

**课 程 设 计**

**课 程：物联网应用系统开发实践**

**题 目：智能农业温湿度采集系统**

**学生姓名：郑志豪**

**学 号：15115061038**

**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）

**智能农业温湿度采集系统的设计与实现**

**摘 要**：一般来说，普通家庭对于温度和湿度的值并不在意。但随着经济不断的发展，人们对于环境的要求越来越来大，特别是近几年环境逐渐恶劣的问题，越来越多的普通老百姓开始关注农业耕作的环境问题。那平时人们如何获知环境中的各项因素值呢？本次课程设计应运而生，针对以往获取温湿度设备的缺点，对传统设备进行创新型改良，使其符合人们对平常农业耕作的要求和改善。

本设计主要针对农业生产的过程，在传统的设备进行创新型的改良。智能农业温湿度采集系统主要分成三部分：传感单元、数据传输单元、终端操控与交互模块。其中传感单元分别有温度传感与湿度传感两部分，采集到的数据通过网络和socket套接字传输到控制系统中进行科学分析，并在页面中显示出来。

用户数据存储于本地和远地服务器端数据库上，用户可以通过控制系统查看实时数据外，还能在交互页面中通过账号密码登录到后台的数据中，查看每个时间段温度与湿度的值以及其数据的走势变化，十分适合现实农产过程中的变化需求。

**关键字：** ZigBee；嵌入式；QT；温湿度

**Design and Implementation of Intelligent Agricultural Temperature and Humidity Acquisition System**

**Abstract:** Generally speaking, the average family does not care about the value of temperature and humidity. However, with the continuous development of the economy, people are increasingly demanding for the environment. Especially in recent years, the environment is getting worse and worse. More and more ordinary people begin to pay attention to the environmental problems of agricultural farming. How do people know the values of the factors in the environment? The curriculum design emerged as the times require. In view of the shortcomings of the previous acquisition of temperature and humidity equipment, innovative improvements have been made to the traditional equipment to make it meet the requirements and improvements of ordinary agricultural cultivation.

This design mainly aims at the process of agricultural production, innovative improvement of traditional equipment. Intelligent agricultural temperature and humidity acquisition system is mainly divided into three parts: sensing unit, data transmission unit, terminal control and interaction module. The sensor unit has two parts: temperature sensor and humidity sensor. The collected data are transmitted to the control system through the network and socket socket socket for scientific analysis and display on the page.

User data is stored in local and remote server-side databases. Users can view real-time data through the control system, and log in to background data through account password in interactive pages. They can check the temperature and humidity values and the trend changes of data in each time period, which is very suitable for the changing needs in the process of agricultural production.

**Key words:** ZigBee；The embedded；Qt；Temperature and humidity ；

**目录**

[1 绪论](#_Toc31189_WPSOffice_Level1) [1](#_Toc31189_WPSOffice_Level1)

[1.1 选题背景](#_Toc4744_WPSOffice_Level2) [1](#_Toc4744_WPSOffice_Level2)

[1.2 研究现状](#_Toc23677_WPSOffice_Level2) [1](#_Toc23677_WPSOffice_Level2)

[1.3 研究的意义和目的](#_Toc32164_WPSOffice_Level2) [2](#_Toc32164_WPSOffice_Level2)

[2 本次设计所涉及的技术](#_Toc10504_WPSOffice_Level1) [2](#_Toc10504_WPSOffice_Level1)

[2.1 IAR开发环境](#_Toc1549_WPSOffice_Level2) [2](#_Toc1549_WPSOffice_Level2)

[2.2 QT编程](#_Toc11875_WPSOffice_Level2) [2](#_Toc11875_WPSOffice_Level2)

[2.3 zigBee传输技术](#_Toc8477_WPSOffice_Level2) [3](#_Toc8477_WPSOffice_Level2)

[2.4 SQLite数据库](#_Toc13420_WPSOffice_Level2) [4](#_Toc13420_WPSOffice_Level2)

[3 系统整体设计方案](#_Toc24727_WPSOffice_Level1) [4](#_Toc24727_WPSOffice_Level1)

[3.1 功能结构分析](#_Toc27643_WPSOffice_Level2) [4](#_Toc27643_WPSOffice_Level2)

[3.2技术流程](#_Toc5387_WPSOffice_Level2) [5](#_Toc5387_WPSOffice_Level2)

[3.3 各模块功能](#_Toc32272_WPSOffice_Level2) [5](#_Toc32272_WPSOffice_Level2)

[3.3.1 采集模块](#_Toc7223_WPSOffice_Level3) [5](#_Toc7223_WPSOffice_Level3)

[3.3.2 数据传输模块](#_Toc32720_WPSOffice_Level3) [6](#_Toc32720_WPSOffice_Level3)

[3.3.3 网关显示模块](#_Toc31189_WPSOffice_Level3) [6](#_Toc31189_WPSOffice_Level3)

[3.3.4 服务器显示模块](#_Toc10504_WPSOffice_Level3) [6](#_Toc10504_WPSOffice_Level3)

[3.3.5 数据库设计](#_Toc24727_WPSOffice_Level3) [7](#_Toc24727_WPSOffice_Level3)

[4 系统的硬件设计](#_Toc21409_WPSOffice_Level1) [7](#_Toc21409_WPSOffice_Level1)

[4.1 系统整体的硬件设计](#_Toc23569_WPSOffice_Level2) [7](#_Toc23569_WPSOffice_Level2)

[4.2 zigBee原理](#_Toc26065_WPSOffice_Level2) [7](#_Toc26065_WPSOffice_Level2)

[4.3 DHT11温湿度传感器](#_Toc20601_WPSOffice_Level2) [8](#_Toc20601_WPSOffice_Level2)

[4.4 ZigBee协调器与终端的程序设计](#_Toc24381_WPSOffice_Level2) [9](#_Toc24381_WPSOffice_Level2)

[5 系统的测试与调试](#_Toc12753_WPSOffice_Level1) [18](#_Toc12753_WPSOffice_Level1)

[5.1 系统测试步骤](#_Toc13224_WPSOffice_Level2) [18](#_Toc13224_WPSOffice_Level2)

[5.2 调试结果](#_Toc12402_WPSOffice_Level2) [19](#_Toc12402_WPSOffice_Level2)

[6 总结展望](#_Toc9227_WPSOffice_Level1) [19](#_Toc9227_WPSOffice_Level1)

**智能农业温湿度采集系统的设计与实现**

# 1 绪论

## 选题背景

候变化对农业的影响取决于气候变暖的程度。在温度增加不大的变暖条件下，其影响是有利有弊的，随着温度的增加，将导致农业生产的不稳定性增加，局部干旱高温危害加重，由于气候变暖后作物发育期提前，使春季霜冻的危害加大。内蒙古草原区春旱加剧，生产力下降。气象灾害造成的农牧业损失加大。

如果不采取适应措施，到2030年，我国种植业生产能力在总体上可能会下降5%~10%，其中小麦、水稻和玉米三大作物均以下降为主。2050年后受到的冲击会更大，主要作物产量和品质将进一步下降，病虫害加重，肥料和水分的有效性降低，农业使用的化肥和灌溉水量将增加，生产成本将提高。

农业及生态系统是适应气候变化的重点和优先领域。这包括不断提高农业对气候变化的应变能力和抗灾减灾水平；选育抗逆品种，采用稳产增产技术；发展包括生物技术在内的新技术；科学地调整种植制度，适应气候变暖。

## 1.2 研究现状

气候变化对作物病虫害的影响。据统计，我国农业产值因病虫害造成的损失大约为农业总产值的20％～25％。气候变暖会使农业病虫害的分布区发生变化。低温往往限制某些病虫害的分布范围，气温升高后，这些病虫害的分布区可能扩大，从而影响农作物生长。同时温室效应还使一些病虫害的生长季节延长，使害虫的繁殖代数增加，一年中危害时间延长，作物受害可能加重。

气候变化对主要农作物产量的影响。对产量的影响可能主要来自于极端气候事件频率的变化。同时，增温直接影响光合作用速率和呼吸速率这两个决定作物生产率的主要过程。大气中CO2浓度倍增后，温度升高、作物原有品种发育速度加快和生育期缩短是产量下降的主要原因。

## 1.3 研究的意义和目的

温度与湿度不仅与我们的生活密切相关，更加是促进农业生产的一部分。气候的变化每时每刻都在影响着农作物的生长。随着全球热化，越来越多的种植基地由于管理不善，以及对温度与湿度的检测不力，导致农作物无法正常种植。所以，研究一个可行的温湿度采集系统是必要的。

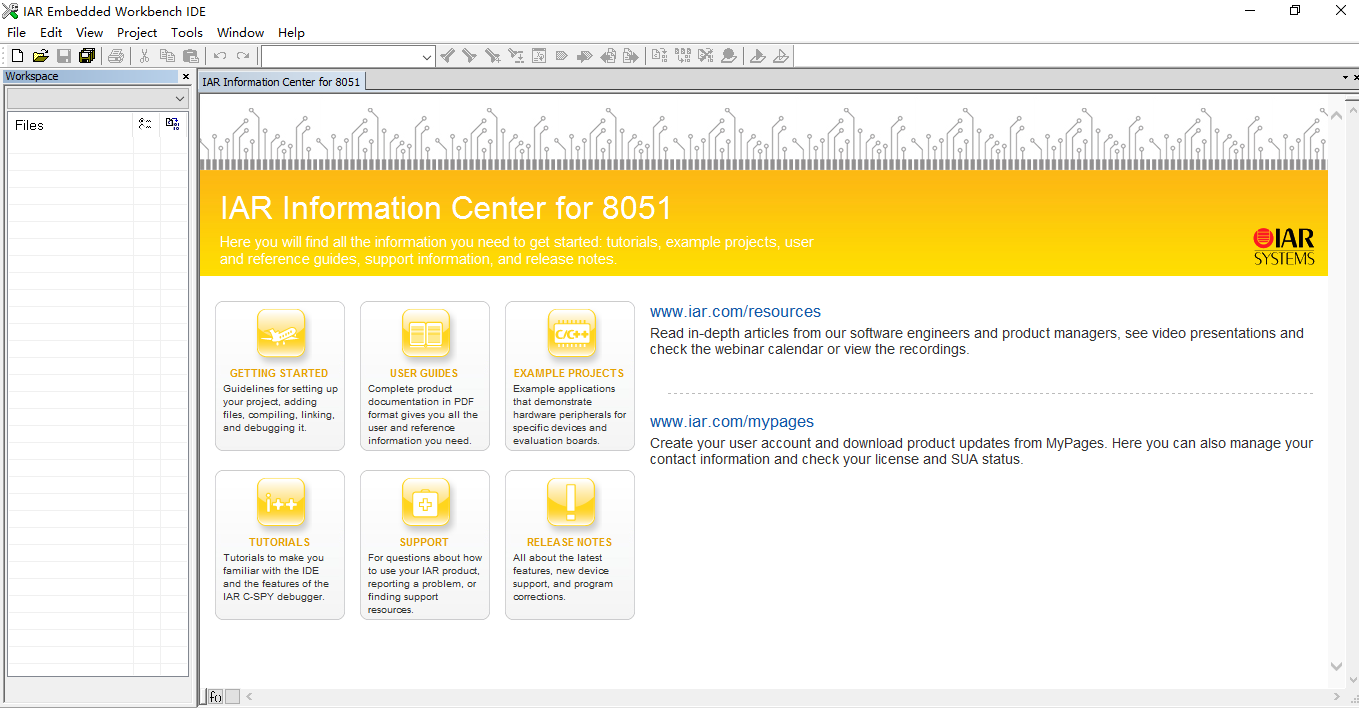
但是由于传统的温湿度采集系统数据传输是有线的方式，这极大地限制了系统的布置和扩展性。本系统突破性地采用zigbee作为数据传输的载体，利用CC2530智能地收集数据并把温度传输到相应的协调器中，zigbee利用其耗电小的优势逐渐在各个领域中发挥应有的作用。切合实际生产的应用。

本系统还通过qt图形化页面实时性地显示数据，并能把数据储存在数据库中，方便主人在后台观测数据的变化，根据数据的走势采取下一步的措施。并能对数据库进行管理操作，极大地方便地用户的操作和使用。

# 2 本次设计所涉及的技术

## 2.1 IAR开发环境

硬件开发用的是Windows环境下的IAR9.10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所示。

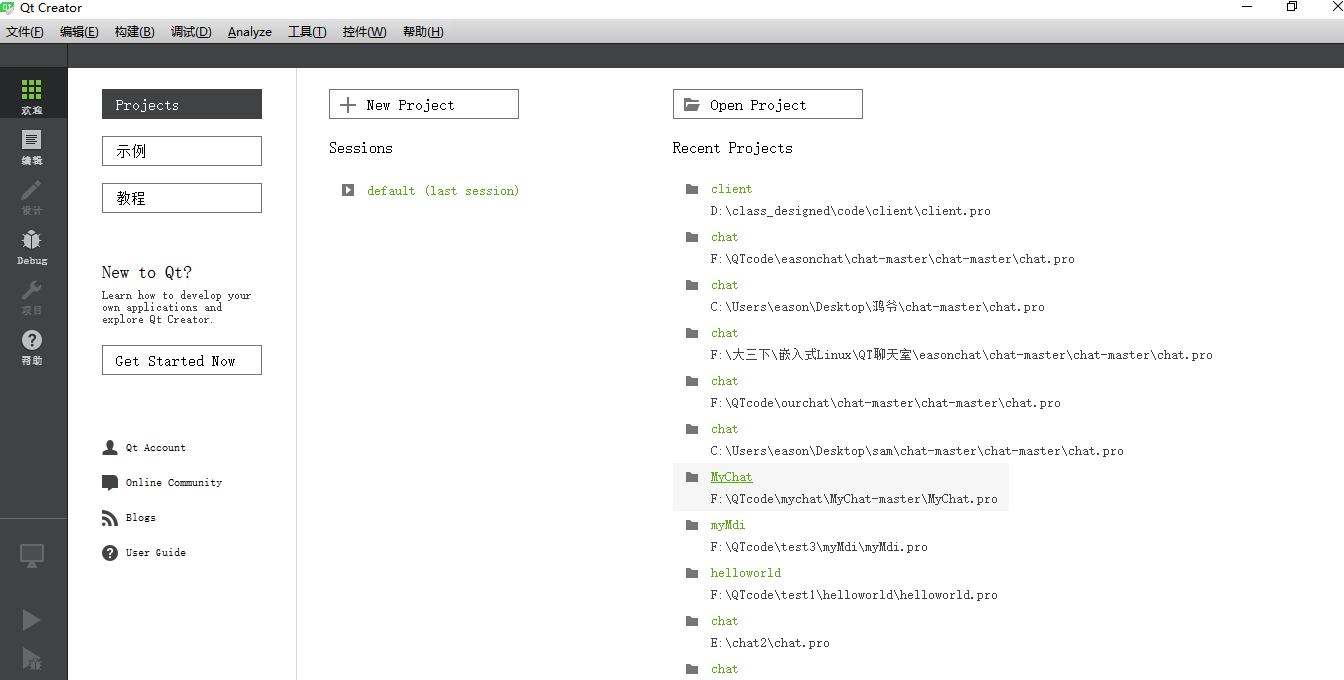


2.2 QT编程

Qt 是一个1991年由Qt Company开发的跨平台[C++](https://baike.baidu.com/item/C++" \t "https://baike.baidu.com/item/qt/_blank)[图形用户界面](https://baike.baidu.com/item/%E5%9B%BE%E5%BD%A2%E7%94%A8%E6%88%B7%E7%95%8C%E9%9D%A2" \t "https://baike.baidu.com/item/qt/_blank)应用程序开发框架。它既可以开发[GUI](https://baike.baidu.com/item/GUI" \t "https://baike.baidu.com/item/qt/_blank)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](https://baike.baidu.com/item/%E8%AF%BA%E5%9F%BA%E4%BA%9A" \t "https://baike.baidu.com/item/qt/_blank)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](https://baike.baidu.com/item/iOS/45705" \t "https://baike.baidu.com/item/qt/_blank)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](https://baike.baidu.com/item/iOS/45705" \t "https://baike.baidu.com/item/qt/_blank)、[Android](https://baike.baidu.com/item/Android/60243" \t "https://baike.baidu.com/item/qt/_blank)、[WP](https://baike.baidu.com/item/WP/5165077" \t "https://baike.baidu.com/item/qt/_blank),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](https://baike.baidu.com/item/X%20Window" \t "https://baike.baidu.com/item/qt/_blank) 上的 Motif，Openwin，GTK 等图形界 面库和 Windows 平台上的 [MFC](https://baike.baidu.com/item/MFC/2236974" \t "https://baike.baidu.com/item/qt/_blank)，OWL，VCL，ATL 是同类型的东西。

本次实验使用终端的图形化设计，在控制系统中加入QT编程的页面，方便用户观察数据的变化。在服务器加入QT图形化设计，能实时地显示在某一个时间段传感器的数值，在把数值传输到数据库中，方便用户的管理以及后期分析，

QT Creator的界面如图二所示：



## 2.3 zigBee传输技术

ZigBee是一个由可多到65535个无线数传模块组成的一个无线数传网络平台，在整个网络范围内，每一个ZigBee网络数传模块之间可以相互通信，每个网络节点间的距离可以从标准的75m无限扩展。

与移动通信的CDMA网或GSM网不同的是，ZigBee网络主要是为工业现场自动化控制数据传输而建立，因而，它必须具有简单，使用方便，工作可靠，价格低的特点。而移动通信网主要是为语音通信而建立，每个基站价值一般都在百万元人民币以上，而每个ZigBee“基站”却不到1000元人民币。

每个ZigBee网络节点不仅本身可以作为监控对象，例如其所连接的传感器直接进行数据采集和监控，还可以自动中转别的网络节点传过来的数据资料。除此之外，每一个ZigBee网络节点(FFD)还可在自己信号覆盖的范围内，和多个不承担网络信息中转任务的孤立的子节点(RFD)无线连接。

## 2.4 SQLite数据库

本次设计服务器端使用SQLite数据库，通过QT能兼容该数据库，是QT编程的首选，该数据库轻量型，适合在嵌入式中使用。不仅如何，该种数据库独立，没有依赖，无需安装。并且能跨平台，支持众多的操作系统。支持NULL，INTEGER，Real，text，blob数据类型，当某个值插入到数据库是，SQlite将会检查他的类型，如果该类型与关联的列不匹配，SQlite则会尝试将改制转换成该列的类型，如果不能转换，则该值将作为本身的类型储存。综合上述特点，SQLite适合本次设计的运用。

# 3 系统整体设计方案

## 3.1 功能结构分析

该温湿度监控管理系统主要包括温湿度传感器、ZigBee模块、虚拟机上的Qt软件以及服务器上的Qt软件六部分组成。

系统会在某区域安装一些温湿度传感器，传感器通过ZigBee无线传输将数据传给虚拟机的Linux系统，用户只需在Linux上就可以实时观察到该区域的温湿度数据，当温度或湿度高于一个数值时，用户可以按照实际的环境温度和湿度，采取下一步急救措施。同时，Linux也会将数据通过TCP协议传输给服务器的Qt软件，并将数据存放在服务器的数据库中。用户可以在服务器端实时查看温湿度情况，同时也能通过访问数据库查看之前的温湿度情况。同时，用户能对数据库进行管理，可清除数据库中的数据，精简数据库内容，方便下一步分析。

## 3.2技术流程

局域网

通信

串口

通信

打开电源

加载zigBee协议栈

协调器与终端建立链接

温湿度传感器采集数据

Linux系统

服务器

如上图所示，当协调器与终端打开电源时，CC2530内部会加载对应的协议栈代码，并通过内部事先设置好的网络号，协调器与终端能建立无线连接，终端触发温湿度传感器工作，每隔1秒在环境中采集温度和湿度，终端收集温度和湿度后，通过zigBee无线传输把数据传输带协调器中，协调器通过串口把数据传输到Linux中，Linux通过读取串口中的数据在QT图形化页面中实时显示出温度和湿度的数值。Linux打开线程通讯后，通过局域网与服务器建立连接，Linux把每一秒的温度和湿度通过套接字传送到服务器中，服务器能通过数据库存储信息。

## 3.3 各模块功能

### 3.3.1 采集模块

该项目使用的温湿度传感器是DHT11数字温湿度传感器，这是一款含有已校准数字信号输出的温湿度复合传感器，它传输的数据格式为：8bit湿度整数数据+8bit湿度小数数据+8bi温度整数数据+8bit温度小数数据+8bit校验和。DHT11的数据引脚与ZigBee终端的p0\_7引脚相连，正极和负极分别接到ZigBee终端上的电源和地。

### 3.3.2 数据传输模块

数据传输模块有一个有线传输和两个无线传输模块。无线模块：ZigBee无线传输模块，局域网传输；有线传输：串口传输。

Zigbee模块来完成温湿度采集终端与协调器的无线通信，即建立一个ZigBee网络。串口模块完成协调器与Linux系统的数据通信。局域网模块完成Linux系统与服务器的连接，使数据通过套接字传送到服务器中。

服务器

采集器

无线

无线单片机CC2530

协调器

串口连接

Linux

局域网

### 3.3.3 网关显示模块

该Qt软件的主要功能有：

1. 通过串口接收温湿度的数据，并将接收到的温度和湿数据显示在屏幕上。
2. 设置两个按钮，可与服务器建立连接或者断开连接。

## 3.3.4 服务器显示模块

该Qt软件的主要功能有：

(1)与网关进行网络通信，并接收网关发送过来的温湿数据，将数据显示出来。

(2)将接收到的温湿度数据存放到数据库中，同时也可以从数据库获取历史数据并显示出来。

服务器端是本毕业设计的重要构成，是数据管理、存储的基础。服务器为移动监护端提供远程监护查询、用户信息文件的创建和管理和预警的功能。

连接Linux系统

储存数据库

显示温湿度

服务器

### 3.3.5 数据库设计

在该项目中，服务器将接收到的温湿度数据存放到SQLite中，其中存放温湿度数据的表的设计由4个字段组成，分别是：年月日、时分秒、温度、湿度。

# 4 系统的硬件设计

## 4.1 系统整体的硬件设计

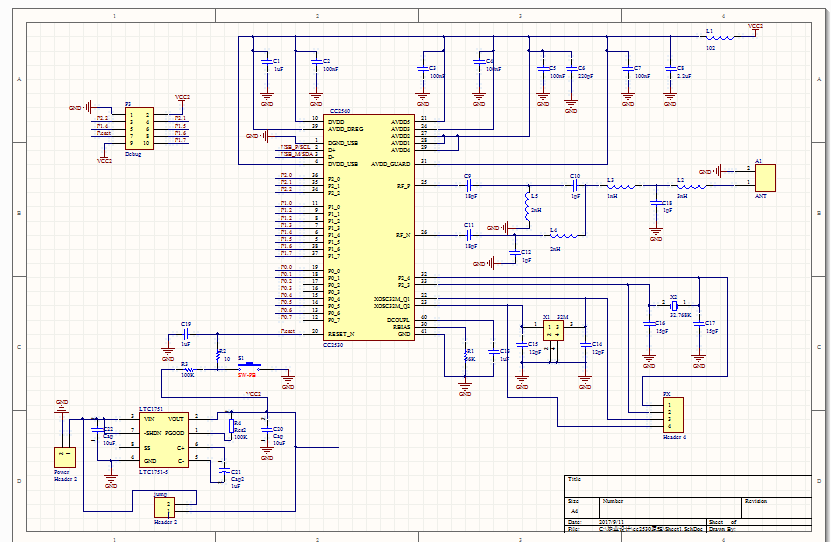
该系统包括的硬件主要有：DHT11温湿度传感器、zigBee终端、zigBee协调器。

## 4.2 zigBee原理

ZigBee 是一种低速无线个域网技术(Lciw Rate Wireless Personal Network,LRWPAN)。它使用与通信数据量不大,数据传输数率相对较低,分布范围较小,但是对数据的安全可靠有一定要求,而且要求成本和功耗非常低,并容易安装使用的场合。它具有可靠性高、安全性好、低功耗、低传输速率、低成本、灵活的工作频段等特点。

本项目使用的ZigBee模块是以CC2530为芯片的ZigBee模块，CC2530是TI公司在2.4GHz频段推出的第二代支持IEEE802.15.4/ZigBee协议的片上系统芯片。其内部集成了高性能射频收发器、工业标准增强型8051MCU内核、256KBFlash ROM和8KB RAM。

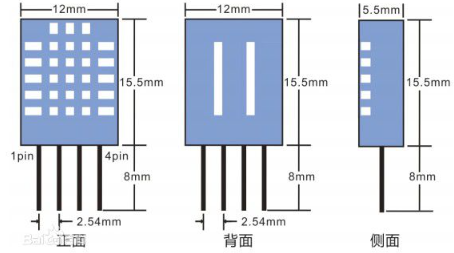
CC2530电路图如下：



## 4.3 DHT11温湿度传感器

DHT11数字温湿度传感器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。

DHT11温湿度传感器实物图如下：



## 4.4 ZigBee协调器与终端的程序设计

协调器启动之后，首先组建ZigBee网络并初始化串口，然后在串口回调函数rxCB()中调用HalUARTRead(0,uartbuf,1)函数读取串口数据，然后调用Senda()函数将接收到的数据广播给各个ZigBee终端。然后，在消息处理函数GenericApp\_ProcessEvent()中，当接收到无线数据时，就会执行case AF\_INCOMING\_MSG\_CMD：中调用的事件处理函数GenericApp\_MessageMSGCB（MSGpkt），在该函数中调用HalUARTWrite(0,buffer,5)将温湿度数据发送给串口。

GenericApp\_MessageMSGCB(afIncomingMSGPacket\_t \*pkt)函数解析：该函数的功能是处理无线接收到的数据，从消息队列中将数据复制到一个数组中，然后将数据写入串口。

osal\_memcpy(buffer,pkt->cmd.Data,5);

HalUARTWrite(0,buffer,5);//往串口写入温湿度数据

终端主程序流程

终端启动之后，先加入协调器组建的ZigBee网络，当ZigBee终端的网络状态发生变化时，就会定时让事件SEND\_DATA\_EVENT发生，通过函数 osal\_set\_event(GenericApp\_TaskID,SEND\_DATA\_EVENT)来实现，当该事件发生时，就会执行函数GenericApp\_SendTheMessage()将温湿度数据发送给协调器。当ZigBee终端接收到协调器发来的无线数据时，就会执行case AF\_INCOMING\_MSG\_CMD：中调用的事件处理函数GenericApp\_MessageMSGCB（MSGpkt）。

主要函数解析：

GenericApp\_ProcessEvent( uint8 task\_id, uint16 events )任务处理函数:该函数主要有两个功能，第一，当终端加入协调器组建的网络时，网络状态就会发生改变，就会执行case ZDO\_STATE\_CHANGE:中的函数。第二，当终端接收到无线数据之后，就会执行case AF\_INCOMING\_MSG\_CMD:中的函数。

case AF\_INCOMING\_MSG\_CMD:

GenericApp\_MessageMSGCB(MSGpkt);//调用任务处理函数

break;

case ZDO\_STATE\_CHANGE:

GenericApp\_NwkState = (devStates\_t)(MSGpkt->hdr.status);//取得ZigBee终端的网络状态

if ( GenericApp\_NwkState == DEV\_END\_DEVICE||

GenericApp\_NwkState == DEV\_ROUTER||

GenericApp\_NwkState == DEV\_ZB\_COORD)

{

osal\_set\_event(GenericApp\_TaskID,SEND\_DATA\_EVENT);

//设置定时器

}

void GenericApp\_SendTheMessage( void )函数：该函数的功能是将温湿度数据通过无线网络发送给协调器。

void GenericApp\_SendTheMessage( void )

{

afAddrType\_t my\_DstAddr;

my\_DstAddr.addrMode=(afAddrMode\_t)Addr16Bit;

my\_DstAddr.endPoint=GENERICAPP\_ENDPOINT;

my\_DstAddr.addr.shortAddr=0x0000;

DHT11();

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

wen[1]=wendu\_ge%10+0x30;

wen[2]=':';

wen[3]= shidu\_shi+0x30;

wen[4]=shidu\_ge%10+0x30;

AF\_DataRequest( &my\_DstAddr, &GenericApp\_epDesc,

GENERICAPP\_CLUSTERID,

osal\_strlen(wen)+1,

wen,

&GenericApp\_TransID,

AF\_DISCV\_ROUTE, AF\_DEFAULT\_RADIUS); }

详细代码如下：

#include <stdio.h>

#include "OSAL.h"

#include "AF.h"

#include "ZDApp.h"

#include "ZDObject.h"

#include "ZDProfile.h"

#include <string.h>

#include <ioCC2530.h>

//#include "GenericApp.h"

#include "DebugTrace.h"

#include "DH.h"

#if !defined( WIN32 )

#include "OnBoard.h"

#endif

/\* HAL \*/

#include "hal\_lcd.h"

#include "hal\_led.h"

#include "hal\_key.h"

#include "hal\_uart.h"

#define SEND\_DATA\_EVENT 0x01

#define DATA\_PIN P0\_7

void Delay\_us(void); //1 us延时

void Delay\_10us(void); //10 us延时

void COM(void); // 温湿写入

void DHT11(void) ; //温湿传感启动

#define uchar unsigned char

#define uint unsigned int

//温湿度定义

uchar shidu\_shi,shidu\_ge,wendu\_shi,wendu\_ge;

uchar ucharFLAG,uchartemp;

uchar ucharT\_data\_H,ucharT\_data\_L,ucharRH\_data\_H,ucharRH\_data\_L,ucharcheckdata;

uchar ucharT\_data\_H\_temp,ucharT\_data\_L\_temp,ucharRH\_data\_H\_temp,ucharRH\_data\_L\_temp,ucharcheckdata\_temp;

uchar ucharcomdata;

//--------------------------------

//-----湿度读取子程序 ------------

//--------------------------------

//----以下变量均为全局变量--------

//----温度高8位== U8T\_data\_H------

//----温度低8位== U8T\_data\_L------

//----湿度高8位== U8RH\_data\_H-----

//----湿度低8位== U8RH\_data\_L-----

//----校验 8位 == U8checkdata-----

//----调用相关子程序如下----------

//---- Delay();, Delay\_10us();,COM();

//--------------------------------

//延时函数

void Delay\_us() //1 us延时

{

MicroWait(1);

}

void Delay\_10us() //10 us延时

{

MicroWait(10);

}

void Delay\_ms(uint Time)//n ms延时

{

unsigned char i;

while(Time--)

{

for(i=0;i<100;i++)

Delay\_10us();

}

}

//温湿度传感

void COM(void) // 温湿写入

{

uchar i;

for(i=0;i<8;i++)

{

ucharFLAG=2;

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

Delay\_10us();

Delay\_10us();

Delay\_10us();

uchartemp=0;

if(DATA\_PIN)uchartemp=1;//如果超过30us还没置0，则数据为1，置0，则数据为0

ucharFLAG=2;

while((DATA\_PIN)&&ucharFLAG++); //等待置0

if(ucharFLAG==1)break;

ucharcomdata<<=1;//ucharcomdata左移一位赋值给ucharcomdata

ucharcomdata|=uchartemp;//两句合起来就是给最高位赋值

}

}

void DHT11(void) //温湿传感启动

{

DATA\_PIN=0;//置低电平

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

DATA\_PIN=1; //置高电平，开始信号发送结束

P0DIR &= ~0x80; //重新配置IO口方向，输入

Delay\_10us();

Delay\_10us();

Delay\_10us();

Delay\_10us(); //高电平，延时40us等待

if(!DATA\_PIN) //如果电平变低，则证明检测到开始信号

{

ucharFLAG=2;

while((!DATA\_PIN)&&ucharFLAG++);//低电平，等待响应结束

ucharFLAG=2;

while((DATA\_PIN)&&ucharFLAG++); //高电平，准备输出

COM();

ucharRH\_data\_H\_temp=ucharcomdata;//湿度高八位

COM();

ucharRH\_data\_L\_temp=ucharcomdata;//湿度低八位

COM();

ucharT\_data\_H\_temp=ucharcomdata;//温度高八位

COM();

ucharT\_data\_L\_temp=ucharcomdata;//温度低八位

COM();

ucharcheckdata\_temp=ucharcomdata;//校验八位

DATA\_PIN=1; //置高，空闲状态

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

if(uchartemp==ucharcheckdata\_temp)//校验

{

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

}

// This list should be filled with Application specific Cluster IDs.

const cId\_t GenericApp\_ClusterList[GENERICAPP\_MAX\_CLUSTERS] =

{

GENERICAPP\_CLUSTERID

};

const SimpleDescriptionFormat\_t GenericApp\_SimpleDesc =

{

GENERICAPP\_ENDPOINT, // int Endpoint;

GENERICAPP\_PROFID, // uint16 AppProfId[2];

GENERICAPP\_DEVICEID, // uint16 AppDeviceId[2];

GENERICAPP\_DEVICE\_VERSION, // int AppDevVer:4;

GENERICAPP\_FLAGS, // int AppFlags:4;

0,

(cId\_t \*)NULL, // 改 byte \*pAppInClusterList;

GENERICAPP\_MAX\_CLUSTERS, // byte AppNumInClusters;//输入簇ID的个数

(cId\_t \*)GenericApp\_ClusterList, // byte \*pAppInClusterList;

};

// This is the Endpoint/Interface description. It is defined here, but

// filled-in in GenericApp\_Init(). Another way to go would be to fill

// in the structure here and make it a "const" (in code space). The

// way it's defined in this sample app it is define in RAM.

endPointDesc\_t GenericApp\_epDesc;

byte GenericApp\_TaskID;

byte GenericApp\_TransID; // This is the unique message ID (counter)

devStates\_t GenericApp\_NwkState;//用于保存节点状态信息

void GenericApp\_MessageMSGCB( afIncomingMSGPacket\_t \*pckt );//消息处理函数

void GenericApp\_SendTheMessage( void );

void GenericApp\_Init( byte task\_id )

{

P1\_0=0;

P1\_1=0;

GenericApp\_TaskID = task\_id;

GenericApp\_NwkState = DEV\_INIT;//节点状态初始化为DEV\_INIT，表示未连接到zigbee上

GenericApp\_TransID = 0;

GenericApp\_epDesc.endPoint = GENERICAPP\_ENDPOINT;

GenericApp\_epDesc.task\_id = &GenericApp\_TaskID;

GenericApp\_epDesc.simpleDesc

= (SimpleDescriptionFormat\_t \*)&GenericApp\_SimpleDesc;

GenericApp\_epDesc.latencyReq = noLatencyReqs;

}

UINT16 GenericApp\_ProcessEvent( byte task\_id, UINT16 events )

{

afIncomingMSGPacket\_t \*MSGpkt;

if ( events & SYS\_EVENT\_MSG )

{

MSGpkt = (afIncomingMSGPacket\_t \*)osal\_msg\_receive( GenericApp\_TaskID );

//读取节点的设备类型

while ( MSGpkt )

{

switch ( MSGpkt->hdr.event )

{

case ZDO\_STATE\_CHANGE:

case AF\_INCOMING\_MSG\_CMD:

GenericApp\_MessageMSGCB(MSGpkt);

GenericApp\_NwkState = (devStates\_t)(MSGpkt->hdr.status);

if (GenericApp\_NwkState == DEV\_END\_DEVICE )

{

// Start sending "the" message in a regular interval.

GenericApp\_SendTheMessage();

osal\_set\_event(GenericApp\_TaskID,SEND\_DATA\_EVENT);

}

break;

default:

break;

}

// Release the memory

osal\_msg\_deallocate( (uint8 \*)MSGpkt );

// Next

MSGpkt = (afIncomingMSGPacket\_t \*)osal\_msg\_receive( GenericApp\_TaskID );

}

// return unprocessed events

return (events ^ SYS\_EVENT\_MSG);

}

if(events & SEND\_DATA\_EVENT)

{

///HalLedBlink(HAL\_LED\_1,0,50,500);

GenericApp\_SendTheMessage();

//HalLedBlink(HAL\_LED\_1,0,50,500);

//check();

osal\_start\_timerEx(GenericApp\_TaskID,SEND\_DATA\_EVENT,1500);//定时1200ms

return(events^SEND\_DATA\_EVENT);

}

//Discard unknown events

return 0;

}

uchar wen[6];

void GenericApp\_SendTheMessage( void )

{

afAddrType\_t my\_DstAddr;

my\_DstAddr.addrMode=(afAddrMode\_t)Addr16Bit;

my\_DstAddr.endPoint=GENERICAPP\_ENDPOINT;

my\_DstAddr.addr.shortAddr=0x0000;

DHT11();

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

wen[1]=wendu\_ge%10+0x30;

wen[2]=':';

wen[3]= shidu\_shi+0x30;

wen[4]=shidu\_ge%10+0x30;

AF\_DataRequest( &my\_DstAddr, &GenericApp\_epDesc,

GENERICAPP\_CLUSTERID,

osal\_strlen(wen)+1,

wen,

//theMessageData,

&GenericApp\_TransID,

AF\_DISCV\_ROUTE, AF\_DEFAULT\_RADIUS);

//HalLedBlink(HAL\_LED\_1,0,50,500);

//HalLedBlink(HAL\_LED\_1,0,90,500);

}

void GenericApp\_MessageMSGCB( afIncomingMSGPacket\_t \*pkt )//消息处理

{

HalLedBlink(HAL\_LED\_1,0,50,500);

unsigned char buffer[2]={'0'};

switch ( pkt->clusterId )

{

case GENERICAPP\_CLUSTERID:

// "the" message

osal\_memcpy(buffer,pkt->cmd.Data,1);//将数据拷贝到缓冲区

if(buffer[0]=='2'){

//HalLedBlink(HAL\_LED\_1,0,50,500);

P1\_0=1;

P1\_1=0;

//HalLedBlink(HAL\_LED\_1,0,50,500);

//osal\_memcpy(theMessageData,"OFF",20);//将数据拷贝到缓冲区

}

else if(buffer[0]=='3'){

P1\_0=0;

P1\_1=0;

HalLedBlink(HAL\_LED\_1,0,50,500);

}

//HalUARTWrite(0,buffer,20);

break;

}

}

# 5 系统的测试与调试

## 5.1 系统测试步骤

（1）检查开发板、串口线以及外扩设备连接是否正常。

（2）下载代码到开发板。

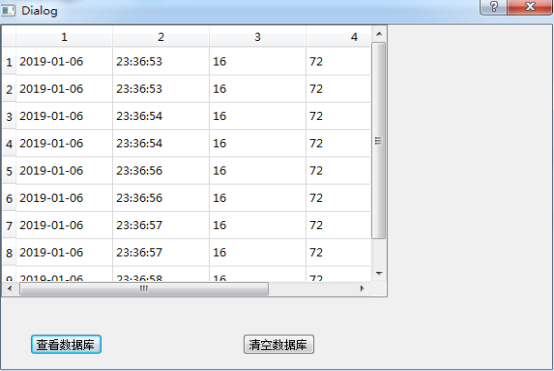
（3）下载传感器节点代码到开发板。

（4）用串口调试助手观察协议栈运行是否正常。

（5）测试上位机软件，PC端能否正常接收数据以及能否将数据显示出来。

## 5.2 调试结果

****

****

如上图所示，是本系统界面应用程序测试结果，其能实时显示当前环境下传感器所测得温湿度数值，数据库能正确录入所测数据。经过多次试验测试，本系统功能有效可用。

# 6 总结展望

在这次基于Zigbee的智能温湿度采集系统的课程设计中，通过本组查阅资料、阅读文献、以及动手实践对Zigbee技术有了全新的认识。Zigbee及技术具有近距离、低复杂度、低功耗、低速率、低成本、等优点，因而成本和功耗方面的是本设计的一大优势。同时，由于Zigbee技术组网方便，网络容量大，应用前景比较广泛。

Zigbee在当今智能行业中有非常大的发展空间。通过课程设计，将当下科技发展趋势与知识结合，回归生活和应用。促进了对Zigbee技术的理解同时能够把所学运用到生活和实践。也极大的激发了设计的灵感和兴趣。

在本次实验中也遇到不少难题，ZigBee技术的协议以及相关的知识对于我们来说都是比较新颖的，了解其技术和协议时大家花费了一定的时间和精力。硬件里的大部分程序是在IAR完成编写和调试的。刚开始在程序调试的总是出现问题，温湿度的显示都是乱码，后来经过仔细分析后，找到了代码的错误，并及时修正。解决难题的过程，也是不断提高自身的过程。ZigBee技术的应用前景是十分明朗的，成本和功耗方面的优势使其在市场中十分具有竞争力。尤其在物联网技术已成为当下热点命题之一，ZigBee技术的应用价值就更为重要了。伴随着物联网技术的成长，ZigBee技术也将日趋成熟。

同时，也非常感谢在这次试验中刘老师提供的帮助和支持，这次一个非常有挑战性的课程设计让我们更进一步了解和学习了Zigbee的技术，同时在试验过程中也提高了自己解决问题的能力。