



31 Days of Windows 8

Windows 8 开发 31 日

第 24 日

光传感器

译者：BeyondVincent(破船)

时间：2013.4.25

版本： 2.0

关于破船

程序猿砌墙于云南昆明!

长期扎根移动软件开发!

爱跑步爱打篮球爱运动!

命中无大富大贵之面相!

愿健康与平淡相随一生!

你可以发邮件与破船取得联系: BeyondVincent@gmail.com

还可以关注破船的微博: [腾讯微博](#)和[新浪微博](#)。

这里是破船的个人博客, 欢迎光临: [破船之家](#)



关于 Windows 8 开发 31 日翻译



Windows 8 开发 31 日是由 Jeff Blankenburg 和 Clark Sell 原创的。

官方站点：<http://31daysofwindows8.com/>

涉及到两个版本：

XAML/C# (由 Jeff Blankenburg 撰写)

HTML5/JS (由 Clark Sell 撰写)

其中涉及到的资源和相关代码请到这里下载：

<https://github.com/csell5/31DaysOfWindows8>

在这里，由于破船对 HTML5/JS 不熟悉，所以只翻译 XAML/C# 相关主题。

建议大家前往看原创内容，如果看不明白，再来这里看我翻译的相关内容。

如果翻译不正确的地方，可以通过上面的联系方式告诉破船。

破船祝你阅读愉快！



目录

关于破船 2

关于 Windows 8 开发 31 日翻译 3

目录 4

第 24 日光传感器 5

 1.0. 介绍5

 1.1. 光传感器的使用5

 1.2. 总结8

第 24 日光传感器



1.0. 介绍

今天，今天，我将介绍 Windows8 设备中可能存在的另外一个传感器：光传感器。通过光传感器，我们可以获知用户设备周围的光亮度，进而调整程序的对比度，亮度以及其它相关值，以利于我们的程序可以在高亮度和低亮度情况下，也能够有很好的用户体验。

1.1. 光传感器的使用

光传感器的使用与昨天介绍的罗盘类似。

- 初始化光传感器
- 如果光传感器可用，创建一个 ReadingChanged eventhandler
- 在 eventhandler 中，从传感器中获取数据并显示到屏幕上

下面是完整的 MainPage.xaml.cs 文件：

```
using System;
using Windows.Devices.Sensors;
using Windows.UI.Core;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Navigation;
```



```

namespace Day24_LightSensor
{
    publicsealedpartialclass MainPage : Page
    {
        public MainPage()
        {
            this.InitializeComponent();
        }

        LightSensor sensor;

        protectedoverridevoid OnNavigatedTo(NavigationEventArgs e)
        {
            sensor = LightSensor.GetDefault();
            if (sensor != null)
            {
                sensor.ReadingChanged += sensor_ReadingChanged;
                Data.Visibility = Visibility.Visible;
            }
            else
            {
                NoSensorMessage.Visibility = Visibility.Visible;
            }
        }

        asyncvoid sensor_ReadingChanged(LightSensor sender, LightSensorReadingChangedEventArgs args)
        {
            await Dispatcher.RunAsync(CoreDispatcherPriority.Normal, () =>
            {
                Lux.Text = args.Reading.IlluminanceInLux.ToString();
                TimeStamp.Text = args.Reading.Timestamp.ToString();
            });
        }
    }
}

```

在获取数据的方法上没有惊奇的地方，不过在不同的机器上，相同的环境中，获得的数据是极有可能不相同的。例如，在房间里面，我的 Qualcomm ARM 设备获得的是 59lux，但是我的 Samsung tablet 则是 42lux，而我的 Surface RT 设备在同样的环境中是 115lux。

出现上面这样情况主要取决于每个设备的光传感器的精确度和质量，不过总



的来说，它们获得的值相差不会太远。下面是在不同环境下的光照度值（来自 [Wikipedia article on Lux](#)）

Examples	
Illuminance	Surfaces illuminated by:
10^{-4} lux	Moonless, overcast night sky (starlight) ^[2]
0.002 lux	Moonless clear night sky with airglow ^[2]
0.27–1.0 lux	Full moon on a clear night ^{[2][3][4]}
3.4 lux	Dark limit of civil twilight under a clear sky ^[5]
50 lux	Family living room lights (Australia, 1998) ^[6]
80 lux	Office building hallway/ toilet lighting ^{[7][8]}
100 lux	Very dark overcast day ^[2]
320–500 lux	Office lighting ^{[9][10][11]}
400 lux	Sunrise or sunset on a clear day.
1,000 lux	Overcast day; ^[2] typical TV studio lighting
10,000–25,000 lux	Full daylight (not direct sun) ^[2]
32,000–130,000 lux	Direct sunlight

如上所示，即使是 100lux 也是一个很暗的值。如果在我的办公室里面，值接近 175。通过上表，你应该创建一个适当的光亮度范围。

例如，如果你识别出当前用户是在光照不足的环境中，你可能需要显示暗背景和白色文字，这样才利于在光线不足的环境中阅读。在明亮的房间里面，应该切换为白色背景和黑色文字。

现在，你已经知道了如何从光传感器中识别出两种不同的环境，并将其使用到程序中。



1.2. 总结

今天，我介绍了光传感器，以及如何使用光传感器来修改用户界面，使程序更加易读。

点击下图，下载本文示例代码：



明天，我将介绍更强健的一个传感器：加速度计。我们可以使用其数据判断出用户设备的旋转。到时候见！



感谢你的阅读！

如果对这篇文章有什么想法，可以与破船联系，破船的联系方式在文章开头。

破船

