

Track Print

Track Print

- Device connection
 - Hardware connection
 - Software connection
- Three-way patrol module
 - Control principle
 - Control pin
 - Code analysis
- Experimental results

Use the serial port to print the data of the three-way patrol module and the direction that the simulated car needs to change.

Device connection

Hardware connection

Use the Type-B data cable to connect the Arduino Uno and the computer.

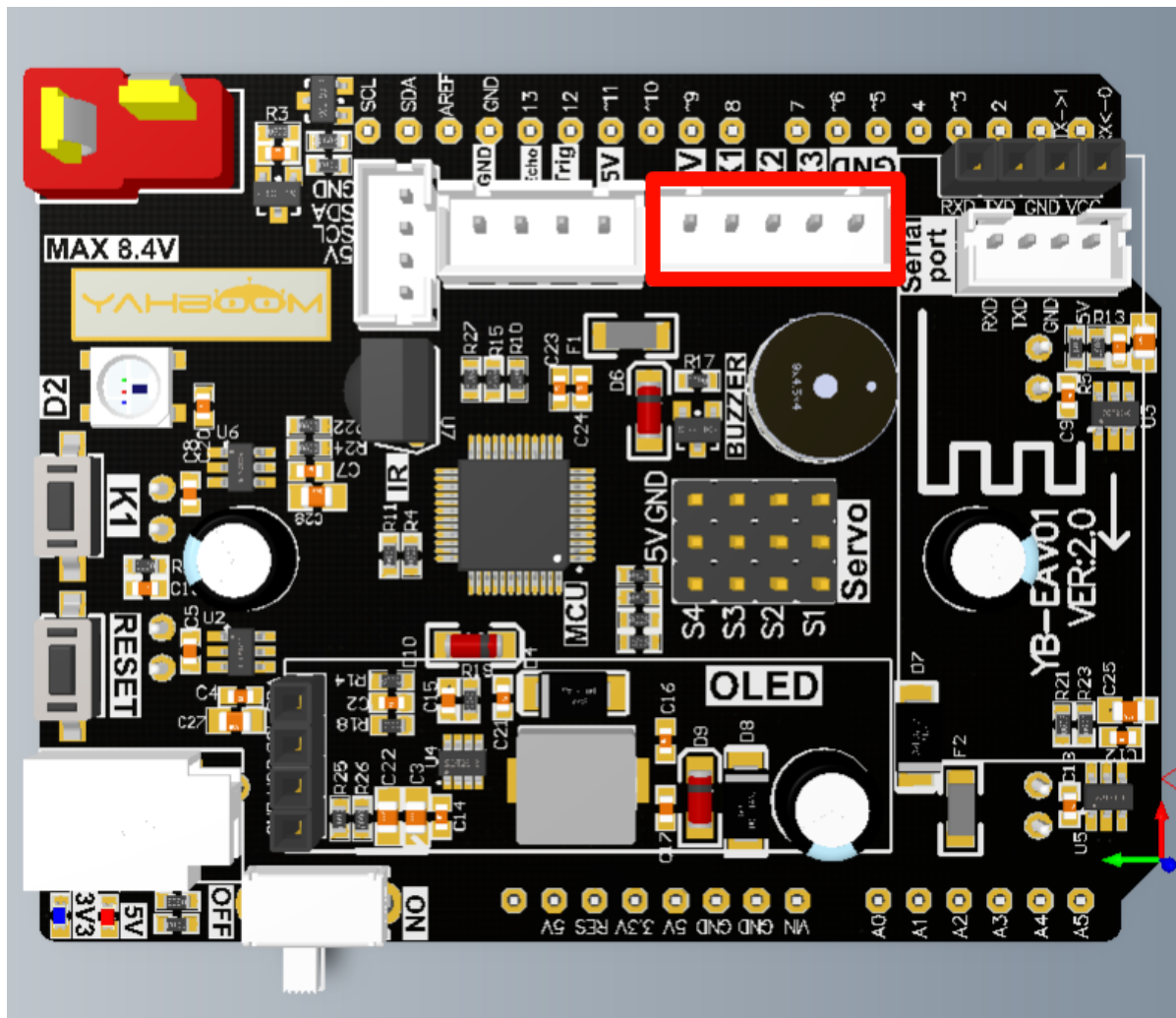
Software connection

Open the "Arduino IDE" software and select the model and serial port number corresponding to the development board.

Three-way patrol module

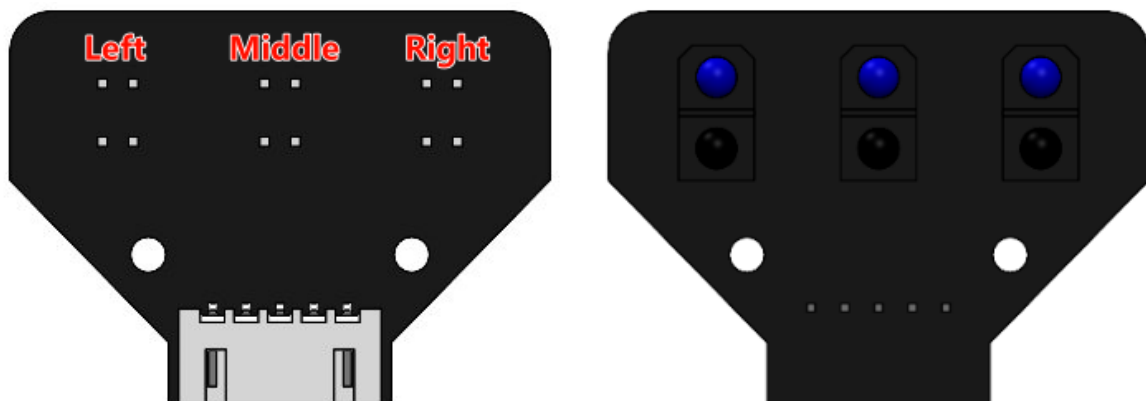
The basic principle of infrared sensor patrol is to use the reflective properties of objects: when infrared rays are emitted to the black line, they will be absorbed by the black line, and when they are emitted to materials of other colors, they will be reflected to the infrared receiver.

The position selected by the red box is the location of the three-way patrol module interface:



Control principle

By reading the analog value data of the three-way line patrol module X1, X2, and X3 interfaces, the location of the black line is determined.



when the three-way tracking module is installed, the infrared tracking sensor faces the ground, so the left, middle, and right sensors in the program are based on the left mark in the above figure

Control pin

The three-way line patrol module and the Robduino expansion board have a special interface for installation and connection.

Peripheral module	Arduino Uno
VCC	VCC
X1	X1 (A0)
X2	X2 (A1)
X3	X3 (A2)
GND	GND

Note: The infrared light emitted by the infrared probe is invisible to the human eye; in order to avoid infrared light interfering with the sensor, we need to use this module indoors.

Code analysis

Here we only briefly introduce the code content. For detailed code, please refer to the corresponding code file, which is provided in the download area!

- Define the three-way line patrol module pins and black line threshold

```
// 定义三路循迹模块引脚 Define the three-way line patrol module pins
#define L_TRACK_PIN A2
#define M_TRACK_PIN A1
#define R_TRACK_PIN A0

// 定义巡线阈值 Define the patrol threshold
const int Threshold = 500;
```

- Get line patrol data and status

```
/**
 * @brief 获取三路巡线模块数据和状态 Get the status of the three-way line patrol
module
 * @param iLeftPin: 左边红外传感器 Left tracking sensor
 * @param iMidPin: 中间红外传感器 Middle tracking sensor
 * @param iRightPin: 右边红外传感器 Right tracking sensor
 * @retval 无 None
 */
void getTrackLineState(int iLeftPin, int iMidPin, int iRightPin) {
    int leftValue = analogRead(iLeftPin);
    int middleValue = analogRead(iMidPin);
    int rightValue = analogRead(iRightPin);

    Serial.print("Left: ");
    Serial.print(leftValue);
    Serial.print(" Middle: ");
    Serial.print(middleValue);
    Serial.print(" Right: ");
```

```

Serial.print(rightValue);

// 进行简单的方向判断打印 The following is a simple judgment
if (leftValue > Threshold && middleValue < Threshold && rightValue < Threshold)
{
    Serial.println("  -> Turn Left");
} else if (leftValue < Threshold && middleValue < Threshold && rightValue >
Threshold) {
    Serial.println("  -> Turn Right");
} else {
    Serial.println("  -> Go Straight");
}
}

```

- Initialization Code

```

void setup() {
    Serial.begin(115200); // 初始化串口波特率115200 Initialize serial communication at
115200 bps
}

```

- Looping code

```

void loop() {
    getTrackLineState(L_TRACK_PIN, M_TRACK_PIN, R_TRACK_PIN); // 获取三路巡线模块数据和
状态 Get the status of the three-way line patrol module
    delay(100);
}

```

Experimental results

After compiling the program successfully, upload the code to the Arduino Uno development board.

After the program is started, the serial port will print the interface data of the three-way patrol module and the direction that the simulated car needs to change!

If there is no display content, you can check whether the serial port baud rate is consistent with the code setting, and then press the RESET button on the development board.

The burning program cannot use other programs to occupy the serial port or an external serial communication module (for example: WiFi camera module), otherwise the program cannot be burned or an error message will be prompted!

09.Track_Print | Arduino IDE 2.3.2

File Edit Sketch Tools Help

Arduino Uno

09.Track_Print.ino

```
1 // 定义三路循迹模块引脚 Define the three-way line patrol module pins
2 #define L_TRACK_PIN A2
3 #define M_TRACK_PIN A1
4 #define R_TRACK_PIN A0
5
6 // 定义巡线阈值 Define the patrol threshold
7 const int Threshold = 500;
8
9 void getTrackLineState(int iLeftPin, int iMidPin, int iRightPin); // 获取三路巡线模块数据和状态 Get the status of the three-way line patrol module
10
11 void setup() {
12   Serial.begin(115200); // 初始化串口波特率115200 Initialize serial communication at 115200 bps
13 }
14
15 void loop() {
16   getTrackLineState(L_TRACK_PIN, M_TRACK_PIN, R_TRACK_PIN); // 获取三路巡线模块数据和状态 Get the status of the three-way line patrol module
17   delay(100);
18 }
19
```

Output Serial Monitor x

Message (Enter to send message to 'Arduino Uno' on 'COM8')

New Line 115200 baud

14:42:45.073 -> Left: 248 Middle: 225 Right: 263 -> Go Straight
14:42:45.759 -> Left: 262 Middle: 244 Right: 277 -> Go Straight
14:42:45.842 -> Left: 284 Middle: 270 Right: 297 -> Go Straight
14:42:45.979 -> Left: 308 Middle: 298 Right: 321 -> Go Straight
14:42:46.071 -> Left: 340 Middle: 334 Right: 352 -> Go Straight
14:42:46.150 -> Left: 374 Middle: 372 Right: 388 -> Go Straight
14:42:46.289 -> Left: 449 Middle: 460 Right: 463 -> Go Straight
14:42:46.382 -> Left: 514 Middle: 527 Right: 520 -> Go Straight
14:42:46.462 -> Left: 558 Middle: 573 Right: 560 -> Go Straight
14:42:46.550 -> Left: 580 Middle: 591 Right: 578 -> Go Straight

Ln 14, Col 1 Arduino Uno on COM8