企业真实案例项目资源清单

序号	名称	链接
1	软件测试综合实训	点击查阅
2	Android 物联网实训应用程序开发实践	点击查阅
3	数据可视化实战	点击查阅
4	网络运行与维护	点击查阅
5	网络管理与维护综合实训	点击查阅

HNBC
广州华南商贸职业学院

《软件测试综合实训》

编者姓名：王芬、李国文

广州华南商贸职业学院云智信息技术学院

广州德爱信息科技有限公司

《软件测试综合实训》

编者姓名：王芬、李国文

广州华南商贸职业学院云智信息技术学院

广州德爱信息科技有限公司

目　录

模块一　前端开发环境

项目一　spirit 响应式官网

实训目标：完成 spirit 响应式官网

实训技能：掌握 bootstrap 框架、spirit 框架、以及开发环境 phpstudy 搭建。

实训内容：搭建开发环境并完成响应式网站创建

任务 1 开发环境准备

1、实训要求：使用 bootstrap 完成页面布局，用 less 预处理语言编写样式，会使用 phpstudy 搭建环境

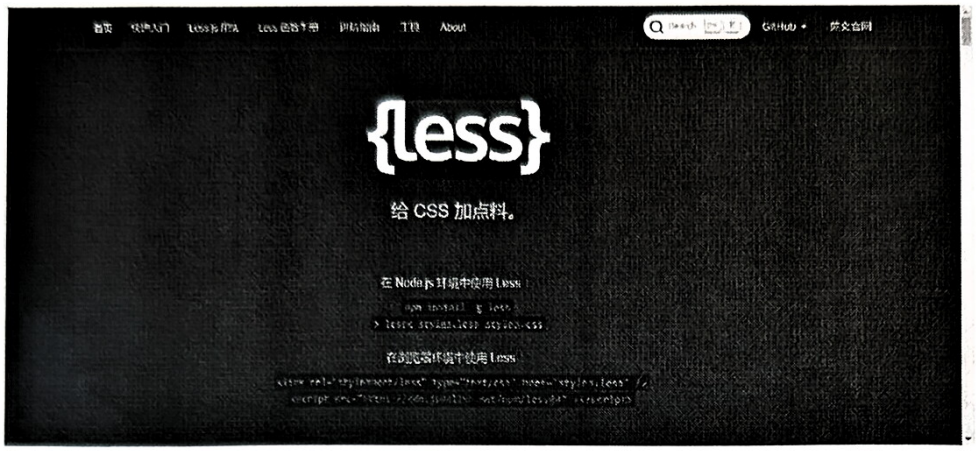
2、操作步骤

（1）下载使用 bootstrap 文件

地址：https://www.bootcss.com/

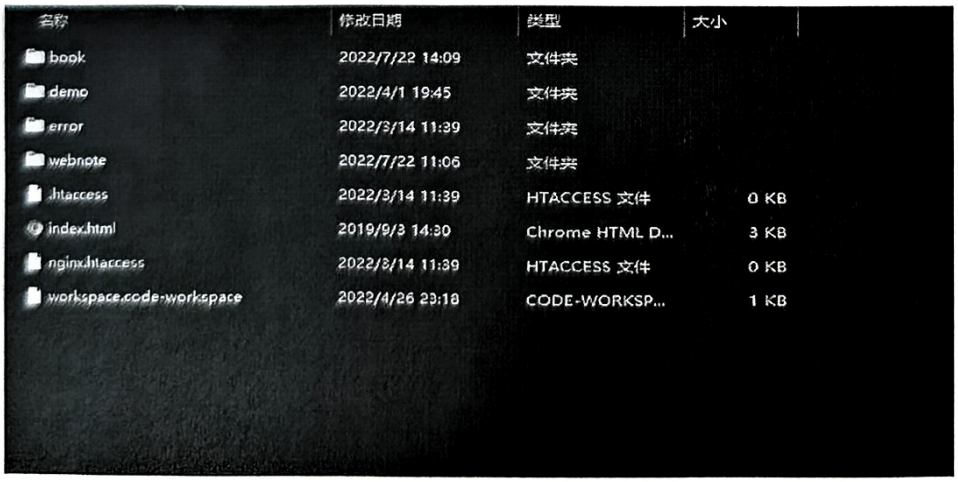
（2）下载 less 文件并引入

地址：https://less.bootcss.com/

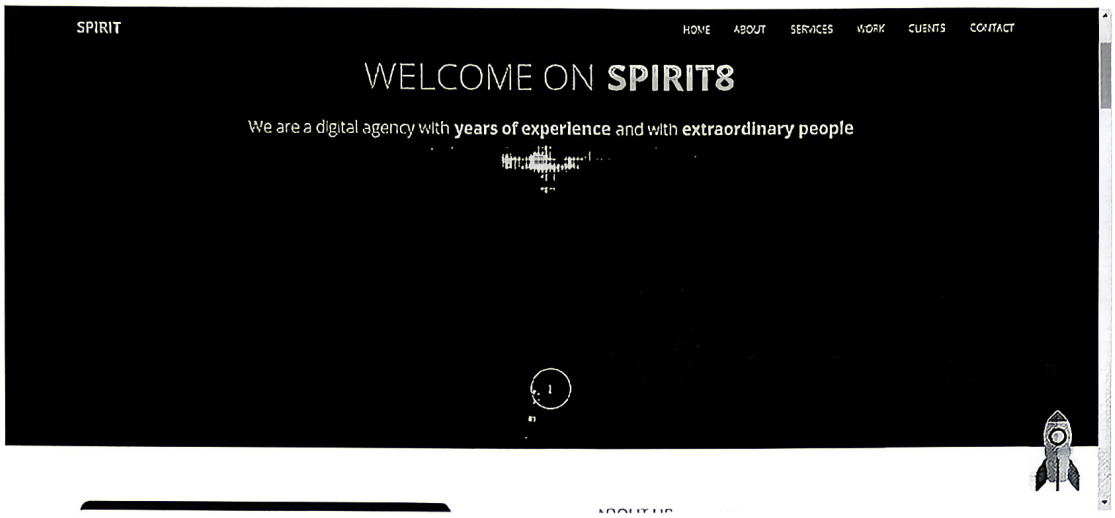


（3）下载 phpstudy

（4）并把文件放在 www 站点环境打开运行



（5）完成 spirit 响应式官网



3、实训结果:

（1）使用 bootstrap 框架（35 分）

（2）spirit 框架（35 分）

（3）phpstudy 搭建环境（30 分）

4、撰写实训报告

任务 2　开发 spirit 响应式官网

1、实训要求：运用 bootstrap 框架、spirit 框架完成头部导航、banner、内容模块、底部制作

2、操作步骤

（1）完成头部导航

```
<nav class="navbar navbar-inverse">
<div class="container">
<!-- logo 部分 -->
<div class="navbar-header">
<button type="button" class="navbar-toggle collapsed" data-toggle="coll
target="#bs-example-navbar-collapse-1" aria-expanded="false">
<span class="sr-only">Toggle navigation</span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
<a class="navbar-brand logo wow fadeInDown" href="#home">Spirit</a>
</div>

<!-- 导航部分 -->
<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1"
<ul class="nav navbar-nav navbar-right">
<li><a class="wow fadeInDown" data-wow-delay="0ms" href="#home">HOME</a
<li><a class="wow fadeInDown" data-wow-delay="100ms" href="#about">Abou
<li><a class="wow fadeInDown" data-wow-delay="200ms" href="#team">SERVI
<li><a class="wow fadeInDown" data-wow-delay="300ms" href="#work">work<
<li><a class="wow fadeInDown" data-wow-d
href="#clients">clients</a></li>
<li><a class="wow fadeInDown" data-wow-d
href="#contact">CONTACT</a></li>
</ul>
</div>
</div>
</nav>
</header>
```

(2) 完成 banner 图

```
<div id="home" class="banner">
<div class="container">
<!-- 标题 -->
<div class="title">
<span class="wow fadeInDown">wELCOME on</span>
<span class="wow fadeInDownBig" data-wow-delay="200ms">spirit8</span>
</div>

<!-- 描述 -->
<div class="info">
<span class="wow fadeInDown">We are a digital agency with</span>
<span class="wow fadeInDownBig" data-wow-delay="700ms">years of experience</span>
<span class="wow fadeInDown">and with</span>
<span class="wow fadeInDownBig" data-wow-delay="700ms">extraordinary people</span>
</div>
<!-- 按钮 -->
<a href="#" class="more wow fadeInDown" data-wow-delay="700ms">↓</a>
</div>
</div>
```

(3) 完成内容模块

```
<!-- 居中 bootstrap -->
<div class="container">
<!-- 左边图片 -->
<div class="left wow fadeInDown">
<img src="./assets/images/mac.png" />
</div>

<!-- 右边内容 -->
<div class="right">
<!-- 标题 -->
<div class="title">
```

```
<div class="short wow fadeInDown" data-wow-delay="100ms">About us</div>
<div class="long wow fadeInDown" data-wow-delay="400ms">
<span>Some</span>
words
<span class="bold">about us</span>
</div>
</div>


<div class="info wow fadeInDown" data-wow-delay="500ms">
We love building and rebuilding brands through our specific skills. Using
fonts, and illustration, we brand companies in a way they will never forge
</div>

<!-- 列表 -->
<ul class="list">
<li class="wow fadeInDown" data-wow-delay="200ms">
<span>Mission - </span>
We deliver uniqueness and quality
</li>
<li class="wow fadeInDown" data-wow-delay="400ms">
<span>Mission - </span>
We deliver uniqueness and quality
</li>
<li class="wow fadeInDown" data-wow-delay="600ms">
<span>Mission - </span>
We deliver uniqueness and quality
</li>
</ul>

<!-- 按钮 -->
<a href="javascript:void(0)" class="button wow fadeInDown" data-wow-delay=
<span class="glyphicon glyphicon-lock"></span>
<span class="name">Browse our work</span>
</a>
</div>
</div>
```

```
</div>
```

（4）完成底部

```
<footer class="footer">
<div class="container">
<div class="copyright wow fadeInDown">
ALL RIGHTS RESERVED. COPYRIGHT © 2014 SPIRIT8
</div>
<div class="links">
<a class="wow fadeInDown" data-wow-delay="100ms" href="javascript:void(0)">
<img src="./assets/images/footer1.png" />
</a>
<a class="wow fadeInDown" data-wow-delay="200ms" href="javascript:void(0)">
<img src="./assets/images/footer2.png" />
</a>
<a class="wow fadeInDown" data-wow-delay="300ms" href="javascript:void(0)">
<img src="./assets/images/footer3.png" />
</a>
<a class="wow fadeInDown" data-wow-delay="400ms" href="javascript:void(0)">
<img src="./assets/images/footer4.png" />
</a>
<a class="wow fadeInDown" data-wow-delay="500ms" href="javascript:void(0)">
<img src="./assets/images/footer5.png" />
</a>
</div>
</div>
</footer>
```

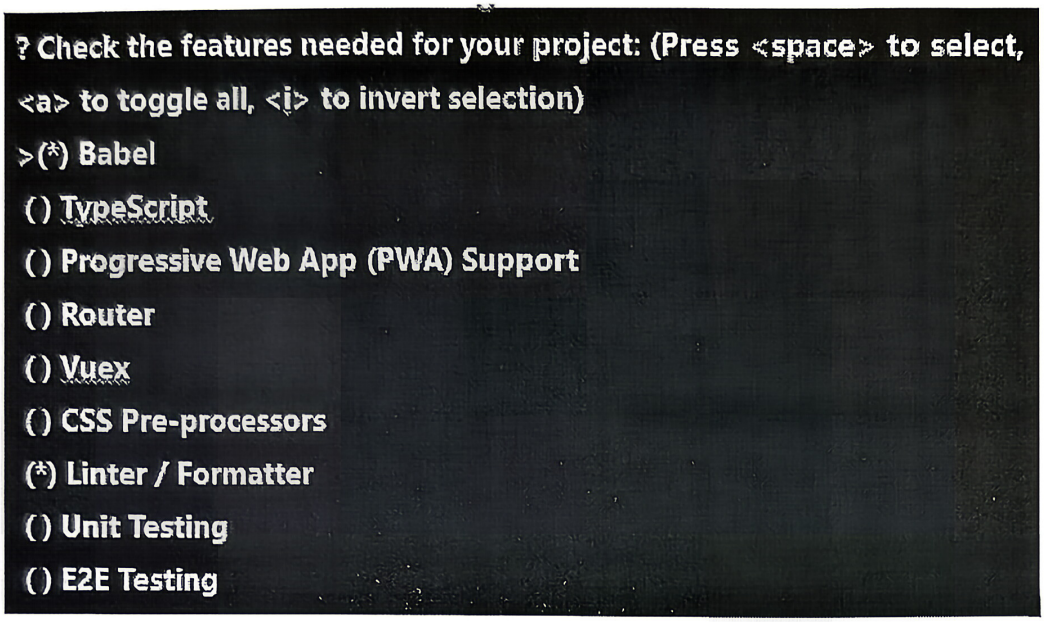
3、实训结果:

(1) 使用 bootstrap 完成页面布局（35 分）

（2）用 less 预处理语言编写样式（35 分）

（3）会使用 phpstudy 搭建环境（30 分）

4、撰写实训报告



3、实训结果:

（1）会创建并运行 vue 的项目（35 分）

（2）下载使用 vue-cli（35 分）

（3）使用 CLI 插件与第三方插件的使用方法（30 分）

4、撰写实训报告

项目二 Vue 购物车

实训目标：完成 Vue 购物车案例

实训技能：使用 Vuex 配置选项 Vuex 中的 API

实训内容：运用 Vuex 完成购物车案例

任务 1　Vue 购物车案例的实现

1、实训要求：了解 Vuex 的基本概念以及下载安装的方法、掌握购物车案例的实现过程。

2、操作步骤：

(1)Vuex 的下载和安装

npm 安装 vue-cli 包：在使用 webpack 进行 Vue 开发时，vue 和 vuex 通过 npm 安装的。

```
npm install vue-cli --save
```

创建 demo 项目：

```
vue init webpack demo
cd demo
```

npm 安装 vuex 包：

```
npm install vuex@3.1.1 --save
```

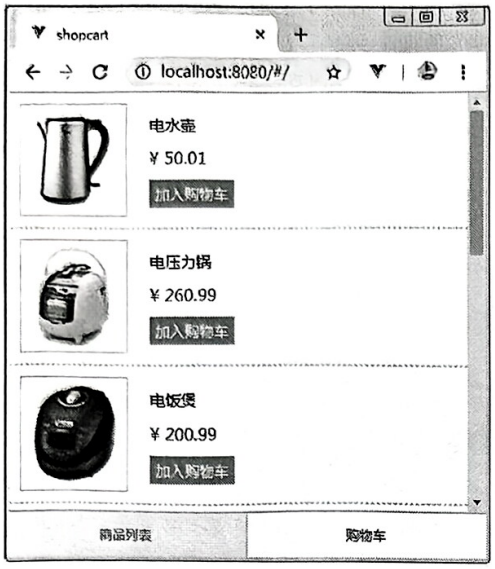
(2)案例分析

单击“加入购物车”按钮，即可将商品添加到购物车

在底部的 Tab 栏中切换到“购物车”页面，可以查看购物车中的商品

在底部显示商品的总价格

如果在购物中单击“删除”按钮，则表示删除商品



(3)通过 vue-cli 脚手架工具快速搭建项目。

```
vue init webpack shopcart  // 创建项目
cd shopcart  // 打开项目
npm install vuex@3.1.1 --save // 安装 vuex 包
npm run dev  // 启动 vue 项目在 http://localhost:8080
```

(4)实现底部 Tab 栏切换:创建 src\components\GoodsList.vue 和 Shopcart.vue 文件。

```
<template>
  <div>GoodsList</div>
</template>
<template>
  <div>Shopcart</div>
</template>
```

创建 src\router\index.js 文件，导入 vue、vue-router、GoodsList 和 Shopcart 组件。

```
import Vue from 'vue'
import Router from 'vue-router'
import GoodsList from '@/components/GoodsList'
import Shopcart from '@/components/Shopcart'
```

编写 src\router\index.js 文件，使用 vue-router 实现页面跳转。

```
Vue.use(Router)
  export default new Router({
  routes: [
  { path: '/', name: 'GoodsList', component: GoodsList },
  { path: '/shopcart', name: 'Shopcart', component: Shopcart }
  ]
})
```

修改 src\App.vue 文件，利用<router-link>实现 Tab 栏切换。

```
<template>
<div id="app">
```

```
        <div class="content">
          <router-view />
        </div>
      </div>
      </template>
```

修改 src\App.vue 文件，利用<router-link>实现 Tab 栏切换。

```
      <div class="bottom">
          <router-link to="/" tag="div">商品列表</router-link>
          <router-link to="/shopcart" tag="div">购物车</router-link>
      </div>
```

创建 src\api\shop.js 文件，定义 data 商品数据信息。

```
      const data = [
       {'id': 1, 'title': '电水壶', 'price': 50.01, src:
'/static/1.jpg'},
        …（省略内部商品数据信息）
    ]
    export default {
      getGoodsList (callback) {
        setTimeout(() => callback(data), 100)
      }
    }
```

编写 src\store\modules\goods.js 文件，管理商品 store。

```
     const state = {
      list: []
    }
      const getters = {}
```

编写 src\store\modules\goods.js 文件，将商品列表保存到 state 中。

```
     // 将商品列表保存到 state 中
       const mutations = {
       setList (state, data) {
       state.list = data
      }
    }
```

编写 src\store\modules\goods.js 文件，获取商品数据信息。

```
          import shop from '../../api/shop'
       // 获取商品列表数据
       const actions = {
         getList ({ commit }) {
           shop.getGoodsList(data => {
             commit('setList', data)
           })
         }
       }
```

创建 src\store\index.js 文件，加载 modules 目录下的 goods.js 和

shopcart.js 模块，并且导出 store 实例。

```
import goods from './modules/goods'
import shopcart from './modules/shopcart' '
export default new Vuex.Store({
  modules: {
    goods,
    shopcart
  }
})
```

修改 src\main.js 文件，将 store 实例导入到 Vue 实例的配置选项中。

```
import store from './store'
new Vue({
  ……（原有代码）
  store
})
```

修改 src\components\GoodsList.vue 文件，输出商品列表。

```
<template>
  <div class="list">
  <div class="item" v-for="goods in goodslist" :key="goods.id">
   …（商品列表信息）
  </div>
  </div>
</template>
```

修改 src\components\GoodsList.vue 文件，输出商品列表。

```
<script>
  import { mapState, mapActions } from 'vuex '
  export default{
  computed: mapState({goodslist: state => state.goods.list}),
  // 商品列表
  created () {this.$store.dispatch('goods/getList')},
  methods: mapActions('shopcart', ['add']),
  filters: {currency (value) { return '¥ ' + value}}
}
</script>
```

创建 src\store\modules\shopcart.js 文件，定义 items 用来保存购物车中的商品数据。

```
const state = {
  items: []
}
const getters = {}
const actions = {}
const mutations = {}
```

编写 src\store\modules\shopcart.js 文件实现购物车添加功能。

```
const actions = {
```

```
    add (context, item) {
      context.commit('add', item)
    },
  const mutations = {
    add (state, item) {添加功能逻辑代码},
  }
  const v = state.items.find(v => v.id === item.id)
      if (v) {
        ++v.num
      } else {
        state.items.push({
          id: item.id,
          title: item.title,
          price: item.price,
          src: item.src,
          num: 1
        })
      }
```

编写 src\store\modules\shopcart.js 文件，实现购物车删除功能。

```
  const actions = {
    del (context, id) {
      context.commit('del', id)
    }
  }
  const mutations = {
    del (state, id) {删除商品的逻辑代码}
  }
  state.items.forEach((item, index, arr) => {
        if (item.id === id) {
          arr.splice(index, 1)
        }
```

修改 src\store\modules\shopcart.js 文件，实现购物车总价格的计算。

```
  const getters = {
    totalPrice: (state) => {
      return state.items.reduce((total, item) => {
        return total + item.price * item.num
      }, 0) .toFixed(2)
    }
  }
```

修改 src\components\Shopcart.vue 文件，输出购物车列表。

```
  <template>
    <div class="list">
      …（购物车商品列表）
      <div class="item-total" v-if="items.length">
```

```
            商品总价：{{total | currency}}
          </div>
          <div class="item-empty" v-else>购物车中暂无商品</div>
        </div>
      </template>
```

修改 src\components\Shopcart.vue 文件，输出购物车列表。

```
import { mapGetters, mapState, mapActions } from 'vuex'
export default{
computed: {
...mapState({items: state => state.shopcart.items}),
// 购物车商品
...mapGetters('shopcart', { total: 'totalPrice '}) ,
// 总价格
 },
methods: mapActions('shopcart', ['del']),
// 删除购物车商品
filters: {currency (value) {return '¥ ' + value }} //添加¥符
号
}
```

3、实训结果:

(1) 单击“加入购物车”按钮，实现将商品添加到购物车（25 分）

（2）实现在底部的 Tab 栏中切换到“购物车”页面，可以查看购物车中的商品（25 分）

（3）实现在底部显示商品的总价格（25 分）

（4）实现在购物中单击“删除”按钮，删除商品（25 分）

4、撰写实训报告

模块三　前端开发实训

项目一 信息系统前端开发

实训目标：制作“信息系统”

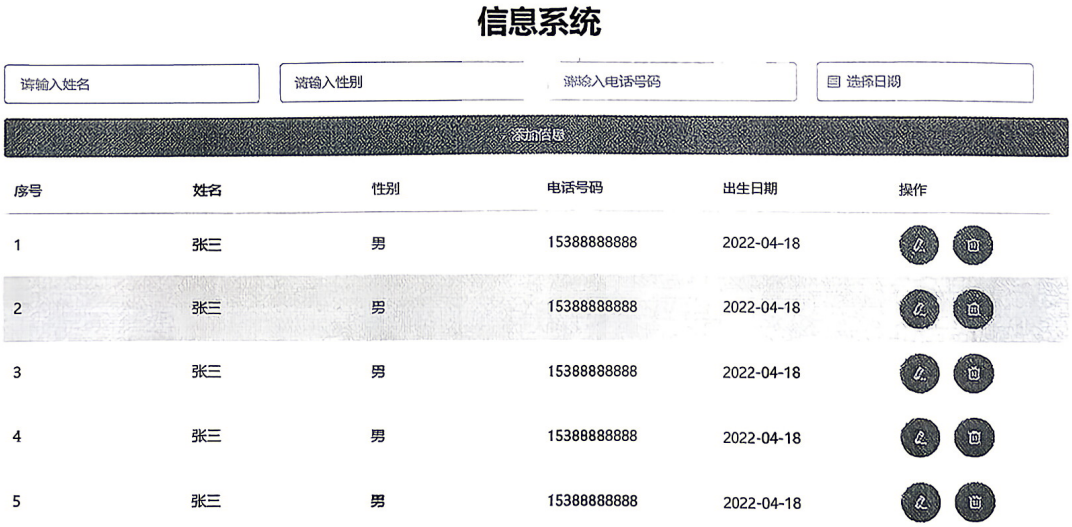
实训技能：element ui 的运用

实训内容：完成“信息系统”

任务 1 完成信息系统前端开发

1、实训要求：掌握 element ui 安装与使用、熟悉 Layout 布局、掌握 element ui 组件的使用、实现增加、删除、编辑功能

2、操作步骤：



（1）安装 element ui

通过 npm 安装 element ui

```
npm i element-ui -S
```

（2）引入 Element ui

在 main.js 中写入以下内容：

```
import Vue from 'vue';
import ElementUI from 'element-ui';
import 'element-ui/lib/theme-chalk/index.css';
```

```
import App from './App.vue';
Vue.use(ElementUI);
new Vue({
  el: '#app',
  render: h => h(App)
});
```

（3）使用 Layout 布局完成页面布局。

```
<div id="app">
        <!-- 输入框 -->
        <div class="head">
            <el-row :gutter="20">
                <el-col :span="6">
                    <div class="grid-content bg-purple">
                    <el-input              v-model="userInfo.name"
placeholder="请输入姓名"></el-input>
                    </div>
                </el-col>
                <el-col :span="6">
                    <div class="grid-content bg-purple">
                        <el-input        v-model="userInfo.gender"
placeholder="请输入性别"></el-input>
                    </div>
                </el-col>
                <el-col :span="6">
                    <div class="grid-content bg-purple">
                        <el-input      v-model="userInfo.phoneNum"
placeholder="请输入电话号码"></el-input>
                    </div>
                </el-col>
                <el-col :span="6">
                    <div class="grid-content bg-purple">
                        <el-date-picker                              v-
model="userInfo.birthday"  type="date"  placeholder="选择日期"
format="yyyy 年 MM 月 dd 日" value-format="yyyy-MM-dd">
                        </el-date-picker>
                    </div>
                </el-col>
            </el-row>
            <el-button type="primary" @click="addUser" class="but">
添加信息</el-button>
        </div>

        <div class="center">
            <template>
                <el-table
                    :data="tableData"
                    style="width: 100%">
                <el-table-column
                    label="序号"
                    width="180">
                <template slot-scope="scop
                    {{scope.$index + 1}}
                </template>
                </el-table-column>
                <el-table-column prop="nam
                </el-table-column>
                <el-table-column prop="gen
                </el-table-column>
                <el-table-column prop="pho
                </el-table-column>
                <el-table-column prop="bir
                </el-table-column>
                <el-table-column prop="bir
                <template slot-scope="scop
                <el-button        type="pr
edit"circle@click="editUser(scope.row)"></
                <el-button        type="da
delete"circle@click="delUser(scope.$index)
                    </template>
                </el-table-column>
                </el-table>
                </template>
            </div>
             <el-dialog     title="   编
"       :visible.sync="dialogVisible"
close="handleClose">
                <div>
                    <el-form   ref="form"
width="80px">
                        <el-form-item labe
                            <el-inputv-mod
input>
                        </el-form-item>
                        <el-form-item label
                            <el-inputv-mode
input>
                        </el-form-item>
                        <el-form-item label
```

```
                <el-table
                    :data="tableData"
                    style="width: 100%">
                <el-table-column
                    label="序号"
                    width="180">
                <template slot-scope="scope">
                    {{scope.$index + 1}}
                </template>
                </el-table-column>
                <el-table-column prop="name" label="姓名" width="180">
                </el-table-column>
                <el-table-column prop="gender" label="性别">
                </el-table-column>
                <el-table-column prop="phoneNum" label="电话号码">
                </el-table-column>
                <el-table-column prop="birthday" label="出生日期">
                </el-table-column>
                <el-table-column prop="birthday" label="操作">
                <template slot-scope="scope">
                <el-button        type="primary"        icon="el-icon-
edit"circle@click="editUser(scope.row)"></el-button>
                <el-button        type="danger"        icon="el-icon-
delete"circle@click="delUser(scope.$index)"></el-button>
                    </template>
                </el-table-column>
                </el-table>
                </template>
            </div>
              <el-dialog     title="    编    辑    用    户    信    息
"       :visible.sync="dialogVisible"       width="30%"       :before-
close="handleClose">
                <div>
                    <el-form   ref="form"   :model="editobj"   label-
width="80px">
                        <el-form-item label="姓名">
                            <el-inputv-model="editobj.name"></el-
input>
                        </el-form-item>
                        <el-form-item label="性别">
                            <el-inputv-model="editobj.gender"></el-
input>
                        </el-form-item>
                        <el-form-item label="电话号码">
```

```
                        <el-inputv-model="editobj.phoneNum"></el-
input>
            </el-form-item>
                <el-form-item label="出生日期">
                 <el-col :span="6">
   <div class="grid-content bg-purple">
<el-date-pickerv-model="editobj.birthday" type="date" placeholder="选
择日期" format="yyyy 年 MM 月 dd 日" value-format="yyyy-MM-dd">
          </el-date-picker>
                          </div>
                      </el-col>
                  </el-form-item>
              </el-form>
            </div>
            <span slot="footer" class="dialog-footer">
          <el-button @click="dialogVisible = false">取 消</el-
button>
          <el-button type="primary" @click="dialogVisible = false">
确 定</el-button>
        </span>
        </el-dialog>
      </div>
```

(4) 实现增加功能

```
  methods: {
            // 添加用户信息
            addUser() {
                // 表单验证
                if (!this.userInfo.name) {
                    this.$message({
                        showClose: true,
                        message: '请输入姓名',
                        type: 'warning'
                    });
                    return;
                }
                if (!this.userInfo.gender) {
                    this.$message({
                        showClose: true,
                        message: '请输入性别',
                        type: 'warning'
                    });
                    return;
                }
                if (!this.userInfo.phoneNum) {
```

```
                    this.$message({
                        showClose: true,
                        message: '请输入电话号码',
                        type: 'warning'
                    });
                    return;
                }
    // 验证手机号码格式 正则
                if
(!/^1[23456789]\d{9}$/.test(this.userInfo.phoneNum)) {
                    this.$message({
                        // showClose: true,
                        message: '请输入正确的电话号码',
                        type: 'warning'
                    });
                    return;
                }
```

（5）实现删除功能

```
this.userInfo = {
                    name: '',
                    gender: '',
                    phoneNum: '',
                    birthday: '',
                }
            },
            // 删除数据
            delUser(inx) {
                this.$confirm('确认删除吗？')
                    .then(_ => {
                        this.tableData.splice(inx, 1);
                    })
                    .catch(_ => {});
                // this.tableData.splice(inx, 1); 删除不提示
            },
```

（6）：实现编辑功能

```
editUser(item) {
                this.editobj = {
                    name: item.name,
                    gender: item.gender,
                    phoneNum: item.phoneNum,
                    birthday: item.birthday,
                };
                this.dialogVisible = true;
                console.log(item);
```

```
            }
            handleClose() {
                this.dialogVisible = false;

            }

        },

    })
```

3、实训结果:

(1) 安装引入Element ui (20分)

(2) 完成页面布局 (30分)

(3) 实现完成增加、删除、编辑功能 (50分)

4、撰写实训报告

HNBC
广州华南商贸职业学院
Guangzhou Huanan Business College

《Android物联网实训应用程序开发实践》

编者姓名：毛振宁、林桂兰

广州华南商贸职业学院云智信息技术学院

广州腾科网络技术有限公司

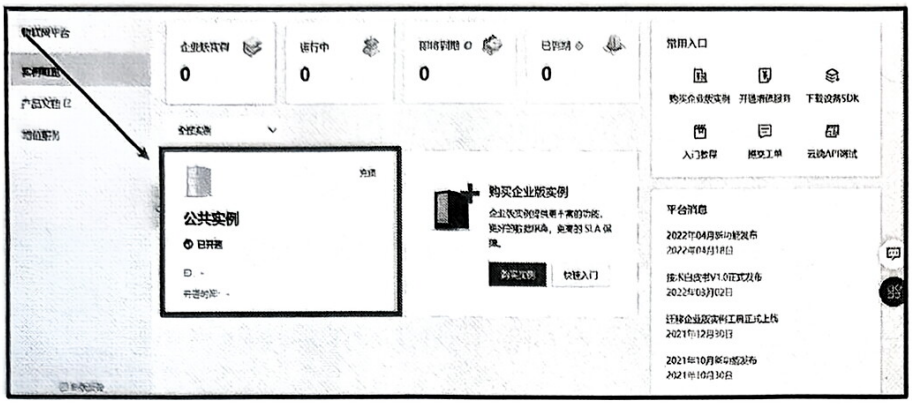
《Android 物联网实训应用程序开发实践》实训

毛振宁

广州华南商贸职业学院云智信息学院

目 录

步骤三：开通公共实例



3、实训结果

4、撰写实训报告

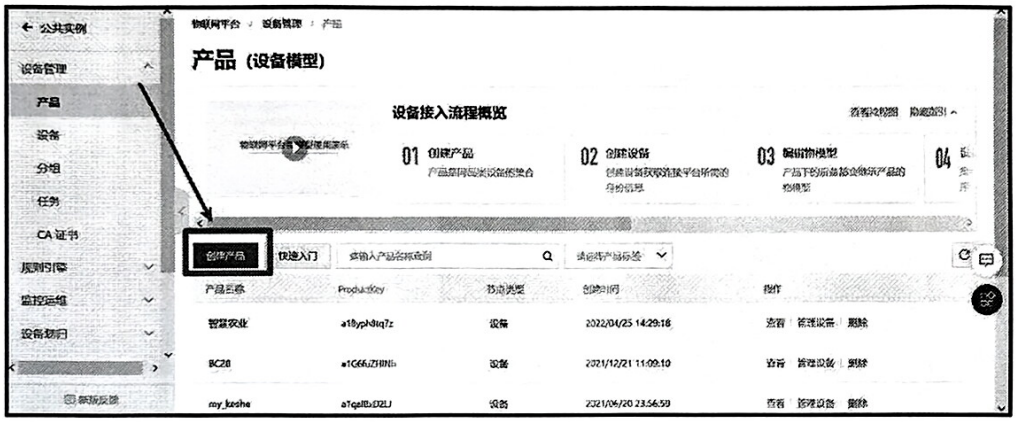
经过实训，学生能够掌握阿里云平台智慧农业产品的创建

任务 2 创建阿里云物联网平台的产品

1、实训要求：掌握创建阿里云物联网平台产品的方法

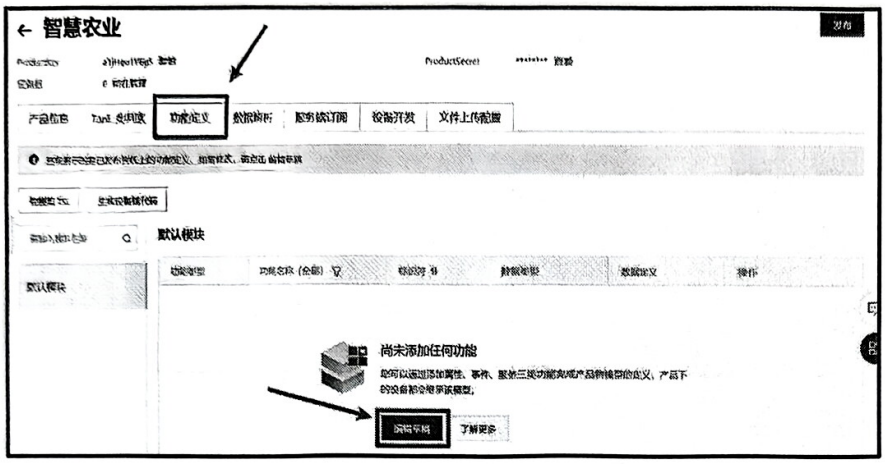
2、操作步骤

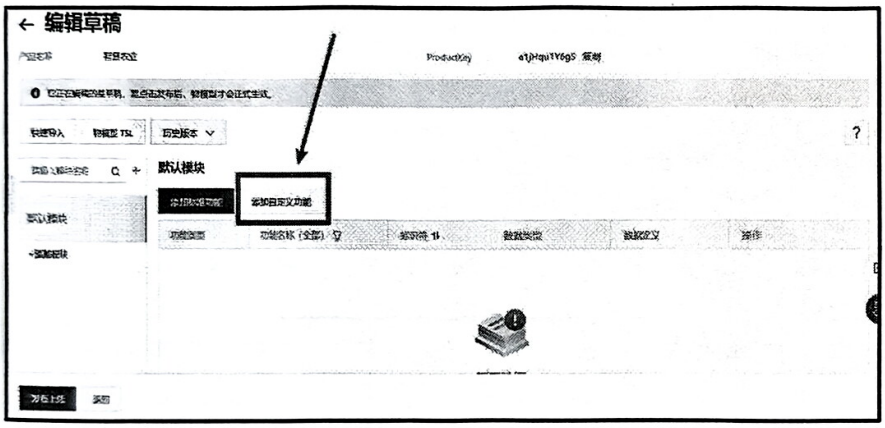
步骤一：创建产品

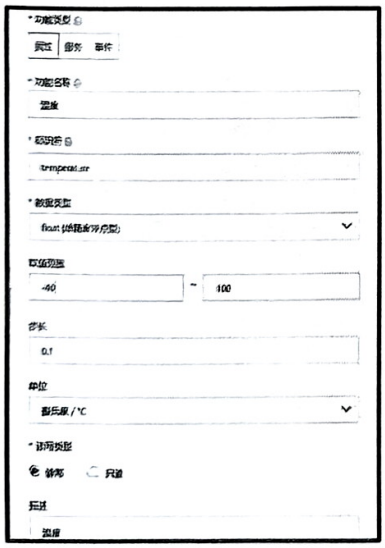


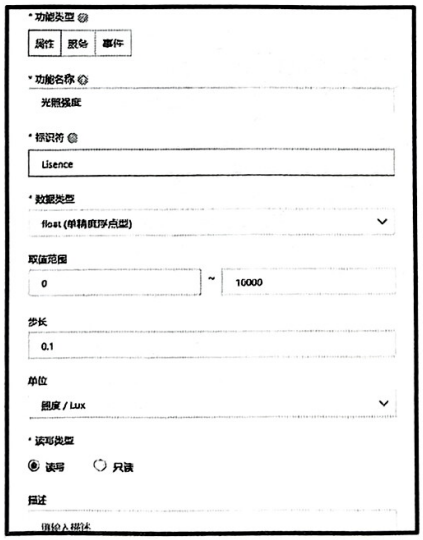
步骤二：填写产品信息并确认

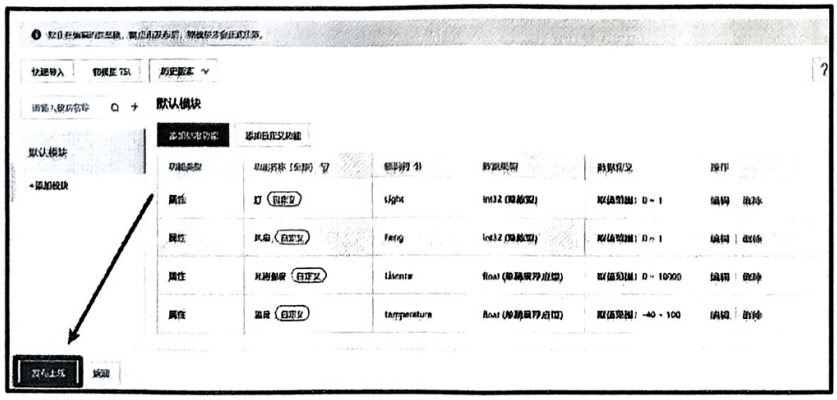
步骤三：添加产品物模型，需要添加四个模型，温度、光照强度、风扇、灯，并发布上线。


智慧农业
功能定义
默认模块
尚未添加任何功能
编辑草稿

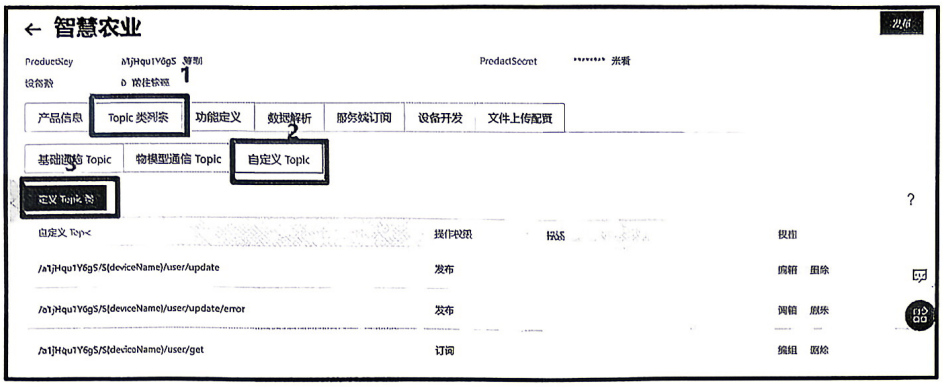

编辑草稿
默认模块
添加自定义功能

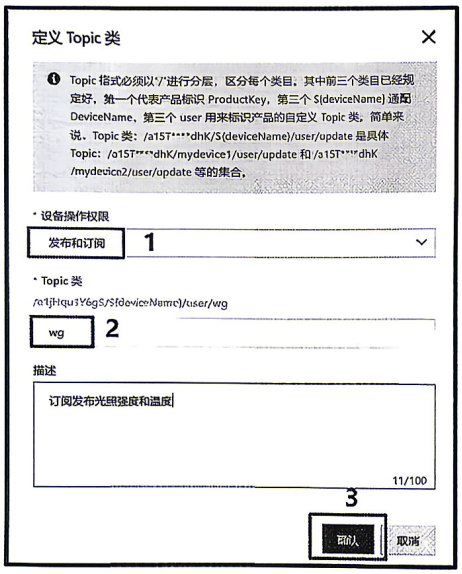

温度
temperature
-40
100
0.1

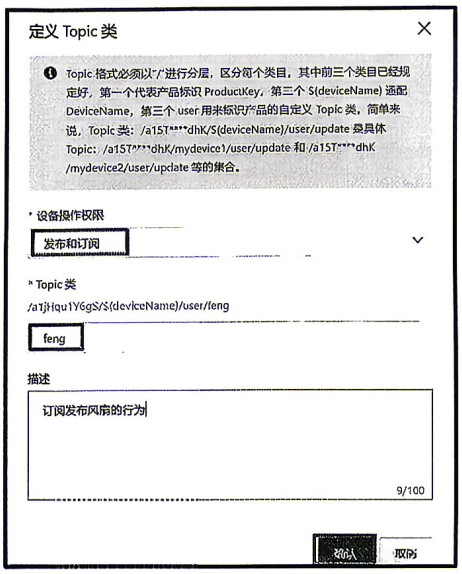

光照强度
Lisence
0
10000
0.1

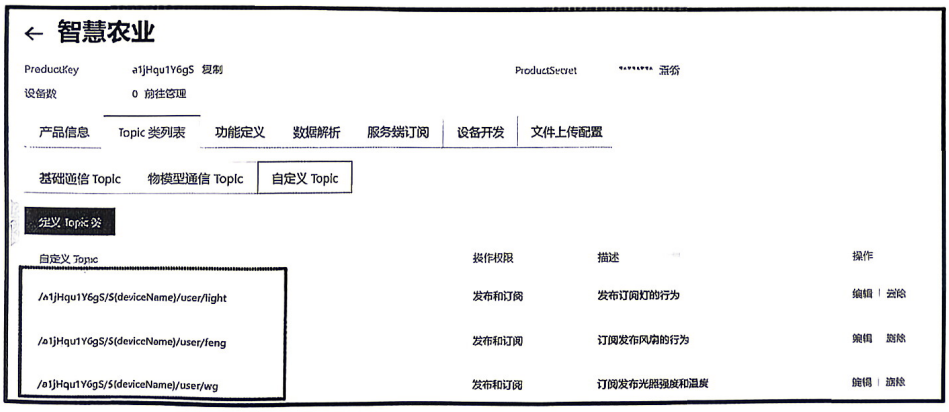

默认模块
发布上线

步骤三：定义 Topic 类，定义三个 Topic，分别如下图的定义：



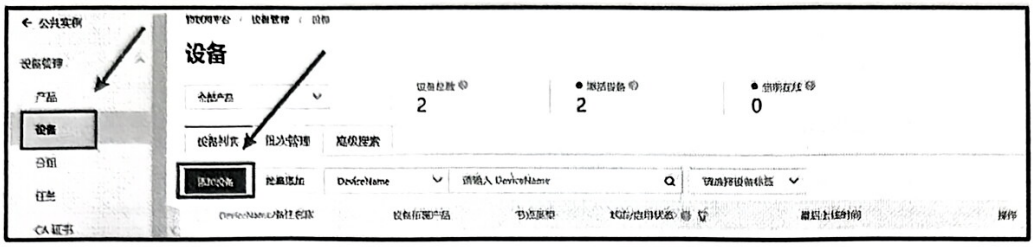


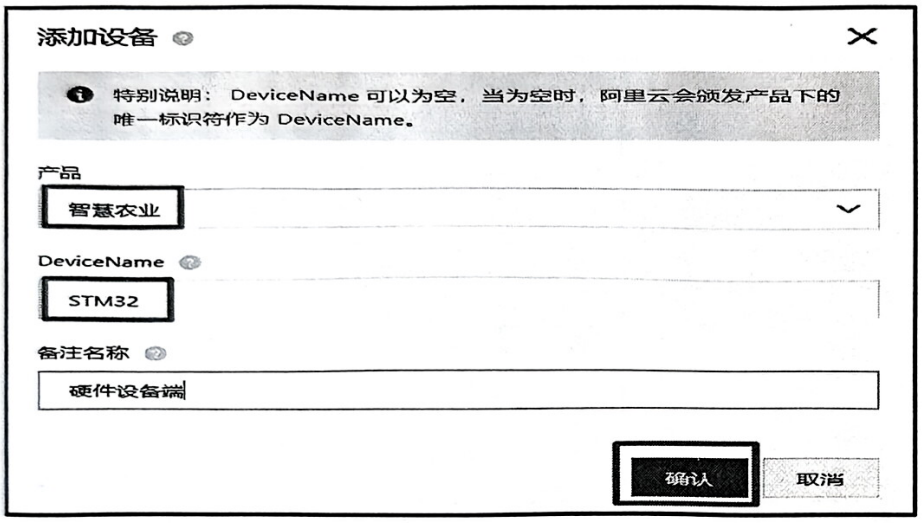


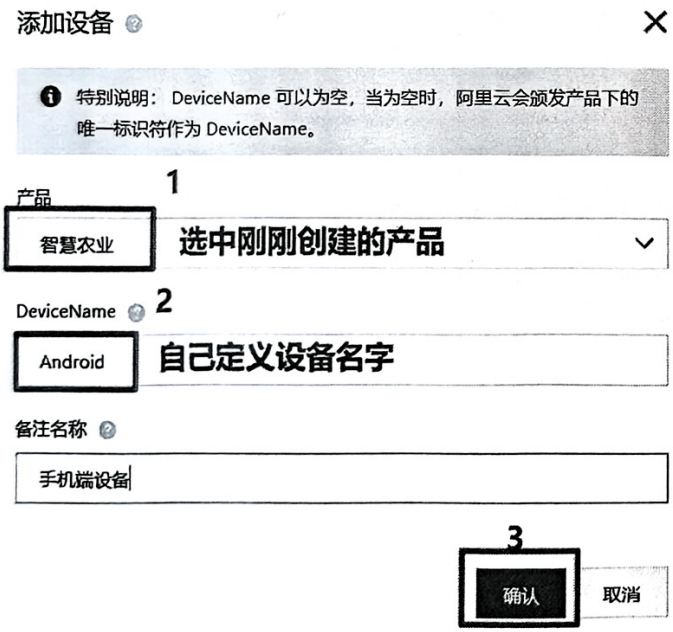


到此产品创建成功

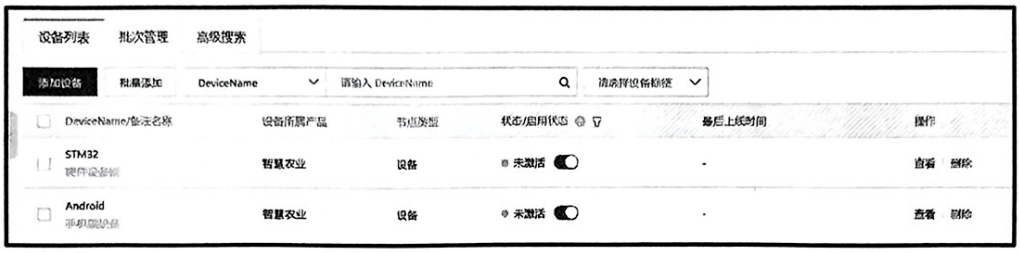
步骤四：创建设备。需要创建两个设备，一个是对应 Android 端的设备，一个是对应硬件端的设备。







3、实训结果



4、撰写实训报告

经过实训，学生能够已经创建好设备，刚刚创建好的设备显示的是未激活状态。

项目二 云产品流转的规则创建

实训目标：创建"云产品流转的规则"，掌握 Android 的物联网应用

实训技能：Android 的使用，物联网平台的使用，NB-IoT 的使用

实训内容：

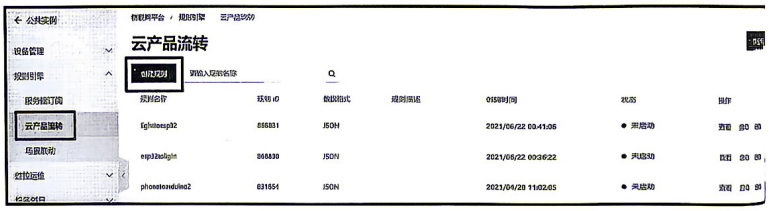
任务 1 规则的创建

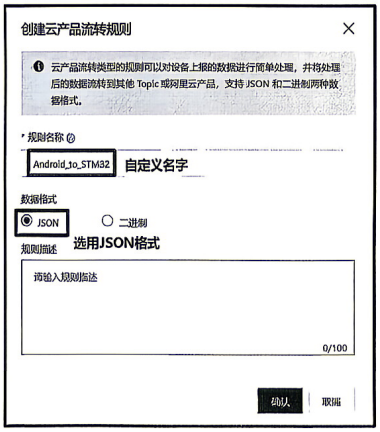
1、实训要求：理解掌握阿里云产品流转的规则使用

2、操作步骤

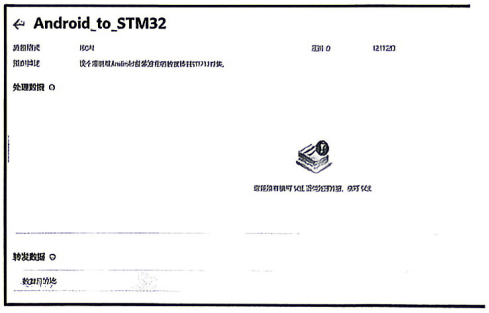
进入云产品流转页面，创建规则。我们需要创建两个规则，一个是 Android 设备发布的数据传递到 STM32 设备，另一个是 STM32 设备发布的数据传递到 Android 设备端。

步骤一：创建规则一，Android 设备发布的数据传递到 STM32 设备

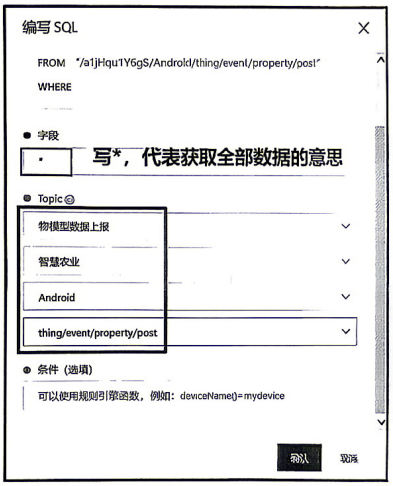




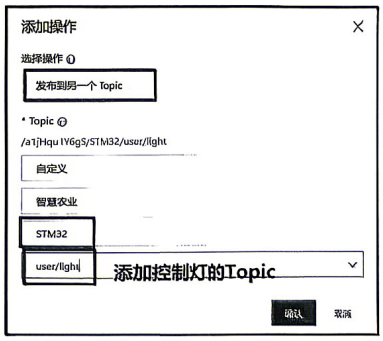
创建成功后查看规则

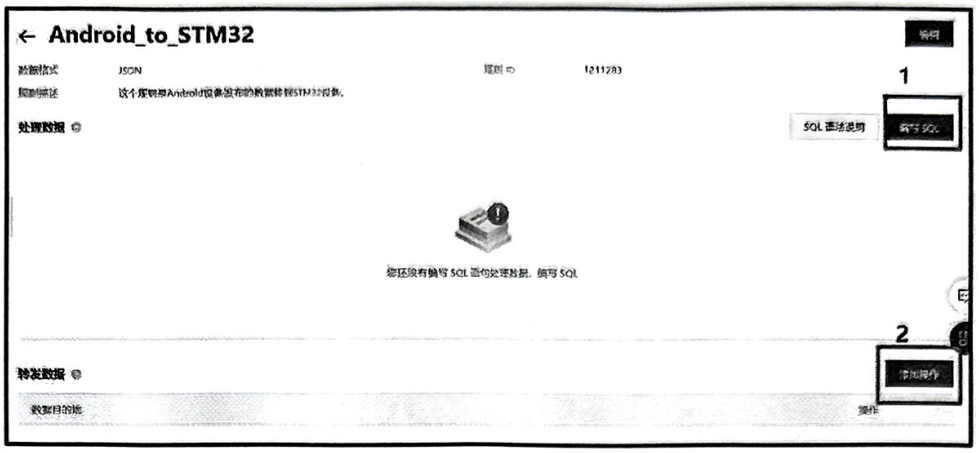


编写 SQL



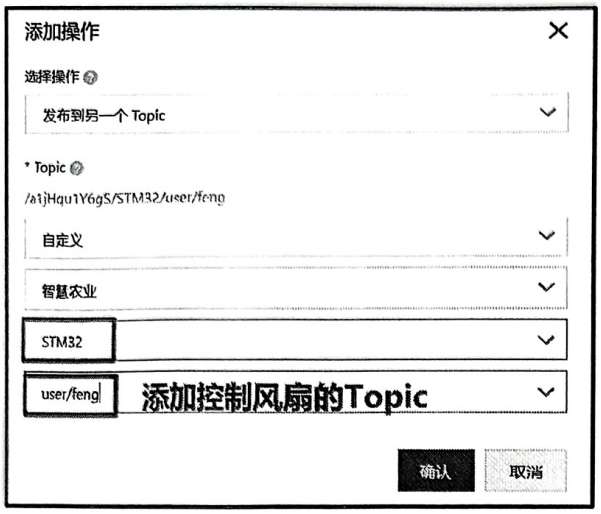
添加操作，添加两个 Topic 操作，一个是控制灯的，一个;



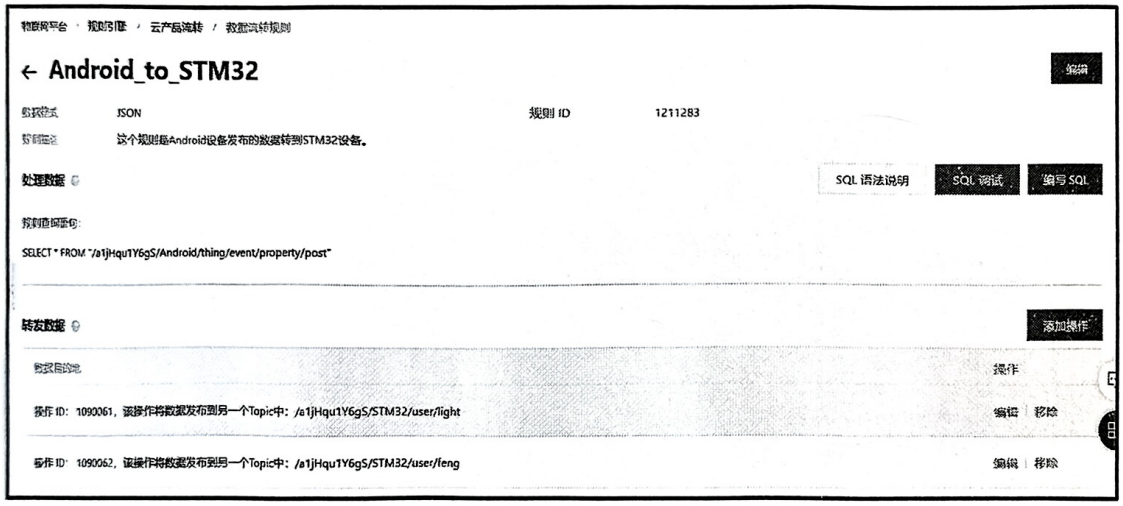


编写 SQL

添加操作，添加两个 Topic 操作，一个是控制灯的，一个是控制风扇的。

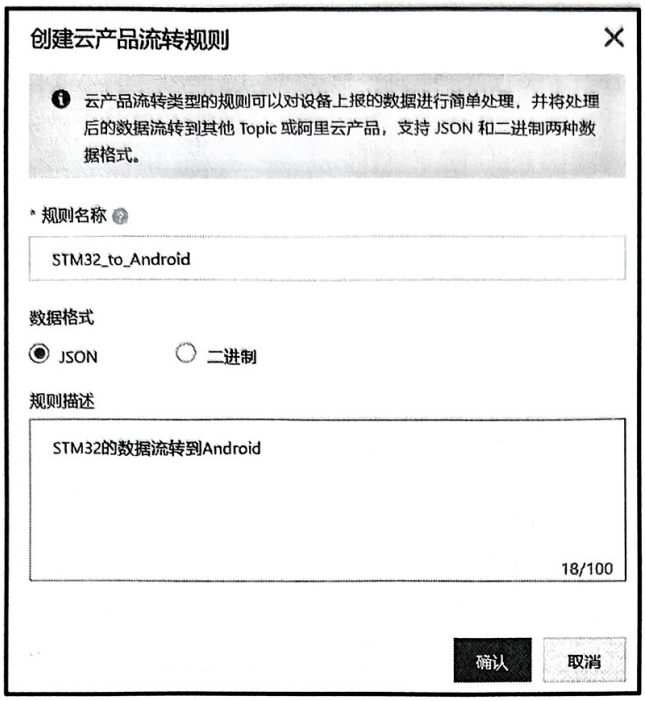


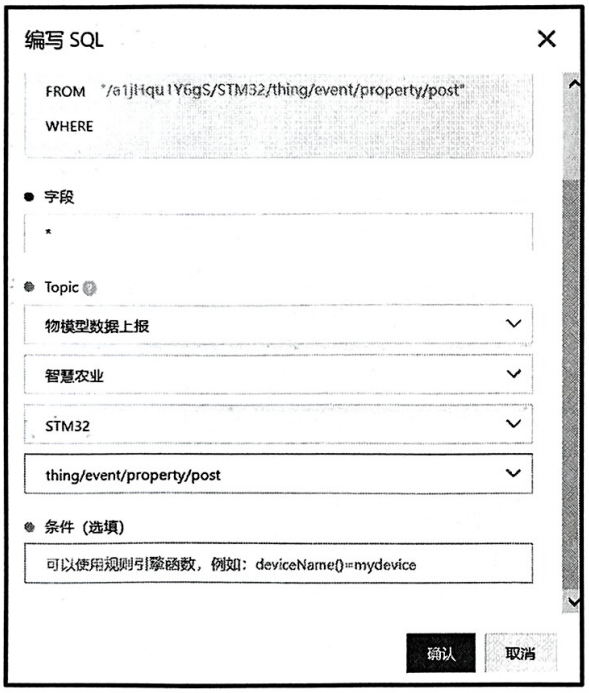
添加成功，要启动规则



启动后的规则显示运行中状态。

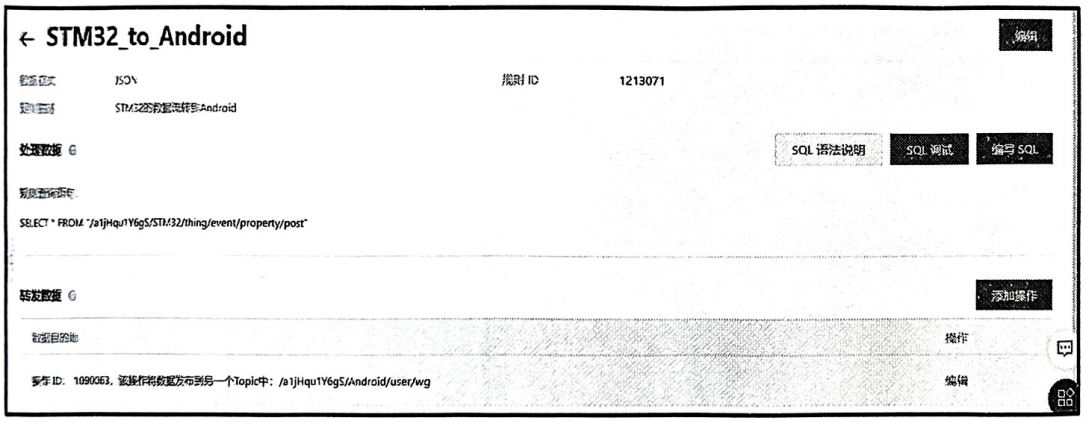
步骤二：创建规则二，STM32 设备发布的数据传递到 Android 设备端



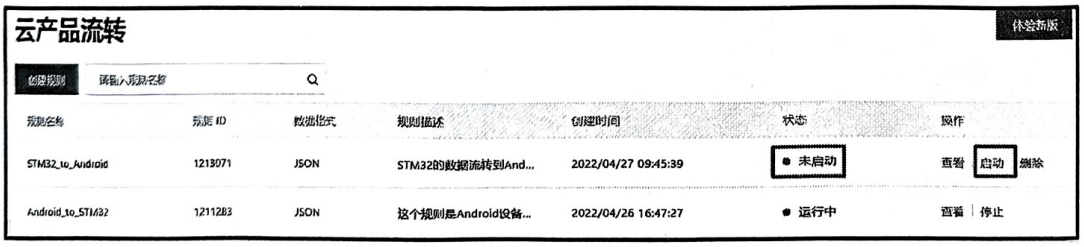


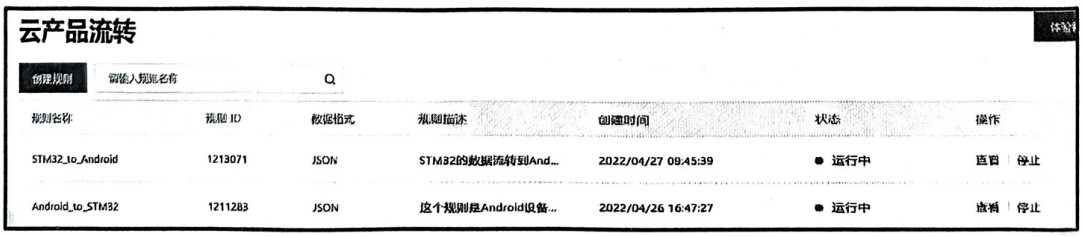
添加操作，转发光照强度和温度到 Android 端。


添加操作
选择操作
发布到另一个 Topic
* Topic
/a1jHqu1Y6gS/Android/user/wg
自定义
智慧农业
Android
user/wg
确认
取消


STM32_to_Android
编辑
JSON
规则 ID
1213071
处理数据
SQL 语法说明
SQL 调试
编写 SQL
SELECT * FROM "/a1jHqu1Y6gS/STM32/thing/event/property/post"
转发数据
添加操作
操作
编辑

启动规则


云产品流转
体验新版
创建规则
规则名称
规则 ID
数据格式
规则描述
创建时间
状态
操作
STM32_to_Android
1213071
JSON
STM32的数据流转到And...
2022/04/27 09:45:39
未启动
查看
启动
删除
Android_to_STM32
1211283
JSON
这个规则是Android设备...
2022/04/26 16:47:27
运行中
查看
停止


云产品流转
创建规则
规则名称
规则 ID
数据格式
规则描述
创建时间
状态
操作
STM32_to_Android
1213071
JSON
STM32的数据流转到And...
2022/04/27 09:45:39
运行中
查看
停止
Android_to_STM32
1211283
JSON
这个规则是Android设备...
2022/04/26 16:47:27
运行中
查看
停止

3、实训结果

4、撰写实训报告

经过实训，学生到此把两个规则添加完成。接下来我们使用 MQTT.fx 配合物联网平台测试一下规则是否添加完成。

我们使用平台 Android 设备的设备调试发布数据模拟 STM32 发布数据.发布温度 28°，光照强度 90Lux，使用 MQTT.fx 模拟 Android 设备接首数据，查看是否成功接受 STM32 发布的数据。

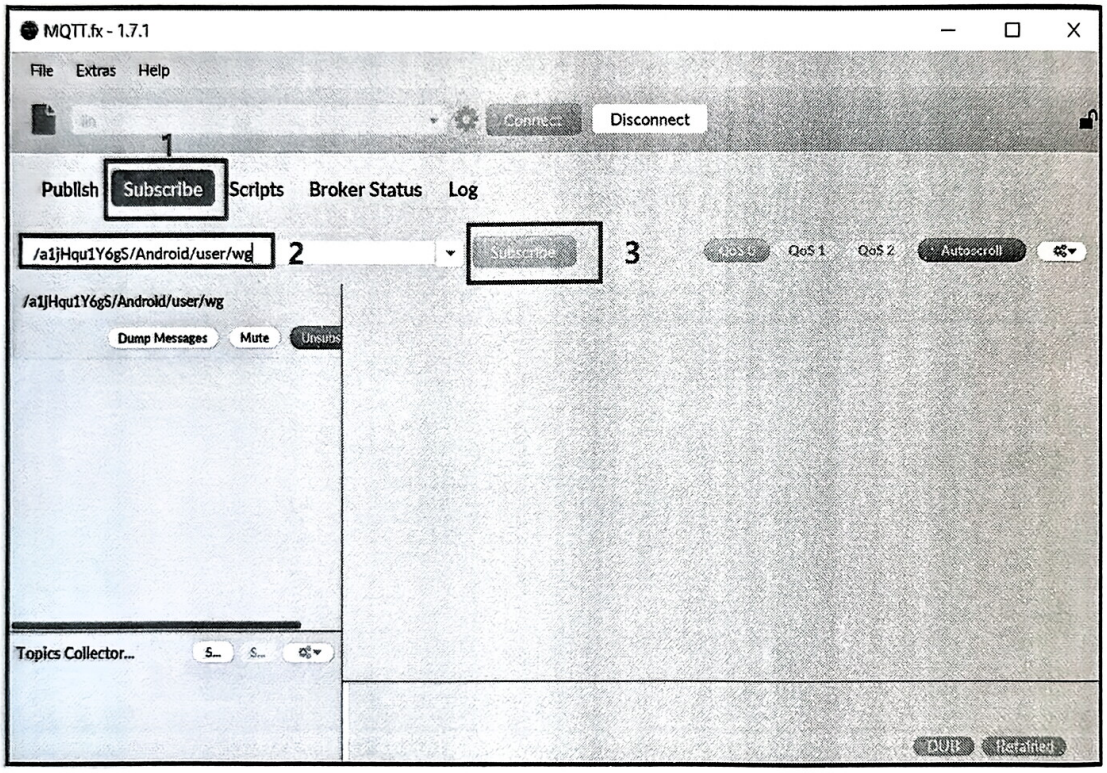
任务 2 温湿度数据的测试

1、实训要求：理解掌握阿里云温湿度数据的测试方法

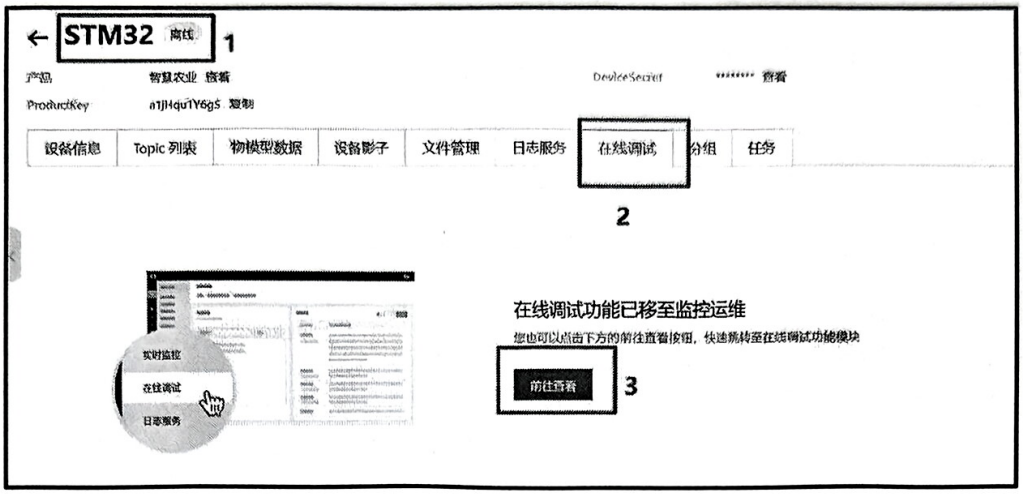
2、操作步骤

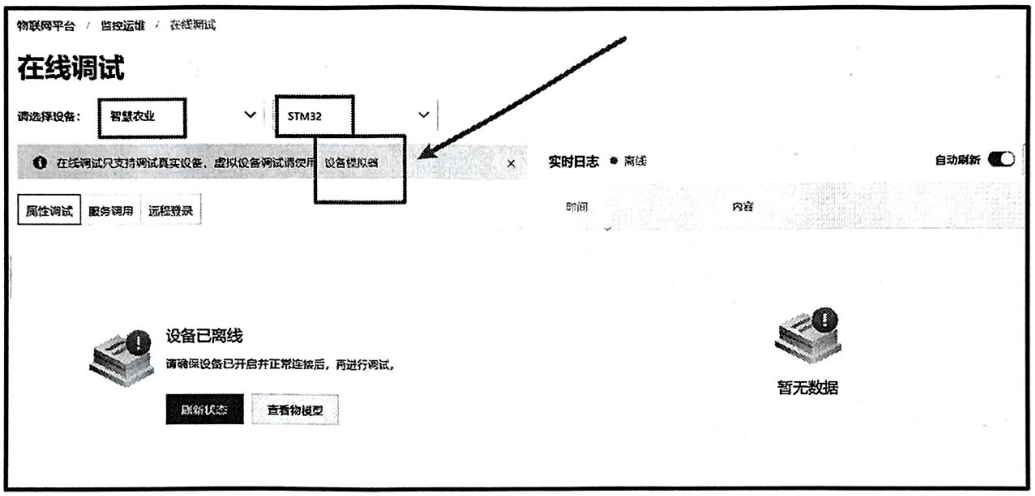
首先测试一下 STM32 发布的温湿度数据，Android 能否收到.

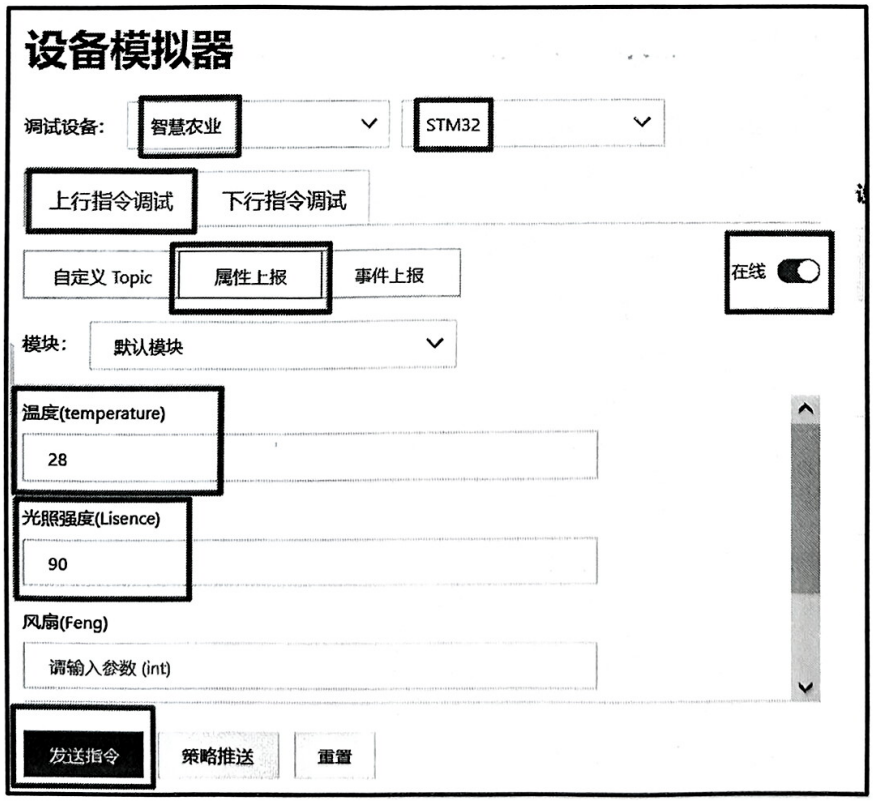
步骤一：打开 MQTT.fx 连接 Android 设备，并订阅我们定义好的 Topic



步骤二：打开平台 STM32 设备的设备调试并发送数据

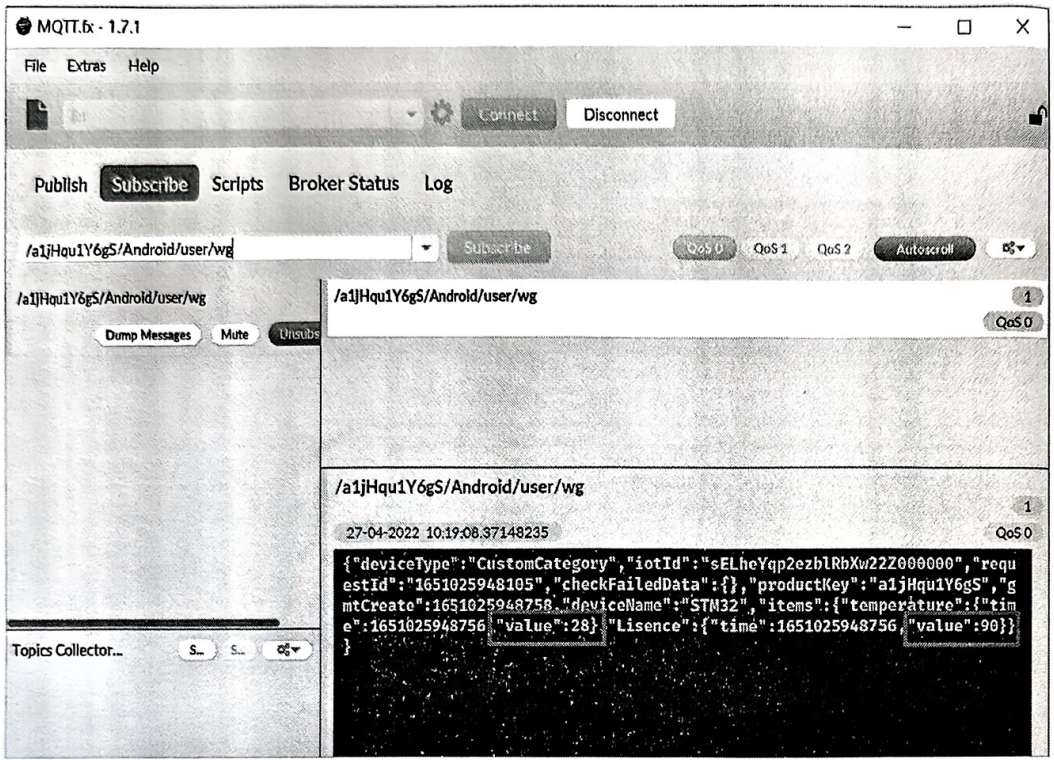






3、实训结果

Android 端成功接收 STM32 数据。



4、撰写实训报告

同学们可以根据同样方式测试 Android 端发布的数据 STM32 是否能成功接收。

模块二　Android 通过 MQTT 连接阿里云平台

项目一 阿里云平台 Android 的 MQTT 使用

实训目标：掌握 Android 的物联网应用

实训技能：Android 的使用，物联网平台的使用，NB-IoT 的使用

实训内容：

任务 1 建立一个空的 Android 新工程

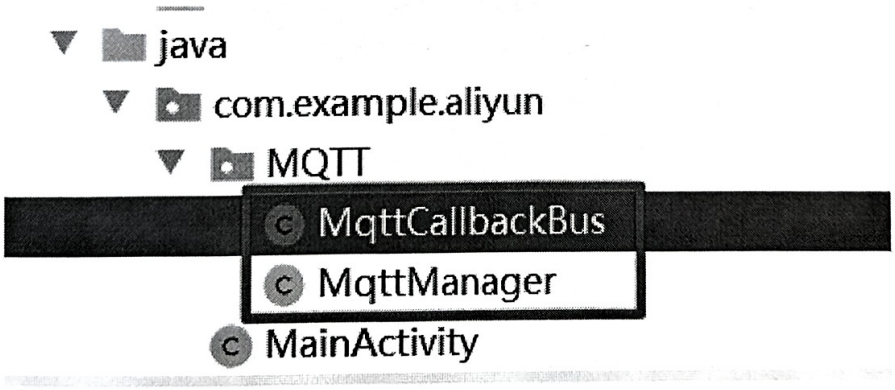
1、实训要求：掌握 Android 的新项目的建立

2、操作步骤

步骤一：首先建立一个空的新工程
步骤二：导入 MQTT 相关的包

aliyun-java-sdk-iot.jar	2021/4/25 12:48	Executable Jar File	9,033 KB
eventbus-3.0.0.jar	2019/12/3 16:33	Executable Jar File	53 KB
fastjson-1.2.5.jar	2015/12/16 15:00	Executable Jar File	402 KB
loger.jar	2019/12/3 9:35	Executable Jar File	5 KB
org.eclipse.paho.client.mqttv3-1.0.2.jar	2019/12/3 11:44	Executable Jar File	168 KB

3、实训结果



4、撰写实训报告

通过实训，学生能够掌握 Android 新项目的建立办法。

任务 2 使用 Android 常见的控件

1、实训要求：掌握 Android 的常用控件的使用方法

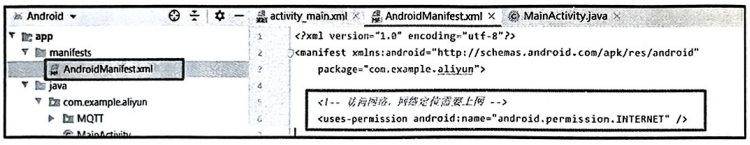
2、操作步骤

步骤一：在 AndroidMainfest.xml 文件里添加语句 。

```
<!-- 访问网络，网络定位需要上网 -->
<uses-permission android:name="android.permission.INTERNET" />
```

只有添加这句话，Android 才能访问网络。

步骤二：添加 xml 界面，如下图所示。



步骤三：添加两个 MQTT 类。

aliyun-java-sdk-iot.jar	2021/4/25 12:48	Executable Jar File	9,033 KB
eventbus-3.0.0.jar	2019/12/3 16:33	Executable Jar File	53 KB
fastjson-1.2.5.jar	2015/12/16 15:00	Executable Jar File	402 KB
loger.jar	2019/12/3 9:35	Executable Jar File	5 KB
org.eclipse.paho.client.mqttv3-1.0.2.jar	2019/12/3 11:44	Executable Jar File	168 KB

3、实训结果

4、撰写实训报告

通过实训，学生能够掌握 Android 常用控件的使用方法。

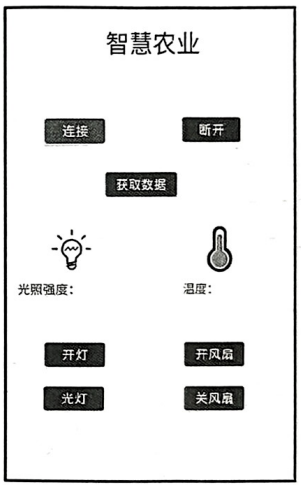
任务 3 xml 界面的测试

1、实训要求：掌握 Android 的 xml 界面的测试方法

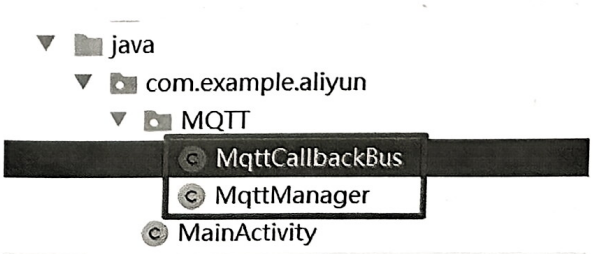
2、操作步骤

步骤一：连接 7 个绿色的 Button。

步骤二：测试光照强度和温度右边的一个空白的 TextView。



步骤三：添加两个 MQTT 类



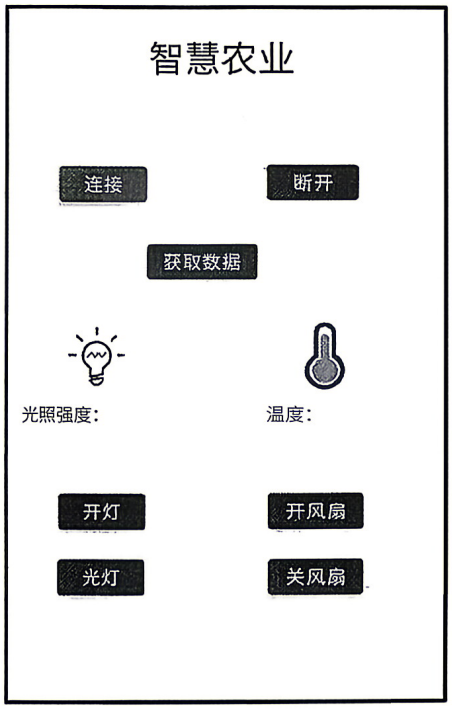
任务 3 xml 界面的测试

1、实训要求：掌握 Android 的 xml 界面的测试方法

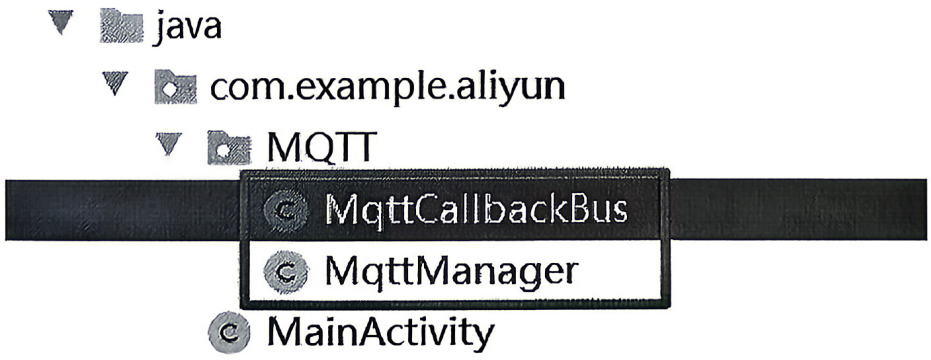
2、操作步骤

步骤一：连接 7 个绿色的 Button。

步骤二：测试光照强度和温度右边的一个空白的 TextView。



步骤三：添加两个 MQTT 类



HNBC
广州华南商贸职业学院
Guangzhou Huanan Business College

《数据可视化实战》

编者姓名：徐胜东、赵文才

广州华南商贸职业学院云智信息技术学院

广州粤嵌通信科技股份有限公司

《数据可视化实战》

编者：徐胜东、赵文才

广州华南商贸职业学院云智信息学院

广州粤嵌通信科技股份有限公司

目 录

模块一　数据可视化实训

项目一 数据可视化的 Javaweb 系统

实训目标：

掌握可视化工具 echarts 的使用

掌握网页开发技术

掌握 MVC 框架开发技术

掌握 Javaweb 系统的开发流程

实训内容：前端使用 HTML+jQuery+ECharts、后台使用 Spring+SpringMVC+mybatis+mysql 来完成一个数据可视化的 Javaweb 系统

任务 1 HTML

1、实训要求

掌握 HTML 常用元素的使用

2、操作步骤

1）、HTML 元素语法

HTML 元素以开始标签起始

HTML 元素以结束标签终止

元素的内容是开始标签与结束标签之间的内容

某些 HTML 元素具有空内容

空元素在开始标签中进行关闭（以开始标签的结束而结束）

大多数 HTML 元素可拥有属性

大多数 HTML 元素可以嵌套（HTML 元素可以包含其他

HTML 元素）。

HTML 文档由相互嵌套的 HTML 元素构成。

```
<!DOCTYPE html>
<html>

<body>
<p>这是第一个段落。</p>
</body>

</html>
```

2）、HTML 属性

HTML 元素可以设置属性

属性可以在元素中添加附加信息

属性一般描述于开始标签

属性总是以名称/值对的形式出现，比如：name="value"。

这是一个链接

适用于大多数 HTML 元素的属性：

属性	描述
class	为 html 元素定义一个或多个类名（classname 样式文件引入)
id	定义元素的唯一 id

style	规定元素的行内样式（inline style）	
title	描述了元素的额外信息 (作为工具条使用)	

3）、HTML 标题

标题（Heading）是通过 <h1> - <h6> 标签进行定义的。

<h1> 定义最大的标题。 <h6> 定义最小的标题。

```
<h1>这是一个标题。</h1>

<h2>这是一个标题。</h2>

<h3>这是一个标题。</h3>
```

4）、HTML 段落

HTML 可以将文档分割为若干段落。

段落是通过 <p> 标签定义的。

```
<p>这是一个段落 </p>

 <p>这是另一个段落</p>
```

5）、HTML 超链接（链接）

HTML 使用标签 <a>来设置超文本链接。

超链接可以是一个字，一个词，或者一组词，也可以是一幅图像，您可以点击这些内容来跳转到新的文档或者当前文档中的某个部分。

当您把鼠标指针移动到网页中的某个链接上时，箭头会变为一只小手。

在标签<a> 中使用了 href 属性来描述链接的地址。

默认情况下，链接将以以下形式出现在浏览器中：

一个未访问过的链接显示为蓝色字体并带有下划线。

访问过的链接显示为紫色并带有下划线。

点击链接时，链接显示为红色并带有下划线。

```
<a href="url">链接文本</a>
```

6）HTML 头部

<title> - 定义了 HTML 文档的标题

使用 <title> 标签定义 HTML 文档的标题

<base> - 定义了所有链接的 URL

使用 <base> 定义页面中所有链接默认的链接目标地址。

<meta> - 提供了 HTML 文档的 meta 标记

使用 <meta> 元素来描述 HTML 文档的描述，关键词，作者，字符集等。

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>文档标题</title>
```

```
</head>
<body> 文档内容.....
</body>
</html>
```

7）HTML 图像- 图像标签（ <img>）和源属性（Src）

在 HTML 中，图像由<img> 标签定义。

<img> 是空标签，意思是说，它只包含属性，并且没有闭合标签。

要在页面上显示图像，你需要使用源属性（src）。src 指"source"。源属性的值是图像的 URL 地址。

定义图像的语法是：

<img src="url" alt="some_text">

URL 指存储图像的位置。

浏览器将图像显示在文档中图像标签出现的地方。如果你将图像标签置于两个段落之间，那么浏览器会首先显示第一个段落，然后显示图片，最后显示第二段。

8）HTML 表格

表格由 <table> 标签来定义。每个表格均有若干行（由 <tr> 标签定义），每行被分割为若干单元格（由 <td> 标签定义）。字母 td 指表格数据（table data），即数据单元格的内容。数据单元格可以包含文本、图片、列表、段落、表单、水平线、表格

等等。

表格实例

实例

```
<table border="1">
    <tr>
        <td>row 1, cell 1</td>
        <td>row 1, cell 2</td>
    </tr>
    <tr>
        <td>row 2, cell 1</td>
        <td>row 2, cell 2</td>
    </tr>
</table>
```

在浏览器显示如下：：

row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

9）HTML 表单

表单是一个包含表单元素的区域。

表单元素是允许用户在表单中输入内容，比如：文本域（textarea）、下拉列表（select）、单选框（radio-buttons）、复选框（checkbox） 等等。

我们可以使用 <form> 标签来创建表单:

实例

<form>

input 元素

</form>

HTML5 新标签

标签	描述
<form>	定义供用户输入的表单
<input>	定义输入域
<textarea>	定义文本域 (一个多行的输入控件)
<label>	定义了 <input> 元素的标签，一般为输
<fieldset>	定义了一组相关的表单元素，并使用外
<legend>	定义了 <fieldset> 元素的标题
<select>	定义了下拉选项列表
<optgroup>	定义选项组
<option>	定义下拉列表中的选项
<button>	定义一个点击按钮
<datalist> New	指定一个预先定义的输入控件选项列表
<keygen> New	定义了表单的密钥对生成器字段

<output> New	定义一个计算结果

10）HTML 框架

通过使用框架，你可以在同一个浏览器窗口中显示不止一个页面。

iframe 语法:

<iframe src="URL"></iframe>

11）HTML 样式- CSS

内联样式

当特殊的样式需要应用到个别元素时，就可以使用内联样式。使用内联样式的方法是在相关的标签中使用样式属性。样式属性可以包含任何 CSS 属性。以下实例显示出如何改变段落的颜色和左外边距。

这是一个段落。

12）HTML 脚本

HTML <script> 标签

<script> 标签用于定义客户端脚本，比如 JavaScript。

<script> 元素既可包含脚本语句，也可通过 src 属性指向外部脚本文件。

JavaScript 最常用于图片操作、表单验证以及内容动态更新。

下面的脚本会向浏览器输出"Hello World!":

实例

```
<script>
document.write("Hello World!");
</script>
```

3、实训结果

略。

4、撰写实训报告

略。

任务 2　jQuery

1、实训要求

掌握 jQuery 的使用方式

2、操作步骤

jQuery 是一个 JavaScript 库。

jQuery 极大地简化了 JavaScript 编程。

1）jQuery 安装

下载 jQuery

有两个版本的 jQuery 可供下载：

Production version - 用于实际的网站中，已被精简和压缩。

Development version - 用于测试和开发（未压缩，是可读的代码）

以上两个版本都可以从 jquery.com 中下载。jQuery 库是一个 JavaScript 文件，您可以使用 HTML 的 <script> 标签引用它：

```
<head>
<script src="jquery-1.10.2.min.js"></script>
</head>
```

2）jQuery 语法

jQuery 语法是通过选取 HTML 元素，并对选取的元素执行

某些操作。

基础语法： $(selector).action()

美元符号定义 jQuery

选择符（selector）"查询"和"查找" HTML 元素

jQuery 的 action() 执行对元素的操作

实例:

$(this).hide() - 隐藏当前元素

$("p").hide() - 隐藏所有 <p> 元素

$("p.test").hide() - 隐藏所有 class="test" 的 <p> 元素

$("#test").hide() - 隐藏 id="test" 的元素

```
$(document).ready(function(){
    // 开始写 jQuery 代码...
});
```

3）jQuery 选择器

jQuery 选择器允许您对 HTML 元素组或单个元素进行操作。jQuery 选择器基于元素的 id、类、类型、属性、属性值等"查找"(或选择)HTML 元素。它基于已经存在的 CSS 选择器，除此之外，它还有一些自定义的选择器。

jQuery 中所有选择器都以美元符号开头：$()。

元素选择器

jQuery 元素选择器基于元素名选取元素。

在页面中选取所有 <p> 元素:

```
$("p")
```

用户点击按钮后，所有 <p> 元素都隐藏:

实例

```
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide();
  });
});
```

#id 选择器

jQuery #id 选择器通过 HTML 元素的 id 属性选取指定的元素。

页面中元素的 id 应该是唯一的，所以您要在页面中选取唯一的元素需要通过 #id 选择器。

通过 id 选取元素语法如下:

```
$("#test")
```

实例

当用户点击按钮后，有 id="test" 属性的元素将被隐藏:

实例

```
$(document).ready(function(){
  $("button").click(function(){
    $("#test").hide();
```

```
    });
});
```

.class 选择器

jQuery 类选择器可以通过指定的 class 查找元素。

语法如下:

```
$(".test")
```

实例

用户点击按钮后所有带有 class="test" 属性的元素都隐藏:

实例

```
$(document).ready(function(){
    $("button").click(function(){
        $(".test").hide();
    });
});
```

4）jQuery 事件

页面对不同访问者的响应叫做事件。

事件处理程序指的是当 HTML 中发生某些事件时所调用的方法。

实例:

在元素上移动鼠标。

选取单选按钮

点击元素

在事件中经常使用术语"触发"（或"激发"）例如："当您按下按键时触发 keypress 事件"。

常见 DOM 事件：

鼠标事件	键盘事件	表单事件	文档/窗口事件
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload
hover			

jQuery 事件方法语法

在 jQuery 中，大多数 DOM 事件都有一个等效的 jQuery 方法。

页面中指定一个点击事件：

```
$("p").click();
```

下一步是定义了点击后触发事件。您可以通过一个事件函数实现：

```
$("p").click(function(){
    // 动作触发后执行的代码!!
});
```

常用的 jQuery 事件方法

```
$(document).ready()
```

$(document).ready() 方法允许我们在文档完全加载完后执行函数。该事件方法在 jQuery 语法 章节中已经提到过。

click()

click() 方法是当按钮点击事件被触发时会调用一个函数。

该函数在用户点击 HTML 元素时执行。

在下面的实例中，当点击事件在某个 <p> 元素上触发时，隐藏当前的 <p> 元素：

实例

```
$("p").click(function(){
    $(this).hide();
});
```

5）jQuery - AJAX

AJAX 是与服务器交换数据的技术，它在不重载全部页面的情况下，实现了对部分网页的更新。

AJAX = 异步 JavaScript 和 XML（Asynchronous JavaScript and XML）。

简短地说，在不重载整个网页的情况下，AJAX 通过后台加载数据，并在网页上进行显示。

使用 AJAX 的应用程序案例：谷歌地图、腾讯微博、优酷视频、人人网等等。

jQuery $.get() 方法

$.get() 方法通过 HTTP GET 请求从服务器上请求数据。

语法：

$.get(URL,callback);

或

$.get(URL [, data] [, callback] [, dataType])

URL：发送请求的 URL 字符串。

data：可选的，发送给服务器的字符串或 key/value 键值对。

callback：可选的，请求成功后执行的回调函数。

dataType：可选的，从服务器返回的数据类型。默认：智能猜测（可以是 xml, json, script, 或 html）。

下面的例子使用 $.get() 方法从服务器上的一个文件中取回数据：

实例

```
$("button").click(function(){
  $.get("demo_test.php",function(data,status){
    alert("数据: " + data + "\n 状态: " + status);
  });
});
```

jQuery ajax() 方法

定义和用法

ajax() 方法用于执行 AJAX（异步 HTTP）请求。

所有的 jQuery AJAX 方法都使用 ajax() 方法。该方法通常用于其他方法不能完成的请求。

语法

$.ajax({name:value, name:value, ... })

该参数规定 AJAX 请求的一个或多个名称/值对。

名称	值/描述
async	布尔值，表示请求是否异步处理 true。
beforeSend(xhr)	发送请求前运行的函数。
cache	布尔值，表示浏览器是否缓存 面。默认是 true。
complete(xhr,status)	请求完成时运行的函数（在请 败之后均调用，即在 success 数之后）。
contentType	发送数据到服务器时所使用 默认是："application/x-www urlencoded"。
context	为所有 AJAX 相关的回调函 值。
data	规定要发送到服务器的数据
dataFilter(data,type)	用于处理 XMLHttpReques 的函数。

其他方法不能完成的请求。

语法

$.ajax({name:value, name:value, ... })

该参数规定 AJAX 请求的一个或多个名称/值对。

名称	值/描述
async	布尔值，表示请求是否异步处理。默认是 true。
beforeSend(xhr)	发送请求前运行的函数。
cache	布尔值，表示浏览器是否缓存被请求页面。默认是 true。
complete(xhr,status)	请求完成时运行的函数（在请求成功或失败之后均调用，即在 success 和 error 函数之后）。
contentType	发送数据到服务器时所使用的内容类型。默认是："application/x-www-form-urlencoded"。
context	为所有 AJAX 相关的回调函数规定 "this" 值。
data	规定要发送到服务器的数据。
dataFilter(data,type)	用于处理 XMLHttpRequest 原始响应数据的函数。

dataType	预期的服务器响应的数据类型。
error(xhr,status,error)	如果请求失败要运行的函数。
global	布尔值，规定是否为请求触发全局 AJAX 事件处理程序。默认是 true。
ifModified	布尔值，规定是否仅在最后一次请求以来响应发生改变时才请求成功。默认是 false。
jsonp	在一个 jsonp 中重写回调函数的字符串。
jsonpCallback	在一个 jsonp 中规定回调函数的名称。
password	规定在 HTTP 访问认证请求中使用的密码。
processData	布尔值，规定通过请求发送的数据是否转换为查询字符串。默认是 true。
scriptCharset	规定请求的字符集。
success(result,status,xhr)	当请求成功时运行的函数。
timeout	设置本地的请求超时时间（以毫秒计）。
traditional	布尔值，规定是否使用参数序列化的传统样式。
type	规定请求的类型（GET 或 POST）。
url	规定发送请求的 URL。默认是当前页面。

username	规定在 HTTP 访问认证 名。
xhr	用于创建 XMLHttpRe

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title></title>
<script src="https://cdn.staticfile.org/jquery/1.1
</script>
<script>
$(document).ready(function(){
    $("button").click(function(){

    $.ajax({url:"demo_ajax_load.txt",async:true,s
t){
            $("div").html(result);
        }});
    });
});
</script>
```

username	规定在 HTTP 访问认证请求中使用的用户名。
xhr	用于创建 XMLHttpRequest 对象的函数。

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title></title>
<script src="https://cdn.staticfile.org/jquery/1.10.2/jquery.min.js">
</script>
<script>
$(document).ready(function(){
    $("button").click(function(){

    $.ajax({url:"demo_ajax_load.txt",async:true,success:function(resul
t){
            $("div").html(result);
        }});
    });
});
</script>
```

```
</head>

<body>



<div><h2> AJAX 可以修改文本内容</h2></div>

<button>修改内容</button>



</body>

</html>
```

3、实训结果

略。

4、撰写实训报告

略。

HNBC
广州华南商贸职业学院
Guangzhou Huanan Business College

《网络运行与维护》

编者姓名：刘茵、黄晓杰

广州华南商贸职业学院云智信息技术学院

广州腾科网络技术有限公司

《网络运行与维护》
综合实训

编者：刘茵、黄晓杰

广州华南商贸职业学院云智信息技术学院

广州腾科网络技术有限公司

目　录

实验一　以太网交换机基本配置

【实验目的】

掌握以太网交换机基本配置

【实验学时】

建议 2 学时

【实验原理】

一、交换机常用命令配置模式

1 业务描述

(1) Quidway 系列产品的系统命令采用分级保护方式，命令被划分为参观级、监控级、配置级、管理级 4 个级别，简介如下：

- ✧ 参观级：网络诊断工具命令（**ping**、**tracert**）、从本设备出发访问外部设备的命令（包括：Telnet 客户端、RLogin）等，该级别命令不允许进行配置文件保存的操作。
- ✧ 监控级：用于系统维护、业务故障诊断等，包括 **display**、**debugging** 命令，该级别命令不允许进行配置文件保存的操作。
- ✧ 配置级：业务配置命令，包括路由、各个网络层次的命令，这些用于向用户提供直接网络服务。
- ✧ 管理级：关系到系统基本运行，系统支撑模块的命令，这些命令对业务提供支撑作用，包括文件系统、FTP、TFTP、XModem 下载、配置文件切换命令、电源控制命令、备板控制命令、用户管理命令、命令级别设置命令、系统内部参数设置命令等。

（2）命令视图：

系统将命令行接口划分为若干个命令视图，系统的所有命令都注册在某个（或某些）命令视图下，只有在相应的视图下才能执行该视图下的命令：

各命令视图的功能特性、进入各视图的命令等的细则：

◆ 命令视图功能特性列表

命令视图	功能	提示符	进入命令	退出命令
用户视图	查看交换机的简单运行状态和统计信息	<Quidway>	与交换机建立连接即进入	quit 断开与交换机连接
系统视图	配置系统参数	[Quidway]	在用户视图下键入 system-view	quit 返回用户视图

命令视图	功能	提示符	进入命令	退出命令
以太网口视图	配置以太网口参数	[Quidway-Ethernet1/0/0]	在系统视图下键入 interface ethernet 1/0/0	quit 返回系统视图
千兆以太网接口视图	配置千兆以太网接口参数	[Quidway-GigabitEthernet6/1/0]	在系统视图下键入 interface gigabitethernet 6/1/0	quit 返回系统视图
AUX 口视图	配置 AUX 口参数	[Quidway-aux0/0/1]	在系统视图下键入 Interface aux 0/0/1	quit 返回系统视图
Loopback 接口视图	配置 Loopback 接口参数	[Quidway-Loopback2]	在系统视图下键入 interface loopback 2	quit 返回系统视图
用户界面视图	管理交换机异步和逻辑接口	[Quidway-ui0]	在系统视图下键入 user-interface 0	quit 返回系统视图
RIP 协议视图	配置 RIP 协议参数	[Quidway-rip]	在系统视图下键入 rip	quit 返回系统视图
OSPF 协议视图	配置 OSPF 协议参数	[Quidway-ospf]	在系统视图下键入 ospf	quit 返回系统视图

2 配置参考

（1）命令行在线帮助

在任一命令视图下，键入“?”获取该命令视图下所有的命令及其简单描述。

<Quidway> ?

键入一命令，后接以空格分隔的“?”，如果该位置为关键字，则列出全部关键字及其简单描述。

<Quidway> display ?

键入一命令，后接以空格分隔的“?”，如果该位置为参数，则列出有关的参数描述。

```
[Quidway] interface ethernet ?
  <3-3>  Slot number
[Quidway] interface ethernet 3?
   /
[Quidway] interface ethernet 3/?
  <0-0>
[Quidway] interface ethernet 3/0?
   /
[Quidway] interface ethernet 3/0/?
```

```
  <0-0>
[Quidway] interface ethernet 3/0/0 ?
  <cr>
```

> 说明：
>
> 其中<cr>表示该位置无参数，在紧接着的下一个命令行该命令被复述，直接键入回车即可执行。

键入一字符串，其后紧接“?”，列出以该字符串开头的所有命令。

```
<Quidway> d?
debugging   delete   dir   display
```

键入一命令，后接一字符串紧接“?”，列出命令以该字符串开头的所有关键字。

```
<Quidway> display h?
history-command
```

> 说明：
>
> 输入命令的某个关键字的前几个字母，按下<tab>键，可以显示出完整的关键字，前提是这几个字母可以唯一标示出该关键字，不会与这个命令的其它关键字混淆。
>
> 以上帮助信息，均可通过执行 language-mode chinese 命令切换为中文显示。

（2）命令行错误信息

用户键入的所有命令，如果通过语法检查，则执行，否则向用户报告错误信息：

英文错误信息	错误原因
Unrecognized command	没有查找到命令
	没有查找到关键字
	参数类型错
	参数值越界
Incomplete command	输入命令不完整
Too many parameters	输入参数太多
Ambiguous command	输入参数不明确

（3）历史命令

命令行接口提供类似 Doskey 功能，将用户键入的历史命令自动保存，用户可以随时调

用命令行接口保存的历史命令，并重复执行。在缺省状态下，命令行接口为每个用户最多可以保存 10 条历史命令：

操作	按键	结果
显示历史命令	**display history-command**	显示用户键入的历史命令
访问上一条历史命令	上光标键或者<Ctrl+P>	如果还有更早的历史命令，则取出上一条历史命令，否则响铃警告。
访问下一条历史命令	下光标键或者<Ctrl+N>	如果还有更晚的历史命令，则取出下一条历史命令，否则清空命令，响铃警告。

（4）显示特性

命令行接口提供了如下的显示特性

为方便用户，提示信息和帮助信息可以用中英文两种语言显示。

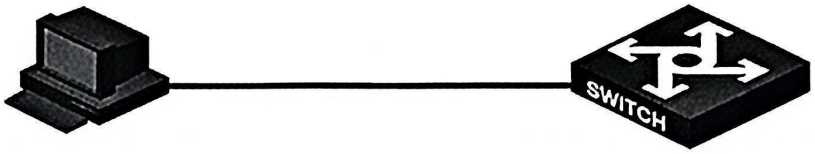
在一次显示信息超过一屏时，提供了暂停功能，这时用户可以有三种选择:

按键或命令	功能
暂停显示时键入<Ctrl+C>	停止显示和命令执行
暂停显示时键入空格键	继续显示下一屏信息
暂停显示时键入回车键	继续显示下一行信息

基本配置命令

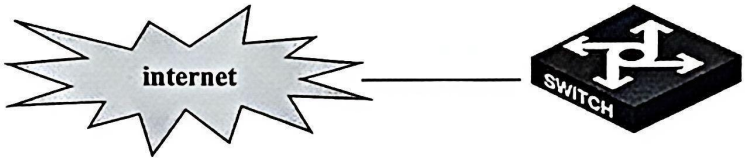
操作	命令
从用户视图进入系统视图	**system-view**
从系统视图返回到用户视图	**quit**
从任意的非用户视图返回到用户视图	**return**
设置交换机名	**sysname** *sysname*
显示系统版本	**display version [** *slot-id* **]**
显示起始配置信息	**display saved-configuration**
显示当前配置信息	**display current-configuration**
显示设备基本信息	**display device [pic-status |** *slot-id* **]**

二、通过 console 口配置



建立本地配置环境，只需将微机（或终端）的串口通过专用配置电缆与交换机主用 MPU 板上的配置口（Console）连接。

三、通过 telnet 方式配置



【实验步骤】

1 通过 console 口配置

（1） 在 PC 机 Windows95，Windows98，Windows NT 等操作系统中，打开超级终端。设置终端通信参数为 9600bit/s、8 位数据位、1 位停止位、无校验和无流控，选择终端类型为 VT100 或自动检测，如图：

（2）交换机上电运行，如果不需要进入 BOOTROM 菜单，则自检结束后提示用户键入回车，直到出现命令行提示符<Quidway>

2 通过 telnet 方式配置

配置交换机的 ip 地址和 PC 的 ip 地址

[Quidway]vlan 10
[Quidway-vlan10]port access ethernet 0/1 to ethernet 0/5
[Quidway-vlan10]quit
[Quidway]interface vlan 10
[Quidway-vlan10-interface]ip address 1.1.1.4 255.0.0.0
配置完交换机的 ip 地址，你还需要配置 PC 的 ip 地址（比如 1.1.1.2/8）
配置 Telnet 方式登录时的密码：
[Quidway] User-interface vty 0 4
[Quidway-ui-vty0] authentication-mode password
[Quidway-ui-vty0] set authentication password simple Huawei
[Quidway-ui-vty0] user privilege level 3
检测 PC 与交换机的连通性，使用 ping 命令检测，能否 ping 通交换机

【实验报告】

实验二　以太网端口配置实验

【实验目的】

1. 掌握以太网端口的基本配置
2. 掌握以太网端口的端口汇聚
3. 掌握以太网端口的端口镜像

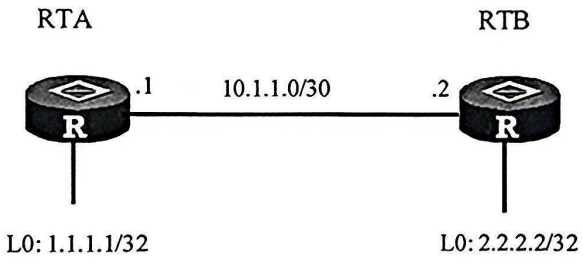
【实验学时】

建议 2 学时

【实验原理】

一、 以太网端口基本配置

1 组网及业务介绍

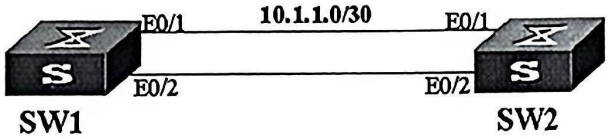


2 命令行列表

操作	命令
进入指定以太网接口的视图	**interface Ethernet** *interface-number*
进入千兆以太网接口的视图	**interface gigabitethernet** *interface-number*
配置 IP 地址	**ip address** *ip-address ip-mask* **[sub]**
配置静态 ARP 映射项	**arp static** *ip-address mac-address*
配置 MTU	**mtu** *ethernet-mtu*
选择 FE 电接口的工作速率	**speed { 10 | 100 | negotiation }**
选择 GE 电接口的工作速率	**speed { 10 | 100 | 1000 | negotiation }**
选择以太网接口的工作模式	**duplex { full | half | negotiation }**
允许对内自环	**Loopback**

二、以太网端口汇聚

1 组网及业务介绍



2 命令行列表

操作	命令
将一组端口设置为汇聚端口	**Link-aggregation** *interface_name1* **to** *interface_name2* **[both]**

【实验步骤】

一 以太网端口基本配置

1 配置步骤

配置 RTA 和 RTB 的接口 IP 地址。其余配置使用缺省值。

2 结果验证

（1）配置前的端口状态；

[RTA]display interface Ethernet 0/0
Ethernet0/0 current state :UP
Line protocol current state :DOWN
Description : Ethernet0/0 Interface
The Maximum Transmit Unit is 1500, Hold timer is 10(sec)
Internet protocol processing : disabled
IP Sending Frames' Format is PKTFMT_ETHNT_2, Hardware address is 00e0-fc17-a1fb
Media type is twisted pair, loopback not set, promiscuous mode not set
100Mb/s, Full-duplex, link type is autonegotiation
Output flow-control is disabled, input flow-control is disabled
Output queue : (Urgent queue : Size/Length/Discards) 0/50/0
Output queue : (Protocol queue : Size/Length/Discards) 0/500/0
Output queue : (FIFO queuing : Size/Length/Discards) 0/75/0
Last 300 seconds input rate 0.00 bytes/sec, 0.00 packets/sec
Last 300 seconds output rate 0.00 bytes/sec, 0.00 packets/sec
Input: 6 packets, 550 bytes, 6 buffers
0 broadcasts, 0 multicasts, 0 pauses
0 errors, 0 runts, 0 giants

0 crc, 0 align errors, 0 overruns
0 dribbles, 0 drops, 0 no buffers
Output:5 packets, 532 bytes, 6 buffers
1 broadcasts, 0 multicasts, 0 pauses
0 errors, 0 underruns, 0 collisions
0 deferred, 0 lost carriers

（2）配置后

[RTA-Ethernet0/0]display interface Ethernet 0/0
Ethernet0/0 current state :UP
Line protocol current state :UP
Description : Ethernet0/0 Interface
The Maximum Transmit Unit is 1500, Hold timer is 10(sec)
Internet Address is 10.1.1.1/30
IP Sending Frames' Format is PKTFMT_ETHNT_2, Hardware address is 00e0-fc17-a1fb
Media type is twisted pair, loopback not set, promiscuous mode not set
100Mb/s, Full-duplex, link type is autonegotiation
Output flow-control is disabled, input flow-control is disabled
Output queue : (Urgent queue : Size/Length/Discards) 0/50/0
Output queue : (Protocol queue : Size/Length/Discards) 0/500/0
Output queue : (FIFO queuing : Size/Length/Discards) 0/75/0
Last 300 seconds input rate 0.00 bytes/sec, 0.00 packets/sec
Last 300 seconds output rate 0.00 bytes/sec, 0.00 packets/sec
Input: 6 packets, 550 bytes, 6 buffers
0 broadcasts, 0 multicasts, 0 pauses
0 errors, 0 runts, 0 giants
0 crc, 0 align errors, 0 overruns
0 dribbles, 0 drops, 0 no buffers
Output:5 packets, 532 bytes, 6 buffers
1 broadcasts, 0 multicasts, 0 pauses
0 errors, 0 underruns, 0 collisions
0 deferred, 0 lost carriers

（3）连通性测试

[RTA-Ethernet0/0]ping 10.1.1.2
PING 10.1.1.2: 56 data bytes, press CTRL_C to break
Reply from 10.1.1.2: bytes=56 Sequence=1 ttl=255 time=2 ms
Reply from 10.1.1.2: bytes=56 Sequence=2 ttl=255 time=1 ms
Reply from 10.1.1.2: bytes=56 Sequence=3 ttl=255 time=1 ms
Reply from 10.1.1.2: bytes=56 Sequence=4 ttl=255 time=1 ms
Reply from 10.1.1.2: bytes=56 Sequence=5 ttl=255 time=1 ms

--- 10.1.1.2 ping statistics ---
5 packet(s) transmitted

5 packet(s) received
0.00% packet loss
round-trip min/avg/max = 1/1/2 ms

3 配置参考

路由器 A 的配置：

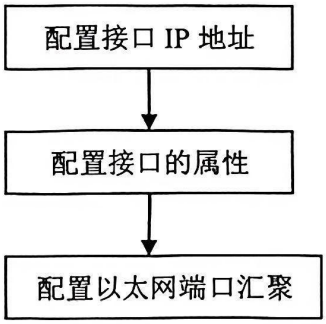
```
[RTA]interface Ethernet 0/0
[RTA-Ethernet0/0]ip address 10.1.1.1 30
```

路由器 B 的配置：

```
[RTB]interface Ethernet 0/0
[RTB-Ethernet0/0]ip address 10.1.1.2 30
```

二、 以太网端口的端口汇聚

1 配置流程图



2 配置步骤

（1）配置接口 IP 地址

在 SW1 和 SW2 上分别创建 VLAN1 的三层地址 10.1.1.1/30 和 10.1.1.2/30。

（2）配置汇聚的端口属性

在配置端口汇聚之前，首先要保证 Sw1 和 Sw2 所有汇聚的端口必须工作在全双工方式下，而且必须工作在相同的速率下（不能工作于自协商模式）。

（3）配置端口汇聚

3 结果验证

在配置完 IP 地址后，用 Ping 检查两台交换机之间的互通性

显然这个组网图会导致交换机之间的环路的存在，主要表现就是出现广播风暴，交换机数据转发灯不停的闪烁。广播风暴可能会导致连接在交换机网口上的 Pc 反应速度变慢，如果出现这种情况，可以先拔掉一根网线消除环路，待配置好端口汇聚以后再将其插上，此时广播风暴就不会发生了。

（1）检查配置的正确性

```
<Sw1>display link-aggregation
```

```
Master port: Ethernet0/1
Other sub-ports:
     Ethernet0/2
Mode: both
```

（2）验证端口汇聚的互为备份这个特性。

首先在 Sw1 上用 Display Mac 命令检查 MAC 地址表，查看目前 Sw2 的 Interface Vlan 1 的 MAC 地址在 E0/1 还是 E0/2，Sw2 的 Interface Vlan 1 的 MAC 地址可以通过在 Sw2 上利用命令 Display Interface Vlan 1 观察到。

```
<Sw1>display mac
MAC ADDR        VLAN ID    STATE           PORT INDEX            AGING TIME
00e0-fc26-2f3c      1      Learned         Ethernet0/2            AGING
  ---  1 mac address(es) found  ---
<Sw2>display interface vlan
Vlan-interface1 current state : UP
Line protocol current state : UP
IP Sending Frames' Format is PKTFMT_ETHNT_2, Hardware address is 00e0-fc26-2f3c
Internet Address is 10.1.1.2/30 Primary
Description : HUAWEI, Quidway Series, Vlan-interface1 Interface
The Maximum Transmit Unit is 1500
```

由上述输出可以知道，目前 Sw1 将要转发到 Sw2 的 Interface Vlan1 数据帧是通过 E0/2 转发的。为了检验汇聚组中 E0/1 是否能够备份 E0/2，我们只要手工关闭 E0/1 在检查 Sw1 和 Sw2 之间的可通性即可。

```
[Sw1]interface Ethernet 0/2
[Sw1-Ethernet0/2]shut
%Oct 10 15:01:57 2005 Sw1 L2INF/5/PORT LINK STATUS CHANGE:
 Ethernet0/2: turns into DOWN state
<Sw2>ping 10.1.1.1
  PING 10.1.1.1: 56  data bytes, press CTRL_C to break
    Reply from 10.1.1.1: bytes=56 Sequence=1 ttl=254 time = 13 ms
    Reply from 10.1.1.1: bytes=56 Sequence=2 ttl=254 time = 10 ms
    Reply from 10.1.1.1: bytes=56 Sequence=3 ttl=254 time = 8 ms
    Reply from 10.1.1.1: bytes=56 Sequence=4 ttl=254 time = 9 ms
    Reply from 10.1.1.1: bytes=56 Sequence=5 ttl=254 time = 10 ms
  --- 10.1.1.1 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 8/10/13 ms
<Sw1>display mac
MAC ADDR        VLAN ID    STATE           PORT INDEX            AGING TIME
00e0-fc26-2f3c      1      Learned         Ethernet0/1            AGING
00e0-fc09-bcf9      1      Learned         Ethernet0/1            AGING
<Sw2>display interface Vlan-interface
             Vlan-interface1 current state : UP
```

Line protocol current state : UP
IP Sending Frames' Format is PKTFMT_ETHNT_2, Hardware address is 00e0-fc26-2f3c
Internet Address is 10.1.1.2/30 Primary
Description : HUAWEI, Quidway Series, Vlan-interface1 Interface
The Maximum Transmit Unit is 1500

4 参考配置

交换机 Sw1 的配置：

```
[Sw1]interface Vlan-interface 1
[Sw1-Vlan-interface1]ip add 10.1.1.1 255.255.255.252
[Sw1-Ethernet0/1]speed 100
[Sw1-Ethernet0/1]duplex full
[Sw1-Ethernet0/2]speed 100
[Sw1-Ethernet0/2]duplex full
[Sw1]link-aggregation Ethernet 0/1 to Ethernet 0/2 both
```

交换机 Sw2 的配置：

```
[Sw2]interface Vlan-interface 1
[Sw2-Vlan-interface1]ip add 10.1.1.2 255.255.255.252
[Sw2-Ethernet0/1]speed 100
[Sw2-Ethernet0/1]duplex full
[Sw2-Ethernet0/2]speed 100
[Sw2-Ethernet0/2]duplex full
[Sw2]link-aggregation Ethernet 0/1 to Ethernet 0/2 both
```

备注：如果用 S3900 系列的交换机做 port link aggregation 命令如下：

```
[Sw1]link-aggregation group 10 mode manual
[Sw1-Ethernet1/0/1]speed 100
[Sw1-Ethernet1/0/1]duplex full
[Sw1-Ethernet1/0/1]port link-aggregation group 10
[Sw1-Ethernet1/0/2]speed 100
[Sw1-Ethernet1/0/2]duplex full
[Sw1-Ethernet1/0/2]port link-aggregation group 10

[Sw2]link-aggregation group 10 mode manual
[Sw2-Ethernet1/0/1]speed 100
[Sw2-Ethernet1/0/1]duplex full
[Sw2-Ethernet1/0/1]port link-aggregation group 10
[Sw2-Ethernet1/0/2]speed 100
[Sw2-Ethernet1/0/2]duplex full
[Sw2-Ethernet1/0/2]port link-aggregation group 10
```

【实验报告】

实验三 利用 TFTP 管理交换机配置

【实验目的】

掌握 TFTP 配置

【实验学时】

建议 2 学时

【实验原理】

一、TFTP 配置

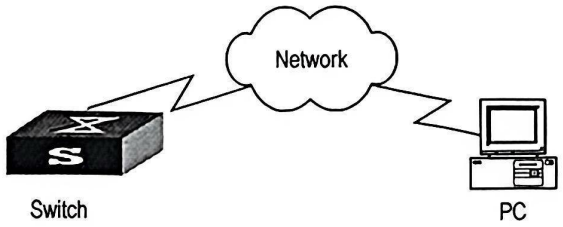
1 TFTP 简介

TFTP（Trivial File Transfer Protocol）是一种简单文件传输协议。相对于另一种文件传输协议 FTP，TFTP 不具有复杂的交互存取接口和认证控制，适用于客户端和服务器之间不需要复杂交互的环境。TFTP 协议一般在 UDP 的基础上实现。

TFTP 协议传输是由客户端发起的。当需要下载文件时，由客户端向 TFTP 服务器发送读请求包，然后从服务器接收数据，并向服务器发送确认；当需要上传文件时，由客户端向 TFTP 服务器发送写请求包，然后向服务器发送数据，并接收服务器的确认。TFTP 传输文件有两种模式：一种是二进制模式，用于传输程序文件；另一种是 ASCII 码模式，用于传输文本文件。

配置 TFTP 之前，网络管理员需要首先配置好 TFTP 客户端和服务器的 IP 地址，并且确保客户端和服务器之间路由可达。

交换机只能作为 TFTP 客户端。



交换机作为 TFTP Client 时的配置

Switch	配置交换机 VLAN 接口的 IP 地址，使其和 TFTP Server 的 IP 地址在同一网段	-	TFTP 适用于客户端和服务器之间不需要复杂交互的环境，请保证交换机 VLAN 接口和 TFTP Server 之间路由可达。
	可以直接使用 TFTP 命令登录远端的 TFTP Server 上传或者下载文件	-	-
PC	启动 TFTP Server，并作了 TFTP 工作目录的配置	-	—

【实验步骤】

一、TFTP 配置

1 用 TFTP 下载文件

请在用户视图下进行下列配置。

用 TFTP 获取文件	**tftp** *tftp-server* **get** *source-file* [*dest-file*]

配置命令中，tftp-server 参数代表 TFTP 服务器的 IP 地址或者主机名；source-file 指的是要下载的 TFTP 服务器上的文件信息；dest-file 参数代表下载后存储在交换机上的文件名。

2 用 TFTP 上传文件

请在用户视图下进行下列配置。

用 TFTP 保存文件	**tftp** *tftp-server* **put** *source-file* [*dest-file*]

配置命令中，source-file 参数代表要上传到服务器的文件。dest-file 指的是文件上传到 TFTP 服务器的存储目录；tftp-server 参数代表 TFTP 服务器的 IP 地址或者主机名。

【实验报告】

实验四 虚拟局域网 VLAN

【实验目的】

1. 掌握 VLAN 的基本配置
2. 掌握端口的三种类型
3. 掌握 GVRP 协议动态创建和注册 VLAN 信息
4. 掌握 isolate-user-vlan 的配置

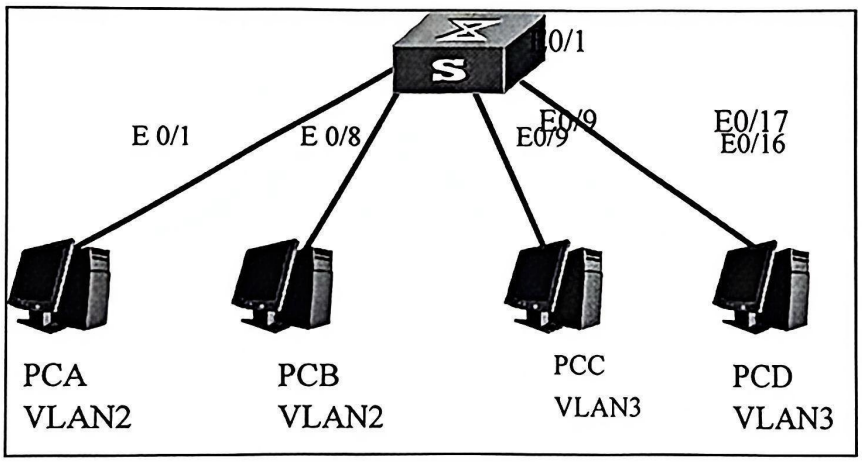
【实验学时】

建议 3 学时

【实验原理】

一、VLAN 的基本配置

1 组网及业务描述



本实验的主要目的是掌握 VLAN 的基本配置。在同一交换机内，要求能够达到同一 VLAN 内的 PC 可以互通，不同 VLAN 间的 PC 不能互通。

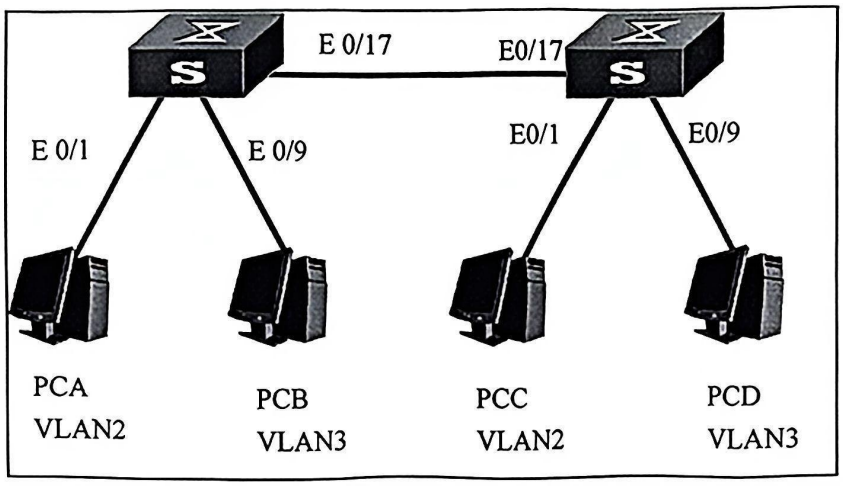
2 命令行列表

操　作	命　令
创建 VLAN	**vlan** *vlan-id* [**alias** *vlan-alias*]
删除 VLAN	**undo vlan** *vlan-id* [**[all]**
VLAN 视图下配置一个或一组端口属于某个 VLAN	**port** interface-type { interface-num [**to** interface-num] } & <1-10>

接口视图下配置该端口属于某个 VLAN	**port access vlan** *vlan-id*

二、Trunk 的基本配置

1 组网及业务描述

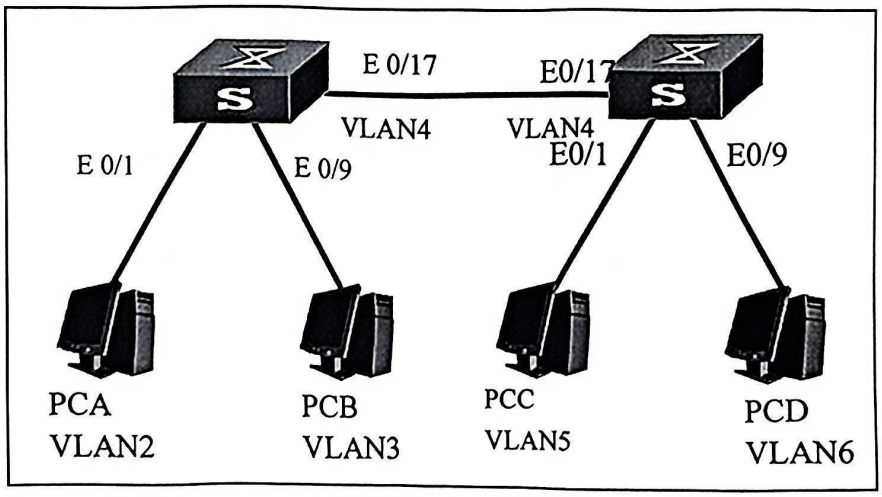


2 命令行列表

操 作	命 令
指定端口类型：trunk, access, hybrid	**port** *link-type* { *trunk/access/hybrid}*
取消端口类型的设置	**undo port** *link-type* { *trunk/access/hybrid}*
设置 Trunk 端口可以通过的 VLAN	**[undo] port trunk permit vlan** { {*vlan-id* [**to** *vlan-id*]}&<1-10> | **all** }
显示 VLAN 的信息	display vlan vlan-id [/all]

三、VLAN 间的三层互通

1 组网及业务描述



PCA,PCB,PCC.PCD 属于不同的 vlan，在 SwithcA 上配置去 vlan5 和 vlan6 的 静态路由，在 SwithcB 上配置去 vlan2 和 vlan3 的 静态路由。

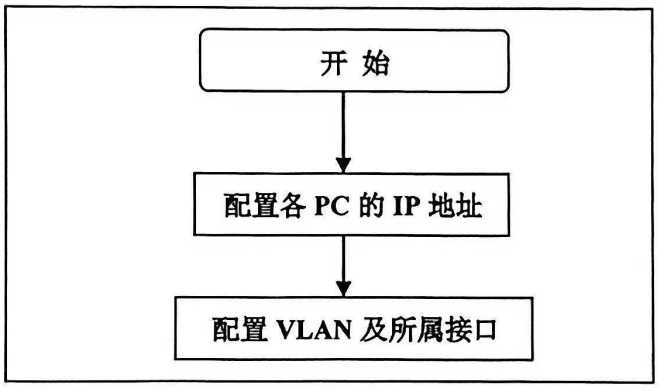
2 命令行列表

操 作	命 令
进入 vlan 三层虚接口视图	interface vlan-interface *vlan-id*
配置静态路由	ip route-static *<ip_address> [<mask>* | <masklen>] interface_name> | <gateway_address> [preference <preference_value>] [reject |backhole]
显示路由信息	display ip routing-table

【实验步骤】

一、VLAN 的基本配置

1 配置流程图



2 配置步骤

（1）配置各 PC 的 IP 地址

按照上图连接各实验设备，配置 PCA IP 地址为 10.1.1.2/24，PCB IP 地址为 10.1.1.3/24，PCC IP 地址为 10.1.2.2/24，PCD IP 地址为 10.1.2.3/24。

（2）配置 VLAN 及所属端口

创建两个 VLAN：VLAN 2 和 VLAN 3，配置端口 Ethernet 0/1 到 Ethernet0/8 属于 VLAN2，端口 Ethernet0/9 到 Ethernet0/16 属于 VLAN3。

3 结果验证

同一 VLAN 内部的 PC 可以互相访问，在 PCA 上 ping PCB，结果如下：

C:\Documents and Settings\x99270>ping 10.1.1.3

Pinging 10.1.1.3 with 32 bytes of data:

Reply from 10.1.1.3: bytes=32 time<1ms TTL=128
Reply from 10.1.1.3: bytes=32 time<1ms TTL=128
Reply from 10.1.1.3: bytes=32 time<1ms TTL=128
Reply from 10.1.1.3: bytes=32 time<1ms TTL=128

Ping statistics for 10.1.1.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PCC 与 PCD 属于同一个 VLAN 3，在 PCC 上 ping PCD，可以 ping 通，结果如下：
C:\Documents and Settings\x99270>ping 10.1.2.3

Pinging 10.1.2.3 with 32 bytes of data:

Reply from 10.1.2.3: bytes=32 time<1ms TTL=128
Reply from 10.1.2.3: bytes=32 time<1ms TTL=128
Reply from 10.1.2.3: bytes=32 time<1ms TTL=128
Reply from 10.1.2.3: bytes=32 time<1ms TTL=128

Ping statistics for 10.1.2.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

不同 VLAN 间的 PC 不能够互相访问。在 PCA 上 ping PCC，不能 ping 通，结果显示如下：
C:\Documents and Settings\x99270>ping 10.1.2.3

Pinging 10.1.2.3 with 32 bytes of data:

Destination host unreachable.
Destination host unreachable.
Destination host unreachable.
Destination host unreachable.

Ping statistics for 10.1.2.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

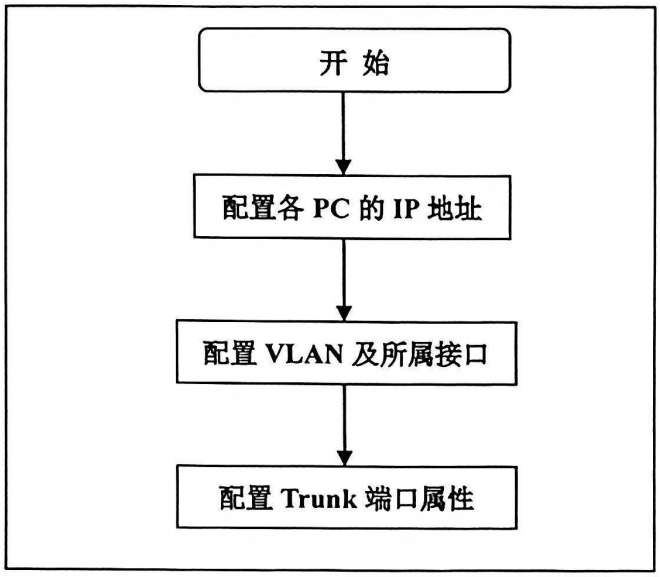
4 配置参考

在交换机上做如下配置：
<Quidway>system-view
Enter system view , return user view with Ctrl+Z.

[Quidway]sysname Switch
[Switch]vlan 2
[Switch-vlan2]port Ethernet 0/1 to Ethernet 0/8
[Switch]vlan 3
[Switch-vlan3]port Ethernet 0/9 to Ethernet 0/16

二、Trunk 的基本配置

1 配置流程图



2 配置步骤

配置各 PC 的 IP 地址

首先按照上图连接各实验设备，然后配置 PCA IP 地址为 10.1.1.2/24，PCB IP 地址为 10.1.2.2/24，PCC IP 地址为 10.1.1.3/24，PCD IP 地址为 10.1.2.3/24。

配置 VLAN 及所属端口

创建两个 VLAN：VLAN 2 和 VLAN 3，配置端口 Ethernet 0/1 到 Ethernet0/8 属于 VLAN2，端口 Ethernet0/9 到 Ethernet0/16 属于 VLAN3。

配置交换机之间的端口为 Trunk 端口，并且允许所能通过的 VLAN

指定端口 Ethernet0/17 为 Trunk 端口，并允许所有 VLAN 可以通过。

3 结果验证

配置完成后，可以看到，同一 VLAN 内部的 PC 可以互相访问，在 PCA 上 ping PCC，结果如下：

```
C:\Documents and Settings\x99270>ping 10.1.1.3

Pinging 10.1.1.3 with 32 bytes of data:

Reply from 10.1.1.3: bytes=32 time<1ms TTL=128
Reply from 10.1.1.3: bytes=32 time<1ms TTL=128
Reply from 10.1.1.3: bytes=32 time<1ms TTL=128
```

```
Reply from 10.1.1.3: bytes=32 time<1ms TTL=128

Ping statistics for 10.1.1.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

同样，在 PCB 上 ping PCD 也可以 ping 通。

不同 VLAN 间的 PC 不能够互相访问，在 PCA 上 ping PCD，得到的结果如下：

```
C:\Documents and Settings\x99270>ping 10.1.2.3

Pinging 10.1.2.3 with 32 bytes of data:

Destination host unreachable.
Destination host unreachable.
Destination host unreachable.
Destination host unreachable.

Ping statistics for 10.1.2.3:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

4 配置参考

（1）配置 VLAN 及所属端口

配置 RTA：

```
[Switch A]vlan 2
[Switch A-vlan2]port ethernet 0/1 to ethernet 0/8
[Switch A]vlan 3
[Switch A-vlan3]port Ethernet 0/9 to Ethernet 0/16
```

配置 RTB：

```
[Switch B]vlan 2
[Switch B-vlan2]port Ethernet 0/1 to Ethernet 0/8
[Switch B]vlan 3
[Switch B-vlan3]port Ethernet 0/9 to Ethernet 0/16
```

（2）配置 Trunk 端口

配置 RTA：

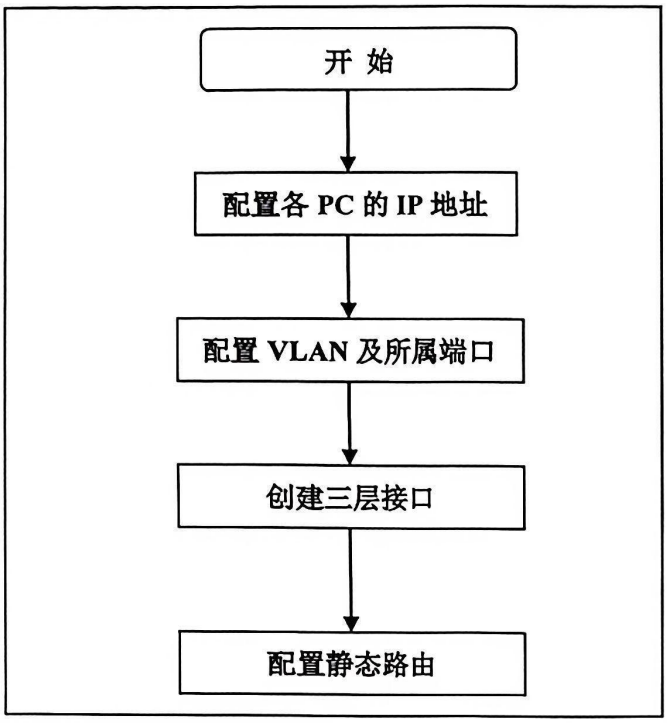
```
[Switch A] interface Ethernet 0/17
[Switch A-Ethernet0/17] port link-type trunk
[Switch A-Ethernet0/17] port trunk permit vlan all //允许所有 VLAN 通过 Trunk 端口
```

配置 RTB：

```
[Switch B] interface Ethernet 0/17
[Switch B-Ethernet0/17] port link-type trunk
[Switch B-Ethernet0/17] port trunk permit vlan all //允许所有 VLAN 通过 Trunk 端口
```

三、VLAN 间的三层互通

1 配置流程图



2 配置步骤

（1）配置各 PC 的 IP 地址及网关地址

首先按照上图连接各实验设备，然后配置 PCA IP 地址为 10.1.2.2/24，网关地址为 10.1.2.1/24，PCB IP 地址为 10.1.3.2/24，网关地址为 10.1.3.1/24，PCC IP 地址为 10.1.5.2/24，网关地址为 10.1.5.1/24，PCD IP 地址为 10.1.6.2/24，网关地址为 10.1.6.1/24。

（2）配置 VLAN 及所属端口

在交换机 A 上创建三个 VLAN：VLAN 2、VLAN 3 和 VLAN 4，配置端口 Ethernet 0/1 到端口 Ethernet0/8 属于 VLAN 2，端口 Ethernet0/9 到 Ethernet0/16 属于 VLAN 3，端口 Ethernet0/17 属于 VLAN 4。

在交换机 B 上创建三个 VLAN：VLAN 5、VLAN 6 和 VLAN 4，配置端口 Ethernet 0/1 到端口 Ethernet0/8 属于 VLAN 5，端口 Ethernet0/9 到 Ethernet0/16 属于 VLAN 6，端口 Ethernet0/17 属于 VLAN 4。

（3）创建三层接口，并配置 IP 地址

在 Switch A 上创建三个三层虚接口，并配置 IP 地址

VLAN2 的 IP 地址是 10.1.2.1，掩码是 255.255.255.0

VLAN3 的 IP 地址是 10.1.3.1，掩码是 255.255.255.0

VLAN4 的 IP 地址是 10.1.4.1，掩码是 255.255.255.0

在 Switch B 上创建三个三层虚接口，并配置 IP 地址

VLAN5 的 IP 地址是 10.1.5.1，掩码是 255.255.255.0

VLAN6 的 IP 地址是 10.1.6.1，掩码是 255.255.255.0

VLAN4 的 IP 地址是 10.1.4.2，掩码是 255.255.255.0

在交换机上配置非直连网段静态路由

在 Switch A 上配置两静态路由：

```
ip route-static 10.1.5.0 255.255.255.0 10.1.4.2
ip route-static 10.1.6.0 255.255.255.0 10.1.4.2
```

在 Switch B 上配置两静态路由：

```
ip route-static 10.1.2.0 255.255.255.0 10.1.4.1
ip route-static 10.1.3.0 255.255.255.0 10.1.4.1
```

3 结果验证

通过 display ip routing-table 查看路由表

```
<Switch A>display ip routing-table
 Routing Table: public net
```

Destination/Mask	Protocol	Pre	Cost	Nexthop	Interface
10.1.2.0/24	DIRECT	0	0	10.1.2.1	Vlan-interface2
10.1.2.1/32	DIRECT	0	0	127.0.0.1	InLoopBack0
10.1.3.0/24	DIRECT	0	0	10.1.3.1	Vlan-interface3
10.1.3.1/32	DIRECT	0	0	127.0.0.1	InLoopBack0
10.1.4.0/24	DIRECT	0	0	10.1.4.1	Vlan-interface4
10.1.4.1/32	DIRECT	0	0	127.0.0.1	InLoopBack0
10.1.5.0/24	STATIC	60	0	10.1.4.2	Vlan-interface4
10.1.6.0/24	STATIC	60	0	10.1.4.2	Vlan-interface4
127.0.0.0/8	DIRECT	0	0	127.0.0.1	InLoopBack0
127.0.0.1/32	DIRECT	0	0	127.0.0.1	InLoopBack0

在 PCA 上 ping PCC，可以得到：

```
C:\Documents and Settings\x99270>ping 10.1.5.2

Pinging 10.1.5.2 with 32 bytes of data:

Reply from 10.1.5.2: bytes=32 time<1ms TTL=126
Reply from 10.1.5.2: bytes=32 time<1ms TTL=126
Reply from 10.1.5.2: bytes=32 time<1ms TTL=126
Reply from 10.1.5.2: bytes=32 time<1ms TTL=126

Ping statistics for 10.1.5.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

同样，在 PCA 上可以 ping 通 PCC、PCD。全网可互通。

4 配置参考

（1）配置 VLAN 及所属端口

配置 RTA：

```
[Switch A]vlan 2
[Switch A-vlan2]port Ethernet 0/1 to ethernet0/8
```

```
[Switch A]vlan 3
[Switch A-vlan3]port Ethernet 0/9 to Ethernet 0/16
[Switch A]vlan 4
[Switch A-vlan4]port Ethernet 0/17
```

配置 RTB：

```
[Switch B]vlan 5
[Switch B-vlan5]port Ethernet 0/1 to ethernet0/8
[Switch B]vlan 6
[Switch B-vlan6]port Ethernet 0/9 to Ethernet 0/16
[Switch B]vlan 4
[Switch B-vlan4]port Ethernet 0/17
```

（2）创建三层接口

配置 RTA：

```
[Switch A]interface Vlan-interface 2
[Switch A-Vlan-interface2]ip address 10.1.2.1 255.255.255.0
[Switch A]interface vlan-interface 3
[Switch A-Vlan-interface3]ip address 10.1.3.1 255.255.255.0
[Switch A]interface vlan-interface 4
[Switch A-Vlan-interface4]ip address 10.1.4.1 255.255.255.0
```

配置 RTB：

```
[Switch B]interface Vlan-interface 5
[Switch B-Vlan-interface5]ip address 10.1.5.1 255.255.255.0
[Switch B]interface vlan-interface 6
[Switch B-Vlan-interface6]ip address 10.1.6.1 255.255.255.0
[Switch B]interface vlan-interface 4
[Switch B-Vlan-interface4]ip address 10.1.4.2 255.255.255.0
```

（3）配置静态路由

配置 RTA：

```
[Switch A]ip route-static 10.1.5.0 255.255.255.0 10.1.4.2
[Switch A]ip route-static 10.1.6.0 255.255.255.0 10.1.4.2
```

配置 RTB：

```
[Switch B]ip route-static 10.1.2.0 255.255.255.0 10.1.4.1
[Switch B]ip route-static 10.1.3.0 255.255.255.0 10.1.4.1
```

【实验报告】

HNBC
广州华南商贸职业学院
Guangzhou Huanan Business College

《网络管理与维护综合实训》

编者姓名：黄英就、欧阳科

广州华南商贸职业学院云智信息技术学院
广州粤嵌通信科技股份有限公司

《网络管理与维护综合实训》

编者 黄英就、欧阳科

广州华南商贸职业学院云智信息学院

广州粤嵌通信科技股份有限公司

目　录

模块一：华为 VRP 系统基本操作实训

实训目标：

本项目通过配置华为设备，了解并熟悉华为 VRP 系统的基本操作。

实训技能：

理解命令行视图的含义以及进入离开命令行视图的方法

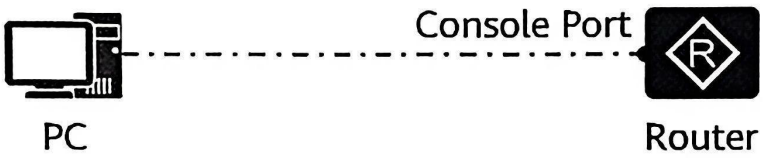
掌握一些常见的命令

掌握使用命令行在线帮助的方法

掌握如何撤销命令

掌握如何使用命令行快捷键

实训内容：



熟悉 VRP 操作系统项目拓扑

1. 实训要求

如组网图所示，Router 是一台全新无配置的路由器，PC 通过串口线缆连接到 Router 的 Console Port，需要对 Router 进行一些初始化操作。

1. 完成设备命名、路由器接口 IP 地址等基础配置
2. 保存设备配置
3. 重启设备

2. 操作步骤

通过 Console 方式登录到 Router 的 CLI

略。

查看设备基本信息

查看设备版本信息

```
<Huawei>display version
Huawei Versatile Routing Platform Software
VRP (R) software, Version 5.160 (AR651C V300R019C00SPC100)
Copyright (C) 2011-2016 HUAWEI TECH CO., LTD
Huawei AR651C Router uptime is 0 week, 0 day, 0 hour, 53 minutes
BKP 0 version information:
1. PCB      Version  : AR01BAK2C VER.B
2. If Supporting PoE : No
3. Board    Type     : AR651C
```

```
4. MPU Slot Quantity : 1
5. LPU Slot Quantity : 1
```

完成设备基本配置

修改 Router 的名字为 Datacom-Router

```
<Huawei>system-view
Enter system view, return user view with Ctrl+Z.
[Huawei]
```

此时设备已经从用户视图进入到了系统视图。

```
 [Huawei]sysname Datacom-Router
[Datacom-Router]
```

此时设备名称已经修改为 Datacom-router。

华为设备提供丰富的功能，相应的也提供了多样的配置和查询命令。为便于用户使用这些命令，华为设备按功能分类将命令分别注册在不同的命令行视图下。配置某一功能时，需首先进入对应的命令行视图，然后执行相应的命令进行配置。

进入接口配置接口的 IP 地址

```
[Datacom-Router]inter                                    //输入 TAB 补全命令
[Datacom-Router]interface                                  //“interface”是唯一可选的关键字
[Datacom-Router]interface g                                //输入 TAB 补全命令
[Datacom-Router]interface GigabitEthernet                   //“GigabitEthernet”是唯一可选的关
键字
[Datacom-Router]interface GigabitEthernet 0/0/1          //手动补全命令
```

输入命令的某个关键字的前几个字母，按下<tab>键，可以显示出完整的关键字，前提是这几个字母可以唯一标示出该关键字，否则，连续按下<tab>键，可出现不同的关键字，用户可以从中选择所需要的关键字。如：
“inter”+TAB，因为当前视图下以inter开头的命令只有interface，则命令直接补全为interface，连续按多次TAB也不会变化。

```
[Datacom-Router-GigabitEthernet0/0/1]
```

此时已经进入到了接口 GigabitEthernet0/0/1 的视图

```
[Datacom-Router-GigabitEthernet0/0/1]i?
  icmp   <Group> icmp command group
  igmp   Specify parameters for IGMP
  ip     <Group> ip command group
  ipsec  Specify IPSec(IP Security) configuration information
  ipv6   <Group> ipv6 command group
  isis   Configure interface parameters for ISIS
```

当用户输入命令时，如果只记得此命令关键字的开头一个或几个字符，可以使用部分帮助获取以该字符串开头的所有关键字的提示。如：
在GigabitEthernet0/0/1接口视图下，输入“i”+“?”，则会显示当前视图下所有“i”开头的命令的可选项，此时可以用TAB键补全，也可以手动补全。其中，“icmp”，“igmp”等为关键字，“<Group> icmp command group”，“Specify parameters for IGMP”等为对关键字的描述。

```
[Datacom-Router-GigabitEthernet0/0/1]ip ?
  accounting          <Group> accounting command group
  address             <Group> address command group
  binding             Enable binding of an interface with a VPN instance
  fast-forwarding     Enable fast forwarding
  forward-broadcast   Specify IP directed broadcast information
  netstream           IP netstream feature
```

```
  verify              IP verify
```

键入一条命令的部分关键字，后接以空格分隔的“?”，如果该位置为关键字，则列出全部关键字及其简单描述。如：
“ip” +空格+“?”，则会显示所有以ip为关键字的命令与对应的解释

```
[Datacom-Router-GigabitEthernet0/0/1]ip address ?
  IP_ADDR<X.X.X.X>    IP address
  bootp-alloc             IP address allocated by BOOTP
  dhcp-alloc              IP address allocated by DHCP
  unnumbered              Share an address with another interface
[Datacom-Router-GigabitEthernet0/0/1]ip address 192.168.1.1 ?
  INTEGER<0-32>       Length of IP address mask
  IP_ADDR<X.X.X.X>    IP address mask
[Datacom-Router-GigabitEthernet0/0/1]ip address 192.168.1.1 24 ?
  sub       Indicate a subordinate address
  <cr>      Please press ENTER to execute command
```

“<cr>”表示该位置没有关键字或参数，直接键入回车即可执行。

```
[Datacom-Router-GigabitEthernet0/0/1]dis this
#
interface GigabitEthernet0/0/1
 ip address 192.168.1.1 255.255.255.0
#
```

display this命令用来查看当前视图的运行配置。对于某些正在生效的配置参数，如果与缺省工作参数相同，则不显示；对于某些参数，虽然用户已经配置，但如果这些参数所在的命令没有成功提交，则不予显示。此命令常用于检查配置。
设备支持不完整关键字输入，即在当前视图下，当输入的字符能够匹配唯一的关键字时，可以不必输入完整的关键字。该功能提供了一种快捷的输入方式，有助于提高操作效率。如：
在接口下使用“**dis this**”，虽然没有输入完整的命令，但由于当前视图下匹配“**dis this**”的命令只有“**display this**”，所有命令可以正常执行。类似的还有“**dis cu**”、“**d cu**”等同于“display current-configuration”。

```
[Datacom-Router-GigabitEthernet0/0/1]quit
```

quit命令用来从当前视图退回到较低级别视图，如果是用户视图，则退出系统。

此时发现接口 IP 地址配置错误，需要将该地址配置到 interface GigabitEthernet 0/0/2 接口

```
[Datacom-Router]interface GigabitEthernet 0/0/1
[Datacom-Router-GigabitEthernet0/0/1]undo ip address
```

需要先删除GigabitEthernet0/0/1的IP地址配置，否则会产生地址冲突无法配置。在命令前加undo关键字，即为undo命令行。undo命令行一般用来恢复缺省情况、禁用某个功能或者删除某项配置。几乎每条配置命令都有对应的undo命令行。

```
[Datacom-Router]interface GigabitEthernet 0/0/2
[Datacom-Router-GigabitEthernet0/0/2]ip address 192.168.1.1 24
[Datacom-Router-GigabitEthernet0/0/2]quit
```

查看设备当前配置

```
[Datacom-Router]display current-configuration
[V200R003C00]
```

```
#
 sysname Datacom-Router
#
 snmp-agent local-engineid 800007DB03000000000000
 snmp-agent
#
 clock timezone China-Standard-Time minus 08:00:00
#
portal local-server load portalpage.zip
#
 drop illegal-mac alarm
#
 set cpu-usage threshold 80 restore 75
#
aaa
 authentication-scheme default
 authorization-scheme default
 accounting-scheme default
 domain default
 domain default_admin
 local-user admin password cipher %$%$K8m.Nt84DZ}e#<O`8bmE3Uw}%$%$
 local-user admin service-type http
#
  ---- More ----
```

当执行某一命令后，如果显示的信息超过一屏时，系统会自动暂停输出信息，以方便用户查看。此时在显示信息的最底部会出现“---- More ----”的字样，此时可以通过：

1. 键入<Ctrl+C>或<Ctrl+Z>，停止显示或命令执行。
2. 键入空格键，继续显示下一屏信息。
3. 键入回车键，继续显示下一行信息。

保存设备当前配置

```
# 返回到用户视图
[Datacom-Router]quit
<Datacom-Router>
```

除了通过quit命令外，也可以通过：

1. return命令，该命令可在任何视图下直接返回到用户视图。
2. ctrl+z快捷键，该快捷键可在任何视图下直接返回到用户视图。

```
# 保存配置
<Datacom-Router>save
  The current configuration will be written to the device.
  Are you sure to continue? (y/n)[n]:y                    //需要输入 y 来确认继续
```

```
  It will take several minutes to save configuration file, please
wait.......
  Configuration file had been saved successfully
  Note: The configuration file will take effect after being activated
```
当前配置已经成功保存！

用户通过命令行可以修改设备的当前配置，而这些配置未被保存的，如果要使当前配置在系统下次重启时仍然有效，在重启设备前，需要将当前配置保存到配置文件中。可以通过**save**直接保存到默认路径并覆盖原有的配置文件，也可以通过命令“**save** *configuration-file*”用来保存当前配置信息到存储设备中的指定文件中。该命令通常情况下不影响系统当前的启动配置文件

比较当前配置与下一次启动所使用的配置

```
<Datacom-Router>compare configuration
 The current configuration is the same as the next startup configuration file.
```
当前的配置与下次启动的配置文件内容一致

操作设备的文件系统

查看当前目录下的文件列表

```
<Datacom-Router>dir
Directory of flash:/

  Idx  Attr     Size(Byte)      Date       Time(LMT)     FileName
    0  -rw-    126,538,240    Jul 04 2016 17:57:22   ar651c-
v300r019c00Sspc100.cc
    1  -rw-         22,622    Feb 20 2020 10:35:18   mon_file.txt
    2  -rw-            737    Feb 20 2020 10:38:36   vrpcfg.zip
    3  drw-              -    Jul 04 2016 18:51:04   CPM_ENCRYPTED_FOLDER
    4  -rw-            783    Jul 10 2018 14:46:16   default_local.cer
    5  -rw-              0    Sep 11 2017 00:00:54   brdxpon_snmp_cfg.efs
    6  drw-              -    Sep 11 2017 00:01:22   update
    7  drw-              -    Sep 11 2017 00:01:48   shelldir
    8  drw-              -    Sep 21 2019 17:14:24   localuser
    9  drw-              -    Sep 15 2017 04:35:52   dhcp
   10  -rw-            509    Feb 20 2020 10:38:40   private-data.txt
   11  -rw-          2,686    Dec 19 2019 15:05:18   mon_lpu_file.txt
   12  -rw-          3,072    Dec 18 2019 18:15:54   Boot_LogFile

510,484 KB total available (386,456 KB free)
```

vrpcfg.zip：配置文件。配置文件必须以“.cfg”或“.zip”作为扩展名。
ar651c-v300r019c00Sspc100.cc：系统软件。系统软件必须以“.cc”作为扩展名。

将当前的配置保存同时命名为test.cfg

```
<Datacom-Router>save test.cfg
 Are you sure to save the configuration to test.cfg? (y/n)[n]:y          //需要输
入y来确认
  It will take several minutes to save configuration file, please wait.......
  Configuration file had been saved successfully
```

```
Note: The configuration file will take effect after being activated
```

再次查看当前目录下的文件列表

```
<Datacom-Router>dir
Directory of flash:/

  Idx  Attr     Size(Byte)          Date    Time(LMT)      FileName
    0  -rw-    126,538,240    Jul 04 2016 17:57:22   ar651c-
v300r019c00Sspc100.cc
    1  -rw-         22,622    Feb 20 2020 10:35:18   mon_file.txt
    2  -rw-            737    Feb 20 2020 10:38:36   vrpcfg.zip
    3  drw-              -    Jul 04 2016 18:51:04   CPM_ENCRYPTED_FOLDER
    4  -rw-            783    Jul 10 2018 14:46:16   default_local.cer
    5  -rw-              0    Sep 11 2017 00:00:54   brdxpon_snmp_cfg.efs
    6  drw-              -    Sep 11 2017 00:01:22   update
    7  drw-              -    Sep 11 2017 00:01:48   shelldir
    8  drw-              -    Sep 21 2019 17:14:24   localuser
    9  drw-              -    Sep 15 2017 04:35:52   dhcp
   10  -rw-          1,404    Feb 20 2020 11:55:17   test.cfg
   11  -rw-            509    Feb 20 2020 11:55:18   private-data.txt
   12  -rw-          2,686    Dec 19 2019 15:05:18   mon_lpu_file.txt
   13  -rw-          3,072    Dec 18 2019 18:15:54   Boot_LogFile

510,484 KB total available (386,452 KB free)
```

配置文件保存成功！

把该文件设置为设备下一次启动所使用的配置文件

```
<Datacom-Router>startup saved-configuration test.cfg
This operation will take several minutes, please wait.....
Info: Succeeded in setting the file for booting system
```

查看下一次启动所用的文件

```
<Datacom-Router>display startup
MainBoard:
  Startup system software:                    flash:/ ar651c-
v300r019c00Sspc100.cc
  Next startup system software:              flash:/ ar651c-
v300r019c00Sspc100.cc
  Backup system software for next startup:    null
  Startup saved-configuration file:           flash:/vrpcfg.zip
  Next startup saved-configuration file:      flash:/test.cfg
  Startup license file:                       null
  Next startup license file:                  null
  Startup patch package:                     null
  Next startup patch package:                 null
  Startup voice-files:                        null
  Next startup voice-files:                 null
```

display startup命令用来查看设备本次及下次启动相关的系统软件、备份系统软件、配置文件、License文件、补丁文件以及语音文件等。

清空配置文件

```
<Datacom-Router>reset saved-configuration
This will delete the configuration in the flash memory.
The device configuratio
ns will be erased to reconfigure.
Are you sure? (y/n)[n]:y                          //需要输入 y 来确认
 Clear the configuration in the device successfully.
```

重启设备

```
<Datacom-Router>reboot
Info: The system is comparing the configuration, please wait.
System will reboot! Continue ? [y/n]:y                          //需要
输入 y 来确认继续
Info: system is rebooting ,please wait...
系统开始重启
<Datacom-Router>
设备重启完成
```

3. 实训结果

通过本次实训，掌握了华为设备配置，了解并熟悉华为 VRP 系统的基本操作

4. 撰写实训报告

略

项目二：IPv4 编址及 IPv4 路由基础项目

实训目标：

掌握接口 IPv4 地址的配置方法

理解 LoopBack 接口的作用与含义

理解直连路由的产生原则

掌握静态路由的配置方法并理解其生效的条件

掌握通过 PING 工具测试网络层联通性

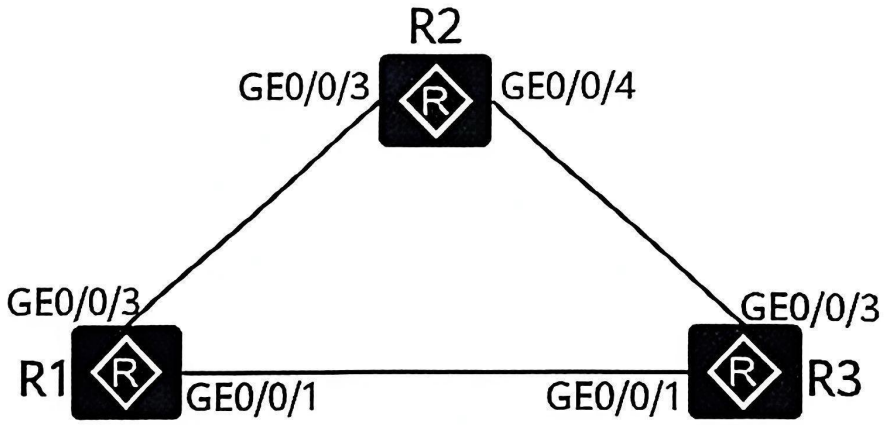
掌握并理解特殊静态路由的配置方法与应用场景

实训技能：

1. 配置路由器上各接口的 IP 地址
2. 配置静态路由来实现互联互通

实训内容：

R1、R2、R3 都是各自网络的网关设备，现在需要通过相应的配置，来实现这些网络之间的互联互通。



IPv4 编址及 IPv4 路由基础项目拓扑

1. 操作要求

能够掌握 IPV4 地址编制方法，理解特殊静态路由的配置方法与应用场景，掌握静态路由的配置方法并理解其生效的条件，理解 LoopBack 接口的作用与含义。

2. 操作步骤

设备基础配置

设备命名

略。

查看路由器当前接口 IP 地址配置与路由表

查看路由器上的接口状态，仅以 R1 为例

```
[R1]display ip interface brief
*down: administratively down
^down: standby
(l): loopback
(s): spoofing
(E): E-Trunk down
The number of interface that is UP in Physical is 3
The number of interface that is DOWN in Physical is 5
The number of interface that is UP in Protocol is 1
The number of interface that is DOWN in Protocol is 10

Interface                         IP Address/Mask      Physical
  Protocol
GigabitEthernet0/0/1              unassigned           up
      down
GigabitEthernet0/0/2              unassigned           up
      down
GigabitEthernet0/0/3              unassigned           up
  down
```

display ip interface brief命令用来查看接口与IP相关的简要信息，包括IP地址、子网掩码、物理状态和协议状态以及处于不同状态的接口数目等。

当前R1上的GigabitEthernet0/0/1和GigabitEthernet0/0/3接口由于尚未配置IP地址，所以IP Address/Mask字段为unassigned状态，Protocol字段为down状态，Physical字段为up状态。

查看路由器上的路由表情况，仅以 R1 为例

```
[R1]display ip routing-table
Route Flags: R - relay, D - download to fib
------------------------------------------------------------------------------
---------
Routing Tables: Public
         Destinations : 4        Routes : 4

Destination/Mask    Proto   Pre  Cost      Flags NextHop
Interface

      127.0.0.0/8   Direct  0    0           D   127.0.0.1
InLoopBack0
      127.0.0.1/32  Direct  0    0           D   127.0.0.1
InLoopBack0
```

```
127.255.255.255/32  Direct  0    0           D   127.0.0.1
InLoopBack0
255.255.255.255/32  Direct  0    0           D   127.0.0.1
InLoopBack0
```

InLoopBack0为设备上默认创建的环回接口，它是一个特殊的、固定的LoopBack接口。

InLoopBack0接口使用环回地址127.0.0.1/8，用来接收所有发送给本机的数据包。该接口上的IP地址是不可以改变的，也不通过路由协议对外发布。

配置路由物理接口的 IP 地址

按照下表配置路由器的物理接口的 IP 地址

路由器	接口	IP Address/Mask
R1	GigabitEthernet0/0/1	10.0.13.1/24
	GigabitEthernet0/0/3	10.0.12.1/24
R2	GigabitEthernet0/0/3	10.0.12.2/24
	GigabitEthernet0/0/4	10.0.23.2/24
R3	GigabitEthernet0/0/1	10.0.13.3/24
	GigabitEthernet0/0/3	10.0.23.3/24

设备物理接口IP

```
<R1>system-view
[R1]interface GigabitEthernet0/0/1
[R1-GigabitEthernet0/0/1]ip address 10.0.13.1 24
[R1-GigabitEthernet0/0/1]quit
[R1]interface GigabitEthernet0/0/3
[R1-GigabitEthernet0/0/3]ip address 10.0.12.1 24
[R1-GigabitEthernet0/0/3]quit

<R2>system-view
[R2]interface GigabitEthernet0/0/3
[R2-GigabitEthernet0/0/3]ip address 10.0.12.2 24
[R2-GigabitEthernet0/0/3]quit
[R2]interface GigabitEthernet0/0/4
[R2-GigabitEthernet0/0/4]ip address 10.0.23.2 24
[R2-GigabitEthernet0/0/4]quit

<R3>system-view
[R3]interface GigabitEthernet0/0/1
[R3-GigabitEthernet0/0/1]ip address 10.0.13.3 24
[R3-GigabitEthernet0/0/1]quit
[R3]interface GigabitEthernet0/0/3
```

```
[R3-GigabitEthernet0/0/3]ip address 10.0.23.3 24
[R3-GigabitEthernet0/0/3]quit
```

使用 ping 工具测试联通性

```
[R1]ping 10.0.12.2
  PING 10.0.12.2: 56  data bytes, press CTRL_C to break
    Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=70 ms
    Reply from 10.0.12.2: bytes=56 Sequence=2 ttl=255 time=50 ms
    Reply from 10.0.12.2: bytes=56 Sequence=3 ttl=255 time=40 ms
    Reply from 10.0.12.2: bytes=56 Sequence=4 ttl=255 time=30 ms
    Reply from 10.0.12.2: bytes=56 Sequence=5 ttl=255 time=50 ms

  --- 10.0.12.2 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 30/48/70 ms


[R1]ping 10.0.13.3
  PING 10.0.13.3: 56  data bytes, press CTRL_C to break
    Reply from 10.0.13.3: bytes=56 Sequence=1 ttl=255 time=50 ms
    Reply from 10.0.13.3: bytes=56 Sequence=2 ttl=255 time=60 ms
    Reply from 10.0.13.3: bytes=56 Sequence=3 ttl=255 time=50 ms
    Reply from 10.0.13.3: bytes=56 Sequence=4 ttl=255 time=30 ms
    Reply from 10.0.13.3: bytes=56 Sequence=5 ttl=255 time=30 ms

  --- 10.0.13.3 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 30/44/60 ms
```

查看 R1 的路由表

```
 [R1]display ip routing-table
Route Flags: R - relay, D - download to fib
------------------------------------------------------------------------------
---------
Routing Tables: Public
         Destinations : 10        Routes : 10

Destination/Mask    Proto   Pre  Cost      Flags NextHop
Interface

      10.0.12.0/24  Direct  0    0           D   10.0.12.1
GigabitEthernet0/0/3
      10.0.12.1/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/3
```

```
    10.0.12.255/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/3
      10.0.13.0/24  Direct  0    0           D   10.0.13.1
GigabitEthernet0/0/1
      10.0.13.1/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/1
    10.0.13.255/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/1
      127.0.0.0/8   Direct  0    0           D   127.0.0.1
InLoopBack0
      127.0.0.1/32  Direct  0    0           D   127.0.0.1
InLoopBack0
127.255.255.255/32  Direct  0    0           D   127.0.0.1
InLoopBack0
255.255.255.255/32  Direct  0    0           D   127.0.0.1
InLoopBack0
```

可以看到，在接口IP地址配置完成之后，针对每个接口自动生成了三条直连路由。分别是：

1. 指向接口所在网段的路由。
2. 指向接口IP地址的主机路由。
3. 指向接口所在网段广播地址的主机路由。

注：主机路由就是掩码长度为32的路由。

创建并配置 LoopBack 接口

按照下表配置设备的 LoopBack 接口

路由器	接口	IP Address/Mask
R1	LoopBack0	10.0.1.1/32
R2	LoopBack0	10.0.1.2/32
R3	LoopBack0	10.0.1.3/32

设备LoopBack接口IP

LoopBack接口属于设备上的逻辑接口，逻辑接口是指能够实现数据交换功能但物理上不存在、需要通过配置建立的接口。LoopBack接口创建后除非手工关闭该接口，否则LoopBack接口物理层状态和链路层协议永远处于UP状态。一般情况下，LoopBack接口使用32位掩码。使用LoopBack接口一般有如下目的：

1. 作为一台路由器的管理地址，起到标识一台设备的作用。
2. 使用该接口地址作为动态路由协议OSPF的router id。
3. 其他提高网络可靠性的用途。

本项目使用LoopBack接口模拟客户端。

```
[R1]interface LoopBack0
[R1-LoopBack0]ip address 10.0.1.1 32
```

```
[R2]interface LoopBack0
[R2-LoopBack0]ip address 10.0.1.2 32

[R3]interface LoopBack0
[R3-LoopBack0]ip address 10.0.1.3 32
```

查看设备上的路由表，以 R1 为例

```
[R1]display ip routing-table
Route Flags: R - relay, D - download to fib
------------------------------------------------------------------------------
Routing Tables: Public
         Destinations : 11        Routes : 11

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

        10.0.1.1/32 Direct  0    0           D   127.0.0.1       LoopBack0
       10.0.12.0/24 Direct  0    0           D   10.0.12.1
GigabitEthernet0/0/3
       10.0.12.1/32 Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/3
     10.0.12.255/32 Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/3
       10.0.13.0/24 Direct  0    0           D   10.0.13.1
GigabitEthernet0/0/1
       10.0.13.1/32 Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/1
     10.0.13.255/32 Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/1
        127.0.0.0/8 Direct  0    0           D   127.0.0.1       InLoopBack0
        127.0.0.1/32 Direct  0    0           D   127.0.0.1       InLoopBack0
127.255.255.255/32 Direct  0    0           D   127.0.0.1       InLoopBack0
255.255.255.255/32 Direct  0    0           D   127.0.0.1       InLoopBack0
```

此时已经生成了相应的直连路由

测试各 LoopBack 接口之间的联通性

```
[R1]ping -a 10.0.1.1 10.0.1.2
  PING 10.0.1.2: 56  data bytes, press CTRL_C to break
    Request time out
    Request time out
    Request time out
    Request time out
    Request time out

  --- 10.0.1.2 ping statistics ---
    5 packet(s) transmitted
    0 packet(s) received
    100.00% packet loss
```

ping -a *source-ip-address destination-ip-address*命令用来指定发送ICMP ECHO-REQUEST报文的源IP地址及目的IP地址。此时由于路由器上没有到底该目的IP的路由条目，所以无法PING通。

配置静态路由

在 R1 上配置到达 R2 和 R3 的 LoopBack0 接口的路由条目

```
[R1]ip route-static 10.0.1.2 32 10.0.12.2
[R1]ip route-static 10.0.1.3 32 10.0.13.3
```

查看 R1 的路由表

```
[R1]display ip routing-table
Route Flags: R - relay, D - download to fib
------------------------------------------------------------------------------
Routing Tables: Public
         Destinations : 13        Routes : 13

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

        10.0.1.1/32  Direct  0    0           D   127.0.0.1       LoopBack0
        10.0.1.2/32  Static  60   0           RD  10.0.12.2
GigabitEthernet0/0/3
        10.0.1.3/32  Static  60   0           RD  10.0.13.3
GigabitEthernet0/0/1
       10.0.12.0/24  Direct  0    0           D   10.0.12.1
GigabitEthernet0/0/3
       10.0.12.1/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/3
     10.0.12.255/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/3
       10.0.13.0/24  Direct  0    0           D   10.0.13.1
GigabitEthernet0/0/1
       10.0.13.1/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/1
     10.0.13.255/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/1
        127.0.0.0/8  Direct  0    0           D   127.0.0.1       InLoopBack0
        127.0.0.1/32 Direct  0    0           D   127.0.0.1       InLoopBack0
127.255.255.255/32  Direct  0    0           D   127.0.0.1       InLoopBack0
255.255.255.255/32  Direct  0    0           D   127.0.0.1       InLoopBack0
```

配置的静态路由被加入到了 IP 路由表中

测试联通性

```
[R1]ping -a 10.0.1.1 10.0.1.2
  PING 10.0.1.2: 56  data bytes, press CTRL_C to break
    Request time out
    Request time out
    Request time out
    Request time out
    Request time out

  --- 10.0.1.2 ping statistics ---
    5 packet(s) transmitted
    0 packet(s) received
    100.00% packet loss
```

还是无法 PING 通 R2 的 LoopBack0 接口，因为此时 R2 上没有到 R1 的 LoopBack0 的路由

在 R2 上添加到达 R1 的 LoopBack0 的路由

```
[R2]ip route-static 10.0.1.1 32 10.0.12.1
```

测试联通性

```
<R1>ping -a 10.0.1.1 10.0.1.2
  PING 10.0.1.2: 56  data bytes, press CTRL_C to break
    Reply from 10.0.1.2: bytes=56 Sequence=1 ttl=255 time=60 ms
    Reply from 10.0.1.2: bytes=56 Sequence=2 ttl=255 time=30 ms
    Reply from 10.0.1.2: bytes=56 Sequence=3 ttl=255 time=10 ms
    Reply from 10.0.1.2: bytes=56 Sequence=4 ttl=255 time=50 ms
    Reply from 10.0.1.2: bytes=56 Sequence=5 ttl=255 time=30 ms

  --- 10.0.1.2 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
round-trip min/avg/max = 10/36/60 ms
```

此时 R1 的 LoopBack0 已经可以和 R2 的 LoopBack0 实现互通。

完成剩余路由条目的配置

```
[R2]ip route-static 10.0.1.3 32 10.0.23.3
```

```
[R3]ip route-static 10.0.1.1 32 10.0.13.1
[R3]ip route-static 10.0.1.2 32 10.0.23.2
```

读者自行测试路由器的 LoopBack0 接口之间的联通性

配置 R1->R3->R2 作为 R1 的 LoopBack0 到 R2 的 LoopBack0 接口的备份路径

配置 R1 和 R2 上的静态路由

```
[R1]ip route-static 10.0.1.2 32 10.0.13.3 preference 100
```

```
[R2]ip route-static 10.0.1.1 32 10.0.23.3 preference 100
```

查看 R1 和 R2 上的路由表

```
[R1]display ip routing-table
Route Flags: R - relay, D - download to fib
------------------------------------------------------------------------------
Routing Tables: Public
         Destinations : 13        Routes : 13

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

        10.0.1.1/32  Direct  0    0           D   127.0.0.1       LoopBack0
        10.0.1.2/32  Static  60   0          RD   10.0.12.2
GigabitEthernet0/0/3
        10.0.1.3/32  Static  60   0          RD   10.0.13.3
GigabitEthernet0/0/1
       10.0.12.0/24  Direct  0    0           D   10.0.12.1
GigabitEthernet0/0/3
```

```
      10.0.12.1/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/3
    10.0.12.255/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/3
      10.0.13.0/24  Direct  0    0           D   10.0.13.1
GigabitEthernet0/0/1
      10.0.13.1/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/1
    10.0.13.255/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/1
      127.0.0.0/8   Direct  0    0           D   127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct  0    0           D   127.0.0.1       InLoopBack0
127.255.255.255/32  Direct  0    0           D   127.0.0.1       InLoopBack0
255.255.255.255/32  Direct  0    0           D   127.0.0.1       InLoopBack0
```

```
[R2]display ip routing-table
Route Flags: R - relay, D - download to fib
------------------------------------------------------------------------------
Routing Tables: Public
         Destinations : 13        Routes : 13

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

       10.0.1.1/32  Static  60   0           RD  10.0.12.1
GigabitEthernet0/0/3
       10.0.1.2/32  Direct  0    0           D   127.0.0.1       LoopBack0
       10.0.1.3/32  Static  60   0           RD  10.0.23.3
GigabitEthernet0/0/4
      10.0.12.0/24  Direct  0    0           D   10.0.12.2
GigabitEthernet0/0/3
      10.0.12.2/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/3
    10.0.12.255/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/3
      10.0.23.0/24  Direct  0    0           D   10.0.23.2
GigabitEthernet0/0/4
      10.0.23.2/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/4
    10.0.23.255/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/4
      127.0.0.0/8   Direct  0    0           D   127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct  0    0           D   127.0.0.1       InLoopBack0
127.255.255.255/32  Direct  0    0           D   127.0.0.1       InLoopBack0
255.255.255.255/32  Direct  0    0           D   127.0.0.1       InLoopBack0
```

此时配置的 preference 为 100 的静态路由没有被加载到路由表中。

关闭 R1 和 R2 之间的链路对应的接口（GigabitEthernet0/0/3），使得优先级高的路由失效。

```
[R1]interface GigabitEthernet0/0/3
[R1-GigabitEthernet0/0/3]shutdown
```

查看 R1 和 R2 的路由表，随着高优先级路由失效，低优先级路由被激活

```
[R1]display IP routing-table
Route Flags: R - relay, D - download to fib
------------------------------------------------------------------------------
Routing Tables: Public
         Destinations : 10        Routes : 10

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

        10.0.1.1/32  Direct  0    0           D   127.0.0.1       LoopBack0
        10.0.1.2/32  Static  100  0           RD  10.0.13.3
GigabitEthernet0/0/1
        10.0.1.3/32  Static  60   0           RD  10.0.13.3
GigabitEthernet0/0/1
       10.0.13.0/24  Direct  0    0           D   10.0.13.1
GigabitEthernet0/0/1
       10.0.13.1/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/1
     10.0.13.255/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/1
       127.0.0.0/8   Direct  0    0           D   127.0.0.1       InLoopBack0
       127.0.0.1/32  Direct  0    0           D   127.0.0.1       InLoopBack0
127.255.255.255/32   Direct  0    0           D   127.0.0.1       InLoopBack0
255.255.255.255/32   Direct  0    0           D   127.0.0.1       InLoopBack0

[R2]display ip routing-table
Route Flags: R - relay, D - download to fib
------------------------------------------------------------------------------
Routing Tables: Public
         Destinations : 10        Routes : 10

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

        10.0.1.1/32  Static  100  0           RD  10.0.23.3
GigabitEthernet0/0/4
        10.0.1.2/32  Direct  0    0           D   127.0.0.1       LoopBack0
        10.0.1.3/32  Static  60   0           RD  10.0.23.3
GigabitEthernet0/0/4
       10.0.23.0/24  Direct  0    0           D   10.0.23.2
GigabitEthernet0/0/4
       10.0.23.2/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/4
     10.0.23.255/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/4
       127.0.0.0/8   Direct  0    0           D   127.0.0.1       InLoopBack0
       127.0.0.1/32  Direct  0    0           D   127.0.0.1       InLoopBack0
127.255.255.255/32   Direct  0    0           D   127.0.0.1       InLoopBack0
255.255.255.255/32   Direct  0    0           D   127.0.0.1       InLoopBack0
```

此时由于链路断开，原先的静态路由失效，低优先级的静态路由被激活。

检查联通性

```
[R1]ping -a 10.0.1.1 10.0.1.2
  PING 10.0.1.2: 56  data bytes, press CTRL_C to break
    Reply from 10.0.1.2: bytes=56 Sequence=1 ttl=254 time=80 ms
    Reply from 10.0.1.2: bytes=56 Sequence=2 ttl=254 time=60 ms
    Reply from 10.0.1.2: bytes=56 Sequence=3 ttl=254 time=60 ms
```

```
  Reply from 10.0.1.2: bytes=56 Sequence=4 ttl=254 time=110 ms
  Reply from 10.0.1.2: bytes=56 Sequence=5 ttl=254 time=80 ms

--- 10.0.1.2 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 60/78/110 ms
```

追踪数据包路径

```
[R1]tracert -a 10.0.1.1 10.0.1.2

 traceroute to  10.0.1.2(10.0.1.2), max hops: 30 ,packet length: 40,press CTRL_C to
break

 1 10.0.13.3 40 ms  30 ms  50 ms

 2 10.0.23.2 80 ms  80 ms  60 ms
```

tracert命令主要用于查看数据包从源端到目的端的路径信息。
可以看到数据包经过了R3的GigabitEthernet0/0/1，再经过R3的GigabitEthernet0/0/3转发给R2的GigabitEthernet0/0/4。
注：部分项目环境下设备出于安全考虑，不会回复ICMP报文，项目现象可能会有所偏差，可以按ctrl+c结束tracert。

通过默认路由实现 R1 的 LoopBack0 接口和 R2 的 LoopBack0 接口互联互通

恢复接口并删除已经配置的路由条目

```
[R1]interface GigabitEthernet0/0/3
[R1-GigabitEthernet0/0/3]undo shutdown
[R1-GigabitEthernet0/0/3]quit
[R1]undo ip route-static 10.0.1.2 255.255.255.255 10.0.12.2
[R1]undo ip route-static 10.0.1.2 255.255.255.255 10.0.13.3 preference 100
```

查看 R1 的路由表

```
[R1]display ip routing-table
Route Flags: R - relay, D - download to fib
------------------------------------------------------------------------------
Routing Tables: Public
         Destinations : 12        Routes : 12

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

        10.0.1.1/32  Direct  0    0           D   127.0.0.1       LoopBack0
        10.0.1.3/32  Static  60   0          RD   10.0.13.3
GigabitEthernet0/0/1
       10.0.12.0/24  Direct  0    0           D   10.0.12.1
GigabitEthernet0/0/3
       10.0.12.1/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/3
     10.0.12.255/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/3
```

```
      10.0.13.0/24   Direct  0    0           D   10.0.13.1
GigabitEthernet0/0/1
      10.0.13.1/32   Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/1
    10.0.13.255/32   Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/1
      127.0.0.0/8    Direct  0    0           D   127.0.0.1       InLoopBack0
      127.0.0.1/32   Direct  0    0           D   127.0.0.1       InLoopBack0
127.255.255.255/32   Direct  0    0           D   127.0.0.1       InLoopBack0
255.255.255.255/32   Direct  0    0           D   127.0.0.1       InLoopBack0
```

此时 R1 上没有到 R2 的 LoopBack0（10.0.1.2/32）的路由条目

在 R1 上配置默认路由

```
[R1]ip route-static 0.0.0.0 0 10.0.12.2
```

查看 R1 的路由条目

```
[R1]display ip routing-table
Route Flags: R - relay, D - download to fib
------------------------------------------------------------------------------
Routing Tables: Public
         Destinations : 13        Routes : 13

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

        0.0.0.0/0   Static  60   0           RD  10.0.12.2
GigabitEthernet0/0/3
       10.0.1.1/32  Direct  0    0           D   127.0.0.1       LoopBack0
       10.0.1.3/32  Static  60   0           RD  10.0.13.3
GigabitEthernet0/0/1
      10.0.12.0/24  Direct  0    0           D   10.0.12.1
GigabitEthernet0/0/3
      10.0.12.1/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/3
    10.0.12.255/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/3
      10.0.13.0/24  Direct  0    0           D   10.0.13.1
GigabitEthernet0/0/1
      10.0.13.1/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/1
    10.0.13.255/32  Direct  0    0           D   127.0.0.1
GigabitEthernet0/0/1
      127.0.0.0/8   Direct  0    0           D   127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct  0    0           D   127.0.0.1       InLoopBack0
127.255.255.255/32  Direct  0    0           D   127.0.0.1       InLoopBack0
255.255.255.255/32  Direct  0    0           D   127.0.0.1       InLoopBack0
```

默认路由已经被激活

测试 R1 的 LoopBack0 接口到 R2 的 LoopBack0 接口的联通性

```
[R1]ping -a 10.0.1.1 10.0.1.2
  PING 10.0.1.2: 56  data bytes, press CTRL_C to break
    Reply from 10.0.1.2: bytes=56 Sequence=1 ttl=255 time=50 ms
    Reply from 10.0.1.2: bytes=56 Sequence=2 ttl=255 time=30 ms
    Reply from 10.0.1.2: bytes=56 Sequence=3 ttl=255 time=20 ms
    Reply from 10.0.1.2: bytes=56 Sequence=4 ttl=255 time=40 ms
```

```
  Reply from 10.0.1.2: bytes=56 Sequence=5 ttl=255 time=20 ms

--- 10.0.1.2 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 20/32/50 ms
```

此时 R1 的 LoopBack0 接口到 R2 的 LoopBack0 接口之间可以互联互通。

3. 实训结果

通过实训，掌握接口 IPv4 地址的配置方法，理解 LoopBack 接口的作用与含义，理解直连路由的产生原则；掌握静态路由的配置方法并理解其生效的条件，掌握通过 PING 工具测试网络层联通性

4. 撰写实训报告

略