**单片机课程设计中期检查记录1（2024年12月 日）**

|  |  |  |
| --- | --- | --- |
| 姓名 | 学号 | 班级 |
| 李林峰 | 2022217253 | 电信科22-2班 |
| **题目：**39.电风扇模拟控制系统设计（STM32） | | |
| 设计方案和思路 | 1. 系统概述  该系统是一个基于MicroPython的风扇控制系统，使用Raspberry Pi Pico作为主控板。系统包括电机控制、键盘输入、温度传感器读取和OLED显示等功能。  2. 主要组件  Motor\_ctrl: 电机控制类，用于控制风扇的转速和启停。  Keyboard: 键盘类，用于扫描和读取键盘输入。  SSD1306\_I2C: OLED显示屏驱动类，用于在OLED屏幕上显示信息。  Main：主控制类，负责系统的初始化、温度读取、风速控制和显示更新。  3. 设计思路  3.1 电机控制  Motor\_ctrl  类初始化时设置PWM引脚和控制引脚。  set\_duty  方法用于设置电机的占空比，从而控制风速。  start  stop  方法用于启动和停止电机。  3.2 键盘输入  Keyboard  类初始化时设置行引脚为输出，列引脚为输入并启用下拉电阻。  scan  方法用于扫描键盘矩阵，检测按键按下。  get\_key  方法用于获取当前按下的键。  3.3 OLED显示  SSD1306\_I2C  类初始化时设置I2C通信和显示参数。  init\_display  方法用于初始化显示屏。  write\_cmd  write\_data  方法用于向显示屏发送命令和数据。  show  方法用于更新显示内容。  3.4 主控制逻辑  Main  类初始化时创建键盘、电机和OLED显示对象，并启动一个新的线程用于风速控制。  change\_speed  方法用于改变风速。  change\_wind\_type  方法用于改变风模式。  get\_temp  方法用于读取温度传感器数据。  wind\_control\_thread  方法在独立线程中运行，根据温度和风模式控制风速，并更新显示内容。  set\_temp\_threshold  方法用于设置温度阈值。  main  方法是主循环，持续读取键盘输入并执行相应的操作。  4. 运行流程  1. 系统初始化，创建各个组件对象。  2. 主循环中不断读取键盘输入，根据按键执行相应的操作（如改变风速、风模式、温度阈值等）。  3. 独立线程中根据温度传感器数据和风模式控制风速，并更新OLED显示内容。  5. 关键点  使用PWM控制电机转速。  使用键盘矩阵读取用户输入。  使用DHT11传感器读取温度数据。  使用SSD1306 OLED显示屏显示系统状态。  多线程处理风速控制和显示更新，确保系统响应及时。 | |
| 已完成情况 | ▢ 设计思路清晰，已完成系统设计框图  ▢ 已完成硬件电路图  ▢ 已完成软件流程图  设计思路清晰，已完成系统设计框图  已完成硬件电路图  已完成软件流程图 | |
| 遇到的问题及拟解决的方法 | 问题1：仅靠Pico引脚提供的TTL逻辑电平无法直接驱动直流电机。  解决办法：采用TB6612电机编码器对输入信号进行放大，使用5v电平为TB6612供电，Pico间接控制直流电机。  问题2：扫描键盘与控制电机在同一线程之中存在冲突。  解决办法：多线程并行运行，互不干扰。  问题3：DHT11模块的DATA线对旁边引脚的逻辑点平有干扰，导致其他引脚传输数据不稳定  解决办法：换一个GPIO接口远离其他的引脚。 | |
| 评分 |  | |