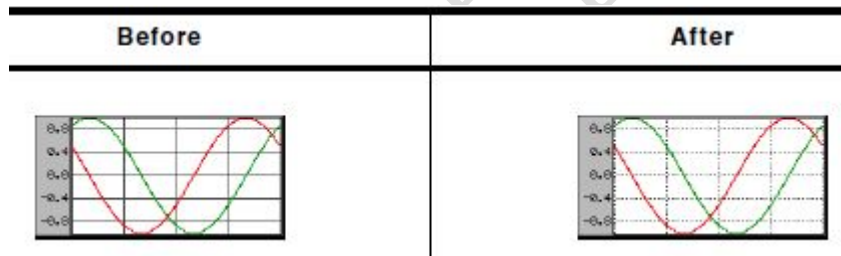
```
unsigned GRAPH_SetGridVis(GRAPH_Handle hObj, unsigned OnOff);
```

参数	含义
hObj	控件的句柄
OnOff	1=网格可见, 0=不可见 (缺省)

返回值

返回之前的可见值

GRAPH_SetLineStyleH (), GRAPH_SetLineStyleV ()



描述

设置水平和垂直网格线的线型

原型

```
U8 GRAPH_SetLineStyleH(GRAPH_Handle hObj, U8 LineStyle);
```

```
U8 GRAPH_SetLineStyleV(GRAPH_Handle hObj, U8 LineStyle);
```

参数	含义
hObj	控件的句柄
LineStyle	使用的线型, 查看支持的线型可参考函数 GUI_SetLineStyle(), 默认为 GUI_LS_SOLID

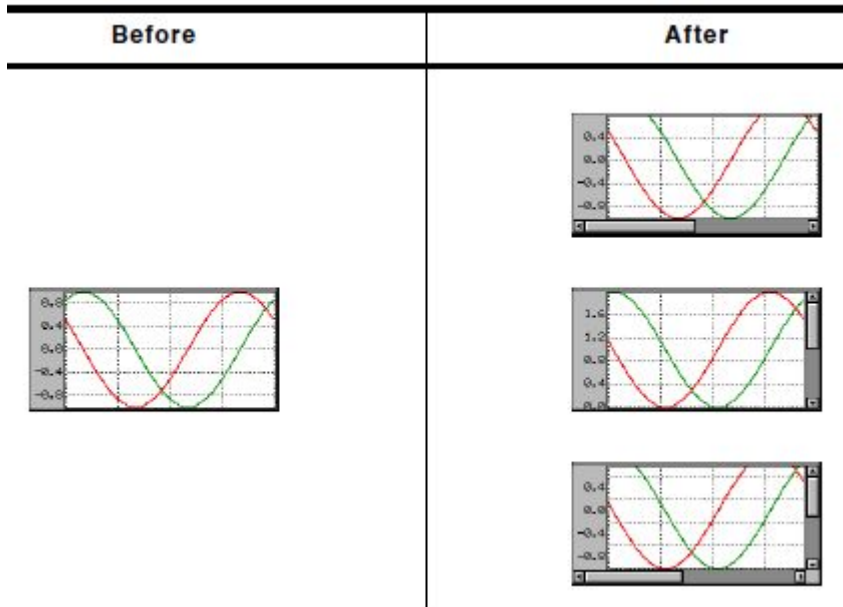
返回值

返回之前的线型

附加信息

注意: 使用 GUI_LS_SOLID 之外的其他线型, 绘制网格时需要花费更多的时间。

GRAPH_SetVSizeX (), GRAPH_SetVSizeY()



描述

设置 X 轴和 Y 轴的虚拟区域大小

原型

unsigned GRAPH_SetVSizeX(GRAPH_Handle hObj, unsigned Value);

unsigned GRAPH_SetVSizeY(GRAPH_Handle hObj, unsigned Value);

参数	含义
hObj	控件的句柄
Value	X 轴和 Y 轴的虚拟大小

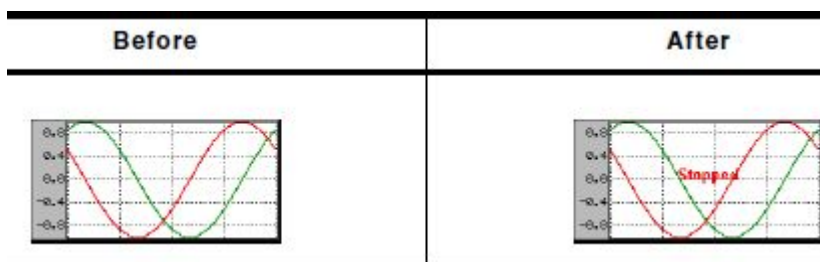
返回值

返回之前 X 轴或 Y 轴虚拟大小

附加信息

如果部件虚拟尺寸大于数据区的可视尺寸，部件自动显示滚动条。例如如果通过函数 GRAPH_DATA_YT_Create() 创建一个数据对象，包含的数据量超出数据区域显示范围，可使用 GRAPH_SetVSizeX() 可以来使能滚动条，函数 GRAPH_SetVSizeX(NumDataItems) 使能滚动条，提供了超出可视区域的 X 轴上的数据量。

GRAPH_SetUserDraw()



描述

设置用户绘图函数，此函数可以在绘图过程中被控件调用，绘制出用户定义的数据。

原型

```
void GRAPH_SetUserDraw(GRAPH_Handle hObj, void (* pUserDraw) (WM_HWIN hObj, int Stage));
```

参数	含义
hObj	控件的句柄
pUserDraw	指向函数的指针，给函数在绘图过程中被控件调用

参数 Stage 允许的取值	
GRAPH_DRAW_FIRST	在填充背景色后调用，如绘制用户定义的网格
GRAPH_DRAW_LAST	在绘制波形之后调用，如用户定义的数据

附加信息

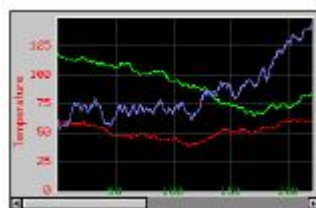
用户绘图函数在填充背景色之后被调用，以及绘制完所有的波形之后被调用，首次调用的裁剪区域被限制在数据区域内，最后调用被显示在除有效框架以外的全部图形控件区域。

举例

```
static void _UserDraw(WM_HWIN hWin, int Stage) {
switch (Stage) {
case GRAPH_DRAW_FIRST:
/* Draw for example a user defined grid... */
break;
case GRAPH_DRAW_LAST:
/* Draw for example a user defined scale or additional text... */
break;
}
}

static void _CreateGraph(void) {
WM_HWIN hGraph;
hGraph = GRAPH_CreateEx(10, 10, 216, 106, WM_HBKWIN, WM_CF_SHOW, 0, GUI_ID_GRAPHO);
GRAPH_SetUserDraw(hGraph, _UserDraw); /* Enable user draw */
...
}
```

7.2 GRAPH_DATA_YT 相关函数

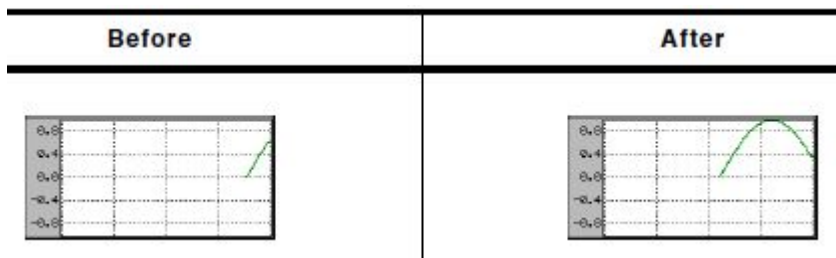


<以上所有信息均为中兴通讯股份有限公司所有，不得外传>

第 12 页

All Rights reserved, No Spreading abroad without Permission of ZTE

GRAPH_DATA_YT_AddValue()



描述

向 GRAPH_DATA_YT 对象增加新的数据

原型

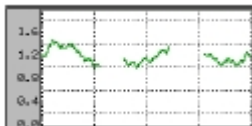
```
void GRAPH_DATA_YT_AddValue(GRAPH_DATA_Handle hDataObj, I16 Value);
```

参数	含义
hDataObj	数据对象的句柄
Value	增加到数据对象的值

附加信息

给定的值被添加到数据对象，如果数据对象是满的，意味着他包含了创建时在参数 MaxNumItems 中定义的一样多的数据，在增加一个新值前先转移数据，因此向一个满的对象增加一个新数据的时候，第一条数据就被转移出去了。

值 0x7FFF 用来处理无效的数据，这些值在绘图时被排除在外，下面的截图显示了带有 2 个间隙的无效数据的波形。



GRAPH_DATA_YT_Create()

描述

创建一个 GRAPH_DATA_YT 类型的数据对象

原型

```
GRAPH_DATA_Handle GRAPH_DATA_YT_Create(GUI_COLOR Color, unsigned MaxNumItems, I16 * pData, unsigned NumItems);
```

参数	含义
Color	绘制数据的颜色
MaxNumItems	数据的最大数量
pData	指向添加到对象的数据的指针，该指针应该指向一个 I16 值的数组
NumItems	被添加的数据数量

返回值

创建成功返回数据对象的句柄，否则返回 0

附加信息

最后的数据显示在数据区域的最右列，如果一个数据对象包含比能够显示在数据区域更多的数据，可以调用 GRAPH_SetVSizeX() 来显示一个滚动条，是的有可能在更大的数据

<以上所有信息均为中兴通讯股份有限公司所有，不得外传>

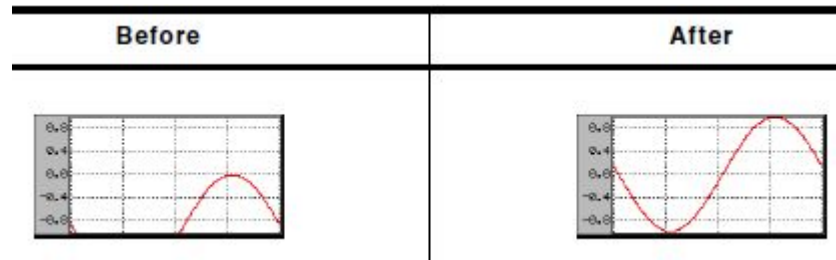
第 13 页

All Rights reserved, No Spreading abroad without Permission of ZTE

对象滚动。

一旦关联到一个 GRAPH 控件，数据对象不需要被应用删除，当删除 GRAPH 控件的时候会被自动删除。

GRAPH_DATA_YT_SetOffY()



描述

设置用来绘制对象数据的偏移量

原型

```
void GRAPH_DATA_YT_SetOffY(GRAPH_DATA_Handle hDataObj, int Off);
```

参数	含义
hDataObj	数据对象的句柄
Off	绘制数据的垂直偏移量

附加信息

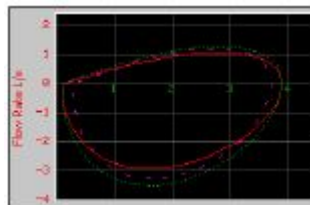
被显示的数据对象的垂直数据范围为 (0) 至 (数据区的 Y 坐标大小-1)，在使用滚动条的情况下当前的滚动位置被增加到范围之内。

举例

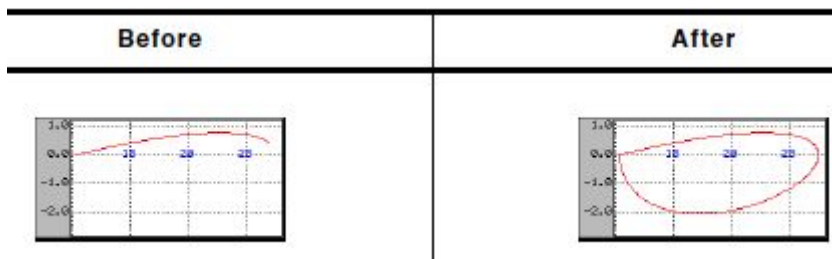
例如，可视数据范围应为-200 到-100 的数据需要向正方向转换 200 个点。

```
GRAPH_DATA_YT_SetOffY(hDataObj, 200);
```

7.3 GRAPH_DATA_XY 相关函数



GRAPH_DATA_XY_AddPoint()



描述

向一个 GRAPH_DATA_XY 对象增加一个新的数据

原型

```
void GRAPH_DATA_XY_AddPoint(GRAPH_DATA_Handle hDataObj, GUI_POINT * pPoint);
```

参数	含义
hDataObj	数据对象的句柄
pPoint	指向一个增加到数据对象的 GUI_PIONT 结构体的指针

附加信息

给出的点被增加到数据对象，如果数据对象是满的，也就是说包含了创建时在参数 MaxNumItems 中定义的数量一样多的数据，这时在增加之前首先对数据进行一次转换，所以当向一个已满的对象增加新点的时候，原有的第一个点就被转换出去了。

GRAPH_DATA_XY_Create()

描述

创建一个 GRAPH_DATA_XY 类型的数据对象

原型

```
GRAPH_DATA_Handle GRAPH_DATA_XY_Create(GUI_COLOR Color, unsigned  
MaxNumItems, GUI_POINT * pData, unsigned NumItems);
```

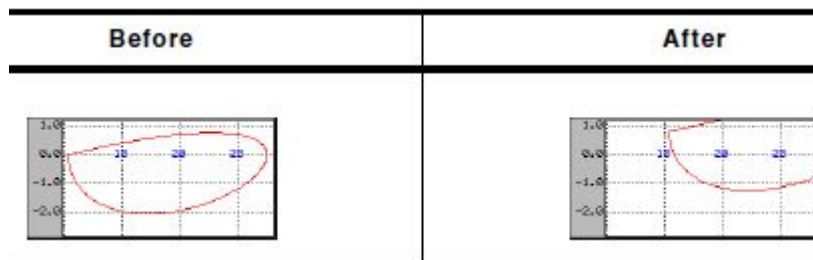
参数	含义
Color	用于绘制数据的颜色
MaxNumItems	最大的点数
pData	指向被增加到对象的数据指针，该指针应该指向 GUI_POINT 类型的数组
NumItems	被增加的点的数量

返回值

如果创建成功，返回数据对象的指针，否则返回 0

附加信息

一旦关联到一个 GRAPH 控件，数据对象不需要被应用删除，在删除控件的时候会被自动删除。

GRAPH_DATA_XY_SetOffX(), GRAPH_DATA_XY_SetOffY()**描述**

设置绘制曲线的垂直和水平偏移量

原型

```
void GRAPH_DATA_XY_SetOffX(GRAPH_DATA_Handle hDataObj, int Off);
```

```
void GRAPH_DATA_XY_SetOffY(GRAPH_DATA_Handle hDataObj, int Off);
```

参数	含义
hDataObj	数据对象的句柄
Off	显示曲线的水平/垂直偏移量

附加信息

数据对象的数据范围为 (0, 0) — (数据对象的 X 轴范围-1, 数据对象的 Y 轴范围-1)，当使用滚动条时，当前的滚动位置被增加到各自的范围内。若要使数据可视范围之外数据可见，应当设置偏移量，这样数据就位于可视范围之内了。

举例

例如，可视数据范围为 (100, -1200) — (200, -1100) 应按如下设置偏移量

```
GRAPH_DATA_XY_SetOffX(hDataObj, -100);
```

```
GRAPH_DATA_XY_SetOffY(hDataObj, 1200);
```

GRAPH_DATA_XY_SetLineStyle()**描述**

设置绘制曲线使用的线型

原型

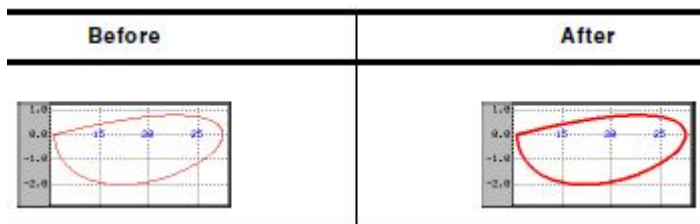
```
void GRAPH_DATA_XY_SetLineStyle(GRAPH_DATA_Handle hDataObj, U8 LineStyle);
```

参数	含义
hDataObj	数据对象的句柄
LineStyle	使用新的线型，所支持的线型请参考函数 GUI_SetLineStyle()

使用限制

请注意，如果线宽大于 1，只能使用 GUI_LS_SOLID(缺省值)。

GRAPH_DATA_XY_SetPenSize()



描述

设置绘制曲线使用的线宽

原型

```
void GRAPH_DATA_XY_SetPenSize(GRAPH_DATA_Handle hDataObj, int PenSize);
```

参数	含义
hDataObj	数据对象的句柄
PenSize	绘制曲线使用的线宽

7.4 比例尺相关函数

GRAPH 控件支持用于标注的水平和垂直比例尺，使用比例尺的函数见下面介绍

GRAPH_SCALE_Create()

描述

创建一个比例尺对象

原型

```
GRAPH_SCALE_Handle GRAPH_SCALE_Create(int Pos, int Align, unsigned Flags, unsigned TickDist);
```

参数	含义
Pos	相对于 GRAPH 控件左侧和顶侧边缘的位置
TextAlign	绘制数字的文字对齐方式，详细信息请参考函数 GUI_SetTextAlign()
Flags	见下表
TickDist	一个标注到另一个的距离

参数 Flags 的取值范围

GRAPH_SCALE_CF_HORIZONTAL	创建一个水平比例尺对象
GRAPH_SCALE_CF_VERTICAL	创建一个垂直比例尺对象

返回值

如果创建成功，返回比例尺对象的句柄，否则返回 0

附加信息

水平比例尺对象从数据区的底部到顶部显示，垂直比例尺对象从左边到右边显示，当第一个位置是 0 点时，参数 TickDist 标识了两个数字之间的距离。

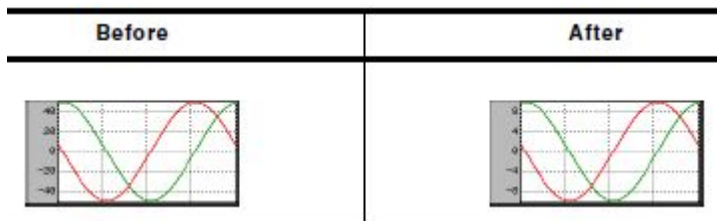
参数 Pos 表示，使用水平比例尺时，控件顶部到比例尺文字之间的距离（像素），使用
<以上所有信息均为中兴通讯股份有限公司所有，不得外传>

垂直比例尺时，这个参数表示从控件左侧边缘到水平文字位置之间的距离，请注意实际的文字位置还依赖于在 TextAlign 参数中定义的文字对齐方式。

比例尺对象在每一个位置上绘制一个数字，使用水平比例尺时有一个例外，如果第一个数字是 0，则不会在对应的位置上绘制。

一旦附加到一个 GRAPH 上，比例尺对象不需要被应用删除，当删除 GRAPH 控件的时候，它会被自动删除。

GRAPH_SCALE_SetFactor()



描述

设置计算所绘制的数字的比例因子

原型

```
float GRAPH_SCALE_SetFactor(GRAPH_SCALE_Handle hScaleObj, float Factor);
```

参数	含义
hScaleObj	比例尺对象的句柄
Factor	比例因子

返回值

返回之前使用的比例因子

附加信息

如果不使用比例因子，比例尺对象的单位是像素，因此设置的比例因子应当转换像素到所需的单位。

GRAPH_SCALE_SetFont()



描述

设置绘制比例尺所使用的字体

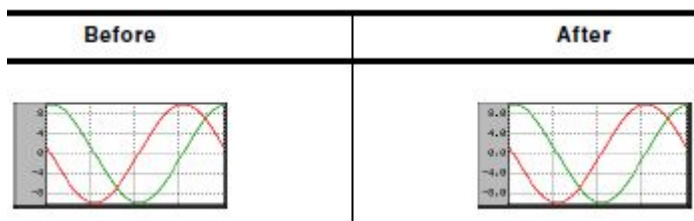
原型

```
const GUI_FONT * GRAPH_SCALE_SetFont(GRAPH_SCALE_Handle hScaleObj, const GUI_FONT * pFont);
```

参数	含义
hScaleObj	比例尺对象的句柄
pFont	使用的字体

返回值

返回之前使用的字体。

GRAPH_SCALE_SetNumDecs()**描述**

设置显示的小数点后的位数

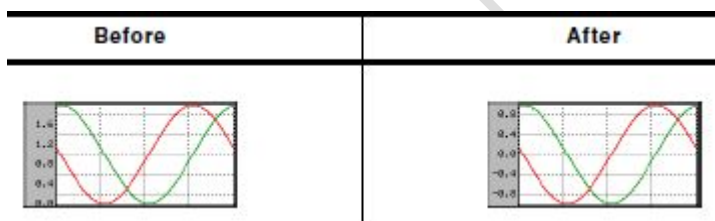
原型

```
int GRAPH_SCALE_SetNumDecs (GRAPH_SCALE_Handle hScaleObj, int NumDecs);
```

参数	含义
hScaleObj	比例尺对象的句柄
NumDecs	小数点后的位数

返回值

返回之前设置的小数点后位数。

GRAPH_SCALE_SetOff()**描述**

设置比例尺对象向正或负方向进行偏移的偏移量

原型

```
int GRAPH_SCALE_SetOff (GRAPH_SCALE_Handle hScaleObj, int Off);
```

参数	含义
hScaleObj	比例尺对象的句柄
Off	绘制比例尺的偏移量

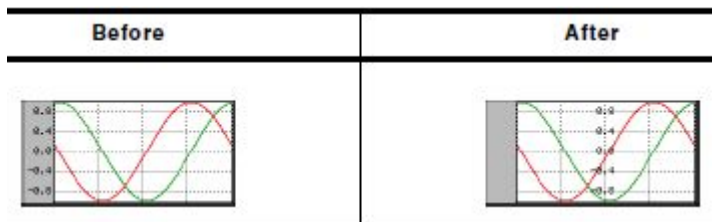
返回值

返回之前使用的偏移量。

附加信息

如同在函数 GRAPH_SCALE_Create() 函数中的描述，一个水平比例尺于是从数据区的底部边缘到顶部，一个垂直比例尺是从左侧边缘到右侧，当第一个位置是 0 点，很多情况下是不可取的，如果比例尺应当向正方向转换，则使用正的偏移量，同样，若想负方向转换，则使用负的偏移量值。

GRAPH_SCALE_SetPos



描述

设置在 GRAPH 控件里显示比例尺对象的位置

原型

```
int GRAPH_SCALE_SetPos(GRAPH_SCALE_Handle hScaleObj, int Pos);
```

参数	含义
hScaleObj	比例尺对象的句柄
Pos	比例尺显示的位置

返回值

返回之前的比例尺对象的位置

附加信息

参数 Pos 表示在使用水平比例尺的情况下，从控件顶部到比例尺文字之间的距离（像素值），使用垂直比例尺时，该参数表示从控件左侧边缘到水平文字位置之间的距离（像素），请注意，实际的文字位置还依赖于比例尺对象的位子对齐方式设置。

GRAPH_SCALE_SetTextColor



描述

设置显示数字的文字颜色

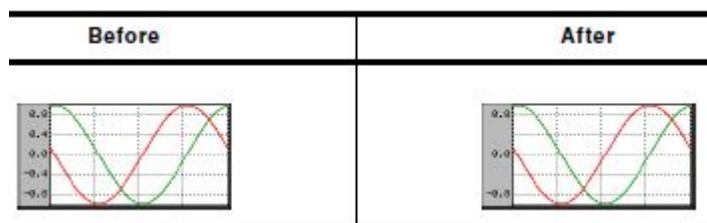
原型

```
GUI_COLOR GRAPH_SCALE_SetTextColor(GRAPH_SCALE_Handle hScaleObj, GUI_COLOR Color);
```

参数	含义
hScaleObj	比例尺对象的句柄
Color	显示数字的颜色值

返回值

返回之前使用的颜色。

GRAPH_SCALE_SetTickDist**描述**

设置一个数字到另一个数字之间的距离

原型

```
unsigned GRAPH_SCALE_SetTickDist(GRAPH_SCALE_Handle hScaleObj, unsigned Dist);
```

参数	含义
hScaleObj	比例尺对象的句柄
Dist	两个数字之间的距离值（像素）

返回值

返回之前的距离。