海盗船 --- 红外开关避障功能

Pirate: IR Obstacle Avoidance Switch

在完成基础功能后，我们对机器人来个升级，给它增加避障功能。避障功能，是机器人最基础的功能之一，也是机器人体现自主意识的第一步。避障机器人能够识别前方的障碍物，并避开前方障碍物。这里，我们选用到的是红外数字避障传感器，这是一款相当容易上手的传感器，非常适合初学者使用。该传感器可帮助机器人看到前方的物体。

After completing the basic functions, we’re now ready to give our robot an upgrade: obstacle avoidance capabilities. Obstacle avoidance is among a robot’s most fundamental capabilities. It enables a robot to recognize and avoid obstacles that lie in its path. Here, we’ll be using digital infrared obstacle avoidance sensors – they’re relatively easy to use and thus ideal for beginners. Using these sensors will allow your robot to detect and avoid objects that might otherwise block its path.

**硬件材料：**

**Hardware components:**

* 3-80cm红外数字避障传感器 × 3

3-80cm infrared (IR) digital obstacle avoidance sensors x 3



* 红外接近开关支架 × 3

IR switch support frame x 3



**组装步骤**

**Assembly Directions**

找到硬件材料后，就可以开始组装了，组装不难，更着操作就行。

After finding the necessary parts, you’re ready to start assembling. Assembly won’t be too difficult – just follow the directions below and you’ll have no problems.

**STEP 1: 摆放材料**

**Step 1: Sort out and arrange materials**

先找到固定红外开关传感器的3个支架，及需要使用的M3×6MM的螺丝和螺母若干。

First, find the 3 support frames for the IR sensors. You’ll need several M3x6mm screws and nuts to fasten these frames.



**STEP 2: 固定红外支架**

**Step 2: Attach the IR support frames**

这里只需将3个红外开关传感器的支架用螺丝拧在上层板上即可。需要注意的一点是，三个支架之间不要间隔太近。否则在检测的时候会造成接收错误信号。

Screw on the three support frames to the Pirate’s top plate. Make sure not to attach the support frames too close together – doing so will cause the sensors to mix up signals, which leads to errors.



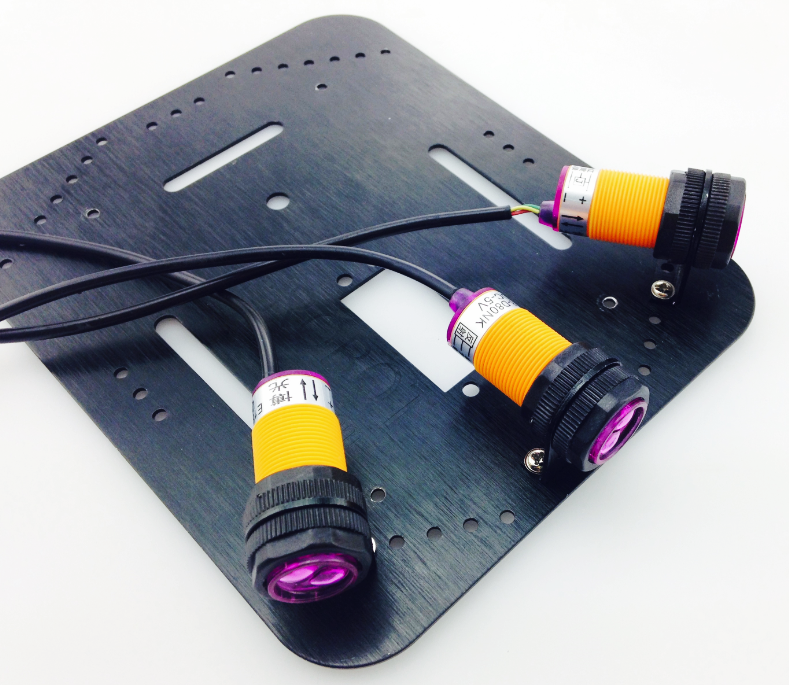
**STEP 3: 安装红外传感器**

**Step 3: Assemble the IR sensors**

先将一个固定环装到传感器上，将传感器从红外支架里往外穿过，并用另一个固定环将其固定住。同样方法安装另外两个。

Place the sensor through the fixed ring on the support frame. The sensor has two rings that can be adjusted to fasten the sensors into place. Use both rings to fasten the sensors into the support ring. The sensors should be facing outwards.

Repeat this process for all three sensors.



**硬件连接**

**Connect the hardware**

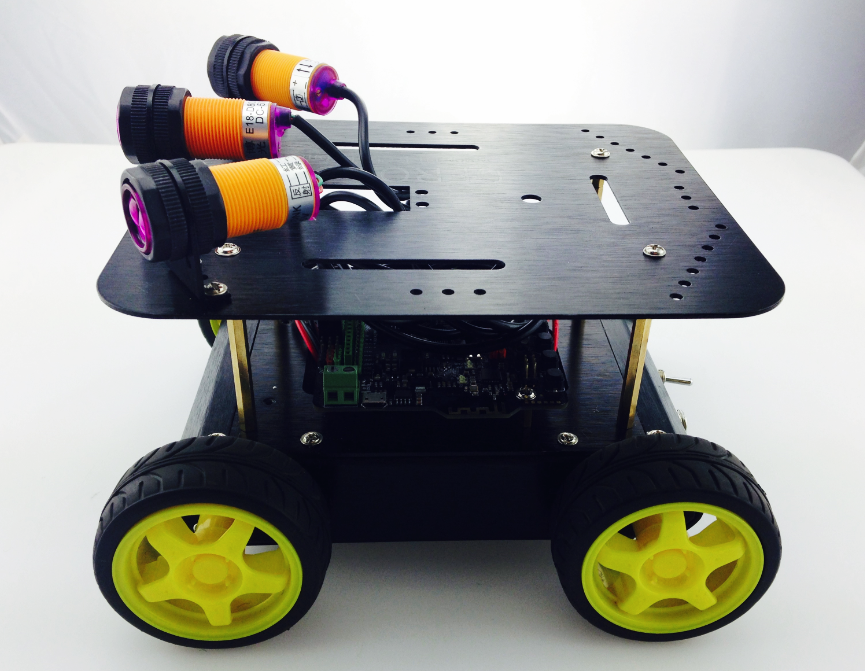
传感器安装完成后，先不要急着把上层板接上去，需要将传感器与Romeo BLE控制器连接。左边这张图显示的是三个传感器ABC在小车上对应的位置，分别对应数字口的10，9，8。，连接时只需留意线序是否正确即可。 安装完成后，固定上层板。

After assembling the sensors, don’t rush to put the Pirate’s top plate back on – before doing that, we need to first connect the sensors with the Romeo BLE control plate. The picture on the left-hand side shows the correct ABC placement of the sensors on the sensor board, which corresponds to pins 10, 9, and 8 on the BLE board. When connecting the sensors, be sure to check that you’ve connected them in the proper sequence. After connecting the sensors, re-attach the Pirate’s top plate back on the body of the car.

**黄色 - 信号线，红色 – VCC, 绿色 – GND。**

**Yellow: Signal line, Red: VCC (positive), Green: GND (negative)**





**传感器调试**

**Adjust the sensors**

红外接近开关是一种集发射与接收于一体的光电开关传感器。如果有信号，也就是前方有障碍物，传感器后侧指示灯亮的亮灭，检测距离可随传感器后侧的电位器进行调节，可调范围3-80cm。

IR approach switches are a type of IR sensor that collects sent and received signals. If a signal is detected – i.e. an obstacle is detected in front of the Pirate – the LED on the back of the sensor will turn on. The screw head can be tightened to adjust the sensor’s distance detection to a range of 3 to 80 cm – that is, the sensor can detect objects as near as 3 cm or as far as 80 cm away.

下载代码前，需要对三个传感器的做调整，先插上USB线，给板子供电。下图白色圈出的位置可见有个一字螺丝，这个就是可用来调整检测距离。拿个小盒子(模拟障碍物)，放在传感器探头前，找到传感器的探测点，一旦找到传感器上的指示灯会亮起。此时，可以拧转螺丝，调到一个合适的距离，推荐在15~20cm左右。

Before downloading the code needed to configure the sensors, we need to first adjust their distance detection. First, take your USB cable and plug it into your microcontroller to power it on. Take a look at the below picture: the small, circled area is a screw head that can be turned to adjust the sensor’s distance detection. Take a **small box** (so as to imitate an obstacle) and place it in front of the head of the sensor. Once the sensor detects the box’s presence, the red LED light on the back of the sensor will light up. We want to calibrate the sensor to detect obstacles 15-20 cm away – to do this, place the box 15-20 cm away from the head of the sensor and tighten the screw head until the sensor’s LED lights up (which signifies that the object has been detected).

同样方法调整另外两个红外传感器，通常建议左右两侧的检测距离略大于中间的检测距离。

Adjust the left and right sensors in the same manner. We recommend that the left and right sensors be adjusted to the same distance (15-20 cm) as the middle sensor.

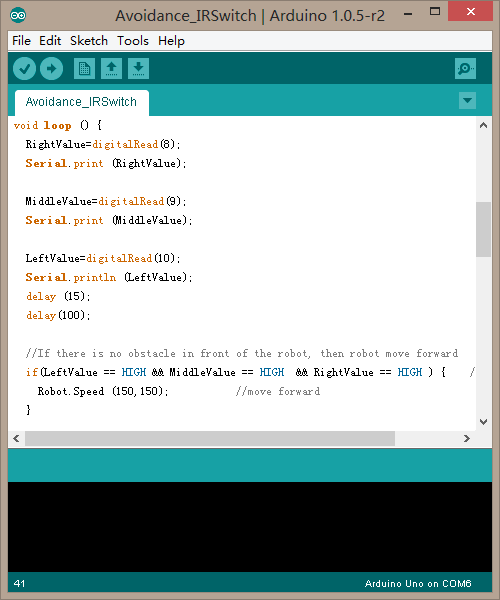
c

**输入代码**

**Coding**

由于代码较长，所以这里就不全部显示出来，你可在软件包中找到Avoidance\_IRSwitch.ino的代码，下载进去后即可。

Since the length of the code is relatively long, we won’t delve into all the specifics. . Download the Arduino code, named “Avoidance\_IRSwitch.ino”, from GitHub.

c

下载完成后，你就可以看到小车的避障效果了。当然，如果你觉得检测距离还不是很满意，也可以做一些微调。直到能正常工作。玩过之后，我们就来简单看下避障原理以及学习下代码，便于你能更好的改造你的机器人！

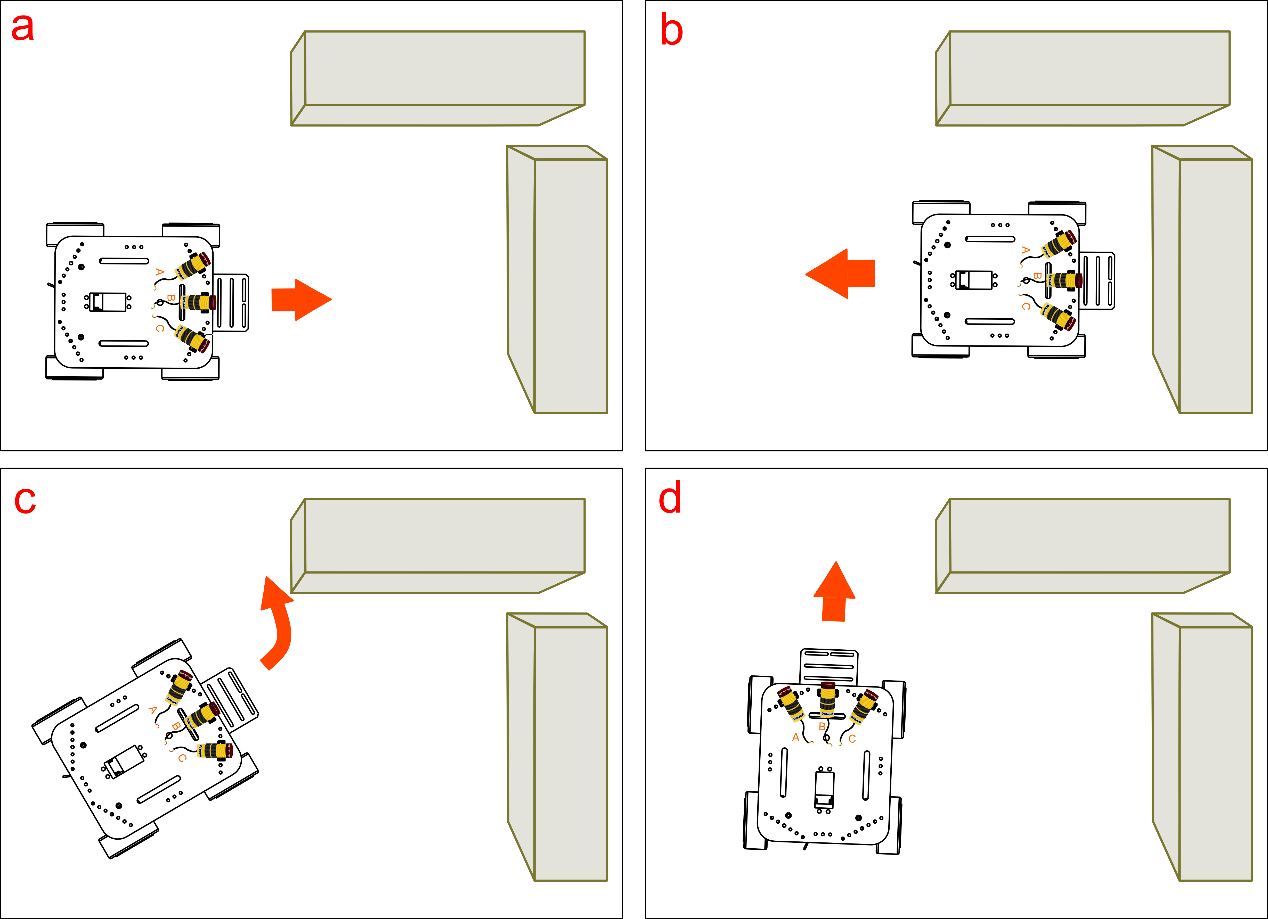
After downloading the code, upload it to your Romeo microcontroller. Once your car starts moving, you’ll be able to see the obstacle avoidance sensors in play – as soon as they detect an object in their path, they’ll light up and re-direct the car. If you feel like the sensor’s distance detection isn’t sufficiently strong, feel free to further adjust the sensors.

**避障原理**

**Obstacle avoidance: how it works**

通过下面a，b，c，d来简单看下如何通过三个传感器来实现避障。在前方没有障碍物的时候，小车前行(图a)，一旦检测到前方有障碍物，先让小车后退(图b)，然后执行左转(图c)，当然这里你也可以右转。如果小车前方一直存在障碍物，小车将会一直执行后退左转，直到前方没有障碍物，小车保持前进。

The following four diagrams (A, B, C, D) briefly illustrate how the three sensors work to avoid obstacles. At first, the car moves forward uninhibited (diagram A). Once it detects an obstacle, the sensors command the car to reverse (diagram B). The car will then attempt to move either left or right (diagram C). In the event that the car is surrounded by obstacles on all sides, the sensors will command the car to reverse and turn until it finds an open space without obstacles, whereupon the car will continue to move forward (diagram D).

****

前面分析的这个场景是小车正前方有障碍物。思考下，如果小车左前方有障碍，或者小车右前方有障碍物，可让小车如何执行？清楚原理之后，我们来看下代码。

The above diagrams depict the car facing an obstacle in front of its path. Consider: if the car has obstacles to both its left and front side, or both its right and front side, how would the car adjust its movement?

当然，这样的避障方式不是唯一的，你也可以有更好的避障方式，可由你自己设定，如果你的逻辑非常紧密，完全可以写出更出色的代码，毫无疑问的可以从你的代码中表现出来。何不试试用自己的思维来写一段控制代码。看完代码回顾，你就能游刃有余的改代码了。

Of course, the Avoidance\_IRSwitch.ino code provided isn’t the only way of programming your sensors. Once you feel comfortable with the principles and basic code behind the sensors, feel free to write your own code and configure the sensors to your liking.

**代码回顾**

**Code Synopsis**

基础功能重复部分就不说了，只说下避障部分的。

There’s no need to discuss the basic code – let’s just take a look at the part involving obstacle avoidance.

int RightValue; //Right line tractor sensor on Pin 8

int MiddleValue; //Middle line tractor sensor on Pin 9

int LeftValue; //Left line tractor sensor on Pin 10

//reading 3 pins values of Line Tracking Sensor

RightValue=digitalRead(8);

MiddleValue=digitalRead(9);

LeftValue=digitalRead(10);

用三个变量RightValue，MiddleValue，LeftValue分别记录3个传感器读到的值，digitalRead(pin)函数就是用来读取数字口的值，不明白的可以查看下[语法手册](http://wiki.dfrobot.com.cn/index.php/Arduino%E7%BC%96%E7%A8%8B%E5%8F%82%E8%80%83%E6%89%8B%E5%86%8C%EF%BC%88%E5%A4%9A%E9%A1%B5%E9%9D%A2%E7%89%88%EF%BC%89)。

Use three variables – RightValue, MiddleValue, LeftValue – to record the 3 sensors’ read values.

The digitalRead(pin) function is used to read the digital Input/Output port value. If these terms are still confusing, please check out our Terminology Manual or the Arduino website.

**红外数字避障传感器数字量传感器，当探头前方有障碍时输出低电平，无障碍物时输出高电平。**

**When the IR obstacle avoidance sensors detect an object in their path, they’ll produce a LOW energy output. When there are no objects detected in their path, they’ll produce a HIGH energy output.**

下面这段代码描述的就是避障原理中所说的那张情况 –- 正前方有障碍。三个传感器同时都检测到障碍物了，也就是输出值都为LOW时，小车先执行后退，再执行左转，前进。

The snippet of code below (example A) illustrates how the obstacle avoidance code works. If/when the three sensors detect an object in front of its path, they will produce a LOW energy output; they will then prompt the car to reverse, turn left, then attempt to move forward once again.

if(LeftValue==LOW && MiddleValue==LOW && RightValue==LOW ) {

Robot.Speed (-200,-200); //back off

delay(800);

Robot.Speed (-200,200); //turn left

delay(400);

Robot.Speed (100,100); //move forward

}

如果只有左侧传感器检测到障碍物，那么小车执行，后退，左转。

If the left sensor detects an obstacle in its path, it will prompt the car to reverse, then turn **left**. See (B) below:

if(LeftValue == LOW) { //obstacle on the left side

Robot.Speed (-200,-200); //back off

delay(400);

Robot.Speed (200,-200); //turn right

delay(250);

Robot.Speed (100,100); //move forward

}

同样，如果只有右侧传感器检测到障碍物，那么小车执行，后退，右转。

And if the right sensor detects an obstacle in its path, it will prompt the car to reverse, then turn **right**. See (C) below:

if(RightValue == LOW ){ //obstacle on the right side

Robot.Speed (-200,-200); //back off

delay(400);

Robot.Speed (-200,200); //turn left

delay(250);

Robot.Speed (100,100); //move forward

}

最后，如果前面没有障碍物的话，小车正常前进。

Lastly, if sensors do not detect any object in their path, then the car will move forward as normal. See (D) below:

if(LeftValue==HIGH && MiddleValue==HIGH && RightValue==HIGH ) {

Robot.Speed (150,150); //move forward

}

代码中还有这么一段。

The code also contains the below snippet:

if( MiddleValue == LOW){ //obstacle in middle

Robot.Speed (-150,-150); //back off

delay(400);

Robot.Speed (-200,200); //turn left

delay(300);

Robot.Speed (100,100); //move forward

}

同样是前方有障碍物，和我们一开始说的三个传感器同时检测到的区别在于，这种情况下，前方障碍物相对远一点，小车后退的速度慢一点。

This is another example of the sensors detecting an obstacle in its path. In this instance, the obstacle detected is relatively further away from the sensors, which leads to the car to reverse at a slightly slower speed.

Robot.Speed (-150,-150); //back off

这里，我们就用到通过Speed函数来调整速度了。根据代码的说明，能否自己尝试变换一下小车的速度，行动方向等等。

Here, we use Arduino’s Speed function to adjust the speed at which the car moves after receiving input from the sensors. Feel free to adjust the speed values (which range from -255 to 255 ) to your liking. Our robot can now detect and maneuver around obstacles. Interestingly, we can apply these same sensors to give our robot “tracking” capabilities. Unlike obstacle avoidance, which seeks to avoid detected objects, tracking capabilities command your car to follow an object once detected. Obstacle avoidance, tracking – these are powerful features that allow you to get creative. Time to start tinkering!

**大改造**

**Considerations**

代码就讲到这里，我们这里做的是一个避障机器人。其实，同样的传感器，改造一下就能成为跟踪机器人了。只是方式相反，这里是，检测到物体后避开，跟踪的话，则是检测到物体后，向物体方向靠近。想一下，给自己做个跟踪机器人。