**课 程 设 计**

**课 程：物联网应用系统开发实践**

**题 目：基于ZigBee的环境温湿度监测系统设计**

**学生姓名：胡辉**

**学 号：15115061050**

**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 11 月—— 2018年 12月**

（教务处制）

**基于ZigBee的环境温湿度监测系统设计**

**摘 要**：本设计主要针对环境的温湿度进行监测，在传统设备的功能基础之上进行改进和创新。温湿度传感器模块检测数据与协调器无线组网，协调器通过串口通信把数据上传到服务器端，服务器端通过TCP通信把数据传送给用户，用户可以在数据库中查看接收数据以来各个时段的温湿度数据。数据传输模块包含ZigBee传输，串口传输，socket。

传感器终端通过单播的形式与协调器联网，监控终端通过建socket通道与用户端进行通信。

在连接数据库方面，本系统用到了Qt自带的一个数据库，也就是Sqlite数据库，用户可以在系统中查看数据库数据，也可以在独立的Sqlite数据库对温湿度数据进行查看。

系统用到的技术主要有ZigBee无线组网，串口通信，TCP/IP通信，数据库的连接管理等。

**关键字**： ZigBee；嵌入式；温湿度监测；QT

**目录**

[1 基于ZigBee的环境温湿度监测系统设计相关技术 1](#_Toc497056372)

[1.1 ZigBee传输技术 1](#_Toc497056373)

[1.2 Z-Stack协议栈 2](#_Toc497056374)

[1.3 IAR开发环境 3](#_Toc497056375)

[1.4终端及QT图形应用技术 3](#_Toc497056376)

[1.5 Sqlite数据库技术 4](#_Toc497056377)

[1.6 TCP/IP协议 5](#_Toc497056377)

[2 系统总体设计方案 6](#_Toc497056378)

[2.1功能结构分析 6](#_Toc497056379)

[2.2技术流程 7](#_Toc497056380)

[2.3 各模块功能 7](#_Toc497056381)

[2.3.1采集模块 7](#_Toc497056382)

[2.3.2协调器模块 8](#_Toc497056383)

[2.3.3数据传输模块 8](#_Toc497056384)

[2.3.4终端控制显示模块 8](#_Toc497056385)

[2.4 数据库设计 8](#_Toc497056386)

[3 系统软件设计与实现 9](#_Toc497056388)

[3.1 硬件部分 9](#_Toc497056389)

[3.1.1 CC2530原理图 9](#_Toc497056390)

[3.1.2 cortex-A8原理图 10](#_Toc497056391)

[3.1.3 传感器模块 10](#_Toc497056391)

[3.2 软件部分 11](#_Toc497056392)

[3.2.1 数据采集模块 1](#_Toc497056393)1

[3.2.2 联网 1](#_Toc497056394)1

[3.2.3软件移植 1](#_Toc497056395)3

[3.2.4连接数据库 1](#_Toc497056396)4

[4系统测试 1](#_Toc497056400)7

[5 总结及展望 2](#_Toc497056403)0

[5.1 总结 2](#_Toc497056404)0

[5.2 展望 2](#_Toc497056404)0

# 

# **1 基于ZigBee的环境温湿度监测系统设计相关技术**

## 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特点开发的温湿度监控，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，与它相连接的是监控终端（即嵌入式设备）。在本系统中，用了一个节点（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。这个节点时温湿度检测模块，协调器完成数据的汇总以及与监控终端的数据交互。



**协调器**

**终端节点**

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。

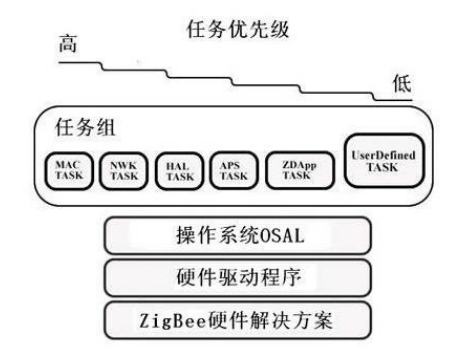


图1-2 Z-Stack软件架构

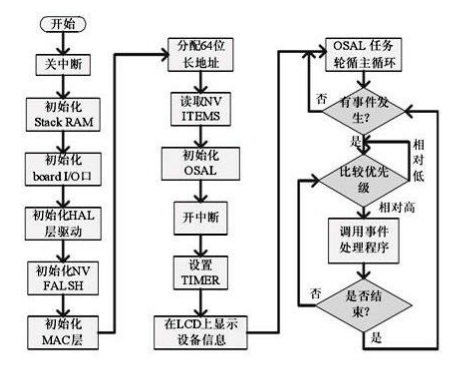


图1-3 Z-Stack软件流程图

## 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。

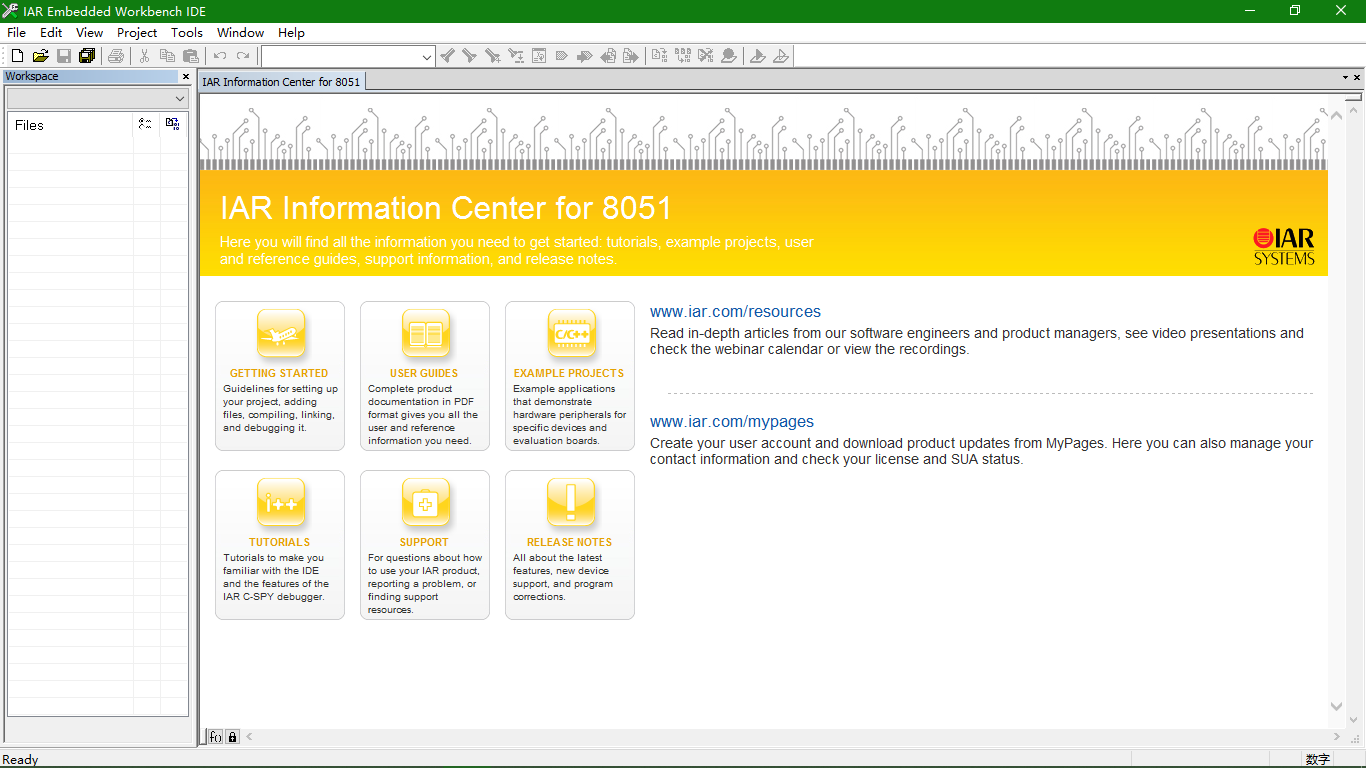


图1-4 IAR集成开发软件界面

## 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，嵌入式设备具有软硬件可裁剪、价格、质量、耗能可控等特点，用它作为终端再适合不过了，因此本设计采用cortex-A8为核心的嵌入式板作为终端设备，配置7寸的电容屏，提供用户交互的功能。

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。

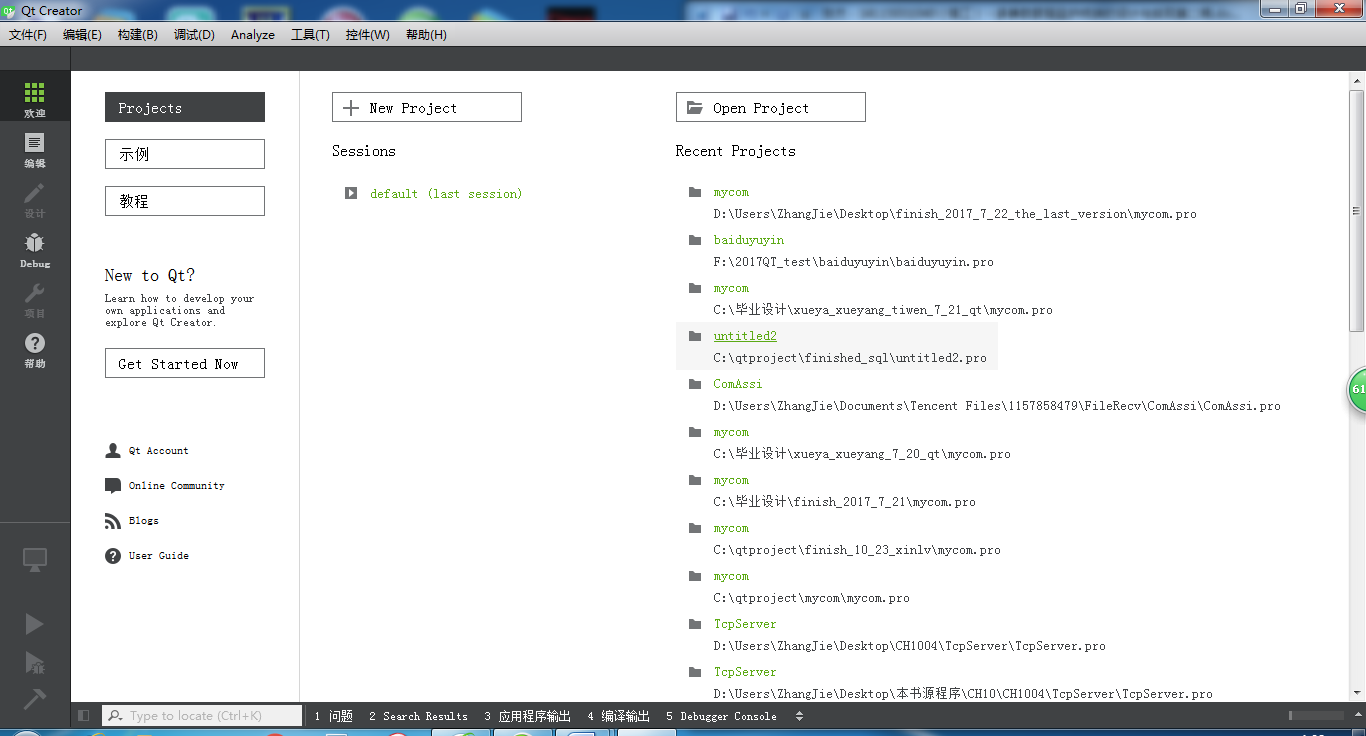


图1-5 Qt Creator集成开发环境界面

## 1.5 Sqlite数据库技术

SQLite由以下几个部分组成：SQL编译器、内核、后端以及附件。SQLite通过利用虚拟机和虚拟数据库引擎(VDBE)，是调试、修改和扩展SQLite的内核变得更加方便。所有SQL语句都被编译成易读的、可以在SQLite虚拟机中执行的程序集。

本设计使用的是Qt本身自带的数据库Sqlite，利用Qt搭建Sqlite数据库，用来简单存储ZigBee节点检测到的温湿度数据。

SQLite 是目前最流行的开源嵌入式数据库，和很多其他嵌入式存储引擎相比（NoSQL），如 BerkeleyDB、MemBASE 等，SQLite 可以很好的支持关系型数据库所具备的一些基本特征，如标准 SQL 语法、实务、数据表和索引等。事实上，尽管 SQLite 拥有诸多关系型数据库的基本特征，然而由于应用场景的不同，它们之间并没有更多的可比性。下面是 SQLite 的主要特征：

1). 管理简单，甚至可以认为无需管理。

2). 操作方便，SQLite 生成的数据库文件可以在各个平台无缝移植。

3). 可以非常方便的以多种形式嵌入到其他应用程序中，如静态库、动态库等。

4). 易于维护。

综上所述，SQLite 的主要优势在与灵巧、快速和可靠性高。SQLite 的设计者为了达到这一目标，在功能上作出了很关键性的取舍，与此同时，也失去了一些对 RDBMS 关键性功能的支持，如高并发、细粒度访问控制（如行级锁）、丰富的内置函数、存储过程和复杂的 SQL 语句等。正式因为这些功能的牺牲才换来了简单，而简单又换来了高效性和高可靠性。

## 1.6 TCP/IP协议

TCP/IP协议，即[Transmission](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=66303746&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank) Control Protocol/Internet Protocol的简写，中译名为传输控制协议/因特网互联协议，又名[网络通讯协议](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=58734906&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)，是Internet最基本的协议、Internet[国际互联网络](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=101669383&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)的基础，由[网络层](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=16964&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)的IP协议和[传输层](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=16962&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)的[TCP协议](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=20113&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)组成。TCP/IP 定义了电子设备如何连入因特网，以及数据如何在它们之间传输的标准。协议采用了4层的层级结构，每一层都呼叫它的下一层所提供的协议来完成自己的需求。通俗而言：TCP负责发现传输的问题，一有问题就发出信号，要求重新传输，直到所有数据安全正确地传输到目的地。而IP是给因特网的每一台联网设备规定一个地址。如图1-6所示。

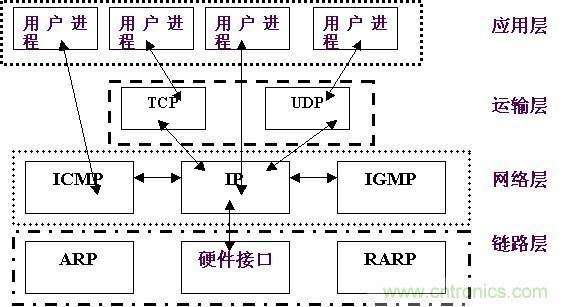
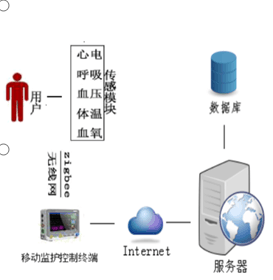


图1-6 TCP/IP协议示意图

1. TCP是[面向连接的](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=144173360&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)通信协议，通过[三次握手](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=7720717&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)建立连接，通讯完成时要拆除连接，由于TCP是面向连接的所以只能用于[端到端](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=37218110&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)的通讯。
2. TCP提供的是一种可靠的 [数据流](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=319177&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)服务，采用“带重传的肯定确认”技术来实现传输的可靠性。TCP还采用一种称为“[滑动窗口](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=7933119&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)”的方式进行流量控制，所谓窗口实际表示接收能力，用以限制发送方的发送速度。
3. 如果IP数据包中有已经封好的TCP数据包，那么IP将把它们向‘上’传送到TCP层。TCP将包排序并进行错误检查，同时实现[虚电路](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=35391&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)间的连接。TCP数据包中包括序号和确认，所以未按照顺序收到的包可以被排序，而损坏的包可以被重传。
4. TCP将它的信息送到更高层的应用程序，例如Telnet的服务程序和客户程序。应用程序轮流将信息送回TCP层，TCP层便将它们向下传送到IP层，设备驱动程序和物理介质，最后到接收方。
5. 面向连接的服务（例如 Telnet、 FTP、 rlogin、 X Windows和 SMTP）需要高度的可靠性，所以它们使用了TCP。DNS在某些情况下使用TCP（发送和接收 域名数据库），但使用UDP传送有关单个主机的信息。

# **2 系统总体设计方案**

## 2.1功能结构分析



用户

图2-1 功能结构框图

如图2-1功能结构框图所示，本设计利用温湿度传感器采集环境实时数据，利用CC2530搭建的ZigBee无线传感网络进行数据的实时传输，数据统一聚合到监护控制终端上做解析、处理、临时显示，数据再经过局域网转发至PC端的Sqlite数据库来存储，用户可以通过PC端对上传的数据进行查看，操作。

监控端搭载Cortex-A8 CPU，主要职责是近距离的无线监护的功能，本功能主要针对缺点为干扰用户活动、不能远距离操控等的问题。本部分利用Qt编写终端图形应用程序，然后将应用程序移植到linux系统的终端上运行。其他功能还有:传感器数据的显示、监控局域网中其他网络的接入等。

## 2.2技术流程

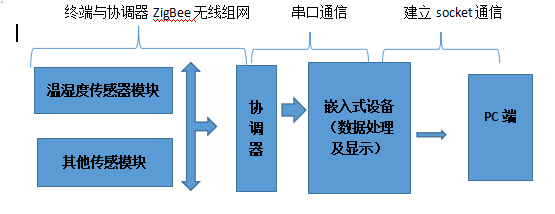


图2-2 实现流程

如图2-2实现流程所示，温湿度传感器结合ZigBee模块与协调器进行无线组网，传感器的数据通过无线网络传送到协调器，协调器再通过串口进行串口通信把数据传送到嵌入式设备上显示嵌入式设备上设置了网络的监听，只有开启了监听，局域网内的其他网络才可以接入，PC端进行数据的接受处理，把数据存储到Sqlite数据库。

## 2.3 各模块功能

### 2.3.1采集模块

系统采集模块也就是温湿度传感器结合ZigBee模块，负责采集环境的温湿度。温湿度传感器模块把数据通过与协调器无线组网，把数据上传。

### 2.3.2协调器模块

协调器模块接收负责接收终端节点上传的数据，通过串口通信把数据传送到嵌入式设备。

### 2.3.3数据传输模块

数据传输部分有一个无线和两个有线模块。无线模块：ZigBee模块；有线模块：UART驱动。

ZigBee模块是来完成温湿度采集与协调器的无线通信的。即建立一个ZigBee网，把采集的数据，无线传送到协调器上，实现无线监护。

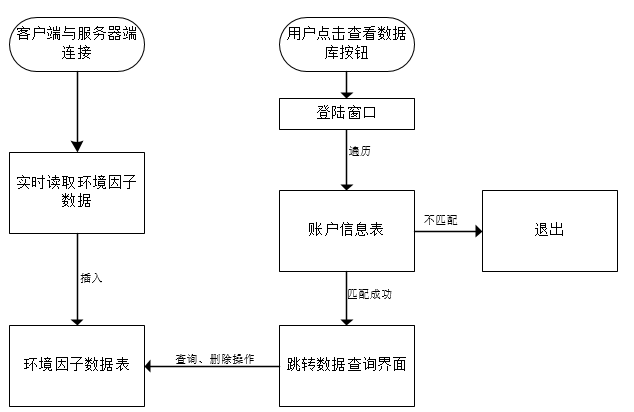
UART驱动，本设计有两处使用了UART传输协议，分别是传感器与CC2530的通信和协调器与Cortex-A8核心板之间的通信。它在本设计的作用主要是使得数据到达终端控制显示模块（Cortex-A8核心板），然后做处理。

### 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，然后移植到Cortex-A8控制的嵌入式板上去，由7寸LCD显示屏显示出来。

## 2.4 数据库设计

本设计用到了Qt自带的一个数据库Sqlite，数据库时刻记录通过联网上传的环境因子数据。数据库需要用户登录才可以查看管理。

**3 系统软件设计****与实现**

## 3.1 硬件部分

### 3.1.1 CC2530原理图

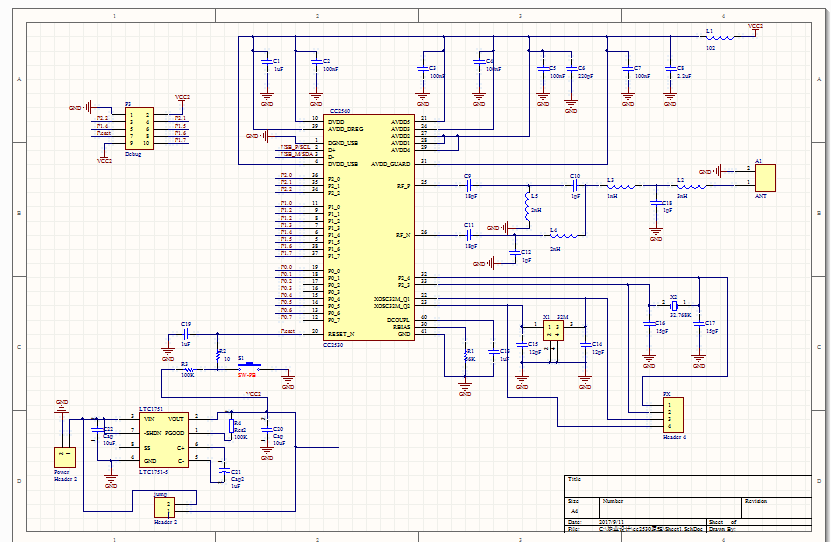


图3-1 CC2530核心板

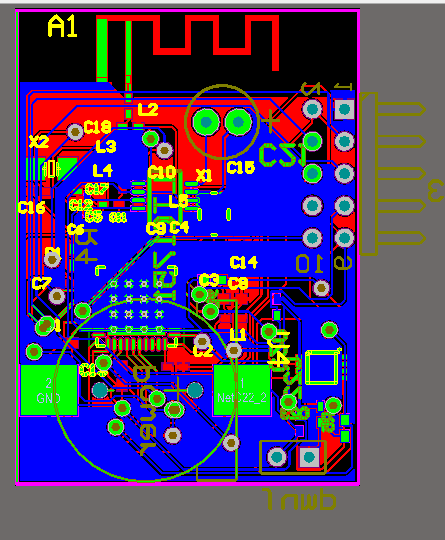


图3-2 CC2530核心板 PCB图

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 cortex-A8原理图

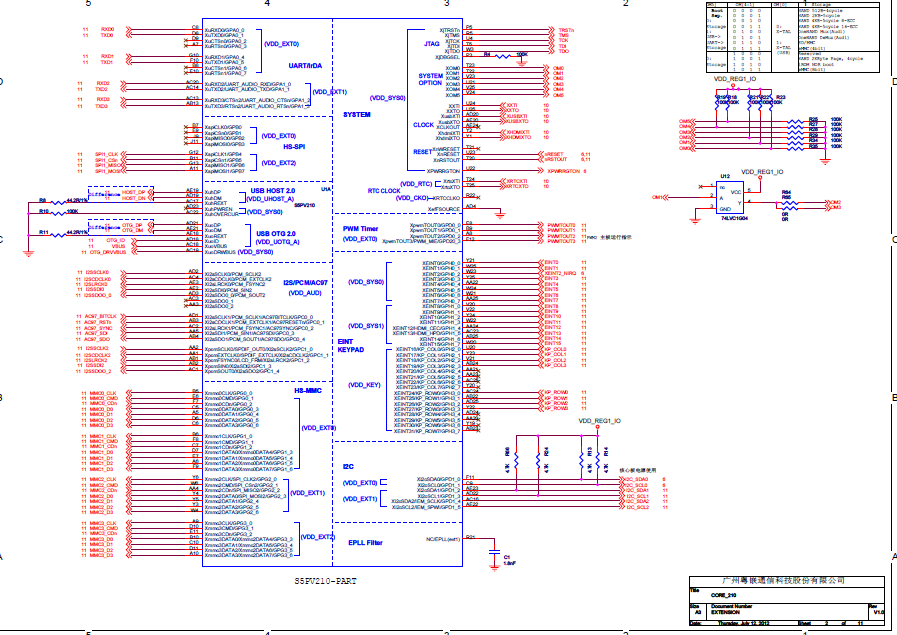


图3-3 Cortex-A8核心板

ARM Cortex-A8是ARMv7架构的一款应用型的CPU，已常用于智能智慧终端的开发中做主控芯片，它能用在十分恶劣复杂的环境中。另外CC2530模块与Cortex-A8之间的采用UART通信协议，即通过串口通信的方式，即把CC2530的P0.2、P0.3引脚对应与Cortex-A8的RX、TX相连便可。

3.1.3 传感器模块

DHT11温湿度传感器模块,DHT11是一款有已校准数字信号输出的[温湿度传感器](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=7810593&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)。 精度湿度+-5%RH， 温度+-2℃，量程湿度20-90%RH， 温度0~50℃。

如图3-4所示。

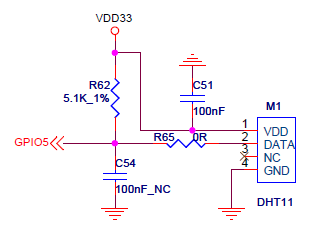


图3-4 DHT11温湿度传感器原理图

## 3.2 软件部分

### 3.2.1 数据采集模块

温湿度传感器模块通过单播的形式与协调器进行无线组网，协调器接收终端发送过来的数据，通过串口通信的形式写入到串口。

### 3.2.2 联网

在虚拟机部分用Qt编写可以烧写进入终端控制显示模块的源码。

首先编写Qt串口程序，设置好协调器上传数据所需要的波特率、数据位、串口号等。只要协调器接收到温湿度采集模块发送过来的数据，通过串口程序，数据就会通过Qt界面形象化的显示出来，让用户更容易发现环境中温湿度的变化。在虚拟机上运行的串口程序有着特定的串口类需要人为进行加载，这串口类就是QextSerialBase，Posix\_QextSerialPort。Qt串口程序第一步是要以自定义的方式打开串口，接着就是读串口的数据，程序中设置的是没1秒钟读取一次。

接着就是，串口接收到的数据通过在线程中创建socket，利用TCP的通信协议创建socket对象server，服务器端负责监听其他网络的接入，所以在这里设置了端口监听，只要检测到有网络请求接入立即接受。程序中是利用线程的运行态和死亡态对socket进行控制，断开线程，另一端的网络将不再接收数据。如图3-5所示。

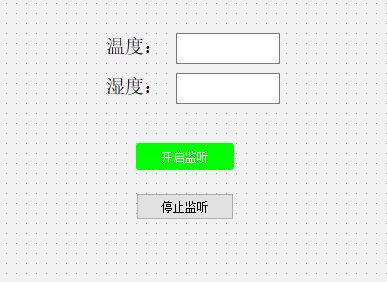


图3-5 服务器端界面

server = new QTcpServer(this);//创建QTcpSever对象server

server->listen(QHostAddress::Any,9090);//监听端口

void MyThread::stop()

{

socket->abort(); //让管道失效

}

在PC端，一样是利用Qt来编写用户端程序，这一程序负责连接服务器端，服务器端的程序的数据只要与用户端设置的IP在同一网段将会被用户端接收并且显示。如图3-6所示。

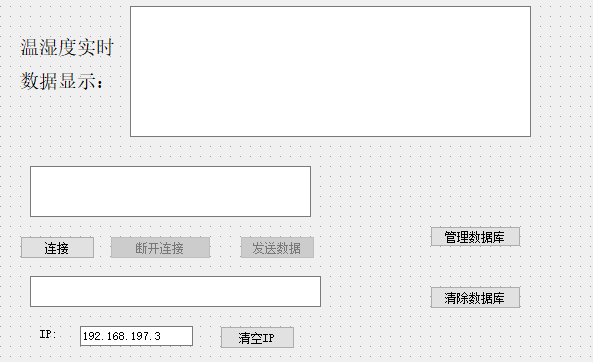


图3-6 用户端界面

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

socket->*connectToHost*(QString(ui->lineEdit\_socket->text()),9090);

### 3.2.3 软件移植

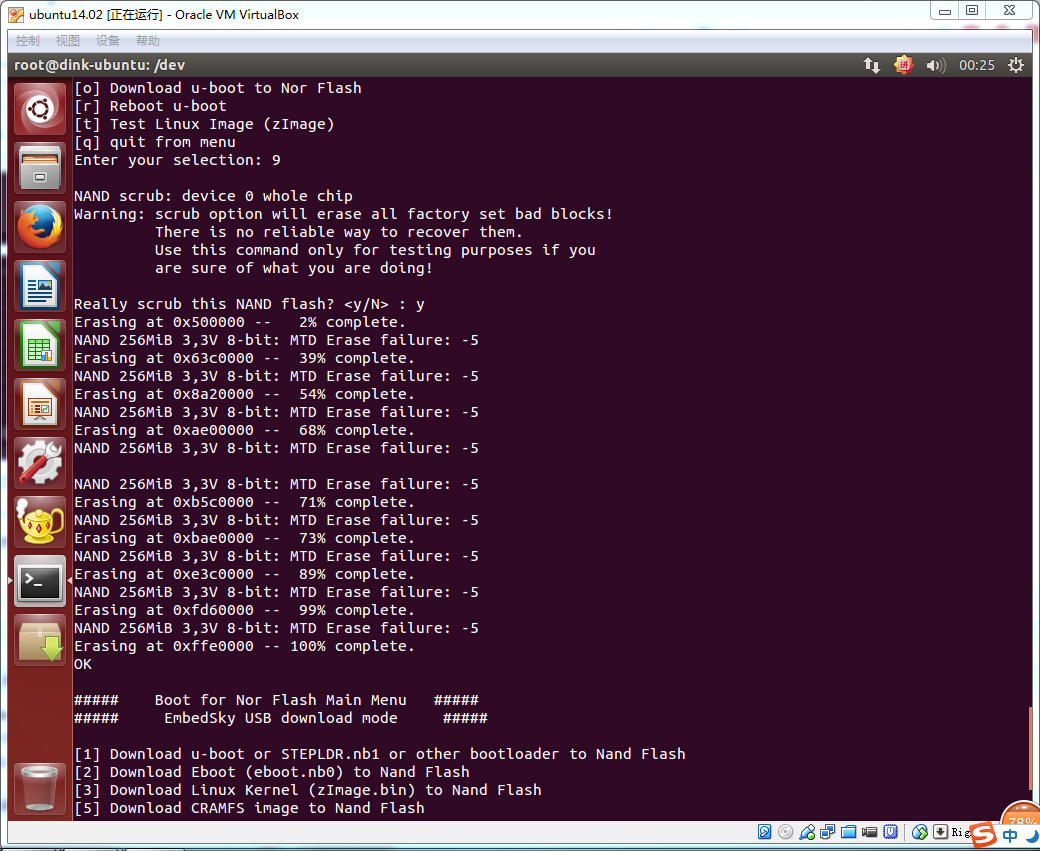


图4-15 linux下终端程序移植

如图4-15，在Qt creator编写好界面程序，需要将开发好的软件应用在linux系统下交叉编译，然后复制到硬件平台（Cortex-A8）上，在此之前先要移植BootLoader、kernel image和rootf。本设计本分功能需要依赖触摸屏，还需要移植tslib，它的作用是校正屏幕。另外为了满足要用到的数据库，所以还需要加入sqlite3，最后添加qt-embedded库。这样就完成了基本的移植，只需要交叉编译好Qt应用程序，放到板子上即可运行了

### 3.2.4 数据库模块

用户可以在PC用户端中点击“查看数据库”按钮，登陆成功后在弹出的窗口中查看数据库中的内容，并可以通过图中所示按键对数据库中的内容进行对应的操作。

(1)首先添加名为“Connection.h”的头文件，在头文件中实现创建数据库并建表的功能。

//创建数据库

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

db.setDatabaseName("ss.db");

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);

return false;

}

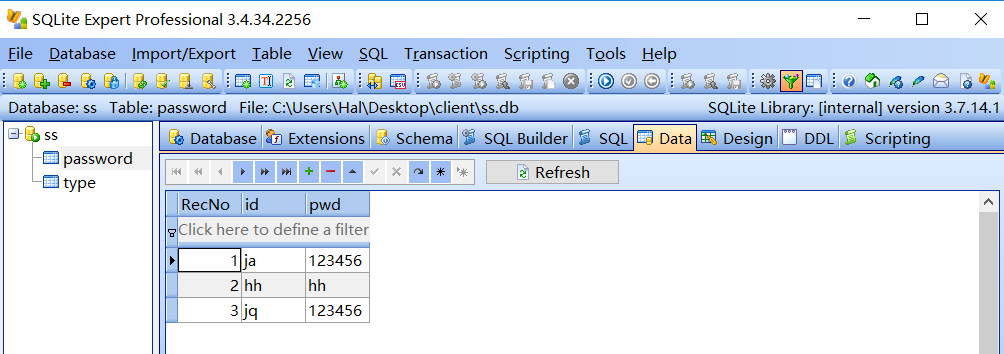
// 创建分类表

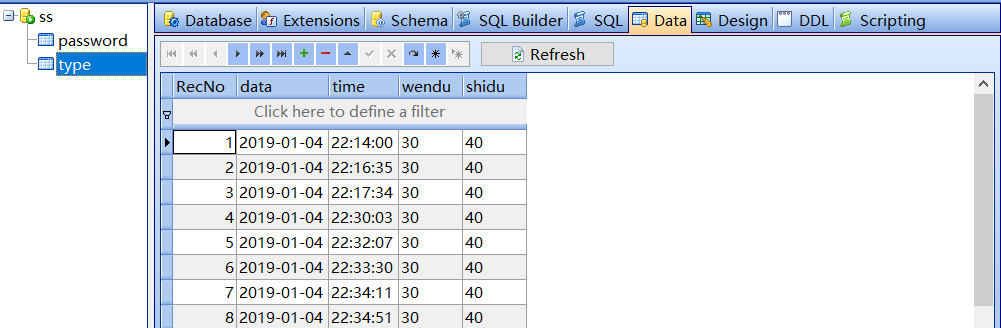
QSqlQuery query;

query.exec("create table type(data varchar(500),time varchar(500),wendu varchar(500), shidu varchar(500))");//执行Sqlite语句

query.exec(QString("insert into type values('%1','%2','30','40')").arg(aaaa).arg(bbbb));

或者使用Sqlite expert professional创建数据库和图表，其中dbo.password保存用户账户，而dbo.type实时储存采集模块传输过来的数据。如下图





(2)将从服务器端接收到的数据保存到数据库表dbo.type中

在客户端中使用占位符位置绑定的方法，实现实时将数据存储入数据库中，代码如下：

void clientWidget::onread()

{

QSqlQuery query;

char bufclient[100]; //缓冲区

char w[5];

char s[5];

…

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

… //显示信息

w[0]=bufclient[6];

w[1]=bufclient[7]; //取得温度数据

s[0]=bufclient[13];

s[1]=bufclient[14]; //取得湿度数据

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

}

}

(3)用户登陆并查看环境因子数据

使用while(query.next()){}遍历dbo.password表，匹配用户成功则跳转页面查询及删除等操作。

if(query.exec("select \* from password"))

{

/\*用户名和密码不能为空\*/

if(ui->lineEdit->text()==NULL&&ui->lineEdit\_2->text()==NULL)

{QMessageBox::warning(this,QString(tr("Waring")),

QString(tr("输入不能为空！请重新输入！")));}

else

{

//输入不为空

QString pwd=" ";

while(query.next())//循环进行查找表中有无对应的用户名

{

if(uname==query.value(0).toString())

{pwd=query.value(1).toString();}

else

{qDebug()<<"none";}}

if (pwd == ui->lineEdit\_2->text()) { //查询用户名与密码是否对应

dia.*exec*(); //跳转页面

}

else

{…//用户名和密码不对应 }}}

/\*在鑫窗口中将数据库数据简单的用QTableView控件表格显示 \*/

…

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

while(query.next())

{

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

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

…

r++;

}

tableView->resize(400,300);

tableView->*setModel*(model);

tableView->show(); //设置QTableVew属性

sum=0;

# **4 系统测试**

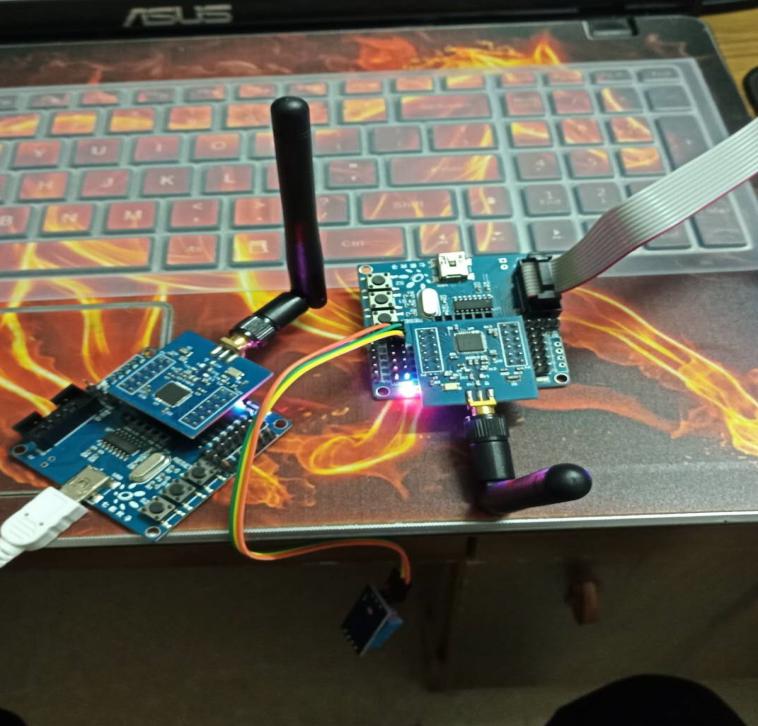


图4-1 传感器模块及节点



图4-2 服务器端串口接收数据显示

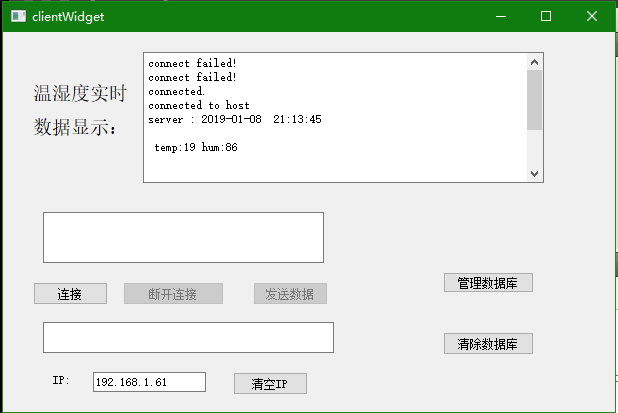


图4-3 用户端数据显示

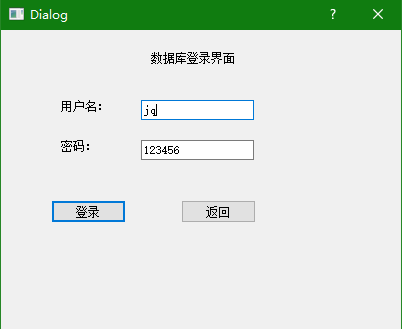


图4-4 数据库登录

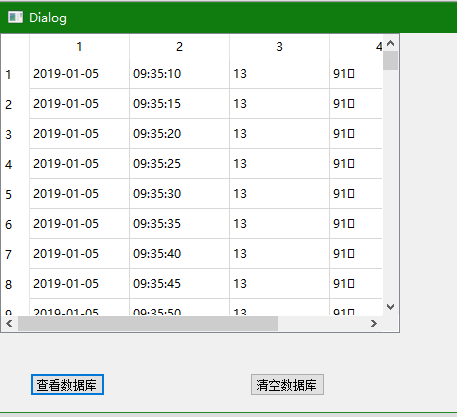


图4-5 数据库数据显示

如图4-1，是本设计使用的传感器模块及节点，收集到的数据统一发到如图4-2所示的监控终端上，其中的通信方式是串口通信，接着就是如图4-3所示，在相同网段的PC端输入监控端的IP，并且在监控终端上，如图4-2所示，点击开始监听，用户端才可以接入网络，实现联网，否则将无法连接。联网之后，监控端的数据就可以在用户端显示，但只要监控端停止监听，用户端将不可以再接收任何数据。点击管理数据库，界面将会出现如图4-4，用户输入用户名、密码将可以登录进Sqlite数据库查看用户端接受的实时数据，如图4-5所示。根据上图所展示的结果可以发现，系统测试成功，串口通信成功、TCP通信成功、数据库连接成功，系统可用。

# **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，四个主要方面分析了温湿度监控的设计和实现，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. 系统中只用到了温湿度传感器模块，至于更多的数据监测模块还需在不断研发跟进中完善；
4. 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，但仍需要增加系统的实用性研究。

## 5.2 展望

由于设备条件的限制，在测试过程中，实验数据跟真实的温湿度存在些许差距，而且系统的算法监测要更加完善，系统可能在精度、准确度上与真实结果有一定的误差，所以在日后的系统完善过程中要对系统进行更加可靠的实用性研究，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. 使用更加精密的数据监测模块；
2. 更加完善数据库的管理、增加手机APP端和微信公众平台；
3. 增强系统功能；
4. 增强系统的实用性等。