

VisualHMI - 安全登录

什么是安全登录？

VisualHMI 提供安全登录功能，用于对界面控件的操作权限进行分级管控。设置安全等级：菜单栏 → 工程 → 工程设置 → 安全等级，在此处定义不同用户及对应密码。可为控件指定所需的安全等级，从而实现对控件操作权限的限制。安全登录的具体配置界面如下图所示

安全登录	
安全等级	3级
1级用户密码	111111
2级用户密码	222222
3级用户密码	333333
保持登录状态	是
超时退出登录	<input type="checkbox"/>
登录键盘位置	默认
设置用户名称	<input checked="" type="checkbox"/>
0级用户名称	User0
1级用户名称	User1
2级用户名称	User2
3级用户名称	User3
工程密码	<input type="button" value="设置..."/>

使用范围：VisualHMI - HMI&M系列&Dx系列

应用下载：[VisualHMI - 安全登录\(点击下载\)](#)

本《VisualHMI - 安全登录》内含有两个DEMO

- 《Project Pwd Leading 0》DEMO支持前导零，和在线文档描述一致。即是：假设密码长度固定为6位，则修改密码为66，实际为000066
- 《Project Pwd》DEMO无前导零，在线文档不体现，主要配合0x011B -系统参数寄存器，0x011A系统参数保存 / 加载 / 恢复。即是：修改密码为66，实际即为66

1.属性说明

- 安全等级：设置工程中用户的等级，支持1 - 8级，0为游客身份。用户密码：根据选择的安全等级，支持分别设置各级别用户的初始密码，支持LUA脚本修改密码；
- 保持登录状态：☒。登录后保持登录状态；
 - 超时退出登录：☐。HMI空闲一段时间后（无触摸），是否自动退出登录状态；【保持登录状态：启动生效】
 - 超时时间：超时退出登录的计时时间，单位为秒；【保持登录状态：启动生效】
 - 返回画面：超时退出后返回的界面，-1保存当前页面；【保持登录状态：启动生效】
- 登录键盘位置：支持默认和居中两种弹出方式；
 - 默认：显示在触摸点击位置；
 - 居中：显示在屏幕中心位置。
- 设置用户名称：不同级别用户设置名称；

安全登录	
安全等级	8级
1级用户密码	111111
2级用户密码	222222
3级用户密码	333333
4级用户密码	444444
5级用户密码	555555
6级用户密码	666666
7级用户密码	777777
8级用户密码	888888
保持登录状态	否
超时退出登录	<input checked="" type="checkbox"/>
超时时间	60
返回画面	0_主界面
登录键盘位置	默认
设置用户名称	<input checked="" type="checkbox"/>
0级用户名称	User0
1级用户名称	User1
2级用户名称	User2
3级用户名称	User3
4级用户名称	User4
5级用户名称	User5
6级用户名称	User6
7级用户名称	User7
8级用户名称	
工程密码	设置...

2.应用

2.1.开启安全登录

点击菜单栏→工程→工程设置，安全等级启用，如下所示：

- 安全等级：5级。用户密码依次为111111,222222,333333,444444,555555
- 保持登录状态：√。
 - 超时退出登录：√
 - 超时时间：5
 - 返回界面：0_主界面



2.2.超时退出登录

2.2.1.画面配置

在主画面中添加功能按钮，用于切换界面时，需要密码登录，如下所示：

- 按钮用途：切换画面
 - 目标画面：1_密码设置
- 用户等级：√
 - 最低要求等级：2

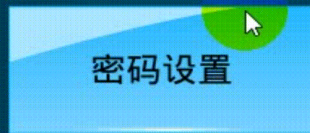


2.2.2.运行预览

运行虚拟屏，点击密码设置，弹出键盘输入密码，输入 ≥ 2 级的密码，即可执行画面跳转，当设置界面无操作10s，返回主界面，再次登录仍需要输入密码，如下所示：

VisualHMI - 密码等级

(密码设置需等级2级及以上用户)



VisualHMI

2.3.密码修改

2.3.1.画面配置

分期密码不支持读取，修改需要通过LUA脚本，`set_stage_pwd(level,pwd,len)`

`set_pwd(level,pwd)`：修改安全等级密码。

- level: 用户等级0~7，对应密码等级1~密码等级10
- pwd: 用户数字密码
- len: 选填。密码长度，不足用前导零填充。len=5，pwd=123，即密码为00123

[!warning|tip:注意]

若需要使用前导0密码，则所有等级密码也需要设置为前导0

hmi的密码不支持读取，需要初始化或修改的时候同步到RW寄存器

用户修改密码或初始化时候，可以将密码写到RW寄存器，修改时可查看当前密码

2.3.2.工程配置

画面上添加5个数值控件，用于修改安全等级的密码接口，配置如下所示：

1. 启用输入：√
2. 读取地址：RW0000/RW0002/RW0004/RW0006/RW0008
3. 数据类型：UINT32



2.3.3.LUA脚本

屏幕首次使用，可以安全等级的密码写入Flash中。新的Flash里的内容都是0xFF，LUA脚本可以读取一个地址(0x000A~0x7FFF，历程中已用了0x0000~0x0008)，用来初始化RW寄存器的标识。

初始化安全等级密码：

```
function on_init()

    local _val = get_uint16(VT_RW, 0x2000)

    if _val ~= 0x0055
    then
        -- 初始化函数中，给对应内部寄存器赋密码初值
        set_uint32(VT_RW, 0x0000, 111111)
        set_uint32(VT_RW, 0x0002, 222222)
        set_uint32(VT_RW, 0x0004, 333333)
        set_uint32(VT_RW, 0x0006, 444444)
        set_uint32(VT_RW, 0x0008, 555555)

        set_pwd(0, 111111, 6)
        set_pwd(1, 222222, 6)
        set_pwd(2, 333333, 6)
        set_pwd(3, 444444, 6)
        set_pwd(4, 555555, 6)

        set_uint16(VT_RW, 0x2000, 0x0055)
        flash_sync()
    else
        set_pwd(0, get_uint32(VT_RW, 0x0000), 6)
        set_pwd(1, get_uint32(VT_RW, 0x0002), 6)
        set_pwd(2, get_uint32(VT_RW, 0x0004), 6)
        set_pwd(3, get_uint32(VT_RW, 0x0006), 6)
        set_pwd(4, get_uint32(VT_RW, 0x0008), 6)
    end
end
```

修改安全登录密码，在on_updata(...)回调函数里面，判断RW寄存器触发，将寄存器的值设置对应的等级密码，并存储。

```
function on_update(slave,vtype,addr)

.....
if vtype == VT_RW
then
    if addr >= 0x0000 and addr <= 0x0008
    then
        local psw = get_uint32(vtype, addr)
        local idx = (addr // 0x0002)
        set_pwd(idx, psw, 6)    -- 设置对应等级密码
        flash_sync()
    end
end
end
```

2.3.4.运行预览

运行虚拟屏，修改密码等级2，为456，密码长度为6，则密码为“000456”，效果如下所示：

