TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN ĐIỆN

BÁO CÁO THIẾT KẾ HỆ THỐNG NHÚNG

ĐỀ TÀI:

HỆ THỐNG CHIẾU SÁNG TỰ ĐỘNG THEO KỊCH BẢN

Sinh viên thực hiện:
- Lê Công Tráng 20164195
- Trần Tiến Quân 20163391
- Vũ Thanh Bình 20160378
- Lê Đức Đạt 20150829

Giảng viên hướng dẫn: PGS.TS Nguyễn Quốc Cường

Nội dung

Nhật ký công việc:

Tuần	Nội dung công việc
1	Tìm hiểu, xác định mục đích, tiêu chí của đề tài. Thiết kế sơ đồ tổng quan hệ thống
2	Thiết kế sơ đồ khối, flowchart diagram. Tìm hiểu các phần cứng phù hợp với đề tài.
3	Tìm hiểu về giao thức MQTT, Cảm biến ánh sáng BH1750, MCU AT89S52, Module BLE HC05
4	Tìm hiểu về Cloud MQTT Broker, Chuẩn giao tiếp I2C, UART, Bluetooth
5	Xây dựng thư viện phần cứng HC05, BH1750, LCD16x2. Tìm hiểu cách kết nối tới MQTT trên Linux
6	Mô phỏng phần cứng bằng Proteus. Xây dựng thuật toán cho các kịch bản chiếu sáng
7	Viết báo cáo slide, làm video giới thiệu về đề tài, viết contents
8	Viết Chương trình xử lí trên vi điều khiển, Chương trình Gateway trên máy tính Linux
9	Triển khai thử nghiệm đề tài, khắc phục những lỗi phát sinh, điều chỉnh một số dòng code
10	Tổng hợp, đánh giá đề tài. Hoàn thành nội diung báo cáo. Chuẩn bị thuyết trình

Phân công công việc:

Thành viên	Công việc	Đóng góp
Vũ Thanh Bình	Thiết kế phần mềm cho vi xử lí, tìm hiểu về ánh sáng, cloud MQTT	26%
Lê Công Tráng	Thiết kế phần mềm, phần cứng, code chương trình gateway xử lí dữ liệu trên Linux	26%
Trần Tiến Quân	Thiết kế phần cứng, mô phỏng trên phần mềm proteus, vẽ mạch nguyên lý	26%
Lê Đức Đạt	Tìm hiểu các chuẩn giao tiếp: I2C, UART, Bluetooth	22%

IOT: Hệ thống chiếu sáng tự động theo kịch bản

Lời nói đầu

Nhắc tới chiếu sáng rất nhiều người trong chúng ta thường lập tức nghĩ ngay tới việc tìm đến công tắc và bật đèn lên. Nhưng ẩn sâu trong đó có rất nhiều các tiêu chuẩn quan trọng để xây dựng một hệ thống chiếu sáng như: độ rọi, chỉ số màu, màu sắc ánh sáng,.. Việc thiết kế một hệ thống chiếu sáng đạt chuẩn rất phức tạp, phải trải qua nhiều quá trình nghiên cứu đo đạc cầu kì. Sự thật không phải hệ thống chiếu sáng trong nhà, khu công nghiệp nào cũng đạt tiêu chuẩn . Không phải mô hình chiếu sáng nào cũng có thể đáp ứng được các tiêu chuẩn chiếu sáng.

Ngày nay với sự phát triển của khoa học công nghệ. Chúng ta cần quan tâm nhiều hơn đến các tiêu chí chiếu sáng, khả năng tiết kiệm năng lượng. Các hệ thống chiếu sáng đang dần được tự động hóa, thân thiện với người dùng và tiết kiệm năng lượng. Để góp phần vào việc xây dựng các hệ thống chiếu sáng thông minh. Đề tài “Hệ thống chiếu sáng tự động theo kịch bản ” được hình hình thành và phát triển.

Lời cảm ơn

Để thiết kế và xây dựng đề tài này chúng tôi đã trải qua rất nhiều quá trình nghiên cứu và học tập. Những khó khăn trong việc tìm hiểu phần cứng, xây dựng chương trình cho các module, triển khai hệ thống thử nghiệm… Trong những giai đoạn này chúng tôi được sự giúp đỡ rất nhiều từ các Thầy/Cô, bạn bè, cộng đồng internet. Đặc biệt chúng tôi xin gửi lời cảm ơn tới thầy giáo PGS.TS **Nguyễn Quốc Cường** đã hướng dẫn và đồng hành cùng trong suốt quá trình xây dựng đề tài.

Xin cảm ơn!.

Chương 1. Mục đích đề tài

1.1 Mục đích

Chiếu sáng có thể chiến tới 10-38% tổng hóa đơn thanh toán tiền điện trong bất cứ thành phố nào trên thế giới. Điều đáng quan tâm nhất trong công nghệ cho lĩnh vực này hiện tại là tự động hóa, tiêu thụ năng lượng và giá thành hợp lý. Tự động hóa có xu hướng giảm sức người với sự giúp đỡ của trí tuệ nhân tạo. Tiết kiệm năng lượng là điều đáng bàn nhất vì nó là nguồn năng lượng có hạn do nhiều lý do khác nhau.
Việc chiếu sáng thực tế đời sống hiện nay vẫn còn đang thủ công. Tức là phần đa trong gia đình, văn phòng, nhà máy… Hệ thống chiếu sáng vẫn đang còn rất đơn giản và bằng tay. Dù là hoạt động gì cũng chỉ bật tắt không cần biết cường độ sáng có phụ hợp hay không .

Độ sáng ảnh hưởng rất lớn tới thị giác của con người. Cao quá sẽ gây hại cho mắt và lãng phí năng lượng. Thấp quá không nhưng ảnh hưởng tới thị giác mà còn làm giảm năng suất làm việc. Độ rọi tiêu chuẩn cho việc đọc sách là 500-550 lux, nghiên cứu 700-750 lux…, Đối với cấc công việc mang tính chất tỉ mỉ, chi tiết như chế tác mỹ ký, chi tiết đồng hồ… độ rọi tối thiểu lên tới 1000 lux. Do tính chất này chúng ta cần một hệ thống có chiếu sáng theo các công việc cụ thể với các giá trị độ rọi đã đưa ra. Hoặc có thể tùy chỉnh theo ý muốn.

Vậy mục đích chính của đồ án là xây dựng một hệ thống chiếu sáng tự động theo kịch bản.

Dựa trên những tiêu chí :

- Thân thiện với người dùng, dễ sử dụng
- Tự động điều chỉnh cường độ sáng phù hợp
- Tiết kiệm năng lượng, giá thành hợp lí
- Tương thích với các hệ thống IoT, Smart Home

1.2 Giới thiệu

- **Thế nào là hệ thống chiếu sáng tự động theo kịch bản?**

Các kịch bản sẽ được các nhà thiết kế xây dựng sẵn. Tương ứng với mỗi kịch bản sẽ có một dải độ rọi phù hợp. Ví dụ: xem phim, giải trí , đọc sách, ngủ …

Lảm sao để chọn các kịch bản đó? Người dùng sẽ trực tiếp chọn kịch bản theo ý muốn thông qua smartphone, laptop.. Các kịch bản sẽ được mã hóa và xử lí ở khối gateway, tín hiệu điều khiển sẽ được khối gateway gửi về khối chiếu sáng từ đó điều chỉnh độ rọi phù hợp.

Để thuận tiện cho quá trình thao tác và giao tiếp với hệ thống. Công nghệ điện toán đám mây sẽ chịu trách nhiệm trao đổi thông tin giữa người dùng và hệ thống thông qua internet. Người dùng có thể điều khiển hệ thống ở bất cứ đâu, chỉ cần có kết nối internet.

Về cơ bản luồng hoạt động của hệ thống sẽ là:

Người dùng chọn kịch bản từ smartphone, laptop .. có kết nối internet. Thông điệp mã hóa kịch bản đó sẽ được gửi tới Cloud. Ngay khi nhận được thông điệp đó cloud sẽ gửi đến khối gateway (khối xử lí và gateway sẽ được tích hợp cùng nhau). Tại khối gateway hệ thống sẽ giải mã thông điệp đó. Nếu nó đúng là những kịch bản thì sẽ được xử lí, tùy theo những kịch bản khối gateway sẽ gửi những tín hiệu điều khiển khác nhau tới khối chiếu sáng. Khối chiếu sáng sẽ tự động thực hiện những công việc được lập trình sẵn như tăng giảm, điều chỉnh độ sáng và có phản hồi giá trị cường độ ánh sáng tới khối xử lí, cho đến khi cường độ ánh sáng phù hợp với kịch bản đã cho.

Một trong những **tính năng đặc biệt của hệ thống** đó là duy trì khả năng tự động điều chỉnh độ sáng theo kịch bản đã đưa ra. Có nghĩa là một khi kịch bản đã được chọn, hệ thống tự điều khiển tăng giảm công suất của bóng đèn bám theo kịch bản đó. Nguồn sáng độc lập với hệ thống (ánh sáng mặt trời, đèn của hệ thống khác…) tăng lên thì hệ thống sẽ giảm công suất đèn xuống và ngược lại. Nguồn sáng độc lập giảm xuống, hệ thống sẽ tăng công suất đèn lên. Đặc tích này hệ thống sẽ giúp người dùng tiết kiệm được năng lượng chiếu sáng. Duy trì độ rọi cần thiết cho các hoạt động, tăng mức độ tập trung công việc.

Chương 2. Cơ sở lý thuyết

2.1 Ánh sáng và cường độ ánh sáng

a. Định nghĩa

Ánh sáng là từ phổ thông dùng để chỉ các bức xạ điện từ có bước sóng nằm trong vùng quang phổ nhìn thấy được bằng mắt thường của con người (tức là từ khoảng 380 nm đến 700 nm)

Tên	Bước sóng	Tần số (Hz)	Năng lượng photon (eV)
Tia gamma	≤ 0,01 nm	≥ 30 EHz	124 keV - 300+ GeV
Tia X	0,01 nm - 10 nm	30 EHz - 30 PHz	124 eV - 124 keV
Tia tử ngoại	10 nm - 380 nm	30 PHz - 790 THz	3.3 eV - 124 eV
Ánh sáng nhìn thấy	380 nm- 700 nm	790 THz - 430 THz	1.7 eV - 3.3 eV
Tia hồng ngoại	700 nm - 1 mm	430 THz - 300 GHz	1.24 meV - 1.7 eV
Vi ba	1 mm - 1 met	300 GHz - 300 MHz	1.7 eV - 1.24 meV
Radio	1 mm - 100000 km	300 GHz - 3 Hz	12.4 feV - 1.24 meV

b. Đơn vị đo độ rọi ánh sáng. (lux)

Cường độ sáng (cd) là đại lượng quang học cơ bản dùng trong việc đo thông số nguồn sáng. Là năng lượng phát ra 1 nguồn ánh sáng trong 1 hướng cụ thể và được tính như sau:

1 candela là cường độ mà một nguồn sáng phát ra 1 lumen đẳng hướng trong một góc đặc.

Quang thông(lumen)(cd/m^2) là đại lượng trắc quang cho biết công suất bức xạ của chùm ánh sáng phát ra từ một nguồn phát sáng điểm. Đơn vị của quang thông trong các hệ đơn vị SI, Quang thông (F) là đại lượng đo công suất phát sáng của 1 nguồn sáng.

Độ rọi (cd/m^4) là đại lượng đặc trưng cho bề mặt được chiếu sáng, là mật độ quang thông trên bề mặt có diện tích S. Có nghĩa là mật độ quang thông của một nguồn sáng 1 lumen trên diện tích 1m^2 . Khi mặt được chiếu sáng không đều độ rọi được tính bằng trung bình đại số của độ rọi các điểm.

- Vào buổi tối : 0.001 - 0.02 Lux
- Ánh trăng : 0.02 - 0.3 lux
- Trời nhiều mây trong nhà : 5 - 50 lux
- Trời nhiều mây ngoài trời : 50 - 500 lux
- Trời nắng trong nhà : 100 - 1000 lux
- Ánh sáng cần thiết để đọc sách: 50 - 60 lux
- Lớp học: 300 – 400 lux
- Phòng làm việc: 300-500 lux
- Đường phố về đêm: 20-50 lux

2.2 Tiêu chuẩn chiếu sáng

Các giá trị độ rọi đưa ra trong Điều 5(**Tiêu chuẩn Quốc gia, TCVN 7114 – 1 : 2008, ISO 8995 – 1 : 2002**) là độ rọi duy trì bao trùm khu vực làm việc trên bề mặt chuẩn có thể là bề mặt ngang, bề mặt đứng hoặc bề mặt nghiêng. Độ rọi trung bình tại mỗi bề mặt làm việc phải không nhỏ hơn giá trị đưa ra trong Điều 5 bất kể độ tuổi và điều kiện lắp đặt. Các giá trị hợp lý cho điều kiện làm việc thị giác bình thường và tính tới các yếu tố sau:

- Các yêu cầu đối với công việc thị giác
- An toàn
- Các khía cạnh tâm
- Sinh lý như tiện nghi thị giác và dễ chịu
- Tiết kiệm
- Kinh nghiệm thực tế

Nếu điều kiện nhìn khác biệt với điều kiện được giả định, giá trị độ rọi có thể điều chỉnh, bằng cách tăng lên hoặc giảm xuống ít nhất một bậc trong thang độ rọi. Giá trị độ rọi phải được tăng lên khi:

- Làm việc với độ tương phản thấp
- Công việc thị giác yêu cầu tập trung cao
- Nếu bị lỗi sẽ gây thiệt hại lớn
- Độ chính xác và năng suất cao là rất quan trọng
- Thị lực của người làm việc dưới mức bình thường

Giá trị độ rọi duy trì có thể giảm khi:

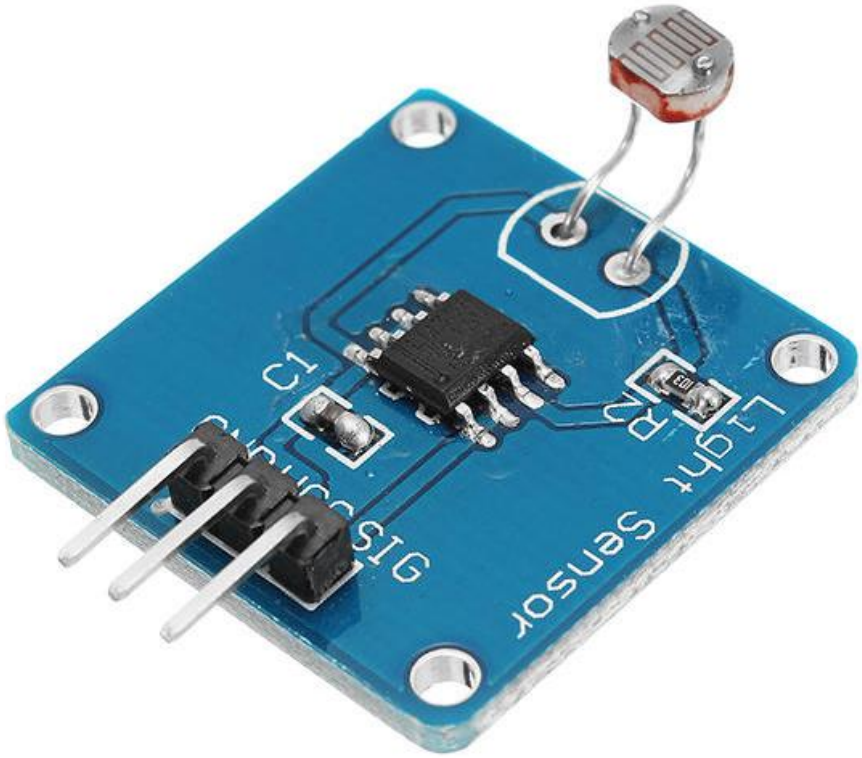
- Chi tiết quan sát có kích thước lớn hoặc độ tương phản cao,
- Công việc thực hiện trong thời gian ngắn Trong vùng mà công việc được tiến hành liên tục, độ rọi duy trì không được nhỏ hơn 200 lux.

2.3 Cảm biến ánh sáng

Là các thiết bị quang điện chuyển đổi năng lượng ánh sáng (photon) cho dù ánh sáng nhìn thấy được hay tia hồng ngoại thành tín hiệu điện (electron). Một trong những cảm biến phổ thông và rẻ nhất phải kể đến cảm biến ánh sáng dùng điện trở quang. Nó được làm bằng cadmium sulfide (CdS) nhạy với ánh sáng nhìn thấy và cả hồng ngoại. Điện trở của CdS tỷ lệ nghịch với ánh sáng chiếu lên nó.

Khi đặt 1 điện trở quang lên một mạch chia điện áp, sự thay đổi về tỷ lệ ánh sáng có thể đo được bằng cách đo điện áp. Cảm biến ánh sáng giúp đo độ rọi của ánh sáng bằng việc đo độ phân bố năng lượng tồn tại trên 1 diện tích, ánh sáng đo là ánh sáng có tần số từ hồng ngoại tới nhìn ánh sáng có tần số nhìn thấy, thậm trí cả phổ tia cực tím.

Cảm biến ánh sáng là thiết bị động chuyển năng lượng ánh sáng dù phổ nhìn được hay ở dạng hồng ngoại thành tín hiệu điện ở đầu ra. Các cảm biến ánh sáng thường được biết đến như là thiết bị điện quang hoặc cảm biến quang bởi vì nó biến đổi năng lượng ánh sáng thành điện.



Thiết bị quang điện có thể được chia làm 2 loại, một là sử dụng điện khi chiếu sáng, như Photo-voltaic hay Phôt-emissive. Hai là thay đổi giá trị điện theo một cách nào đó chẳng hạn như Photo-resistors hoặc Photo-conductors. Điều này dẫn tới phân loại các thiết bị điện

- Điện trở phụ thuộc ánh sáng (LDR)

- Là điện trở bán dẫn làm từ cadmium sulphide có khả nawg=ng thay đổi điện trở điện từ hàng ngàn Ohms trong bóng tối tới chỉ vài trăm Ohms khi chiếu sáng.

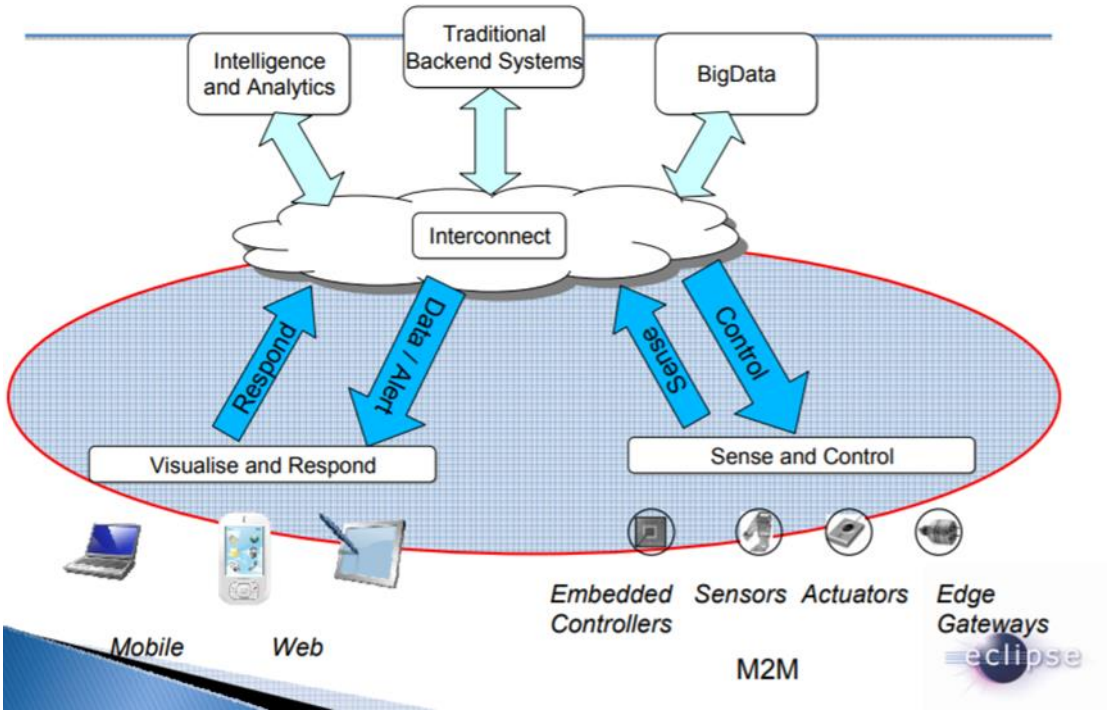
- Cảm biến môi trường (ALS)

- Có khả năng sấp xỉ với mắt con người trong việc nhận diện cường độ ánh sáng với điều kiện chiếu sáng khác nhau và theo sự biến đổi mỏng đi của chất liệu.

- Sự khác biệt giữa LDR và ALS

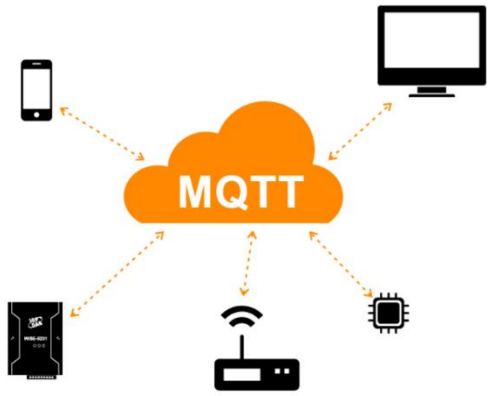
- LDR là một dạng của ALS, nó được kết nối với một rơ le hoặc công tắc bán dẫn để mà có thể điều khiển năng nượng cần thiết. Với một LDR chúng ta có thể đọc ánh sáng một trường, thâm trí nó không chính xác 100% và phù hợp cho việc đo. Do đó LDR phù hợp với thiết bị dò ánh sáng hơn.

2.4 Giao thức MQTT



2.4.1 MQTT là gì?

- **MQTT** là viết tắt của **Message Queuing Telemetry Transport** là một trong nhưng giao thức được sử dụng phổ biến trong các dự án IoT.



- Đây là một giao thức truyền thông điệp (message) theo mô hình Publish/Subcribe (xuất bản-theo dõi), sử dụng băng thông thấp, độ tin cậy cao và có khả năng hoạt động trong điều kiện đường truyền thông không ổn định.

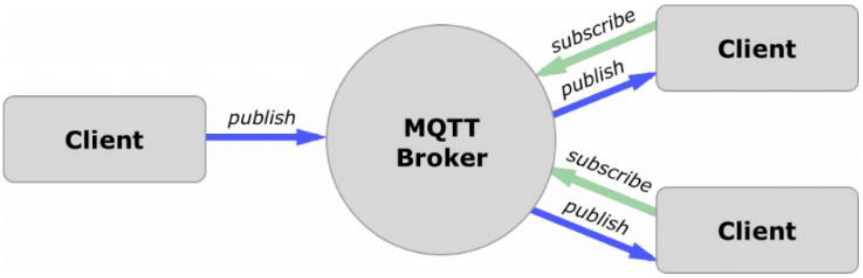
*** ***Tại sao MQTT lại thông dụng trong IoT:

- MQTT có những đặc điểm nổi trội mà ta có thể thấy trong hầu hết các giao thức khác:

- MQTT là 1 giao thức nhẹ. Do đó, ta dễ dàng thêm vào phần mềm và nhanh trong việc truyền dữ liệu.
- MQTT dựa trên công nghệ truyền tin nên việc truyền đi tin nhắn là rất nhanh.
- Tối thiểu các gói dữ liệu do đó sử dụng mạng lưới thấp.
- Sử dụng năng suất thấp nên thiết bị có khả năng tiết kiệm pin.
- Chạy trên thời gian thực nên nó là hoàn hảo cho hoàn hảo cho các dự án IoT.

2.4.2 MQTT hoạt động như thế nào ?

- Giống như các giao thức internet khác, MQTT cũng dựa trên các client và 1 server. Trong đó, server chịu trách nhiện cho việc nắm dữ yêu cầu nhận hoặc gửi dữ liệu của client.



- Server MQTT được gọi là 1 broker và các client là các thiết bị kết nối. Do đó:
    - Khi 1 thiết bị (client) muốn gửi dữ liệu tới broker, ta gọi quá trình này là 1 "publish".
    - Khi 1 thiết bị(client) muốn nhận dữ liệu từ broker -> "subscribe".

2.4.3 Tổng quan các thành phần của MQTT:

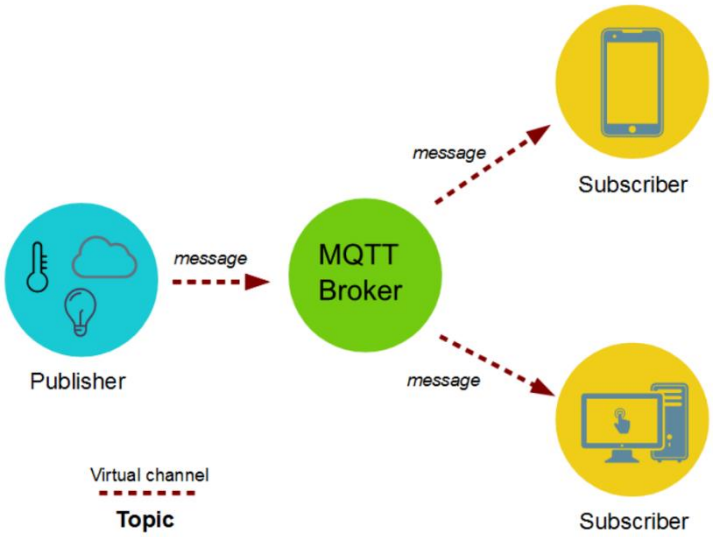
a. *BROKER*: là server có chức năng điều khiển truyền dữ liệu giữa các client.

b. *TOPIC*: là nơi là thiết bị muốn đặt tin nhắn vào hoặc lấy nó ra.

c. *MESSAGE*: là dữ liệu mà 1 thiết bị nhận khi Subscribe từ 1 Topic hoặc gửi khi Publish tới 1 Topic.

d. *PUBLISH*: là quá trình 1 thiết bị gửi Message tới Broker.

e. *SUBSCRIBE*: là quá trình ở đó 1 thiết bị nhận Message từ Broker.



2.4.4 Subscribe/Publish

Khái niệm đầu tiên phải kể đến là hệ thống Publish và Subscribe. Trong 1 hệ thống Publish và Subscribe, 1 thiết bị có thể xuất 1 tin nhắn vào 1 chủ đề hoặc nó được theo dõi tới 1 chủ đề cụ thể để nhận được tin nhắn.

- Thiết bị 1 Publish vào 1 topic
- Thiết bị 2 được Subscribe tới cùng 1 topic như thiết bị 1
- Nên thiết bị 2 nhận được tin nhắn.

****QoS**

Ở đây có 3 tuỳ chọn **QoS (Qualities of service)** khi "publish" và "subscribe":

- **QoS0** Broker/client sẽ gởi dữ liệu đúng 1 lần, quá trình gởi được xác nhận bởi chỉ giao thức TCP/IP, giống kiểu đem con bỏ chợ.
- **QoS1** Broker/client sẽ gởi dữ liệu với ít nhất 1 lần xác nhận từ đầu kia, nghĩa là có thể có nhiều hơn 1 lần xác nhận đã nhận được dữ liệu.
- **QoS2** Broker/client đảm bảm khi gởi dữ liệu thì phía nhận chỉ nhận được đúng 1 lần, quá trình này phải trải qua 4 bước bắt tay.

2.4.5 Messages

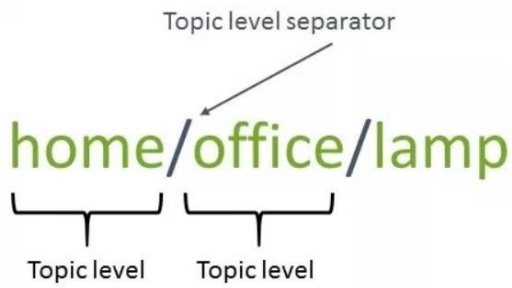
- Trong giao thức MQTT, message còn được gọi là “message payload”, có định dạng mặc định là plain-text(chữ viết người đọc), tuy nhiên người sử dụng có thể cấu hình thành các định dạng khác.

- Messages là các thông tin mà ta muốn trao đổi giữa các thiest bị. Nó là yêu cầu hoặc dữ kiệu.

2.4.6 Topics

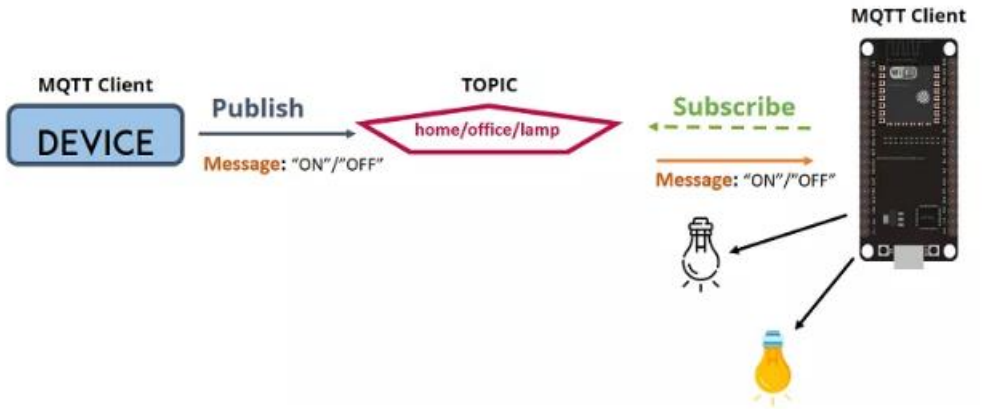
-Topics là đường dẫn ta đăng kí việc quan tâm cho các Message tiếp theo hoặc làm sao để phân loại chỗ nào mà ta muốn Publish Message.

- Topics là các chuỗi phân cách bởi gạch chéo “/”. Mỗi dấu gạch lại chỉ 1 mức độ của Topic. Ví dụ:



- Topic có thể coi như một "đường truyền" logic giữa 2 điểm là publisher và subscriber. Về cơ bản, khi message được publish vào một topic thì tất cả những subscriber của topic đó sẽ nhận được message này.

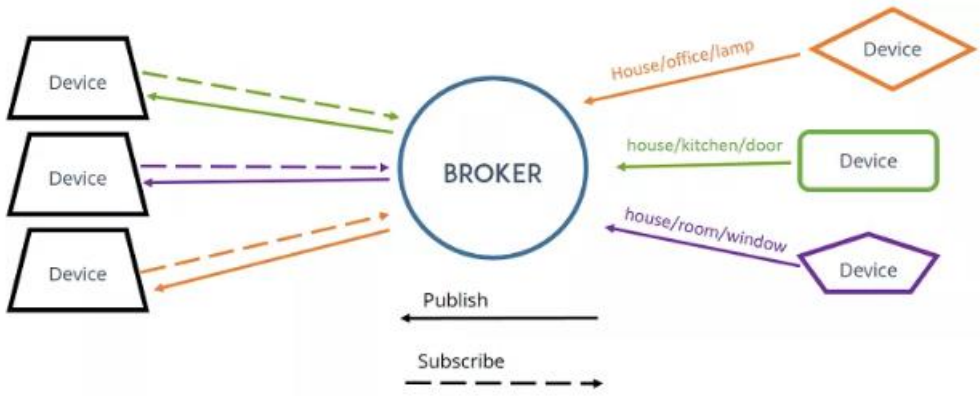
- Nếu ta muốn bật đèn ở trong tòa nhà công sở mà có sử dụng MQTT, có thể hình dung như tình huống dưới đây:



1) Ta có 1 thiết bị Publish Message "on" và "off" vào Topic "home/office/lamp".
2) Ta cũng có 1 thiết bị điều khiển đèn(ESP32 như trong hinh). Thiết bị này điều khiển đèn, và được Subscribe tới topic "home/office/lamp".
3) Do đó, khi một Message được Publish vào Topic đó, ESP32 nhận Message "on" hoặc "off" và bật hoặc tắt đèn.

2.4.7 Broker

a. Nhiệm vụ của Broker



- Broker chủ yếu quản lý việc nhận tất cả các Message, lọc chúng, xác định người quan tâm các Message và Publish Message đó tới tất cả các Client đã Subscribe.
- Lượng thiết bị (client) được kết nối tới broker dựa vào người cung cấp trình phục vụ Broker.

- Những app dựa trên Broker có thể nhanh chóng chia sẻ dữ liệu mà không giới hạn.

b. Mosquitto broker

Mosquitto là một Message Broker mã nguồn mở thực hiện theo giao thức MQTT. Nó thích hợp cho các thiết bị từ bảng mạch công suất thấp như Arduino tới cả máy tính và các server.

Thay vì sử dụng Mosquito trên máy tính cục bộ, ta sẽ cần sử dụng server đám mây đã bao hàm Mosquitto Broker. Điều này cần thiết cho việc làm dự án IoT mà được điều khiển qua interet.

Các Mosquitto Broker đám mây:

- ThingMQ
- ThingStudio
- MQTT.io
- CloudMQTT

**** CloudMQTT Broker**

- Là một trong những Mosquitto Broker đám mây dễ và tốt nhất.
- Có những kế hoạch miễn phí cho phép xây dựng CloudMQTT broker nhanh chóng, có thể hoạt động ngay trên server phần cứng. Do đó, ta có có một Broker online luôn sẵn để sử dụng trong các dự án IoT.
- Nó cũng có GUI được thiết kế rất tốt để điều khiển việc Publish và Subscribe và các Topic.

2.4.8 Tại sao không sử dụng giao thức HTTP để chia sẻ dữ liệu giữa các thiết bị?

- Giao thức HTTP chậm hơn, tiêu thụ năng lượng nhiều hơn MQTT:

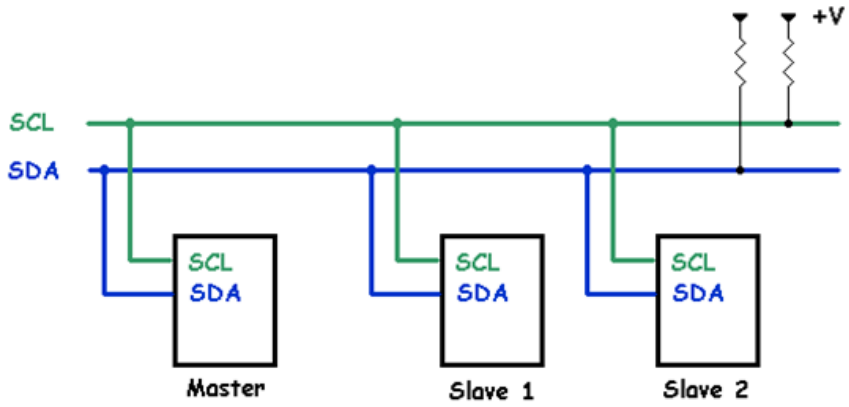
- Chậm: Vì nó dùng gói dữ liệu lớn hơn để giao tiếp với server.
- Overhead: HTTP yêu cầu mở và đóng kết nối ở mỗi yêu cầu, trog khi đó giữ trạng thái online để kênh luôn luôn mở giữa Broker “server” và Client.
- Công suất tiêu thụ: vì HTTP đòi hỏi nhiều thời gian hơn và nhiều gói dữ liệu hơn, do đó nó sử dụng nhiều công suất hơn.

		3G		Wifi	
		HTTPS	MQTT	HTTPS	MQTT
receive	messages / hour	1,708	160,278	3,628	263,314
	% battery / msg	0.01709	0.00010	0.00095	0.00002
	msgs (note losses)	240 / 1024	1024 / 1024	524 / 1024	1024 / 1024
send	msg / hour	1,926	21,685	5,229	23,184
	% battery / msg	0.00975	0.00082	0.00104	0.00016

2.5 Các chuẩn giao tiếp sử dụng trong đề tài

2.5.1 Giao tiếp I2C

- Là một chuẩn giao tiếp nối tiếp 2 dây
- Đây là đường Bus giao tiếp giữa các IC với nhau
  Bus I2C được sử dụng làm bus giao tiếp ngoại vi cho rất nhiều loại IC khác nhau như các loại Vi điều khiển 8051, PIC, AVR, ARM…,chip nhớ như: RAM tĩnh (Static Ram), EEPROM, bộ chuyển đổi tương tự số (ADC), số tương tự(DAC), IC điều khiển LCD, LED...



Bus I2C và các thiết bị ngoại vi

a. Cấu tạo:

I2C sử dụng hai đường truyền tín hiệu

- Một đường xung nhịp đồng hồ(SCL) chỉ do Master phát đi (thông thường ở 100kHz và 400kHz. Mức cao nhất là 1Mhz và 3.4MHz).
- Một đường dữ liệu(SDA) theo 2 hướng.
  2 chân này luôn hoạt động ở chế độ mở, vì vậy để sử dụng được cần phải có trở kéo. tức là nối +5v => trở => I2C bởi các thiết bị trên bus i2c hoạt động ở mức thấp. Giá trị của các điện trở này khác nhau tùy vào từng thiết bị và chuẩn giao tiếp, thường dao động trong khoảng 1K đến 4.7k

b. Nguyên lý:

Một thiết bị hay một IC khi kết nối với bus I2C, ngoài một địa chỉ (duy nhất) để phân biệt, nó còn được cấu hình là thiết bị chủ hay tớ. trên một bus I2C thì quyền điều khiển thuộc về thiết bị chủ. Thiết bị chủ nắm vai trò tạo xung đồng hồ cho toàn hệ thống, khi giữa hai thiết bị chủ-tớ giao tiếp thì thiết bị chủ có nhiệm vụ tạo xung đồng hồ và quản lý địa chỉ của thiết bị tớ trong suốt quá trình giao tiếp. Thiết bị chủ giữ vai trò chủ động, còn thiết bị tớ giữ vai trò bị động trong việc giao tiếp.

c. Chế độ hoạt động (tốc độ truyền)

-Chế độ tiêu chuẩn(Standard mode):

1. Đây là chế độ tiêu chuẩn ban đầu được phát hành vào đầu những năm 80
2. Nó có tốc độ dữ liệu tối đa 100kbps
3. Nó sử dụng 7-bit địa chỉ, và 112 địa chỉ tớ

-Chế độ nhanh (Fast mode)

1.Tốc độ dữ liệu tối đa được tăng lên đến 400 kbps.

2. Để ngăn chặn gai tiếng ồn, Ngõ vào của thiết bị Fast-mode là Schmitt-triggered.

3. chân SCL và SDA của một thiết bị tớ I ² C ở trạng thái trở kháng cao khi không cấp nguồn.

-Chế độ cao tốc (High-Speed):

Chế độ này đã được tạo ra chủ yếu để tăng tốc độ dữ liệu lên đến 36 lần nhanh hơn so với chế độ tiêu chuẩn. Nó cung cấp 1,7 Mbps (với Cb = 400 pF), và 3.4Mbps (với C> b = 100pF)

2.5.2 Giao Tiếp UART

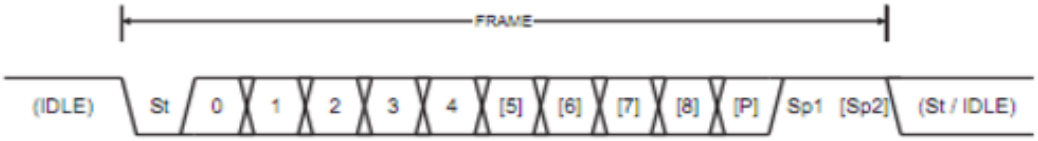
UART (Universal Asynchronous Receive/Transmit) là chuẩn giao tiếp truyền nhận dữ liệu không đồng bộ. Đây là chuẩn giao tiếp phổ biến và dễ sử dụng , thường dùng trong giao tiếp giữa vi điều khiển với nhau hoặc với các thiết bị khác

Giao tiếp UART được sử dụng nhiều trong các ứng dụng như giao tiếp máy tinh, giao tiếp module SIM, module GPS...

Cách hoạt động

Hai thiết bị giao tiếp UART với nhau thông qua hai đường dẫn RX và TX

Vì là giao tiếp không đồng bộ nên hai thiết bị phải được cài đặt thống nhất về khung truyền (Hình 2), tốc độ truyền



St	Start bit, always low.
(n)	Data bits (0 to 8).
P	Parity bit. Can be odd or even.
Sp	Stop bit, always high.
IDLE	No transfers on the communication line (RxDn or TxDn). An IDLE line must be high.

Khung truyền của giao tiếp UART.

Các chế độ hoạt động của UART trong 8051

SM0	SM1	Chế độ	Khung truyền	Tốc độ Baudrate
0	0	0	8-bit Shift Register	Oscillator / 12
0	1	1	8-bit UART	Xác định qua TIMER1
1	0	2	9-bit UART	Oscillator/ 32
1	1	3	9-bit UART	Xác định qua TIMER1

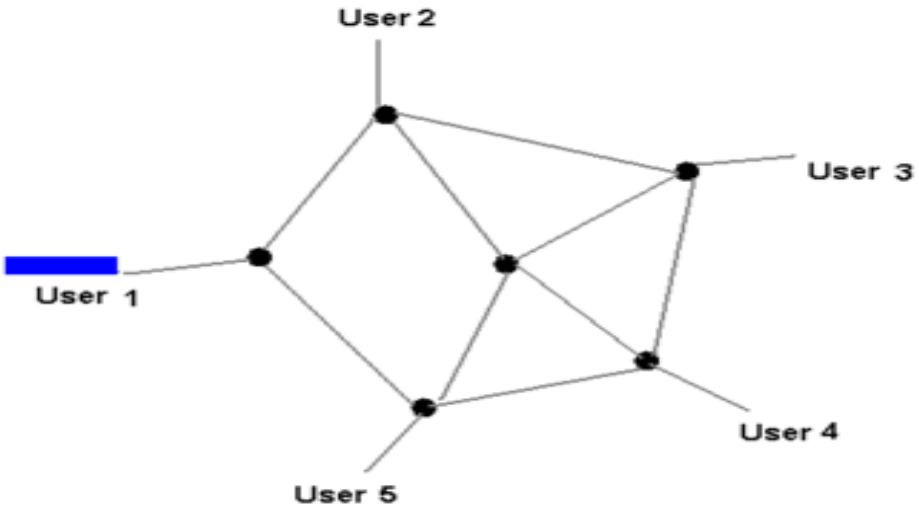
2.5.3 Giao Tiếp Bluetooh

Bluetooth về cơ bản là một giao tiếp bằng sóng radio ở băng tần 2.4 đến 2.480 GHz, rất gần với chuẩn Wifi 2.4GHz hiện nay. Tuy nhiên, khác với Wifi hay các sóng radio khác hoạt động ở 1 băng tầng cố định, Bluetooth triển khai theo khái niệm "nhảy tần trải phổ" (Frequency Hopping Spread Spectrum), có nghĩa là băng tần hoạt động của Bluetooth thay đổi liên tục với 79 kênh (từ 2.400 GHz đên 2.480 GHz).

Bước sóng của bluetooth là khoảng 12cm. Đây cũng là chiều dài thường thấy của ăn-ten mà các bạn thấy trên các module

Nguyên lý:

Bluetooth thực hiện giao tiếp với nhau theo kiểu chủ-tớ (Master-Slave), và thông thường 1 chủ có thể nối với 7 thiết bị tớ cùng 1 lúc thành một hệ thống mạng mini. Dĩ nhiên các thiết bị có thể đổi vai trò, tùy vào điều kiện tiếp nối



Giả sử ta có 1 mạng 6 thiết bị, User 1 muốn gửi thông tin đến User 5, và User 2 muốn gửi thông tin đến User 4. User 1 sẽ chia thông tin cần chuyển thành 5 gói nhỏ 54321 và User 2 chia thông tin của mình thành 4 gói 4321. Các gói thông tin này được truyền đến người nhận bằng đường truyền khác nhau, không nhất thiết là đi trực tiếp User 1 => User 5 mà có thể qua đường User 1 => User 2 => User 4 => User 5. Như vậy thông tin có thể truyền đi rất nhanh chóng. Sau khi tất cả các gói đã được nhận được ở điểm đến. chúng sẽ được sắp xếp lại theo thứ tự ban đầu.

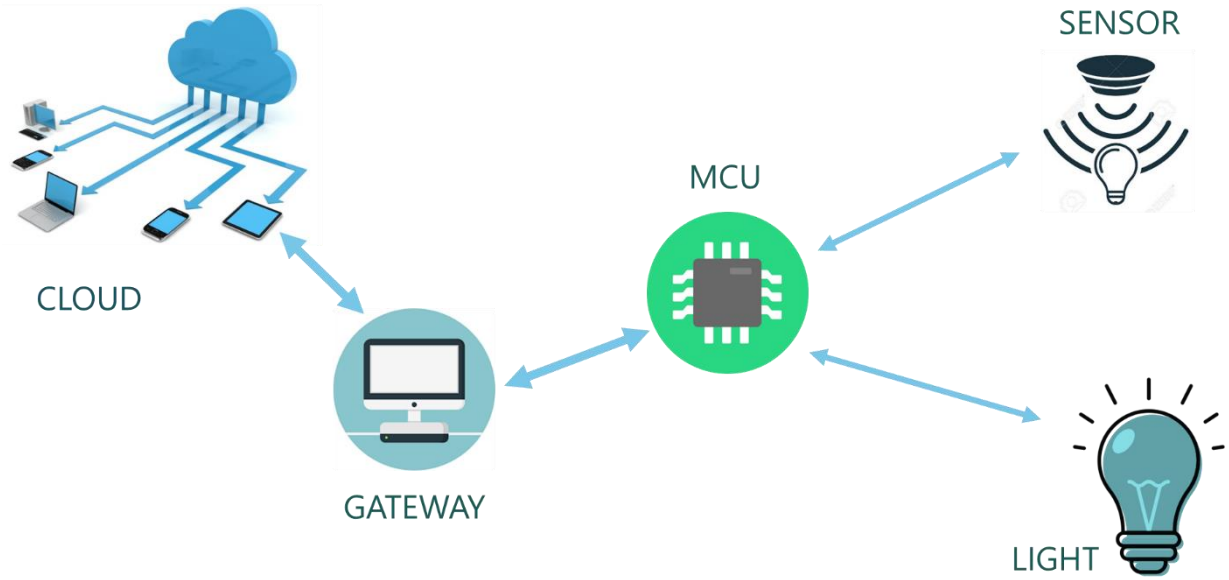
- Hiện nay có 2 phương án tiếp cận với Bluetooth. Phương án "truyền thống" Classic Bluetooth dành cho giao tiếp với dung lượng dữ liệu lớn như headphone hay truyền ảnh, và phương án "năng lượng thấp" BLE cho các ứng dụng Internet của Vạn Vật

Khoảng cách hoạt động	100 m (330 ft)	>100 m (>330 ft)
Tốc độ truyền	1–3 Mbit/s	1 Mbit/s
Thông lượng	0.7–2.1 Mbit/s	0.27 Mbit/s
Số lượng Slavez	7	Tùy vào ứng dụng
Mã hóa	56/128-bit	128-bit AES và Counter Mode CBC-MAC
Độ trễ (từ lúc chưa kết nối)	100 ms	6 ms
Thời gian tối thiểu để gửi dữ liệu	100 ms	3 ms
Truyền giọng	Có	Không
Tôpô mạng	Scatternet	Scatternet
Năng lượng tiêu thụ	1 W	0.01 đến 0.5 W
Nguồn tiêu thụ	<30 mA	<15 mA
Profile	Có	Có

Chương 3. Tổng quan hệ thống

3.1 Sơ đồ khối - Block diagram

Dựa trên những chức năng và mục đích của đề tài. Hệ thống sẽ bao gồm **cảm biến ánh sáng** để dùng để đo độ rọi của không gian chiếu sáng. **Đèn** có hỗ trợ điều chỉnh độ sáng thông qua độ rộng xung PWM. Một **Vi điều khiển** đơn giản nhằm mục đích tiền xử lí, giao tiếp với cảm biên sánh sáng để tính toán và nhận độ sáng, gửi giá trị độ rộng xung tới đèn để tăng giảm độ sáng.



Để tiện trong quá trình thiết kế xây dựng thử nghiệm đề tài. Có thể sử dụng một **màn hình LCD** để hiển thị giá trị cường độ ánh sáng. LCD này mục đích chủ yếu để hiện thị trực tiếp giá trị cường độ ánh sáng và độ rộng của băm xung (coi như độ sáng đèn).

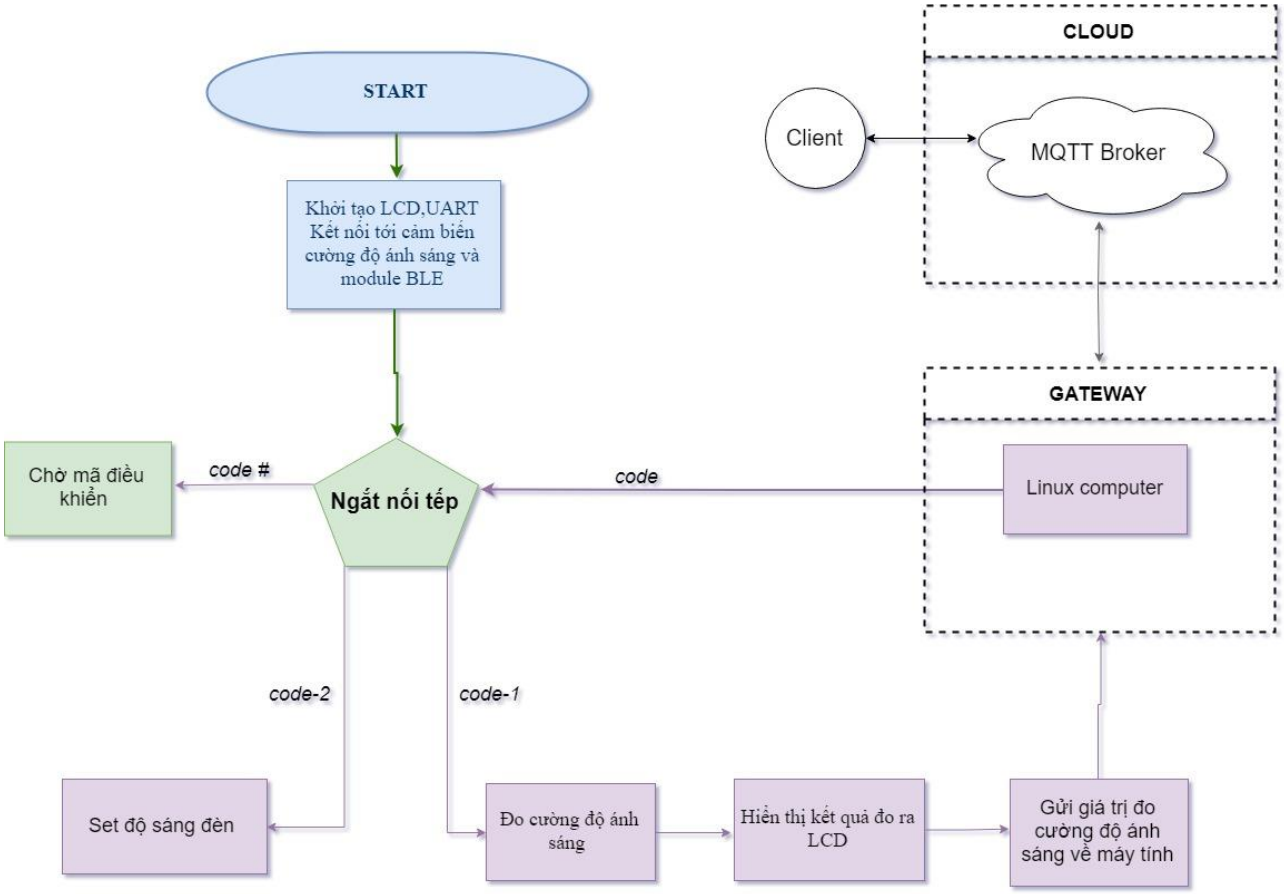
Để giao tiếp với Gateway chúng ta có thể dùng Zigbee, Wifi,Bluetooth... Trong giai đoạn thử nghiệm để tiện cho việc xây dựng và tìm kiếm phần cứng. Chúng ta sẽ dùng chuẩn Bluetooth để truyền thông. Như vậy vi điều khiển và Gateway sẽ giao tiếp với nhau thông qua truyền thông bluetooth.

Gateway sẽ đảm nhận việc giao tiếp với khối chiếu sáng và Cloud, nhận dữ liệu từ người dùng thông qua Cloud, giải mã và xử lí gửi mã điều khiển về cho khối chiếu sáng. Để chịu trách nhiệm cho những công việc trên chúng ta có thể dùng một máy tính nhúng đơn giản.

Trong thực tế trên hệ thống IoT như Smart home cần dùng nhiều thiết bị đo lường như: cảm biến nhiệt độ, độ ẩm, âm thanh,cảm biến ánh sáng,...Để xử lí dữ liệu đa số các hệ thống thường dùng những máy tính nhúng như: *Raspberry Pi 3 B+*,*Qualcomm Snapdragon*, *BeagleBone Black* .. Đa số trong đó sử dụng hệ điều hành nhúng Embeded Linux. Do đó để tương thích với các hệ thống trên trong quá trình thử nghiệm chúng ta sẽ dùng **máy tính sử**

dụng hệ điều hành Linux Distro.Về phía cloud chúng ta sẽ sử dụng **MQTT Broker**, đã được đề cập ở *phần 2.4*

3.2 Lưu đồ thuật toán - Flow Chart



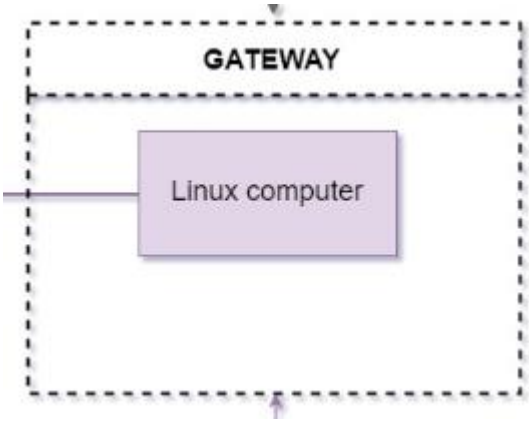
3.2.1 Khối vi điều khiển

Để giao tiếp giữa vi điều khiển và máy tính Linux chúng ta sẽ sử dụng truyền thông Bluetooth. Chương trình trên vi điều khiển sẽ được viết theo kiểu master – slave. Trong đó máy tính Linux đóng vai trò master, có nghĩa là chỉ khi có thông điệp gửi về từ máy tính thì vì điều khiển mới được phép hoạt động, thực hiện yêu cầu của những thông điệp đó. Vi điều khiển sẽ nhận thông điệp này thông qua ngắt nối tiếp. Tức là khi có thông điệp vi điều khiển sẽ gọi chương trình phục vụ ngắt- chương trình này được lập trình sẵn để giải mã và xử lí yêu cầu. Khi chưa nhận được thông điệp vi điều khiển sẽ ở trạng thái rảnh – chờ thông điệp gửi về. Để khai thác tối đa các chức năng và linh hoạt trong khâu thiết kế chương trình xử lí dữ liệu trên máy tính Linux. Các công việc của vi xử lí sẽ được đóng gói và được mã hóa bằng kí tự ASCII:

- **“a” :** Tắt bóng đèn
- **“b”:** Bật bóng đèn
- **“c”:** Tăng độ sáng của bóng đèn
- **“d”:** Giảm độ sáng của bóng đèn
- **“e”:** Đo cường độ ánh sáng và gửi về máy tính

3.2.2 Máy tính Linux

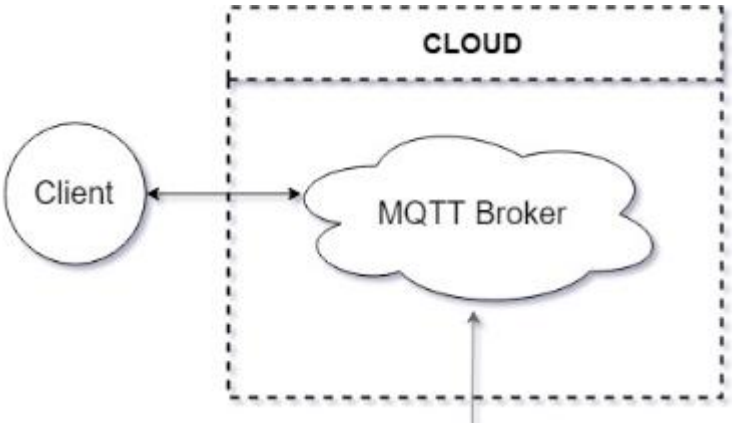
Chương trình trên máy tính Linux sẽ có nhiệm vụ nhận kịch bản mà người dùng đã chọn thông qua Cloud, giải mã và xử lí yêu cầu của những kịch bản đó. Trên máy tính linux các kịch bản được thiết kế sẵn và chờ lệnh gọi. Về cơ bản sẽ có những kịch bản sau:



- Bật bóng đèn với độ sáng cao nhất
- Tắt bóng đèn
- Đọc sách với dải độ rọi 500-550 lux
- Xem phim với dải độ rọi 250-300 lux
- Giải trí với dải độ rọi 350-400 lux
- Ngủ với dải độ rọi 100-150 lux
- Tùy chỉnh với dải độ rọi phụ thuộc vào người dùng

3.2.3 Cloud

Về phía cloud chúng ta sẽ sử dụng MQTT Broker. Nhiệm vụ của nó cơ bản là nhận thông điệp từ người dùng và xuất bản tới máy tính linux. Từ đó máy tính sẽ xác nhận thông điệp và xử lí có phản hồi về Cloud. Các kịch bản sẽ được mã hóa dưới dạng thông điệp:



- “bat” : Bật đèn
- “tat”: Tắt đèn
- “docsach”: Đọc sách
- “xemphim”: Xem phim
- “gaitri”: Giải trí
- “ngu”: Ngủ
- “tuychinh-xxxx”: Tùy chỉnh với độ rọi từ 0000-2000 lux

Chương 4. Thiết kế phần cứng

4.1 Lựa chọn phần cứng

Với mô hình thử nghiệm, việc lựa chọn phần cứng làm sao cho phù hợp với mục đích, đáp ứng được mọi yêu cầu và đặc biệt là cần độ chính xác cao. Chúng tôi đã lựa chọn những phần cứng sau:

- Cảm biến ánh sáng : BH1750
- Màn hình hiển thị : LCD16x2
- Modul Bluetooth: : HC-05
- Vi xử lý (VXL) : AT89S52

Và phần dưới đây sẽ đi rõ vào từng phần cứng riêng

4.2 Phân tích datasheet

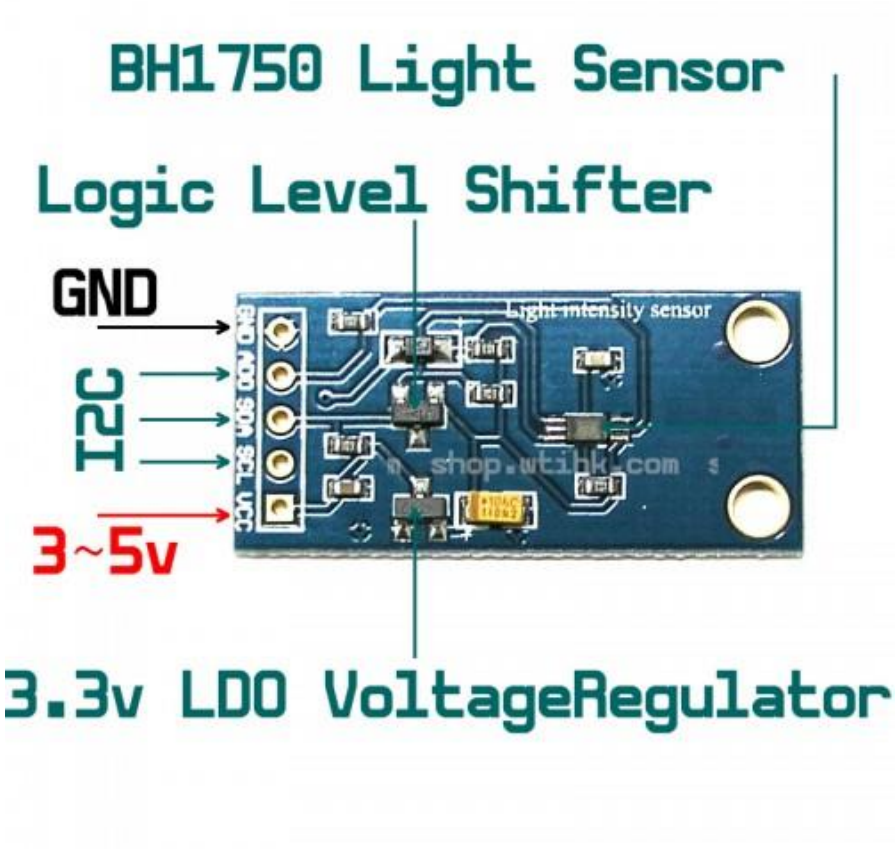
4.2.1 Cảm biến ánh sáng BH1750

- Cảm biến tích hợp ADC và bộ tiền xử lý nên giá trị đầu ra là giá trị trực tiếp cường độ ánh sáng lux.

- Sử dụng chuẩn giao tiếp chuẩn I2C nên dễ dàng giao tiếp với VXL.

- Khoảng đo rộng từ 1-65535 lux.

- Tắt nguồn khi dòng thấp .



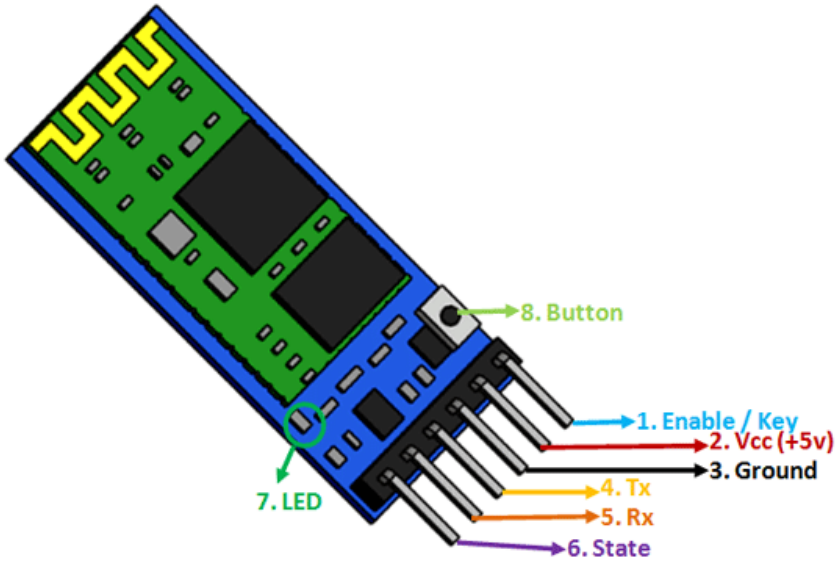
- Tần số quét 50Hz/60Hz Lọc ánh sáng nhiễu tốt.

- Độ biến thiên đo lường nhỏ (+/- 20%)

- Ảnh hưởng bởi tia hồng ngoại là rất nhỏ.

- Nguồn cung cấp 2,4-3,6 (V) phù hợp với điện áp của các thành phần khác.

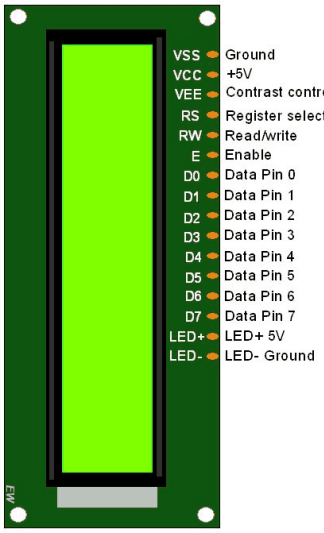
4.2.2 Modul Bluetooth



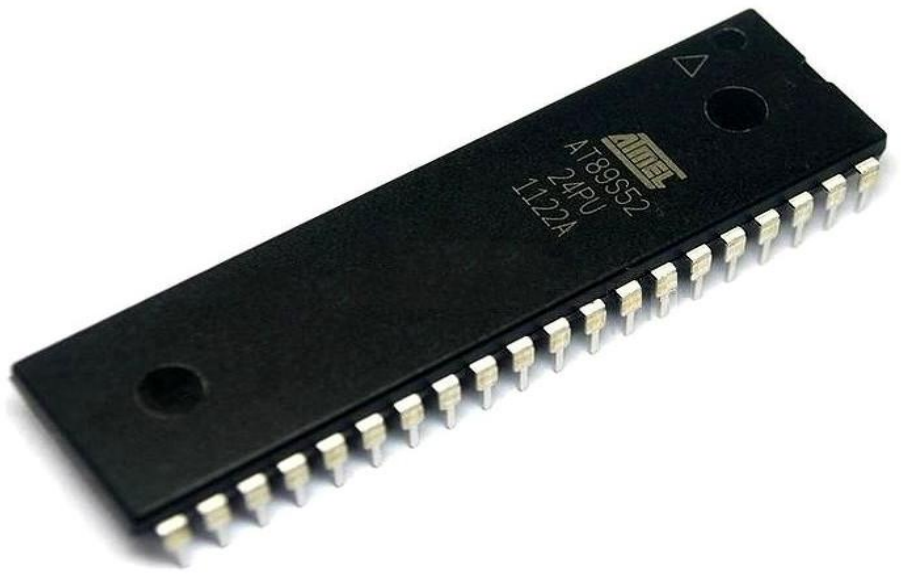
- HC05 là một module có chức năng truyền (nhận) không dây hai chiều (song công hoàn toàn).
- Module có thể sử dụng để liên lạc giữa hai bộ vi điều khiển nhưarduino hoặc giao tiếp với bất kỳ thiết bị nào có chức năng Blue-tooth như điện thoại hoặc máy tính xách tay
- Điện áp hoạt động từ 3-5,5(V) phù hợp với các chân của các thành phần khác của mạch
- Giao tiếp bằng phưng thức UART đặc biệt có thể vừa là Slave vừa là Master không như HC-06 chỉ có chế độ Slave, phù hợp với quá trình gửi và nhận thông tin
- Baudrate UART có thể chọn được: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200

4.2.3 Màn hình hiển thị LCD 16x2

- Nguồn cung cấp 3V hoặc 5V phù hợp với nguồn cung cấp cho các thành phần khác của mạch
- Có chế độ hiển thị cả số, chữ và kí tự
- Chế độ hiển thị 2 dòng phù hợp với mục đích
- Có thể giao tiếp 4bit giúp giảm nối chân, mạch đơn giản hơn



4.3.4 Vi xử lý AT89S52



- Là VXL 8bit họ MCS-51 khá quen thuộc với chúng em
- Nguồn cung cấp là 4V-5,5V cùng nguồn với các thành phần khác
- Có 8k Bộ nhớ flash trong, có 32 I/O pin đủ để kết nối với các thành phần của mạch
- Có kênh kết nối UART nên có thể kết nối với HC05
- Tuy không hỗ trợ chuẩn I2C nhưng có thể lập trình, việc đó giúp chúng em hiểu rõ cách hoạt động của giao tiếp
- Có thể nạp lại bộ nhớ chương trình khi đang được cắm vào socket, không bất tiện như 89Cxx

4.3 Chế độ sử dụng

4.3.1 Cảm biến ánh sáng BH1750

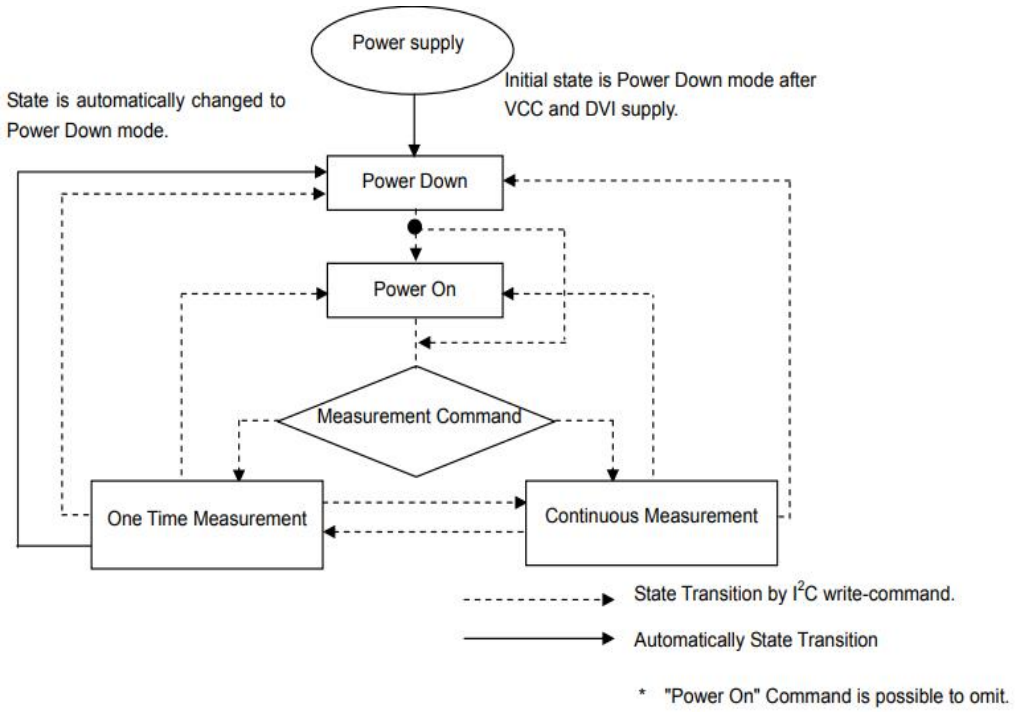
- Có 2 chế độ địa chỉ :

Chân số	Tên	Chức năng
1	VCC	Chân cung cấp nguồn 2,4-3,6 V DC
2	SCL	Chân clock
3	SDA	Chân data

4	ADD	Chân địa chỉ
5	GND	Chân đất

ADDR = ‘H’ (ADDR ≧ 0.7VCC) → “1011100“

ADDR = 'L' (ADDR ≦ 0.3VCC) → “0100011“



Sau khi cấp nguồn thì trạng thái của cảm biến là Power Down

Muốn nạp lệnh cho Cảm biến ta phải đưa nó về trạng thái Power on (có thể bỏ qua bước này)

Sau đó mình sẽ nạp lệnh đo cho cảm biến, có 2 chế độ đo là:

- Đo 1 lần
- Đo liên tục

- Với chế độ đo 1 lần

Có 3 chế độ đo:

- One times H-Resolution Mode
- One times H-Resolution Mode2
- One times H-Resolution

Ta có bảng so sánh sau:

	H-Resolution Mode	H-Resolution Mode2	H-Resolution Mode
Lệnh	00100000	00100010	00100011

Độ phân giải ban đầu(lx)	1	0.5	4
Thời gian đo	120	120	16

Sau khi quá trình đo được thực hiện xong thì trạng thái của nó tự động quay lại chế độ Power Down

- Chế độ đo liên tục

Có 3 chế độ đo:

– Continously H-Resolution Mode

– Continously H-Resolution Mode2

– Continously H-Resolution

Ta có bảng so sánh

	H-Resolution Mode	H-Resolution Mode2	H-Resolution Mode
Lệnh	00010000	00010001	00010011
Độ phân giải ban đầu(lx)	1	0.5	4
Thời gian đo	120	120	16

Kết quả đo: Gỉa sử kết quả đo được 8 bit cao là 10010000 và 8 bit thấp là 00010001 thì giá trị của cường độ sáng là:

(2^15+2^12+2^4+2^0)/1.2= 30734 lux

4.3.2 Modul Bluetooth HC05

Số thứ tự	Tên	Chức năng
1	State	Chân test
2	RX	Đây là hai chân UART để giao tiếp module hoạt động ở mức logic 3.3V
3	TX	
4	GND	Nối với chân nguồn GND
5	VCC	chân này có thể cấp nguồn từ 3.6V đến 6V
6	EN	Chân này để chọn chế độ hoạt động AT Mode hoặc

		Data Mode

Hoạt động ở 2 chế độ command và thông thường

- Chế độ command:

Đầu tiên cấp nguồn và đồng thời đưa chân EN lên mức cao

  - Ngắt nguồn và ấn giữ nút trên modul
  - Cấp lại nguồn và thả nút bấm
  - Chế độ thông thường
  - Cấp nguồn và chân EN nối đất, hoạt động ở chế độ mặc định

Có thể là slave hoặc master:

  - chế độ slave: cần thiết lập kết nối giữa smartphone, laptop,.. với modul với mật khẩu được cấu hình cho mudul
  - chế độ master: Modul sẽ sự tìm kiếm và kết nối với các Bluetooth khác.

4.3.3 LCD 16x2

Nguồn cung cấp là 4,7 – 5,5 V DC

Hoạt động ở chế độ 4 bit hoặc 8 bit

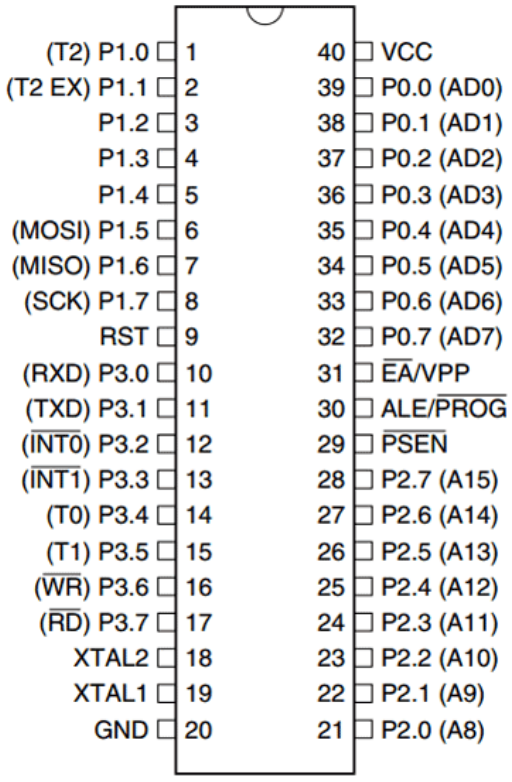
Chân số	Tên	Chức năng
1	VSS	Chân nối đất cho LCD, khi thiết kế mạch ta nối chân này với GND của mạch điều khiển
2	VDD	Chân cấp nguồn cho LCD, khi thiết kế mạch ta nối chân này với VCC=5V của mạch điều khiển
3	VEE	Chân này dùng để điều chỉnh độ tương phản của LC
4	RS	Chân chọn thanh ghi (Register select). Nối chân RS với logic “0” (GND) hoặc logic “1” (VCC) để chọn thanh ghi. + Logic “0”: Bus DB0-DB7 sẽ nối với thanh ghi lệnh IR của LCD (ở chế độ “ghi” - write) hoặc nối với bộ đếm địa chỉ của LCD (ở chế độ “đọc” - read) + Logic “1”: Bus DB0-DB7 sẽ nối với thanh ghi dữ liệu DR bên trong LCD
5	RW	Chân chọn chế độ đọc/ghi (Read/Write). Nối chân R/W với logic “0” để LCD hoạt động ở chế độ ghi, hoặc nối với logic “1” để

		LCD ở chế độ đọc
6	E	Chân cho phép (Enable). Sau khi các tín hiệu được đặt lên bus DB0-DB7, các lệnh chỉ được chấp nhận khi có 1 xung cho phép của chân E. + Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào(chấp nhận) thanh ghi bên trong nó khi phát hiện một xung (high-to-low transition) của tín hiệu chân E. + Ở chế độ đọc: Dữ liệu sẽ được LCD xuất ra DB0-DB7 khi phát hiện cạnh lên (low to-high transition) ở chân E và được LCD giữ ở bus đến khi nào chân E xuống mức thấp
7-14	D0-7	Tám đường của bus dữ liệu dùng để trao đổi thông tin với MPU. Có 2 chế độ sử dụng 8 đường bus này: + Chế độ 8 bit: Dữ liệu được truyền trên cả 8 đường, với bit MSB là bit DB7. + Chế độ 4 bit: Dữ liệu được truyền trên 4 đường từ DB4 tới DB7, bit MSB là DB7

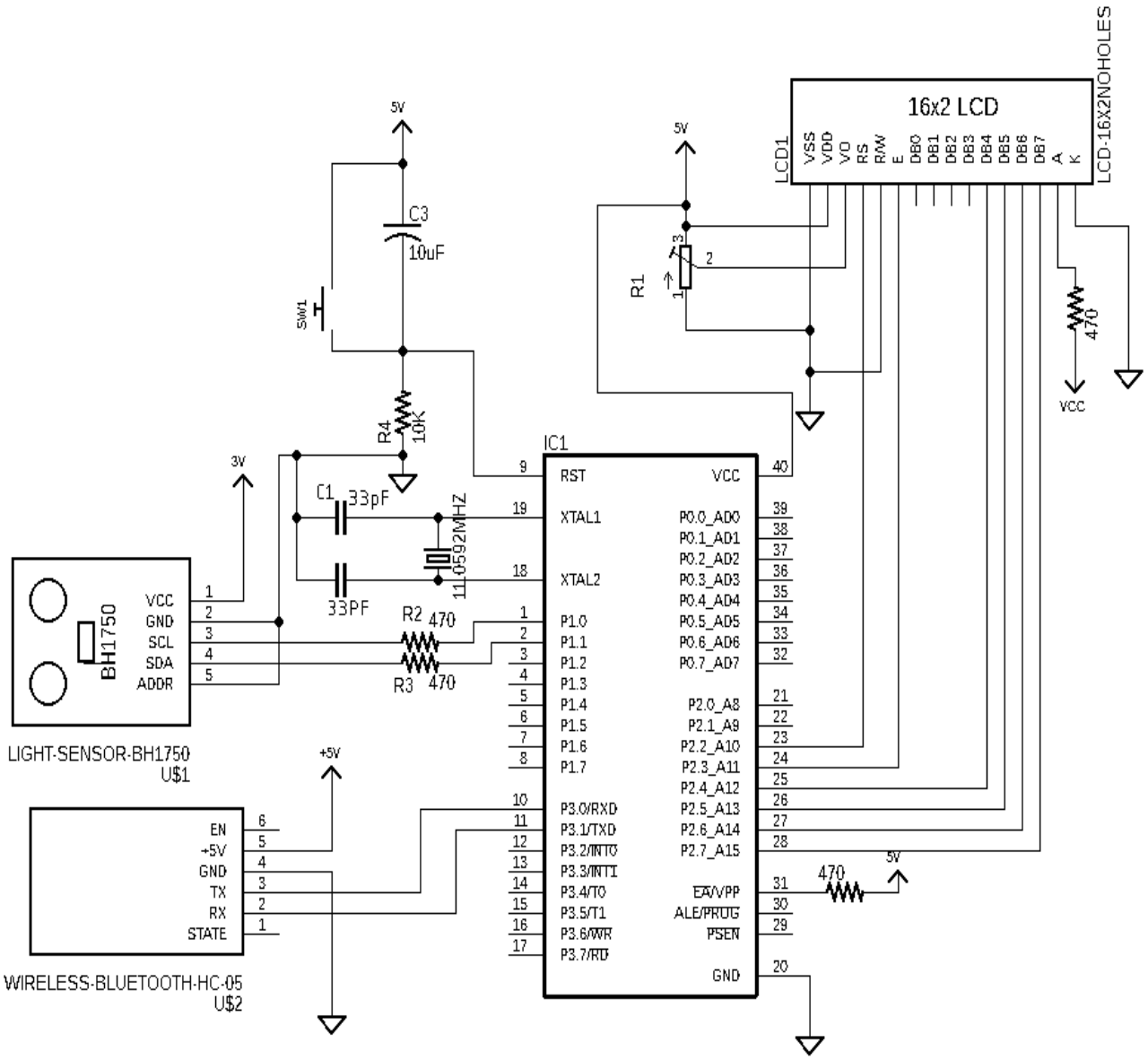
4.3.4 Vi xử lý AT89S52

- Vi xử lý được nạp chương trình vào để điều khiển các khối xung quanh

- Chi tiết xem rõ phần Thiết kế phần mềm.

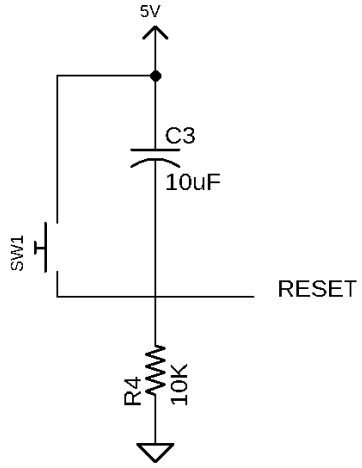


4.4 Sơ đồ mạch nguyên lý

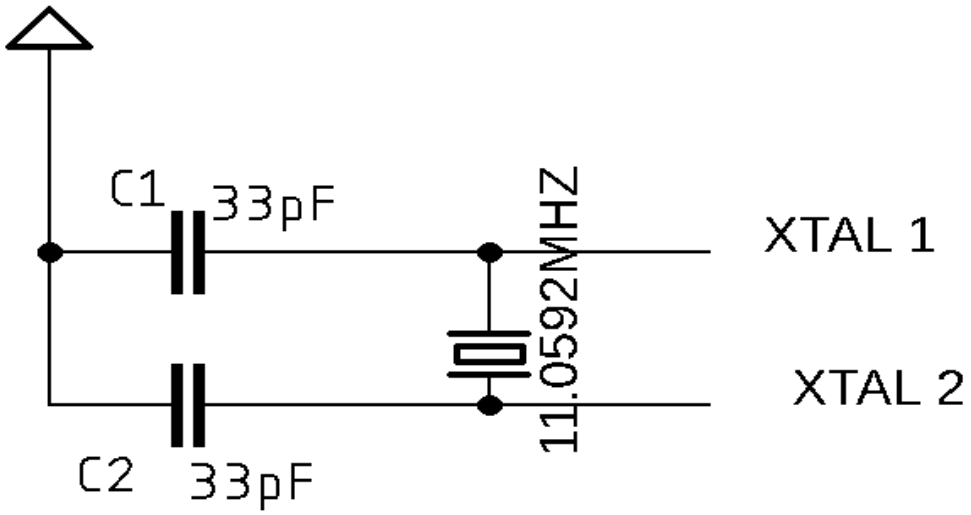


4.4.1 Mạch reset:

- Khi bấm nút VXL được reset lại
- Mạch được nối vào chân số 9 của VXL



4.4.2 Mạch tạo dao động:



- Sử dung thạch anh với tần số 11.0592 Mhz

- Tụ hóa 33pF

- Nguồn +5V

- Mạch được nối vào chân 10,11 của VXL

4.4.3 Kết nối Module cảm biến ánh sáng BH1750

- Chân SCL, SDA lần lượt nối với chân số 1, 2 của VXL thông qua điện trở R2, R3 có giá trị 470 (Ω)

- Chân VCC nối với nguồn cấp 3,3 (V)

- Chân VDD nối đất

4.4.4 Kết nối Module Bluetooth HC05

- Chân TX, RX lần lượt nối với chân 10, 11 của VXL

- Chân VCC nối với nguồn cung cấp 5V

- Chân VDD nối đất

4.4.5 Kết nối LCD

- Chân dữ liệu từ D4 - D7 lần lượt được nối với chân từ 25 – 28 của VXL

- Chân VSS, RW được nối đất

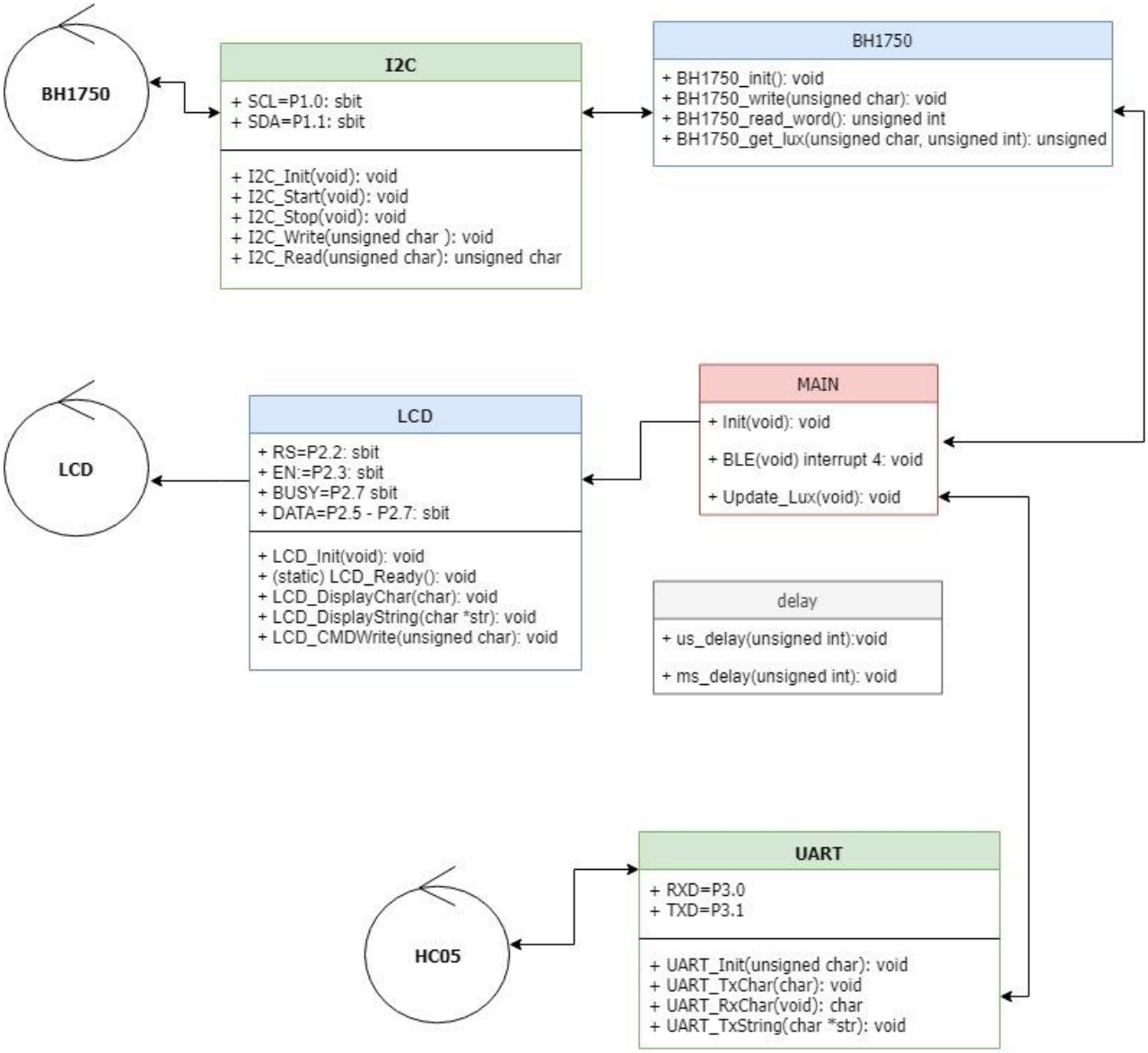
- Chân VDD nối với nguồn 5V

- Chân VO nối với biến trở

- Chân RS nối với chân 23 của VXL

Chương 5. Thiết kế phần mềm

5.1 Thiết kế chương trình cho vi điều khiển



Với chương trình. Khi có mã ASCII từ máy tính gửi về thông qua module bluetooth HC05, chương trình phục vụ ngắt nối tiếp ***BLE(void) interrupt 4*** sẽ được gọi để giải mã ASCII đó và thực hiện các chức năng đã được định danh trước.

Ví dụ:

- Khi mã “e” được nhận , mã này sẽ được định danh cho chức năng cập nhật độ rọi,gửi giá trị cho máy tính, hiển thị ra LCD.Với mã này vi điều khiển sẽ thực hiện hàm ***Update_Lux()***

Tương tự với các mã khác vi điều khiển sẽ gọi các hàm tương ứng:

- Khi nhận được mã ”a”: Vi điều khiển sẽ gửi giá trị PWM = 0 thông qua I2C cho khối LED, đồng nghĩa với việc đèn sẽ bị tắt.

- Khi nhận được mã “b”: Tương tự với mã “a” nhưng với độ max độ rộng xung PWM=255, tức là đèn sáng với cường độ lớn nhất

- Khi nhận được mã “c”: Với mã này vi điều khiển sẽ thực hiện tăng giá trị độ rộng xung PWM lên một đơn vị . Có nghĩa là độ sáng của đèn LED sẽ được tăng lên

- Khi nhân chữ “d”: Tương tự với mã “c” mã này sẽ giảm độ sáng của đèn LED .

5.2 Xây dựng chương trình trên máy tính

Chương trình trên máy tính bao gồm hai chức năng cơ bản:

1. Kết nối với vi điều khiển thông qua module bluetooth HC05
2. Nhận thông điệp tương ứng với những kịch bản từ Cloud
3. Giải mã và xử lí những yêu cầu của kịch bàn

Trong quá trình thử nghiệm đề tài có thể bỏ qua thiết kế giao diện cho chương trình. Chương trình sẽ chạy trên Terminal của máy tính chạy hệ điều hành linux.

Để kết nối với HC05 cần phải lập trình Bluetooth. Trước tiên phải cài đặt thư viện phát triển Bluetooth BlueZ (BlueZ development libraries). Sau đó cần phải chọn một giao thức truyền (transport protocol) Bluetooth. Có 3 giao thức phổ biến hay được sử dụng như: RFCOMM, L2CAP with infinite retransmit , L2CAP (0-1279 ms retransmit)…

Requirement	Internet	Bluetooth
Reliable, streams-based	TCP	RFCOMM
Reliable, datagram	TCP	RFCOMM or L2CAP with infinite retransmit
Best-effort, datagram	UDP	L2CAP (0-1279 ms retransmit)

Giao thức RFCOMM cung cấp dịch vụ đảm bạo và tin cậy gần giống như TCP. Mặc dù thông số kỹ thuật nói rõ rằng nó được thiết kế để mô phỏng các cổng nối tiếp RS-232 (để giúp các nhà sản xuất dễ dàng thêm Bluetooth vào các thiết bị có cổng nối tiếp của họ), nhưng khá đơn giản để sử dụng nó trong nhiều tình huống giống như TCP . Vì tính đơn giản và độ tin cậy cao, sẽ dùng giao thức RFCOMM để truyền dữ liệu.

Để tìm ra cách giao tiếp với một máy từ xa, một khi đã biết địa chỉ số và giao thức truyền tải, là chọn số cổng. Hầu như tất cả các giao thức truyền tải Internet được sử dụng phổ biến đều được thiết kế với khái niệm số cổng, do đó nhiều ứng dụng trên cùng một máy chủ có thể đồng thời sử dụng giao thức truyền tải. Bluetooth cũng không ngoại lệ, nhưng sử dụng thuật ngữ hơi khác nhau.

protocol	terminology	reserved/well-known ports	dynamically assigned ports
TCP	port	1-1024	1025-65535
UDP	port	1-1024	1025-65535
RFCOMM	channel	none	1-30
L2CAP	PSM	odd numbered 1-4095	odd numbered 4097 - 32765

Trong L2CAP, các cổng được gọi là bộ ghép kênh dịch vụ giao thức và có thể nhận các giá trị số lẻ trong khoảng từ 1 đến 32767. Trong RFCOMM, các kênh 1-30 có sẵn để sử dụng. Bộ ghép kênh dịch vụ giao thức (Protocol Service Multiplexer) và kênh đều phục vụ cùng một mục đích chính xác như các cổng thực hiện trong TCP / IP. L2CAP, không giống như RFCOMM, có một loạt các số cổng dành riêng (1-1023) không được sử dụng cho các ứng dụng và giao thức tùy chỉnh.

Lập trình chương trình Bluetooth sử dụng Python – PyBlueZ. Python là một ngôn ngữ hướng đối tượng linh hoạt và mạnh mẽ, cung cấp cú pháp rõ ràng cùng với tích hợp quản lý bộ nhớ để lập trình viên có thể tập trung vào thuật toán trong tay mà không phải lo lắng về rò rỉ bộ nhớ hay các dấu ngoặc . Mặc dù Python có một thư viện tiêu chuẩn lớn và toàn diện, hỗ trợ Bluetooth chưa phải là một phần của phân phối chuẩn.. PyBluez là một mô-đun mở rộng Python được viết bằng C, cung cấp quyền truy cập vào tài nguyên Bluetooth của hệ thống theo cách thức mô-đun hướng đối tượng. Nó được viết cho Windows XP (Microsoft Bluetooth stack) và GNU / Linux (BlueZ stack).

Dươi đây là đoạn mã sử dụng để truyền thông Bluetooth giữa Linux và HC05 sử dụng module BlueZ.

```
import paho.mqtt.client as mqtt
import bluetooth
import os
import time
import signal

#BLE Comunication
bd_addr ="00:21:13:05:E1:45"
port = 1
sock=bluetooth.BluetoothSocket(bluetooth.RFCOMM )
sock.connect((bd_addr,port))
```

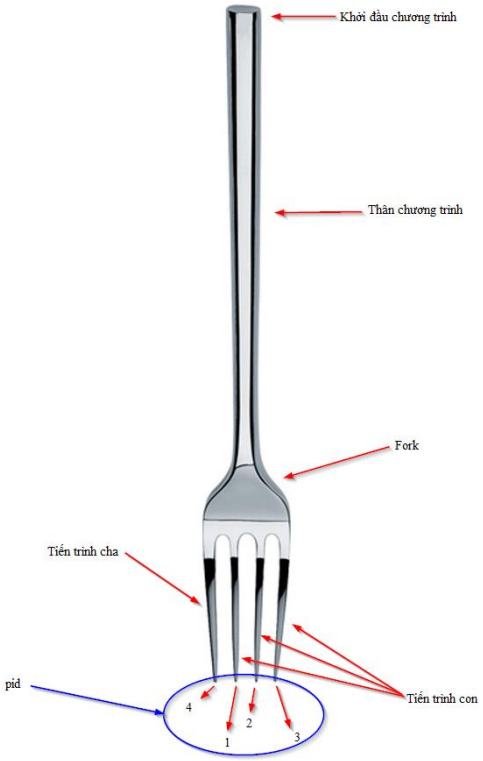
Cụ thể đoạn mã này sẽ liên kết cổng Bluetooth với cổng nối tiếp trên Linux. Địa chỉ MAC của HC05: 00:21:13:05:E1:45 sẽ được nhận dạng và kết nối. Và sẽ được kéo về cổng nối tiếp RFCOMM. Thông qua socket API sử dụng các hàm để gửi và nhận dữ liệu.

Sau khi đã kết nối được với module HC05. Sau đó chúng ta sẽ viết các hàm xử lí dữ liệu:

```
def Change_PWM(inc_or_dec,dest): ...
def Setting_Mode(min_lux,mid_lux,max_lux): ...
def LED_Handler(mode): ...
```

- **Change_PWM**: Hàm này sẽ có hai tham biến *inc_or_dec* và *dest.* Tham biến *inc_or_dec* dùng để nhận dạng xử lí tăng, giảm độ sáng đèn, *dest* là đích độ rọi cần điều chỉnh
- **Setting_Mode**: Hàm này sẽ nhận dải độ rọi *(min_lux, mid_lux, max_lax)* từ đó xử lí điều chỉnh độ rọi phù hợp thông qua hàm *Change_PWM*
- **LED_Handler**: Hàm sẽ nhận chế độ từ Cloud giải mã và đưa ra các dải độ sáng theo kịch bản sau đó gọi các hàm *Change_PWM, Setting_Mode* để xử lí

Trong quá trình thiết kế phần mềm cho máy tính linux có sử dụng kĩ thuật fork tức là tạo ra các child process hoạt động độc lập với chương trình chính. Để duy trì được dải độ sáng theo thời gian thực chúng ta sẽ phải tạo cho nó một vòng lặp để liên tục kiểm tra độ rọi và tự động điều chỉnh. Do vòng lặp vô hạn nên chúng ta sẽ tách nó ra khỏi chương trình chính và xử lí độc lập.



Khi có thông điệp từ cloud để chuyển sang kịch bản khác chương trình chính sẽ kill process của kịch bản hiện tại và khởi tạo process mới phục phụ kịch bản mới đó.

Đoạn mã khởi tạo child process và kill process

```
#Child process
with open("log.txt",'w') as log:
def child_process(min_lux,mid_lux,max_lux):
    try:
        pid = os.fork()
    except OSError:
        exit("Could not create a child process")
    if pid == 0:
        with open("log.txt",'w') as log:
                log.write(str(os.getpid()))
                log.close()
        while True:
            Setting_Mode(min_lux,mid_lux,max_lux)
def kill_process():
    with open("log.txt",'r') as data:
        id_process=int(data.read())
        data.close()
    print(id_process)
    if int(id_process)>int('0'):
        print(id_process)
        os.kill(id_process,signal.SIGTERM)
```

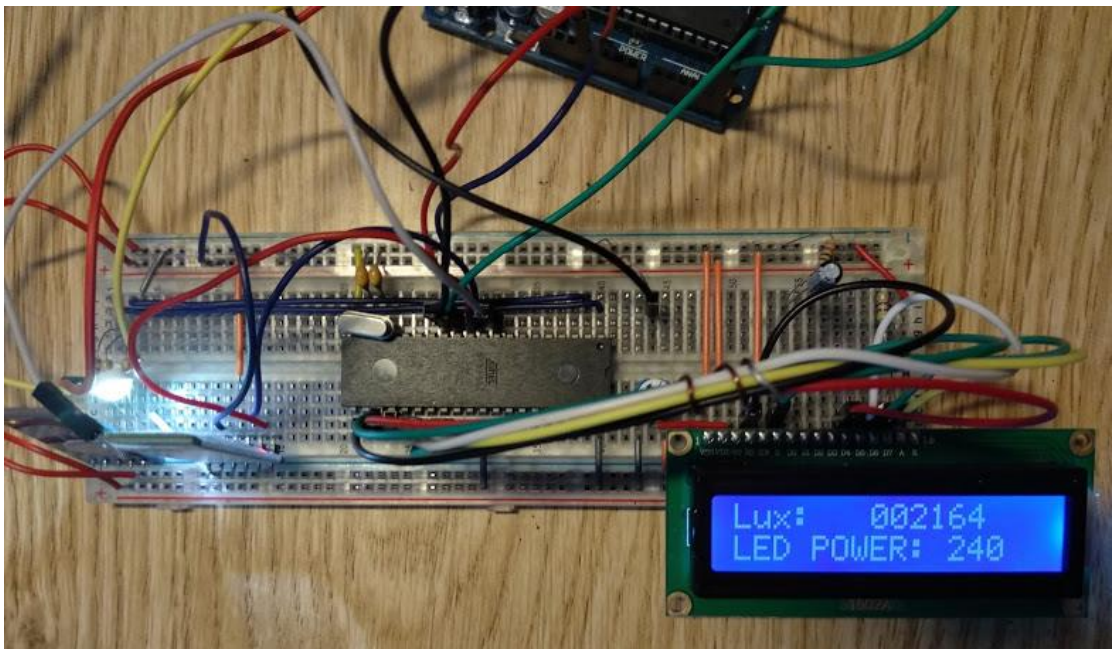
Về phần cloud chương trình sẽ sử dụng thư viện Paho-một thư viện hỗ trợ giao tiếp với cloud MQTT Broker. Chương trình sẽ liên tục chờ thông điệp được xuất bản, một khi nhận được thông điệp chương trình sẽ giải mã thông điệp đó. Dưới đây là đoạn mã kết nối với Cloud MQTT Broker. Subcribe/Publich

```
# Callbacks
def on_connect(client,userdata,flags,rc):
    print("Connected with Code:" + str(rc))
    # Subscribe Topic
    client.subscribe("tranglc/#",qos=0)

def on_message(client,userdata,msg):
    message=str(msg.payload)
    print(message)
    LED_Handler(message)
# This is the Publisher

client = mqtt.Client()
client.on_connect=on_connect
client.on_message=on_message

client.connect("m16.cloudmqtt.com",12615,60)
client.username_pw_set("qnqbhzyl","agzedLO5Jdqo")

client.loop_forever()
```

Chương 5. Triển khai thử nghiệm đề tài

5.1 Mô hình thử nghiệm

Từ sơ đồ mạch nguyên lý chúng ta dễ dàng xây dựng mạch test. Hình dưới là mạch test đề tài sau khi đã hoàn thành công đoạn thiết kế phần cứng và phần mềm.

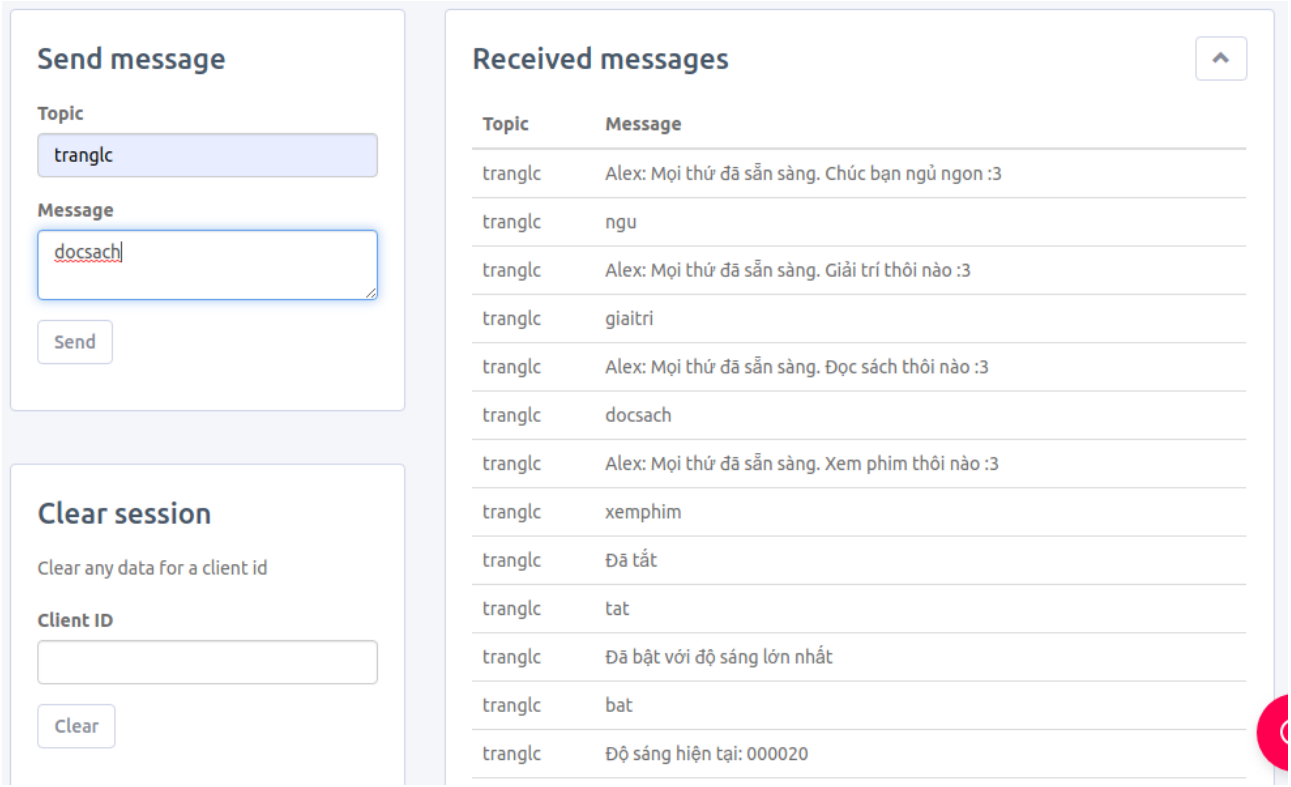


LCD sẽ hiển thị giá trị độ rọi sáng mỗi lần thực hiện mã “e” cập nhật giá trị lux. Ngoài ra giá trị độ sáng của LED cũng được hiện thị ra LCD ám chỉ độ sáng hiện tại của đèn LED.

Về cơ bản hệ thống phần cứng hoạt động khá tốt, ứng với mã được gửi về từ máy tính các chức năng đều được thực hiện và thể hiện rõ sự thay đôi qua màn hình LCD.

Giao diện Cloud MQTT Broker. Mặc định khối gateway sẽ subcribe một topic là : tranglc.

Trên giao diện GUI MQTT Broker chúng ta sẽ publish kịch bản thông qua topic đó. Chỉ cần có kết nối internet là chúng ta có thể cài đặt được các kịch bản, điều khiển hệ thống chiếu sáng.



5.2 Đánh giá đề tài

Hệ thống được thiết kế theo hướng mở, tức là mỗi khối trong đó sẽ có những chức năng riêng và hoạt động độc lập. Các khối sẽ giao tiếp với nhau thông qua các mã được xây dựng sẵn. Tất cả các chức năng có thể có của khối chiếu sáng (mỗi chức năng sẽ chỉ thực hiện một công việc) đều sẽ được lập trình và định danh bởi một mã ASCII nhất định. Từ những chức năng nhỏ và cụ thể đó chúng ta có thể dễ dàng thiết kế ra những thuật toán linh hoạt để phục vụ cho mục đích sử dụng. Do tính chất này chúng ta có thể lắp đặt khối chiếu sáng cố định. Khi chúng ta cần chỉnh sửa các kịch bản chiếu sáng, thay đổi một số chức năng của hệ thống chỉ cần truy cập vào máy tính nhúng để lập trình.

Hệ thống sử dụng máy tính nhúng chạy hệ điều hành Embedded Linux để điều khiển nên tương thích với các dự án smartHome. Với cấu hình của máy tính nhúng, không chỉ dừng lại ở đề tài này, chúng ta có thể phát triển thêm các hệ thống khác. Tạo thành một khối hoàn chỉnh tạo tiền đề phát triển các dự án smartHome, Internet of Things…

Lời kết

Một lần nữa chúng tôi xin cảm ơn tất cả những người đã đồng hành cùng trong quá trình phát triển và xây dựng đề tài.

Tài liệu tham khảo

1. http://jsjyl.chd.edu.cn/The_8051_Microcontroller_and_Embedded_Systems_Using_Assembly_and_C.pdf
2. *http://www.electronicaestudio.com/docs/istd016A.pdf*
3. *https://www.mouser.com/ds/2/348/bh1750fvi-e-186247.pdf*
4. *https://people.csail.mit.edu/albert/bluez-intro/*
5. *http://www.ti.com/lit/an/slva704/slva704.pdf*
6. *http://hccc.ee.ccu.edu.tw/courses/bt/vg/rfcomm.pdf*
7. *https://doc.lagout.org/programmation/python/Head%20First%20Python%2C%20First%20Edition%20%282010%29.pdf*
8. *https://www.sparkfun.com/datasheets/LCD/ADM1602K-NSW-FBS-3.3v.pdf*