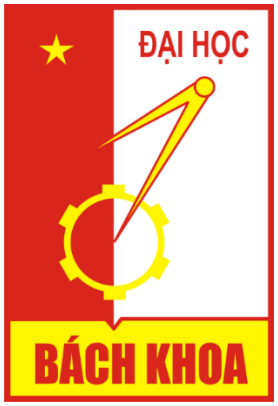
ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

****



BÁO CÁO THỰC HÀNH

HỌC PHẦN: HỆ NHÚNG

BÀI THỰC HÀNH SỐ 2

Xây dựng Hệ nhúng với Arduino

Họ và tên	:	Trương Văn Hiển
Mã số sinh viên	:	20194276
Lớp	:	727602
Giảng viên hướng dẫn	:	TS. Ngô Lam Trung

Hà Nội, tháng 06 năm 2023

MỤC LỤC

3.0. Kiểm tra hoạt động của mạch ESP32

Đèn LED nháy:

3.1. Tìm hiểu module DS1307 + AT24C32 Tiny RTC

Dựa trên datasheet của DS1307, AT24C32, DS18B20 kết hợp nghiên cứu sơ đồ mạch:

- Khả năng hoạt động đồng thời của DS1307, AT24C32 và DS18B20:
  - DS1307: Đây là một IC RTC (Real-Time Clock) và có nhiệm vụ duy trì thông tin thời gian. Nó sử dụng giao tiếp I2C
  - AT24C32: Đây là một IC EEPROM (Electrically Erasable Programmable Read-Only Memory) với dung lượng 32Kbit. Nó cũng sử dụng giao tiếp I2C.
  - DS18B20: Đây là một cảm biến nhiệt độ 1-Wire. Nó sử dụng giao tiếp 1-Wire

  Ta có thể kết nối DS1307 và AT24C32 cùng một bus I2C và kết nối DS18B20 thông qua giao tiếp 1-Wire. Vì vậy trong cùng một mạch thì DS1307, AT24C32 và DS18B20 có thể hoạt động đồng thời.

- Các chân tín hiệu cần để ghép nối với module Tiny RTC:
  - Chân nguồn: Các chân nguồn (VCC và GND) của module Tiny RTC cần được kết nối đến nguồn cung cấp điện phù hợp của mạch

- Chân tín hiệu: Chân tín hiệu SDA (Serial Data) của module Tiny RTC cần được ghép nối với chân SDA của DS1307 và AT24C32. Tương tự, chân tín hiệu SCL (Serial Clock) của module Tiny RTC cần được ghép nối với chân SCL của DS1307 và AT24C32

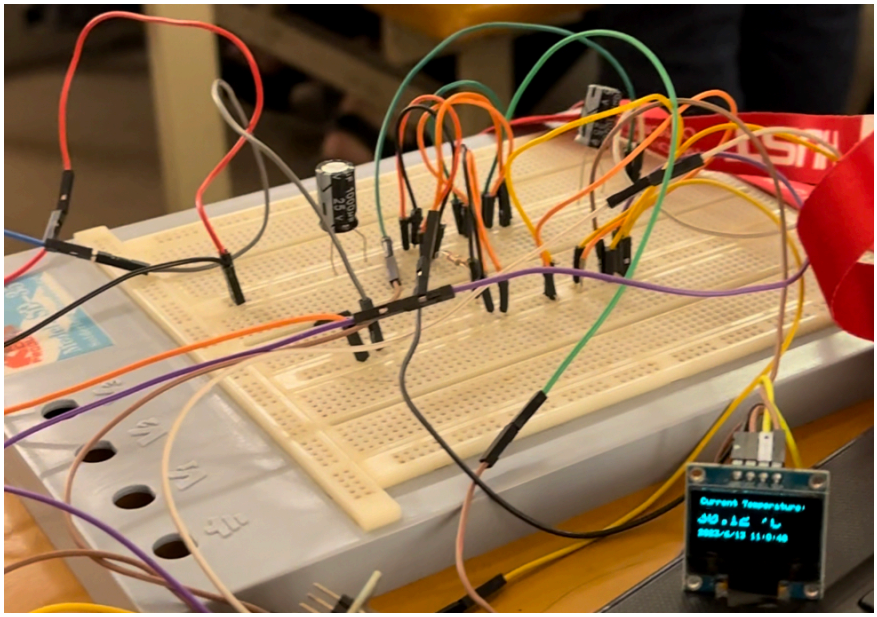
- Địa chỉ của DS1307 và AT24C32 tương ứng:
  - DS1307: Địa chỉ I2C của DS1307 là 0x68
  - AT24C32: Địa chỉ I2C của AT24C32 có thể được chọn bằng cách kết nối các chân A0, A1, A2 (chân địa chỉ) với GND hoặc VCC. Theo mặc định, khi không kết nối chân địa chỉ, địa chỉ của AT24C32 được coi là 0x50

3.2. Tìm hiểu module OLED SSD1306

Là module màn hình đồ họa với độ phân giải 128x64

3.3. Thiết kế sơ đồ mạch

Mạch sau khi lắp theo sơ đồ:



3.4. Lập trình ghép nối SSD1306

Viết ra màn hình dòng chữ: “Hello from ESP32”:

3.5. Lập trình ghép nối DS1307

3.6. Lập trình ghép nối DS18B20

3.7. Bài tập tổng hợp

- Mô tả tổ chức dữ liệu trên EEPROM của ESP32:
  - Địa chỉ: EEPROM trên ESP32 có 1 không gian địa chỉ tuyến tính, từ 0 đến kích thước tổng cộng của EEPROM
  - Dữ liệu: Có thể tổ chức dữ liệu trên EEPROM theo cách mà phù hợp với ứng dụng của bạn. Ví dụ sử dụng các cấu trúc dữ liệu như struct hoặc class để lưu trữ và truy cập các giá trị dễ dàng
  - Phương thức ghi và đọc: ESP32 cung cấp các phương thức API để ghi và đọc dữ liệu từ EEPROM. Ta có thể sử dụng các hàm như EEPROM.write() để ghi 1 byte vào địa chỉ cụ thể trên EEPROM và EEPROM.read() để đọc byte từ địa chỉ cụ thể
  - Tuổi thọ EEPROM: Mỗi vùng lưu trữ trên EEPROM có số lần ghi hạn chế trước khi bị hỏng. Vì vậy, cần chú ý đến việc sử dụng và quản lý việc ghi dữ liệu để đảm bảo tuổi thọ của EEPROM

- Mô tả thuật toán nhận và xử lý lệnh gửi từ PC:

Trong đoạn code của em, thuật toán nhận và xử lý lệnh gửi từ PC được thực hiện trong hàm **processSerialCommand()**

- Hàm **processSerialCommand()** được gọi trong hàm **loop()** để kiểm tra xem có lệnh nào được gửi từ PC thông qua cổng serial hay không
- Thuật toán bắt đầu bằng việc đọc dữ liệu từ cổng serial sử dụng hàm **Serial.available()** để kiểm tra xem có dữ liệu đến hay không
- Nếu có dữ liệu đến, thuật toán sử dụng hàm **Serial.read()** để đọc ký tự đầu tiên của lệnh
- Sau đó, thuật toán sử dụng câu lệnh **switch-case** để xác định và xử lý các trường hợp khác nhau dựa trên ký tự đầu tiên của lệnh
- Trường hợp 1: *'S' (lệnh "START").* Mạch bắt đầu đo nhiệt độ, hiển thị lên màn hình, ghi vào EEPROM và gửi dữ liệu về PC qua serial với chu kỳ 5 giây. Các bước thực hiện:
    1. Mở kết nối serial bằng cách gọi **Serial.begin()**
    2. Hiển thị thông báo "*Measurement started*" trên màn hình OLED bằng cách gọi hàm **displayMeasurementStarted()**
    3. Biến **isMeasuring**: đánh dấu đang trong quá trình đo nhiệt độ
    4. Hàm **startMeasurement()**: bắt đầu quá trình đo nhiệt độ, hiển thị lên màn hình, ghi vào EEPROM và gửi dữ liệu về PC
    5. Trong hàm **startMeasurement()**, sử dụng hàm **millis()** để theo dõi thời gian và thực hiện các hoạt động cần thiết sau mỗi khoảng thời gian 5 giây:
        - Đọc nhiệt độ từ cảm biến DS18B20
        - Đọc thời gian từ module DS1307
        - Hiển thị nhiệt độ và thời gian lên màn hình OLED
        - Ghi dữ liệu nhiệt độ và thời gian vào EEPROM
        - Gửi dữ liệu nhiệt độ và thời gian về PC qua serial
- Trường hợp 2: *'E' (lệnh "STOP").* Mạch dừng hoạt động đo nhiệt độ và hiển thị thông báo "*Measurement stopped*" trên màn hình OLED bằng cách gọi hàm **displayMeasurementStopped()**. Các bước thực hiện:

1. Biến **isMeasuring**: đánh dấu rằng không còn trong quá trình đo nhiệt độ
2. Đóng kết nối serial bằng cách gọi **Serial.end()**

- Trường hợp 3: *'G' (lệnh "GETMIN" hoặc "GETMAX")*. Mạch sẽ gửi dữ liệu nhiệt độ thấp nhất/cao nhất và thời gian tương ứng được lưu trữ trong EEPROM về PC qua serial. Các bước thực hiện:
    1. Hàm **sendMinTemperature()**: gửi dữ liệu nhiệt độ thấp nhất và thời gian tương ứng từ EEPROM về PC
    2. Hàm **sendMaxTemperature()**: gửi dữ liệu nhiệt độ tối cao nhất và thời gian tương ứng từ EEPROM về PC
- Trường hợp 4: mặc định, được sử dụng để xử lý các lệnh không hợp lệ. Trong trường hợp này, mạch sẽ gửi thông báo "Invalid command" về PC qua serial bằng cách gọi hàm **Serial.println("***Invalid command***")**
- Sau khi xử lý lệnh từ PC, thuật toán tiếp tục vòng lặp **loop()** để tiếp tục nhận và xử lý các lệnh tiếp theo nếu có

- Kết quả chạy bài 7: Em đã có gửi video kèm theo trong Assignment Báo cáo bài thực hành số 2.