

**课 程 设 计**

**课 程：物联网工程开发与应用实例**

**题 目：基于Zigbee的智能温湿度与开关灯检测系统的设计与实现**

**学生姓名：康楷文**

**学 号：15115061007**

**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）

摘 要

**摘要：**随着移动设备的流行和发展，嵌入式系统已经成为一个热点。它并不是最近出现的新技术，只是随着微电子技术和计算机技术的发展，微控制芯片功能越来越大，而嵌入微控制芯片的设备和系统越来越多，从而使得这种技术越来越引人注目。它对软硬件的体积大小、成本、功耗和可靠性都提出了严格的要求。嵌入式系统的功能越来越强大，实现也越来越复杂，随之出现的就是可靠性大大降低。最近的一种趋势是一个功能强大的嵌入式系统通常需要一种操作系统来给予支持，这种操作系统是已经成熟并且稳定的，可以是嵌入式的Linux，WINCE等等。本文所要研究的就是基于**Cortex-A8**和zigbee嵌入式系统的温湿度监控系统的设计与实现。本设计采用了**Cortex-A8**为微处理器作为核心处理器。

其中传感单元负责温湿度数据采集；数据传输模块包含ZigBee传输，串口传输，socket网络传输。用户可以通过基于Cortex-A8的网关查看温湿度，并且通过网关实现对Zigbee节点的控制，除此之外，传输数据存储于PC端服务器数据库上，PC端实时监控温湿度，并且对数据库的数据进行增删查改，PC端还可以对Zigbee终端节点进行控制。

关键词：**嵌入式系统；Cortex-A8；zigbee；温湿度**

**Design and Implementation of Intelligent Temperature and Humidity and Switching Lamp Detection System Based on Zigbee**

**Abstract:** With the popularity and development of mobile devices, embedded systems have become a hot spot. It is not a new technology that has emerged recently. However, with the development of microelectronics and computer technology, the functions of micro-control chips are getting larger and larger, and more and more devices and systems are embedded in micro-control chips. It is more and more eye-catching. It imposes strict requirements on the size, cost, power consumption and reliability of hardware and software. Embedded systems are becoming more powerful and more complex to implement, and the reliability is greatly reduced. A recent trend is that a powerful embedded system usually requires an operating system to support it. This operating system is mature and stable, and can be embedded Linux, WINCE, and so on. The research and implementation of the temperature and humidity monitoring system based on Cortex-A8 and zigbee embedded system is studied in this paper. This design uses the Cortex-A8 as the core processor.

The sensing unit is responsible for temperature and humidity data collection; the data transmission module includes ZigBee transmission, serial transmission, and socket network transmission. Users can view the temperature and humidity through the Cortex-A8-based gateway, and control the Zigbee node through the gateway. In addition, the transmission data is stored in the PC server database, and the PC monitors the temperature and humidity in real time and the data to the database. The addition, deletion, and modification can also be performed on the PC side to control the Zigbee terminal node.

**Key words:** ZigBee；The embedded；；Qt；gec210;

目录

[1设计方案 1](#_Toc8951_WPSOffice_Level1)

[1.1模块划分 1](#_Toc12136_WPSOffice_Level1)

[1.2可行性分析 1](#_Toc27531_WPSOffice_Level1)

[1.3 ZigBee技术 1](#_Toc18407_WPSOffice_Level1)

[1.3.1 协议层 2](#_Toc12136_WPSOffice_Level2)

[1.3.2 网络拓扑结构 2](#_Toc27531_WPSOffice_Level2)

[1.4 sqlite数据库技术 3](#_Toc18293_WPSOffice_Level1)

[1.5终端及QT图形应用技术 4](#_Toc30232_WPSOffice_Level1)

[2系统的硬件设计 5](#_Toc24760_WPSOffice_Level1)

[2.1 电源电路 5](#_Toc493_WPSOffice_Level1)

[2.2 Cortex-A8处理器介绍 6](#_Toc27905_WPSOffice_Level1)

[2.3 GEC-210开发板的介绍 6](#_Toc10712_WPSOffice_Level1)

[2.4 主控电路 7](#_Toc31376_WPSOffice_Level1)

[2.5传感器电路 8](#_Toc11878_WPSOffice_Level1)

[3 网关软件设计 9](#_Toc18407_WPSOffice_Level2)

[3.1 QT程序设计 9](#_Toc12136_WPSOffice_Level3)

[3.2 下位机程序设计 14](#_Toc27531_WPSOffice_Level3)

[4 运行结果及分析 20](#_Toc18293_WPSOffice_Level2)

[4.1 测试环境 20](#_Toc18407_WPSOffice_Level3)

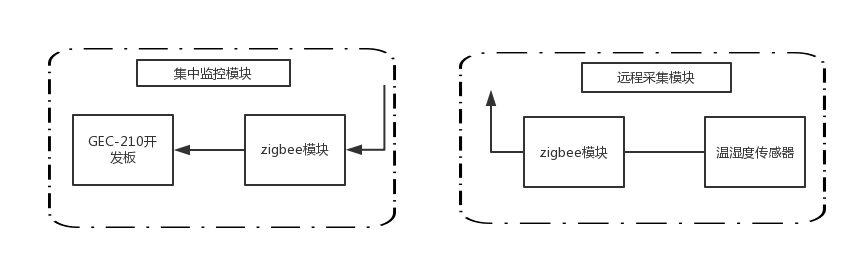
4.2 运行结果 23

[5总结与展望 25](#_Toc11736_WPSOffice_Level1)

**1设计方案**

1.1模块划分

针对智能检测系统的温湿度监控，系统采用集中监控的方式，系统结构框图如图１所示。系统由集中监控模块和远程终端采集模块两部分组成。以单片机为核心构成远程终端控制模块，数字温湿度传感器DHT11采集温湿度，经单片机处理后，实现对坏境温湿度自动监控。



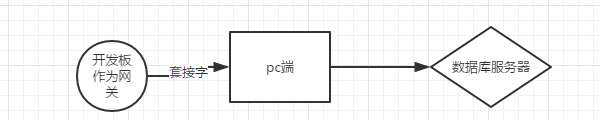


图1 系统模块框图

1.2可行性分析

基于ZigBee技术的远程温湿度监测系统设计主要是针对于我国目前大棚产业或者智能家居的现状而设计，设计的特点是利用ZigBee技术，利用其独特的组建局域网的模式，实现将各个终端数据汇总到一起，实时动态的监测每个终端状态，可以减少生产人员的工作量，在没有异常的情况下，人员可以节省时间，提高工作效率。从而达到以最小的付出收获最大的利益，这也是科技是第一生产力的最终目的。

1.3 ZigBee技术

ZigBee是近几年新兴一种的短距离、低速率、稳定可靠的无线网络技术，是无线传感网络(WSN, wireless sensor network)的核心技术之一。是基于IEEE.802.15.4无线标准研发的，关于组网、应用软件和安全的技术标准。ZigBee的特点就是可靠性高、无人工自组网和断网自恢复能力强、多工作频段工作和开发难度相对简单。

ZigBee的起源却很有意思，它的命名主要是人们对蜜蜂采蜜的观察，蜜蜂在采蜜的过程中，会跳着优美的舞蹈并发出“嗡嗡”的声音，其舞蹈的轨迹很像“Z”字形状；蜜蜂个体比较小，但是群体多，单个个体能量小，能携带花粉。因此，我们用ZigBee技术来表示低成本，低功耗，能量小，传输速率低的无线通信技术。在中文翻译中，通常用字面翻译来解释“紫蜂”。

在ZigBee技术中，我们学习和开发时，通常都要从层的角度去理解。当然它也是和我们熟知的TCP/IP协议的层结构类似，但由于其本身是简单的，低功耗的，低速率的无线传输协议，所以层的结构相对TCP/IP来说要简单的多。

1.3.1 协议层

蓝牙、WIFI等其他网络协议一般有7个层，分别为物理层、数据链路层、网络层、传输层、会话层、表示层、应用层。而ZigBee协议层相对简单，仅为3层，分别为PHY层、MAC层以及应用层。PHY提供两种两种类型的服务：即通过实体接口(PLMN)对PHY层数据和管理提供服务，PHY层数据服务可通过无线物理信道接受和发送物理层协议数据单(PPDU)来实现。

PHY层主要作用是负责无线数据的收发、数据能量的检测、信道的选择、清除信道选择(CCA)、及通过物理媒体对数据包进行收发。

MAC主要的作用就是信道的管理、信道接入、时隙管理、发送确认帧、发送状态连接和断开状态的请求。还提供一些合适的安全机制。

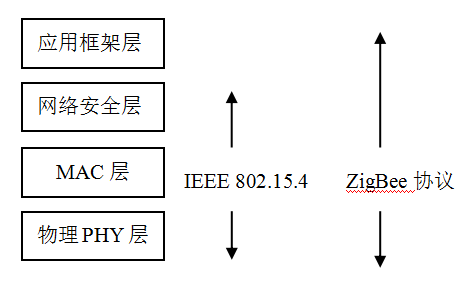


图2 ZigBee协议层示意图

### 1.3.2 网络拓扑结构

ZigBee技术根据相关的应用有三种拓扑结构分别为：星形拓扑结构、树形和网状拓扑结构。如图3所示。

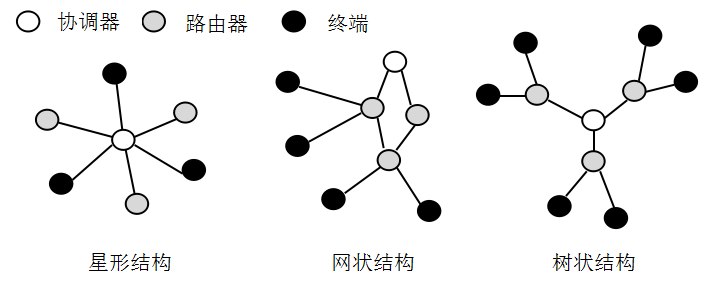


图3 zigbee拓扑结构

星形结构是由一个协调器、多个路由器和终端组成，协调器能同时和路由器和终端组成网络通信。其中路由器和终端不能直接通信，必须通过协调器进行转换操作，才能实现数据的交换。

网状结构与星形结构不同的是，它各路由器之间也能相互通信，各路由器又能和协调器进行数据的交换，路由器分别把终端采集的数据通过路由器的转换、通信反馈给协调器。树状结构特点更明确，类似于等级似的数据交换，一个路由器只负责自己旗下的几个终端节点，不与其他路由器通信，只与协调器进行数据交换。然后协调器要控制终端也必须经过他们的各自所归属的路由器。

1.4 sqlite数据库技术

SQLite，是一款轻型的数据库，是遵守ACID的关系型数据库管理系统，它包含在一个相对小的C库中。它是D.RichardHipp建立的公有领域项目。它的设计目标是嵌入式的，而且目前已经在很多嵌入式产品中使用了它，它占用资源非常的低，在嵌入式设备中，可能只需要几百K的内存就够了。它能够支持Windows/Linux/Unix等等主流的操作系统，同时能够跟很多程序语言相结合，比如 Tcl、C#、PHP、Java等，还有ODBC接口，同样比起Mysql、PostgreSQL这两款开源的世界著名数据库管理系统来讲，它的处理速度比他们都快。SQLite第一个Alpha版本诞生于2000年5月。 至2015年已经有15个年头，SQLite也迎来了一个版本 SQLite 3已经发布。

不像常见的客户-服务器范例，SQLite引擎不是个程序与之通信的独立进程，而是连接到程序中成为它的一个主要部分。所以主要的通信协议是在编程语言内的直接API调用。这在消耗总量、延迟时间和整体简单性上有积极的作用。整个数据库(定义、表、索引和数据本身)都在宿主主机上存储在一个单一的文件中。它的简单的设计是通过在开始一个事务的时候锁定整个数据文件而完成的。

## 1.5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；因此本设计采用cortex-A8为核心的嵌入式板作为网关，配置7寸的电容屏，提供用户交互的功能，考虑到电脑已经普及，因此将PC端作为服务器和控制端。

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。Qt 是一个1991年由Qt Company开发的跨平台C++图形用户界面应用程序开发框架。它既可以开发GUI程序，也可用于开发非GUI程序，比如控制台工具和服务器。Qt是面向对象的框架，使用特殊的代码生成扩展（称为元对象编译器(Meta Object Compiler, moc)）以及一些宏，Qt很容易扩展，并且允许真正地组件编程。2008年，Qt Company科技被诺基亚公司收购，Qt也因此成为诺基亚旗下的编程语言工具。2012年，Qt被Digia收购。2014年4月，跨平台集成开发环境Qt Creator 3.1.0正式发布，实现了对于iOS的完全支持，新增WinRT、Beautifier等插件，废弃了无Python接口的GDB调试支持，集成了基于Clang的C/C++代码模块，并对Android支持做出了调整，至此实现了全面支持iOS、Android、WP,它提供给应用程序开发者建立艺术级的图形用户界面所需的所有功能。基本上，Qt 同 X Window 上的 Motif，Openwin，GTK 等图形界 面库和 Windows 平台上的 MFC，OWL，VCL，ATL 是同类型的东西。

**2系统的硬件设计**

2.1 电源电路

CC2530及外围传感器工作电压为3.3V的工作电压，所以在电源电路设计的时候将直流输入的5V电压转换成3.3V，如图所示。

图4中，电压变换采用了LM1117电压转换芯片，该芯片的目的就是将输入的5V电压转换成微处理器需要的3.3V。其中DCIN为直流5V输入接口，经过F1的保险丝和开关进入LM1117的3号管脚，经芯片转换过后，2号引脚输出3.3V电压。其中在2号管脚和3号引脚都会有电解电容、普通瓷片电容接地；瓷片电容的目的是滤除高频杂波。在转换正常的情况下，二极管LED指示灯会接通，并发光。

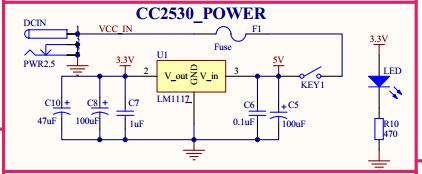


图 4 cc2530电源电路

## 2.2 Cortex-A8处理器介绍

## ARM Cortex-A8处理器是第一款基于下一代ARMv7架构的应用处理器，是有史以来ARM开发的性能最高、最具功率效率的处理器，使用了能够带来更高性能、功耗效率和代码密度的Thumb-2技术。

## Cortex-A8处理器配置了先进的超标量体系结构管线，能够同时执行多条指令，并且提供超过2.0 DMIPS/MHz。处理器集成了一个可调尺寸的二级高速缓冲存储器，能够同高速的16K或者32K一级高速缓冲存储器一起工作，从而达到最快的读取速度和最大的吞吐量。Cortex-A8处理器使用了先进的分支预测技术，并且具有专用的NEON整型和浮点型管线进行媒体和信号处理。Cortex-A8处理器运行速度最高可达到1GHz，从而满足高性能消费产品设计的需要。

## 2.3 GEC-210开发板的介绍



图5 gec210开发板

GEC210 是一款高性能的 Cortext-A8 核心板，它采用三星S5PV210作为主处理器，运行主频可高达1GHz。GEC210开发板可流畅运行Android、Linux和WinCE6等高级操作系统。

GEC-210具有两种 USB 接口，一种是 USB Host(2.0)接口，共 2 个，它和普通 PC 的 USB 接口是一样的，可以接 USB 摄像头、 USB 键盘、 USB 鼠标、优盘等常见的 USB 外设；另外一种是 miniUSB(2.0)，主要用于 Android 系统下的 ADB 功能，用于软件安装和程序调试。本文的智能农场系统中ZigBee协调器模块通过USB Host(2.0)接口于控制终端相连。原理图如图所示 。

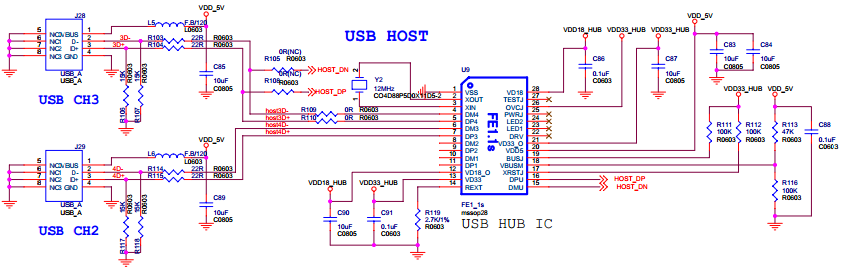


图6 USB Hos 电路图

2.4 主控电路

CC2530的工作正常工作电压3.3V，因为该芯片内部自带AD转换芯片，所以这里会有模拟电压和数字电压的区别，如图10所示。芯片引脚中标有DVDD的均为数字电源正，标有AVDD的均为模拟电源正。32、33、22、23引脚都接上了晶振，但不同的是：22、23接的是32M的晶振，为系统提供正常工作提供基准时序。32、33引脚接上的晶振为32.768K，其作用就是为系统时间提供基准时序，开发者可以采用这个时序基准开发RTC实时时钟；另外，当单片机处于休眠状态时，该晶振还可以继续提供实时时钟的时序。

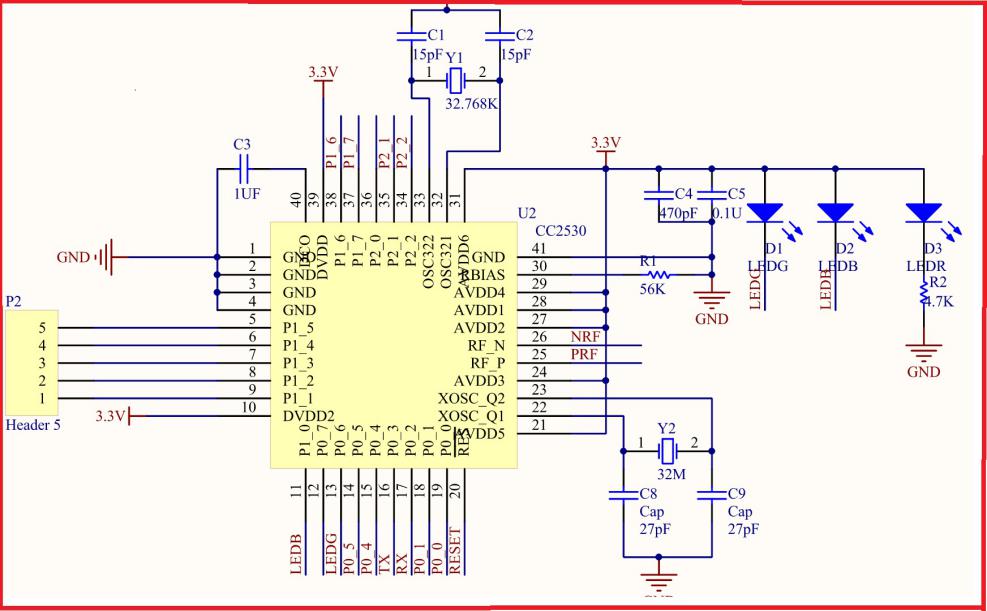


图7 CC2530电路



图8 CC2530实物图

2.5传感器电路

DHT11[数字温湿度传感器](https://baike.baidu.com/item/%E6%95%B0%E5%AD%97%E6%B8%A9%E6%B9%BF%E5%BA%A6%E4%BC%A0%E6%84%9F%E5%99%A8)是一款含有已校准数字信号输出的温湿度复合传感器，它应用专用的数字模块采集技术和温湿度传感技术，确保产品具有极高的可靠性和卓越的长期稳定性。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湿度校验室中进行校准。校准系数以程序的形式存在OTP内存中，传感器内部在检测信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，使其成为该类应用中，在苛刻应用场合的最佳选择。产品为4针单排引脚封装，连接方便。

·湿度测量范围：20%∼90%

·湿度精度：±5%RH

·温度精度：±2℃

·分辨率：1

·封装：4针单列直插

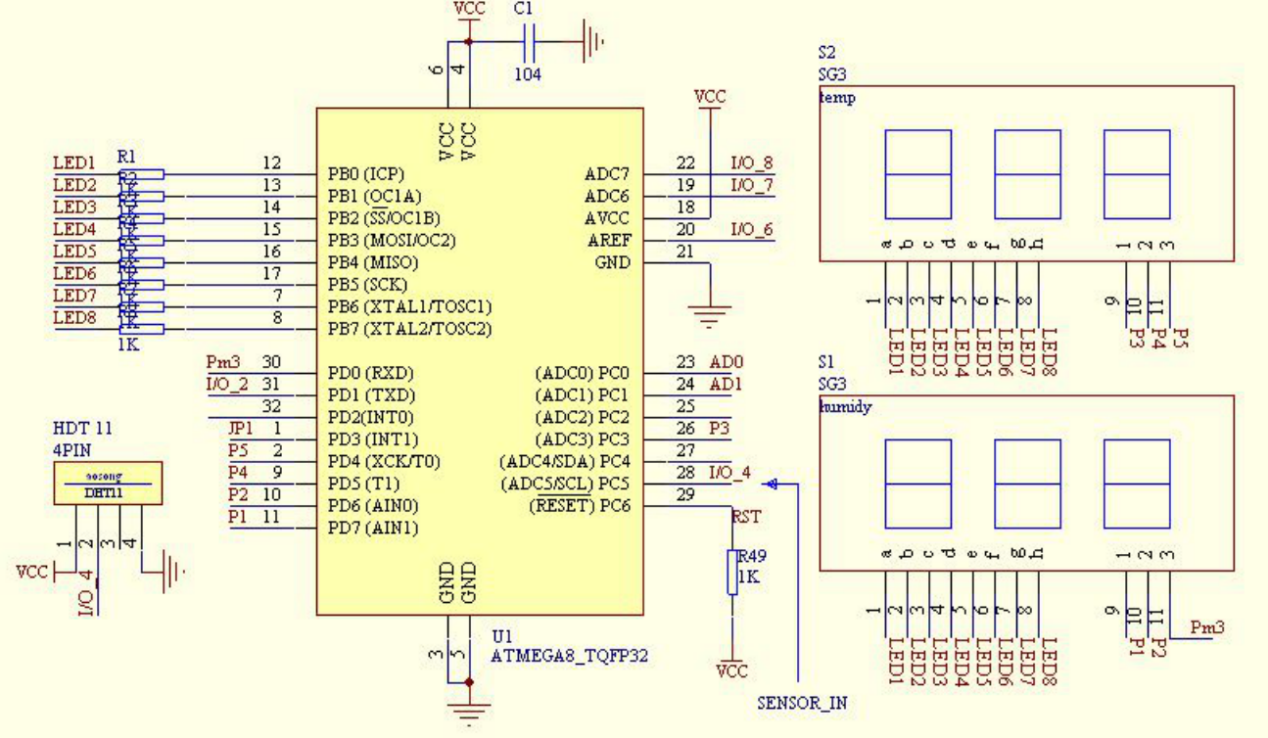


图9 DHT11原理图

**3 网关软件设计**

系统选用Linux嵌入式操作系统作为GEC-210监控中心的操作系统，在建立交叉编译环境后，采用韩国 MIZI公司开发的VIVI软件，完成ARM9嵌入式系统的BootLoador移植；采用挪威Trolltech公司开发的ＱＴ图形软件完成嵌入式系统的用户图形界面的开发。在实时监控界面中，需要通过套接字完成数据的传输，下面重点介绍系统的PC端程序设计。

3.1 QT程序设计

系统总体结构如图10.所示，整个智能温湿度与开关灯监控系统由一块coter-a8核心的S3C2440开发板作为网关、温湿度采集终端、zigbee无线网络,pc端，数据库组成。温湿度采集终端将采集到的温湿度数据按照一定格式封装后，通过zigbee无线路由传送到gec210开发板；开发板接收到温湿度数据后，通过套接字实时显示监控与控制快闪和慢闪灯的交互。

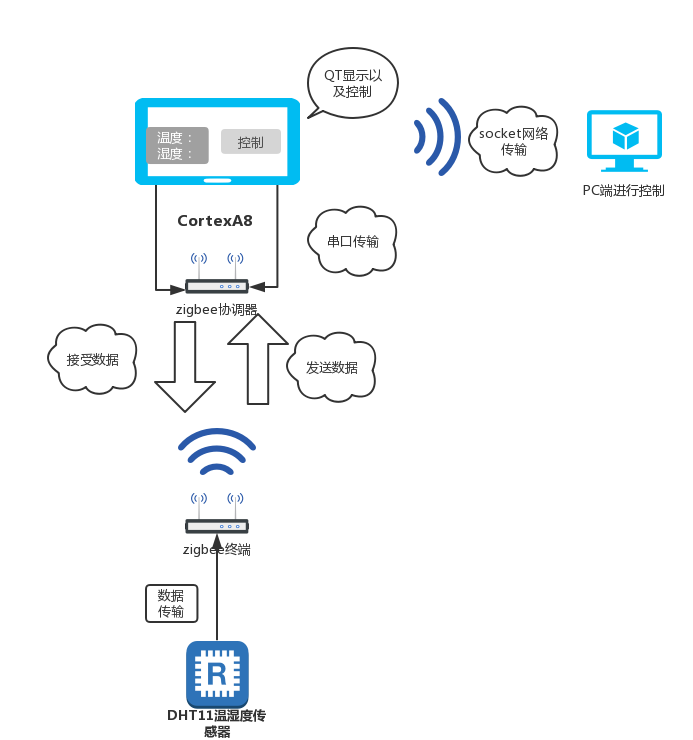


图10 系统结构框图

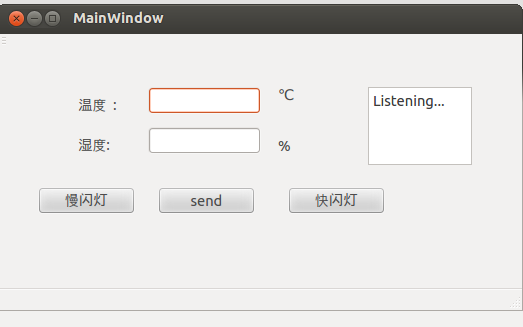


图11 网关界面图

网关界面简洁美观，当PC与网关用TCP Socket连接成功时，文本框会显示“connect”表示连接成功，并且可以实时显示温湿度，通过“send”将温度湿度的数据发送给PC端，除此之外还可以可以通过网关对节点的开关灯控制。

关键代码：

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QtGui>

#include <QAction>

#include <QSharedMemory>

#include "stdio.h"

#include<QTimer>

#include"qextserialport.h"

MainWindow::MainWindow(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::MainWindow)

{

ui->setupUi(this);

warnboxflag=0;

mycom=new Posix\_QextSerialPort( "/dev/ttyUSB0" ,QextSerialBase::Polling);//定义串口，传递参数，初始化串口，这句在linux下定义的"/dev/ttyS0"

mycom->open(QIODevice::ReadWrite);//以读写方式打开串口

mycom->setBaudRate(BAUD115200);

mycom->setDataBits(DATA\_8);

mycom->setParity(PAR\_NONE);

mycom->setStopBits(STOP\_1);

mycom->setFlowControl(FLOW\_OFF);//设置数据流控制，我们使用无数据流控制的默认设置

QTimer \*readTimer=new QTimer(this);//定时器关联

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

readTimer->start(3000);//延时100ms

ui->centralWidget->setStyleSheet("transparent");

server = new QTcpServer(this);

if(!server->listen(QHostAddress::Any,9090)) //如果服务没监听到9090端口执行{}里信息

{

ui->textEdit->append("Listening filed!");

return;

}

ui->textEdit->append("Listening...");

connect(server,SIGNAL(newConnection()),this,SLOT(imcoming())); //如果有新的连接信息newConnection(),触发imcoming()

connect(ui->pushButton,SIGNAL(clicked()),this,SLOT(onsend())); //点击发送，触发相应槽

connect(ui->pushButton\_2,SIGNAL(clicked()),this,SLOT(onlightfast()));

connect(ui->pushButton\_3,SIGNAL(clicked()),this,SLOT(onlightslow()));

}

MainWindow::~MainWindow()

{

delete ui;

}

void MainWindow::readMyCom()//读串口函数

{

QByteArray temp = mycom->readAll();

//读取串口缓冲区的所有数据给临时变量temp

QByteArray a=temp.left(6);

QByteArray b=temp.right(4);

QByteArray aa=a.right(2);

QByteArray bb=b.right(2);//截取串口数据

ui->lineEdit->setText(a.right(2));

ui->lineEdit\_2->setText(b);//将串口的数据显示在窗口的文本浏览器中

aa=a.right(2);

bool ok;

if (aa.toInt(&ok)>32 && (warnboxflag==0) )

{

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

tr("temperature is to higher!"),

QMessageBox::Yes);

warnboxflag=1;

}

}

void MainWindow::delay()//延时函数

{

//QTimer \*timer=new QTimer(this);

//readTimer->stop();

ui->lineEdit->insert("1");

//ui->lineEdit->clear();

}

void MainWindow::imcoming()

{

socket = server->nextPendingConnection();//把客户端的socket端口存入到服务器端的socket端口

ui->textEdit->append("connected.");

connect(socket,SIGNAL(readyRead()),this,SLOT(onread())); //管道准备好读取数据，触发onread()

}

void MainWindow::onread()

{

char buf[100]; //暂存数据中转站

memset(buf,0,sizeof(buf)); //先清0

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

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

mycom->write(buf,4);

}

void MainWindow::onsend()

{

QString data=ui->lineEdit->text();

QString data2=ui->lineEdit\_2->text();

QString data3=" ";

data=data+data3+data2;

char sendbuf[100]={0};

QByteArray ba = data.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

memset(sendbuf,0,sizeof(sendbuf));

}

void MainWindow::onlightfast(){

QByteArray a="tr";

mycom->write(a,4);

}

void MainWindow::onlightslow(){

QByteArray b="fa";

mycom->write(b,4);

}

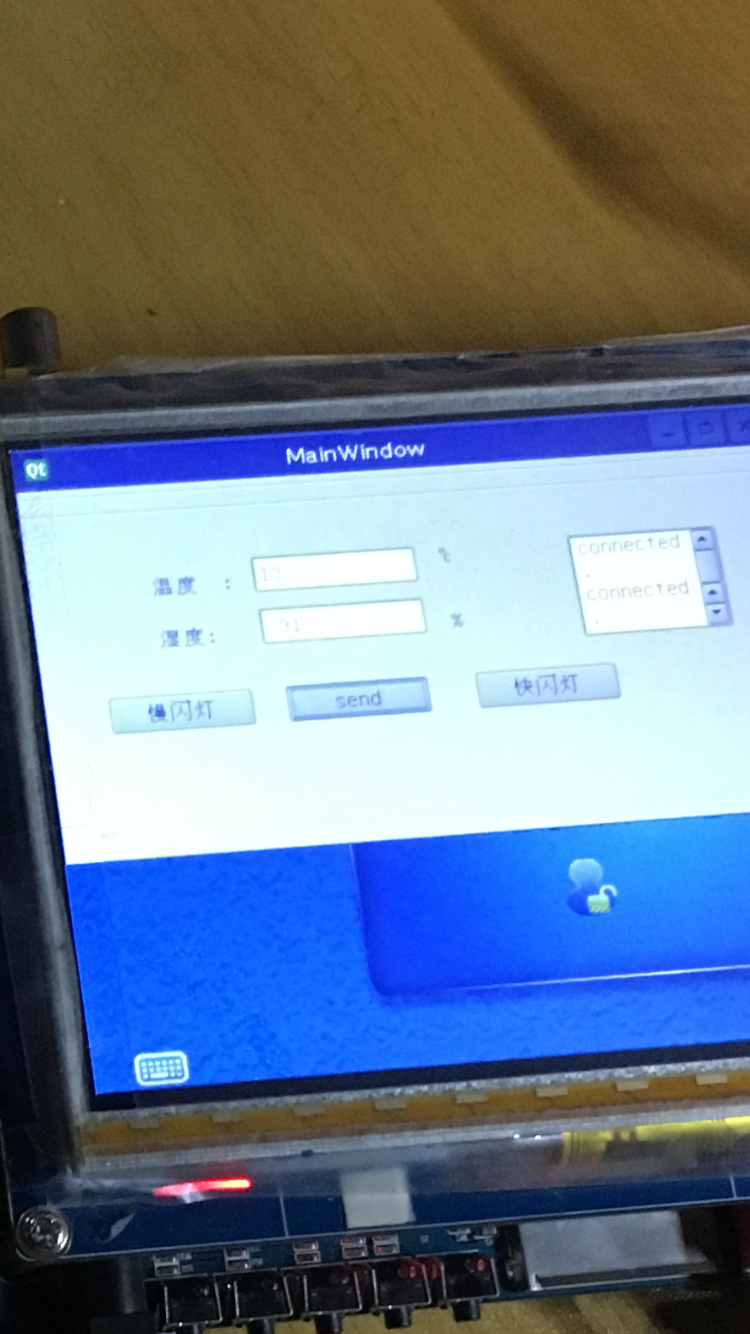


图12 连接成功的界面显示

3.2 下位机程序设计

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。程序流程图如图所示。

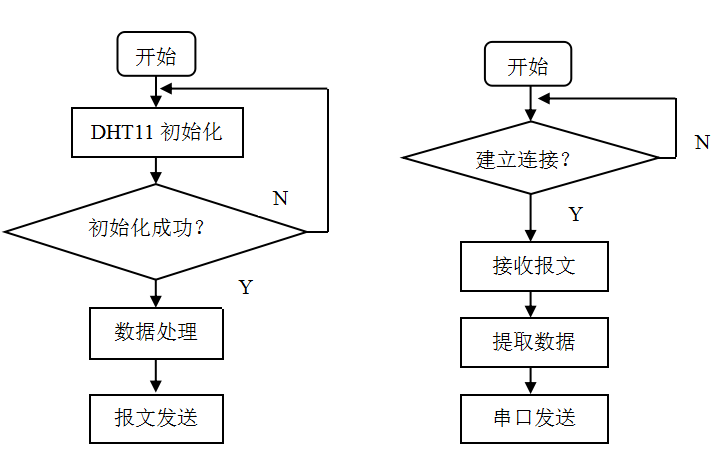


图13 下位机程序流程图

在节点程序设计时，首先要对DHT11驱动的子程序编写，测试并通过；然后移植到协议栈的程序代码中调试。能在协议栈中也能实现温湿度采集，并实现数据的发送程序编写。

在协调器程序设计时，首要问题就是报文的提取。在Z-Stack协议栈中，接收到的报文并不是单独的数据，所以必然要将需要的数据包提取出来，然后解析数据并编写协议栈环境下的串口发送程序。

## 3.2.1 温湿度采集与传输

DHT11传感器在极为严格的温湿度校验室中进行校准，校准系数以程序的形式存储在otp内存中。传感器内部接收到启动信号后，自动完成温湿度的测量，在处理过程中调用这些校准系数进行修正。单片机与DHT11之间的通信和同步，采用单总线数据格式，一次通信时间４ms左右，按照一定的时序进行读写。单片机与 DHT11通信流程如图4.3所示。通信开始后，主机发送启动信号给DHT11，按照时序延时一定时间，等待DHT11响应；当DHT11正确接收启动信号后，向单片机发送响应信号，启动自身的数据采集，完成温湿度信号的采集和转换；单片机在接收到DHT11的响应信号后，延时，保证DHT11信号采集和转化的完成；最后主机读取DHT11的温湿度数据，结束一次数据的读取。

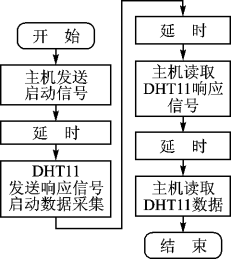


图14 zigbee与ＤＨＴ１１通信流程图

关键代码：

void DHT11(void) //温湿传感启动

{

DATA\_PIN=0;

Delay\_ms(19); //>18MS

DATA\_PIN=1;

P0DIR &= ~0x80; //重新配置IO口方向

Delay\_10us();

Delay\_10us();

Delay\_10us();

Delay\_10us();

if(!DATA\_PIN)

{

ucharFLAG=2;

while((!DATA\_PIN)&&ucharFLAG++);

ucharFLAG=2;

while((DATA\_PIN)&&ucharFLAG++);

COM();

ucharRH\_data\_H\_temp=ucharcomdata;

COM();

ucharRH\_data\_L\_temp=ucharcomdata;

COM();

ucharT\_data\_H\_temp=ucharcomdata;

COM();

ucharT\_data\_L\_temp=ucharcomdata;

COM();

ucharcheckdata\_temp=ucharcomdata;

DATA\_PIN=1;

uchartemp=(ucharT\_data\_H\_temp+ucharT\_data\_L\_temp+ucharRH\_data\_H\_temp+ucharRH\_data\_L\_temp);

if(uchartemp==ucharcheckdata\_temp)

{

ucharRH\_data\_H=ucharRH\_data\_H\_temp;

ucharRH\_data\_L=ucharRH\_data\_L\_temp;

ucharT\_data\_H=ucharT\_data\_H\_temp;

ucharT\_data\_L=ucharT\_data\_L\_temp;

ucharcheckdata=ucharcheckdata\_temp;

}

wendu\_shi=ucharT\_data\_H/10;

wendu\_ge=ucharT\_data\_H%10;

shidu\_shi=ucharRH\_data\_H/10;

shidu\_ge=ucharRH\_data\_H%10;

}

else //没用成功读取，返回0

{

wendu\_shi=0;

wendu\_ge=0;

shidu\_shi=0;

shidu\_ge=0;

}

P0DIR |= 0x80; //IO口需要重新配置

}

### 3.2.2 zigbee无线网络设计

短距离无线通信技术目前已成为无线通信的一个重要分支。现实环境应用中，传输数据通常为少量、突发信号，以数据量小、实时传送为数据特征，技术应运而zigbee技术是一种功耗低、成本低、可靠性高、网络容量大、数据安全和通用性好的短距离无线通信技术。

zigbee技术在IEEE802.15.4标准的推动下，在农业、工业、环境、医疗等领域取得了成功的应用。，RFD为终端设备，主要负责与FFD设备 通 信。FFD设备主要包括网络协调器和网络路由器，其中网络协调器主要负责建立网络，配置无线网络的相关参数；网络路由器主要负责寻找、建立和修复网络报文。

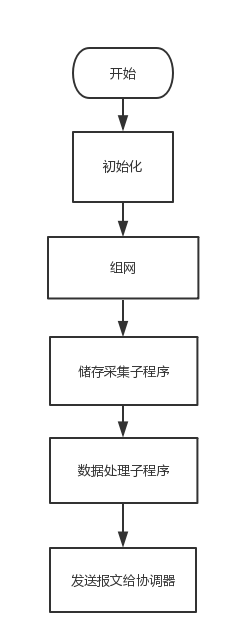


图15 zigbee终端主程序流程图

终端关键代码：

//将温湿度的转换成字符串

temp[0] = wendu\_shi+0x30;

temp[1] = wendu\_ge+0x30;

temp[2] = '\0';

humidity[0] = shidu\_shi+0x30;

humidity[1] = shidu\_ge+0x30;

humidity[2] = '\0';

//将数据整合后方便发给协调器

osal\_memcpy(strTemp, temp, 2);

osal\_memcpy(&strTemp[2], " ", 2);

osal\_memcpy(&strTemp[4], humidity, 3);

协调器关键代码：

//获得的温湿度通过串口输出

HalUARTWrite(0, "T&H:", 4);

HalUARTWrite(0, (uint8 \*)strTemp, 6);

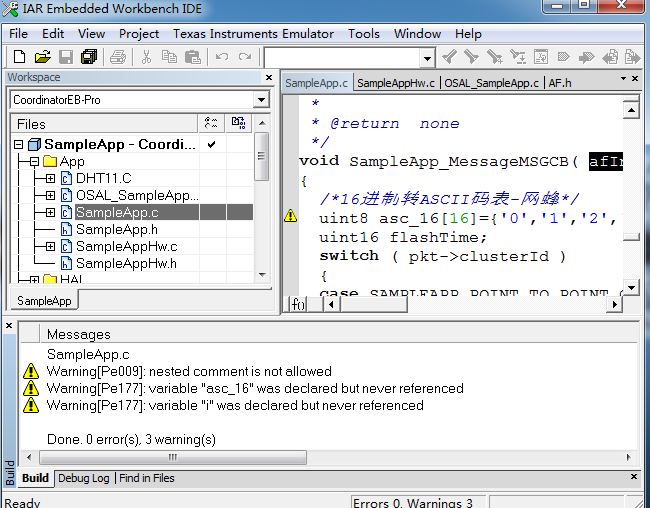
HalUARTWrite(0, "\n",1);

**4 运行结果及分析**

## 4.1 测试环境

经过系统流程的设计、代码的编写、移植、修改、调试，将协调器程序代码编译后通过JTAG下载的方式下载到CC2530芯片的Flash中；接通电源后，此时的协调器具有组建网络的能力。接下来把程序切换到终端模式，对程序进行编译、下载。同上电后，终端节点会自动查找协调器，自行搜索是否有对应的协调器的存在；若搜索到协调器的存在会主动建立网络连接。

程序编写好后，点击编译按钮，如图所示。

 图16 程序编译调试界面

编译通过后，通过JTAG将CC2530的下载口与PC机相连，协调器程序下载需在左上角的工程选项复选框选择CoordinatorEB-Pro方式，点击下载；下载终端节点的程序需要更改左上角工程选项复选框选择EndDeviceEB-Pro方式，点击下载。下载界面结果如图。

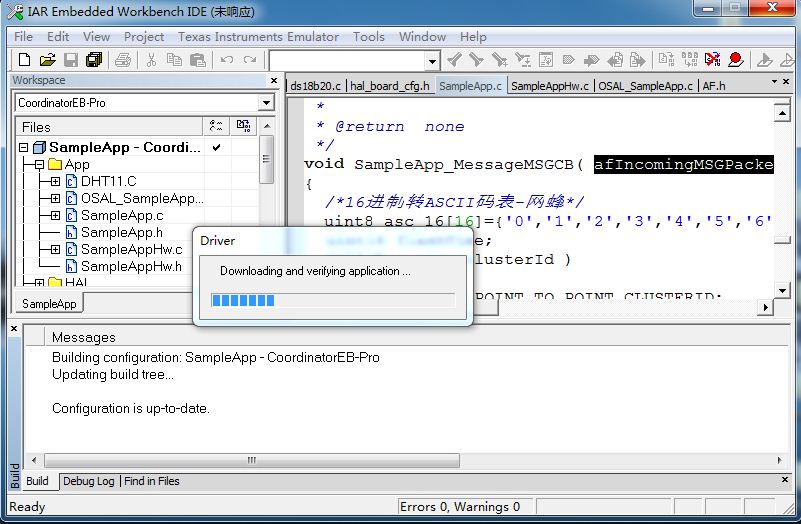
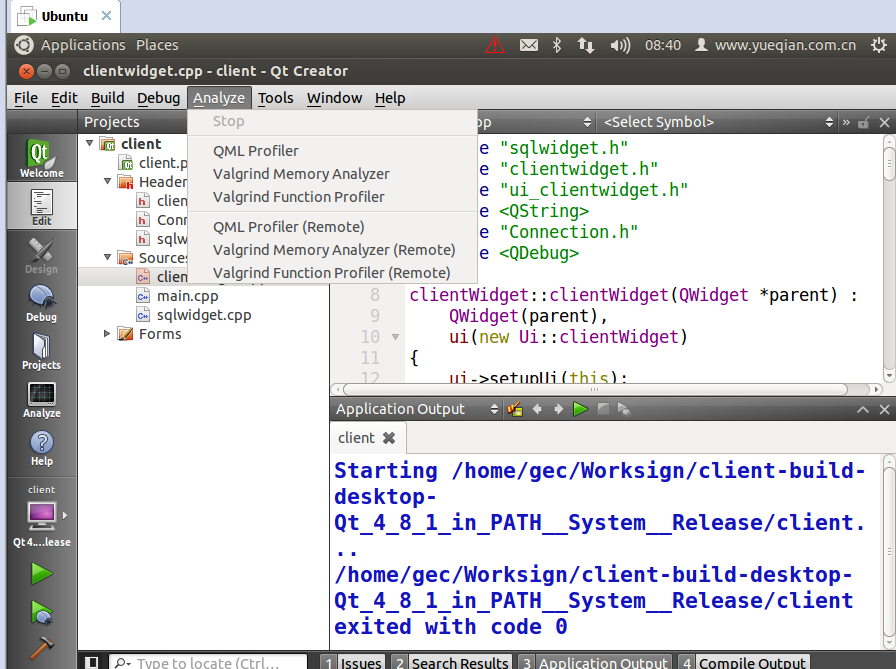


图17 JTAG程序下载界面

嵌入式系统通常是一个资源受限的系统，因此直接在嵌入式系统的硬件平台上编写软件比较困难， 甚至是不可能的。首先在通用计算机上编写程序；然后通过本地编译或者交叉编译生成目标平台上可以运行的二进制代码格式；最后再下载到目标平台上的特定位置上运行。

图18 QT程序的设计与调试

4.2 运行结果

程序代码编译下载到芯片内，分别给协调器和终端供电；协调器的组网指示灯会闪烁，等到有终端节点加入其网络的时候，指示灯就会稳定的指示。终端节点的网络指示灯也会一直指示网络连接状态。在终端节点的设备上，协调器（下）、终端节点（上）组网状态的实物设备如图所示。

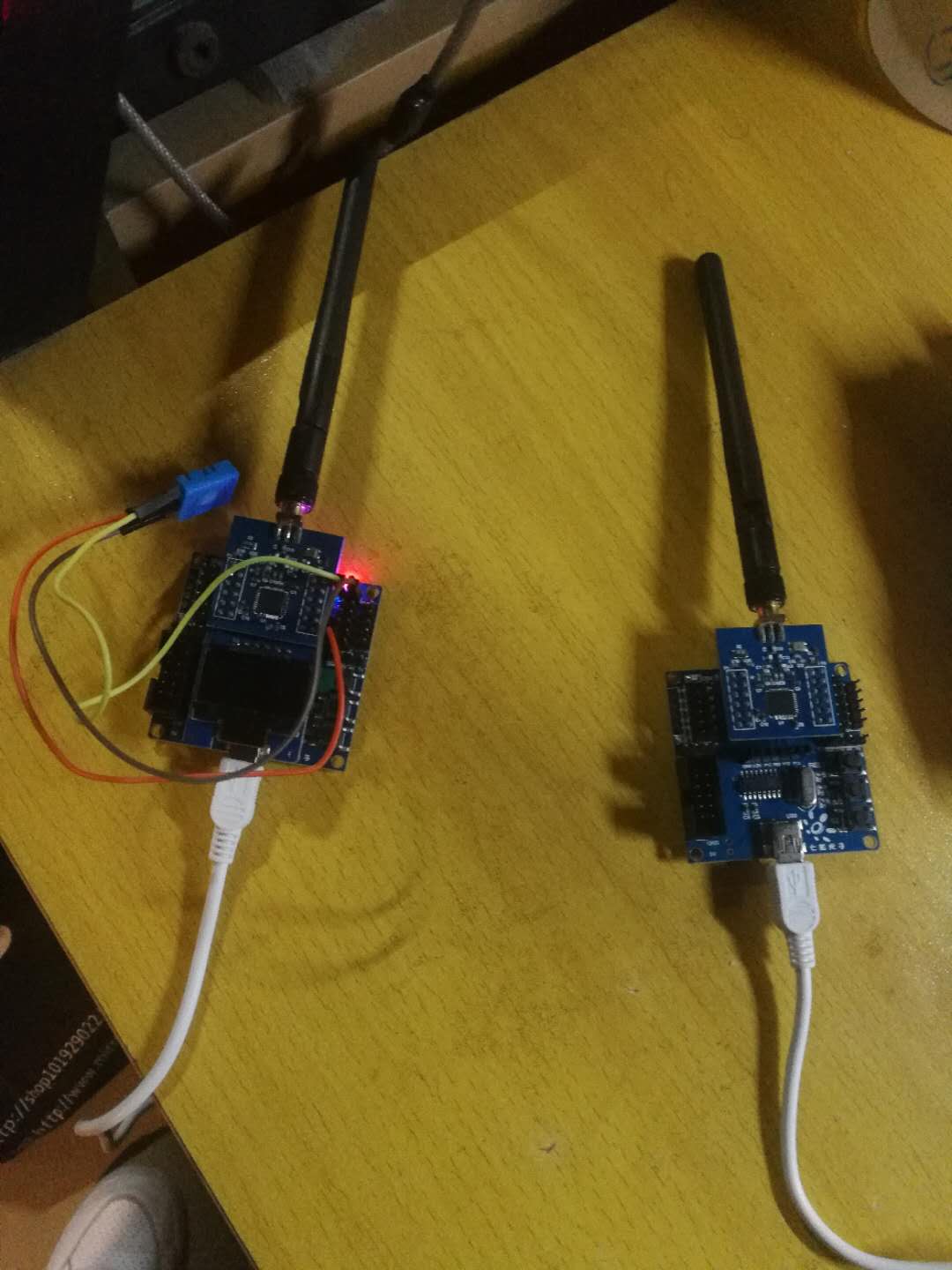


图19 zigbee连接实物图

****

图20 实物连接图

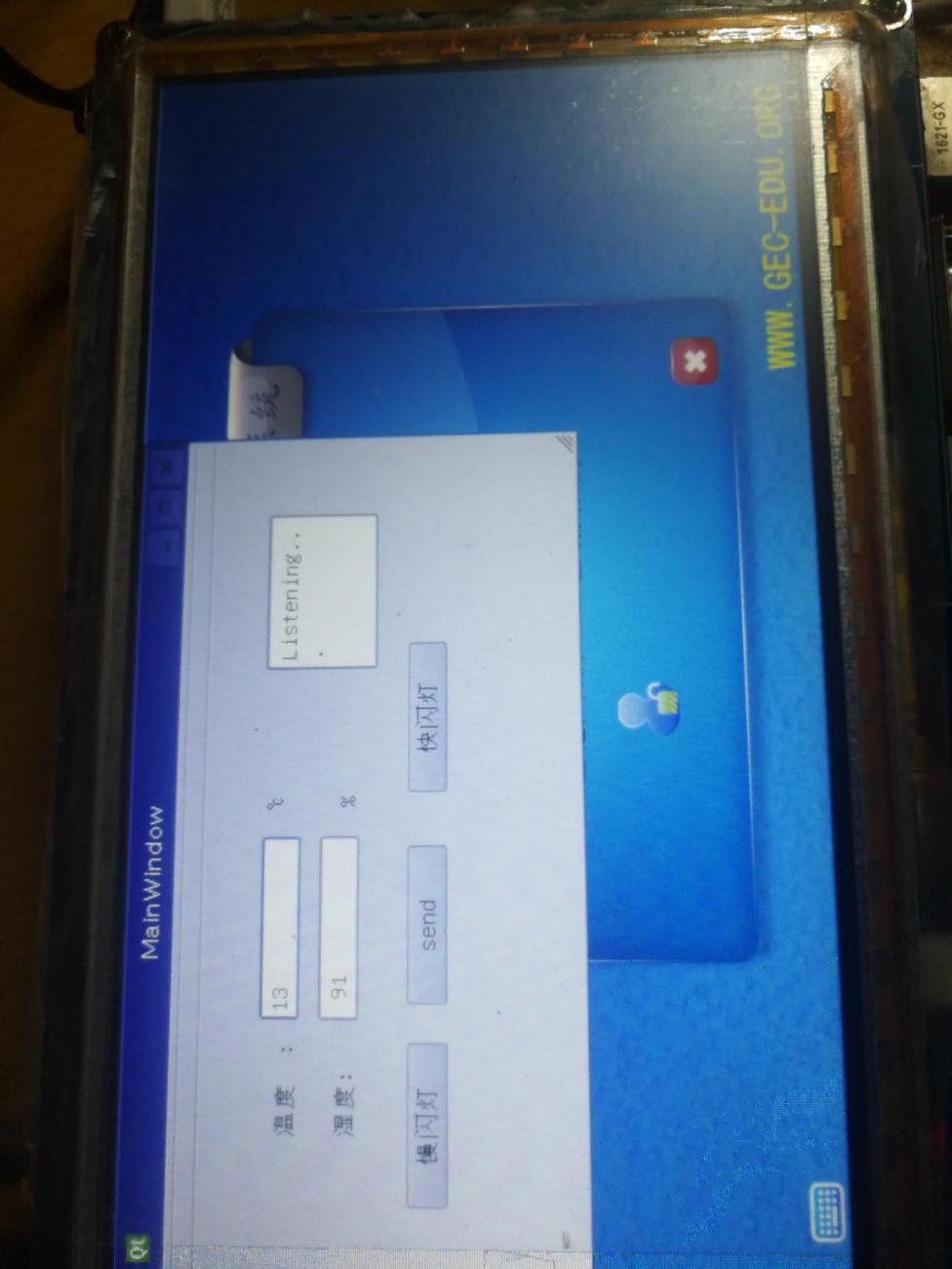


图21 开发板作为网关图

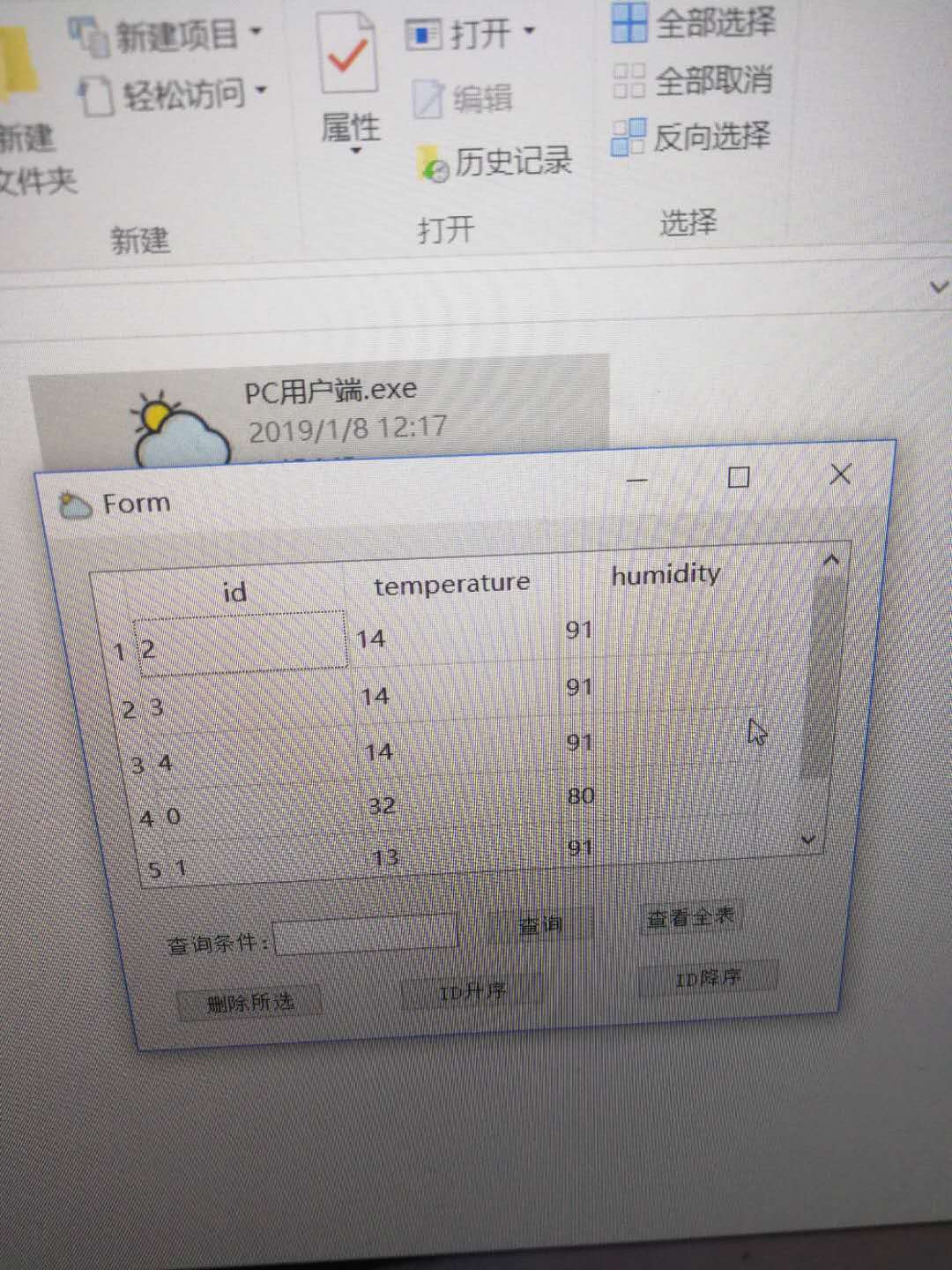


图22 数据库图表如上

5总结与展望

通过多次连接、组网、通信，数据基本都能稳定的上传，上位机也能很快的识别并且稳定的显示其上传结果。当将终端节点设备拿到很远的地方；并且有很多建筑物阻挡的地方时，数据通信会不稳定甚至出现断网的情况。若供电不稳定或者终端节点和协调器上电过于频繁，也会出现数据的突变为0的情况，但持续时间很短，为一个数据更新周期。下一个更新周期的开始就会回复正常显示状态。理论上来说，根据天线的发射功率，能达到2km的通信距离，当然是在无障碍物的前提下，解决这样的问题需要增大天线的发射功率，更换更好的接收天线。但由于经济因素和时间情况不允许，所以这样问题在以后进行改进。另外虽然数据突变情况出现次数很少，但是也反映出一个问题，需要解决这样的问题，需要对数据进行滤波处理，这里的滤波采用数字滤波，可以采用平滑滤波算法，平均值滤波算法等，但由于本人目前知识能力有限，对这些算法知识了解不够。在以后的学习中会花更多的时间解决这样的问题。

数据库的完善也不能说是完美的，只能说趋于一个形态，在这方面我们会更加努力，将功能完善的更好，PC端的界面也可以制作的更加精美，PC端控件也将符合大众审美与操作的方便性。

参考文献：

[1]　 孙天泽,袁文.嵌入式设计 Linux 驱动开发指南:基于 ARM9 处理器 [M]. 北京 : 电子工业出版社 ,2007

[2]　 CHTM-02 温湿度传感器模块产品使用说明书 . 广州西博臣科 技有限公司 ,2005.01 [3]　 楚红雨，李磊民，黄玉清，张静 . 实时操作系统 Linux 在 ARM9 上移植的实现 [J]. 计算机工程 ,2005,31(10)：226-228