物联网智能宠物看护系统的设计

|  |  |
| --- | --- |
| 学生专业： |  |
| 学生姓名： |  |
| 指导教师： |  |

# 摘 要

如今社会随着科学技术的进步和居民生活水平的提高，更多的人开始选择喂养宠物。然而，宠物喂养也存在不少难题。例如，由于工作、旅游和学习等原因，导致主人不能随时看护宠物，因此，为了解决宠物无人照料的问题，研制了一种能够自动给宠物喂食、喂水并为宠物提供合适的生活环境的宠物家庭系统。

本文旨在设计一种低成本、低功耗的系统，以STM32为核心芯片，用温湿度、压力、水位、光照等传感器来检测宠物所需的喂食、给水以及所处的环境条件，单片机经过与阈值的比较处理，驱动风扇模块，加湿模块，水泵模块，舵机模块，LED灯，使宠物达到理想生存环境。同时利用Wi-Fi技术将宠物喂养数据上传到OneNET物联网平台，由物联网平台将数据传送到手机APP进行数据监测与反向控制，适用于家庭、宠物店等环境。

绘制原理图和制作PCB板使用Altium Designer软件，相关程序使用Keil软件编程调试，使用OneNET物联网平台作为数据上传中心平台。经过调试最终实现设计目标

**关键词：**宠物看护；传感器；物联网平台；Altium Designer；Keil

# ABSTRACT

Nowadays, with the progress of science and technology and the improvement of residents' living standards, more people begin to choose to feed pets. However, there are also many problems with pet feeding. For example, due to work, travel and study, the owner cannot take care of the pet at any time. Therefore, in order to solve the problem of unattended pets, a pet home system that can automatically feed and water the pet and provide a suitable living environment for the pet is developed.

This paper aims to design a low cost, low power consumption system, with STM32 as the core chip, using pressure, water level, temperature and humidity, light and other sensors to detect the feeding and feeding water and environmental conditions required by pets, microcontroller after comparison with the threshold, driving fan module, humidifier module, water pump module, servo module, LED lamp, Make the pet achieve the ideal living environment. At the same time, the Wi-Fi technology is used to upload the pet feeding data to the OneNET Internet of things platform, and the Internet of things platform transmits the data to the mobile APP for data monitoring and reverse control, which is suitable for families, pet stores and other environments.

Altium Designer software was used to draw schematic diagrams and make PCB boards, Keil software was used to program and debug related programs, and OneNET Internet of things platform was used as the data upload center platform. After debugging, the design goal is finally achieved

**Key words:** Pet care，Sensor，Internet of things platform，Altium Designer，Keil

# 目 录

[摘 要 I](#_Toc136854818)

[ABSTRACT II](#_Toc136854819)

[第1章 绪论 1](#_Toc136854822)

[1.1研究背景和意义 1](#_Toc136854823)

[1.2国内外智能宠物看护设备研究现状 2](#_Toc136854824)

[1.2.1国内研究现状 2](#_Toc136854825)

[1.2.2国外研究现状 3](#_Toc136854826)

[1.3设计主要内容 4](#_Toc136854827)

[第2章 总体设计方案 5](#_Toc136854828)

[2.1系统总体结构 5](#_Toc136854829)

[2.2设计方案的选择 6](#_Toc136854830)

[第3章 硬件设计 10](#_Toc136854831)

[3.1 STM32F103C8T6最小系统 10](#_Toc136854832)

[3.2电源部分设计 11](#_Toc136854833)

[3.2.1电源输入电路设计 11](#_Toc136854834)

[3.2.2 WiFi电源转换电路设计 12](#_Toc136854835)

[3.3 WiFi模块部分设计 13](#_Toc136854836)

[3.4传感器部分设计 15](#_Toc136854837)

[3.4.1温湿度检测部分 15](#_Toc136854838)

[3.4.2压力检测部分 16](#_Toc136854839)

[3.4.3水位检测部分 17](#_Toc136854840)

[3.4.4光照检测部分 18](#_Toc136854841)

[3.5显示部分设计 19](#_Toc136854842)

[3.6按键部分设计 20](#_Toc136854843)

[3.7驱动部分设计 21](#_Toc136854844)

[3.7.1光照模块设计 21](#_Toc136854845)

[3.7.2风扇及加水加湿模块设计 22](#_Toc136854846)

[3.7.3投食模块设计 23](#_Toc136854847)

[3.8报警电路设计 25](#_Toc136854848)

[第4章 系统软件设计 26](#_Toc136854849)

[4.1主程序流程图 26](#_Toc136854850)

[4.2 WiFi模块子程序流程图 28](#_Toc136854851)

[4.3传感器部分子程序流程图 29](#_Toc136854852)

[4.4显示子程序流程图 30](#_Toc136854853)

[4.5驱动部分子程序流程图 31](#_Toc136854854)

[4.6物联网云平台设计 32](#_Toc136854855)

[4.6.1 MQQT传输协议 32](#_Toc136854856)

[4.6.2物联网控制台应用程序创建 33](#_Toc136854857)

[4.6.3物联网平台通讯程序设计 35](#_Toc136854858)

[第5章 系统的调试与实验 36](#_Toc136854859)

[5.1电路板的焊接与测试 36](#_Toc136854860)

[5.2电路板上电调试 37](#_Toc136854861)

[5.2.1供电测试 37](#_Toc136854862)

[5.2.2传感器数据采集功能测试 38](#_Toc136854863)

[5.2.3驱动部分功能测试 39](#_Toc136854864)

[5.3 软件部分调试 40](#_Toc136854865)

[5.3.1设备接入OneNET云平台测试 40](#_Toc136854866)

[5.3.2数据流监测 40](#_Toc136854867)

[5.3.3客户端远程控制测试 43](#_Toc136854868)

[第6章 经济分析与环保分析 44](#_Toc136854869)

[6.1经济性分析 44](#_Toc136854870)

[6.2环保性分析 44](#_Toc136854871)

[结 论 45](#_Toc136854872)

[参考文献 46](#_Toc136854873)

[致 谢 49](#_Toc136854874)

[附录1系统硬件电路图 50](#_Toc136854875)

[附录2系统部分程序代码 51](#_Toc136854876)

# CONTENTS

[ABSTRACT(Chinese) I](#_Toc136853611)

[ABSTRACT II](#_Toc136853612)

[Chapter 1 Introduction 1](#_Toc136853615)

[1.1 Research Background and significance 1](#_Toc136853616)

[1.2 Research status of intelligent pet care equipment at home and abroad 2](#_Toc136853617)

[1.2.1 Domestic research status 2](#_Toc136853618)

[1.2.2 Foreign research Status 3](#_Toc136853619)

[1.3 Main contents of Design 4](#_Toc136853620)

[Chapter 2 Overall design scheme 5](#_Toc136853621)

[2.1 Overall System Architecture 5](#_Toc136853622)

[2.2 Selection of Design Scheme 6](#_Toc136853623)

[Chapter 3 Hardware Design 10](#_Toc136853624)

[3.1 STM32F103C8T6 minimum system 10](#_Toc136853625)

[3.2 Power Supply Design 11](#_Toc136853626)

[3.2.1 Power Input circuit Design 11](#_Toc136853627)

[3.2.2 WiFi power conversion circuit design 12](#_Toc136853628)

[3.3 Part design of WiFi module 13](#_Toc136853629)

[3.4 Sensor Design 15](#_Toc136853630)

[3.4.1 Temperature and Humidity Test 15](#_Toc136853631)

[3.4.2 Pressure Detection Part 16](#_Toc136853632)

[3.4.3 Water Level Detection Part 17](#_Toc136853633)

[3.4.4 Light Detection Part 18](#_Toc136853634)

[3.5 Display part Design 19](#_Toc136853635)

[3.6 Key Design 20](#_Toc136853636)

[3.7 Drive part design 21](#_Toc136853637)

[3.7.1 Lighting Module Design 21](#_Toc136853638)

[3.7.2 Design of fan and water humidification module 22](#_Toc136853639)

[3.7.3 Feeding module design 23](#_Toc136853640)

[3.8 Alarm circuit design 25](#_Toc136853641)

[Chapter 4 System software design 26](#_Toc136853642)

[4.1 Main program flow chart 26](#_Toc136853643)

[4.2 WiFi module Subroutine flow chart 28](#_Toc136853644)

[4.3 Molecular program flow chart of sensor part 29](#_Toc136853645)

[4.4 Display Subroutine flow chart 30](#_Toc136853646)

[4.5 Molecular program Flow chart of the driving part 31](#_Toc136853647)

[4.6 Internet of Things Cloud Platform Design 32](#_Toc136853648)

[4.6.1 MQQT Transport Protocol 32](#_Toc136853649)

[4.6.2 Internet of Things Console Application creation 33](#_Toc136853650)

[4.6.3 Design of Communication Program for IoT Platform 35](#_Toc136853651)

[Chapter 5 System debugging and experiment 36](#_Toc136853652)

[5.1 Welding and testing of circuit board 36](#_Toc136853653)

[5.2 Commissioning the Circuit Board 37](#_Toc136853654)

[5.2.1 Power Supply Test 37](#_Toc136853655)

[5.2.2 Sensor Data Collection Function Test 38](#_Toc136853656)

[5.2.3 Driving some Function tests 39](#_Toc136853657)

[5.3 Software Debugging 40](#_Toc136853658)

[5.3.1 Connecting the Device to the OneNET Cloud Platform 40](#_Toc136853659)

[5.3.2 Data Flow Monitoring 40](#_Toc136853660)

[5.3.3 Remote Control Test on the Client 43](#_Toc136853661)

[Chapter 6 Economic analysis and Environmental analysis 44](#_Toc136853662)

[6.1 Economic analysis 44](#_Toc136853663)

[6.2 Environmental Protection Analysis 44](#_Toc136853664)

[Conclusion 45](#_Toc136853665)

[References 46](#_Toc136853666)

[Acknowledgements 49](#_Toc136853667)

[Appendix 1 System hardware circuit Diagram 50](#_Toc136853668)

[Appendix 2 Part of the system program code 51](#_Toc136853669)

# 第1章 绪论

## 1.1 研究背景和意义

首先，随着人们生活，消费水平的提高，越来越多的家庭选择养宠物，这也为宠物市场的发展提供了机会。其次，由于很多场合，比如外出旅游、学习、工作等都无法将宠物带在身边，那么宠物的喂养在无人的情况下就难以得到满足。此外，随着人们对宠物健康和安全的关注度越来越高，他们也更关注宠物所面临的挑战和风险。因此，智能化的宠物管理监测系统可为消费者提供更加便捷和有效的服务。

其次是智能宠物看护设备的市场自我更新需要，随着物联网技术不断发展，相关产品的更新迭代越来越快，以及生活水平的提高，人们对于宠物的关注度也在不断上升，一般的产品不能满足用户的需求。同时也有越来越多的人开始投入到这一领域中来，越来越多的人开始尝试着去养宠物，从而带动了与宠物相关的市场迅速发展，并且还在以极高的速度快速发展与壮大。这就使得智能宠物看护产品在市场中也占据了一席之地。然而，智能宠物看护产品的快速发展也是一把双刃剑，由于市场中还没有足够多的智能化程度高且经济实惠的智能宠物看护设备，因此用户在使用过程中，所体验到的产品使用体验满意度较低，差强人意。

最后就是智能宠物看护设备便于宠物主人管理，物联网智能宠物看护系统可以为宠物主人提供一个便捷的宠物管理平台，可以让他们随时随地地对自己的宠物进行监测和管理，通过手机上的APP等终端设备，宠物主人可以实时了解到他们的宠物的健康状况、喂食等信息，及时采取措施对其进行照顾，同时也可以方便快捷地对其进行检查和诊断，及时发现问题并进行有效处理。能够有效解决目前因主人外出导致的宠物无人照看的问题

综上所述，物联网智能宠物看护系统的研究意义不仅仅在于可以为宠物提供更好的健康管理和喂养服务，还可以为宠物主人提供更为便捷的管理服务，实现更加智能化、自动化和个性化的管理。此外，随着物联网技术的发展，还可以实现智能家居建设，从而为宠物家庭带来更多便利和安全。

## 1.2 国内外智能宠物看护设备研究现状

### 1.2.1 国内研究现状

2017年，东南大学成贤学院电子与计算机学院的邵彦达、沈睿、眭耀宇发表的基于AT89C51单片机的智能宠物家居一文中，对于智能家居的研究中以AT89C51单片机为核心，利用几个I/O口实现对压力传感器的控制来实现对食物重量和饮用水重量的检测，来实现简单的宠物无人喂养，该设计简约方便，成本低，功耗低，且该设计抗干扰性能好，可靠性高[1]。但存在些许缺点，检测参数少，不能充分实现对宠物的喂养，另外不能实现远程监测和控制功能。2019年，周锦荣、张振宇、袁忠强设计的无线物联网控制的视频监控相结合的宠物喂养系统模型，以拓宽宠物智能喂养和管理方法为目的，使用STM32为主控芯片，驱动控制芯片，将以无线网络传感器为核心的物联网技术（IOT）应用到宠物喂食器的设计中，通过CCD传感器与微处理器控制技术，实现对宠物的智能喂食与监控，使宠物的饲养具有连接与感知的功能，从而提高饲养的精度与管理的效率[2]。但是没有遥控，只有喂食和监控的功能。2021年，由吴明泽，胡立夫，胡家瑞，吴晓宇等人联合研发，研制出了一款具有自主喂食、饮水、远程监控、远程控制等多种功能的智能宠物家庭。由STC89C51单片机作为平台，手机客户端和机器人终端构成，机器人终端使用摄影技术对宠物猫狗进行实时的录像，并通过视频通信技术将录好的视频发送到手机的客户端[3]。无线通信以网络为主要的传输媒体，机器人会按照主人制定的饮食规则，在一定的时间内给宠物们分发食物和饮料，并把宠物们的饮食信息发送到手机客户端。用户还可以通过电脑控制机器人上的摄像头，实时监控宠物的状态，保证宠物的喂食。缺点是成本较高，监测数据不全。

国内市场上有着不少宠物喂养设备，在京东APP上搜索查看后发现总体可分为两大类：单一喂食与喂水喂食两大类。其中单一喂食拿小米旗下的产品作为例子，众所周知如今的小米的智能家居生态链做的如火如荼，在宠物家居方面也是同样有所涉及。在京东上的一款产品可以连接米家APP，可随时随地控制出粮时间与数量，为宠物设置全天候定时定量自动喂食计划，放心出行，使喂养更科学更便捷[4]。但是缺点也显而易见：功能单一，仅有喂食功能，且不能监测宠物的生存环境状况。第二类就是既能喂食又能喂水，以联想旗下的产品为例：饮水、喂食、监控三合一减少多台设备的占地面积，让家里更整洁,宠物拥有更大玩耍空间一体化设计,防止宠物打翻。净水污水分离，可称重食盘，智能APP控制，红外高清摄像头，配备多猫识别功能。可谓是功能强大，一千多的价格也是令人望而却步，而且在用户的使用体验中也存在着不少抱怨，产品质量不合格，个别功能无法实现等。用户花了如此大的价钱却得不到相应的体验。总之如今的市场急需一款经济实用的智能宠物家居设备。

### 1.2.2 国外研究现状

现在，国外也在研究智能宠物家庭系统。2016年Kim Seungcheon在《Smart Pet Care System using Internet of Things》一文介绍了一种在物联网环境下工作的智能宠物护理系统。拟建智能宠物护理系统的基本服务包括:远程喂食、遥控自动排便、闭路电视服务、智能手机APP，可提供上述服务的控制信息[5]。自动给料机的基本功能和市场上能找到的产品大多是一样的。区别在于沟通能力和感知能力。设计了带有重量传感器和定时器的自动给料机。它可以测量食物的量和检查时间间隔。并可定时操作，设有定时装置。用户可以为宠物设定定时喂食时间间隔。此外，用户还可以根据一次性食物量的重量来设定食物量。每一个设置的控制都可以通过用户的智能手机完成。采用Arduino单片机实现了自动给料机。用户可以通过开发的特定智能手机APP查看自动给料机的状态。2020年Nenny Anggraini在研究《Mobile-based monitoring system for an automatic cat feeder using Raspberry Pi》中提出了一种基于树莓派的猫咪自动喂食系统，该项目的目标是利用一台网络摄像机，将一台步进马达与一台树莓派主机相连，用以监视猫咪吃食的情况。你可以用一个摄像头来拍照或者录像。步进马达可转动给料阀，经由GPIO管脚，向树莓派中插入程序。接下来，树莓派将连接到互联网和服务器网络，这样就可以通过移动端的网络浏览器或远程网络视图来完成系统控制。该系统的整体功能包括直接或定时给猫喂食，以及监测喂食周围的照片或视频[6]。

国外的实用设备也有不少公司在做，现列举以下几个公司，Petnet：这是一家位于美国的公司，主要生产智能宠物喂食器。它的智能喂食器可以通过智能手机应用程序进行控制和监测，可以根据宠物的体重、年龄、品种和活动水平等信息，自动计算每顿饭的分量和喂食时间；SureFlap：这是一家英国的公司，生产智能宠物喂食器和智能宠物门。它的智能喂食器可以通过微型芯片识别宠物，并根据宠物的食量和饮食需求，自动分配食物；Feed and Go：这是一家位于英国的公司，生产智能宠物喂食器。它的喂食器可以通过智能手机应用程序进行控制，可以设置多个喂食时间和分量，并监测宠物的食量和饮食情况。

整体上分析来说国外的智能宠物喂食产品存在的情况与国内大同小异，功能不够齐全，成本偏高，用户体验不够好。

## 1.3 设计主要内容

《物联网智能宠物看护系统的设计》以STM32单片机为主控，通过光照，温湿度,压力,水位传感器，来检测宠物需要的投食量供水量和适宜环境数据，单片机经过与阈值的比较处理，使驱动加湿模块，风扇模块，喂食喂水模块使宠物达到理想生存环境。同时利用Wi-Fi技术将宠物喂养数据上传到物联网平台，由物联网平台将数据传送到远端App进行数据监测与反向控制，适用于家庭、宠物店等环境。

主要研究内容分为以下三个部分:

1. 硬件系统部分：各传感器模块采集数据发送给主控芯片，根据主控芯片分析结果，控制水泵、风扇、舵机、加湿器等部分。
2. 物联网平台通信部分：与硬件接口部分、利用Wi-Fi模块与物联网平台通信部份、与软件系统接口部分。
3. 软件系统部分：通信的接口部分和客户端App界面部分。

# 第2章 总体设计方案

物联网智能宠物看护系统是对宠物平时的喂养看护进行的设计，本章会对不同的设计方案进行简单介绍与比较并选取最佳的方案来实现设计目标。

## 2.1 系统总体结构

系统总体结构框图如下图2-1所示



图2-1 系统总体结构框图

如图所示，整个设计以STM32F103C8T6为主控芯片，通过温湿度传感器，压力传感器，水位传感器,光照传感器来检测采集宠物的各种数据，然后传输到STM3单片机，经过处理后显示在OLED上，并且与预先设定好的阈值比较后，驱动加湿模块，风扇模块，投食供水模块，光照模块进行相应的动作。同时通过WiFi模块将采集到的各种数据上传到物联网平台，由物联网平台再将这些数据传送到手机APP终端来进行数据的监测与反向控制，完成整体系统的实现[7]。

## 2.2 设计方案的选择

整个系统设计不仅需要主控芯片完成对多个传感器：压力传感器，温湿度传感器，水位传感器，光照传感器采集的信息进行检测和读取并上传到物联网平台，还需要驱动风扇模块，加湿模块，投食喂水模块，光照模块。通过OLED屏幕和WiFi模块与手机连接后实现人机交互，对宠物生活的各项数据进行监测与管理。结合各方案的可行性研究，总结了以下几种主流的嵌入式智能宠物喂养系统的方案。

方案一：基于STC89C51的设计方案：STC89C51是STC MCU系列中的一种单片机芯片。该系统以FLASH核心为核心，运行速度快，运算能力强，外围设备丰富。下面是关于STC89C51的详情：

1. 内存：STC89C51包含32KB的闪存程式记忆体，可用来储存程式码及资料。另外，其SRAM容量为1KB。
2. 外设资源：STC89C51整合了多种外设资源，其中包含四个16位的计时器/计数器，两个串口（UART)，一个SPI总线接口，一个I2C总线接口，一个PCA计时/计数器，以及一个ADC模拟-数字转换器等[8]。
3. 接口：STC89C51的输入输出接口40个，其中常规接口32个，可编程中断接口8个。
4. 时钟部分：STC89C51以12兆赫兹的晶体振荡为主要频率，经锁相环倍频后，其工作频率最高可达72兆赫兹。
5. 电源部分：STC89C51芯片具有低功耗模式，掉电检测，重置电路等多种功率控制功能。
6. 编程方法：STC89C51支持内部系统编程（ISP）的编程方法，也就是说，可以在该系统中对FLASH程序内存进行直接的编程。

总结来说，STC89C51是一款有着非常优秀性能的单片机芯片，它拥有非常丰富的外部设置资源和强大的计算能力，并且还支持ISP编程，因此它可以被广泛地运用到各种应用中去。尽管STC89C51是一种性能优良的MCU芯片，但是STC89C51也有其不足之处，第一存储容量比较小：STC89C51的 flash程序存储容量仅为32KB，相对于现在的单片机芯片来说，稍显不足。第二缺少DMA功能：由于STC89C51没有直接存储器存储器存储器存储器直接存取（DMA）能力不足，造成了数据传输的效率比较低下。第三没有足够的中断向量：STC89C51仅仅支持5个中断向量，相对来说比其他芯片较少，可能会限制设计应用的需求。综上使用STC89C51作为主控芯片的话虽然在程序编程方面较简单，成本较低，但其处理能力有限，在运行多项功能时可能会发生宕机情况，造成芯片发热，数据异常等现象。

方案二：基于树莓派4B的设计方案：树莓派4B是由Raspberry Pi基金会开发的一款功能强大的单板计算机。首次发布于2019年，是树莓派3B+的升级版，提供了更高的性能和更好的扩展性，是一款功能丰富的单板计算机，可以广泛应用于物联网、机器人、智能家居等领域，以下是树莓派4B的主要特点：

1. 处理器：树莓派4B采用Quad-core Cortex-A72 64位ARMv8处理器，主频高达1.5GHz，比树莓派3B+性能提升三倍以上，可轻松运行各种应用程序和多媒体任务。
2. 内存：树莓派4B最高可搭载8GB LPDDR4-3200 SDRAM，比树莓派3B+提升四倍以上，能更快地处理大量数据，并运行更大规模的程序。
3. 存储空间：树莓派4B提供两个micro HDMI接口，支持4K显示输出，还有两个USB3.0接口，可使用更快的存储设备。
4. GPIO：树莓派4B提供了40个GPIO引脚，可用于连接各种传感器、执行器和其他外设。
5. 操作系统：树莓派4B可运行多种操作系统，包括Raspberry Pi、Ubuntu等，并支持Docker和Kubernetes等主流容器服务。

树莓派4B拥有着丰富的功能，但与此同时也存在一些缺点。首先，树莓派4B价格比之前的版本略高，这使得它在同类产品中并不是最为经济实惠的选择。其次，随着处理器性能的提升，整体功耗也有所增加，容易引起散热问题。使用时需要特别注意散热，否则会影响系统的稳定性和寿命。最后，由于硬件配置的变化，一些老的软件和扩展板可能不再兼容，需要重新编译或更新。

方案三：基于MSP430的设计方案：MSP430是由德州仪器（Texas Instruments）公司开发的具有极低功耗，高性能，操作简单的MSP430芯片。MSP430芯片因其独特的结构和优异的计算性能，在嵌入式、物联网等领域得到了广泛的应用。其主要特征如下：

1. 低功耗：MSP430的功耗非常低，它具有多种进入睡眠状态的方式，在低功耗模式下，耗电量可以降低到微安级别，非常适合于需要长时间待机的嵌入式应用。
2. 高性能：MSP430采用16位RISC架构，能够在相对较短的时间内完成复杂的运算，并支持DSP指令集和提供硬件加速器的FRAM存储器。
3. 易于编程：MSP430支持多种编程工具和编程语言，如MSP430-GCC、Code Composer Studio（CCS）和IAR Embedded Workbench等，还可通过USB接口进行在线编程和调试。
4. 强大的外设：MSP430内置了丰富的外设，包括多个定时器、比较器、ADC和DAC等，同时还支持多种通信接口和协议，如SPI、I2C、UART、USB等。

总之，MSP430芯片具有低功耗、高性能、易用等特点，在嵌入式、物联网等领域有着广阔的应用前景。但是也有其缺陷。首先，其处理器运行速度缓慢，仅为25MHz，不能满足高性能计算的需要。其次，一般情况下，MSP430的闪存空间仅为几KB至几十KB，不能满足较大规模的应用。此外，MSP430的外设资源相对较少，例如只有一个UART、一个SPI和一个I2C接口，无法满足复杂应用的需求。

方案四：基于STM32的设计方案：STM32单片机系列是一种以ARMCortex-M核为核心的嵌入式微处理器，这是一个面向高性能，低成本，低功耗，低功耗的嵌入式系统，其特征如下：

1. 较为丰富的外围接口：多个通用的输入输出口（GPIO）、模数转换器（ADC）、数字模拟转换器（DAC）、通用同步/异步收发器（USART）、串行外设接口（SPI）等等，可以满足用户的各种需求[9]。
2. 处理性能强：采用了ARM Cortex-M系列架构，处理性能高，适用于高速数据处理和复杂的算法。
3. 低功耗设计：STM32单片机具有较强的低功率特性，可根据需要自动切换到休眠状态和备份状态，从而达到最小化功耗的目的。
4. 灵活的开发工具和环境：STM32单片机可以使用多种开发工具如Keil、IAR等进行开发，同时也支持多种编程语言，如Assembly、C++等等。
5. 安全性高：STM32单片机具有硬件加密，防篡改，以及内部不掉电操作的保护等先进安全设计。

综上所述，STM32 MCU有较强的处理能力，较低的功耗，丰富的外部接口，灵活的开发方式，以及较高的安全性，可用于多种应用场合。虽然略微存在一些不足，比如需要更多的专业知识，对编程语言能力要求高等，但经过学习都可以克服。

通过以上四种方案的分析与对比，由于设计需要检测宠物各项生活数据，驱动各模块准确运行，与手机APP相接并反向控制等要求，且应具备低成本、可以量产商用的特性，并可以根据方案其他需求扩展相关外设，故选用方案四的基于STM32单片机的设计方案。

# 第3章 硬件设计

本章主要是对硬件部分进行设计，其中包括充最小系统电路，电源电路，WiFi电路，传感器部分，显示部分，按键部分，驱动部分，报警电路的介绍；

## 3.1 STM32F103C8T6最小系统

STM32F103C8T6是一款由意法半导体公司推出的基于Cortex-M3内核的32位微控制器，硬件采用LQFP48封装，属于ST公司微控制器中的STM32系列，它的GPIO有8种模式，包括上拉、下拉输入模式，所以在连接按钮时可以不设计上下拉电阻[10]。

STM32F103C8T6单片机具体参数如下：

1. GPIO：总共有37个GPIO，分别为PA0-PA15、PB0-PB15、PC13-PC15、PD0-PD1。
2. ADC：有2个12bit ADC合计12路通道，外部通道：PA0到PA7+PB0到PB1，内部通道：温度传感器通道ADC\_Channel\_16和内部参考电压通道ADC\_Channel\_17[11]。
3. Timers：4个16bit定时器/计数器，分别为TIM1、TIM2、TIM3、TIM、TM1带死区插入，常用于产生PWM控制电机；2个看门狗定时器（独立看门狗IWDG、窗口看门狗WWDG）；1个24bit向下计数的滴答定时器[12]。
4. 通信串口：2\*IIC，2\*SPI，3\*USART，1\*CAN。
5. 系统时钟：可以使用内部8MHz时钟HSI，最高倍频达到64MHz；也可以使用外部8MHz时钟HSE，最高可倍频可达到72MHz[13]。

STM32F103C8T6的最小系统主要由时钟电路，复位电路两部分组成，如下图3-1所示。其时钟电路包含两个主要部分：外部振荡器和内部时钟源。外部振荡器是用来提供芯片整体的系统时钟，通常使用一个陶瓷振荡器或者晶体振荡器作为外部振荡器，其频率通常为8MHz。此外，还可以连接一个30pF的并联电容以稳定振荡器的工作。

复位电路是一项非常重要的设计，它用于确保芯片在开机或者其他异常情况下能够正确地启动并进入预设状态。包括一个复位电平检测电路和一个RC电路，用于检测系统电源是否正常以及芯片是否工作不正常。当系统电源异常或者芯片出现故障时，复位电路会强制将芯片复位，进入复位状态，执行复位向量指向的地址，重新启动系统。



图3-1 STM32F103C8T6最小系统

## 3.2 电源部分设计

### 3.2.1 电源输入电路设计

电源输入部分考虑使用5V电源输入，一方面满足低功耗安全节能的目的，另一方面满足一些模块需要5V电压供电的需求，而对于STM32单片机和其他后续需要3.3V电压供电的模块，再进行电压转换。这里电源接入使用的是DC-005插座，材料耐高温，价格低廉，性价比高。电源输入电路原理图如下图3-2所示。



图3-2 电源输入电路原理图

### 3.2.2 WiFi电源转换电路设计

由于使用的WiFi模块ESP8266-01S的输入电压要求是3V-3.6V与STM32F103C8T6要求的电压相同，不可以使用电源直接输入的电压，所以这里需要设计一个WiFi电源转换电路将5V电压转为3.3V。

电源转换电路使用稳压器AMS117-3.3，它的稳压调整管是由一个PNP驱动的NPN管组成的，漏失电压定义为：

 (3-1)

*Vdrop*————芯片内部导通时产生的压降;

*VBE*————三极管的基极和发射极之间的电压;

*VSAT*————三极管饱和电压

其基本电压转换公式为：

 (3-2)

*Vin*————输入电压；

*Vout*————输出电压；

*Vadj*————外部校准电压。

其中Vdrop和Vadj在AMS1117的数据手册中提供了标准值[14]。需要注意的是，这里的输入电压和输出电压都是直流电压。如果需要将交流电压转换为直流电压，则需要使用整流和滤波等电路进行预处理。

在电源输入电路中，电容并联可以起到减少电源噪声、过滤掉高频噪声等作用，从而提供更稳定、更纯净的电源给负载电路。当电源由于传输线路、开关电源等因素导致的干扰噪声时，加入电容可以削弱这些噪声，起到去除干扰的作用。另外，电容并联还可以对电源电流进行平滑，防止瞬间电流过大造成电源电压下降，从而保证负载电路的正常工作。

具体来说，电容的充放电过程是一个相位相反的过程。当电源电压下降时，电容器可以通过释放其储存的电能，为负载电路提供额外的电流，从而稳定输出电压；当电源电压上升时，电容会吸收过剩的电流，防止过压，保护负载电路。转换电路的原理图如下图3-3所示。



图3-3 WiFi电源转换电路

为了保证AMS1117工作的稳定性，使用可调电压版本时，需连接至少一个22μF的钽电容至输出端，而固定电压版本则可使用较小的电容，具体容量还需据实际应用而定[15]。一般而言，随着输出电流增加，线性调节器的稳定性会降低。

## 3.3 WiFi模块部分设计

WiFi模块部分计划采用ESP8266-01S作为首选，可以用于快速连接互联网的物联网设备和小型嵌入式系统。ESP8266由中国乐鑫公司推出，是一种低功耗、低成本、高性能的系统级芯片，可在-40℃~125℃工作。选择使用ESP-01S型模块，它可以采用串口的无线AP（COM\_AP）模式、串口的无线STATION（COM\_STA）模式、串口的无线AP+STA（COM\_AP&STA）模式，并且还具有一键配网（smartConfig）功能[16]。

ESP8266-01S这是一个类似于ESP-01的无线网络模块，但是在原来的基础上得到了更新和改善。本模组自带Wi-Fi功能，可与Wi-Fi直接联接，是一款性价比很高的物联网装置。ESP-01S具有八个插脚，其中两个插脚（VCC,GND)，两个TX/RX通讯插脚，四个GPIO插脚。电路原理图如下图3-4所示。各引脚功能介绍如下：

1. VCC：支持3.0V至3.6V电压输入的电源供电引脚。
2. GND：接地引脚。
3. CH\_PD：芯片使能脚，用于将模块从低功耗模式的状态唤醒。
4. RST：重置引脚，将引脚电平拉低即可重置该模块。
5. TXD：串行通信发送数据，可以通过AT指令或固件程序来进行控制。
6. RXD：串行通信接收数据，可以通过AT指令或固件程序来进行控制。
7. GPIO0：通用的输入/出引脚，可控制LED和其他外部设备。
8. GPIO2：通用的输入/出引脚，可控制LED和其他外部设备。



图3-4 ESP8266-01S电路原理图

需要注意的是，在ESP8266ESP-01S通电后，系统将自动切换到串行烧录模式。如果要使固件程序正常运行，则必须同时拉高GPIO0和RST管脚（也就是与VCC连接）。

ESP8266-01S具有COM-AP模式，COM-STATIOM模式，以及COM-STA+AP模式，每一种模式都有其独特的作用和适用场合。其中，串口型的Wireless AP（COM\_AP）模式，以该模块为热区，利用其它设备对ESP8266-01S模块进行访问，并利用串口进行信息交换。串口无线STA（COM\_STA）模式则是将模块作为客户端，用于将模块接入其他热点来构建WiFi网络。而串口无线AP+STA（COM\_AP&STA）模式则是既可以将模块作为热点，供其他设备访问，又可以接入到其他WiFi网络，它是前两种模式的综合[17]。

在开始初始化的时候，单片机通过串口通信的方式，模块的最初配置和发送命令的操作是利用AT指令来完成的，其中的一些AT指令见表3-1。

表3-1 AT指令

|  |  |
| --- | --- |
| AT指令 | 描述 |
| AT | 测试AT启动 |
| AT+RST | 重新启用模块 |
| AT+GMR | 查看版本信息 |
| AT+GSLP | 开启深度睡眠功能 |
| ATE | 开启和关闭回显功能 |
| AT+RESTORE | 恢复原出厂设置 |
| AT+UART | 设置串口的配置 |
| AT+CWMODE=1/2/3 | 选择不同的工作模式 |

## 3.4 传感器部分设计

### 3.4.1 温湿度检测部分

温湿度检测部分利用DHT11温湿度传感器对宠物生活环境进行温度和湿度的监测。常见的DHT11有以下4个引脚，电气原理图如下图3-5所示。

1. VCC：传感器的正极供电引脚，工作电压的范围是3V到5V。
2. GND：传感器的负极供电引脚，需要进行接地。
3. OUT：传感器输出的数据信号引脚，可以通过单片机进行采集并处理。
4. RES：传感器的复位引脚，将引脚电平拉低一段时间即可对进行复位。



图3-5 DHT11电气原理图

应当指出，DHT11的数据信号管脚仅能进行数字输出，而不能进行模拟输出。当读取传感器的数据时，特殊的通信协议是必需的。DHT11在单片机能够读取一个确认信号时，就发出一个80微秒的高电平信号，并以50微秒的时间间隔将数据上传至MCU。湿度计算公式为：

 (3-3)

*RH*——湿度值；

rh高——湿度数据高8位；

rh低——湿度数据低8位；

温度计算的方法同湿度的计算方式相同。基于此，可以读取此时的温湿度信息。在此状态下，传感器开启一次温湿度采集，然后通过ESP8266上传到物联网云平台。当采集完后，DHT11自动切换到低功耗模式[18]。

### 3.4.2 压力检测部分

HX711是一款24位模数转换芯片，用于高精密载荷传感器，它的特点是高集成度，高响应速度，高抗干扰能力。HX711在读出感测器数据时，使用差分输入方式，能有效地排除外界的干扰，使感测器的工作更稳定、更准确。支持两种工作模式：编程模式和工作模式。编程模式是用于初始化和校准HX711的内部参数，工作模式是用于读取外部传感器的数据。在使用HX711时，首先需要与外部传感器相连接。通常情况下，HX711需从外界传感器接受正、反两种输入讯号，才能对输出讯号进行差分读取。之后，经由PD\_SCK (引脚选择位）、DOUT(数据输出管脚）与单片机进行SPI通讯，以实现数据的读出。

计算传感器供电电压：HX711可以在产生VAVDD和AGND电压，即HX711模块上的E+和E-电压[19]。通过下式（3-2）计算：

 (3-4)

*VAVDD*——内部稳压电路的输出电压；

*VBG*——芯片的输出参考电压

例如：VBG为2.5V，R1=10K，R2=5.2K，因此得出VAVDD=7.31V。

测重原理：满量程输出电压=激励电压\*[灵敏度](https://so.csdn.net/so/search?q=%E7%81%B5%E6%95%8F%E5%BA%A6&spm=1001.2101.3001.7020)1.0mV/V。例如：供电电压3.3V乘以灵敏度1.0mV/V=满量程3.3mV，相当于有3.3Kg重力产生时候产生3.3mV的电压[20]。

HX711该芯片需经四根管脚与单片机相连。其中，两个引脚为通讯用，另两个引脚为接收换能器的模拟电信号。原理图如下图3-6所示。

1. VCC：为芯片提供3.3V或5VDC供电的数字电源管脚。
2. GND：地线管脚，要求与该系统的地线相连。
3. SCK：MCU用于控制晶片内变换时序的时钟线路。
4. DOUT：在操作方式中，从其输出芯片测得的ADC值（24位）的数据输出管脚。



图3-6 HX711原理图

### 3.4.3 水位检测部分

液位传感器可采用物理接触法、压力传感法、声纳法等多种方法对液位进行检测。在此设计中，使用了Water Sensor作为水位感应器，该感应器是一种用于探测水位或水滴的辨识感应器，使用方便，性能价格比高。

Water Sensor有两个电源引脚VCC和GND，以及一个输出引脚：将检测到的水分信号转化为数字或模拟电信号输出，可以通过该引脚读取传感器检测到的水分含量数据。计算公式如下：

 (3-5)

*h*——表示液位高度；

*C*——表示传感器测得的电容值；

*C0*——表示传感器在空气中的电容值；

*k*——表示传感器的灵敏度常数。

Water Sensor的检测原理是通过一系列的暴漏的平行导线线迹对水滴或者水量的大小进行测量进而判断水位[21]。在水与传感器的电极相接触时，将产生一个回路，使电流通过该回路，再回到控制系统中。该模块主要利用的是三极管的电流放大原理，当液位高度使三极管的基极与电源正极导通时，在三极管的基极和发射极之间就会产生一定的电流，这时在三极管的集电极和发射极之间就会产生一个一定放大倍数的电流，通过发射极的电阻产生特征电压，被AD转换器采集[22]。完成水位到模拟信号的转换之后，可以直接将输出的模拟信号连接至单片机，实现对水位进行测量。该传感器模块的电气原理图如下图3-7所示。



图3-7 水位检测原理图

### 3.4.4 光照检测部分

光照传感器（Light Sensor）是一种对外界光线进行检测的光电传感器。他们能侦测到光的出现以及光的明度，然后把这些量度转换成电信号，以便电脑或其它装置来读取。此部分采用光敏电阻5528，这是是一种根据内光电效应，把光照射到敏感表面上，使电阻值变化的半导体元件。光照越强，阻值就越低，随着光照强度的提高，电阻值急剧下降，甚至可小到1Ω以下[23]。因此，这种光敏电阻应用于光电传感器时，可根据光亮的强弱来进行控制。

单片机的PA5引脚用于AD采集，又因为STM32单片机的电压限制不能高于3.6V，而5528的电压为5V，所以为了使用两个电阻起分压作用来保证电压不超过单片机的正常工作电压。电气原理图如下图3-8所示。



图3-8 光照检测部分原理图

光敏电阻的阻值用*R5528*表示，PA5的电压用VAD表示，R11阻值为1000Ω，R12阻值为1500Ω，则可得下式：

 (3-6)

单片机的正常工作电压为3.3V，ADC最大值为4096，将光敏电阻实时对应的AD值用*X*表示则可得下式：

 (3-7)

然后将式（3-4）与式（3-5）联立可得：

 (3-8)

通过ADC可以计算出*X*，也就可以根据上式求出光敏电阻的值，根据不同的电阻便可得到不同的光照强度大小。

## 3.5 显示部分设计

OLED屏幕具有高对比度、自发光、薄厚度、无需背景光源、宽视角、快速响应的优点。分为七针的SPI/IIC兼容模块，以及四针的IIC模块，设计中使用的是的四针的IIC模块。以现在的科技水平，OLED的体积还无法扩大，但分辨率却是可以达到的，这种屏幕的特点如下：

1. 0.96寸的发光二极管，有黄色、蓝色、白色三种色彩。
2. 分辨率为128x64。
3. 多种接口方式；OLED裸屏一共有以下种接口，分别是：6800,8080两种并行接口方式，3线或4线串行SPI接口模式，IIC接口模式（只要两条线路就能对OLED进行控制)，这五种接口是通过屏上的BS0-BS2来配置的[24]。

原理图如下图3-9所示。0.96英寸4针OLED有四根管脚，分别是正极（VCC）、负极（GND）、时钟线（SCL）、数据线（SDA)，其中VCC和 GND管脚是用来连接电源的，通常是3.3伏或者5伏。其中，SCL、SDA管脚分别为两条IIC接口的信号线路，用以与单片机进行通讯，完成对OLED器件的控制及数据传送。



图3-9 OLED原理图

## 3.6 按键部分设计

按键部分采用独立按键，是一种常用的电子器件，在电子器件中用来进行输入控制。它一般由一个按钮和连接到该按钮的两个引脚组成，当使用者按下按钮时，电路会接通，执行相应的动作。

独立按键个数很少，可以按照要求灵活地编写代码。而在矩阵式键盘中，键的位置则由行、列编号来决定。独立按键的工作原理比较简单，当按键被按下时，会断开引脚间的电路连接，并让单片机探测到电路中的变化，进而在程序中进行相关的控制与操作。

在利用STM32单片机进行按键转态读取时，将与按键相连的STM32单片机接口分配为1，若按下按键，因为按键的另一端接地，则该管脚被拉低。下图3-10为其原理图。



图3-10 独立按键部分原理图

共使用4个独立按键，这里命名为SW2、SW3、SW4、SW5。设计的功能分别为：用SW2控制模式的切换，即对温度，湿度的阈值进行设置；SW3按下为加；SW4按下为减；SW5为手动控制灯光的开关。

## 3.7 驱动部分设计

### 3.7.1 光照模块设计

光照部分计划使用LED灯来模拟宠物的灯光照射，有两种模式，第一种是自动模式灯光常亮，但是会根据光照传感器检测到的光照强度变换灯光的亮度，光照强度大即环境比较亮的时候，LED灯亮度较弱，当光照强度小即环境比较暗的时候，LED灯会随之变亮；第二种是手动模式，可以通过开关SW5来控制LED灯的亮灭，比如宠物需要休息时，可手动关闭灯光。

使用一个NPN型三极管（FHT9014Y-ME），为使NPN型三极管工作，需在其发射结加正向电压，集电结加反向电压的条件下才能达到导通电压[25]。发射结正偏表现为P端电压高于N端电压（需有适当的电压差），而集电结反偏表现为集电结电压高于基极电压。



图3-11 光照电路原理图

上图3-11为光照电路的原理图。当外部光线强度较弱时，光敏电阻的阻值较大，其两端电压升高，导致三极管基极电压上升，未能达到发射结正偏的条件，此时三极管集电极无电流通过，因此二极管不会发光；相反，在较强的光照下，光敏电阻阻值较小，其两端电压降低，基极电压随之降低并满足发射结正偏条件，此时集电极电压基本等于电源电压，高于基极电压，同时满足了集电结反偏条件，因此形成放大电路，有电流通过，二极管发光。并且光敏电阻阻值的变化，引起电压的变化，即LED灯光的亮度也会随之改变。

### 3.7.2 风扇及加水加湿模块设计

对于此部分计划使用三个继电器，来对风扇模块、加湿模块、加水模块进行模拟，并且可以直接在继电器的另一端接上相应的设备如：风扇，加湿器，水泵，继电器相当于开关使另一端对应的风扇、加湿器，水泵开始工作。

通过单片机接继电器电路，可以实现弱电对强电的控制。这种控制方式常见于工业自动化领域，如机器人、汽车生产线等的控制。通过单片机管脚的高低电平输入，可以实现对大电流负载的控制。

原理图如下图3-12所示。具体来说，MCU的一个输出端口连接到继电器控制电路中的控制端口，当该端口输出高电平时，继电器线圈不被激活，继电器处于关闭状态；当该端口输出低电平时，继电器线圈被激活，继电器吸合，处于闭合状态。这种电路可用于控制灯光、电机、风扇等大电流负载的开关。同时，接入LED灯能够非常直观地展示继电器的闭合情况，方便用户观察判断。



图3-12 继电器电路原理图

继电器驱动有两个主要含义：首先是对继电器本身进行驱动，因为继电器是一个功率器件，需要使用单片机来控制它的开/关状态；其次是利用其去驱动其他大功率负载，比如直接驱动接触器或者中间继电器。因此，继电器驱动不仅是单片机与外部大功率负载的接口，也是实现弱电信号控制强电负载的重要途径[26]。

### 3.7.3 投食模块设计

投食模块的设计采用一个舵机模拟宠物粮仓的开启和关闭，即当称重模块采集到食物重量不足时，经过单片机处理后发出信号，控制舵机旋转一个角度，表示进行投食。经常使用的标准SG90舵机共有三根接线：电源导线 VCC，接地导线GND，控制信号导线。其原理图如下图3-13所示。



图3-13 舵机原理图

舵机使用PWM信号进行驱动，其周期一般为20毫秒（50Hz），脉宽的范围为0.5毫秒到2.5毫秒。在每个周期内，舵机接收到一段PWM信号，通过信号的脉宽来确定需要转动的角度。理论上，脉宽为0.5毫秒对应着0度角，脉宽为2.5毫秒对应着180度角，中间的脉宽则会对应着这两个极端之间的角度值。不同脉冲舵机对应旋转角度如下图3-14所示。



图3-14 不同脉冲舵机对应旋转角度

单片机可以接收来自信号线的PWM控制信号，接着通过内部逻辑电路处理并输出控制信号，来控制舵机电机的转动。同时，电机驱动着齿轮组，经过减速后传动至输出的舵盘。此外，舵盘的输出轴与位置反馈电位计相连。在舵盘进行旋转时，位置反馈电位计也随之旋转，并输出电压信号给控制电路板进行反馈。控制电路板根据位置反馈信号来确定电机的转动方向和速度，从而使目标停止[27]。

## 3.8 报警电路设计

报警电路的设计目的是当宠物余量不足时发声警示。其电路主要由一个蜂鸣器，一个NPN三极管，一个隔离二极管以及一个偏置电阻组成。首先选择一个合适的蜂鸣器，它应该能够产生足够大的音量和频率来引起人们的注意。然后将蜂鸣器的正极连接到一个隔离二极管，以防止反向电流损坏电路。连接一个NPN三极管，将其基极与信号源即单片机的PA8引脚连接，中间接入偏置电阻R16进行保护，收集极与地线连接，发射极与蜂鸣器的负极相连，用来作为开关元件，将输入信号转换为输出信号。在报警电路中，三极管开关可以控制输出负载的通断，从而实现报警功能。报警电路如下图3-15所示：



图3-15 报警电路原理图

三极管的基极保持一个稳定的高电平，当PA8处于高电平状态时三极管反向截止，蜂鸣器无电流通过，不工作。当PA8处于低电平时，三极管导通，蜂鸣器通过电流形成回路发出声响开始工作，直到PA8再次拉高时停止报警。

# 第4章 系统软件设计

本章是对软件部分进行设计，主要是主程序流程设计，以及WiFi，传感器，OLED显示，驱动，物联网平台子程序的设计。

## 4.1 主程序流程图

本程序的主程序流程图如下图4-1所示。



图4-1 主程序流程图

软件主要实现如下功能：

1. 实时检测宠物环境温度，湿度，光照强度。
2. 通过独立按键设置适宜的温度和湿度阈值，当温度高于阈值时打开风扇降温，当湿度低于阈值时，打开加湿器加湿。
3. 根据环境亮度自动调节灯光亮度，也可以手动关闭灯光。
4. 监测宠物粮食余量，当宠物粮食不足时发出报警。
5. 监测宠物饮水机水量情况，当水量不足时自动加水。
6. 使用手机APP查看数据。网页下发指令控制。

程序开始后主控以及OLED屏幕初始化，LED初始化，按键初始化，水位初始化，舵机初始化，DHT11初始化，BEEP初始化，WiFi初始化最后进行称重模块初始化，并且对涉及的变量进行定义。然后判断WiFi模块的工作模式，默认为自动模式即ESP8266接受信号“I”时，当接受到信号“J”时开启手动模式；在自动模式下，各传感器将开始工作，采集信息数据并显示在OLED屏幕上。同时，数据也会被上传到物联网云平台。接着再次判断工作模式，为手动模式时则根据接受到的信号进行相应的动作；，否则直接将数据与所设阈值进行比较，然后判断是否通过按键改变阈值，没有则单片机发出信号使驱动模块进行相应的动作有则再次比较。当工作模式为手动时，各驱动模块根据下发的信号进行相应的动作，信号指令如表4-1所示。然后各传感器工作，采集的数据显示在OLED屏幕上，接着判断工作模式以此循环。

表4-1 信号指令表

|  |  |
| --- | --- |
| 信号指令 | 相应动作 |
| A | 风扇开启 |
| B | 风扇关闭 |
| C | 加湿器开启 |
| D | 加湿器关闭 |
| E | 水泵开启 |
| F | 水泵关闭 |
| H | 舵机右转动90° |
| G | 舵机左转动90° |
| I | 自动模式 |
| J | 手动模式 |

## 4.2 WiFi模块子程序流程图

如图4-2所示，首先，进行WIFI模块初始化并设置为STA模式工作，在该模式下ESP8266通过接收路由器的信号而能够连接互联网，如果把它安装在硬件设备上，就可以实现硬件设备的远程控制，可以理解为此时ESP8266是主动去连接其他设备[28]。判断是否连接成功，成功则进入下一步；若未连接则循环执行此步骤。直到连接成功，云平台的连接流程同上，云平台连接成功后，检查是否有数据收发任务。如果有，则执行收发任务；如果没有，则进行热点与云平台连接状态检查的循环。



图4-2 WiFi子程序流程图

## 4.3 传感器部分子程序流程图

按照总体方案设计，传感器信息采集与传输分为多个步骤。如图4-3所示，首先进行温湿度传感器数据采集，然后进行液位传感器信息采集，接着是光照传感器信息采集，最后进行压力传感器数据采集。采集到的数据由信息采集芯片对其进行处理，再由串行接口传送给通信芯片。然后，ESP8266模块会把资料传给OneNET的物联网云端，也会传给移动端的应用程序，实现数据的远程监测与管理。



图4-3传感器子程序流程图

## 4.4 显示子程序流程图

显示部分采用了0.96英寸的OLED显示屏来显示本地数据，并使用IIC控制方法进行数据通讯。当系统通电后，STM32单片机首先需要将与OLED相连接的引脚设置为IIC功能。然后，STM32单片机需要初始化OLED的寄存器，包括地址、片选等寄存器的配置。在初始化完成后，单片机会清空屏幕并开始展示所需的字符，这些字符包括温度湿度名称、亮度、水位、余量，在展示所有固定信息后，STM32单片机会显示每个器件的状态信息，对应于相应的字符。此时，STM32单片机已经成功接收到了每个器件的状态信息。详细的控制流程可参考图4-4。



图4-4 OLED显示子程序流程图

## 4.5 驱动部分子程序流程图

驱动部分程序流程图如下图4-5所示，程序首先进行初始化，如果WiFi模块连接失败则继续初始化，反之连接成功则将各传感器采集到的数据打包发送在OLED屏幕上显示。接着判断工作模式若为手动则根据下发信号控制各驱动模块工作；自动模式下则与阈值进行比较进而通过单片机控制各驱动执行相应命令。



图4-5 驱动部分子程序流程图

## 4.6 物联网云平台设计

### 4.6.1 MQQT传输协议

MQTT是一种以TCP/IP为基础，由IBM于1999年推出的一种基于发布/订阅的轻量级通信协议，现已更新到v3.1.1版本。在MQTT协议的通讯过程中，主要有三个角色参与：发布者（Publish）、代理/服务器（Broker）以及订阅者（Subscribe）[29]。

信息的发布方和订阅方均为客户端，而信息代理则由服务器来提供。同时，信息的发布方也可能是用户本身。通过MQTT协议实现，消息发布者将消息发送到服务器，服务器将消息传给订阅者，从而实现了消息的传输[30]。如下图4-6所示



图4-6 MQQT协议示意图

使用MQTT方式进行数据传输时，当使用的设备接入WiFi网络后，STM32单片机作为客户端会主动连接MQTT服务器。下表4-2为MQQT控制报文类型。

客户端在向服务器发送“CONNECT”报文之前，客户端需要完成一个关键步骤即连接入口的验证过程。一旦服务器确认连接有效，则会返回一个“CONNACK”报文。客户端可以是发布者也可以是订阅者，在实现数据通信前都需要连接到服务器，并完成“CONNECT”和“CONNACK”两个报文的交换。当客户端与服务器成功建立连接后，客户端的联机状态仍然会被客户端与服务器用来确认。

客户端必须周期性地将“PINGREQ”心跳数据包发送到服务器，以指示客户端的联机状态。一旦接收到一个“PINGREQ”的心跳数据包，服务器就会回复一个“PINGRESP”。如果一个客户端没有按时发送一个“PINGREQ”心跳数据包，那么这个客户端就会被服务器所切断[31]。

表4-2 MQQT报文控制类型

|  |  |  |  |
| --- | --- | --- | --- |
| 名字 | 值 | 报文流动方向 | 描述 |
| Reserved | 0 | 禁止 | 保留 |
| CONNECT | 1 | 客户端到服务端 | 客户端请求连接服务端 |
| CONNACK | 2 | 服务端到客户端 | 连接报文确认 |
| PUBLISH | 3 | 两个方向都允许 | 发布消息 |
| PUBACK | 4 | 两个方向都允许 | QoS1消息发布收到确认 |
| PUBREC | 5 | 两个方向都允许 | 发布收到 |
| PUBREL | 6 | 两个方向都允许 | 发布释放 |
| PUBCOMP | 7 | 两个方向都允许 | QoS2消息发布完成 |
| SUBSCRIBE | 8 | 客户端到服务端 | 客户端订阅请求 |
| SBBACK | 9 | 服务端到客户端 | 订阅请求报文确认 |
| UNSUBSCRIBE | 10 | 客户端到服务端 | 客户端取消订阅请求 |
| UNSUBACK | 11 | 服务端到客户端 | 取消订阅报文确认 |
| PINGEEQ | 12 | 客户端到服务端 | 心跳请求 |
| PINGRESP | 13 | 服务端到客户端 | 心跳响应 |
| DISCONNECT | 14 | 客户端到服务端 | 客户端断开连接 |
| Reserved | 15 | 禁止 | 保留 |

### 4.6.2 物联网控制台应用程序创建

登录OneNET官网注册自己的账号，然后进入到平台的开发者中心选项。点击控制台的多协议接入即可[32]。具体的添加产品如下图4-7所示。在完成创建产品后，还需要在该产品下创建设备，以对接硬件数据消息。

每个设备都有一个唯一的标识符（ID），可以使用鉴权码来确保它们之间的相互独立性。在创建一个产品时，需要填写名称、鉴权信息等信息，其中鉴权信息是用于验证设备或应用程序访问物联网平台的凭证。通过鉴权信息，可以对设备或应用程序进行身份验证，从而能够一定程度上保证数据的安全性和可靠性。

在本次设计中，设备名称为petcare，鉴权信息为dht11。设备详情信息如下图4-8所示。

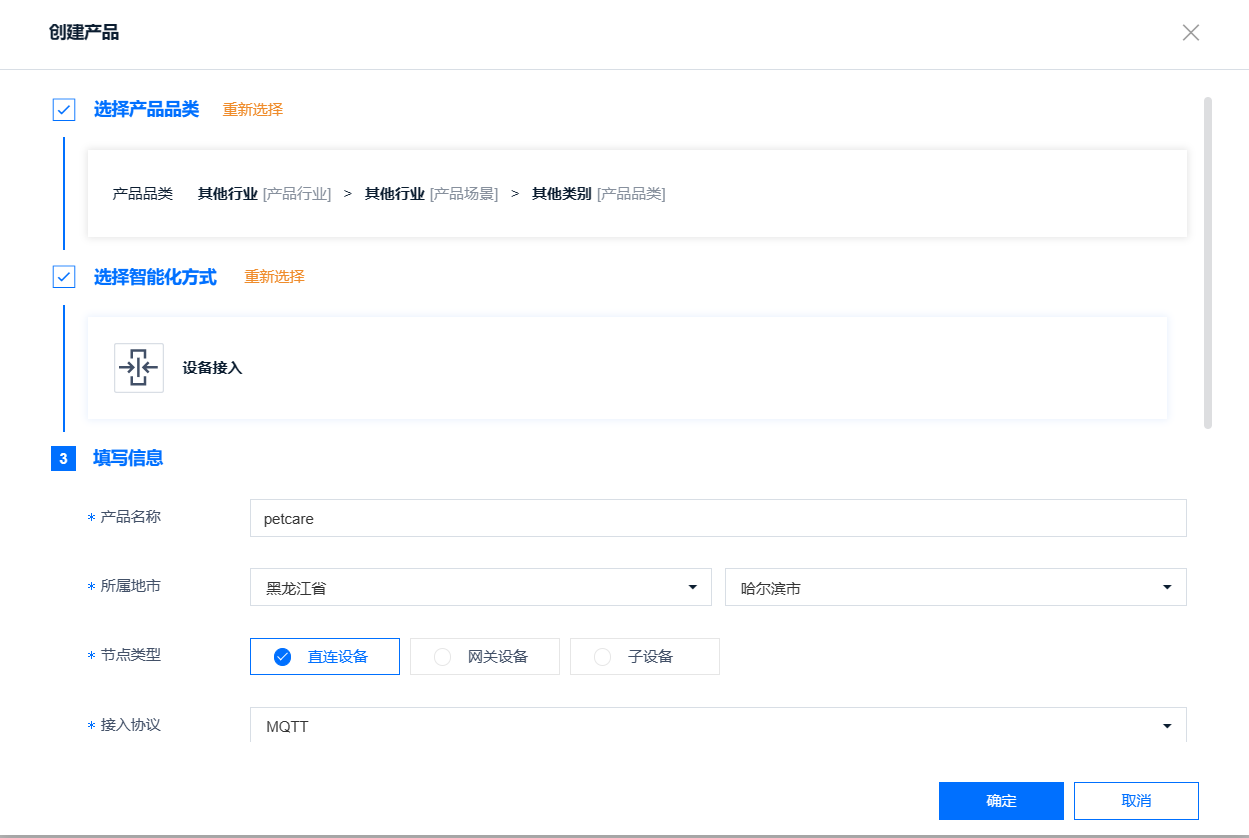


图4-7 创建的产品信息



图4-8 加入设备信息

### 4.6.3 物联网平台通讯程序设计

如图4-9所示，物联网平台的通信流程如下：首先，任务初始化，然后检查ESP8266是否成功连接到OneNET物联网云平台；若连接成功，则执行下一步，否则循环检查直到成功为止。接着判断是否已登录云平台，如果失败则执行登录操作，并再次检查是否已登录。若成功登录，则检查是否收到了来自平台的数据。如果有数据，则解析并执行指令；如果没有数据，则检查是否需要将数据发送到云平台。如果需要，则执行“发送数据到云平台”指令；如果不需要，则返回到WiFi检查，并重复上述流程。



图4-9 物联网平台通讯子程序流程图

# 第5章 系统的调试与实验

本章为系统的实物制作与调试，主要是对PCB板的焊接与调试，以及软件部分的调试，硬件软件相互验证，以达到预期目标。

## 5.1 电路板的焊接与测试

电路板是控制器顺利运行的基础，也是所有元器件的支撑平台。电路板上的器件焊接质量是确保系统正常运行的前提。如果电路板上的器件焊接出现错误，那么下一步电源电路的测试就可能会对控制器造成不可逆转的损伤。因此，将电路板测试放置在整个测试流程的第一步非常重要[33]。

首先将制作好的PCB板与绘制好的原理图进行对比，使用万用表的测电阻功能对各个模块的引脚进行测试，确保连接无误。同时使用万用表仔细检查各焊点避免焊接过程中出现虚焊，缺焊的失误。另外还可以使用万用表在不供电的前提下检查各元件的正负极是否焊接正确。

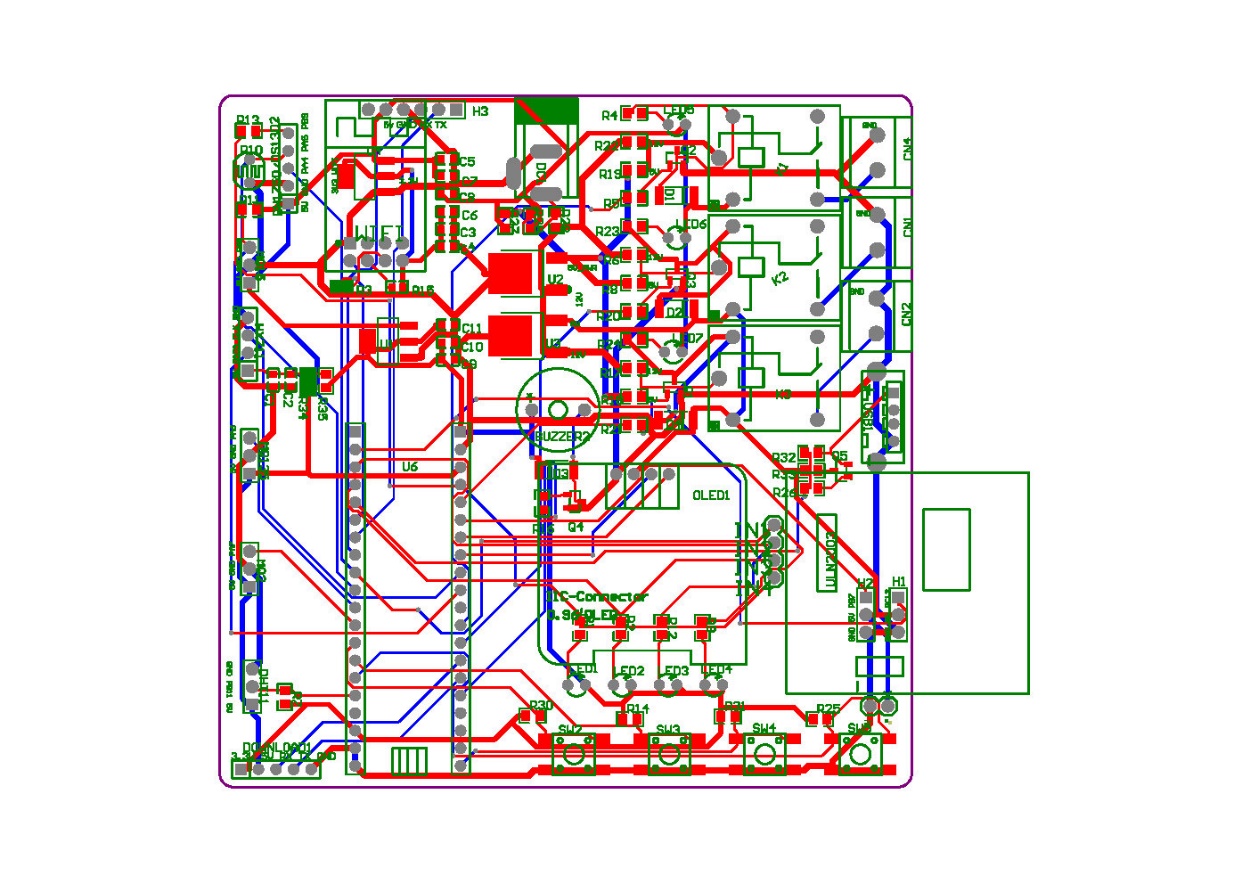


图5-1 PCB设计图

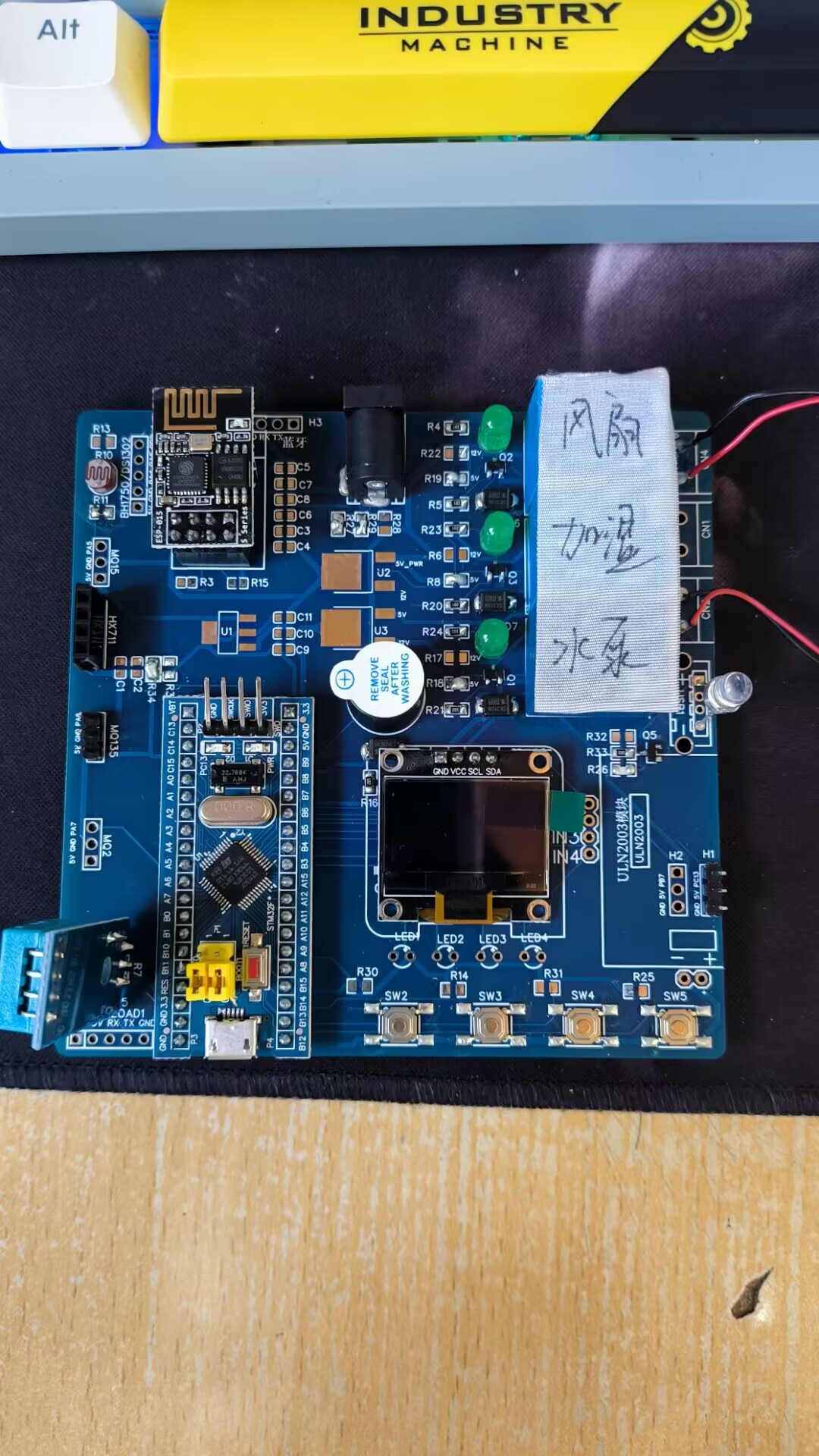


图5-2 PCB实物图

## 5.2 电路板上电调试

### 5.2.1 供电测试

硬件测试主要涵盖以下方面：OLED显示屏、WiFi模块、传感器数据采集模块、阈值设置和超出阈值后各驱动模块是否能按照设计要求驱动功能验证。

首先先进行电路板供电测试，为了避免线路短路损坏元器件，已经在上一步进行过线路检查。现在，将使用充电宝的5V供电口进行供电，并利用AMS1117稳压芯片将5V电压转换为3.3V电压[34]。同时，使用万用表对电路电压进行测量。当供电后电路板上各元器件工作正常。STM32单片机供电理论电压3.3V，实际测量电压3.29V，测量结果在误差范围之内，满足设计要求。测量结果如下图5-3所示。

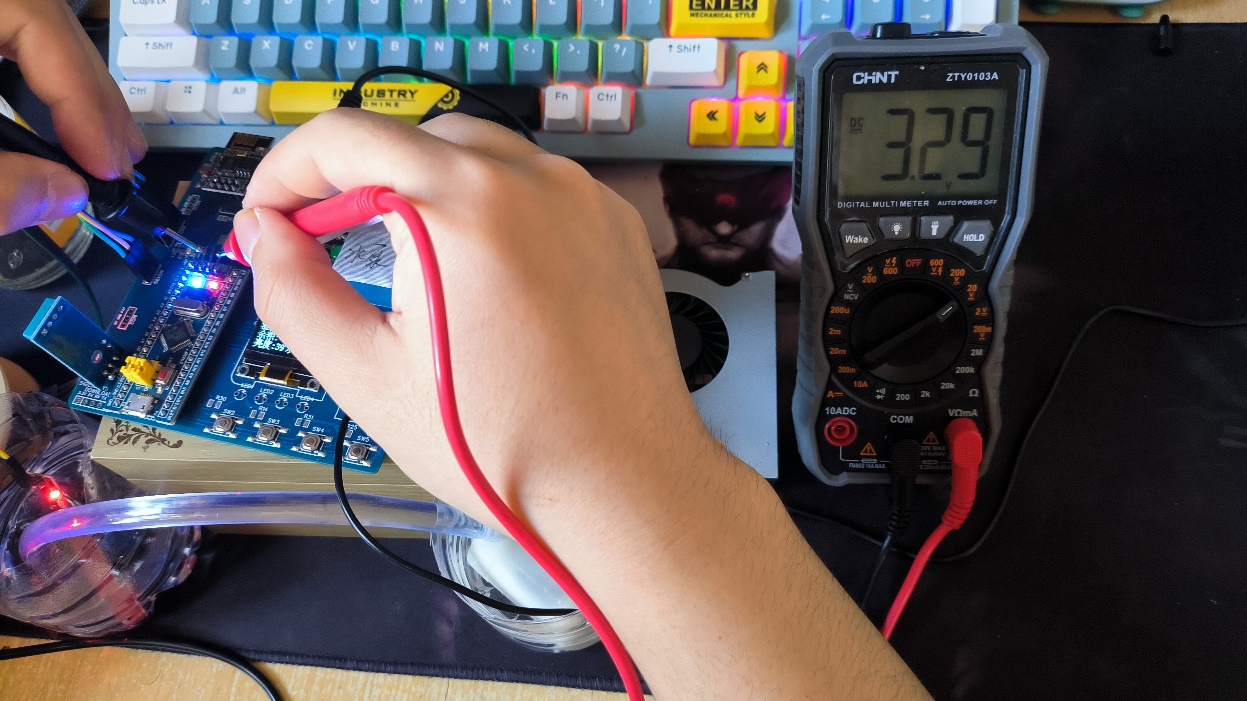


图5-3 供电测试

### 5.2.2 传感器数据采集功能测试

整个系统设计共包含以下几种传感器：温湿度传感器，水位传感器，压力传感器，光照传感器。在系统完成上电工作并运行一段时间后，各传感器将采集环境温度，湿度，水位，食物重量，光照强度等信息传送到物联网云平台，然后传输到手机端APP实时监测，同时也将在OLED屏幕上显示。测试结果如下图5-4所示。

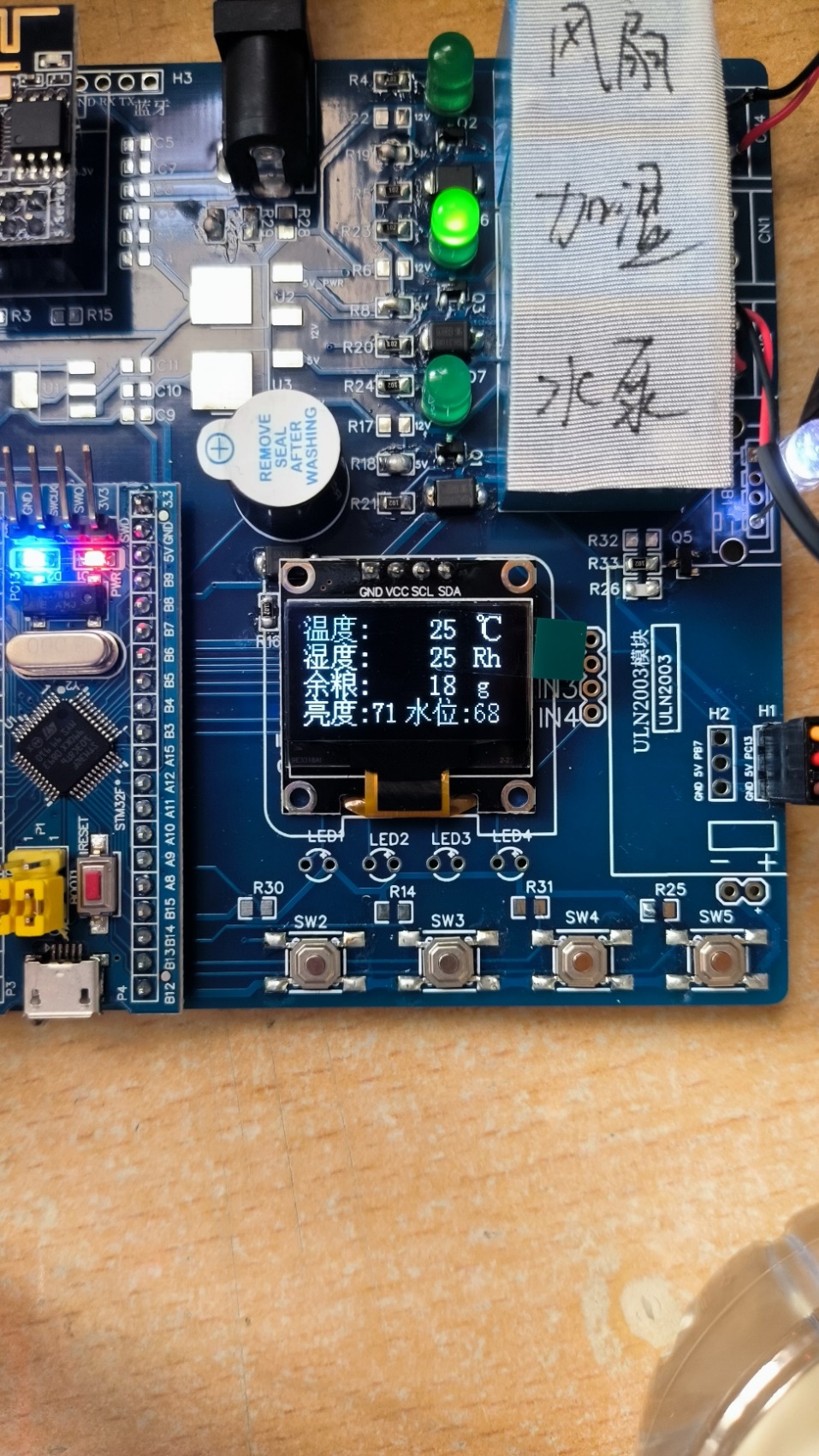


图5-4 传感器数据采集功能测试

### 5.2.3 驱动部分功能测试

驱动部分包括以下几个模块：风扇，加湿器，水泵，舵机，LED灯。当传感器数据采集功能正常运行后，进行下一步测试也就是驱动部分的功能验证，将电路板供电运行一段时间后，人为改变环境各数据，记录OLED屏幕显示的数据信息，并观察驱动部分相应动作，进行四组测试，系统阈值设定统一为：温度25℃，湿度60Rh，余粮5g，水位20。测试结果如表5-1所示。

表5-1 驱动部分测试结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 第一组 | 第二组 | 第三组 | 第四组 |
| 温度 | 19℃ | 28℃ | 30℃ | 33℃ |
| 湿度 | 36Rh | 49Rh | 68Rh | 43Rh |
| 水位 | 33 | 66 | 7 | 21 |
| 余粮 | 2g | 13g | 4g | 9g |
| 光强 | 88 | 66 | 49 | 20 |
| 风扇 | 关闭 | 开启 | 开启 | 开启 |
| 加湿器 | 开启 | 开启 | 关闭 | 开启 |
| 水泵 | 关闭 | 关闭 | 开启 | 关闭 |
| 舵机 | 转动90° | 静止 | 转动90° | 静止 |
| LED灯 | 较暗 | 暗 | 亮 | 较亮 |

根据以上四组测试结果可以得出驱动部分各模块工作正常，满足设计要求。另外可以通过四个按键SW2、SW3、SW4、SW5改变温湿度阈值以及控制LED灯的开关，在SW5按键按下后灯熄灭，再次按下灯亮，SW5按键工作正常。然后在表5-1中第二组的数据下通过按键改变温湿度阈值进行四组测试，测试结果如下表5-2所示。

表5-2 按键测试结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 第一组 | 第二组 | 第三组 | 第四组 |
| 温度阈值 | 25℃ | 20℃ | 20℃ | 30℃ |
| 湿度阈值 | 60Rh | 45Rh | 65Rh | 55Rh |
| 风扇 | 开启 | 开启 | 开启 | 关闭 |
| 加湿器 | 开启 | 关闭 | 开启 | 开启 |

## 5.3 软件部分调试

### 5.3.1 设备接入OneNET云平台测试

设备上电之后，可以正常工作，数据显示正常，现在进行WiFi连接，发现WiFi模块一直无法连接到手机热点，仔细检查代码后并无问题，在经过网上查找资料后找到了原因：ESP8266目前只能接受2.4Ghz的信号，而手机热点可以选择2.4G和5G，将手机热点频段改为2.4G后连接成功。如下图5-5所示。



图5-5 设备在线记录

### 5.3.2 数据流监测

设备成功连接到OneNET云平台后，设备需要按照指定格式发送数据才能被平台接收。一旦平台成功接收到设备发送的数据包，用户可以在网页控制台或手机APP中查看数据流。这些数据流的名称是由单片机在上传数据时定义的。其中temperature和humility为温湿度传感器dht11测量的环境温湿度值，范围为0-100，单位分别为“℃”和“Rh”。shuiwei为水位传感器Water Sensor测量的数值，范围为0-100。yuliang为压力传感器HX711测量的数值，范围为0-50，单位为“g”。light为光照传感器测量的光照强度数值，范围为0-100。并且每个数据流可以根据不同时间段对测量数据进行详细的查看。比如点开light的数据流详情，可以看到每个时间段对应的光照强度大小。具体数据流数据查看如下图5-6所示

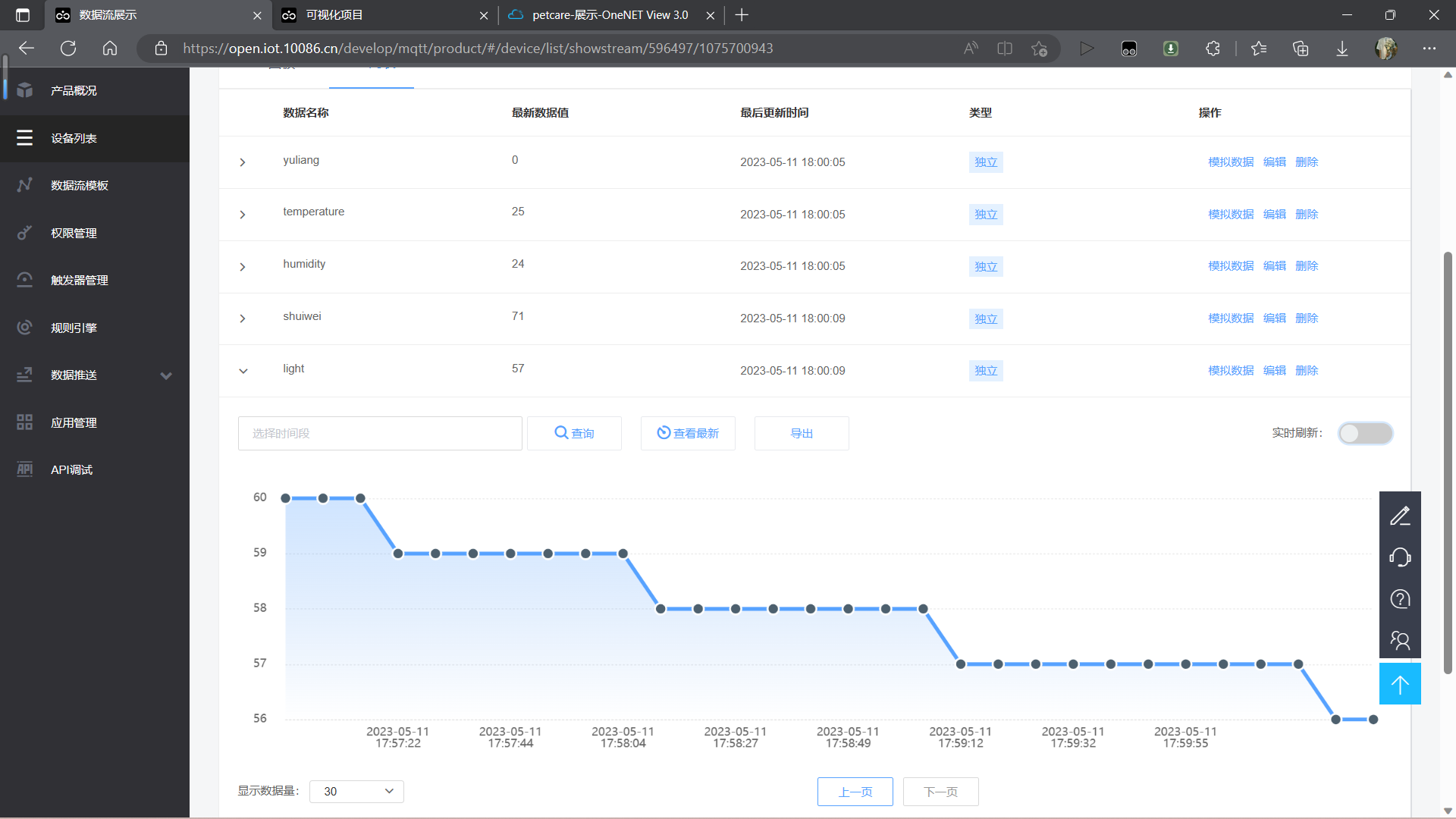


图5-6 网页端数据流显示图

手机端使用OneNET附带的APP设备云也可以进行监测，首先使用OneNET账号登录后，便可进行查看，如下图5-7所示。



图5-7 手机端数据流显示图

在获取数据后，还可以通过OneNET物联网云平台的数据可视化服务将采集到的数据用图表的方式实时展现出来，便于更清楚更简单的查看相应的数据。可以自由编辑页面，可以选择网页类型：电脑页面或者移动端页面，这里选择创建一个电脑页面。然后选择使用仪表盘分别展示温度、湿度、水位、余粮和光照的实时大小，并用折线图显示这些数据的变化情况。制作完成效果如下图5-8所示。

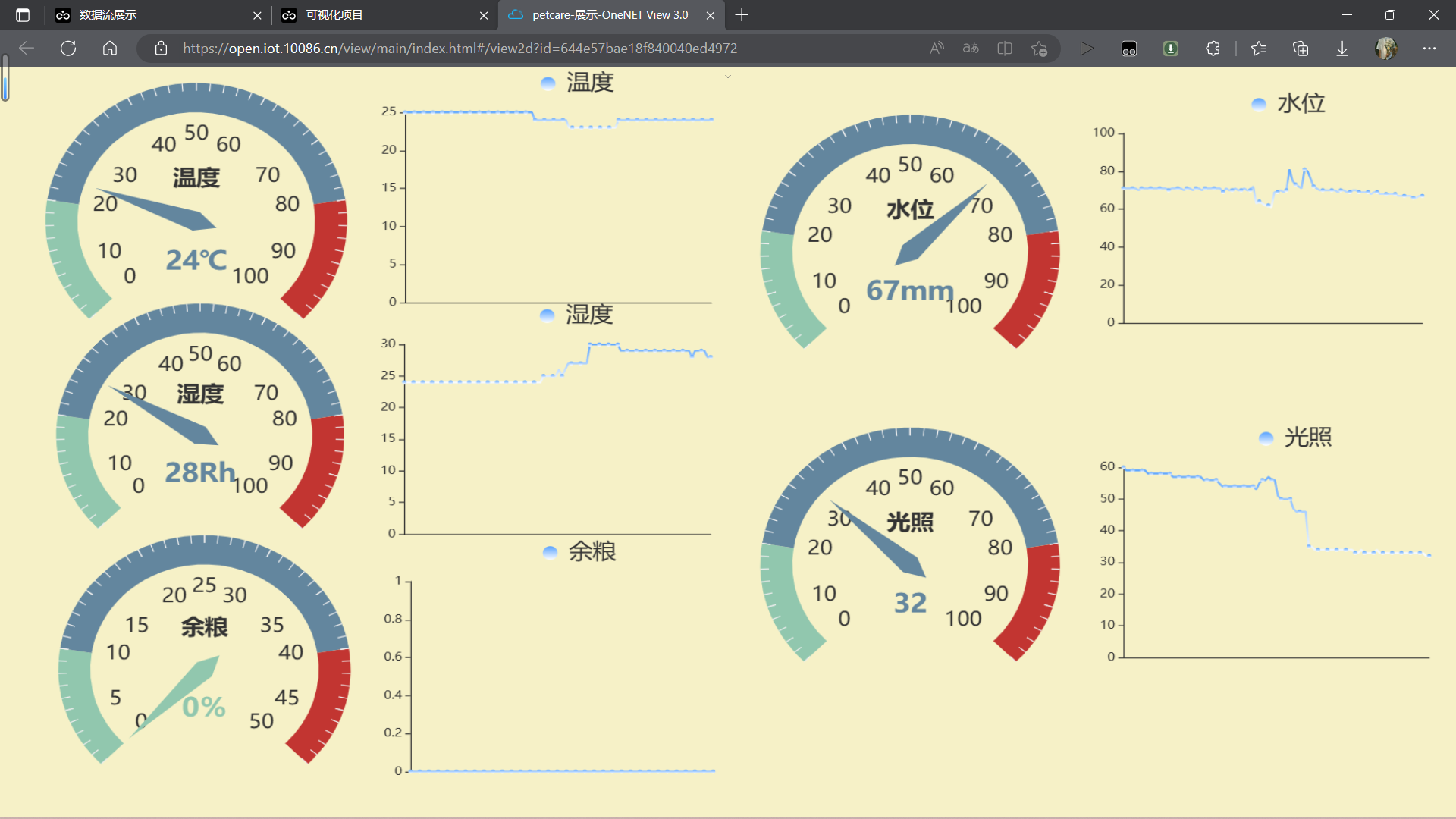


图5-8 数据可视化页面

在制作完成后还可以将制作的此项目发布，可以随时随地通过链接访问上述创建的数据可视化页面。

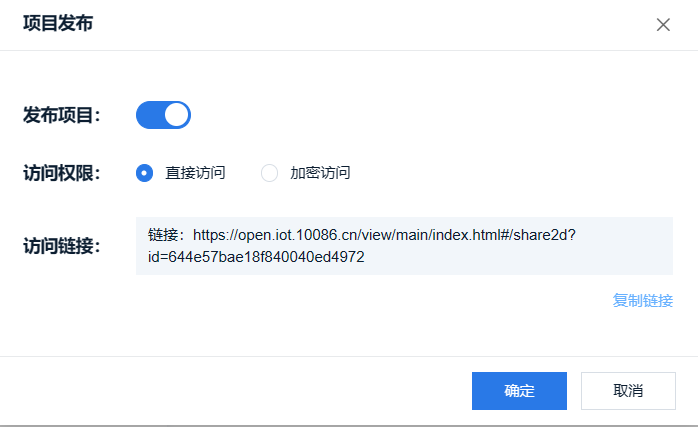


图5-9 项目发布

### 5.3.3 客户端远程控制测试

经过上面的各种调试，已经基本实现了设计要求，最后调试一下使用OneNET网页端远程下发指令控制设备实现相应动作，如下图5-10所示。下发命令对应的动作已经在上文的表4-1信号指令表中说明，这里不再赘述。

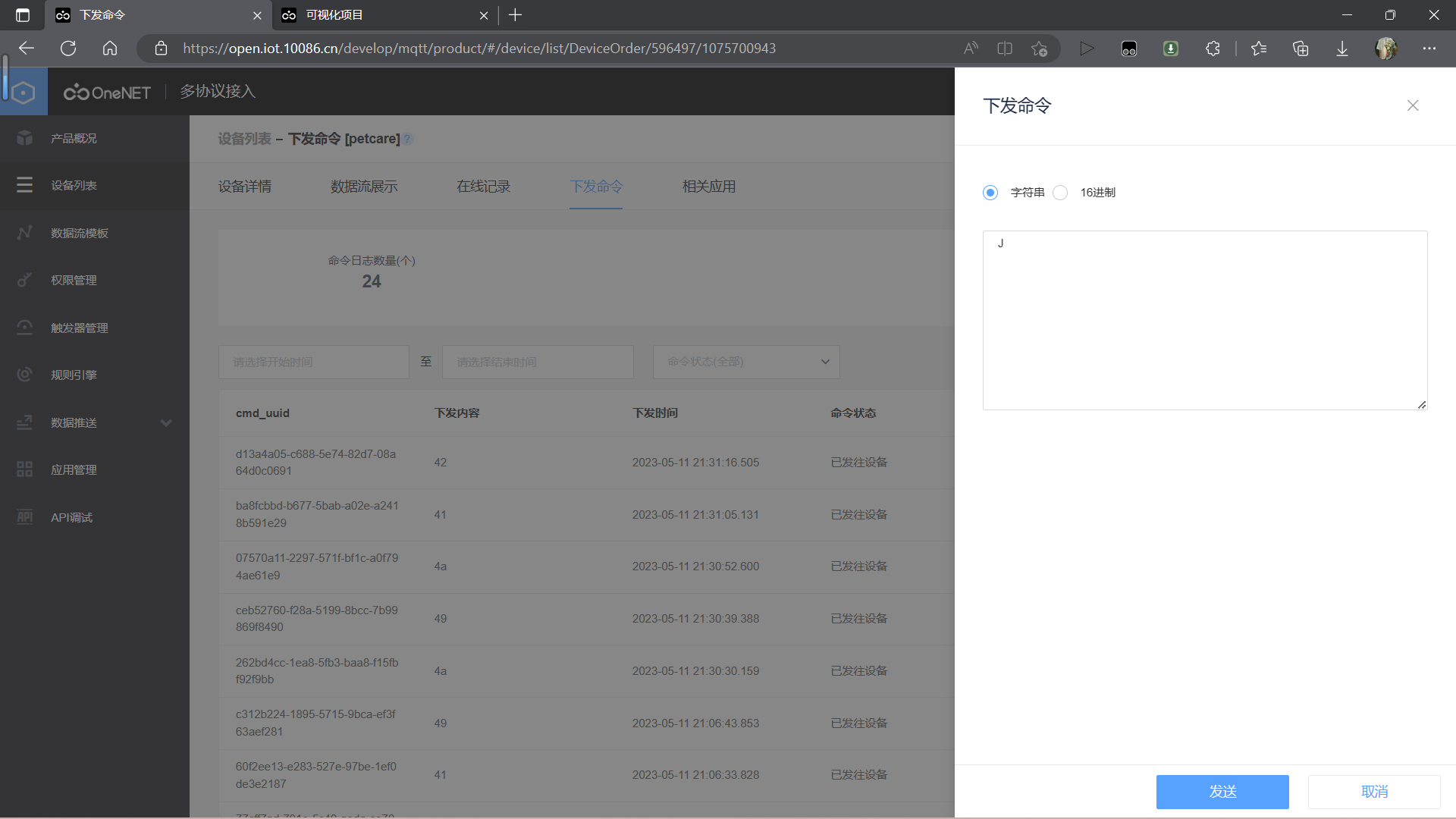


图5-10 下发命令页面

在调试过程中发现设备连接后下发命令，ESP8266指示灯闪烁表示WiFi模块已经接受到指令，但是设备并无响应，经过多次调试与检查后发现需要先下发命令“J”关闭自动模式后，再下发其他指令才可进行远程控制。下发指令“I”再次开启自动模式。将此问题弄清楚后，调试成功，进行了多次测试，设备均可准确运行。

到此整个设备调试完成，达到预期设计目标。

# 第6章 经济分析与环保分析

## 6.1 经济性分析

物联网智能宠物看护系统具有多项经济性优势。

首先，该系统可以通过智能化的监控和控制，降低照料宠物的成本。传统的宠物看护方式需要雇佣专业人员或者将宠物送到宠物店进行照料，这些方式都需要花费一定的费用。而智能宠物看护系统则可以通过智能喂食、自动换水等功能，降低了雇佣专业人员的成本，并且可以减少宠物寄养的费用。

其次，该系统可以提高宠物的健康水平，从而减少兽医费用。智能宠物看护系统可以通过实时监测宠物的饮食、生活环境各项数据，及时发现宠物的问题并采取相应的措施，从而减少宠物生病的概率。这样不仅可以降低兽医费用，还可以延长宠物的寿命，进一步降低了养宠物的成本。

第三，该系统可以提供更好的用户体验，增加用户黏性和忠诚度。智能宠物看护系统可以通过手机App等方式提供实时监控，让宠物主人能够随时随地监控宠物的情况，并及时做出相应的调整。这种个性化的服务可以提高用户的满意度，增加用户对产品的依赖和信任。

综上所述，物联网智能宠物看护系统具有多项经济性优势，可以降低成本，提高用户体验，是一项非常具有投资价值的技术。

## 6.2 环保性分析

能耗方面：使用的STM32芯片本身具有低功耗特点而且同时还保证了系统整体的正常运行，因此，整个系统有着很高的能耗比。

使用寿命：物联网智能宠物看护系统的寿命也是影响其环保性的因素之一。如果系统寿命较短，将会增加更新和更换的频率，导致更多的废弃物和资源浪费。因此，在设计系统时考虑到其长久的使用寿命，使用PCB打板代替普通洞洞板焊接，使得电路板的使用寿命大大延长。

健康安全：物联网智能宠物看护系统在使用中符合健康安全标准，不会对人和宠物造成伤害和危害。使用的是环保材料元件，以确保系统安全环保。

结 论

本文基于一款STM32单片机，使用多种传感器检测各项数据，在OLED屏幕上显示的同时将这些数据通过ESP8266模块上传到OneNET物联网云平台，可以在网页端实时监测也可以在手机APP上远程监测。系统设计有两种工作模式，自动模式下单片机会将数据与阈值比较，然后给驱动模块相应信号，使其做出相应动作。手动模式需要通过物联网平台先下发命令关闭自动模式，之后便可通过下发不同的命令信号控制驱动模块。

经过硬件电路设计和软件设计，成功完成了实物的制作以及调试工作。基本上能够实现远程看护宠物的要求，并且经济成本较低，微型化，低能耗能够广泛应用，传感器测量结果误差小，可靠性高，物联网技术迎合当前热门行业，具有广阔的发展空间。

由于时间原因和个人精力原因未能够做到完美，我认为还可以从以下几个方面进行改进和完善：

1. 不能监测宠物的运动轨迹，在后期研究可以加入定位功能或者加入摄像头实时监测宠物的状态。
2. 由于时间原因，硬件电路整体是无遮盖的，后续可以进行外观设计，3D打印一套外壳，使其更加美观实用。

参考文献

1. 邵彦达，沈睿，眭耀宇.基于AT89C51单片机的智能宠物家居[J]. 科技创新导报，2017，14(14)：1-2.
2. 周锦荣，张振宇，袁忠强.家居宠物智能喂食管理控制系统设计与应用[J]. 家畜生态学报，2019，40(04)：77-80.
3. 吴明泽，胡立夫，胡家瑞，等.智能宠物护理机器人[J]. 中国科技信息，2021，(06)：79-80.
4. 王凝.关于宠物喂食器的智能化发展趋势及设计方案研究[J]. 科技与创新，2022，(11)：93-95.
5. Kim Seungcheon.Smart Pet Care System using Internet of Things[J]. International Journal of Smart Home，2016，10(3)：164-166.
6. Anggraini Nenny，Rahman Dzul Fadli，Wardhani Luh Kesuma，et al.Mobile based monitoring system for an automatic cat feeder using Raspberry Pi[J]. TELKOMNIKA (Telecommunication Computing Electronics and Control)，2020，18(2)：354-357.
7. 张帅，张迪，樊京，等.基于阿里云的电动汽车无线充电监控系统设计[J]. 电工技术，2021，(01)：102-104.
8. 李雪健，陈燕春，夏文倩，等.黑板自动除尘系统[J]. 电子技术与软件工程，2022(20)：71-74.
9. 张一波，董增寿，常春波.基于STM32F4的换热站控制系统的设计[J]. 太原科技大学学报，2021，42(06)：434-440.
10. 迪力阿热木·吐尔地玉苏甫，郭斌.基于目标检测的自动断电实验台控制器的设计[J]. 电脑知识与技术，2022，18(18)：80-82+85.
11. Biotechnology Companies; IAR Systems supports the new STM32 F0 low-cost ARM(R) CortexTM-M0 microcontrollers from STMicroelectronics(R)[J]. Technology & Business Journal，2012：67-70.
12. Ganssle， Jack G.Watchdogs redux[J]. Embedded Systems Design，2011，24(2)：261-265.
13. 魏永生.STM32单片机定时数据采集系统参数设置与编程[J]. 电子元器件与信息技术，2022，6(03)：23-25+28.
14. 詹治国，林少波，郭栋，等.智慧楼宇通信技术架构及关键技术研究[J]. 计算机测量与控制，2022，30(12)：175-179+217.
15. 孔祥斌. 某产品水下运动姿态测量系统设计[D]. 中北大学，2014：18-19.
16. 张桂莲，石宜金，谭贵生，等.基于点灯科技的智能家居控制系统设计[J]. 现代信息科技，2023，7(08)：185-188.
17. 沈晨航，周俊.基于ESP8266WiFi模块和MQTT协议的游泳馆水质监测系统设计[J]. 数字技术与应用，2020，38(05)：148-151.
18. 史艳红，张玉杰.基于ZigBee和LabVIEW的多地无线温湿度监测系统设计[J]. 机床与液压，2015，43(22)：131-134.
19. 王娇娇，田歌，王巧华，等.一体式禽蛋呼吸检测装置设计与试验[J]. 食品科技，2023，48(01)：50-55.
20. 周乃迅，董文志，邓航宇.基于STC89C52的单手式水果称重采摘器研究[J]. 南方农机，2019，50(17)：27-28.
21. Sawicka Marta J，Wróblewska Elwira K.The application of 7H-indolo[1，2-a]quinolinium merocyanine as a new water sensor in organic solvents.[J]. Chemicke zvesti，2018，72(3)：78-82.
22. 骆雪汇，纪正辉，覃沛泳，等.微生态花园控制系统设计[J]. 机电信息，2023(06)：46-49.
23. 朱磊，李辉，李广政，等.一种新型智能晾衣架系统设计[J]. 船电技术，2023，43(02)：38-41.
24. 陈长征，窦宝茹，师亮.基于STM32单片机的智能停车场车位管理系统的设计与实现[J]. 西安航空学院学报，2020，38(05)：60-64.
25. 蒋丹.识别“八大元件”的快捷技法[J]. 职业，2006(S1)：107-109.
26. 宋锦.电子密码锁设计与调试[J]. 电脑知识与技术，2011，7(33)：8218-8220.
27. 周树勋. 小型无人直升机机载飞行控制系统研究与设计[D]. 上海交通大学，2012：16-19.
28. 张伟涛，刘保菊.智能化蔬菜大棚的设计与实现[J]. 工业控制计算机，2023，36(05)：154-155+158.
29. 黎扬，何适.基于CANoe MQTT节点仿真的T-Box远程功能自动化测试[J]. 汽车电器，2022(07)：91-92+95.
30. Ali Jawad，Haseeb Zafar Mohammad. Improved End-to-end service assurance and mathematical modeling of message queuing telemetry transport protocol based massively deployed fully functional devices in smart cities[J]. Alexandria Engineering Journal，2023，72：547-549.
31. 王红玉. OneNET云平台下基于WiFi的智能家居监控系统的设计与实现[D]. 内蒙古大学，2021：42-44.
32. 李嘉明，冯建，廖明华，等.基于ESP32的电子秤系统设计[J]. 电脑知识与技术，2021，17(12)：216-219.
33. 陈陈. 基于阿里云IoT平台的智能空调控制系的设计与实现[D]. 合肥工业大学，2021：61-65.
34. Diaconu E.M..Electrical Device Control System Through Wi-Fi Technology[J]. The Scientific Bulletin of Electrical Engineering Faculty，2021，21(1)：407-408.

附录1 系统硬件电路图



附录2 系统部分程序代码

int main(void)

{

u8 a=0;

u8 val = 0;

Timer\_SRD\_Init(9998,7199);//time3 定时器初始化

NVIC\_PriorityGroupConfig(NVIC\_PriorityGroup\_2);

delay\_init();

OLED\_Init();

LED\_Init();

KEY\_Init();

Adc\_Init();

SG90\_Init();

DHT11\_Init();

TIM2\_PWM\_Init(100,719);

uart\_init(115200);

uart2\_init(115200);

WiFi\_RxCounter=0; memset(WiFi\_RX\_BUF,0,WiFi\_RXBUFF\_SIZE); WiFi\_ResetIO\_Init();

u1\_printf("U1 OK");

u2\_printf("U2 OK");

SG90\_angle(90);

delay\_ms(100);

LED\_AllOff();

BEEP\_OFF;

RELAY1\_OFF;

RELAY2\_OFF;

RELAY3\_OFF;

SG90\_angle(0);

OLED\_Clear();

Init\_HX711pin();

delay\_ms(100);

Get\_Maopi();

delay\_ms(100);

Get\_Maopi();

delay\_ms(100);

OLED\_ShowCH(0,0,"温度:");

OLED\_ShowCH(102,0,"℃");

OLED\_ShowCH(0,2,"湿度:");

OLED\_ShowCH(101,2,"%");

OLED\_ShowCH(0,4,"余粮:");

OLED\_ShowCH(102,4,"%");

OLED\_ShowCH(0,6,"亮度:");

OLED\_ShowCH(60,6,"水位:");

while(1){

if(strstr(WiFi\_RX\_BUF,"I"))

{

u1\_printf(WiFi\_RX\_BUF);

auto\_k1=1;

WiFi\_RxCounter=0;

memset(WiFi\_RX\_BUF,0,WiFi\_RXBUFF\_SIZE);

}

if(strstr(WiFi\_RX\_BUF,"J"))

{

u1\_printf(WiFi\_RX\_BUF);

auto\_k1=0;

RELAY1\_OFF;

RELAY2\_OFF;

RELAY3\_OFF;

WiFi\_RxCounter=0; memset(WiFi\_RX\_BUF,0,WiFi\_RXBUFF\_SIZE); }

if(auto\_k1==0)

{wifi\_C();

if(sg90munber==0) SG90\_angle(0);

if(sg90munber==90) SG90\_angle(90);

delay\_ms(100);

}

if(KEY1\_IN\_STA==0){

delay\_ms(100);

if(KEY1\_IN\_STA==0){

val++ ;

OLED\_Clear();

}

}

if(KEY4\_IN\_STA==0){

delay\_ms(100);

if(KEY4\_IN\_STA==0){

light\_onoff ++;

}

}

switch (val%3)

{case 0: if(a%10==0)

{get\_shuju();

a=0;

}

break;

case 1: set\_wendu();

break;

case 2: set\_shidu();

break;

default: return 0;

}

if(auto\_k1==1) auto\_kongzhi();

}

}