

**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：陈嘉俊**

**学 号：15115061015**

**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** 1](#_Toc534616438)

[1.1 ZigBee传输技术 1](#_Toc534616439)

[1.2 Z-Stack协议栈 2](#_Toc534616440)

[1.3 IAR开发环境 3](#_Toc534616441)

[1.4终端及QT图形应用技术 3](#_Toc534616442)

[1.5数据库技术 4](#_Toc534616443)

[**2 系统总体设计方案** 5](#_Toc534616444)

[2.1功能结构分析 5](#_Toc534616445)

[2.2技术流程 5](#_Toc534616446)

[2.3 各模块功能 6](#_Toc534616447)

[2.3.1采集模块 6](#_Toc534616448)

[2.3.2数据传输模块 6](#_Toc534616449)

[2.3.3协调器模块 7](#_Toc534616450)

[2.3.4 终端控制显示模块 8](#_Toc534616451)

[2.4 数据库设计 9](#_Toc534616452)

[**3 数据库的设计与实现** 10](#_Toc534616453)

[3.1 调用数据库 10](#_Toc534616454)

[3.2设按钮槽函数 10](#_Toc534616455)

[3.3 读取管道数据并录入数据库 11](#_Toc534616456)

[3.4 读取数据库 12](#_Toc534616457)

[**4 系统测试** 13](#_Toc534616458)

[4.1 硬件测试 13](#_Toc534616459)

[4.2 软件测试 14](#_Toc534616460)

[**5 总结及展望** 15](#_Toc534616461)

[5.1 总结 15](#_Toc534616462)

[5.2 展望 15](#_Toc534616463)

**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。

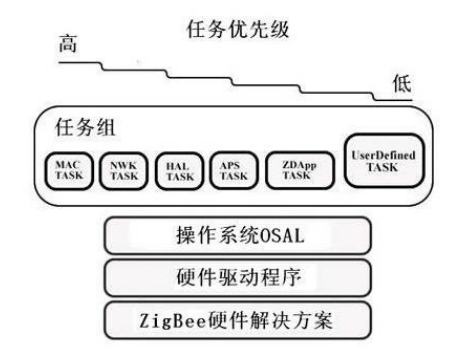


图1-2 Z-Stack软件架构

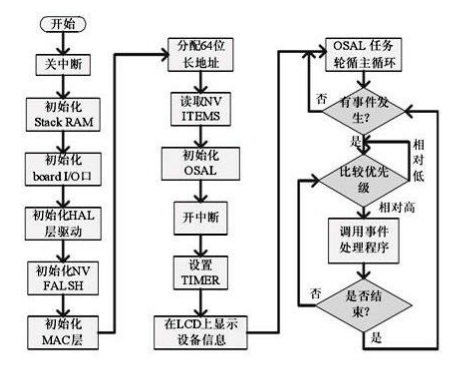


图1-3 Z-Stack软件流程图

## 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。

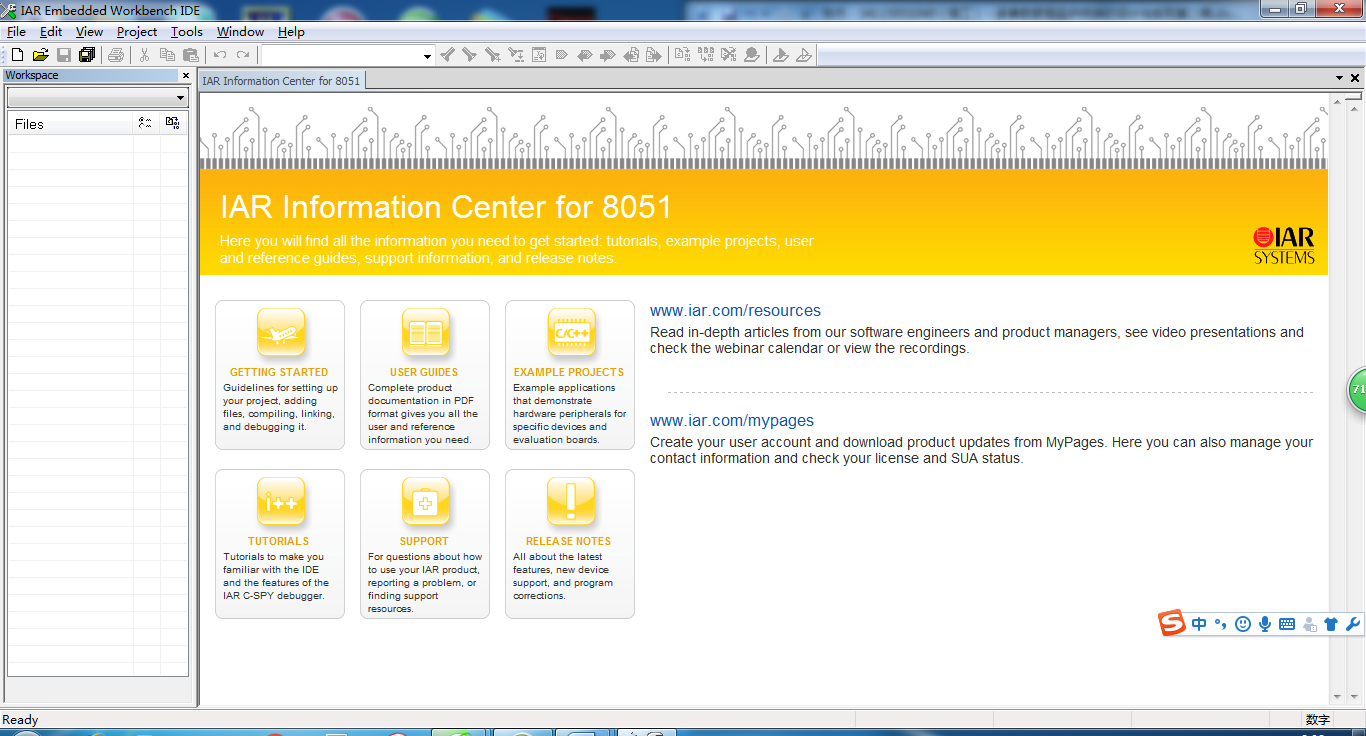


图1-4 IAR集成开发软件界面

## 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。

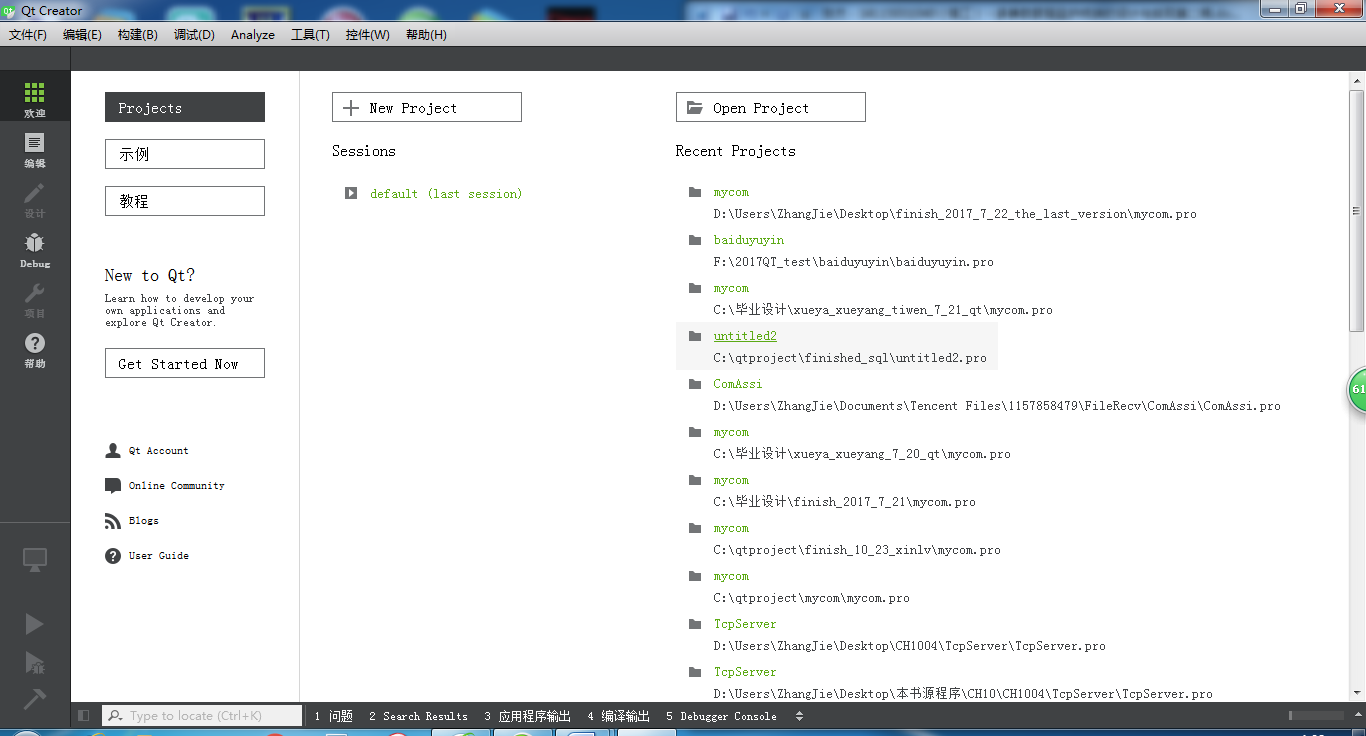


图1-5 Qt Creator集成开发环境界面

## 1.5数据库技术

qt中含有一个自带的数据库，本次课程设计通过QSqlDatabase调用的是qt自身的数据库。QSqlDatabase类提供一个通过数据库连接访问数据库的接口。一个QSqlDatabase的实例代表了一个数据库连接

用静态方法 addDatabase() 创建一个数据库连接(也就是QSqlDatabase的实例)，指定驱动或者驱动种类(也就是你要访问什么类型数据库)以及连接名称。QSqlDatabase还支持默认连接概念，没有连接名称的连接就是默认连接。要创建默认连接，在调用addDatabase()方法时不传递连接名称参数(connectionName)即可。

调用 database() 方法来获得连接：

QSqlDatabase db = QSqlDatabase::database();

如果创建多重数据库连接，调用 addDatabase() 时，要为每个连接指定独一无二的名字。调用 database() 方法并传递连接名称来获取该连接。通过调用带有连接名称参数的 removeDatabase() 方法来移除一个连接。当试图移除一个已经被其他 QSqlDatabase 对象引用的连接时 QSqlDatabase 将会输出警告。调用contains() 方法来查看给出的连接名称是否在连接列表中

# **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。

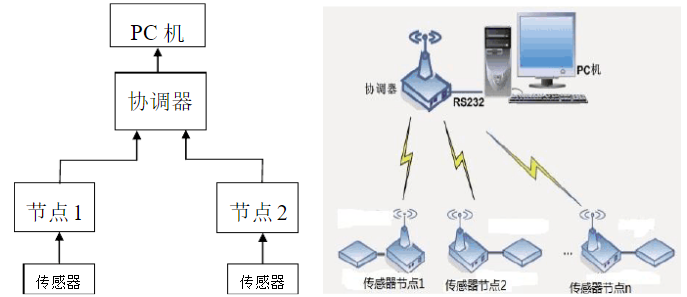


图2-1 系统功能结构框架

## 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。

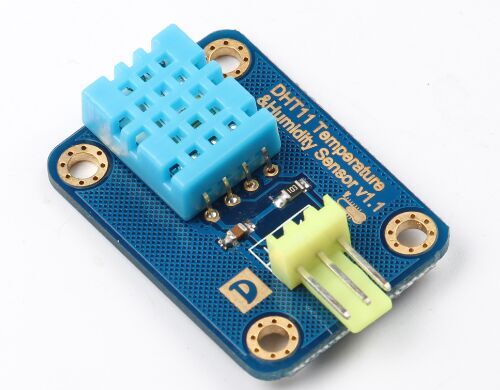


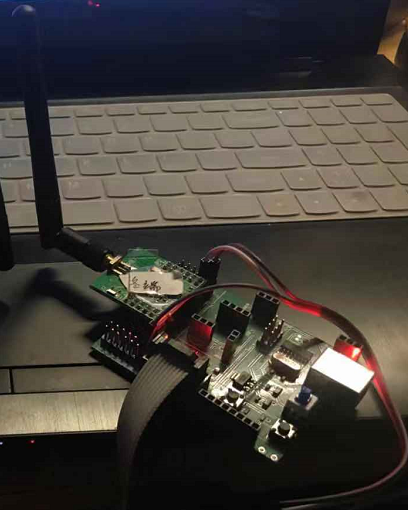
图2.1 DHT11温湿度传感器

### 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。

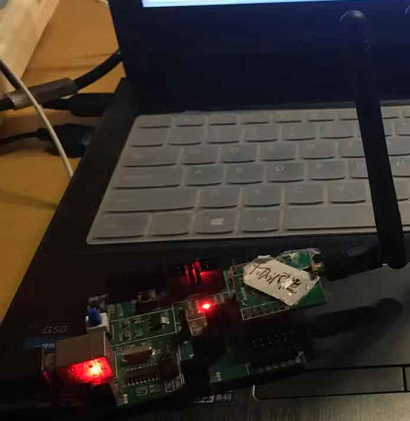
UART驱动和USB串口接收，本设计有一处使用了UART传输协议和USB串口传输，分别是传感器与CC2530的通信和协调器与PC机进行数据传输。它在本设计的作用主要是串联各种数据，使得数据到达PC端控制显示温湿度数据，然后做处理。



终端模块

### 2.3.3协调器模块

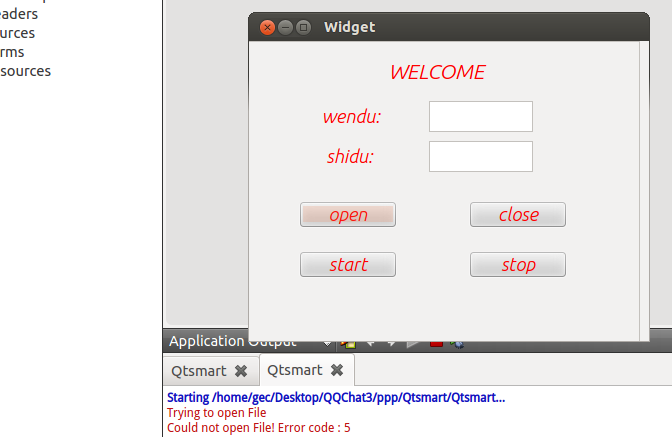
协调器模块是通过Zigebee网络与终端模块进行通信，通过Zigbee协议将传感器的数据进行接收，然后在协调器通过函数处理后通过串口连接发送到PC端，具体如图所示



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。



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，由两部分组成。分别是usr\_id(用户名)、usr\_pwd（密码）。



数据库登陆界面

数据显示表设计如图，第一列为日期，第二列为检测时间，第三列为测得的温度，第四列为检测的到的湿度。其中还含有清空数据功能。



数据库数据界面

# **3 数据库的设计****与实现**

通过调用qt中的数据库，进行录入和显示。

## 3.1 调用数据库

QSqlDatabase类提供一个通过数据库连接访问数据库的接口。调用 database() 方法来连接：QSqlDatabase db = QSqlDatabase::database();代码如下：

static bool createConnection()

{ qDebug()<<"aa";

QSqlDatabase db = QSqlDatabase::addDatabase("QSQLITE");

qDebug()<<"bb";

db.setDatabaseName("ss.db");

qDebug()<<"cc";

if (!db.open()) {

QMessageBox::critical(0, QObject::tr("无法打开数据库"),

"无法创建数据库连接！", QMessageBox::Cancel);

qDebug()<<"dd";

return false;

} QSqlQuery query;

## 设置按钮槽函数

将相关按钮设置相应的槽函数

代码如下：

clientWidget::clientWidget(QWidget \*parent) :

QWidget(parent),

ui(new Ui::clientWidget)

{

ui->setupUi(this);

socket = new QTcpSocket(this);

connect(ui->pushButton\_connect,SIGNAL(clicked()),this,SLOT(onConnect())); //点击相关按钮触发相关槽函数

connect(ui->pushButton\_disconnect,SIGNAL(clicked()),this,SLOT(onDisconnect()));

connect(ui->pushButton\_send,SIGNAL(clicked()),this,SLOT(onSend()));

connect(socket,SIGNAL(connected()),this,SLOT(imcoming())); //管道建立后触发相关函数

ui->pushButton\_send->setShortcut(tr("Ctrl+R")); //设置发送相关快捷键

}

## 3.3 读取管道数据并录入数据库

读取管道中的数据并放入数据库中

代码如下：

void clientWidget::onread()

{

QSqlQuery query;

char bufclient[100]; //缓冲区

char w[5];

char s[5];

QString da;

QString tt;

memset(bufclient,0,sizeof(bufclient)); //清空缓冲区

int lengthc = socket->bytesAvailable(); //从管道中计算传送数据长度

if(lengthc>0)

{

socket->read(bufclient,lengthc); //把指定长度数据从管道中放入缓冲区buf

ui->textEdit->insertPlainText("\n");

ui->textEdit->insertPlainText("server : "); //显示相关发信息人，以及时间等

ui->textEdit->insertPlainText(QDateTime::currentDateTime().toString("yyyy-MM-dd "));

ui->textEdit->insertPlainText(QTime::currentTime().toString("hh:mm:ss"));//textEdit文本框显示当前时间

ui->textEdit->insertPlainText("\n");

ui->textEdit->append(bufclient); //显示信息

ui->textEdit->insertPlainText("\n");

da=QDateTime::currentDateTime().toString("yyyy-MM-dd");

tt=QTime::currentTime().toString("hh:mm:ss");

memset(w,0,sizeof(w));

memset(s,0,sizeof(s));

w[0]=bufclient[0];

w[1]=bufclient[1];

s[0]=bufclient[3];

s[1]=bufclient[4];

query.exec(QString("insert into type values('%1','%2','%3','%4')").arg(da).arg(tt).arg(w).arg(s));

}

## 3.4 读取数据库

将数据库中的数据读出

代码如下：

Dialog::Dialog(QWidget \*parent) :

QDialog(parent),

ui(new Ui::Dialog)

{

ui->setupUi(this);

tableView=new QTableView(this);

}

Dialog::~Dialog()

{

delete ui;

}

void Dialog::on\_pushButton\_clicked()

{

int r=0;

QSqlQuery query;

query.exec("select \* from type");

QStandardItemModel \*model=new QStandardItemModel(sum,4,this);

while(query.next())

{

QStandardItem \*item=new QStandardItem(query.value(0).toString());

model->setItem(r,0,item);

QStandardItem \*item1=new QStandardItem(query.value(1).toString());

model->setItem(r,1,item1);

QStandardItem \*item2=new QStandardItem(query.value(2).toString());

model->setItem(r,2,item2);

QStandardItem \*item3=new QStandardItem(query.value(3).toString());

model->setItem(r,3,item3);

r++;

}

tableView->resize(400,300);

tableView->setModel(model);

tableView->show();

sum=0;

}

# **4 系统测试**

## 4.1 硬件测试

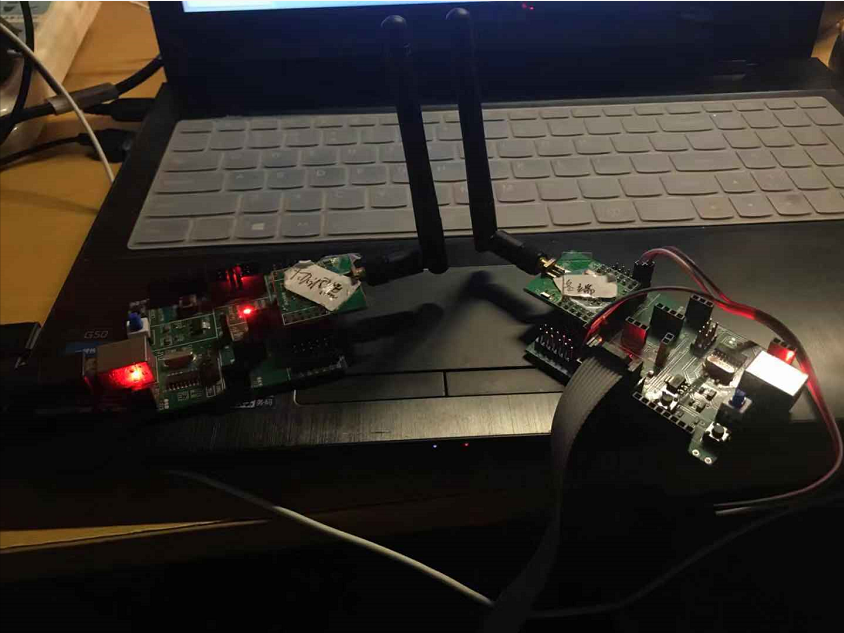


图4-1 协调器与终端进行通信

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

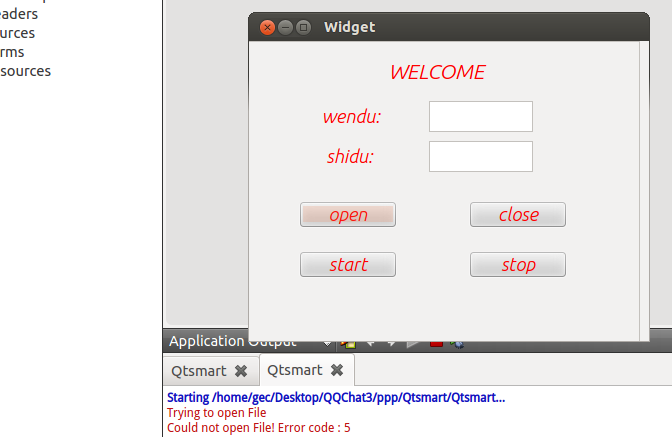


图4-2 接收数据界面



图4-3 数据库界面

经过多次试验测试，数据库内的信息都较为符合环境的实际情况，因此系统的功能基本正确实现，检测结果可用，符合实际。课程设计达到应有的要求。

# **5 总结及展望**

## 5.1 总结

通过本次课程设计，学习和了解基于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. 做项目前需做好准备，提前寻找相关资料，了解什么技术可以完成设计提前准备，哪些部分可以通过调用节省开发资源。
2. 设计的前期工作比技术更为重要，有一个好的准备和方案可以快速让自己掌握工作流程和各细节
3. 注意代码编写的标准性来增强软件的可读性，

## 5.2 展望

由于技术，设备条件的限制，本次设计较为简单。功能不齐全，测试效果有误差等。希望通过本设计，今后能完善自己的不足，不断的改进自己的技术。

# 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.**