

修订历史

版本	日期	原因	编制	审查
V1.0	2016/12/26	创建文档	刘仁武	
V1.1	2018/05/03	添加 show、hide 介绍，修改部分字误	刘启鑫	



销售与服务

广州大彩光电科技有限公司

电话：020-82186683

传真：020-82187676

Email: hmi@gz-dc.com（公共服务）

网站: www.gz-dc.com

地址：广州高新技术产业开发区玉树工业园富康西街 8 号 C 栋 303 房

官网零售淘宝店: <https://gz-dc.taobao.com>

目录

1. 协议与变量设置.....	1
1.1 协议设置.....	1
1.2 变量定义.....	1
1.3 逻辑处理.....	2
1.3.1 控件绑定.....	3
1.3.2 控件显示.....	3
1.3.3 动画播放.....	3
1.3.4 按钮按下与弹起.....	4
1.3.5 条件执行.....	4
1.3.6 切换画面.....	4
1.3.7 设置变量.....	5
1.3.8 汇编指令.....	5
1.3.9 事件记录.....	5
1.3.10 曲线绑定.....	6
1.3.11 颜色设置.....	6
1.3.12 文本闪烁.....	6
1.3.13 蜂鸣器控制.....	7
1.3.14 外接矩阵键盘.....	7
1.3.15 写入时间.....	7
2. MINIC 脚本编程.....	8
2.1 MINIC 介绍.....	8
2.2 变量定义.....	8
2.3 系统入口函数.....	9
2.3.1 系统执行初始化 on_init.....	9
2.3.2 定期执行任务 on_systick.....	10
2.3.3 定时器超时通知 on_timer.....	10
2.3.4 变量更新通知 on_variant_upate.....	10
2.3.5 控件值更新通知 on_control_notify.....	10
2.3.6 画面切换通知 on_screen_change.....	11
2.4 系统变量.....	11
2.4.1 波特率 sys.baudrate.....	11
2.4.2 背光亮度 sys.backlight.....	11
2.4.3 蜂鸣器使能 sys.beep_en.....	11
2.4.4 系统通信故障 sys.com_err.....	11
2.4.5 当前画面 sys.current_screen.....	11
2.4.6 系统上电时间 sys.tick（单位秒）.....	11
2.4.7 系统日期 sys.year 年、sys.month 月、sys.day 日、sys.week 星期.....	11
2.4.8 系统时间 sys.hour 时、sys.minute 分、sys.second 秒.....	11
2.4.9 系统定时器 sys.timer0~sys.timer9（单位毫秒）.....	12
2.5 系统功能调用.....	12
2.5.1 启动定时器：start_timer(timer_id, timeout, countdown, repeat).....	12

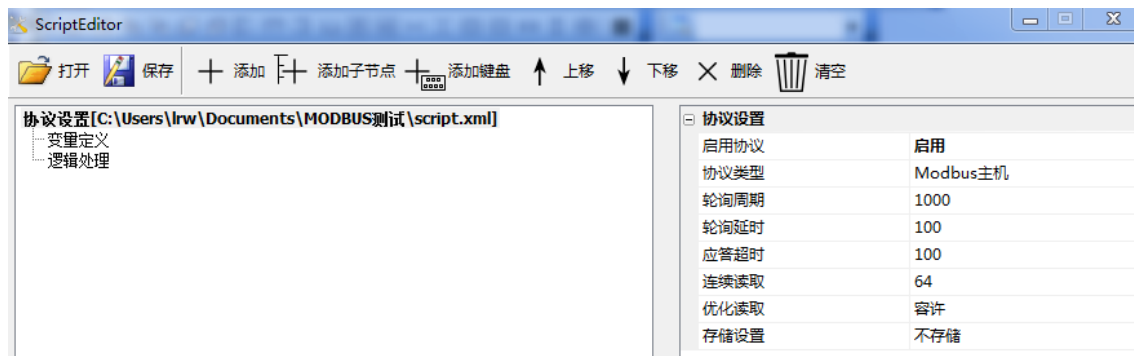


2.5.2	停止定时器: stop_timer(timer_id).....	12
2.5.3	设置系统日期: set_date(year, month, day).....	12
2.5.4	设置系统时间: set_time(hour, minute, second).....	12
2.5.5	设置变量: set(variant,value).....	12
2.5.6	显示控件: show (screen_id, control_id)	12
2.5.7	显示控件: hide (screen_id, control_id)	12

1. 协议与变量设置

此文档中描述的功能，仅适用于含 MODBUS 功能的固件版本。

使用 VisualTFT 组态软件，通过菜单“工具”-》“协议与变量设置”，打开如下图所示配置工具。

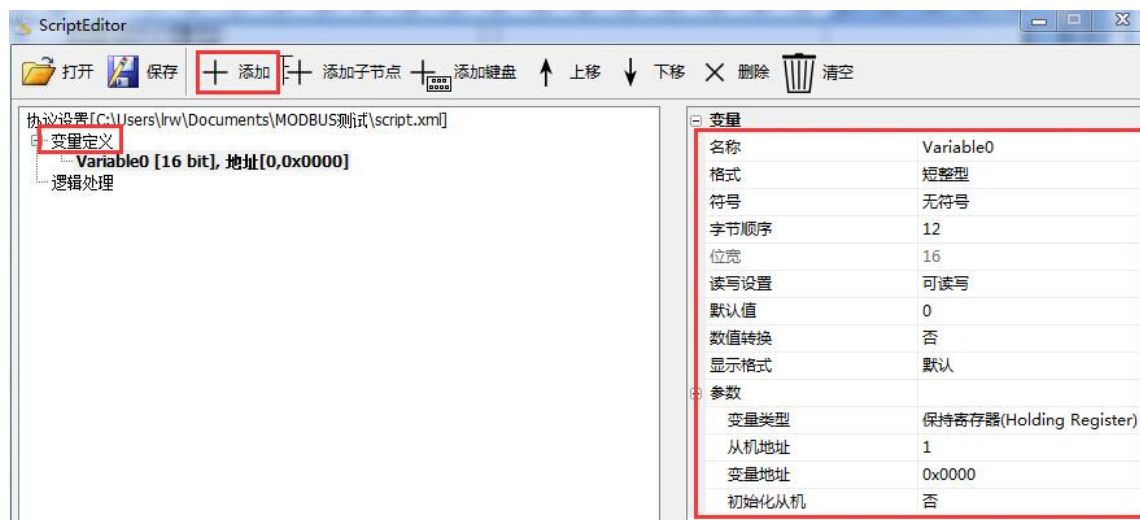


1.1 协议设置

- 启用协议：需要启用协议，才支持 MODBUS 主机/从机、PLC 通信；
- 协议类型：可选的通信协议包括
 - MODBUS 主机
 - MODBUS 从机
 - 三菱 PLC 通信协议
 - XGUS
- 从机地址：MODBUS 从机地址；
- 轮询周期：作主机时，变量的读取周期，默认 1 秒；
- 轮询延时：作主机时，相邻两次读取命令的间隔时间，默认为 100 毫秒；
- 应答超时：作主机时，允许从机的最大延时，默认 100 毫秒；
- 连续读取：作主机时，设置连续地址变量的最大读取数；
- 优化读取：作主机时，开启优化读取后，只读取与当前画面相关的变量；
- 存储设置：
 - 不存储：不存储，断电重启后变量恢复默认值；
 - 全部存储：不推荐，因为频繁存储会导致 FLASH 坏块增加；
 - 存储指定变量：推荐使用，一般用来存储很少更改的系统配置参数；
- 存储标记：识别存储在屏幕中的数据版本标记，当存储数据格式发生改动时，一般需要手动修改此存储标记值。

1.2 变量定义

选择左侧的“变量定义”节点，点击工具栏的“添加”按钮，可以添加变量，然后根据需要变量进行属性设置。



选择变量节点 Variable1，然后通过属性窗口进行设置：

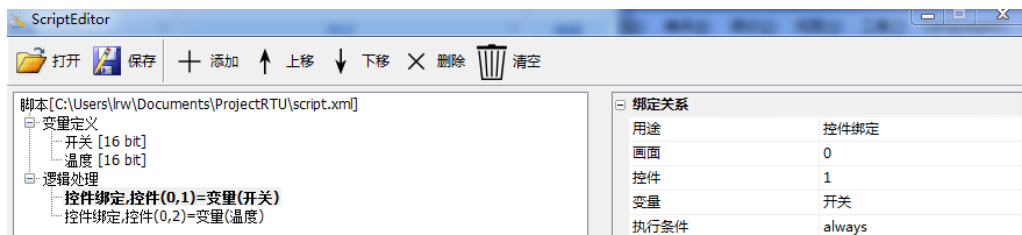
- 名称：可以修改为有意义的名称，如“温度”
- 格式：短整型(2 字节)，长整形(4 字节)，浮点数(4 字节)，字符串
- 符号：对于整数类型，可以设置为有符号/无符号
- 字节顺序：数据的高低字节排列顺序
- 位宽：变量的位宽与格式对应，字段可以设置位宽
- 读写设置：设置变量的读写权限
- 默认值：变量的默认值
- 数值转换：实际值=原始值*缩放+平移
- 显示格式：可以设置整数或小数显示补零

MODBUS 相关设置：

- 从机地址：主机模式时才需要设置
- 变量地址：变量（或寄存器）的地址
- 变量类型：线圈(Coils)，离散输入(Discrete Inputs)，保持寄存器(Holding Register)，输入寄存器 (Input Register)
- 内存变量：用于数据处理的变量（存储计算结果），此变量更改不发生串口通信
- 系统变量：屏幕中预定义的变量，包括波特率、背光亮度、蜂鸣器、日期时间等

1.3 逻辑处理

选择左侧“逻辑处理”节点，点击工具栏按钮“添加”逻辑处理。如下图所示，选择所示新建的逻辑关系，然后通过属性窗口进行设置。



通用属性参数介绍：

用途：设置当前逻辑的用途，例如控件绑定、控件显示等
画面：控件所在的画面 ID
控件：控件 ID
变量：控件关联变量名称
执行条件：可设置该逻辑的执行条件，Always 表示总是执行，Changed 表示变量改变时执行。

1.3.1 控件绑定

设置控件关联的变量：当控件改变时，自动更新变量值；反之变量更新时，更新控件显示。
使用 VisualTFT 软件 765 及以上版本，也可直接在控件属性窗口设置关联变量名称。

基本	
名称	Button1
编号	2
位置	
x坐标	139
y坐标	186
宽度	170
高度	89
关联变量	
变量名称	Variable0
控件设置	
触摸虚框	是
事件通知	是

1.3.2 控件显示

当设置的“执行条件”成立时显示控件，否则隐藏控件。

逻辑处理	
用途	控件显示
画面	1
控件	2
执行条件	
变量	Variable1
数值	1
标签	

1.3.3 动画播放

当设置的“执行条件”成立时开始播放动画，否则停止播放。

逻辑处理	
用途	动画播放
画面	1
控件	2
执行条件	==
变量	Variable1
数值	1
标签	

1.3.4 按钮按下与弹起

按钮按下时或者弹起时，如果满足“执行条件”，则“执行动作”。

逻辑处理	
用途	按钮按下
画面	1
控件	2
执行动作	设置变量
变量	Variable0
数值	0
执行条件	==
变量	Variable1
数值	1
标签	

1.3.5 条件执行

满足“执行条件”时“执行动作”，为防止重复执行，该逻辑仅在满足条件的瞬间执行一次（类似上升沿触发）。

画面条件：限定此逻辑仅在某个画面下有效。

逻辑处理	
用途	条件执行[触发]
执行动作	设置变量
变量	Variable0
数值	0
执行条件	==
变量	Variable1
数值	1
画面条件	任意
标签	

1.3.6 切换画面

满足“执行条件”时，切换到指定的目标画面，该逻辑为触发方式。

目标参数：目标画面可以为变量或者常量。

画面条件：限定此逻辑仅在某个画面下有效。

逻辑处理	
用途	切换画面[触发]
目标参数	变量
目标画面	Variable0
执行条件	
变量	Variable1
数值	1
画面条件	任意
标签	

1.3.7 设置变量

满足“执行条件”时，设置变量为指定数值。

写入优化：启用优化时，仅在改变目标变量时，执行写入。

画面条件：限定此逻辑仅在某个画面下有效。

逻辑处理	
用途	设置变量
变量	Variable0
数值	1
执行条件	
变量	Variable1
数值	1
模式	普通
写入优化	禁用
画面条件	任意
标签	

1.3.8 汇编指令

支持的汇编指令包括：

Jump 跳转：当“执行条件”成立时，跳转到“目标标签”，只能向后跳转。

Mov 赋值：MOV A B，把“参数变量或常量 B”设置到“目标变量 B”。

Add 加法：ADD A B，结果 A = A+B。

Sub 减法：SUB A B，结果 A = A-B。

And 逻辑与：AND A B，结果 A = A&B。

Or 逻辑或：OR A B，结果 A = A|B。

Mul 乘法：MUL A B，结果 A = A*B。

Div 除法：DIV A B，结果 A = A/B。

1.3.9 事件记录

当“执行条件”成立时，记录事件 ID。

逻辑处理	
用途	事件记录[触发]
画面	1
控件	1
事件参数	常量
事件ID	1
执行条件	
变量	温度
数值	50
标签	

1.3.10 曲线绑定

当“执行条件”成立时，在“历史曲线”中记录指定变量。

逻辑处理	
用途	曲线绑定
画面	1
控件	1
通道	0
变量	温度
采集周期	1
执行条件	always
标签	

1.3.11 颜色设置

当“执行条件”成立时，设置控件为指定颜色。

逻辑处理	
用途	颜色设置
画面	1
控件	1
颜色	<div></div> 255; 255; 255
执行条件	always
标签	

1.3.12 文本闪烁

当“执行条件”成立时，设置文本控件按指定周期闪烁。

逻辑处理	
用途	文本闪烁
画面	1
控件	1
闪烁周期	1
执行条件	always
标签	

1.3.13 蜂鸣器控制

当“执行条件”成立时，控制蜂鸣器发出声音。
模式：可设置为“单次”鸣叫或周期性鸣叫；
控制：对于周期性模式，可以设置为开始鸣叫、结束鸣叫、由条件控制；
时间：对于“单次”模式，可以设置鸣叫时间长度；
周期：对于“周期”模式，可以设置鸣叫的周期；

逻辑处理	
用途	蜂鸣器控制
模式	周期性
控制	由条件控制
周期	1
执行条件	
变量	Variable0
数值	1
标签	

1.3.14 外接矩阵键盘

该功能为定制功能，常规版本不支持。

1.3.15 写入时间

屏做主机时，可以定时写指定寄存器，设置从机时间。

逻辑处理	
用途	写入时间
从机	1
寄存器	0
周期	1
执行条件	always
标签	

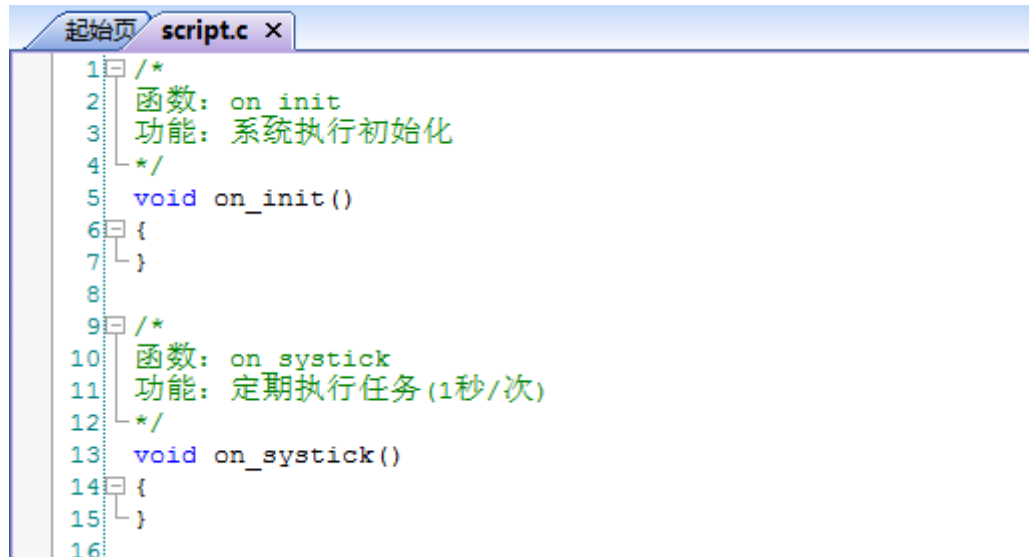
寄存器
时间寄存器起始地址，格式：年、月、日、星期、时、分、秒。

2. MINIC 脚本编程

2.1 MINIC 介绍

MINIC 是一种用于串口屏的脚本语言。它的语法类似 C 语言，但只保留最常用的功能。

使用 VisualTFT 组态软件，通过菜单“工具”-》“MINIC 脚本编程”，打开如下图脚本编辑窗口。



```
1  /*
2  函数: on_init
3  功能: 系统执行初始化
4  */
5  void on_init()
6  {
7  }
8
9  /*
10  函数: on_systick
11  功能: 定期执行任务(1秒/次)
12  */
13  void on_systick()
14  {
15  }
16
```

MINIC 包括的功能有：算数运算、位运算、逻辑运算、比较运算、赋值、IF 选择语句、系统功能调用等，具体语法与 C 语言一致。

算数运算：+加、-减、*乘、/除、%取模 ++自加 -自减

位运算：|位或、&位与、^按位异或、~按位取反

逻辑运算：||逻辑或、&&逻辑与、!逻辑非

比较运算：==、!=、<=、>=、<、>

赋值：= ；同时“=”支持数值和字符串相互赋值（MB2.22.1025.259 版本固件以上支持）

@操作符：符判定某个寄存器是否发送改变，例如 if(@“温度”)

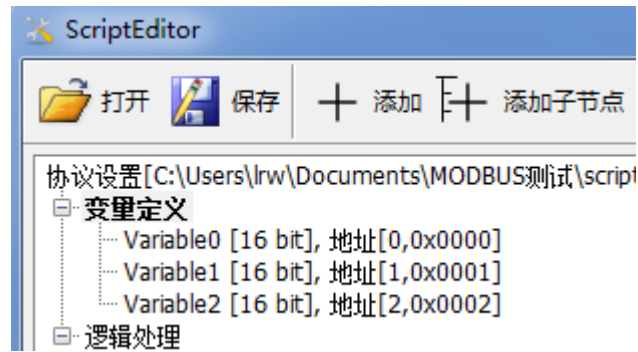
常量：包括整数(168)、浮点数(3.14)、十六进制数(0xAABB)

2.2 变量定义

变量包括：全局变量、局部变量(函数内部)。

全局变量定义方法如下图，全局变量可以直接在函数中访问。

如果变量为中文，需要用双引号包围，例如“温度”= 32。



支持的变量数据类型如下表：

变量类型	关键字	或者
有符号字节	char	int8
无符号字节	unsigned char	uint8
有符号短整型	short	int16
无符号短整型	unsigned short	uint16
有符号整数	int	int32
无符号整数	unsigned int	uint32
单精度浮点数	float	无

局部变量的定义方法如下图：

```

31 void on_timer(int timer_id)
32 {
33     int second = 0; //局部变量定义，只能在函数的最前前面
34
35     if(timer_id==0) //更新倒计时
36     {
37         second = (sys.timer1+900)/1000; //毫秒转换为秒
38
39         //全局变量为中文时，需要用双引号包围
40         "倒计时秒钟" = second%60;
41         "倒计时分钟" = (second/60)%60;
42         "倒计时小时" = (second/3600)%24;
43     }
44     else if(timer_id==1) //运行时间到
45     {
46         "运行" = 0; //停止运行
47     }
48 }

```

2.3 系统入口函数

系统入口函数，即系统中预定义的函数，不能随意更改。

目前定义的入口函数：on_init、on_systick、on_timer、on_variant_upate、on_control_notify。

2.3.1 系统执行初始化 on_init

/*

函数：on_init

功能：系统执行初始化，加载工程的时候执行一次

*/

```
void on_init()
{}
```

2.3.2 定期执行任务 on_systick

/*

函数：on_systick

功能：周期性执行任务，1 秒执行一次

*/

```
void on_systick()
{}
```

2.3.3 定时器超时通知 on_timer

/*

函数：on_timer

功能：定时器超时通知

参数：timer_id，定时器 ID

需要收到定时器通知，首先要启动定时器

启动定时器：start_timer(timer_id,timeout,countdown,repeat)

timer_id-定时器 ID (0~9)

timeout-超时时间，毫秒单位

countdown-0 顺计时，1 倒计时，决定 sys.timer 递增或递减

repeat-重复次数，0 表示无穷

停止定时器：stop_timer(timer_id)

定时器数值：sys.timer0~sys.timer9，毫秒单位

*/

```
void on_timer(int timer_id)
{}
```

2.3.4 变量更新通知 on_variant_upate

/*

函数：on_variant_upate

功能：通过串口读写寄存器导致变量更新时，执行此函数。

使用“@”操作符判定某个寄存器是否发送改变，例如 if(@“温度”)。

*/

```
void on_variant_upate()
{}
```

2.3.5 控件值更新通知 on_control_notify

/*

函数: on_control_notify

功能: 控件值更新通知, 例如用户点击按钮、通过键盘输入数据时执行

参数: screen_id, 产生此通知的画面 ID

参数: control_id, 产生此通知的控件 ID

参数: value, 数值, 例如按钮按下为 1, 按钮弹起为 0, 或者文本控件录入的值

*/

```
void on_control_notify(int screen_id,int control_id,int value)
```

```
{}
```

2.3.6 画面切换通知 on_screen_change

/*

函数: on_screen_change

功能: 画面切换通知, 当前画面 ID 发生变化时执行此函数

参数: screen_id, 当前画面 ID

*/

```
void on_screen_change(int screen_id)
```

```
{
```

```
}
```

2.4 系统变量

系统变量, 即系统中预先定义的变量, 方便在脚本中直接访问。

2.4.1 波特率 sys.baudrate

可读写, 表示当前串口的通信波特率, 此参数断电会保存。

2.4.2 背光亮度 sys.backlight

设置当前背光亮度值 0 最暗~255 最亮

2.4.3 蜂鸣器使能 sys.beep_en

写入 0 禁用蜂鸣器, 写入 1 启用蜂鸣器。

2.4.4 系统通信故障 sys.com_err

系统最后一次读写寄存器的操作是否发生故障, 该变量大于 0 表示有故障。

2.4.5 当前画面 sys.current_screen

可以通过该变量判断当前所在画面, 或者修改该变量实现画面切换。

2.4.6 系统上电时间 sys.tick (单位秒)

系统从上电以来的运行时间计数, 单位为秒, 重新断电会从 0 开始。

2.4.7 系统日期 sys.year 年、sys.month 月、sys.day 日、sys.week 星期

sys.year 年、sys.month 月、sys.day 日、sys.week 星期

这些变量为只读, 如果需要修改系统日期, 请使用 set_date 系统功能

2.4.8 系统时间 sys.hour 时、sys.minute 分、sys.second 秒

sys.hour 时、sys.minute 分、sys.second 秒

这些变量为只读，如果需要修改系统时间，请使用 set_time 系统功能

2.4.9 系统定时器 sys.timer0~sys.timer9（单位毫秒）

只读变量，该值可为顺计时或倒计时，参见 start_timer 系统功能

2.5 系统功能调用

系统功能即系统中预定义的函数，在脚本中可以直接调用。

2.5.1 启动定时器：start_timer(timer_id, timeout, countdown, repeat)

- timer_id-定时器 ID（0~9）
- timeout-超时时间，毫秒单位
- countdown-0 顺计时，1 倒计时，决定 sys.timer 递增或递减
- repeat-重复次数，0 表示无穷
- 定时器数值：通过 sys.timer0~sys.timer9 访问，毫秒单位

2.5.2 停止定时器：stop_timer(timer_id)

2.5.3 设置系统日期：set_date(year, month, day)

2.5.4 设置系统时间：set_time(hour, minute, second)

2.5.5 设置变量：set(variant,value)

该功能与赋值操作“=”类似，但有所差异，以 A=B 和 set(A,B)来说明。

相同之处：执行完毕之后 A 的值等于 B 的值；

不同之处：如果 A 是寄存器，A 的值没有改变时，使用赋值操作不会发送串口写入命令，而使用 set 功能则会发送串口写入命令。

同时该函数支持数值和字符串相互赋值（注：MB2.22.1025.259 版本固件以上支持）。

2.5.6 显示控件：show（screen_id, control_id）

该函数用于显示指定的控件（注：MB2.22.1025.259 版本固件以上支持）。

- screen_id 指定的画面 ID
- control_id 指定的控件 ID

2.5.7 显示控件：hide（screen_id, control_id）

该函数用于隐藏指定的控件（注：MB2.22.1025.259 版本固件以上支持）。

- screen_id 指定的画面 ID
- control_id 指定的控件 ID