

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP HCM
KHOA CÔNG NGHỆ THÔNG TIN



Nhập môn lập trình kết nối vạn vật
Smart Lighting System – Final Project proposal

Giảng viên hướng dẫn:

- Nguyễn Đức Hoàng Hạ
- Đỗ Thị Thanh Hà

Lớp: 21KHDL

MSSV: 21127412

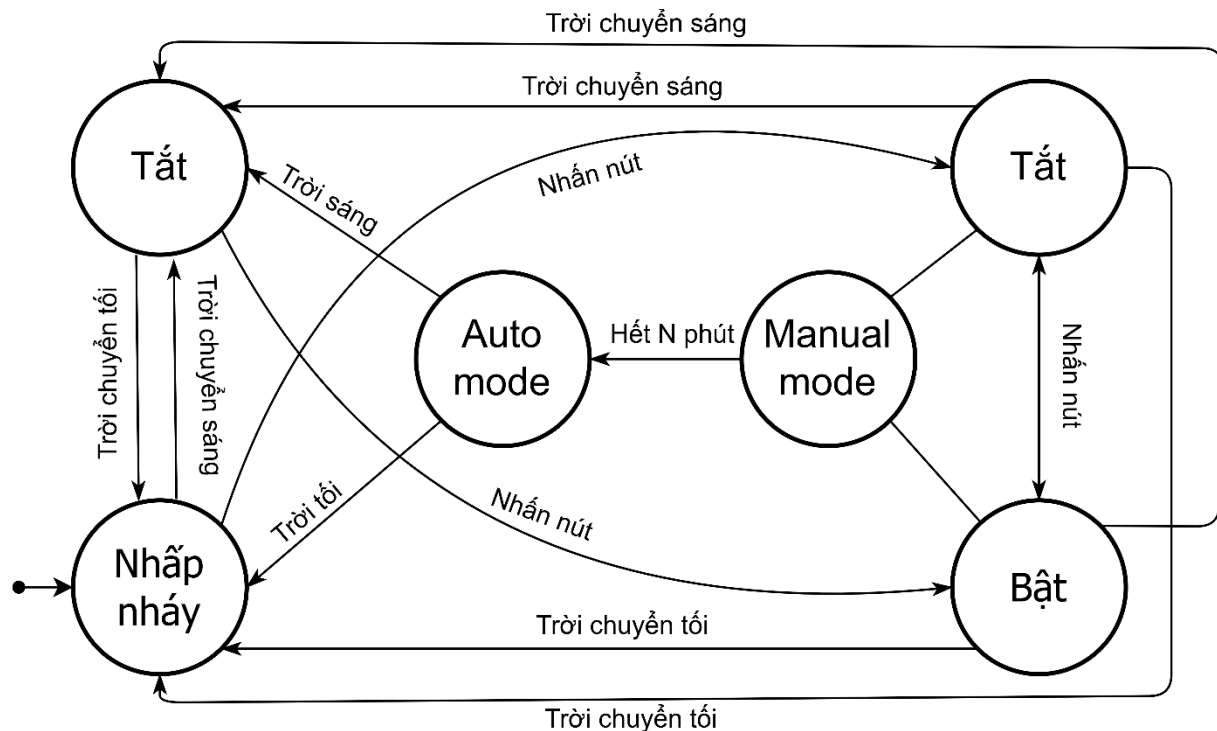
Họ tên: Hồ Bạch Như Quỳnh

NỘI DUNG

<i>I. ĐẶC TẢ YÊU CẦU</i>	1
<i>II. THIẾT KẾ HỆ THỐNG</i>	3
1. Thiết kế tổng quát.....	3
2. Thiết kế chi tiết (đề xuất các giải pháp)	4
<i>III. KIỂM THỬ</i>	7
1. Kiểm tra từng thành phần.....	7
2. Kiểm tra tích hợp	7
<i>IV. TÀI LIỆU THAM KHẢO</i>	7

I. ĐẶC TẢ YÊU CẦU

- Hệ thống gồm một cảm biến ánh sáng, một đèn LED và một nút nhấn
- Hệ thống hoạt động ở hai chế độ: Tự động và Thủ công.
 - Chế độ tự động (auto mode):
 - ✓ Cảm biến ánh sáng dùng để nhận biết ngày hoặc đêm (quy ước giá trị cảm biến < 1000 là ban đêm, còn lại là ban ngày)
 - ✓ Vào ban ngày, đèn sẽ tắt. Vào ban đêm, đèn sẽ nhấp nháy.
 - Chế độ thủ công (manual mode):
 - ✓ Một nút nhấn cho phép bật hoặc tắt đèn theo cách thủ công. Nếu đèn tắt, khi nhấn nút đèn sẽ bật và ngược lại.
 - ✓ Sau N phút hoặc thay đổi chu kỳ ngày/đêm, hệ thống sẽ tự động chuyển về chế độ tự động.
- Hệ thống hỗ trợ giao diện web
 - Hiển thị giá trị đo được từ cảm biến ánh sáng, trạng thái của đèn LED (bật/tắt/nhấp nháy)
 - Nút nhấn để điều khiển bật/tắt đèn LED, chức năng tương tự như trên phần cứng
 - Tích hợp chat GPT, giao diện chatbot cho phép người dùng quản lý hệ thống bằng lệnh thoại (giọng nói) hoặc lệnh văn bản. Đồng thời lưu lại lịch sử trò chuyện có dấu thời gian. *(bổ sung)*
- Hệ thống phải có chức năng nâng cao như điều khiển từ xa IR, Bluetooth hoặc kết nối WiFi, tích hợp giao tiếp MQTT, cho phép kết nối với internet để gửi và nhận tin nhắn. Tính năng này đảm bảo tương tác liền mạch với các thiết bị hoặc ứng dụng khác, giao diện tương tự web. *(bổ sung)*
- **Sơ đồ trạng thái (State diagram) của hệ thống:**

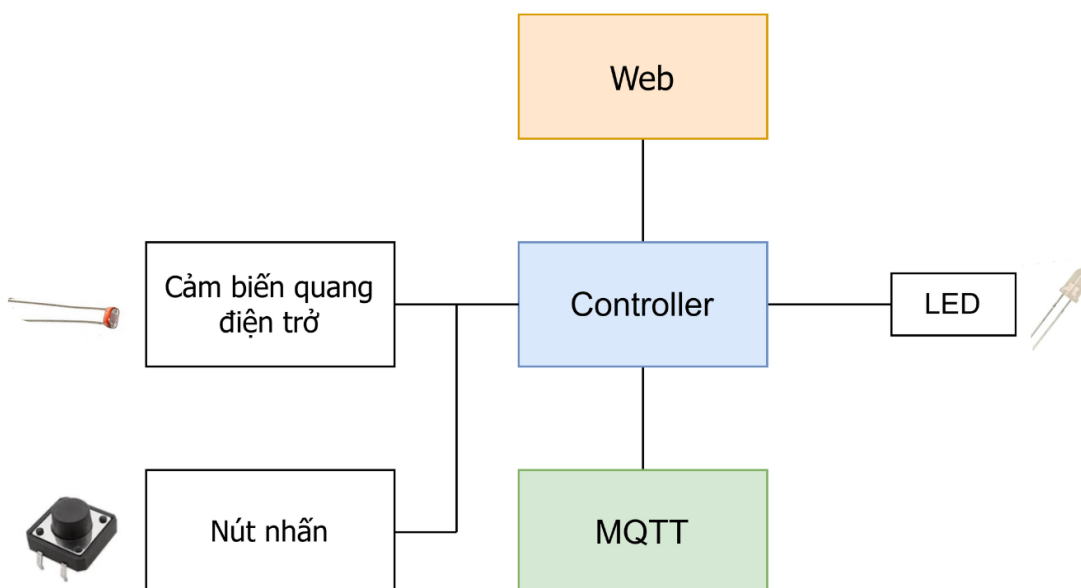


Current State	Input	Next State	Output
Nhấp nháy	Nhấn nút	Tắt (Manual mode)	Đèn tắt, trên web hiện trạng thái "OFF"
	Trời chuyển sáng	Tắt (Auto mode)	Đèn tắt, trên web hiện trạng thái "OFF"
Tắt (Auto mode)	Nhấn nút	Bật	Đèn bật, trên web hiện trạng thái "ON"
	Trời chuyển tối	Nhấp nháy	Đèn nhấp nháy, trên web hiện trạng thái "BLINKING"
Tắt (Manual mode)	Nhấn nút	Bật	Đèn bật, trên web hiện trạng thái "ON"
	Hết N phút khi trời sáng	Tắt (Auto mode)	Đèn tiếp tục tắt, trên web hiện trạng thái "OFF"
	Hết N phút khi trời tối	Nhấp nháy	Đèn nhấp nháy, trên web hiện trạng thái "BLINKING"
	Trời chuyển sáng	Tắt (Auto mode)	Đèn tiếp tục tắt, trên web hiện trạng thái "OFF"

	Trời chuyển tối	Nhấp nháy	Đèn nhấp nháy, trên web hiện trạng thái "BLINKING"
Bật	Nhấn nút	Tắt (Manual mode)	Đèn tắt, trên web hiện trạng thái "OFF"
	Hết N phút khi trời sáng	Tắt (Auto mode)	Đèn tắt, trên web hiện trạng thái "OFF"
	Hết N phút khi trời tối	Nhấp nháy	Đèn nhấp nháy, trên web hiện trạng thái "BLINKING"
	Trời chuyển sáng	Tắt (Auto mode)	Đèn tắt, trên web hiện trạng thái "OFF"
	Trời chuyển tối	Nhấp nháy	Đèn nhấp nháy, trên web hiện trạng thái "BLINKING"

II. THIẾT KẾ HỆ THỐNG

1. Thiết kế tổng quát



2. Thiết kế chi tiết (đề xuất các giải pháp)

Component /Module	Option 1	Option 2	Option 3	Selected option
Bộ điều khiển	Arduino Uno - Khoảng 120k	ESP32 - Khoảng 140k	ESP32_CAM - Khoảng 250k	ESP32
Cảm biến ánh sáng	Cảm biến quang trở (LDR) - 1k - 10k	Mô-đun cảm biến ánh sáng BH1750 - Khoảng 25k		Cảm biến quang trở (LDR)
Đèn LED	LED đơn màu	LED RGB	LED dải nhiều màu	LED đơn màu
Nút nhấn	Nút nhấn vật lý kết nối với arduino	Nút nhấn cảm ứng trên web	Kết hợp cả nút nhấn vật lý và cảm ứng trên web	Kết hợp cả nút nhấn vật lý và cảm ứng trên web
Giao diện web	Streamlit	Flask	Panel	Panel
Dịch vụ web	Local - miễn phí	Cloud - tốn phí duy trì		Local
MQTT	Node Red	IoT MQTT Panel (mobile) - dễ sử dụng		IoT MQTT Panel

- BỘ ĐIỀU KHIỂN

- **Option 1 (Arduino Uno):**

- ✓ **Ưu điểm:** Phổ biến, dễ lập trình, nhiều tài liệu hỗ trợ, dễ kết nối với nhiều loại module và cảm biến.
- ✓ **Nhược điểm:** Không có WiFi hoặc Bluetooth tích hợp, yêu cầu module mở rộng để kết nối mạng

- **Option 2 (ESP32):**

- ✓ **Ưu điểm:** Tích hợp WiFi và Bluetooth, mạnh mẽ hơn về khả năng xử lý, phù hợp cho các ứng dụng IoT.
- ✓ **Nhược điểm:** Đắt hơn, phức tạp hơn trong lập trình, tiêu thụ năng lượng cao hơn Arduino thông thường.

- **Option 3 (ESP32_CAM):**

- ✓ **Ưu điểm:** Tích hợp camera và WiFi, phù hợp cho các ứng dụng giám sát.
- ✓ **Nhược điểm:** Giá cao hơn, phức tạp trong lập trình và kết nối.

Lý do chọn ESP32: Đồ án yêu cầu kết nối không dây

- CẢM BIẾN ÁNH SÁNG

- **Option 1 (LDR):**
 - ✓ **Ưu điểm:** Rẻ, dễ sử dụng, không cần giao tiếp phức tạp với Arduino.
 - ✓ **Nhược điểm:** Độ chính xác thấp, bị ảnh hưởng bởi nhiệt độ và độ nhạy thấp khi ánh sáng yếu.
- **Option 2 (BH1750):**
 - ✓ **Ưu điểm:** Đo cường độ ánh sáng chính xác cao, độ phân giải cao, giao tiếp I2C, không phụ thuộc vào nhiệt độ.
 - ✓ **Nhược điểm:** Giá cao hơn, yêu cầu kỹ năng lập trình để điều khiển.

Lý do chọn LDR: Đáp ứng được yêu cầu cơ bản về phát hiện ánh sáng, dễ kết nối và giá thành thấp.

- ĐÈN LED:

- **Option 1 (LED đơn màu):**
 - ✓ **Ưu điểm:** Giá thành rẻ, dễ điều khiển, chỉ cần một chân GPIO.
 - ✓ **Nhược điểm:** Không thể thay đổi màu sắc, chỉ có thể bật hoặc tắt.
- **Option 2 (LED RGB):**
 - ✓ **Ưu điểm:** Có thể thay đổi màu sắc, tạo nhiều hiệu ứng ánh sáng khác nhau, phù hợp với các ứng dụng yêu cầu tín hiệu màu sắc.
 - ✓ **Nhược điểm:** Cần 3 chân GPIO, phức tạp hơn trong điều khiển, giá cao hơn LED đơn.
- **Option 3 (Dải LED):**
 - ✓ **Ưu điểm:** Cho phép hiển thị đa màu và hiệu ứng ánh sáng linh hoạt, tạo ấn tượng mạnh.
 - ✓ **Nhược điểm:** Cần điều khiển qua PWM hoặc bộ điều khiển riêng, tiêu thụ năng lượng lớn hơn, giá cao hơn.

Lý do chọn LED đơn màu: Đủ đáp ứng yêu cầu bật/tắt/nhấp nháy, tiết kiệm chi phí và dễ lập trình.

- NÚT NHẤN

- **Option 1 (Nút nhấn vật lý):**
 - ✓ **Ưu điểm:** Đơn giản, dễ kết nối, phản hồi ngay lập tức.
 - ✓ **Nhược điểm:** Không thể điều khiển từ xa qua web, chỉ có thể điều khiển trực tiếp trên phần cứng.

- **Option 2 (Nút nhấn cảm ứng trên web):**
 - ✓ **Ưu điểm:** Có thể điều khiển từ xa qua giao diện web, thuận tiện cho điều khiển qua mạng.
 - ✓ **Nhược điểm:** Phụ thuộc vào kết nối mạng và giao diện web, không có cảm giác phản hồi vật lý.
- **Option 3 (Kết hợp cả nút nhấn vật lý và cảm ứng trên web):**
 - ✓ **Ưu điểm:** Cung cấp tính linh hoạt trong việc điều khiển từ xa và trực tiếp.
 - ✓ **Nhược điểm:** Phức tạp hơn trong thiết kế và lập trình, cần đồng bộ trạng thái giữa nút nhấn vật lý và giao diện web.

Lý do chọn kết hợp cả hai: Đáp ứng nhu cầu linh hoạt của người dùng, có thể điều khiển từ xa và trực tiếp. (yêu cầu của đồ án)

- GIAO DIỆN WEB

- **Option 1 (Streamlit):**
 - ✓ **Ưu điểm:** Nhanh chóng tạo giao diện web cho Python, dễ tích hợp với dữ liệu từ Python.
 - ✓ **Nhược điểm:** Hạn chế trong tùy chỉnh giao diện và thiết kế, khó tích hợp các tính năng điều khiển thiết bị.
- **Option 2 (Flask):**
 - ✓ **Ưu điểm:** Đầy đủ tính năng backend, linh hoạt để xây dựng giao diện điều khiển và hiển thị dữ liệu cảm biến.
 - ✓ **Nhược điểm:** Cần kiến thức lập trình Python và cấu hình Flask.
- **Option 3 (Panel):**
 - ✓ **Ưu điểm:** Tích hợp tốt với Python, hỗ trợ giao diện tương tác.
 - ✓ **Nhược điểm:** Ít phổ biến hơn Flask, hạn chế tài liệu tham khảo.

Lý do chọn Panel: đáp ứng tốt các yêu cầu điều khiển LED và hiển thị dữ liệu cảm biến, ngoài ra có sẵn nhiều widget như text-to-speech hay chat bot phục vụ cho yêu cầu đồ án.

- DỊCH VỤ WEB

- **Option 1 (Local):**
 - ✓ **Ưu điểm:** Miễn phí, dễ thiết lập, không phụ thuộc mạng internet công cộng.
 - ✓ **Nhược điểm:** Hạn chế trong việc điều khiển từ xa qua internet.
- **Option 2 (Cloud):**
 - ✓ **Ưu điểm:** Có thể điều khiển từ xa, dễ truy cập mọi nơi.
 - ✓ **Nhược điểm:** Tốn phí duy trì, phụ thuộc vào dịch vụ bên thứ ba.

Lý do chọn Local: vì phù hợp với mục tiêu thử nghiệm và không tốn phí duy trì.

- MQTT

- **Option 1 (Node Red)**

- ✓ **Ưu điểm:** Giao diện trực quan, dễ thiết lập cho người mới.
- ✓ **Nhược điểm:** Hạn chế tính năng nâng cao so với ứng dụng chuyên dụng.

- **Option 2 (IoT MQTT Panel - Mobile):**

- ✓ **Ưu điểm:** Dễ sử dụng, tiện lợi khi điều khiển từ điện thoại.
- ✓ **Nhược điểm:** Giới hạn về cấu hình và tính năng so với Node Red.

Lý do chọn IoT MQTT Panel: vì dễ sử dụng và tiện lợi trên thiết bị di động.

III. KIỂM THỬ

1. Kiểm thử từng thành phần

Thành phần	Mục tiêu kiểm thử
ESP32	Kiểm tra kết nối internet
Cảm biến ánh sáng	Đảm bảo cảm biến có thể nhận và phát hiện ánh sáng đúng cách.
LED	Đảm bảo LED sáng và tắt theo tín hiệu từ ESP32.
Nút nhấn	Đảm bảo nút nhấn có thể kích hoạt và gửi tín hiệu đến ESP32.

2. Kiểm thử tích hợp

Bước	Tích hợp	Mục tiêu kiểm thử
1	ESP32 + nút nhấn + LED	Đảm bảo LED bật và tắt luân phiên khi nhấn nút
2	ESP32 + cảm biến ánh sáng + LED	Kiểm tra xem LED có thay đổi trạng thái theo tín hiệu từ cảm biến ánh sáng hay không.
3	ESP32 + cảm biến ánh sáng + nút nhấn + LED	Kiểm tra hoạt động đồng thời của cảm biến ánh sáng và nút nhấn, điều khiển LED phù hợp.
4	ESP32 + cảm biến ánh sáng + nút nhấn + LED + Web + MQTT	Xác minh khả năng giao tiếp giữa ESP32 với Web và MQTT, kiểm tra cập nhật trạng thái của cảm biến và LED, điều khiển LED từ web và MQTT

IV. TÀI LIỆU THAM KHẢO

Tài liệu cung cấp trên Moodle