

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA ĐIỆN – ĐIỆN TỬ

BỘ MÔN VIỄN THÔNG

-----o0o-----



ĐỒ ÁN MÔN HỌC

# THIẾT KẾ VƯỜN THÔNG MINH DỰA TRÊN OPENHAB

GVHD: TS. Lê Đăng Quang

SVTH: Lê Hồ Bảo Anh

1410070

Nguyễn Trần Quốc Thái

1413528

TP. HỒ CHÍ MINH, THÁNG 6 NĂM 2017

## ***LỜI MỞ ĐẦU***

Ngày nay trên thế giới với sự phát triển mạnh mẽ của khoa học – công nghệ đã góp phần giúp chất lượng cuộc sống con người không ngừng được nâng cao. Thêm vào đó sự xâm nhập trực tiếp của internet vào cuộc sống của chúng ta ngày càng nhiều. Nhu cầu của con người ngày càng cao đưa đến những thách thức liên quan đến việc quan sát, nhận biết, xử lý tự động các dữ liệu online một cách hiệu quả. Từ những thực tiễn đó, kích thích sự tò mò, tìm hiểu của con người, là nguồn cảm hứng cho các ứng dụng thông minh ra đời. Là một sinh viên khoa điện trường đại học Bách Khoa Hồ Chí Minh, với những kiến thức đã học cùng với mong muốn được tìm hiểu và được thử sức thiết kế một đề tài hay và thú vị trong việc thực thi các nhu cầu của con người thông qua mạng Internet. Đề tài “Vườn thông minh” mà em chọn cho đồ án môn học là sẽ giúp ích cho con người về việc quan sát, kiểm tra từ đó đưa ra cảnh báo hoặc thực thi một nhu cầu nào đó của chính người sử dụng cụ thể áp dụng trong nông nghiệp có tính qui mô lớn.

Trong quá trình thực hiện bài báo cáo của mình, em đã cố gắng hết sức để hoàn thành một cách tốt nhất. Nhưng với kiến thức và sự non nớt khi mới thực hiện nên khó tránh khỏi những sai sót mong thầy cô đóng góp đề đề tài của em được hoàn thiện hơn.

Em xin gửi lời cảm ơn đến thầy Lê Đăng Quang đã nhiệt tình quan sát, giúp đỡ và hướng dẫn em qua từng yêu cầu của đề tài.

Em xin chân thành cảm ơn!

*Tp. Hồ Chí Minh, ngày 20 tháng 6 năm 2017*

**Sinh viên**

**Lê Hồ Bảo Anh    Nguyễn Trần Quốc Thái**

## TÓM TẮT ĐỒ ÁN

*Đồ án này trình bày về cách thức thực hiện một ứng dụng quản lý "vườn thông minh" sử dụng OpenHAB. Trong đồ án này chúng em sẽ đi chi tiết về phần cài đặt, thiết kế và thực hiện phần cứng, thiết kế và thực hiện phần mềm và cách thiết lập cũng như sử dụng đề tài. Cụ thể là trong phần cài đặt sẽ nêu ra các phần mềm cần download và cài đặt, trong phần cứng sẽ được nêu ra trong phần lý thuyết và cách kết nối giữa chúng, thiết kế và thực hiện phần mềm sẽ gồm sơ đồ khối và code (được liệt kê đầy đủ trong phần phụ lục). Đề tài sẽ cho phép người sử dụng theo dõi được các yếu tố thời tiết và hệ thống sẽ tự điều chỉnh hoặc người dùng có thể tự điều chỉnh theo ý muốn, kèm theo là khả năng tưới nước theo 3 cách: bằng tay, hẹn giờ hoặc là đặt lịch tưới cụ thể.*

## MỤC LỤC

1. GIỚI THIỆU .....	1
1.1 Tổng quan.....	1
1.2 Nhiệm vụ đề tài .....	3
1.3 Phân chia công việc trong nhóm .....	5
2. LÝ THUYẾT .....	7
3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG .....	19
4. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM .....	22
5. KẾT QUẢ THỰC HIỆN.....	25
6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	26
6.1 Kết luận .....	26
6.2 Hướng phát triển .....	26
7. TÀI LIỆU THAM KHẢO.....	26
8. PHỤ LỤC.....	27

## DANH SÁCH HÌNH MINH HỌA

Hình 5-1 Kết thi công.....**Error! Bookmark not defined.**

Hình 5-2 Kết quả mô phỏng.....**Error! Bookmark not defined.**

**No table of figures entries found.**

## DANH SÁCH BẢNG SỐ LIỆU

Bảng 1 Thông số hệ thống .....**Error! Bookmark not defined.**

## 1. GIỚI THIỆU

### 1.1 Tổng quan

Trong những năm qua, việc trồng thực phẩm tại các khu vực thành thị đã trở thành một xu hướng toàn cầu, và nó tiếp tục phát triển. Nó đã trở nên hiện đại, tạo thành xu thế, nó đang được khẳng định là tương lai của thực phẩm, và những thương hiệu làm vườn thông minh mới xuất hiện nhanh hơn bao giờ hết. Những lợi ích mà vườn thông minh mang lại là không thể phủ nhận:

#### • *Hoạt động hiệu quả hơn*

- Với sự giúp đỡ của nông nghiệp thông minh, người ta có thể sản xuất thậm chí nhiều gấp 100 lần thực phẩm hơn so với canh tác truyền thống (mỗi mét vuông). Làm thế nào là nó có thể? Hầu hết các trang trại đô thị được thiết kế theo chiều dọc, cho phép để phát triển sản phẩm ở nhiều cấp độ.

- Thời tiết có tác động khá nhiều và làm giảm năng suất trong một năm nhất định. Nhưng trong hệ thống nông nghiệp thông minh, thời tiết là yếu tố thực sự rất nhỏ đối với sản xuất lương thực vì nó thường diễn ra trong nhà và dựa vào hệ thống nước được lắp đặt bao gồm ánh sáng nhân tạo và các chất dinh dưỡng đã được có sẵn trong đất thông qua các hệ thống đo dưỡng chất, độ PH, độ ẩm...



Hình 1.1: Tăng năng suất bằng cách trồng trên các nông trại thẳng đứng

#### • *Tình hình tài chính bền vững hơn*

- Hầu hết các hệ thống làm vườn thông minh đều tiết kiệm nước, năng lượng và không gian đáng kể. Nếu chúng ta nói về các hệ thống canh tác đô thị Click & Grow, họ sử dụng khoảng 90% nước ít hơn và 4 lần không gian hơn, nếu so sánh với canh tác truyền thống. Nhiều

người chỉ ra rằng đầu tư một trang trại ở đô thị có thể tốn kém. Trên thực tế, một trang trại ở đô thị với \$ 500 tiền đầu tư sẽ "hoàn vốn" trong một năm.

- Bên cạnh đó, tiền đầu tư không lãng phí, nó mang lại nhiều hiệu quả về mặt tài chính, giúp tài chính của chúng ta bền vững. Điều này được chứng minh bằng việc tiết kiệm nước, điện và không gian. Ngoài ra chúng ta còn góp phần bảo vệ môi trường sống chẳng hạn như làm giảm sự phát triển của hạn hán, xói mòn đất và các vấn đề tương tự.

- ***Hạn chế sử dụng các chất hóa học đối với thực phẩm***



Hình 1.2: Hệ thống canh tác truyền thống sử dụng nhiều chất hóa học

- Nông nghiệp thông minh buộc nông dân phải trồng cây trồng một cách có kiểm soát và có ý thức hơn, dẫn đến việc trồng cây đơn thuần mà không cần đầu tư thêm. Làm thế nào là nó có thể? Lý do là vì nông dân buộc phải sử dụng thuốc trừ sâu hóa học để đảm bảo sản lượng tốt khi các yếu tố môi trường tác động xấu đối với cây trồng. Như chúng ta đã đề cập ở trên, thời tiết, yếu tố môi trường được giảm xuống tối thiểu ở nông nghiệp thông minh, do đó không

có nhu cầu thực sự sử dụng các chất điều chỉnh tăng trưởng hóa học.

- ***Cải thiện không gian sống, cung cấp lương thực sạch quanh năm và có cấu trúc đơn giản***

- Nông nghiệp thông minh có thể áp dụng ở nhiều nơi, kể cả những nơi có diện tích hạn chế, có thể sử dụng ngay trong nhà. Việc áp dụng nông nghiệp thông minh trong nhà sẽ cho phép bạn có thể tự quản lý ngay chính “Vườn thông minh” của chính mình, điều chỉnh hợp lý cho loại cây trồng mà chủ nhà áp dụng... Từ đó, việc sử dụng lương thực sạch cụ thể là rau, củ... sạch sẽ không còn là điều khó khăn do “vườn thông minh” của bạn có thể cung cấp lượng rau củ sạch quanh năm. Bên cạnh



Hình 1.3: Canh tác thực phẩm trong nhà



đó, việc đầu tư và lắp đặt vô cùng đơn giản nhưng đem lại hiệu quả vô cùng to lớn.

a. Tình hình nghiên cứu ngoài nước:

Có thể nói Israel, Singapore và Nhật Bản là những quốc gia đi tiên phong trong phát triển nông nghiệp cao với những công nghệ chăm sóc cây trồng vô cùng độc đáo như hệ thống tưới nhỏ giọt, hệ thống tưới phun sương, hệ thống tưới phun mưa, hệ thống tưới cảnh quan và ngay cả hệ thống thủy canh trồng cây không dùng đất. Và gần đây nhất hướng đến nhu cầu trồng cây trên những diện tích nhỏ hẹp nên có thêm giải pháp trồng rau xanh trên những nông trại thẳng đứng " Vertical farm " đang là xu hướng được quan tâm tại nhiều nơi hiện nay bởi vừa tận dụng tối đa diện tích canh tác, vừa cho rau sạch với sản lượng lớn, phục vụ nhu cầu cho nhiều hộ gia đình.

Tóm lại, hiện nay nước ngoài đã áp dụng hình thức "Vườn thông minh" từ rất lâu và cho rất nhiều nơi và ứng dụng này thậm chí còn được sử dụng tại ngay những hộ gia đình nhỏ. Chính vì sự tiện lợi mà nó đem lại nên nó dần trở thành ứng dụng khá phổ biến hiện nay trên toàn thế giới.



Hình 1.4: Trang trại thẳng đứng tại Nhật Bản

**b. Tình hình nghiên cứu trong nước:**

Nước ta hiện nay thuật ngữ nông nghiệp hiện đại không còn xa lạ gì với nhiều người nhưng do chi phí đầu tư hệ thống chăm sóc cây tự động khá cao do phải nhập chu yếu từ nước ngoài, dẫn đến việc nhân rộng mô hình chưa phổ biến, và hạn chế những hệ thống chăm sóc cây tự động chủ yếu hướng tới đối tượng người dùng phải có diện tích trồng cây rộng lớn, chưa chú trọng đến thị trường người dùng có diện tích trồng eo hẹp.

Trong nước cũng có rất nhiều nghiên cứu như hệ thống tưới cây tự động của kỹ sư Vi Toàn Nghĩa 2013, sau đó là hàng loạt công trình nghiên cứu hệ thống trồng cây điều khiển từ xa, hệ thống trồng cây tự động hướng đến cả phân khúc người sử dụng có quỹ tích eo hẹp. Nhưng hạn chế những nghiên cứu chưa tận dụng hạ tầng internet hiện có để mở rộng người dùng với hệ thống trồng cây có giám sát tự động từ xa, hướng đến cả người dùng không những có diện tích trồng eo hẹp, không có thời gian chăm sóc nhưng vẫn có vườn an toàn cho riêng mình.



**Hình 1.5:** Hệ thống tưới cây tự động của kỹ sư Vi Toàn Nghĩa

Từ những đánh giá trên, chúng em quyết định làm một hệ thống quản lý “Vườn thông minh” với tính năng đáp ứng được cơ bản nhu cầu của người sử dụng.

## **1.2 Nhiệm vụ đề tài**

■ Yêu cầu: Đề tài cần có những yêu cầu sau:

- Xây dựng một hệ thống nhỏ gọn hoạt động ổn định.
- Hệ thống có giao diện hợp lý, bắt mắt và dễ dàng thao tác với người sử dụng.
- Thích ứng được với nhiều loại cảm biến và được sử dụng rộng rãi.

Từ những yêu cầu trên chúng em quyết định sử dụng Arduino và OpenHAB để thiết kế đề tài này.

■ Kết quả cần đạt: Xây dựng được một mô hình thu nhỏ "vườn thông minh" cho phép người dùng kiểm soát và điều khiển chính khu vườn của mình.

Đề tài cần cung cấp những ứng dụng sau:

- Nhận dữ liệu bật tắt các công tắc (switch), đồng thời gửi giá trị trạng thái bật tắt hiện tại của nó lên webserver.
- Đọc các giá trị nhiệt độ, độ ẩm, quang trở và gửi lên webserver và hiển thị trên biểu đồ để người dùng dễ dàng theo dõi.
- Tự động bật tắt các công tắc dựa trên giá trị cảm biến ánh sáng.
- Cho phép 3 cơ chế tưới:
  - + Set bật tắt bằng tay (ON/OFF).
  - + Bật tưới ngay lập tức và có hẹn giờ tự động tắt
  - + Đặt lịch tưới theo tuần và hệ thống tự động tưới khi đến thứ/ giờ đó.

Để thực hiện được những kết quả trên cần phải tìm hiểu và thực hiện các nội dung:

- Nội dung 1: Tìm hiểu nguyên lý, lý thuyết về các giao thức lớp application (MQTT) và gói tin truyền trên OpenHAB và Arduino.
- Nội dung 2: Tìm hiểu về cảm biến DHT11, quang trở, Arduino Uno R3 & Ethernet shield.
- Nội dung 3: Tìm hiểu về OpenHAB, thiết lập những cài đặt liên quan.
- Nội dung 4: Thiết kế các giải thuật điều khiển và giao diện sao cho hợp lý và dễ dàng với người sử dụng (cả cho Arduino và OpenHAB).

■ Giới hạn đề tài: Bên cạnh những ưu điểm của việc sử dụng arduino và OpenHAB thì vẫn có những nhược điểm:




- Hệ thống công kênh, sử dụng tiêu tốn năng lượng do không có chế độ sleep.
- Phạm vi ứng dụng còn hạn chế do chưa kết nối với các cảm biến chuyên dụng cho nông nghiệp thông minh như cảm biến đo độ PH đất, nước, tính toán chất dinh dưỡng trong đất.... và chưa có chế độ tưới theo độ ẩm đất.

### 1.3 Phân chia công việc trong nhóm

Thời gian Nội dung	Năm 2016				Năm 2017																																			
	Tháng 12				Tháng 1				Tháng 2				Tháng 3				Tháng 4				Tháng 5				Tháng 6															
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4												
Nội dung 1																																								
Nội dung 2																																								
Nội dung 3																																								
Nội dung 4																																								
Nội dung 5																																								
Nội dung 6																																								
Nội dung 7																																								
Nội dung 8																																								
Nội dung 9																																								
Nội dung 10																																								

Bảng 1: Lịch phân chia công việc theo tuần

• CHÚ THÍCH:

- Người thực hiện: Bảo Anh 
- Quốc Thái 
- Cả hai 

• Nội dung thực hiện:

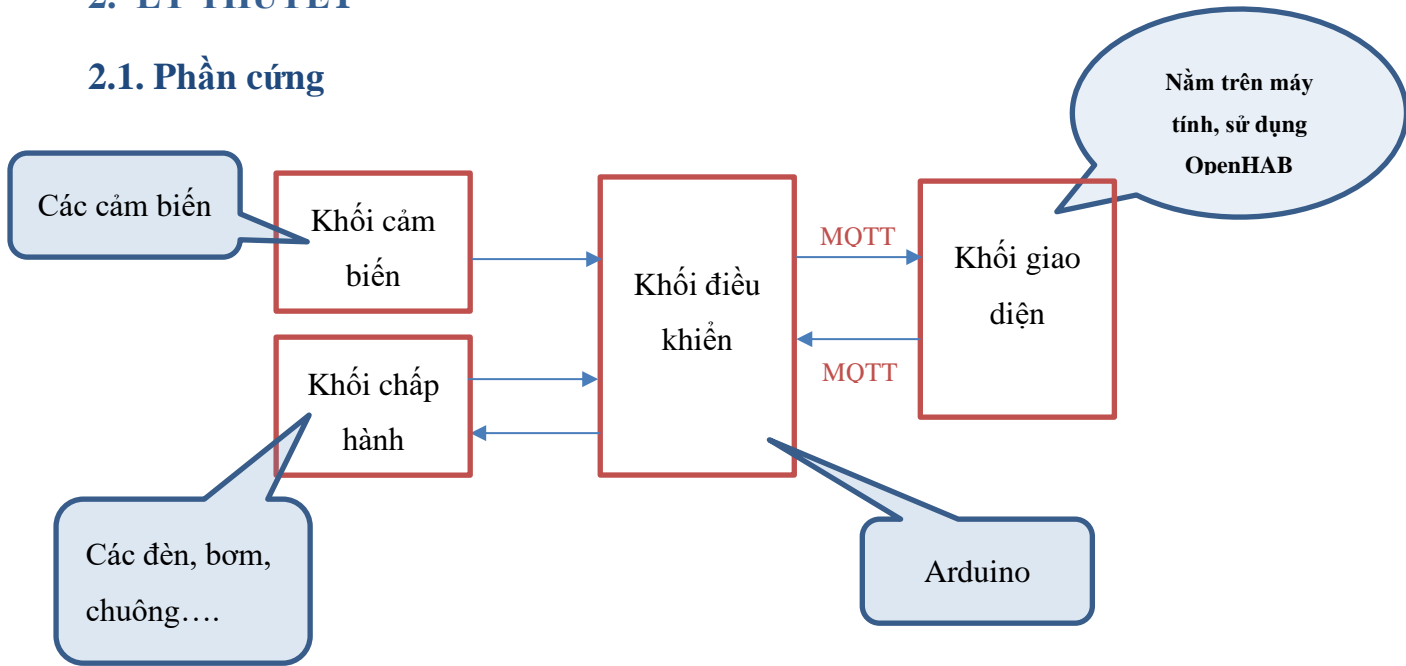
- Nội dung 1:** Tìm hiểu nguyên lý, lý thuyết về các giao thức lớp application (MQTT) và gói tin truyền trên OpenHAB và Arduino.
- Nội dung 2:** Tìm hiểu về cảm biến DHT11, quang trở, Arduino Uno R3 & Ethernet shield.
- Nội dung 3:** Tìm hiểu về OpenHAB, thiết lập những cài đặt liên quan.
- Nội dung 4:** Thiết kế phần cứng và tìm hiểu về kết nối các thiết bị
- Nội dung 5:** Tìm hiểu và nghiên cứu code cho Arduino
- Nội dung 6:** Tìm hiểu và nghiên cứu code trên OpenHAB
- Nội dung 7:** Hỗ trợ code giao diện cho OpenHAB
- Nội dung 8:** Kết nối lại với phần cứng và kiểm tra hoạt động của đề tài
- Nội dung 9:** Phát triển sản phẩm sao cho hoàn thiện và dễ dàng thao tác cho người sử dụng
- Nội dung 10:** Viết báo cáo và làm slide thuyết trình

✓ Quy định riêng:

- Deadline là 1 tuần từ ngày có công việc phát sinh
- Có thể hỗ trợ công việc lẫn nhau trong quá trình làm việc
- Lắng nghe kỹ càng các ý kiến của thành viên khác
- Thời gian họp team là sáng thứ 2 và chiều thứ 5 hàng tuần
- Phần kiểm tra hoạt động bắt buộc phải tham gia và đóng góp ý kiến

## 2. LÝ THUYẾT

### 2.1. Phần cứng



#### 2.1.1. Arduino

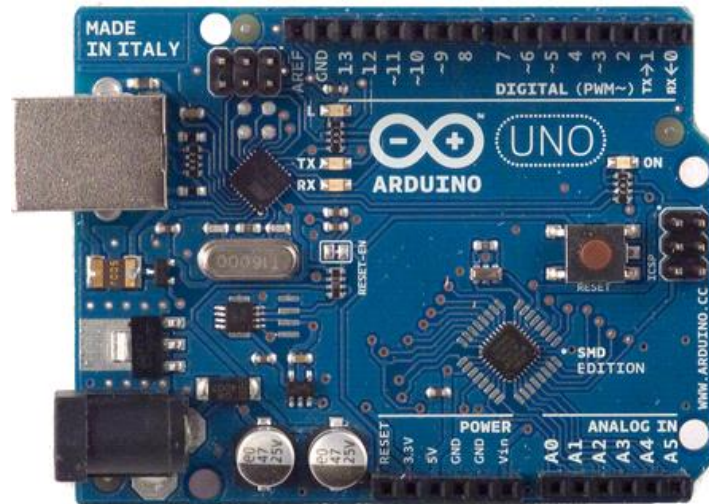
##### 2.1.1.a. Giới thiệu về arduino

- Arduino là một board mạch vi xử lý, nhằm xây dựng các ứng dụng tương tác với nhau hoặc với môi trường được thuận lợi hơn. Phần cứng bao gồm một board mạch nguồn mở được thiết kế trên nền tảng vi xử lý AVR Atmel 8bit, hoặc ARM Atmel 32-bit. Những Model hiện tại được trang bị gồm 1 cổng giao tiếp USB, 6 chân đầu vào analog, 14 chân I/O kỹ thuật số tương thích với nhiều board mở rộng khác nhau.

##### 2.1.1.b. Arduino Uno R3

- Arduino UNO có thể sử dụng 3 vi điều khiển họ 8bit AVR là ATmega8, ATmega168, ATmega328. Bộ não này có thể xử lý những tác vụ đơn giản như điều khiển đèn LED nhấp nháy, xử lý tín hiệu cho xe điều khiển từ xa, làm một trạm đo nhiệt độ - độ ẩm và hiển thị lên màn hình LCD,...





*Hình 2.1: Arduino UNO R3*

**\* Một vài thông số**

Vi điều khiển	ATmega328 họ 8bit
Điện áp hoạt động	5V DC (chỉ được cấp qua cổng USB)
Tần số hoạt động	16MHz
Dòng tiêu thụ	Khoảng 30mA
Điện áp vào khuyến dùng	7-12V DC
Điện áp vào giới hạn	6-20V DC
Số chân Digital I/O	14 (6 chân hardware PWM)
Số chân Analog	6 (độ phân giải 10bit)
Dòng tối đa trên mỗi chân I/O	30mA
Dòng ra tối đa (5V)	500mA
Dòng ra tối đa (3.3V)	50mA
Bộ nhớ flash	32KB (ATmega328) với 0.5 KB dùng bởi bootloader
SRAM	2KB (ATmega328)
EEPROM	1KB (ATmega328)

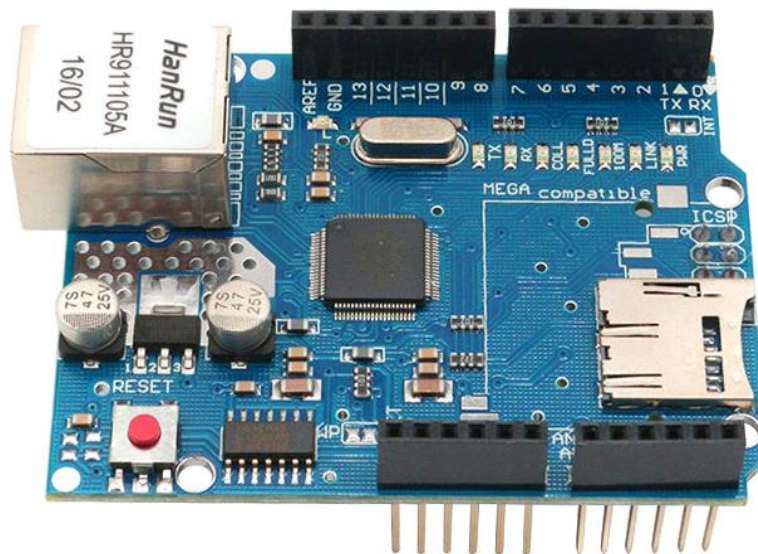
*Bảng 2: Một vài thông số của Arduino*

### 2.1.2. Ethernet shield

- Ethernet shield là một mạch mở rộng cho arduino, giúp arduino có thể kết nối với thế giới internet rộng lớn. Ứng dụng của shield này là truyền nhận thông tin giữa arduino với thiết bị bên ngoài sử dụng internet, shield này đặc biệt hữu ích cho các ứng dụng IoT, điều khiển và kiểm soát hệ thống vì internet luôn liên tục, dữ liệu truyền đi nhanh, làm đơn giản đi vấn đề về khoảng cách.

- Có tích hợp khe cắm thẻ micro SD, có thể được sử dụng để lưu trữ các tập tin phục vụ qua mạng. *Arduino Ethernet Shield* tương thích với Arduino Uno và Mega (sử dụng Ethernet Thư viện). Có thể truy cập vào khe cắm thẻ trên board và sử dụng thư viện SD được bao gồm tích hợp trong bộ thư viện có sẵn trong trình biên dịch arduino.

IC điều khiển W5100 trên Arduino Ethernet Shield có thể thực hiện truyền dữ liệu thông qua 2 giao thức là TCP và UDP. Số đường truyền dữ liệu song song tối đa là 4.



Hình 2.2: *Etherner Shield*

- Để sử dụng phải có board mạch Arduino đi kèm
- Hoạt động tại điện áp 5V (được cấp từ mạch Arduino)
- Chip Ethernet: W5100 với buffer nội 16KB
- Tốc độ kết nối: 10/100Mb
- Kết nối với mạch Arduino qua cổng SPI
- Thư viện và code mẫu có sẵn trong chương trình Arduino

### 2.1.3. Cáp mạng RJ45

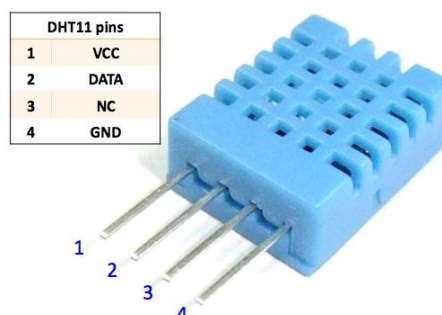


*Hình 2.3: Cáp RJ45*

Sau khi thực hiện ghép Arduino với Ethernet Shield (gọi tắt là Arduino) thì cần sử dụng cáp RJ45 để kết nối Arduino với các modun mạng hay kết nối trực tiếp với máy tính như trong cách thực hiện này để tiện cho quá trình demo.

### 2.1.4. Cảm biến DHT11

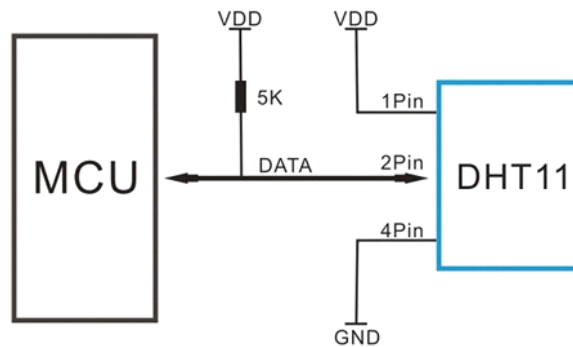
*Hình 2.4: DHT11*



DHT11 pins	
1	VCC
2	DATA
3	NC
4	GND



- DHT11 là cảm biến nhiệt độ và độ ẩm. Nó ra đời sau và được sử dụng thay thế cho dòng SHT1x ở những nơi không cần độ chính xác cao về nhiệt độ và độ ẩm.
- DHT11 có cấu tạo 4 chân như hình. Nó sử dụng giao tiếp số theo chuẩn 1 dây.
- Thông số kỹ thuật:
  - o Do độ ẩm: 20%-95%
  - o Nhiệt độ: 0-50°C
  - o Sai số độ ẩm  $\pm 5\%$
- Sơ đồ kết nối vi xử lý:



Hình 5.1: Sơ đồ kết nối

- Nguyên lý hoạt động:

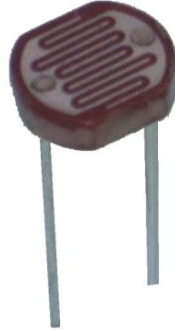
Để có thể giao tiếp với DHT11 theo chuẩn 1 chân vi xử lý thực hiện theo 2 bước:

  - o Gửi tín hiệu muốn đo (Start) tới DHT11, sau đó DHT11 xác nhận lại.
  - o Khi đã giao tiếp được với DHT11, Cảm biến sẽ gửi lại 5 byte dữ liệu và nhiệt độ đo được.
- Bước 1: gửi tín hiệu Start
- Bước 2: đọc giá trị trên DHT11

### 2.1.5. Quang trở

- Quang trở là một loại "vật liệu" điện tử rất hay gặp và được sử dụng trong những mạch cảm biến ánh sáng. Hay quang trở là một loại điện trở có giá trị thay đổi theo cường độ ánh sáng. Nếu đặt ở môi trường có ít ánh sáng, có bóng

râm hoặc tối thì điện trở của quang trở sẽ tăng cao còn nếu đặt ở ngoài nắng, hoặc nơi có ánh sáng thì điện trở sẽ giảm.



*Hình 2.6: Quang trở*

## 2.2 Phần mềm

### 2.2.1 OpenHAB

- OpenHAB là một phần mềm mã nguồn mở có chức năng là bộ điều khiển trung tâm, với khả năng kết nối giao tiếp và điều khiển tới nhiều loại thiết bị khác nhau trong hệ thống SmartHome. OpenHab cung cấp nhiều giao diện người dùng (web site, android, ios....) giúp cho quá trình làm việc với OpenHab dễ dàng và thuận tiện hơn.

### 2.2.2 OpenHAB designer

- Là phần mềm hỗ trợ cho việc thiết lập:
  - 1 trang mạng cho openHAB cùng với địa chỉ của nó.
  - Giao diện openHAB.
  - MQTT cho openHAB.
- Các giao thức truyền nhận dữ liệu, cũng như các quy luật làm việc cho openHAB...

### 2.2.3 Java

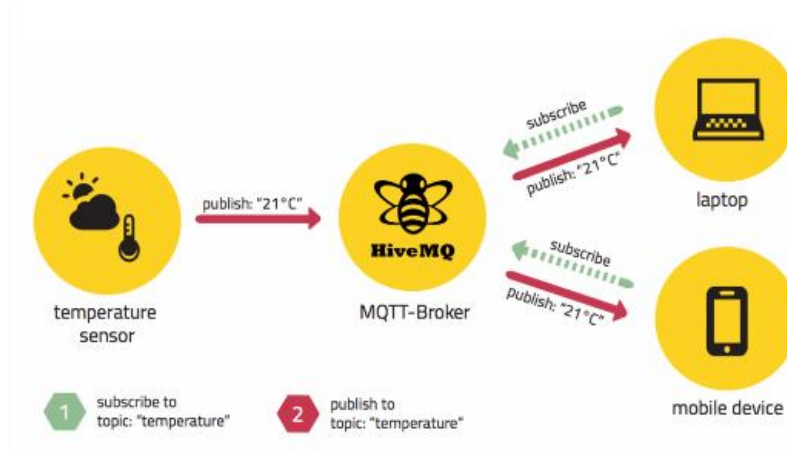
- OpenHAB được cài đặt dưới dạng một website, có thể chạy trên rất nhiều platform (Windows, Linux, ARM...), nó được viết bằng ngôn ngữ Java. Vì vậy các thiết bị cần có hỗ trợ JVM (Java Virtual Machine) để có thể cài đặt được Openhab.

## 2.2.4 MQTT

- MQTT (Message Queuing Telemetry Transport) là một giao thức gửi dạng publish/subscribe sử dụng cho các thiết bị Internet of Things với băng thông thấp, độ tin cậy cao và khả năng được sử dụng trong mạng lưới không ổn định.
- Giao thức này sử dụng băng thông thấp trong môi trường có độ trễ cao nên nó là một giao thức lý tưởng cho các ứng dụng M2M và IoT.

### ■ Mô hình Publish/Subscribe

- Mô hình publish/subscribe (pub / sub) là một thay thế cho mô hình client-server truyền thống. Tuy nhiên Pub / Sub có những điểm đặc biệt, đó chính là nhà xuất bản (publish) và người đăng ký (subscribe) không biết về sự tồn tại của nhau. Có một thành phần thứ ba, được gọi là môi giới, được biết bởi cả nhà xuất bản và thuê bao, bộ lọc này lọc tất cả các tin nhắn gửi đến và phân phối chúng cho phù hợp.



- Việc tách riêng Pub/Sub có 3 chiều:
  - +Không gian : Nhà xuất bản và thuê bao không cần phải biết nhau (theo chỉ ip và cổng)
  - +Thời gian tách: Nhà xuất bản và người đăng ký không cần phải chạy đồng thời.
  - +Đồng bộ: Các thao tác trên cả hai thành phần không bị dừng trong quá trình xuất bản hoặc tiếp nhận

### ■ Kết nối MQTT

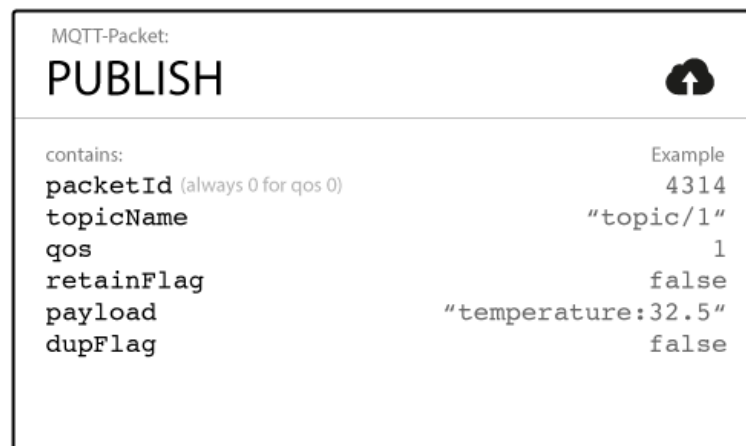
- Giao thức MQTT dựa trên giao thức TCP / IP. Cả khách hàng (MQTT client) và nhà môi giới (broker) đều cần có ngăn TCP / IP.



- Kết nối được bắt đầu thông qua một máy khách gửi một thông báo CONNECT tới người môi giới. Phản hồi của người môi giới với CONNACK và mã trạng thái. Khi kết nối được thiết lập, người môi giới sẽ giữ kết nối luôn mở, miễn là khách hàng không gửi lệnh ngắt kết nối hoặc mất kết nối.

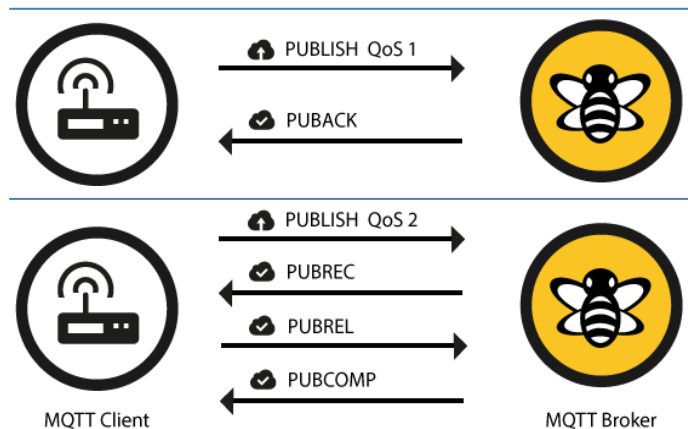
### ■ MQTT Publish, Subscribe & Unsubscribe

- **Publish:** Sau khi một MQTT client được kết nối với một broker, client có thể publish tin nhắn. MQTT có một bộ lọc theo chủ đề dựa trên các message trên broker, do đó mỗi tin nhắn phải chứa một chủ đề, broker dễ dàng phân loại và chuyển tiếp thông điệp tới những khách hàng nào quan tâm.



- Topic Name: Một chuỗi đơn giản, được cấu trúc theo thứ bậc với dấu gạch chéo phía trước như các dấu phân cách. Ví dụ như là "myhome / livingroom / temperature" hoặc "Germany / Munich / Oktoberfest / people".
- Qos: Mức Chất lượng Dịch vụ (QoS). Mức (0,1 hoặc 2) xác định sự bảo đảm của một thông điệp đến đầu kia (client hoặc broker).





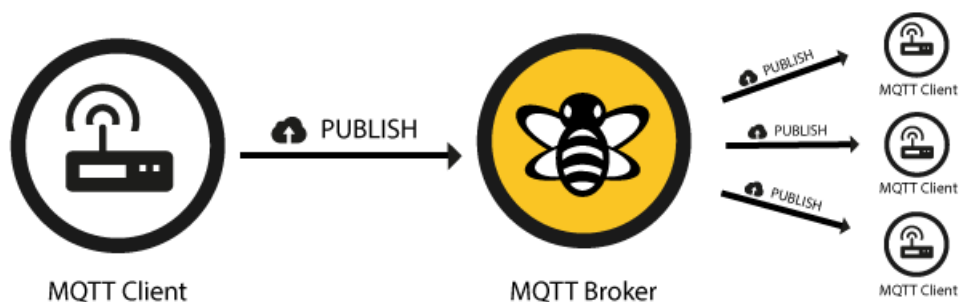
➤ **Retain-Flag:** Cờ này được xác định nếu message sẽ được lưu bởi broker cho một chủ đề nhất định. Client mới có thể đăng ký vào chủ đề đó và sẽ nhận được message được giữ lại cuối cùng về chủ đề đó ngay sau khi đăng ký.

➤ **Payload:** Đây là nội dung thực tế của tin nhắn.

➤ **Packet Identifier:** Bộ nhận dạng gói tin là sự nhận dạng duy nhất giữa client và broker để xác định một gói tin trong luồng chảy dữ liệu. Chỉ được đề cập khi sử dụng các QoS lớn hơn QoS 0. Việc đặt mã nhận dạng nội bộ MQTT này là trách nhiệm của thư viện ở client và / hoặc là trách nhiệm của broker.

➤ **DUP Flag:** Cờ trùng lặp chỉ ra, rằng message này là một bản sao và bị gửi lại bởi vì đầu kia không chấp nhận message gốc. Điều này chỉ xảy ra ở các QoS lớn hơn QoS 0. Cơ chế gửi lại / trùng lặp này thường được xử lý bởi thư viện ở MQTT client hoặc broker.

⇒ Vì vậy, khi một client gửi publish tới một MQTT broker, broker sẽ đọc, chấp nhận mess nếu cần (theo Cấp QoS) và sau đó xử lý nó. Quá trình xử lý bao gồm xác định những client nào đã đăng ký vào chủ đề đó và sau đó gửi thông tin đó tới các khách hàng đã chọn đăng ký vào chủ đề đó.

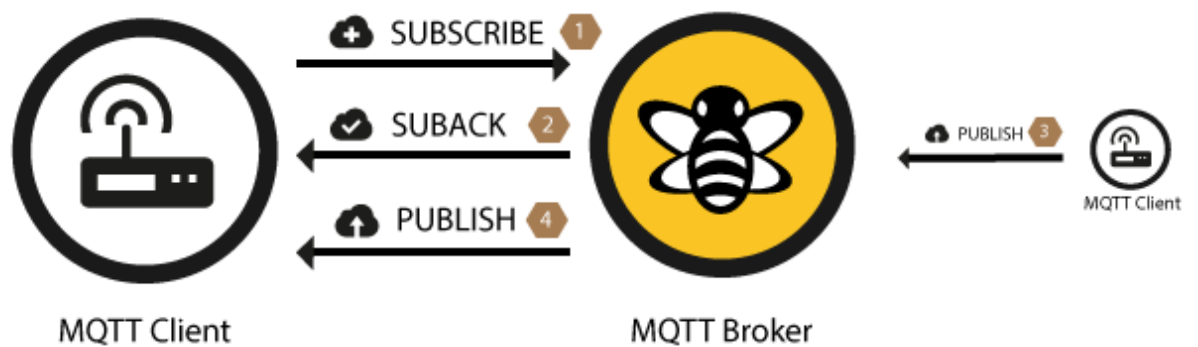


• **Subscribe và Suback:** là quá trình đăng kí để nhận dữ liệu từ broker của 1 client nào đó. Quá trình đó diễn ra bắt đầu khi một client bất kì muốn đăng kí nhận dữ liệu cho một topic của broker, client sẽ gửi gói subscribe đến, khi đó broker sẽ chấp nhận và gửi lại gói Suback có ý nghĩa là chấp nhận và bắt đầu gửi dữ liệu publish cho client.

MQTT-Packet: SUBSCRIBE	
contains:	Example
<b>packetId</b>	4312
<b>qos1</b> } (list of topic + qos)	1
<b>topic1</b>	"topic/1"
<b>qos2</b> }	0
<b>topic2</b>	"topic/2"
...	...

MQTT-Packet: SUBACK	
contains:	Example
<b>packetId</b>	4313
<b>returnCode 1</b>	(one returnCode for each topic from SUBSCRIBE, in the same order)
<b>returnCode 2</b>	2
...	0
	...

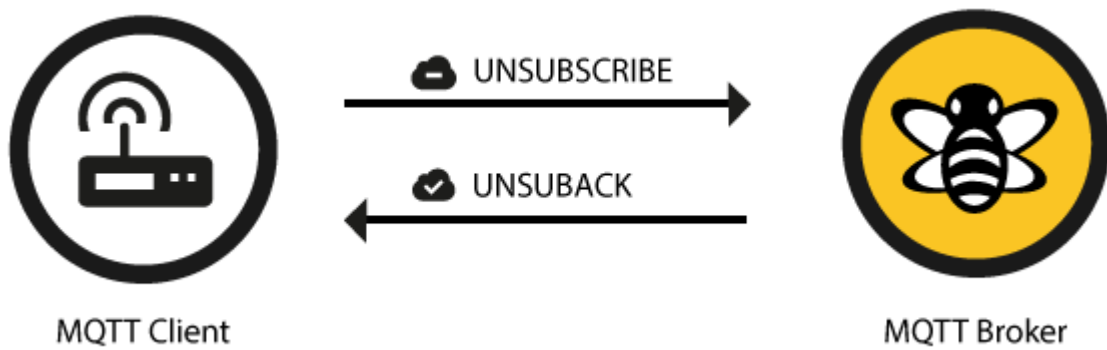
Return Code	Return Code Response
0	Success – Maximum QoS 0
1	Success – Maximum QoS 1
2	Success – Maximum QoS 2
128	Failure



- **Unsubscribe và Unsuback:** Ngược lại với subscribe và suback, khi client không muốn nhận dữ liệu của 1 topic nào đó, nó sẽ gửi gói unsubscribe và khi đó, broker sẽ trả lại gói unsuback xác nhận kết thúc quá trình truyền nhận dữ liệu và không gửi dữ liệu topic đó cho client nữa.

MQTT-Packet: UNSUBSCRIBE	
contains:	Example
<b>packetId</b>	4315
<b>topic1</b> } (list of topics)	"topic/1"
<b>topic2</b>	"topic/2"
...	...

MQTT-Packet: UNSUBACK	
contains:	Example
<b>packetId</b>	4316



- Thiết lập và liên kết được với MQTT là mấu chốt cho việc thực hiện đề tài này.
- Việc cài đặt MQTT thực hiện bởi việc thêm tập tin 'org.openhab.binding.mqtt-1.8.3' vào mục addons của openhab.
- Kể đến nó được đặt địa chỉ thông qua việc thiết kế openHAB và thiết kế code trên Arduino.

### 2.2.5 Mosquitto

- Mosquitto là một phần mềm mã nguồn mở MQTT broker sử dụng giao thức MQTT version 3.1 và 3.1.1. Nó đóng vai trò trung gian để tiếp nhận, xử lý và chuyển tiếp các message đến các thiết bị trong hệ thống. Data được truyền và nhận thông qua các topic (hay còn gọi là channel). Với các channel, ta có thể gửi data tới đó hoặc đăng ký nhận bất kỳ dữ liệu nào gửi tới nó.

### 2.2.6 Arduino IDE

- Arduino IDE là phần mềm hỗ trợ phát triển tích hợp IDE (Integrated Development Environment) dùng để soạn thảo, biên dịch code và nạp chương cho board arduino và các module vi điều khiển khác. Trình biên dịch Arduino IDE sử dụng phiên bản đơn giản hóa của ngôn ngữ C++.

### 2.2.7 MQTT lends

- MQTT lends là công cụ được hỗ trợ trên cốc cốc có chức năng quan sát, xác nhận khi các liên kết được thiết lập, quan sát quá trình truyền nhận dữ liệu.

### **2.3 Ngôn ngữ**

- Sử dụng ngôn ngữ C++ để thiết kế phần code cho Arduino và ngôn ngữ Java để thiết kế giao diện web openhab.

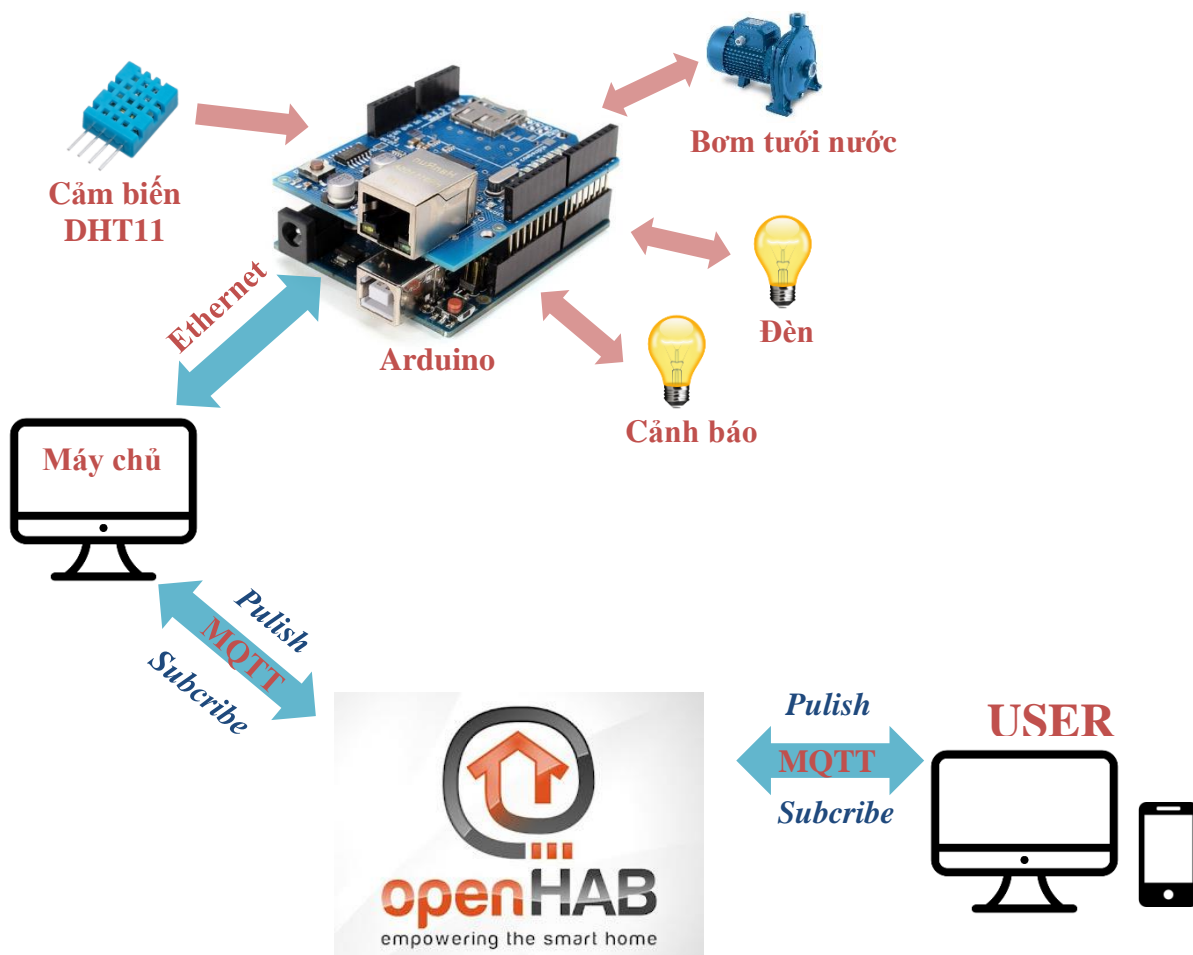


### 3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG

#### 3.1. Các linh kiện sử dụng:

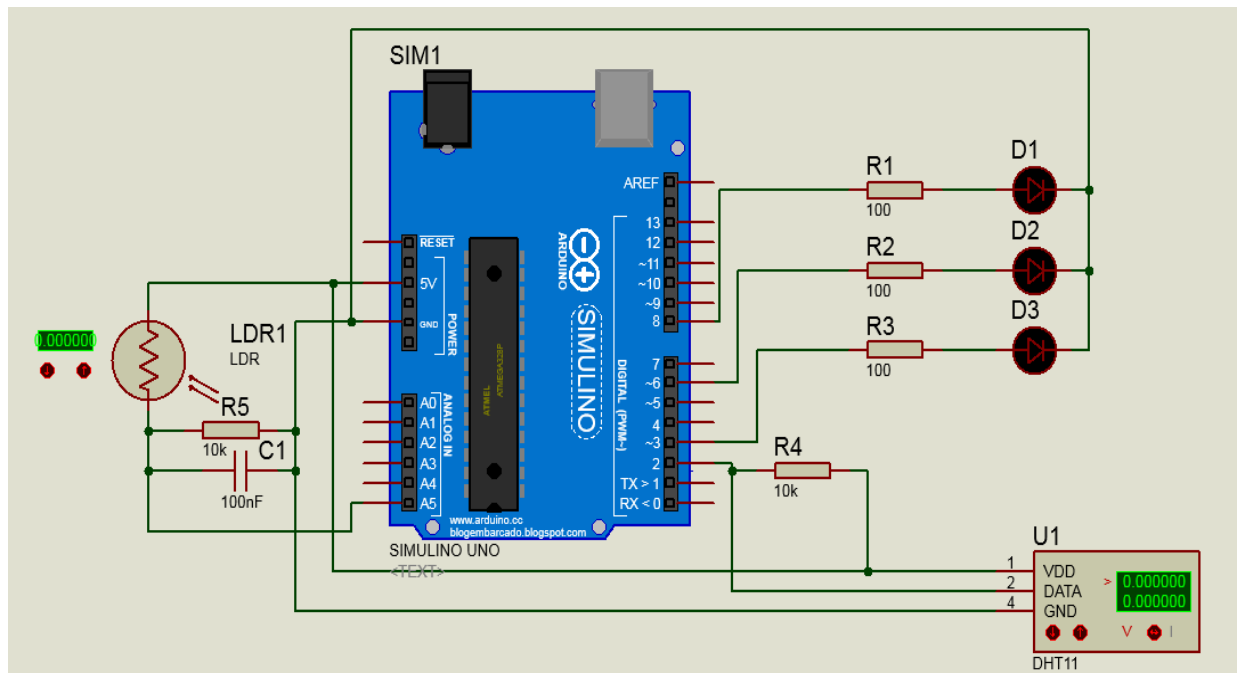
- Arduino Uno R3 và Ethernet Shield
- Cảm biến nhiệt độ, độ ẩm DHT11
- Quang trở
- Các điện trở
- LED

#### 3.2. Sơ đồ khối kết nối phần cứng



Hình 3.1: Sơ đồ kết nối phần cứng

### 3.3. Sơ đồ thiết kế phần cứng



Hình 3.2: Schematic

- Cảm biến DHT11 dùng để xác định giá trị nhiệt độ, độ ẩm, hoạt động bởi nguồn 5V, chân dữ liệu nối vào chân digital số 2 của Arduino.
- Quang trở LDR1 dùng để xác định giá trị điện trở theo ánh sáng, hoạt động bởi nguồn 5V, tín hiệu được nối vào chân analog A5.
- Các đèn led tượng trưng cho các công tắc, switch:
  - D3 nối vào chân digital số 3 của Arduino dùng để kiểm tra, thực thi hoạt động bật tắt.
  - D2 nối vào chân digital số 6 của Arduino dùng để kiểm tra, thực thi hoạt động bật tắt.
  - D1 nối vào chân digital số 8 của Arduino, thực thi bật tắt theo giá trị quang trở.

### 3.3. Layout

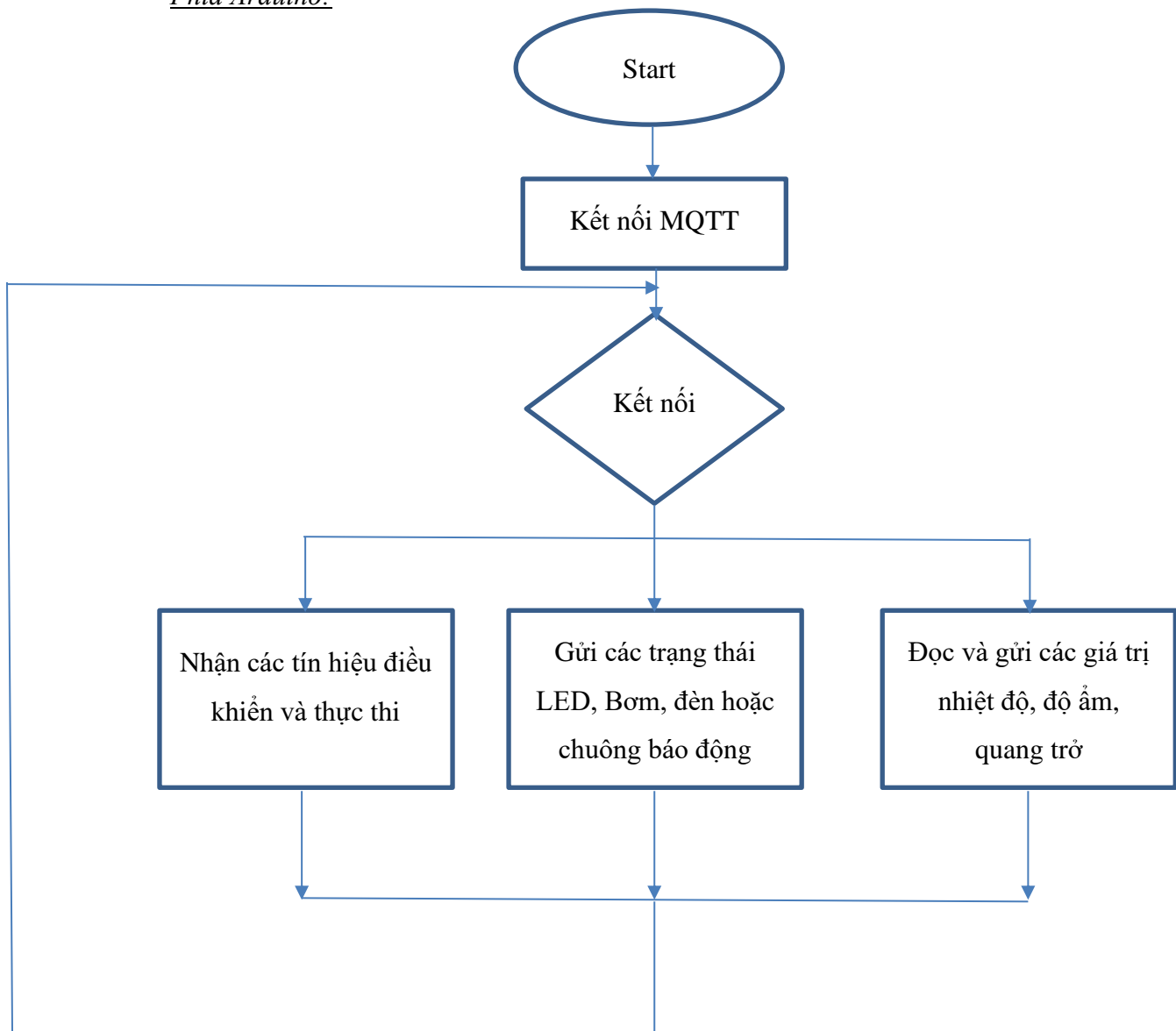
## 4. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM

### 4.1. Các vấn đề khi thiết kế:

- Yêu cầu đặt ra cho phần mềm (ghi rõ ra cụ thể như độ tin cậy cao, thời gian đáp ứng nhanh, giao diện dễ sử dụng...)
  - Phần mềm phải đáp ứng được những nhu cầu của người sử dụng.
  - Những ứng dụng phải phù hợp với thực tế.
  - Viết ứng dụng giao diện phải hợp lý và dễ dàng thao tác cho người sử dụng.
- Phân tích: Muốn đạt được những yêu cầu trên cần thiết lập một số các
  - Muo

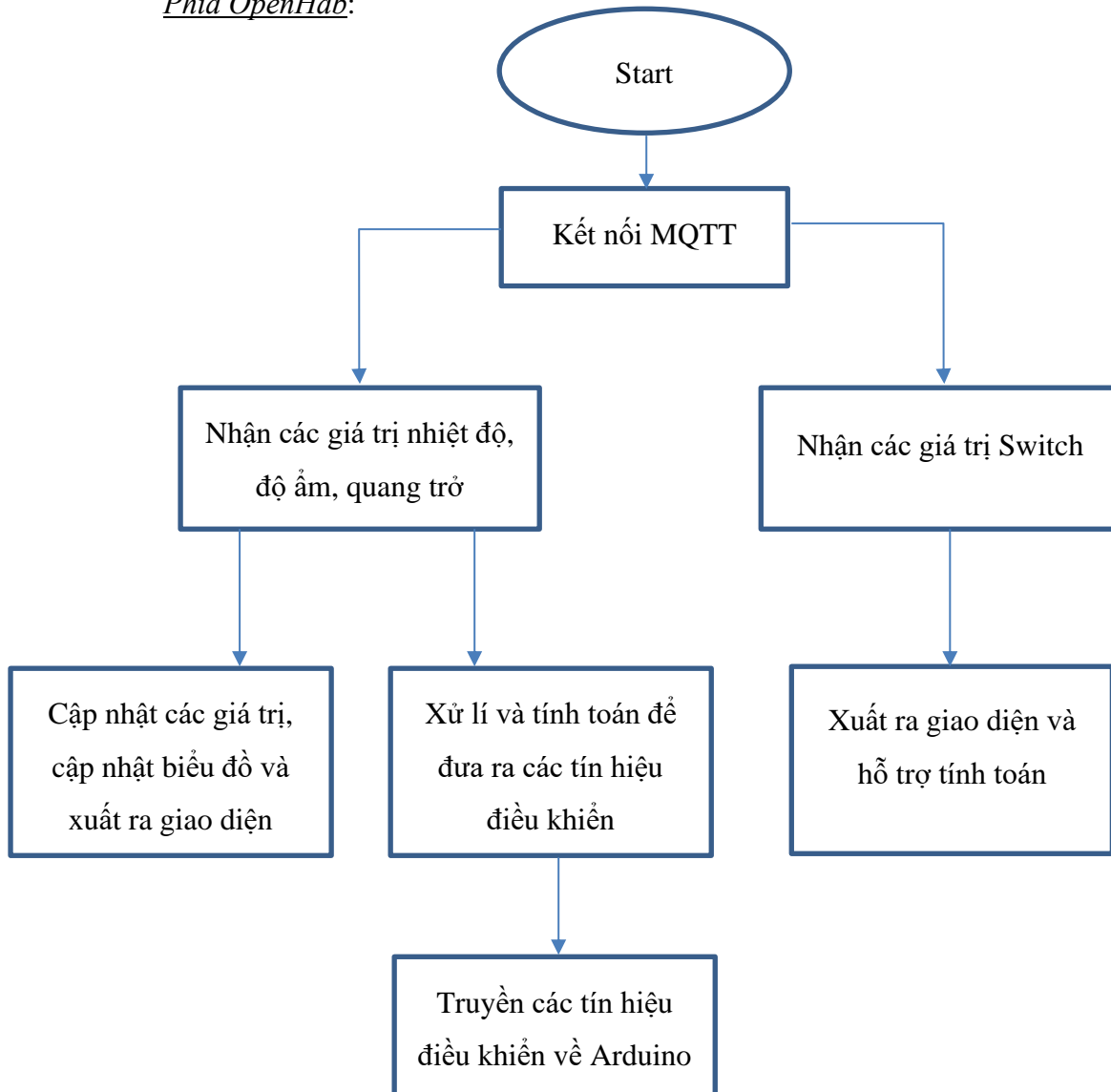
- Lưu đồ giải thuật tổng quát (ghi rõ các thông điệp truyền giữa máy tính và ar )

Phía Arduino:



Hình 4.1: Lưu đồ giải thuật bên phía Arduino

Phía OpenHab:



Hình 4.2: Lưu đồ giải thuật bên phía OpenHAB

- Lưu đồ giải thuật chi tiết  
Bật đèn theo cảm biến ánh sáng

Bật cảnh báo dựa vào cảm biến nhiệt độ

Tưới nước set bằng tay và tự động hẹn giờ tắt

Tưới nước theo lịch trình đặt sẵn theo tuần

## 5. KẾT QUẢ THỰC HIỆN

- Người dùng có thể chạy ứng dụng trên nhiều thiết bị từ smartphone, máy tính bảng cho đến máy tính cá nhân. Theo đó, người dùng có thể quản lý hệ thống vườn cây của mình từ xa thông qua kết nối Internet tại bất cứ đâu. Trong phần báo

cáo này chúng em xin được trình bày thông qua máy tính cá nhân ( chạy OpenHAB trên windows)

- Những tính năng cơ bản của Smart Garden có thể vận hành tự động hoặc bán tự động, bao gồm: Điều khiển hệ thống trực tiếp từ các thiết bị di động và máy tính, theo dõi các điều kiện trồng (nhiệt độ, độ ẩm, ánh sáng,...) thông qua biểu đồ ( theo giờ, ngày, tháng..), cảnh báo khi có sự cố gây nguy hại cho vườn cây và khả năng tưới nước bán tự động (điều khiển bằng tay bật hoặc tắt ), hẹn giờ tưới hoặc tưới theo điều kiện môi trường ( có thể là độ ẩm hoặc nhiệt độ)
- Mô hình sản phẩm bao gồm một arduino kết nối với ethernet shield kết nối với máy tính cá nhân, các cảm biến... Từ đó chúng ta có thể dễ dàng thao tác các tính năng như trên.

## **6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

### **6.1 Kết luận**

### **6.2 Hướng phát triển**

## **7. TÀI LIỆU THAM KHẢO**

Tài liệu và một số trang web tham khảo:

- [1] <https://community.openhab.org/> trang web mã nguồn mở để chia sẻ những đoạn code thực thi cho những tính năng nào đó trên OpenHAB.



## 8. PHỤ LỤC

### CÁC BƯỚC CÀI ĐẶT PHẦN MỀM ĐỂ THỰC HIỆN ĐỀ TÀI

#### I. CÀI ĐẶT MOSQUITO

Mosquito là một MQTT broker nguồn mở, đóng vai trò trung gian giúp cho các chương trình và thiết bị dễ dàng trao đổi data với nhau. Data được truyền và nhận thông qua các channel (hay còn gọi là topic). Với các channel ta có thể gửi data tới đó hoặc đăng kí nhận bất kì dữ liệu nào nó gửi tới nó.

##### 1. CHUẨN BỊ

- File cài đặt mosquito: Tải về theo đường link

<https://mosquitto.org/download/>

Sau đó tải bản win32.exe

**Donation**

Developing Mosquitto takes a not insignificant amount of time and effort. If you find it useful, please consider making a donation.

[Donate](#)

**Source**

- [mosquitto-1.4.12.tar.gz](#) (319kB) (GPG signature)
- [mosquitto-1.4.12.tar.gz](#) (via Eclipse)
- [Git source code repository](#) (github.com)

Older downloads are available at <http://mosquitto.org/files/>

The deprecated javascript/websockets client is available as:

- <http://mosquitto.org/js/mosquitto-1.1.js>
- <http://mosquitto.org/js/mosquitto-1.1.min.js>

Please consider using the [Paho Javascript](#) client instead.

**Binary Installation**

**Windows**

- [mosquitto-1.4.12-install-win32.exe](#) (~200 kB) (Native build, Windows Vista and up, built with Visual Studio Community 2013)
- [mosquitto-1.4.12-install-cygwin.exe](#) (~200 kB) (Cygwin build, Windows XP and up)

See the [readme-windows.txt](#) after installing for Windows specific details and dependencies.

**Mac**

Mosquitto can be installed from the [homebrew](#) project. See <http://brew.sh/> and then use "brew install mosquitto"

**Bugs / Support**

[Documentation](#)

[Python](#)

[Downloads](#)

**Links**

[Mosquitto test server](#)

[MQTT community](#)

[mqtt2cosm](#)

[Paho](#)

**Donate**

If you find mosquitto useful, please consider a donation to help support development and the running of the test server.

[Donate](#)

**Meta**

[Log in](#)

[Entries RSS](#)

[Comments RSS](#)

[WordPress.org](#)

- Tải các thư viện về để cài đặt

- Tải OpenSSL tại địa chỉ:

Download Win32 OpenSSL today using the links below!

File	Type	Description
<a href="#">Win32 OpenSSL v1.1.0f Light</a>	3MB Installer	Installs the most commonly used essentials of Win32 OpenSSL v1.1.0f (Recommended for users by the creators of <a href="#">OpenSSL</a> ). Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
<a href="#">Win32 OpenSSL v1.1.0f</a>	30MB Installer	Installs Win32 OpenSSL v1.1.0f (Recommended for software developers by the creators of <a href="#">OpenSSL</a> ). Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
<a href="#">Win64 OpenSSL v1.1.0f Light</a>	3MB Installer	Installs the most commonly used essentials of Win64 OpenSSL v1.1.0f (Only install this if you need 64-bit OpenSSL for Windows. Only installs on 64-bit versions of Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
<a href="#">Win64 OpenSSL v1.1.0f</a>	33MB Installer	Installs Win64 OpenSSL v1.1.0f (Only install this if you are a software developer needing 64-bit OpenSSL for Windows. Only installs on 64-bit versions of Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
<a href="#">Win32 OpenSSL v1.0.2L Light</a>	2MB Installer	Installs the most commonly used essentials of Win32 OpenSSL v1.0.2L (Recommended for users by the creators of <a href="#">OpenSSL</a> ). Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
<a href="#">Win32 OpenSSL v1.0.2L</a>	20MB Installer	Installs Win32 OpenSSL v1.0.2L (Recommended for software developers by the creators of <a href="#">OpenSSL</a> ). Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
<a href="#">Win64 OpenSSL v1.0.2L Light</a>	3MB Installer	Installs the most commonly used essentials of Win64 OpenSSL v1.0.2L (Only install this if you need 64-bit OpenSSL for Windows. Only installs on 64-bit versions of Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
<a href="#">Win64 OpenSSL v1.0.2L</a>	23MB Installer	Installs Win64 OpenSSL v1.0.2L (Only install this if you are a software developer needing 64-bit OpenSSL for Windows. Only installs on 64-bit versions of Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.

**Donate to Shining Light Productions**

Shining Light Productions puts forth a lot of effort into developing Win32 OpenSSL. As such, if you find it useful, a time-saver, or helps to solve a frustrating problem, seriously consider giving a donation to continue developing this software.

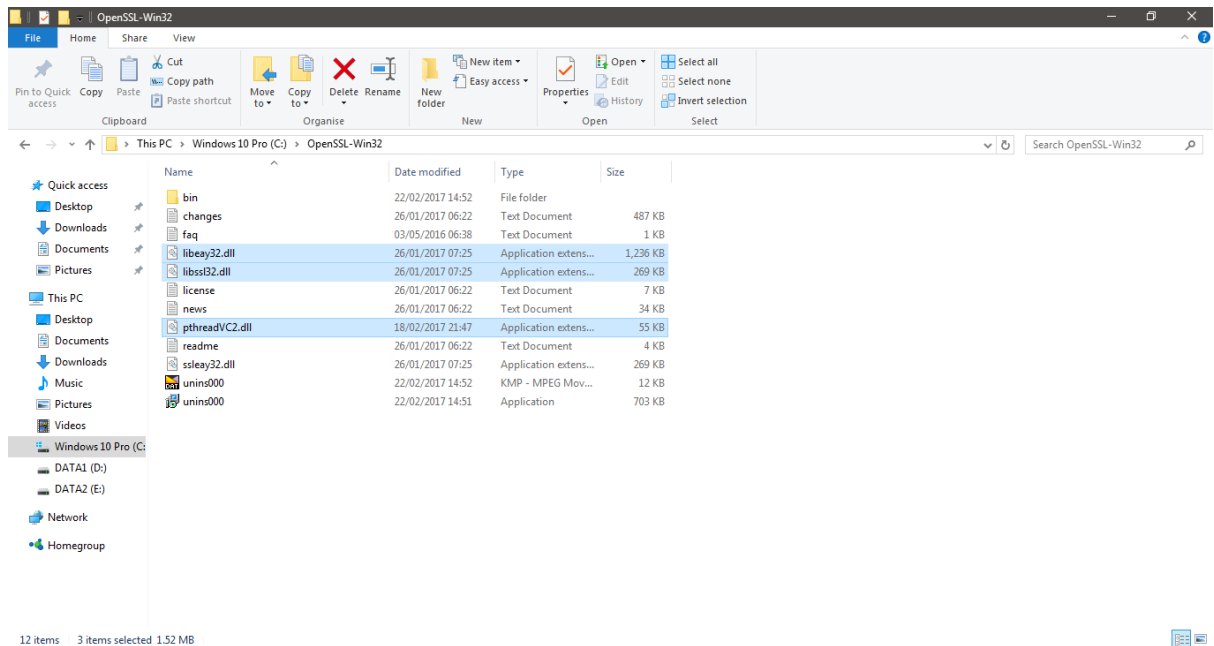
Shining Light Productions uses PayPal for all donations because it is fast, easy, and secure.

- Tải pthreads tại địa chỉ:

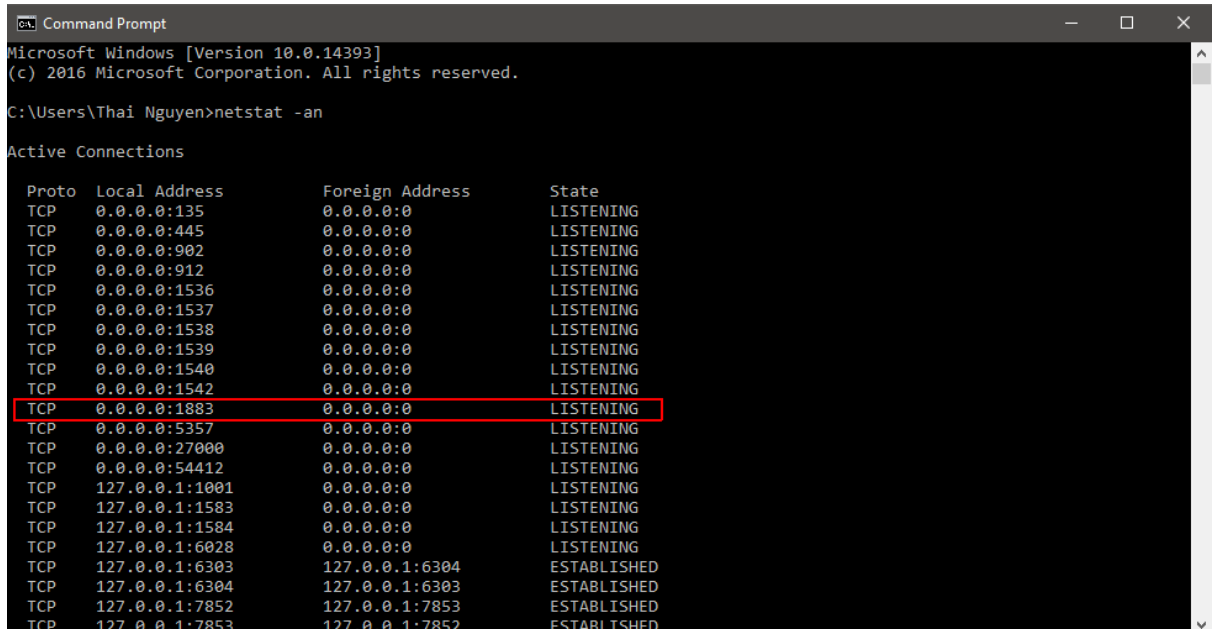
<ftp://sources.redhat.com/pub/pthreads-win32/dll-latest/dll/x86/>

## 2. CÀI ĐẶT

- Cài đặt file mosquito vừa download ở bên trên.
- Copy thư viện pthreadVC2.dll vừa tải, 2 thư viện libeay32.dll ở đường link <http://www.dlldownloader.com/libeay32-dll/> (chọn win phù hợp và tải) và thư viện ssleay32.dll ở đường link <https://www.dll-files.com/ssleay32.dll.html> sau đó copy hết vào thư mục vừa cài đặt OpenSSL.



- Cài đặt lại mosquito (vẫn sử dụng file đã tải về).
- ### 3. KIỂM TRA HOẠT ĐỘNG CỦA MOSQUITO
- Mở cửa sổ command window: gõ vào dòng lệnh netstat -an
  - Kiểm tra kết quả, nếu kết quả được như hình dưới là đã thành công.



Có thể tham khảo video từ Youtube để cài đặt theo đường link:

<https://www.youtube.com/watch?v=daDuAZBgSDc>

## II. DOWNLOAD VÀ CÀI ĐẶT MỘT SỐ PHẦN MỀM HỖ TRỢ KHÁC

### 1. JAVA

Java là một ngôn ngữ lập trình và là một Platform. Xét về ngôn ngữ lập trình, Java là một ngôn ngữ lập trình có tính bảo mật cao, hướng đối tượng, bậc cao và mạnh mẽ. Bất cứ môi trường phần cứng hoặc phần mềm nào mà trong đó một chương trình chạy thì được biết đến như là một Platform. Java được gọi là Platform.

Đối với OpenHAB thì cần có cài đặt java 1.7 hoặc cao hơn để làm việc.

Tiến hành download Java từ địa chỉ <https://java.com/en/> và tiến hành cài đặt.

### 2. OPENHAB CORE RUNTIME

- Đây là chương trình chính để chạy openhab, hiển thị các trạng thái,...

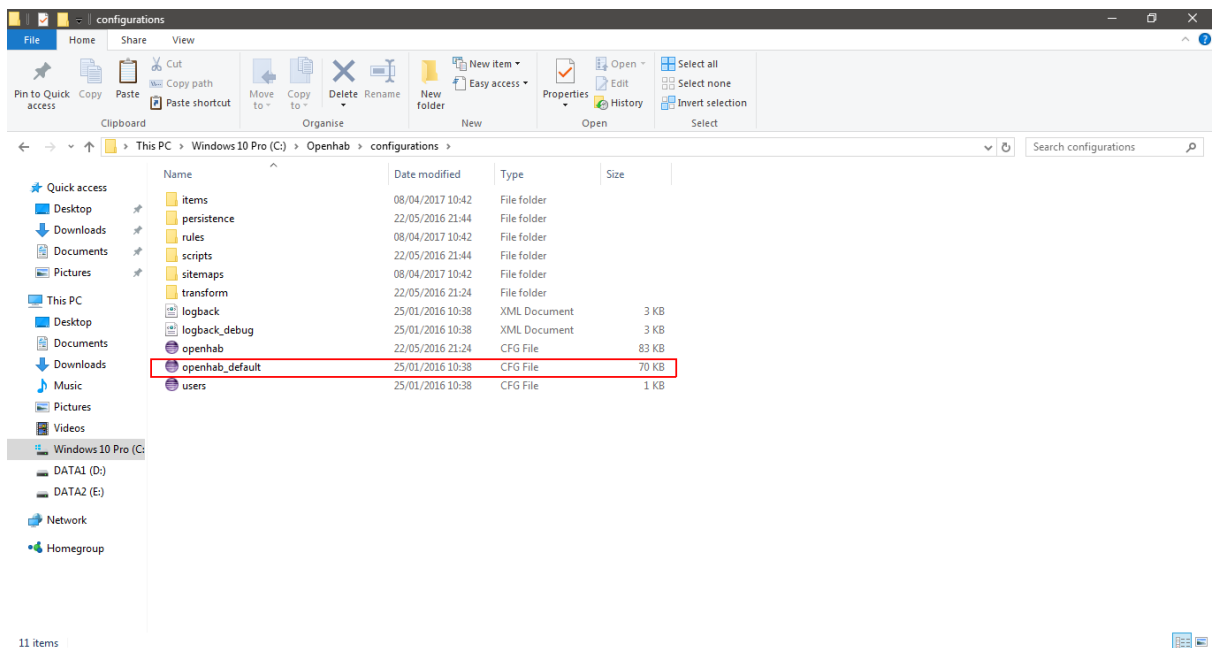
### 3. OPENHAB DESIGNER

- Đây là phần mềm dùng để thiết kế giao diện người dùng, các cách thức truyền nhận data, quản lý các phần tử truyền nhận, các rule chạy cho hệ thống mà người dùng muốn thiết kế.

### 4. CÁC RÀNG BUỘC

## III. THIẾT LẬP OPENHAB BẰNG OPENHAB DESIGNER

Tạo ra 1 file openhab.cfg có định dạng tương tự như openhab\_default.cfg trong thư mục configurations của Openhab.



Tiến hành khởi động openHAB-Designer → tìm và mở mục configuration của openhab.

## 1. CẤU HÌNH MQTT BINDING

- Tạo ra file config cho OpenHab từ file config\_default.cfg và sửa lại:

```
sudo cp /opt/openhab/configurations/openhab_default.cfg/opt/openhab/configurations/openhab.cfg
```

```
sudo nano /opt/openhab/configurations/openhab.cfg
```

- Trong cửa sổ editor, scroll xuống phần MQTT Transport section (trong mục Transport Configurations), bỏ qua phần MQTT Persistence. Tìm đến dòng <broker>.url và <broker>.retain, bỏ dấu # ở đầu mỗi dòng và sửa thành:

```
mqtt:mymosquitto.url=tcp://localhost:1883
```

```
mqtt:mymosquitto.retain=true
```

- Thoát và sao lưu lại.

## 2. CẤU HÌNH CÁC THIẾT BỊ (ITEMS) VÀ GIAO DIỆN (SITEMAP) CHO OPENHAB

### a. Demo.items

- Items file chứa danh sách các thiết bị mà OpenHAB điều khiển và theo dõi, ở đây bao gồm 3 đèn báo hiệu và một cảm biến nhiệt độ và độ ẩm.

- Ta cần phải tạo một file demo.items trong thư mục của OpenHAB

- Tiếp đó, chúng ta sửa các dòng lệnh thiết lập item trong file demo.items

### b. Default.sitemap

- File này dùng để thiết lập cấu hình dành cho giao diện OpenHAB, có thể nói đây là thứ tương tác chính giữa người sử dụng và sản phẩm, vì vậy cần phải được thiết kế hợp lý

### c. Demo.rules

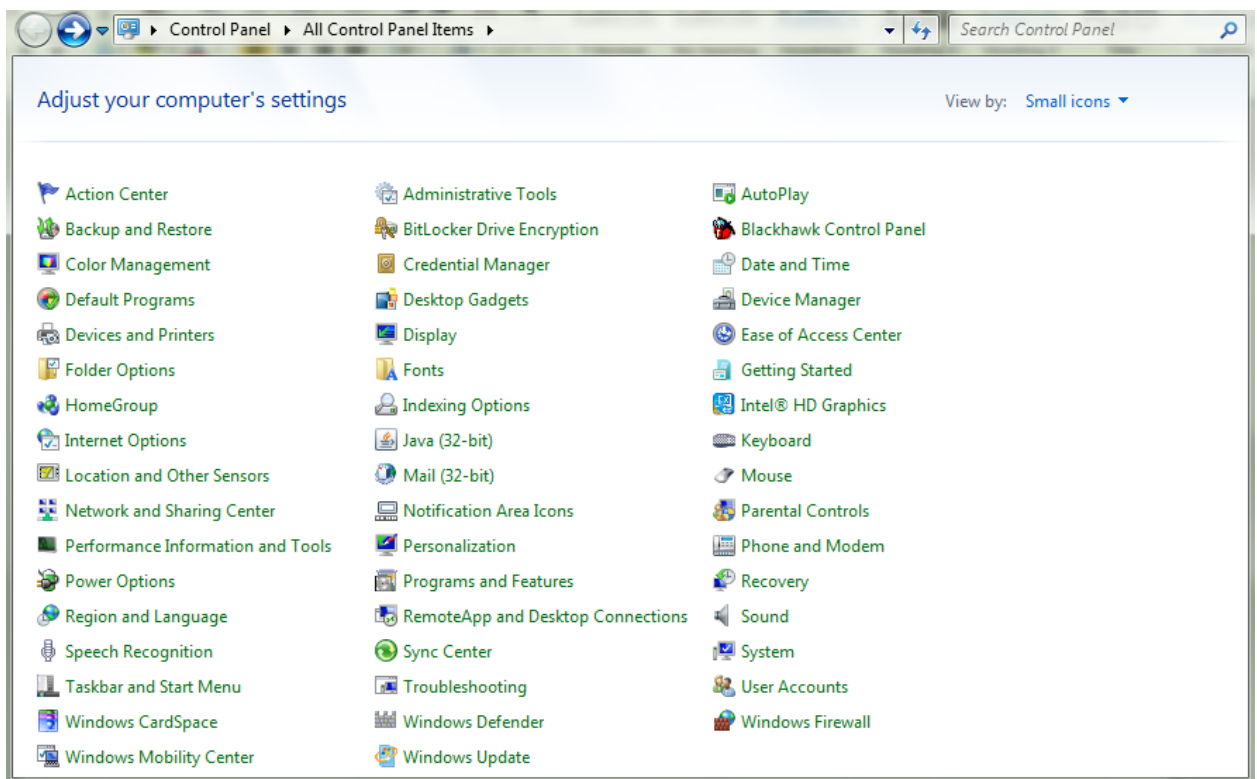
- Thiết lập các rule để thực hiện những chức năng của sản phẩm ví dụ như cảnh báo, tưới.... và tương tác thông qua giao diện

**Các bước thiết lập MQTT để demo sản phẩm:**

- 1) Kết nối phần cứng: theo như lý thuyết gắn cảm biến nhiệt độ, độ ẩm DHT11 vào các chân của arduino, dùng cáp RJ45 để kết nối giữa ethernet shield và laptop.
- 2) Mở phần mềm viết code cho arduino, kiểm tra code, verify sau đó upload lên board. Kiểm tra kết nối serial bằng cách vào tools->port. Sau đó mở “Serial Monitor”.
- 3) Mở MQTTlens, ứng dụng cài trên chorme.
- 4) Trong thư mục Openhab chạy start.bat, bên cạnh đó mở <http://192.168.100.17:8080/openhab.app> và xem kết quả hiển thị trên “Serial Monitor”. Điều chỉnh các chế độ hoạt động ở trong giao diện openhab tại IP trên và xem các gói tin chuyển bằng MQTTlens.

\*Nếu sau khi kết nối hết ở trên mà trong “Serial Monitor” vẫn hiện

**Failed to connect** thì tắt Windows Firewall trong controlpanel

**IV. ĐĂNG KÍ CÁC MẪU TIN TỪ BROKER**

- Sử dụng MQTTlens làm công cụ để quan sát các giá trị được gửi từ webserver xuống arduino hay từ arduino lên webserver

- Tiến hành cài đặt MQTTlens trên trình duyệt Chrom
- Sau khi cài đặt tiến hành chạy MQTTlens
- Chọn add a connection và cài đặt như hình với
  - Connection name và client ID là tùy ý
  - Địa chỉ là địa chỉ IP máy tính
- Sau đó lưu lại


The screenshot shows the 'Add a new Connection' window in MQTTlens. It has a title bar with a close button. Below the title bar is a tabbed interface with 'Connections' selected. The 'localbroker' connection is highlighted in green. The 'Connection Details' section includes: 'Connection name' (localbroker), 'Connection color scheme' (green bar), 'Hostname' (tcp:// 192.168.100.17), 'Port' (1883), 'Client ID' (lens\_a4iOrflqmA4p39os2LOFEiTM1q0), 'Session' (Clean Session checked), 'Automatic Connection' (checked), 'Keep Alive' (120 seconds), and 'Credentials' (Username and Password fields). A 'Generate a random ID' button is located next to the Client ID field.

→ Thành công khi nhận được biểu tượng như hình dưới


- ❖ Bắt đầu đăng kí các mẫu Topic sử dụng trong bài thiết kế này:
  - localnet/openHAB/devices/# (Để đăng kí nhận những giá trị điều khiển Led từ trang openHAB)
  - openhab/himitsu/humidity (Để đăng kí nhận những giá trị độ ẩm từ arduino gửi lên)
  - openhab/himitsu/temperature (Để đăng kí nhận những giá trị nhiệt độ từ arduino gửi lên)

Subscriptions


---

Topic: "localnet/openHAB/devices/#" Showing the last 1 messages — +  Messages: 0/2 ^


#	Time	Topic	QoS
1	12:22:02	localnet/openHAB/devices/led1	0

Message: OFF 

---

Topic: "openhab/himitsu/humidity" Showing the last 1 messages — +  Messages: 0/721 ^

#	Time	Topic	QoS
720	12:44:21	openhab/himitsu/humidity	0

Message: 76 

---

Topic: "openhab/himitsu/temperature" Showing the last 1 messages — +  Messages: 0/721 ^

#	Time	Topic	QoS
720	12:44:21	openhab/himitsu/temperature	0

Message: 29 