

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**  
**KHOA ĐIỆN – ĐIỆN TỬ**  
**BỘ MÔN ĐIỆN TỬ**

-----o0o-----



**BÁO CÁO BÀI TẬP LỚN ÔN THIẾT KẾ HỆ THỐNG NHÚNG**  
**(EE3003)**

***ĐỀ TÀI:***

**Đồng hồ kỹ thuật số hiển thị thời gian thực và trạng thái của đèn  
trên LCD**

<b>GVHD</b>	<b>:</b>	<b>ThS. Bùi Quốc Bảo</b>
<b>LỚP</b>	<b>:</b>	<b>L01</b>
<b>NHÓM</b>	<b>:</b>	<b>04</b>
<b>HỌC KỲ</b>	<b>:</b>	<b>242</b>

***TP. HỒ CHÍ MINH, 08 THÁNG 05 NĂM 2025***

## TÓM TẮT ĐỀ TÀI

**Tên đề tài:** Thiết Kế và Thi Công Mạch Đồng Hồ Số Hẹn Giờ Bật Tắt Đèn  
(Hoặc: Hệ Thống Điều Khiển Chiếu Sáng Tự Động Theo Thời Gian Thực)

**Mục tiêu đề tài:** Xây dựng một hệ thống điện tử có khả năng tự động điều khiển việc bật/tắt một thiết bị điện (cụ thể là đèn) dựa trên thời gian thực được cài đặt trước. Hệ thống cũng hiển thị thời gian hiện tại cho người dùng.

**Mô tả tóm tắt hệ thống:** Đề tài tập trung vào việc thiết kế và triển khai một mạch điện tử hoàn chỉnh bao gồm các khối chức năng chính:

1. **Khối Vi điều khiển Trung tâm (ATmega328P):** Là "bộ não" của hệ thống, chịu trách nhiệm nhận dữ liệu thời gian, xử lý logic điều khiển và giao tiếp với các khối khác.
2. **Khối Thời gian thực (Module DS3231):** Cung cấp thông tin thời gian (giờ, phút, giây, ngày, tháng, năm) một cách chính xác và liên tục, có pin dự phòng để duy trì thời gian khi mất nguồn chính. Giao tiếp với vi điều khiển qua chuẩn I2C.
3. **Khối Hiển thị (LCD 1602):** Hiển thị thời gian hiện tại và có thể cả trạng thái hoạt động của hệ thống cho người dùng theo dõi.
4. **Khối Module Relay :** Nhận tín hiệu điều khiển từ vi điều khiển để đóng/ngắt relay, qua đó bật hoặc tắt đèn được kết nối với relay.
5. **Khối Nguồn:** Cung cấp nguồn điện ổn định (5V) cho toàn bộ hệ thống hoạt động, được chuyển đổi từ nguồn đầu vào (ví dụ: pin 9V hoặc adapter) thông qua IC ổn áp LM7805.

**Nguyên lý hoạt động chính:** Vi điều khiển ATmega328P liên tục đọc thông tin thời gian từ module DS1307. Dựa trên chương trình đã được nạp, vi điều khiển sẽ so sánh thời gian hiện tại với thời gian bật/tắt đèn đã được thiết lập trước. Khi đến thời điểm thích hợp, vi điều khiển sẽ gửi tín hiệu đến khối relay để thay đổi trạng thái của đèn (bật hoặc tắt). Thời gian hiện tại được hiển thị liên tục trên màn hình LCD.

## MỤC LỤC

<b>I. Cơ sở lý thuyết</b> .....	4
1. Các linh kiện .....	4
1.1 <i>Arduino Uno R3</i> .....	4
1.2 <i>Atmega328P</i> .....	5
1.3 <i>LCD 1602 và giao tiếp I2C</i> .....	6
1.4 <i>Relay 1 kênh 5V</i> .....	8
1.5 <i>Module thời gian thực DS3231</i> .....	9
1.6 <i>Điện trở và đèn LED</i> .....	11
<b>II. Thiết kế sơ đồ khối và sơ đồ nguyên lý và lập trình</b> .....	14
1. Thiết kế: .....	14
2. Sơ đồ nguyên lý ( Schematic ) .....	16
3. Lập trình .....	20
<b>III. Kết quả thực hiện:</b> .....	23
1. PCB và 3D: .....	23
2. Mạch thực tế (thành phẩm): .....	25

## I. Cơ sở lý thuyết

### 1. Các linh kiện

#### 1.1 Arduino Uno R3

Giới thiệu về Arduino R3:

Arduino Uno R3 là một bảng mạch vi điều khiển phổ biến và dễ sử dụng, rất lý tưởng cho cả người mới bắt đầu lẫn các nhà phát triển có kinh nghiệm trong lĩnh vực điện tử và lập trình nhúng.

Đặc điểm nổi bật:

- Vi điều khiển: Arduino Uno R3 sử dụng ATmega328P, một vi điều khiển 8-bit mạnh mẽ. Nó có:
  - 32KB Flash: để lưu trữ chương trình.
  - 2KB SRAM: bộ nhớ tạm thời cho dữ liệu khi chạy.
  - 1KB EEPROM: bộ nhớ không bị xóa khi mất điện.
- Chân I/O:
  - 14 chân kỹ thuật số: trong đó 6 chân hỗ trợ PWM (điều chế độ rộng xung) để điều khiển tín hiệu như độ sáng LED hoặc tốc độ động cơ.
  - 6 chân analog: dùng để đọc tín hiệu tương tự từ cảm biến.
- Giao tiếp: Hỗ trợ các giao thức:
  - UART: giao tiếp nối tiếp.
  - SPI: giao tiếp tốc độ cao.
  - I2C: kết nối với nhiều thiết bị ngoại vi.
- Nguồn điện: Có thể cấp nguồn qua:
  - USB: kết nối với máy tính.
  - Nguồn ngoài: từ 7-12V qua jack nguồn.

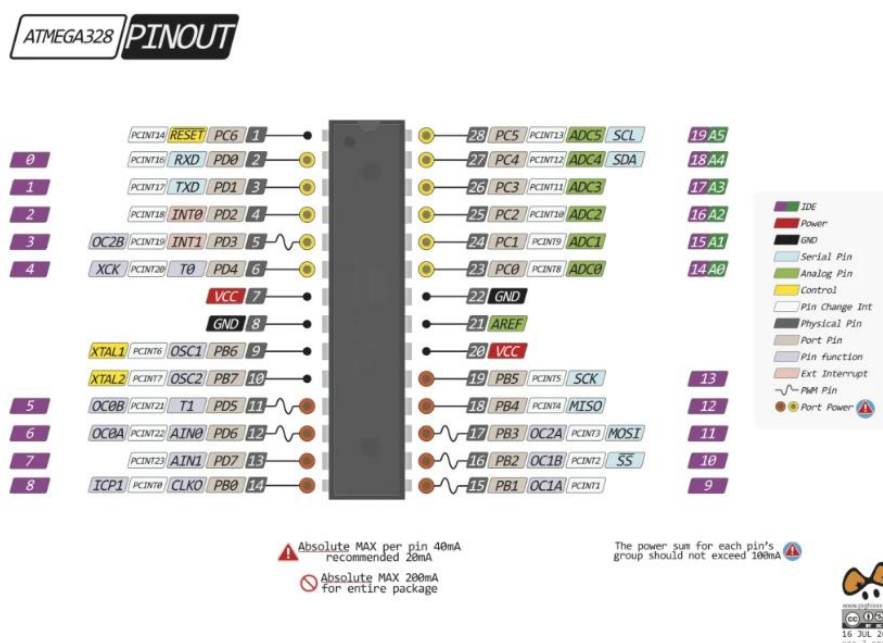


- 23 chân đầu vào/đầu ra (I/O).
- 6 chân PWM (điều chế độ rộng xung).
- 6 chân ADC (chuyển đổi tín hiệu tương tự sang số).
- Điện áp hoạt động: Từ 1.8V đến 5.5V.
- Tần số: Lên đến 20MHz.

Ứng dụng:

ATmega328P được sử dụng rộng rãi trong các dự án điện tử như:

- Điều khiển thiết bị thông minh.
- Cảm biến và hệ thống IoT.



ATMEGA328P Pinout (28-pin PDIP)

- Các dự án DIY dựa trên Arduino.

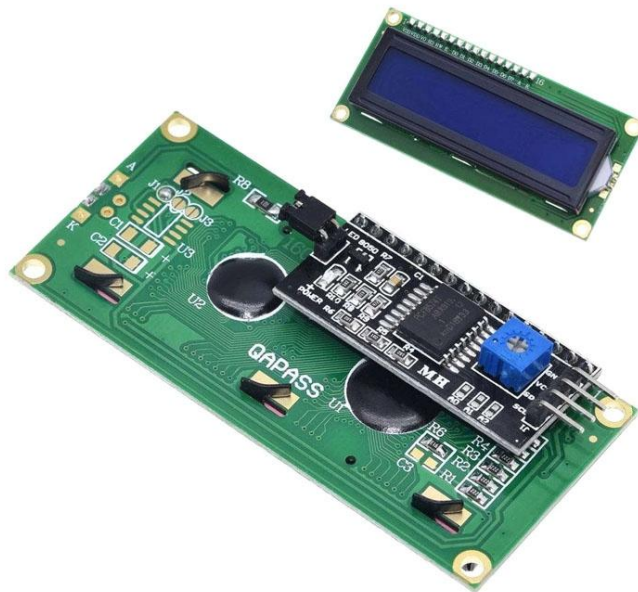
### 1.3 LCD 1602 và giao tiếp I2C

Giới thiệu về LCD1602:

- Màn hình LCD thường được tích hợp vào nhiều loại thiết bị như hệ thống

nhúng, các thiết bị kiểm soát, máy đo và ứng dụng khác để hiển thị thông tin và dữ liệu. Các mẫu màn hình LCD phổ biến như 16x02 và 20x04 thường được kết hợp với bộ điều khiển riêng biệt, giúp tương tác với vi điều khiển hoặc bo mạch chính.

- Sử dụng giao thức I2C (Inter-Integrated Circuit) giúp giảm số lượng chân kết nối cần thiết, làm cho việc quản lý màn hình LCD trở nên thuận tiện hơn. Thay vì cần sử dụng nhiều chân để điều khiển màn hình LCD, chỉ cần hai dây (SDA và SCL) để giao tiếp I2C. Điều này mang lại lợi ích đáng kể cho việc thiết kế và kết nối, giúp tối ưu hóa không gian và tăng tính linh hoạt trong ứng dụng



Hình ảnh về màn hình LCD đính kèm module I2C

Thông số kỹ thuật:

- Điện áp hoạt động: 5VDC
- Dòng điện tiêu thụ: 350uA - 600uA.

- Nhiệt độ hoạt động: -30°C đến 75°C.
- Kích thước 96 x 60 mm, chữ đen, nền xanh lá.
- Đèn Led nên có thể điều khiển bằng biến trở hoặc PWM.
- Có thể điều khiển bằng 6 chân tín hiệu.
- Hỗ trợ hiển thị bộ kí tự tiếng Anh và tiếng Nhật.

#### *1.4 Relay 1 kênh 5V*

Giới thiệu về Relay:

Module Relay 1 Kênh của Bạch Tử 360 là một thiết bị điều khiển điện tử nhỏ gọn và linh hoạt, được thiết kế để bật/tắt các thiết bị điện áp cao (lên đến 220VAC, dòng tối đa 10A) bằng tín hiệu điện áp thấp (5V) từ vi điều khiển như Arduino hoặc Raspberry Pi. Với thành phần chính là relay SONGLE SRD-05VDC-SL-C, module này mang đến khả năng chịu tải mạnh mẽ và độ tin cậy cao, phù hợp cho nhiều ứng dụng trong tự động hóa và IoT.

Đặc điểm nổi bật:

- Relay SONGLE SRD-05VDC-SL-C:

Hoạt động với điện áp điều khiển 5VDC, relay này có khả năng đóng/ngắt mạch điện với thông số kỹ thuật ấn tượng:

- Chịu tải tối đa 10A ở 250VAC.
- Chịu tải tối đa 10A ở 30VDC.
- Hỗ trợ hai chế độ kích hoạt:
  - High-level trigger (kích hoạt mức cao): Relay bật khi tín hiệu đầu vào là 5V.
  - Low-level trigger (kích hoạt mức thấp): Relay bật khi tín hiệu đầu vào là 0V.

Chế độ có thể được chuyển đổi dễ dàng thông qua nút nhảy (jumper) trên mạch.

- Thiết kế nhỏ gọn:



- Bảng mạch PCB màu đỏ, kết hợp với hai dây đầu nối màu xanh dương, giúp dễ dàng nhận diện và tích hợp vào các dự án.
- Một dây đầu nối (VCC, GND, IN) dùng để cấp nguồn và nhận tín hiệu điều khiển.
- Dây còn lại (COM, NO, NC) dùng để kết nối với tải điện như đèn, động cơ, hoặc thiết bị gia dụng.

- Đèn LED báo hiệu:

Được tích hợp trên mạch để hiển thị trạng thái hoạt động của relay (bật/tắt), giúp người dùng dễ dàng theo dõi.

Ứng dụng thực tế:

Module này là lựa chọn lý tưởng cho các dự án điện tử và tự động hóa, bao gồm:

- Điều khiển đèn chiếu sáng trong hệ thống nhà thông minh.
- Bật/tắt động cơ hoặc quạt điện.
- Kiểm soát các thiết bị gia dụng sử dụng điện áp 220VAC thông qua vi điều khiển.



Hình ảnh về Relay

### 1.5 Module thời gian thực DS3231

Giới thiệu về module DS3231:

Module DS3231 là một mô-đun đồng hồ thời gian thực (Real-Time Clock - RTC) được sử dụng phổ biến trong các dự án điện tử để cung cấp thời gian và ngày tháng

chính xác. Nó tích hợp IC DS3231, nổi bật với độ chính xác cao và khả năng duy trì thời gian ngay cả khi mất nguồn nhờ pin dự phòng. Module này thường được kết nối với các vi điều khiển như Arduino hoặc Raspberry Pi thông qua giao tiếp I2C.

#### Đặc điểm chính:

- Độ chính xác cao: Sai số chỉ khoảng  $\pm 2$  phút mỗi năm nhờ bộ dao động tinh thể tích hợp và khả năng tự điều chỉnh theo nhiệt độ.
- Pin dự phòng: Sử dụng pin CR2032 để giữ thời gian và dữ liệu khi mất nguồn chính.
- Giao tiếp I2C: Kết nối dễ dàng với vi điều khiển qua hai chân SDA và SCL.
- Chức năng bổ sung:
  - Hai báo thức (Alarm) có thể cài đặt độc lập.
  - Xuất tín hiệu xung vuông (Square wave) với tần số tùy chỉnh (1Hz, 4kHz, 8kHz, 32kHz).
  - Cảm biến nhiệt độ tích hợp để theo dõi nhiệt độ môi trường.
- Nguồn điện: Hoạt động với nguồn 3.3V hoặc 5V, tương thích với nhiều nền tảng.

#### Ứng dụng thực tế:

Module DS3231 phù hợp cho nhiều dự án yêu cầu thời gian chính xác, chẳng hạn:

- Đồng hồ kỹ thuật số: Hiển thị thời gian trên màn hình LCD hoặc LED.
- Hệ thống ghi dữ liệu: Gắn dấu thời gian cho dữ liệu từ cảm biến.
- Tự động hóa: Điều khiển thiết bị theo lịch trình (ví dụ: bật/tắt đèn, tưới cây).
- Dự án IoT: Đồng bộ thời gian cho các thiết bị kết nối internet.

#### Lưu ý khi sử dụng:

- Pin CR2032: Đảm bảo pin còn tốt để duy trì thời gian khi mất nguồn.

- Địa chỉ I2C: Kiểm tra xung đột nếu dùng nhiều thiết bị I2C.
- Độ chính xác: Sai số rất nhỏ, nhưng có thể điều chỉnh nếu cần độ chính xác tuyệt đối.



Hình ảnh về Module Dule DS3231

## 1.6 Điện trở và đèn LED

### 1.6.1 Điện trở

Điện trở là một linh kiện thụ động có khả năng hạn chế dòng điện chạy qua nó. Nó được sử dụng rộng rãi trong các mạch điện để điều chỉnh dòng điện, chia điện áp, hoặc bảo vệ các linh kiện khác.

Cấu tạo và nguyên lý hoạt động:

- Cấu tạo: Điện trở thường được làm từ các vật liệu như carbon, kim loại, hoặc hợp kim. Nó có hai chân để kết nối vào mạch và giá trị điện trở được đo bằng đơn vị ohm ( $\Omega$ ).
- Nguyên lý hoạt động: Dựa trên định luật Ohm, dòng điện ( $I$ ) chạy qua điện trở tỷ lệ thuận với điện áp ( $V$ ) đặt lên nó và tỷ lệ nghịch với giá trị điện trở ( $R$ ), theo công thức:  $I = \frac{V}{R}$   $I = \frac{V}{R}$ . Điều này cho phép điện trở kiểm soát dòng điện hoặc điện áp trong mạch.

### Ứng dụng:

- Hạn chế dòng điện: Bảo vệ các linh kiện như LED khỏi dòng điện quá lớn.
- Chia điện áp: Tạo ra các mức điện áp khác nhau trong mạch.
- Mạch lọc: Loại bỏ nhiễu hoặc điều chỉnh tín hiệu trong các mạch analog.
- Thông số kỹ thuật
- Giá trị điện trở: Đo bằng ohm ( $\Omega$ ), có thể từ vài ohm đến hàng megaohm.
- Công suất chịu đựng: Đo bằng watt (W), cho biết khả năng chịu nhiệt của điện trở.
- Độ chính xác: Thường là  $\pm 5\%$  hoặc  $\pm 1\%$ , cho biết sai số của giá trị điện trở.



### Hình ảnh về điện trở

#### 1.6.2 Đèn LED

#### Giới thiệu về đèn LED :

Đèn LED (Light Emitting Diode) là một loại diode phát quang, chuyển đổi điện năng thành ánh sáng khi có dòng điện chạy qua. Nó được sử dụng phổ biến trong các ứng dụng hiển thị và chiếu sáng.

#### Cấu tạo và nguyên lý hoạt động:

Cấu tạo: Đèn LED được làm từ các vật liệu bán dẫn như gallium arsenide (GaAs) hoặc gallium phosphide (GaP). Nó có hai chân: anode (dương) và cathode (âm).

Nguyên lý hoạt động: Khi có dòng điện thuận chạy qua, các electron và lỗ trống trong vật liệu bán dẫn kết hợp, giải phóng năng lượng dưới dạng ánh sáng. Màu sắc của ánh sáng phụ thuộc vào loại vật liệu bán dẫn sử dụng (ví dụ: đỏ, xanh lá, xanh dương).

Ứng dụng:

- Hiển thị trạng thái: Dùng trong các thiết bị để báo hiệu (bật/tắt, cảnh báo).
- Chiếu sáng: Trong đèn pin, đèn trang trí, hoặc màn hình.
- Ứng dụng quang học: Trong cảm biến quang, truyền dữ liệu qua ánh sáng.
- Thông số kỹ thuật
- Màu sắc: Phụ thuộc vào vật liệu bán dẫn.
- Độ sáng: Đo bằng millicandela (mcd), cho biết cường độ ánh sáng.
- Điện áp thuận: Thường từ 1.8V đến 3.3V, tùy màu sắc.
- Dòng điện tối đa: Thường là 20mA cho LED thông dụng.

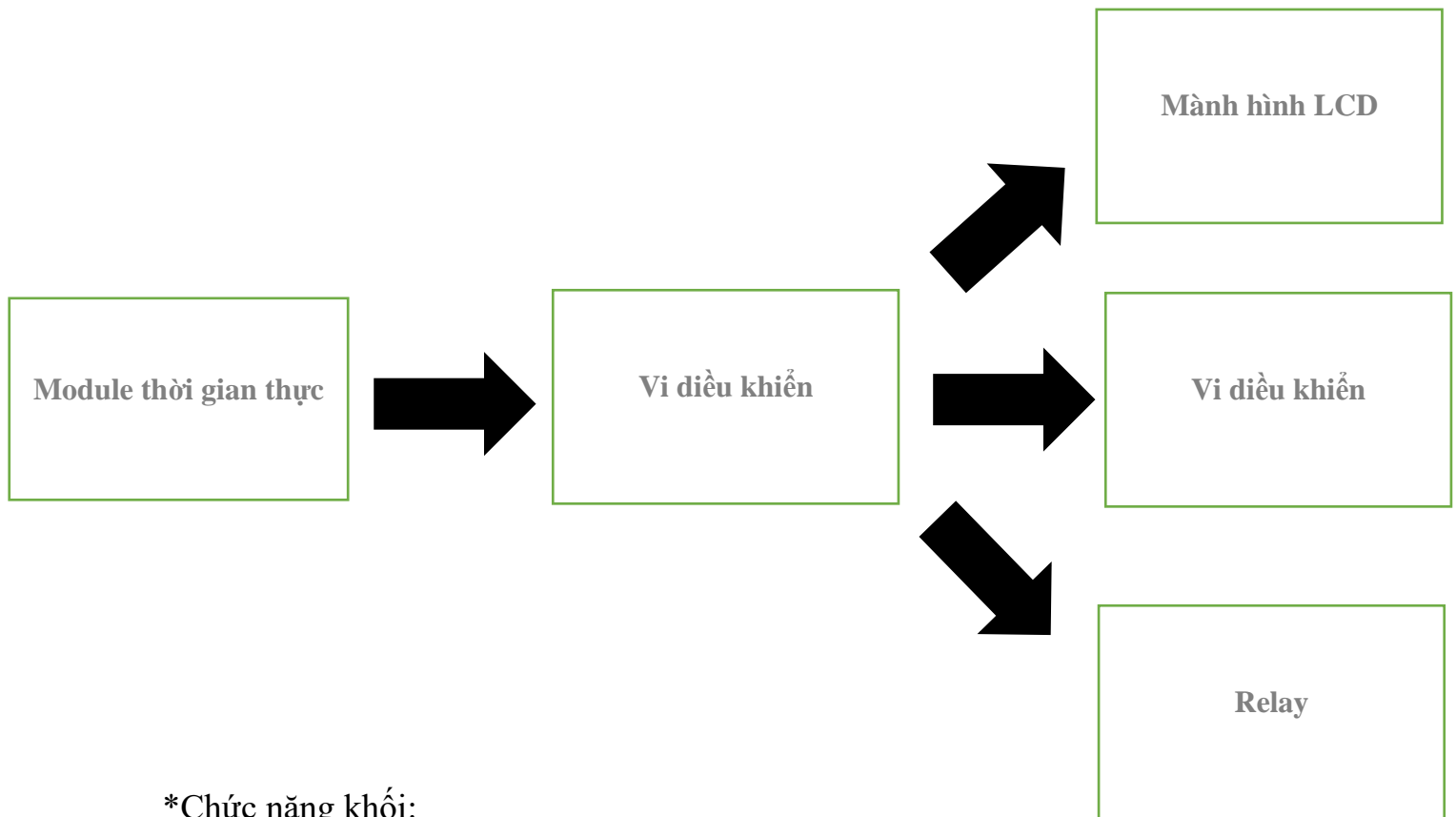
## II. Thiết kế sơ đồ khối và sơ đồ nguyên lý và lập trình

### 1. Thiết kế:

\*Yêu cầu thiết kế:

- Hiển thị được thời gian thực.
- Relay bật tắt đèn theo thời gian

\*Sơ đồ khối chi tiết:



\*Chức năng khối:

➤ Module thời gian thực (Real-time clock module):

- Chức năng: Cung cấp thông tin thời gian (giờ, phút, giây, ngày, tháng, năm) một cách chính xác và liên tục, ngay cả khi hệ thống bị mất nguồn tạm thời (thường nhờ có pin dự phòng riêng). Dữ liệu thời gian này sẽ được gửi đến vi điều khiển.

➤ Vi điều khiển (Microcontroller) - Khối trung tâm:

- Chức năng: Đây là "bộ não" của hệ thống.
- Nhận dữ liệu thời gian từ "Module thời gian thực".

- Xử lý thông tin theo logic đã được lập trình.
- Điều khiển "Màn hình LCD" để hiển thị thời gian thực (theo yêu cầu thiết kế "Hiển thị được thời gian thực").
- Điều khiển "Relay" để bật/tắt đèn theo thời gian đã được cài đặt (theo yêu cầu thiết kế "Relay bật tắt đèn theo thời gian").

➤ *Lưu ý:* Trong sơ đồ có hai khối "Vi điều khiển". Nếu đây là một hệ thống đơn giản, rất có thể khối "Vi điều khiển" thứ hai ở bên phải (nhận tín hiệu từ khối "Vi điều khiển" trung tâm và xuất ra "Relay") là một phần của khối "Vi điều khiển" trung tâm, hoặc là một vi điều khiển phụ trợ nếu có các tác vụ phức tạp hơn cần phân chia. Tuy nhiên, dựa trên sơ đồ đơn giản này, chúng ta có thể hiểu rằng có một đơn vị xử lý trung tâm.

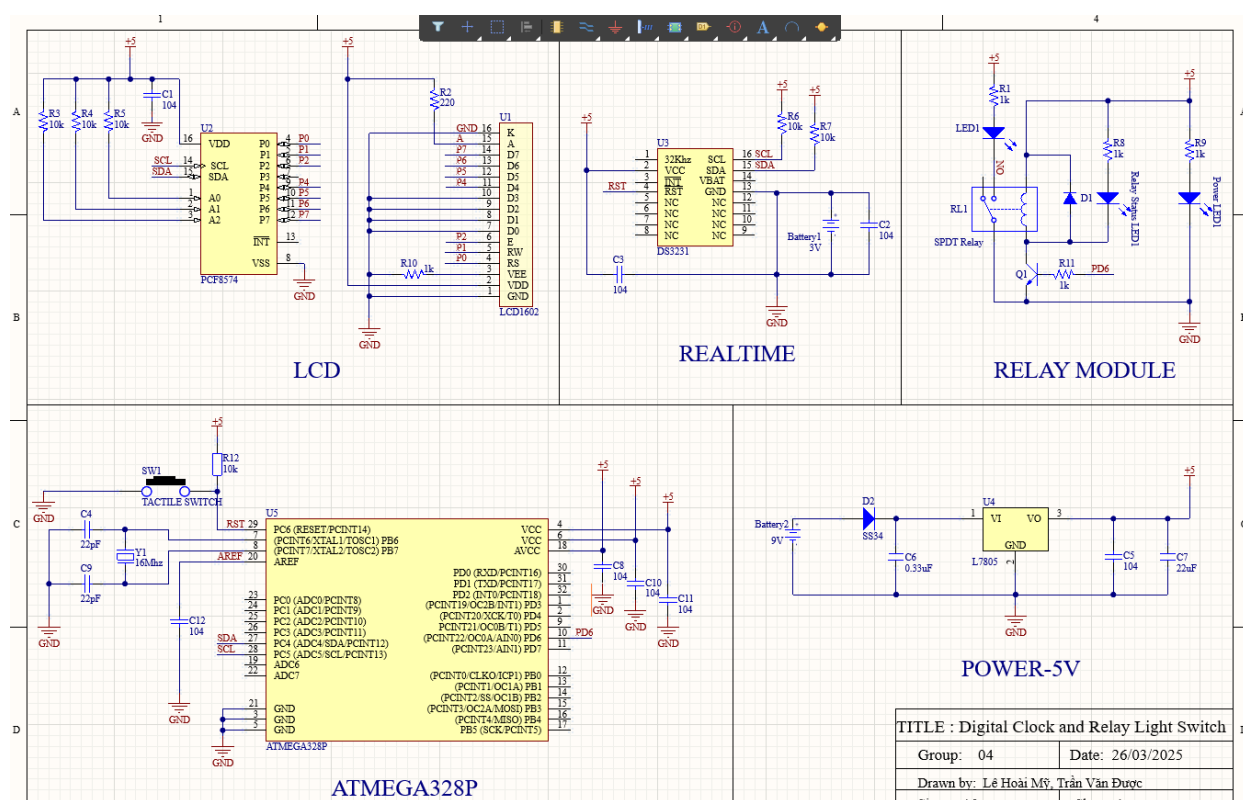
➤ **Màn hình LCD (Liquid Crystal Display):**

Chức năng: Hiển thị thông tin cho người dùng. Trong trường hợp này, nó được sử dụng để hiển thị thời gian thực nhận được từ vi điều khiển.

➤ **Relay:**

Chức năng: Là một công tắc điện tử được điều khiển bởi vi điều khiển. Nó cho phép một tín hiệu điện áp thấp từ vi điều khiển có thể đóng/ngắt một mạch điện có điện áp hoặc dòng điện cao hơn nhiều (ví dụ như mạch đèn). Trong thiết kế này, relay sẽ bật hoặc tắt đèn dựa trên tín hiệu điều khiển theo thời gian từ vi điều khiển.

## 2. Sơ đồ nguyên lý ( Schematic )



Nhìn tổng thể, sơ đồ này mô tả một hệ thống điều khiển đèn dựa trên thời gian thực, có hiển thị LCD, sử dụng vi điều khiển ATmega328P làm trung tâm xử lý. Các khối chính bao gồm:

### ***KHỐI VI ĐIỀU KHIỂN (ATMEGA328P):***

Linh kiện chính: IC U3 - ATmega328P. Đây là "bộ não" của toàn bộ hệ thống.

Chức năng:

Đọc dữ liệu thời gian từ khối Realtime (DS1307).

Xử lý logic chương trình để quyết định khi nào bật/tắt relay.

Giao tiếp và hiển thị thông tin (thời gian, trạng thái) lên khối LCD.

Nhận tín hiệu từ nút nhấn (SW1) để có thể cài đặt hoặc điều khiển thủ công (tùy theo lập trình).

Các kết nối quan trọng:



Cụm thạch anh (Y1 - 16MHz, C8, C9 - 22pF): Tạo dao động xung nhịp cho vi điều khiển hoạt động.

Chân RESET (PC6/RESET): Kết nối với tụ C4 và điện trở R5 tạo thành mạch reset thủ công qua nút nhấn SW1 (TACTILE SWITCH).

Chân AREF: Kết nối với tụ C12, là chân điện áp tham chiếu cho bộ ADC (Analog-to-Digital Converter), thường được nối với VCC qua tụ lọc nếu không dùng điện áp tham chiếu ngoài.

VCC và GND: Các chân cấp nguồn cho IC. Tụ C10, C11 (104 - tức 0.1uF) là các tụ lọc nhiễu cho nguồn cấp.

Các chân I/O (Input/Output):

Kết nối với LCD (P0-P7, RS, RW, E).

Kết nối với Module Realtime qua giao thức I2C (SDA - PC4, SCL - PC5).

Kết nối với Relay Module (PD6).

Kết nối với nút nhấn (PC6/RESET).

### ***KHÔI HIỂN THỊ LCD (LCD):***

Linh kiện chính: LCD1602 (U1 - PCF8574 và bản thân module LCD). PCF8574 là IC chuyển đổi giao tiếp I2C sang giao tiếp song song 8-bit cho LCD, giúp tiết kiệm chân cho vi điều khiển. Tuy nhiên, nhìn vào cách kết nối trực tiếp P0-P7, RS, RW, E từ ATmega328P đến khối LCD (U1 được ký hiệu là LCD1602), có vẻ như đây là kết nối trực tiếp kiểu song song 4-bit hoặc 8-bit chứ không qua I2C expander PCF8574 như thường thấy với ký hiệu U1. *Nếu U1 là LCD1602 module thông thường thì các chân P0-P7 sẽ tương ứng với D0-D7 của LCD.*

Chức năng: Hiển thị thông tin như thời gian hiện tại, trạng thái bật/tắt của đèn, hoặc các thông báo cài đặt.

Các kết nối:

RS, RW, E: Các chân điều khiển của LCD.

D0-D7 (kết nối với P0-P7 của ATmega328P): Các chân dữ liệu. Thông thường LCD1602 sẽ dùng 4 chân dữ liệu (D4-D7) hoặc 8 chân (D0-D7). Sơ đồ đang vẽ kết nối cả 8 chân.

A, K: Chân cấp nguồn cho đèn nền LCD (Backlight). Điện trở R2 (220 Ohm) dùng để hạn dòng cho đèn nền.

VSS, VDD, V0: Chân nguồn và chân điều chỉnh độ tương phản (thường nối với biến trở, ở đây có thể được điều chỉnh ngàm hoặc thiếu trong sơ đồ chi tiết của module LCD).

### ***KHỐI THỜI GIAN THỰC (REALTIME):***

Linh kiện chính: U2 - DS1307. Đây là IC thời gian thực.

Chức năng: Duy trì và cung cấp thông tin thời gian (giờ, phút, giây, ngày, tháng, năm) một cách chính xác, ngay cả khi hệ thống mất nguồn chính, nhờ vào pin dự phòng (Battery1 - 3V).

Các kết nối:

SCL, SDA: Giao tiếp với vi điều khiển ATmega328P qua chuẩn I2C. Điện trở kéo lên R3, R4 (10k Ohm) là cần thiết cho bus I2C.

X1, X2: Kết nối với thạch anh 32.768 kHz (thường được tích hợp sẵn trong module DS1307 hoặc cần gắn ngoài).

VBAT: Nối với pin dự phòng 3V (Battery1).

VCC, GND: Chân cấp nguồn cho IC. Tụ C2 (104) là tụ lọc nhiễu.

SQW/OUT: Chân có thể xuất ra tín hiệu xung vuông, thường không dùng trong ứng dụng cơ bản.

### ***KHỐI RELAY (RELAY MODULE):***

Linh kiện chính: RL1 (SPDT Relay), Transistor Q1 (ví dụ NPN như BC547, 2N2222), Diode D1 (ví dụ 1N4007), LED1 (báo trạng thái), các điện trở R7, R8, R11.

Chức năng: Đóng/ngắt một thiết bị có công suất lớn hơn (ví dụ: đèn 220V) bằng tín hiệu điều khiển mức thấp từ vi điều khiển.

Hoạt động:

Khi vi điều khiển xuất tín hiệu mức CAO ra chân PD6, dòng điện qua R11 làm transistor Q1 dẫn.

Khi Q1 dẫn, cuộn dây của relay RL1 được cấp điện, tạo ra từ trường hút tiếp điểm của relay, làm đóng hoặc mở mạch điện của tải (đèn).

Diode D1 (Flyback diode) được mắc song song ngược với cuộn dây relay để dập tắt dòng điện cảm ứng sinh ra khi relay ngắt, bảo vệ transistor Q1.

LED1 (Power LED trên sơ đồ, nhưng có vẻ là LED báo trạng thái relay) cùng với điện trở hạn dòng R7 sáng lên khi relay được kích hoạt (nếu mắc song song với cuộn dây relay) hoặc khi có nguồn cấp cho module (nếu là Power LED của module). Nhìn cách mắc, LED1 và R7 có vẻ như báo hiệu relay đang được cấp nguồn để hoạt động (khi PD6 ở mức cao).

R8 và LED2: Là một LED báo nguồn (Power LED) cho toàn mạch hoặc cho khối này, sáng khi có nguồn +5V.

### ***KHỐI NGUỒN 5V (POWER-5V):***

Linh kiện chính: U4 - LM7805 (ổn áp tuyến tính 5V), Diode D2 (SB34 - Diode Schottky, có thể là diode chống ngược cực hoặc chỉnh lưu nếu nguồn vào là AC), các tụ lọc C5, C6, C7.

Chức năng: Chuyển đổi điện áp đầu vào (từ Battery2 - 9V) thành điện áp ổn định +5V để cấp cho toàn bộ hệ thống (vi điều khiển, LCD, DS1307, cuộn dây relay).

Hoạt động:

Điện áp từ nguồn Battery2 (9V) đi qua diode D2 (giảm sụt áp khoảng 0.2-0.4V nếu là Schottky).

Tụ C6 (0.33uF hoặc 100nF) lọc nhiễu đầu vào cho LM7805.

LM7805 (U4) ổn định điện áp đầu ra ở mức +5V.

Tụ C5 (10uF) và C7 (104 - 0.1uF) lọc nhiễu và tăng độ ổn định cho điện áp đầu ra +5V.

Lưu ý: "Battery2 9V" là nguồn cấp chính cho mạch.

Tóm lại:

Vi điều khiển ATmega328P là trung tâm, nhận thời gian từ DS1307 qua I2C.

Dựa vào thời gian và chương trình nạp sẵn, ATmega328P điều khiển relay RL1 (thông qua transistor Q1) để bật/tắt đèn.

Thông tin thời gian và trạng thái được hiển thị trên LCD1602.

Toàn bộ mạch được cấp nguồn +5V ổn định từ khối nguồn dùng LM7805, với đầu vào là pin 9V.

### 3. Lập trình

Để có thể khởi chạy hệ thống thì ta cần nạp code vào kit Arduino Uno, sau khi nạp code vào thì hoặc là cấp nguồn từ laptop hoặc là cấp nguồn riêng cho Arduino từ pin, đoạn code như sau:

```
#include <DS3231.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2); //I2C của LCD là 0x27, LCD 16x02

int Relay = 6;

DS3231 rtc(SDA, SCL);
Time t;

int giomo = 13;
int phutmo = 00;
int giaymo[] = {00, 10, 20, 30, 40, 50};

int giotat = 13;
int phuttat = 10;
int giaytat[] = {05, 15, 25, 35, 45, 55};

void setup() {
  Serial.begin(115200);
  Wire.begin();//Khởi tạo I2C

  lcd.backlight(); //Mở đèn LCD
  lcd.init(); //Khởi động LCD
  lcd.setCursor(0,0);
  lcd.print("Time:");
  lcd.setCursor(0,1);
  lcd.print("Light Mode: ");
  lcd.setCursor(12,1);
  lcd.print("OFF");

  rtc.begin();
  rtc.setDOW(TUESDAY); // Cài đặt thứ trong tuần
  rtc.setTime(12, 59, 55); // Cài đặt giờ, phút, giây (24)
  rtc.setDate(6, 5, 2025); // Ngày, tháng, năm

  pinMode(Relay, OUTPUT);
  digitalWrite(Relay, LOW);
```

```

}

void loop()
{
    t = rtc.getTime();
    // Serial.print(rtc.getDOWStr());
    // Serial.print(" ");

    // Serial.print(rtc.getDateStr());
    // Serial.print(" -- ");
    // lcd.setCursor(6,1);
    // lcd.print(rtc.getDateStr());

    // Serial.println(rtc.getTimeStr());

    lcd.setCursor(6,0);
    lcd.print(rtc.getTimeStr());

    delay (200);

    if(t.hour == giomo && t.min == phutmo)
    {

        bool found_giaymo = false;
        int arraySize_giaymo = sizeof(giaymo) / sizeof(giaymo[0]);
        for (int i = 0; i < arraySize_giaymo; i++) {
            if (t.sec == giaymo[i]) {
                found_giaymo = true;
                break;
            }
        }

        bool found_giaytat = false;
        int arraySize_giaytat = sizeof(giaytat) / sizeof(giaytat[0]);
        for (int i = 0; i < arraySize_giaymo; i++) {
            if (t.sec == giaytat[i]) {
                found_giaytat = true;
                break;
            }
        }

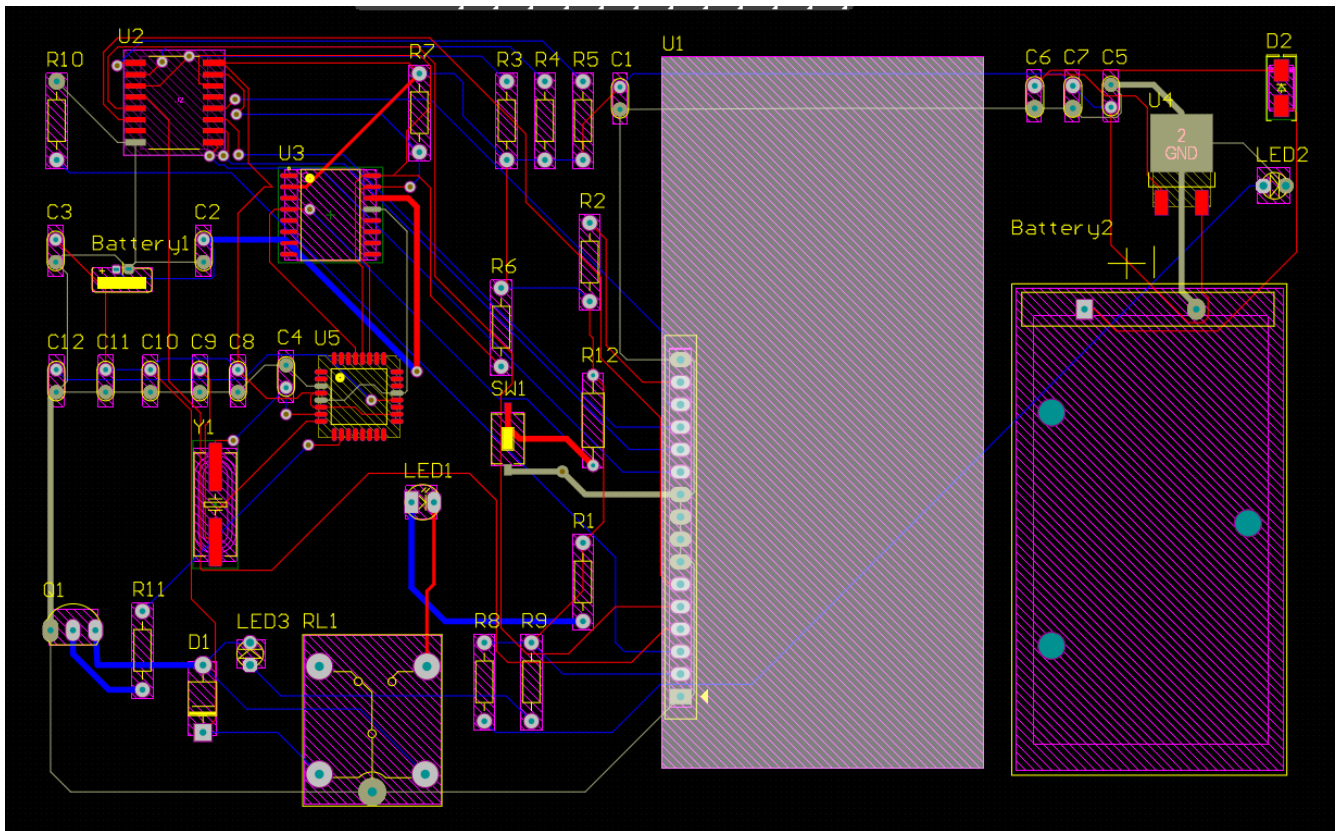
        if (found_giaymo) {
            digitalWrite(Relay,HIGH);
            lcd.setCursor(0,1);
            lcd.print("Light Mode: ");
            lcd.setCursor(12,1);

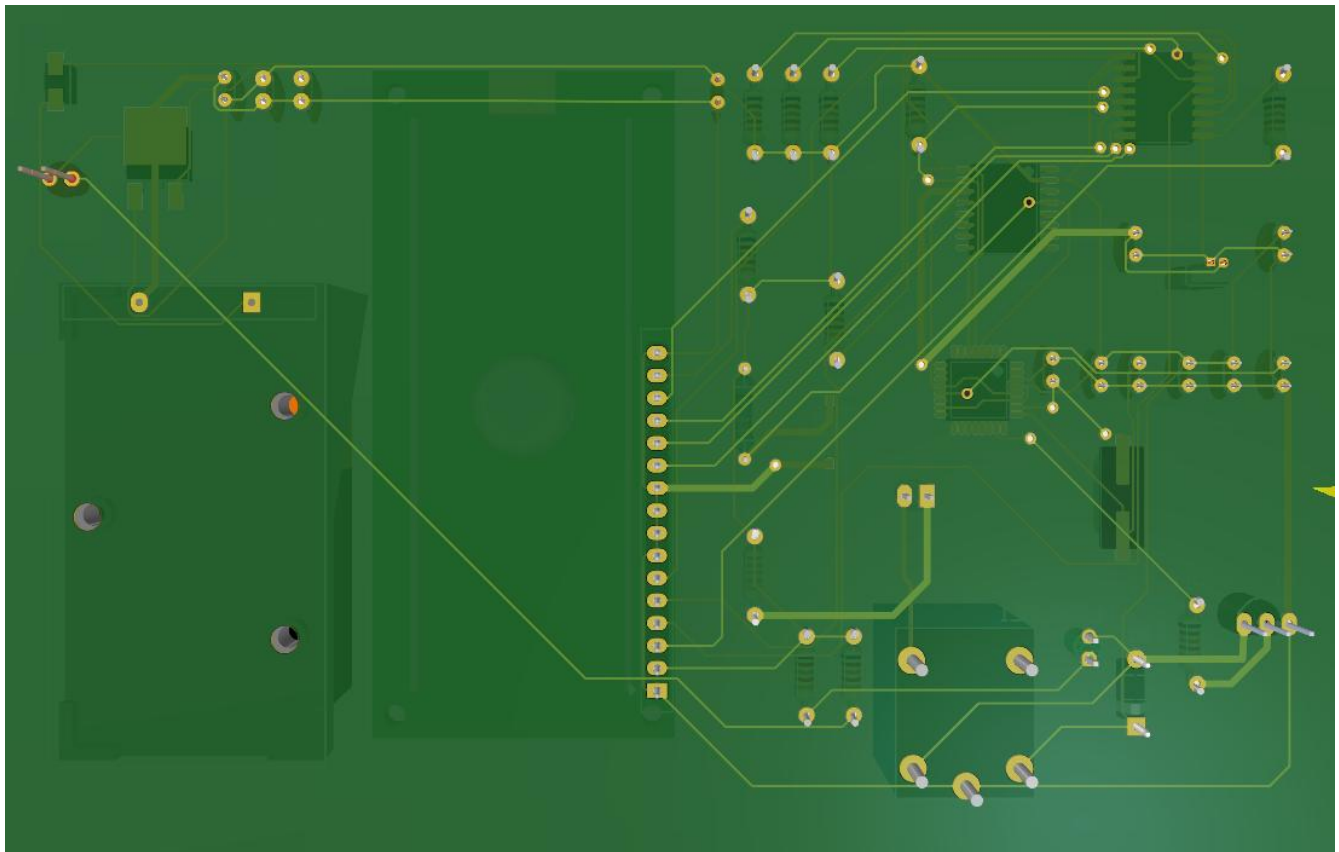
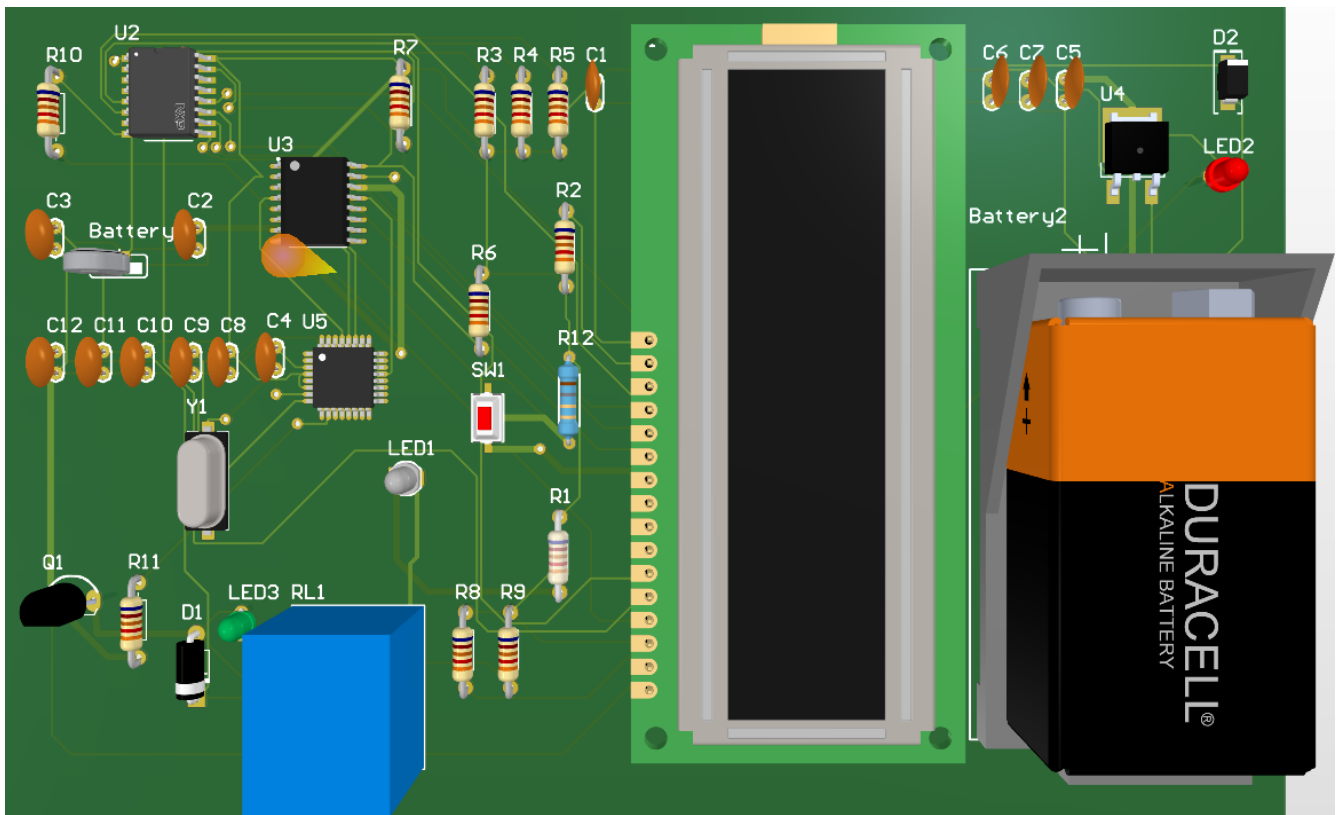
```

```
        lcd.print("ON");  
        lcd.setCursor(14,1);  
        lcd.print(" ");  
    } else if (found_giaytat) {  
        digitalWrite(Relay,LOW);  
        lcd.setCursor(0,1);  
        lcd.print("Light Mode: ");  
        lcd.setCursor(12,1);  
        lcd.print("OFF");  
    }  
}  
}
```

### III. Kết quả thực hiện:

#### 1. PCB và 3D:







## 2. Mạch thực tế (thành phẩm):

