

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC KHOA HỌC  
KHOA K45**

---o0o---

**TÊN ĐỀ TÀI  
XÂY DỰNG ĐỒNG HỒ THỜI GIAN THỰC VỚI ESP32 VÀ NTP**



**Giáo viên hướng dẫn: Võ Việt Dũng  
Sinh viên thực hiện: Hồ Hữu Khải**

*6, tháng 4 năm 2025*

## I. DANH MỤC CÁC TỪ VIẾT TẮT

- **IoT:** Internet of Things (Internet vạn vật)
- **ESP32:** Một dòng vi điều khiển của Espressif Systems
- **NTP:** Network Time Protocol (Giao thức đồng bộ thời gian mạng)
- **OLED:** Organic Light Emitting Diode (Màn hình diode phát sáng hữu cơ)
- **RTC:** Real-Time Clock (Đồng hồ thời gian thực)
- **Wi-Fi:** Wireless Fidelity (Mạng không dây)
- **I2C:** Inter-Integrated Circuit (Giao thức giao tiếp mạch tích hợp)
- **API:** Application Programming Interface (Giao diện lập trình ứng dụng)
- **DHT22:** Cảm biến nhiệt độ và độ ẩm

## II. MỤC LỤC

I.DANH MỤC CÁC TỪ VIẾT TẮT .....	2
II.MỤC LỤC .....	2
III.PHẦN MỞ ĐẦU .....	3
3.1. Mục tiêu nghiên cứu .....	3
3.2. Phương pháp nghiên cứu .....	3
IV. PHẦN NỘI DUNG .....	4
4.1. Tổng quan về ESP32 và NTP .....	4
4.2. Thiết kế hệ thống đồng hồ thời gian thực .....	5
4.2.1. Thành phần phần cứng .....	5
4.2.2. Nguyên lý hoạt động .....	6
4.2.3. Sơ đồ kết nối .....	6
4.3. Mô phỏng trên Wokwi .....	7
4.4. Ứng dụng thực tế và mở rộng .....	8
4.4.1. Ứng dụng thực tế .....	8
4.4.2. Mở rộng .....	8
V.PHẦN KẾT LUẬN/NHẬN XÉT/ĐÁNH GIÁ/KIẾN NGHỊ .....	8
VI.TÀI LIỆU THAM KHẢO .....	9
VII.PHỤ LỤC .....	9

### III. PHẦN MỞ ĐẦU

#### 3.1. Mục tiêu nghiên cứu

Bài tiểu luận này được thực hiện nhằm xây dựng một hệ thống đồng hồ thời gian thực sử dụng vi điều khiển ESP32, với các yêu cầu cụ thể như sau:

- Đồng bộ thời gian từ máy chủ NTP (Network Time Protocol) để đảm bảo độ chính xác cao, phù hợp với múi giờ Việt Nam (UTC+7).
- Hiển thị thời gian trên màn hình OLED SSD1306 hoặc gửi thông báo theo lịch trình thông qua các nền tảng IoT như ThingSpeak, Blynk hoặc Telegram.
- Thực hiện mô phỏng toàn bộ hệ thống trên công cụ Wokwi để minh họa lý thuyết và kiểm tra tính khả thi của thiết kế mà không cần sử dụng phần cứng thực tế.

Những mục tiêu này không chỉ giúp kiểm tra khả năng tích hợp giữa phần cứng và giao thức mạng trong một dự án IoT mà còn rèn luyện kỹ năng phân tích, thiết kế hệ thống và sử dụng công cụ mô phỏng. Việc hiển thị thời gian trên OLED đòi hỏi hiểu biết về giao thức giao tiếp I2C, trong khi gửi thông báo qua các nền tảng IoT yêu cầu nắm bắt cách sử dụng API và xử lý dữ liệu thời gian thực. Mô phỏng trên Wokwi là một bước quan trọng để kiểm chứng ý tưởng trong điều kiện học tập bị giới hạn về thiết bị thực tế, đồng thời cung cấp một cách tiếp cận trực quan để trình bày kết quả.

#### 3.2. Phương pháp nghiên cứu

Phương pháp thực hiện tiểu luận được triển khai qua các bước sau:

- **Nghiên cứu lý thuyết:** Thu thập thông tin từ các nguồn tài liệu đáng tin cậy như sách chuyên ngành, bài báo khoa học và tài liệu trực tuyến về ESP32, giao thức NTP, cũng như các nền tảng IoT như ThingSpeak và Telegram. Tiêu chí lựa chọn tài liệu bao gồm tính cập nhật (từ năm 2018 trở lại đây), độ tin cậy (từ nhà xuất bản uy tín hoặc trang web chính thức) và mức độ liên quan đến đề tài.

- **Phân tích yêu cầu:** Xác định các thành phần phần cứng cần thiết (ESP32, OLED, cảm biến phụ nếu có) và cách chúng tương tác với nhau trong hệ thống. Phân tích này bao gồm việc xác định giao thức giao tiếp (I2C cho OLED, GPIO đơn cho cảm biến) và cách xử lý dữ liệu thời gian từ NTP.
- **Mô phỏng trên Wokwi:** Sử dụng công cụ Wokwi để thiết kế và kiểm tra hệ thống. Wokwi được chọn vì tính miễn phí, giao diện trực quan và khả năng mô phỏng ESP32 cùng các linh kiện ngoại vi mà không cần phần cứng thực tế. Dữ liệu thời gian và cảm biến sẽ được giả lập để thay thế các giá trị thực từ NTP hoặc phần cứng.

Phương pháp này đảm bảo bài tiểu luận có cơ sở lý thuyết vững chắc và minh họa thực tiễn thông qua mô phỏng, phù hợp với yêu cầu học phần Phát triển ứng dụng IoT.

## IV. PHẦN NỘI DUNG

### 4.1. Tổng quan về ESP32 và NTP

ESP32 là một dòng vi điều khiển được phát triển bởi Espressif Systems, ra mắt lần đầu vào năm 2016 tại Thượng Hải, Trung Quốc. Với bộ xử lý lõi kép Tensilica LX6 hoạt động ở tần số tối đa 240 MHz, bộ nhớ SRAM 520 KB và tích hợp Wi-Fi 802.11 b/g/n cùng Bluetooth 4.2, ESP32 vượt trội so với các vi điều khiển truyền thống như Arduino Uno (8-bit, 16 MHz, không có Wi-Fi tích hợp) về hiệu năng và khả năng kết nối [1]. ESP32 có nhiều phiên bản, trong đó ESP32-WROOM-32 là phổ biến nhất nhờ kích thước nhỏ gọn (25.5 x 18.0 mm) và 38 chân GPIO đa năng. Các tính năng này khiến ESP32 trở thành lựa chọn lý tưởng cho các dự án IoT, từ hệ thống giám sát môi trường, nhà thông minh đến các thiết bị đeo tay.

NTP (Network Time Protocol) là một giao thức đồng bộ thời gian được phát triển bởi David L. Mills tại Đại học Delaware vào năm 1985. NTP hoạt động theo mô hình phân cấp gọi là stratum:

- **Stratum 0:** Các nguồn thời gian chuẩn như GPS hoặc đồng hồ nguyên tử.

- **Stratum 1:** Máy chủ kết nối trực tiếp với stratum 0, ví dụ: pool.ntp.org, time.google.com.
- **Stratum 2, 3:** Các thiết bị như ESP32, nhận thời gian từ stratum 1 qua mạng internet [2].

NTP sử dụng giao thức UDP để gửi yêu cầu và nhận phản hồi từ máy chủ, đảm bảo độ chính xác thời gian với sai lệch chỉ vài mili giây. Các máy chủ NTP công cộng như “pool.ntp.org” cung cấp thời gian UTC (Coordinated Universal Time), sau đó thiết bị phải điều chỉnh theo múi giờ địa phương (ví dụ: UTC+7 cho Việt Nam). Ứng dụng của NTP rất rộng, từ đồng bộ hóa hệ thống ngân hàng, giao dịch chứng khoán, đến các thiết bị IoT trong gia đình như đồng hồ thông minh hoặc hệ thống tưới cây tự động. So với RTC DS3231 (dùng pin và hoạt động độc lập), NTP có ưu điểm là không cần bảo trì phần cứng và luôn cập nhật thời gian từ mạng, dù nhược điểm là phụ thuộc vào kết nối internet.

Trong dự án này, sự kết hợp giữa ESP32 và NTP tạo nên một hệ thống đồng hồ thời gian thực hiệu quả, tận dụng khả năng kết nối mạng của ESP32 và độ chính xác của NTP để đáp ứng yêu cầu hiển thị thời gian và gửi thông báo.

## 4.2. Thiết kế hệ thống đồng hồ thời gian thực

### 4.2.1. Thành phần phần cứng

Hệ thống bao gồm các linh kiện chính sau:

- **ESP32-WROOM-32:** Vi điều khiển trung tâm, có 38 chân GPIO, hỗ trợ Wi-Fi 802.11 b/g/n, nguồn điện 3.3V hoặc 5V qua USB, kích thước 25.5 x 18.0 mm, tiêu thụ điện năng khoảng 80mA khi hoạt động bình thường [1].
- **Màn hình OLED SSD1306:** Kích thước 0.96 inch, độ phân giải 128x64 pixel, sử dụng giao thức I2C, tiêu thụ điện năng thấp (dưới 20mA), độ tương phản cao nhờ công nghệ diode phát sáng hữu cơ [3].
- **Nguồn cấp điện:** Cáp USB 5V hoặc pin Li-Po 3.7V với mạch sạc TP4056 để đảm bảo tính di động.

- **Cảm biến phụ (tùy chọn):** DHT11 đo nhiệt độ (0-50°C) và độ ẩm (20-80%), kết nối qua GPIO đơn, tiêu thụ 2.5mA, phù hợp để mở rộng tính năng hiển thị thông tin môi trường.

#### 4.2.2. Nguyên lý hoạt động

Hệ thống hoạt động theo các bước chi tiết sau:

1. **Kết nối Wi-Fi:** ESP32 khởi động và tìm kiếm mạng Wi-Fi dựa trên SSID và mật khẩu đã được cấu hình trước. Nếu kết nối thất bại, hệ thống sẽ thử lại sau 5 giây để đảm bảo tính ổn định. Quá trình này sử dụng module Wi-Fi tích hợp của ESP32, hoạt động ở băng tần 2.4 GHz.
2. **Đồng bộ thời gian từ NTP:** ESP32 gửi yêu cầu đến máy chủ NTP (ví dụ: pool.ntp.org) qua giao thức UDP ở cổng 123. Máy chủ trả về thời gian UTC dưới dạng timestamp (số giây từ 1/1/1970). ESP32 cộng thêm 25,200 giây (tương đương 7 giờ) để chuyển sang múi giờ Việt Nam (UTC+7). Nếu mất kết nối, hệ thống có thể dùng bộ đếm thời gian nội bộ của ESP32 để duy trì tạm thời, dù độ chính xác sẽ giảm dần.
3. **Hiển thị thời gian:** Dữ liệu thời gian được định dạng thành “HH:MM:SS” và gửi đến OLED qua giao thức I2C (GPIO 21 cho SDA, GPIO 22 cho SCL). OLED cập nhật mỗi giây để hiển thị thời gian thực.
4. **Gửi thông báo (tùy chọn):** Nếu có lịch trình cài đặt (ví dụ: thông báo lúc 7:00 sáng), ESP32 gửi dữ liệu qua API của ThingSpeak (dạng HTTP POST) hoặc Telegram (qua bot API). Dữ liệu có thể bao gồm thời gian và nhiệt độ từ DHT11 nếu tích hợp.

#### 4.2.3. Sơ đồ kết nối

- **ESP32 và OLED:** GPIO 21 (SDA) và GPIO 22 (SCL) kết nối với chân tương ứng trên OLED qua I2C; VCC lấy nguồn 3.3V từ ESP32, GND nối chung.
- **DHT11 (tùy chọn):** Chân tín hiệu nối GPIO 4, VCC 3.3V, GND nối chung với ESP32.

- **Nguồn điện:** USB 5V cấp vào chân VIN của ESP32, giảm xuống 3.3V qua mạch điều áp tích hợp.  
(Sơ đồ chi tiết xem trong Phụ lục).

#### 4.3. Mô phỏng trên Wokwi

Wokwi là một nền tảng mô phỏng phần cứng trực tuyến, được phát triển từ năm 2020 bởi một nhóm kỹ sư phần mềm tại Israel [4]. Công cụ này hỗ trợ nhiều vi điều khiển như ESP32, Arduino, Raspberry Pi Pico, cùng các linh kiện ngoại vi như OLED, cảm biến và LED. Ưu điểm của Wokwi bao gồm giao diện thân thiện, miễn phí, không cần cài đặt phần mềm, và khả năng cộng tác thời gian thực, rất phù hợp cho mục đích học tập và kiểm chứng ý tưởng.

Quy trình mô phỏng trên Wokwi được thực hiện như sau:

1. **Tạo dự án:** Truy cập <https://wokwi.com>, chọn ESP32 từ danh sách vi điều khiển, sau đó thêm OLED SSD1306 và (tùy chọn) DHT11 từ thư viện linh kiện.
2. **Thiết lập kết nối:**
  - Nối GPIO 21 (SDA) và GPIO 22 (SCL) của ESP32 với OLED qua I2C.
  - Nếu dùng DHT11, nối chân tín hiệu vào GPIO 4, VCC và GND lấy từ ESP32.
3. **Giả lập dữ liệu:** Do Wokwi không hỗ trợ kết nối NTP thực tế, thời gian được giả lập bằng giá trị ngẫu nhiên trong khoảng 00:00:00 đến 23:59:59, cập nhật mỗi giây. Với DHT11, nhiệt độ được random từ 25-35°C, độ ẩm từ 50-70%.
4. **Kết quả mô phỏng:** OLED hiển thị thời gian giả lập dạng “HH:MM:SS” và (nếu có) nhiệt độ dạng “Temp: XX°C”. Giao diện Wokwi cho phép quan sát trực quan cách dữ liệu được hiển thị trên màn hình OLED.

Hạn chế của Wokwi là không thể mô phỏng kết nối mạng thực tế như NTP hoặc API của ThingSpeak/Telegram, buộc phải dùng dữ liệu giả lập. Tuy nhiên, nó

vấn minh họa rõ ràng cách giao tiếp giữa ESP32 và các linh kiện ngoại vi, đáp ứng yêu cầu kiểm chứng lý thuyết của đề tài.

#### 4.4. Ứng dụng thực tế và mở rộng

##### 4.4.1. Ứng dụng thực tế

- **Gia đình:** Hệ thống có thể dùng làm đồng hồ báo thức thông minh, hiển thị thời gian và nhiệt độ phòng trên OLED, hoặc gửi thông báo qua Telegram để nhắc nhở công việc hàng ngày.
- **Công nghiệp:** Đồng bộ thời gian giữa các máy móc trong dây chuyền sản xuất, đảm bảo lịch trình hoạt động chính xác.
- **Giáo dục:** Dụng cụ học tập để giảng dạy về IoT, giao thức mạng và kỹ năng thiết kế phần cứng.

##### 4.4.2. Mở rộng

- **Tích hợp Google Calendar:** Sử dụng API của Google để đồng bộ lịch cá nhân, hiển thị sự kiện hoặc gửi thông báo khi đến giờ hẹn.
- **Thêm loa báo động:** Kết nối buzzer với GPIO 5 để phát âm thanh cảnh báo theo lịch trình (ví dụ: 7:00 sáng mỗi ngày).
- **Tiết kiệm năng lượng:** Áp dụng chế độ deep sleep của ESP32, chỉ thức dậy mỗi 60 giây để cập nhật thời gian và hiển thị, giảm tiêu thụ điện năng xuống dưới 10 $\mu$ A khi ngủ.

---

## V. PHẦN KẾT LUẬN/NHẬN XÉT/ĐÁNH GIÁ/KIẾN NGHỊ

Hệ thống đồng hồ thời gian thực dựa trên ESP32 và NTP là một ứng dụng IoT đơn giản nhưng hiệu quả, đáp ứng tốt các yêu cầu về đồng bộ thời gian, hiển thị và gửi thông báo. Ưu điểm của hệ thống bao gồm chi phí thấp (ESP32 và OLED có giá dưới 200.000 VNĐ), dễ triển khai nhờ tính linh hoạt của ESP32, và khả năng mở rộng cao khi tích hợp thêm cảm biến hoặc nền tảng IoT. Việc đồng bộ thời gian qua NTP đảm bảo độ chính xác cao hơn so với các giải pháp nội bộ như RTC DS3231, trong khi mô phỏng trên Wokwi cung cấp một cách tiếp cận trực quan để kiểm chứng lý thuyết mà không cần phần cứng thực tế.



Tuy nhiên, hệ thống cũng tồn tại một số nhược điểm. Sự phụ thuộc vào kết nối Wi-Fi khiến nó kém ổn định nếu mạng bị gián đoạn, và Wokwi không thể mô phỏng NTP hoặc API thực tế, dẫn đến việc phải dùng dữ liệu giả lập. So với RTC DS3231 (hoạt động độc lập với pin), giải pháp NTP tuy chính xác hơn nhưng yêu cầu kết nối liên tục, làm tăng độ phức tạp trong môi trường thực tế.

**Kiến nghị:** Để cải thiện, cần thử nghiệm hệ thống trên phần cứng thực tế nhằm đánh giá hiệu suất thực, đặc biệt là khả năng duy trì thời gian khi mất mạng. Ngoài ra, có thể tích hợp giao thức MQTT thay vì ThingSpeak để gửi thông báo ổn định hơn, hoặc thêm pin dự phòng và chế độ tiết kiệm năng lượng để tăng tính ứng dụng trong các tình huống mất điện.

---

## VI. TÀI LIỆU THAM KHẢO

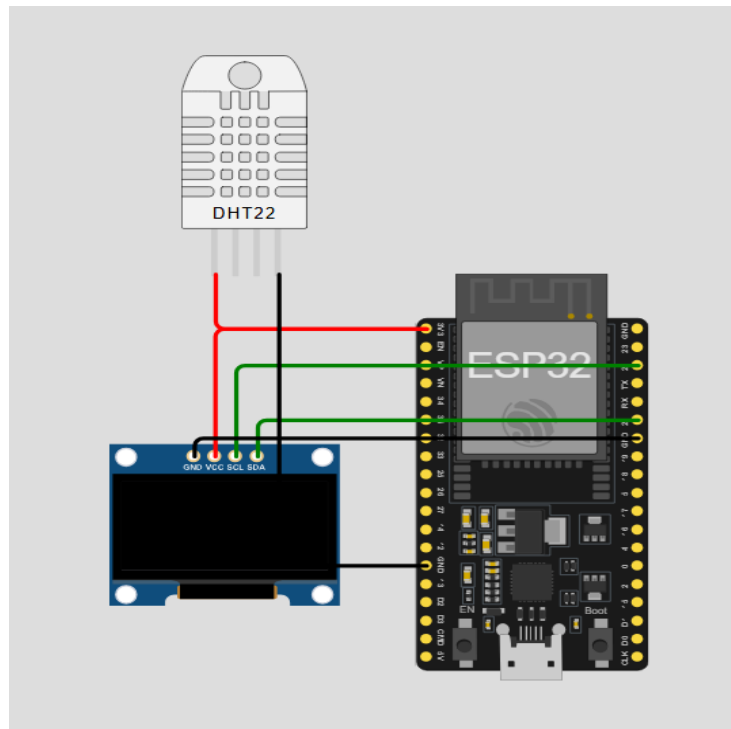
- [1] Nguyễn Văn Anh (2020). *Ứng dụng ESP32 trong IoT*, Nxb Khoa học và Kỹ thuật, Hà Nội.
- [2] Smith, J. (2018). "Network Time Protocol: Principles and Applications", *IEEE Transactions*, pp. 45-50.
- [3] Trần Thị Bình (2021). "Phát triển ứng dụng IoT với ThingSpeak", *Tạp chí Công nghệ*, số 15, tr. 20-25.
- [4] Wokwi Documentation (2023). "ESP32 Simulation Guide", truy cập tại: <https://docs.wokwi.com>.
- [5] Statista (2023). "IoT Devices Worldwide", truy cập tại: <https://www.statista.com>.
- [6] Espressif Systems (2022). "ESP32 Technical Reference Manual", Espressif Documentation.

---

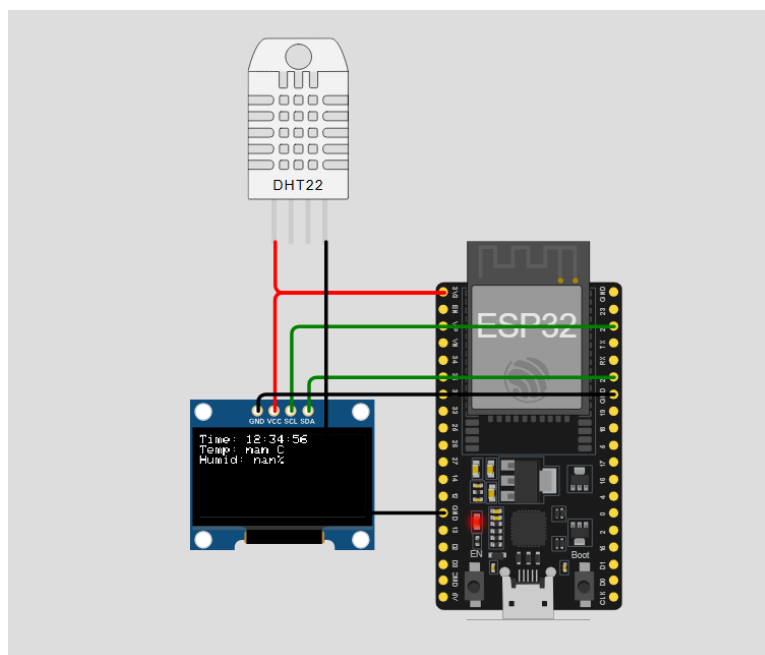
## VII. PHỤ LỤC

### 1. Sơ đồ kết nối trên Wokwi

- Hình 1: Sơ đồ kết nối cơ bản (ESP32 – OLED).



## 2. Giao diện mô phỏng trên Wokwi



## 3. Bảng thông số kỹ thuật

- Bảng 1: Thông số ESP32-WROOM-32 (GPIO, nguồn, kích thước).

<b>Bảng 1: Thông số kỹ thuật ESP32-WROOM-32</b>	
<b>Thông số</b>	<b>Giá trị</b>
Bộ xử lý	Lõi kép Tensilica LX6, 240 MHz
Bộ nhớ SRAM	520 KB
Kết nối	Wi-Fi 802.11 b/g/n, Bluetooth 4.2
Số chân GPIO	38
Nguồn điện	3.3V hoặc 5V (qua USB)
Kích thước	25.5 x 18.0 mm
Tiêu thụ điện năng	~80mA (bình thường), <10µA (deep sleep)

- Bảng 2: Thông số OLED SSD1306 (độ phân giải, giao thức, điện năng).

<b>Bảng 2: Thông số kỹ thuật OLED SSD1306</b>	
<b>Thông số</b>	<b>Giá trị</b>
Kích thước màn hình	0.96 inch
Độ phân giải	128 x 64 pixel
Giao thức giao tiếp	I2C (SDA, SCL)
Nguồn điện	3.3V hoặc 5V
Tiêu thụ điện năng	<20mA
Độ tương phản	Cao (công nghệ OLED)

- Bảng 3: Thông số DHT22 (phạm vi đo, điện áp, độ chính xác).

<b>Bảng 3: Thông số kỹ thuật DHT22</b>	
<b>Thông số</b>	<b>Giá trị</b>
Phạm vi đo nhiệt độ	-40 đến 80°C ( $\pm 0.5^{\circ}\text{C}$ )
Phạm vi đo độ ẩm	0-100% ( $\pm 2-5\%$ )
Giao tiếp	Single-wire digital (DATA)
Nguồn điện	3.3V hoặc 5V
Tiêu thụ điện năng	$\sim 2.5\text{mA}$
Tần suất lấy mẫu	0.5 Hz (mỗi 2 giây)