

分类号： TP391.41

密 级：

单位代码： 10431

学 号： 2016025

齐鲁工业大学

工程硕士学位论文

基于 STM32 的温室环境监测和控制系统

作 者 姓 名 _____ 郭磊 _____

工 程 领 域 _____ 控制工程 _____

所 在 学 院 _____ 电气工程与自动化学院 _____

指导教师姓名 _____ 孙宝江 _____

专业技术职务 _____ 教授 _____

2021 年 06 月 10 日

分类号： TP391.41

密 级：

单位代码： 10431

学 号： 2016025

齊魯工業大學

工程硕士学位论文

基于 STM32 的温室环境监测和控制系统

作 者 姓 名 _____ 郭磊

工 程 领 域 _____ 控制工程

所 在 学 院 _____ 电气工程与自动化学院

指导教师姓名 _____ 孙宝江

专业技术职务 _____ 教授

2021 年 06 月 10 日

**A Thesis Submitted for the Application of
the Project Master's Degree**

**Greenhouse environment monitoring and
control system based on STM32**

Candidate: Guo Lei

Specialty: Control engineering

Supervisor: Professor Sun Baojiang

Qilu University Of Technology, Jinan, China

June, 2021

目 录

摘 要	I
ABSTRACT	II
第 1 章 绪论	1
1.1 论文背景与研究意义	1
1.2 国内外发展现状	2
1.2.1 国外温室	2
1.2.2 国内温室	2
1.3 本文的研究内容	4
1.4 论文的组织结构	4
第 2 章 系统的实施方案和应用的技术	5
2.1 系统的功能分析与设计	5
2.2 ZigBee 技术	6
2.2.1 ZigBee 技术概述	6
2.2.2 ZigBee 网络协议	6
2.2.3 ZigBee 网络拓扑结构	7
2.3 NB-IoT 技术	8
2.3.1 NB-IoT 技术概述	8
2.3.2 NB-IoT 技术的特点	9
2.3.3 ZigBee、LoRa、NB-IoT 比较	9
2.4 嵌入式系统	10
2.4.1 嵌入式系统概述	10
2.4.2 嵌入式系统的几个特征	10
2.4.3 嵌入式系统的分类	11
2.4.4 μ COS 嵌入式操作系统	12
第 3 章 系统硬件电路设计	15

3.1 系统总体硬件设计	15
3.2 采集节点部分硬件设计	15
3.2.1 单片机选型与设计	16
3.2.2 ZigBee 模块选型与设计	19
3.2.3 传感器选型与设计	23
3.2.4 采集节点部分 PCB 电路板图设计	26
3.3 通信控制器硬件设计	27
3.3.1 处理器选型与设计	27
3.3.2 NB-IoT 选型与设计	28
3.3.3 通信控制器部分 PCB 电路板图设计	32
第 4 章 系统软件设计	33
4.1 ZigBee 无线通信协议模块软件设计	33
4.1.1 概述	33
4.1.2 Zigbee 组网	33
4.1.3 通信协议	33
4.2 采集节点模块软件设计	35
4.2.1 温湿度采集部分程序设计	35
4.2.2 光照强度采集部分程序设计	37
4.2.3 MG811 二氧化碳浓度采集部分程序设计	39
4.2.4 单片机程序设计	41
4.3 通信控制器部分程序设计	45
4.3.1 STM32 处理器的程序设计	45
4.3.2 基于 μ CosII 的软件设计与移植	47
4.4 云平台设计	50
4.4.1 WH-NB75 的工作模式	50
4.4.2 有人透传云平台设计	53
4.5 农作物生长模型建立	55
第 5 章 系统测试结果与分析	57
5.1 无线数据通信测试	57
5.2 系统功能测试	57

第 6 章 总结与展望.....	61
参考文献.....	63
致 谢.....	67
在学期间主要科研成果.....	69
一、发表学术论文	69
二、其他科研成果	69

摘 要

我国淮河以北的地区冬季气温较低,为了满足种植物生长对环境需要,建设了很多温室。近几年,伴随着农业生产方式的改变,农村经济不断发展,渐渐从粗放型发展为集约型,在一些农村经济发展较快的地区,建设了一部分示范区,重点发展科技、生态产业,成为行业的标杆。这些变化都得益于信息技术的进步,为高科技智能化温室创造了便利条件。提高了温室的智能化,不仅使农作物产量和质量得到提升,还改变了温室作物的生长周期,使反季节瓜果蔬菜种类增多,满足了人们的需求。利用信息技术可以实现远程监测和控制,进而减轻生产人员的劳动强度,降低对从业人员种植水平的要求。

目前应用中的监测装置结构简单,仅仅实现监测数据的传输,未在系统内部建立种植物实际生长环境参数模型,也未实现根据导入的参数模型对环境因子进行自动调控,占用了很多人力和财力,却没有收到很好的效果。尤其是单个种植户管理的温室数量越来越多,管理难度增大,他们需要一种可以远程管理的自动控制设备作为辅助。结合已有问题和实际需要,本文应用STM32作为总控制器处理器,STC单片机作为采集部分的处理器,选用可以检测温室空气参数的传感器作为输入,每个温室里装有一个ZigBee设备和总控制器传送数据,园区内所有ZigBee设备组成内部网,总控制器装有一个运用NB-IoT技术的无线模块,将园区状态传到手持终端。这样设计的目的是利用局域网的稳定可靠优势,一个园区一个远程传输模块,既节省成本又保证通信的可靠性。在移动终端利用有人透传云平台接收总控制器传输的数据,并显示出来,将采集到的数据与设置的阈值比较,超过阈值显示报警信息。将常见温室种植农作物生长过程中的适宜环境参数加载到软件中,供生产人员参考,降低生产人员的工作难度。设备运用的技术涵盖嵌入式系统、传感器技术、短距离窄带远程通信等内容,设备测试结果表明传输数据准确,响应迅速,数据传输效果稳定可靠,克服了传统监测设备的弊端,达到了预期目标。

系统较之前的监测设备,具有监测结果误差小,响应快,监测范围大、智能化程度高等优点。主要创新点是可以实时保存种植物生长过程中的环境参数,建立实际生长参数模型,还可以导入生长参数模型实现环境因子的自动调控,将多个温室组成局域网,提高农业生产的先进性,在农业经济发展理论及应用研究方面都具有重要意义。

关键词: 温室环境参数; 嵌入式技术; 传感器技术; ZigBee 通信; NB-IoT 窄带通信

ABSTRACT

In order to meet the needs of growing plants, many greenhouses have been built because of the low temperature in winter in the areas north of Huaihe River in China. In recent years, with the change of agricultural production mode, agriculture and rural economy continue to develop, gradually from extensive development to intensive development. In areas with relatively good agricultural development, some demonstration areas have been built, focusing on the development of science and technology, ecological agricultural industry, which has become the benchmark of the industry. These changes benefit from the progress of information technology, which created basic conditions for high-tech intelligent greenhouse. Improved greenhouse intelligence, not only to improve crop yield and quality, but also change the production cycle of greenhouse crops, so that off-season fruits and vegetables to meet people's needs. Information technology can realize remote monitoring and control, thus reducing the technical requirements and labor intensity of production personnel.

At present, the monitoring device is simple in structure, and only realizes the transmission of monitoring data, without establishing the actual growth environment parameter model of plants in the system, nor realizing the automatic regulation of environmental factors according to the imported parameter model, which takes up a lot of manpower and financial resources, but does not get good results. In particular, the number of greenhouses managed by individual growers is increasing, and the management difficulty is increasing, so they need an automatic control equipment which can be managed remotely as an assistant. Combined with the existing problems and the actual need, this paper applies STM32 processor as the general controller, on STC microcontroller as the processor of acquisition part, choose the sensors can detect the greenhouse air parameters as input, each greenhouse is equipped with a ZigBee devices and total controller transmitting data, the park all ZigBee equipment Intranet, The master controller is equipped with a wireless module using NB-IoT technology to transmit the park status to the handheld terminal. The purpose of this design is to make use of the stable and reliable advantages of LAN, a remote transmission module in one park, which saves cost and ensures the reliability of communication. In the mobile terminal, the data transmitted by the total controller is received by the human-transmitted cloud platform and displayed. The collected data is compared with the set threshold value and the alarm information is displayed when the threshold value

is exceeded. The suitable environmental parameters in the growth process of common greenhouse crops are loaded into the software for the reference of production personnel, which reduces the difficulty of work for production personnel. The technology used by the equipment covers embedded system, sensors, short-range and narrow-band remote communication, etc. The test results of the equipment show that the transmission data is accurate, the response is rapid, the data transmission effect is stable and reliable, which overcomes the disadvantages of the traditional monitoring equipment and reaches the expected target.

Compared with previous monitoring facilities, the system has the advantages of small error of monitoring results, fast response, large monitoring range and high intelligence. Main innovation point is a real kind of environmental parameters in the process of plant growth and actual growth parameters model is set up, you can import growth parameters model to realize automatic control of environmental factors, can be more greenhouse into local area network (LAN), raise the scientific nature of the agricultural production, agricultural application and theory research is very necessary.

Key words: Greenhouse environmental parameters; embedded technology; sensor technology; ZigBee communication; NB IOT narrowband communication

第1章 绪论

1.1 论文背景与研究意义

当前,农业科技取得了长足的进步,促进了农业的飞速发展,可以对温室内环境参数进行监测的智能温室如雨后春笋,出现在我国很多地区。以瓜果蔬菜为例,随着智能化温室的大量出现,对瓜果蔬菜产业的影响也逐步显现。出现了规模化的蔬菜种植集聚区,降低了生产成本,增加了作业人员的收入,促进了城市周边农业经济的发展。

具备环境参数监测功能是科技大棚的必备条件,其中温湿度、光强和CO₂含量,可以改变种植物的状态。对温室环境进行实时监测,可以实现作业人员通过计算机或移动终端监测温室内农作物的生长状态,进而积累多种种植物的状态模型,还可以用于新品种的试种测试。而且,温室环境监测的一个重要作用在于,通过对环境的监测,种植户可以根据各种农作物生长参数模型,定量调节温室环境因子,使得温室内的因子始终符合种植物生长状态^[1]。尤其是在育苗过程中对环境要求苛刻,使得育苗成功率得到提升,大大减少技术人员数量。

仅仅对环境因子进行检测是不够的,想要改善温室内环境状态,还离不开加湿设备、送风、调温、调光和CO₂机等电气设施,实现自动调节湿度、温度、光强和二氧化碳浓度等参数,使反季节瓜果蔬菜的种植得到普及,进而提高了人们的日常生活水平,增加了反季节瓜果蔬菜的种类,同时也提升了瓜果蔬菜的品质,使种植户和消费者共同受益,得以实现双赢的目的。

近几年,虽然有一些规模比较大的种植户应用了一些进口的计算机智能温室控制设备,例如温室环境监测控制设备、施肥机、喷灌控制设备、流水线育苗设备等电气设施,这些设备减轻了种植户的一些工作强度,但是存在着耗资过多、系统维修流程复杂等各种问题,另外,因为进口设备操控界面和人机交互按钮均为英文,对从业者的要求过高,当前我国大部分农民还不能满足,导致了国外相关产品无法国内得到广泛应用。设计一种廉价、适用型的温室智能化监控设备对于促进我国温室生产自动化以及智能化的发展有着重要的影响,而且也具有广阔的市场前景^[2]。据统计,现在种植户急需廉价、易操控的温室测控设备。针对目前这一现状,以及我国温室数量多、覆盖面广的特征,研究设计一款既与我国农业水平实际相符合又切合于种植户的实际经济状况、技术水平等同于国外相关产品的测控设备是当下课题要完成的内容。

1.2 国内外发展现状

1.2.1 国外温室

二十世纪六十年代,西方国家将电脑应用在大棚测控中。截止1985年,日本的一千余台和荷兰的五千余台计算机用于温室环境监测。

对该领域的探究,国外比国内早很多。模拟量测试仪表出现于二十世纪七十年代,国外开始将该技术用于温室环境测控中,可以指示参数值,人工进行记录和分析。1978年,日本的技术人员第一次设计出基于微型处理器的温室环境测控设备;八十年代开始出现数字量测量表,仍然需要人工进行记录和分析;八十年代末,计算机应用更广泛,分布式监测设备随之得到应用;近期国外开始着手设计一种应用现代计算机技术的多因子参数测控设备。近期,包括日本和欧美在内的发达国家可以对温室内的多个环境参数进行测量,然后根据农作物的生长习性对温室环境进行有效控制。美国、荷兰还应用了温度调控设备,可以对瓜果蔬菜的成熟期和花卉的开花日期进行调控,使人们的多样化需求得到满足。今后设计逐步向着科技化、自主化、多元化演变,更好节省人力成本^[3]。日本还运用了无线设备,实现了对大棚环境的远距离观测^[4]。英国农业相关的管理部门在农业相关的科研院所建立了温室大棚实验室,对包括温湿度、光强、CO₂含量在内的大棚封闭空间小气候进行数据模型建立、分析,并对执行机构进行了最优化的改良。

1.2.2 国内温室

在我国,温室最早出现于汉代,种植的主要农作物有大葱、韭菜等蔬菜,二十世纪三十年代,我国开始出现仅仅利用太阳光照不进行加温的近代温室大棚,二十世纪七十年代末,八十年代初大量的温室大棚开始出现^[5],截止到2000年,国内存量为120万平方米,达到世界排名第一的位置,尤其是在种植技术、建设技术和周边辅助设备的完善方面得到了快速发展,开始出现多种层次、多种体系产品品牌,行业占比得到进一步提高。

1975年,计算机在国内开始应用,到1980年,开始应用于农业生产,利用计算机对温室进行管理和测控的情景开始出现^[6],自二十世纪八十年代以来,国内技术研发人员开始借鉴国外智能化温室技术,首代产品仅仅可以测量一个参数,但是后期发现这一方案存在一定局限性。

1987年,中国农业科学院早期在国外购入温室管理设备FELIXC 512,国内出现首个计算机技术应用于温室相关的案例^[7]。1990年,计算机在大棚参数测量和控制领域开始得到运用。

2000年,金钰制作了一种以IPC为核心的智能温室测控设备^[8]。该设计是利用工业控制机作为主控制器对环境参数进行采集,控制器外接执行驱动机构,可以

对密闭大棚参数进行调控，但是数据传送需要铺设导线，导致成本高，操控难度大。

2005年，杜辉等设计了一种分布式温室测控设备^[9]，运用了蓝牙短距离通信技术。结合了蓝牙、总线技术对多个温室进行监测和控制，增强了测控的稳定性，有效减少了数据通信过程中的干扰。但是当时蓝牙短距离通信技术尚不稳定，尤其是在其他设备的影响下导致程序运行出现紊乱，使得系统在实际温室中的应用受到限制。

2007年，唐娟等设计了一种温室环境监测设备^[10]，运用了当时比较流行的AVR单片机作为主控制器。该设备既可以对单个温室进行监测，还可以将多个温室进行组网监测，实现了对多个温室的智能化、规模化和自动化测控。

2008年，周茂雷、郭康权设计了一种温室环境测控设备^[11]，运用ARM7微处理器作为核心控制芯片。该设备利用模数转换器对温室中各个模拟参数、开关参数进行实时的转换和采集，将得到的数据分析、计算，输出信号给调控机构来改变大棚状态。

2013年，王斌基于GPRS技术开发的日光温室综合环境集散控制系统等。这使得我国的温室控制技术紧紧地跟随着世界的发展潮流^[12]。

2017年，王亮凡等研究人员结合物联网技术，研发了具有实时监测环境参数功能的温室数字化管理系统，且能够自动调节环境参数，使其稳定在设定值范围内^[13]。

综合以上研究成果所述，促使农业朝着信息化、智能化的方向发展，但有些系统在易用性、稳定性、可靠性等方面仍需改进。随着物联网技术的不断进步，物联网结合农业生产的设备相继出现，为解决农业种植过程中理论参数匮乏、经验不足、历史数据遗失等问题，满足系统稳定、可靠的要求，提出了运用物联网和嵌入式系统等技术设计温室环境参数的监测系统。

目前，随着国内信息、电气技术的不断进步，并且不断引进国外先进的温室测控设备，这些设备的引用，对于我国的农业科学技术发展起到积极的促进作用。我国计算机技术在温室中的应用经历了直接运用、简单理解到设计开发的阶段推进，而且有些设备进行了创新性实用性的改进。但是，应用过程中还存在一些缺陷。技术层面，运用单片机作为主控制器多为单环境因子监测，还没有出现真正意义上的多环境因子参数综合采集分析设备，与国外对照仍存在着缺陷，仍需要进一步探究。

九十年代开始，嵌入式软硬件系统、农业科学、传感器、电气控制以及无线电路应用广泛，西方发达国家已经将智能化大棚的评估与建设提高至重要的位置，致力于充分利用现有的农村耕地资源，提高农作物的质量和产量，降低种植成本，节省人力资源。

1.3 本文的研究内容

论文中运用STM32作为核心控制芯片，配合NB-IoT窄带无线模块，系统可以实现的功能有对多个温室内多个环境因子的参数值进行监测，然后将采集的参数值利用窄带无线模块远程传送给移动终端，进行进一步的显示和处理。论文中，首先对系统中运用的几种关键技术进行了介绍，然后确定了系统的总体设计方案，接下来对系统中的硬件部分做了详细的阐述，分为采集节点部分和通信控制器部分。再根据系统功能要求，对每个传感器、采集节点控制器、通信控制器、系统通信协议和云平台部分的软件设计进行了描述。为提高设备的效率，将 μ COS II移植到通信控制器处理器中，为方便种植户观测参数值，系统运用移动终端进行参数值的显示、比较，并具有报警提示功能。

1.4 论文的组织结构

论文内容包含六个章节，主要内容如下：

第1章：绪论，首先说明本篇论文所涉及课题的研究背景和意义，然后分析了国内外温室环境测控设备的历史与近况，最后介绍了本课题的研究内容和本论文的组织架构。

第2章：系统的实施方案和应用的技术，首先确定了系统的总体方案框图，然后针对项目中应用的几种关键技术进行了分析与比较，包括嵌入式软硬件系统、ZigBee近距离无线通讯、NB-IoT窄带无线通讯。

第3章：系统硬件电路设计，详细描述了系统中用到的各种采集传感器、采集节点处理器芯片、ZigBee无线通信模块、通信控制器处理器芯片、NB-IoT模块的原理与选型，接下来进行了各个部分的硬件电路设计。

第4章：系统软件设计，详细描述了各部分的程序调试方法，涉及采集部分处理器程序、各传感器程序、通信控制器处理器软件设计、云平台报警设计、各部分之间的通信协议制定和农作物生长模型建立。

第5章：系统测试结果与分析，首先针对无线通信模块组网配置方法进行了介绍，然后对电路板进行通电检测，并对移动终端接收到的数据进行分析。

第6章：总结与展望，对系统的功能和应用场景进行了描述，然后列举了系统后期改进时可扩展的几种功能。

第 2 章 系统的实施方案和应用的技术

目前存在的温室环境监测设备仅仅实现农业温室种植环境参数的监测和传输，论文在传统设备的基础上进行了创新性改进，运用了无线通信技术和云平台管理监控技术，依据实际农业生产过程中的需要，实现温室环境因子的数据保存、实际生长模型建立、生长模型的导入，确定了设备所要实现的功能，并对设备中运用的ZigBee无线通讯、NB-IoT窄带通讯、嵌入式技术进行了描述。

2.1 系统的功能分析与设计

本论文设计的温室监控设备可以做到对园区内多个温室大棚内的温湿度、光强、CO₂含量的全天候采集，园区内为局域网。设计以STM32微控制芯片为平台，对温湿度、光照和CO₂浓度传感器输出的信号进行采集。利用ZigBee无线通讯将参数值传输到通信主控，再利用NB-IoT通讯传输给远端的电脑和手机终端，实现远程监控。当某个温室内采集到的某个参数超过设置的阈值时，设备可以发出报警并将报警内容通过NB-IoT网络传输给种植户的移动终端^[14]。设备总体框图如图2.1所示，单个温室内设备框图如图2.2所示。

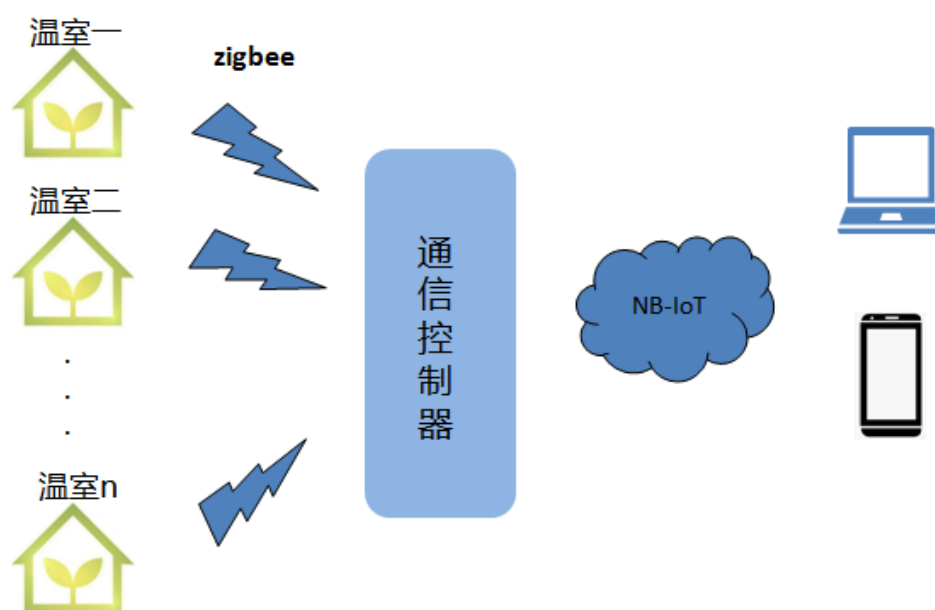


图 2.1 温室监测系统结构图

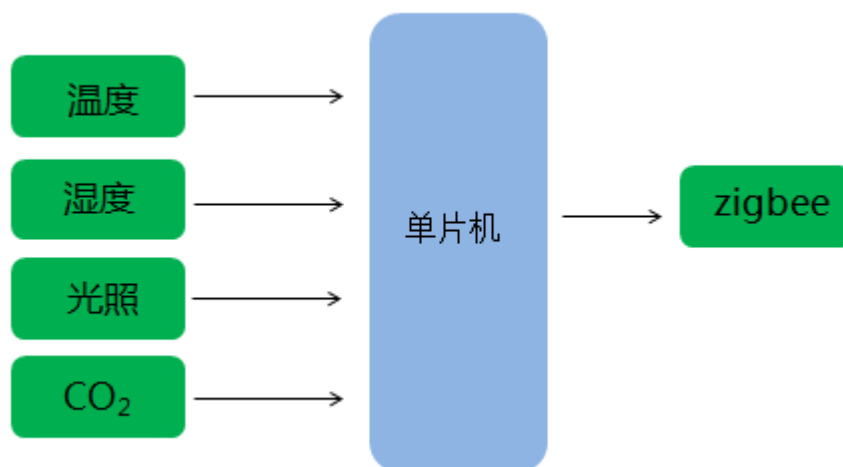


图 2.2 温室内系统结构图

2.2 ZigBee 技术

2.2.1 ZigBee 技术概述

ZigBee，又名紫蜂，一种无线通信协议，ZigBee内部对短距离、低速率通信传送制定了若干准则^[15]。遵守这一协议无线网通常工作的频段为868MHz（欧洲）、915MHz（北美）、2.4GHz（全球），传送速率一般小于250bps。遵守ZigBee协议的无线网络在能量很小时也可以可靠的在各个节点进行数据的传输。综上所述，紫蜂技术的特点包括高数据传输率、高可靠性、低成本、低耗能、高安全性。广泛应用于内部组网、无线传感系统、嵌入式电路、短距离无线通信。

2.2.2 ZigBee 网络协议

ZigBee网络协议运用分层设计的方法，在协议的各个层实现不同的功能，只有靠近的层才可以进行通信^[16]。在分层设计的网络架构的基础上，参照了通常无线通信网络的特征进行设计完成的。底部两物理层、介质访问控制层遵守IEEE802.15.4协议，最上面的网络层和应用层是ZigBee联盟设计规定的。应用层分三部分，即应用程序支持子层、协议设备对象、应用程序对象。如图2.3所示。

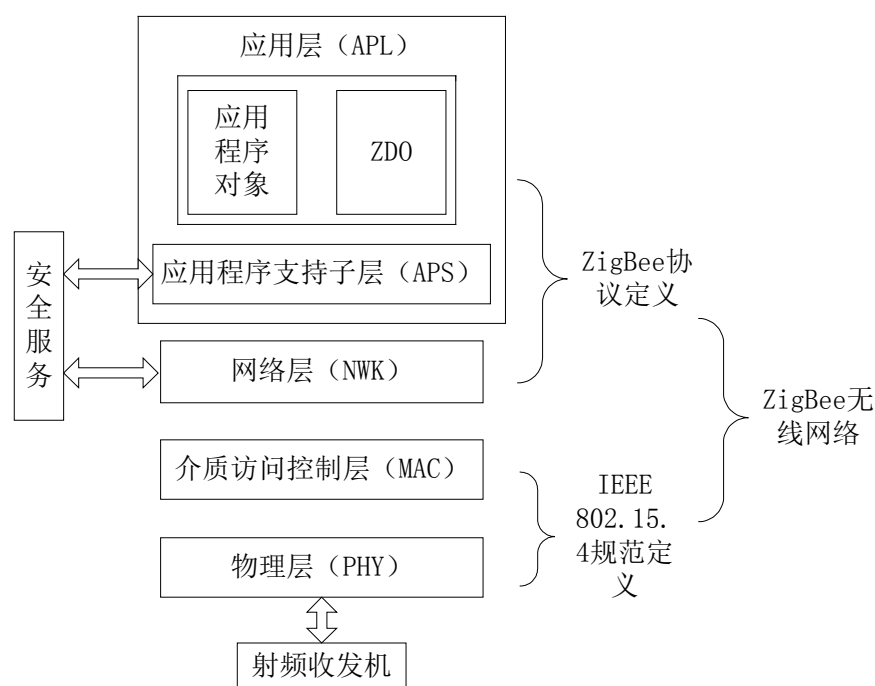


图 2.3 ZigBee 无线网络分层图

(1) 物理层。PHY层的作用是完成将要发送数据进行调制，然后发送，数据的接收也是由这一层完成的，通信之前需要设置选择信道频率。设计传感器采集节点时，涉及PHY层方面的因素包括成本、耗能、体积等。该层包含16个无线通道以及与MAC层的接口。

(2) 介质访问控制层。MAC层主要功能是帧节的检测、信道的复用、纠错，保证局域网中所有节点的通信畅通和准确。还可以建立和解除连接。

(3) 网络层。NWK层的作用是：形成网络、选择路径，发现、选择路由，传送、接受，分配地址，还具有与APL层通信的入口。

(4) 应用层。APL层是最高协议层，包括APS、ZDO和客户应用程序^[17]。该层还可以管理应用对象。单个模块中的应用对象数最多为240个，技术人员设计出应用对象来为各种应用程序定制模块。还具有同步功能，使得正在工作的模块与当地时钟处于同步状态。定位服务是首先其他节点的物理位置是确定已知的，根据接收数据的信号强度计算被测节点的坐标位置，在室内定位应用中得到广泛的应用。

2.2.3 ZigBee 网络拓扑结构

每个节点有唯一地址和接入点，基于这一点实现了各个节点之间的相互通信，组网成功后，网络中的节点会自动得到一个16位的地址。节点的绝对地址是指64位的MAC地址。三种网络结构如图2.4所示。

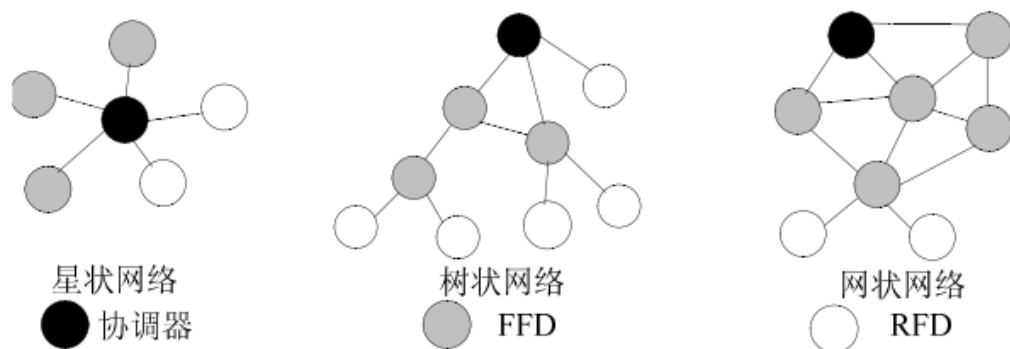


图 2.4 ZigBee 网络拓扑结构图

(1) 星状网络拓扑结构

星状网络拓扑比较简单，仅仅实现协调器和终端或路由之间的相互通讯，且路径唯一。路由及终端两者没有办法数据通讯，结构过于简单，不要求用ZigBee协议栈。

(2) 树状网络拓扑结构

树状拓扑终端及路由之间可以进行通讯，但要以子节点和父节点为基础。该结构遵循就近传输的原则，即相邻较远的节点之间通信需要先传输给离自己最近的节点，再传输给想要传输的节点，通信时路由路径唯一。

(3) 网状网络拓扑结构

网状结构是多跳系统，网络中的每个节点既可以当路由又可以作为终端节点，均起到中继的作用^[18]。各个节点进行数据通信时，经过每个路由节点均作为中继，避免了树状网络路由路径唯一的问题。网络中的路由具有自我建立和优化功能，实现了自我修复通信过程中遇到的障碍，使得通信更稳定。

2.3 NB-IoT 技术

2.3.1 NB-IoT 技术概述

NB-IoT，窄带无线通讯，利用现有的蜂窝网络，运营商针对物联网的需要而推出的^[19]，所占带宽非常窄，仅仅180千赫兹，并且属于License频段。可以同已存在网络同时存在，兼容GSM、UMTS、LTE，也就是2G、3G、4G，减少了成本，网络利用率顺利得到提升。

蜂窝网应用广泛、范围大，因此具有很好的接入能力。这些优点是WiFi、蓝牙、ZigBee无法比拟的，越来越多的设计者倾向于窄带无线通讯附加在蜂窝网络中的应用，正在迅速被应用在万物互联的场景中。

2.3.2 NB-IoT 技术的特点

NB-IoT含有4个主要特点:

(1) 广覆盖。较传统网络, 比如GSM、LTE, 覆盖率提高20dB, GSM基站最大覆盖方圆35km的范围, GSM信号无法实现在非常深的地下传输, NB-IoT网络可以实现^[20]。这些优点得益于: 带宽窄、增加功率谱密度、多次传输, 从而实现时间分集增益。

(2) 大连接。NB-IoT单个基站可接入的数量是传统无线网络的近百倍, 单个社区接入数达5万, 这是物联网建立的必要条件之一。这些优点得益于: 具有时延不敏感的特征, 和移动通话用同了一种技术, 存入若干接入模块通讯时的上下文, 反复更换休眠态、激活态; 窄带通讯具有较小的上行调度颗粒, 提升了资源利用率^[21]; 使得空口信令交互变少, 进而使得频谱密度变大^[22]。

(3) 低功耗。接入设备仅仅1%的时间处于工作状态, 其余时间均处于休眠状态, 而且具有很多省电模式, 最长甚至可以待机十年。另外还有功耗更少的两种模式: PSM和eDRX。

(4) 低成本。软硬件均可以根据要求进行简化定制, 因此NB-IoT模块价格非常低, 单个不到五美元。

低速率、低移动性是窄带通讯的两大特征。低速率, 上行只有56千位每秒, 最大下行速率21.25千位每秒; 低移动性, 接入设备移动时速不能超过30公里每小时, 较4G的250公里每小时小很多, 高铁中利用了特殊的网络, 移动时速低于450公里每小时即可。

2.3.3 ZigBee、LoRa、NB-IoT 比较

物联网中的节点通信方式分有线和无线两种, 有线方式具有布线复杂、施工难度大、成本高的特点。无线方式主要包括ZigBee、LoRa、NB-IoT三种, LoRa、NB-IoT为LPWAN技术, 主要特征是覆盖范围大、连接设备数量多、传输速率低、成本低、功耗低^[23]。

NB-IoT应用场景中可以没有网关, 可以直接将要传输的数据发送到云平台, 无需考虑安装网关带来的选址难、有无干扰等问题, 使得施工更简单。3种无线通讯网络对比如表2.1所示。

表2.1 NB-IoT、LoRa、ZigBee无线网络技术对比

	NB-IoT	LoRa	ZigBee
网络形式	利用蜂窝网络	通过 LoRa 网关	通过 Zigbee 网关
网络布局方法	只需要节点	需要节点和网关	需要节点和网关
传送范围	大 (十公里以上)	大 (市区两公里, 空旷地二十公里)	小 (一般小于一百米)

接入设备数量	二十万	理想值六万，通常小于五千	理想值六万，通常小于五百
电池供电时长	理想十年/AA 电池	理想十年/AA 电池	理想两年/AA 电池
价格	小于十美元，有望达到一美元	五美元	小于两美元
工作频段	License，运营商	unlicense，Sub-GHz (433 MHz、868 MHz、915MHz)	Unlicense，2.4GHz
传送速率	理想值 160 千位每秒 - 250 千位每秒，通常不大于 100 千位每秒，受串口波特率影响	50 千位每秒以内	理想值 250 千位每秒，通常不大于 100 千位每秒，受串口波特率影响
延迟时间	6 秒 -10 秒	TBD	小于 1 秒
应用场景	多用于室外，大范围组网。	多用于室外，结合 NB-IoT 作为补充，可以延伸到没有蜂窝的地方。	多用于室内，室外也可以，小面积使用可以组成个人局域网。

2.4 嵌入式系统

2.4.1 嵌入式系统概述

一套完整的嵌入式系统包括硬件部分和软件部分，其中硬件部分包括主处理器和外围应用电路，软件部分包括操作系统和符合功能要求的应用程序，集合了软硬件开发技术，是一门比较综合的技术。

2.4.2 嵌入式系统的几个特征

(1) 系统内核的容量非常小。嵌入式设备一般是消费类电子设备，因此体积比较小，而且资源较少。这就要求系统的内核占用的存储空间比较小，相比常见的操作系统小很多。例如 OSE 系统，内核的大小仅仅 5KB，是由 ENEA 开发的，相比常见的 Windows 操作系统的内核容量小得多^[24]。

(2) 显著的专用特性。嵌入式设备软硬件紧密配合，一般是根据用户要求进行定制开发的。即便是同一个公司的类似产品，在做系统移植时，也要进行不断地编辑和调试。如果产品功能有所改变，会导致软硬件同时进行非常复杂的修改，因此嵌入式设备一般是个性化定制的，功能升级比软件升级复杂的多。

(3) 系统要求简化。嵌入式设备软件部分，不区分系统和应用软件，只要实现功能即可，软件部分不要求过于复杂，有利于降低成本和系统的安全。

(4) 嵌入式设备软件部分的设计需要遵循一定的标准，便于团队其他设计人员的修改和升级，其实用户应用程序可以直接运行在主处理器内而不需要操作系统，但是会使程序执行效率大大降低，因此为了合理调度任务，有必要移植支持多任务运行的操作系统^[25]。结合资源、函数、库函数等接口，选择合适的操作系统，才可以保障软件程序的实时性、稳定性，进而提高产品质量。

(5) 设计基于特定的软件。因为嵌入式系统不带有自我设计的功能，开发结束之后，产品使用人员无法进行软件功能的调试和更改。所有需要基于特定的开发软件才可以进行软件功能的更改和调试，而且这些软件通常需要运行在计算机上面。设计的过程中，为提高效率，通常设立主机、目标机，主机作用是调试软件，目标机的作用是产品应用，设计过程中，二者轮流开展。

2.4.3 嵌入式系统的分类

鉴于嵌入式系统包括硬件部分、软件部分，因此分类介绍分两部分进行：

(1) 硬件部分。硬件部分包括主处理器和外围电路部分，其中主处理器是关键。目前，全球微处理器种类总量有一千余种。常用体系架构三十余个，例如 51 架构应用最广泛。制造 51 微处理器的公司达二十余家，合计三百五十余个系列芯片，其中飞利浦厂家产品达一百余个。当下微处理器的可寻地址范围是 64 至 256MB，运行速度范围是 0.1 至 2000MIPS。

目前微处理器正朝着微型化、节能化、高效性方向演变，厂家特色愈加明显，有些公司业务仅仅负责开发 IP 核，其中应用比较广泛的比较有 ARM、MIPS，他们开发高效的、高性价比的内核，然后半导体公司负责设计和制造不同应用场景的微处理器。硬件部分分类如图 2.5 所示。

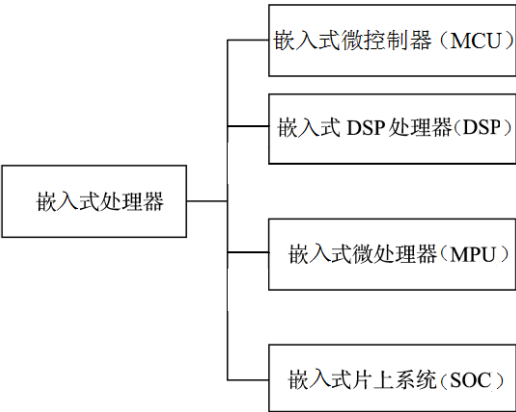


图 2.5 嵌入式硬件系统分类

(2) 软件部分。软件部分包括系统软件 and 应用程序软件，其中硬件电路和用户应用程序之间的纽带是操作系统，有两个作用，即硬件得以运行以及高效率的调用处理器的资源。当下，嵌入式系统分类如图 2.6 所示。

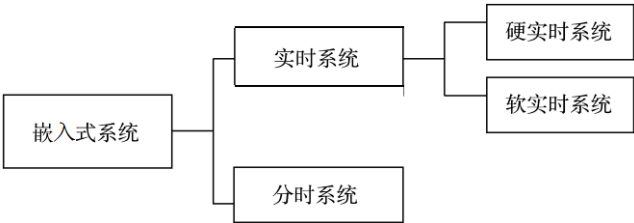


图 2.6 嵌入式软件系统分类

实时系统的特点是实时性，可以实时调控系统的运行。它的作用首先是分配

可以应用的硬件资源进行任务的执行, 然后是提高硬件的运行效率^[26]。它的主要特征是通过切换任务来保证在要求的时间内完成相应的功能。它和分时系统不同, 分时系统最大的特点是不要求效率, 即时间有所延迟, 对功能或者允许效果不产生影响。但是实时系统最重要的指标是对当前任务的及时运行, 即便事件不按要求的时间节点产生或者系统在超负荷状态下运行, 系统也需要在较短的时间内及时做出回应, 否则将带来严重后果。实时系统还有可确定性的特征, 它可以对运行结果的正确与否作出判断。

硬实时、软实时系统均属于实时操作系统。二者的不同之处是响应外部事件的时间长短。硬实时系统的响应时间短, 如果超过限制时间有可能造成严重后果。比如导弹系统出现故障会导致灾难性后果。

软实时系统是一种对时限要求不严格的一种操作系统, 系统超负荷运行时, 导致的时间发生延迟对系统运行不会造成严重后果。例如显示模块, 偶尔延迟不会导致严重的后果。

之所以导致以上的差异, 原因在于二者的调度运算方法不一样。软实时系统运用的是优先级调度的运算方法, 可以达到软实时系统要求的条件, 还具有快速回应和较大的运行数据量, 就硬实时系统而言, 使用了简易、迅速响应的实时进行调度的运算方法。

2.4.4 μ COS 嵌入式操作系统

μ COS 是一种实时系统, 代码开源且免费, μ COS II 具有以下 7 个特征:

(1) 程序代码开源, 这是它最重要的特征, 也是它之所以得到广泛应用最重要的原因, 可以将其移植到多种嵌入式硬件电路中。

(2) 容易固化在硬件中, 可以利用编程软件或者程序烧写软件将 μ COS II 系统固化到设计者开发的产品内, 这也是用户最终产品的组成之一。

(3) 支持多任务运行。具有保护资源共享的特点。它经过增加信号量的方法来避免资源共享带来的冲突问题。

(4) 支持内核裁剪, 通过选择和删减 μ COS II 中的系统服务程序, 达到提高硬件资源利用率的目的

(5) 移植简单, μ COS II 的内部程序使用 ANSI C 语言编写, 移植难度小。与硬件相关的很小一部分代码是基于汇编语言, 使得移植更简单。

(6) 内核形式为占先式, 它的特点是首先响应优先级高的任务或事件。绝大部分商业内核形式均为占先式。

(7) 执行时间的可确定性, 每个函数执行和响应的运行时间是可确定的^[27]。

鉴于 μ COS II 的实时性, 架构简单, 我们可以编辑和运用的是 API 函数接口, 但是还需要设计者完成其他的一些操作。移植完成只是 μ COS II 应用的第一步,

可以根据系统要求对内核进行裁剪，还可以进行扩展，有最基础的硬件驱动、文件管理系统、人机交互界面等，直到创建成为符合功能要求的 RTOS。

第3章 系统硬件电路设计

硬件电路设计是电子系统设计过程的组成部分。本章首先确定了硬件电路设计总体框图,把系统硬件设计分为两部分,即采集节点部分、通信控制器部分^[28]。并在总体框图中说明了由哪些电路模块构成。然后分模块介绍了各个组成电路的关键元器件参数、设计方法和原理图。最后进行了两部分电路的 PCB 电路板图的设计说明。结合大棚温室的实际情况,硬件设计应遵循体积小、低成本、耗电少、传输距离远、可扩展性、防潮湿等原则进行设计与制作。

3.1 系统总体硬件设计

系统硬件设计包括采集节点部分设计和通信控制器部分设计两部分,采集节点部分包括单片机的选型与外围电路设计、基于 CC2530 的 ZigBee 模块和传感器选型与设计三部分;通信控制部分包括微处理器选型与设计、NB-IoT 模块选型两部分,如图 3.1 所示。

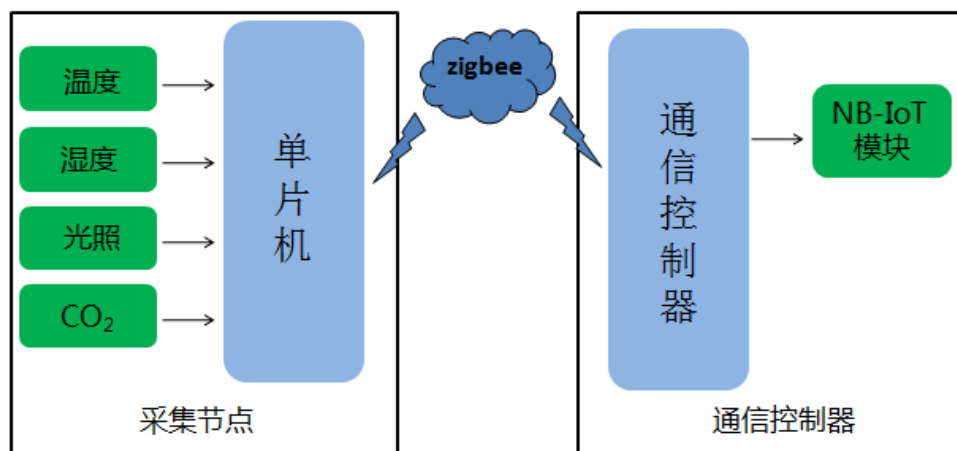


图 3.1 系统硬件总体框图

3.2 采集节点部分硬件设计

每个温室需要放置一个采集节点设备,负责监测所在温室的环境因子参数,经过对数据的滤波处理,将数据通过 ZigBee 无线模块传送给通信控制器。该部分内容包括确定了单片机的型号,进行了单片机外围电路的设计,选定了基于 CC2530 的无线模块,对环境参数检测传感器做了比较与选型,并进行了与单片机的接口设计。

3.2.1 单片机选型与设计

采集节点部分外围电路包括四种传感器和无线模块, 占用资源较少, 因此选用了 51 系列单片机作为主控制器。STC 厂商生产的单周期增强型单片机编程方法与传统 51 单片机完全一样, 管脚顺序也一致。程序存储器是经过 FLASH 工艺制造的, 而且容量非常大^[29]。例如型号为 STC12C5A60S2 的单片机, 芯片内部集成了 64K 的程序存储器, 支持进行电擦除和修改。程序烧写端口为串口, 相较传统 51 单片机的 SPI 端口简便许多。因此, STC 单片机使得开发更便捷。而且在程序烧写过程中, 自动加密, 杜绝了模仿者的抄袭行为, 保护了设计者的合法权益。值得一提的是, 价格与之前常用的 51 系列单片机差不多, 容易采购, 真正实现了非常高的性价比。

STC12C5A60S2 是单周期单片机, 编程方法与传统 51 单片机相同, 但是程序执行速度提高了 8 至 12 倍, 特点是速度快、耗能少、抗干扰能力强。无需外接复位电路, 内部已经具有复位功能, 针对电气控制和干扰比较多的情景的应用, 芯片内部集成了 PWM 产生与输出、多路模数转换。该单片机还具有以下特征:

- (1) 增强型 51 内核, 单周期, 程序指令集与常用 8051 相同^[30];
- (2) 电源电压范围: 5.5V-3.3V (5V 单片机), 3.6V-2.2V (3V 单片机);
- (3) 自带 EEPROM, 注意 STC12C5A62S2 除外;
- (4) 内部含有四个十六位定时器, 其中 T0、T1 和普通 51 单片机用法一样, 内部无定时器 T2, 其中内部用于产生串口通信波特率的部分是它内置有专门的独立的波特率发生器^[31], 另外, 两个 PCA 用于组成其余两个十六位定时器;
- (5) 芯片内自带 1280 字节数据存储器^[32];
- (6) 根据封装差异, 通用端口引脚数量为 36 个、40 个、44 个, 复位之后端口为准双向口/弱上拉模式, 相当于传统 51 单片机端口, 另外还有三种形式, 即推挽/强上拉、仅作为输入/高阻态、漏极开路。芯片电流小于 55 毫安左右, 端口电流 20 毫安左右;
- (7) 支持 ISP、IAP, 不需要编程器和仿真器, 仅仅连接串口至电脑, 就可以进行程序烧写, 根据程序大小不同, 烧写时间为数秒;
- (8) 执行频率范围: 0-35MHz, 是常用 51 单片机的 12 倍;
- (9) 外部中断端口数量为七个, 除了具有下降沿、低电平触发中断, 还添加了 PCA 用于实现上升沿触发中断。外部中断的一个功能是低功耗模式下的唤醒, INT0 端口 P3.2, INT1 端口 P3.3, T0 端口 P3.4, T1 端口 P3.5, RxD 端口 P3.0, CCP0 端口 P1.3 (利用程序可改为 P4.2), CCP1 端口 P1.4 (利用程序可改为 P4.3);
- (10) 芯片内自带 MAX810 专用复位电路, 当晶振频率低于 12MHz 时, 复位端口和地之间接 1K 欧姆的电阻即可;

(11) 支持外接失电检测电路:端口 P4.6 内部集成了低压门槛比较器电路,其中 5V 工作电压的单片机为 1.32V,误差范围是 $\pm 5\%$, 3.3V 工作电压的单片机为 1.30V,误差范围是 $\pm 3\%$;

(12) 程序执行时钟来源:有两种时钟源可以选择,即外接晶振、使用芯片内集成 RC 振荡器,在常温条件下,内部振荡器频率为 11-15.5MHz (供电电压 5V 单片机)^[33], 8-12MHz (3.3V 供电电压的单片机),对程序执行时序不严格的场景下,可以使用内部振荡器,原因是存在一定的温度漂移和较大误差;

(13) 内部自带看门狗;

(14) 时钟输出端口的数量为两个,即定时器 T0 溢出输出端口 P3.4 和定时器 T1 溢出输出端口 P3.5;

(15) 支持八路模数转换,精度为十位,最大转换频率 250K 每秒,即 1 秒可以转换 250 千次;

(16) 可输出两路 PWM,内置两路 PCA,还具有其他功能,可以作为两路数模转换、两个定时器、两个外部中断 (和其他外部中断端口功能一样);

(17) 程序存储器空间:8K-62K 字节;

(18) 基于该单片机的高速特性,具有使用定时器、PCA 通过软件设置为多个串口的功能;

(19) 型号最后“S2”说明具有两个串口,其中串口二 RxD2 端口为 P1.2 (利用程序可改为 P4.2), TxD2 端口为 P1.3 (利用程序可改为 P4.3);

(20) 工业级芯片使用温度介于-40 到 85 摄氏度之间,商业级使用温度介于 0 到 75 摄氏度之间^[34];

(21) 具有双列直插 DIP40、LQFP、LQFP 三种封装,当端口不能满足要求时,可以利用 74HC595 进行端口扩展,级联后每片 74HC595 可扩展 8 个端口,实际应用中可以将模数转换端口用于按键识别,达到节省端口的目的,还可以运用两片单片机进行通信扩展端口、增加串口^[35]。

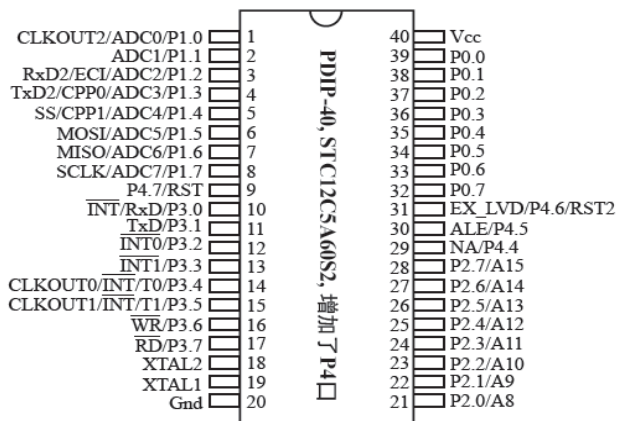


图 3.2 STC12C5A60S2 管脚图

每个引脚的使用说明如下:

VCC: 接供电电源正极;

GND: 接供电电源地;

P0 口: P0.0~P0.7 八个端口, 为漏极开路输入输出端口。给端口写入高电平时, 设置成为高阻态输入端口。当外接程序、数据存储器, 被用作数据或地址的低 8 位。

P1 口: P1.0~P1.7 八个端口, 为内置上拉电阻的输入输出端口。给端口写入高电平时, 设置成为输入端口。内置上拉电阻的电路结构使端口输出低电平的情况下会输出电流;

P2 口: P2.0~P2.7 八个端口, 为内置上拉电阻的输入输出端口, 给端口写入高电平时, 设置成为输入端口。内置上拉电阻的电路结构使端口输出低电平的情况下会输出电流。当外接程序、数据存储器时, 被用作数据或地址的高 8 位。

P3 口: P3.0~P3.7 八个端口, 为内置上拉电阻的输入输出端口, 给端口写入高电平时, 设置成为输入端口。内置上拉电阻的电路结构使端口输出低电平的情况下会输出电流。P3 口各端口具有第二功能如下:

P3.0: RXD(串行输入口)

P3.1: TXD(串行输出口)

P3.2: INT0(外部中断 0)

P3.3: INT1(外部中断 1)

P3.4: T0(定时器 0 或计数器 0 外部输入)

P3.5: T1(定时器 1 或计数器 1 外部输入)

P3.6: WR (外 RAM 写)

P3.7: RD (外 RAM 读)

RST: 复位引脚。当设计复位电路时, 需满足的条件是, RST 端口保持高电平的时长大于 2 个机器周期;

ALE / PROG 引脚: 该引脚不常用, 在外接存储器的情况下, 该引脚的作用是锁存地址的低位。通常情况下, 该引脚持续输出六分之一晶振频率的正脉冲信号, 接入外部电路当做脉冲输入和定时。将地址为 8EH 特殊功能寄存器置为低电平, 可以关闭 ALE 引脚输出。

PSEN 引脚: 输出外部 ROM 选通信号, 使用时取址期间, 单位时间内两次有效。另外, 访问内部 RAM 时, 无 PSEN 信号输出;

EA/VPP 引脚: 如果需要与外部程序存储器数据通信, 需要将 EA 置 0; 需要注意的是, 当加密模式为 1 的时候, EA 引脚将内部锁定成复位功能; 通常应用下, 不需要外接程序存储器, 因此使用内部程序存储器保持 EA 为 1 即可;

XTAL1: 接晶振外部时钟电路;

XTAL2：接晶振外部时钟电路；
单片机外围电路图如图 3.3 所示：

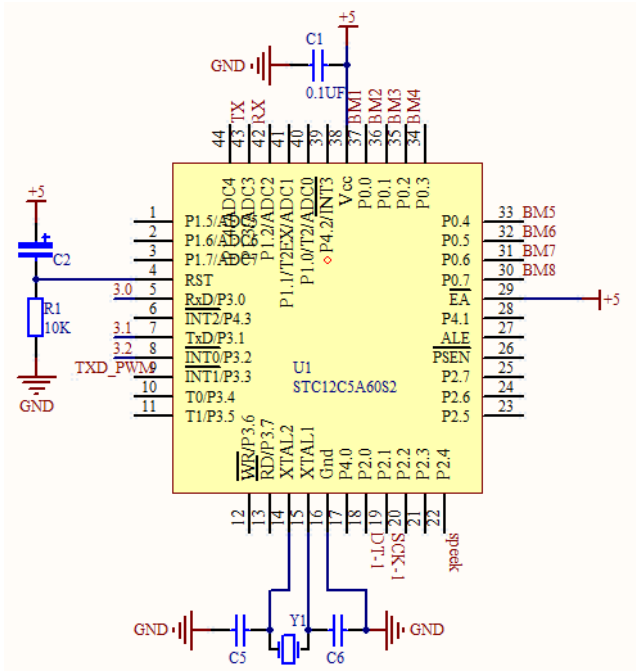


图 3.3 单片机外围电路

3.2.2 ZigBee 模块选型与设计

CC2530 是由德州仪器（TI）公司设计生产的，可用在 2.4GHz IEEE 802.15.4, ZigBee 和 RF4CE 应用。内部包含有高质量的 RF 收发器和工业级 51 内核，因此也称作无线单片机，RAM 容量 8KB，可以在多种模式下运行，应用在低功耗的场景下具有比较大的优势，运用了 ZigBee 协议栈（Z-Stack™），厂商提供了许多解决设计方案供用户选择。

CC2530 应用场景非常广泛，例如在 2.4GHz IEEE 802.15.4 电路、RF4CE 电路、ZigBee 电路，自动化控制系统，照明控制设备，工业生产控制系统，传感器数据的无线传输，常用电子产品和医疗器械等。

（1）CC2530 芯片的特点：

CC2530 为 2.4GHz IEEE 802.15.4 和 Zigbee 结合的 SOC 设计方案，使得基于 ZigBee 的 2.4 GHz ISM 波段运用成为可能，满足了物联网领域的低成本、耗能少的要求。内置了高质量的频率为 2.4 G 赫兹 DSSS(直接序列扩频)RF 收发器电路，以及工业级 51 内核^[36]。

CC2530 使用了 0.18 微米 CMOS 生产工艺，正常工作电流为 20mA；处于接收状态时，电流低于 30mA，处于发射状态时，电流低于 40mA^[37]。而且由休眠变为主动模式耗时很短，多用在电池供电的场景中。

CC2530 的主要特点如下：

内置了 8051 内核，高质量、耗能少，而且自代源码预取机制。

内部 Flash 容量有 32-256KB 多种型号。

内部数据存储器容量为 8KB。

RF 收发器支持 2.4GHz IEEE 802.15.4。

支持 RSSI（接收信号强弱）和 LQI（链路性能）读取^[38]。

可编程输出功率低于 4.5dBm。

内部集成模数转换通道数为八个，而且可以修改分辨率，最高 12 位。

内部 DMA 通道数量为五个。

IR 产生外围电路。

内置两路高性能串口 UART。

内含一个 MAC 定时器，执行 IEEE 802.15.4 标准，三个定时器位数分别为：一个十六位、两个八位。

支持的电源电压为 2.0 至 3.6V。

内置电源监控电路和测温电路。

休眠时的电流为 0.4 微安，可以通过外部中断、RTC 来恢复。

待机时电流小于 1 微安，可以通过外部中断来恢复。

具有调试功能，开发平台比较完善。

外围电路非常简单。

(2) CC2530 引脚图示及描述：

CC2530 的封装类型为 QFN40, 尺寸大小为 6*6 毫米，引脚数量为 40。引脚排列如图 3.4 所示。

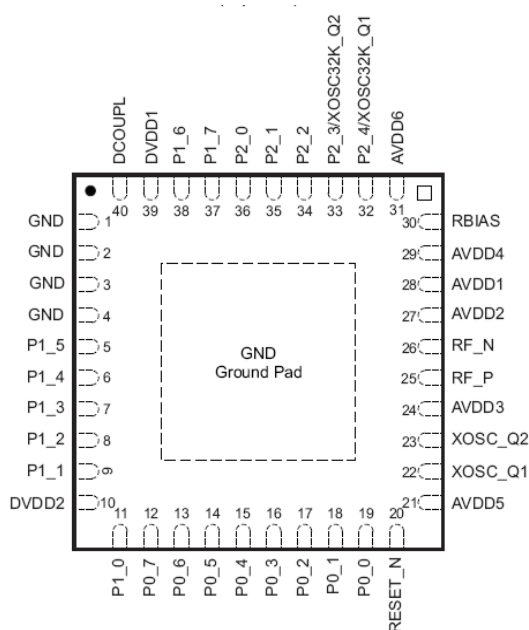


图 3.4 引脚排列图

引脚描述:

CC2530 引脚可以划归为三类,即端口、电源、控制引脚。其中端口数量为二十一个,包括两组八位端口 P0 和 P1,一组五位端口 P2。利用程序给一个 SFR 寄存器赋值,可以设置引脚功能为通用端口、模数转换、定时计数器、串口。CC2530 的引脚功能描述如表 3.1 所示。

表 3.1 CC2530 引脚功能描述

引脚名称	引脚	引脚类型	描述
AVDD1	28	电源 (模拟)	2V-3.6V 模拟电源连接
AVDD2	27	电源 (模拟)	2V-3.6V 模拟电源连接
AVDD3	24	电源 (模拟)	2V-3.6V 模拟电源连接
AVDD4	29	电源 (模拟)	2V-3.6V 模拟电源连接
AVDD5	21	电源 (模拟)	2V-3.6V 模拟电源连接
AVDD6	31	电源 (模拟)	2V-3.6V 模拟电源连接
DCOUP1	40	电源 (数字)	1.8V 数字电源去耦。不使用外部电路供应。
DVDD1	39	电源 (数字)	2V-3.6V 数字电源连接
DVDD2	10	电源 (数字)	2V-3.6V 数字电源连接
GND	-	接地	接地
GND	1-4	未使用的引脚	连接到 GND
P0_0	19	数字 I/O	端口 0.0
P0_1	18	数字 I/O	端口 0.1
P0_2	17	数字 I/O	端口 0.2
P0_3	16	数字 I/O	端口 0.3
P0_4	15	数字 I/O	端口 0.4
P0_5	14	数字 I/O	端口 0.5
P0_6	13	数字 I/O	端口 0.6
P0_7	12	数字 I/O	端口 0.7
P1_0	11	数字 I/O	端口 1.0-20mA 驱动能力
P1_1	9	数字 I/O	端口 1.1-20mA 驱动能力
P1_2	8	数字 I/O	端口 1.2
P1_3	7	数字 I/O	端口 1.3
P1_4	6	数字 I/O	端口 1.4
P1_5	5	数字 I/O	端口 1.5
P1_6	38	数字 I/O	端口 1.6
P1_7	37	数字 I/O	端口 1.7

P2_0	36	数字 I/O	端口 2.0
P2_1	35	数字 I/O	端口 2.1
P2_2	34	数字 I/O	端口 2.2
P2_3	33	数字 I/O	端口 2.3 或 32.768 kHz 晶振
P2_4	32	数字 I/O	端口 2.4 或 32.768 kHz 晶振
RBIAS	30	模拟 I/O	用于参考电流的偏置电阻
RESET_N	20	数字输入	复位至低电平
RF_N	26	RF I/O	负 RF 输入到 LNA
RF_P	25	RF I/O	正 RF 输入到 LNA
XOSC_Q1	22	模拟 I/O	用于 RF 的时钟输入或 32MHz 晶振引脚
XOSC_Q2	23	模拟 I/O	32MHz 晶振引脚

CC2530 外围电路图如图 3.5 所示:

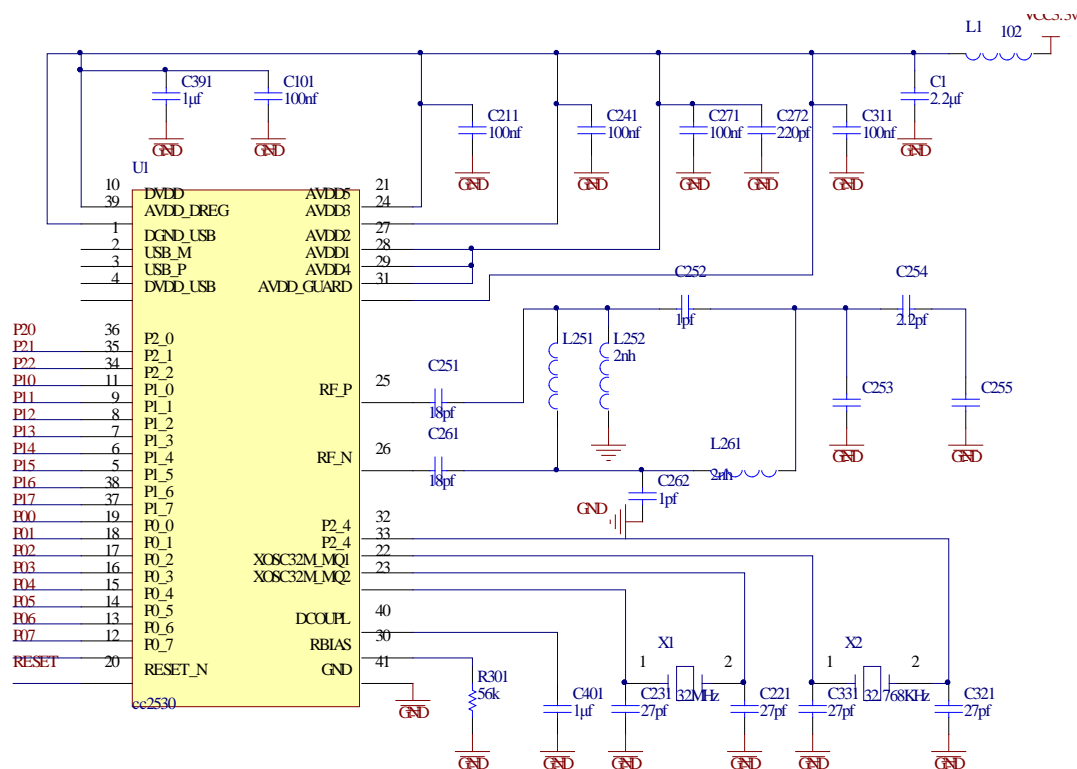


图 3.5 CC2530 外围电路图

CC2530 的电压输入范围较宽, 采集单元工作电压为 3.3V, 在 CC2530 供电电压范围内。该部分电源电路的设计, 采用 REG1117-3.3 芯片实现 3.3V 稳压供电, 如图 3.6 所示。

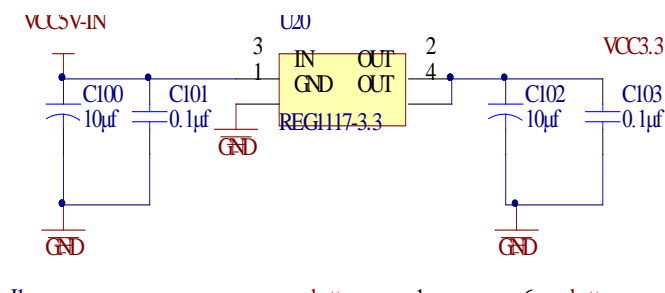


图 3.6 采集模块电源设计图

3.2.3 传感器选型与设计

电子系统通常利用传感器将非电量感知参数转换成可用的电信号实现感知信号的定量采集。传感器的出现和设计，使得一些只能用语言形容的事物更准确的形态呈献给大家。采集节点单元电路模块用到的传感器有 DHT11 温湿度传感器、BH1750FVI 光照传感器和 MG811 二氧化碳传感器，可以有效、快速、准确地检测环境参数中的温度、湿度、光照强度和二氧化碳气体浓度，下面我们进行一一阐述。

(1) 温湿度传感器

DHT11 温湿度传感器输出信号为数字信号，数据已进行滤波和校准，可以同时测量并输出温度和湿度参数，是一种综合型传感器。内部由两部分组成，即测温元件、电阻式感湿元件。湿度测量误差 $\pm 5\%$ ，测量范围 20%–90%RH；温度测量误差 $\pm 2^{\circ}\text{C}$ ，测量范围 0–50 $^{\circ}\text{C}$ 。是将温度和湿度测量集于一体及含有已校准数字输出的复合传感器，它由一个测量温元件和一个电阻模式的感湿元件组成，内置集成了八位单片机作为数据采集处理控制器，使得传感器数据采集更准确、稳定。该传感器生产的最后一个环节是在专用的实验室校准测量参数，并且将校准表参数保存在 OTP 里。传感器共有 4 个引脚，两个电源引脚，1 个空引脚，1 个单总线数据通信引脚，从而节省了单片机通讯端口。DHT11 实物、接口电路如图 3.7、3.8 所示。

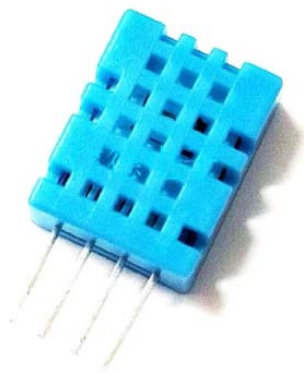


图 3.7 DHT11 实物图

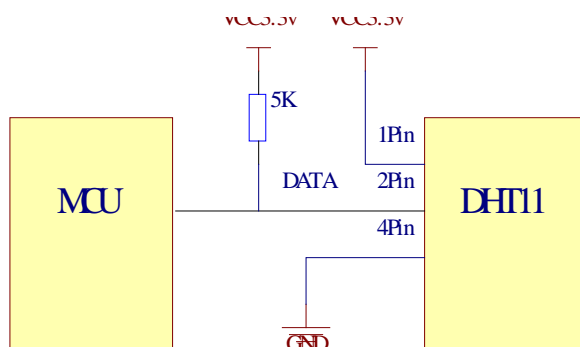


图 3.8 DHT11 接口电路图

(2) 光照强度传感器

常见的采集光照参数的元器件是光敏电阻，衡量光敏元器件性能好坏的指标是该元器件的光电特性是否呈线性，光敏电阻为非线性，因此适用于开关量光照参数检测的场景中，如声控灯只有光线暗的时候被点亮。光敏电阻在光照强度精确检测场合应用较少，而且光敏电阻外围电路输出信号为模拟信号，需增加模数转换电路将模拟信号转换为单片机适用的数字信号，增加了系统设计难度和成本。综上所述，本论文采用日本 RHOM 株式会社生产的 BH1750FVI 作为光照强度采集传感器，I²C 数据通信总线数字量输出，十六位精度（1~65535 lx 的高分辨率），不区分光源，检测范围大，由于内部集成了将光照强度转换为数字量的电路，有效降低了信号放大、模数转换电路引起的误差。BH1750FVI 结构框图如图 3.9 所示。

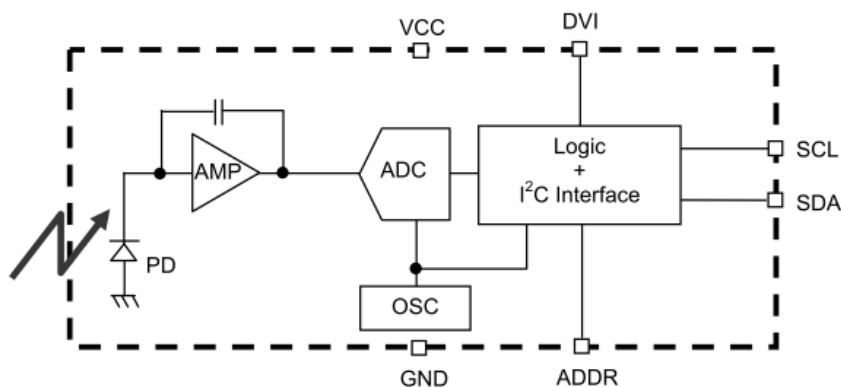


图 3.9 BH1750FVI 结构框图

BH1750FVI 工作原理是光敏二极管 PD 将照射光强度变化转换为电流的变化，经运算放大器 AMP 转换为电压的变化，再由模数转换电路（ADC）转换为十六位数字量的变化，最后进入数据处理单元，将数据校准后输出。晶振 OSC 为转换提供时钟信号，外部微处理器按照传感器时序要求通过 I²C 数据通信总线与之通讯，可以获取光照强度数据，系统简单、精确、稳定。BH1750FVI 实物、接口电路如图 3.10、3.11 所示。



图 3.10 BH1750FVI 实物图

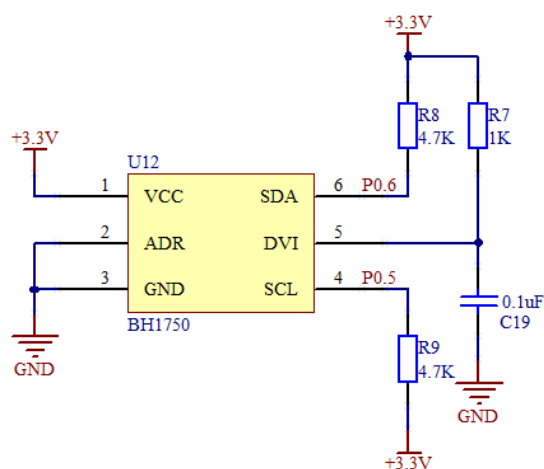


图 3.11 BH1750FVI 接口电路图

(3) 二氧化碳浓度传感器

二氧化碳浓度值是影响种植物生长过程非常重要的参数，为实现对该参数的不间断监测，进而调节种植物的生长周期，通过设计方案的对比，本系统选择 MG811 作为二氧化碳浓度检测传感器^[39]，具有灵敏度高、选择性较好、稳定、线性度高、不受外围环境干扰等特性。外接集成运算放大器 CA3140 实现阻抗变换，输出模拟信号给微处理器。MG811 实物、接口电路如图 3.12、3.13 所示。



图 3.12 MG811 实物图

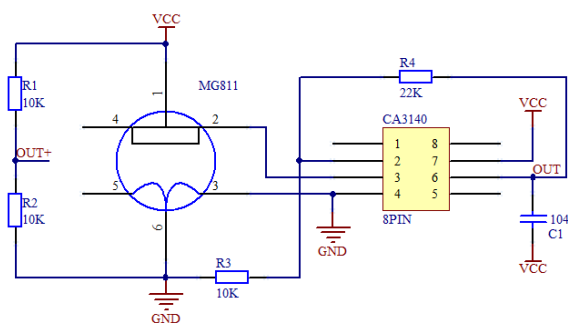


图 3.13 MG811 接口电路图

3.2.4 采集节点部分 PCB 电路板图设计

(1) 利用 Altium Designer 进行电子产品 PCB 板图设计的一般方法和步骤：

第一步，按要求新建 PCB 板图文件，并设置参数，根据外壳调整电路板外形、放置固定孔、倒圆角等。

第二步，制作自己的封装库。

第三步，加载封装库，修改元器件封装属性。

第四步，导入网络表。

第五步，电路板元件布局。

第六步，设置布线规则并布线。

第七步，设置覆铜规则并覆铜。

第八步，根据需要放置电路名称、标识、单位名称和 LOGO、版本号、设计日期等。

第九步，元件列表和图形输出与打印。

(2) 按照 Altium Designer 软件 PCB 板图设计的步骤进行采集节点部分 PCB 的绘制，需要注意的是，要根据外壳确定电路板外形尺寸和固定孔位置。

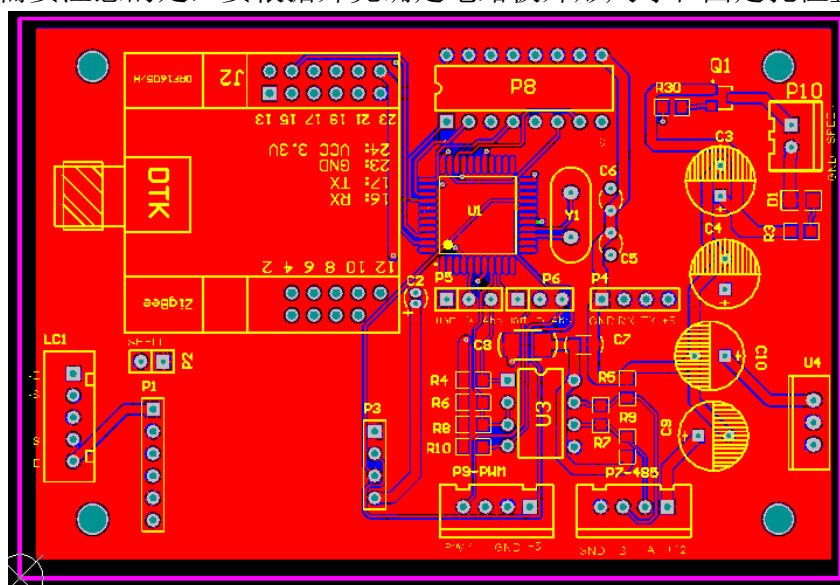


图 3.14 采集节点部分 PCB 电路板图

3.3 通信控制器硬件设计

3.3.1 处理器选型与设计

STM32 处理器按功能分为几个系列,即基本型、增强型、USB 基本型、互补型系列。其中增强型和基本型的时钟频率分别为 72MHz 和 36MHz,增强型系列时钟频率最高。STM32 系列处理器价格与 16 位同类产品相当,但性能大大提升,因此得到广泛的应用。二者内部集成的闪存大小从 32K 到 128K 不等,当处理器工作在 72MHz 时,功耗为 36mA,相当于 0.5mA/MHz,是同类产品最低的。本系统采用 STM32F103RCT6 处理器作为主控制器,时钟频率为 72MHz,增强型,程序存储器容量 256K,数据存储器 48K,内置多达 7 个定时器,2 个 12 位 ADC,7 通道 DMA,9 个通信接口,单周期乘法和硬件除法,大部分端口可以接 5V,LQFP 封装。

(1) 引脚排列:

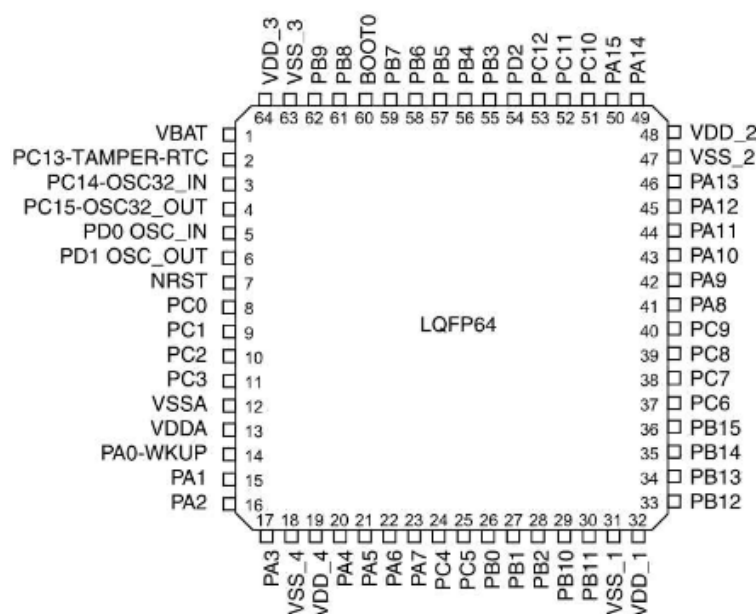


图 3.15 STM32F103 引脚图

(2) STM32F103RCT6 最小系统:

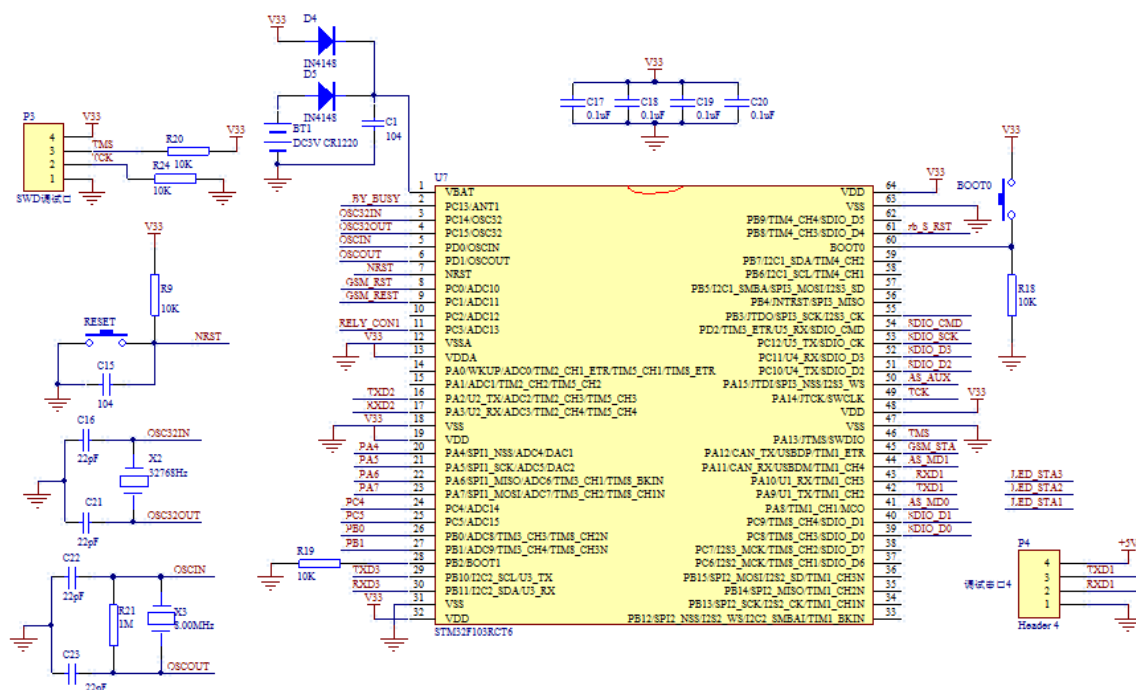


图 3.16 STM32F103RCT6 最小系统原理图

3.3.2 NB-IoT 选型与设计

本系统选用的 NB-IoT 模块为上海稳恒电子生产的 WH-NB75，兼容我国三大运营商 NB-IoT 网络连接。电路支持有人物联网公司的 7S3 系列模块，可以直接替换。此模块具有丰富的软件资源，可以接入两路 UDP，利用上位机软件进行 AT 指令配置，可以将模块设置成为具有 NB-IoT 网络转串口双向直传功能的模块。另外还符合两种协议，即 UDP、CoAP，这就为设计者创立服务器平台提供了协议支持。鉴于该模块使用场所多为室外，内部电路对低功耗进行了特殊的设计，使得该模块功耗非常低，电池供电情况下可以续航数年。WH-NB75 模块实物图如图 3.17 所示。



图 3.17 WH-NB75 模块实物图

覆盖能力强: NB-IoT 在室内应用场景下, 相较 LTE 表现出了高于 20dB 增益的覆盖率, 转换为空间增益为 100 倍;

功耗非常低：前面已有介绍，物联网系统多为电池供电，选取的无线模块功耗多少成为衡量该模块性能的主要参数；

成本非常低：相较其他无线技术，NB-IoT 技术简单的外围电路结构使得该模块成本非常低。

(2) WH-NB75 功能特点:

支持三种不同运营商 NB-IoT 网络的模块型号不同:

支持两路 UDP 简单透传模式;

支持七路 UDP 指令传输模式;

支持 CoAP 通信模式;

支持注册包功能;

支持串口和网络心跳包功能;

支持超低功耗模式:

支持 3GPP 标准指令集;

支持稳恒扩展指令集。

(3) 引脚定义:

WH-NB75 模块的封装为双列排针直插式，前 12 个引脚为一组排针，后 11 个引脚为一组排针，这样不对称结构可以有效防止模块被反接。WH-NB75 的正面引脚对应图如图 3.18 所示。



图 3.18 WH-NB75 引脚图

引脚定义如表 3.2 所示。

表 3.2 WH-NB75 的引脚定义

管脚	名称	I/O	引脚电平	说明
----	----	-----	------	----

1	NC			未使用引脚
2	NC			未使用引脚
3	NC			未使用引脚
4	NC			未使用引脚
5	UTXD2	O	3V	模块 LOG 串口发送脚
6	UTXD1	O	3V	串口发送引脚
7	URXD1	I	3V	串口接收引脚
8	URXD2	I	3V	模块 LOG 串口接收脚
9	WORK	O	3V	上电后为高电平，可以接灯用于运行标志
10	NC			未使用引脚
11	GND	G	/	模块供电负极
12	GND	G	/	模块供电负极
13	VIN	P	5-16V	电源电压：DC5-16V。
14	VIN			
15	NET	O	3V	该引脚接灯用于显示联网状态
16	VBAT	P	3.1-4.2V	供电引脚，禁止与 VIN 一起接入 电源电压：DC3.1-4.2V
17	GPIO8	I/O	3V	输入输出端口（不可用）
18	RELOAD	I	3V	保持该引脚电平为低 3-15 秒可以恢复出厂设置（不可用）
19	RESET	I	3V	复位引脚，拉低 200ms 以上模块复位
20	NC			未使用引脚
21	NC			未使用引脚
22	NC			未使用引脚
23	NC			未使用引脚

注：

NC 为空引脚，不接。

P 表示电源类引脚。

I 表示输入引脚。

O 表示输出引脚。

I/O 表示双向数据传输引脚。

RF 表示无线射频。

AI 表示模拟信号输入引脚。

AO 表示模拟信号输出引脚。

(4) 模块应用框图：

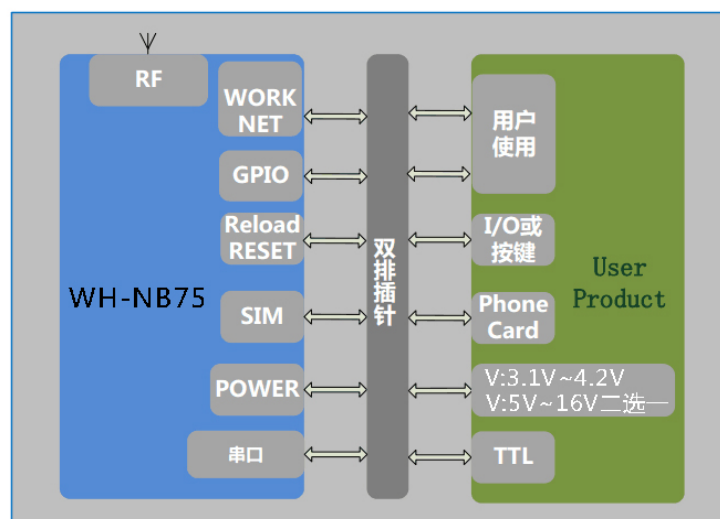


图 3.19 WH-NB75 应用框图

(5) SIM 卡接口：

WH-NB75 模块已内置符合标准的 SIM 卡接口，可以接两种电压的 SIM 卡，即 3.0、1.8 伏，模块可以通过内部程序检测卡的类型。在标准和低功耗两种模式下，模块提供的时钟频率分别为 3.25 和 1.08MHz。具有关闭时钟的功能，还可以经过参数设置，兼容快速 USIM 卡。可以进行 DMA 收发。在接收时，具有同时奇偶校验的功能。

设计者还可以按照系统要求重新设计 SIM 卡接口，需要注意的是，为避免更换卡时人体静电击穿 SIM 卡，务必考虑 ESD 电路设计，可以利用 TVS 管达到防静电的目的。元器件参数要求： V_{rwm} 为 5 伏， C_j 小于 10 皮法。另外，TVS 管需与模块共地，并且紧邻 SIM 接口。

(5) WH-NB75 外围电路如图3.20 所示。

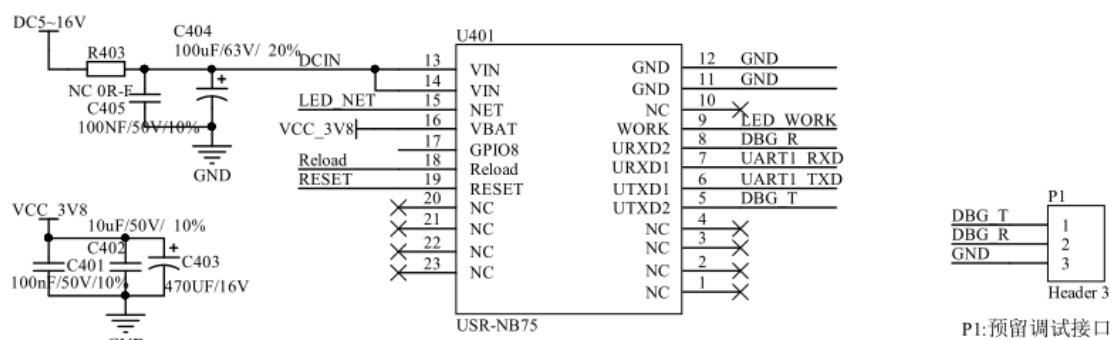


图 3.20 WH-NB75 应用接口电路

3.3.3 通信控制器部分 PCB 电路板图设计

按照 Altium Designer 软件 PCB 版图设计的步骤进行通信控制器部分 PCB 的绘制, 需要注意的是, 要根据外壳确定电路板外形尺寸和固定孔位置。

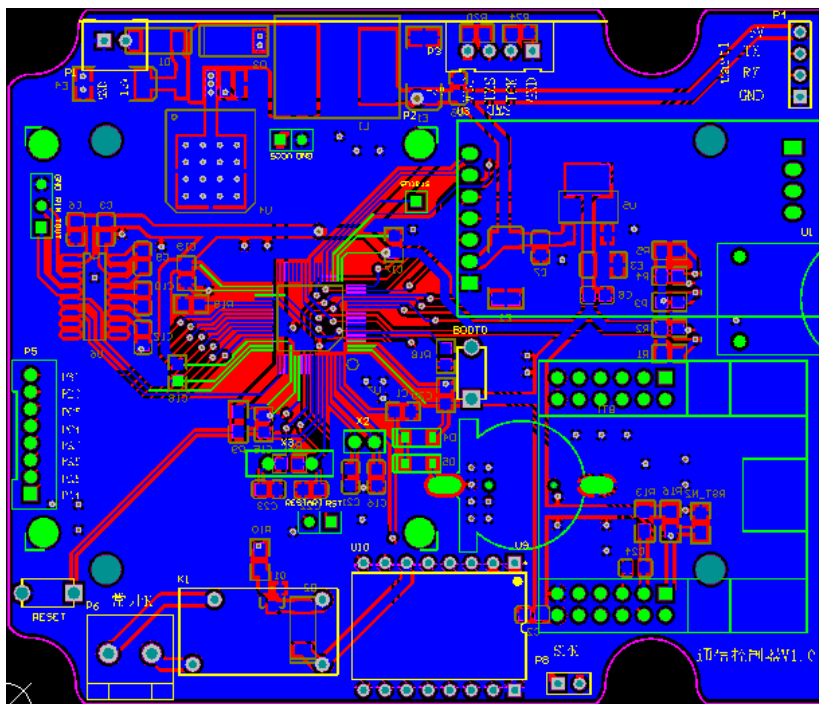


图 3.21 通信控制器部分 PCB 电路板图

第4章 系统软件设计

系统硬件电路设计完成后, 要实现系统的功能, 还需要进行与之匹配的软件系统设计。本论文采用模块化软件设计思想, 本章就以下几个模块程序的设计过程进行阐述: ZigBee 无线通信协议模块、采集节点模块、通信控制器模块、云平台模块和农作物生长模型模块。

4.1 ZigBee 无线通信协议模块软件设计

4.1.1 概述

以温室为单位, 通信控制器通过 ZigBee 通信网络采集每个温室的数据和状态, 并利用 NB-IoT 通信网络将信息进行上传。

4.1.2 Zigbee 组网

通信波特率: 9600bps, 8, 1, none。

Zigbee 配置方法为:

- (1) 通信控制器所置 ZigBee 为协调器 (Coordinator)。
- (2) 每个温室装有 1 个 Zigbee 模块, 需配置为终端通信控制器 (Router)。
- (3) 通信控制器和内网通信的每个温室的 Zigbee 模块 PAN ID 要相同。
- (4) 不同通信控制器的 Zigbee 协调器模块需配置为不同的 PAN ID。

4.1.3 通信协议

- (1) 所有通信采用如下通信格式如表 4.1 所示。

表 4.1 通信格式

数据名称	开始	地址	命令	数据	校验码
数据长度	1 字节	1 字节	1 字节	4 字节	一字节
说明	68H				CS

其中, 68H 是帧头, 校验码的值等于通信格式内前七个字节二进制算术和的低 8 位。

- (2) 地址通信格式如表 4.2 所示。

表 4.2 地址通信格式

地址	功能	说明
0x01	温室地址	1 号温室
0x02	温室地址	2 号温室

...
n	温室地址	n 号温室

(3) 命令通信格式如表 4.3 所示。

表 4.3 命令通信格式

命令	功能	说明
0x01	命令	问温度
0x02	命令	问湿度
0x03	命令	问光照度
0x04	命令	问二氧化碳浓度

(4) 数据通信格式，通信控制器下发数据通信格式如表 4.4 所示。

表 4.4 通信控制器下发数据通信格式

数据名称	数据块
长度	4 字节
说明	数据为四个 0x00

通信控制器空闲时轮询采集表中的每个参数值，每次查询计时 1000ms 等待采集终端的应答，收到应答后退出等待，进行下一项查询，超出时间无应答认为无此数据。

采集节点应答数据通信格式如表 4.5 所示。数据采集表如表 4.6 所示。

表 4.5 采集节点应答数据通信格式

数据名称	数据块
长度	4 字节
说明	未使用字节前置为 0x00, 已使用字节见数据采集表

表 4.6 数据采集表

序号	名称	长度	备注
1	温度	2 字节	两个字节代表温度值的 10 倍，高 8 位在前，低 8 位在后，比如 0x01, 0x64 代表的是 35.6℃。
2	湿度	2 字节	两个字节代表湿度值的 10 倍，高 8 位在前，低 8 位在后，比如 0x01, 0x64 代表的是 35.6%。
3	光照度	3 字节	三个字节代表光照度值，自左向右依次为：高 8 位、中 8 位、低 8 位，比如 0x01, 0x02, 0x64 代表的是 66148lux。
4	二氧化碳浓度	2 字节	两个字节代表二氧化碳浓度值，高 8 位在前，低 8 位在后，比如 0x01, 0x64 代表的是 356ppm。

(5) 举例: (16 进制)

问 1 号温室温度: 68 01 01 00 00 00 00 6A

1 号温室答: 68 01 01 00 00 01 64 CF (35.6℃)

问 1 号温室湿度: 68 01 02 00 00 00 00 6B

1 号温室答: 68 01 02 00 00 01 64 D0 (35.6%)

问 2 号温室光照度: 68 02 03 00 00 00 00 6D

2 号温室答: 68 02 03 00 01 02 64 D4 (66148lux)

问 2 号温室二氧化碳浓度: 68 02 04 00 00 00 00 6E

2 号温室答: 68 02 04 00 00 01 64 D3 (356ppm)

4.2 采集节点模块软件设计

4.2.1 温湿度采集部分程序设计

DHT11 的通讯方式与温度传感器 DS18B20 类似, 即单总线方式, 只需要占用单片机的一个端口进行数据传送。收到单片机取数据指令后, DHT11 将温度和湿度的小数、整数按顺序打包发送至单片机。数据包顺序为: 首先传送八位的湿度整数、八位的湿度小数, 然后是八位的温度整数、八位的温度小数, 最后传送八位的校验和。

微处理器端口与 DHT11 的 DATA 引脚连接后, 处理器主机发低电平开始信号, 然后拉高并延时等待, 此刻, DHT11 进入高速状态, 待处理器开始信号消失, 即转为低电平, 传感器开始响应并给处理器传送数据, 同时开始新一轮的数据转换。数据传输完毕后, DHT11 进入低速状态, 准备新一轮的传送数据请求。其通讯过程如图 4.1 所示。

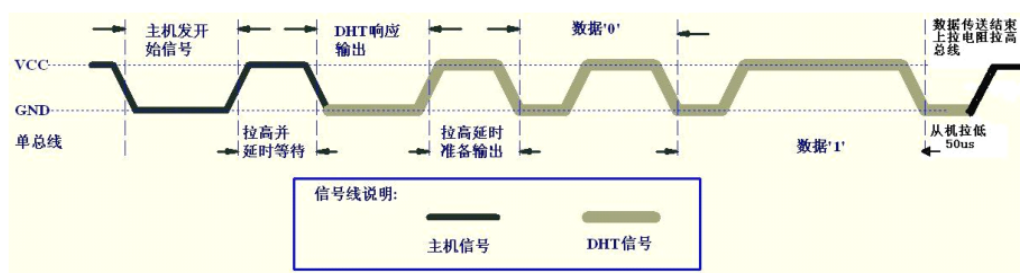


图 4.1 DHT11 通讯过程

与 DHT11 通讯的时序要求为: 单片机输出低电平开始信号, 延时大概 18ms, 然后输出高电平, 延时 20-40 μ s 后进行 DHT11 的响应的读取, DHT11 会首先输出低电平, 延时 30-50 μ s, 再输出高电平, 再延时 40 μ s, 然后开始数据包的传送。单片机接收到温湿度数据后, 根据通信格式进行分类解析, 再对数据进行滤波, 然后将数据暂存在内部数组中, 等待通信控制器的传送请求, 其具体流程如图 4.2 所示。

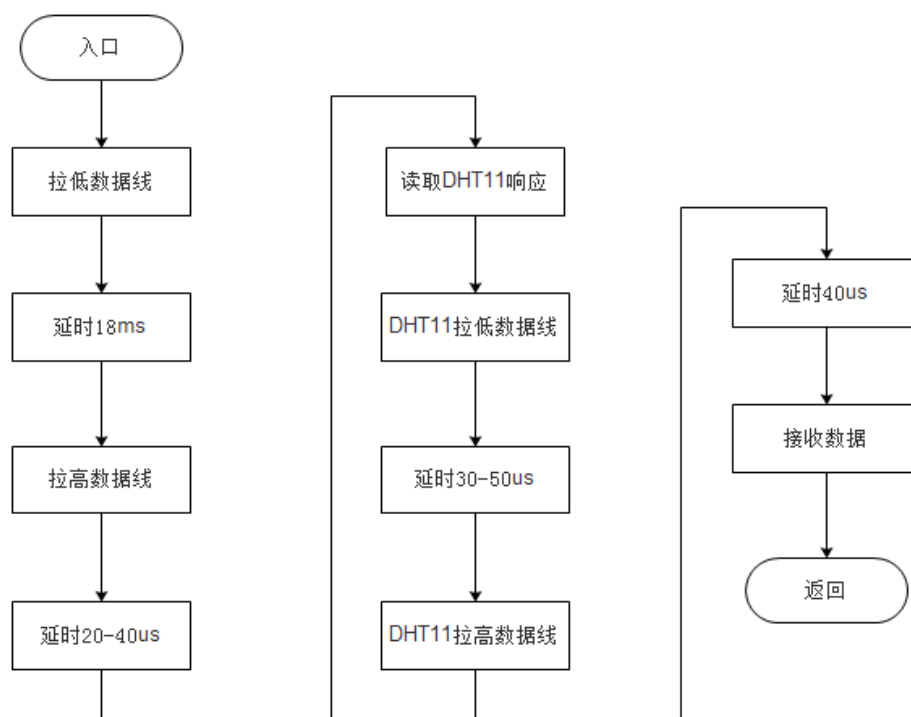


图 4.2 DHT11 工作流程图

程序清单:

```
uchar Read_Dat()
```

```
{
```

```
    uchar I;uchar Dat;
```

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

```
    {
```

```
        while( !DQ ); //DQ=0, 循环等待; DQ=1, 跳出循环
```

Delay60us(); //等待 60 μ s, 如果读取得到的数据为“0”, 说明高电平在 60 μ s 内已出现, 现在处于低电平已维持 30 μ s 的位置。反之, 现在仍处于高电平状态。

```
        Dat <<= 1;
```

```
        if( DQ )
```

```
        {
```

```
            Dat |= 1;
```

```
        } //若读取得到的数据是“0”, 则不执行该语句, 继续去执行循环等待函数 while(!DQ)直至耗尽 20  $\mu$ s 的低电平状态时间, 继续新一轮的通讯。
```

反之, 执行循环等待函数 while (DQ) 直至高电平消失, 再运行 while (!DQ) 语句直至耗尽 50 μ s 的低电平状态时间, 继续新一轮的通讯。

```
        while( DQ );
```

```

        }
        return( Dat );
    }

uint Read_DHT11()    // DHT11 响应时将数据线置低电平, 准备时置高电平。
{
    uint Dat;
    uchar RH1 , RH2 , TEMP1 , TEMP2 , JYW;
    //依次代表湿度整数、小数部分, 温度整数、小数部分, 校验值。
    DQ = 1;
    _nop_();
    DQ = 0;        //将数据线置低电平约 18ms。
    Delay19ms();
    DQ = 1;
    while( DQ ); //单片机置数据线为高电平, 然后进行 DHT11 响应的读取。
    while( !DQ ); //读取到 DHT11 响应, 延时等待 DHT11 准备就绪。
    while( DQ ); //读取到 DHT11 准备就绪后, 延时等待 DHT11 进行数据的
发送。

    RH1 = Read_Dat();
    RH2 = Read_Dat();
    TEMP1 = Read_Dat();
    TEMP2 = Read_Dat();
    JYW = Read_Dat();
    if( JYW == (RH1 + RH2 + TEMP1 + TEMP2) ) //校验值是否正确
        Dat = (( uint ) RH1 << 8) | TEMP1;
    else
        Dat = 0;
    return( Dat );
}

```

4.2.2 光照强度采集部分程序设计

BH1750 与单片机通过 I²C 总线进行通讯连接, 数据输出过程为: 首先通电发送指令 0x01, 然后发送模式指令, 延时等待光强测量至完成, 最后进行数据的传送。光照强度的计算公式为 (返回值*分辨率) / (1.2*灵敏度)。

下面详细说明通信过程, 主机把开始指令传给 BH1750, 同时传送地址和写指令, 延时等待 BH1750 响应, 下一步单片机把寄存器地址给 BH1750, 单片机延时

等待 BH1750 响应，最后单片机把停止指令给 BH1750，时序如图 4.3 所示。



图 4.3 时序图一

程序清单：

```
#define S_Add 0x46 //ADDR 设为低电平，模块地址是 0x46（即 01000110）
/*****
** 函数名：BH1750_Wr。
** 功能描述：BH1750 写寄存器。
** 输入参数：R_Add 寄存器地址。
** 输出参数：无。
** 返回：无。
*****/
void BH1750_Wr(u8 R_Add)
{
    IIC_Start();           //起始指令。
    IIC_Send_Byte(S_Add);  //把地址和写指令给传感器。
    IIC_Wait_Ack();        //延时等待响应。
    IIC_Send_Byte(R_Add);  //内部寄存器的地址。
    IIC_Wait_Ack();        //延时等待响应。
    IIC_Stop();            //停止指令。
}
```

BH1750 进行一次光照强度检测的时间小于 180 毫秒，输出数据长度为十六位，传输时高八位和低八位分开发送。主机把开始指令传给 BH1750，同时传送地址和写指令，延时一段时间后，BH1750 发送数据的高八位，完成后回传应答信号，下一步继续发送数据的低八位，本次不再回传应答信号，单片机把停止指令给 BH1750，至此完成一次数据传输过程。时序如图 4.4 所示。

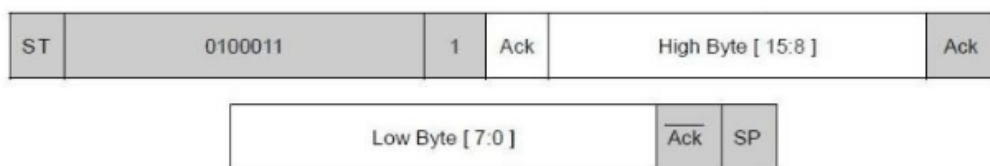


图 4.4 时序图二

程序清单：


```

/*****
** 函数名: BH1750_Read_Data
** 功能描述: 读取数据的高八位和低八位
** 输入参数: 无
** 输出参数: 无
** 返    回: 读取的数据
*****/
u16 BH1750_Read_Data(void)
{
    u16 hbyte=0, lbyte=0;
    IIC_Start();           //传送起始指令。
    IIC_Send_Byte(S_Add+1); //把地址和读指令给传感器。
    IIC_Ack();             //传感器响应。
    hbyte=IIC_Read_Byte(1);
    lbyte=IIC_Read_Byte(0);
    IIC_Stop();            //传送停止指令。
    return ((hbyte<<8)|lbyte);
}

```

4.2.3 MG811 二氧化碳浓度采集部分程序设计

MG811 传感器常用于探测家庭、温室环境内 CO₂ 的浓度, 可以将 0 至 10000ppm 的二氧化碳浓度值转换为 2-0V 的模拟量电压, 输出的电压值与浓度值成反比, 为降低环境温度对测量结果的影响, 内置了温度补偿电路, 提高了抗干扰性和数据的准确性。空间温度为标称值时, T_{cm}=0.5VCC, 还可以利用程序对测量数据进行校准, 进一步减小温度对探头的影响。

MG811 的灵敏度特性曲线如图 4.5 所示, 曲线说明了 MG811 敏感元件在几种常见气体, 各种浓度时的输出电压值。测试环境如下: 温度为 28 摄氏度, 湿度为 65% RH, 氧气的浓度为 21%。

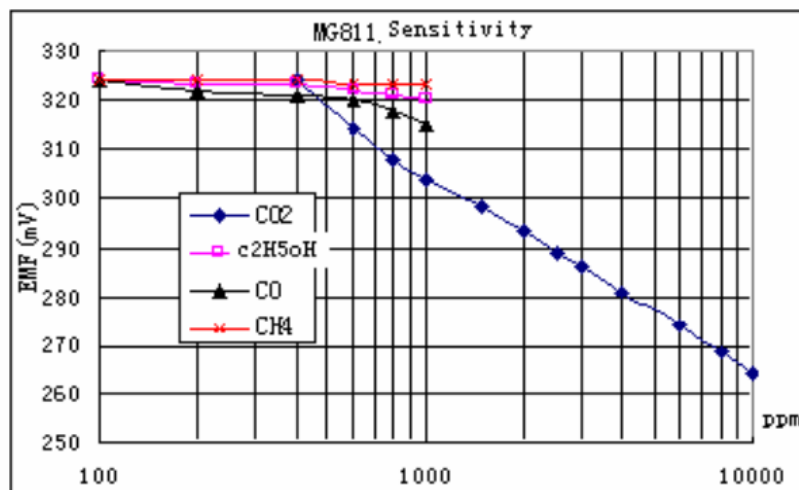


图 4.5 MG811 传感器的灵敏度特性曲线

程序清单:

```
#define Channel_MG811 0xe4      //P1.4 作为 MG811 传感器的 A/D 输入;
sfr P1ASF    = 0x9D;           //P1 口中哪一个引脚作为 AD 输入, 特殊
                                //功能寄存器 P1ASF 相对应的位要置 1。

sfr ADC_CONTR = 0xBC;          //A/D 转换控制寄存器。
sfr ADC_RES   = 0xBD;          //A/D 转换数据的高八位。
sfr ADC_RESL  = 0xBE;          //A/D 转换数据的低两位。

unsigned char ADC_change(unsigned char channel)    //AD 转换程序;
{
    ADC_CONTR=0x80;            //打开 A/D 转换电源。
    delay(10);                 //需要延时一段时间, 1 毫秒即可。
    P1ASF=(0x01<<channel);    //将 A/D 输入口设作模拟功能端口。
    switch(channel)            //转换通道选择;
    {
        case 4: ADC_CONTR=Channel_MG811; break;
    }

    delay(10);                //A/D 转换通道不变时, 可以去掉该语句, 当通道发生变化时, 为了保证转换更精确, 需要保证电源电压的恒定, 若电源内阻小于 10 千欧,
                                //可以去掉该语句, 否则需要增加 20 至 200 微秒的延时程序。

    ADC_RES =0;               //将高八位寄存器清零。
    // ADC_RESL=0;           //将低两位寄存器清零。
    ADC_CONTR = ADC_CONTR|0x08; //AD 转换开始
    delay(10);                //因上一语句刚对 ADC_CONTR 寄存器写
```

入新的数据,此时需要延时一段时间,才可以进行读操作。

```
while((ADC_CONTR & 0x10)==0); //0001,0000 循环至 A/D 转换完成。
ADC_CONTR=0;          //将 ADC_FLAG、ADC_START 位置零,A/D 转换结束。
return(ADC_RES);
}
```

4.2.4 单片机程序设计

(1) STC12C5A60S2 内置 AD 的应用:

STC12C5A60S2 单片机自带八路十位电压输入 A/D 转换器,简化了模数转换外围电路的设计,降低了成本,广泛应用于温度、按键、频谱、电压等采集中。转换频率最高为 250KHz,为 P1.0-P1.7 端口的第二功能。根据硬件电路,查看被测信号接入哪一个引脚,然后利用程序将该引脚设置为 AD 输入引脚,P1 口的其他引脚功能、特性不变。与 A/D 转换有关的特殊功能控制寄存器如下:

P1ASF 寄存器: P1 口中哪一个引脚作为 AD 输入,特殊功能寄存器 P1ASF 相对应的位要置 1。默认值为 0,即用作普通输入输出端口。不可以进行位寻址操作。

ADC_CONTR 寄存器: 不可以进行位寻址操作,下面按位介绍功能:

第七位, ADC_POWER, 模数转换开关,使用时置 1,不使用时置 0;

第六位, SPEED1, 第五位, SPEED0, 组合设置模数转换速率,“00”时,完成一次转换需要 540 个时钟周期,“01”、“10”、“11”时,分别需要 360、180、90 个。使用时需要合理选择设置,转换数据的准确度会随着转换速率的提高而降低。

第四位, FLAG, 模数转换完成后,该位自动置 1,可以作为程序转换完成检测,需要注意的是,该位需要通过软件置 0。

第三位, SRART, 该位被置“1”时,模数转换开始。

第二位至第零位, CHS2-0, 模拟输入通道选择设置位,为 8421BCD 码,如表 4.7 所示。

表 4.7 设置 CHS2、CHS1、CHS0 选择模拟输入通道

CHS2	CHS1	CHS0	模拟输入通道选择
0	0	0	选择 P1.0 作为 A/D 输入来用
0	0	1	选择 P1.1 作为 A/D 输入来用
0	1	0	选择 P1.2 作为 A/D 输入来用
0	1	1	选择 P1.3 作为 A/D 输入来用
1	0	0	选择 P1.4 作为 A/D 输入来用
1	0	1	选择 P1.5 作为 A/D 输入来用
1	1	0	选择 P1.6 作为 A/D 输入来用
1	1	1	选择 P1.7 作为 A/D 输入来用

ADC_RES: 模数转换数据高八位。

ADC_RESL: 模数转换数据高两位。

AUXR1: 一般作为辅助用, 默认设置即可。

IE 寄存器: 与模数转换相关的是第六位, AD 中断允许位, 使用时需要置 1, 另外还需要将第七位 EA 置 1。

IP: 优先级控制低字节。

IPH: 优先级控制高字节。

IP 和 IPH 的第五位作为模数转换中断优先级设置位: (其中 0 表示最低, 3 表示最高)

在 PADCH、PADC 均为 0 的情况下, 优先级为 0。

在 PADCH=0、PADC=1 的情况下, 优先级为 1。

在 PADCH=1、PADC=0 的情况下, 优先级为 2。

在 PADCH、PADC 均为 1 的情况下, 优先级为 3。

需要注意的是, 仅仅普通 51 含有的寄存器支持位寻址, 因此以上寄存器, 只有 IE 和 IP 可以进行位寻址操作, 其余的不支持。我们通常在程序中使用“#include<reg51.h>”语句, 然后利用 sfr 定义新添加的特殊功能寄存器, 程序中使用“|”和“&”进行寄存器的字节赋值即可, 避免了 sbit 位定义带来的麻烦。

(2) 程序清单 (以 10 号温室为例):

```

/*****
芯片: STC12C5A60S2
晶振频率: 11.0592MHz 波特率: 9600bps
串口二端口: 发送端口 (TXD2): P1.3; 接收端口 (RXD2): P1.2
*****/
unsigned char BUF_WD[8]={0x68, 0x0a, 0x01, 0, 0, 0, 0, 0}; //应答温度。
unsigned char BUF_SD[8]={0x68, 0x0a, 0x02, 0, 0, 0, 0, 0}; //应答湿度。
unsigned char BUF_GZ[8]={0x68, 0x0a, 0x03, 0, 0, 0, 0, 0}; //应答光照。
unsigned char BUF_CO2[8]={0x68, 0x0a, 0x04, 0, 0, 0, 0, 0}; //应答二氧化碳。
unsigned char BUF_UART2[16]={0x68, 0x0a, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
//无线串口接收缓冲。

//unsigned char BUF_WD[8]={0x68, 0x0a, 0x01, 0, 0, 0, 0, 0x73}; //问温度。
//unsigned char BUF_SD[8]={0x68, 0x0a, 0x02, 0, 0, 0, 0, 0x74}; //问湿度。
//unsigned char BUF_GZ[8]={0x68, 0x0a, 0x03, 0, 0, 0, 0, 0x75}; //问光照。
//unsigned char BUF_CO2[8]={0x68, 0x0a, 0x04, 0, 0, 0, 0, 0x76}; //问二氧化碳浓度。

void InitUART(void)

```

```

{
    IPH2=0x01;
    IP2=0x01;
    S2CON = 0x50;    //设置为方式一，S2REN 位置高，可以接收。
    BRT = 0xFD;      //设置相关寄存器初值。
    AUXR = 0x10;      //BRTR 置高，启动计数。
    IE2 =0x01;        //允许串口二中断，置高 ES2。
    TMOD |= 0x01;     //设置 T0 工作方式。
    TH0 = (65536-50000)/256;
    TL0 = (65536-50000)%256;
    TR0 = 0;          //停止定时器。
    ET0 = 0;
    EA =1;             //允许总中断。
}

/*****串口 2 中断处理函数*****/
void UART_2Interrupt(void) interrupt 8
{
    //    UART_2Interrupt_flag=1;
    if (S2CON&S2RI)
    {
        S2CON&=~S2RI;
        TH0 = (65536-50000)/256;
        TL0 = (65536-50000)%256;
        TR0 = 1;
        ET0 = 1;
        BUF_UART2[j] = S2BUF;    //接收数据。
        j++;
    }
}

void Time_0(void) interrupt 1    //定时 50ms。
{
    if(j==8)
    {
        if (BUF_UART2[7]==0x73)
            if (BUF_UART2[0]==0x68)

```

```

        if (BUF_UART2[1]==0x0a)
            if (BUF_UART2[2]==0x01)        //应答温度。
        {
            for(i=0;i<8;i++)
            {
                S2BUF = BUF_WD[i];
                while(!(S2CON&S2TI)); //如果 S2TI 为零，循环等待。
                S2CON&=~S2TI;        //S2TI=0。
            }
        }
        if (BUF_UART2[7]==0x74)
            if (BUF_UART2[0]==0x68)
                if (BUF_UART2[1]==0x0a)
                    if (BUF_UART2[2]==0x02)        //应答湿度。
        {
            for(i=0;i<8;i++)
            {
                S2BUF = BUF_SD[i];
                while(!(S2CON&S2TI)); //如果 S2TI 为零，循环等待。
                S2CON&=~S2TI;        //S2TI=0。
            }
        }
        if (BUF_UART2[7]==0x75)
            if (BUF_UART2[0]==0x68)
                if (BUF_UART2[1]==0x0a)
                    if (BUF_UART2[2]==0x03)        //应答光照。
        {
            for(i=0;i<8;i++)
            {
                S2BUF = BUF_GZ[i];
                while(!(S2CON&S2TI)); //如果 S2TI 为零，循环等待。
                S2CON&=~S2TI;        //S2TI=0。
            }
        }
        if (BUF_UART2[7]==0x76)

```

```

        if (BUF_UART2[0]==0x68)
            if (BUF_UART2[1]==0x0a)
                if (BUF_UART2[2]==0x04)    //应答二氧化碳。
                {
                    for (i=0;i<8;i++)
                    {
                        S2BUF = BUF_CO2[i];
                        while(!(S2CON&S2TI)); //如果 S2TI 为零，循环等待。
                        S2CON&=~S2TI;        //S2TI=0。
                    }
                }
            }
        j=0;
        TR0 = 0;
        ET0 = 0;
    }

```

4.3 通信控制器部分程序设计

4.3.1 STM32 处理器的程序设计

通信控制器部分 STM32 处理器程序设计主要涉及到两个串口的编程，即串口转 ZigBee 无线模块和串口转 NB-IoT 远程通信模块。其中前者负责与给各温室采集节点部分进行数据通讯，系统调试时，将设置同一网络内每一个 ZigBee 无线模块的信道和 PIN ID 为相同参数，其中通信控制器所置无线模块为协调器，各温室所置无线模块为终端。通电后，自动组网，实现数据透传。STM32 通过串口发送指令和接收数据，无线模块负责将串口上的数据通过无线的方式发送出去。采集节点单片机通过串口接收指令和发送数据，STM32 将串口接收到的数据解析处理后再经过另一串口发送给 NB-IoT 通信模块。根据以上通讯过程原理，仅仅考虑串口收发程序编程即可，不必考虑无线通信的机理，无线模块硬件和 ZigBee 协议栈负责数据的无线传送。

后者是 STM32 与 WH-NB75 模块之间的数据通讯，二者之间通信遵守固定的通信协议。STM32 将串口接收到的各个温室的环境参数数据，经过解析和处理，再通过其他串口发送给 WH-NB75 模块，然后利用 NB-IoT 网络把数据远距离发送给云平台服务器。本系统采用了有人科技的透传云平台，设计者可以在平台上增加或减少设备，仅仅考虑串口收发程序编程即可，不必考虑远距离无线通信的机理，

WH-NB75 模块硬件和 CoAP 协议负责数据准确、稳定的远距离传送。该透传云平台集成了 APP 和网页开发示例，只需稍作修改，即可应用到实际项目中，其具体设计方法将在后面的模块进行描述。

串口初始化和服务程序：

```
void uart_init(u32 bound) {
    GPIO_InitTypeDef GPIO_InitStructure;    //GPIO 端口初始化
    USART_InitTypeDef USART_InitStructure;
    NVIC_InitTypeDef NVIC_InitStructure;
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1|RCC_APB2Periph_GPIOA,
ENABLE);                                     //允许串口 1、GPIOA 时钟
    USART_DeInit(USART1);    //复位串口 1      //串口 1 的 TX 引脚 PA. 9
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_9; //PA. 9
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP; //复用推挽输出
    GPIO_Init(GPIOA, &GPIO_InitStructure); //初始化 PA. 9
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_10; //串口 1 的 RX 引脚 PA. 10
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING; //浮空输入
    GPIO_Init(GPIOA, &GPIO_InitStructure); //初始化 PA. 10
    //对串口 1 的 NVIC 配置
    NVIC_InitStructure.NVIC_IRQChannel = USART1_IRQn;
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority=2 ;
                                                    //抢占优先级 3
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 1; //子优先级 3
    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE; //IRQ 通道使能
    NVIC_Init(&NVIC_InitStructure);                //配置 NVIC 寄存器
    //USART 初始化设置
    USART_InitStructure.USART_BaudRate = bound; //一般设置为 9600;
    USART_InitStructure.USART_WordLength = USART_WordLength_8b;
                                                    //字长为 8 位数据格式
    USART_InitStructure.USART_StopBits = USART_StopBits_1; //一停止位
    USART_InitStructure.USART_Parity = USART_Parity_No; //无奇偶校验
    USART_InitStructure.USART_HardwareFlowControl=
    USART_HardwareFlowControl_None;                //无硬件数据流控制
    USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;
    //收发模式
```



```

    USART_Init(USART1, &USART_InitStructure); //初始化串口
//中断配置
//USART_ITConfig(USART1, USART_IT_RXNE, ENABLE);
USART_ITConfig(USART1, USART_IT_TC, DISABLE);
USART_ITConfig(USART1, USART_IT_RXNE, DISABLE);
USART_ITConfig(USART1, USART_IT_IDLE, ENABLE);
MYDMA_Config(RCC_AHBPeriph_DMA1, DMA1_Channel5, DMA_DIR_PeripheralSRC, (u32)&USART1->DR, (u32)USART_RX_BUF, USART_REC_LEN, DMA_Priority_VeryHigh); //DMA1 通道 5, 外设为串口 1, 存储器为 updata_table, 长度 80.
MYDMA_Config(RCC_AHBPeriph_DMA1, DMA1_Channel4, DMA_DIR_PeripheralDST, (u32)&USART1->DR, (u32)updata_table, 80, DMA_Priority_VeryHigh); //DMA1 通道 4, 外设为串口 1, 存储器为 updata_table, 长度 80.
//采用 DMA 方式发送和接收
USART_DMAMCmd(USART1, USART_DMAMReq_Tx|USART_DMAMReq_Rx, ENABLE);
//使能 channel5
DMA_Cmd(DMA1_Channel5, ENABLE); //使能串口
USART_Cmd(USART1, ENABLE);
}
void USART1_IRQHandler(void)
{
    if(USART_GetITStatus(USART1, USART_IT_IDLE) != RESET)
    {
        USART_RX_STA = USART1->SR;
        USART_RX_STA = USART1->DR; //清 USART_IT_IDLE 标志
        DMA_Cmd(DMA1_Channel5, DISABLE);
        USART_RX_STA=USART_REC_LEN-DMA_GetCurrDataCounter(DMA1_Channel5);
        USART_RX_STA|=1<<15; //强制标记接收完成
        DMA_SetCurrDataCounter(DMA1_Channel5, USART_REC_LEN);
        DMA_Cmd(DMA1_Channel5, ENABLE);
    }
}

```

4.3.2 基于 μ CosII 的软件设计与移植

(1) 软件平台与移植步骤:

软件设计与移植利用的平台为 RVMDK。它是在 Keil 的基础上集成了 ARM 开发

编译、调试功能,具有非常友好的开发环境,开发基于 Cortex-M3 内核的微处理器程序非常方便^[40]。如果设计者有 51 单片机的 Keil 软件开发基础,再进行 STM32 的程序设计与系统移植会非常容易。移植过程中用到的两个程序文件,一个是通信控制器程序,另一个是基于 STM32 的 μ COS II 移植系统文件^[41]。

μ COS II 系统的移植步骤分为三步,首先要明确通信控制器程序文件和 μ COS II 移植文件的文件组成;然后需要进行 μ COS II 文件的移植和把工程文件添加到 MDK 软件里^[42];再看一下满足系统运行所必需的条件,并进行相应程序的改进。

(2) 文件结构:

通信控制器程序文件由 4 个文件夹组成。其中,输出文件夹的作用是存放程序编译过程中出现的临时性文件;Project 文件夹的作用是存放一些启动性的文件;Libraries 文件夹的作用是存放 STM32 微处理器和外围元器件驱动头文件^[43];MyCode 文件夹的作用是存放设计者的应用程序文件。

Libraries 文件夹是由 2 个文件夹组成的。CMSIS 文件夹的作用是存放 STM32 微处理器内核的启动性文件;STM32F10x_StdPeriph_Driver 文件夹的作用是存放 STM32 微处理器里面和所连接芯片的驱动程序。

基于 STM32 的 μ COS II 移植文件夹中包含了整个 μ COS II 内核和通信协议文件。 μ COS II 文件中包含了四个文件夹。Ports 文件夹包含了 μ COS II 的接口文件;uC-CPU 文件夹包含了关于 CPU 的驱动文件^[44];User 文件夹中是用户对 μ COS II 初始化配置文件;Source 文件夹包含 μ COS II 主要源文件。

(3) μ COS II 文件移植:

首先,在通信控制器程序文件主文件夹新建一个 μ COS II 文件夹,然后将 Port、Source、uC-CPU 三个文件夹复制到 μ COS II 文件夹中^[45],将 User 文件夹中的三个头文件复制到 MyCode 文件夹中。

启动 MDK 软件,打开添加文件对话框,添加两个文件夹并命名为: μ COS II/Port、 μ COS II/Source^[46]。接着在每个工程文件夹里添加之前复制的文件,并注意分类。单击保存按钮,至此完成了 MDK 对 μ COS II 移植文件的添加。下一步把添加移植文件所需的头文件的路径导入到工程里。

至此, μ COS II 文件添加完毕,单击编译按钮,对以上操作结果进行检查。移植并编译成功的标志是软件编译提示窗口显示“0 errors, 0 warnings”。

(4) 需要修改的代码:

第一处: μ COS II 任务切换是通过 STM32 中的软中断实现的,需要将 STM32 中的软中断名修改成与 μ COS II 软中断函数名一致^[47]。其修改过程如图 4.6、图 4.7 所示。

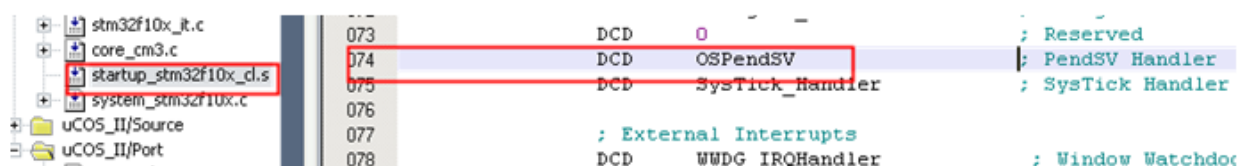


图 4.6 软中断名的修改一

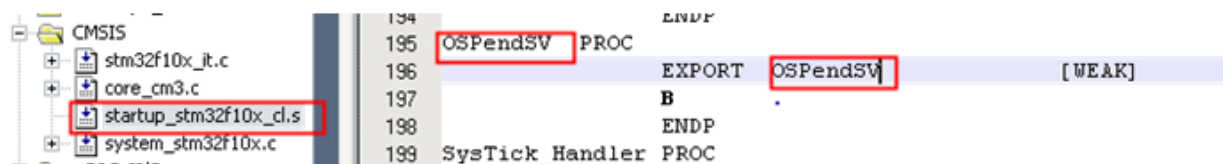


图 4.7 软中断名的修改二

第二处： μ COS II 任务切换是以 STM32 中的 SysTick 作为时间基准^[48]，需要编写如图 4.8、图 4.9 所示的 SysTick_Handler() 函数和 SysTick 初始化函数。

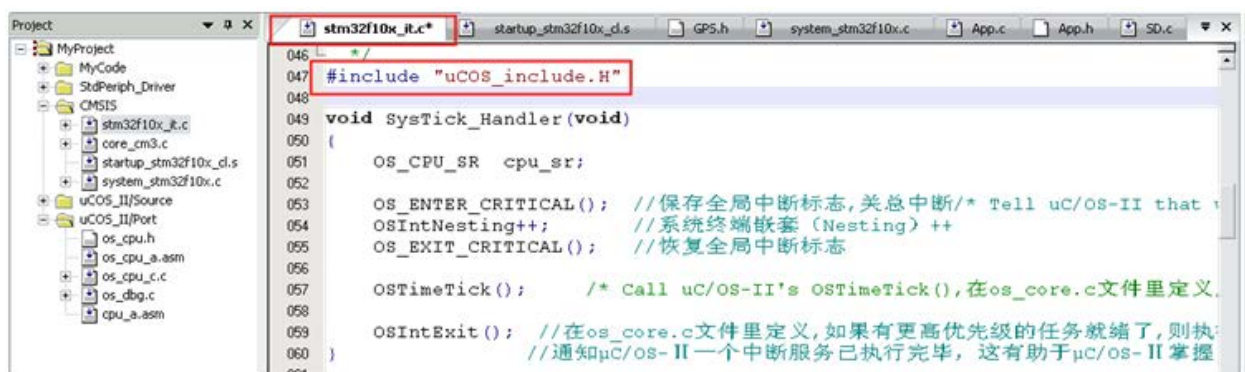


图 4.8 SysTick 中断函数

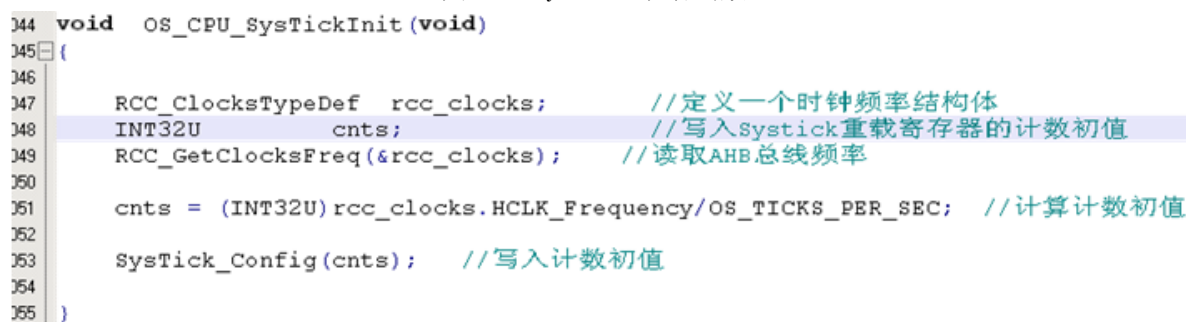


图 4.9 SysTick 初始化函数

第三处：用户可以根据实际情况修改任务切换的时间^[49]，其修该位置为如图 4.10 所示红框中的数值，10 表示任务切换的时间间隔为 1000/10ms。

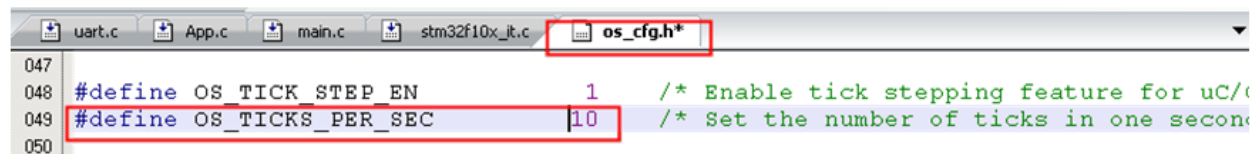


图 4.10 任务切换的时间设置

至此，代码编辑结束。单击编译按钮，对代码编辑的结果进行检查。代码编

辑并编译成功的标志是软件编译提示窗口显示“0 errors, 0 warnings”。

最后, 系统的运行需要包含任务, 根据系统的功能要求, 把程序中需要完成的工作分类加入至任务列表里, 任务之间还需要进行必要的通信, 再创建信号量、消息邮箱、消息队列等达到实现任务间的通信的目的。

4.4 云平台设计

本系统通过有人科技的透传云平台进行远程数据的接收和发送, 设计者可以在平台上增加或减少设备, 还具有编辑设备信息、设置报警阈值、短信或微信报警等功能。设计者仅仅考虑网页或终端软件的编程即可, 不必考虑数据远距离无线通信的收发机理, CoAP 协议负责通讯的准确率、稳定性。该透传云平台提供了 APP 和网页开发示例源代码, 只需根据功能要求稍作修改, 便可运用至自己的项目里。另外, 平台支持给所添加的设备传输数据, 实现了双向通讯。

4.4.1 WH-NB75 的工作模式

WH-NB75 可以利用串口 AT 指令改变工作模式, 包括 CMD 指令 (含 AT 指令 UDP 功能和 AT 指令 CoAP 功能)、CoAP 透传和简单透传 (NET) 三种模式^[50], 下面进行一一描述。

(1) CMD 指令模式:

出厂时, 已将模块设置为 CMD 模式, 该模式支持参数配置指令、UDP 通信指令、CoAP 通信指令三种指令。可以改变模块的参数, 建立 UDP 通道, 与符合 CoAP 的云服务之间通信。

CMD 指令模式中的 AT 指令 UDP 功能使用起来比简单透传 (NET) 模式更加方便, 但是也有不足的地方在于该模式 AT 指令使用过于频繁。二者的相同之处是创建连接、收发数据均需要操作 AT 指令。需要注意的是, WH-NB75 可以进行七路网络连接, 但是只可以是 IP 地址, 不能用域名。

CMD 指令模式中指令 CoAP 功能的优缺点同上, 二者均需要在成功连接网络且确定驻网方式后才可以数据收发, 区别在于, CoAP 通信严格遵守 CoAP 协议, 因此比 UDP 功能模式稳定一些。

(2) CoAP 透传模式:

CoAP 透传模式不要与 CMD 指令模式中的指令 CoAP 功能混淆, 二者截然不同。前者是一种收发机制, 仅仅确定服务器地址和端口号就可以, 然后装有 WH-NB75 的串口系统便可以将数据传输到固定的 CoAP 远程服务器, 反方向的通信也是成立的, 因此实现了数据的双向传输。

(3) NET 简单透传模式:

和 CoAP 透传模式操作方法相同, 经过简单设置, 可以实现串口系统与网络服

务器之间数据的双向传输，又叫 UDP 透传模式。通信时的数据转换由 WH-NB75 内置程序负责实现，可以最多进行两路连接。

（4）心跳包

当工作模式为 NET 简单透传时，模块支持传输心跳包，接收方是网络服务器端或者串口系统，但只能有一个接收方。该功能主要应用于以下两个场景中：一是模块和服务需要始终通信，可以达到低功耗的目的；二是模块起到服务器的作用，向采集节点发送数据请求，可以达到节省流量的目的。

心跳包的长度小于二十个字节。

（5）注册包

为了提高通讯的安全性，使服务器区分模块的编号或者为模块连接服务器设置密码。模块透传模式时支持发送注册包功能，使用时将 ICCID、IMEI、IMSI 码置于数据包前，作为握手数据，设计者还可以根据需要来设定，长度小于二十个字节即可。

（6）成帧机制

分为两种触发模式，即时间和长度。其中，时间触发是指接收到一个字节数据后，系统启动计时，如果计时时间大于固定时间后，未接收到下一字节数据，停止接收。如果计时时间均小于固定时间，则以打包长度为停止接收信号。该固定时间缺省值 100 毫秒，可由程序设置为介于 10 到 500 毫秒之间的数值。长度触发是指接收数据长度固定，以长度为停止接收信号，固定长度缺省值 512，还可由程序“AT+UARTTL”配置为介于 1 到 512 之间的数值。

（7）AT 指令配置

如果 WH-NB75 目前处于非 AT 指令模式，利用上位机或处理器给模块传送固定数据包，可以使 WH-NB75 进入指令设置模式，如果设置结束，再传送“AT+ENTM”使之切换至原来的状态，并回传“+OK”作为提示，时间在 3 秒以内。

需要注意的是，第一次使用 WH-NB75 时，模块一般处于 AT 指令模式状态，随后根据需要发送 AT 指令进行配置。如果当前工作状态为 NET 模式，配置结束时，需要发送“AT+S”对配置内容进行保存，否则需要重新进行配置。

（8）AT 指令集如表 4.8 所示。

表 4.8 AT 指令集

指令	功能描述
	3GPP 标准指令
AT	测试指令
AT+CGMI	查询制造商信息
AT+CGMM	查询模块型号
AT+CGMR	查询模块软件版本信息
AT+CGSN	查询模块序列号

AT+CEREG	查询注网状态
AT+CSCON	查询信号状态
AT+CLAC	列出可用指令
AT+CSQ	查询信号强度
AT+COPS	接入点参数设置
AT+CGATT	PS 网络激活操作
AT+CGACT	PDP 上下文激活或停用
AT+CIMI	查询 SIM 卡的 IMSI 值
AT+CGPADDR	查询模块获取的 IP
AT+CGDCONT	定义 PDP 接入点信息
AT+CFUN	设置终端射频功能
AT+CMEE	移动终端错误报告
AT+CCLK	获取网络时间
AT+CPSMS	省电模式设置
AT+CEDRXS	eDRX 参数设置
AT+CEDRXRXP	eDRX 阅读动态参数 特殊指令
AT+NMGS	发送数据到 CDP 服务器
AT+NMGR	接收 CDP 服务器数据
AT+NNMI	数据到达提示
AT+NSMI	设置数据发送提示
AT+NQMGR	汇总消息接收情况
AT+NQMGS	汇总消息发送情况
AT+NMSTATUS	汇总消息发送情况
AT+NCDP	设置/查询 CDP 服务器设置
AT+NUESTATS	模块当前状态查询
AT+NEARFCN	指定搜索频段
AT+NSOCR	创建一个 socket
AT+NSOST	通过指定 Socket 发送一条信息
AT+NSOSTF	通过指定 Socket 发送标记数据
AT+ NSORF	读取 UDP 数据
AT+NSOCL	关闭 Socket
+NSONMI	数据接收标志
AT+NPING	PING 功能
AT+NBAND	设置 Band
AT+NLOGLEVE	设置 LOG 等级
L	
AT+NCONFIG	模块功能配置
AT+NATSPEED	配置串口波特率
AT+NCCID	读取 ICCID
AT+NCHIPINF	读取系统信息
O	
AT+NRB	模块重启
	稳恒通用扩展指令

AT+BUILD	查询固件时间版本
AT+VER	固件版本号
AT+PDTIME	查询模块生产时间
AT+SN	查询 SN
AT+WKMOD	设置/查询工作模式
AT+S	保存参数
AT+ENTM	退出临时 AT 指令模式
AT+RELD	恢复出厂设置
AT+RSTPF	启停打印输出
AT+SOCKAEN	设置/查询 socketA 使能
AT+SOCKPORT	设置/查询 socketA 本地端口
A	
AT+SOCKA	配置 A 目标 IP 地址以及端口
AT+SOCKBEN	设置/查询 socketB 使能
AT+SOCKPORT	设置/查询 socketB 本地端口
B	
AT+SOCKB	配置 B 目标 IP 地址以及端口
AT+UARTTL	配置数据通信的打包时间、长度
AT+HEARTEN	设置/查询心跳包使能
AT+HEARTTP	设置/查询心跳发送方式
AT+HEARTTM	设置/查询心跳时间
AT+HEARTDT	设置/查询心跳包内容
AT+REGEN	设置/查询注册包使能
AT+REGTCP	设置/查询注册方式
AT+REGUSR	设置/查询注册包内容
AT+COAPAGE	设置/查询 coap 超时时间
AT+XVER	查询固件小版本
AT+COAPRPY	配置是否进行 COAP 传输

4.4.2 有人透传云平台设计

(1) 本系统使用了 WH-NB75 模块的 CoAP 透传工作方式，操作容易，仅仅通过上位机配置工作模式由 AT 指令切换至 CoAP 透传，然后输入 COAP 服务器 IP 地址和端口就可以完成对模块的配置。

(2) 启动 USR-NB V1.0.0 上位机软件，选择好串口号、波特率、校验位，单击“打开串口”按钮，切换至“参数设置”面板，首先单击“读取当前参数”按钮，接着单击“CoAP 透传模式”单选框，在地址和端口栏键入“117.60.157.137”和“5683”，最后左键单击“设置所有参数”按钮，结束配置，延时等待 WH-NB75 重启完成。

需要注意的是，完成以上配置的 WH-NB75 已处于 CoAP 透传工作方式，若要对更改设置，首先将模块切换至配置状态，然后进行 AT 指令通信即可。

(3) 下一步，需要验证网络连接情况，利用 USR-NB V1.0.0 上位机软件，也

可以使用串口调试助手软件, 设置好串口通信相关参数, 然后给模块传输数据, 若联网成功, 将自动上传至 CoAP 服务器。

(4) 本系统经过对比确定选用有人科技透传云作为系统数据的云平台, 打开网址: <http://cloud.usr.cn/>, 第一步需要添加账户, 添加完成后登录即可。

(5) 左键单击网页右上角的“注册”, 在弹出的对话框中键入用户名、密码、公司名、邮箱后单击“注册”按钮。然后打开邮箱, 打开验证链接, 最后回到网页首页, 单击右上角“登录”按钮。

(6) 依次单击“设备管理”、“添加设备”, 在打开的网页内输入下面的内容:
所属用户栏选择缺省值;

设备分组栏选择默认分组;

设备名称栏可以键入温室大棚种植区名称、地址、编号等信息, 例如, 此处我们键入“WS001”;

设备类型栏单击选中“电信 CoAP/NB-IoT”;

IMEI、SN 栏依次键入模块外包装中标注的 IMEI、SN 码信息, 需要注意的是, SN、IMEI 码的位数是 15, 其中, SN 码仅仅输入数字即可。

通信协议栏支持设计者编辑或“数据透传”, 在此, 我们选择后者。

(7) 单击“下一步”按钮, 打开并添加模块图像和地图定位。然后单击“保存”按钮, 新增模块结束, 打开设备列表界面验证是否成功。

(8) 打开“二次开发”面板, 云平台提供了 PC 端开发、Web 网页端开发、JAVA 或安卓开发三种方式, 在此, 我们选择 Web 网页端开发之在线演示, 接着输入用户名和密码, 单击“连接”按钮。

(9) 打开“云交换机”面板, 进行订阅模块, 选择“单个设备”, 在“设备 ID”栏键入模块外包装中的 IMEI 码。单击“订阅”按钮, 提示“订阅成功”, 如图 4.11 所示。至此, 实现了 WH-NB75 与服务器之间的数据相互传输设置, 打开云交换机面板的“收到消息”框和串口收发界面可以查看二者之间的通讯内容。

(10) 需要注意的是, PSM 为省电模式, 当打开该模式时, WH-NB75 把数据传输给服务器之后不久将自动切换至省电状态, 导致服务器无法再给 WH-NB75 传输内容, 需要在切换至省电状态之前进行传输才有效。当禁用该模式时, 支持任一时刻双向传输, 因此根据需要选择是否打开 PSM 模式。

云交换机

订阅 & 取消订阅 (订阅后收到设备推送消息)

操作

● 单个设备

设备ID :

(多个设备用英文格式的","分隔)

● 账号下全部设备

订阅

取消订阅

订阅成功

● 单发 设备ID:

● 群发 ☒ 16进制

01 03 00 63 00 03 F5 D5

发送

已订阅设备

• 356566071234561

图 4.11 订阅单个设备

4.5 农作物生长模型建立

将常见温室种植农作物的生长过程中的适宜环境参数加载到软件中，供生产人员参考，降低了对生产人员的工作难度。以西红柿为例说明农作物生长模型的建立过程。

西红柿的生长过程对温度要求比较严格，不宜过高和过低，为喜温种植物类。如果西红柿的生长环境出现光线比较暗、温度比较高、土壤太湿等情况会造成幼苗徒长。采取起垄的措施后，可以保证土壤湿度适中，但是另外还需要保证温度适宜，以及充足的光线。

成苗过程中，为了提高出苗率，需要保证环境温度介于 25 至 30℃之间，幼苗七成高度时需要移走保温地膜以达到通风的作用，此时床温适合介于 20 至 25℃之间，如果出现了幼苗的子叶，此时床温适合介于 15 至 20℃之间，如果出现了一至两片真叶，此时白天的床温适合介于 25 至 30℃之间，晚上介于 15 至 20℃之间。在分苗完成之后，需要在苗圃上方安装拱棚架和塑料膜，保证床温介于 30 至 35℃之间，地温保证高于 20℃。如果正值冬季，缓苗对温度要求较高，要注意保温，此时需要保证白天棚温介于 28℃至 30℃之间，晚上棚温介于 17℃至 20℃之间，地温保证高于 20℃。渡过缓苗期，对温度要求稍减，可以使白天棚温介于 22℃至 26℃之间，晚上 15℃至 18℃之间。雨雪场景下，注意光照强度，必要时可采取补光措施。此时确保白天棚温介于 20℃至 30℃之间，晚上介于 13℃至 15℃之间，禁止小于 8℃。天气晴朗的时候，中午气温稍高，适宜打开棚膜。春末夏初，太阳光照时长变长，我们适宜采取措施提高苗圃受光时长。需要注意的是，

55

中国知网 <https://www.cnki.net>

保持棚膜透光度，即勤打扫，另外适当增加打开棚膜的次数和时长，保证空气流通。天气晴朗的时候，上午棚温适宜介于 25℃至 28℃之间，下午棚温适宜介于 25℃至 20℃之间，前半夜棚温适宜介于 15℃至 20℃之间，后半夜棚温适宜介于 13℃至 15℃之间。雨雪场景下，白天棚温适宜介于 25℃至 30℃之间，晚上棚温适宜介于 10℃至 15℃之间。

开花过程中，在此期间需要注意避免低温、保暖、提高缓苗的速度。完成定植三至四日，不宜打开棚膜，这样可以提高温度提高缓苗的效率。在西红柿开花、结实的时候，要适当使棚温稍低，白天棚温适宜介于 20 至 25℃之间，晚上棚温适宜介于 13 至 15℃之间。湿度最好在百分之五十左右，地温恒定 15℃为宜。逐渐增加通风频率和时长；在西红柿开花期间禁止出现超过 30℃的情况，即使短时间超温也不允许，主要是取决于西红柿花粉发芽的环境条件。

成果过程中，在此期间要求棚温持续比较高，而且非常敏感。如果出现高于 10 摄氏度日积温小于 400℃的情况，会导致减产的后果。全天地温需要保持在 20 至 23℃之间，禁止小于 13℃。比较科学的操作是，上午适当减少打开棚膜的时间，可以确保棚温介于 25 至 28℃之间；中午打开棚膜，空气得以流通，棚温可介于 20 至 25℃之间，下午三点以后关闭棚膜进行保温；夜晚五到八点，使棚温介于 14 至 17℃之间；夜晚其余时间，使棚温恒定在 6 至 7℃。只有这样进行科学的种植，才可以使西红柿果实饱满增收。

成熟过程中，室内湿度恒定于百分之五十左右，否则会导致病虫害现象的出现，另外在灌溉、雨雪天气过后，可以采取打开棚膜的措施使湿度变小。如果晚上气温高于 15℃，可以选择全天打开棚膜使空气流通，否则高温会导致西红柿上色困难。

第 5 章 系统测试结果与分析

系统测试可以提高设计的有效性。首先按照原理图焊接好采集节点和通信控制器电路板，焊接完成后，检查有无漏焊、连焊、虚焊，有无元件极性装反等情况，检查无误后，用万用表电阻档测量电路板电源两端，若电阻过小，说明电路中存在短路情况，需重新检查。然后测试组网是否成功，进行了 ZigBee 数据包抓包实验分析，提高数据传输的准确性、稳定性是至关重要的。另外，对无线通信的各个环节，包括采集节点、通信控制器、云服务器端，可以利用串口调试助手进行数据通信监控，以测试收发数据的准确性。

5.1 无线数据通信测试

首先对 ZigBee 模块进行通道、模式和 PIN ID 设置，配置 1 个协调器（Coordinator）和 3 个终端（Router）Zigbee 模块。协调器安装在通信控制器电路板上，每个温室采集节点电路板放置 1 个终端，系统利用 3 个温室进行测试。节点的 LED 灯快速闪烁说明正在寻求入网且正常工作，当 LED 常亮时说明组网成功，如图 5.1 所示。

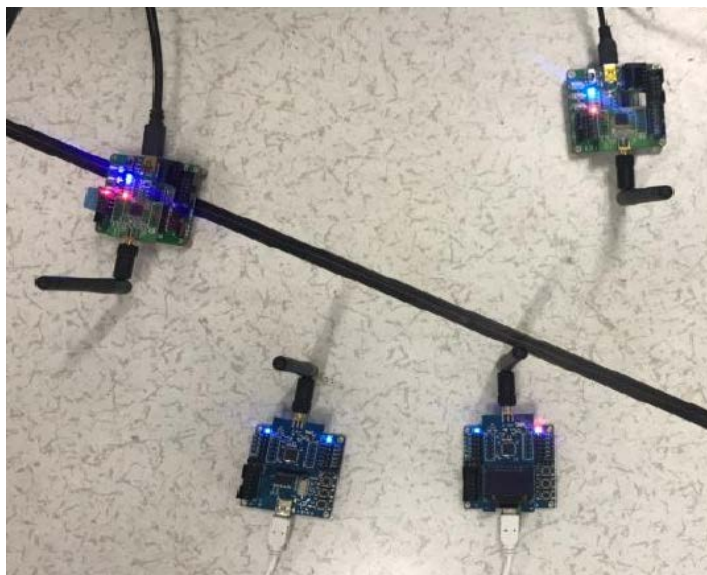


图 5.1 节点组网示意图

5.2 系统功能测试

焊接完成的采集节点和通信控制器电路板如图 5.2 所示，检测无误后接入传感器模块，进行通电测试。

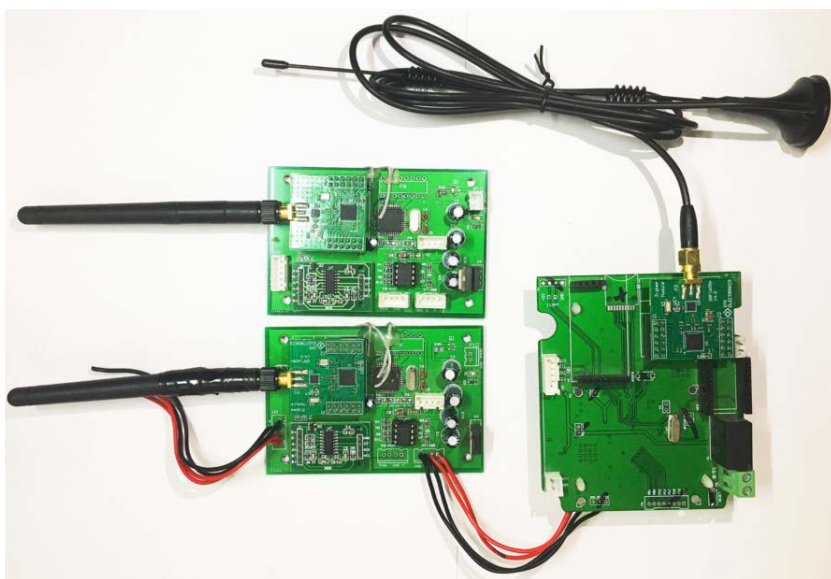


图 5.2 采集节点和通信控制器电路板

在有人透传云二次开发 SDK Demo 的基础上修改程序，开发符合温室环境监测系统通信协议的终端 APP。在终端 APP 登录界面输入账号和密码，验证正确后进入检测界面，如果验证失败会要求重新输入。登录成功后的检测界面如图 5.3 所示。界面中实时显示温室 1 的温度、湿度、光照强度和二氧化碳浓度值，以及所有被检测温室的报警信息。点击切换账号登录可以更换登录信息。点击关闭程序退出。点击温室选择下拉菜单，可以选择需要查看的温室编号，如图 5.4 所示，点击进入该温室监测界面。点击设置阈值，可以设置被检测参数的报警上下限，如图 5.5 所示，不需要设置阈值的位置空缺即可，点击返回主界面可返回监测界面。



图 5.3 系统主界面



图 5.4 选择温室界面

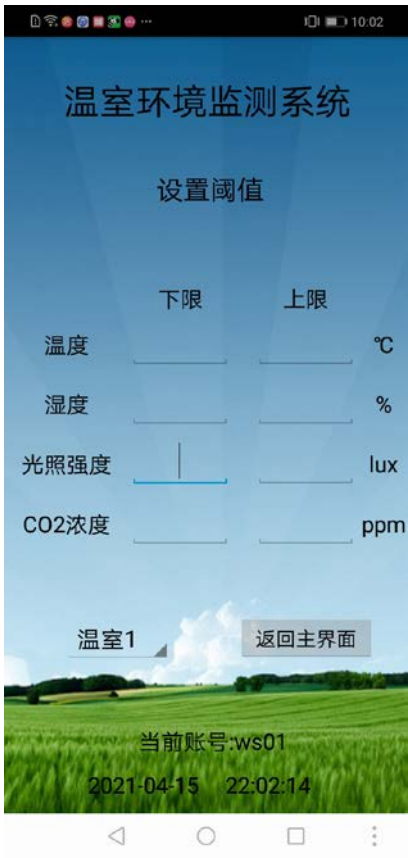


图 5.5 设置阈值界面

第6章 总结与展望

本篇论文针对温室农作物种植过程中遇到的实际困难提出了解决方案,设计并制作了一种基于 STM32 的温室环境监测和控制系统,达到了预期的设计要求。当前种植工作人员对温室环境参数的重视程度逐渐提高,温室内的环境状态直接影响着种植物的生长周期长短、质量高低、产量大小,本课题的研究成果可以使种植户实时监测并根据生长模型控制温室环境,进而控制农作物的生长过程。总的来说,主要研究成果如下:

- (1) 根据系统的功能情况,制定了系统中各模块间以及对外的通信协议,在系统应用过程中未出现数据传输错误的情况,说明了该通信协议的可靠性。
- (2) 对采集节点模块和通信控制器模块的硬件、程序进行了设计。
- (3) 实现了温室大棚环境因子的远程无线实时采集、显示,当某个参数值超过设定阈值时,系统可以利用 NB-IoT 网络提醒种植户。
- (4) 完成了终端软件的设计,经多次应用测试说明软件的稳定性、易操作性,而且响应迅速。

该系统经过简单修改还可以用于以下情景:

- (1) 房屋保温、保湿性能评价。
- (2) 微机室、档案管理部门、历史文物储藏室、医院等场合的环境监测管理。
- (3) 仓库的温度、湿度监测管理。
- (4) 蘑菇栽培的温度、湿度监测管理。
- (5) 孵化室温度、湿度监测管理。

虽然本论文达到了预期的系统各项功能指标,但研究过程中仍存在研究时间较短、水平有限等问题,因此该系统仍有一些缺点,离真正的实际生产应用和商品化还有一段距离。今后还将会对该系统进行更深入的研究,计划作如下改进:

- (1) 多个温室同时采集时,大量传感器节点进行组网,需要设置每个温室采集节点地址,进一步研究设计出适合用户的地址设置方法,使设置过程更简便。
- (2) 设计实现标准的温室执行机构,可以使种植户对大棚内的环境因子进行精准的控制。
- (3) 增加农作物生长模型种类,在界面中增加开始种植功能,结合季节等因素对农作物种植过程中的适宜环境参数进行实时提示,使用户更便捷的获取科学的农作物生长环境参数。
- (4) 使产品通过行业认证,尽快实现商品化,实现该系统的社会效益。

参考文献

- [1] 王冉冉.针对西红柿生长特性的物联网集成控制的温室智能通风系统的研究与设计[J].智能计算机与应用, 2015,3:72-76.
- [2] 韩成春.日光温室设施果品生产智能监控信息化技术研究[J].福建电脑, 2016, 3:15-16.
- [3] 贺文通.自动化技术在电子信息工程设计中的应用探讨[J].中国战略新兴产业, 2017, 38:104-105.
- [4] 龚志远.基于单片机与 PLC 协作的大棚智能监控[J].制造业自动化, 2019, 5:5-8.
- [5] 张鹏.基于无线网络的粮仓环境监测系统[J].商场现代化, 2013, 28: 13-15.
- [6] 肖天成.无线通信技术在远程测量系统中的应用[D].华中科技大学, 2012.
- [7] 欧亚军.基于 ZigBee 技术的智能家居系统设计与实现[D].湖南大学, 2013.
- [8] 金帅.基于 ZigBee 的无线自组网研究[D].南昌航空大学, 2011.
- [9] 雷晓明.基于物联网的无线消防系统的研究与应用[D].华北电力大学, 2013.
- [10] 谭玉波.基于短消息通信的车辆监控系统终端[J].电子产品世界, 2013, 12: 10-12.
- [11] 王小强.ZigBee 无线传感器网络设计与实现[M].北京:化学工业出版社, 2012:76-80.
- [12] 龚志猛, 牟杰, 向锋, 等.基于无线传感网络的温室远程监控系统[J].电子世界, 2014, 6:74-75.
- [13] 王良帆.基于物联网的温室大棚数字化管理系统设计与应用[D].合肥工业大学, 2017.
- [14] 邵宏信.基于 NB-IOT 的智能卷烟在途监管系统[J].科技风, 2020,5:12-15.
- [15] 赵丙辰.基于 ZigBee 的智能交通控制系统研究[J].电脑知识与技术, 2008,10: 141-142.
- [16] 周权.基于 Zigbee 的无线传感器网络设计[J].中国科技信息, 2014,11:117-118.
- [17] 吕治安.ZigBee 网络原理与应用开发[M].北京:北京航空航天大学出版社, 2008:71-73.
- [18] 朱彪凯.无线网状节点的电子数据取证应用研究[J].山西警察学院学报, 2020, 1:115-119.
- [19] 王成林.电信运营商的创新转型—物联网接入技术 NB-IoT[J].信息通信, 2017,5:221-223.
- [20] 吴坚.基于 NB-IoT 的实时被动式井盖监测系统[J].浙江科技学院学报, 2018,1:26-31.
- [21] 段恩传.窄带物联网 NB-IoT 技术及应用[J].数字化用户, 2019,16:18.

- [22] 欧阳龙.简析 NB-IOT 关键技术特性及行业应用[J].通讯世界, 2017,18:1-2.
- [23] 李鹏飞.窄带物联网技术要点研究[J].通讯世界, 2017,5:60-61.
- [24] 钟锡昌.嵌入式软件与 Hopen 系统[M].北京:北京航空航天大学出版社, 2004:62-65.
- [25] 冯祥胜.大学计算机基础[M].北京:电子工业出版社, 2007:43-45.
- [26] 蒲宝明.一种实时硬操作系统的设计与实现[J].微计算机信息, 2009,29:95-96.
- [27] 蔡长安.4 种嵌入式实时操作系统的两种主要技术分析和选择[J].重庆工商大学学报, 2007,2:161-166.
- [28] 辛改芳.基于 CAN 总线的车辆电子驻车制动系统的设计[J].仪表技术与传感器, 2013,3:44-46.
- [29] 马淑华.单片机原理与接口技术[M].北京:北京邮电大学出版社, 2007:35-37.
- [30] 李文仲.CC1110/CC2510 无线单片机和无线自组织网络入门与实战[M].北京:北京航空航天大学出版社, 2008:115-116.
- [31] 聂磊.基于 HART 协议的智能仪表通信系统设计[D].电子科技大学,2008.
- [32] 周鹏.基于 uC/OS-II_STC12C5A60S2 单片机的测周期系统设计[J].电子世界, 2013,15:128-129.
- [33] 李巍.基于 STC 系列单片机的串行通信[J].现代制造技术与装备, 2007,6:60-61.
- [34] 陈迪.智能循迹避障灭火小车[J].数码世界, 2016,12:224.
- [35] 刘小芳.单片机的多串口扩展技术的设计[J].计算机测量与控制, 2004,11:1088-1090.
- [36] 汤牛明.基于 CC2430 的城市车辆限行系统研究[J].现代电子技术, 2011,7:205-210.
- [37] 何鸿.基于无线 HART 低功耗网络设计与实现[J].计算机测量与控制, 2010,11:2605.
- [38] 王振东.基于 BitCloud 的室内定位系统[J].计算机系统应用, 2014,6:74-78.
- [39] 王宜君.基于 Zigbee 的大棚环境参数监控系统的设计[J].通讯世界, 2016,1:195.
- [40] Rumelhart D E. Learning representations by back-propagating errors [J].Nature, 2012, 323(9):533-536.
- [41] R.S.Kawitkar, P.K.Shevgaonkar.Design of Smart AntennaTestbed Prototype [J], IEEE, 2013:299-301.
- [42] LI HAO.Wireless Network Technology Based on Zigbee and Its Application[J].Information Technology, 2008 , 1:12-14.
- [43] REFERENCES Wheeler Andrew. Commercial applications of wireless sensor Network susing Zigbee[J]. IEEE Communications Magazine, 2007:70-77.
- [44] IF AKYILDIZ, SUW, SANKA RASUBRA MANIAMY,etal.Wireless sensor

- network:a survey [J]. Computer Networks,2002, 38(4):393-422.
- [45] Pottie GJ, Kaiser WJ. Wireless intergrated network sensors[J]. Cornrnunication of the ACM 2000, 43(5):51-58.
- [46] Liang J, Lee D H, Xu G, et al. Analysis of Passive Boost Power Converter for Three-Phase SR Drive[J]. IEEE Transactions on Industrial Electronics, 2010, 57(9): 2961-2971.
- [47] Zhi Yang, Fei Shang, Ian P. Brown, et al. Comparative Study of Interior Permanent Magnet Induction and Switched Reluctance Motor Drives for EV and HEV Applications[J]. IEEE Transactions on Transportation Electrification, 2015, 1(1): 245-254.
- [48] R Abdollahi. Harmonic Mitigation Using 36-Pulse AC-DC Converter for Direct Torque Controlled Induction Motor Drives[J]. Journal of Applied Research and Technology, 2015, 13(1): 135-144.
- [49] Takeno M, Chiba A, Hoshi N, et al. Test Results and Torque Improvement of the 50kW Switched Reluctance Motor Designed for Hybrid Electric Vehicles. IEEE Transactions on Industry Applications, 2012, 48(4): 1327-1334.
- [50] Song S, Xia Z, Zhang Z, et al. Control Performance Analysis and Improvement of a Modular Power Converter for 3-phase SRM with Y-Connected Windings and Neutral Line. IEEE Transactions on Industrial Electronics, 2016, 63(10): 6020-6030.

致 谢

本论文是在导师孙宝江教授的悉心指导下完成的。从拟定题目到定稿，历时数月。在本论文完成之际，首先要向我的导师致以诚挚的谢意。在完成设计和论文的写作过程中，老师给了我许许多多的帮助和关怀，老师学识渊博、治学严谨，平易近人，他不仅教我专业知识和技能，还教我很多为人处事的道理。同时他对工作的积极热情、认真负责、有条不紊、实事求是的态度，给我留下了深刻的印象，使我受益非浅。在此我谨向指导老师表示衷心的感谢和深深的敬意。

同时，我要感谢给我们授课的各位老师，正是由于他们的传道、授业、解惑，让我学到了专业知识，并从他们身上学到了求知治学的方法和为人处事的智慧。我还要感谢我的母校，是她提供了良好的学习环境和生活环境，让我的在职研究生生活丰富多彩，为我的人生留下精彩的一笔。

最后，衷心感谢我的队友们，在我毕业论文写作中，与他们的探讨交流使我受益颇多；同时，他们也给了我很多无私的帮助和支持，我在此深表谢意。

在学期间主要科研成果

一、发表学术论文

- [1] 院冬珠, 郭磊.基于单片机的高空作业防坠落安全辅助设备的设计[J].科学与财富, 2018, 10 (19): 110
- [2] 郭磊.基于 STM32 的温室环境监测和控制系统[J].科学与生活, 2021, 10 (已录用未发表)

二、其他科研成果

实用新型专利, 公开号 CN207307009U (本人第一作者);
实用新型专利, 公开号 CN207410758U (本人第一作者);
实用新型专利, 公开号 CN207305753U (本人第一作者);
实用新型专利, 公开号 CN207695260U (本人第二作者);
实用新型专利, 公开号 CN207837424U (本人第二作者)。