aardio 范例: 电源管理关屏

```
//电源管理关屏
import fonts.fontAwesome;
import win.ui;
/*DSG{{*/
var winform = win.form(text="自动关屏";left=-20;top=50;right=319;bottom=99;bgcolor=16777215;border="none";max=false)
winform.add(
btnClose={cls="close";text="×";left=297;top=11;right=325;bottom=39;bgcolor=13405076;dr=1;dt=1;font=LOGFONT(h=-21);z=3};
btnCloseLcd={cls="plus";text="关屏";left=191;top=14;right=240;bottom=33;color=8388608;notify=1;textPadding={left=5};z=5};
chkLock={cls="plus";text="锁屏";left=242;top=14;right=296;bottom=33;align="left";iconStyle={align="left";font=LOGFONT(name='FontAwesome')};iconText='\uF0C8 ';notify=1;textPadding=
datetimepick={cls="datetimepick";left=111;top=14;right=186;bottom=38;edge=1;transparent=1;updown=1;z=1};
lbTitle={cls="static";text="空闲时间超过: ";left=0;top=18;right=103;bottom=40;align="right";transparent=1;z=2}
)
/*}}*/

import win.ui.tooltip
var balloonTipCtrl = win.ui.tooltip.tracking(winform,false)

import sys.power;
winform.datetimepick.onDateTimeChanged = function(dateTime){

    if(tonumber(dateTime)<5){
        var x,y,cx,cy = winform.datetimepick.getPos(true);
        balloonTipCtrl.setText("小于 5 秒，已禁用自动关屏。").trackPopup(true,x+20,y+cy );

        sys.power.setMonitorTimeout(0);
    }
    else {
        balloonTipCtrl.trackPopup(false);
        sys.power.setMonitorTimeout(tonumber(dateTime));
    }
}

var timeout = sys.power.getMonitorTimeout() : 0;
winform.datetimepick.time = time.iso8601(timeout);
if(timeout<5) winform.datetimepick.onDateTimeChanged(timeout);
winform.datetimepick.setFormat("' 'HH':'mm':'ss");

import fsys.config;
var config = fsys.config(io.appData("aardio/std/closeLcd"))

//注册电源设置事件通知窗体
import sys.power.notification;
var sn = sys.power.notification(winform,"6FE69556-704A-47A0-8F24-C28D936FDA47");

import sys;
sn.onPowerSettingChange = function(guid,data){
    if(data == 0 && config.setting.lock){
        sys.lock(); //关屏时自动锁屏
    }
}

winform.onMouseDown = function(){
    balloonTipCtrl.trackPopup(false);
    winform.hitCaption();
}

winform.chkLock.oncommand = function(id,event){
    config.setting.lock = winform.chkLock.checked;
    config.setting.save();
}

//主动关屏
winform.btnCloseLcd.oncommand = function(id,event){
    //延时异步执行，避免显示器刚关掉又被唤醒
    winform.setTimeout(
        function(){
            //用:User32.SendMessage会卡，改用异步的::User32.SendNotifyMessage
            ::User32.SendNotifyMessage( 0xFFFF/*_HWND_BROADCAST*/ ,0x112/*_WM_SYSCOMMAND*/, 0xF170/*_SC_MONITORPOWER*/ ,2);
        },200 //给一点延时，万一鼠标还在晃呢?
    )
}

winform.chkLock.checked = config.setting.lock;

winform.btnCloseLcd.skin({
    color={
        active=0xFF00FF00;
        default=0xFF000080;
        disabled=0xFF6D6D6D;
        hover=0xFFFF0000
    }
})

winform.chkLock.skin(
    color = {
        hover = 0xFFFF0000;
        active = 0xFF00FF00;
    }
    checked = {
        iconText = '\uF14a';
    }
)

//点击 close 控件关闭窗口。
winform.btnClose.oncommand = function(id,event){
    winform.close();
}
winform.btnClose.show(true);

import win.ui.shadow;
win.ui.shadow(winform);

winform.show();
win.loopMessage();
```

Markdown 格式