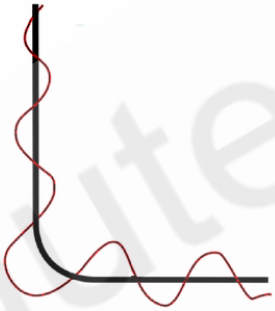
fischertechnik ®

# 《摄像头的线条探测》

## 功能介绍

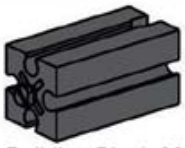
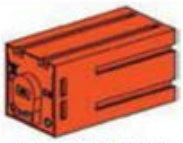


我们需要通过设置选项来告诉摄像头要寻找什么。

最简单的循迹是追踪那些在白色背景上的黑线。 由于我们的摄像头可以区分不同的颜色，因此循迹的能力是无止境的。 摄像头还可以一次看到几条线，如果我们清楚这些线条的不同条件，那么我们可以告诉程序何时停止跟踪一条线条并开始追踪其他的线条。

以下的学习将帮助你理解摄像头是如果看到线条和颜色，并如何实现追踪。

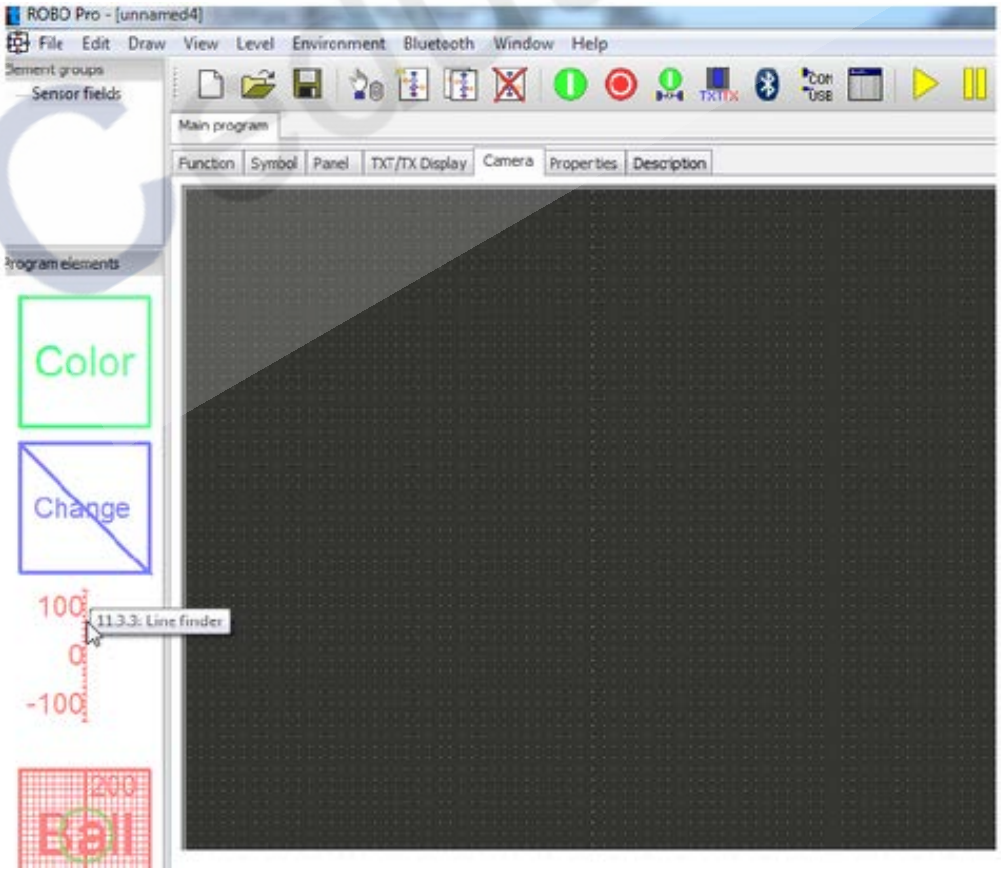
**实验组件：**

| | | |
|---|---|---|
| TXT Controller X1 |  Building Block 30 X14 | USB Camera X1 |
| Cable clamp X1 | Map with lines 画有线条的图纸 |  Encoder motor X2 |

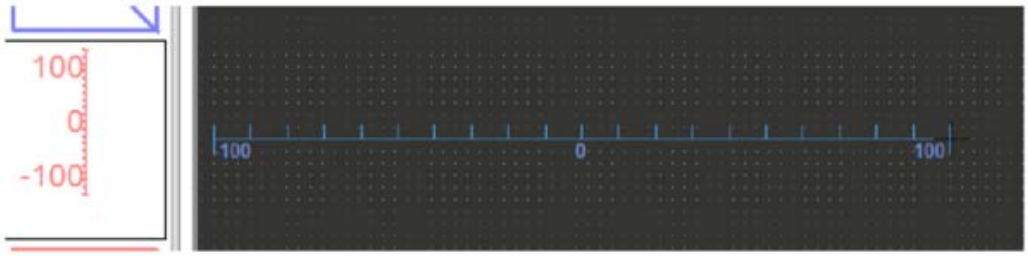
我们用上述组件搭建一个如下的结构：

下面开始摄像头的配置：

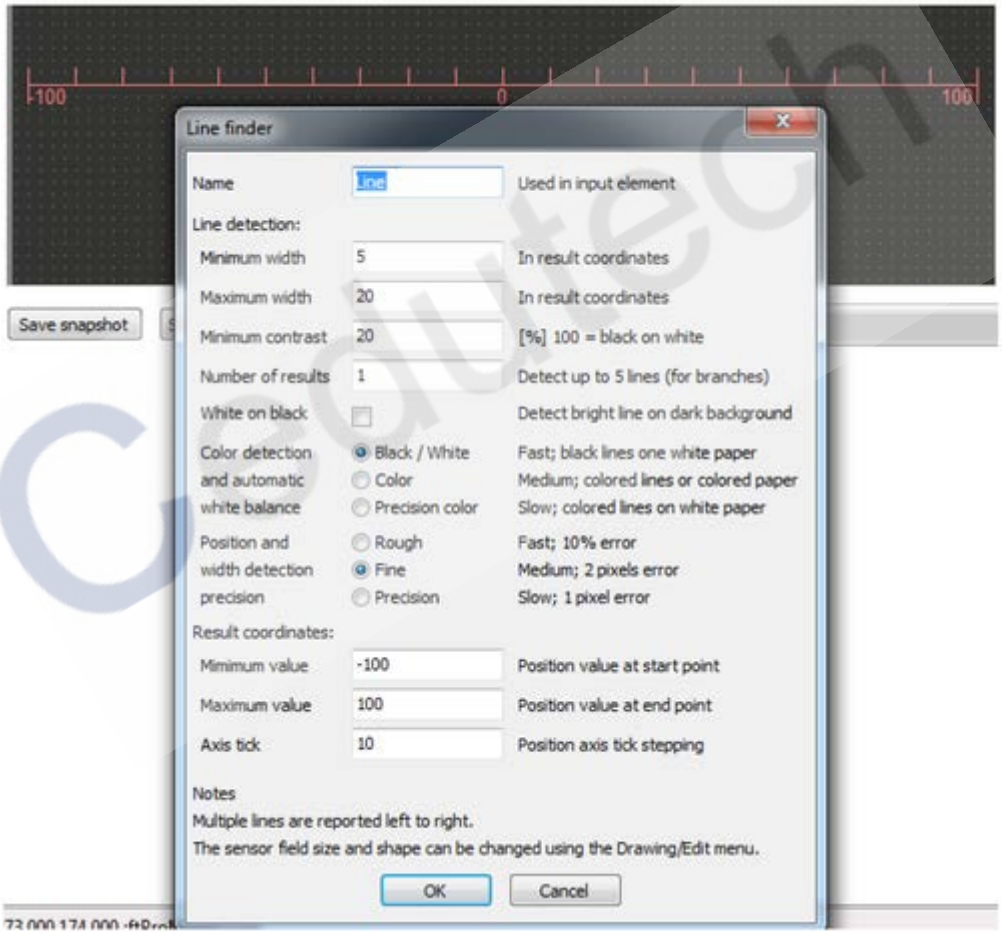
在软件中新建一个程序，打开摄像头窗口。 点击左上角的“传感器域”，在下面的程序模块中选择线条探测模块：



单击“线条探测”模块后，鼠标形状变为笔形，现在在视频监控窗口中绘制监测标尺，可以拖动监测标尺两端的锚点以改变形状，确定形状后，点击监测标尺以外的地方以确定设置，鼠标变回光标形状：




之后右键单击监测标尺以弹出属性设置面板：

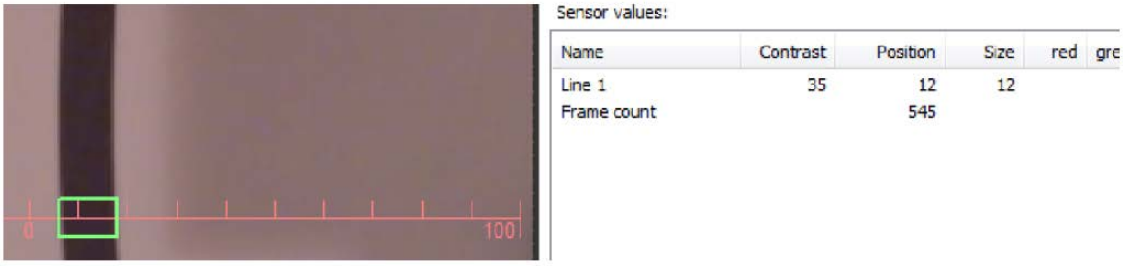



在这个属性框中有很多设置，在本实例中，我们将经常返回这里进行属性值的更改，看看会如何影响摄像头的线条循迹。

由于我们会在程序中对某些属性进行调用，因此我们对属性的设置还将影响编程界面的内容。

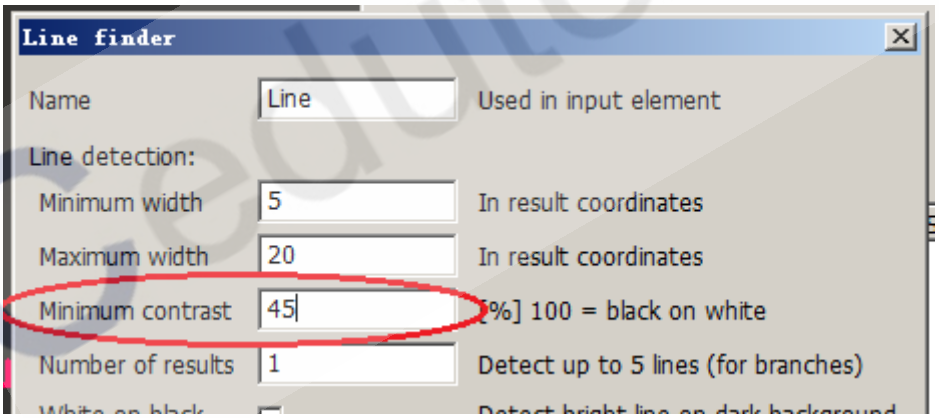
现在，就从这些默认设置开始。 选择“OK”。

将画有线条的纸张放置在摄像头的正下方，然后勾选“实时预览”复选框，从监测窗口中看看摄像头正在监测什么：

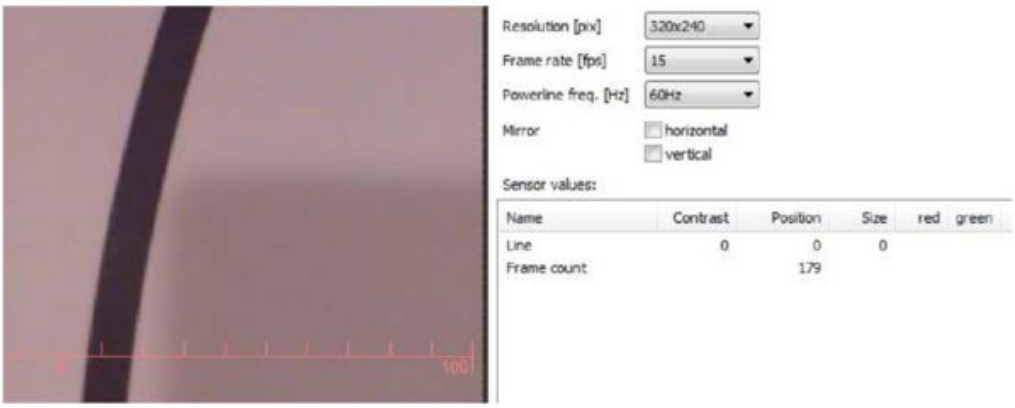



上图中，在摄像头下方有一个黑色的线条，监测到的数据信息为：对比度为 35，位置为 12（这里的位置是指该黑色线条宽度的中心点距离监测标尺中心点“0”之间的距离，这个值是带有正负的，正值表示在监测线条的右侧，如果是负值，表示在监测线条的左侧），尺寸 12（尺寸指在标尺线上的黑色线条宽度）。

现在取消“实时预览”。 右键单击监测标尺以再次弹出属性框。 在上图中，对比度为 35。现在我们将“Minimum contrast（最小对比度）”的值设置的高于这个监测到的值（比如 40 或 45）：

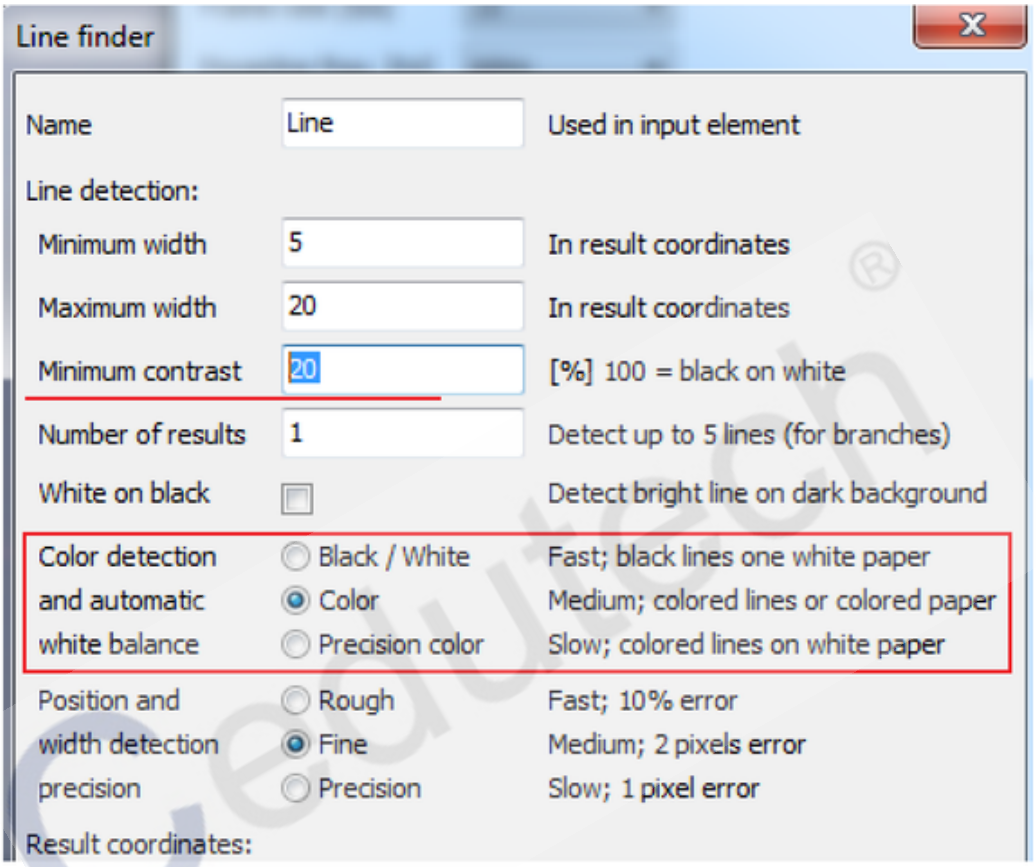



选择“确定”，然后再次激活“实时预览”。 观察发生了什么？

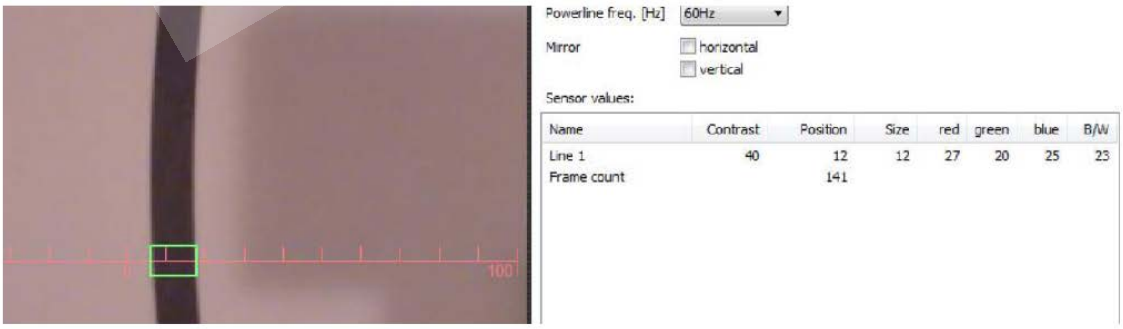


我们注意到，当“最小对比度”值设置的高于实测对比度时，我们的摄像头现在对黑色线条“视而不见”。

取消“实时预览”，再次调出属性面板，将“最小对比度”的值重置为 20，在“detection and automatic white balance（颜色检测和自动白平衡）”选项区中，选中“Color（颜色）”单选按钮：




点击确认，打开“实时预览”再次观察监测效果：



这一次你应该能看到更多的信息。你会注意到在“传感器值”窗口中有更多的颜色信息。这些属性值将允许我们决定要追踪的颜色线条。 现在，将不同的颜色线条移动到摄像头的下方，并记录下不同的数据：

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Black Line | Name | Contrast | Position | Size | red | green | blue | B/W |
| | Line 1 | 40 | 12 | 12 | 27 | 20 | 25 | 23 |
| Red Line | Name | Contrast | Position | Size | red | green | blue | B/W |
| | Line 1 | 27 | 13 | 12 | 52 | 25 | 32 | 34 |
| Green Line | Name | Contrast | Position | Size | red | green | blue | B/W |
| | Line 1 | 37 | 4 | 12 | 23 | 26 | 25 | 25 |

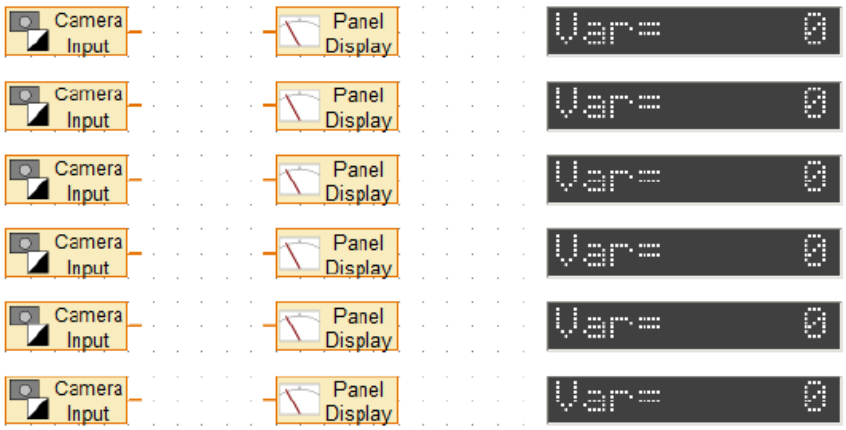
在我们编写程序时，这些数据可以帮助我们进行条件设定。另外需要注意，不同的室内的照明会对摄像头的循迹造成影响（监测数值会发生变化）。

## 程序测试

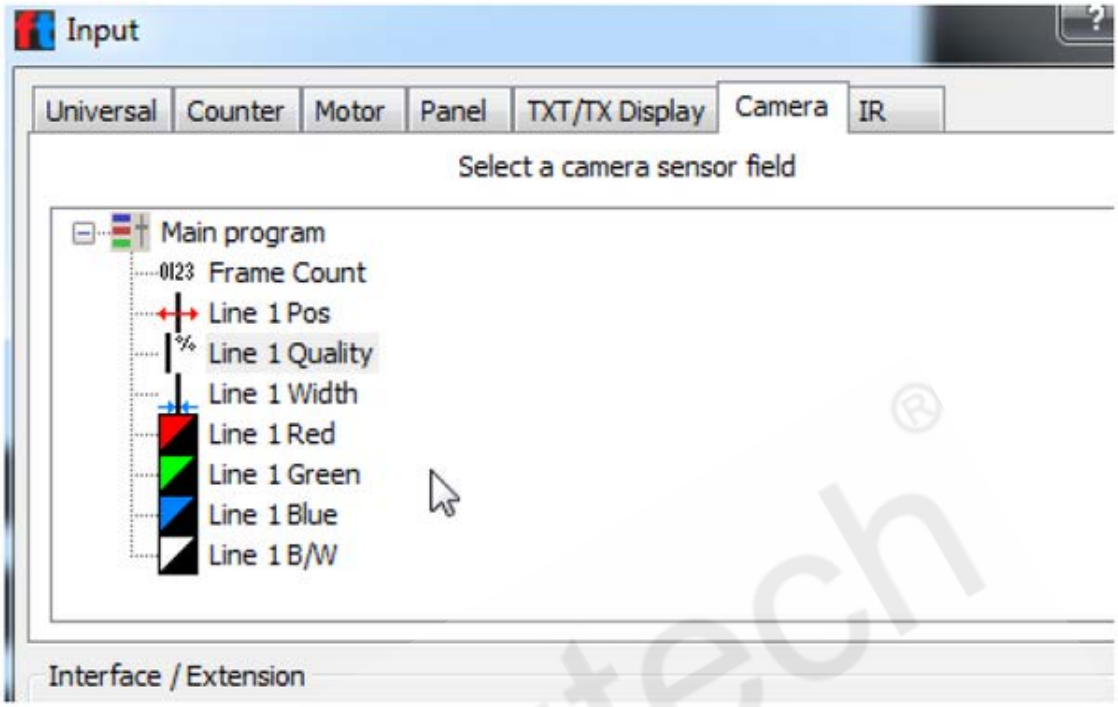
切换到编程界面，从“输入，输出”子分类中拖拽“摄像头输入”和“面板显示”模块到程序窗口中：

从操作模块下的“显示”子分类中，拖拽一个“文本显示”模块到程序窗口中：

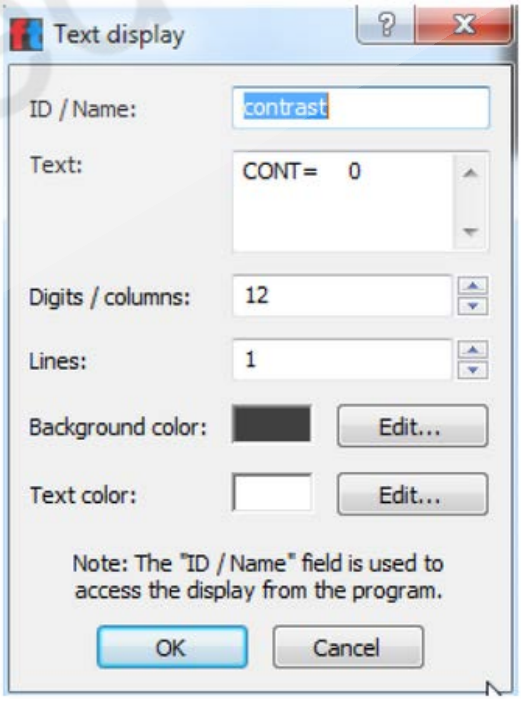
复制上面的三个模块如下：



右键单击每一个“摄像头输入”模块，在其属性面板上，为其指向“监测标尺”返回的每一个属性：

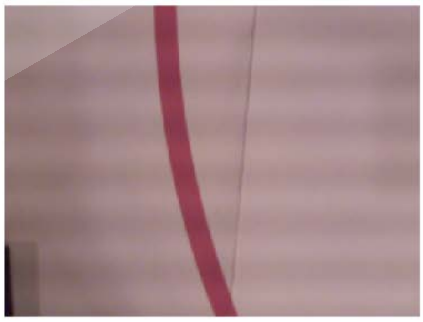



右键单击每一个“文本显示”模块，为其设置相应的命名，并修改默认的文字显示：

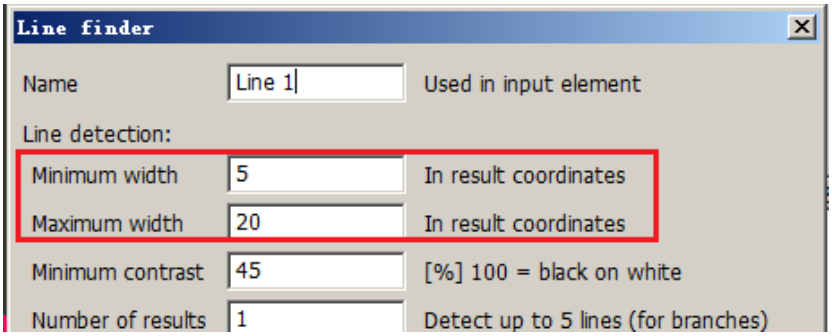


之后，右键单击每一个“面板显示”模块，将其指向对应的“文本显示”模块，然后画出流程线，将程序设置成下面的样子：

从“操作模块”分类中拖拽一个“Camera viewer( 摄像头查看器 )”模块到编程界面中，然后点击运行该程序，然后将画有不同颜色线条的纸张放置到摄像头下方，这时我们在程序界面中可以看到拍摄到的画面以及监测到的线条的各个属性值：



通过测试你会发现在刚才的那种属性配置下，摄像头可以监测到任何颜色的线条。在“监测标尺”的属性面板中我们可以通过设置线条的宽度来追踪特定宽度的线条：

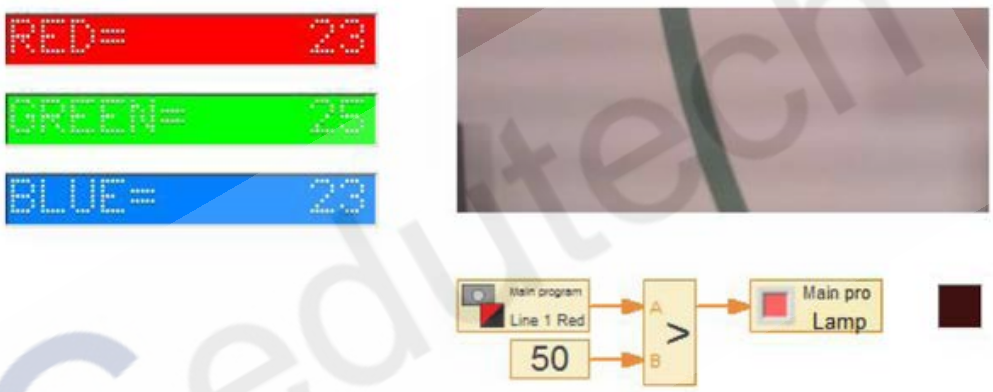


但是，有些时候我们想追踪指定颜色的线条而屏蔽掉其他的颜色，这要怎么做呢？

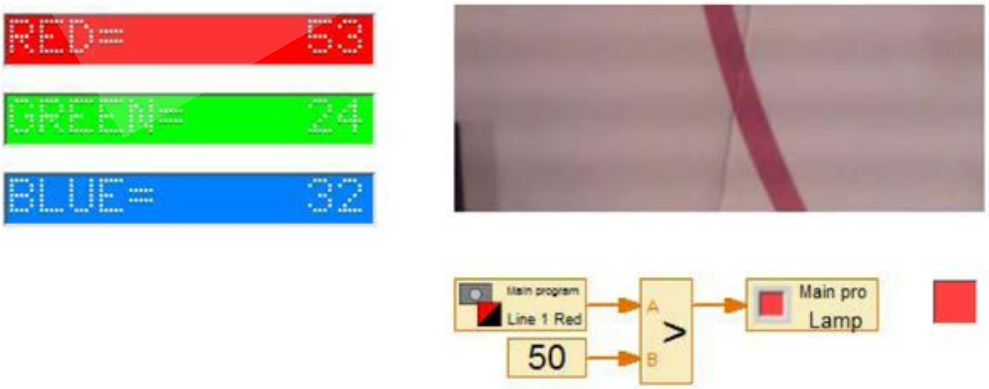
很简单，假设我们要监测红色的线条而排除其他的颜色，那么通过数据分析，我们发现红色线条的“Red”通道返回的值是唯一大于 50 的数值，那么我们用这个属性值作为判定条件：

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Black Line | Name | Contrast | Position | Size | red | green | blue | B/W |
| | Line 1 | 40 | 12 | 12 | 27 | 20 | 25 | 23 |
| Red Line | Name | Contrast | Position | Size | red | green | blue | B/W |
| | Line 1 | 27 | 13 | 12 | 52 | 25 | 32 | 34 |
| Green Line | Name | Contrast | Position | Size | red | green | blue | B/W |
| | Line 1 | 37 | 4 | 12 | 23 | 26 | 25 | 25 |

我们只需设置一个运算符来检查一个大于 50 的红色值即可：

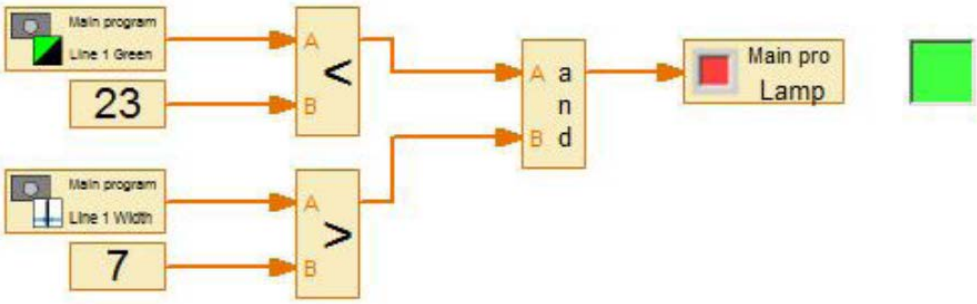



运行该程序，之后线条的红色通道返回的值大于 50 的时候，程序才识别这个线条：



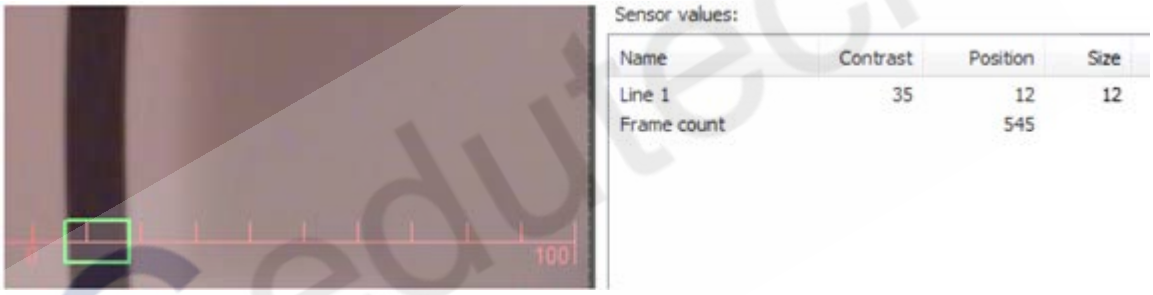

通过上述这种方法，我们可以让程序只注意到一种颜色的线条，再重复一遍这个逻辑，比较不同颜色线条的返回值，查看指定线条的参数值与其他线条的参数值有何不同，当一个 RGB 通道的数值不足以定位到该线条时，我们可以使用多个参数来设置判定条件。多参数

判定的程序编写范例如下：

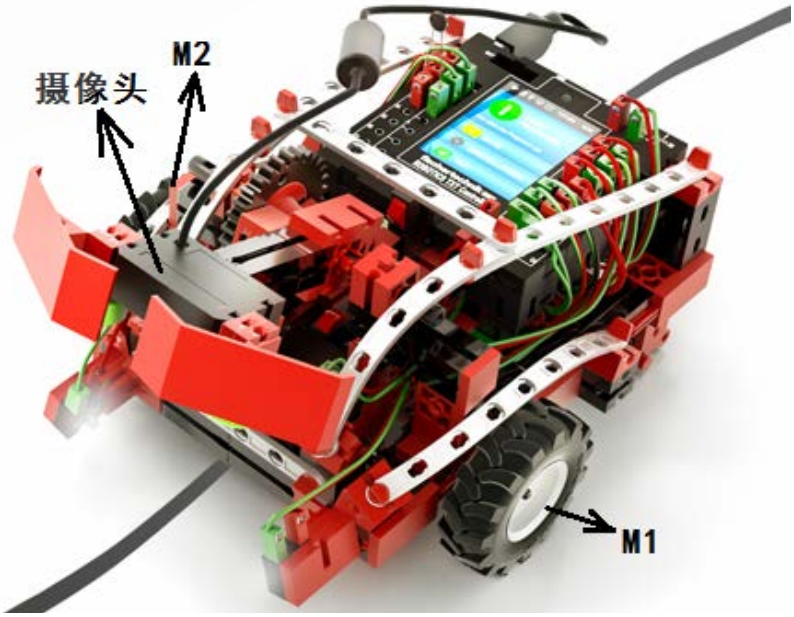



下面我们学习使用“position（位置）”这个属性，通过这个属性我们可以实现摄像头对线条的循迹。

如下图所示，我们在前面也提高过，线条在标尺上会有一个位置值，这个值可以是负的，也可以是正的，如果线条正好在标尺的中心点上，那么显示为 0，我们正是通过它具有的这三种数值情况来进行条件判断的：

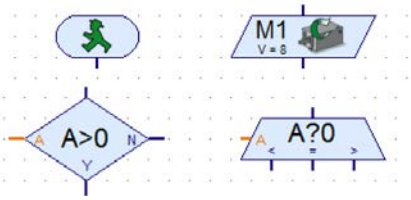



参考下面的这张图片，小车在行进的过程中，我们可以通过 position 的值来决定车轮该向哪个方向偏转（通过左右两个马达的速度差来实现）：

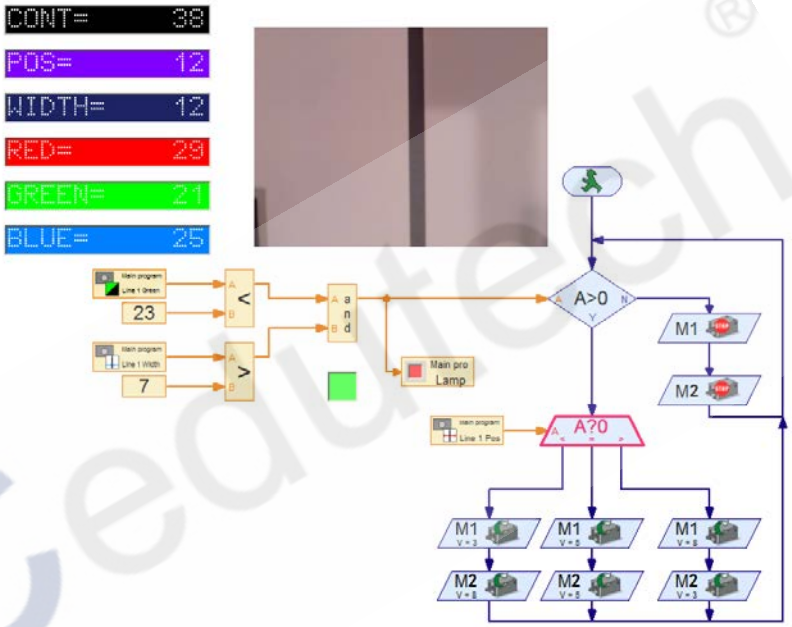


我们不需要实际搭建上图这个实物，你可以仅仅在 TXT 控制器上接入 2 个马达来简单地测试一下这个功能，了解其原理就可以了。

向编程窗口中拖拽下面这四个模块：



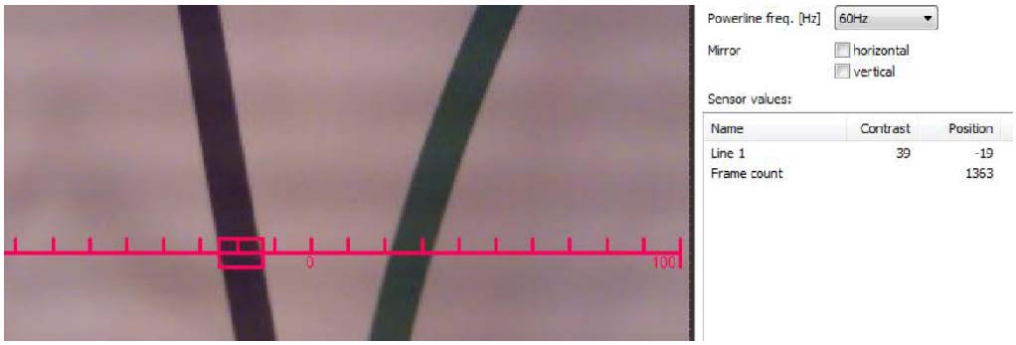

再复制 6 个“马达输出”模块，修改其属性，将你的程序修改成下面的样子：



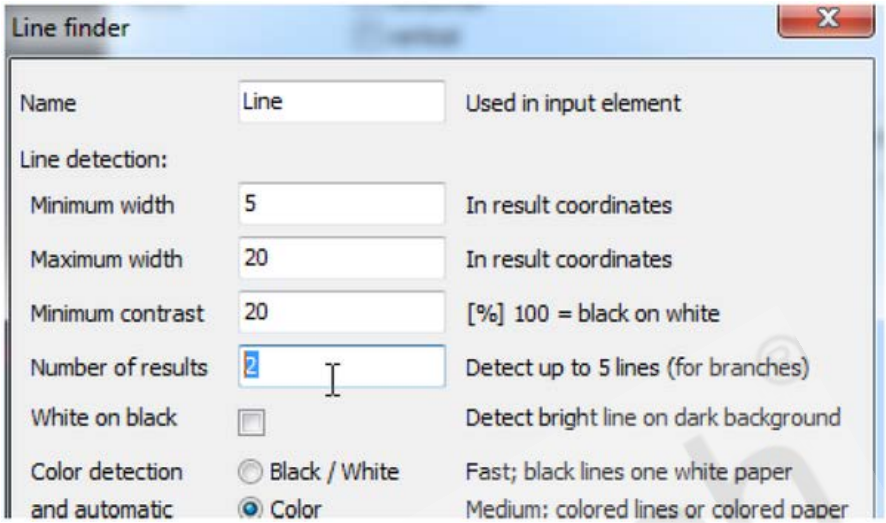

上面这个程序的意思是实现对黑色线条的循迹，首先通过多条件判断来确定是黑色线条，然后根据线条在“监测标尺”上的位置来决定 M1 和 M2 的马达转速，由于两个马达存在速度差，因而可以实现小车的方向调整。

下面我们继续学习“Multiple lines（多线条）”属性的应用。

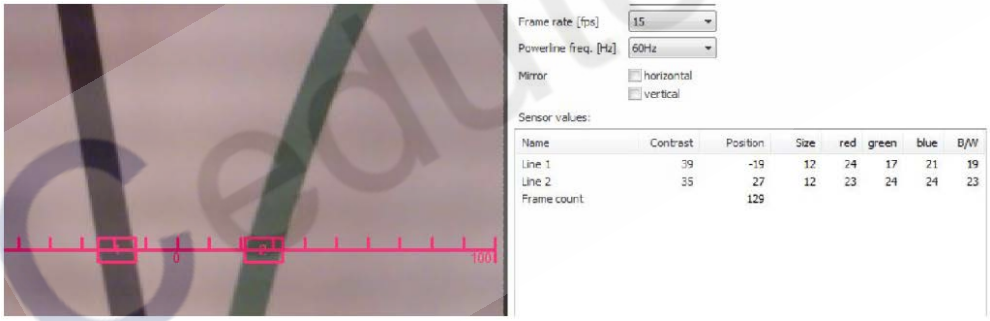
切换到摄像头窗口，在摄像头的下方放置一张含有两条线段的纸张，打开“实时预览”：



默认的情况，我们的摄像头仅识别到一条线。现实中，当我们要在两条线段中做出到底循迹哪条线的决策前，我们必须首先要都看到这两条线才行。现在关闭“实时预览”，点开“监控标尺”的属性面板，将“Number of results（监测条数）”的值改为 2：



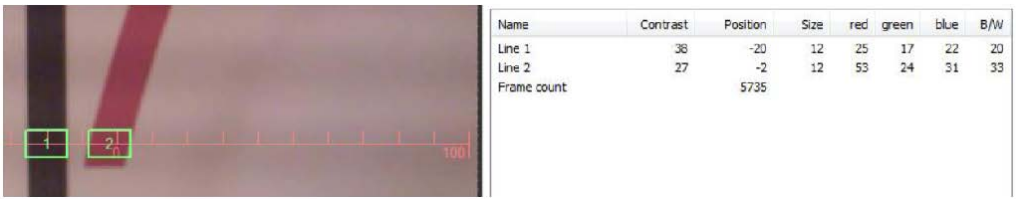

点击确定， 打开“实时预览”，你将看到两条线段都被识别到了：



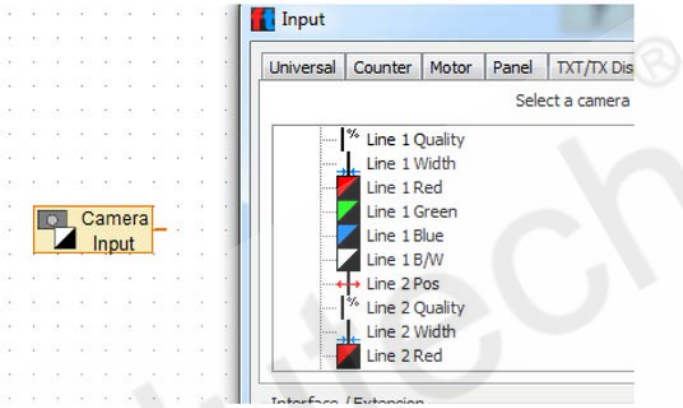

对于多条线段，摄像头会将左侧，也就是标尺上负坐标一侧最先出现的线段命名为 line 1,然后从左向右依次类推：line 2、line3、lin4...

制作一张下图中的纸张，然后沿着黑线滑动摄像头。 注意它看到了什么：

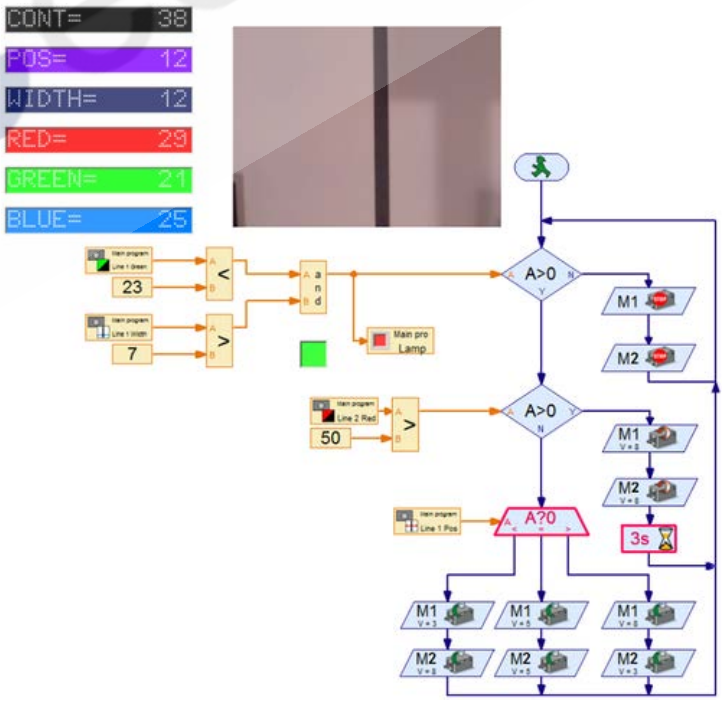
当新的红色线段划过标尺时，摄像头立刻识别到了这个线段：




现在切换到编程窗口，再拖拽一个“摄像头输入”模块，右键单击打开属性面板，你将看到这里列出了更多的属性名称（两条线段的属性）：




现在修改我们的程序为下图，解释一下这个程序的逻辑是要实现什么？




## 实验

考虑一下，用摄像头可以实现简单的条码扫描功能么？