



太原理工大学  
TAIYUAN UNIVERSITY OF TECHNOLOGY

## 课程设计

课程名称： 物联网传输课程设计

设计名称： 基于 Zigbee 的智能农业大棚环境检测

专业班级： 物联网 1501 学号： 2015001965

学生姓名： 高磊

指导教师： 张昊

2018 年      6 月      20 日

## 5 基于 Zigbee 的智能农业大棚环境检测

### 5.1 【问题描述】

我们周围的环境对生活有很大的影响，设计一个 Zigbee 的智能农业大棚环境检测系统，自动检测农业生态信息，可以自动开启或关闭指定设备。并且设计一个显示界面，在 PC 机上实时显示相应的环境变量。

### 5.2 【基本要求】

查阅资料，首先搭建一个智能农业大棚环境检测系统，然后编写传感器采集程序、协调器串口发送程序和 PC 机显示界面程序。能够实现环境变量的采集、显示甚至调节的功能。

（实现提示：对环境的监测需要选择合适的传感器，首先根据不同传感器的特性进行选型。传感器模块要实现环境变量的采集，并将数据传送给协调器。在 PC 机侧，建立一个合适的显示界面。并将接受到的数据存入到数据库中。）

### 5.3 【功能分析】

基于 Zigbee 的智能农业大棚环境检测主要包括自动分析农业大棚内部环境的状况，并实时作出相应措施。

- 1) 用 USB 转串口来完成对温湿度、光照等传感器信息的采集。
- 2) 通过 USB 转串口来控制电机的运行。
- 3) 功能：
  - a) 接受来自传感器的温湿度和光照信息。
  - b) 如果农业大棚环境中温度过高则开启电机（风扇）、如果温度适合则关闭（风扇）；
  - c) 如果光照传感器检测到的光照强度弱，则开启 led 灯。光照强度适合则关闭 led 灯。

### 5.4 【概要设计】

#### 5.4.1 系统组成

系统大体组成包括：感知数据、传送数据、处理数据、数据呈现。系统框图如图所示。

由于电机模块没有外接电路，因此需要通过中间的协调器以及 USB 将数据转为串口形式发送到 PC 机上。

5.4.2 传感器选型

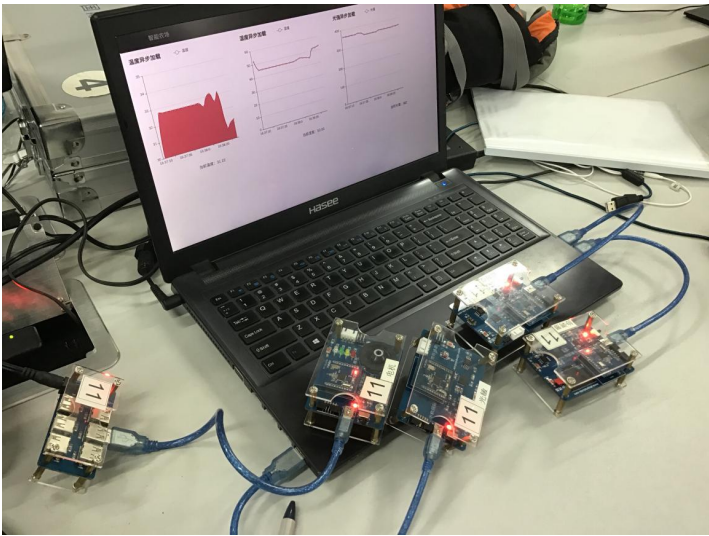
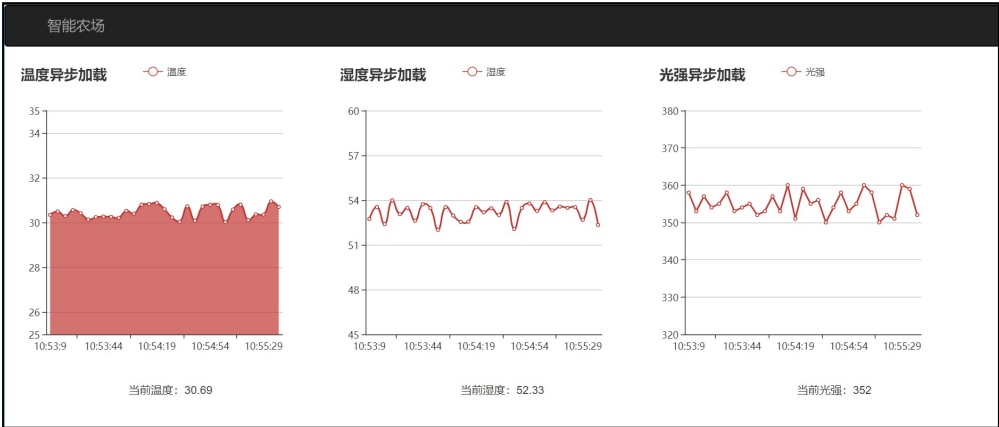
WSN 温湿度传感器、WSN 光敏传感器、WSN 电机传感器的电机模块、WSN 点击传感器的 LED 模块。

5.5 【详细设计】

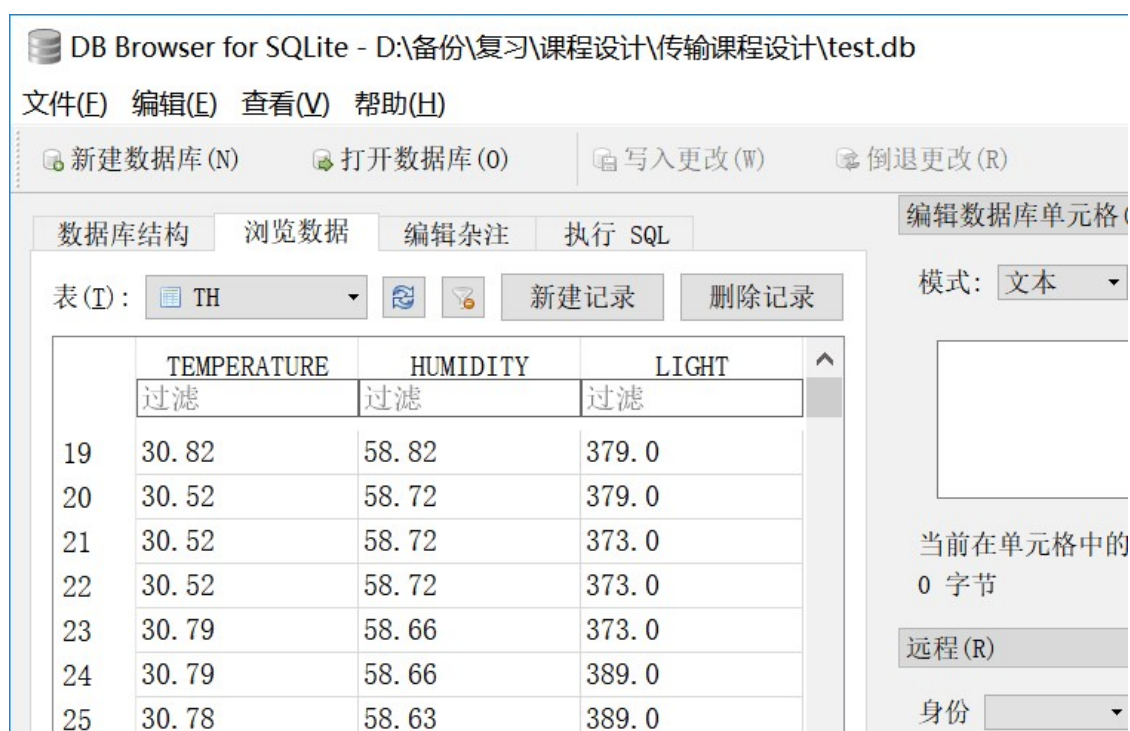
通过传感器来获取当前环境的基本情况。接受来自传感器的温湿度和光照信息。如果农业大棚环境中温度过高则开启电机（风扇）、如果温度适合则关闭（风扇）；如果光照传感器检测到的光照强度弱，则开启 led 灯。光照强度适合则关闭 led 灯。

5.6 【系统说明】

本系统搭建了一个智能农业大棚采集显示环境，能够采集周围环境的图像，并且实时在 PC 机上显示，PC 机实时显示界面如图：



数据库设计如图所示：



## 5.7 【系统源码】

```
import serial
import sqlite3
import json

ser = serial.Serial('/dev/ttyUSB0', 115200, timeout=None, parity=serial.PARITY_NONE, rtscts=0, stopbits=1, writeTimeout=None)

conn = sqlite3.connect('test.db')
c = conn.cursor()
temperature = 0
humidity = 0
light = 0

while 1:
    info = ser.read(16)
    # 温湿度传感器
    if (info[2] == 1) and (info[3] == 3) and (info[4] == 1):
        temperature = (int(info[5])*256 + int(info[6]))/100
        humidity = ((int(info[7])*256 + int(info[8]))/100)

    # 光照传感器
```

```

elif (info[2] == 1) and (info[3] == 2) and (info[4] == 1):
    light = (int(info[5]))*256 + int(info[6])

else:
    continue

print("温度"+str(temperature),end='\t')
print("湿度"+str(humidity),end='\t')
print("光强"+str(light))

if(temperature>30):
    # 温度高，电机开始转
    ser.write(b'\xcc\xee\x01\x09\x09\x00\x00\x00\x00\x00\x00\x00\x00\xff')
else:
    # 温度低，电机停止
    ser.write(b'\xcc\xee\x01\x09\x0b\x00\x00\x00\x00\x00\x00\x00\x00\xff')

if(light < 300):
    ser.write(b'\xcc\xee\x01\x09\x0c\x00\x00\x00\x00\x00\x00\x00\x00\xff')
else:
    ser.write(b'\xcc\xee\x01\x09\x01\x00\x00\x00\x00\x00\x00\x00\x00\xff')

sql = "insert into TH(TEMPERATURE,HUMIDITY,LIGHT) values"
("+str(temperature)+","+str(humidity)+","+str(light)+")"
print(sql)
c.execute(sql)
conn.commit()
data = {
    'temperature': temperature,
    'humidity': humidity,
    'light': light
}
with open('thInfo.json', 'w') as f:
    json.dump(data, f)

```

```

<!DOCTYPE html>
<html lang="zh-CN">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- 上述 3 个 meta 标签*必须*放在最前面，任何其他内容都*必须*跟随其后！ -->
    <title>farm</title>

```

```

<!-- Bootstrap -->
<link href="/css/bootstrap.min.css" rel="stylesheet">
<script src="echarts.min.js"></script>
<script src="jquery.js"></script>
</head>
<body>
  <nav class="navbar navbar-inverse">
    <div class="container">
      <div class="navbar-header">
        <a class="navbar-brand" href="#">智能农场</a>
      </div>
      <div id="navbar" class="collapse navbar-collapse">
      </div><!--/.nav-collapse -->
    </div>
  </nav>
  <div class="container-fluid">
    <center>
      <div class="row">
        <div class="col-md-4" id="temperature" style="width: 400px;height:400px;"></div>
        <div class="col-md-4" id="humidity" style="width: 400px;height:400px;"></div>
        <div class="col-md-4" id="light" style="width: 400px;height:400px;"></div>
      </div>
      <div class="row">
        <div class="col-md-4" >
          <p class="text-center" id="temperatureText"></p>
        </div>
        <div class="col-md-4" >
          <p class="text-center" id="humidityText"></p>
        </div>
        <div class="col-md-4" >
          <p class="text-center" id="lightText"></p>
        </div>
      </div>
    </div>
    <script type="text/javascript">
      // 基于准备好的 dom，初始化 echarts 实例
      var myChartTemperature = echarts.init(document.getElementById('temperature'));
      var myChartHumidity = echarts.init(document.getElementById('humidity'));
      var myChartLight = echarts.init(document.getElementById('light'));
      // 显示标题，图例和空的坐标轴
      myChartTemperature.setOption({
        title: {
          text: '温度异步加载'

```

```
    },
    tooltip: {},
    legend: {
        data:['温度']
    },
    xAxis: {
        data: []
    },
    yAxis: {
        min: 25,
        max: 35
    },
    series: [{
        smooth: true,
        areaStyle: {},
        name: '温度',
        type: 'line',
        data: [],
    }]
});

myChartHumidity.setOption({
    title: {
        text: '湿度异步加载'
    },
    tooltip: {},
    legend: {
        data:['湿度']
    },
    xAxis: {
        data: []
    },
    yAxis: {
        min: 45,
        max: 60
    },
    series: [{
        smooth: true,
        name: '湿度',
        type: 'line',
        data: []
    }]
});

myChartLight.setOption({
```

```

        title: {
            text: '光强异步加载'
        },
        tooltip: {},
        legend: {
            data:['光强']
        },
        xAxis: {
            data: []
        },
        yAxis: {
            min: 320,
            max: 380
        },
        series: [{
            name: '光强',
            type: 'line',
            data: []
        }]
    });
    var temperatures = [];
    var humidities = [];
    var lights = [];
    var date = [];
    var now = new Date();
    // 异步加载数据

    function timer() {
        now = [now.getHours(),now.getMinutes(),now.getSeconds()].join(':');
        date.push(now);

        $.get('thInfo.json').done(function (jsondata) {
            temperatures.push(jsondata.temperature)
            humidities.push(jsondata.humidity)
            lights.push(jsondata.light)
            document.getElementById("temperatureText").innerHTML=" 当前 温 度 :
"+jsondata.temperature;
            document.getElementById("humidityText").innerHTML=" 当前 湿 度 :
"+jsondata.humidity;
            document.getElementById("lightText").innerHTML="当前光强: "+jsondata.light;
            // 填入数据
            myChartTemperature.setOption({
                xAxis: {
                    data: date

```



```

        },
        series: [{
            // 根据名字对应到相应的系列

            name: '温度',
            data: temperatures
        }]
    });
    myChartHumidity.setOption({
        xAxis: {
            data: date
        },
        series: [{
            // 根据名字对应到相应的系列
            name: '湿度',
            data: humiditys
        }]
    });
    myChartLight.setOption({
        xAxis: {
            data: date
        },
        series: [{
            // 根据名字对应到相应的系列
            name: '光强',
            data: lights
        }]
    });
});
now = new Date();

}

setInterval(timer,5000)
</script>

<script src="jquery.js"></script>

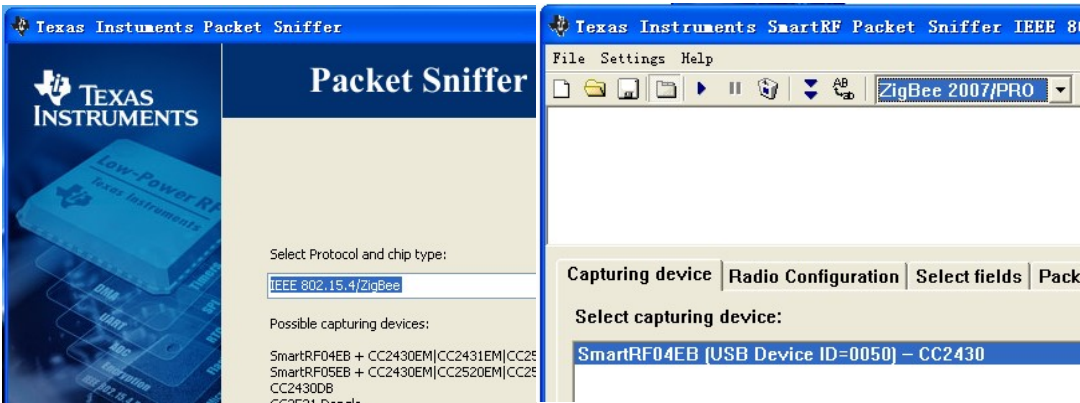
<script src="./js/bootstrap.min.js"></script>
</body>
</html>

```

## 6 ZigBee 环境检测数据包分析

### 6.1 【问题描述】

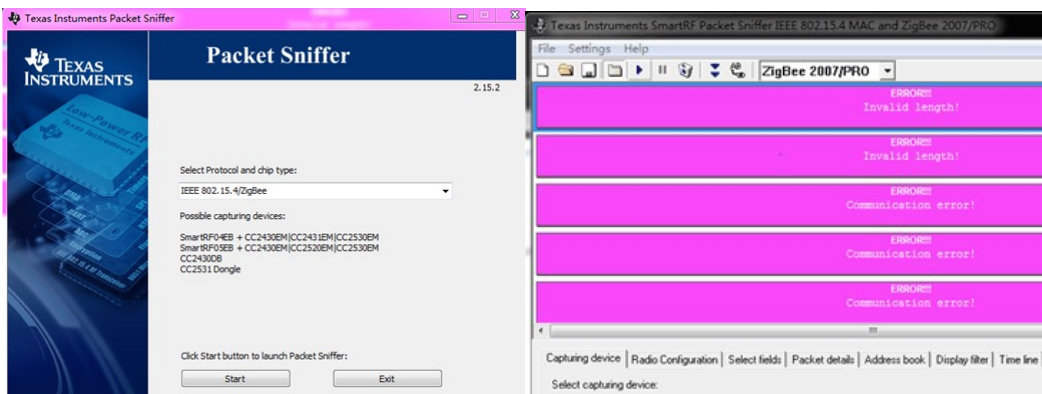
使用 Texas Instruments Packet Sniffer 软件进行抓包，然后分析捕获的数据包，进而形象地理解 ZigBee 网络数据的传输过程。如图所示，在下列列表框中选择 IEEE 802.15.4/ZigBee，最后单击 Start 按钮。



### 6.2 【基本要求】

从 Texas Instruments Packet Sniffer 软件抓到的数据包可以看到每个数据包有很多段组成，这与 ZigBee 协议是对应的，由于 ZigBee 协议栈是采用分层结构实现的，所以数据包显示时也是不同的层使用不同的颜色，进而可以查看分析相应的数据。分析网络的建立过程以及数据传输过程中，用户数据在数据包的哪个位置。

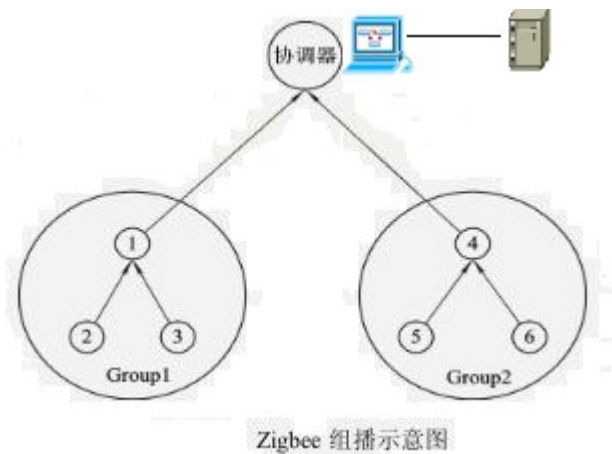
### 6.3 【实验结果】



# 7 WSN 传输组网

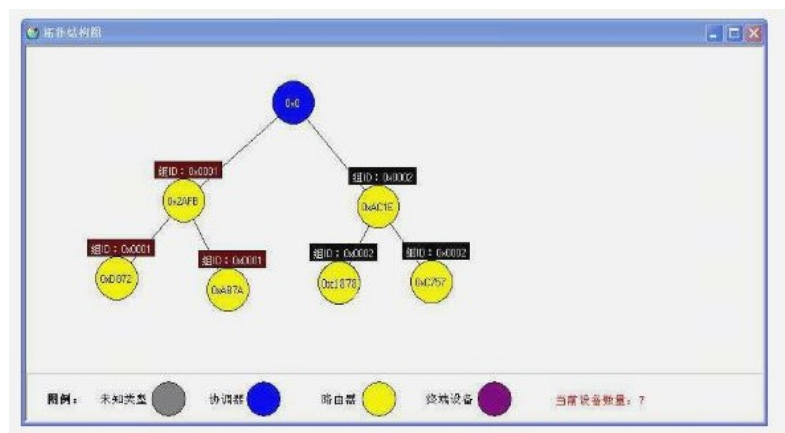
## 7.1 【问题描述】

Zigbee 数据传输也可以采用组播的方式：在一个 Zigbee 网络中分为几个组，每一组分配一个组头负责和协调器进行通信。本实验采用 6 个路由器或终端节点，每 3 个路由器/终端节点为一组，分成两组，其中每一组有一个组头负责和协调器进行通信。如图 S-1 所示。



## 7.2 【基本要求】

基于现有 Zigbee 实验系统，构建上述网络拓扑，按下图格式出具拓扑结构图。



其网络节点 ID、组 ID 自行规划。如：Group1 中的三个节点编号为 1、2、3，且 1 号节点为 Group1 的组头；Group2 中的三个节点编号为 4、5、6，且 4 号节点为 Group2 的组头。通过 IAR 对相应程序进行修改，报告中包含主要程序。程序包括两部分：协调器程序的编写和路由器/终端设备程序的编写。具体参考程序见 6.3 部分。

测试数据传输，将多于 2 终端的采集数据发送到协调器，协调器将收集的数据通过串口传送至 PC 机。在 PC 机上可以观察收集的信息，如图所示。在报告中描述测试结果。

