

**本科生毕业论文（设计）**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 题目 | : | 基于NB-IOT技术的 | | | |
|  |  | 大型医疗设备环境监测系统 | | | |
| 姓名 | : | 陆正大 | | | |
| 学院 | : | 生物医学工程与信息学院 | | | |
| 专业 | : | 生物医学工程 | | | |
| 班级学号 | : | 15060206 | | | |
| 指导教师 | : | 汪缨 | 职称 | : | 工程师 |
| 指导教师 | : | 许翔 | 职称 | : | 工程师 |
| 2019 年 6 月 19 日  南京医科大学教务处制 | | | | | |

# 摘 要

随着医院的的发展和建设，越来越多的大型医疗设备如磁共振成像MRI、多排螺旋CT、DSA心血管机、CR、DR激光相机、PET-CT正电子发射计算机断层显像等投入医院使用，而由于现代化医疗设备大部分是由微机及智能电子电路控制，辅以高压电路，所以对环境中温度、湿度、粉尘颗粒含量等因素的影响较为敏感，而我院所处南京市，为典型的亚热带季风气候，空气湿润，严重干扰了医疗设备的正常运行。所以如何保障医疗设备在安全坏境中运行，应该成为医疗设备管理工作一个重点。

通过调查，目前太多数医院采取的是派遣工程人员早晚定时巡查机房设备状况，这不仅加重了管理人员和工程人员的负担，而且还不能及时的把故障排查，而市场上存售的关于温湿度检测系统，利用WiFi作为传输方式，不仅信号时常受外界坏境干扰，而且造价昂贵，本人利用所学，依托物联网和云平台技术，设计出一款能够长距离、低功耗、实时传输的温湿度检测系统。

本设计Arduino Nano采用作为主控芯片，选用DHT22作为温湿度传感器采集数据，采用基于NB-IOT技术设计的开发板ME3616作为通讯模块，首先将温湿度数据通过OLED12864实时呈现，同时将温湿度数据实时上传到中国移动云平台ONENET，通过云端监测，从而实现对机房温湿度的实时性监测。发生故障时，工程师可以通过分析历史温湿度数据，及时了解机房状态，加快出工速度，降低使用人员风险，减少设备故障率，缩短故障排查时间，并且有利于工程师有针对性进行预防性维护，提高工作效率。

**关键词**：Arduino Nano，DHT22，NB-IOT，ONENET

# Abstract

With the development and construction of hospitals, more and more large medical equipments such as magnetic resonance imaging MRI, multi-slice spiral CT, DSA cardiovascular machine, CR, DR laser camera, PET-CT positron emission computed tomography are put into use in hospitals. Because most of modern medical equipments are controlled by computers and intelligent electronic circuits, assisted by high voltage circuits, the temperature and humidity in the environment are high. The influence of degree, dust particle content and other factors is more sensitive, while Nanjing, where our hospital is located, is a typical subtropical monsoon climate with humid air, which seriously interferes with the normal operation of medical equipment. Therefore, how to ensure the operation of medical equipment in a safe environment should become a key point in the management of medical equipment.

Through investigation, at present, too many hospitals are dispatching engineers to inspect the equipment condition of the computer room in time, which not only increases the burden of managers and engineers, but also fails to detect faults in time. The temperature and humidity detection system on the market, using WiFi as a transmission mode, not only often interferes with the external environment, but also costs a lot. I use what I have learned, relying on the Internet of Things and cloud platform technology, to design a temperature and humidity detection system that can be long-distance, low power consumption, real-time transmission.

This design uses Arduino Nano as the main control chip, DHT22 as the temperature and humidity sensor to collect data, and ME3616 as the communication module based on NB-IOT technology. Firstly, temperature and humidity data are presented in real time through OLED12864. At the same time, temperature and humidity data are uploaded to ONENET, China Mobile Cloud Platform. Through cloud monitoring, temperature and humidity in computer room can be real-time monitored. Temporal monitoring. When a fault occurs, engineers can analyze the historical temperature and humidity data, timely understand the status of the computer room, speed up the work, reduce the risk of users, reduce the rate of equipment failure, shorten the time of troubleshooting, and help engineers to carry out preventive maintenance and improve work efficiency.

**Key words:** Arduino Nano, DHT22, NB-IOT, ONENET

# 目 录

[摘 要 I](#_Toc10769)

[Abstract II](#_Toc14952)

[第 1 章 绪论 1](#_Toc32444)

[1.1课题研究背景与意义 1](#_Toc22759)

[1.2 NB-IOT技术国内外研究现状 2](#_Toc32608)

[1.3主要研究内容 3](#_Toc11705)

[1.4组织结构安排 3](#_Toc15178)

[第2章 系统设计方案 4](#_Toc3828)

[2.1系统模块的选择与论证 4](#_Toc3152)

[2.1.1控制器模块的选择与论证 4](#_Toc8926)

[2.1.2 显示模块的选择与论证 4](#_Toc3136)

[2.1.3通讯模块的选择与论证 5](#_Toc32295)

[2.1.4基于NB-IOT技术的大型医疗设备环境监测系统的可行性方案 5](#_Toc16058)

[2.2系统选择方案 6](#_Toc6363)

[2.3设计对象质控 7](#_Toc23262)

[第3章 系统硬件设计 8](#_Toc30671)

[3.1单片机的选择 8](#_Toc15229)

[3.1.1 STM32L031K6 8](#_Toc5792)

[3.1.2 Arduino Nano 11](#_Toc29093)

[3.2温湿度传感器DHT22 12](#_Toc15993)

[3.2.1引脚说明 13](#_Toc5317)

[3.2.2 DHT22测量温湿度说明 14](#_Toc6203)

[3.3 NB-IOT通讯模块ME3616 15](#_Toc1212)

[3.3.1 ME3616 NB-IOT特性 15](#_Toc31936)

[3.3.2 ME3616引脚功能 16](#_Toc13709)

[3.3.3 ME3616内部电路图 17](#_Toc23627)

[3.4 显示电路 17](#_Toc6546)

[3.4.1 OLED 12864基本特性 18](#_Toc7782)

[3.4.2 OLED 12864引脚图功能 18](#_Toc9826)

[3.4.3 OLED 12864原理图 19](#_Toc9656)

[第4章软件设计 20](#_Toc4549)

[4.1软件开发平台和开发语言的介绍 20](#_Toc16291)

[4.1.1 设计语言的选择 20](#_Toc3946)

[4.1.2 软件开发平台 21](#_Toc17825)

[4.2 ONENET云平台 23](#_Toc23875)

[4.2.1 设备接入 23](#_Toc9916)

[4.2.2触发器设置 23](#_Toc7078)

[4.2.3传输协议 24](#_Toc406)

[4.3数据采集 25](#_Toc6063)

[4.4 DHT22温度传感器程序设计 27](#_Toc31704)

[4.5 显示电路子程序 28](#_Toc8564)

[第5章 软硬件调试 29](#_Toc10760)

[5.1通讯模块调试 29](#_Toc5382)

[5.2 控制模块调试 29](#_Toc13072)

[第6章实验结果与分析 30](#_Toc13400)

[6.1实验测试结果展示 30](#_Toc29949)

[6.1.1实验测试结果 30](#_Toc8578)

[6.1.2误差分析 31](#_Toc23749)

[6.2实验总结和展望 31](#_Toc1019)

[综 述 32](#_Toc28936)

[致 谢 43](#_Toc30910)

[附录Ⅰ 36](#_Toc20534)

[附录Ⅱ 38](#_Toc11380)

[参考文献 44](#_Toc9337)

# 

# 第一章 绪论

**1.1课题研究背景与意义**

随着医疗科技的高速发展，越来越多的大型医疗设备如磁共振成像MRI、多排螺旋CT、DSA心血管机、CR、DR激光相机、PET-CT正电子发射计算机断层显像等投入医院使用，而由于现代化医疗设备大部分是由微机及智能电子电路控制，所以对环境中湿度、温度、粉尘颗粒等因素的影响较为敏感，而我院所处南京市，为典型的亚热带季风气候，空气湿润，严重干扰了医疗设备的正常运行。

CT是大型精密仪器, 因此对其所在环境的温湿度有很高的要求, 需要对扫描间与操作间的温湿度进行监测[[[1]](#endnote-1)]。在工作中过程中时常会遇到CT检查扫描不能进行，控制台不断报警，指示湿度过高，经现场检查发现尘埃废絮堵塞了水风冷机进风口的金属滤网，热量不能及时从机架内散发出去造成多排螺旋CT出现机架内温度过高触发保护装置停机，通过清理，使得机架内温度降低，报警解除[[[2]](#endnote-2)]。又如直线加速器启动时数次自检不能通过，等待一段时间后，自检可以通过，但是当机器次日重新启动时，类似故障再次出现，观察对比发现设备间湿度较高[[[3]](#endnote-3)]，而加速器驱动电路和控制采样结构中包含运放器，运放器属于微电子结构，并且运放电路输入阻抗较高，当空气中湿度不符合设备的规范时，极易造成零漂增大，使得采样信号不稳定，导致计算机自检不能通过，出现漂移等故障；还有磁共振的许多物理参数受环境变化和时间的影响同样会发生漂移,

例如磁体间的温度变化导致低场磁共振的中心频率发生漂移,磁场的均匀性随时间慢慢变差，使得图像质量下降，导致压脂现象以及磁共振波谱的质量下降[[[4]](#endnote-4)]。特别当MR制冷系统失常时，会使空气湿度达到饱和状态，导致在高压电缆表面冷凝成水，并伴随时间的延长而增多，滴落高压部件极易引起短路，造成失超，导致不可挽回的损失，所以为了保障确切的病灶图像质量，帮助医生做出准确的判断，维护病人和医院的权益，对大型医疗设备的使用坏境进行监测就显得尤为重要。

而传统的方法是利用具有只有温度和湿度的检测仪表，依靠人工对机房温度和湿度进行检测和控制，通过这种方法检测出来的温度和相对湿度误差会很大，而且具有时延性，导致不能维持坏境参数在设备的规范之中，当因温湿度失常导致设备故障时，我们的工程人员将无法及时赶赴现场处理问题，也无从得知是否为温湿度失衡而导致的机器故障，需要通过排查才能确定故障来源，工作效率无法得到提升。

基于上述情况，通过研究调查，我们选择了利用NB-IOT技术设计用于大型医疗设备环境的监测系统，实现对医疗设备机房坏境温湿度的远程实时监控，该系统具有历史查询、趋势分析、和报警通知的功能，帮助工程师发现安全隐患，同时为工程师提供了历史数据作为参考信息加快判断故障来源，提高工作效率。

**1.2 NB-IOT技术国内外研究现状**

物联网是指“物-物相连的互联网”,是通过红外感应器、温湿度传感器全球定位系统GPS射频识别等信息传感设备,按约定的协议,包括MQTT、EDP、HTTP，把指定物品与互联网相连,进行信息交换和通信,以实现智能化识别、定位、跟踪、监控和和管理的一种物与物相连的网络，据预估，到2020年, 预计将有数十亿的服务和设备开始走入现实[[[5]](#endnote-5)]，就目前而言，已经通过物联网方案解决了一些城市问题，如智能抄表，城市摄像头…

作为新一代的技术宠儿，多个国家将物联网发展作为推动经济发展的一项国政方针。其中作为世界发展大国的美国，早在二十一世纪初便将物联网囊括在五大国防领域之一，并推动位于美国的众多世界学术高校大量投入物联网的研究；还有作为世界第一大经济体的欧盟，甚至早于美国提出要建立无物不通的网络社会，于2009年为此投入九亿美元研究，且逐年递增；作为一个善于学习的国家，日本政府在U-Japan提出将传感器技术作为日本发展四项重点之一，并推崇要让物联网成为国家发展战略之一。当然中国政府也高度重视物联网的发展，在物联网概念萌芽之时，便迅速将其立为国家发展战略，并得到显著成果，部分传感器网络已达到世界领先水平[[[6]](#endnote-6)]。

由于物联网的数据传输技术种类多，如何选择传输技术便成为一项核心问题，通过物联网获取的数据具有实时动态、大量散乱的特点，而远距离数据传输也是一个巨大挑战。目前而言，长距离传输技术主要包括蜂窝网络技术与低功耗广域网技术，而蜂窝通信进行广域数据传输存在功耗大、成本高、网络速率与应用需求不匹配等问题，据GSMA相关统计可知，为了匹配应用场景，到2020年60%左右的物联网应用对网络连接的要求将是低速率，不在是目前的高速率要求。经过多年的应用，比如智能抄表、智能垃圾桶的提出，我们发现广域网技术适配低速率场景，发展前景更加广阔。目前经授权频段的广域网技术主要包括窄带物联网（NB-IOT）与LTE（eMTC）演进技术，目前eMTC主要用于语音场景，受到一定的限制，所以NB-IOT成为了我们的首选，NB-IOT技术具有四大优势：覆盖广、容量大、成本低、功耗低，适应了物联网应用场景的数据特点。

**1.3主要研究内容**

本设计主要采用NB-IOT技术作为数据传输方式，采用Arduino Nano作为主控芯片，利用的DHT22温湿度传感器采取温湿度数据，在通过SSD1306点阵式显示系统呈现数据的同时，利用基于NB-IOT技术构成的通讯模块ME3616上传数据至中国移动云平台ONENET，根据事先设定好所要监控的数据流，根据触发条件判断是否报警，并且平台将记录实时数据，从而完成基于NB-IOT技术的医疗设备坏境监测。

**1.4组织结构安排**

本文组织结构上共分为七章，主要内容安排如下：

第1章 绪论：阐明本课题提出的研究背景、研究意义、NB-IOT技术国内外研究 现状及主要研究内容

第2章 系统设计方案：根据物联网数据特点综合论证系统模块的选择，提出基 于NB-IOT技术的大型医疗设备环境监测系统的可行性方案，依据大型医 疗设备监控系统硬件规范进行分析

第3章 系统硬件设计：对本次设计涉及的硬件，根据其特点、构成原理等进行 分析

第4章 软件设计：介绍本次课题所采用的开发坏境、设计语言，阐述了如何应 用计算机软 件技术和物联网技术对医疗设备监控系统进行软件开发和 系统实现

第5章 调试：对课题中软硬件进行功能调试，在设计过程中排除软硬件功能损 坏等故障

第6章 实验结果与分析：展示系统实验结果，总结系统的优势与缺点，并对下 一步工作进行展望

# 第2章 系统设计方案

**2.1系统模块的选择与论证**

### 2.1.1控制器模块的选择与论证

本系统中主控制器主要为控制DHT22采集数据，同时发出指令控制ME3616上传数据到ONENET云平台。

方案1：51单片机作为控制芯片。51是应用最广泛的8位单片机之一，其具有经典硬件和软件结构，为其之后的单片机发展奠定了基础，但其对基础要求较高，需要综合学习单片机与接口技术才能熟练使用，51学习的最大亮点是其实践性，一般用于对性能要求不高和教学场所。

方案2：Motorola作为控制芯片，Motorola资源少且技术一般，对于小客户而言，基本不支持，但是Motorola拥有强抗干扰力、功能类型多和不需向外扩展，适合大型企业构建内部网络。

方案3：Arduino Nano作为控制芯片。是一款基于Arduino Nano 3.0的小型完整可直插面包板的开发板，具有高集成度、高性能、低成本、外设一流，功耗低且速度快，开发环境简单，最大亮点是，汇编语言通俗易懂，寻址方式多，指令少。适用于低功耗及超低功耗的场合。

方案比较与论证：方案1的51单片机性能较低，而方案2的Motorola虽性能高，但是就体积而言，就面对用户方向，方案3的更占优势，但其价格相对更低，性价比最高，故选择方案3

### 2.1.2 显示模块的选择与论证

本系统中显示模块用于显示DHT22传感器采集温湿度，通过单片机输出的 结果，以便进行判定结果的观察、硬件的调整以及软件的调试。

方案1：LCD12864显示屏作为显示模块。LCD12864是一种点阵图形液晶显示屏，具有灵活的接口方式，简单、方便的操作指令和大尺寸量产的支持，含有简体中文，功能齐全，但是存在响应速度较慢、可视角度小、工艺复杂和连线较为复杂的缺点。

方案2：OLED显示屏作为显示模块。OLED显示屏是利用有机发光二极管制成的显示屏，具有自发光、广视角、极高反应速度、轻薄、高对比度、工艺简单和抗震性能好等优势，但是存在寿命远低于LCD和对大尺寸不支持的缺陷。

方案比较与论证：本装置的显示模块需要显示数量不多的数字和汉字，但是不需要具有高寿命和大尺寸等特性，由于OLED具有快速响应、轻薄、工艺简单和抗震性好等优点，同时连线较为简单，利于电路的检查。故选择方案2。

### 2.1.3通讯模块的选择与论证

WiFi基于无线路由器作为组网方式，传输速率高，但续航能力及传输距离较NB-IOT技术和LoRa略显弱势，而蓝牙仅能维持在10m范围的拥有传输信号，不符合本系统设计的远程监控的要求，LoRa技术目前应用于美国本土，中国大陆尚未布置相关节点，且其节点和网关部署要求极高，成本过高，而NB-IOT技术基于现有的蜂窝网络，能够融入当前的网络，无需重新布置节点和网关，不仅节约了成本，同时扩大了现有网络供应商的服务范围；综上各通讯方式，我们选择了基于 NB-IOT技术的ME3616，该模块具备NB-IOT通讯技术低成本、广覆

盖、续航久等优势。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | NB-IOT | Wifi | LoRa | 蓝牙 |
| 组网方式 | 基于现有蜂窝组网 | 基于无线路由器 | 基于lora网关 | 蓝牙mesh网关 |
| 网络部署方式 | 节点 | 节点+路由器 | 节点+网关（网关部署位置要求高） | 节点 |
| 续航能力 | 10年 | 数小时 | 10年 | 数天 |
| 传输距离 | 10km | 50m | 20km | 10m |
| 传输速度 | 160kpbs-250kpbs | 300Mbps | 0.3-50kpbs | 1M |
| 成本 | 2-10￥ | 7-8￥ | 约5￥ |  |
| 频段 | Lcense、运营商频段 | 2.4G或5G | unlcense频段  Sub-GHZ(433、868、915MHZ) | 2.4G |

### 2.1.4基于NB-IOT技术的大型医疗设备环境监测系统的可行性方案

综上所述：本设计Arduino Nano采用作为主控芯片，选用DHT22作为温湿度传感器采集数据，采用基于NB-IOT技术设计的开发板ME3616作为通讯模块，首先将温湿度数据通过OLED12864呈现；同时将温湿度数据实时上传到中国移动云平台ONENET，ONENET自带应用模块将自动识别采集数据，绘制折线图，并利用触发器模块，当检测到机房温湿度出现异常，触发阈值，系统触发报警功能，发送报警提示文件通知工程师及时赶往现场进行检修；此外当医疗设备故障发生时，工程师也可通过观察温湿度历史数据，筛查故障发生来源，提高工程师的工作效率，实现大型医疗设备环境的实时性检测，这种方式不仅经济实惠，而且安全可靠。

**2.2系统选择方案**

一、 数据采集：  
 由温湿度传感器DHT22采集环境或者设备的状态值，通过内部芯片进行数据处理，转换为使用于工业通讯的格式，同时内部会生成必要的数据校验，为控制器处理数据做准备。  
二、 MCU处理与发送以及接收：  
 Arduino平台设计程序，利用MCU主动问询或者接收由传感器采集的数据点，进行必要的校验无误后，进行数据的简单处理，换算，进而进行数据的封包，进行数据整合后封装成适合通讯模组的格式发送至通讯模组，模组接受到特定指令后，下发命令，当控制器收到通讯模组回传的下发命令，将进行解析并控制硬件执行相关的动作

三、 通讯模组数据推送  
 通讯模组ME3616接收控制器Arduino Nano封装好的数据，内部芯片利用支持的通讯协议LWM2M将数据发送至中国移动云平台Onenet，同时也接收服务器下发的命令，转发给控制器来执行命令。  
四、 服务器接收：  
 服务器接收到硬件设备发来的数据进行校验和解密后，将数据推送至队列或者缓存至数据库，用于OLED12864前端显示与用户界面的展示，设置触发器，接受到异常信息，发送预警信息；同时也接收用户发来的指令，将消息下发至MCU进而控制设备执行动作。

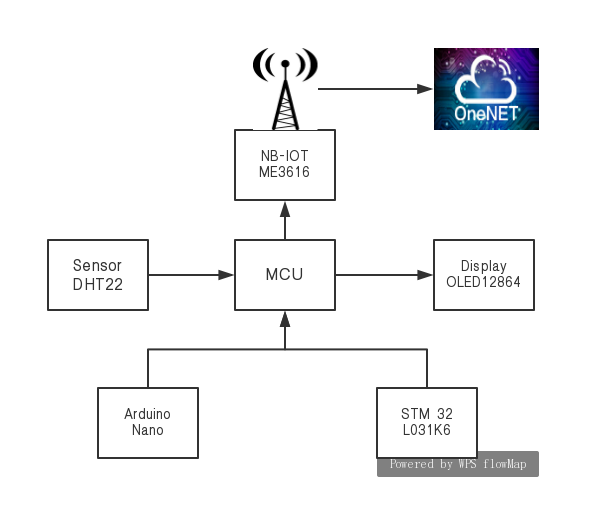


图 1硬件架构示意图

**2.3设计对象质控**

表 1SPECT/CT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 温度  （℃） | 温度变化率  （℃/h） | 湿度  （%） | 湿度变化率  （%/h） | 散热量 |
| 扫描间 | 18-26 | <=3 | 30-60 | <=5 | 9.2kw |
| 操作间 | 18-26 | <=3 | 30-60 | <=5 | 2.7kw |

表 2PET/MR

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 温度  （℃） | 温度变化率  （℃/h） | 湿度  （%） | 湿度变化率  （%/h） |
| 磁体间 | 15-21 | <=5 | 30-60 | <=5 |
| 控制间 | 15-21 | <=5 | 30-70 | <=5 |
| 设备间 | 15-21 | <=5 | 30-10 | <=5 |

# 第3章 系统硬件设计

**3.1单片机的选择**

### 3.1.1 STM32L031K6

STM32L031K6是采用高性能、超低功耗的32位ARM，系统采用精简指令集RISC，拥有高达32 KB的闪存程序存储器，1KB的数据EEPROM和8 KB的RAM以及广泛的增强型I/O和外设接口，工作频率为32 MHz，可提供高功率效率，适用于各种低功耗设备，且内部集成了ST-LINK调试器/编程器，不需要任何单独的探头，可以配合Keil软件使用；同时STM32L031K6通过大量外部和内部时钟源，为系统提供精准时钟信号，利用内部电压适配实现多种低功耗模式。内部还集成多个低功耗定时器和比较器，包括LPTIM、RTC 和基于时基的SysTick，由12位的ADC，同时STM32L031K6还设置了两个看门狗，一个窗口看门狗基于总线时钟，还有一个看门狗具有独立时钟和窗口功能。此外，本设计需要多个外接通信端口，而STM32L031K6拥有丰富的通信接口，包括SPI，I2C，USART和一个低功耗UART（LPUART），能够兼容LED、蜂鸣器和OLED12864等元器件，且满足符合NB-IOT低功耗、精确度高等要求。

电源电压：2.0-3.6 V。

工作温度范围：-40-+105°C。

晶振范围：4～32 MHz。

低功耗模式：睡眠、停机和待机。

复位方式：外部、掉电复位。

调试方式：串行线调试（SWD）。

1×12位，1.0us的ADC（最多10个通道）。

存储器：32 Kbytes片内Flash存储器；4 Kbytes带有硬件校验片内 SRAM存储器，20KB备份寄存器

通讯方式：1个USART支持主同步SPI和调制解调器控制1个SPI高达 18Mbit/s；1个I2C接口，支持快速模式加（1Mbit/s）

多达9计时器：一个低功耗定时器（LPTIM），三个通用16位定时器， 一 个RTC 和一个可以用作时基的SysTick。一个具有独立时钟和窗 口功能的看门狗，以及一个基于总线时钟的窗口看门狗。

内核基于ARM® 32-bit Cortex®-M0 CPU与STM32单片机超低功耗特性结合,主频高达48 MHz。

电流消耗参考值：

• 动态运行模式：低至87 μA/MHz

• 超低功耗模式 + 全RAM + 低功耗定时器：440 nA（16根唤醒线）

• 超低功耗模式 + 备份寄存器：250 nA（3个唤醒引脚）

• 唤醒时间：3.5 µs

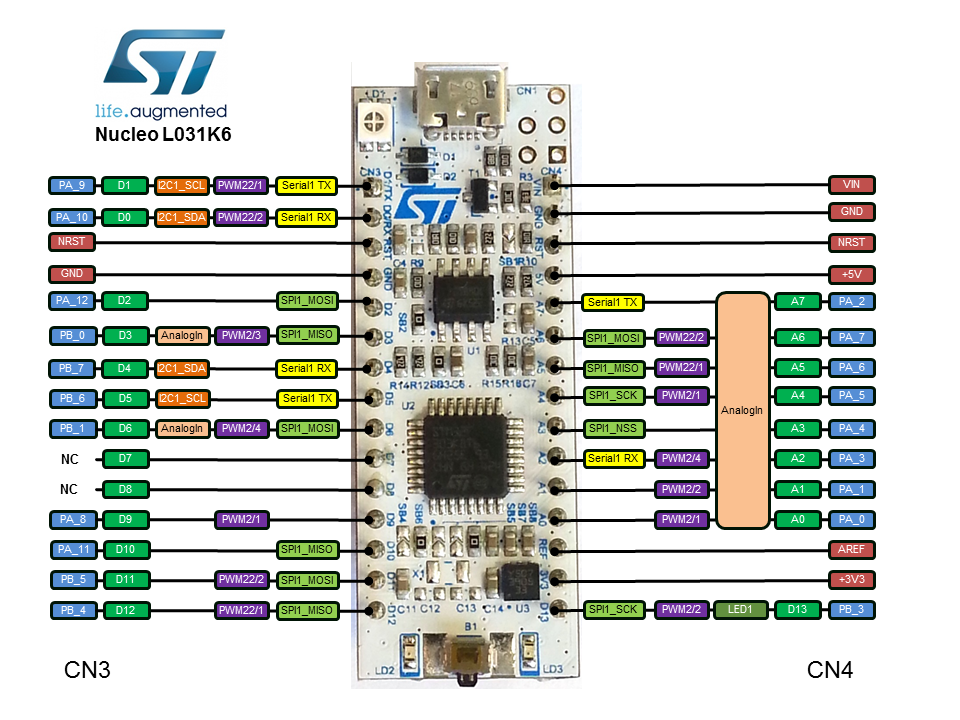


图 2 L032K6接口分布

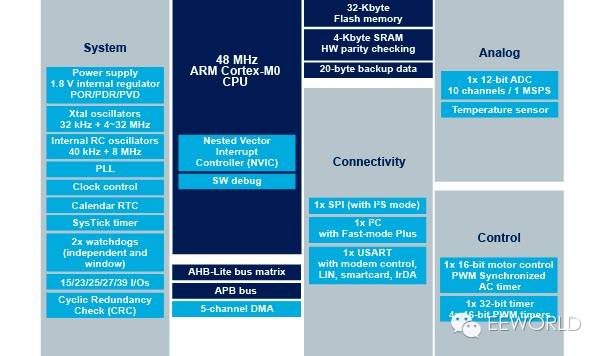


图 3STM32F031K6微控制器的内部结构原理

### 3.1.2 Arduino Nano

Arduino Nano开发板是当今市场上最流行的开发板之一（见下图），是一款基于Arduino Nano 3.0的小型完整可直插面包板的开发板，采用无电源插座和Mini-B USB接口，并基于Arduino开发板设计了多种引脚功能，其中ICSP在线编程接口，能够让用户通过编程电缆直连编程器，实现在线编程，使得开发调试方便进行；此外Arduino Nano包括14个数字输入/输出引脚，具有PWM功能多达六个；还有既可以作为模拟输入又可以作数字输入/输出的模拟引脚；同时Arduino Nano开发板上集成了16MHz晶体振荡器、复位按钮和ICSP header，功能齐全，让开发者操作过程中更加灵活。

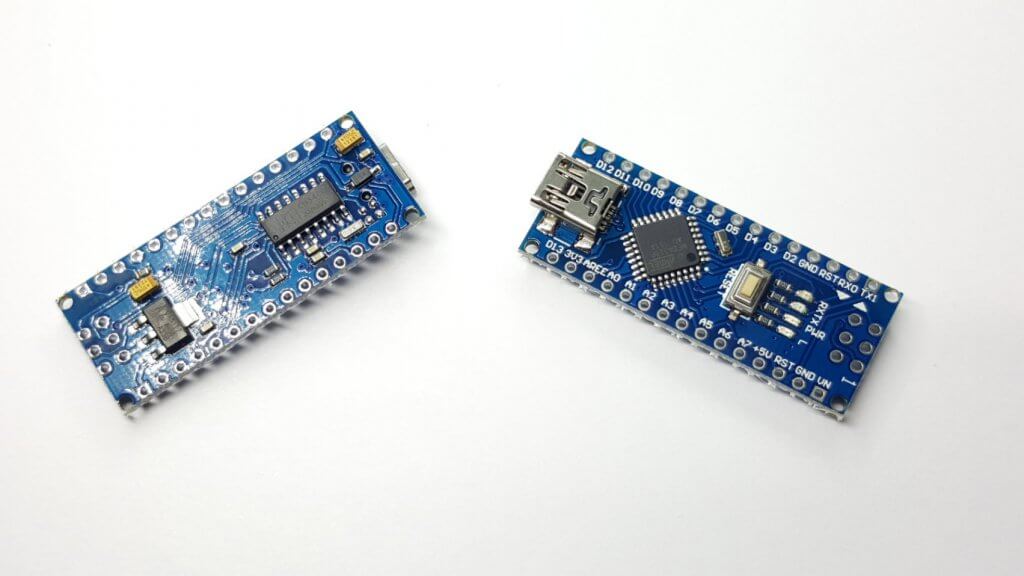


图 4 Arduino Nano

**技术参数：**

工作电压：5V

输入电压：6V～20V

3.3V口输出能力：50mA

I/O端口输出能力：20mA

时钟晶振：16MHz

供电方式：直流电源插孔、VIN引脚、USB电缆

通信协议：基于USB的虚拟串口、SPI接口、TWI(兼容I2C)接口、

引脚内容：14路数字IO脚、6路模拟输入脚

存储器：2KB的SRAM、1KB的EEPROM、32KB的FLASH

触发方式：低电平触发、电平改变触发、上升沿触发、下降沿触发

Arduino Nano接口定义图

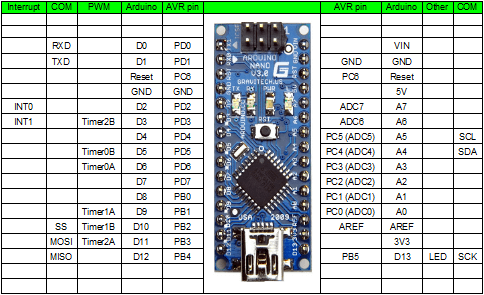
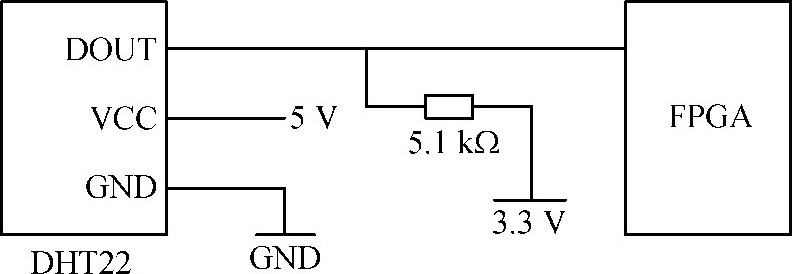
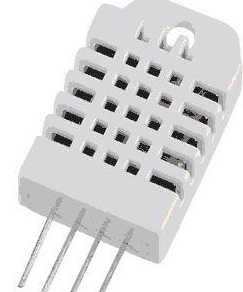


图 5 Arduino Nano

**3.2温湿度传感器DHT22**

DHT22（实物图下左）又名AM2302，4针单排引脚封装，由一个电容式感湿原件和NTC测温元件组成，该温湿度复合采集元件上综合了温湿度传感技术和数字模块采集技术的，可连接一个高性能的八位单片机，根据控制芯片的指令采集数据。该产品不仅在出厂之前在特定坏境中进行过精准较准之外，采集数据之前进行自检，利用出厂时便存储在OTP内存中的校正系数进行自我校准，这样不仅有效提高了采取数据的稳定性，更提高了所获得数据的准确性。该款产品不仅具有造价低、响应速率快、低功耗等优点，还具有强大的抗干扰能力，使得该产品适用于不同类型环境恶劣的场所[[[7]](#endnote-7)]。下图右为单片机与DHT22连接的引脚图。



**技术参数**

|  |  |  |
| --- | --- | --- |
|  | 湿度 | 温度 |
| 精度 | ±0.3%RH | ±0.5℃ |
| 迟滞 | ＜0.3%RH |  |
| 分辨率 | 0.1%RH（16bit输出） | 0.1%℃（16bit输出） |
| 可重复性 | ±0.3%RH | ±0.2℃ |
| 采样周期 | 2s |  |
| 响应时间 | 5s | ＜10s |
| 年稳定性 | ＜±0.5%RH |  |
| 互换性 | 完全互换 | 完全互换 |
| 漂移 | ＜0.5RH | ±0.3℃ |

**电气特性**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 参数 | 条件 | MIN | TYP | MAX | 单位 |
| 供电 | DC | 3.3 |  | 6 | V |
| 供电电流 | 测量 | 1 |  | 1.5 | mA |
| 待机 | 40 |  | 50 | uA |
| 采样周期 | 秒 |  | 2 |  | 次 |

### 3.2.1引脚说明

|  |  |  |
| --- | --- | --- |
| Pin | 名称 | 注释 |
| 1 | VDD | 电源3.3～5V |
| 2 | DATA | 单总线双向口，串行数据口 |
| 3 | NC | 空脚 |
| 4 | GND | 地 |

### 3.2.2 DHT22测量温湿度说明

DHT22温湿度传感器采用典型的单总线双向口，通过仅有的一根数据线DATA引脚与单片机进行控制、通讯和同步，单次通讯时间为4ms，完整的一次数据传输为40bit，5个字节，高位先出，可分为小数和整数两个部分，通过SDA数据总线向单片机依次送湿度高8bit、湿度低8bit、温度高8bit、温度低8bit、校验8bit。

数据格式：40bit=16位湿度数据+16位温度数据+8位校验和

DHT22校验值取温湿度数据之和的低八位，通过DATA引脚输出的数字信息为未编码的二进制数据，依据其输出特性，可知第一个字节为校验和，第二个字节为温度小数，依次为温度整数、湿度小数和湿度整数，可以通过计算得到温湿度数据，计算方法7为：

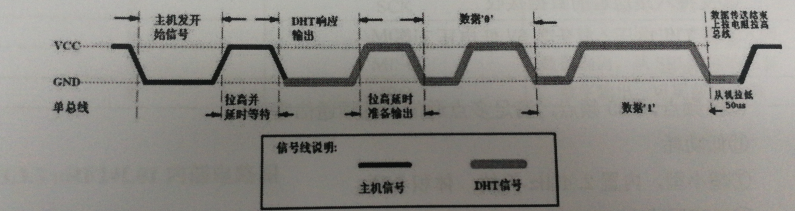
湿度示数=8位湿度整数数据+8位湿度小数数据

温度示数=8位温度整数数据+8位温度小数数据

校验和=取低八位（16bit湿度数据+16bit温度数据）

当用户发出起始信号后，DHT22模式就会发生转变，从原本的低功耗模式被动唤醒，开始进入稳定工作状态，此时传感器等待用户拉低总线信号必须大于18ms，直到主机检测到有效的起始信号，在DHT22温湿度传感器接收到主机发出的有效的起始信号后，是否需要查询温湿度数据，判定需要后DHT22就会对主机发出响应信号，从机拉高总线响应80us左右，随后进入数据传输过程，输出大小为40bit的温湿度数据。而在主机发送开始信号结束后，约20-40us左右，便可检测到DHT22的响应信号，在发送开始信号之后，可选择拉高电平输出亦或者切换到输入模式7。

通讯流程：



**3.3 NB-IOT通讯模块ME3616**

ME3616（下图）是一款支持NB-IOT通信标准的窄带蜂窝物联网通信模块。在NB-IOT模式,只消耗180KHZ的带宽便可以提供最大66K波特率的上行速率和34K波特率的下行速率，拥有三种部署方式包括保护带、带内和独立载波确保融入现有网络，无需重新搭建基站，为国家缓解经济压力，同时该模块可以支持多种网络协议包括MQTT、TCP/UDP、OMA-LWM2M、CoAP、ONENET和多种低功耗模式。这些协议和功能同时加上该模块LCC紧凑型封装模块，尺寸仅16mm×18mm，具有移动性，使得该模块应用场景极其广泛，目前已投入使用包括智能电表、无人智能停车场、坏境监测和可穿戴式设备等多种NB-IOT及M2M的应用场景中，内置eSIM的同时植入了物联网安全相关功能，集安全性、可靠性和高集成度于一体，拥有良好的发展前景。

### 3.3.1 ME3616 NB-IOT特性

⑴超低功耗：①DRX模式（非连续接受）：在指定时间内才能进入工作状态， 其余时间进入休眠状态，节省电力②eDRX（扩展不连续接受）：设 定PTW（寻呼时间窗口）内查询信号，进入工作状态，窗口之外时 间进入休眠状态以节省电力③PSM模式（省电模式）：在省电模式 下关闭终端射频信号，脱离工作状态，由终端提供数据通道恢复工 作状态；

⑵超强覆盖:同频段下，NB-IoT比现有的网络增益20dB，覆盖面积扩大100倍

⑶超多连接：数十万个网络同时连接，支持低功耗设备连接、超低设备成本、低 延时敏感度和高优化的网络架构；

⑷超低成本：利用现有的蜂窝数据网络，无需重新搭建基站。

ME3616电气特性

发射功率：NB-IOT：23±2.7dbm

电源供电：2.85V～3.6V

GNSS追踪灵敏度：-158dbm

GNSS捕获灵敏度：-145dbm

GNSS首次定位时间:冷启动 28s（典型值）

热启动 1s（典型值）

ME3616环境适应性

工作温度：-30℃ ～+75℃

扩展工作温度：-40℃ ～ +85℃

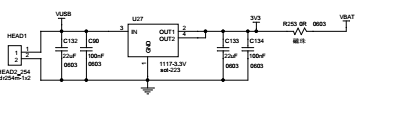
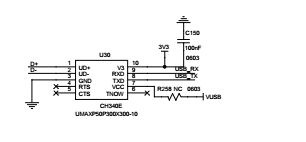
工作湿度：5% ～ 95%

存储温度：-40℃ ～ +85℃

### 3.3.2 ME3616引脚功能

|  |  |  |
| --- | --- | --- |
| 序号 | 引脚名称 | 引脚功能 |
| 1 | 供电接口 | 5V适配器供电接口 |
| 2 | 供电切换开关 | 拨码开关，用于USB供电和5V适配器供电选择开关 |
| 3 | USB | USB供电和程序下载 |
| 4 | AT\_UART | At通信口 |
| 5 | Debug\_UART | MICRO USB接口，用于程序下载 |
| 6 | PWE ON | 开关机键（支持唤醒PSM） |
| 7 | USB\_DL(USBdownload) | USB下载程序用（USB下载需焊接模块底部引脚） |
| 8 | RESET N | 复位按键 |
| 9 | WAKE\_IN | 唤醒按键支持唤醒（PSM） |
| 10 | SIM卡槽 | 插SIM卡 |
| 11 | GPS\_ANT | 连接GPS天线 |
| 12 | MAIN ANT | 连接主天线 |
| 13 | 信号接口 | 排针引出的模块相关信号 |
| 14 | 直流源供电 | 推荐用于低功耗测试时供电（供电参考电压3.4V） |

### 3.3.3 ME3616内部电路图

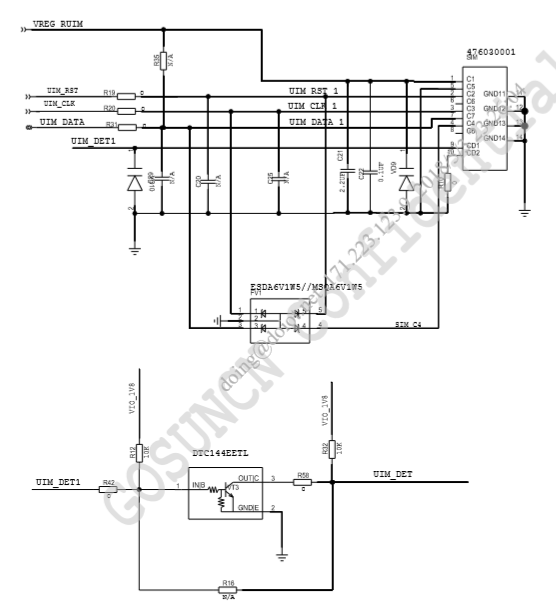
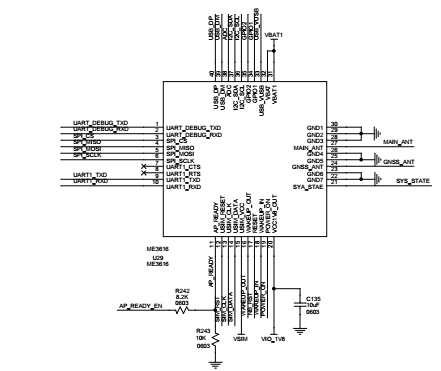


图 7ME3616内部引脚图 图 8 SIM CARD原理图

**3.4 显示电路**

OLED，即有机发光二极管（Organic Light-Emitting Diode）。具备自发光，不需背光源，且超薄、省电、广视角、对比度高、可弯曲、反应速度快、使用温度范围广、构造及制程较简单等优异之特性[[[8]](#endnote-8)]，被认为是下一代的平面显示器新兴应用技术。这也是我们从众多显示屏中选择SSD1306 OLED的理由。通过电源提供IIC、SPI和6800/8000多种接口方式，采取的是点阵液晶显示，整个屏幕的像素矩阵的划分较为特殊, 整个屏幕水平方向划分为8个page, 垂直方向则是按像素划分为128 column. 每个page-column包含8个像素[[[9]](#endnote-9)], 通过一个十六进制数(其实就是一个字节, 8个bit)来控制, 每个bit控制一个像素，通过发送二进制数字，确定该点亮灭。（如下图所示）与传统的LCD相比，OLED采取自发光模式，不再利用背光灯发光显示数据，在节约电力的同时，也有效的提高了显示分辨率。

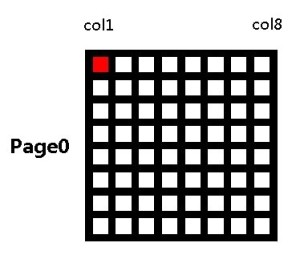


图 9 OLED显示原理

### 3.4.1 OLED 12864基本特性

显示分辨率：128x64

电源：（1）1.65V-3.3V（IIC驱动）（2）7V - 15V（面板驱动）

工作温度范围：-40°C to 85°C

引脚选择MCU接口：(1)256级对比度(2)最大电流为100uA

(3)最大反向电流为 15mA(4)最大输出驱动电压15V

拥有连续滚动在垂直和水平两个方向的保存功能

多重比率和可编程帧率

RAM写同步信号

行列重映射

片上晶振

点阵显示

### 3.4.2 OLED 12864引脚图功能

|  |  |  |
| --- | --- | --- |
| 引脚号 | 引脚名称 | 功能说明 |
| 1 | VCC | 电源正（2.8V～5.5V） |
| 2 | GND | 电源地 |
| 3 | D0 | 时钟引线 |
| 4 | D1 | 数据线 |
| 5 | RES | 复位线 |
| 6 | DC | 数据/命令控制引脚 |
| 7 | CS | 片选信号 |

### 3.4.3 OLED 12864原理图

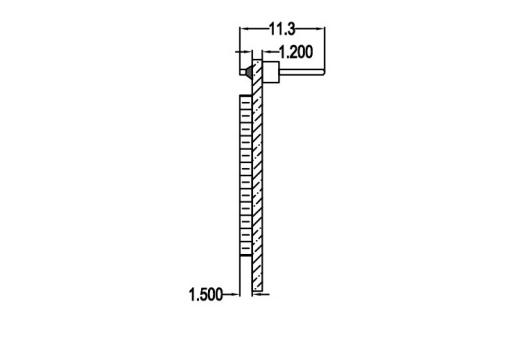
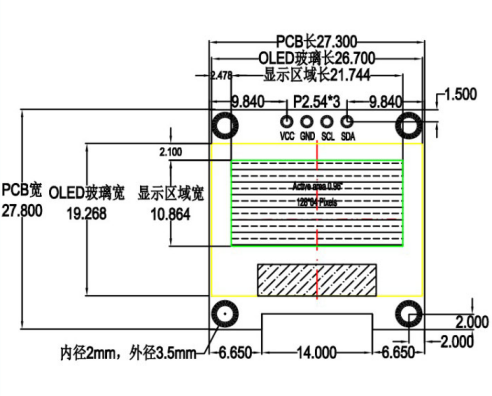


图 10 OLED12864结构图

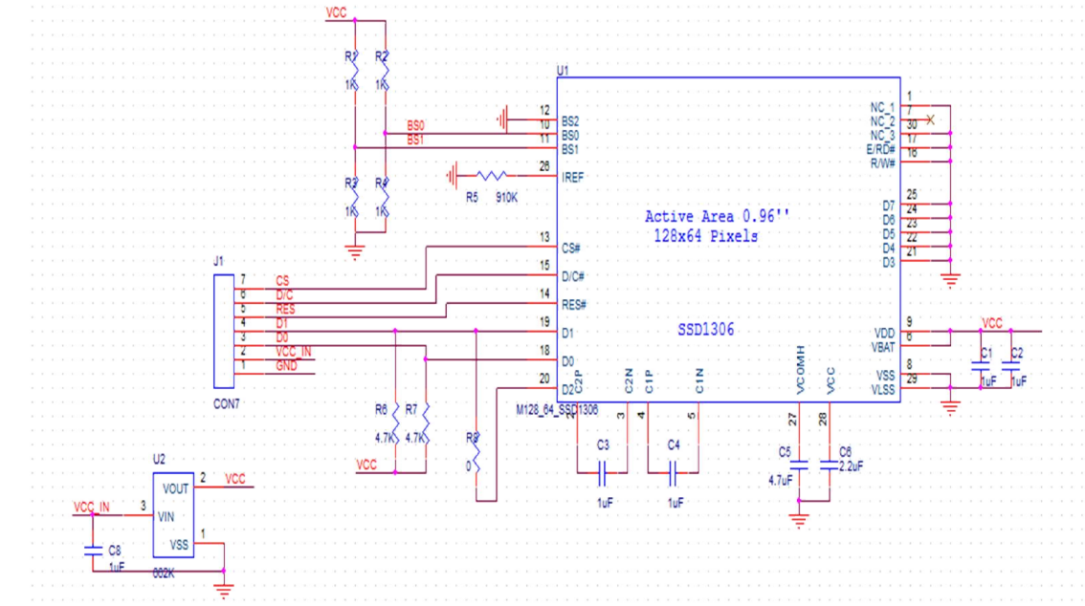


图 11OLED显示模块驱动原理及应用

# 

# 第4章 软件设计

本次的软件设计主要包括数据采集端和云平台端两部分，软件设计架构图如下图所示，先通过温湿度传感器DHT22采集数据，然后利用TCP/IP与云端服务器ONENET相连接，通过控制器发布通讯指令建立数据传输通道，通道建立后，根据CoAP（LWM2M）协议上传数据，将上传的温湿度数据通过ONENET云平台进行数据展示，实现数据的查询、记录、预警等功能。

**4.1软件开发平台和开发语言的介绍**

**4.1.1 设计语言的选择**

C语言：

二十世纪中叶，C语言于美国贝尔实验室横空出世，经不断改良后，成为一门广泛的用于设计计算机应用程序的基础语言，广泛应用于各种微型计算机上，也是一种在国际上被广泛应用于程序设计的高级语言，拥有十分丰富的功能，适用于电子计算机系统，能够确保计算机系统运转流畅，有效提高运行系统的稳定性。同时作为一门结构性语言，C语言拥有面对对象广、表达能力强、效率高、通俗性、可移植性强以及丰富的数据类型等优势，既能够编写系统软件，也能够编写应用软件，同时能与多种语言搭配使用，做到取长补短，使结构清晰，运行速率加快。例如当我们要运行复杂且冗长的设计程序时，可以配合汇编语言使用[[[10]](#endnote-10)]，利用汇编语言可以从直接存储器中读取数据的特性直接调用C语言子程序，这样既加快了CPU运行速率，也节约了存储空间。

Java：

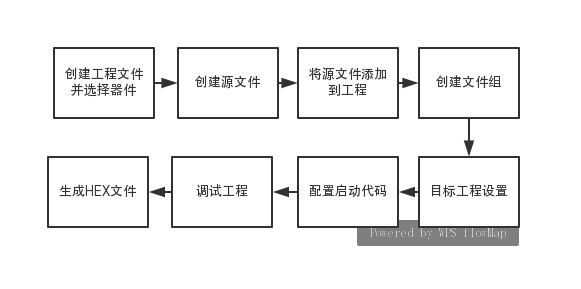
当人们利用C语言对机顶盒进行开发过程中，发现C语言指针向量极易引起错误取代，运算符重装、接口取代等问题，为解决这些问题，工程人员对C语言进行了改善，加入互联网新特征，增加接口转换等功能，这就是Java的来源。Java能够实现跨平台的移植，适合因特网的应有程序开发；Java语言具备简单性、独立性、面向对象、多线程并行和适应性强等特性[[[11]](#endnote-11)]。因为Java脱胎于C语言，故语言风格类型十分贴近C语言，它继承了C语言很多的优势，如面对对象，增加了动态联编的特性，扬新弃旧，增加了垃圾回收器，用于处理多余的内存空间，使得汇编过程更加流畅。随着Java技术的不断更新，在全球云计算和移动互联网的产业坏境下，Java具备更加广阔的前景。

论证：相比于于C语言，Java在开发过程中，对于字节代码的解释工作十分繁重，导致运行速度十分缓慢，需要特定的处理器才能加快转码速率，而C语言语言简洁、紧凑，更加接近自然语言，且生成代码质量高，执行效率远超Java，适用于本系统的开发条件，故本系统采用C语言作为编程语言。

**4.1.2 软件开发平台**

Keil ：

Keil uVision5 也称MDK-ARM，它是德国知名软件公司ARM公司开发的微控制器软件开发平台，提供丰富的库函数和功能强大的集成开发调试工具供开发人员使用，老版本Keil uVision4提供工程管理窗口，能够让工程人员使用多台监视器，完全控制窗口位置，使得开发应用程序变得更加清晰、高效[[[12]](#endnote-12)]。在其基础上，Keil MDK 5在提供安装包、编译器和调试跟踪的基础上，新增包管理器功能；同时支持LWIP，其SWD下载速度也是KEIL4的5倍，与Keil uVision4不同的是，为使得操作更加便捷，新版Keil MDK5将功能分为MDK Core和Software Packs分成两部分，其中MDK Core不仅要包含ARM Compiler5和IDE集成开发环境，Software Packs则可以在不更换MDK Core的情况下，单独管理（下载、更新、移除）设备支持包和中间件更新包[[[13]](#endnote-13)]，新版的Software Packs又分为Mdidleware、Device和CMSIS。下图为Keil uVision5创建工程流程。



Keil uVision5的特点：

功能齐全的源代码编译器

配置开发工具的设备数据库

用于创建和维护工程的工程管理器

继承汇编、编译和链接工具的make功能

用于所有开发坏境设置的对话框

具有高速CPU和外设仿真器的真正集成的源级和汇编调试器

为连接到keil ULINK适配器的目标硬件提供软件调试用的先GDI结构

供下载应用程序用的FLASH编译器

链接到用户手册、在线帮助、器件手册和用户指南各种文档[[[14]](#endnote-14)]

Arduino：

Arduino 是一款包含软件和硬件的、极易上手便捷灵活的开源开放电子原型平台，通过各种各样的传感器来感知环境获取信息，通过控制器处理后，可控制指定装置做出相应响应。本设计中也采用了DHT22作为感知环境采集数据的传感器，硬件Arduino Nano作为本次设计的控制芯片。Arduino具有丰富的接口功能，包括模拟I/O口、数字I/O口，同时支持IIC,SPI,UART串口通信；其中Arduino汇编语言以C语言作为基础，拥有C通俗易懂、表达能力强、效率高等特性，适用于单片机与汇编能力基础相对薄弱的新入门者；不仅如此，Arduino硬件具有价格低廉、扩展能力强等优点，加上简便的编程环境IDE，丰富的库函数，为新入门开发者开发程序功能提供了便捷服务，符合本次设计的简单、经济、高效的要求。

Arduino五大特点：

（1）Arduino有丰富的接口功能，可与不同电子元件、传感器进行简单连接，比如本设计使用的DHT22，还有各种红外线、光敏电阻等，适用于一些如城市智能垃圾桶应用场景，简单，快捷高效。

（2）Arduino以开源作为自己的理念，便于二次开发，在技术上和软硬件完全开放，拥有丰富的库函数，还为开发者精心准备了样例程序，如blink、fade、diaplay等，支持开发者进行简单修改，编写自己的目标程序

（3）支持ISP线上烧入器，通过USB线直接完成烧录代码，不再需要烧录器，降低开发成本;其它Arduino配套硬件相对低廉，例如采用低价格的微处理控制器ATmega8或ATmega128，支持新学者入门单片机开发

（4）Arduino汇编语言以C语言作为基础，拥有C通俗易懂、表达能力强、效率高等特性，加上为开发者提供各式各样的样例程序，能够让新入门者迅速进入开发角色

（5）支持各式各样的互动程序，如VVVV，C、PurEDAta

在设计过程中，我们先是通过Arduino软件平台对工程代码进行调试，在调试成功后进行拓展，又选用了Keil作为调试软件，经调试得出与Arduino相同的结果。经对比，在本次设计中，Arduino的语言更加简洁，特别是自带提供开发板设计模板，使得调试过程更加便捷、快速和高效，最终我们选择了Arduino作为软件开发环境。

**4.2 ONENET云平台**

ONENET是中国移动为物联网开发者推出的物联网开放平台，定位为PaaS服务，通过支持多种行业及主流标准协议的设备接入，如CoAP（LWM2M）、MQTT、EDP、HTTP，以无物不连作为最终实现目标，在真实设备和物联网应用之间搭建稳定、安全、高效的数据联通通道；同时提供提供丰富的API接口和应用模板，满足多种应用场景的使用需求。对设备进行增删改查、数据流创建、数据点上传、命令下发等操作。本系统采用ONENET平台作为服务器进行数据管理[[[15]](#endnote-15)]。

**4.2.1 设备接入**

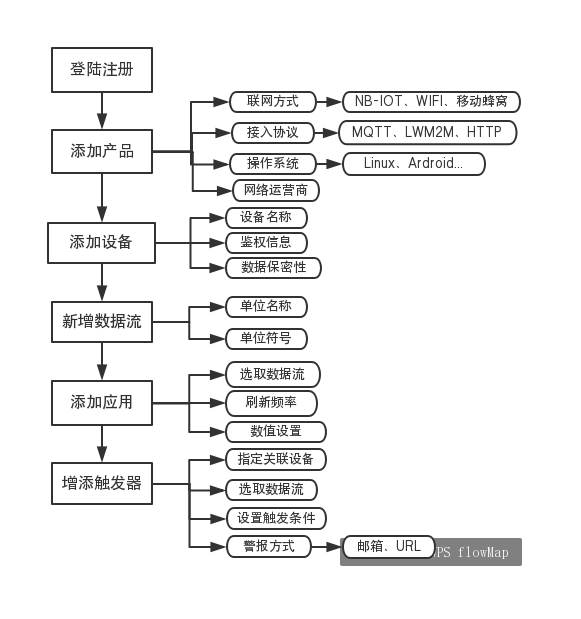


图 12 Onenet平台设置流程图

**4.2.2触发器设置**

为实现无人实时监控数据流，中国移动ONENET为开发者提供基于数据流逻辑判断的触发器预警功能。开发者可以先通过ONENET云平台自带的触发器管理功能，选择相应设备数据流进行监控，且单个触发器可以同时监控多个设备相同类型的数据流，然后设置触发条件，选择条件触发后预警的接收方式，当数据流的数据和触发器条件相符合时，平台可以通过邮件、URL地址、短信等方式向用户发送预警信息，通知工程人员及时处理。以MR机房为例，我们了解到磁共振的许多物理参数很容易受环境变化和时间的影响发生漂移,，MR设备机房包括操作间、控制间和设备间，全都对坏境温湿度有严格的规范，要求机房的温度维持在15℃～21℃，湿度要求维持在30%～60%之间。因为触发器一次只能设置单个预警条件，系统将高于20℃和低于15℃设置为一个触发器；高于60%和低于30%设置为一个触发器；当符合触发条件时，平台将通过邮件通知工程人员机房状态异常，请及时处理。



**4.2.3传输协议**

为实现无物不连的目标，ONENET面对设备提供了不同的标准协议适配不同的应用场景，包括MQTT、LWM2M（CoAP）、Modbus、HTTP/HTTPS、TCP等协议，根据不同场景的特点我们选用不同的接入协议，例如在，LWM2M（CoAP）协议适用于低功耗以及广覆盖的场景；而在工业通信中，我们一般采用DTU+Modbus作为我们的通信协议；单纯上报数据采用HTTP/HTTPS居多；本次设计目的为远程实时监控机房温湿度变化，首先要做到远程，采用LWM2M（CoAP）作为本次设计协议，利用其广覆盖的基础上，配合NB-IOT技术的远程数据传输，实现对大型医疗设备机房的实时性监控，故本次设计采用LWM2M（CoAP）协议配合其它元件完成本次设计。

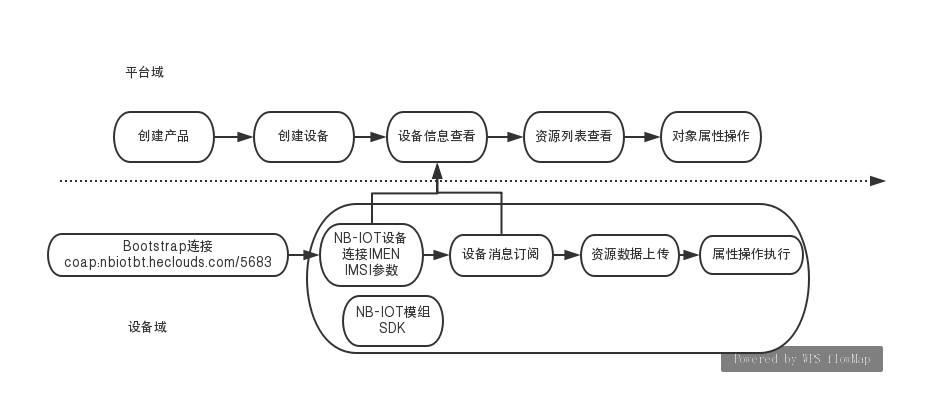
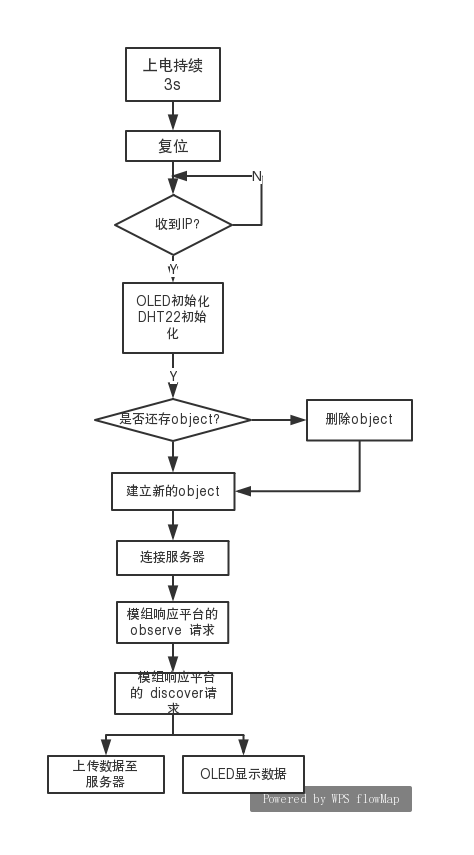


图 13LWM2M（CoAP）协议接入流程图

**4.3数据采集**

下图为本次设计数据采集流程图及对应程序（程序调用详见附录Ⅱ）



void setup()

{

Serial.begin(115200);

Serial.setTimeout(100);

beginME3616();//上电，复位，通过判断ip确认入网

Serial.println("DHTxx test!");

dht.begin();

while (1)

{

while (!sendCMD("AT+MIPLCREATE"))//判断是否能建立远程实例

{

Serial.println("Failed to create a onenet instance. Try again ...");

sendCMD("AT+MIPLCLOSE=0");//关闭前一次实例

sendCMD("AT+MIPLDELOBJ=0," + OBJ\_ID\_TEMP);

sendCMD("AT+MIPLDELOBJ=0," + OBJ\_ID\_HUMI);

sendCMD("AT+MIPLDELETE=0");

}

if (!sendCMD("AT+MIPLADDOBJ=0," + String(OBJ\_ID\_TEMP) + ",1,\"1\",1,0"))//建新的实例

if (!sendCMD("AT+MIPLOPEN=0,3600"))//连接服务器

if (!sendCMD("AT+MIPLOBSERVERSP=0," + tmpMsgid + ",1"))//模组响应平台下发observe

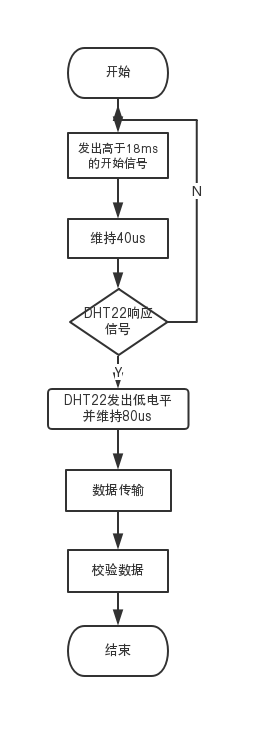
if(!sendCMD("AT+MIPLDISCOVERRSP=0,"+tmpMsgid+ ",1,4,\"" + String(RES\_ID\_VAL) + "\""))//模组响应平台下发的discover

sendData(msgid[0], OBJ\_ID\_TEMP, RES\_ID\_VAL, temperature）;//发送温度

sendData(msgid[1], OBJ\_ID\_HUMI, RES\_ID\_VAL, humitidy);//发送湿度

**4.4 DHT22温度传感器程序设计**

本设计采用DHT22作为温湿度传感器，下图为其程序流程图。



温湿度数据读取程序如下

delay(2000);//等待传感器响应

dht.readHumidity（）; //读取温度为摄氏度（默认值）

float h = dht.readHumidity();

float t = dht.readTemperature();// 读取温度为摄氏度（默认）

float f = dht.readTemperature(true);// 读取温度为华氏温度

if (isnan(h) || isnan(t) || isnan(f)==0) //检查是否读取成功

{

While（isnan(h) || isnan(t)）；

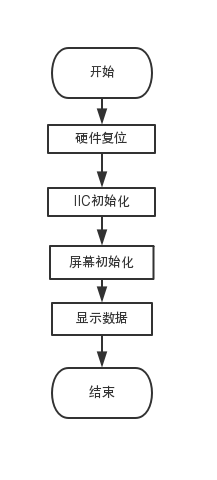
Serial.print(h);

Serial.print(t);

......//通过串口传输数据}

**4.5 显示电路子程序**

本设计采用SSD1306作为温湿度显示设备，首先初始化 OLED 屏幕，然后我们再调用OLED\_CLEAR(),完成屏幕初始化清屏，调用温湿度读取程序获得数据后设选定屏幕坐标，调用函数库，选择汉字、数字亦或者字符串，设置长度、大小后显示数据（下图为OLED程序流程图）。



DelayInit();

OLED\_Init();

while(1)

{

OLED\_Fill(0xFF);//全屏点亮

DelayS(2);

OLED\_Fill(0x00);//全屏灭

DelayS(2);

for(i=0;i<5;i++)

{

OLED\_ShowCN(22+i\*16,0,i);//测试显示中文

}

DelayS(2);

OLED\_ShowStr(0,3,"HelTec Automation",1);//测试6\*8字符

OLED\_ShowStr(0,4,"Hello Tech",2); //测试8\*16字符

DelayS(2);

OLED\_CLS();//清屏

OLED\_OFF();//测试OLED休眠

DelayS(2);

OLED\_ON();//测试OLED休眠后唤醒

OLED\_DrawBMP(0,0,128,8,(unsigned char \*)BMP1);//测试BMP位图显示

DelayS(2);}

# 第5章 软硬件调试

在设计过程中常常结果与预期发生偏差，此时便中断实验，进行错误排查，为提高效率，本设计分别将软硬件进行调试，从硬件开始依次排除通讯模块ME3616、传感器DHT22、控制芯片Arduino Nano、OLED12864干扰。

**5.1通讯模块调试**

单独将通讯模块ME3616进行功能检查，利用串口助手依次发布指令，观察ME3616是否能与云端进行数据通讯，

AT+MIPLCREATE//创建 OneNET instance

AT+MIPLADDOBJ=0,3303,3,"100",1,0//创建一个object（对象）

AT+MIPLADDOBJ=0,3304,3,"100",1,0//创建一个object（对象）

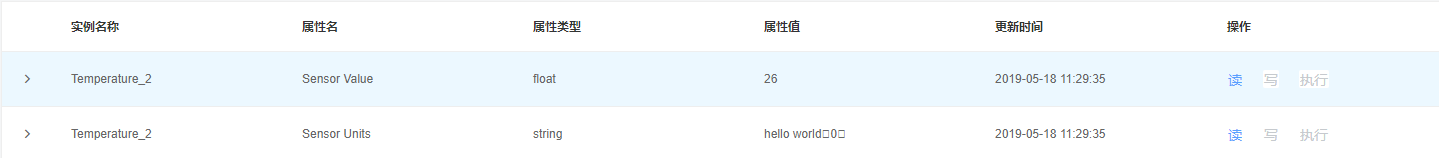
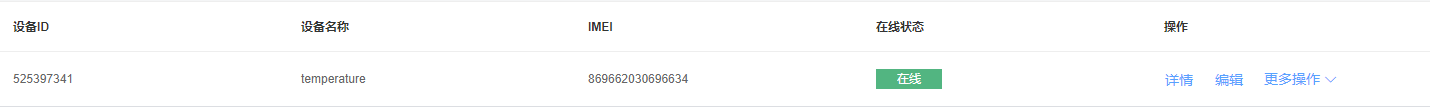
AT+MIPLOPEN=0,3600//设备注册到 OneNET 平台，3600秒

AT+MIPLOBSERVERSP=0,115493,1//OneNET 平台向模组发起 observe 请求

AT+MIPLDISCOVERRSP=0,49958,1,9,"5700;5700"//模组响应平台observe 请求

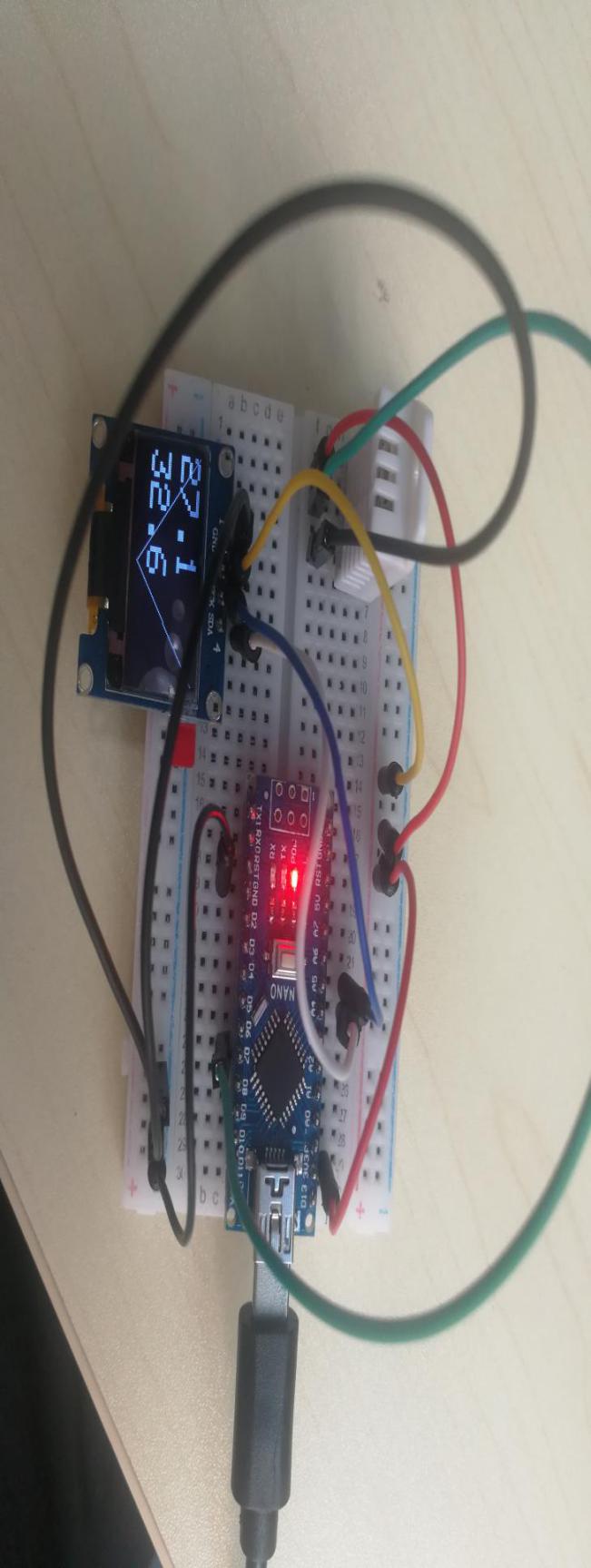
AT+MIPLNOTIFY=0,115493,3303,2,5700,4,,data,1,1//上传数据

经调试，ME3616能够将数据顺利上传的云平台（见下图），可知通讯模块ME3616功能完备



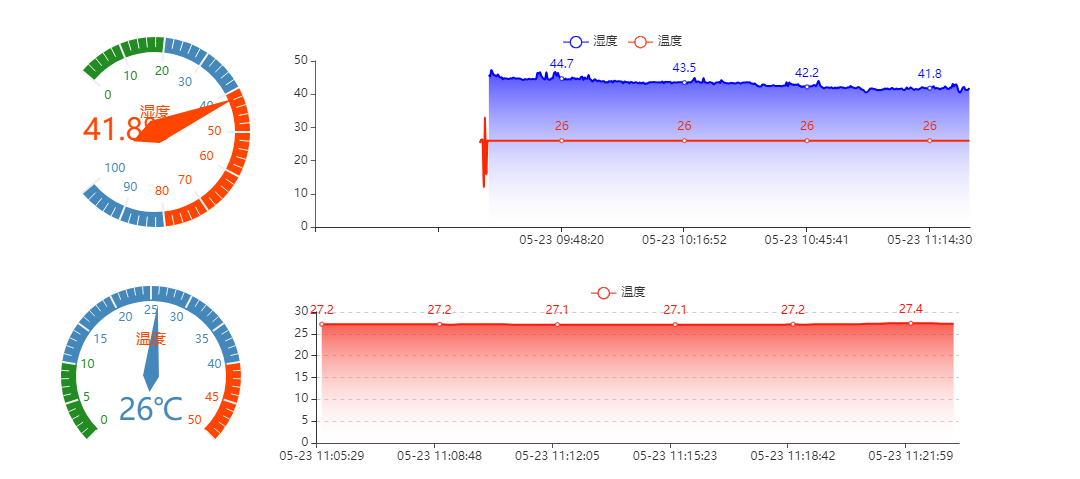
**5.2 控制模块调试**

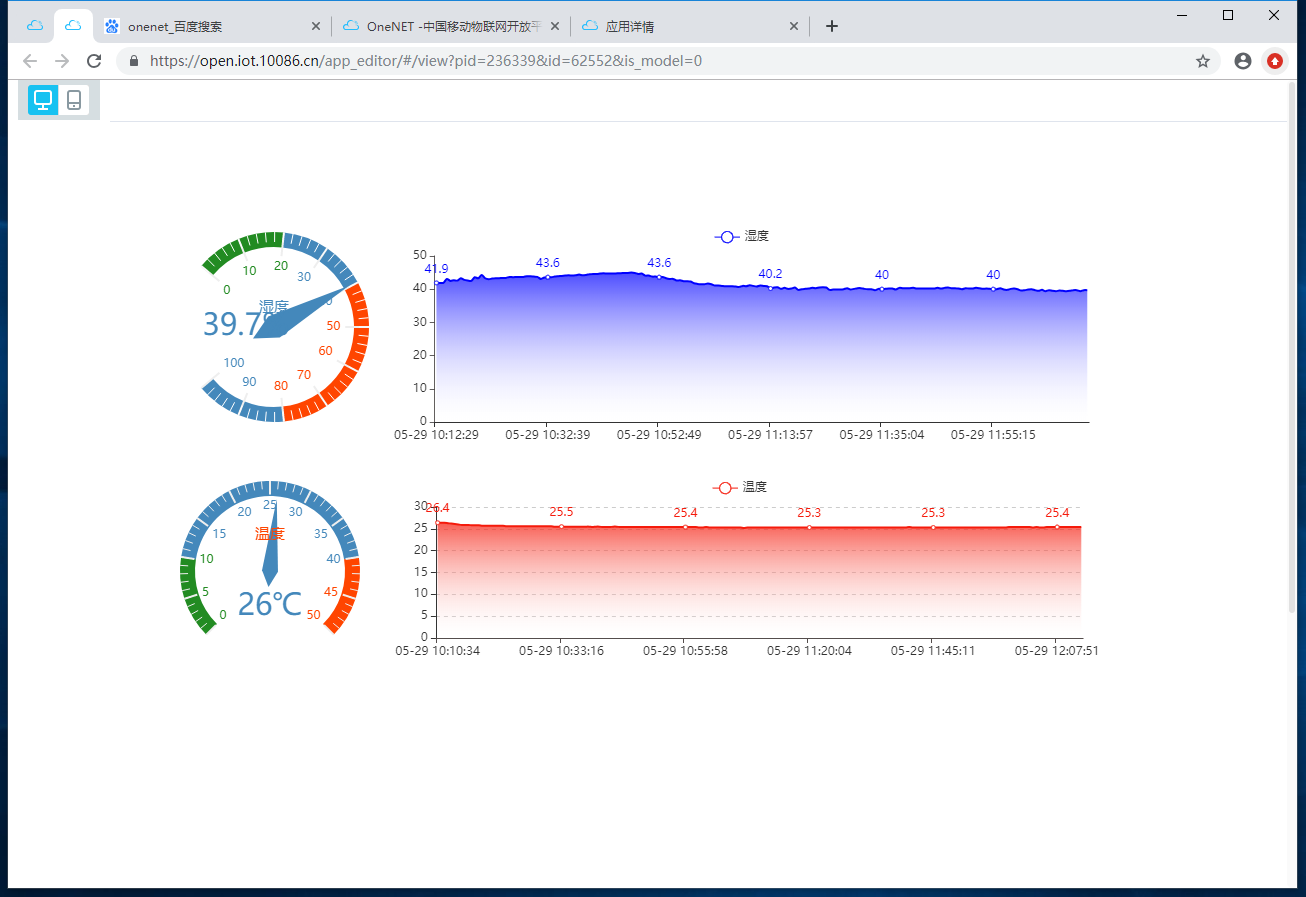
通过控制器，将温湿度和显示屏同时进行调试。首先利用传感器DHT22采集模拟数据，A/D转换后通过DATA引脚将数据传输至控制器Arduino Nano，控制器将传输的数据惊醒相应的处理后，通过IIC引脚连接OLED12864，由控制器将处理后的数据传输至显示屏并将数据显现。经调试，模块正常运行。（调试程序详见附录Ⅰ）

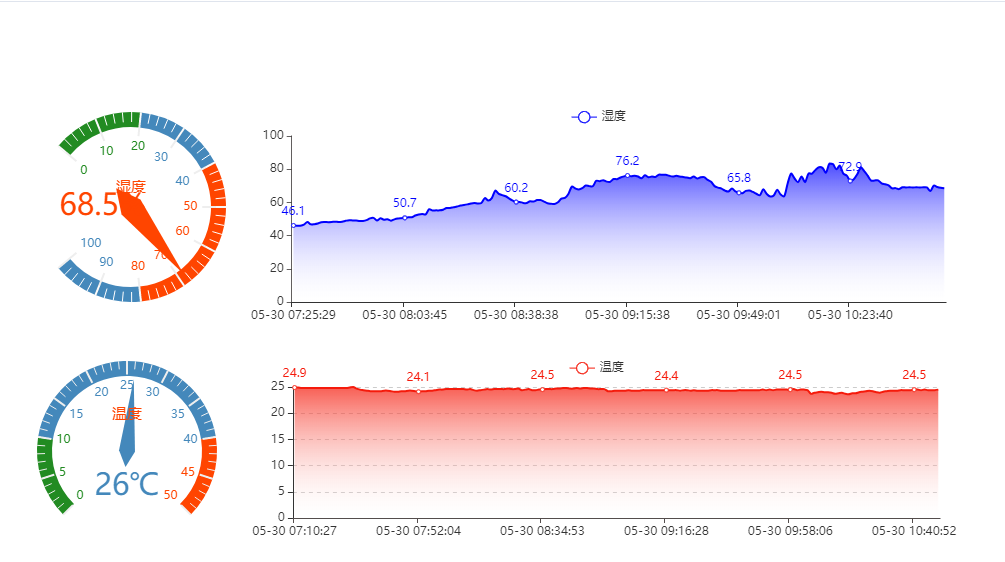


# 第6章实验结果与分析

**6.1实验测试结果展示**





****

**6.1.1实验测试结果**

|  |  |  |  |
| --- | --- | --- | --- |
| 实际温度（℃） | 测量温度（℃） | 实际湿度（%） | 测量湿度（%） |
| 27.9 | 27.2 | 44.5 | 44.7 |
| 27.9 | 27.1 | 42.9 | 43.5 |
| 27.9 | 27.1 | 42.8 | 43.2 |
| 27.5 | 27.1 | 41.0 | 41.8 |
| 27.5 | 27.2 | 42.3 | 41.6 |
| 27.5 | 27.4 | 41.9 | 42.8 |

表 3随机实验测试数据

**6.1.2误差分析**

（1）空气中存在团状水雾，且温度随着气浪波动，在本次设计中，单位时间内传感器DHT22采集数据，数据经平均处理后显示，而实际温湿度不稳定，故实际温湿度数据不稳定。

（2）本设计采用多个电子元器件集成，使用直流电，采集温湿度过程中，电流不稳定使得元器件周围温度升高。

（3）传感器灵敏度跟不上采样频率

（4）传感器精度有待提高。

**6.2实验总结和展望**

经过一周的测试，能够采集温湿度数据上传服务器，实现该设计的最初目标，达成远程实时进行湿度监测、湿度预警、温度监测和温度预警的目的。

因时间和能力有限，该设计还有诸多可完善地方，比如加入一个蜂鸣器，当温湿度不符合规定，立刻进行现场示警；同时医疗设备受诸多坏境因素影响，不仅包括温湿度，还有空气中粉尘颗粒含量、噪声等，可以通过物联网技术、传感器技术等添加此类医疗设备环境监控，这也是未来工作生活中将要继续推进的方向。

# 文献综述

基于NB-IOT技术的大型医疗设备环境监测系统

陆正大

南京医科大学生物医学工程与信息学院学院生物医学工程系 15060206

**[摘要]**由于现代化医疗设备大部分是由微机及智能电子电路控制，辅以高压电路，所以对环境中湿度、温度、粉尘颗粒等因素的影响较为敏感，严重干扰了医疗设备的正常运行，课题以Arduino Nano单片机作为控制芯片和基于物联网云平台搭建大型医疗设备坏境监测系统，把利用温湿度传感器DHT22采集的数据信息，通过NB-IOT技术构建通信通道，上传到ONENET云平台上进行展示和分析，为工程人员的日常巡检、故障诊断、维护保养提供便捷。

**[关键词]**环境监测；Arduino Nano；NB-IOT；ONENET云平台；

**Abstract:** Because most modern medical equipment is controlled by microcomputers and intelligent electronic circuits, supplemented by high-voltage circuits, it is sensitive to the influence of humidity, temperature, dust and other factors in the environment, which seriously interferes with the normal operation of medical equipment. The subject is Arduino Nano.The MCU is used as a control chip and an IoT cloud platform to build a large medical equipment environment monitoring system. The data information collected by the temperature and humidity sensor DHT22 is used to construct a communication channel through the NB-IOT technology, and uploaded to the ONENET cloud platform for display and analysis.It provides convenient for daily inspection, fault diagnosis and maintenance of engineers.

**Keywords**:Environmental Monitoring; ArduinoNano;Nb-Iot; Onenet Cloud Platform;

随着科学技术的高速发展，越来越多的大型医疗设备投入医院使用，这些仪器电路包含大量的电子元器件，所以为温湿度的变化对仪器有重大的影响，例如CT/SPECT，使用手册标明扫描间和操作间温度应在18℃到26℃之间且其变化率必须不超过3℃/h，否则探头内晶体可能会因温度的剧烈变化而导致变形破裂；而湿度要求保持在30%到60%之间，变化率每小时不超过5%，若湿度过高，构成探头内的光电倍增管和碘化钠晶体就会吸附水汽氧化，过低则有可能使得晶体炸裂[[[16]](#endnote-16)]。CT作为医院目前使用率最高的影像设备，经常性会在工作中过程中遇到扫描停止，控制台不断报警，指示温湿度过高，经现场检查发现尘埃废絮堵塞了水风冷机进风口的金属滤网，热量不能及时从机架内散发出去造成多排螺旋CT出现机架内温度过高触发保护装置停机，通过清理，使得机架内温度降低，报警解除2。当MR制冷系统失常时，会使空气湿度达到饱和状态，导致在高压设备的表面冷凝成水，并伴随时间的延长而增多，滴落高压部件后，极易引起短路，造成失超，导致不可挽回的损失，为了不影响仪器的正常使用，因此我们需要对温湿度进行严格的把控，防止其影响成像质量，影响医生对病情做出正确判断。

而传统的方法是利用具有只有温度和湿度的检测仪表，依靠人工对机房温度和湿度进行检测和控制，通过这种方法检测出来的温度和相对湿度误差会很大，而且具有时延性，不能维持坏境参数在规范之中，当因温湿度失常导致设备故障时，我们的工程人员将无法及时赶赴现场处理问题，也无从得知是否为温湿度失衡而导致的机器故障，需要通过现场排查才能确定故障来源，降低了工作效率。基于上述情况，通过研究调查，本设计在基于WIFI温湿度实时监控上进行创新，以Arduino Nano采用作为主控芯片，采用基于NB-IOT技术设计的开发板ME3616作为通讯模块，选用DHT22作为温湿度传感器采集数据，首先将温湿度数据通过OLED12864呈现；同步将温湿度数据实时上传到中国移动云平台ONENET，ONENET自带应用模块将自动识别采集数据，绘制折线图，当检测到机房温湿度出现异常，立即触发阈值，启动报警功能，发送报警提示文件通知工程师及时赶往现场进行检修；工程师还可以通过观察温湿度历史数据，排查是否为温湿度失常造成的故障，一方面实现了医疗设备坏境的实时检测，另一方面提高了工作效率。

本次采用Arduino Nano作为控制芯片，这是一款基于Arduino Nano 3.0的小型完整可直插面包板的开发板，采用无电源插座和Mini-B USB接口，并基于Arduino开发板设计了多种引脚功能，其中ICSP在线编程接口，能够让用户通过编程电缆直连编程器，实现在线编程；很方便进行开发调试；此外Arduino Nano包括14个数字输入/输出引脚，其中有六个具有PWM功能；还有既可以作为模拟输入又可以作数字输入/输出的模拟引脚；同时Arduino Nano集成了一个复位按钮、一个ICSP header、和一个16MHz晶体振荡器。在本次设计中，DHT22采集的温湿度数据具有离散的特性，可利用Arduino Nano自带的串口数字引脚进行PWM波控制引脚输出，并且ME3616开发板与Arduino Nano的功能引脚能够相匹配，故采用Arduino Nano为本次的控制芯片既方便又灵活。

本次设计中最有意义的就在于它的通讯模块，可谓设计的绝对精髓。目前存有多种传输方式，例如WiFi、LoRa、蓝牙等，经研究发现，WiFi基于无线路由器作为组网方式，传输速率高，但续航能力及传输距离较NB-IOT技术和LoRa略显弱势，而蓝牙仅能维持在10m范围的拥有传输信号，不符合本系统设计的远程监控的要求，LoRa技术节点和网关部署要求极高，需要重新布置节点，成本过高，只有NB-IOT技术，基于现有的蜂窝网络，能够融入当前的网络，无需重新布置节点和网关，节约成本的同时，扩大了现有网络供应商的服务范围；而我们要进行的温湿度数据采集具有实时动态、大量散乱的特点，而远距离数据传输也是一个巨大挑战。目前而言，长距离传输技术主要包括蜂窝网络技术与低功耗广域网技术，而蜂窝通信进行广域数据传输存在功耗大、成本高、网络速率与应用需求不匹配等问题，广域网技术适配低速率场景，发展前景广阔。目前经授权频段的广域网技术主要为窄带物联网（NB-IOT）与LTE（eMTC）演进技术，而eMTC主要用于语音场景，所以NB-IOT成为了我们的首选，NB-IOT技术具有四大优势：功耗低、容量大、覆盖广、成本低，适应了本设计中的数据特点，同时能够实现远距离观测温湿度的目的。

参考CT/SPECT的性能指标，每小时温度变化不能超过3℃，对于精准度、稳定性和响应时间要求极高，所以不难看出，传感器的选择能够直接决定本设计的成败，在反复验证后，我们选择了DHT22作为温湿度传感器，DHT22为4针单排引脚封装，连接方便；内部由一个电容式感湿原件和NTC测温元件组成，连接了一个高性能的八位单片机，，是一种应用了集成温湿度传感技术和数字模块采集技术的温湿度复合采集元件，该产品不仅在出厂之前在特定坏境中进行过精准较准之外，还能够在开始数据采集之前，利用出厂时存储在OTP内存中的校正系数进行自我精准校准，在采集数据时，有效提高了采集数据的准确性，测量精度到达±0.3%RH；根据测试，该产品漂移仅在±0.3℃，响应时间不超过5s，还具有强大的抗干扰能力，年稳定性不超过±0.5%，使得该产品能够适用于不同类型环境恶劣的场所。

我们选用OLED12864作为显示模块，OLED，即有机发光显示器（Organic Light-Emitting Diode）。具备发光均匀性高、超薄、宽视角、低功耗、可弯曲、反应速度快、使用温度范围广、构造及制程较简单等优异之特性4，具有广阔的发展前景。这也是我们从众多显示屏中选择OLED 12864的理由。在功能上，OLED 12864提供多种接口方式，例如IIC、SPI和6800/8000；显示方式采取的是点阵液晶，将整个屏幕的像素矩阵均匀划分, 水平方向划分为8个page, 垂直方向则是按像素划分为128 column. 每个page-column包含8个像素, 通过一个十六进制数(其实就是一个字节, 8个bit)来控制, 每个bit控制一个像素，通过发送二进制数字，采取自发光的特性，控制该点亮灭，发光强度与注入的电流成正比。与传统的LCD相比，不仅亮度更高、体积更薄、响应速度更快等优势，而且不再利用背光灯发光显示数据，在节约电力的同时，有效提高了显示分辨率。

设计选用onenet作为云端服务器，ONENET是中国移动为物联网开发者推出的物联网开放平台，通过支持多种行业及主流标准协议的设备接入，如CoAP（LWM2M）、MQTT、EDP、HTTP，以无物不连作为最终实现目标，在真实设备和物联网应用之间搭建安全、高效、稳定的数据联通通道；同时提供提供丰富的API接口和应用模板，满足多种应用场景的使用需求。ONENET平台提供触发器功能,

通过设置触发条件，选择触发后预警的接收方式，当指定数据流中的数据符合触发器条件时，平台可以通过邮件、URL地址、短信等方式向用户发送预警信息，通知工程人员及时处理，实现实时监控；此外，平台将记录控制器通过ME3616上传的历史数据，当故障发生时，维修人员可以参考历史数据进行排查，是否为温湿度失常所造成的设备故障，能够有效提高工作效率。

经过软硬件调试，我们首先确定了软硬件设备状态完好，然后开始试验，选择一个具有代表性的CT机房，温度要求18℃到26℃，湿度30%到60%，经过连续一周的测试，最终的检测结果显示，温湿度准确率可达95%以上，误差数据经计算在3%以下。在经过综合分析，我们将误差锁定为以下三个原因，第一可能因为空气中存在团状水雾，且温度随着气浪波动，在本次设计中，单位时间内传感器DHT22采集数据，数据经平均处理后显示，而实际温湿度不稳定，故实际温湿度数据不稳定。第二，本设计采用多个电子元器件集成，使用直流电，采集温湿度过程中，电流不稳定使得元器件周围温度升高；第三，传感器灵敏度跟不上采样频率，传感器精度有待提高。

本次课题设计的基于NB-IOT技术的大型医疗设备环境监测系统，综合利用物联网技术，传感器技术，计算机技术，云平台技术。做到了医疗设备的温湿度监测，但因时间和能力限制，在未来的日子，本人将继续完善本次课题设计。

# 附录Ⅰ

// DHT Temperature & Humidity Sensor

// Unified Sensor Library Example

#include <Adafruit\_Sensor.h>

#include <DHT.h>

#include <DHT\_U.h>

#include <Adafruit\_ssd1306syp.h>

#define SDA\_PIN D2

#define SCL\_PIN D1

#define DHTPIN D7

#define DHTTYPE DHT22 // 使用中的传感器类型：DHT 22 (AM2302)

Adafruit\_ssd1306syp display(SDA\_PIN,SCL\_PIN);

DHT\_Unified dht(DHTPIN, DHTTYPE);

uint32\_t delayMS;

void setup() {

Serial.begin(9600);//设置数据传输波特率

dht.begin(); //初始化设备。

sensor\_t sensor;

dht.temperature().getSensor(&sensor);

Serial.println(F("------------------------------------"));

Serial.println(F("Temperature Sensor"));

Serial.print (F("Sensor Type: ")); Serial.println(sensor.name);//传感器名称

Serial.print (F("Driver Ver: ")); Serial.println(sensor.version);//传感器版本

Serial.print (F("Unique ID: ")); Serial.println(sensor.sensor\_id);

Serial.print (F("Max Value: ")); Serial.print(sensor.max\_value); Serial.println(F("°C"));

Serial.print (F("Min Value: ")); Serial.print(sensor.min\_value); Serial.println(F("°C"));

Serial.print (F("Resolution: ")); Serial.print(sensor.resolution); Serial.println(F("°C"));

Serial.println(F("------------------------------------"));

delayMS = sensor.min\_delay / 1000;

delay(1000);

display.initialize();

}

void loop() {

// Delay between measurements.

delay(delayMS);

// Get temperature event and print its value.

sensors\_event\_t event;

dht.temperature().getEvent(&event);

if (isnan(event.temperature)) {

Serial.println(F("Error reading temperature!"));

}

else {

Serial.print(F("Temperature: "));

Serial.print(event.temperature,1);

Serial.println(F("°C"));

display.drawLine(0, 0, 63, 63,WHITE);

display.update();

delay(1000);

display.clear();

display.setTextSize(3);//设置字体大小

display.setTextColor(WHITE);//设置背景

display.setCursor(0,0);//显示位置

//display.setCursor(0,0);//列127 行63坐标

display.println(event.temperature,1);

}

// Get humidity event and print its value.

dht.humidity().getEvent(&event);

if (isnan(event.relative\_humidity)) //判断是否为有效湿度

{

Serial.println(F("Error reading humidity!"));

}

else {

Serial.print(F("Humidity: "));

Serial.print(event.relative\_humidity,1);//串口输出

Serial.println(F("%"));

//湿度显示

display.drawLine(127, 0, 63,63,WHITE);

display.setCursor(0,32);//列行，显示位置

display.setTextSize(3);//设置字体大小

display.setTextColor(WHITE);//背景颜色

display.println(event.relative\_humidity,1);

}

}

# 

# 附录Ⅱ

#include <Arduino.h>

#include <Adafruit\_ssd1306syp.h>

#include "DHT.h"

#define SDA\_PIN A4

#define SCL\_PIN A5

#define OBJ\_ID\_TEMP 3303

#define OBJ\_ID\_HUMI 3304

#define RES\_ID\_VAL 5700

#define PIN\_RST PD3

#define PIN\_PWR PD4

#define DHTTYPE DHT22

#define DHTPIN PD7 // 选择dht输入输出脚

DHT dht(DHTPIN, DHTTYPE);

Adafruit\_ssd1306syp display(SDA\_PIN,SCL\_PIN);

String buffer = "";

String tmpMsgid = "";

String msgid[2] = "";

void beginME3616()

{

pinMode(PIN\_PWR, OUTPUT);//power on上电

digitalWrite(PIN\_PWR, HIGH);

delay(3000);

digitalWrite(PIN\_PWR, LOW);

//reset复位

pinMode(PIN\_RST, OUTPUT);

digitalWrite(PIN\_RST, HIGH);

delay(100);

digitalWrite(PIN\_RST, LOW);

while (1)

{

while (Serial.available() == 0)

;

buffer = Serial.readString();

if (buffer.indexOf("IP") != -1) //返回字符串包含IP

break;

}

}

bool sendCMD(String cmd)

{

long tm = millis();

Serial.println(cmd);

while (1)

{

tm = millis();

while (Serial.available() == 0) //等待串口缓冲区有数据

{

if ((millis() - tm) > 10000) //10s无数据返回超时

{

Serial.println("TIMEOUT");

return false;

}

}

buffer = Serial.readString(); //读缓冲区

if (buffer.indexOf("OK") != -1) //返回信息包含OK

return true;

if (buffer.indexOf("ERROR") != -1) //返回信息包含ERROR

{

Serial.println("ERROR");

return false;

}

}

}

void sendData(String msgid, int objid, int resid, float data)

{

// AT+MIPLNOTIFY=<ref>, <msgid>, <objectid>, <instanceid>, <resourceid>, <valuetype>, <len>, <value>, <index>, <flag><CR>

// <ref> : AT+MIPLCREATE消息发送成功后，返回的OneNET instance值

// <msgid>:整型，消息id

// <Objected> : 整型, 发送notify请求的object id

// <instanceid> : 整型, 发送notify请求的instance id

// <resourceid >:整型, 发送notify请求的resource id

// <valuetype> : 整型, 发送notify请求的上报数据类型（1 string; 2 opaque; 3 integer; 4 float; 5 bool ）

// <len> : 整型, 发送notify请求的上报数据长度

// <value> : 整型, 发送notify请求的上报数据值

// < index >整型：配置文件表示，范围从N-1到0

// < flag >整型：消息标志，1 first message; 2 middle message; 0 last message (请填0, 用index来区分, 后续所有AT命令里的flag都填0)

char arr[5];

dtostrf(data, 2, 1, arr);

Serial.println("AT+MIPLNOTIFY=0," + msgid + "," + String(objid) + ",0," + String(resid) + ",4,4," + String(arr) + ",0,0");

}

String skipResponse(int offset) //跳过无用的响应，返回最后一次响应的字符串

{

String buffer;

long tm = millis();

while (offset > 0)

{

tm = millis();

while (Serial.available() == 0)

{

if ((millis() - tm) > 10000) //10s无数据返回超时

{

Serial.println("TIMEOUT");

return "";

}

}

buffer = Serial.readString();

offset--;

}

return buffer;

}

String getMsgId(String buffer) //获取字符串中包含的Message ID

{

int firstComma = buffer.indexOf(",");//找到字符串中第一个“，”

int secondComma = buffer.indexOf(",", firstComma + 1);//从第一个‘，’开始找到第二个‘，’

return buffer.substring(firstComma + 1, secondComma);//截取两个‘，’之间的message id

}

void setup()

{

Serial.begin(115200);

Serial.setTimeout(100);

delay(1000);

display.initialize();//初始化显示屏

beginME3616();//上电，复位，通过判断ip确认入网

while (1)

{

while (!sendCMD("AT+MIPLCREATE"))//判断是否能建立远程实例

{

Serial.println("Failed to create a onenet instance. Try again ...");

sendCMD("AT+MIPLCLOSE=0");//关闭实例

sendCMD("AT+MIPLDELOBJ=0," + OBJ\_ID\_TEMP);

sendCMD("AT+MIPLDELOBJ=0," + OBJ\_ID\_HUMI);

sendCMD("AT+MIPLDELETE=0");

}

// AT+MIPLADDOBJ= <ref>, <objectid>, <instancecount>, <instancebitmap>, <attributecount>, <actioncount><CR>

// <ref> : AT+MIPLCREATE消息发送成功后，返回的OneNET instance值

// < objectid >整型：object id

// < instancecount >整型：instance 数量

// < instancebitmap >二进制字符串：instance bitmap 例: “00101” (5 instances, only instance 1 & 3 are available)

// < attributecount >整型：attribute count (具有Read/Write操作的object有attribute)

// < actioncount >整型：action count (具有Execute操作的object有action)

if (!sendCMD("AT+MIPLADDOBJ=0," + String(OBJ\_ID\_TEMP) + ",1,\"1\",1,0"))//建立温度实例

continue;

if (!sendCMD("AT+MIPLADDOBJ=0," + String(OBJ\_ID\_HUMI) + ",1,\"1\",1,0"))//建立湿度实例

continue;

if (!sendCMD("AT+MIPLOPEN=0,3600"))//连接服务器

continue；

buffer = skipResponse(2);

if (buffer == "" || buffer.indexOf("+MIPLEVENT: 0, 3") != -1)

continue;

buffer = skipResponse(4);

if (buffer == "")

continue;

tmpMsgid = getMsgId(buffer);//将获取的字符串赋给getmsgld函数进行处理，得到misgid

msgid[0] = tmpMsgid;

// AT+MIPLOBSERVERSP=<ref>, <msgid>, <result><CR>

// <ref> : AT+MIPLCREATE消息发送成功后，返回的OneNET instance值

// <msgid>:整型，消息id，+MIPLOBSERVE上报的消息 id

// <result> : 整型, observe请求的结果,( (cancel) observe result, 1 indicates (cancel) observe success)

if (!sendCMD("AT+MIPLOBSERVERSP=0," + tmpMsgid + ",1"))

continue;

buffer = skipResponse(2);

if (buffer == "")

continue;

tmpMsgid = getMsgId(buffer);

msgid[1] = tmpMsgid;

if (!sendCMD("AT+MIPLOBSERVERSP=0," + tmpMsgid + ",1"))

continue;

buffer = skipResponse(2);

if (buffer == "")

continue;

tmpMsgid = getMsgId(buffer);

// AT+MIPLDISCOVERRSP=<ref>, <msgid>, <result>, <length>, <valuestring><CR>

// <ref> : AT+MIPLCREATE消息发送成功后，返回的OneNET instance值

// <msgid>:整型，消息id，+ MIPLDISCOVER上报的消息 id

// <result> : 整型, DISCOVER请求的结果,( 1 indicates read success, should provide read content in the same time )

// <length> : 整型, 响应DISCOVER请求的valuestring

// <valuestring> : 整型, 响应DISCOVER请求的object对应的属性(attribute; …; attribute; action; …; action),

// must start with " and end with "

// attribute + action count in valuestring is the same to attributecount + actioncount in AT+MIPLADDOBJ)

if (!sendCMD("AT+MIPLDISCOVERRSP=0," + tmpMsgid + ",1,4,\"" + String(RES\_ID\_VAL) + "\""))//响应模组下发的diacover//

continue;

buffer = skipResponse(2);

if (buffer == "")

continue;

tmpMsgid = getMsgId(buffer);

if (!sendCMD("AT+MIPLDISCOVERRSP=0," + tmpMsgid + ",1,4,\"" + String(RES\_ID\_VAL) + "\""))

continue;

break;

}

}

void loop()

{

static long tm = millis();

static int count = 0;

static float flag= 0;

static float all=0;

if ((millis() - tm) > 2000)

{ delay(4000);//延时四秒

float h = dht.readHumidity();//六秒采一次样，读取湿度

// Read temperature as Celsius

float t = dht.readTemperature();//读取温度

// Read temperature as Fahrenheit

float f = dht.readTemperature(true);

}

flag+=h;

all+=t;

count++;

tm = millis();//Get Clock Data

}

if (count > 10)

{

all=all/count;//计算一分钟采样10次内温度平均值

flag=flag/count;//计算一分钟采样10次内湿度平均值

sendData(msgid[0], OBJ\_ID\_TEMP, RES\_ID\_VAL, all);

display.update();

delay(1000);

display.clear();

display.setTextSize(1);//设置显示字大小

display.setTextColor(WHITE);

display.setCursor(0,0);//0行0列

display.println(all,1);//显示温度，保留一位小数

sendData(msgid[1], OBJ\_ID\_HUMI, RES\_ID\_VAL, flag);

display.setTextSize(1);//设置显示字大小

display.setTextColor(WHITE);

display.setCursor(32,0);//32行0列

display.println(all,1);//显示湿度，保留一位小数

count = 0;

flag=0;

all=0;

}

}

void serialEvent() //串口中断

{

String buffer = "";

while (Serial.available())

{

buffer += char(Serial.read());

delay(2);

}

if (buffer.indexOf("ERROR") != -1) //返回信息包含ERROR

Serial.println("NOTIFY ERROR");

}

# 

# 致 谢

转瞬间，我在江苏省人民医院为期半年的实习生活已经接近尾声了，在这半年中，我遇到了很多人，很多事，在这些人与事之间，我学到了很多书本上接触不到的知识，看到了一个不一样的世界，不仅学习到了很多专业知识，也学到了不少为人处世的道理，在此我要感谢每一位帮助我的领导、老师和学长。

首先，我要感谢的指导老师：汪缨老师。汪缨老师在这半年时间内于我而言是一位亦师亦父的存在，不论是在生活还是在学习上，都给予我一定的支持与鼓励，不仅为我的人生方向提出了宝贵的建议，在毕业设计上面，也为我提供了很多硬件和软件上的支持，对于设计过程中遇到的问题总是能十分耐心的给予我指导，提供了十分有价值的意见，也因此我才能在规定时间内保质保量的完成设计。

其次我要感谢我的各位学长，孙小磊学长，赵峰学长、徐翔学长、汤福南学长，在实习过程中，各位学长带着我参与了很多设备的维修，为我细心和耐心的讲述了设备维修过程中遇到的问题，讲述了书本与实际操作中的区别与共通点，让我明白了纸上得来终觉浅，绝知此事要躬行；另外诸位学长也为我在人生做人做事上面提出了宝贵的建议。

还要感谢各位实习工作中的遇到的老师：张晖老师、彭茂老师、胡益斌老师、李开良、陶元娟老师，感谢他们在实习过程中对我的帮助；特别的感谢我们的

高处在我毕业求职方面的帮助，和对我们实习生工作生活的关心。

在此，我还要感谢和我一起愉快的度过大学生活各位尊敬的老师，是你们给了我这个实习机会，无私传授我们知识，正是由于你们的支持与陪伴，我才能在一个又一个的困难中走过来，直至本文的顺利完成。

最后我要感谢我的父母，是你们养育了我，用自己的心血栽培我，我知道你们是我们的坚强就后盾，感恩成为你们的子女。

在论文即将完成之际，我的心情无法平静，从开始进入课题到论文的顺利完成，有多少可敬的师长、同学、朋友给了我无言的帮助，在这里请接受我诚挚的谢意!谢谢你们!

# 

# 参考文献

1. [] 钱建国,李维嘉,李斌. 区域性大型医疗设备预防性维护规范探讨[J]. 医疗卫生装 备,2010,31(08):5-6+12. [↑](#endnote-ref-1)
2. [] 邵伟新. 现代医疗设备与环境条件[A]. .中国人民解放军医学会第九届放射诊疗专业学会论文集[C].:,2005:1. [↑](#endnote-ref-2)
3. [] 张连珍,郝萌. 环境温湿度条件对医疗设备运行的影响[J]. 医疗装备,2012,25(01):85-86. [↑](#endnote-ref-3)
4. []房坤,杨婷,徐国庆,陈怀亮,陈玉俊,沈爱宗. 临床医学工程师在MRI设备使用质量监督管理中的职能作用探讨[J]. 中国医学装备,2016,13(07):100-102. [↑](#endnote-ref-4)
5. [5]祝晓悦.窄带物联网(NB-IoT):打造超连接物联网的蜂窝技术[J].中国集成电路,2018,27(05):21-25. [↑](#endnote-ref-5)
6. [] 潘可佳. 基于物联网技术的机房智能监控系统设计与实现[D].电子科技大学,2013. [↑](#endnote-ref-6)
7. [] 林珠妹.基于DHT22的数字式温湿度计的设计[J].佳木斯大学学报（自然科学版）,2016,34(2):223-225. DOI:10.3969/j.issn.1008-1402.2016.02.019. [↑](#endnote-ref-7)
8. [] 杨欣.基于单片机的OLED显示终端设计分析与研究[J].数字通信世界,2019,(1):112. DOI:10.3969/J.ISSN.1672-7274.2019.01.081. [↑](#endnote-ref-8)
9. [] J. M. Dautria,C. Dupuy,D. Takherist,J. Dostal. Carbonate metasomatism in the lithospheric mantle: peridotitic xenoliths from a melilititic district of the Sahara basin[J]. Contributions to Mineralogy and Petrology,1992,111(1). [↑](#endnote-ref-9)
10. [] 侯昌昌,朱玉龙.汇编语言与C语言在编程中的交叉应用[J].安徽工业大学学报(自然科学版),2003,02:148-151. [↑](#endnote-ref-10)
11. [] 张天怡.Java技术的发展趋势与应用分析[J].河北农机,2018,09:38-39. [↑](#endnote-ref-11)
12. [] 陈龙. 智能家居自动控制与监测系统的设计与实现[D].内蒙古大学,2015. [↑](#endnote-ref-12)
13. [] 龙柄全.LabVIEW与ARM嵌入式的结合系统在岩体裂缝检测中的应用[D].长江大学,2018. [↑](#endnote-ref-13)
14. [] 邹跃华.CCD的图像采集系统研究[D].浙江:浙江工业大学,2014. [↑](#endnote-ref-14)
15. []许翔,汪缨.基于物联网的核医学机房环境监测系统设计[J].中国医疗设备,2019,34(1):101-105. DOI:10.3969/j.issn.1674-1633.2019.01.028. [↑](#endnote-ref-15)
16. [] [期刊论文] 张峰,程木华,郑子梅,张勇,吴春兴,陈谊,ZHANG Feng,CHENG Mu-hua,ZHENG Zi-mei,ZHANG Yong,WU Chun-xing,CHEN Yi- 《现代医学仪器与应用》2007年3期 [↑](#endnote-ref-16)