

G · 格子达论文检测报告【全文标注】

报告编号:35E914A0726B464995084C6DA3B7DE68

作者:夏庆生

送检时间:2021-06-04 14:51:45

送检单位:安徽建筑大学

送检文档:多参数水质监测系统研制

对比索引检测检测范围:

1989-01-01至2021-06-04

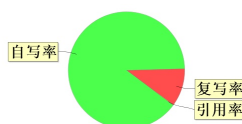
学术期刊库 近百亿+	学位论文库 13亿+	会议论文库 7000万+	报纸资源库 900万+	互联网资源库 数千万+
共享资源库 数百亿+	本科论文共享库 近亿+	大学生同届库 实时更新	专利库 近亿+	格子达论文库 30亿+

检测结果:

总相似比:10.57%

论文总字数:14673个

检测指标:



自写率:89.43%

复写率:10.57%

引用率:0.0%

其他类型检测结果:去除引用后总相似比:10.57% 同校同届总相似比:3.13%

其他指标:

图件	表格数	引用句子数	相似片段数
63	5	0	59

复写率来源:	学术期刊:0.0%	学位论文:0.21%	会议论文:0.4%
	报纸资源:0.0%	互联网:1.5%	大学生本科论文:8.46%
	共享资源库:0.0%	专利库:0%	

复写相似文献列表:

序号	相似文献	相似片段数	相似字数	相似比
1	来源:大学生本科毕业论文联合共享库 出处: 2017	14个	519	4.02%
2	来源:大学生本科毕业论文联合共享库 出处: 2018	5个	194	1.5%
3	来源:大学生本科毕业论文联合共享库 出处: 2019	1个	60	0.46%
4	篇名:基于STC89C52的鱼塘水质检测系统 来源:互联网论文库 出处:http://www.baidu.com/link?url=MfzUYrH1iEPtJBK-w... 0	1个	57	0.44%

序号	相似文献	相似片段数	相似字数	相似比
5	来源:大学生本科毕业论文联合共享库 出处: 2020	1个	55	0.43%
6	篇名:STM32F415RG - STMicroelectronics 来源:互联网文库 出处:http://www.baidu.com/link?url=_eS1_E6Y0Y3wZnKiq...	1个	54	0.42%
7	篇名:多路温、湿度传感器检测系统设计 来源:会议论文库 作者:王旭 刘宗尧 孔诗媛 出处:第34届中国气象学会年会 2017	1个	51	0.4%
8	来源:大学生本科毕业论文联合共享库 出处: 2019	1个	52	0.4%
9	来源:大学生本科毕业论文联合共享库 出处: 2019	1个	47	0.36%
10	来源:大学生本科毕业论文联合共享库 出处: 2019	1个	44	0.34%
11	来源:大学生本科毕业论文联合共享库 出处: 2020	1个	40	0.31%
12	来源:大学生本科毕业论文联合共享库 出处: 2020	2个	35	0.27%
13	来源:大学生本科毕业论文联合共享库 出处: 2017	1个	31	0.24%
14	篇名:SF6分解产物检测装置的研制 来源:学位论文库 作者:李智 出处:硕博学位论文 2011	1个	27	0.21%
15	篇名:列车行车安全监控无线数据传输系统的研究 - docin.com豆丁网 来源:互联网文库 出处:http://www.baidu.com/link?url=mWwOGJqjJ4zBBpC8y...	1个	19	0.15%
16	篇名:1.3本课题的研究目的及意义_前沿Lab_分析测试百科网 来源:互联网文库 出处:http://www.baidu.com/link?url=18cvyeo6EKGfyOMnN...0	1个	17	0.13%
17	篇名:第03章Arduino基本函数-豆丁网 来源:互联网文库 出处:http://www.baidu.com/link?url=z9y0_pejkA6XSB8nD...0	1个	15	0.12%
18	篇名:【推荐】-》电导率传感器发展概况 - 技术总结 - 道客巴巴 来源:互联网文库 出处:http://www.baidu.com/link?url=-HY2GJqjJ4zBBpC8y...	1个	16	0.12%
19	来源:大学生本科毕业论文联合共享库 出处: 2018	1个	15	0.12%
20	篇名:电导率传感器的原理_电导率传感器使用方法-电子发烧友网 来源:互联网文库 出处:http://www.baidu.com/link?url=Pfenlp9rfhHeudUap...0	1个	16	0.12%

同届相似文献列表:

序号	学院	院系专业	相似文献	相似片段数	相似字数	相似比
1	安徽建筑大学	电子与信息工程学	篇名:植物智能养护系统的设计	4个	62	0.48%

序号	学院	院系专业	相似文献	相似片段数	相似字数	相似比
		院 通 信工程				
2	安徽建筑大学	机械与电 气工程 学院 机械电子工程	篇名:17210070206、崔梦翔、轮式车辆的控制系统设计	1个	58	0.45%
3	安徽建筑大学	电子与信 息工程 学院 物联网工程	篇名:冷链运输保温箱在线监测节点的设计	1个	36	0.28%
4	安徽建筑大学	电子与信 息工程 学院 通信工程	篇名:刘欢欢毕业论文222	1个	25	0.19%
5	安徽建筑大学	电子与信 息工程 学院 物联网工程	篇名:基于单片机的安全报警器的设计 (1)	1个	19	0.15%
6	安徽建筑大学	机械与电 气工程 学院 机械电子工程	篇名:17210070145_柏家正_多功能数字时钟系统设计	2个	20	0.15%
7	安徽建筑大学	电子与信 息工程 学院 建筑电气与智能化	篇名:17211040233-王灵灵-基于单片机的恒温控制箱系统设计	1个	19	0.15%
8	安徽建筑大学	电子与信 息工程 学院 通信工程	篇名:基于单片机的颜色识别系统的设计与实现	1个	18	0.14%
9	安徽建筑大学	环境与能 源工程 学院 能源与动力工程	篇名:工业余热用于集中供热研究-毕业论文	1个	18	0.14%
10	安徽建筑大学	外国语学院 电子信息工程	篇名:陈乐-17205010101-一种儿童滞留车内安全报警系统的研制	1个	18	0.14%
11	安徽建筑大学	机械与电 气工程 学院 电气工程及其自动化	篇名:基于单片机的电缆沟监测系统	1个	17	0.13%
12	安徽建筑大学	机械与电 气工程 学院 电气工程及其自动化	篇名:毕业设计初稿-王小虎	1个	17	0.13%
13	安徽建筑大学	机械与电 气工程 学院 电气工程及其自动化	篇名:17210020216-李利-农业大棚温湿度自动监测系统	1个	15	0.12%
14	安徽建筑大学	外国语学院 电子信息工程	篇名:毕业论文查重	1个	16	0.12%
15	安徽建筑大学	电子与信 息工程 学院 通信工程	篇名:储德祥毕设论文6.4 - 副本	1个	16	0.12%

序号	学院	院系专业	相似文献	相似片段数	相似字数	相似比
16	安徽建筑大学	外国语学院 电子信息工程	篇名: 17205010142张恒源基于射频的门禁系统设计	1个	15	0.12%
17	安徽建筑大学	公共管理学院 人力资源管理	篇名: 某科技教育有限公司人力资源管理体系优化	1个	15	0.12%

全文内容:



安徽建筑大学
ANHUI JIANZHU UNIVERSITY

毕业论文

题目: 多参数水质监测系统研制
姓名: 夏庆生
学号: 17205040229
学院: 电子与信息工程学院
专业: 通信工程
指导教师: 徐荃
完成时间: 2021年6月2日

目录

第一章 绪论1

1.1 本课题背景2

1.2 本课题的研究现状及发展趋势2

1.3 本课题的研究目的及意义3

1.4 本课题的主要研究工作3

第二章 系统方案设计4

2.1 系统功能分析4

2.2 系统方案选择4

2.3 系统硬件框图5

第三章 硬件电路设计7

3.1 原理图绘制软件简介7

3.2 硬件电路总原理图7

3.3 STM32单片机主控模块7

3.3.1 STM32单片机简介7

3.3.2 STM32最小系统原理图8

3.4 DS18B20温度传感器10

3.4.1 DS18B20简介10

3.4.2 DS18B20电路原理图11

3.5 PH值传感器11

3.5.1 PH值传感器简介11

3.5.2 PH值传感器电路原理图12

3.6 浑浊度传感器13

3.6.1 浑浊度传感器简介13

3.6.2 浑浊度传感器电路原理图14

3.7电导率传感器14

3.7.1 电导率传感器简介14

3.7.2 电导率传感器电路原理图15

3.8液晶显示模块16

3.8.1 液晶显示屏简介16

3.8.2 液晶模块电路原理图20

第四章 软件程序设计23

4.1 开发软件简介23

4.2 系统主程序设计23

4.3 DS18B20温度传感器模块设计25

4.4 PH值、浑浊度、电导率传感器模块设计27

4.5液晶模块设计29

第五章 系统制作与调试31

5.1 系统硬件制作31

5.2 软件调试32

第六章 总结与展望34

致谢35

参考文献36

附录 系统源程序38

绪论

本课题背景

水资源是人类赖以生存的自然资源之一，然而，近些年来，全球许多地区的水污染问题已成愈演愈烈之势，有些地方的生活用水已被严重污染，以至于根本无法供人类正常生产生活所用，必须要经过复杂的污水处理工序才能被使用。因此，如何确保水质安全已逐渐成为一项全民关注的热点问题。

我们在日常生活中，通常既不清楚自己生活用水的来源，也不清楚自己饮用水的卫生状况如何，以及究竟是否可以供我们正常生产生活所用。为此，本文从社会需求出发，根据人们对于饮用水的卫生需求，设计了一款基于STM32F407ZGT6芯片的多参数水质检测系统，它可以从所取水样中实时采集温度、pH值、电导率、浑浊度等数据，经分析处理后，以一种简单易懂的方式将其显示在液晶屏上以作监测之用。

除了温度、pH值、电导率和浑浊度之外，表征水质状况的参数还有游离氯、溶解氧等，但这些参数测量起来比较复杂，而温度、pH值、电导率和浑浊度这四项参数指标已基本可以反映我们日常饮用水的水质状况。

本课题的研究现状及发展趋势

过去几十年来，“智慧水”作为智慧城市的一个重要组成部分被大众熟知，因此，在近些年来，人们不断提出新的有关于水资源管理、检测和治理的方法，在论文《基于物联网的实时饮用水管道污染检测系统》中，作者们提出了一种监测水管中水质的方法：使用智能传感器实时测量水管中的数据，使用ZigBee通信，并将LED显示屏用于显示结果，但用户需要手动检查各项参数。在论文《一种低功耗远程实时监测评估水质系统》中，作者提出了一种远程实时水质检测的低功耗系统。在论文《基于STC89C52的鱼塘水质检测系统》中，作者设计了一种可以检测并调节浑浊度、温度以及水深的水质检测系统，其以STC89C52作为主控芯片，使用DS18B20作为温度传感器，采用AD转换器来采集数字信号并处理，同时系统还具有报警功能。

根据上述的研究现状综述，不难发现，目前国内外关于水质检测呈现出以下发展趋势：

参数多元化。水体中含有多种矿物质及多种元素，因而也会呈现出不同的物理和化学特性，显然要想较为准确地描述一份水样的具体水质状况，参数越多，则描述越为精确。

标准化。目前制作一种简易实用的水质检测系统已有多套较为成熟的设计方案，器件种类也基本相对固定。

成本低。随着制造工艺的提升及检测系统设计方案标准化规范化，目前无论是批量生产还是自制水质检测系统所花费的成本都越来越低，并且检测精度越来越高。

1.3本课题的研究目的及意义

基于单片机的多参数水质检测系统会给人们的生产生活带来许多便捷，实时数据测量、传输、存储和分析的

成本更低。我们可以开发出一种即插即用的产品，它可以被放置在任意的水源处，并测量一些重要的指标，进行分析并向用户展示分析结果，使得即便是非技术人员也能够了解检测的水质情况。类似于这样的技术方案有助于提升大众卫生意识，用户们将会了解水质污染所带来的影响及危害。

1. 4本课题的主要研究工作

本文主要研究基于STM32F407ZGT6单片机的多参数水质参测系统，主要实现的功能是实时检测水样中的温度、pH值、电导率和浑浊度。本系统的设计流程为：查阅文献资料确定系统设计方案及器件选材购买元器件硬件系统搭建绘制原理图源程序开发系统调试。其中，各步骤主要内容如下所示：

查阅中英文文献资料，确定一个简易的水质检测系统应包括哪些模块。

查阅相关器件手册及传感器数据手册，了解元器件参数，确定系统设计方案。

根据系统设计方案及所选元器件型号购买器件。

根据器件手册上的引脚说明将传感器和显示模块分别同开发板连接起来。

用Altium Designer软件根据各个模块手册上的参考原理图及元器件封装库绘制电路原理图。

查阅STM32F407数据手册及传感器数据手册，据此编写各个外设及传感器模块的驱动程序（其中I/O口的配置与步骤3中的硬件连接要相对应）。

检查各个模块的硬件连接，将源程序下载到开发板并调试。

方案设计

2. 1系统功能分析

本文旨在设计一种能够实时精确检测水质参数并向用户反馈的水质检测系统。如前所述，本系统应能够同时检测所取水样中的温度、浑浊度、pH值和电导率，并直观地显示在显示屏上，以供用户实时查看。

2. 2 系统方案选择

方案一：

采用STC89C51作为主控芯片，外加水质参数传感器模块、ADC转换芯片、DMA芯片以及1602显示屏。

STC89C51系列单片机是一种8位微控制器，内嵌64KB的Flash，支持ISP(In-System-Programming)，即系统内编程—支持用户在系统内修改程序，而不需将微控制器拆卸下来，芯片内部还有1280字节或512字节的片上RAM，以供程序开发备用。STC89系列单片机保留了标准80C51单片机的所有特征，另外，STC89系列单片机还有额外的I/O端口(P4)，片上晶振。但是，STC89C51没有片上ADC外设，以及DMA外设，如果其他模块需要用到这两个外设，则需要另外添加。它使用寄存器编程方法。

1602字符型液晶显示器是一种专门用于显示数字、字母、符号的点阵式LCD，目前常用的模块大小分为16×1, 16×2, 20×2和40×2行等。它具有显示质量高、体积小、重量轻以及功耗低等优点。常见的1602显示屏其驱动电压为3.3V或5V，工作温度为-20℃~+70℃，与MCU的通讯接口为8位或4位并行接口、3位串行接口。

方案二：

采用STM32F407ZGT6芯片作为主控芯片，由于该芯片内部集成了ADC、DMA、USART等外设，外设只需要接水质参数传感器模块和4.3寸的液晶显示屏即可。

STM32F407ZGT6是STM32F407家族的一员，它基于高性能的ARM Cortex-M4的32位RISC内核，工作主频高达168MHz，Cortex-M4内核有一个FPU(Floating point Unit)，即浮点运算单元，可以处理所有ARM版的单精度数据处理指令和数据类型。同时STM32F407家族芯片内嵌高速存储器，包含存储容量高达1MB的Flash，以及192KB的SRAM，4KB的后备SRAM。同时，STM32F407ZGT6内部还有3个12位ADC，3个I2C, 4个USART以及其他外设。它使用固件库函数编程方法。

4.3寸液晶显示屏的屏幕分辨率为800×480，采用16位并口8080时序，供电电压为5V和3.3V，需要同时供电，控制器为自带的NT35510控制器，触摸芯片为GT917S。

若采用方案一，即使用STC89C51作为主控芯片，则由于PH值、电导率和浑浊度传感器都需要配套ADC进行模数转换，而需要另外添加ADC外设，同时也要添加DMA外设，但如果使用STM32作为主控芯片，由于其内嵌ADC和DMA，则硬件系统搭建比较方便。另外，STC89C51使用寄存器编程方法，而STM32F407ZGT6使用固件库编程方法，后者由前者发展而来，采用固件库编程时开发程序更为快捷，基于STM32F407ZGT6的这些优点，本文使用它作为主控芯片。由于1602显示屏分辨率较低，一般只能显示两行，而本文所设计的检测系统需同时检测四种水质参数，至少需要同时显示四行信息，而4.3寸和4.5寸液晶显示屏的分辨率均可满足需求，从经济角度出发，本文选择

4. 3寸液晶显示屏。

综上所述，本文使用方案二来设计硬件系统。

2. 3系统硬件框图

如图2-1，温度传感器的输出数据为数字量，故可以直送到单片机的I/O口，而浑浊度、pH值和电导率传感器的输出数据均为模拟量，故均需通过模数转换器 (ADC) 再送往主控芯片STM32F407ZGT6中，最后再将这些数据统一发往液晶显示屏的显存中，以便后续显示

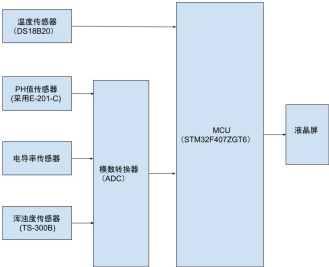


图2-1 系统硬件框图

硬件电路

原理图绘制软件简介

本文绘制原理图的工具为Altium Designer (AD)，它是由澳大利亚公司Altium Limited开发出来的，在2005年之前，这款软件名叫Protel，2005年之后更名为Altium Designer。AD画原理图步骤：

新建原理图文件。

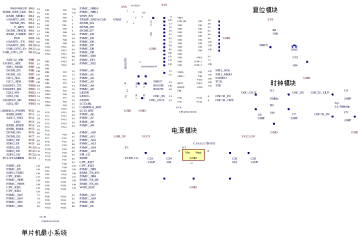
导入所要绘制的模块 (如STM32F407ZGT6最小系统版) 的元器件封装库。

放置元器件、布线、合理设置器件参数。

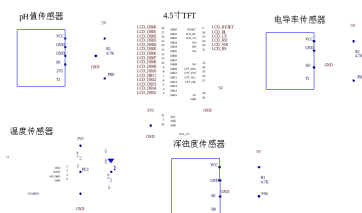
保存原理图。

硬件电路总原理图

图3-1 (a) 和图3-1 (b) 为硬件电路总原理图，由单片机最小系统和传感器以及显示屏模块组成。单片机最小系统为使用最少的元器件组成使单片机可以正常工作的系统，故最小系统由电源、时钟、复位和主控芯片模块四部分组成。而为了实现检测并显示水质参数至少还需要四个传感器外加一个显示屏模块。



(a) STM32最小系统



(b) 传感器及显示屏模块

图3-1 硬件电路总原理图

3.3 STM32单片机主控模块

3.3.1 STM32单片机简介

本文选用的单片机STM32F407ZGT6是STM32F4系列家族的一员,它采用高性能的ARM Cortex-M4内核,工作频率高达168MHz。**Cortex-M4内核的特征是有一个单精度浮点运算单元(FPU),支持各种ARM单精度数据处理指令和数据类型**,它也可以执行所有的DSP指令,另外,它还有一个可以提升应用安全的内存保护单元(MPU)。

STM32F4系列单片机拥有高速内存(Flash容量高达1MB,SRAM容量高达192KB),后备存储空间SRAM容量高达4KB,同时有许多I/O口和外设连接到2根APB总线,3根AHB总线和一个32位多AHB总线矩阵。

STM32F4系列单片机提供3个12位ADC,2个DAC,一个低压RTC,12个16位通用定时器。它们还提供一个随机数发生器(RNG),同时,该系列单片机拥有下列标准和先进的通信接口:

- 3个外设。

I^2C

· 3个SPI外设,2个采用全双工的S外设。为了实现声音分类的精度,S外设的同步时钟可由一个专用的内部音频PLL(Phase Locked Loops)或外部时钟提供。

I^2

- 4个USART外设和2个UART外设
- 2个CAN总线接口。
- 以太网和摄像头接口。
- 灵活的静态存储器控制接口FSMC(Flexible Static Memory Controller)。

STM32F407系列单片机的工作温度在-40~105℃,供电范围在1.8~3.6V之间。**但是当采用外部电源供电且设备工作在070℃之间时,供电电压可以降低到1.7V。**

:

3.3.2 STM32最小系统原理图

最小系统电路原理图如图3-2所示,包括STM32F407ZGT6主控芯片、复位电路、时钟电路、电源电路。

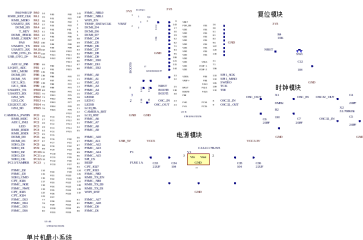


图3-2 STM32F407ZGT6最小系统原理图

图3-3为电源模块，电源模块负责整个系统包括单片机和传感器的供电，其中，USB_5V为USB接口，它的输入电压为5V，VCC5给需要5V电压的模块供电，F1为自恢复保险丝，用于保护USB，CJA1117B-3.3为稳压芯片，通过USB_5V输入的5V电压经过保险丝，再经过VCC5和稳压芯片，最终到达VCC3.3V输出口，可为系统提供3.3V的电压。



图3-3 电源模块

图3-4为时钟模块，OSC_out和OSC_IN引脚分别接单片机的PH0和PH1引脚，OSC_out和OSC32_IN引脚分别接单片机的PC14和PC15引脚。图中X1为晶振，晶振的全称为晶体振荡器，它的作用是为单片机提供基准时钟信号，晶振分为有源晶振和无源晶振，图3-4中使用的2个晶振均为无源晶振，分别为8MHz和32.768KHz，分别作为单片机的高速时钟输入和低速时钟输入。晶振两侧的引脚相当于电阻的两个引脚，分别接于单片机的两个相邻的晶振引脚上。晶振两侧所接电容的作用均为帮助晶振起振和稳定振荡信号，其容值一般选在10~40pF之间，这里我们选择经验值20pF。



图3-4 时钟模块

STM32单片机采用低电平复位，图3-5为低电平有效的复位电路。复位电路的NRST接口与主控芯片的25号引脚NRST相连。当按键断开时，在上拉电阻的作用下，NRST接口为高电平，相应地主控芯片NRST口也为高电平，主控芯片可以控制程序正常运行。当按键SW1按下时，复位电路的NRST接口与主控芯片的NRST接口同时变为低电平，由于电容两端的电压不能突变，故电容两端的电压会缓慢抬高，直到NRST接口变为高电平，复位过程结束，程序正常运行。



图3-5 复位模块

3.4 DS18B20温度传感器

3.4.1 DS18B20简介

DS18B20温度传感器提供9~12位(二进制)的测量精度，其可通过1-Wire接口总线收发数据。采用1-Wire总线连接时，中央处理器与DS18B20之间仅需通过一根总线来连接，这根总线可以完成所有的任务：供电、读写数据以及实现温度转换(不需要提供外部供电电源)。DS18B20温度测量范围为 $-55^{\circ}\text{C} \sim +125^{\circ}\text{C}$ ，并且在 $-10^{\circ}\text{C} \sim +85^{\circ}\text{C}$ 的测量范围内精度为 $\pm 0.5^{\circ}\text{C}$ ，足以满足大多数情况下测量范围与精度要求。它的一项重要特征是温度检测结果可直接在传感器模块内部转换为数字值，出厂默认的数字位数为12位，不需要通过ADC转换即可直接将测量结果传输给MCU。

3.4.2 DS10B20电路原理图

图3-6为温度传感器DS18B20的电路原理图，传感器模块使用3.3V供电电压，故电源模块的3V3引脚与DS18B20的VDD引脚相连，而DS18B20的DATA引脚输出的数据直接为数字量，故可直接与主控芯片的PE3引脚相连

，DS18B20的GND引脚接地，NC/GND引脚直接悬空。



图3-6 DS18B20电路原理图

3.5 pH值传感器

3.5.1 pH值传感器简介

酸碱度(pH值)是溶液的一种重要特性，本文所使用的pH值传感器模块由一个电压转换模块及pH复合电极组成。电压转换模块通过BNC接头与PH复合电极相连。pH复合电极的输出信号为mV级的电压信号，单片机无法识别并处理，而通过电压转换模块进行放大，就可以输出0~3V或0~5V输出信号，再通过ADC对此输出模拟信号进行采集得到数字信号。**调节电位器旋钮可以改变放大倍数**。模块电源为+5.00V, 模块尺寸为37mm×28mm，测量范围为0~14PH，测量温度为0~60℃，测量精度为±0.01PH(25℃)，响应时间1min。图3-7为pH传感器实物图：

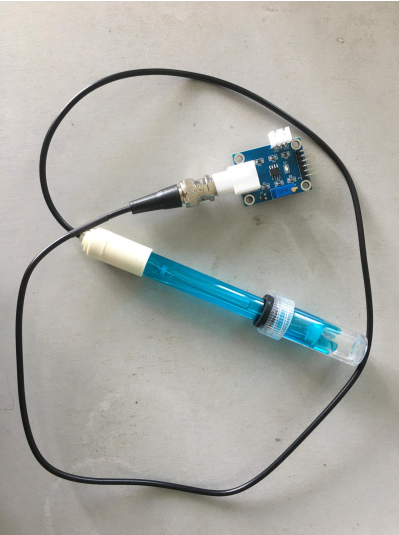


图3-7 pH传感器模块实物图

表3-1 电压转换模块基本参数

序号	引脚定义	功能描述	备注
1	VCC	供电电压正极，5V	不可使用3V3
2	GND	供电电压负极	
3	GND	模拟信号输出负极	
4	P0	模拟信号输出正极	输出电压范围
5	2V5	基准电压2V5输出口	测试用，不外接电源
6	T1	温度传感器DS18B20信号输出口	可通过软件进行温度补偿

3.5.2 pH值传感器原理图

图3-8为pH值传感器原理图，pH值传感器供电电压为5V，输出模拟量，VCC与电源模块的5V引脚相连，P0引脚接主控芯片的PB0引脚，2V5为基准电压2V5输出口, 用于测试，T1为温度补偿模块接口，本系统未使用2V5和T1引脚，故使其悬空。



图3-8 pH值传感器原理图

3.6 浑浊度传感器

3.6.1浑浊度传感器简介

浑浊度传感器模块由一个电压转换模块和浊度传感器TSW-30组成，电压转换模块通过3pin XH-2.54接头与浊度传感器TSW-30连接，电压转换模块的作用也是将TSW-30的输出电压转换到单片机可识别的范围内。本传感器模块同时具有数字量和模拟量输出接口，本文使用模拟量输出接口。模拟量可通过主控芯片中的ADC采集为数字量，从而得知所取水样的浑浊度。浑浊度传感器在出厂时会由厂家设定一默认阈值，用户也可在程序中自己设定阈值，一旦浑浊度达到预先设定的阈值，传感器模块上的绿色指示灯就会被点亮，传感器模块输出也会由高电平转变为低电平，观察指示灯的亮和灭或通过监测模块输出高低电平的变化就能发现水样的浑浊度是否超标。图3-9为浑浊度传感器模块：

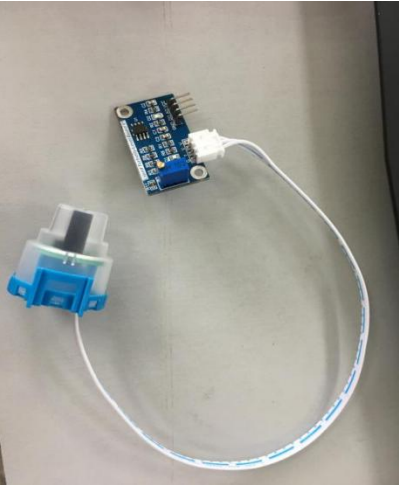


图3-9浑浊度传感器模块

浑浊度传感器模块参数如表3-2所示：

表3-2 浑浊度传感器基本参数

序号	引脚定义	功能描述	备注
1	VCC	供电电压，5V	不使用3V3
2	A0	模拟信号输出	输出电压范围0~5V

3	D0	数字信号输出	小于浑浊度设定阈值时输出高电平，大于设定阈值时输出低电平
4	GND	供电电压负极	与开发板GND引脚相连

3. 6. 2 浑浊度传感器电路原理图

图3-10为浑浊度传感器电路原理图，浑浊度传感器为5V电压供电，故VCC与电源模块的5V引脚相连，A0引脚为模拟量输出口，直接与单片机的PA6引脚相连，以供ADC采集数据，D0为数字量输出口，本系统不使用数字信号输出口，故D0引脚悬空。



图3-10 浑浊度传感器电路原理图

3. 7 电导率传感器

3. 7. 1 电导率传感器简介

专业的电导率检测设备价格昂贵，且缺少相关技术资料，本文所用电导率传感器模块如图3-11所示，由一个电压转换模块和电导率电极组成，电压转换模块通过BNC接口与电极相连，其作用与pH传感器及浑浊度传感器模块中的电压转换模块均相同。本传感器模块具有价格低廉和使用方便等优点，可用于评估水质、水产养殖和环境水样检测等领域。



图3-11 电导率传感器模块实物图

电导率传感器模块参数如表3-3所示：

表3-3电导率传感器基本参数

序号	引脚定义	功能描述	备注
1	VCC	供电电压正极，5V	不使用3V3
2	GND	供电电压负极	与开发板GND引脚相连
3	T1	温度传感器数据输出口	悬空
4	A0	模拟量输出口	输出电压范围0~3. 4V

3.7.2 电导率传感器电路原理图

图3-12为电导率传感器电路原理图，电导率传感器使用5V供电电压，故VCC引脚与电源模块的5V引脚相连，GND引脚接地，A0引脚为传感器模拟量输出口，与单片机PB1引脚相连，以供ADC采集，T1引脚为温度传感器数据输出口，本系统有单独的温度传感器DS18B20，故不使用T1，令其处于悬空状态。



图3-12电导率传感器电路原理图

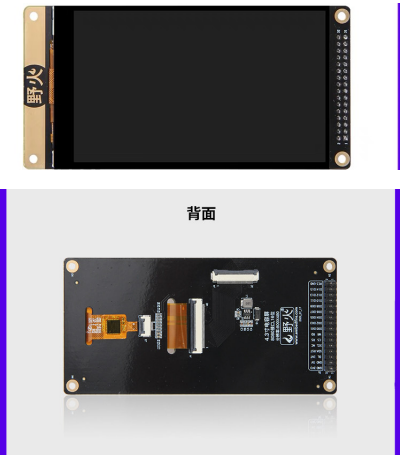
3.8 液晶显示模块

3.8.1 液晶显示屏简介

本文所用的显示屏为自带NT35510液晶控制器的野火4.3寸薄膜晶体管液晶显示屏(TFT LCD)，**屏幕分辨率为800×480**，它由以下三部分组成：**液晶显示面板、电容触摸面板和PCB底板**。电容触摸面板上有触摸控制芯片，PCB底板上带有液晶控制器。表3-4为液晶显示屏硬件参数，图3-13（a）和3-13（b）分别为液晶屏实物图的正面和背面。

表3-4 液晶显示屏硬件参数表

PCB尺寸	124*62mm
TFT	4.3寸、分辨率480*800
触摸	5点触控电容屏
接口	16位并口 8080时序
供电	5V和3.3V，需同时供电
背光	200mA即可点亮
帧率	75帧
控制器	自带NT35510控制器
颜色格式	RGB565
触摸芯片	GT917S



(a) 正面

(b) 背面

图3-13液晶屏实物图

图3-14为液晶屏与主控芯片连接示意图，本文所使用的液晶显示屏自带液晶控制器NT35510，且NT35510中含显存，当需要使用显示屏时，STM32通过FSMC外设模拟8080时序来传送要显示的字符数据到NT35510的显存中，NT35510再通过RGB接口向液晶面板传输像素数据。

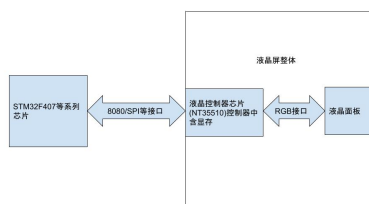


图3-14液晶屏与主控芯片连接示意图

3. 8. 2 液晶显示屏电路原理图

液晶显示屏电路原理图如图3-15所示，由于主控芯片采用FSMC外设模拟8080时序与液晶控制器NT35510进行通信，故显示屏引出的这些信号线实际上为8080通讯接口，其中LCD_DB[15:0]引脚用于传输数据信号，LCD_RD引脚用于传输读数据信号，低电平有效，LCD_RS引脚用于选择传输数据/命令信号，本引脚为高电平时，表示LCD_DB[15:0]传输的是数据，反之表示传输的是命令。**LCD_RESET引脚用于传输复位信号，低电平有效**，**LCD_WR引脚用于传输写数据信号，低电平有效**。LCD_CS用于传输片选信号，低电平有效。LCD_BL用于传输背光信号，低电平有效。

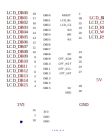


图3-15液晶显示屏电路原理图

第四章 软件程序设计

4. 1开发软件简介

本文所使用的软件开发环境是Keil MDK (Microcontroller Development Kit)，它主要包括MDK-Core以及Arm C/C++编译器。其中，MDK-Core基于，它支持所有的Cortex-M设备(本文使用的设备为Cortex-M4内核)，包括最新版的ARMv8-M架构。**ARM C/C++ 编译器包括汇编器、链接器以及高性能的运行环境库。用户也可以随时将软件包添加到MDK-Core中，从而支持新设备，并使中间件更新独立于工具链之外。**

μ Vsion

使用Keil软件编写源程序，在创建工程时需要根据所使用的芯片型号和仿真工具来设置下载和仿真环境，其余步骤与编写一个普通的C语言源程序相同，如下图所示：

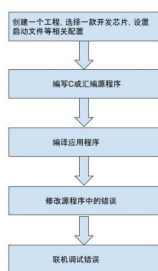


图 4-1 KEIL 软件开发程序流程图

4.2 系统主程序设计

本文所设计的水质参数检测系统主要检测四种参数：温度、浑浊度、pH值和电导率。由于对这四个参数的检测并显示是相对独立的过程，故采用模块化程序设计方法。同时由于温度检测模块驱动程序的设计方法与其他三种区别较大，pH值、电导率、浑浊度检测模块的源代码设计方法基本相同，而液晶显示方法原理均相同，故按照温度pH值电导率浑浊度，以及“先测量、后显示”的顺序编写源程序。

主函数流程图如图4-2所示:

声明并初始化主函数中需要的用到的变量：如定义一个显示缓冲区变量dis_buf，将要显示地字符插入dis_buf中即可。

初始化系统定时器、液晶控制器NT35510、ADC、DMA、LED端口，初始化各个传感器检测模块。

读取温度并送往显示缓冲区中，显示温度。

浑浊度、pH值和电导率检测模块程序设计原理均相同：ADC端口实时采集传感器模块输出的模拟量，并转化为数字量，再将数字量转化为可用于显示的字符，最后将要显示的字符插入到显示缓冲区dis_buf中即可显示。

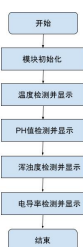


图4-2主函数流程图

主函数部分程序如下所示:

```

fclose(f);

/* 显示初始温度并显示 */
temperature=DS18B20_Get_Temp();

sprintf(char* dia_buf, "T=%.0 degree Celsius", temperature);
TM1630_DispatchingLine(EN_LDR1(1), dia_buf);

//2. 温度报警显示

ADC_ConvertedValueToLocal = (float) ADC_ConvertedValue/1024*3.3; // 读取转换的电压
VR_Value = 5.7541-ADC_ConvertedValue/LocalClock/10.4645;

if(VR_Value<0){VR_Value=0;}
if(VR_Value>10){VR_Value=10;}

/* 显示电压 */
TM_Write((uint8_t)VR_Value*1000+0.5);
TM_Write((uint8_t)(VR_Value*100+0.004+0.5));
TM_Write((uint8_t)VR_Value*10+0.004+0.5);
TM_Write((uint8_t)(VR_Value*1000+0.5)/10);
TM_Write((uint8_t)(VR_Value*100+0.5));
TM_Write((uint8_t)VR_Value); // 字符串结束标志

TM1630_DispatchingLine(EN_LDR1(1), "pH test ");
sprintf(char* dia_buf, "pH=%.0", TM_Write);
TM1630_DispatchingLine(EN_LDR1(1), dia_buf);

```

4.3 DS18B20温度传感器模块设计

温度传感器DS18B20需要单独编写驱动程序。编写DS18B20的驱动程序与编写其他所有外设驱动程序的方法类似：首先要配置DS18B20用到的通用输入输出接口口（GPIO口），即定义一个GPIO_InitTypeDef类型的初始化结构体，并对这个初始化结构体的成员一一赋值。之后还需要编写一些用于主控芯片与DS18B20之间通信的函数，如用于主机（MCU）给从机（DS18B20）发送复位脉冲的函数、检测从机给主机发送存在脉冲的函数、读写DS18B20的函数以及读取温度的函数。另外，DS18B20的直接输出为数字量（二进制数据），且其数据精度可调（可选9位、10位、11位或12位），默认读取精度为12位。下表显示了测量温度与输出数据之间的关系：

表4-1 测量温度与输出数据之间的关系

[illegible]

上表假设输出精度为12位，共16位(二进制补码)，高5位为符号位，中间7位为整数位，低5位为小数位。其中如果将DS18B20的输出精度配置为更低的位数，有些位将会置0。



图4-3读取温度函数程序框图

温度传感器模块部分程序如下所示:

```
float DS18B20_Get_Temp(void)
{
    uint8_t tmpmax, tmp1sb;
    short s_tem;
    float f_tem;

    DS18B20_Rst();
    DS18B20_Presence();
    DS18B20_Write_Byte(0xCC); // 跳过 ROM +
    DS18B20_Write_Byte(0x44); // 开始转换 +

    DS18B20_Rst();
    DS18B20_Presence();
    DS18B20_Write_Byte(0xCC); // 跳过 ROM +
    DS18B20_Write_Byte(0xEE); // 读温度值 +

    tmp1sb = DS18B20_Read_Byte();
    tmpmax = DS18B20_Read_Byte();

    s_tem = tmpmax<<8;
    s_tem = s_tem | tmp1sb;

    if( s_tem < 0 ) // 负温度 -
        f_tem = (~s_tem+1) * 0.625;
    else
        f_tem = s_tem * 0.625;

    return f_tem;
}
```

4.4 pH值、浑浊度、电导率传感器模块设计

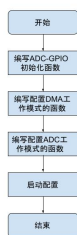


图4-4 ADC外设和DMA外设驱动程序的程序框图

ADC外设的部分驱动程序如下所示:

```

1  def __init__(self, encoder, decoder, num_embeddings):
2      # 编码器
3      self.encoder = encoder
4      self.encoder.reset_parameters()
5      # 解码器
6      self.decoder = decoder
7      self.decoder.reset_parameters()
8      # 嵌入矩阵
9      self.embedding = nn.Embedding(num_embeddings, embedding_dim)
10     self.embedding.reset_parameters()
11     # 初始化权重
12     self.decoder.init_weights()
13     self.encoder.init_weights()
14     # 初始化偏置
15     self.decoder.init_bias()
16     self.encoder.init_bias()
17     # 初始化偏置
18     self.decoder.init_bias()
19     self.encoder.init_bias()
20     # 初始化偏置
21     self.decoder.init_bias()
22     self.encoder.init_bias()
23     # 初始化偏置
24     self.decoder.init_bias()
25     self.encoder.init_bias()
26     # 初始化偏置
27     self.decoder.init_bias()
28     self.encoder.init_bias()
29     # 初始化偏置
30     self.decoder.init_bias()
31     self.encoder.init_bias()
32     # 初始化偏置
33     self.decoder.init_bias()
34     self.encoder.init_bias()
35     # 初始化偏置
36     self.decoder.init_bias()
37     self.encoder.init_bias()
38     # 初始化偏置
39     self.decoder.init_bias()
40     self.encoder.init_bias()
41     # 初始化偏置
42     self.decoder.init_bias()
43     self.encoder.init_bias()
44     # 初始化偏置
45     self.decoder.init_bias()
46     self.encoder.init_bias()
47     # 初始化偏置
48     self.decoder.init_bias()
49     self.encoder.init_bias()
50     # 初始化偏置
51     self.decoder.init_bias()
52     self.encoder.init_bias()
53     # 初始化偏置
54     self.decoder.init_bias()
55     self.encoder.init_bias()
56     # 初始化偏置
57     self.decoder.init_bias()
58     self.encoder.init_bias()
59     # 初始化偏置
60     self.decoder.init_bias()
61     self.encoder.init_bias()
62     # 初始化偏置
63     self.decoder.init_bias()
64     self.encoder.init_bias()
65     # 初始化偏置
66     self.decoder.init_bias()
67     self.encoder.init_bias()
68     # 初始化偏置
69     self.decoder.init_bias()
70     self.encoder.init_bias()
71     # 初始化偏置
72     self.decoder.init_bias()
73     self.encoder.init_bias()
74     # 初始化偏置
75     self.decoder.init_bias()
76     self.encoder.init_bias()
77     # 初始化偏置
78     self.decoder.init_bias()
79     self.encoder.init_bias()
80     # 初始化偏置
81     self.decoder.init_bias()
82     self.encoder.init_bias()
83     # 初始化偏置
84     self.decoder.init_bias()
85     self.encoder.init_bias()
86     # 初始化偏置
87     self.decoder.init_bias()
88     self.encoder.init_bias()
89     # 初始化偏置
90     self.decoder.init_bias()
91     self.encoder.init_bias()
92     # 初始化偏置
93     self.decoder.init_bias()
94     self.encoder.init_bias()
95     # 初始化偏置
96     self.decoder.init_bias()
97     self.encoder.init_bias()
98     # 初始化偏置
99     self.decoder.init_bias()
100    self.encoder.init_bias()

```

ADC外设的部分驱动程序如下所示:



4.5 液晶模块设计

本文所使用的显示屏为野火4.3寸液晶屏，内置液晶控制器NT35510，NT35510中内置显存，当我们想要显示数据时，其实就是将数据写入显存中，STM32是使用FSMC外设来模拟8080时序来控制液晶屏的。

STM32F407ZGT6内含FSMC (Flexible Static Memory Controller)，它可以用来管理对用于单片机存储容量进行扩展的存储器，包括SRAM、NOR Flash以及NAND Flash等。由于MCU控制液晶屏显示实际上是往液晶控制器的显存(通常为SRAM)中写入像素数据，故FSMC也可以用来控制液晶屏。

图4-5为FSMC外设框图。由于FSMC可以控制不同种类的存储器，相应地控制引脚也比较多，在图4-5中，右侧为FSMC外设的控制引脚，其中，NOR/PSRAM信号左边的花括号括起来的引脚FSMC[4:1]及FSMC_NL等引脚为NOR Flash以及PSRAM的控制器专用引脚，共享信号左边的花括号括起来的引脚为所有类型存储器的共用引脚。

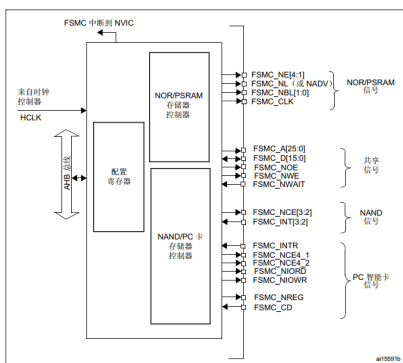


图4-5 FSMC外设框图

采用FSMC外设控制访问存储器与采用和SPI访问存储器的方式不同，后二者是给出一个要访问的存储器地址，再往地址中读写数据，地址和数据都要存储在存储器中的不同地址单元，读写数据时还要使用代码发送命令。而使用FSMC访问存储器时，通过一种地址映射的方式将STM32内核地址空间中的0x60000000~0x6FFFFFFF这段地址范围映射到外部扩展的存储器中，如图4-6所示，FSMC外设将内核地址空间映射到外部存储器中，若我们在程序中设一地址指针指向内核地址空间0x60000000~0x6FFFFFFF，则我们通过地址指针对该地址范围读写数据时，FSMC会自动进行转换，并实现对外部存储器对应存储区域的读写。

I²C

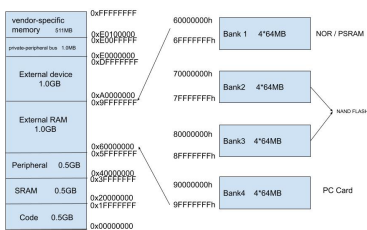


图4-6 FSMC外设对内核地址的映射

8080接口是由Intel公司推出的一种LCD接口通讯方式，MCU可以通过8080接口快速访问液晶显示屏。MCU通过8080接口向液晶屏（实际上为液晶控制器）的显存中发送命令与数据。同I2C和SPI总线接口类似，8080总线接口也有它对应的读写时序。图4-7为8080总线写时序图(本文只用到了写时序，当MCU要向LCD写入命令(带参数)时，分为两个阶段传输。第一阶段传输命令地址，先将片选信号CS拉低，将数据/命令选择信号拉低，表示写入LCD的是命令地址，写信号WR为低电平，读信号RD为高电平，表示读写方向为MCULCD, 同时在数据线DB[23:0]上传

输命令地址。第二阶段传输命令参数，将DC信号置为高电平，表示传输的是参数(数据)，其余信号与第一阶段相同。当MCU要向LCD写入像素数据时，传输过程与上述写入命令的第二阶段相同。

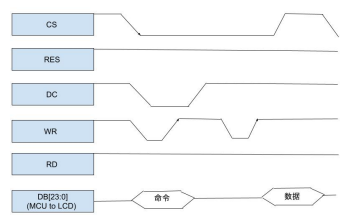


图4-7为8080总线写时序图

对于液晶控制器模块的程序设计主要是编写NT35510的驱动程序和FSMC外设的驱动程序。液晶控制器模块的程序框图如图4-8所示：

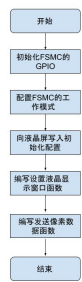


图4-8液晶控制器模块的程序框图

液晶控制器模块的部分驱动程序如下所示：

```
1 // 液晶显示模块驱动函数
2 // 液晶显示模块驱动函数
3 // 液晶显示模块驱动函数
4 // 液晶显示模块驱动函数
5 // 液晶显示模块驱动函数
6 // 液晶显示模块驱动函数
7 // 液晶显示模块驱动函数
8 // 液晶显示模块驱动函数
9 // 液晶显示模块驱动函数
10 // 液晶显示模块驱动函数
11 // 液晶显示模块驱动函数
12 // 液晶显示模块驱动函数
13 // 液晶显示模块驱动函数
14 // 液晶显示模块驱动函数
15 // 液晶显示模块驱动函数
16 // 液晶显示模块驱动函数
17 // 液晶显示模块驱动函数
18 // 液晶显示模块驱动函数
19 // 液晶显示模块驱动函数
20 // 液晶显示模块驱动函数
21 // 液晶显示模块驱动函数
22 // 液晶显示模块驱动函数
23 // 液晶显示模块驱动函数
24 // 液晶显示模块驱动函数
25 // 液晶显示模块驱动函数
26 // 液晶显示模块驱动函数
27 // 液晶显示模块驱动函数
28 // 液晶显示模块驱动函数
29 // 液晶显示模块驱动函数
30 // 液晶显示模块驱动函数
31 // 液晶显示模块驱动函数
32 // 液晶显示模块驱动函数
33 // 液晶显示模块驱动函数
34 // 液晶显示模块驱动函数
35 // 液晶显示模块驱动函数
36 // 液晶显示模块驱动函数
37 // 液晶显示模块驱动函数
38 // 液晶显示模块驱动函数
39 // 液晶显示模块驱动函数
40 // 液晶显示模块驱动函数
41 // 液晶显示模块驱动函数
42 // 液晶显示模块驱动函数
43 // 液晶显示模块驱动函数
44 // 液晶显示模块驱动函数
45 // 液晶显示模块驱动函数
46 // 液晶显示模块驱动函数
47 // 液晶显示模块驱动函数
48 // 液晶显示模块驱动函数
49 // 液晶显示模块驱动函数
50 // 液晶显示模块驱动函数
51 // 液晶显示模块驱动函数
52 // 液晶显示模块驱动函数
53 // 液晶显示模块驱动函数
54 // 液晶显示模块驱动函数
55 // 液晶显示模块驱动函数
56 // 液晶显示模块驱动函数
57 // 液晶显示模块驱动函数
58 // 液晶显示模块驱动函数
59 // 液晶显示模块驱动函数
60 // 液晶显示模块驱动函数
61 // 液晶显示模块驱动函数
62 // 液晶显示模块驱动函数
63 // 液晶显示模块驱动函数
64 // 液晶显示模块驱动函数
65 // 液晶显示模块驱动函数
66 // 液晶显示模块驱动函数
67 // 液晶显示模块驱动函数
68 // 液晶显示模块驱动函数
69 // 液晶显示模块驱动函数
70 // 液晶显示模块驱动函数
71 // 液晶显示模块驱动函数
72 // 液晶显示模块驱动函数
73 // 液晶显示模块驱动函数
74 // 液晶显示模块驱动函数
75 // 液晶显示模块驱动函数
76 // 液晶显示模块驱动函数
77 // 液晶显示模块驱动函数
78 // 液晶显示模块驱动函数
79 // 液晶显示模块驱动函数
80 // 液晶显示模块驱动函数
81 // 液晶显示模块驱动函数
82 // 液晶显示模块驱动函数
83 // 液晶显示模块驱动函数
84 // 液晶显示模块驱动函数
85 // 液晶显示模块驱动函数
86 // 液晶显示模块驱动函数
87 // 液晶显示模块驱动函数
88 // 液晶显示模块驱动函数
89 // 液晶显示模块驱动函数
90 // 液晶显示模块驱动函数
91 // 液晶显示模块驱动函数
92 // 液晶显示模块驱动函数
93 // 液晶显示模块驱动函数
94 // 液晶显示模块驱动函数
95 // 液晶显示模块驱动函数
96 // 液晶显示模块驱动函数
97 // 液晶显示模块驱动函数
98 // 液晶显示模块驱动函数
99 // 液晶显示模块驱动函数
100 // 液晶显示模块驱动函数
```

第五章 系统制作与调试

5.1 系统硬件制作

将液晶显示屏插入开发板的排母上，并将四个传感器模块分别连接到开发板上对应的供电电源及I/O口，连接状况分别如下：温度传感器模块：VCC3.3V；GND CGND；DOCPB3；PH传感器模块：VCC5V；GNDGND；POCPB0；EC传感器模块：VCC5V；GNDGND；AOCBP1；浑浊度传感器模块：VCC5V；GNDGND；AOCPA6；硬件连接实物图如图5-1所示：

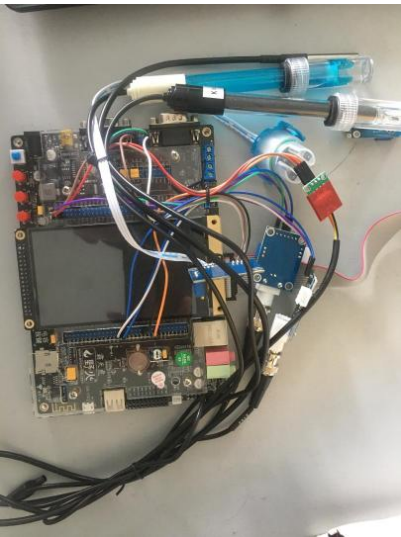


图5-1硬件连接实物图

5.2 软件调试

在程序调试时，共进行5个检测实验，实验~4用于调试各传感器模块源程序，实验5为综合检测实验。

实验1检测温度传感器模块，将程序下载到开发板中，再将温度传感器探头分别放置到热水与冷水中，打开电源并按下复位按键，结果分别如图5-2和5-3所示。检测66℃热水以及冷水水温时，液晶显示屏中温度示数分别为63.188和24.125，说明温度传感器模块检测效果较好。

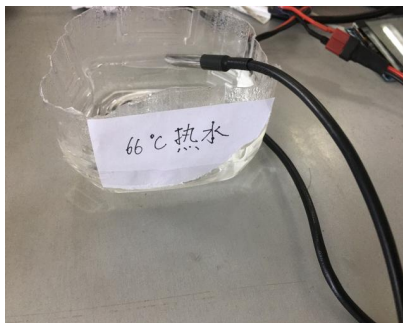


图5-2检测热水水温

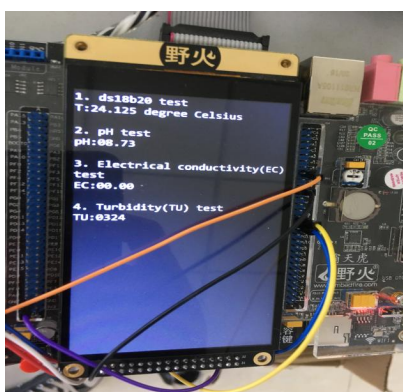


图5-3 检测冷水水温

实验2检测pH传感器模块，将pH传感器探头分别放置到pH值4.00和9.18的溶液池中，打开电源并按下复位按钮，结果分别如图5-4和5-5所示。检测pH值分别为4.00和9.18的溶液pH时，显示屏的pH值示数分别为5.08和9.05，说明pH值传感器模块检测效果较好。

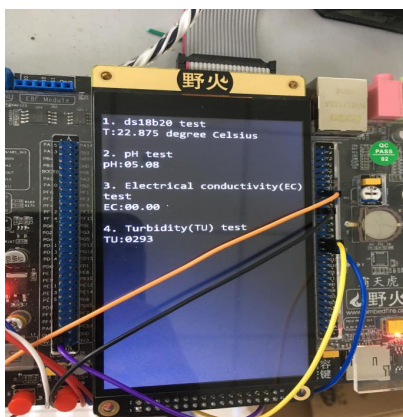


图5-4 检测pH为4.00的溶液PH值

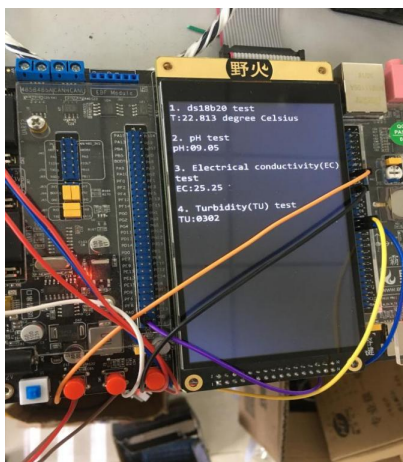
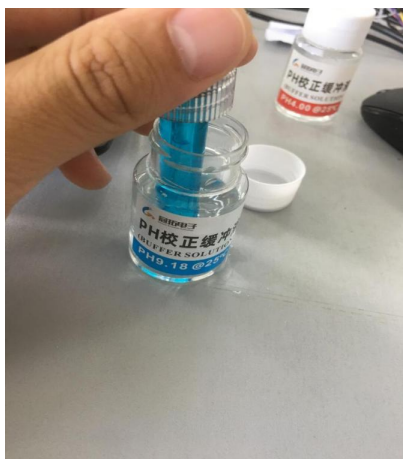


图5-5检测pH为9.18的溶液PH值

实验3检测浑浊度传感器模块，将浑浊度传感器探头分别放置到墨水以及清水中，打开电源并按下复位按键，结果分别如图5-6和5-7所示。检测墨水和清水浑浊度时，显示屏的浑浊度示数分别为2506和191，墨水浑浊度示数远高于清水浑浊度，检测效果较好。

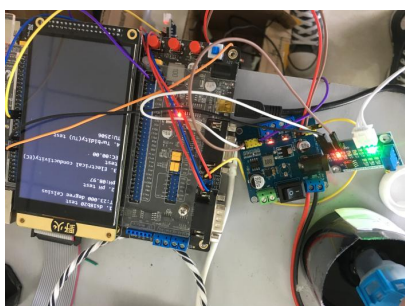


图5-6 检测墨水浑浊度

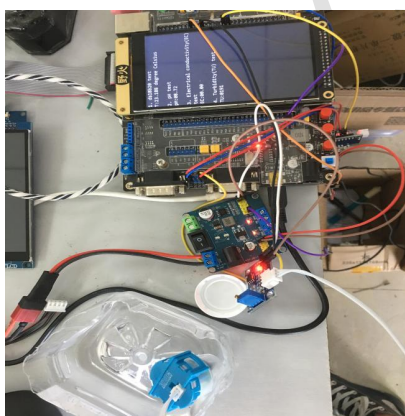


图5-7检测清水浑浊度

实验4检测电导率传感器模块，将电导率传感器探头分别放置到电导率为1413/cm和12.88ms/cm的溶液中，打开电源并按下复位按键，结果分别如图5-8和5-9所示。，检测电导率为1413/cm和12.88ms/cm的溶液电导率时，显示屏的电导率示数分别为1.01和11.11，检测效果较好。

μS



图5-8检测电导率为1413/cm溶液的电导率

μs

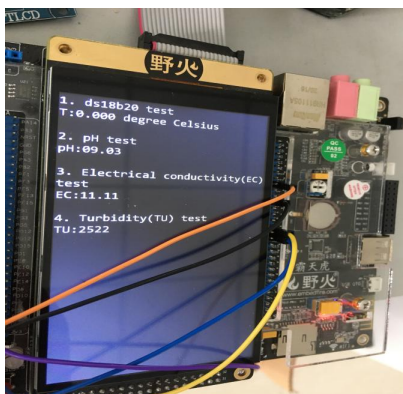


图5-9 检测电导率为12.88ms/cm溶液的电导率

实验5为对掺杂了多种溶液的混合溶液的各个参数同时检测。结果如图5-10所示：

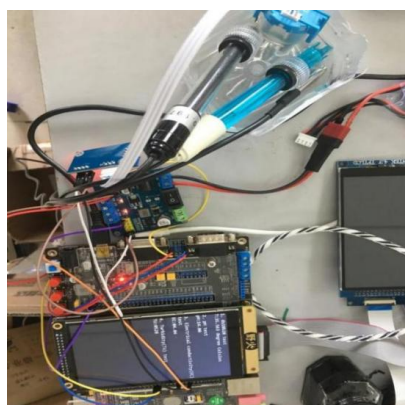


图5-10 综合检测

第六章 总结与展望

本次毕业设计历时一个学期，经过查阅资料、方案设计、购买元器件、硬件系统设计、原理图绘制、查阅器件手册、编写源程序等多个步骤，**设计出了一种多参数水质监测系统**。该系统采用STM32F407ZGT6作为主控芯片，结合温度传感器、pH值传感器、电导率传感器、浑浊度传感器及STM32中内嵌的外设（ADC、DMA、FSMC）等，最终实现了以下功能：

能够实时检测水样的温度、浑浊度、PH值和电导率。

液晶显示屏能够实时显示检测数据，方便用户实时观察水质。

经测试，本系统性能稳定、检测精度高，此外系统还具有使用便捷、价格低廉等特点。

当然，还可对本检测系统添加蜂鸣器或LED闪烁等报警功能模块，可对四项参数均事先设置安全阈值，一旦检测结果超出安全阈值，报警模块即发出蜂鸣声或红灯闪烁以提醒用户当前水质可能已不能供正常使用。另外，本系统对检测数据也未进行存储，每进行一次新的检测就会将上一次的检测数据覆盖掉，系统关机重启之后所有的检测数据也会丢失。若可为本系统建立一个服务器端数据库，将本系统作为客户端，并添加网络通信功能，一旦客户端检测到数据就立即上传到服务器端，而服务器端的海量数据可以应用于污水处理、水质监测等领域。

参考文献

- [1] 申伟. 杜文娟. 基于ZigBee无线传感技术的水库水质多参数监测系统[J]. 自动化技术与应用, 2018, 37(4): 40-43.
- [2] 周皓东. 黄燕, 刘炜. 基于WiFi无线传感器网络的水质监测系统设计[J]. 传感器与微系统, 2015, 34(5): 99-101+105.
- [3] 赵敏华. 李莉, 呼娜. 基于无线传感器网络的水质监测系统设计[J]. 计算机工程, 2014, 40(2): 92-96.
- [4] 梁正宁. 黄晔. 基于Lora的水质检测系统设计[J]. 无线通信技术, 2018, 4: 46-50+55.

- [5] 杨磊, 熊卫华, 姜明. 基于NB_IoT技术的家庭水质检测系统[J]. 计算机系统应用, 2019, 28(12): 129-133
- [6] 王英, 陈聪伟, 肖金球. 水质多参数监测系统设计[J]. 苏州科技学院学报(自然科学版), 2015, 32(1): 54-58+63
- [7] 谷春英, 姚青山. 基于无线传感器的水质监测系统仿真设计[J]. 计算机仿真, 2013, 30(1): 340-343.
- [8] 孙雷霸. 基于无线传感器网络的水环境多参数监测系统研究与实现[D]. 江苏大学, 2009.
- [9] 周新. 从零开始学Altium Designer电路设计与PCB制版[M]. 北京: 化学工业出版社, 2020.
- [10] 马忠梅, 王美刚, 王拓. 单片机的C语言应用程序设计(第6版)[M]. 北京: 北京航空航天大学出版社, 2017.
- [11] Ahmed Abbas Fadel, Mohamed Ibrahim Shujaa. Water Quality Monitoring System Based on IOT Platform[J]. IOP Conference Series: Materials Science and Engineering, 2020, 928(3): 032054.
- [12] Himanshu Jindal, Sharad Saxena, Singara Singh Kasana. Sewage Water Quality Monitoring Framework Using Multi-parametric Sensors[J]. Wireless Personal Communications, 2017, 97(1): 881-913.

指标说明:

- 1、总相似比: 类似于重合率, 即送检论文内容与所选检测资源范围内所有文献相似的部分(包括参考引用部分), 占整个送检论文内容的比重, 总相似比=复写率+引用率;
- 2、复写率: 即送检论文内容与所选检测资源范围内所有文献相似的部分(不包括参考引用部分), 占整个送检论文内容的比重;
- 3、引用率: 即送检论文内容中被系统识别为引用的部分, 占整个送检论文内容的比重(引用部分一般指正确标示引用的部分);
- 4、自写率: 即送检论文内容中剔除相似片段和引用片段后, 占整个送检论文内容的比重, 一般可用于论文的原创新性和新颖性评价, 自写率=1-复写率-引用率;
- 5、同届相似比: 即送检论文内容与校方所选同届库检测资源范围内所有文章相似的部分(不包括参考引用部分), 占整个送检论文内容的比重;
- 6、报告中, 红色与橙色文字表示复写片段, 蓝色与深蓝色文字表示引用片段, 紫色文字表示同届相似片段, 黑色文字表示自写片段。

免责声明:

- 1、本报告为G·格子达系统检测后自动生成, 鉴于论文检测技术及论文检测样本库的局限性, G·格子达不保证检测报告的绝对准确, 仅对您所选择的检测资源范围内的检验结果负责, 相关结论仅供参考, 不做法律依据;
- 2、G·格子达论文检测服务中使用的论文样本, 除特别声明者外, 其著作权归各自权利人享有。根据中华人民共和国著作权法相关规定, G·格子达网站为学习研究、介绍、评论、教学、科研等目的引用其论文片段属于合理使用。除非经原作者许可, 请勿超出合理使用范围使用其内容和本网提供的检测报告。

联系我们:



防伪二维码



关注微信公众号

官方网站: www.gocheck.cn

客服热线: 400-699-3389

客服QQ: 800113999