

1. Description



The line tracking sensor is actually an infrared sensor. There is a 3-channel line tracking in front of the driver base board of the car, and the component used here is TCRT5000 IR tube. It works by using the different reflectance of IR light on colors and then converting the intensity of the reflected signal into an electric current signal.

Black is active at high voltage, while white is active at low voltage when detecting. The detection height is 0-3 cm. There are three blue rotary potentiometers to adjust the detection sensitivity of the sensor.

2. Component Knowledge

Working voltage: 3.3-5V (DC)

Interface: 5PIN interface (connect to A0,A1,A2)

Output signal: Digital signals(The analog port can also be used as a digital signal, A0 corresponds to D13 and A1 corresponds to D14)

Height detection: 0-3 cm

Principle: It works by using the different reflectance of IR ray on colors and then converting the intensity of the reflected signal into an electric current signal. When powering on, the diode emits IR light. FB- is the potentiometer, adjusting it can get a threshold voltage for the 4,

keyestudio

6 and 8 pins of the voltage comparator LM339D. The voltage value can be determined according to the actual situation.

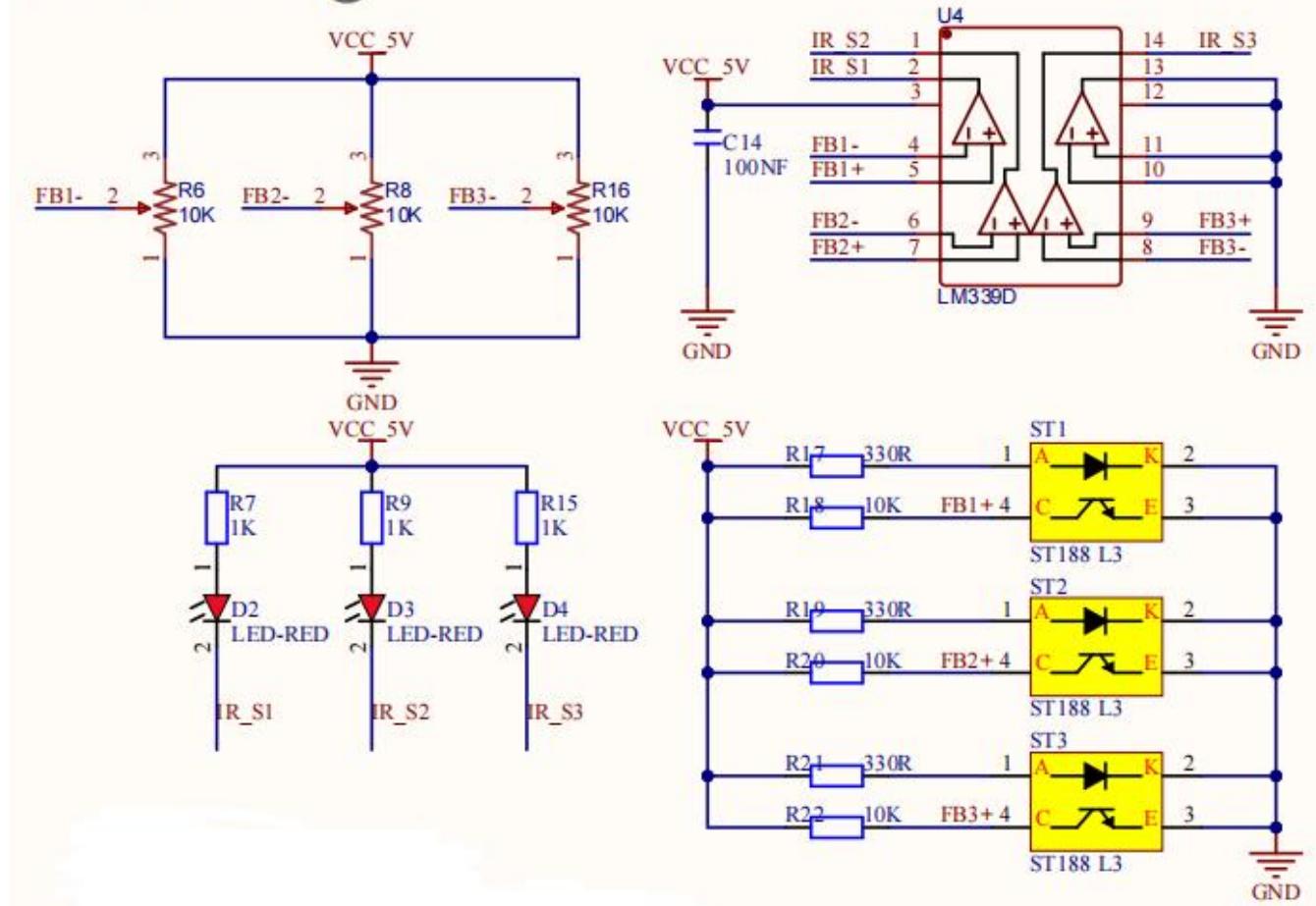
When infrared diode receives the infrared light, they will generate a current, and as the infrared light changes from weak to strong, the current increases from small to large. However, when there is no infrared light or it is very weak, the output signal is close to VCC.

After passing the LM339D comparator, the receiving detection pin outputs a high level. When the light intensity and the current increase, the output voltage will become smaller, when it is small to a certain extent, the pin will become low level.

Furthermore, if black or no object is detected, the signal end of the MCU will receive a high level. While a white object is detected, a low level will be received.

keyestudio

IR Tracing



keyestudio

3. Test Code

```
//*****
/*
Keyestudio 4WD Mecanum Robot for Arduino
lesson 5
Line Tracking Sensor
http://www.keyestudio.com
*/
*****Define the pin of Line Tracking Sensor*****
#define SensorLeft    A0 //input pin of left sensor
#define SensorMiddle  A1 //input pin of middle sensor
#define SensorRight   A2 //input pin of right sensor

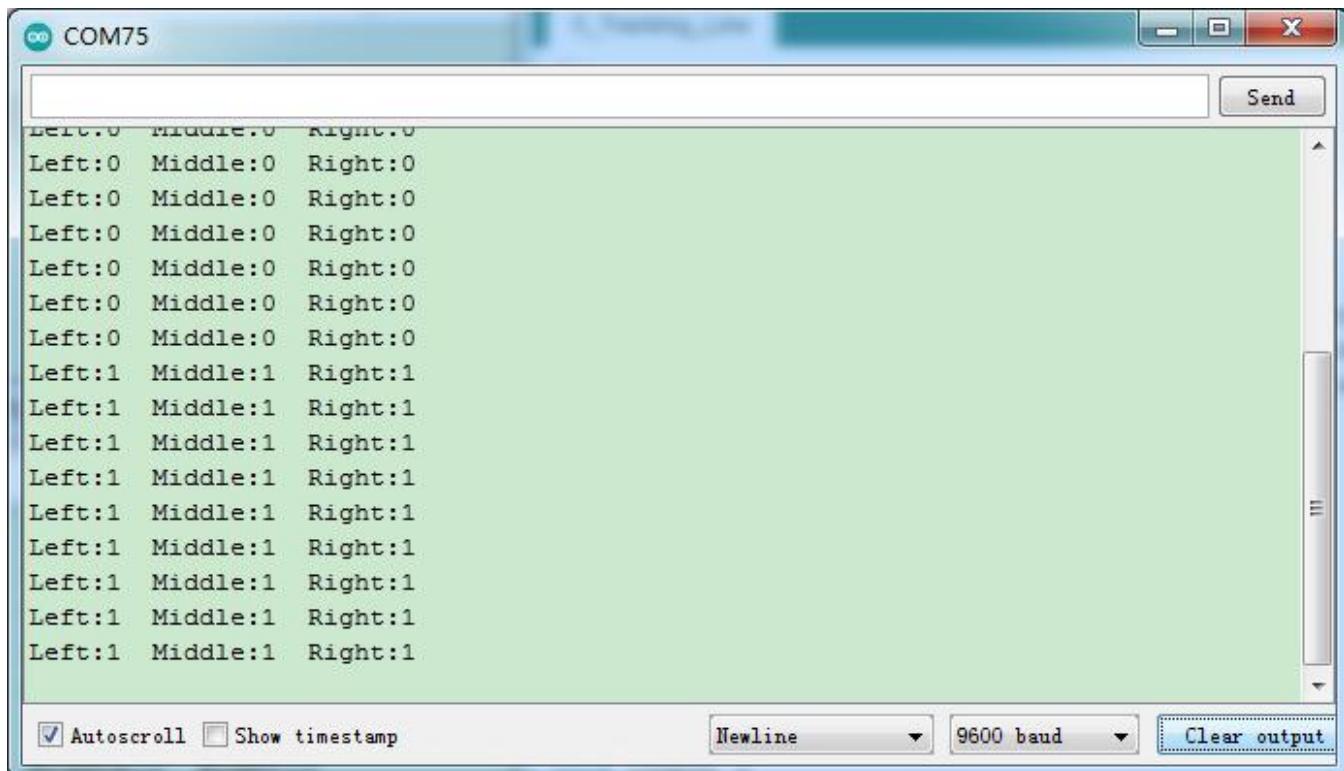
void setup() {
  Serial.begin(9600); //Start the serial monitor and set the baud rate to 9600
  /**All Line Tracking Sensor interface are set to input mode**/
  pinMode(SensorLeft, INPUT);
  pinMode(SensorMiddle, INPUT);
  pinMode(SensorRight, INPUT);
}

void loop() {
  uint8_t SL = digitalRead(SensorLeft); //Read the value of the left line tracking sensor
  uint8_t SM = digitalRead(SensorMiddle); //Read the value of the intermediate line tracking sensor
  uint8_t SR = digitalRead(SensorRight); //Read the value of the right line tracking sensor
  Serial.print("Left:");
  //Serial port prints the information
  Serial.print(SL);
  Serial.print(" Middle:");
  Serial.print(SM);
  Serial.print(" Right:");
  Serial.print(SR);
  Serial.println(); //Line feed
  delay(100);
}
//*****
```

keyestudio

4. Test Result

After compiling and uploading the code, turn the DIP switch to ON end and open the serial monitor. We can see the digital signal received by the three-channel line tracking sensor printed on the monitor. When we cover it with white paper, the output is 0, when using black paper or hanging the car, the output is 1.



```
Left:0  Middle:0  Right:0
Left:1  Middle:1  Right:1
```

The screenshot shows a Windows-style serial monitor window titled "COM75". The main text area displays a series of lines of text, each consisting of three values: "Left:0", "Middle:0", and "Right:0". This pattern repeats several times, followed by a sequence where "Left" and "Middle" values change to "1" while "Right" remains at "0". At the bottom of the window, there are several control buttons: "Autoscroll" (checked), "Show timestamp" (unchecked), "Newline" (dropdown menu), "9600 baud" (dropdown menu), and "Clear output" (button).

keyestudio

5. Code Explanation

#define SensorLeft A0	Set the left pin of the sensor to A0
Serial.begin(9600);	Start the serial monitor and set the baud rate to 9600
pinMode(SensorLeft, INPUT);	Set pin to input mode
digitalRead(SensorLeft);	Read the digital level of the pin, when the pin is 3.3V, the value we read is 1, when the pin is 0V, the value we read is 0, after reading we use a variable SL to save them
Serial.print("Left:");	Serial port monitor non-newline prints "Left:
Serial.println();	Serial port monitor newline printing