



fit@hcmus

Báo cáo Đồ án giữa kỳ

Đề tài: Smart Lighting Control System

Sinh viên thực hiện:

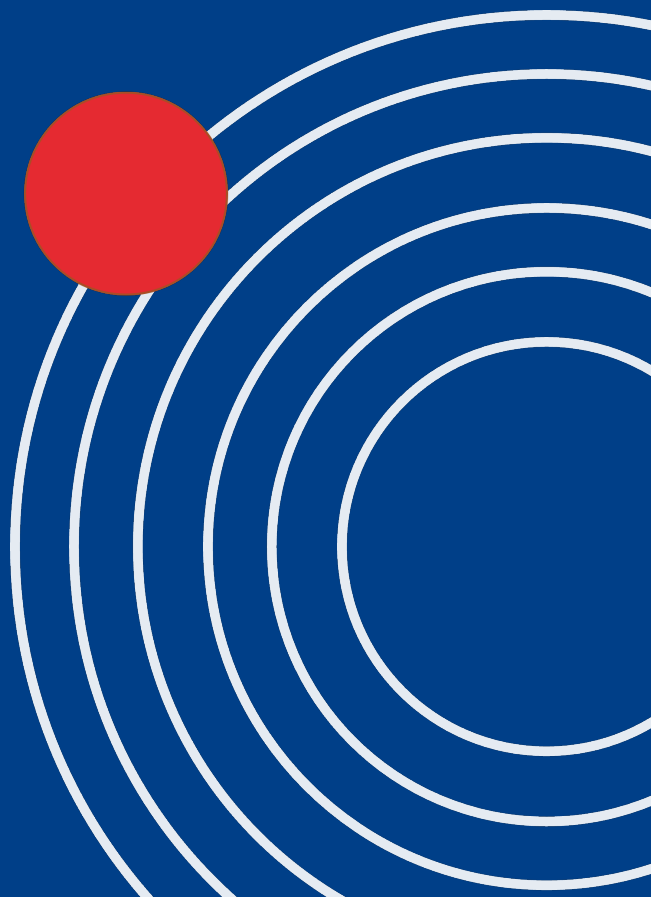
Trần Hiếu Tâm (21127423)

Giảng viên hướng dẫn:

TS. Nguyễn Đức Hoàng Hạ

ThS. Đỗ Thị Thanh Hà

TP. Hồ Chí Minh, tháng 11 năm 2024



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



Báo cáo Đồ án giữa kỳ

Đề tài: Smart Lighting Control System

Học phần: Nhập môn lập trình kết nối vạn vật - 21KHDL

Sinh viên thực hiện:

Trần Hiếu Tâm (21127423)

Giảng viên hướng dẫn:

TS. Nguyễn Đức Hoàng Hạ

ThS. Đỗ Thị Thanh Hà

TP. Hồ Chí Minh, tháng 11 năm 2024

Mục lục

| | | |
|----------|---|-----------|
| 1 | Đặc tả yêu cầu | 1 |
| 2 | Thiết kế hệ thống | 3 |
| 2.1 | Thiết kế tổng quát | 3 |
| 2.2 | Thiết kế chi tiết | 4 |
| 2.3 | Sơ đồ mạch điện đề xuất | 5 |
| 2.4 | Sơ đồ máy trạng thái hữu hạn (FSM - Finite-state Machine) | 6 |
| 2.5 | Sơ đồ truyền nhận dữ liệu | 8 |
| 2.5.1 | Truyền dữ liệu, tín hiệu giữa Controller và các thiết bị | 9 |
| 2.5.2 | Truyền dữ liệu, tín hiệu từ Controller đến Web Service | 10 |
| 2.5.3 | Truyền dữ liệu, tín hiệu từ Web Service đến Controller | 11 |
| 3 | Đề xuất các giải pháp | 12 |
| 4 | Kế hoạch kiểm thử | 15 |
| 4.1 | Kiểm tra từng thành phần | 15 |
| 4.2 | Kiểm tra tích hợp | 16 |
| 5 | Kết quả triển khai | 17 |
| 5.1 | Giả lập hệ thống trên Tinkercad | 17 |
| 5.2 | Triển khai hệ thống trên thiết bị thật | 18 |
| 5.3 | Triển khai Website | 19 |
| 5.4 | Giới thiệu chức năng website | 21 |
| 5.5 | Release | 22 |
| | Tài liệu tham khảo | 23 |

Danh sách bảng

| | | |
|---|---|----|
| 1 | Danh sách các thành phần của hệ thống | 4 |
| 2 | Bảng chuyển đổi trạng thái của đèn LED trong hệ thống | 7 |
| 3 | Các thiết bị truyền dữ liệu, tín hiệu đến Controller | 9 |
| 4 | Các thiết bị nhận tín hiệu từ Controller | 9 |
| 5 | Bảng mô tả ý nghĩa của các trường dữ liệu gửi từ Controller đến Web Service | 10 |
| 6 | Bảng mô tả ý nghĩa của các trường dữ liệu gửi từ Web Service đến Controller | 11 |
| 7 | Danh sách đề xuất các giải pháp cho hệ thống | 14 |
| 8 | Kiểm tra các thành phần hệ thống | 15 |
| 9 | Các bước tích hợp và kiểm tra hệ thống | 16 |

Danh sách hình vẽ

| | | |
|---|---|----|
| 1 | System Design | 3 |
| 2 | Sơ đồ mạch điện dự kiến triển khai hệ thống | 5 |
| 3 | Sơ đồ máy trạng thái hữu hạn của đèn LED trong hệ thống | 6 |
| 4 | Sơ đồ truyền nhận dữ liệu giữa các thiết bị, Controller và Web Server | 8 |
| 5 | Triển khai mạch Arduino trên Tinkercad | 17 |
| 6 | Triển khai hệ thống thực tế | 18 |
| 7 | Triển khai hệ thống thực tế tích hợp Web Service | 18 |
| 8 | Giao diện Web Service sau khi kết nối với Web Server thành công | 20 |
| 9 | Các thành phần chính của Website | 21 |

1 Đặc tả yêu cầu

Hệ thống điều khiển ánh sáng thông minh (tiếng Anh: Smart Lighting Control System) là một hệ thống tự động điều chỉnh đèn dựa trên điều kiện ánh sáng xung quanh và hỗ trợ người dùng điều khiển thủ công khi cần thiết nhằm tiết kiệm năng lượng điện cũng như có thể vận hành hiệu quả, chính xác mà không cần sự can thiệp của con người.

Yêu cầu:

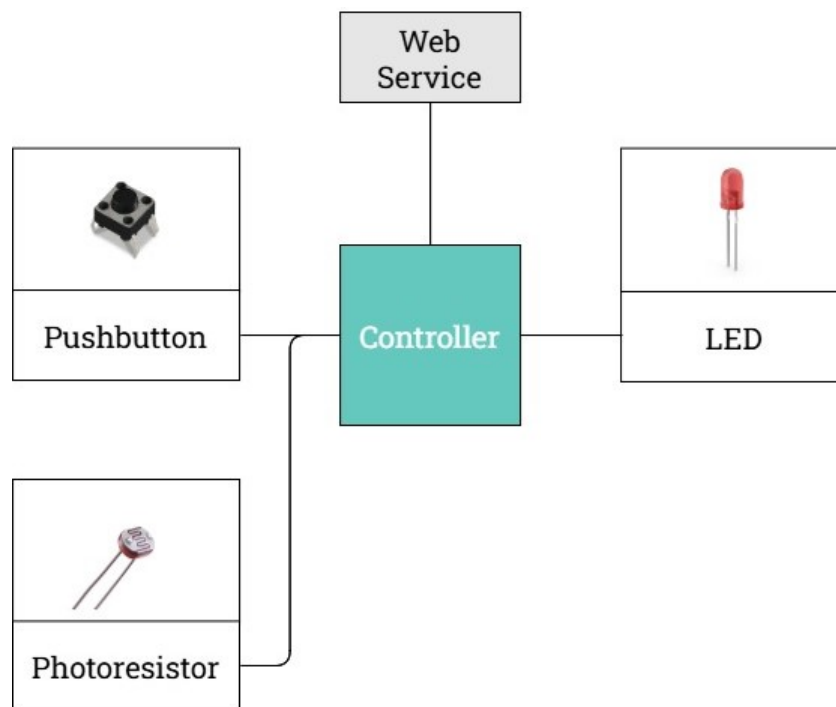
- Hệ thống gồm 01 đèn LED, 01 cảm biến ánh sáng dùng để điều khiển tự động đèn LED theo điều kiện ánh sáng xung quanh (Auto Mode) và 01 nút bấm dùng để điều khiển thủ công (Manual Mode).
- Hệ thống vận hành theo 2 chế độ (mode): Auto Mode và Manual Mode.
- Khi bắt đầu chạy thì hệ thống sẽ chạy theo Auto Mode: Đèn LED sẽ tự động tắt (Off) khi trời sáng và sẽ tự động sáng nhấp nháy (Blinking) khi trời tối.
- Khi nhấn vào nút bấm thì hệ thống sẽ chuyển từ chế độ Auto Mode sang Manual Mode: Đèn sẽ tạm thời không vận hành tự động (Auto Mode) theo điều kiện ánh sáng môi trường. Theo đó, nếu người dùng nhấn nút bấm thì đèn sẽ tắt nếu đèn đang sáng và ngược lại, nếu người dùng nhấn nút bấm thì đèn sẽ bật nếu hiện tại đèn đang tắt. Nếu người dùng nhấn nút khi đèn đang blinking thì đèn sẽ tắt.
- Nếu hệ thống đang trong Manual Mode, khi có sự chuyển đổi ánh sáng ngày đêm thì hệ thống sẽ mặc định chuyển về Auto Mode và đèn LED sẽ vận hành tự động theo điều kiện ánh sáng môi trường.
- Ngoài ra, khi hệ thống đang trong Manual Mode mà sau N phút người dùng không nhấn nút bấm điều khiển bật tắt đèn thủ công thì hệ thống sẽ tự động chuyển sang Auto Mode. Người dùng có thể thiết lập giá trị N này qua website (Nếu người dùng cài đặt $N = 0$ thì hệ thống sẽ chỉ chuyển từ Manual Mode sang Auto Mode khi có sự chuyển đổi ánh sáng ngày đêm).
- Hệ thống sẽ hỗ trợ website để người dùng theo dõi trạng thái đèn hiện tại (Bật, Tắt, Blinking), mode (Auto Mode hay Manual Mode), điều kiện ánh sáng hiện tại (Light hay Night), điều kiện thời gian N để chuyển mode cho hệ thống. Ngoài ra, người dùng có thể bật, tắt đèn thông qua Website này, cũng như điều chỉnh thông số N (Đơn vị: millisecond).

Ràng buộc:

- Hệ thống mô phỏng chỉ được thử nghiệm trên nguồn điện thấp (5-10V), không sử dụng điện dân dụng (220V) để thử nghiệm và vận hành hệ thống này.
- Hệ thống không được sử dụng các loại cảm biến thu thập dữ liệu riêng tư của con người dưới mọi hình thức.
- Hệ thống luôn phải được cung cấp điện liên tục khi vận hành.
- Thiết bị và linh kiện nên phổ biến ở thị trường Việt Nam, có thể dễ dàng tìm kiếm trên địa bàn Thành phố Hồ Chí Minh. Đồng thời ưu tiên sử dụng các thiết bị và linh kiện mà Tinkercad có hỗ trợ mô phỏng.
- Chi phí không vượt quá 200.000 VNĐ (bằng chữ: Hai trăm nghìn Việt Nam Đồng chẵn)

2 Thiết kế hệ thống

2.1 Thiết kế tổng quát



Hình 1: System Design

Block Diagram của hệ thống được mô tả qua hình 1 cho biết thiết kế tổng quát của đề án này. Theo đó, hệ thống gồm các thành phần sau: (1) Các thiết bị Input là Pushbutton (tên tiếng Việt: nút nhấn) để nhận tín hiệu bật tắt đèn của người dùng và Photoresistor (tên tiếng Việt: quang trở) để đo lường độ sáng của môi trường xung quanh hệ thống; (2) Thiết bị Output là đèn LED; (3) Controller đóng vai trò là thiết bị để nhận dữ liệu từ thiết bị Input, điều khiển thiết bị Output cũng như truyền và nhận dữ liệu qua Web Service; (4) Web Service là một ứng dụng web được triển khai nhằm mục đích cho người dùng biết được trạng thái hiện tại của đèn LED cũng như điều khiển đèn LED từ xa thông qua nút trên Website. Các thành phần trên sẽ được mô tả chi tiết tại bảng 1.

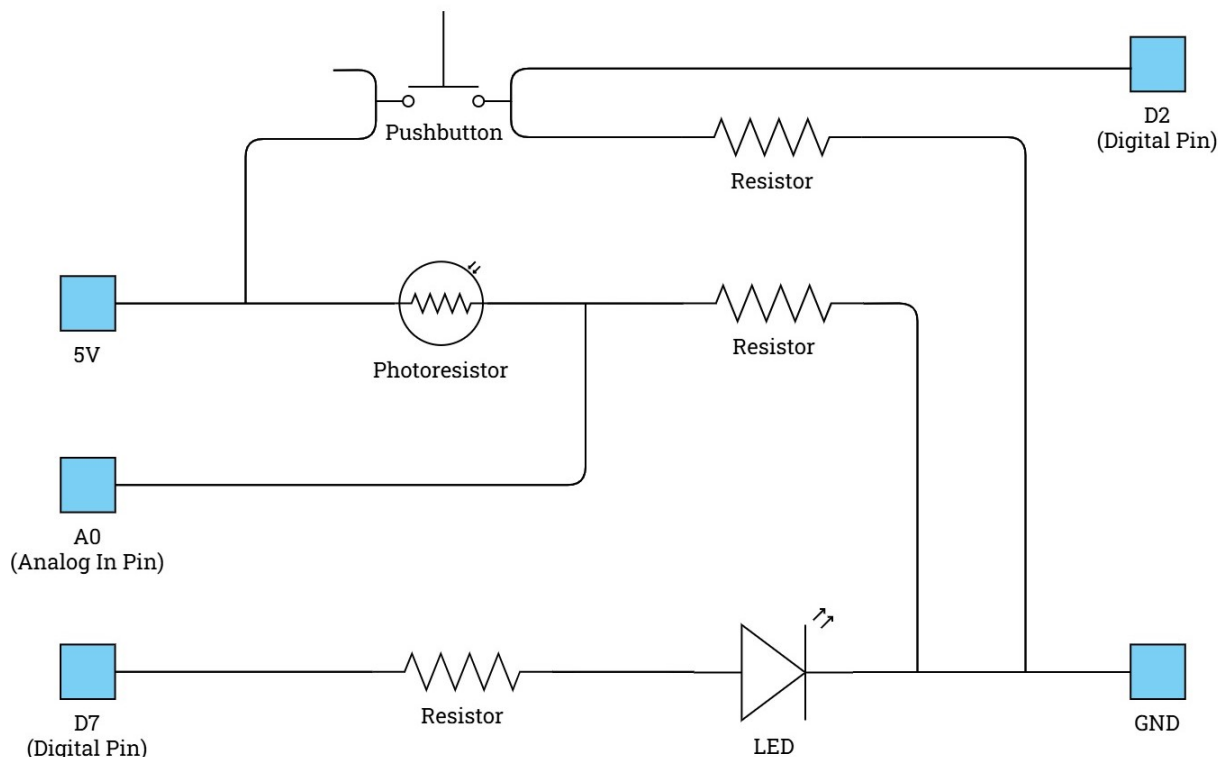
2.2 Thiết kế chi tiết

| TT | Thành phần | Tên thiết bị/Công nghệ đề xuất | Cách thiết lập |
|----|---------------|---|---|
| 1 | Controller | Arduino Uno R3 AT-mega328 | <ul style="list-style-type: none"> - Nguồn và GND: Nối chân 5V và GND của Arduino với nguồn điện và GND cho các thiết bị. - Chân Digital: Nối các thiết bị như LED và Push-button vào chân Digital (Pin 2-13) của Arduino để truyền/nhận tín hiệu điều khiển. - Chân Analog In: Nối cảm biến ánh sáng vào chân Analog In (Pin A0-A5) để đọc tín hiệu ánh sáng. |
| 2 | Web Service | Local Host, triển khai bằng Flask, HTML, CSS | <ul style="list-style-type: none"> - Triển khai trên Local Host: Cài đặt Flask trên laptop để tạo server cục bộ. - Gửi/Nhận Dữ liệu: Thiết lập kết nối giữa Arduino và Web Service thông qua cổng USB của Arduino để trao đổi dữ liệu điều khiển và giám sát trạng thái đèn LED từ Web Service đến Arduino và ngược lại. |
| 3 | Pushbutton | 6x6x5 mm Miniature Micro Momentary Tactile Tact Touch Push Button | <ul style="list-style-type: none"> - Kết nối với Arduino: Nối một chân của Pushbutton với chân Digital của Arduino và chân còn lại với GND. - Điện trở: Thêm điện trở giữa chân Digital và GND để ổn định tín hiệu, tránh nhiễu. Arduino sẽ đọc tín hiệu từ Pushbutton để điều khiển đèn LED. |
| 4 | Photoresistor | Lm393 Optical Photosensitive Ldr Light Sensor Module | <ul style="list-style-type: none"> - Kết nối với Arduino: Nối chân tín hiệu của module cảm biến ánh sáng (LDR) với chân Analog In của Arduino. Nối chân VCC và GND của module với chân 5V và GND của Arduino. - Đọc cường độ ánh sáng: Arduino sẽ đọc giá trị từ LDR để xác định trạng thái sáng tối của môi trường xung quanh. |
| 5 | LED | 5mm Color Led | <ul style="list-style-type: none"> - Kết nối với Arduino: Nối chân dương của LED với chân Digital của Arduino qua một điện trở để bảo vệ LED. Nối chân âm của LED với GND. - Điều khiển Bật/Tắt: Arduino sẽ điều khiển LED dựa trên tín hiệu từ cảm biến ánh sáng hoặc Pushbutton hoặc nhấn nút bật tắt từ Website. |

Bảng 1: Danh sách các thành phần của hệ thống

Bảng 1 cho biết mô tả chi tiết về các thành phần Controller, Web Service, Pushbutton, Photoresistor, đèn LED của hệ thống đã được thiết kế trên System Design ở hình 1. Theo đó, bảng 1 đưa ra đề xuất thiết bị, công nghệ phù hợp cũng như cách thiết lập các thiết bị đó để có cơ sở thiết kế chi tiết và triển khai ở các giai đoạn sau (như thiết kế sơ đồ mạch điện, FSM, sơ đồ truyền nhận dữ liệu, triển khai giả lập và triển khai trên thiết bị thật).

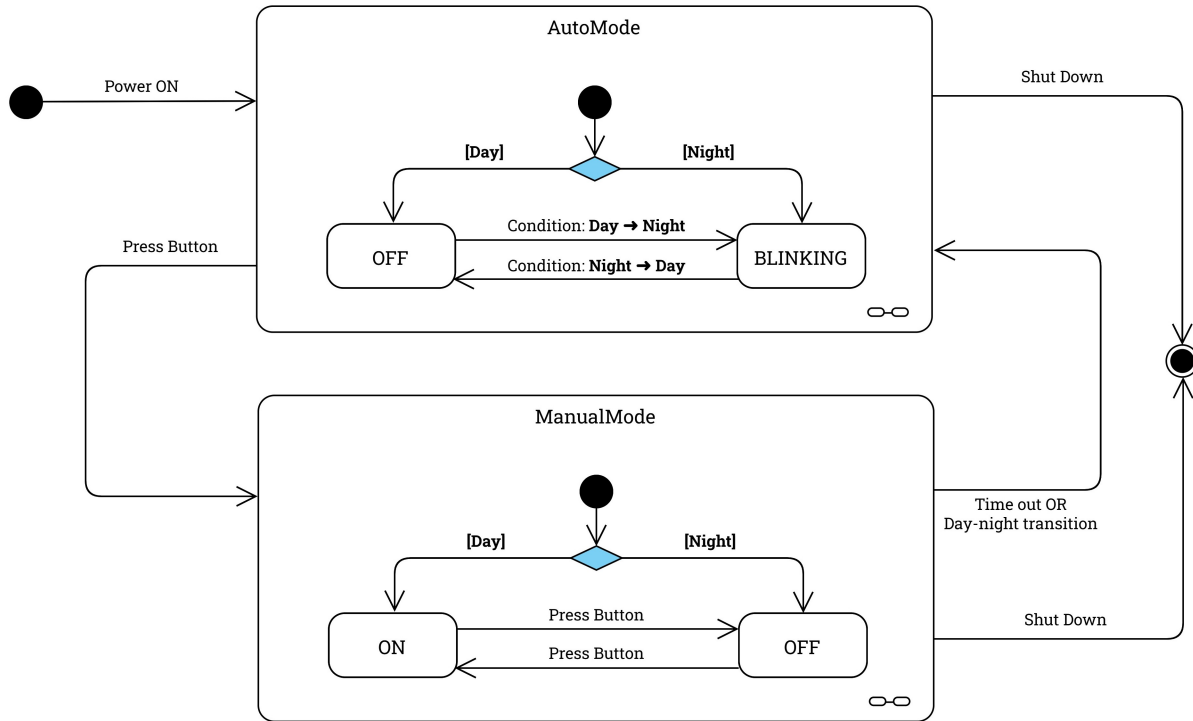
2.3 Sơ đồ mạch điện đề xuất



Hình 2: Sơ đồ mạch điện dự kiến triển khai hệ thống

Sơ đồ mạch điện trên hình 2 mô tả một mạch bao gồm một nút nhấn (Pushbutton), một quang trở (Photoresistor) và một đèn LED được kết nối với các chân của vi điều khiển Arduino UNO. Các chân Arduino UNO được sử dụng bao gồm: chân nguồn 5V và GND; chân Digital In D2 và chân Digital Out D7; chân Analog In A0. Nguồn điện 5V được cung cấp cho mạch qua chân 5V và chân GND. Khi nút nhấn được bấm, mạch cho phép dòng điện đi qua các điện trở và quang trở đến chân A0 (chân Analog) để đo ánh sáng từ quang trở. Chân D2 (Digital Pin) được sử dụng để theo dõi trạng thái của nút nhấn và D7 (Digital Pin) được sử dụng để điều khiển đèn LED. Đèn LED, quang trở và nút nhấn được mắc nối tiếp qua các điện trở để hạn chế dòng điện nhằm bảo vệ các thiết bị trên khỏi dòng điện quá lớn và đảm bảo mạch hoạt động ổn định.

2.4 Sơ đồ máy trạng thái hữu hạn (FSM - Finite-state Machine)



Hình 3: Sơ đồ máy trạng thái hữu hạn của đèn LED trong hệ thống

Sơ đồ FSM (Finite State Machine) ở hình 3 mô tả hệ thống điều khiển đèn với hai chế độ: Auto-Mode và ManualMode. Khi Arduino UNO được nạp điện (Power ON), chế độ AutoMode được kích hoạt. Trong AutoMode, nếu là ban ngày (Day), đèn sẽ ở trạng thái OFF; nếu chuyển sang ban đêm (Night), đèn sẽ chuyển sang trạng thái BLINKING. Khi trời sáng trở lại (Night sang Day), đèn sẽ quay về trạng thái OFF và ngược lại, khi trời chuyển từ sáng sang tối (Day sang Night) thì đèn sẽ sang trạng thái BLINKING. Khi người dùng nhấn nút, hệ thống chuyển sang ManualMode.

Trong ManualMode, đèn có thể được bật (ON) hoặc tắt (OFF) thông qua nút nhấn. Ở trạng thái ON, nếu nhấn nút lần nữa, đèn sẽ chuyển sang OFF và ngược lại. Hệ thống sẽ quay lại AutoMode khi có sự chuyển đổi từ ngày sang đêm, đêm sang ngày hoặc khi hết thời gian quy định (timeout). Ở bất kỳ trạng thái nào, hệ thống sẽ kết thúc hoạt động khi bị ngắt nguồn (tức là Shut Down).

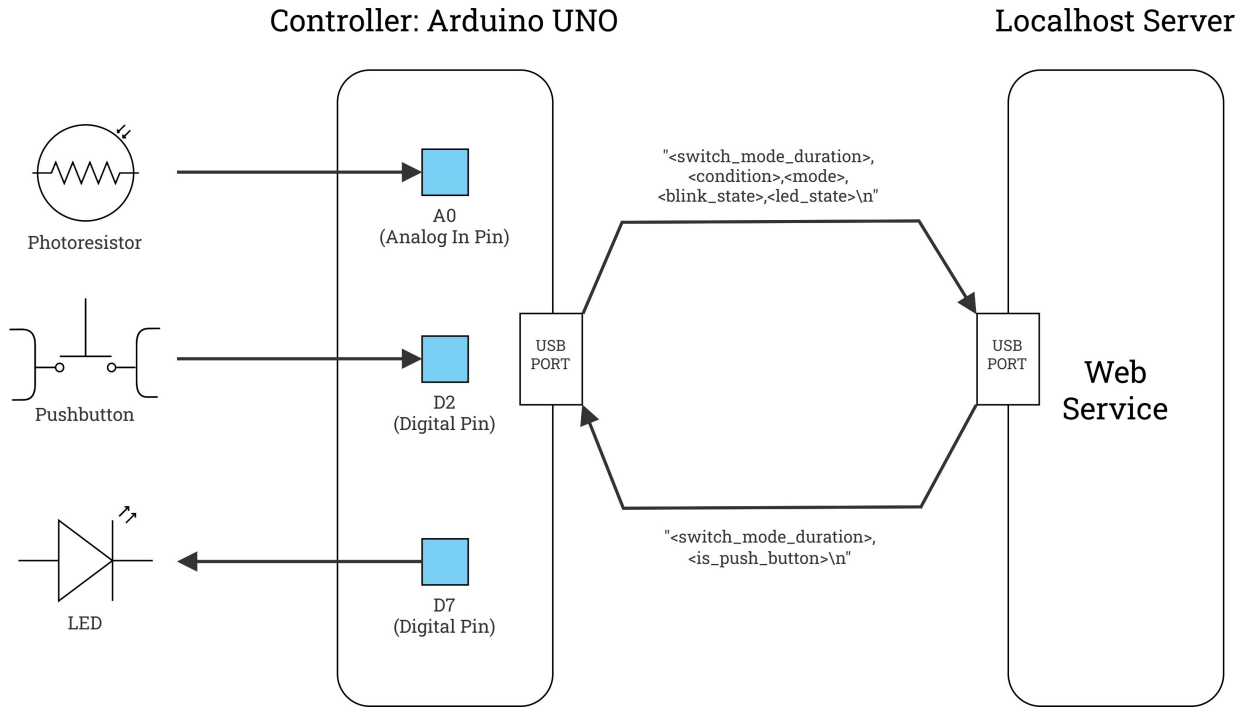
Sự chuyển đổi trạng thái của sơ đồ FSM ở hình 3 được mô tả thông qua bảng 2: Bảng chuyển đổi trạng thái của đèn LED trong hệ thống.

| Current State | Input | Condition | Next State | Output |
|---------------------|------------------------|----------------------------|---------------------|---|
| Initial State | Power ON | Current Condition == Day | AutoMode - OFF | Start AutoMode |
| Initial State | Power ON | Current Condition == Night | AutoMode - BLINKING | Start blinking LED with AutoMode |
| AutoMode - OFF | Day → Night Transition | - | AutoMode - BLINKING | Start blinking LED |
| AutoMode - BLINKING | Night → Day Transition | - | AutoMode - OFF | Turn off LED |
| AutoMode - OFF | Press Button | - | ManualMode - ON | Switch to ManualMode and turn on LED |
| AutoMode - BLINKING | Press Button | - | ManualMode - OFF | Switch to ManualMode and turn off LED |
| ManualMode - ON | Press Button | - | ManualMode - OFF | Turn off LED |
| ManualMode - OFF | Press Button | - | ManualMode - ON | Turn on LED |
| ManualMode - ON | Day-Night Transition | Current Condition == Day | AutoMode - OFF | Switch to AutoMode and turn off LED |
| ManualMode - ON | Day-Night Transition | Current Condition == Night | AutoMode - BLINKING | Switch to AutoMode and start blinking LED |
| ManualMode - ON | Timeout | Current Condition == Day | AutoMode - OFF | Switch to AutoMode and turn off LED |
| ManualMode - ON | Timeout | Current Condition == Night | AutoMode - BLINKING | Switch to AutoMode and start blinking LED |
| ManualMode - OFF | Day-Night Transition | Current Condition == Day | AutoMode - OFF | Switch to AutoMode and turn off LED |
| ManualMode - OFF | Day-Night Transition | Current Condition == Night | AutoMode - BLINKING | Switch to AutoMode and start blinking LED |
| ManualMode - OFF | Timeout | Current Condition == Day | AutoMode - OFF | Switch to AutoMode and turn off LED |
| ManualMode - OFF | Timeout | Current Condition == Night | AutoMode - BLINKING | Switch to AutoMode and start blinking LED |
| AutoMode - OFF | Shut Down | - | Shutdown State | Shut Down System |
| AutoMode - BLINKING | Shut Down | - | Shutdown State | Shut Down System and turn off LED |
| ManualMode - ON | Shut Down | - | Shutdown State | Shut Down System and turn off LED |
| ManualMode - OFF | Shut Down | - | Shutdown State | Shut Down System |

Bảng 2: Bảng chuyển đổi trạng thái của đèn LED trong hệ thống

Căn cứ FSM trên hình 3, bảng 2 ở trên mô tả chuyển đổi trạng thái của đèn LED trong hệ thống.

2.5 Sơ đồ truyền nhận dữ liệu



Hình 4: Sơ đồ truyền nhận dữ liệu giữa các thiết bị, Controller và Web Server

Sơ đồ tại hình 4 mô tả quá trình truyền nhận dữ liệu giữa Arduino UNO và Web Service qua cổng USB. Arduino đọc giá trị từ quang trở (qua cổng A0), nút nhấn (qua cổng D2) và căn cứ vào các dữ liệu đó để điều khiển đèn LED (qua cổng D7). Trong quá trình xử lý, Arduino UNO gửi liên tục các thông tin hiện thời về thời gian timeout (đơn vị: millisecond), chế độ đang vận hành hệ thống (AutoMode hay ManualMode), điều kiện ánh sáng môi trường (Day hay Night), trạng thái nhấp nháy (đèn có đang Blinking không) và trạng thái đèn (đang sáng - ON hay đang tắt - OFF) đến Web Service thông qua USB Port dưới dạng 1 chuỗi ký tự. Ở chiều ngược lại, Web Service cũng có thể gửi lệnh điều khiển dưới hình thức người dùng thao tác trên Website và Web Service sẽ xử lý thành 1 chuỗi các trường dữ liệu và gửi tới Arduino để thay đổi trạng thái hệ thống. Việc truyền nhận dữ liệu giữa Arduino UNO và Web Service thông qua cổng USB được diễn ra liên tục nhằm cập nhật chính xác nhất trạng thái hiện thời của hệ thống.

Sơ đồ tại hình 4 được mô tả chi tiết và rõ ràng hơn thông qua các phần 2.5.1, 2.5.2, 2.5.3.

2.5.1 Truyền dữ liệu, tín hiệu giữa Controller và các thiết bị

| Thiết bị | Pin giao tiếp | Điện thế | Giá trị | Ý nghĩa |
|---------------|---------------|-----------|------------|--|
| Photoresistor | A0 | 0V đến 5V | 0 đến 1023 | Cường độ ánh sáng của môi trường xung quanh hệ thống |
| Pushbutton | D2 | 0V | LOW | Released |
| | | 5V | HIGH | Pressed |

Bảng 3: Các thiết bị truyền dữ liệu, tín hiệu đến Controller

| Thiết bị | Pin giao tiếp | Điện thế | Giá trị | Ý nghĩa |
|----------|---------------|----------|---------|--------------------------|
| LED | D7 | 5V | HIGH | ON: tức bật sáng đèn LED |
| | | 0V | LOW | OFF: tức tắt đèn LED |

Bảng 4: Các thiết bị nhận tín hiệu từ Controller

Người đọc có thể tham khảo sơ đồ mạch điện trên hình 2, sơ đồ truyền nhận dữ liệu trên hình 4 và mạch giả lập hệ thống trên Tinkercad tại hình 5 để có góc nhìn rộng hơn về các thiết bị này và các pin liên quan được đề cập trong bảng 3 và bảng 4.

2.5.2 Truyền dữ liệu, tín hiệu từ Controller đến Web Service

- Cấu trúc:

```
1 <switch_mode_duration>,<condition>,<mode>,<blink_state>,<led_state>\n"
```

- Mỗi thông tin cách nhau bởi dấu phẩy, kết thúc chuỗi thông tin gửi bằng dấu “\n”

| Trường dữ liệu | Ý nghĩa | Giá trị gửi | Giá trị hiển thị trên Website |
|----------------------|---|---------------------|---|
| switch_mode_duration | Thời gian chờ từ lần nhấn nút cuối cùng để hệ thống tự động chuyển từ Manual Mode sang Auto Mode, có thể được người dùng cài đặt trên Website. Lưu ý: nếu đặt giá trị là 0, điều kiện timeout này sẽ không được áp dụng. Đơn vị: millisecond. | 0 đến 4,294,967,295 | 0 đến 4,294,967,295 (millisecond) |
| condition | Điều kiện ánh sáng của môi trường xung quanh hệ thống (sáng hay tối) | 0 | Night |
| | | 1 | Day |
| mode | Chế độ hiện tại điều khiển hệ thống là Tự động (Auto) hay Thủ công (Manual) | 0 | Manual |
| | | 1 | Auto |
| blink_state | Cho biết đèn LED có đang trong trạng thái chớp nháy hay không. | 0 | <i>Not Blinking (thực tế: không hiện trên Website trạng thái này)</i> |
| | | 1 | Blinking |
| led_state | Cho biết đèn LED có đang tắt hay không. | 0 | Off |
| | Cho biết đèn LED có đang sáng hay không. | 1 | On |

Bảng 5: Bảng mô tả ý nghĩa của các trường dữ liệu gửi từ Controller đến Web Service

2.5.3 Truyền dữ liệu, tín hiệu từ Web Service đến Controller

- Cấu trúc:

```
1 "<switch_mode_duration>,<is_push_button>\n"
```

- Mỗi thông tin cách nhau bởi dấu phẩy, kết thúc chuỗi thông tin gửi bằng dấu “\n”

| Trường dữ liệu | Ý nghĩa | Giá trị nhập trên Website | Giá trị gửi |
|----------------------|---|-----------------------------------|---------------------|
| switch_mode_duration | Thời gian chờ từ lần nhấn nút cuối cùng để hệ thống tự động chuyển từ Manual Mode sang Auto Mode, có thể được người dùng cài đặt trên Website. Lưu ý: nếu đặt giá trị là 0, điều kiện timeout này sẽ không được áp dụng. Đơn vị: millisecond. | 0 đến 4,294,967,295 (millisecond) | 0 đến 4,294,967,295 |
| is_push_button | Button trên Website có đang được nhấn hay không để bật, tắt đèn LED của hệ thống. | Đang không nhấn | 0 |
| | | Đã nhấn | 1 |

Bảng 6: Bảng mô tả ý nghĩa của các trường dữ liệu gửi từ Web Service đến Controller

3 Đề xuất các giải pháp

| Component /Module | Giải pháp 1 | Giải pháp 2 | Giải pháp 3 | Quyết định giải pháp |
|-------------------|--|--|---|--|
| Controller | Arduino Uno R3 ATmega328 - Chi phí: ~100.000 VND - Dễ lập trình và mở rộng - Phù hợp cho các hệ thống cơ bản | Raspberry Pi 3B+ - Chi phí: >1.890.000 VND - Có khả năng xử lý cao hơn - Hỗ trợ hệ điều hành - Phù hợp cho các ứng dụng phức tạp | ESP32 - Chi phí: ~150.000 VND - Tích hợp sẵn Wifi và Bluetooth - Phù hợp cho IoT và hệ thống có kết nối mạng | Arduino Uno R3 ATmega328 Đáp ứng đủ yêu cầu, chi phí hợp lý, dễ sử dụng. |
| Web Service | Local Host (Flask, HTML, CSS) - Triển khai cục bộ trên máy tính - Chi phí: Miễn phí - Thích hợp cho giai đoạn thử nghiệm | Cloud Hosting - Dùng nền tảng đám mây (AWS, Azure) - Chi phí: phí duy trì hàng tháng - Khả năng mở rộng tốt - Truy cập từ xa tiện lợi | Raspberry Pi làm Server - Kết nối trực tiếp với cảm biến - Chi phí thiết lập cao hơn - Phù hợp cho các hệ thống nhỏ gọn, tự chủ | Local Host (Flask, HTML, CSS) Chi phí thấp và đáp ứng được yêu cầu cơ bản của đồ án. |

| Component /Module | Giải pháp 1 | Giải pháp 2 | Giải pháp 3 | Quyết định giải pháp |
|-------------------|--|---|---|---|
| Pushbutton | 6x6x5 mm Tactile Push Button <ul style="list-style-type: none"> - Chi phí thấp - Đơn giản, dễ lắp đặt | Công tắc cảm ứng chạm <ul style="list-style-type: none"> - Chi phí cao hơn (khoảng 20.000 VNĐ) - Tạo trải nghiệm hiện đại - Có thể nhúng vào thiết kế đẹp mắt | Công tắc nhấn IP67 <ul style="list-style-type: none"> - Chống nước và bụi, phù hợp môi trường ngoài trời - Chi phí cao hơn (khoảng 20.000 VNĐ) - Độ bền cao | 6x6x5 mm Tactile Push Button <p>Đơn giản, chi phí thấp, dễ sử dụng.</p> |
| Photoresistor | Lm393 Optical Photosensitive LDR <ul style="list-style-type: none"> - Chi phí thấp (khoảng 20.000 VNĐ) - Độ nhạy cao với ánh sáng - Dễ kết nối với Arduino | Cảm biến ánh sáng BH1750 <ul style="list-style-type: none"> - Chi phí cao hơn (~40.000 VNĐ) - Độ chính xác cao hơn - Kết nối qua giao thức I2C | Cảm biến TSL2561 <ul style="list-style-type: none"> - Chi phí khoảng 140.000 VNĐ - Đo ánh sáng với độ chính xác cao, dải rộng - Phù hợp cho các dự án cần đo ánh sáng chính xác | Lm393 Optical Photosensitive LDR <p>Đáp ứng yêu cầu chi phí thấp và dễ sử dụng trong môi trường ánh sáng đơn giản.</p> |

| Component /Module | Giải pháp 1 | Giải pháp 2 | Giải pháp 3 | Quyết định giải pháp |
|-------------------|--|--|---|--|
| LED | 5mm Color LED - Đơn giản, dễ lắp đặt - Chi phí thấp (~9.000 VNĐ/30 đèn) | LED RGB - Đổi màu, tạo hiệu ứng - Chi phí cao hơn (~16.000 VNĐ) - Phù hợp nếu cần thay đổi màu sắc | Grove - LED Button - Có tích hợp sẵn kèm theo với Push-button. - Tuổi thọ sử dụng cao hơn (100.000 lần bật tắt) - Chi phí cao hơn (~66.000 VNĐ) | 5mm Color LED Chi phí thấp, đáp ứng yêu cầu chiếu sáng cơ bản. |

Bảng 7: Danh sách đề xuất các giải pháp cho hệ thống

Bảng 7 đưa ra mô tả các giải pháp khả thi cho các Component/Module của hệ thống. Việc đa dạng hóa các giải pháp giúp hệ thống có nhiều phương án dự phòng, nhiều phương án phát triển hệ thống theo các hướng đi mới hiệu quả hơn và có khả năng khắc phục được những hạn chế còn tồn đọng (nếu có). Cần căn cứ vào Đặc tả yêu cầu và những ràng buộc, hạn chế cũng như bản thiết kế hệ thống để có thể đưa ra được *Quyết định giải pháp* hợp lý nhất cho quá trình phát triển hệ thống ở thời điểm hiện tại. Các giải pháp có chi phí thấp, dễ mua, dễ thiết lập và sử dụng, có phiên bản giả lập, đáp ứng đầy đủ ở mức cơ bản mà đặc tả yêu cầu đề ra là những giải pháp được chọn hiện tại dựa trên cơ sở như còn gặp nhiều hạn chế về kinh phí và trình độ hiện tại của lập trình viên hệ thống.

4 Kế hoạch kiểm thử

4.1 Kiểm tra từng thành phần

| Thành phần | Kiểm tra |
|--|---|
| Controller (Arduino Uno R3 ATmega328) | <ul style="list-style-type: none">- Kiểm tra Controller khởi tạo đúng và tất cả các chân (pins) được cấu hình theo thiết kế.- Kiểm tra giao tiếp nối tiếp dữ liệu Serial giữa Arduino và máy tính thông qua USB Port để đảm bảo khả năng truyền dữ liệu chính xác. |
| Web Service (Local Host với Flask, HTML, CSS) | <ul style="list-style-type: none">- Kiểm tra các route của Flask xử lý yêu cầu và trả về kết quả như mong đợi.- Giả lập dữ liệu gửi từ Arduino và kiểm tra dữ liệu được nhận và hiển thị chính xác. Đồng thời xuất kết quả gửi từ Flask trên terminal để đảm bảo tính chính xác của dữ liệu gửi đến Arduino. |
| Pushbutton (6x6x5 mm Miniature Micro Momentary Tactile Tact Push Button) | <ul style="list-style-type: none">- Xác nhận nút nhấn ghi nhận trạng thái thay đổi (nhấn và thả) và gửi dữ liệu đến Controller. |
| Photoresistor (Lm393 Optical Photosensitive LDR Light Sensor Module) | <ul style="list-style-type: none">- Đo độ nhạy sáng và đảm bảo đọc giá trị analog chính xác trên chân A0.- Kiểm tra khả năng của quang trở để phân biệt giữa ngưỡng sáng ngày và đêm. (Đặc biệt tại môi trường dự kiến triển khai hệ thống) |
| LED (5mm Color LED) | <ul style="list-style-type: none">- Kiểm tra đèn LED phản hồi tín hiệu kỹ thuật số từ chân D7 cho các trạng thái BẬT, TẮT và NHẮP NHÁY.- Kiểm tra độ sáng/độ rõ của đèn LED trong điều kiện bình thường. |

Bảng 8: Kiểm tra các thành phần hệ thống

Quy trình kiểm tra các thành phần hệ thống được mô tả ở bảng 8 phản ánh sự cần thiết trong việc kiểm thử độc lập từng thiết bị để xác định khả năng sử dụng hiện tại của các thiết bị cũng như tính đúng đắn của dữ liệu mà các thiết bị này trả về. Đây là bước quan trọng trong việc xác định lỗi (nếu có) đến từ thiết bị nào để có phương án khắc phục phù hợp về phần mềm và phần cứng. Ngoài ra, việc kiểm thử này cũng giúp lập trình viên hệ thống có thể cập nhật thông số môi trường sẽ vận hành hệ thống để có sự hiệu chỉnh thông số trong phần mềm một cách phù hợp nhất. Việc kiểm tra từng thành phần này sẽ cần được triển khai cho tất cả phiên bản của hệ thống (trong trường hợp này là phiên bản giả lập và phiên bản triển khai thực tế).

4.2 Kiểm tra tích hợp

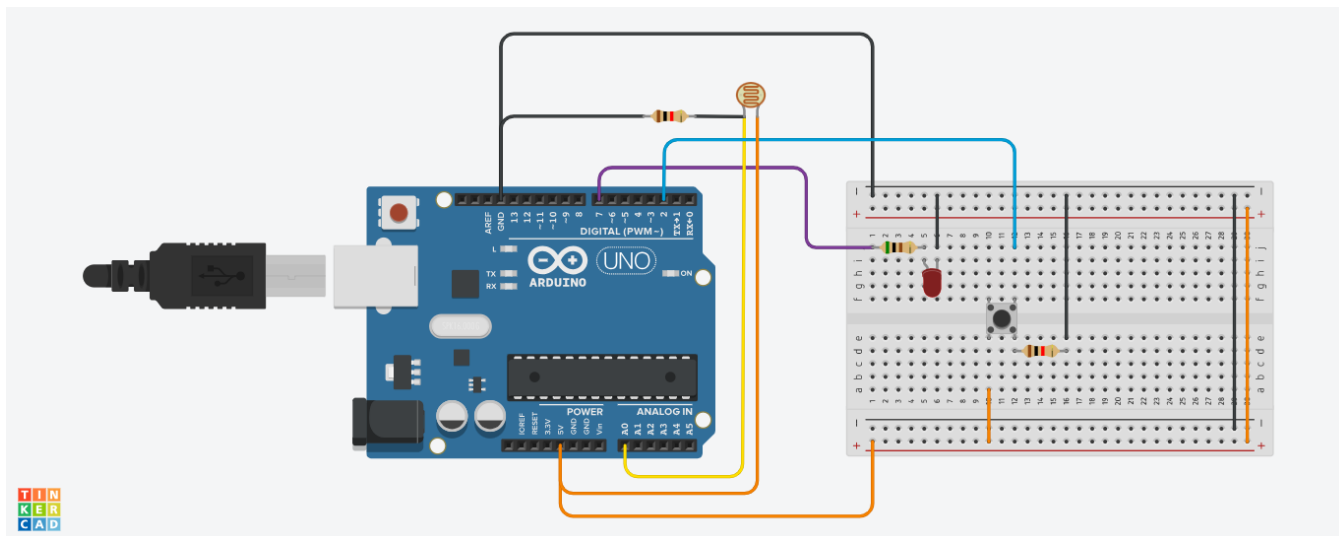
| Bước | Tích hợp | Kiểm tra |
|------|--|--|
| 1 | Arduino + Dịch vụ Web | - Kiểm tra phản hồi của dịch vụ web khi Arduino gửi dữ liệu giả lập từ nút nhấn, cảm biến ánh sáng và các dữ liệu kèm theo khác. |
| 3 | Arduino + Dịch vụ Web + Quang trở | - Thay đổi điều kiện ánh sáng xung quanh để kích hoạt quang trở và kiểm tra sự thay đổi trạng thái trên dịch vụ web. |
| 4 | Arduino + Dịch vụ Web + Quang trở + LED | - Kiểm tra đèn LED BẬT, TẮT hoặc NHẮP NHÁY dựa trên đầu vào từ cảm biến ánh sáng, được ghi nhận trên dịch vụ web. |
| 4 | Arduino + Dịch vụ Web + Quang trở + LED + Nút nhấn | - Kiểm tra đèn LED BẬT, TẮT hoặc NHẮP NHÁY dựa trên đầu vào từ nút nhấn, được ghi nhận trên dịch vụ web. |
| 5 | Tích hợp toàn bộ hệ thống | <ul style="list-style-type: none"> - Thực hiện kiểm tra toàn diện cho cả hệ thống khi tất cả các thành phần hoạt động, đảm bảo hành vi FSM khớp với thiết kế (TẮT, BẬT, NHẮP NHÁY). - Xác nhận trạng thái của đèn LED thay đổi chính xác dựa trên cả nút nhấn vật lý và nút nhấn trên Website cũng như dựa trên đầu vào từ cảm biến ánh sáng, và tất cả dữ liệu được ghi lại trên dịch vụ web. - Xác nhận các thông tin khác hiển thị trên Website được nhận từ Arduino khớp với thực tế. |

Bảng 9: Các bước tích hợp và kiểm tra hệ thống

Quy trình kiểm tra tích hợp hệ thống thể hiện qua bảng 9 nhằm kiểm tra xem việc vận hành giữa các thành phần với nhau đúng với đặc tả yêu cầu và mô tả hệ thống đặt ra. Quy trình này đảm bảo rằng tất cả kịch bản kiểm thử sẽ bao phủ càng toàn diện càng tốt các yêu cầu mà đặc tả đã đề cập đến. Việc kiểm tra tích hợp này sẽ cần được triển khai cho tất cả phiên bản của hệ thống (trong trường hợp này là phiên bản giả lập và phiên bản triển khai thực tế).

5 Kết quả triển khai

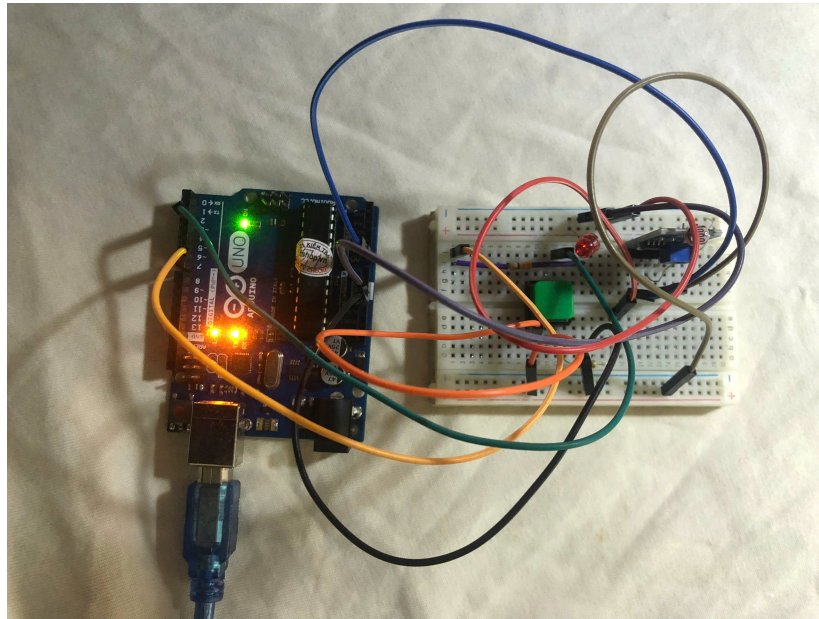
5.1 Giả lập hệ thống trên Tinkercad



Hình 5: Triển khai mạch Arduino trên Tinkercad

Hệ thống được giả lập trên **Tinkercad** được mô tả như hình 5 này sử dụng Arduino Uno R3 và bao gồm các thành phần: cảm biến ánh sáng (LDR), đèn LED và nút nhấn. Cảm biến ánh sáng được kết nối với cổng Analog In A0 của Arduino thông qua một điện trở cho phép Arduino đo mức điện áp thay đổi theo cường độ ánh sáng xung quanh. Đèn LED được điều khiển thông qua cổng Digital D7 của Arduino, có thể phản hồi lại theo tín hiệu từ cảm biến ánh sáng hoặc tín hiệu từ nút nhấn. Nút nhấn cho phép người dùng chuyển chế độ vận hành hệ thống từ AutoMode sang ManualMode và điều khiển đèn LED bật/tắt một cách thủ công. Thông qua đó ta thấy hệ thống phiên bản giả lập trên Tinkercad giải quyết tương đối đầy đủ những yêu cầu mà đặc tả và bản thiết kế hệ thống đã đề ra. Ngoài ra, người dùng có thể mở cửa sổ Serial Monitor của Tinkercad để giả lập sự truyền nhận dữ liệu giữa Arduino Uno R3 và Web Service thông qua các câu lệnh như `Serial.print()`, `Serial.available()` và `Serial.readStringUntil('\n')`.

5.2 Triển khai hệ thống trên thiết bị thật



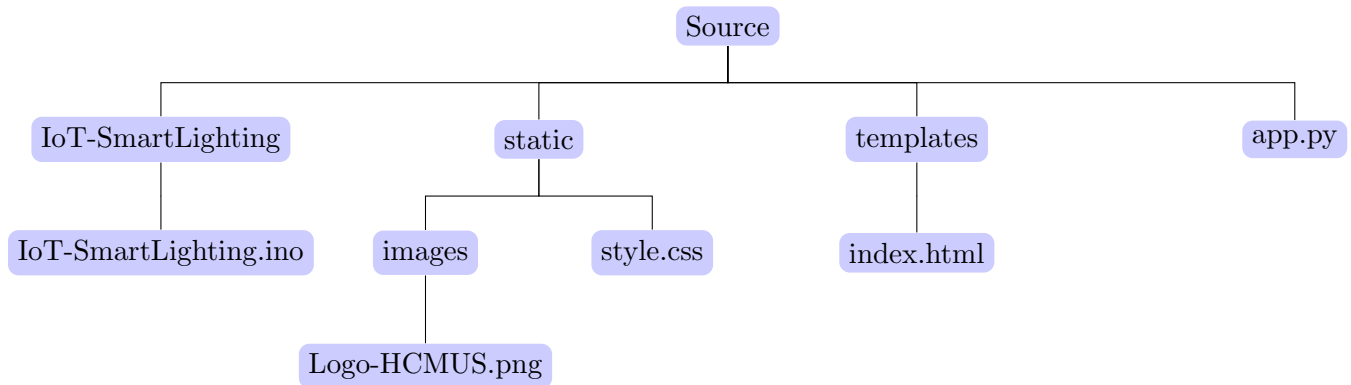
Hình 6: Triển khai hệ thống thực tế



Hình 7: Triển khai hệ thống thực tế tích hợp Web Service

Hệ thống thực tế trên hình 6 và hình 7 được bố trí và lắp đặt tương tự như triển khai giả lập trên Tinkercad ở hình 5. Mạch được cung cấp nguồn bằng USB Port nối với Laptop và sẽ truyền nhận dữ liệu giữa mạch và Laptop thông qua cổng này.

5.3 Triển khai Website



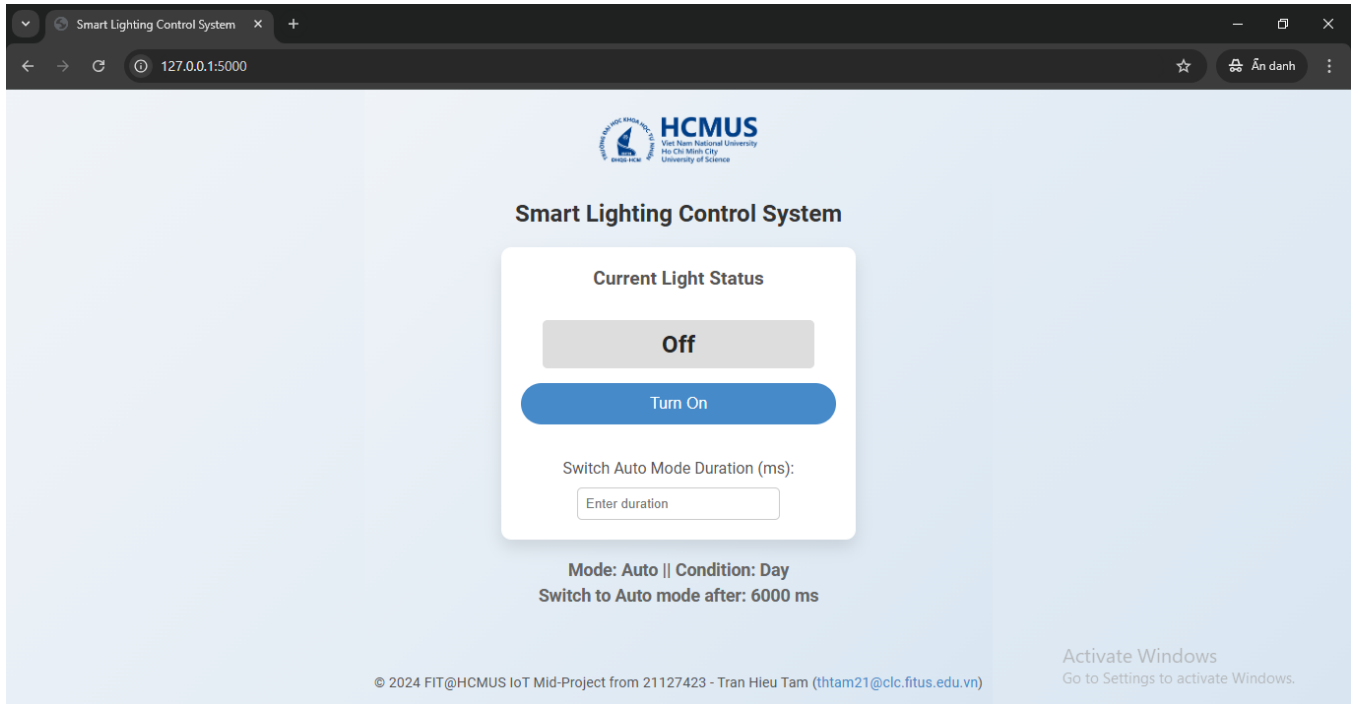
- **Bước 1:** Cài đặt Python, Flask và các thư viện liên quan
- **Bước 2:** Lập trình và tổ chức các file sau
 - Source\IoT-SmartLighting\IoT-SmartLighting.ino: Đây là file mã nguồn cho Arduino, được viết bằng ngôn ngữ C/C++ trong môi trường Arduino IDE. File này chứa các đoạn mã để điều khiển các phần cứng trong hệ thống Smart Lighting, như đọc dữ liệu từ cảm biến ánh sáng, điều khiển đèn LED, và xử lý nút bấm.
 - Source\static\images\Logo-HCMUS.png: Hình ảnh hiển thị trên website.
 - Source\static\style.css: File CSS này chứa các quy tắc và định dạng giao diện cho trang web. Nó quy định cách thức hiển thị các thành phần như màu sắc, phông chữ, bố cục, và phong cách của trang web.
 - Source\templates\index.html: Đây là file HTML chính để tạo giao diện người dùng cho ứng dụng web. File này chứa cấu trúc của trang web, nơi người dùng có thể tương tác với hệ thống IoT để điều khiển cũng như theo dõi trạng thái hiện tại của đèn LED.
 - Source\app.py: Đây là file Python chứa mã nguồn cho web server của ứng dụng, được xây dựng bằng Flask framework. File này xử lý các yêu cầu từ trình duyệt web, giao tiếp với Arduino để nhận dữ liệu cảm biến và gửi lệnh điều khiển đến các thiết bị. Nó là thành phần quan trọng giúp kết nối giữa người dùng và phần cứng của hệ thống IoT.
- **Bước 3:** Sử dụng Arduino IDE để nạp code .ino vào Controller.
- **Bước 4:** Kết nối Arduino (đã được nạp code trước) với Laptop thông qua USB Port. Lưu ý, không mở ứng dụng khác sử dụng chung port với Flask đang chạy.

- **Bước 5:** Chạy localhost thông qua cmd từ folder Source

```
1 flask --app app run
```

- **Bước 6:** Xem trên Localhost và Terminal

```
1 http://127.0.0.1:5000/
```

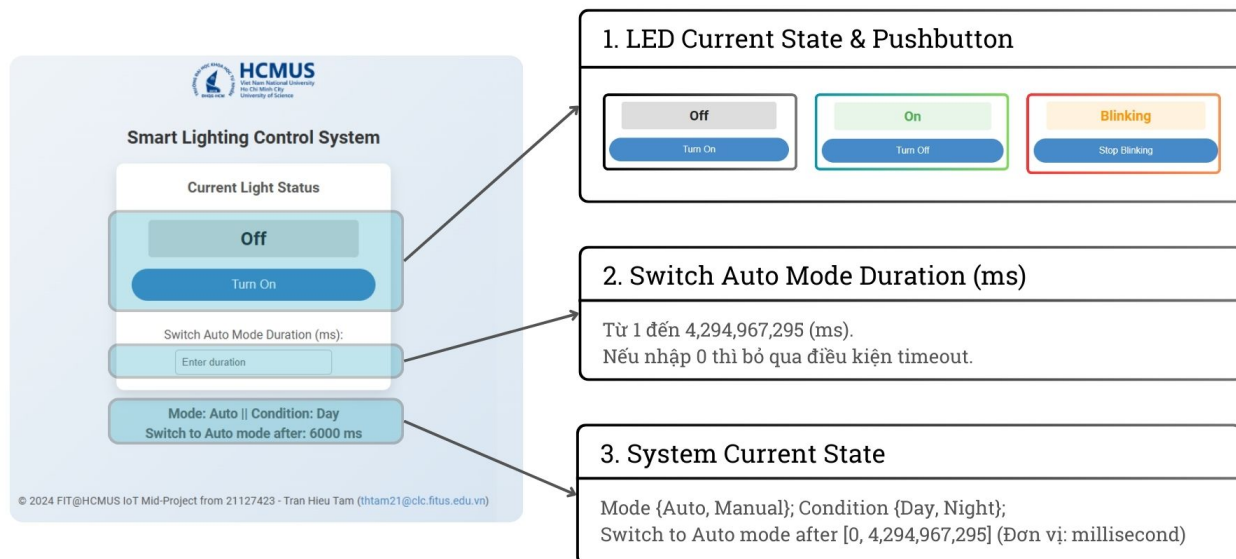


Hình 8: Giao diện Web Service sau khi kết nối với Web Server thành công

5.4 Giới thiệu chức năng website

Website gồm có các chức năng sau:

- Theo dõi trạng thái đèn bao gồm Off, On và Blinking.
- Theo dõi chế độ vận hành hệ thống hiện tại (Auto Mode hay Manual Mode), trạng thái ánh sáng môi trường xung quanh hệ thống (Day hay Night), timeout từ Manual Mode sang Auto Mode bao nhiêu Millisecond (nếu hiển thị 0 thì hệ thống sẽ bỏ qua điều kiện timeout này khi vận hành)
- Khi nhấn vào nút bấm trên Website thì nó vận hành giống như nút bấm vật lý thực tế của hệ thống (thay đổi trạng thái hiện tại của đèn LED trong Manual Mode từ On sang Off, Off sang On, Blinking sang Off).
- Khi nhập giá trị vào Switch to Auto Mode Duration và Enter thì giá trị sẽ được gửi cho hệ thống theo đơn vị Millisecond, không tác động đến trạng thái nút bấm trên Website. Nếu nhập vào giá trị 0 thì hệ thống mặc nhiên bỏ điều kiện này khi vận hành.



Hình 9: Các thành phần chính của Website

5.5 Release

1 21127423, Tran Hieu Tam, <https://www.youtube.com/watch?v=MielNnFK-BU>,
<https://drive.google.com/drive/folders/1Cm12i8CtkhcVk8lisDeS54gof4zHMSak?usp=sharing>

Project Information

- **Release Version:** 1.0
- **Release Date:** November 11, 2024
- **Author:** Trần Hiếu Tâm
- **Contact Email:** thtam21@clc.fitus.edu.vn
- **Location:** Ho Chi Minh City, Viet Nam
- **GitHub Repository:** <https://github.com/HieuTam/HCMUS-IoT-SmartLightingControlSystem>

Copyright

Faculty of Information Technology, VNUHCM-University of Science



Tài liệu

- [1] Arduino. Arduino Home Page, 2024. <https://www.arduino.cc/>.
- [2] krshelar. Light Sensor using Arduino, 2020. <https://www.tinkercad.com/things/ek9dwFldgEI-light-sensor-using-arduino>.
- [3] Pallets. Flask User's Guide, 2010. <https://flask.palletsprojects.com/>.
- [4] Công ty cổ phần công nghệ giáo dục F8. HTML CSS từ Zero đến Hero, 2024. <https://fullstack.edu.vn/courses/html-css>.
- [5] W3Schools. CSS Tutorial, 2024. <https://www.w3schools.com/css/>.
- [6] W3Schools. HTML Tutorial, 2024. <https://www.w3schools.com/html/>.
- [7] Nguyễn Đức Hoàng Hạ. An Introduction to Programming the Internet of Things (IoT) Lecturer Slides, 2024. <https://drive.google.com/drive/folders/1Vo7YCZSkvnj8ns3Y5iACijcrUE82hDja>.