**TỔ CHỨC THỰC HIỆN: NHÓM 10**

**Dự án: Hệ thống SmartHome**

Giới thiệu thành viên của dự án

|  |  |
| --- | --- |
| (ảnh) | Họ và tên: Phùng Minh Đức  MSSV: 20191766  Phụ trách công việc |
| (ảnh) | Họ và tên: Cao Mạnh Huy  MSSV: 20191889  Phụ trách công việc |
| (ảnh) | Họ và tên: Chử Văn Minh  MSSV: 20191958  Phụ trách công việc |

Yêu cầu dự án

* Thiết bị Device (vitrual hoặc physic): MQTT,  CoAP (cảm biến ánh sáng, nhiệt độ cần low power).
* Gateway (edge) (sử dụng HTTP để kết nối tới cloud).
* Cloud (server) (có thể sử dụng cloud có sẵn).
* App (mobile) (tính năng nâng cao, có thể làm trên điện thoại, hoặc đơn giản chỉ cần dùng trình duyệt để điều khiển).
* Sử dụng truyền thông Wifi kết nối Devices với Gateway.
* Điều khiển local và điều khiển qua Internet (khi không có Internet thiết bị trong mạng vẫn có thể điều khiển local được).
* Xây dựng các tính năng của hệ thống smarthome (sinh viên tùy vào năng lực của nhóm xây dựng các kịch bản, có thể bổ sung nâng cấp, trên cơ sở kiến trúc không đổi).
* Đánh giá hiệu năng hệ thống (số bản tin trao đổi,...).

Nhật ký

Tuần 9

Tổ chức

* Địa điểm: Microsoft Teams
* Thời gian: 25/11/2022
* Tham dự: 3/3

Nội dung cuộc họp

* Nội dung 1: Xây dựng các tính năng của hệ thống.
* Nội dung 2: Xây dựng kiến trúc hệ thống.
* Nội dung 3: Tìm hiểu về giao thức MQTT, HTTP.

Nhiệm vụ từng thành viên

|  |  |  |
| --- | --- | --- |
| Thành viên | Nhiệm vụ | Kết quả |
| Phùng Minh Đức | - Xây dựng các tính năng, kiến trúc hệ thống  - Tìm hiểu giao thức MQTT | Báo cáo |
| Chử Văn Minh | - Xây dựng các tính năng, kiến trúc hệ thống  - Tìm hiểu giao thức MQTT | Báo cáo |
| Cao Mạnh Huy | - Xây dựng các tính năng, kiến trúc hệ thống  - Tìm hiểu về giao thức HTTP | Báo cáo |

Báo cáo thực hiện nhiệm vụ của các thành viên được lưu trữ trên Github qua link dưới đây.

**Link Github:** https://github.com/minhduc21092001/Networks-IoT-Protocols-

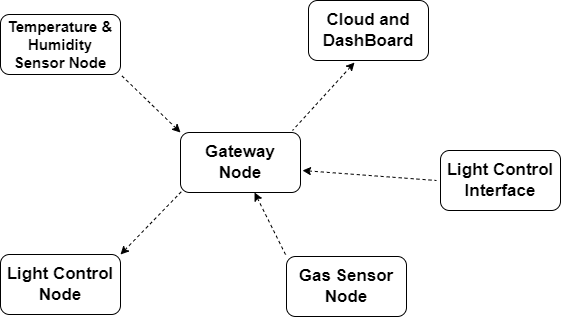
Báo cáo

**1. Xây dựng các tính năng của hệ thống**

Hệ thống SmartHome nhóm đề xuất sẽ có các tính năng sau:

* Điều khiển bật/tắt đèn thông qua Internet.
* Giám sát nhiệt độ và độ ẩm trong nhà.
* Giám sát nồng độ khí gas trong nhà.

**2. Thiết kế mô hình hệ thống**



Hình : Kiến trúc hệ thống SmartHome

**3. Tìm hiểu về giao thức MQTT và HTTP**

**3.1 MQTT**

MQTT (Message Queueing Telemetry Transport) là một giao thức mạng kích thước nhỏ (lightweight), hoạt động theo cơ chế publish – subscribe (tạm dịch: xuất bản – đăng ký) theo tiêu chuẩn ISO (ISO/IEC 20922) và OASIS mở để truyền tin nhắn giữa các thiết bị.

Giao thức này hoạt động trên nền tảng TCP/IP. MQTT được thiết kế cho các kết nối cho việc truyền tải dữ liệu cho các thiết bị ở xa, 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.

**3.1.1 MQTT Broker**

MQTT Broker hay máy chủ mô giới được coi như trung tâm, nó là điểm giao của tất cả các kết nối đến từ Client (Publisher/Subscriber).

Nhiệm vụ chính của Broker là nhận thông điệp (message) từ Publisher, xếp vào hàng đợi rồi chuyển đến một địa điểm cụ thể. Nhiệm vụ phụ của Broker là nó có thể đảm nhận thêm một vài tính năng liên quan tới quá trình truyền thông như: bảo mật message, lưu trữ message, logs, ….

**3.1.2 MQTT Client**

Là các thiết bị/ứng dụng Client kết nối đến Broker để thực hiện truyền nhận dữ liệu .Client thì được chia thành hai nhóm là Publisher và Subscriber. Một Client có thể có 1 trong 2 nhiệm vụ hoặc cả 2.

Publisher là thiết bị gửi bản tin lên broker

Subscriber là người nhận bản tin mỗi khi có bản tin mới gửi lên Broker.

**3.1.3 Cơ chế hoạt động của giao thức MQTT**

Một phiên MQTT được chia thành bốn giai đoạn: kết nối, xác thực, giao tiếp và kết thúc.

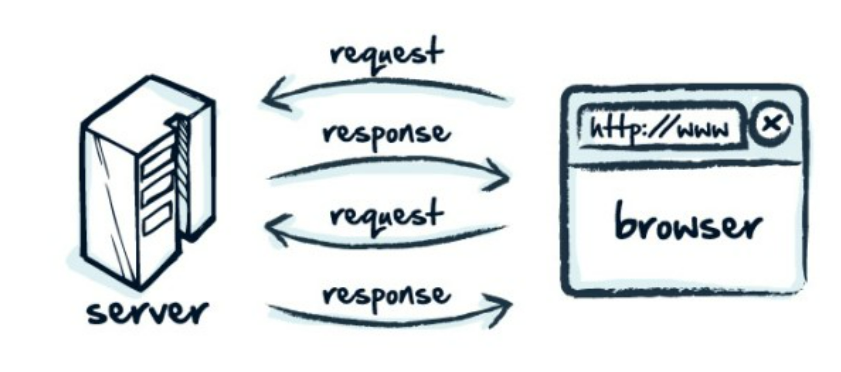
1. Client (máy khách) bắt đầu bằng cách tạo kết nối Transmission Control Protocol/Internet Protocol (TCP/IP) tới broker bằng cách sử dụng cổng tiêu chuẩn hoặc cổng tùy chỉnh được xác định bởi các nhà phát triển broker.
2. Các cổng tiêu chuẩn là 1883 cho giao tiếp không mã hóa và 8883 cho giao tiếp được mã hóa – sử dụng Lớp cổng bảo mật (SSL) / Bảo mật lớp truyền tải (TLS). Trong quá trình giao tiếp SSL/TLS, máy khách cần kiểm chứng và xác thực máy chủ.
3. Sau đó, Client sẽ gửi bản tin lên broker nếu là Publisher hoặc nhận bản tin từ broker về nếu là Subscriber. Quá trình kết nối này sẽ được giữ đến khi Kết thúc kết nối.
4. Sau khi kết thúc để có thể truyền nhận MQTT, chúng ta lại tiếp tục quay lại các bước trên.

**3.2 HTTP**

**3.2.1 Giới thiệu HTTP**

Http là giao thức để thông qua đó, máy chủ và máy khách giao tiếp với nhau.

Http là chữ viết tắt của HyperText Transfer Protocol (giao thức truyền tải siêu văn bản). Đây là một giao thức ứng dụng được sử dụng thường xuyên nhất trong bộ các giao thức TCP/IP (gồm một nhóm các giao thức nền tảng cho internet).

Http hoạt động dựa trên mô hình Client (máy khách) – Server (máy chủ). Các máy tính của người dùng sẽ đóng vai trò làm máy khách (Client). Sau một thao tác nào đó của người dùng, các máy khách sẽ gửi yêu cầu đến máy chủ (Server) và chờ đợi câu trả lời từ những máy chủ này.

Hình 2. Giao thức HTTP

**3.2.2 Mô hình giao thức HTTP**

Để dễ hình dung, mỗi khi người dùng sử dụng trình duyệt và truy cập vào một website, một phiên làm việc HTTP (gọi là Session) sẽ được diễn ra với client là máy tính của người dùng và serer là máy chủ của website. Mặc định HTTP sẽ được thực hiện thông qua port 80, đây là port chuẩn của giao thức được định nghĩa bởi tổ chức [IANA](https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml) quy định.

Quá trình của một phiên làm việc HTTP diễn ra như sau:

1. HTTP client thiết lập một kết nối TCP đến server. Nếu thiết lập thành công, client và server sẽ truyền nhận dữ liệu với nhau thông qua kết nối này, kết nối được thiết lập còn gọi là socket interface bao gồm các thông tin: địa chỉ IP, loại giao thức giao vận (chính là TCP), và port (mặc định là 80).
2. Sau khi kết nối thành công, client gửi một HTTP request đến server thông qua socket interface vừa được thiết lập.  Trong gói tin request sẽ chứa đường dẫn yêu cầu (path name) là /category/internet-of-thing.
3. Server sẽ nhận và xử lý request từ client thông qua socket, sau đó đóng gói dữ liệu tương ứng và gửi một HTTP response về cho client. Dữ liệu trả về sẽ là một file HTML chứa các loại dữ liệu khác nhau như văn bản, hình ảnh,…
4. Server đóng kết nối TCP.
5. Client nhận được dữ liệu phản hồi từ server và đóng kết nối TCP.