

**课 程 设 计**

**课 程：物联网工程开发与应用实例**

**题 目：基于Zigbee的智能温湿度检测系统-QT图形界面设计**

**学生姓名：林海**

**学 号：15115061027**

**二级学院：信息科学与工程学院**

**专 业：物联网工程**

**班 级：2015级01班**

**指导教师姓名及职称：刘晓樑 讲师**

**起止时间：2018年 9 月—— 2019年 1月**

（教务处制）

**基于Zigbee的智能温湿度检测系统**

**摘 要**： 随着互联网技术的不断成熟，各行各业的信息化已经越来越广泛，智能农业已经来到我们的身边，针对温室大棚传统的温湿度检测系统存在效率低、功率大和成本高等问题，设计出了一种无线温湿度检测系统，系统以无线射频芯片CC2530为核心设计了用与温湿度检测的Zigbee无线传感器网络，传感器通过局域网进行通信，利用QT生成了检测界面，实现了采集数据的可视化。

**关键字**： ZigBee；局域网；QT

**Zigbee-based intelligent temperature and humidty detection system**

**Abstract:** With the continuous maturity of Internet technology, the informationization of all walks of life has become more and more extensive. Intelligent agriculture has come to us. The traditional temperature and humidity detection system for greenhouses has problems such as low efficiency, high power and high cost. A wireless temperature and humidity detection system is designed. The system uses the wireless radio frequency chip CC2530 as the core to design the Zigbee wireless sensor network with temperature and humidity detection. The sensor communicates through the local area network, and the detection interface is generated by QT, which realizes the visualization of the collected data.

**Keywords:** Zigbee ;Local Area Network; QT

**目录**

[1 温湿度检测的设计相关技术 3](#_Toc534642892)

[1.1 ZigBee传输技术 3](#_Toc534642893)

[1.2 Z-Stack协议栈 4](#_Toc534642894)

[1.3 IAR开发环境 5](#_Toc534642895)

[1.4终端及QT图形应用技术 5](#_Toc534642896)

[**2 系统总体设计方案** 6](#_Toc534642897)

[2.1功能结构分析 6](#_Toc534642898)

[2.2技术流程 7](#_Toc534642899)

[2.3 各模块功能 7](#_Toc534642900)

[2.3.1采集模块 7](#_Toc534642901)

[2.3.2数据传输模块 8](#_Toc534642902)

[2.3.3协调器模块 8](#_Toc534642903)

[2.3.4 终端控制显示模块 9](#_Toc534642904)

[2.4 数据库设计 9](#_Toc534642905)

[3 系统软件的设计与实现 11](#_Toc534642906)

[3.1 硬件部分 11](#_Toc534642907)

[3.1.1 CC2530原理图 11](#_Toc534642908)

[3.1.2 传感器原理图 12](#_Toc534642909)

[3.2软件部分 13](#_Toc534642910)

[3.2.1数据采集模块 13](#_Toc534642911)

[3.2.2 QT图形化界面设计 13](#_Toc534642912)

[**4 系统测试** 19](#_Toc534642913)

[4.1 硬件测试 19](#_Toc534642914)

[4.2 软件测试 20](#_Toc534642915)

[**5 总结及展望** 20](#_Toc534642916)

[5.1 总结 20](#_Toc534642917)

[5.2 展望 21](#_Toc534642918)

[参考文献： 21](#_Toc534642919)

**基于Zigbee的智能温湿度检测系统-QT图形界面设计**

# 1 温湿度检测的设计相关技术

## 1.1 ZigBee传输技术

ZigBee传输技术是本设计十分重要的组成，本设计是立足于无线检测温湿度，那么，如何选择无线传输技术就显得相当重要。

下面本文通过比较几种常见的无线技术来说明本设计选择ZigBee的理由。

红外技术是比较早出现的无线技术，虽然有成本低、简单易用的特点，但是，有物体阻挡时，便显得无可奈何了，且仅可以直线传播，所以，不符合本设计需求；WiFi是应用的最多、最流行无线通信技术，传输最远范围是100-300米，它的最高速率将近300Mbps，功耗在10-15mA之间；Bluetooth的有效覆盖半径仅有2-30米，功耗小于WiFi，但大于ZigBee；ZigBee有效覆盖半径在50-300米之间，功耗5mA，还可以自组网，节点数为216个。显然不管是在功耗还是成本，或是传输距离方面ZigBee都是最适合本设计的。

经科学地分析和比对，本系统选择了ZigBee技术作为本系统的无线传输技术支持。

另外ZigBee通信需要至少需要两个CC2530模块，如图1-1所示，左边的节点是用于与传感器相连接的，右边的协调器是用于接收节点数据或者发送指令的。在本系统中，用了1个节点（节点是指连接传感器的主控模块），和一个协调器模块来搭建无线网和完成数据的汇总。



图1-1 ZigBee组网套件

## 1.2 Z-Stack协议栈

Z-Stack是用于CC2530组件的小型的操作系统，它是一个ZigBee模块化的协议栈，能用于于多个平台开发。它是一种小型的操作系统，它里面集成了很多函数方法，比如、、、等，可以十分容易的初始化操作系统，配置硬件，还可以十分方便的使用定时、串口、点灯等功能，可以大大的提高程序员编程效率。它主要进行了CC2530模块硬件的初始化，如WatchDog的使能、中断的初始化、显示屏的初始化等。Z-Stack协议栈架构如图1-2，分为物理层、MAC、NWK、APL层面。其工作流程如图1-3所示。

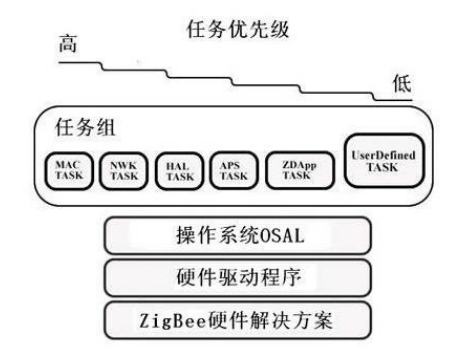


图1-2 Z-Stack软件架构

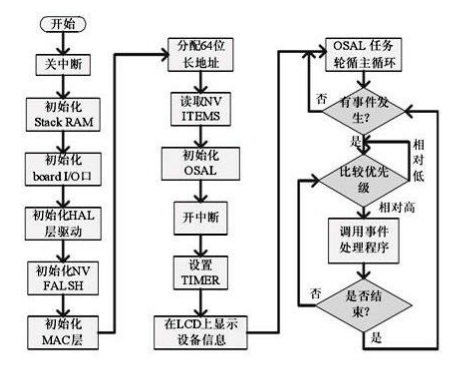


图1-3 Z-Stack软件流程图

## 1.3 IAR开发环境

硬件开发用的是Windows环境下的IAR8.10版本，IAR可以用于CC2530的开发，它是优秀的C编译器，它能够支持非常多半导体企业的芯片，例如ARM、CC2540、CC2530等芯片。它能够打开Z-Stack协议栈，在Z-Stack协议栈的配合下，可以很方便的开发出ZigBee+传感器框架的各种产品，开发软件界面如图1-4所示。

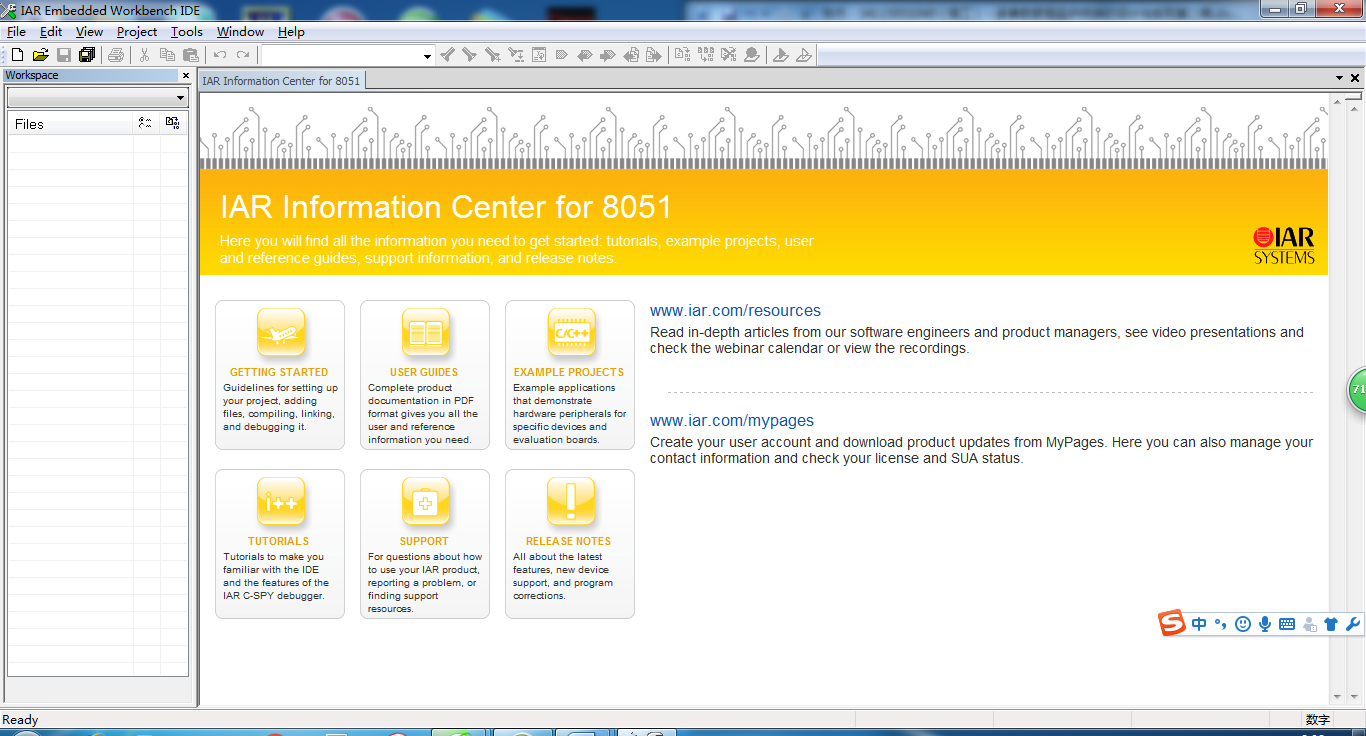


图1-4 IAR集成开发软件界面

## 1.4终端及QT图形应用技术

被经常用作终端机的主要设备分别是三种手机、电脑及嵌入式设备。首先手机不能直接和ZigBee通信，需要通过复杂的转换，且难以保证24小时不间断监护，所以本文不推荐使用手机作为终端；通过在电脑生成QT界面，直接与协调器接收数据，比较简单，因此选择用电脑作为数据显示的载体。

显示方面，本设计使用Qt图形应用界面开发技术，它最大的特点是一个平台开发，可以多个平台编译，可以十分方便的移植到嵌入式设备上去。Qt是一个面向对象的C++架构，方便扩展，被普遍应用于linux平台、Windows平台应用程序的开发上，当然，它也支持Android软件应用的开发，它的开发平台是Qt Creator。

Qt Creator是由美国某企业研发的一个多方向的C++语言图形应用界面程序的集成开发软件。它能够开发GUI（图形用户界面）程序，还能开发服务器等其他非GUI的应用。它的软件界面如图1-5所示。

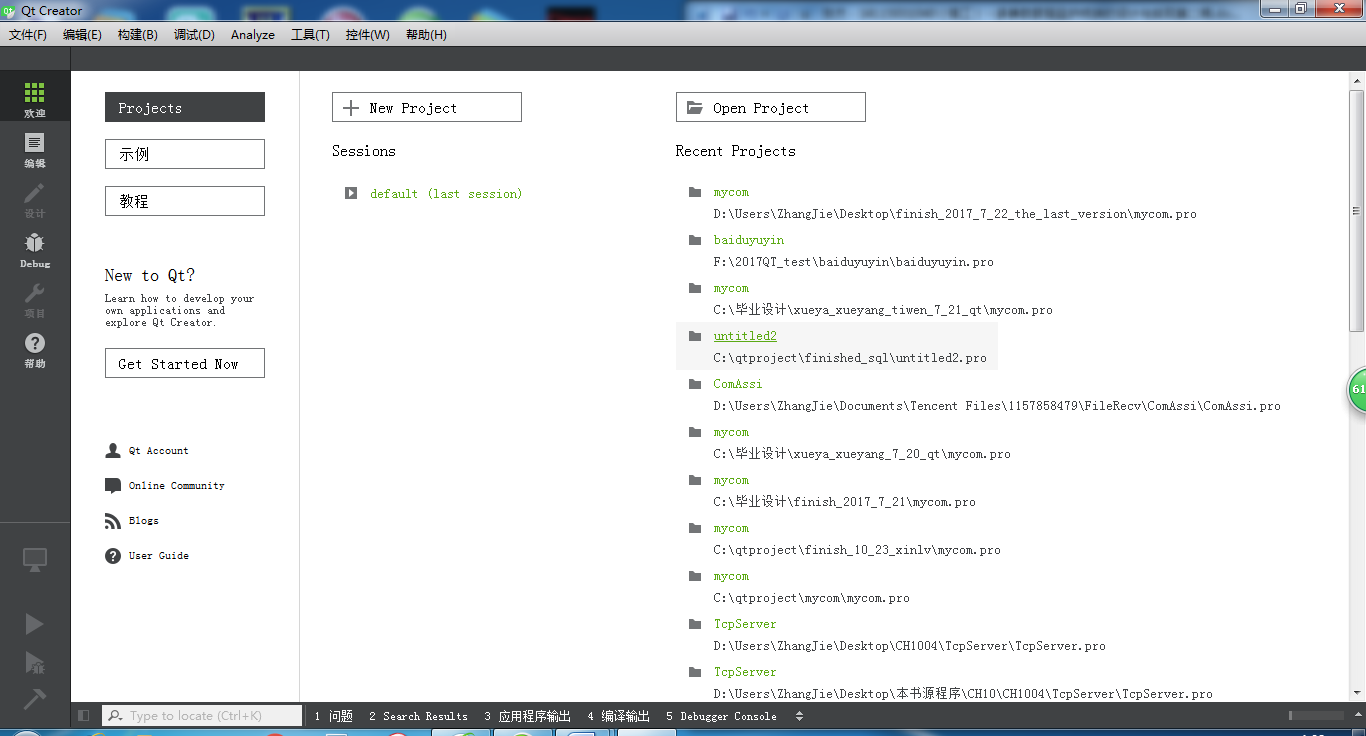


图1-5 Qt Creator集成开发环境界面

# **2 系统总体设计方案**

## 2.1功能结构分析

本设计利用温湿度传感器采集农作物、家庭周围环境的温湿度利用CC2530搭建的ZigBee无线传感网络进行数据的实时传输，所有数据统一聚合到温湿度终端上做解析、处理、存储，数据再经过局域网转发至QT界面上，再由数据库来存储。其中，数据库表格包含用户登录报表，环境温湿度报表等。

根据设计设计要求，构思出系统整体框架。温湿度传感器采集数据；采集节点进行数据的处理，并通过无线传输给协调器；协调器通过点对点的通讯方式和各个节点联系，并实时的接受各节点的数据。协调器与PC机通信，把各节点的信息发送PC端。这里由于实际情况的限制，不可能采取N多个节点；所以设计采用2个节点为例。

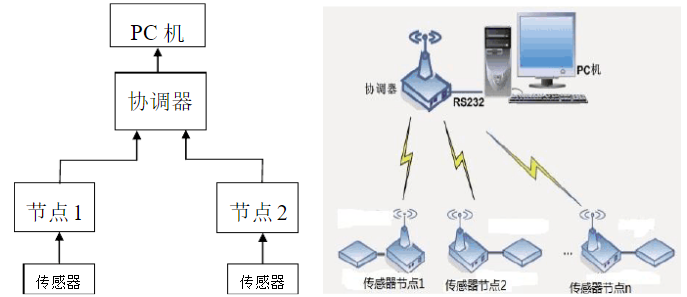


图2-1 系统功能结构框架

## 2.2技术流程

使用温湿度传感器结合CC2530模块（温湿度模块）搭建传感器节点模组，来采集周围环境各种信号，再通过cc2530模块搭建zigbee无线网络，将温湿度信号无线转发至协调器端，协调器再利用串口通信的方式有线给电脑，最终在电脑上进行数据分析处理，在嵌入式设备搭建的QT图形界面显示实时提供温湿度信息等。

## 2.3 各模块功能

### 2.3.1采集模块

系统采集模块能采集周围环境的温湿度。

DHT11温湿度传感器是一款有已校准数字信号输出的温湿度传感器。精准湿度+-5%RH，温度+-2℃，量程湿度20-90%RH，温度0-50℃。传感器包括一个电阻式感湿元件和一个NTC测温元件，并与一个高性能8位单片机相连接。因此该产品具有品质卓越、超快响应、抗干扰能力强、性价比极高等优点。每个DHT11传感器都在极为精确的温湿度检验室中进行校准，校准系数以程序的形式存在OTP内存中，传感器内部在检测型号的处理过程中要调用这些校准系数，单线制串行接口，使系统集成变得简易快捷。超小的体积、极低的功耗，使其成为最为苛刻的应用场合的最佳选择。产品为3针单排引脚封装。

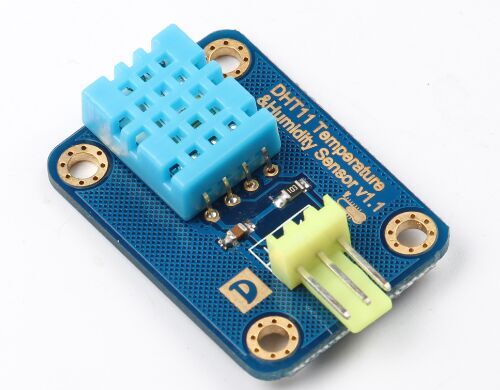


图2.1 DHT11温湿度传感器

### 2.3.2数据传输模块

数据传输部分有一个无线和一个有线模块。无线模块：ZigBee模块；

ZigBee模块是来完成温湿度检测数据的无线通信的。即建立一个ZigBee网，把所有温湿度数据进行收集，无线传送到协调器上，实现实时检测数据。

UART驱动和USB串口接收，本设计有一处使用了UART传输协议和USB串口传输，分别是传感器与CC2530的通信和协调器与PC机进行数据传输。它在本设计的作用主要是串联各种数据，使得数据到达PC端控制显示温湿度数据，然后做处理。

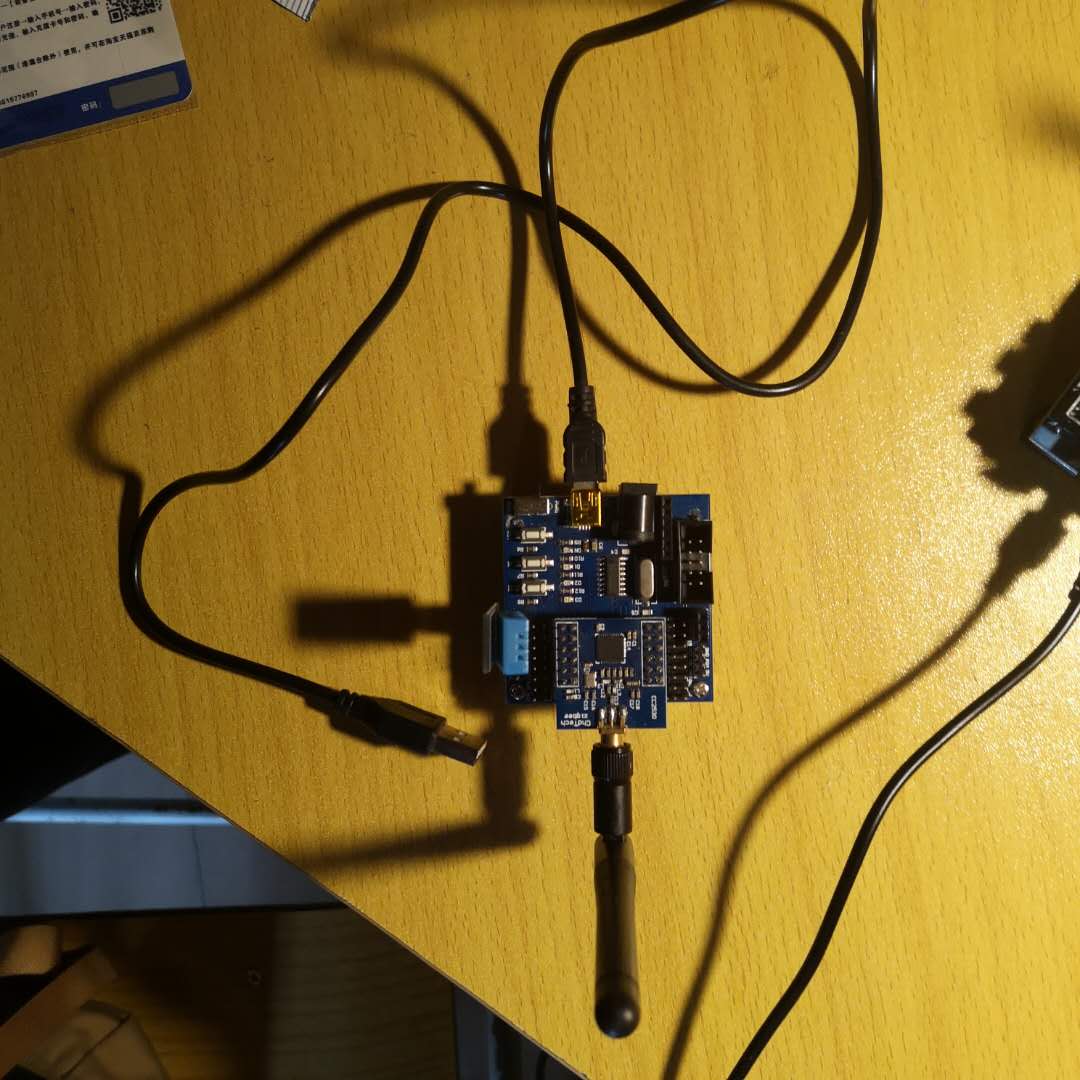


图2-2 终端模块

### 2.3.3协调器模块

协调器模块是通过Zigebee网络与终端模块进行通信，通过Zigbee协议将传感器的数据进行接收，然后在协调器通过函数处理后通过串口连接发送到PC端，具体如图所示

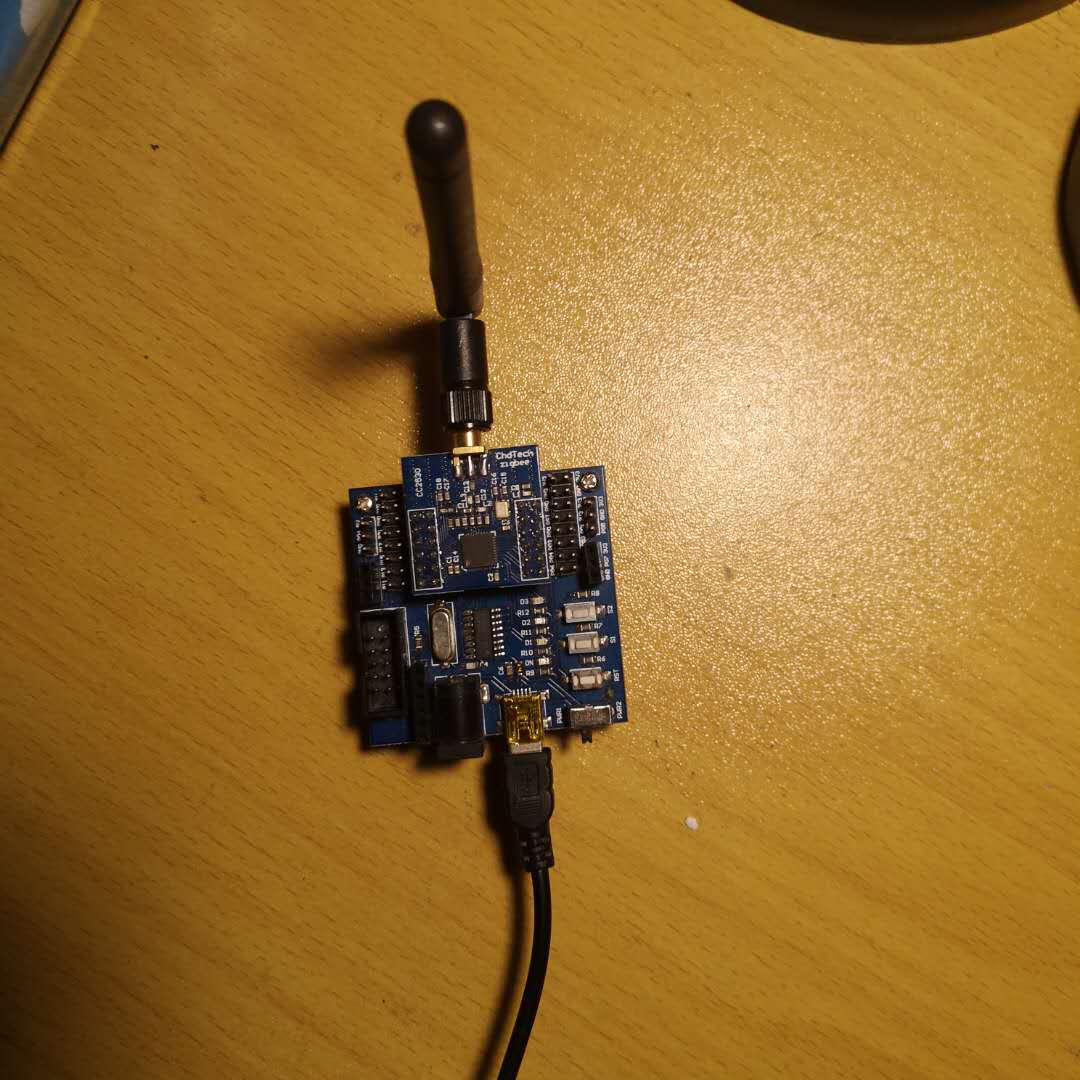


图2-3 协调器模块

### 2.3.4 终端控制显示模块

终端控制显示模块是由Qt来编写，通过虚拟机直接在PC端显示实时温湿度数据，其内容如图2-2所示。

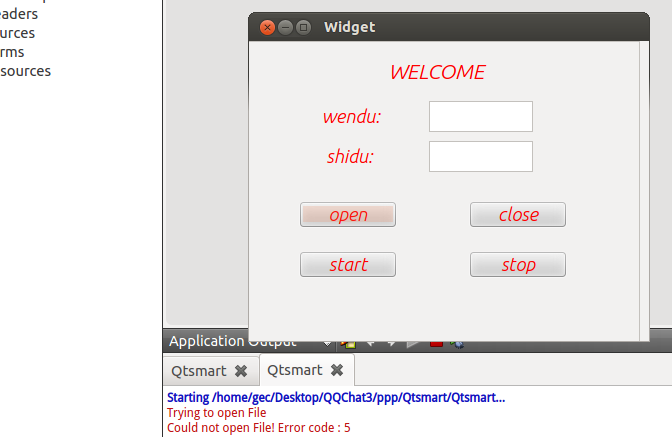


图2-4 温湿度显示界面

## 2.4 数据库设计

本设计中，用户登录信息和用户监测的温湿度数据通过IP连接然后将数据发送到数据库中

其中用户登录信息表设计如图3-5，由5个字段组成。分别是usr\_id(登录名)、usr\_pwd（用户密码）。



图2-5 数据库登陆界面



图2-6 数据库数据界面

# 3 系统软件的设计与实现

## 3.1 硬件部分

### 3.1.1 CC2530原理图

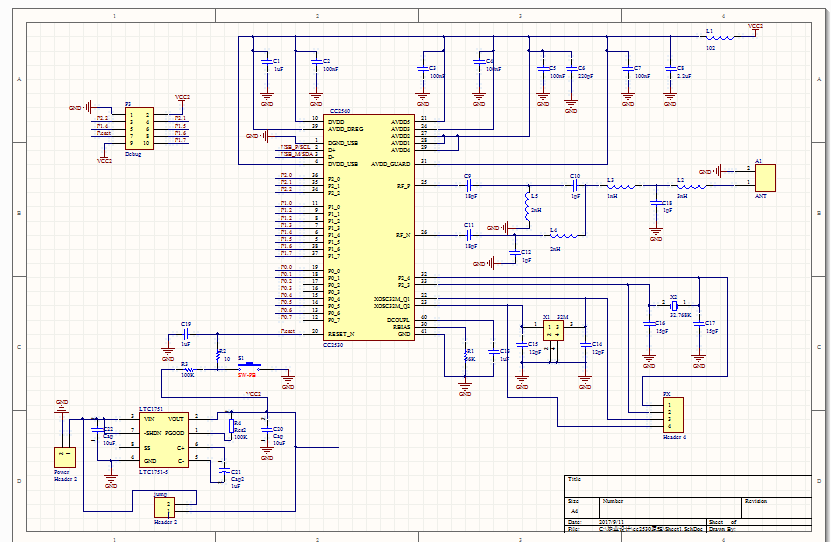


图3-1 CC2530核心板

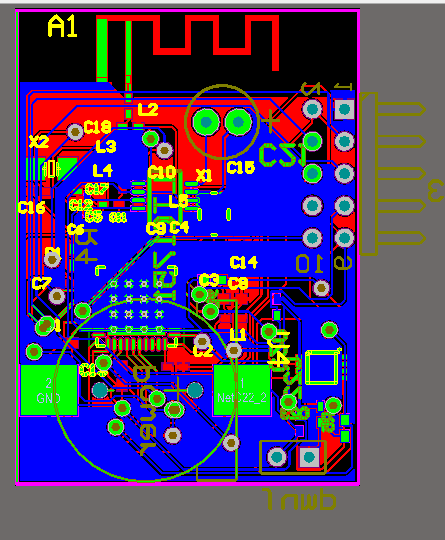


图3-2 CC2530核心板 PCB图

如图3-1 CC2530原理图，它是CC2530最小系统，其中包含了稳压模块的设计、时钟、复位、debug与射频电路。根据CC2530数据手册可知，CC2530的宽电源电压范围是2V-3.6V，本设计可以使用电源为3V的cr2032电池或5v的USB电源，分别为稳定输出电源选择了LTC1751-3.3稳压模块将电源稳定在3.3V和74LVC245将5v电源转为3.3v。其中LTC1751-3.3电压输入范围是2v-4.4v，输出电压典型值为3.3V，74LVC245最高输入电压可达5V，输出约2.7v-3.6v满足CC2530供电需求。图3-2为cc2530 PCB版图。

### 3.1.2 传感器原理图

本系统采用的是DHT11温湿度传感器，下图为传感器的类型，通过一个湿度元件和温度元件进行测量，测出了十六位的二进制的数据，通过算法将其变成十进制的数字，接着通过Zigbee发送到协调器，再由协调器通过USB串口发送到PC端

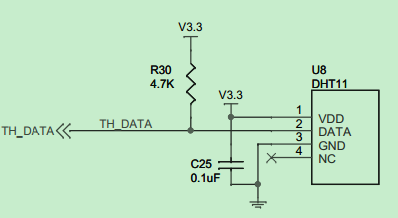


图3-3 DHT11连接示意图

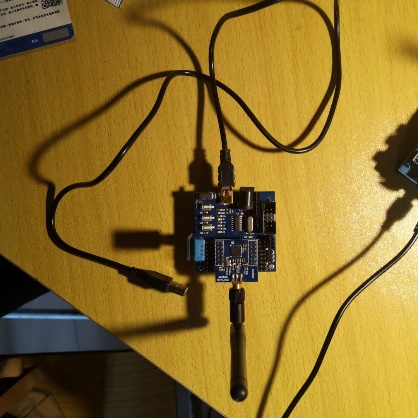


图3-4 DHT11实物连接图

## 3.2软件部分

### 3.2.1数据采集模块

任何的程序代码编写都要事先进行程序流程图的设计，本设计的下位机程序设计是

根据Z-Stack协议栈规范的，所以在程序设计时，与以往的裸板开发不一样，在理解Z-Stack协议栈的APP层的同时，加入自己的逻辑设计、应用程序部分。程序流程图如图3-5所示。

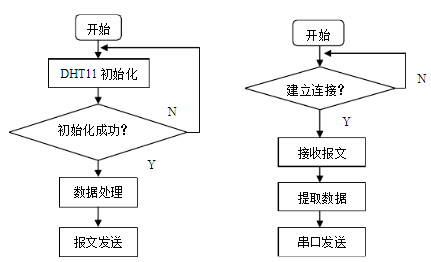


图3-5 节点发送、协调器接受程序流程图

### 3.2.2 QT图形化界面设计

本人负责的是QT图形化界面设计，所以针对QT图形化界面进行相关的介绍。

如图3-6所示，设计了一个温湿度采集的数据显示端，每隔0.9秒更新一次数据。当按下open时，开始采集数据，按下star时建立线程通信，开始发送温湿度的数据到用户端，当按下close时停止采集数据，按下stop时终止线程，停止发送数据到用户端。



图3-6 温湿度显示端

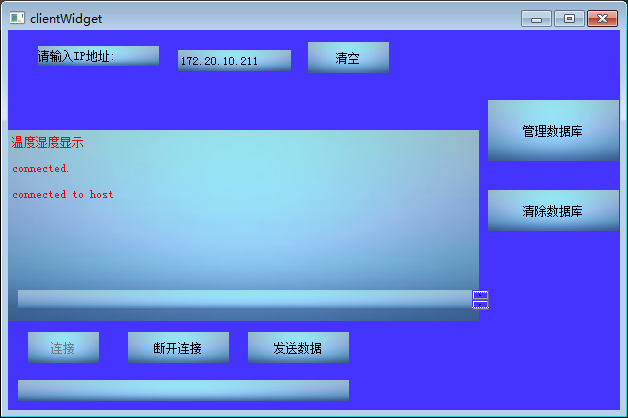
如图3-7和图3-8所示，当用户通过输入终端的IP地址实现协调器和终端的通信，连接后开始采集数据，并上传到数据库。

图3-7终端的连接界面



图3-8 数据库端显示图

部分QT串口通信的重要代码如下：

Widget::Widget(QWidget \*parent) :

QWidget(parent),

ui(new Ui::Widget)

{

ui->setupUi(this);

//connect(t,SIGNAL(ab()),this,SLOT(change()));

myCom=newPosix\_QextSerialPort("/dev/ttyUSB0",QextSerialBase::Polling);

//readTimer->start(100);

myCom->open(QIODevice::ReadWrite);

//以读写方式打开串口

myCom->setBaudRate(BAUD9600);

//波特率设置，我们设置为9600

myCom->setDataBits(DATA\_8);

//数据位设置，我们设置为8位数据位

myCom->setParity(PAR\_NONE);

//奇偶校验设置，我们设置为无校验

myCom->setStopBits(STOP\_1);

//停止位设置，我们设置为1位停止位

myCom->setFlowControl(FLOW\_OFF);

//数据流控制设置，我们设置为无数据流控制

myCom->setTimeout(10);

readTimer=new QTimer(this);

readTimer->start(1000);

connect(readTimer,SIGNAL(timeout()),this,SLOT(readMyCom()));

ui->textEdit->insertPlainText("1");

}

void Widget::change()

{

ui->startthread\_button->setEnabled(true);

ui->stopthread\_button->setEnabled(true);

}

void delay(unsigned int a)

{

unsigned int i;

while (--a!=0)

{

for(i=0;i<600;i++);

}

}

Widget::~Widget()

{

delete ui;

}

void Widget::readMyCom()//读取串口数据并显示出来

{

ui->textEdit->clear();

ui->textEdit\_2->clear();

QString temp = myCom->readAll();

//ui->textBrowser->insertPlainText(temp);

//读取串口缓冲区的所有数据给临时变量temp

int index = temp.indexOf(":");

//int index=2;

wendu= temp.mid(index-2,2);

//delay(1000);

shidu= temp.mid(index+1,2);

// delay(1000);

ui->textEdit->insertPlainText(wendu);

// ui->textEdit->insertPlainText(temp);

//delay(1000);

ui->textEdit\_2->insertPlainText(shidu);

//将串口的数据显示在窗口的文本浏览器中

/\*char sendbuf[100]={0};

QByteArray ba = temp.toLatin1();

strcpy(sendbuf,ba.data());

socket->write(sendbuf,strlen(sendbuf)); //把相关数据送入管道

socket->flush(); //防止数据较少不能自动发送，所以推一把

\*/

thread.add(temp);

delay(100);

temp.clear();

}

void Widget::on\_pushButton\_clicked()

{

}

void Widget::on\_startthread\_button\_clicked()

{

thread.start();

ui->startthread\_button->setEnabled(false);

ui->stopthread\_button->setEnabled(true);

}

void Widget::on\_stopthread\_button\_clicked()

{

if(thread.isRunning())

{

thread.stop();

ui->startthread\_button->setEnabled(true);

ui->stopthread\_button->setEnabled(true);

}

}

# **4 系统测试**

## 4.1 硬件测试

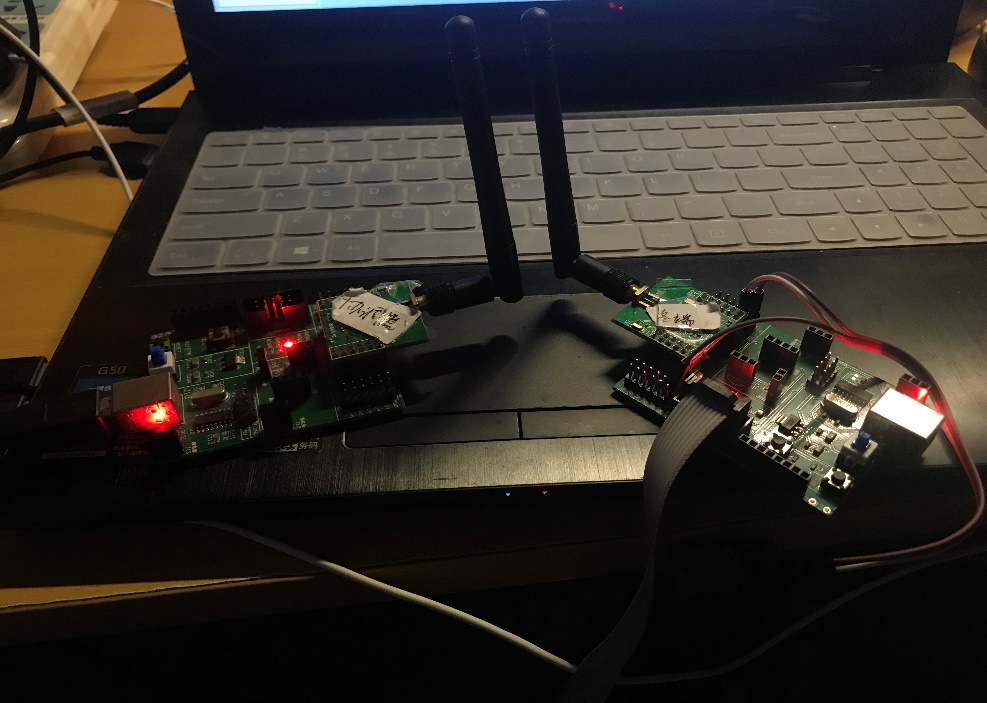


图4-1 协调器与终端进行通信

通过上图我们可以看到硬件模块成功调试并进行了通信，通信后所产生的温湿度数据通过串口发送到PC端。

## 4.2 软件测试

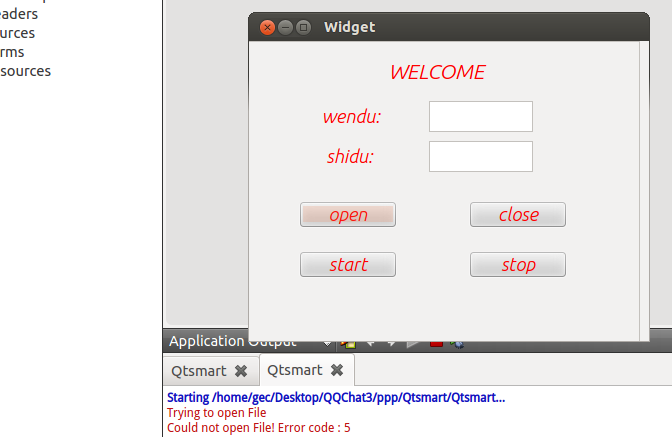


图4-2 接收数据界面



图4-3 数据库界面

经过多次试验测试，观察数据库的数据，可知系统基本符合周围环境的温度和湿度，检测结果基本准确，数据库表单等内容功能可用。

# **5 总结及展望**

## 5.1 总结

本文主要从设计需求、系统设计方案、技术方案以及系统实现，三个主要方面分析了基于Zigbee的温湿度检测系统的设计和实现，经测试得出以下总结：

1. 做项目时要系统地规划、认真的分析、要有明确的体系框架；
2. 作品的前期方案比技术更为重要，方案决定了一个设计的优劣，技术只是作为支撑；
3. 需要注意代码编写的标准性来增强软件的可读性，等；

经测试，系统框架完整，功能完善，扩展性强，功能可用。

## 5.2 展望

由于设备条件的限制，在测试过程中，检测的数据与周围环境的温湿度可能在精度、准确度上与真实结果有较大的误差，所以我希望本设计今后能进行更严谨的更准确的测试，并对其中的不足进行改进。

另外，在系统中有待改进的地方如下：

1. 使用自己设计的传感器模块代替购买的模块；
2. 增强系统功能；
3. 增加多个传感器进行检测；

# 参考文献：

**[1]唐瑜梅.基于Zigbee的温湿度数据采集与组网设计[J].电脑迷,2018(08):254-255.**

**[2]杨怀德.基于ZigBee的无线温湿度采集系统设计[J].电脑知识与技术,2016,12(14):229-231.**

**[3]邓呈轩. 基于ZigBee的远程无线温湿度测控系统的研究[D].兰州交通大学,2016.**