****

课程设计报告书

**题目：智能鱼缸**

**学 院 电子与信息学院**

**专业班级 信息工程1班**

**学生姓名 郑雨萌 徐雪莹 宋蚺 崔子涵 张清智**

**指导教师 邢晓芬**

**课程编号 035100691**

**课程学分 1.0**

**起始日期 2022年2月23日**

[一、性能级描述 3](#_Toc105286264)

[（一）、选题背景 3](#_Toc105286265)

[（二）、性能级设计 4](#_Toc105286266)

[二、需求分析 4](#_Toc105286267)

[（一）STM32F103ZET6 4](#_Toc105286268)

[（二）外围元件…… 5](#_Toc105286269)

[（三）小程序端实现 5](#_Toc105286270)

[三、方案论证 6](#_Toc105286271)

[（一）FPFA和外围元件的关系 6](#_Toc105286272)

1. [显示模块 6](#_Toc105286273)

[2. 温度控制模块 7](#_Toc105286274)

[3. 自动喂食模块 9](#_Toc105286275)

[4. 硬件原型设计 12](#_Toc105286276)

[…… 13](#_Toc105286277)

[（二）顶层模块划分 13](#_Toc105286278)

[1.显示模块 13](#_Toc105286279)

[2.温度控制模块 13](#_Toc105286280)

[3.自动喂食模块 13](#_Toc105286281)

[4.服务器配置与通信模块 14](#_Toc105286282)

[5.下行数据发送模块 14](#_Toc105286283)

[6.上行数据接收模块 14](#_Toc105286284)

[7.用户登陆模块 14](#_Toc105286285)

[…… 14](#_Toc105286286)

[四、分工 15](#_Toc105286287)

[（一）、时间规划 15](#_Toc105286288)

[（二）、小组分工… 16](#_Toc105286289)

[1.郑雨萌工作 16](#_Toc105286290)

[2.徐雪莹工作 17](#_Toc105286291)

[3.宋蚺工作 17](#_Toc105286292)

[…… 17](#_Toc105286293)

[五、模块设计 17](#_Toc105286294)

[（一）显示模块设计 17](#_Toc105286295)

[1. 功能实现 17](#_Toc105286296)

[2. 端口设置 18](#_Toc105286297)

[3 实现函数及关键性代码 18](#_Toc105286298)

[（二）温度控制模块设计 19](#_Toc105286299)

[1. 功能实现 19](#_Toc105286300)

[2. 端口设置 19](#_Toc105286301)

[3 实现函数及关键性代码 20](#_Toc105286302)

[（三）自动喂食模块设计 20](#_Toc105286303)

[1. 功能实现 20](#_Toc105286304)

[2. 端口设置 20](#_Toc105286305)

[3 实现函数及关键性代码 21](#_Toc105286306)

[（四）服务器配置与通信模块 21](#_Toc105286307)

[1.配置介绍 21](#_Toc105286308)

[2.配置详情 22](#_Toc105286309)

[（五）下行数据发送模块 22](#_Toc105286310)

[1.功能介绍 22](#_Toc105286311)

[2.上下行传输命令 22](#_Toc105286312)

[2.1灯光控制命令 22](#_Toc105286313)

[2.2换水控制命令 23](#_Toc105286314)

[2.3温度控制命令 23](#_Toc105286315)

[2.4喂食时间控制命令 23](#_Toc105286316)

[3.原型界面 23](#_Toc105286317)

[（六）上行数据接收模块 24](#_Toc105286318)

[1.功能介绍 24](#_Toc105286319)

[2.上下行传输命令 24](#_Toc105286320)

[2.1连接服务器与订阅上行数据topic 24](#_Toc105286321)

[2.2类型转换命令 24](#_Toc105286322)

[2.3数据绑定命令 25](#_Toc105286323)

[2.4数据显示命令 25](#_Toc105286324)

[3.效果展示 25](#_Toc105286325)

[（七）用户登陆模块 25](#_Toc105286326)

[1.功能介绍 25](#_Toc105286327)

[2.具体展示与说明 25](#_Toc105286328)

[…… 28](#_Toc105286329)

[六、系统联调与测试 28](#_Toc105286330)

[七、总结与心得体会 28](#_Toc105286331)

[1.郑雨萌 28](#_Toc105286332)

[2.宋蚺、徐雪莹 29](#_Toc105286333)

[…… 29](#_Toc105286334)

## 一、性能级描述

### （一）、选题背景

随着我国人民生活水平的提高，吃鱼找健康，观赏鱼找和谐，已经成为当今人们生活中的一种时尚。养鱼不仅可以静心养眼，鱼缸蒸发的水汽还能调节室内空气的干湿度，有着缓解视觉疲劳、美化环境、调节室内空气的氧含量和湿度等优点。

然而，观赏鱼的饲养是一项技术性较强的工作，由于饲养者经验不足或由于工作忙、 出差、旅游等原因，没有时间无精力照顾鱼儿，导致出现病鱼、死鱼的现象。不仅对鱼儿造成了伤害，还会严重影响养鱼人的心情。传统鱼缸无法实时监控环境，远程控制，辅助用户饲养，使得养鱼者需要付出极大的人工成本。本项目将基于STM32设计一款智能鱼缸系统，该系统有望弥补传统鱼缸存在的养鱼成本高、难度大等缺陷。

### （二）、性能级设计

为了能够智能鱼缸真正方便养鱼群体的需求，我们小组发布问卷并收集整理了相关信息。经调查发现，绝大多数用户使用传统鱼缸养鱼的过程中遇到过换水不便、水质浑浊、无暇照顾、易出现病鱼死鱼的困扰；而对于智能鱼缸的期待，超半数用户表示喜欢智能水循环、自动喂食和智能恒温功能，不少用户表示喜欢远程控制灯光和冲浪泵功能。

与此同时，我们也搜索了国内外市场上现有高端鱼缸实现的功能，从中吸收了一些有用的想法。目前市场有一些高端鱼缸具有水循环过滤系统，可以较长时间不用换水；一些鱼缸加入了加热和增氧功能等等。这些设计在一定程度上简化了养鱼过程，但仍然存在一些弊端，比如用户无法远程操作，很难做到出门在外多日而不必担心家中观赏鱼无人照料。

综上分析，智能生态鱼缸有望实现的所有功能如下：

1. 对水质、水位、温度实时监测，用户可通过鱼缸显示屏和微信小程序两种途经收知数据信息；
2. 用户可根据鱼群具体需求为其设置适宜的生存环境，并且智能鱼缸能够据此做出实时调整，如水质浑浊水位低开启水泵换水、温度过低开启加热片加热等；
3. 增加喂食功能，用户可在微信小程序端远程操控，方便用户日常照料；
4. 增加灯光系统，用户可通过微信小程序远程控制，使鱼缸兼具夜晚照明和观赏功能；
5. 添加微型摄像头，便于用户检查观赏鱼健康状况；

## 二、需求分析

### （一）STM32F103ZET6



本次课程设计选用的硬件平台是野火STM32-F103ZE-霸道\_V2开发板，采用的是 Cortex-M3 内核，由 ARM 公司设计；选用的软件平台是Keil5和微信开发者工具。

### （二）外围元件……

根据智能鱼缸的功能分析，本次课设分为显示、通信、温度控制、水位控制、水质控制、自动喂食、灯光、健康监测共八个模块。以下依次为各个模块使用到的外围元件：

1. 显示模块：I2C接口的OLED用于显示鱼缸的基本环境信息；
2. 通信模块：
3. 温度控制模块：温度传感器DS18B20用于检测鱼缸水温、蜂鸣器用于报警、24V100W PI膜加热片用于加热水温、1路5V继电器用于控制回路通断
4. 水位控制模块：
5. 水质控制模块：
6. 自动喂食模块：24V5A变压器用于给步进电机驱动器供电、EBF-MSD4805步进电机驱动器用于驱动步进电机、57步进电机用于喂食实现
7. 灯光模块：
8. 健康监测模块：

### （三）小程序端实现

Vue是一套构建用户界面的渐进式框架。与其他重量级框架不同的是，Vue采用自底向上增量开发的设计。Vue 的核心库只关注视图层，并且非常容易学习，非常容易与其它库或已有项目整合。Vue 完全有能力驱动采用单文件组件和Vue生态系统支持的库开发的复杂单页应用。由于Vue易用、灵活、性能好，所以我们选择在Vue框架的基础上进行微信小程序的开发。在开发的过程中我们使用了Vant Weapp小程序 UI 组件库，Vant 是一个轻量、可靠的移动端组件库，给我们的样式开发带来了很大的方便。

考虑到微信网络配置服务器域名只支持https和wss协议（加密的网络通信），使用EMQX的在线MQTT 5服务器的8883（TCP/TLS端口）与8084（Websocket/TLS端口）无法使用，故选用自己创建的支持加密通信的阿里云MQTT服务器。

我们需要在小程序中使用mqtt来进行tcp的长连接，实时监听设备的返回状态，eclipse开发的paho-mqtt与个人开发的mqtt.js都可以实现该功能，而mqqtt.js比较灵活对小程序也有很好的支持，所以采用了这个库来做连接，且已有对应的npm包，使用方便快捷。

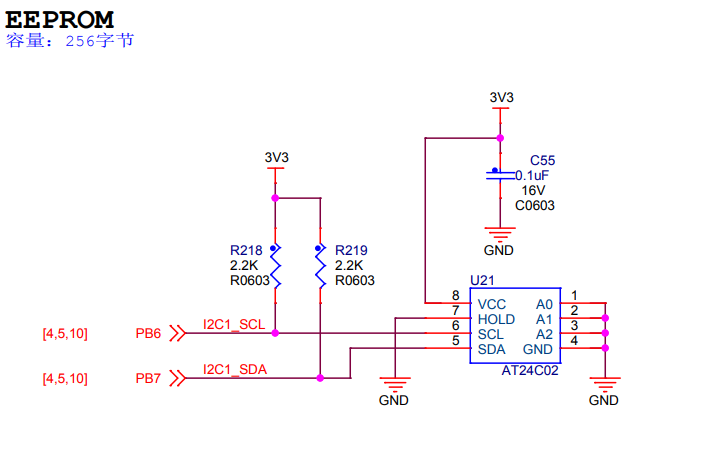
## 三、方案论证

### （一）FPFA和外围元件的关系

#### 1.显示模块

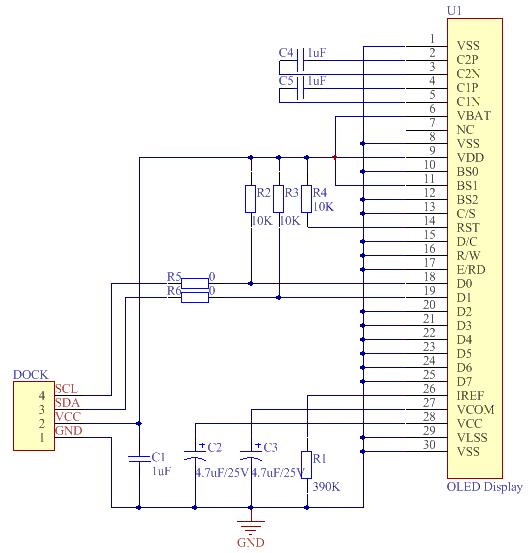
野火STM32-F103ZE-霸道\_V2开发板并无自带OLED，但是提供了能够通过I2C总线读写的EEPROM和通过SPI总线读写的串行FLASH，经过综合考虑，本次课程设计采用I2C通信方式实现OLED的显示。

EEPROM模块的原理图如下：



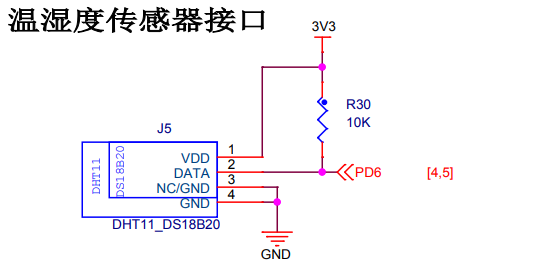
因此，使用OLED时，先要配置CPU的硬件I2C，即可向寄存器中读写字节显示于OLED屏幕上，OLED的实物图和原理图如下：





#### 2. 温度控制模块

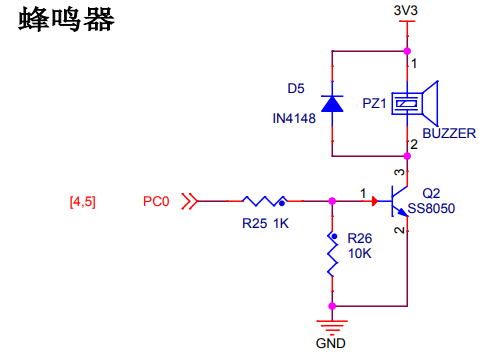
采用野火STM32-F103ZE-霸道\_V2开发板提供的温湿度传感器接口，原理图如下，由于只采集温度数据，因此使用到VDD、DATA、NC/GND三个引脚。



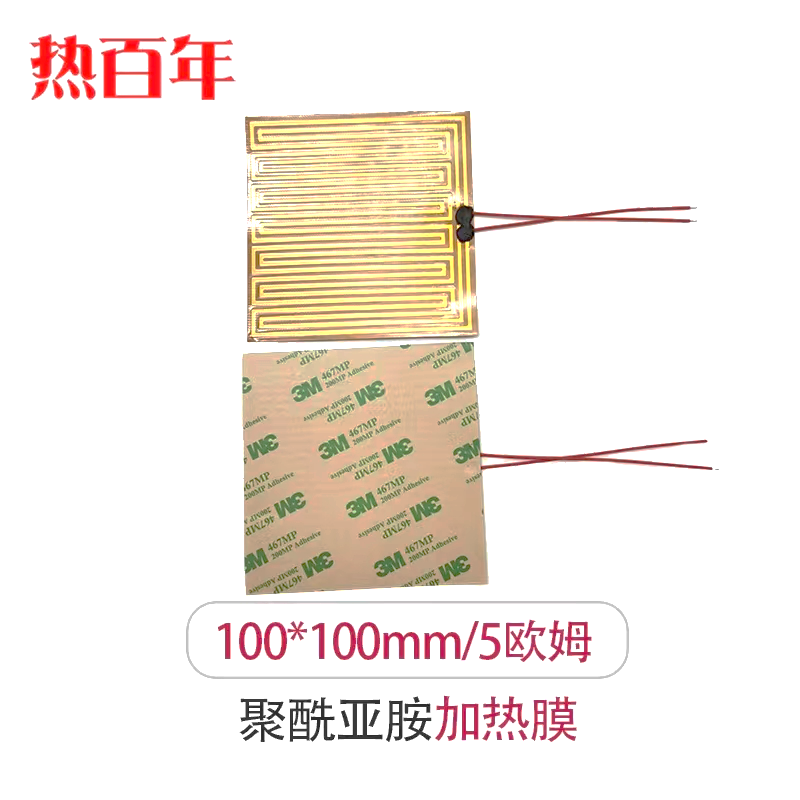
温度传感器则使用不锈钢封装的防水型DS18B20数字温度传感器，其实物图如下：



当温度超过用户设定最高温度时，负责报警使用的蜂鸣器为开发板自带模块，其原理图如下：



当温度低于用户设定最低温度时，加热水温使用的24V100W PI膜加热片实物图如下：

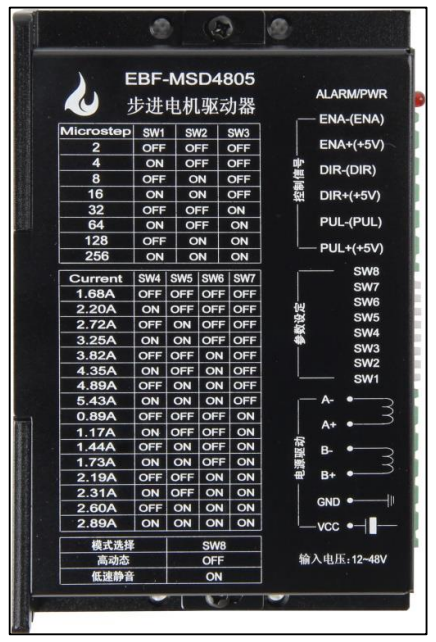


此外，使用继电器的COM端和NO端使得开发板输出的5V电压控制加热片供电电压24V的通断，其工作的最大功率为直流30V/10A，实物图如下：

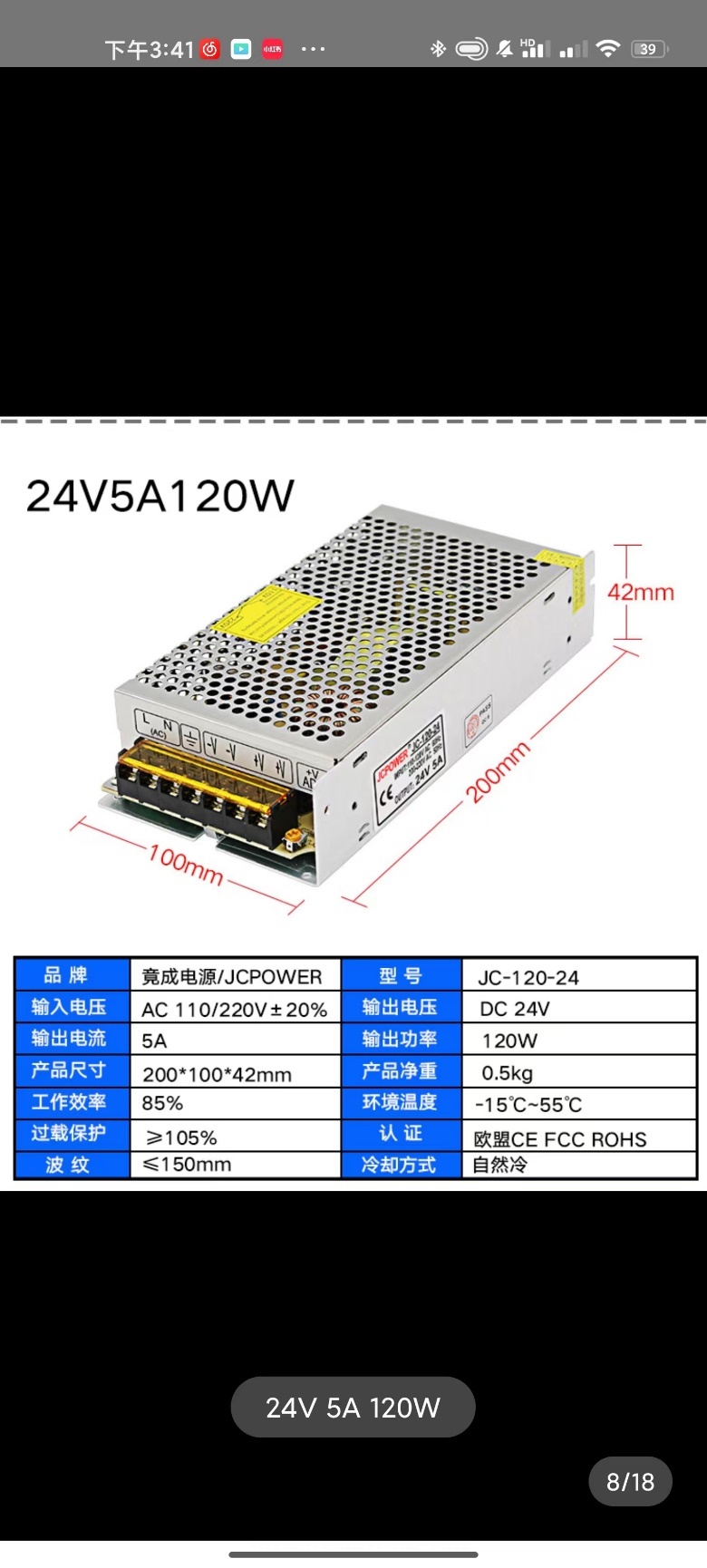


#### 3. 自动喂食模块

使用步进电机时首先应该考虑到其驱动方式，本次课程设计采用的是野火科技推出的EBF-MSD4805步进电机驱动器，它是一款以双极恒流PWM 驱动输出控制电机的驱动器，驱动电压范围 DC12V～48V，适合外径为 42mm、57mm、86mm 系列，驱动电流在 5.5A 以下的所有两相混合式步进电机，其实物图如下：



为驱动器提供输入电压的是24V5A变压器，其实物图如下：



本次课程设计采用的是外径为57mm的步进电机，其额定电流为3.5A，步距角度为1.8°。由于驱动器的输出电流需要根据步进电机的额定电流设定，一般略小于步进电机额定电流，因此设置输出电流为3.25A，即驱动器拨码开关SW4、SW5、SW6、SW7的状态分别为ON、ON、OFF、OFF。并且设置驱动器的细分为4细分，因此电机旋转一圈需要360°/1.8°=200个脉冲，4细分一圈需要的脉冲数为200\*4=800个，即驱动器的拨码开关SW1、SW2、SW3分别设置为ON、OFF、OFF。

表1和表2分别为细分微步参数使用拨码开关设置说明、电流参数使用拨码开关设置说明。

表1 细分微步参数使用拨码开关设置说明

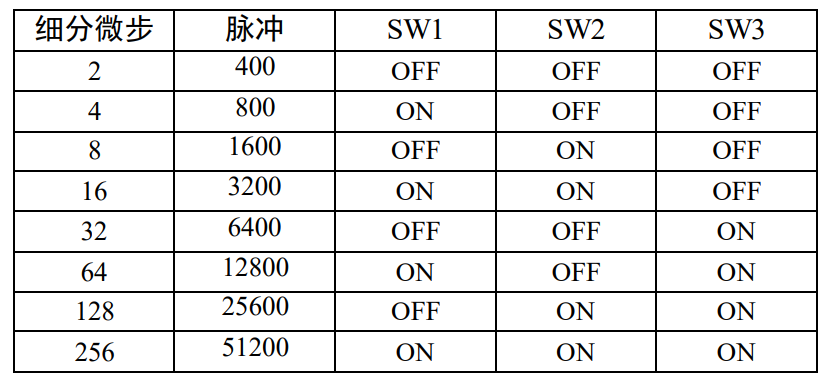
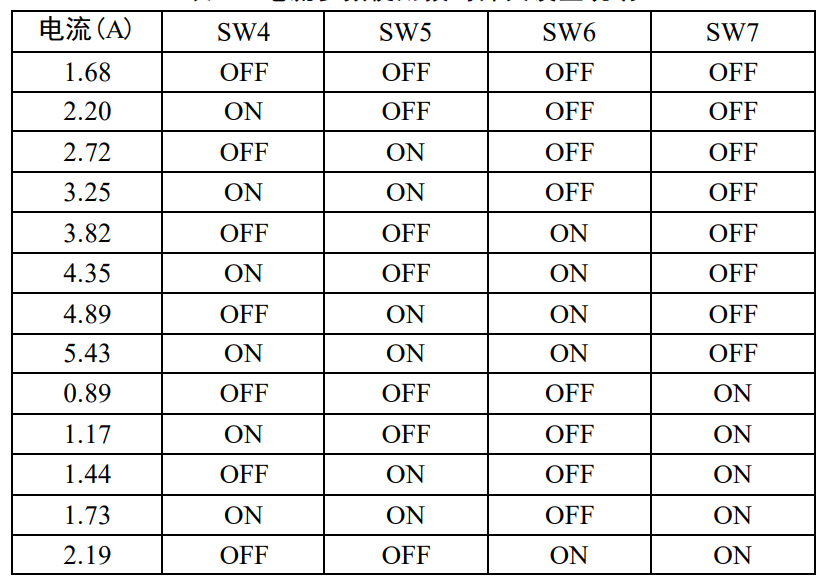
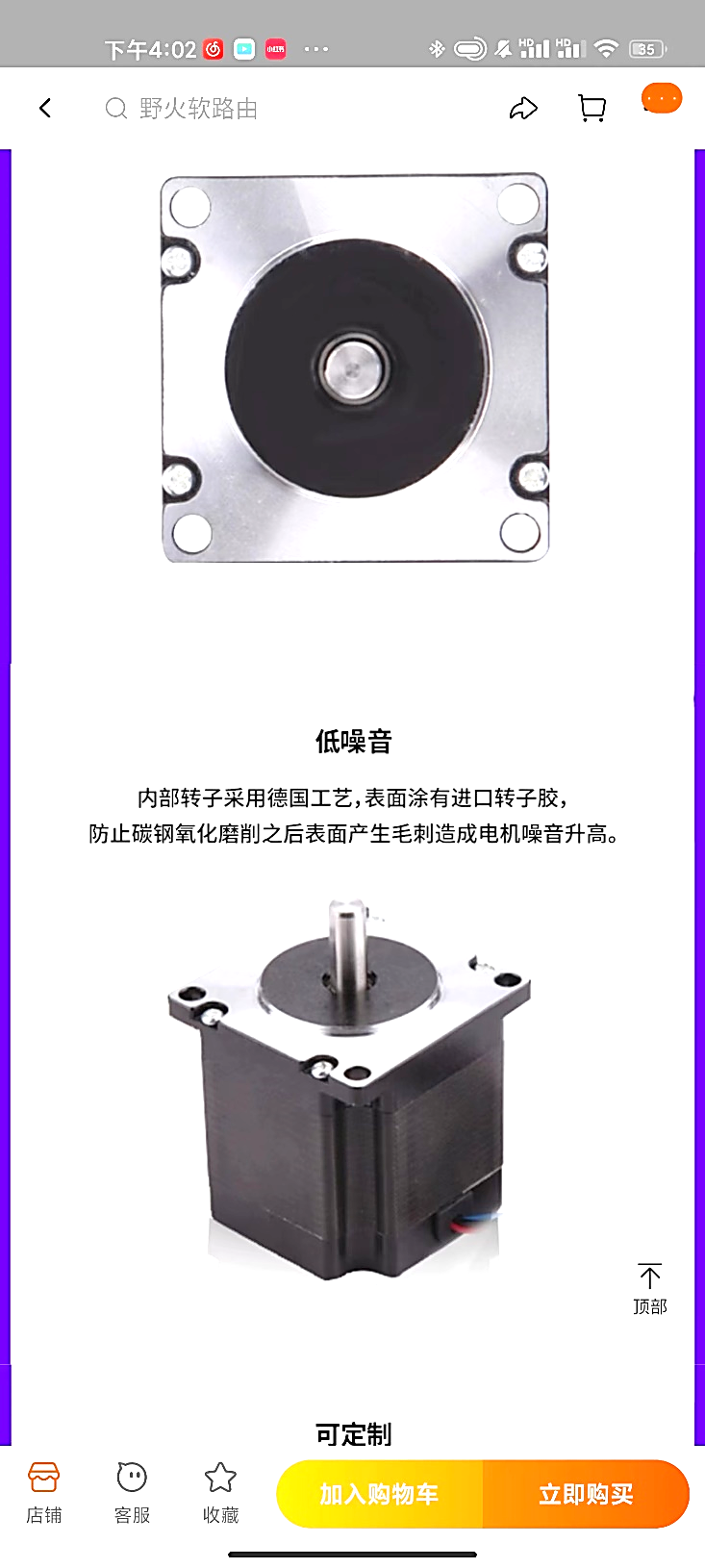


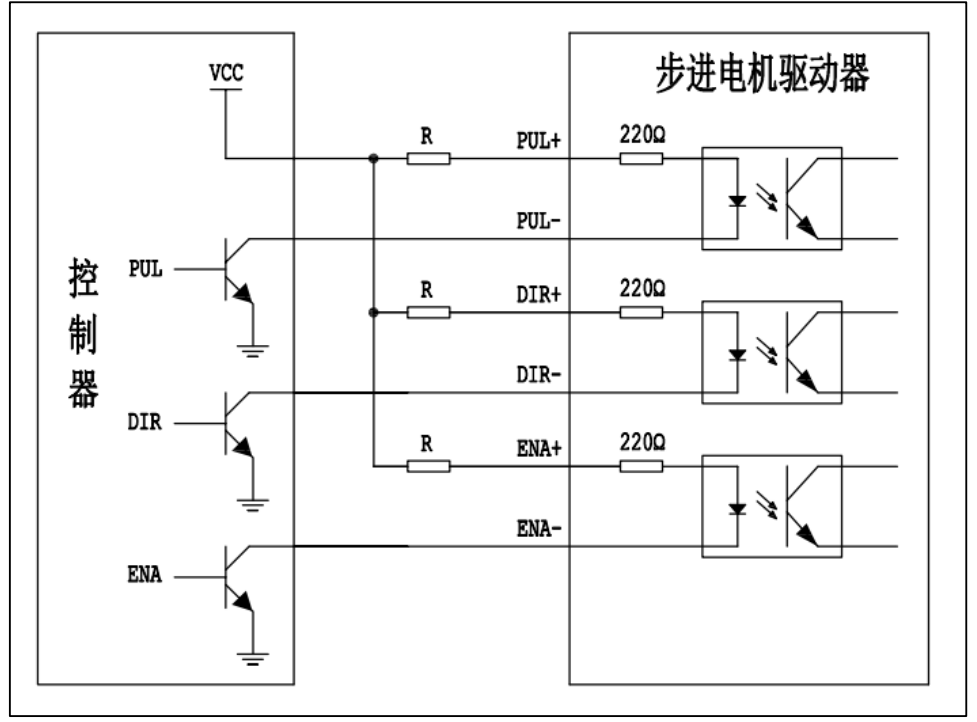
表2 电流参数使用拨码开关设置说明



步进电机实物图如下：



步进电机和驱动器采用共阳极接法，连线如下：



#### 4.硬件原型设计

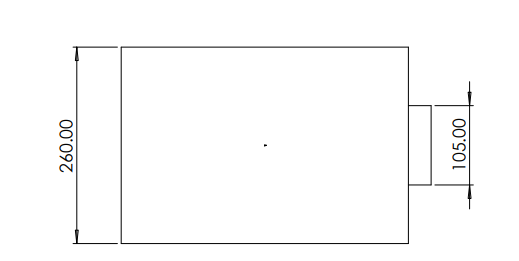
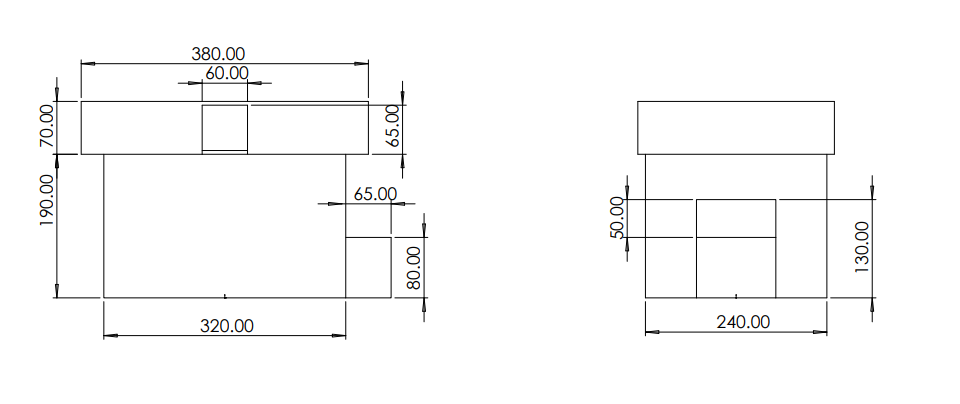


图11 原型三视图

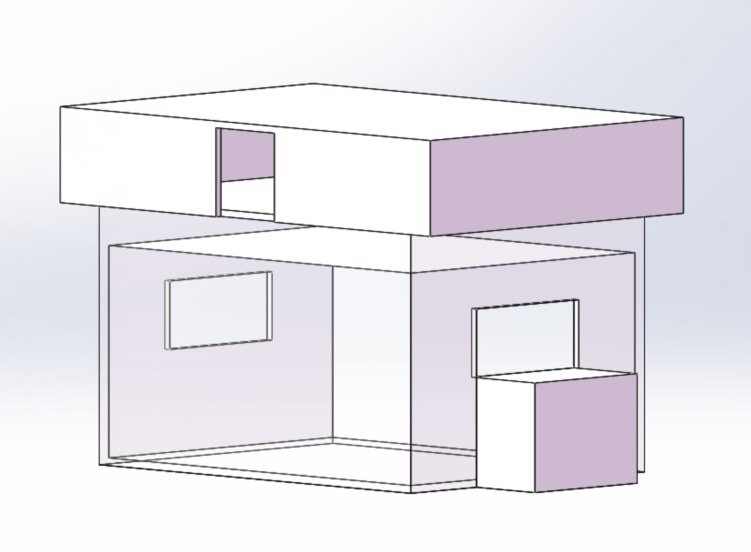


图12 原型模型

#### ……

### （二）顶层模块划分

根据实际情况和性能要求，可以采用模块化的系统设计方法，对整个系统做顶层模块的划分：

#### 1.显示模块

显示模块的功能是显示鱼缸内水温、水质的等基本信息，将温度传感器、水质传感器收集到的信息通过I2C总线读入寄存器中，显示于OLED屏幕，每2s进行一次更新，温度值以字符串的格式实时显示，若温度失常则通过OLED显示提醒用户此时温度过高还是过低；而水质以一阈值划定、显示“清澈、浑浊”字样。

此外，考虑到智能鱼缸的美观性，还增加了开机动画，当用户通电重置智能鱼缸时，OLED屏幕显示“欢迎使用智能鱼缸”字样。

#### 2.温度控制模块

温度控制模块的功能是通过温度传感器每间隔1s检测一次鱼缸内水温，并将水温与微信小程序中用户设定的阈值进行对比判断，如果温度位于合理的范围内——低于最高温度、高于最低温度，则正常显示于OLED屏幕；如果温度值过高，则驱动蜂鸣器报警，并在OLED屏幕上显示“过高”字样提醒用户；如果温度值过低，则控制加热片升高水温，并在OLED屏幕上显示“过低”字样提醒用户，当水温高于最低温度后自动切断加热。

#### 3.自动喂食模块

自动喂食模块的功能实现是用户通过微信小程序设定喂食量与喂食间隔，于是，在计数器到达既定的喂食间隔后，电机将自动启动，驱动喂食器向鱼缸中投喂用户设定克数的鱼食。

#### 4.服务器配置与通信模块

通信模块用于小程序端和服务器的连接、小程序端订阅设备上行数据Topic，是订阅上行数据和发送下行数据的基础。

#### 5.下行数据发送模块

通过小程序端切换开关或更改设置将命令发送至服务器，由上行设备读取命令，实现小程序端对设备端的远程控制。

#### 6.上行数据接收模块

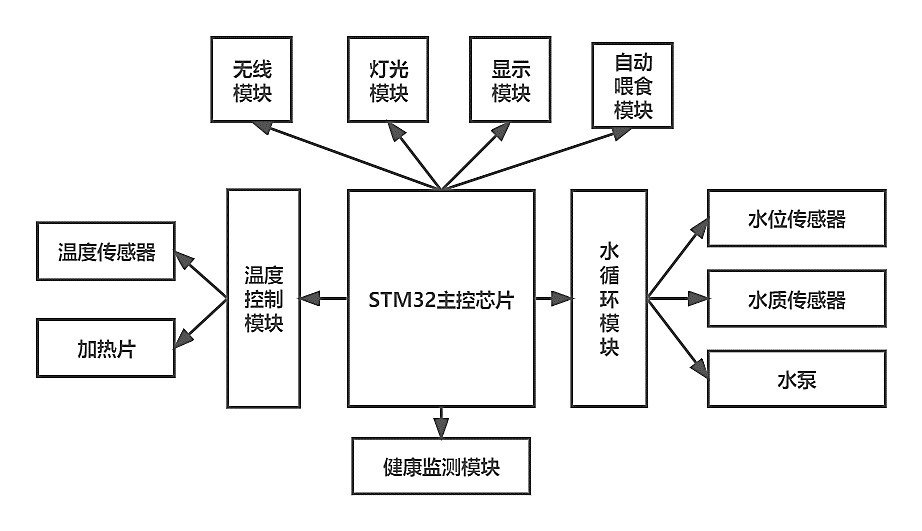
连接服务器，订阅设备上行数据topic，并对来自设备端的信息进行处理和显示。

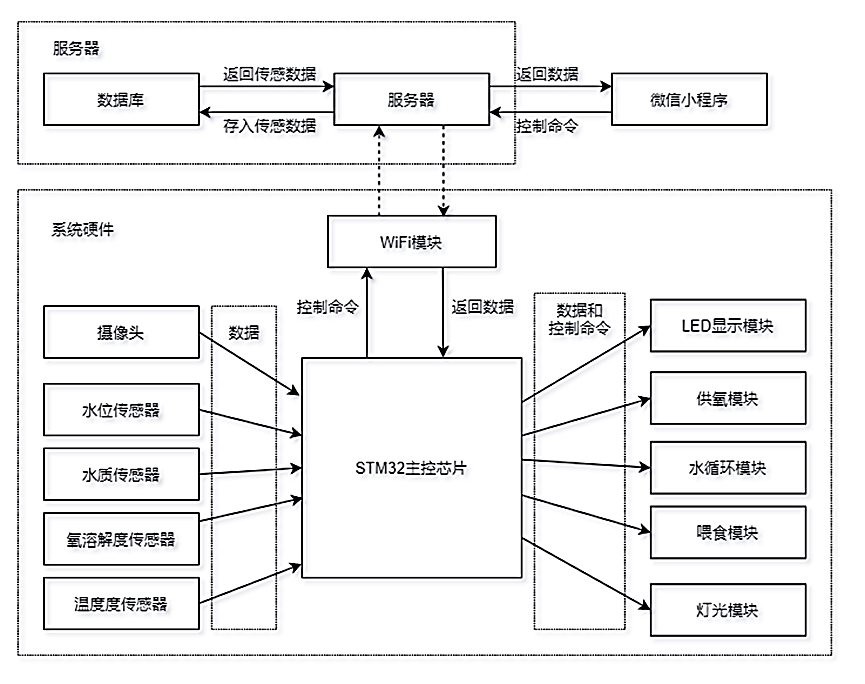
#### 7.用户登陆模块

具有注册、登陆、通过手机号重置用户名与密码的功能，并在相应步骤给予应有反馈与提示。登陆成功后方能进入主页。

#### ……

综上分析，得到智能鱼缸的顶层模块划分示意图入下：





## 四、分工

### （一）、时间规划

从立项发表的第三周开始，至项目完成的第十五周结束，我们小组的进度汇报如下：

第四周：软件、硬件小组分别收集资料、整合浏览，初步制定一个月的目标；

第五周到七周：开始着手尝试，此时处于尝试阶段，主要以学习为主，辅以某些简单功能或通用模块的实现；

第八周到第十三周：两个小组分别完成各组制定的所有功能；

第十四周：单独测试两个部分，调整成功后整合软硬件全部功能，完成整个项目的测试；

第十五周：验收。

### （二）、小组分工…

|  |  |
| --- | --- |
| 郑雨萌 | 显示模块、温度控制模块、自动喂食模块 |
| 张清智 |  |
| 崔子涵 |  |
| 徐雪莹 | 服务器配置、下行数据发送模块、设置页面和详情页面的页面设计、硬件原型的部分搭建 |
| 宋蚺 | 上行数据接收模块、用户登陆模块和主页的设计、硬件原型的部分搭建 |

#### 1.郑雨萌工作

本人在团队中担任组长和实施人员的角色。

在开发前期积极组织组员讨论，内容包括选题、开发板选择、任务分配等等。经过线上线下多次商量，我们小组确立了选题为“智能鱼缸”、硬件开发平台为野火STM32-F103ZE-霸道\_V2开发板。除此之外，本人还负责开发板和初期使用的基本元件采购。

在开发过程中，起到组织协调分配任务、督促课设进度作用。在前期五到十一周，每两周组织大家开展一次线下讨论，主要目的是计划安排课设下一步进展、因时制宜地讨论各个模块的实现方法。在后期的十三周到十五周，密集地组织大家开展了五次线上线下讨论并进行了记录。5月4日讨论确立了大家应该在本周六前完成个人负责模块的目标并决定周六进行第一次整体联调，对接软件硬件部分；5月7日在实验室进行了整体联调，并督促大家尽快完成个人模块，同时决定鱼缸的机械部分由徐雪莹、宋蚺两位同学负责；5月12日线上讨论，询问大家负责模块进展，主要目的是讨论各自模块需要进行上传的数据和接受的指令，为软硬件的对接做准备；5月17日在实验室讨论硬件对各自模块机械搭建的需求，比如摆放位置、具体接线要求，并且组织硬件小组、软件小组分工进展，硬件小组进行整体联调的同时软件小组进行机械搭建；5月31日组织大家在实验室进行最后的调试。

同时，作为项目的实施人员，本人在开题阶段负责立项申请书的“拟采取的研究方案”部分、开题汇报PPT制作；在开发阶段负责显示模块、温度控制模块、自动喂食模块的软件编程和机械搭建，对应于代码中的displaymodule、tempmodule、MicroStepDriver，包括工程文件的底层搭建，如bsp\_ds18b20、bsp\_beep、bsp\_SysTick、OLED\_I2C、relay等，并最终将功能实现于main函数中；在结题阶段负责结题报告中对应模块的编写、视频及PPT制作。

#### 2.徐雪莹工作

本人在团队开发中担任实施人员的角色，负责做好组长布置的任务和处理未完成的事务。在开发过程中，我负责的工作有项目开题时的项目预期成果部分；软件实现的服务器配置、下行数据发送模块、设置页面和详情页面的页面设计、硬件原型的部分搭建；结题报告的个人负责部分。

#### 3.宋蚺工作

本人在团队开发中担任实施人员的角色，负责做好组长布置的任务和处理未完成的事务。在开发过程中，我负责的工作有项目开题时的立项依据部分；软件实现的上行数据接收模块、用户登陆模块和主页的设计、硬件原型的部分搭建；结题报告的个人负责部分。

#### ……

## 五、模块设计

### （一）显示模块设计

#### 1. 功能实现

为实现OLED的I2C通信方式，首先应该对STM32F103ZET6的I2C进行I/O初始化，配置GPIO外设时钟、控制引脚及其速率为50MHz、模式为开漏输出等等，并且还需定义向OLED寄存器地址读写一个字节数据的函数。

此后，可以定义OLED初始化函数OLED\_Init、清屏函数OLED\_CLS、字符串显示函数OLED\_ShowStr、汉字显示函数LED\_ShowCN等函数，并将全部封装为一个模块方便使用时调用。由于ASCII字符为6\*8或8\*16两种类型有限数量点阵，而汉字为16\*16点阵字体而且数量庞大，因此显示汉字时需要根据具体情况做出补充。在此次课程设计中，为了方便，采用128×64取字软件，通过设置软件字模选项，便可生成汉字的对应点阵字节和索引。由此，生成了“智、能、鱼、缸、温、度、水、质、清、澈、浑、浊、过、低、过、高、欢、迎、使、用”20个汉字的点阵，索引为0到19。

在显示模块displaymodule中通过对OLED\_I2C各种函数的调用，根据汉字点阵索引顺序定义了开机动画显示BootAnimation、鱼缸功能显示FunctionDisplay、高温报警HighTempAlarm、低温报警LowTempAlarm、水质清澈CleanWater、水质混浊TurbidWater多个函数。如此，各模块需要在OLED屏幕上显示时直接调用即可。

#### 2. 端口设置

#define OLED\_I2C I2C1

#define OLED\_I2C\_CLK RCC\_APB1Periph\_I2C1

#define OLED\_I2C\_CLK\_INIT RCC\_APB1PeriphClockCmd

#define OLED\_I2C\_SCL\_PIN GPIO\_Pin\_6

#define OLED\_I2C\_SCL\_GPIO\_PORT GPIOB

#define OLED\_I2C\_SCL\_GPIO\_CLK RCC\_AHB1Periph\_GPIOB

#define OLED\_I2C\_SCL\_SOURCE GPIO\_PinSource6

#define OLED\_I2C\_SCL\_AF GPIO\_AF\_I2C1

#define OLED\_I2C\_SDA\_PIN GPIO\_Pin\_7

#define OLED\_I2C\_SDA\_GPIO\_PORT GPIOB

#define OLED\_I2C\_SDA\_GPIO\_CLK RCC\_AHB1Periph\_GPIOB

#define OLED\_I2C\_SDA\_SOURCE GPIO\_PinSource7

#define OLED\_I2C\_SDA\_AF GPIO\_AF\_I2C1

#### 3 实现函数及关键性代码

void I2C\_Configuration(void);

void I2C\_WriteByte(uint8\_t addr,uint8\_t data);

void WriteCmd(unsigned char I2C\_Command);

void WriteDat(unsigned char I2C\_Data);

void OLED\_Init(void);

void OLED\_SetPos(unsigned char x, unsigned char y);

void OLED\_Fill(unsigned char fill\_Data);

void OLED\_CLS(void);

void OLED\_ON(void);

void OLED\_OFF(void);

void OLED\_ShowStr(unsigned char x, unsigned char y, unsigned char ch[], unsigned char TextSize);

void OLED\_ShowCN(unsigned char x, unsigned char y, unsigned char N);

void OLED\_DrawBMP(unsigned char x0,unsigned char y0,unsigned char x1,unsigned char y1,unsigned char BMP[]);

void BootAnimation(void);

void FunctionDisplay(void);

void HighTempAlarm(void);

void LowTempAlarm(void);

void CleanWater(void);

void TurbidWater(void);

### （二）温度控制模块设计

#### 1. 功能实现

使用DS18B20检测水温前，先需要对DS18B20进行初始化，设置时钟和引脚及引脚速率为50MHz、引脚模式为推挽输出。

在读写数据之前还需检测是否存在DS18B20，如果存在则给主机返回存在脉冲，此后再为DS18B20-DATA引脚定义输入和输出模式，用于温度值的读写，需另定义函数以字节形式进行。由于温度值存如存储器后并非能够直接读取，还需存在一个温度转换函数，如果为负温度值，则将温度值加1乘0.0625，如果为正温度值，则直接与0.0625相乘。

温度检测模块tempmodule的主要功能是根据当前温度进行判断，当前温度通过函数DS18B20\_GetTemp\_MatchRom获取。如果温度在正常范围内则调用函数OLED\_ShowStr，将温度值显示于OLED屏幕上；如果温度低于最低温度，则调用LowTempAlarm函数，在OLED屏幕上报警，显示“过低”字样，并且引入继电器控制函数relay，给继电器控制端口输出高脉冲打开加热片升高水温；如果温度高于最高温度，则调用HighTempAlarm函数，在OLED屏幕上报警，显示“过高”字样，并且引入蜂鸣器函数bsp\_beep，每隔500ms鸣响一次，每次鸣响500ms。

#### 2. 端口设置

#define DS18B20\_DQ\_SCK\_APBxClock\_FUN RCC\_APB2PeriphClockCmd

#define DS18B20\_DQ\_GPIO\_CLK RCC\_APB2Periph\_GPIOD

#define DS18B20\_DQ\_GPIO\_PORT GPIOD

#define DS18B20\_DQ\_GPIO\_PIN GPIO\_Pin\_6

#define RELAY1\_GPIO\_PORT GPIOB

#define RELAY1\_GPIO\_CLK RCC\_APB2Periph\_GPIOB

#define RELAY1\_GPIO\_PIN GPIO\_Pin\_12

#define BEEP\_GPIO\_PORT GPIOC /\* GPIO端口 \*/

#define BEEP\_GPIO\_CLK RCC\_APB2Periph\_GPIOC /\* GPIO端口时钟 \*/

#define BEEP\_GPIO\_PIN GPIO\_Pin\_0 /\* 连接到蜂鸣器的GPIO\*/

#### 3 实现函数及关键性代码

uint8\_t DS18B20\_Init ( void );

void DS18B20\_ReadId ( uint8\_t \* ds18b20\_id );

float DS18B20\_GetTemp\_SkipRom ( void );

float DS18B20\_GetTemp\_MatchRom ( uint8\_t \* s18b20\_id );

double GetTemp(double HighTemp, double LowTemp, double CurrentTemp);

### （三）自动喂食模块设计

#### 1. 功能实现

首先需要对步进电机及其相关外设进行初始化，包括定时器中断优先级配置、初始化电机驱动用到的引脚、时基结构体初始化三个部分。

根据步进电机的原理为其定义函数，用于将步进电机移动给定步数。需要设置步进电机的最大加速度、最大减速度和最大速度，以及移动的步数。此处，对最大加速度和最大减速度的设置都不能过小，否则步进电机将会转动很慢，还没达到最大速度便开始减速。经过实际测试，喂食量的多少通过设置电机步数的大小较为方便，而统一设置最大加速度、最大减速度和最大速度为320rad/sec^2、320rad/sec^2和10rad/sec较为合理，此时电机的转速和转动时间适中。

对于喂食模块的设计，喂食量可根据电机步数的大小调整，喂食间隔的设置则是根据温度控制模块和主函数中的整个while循环的时序得出。具体的方案是通过电机带动投放1g鱼食，测量此时转动的电机步数，然后将其作为系数，用户投放的克数乘此系数便可得到电机应该转动的步数。而由于人为设置，温度检测模块tempmodule进入三种判断情况的时延均为1s，而主函数while循环中耗时的主体为延时1s。因此可将用户设定的天时分全部转换为秒钟，将其除以总时延2s得到循环条件。也就是给定一个用于计数的变量，每执行一次whlie循环，计数值便加1，直到计数值大于循环条件时，开启步进电机投喂鱼食。这样便做到了定时投放定量的鱼食。

#### 2. 端口设置

#define SMD\_PULSE\_TIM TIM8

#define SMD\_PULSE\_TIM\_APBxClock\_FUN RCC\_APB2PeriphClockCmd

#define SMD\_PULSE\_TIM\_CLK RCC\_APB2Periph\_TIM8

#define SMD\_PULSE\_TIM\_CHANNEL TIM\_Channel\_1

#define SMD\_PULSE\_TIM\_SETCOMPARE TIM\_SetCompare1

#define SMD\_PULSE\_TIM\_CC\_ENABLE (SMD\_PULSE\_TIM->CCER |= 1<<0)

#define SMD\_PULSE\_TIM\_CC\_DISABLE (SMD\_PULSE\_TIM->CCER &= ~(1<<0))

// 定时器输出PWM通道，PA0是通道1

#define SMD\_PULSE\_OCx\_Init TIM\_OC1Init

#define SMD\_PULSE\_OCx\_PreloadConfig TIM\_OC1PreloadConfig

// 定时器中断

#define SMD\_PULSE\_TIM\_IRQ TIM8\_UP\_IRQn

#define SMD\_PULSE\_TIM\_IRQHandler TIM8\_UP\_IRQHandler

// PWM 信号的频率 F = TIM\_CLK/{(ARR+1)\*(PSC+1)}

#define SMD\_PULSE\_TIM\_PERIOD (1)

#define SMD\_PULSE\_TIM\_PSC (72)

// 步进电机脉冲输出通道

#define SMD\_PULSE\_GPIO\_CLK RCC\_APB2Periph\_GPIOC

#define SMD\_PULSE\_PORT GPIOC

#define SMD\_PULSE\_PIN GPIO\_Pin\_6

// 步进电机方向控制

#define SMD\_DIR\_GPIO\_CLK RCC\_APB2Periph\_GPIOE

#define SMD\_DIR\_PORT GPIOE

#define SMD\_DIR\_PIN GPIO\_Pin\_4

// 步进电机输出使能引脚

#define SMD\_ENA\_GPIO\_CLK RCC\_APB2Periph\_GPIOC

#define SMD\_ENA\_PORT GPIOC

#define SMD\_ENA\_PIN GPIO\_Pin\_4

#### 3 实现函数及关键性代码

void SMD\_Init(void);

void SMD\_ENA(FunctionalState NewState);

void SMD\_Move(signed int step, unsigned int accel, unsigned int decel, unsigned int speed);

### （四）服务器配置与通信模块

#### 1.配置介绍

由微信小程序开发者文档得，每个微信小程序需要事先设置通讯域名，小程序只可以跟指定的域名进行网络通信，不能使用 IP 地址。且小程序必须使用 HTTPS/WSS 发起网络请求。请求时系统会对服务器域名使用的 HTTPS 证书进行校验，如果校验失败，则请求不能成功发起。所以在配置服务器时必须申请域名和配置SSL证书，又因为网络配置服务器域名只支持https和wss协议（加密的网络通信），还需要对服务器的安全组进行配置，开放相应的支持通信的端口。

#### 2.配置详情

图1 安全组配置

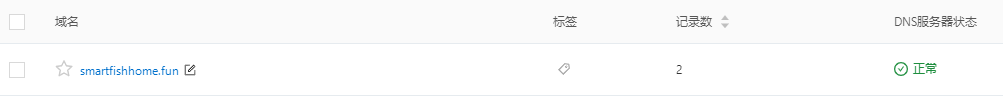


图2 已解析的域名

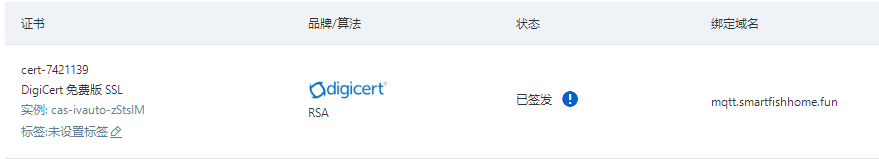


图3 域名的SSL证书

### （五）下行数据发送模块

#### 1.功能介绍

根据硬件设备的上下行传输命令分别在按钮、切换开关绑定事件，当触发相应事件函数时，读取开关/输入框的数据状态，利用client.publish方法，向服务器下行Topic：/mysmarthome/sub发送相应的命令，再由上行设备读取命令，实现小程序端对设备端的远程控制。

由于硬件读取服务器连续发送的命令会产生丢失，所以需要在发送数据时设置相应的延时。利用setTimeout方法，在连续发送的命令之间产生一定的延时，才能使硬件端正确读取。

#### 2.上下行传输命令

##### 2.1灯光控制命令

{"target":"LIGHT","value":"00"}//关闭所有灯光

{"target":"LIGHT","value":"01"}//打开彩灯，关闭白灯

{"target":"LIGHT","value":"10"}//打开白灯，关闭彩灯

{"target":"LIGHT","value":"11"}//打开白灯与彩灯

##### 2.2换水控制命令

{"target":"WATER","value":1}//进行一次换水

//只要换水模块运行一次，就会向云端发送一次消息（硬件软件控制都会返还）

{"target":"WATER\_SUCCESS","value":0}//换水成功返还消息

##### 2.3温度控制命令

{"target":"TEMP\_L","value":25}//设置的最低温度

{"target":"TEMP\_H","value":40}//设置的最高温度

##### 2.4喂食时间控制命令

{"target":"FEED\_D","value":1}//喂食时间，天

{"target":"FEED\_HO","value":10}//喂食时间，时

{"target":"FEED\_M","value":30}//喂食时间，分钟

{"target":"FEED\_A","value":5}//喂食量，单位：克（g）

#### 3.原型界面



图4 设置页面的控制部分



图5 详情页面的控制部分



图6 小程序端发送命令到服务器测试

### （六）上行数据接收模块

#### 1.功能介绍

订阅上行数据topic，将接收消息与一般数据的绑定，并对其进行简单的逻辑处理，和类型转换，再加以显示。实现小程序端对设备端数据在设置页面和详情页面的有效显示。

#### 2.上下行传输命令

##### 2.1连接服务器与订阅上行数据topic

// import { connect } from "mqtt"

import { connect } from "mqtt/dist/mqtt.js";

// const mqttHost = "mqtt.smartfishhome.fun"; //mqtt 服务器域名/IP

const mqttHost = "120.25.125.107"; //mqtt 服务器域名/IP

const mqttPort = 8084; //mqtt 服务器域名/IP

const mqttUrl = `wxs://${mqttHost}:${mqttPort}/mqtt`; // mqtt连接路径

var that = this;

that.client = connect(mqttUrl);

that.client.on("connect", function () {

console.log("成功连接mqtt服务器!");

that.client.subscribe("/mysmarthome/pub", function (err) {

if (!err) {console.log("成功订阅设备上行数据Topic!"); }

})})

##### 2.2类型转换命令

let dataFromDev = {};

dataFromDev = JSON.parse(message);

console.log(dataFromDev);2.2.3温度控制命令

##### 2.3数据绑定命令

that.Temp = dataFromDev.Temp;

that.Turbidity = dataFromDev.Turbidity;

that.Water\_level = dataFromDev.Water\_level;

##### 2.4数据显示命令

<div class="data-description">{{Temp}}℃</div>

<div class="data-description">{{Water\_level}}cm</div>

<div class="data-description">{{Turbidity?'浑浊':'清澈'}}</div>

#### 3.效果展示

### （七）用户登陆模块

#### 1.功能介绍

具有注册、登陆、通过手机号重置用户名与密码的功能，并在相应步骤给予应有反馈与提示。登陆成功后方能进入主页。借助Vant Weapp小程序 UI 组件库进行样式设计。使用微信API数据缓存wx.setStorage与wx.getStorage模拟数据库。

#### 2.具体展示与说明

默认页面为登录页，如果未曾注册应先点击注册账户进行注册，注册成功后方可进行登陆。如果忘记密码，可以点击忘记密码，根据手机号找到用户名，进行密码重置。登陆成功后即可跳转到主页。在相应步骤具有应有反馈与提示。



图7 登录页



图8 注册页





图9 忘记密码

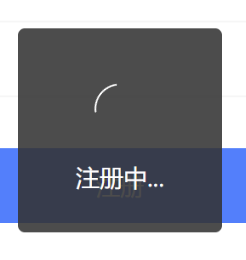


图10 反馈与提示



图10 主页

### ……

## 六、系统联调与测试

## 七、总结与心得体会

### 1.郑雨萌

本次课程设计对于我从各方面来说都是一次挑战，虽然过程比较曲折，但还是在大家的共同努力下获得了较为满意的结果。

在课设立项阶段，进展比较顺利，大家能够积极配合，快速地确立立项选题、硬件开发平台、任务分配等等内容。在项目开发的过程中，由于我们小组整体缺乏经验，进度稍有落后，此外，我个人也遇到了一些困难。一个困难是在此之前我尚未接触过STM32相关知识，从51单片机向STM32过渡花费了一定时间，为此也有过焦虑。但是通过此次课程设计，我入门了STM32开发，C语言编程能力和keil的使用也更加娴熟。第二个困难是遇到了多次编译、烧录成功但是功能实现失败的情况，例如我编写完温度控制模块后烧录入STM32并不能将数值显示在OLED屏幕上，在排除了接线问题后，我运用串口调试助手，一行一段地测试代码，每执行成功一个函数，均通过串口输出，以便告知我该代码以前均无错误，最终我发现问题出在调用DS18B20\_GetTemp\_MatchRom函数时需要匹配ROM。经过这次课程设计，我解决问题的能力也得到了提高，也增加了耐心和细心。

### 2.宋蚺、徐雪莹

通过完成本次电子综合系统课程设计，让我们对软件语言和硬件语言有了更进一步的认识，也统筹了和复习了很多以前学科学习的知识，将课堂上学习到的知识学以致用，是一次非常难得的学习体验。电子综合系统课程设计也让我们小组成员之间共同线上线下进行交流、沟通、开会探讨，发现问题、分析问题、解决问题，在知识的交流中逐步完善模块的功能，和之前单人的课程设计相比，也增强了我们修改问题、提出解决方案的能力。

同时，也使我们了解到团队合作和沟通的重要性，大家只有及时沟通，按时、按计划完成自己的任务，整个团队最后的作品才会是协调完整的。我们在大家一起解决从未遇到的种种问题的过程中，得到了锻炼，加强了自学能力和应变能力。

### ……