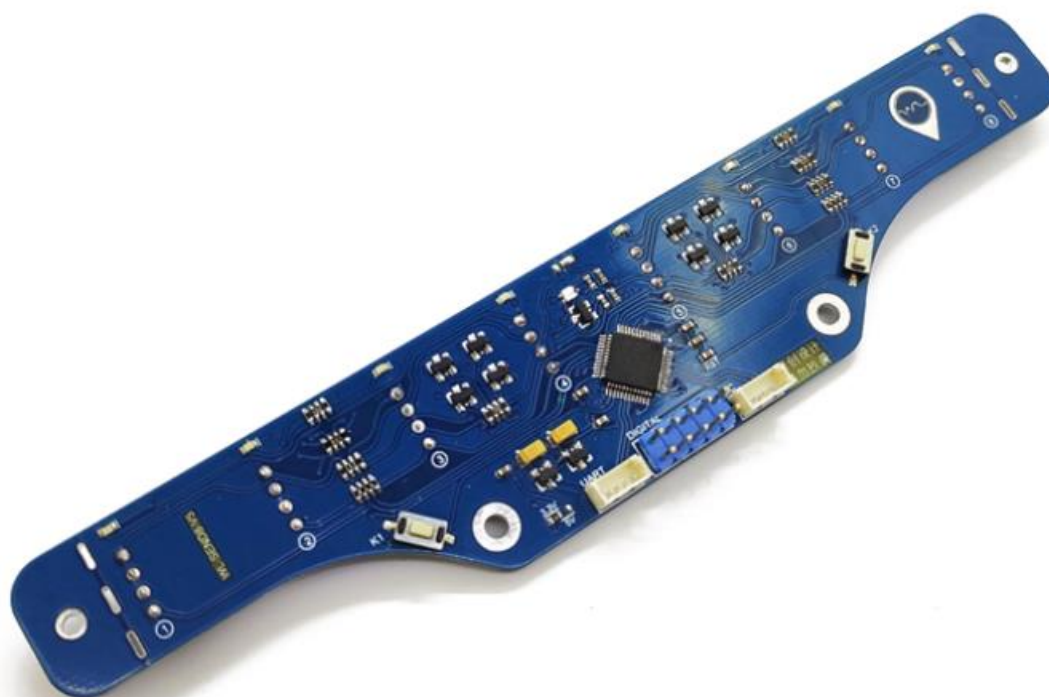


第五代数字量智能灰度传感器

说明书



作者：未来世界

修订时间：2024 年 08 月 26 日

目录

一 传感器背景.....	3
二 传感器参数.....	4
2.1 原理介绍.....	4
2.2 光源知识.....	4
2.3 功能描述.....	5
2.4 技术参数.....	5
2.5 尺寸图（单位：mm）.....	6
2.6 接口介绍.....	7
2.7 算法.....	8
2.8 数据刷新频率.....	8
三 硬件与调试.....	9
3.1 硬件介绍.....	9
3.2 IIC 通信与显示屏切换.....	10
3.3 RGB 指示灯状态说明.....	11
3.4 恢复出厂设置.....	11
四 数据传输方式.....	12
4.1 串口输出.....	12
4.2 数字口输出.....	12
4.3 IIC 输出.....	12
五 通信协议.....	14
5.1 串口通信.....	14
5.1.1 串口通信格式.....	14
5.1.2 串口数据输出格式.....	14
5.2 数字口并行输出.....	20
5.3 IIC 通信.....	20
5.3.1 设置模块地址/ID.....	21
5.3.2 获取灰度数据.....	21
六 传感器调试.....	23
6.1 按键调试.....	23
6.2 串口命令调试.....	24
6.3 灵敏度调试.....	24
七 传感器循迹技巧.....	25
7.1 循线解决方案.....	25
7.1.1 走直线和圆形线路.....	25
7.1.2 走弧形弯道.....	25
7.1.3 判断“T”字形路口.....	26
7.1.4 判断“十”字形路口.....	26
7.1.5 判断不规则路口.....	27
7.2 转向解决方案.....	27
八 版本历史记录.....	27

一 传感器背景

世界正处在科技革命和产业革命的交汇点上,科学技术在广泛交叉和深度融合中不断创新,以信息、生命、纳米、材料等科技为基础的系统集成创新,以前所未有的力量驱动着经济社会发展。而且,随着信息化、工业化不断融合,以机器人科技为代表的智能产业蓬勃兴起,成为现代科技创新的一个重要标志。

机器人运行时需要不断地循迹,即需要借助传感器探测地面色调迥异的两种色彩以修正其运动轨迹。目前,市场上广泛使用的传感器有颜色传感器、光敏电阻灰度传感器和激光传感器,其中,颜色传感器容易受外界光线影响,需要在黑暗环境下使用,而且颜色传感器获得的信号是反应 RGB 三色的复杂数据信号,因此,颜色传感器的通信过程非常复杂,更重要的是价格昂贵。光敏电阻灰度传感器同样易受外界光源的干扰,而且稳定性差,采集的灰度对比值偏差较大,机器人循迹时容易出错。由于激光传感器的接收器是接收激光的散射光,当多个激光传感器在同一空间循迹时,激光传感器会相互干扰,导致机器人循迹失败。于是智能版灰度传感器应运而生。

二 传感器参数

2.1 原理介绍

灰度传感器有数字传感器和模拟传感器两种，其原理大致相同。原理：一只发光二极管和一只光敏二极管，安装在同一面上。灰度传感器利用不同颜色的检测面对光的反射程度不同的原理进行颜色深浅检测。在有效的检测距离内（理论上距离可以无限远，实际受外界光源的影响，最佳距离为 10mm 至 50mm，不推荐距离过高，有太阳光时需要进行遮光），发光二极管发出的光，照射在检测面上，检测面反射部分光线，光敏二极管检测此光线的强度并将其转换为单片机可以识别的电信号。这个电信号是一个模拟值，单片机可以根据模拟值的大小进行二值化处理，也就是给一个电平分界线，当电压大于一个值的时候给一个高电平（或低电平），当电压小于一个值的时候给一个低电平（或高电平）。当然这里可以用电压比较器作为电平的参考电压。这就是我们用的数字量灰度传感器。数字量的灰度传感器如果加入的是单片机处理数据，那么可以把得到的原始数据加入滤波算法、数据混合算法得到一个波动小、适应环境强的数字传感器。

如果对采集的原始的数据不做处理，那么就是模拟值输出。也就是我们的模拟量灰度传感器。因为每路传感器相互之间都是有误差的，所以同样的条件下模拟值肯定不是一样。（关于模拟量传感器的算法处理请参考模拟量智能版）

在数字量和模拟量外，我们又创新出一种新的数据形式——偏移量。偏移量又称误差，所谓的偏移量就是输出传感器在循线所处的位置的值。其实偏移量是源自我们经常用到的 PID 算法，P 值是误差值的系数，I 值是误差值的积分系数，D 是误差值的微分系数。所以 PID 实际就是对误差的处理，我们通过单片机把这种误差通过运算直接可以通过串口输出，省去了我们测量模拟值然后再进行误差计算的过程。我们可以把得到的偏移量直接用于 PID 算法，PID 算法和速度结合，也就成了 PID 循线。

如果要减小外界光线对数据的影响，可以传感器上加[遮光罩](#)。

2.2 光源知识

为什么选择白色光？白光是有红、绿、蓝三原色组成。理论上白色可以分成不同颜色的光（比如三棱镜试验）。所以选择白光作为补光源是最合适的选择。白光可以在所有色域的物体上返回光线（黑色除外，理论上黑色可以吸收所有的光）。

以绿色为例，物体为什么会是绿色的呢？因为光线照在物体上，这个物体不吸收绿色的光，所以把绿色反射出来，反射到眼睛里，人眼就看到了绿色的光。

那么颜色又是什么意思？不同的波长的光，在人的眼睛里被识别成颜色。比如波长范围:577~492 纳米的光线，反射到人的眼里就被眼睛认为是绿色的。综上所述，我们看到的物体之所以是绿色的，是因为物体本身把 577~492 纳米的光线没有吸收，这些光反射到人的眼睛里，大脑就把它认为是绿色的。

我们传感器怎么识别颜色的呢？灰度传感器采用的是光敏三极管，有光子反射到光敏三极管的基极上，使集电极和发射极导通。这里的光子是不分颜色的，光子发射到基极的越多，集电极和发射极的导通量越大，这种导通量被我们用电压的大小来表示。电压的大小我们通过单片机的 ADC 转换器变成我们的可以识别的数字信号。例如：我们要识别绿色和白色，白光照在绿色上，那么绿色的光就会反射回来，绿色的这部分光子打在光敏三极管的基极上，

然后会产生一个电压 $A1$ ；白光照在白色上，那么白色理论上是不吸收光的，所以所有的光都会反射回来，所有的光子都会打在光敏三极管的基极上，然后会产生一个电压 $A2$ 。那么 $A2$ 的光子返回的多， $A1$ 的光子返回的少，所以 $A2$ 肯定是大于 $A1$ 的。这样我们就可以区分出两种颜色了。

仔细推敲可以看出，光敏三极管，只能判断光子返回的多少，并不能判断是什么波长的光子返回来的，所以灰度传感器只能识别颜色的深浅，不能判断具体的颜色。这也就是灰度的定义：物体反射出光子的多少，我们称之为灰度。

拓展内容：1.世界上所有的反射都不是全反射，所以绿色的物体不只是反射绿色，可能 99.9%返回的绿色，0.1%返回的是其他颜色。因为绿色返回的太多，其他的颜色也就微不足道了。2.光敏三极管我们可以理解为一个水龙头，基极就相当于水龙头的开关，集电极相当于进水口：流入，发射极相当于出水口：流出。通过开关闭合的多少控制水的流量。

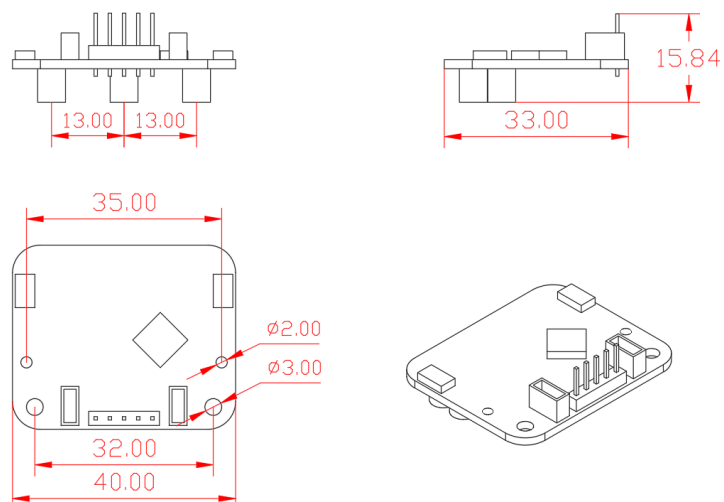
2.3 功能描述

- a) 五种数据输出：数字量、偏移量、模拟量、出线数据检测、压线数量检测
- b) 三种通信方式：UART、数字口并行输出、IIC
- c) 三种巡线模式：亮灯模式、灭灯模式、自适应模式
- d) 偏移量可以输出高精度误差数据。
- e) 内置 32 处理芯片，高速的数据处理能力
- f) 采用数据混合算法、滤波算法，适应能力更强
- g) PID 巡线算法，可以在短时间内修正巡线路径。
- h) 数字量输出不同颜色上的电平可以设置
- i) 模块可设置 ID 编号，IIC 可以多个模块并联接入一个主机
- j) 可直接插 12864 显示屏观看状态信息
- k) 内置干扰过滤算法，配合遮光罩可以在多彩灯、频闪灯、红外灯、太阳下使用
- l) 模块可设置波特率：9600、14400、19200、28800、38400、57600、115200 可选
- m) 电源滤波：采用高频滤波钽电容
- n) 自带稳压器，3.3V 至 5V 宽电压供电，推荐 5V
- o) 集成电源保护电路，电源反接不会烧坏电路

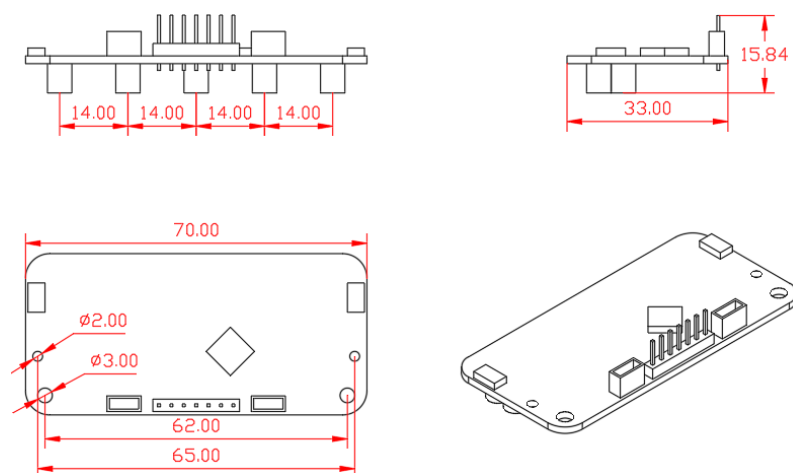
2.4 技术参数

参数名称	典型值	注释
电压	3.3V~5.5V	推荐 5V，选稳定的 降压模块 供电
电流	58mA、78mA、94mA、104mA、141mA、173mA	分别对应 3 路、5 路、7 路、8 路、12 路、16 路
是否有电源保护	有反接保护	
适用机型	Arduino、51 单片机、STM32、K60、树莓派等	PLC 可以用 RS232、485 模块进行电平转换
通信接口	UART	跳线 3.3V 时，数字口高电平信号为 3.3V，跳线 5V 时，高电平信号为 5V
	Digital	
	IIC	

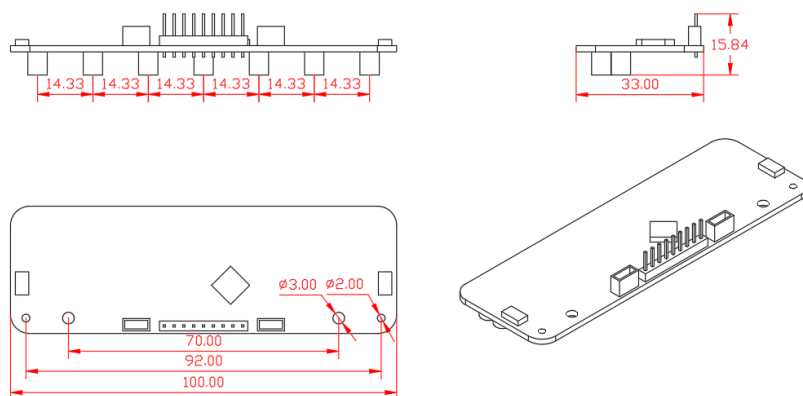
2.5 尺寸图（单位：mm）



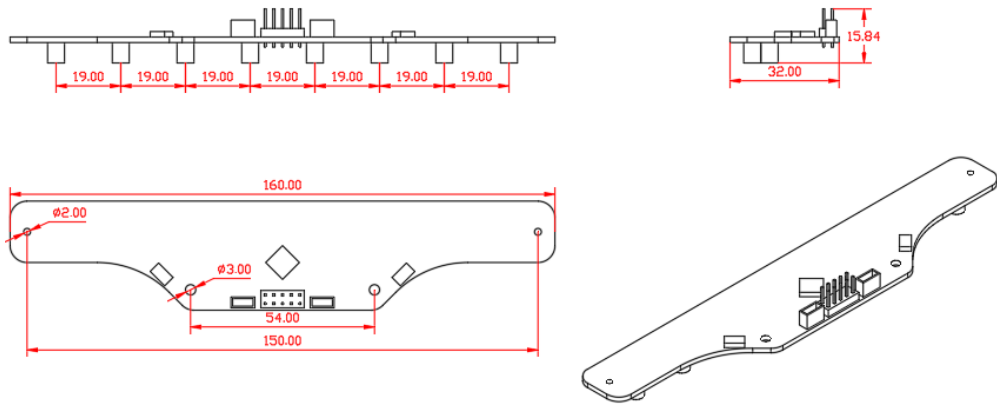
2.3-1 图 3 路数字量灰度传感器尺寸



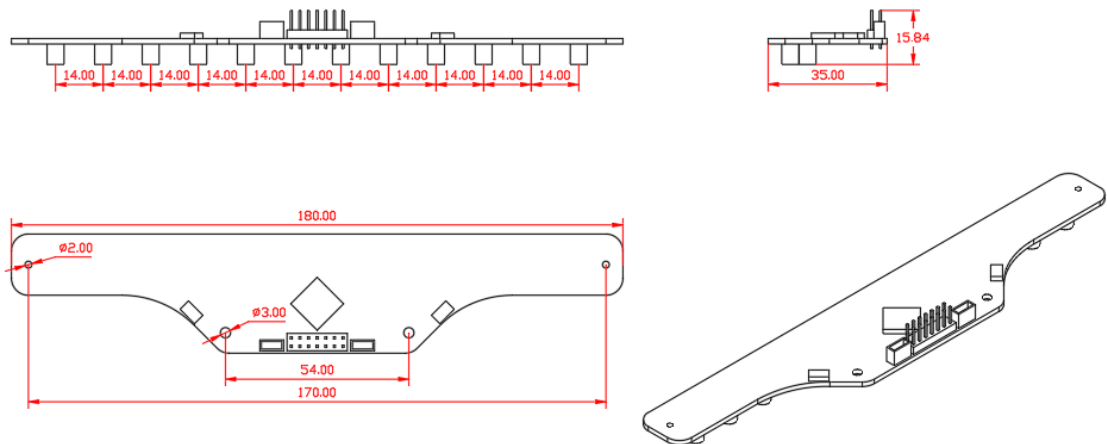
2.3-2 图 5 路数字量灰度传感器尺寸



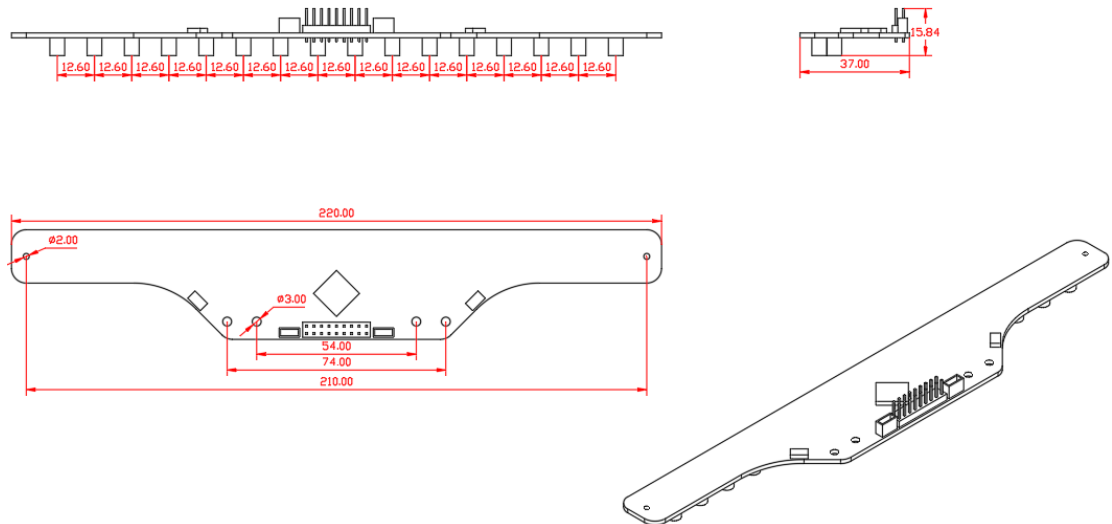
2.3-3 图 7 路数字量灰度传感器尺寸



2.3-4 图 8 路数字量灰度传感器尺寸



2.3-5 图 12 路数字量灰度传感器尺寸



2.3-6 图 16 路数字量灰度传感器尺寸

2.6 接口介绍

数字口高电平状态选择：通过跳线电阻可以改变数字口的高电平电压，电阻接 3.3V 的位置，说明数字口高电平为 3.3V；电阻接 5V 的位置，说明数字口高电平为 5V。无论是 3.3V 的单片机还是 5V 单片机都可以识别 3.3V 高电平信号，默认使用 3.3V 即可，无需修改。5V 电平是为有特殊需求的客户设计。

UART 接口、数字接口和 IIC 接口 3 个任意接一个就可以进行数据的读取。也可以同时接 2 个或 3 个接口同时获取数据。

IIC 接口是 IIC 通信和显示屏的共用接口。默认是作为 IIC 通信使用，使用过程中如果需要显示屏观察数据，可以通过按键设置 IIC 接口作为显示屏接口使用，切换方法见[第三章](#)。

接口功能表：

接口类型	注释
UART 接口	串口通信接口，用于传输数字量、模拟量和偏移值等数据。
数字口	与主机数字口相连，用于主机读取高低电平信号
IIC 接口	默认作为 IIC 通信传输灰度值数据。通过按键可设置为显示屏接口，接 0.96 寸 IIC 通信 OLED 显示屏，主要用在调试过程观察数据变化

2.7 算法

第 5 代灰度传感器，采用的是 32 位单片机进行数据的采集和解析。传感器内置了消抖算法、数据融合算法、滤波算法、抗干扰算法、学习记忆算法等。适用于各种复杂的场地。

供用户巡线的算法有 PID 巡线算法、数字寻迹逻辑、数据融合算法、路口检测。算法的具体应用在视频讲解里查看。

2.8 数据刷新频率

刷新频率对比表：（单位：Hz）

传感器名称	数字量频率	偏移量频率	模拟量频率	全部输出频率
3 路数字传感器	7K	6K	6.9K	6K
5 路数字传感器	4.4K	3.8K	4.2K	3.7K
7 路数字传感器	3.2K	2.8K	3.1K	2.8K
8 路数字传感器	2.8K	2.5K	2.7K	2.4K
12 路数字传感器	2K	1.8K	2K	1.7K
16 路数字传感器	1.5K	1.3K	1.4K	1.3K

三 硬件与调试

3.1 硬件介绍

8 路传感器示意图：

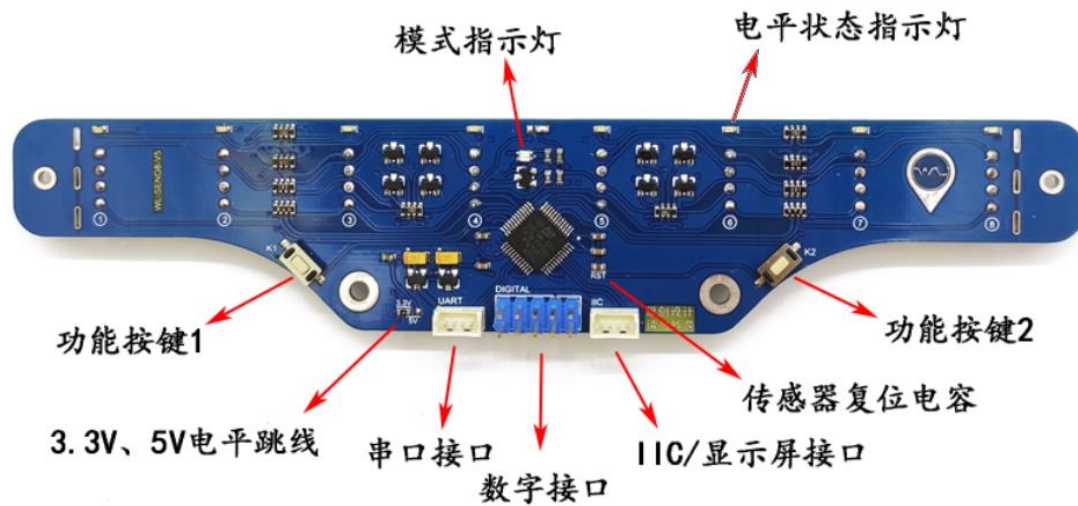


图 3.1 功能示意图

显示屏开机界面示意图：



图 3.2 显示屏开机界面

名称	含义
ADC-N	通道数：3、5、7、8、12、16
Mode	巡线模式：Bright（亮灯模式）、Close（灭灯模式）、Auto（自适应模式）
Dark	暗颜色电平状态：1（暗颜色高电平）、0（暗颜色低电平）
Out	串口数据类型：Digital（数字量）、Deviate（偏移量）、Analog（模拟量）、All（全输出）
ID	传感器地址：0 至 127
Baud	波特率：9600、14400、19200、28800、38400、57600、115200

显示屏采集颜色结束界面示意图：

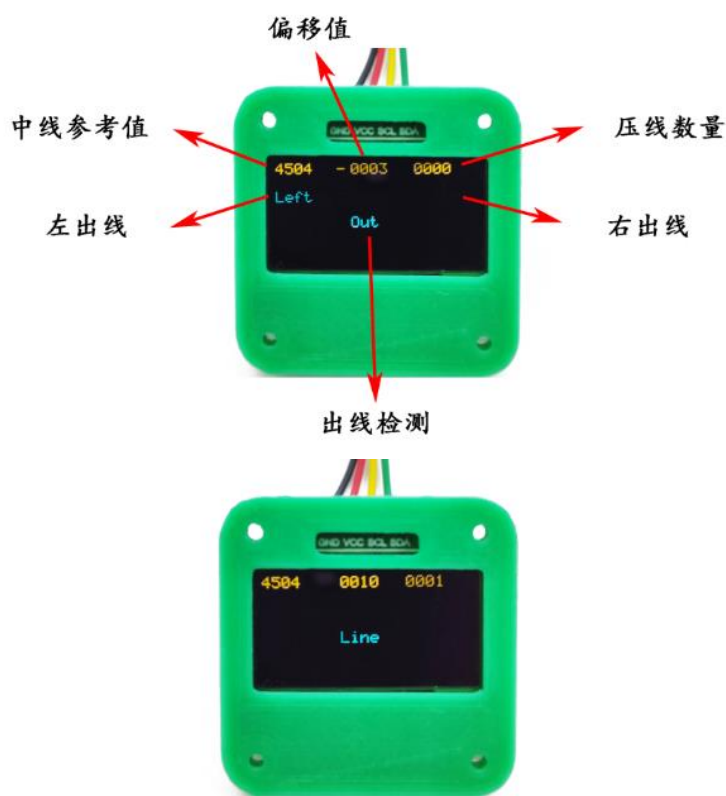


图 3.3 显示屏采集颜色结束界面

名称	含义
中线参考值	理论值：3 路 2000，5 路 3000，7 路 4000，8 路 4500，12 路 6500，16 路 8500
偏移量	线靠近传感器中间为 0，靠左为负，靠右为正
压线数量	取值：0 至 16
左出线	线从传感器左侧脱线：left
右出线	线从传感器右侧脱线：right
出线检测	线不在传感器下方：Out，线在传感器下方：Line。线不可过细，否则在两灯之间检测不到

注释：暗颜色（深色）是指多种颜色作对比时，反光能力最差的那个颜色称为暗颜色（深色）。比如黑色和白色相对比，黑色反光差，所以黑色在两种颜色中叫暗颜色（深色）；再比如红色和白色相对比，红色反光差，所以红色在两种颜色中叫暗颜色（深色）。

3.2 IIC 通信与显示屏切换

IIC 接口默认模式是作为数据传输的接口。如果想接显示屏显示传感器信息，可以通过按键切换为显示屏显示模式。具体方法如下：

首先保证使用的显示屏是 0.96 寸 IIC 通信 OLED 显示屏（其他显示屏无法使用），然后把显示屏按线序接在传感器上。方法一：接通电源，此时传感器 IIC 接口处于数据传输模式。长按功能按键 K2，然后短接一下复位电容 RST，当显示屏亮起后松开 K2 按键（此时已进入显示模式）。方法二：长按功能按键 K2，接通电源，当显示屏亮起后松开 K2 按键（此时已进入显示模式）。

进入显示模式后，会看到如图 3.2 所示的内容。如果显示的内容有错行（错行是由于按键按下时有抖动引起初始化失败引起），请重新上电按上述流程再操作一遍即可。

3.3 RGB 指示灯状态说明

功能	指示灯状态
数据输出类型为数字量	开机通电，RGB 指示灯不亮
数据输出类型为偏移量	开机通电，红色指示灯亮一下
数据输出类型为模拟量	开机通电，绿色指示灯亮一下
数字量、偏移量、模拟量全部输出	开机通电，蓝色指示灯亮一下
亮灯模式	巡线状态，RGB 指示灯不亮
灭灯模式	巡线状态，绿色指示常亮
自适应模式	巡线状态，蓝色指示常亮

3.4 恢复出厂设置

恢复出厂设置有两种方式：第一种是通过按键进行出厂设置，第二种是通过串口发送复位命令进行出厂设置。

按键恢复出厂设置：开机状态下，同时按下功能按键 K1 和 K2，长按 4 秒钟左右。模式指示灯红灯亮起，松开按键，红灯亮起 1 秒钟，然后绿灯和蓝灯依次闪烁一下，复位完成。

串口恢复出厂设置：为了观察调试是否成功，我们可以选用串口助手发送复位命令。首先把我们的传感器通过[串口模块](#)连接在电脑上。选择一款串口助手设置好串口号、波特率等参数，在 HEX 模式（十六进制）下发送数据 4C 00 0B 01，然后我们也用 HEX 模式（十六进制）接收数据，如果返回 0B，说明恢复出厂设置成功。界面设置如图 3.3 所示

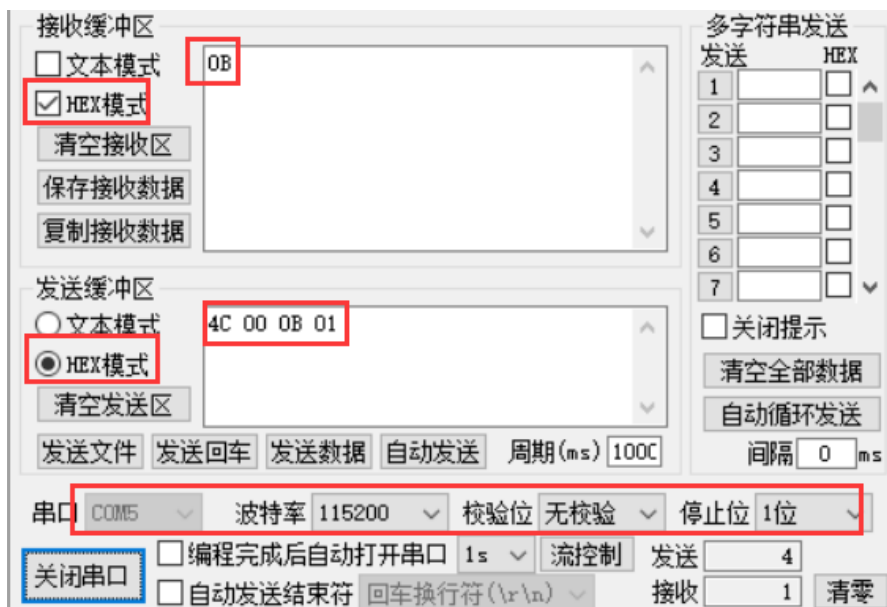


图 3.4 上位机界面设置

四 数据传输方式

4.1 串口输出

串口输出方式是指主机通过 UART 接口发送命令信号给灰度传感器，灰度传感器接收到信号后发送数据给主机的通信方式。适用于单个模块与主机串口进行通信。通信协议参考[第五章](#)。

通信示意图：

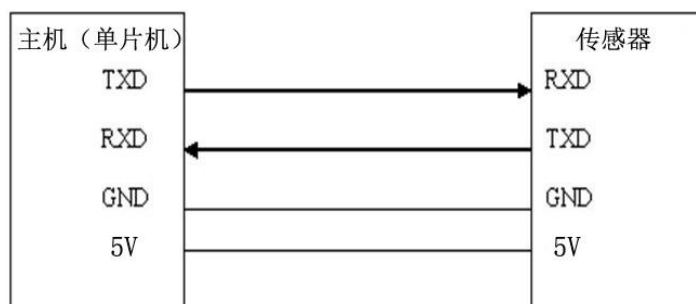


图 4.1 串口通信连线图

4.2 数字口输出

数字口输出方式是指灰度传感器通过 Digital 接口发送数据给主机的通信方式。数字口输出方式适用于单个模块与主机普通 IO 口进行通信。通信协议参考[第五章](#)。

通信示意图：

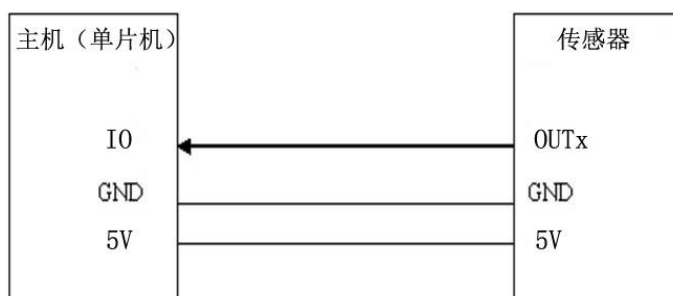


图 4.2 数字口连线图

4.3 IIC 输出

IIC 输出方式是指主机通过 IIC 接口发送传感器编号给灰度传感器，灰度传感器接收到信号后发送数据给主机的通信方式。适用于多个模块与主机进行通信。通信协议参考[第五章](#)。

IIC 通信时，数据输出类型和需要读取的数据一一对应。比如想要读取模拟量，那么数据输出类型就要设置为模拟量输出或者数据全部输出。需要通过串口进行设置，具体设置方法参考[表五](#)。

通信示意图：

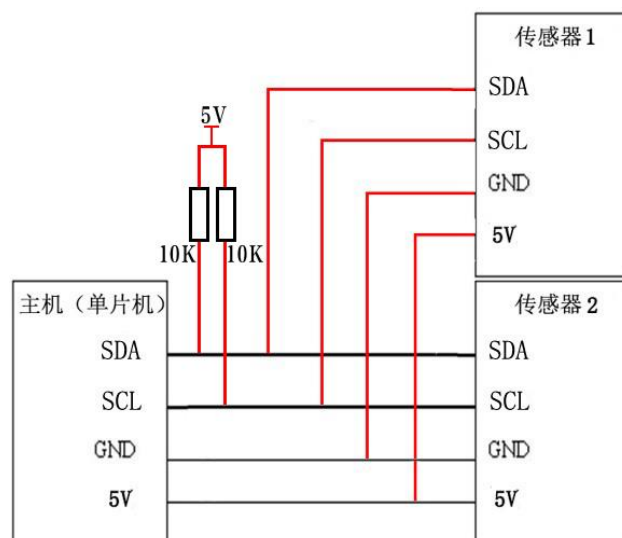


图 4.3 IIC 通信连线图

注意：为提高数据稳定性，主机（单片机）SDA 和 SCL 引脚需要加 4.7K 或 10K 上拉电阻。否则会造成通信失败。除非单片机有上拉输入模式，可以不用加上拉电阻。

五 通信协议

5.1 串口通信

5.1.1 串口通信格式

波特率： 115200（默认） ， 校验位： N ， 数据位： 8 ， 停止位： 1

5.1.2 串口数据输出格式

表一：数字量数据意义对照表

顺序	主机发送 (TXD)	主机接收 (RXD)	注释
1	0x57		起始信号（“W”的ASCII值）
2	ID		传感器编号
3		0x75	帧头
4		Data0	Bit7、Bit6、Bit5、Bit4、Bit3、Bit2、Bit1、Bit0 8路、7路、6路、5路、4路、3路、2路、1路
5		Data1	Bit7、Bit6、Bit5、Bit4、Bit3、Bit2、Bit1、Bit0 16路、15路、14路、13路、12路、11路、10路、9路
6		0x02、0x03	帧尾（3、5、7、8路为0x02，12、16路为0x03）

0x57：命令起始信号，“W”的ASCII值。

0x75：每帧数据的帧头，主机可以通过识别帧头判断数据是否是灰度的数据。

Data0：8位的数据，从低位依次对应1至8路数据，如果是3、5、7、8路传感器，只有一个8位数据Data0。

Data1：8位的数据，从低位依次对应9至16路数据，只有是12路和16路传感器才会有Data0和Data1两个字节数据。

0x02、0x03：每帧数据的帧尾，帧尾代表传感器发送几个字节的数据（不包含帧尾），主机可以通过识别帧尾来判断一帧数据传输是否正确。也可以用识别帧尾和字节数判断数据传输是否完成。

3、5、7、8路一个通信周期有5个字节，12路和16路一个通信周期有6个字节，每个字节8位，这些数据依次按位传送。

表二：偏移量数据意义对照表

顺序	主机发送 (TXD)	主机接收 (RXD)	注释
1	0x57		起始信号（“W”的ASCII值）
2	ID		传感器编号
3		0x75	帧头
4		Data0	1位空闲+1位出线检测+1位偏移量正负+5位压线数量
5		Data1	偏移值高8位

6		Data2	偏移值低 8 位
7		0x04	帧尾

0x57: 命令起始信号,“W”的 ASCII 值。

0x75: 每帧数据的帧头,主机通过识别帧头判断数据是否是灰度的数据。

Data0 ~ 2: 8 位的数据。Data0 的 Bit5 值为 1, 偏移值为正数; Bit5 值为 0, 偏移值为负数。

Data0 的 Bit6 值为 1, 为右出线; Bit6 值为 0, 为左出线。

0x04: 每帧数据的帧尾,帧尾代表传感器发送几个字节的数据(不包含帧尾),主机可以通过识别帧尾来判断一帧数据传输是否正确。也可以用识别帧尾和字节数判断数据传输是否完成。

一个通信周期有 7 个字节,每个字节 8 位,这些数据依次按位传送。

表三: 模拟量数据意义对照表

顺序	主机发送 (TXD)	主机接收 (RXD)	注释
1	0x57		起始信号(“W”的 ASCII 值)
2	ID		传感器编号
3		0x75	帧头
4		Data0	第 1 个通道模拟值高 8 位
5		Data1	第 1 个通道模拟值低 8 位
6		Data2	第 2 个通道模拟值高 8 位
7		Data3	第 2 个通道模拟值低 8 位
8		Data4	第 3 个通道模拟值高 8 位
9		Data5	第 3 个通道模拟值低 8 位
10		Data6	第 4 个通道模拟值高 8 位
11		Data7	第 4 个通道模拟值低 8 位
12		Data8	第 5 个通道模拟值高 8 位
13		Data9	第 5 个通道模拟值低 8 位
14		Data10	第 6 个通道模拟值高 8 位
15		Data11	第 6 个通道模拟值低 8 位
16		Data12	第 7 个通道模拟值高 8 位
17		Data13	第 7 个通道模拟值低 8 位
18		Data14	第 8 个通道模拟值高 8 位
19		Data15	第 8 个通道模拟值低 8 位
20		Data16	第 9 个通道模拟值高 8 位
21		Data17	第 9 个通道模拟值低 8 位
22		Data18	第 10 个通道模拟值高 8 位
23		Data19	第 10 个通道模拟值低 8 位
24		Data20	第 11 个通道模拟值高 8 位
25		Data21	第 11 个通道模拟值低 8 位
26		Data22	第 12 个通道模拟值高 8 位
27		Data23	第 12 个通道模拟值低 8 位
28		Data24	第 13 个通道模拟值高 8 位
29		Data25	第 13 个通道模拟值低 8 位
30		Data26	第 14 个通道模拟值高 8 位

31		Data27	第 14 个通道模拟值低 8 位
32		Data28	第 15 个通道模拟值高 8 位
33		Data29	第 15 个通道模拟值低 8 位
34		Data30	第 16 个通道模拟值高 8 位
35		Data31	第 16 个通道模拟值低 8 位
36		0x07、0x0B、 0x0F、0x11、 0x19、0x21	帧尾（3 路为 0x07，5 路为 0x0B，7 路为 0x0F，8 路为 0x11， 12 路为 0x19，16 路为 0x21）

0x57: 命令起始信号，“W”的 ASCII 值。

0x75: 每帧数据的帧头，主机通过识别帧头判断数据是否是灰度的数据。

Data0 ~ 31: 8 位的数据，每两个 8 位数据合成一个 16 位的数据（最大值是 4095，16 进制表示为 0x0FFF）。3 路缺省 Data6 ~ 31 数据，5 路缺省 Data10 ~ 31 数据，7 路缺省 Data14 ~ 31 数据，8 路缺省 Data16 ~ 31 数据，12 路缺省 Data24 ~ 31 数据。

0x07、0x0B、0x0F、0x11、0x19、0x21: 每帧数据的帧尾，帧尾代表传感器发送几个字节的数据（不包含帧尾），主机可以通过识别帧尾来判断一帧数据传输是否正确。也可以用识别帧尾和字节数判断数据传输是否完成。

3 路一个通信周期有 10 个字节，5 路一个通信周期有 14 个字节，7 路一个通信周期有 18 个字节，8 路一个通信周期有 20 个字节，12 路一个通信周期有 28 个字节，16 路一个通信周期有 36 个字节，每个字节 8 位，这些数据依次按位传送。

表四：全部数据输出数据意义对照表

顺序	主机发送 (TXD)	主机接收 (RXD)	注释
1	0x57		起始信号（“W”的 ASCII 值）
2	ID		传感器编号
3		0x75	帧头
4		Data0	Bit7、Bit6、Bit5、Bit4、Bit3、Bit2、Bit1、Bit0 8 路、7 路、6 路、5 路、4 路、3 路、2 路、1 路
5		Data1	Bit7、Bit6、Bit5、Bit4、Bit3、Bit2、Bit1、Bit0 16 路、15 路、14 路、13 路、12 路、11 路、10 路、9 路
6		Data2	1 位空闲+1 位出线检测+1 位偏移量正负+5 位压线数量
7		Data3	偏移值高 8 位
8		Data4	偏移值低 8 位
9		Data5	第 1 个通道模拟值高 8 位
10		Data6	第 1 个通道模拟值低 8 位
11		Data7	第 2 个通道模拟值高 8 位
12		Data8	第 2 个通道模拟值低 8 位
13		Data9	第 3 个通道模拟值高 8 位
14		Data10	第 3 个通道模拟值低 8 位
15		Data11	第 4 个通道模拟值高 8 位
16		Data12	第 4 个通道模拟值低 8 位
17		Data13	第 5 个通道模拟值高 8 位
18		Data14	第 5 个通道模拟值低 8 位

19		Data15	第 6 个通道模拟值高 8 位
20		Data16	第 6 个通道模拟值低 8 位
21		Data17	第 7 个通道模拟值高 8 位
22		Data18	第 7 个通道模拟值低 8 位
23		Data19	第 8 个通道模拟值高 8 位
24		Data20	第 8 个通道模拟值低 8 位
25		Data21	第 9 个通道模拟值高 8 位
26		Data22	第 9 个通道模拟值低 8 位
27		Data23	第 10 个通道模拟值高 8 位
28		Data24	第 10 个通道模拟值低 8 位
29		Data25	第 11 个通道模拟值高 8 位
30		Data26	第 11 个通道模拟值低 8 位
31		Data27	第 12 个通道模拟值高 8 位
32		Data28	第 12 个通道模拟值低 8 位
33		Data29	第 13 个通道模拟值高 8 位
34		Data30	第 13 个通道模拟值低 8 位
35		Data31	第 14 个通道模拟值高 8 位
36		Data32	第 14 个通道模拟值低 8 位
37		Data33	第 15 个通道模拟值高 8 位
38		Data34	第 15 个通道模拟值低 8 位
39		Data35	第 16 个通道模拟值高 8 位
40		Data36	第 16 个通道模拟值低 8 位
41		0x0B、0x0F、 0x13、0x15、 0x1E、0x26	帧尾（3 路为 0x0B，5 路为 0x0F，7 路为 0x13，8 路为 0x15，12 路为 0x1E，16 路为 0x26）

0x57: 命令起始信号，“W”的 ASCII 值。

0x75: 每帧数据的帧头，主机通过识别帧头判断数据是否是灰度的数据。

Data0 ~ 36: 8 位的数据，3 路缺省 Data1、Data11 ~ 36 数据，5 路缺省 Data1、Data15 ~ 36 数据，7 路缺省 Data1、Data19 ~ 36 数据，8 路缺省 Data1、Data21 ~ 36 数据，12 路缺省 Data29 ~ 36 数据。

0x0B、0x0F、0x13、0x15、0x1E、0x26: 每帧数据的帧尾，帧尾代表传感器发送几个字节的数据（不包含帧尾），主机可以通过识别帧尾来判断一帧数据传输是否正确。也可以用识别帧尾和字节数判断数据传输是否完成。

3 路一个通信周期有 16 个字节，5 路一个通信周期有 20 个字节，7 路一个通信周期有 22 个字节，8 路一个通信周期有 24 个字节，12 路一个通信周期有 33 个字节，16 路一个通信周期有 41 个字节，每个字节 8 位，这些数据依次按位传送。

表五：设置状态数据意义对照表

主机发送（TXD）					主机接收（RXD）	注释
Byte1: 功能码	Byte2: ID	Byte3: 寄存器	Byte4: 数据	Byte5: 数据	Byte1	

		地址				
0x4C	ID (或 0x00)	0x01	0x00		0x00	数据输出类型为数字量
			0x01		0x01	数据输出类型为偏移量
			0x02		0x02	数据输出类型为模拟量
			0x03		0x03	数字量、偏移量、模拟量全部输出
		0x02	0x00		0x00	深颜色的场地的为低电平
			0x01		0x01	深颜色的场地的为高电平
		0x03	0~0x27	0~0xFF	data	设置中线参考值
		0x04	0~0x7F		0~0x7F	设置 ID, 默认 0x01, 返回 0xFF
						设置错误
			0xFF		ID 值	查看 ID
		0x05	0x00		0x00	波特率: 115200, 重启生效
			0x01		0x01	波特率: 57600, 重启生效
			0x02		0x02	波特率: 38400, 重启生效
			0x03		0x03	波特率: 28800, 重启生效
			0x04		0x04	波特率: 19200, 重启生效
			0x05		0x05	波特率: 14400, 重启生效
			0x06		0x06	波特率: 9600, 重启生效
		0x06	0x00		0x00	采集背景色
			0x01		0x01	采集线色
			0x02		0x02	采集居中值
			0x03		0x03	自动采集背景和线
		0x07	0x00		0x00	亮灯模式
			0x01		0x01	灭灯模式
			0x02		0x02	自适应模式
		0x0B	0x01		0x01	模块复位
		0x0C	0x01		版本号	查询版本 (在文本模式下显示)

0x4C: 功能码, “L” 的 ASCII 值。

“ ”底纹为模块默认设置。

设置状态时建议把模块和串口模块连接, 用上位机软件发送和接收数据。上位机设置比较方便, 资料里附带的有推荐的上位机。

示例: 设置传感器输出数据类型为偏移量, 应该在串口上位机以 HEX 模式 (16 进制模式) 发送 4C 00 01 01, 上位机 HEX 模式接收区收到返回值 01, 如图 5.2-1:

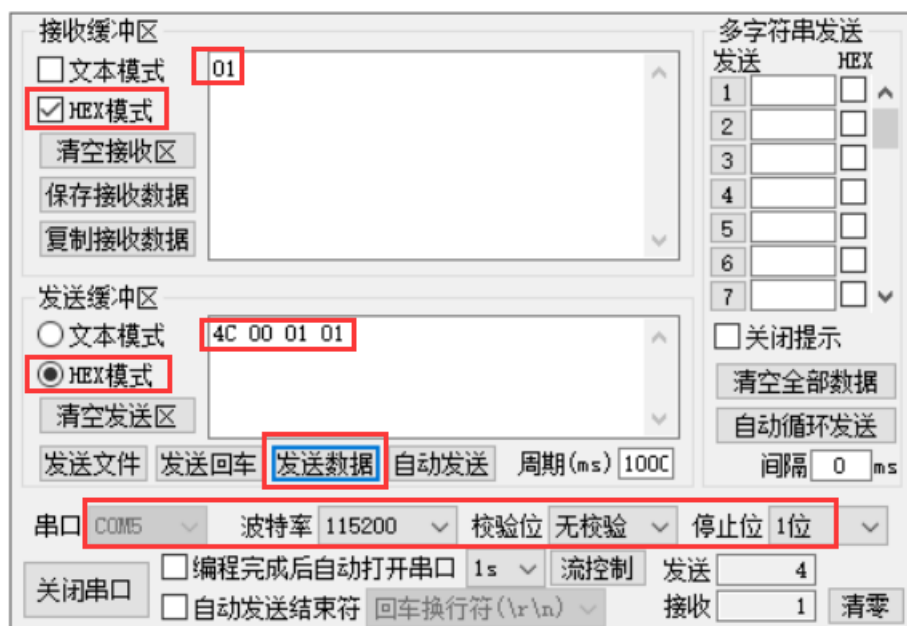


图 5.1-1

注释：偏移值的定义方法。没有具体取值范围，理论寻迹线越宽，取值范围越大。

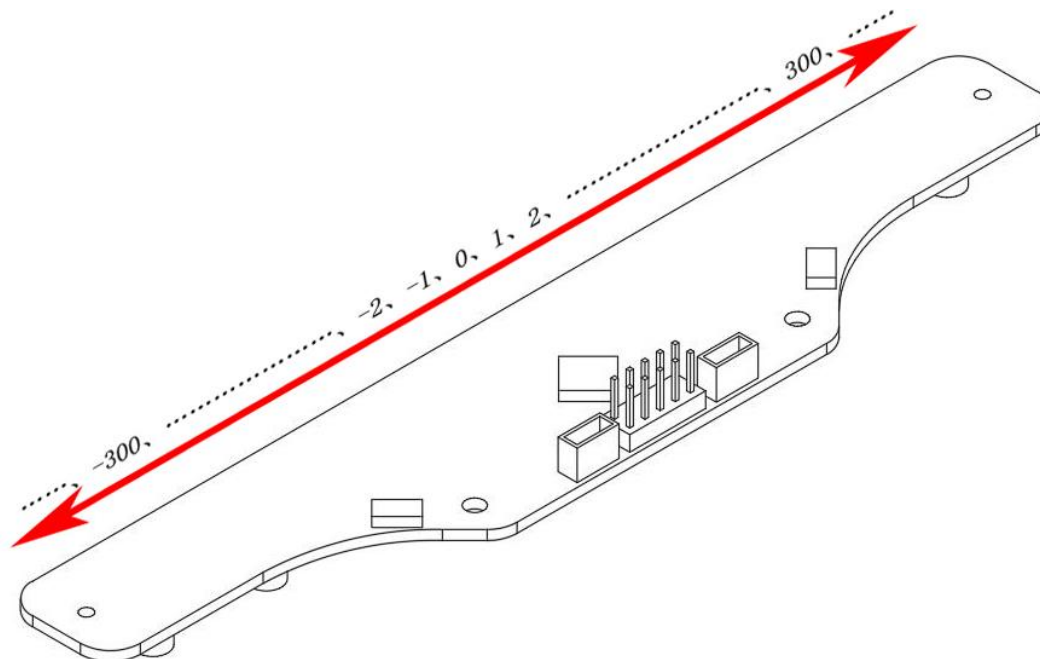


图 5.1-2 偏移量数值分布图

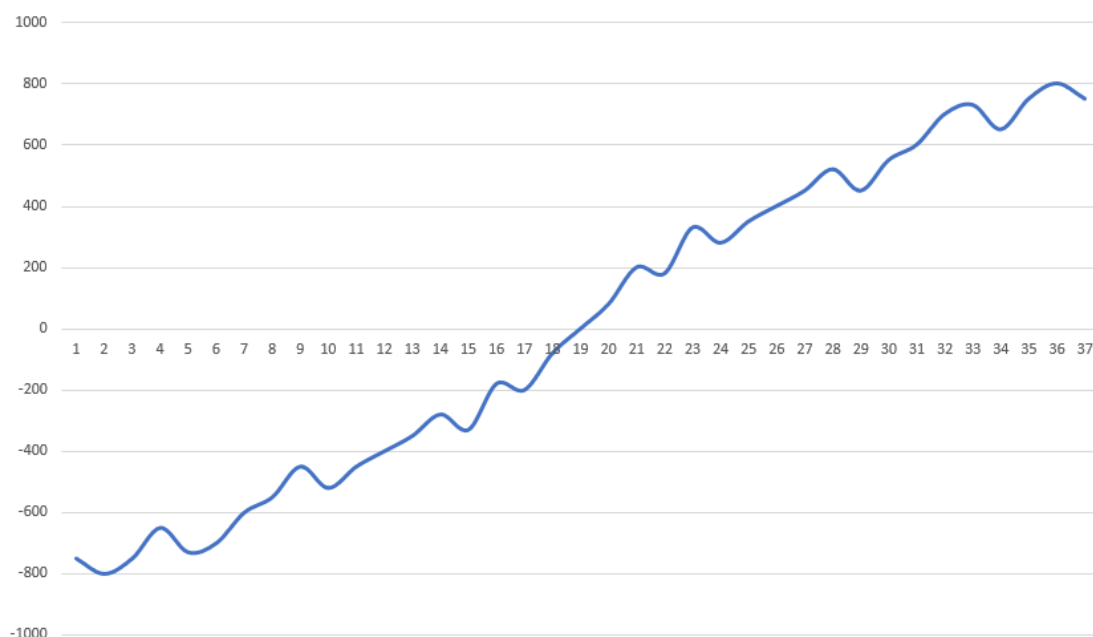


图 5.1-3 数据变化图（示例数据）

5.2 数字口并行输出

数字口可以直接连接在我们单片机的 IO 口上，设置 IO 口的模式为输入模式（和按键的配置一样），读取每路传感器的输出。数字口有高电平和低电平两种电平信号。如果跳线电阻跳 3.3V，那么输出的高电平就是 3.3V 左右的电压。如果跳线电阻跳 5V，那么输出的高电平就是 5V 左右的电压。低电平都是 0V。无论跳到那个位置，跳线电阻一定要焊接。

如果使用的单片机既不是 3.3V 单片机又不是 5V 单片机，可以通过中间继电器作为转接，或者 [TTL 转 RS232 模块](#) 进行电平转换。例如 PLC。

5.3 IIC 通信

当 IIC 接口作为通信接口使用时，主机可以通过发送地址获取传感器的灰度数据。此时模块相当于 24C02，EEPROM 存储器，通信非常简单。主机可以改变传感器的 ID 编号，实现一个主机和多个从机通信的目的（总线上挂载多个从机设备）。

IIC 协议：IIC 时钟 250K 以下，模块 IIC 地址为 7bit。1bit 的读/写指令标志位，写操作为 0，读操作为 1。IIC 主设备代码中 Slave address 是由 7 位的地址位和 1 位的读写指令标志位组成，7bit 的地址为放在高位，1bit 的读写指令标志位放在最低位。模块默认地址为 0x01，0x01 的二进制编码是 0000 0001，如果是写 0x01 地址命令，那么主设备中从设备地址代码应该把 0000 0001 左移一位，然后把最后一位置 0，变成 0000 0010，即 0x02；如果是读 0x01 地址命令，那么主设备中从设备地址代码应该把 0000 0001 左移一位，然后把最后一位置 1，变成 0000 0011，即 0x03。

Slave address 数据含义：

Bit	7	6	5	4	3	2	1	0
Name	address							R/W

模块寄存器地址及含义：

模块内部寄存器地址	地址的数据含义
0x10	地址数据。1 个字节：7 位地址位，1 位读写标志位。取值 0x00~0x7F
0XA0	数字量数据；3、5、7、8 路为一个字节，12、16 路为 2 个字节
0XA1	偏移量数据。3 个字节：第 1 个字节为位置数据（1 位空闲+1 位出线检测+1 位偏移量正负+5 位压线数量），第 2 字节为偏移量高 8 位，第 3 字节为偏移量低 8 位
0XA2	模拟量数据。3 路为 6 个字节，5 路为 10 个字节，7 路为 14 个字节，8 路为 16 个字节，12 路为 24 个字节，16 路为 32 个字节；每两个相邻的字节为一个 16 位的模拟值数据，高 8 位在前，低 8 位在后。

5.3.1 设置模块地址/ID

写操作，主设备发送起始命令和从设备地址信息（R/W 位置零）给从器件，在从设备产生应答信号后，主设备发送模块的字节地址，主设备在收到从设备的另一个应答信号后，再发送数据到被寻址的存储单元模块再次应答，并在主设备产生停止信号后开始内部数据的擦写，在内部擦写过程中，模块不再应答主设备的任何请求。

写操作时序：

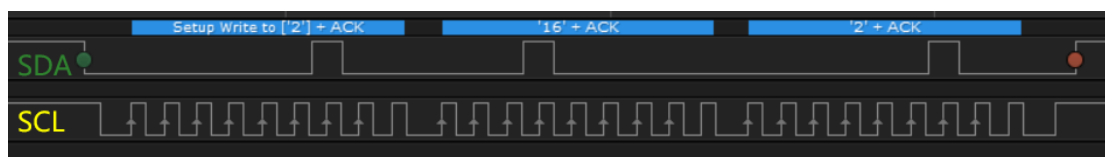


图 5.3.1 设置 ID 时序

修改模块地址 0x01 为 0x02 时序表

Start	Slave address	ack	byte address	ack	data	ack	Stop
起始	0x01<<1	模块	0x10	模块	模块新地址	模块	结束

注：新地址为 7bit。该设置自动保存新地址，上电重启生效。

5.3.2 获取灰度数据

读操作：读操作允许主设备对寄存器的任意字节进行读操作，主设备首先通过发送起始信号、从设备地址和它想读取的字节数据的地址执行一个伪写操作。在模块应答之后，主设备重新发送起始信号和从设备地址，此时 R/W 位置 1，模块响应并发送应答信号，然后输出所要求的字节数据，主器件不发送应答信号但产生一个停止信号。

获取数字量的数据时，传感器数据输出类型应该设置为数字量输出或者数据全部输出。数据输出类型默认是数字量输出。需要通过串口设置，具体设置方法参照[表五](#)。

以 8 路数字灰度读取数字量为例，读操作时序：



图 5.3.2-1 读取数字量时序

获取灰度数字量数据时序表

Start	Slave address	ack	byte address	ack	Stop	Start	Slave address	ack	data	nack	Stop
起始	0x01<<1	模块	0xA0	模块	结束	起始	(0x01<<1)+1	模块	灰度数据	主机	结束

获取偏移量的数据时，传感器数据输出类型应该设置为偏移量输出或者数据全部输出。数据输出类型默认是数字量输出。需要通过串口设置，具体设置方法参照[表五](#)。

以 8 路数字灰度读取偏移量为例，读操作时序：



图 5.3.2-2 读取偏移量时序

获取灰度偏移量数据时序表

Start	Slave address	ack	byte address	ack	Stop	Start	Slave address	ack	data	nack	Stop
起始	0x01<<1	模块	0XA1	模块	结束	起始	(0x01<<1)+1	模块	偏移数据	主机	结束

获取模拟量的数据时，传感器数据输出类型应该设置为模拟量输出或者数据全部输出。数据输出类型默认是数字量输出。需要通过串口设置，具体设置方法参照[表五](#)。

以 8 路数字灰度读取模拟值为例，读操作时序：



图 5.3.2-3 读取模拟量时序

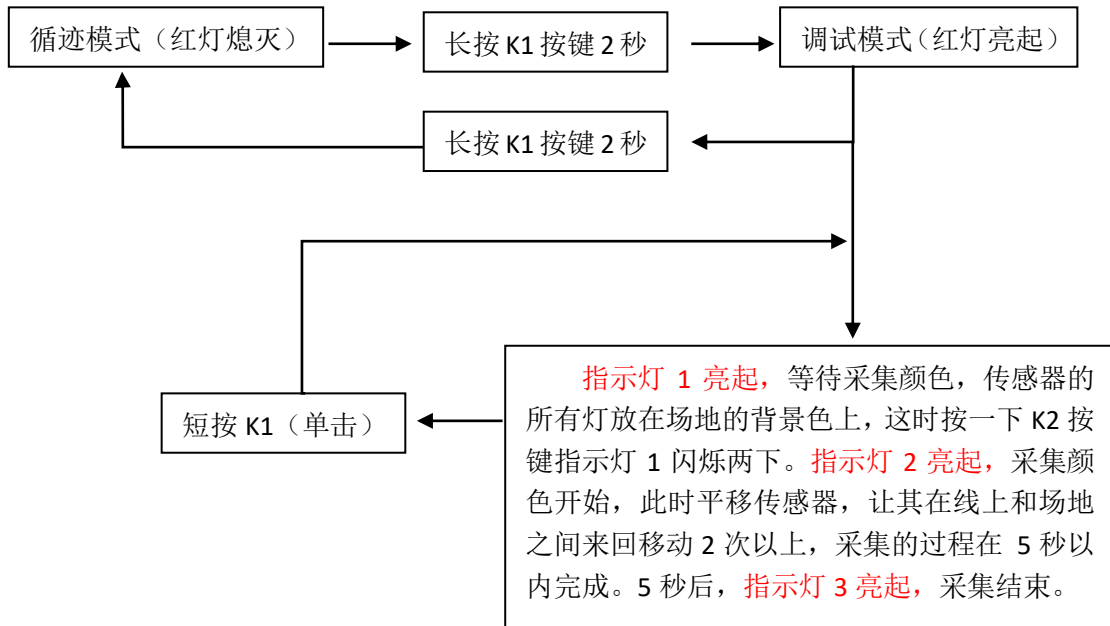
获取灰度模拟量数据时序表

Start	Slave address	ack	byte address	ack	Stop	Start	Slave address	ack	data	nack	Stop
起始	0x01<<1	模块	0XA2	模块	结束	起始	(0x01<<1)+1	模块	模拟数据	主机	结束

六 传感器调试

6.1 按键调试

操作流程图：



6.2 串口命令调试

详细操作请看串口设置参数视频。

6.3 灵敏度调试

智能版灰度传感器采用的是自动调节功能，无需电位器。只需采集颜色，传感器自己调节灵敏度。

灵敏度调节有不同的说法，还有一种说法是高度的调节，其实调节的是同一个位置。根据安装经验，传感器安装的高度要符合以下理论：识别反光强的场地，传感器安装高度尽量高点；识别反光弱的场地，传感器安装高度应尽量低点。传感器灵敏度调节要符合以下理论：传感器在区分两种及以上的不同的颜色时，应选择反光最强和反光次强，或者选择反光最差和反光次差的两种颜色最为参考颜色。

调试的时候我们只需要接正负极两根线即可，信号线在调试时不建议接，这是为了防止IO口配置错误导致调试不准确。然后我们把传感器安装在我们需要的高度（高度范围是10至50mm）。然后按调试视频进行采集数据即可。

七 传感器循迹技巧

7.1 循线解决方案

7.1.1 走直线和圆形线路



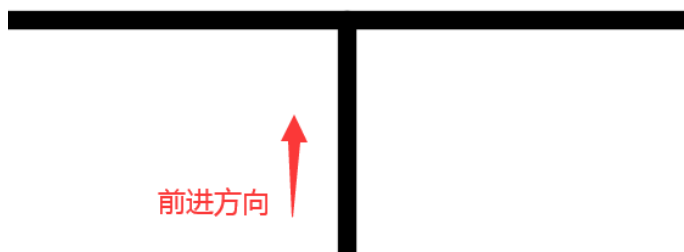
这两种情况比较简单，直接参考我们的走直线的程序即可。

7.1.2 走弧形弯道



这种路线如果走的比较慢可以用走直线的方法寻迹,如果是跑的比较快的话就需要用到[激光传感器](#),进行路口提前检测,然后降速通过弯道。

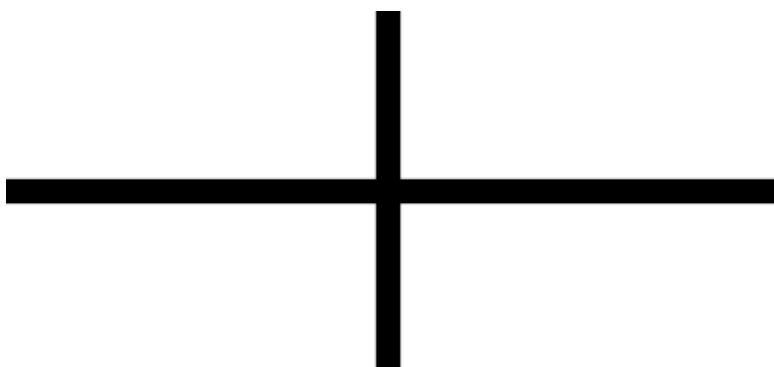
7.1.3 判断“T”字形路口



这里以八路灰度传感器为例进行讲解，白底黑线，线宽 24mm。图中箭头是我们小车行驶的方向，如果我们的车没有到达 T 形路口时是不是只有一部分灰度传感器在黑线上呢（只要我们小车跑的时候不脱离跑道），当我们的灰度传感器所有的灯或则多于 4 个灯在黑线上时，是不是就可以认为我们的车到达了前面的黑线处了。那么我们知道了这个，程序就不难写了，我们循直线的程序里就可以加一个判断 T 形路口的子程序。子程序可以这样写：当灰度灯有 4 个或 4 个以上在黑线上时就认为到达路口（这里灯的数量根据实际的线宽而定），然后再执行停止的命令。

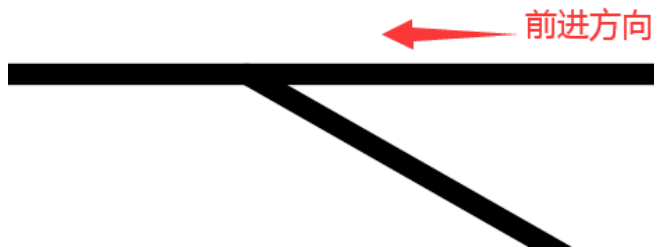
如果小车从右边向左或者从左边向右跑怎么办？方法是一样的。也可以单独用 1 路传感器或者激光专门检测路口，但是需要注意的是，函数一定要加消抖函数。关于消抖的知识有专门的文档讲解。

7.1.4 判断“十”字形路口



方法和判断 T 形路口一样。

7.1.5 判断不规则路口



这样的一般都是用最左边或者最右边的那两个灰度传感器检测路口。也可以单独用 1 路传感器或者激光专门检测路口。

7.2 转向解决方案

路口转弯一般有三种形式：改变前轮的方向、原地旋转、万向轮。

改变前路运动方向。向我们的汽车一样，可以通过方向盘操控运动的方向。用在我们的智能车上，就可以用舵机改变方向。可以配合两个集成灰度传感器实现精确转向。优点转弯平稳不会出现车体震动。缺点是不能用于十字路口和急转弯。

原地旋转。以四轮驱动智能车，右转弯为例。左边两个轮向前运动，右边两个轮向后运动，实现原地旋转。可以配合两个集成灰度传感器实现精确转向。优点转弯速度快，适用于十字路口。缺点车体震动较大，轮胎易磨损。

万向轮。一般有[麦克纳姆轮](#)和福来轮。可以直接通过程序控制电机的运动方向实现智能车的运动方向。优点无需转向就可以实现自由的运动。缺点需要多个（2 个以上）集成的灰度传感器实现循线。

八 版本历史记录

版本	日期	改变
V1.0	2014-10-10	第一代灰度传感器诞生
V2.0	2016-12-16	元件变更，优化电路
V3.0	2018-01-05	尺寸变更，优化布局
V4.0	2019-01-16	升级为智能版，增加串口和显示屏
V5.0	2023-06-06	程序升级，增加 IIC 输出，波特率可修改，巡线模式增加