

# Ruff教程 - 防盗报警器

---

本教程需要的模块是

- 人体红外传感器 <https://rap.ruff.io/raps/hc-sr501> (Ruff开发板配件中没有, 可以淘宝购买, 大约5元一个)
- 继电器 <https://rap.ruff.io/raps/relay-1c>
- 彩色LED模块 <https://rap.ruff.io/raps/ky-016>
- 蜂鸣器 <https://rap.ruff.io/raps/buzzer-gpio>

会分步完成所有的功能

- 接入人体红外传感器, 并初步检测的代码
- 在传感器检测到结果的时候, 控制继电器接通
- 继电器控制LED闪烁和蜂鸣器鸣响
- 微信推送报警信息

## 准备工作

---

在群里发现有些朋友接入网络的时候经常遇到问题, 这里我建议使用串口的方法配置WiFi, 否则在周围WiFi热点较多的时候, 需要多次重试才能成功连接。详见 [https://ruff.io/zh-cn/docs/network-configuration.html#USB\\_u4E32\\_u53E3\\_u6A21\\_u5F0F。](https://ruff.io/zh-cn/docs/network-configuration.html#USB_u4E32_u53E3_u6A21_u5F0F。)

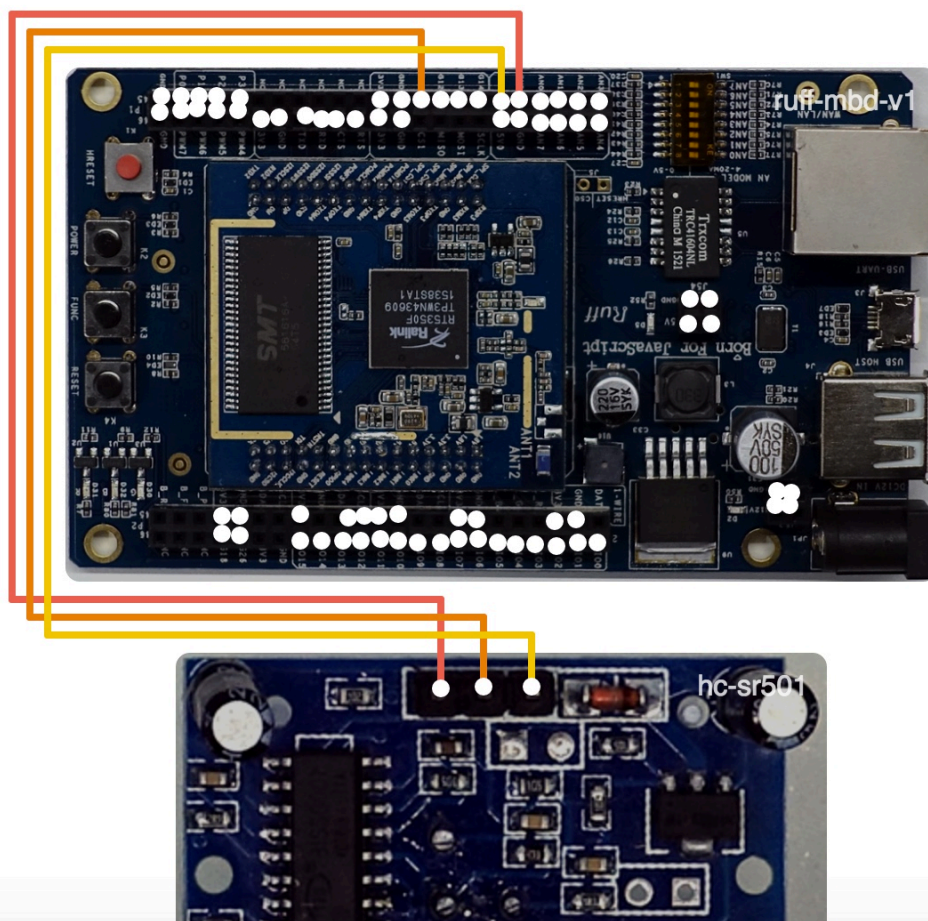
然后配置环境变量, 并新建工程, 我的工程名是 `ruff-alarm`。细节不再一一赘述。

## 人体红外传感器的用法

---

在上文提到的链接中有传感器的文档, 简单看了下就可以明白。

首先运行 `rap device add hc-sr501`, 一路回车添加硬件。然后运行 `rap deploy --visual` 查看接线图, 并对应接线。



编辑 `src/index.js`，修改为

```
'use strict';

$.ready(function (error) {
  if (error) {
    console.log(error);
    return;
  }

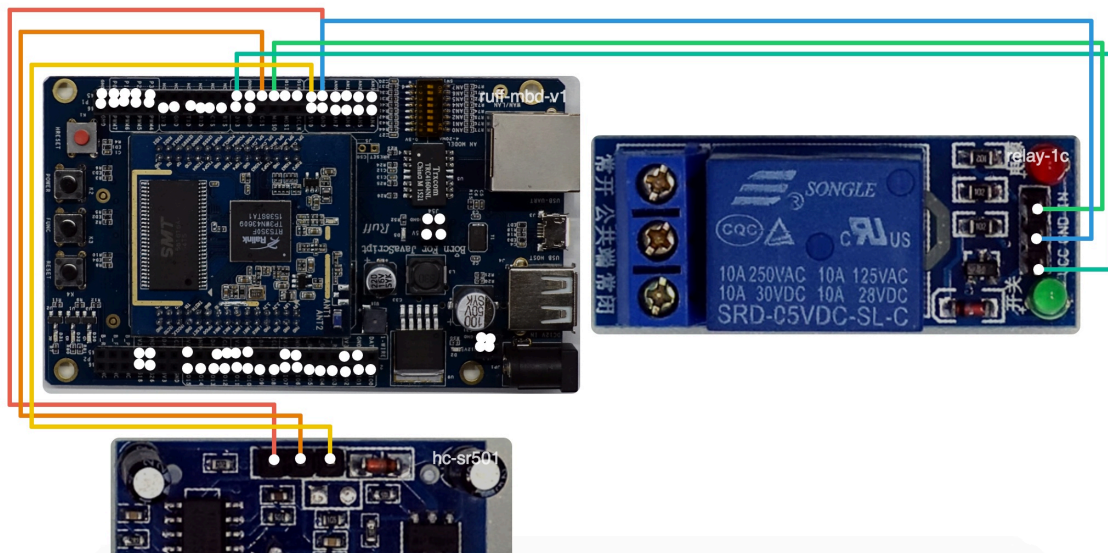
  var sensor = $('#hc-sr501');
  sensor.on('presence', function () {
    console.log('Presence.');
```

运行 `rap deploy -s` 部署运行我们的代码，尝试靠近和远离传感器，在 `rap log` 中就可以看到输出了。（传感器非常灵敏，可能需要离远点）

```
Aug 30 21:52:40: Console server bound.
Aug 30 21:52:41: Absence.
Aug 30 21:52:43: Precense.
Aug 30 21:52:46: Absence.
Aug 30 21:52:53: Precense.
```

## 控制继电器

运行 `rap device add delay-1c` 添加继电器硬件，然后对照布局图对应接线。



然后你就会发现一个问题，传感器最左边一根线和继电器中间一根线在开发板上的接口是一个，仔细看了下，发现这根线是GND，你可以认为是全局的一个回路，而开发板上的GND都是联通的，所以我就又随便找了一个GND接口插上了。

查看继电器文档，发现基本用法还是很简单的，一个 `turnOf`、一个 `trunOff` 方法，但是我们需要一个延时断开，因为人在传感器前面快速通过的时候，可能检测到的时间非常短。这里就需要使用 `setTimeout` 函数了，这个函数接收两个参数，第一个是回调函数，第二个是时间，单位是毫秒。

比如你可以写这样的代码，

```
setTimeout(function () {
  console.log("Hello world");
}, 3000)
```



我们只要修改为继电器接通后，10秒后断开就好了，代码还是很简单，而且这里我们不需要关心传感器“离开”事件回调了。

```
'use strict';

$.ready(function (error) {
  if (error) {
    console.log(error);
    return;
  }

  var sensor = $('#hc-sr501');
  var relay = $('#relay-1c');
  var isRelayOn = false;
  sensor.on('presence', function () {
    console.log('Precense.');
    if (!isRelayOn) {
      isRelayOn = true;
      relay.turnOn(function () {
        console.log('relay turn on');
        setTimeout(function () {
          relay.turnOff(function () {
            console.log('relay turn off');
            isRelayOn = false;
          });
        }, 10000);
      });
    }
  });
});

});
```

为防止在继电器接通的时候再次触发接通时间，我们使用了一个变量保存继电器状态，如果继电器是接通状态，那就忽略当前的事件。

`rep deploy -s` 测试下，是不是你一靠近继电器就会接通？（继电器接通的时候可以听到声音和看到绿灯亮）

## 使用继电器控制负载

使用继电器是为了更加方便控制第三方的设备，比如你的门锁，家里的灯之类的，但是我们这里就只使用Ruff开发板配件里面的蜂鸣器和LED等做示范。如果不需要控制第三方的设备的话，在上一步中直接控制蜂鸣器和LED等就可以。

`rap device add ky-016` 和 `rap device add fc-49`，然后查看连线图。因为有点乱，就不贴了。

但是注意，两个设备都有一个GND线，先不要将这两根线连接到Ruff开发板。我们要使用继电器控制这根线的通断，从而模拟第三方的设备。

接下来我们需要将这两根线的尾端剪短，去掉绝缘皮，两根线捻在一起，接到继电器的常开端口。然后在继电器的公共端口上，再连接一根杜邦线，连接到开发板上任意一个GND就可以。这样继电器接通的时候，蜂鸣器和LED就可以联通了。

参考蜂鸣器和LED的文档，我们可以写出以下的代码，其中 `setInterval` 是js中定时循环的函数，和 `setTimeout` 用法基本一致。

```
'use strict';

$.ready(function (error) {
  if (error) {
    console.log(error);
    return;
  }

  var isRelayOn = false;
  var sensor = $('#hc-sr501');
  var relay = $('#relay-1c');
  var buzzer = $('#fc-49');
  var LED = $('#ky-016');

  // 红蓝闪烁
  var colors = [[0xff, 0x0, 0x0], [0x0, 0x0, 0xff]];
  var index = 0;

  sensor.on('presence', function () {
    console.log('Precense.');
    if (!isRelayOn) {
      isRelayOn = true;
      relay.turnOn(function () {
        console.log('relay turn on');
        buzzer.turnOn(function () {
          console.log('buzzer turned on');
        });

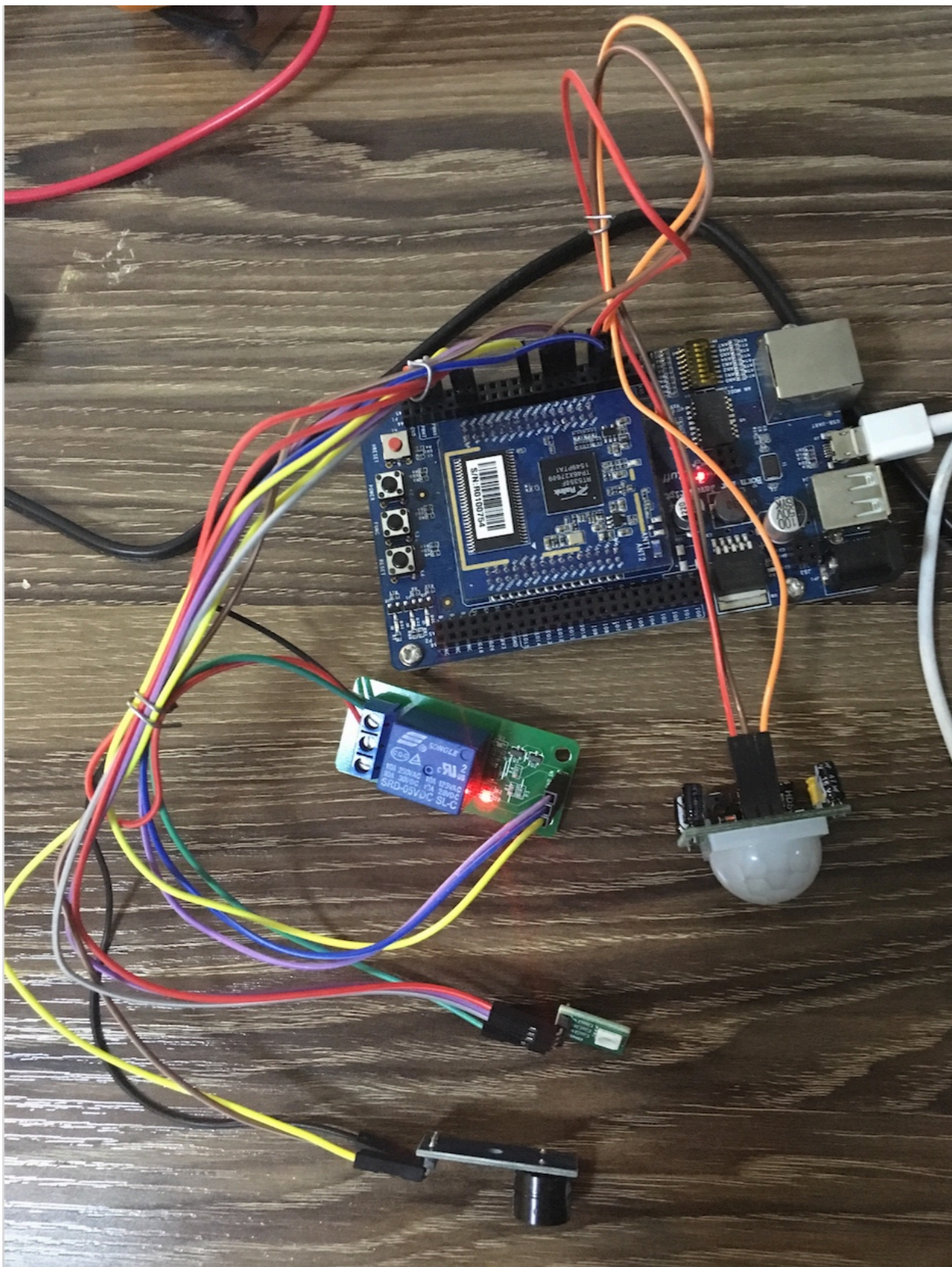
        LED.turnOn(function () {
          console.log('LED turned on');
        });

        setInterval(function () {
          LED.setRGB(colors[index % 2]);
          index = (index + 1) % 2;
        }, 200);
      });
    }
  });
});
```

```
setTimeout(function () {  
  buzzer.turnOff(function () {  
    console.log('buzzer turned off');  
  });  
  LED.turnOff(function () {  
    console.log('LED turned off');  
  });  
  
  relay.turnOff(function (error) {  
    isRelayOn = false;  
    console.log('relay trun off');  
  });  
}, 10000);  
});  
});  
});
```

最终的接线图是这样的





## 使用微信通知

这里我们使用一个叫Server酱的服务，地址是 <http://sc.ftqq.com/3.version>。只要发送一个GET请求，就可以把信息推送到你的微信。具体注册登录方法见网站帮助。



然后在 <http://sc.ftqq.com/?c=code> 可以看到一个SECRET，复制，然后填写到代码中。

根据网站帮助文档，我们只需要发送一个类似

`http://sc.ftqq.com/{SECRET}.send?text={TEXT}` 的GET请求就可以了。

Ruff SDK中自带http相关模块，地址是 <https://ruff.io/zh-cn/api/http.html> 用法和jQuery的ajax很像。

所以最终代码就是

```
'use strict';

$.ready(function (error) {
  if (error) {
    console.log(error);
    return;
  }

  var http = require("http");

  var isRelayOn = false;
  var sensor = $('#hc-sr501');
  var relay = $('#relay-1c');
  var buzzer = $('#fc-49');
  var LED = $('#ky-016');

  var weixinSecret = "YOUR SECRET";

  var colors = [[0xff, 0x0, 0x0], [0x0, 0x0, 0xff]];
  var index = 0;

  sensor.on('presence', function () {
    console.log('Presence. ');
    if (!isRelayOn) {
      isRelayOn = true;
      relay.turnOn(function () {
        console.log('relay turn on');
        buzzer.turnOn(function () {
          console.log('buzzer turned on');
        });
      });

      LED.turnOn(function () {
        console.log('LED turned on');
      });

      http.get('http://sc.ftqq.com/' + weixinSecret + '.send?text=' + encodeURIComponent('检测到家中有人活动，请注意'), function (res) {
        console.log("response: " + res.statusCode);
        res.setEncoding('utf8');
        res.on('data', function (chunk) {
          console.log(chunk);
        });
      });
    }
  });
});
```



```

    });
  }).on('error', function (e) {
    console.log("error: " + e.message);
  });

  setInterval(function () {
    LED.setRGB(colors[index % 2]);
    index++;
  }, 200);

  setTimeout(function () {
    buzzer.turnOff(function () {
      console.log('buzzer turned off');
    });
    LED.turnOff(function () {
      console.log('LED turned off');
    });

    relay.turnOff(function (error) {
      isRelayOn = false;
      console.log('relay trun off');
    });
  }, 10000);
});
});
});

```

## 注意

- 金属不要触碰到开发板表面，比如杜邦线的公头和你的金属工具。
- 蜂鸣器很吵，调试的时候可以先把它的GND拔掉。

## 思考

- 怎么远程控制报警器的开关？要注意的是开发板连接到路由器，没有公网IP。
- 怎么调节人体红外传感器来达到最好的性能，这里有一份资料  
<http://yunpan.taobao.com/s/jpgOuX6v9X>
- 我们忽略了所有的错误处理，怎么保证一个硬件故障的时候，系统还能正常工作。
- 代码层面上，怎么避免嵌套过多的回调函数？

