LineFollowerArray5s

使用说明

介绍

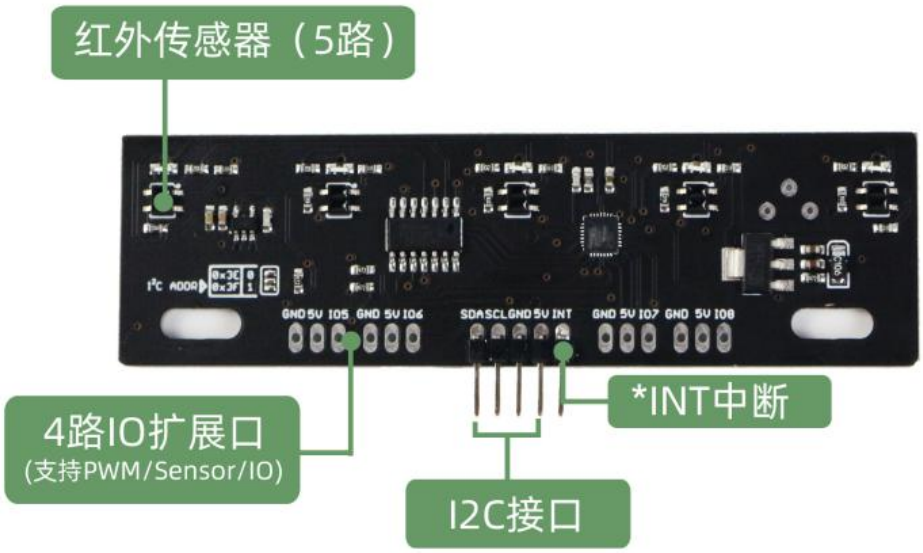
该巡线模块由 I2C 接口通讯，不占用开发板额外端口。模块包含 5 路巡线传感器。预留 4 路 IO 接口，支持 INPUT/OUTPUT，支持 PWM 输出（0-255），可用于扩展 IO 端口使用。

注意：

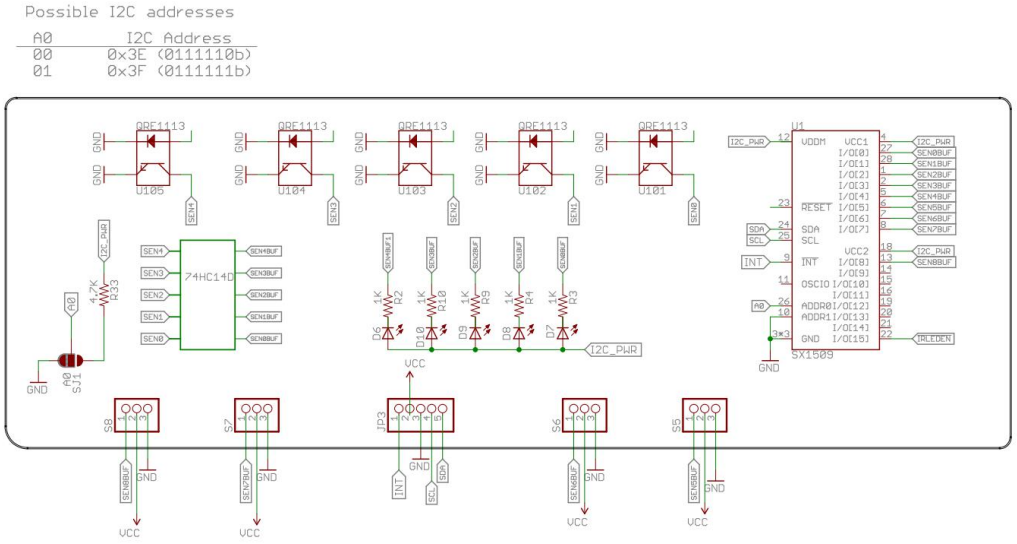
1、太阳会发出大量的红外光，因此在直射阳光下的效率要差得多，户外使用要注意；

2、Arduino I2C 总线会占用系统中断，使用不能在计时器内部使用该模块的方法。

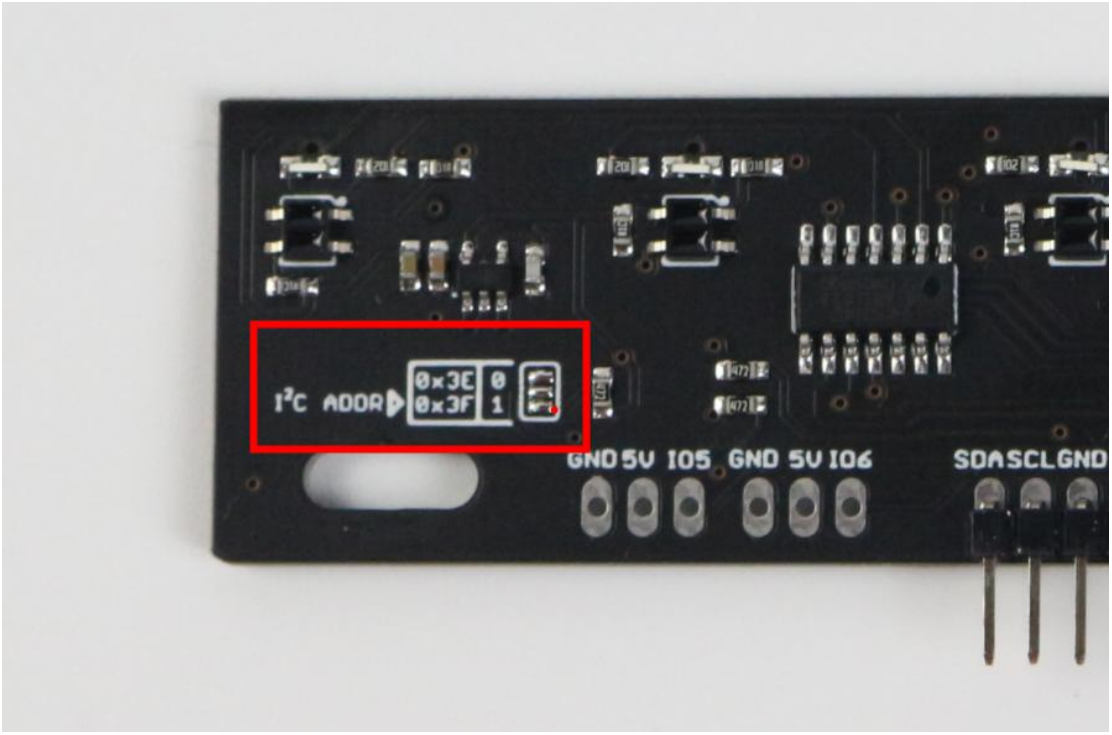
硬件总览



原理图（请查看高清版）



设置 I2C 地址跳线



模块默认 I2C 地址是 0x3E，如果需要更改 I2C 的地址，断开 0 焊盘，焊接 1 焊盘。

安装

组装很容易，与开发板引脚连接如下。

信号/说明	巡线模块	Arduino uno
电源-5v DC	5V	5V
GND	GND	GND
I2C Data	SDA	SDA / A4
I2C Clock	SCL	SCL / A5
INT（*）	NC	NC

*注意：INT 引脚不是必需的，但如果用到中断功能，则可以连接到开发板任何输入端。

使用场地需是白底黑线或者黑底白线，红外传感器有效探测高度是 5mm，所以要根据实际环境调节高度和灵敏度。

QGPMaker_IICSensorbar 库入门(Arduino 库)

示例讲解：

```
#include <Wire.h> // 导入 I2C library
#include "QGPMaker_IICSensorbar.h"

SensorBar mySensorBar; //

const byte IR_S0 = 0; // S0 红外传感器引脚

void setup()
{
  Serial.begin(9600);
  // mySensorBar.begin(<address>) 如果初始化成功，返回 1
  if (!mySensorBar.begin())
  {
    Serial.println("Failed to communicate.");
    while (1) ;
  }
  // 使用 mySensorBar.pinMode(<pin>, <mode>)
  //设置 pin 引脚模式：INPUT or INPUT_PULLUP.默认为 INPUT
  //  mySensorBar.pinMode(1, INPUT);
}
```

```
void loop()
{
    Serial.print("S0 status: ");
  // 读取 S0 状态：  0 or 1
    Serial.println(mySensorBar.digitalRead(S0));
    delay(10);
}
```

首先，导入“ QGPMaker_IICSensorbar.h”库（以及“ Wire.h”库），并在全局区域中创建一个对象：

```
#include <Wire.h> // Include the I2C library (required)
#include “QGPMaker_IICSensorbar.h” // Include SX1509 library
SensorBar mySensorBar; // Create an SensorBar object
```

从现在开始，我们将使用 mySensorBar 对象。然后初始化改对象，并确保其正确通信。请调用 mySensorBar.begin(<address>)，其中<address>是扩展器的 I 2 C 地址（默认为 0x3E）。检查返回值，如果返回 1，则保通讯正常。

```
if (!mySensorBar.begin(0x3E))
{
  // If we failed to communicate, turn the pin 13 LED on
  digitalWrite(13, HIGH);
  while (1); // And loop forever.
}
```

初始化成功后，我们就可以像使用 Arduino 的 IO 方法一样使用。只需要前面加上 mySensorBar 对象。

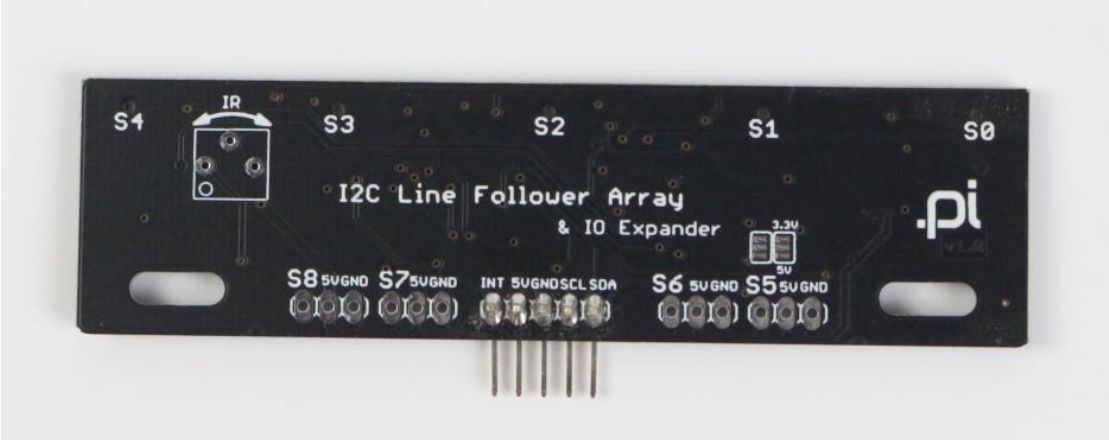
比如：

mySensorBar.pinMode(<pin>, <mode>);//mode:INPUT,OUTPUT,or INPUT_PULLUP

mySensorBar.digitalWrite(<pin>,<value>);//pin:[5-8],value:HIGH or LOW.

mySensorBar.digitalRead(<pin>);//范围[0-8],其中[5-8]是扩展接口

读取红外传感器状态




```
//获取第 i 个,i 范围[0-8],其中[5-8]是扩展接口
int irSensor=mySensorBar.digitalRead(i);
```

模拟输出（PWM）

还可以通过使用 `analogWrite(<pin>, <0-255>)`功能将 S[5-8]端口用作“模拟”（PWM）输出-就像 Arduino 模拟输出一样！需要注意的区别：

- **ANALOG_OUTPUT**：如果您希望引脚产生 PWM 信号，请调用 `pinMode(<pin>, ANALOG_OUTPUT)`设置。
- PWM 模式用于舵机驱动（不支持舵机驱动）

以下是示例代码，用于调节 S5 引脚外接的 LED 灯亮度：

```
#include <Wire.h> // Include the I2C library (required)
#include <SparkFunSX1509.h> // Include SX1509 library
SensorBar io; // Create an SX1509 object
// SX1509 pin definitions:
// 注意: 这不是 Arduino 主板的 pins.是巡线扩展板的 S5 端口
const int LED_PIN_S5 = 5; // LED 连接到 S5 端口
void setup()
{
  io.begin(0x3E); // 初始化
  // 设置 S5 的模式为 ANALOG_OUTPUT, 使其开启模拟输出（PWM）功能.
  io.pinMode(SX1509_LED_PIN, ANALOG_OUTPUT);
}
void loop()
```

```
{
  for (int i=0; i<256; i++)
  {
    // PWM the LED from 0 to 255
    io.analogWrite(SX1509_LED_PIN, i);
    delay(2); // Delay 2ms between each
  }
  delay(500); // Delay half-second at the top.
  for (int i=255; i>=0; i--)
  {
    // PWM the LED from 255 to 0
    io.analogWrite(SX1509_LED_PIN, i);
    delay(2); // Delay 2ms between each
  }
  delay(500); // Delay half-second at the bottom.
}
```

这样就完成了一个呼吸灯的效果。