# 



## 《模拟传感器:颜色传感器》

				8
ID	名称	中文名称	数量	图片
1	TX Controller	TX 控制器	1	
2	Power Supply	电源	1	
3	Wires	电线	1	
4	Color Sensor	颜色传感器	1	C C C C C C C C C C C C C C C C C C C
5	Building block 30	30mm 构建块	4	
6	Link 15	15mm 榫键	2	15
7	Aluminum Strut 90	90mm 铝型材	1	90

### 二、<u>实验流程:</u>

光学颜色传感器可以测量物体表面的颜色或光源的特性。测试数值取决于房间照明、待测表面的距离 和待测物体的形状。

光学颜色传感器是一个有源传感器,需要连接 9V 电源,以使其正常运作。传感器由一个明亮的 LED 光源和一个捕获反射光的接收器组成。所有被测物体都会根据自身颜色和表面的纹理吸收不同量的光。



颜色传感器带有三根线缆,其中红色的线缆(电源正极线)接到控制器上的 9V 输出端,绿色的线缆可以接到控制器上的任何一个接地端子,在本实验中接到 I1 的接地端子。黑色的线缆是信号线,本实验中接到 I1 的输入端子。三根线缆的接线位置如下图所示:



搭建一个传感器支架,结构类似于下面的示意图:



整体示意图如下:



无论何时当你把一个传感器连接到控制器时,都需要在 ROBO Pro 软件的接口测试窗口中进行正确的 传感器输入类型设置,只有配置正确,软件才会返回正确的传感器测试数值。

本实验中,我们将 I1 输入端口的类型设置为"Analog 10V (color sensor)",如下图所示:

Interface test	3 X
Inputs / Outputs Info	
Inputs I1 V 2273 (Analog 10V (Color sensor)	Outputs: M1 mode steps © ccw () Stop () cw
I2 0 Digital SkOhm (Switch,)	M1
I3 🔲 0 Digital SkOhm (Switch,) 🔻	M2 mode steps  Ccw  Stop  cw
I4 🔲 0 Digital 5kOhm (Switch,) 🔻	03+04 0 512 8
I5 0 Digital SkOhm (Switch,)	M3 mode steps ccw  Stop  cw

#### 三、实验结论

使用四种不同颜色的片状物体或纸张。将每一个插入你的传感器测试结构中,记录软件界面中返回的数值。



下面是四组测试结果:



#### 四、程序编写:

下面这个程序是一个分类程序。软件让我们知道当前的被测纸张是什么颜色的,并且在程序界面中通 过模拟的 LED 灯显示其颜色。



上面这个程序是如何编写的呢,详情如下:

在 RoboPro 软件中建立一个新的文件。设置编程环境【Environment】为"TX/TXT 控制器",设置 【级别】为"3级:变量"。在【COM / USB】设置界面进行配置:"USB 连接方式和 TX 控制器"。



在类别【Program elements / 程序模块】中的子类别【Basic elements / 基本模块】中找到下面的 两个模块拖至到编程窗口:



从子类别【Commands / 指令】中放置一个"指令"元素:



从子类别【Branch, wait... / 分支, 等待】中放置一个"带数据输入的分支"模块:



从子类别【Inputs, outputs/输入,输出】中拖拽以下这两个模块到程序窗口:



从类别【Operating elements / 操作模块】中的子类别【Displays / 显示】中将下列模块放置在程序窗口中:



在编程窗口中复制"指令"元素,复制方法是按住Ctrl键,在要复制的模块上点击鼠标左键:



右键分别单击上面的两个"指令"模块,在其属性中,一个设置为"On/开",一个设置为"Off/关":

	Command elemen	t ? X
	Command:	<b>+</b>
· · ·	Value:	CW CCW Stop
 Ц	Description of value:	On Off
1>-	Data type:	=
	Integer -3276732	• .
	Floating point 48bit	Append Remove
	Data input for comm	Swap Reset Distance
	ОК	Synchronous +

+

对这两个"指令"模块,分别各复制出3个,共计8个"指令"模块:



再复制三个"带数据输入的分支"模块,共计4个。分别在属性面板中设置这四个模块的条件判断参数,使得我们在前面所测量的四张纸片的颜色数值分别在这四个参数形成的数值区间内。

 Branch
 Condition:
Input value > 1850
Data type: >=
 Integer -32 = 7
Floating poi
 Swap Y/N branches
 Leave Y/N branches as they are
 🔘 Swap Y/N branches
 OK Cancel

右键单击 "Text display / 文本显示"模块,并设置显示框名称为 "sensor",默认的显示文字为 "Sensor= 0",将显示框的列宽度(显示位)设置为14,如果愿意的话,还可以改变背景颜色。

Sensor= 0			
Text display	? ×		
ID / Name:	sensor		
Text:	Sensor= 0		
	-		
Digits / columns:	14		
Lines:	1		
Background color:	Edit		
Text color:	Edit		
Note: The "ID / Name" field is used to access the display from the program.			
ОК	Cancel		

再复制 3 个 "Display Lamp / 显示灯"模块。分别给每个设置一种与前面测试纸片所对应的颜色和 名称。它们分别是 "yellow"、"white"、"orange"、"blue"。

	•		•		:	1	📸 Display Lamp 🛛 🕄 🗾 🏹
•	•	•	•	•			ID / Name: yellow
	•	•		•	•	l	Color: Edit
							Note: The "ID / Name" field is used to
•	•	•	•	•	:		access the lamp from an input element.
					•		OK Cancel
				•		·	

再复制出 7 个 "Panel display / 面板显示"模块,分别在其属性面板中配置到不同颜色的显示灯上。



用流程箭头连接各模块。完成后,你的程序应该看起来类似于下面的显示。



现在将 TX 控制器连接到计算机,打开控制器的电源并运行程序。

将不同颜色的纸张在传感器测试结构上依次插入,查看程序的运行结果,如果程序反馈了不正确的颜 色显示,那么请调整"带数据输入的分支"模块的参数设置。