

基于 ARM Cortex-M4 内核单片机的农用设备及灌溉阀门控制系统

深圳市华夏中龙电子科技有限公司

2025 年 11 月 10 日



摘 要

本文针对农业自动化设备控制系统的研究,以 ARM Cortex-M4 内核单片机为核心,构建了低功耗、高可靠、智能化的嵌入式控制系统。论文系统分析了嵌入式系统控制原理、传感器数据采集与执行机构驱动技术。重点研究了 PID、模糊、自适应及模型预测控制算法在农用设备中的应用,并结合信号滤波与边缘计算提高控制精度和系统实时性。在通信层面,分析了 LoRa、4G 等无线通信技术,并提出分布式协同控制方案。最后,从能耗优化、电源管理、系统可靠性与健康健康管理角度提出完整理论框架,为智慧农业的应用提供理论支撑。

关键词: ARM Cortex-M4; 农用设备; 嵌入式系统; 传感器控制; 阀门控制; 低功耗



ABSTRACT

This paper focuses on the research of control systems for agricultural automation equipment, utilizing an ARM Cortex-M4 core microcontroller as the core to construct a low-power, high-reliability, and intelligent embedded control system. The study systematically analyzes the principles of embedded system control, sensor data acquisition, and actuator drive technologies. It emphasizes the application of PID, fuzzy, adaptive, and model predictive control algorithms in agricultural equipment, while integrating signal filtering and edge computing to enhance control precision and system real-time performance. At the communication level, wireless technologies such as LoRa and 4G are analyzed, and a distributed collaborative control scheme is proposed. Finally, a comprehensive theoretical framework is provided from the perspectives of energy consumption optimization, power management, system reliability, and health management, offering theoretical support for the application of smart agriculture.

Keywords: ARM Cortex-M4; Agricultural Equipment; Embedded Systems; Sensor Control; Valve Control; Low-Power Consumption



目 录

摘 要	I
ABSTRACT	II
第一章 绪论	1
1.1 研究背景	1
1.2 国内外研究现状	1
1.2.1 国外研究进展	1
1.2.2 国内研究进展	1
1.2.3 研究现状分析	2
1.3 研究意义	2
1.3.1 技术层面意义	2
1.3.2 经济与社会意义	2
1.3.3 学术与应用价值	2
第二章 ARM Cortex-M4 架构及控制理论基础	3
2.1 ARM 架构概述	3
2.2 Cortex-M4 内核结构	3
2.3 指令系统与性能特性	4
2.4 控制理论基础	4
第三章 嵌入式系统控制原理与模型	5
3.1 嵌入式系统的概念与特征	5
3.2 嵌入式系统的层次结构	5
3.3 实时操作系统与任务调度机制	5
3.4 嵌入式系统的可靠性建模	6
3.5 嵌入式系统的建模与仿真	6
第四章 农用设备自动控制系统设计理论	7
4.1 农用设备自动控制系统总体结构	7
4.2 系统功能模型	7
4.3 控制算法体系	8
4.4 系统通信与协同机制	8
第五章 控制算法与信号处理理论	10
5.1 农用设备控制算法的总体特征	10



5.2 经典 PID 控制算法及其优化.....	10
5.2.1PID 控制的基本原理.....	10
5.2.2 PID 参数整定方法.....	10
5.3 模糊控制原理	10
5.3.1 基本思想	10
5.3.2 模糊控制器结构	10
5.3.3 程序实现	10
5.4 自适应控制原理	10
5.4.1 基本思想	10
5.4.2 自适应控制器结构	10
5.4.3 程序实现	10
5.5 PID 控制、模糊控制、自适应控制对比分析.....	10
5.5.1 不同维度的难易程度对比	10
5.5.2 控制算法总结	10
第六章 通信与网络系统设计理论	14
6.1 农业物联网通信需求分析	14
6.2 无线通信技术比较	14
6.3 网络拓扑与通信架构	14
6.4 边缘计算与分布式协调机制	14
第七章 系统能耗优化与可靠性分析	16
7.1 农业控制系统的能耗特征分析	16
7.2 低功耗设计理论基础	16
7.3 能耗建模与分析方法	16
第八章 嵌入式电路设计及程序设计	18
8.1 单片机最小系统	18
8.2 直流电机阀驱动和角度检测	19
8.2.1 直流电机驱动	19
8.2.2 电机角度检测及计算	20
8.2.3 阀门初始角度及角度标定	21
8.2.4 角度电压计算程序	22
8.3 传感器及 I2C 接口扩展	23
8.4 4~20mA 压力传感器采样	25
8.5 4G Cat1 通讯电路.....	28



8.6 LoRa 通讯连接与系统扩展功能扩展接口	299
8.7 电源及电池充放电电路	30
8.8 传感器及电机升降压电路	30
8.9 电池电量测量	31
第九章 系统传感器、功耗和通讯测试	32
9.1 上位机程序	32
9.2 角度旋转精度	32
9.3 气压传感器测量精度	33
9.4 低功耗测试	33
9.5 通讯响应测试	34
结束语	36
谢 辞	36
参考文献	37



第一章 绪论

1.1 研究背景

21 世纪以来，全球农业正经历从传统生产方式向自动化、信息化与智能化转型的关键阶段。农业生产的规模化与集约化，使得单纯依赖人工的操作方式已无法满足现代农业的高效率、精确化和可持续性要求。在此背景下，农业装备与信息技术、嵌入式系统技术、传感器网络技术的深度融合，逐渐成为智慧农业发展的核心动力。通过嵌入式控制系统实现农用设备的自动监测与控制，能够显著提高作业效率，降低资源浪费，并为农业环境管理提供可靠的数据支持。

与此同时，国家层面提出了“智慧农业”“数字乡村”等战略方针，推动农业信息化、机械化和智能化的协同发展。农业设备正从传统机械控制向智能嵌入式控制系统过渡。这类控制系统在工作过程中需要实时采集环境信息、执行决策算法，并驱动执行机构完成作业任务。因此，系统的控制核心应具备高性能计算能力、丰富的外设接口、低功耗特性及良好的稳定性。

ARM Cortex-M4 内核单片机因其优越的性能、强大的数字信号处理能力以及低功耗特性，在农业自动化领域具有显著的应用潜力。Cortex-M4 内核支持硬件浮点单元（FPU）及 DSP 指令集，可有效应对传感器数据滤波、PID 控制及模糊控制等算法计算需求；同时，其多级中断控制与实时操作系统（RTOS）支持，使其非常适合应用于需要高实时性响应的农用控制场景。

1.2 国内外研究现状

1.2.1 国外研究进展

国外在农业自动化控制方面的研究起步较早，形成了较为成熟的技术体系。欧美国家早在 20 世纪 90 年代就已提出“精准农业(Precision Agriculture)”的概念，其核心思想是通过传感器网络与控制系统实现对农业环境的精细化管理。例如，美国的 John Deere 公司研发的自动导航拖拉机系统，已实现基于 GPS 的高精度路径控制与自动播种功能；荷兰和以色列等国的温室控制系统则利用多变量传感数据实现自动灌溉与气候调节。这些系统普遍基于嵌入式微处理器平台构建，强调高实时性与能效比。

1.2.2 国内研究进展

我国在农业自动化领域的研究起步较晚，但发展迅速。近年来，随着嵌入式处理器和物联网技术的普及，国内众多高校与科研机构开始在农用设备控制系统方面展开研究。例如，在智能灌溉系统中，通过土壤湿度传感器采集数据，并以单片机为控制核心，根据设定阈值实现自动灌溉；在温室环境控制系统中，基于微控制器的系统能够自动调节风机、加热器及遮阳网，以维持作物生长的适宜环境。总体来看，国内研究正逐渐从单点自动化向系统集成化、网络化方向演进。



1.2.3 研究现状分析

综合来看，国内外农业控制系统在理论与实践方面均取得了显著进展，但仍存在若干问题：

- ① 农业设备的嵌入式控制系统缺乏标准化设计与统一架构；
- ② 系统能耗偏高，难以满足野外长期运行需求；
- ③ 实时性与可靠性在复杂环境下仍有待提高；
- ④ 智能决策算法的嵌入式实现尚不完善。

基于低功耗、高性能的 ARM Cortex-M4 的控制系统研究，能够针对这些问题提供较优的解决方案，为实现农业自动化的高效、智能和节能提供理论支撑。

1.3 研究意义

本研究旨在从理论层面对基于 ARM Cortex-M4 内核单片机的农用设备控制系统进行系统分析与模型构建，具有以下重要意义：

1.3.1 技术层面意义

通过深入研究 Cortex-M4 的体系结构与控制机制，探索其在农业自动化场景中的性能优势，为农业嵌入式控制系统提供可行的硬件与算法基础。

1.3.2 经济与社会意义

研究成果有助于降低农用设备的开发成本与能耗，提升生产效率，推动农业从劳动密集型向技术密集型转变。

1.3.3 学术与应用价值

该研究可为嵌入式控制理论与农业信息化技术的交叉研究提供理论支撑，未来 AIoT（人工智能物联网）在农业领域的落地奠定基础。

本研究从系统理论的角度出发，主要围绕以下几个方面展开：

- （1）ARM Cortex-M4 内核的体系结构与嵌入式控制原理；
- （2）农用设备控制系统的组成、功能与控制模型；
- （3）基于实时控制与信号处理算法的农用设备智能控制方法；
- （4）通信网络与能耗管理机制的理论分析；
- （5）系统可靠性与抗干扰设计的理论建模；
- （6）未来发展方向与智能化趋势展望。



第二章 ARM Cortex-M4 架构及控制理论基础

2.1 ARM 架构概述

ARM (Advanced RISC Machine) 是一种基于精简指令集 (RISC) 的处理器架构, 其设计理念是以低功耗、高性能为核心。与传统的复杂指令集计算机 (CISC) 不同, ARM 处理器通过简化指令系统和优化流水线结构, 实现了高指令吞吐率与低功耗的平衡。

ARM 公司设计的处理器并不直接生产芯片, 而是通过授权形式使各芯片厂商, 国外品牌有 ST、NXP、TI 等, 国内品牌有兆易创新、国民技术、灵动微等, 型号选型丰富, 国内厂商的产品有很好的性价比。在其架构基础上开发不同型号的单片机, 广泛应用于消费电子、工业控制、医疗设备及农业自动化系统等领域。

Cortex 系列是 ARM 公司针对不同应用场景推出的核心架构, 分为三大类:

- Cortex-A 系列: 面向高性能操作系统平台;
- Cortex-R 系列: 面向实时控制领域;
- Cortex-M 系列: 面向低功耗嵌入式控制系统。

其中 Cortex-M 系列以其高性价比、低功耗及实时响应能力, 成为嵌入式控制的主流选择。

2.2 Cortex-M4 内核结构

Cortex-M4 属于 ARMv7-M 架构的微控制器内核, 其结构设计兼顾了信号处理与控制应用的需求。其主要组成包括:

1. 中央处理单元 (CPU Core)

采用三段流水线结构 (取指、译码、执行), 支持 Thumb-2 指令集。M4 内核具有可选的单精度浮点单元 (FPU), 能够快速执行复杂的浮点计算, 适用于数字滤波、PID 控制等算法。

2. 内存体系结构

包含独立的指令与数据总线 (Harvard 架构), 通过总线矩阵 (Bus Matrix) 实现高效的数据交互。同时支持 4GB 线性寻址空间, 满足多任务并行需求。

3. 中断系统 (NVIC)

嵌入式系统中中断响应速度直接决定了系统的实时性能。Cortex-M4 的 NVIC (Nested Vectored Interrupt Controller) 支持多达 240 个中断向量与 256 级优先级, 可在硬件层面实现快速响应与中断嵌套。

4. 外设与接口

Cortex-M4 通常集成丰富的接口资源, 包括 UART、SPI、I2C、CAN、ADC、DAC、PWM 等, 能够与多种传感器与执行机构连接。对于农业控制系统, 这种灵活性尤为重要。

5. 低功耗设计

内核支持多级节能模式 (Sleep、Deep Sleep、Standby), 并通过时钟门控技术 (Clock Gating) 实现动态功耗管理, 保证在长期运行场景下的能效最优。



2.3 指令系统与性能特性

Cortex-M4 采用 Thumb-2 指令集，该指令集兼具 16 位与 32 位指令形式，在保证高性能的同时有效减少代码体积。其在性能上的突出优势包括：

1. DSP 指令扩展
M4 内核支持乘加（MAC）指令、饱和运算与位反转操作，可在硬件层面加速信号处理与控制算法执行。
2. 浮点运算支持
FPU 可进行单精度浮点加减乘除运算，支持 IEEE-754 标准，适合实时控制与动态补偿算法。
3. 可预测中断延迟
Cortex-M4 的中断响应时间在 12 个时钟周期内，显著优于多数 8 位或 16 位微控制器，保证农业设备控制系统的实时响应性。
4. 高可靠性特性
内核支持 MPU（Memory Protection Unit）用于内存保护，防止非法访问；同时具备硬件除错（Debug）与断点功能，方便系统调试与维护。

2.4 控制理论基础

在嵌入式控制系统中，Cortex-M4 通常承担传感器数据采集、信号处理、控制算法计算与执行指令输出等功能。控制系统的设计理论可从以下几方面展开：

1. 控制系统的基本模型
控制系统通常可抽象为被控对象、控制器与执行机构三部分。被控对象代表农业设备或环境参数（如温度、湿度）；控制器根据反馈信号计算控制量；执行机构根据控制量作用于被控对象，从而实现闭环控制。
2. 闭环反馈控制理论
农业控制系统普遍采用闭环结构，通过实时采集环境参数，计算偏差并进行调整。例如，在智能温室中，当温度高于设定值时，控制系统启动通风设备以降低温度，实现自动调节。
3. 实时性理论
实时控制要求系统在有限时间内响应外部事件。M4 内核结合 RTOS，可实现任务优先级管理与周期性任务调度，从而保证系统的确定性响应。
4. 可靠性与稳定性分析
稳定性是控制系统的核心指标。嵌入式控制中常通过李雅普诺夫（Lyapunov）方法或根轨迹分析评估系统稳定性；而可靠性则通过冗余设计与异常检测机制实现。



第三章 嵌入式系统控制原理与模型

3.1 嵌入式系统的概念与特征

嵌入式系统 (Embedded System) 是以应用为中心、以计算机技术为基础、软硬件可裁剪的专用计算机系统。它通常嵌入到特定的设备中, 完成对目标对象的检测、控制或信息处理任务。与通用计算机系统相比, 嵌入式系统具有以下显著特征:

专用性强, 嵌入式系统针对特定应用进行定制设计, 软硬件结构紧密结合, 资源利用效率高。

实时性要求高, 系统必须在规定时间内对外部事件作出响应。对于农用设备控制系统而言, 实时性直接关系到作业安全与控制精度。

可靠性与稳定性要求高, 农业环境复杂多变, 设备常年运行于高温、高湿或粉尘环境下, 因此嵌入式系统需具备强抗干扰能力与长期稳定性。

低功耗与小型化, 农田分布广、供电条件有限, 因此控制系统必须在低功耗设计下实现长时间独立运行。

软硬件协同设计, 嵌入式系统的功能实现往往依赖硬件与软件的深度配合, 通过优化驱动层、算法层与任务调度机制提升系统整体性能。

在农用设备控制领域, 嵌入式系统是连接物理农业环境与数字控制逻辑的关键桥梁, 其性能与可靠性决定了整个智能农业系统的效率与精度。

3.2 嵌入式系统的层次结构

一个完整的嵌入式控制系统通常由以下几个层次组成:

硬件层, 包括微控制器 (如 ARM Cortex-M4)、存储器、传感器接口、执行机构驱动电路及通信模块。该层决定系统的物理性能和信号处理能力。

驱动层, 驱动层是硬件与上层应用之间的桥梁, 负责传感器、ADC、PWM、通信接口等资源的管理。设计良好的驱动层可显著提升系统稳定性与代码可重用性。

操作系统层, 在较复杂的农业控制系统中, 往往需要引入实时操作系统 (RTOS) 如 FreeRTOS、RTX 或 $\mu C/OS-II$, 用以实现任务调度、资源管理、通信同步和中断响应等功能。

应用层, 包含具体的控制算法、数据采集逻辑、通信协议实现及人机交互界面。该层通常依据具体的农业设备类型 (如温室控制、灌溉控制、无人机喷洒系统等) 进行定制。

这种分层结构保证了系统的模块化和可扩展性, 使其能够根据农业应用的复杂程度进行功能拓展。

3.3 实时操作系统与任务调度机制

(1) 实时操作系统的必要性

随着农用设备控制系统复杂度的提高, 简单的循环轮询或中断驱动方式已无法满足多任务并发执行的需求。实时操作系统 (RTOS) 能够有效管理多任务的并行运行, 确保高优先级任务及时响应。RTOS 的核心功能包括:



- 任务管理：对多个任务的创建、删除、挂起和恢复进行统一管理；
- 任务调度：依据优先级或时间片分配 CPU 资源；
- 时间管理：提供系统时钟与延时机制；
- 中断管理：响应外部事件；
- 通信与同步机制：包括信号量（Semaphore）、消息队列（Queue）、互斥量（Mutex）等。

(2) 任务调度策略

嵌入式实时系统中的调度策略决定了系统的响应能力和任务执行的公平性。常见的调度算法包括：

- 固定优先级调度（Rate Monotonic, RM）：任务周期越短，优先级越高，适合周期性任务。
- 最早截止时间优先（Earliest Deadline First, EDF）：任务根据截止时间动态调整优先级。
- 时间片轮转（Round Robin）：保证多任务公平性，但实时性较弱。

对于农业控制系统，通常采用 RM 与 EDF 混合调度，以确保实时控制任务（如传感采样、执行机构控制）优先执行，而数据通信与显示任务可在空闲时段完成。

3.4 嵌入式系统的可靠性建模

嵌入式系统在农业场景下运行时间长、维护不便，因此必须具备高可靠性。可靠性模型通常采用概率统计方法描述系统失效率，常用指标为 MTBF（Mean Time Between Failures）。

可靠性建模可从以下方面展开：

1. 硬件冗余
对关键传感器或控制模块采用双机热备或多通道采样，提高抗故障能力。
2. 软件容错
通过异常捕获、任务自恢复和看门狗（Watchdog）机制保证系统在软件异常时自动复位。
3. 环境适应性
加强电磁兼容（EMC）设计与防雷击保护，以应对农业电网波动与雷电干扰。
4. 可靠性函数

系统的可靠性函数可表示为 $R(t) = e^{-\lambda t}$ ，其中 λ 为失效率。通过降低 λ 可显著提升整体可靠度。

3.5 嵌入式系统的建模与仿真

为验证控制系统的性能与稳定性，常采用数学建模与仿真方法。Cortex-M4 内核系统可通过 MATLAB/Simulink 建立模型，包括传感输入模块、PID 控制模块与执行输出模块。通过仿真分析响应时间、稳态误差及能耗特性，可为系统设计提供理论依据。



第四章 农用设备自动控制系统设计理论

4.1 农用设备自动控制系统总体结构

基于 ARM Cortex-M4 的农用设备控制系统通常由以下五个部分构成：

1. 数据采集模块
负责采集土壤湿度、空气温度、光照强度、二氧化碳浓度等环境参数，常用传感器包括 SHT31、BH1750、YL-69、MQ 系列气体传感器等。
2. 控制核心模块
以 Cortex-M4 单片机为中心，执行数据处理与控制算法，实现输入信号与输出执行的闭环控制。
3. 执行机构模块
包括电磁阀、电压阀、脉冲阀、水泵、风机、喷雾器等机械装置，用于直接作用于农业环境。
4. 通信与人机交互模块
通过 LoRa、4G CAT1、Wi-Fi、蓝牙或 NB-IoT 模块实现数据上传与远程控制。
5. 电源管理模块
提供稳压电源、太阳能板能量采集与能量存储系统，电池电量测量、电量预警，确保系统长期稳定运行。

该系统结构的设计思想是模块化、可扩展和高可靠性。通过标准化接口设计，系统可根据不同农作场景灵活配置。

4.2 系统功能模型

整个控制系统可用控制理论框架建模，如下：

$$y(t) = G_p(s)u(t) + d(t)$$

其中 $G_p(s)$ 为被控对象（农业设备或环境参数）的传递函数， $u(t)$ 为控制量， $y(t)$ 为输出（测量值）， $d(t)$ 为外部扰动。控制器目标是使输出 $y(t)$ 跟踪期望值 $r(t)$ ，

并最小化误差 $e(t) = r(t) - y(t)$ 。

系统的核心任务包括：

- 精确测量环境变量；
- 实现稳定控制；



- 具备抗扰动能力;
- 保证能效最优。

在 ARM Cortex-M4 平台上, 可通过 ADC 模块进行高精度采样, 通过 PWM 驱动执行器, 并利用定时中断控制循环周期。

4.3 控制算法体系

农业控制系统的算法设计是整个理论研究的关键部分。常见的算法体系包括:

1. PID 控制算法

是最常用的反馈控制算法之一, 通过比例 (P)、积分 (I)、微分 (D) 环节调节控制输出。

其控制规律为:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt}$$

ARM Cortex-M4 的 FPU 能够快速执行浮点运算, 使其非常适合在嵌入式环境中实现连续控制。

2. 模糊控制算法

对于非线性、时变或不确定性较强的农业系统, 如温室气候控制, 可采用模糊逻辑算法。该算法利用“温度较高”“湿度适中”等语言规则构建控制规则库, 实现智能化调节。

3. 自适应控制算法

自适应控制可根据系统参数变化自动调整控制器参数, 如增益调节、积分系数调整, 以应对环境扰动和设备老化问题。

4. 多变量耦合控制

在复杂环境中, 多个变量 (温度、湿度、光照) 之间存在耦合关系。通过多输入多输出 (MIMO) 控制模型, 可实现整体协调控制, 防止单变量控制导致系统不稳定。

4.4 系统通信与协同机制

农用设备的分布式特征决定了系统必须具备多节点协同与远程通信能力。

Cortex-M4 控制器通过以下方式实现通信与协调:

1. 有线通信: 通过 RS485 或 CAN 总线实现节点间的数据交换, 适用于近距离、稳定性要求高的场景。
2. 无线通信: 采用 LoRa 或 NB-IoT 进行远距离低功耗通信, 实现跨区域数据传输。
3. 边缘协同机制: 通过本地控制节点间的分布式协调, 减少云端依赖, 提高系统响应速度。

通信协议层可采用 Modbus RTU、MQTT 或 CoAP, 以支持多协议互联。

4.5 环境适应性设计理论

农业控制系统面临高湿、高粉尘、强电磁干扰等极端环境, 因此必须在系统设计



中融入环境适应性理论：

1. 电源防护：采用隔离稳压电路和过压保护设计，防止农用电网波动导致系统损坏。
2. 抗干扰措施：包括信号滤波、接地分区与屏蔽布线。
3. 防水防尘结构：控制器外壳应达到 IP65 以上防护等级。
4. 温度补偿机制：在传感器与控制算法中加入温度补偿模型，确保测量精度稳定。



第五章 控制算法与信号处理理论

5.1 农用设备控制算法的总体特征

在基于 ARM Cortex-M4 内核的农用设备控制系统中，控制算法是实现环境参数调节与执行机构控制的核心部分。农业生产过程具有明显的非线性、强耦合和时变性特征，不同于工业控制的高稳定性环境。为了实现高精度、低延迟和高鲁棒性的控制效果，控制算法必须具备以下特征：

- 1，实时性强：控制算法需在毫秒级完成数据处理与执行指令的输出，以满足农业设备实时响应需求；
- 2，适应性高：系统必须根据环境变化动态调整控制参数，保证在外部扰动下的稳态性能；
- 3，鲁棒性强：控制系统必须能够在传感器误差、通信延迟等不确定因素下维持稳定；
- 4，低计算负荷：由于嵌入式系统资源有限，算法设计需兼顾运算复杂度与控制精度的平衡。

ARM Cortex-M4 的 DSP 指令集和浮点运算单元（FPU）使其在控制算法实现上具有极大优势，能够在嵌入式层面直接实现复杂的滤波、积分和非线性控制算法，从而为智能农业控制系统提供高性能计算基础。

5.2 经典 PID 控制算法及其优化

5.2.1 PID 控制的基本原理

PID 控制（Proportional-Integral-Derivative）是最常用的反馈控制算法，其控制规律可表示为：

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

其中： $e(t) = r(t) - y(t)$ ：系统偏差； K_p, K_i, K_d ：比例、积分、微分系数； $u(t)$ ：控制输出信号。

在农用设备控制中，PID 算法常用于温度、湿度、水压、流量等连续变量的稳定控制。Cortex-M4 内核通过高效的定点或浮点运算可在毫秒级实现控制环节更新。

在本系统中直流电机阀的角度控制，以及电机阀门的开合角度控制与水压的对应关系将是 PID 算法的重要应用。

5.2.2 PID 参数整定方法

PID 控制器性能依赖于参数 K_p, K_i, K_d 的合理配置。常见的参数整定方法



包括：经验整定法（Z-N 法）：通过阶跃响应曲线整定系统参数；临界比例法：逐步增加比例系数，直到系统产生持续振荡；自动整定（Auto-Tuning）：基于模型识别算法，自动寻优 PID 参数。

5.3 模糊控制原理

5.3.1 基本思想

模糊控制不依赖被控对象的精确数学模型，而是将人的经验和语言规则转化为控制策略，其特点是非线性，规则驱动。对系统不确定性和扰动鲁棒性强，不需要系统精确模型。典型形式：

“如果误差 大，且误差变化率 快，那么控制量 增大”

5.3.2 模糊控制器结构

模糊控制系统一般由四个部分组成：

A, 模糊化（Fuzzification）

将精确输入（如误差 e 、误差变化率 ec ）转换为模糊语言变量

.NB（负大）
.NM（负中）
.NS（负小）
.ZO（零）
.PS / PM / PB

B, 模糊规则库（Rule Base），

规则形式：IF e is PB AND ec is PS THEN u is PM

C, 模糊推理（Inference），

常用：Mamdani 推理、Max - Min 推理

D, 解模糊（Defuzzification）

将模糊输出转换为精确控制量，常用方法：重心法（Centroid）

5.3.3 程序实现

以 STM32 单片机为例，程序结构：

采样传感器 →

计算误差 e 、 ec →

模糊化 →

模糊推理 →

解模糊 →

输出控制量（PWM / DAC）

程序代码：

```
float Fuzzy_Control(float e, float ec)
{
    Fuzzify(e, ec);           // 模糊化
    RuleInference();          // 规则推理
}
```



```
u = Defuzzify();           // 解模糊
return u;
}
```

5.4 自适应控制原理

5.4.1 基本思想

自适应控制有较强的依赖数据值得稳定性，需要防止参数伐善，并且对 MCU 有比较高得运算能力的要求。通常应用于负载变化大的系统中，且参数随着时间的变化而变化。自适应控制通过在线识别或调整参数，使控制器能够适应系统参数变化。核心思想是控制器参数不是固定的，而是随系统变化自动调整。

5.4.2 自适应控制的基本结构

典型结构包括：

- A, 被控对象
- B, 参数估计器
- C, 自适应律
- D, 可调控制器

两种常见形式：

- A, 模型参考自适应控制（MRAC）
- B, 自校正控制（STR）

5.4.3 基本结构

- A, 设定一个理想参考模型
- B, 实际输出与模型相比较
- C, 根据误差调整控制器参数

参数调整率公式：

$$\dot{\theta} = -\gamma ex$$

- θ : 控制器参数
- e: 跟踪误差
- γ : 自适应增益)

5.4.4 程序实现

以 STM32 单片机为例，程序结构：

采样 →
计算误差 →
更新参数（自适应律） →
计算控制量 →
输出
程序代码：



```
void Adaptive_Control(void)
{
    e = r - y;
    theta += gamma * e * x;    // 参数更新
    u = theta * x;             // 控制律
}
```

5.5 PID 控制、模糊控制、自适应控制对比分析

5.5.1 不同维度的难易程度对比

此三种控制方式按照其特点可用于不同的应用场景中，其不同维度下的运用难易程度总结如下表格：

对比维度	PID	模糊控制	自适应控制
实现难度	★	★★	★★★
对模型依赖	★★	★	★★★
控制精度	★★	★★	★★★
鲁棒性	★★	★★★	★★★
工程成熟度	★★★	★★	★★
MCU 资源需求	低	中	高

5.5.2 控制算法总结

PID 控制结构简单、工程成熟，适用于模型相对明确、参数变化较小的系统；模糊控制通过引入经验规则，增强了系统对非线性和不确定性的适应能力；自适应控制通过在线调整控制器参数，在系统参数变化较大的情况下仍能保持良好的动态性能和稳态精度。三种控制方法各具优势，在实际工程中常通过融合控制策略以获得更优的控制效果。



第六章 通信与网络系统设计理论

6.1 农业物联网通信需求分析

现代农业控制系统已不再是单一的现场闭环系统，而是逐步向智能化、网络化和分布式结构发展。通信系统是实现数据采集、远程控制与云平台集成的关键环节。农业物联网通信系统需满足以下要求：

1. 广域覆盖：农场通常地理分布广，通信系统需支持数公里级传输距离；
2. 低功耗运行：现场节点多依赖电池或太阳能供电，通信模块必须具备低功耗特性；
3. 高可靠性：在强干扰、恶劣天气下仍需保持通信稳定；
4. 多节点协同：系统需支持多设备同时接入与数据同步；
5. 可扩展性：通信结构应支持节点动态扩展与远程升级。

6.2 无线通信技术比较

在农用控制系统中，常见的无线通信技术包括 LoRa、NB-IoT、ZigBee、Wi-Fi 与蓝牙。其特性如下表所示：

技术	传输距离	功耗	速率	优点	典型应用
Lora	1-15km	极低	0.2~100kbps	远距离、低功耗	中、远程阀门控制
Zigbee	10-100m	低	250kbps	组网灵活、低延迟	传感器高密度组网
Wi-Fi	50-100m	低	10-100Mbps	高速传输	传感器组网
4G Cat1	无限制	较高	10-100Mbps	高速传输	阀门控制
蓝牙 BLE	10-30m	低	1Mbps	低功耗	与手持终端交互

综合比较，在偏远的郊区，LoRa 与 4G Cat1 最适合分布广、功耗要求高的农用设备系统，本文的设计将采用这两种通讯方式。

6.3 网络拓扑与通信架构

根据农场布局与设备类型，通信系统可采用以下拓扑结构：

1. 星型网络：中心节点管理多个终端节点，结构简单，适合集中式控制；
2. 树型网络：可扩展性强，适合多区域层级管理；
3. 网状网络（Mesh）：节点间相互转发数据，具有高可靠性与自愈能力。

ARM Cortex-M4 控制器可通过 SPI 或 UART 接口与通信模块连接，实现节点间数据传输。通信协议层常采用 MQTT 或 Modbus，以实现跨系统兼容性。

6.4 边缘计算与分布式协调机制

在智能农业中，边缘计算成为实现高效数据处理的重要理论方向。边缘节点（由 Cortex-M4 控制器构成）在本地完成数据分析与控制决策，仅将关键信息上传云端。该机制具有以下优势：

1. 降低通信负载：减少原始数据传输，提高带宽利用率；



2. 提升实时性：本地快速响应，不依赖远程云端延迟；
3. 增强系统可靠性：当网络中断时，边缘节点仍可独立运行；
4. 实现协同优化：通过多节点交互实现区域性调度与资源共享。

分布式协调机制基于多控制器间的信息交换与同步控制，可通过时间同步协议（如 NTP）与分布式调度算法实现全网一致性控制。



第七章 系统能耗优化与可靠性分析

7.1 农业控制系统的能耗特征分析

农用设备控制系统通常分布在广阔的田间环境中，供电条件受限，多数设备依赖太阳能、蓄电池或低压直流电源。因此，系统能耗优化是提高设备续航能力与稳定运行时间的关键环节。基于 ARM Cortex-M4 内核的控制系统，在低功耗运行模式下可显著降低能耗，但其能耗来源仍主要包括以下几个方面：

1. 处理器核心功耗：主频与工作电压是影响能耗的主要因素。
2. 传感器与执行机构功耗：传感器采样频率、执行器工作时间直接影响系统能量消耗。
3. 通信模块功耗：无线通信（如 LoRa、NB-IoT）在数据传输期间能耗较高。
4. 电源转换与稳压损耗：DC-DC 转换与稳压模块会带来附加损失。

在复杂农业场景下，能耗不仅影响系统运行时间，还会制约通信频率与控制精度，因此必须在理论层面建立能耗优化模型。

7.2 低功耗设计理论基础

低功耗设计的核心在于硬件优化与软件调度协同。基于 ARM Cortex-M4 架构，可从以下方面实现能耗控制：

（1）硬件层低功耗机制

1. 动态电压与频率调整（DVFS）

Cortex-M4 内核支持动态调整主频与供电电压。通过在任务低负载时降低频率可有效减少功耗，理论上功耗与频率呈线性关系，与电压呈平方关系：

$$P \propto CV^2f$$

其中 C 为负载电容，V 为供电电压，f 为时钟频率。

2. 多电源域设计

将系统划分为核心域、外设域、通信域等独立电源区域，可在空闲时关闭部分电路。

3. 睡眠与待机模式

Cortex-M4 提供多级休眠模式（Sleep、Deep Sleep、Stop、Standby），可根据任务实时性自动切换状态，保证系统按需唤醒。

4. 传感器功耗控制

对间歇采样的传感器（如光照、温湿度）采用定时唤醒与休眠机制，减少持续采样能耗。

7.3 能耗建模与分析方法

能耗建模旨在定量描述系统的功耗构成与动态变化规律。常见建模方法包括：

1. 统计模型法

通过实验测量不同工作状态下的电流与时间占比，建立平均功耗模型：



$$P_{avg} = \sum_i I_i V_i \frac{t_i}{T}$$

其中 I_i 为状态电流， V_i 为电压， t_i/T 为时间占比。

2. 状态机建模法

构建系统功耗状态转移图，如“采样→计算→通信→休眠”，并根据转移概率计算期望能耗。

3. 能耗仿真分析

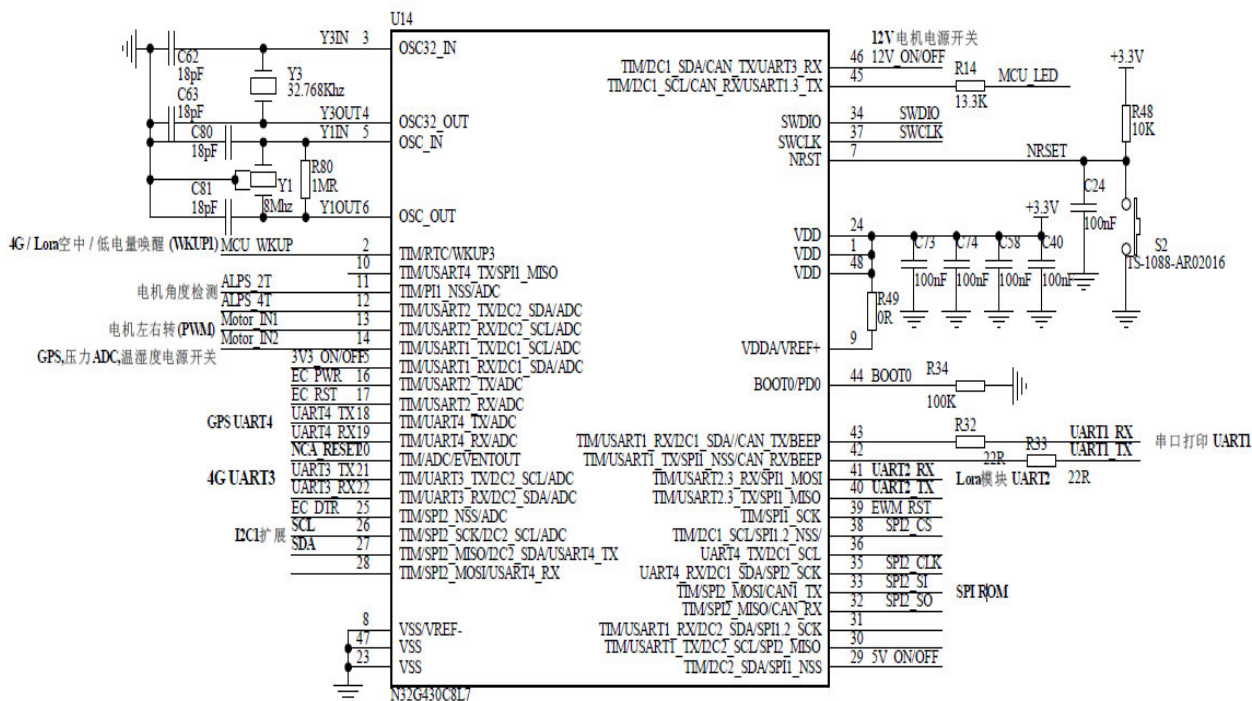
借助软件工具（如 PowerTutor、EnergyTrace）模拟运行过程，优化调度与任务周期。



第八章 嵌入式电路设计及程序设计

8.1 单片机最小系统

在硬件电路设计中，处理器选择国民技术的 N32G430C8L7，这是一个基于 ARM Cortex-M4F 内核的通用微控制器，其最大的亮点是极高的性价比。它在保持较低成本的同时，提供了远超同级 Cortex-M0/M0+ 芯片的性能和丰富的外设资源，特别适合需要一定算力且成本敏感的应用。芯片集成硬件有 FPU，支持单精度浮点运算。对于涉及电机控制、数字电源、复杂算法（如 PID 控制、滤波）的应用，FPU 能极大地加速运算，减少 CPU 占用，使软件设计更高效。主频高达 128MHz，提供了充足的指令处理能力，能够应对复杂和实时的控制任务。64KB Flash + 16KB SRAM，对于大多数中小型应用来说，这个容量是“黄金配置”，既不会浪费，又能满足程序代码、通信协议栈和运行数据的需求。支持存储加密，内置密码算法硬件加速引擎，支持 AES, DES, SHA 等，增强了产品的安全性，防止代码被轻易窃取或篡改。高级定时器，包含多个 16 位高级 PWM 定时器，支持带死区控制的互补输出。这对于驱动三相电机（BLDC/PMSM）和开关电源是必不可少的。常见的 UART, I2C, SPI 等通讯接口都俱全，在低功耗方面表现优异，低功耗运行电流仅 100uA/MHz，其最小系统电路如下图：



在这个控制系统中，需要控制农业相关的各类阀门，包括直流电机 L 型阀，电压阀，4~40mA 电流阀，传感器有温湿度传感器、水管压力传感器两个，要连接的模块有 GPS 模块、LoRa 模块，4G Cat1 模块，各 IO 口的功能分配如下表：

信号定义	信号功能
MCU_WKUP	单片机唤醒

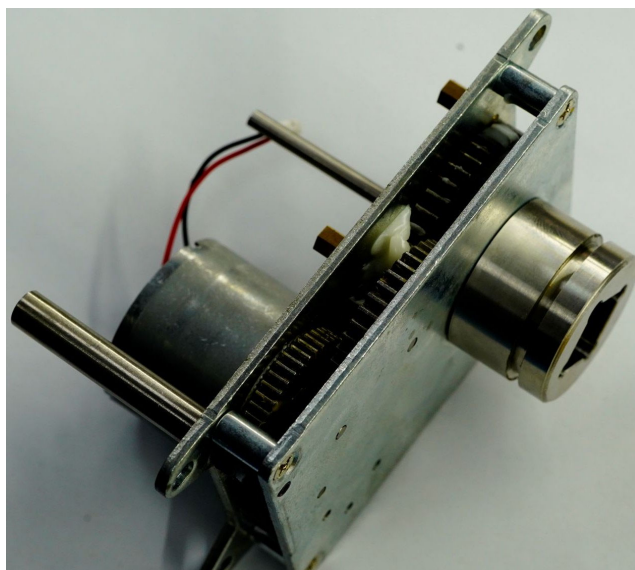


ALPS_2T,ALPS_2T,	电机旋转角度传感器
Motor_IN1,Motor_IN12	直流电机
3V3_ON/OFF	GPS I2C 传感器电源开关
EC_PWR	4G Cat1 电源开关
EC_RST	4G Cat1 复位信号
UART4_TX,UART4_RX,	GPS 模块串口
NCA_RESET	I2C 扩展芯片复位信号
UART3_TX,UART3_RX,	4G Cat1 模块串口
EC_DTR	4G Cat1 模块唤醒信号
SCL,SDA	I2C 扩展芯信号
5V_ON/OFF	5V DC/DC 电路开关
SPI2_CLK, SPI2_SI, SPI2_SO, SPI2_CS	SPI Flash ROM
EWM_RST	Lora 模块复位信号
UART2_TX,UART2_RX	Lora 模块串口
UART1_RX, UART1_TX	MCU 有调试、上位机通讯接口
SWDIO, SWCLK	程序下载接口
MCU_LED	单片机工作状态指示
12V_ON/OFF	12V DC/DC 开关信号

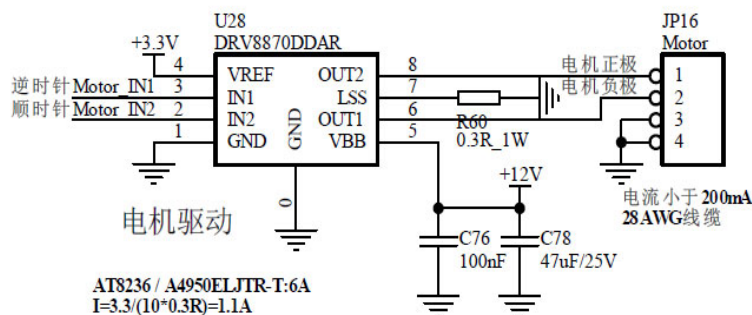
8.2 直流电机阀驱动和角度检测

8.2.1 直流电机驱动

直流电机阀、L 型双通阀，是一种用两线直流电机带动中间转轴旋转，使阀门向一个方向或两个方向打开的阀门，这种阀门随着口径的增大，旋转角度的力矩也增大，所以这种阀门要用到带有减速齿轮组的电机，使转动力矩大于 50N/米，电机的工作电压 12V，电机的堵转电流 2A，下图就是带有电机和简述车轮旋转机构总成：



这种两线直流电机可用 H 桥驱动芯片 DRV8870 可以满足设计需求, 堵转保护电流设定在 1.1A, 及 LSS 的下拉电阻用 0.3 欧姆, 电阻的功率应不小于 1W。驱动更大力矩的阀门时, 型号可改用 AT8236, 它的驱动电流可达 6A, 原理图如下图 1。



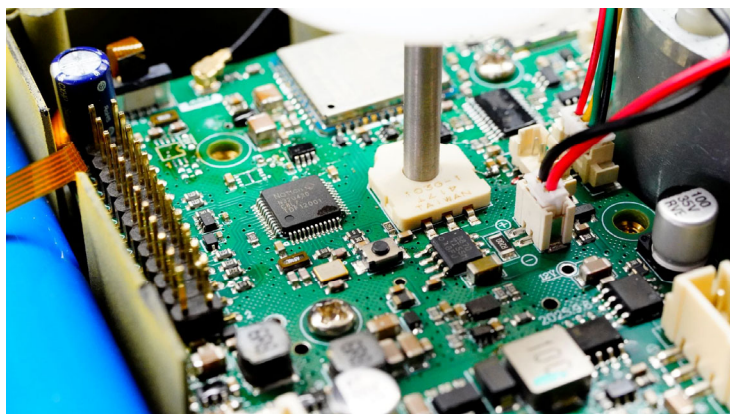
(图 1)

控制方法，U28 是一个仅需单片机 IO 输出高低电平或 PWM 信号即可控制的 H 桥驱动芯片，PWM 控制可以节省电能，但是直流电机的速度会下降，在实际的试验中，用 PWM 方式使阀门旋转一圈需要 20 秒中以上，所以在本设计中，用高低电平的方式控制，控制逻辑图下表：

	Motor_IN1	Motor_IN1	旋转方向及输出状态
逻辑电平	H	L	逆时针
	L	H	顺时针
	L	L	H 桥高阻芯片休眠
	H	H	停机

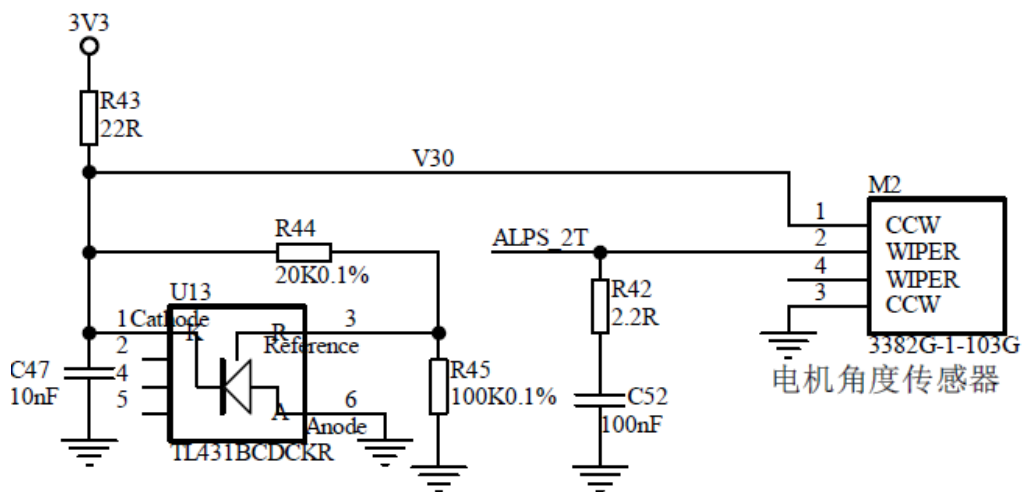
8.2.2 电机角度检测及计算

电机的角度检测方式有多种方式，如磁极角度检测，其原理是在转轴的中心周上安装一个圆形磁铁，当电机旋转时带动磁铁旋转，再利用磁角度传感器检测磁极角度，这种方式安装和角度检测都较为方便，但是当希望将阀门的转轴方向作为一个可直观查看阀门状态的部件时，转轴的高度要接近控制器的壳体以方便安装方向指针盘，所以在本设计中放弃这种方式。本系统采用精密多圈电位器 3382G-1-103G 作为角度传感器，将此传感器穿入转轴，以转轴带动电位器旋转，电机、电路板、角度传感器的安装实物如下图：





电位器的中间抽头信号 ALPS_2T 接单片机的 ADC，单片机计算 ALPS_2T 的电压即可对应转轴的旋转角度。为使 ALPS_2T 信号不受电源负载波动带来的影响，3382G-1-103G 的电源端用 TL431 将其电压稳定在 3.0V，原理图如下图 2：



(图 2)

在单片机最小系统电路中，已经将 ADC 的参考电压设置为 3.3V，单片机读取 ADC

值并计算电压： $V_{out} = \text{ADC值} \times \frac{V_{ref}}{\text{ADC最大值}}$ ，转换为角度值： $\text{角度} = V_{out} \times \frac{\text{最大角度}}{V_{ref}}$ ，例如，

对于 360° 的双向方式： $\text{角度} = V_{out} \times 72$

为保证角度旋转的准确性，需要在电机的控制过程中快速的计算 ADC 的电压值，与已标定的角度电压值进行的比对，驱动电机旋转而变化的电压值与标定电压值之间就形成一个联动的 PID 闭环，电机旋转角度计算的 PID 闭环程序实现：

```
float PID_Update(PID_TypeDef *pid, float target, float current)
{
    float error = target - current;
    pid->integral += error;
    float derivative = error - pid->prev_error;
    float output = pid->Kp * error +
        pid->Ki * pid->integral +
        pid->Kd * derivative;
    pid->prev_error = error;
    return output;
}
```

8.2.3 阀门初始角度及角度标定

直流电机带动减速齿轮组的中心转轴为 D 型轴，且在出场时已经将 D 型轴



的方向设定为阀门的闭合方向一致，所以我们可以将这个方向标定 0° ，阀门顺时针旋转 90° 时为向左边打开，且可以 $0^\circ \sim 90^\circ$ 任意角度打开。阀门逆时针旋转 90° 时为向右边打开，且可以 $0^\circ \sim -90^\circ$ 任意角度打开。D 型轴的三种状态如下图：



8.2.4 角度电压计算程序

```
// 电压监控结构体
typedef struct {
    float current_voltage;
    float previous_voltage;
    float min_voltage;
    float max_voltage;
    uint32_t update_time;
} Potentiometer_Monitor_t;

// 初始化监控器
void Potentiometer_Monitor_Init(Potentiometer_Monitor_t* monitor)
{
    monitor->current_voltage = 0.0f;
    monitor->previous_voltage = 0.0f;
    monitor->min_voltage = 3.3f;
    monitor->max_voltage = 0.0f;
    monitor->update_time = 0;
}

// 更新监控数据
void Update_Potentiometer_Monitor(Potentiometer_Monitor_t* monitor)
{
    monitor->previous_voltage = monitor->current_voltage;
    monitor->current_voltage = Read_Potentiometer_Voltage_Filtered(5);
    monitor->update_time = HAL_GetTick();

    // 更新最小/最大值
    if(monitor->current_voltage < monitor->min_voltage) {
        monitor->min_voltage = monitor->current_voltage;
    }
    if(monitor->current_voltage > monitor->max_voltage) {
```



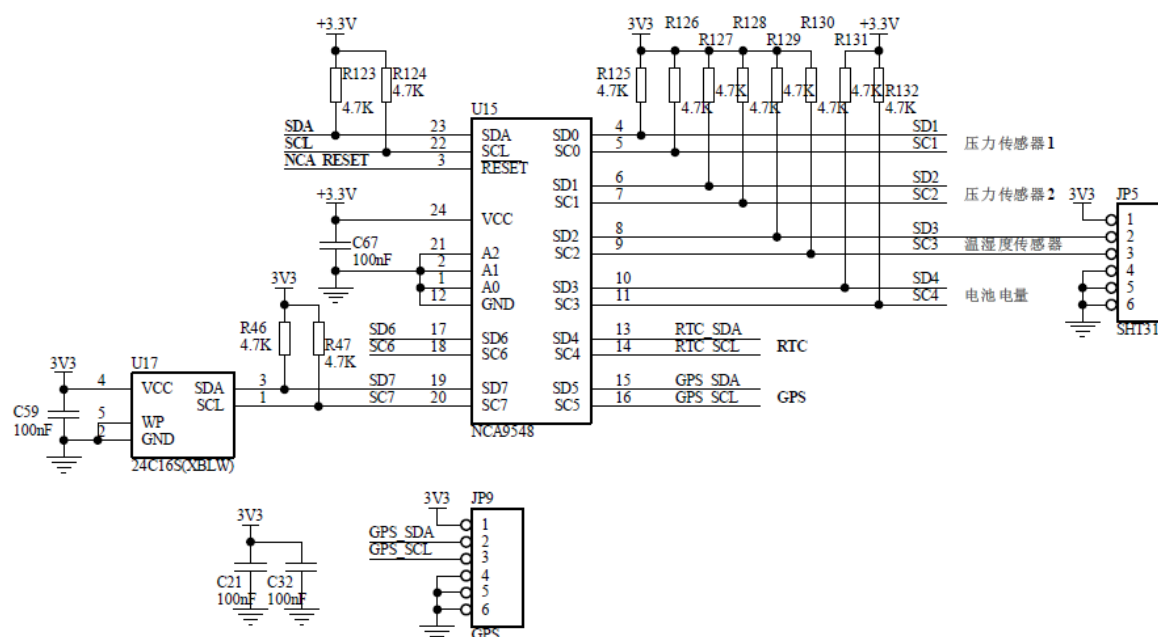
```

monitor->max_voltage = monitor->current_voltage;
}
}
// 检查电压变化是否超过阈值
uint8_t Is_Voltage_Changed(Potentiometer_Monitor_t* monitor, float threshold)
{
float delta = fabs(monitor->current_voltage - monitor->previous_voltage);
return (delta > threshold);
}

```

8.3 传感器及 I2C 接口扩展

在本系统中要连接多种传感器，部分传感器不带有 I2C 地址，所以需要将 I2C 接口用扩展芯片进行扩展，NCA9485 可以将 1 路 I2C 接口扩展成 8 路 I2C 接口，且芯片具有高精度、低噪声，带有双通到运算放大器，轨到轨输入输出，可为系统扩展 I2C 接口的 GPS 模块，两路气压传感器，1 路温湿度传感器，1 路电池电量芯片，也可以接入 1 路实时钟电路，1 路 EEPROM，电路原理图如下图：



驱动程序如下：

```

int main(void)
{
    HAL_Init();
    SystemClock_Config();
    // 初始化 ADC
    MX_ADC1_Init();
    // 配置 NCA948
    nca948.mode = NCA948_MODE_NON_INVERTING;
}

```

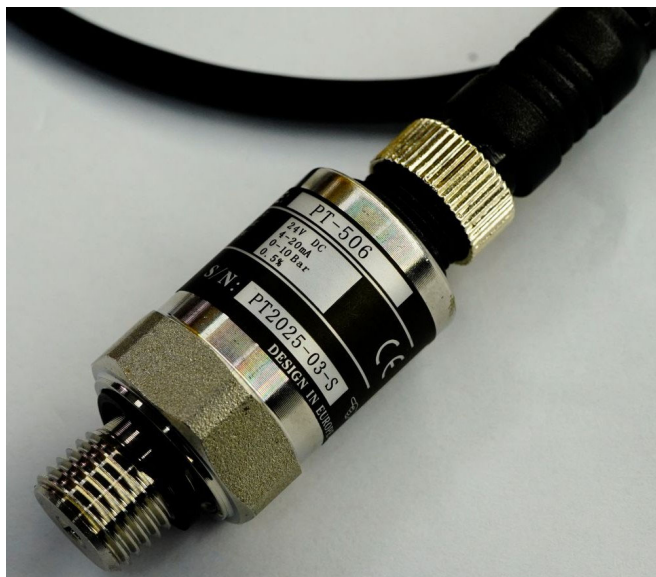


```
nca948.channel = NCA948_CHANNEL_A;
nca948.gain = 10.0f; // 10 倍增益
nca948.adc_channel = ADC_CHANNEL_0;
nca948.hadc = &hadc1;
// 初始化 NCA948
NCA948_Init(&nca948);
// 初始化滤波器
NCA948_Filter_Init(&voltage_filter, 0.1f); // 滤波系数 0.1
// NCA948 自检
if (NCA948_SelfTest(&nca948)) {
    printf("NCA948 自检通过\r\n");
} else {
    printf("NCA948 自检失败\r\n");
}
while (1)
{
    // 读取滤波后的电压值
    float voltage = NCA948_ReadVoltage_Filtered(&nca948, &voltage_filter, 5);
    // 计算实际输入电压 (考虑放大器增益)
    float input_voltage = voltage;
    printf("放大器输出: %.3fV, 输入电压: %.3fV\r\n",
        voltage, input_voltage);
    HAL_Delay(100); // 100ms 采样间隔
}
}
// ADC 初始化函数
void MX_ADC1_Init(void)
{
    ADC_ChannelConfTypeDef sConfig = {0};
    hadc1.Instance = ADC1;
    hadc1.Init.ScanConvMode = ADC_SCAN_DISABLE;
    hadc1.Init.ContinuousConvMode = DISABLE;
    hadc1.Init.DiscontinuousConvMode = DISABLE;
    hadc1.Init.ExternalTrigConv = ADC_SOFTWARE_START;
    hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
    hadc1.Init.NbrOfConversion = 1;
    HAL_ADC_Init(&hadc1);
    sConfig.Channel = ADC_CHANNEL_0;
    sConfig.Rank = ADC_REGULAR_RANK_1;
    sConfig.SamplingTime = ADC_SAMPLETIME_28CYCLES_5;
    HAL_ADC_ConfigChannel(&hadc1, &sConfig);
}
```

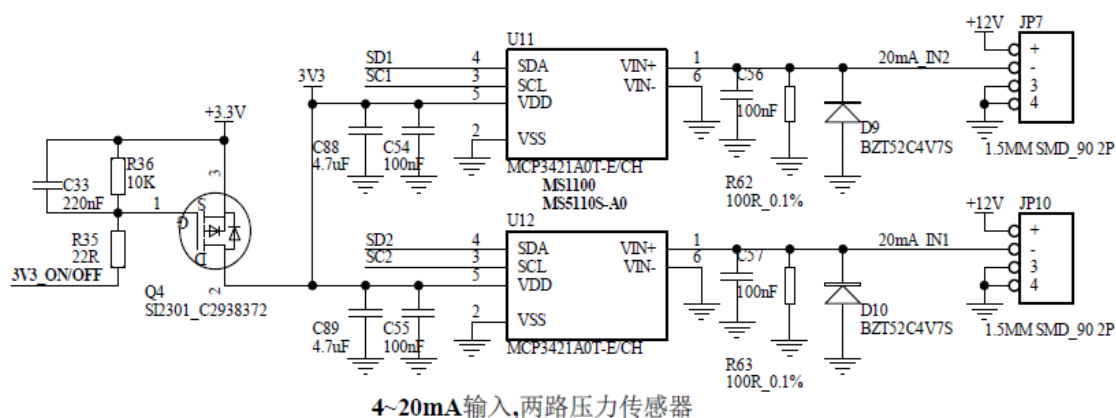


8.4 4~20mA 压力传感器采样

在本系统，水管压力是灌溉系统的一种必要监测的状态，当阀门打开或关闭时，或这水管破裂时都可体现水管压力的变化，以此推断灌溉系统的运行状态，水管压力的范围可在 0par~1Mpar, 选用两线制无源的 4~20mA 压力传感器可以很容易的接入我们的系统中，本系统中选用 PT-506, 这个传感器的工作电压范围 DC12~24V, 压力测量范围 0~10Bar, 也就是 0~1Mpar, 精度 0.5%:



4~20mA 采样芯片可选用 MCP3421, 这是一款 I2C 接口的 18bit 高精度单通道 Δ - Σ 模数转换器，且连续采样功耗低至 145uA, 非常适合我们的低功耗系统应用，原理图如下：



4~20mA输入,两路压力传感器

在这个原理图中，设计了两路压力传感器输入，所以有两个 MCP3421 做 ADC 转换，为了降低功耗，在 ADC 芯片的电源端增加了一个 Q4 PMOS 作为电源开关，只有在需要读取传感器的压力值时才使 ADC 芯片进入工作模式。

驱动程序如下：

```
#include "mcp3421.h"
```



```
#include <math.h>

// 构建配置字节
static uint8_t MCP3421_BuildConfig(MCP3421_HandleTypeDef* hmcp)
{
    uint8_t config = 0x00;
    config |= hmcp->mode;           // 模式位
    config |= hmcp->sample_rate << 2; // 采样率
    config |= hmcp->gain;           // 增益
    hmcp->config_register = config;
    return config;
}

// 初始化 MCP3421
void MCP3421_Init(MCP3421_HandleTypeDef* hmcp)
{
    // 设置默认配置
    hmcp->sample_rate = MCP3421_SPS_375; // 18 位精度
    hmcp->gain = MCP3421_GAIN_1;         // 增益 1
    hmcp->mode = MCP3421_MODE_ONESHOT;   // 单次模式
    // 构建配置字节
    MCP3421_BuildConfig(hmcp);
    // 发送初始化配置
    HAL_I2C_Master_Transmit(hmcp->hi2c, hmcp->address,
                            &hmcp->config_register, 1, HAL_MAX_DELAY);
}

// 设置配置参数
void MCP3421_SetConfig(MCP3421_HandleTypeDef* hmcp, MCP3421_SampleRate rate,
                      MCP3421_Gain gain, MCP3421_Mode mode)
{
    hmcp->sample_rate = rate;
    hmcp->gain = gain;
    hmcp->mode = mode;
    // 更新配置寄存器
    MCP3421_BuildConfig(hmcp);
    // 发送新配置
    HAL_I2C_Master_Transmit(hmcp->hi2c, hmcp->address,
                            &hmcp->config_register, 1, HAL_MAX_DELAY);
}

// 开始转换
bool MCP3421_StartConversion(MCP3421_HandleTypeDef* hmcp)
{

```



```
HAL_StatusTypeDef status;
// 发送配置字节开始转换
status = HAL_I2C_Master_Transmit(hmcp->hi2c, hmcp->address,
                                  &hmcp->config_register, 1, HAL_MAX_DELAY);

return (status == HAL_OK);
}

// 检查转换是否完成
bool MCP3421_IsReady(MCP3421_HandleTypeDef* hmcp)
{
    uint8_t rx_data[3];
    HAL_StatusTypeDef status;
    // 读取3个字节（18位模式）
    status = HAL_I2C_Master_Receive(hmcp->hi2c, hmcp->address, rx_data, 3, 10);
    if (status == HAL_OK) {
        // 检查RDY位（第7位）
        return ((rx_data[2] & 0x80) == 0x00);
    }
    return false;
}

// 读取转换结果
bool MCP3421_ReadConversion(MCP3421_HandleTypeDef* hmcp, MCP3421_ResultTypeDef* result)
{
    uint8_t rx_data[3];
    HAL_StatusTypeDef status;
    int32_t raw_value = 0;
    // 读取3个字节
    status = HAL_I2C_Master_Receive(hmcp->hi2c, hmcp->address, rx_data, 3, HAL_MAX_DELAY);
    if (status != HAL_OK) {
        result->is_ready = false;
        return false;
    }
    // 检查RDY位
    result->is_ready = ((rx_data[2] & 0x80) == 0x00);
    if (!result->is_ready) {
        return false;
    }
    // 解析18位数据（有符号）
    raw_value = (rx_data[0] & 0x03) << 16; // 高2位
    raw_value |= rx_data[1] << 8;          // 中8位
    raw_value |= rx_data[2];                // 低8位
    // 符号扩展（18位有符号数）
```

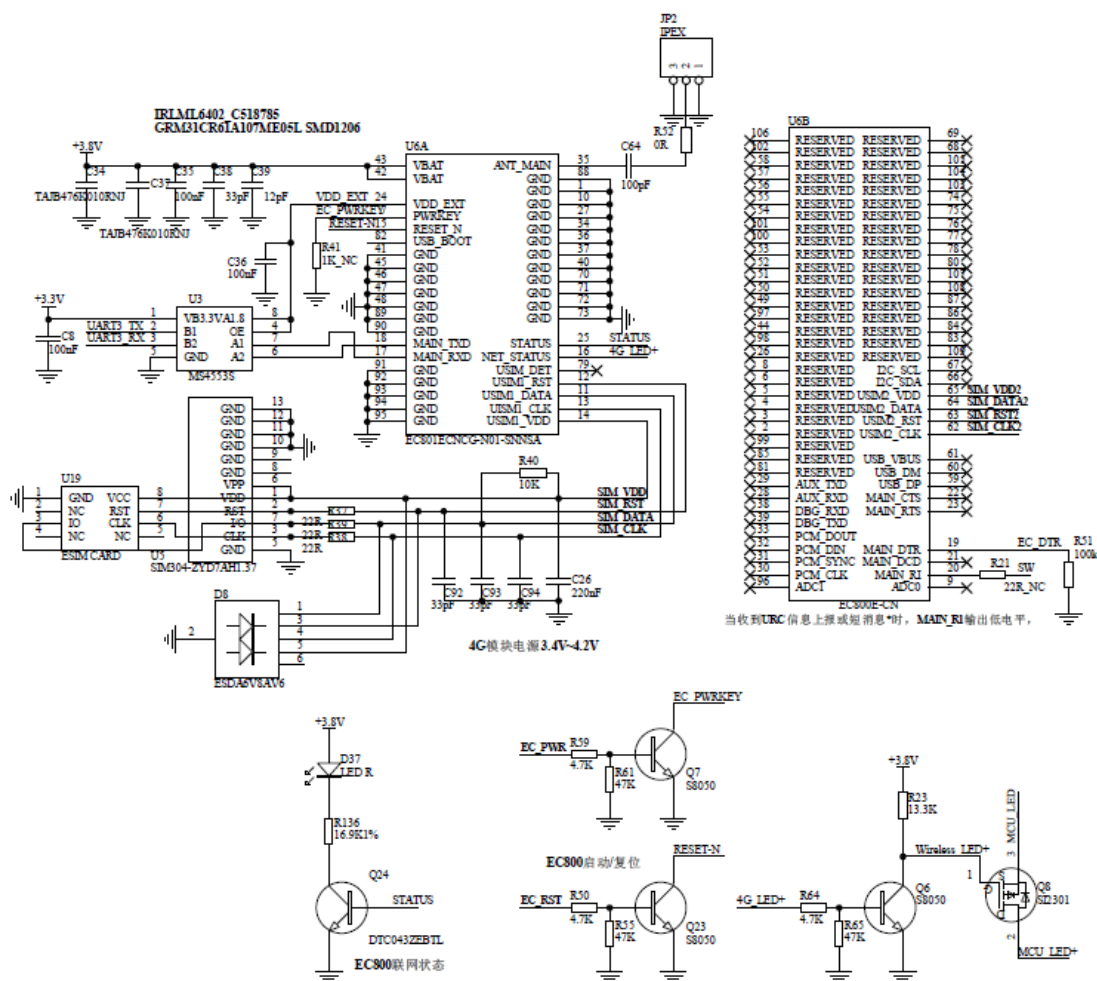


```
if (raw_value & 0x20000) { // 检查符号位 (第 17 位)
    raw_value |= 0xFFFC0000; // 扩展到 32 位有符号
}
result->raw_value = raw_value;
// 计算电压值
// LSB = 2.048V / (增益 * 2^(分辨率-1))
// 18 位分辨率: LSB = 2.048 / (增益 * 131072)
float lsb_size = 2.048f / ((1 << (hmcp->gain)) * 131072.0f);
result->voltage = raw_value * lsb_size;
return true;
}
// 读取电压值 (简化函数)
float MCP3421_ReadVoltage(MCP3421_HandleTypeDef* hmcp)
{
    MCP3421_ResultTypeDef result;
    // 开始转换
    if (!MCP3421_StartConversion(hmcp)) {
        return 0.0f;
    }
    // 等待转换完成 (超时处理)
    uint32_t timeout = 1000; // 超时时间
    while (!MCP3421_IsReady(hmcp) && timeout > 0) {
        HAL_Delay(1);
        timeout--;
    }
    if (timeout == 0) {
        return 0.0f; // 超时
    }
    // 读取结果
    if (MCP3421_ReadConversion(hmcp, &result)) {
        return result.voltage;
    }
    return 0.0f;
}
```

8.5 4G Cat1 通讯电路

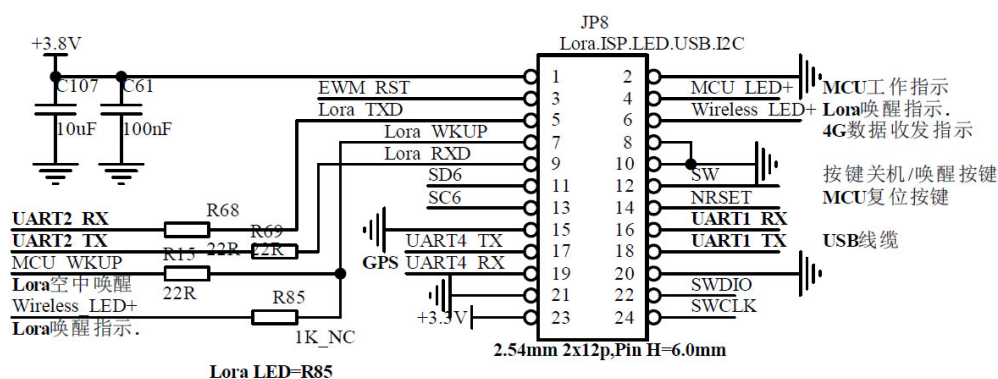
农业、农场的生产环境通常为城市的郊外，或者人口不集聚的偏远山区、乡村，所以要考虑通讯网络部署难度及成本，利用现有的移动通讯网络进行云端或远端对农用设备进行控制、数据采集就是最便利和可行的通讯方式。以当前我们通讯网络中 4G 的普及程度，设备控制的上行和下行的数据量，以及零件成本，选用 4G Cat1 最为合适。本系统中选用移远通讯的 EC801ECNCG-N01-SNNSA 模组作为

4G 通讯模块，原理图如下：



8.6 LoRa 通讯连接与系统扩展功能扩展接口

当系统不用接入云端服务器以本地上位机方式运行时，可选用远距离的 LoRa 通讯，在本设计中以扩展的方式接入，并留有扩展插座，并将单片机的 UART2 分配给 LoRa 模块，UART4 也可以用 GPS 模块的扩展，UART1 用做与上位机通讯接口，在这个扩展插座中还分配有单片机的工作指示灯、4G 模块联网指示灯、单片机唤醒、休眠、复位按钮，原理图如下图：



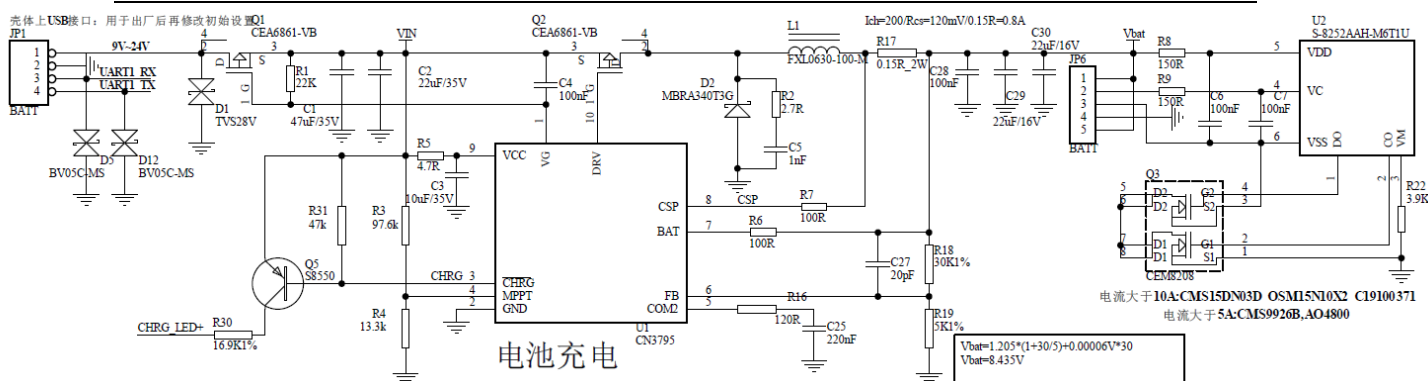
8.7 电源及电池充放电电路

农用的设备通常在田间、野外，用太阳能板供电是最容易实现的电能来源，同时要有可充电电池储备电能，以备在阴雨天或夜晚能持续供电，在电源电路设计时要考虑将太阳能板的电能以比较高的效率转换并充入电池中，所以我们在电源电路中采用了带有 MPPT 最大功率点跟踪的充电芯片 CN3795, 这个芯片支持 6.6~30V 的太阳能板输入，芯片的最大充电电流 4A。太阳能板的型号有比较大的选择范围，通常会选择大于 12V 的太阳能板，所以在电池的串并联组合上，我们采用两颗锂离子电池串联方式，串联后的电压是 7.4V~8.4V, 当要将电池电压升压到阀门的 12V 电压时，能获得比较高的转换效率。串联最高的充电电压设置在 8.4V，充电电流设置在 0.8A。

系统采用长虹超宽温度范围的 21700 电池锂离子电池，此电池可在 -40°~+75° 温度范围内工作，单颗容量 5Ah, 可应对我们国家大部分的气候环境：



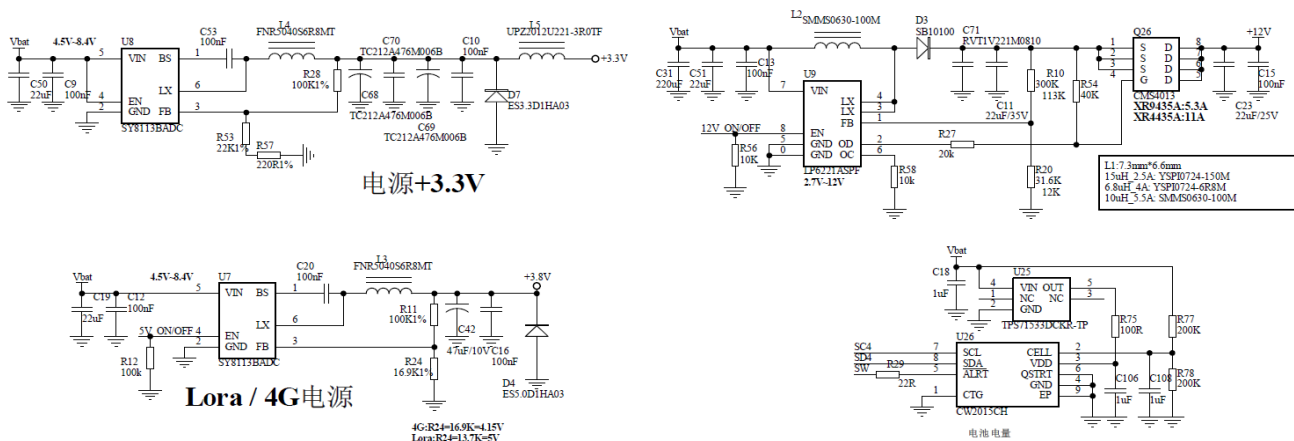
为确保锂离子在可靠、安全的工作，还应该考虑电池的过充、过压、过流保护，所以在电路中设计一个 S-8252AAH-M6T1U 芯片用作电压监测，其控制的低边 PMOS (CEM820) 可有效且快速的切断电池与负载的供电。



8.8 传感器及电机升降压电路

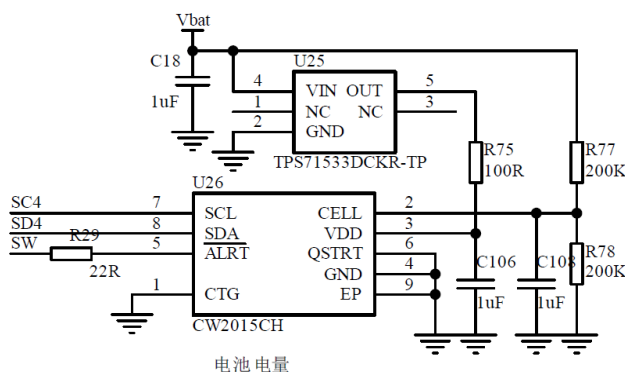
系统中单片机、ADC、等电路都用 3.3V 供电，4G Cat1、Lora 模块用 3.8V 供电，单片机的工作电流超过 20mA，4G Cat1 模块的启动电流接近 1A，电池的最高电压 8.4V，与 3.3V 之间有比较高的电压差，所以两电压之间的转换应该选用高效率的 DC/DC 降压电路，本设计中用两个 SY8113BADC 将电池电压将为 3.3V 和 3.8V。

L 型直流电机阀和电压阀的工作电压都是 12V，常见的低成本升压电路可选 Boost 类型，但类芯片在原理上不能完全关断电路的输出，电池会经过升压回路的电感和二极管造成能量消耗，芯片 LP6221ASPF 是一个带有输出关断的芯片，自带输出使能，可以在关断模式时关断输出，可以杜绝升压电路在关断模式对电池的消耗，电路如下图：



8.9 电池电量测量

在农业设备控制系统中，电池的电量直接关系到系统的可靠运行，即时获知电池电量信息可谓系统管理带来便利，所以电池电量的监控是一个很有必要的功能，本设计中采用 CW2015 这个芯片作为电池电量监控，这个芯片根据预先测试获得的电池模型推算电池的 SOC，原理图如下：



以长虹 21700 电池为例制作的电池模型数组：

```
#define SIZE_BATINFO 64
static unsigned char cw_bat_config_info[SIZE_BATINFO] = {
    0x15, 0x39, 0x78, 0x3C, 0x2F, 0x4D, 0x62, 0x69, 0x64, 0x54,
    0x51, 0x59, 0x58, 0x5A, 0x61, 0x62, 0x63, 0x5C, 0x5A, 0x6A,
    0x6D, 0x72, 0x72, 0x7F, 0x83, 0x92, 0x06, 0x66, 0x38, 0x59,
    0x6F, 0x86, 0x8F, 0x9D, 0x9E, 0xA8, 0x48, 0x1F, 0xFF, 0x5B,
    0x09, 0x74, 0x18, 0x48, 0x7E, 0x93, 0x9D, 0x42, 0x4C, 0x7E,
    0x97, 0x97, 0x80, 0x64, 0x52, 0xCB, 0x2F, 0x00, 0x64, 0xA5,
    0xB5, 0xC1, 0x46, 0xAE
};
```

第九章 系统传感器、功耗和通讯测试

9.1 上位机程序

要测试系统完整的功能，需要设计一个电脑端上位机程序，程序兼顾了系统各传感器的测量反馈值，阀门的初始角度校准，系统的初始化设置，上位机程序界面如下图：



调试工具

通信接口

串口号: COM16 - USB-SERIAL CH340 波特率: 115200

关闭串口 查询所有

设备信息 4G 联网信息 阀门控制 固件升级

设备重启

温度: 24.2 °C 湿度: 52.9 % 查询

电池电量: 100 % 电池电压: 4190 mV 电量低告警: 10 % 查询 设置

压力1: 31.4 kPa 压力2: 0.0 kPa 0-300kPa 0-600kPa 0-1Mpa 查询 设置里程

压力1: 5.68 mA 压力2: 0.00 mA

压力1: 0.565 V 压力2: 0.004 V

设备日期时间: 2000-01-01 23:04:19 查询 校正时间

自动开机时间: 09:30:00 查询 设置

自动关机时间: 00:00:00

设备类型

A-电动双通形球阀(电机齿轮箱驱动)

附带传感器

☒ 温度传感器 ☒ 湿度传感器 ☐ 风速传感器 ☐ 风向传感器 ☐ 光照度传感器 ☐ 降雨量传感器 ☐ 酸碱度传感器 ☐ 一氧化碳传感器 ☐ 二氧化碳传感器 ☐ 氧气浓度传感器

通信方式

☒ 4G CAT1 ☐ Lora ☐ WiFi

查询

9.2 角度旋转精度

在角度控制的算法中使用了 PID 算法，即测量与旋转动作是一个多次循环的闭环程序，以 -90° 为例，可以看到角度旋转的精度误差为 1° ，从传感器获取的电压值误差在 10mV 以内，是一个高精度的角度控制程序。

阀门角度: -90.7 度 角度电压: 1412 mV 状态: 停止 查询

角度-90校准 角度0校准 角度90校准

阀门角度: -90.0 度 角度电压: 1407 mV 状态: 停止 查询

角度-90校准 角度0校准 角度90校准

阀门角度: -89.6 度 角度电压: 1408 mV 状态: 停止 查询

角度-90校准 角度0校准 角度90校准

9.3 气压传感器测量精度

系统支持连接两个气压传感器，我们以一个 0~300kpa 气压传感器作为测试对象，可以看到压力值和电压值是非常稳定的，得益于系统中使用了一个 18bit 的 ADC 芯片：



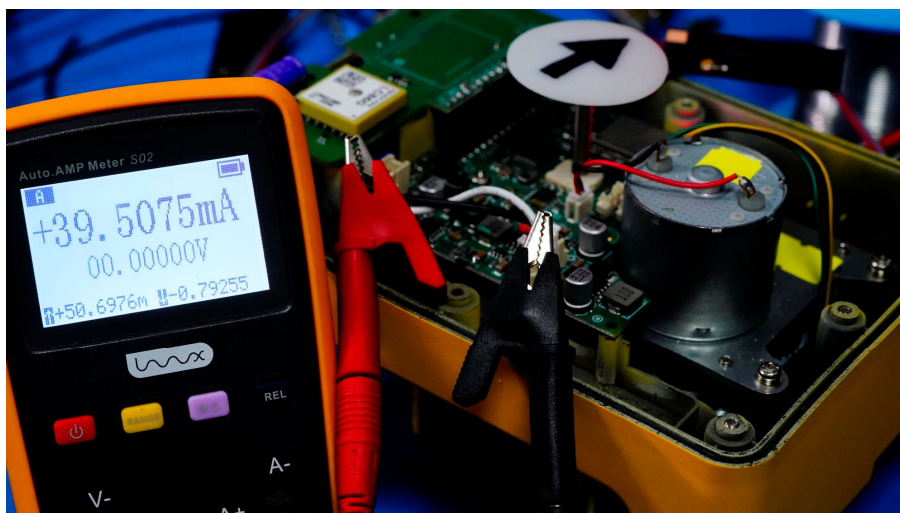
压力1:	31.3	kpa	压力2:	0.0	kPa	☒ 0-300kpa	查询 设置里程
压力1:	5.67	mA	压力2:	0.00	mA	☐ 0-600kpa	
压力1:	0.565	V	压力2:	0.004	V	☐ 0-1Mpa	

压力1:	31.2	kpa	压力2:	0.0	kPa	☒ 0-300kpa	查询 设置里程
压力1:	5.67	mA	压力2:	0.00	mA	☐ 0-600kpa	
压力1:	0.564	V	压力2:	0.004	V	☐ 0-1Mpa	

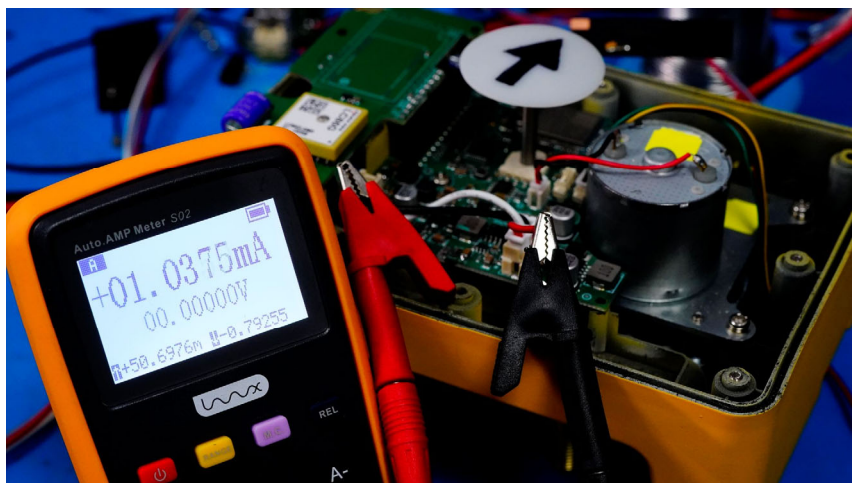
压力1:	31.4	kpa	压力2:	0.0	kPa	☒ 0-300kpa	查询 设置里程
压力1:	5.68	mA	压力2:	0.00	mA	☐ 0-600kpa	
压力1:	0.565	V	压力2:	0.004	V	☐ 0-1Mpa	

9.4 低功耗测试

阀门控制器通常部署于分散、无人值守的应用环境，其供电条件受限，系统需长期稳定运行。低功耗设计不仅能够显著延长设备续航时间、降低运维成本，而且有助于提升系统的可靠性和稳定性，是阀门控制器实现规模化、智能化和工程化应用的关键技术之一，本系统在正常工作时的电流约 40mA：



休眠时的电流测试

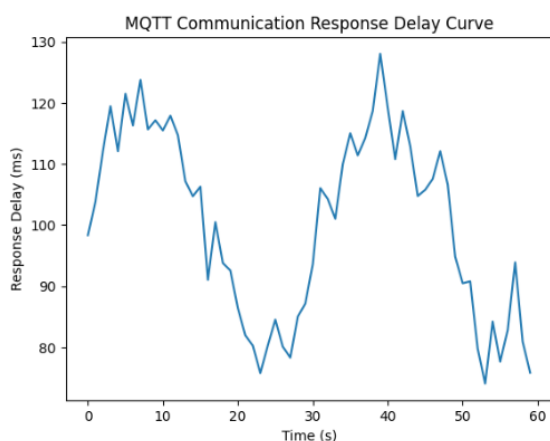


休眠电流是包含了电路上各路 DCDC 电路的静态电流，GPS 模块、4G 模块的休眠电流，电流接近 1mA, 所以本系统具有比较的休眠功耗。系统辅以一个 12V100mA 的太阳能板供电即可应对白天的电能消耗以及夜晚日照时的功率消耗。无日照时的功率消耗计算： $W = \text{工作电流} \times \text{无日照时间}$ ，以 12 小时夜晚作为无日照时间为例， $W = 40\text{mA} \times 12\text{h} = 480\text{mAh}$ ，2V100mA 的太阳能板可以用一半以上的发电功率给电池补足夜间电能的消耗。

9.5 通讯响应测试

本系统最终采用 MQTT 协议与服务端通讯，通讯延迟范围在 75~130ms。

从 4G 模块发起网络连接信令开始至 60 秒的区间内，是一个发布（Publish）到接收响应（Subscribe/ACK）的时间过程，其实是一个网路抖动和服务器负载变化的过程，测试曲线图如下：



由上图可以看出，系统通讯的延迟整体保持在较低水平，仅在个别时刻出现小幅波动，表明 MQTT 协议在该系统中具有较好的实时性和稳定性，能够满足阀门控制器对远程控制与状态反馈的时延要求。



结束语

本论文围绕基于 ARM Cortex-M4 内核单片机的农用设备控制系统展开研究，从系统架构、控制算法、通信机制、能耗优化到仿真性能评估进行了系统性的分析与设计。研究表明，ARM Cortex-M4 平台在运算能力、实时性与低功耗之间取得良好平衡，能够有效满足农业设备在复杂环境下的监测与控制需求。所提出的多级低功耗策略与可靠性机制显著提升了系统的续航能力与稳定性，仿真结果进一步验证了所设计控制方法与通信方案的可行性和有效性。

尽管本研究取得了较为全面的理论成果，但仍存在进一步优化空间，如在未来工作中，可引入更智能的自适应算法，提高系统在多变量耦合与强不确定性农业环境中的自主管理能力；进一步探索边缘计算与大规模节点协同，以提升整体网络的可靠性与实时性；同时，结合实际农田部署场景开展长期实验，将更有助于推进本系统的工程化与规模化应用。

本研究为智慧农业设备控制系统的开发与优化提供了理论基础，也为后续深入研究奠定了重要支撑。



参考文献

- [1], 朱会霞,《基于无线传感器网络的精细灌溉自动控制系统》.《农机化研究》2012 年第 34 卷第 1 期.
- [2], 王立坤, 侯振强, 徐晓等,《基于物联网的智能灌溉系统设计》.《农机化研究》2014 年第 36 卷第 11 期.
- [3], 李迅, 刘俊, 蔡肯,《农田土壤水分无线传感器网络节点布局优化》.《农机化研究》2012 年第 34 卷第 9 期, 第 106-109 页.
- [4], 付雪峰, 戈振扬, 张汝坤,《基于模糊控制的节水灌溉系统研究》.《农机化研究》2009 年第 32 卷第 1 期, 第 154-157 页.
- [5], 刘玮, 刘涛, 刘鹏,《基于 MQTT 的物联网通信系统的设计与实现》.《计算机测量与控制》.2017 年第 25 卷第 5 期, 第 298-301 页.
- [6], 谢涛, 陈志聪, 陈俊仁,《基于 STM32 和 NB-IoT 的农业环境监测系统设计》.《现代电子技术》2019 年第 42 卷第 15 期, 第 42-45, 50 页.
- [7], 谢涛, 苏丽媛, 陈俊仁,《基于 STM32 和 MQTT 的智能家居控制系统》.《龙岩学院学报》2019 年第 37 卷第 2 期, 第 52-58 页.
- [8], 杨建全, 李振华, 赵继春,《基于 STM32 和 LoRa 的智慧农业监控系统》.《电子测量技术》2019 年第 42 卷第 15 期, 第 115-119 页.
- [9], 王永顺, 刘春涛,《两线制 4~20mA 信号采集接口的设计》.《化工自动化及仪表》2010 年第 37 卷第 12 期, 第 117-119 页.
- [10], 李学海,《I²C 总线在单片机系统中的应用》.《电子技术应用》2000 年第 26 卷第 9 期, 第 66-68 页.
- [11], 张睿, 吴勋,《24 位 $\Sigma-\Delta$ ADC 与高速单片机的接口设计》.《国外电子测量技术》2007 年第 26 卷第 2 期, 第 40-42, 46 页.