

BÁO CÁO THỰC HÀNH HỆ NHÚNG BUỔI 2

VŨ HUY HOÀNG 20204981

3.1 Tìm hiểu module DS1307 + AT24C32 Tiny RTC

Yêu cầu: Đọc datasheet của DS1307, AT24C32, DS18B20 kết hợp nghiên cứu sơ đồ mạch và cho biết:

3 IC này có thể đồng thời hoạt động được không?

- Các IC DS1307, AT24C32 và DS18B20 có thể hoạt động đồng thời. Tuy nhiên, để đảm bảo tính ổn định và tránh xung đột, bạn cần đảm bảo rằng các chân tín hiệu và nguồn cung cấp được kết nối chính xác và phù hợp với sơ đồ mạch cụ thể.

Các chân tín hiệu cần để ghép nối với module Tiny RTC?

- Chân nguồn
 - Chân GND (Ground) của DS1307 nối với chân GND của module Tiny RTC.
 - Chân VCC (Power) của DS1307 nối với chân VCC của module Tiny RTC.
- Chân tín hiệu
 - Chân SDA (Serial Data) của DS1307 nối với chân SDA của module Tiny RTC.
 - Chân SCL (Serial Clock) của DS1307 nối với chân SCL của module Tiny RTC.
 - Chân SQW (Square Wave) của DS18B20 nối với chân DQ (Data Line) của module Tiny RTC.

Địa chỉ của DS1307 và AT24C32 tương ứng là bao nhiêu?

- Địa chỉ của DS1307 trong giao tiếp I2C là 0x68.
- Địa chỉ của AT24C32 trong giao tiếp I2C có thể được lựa chọn bằng các chân A0, A1 và A2. Với các chân này được kết nối đúng, địa chỉ của AT24C32 là 0x50.

3.2 Tìm hiểu module OLED SSD1306

3.3 Thiết kế sơ đồ mạch

3.4 Lập trình ghép nối SSD1306

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
```

```
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
```

```

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
// The pins for I2C are defined by the Wire-library.
// On an arduino UNO:      A4(SDA), A5(SCL)
// On an arduino MEGA 2560: 20(SDA), 21(SCL)
// On an arduino LEONARDO:  2(SDA),  3(SCL), ...
#define OLED_RESET      -1 // Reset pin # (or -1 if sharing Arduino
reset pin)
#define SCREEN_ADDRESS 0x3C ///< See datasheet for Address; 0x3D for
128x64, 0x3C for 128x32
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);

#define NUMFLAKES      10 // Number of snowflakes in the animation
example

#define LOGO_HEIGHT    16
#define LOGO_WIDTH     16

void setup() {
  Serial.begin(9600);

  // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V
internally
  if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;); // Don't proceed, loop forever
  }

  // Show initial display buffer contents on the screen --
  // the library initializes this with an Adafruit splash screen.
  display.display();
  delay(2000); // Pause for 2 seconds

  printtext();
}

void loop() {
}

void printtext(){
  display.clearDisplay();

  // Hiển thị dòng chữ "Hello from ESP8266"
  display.setTextSize(2);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);

```

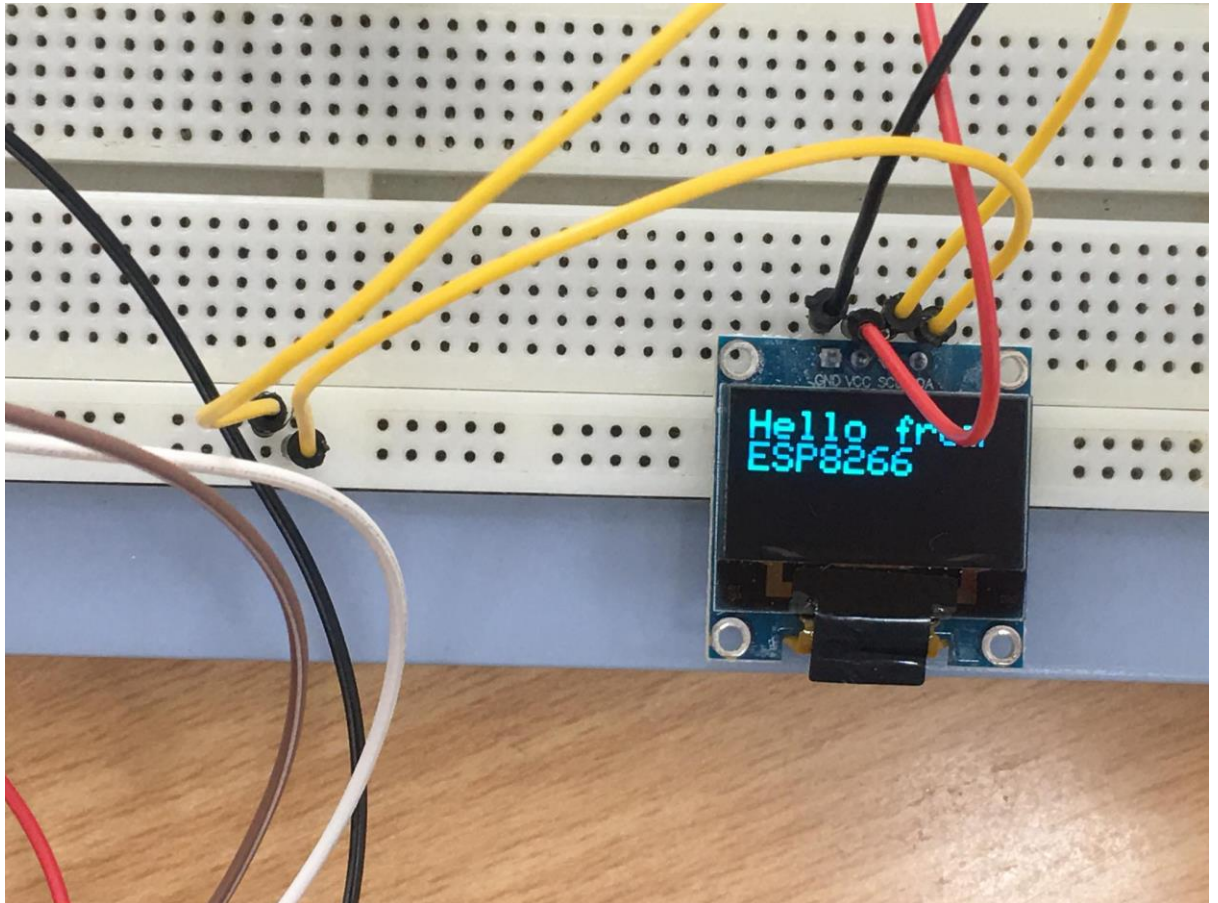
```

display.println("Hello from");
display.println("ESP8266");

// Hiển thị lên màn hình
display.display();
delay(2000);
}

```

Kết quả :



3.5 Lập trình ghép nối DS1307

```

// Date and time functions using a DS1307 RTC connected via I2C and
Wire lib
// DS1307
#include "RTClib.h"

RTC_DS1307 rtc;
// DS1306
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels

```

```
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
#define OLED_RESET      -1 // Reset pin # (or -1 if sharing Arduino
reset pin)
#define SCREEN_ADDRESS 0x3C ///< See datasheet for Address; 0x3D for
128x64, 0x3C for 128x32
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);
// Start
```

```
void setup() {
    Serial.begin(115200);

    // Khởi tạo giao tiếp I2C
    Wire.begin();

    // Khởi tạo màn hình SSD1306
    if (!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
        Serial.println(F("Màn hình SSD1306 không kết nối được."));
        while (true);
    }

    // Khởi tạo DS1307
    if (!rtc.begin()) {
        Serial.println(F("DS1307 không kết nối được."));
        while (true);
    }

    // Xóa màn hình
    display.clearDisplay();
}
```

```
void loop() {
    // Đọc ngày, tháng, năm từ DS1307
    DateTime now = rtc.now();
    int day = now.day();
    int month = now.month();
    int year = now.year();
    int hour = now.hour();
    int min = now.minute();
    int sec = now.second();
    // Hiển thị ngày tháng năm lên màn hình
    display.clearDisplay();
    display.setTextSize(2);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0, 0);
    if (day > 9){
        display.print(day);
```

```

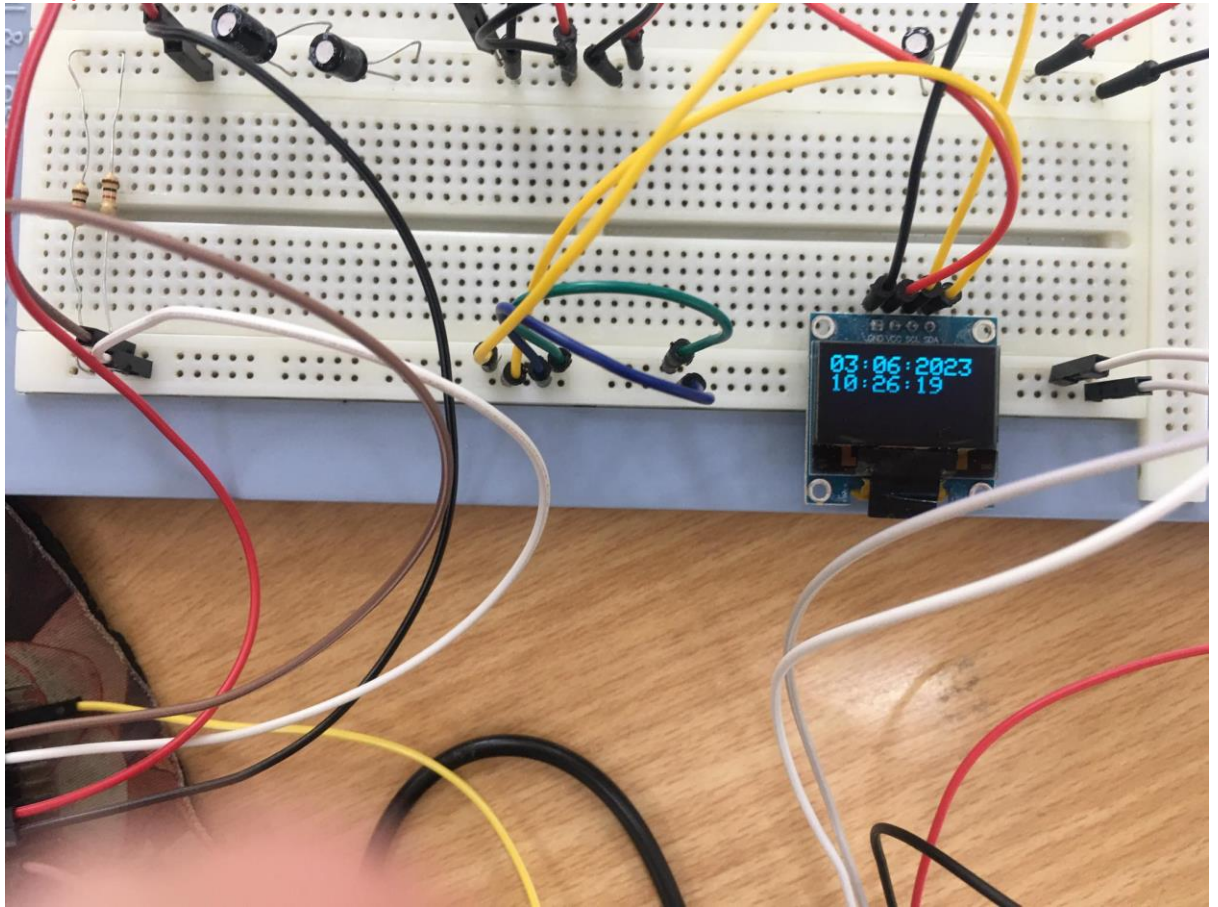
    }
    else{
        display.print("0");
        display.print(day);
    }

    display.print(":");
    if (month > 9){
        display.print(month);
    }
    else{
        display.print("0");
        display.print(month);
    }
    display.print(":");
    display.println(year);
    if (hour > 9){
        display.print(hour);
    }
    else{
        display.print("0");
        display.print(hour);
    }
    display.print(":");
    if (min > 9){
        display.print(min);
    }
    else{
        display.print("0");
        display.print(min);
    }
    display.print(":");
    if (sec > 9){
        display.print(sec);
    }
    else{
        display.print("0");
        display.print(sec);
    }
    display.display();

    // Đợi 1 giây trước khi cập nhật lại ngày tháng năm
    delay(1000);
}

```

Kết quả :



3.6 Lập trình ghép nối DS18B20

```
// Include the libraries we need
#include <OneWire.h>
#include <DallasTemperature.h>

// Data wire is plugged into port 2 on the Arduino
#define ONE_WIRE_BUS D6

// Setup a oneWire instance to communicate with any OneWire devices
(not just Maxim/Dallas temperature ICs)
OneWire oneWire(ONE_WIRE_BUS);

// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);

/*
 * The setup function. We only start the sensors here
 */

// DS1306
#include <SPI.h>
#include <Wire.h>
```



```

#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
#define OLED_RESET      -1 // Reset pin # (or -1 if sharing Arduino
reset pin)
#define SCREEN_ADDRESS 0x3C ///< See datasheet for Address; 0x3C for
128x64, 0x3D for 128x32
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);
// Start

void setup(void)
{
  // start serial port
  Serial.begin(9600);
  Serial.println("Dallas Temperature IC Control Library Demo");

  // Start up the library
  sensors.begin();
  // Khởi tạo màn hình SSD1306
  if (!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
    Serial.println(F("Màn hình SSD1306 không kết nối được."));
    while (true);
  }

  display.clearDisplay();

}

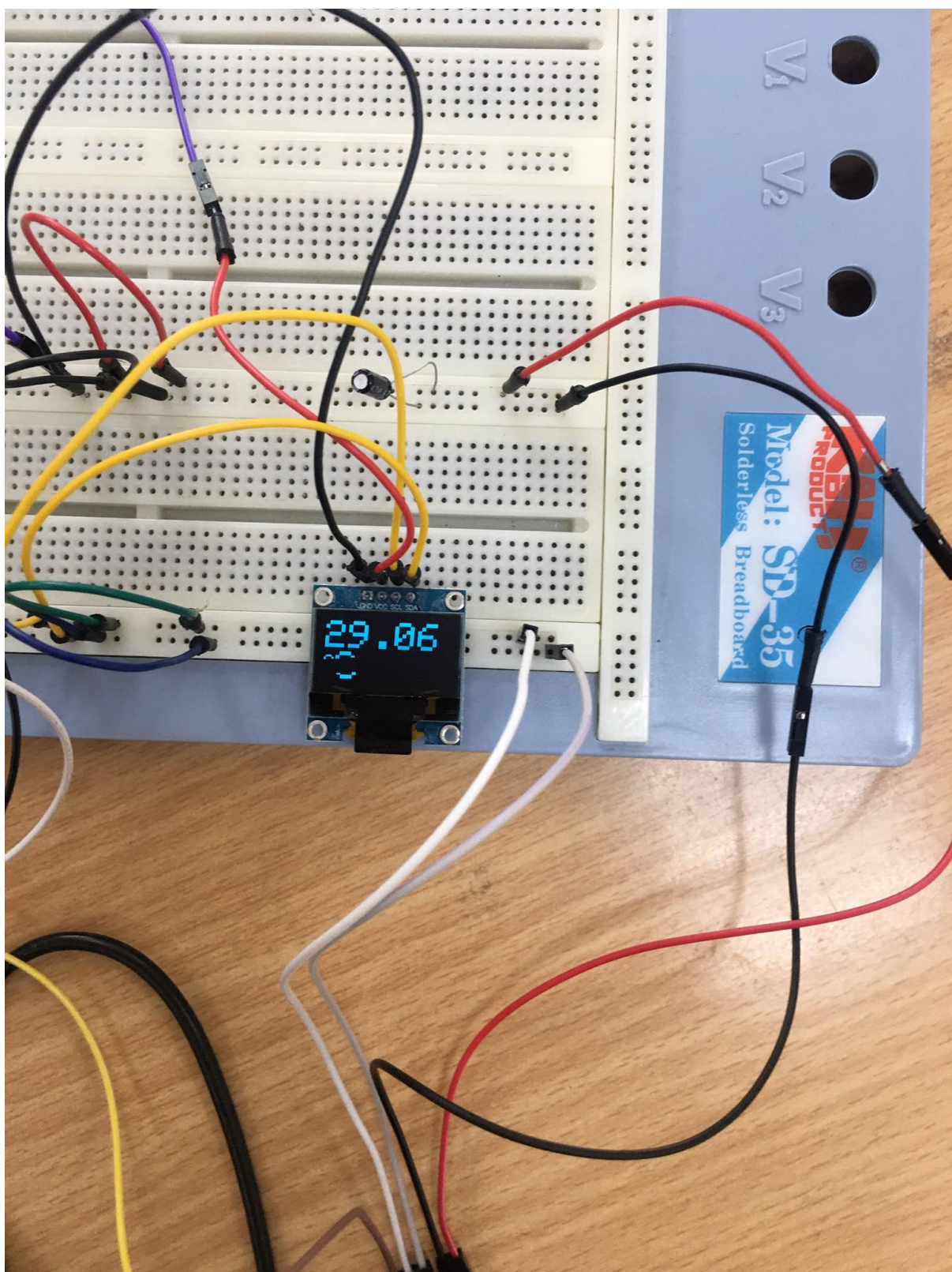
/*
 * Main function, get and show the temperature
 */
void loop(void)
{
  // call sensors.requestTemperatures() to issue a global temperature
  // request to all devices on the bus
  Serial.print("Requesting temperatures...");
  sensors.requestTemperatures(); // Send the command to get
temperatures
  Serial.println("DONE");
  // After we got the temperatures, we can print them here.
  // We use the function ByIndex, and as an example get the temperature
  from the first sensor only.
  float tempC = sensors.getTempCByIndex(0);

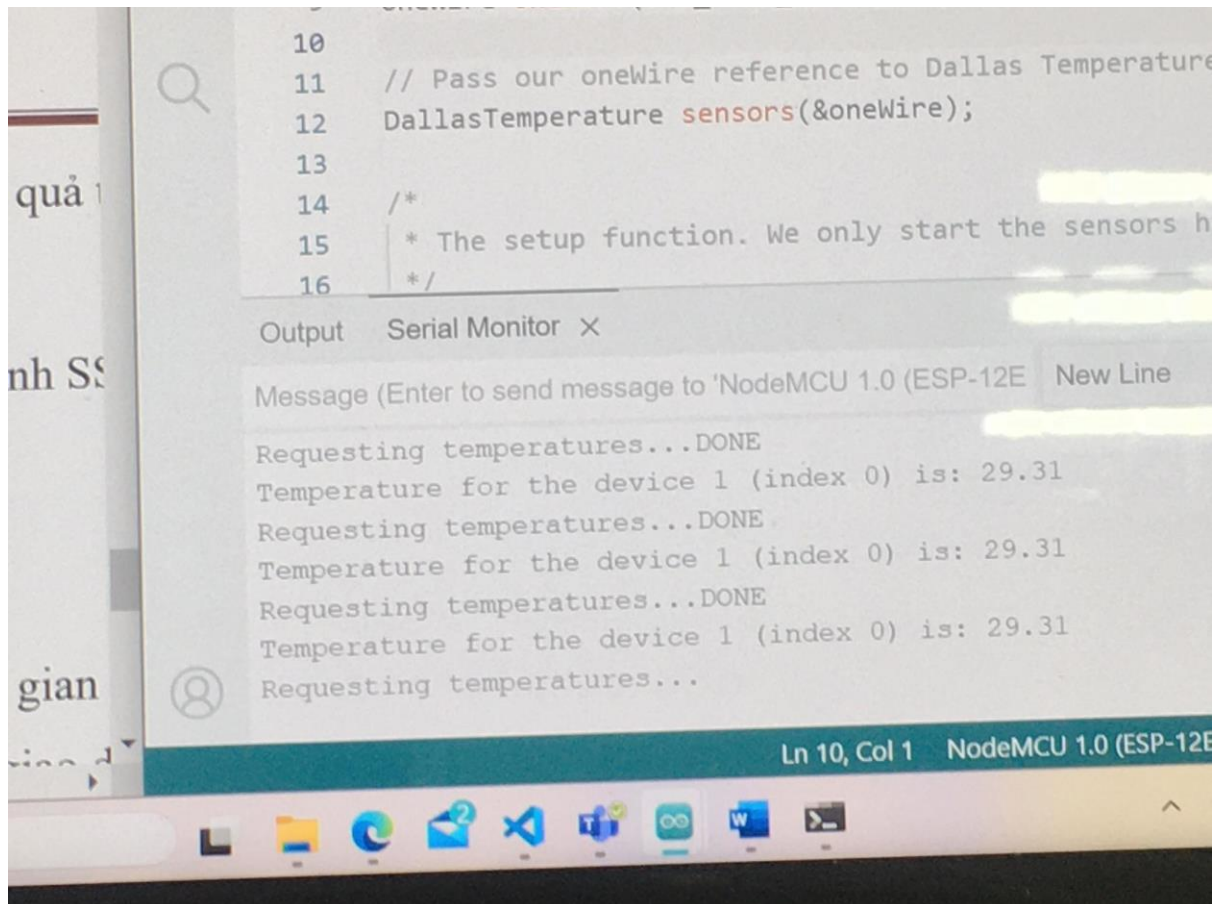
  // Check if reading was successful

```

```
if(tempC != DEVICE_DISCONNECTED_C)
{
    Serial.print("Temperature for the device 1 (index 0) is: ");
    Serial.println(tempC);
    display.clearDisplay();
    display.setTextSize(4);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0, 0);
    display.println(tempC);
    display.setTextSize(2);
    display.print("o");
    display.setTextSize(4);
    display.print("C");
    display.display();
    delay(3000);
}
else
{
    Serial.println("Error: Could not read temperature data");
    display.clearDisplay();
    display.setTextSize(2);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0, 0);
    display.println("Error: Could not read temperature data");
    display.display();
    delay(3000);
}
}
```

Kết quả:





3.7 Bài tổng hợp

TỔ CHỨC DỮ LIỆU TRÊN EEPROM:

- Em sử dụng put và get để lưu và lấy dữ liệu ra
- Cơ chế lưu dữ liệu : Em lưu dữ liệu theo 8 byte , 4 cho temp (float) và 4 cho unixsec(unsigned int) và lưu trên EEPROM theo kiểu stack với 10 ngăn (do bài yêu cầu trích xuất 10 dữ liệu mới nhất để tiết kiệm bộ nhớ)
- Với 10 stack, các dữ liệu sẽ được lưu theo các address từ 0-> 79, với 1 stack là 8 byte, 4 byte đầu là temp, 4 byte sau là unixsecs.
- Cơ chế lấy dữ liệu, sau khi read, sẽ đọc từ stack index 9 về index 0 với cơ chế 4 byte đầu của stack -> temp và 4 byte sau về unixsecs
- Ở đây em cũng có 1 hàm unixtodt để chuyển từ unixsecs về date time

NHẬN VÀ XỬ LÝ LỆNH GỬI TỪ PC

Để nhận lệnh từ PC thông qua Serial Monitor, em dùng hàm

`Serial.readStringUntil('\n')`

Để đọc chuỗi cho tới khi kết thúc, để liên tục nhận lệnh từ PC thì hàm này phải luôn chạy, tức là nó phải nằm trực tiếp trong loop (không nằm trong một hàm nào khác bên trong loop)

Tương tự, sau khi ấn start, thì bắt đầu chạy hàm start để đo và ghi dữ liệu, tức là start cũng phải chạy liên tục, song song với hàm readString kia, vì vậy chúng em đã tạo 1 biến check = 0 (False) để chỉ trạng thái đo hoặc ko đo của start, khi ấn start thì đổi check = 1, chỉ cần check = 1 thì hàm start sẽ luôn được chạy trực tiếp trong loop, tương tự thì khi ấn stop thì check = 0. Start sẽ dừng

Như đã giải thích ở phần EEPROM, khi ấn read thì em sẽ cho đọc từ stack 9 về ngược lại stack 0, tuy nhiên em đang gặp vấn đề về này thảng khi nó ra kết quả không được chính xác, và em đang để ghi chu kì xấp xỉ 1s (để dễ check), sau có thể sửa lại thành delay khoảng 4.8s (khoảng 0.2 s chạy code) để đạt được chu kì 5s

MÃ NGUỒN :

```
// Include the libraries we need
#include <OneWire.h>
#include <DallasTemperature.h>
#include<string>
#include<iostream>
using namespace std;
// Data wire is plugged into port 2 on the Arduino
#define ONE_WIRE_BUS D6

// Setup a oneWire instance to communicate with any OneWire devices
(not just Maxim/Dallas temperature ICs)
OneWire oneWire(ONE_WIRE_BUS);

// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);

/*
 * The setup function. We only start the sensors here
 */

// DS1306
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
```

```

#define OLED_RESET      -1 // Reset pin # (or -1 if sharing Arduino
reset pin)
#define SCREEN_ADDRESS 0x3C ///< See datasheet for Address; 0x3C for
128x64, 0x3C for 128x32
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);
// DS1307
#include "RTClib.h"
RTC_DS1307 rtc;
#include <EEPROM.h>
#include <ctime>
// Start
int address = 0;
void unixtodt(unsigned int unix){
    std::time_t timeValue = static_cast<std::time_t>(unix);
    std::tm* timeinfo = std::localtime(&timeValue);

    // Lấy các thành phần ngày, tháng, năm, giờ, phút, giây từ struct
tm
    int year = timeinfo->tm_year + 1900;
    int month = timeinfo->tm_mon + 1;
    int day = timeinfo->tm_mday;
    int hour = timeinfo->tm_hour;
    int minute = timeinfo->tm_min;
    int second = timeinfo->tm_sec;
    Serial.print(day);
    Serial.print("/");
    Serial.print(month);
    Serial.print("/");
    Serial.println(year);
    Serial.print(hour);
    Serial.print(":");
    Serial.print(minute);
    Serial.print(":");
    Serial.println(second);
}
void setup(void)
{
    // start serial port
    Serial.begin(9600);
    if (! rtc.begin()) {
        Serial.println("Couldn't find RTC");
        Serial.flush();
        while (1) delay(10);
    }

    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));

```



```

Serial.println("Dallas Temperature IC Control Library Demo");
EEPROM.begin(512);
// Start up the library
sensors.begin();
    // Khởi tạo màn hình SSD1306
if (!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
    Serial.println(F("Màn hình SSD1306 không kết nối được."));
    while (true);
}
// Start ds1037
Wire.begin();

display.clearDisplay();

}

/*
 * Main function, get and show the temperature
 */

void start(){
    // call sensors.requestTemperatures() to issue a global temperature
    // request to all devices on the bus
    Serial.println("Requesting temperatures and date time");
    sensors.requestTemperatures(); // Send the command to get
    temperatures

    DateTime now = rtc.now();
    Serial.flush();

    //Serial.println("DONE");
    // After we got the temperatures, we can print them here.
    // We use the function ByIndex, and as an example get the temperature
    from the first sensor only.
    float tempC = sensors.getTempCByIndex(0);
    // Check if reading was successful
    if(tempC != DEVICE_DISCONNECTED_C)
    { //write data to EEPROM
        //Serial.println(now.unixtime());
        if (address >= 80){
            address = 0;
        }

        EEPROM.put(address,tempC);
        EEPROM.put(address+sizeof(float),now.unixtime());
        EEPROM.commit();
    }
}

```

```

// // read data
// float temp ;
// EEPROM.get(address,temp);
// unsigned int unixsecs;
// EEPROM.get(address+sizeof(float),unixsecs);
// // //Serial.print("Unix time: ");
// // //Serial.println(unixsecs);
// unixtodt(unixsecs);
// Serial.println(temp);
// Serial.print(address);
address += sizeof(unsigned int);

display.clearDisplay();
int day = now.day();
int month = now.month();
int year = now.year();
int hour = now.hour();
int min = now.minute();
int sec = now.second();

// Hiển thị ngày tháng năm lên màn hình
display.clearDisplay();
display.setTextSize(2);
display.setTextColor(SSD1306_WHITE);
display.setCursor(0, 0);
//print time
{ if (day > 9){
    display.print(day);
}
else{
    display.print("0");
    display.print(day);
}

display.print(":");
if (month > 9){
    display.print(month);
}
else{
    display.print("0");
    display.print(month);
}
display.print(":");
display.println(year);
if (hour > 9){
    display.print(hour);
}
else{

```

```

        display.print("0");
        display.print(hour);
    }
    display.print(":");
    if (min > 9){
        display.print(min);
    }
    else{
        display.print("0");
        display.print(min);
    }
    display.print(":");
    if (sec > 9){
        display.println(sec);
    }
    else{
        display.print("0");
        display.println(sec);
    }
}
display.print(tempC);
display.setTextSize(1);
display.print("o");
display.setTextSize(2);
display.print("C");
display.display();
//delay(3000);
}
else
{
    Serial.println("Error: Could not read temperature data");
    display.clearDisplay();
    display.setTextSize(2);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0, 0);
    display.println("Error: Could not read temperature data");
    display.display();
    delay(3000);
}
delay(1000);
}
int check = 0;

void loop(void)
{
    if (check == 1){
        start();
    }
}

```



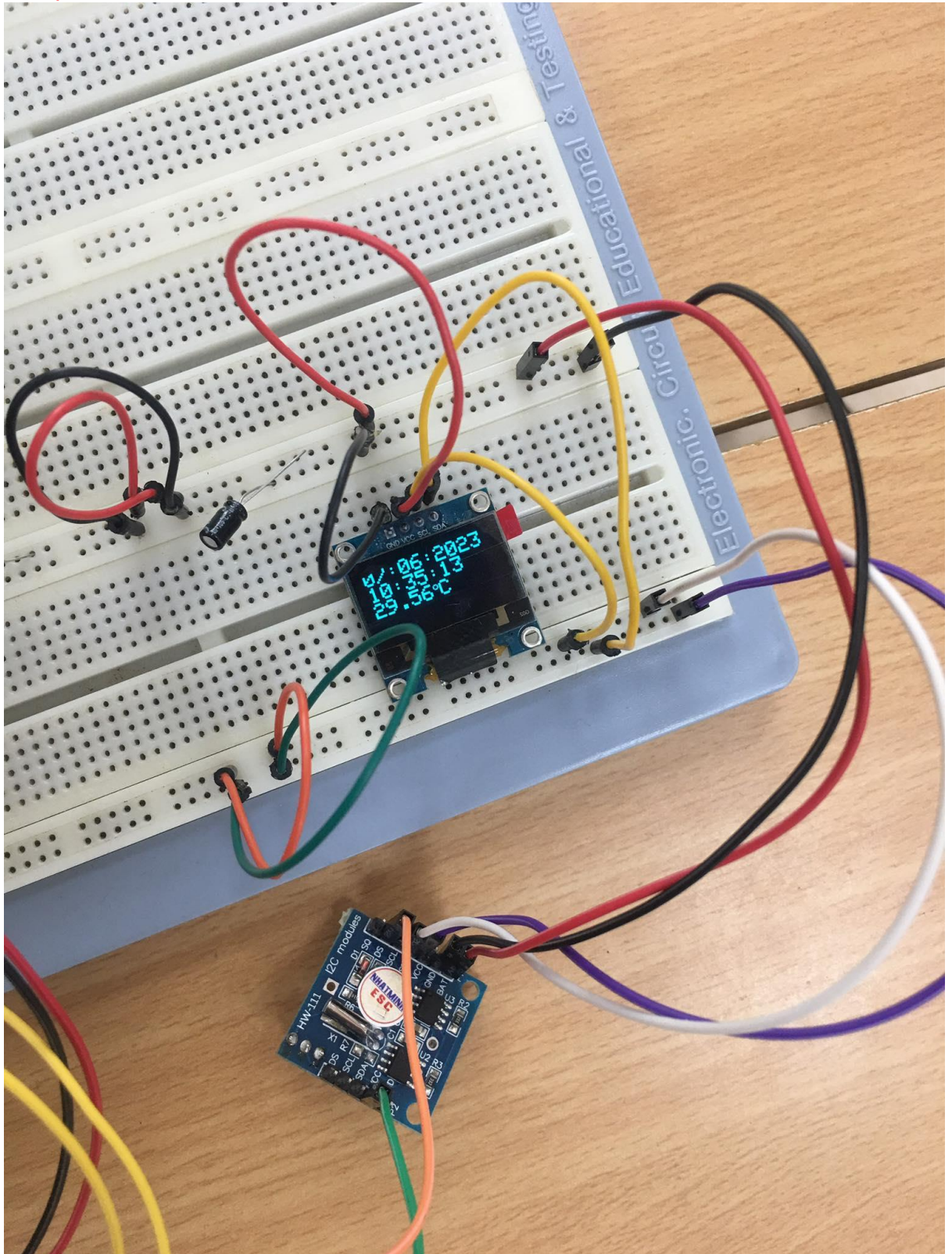
```

String receivedString = Serial.readStringUntil('\n');
if (receivedString.equals("start")){
    check = 1;
}
if (receivedString.equals("stop")){
    check = 0;
    delay(10000);
}
if (receivedString.equals("read")){
    int a ;
    float temp ;
    unsigned int unixsecs;

    for(int i = 0;i<10;i++){
        a = i*8;
        EEPROM.get(a,temp);
        EEPROM.get(a+sizeof(float),unixsecs);
        //Serial.print("Unix time: ");
        //Serial.println(unixsecs);
        unixtodt(unixsecs);
        Serial.println(temp);
        Serial.println(a);
        Serial.println("-----");
    }
}
}

```

Kết quả:



- Phần chạy code trên Serial em quên không chụp lại ạ