

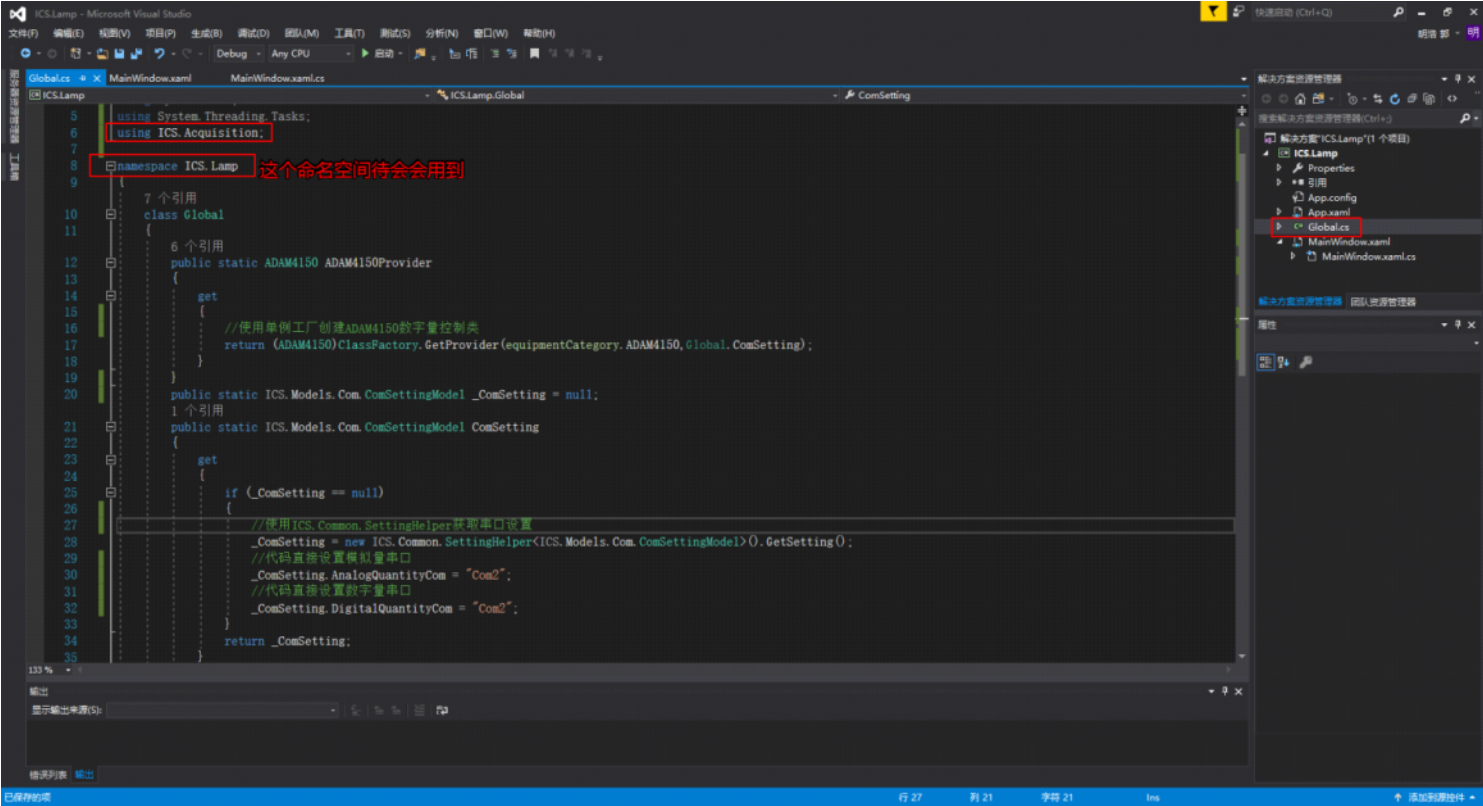
C Sharp LED灯控制模块

2017年3月20日 22:19

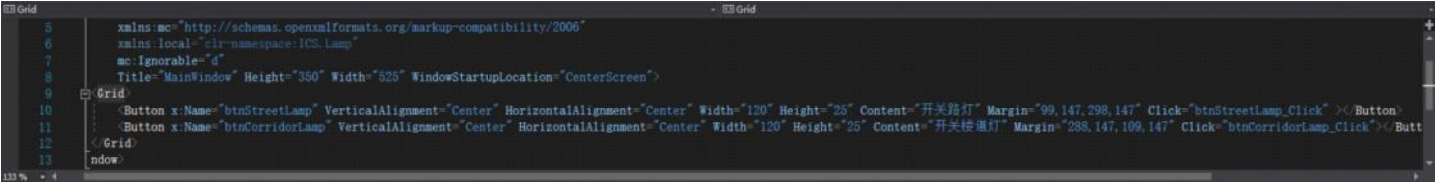
1. 引用3个dll文件。
- ICS.Acquisition.dll

ICS.Common.dll

ICS.Models.dll
2. 在项目中添加一个类用于配置Com口信息等，见下图。



3. 自行绘制两个按钮或者其他控件用来控制路灯和楼道灯的开关，新大陆给的例程中界面过于繁重，并未采取该方式，只是简单的用了两个按钮。两个按钮的点击分别对应接下来写的两个LED的开关函数。



4. LED开关函数，详细代码如下。
- a. 使用两个全局变量去记录两个LED灯的开关情况，程序运行时两标志位默认为关闭。

b. 当按钮点击时，将标志位取反，并在开关函数中分别调用重载后的开关函数以改变LED灯的状态。

```
//LED灯开关标志位
private bool streetLampStatus = false; //路灯
private bool corridorLampStatus = false; //楼道灯

1 个引用
private void OnOffStreetLamp()
{
    //修改LED灯开关标志位
    streetLampStatus = !streetLampStatus;
    //调用重载方法去开关LED
    OnOffStreetLamp(streetLampStatus);
}

1 个引用
private void OnOffCorridorLamp()
{
    corridorLampStatus = !corridorLampStatus;
    OnOffCorridorLamp(corridorLampStatus, true);
}
```

- c. 使用Global.ADAM4150Provider.CheckSerialPort来判断相应串口是否打开，否则返回错误信息。需要注意的是上面提到会用到的命名空间，如果Global类的命名空间与现在的命名空间不符的话就需要引用。

```
/// <summary>
/// 开关路灯
/// </summary>
/// <param name="onOff">true为开false为关</param>
1 个引用
private void OnOffStreetLamp(bool onOff)
{
    //使用Global.ADAM4150Provider.CheckSerialPort来判断相应串口是否打开
    ICS.Common.ResultEntity result = Global.ADAM4150Provider.CheckSerialPort(Global.ADAM4017Provider);
    //若执行状态 (ret.Status) 为失败 (Result.Failure)，则表示串口未打开。设置错误信息用于返回
    if (result.Status == RunStatus.Failure)
    {
        MessageBox.Show(result.ResultMessage);
        return;
    }
}
```

- d. 将ADAM4150数字量控制类赋值给新变量，方便下面的使用。
- e. 依据调用开关函数时传进来的参数去判断是开关还是关闭LED灯，使用上面建立的ADAM4150数字量控制类的OnOff方法去发送开启或者关闭LED灯的命令。

f. 判断操作结果是否成功，不成功则告知用户。

```
//将ADAM4150数字量控制类赋值给新变量，方便下面的使用
ADAM4150 adam = Global.ADAM4150Provider;

//判断是打开还是关闭路灯，adam.OnOff(data);执行命令返回执行结果
if (OnOff == false)
{
    //byte[]数组存放关闭路灯命令
    byte[] data = new byte[] { 0x01, 0x05, 0x00, 0x12, 0x00, 0x00, 0x6D, 0xCF };
    OnOff = adam.OnOff(data);
    btnStreetLamp.Content = "开关路灯";
}
else
{
    //byte[]数组存放开启路灯命令
    byte[] data = new byte[] { 0x01, 0x05, 0x00, 0x12, 0xFF, 0x00, 0x2C, 0x3F };
    OnOff = adam.OnOff(data);
    btnStreetLamp.Content = "关闭路灯";
}

//判断执行结果是否成功
if (OnOff == false)
    MessageBox.Show("开启路灯失败。");
}
```

5. 附上上面没有出现的代码。

```
<summary>
// 开关楼道灯
</summary>
<param name="onOff">true为开false为关</param>
<param name="infrared">用于控制是否需要红外线感应才开启</param>
1 个引用
private void OnOffCorridorLamp(bool OnOff, bool infrared)
{
    //使用Global.ADAM4150Provider.CheckSerialPort来判断相应串口是否打开
    ICS.Common.ResultEntity result = Global.ADAM4150Provider.CheckSerialPort(Global.ADAM4150Provider.ADAM4017Provider);
    //若执行状态 (retData.Status) 为失败 (Result.Failure)，则表示串口未打开，设置错误信息用于返回
    if (result.Status == RunStatus.Failure)
    {
        MessageBox.Show(result.ResultMessage);
        return;
    }

    //将ADAM4150数字量控制类赋值给新变量，方便下面的使用
    ADAM4150 adam = Global.ADAM4150Provider;

    //判断是打开还是关闭楼道灯，byte[]数组存放开关灯命令，adam.OnOff(data);执行命令返回执行结果
    if (OnOff == false)
    {
        //byte[]数组存放关闭楼道灯命令
        byte[] data = new byte[] { 0x01, 0x05, 0x00, 0x11, 0x00, 0x00, 0x9D, 0xCF };
        OnOff = adam.OnOff(data);
        btnCorridorLamp.Content = "开关楼道灯";
    }
    else
    {
        //byte[]数组存放开启楼道灯命令
        byte[] data = new byte[] { 0x01, 0x05, 0x00, 0x11, 0xFF, 0x00, 0xDC, 0x3F };
        OnOff = adam.OnOff(data);
        btnCorridorLamp.Content = "关闭楼道灯";
    }

    if (OnOff == false)
        MessageBox.Show("操作楼道灯失败！");
}
```

```
1 个引用
private void btnStreetLamp_Click(object sender, RoutedEventArgs e)
{
    OnOffStreetLamp();
}

1 个引用
private void btnCorridorLamp_Click(object sender, RoutedEventArgs e)
{
    OnOffCorridorLamp();
}
```