ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KỸ THUẬT MÁY TÍNH

BÁO CÁO ĐỒ ÁN CUỐI KỲ

Môn: Thiết kế hệ thống nhúng

Tên đồ án: Weather Station

GVHD: Trần Ngọc Đức

SVTH: Võ Trọng Hiếu – 22520450
SVTH: Lương Lê Công Nhẫn - 22521000
SVTH: Nguyễn Minh Nhật – 22521021

Thành phố Hồ Chí Minh, ngày 30 tháng 12 năm 2024.

LỜI MỞ ĐẦU

Trong bối cảnh hiện đại, việc theo dõi và dự đoán thời tiết trở thành một nhu cầu thiết yếu trong nhiều lĩnh vực, từ nông nghiệp đến du lịch. Với sự phát triển của công nghệ điện tử và Internet of Things (IoT), việc xây dựng các hệ thống theo dõi thời tiết tự động và chính xác ngày càng trở nên khả thi hơn.

Báo cáo này trình bày về đồ án sử dụng các vi điều khiển STM32 và ESP32 để phát triển một hệ thống theo dõi thời tiết. STM32, với khả năng xử lý mạnh mẽ và linh hoạt, kết hợp với ESP32, nổi bật với khả năng kết nối Wi-Fi và Bluetooth, sẽ tạo ra một giải pháp hiệu quả trong việc thu thập và truyền tải dữ liệu thời tiết theo thời gian thực.

Hệ thống sẽ được thiết kế để đo đạc các thông số như nhiệt độ, độ ẩm, lượng mưa từ đó cung cấp thông tin hữu ích cho người dùng. Qua đó, đồ án không chỉ nhằm mục đích nghiên cứu và phát triển công nghệ mà còn góp phần nâng cao nhận thức về biến đổi khí hậu và tầm quan trọng của việc theo dõi thời tiết trong cuộc sống hàng ngày.

Mong rằng báo cáo này sẽ cung cấp cái nhìn sâu sắc về quy trình thiết kế, triển khai và ứng dụng của hệ thống theo dõi thời tiết dựa trên STM32 và ESP32.

I. TÓM TẮT VÀ GIẢI THÍCH VỀ ĐỒ ÁN

-Đồ án này là sự vận dụng các kiến thức đã được học trong xuyên suốt một học kì vừa qua ở môn học Thiết kế hệ thống nhúng.

-Đồ án sẽ triển khai hệ thống sử dụng ESP32 để kết nối Internet, thực hiện các yêu cầu đến API và lấy dữ liệu từ server. Sau đó, dữ liệu được truyền về STM32 để xử lý, hiển thị lên màn hình và tích hợp điều khiển cảm ứng (touch) nhằm nâng cao tính tương tác.

1. Một số IC được sử dụng:

-STM32F401CCU6, ESPWROOM32

2. Các module khác:

-DHT11, LCD SPI 2.4inch

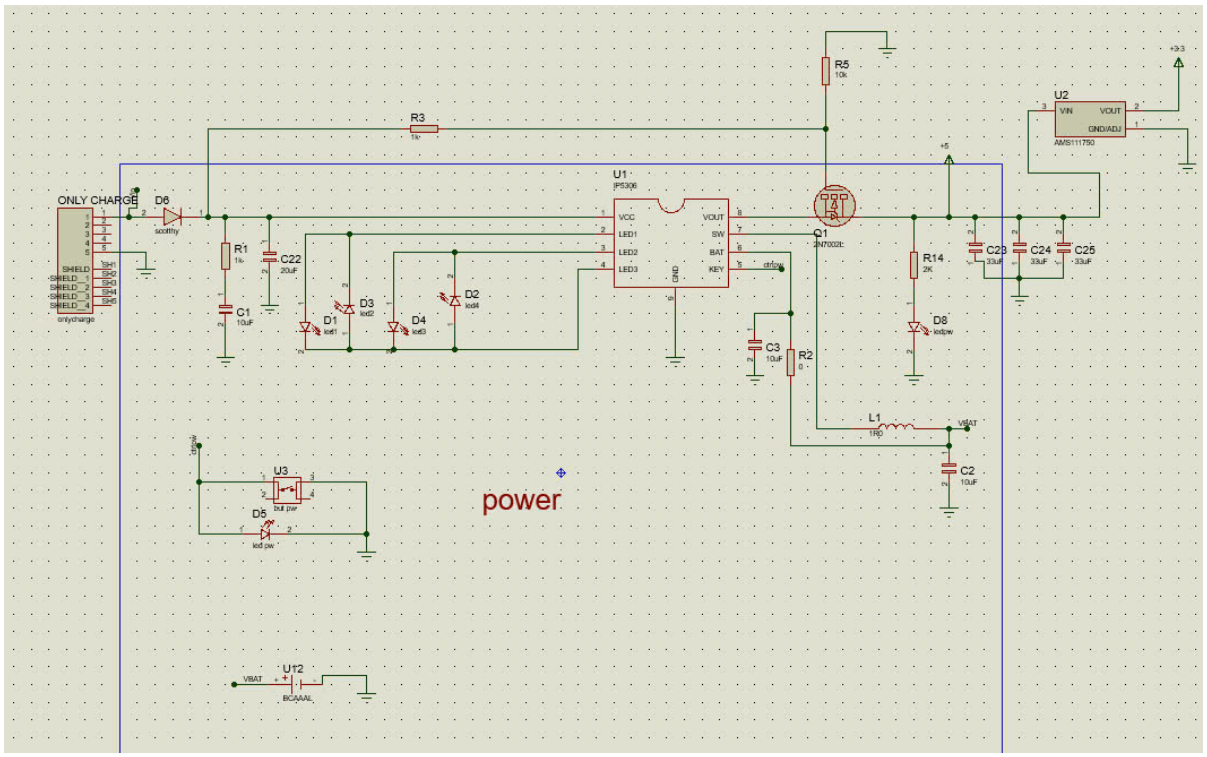
3. Danh sách thiết bị sử dụng:

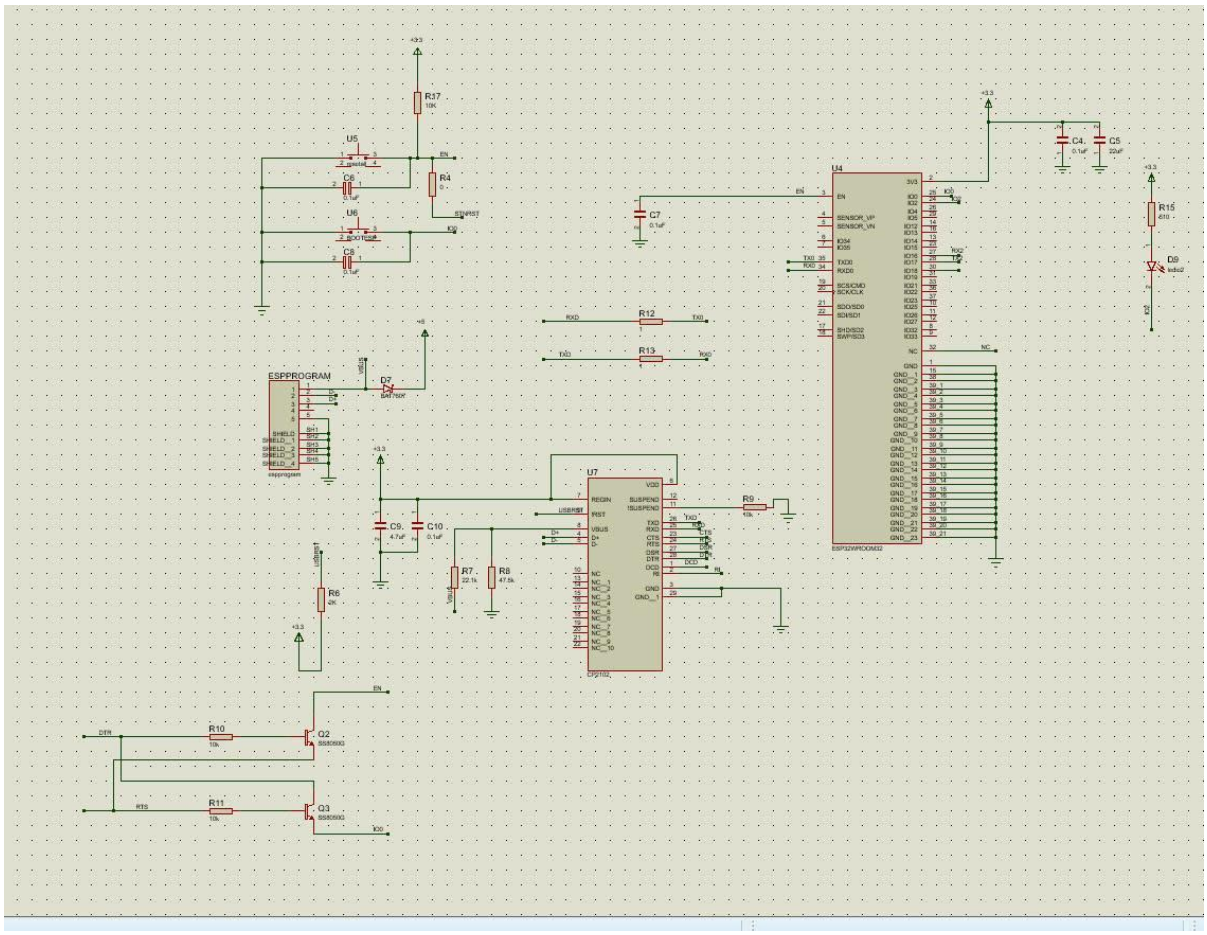
-STM32F401CCU6
-ESPWROOM32
-LCD SPI 2.4inch
-DHT11
-các thiết bị phụ trợ khác trên PCB

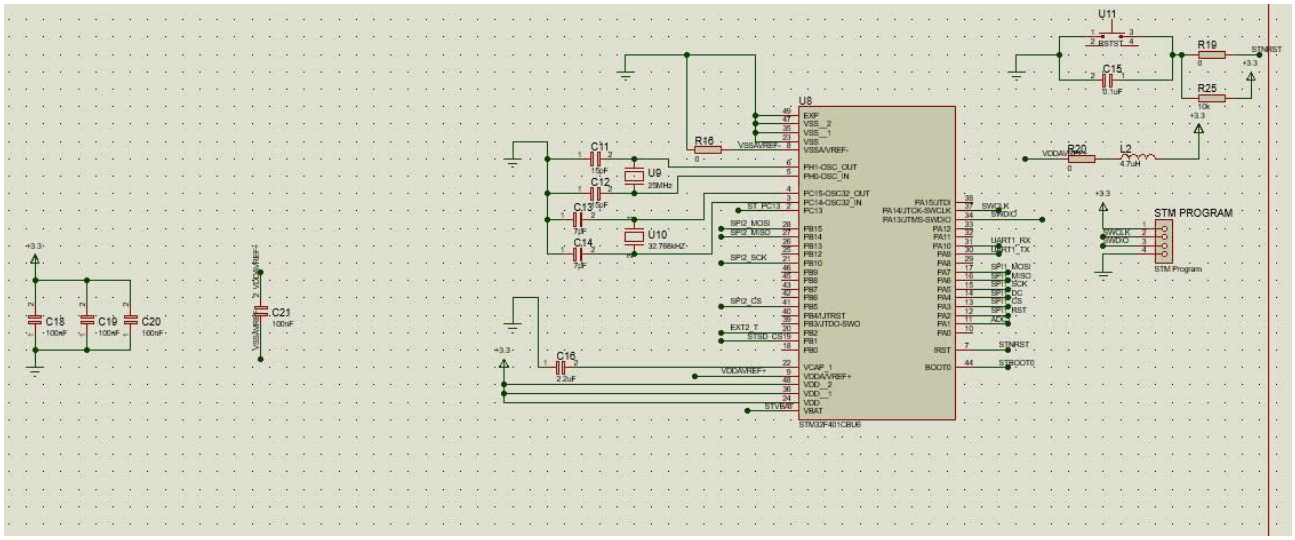
4. Về cách hoạt động:

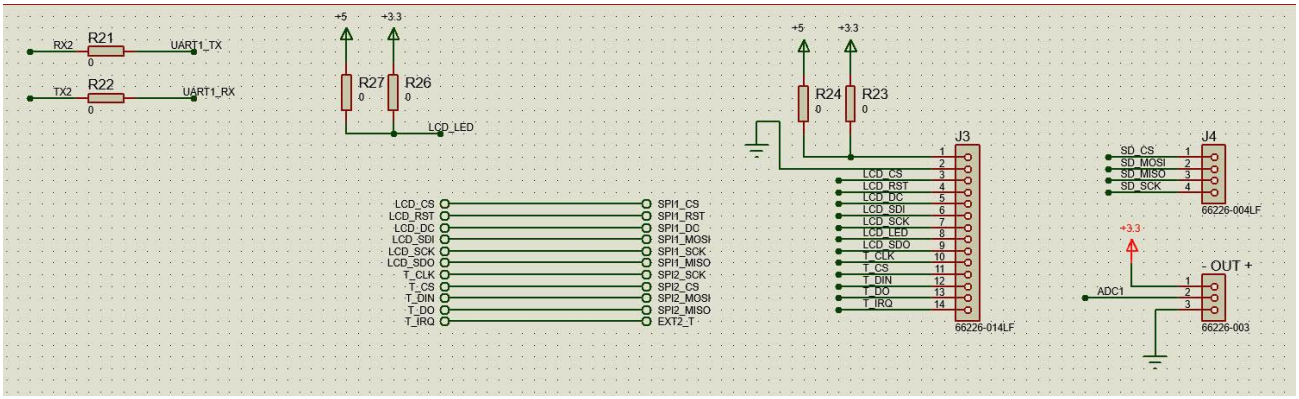
-Kết Nối Internet bằng ESP32:
ESP32 sẽ đảm nhiệm vai trò kết nối với Internet thông qua Wi-Fi. Nó sẽ gửi yêu cầu đến một API thời tiết trực tuyến để lấy dữ liệu thời tiết hiện tại.
-Lấy Dữ Liệu Từ API:
Sau khi kết nối thành công, ESP32 sẽ thực hiện các yêu cầu GET đến API thời tiết. Dữ liệu trả về thường ở định dạng JSON, bao gồm thông tin như nhiệt độ, độ ẩm, áp suất không khí và dự báo thời tiết.
-Xử Lý Dữ Liệu:
Dữ liệu nhận được từ ESP32 sẽ được truyền về STM32 để xử lý. STM32 sẽ phân tích dữ liệu JSON và trích xuất các thông tin cần thiết.
-Hiển Thị Thông Tin:
Sau khi dữ liệu đã được xử lý, STM32 sẽ điều khiển một màn hình hiển thị (LCD hoặc OLED) để trình bày thông tin thời tiết một cách trực quan. Thông tin này có thể bao gồm nhiệt độ hiện tại, độ ẩm và trạng thái thời tiết (như nắng, mưa, v.v.).
-Tính Năng Điều Khiển Cảm Ứng:
Hệ thống cũng sẽ tích hợp tính năng điều khiển cảm ứng, cho phép người dùng tương tác để cập nhật dữ liệu thời tiết mới hoặc thay đổi cài đặt hiển thị.
-Cập Nhật Dữ Liệu Định Kỳ:
ESP32 sẽ được lập trình để tự động gửi yêu cầu đến API sau một khoảng thời gian định kỳ nhằm đảm bảo thông tin thời tiết luôn được cập nhật mới nhất.

II. THIẾT KẾ TRÊN PROTEUS






U11
R19
C15
R25
U8
R16
C11
U9
C12
C13
U10
C14
R20
L2
STM PROGRAM
C18
C19
C20
C21
C16
STM32F401C8U6


R21
R22
UART1_TX
UART1_RX
R27
R26
LCD_LED
R24
R23
J3
J4
LCD_CS
LCD_RST
LCD_DC
LCD_SDI
LCD_SCK
LCD_LED
LCD_SDO
T_CLK
T_CS
T_DIN
T_DO
T_IRQ
SPI1_CS
SPI1_RST
SPI1_DC
SPI1_MOSI
SPI1_SCK
SPI1_MISO
SPI2_SCK
SPI2_CS
SPI2_MOSI
SPI2_MISO
EXT2_T
SD_CS
SD_MOSI
SD_MISO
SD_SCK
66226-004LF
66226-014LF
ADC1
66226-003

III. CODE C ESP WROOM32

Chức năng:

- Nhận tọa độ latitude và longitude
- Gửi yêu cầu API (HTTP GET)
- Phân tích và lưu trữ dữ liệu thời tiết
- Gửi dữ liệu thời tiết của một ngày và giờ cụ thể

1. Khai báo thư viện và các hằng số

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>  // Thư viện JSON
#include <SoftwareSerial.h>

#define Serial_TX 17
#define Serial_RX 16
```

2. Định nghĩa các biến cần thiết

```
char WiFi_SSID[] = "UIT Public";          // Thay thế bằng SSID của bạn
char WiFi_Password[] = "";  // Thay thế bằng mật khẩu của bạn

SoftwareSerial SerialSTM = SoftwareSerial(Serial_RX, Serial_TX);

String latitude;
String longitude;

int ESP_API = 0;   //Bat = 1 --> Call API
int ESP_DAY = 0;   //Chon ngay tra thong tin ve cho STM
int ESP_TIME = 0;  //Chon gio tra ve cho STM
```

3. Các struct cần dùng

```
struct data {
  float temp;
  float precip;
  float hud;
  float D_DAY;        //Them gui lai bien ngay
  float D_TIME;       //Them gui lai bien gio
};

struct city {
  int GET_API;
  int GET_DAY;
  int GET_TIME;
  float STM_Latitude;
  float STM_Longitute;
};
```

4. Hàm khởi tạo dữ liệu thời tiết

```
void Init_Weather_Data(DynamicJsonDocument doc, data Weather_Data[7][24]) {
    for (int i = 0; i < 7; i++) { // 7 ngày
        for (int j = 0; j < 24; j++) { // 24 giờ
            Weather_Data[i][j].temp = doc["hourly"]["temperature_2m"][i * 24 + j];
            Weather_Data[i][j].precip = doc["hourly"]["precipitation"][i * 24 + j];
            Weather_Data[i][j].hud = doc["hourly"]["relative_humidity_2m"][i * 24 + j];
            Weather_Data[i][j].D_DAY = doc["daily"]["date"][i];
            Weather_Data[i][j].D_TIME = doc["hourly"]["time"][i * 24 + j];
        }
    }
}
```

5. Hàm setup kết nối wifi

```
void setup() {
  Serial.begin(115200);  // Khởi tạo Serial Monitor
  WiFi.begin(WiFi_SSID, WiFi_Password);

  while (WiFi.status() != WL_CONNECTED) {
    Serial.print("Connecting to WiFi...");
    delay(1000);
  }

  Serial.println("Connected to WiFi");
  digitalWrite(2, LOW);
  SerialSTM.begin(9600, SWSERIAL_8N1);
  uart_lasttime = millis();
}
```

6. Hàm gọi API

```
void callApi() {
  //Them cai ESP_API = 1; vao if
  if (WiFi.status() == WL_CONNECTED && ESP_API == 1) {          //Neu wifi ket noi va ESP_API trang thai = 1 thi GET API
    HTTPClient http;
    ESP_API = 0;    //RESET ESP_API
    //Dự báo thời tiết cho một thành phố bất kì khi có KINH ĐỘ và VĨ ĐỘ

    String weatherUrl = "https://api.open-meteo.com/v1/forecast?latitude=" + latitude + "&longitude=" + longitude + "&hourly=temperature_2m,relative_humidity_2m,precipitation";
    http.begin(weatherUrl);

    int httpResponseCode = http.GET();

    if (httpResponseCode > 0) {
      String payload = http.getString();                //Thong tin cua PAYLOAD
      Serial.println("Weather Data Response: ");         //In PAYLOAD ra man hinh
      Serial.println(payload);  // In payload ra Serial Monitor
      // Phân tích cú pháp JSON
```

```
    DynamicJsonDocument doc(2048);                  // Tạo một tài liệu JSON
    DeserializationError error = deserializeJson(doc, payload);  // Phân tích cú pháp JSON

    if (error) {
      Serial.print("JSON Error: ");
      Serial.println(error.f_str());
      return;
    }
    // Lấy nhiệt độ và lượng mưa từ payload cập nhật vào mảng thời
    Init_Weather_Data(doc, Weather_Data);

    data_trant.temp = Weather_Data[ESP_DAY][ESP_TIME].temp;;
    data_trant.precip = Weather_Data[ESP_DAY][ESP_TIME].precip;
    data_trant.hud = Weather_Data[ESP_DAY][ESP_TIME].hud;

    Serial.print("Temperature: ");
    Serial.print(Weather_Data[ESP_DAY][ESP_TIME].temp);
    Serial.println(" °C");

    Serial.print("Precipitation: ");
    Serial.print(Weather_Data[ESP_DAY][ESP_TIME].precip);
    Serial.println(" mm");

    Serial.print("humidity: ");
    Serial.print(Weather_Data[ESP_DAY][ESP_TIME].hud);
    Serial.println(" mm");
   }
   else {
    Serial.print("Error: ");
    Serial.println(httpResponseCode);
   }
   http.end();
  } else {
   Serial.println("WiFi disconnected!");
  }
}
```

7. Hàm giao tiếp UART

```
void Read_UART() {
  SerialSTM.read((uint8_t*)&data_receive, sizeof(city));
  latitude = String(data_receive.STM_Latitude);    //Nap latitude       //Nhan xong moi ep kieu
  longitude = String(data_receive.STM_Longitute);  //Nap longtitude
  ESP_API = data_receive.GET_API;            //Bien nay la 1
  ESP_DAY = data_receive.GET_DAY - 1;
  ESP_TIME = data_receive.GET_TIME;
}
```

8. Vòng lặp để đọc UART liên tục và gọi API khi cờ ESP_API bật

```
void loop() {
  if (SerialSTM.available()) {
    Read_UART();
    delay(100);
}
  callApi();
  SerialSTM.write((uint8_t*)&data_trant, sizeof(data_trant));
}
```

IV. CODE C STM32

9. Khai báo các thư viện, biến và tiền khai báo các hàm cấu hình ngoại vi

```
/* USER CODE BEGIN Includes */
#include "ili9341.h"
#include "ili9341_touch.h"
#include "fonts.h"
#include "stdio.h"
#include "image.h"
#include "fonts.h"
#include "project_city.h"
/* USER CODE END Includes */
ADC_HandleTypeDef hadc1;

SPI_HandleTypeDef hspi1;
SPI_HandleTypeDef hspi2;

TIM_HandleTypeDef htim2;

UART_HandleTypeDef huart1;
//Toa do
uint16_t xa;
uint16_t ya;
//-----------------------------------------------
int Day = 0;                                    //Bien dem ID ngay
int Time = 0;
int Var_City = 0;                               //Bien dem ID thanh pho

int Slide = 0;                                  //Chuyen slide menu

uint32_t ADC_Data;                              //Khoi tao bien cho ADC_Data
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
        HAL_UART_Receive_IT(&huart1,(uint8_t *) &data_receive, sizeof(data_receive));
}
```

```
/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_ADC1_Init(void);
static void MX_TIM2_Init(void);
static void MX_USART1_UART_Init(void);
static void MX_SPI1_Init(void);
static void MX_SPI2_Init(void);
```

10. Khởi tạo tọa độ thành phố và các ngoại vi trong hàm main

```
        City_Init();                                    //Khoi tao toa do cac thanh pho
 /* USER CODE END 1 */
 /* MCU Configuration--------------------------------------------------------*/
 /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
 HAL_Init();
 /* USER CODE BEGIN Init */
 /* USER CODE END Init */
 /* Configure the system clock */
 SystemClock_Config();
 /* USER CODE BEGIN SysInit */
 /* USER CODE END SysInit */
 /* Initialize all configured peripherals */
 MX_GPIO_Init();
 MX_ADC1_Init();
 MX_TIM2_Init();
 MX_USART1_UART_Init();
 MX_SPI1_Init();
 MX_SPI2_Init();
```

11. Khai báo các cờ và lấy tín hiệu từ cảm biến

```
	/* USER CODE BEGIN WHILE */
int menu_load_flag = 1;		//Ban dau la 1 de no load Slide 0 len
int city_load_flag = 0;		//Co 1 bat len load city
int update_load_flag = 0;	//Bat len thi update thong tin nhung khong chuyen trang
//-------------------------VONG LAP XU LI CHINH CUA CHUONG TRINH
while(1){
	//---------------LAY TINH HIEU TU CAM BIEN----------------------
	HAL_ADC_Start_IT(&hadc1);
	ADC_Data  = HAL_ADC_GetValue(&hadc1);
	HAL_ADC_Stop(&hadc1);
```

12. Chuyển slide và cảm ứng chọn các thành phố

```
	//Slide chi duoc chuyen qua lai khi no dang o trang thai lua slide la 0 hoac 1
	if(ILI9341_TouchGetCoordinates(&xa, &ya)){ //Xử lí khi touch vào vùng này
					//Chuyen slide
	if ( xa >= 24 && xa <= 64 && ya >= 130  && ya <= 229){
					if(Slide == 0){
						Slide = 1;
					}
					else {
						Slide = 0;
					}
					menu_load_flag = 1;
					}
	else if (xa >= 140 && xa <= 254 && ya >= 18 && ya <= 69){
					if(Slide == 0){
						city_load_flag = 1;
						Var_City = 1;		//HaNoi
					}
					else if(Slide == 1){
						city_load_flag = 1;
						Var_City = 5;		//DaNang
					}
```

```
                    }
else if (xa >= 140 && xa <= 302 && ya >= 69 && ya <= 120){
                    if(Slide == 0){
                        city_load_flag = 1;
                        Var_City = 2;       //HoChiMinh
                    }
                    else if(Slide == 1){
                        city_load_flag = 1;
                        Var_City = 6;       //HaiPhong
                    }
                    }
else if (xa >= 140 && xa <= 302 && ya >= 120 && ya <= 171){
                    if(Slide == 0){
                        city_load_flag = 1;
                        Var_City = 3;       //CanTho
                    }
                    else if(Slide == 1){
                        city_load_flag = 1;
                        Var_City = 7;       //Tokyo
                    }
                    }
else if (xa >= 140 && xa <= 302 && ya >= 171 && ya <= 222){
                    if(Slide == 0){
                        city_load_flag = 1;
                        Var_City = 4;       //DongNai
                    }
                    else if(Slide == 1){
                        city_load_flag = 1;
                        Var_City = 8;       //Seoul
                    }
                    }
                    HAL_Delay(500);
```

13. Xử lý thông tin của thành phố được chọn để load lên màn hình

```
//--------------------PHAN NAY SE LA PHAN XU LI THONG TIN DE LOAD MAN
        HINH----------
         if(Slide == 0 && menu_load_flag == 1){
                                        ILI9341_DrawImage(0, 0, 320, 240, Menu_0);
                                        ILI9341_DrawImage(256, 130, 40, 99, SW_0);
                                        Turn_On_City_API_Status();              //Goi de bat
        GET_API = 1

                                        ILI9341_WriteString(73, 34, "HANOI", Font_11x18,
        ILI9341_BLACK ,0x23BF);
                                        ILI9341_WriteString(51, 85, "HOCHIMINH",
        Font_11x18, ILI9341_BLACK, 0x23BF);
                                        ILI9341_WriteString(68, 136, "CANTHO",
        Font_11x18, ILI9341_BLACK, 0x23BF);
                                        ILI9341_WriteString(63, 187, "DONGNAI",
        Font_11x18, ILI9341_BLACK, 0x23BF);

                                        Reset_GET_DayTime_Status();   //Reset DAY &&
        TIME
                                        menu_load_flag = 0;
         }
         else if(Slide == 1 && menu_load_flag == 1){
                                        ILI9341_DrawImage(0, 0, 320, 240, Menu_0);
                                       //Ve cai khac display string
                                        ILI9341_DrawImage(256, 130, 40, 99, SW_1);

                                        ILI9341_WriteString(68, 34, "DANANG",
        Font_11x18, ILI9341_BLACK, 0x23BF);
                                        ILI9341_WriteString(61, 85, "HAIPHONG",
        Font_11x18, ILI9341_BLACK, 0x23BF);
                                        ILI9341_WriteString(73, 136, "TOKYO",
        Font_11x18, ILI9341_BLACK, 0x23BF);
                                        ILI9341_WriteString(73, 187, "SEOUL",
        Font_11x18, ILI9341_BLACK, 0x23BF);

                                        Turn_On_City_API_Status();              //Goi de bat
```

```
        GET_API = 1
                                Reset_GET_DayTime_Status();
                                menu_load_flag = 0;
        }
        else if((Slide == 0 || Slide == 1) && Var_City >= 1 && Var_City <= 8 &&
        city_load_flag == 1){
                                ChooseTheCity(Var_City);            //Goi API
        cua mot trong cac thanh pho, get thong tin
                                Turn_Off_City_API_Status();
                                city_load_flag = 0;
                                Slide = -1;                 //Cho biet la neu ham nay
        duoc goi se chuyen sang thanh pho, khong con o menu nua
                                ILI9341_DrawImage(0, 0, 320, 240, Display);
                               //Ve icon va cac thong tin cua thanh pho do

                                Write_City_Name(Var_City);
                                Write_City_Information();
        }
        HAL_Delay(500);
```

14. Xử lý chuyển đổi ngày giờ và cập nhật dữ liệu

```
                               //----------------XU LI KHI O CAC STATUS----------
-------------
                               //----------------CHU YEU XU LI NGAY - GIO-------
--------------
                               if(ILI9341_TouchGetCoordinates(&xa, &ya)){
                               if(Var_City >= 1 && Var_City <= 8){
                              //Khi dang o Status 2-->9 tuc la o trang thai thanh pho
                                    if(xa >= 254 && xa <= 320 && ya >= 0 &&
ya <= 64){                    //An nut Menu                                  //Viet
toa do vao if
                                          Slide = 0;                         //Dat
Slide = 1: Menu_0
                                          menu_load_flag = 1;        //Dat
menu_load_flag = 1: Chuan bi load Menu_0
                                          Var_City = 0;
                              //Khong load thanh pho
                                    }
```

```
                                        else if(xa >= 254 && xa <= 286 && ya >=
176 && ya <= 208){              //An nut Ngay_0: Giam ngay
                                                update_load_flag = 1;
                                                if(Day>1)Day-=1;
                                                else Day = 7;
                                        }
                                        else if(xa >= 0 && xa <= 32 && ya >= 176
&& ya <= 208){                  //An nut Ngay_1: Tang ngay
                                                update_load_flag = 1;
                                                if(Day<7)Day+=1;
                                                else Day = 1;
                                        }
                                        else if(xa >= 254 && xa <= 286 && ya >=
208 && ya <= 240){              //An nut Gio_0: Giam gio
                                                update_load_flag = 1;
                                                if(Time>0)Time-=1;
                                                else Time=23;
                                        }
                                        else if(xa >= 0 && xa <= 208 && ya >= 32
&& ya <= 240){                  //An nut Gio_1: Tang gio
                                                update_load_flag = 1;
                                                if(Time<23)Time+=1;
                                                else Time=0;
                                        }
                                        Update_City_Information(Var_City, Day,
Time);                          //Cap nhat thong tin
                                        ChooseTheCity(Var_City);                //Gui
thong tin cho ESP yeu cau nhan du lieu cap nhat
                                }
                                }
                                //GOI CAP HIEN THI THONG TIN TREN GUI
THANH PHO
                                if(update_load_flag == 1){
                                update_load_flag = 0;           //Tat co update sau
khi vao xu li;
                                //ILI9341_WriteString();        //Cap nhat thong
tin, viet may cai da nhan trong ChooseTheCity
```

15. Cấu hình system clock

```
void SystemClock_Config(void)
{
 RCC_OscInitTypeDef RCC_OscInitStruct = {0};
 RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

 /** Configure the main internal regulator output voltage
 */
 __HAL_RCC_PWR_CLK_ENABLE();

       __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTA
       GE_SCALE2);

 /** Initializes the RCC Oscillators according to the specified parameters
 * in the RCC_OscInitTypeDef structure.
 */
 RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
 RCC_OscInitStruct.HSEState = RCC_HSE_ON;
 RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
 RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
 RCC_OscInitStruct.PLL.PLLM = 25;
 RCC_OscInitStruct.PLL.PLLN = 336;
 RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV4;
 RCC_OscInitStruct.PLL.PLLQ = 4;
 if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
 {
  Error_Handler();
 }

 /** Initializes the CPU, AHB and APB buses clocks
 */
 RCC_ClkInitStruct.ClockType =
       RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                  |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
 RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
 RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
```

```
  RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) !=
        HAL_OK)
  {
    Error_Handler();
  }
}
```

16. Cấu hình MX_ADC1

```
static void MX_ADC1_Init(void)
{

  /* USER CODE BEGIN ADC1_Init 0 */

  /* USER CODE END ADC1_Init 0 */

  ADC_ChannelConfTypeDef sConfig = {0};

  /* USER CODE BEGIN ADC1_Init 1 */

  /* USER CODE END ADC1_Init 1 */

  /** Configure the global features of the ADC (Clock, Resolution, Data Alignment and
number of conversion)
  */
  hadc1.Instance = ADC1;
  hadc1.Init.ClockPrescaler = ADC_CLOCK_SYNC_PCLK_DIV4;
  hadc1.Init.Resolution = ADC_RESOLUTION_12B;
  hadc1.Init.ScanConvMode = DISABLE;
  hadc1.Init.ContinuousConvMode = DISABLE;
  hadc1.Init.DiscontinuousConvMode = DISABLE;
  hadc1.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;
  hadc1.Init.ExternalTrigConv = ADC_SOFTWARE_START;
  hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
```

```
  hadc1.Init.NbrOfConversion = 1;
  hadc1.Init.DMAContinuousRequests = DISABLE;
  hadc1.Init.EOCSelection = ADC_EOC_SINGLE_CONV;
  if (HAL_ADC_Init(&hadc1) != HAL_OK)
  {
    Error_Handler();
  }
```

17. Cấu hình giao tiếp MX_SPI1

```
static void MX_SPI1_Init(void)
{

  /* USER CODE BEGIN SPI1_Init 0 */

  /* USER CODE END SPI1_Init 0 */

  /* USER CODE BEGIN SPI1_Init 1 */

  /* USER CODE END SPI1_Init 1 */
  /* SPI1 parameter configuration*/
  hspi1.Instance = SPI1;
  hspi1.Init.Mode = SPI_MODE_MASTER;
  hspi1.Init.Direction = SPI_DIRECTION_2LINES;
  hspi1.Init.DataSize = SPI_DATASIZE_8BIT;
  hspi1.Init.CLKPolarity = SPI_POLARITY_LOW;
  hspi1.Init.CLKPhase = SPI_PHASE_1EDGE;
  hspi1.Init.NSS = SPI_NSS_SOFT;
  hspi1.Init.BaudRatePrescaler = SPI_BAUDRATEPRESCALER_16;
  hspi1.Init.FirstBit = SPI_FIRSTBIT_MSB;
  hspi1.Init.TIMode = SPI_TIMODE_DISABLE;
  hspi1.Init.CRCCalculation = SPI_CRCCALCULATION_DISABLE;
  hspi1.Init.CRCPolynomial = 10;
  if (HAL_SPI_Init(&hspi1) != HAL_OK)
  {
    Error_Handler();
  }
```

18. Cấu hình giao tiếp MX_SPI2

```
static void MX_SPI2_Init(void)
{

  /* USER CODE BEGIN SPI2_Init 0 */

  /* USER CODE END SPI2_Init 0 */

  /* USER CODE BEGIN SPI2_Init 1 */

  /* USER CODE END SPI2_Init 1 */
  /* SPI2 parameter configuration*/
  hspi2.Instance = SPI2;
  hspi2.Init.Mode = SPI_MODE_MASTER;
  hspi2.Init.Direction = SPI_DIRECTION_2LINES;
  hspi2.Init.DataSize = SPI_DATASIZE_8BIT;
  hspi2.Init.CLKPolarity = SPI_POLARITY_LOW;
  hspi2.Init.CLKPhase = SPI_PHASE_1EDGE;
  hspi2.Init.NSS = SPI_NSS_SOFT;
  hspi2.Init.BaudRatePrescaler = SPI_BAUDRATEPRESCALER_128;
  hspi2.Init.FirstBit = SPI_FIRSTBIT_MSB;
  hspi2.Init.TIMode = SPI_TIMODE_DISABLE;
  hspi2.Init.CRCCalculation = SPI_CRCCALCULATION_DISABLE;
  hspi2.Init.CRCPolynomial = 10;
  if (HAL_SPI_Init(&hspi2) != HAL_OK)
  {
    Error_Handler();
  }
```

19. Cấu hình cho timer MX_TIM2

```
static void MX_TIM2_Init(void)
{

  /* USER CODE BEGIN TIM2_Init 0 */

  /* USER CODE END TIM2_Init 0 */

  TIM_SlaveConfigTypeDef sSlaveConfig = {0};
  TIM_MasterConfigTypeDef sMasterConfig = {0};

  /* USER CODE BEGIN TIM2_Init 1 */

  /* USER CODE END TIM2_Init 1 */
  htim2.Instance = TIM2;
  htim2.Init.Prescaler = 41999;
  htim2.Init.CounterMode = TIM_COUNTERMODE_UP;
  htim2.Init.Period = 0;
  htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
  htim2.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
  if (HAL_TIM_Base_Init(&htim2) != HAL_OK)
  {
    Error_Handler();
  }
  sSlaveConfig.SlaveMode = TIM_SLAVEMODE_EXTERNAL1;
  sSlaveConfig.InputTrigger = TIM_TS_ITR0;
  if (HAL_TIM_SlaveConfigSynchro(&htim2, &sSlaveConfig) != HAL_OK)
  {
    Error_Handler();
  }
  sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
  sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
  if (HAL_TIMEx_MasterConfigSynchronization(&htim2, &sMasterConfig) !=
HAL_OK)
  {
```

20. Cấu hình cho giao tiếp UART1

```
static void MX_USART1_UART_Init(void)
{

  /* USER CODE BEGIN USART1_Init 0 */

  /* USER CODE END USART1_Init 0 */

  /* USER CODE BEGIN USART1_Init 1 */

  /* USER CODE END USART1_Init 1 */
  huart1.Instance = USART1;
  huart1.Init.BaudRate = 9600;
  huart1.Init.WordLength = UART_WORDLENGTH_8B;
  huart1.Init.StopBits = UART_STOPBITS_1;
  huart1.Init.Parity = UART_PARITY_NONE;
  huart1.Init.Mode = UART_MODE_TX_RX;
  huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE;
  huart1.Init.OverSampling = UART_OVERSAMPLING_16;
  if (HAL_UART_Init(&huart1) != HAL_OK)
  {
    Error_Handler();
  }
```

21. Cấu hình cho các GPIO

```
static void MX_GPIO_Init(void)
{
  GPIO_InitTypeDef GPIO_InitStruct = {0};
/* USER CODE BEGIN MX_GPIO_Init_1 */
/* USER CODE END MX_GPIO_Init_1 */

  /* GPIO Ports Clock Enable */
  __HAL_RCC_GPIOH_CLK_ENABLE();
  __HAL_RCC_GPIOA_CLK_ENABLE();
  __HAL_RCC_GPIOB_CLK_ENABLE();
```

```
  /*Configure GPIO pin Output Level */
  HAL_GPIO_WritePin(GPIOA,
GPIO_PIN_0|GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4, GPIO_PIN_RESET);

  /*Configure GPIO pin Output Level */
  HAL_GPIO_WritePin(GPIOB, GPIO_PIN_5, GPIO_PIN_RESET);

  /*Configure GPIO pins : PA0 PA2 */
  GPIO_InitStruct.Pin = GPIO_PIN_0|GPIO_PIN_2;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

  /*Configure GPIO pins : PA3 PA4 */
  GPIO_InitStruct.Pin = GPIO_PIN_3|GPIO_PIN_4;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
  HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

  /*Configure GPIO pin : PB2 */
  GPIO_InitStruct.Pin = GPIO_PIN_2;
  GPIO_InitStruct.Mode = GPIO_MODE_IT_FALLING;
  GPIO_InitStruct.Pull = GPIO_PULLUP;
  HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

  /*Configure GPIO pin : PB5 */
  GPIO_InitStruct.Pin = GPIO_PIN_5;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

  /* EXTI interrupt init*/
```

V. MÔ PHỎNG VÀ CHẠY THỰC TẾ

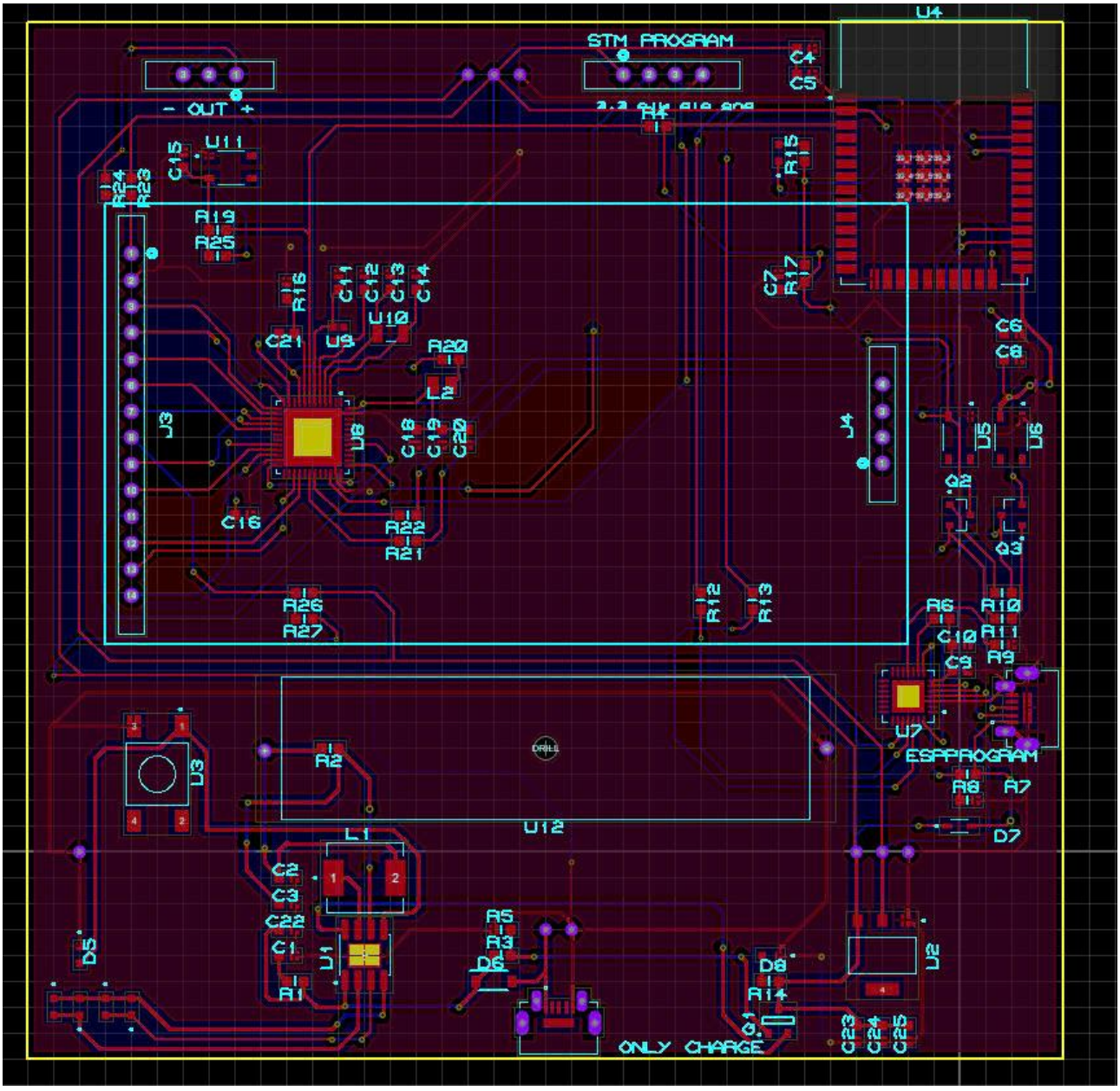
1. Link video đồ án:

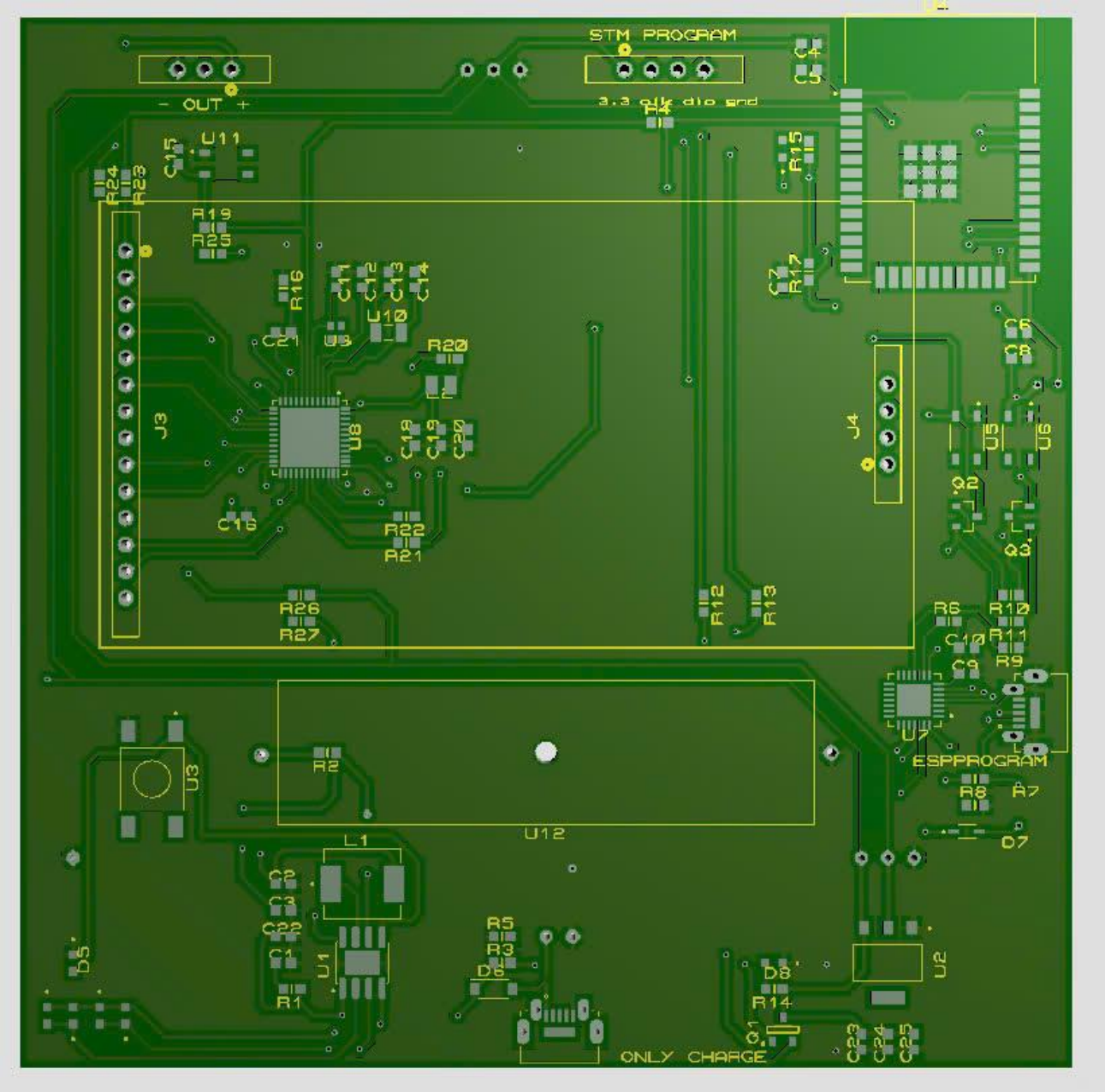
https://drive.google.com/file/d/13hSycpX5cRP29KgyhUVR9i06xU6ylTWZ/view?usp=sharing

2. Link source code assembly 8051:

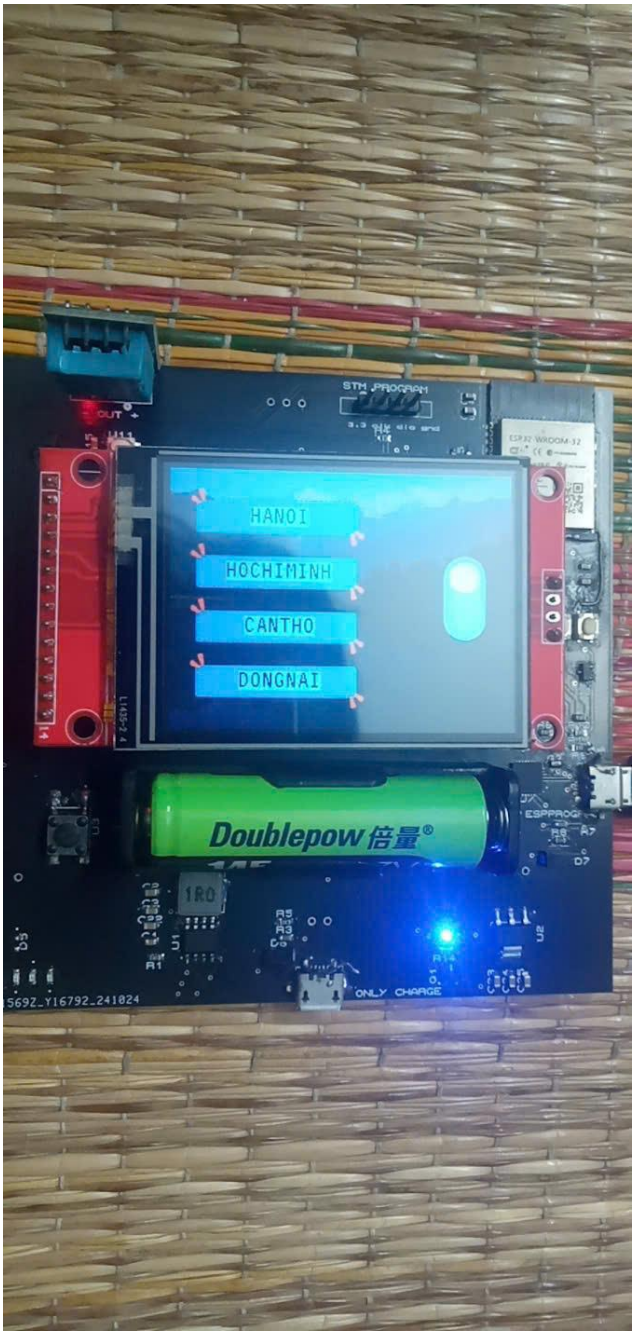
https://github.com/VoHieu1909/Weather_Station

VI. PCB LAYOUT




STM PROGRAM
- OUT +
3.3 clk dio gnd
ESPPROGRAM
ONLY CHARGE
U11
U10
U12
U7
U8
U1
U2
U3
U5
U6
J3
J4
L1
L2
Q1
Q2
Q3
D5
D6
D7
D8

VII. HÌNH ẢNH SẢN PHẨM



MỘT SỐ TÀI LIỆU THAM KHẢO

-Tài liệu STM32F401:

https://www.st.com/resource/en/datasheet/stm32f401cb.pdf

-Tài liệu ESP WROOM32:

https://www.mouser.com/datasheet/2/891/esp-wroom-32_datasheet_en-1223836.pdf

-Giao tiếp SPI:
https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/all

LỜI KẾT

Hành trình thực hiện đồ án "Thiết kế weather station sử dụng STM32 và ESP32" không chỉ là một thử thách mà còn là một cơ hội để chúng em khám phá sâu hơn về thế giới vi điều khiển và công nghệ nhúng. Qua từng bước đi, từ việc thiết kế phần cứng, lập trình phần mềm cho đến tích hợp và kiểm thử, chúng em càng thấm thía giá trị của sự sáng tạo, kiên nhẫn và tinh thần làm việc nhóm.

Dưới sự hướng dẫn tận tình của Thầy, chúng em không chỉ hoàn thiện được dự án mà còn nhận ra tầm quan trọng của việc ứng dụng kiến thức vào thực tiễn. Đây thực sự là một hành trình học tập đầy ý nghĩa, giúp chúng em trưởng thành hơn trong tư duy kỹ thuật và kỹ năng giải quyết vấn đề.

Chúng em xin gửi lời tri ân sâu sắc đến Thầy vì những đóng góp và sự hỗ trợ quý báu trong suốt quá trình thực hiện đồ án. Chúng em hy vọng rằng sản phẩm của mình không chỉ đáp ứng được yêu cầu học thuật mà còn thể hiện tinh thần sáng tạo và nhiệt huyết của chúng em đối với lĩnh vực công nghệ nhúng.