|  |
| --- |
| **ĐẠI HỌC QUỐC GIA HÀ NỘI**  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**    **Lê Xuân Phú**  **NGHIÊN CỨU GIAO THỨC VÀ GIAO DIỆN GIAO TIẾP ẢO CHO ĐIỀU KHIỂN CÁC THIẾT BỊ IOT CỦA NGÔI NHÀ THÔNG MINH**  **KHÓA LUẬN TỐT NGHIỆP HỆ CHẤT LƯỢNG CAO**  **Ngành: Khoa học máy tính CLC**  **HÀ NỘI – 2021**  **ĐẠI HỌC QUỐC GIA HÀ NỘI**  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**  **Lê Xuân Phú**  **NGHIÊN CỨU GIAO THỨC VÀ GIAO DIỆN GIAO TIẾP ẢO CHO ĐIỀU KHIỂN CÁC THIẾT BỊ IOT CỦA NGÔI NHÀ THÔNG MINH**  **KHÓA LUẬN TỐT NGHIỆP HỆ CHẤT LƯỢNG CAO**  **Ngành: Khoa học máy tính CLC**  **Cán bộ hướng dẫn: TS. Ma Thị Châu**  **HÀ NỘI – 2021**  **VIETNAM NATIONAL UNIVERSITY, HANOI**  **UNIVERSITY OF ENGINEERING AND TECHNOLOGY**  **Lê Xuân Phú**  **RESEARCH ARTICLE ON VIRTUAL INTERFACE AND PROTOCOL FOR CONTROLLING SMART HOME IOT DEVICES**  **GRADUATE THESIS FOR HIGH QUALITY EDUCATIONAL SYSTEM**  **Major: Computer Science**  **Instructor: Doctor Ma Thị Châu**  **HÀ NỘI – 2021** |

**TÓM TẮT**

Hiện nay, với sự phát triển vượt bậc của công nghệ, những căn nhà thông minh đã trở nên rất phổ biến với con người. Tuy vậy, những ngôi nhà thông minh ấy còn có nhiều thiếu sót nếu đưa vào sử dụng cho những người bị khuyết tật nặng. Chính vì vậy, em cùng với nhóm đã nghiên cứu và phát triển một hệ thống cho nhà thông minh đặc biệt nhắm vào nhóm những người trên. Trong khóa luận này, em muốn trình bày thiết kế của giao diện giao tiếp ảo cho hệ thống điều khiển các thiết bị IoT của ngôi nhà thông minh trên cùng với đó là giao thức sử dụng để điều khiển các thiết bị đó.

**Từ khóa:** Smarthome GUI, IoT Protocol.

**ABSTRACT**

Currently, with the great development of technology, smart homes have become very popular with people. However, those smart homes still have many shortcomings if put into use for people with severe disabilities. That's why, along with my team, I have researched and developed a system for smart homes specifically targeting this group of people. In this thesis, I want to present the design of the virtual interface for the control system of IoT devices of the smart home along with the protocol used to control those devices.

**Keyword:** Smarthome GUI, IoT protocol.

**LỜI CAM ĐOAN**

Em xin cam đoan đề tài: “Nghiên cứu giao thức và giao diện giao tiếp ảo cho điều khiển các thiết bị IoT của ngôi nhà thông minh” là một công trình nghiên cứu độc lập không có sự sao chép của người khác. Đề tài là một sản phẩm mà em đã nỗ lực nghiên cứu trong quá trình học tập tại trường. Trong quá trình viết bài có sự tham khảo một số tài liệu có nguồn gốc rõ ràng, dưới sự hướng dẫn của Cô Ma Thị Châu. Em xin cam đoan nếu có vấn đề gì em xin chịu hoàn toàn trách nhiệm.

Hà Nội,

Tác giả khóa luận

Lê Xuân Phú

**LỜI CẢM ƠN**

Đầu tiên cho phép em gửi lời cảm ơn chân thành đến cô Ma Thị Châu đã đồng ý trở thành giảng viên hướng dẫn khóa luận tốt nghiệp cho em và tạo cơ hội để em có thể tham gia hoạt động trong phòng thí nghiệm tương tác người máy HMI hiện đại. Cảm ơn cô đã dành ra những thời gian quý báu để hướng dẫn em trong quá trình thực hiện khóa luận cũng như những lời khuyên bổ ích và góp ý chân thành.

Em cũng xin gửi lời cảm ơn sâu sắc đến các thầy, các cô của Trường Đại học Công Nghệ đã tạo cảm hứng, giúp đỡ và giảng dạy cho em các kiến thức quý báu trong khoảng thời gian em học tập tại trường.

Em cũng xin cảm ơn đến bạn bè, anh chị tại phòng thí nghiệm tương tác người máy HMI trường Đại học Công Nghệ đã đồng hành và hỗ trợ em trong suốt quá trình phát triển dự án.

Em xin được chân thành cảm ơn!

**MỤC LỤC**

[**Chương 1.** **Giới thiệu chung** 1](#_Toc92664522)

[**Chương 2.** **Giới thiệu các sản phẩm hỗ trợ hiện có và công nghệ sử dụng** 4](#_Toc92664523)

[1. QT [1]: 4](#_Toc92664524)

[2. C++ [2]: 5](#_Toc92664525)

[3. Visual Studio [3]: 6](#_Toc92664526)

[4. Giao thức MQTT (Message Queue Telemetry Transport) [4]: 7](#_Toc92664527)

[5. Tobii eye tracking [5]: 8](#_Toc92664528)

[**Chương 3.** **Phân tích thiết kế yêu cầu** 11](#_Toc92664529)

[1. Giao diện người dùng 11](#_Toc92664530)

[**1.1.** **Mô tả tổng quan [6]:** 11](#_Toc92664531)

[**1.2.** **Màu sắc [7]:** 11](#_Toc92664532)

[**1.3.** **Ngôn ngữ:** 14](#_Toc92664533)

[**1.4.** **Bố cục [8]:** 14](#_Toc92664534)

[**1.5.** **Kích thước:** 17](#_Toc92664535)

[2. API [9]: 18](#_Toc92664536)

[3. Tobii Eye Tracker [11]: 19](#_Toc92664537)

[**Chương 4.** **Thiết kế giao diện điều khiển** 21](#_Toc92664538)

[1. Giao diện nhà thông minh: 21](#_Toc92664539)

[2. Giao diện tùy chỉnh: 24](#_Toc92664540)

[3. Giao diện điều khiển đèn: 26](#_Toc92664541)

[4. Giao diện điều khiển quạt: 28](#_Toc92664542)

[5. Giao diện điều khiển vô tuyến: 30](#_Toc92664543)

[6. Giao diện điều khiển điều hòa: 34](#_Toc92664544)

[**Chương 5.** **Thiết kế API** 37](#_Toc92664545)

[1. API lấy danh sách thiết bị 37](#_Toc92664546)

[**1.1.** **Thử nghiệm các thiết kế khả thi** **Error! Bookmark not defined.**](#_Toc92664547)

[**1.1.1.** **Sử dụng giao thức MQTT:** 37](#_Toc92664548)

[**1.1.2.** **Sử dụng giao thức HTTP:** 37](#_Toc92664549)

[**1.2.** **Kết quả thiết kế** 39](#_Toc92664550)

[2. API điều khiển thiết bị 40](#_Toc92664551)

[**2.1.** **Thử nghiệm các thiết kế khả thi** **Error! Bookmark not defined.**](#_Toc92664552)

[**2.2.** **Kết quả thiết kế** 40](#_Toc92664553)

[**1.1.** **Điều khiển đèn** 40](#_Toc92664554)

[**1.2.** **Điều khiển quạt** 41](#_Toc92664555)

[**1.3.** **Điều khiển vô tuyến** 41](#_Toc92664556)

[**1.4.** **Điều khiển điều hòa** 41](#_Toc92664557)

[**Chương 6.** **Kết luận và phương hướng cải tiến** 42](#_Toc92664558)

[1. Kết luận: 42](#_Toc92664559)

[2. Phương hướng cải tiến: 42](#_Toc92664560)

**DANH MỤC HÌNH ẢNH**

Hình 1-1: Mô hình tổng thể của hệ thống

Hình 2-1: Kiến trúc của giao thức MQTT

Hình 2-2: Thiết bị theo dõi mắt Tobii

Hình 3-1: Đắm mình trong một loại màu thuần túy có ảnh hưởng đến thể chất và tinh thần

Hình 3-2: Giao diện được biểu diễn dưới dạng lưới

Hình 3-3: Ứng dụng kiến trúc MQTT trong thực tế

Hình 3-4: Cách bố trí vị trí sử dụng cho Tobii Eye Tracker

Hình 4-1: Giao diện nhà thông minh

Hình 4-2: Nút chọn thiết bị đèn được đăng ký trên server với tên “Đèn 1”

Hình 4-3: Nút chọn thiết bị quạt được đăng ký trên server với tên “Quạt 1”

Hình 4-4: Nút chọn thiết bị vô tuyến được đăng ký trên server với tên “Tivi 1”

Hình 4-5: Nút chọn thiết bị điều hòa được đăng ký trên server với tên “Điều hòa 1”

Hình 4-6: Nút quay trở lại giao diện chính

Hình 4-7: Nút chuyển sang giao diện tùy chỉnh

Hình 4-8: Giao diện tùy chình

Hình 4-9: Nút chuyển sang ngôn ngữ khác

Hình 4-10: Nút chuyển sang màu sắc hiển thị khác

Hình 4-11: Nút chuyển về giao diện nhà thông minh

Hình 4-12: Giao diện điều khiển đèn

Hình 4-13: Nút bật đèn

Hình 4-14: Nút tắt đèn

Hình 4-15: Giao diện điều khiển quạt

Hình 4-16: Nút chuyển sang mức gió đầu tiên

Hình 4-17: Nút chuyển mức gió thứ hai

Hình 4-18: Nút chuyển sang mức gió thứ ba

Hình 4-19: Giao diện điều khiển vô tuyến

Hình 4-20: Nút chuyển sang kênh kế tiếp

Hình 4-21: Nút chuyển sang kênh trước đó

Hình 4-22: Nút tăng âm lượng

Hình 4-23: Nút giảm âm lượng

Hình 4-24: Bàn phím số

Hình 4-25: Giao diện điều khiển điều hòa

Hình 4-26: Nút tăng nhiệt độ

Hình 4-27: Nút giảm nhiệt độ

Hình 4-28: Nút bật/tắt cánh điều gió

Hình 5-1: Format của dữ liệu nhận từ server

Hình 5-2: Format của dữ liệu gửi lên server

**BẢNG KÝ HIỆU CHỮ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **Chữ viết tắt** | **Ý nghĩa** |
| 1 | IOT | Internet of Things |
| 2 | HMI | Human Machine Interaction |
| 3 | MQTT | Message Queue Telemetry Transport |
| 4 | API | Application Programming Interface |
| 5 | GUI | Graphic User Interface |
| 6 | IDE | Integrated Development Environment |
| 7 | HTTP | Hypertext Transfer Protocol |

1. **Giới thiệu chung**

Dưới sự phát triển mạnh mẽ của khoa học và công nghệ, cuộc sống của con người ngày càng được đáp ứng một cách đầy đủ. Vào những năm trở lại đây, có thể dễ dàng nhận thấy thế giới nói chung và Việt Nam nói riêng có nhiều thay đổi rõ rệt. Một trong số những thay đổi đó được thể hiện trong lĩnh vực chăm sóc sức khỏe hay còn gọi là lĩnh vực Y Tế.

Nói về vấn đề chăm sóc sức khỏe, một trong những tiêu chí quan trọng bậc nhất đó là kịp thời và tiện nghi. Bệnh nhân khi cảm thấy thoải mái vì được chăm sóc một cách đầy đủ sẽ không chỉ tạo ra sự chuyển biến tích cực về mặt sinh lý mà còn cả về mặt tinh thần.

Nhờ vào sự phát triển mạnh mẽ của Internet, đã có một giải pháp hoàn hảo cho vấn đề nêu trên. Giải pháp đó là IoT hay còn gọi là “Internet of Things”. IoT hay còn gọi là “Mạng lưới vạn vật kết nối Internet”, là một kịch bản của thế giới, khi mà mọi đồ vật cũng như con người được gắn một mã số định danh riêng, và tất cả đều có khả năng truyền tải, trao đổi thông tin cũng như dữ liệu qua một mạng duy nhất mà không cần đến sự tương tác trực tiếp giữa người với người hay người với máy tính.

IoT trong y tế hiện nay đã có rất nhiều ứng dụng như: Quản lý dữ liệu chăm sóc thời gian thực, Tăng mức độ quan tâm của người bệnh, Giảm chi phí cho sức khỏe, Phân tích dữ liệu y tế, Cảnh báo tình trạng sức khỏe kịp thời, Tạo các tiện ích cho các nhu cầu sử dụng khác nhau, etc.. Mặc dù vậy, vì chỉ mới được quan tâm từ những năm gần đây nên còn cần cải tiến và phát triển rất nhiều trong tương lai.

Nhận thấy nhu cầu về mặt nhân sự trong y tế đang ngày càng gia tăng trong tình hình đại dịch COVID-19 hiện nay, cần có giải pháp để thay thế sức lao động của con người bằng máy tính một cách hiệu quả. Vì vậy phát triển một hệ thống điều khiển các thiết bị thông dụng như đèn, quạt, điều hòa, vô tuyến một cách thông minh là vô cùng cần thiết. Hệ thống không chỉ giúp giảm thiểu nhân sự sử dụng trong ngành chăm sóc sức khỏe mà còn giúp người sử dụng cảm thấy thoải mái và chủ động hơn.

Em tham gia vào nhóm nghiên cứu về phát triển sản phẩm IoT của lab HMI, chúng em cùng nghiên cứu và phát triển hệ thống này

Mô hình hệ thống được phân thiết kế như sau:



*Hình 1-1: Mô hình tổng thể của hệ thống*

Trong đó, PC là nơi cài đặt các phần mềm xử lý bao gồm nhận dữ liệu, phân tích, tính toán và xử lý dữ liệu rồi truyền tải tới các thiết bị khác để thực hiện điều khiển chúng. Dữ liệu nhận vào có thể thông qua các thiết bị nhận tín hiệu âm thanh, điện não hoặc trực tiếp bằng thao tác tay.

Giao diện điều khiển sẽ giúp cho người dùng tương tác với hệ thống, các lệnh điều khiển thiết bị của người sử dụng sẽ được nhận bởi hệ thống thông qua giao diện này cũng như những thông tin, thông số cần theo dõi như nhiệt độ hiện tại, mức gió, etc.. của thiết bị cũng sẽ được hiển thị tại đây. Để tăng khả năng tương tác với hệ thống, ta có Eye tracker (thiết bị theo dõi tín hiệu mắt) và EEG Transfer (thiết bị theo dõi tín hiệu điện não) sẽ nhận các tín hiệu của người dùng để truyền vào hệ thống.

Router Wifi sẽ nhận nhiệm vụ duy trì kết nối giữa các thiết bị. Điều này rất quan trọng cho sự vận hành trơn tru của hệ thống nên router Wifi sẽ cần đạt yêu cầu về sự ổn định. Cũng có thể sử dụng mạng Ethernet cho một số kết nối cần thiết.

Công tắc thông minh kết nối với bộ định tuyến qua mạng wifi và giao tiếp với toàn bộ các thiết bị trong hệ thống. Bộ phận này sẽ đảm nhiệm chức năng bật/tắt, điều chỉnh các thiết bị quạt, đèn, điều hòa,.. qua phương thức điều phối nguồn điện.

Bộ điều khiển trung tâm được kết nối với hệ thống bằng mạng Wifi hoặc Ethernet có tác dụng quản lý phần cứng trong hệ thống, nó có thể thêm mới hoặc xóa bỏ các thiết bị được người dùng điều khiển.

Trong đó, em tập trung đi sâu vào phần giao diện điều khiển và cách nó tương tác với người dùng cũng như phần còn lại của hệ thống.

1. **Giới thiệu các sản phẩm hỗ trợ hiện có và công nghệ sử dụng**
2. **QT [1]:**

Là một công cụ hỗ trợ giúp tạo ra các giao diện đồ họa người dùng mạnh mẽ. Ứng dụng tạo ra bởi QT có khả năng chạy trên đa nền tảng như Windows, Mac OS, Linux, Symbian, Android,… Không chỉ vậy, QT còn có framework hoàn chỉnh cho đồ họa 2D và 3D, network, XML, SQL,... Do đó, QT được xử dụng rộng rãi bởi các công ty lớn. Một số ứng dụng nổi tiếng đã được tạo ra bởi QT như: Autodesk Maya, Skype, VirtualBox, Adobe Photoshop Elements, Mathematica, VLC Media Player, KDE...

QT thiết kế một mô hình đối tượng riêng, chúng có các đặc điểm tính chất cực kỳ hữu dụng trong lập trình như:

* + - Cung cấp một cơ chế mạnh mẽ để giao tiếp giữa các đối tượng một cách liền mạch gọi là tín hiệu (signal) và khe cắm (slot).
    - Thuộc tính của đối tượng có thể truy vấn (queryable) và tự thiết kế (designable)
    - Thư viện sự kiện (event) lớn và có cung cấp các bộ lọc sự kiện (event filter).
    - Có khả năng dịch các chuỗi theo tiêu chuẩn chung
    - Bao gồm bộ hẹn giờ theo thời gian một cách chi tiết giúp tích hợp nhiều tác vụ trong trình điều khiển sự kiện của giao diện người dùng (GUI).
    - Cây đối tượng phân cấp rõ ràng và có thể truy vấn giữa các lớp kế thừa một cách tự nhiên.
    - Con trỏ riêng (QPointer) được thiết kế để tự động đặt thành 0 mỗi khi đối tượng được tham chiếu bị hủy, không như các con trỏ trong C++ bị treo vô định khi đối tượng được tham chiếu mất đi.
    - Có khả năng ép kiểu (cast) một cách linh hoạt giữa các thư viện.

1. **C++ [2]:**

C++ là ngôn ngữ lập trình rất phổ biến hiện nay, được phát triển như một cải tiến của ngôn ngữ lập trình C để bao gồm cả mô hình hướng đối tượng. Nó có thể theo kiểu lập trình mệnh lệnh và lập trình hướng đối tượng tùy theo nhu cầu sử dụng, đồng thời cũng là ngôn ngữ biên dịch.

C++ là một ngôn ngữ cấp trung, mang lại lợi thế cho việc lập trình các ứng dụng cấp thấp (trình điều khiển, chương trình cốt lõi - kernels) và thậm chí cấp cao hơn (các trò chơi, GUI, ứng dụng cho máy tính để bàn, v.v.). Cú pháp cơ bản và cấu trúc mã của cả C và C++ đều giống nhau.

Ngôn ngữ C++ có một số chức năng và điểm mấu chốt như sau:

* + - Đơn giản: Nó là một ngôn ngữ đơn giản theo nghĩa là các chương trình có thể được chia nhỏ thành các đơn vị và các phần một cách hợp lý, đồng thời cũng hỗ trợ thư viện phong phú và nhiều kiểu dữ liệu khác nhau.
    - Độc lập với máy nhưng không phụ thuộc vào nền tảng: Một tệp thực thi C++ không độc lập với nền tảng (Ví dụ như các chương trình được biên dịch trên hệ điều hành Linux sẽ không chạy trên hệ điều hành Windows), tuy nhiên chúng độc lập với máy.
    - Ngôn ngữ cấp trung: Nó là ngôn ngữ cấp trung vì chúng ta có thể thực hiện cả lập trình hệ thống (trình điều khiển, kernels, mạng, etc...) và có thể xây dựng các ứng dụng quy mô lớn (Media Players, Photoshop, Game Engines, etc...)
    - Hỗ trợ thư viện phong phú: C++ có hỗ trợ thư viện phong phú (Cả cấu trúc dữ liệu chuẩn hay tích hợp sẵn, thuật toán, etc...) cũng như thư viện của bên thứ 3 (ví dụ: thư viện QT) giúp phát triển ứng dụng một cách nhanh chóng và tốc độ.
    - Tốc độ thực thi: Các chương trình C++ vượt trội các ngôn ngữ về tốc độ thực thi. Bởi vì, nó là một ngôn ngữ đã được biên dịch, cũng như là cực kỳ theo thủ tục. Các ngôn ngữ mới hơn có thêm các tính năng mặc định được tích hợp sẵn như dọn dẹp các phần thừa thãi, nhập thông tin linh hoạt, etc... làm chậm quá trình thực thi chương trình nói chung. Vì không có chi phí xử lý bổ sung như trên, trong C ++, chương trình được thực thi rất nhanh.
    - Con trỏ và truy cập bộ nhớ trực tiếp: C++ cung cấp hỗ trợ con trỏ giúp người dùng thao tác trực tiếp với địa chỉ lưu trữ. Điều này giúp thực hiện lập trình cấp thấp (nơi người ta có thể cần phải có quyền kiểm soát rõ ràng đối với việc lưu trữ các biến).
    - Hướng đối tượng: Một trong những điểm mạnh nhất của ngôn ngữ này ngoài C. Hỗ trợ hướng đối tượng giúp C ++ tạo ra các chương trình có thể bảo trì và mở rộng, tức là có thể xây dựng các ứng dụng quy mô lớn. Tuy nhiên mã thủ tục (procedural code) trở nên khó duy trì khi kích thước mã tăng lên.
    - Ngôn ngữ biên dịch: C++ là ngôn ngữ biên dịch, góp phần vào khả năng đấy nhanh tốc độ biên dịch.

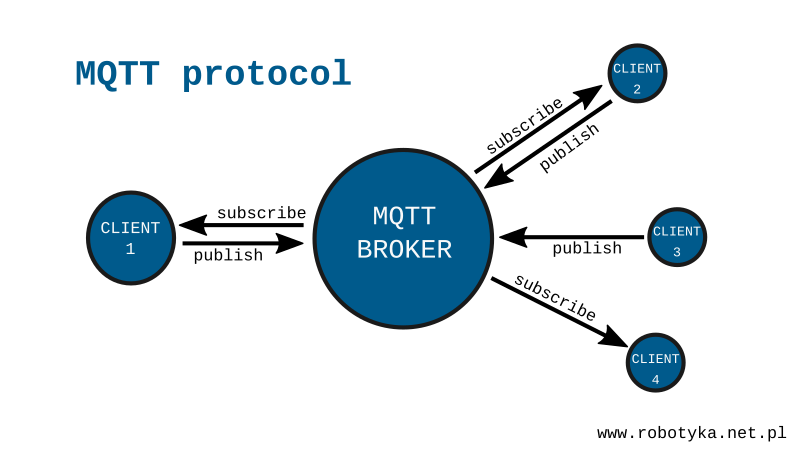
C++ có thể đóng nhiều vai trò trong một ứng dụng, điều này khiến nó được sử dụng một cách đa dạng như:

* + - Hệ điều hành và lập trình hệ thống như Linux-based OS, Ubuntu
    - Trình duyệt như là Chrome hay Firefox
    - Đồ họa và Game Engine như Blender, Photoshop, Unreal Engine
    - Bộ máy cơ sở dữ liệu như MySQL, MongoDB, Redis
    - Hệ thống đám mây, hệ thống phân tán

1. **Visual Studio [3]:**

Visual Studio là Môi trường phát triển tích hợp (IDE) được phát hành và cập nhật bởi Microsoft để phát triển GUI (Giao diện đồ họa người dùng), bảng điều khiển, ứng dụng web, ứng dụng di động, đám mây và dịch vụ web, etc... Với sự trợ giúp của IDE này, người dùng có thể viết mã được quản lý cũng như là ngôn ngữ máy. Nó sử dụng các nền tảng đa dạng của Microsoft software development như Windows store, Microsoft Silverlight và Windows API, etc... Nó không phải là IDE dành riêng cho một ngôn ngữ đặc thù nào cả nên người dùng có thể sử dụng IDE này để viết mã bằng ngôn ngữ khác nhau như C #, C ++, VB (Visual Basic), Python, JavaScript và nhiều ngôn ngữ khác. Visual Studio cung cấp sự hỗ trợ cho 36 ngôn ngữ lập trình khác nhau. Và nó có sẵn cho hệ điều hành Windows cũng như macOS.

Sự phát triển của Visual Studio: Phiên bản đầu tiên của VS (Visual Studio) được phát hành vào năm 1997, được đặt tên là Visual Studio 97 với số phiên bản 5.0. Phiên bản mới nhất của Visual Studio là 15.0 được phát hành vào ngày 7 tháng 3 năm 2017. Nó còn được gọi là Visual Studio 2017. Phiên bản .Net Framework được hỗ trợ trong Visual Studio mới nhất là 3.5 đến 4.7. Java đã được hỗ trợ trong các phiên bản Visual Studio cũ nhưng trong phiên bản mới nhất không cung cấp bất kỳ hỗ trợ nào cho ngôn ngữ Java.

1. **Giao thức MQTT (Message Queue Telemetry Transport) [4]:**

*Hình 2-1: Kiến trúc của giao thức MQTT*

MQTT là một giao thức truyền tải giữa Client và Server theo cơ chế subcribe và publish các gói tin. Nó nhẹ, mở, đơn giản và được thiết kế để dễ thực hiện. Những đặc điểm này khiến nó trở nên lý tưởng để sử dụng trong nhiều tình huống, bao gồm các môi trường hạn chế như để giao tiếp trong bối cảnh Máy với Máy (M2M) và Internet of Things (IoT), nơi yêu cầu code footprint nhỏ hoặc băng thông mạng bị hạn chế.

Giao thức MQTT được phát minh vào năm 1999 bởi Andy Stanford –Clark (IBM) và Arlen Nipper (Arcom, nay là Cirrus Link). Họ cần một giao thức để giảm thiểu lượng pin tiêu thụ và lượng băng thông tối thiểu cho kết nối. Hai nhà phát minh đã chỉ định một số yêu cầu cho giao thức tương lai:

* Thực hiện đơn giản
* Đảm bảo chất lượng của dịch vụ truyền dữ liệu
* Nhẹ và sử dụng băng thông hiệu quả
* Dữ liệu bất khả tri (data agnostic)
* Có khả năng nhận thức liên tục về các phiên

Không chỉ là một giao thức mạng với kích thước nhỏ, dung lượng nhẹ bên phía client, và sử dụng băng thông mạng một cách hiệu quả. Giao thức MQTT hỗ trợ đảm bảo vận chuyển các gói tin theo kiểu fire-and-forget nghĩa là chuyển đi và có thể quên chúng vì chúng sẽ tự tìm đến mục tiêu mong muốn. Trong giao thức này, các gói tin được tách khỏi ứng dụng. Mức độ và cách thức tách trong một ứng dụng sẽ phụ thuộc vào việc lập trình của phía MQTT Client cũng như phía MQTT server. Khi gói tin được vận chuyển, ứng dụng sẽ được giải phóng khỏi trạng thái kết nối với server cũng như trạng thái chờ gói tin gửi đến. Mô hình tương tác này giống như email, nhưng đã được tối ưu hóa để phù hợp trong lập trình ứng dụng. MQTT Clients tương thích hầu hết với các nền tảng hệ điều hành phổ biến hiện tại như Windows, Mac OS, Linux, Android, iOS,…

Lớp trừu tượng trong MQTT đặc tả rất đầy đủ và chi tiết về mọi thứ của MQTT. Là một giao thức nhị phân với trọng lượng rất nhẹ, và do chi phí gói đã được tối giản, MQTT vượt trội khi truyền dữ liệu qua dây nối so với các giao thức như HTTP. Một khía cạnh quan trọng khác của giao thức đó là MQTT cực kỳ dễ triển khai ở phía máy khách. Tính dễ sử dụng là một trong các trọng tâm phát triển của MQTT. Điều đó làm cho nó trở nên hoàn toàn phù hợp với các thiết bị hạn chế với nguồn lực hạn chế vào ngày nay.

1. **Tobii eye tracking [5]:**

Theo dõi mắt (eye tracking) là công nghệ cảm biến có thể phát hiện dáng điệu của một người và theo dõi những gì họ đang nhìn trong thời gian thực. Công nghệ này cho phép chuyển đổi chuyển động của mắt thành một luồng dữ liệu chứa thông tin như vị trí đồng tử, vectơ ánh nhìn cho mỗi mắt và điểm nhìn. Về cơ bản, công nghệ giải mã chuyển động của mắt và chuyển chúng thành thông tin chi tiết có thể được sử dụng trong nhiều ứng dụng hoặc như một phương thức đầu vào bổ sung.

Nói một cách đơn giản về phương thức hoạt động, hệ thống theo dõi mắt bao gồm một hoặc nhiều camera, một số nguồn sáng và khả năng tính toán. Cũng với đó là các thuật toán có khả năng chuyển nguồn cấp dữ liệu của máy ảnh (camera feed) thành các điểm dữ liệu với sự trợ giúp của học máy (machine learning) và xử lý hình ảnh nâng cao (advanced image processing). 

*Hình 2-2: Thiết bị theo dõi mắt Tobii*

Tobii là một công ty công nghệ cao của Thụy Điển đã nghiên cứu và phát triển công nghệ theo dõi mắt được một thời gian dài. Họ cho ra đời một sản phẩm theo dõi mắt của riêng họ có tên là Tobii eye tracker. Sản phẩm được áp dụng những công nghệ tiên tiến nhất về theo dõi mắt. Nó đã được thiết kế để hoạt động được cho phạm vi có sự hiện diện nhiều người nhất có thể, cho mọi môi trường làm việc và trên nhiều loại máy móc, thiết bị khác nhau – bao gồm máy tính xách tay, máy tính bảng, xe cộ, tai nghe VR và AR, cũng như các hệ thống y tế, giáo dục và đào tạo được thiết kế riêng phục vụ cho đúng mục đích.

Các giải pháp của tobii được xây dựng để phục vụ nhu cầu dữ liệu chi tiết của các nhà khoa học, giải quyết các yêu cầu về tính cơ động và dữ liệu nhỏ gọn có chọn lọc của các nhà phát triển ứng dụng thương mại, đồng thời cung cấp cho người dùng trải nghiệm hoàn hảo, duy trì quyền riêng tư của họ.

Mặc dù tương đối dễ dàng để phát triển một hệ thống theo dõi mắt hoạt động trong một môi trường được kiểm soát, nhưng việc tạo ra một hệ thống theo dõi phù hợp với mọi người, ở mọi nơi, mọi nguyên tắc của thiết bị thì không. Trong hơn hai thập kỷ, Tobii đã làm việc không ngừng để tạo ra một công nghệ đáp ứng nguyên tắc này và các đưa ra được các giải pháp:

* + - Hỗ trợ sự thay đổi từ hẹp nhất đến rộng nhất của hình dạng mắt, màu sắc và độ phản xạ của võng mạc
    - Có khả năng làm việc không bị ảnh hưởng bởi ánh sáng xung quanh, thiết bị mục tiêu và trường hợp sử dụng
    - Hỗ trợ điều chỉnh chuyển động của người dùng
    - Tuân thủ các tiêu chuẩn và hỗ trợ các nguyên tắc tinh gọn về truyền dữ liệu, xử lý và tính toán
    - Đề cao quyền riêng tư và tính toàn vẹn của người dùng
    - Có thể lặp lại trong sản xuất bền vững quy mô lớn

1. **Phân tích thiết kế yêu cầu**
2. **Giao diện người dùng**
   1. **Mô tả tổng quan [6]:**

Như được biết, tâm lý ảnh hưởng rất nhiều đến trạng thái cơ thể của một người. Trong một bài viết của Heathwise có nói, tinh thần và cơ thể là hai đồng minh mạnh mẽ. Cách chúng ta suy nghĩ ảnh hưởng đến cách chúng ta cảm nhận và cách chúng ta cảm nhận sẽ ảnh hưởng lại đến cách chúng ta suy nghĩ. Một ví dụ của kết nối tinh thần – cơ thể là cách chúng phản ứng với stress. Thường xuyên lo lắng và căng thẳng về công việc, tài chính hoặc các vấn đề khác có thể gây căng cơ, đau, nhức đầu và các vấn đề về dạ dày. Nó cũng có thể dẫn đến huyết áp cao hoặc các vấn đề nghiêm trọng khác. Mặt khác, cơn đau liên tục hoặc một vấn đề sức khỏe như bệnh tim có thể ảnh hưởng đến cảm xúc. Họ có thể trở nên trầm cảm, lo lắng và căng thẳng, điều này có thể ảnh hưởng đến cách họ điều trị, kiểm soát hoặc đối phó với bệnh tật của mình. Nhưng tâm trí cũng có thể có tác động tích cực đến sức khỏe. Có một cái nhìn tích cực về cuộc sống có thể giúp người ta xử lý tốt hơn cơn đau hoặc căng thẳng và sống khỏe mạnh hơn là một người ít hy vọng. Và thứ ảnh hưởng đến tâm lý con người nhiều nhất đó là những gì mà họ nhìn thấy. Chính vì vậy thiết kế giao diện cũng là một công việc đòi hỏi sự nghiên cứu kỹ lưỡng về tâm lý con người, làm sao để họ có được một tâm thái phù hợp với mong muốn đề ra. Trong trường hợp này, giao diện phải khiến người dùng cảm thấy thoải mái khi sử dụng, khiến họ có một cái nhìn tích cực hơn về tình trạng sức khỏe hiện tại của mình thông qua sự chăm sóc của ngôi nhà thông minh.

* 1. **Màu sắc [7]:**

Theo một nghiên cứu của giáo sư Stephen Westland, màu sắc ảnh hưởng đáng kể đến tinh thần và cơ thể con người. Ví dụ như màu đó khiến tim đập nhanh hơn. Chúng ta thường xuyên thấy được điều này cũng như là rất nhiều các nghiên cứu khác được công bố về tác động của các màu sắc khác nhau lên tinh thần và cơ thể con người. Các cơ chế sinh lý làm nền tảng cho thị giác màu sắc của con người đã được quan tâm và tìm hiểu kỹ càng trong một thế kỷ vừa qua. Nhưng phải cho đến thập kỷ vừa rồi, con người mới bắt đầu tìm được một cách tiếp cận đặc biệt khác đến các hiệu ứng phi thị giác của màu sắc. Giống như tai, nơi không chỉ cung cấp cho con người thông tin về tín hiệu âm thanh mà còn cung cấp cảm giác thăng bằng, mắt cũng được phát hiện rằng có khả năng thực hiện hai chức năng. Các tế bào nhạy cảm với ánh sáng được gọi là tế bào hình nón nằm bên trong võng mạc ở phía sau của mắt gửi các tín hiệu điện hóa đến một khu vực của não được gọi là vỏ não thị giác, nơi hình thành các hình ảnh thị giác mà chúng ta nhìn thấy sau quá trình này. Tuy nhiên, hiện nay chúng ta biết thêm rằng một số tế bào hạch võng mạc phản ứng với ánh sáng bằng cách gửi tín hiệu chủ yếu đến vùng não trung tâm gọi là vùng dưới đồi, vùng này không đóng vai trò gì trong việc hình thành hình ảnh thị giác. Nghĩa là chúng ta phản ứng cả với ánh sáng ngay cả khi không phân biệt được hình ảnh của chúng. Điều này hay còn được gọi là “Light but not vision”.

*Hình 3-1: Đắm mình trong một loại màu thuần túy có ảnh hưởng đến thể chất và tinh thần*

Vùng dưới đồi là phần quan trọng của não, chúng chịu trách nhiệm tiết ra một số hóc-môn kiểm soát nhiều khía cạnh của quá trình tự điều chỉnh của cơ thể, bao gồm nhiệt độ, giấc ngủ, cảm giác đói và nhịp sinh học. Tiếp xúc với ánh sáng vào buổi sáng, đặc biệt là ánh sáng xanh lam, xanh lá cây, thúc đẩy giải phóng hormone cortisol giúp kích thích và đánh thức con người, đồng thời ức chế giải phóng melatonin. Vào buổi tối muộn khi lượng ánh sáng xanh trong ánh sáng mặt trời giảm, melatonin được giải phóng vào máu và họ trở nên buồn ngủ.

Không chỉ dừng lại ở đó, giáo sư Stephen Westland còn đứng đầu một nhóm nghiên cứu Thiết kế trải nghiệm (Experience Design) ở trường đại học Leeds, nơi họ có một phòng thí nghiệm chiếu sáng được thiết kế đặc biệt để đánh giá tác động của ánh sáng đối với hành vi và tâm lý của con người. Hệ thống chiếu sáng này độc đáo ở chỗ nó có thể làm căn phòng tràn ngập ánh sáng màu có bước sóng cụ thể (ánh sáng màu khác thường sử dụng hỗn hợp thô của ánh sáng đỏ, lục và lam). Qua đó họ nghiên cứu và chỉ ra được một tác động nhỏ của ánh sáng màu đối với nhịp tim và huyết áp: ánh sáng đỏ dường như làm tăng nhịp tim, trong khi ánh sáng xanh lam làm giảm nó. Hiệu ứng tuy nhỏ nhưng cũng đã được chứng thực trong một bài báo năm 2015 ở Úc.

Năm 2009, đèn xanh đã được lắp đặt ở cuối các sân ga trên tuyến đường sắt Yamanote của Tokyo để giảm tỷ lệ tự tử. Các con số đã cho thấy được kết quả thành công của những chiếc đèn này. Số vụ tự tử giảm 74% ở các ga có lắp đèn màu xanh lam. Hệ thống chiếu sáng màu tương tự đã được lắp đặt tại các sân ga của Sân bay Gatwick. Các quyết định này được thực hiện dựa trên tuyên bố rằng ánh sáng xanh lam có thể làm cho mọi người bớt bốc đồng và bình tĩnh hơn, nhưng thực tế vẫn có rất ít bằng chứng khoa học chứng minh cho những tuyên bố này. Mặc dù vậy, các nghiên cứu tương tự vẫn tiếp tục được tiến hành trong các phòng thí nghiệm để khám phá ảnh hưởng của màu sắc đối với chất lượng giấc ngủ của con người.

Rõ ràng là ánh sáng, và đặc biệt là màu sắc, có thể ảnh hưởng đến con người theo những cách vượt xa những thường thức về màu thông thường. Việc khám phá ra thị giác không hình ảnh (non-image-forming) đã tạo động lực mới cho nghiên cứu tìm tòi ra cách con người phản ứng, cả về mặt sinh lý và tâm lý, với màu sắc xung quanh. Do đó, các hình ảnh trên nút bấm của giao diện được em lấy màu sắc xanh lam và xanh lá cây làm chủ đạo. Mỗi khi một người thức dậy sau giấc ngủ dài sáu đến tám tiếng, những điều đầu tiên mà con người nhìn thấy sẽ ảnh hưởng rất lớn đến tinh thần của họ trong cả một ngày dài. Việc nhìn thấy những hình ảnh có màu xanh lam và xanh lục như vậy khiến người sử dụng cảm thấy thư thái và phấn chấn, sẵn sàng cho một ngày mới đầy tích cực. Kèm theo đó, giao diện có các cài đặt sẵn như điều chỉnh tông màu, việc cho phép các hình ảnh và màu chữ thay đổi giữa bộ tông màu xanh/đỏ cũng sẽ phần nào hỗ trợ được người bệnh cao/thấp huyết áp. Hoặc đơn giản là thỏa mãn sở thích về màu sắc của người dùng.

* 1. **Ngôn ngữ:**

Hướng tới các đối tượng người dùng không chỉ ở Việt Nam mà còn ở các quốc gia khác, giao diện cho phép chuyển đổi ngôn ngữ sử dụng giữa Tiếng Việt và Tiếng Anh, tăng thêm tính khả dụng của hệ thống. Thay vì sử dụng hoàn toàn là ký hiệu cho các nút bấm, em quyết định sử dụng đan xen cả ký hiệu và chữ viết. Vì nếu chỉ sử dụng ký hiệu không, một số chức năng có thể khó tìm được hình ảnh tượng trưng để dễ dàng liên tưởng, khiến người dùng cảm thấy thiếu đi sự thân thiện cần có. Nhưng nếu dùng các chữ viết, hiển thị của nút bấm có thể trở nên khá đơn điệu với những phông chữ cơ bản. Để khắc phục điều này, em lựa chọn cách thiết kế chữ viết dưới dạng hình ảnh. Chữ được vẽ bằng công cụ điện tử sẽ được cách điệu làm cho hình ảnh nút bấm trở nên hiện đại mà vẫn có thể giữ được khả năng truyền tải thông tin về chức năng đến người dùng một cách dễ dàng.

* 1. **Bố cục [8]:**

Về mặt bố cục, có rất nhiều cách thức để bố cục một giao diện, mỗi một cách thức sẽ có ưu nhược điểm riêng theo chúng. Một số phương án đã được em xem xét như:

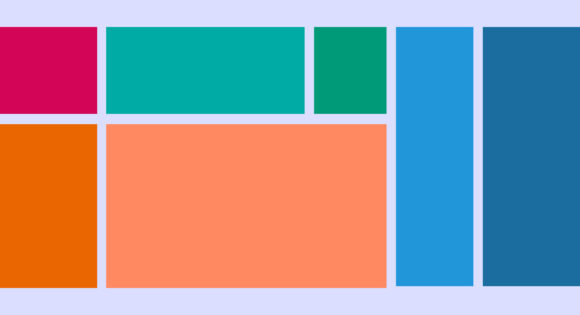
- Bố cục dạng thẻ: Bố cục kiểu thẻ cho phép các nội dung khác nhau được sắp xếp một cách có trật tự trong khi vẫn giữ tất cả các phần riêng biệt. Những chiếc thẻ cũng tạo nên cảm giác quen thuộc cho mọi người vì họ chẳng lạ gì những vật phẩm hình dạng như thế đến từ thế giới thực. Chúng có ý nghĩa về mặt tâm lý và mọi người dễ sử dụng ngay cả khi họ mới sử dụng một trang web hay một ứng dụng.

- Bố cục dạng lưới: Bố cục dạng lưới từ lâu đã là một phần của thiết kế giao diện người dùng, bắt đầu từ những bố cục dựa trên bảng vào cuối những năm 1990. Vì vậy ưu điểm đầu tiên của bố cục này là sự quen thuộc với người dùng. Lưới cũng cung cấp sự cân bằng trực quan và trật tự cho một thiết kế, giúp người sử dụng dễ dàng tiếp nhận nội dung hơn. Đồng thời, lưới có thể cung cấp nhiều tính linh hoạt trong bố cục. Phần lớn các hệ thống lưới sử dụng 12 hoặc 16 cột cùng với máng xối (gutter) ở giữa nếu muốn. Rất nhiều trang web và ứng dụng cũng sử dụng bố cục dựa trên lưới làm cho các lưới trở thành một trong các bố cục nổi bật dùng trong thiết kế. Thêm vào đó, việc bố cục dạng lưới có khả năng phân chia không gian cho các phần tử một cách dễ dàng và hợp lý. Việc nhìn thấy ứng dụng được sắp xếp một cách kỷ luật và trật tự dưới dạng lưới cũng có thể khiến người dùng cảm thấy tin cậy.

- Bố cục dạng băng chuyền: Nội dung được bố cục theo dạng băng chuyền thường được thấy trong phần tiêu đề của một trang web hay phần chính trong một ứng dụng. Chúng có thể chứa được một dung lượng thông tin rất lớn bao gồm cả hình ảnh và văn bản, được chia thành các phần nhỏ và hiển thị lên trong cùng một màn hình. Điều này đồng nghĩa với việc không cần thiết phải thay đổi màn hình trong quá trình người dùng thao tác. Bố cục băng chuyền cũng tạo ra một sự mượt mà nhất định khi các phần nhỏ trong giao diện thay phiên nhau hiển thị lên màn hình. Bố cục băng chuyền hoạt động thực sự tốt nếu đúng mục đích sử dụng. Chúng là một lựa chọn phổ biến cho cả nội dung giao diện chính và các giao diện riêng lẻ cho các danh mục nội dung hoặc sản phẩm cụ thể.

- Bố cục dạng đối xứng qua tâm: Bố cục dạng đối xứng qua tâm hay còn gọi là đối xứng tròn là một dạng bố cục làm nổi bật lên phần trung tâm mà các đối tượng khác được đặt đối xứng ở xung quanh nó. Điều này có thể dễ dàng khiến người dùng thấy được phần chính của giao diện mà mình muốn nhắm đến, đồng thời cũng tự động phân các đối tượng quanh nó có mức độ quan trọng bằng nhau. Bố cục dạng này khiến thị giác người dùng cảm thấy thoải mái vì một số nghiên cứu đã chỉ ra rằng con người có xu hướng thả lỏng khi nhìn hay sờ vào các vật tròn và cảm thấy ức chế khi nhìn vào các vật có góc cạnh sắc nhọn. Khi giao diện chuyển động, bố cục đối xứng qua tâm còn tạo ra một loại nét thẩm mỹ và thu hút người nhìn. Nhìn chung, các mẫu hình tròn có khả năng khuyến khích người dùng liên tục chuyển trọng tâm tầm nhìn của họ đến một điểm theo cách rất tự nhiên.

Qua đánh giá, bố cục dạng lưới được coi là khả thi nhất, mặc dù có khuyết điểm khó phân chia thứ cấp trong lưới nhưng điều này gần như không cần quan tâm ở hệ thống này. Bởi vì các thiết bị được đăng ký đều có thứ bậc là hoàn toàn bằng nhau khi hiển thị, điều này đã triệt tiêu được khiếm khuyết, cùng với đó là các ưu điểm đã được nêu ra của bố cục dạng lưới hoàn toàn phù hợp với yêu cầu và mong muốn của người sử dụng chương trình.



*Hình 3-2: Giao diện được biểu diễn dưới dạng lưới*

* 1. **Kích thước:**

Khi đưa vào sử dụng, màn hình có thể được đặt gần hoặc xa với người sử dụng. Trong trường hợp đặt xa, một số người có thị lực yếu có thể sẽ không nhìn rõ được các nút trên màn hình. Vì vậy các nút bấm được thiết kế với độ lớn ít nhất là khoảng 1/25 màn hình để chắc chắn rằng ký hiệu và chữ trên đó được hiển thị rõ ràng cũng như diện tích đủ lớn để đảm bảo thuận tiện cho những người điều khiển bằng mắt.

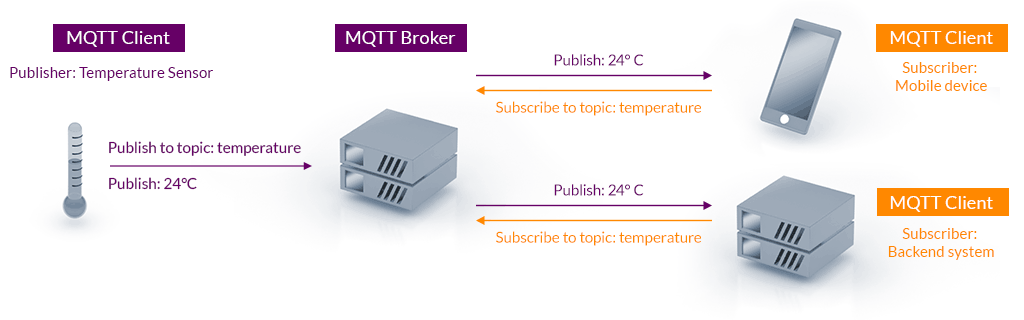
Các hình ảnh và văn bản bên trong từng nút bấm cũng được thiết kế với kích cỡ thay đổi vừa đủ phù hợp với nút bấm, không để bị quá nhỏ khiến người sử dụng gặp khó khăn trong quá trình sử dụng.

* 1. **Biểu tượng:**

Một trong các thành phần chính của giao diện tạo nên độ thân thiện với người dùng là các biểu tượng. Những biểu tượng này sẽ thay thế văn bản để đem nội dung cần thể hiện đến với người dùng. Ưu điểm của biểu tượng là ngắn ngọn, súc tích. Một biểu tượng đơn giản có thể thay thể cho một vài từ, điều này sẽ giúp tối ưu diện tích trên lưới để hiển thị nội dung. Quan trọng hơn, biểu tượng được coi là một ngôn ngữ riêng, chính vì vậy, việc sử dụng biểu tượng có thể giúp người dùng hiểu được ý nghĩa mà không cần quan tâm họ biết những ngôn ngữ thông dụng nào. Bên cạnh đó, biểu tượng cũng góp phần tạo nên độ trẻ trung, linh hoạt cho giao diện, giúp giao diện không chỉ có những dòng chữ thuyết minh cứng nhắc.

1. **API [9]:**

Về thiết kế API, em chọn MQTT làm giao thức chính để truyển – nhận tin giữa các thiết bị. Giao thức này hoạt động chính trên nền tảng TCP/IP; tuy nhiên MQTT cũng hỗ trợ các giao thức mạng khác làm tăng độ linh hoạt của hệ thống. MQTT được tập trung thiết kế cho các kết nối cho việc truyền tải dữ liệu cho các thiết bị hay vi điều khiển nhỏ có tài nguyên hạn chế, hoặc trong các ứng dụng có băng thông mạng bị hạn chế. Vì vậy, giao thức MQTT rất phù hợp sử dụng trong hệ thống này.

Kiến trúc mức cao của MQTT gồm 2 phần chính là Broker và Clients. Broker được coi là trái tim, là điểm giao nhau giữa các kết nối đến client. Broker đảm nhận nhiệm vụ tiếp nhận message được publish từ client, sắp xếp các gói tin theo một thứ tự hàng đợi rồi gửi chúng tới một địa chỉ topic được xác định. Ngoài ra, broker còn đảm nhận một số nhiệm vụ phụ như bảo mật các gói tin, cung cấp các logs trong quá trình truyền nhận, lưu trữ gói tin,... Phần chính còn lại của kiến trúc MQTT là Client. Phần chính của Client bao gồm 2 bộ phận là publisher và subscriber. Client ở đây là các phần mềm trong các thiết bị, chúng chỉ làm ít nhất một trong hai việc là publish các gói tin lên một địa chỉ topic cụ thể hoặc subscribe một địa chỉ topic để có thể nhận được các gói tin từ topic này. 

*Hình 3-3: Ứng dụng kiến trúc MQTT trong thực tế*

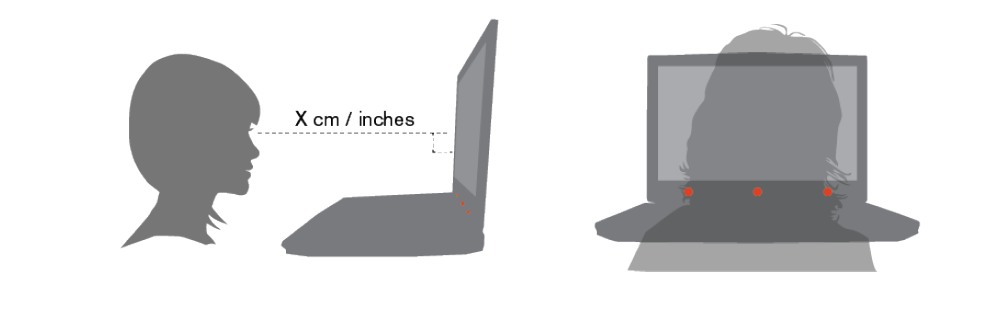
Trong hệ thống, em sử dụng topic iot/control đảm nhận nhiệm vụ nhận những câu lệnh điều khiển thiết bị từ phía client rồi gửi trả về phía server.

Khi người dùng tương tác với nút bấm có chức năng điều khiển thiết bị, một gói tin chứa thông tin về thiết bị cần điều khiển và chức năng cần thực hiện sẽ được gửi lên topic iot/control.

Để có được danh sách thiết bị đã đăng ký trên máy chủ, client sẽ gửi một HTTP request đến địa chỉ có sẵn trong hệ thống và ghi dữ liệu nhận được vào một tập tin. Sau đó chương trình đọc tập tin trên và lưu giữ, xử lý thông tin đó.

1. **Tobii Eye Tracker [11]:**

Như đã nêu ở chương 2.3. Tobii eye tracker là thiết bị theo dõi mắt có khả năng nhận tín hiệu thị giác của người dùng và theo dõi những gì họ đang nhìn trong thời gian thực. Tuy vậy, để thiết bị này có thể hoạt động bình thường cũng đòi hỏi đáp ứng những tiêu chuẩn nhất định, đặc biệt là về khoảng cách và phương hướng từ thiết bị theo dõi mắt đến mắt của người dùng.

Đầu tiên, khoảng cách hoạt động của thiết bị theo dõi mắt Tobii là 50 – 95 centimet (20 – 37 inches). Nó cần được gắn ở dưới cùng của màn hình và bản thân người dùng phải nằm trong trường quan sát của cảm biến. 

*Hình 3-4: Cách bố trí vị trí sử dụng cho Tobii Eye Tracker*

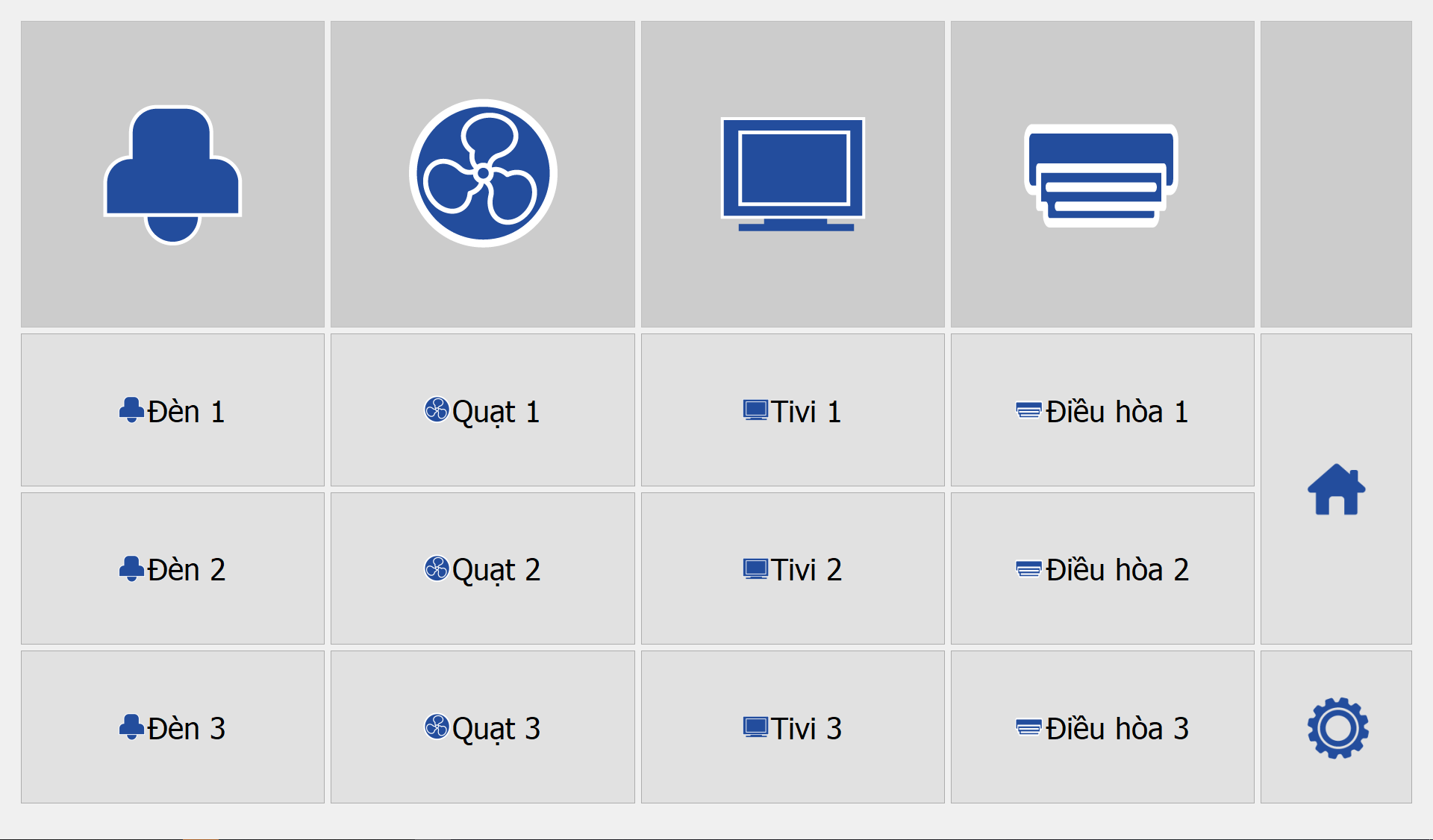
Khoảng cách tối ưu từ người dùng nên đến Tobii Eye Tracker phụ thuộc vào kích thước của màn hình. Người sử dụng nên chọn vị trí tối ưu cho bản thân trước, sao cho mọi hoạt động của mình được thoải mái nhất có thể. Sau đó dựa vào đó để thiết kế vị trí đặt thiết bị theo dõi mắt. Kích thước của màn hình càng lớn thì khoảng cách giữa người dùng tới thiết bị theo dõi mắt và màn hình càng lớn. Kích thước được đề xuất tối đa là 27 inches cho màn hình có tỉ lệ 16:9 và 30 inches cho màn hình có tỉ lệ 21:9.

Bộ theo dõi mắt cũng cho phép chuyển động đầu tự do. Sau khi thiết bị theo dõi mắt được hiệu chỉnh đúng cách và đặt trước mặt người dùng, thiết bị có thể hoạt động trơn tru mà không cần điều chỉnh gì thêm.

1. **Thiết kế giao diện điều khiển**

Về tổng thể, các màn hình điều khiển thiết bị trong giao diện đều được thiết kế đồng đều dưới dạng lưới 5x5. Nút bấm quay lại đều được đặt ở phía trên bên phải tạo nên sự thống nhất giúp người dùng có khả năng dễ dàng tìm được sau một thời gian ngắn sử dụng.

1. **Giao diện nhà thông minh:**



*Hình 4-1: Giao diện nhà thông minh*

Giao diện nhà thông minh gồm 5 cột. Trong đó 4 cột đầu tính từ trái sang phải sẽ lần lượt liệt kê các thiết bị. Cột thứ năm gồm hai nút quay lại màn hình chính và chuyển sang màn hình tùy chỉnh. Nút quay lại màn hình chính được thiết kế có độ cao gấp đôi vì người dùng sẽ sử dụng chức năng này nhiều hơn chức năng tùy chỉnh và đồng thời cũng để phù hợp với bố cục giao diện.

Giao diện nhà thông minh bao gồm các nút có chức năng:

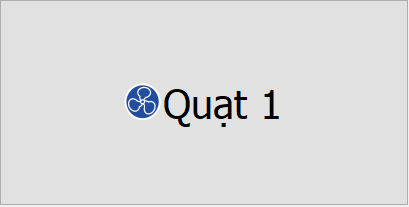
- Lựa chọn một thiết bị đèn đã được đăng ký trước



*Hình 4-2: Nút chọn thiết bị đèn được đăng ký trên server với tên “Đèn 1”*

Chuyển màn hình giao diện điều khiển đèn, cho phép người dùng truy cập vào một thiết bị đèn được đăng ký và định danh từ trước trên server để thực hiện các chức năng điều khiển có trong đó.

- Lựa chọn một thiết bị quạt đã được đăng ký trước



*Hình 4-3: Nút chọn thiết bị quạt được đăng ký trên server với tên “Quạt 1”*

Chuyển sang giao diện điều khiển quạt, cho phép người dùng truy cập vào một thiết bị quạt được đăng ký và định danh từ trước trên server để thực hiện các chức năng điều khiển có trong đó.

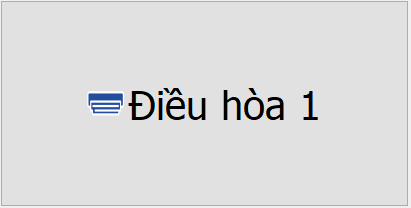
- Lựa chọn một thiết bị vô tuyến đã được đăng ký trước



*Hình 4-4: Nút chọn thiết bị vô tuyến được đăng ký trên server với tên “Tivi 1”*

Chuyển sang giao diện điều khiển vô tuyến, cho phép người dùng truy cập vào một thiết bị vô tuyến được đăng ký và định danh từ trước trên server để thực hiện các chức năng điều khiển có trong đó.

- Lựa chọn một thiết bị điều hòa đã được đăng ký trước



*Hình 4-5: Nút chọn thiết bị điều hòa được đăng ký trên server với tên “Điều hòa 1”*

Chuyển sang giao diện điều khiển điều hòa, cho phép người dùng truy cập vào một thiết bị điều hòa được đăng ký và định danh từ trước trên server để thực hiện các chức năng điều khiển có trong đó.

- Quay trở lại giao diện chính



*Hình 4-6: Nút quay trở lại giao diện chính*

Đưa màn hình trở về giao diện chính của hệ thống

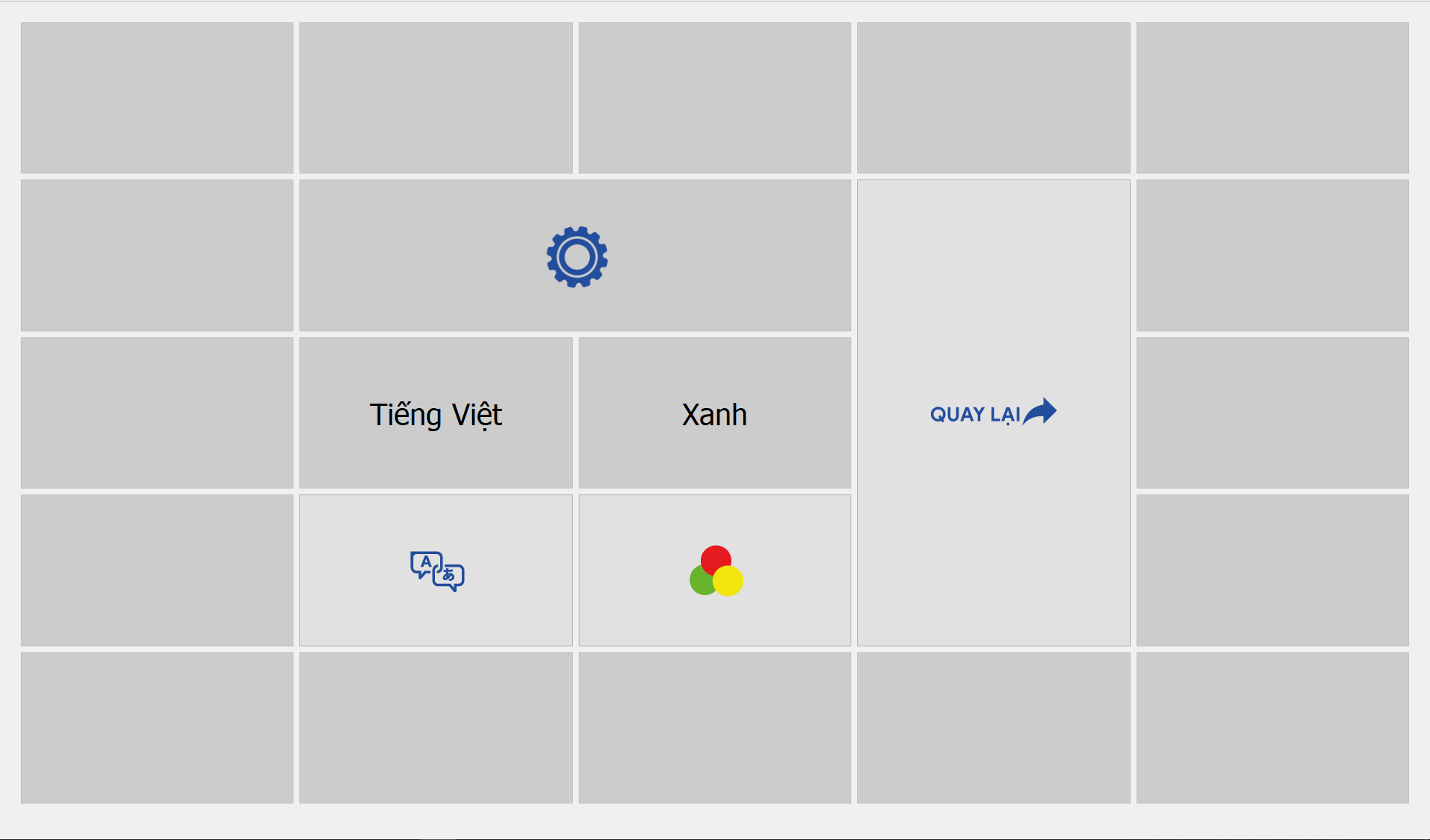
- Truy cập vào giao diện tùy chỉnh



*Hình 4-7: Nút chuyển sang giao diện tùy chỉnh*

Chuyển sang giao diện tùy chỉnh, cho phép người dùng có thể thay đổi các hiển thị trong giao diện theo cách mình muốn.

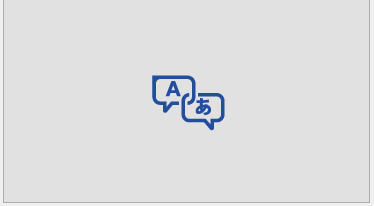
1. **Giao diện tùy chỉnh:**



*Hình 4-8: Giao diện tùy chình*

Giao diện tùy chỉnh bao gồm các nút có chức năng:

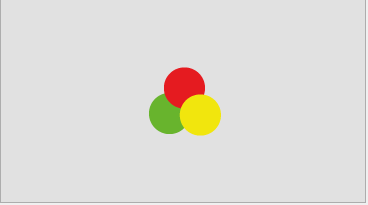
- Chuyển đổi ngôn ngữ:



*Hình 4-9: Nút chuyển sang ngôn ngữ khác*

Chức năng này sẽ giúp người dùng thay đổi ngôn ngữ hiển thị trên các nút trong các màn hình giao diện. Hiện tại, giao diện có hỗ trợ Tiếng Anh và Tiếng Việt.

- Chuyển đổi màu sắc văn bản:



*Hình 4-10: Nút chuyển sang màu sắc hiển thị khác*

Chức năng này cho phép người dùng thay đổi màu sắc hiển thị trên các nút bấm trong các màn hình giao diện. Hiện tại, giao diện có hỗ trợ màu đỏ và màu xanh lam.

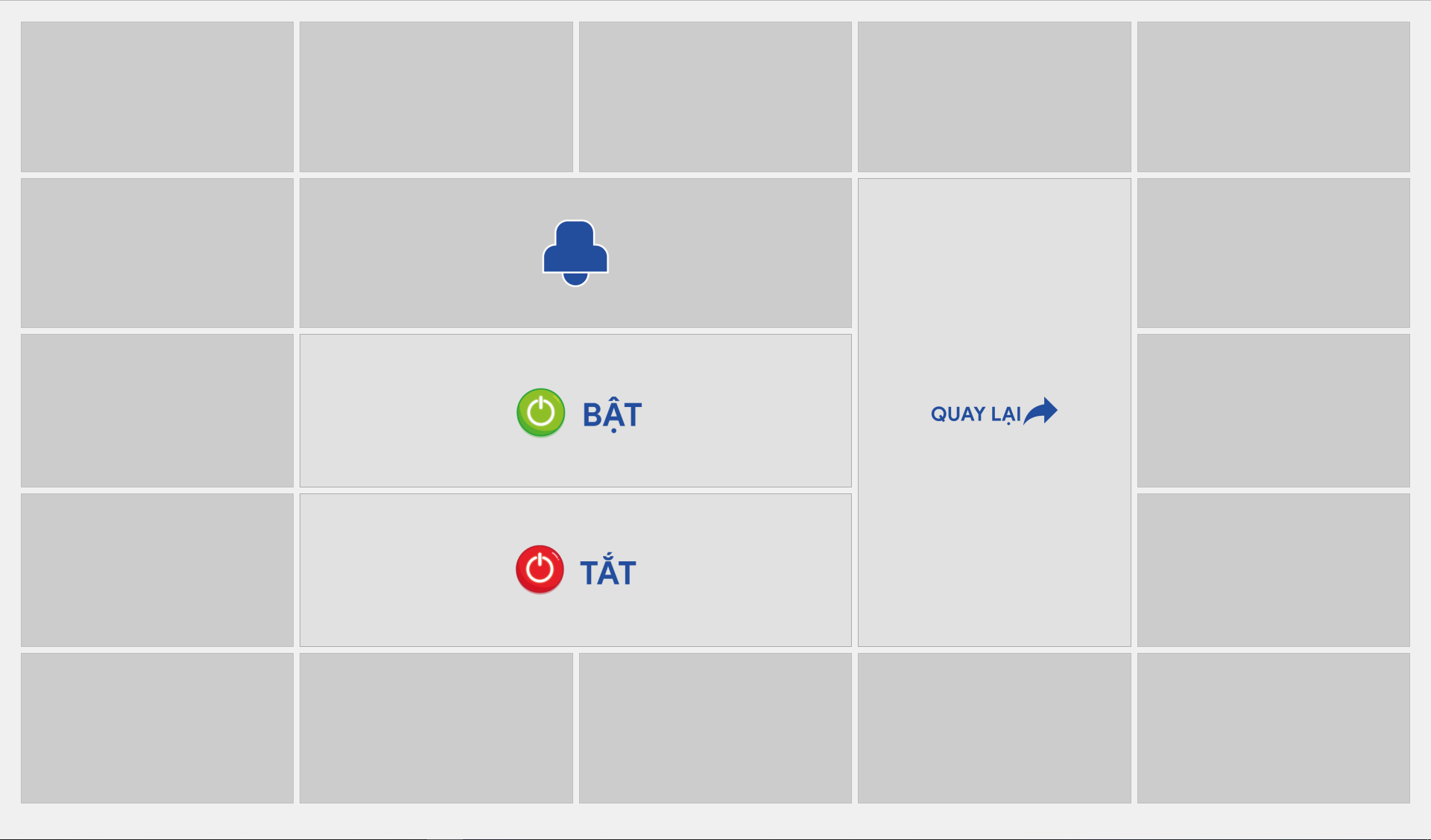
- Quay lại giao diện nhà thông minh:



*Hình 4-11: Nút chuyển về giao diện nhà thông minh*

Chuyển về giao diện nhà thông minh để người dùng có thể lựa chọn thiết bị mình muốn điều khiển hoặc tùy chỉnh giao diện hệ thống.

1. **Giao diện điều khiển đèn:**



*Hình 4-12: Giao diện điều khiển đèn*

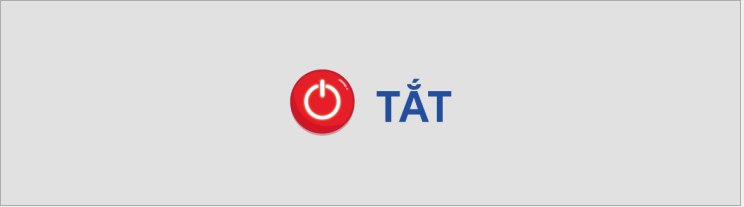
Giao diện điều khiển đèn bao gồm các nút có chức năng:  
- Bật đèn:



*Hình 4-13: Nút bật đèn*

Thực hiện hành động bật thiết bị đèn tương ứng đã được người dùng lựa chọn trước khi chuyển sang màn hình này.

- Tắt đèn:

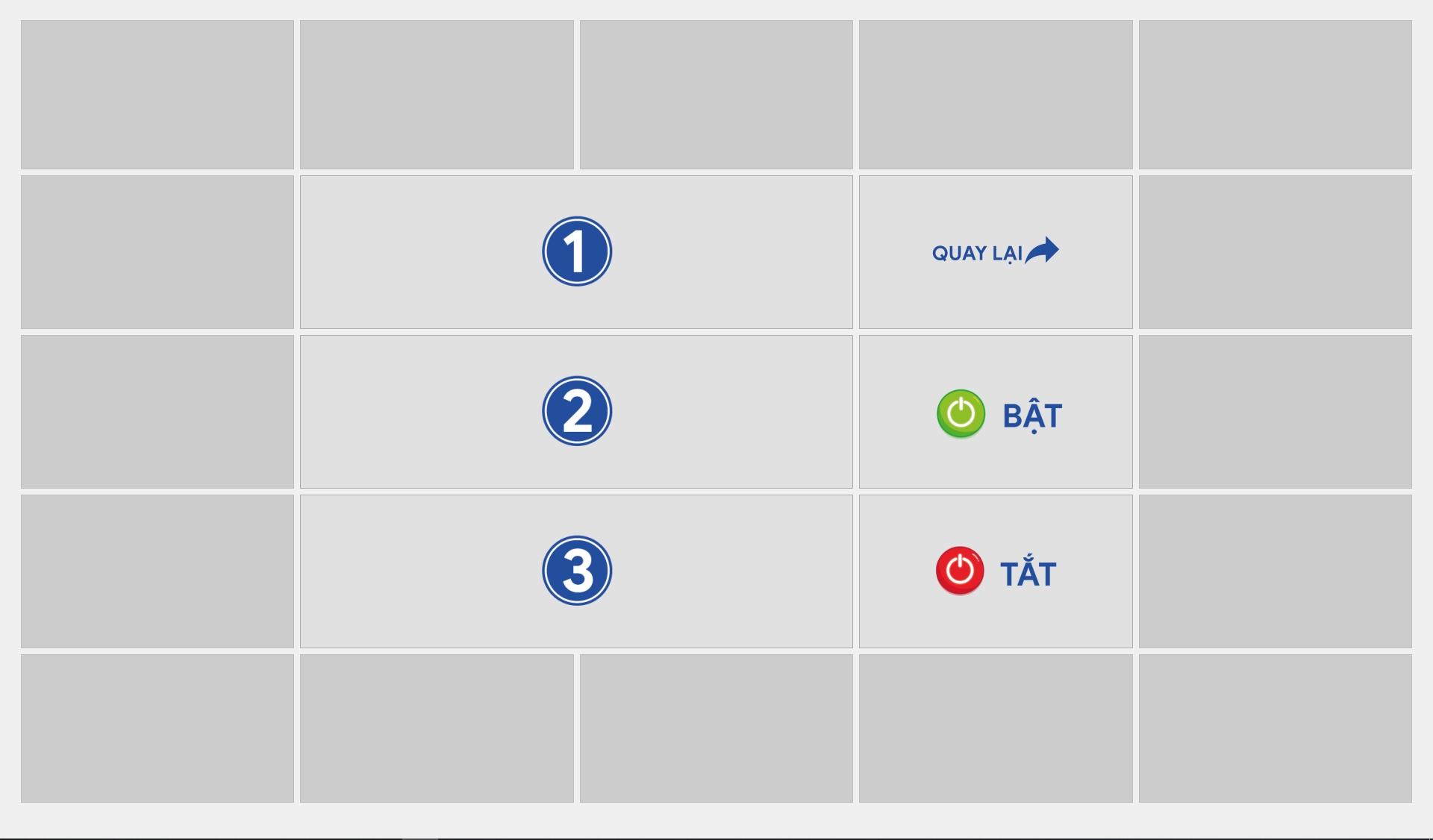


*Hình 4-14: Nút tắt đèn*

Thực hiện hành động tắt thiết bị đèn tương ứng đã được người dùng lựa chọn trước khi chuyển sang màn hình này.

- Quay lại giao diện nhà thông minh: Chuyển về giao diện nhà thông minh để người dùng có thể lựa chọn thiết bị mình muốn điều khiển hoặc tùy chỉnh giao diện hệ thống (Hình 4-11).

1. **Giao diện điều khiển quạt:**



*Hình 4-15: Giao diện điều khiển quạt*

Giao diện điều khiển quạt bao gồm các nút có chức năng:

- Bật quạt: Thực hiện hành động bật thiết bị quạt tương ứng đã được người dùng lựa chọn trước khi chuyển sang màn hình này (Hình 4-13).

- Tắt quạt: Thực hiện hành động tắt thiết bị quạt tương ứng đã được người dùng lựa chọn trước khi chuyển sang màn hình này (Hình 4-14).

- Chọn mức gió đầu tiên:



*Hình 4-16: Nút chuyển sang mức gió đầu tiên*

Thực hiện hành động chuyển sang mức gió đầu tiên cho thiết bị quạt tương ứng đã được người dùng lựa chọn trước khi chuyển sang màn hình này.

- Chọn mức gió thứ hai:



*Hình 4-17: Nút chuyển mức gió thứ hai*

Thực hiện hành động chuyển sang mức gió thứ hai cho thiết bị quạt tương ứng đã được người dùng lựa chọn trước khi chuyển sang màn hình này.

- Chọn mức gió thứ ba:

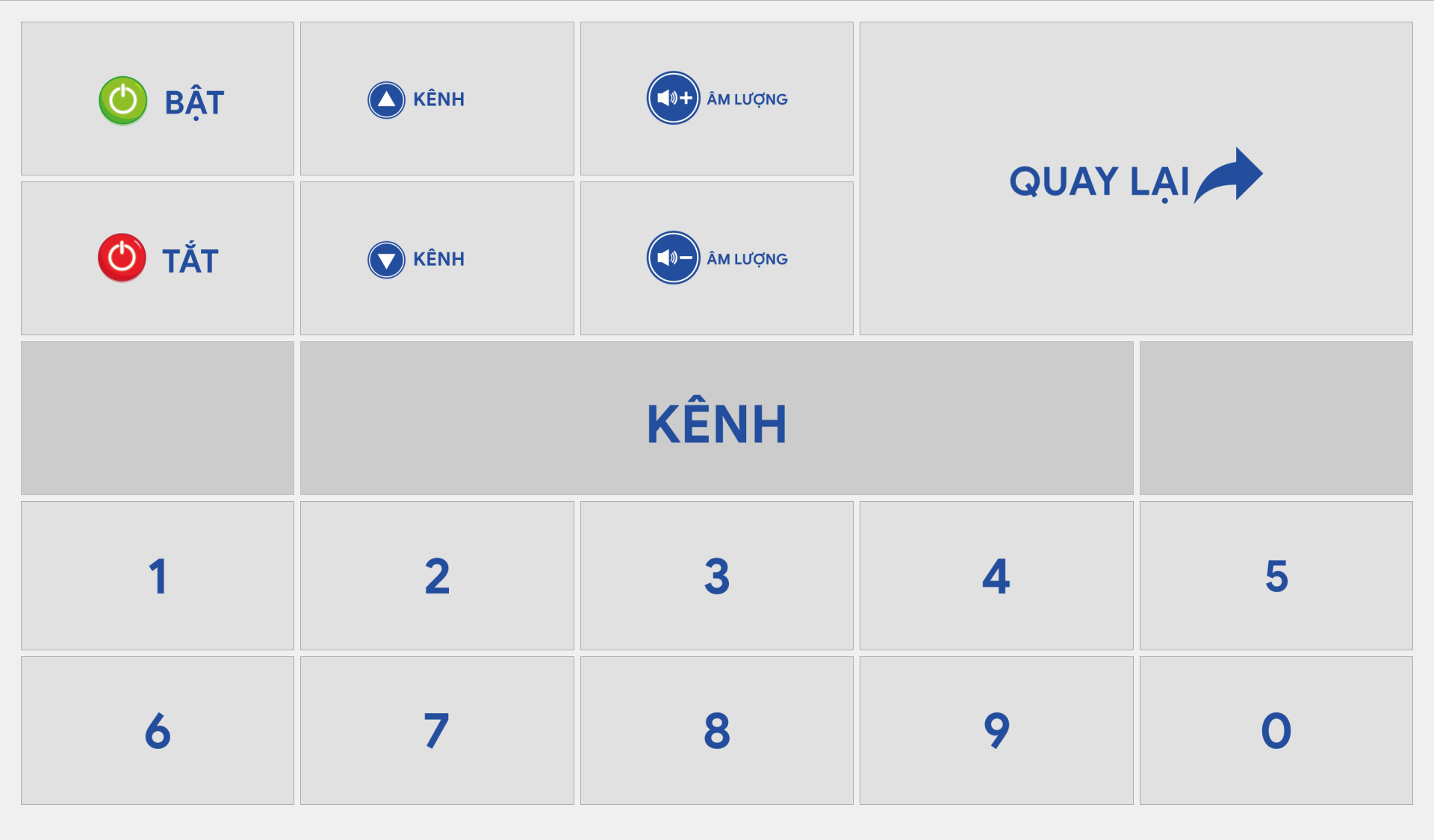


*Hình 4-18: Nút chuyển sang mức gió thứ ba*

Thực hiện hành động chuyển sang mức gió thứ ba cho thiết bị quạt tương ứng đã được người dùng lựa chọn trước khi chuyển sang màn hình này.

- Quay lại giao diện nhà thông minh: Chuyển về giao diện nhà thông minh để người dùng có thể lựa chọn thiết bị mình muốn điều khiển hoặc tùy chỉnh giao diện hệ thống (Hình 4-11).

1. **Giao diện điều khiển vô tuyến:**



*Hình 4-19: Giao diện điều khiển vô tuyến*

Giao diện điều khiển vô tuyến bao gồm các nút có chức năng:

- Bật vô tuyến: Thực hiện hành động bật thiết bị vô tuyến tương ứng đã được người dùng lựa chọn trước khi chuyển sang màn hình này (Hình 4-13).

- Tắt vô tuyến: Thực hiện hành động tắt thiết bị vô tuyến tương ứng đã được người dùng lựa chọn trước khi chuyển sang màn hình này (Hình 4-14).

- Chuyển sang kênh tiếp theo:



*Hình 4-20: Nút chuyển sang kênh kế tiếp*

Thực hiện hành động chuyển sang kênh tiếp theo cho thiết bị vô tuyến tương ứng đã được người dùng lựa chọn trước khi chuyển sang màn hình này.

- Chuyển sang kênh trước đó:



*Hình 4-21: Nút chuyển sang kênh trước đó*

Thực hiện hành động chuyền về kênh trước đó cho thiết bị vô tuyến tương ứng đã được người dùng lựa chọn trước khi chuyển sang màn hình này.

- Tăng âm lượng:



*Hình 4-22: Nút tăng âm lượng*

Thực hiện hành động tăng âm lượng cho thiết bị vô tuyến tương ứng đã được người dùng lựa chọn trước khi chuyển sang màn hình này.

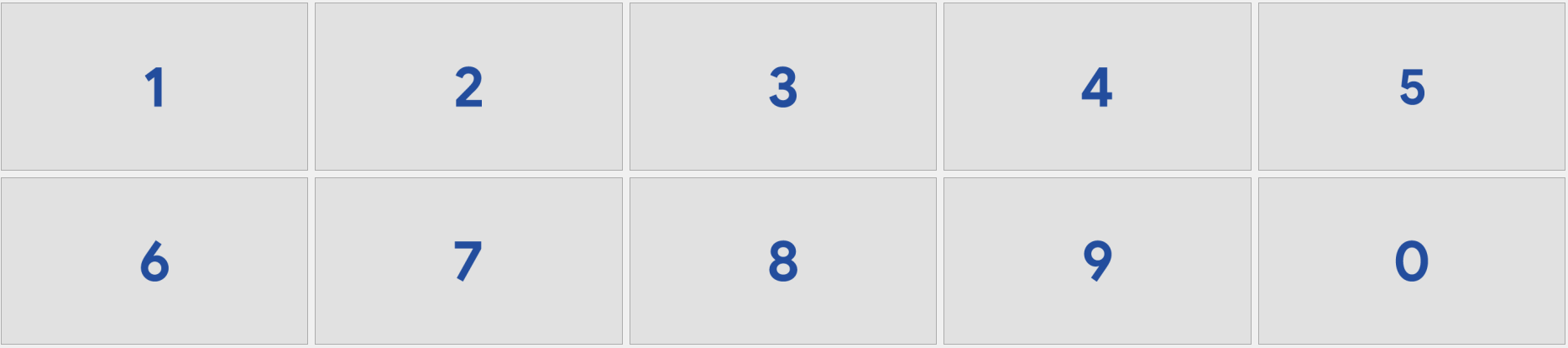
- Giảm âm lượng:



*Hình 4-23: Nút giảm âm lượng*

Thực hiện hành động giảm âm lượng cho thiết bị vô tuyến tương ứng đã được người dùng lựa chọn trước khi chuyển sang màn hình này.

- Bàn phím đánh số từ 0 đến 9:

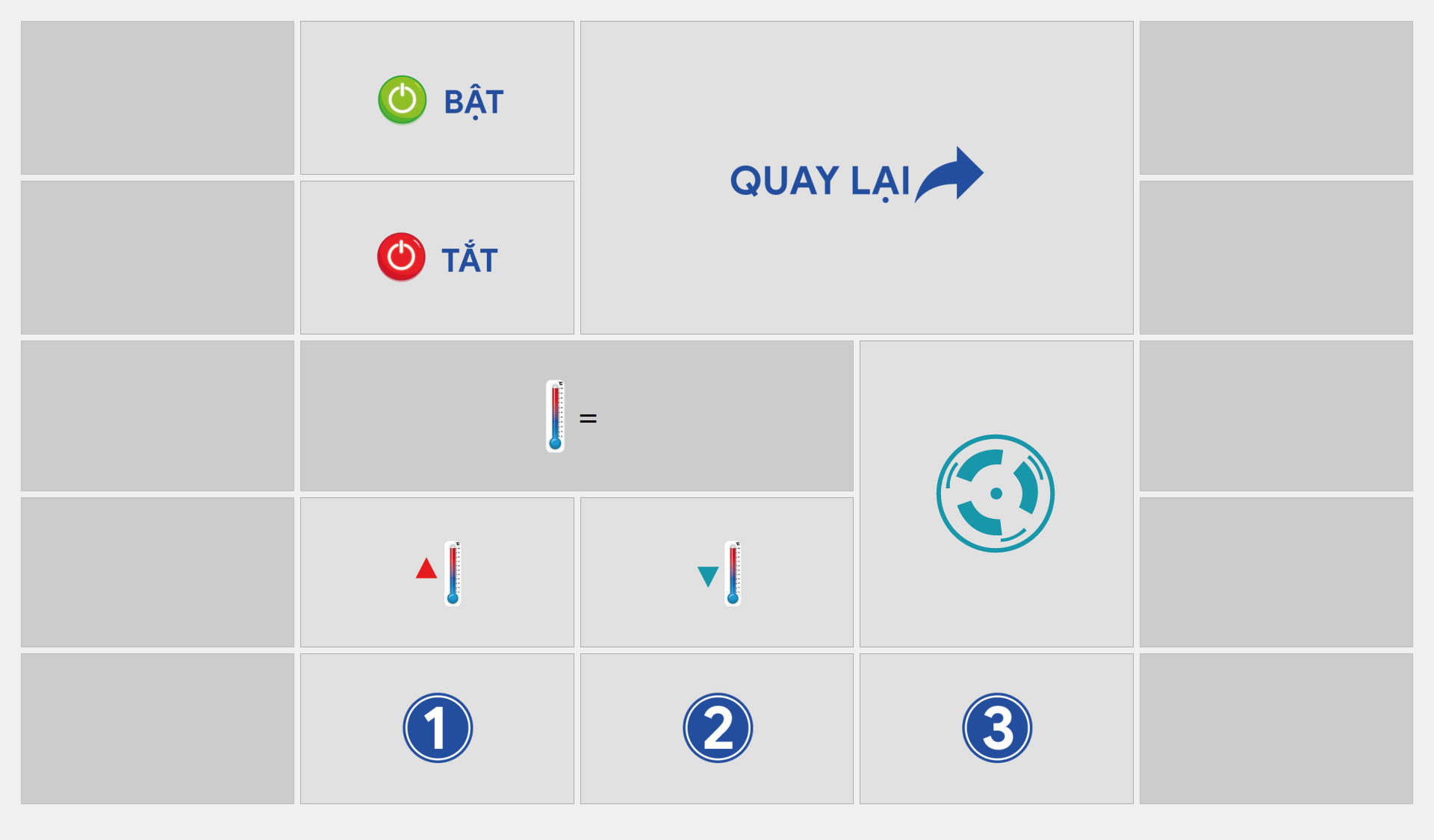


*Hình 4-24: Bàn phím số*

Thực hiện hành động gửi một trong các số từ 0 đến 9 được người dùng lựa chọn để thực hiện quá trình chuyển kênh cho thiết bị vô tuyến tương ứng đã được họ lựa chọn trước khi chuyển sang màn hình này.

- Quay lại giao diện nhà thông minh: Chuyển về giao diện nhà thông minh để người dùng có thể lựa chọn thiết bị mình muốn điều khiển hoặc tùy chỉnh giao diện hệ thống (Hình 4-11).

1. **Giao diện điều khiển điều hòa:**



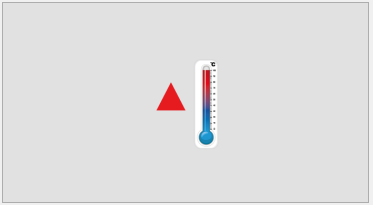
*Hình 4-25: Giao diện điều khiển điều hòa*

Giao diện điều khiển điều hòa bao gồm các nút có chức năng:

- Bật điều hòa: Thực hiện hành động bật thiết bị điều hòa tương ứng đã được người dùng lựa chọn trước khi chuyển sang màn hình này (Hình 4-13).

- Tắt điều hòa: Thực hiện hành động tắt thiết bị điều hòa tương ứng đã được người dùng lựa chọn trước khi chuyển sang màn hình này (Hình 4-14).

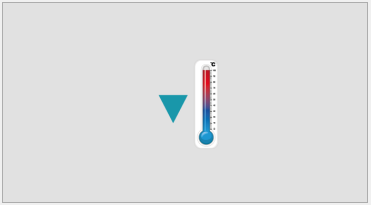
- Tăng nhiệt độ:



*Hình 4-26: Nút tăng nhiệt độ*

Thực hiện hành động tăng nhiệt độ thêm 1 đơn vị nhiệt độ cho thiết bị điều hòa tương ứng đã được người dùng lựa chọn trước khi chuyển sang màn hình này.

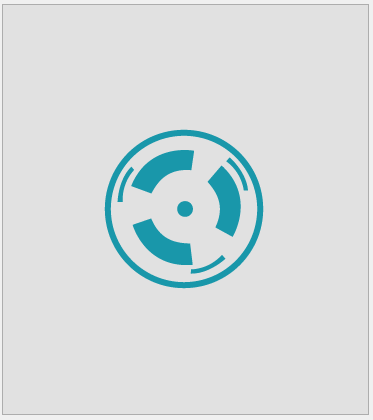
- Giảm nhiệt độ:



*Hình 4-27: Nút giảm nhiệt độ*

Thực hiện hành động giảm 1 đơn vị nhiệt độ cho thiết bị điều hòa tương ứng đã được người dùng lựa chọn trước khi chuyển sang màn hình này.

- Xoay cánh điều gió:



*Hình 4-28: Nút bật/tắt cánh điều gió*

Khởi động/Tắt đi cánh điều gió của thiết bị điều hòa tương ứng đã được người dùng lựa chọn trước khi chuyển sang màn hình này.

- Chọn mức gió đầu tiên: Thực hiện hành động chuyển sang mức gió đầu tiên cho thiết bị điều hòa tương ứng đã được người dùng lựa chọn trước khi chuyển sang màn hình này (Hình 4-16).

- Chọn mức gió thứ hai: Thực hiện hành động chuyển sang mức gió thứ hai cho thiết bị điều hòa tương ứng đã được người dùng lựa chọn trước khi chuyển sang màn hình này (Hình 4-17).

- Chọn mức gió thứ ba: Thực hiện hành động chuyển sang mức gió thứ ba cho thiết bị điều hòa tương ứng đã được người dùng lựa chọn trước khi chuyển sang màn hình này (Hình 4-18).

- Quay lại giao diện nhà thông minh: Chuyển về giao diện nhà thông minh để người dùng có thể lựa chọn thiết bị mình muốn điều khiển hoặc tùy chỉnh giao diện hệ thống (Hình 4-11).

1. **Thiết kế API dựa trên giao thức MQTT và HTTP**
   1. **API lấy danh sách thiết bị**
   2. **Các phương án thiết kế**
      1. **Sử dụng giao thức MQTT:**

- Ý tưởng: Khi người dùng bắt đầu mở màn hình giao diện nhà thông minh, một gói tin sẽ được gửi lên topic iot/request với payload là "{\"data\": \"devices\"}". Kế tiếp, client sẽ chờ server phản hồi. Sau khi server nhận được thông tin client gửi lên và xác nhận thì sẽ trả về một gói tin có payload mang thông tin các thiết bị đã được đăng ký trên hệ thống qua topic iot/request phía trên. Phía client đã subscribe vào topic iot/request từ trước đó nên khi server gửi trả thông tin vào đó, client sẽ có thể nhận được thông tin về các thiết bị và lưu trữ nó.

- Kết quả thử nghiệm: Với một lượng thông tin nhỏ khoảng 1 – 2 thiết bị được đăng ký, giao thức có thể gửi được toàn vẹn thông tin của chúng đến phía client. Nhưng khi lượng thông tin trở nên lớn hơn, khoảng từ 3 thiết bị trở lên, payload trả về của giao thức MQTT đã không thể lưu trữ được thông tin với dung lượng lớn như vậy nên nó đã trả về dữ liệu rỗng. Điều này chứng tỏ giao thức MQTT không còn phù hợp trong việc lấy dữ liệu thiết bị được người dùng đăng ký trên hệ thống.

* + 1. **Sử dụng giao thức HTTP:**

- Ý tưởng: Là một thay thế cho việc sử dụng giao thức MQTT. Với khả năng định hướng địa chỉ rõ ràng, giao thức HTTP cũng là một phương án tốt trong việc lấy danh sách thiết bị từ hệ thống.

- Thử nghiệm giao thức HTTP với thư viện Poco/C++: Khi người dùng bắt đầu mở màn hình giao diện nhà thông minh, chương trình sẽ khởi tạo một kết nối HTTP có thông tin địa chỉ là “address:port/desvc” tương ứng với API trên server. Thư viện cũng cũng cấp sẵn phương thức cho phép lưu lại thông tin trả về từ server. Kết quả thử nghiệm cho thấy với một dung lượng thông tin lớn được gửi về, tuy lần này server đã gửi thành công toàn bộ thông tin về thiết bị về client, nhưng lượng thông tin lớn đã làm cho con trỏ trong phương thức của thư viện Poco bị tràn bộ nhớ khiến cho không thể đọc và lưu được các dữ liệu trên.

- Thử nghiệm giao thức HTTP với thư viện stdio.h/C++ và requests/Python: Khi người dùng bắt đầu mở màn hình giao diện nhà thông minh, một phương thức của thư viện stdio.h sẽ cho phép chương trình biên dịch một đoạn mã python. Đoạn mã python này sử dụng thư viện requests để gửi một HTTP request đến địa chỉ “address:port/desvc” tương ứng với API trên server. Sau đó nhận kết quả và in ra console. Chương trình sau đó sẽ sử dụng một phương thức khác của thư viện stdio.h cho phép đọc kết quả được in ra trên console và lưu trữ chúng. Kết quả thử nghiệm phương án này cho thấy sự tiến triển tích cực. Giao diện đã có thể nhận được thông tin các thiết bị gửi về từ server dù dung lượng lớn hay nhỏ. Tuy vậy, phương thức đọc dữ liệu của thư viện stdio.h không thể đọc dữ liệu trên console và lưu trữ chúng dưới định dạng Unicode. Điều này khiến cho khi người dùng đăng ký thiết bị với tên Tiếng Việt có dấu, phương thức đọc sẽ bị lỗi và trả về rỗng. Mặc dù đã thành công trong quá trình nhận dữ liệu, nhưng việc không thể sử dụng Tiếng Việt có dấu khi đặt tên là một bất cập rất lớn vì thế vẫn yêu cầu chương trình cần phải cải tiến thêm.

- Thử nghiệm giao thức HTTP với thư viện stdio.h/C++, QFile và request/python: Tương tự với ý tưởng trước đó, nhưng lần này thay vì ghi dữ liệu nhận về lên console, đoạn mã python sẽ ghi dữ liệu nhận về vào một tập tin với kiểu .txt. Sau đó, chương trình chính sẽ sử dụng thư viện QFile của QT để đọc dữ liệu từ tập tin .txt mới được ghi trên để lưu dữ liệu. Kết quả lần thử nghiệm này cho thấy chương trình đã có thêm khả năng nhận dữ liệu với định dạng Unicode từ server, qua đó hoàn thiện API lấy danh sách thiết bị.

* 1. **Kết quả thiết kế**

Khi người dùng bắt đầu mở màn hình giao diện nhà thông minh, một phương thức của thư viện stdio.h sẽ cho phép chương trình biên dịch một đoạn mã python. Đoạn mã python này sử dụng thư viện requests để gửi một HTTP request đến địa chỉ “address:port/desvc” tương ứng với API trên server. Sau khi nhận được thông tin về danh sách thiết bị, dữ liệu trên sẽ được cấu trúc lại dưới định dạng json sau đó lưu vào một tập tin có tên là “DeviceList.txt”.



*Hình 5-1: Format của dữ liệu nhận từ server*

Sau đó, chương trình chính sẽ sử dụng thư viện QFile đọc dữ liệu từ tập tin “DeviceList.txt” trên, qua đó lưu trữ dữ liệu vào các mảng có kiểu dữ liệu QJsonObject của QT. Do chương trình đa số sử dụng các lớp có sẵn của QT, vì vậy định dạng QJsonObject có thể dễ dàng được chuyển đổi sang dạng QString và sử dụng.

Qua các bước trên, client đã có thể lấy được danh sách thông tin về các thiết bị đã được đăng ký trước trên server. Trong các thuộc tính thông tin trên, đặc biệt cần lưu ý 2 thuộc tính “id” và “name”. Thuộc tính “id” dùng để định danh thiết bị nào được gắn vào nút nào, thuộc tính “name” cũng có tác dụng tương tự và thêm vào là để hiển thị tên thiết bị tương ứng trên nút bấm, giúp người dùng có thể dễ dàng phân biệt giữa các thiết bị cùng loại.

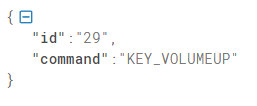
* 1. **API điều khiển thiết bị**
  2. **Phương án thiết kế**

- Ý tưởng: Sử dụng giao thức MQTT để gửi các gói tin chứa thông tin về thiết bị cần điều khiển và hành động cần thực hiện lên server. Khi người dùng lựa chọn một nút bấm có chức năng điều khiển thiết bị. Một MQTT client sẽ được tạo, sau đó gói tin chứa thông tin tương ứng sẽ được tạo và truyền lên server.

- Kết quả thử nghiệm: Chương trình làm việc đúng với mong muốn. Gói tin được gửi thành công lên server, không bị mất mát dữ liệu hay lỗi cho dù với câu lệnh có độ dài dài nhất.

* 1. **Kết quả thiết kế**

Khi người dùng lựa chọn một thiết bị trên màn hình giao diện nhà thông minh, thiết bị đó đã có sẵn “id” được lưu trữ. Sau đó, khi người dùng nhấn các nút điều khiển, payload của các gói tin sẽ được thay đổi theo từng lệnh và “id” tương ứng. Tiếp theo, chương trình sẽ gửi các gói tin trên lên server thông qua phương thức MQTTClient\_publishMessage có sẵn của thư viện MQTTClient.h/C++. Payload sẽ có dạng:



*Hình 5-2: Format của dữ liệu gửi lên server*

Trong đó, “id” là “id” của thiết bị mà người dùng mong muốn điều khiển, “command” là mã lệnh để thực hiện chức năng tương ứng của thiết bị đó. Các command tương ứng với chức năng của thiết bị như sau:

* 1. **Điều khiển đèn**

- Lệnh bật đèn: “command”: “100”

- Lệnh tắt đèn: “command”: “0”

* 1. **Điều khiển quạt**

- Lệnh bật quạt: “command”: “KEY\_POWERON”

- Lệnh tắt quạt: “command”: “KEY\_POWEROFF”

- Lệnh chuyển sang mức một: “command”: “KEY\_LEVEL\_1”

- Lệnh chuyển sang mức gió hai: “command”: “KEY\_LEVEL\_2”  
- Lệnh chuyển sang mức gió ba: “command”: “KEY\_LEVEL\_3”

* 1. **Điều khiển vô tuyến**

- Lệnh bật vô tuyến: “command”: “KEY\_POWERON”

- Lệnh tắt vô tuyến: “command”: “KEY\_POWEROFF”

- Lệnh chuyển kênh tiếp theo: “command”: “KEY\_NEXTCHANNEL”

- Lệnh chuyển kênh trước đó: “command”: “KEY\_PREVCHANNEL”  
- Lệnh tăng âm lượng: “command”: “KEY\_VOLUMEUP”  
- Lệnh giảm âm lượng: “command”: “KEY\_VOLUMEDOWN”

- Lệnh ghi số kênh: “command”: “KEY\_NUM\*”

* 1. **Điều khiển điều hòa**

- Lệnh bật điều hòa: “command”: “KEY\_POWERON”

- Lệnh tắt điều hòa: “command”: “KEY\_POWEROFF”

- Lệnh tăng nhiệt độ: “command”: “KEY\_TEMPUP”  
- Lệnh giảm nhiệt độ: “command”: “KEY\_TEMPDOWN”  
- Lệnh bật/tắt cánh điều gió: “command”: “KEY\_SWING”  
- Lệnh chuyển sang mức một: “command”: “KEY\_LEVEL\_1”

- Lệnh chuyển sang mức gió hai: “command”: “KEY\_LEVEL\_2”  
- Lệnh chuyển sang mức gió ba: “command”: “KEY\_LEVEL\_3”

1. **Kết luận và phương hướng cải tiến**
   1. **Kết luận:**

Trong một hệ thống, giao diện để hệ thống có thể giao tiếp với người dùng là rất quan trọng. Người sử dụng sẽ nhìn vào những hình ảnh trong giao diện đầu tiên trước khi được tiếp xúc với các chức năng của hệ thống. Đôi khi để mất đi thiện cảm ban đầu sẽ rất khó để có thể lấy lại được cho dù sản phẩm có hoạt động tốt và đầy đủ chức năng thế nào. Vì vậy nên việc thiết kế một giao diện phù hợp với thị giác của người sử dụng đóng một vai trò vô cùng quan trọng trong quá trình lấy được thiện cảm của họ từ những giây phút đầu tiên khi tiếp xúc với hệ thống.

Nội dung phía trên của khóa luận đã trình bày quá trình thiết kế giao diện của hệ thống. Đi từ những phản ứng tự nhiên nhất của con người với các sự vật xung quanh để có thể đưa ra nhưng bố cục và hình ảnh tạo nên độ thiện cảm với người dùng nhất có thể. Bên cạnh đó, khóa luận cũng chỉ rõ giao diện phải hoạt động mượt mà, không bị đóng băng, phải điều chỉnh được một số hiển thị phù hợp với mong muốn người dùng. Để làm được như vậy, ngoài việc lập trình có cấu trúc và thuật toán tốt, còn phải sử dụng giao thức hợp lý, tránh việc khi đưa vào sử dụng với dữ liệu lớn có thể bị lỗi hoặc chậm do thiếu băng thông mạng.

Giao diện hiện tại đã có gần như đầy đủ các màn hình hiển thị đã được thiết kế theo mong muốn cũng như là đầy đủ chức năng để có thể đưa vào thực tiễn sử dụng. Tuy nhiên, nhu cầu của con người ngày càng tăng lên theo thời gian cùng với sự tiến bộ của ngành thiết kế. Vì vậy còn cần phải cải tiến hệ thống nói chung và giao thức, giao diện nói riêng rất nhiều trong tương lai.

* 1. **Phương hướng cải tiến:**

Trong tương lai, giao diện và giao thức còn có thể cải tiến ở nhiều phương diện. Nếu hệ thống cho phép điều khiển nhiều hơn 4 thiết bị như hiện tại, giao diện cũng cần phải cải tiến để có thể hiển thị được nhiều thiết bị hơn. Trong phần tùy chỉnh, có thể tăng thêm số lượng ngôn ngữ và màu sắc cho phép hiển thị cũng như là phát triển thêm các khía cạnh thay đổi khác để đáp ứng nhiều hơn nữa nhu cầu chỉnh sửa theo ý thích của người dùng. Không chỉ vậy, khi thế giới dần tiến bộ, có thể phong cách của con người sẽ thay đổi, điều này cũng khiến cho giao diện cần phải cập nhật các văn bản và hình ảnh biểu tượng để theo kịp xu hướng của thời đại.

Giao thức trong tương lai có thể thay đổi nếu tìm được phương án hợp lý hơn khi đưa vào sử dụng trong hệ thống. API cũng có thể được thiết kế tối ưu hóa hơn, các gói tin gửi lên hệ thống có thể được cấu trúc lại để ngắn gọn, nhẹ nhất có thể trong khi vẫn tiếp nhận hoặc gửi đi thông tin đầy đủ.

**TÀI LIỆU THAM KHẢO**

[1] “Introduction to QT – Team Inria Fr”

<https://team.inria.fr/imagine/files/2013/09/Introduction-to-Qt.pdf>

[2] “Introduction to C++ Programming Language – Chinmoy Lenka”

<https://www.geeksforgeeks.org/introduction-to-c-programming-language/>  
[3] “Introduction to Visual Studio – Anshul\_Aggarwal”

<https://www.geeksforgeeks.org/introduction-to-visual-studio/>

[4] “Introducing the MQTT Protocol - MQTT Essentials: Part 1 – HiveMQ Team”

<https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt/>

[5] “What is eye tracking? – Tobii Company”

<https://www.tobii.com/group/about/this-is-eye-tracking/>

[6] “Mind-Body Wellness – Heathwise Staff”

[https://www.uofmhealth.org/health-library/mente#hw162080](https://www.uofmhealth.org/health-library/mente%23hw162080)

[7] “Does colour really affect our mind and body? A professor of colour science explains – Stephen Westland”

<https://theconversation.com/does-colour-really-affect-our-mind-and-body-a-professor-of-colour-science-explains-84382>

[8] “Web Layout Best Practices: 12 Timeless UI Patterns Analyzed – Cameron Chapman”

<https://www.toptal.com/designers/ui/web-layout-best-practices>

[9] “How MQTT Works -Beginners Guide – Steve”

[http://www.steves-internet-guide.com/mqtt-works/](http://www.steves-internet-guide.com/mqtt-works/%20)

[10] “OOP là gì? 4 đặc tính cơ bản của OOP – Itviec”

<https://itviec.com/blog/oop-la-gi/>

[11] “Positioning in front of an eye tracker – Tobii Company”

<https://help.tobii.com/hc/en-us/articles/210250305-Positioning-in-front-of-an-eye-tracker>