

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



Nhập môn lập trình kết nối vạn vật
Hello IoT – Project proposal

Giảng viên hướng dẫn:

- Nguyễn Đức Hoàng Hạ
- Đỗ Thị Thanh Hà

Lớp: 21KHDL

MSSV: 21127412

Họ tên: Hồ Bạch Như Quỳnh

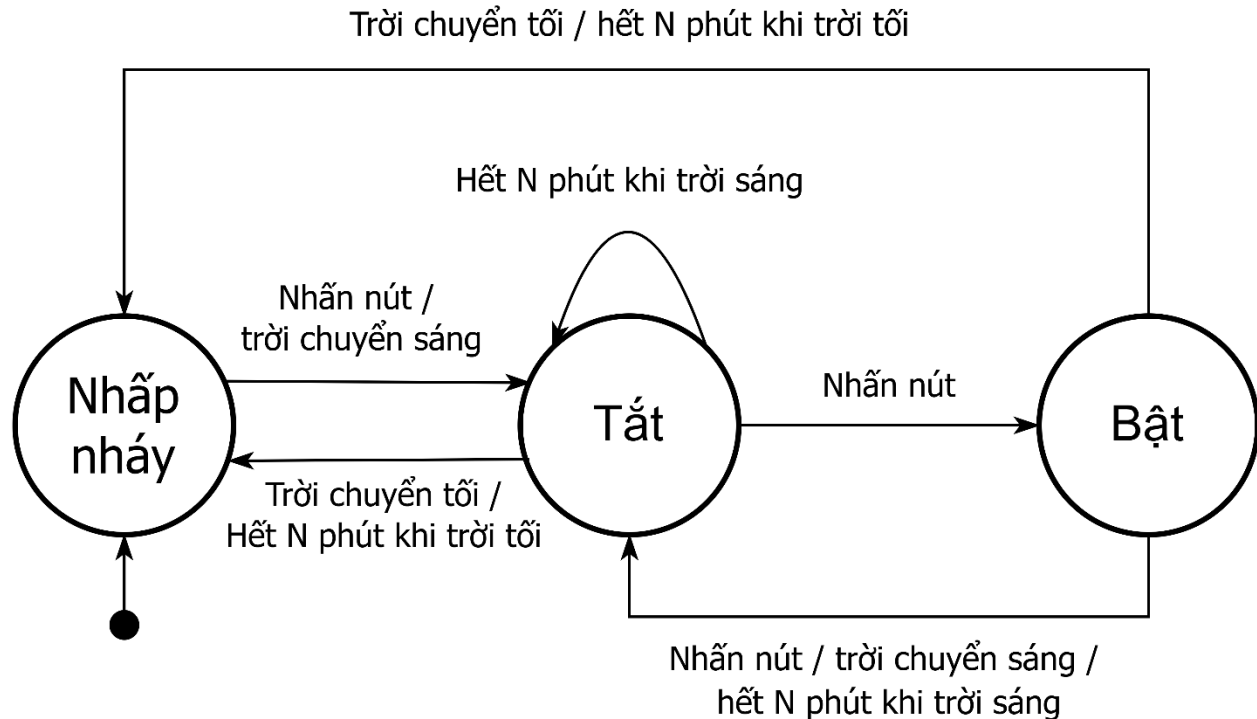
NỘI DUNG

<i>I. ĐẶC TẢ YÊU CẦU</i>	1
<i>II. THIẾT KẾ HỆ THỐNG</i>	2
1. Thiết kế tổng quát.....	2
2. Thiết kế chi tiết (đề xuất các giải pháp).....	3
<i>III. KIỂM THỬ</i>	6
1. Kiểm tra từng thành phần.....	6
2. Kiểm tra tích hợp.....	6
<i>IV. TÀI LIỆU THAM KHẢO</i>	6

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 (automation):
 - ✓ Cảm biến ánh sáng dùng để nhận biết ngày hoặc đêm (quy ước giá trị cảm biến < 50 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):
 - ✓ 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

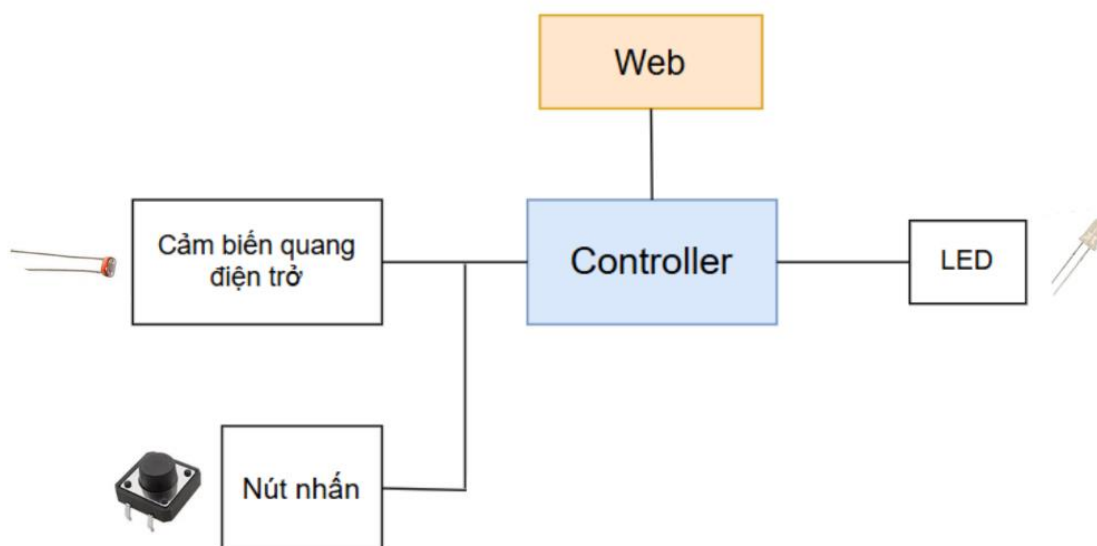
- Sơ đồ trạng thái (State diagram) của hệ thống:



Current State	Input	Next State	Output
Nhấp nháy (blink)	Nhấn nút	Tắt	Đèn tắt, trên web hiện trạng thái "OFF"
	Trời chuyển sáng		
Tắt (off)	Trời chuyển tối	Nhấp nháy	Đèn nhấp nháy, trên web hiện trạng thái "BLINKING"
	Hết N phút khi trời tối		
	Nhấn nút	Bật	Đèn bật, trên web hiện trạng thái "ON"
	Hết N phút khi trời sáng	Tắt	None
Bật (on)	Nhấn nút	Tắt	Đèn tắt, trên web hiện trạng thái "OFF"
	Trời chuyển sáng		
	Hết N phút khi trời sáng		
	Trời chuyển tối	Nhấp nháy	Đèn nhấp nháy, trên web hiện trạng thái "BLINKING"
	Hết N phút khi trời tối		

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
Arduino	Arduino Uno - Khoảng 120k	Arduino Nano - Khoảng 30k	ESP32	Arduino Uno
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	HTMML/CSS cơ bản	Streamlit	Flask	Flask
Giao tiếp Arduino với web	Serial (USB)	Kết nối qua Wifi	Kết nối qua Bluetooth	Serial (USB)

- Arduino

- **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 (Arduino Nano):**
 - ✓ **Ưu điểm:** Nhỏ gọn, tiết kiệm không gian, chi phí thấp.
 - ✓ **Nhược điểm:** Tương tự Uno, không có WiFi/Bluetooth, ít chân GPIO hơn, khó sử dụng khi cần nhiều kết nối ngoại vi.
- **Option 3 (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.

Lý do chọn Arduino Uno: Dễ sử dụng, đủ khả năng cho hệ thống cơ bản, chi phí thấp.

- 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 (HTML/CSS cơ bản):**
 - ✓ **Ưu điểm:** Dễ triển khai, không cần nhiều tài nguyên.
 - ✓ **Nhược điểm:** Thiếu tính năng động, không hỗ trợ backend để xử lý dữ liệu cảm biến hoặc điều khiển LED.
- **Option 2 (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 3 (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.

Lý do chọn Flask: Đáp ứng tốt yêu cầu điều khiển LED và hiển thị dữ liệu cảm biến.

- Giao tiếp giữa web và Arduino

- **Option 1 (Serial/USB):**
 - ✓ **Ưu điểm:** Đơn giản, ổn định, không cần module bổ sung, dễ dùng cho ứng dụng cục bộ.
 - ✓ **Nhược điểm:** Cần kết nối dây USB, hạn chế khoảng cách giữa Arduino và máy tính/web server.
- **Option 2 (WiFi):**
 - ✓ **Ưu điểm:** Cho phép điều khiển từ xa qua mạng, phù hợp với các ứng dụng IoT.
 - ✓ **Nhược điểm:** Yêu cầu phần cứng bổ sung (như ESP8266), phức tạp hơn trong thiết lập mạng.

- **Option 3 (Bluetooth):**

- ✓ **Ưu điểm:** Có thể điều khiển không dây trong phạm vi gần, tiết kiệm năng lượng hơn WiFi.
- ✓ **Nhược điểm:** Khoảng cách ngắn, cần cấu hình phức tạp hơn so với Serial, không phù hợp với điều khiển qua web nếu không có máy chủ trung gian.

Lý do chọn Serial (USB): Đáp ứng tốt cho việc thử nghiệm cục bộ, đơn giản, chi phí thấp.

III. KIỂM THỬ

1. Kiểm tra từng thành phần

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

2. Kiểm tra tích hợp

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

IV. TÀI LIỆU THAM KHẢO

Tài liệu cung cấp trên Moodle