

**INDUSTRIAL UNIVERSITY OF HO CHI MINH CITY
FACULTY OF INFORMATION TECHNOLOGY**



**LE BA CHUAN
LE ANH TU**

**CLOUD IOT, BIGQUERY, DATASTUDIO
AND SMART HOME MODEL APPLICATION**

Major: Information Technology

Supervisor: M.Sc. Nguyen Thanh Thai

HO CHI MINH CITY, 2020

ABSTRACT

Title: Cloud IoT, BigQuery, DataStudio and Smarthome Model Application.

Remote device control is increasingly developing in scientific technology. Today's devices are almost automatic, typically smart home, this is a popular system and the world is developing.

This project presents the design, implementation to build a smart home model that can connect to the internet to exchange data between devices and the Cloud IoT.

The organization system of project perform a completely of google cloud platform which include end devices, each of them can connect with gateways in home as light, fan or sensor temperature, humidity. Use gateways on Google Cloud IoT Core can collect data from network or receiving data from outside to managing devices in house.

In this way, everyone can control all devices in smart home by Internet or smart phone application wherever you want, you just need to connect to the Internet to control, manage and monitor devices in the house.

Project described the integration: Smart Home, IoT, and Cloud Computing. To orchestrate and timely manage the vast data flow in an efficient and balanced way, utilizing the strengths of each component. Proposes a central real-time event handling application for in-house device management.

LỜI CẢM ƠN

Trong quá trình thực hiện Đồ án tốt nghiệp, chúng em xin gửi lời cảm ơn đến các quý thầy cô trong Khoa Công Nghệ Thông Tin đã tận tình dạy bảo, truyền đạt cho chúng em những kiến thức và kinh nghiệm. Đó là động lực rất lớn giúp chúng em hoàn thành tốt Khóa luận tốt nghiệp này.

Đặc biệt, chúng em xin gửi lời cảm ơn chân thành đến ThS. Nguyễn Thành Thái. Người đã trực tiếp hướng dẫn và tận tình giúp đỡ tạo điều kiện để hoàn thành tốt đề tài. Là người thầy nhiệt tình, tận tâm và hướng dẫn bọn em trong quá trình thực hiện đề tài, luôn đưa ra những ý kiến, nhận xét đúng đắn để chúng em hoàn thành được đề tài tốt nghiệp kịp thời và hoàn thiện nhất có thể.

Sau cùng, tuy có nhiều nỗ lực, nhưng do thời gian thực hiện đề tài không nhiều, kiến thức và kinh nghiệm còn hạn chế nên đồ án tốt nghiệp chúng em không tránh khỏi những thiếu sót. Do đó, chúng em rất mong nhận được những ý kiến của quý thầy cô, bạn bè để chúng em hoàn thiện tốt hơn.

Chúng em xin chân thành cảm ơn !

Tp. HCM, ngày..... tháng..... năm 2020

Người thực hiện đề tài

NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Điểm: (Bằng chữ:)

Tp. HCM, ngày..... tháng..... năm 2020

Giảng viên hướng dẫn

NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIÁO VIÊN PHẢN BIỆN 1

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Điểm: (Bằng chữ:)

Tp. HCM, ngày..... tháng..... năm 2020

Giảng viên phản biện 1

NHẬN XÉT VÀ PHẢN BIỆN CỦA GIÁO VIÊN PHẢN BIỆN 2

This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Điểm: (Bằng chữ:)

Tp. HCM, ngày..... tháng..... năm 2020

Giảng viên phản biện 2

MỤC LỤC

Chương 1. TỔNG QUAN	1
1.1. ĐẶT VẤN ĐỀ	1
1.2. MỤC ĐÍCH NGHIÊN CỨU	2
1.3. NHIỆM VỤ NGHIÊN CỨU	2
1.4. PHƯƠNG PHÁP NGHIÊN CỨU	3
Chương 2. CƠ SỞ LÝ THUYẾT	4
2.1. GIỚI THIỆU VỀ IOT	4
2.1.1. Định nghĩa	4
2.1.2. Các thành phần cơ bản của một nền tảng IOT	5
2.1.3. Ứng dụng của IOT	6
2.2. GIỚI THIỆU VỀ NHÀ THÔNG MINH ỨNG DỤNG IOT	8
2.2.1. Định nghĩa về nhà thông minh	8
2.2.2. Cách hoạt động của hệ thống nhà thông minh	8
2.3. GIỚI THIỆU CLOUD	9
2.3.1. Định nghĩa	9
2.3.2. Lợi ích Cloud	10
2.3.3. Các mô hình dịch vụ	10
2.3.4. Vai trò và lợi ích của Cloud với IOT	11
2.4. GIỚI THIỆU VỀ GOOGLE CLOUD IOT CORE	12
2.4.1. Khái niệm	12
2.4.2. Các thành phần Google Cloud IoT Core	13
2.4.3. Cách thức bảo mật, kết nối thiết bị với GCIC	14
2.4.4. Quan hệ giữa thiết bị IoT với GCIC	15
2.5. GIỚI THIỆU VỀ GOOGLE BIG QUERY, DATASTUDIO	16
2.5.1. BigQuery	16
2.5.2. DataStudio	17
2.6. GIỚI THIỆU VỀ GOOGLE FIREBASE	19
2.6.1. Khái niệm về Firebase	19
2.6.2. Dịch vụ cơ bản Firebase	19

2.6.3. Lợi ích firebase cung cấp	20
2.7. GIỚI THIỆU MỘT SỐ MODULE SỬ DỤNG TRONG MÔ HÌNH.	21
2.7.1. Module Raspberry Pi 3	21
2.7.2. Module Arduino UNO R3	23
2.7.3. Module cảm biến nhiệt độ, độ ẩm DHT11	24
2.7.4. Module cảm biến khí ga MQ 135	26
2.7.5. Module cảm biến thân nhiệt chuyển động PIR HS-SR501	27
2.7.6. Module Relay Opto Cách Ly 5VDC.....	28
Chương 3. TÍNH TOÁN THIẾT KẾ, XÂY DỰNG VÀ THI CÔNG HỆ THỐNG	30
3.1. THIẾT KẾ VÀ XÂY DỰNG.....	30
3.1.1. Mô hình tổng quan	30
3.1.2. Sơ đồ khối hệ thống	31
3.1.3. Thiết kế mô hình	32
3.1.4. Thiết kế mạch điều khiển với hệ thống cảm biến	33
3.1.5. Thiết kế mạch điều khiển với hệ thống điều khiển.....	34
3.1.6. Thiết kế website	35
3.2. THI CÔNG HỆ THỐNG	38
3.2.1. Thiết lập Raspberry	38
3.2.2. Thiết lập Google Cloud IoT Core	39
3.2.3. Thiết lập Google Firebase	40
3.2.4. Triển khai dự án lên Hosting	41
3.2.5. Triển khai các hàm chức năng Functions	44
3.2.6. Thiết lập BigQuery và DataStudio.....	45
Chương 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	50
4.1. KẾT QUẢ ĐẠT ĐƯỢC	50
4.2. HẠN CHẾ	57
4.3. HƯỚNG PHÁT TRIỂN.....	57
TÀI LIỆU THAM KHẢO.....	58

MỤC LỤC HÌNH ẢNH

Hình 2. 1: Định nghĩa Internet of Things.....	4
Hình 2. 2: Cấu trúc cơ bản của một hệ thống IoT.....	5
Hình 2. 3: Ứng dụng của IoT trong xã hội.....	7
Hình 2. 4: Đồng hồ thông minh là sản phẩm cho sự phát triển IoT.....	7
Hình 2. 5: Nhà thông minh ứng dụng IoT.....	8
Hình 2. 6: Một số dịch vụ Cloud Computing.....	9
Hình 2. 7: Google Cloud IoT Core với một số dịch vụ khác của Google Cloud.....	12
Hình 2. 8: Cấu trúc chính của Google Cloud IoT Core	13
Hình 2. 9: Cơ chế xác thực giữa Device với Cloud	14
Hình 2. 10: Sự tương tác giữa các Device với Cloud IoT qua giao thức MQTT	15
Hình 2. 11: Một số dịch vụ của BigQuery	16
Hình 2. 12: Google DataStudio với các dịch vụ hỗ trợ.....	18
Hình 2. 13: Trao đổi dữ liệu giữa Firebase và SDK Device	19
Hình 2. 14: Module Raspberry Pi 3 B+	21
Hình 2. 15: Phương thức truyền dữ liệu qua giao tiếp UART	22
Hình 2. 16: Module Arduino UNO	23
Hình 2. 17: Cảm biến nhiệt độ độ ẩm DHT11	25
Hình 2. 18: Cảm biến khí Gas MQ 135	26
Hình 2. 19: Cảm biến than nhiệt chuyển động.....	27
Hình 2. 20: Relay opto cách ly mức cao thấp 5VDC.....	28
Hình 3. 1: Mô hình tổng quan hệ thống	30
Hình 3. 2: Sơ đồ khối hoạt động của hệ thống.....	31
Hình 3. 3: Bản vẽ mô hình thiết kế	32
Hình 3. 4: Mô hình triển khai với hệ thống cảm biến	33
Hình 3. 5: Sơ đồ kết nối hệ thống cảm biến.....	34
Hình 3. 6: Mô hình triển khai hệ thống điều khiển	34
Hình 3. 7: Sơ đồ kết nối hệ thống điều khiển.....	35
Hình 3. 8: Giao diện đăng nhập	36

Hình 3. 9: Giao diện điều khiển	36
Hình 3. 10: Giao diện hiển thị dữ liệu nhiệt độ, độ ẩm.....	37
Hình 3. 11: Biểu đồ dữ liệu theo thời gian thực.....	37
Hình 3. 12: Kiểm tra version node js	41
Hình 3. 13: Khởi tạo dự án.....	42
Hình 3. 14: Chọn project cần triển khai	42
Hình 3. 15: Lựa chọn ngôn ngữ để viết các functions	42
Hình 3. 16: Cài đặt các gói thư viện cần thiết.....	43
Hình 3. 17: Cấu hình hosting	43
Hình 3. 18: Cấu trúc thư mục của project	43
Hình 3. 19: Triển khai dự án lên hosting	44
Hình 3. 20: Cấu trúc thư mục functions.....	44
Hình 3. 21: Functions đã triển khai thành công trên firebase	45
Hình 3. 22: Tạo Dataset	45
Hình 3. 23: Tạo bảng mới để lưu trữ dữ liệu	46
Hình 3. 24: Tạo bảng để lưu trữ dữ liệu.....	47
Hình 3. 25: Explore dữ liệu thông qua Goggle DataStudio	48
Hình 3. 26: Biểu đồ dữ liệu của DataStudio	49
Hình 3. 27: Một số phương thức tính toán dữ liệu DataStudio hỗ trợ	49
Hình 4. 1: Mô hình nhà thông minh.....	50
Hình 4. 2: Giao diện website.....	51
Hình 4. 3: Biểu đồ dữ liệu real-time	51
Hình 4. 4: Bảng dữ liệu thu thập nhiệt độ, độ ẩm	52
Hình 4. 5: Dữ liệu của đèn led trên Pub/Sub.....	52
Hình 4. 6: Dữ liệu nhiệt độ, độ ẩm trên Pub/Sub.....	53
Hình 4. 7: Dữ liệu đèn led trên Firestore	53
Hình 4. 8: Dữ liệu trên Firebase Real-time	54
Hình 4. 9: Dữ liệu nhiệt độ, độ ẩm trên BigQuery.....	54
Hình 4. 10: Dữ liệu khí gas trên BigQuery	55

Hình 4. 11: Truy vấn dữ liệu khí ga trên BigQuery	55
Hình 4. 12: Biểu đồ dữ liệu trong Google DataStudio.....	56
Hình 4. 13: Điều khiển nhà thông minh.....	56

DANH MỤC CÁC TỪ VIẾT TẮT

STT	Từ viết tắt	Cụm từ đầy đủ
1	AC	Alternating Current
2	API	Application Programming Interface
3	BQ	BigQuery
4	DC	Direct Current
5	GCP	Google Cloud Platform
6	GCIC	Google Cloud IoT Core
7	HTTP	HyperText Transfer Protocol
8	IOT	Internet of things
9	JWT	JSON Web Token
10	MQTT	Message Queuing Telemetry Transport
11	NFC	Near-Field Communications
12	RFID	Radio Frequency Identification
13	QR	Quick Response
14	SDK	Software Development Kit
15	TLS	Transport Layer Security
16	USB	Universal Serial Bus
17	UART	Universal Asynchronous Receiver Transmitter

TÓM TẮT NỘI DUNG ĐỀ TÀI

Hiện nay với sự phát triển rất mạnh mẽ về khoa học và công nghệ, các hệ thống thông minh ứng dụng IoT tuy không mới nhưng rất thịnh hành. IoT đã và đang làm thay đổi cả thế giới, tại Việt Nam trong khoảng 4 năm trở lại đây, IoT đã từng bước được ứng dụng trong một số lĩnh vực của cuộc sống, tuy nhiên chỉ dừng lại ở mức rời rạc, chưa đồng bộ. Những lợi ích và tiềm năng phát triển của IoT là rất lớn, ở đâu có kết nối Internet ở đó đều có khả năng xuất hiện các thiết bị IoT mang đến giá trị thông qua việc truyền tải và trao đổi thông tin, dữ liệu.

Diễn hình trong số ứng dụng IoT là ứng dụng xây dựng hệ thống nhà thông minh – Smart Home. Hầu hết các hệ thống nhà thông minh được điều khiển bởi điện thoại thông minh và vi điều khiển. Một ứng dụng điện thoại thông minh được sử dụng để kiểm soát và giám sát các chức năng trong hệ thống Smart Home bằng kết nối Internet.

Bằng cách này, mọi người đều có thể quản lý, giám sát và điều khiển tất cả các thiết bị trong nhà thông qua ứng dụng Web hoặc ứng dụng điện thoại thông minh bất cứ nơi nào chỉ cần có kết nối với Internet.

Chính vì vậy nhóm em đã nghiên cứu và tìm hiểu để triển khai mô hình nhà thông minh tích hợp các dịch vụ IoT và điện toán đám mây với nhau, bằng cách sử dụng Cloud IoT để kết nối giao tiếp giữa các thiết bị với hệ thống Cloud, giúp các thiết bị tương tác với nhau để có thể dễ dàng truy cập, xử lý và trao đổi dữ liệu.

Trong nội dung nghiên cứu của đề tài nhóm em tập trung triển khai xây dựng mô hình hệ thống nhà thông minh kết hợp với hệ thống Cloud IoT. Qua đó chúng ta có thể điều khiển, giám sát các thiết bị từ xa bất kỳ ở đâu thông qua Internet, ngoài ra còn lưu trữ, xử lý và phân tích dữ liệu từ các hệ thống cảm biến.

Chương 1. TỔNG QUAN

1.1. ĐẶT VẤN ĐỀ

Hiện nay trên thị trường, công nghệ điều khiển từ xa vẫn còn khá mới mẻ, chưa được phổ biến và có giá thành rất cao. Vì vậy việc nghiên cứu chế tạo ra mô hình “Ngôi nhà thông minh - Smart Home” sẽ giúp chúng em tìm hiểu sâu hơn về các công nghệ mới, qua đó tìm ra các phương án giảm chi phí, tiết kiệm năng lượng trong các hệ thống Smart Home thực tế.

Bên cạnh những yêu cầu liên quan đến lĩnh vực trong đời sống, việc thực hiện đề tài nhằm chế tạo ra một mô hình phục vụ cho học tập, nghiên cứu. Chúng em muốn có thể sử dụng mô hình để thực nghiệm, chứng minh các kiến thức đã tìm hiểu và nghiên cứu.

Thêm vào đó, ngày nay các thiết bị điện thoại di động ngày càng được phổ biến trong cuộc sống. Cùng với đó là nhu cầu cuộc sống ngày càng tăng. Từ đó đã hình thành, nảy sinh một ý tưởng về việc điều khiển các thiết bị điện một cách tự động thông qua điện thoại di động có kết nối Internet.

Diễn hình của một hệ thống điều khiển thiết bị trực tiếp bằng tay hoặc điều khiển bằng giọng nói thông qua Wifi gồm có các thiết bị đơn giản như bóng đèn, nhiệt độ, độ ẩm, hệ thống đóng mở cửa tự động... Các thiết bị sẽ kết nối và giao tiếp với nhau thông qua hệ thống IoT kết hợp với Cloud.

Bình thường, các thiết bị trong nhà có thể được điều khiển từ xa thông qua Internet. Chẳng hạn như việc tắt đèn khi chủ nhà rời đi mà quên chưa tắt. Hoặc là việc tự động mở cửa khi nhà có khách chỉ bằng một vài thao tác trên điện thoại di động. Còn khi có sự cố mang tính khẩn cấp như hỏa hoạn, thì sẽ có còi báo động, lúc này hệ thống sẽ tự động phát hiện ra hỏa hoạn nhờ vào các cảm biến khí gas.

Từ những yêu cầu thực tế, những đòi hỏi ngày càng cao của cuộc sống, nên chúng em đã chọn đề tài "Cloud Iot, Bigquery with DataStudio và ứng dụng nhà thông minh" để đáp ứng được nhu cầu ngày càng cao của con người. Đề tài của

nhóm tập trung xây dựng và triển khai một mô hình nhà thông minh kết hợp giữa các dịch vụ IoT với điện toán đám mây để có thể kết nối các thiết bị với nhau. Từ đó chúng ta có thể quản lý, giám sát trạng thái, điều khiển và bật tắt của thiết bị.

1.2. MỤC ĐÍCH NGHIÊN CỨU

Tìm hiểu và thực hiện điều khiển các thiết bị trong nhà như đèn, quạt,... thông qua mạng Internet. Cụ thể là tìm hiểu raspberry để điều khiển thiết bị thông qua mạng Internet.

Tìm hiểu và tiến hành xây dựng cơ sở dữ liệu, kết nối và trao đổi dữ liệu giữa các thiết bị và Cloud.

Xây dựng ứng dụng website để điều khiển và giám sát thiết bị trong nhà.

Thiết kế và thi công mô hình để tạo ra một hệ thống "Ngôi nhà thông minh Smart Home" hoàn chỉnh.

1.3. NHIỆM VỤ NGHIÊN CỨU

Tìm hiểu cách thức hoạt động của các hệ thống nhà thông minh. Sự tương tác trao đổi dữ liệu giữa các thiết bị trên hệ thống IoT.

Tìm hiểu về vi xử lý Raspberry, Board mạch Arduino và Module cảm biến.

Thiết kế và xây dựng ứng dụng website để điều khiển và giám sát thiết bị.

Thiết kế và thi công mô hình nhà có các chức năng thông minh cơ bản.

Viết chương trình điều khiển cho Raspberry và Arduino trao đổi và xử lý các dữ liệu từ cảm biến.

Thử nghiệm và điều chỉnh phần cứng cũng như chương trình để mô hình được tối ưu và hoàn thiện nhất có thể.

Viết báo cáo luận văn.

Tiến hành báo cáo đồ án tốt nghiệp.

1.4. PHƯƠNG PHÁP NGHIÊN CỨU

Hệ thống sử dụng Raspberry Pi 3 B+ để kết nối Internet làm vi điều khiển chính của hệ thống.

Sử dụng các cảm biến thông dụng như: cảm biến nhiệt độ độ ẩm DHT11, cảm biến chuyển động PIR, cảm biến khí ga MQ2,... để thu thập các thông tin.

Sử dụng hệ thống chuông, đèn báo động khi phát hiện sự cố cháy nổ.

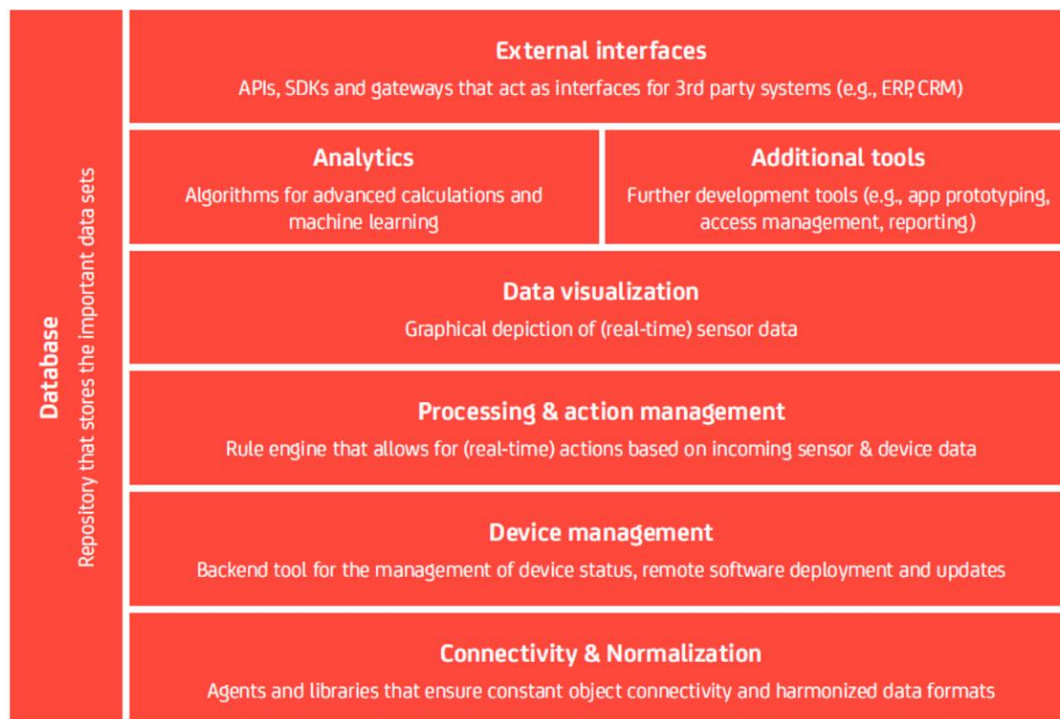
Sử dụng hệ thống đèn led, relay, để điều khiển tăng tính ứng dụng. Không chỉ áp dụng cho các thiết bị trong gia đình mà còn phục vụ cho lĩnh vực công nghiệp, sản xuất.

Sử dụng phần mềm Visual Studio Code để lập trình ứng viết code xử lý, ứng dụng Web và Arduino IDE để lập trình phía Arduino Uno.

Ngôn ngữ sử dụng chính: C++, Python, Html, Css, Javascript,...

2.1.2. Các thành phần cơ bản của một nền tảng IOT

Kiến trúc chung của mô hình IoT phổ biến nhất hiện nay, được cấu thành bởi các thành phần sau:



Hình 2. 2: Cấu trúc cơ bản của một hệ thống IoT

Connectivity and Normalization: Thành phần này đưa các giao thức khác nhau và các định dạng dữ liệu khác nhau vào một giao diện của Phần mềm, đảm bảo truyền dữ liệu và tương tác chính xác với tất cả các thiết bị.

Device Management: Đây là thành phần đảm bảo kết nối các thiết bị hoạt động bình thường, chạy liên mạch các bản vá và cập nhật cho phần mềm và ứng dụng chạy trên thiết bị hoặc các cổng cạnh

Database: Đây là thành phần được coi quan trọng của một nền tảng là nơi lưu trữ dữ liệu thiết bị có thể mở rộng đưa các yêu cầu đối với cơ sở dữ liệu dựa trên đám mây Thành phần này phải có khả năng mở rộng về khối lượng dữ liệu, tính đa dạng, vận tốc và tính xác thực.

Processing and Action Management: làm cho dữ liệu trở nên sống động với các trình kích hoạt sự kiện dựa trên quy tắc cho phép thực thi các hành động thông minh dựa trên dữ liệu cảm biến cụ thể.

Analytics: Thành phần này có chức năng thực hiện hàng loạt các phân tích phức tạp từ việc phân cụm dữ liệu cơ bản và khả năng tự học để tự phân tích, dự đoán, trích xuất những dữ liệu giá trị nhất trong luồng dữ liệu IoT.

Data Visualization: cho phép con người nhìn thấy các mẫu và quan sát xu hướng từ bảng điều khiển trực quan hóa, nơi dữ liệu được mô tả một cách sinh động thông qua các biểu đồ đường, xếp chồng, hoặc hình tròn,...

Additional Tools: cho phép các nhà phát triển IoT nguyên mẫu, thử nghiệm và tiếp thị trường hợp sử dụng IoT tạo các ứng dụng hệ sinh thái nền tảng để trực quan hóa, quản lý và kiểm soát các thiết bị được kết nối.

External Interfaces: tích hợp với các hệ thống của bên thứ ba và phần còn lại của hệ thống công nghệ thông tin qua các giao diện lập trình ứng dụng (API), các bộ phát triển phần mềm (SDK) và các gateways.

2.1.3. Ứng dụng của IOT

IoT được ứng dụng trong hầu hết các lĩnh vực:

Thành phố thông minh bao gồm giám sát thông minh, vận chuyển tự động, hệ thống quản lý năng lượng thông minh hơn, phân phối nước, an ninh đô thị và giám sát môi trường...

Internet công nghiệp là hỗ trợ kỹ thuật công nghiệp với các cảm biến, phần mềm lớn để tạo ra những cỗ máy vô cùng thông minh. Máy móc sẽ có tính chính xác và nhất quán hơn con người trong giao tiếp thông qua dữ liệu. Từ những dữ liệu thu thập được giúp các công ty, nhà quản lý giải quyết các vấn đề sớm hơn, đạt hiệu quả cao hơn...



Hình 2. 3: Ứng dụng của IoT trong xã hội

Các thiết bị đeo thông minh như các loại kính, ba lô, vòng tay được cài đặt cảm biến thu thập thông tin người dùng như huyết áp, bước đi bộ mỗi ngày,...



Hình 2. 4: Đồng hồ thông minh là sản phẩm cho sự phát triển IoT

Tác động của IoT rất đa dạng, trên các lĩnh vực: quản lý hạ tầng, y tế, xây dựng và tự động hóa, giao thông....

2.2. GIỚI THIỆU VỀ NHÀ THÔNG MINH ỨNG DỤNG IOT

2.2.1. Định nghĩa về nhà thông minh

Khái niệm ngôi nhà thông minh vốn khá phổ biến ở các nước công nghệ phát triển và được du nhập qua các quốc gia có cơ sở hạ tầng công nghệ thông tin đang phát triển. Ngôi nhà thông minh hay còn gọi là ngôi nhà số là một giải pháp điều khiển tích hợp cho các căn hộ cao cấp, tích hợp các thiết bị điện tử, nghe nhìn, truyền thông thành một hệ hoàn chỉnh và thống nhất, có thể tự vận hành tất cả các hệ thống theo sự điều khiển từ xa của người dùng. Các hệ thống như đèn chiếu sáng, cửa tự động... đều được phối hợp vận hành thành một hệ đồng nhất.



Hình 2. 5: Nhà thông minh ứng dụng IoT

2.2.2. Cách hoạt động của hệ thống nhà thông minh

Nhà thông minh không phải là các thiết bị và ứng dụng riêng biệt, chúng làm việc cùng nhau để tạo ra một mạng lưới có thể điều khiển từ xa. Chủ nhà điều khiển tất cả các thiết bị được kiểm soát bởi một bộ điều khiển tự động, được gọi là Smart Home Hub.

Smart Home Hub là một thiết bị phần cứng hoạt động như điểm trung tâm của hệ thống Smart Home có thể cảm nhận, xử lý dữ liệu và truyền thông không dây. Nó kết hợp tất cả các ứng dụng riêng lẻ vào một ứng dụng duy nhất có thể

được kiểm soát từ xa bởi chủ nhà. Ví dụ các hệ thống Amazon Echo, Google Home, Insteon Hub Pro, Samsung SmartThings và Wink Hub.

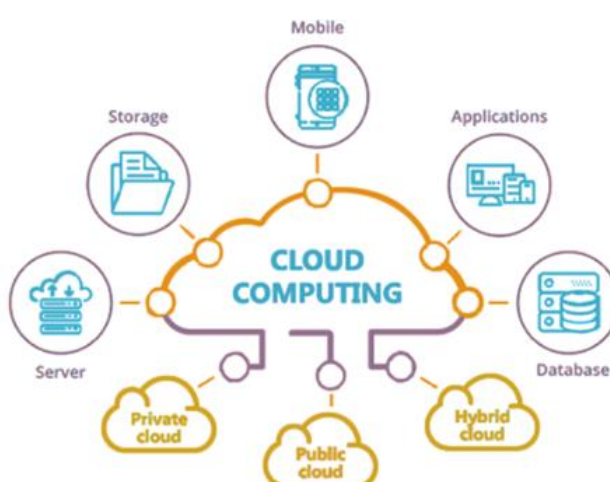
Một số hệ thống Smart Home được tạo ra từ đầu, ví dụ như sử dụng Raspberry Pi hoặc Intel®Edison.... Những sản phẩm khác có thể được mua như một bộ Smart Home Kit - còn được gọi là nền tảng Smart Home - có chứa các phần cần thiết để bắt đầu dự án tự động hóa nhà ở.

Trong các mô hình Smart Home đơn giản, các sự kiện có thể được tính giờ hoặc kích hoạt. Sự kiện tính giờ dựa theo đồng hồ, ví dụ như hạ thấp rèm vào lúc 6 giờ chiều, sự kiện được kích hoạt dựa trên hành động trong hệ thống tự động; như khi điện thoại thông minh của chủ nhà đến gần cửa, khóa thông minh sẽ mở ra và đèn thông minh sẽ sáng lên.

2.3. GIỚI THIỆU CLOUD

2.3.1. Định nghĩa

Điện toán đám mây là việc phân phối các tài nguyên Công nghệ thông tin theo nhu cầu qua Internet với chính sách thanh toán theo mức sử dụng. Thay vì mua, sở hữu và bảo trì các trung tâm dữ liệu và máy chủ vật lý, bạn có thể tiếp cận các dịch vụ công nghệ, như năng lượng điện toán, lưu trữ và cơ sở dữ liệu từ nhà cung cấp dịch vụ đám mây.



Hình 2. 6: Một số dịch vụ Cloud Computing

2.3.2. Lợi ích Cloud

Đám mây cho phép bạn dễ dàng tiếp cận nhiều công nghệ để bạn có thể đổi mới nhanh hơn và phát triển gần như mọi thứ mà bạn có thể. Bạn có thể nhanh chóng thu thập tài nguyên khi cần từ các dịch vụ cơ sở hạ tầng, như điện toán, lưu trữ, và cơ sở dữ liệu, đến Internet of Things, Machine Learning, kho dữ liệu và phân tích,...

Cho phép triển khai các dịch vụ công nghệ một cách nhanh chóng. Điều này cho phép tự do thử nghiệm, kiểm thử những ý tưởng, chức năng mới để nâng cao chất lượng dịch vụ, sản phẩm đáp ứng nhu cầu ngày càng cao của khách hàng.

Với điện toán đám mây, bạn không phải cung cấp tài nguyên quá mức để xử lý các hoạt động kinh doanh ở mức cao nhất trong tương lai. Thay vào đó, bạn cung cấp lượng tài nguyên mà thực sự cần. Có thể tăng hoặc giảm quy mô của các tài nguyên này ngay lập tức để tăng và giảm dung lượng khi nhu cầu kinh doanh của thay đổi.

Nền tảng đám mây cho phép thay chi phí vốn (trung tâm dữ liệu, máy chủ vật lý,...) bằng chi phí biến đổi và chỉ phải chi trả cho những tài nguyên mà bạn sử dụng.

Với đám mây, cho phép mở rộng các dịch vụ sang các khu vực địa lý mới và triển khai trên toàn cầu trong thời gian ngắn và nhanh chóng. Ví dụ: GOOGLE CLOUD có cơ sở hạ tầng trên toàn thế giới. Vì vậy chúng ta có thể triển khai ứng dụng ở nhiều địa điểm trên thực tế.

2.3.3. Các mô hình dịch vụ

Ba loại điện toán đám mây chính bao gồm Cơ sở hạ tầng dưới dạng dịch vụ (IaaS), Nền tảng dưới dạng dịch vụ (PaaS), Phần mềm dưới dạng dịch vụ (SaaS).

Mỗi loại điện toán đám mây cung cấp các cấp độ kiểm soát, tính linh hoạt và quản lý khác nhau để bạn có thể chọn bộ dịch vụ phù hợp với nhu cầu của hệ thống.

2.3.4. Vai trò và lợi ích của Cloud với IOT

2.3.4.1. Vai trò

Vai trò của điện toán đám mây trong IoT hoạt động như một phần của sự hợp tác giữa các thiết bị và Cloud và được sử dụng để lưu trữ dữ liệu IoT, cho phép truy cập dễ dàng khi cần thiết. Điện toán đám mây là một phương pháp di chuyển dễ dàng cho các gói dữ liệu lớn do IoT tạo ra thông qua Internet.

Điện toán đám mây cung cấp các công cụ và dịch vụ cần thiết để tạo các ứng dụng IoT. Cloud giúp đạt được hiệu quả, độ chính xác, tốc độ trong việc triển khai các ứng dụng IoT. Đám mây giúp phát triển ứng dụng IoT nhưng IoT không phải là điện toán đám mây.

Ví dụ, trong trường hợp một số lượng lớn thiết bị IoT được kết nối với mạng và sẽ tạo ra các byte dữ liệu khổng lồ và các tổ chức cần một hệ thống thích hợp để lưu trữ và truy xuất dữ liệu đó.

Nhiều nhà cung cấp dịch vụ đám mây đã xác định nhu cầu này và bắt đầu cung cấp dịch vụ cụ thể cho IoT cho các công ty để tạo ra các giải pháp IoT tốt hơn. Các công ty lớn như Microsoft, Amazon, IBM và SAP đã triển khai các thành phần liên quan đến IoT vào nền tảng đám mây của họ. Google có Google Cloud IoT, Microsoft có Azure IoT, Amazon AWS IoT, SAP Hana và IBM Cloud,... Điều này mở rộng chức năng xây dựng các ứng dụng, dịch vụ IoT trên đám mây.

2.3.4.2. Lợi ích

Có thể truy cập dữ liệu từ xa và dễ dàng để vẫn thực hiện các hành động trên thiết bị, ở bất kỳ đâu chỉ cần có kết nối Internet.

Sử dụng Cloud với IoT giúp tăng cường bảo mật, vì các bản cập nhật thường xuyên có thể được gửi và kiến thức về bất kỳ vi phạm nào trong cơ sở hạ tầng có thể được gỡ bỏ ngay lập tức.

Khả năng mở rộng cao, cho phép nhiều thiết bị có thể nối lại với nhau.

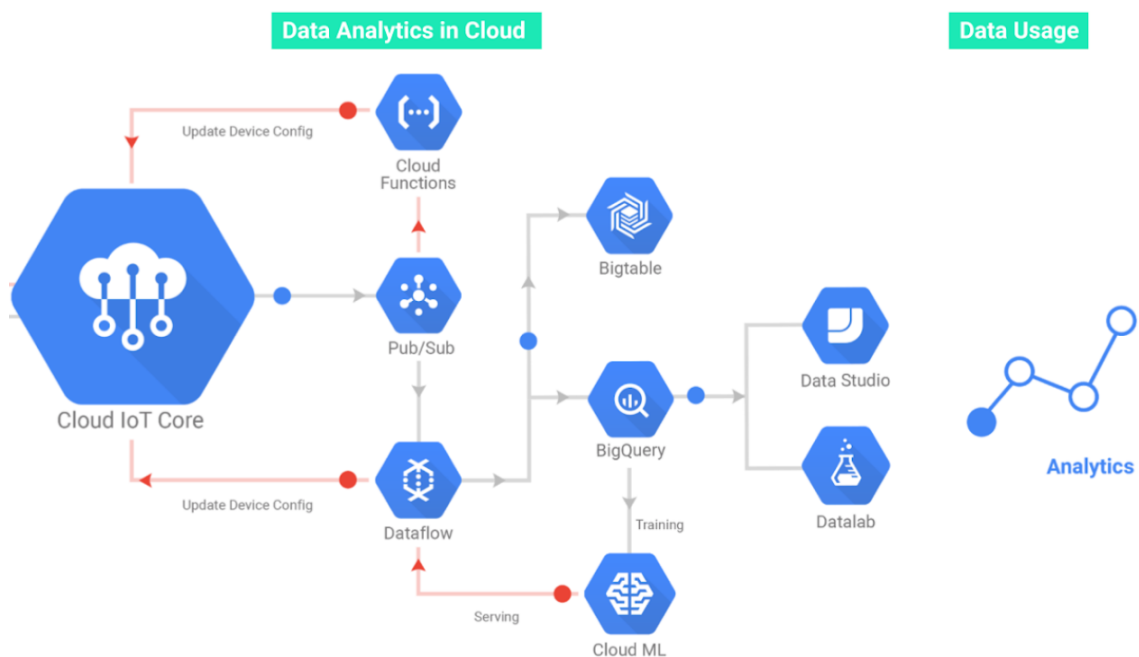
Nhìn chung, sử dụng Cloud kết hợp với IoT giúp chúng ta có thể giao tiếp, kết nối và truyền dữ liệu thành công giữa các thiết bị, theo cách nhanh chóng và hiệu quả nhất có thể.

2.4. GIỚI THIỆU VỀ GOOGLE CLOUD IOT CORE

2.4.1. Khái niệm

Cloud IoT Core là một dịch vụ nằm trong nền tảng Google Cloud Platform. Đây là một dịch vụ được quản lý hoàn toàn cho phép bạn kết nối, quản lý và nhập dữ liệu một cách dễ dàng và an toàn từ hàng triệu thiết bị phân tán trên toàn cầu.

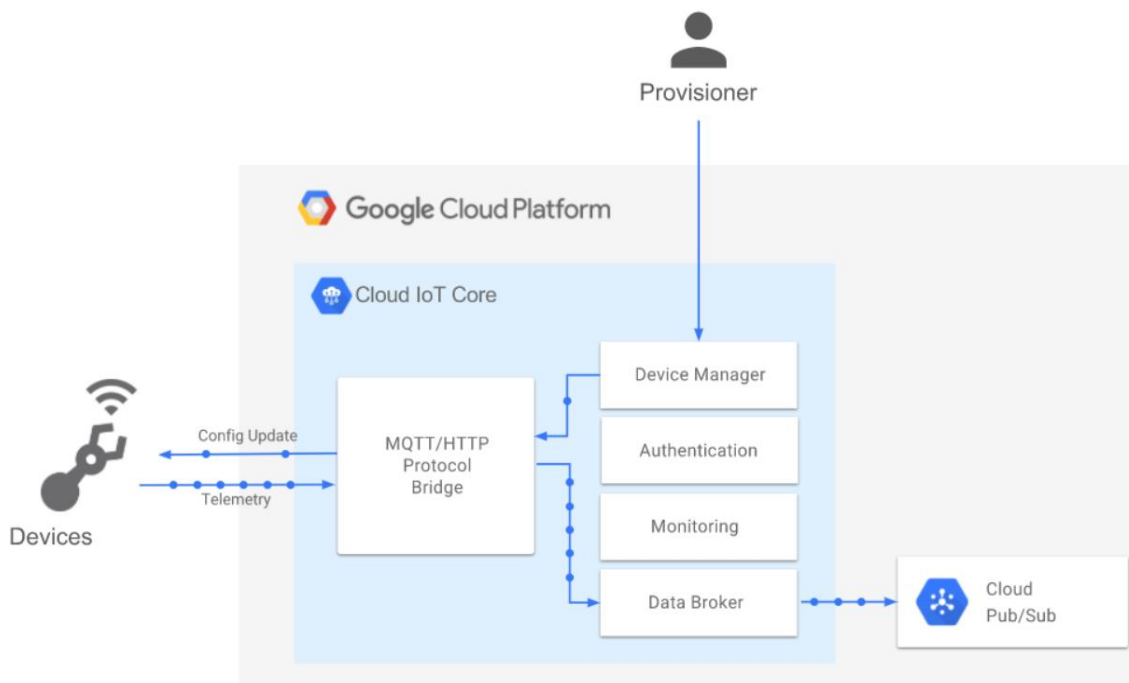
Cloud IoT Core, kết hợp với các dịch vụ khác trên nền tảng Cloud IoT (Google BigQuery, Google Data Studio,...), cung cấp một giải pháp hoàn chỉnh để thu thập, xử lý, phân tích và trực quan hóa dữ liệu IoT trong thời gian thực để hỗ trợ cải thiện hiệu quả hoạt động.



Hình 2. 7: Google Cloud IoT Core với một số dịch vụ khác

2.4.2. Các thành phần Google Cloud IoT Core

Các thành phần chính của Cloud IoT Core là trình quản lý thiết bị và cầu nối giao thức (Device Manager and Protocol Bridges).



Hình 2. 8: Cấu trúc chính của Google Cloud IoT Core

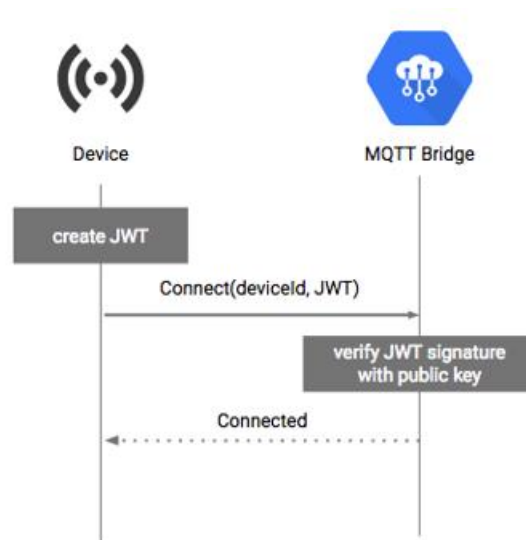
Device Manager: Trình quản lý thiết bị để đăng ký thiết bị với dịch vụ, có thể theo dõi và cấu hình các thiết bị IoT với GCIC. Các thiết bị kết nối tới Google Cloud IoT Core sẽ được xem như là một “device”. Nhóm các “device” lại với nhau, chúng ta sẽ có một “registry” là ô chứa đại diện cho cả các “device” đó. Việc khởi tạo và quản lý “device” với “registry” sẽ phụ thuộc vào cách tổ chức của chúng ta.

Protocol Bridges: Cầu nối giao thức (MQTT và HTTP) mà các thiết bị có thể sử dụng để kết nối với Google Cloud Platform. Đây là thành phần cung cấp các giao thức truyền thông để chúng ta có thể truyền nhận dữ liệu với GCIC, hiện GCIC hỗ trợ 2 loại giao thức rất phổ biến đó là MQTT (sử dụng TLS) và HTTPS. Và một thành phần có tên Data Broker sẽ có nhiệm vụ chuyển tiếp, phân phối các luồng dữ liệu đến một dịch vụ khác để thực hiện lưu trữ xử lý có tên gọi là Cloud Pub/Sub.

2.4.3. Cách thức bảo mật, kết nối thiết bị với GCIC

2.4.3.1. Bảo mật

Google IoT Core sử dụng kết nối TLS1.2 cho tầng transport và JWT cho tầng application để mã hóa và xác thực danh tính thiết bị bằng mã thông báo Web Json (JWTs, RFC 7519).



Hình 2. 9: Cơ chế xác thực giữa Device với Cloud

Khi một thiết bị kết nối với cầu MQTT, thiết bị kèm chuỗi JWT làm mật khẩu trong CONNECT tin nhắn MQTT để giải mã và xác thực thiết bị.

Cloud IoT Core sử dụng xác thực dựa trên chữ ký kỹ thuật số, cho cả mã thông báo đã ký RSA và Elliptic Curve.

Mỗi thiết bị có một khóa Public key và Private key riêng biệt. Thiết bị IoT (Client) sẽ sử dụng Private key để tạo phần signature cho JWT. Mã JWT này sẽ được gửi đến Google IoT Core mỗi khi cần kết nối để chứng minh danh tính cho thiết bị. Google IoT Core (Cloud) sẽ sử dụng Public key để giải mã và xác nhận danh tính của thiết bị.

2.4.3.2. Kết nối

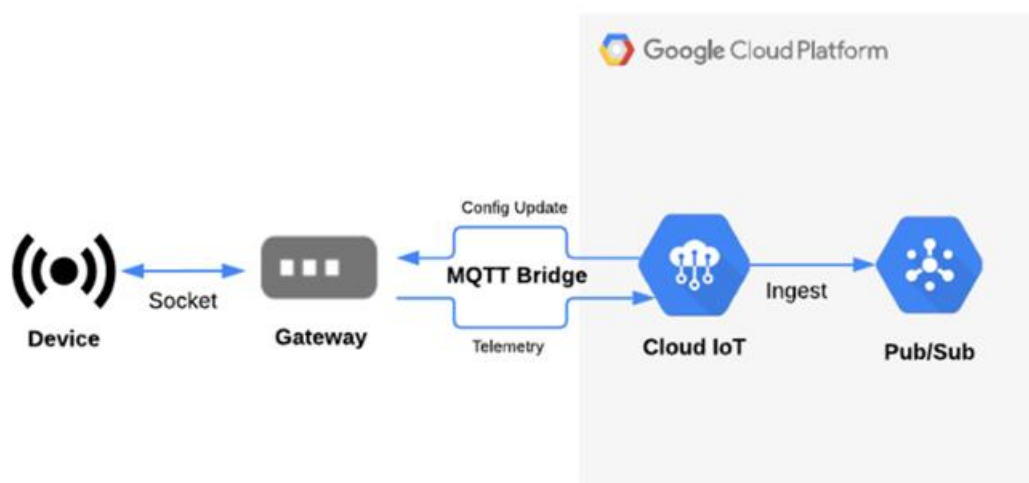
Cloud IoT Core hỗ trợ hai giao thức cho kết nối và giao tiếp thiết bị: MQTT và HTTP. Các thiết bị giao tiếp với Cloud IoT Core qua một "Bridge" - cầu nối MQTT hoặc cầu HTTP. Bridge MQTT/HTTP là một thành phần trung tâm của Cloud IoT Core.

MQTT 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.

HTTP là giao thức không kết nối: với cầu nối HTTP, các thiết bị không duy trì kết nối với Cloud IoT Core. Thay vào đó, họ gửi yêu cầu và nhận phản hồi. Cloud IoT Core chỉ hỗ trợ HTTP 1.1.

2.4.4. Quan hệ giữa thiết bị IoT với GCIC

Cloud IoT Core hỗ trợ giao thức MQTT bằng cách chạy một nhà môi giới được quản lý lắng nghe cổng `mqtt.googleapis.com:8883`. Cổng 8883 là cổng TCP tiêu chuẩn dành riêng để kết nối MQTT an toàn. Các kết nối với cổng này phải sử dụng truyền tải TLS, được hỗ trợ bởi các máy khách nguồn mở như Eclipse Paho.



Hình 2. 10: Sự tương tác giữa các Device với Cloud IoT qua MQTT

Trong một hệ thống sử dụng giao thức MQTT, nhiều client kết nối tới một server. Trong đó server được gọi là MQTT Broker, mỗi client sẽ đăng ký theo dõi các kênh thông tin (Topic) hoặc gửi dữ liệu lên kênh thông tin đó.

Quá trình đăng ký này gọi là “subscribe” và hành động một client gửi dữ liệu lên kênh thông tin được gọi là “publish”. Mỗi khi kênh thông tin đó được cập nhật dữ liệu (dữ liệu này có thể đến từ các client khác) thì những client nào đã đăng ký theo dõi kênh này sẽ nhận được dữ liệu cập nhật đó.

2.5. GIỚI THIỆU VỀ GOOGLE BIG QUERY, DATASTUDIO

2.5.1. BigQuery

2.5.1.1. Định nghĩa

Big query là một hệ thống lưu trữ dữ liệu bằng Google Cloud Storage được thiết kế cho một lượng rất lớn dữ liệu phân tán cao, cho phép các truy vấn SQL được thực thi trên nhiều cơ sở dữ liệu ở các mức cấu trúc khác nhau.



Hình 2. 11: Một số dịch vụ của BigQuery

Thay vì một chỉ mục cơ sở dữ liệu quan hệ SQL theo định hướng, theo hàng, truyền thống, BigQuery sử dụng một hệ thống lưu trữ cột trong đó các thành phần

của các bản ghi được xếp chồng lên nhau và được truyền trực tiếp đến một hệ thống lưu trữ song song.

2.5.1.2. Tính năng

Google BigQuery sử dụng các SQL quen thuộc tuân thủ tiêu chuẩn ANSI và cung cấp các trình điều khiển ODBC và JDBC giúp cho việc tích hợp dữ liệu của bạn trở nên nhanh chóng và dễ dàng.

Google BigQuery có thể giải quyết vấn đề về phân tích thời gian thực nhờ tận dụng cơ sở hạ tầng không cần máy chủ của Google với khả năng tự động mở rộng quy mô và tính năng nhập trực tuyến hiệu suất cao để tải dữ liệu.

Bộ nhớ được quản lý theo cột của BigQuery cùng quy trình thực thi song song với số lượng lớn và tính năng tối ưu hóa hiệu suất tự động giúp người dùng có thể phân tích dữ liệu một cách đồng thời và nhanh chóng mà không bị ảnh hưởng bởi số lượng người dùng hay kích cỡ của dữ liệu.

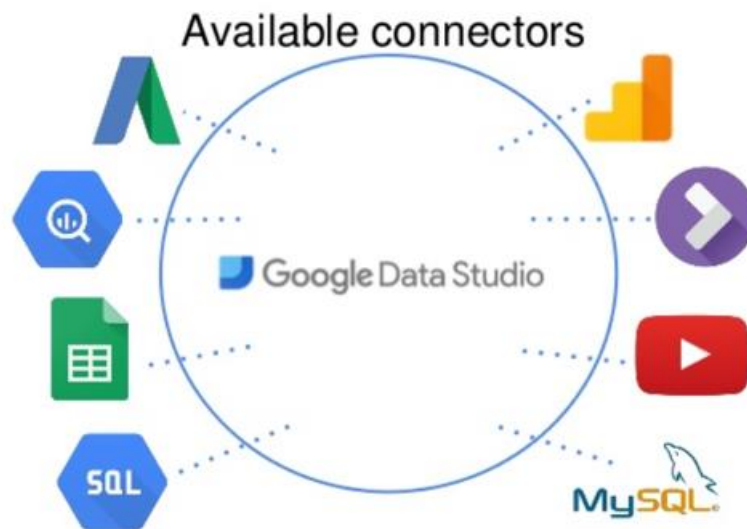
BigQuery cung cấp cho một cái nhìn tổng quát về dữ liệu bằng việc query liền mạch những dữ liệu được lưu trữ và quản lý bởi Big Query, Google Cloud, Google Cloud Big Table, Google sheet & Google Drive. BigQuery sẽ tích hợp với những công cụ như Tableau, MicroStrategy, Looker, Google DataStudio mà tự động nhập và trực quan hóa dữ liệu.

BigQuery giúp cho việc duy trì tính bảo mật với nhận dạng chi tiết và quyền kiểm soát quản lý truy cập trở nên dễ dàng hơn. Dữ liệu trên BigQuery luôn được mã hóa, lưu trữ hoặc luân chuyển nhằm bảo đảm tính bảo mật và toàn vẹn dữ liệu.

2.5.2. DataStudio

2.5.2.1. Định nghĩa

Google Data Studio là một công cụ giúp bạn kết nối, tổng hợp, phân tích, báo cáo dữ liệu. Google Data Studio giúp bạn kết nối dữ liệu từ nhiều nguồn khác nhau, giúp bạn phân tích để đưa ra được những báo cáo trực quan, dễ dàng chia sẻ và ra quyết định cụ thể.



Hình 2. 12: Google DataStudio với các dịch vụ hỗ trợ

Từ Data Studio có thể tổng hợp thành rất nhiều dạng biểu đồ như: Table, Pivot Table Scorecard, Time series, Bar, Pie, Geo Map, Line...

2.5.2.2. Tính năng

Xây dựng hệ thống báo cáo một cách trực quan và dễ sử dụng thông qua Visual hoá dữ liệu thông qua biểu đồ, bảng biểu, và giúp giảm chi phí cho việc xây dựng hệ thống báo cáo.

Có thể chia sẻ báo cáo tới mọi người, thống kê được với team, công ty, đối tác, khách hàng...

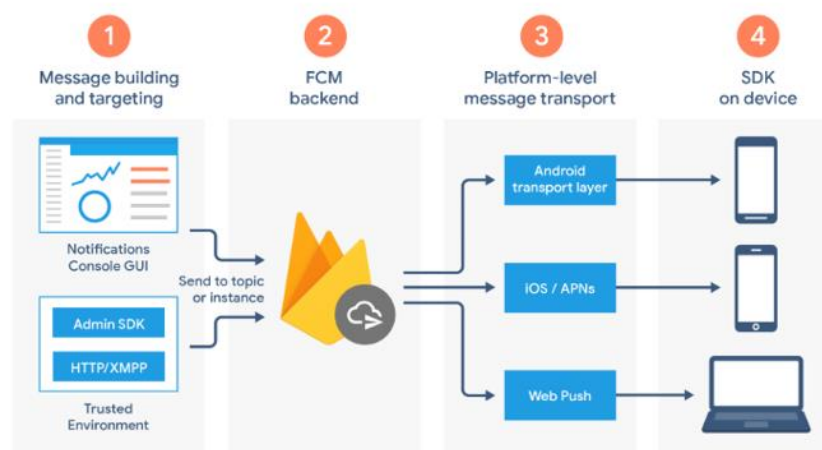
Dễ dàng kết nối với nhiều nguồn dữ liệu, chuyển đổi dữ liệu thô thành số liệu và kích thước cần thiết để tạo báo cáo và bảng điều kiện dễ theo dõi.

Các báo cáo sẽ được lưu trên Google Drive nên an toàn, không sợ bị mất dữ liệu hay gặp rủi ro lưu trữ.

2.6. GIỚI THIỆU VỀ GOOGLE FIREBASE

2.6.1. Khái niệm về Firebase

Firebase là một dịch vụ cơ sở dữ liệu thời gian thực hoạt động trên nền tảng đám mây được cung cấp bởi Google nhằm giúp các lập trình phát triển nhanh các ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu.



Hình 2. 13: Trao đổi dữ liệu giữa Firebase và SDK

Dữ liệu được lưu trữ trên Cloud (Cloud Hosted Database) dưới định dạng JSON và đồng bộ với tất cả máy khách theo thời gian thực kể cả khi Offline.

2.6.2. Dịch vụ cơ bản Firebase

2.6.2.1. Firebase Realtime Database

FRD là cơ sở dữ liệu theo mô hình “NoSQL database”, lưu trữ dữ liệu database dưới dạng JSON và thực hiện đồng bộ database tới tất cả các client theo thời gian thực. Cụ thể hơn là xây dựng được client đa nền tảng (Cross-Platform Client) và tất cả các client này sẽ cùng sử dụng chung một database đến từ Firebase và có thể tự động cập nhật mỗi khi dữ liệu trong database được thêm mới hoặc sửa đổi.

Nền tảng này hỗ trợ đồng bộ hóa dữ liệu của người dùng kể cả khi không có kết nối mạng, tạo nên trải nghiệm xuyên suốt bất chấp tình trạng kết nối internet của

người sử dụng. Realtime Database của Firebase hỗ trợ phát triển ứng dụng trên: Android, IOS, Web, C++, Unity, và cả Xamarin.

2.6.2.2. Cloud Firestore

Cloud Firestore là một cơ sở dữ liệu NoQuery được lưu trữ trên đám mây. Cloud Firestore hỗ trợ các cấu trúc dữ liệu linh hoạt, phân cấp dữ liệu. Lưu trữ dữ liệu của bạn trong các document, được tổ chức thành các collection cho bạn tổ chức dữ liệu và thực hiện truy vấn. Cloud Firestore phong phú hơn, nhanh hơn và có khả năng mở rộng siêu việt hơn so với Realtime Database.

2.6.2.3. Firebase Hosting

Firebase cung cấp các hosting và được phân phối qua SSL từ CDN sẽ giúp bạn tiết kiệm được rất nhiều thời gian trong việc xây dựng và phát triển ứng dụng. Ngoài ra có thể kết nối Firebase Hosting với các dịch vụ Cloud Function hoặc Cloud Run để xây dựng và lưu trữ Microservice trên Firebase.

2.6.3. Lợi ích firebase cung cấp

Xây dựng ứng dụng nhanh chóng mà không tốn thời gian, nhân lực để quản lý hệ thống và cơ sở hạ tầng phía sau. Cung cấp các chức năng như phân tích, cơ sở dữ liệu, báo cáo hoạt động và báo cáo các sự cố lỗi để có thể dễ dàng phát triển, định hướng ứng dụng nhằm đem lại các trải nghiệm tốt nhất cho người dùng.

Quản lý cấu hình và trải nghiệm các ứng dụng của Firebase tập trung trong một giao diện website đơn giản, các ứng dụng này hoạt động độc lập nhưng liên kết dữ liệu phân tích chặt chẽ.

Với việc sử dụng các kết nối thông qua giao thức bảo mật SSL hoạt động trên nền tảng cloud đồng thời cho phép phân quyền người dùng database bằng cú pháp javascript cũng nâng cao hơn nhiều độ bảo mật cho ứng dụng của bạn.

Firebase hoạt động dựa trên nền tảng cloud cung cấp bởi Google và hoạt động trên nền tảng Cloud vì vậy việc nâng cấp hay bảo trì Server cũng diễn ra rất đơn giản và nhanh chóng mà không ảnh hưởng đến Server.

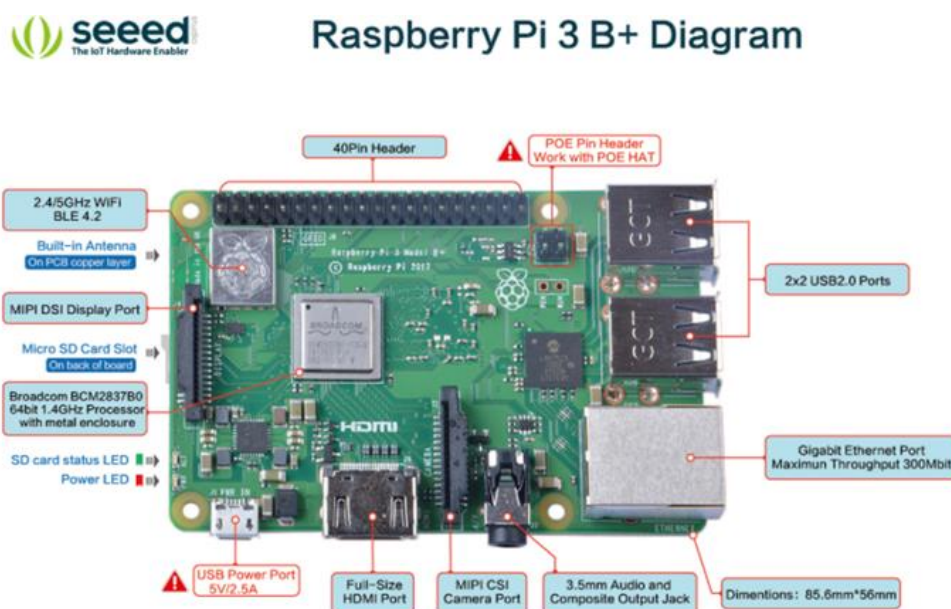
2.7. GIỚI THIỆU MỘT SỐ MODULE SỬ DỤNG TRONG MÔ HÌNH

2.7.1. Module Raspberry Pi 3

2.7.1.1. Giới thiệu

Là khối điều khiển trung tâm đóng vai trò như một bộ não có khả năng điều khiển tất cả các thiết bị khác trong nhà như bóng đèn, cảm biến gas, cảm biến nhiệt độ, độ ẩm,... trao đổi dữ liệu với Cloud thông qua giao thức MQTT.

Raspberry Pi là máy tính kích cỡ như một Smart Phone và chạy trên hệ điều hành Raspbian hoặc Linux sử dụng chip tích hợp cao - System on Chip (SoC), có khả năng xử lý và lưu trữ dữ liệu.



Hình 2. 14: Module Raspberry Pi 3 B+

2.7.1.2. Thông số kỹ thuật

Vi xử lý: Broadcom BCM2837B0, quad-core A53 64-bit SoC @1.4GHz.

RAM: 1GB LPDDR2 SDRAM.

Kết nối: 2.4GHz and 5GHz IEEE 802.11 b/g/n/ac wireless LAN, Bluetooth 4.2, BLE, Gigabit Ethernet over USB 2.0 (Tối đa 300Mbps).

Cổng USB: 4 x 2.0.

Cổng mở rộng: 40-pin GPIO.

Video và âm thanh: 1 cổng full-sized HDMI, Cổng MIPI DSI Display, cổng MIPI CSI Camera, cổng stereo output và composite video 4 chân.

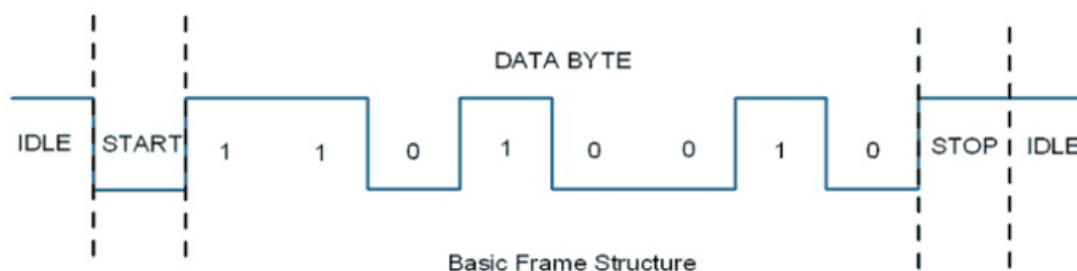
Multimedia: H.264, MPEG-4 decode (1080p30), H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics.

Nguồn điện sử dụng: 5V/2.5A DC cổng microUSB, 5V DC trên chân GPIO, Power over Ethernet (PoE) (yêu cầu thêm PoE HAT).

2.7.1.3. Giao tiếp UART trên Raspberry

UART – là viết tắt của Universal Asynchronous Receiver – Transmitter có nghĩa là truyền nhận dữ liệu nối tiếp bất đồng bộ.

Giao thức truyền thông nối tiếp UART sử dụng cấu trúc khung xác định cho các byte dữ liệu của chúng.



Hình 2. 15: Phương thức truyền dữ liệu qua giao tiếp UART

Thông thường, khung giao tiếp nối tiếp không đồng bộ bao gồm: bit START (1 bit) theo sau là byte dữ liệu (8 bit) và sau đó là bit STOP (1 bit), tạo thành khung 10 bit như trong hình trên. Khung cũng có thể bao gồm 2 bit STOP thay vì một bit đơn và cũng có thể có bit PARITY sau bit STOP.

Raspberry Pi có hai loại UART là: PL011 và mini UART. Trong Raspberry Pi 3, mini UART được sử dụng cho đầu ra bảng điều khiển Linux trong khi PL011 được kết nối với mô-đun Bluetooth On-board.

2.7.2. Module Arduino UNO R3

2.7.2.1. Giới thiệu

Arduino Uno là một bảng mạch vi điều khiển nguồn mở dựa trên vi điều khiển Microchip ATmega328 được phát triển bởi Arduino.cc. Bảng mạch được trang bị các bộ chân đầu vào/ đầu ra Digital và Analog có thể giao tiếp với các bảng mạch mở rộng khác nhau.



Hình 2. 16: Module Arduino UNO

Arduino Uno có 14 chân đầu vào và đầu ra kỹ thuật số (trong đó 6 chân có thể được sử dụng làm đầu ra PWM), 6 đầu vào tương tự, bộ cộng hưởng gồm 16 MHz (kết nối USB, giắc cắm nguồn, tiêu đề ICSP và nút đặt lại). Nó chứa mọi thứ cần thiết để hỗ trợ vi điều khiển; chỉ cần kết nối nó với máy tính bằng cáp USB hoặc cấp nguồn cho nó bằng bộ chuyển đổi AC-DC hoặc pin để bắt đầu.

Arduino Uno khác với tất cả các bo mạch trước ở chỗ nó không sử dụng chip điều khiển FTDI USB-to-serial. Thay vào đó, nó có chip vi điều khiển Atmega8U2 được lập trình như một bộ chuyển đổi từ USB sang nối tiếp.

2.7.2.2 Thông số kỹ thuật

Chip điều khiển chính: ATmega328P.

Chip nạp và giao tiếp UART: ATmega16U2.

Nguồn điện sử dụng: 5VDC từ cổng USB hoặc nguồn ngoài cắm từ giắc tròn DC. Nên cấp nguồn từ 6~9VDC để đảm bảo mạch hoạt động tốt, nếu bạn cắm 12VDC thì IC ổn áp rất nóng, dễ cháy và gây hư hỏng mạch.

Số chân Digital I/O: 14 (trong đó 6 chân có khả năng xuất xung PWM).

Số chân PWM Digital I/O: 6.

Số chân Analog Input: 6.

Dòng điện DC Current trên mỗi chân I/O: 20 mA.

Dòng điện DC Current chân 3.3V: 50 mA.

Flash Memory: 32 KB (ATmega328P), 0.5 KB dùng cho bootloader.

SRAM: 2 KB (ATmega328P).

EEPROM: 1 KB (ATmega328P).

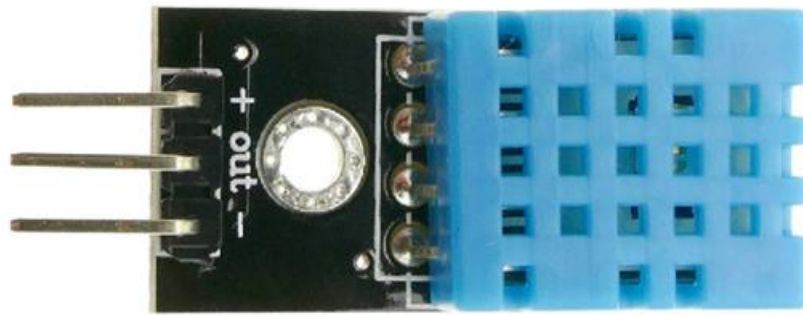
Clock Speed: 16 MHz.

Kích thước: 68.6 x 53.4 mm.

2.7.3. Module cảm biến nhiệt độ, độ ẩm DHT11

2.7.3.1 Giới thiệu

Cảm biến nhiệt độ, độ ẩm DHT11 có khả năng đo nhiệt độ và độ ẩm với một đầu ra kỹ thuật số. Module lấy dữ liệu thông qua giao tiếp 1 wire (giao tiếp 1 dây). Bộ tiền xử lý tín hiệu tích hợp trong cảm biến giúp bạn có được dữ liệu chính xác mà không cần phải qua bất kỳ tính toán nào. Module được thiết kế hoạt động ở mức điện áp 5VDC.



Hình 2. 17: Cảm biến nhiệt độ độ ẩm DHT11

DHT11 gửi và nhận dữ liệu với một dây tín hiệu DATA, với chuẩn dữ liệu truyền 1 dây này, chúng ta phải đảm bảo sao cho ở chế độ chờ (IDLE) dây DATA có giá trị ở mức cao, nên trong mạch sử dụng DHT11, dây DATA phải được mắc với một trở kéo bên ngoài (thông thường giá trị là $4.7k\Omega$).

Dữ liệu truyền về của DHT11 gồm 40bit dữ liệu theo thứ tự: 8 bit biểu thị phần nguyên của độ ẩm + 8 bit biểu thị phần thập phân của độ ẩm + 8 bit biểu thị phần nguyên của nhiệt độ + 8 bit biểu thị phần thập phân của nhiệt độ + 8 bit check sum.

2.7.3.2. Thông số kỹ thuật

Điện áp hoạt động: 5VDC.

Chuẩn giao tiếp: TTL, 1 wire.

Khoảng đo độ ẩm: 20%-80%RH sai số $\pm 5\%$ RH.

Khoảng đo nhiệt độ: 0-50°C sai số $\pm 2^\circ\text{C}$.

Tần số lấy mẫu tối đa 1Hz (1 giây / lần).

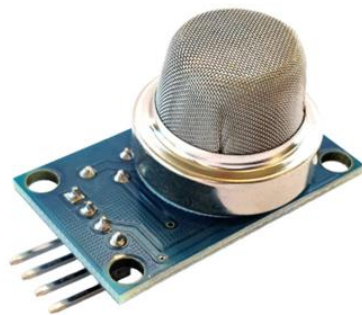
Kích thước: 28mm x 12mm x 10mm.

2.7.4. Module cảm biến khí ga MQ 135

2.7.4.1. Giới thiệu

Cảm biến khí ga MQ-135 là một trong những loại cảm biến được sử dụng để nhận biết các thông số khí ga như: LPG, I-Butan, Propane, Methane , Alcohol, Hydrogen, Smoke.

Được thiết kế với độ nhạy cao, thời gian đáp ứng nhanh. Giá trị đọc được từ cảm biến sẽ được đọc về từ chân Analog của vi điều khiển.



Hình 2. 18: Cảm biến khí Gas MQ 135

Vật liệu nhạy cảm của cảm biến khí MQ-135 là SnO_2 , có độ dẫn thấp hơn trong không khí sạch. Khi phát hiện có khí đốt dễ cháy, độ dẫn của cảm biến nhiệt độ cao hơn cùng với nồng độ khí tăng lên.

2.7.4.2. Thông số kỹ thuật

Điện áp nguồn: 5V DC.

Điện áp của heater: $5V \pm 0.1$ AC/DC.

Điện trở tải: thay đổi được ($2k\Omega$ - $47k\Omega$)

Điện trở của heater: $33\Omega \pm 5\%$.

Công suất tiêu thụ của heater: ít hơn 800mW.

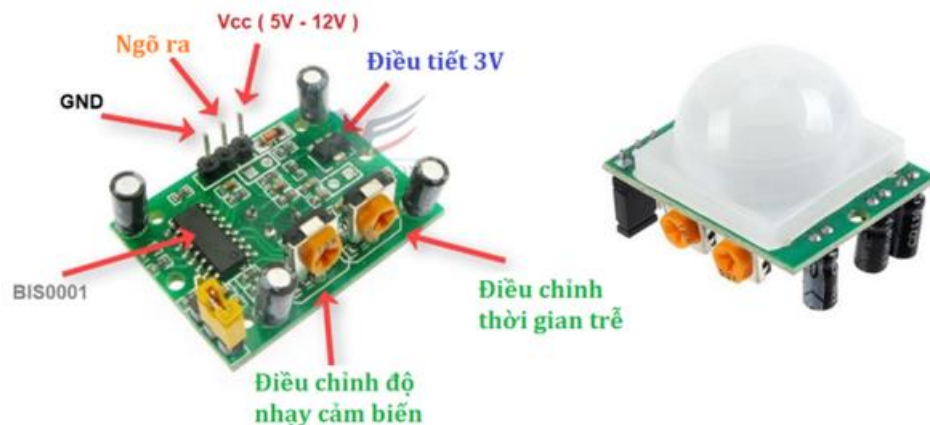
Khoảng phát hiện: 10 - 300 ppm NH_3 , 10 - 1000 ppm Benzene,...

Kích thước: 32mm*20mm.

2.7.5. Module cảm biến thân nhiệt chuyển động PIR HS-SR501

2.7.5.1. Giới thiệu

Cảm biến thân nhiệt chuyển động PIR (Passive infrared sensor) HC-SR501 được sử dụng để phát hiện chuyển động của các vật thể phát ra bức xạ hồng ngoại của con người, con vật, các vật phát nhiệt,...



Hình 2. 19: Cảm biến thân nhiệt chuyển động

Cảm biến có thể chỉnh được độ nhạy để giới hạn khoảng cách bắt xa gần cũng như cường độ bức xạ của vật thể mong muốn. Ngoài ra cảm biến còn có thể điều chỉnh thời gian kích trễ (giữ tín hiệu bao lâu sau khi kích hoạt) qua biến trở tích hợp sẵn.

2.7.5.2. Thông số kỹ thuật

Điện áp hoạt động: 5V ~ 12V DC (khuyến dùng: 5V)

Dòng điện tiêu thụ: 65mA

Điện áp đầu ra: mức cao 3,3V, mức thấp 0V

Thời gian trễ: Điều chỉnh (0,3 giây ~ 18 giây)

Phạm vi cảm ứng: góc dưới 110° và xa 7m

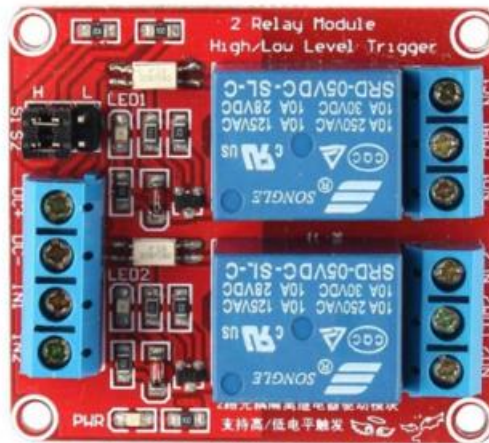
Nhiệt độ hoạt động: -20°C ~ +80°C

Kích thước mạch: 32 * 24mm, lỗ vít 28mm, đường kính vít 2 mm, nắp cảm ứng (đường kính): 23mm

2.7.6. Module Relay Opto Cách Ly 5VDC

2.7.6.1. Giới thiệu

Module 2 Relay với opto cách ly 5v nhỏ gọn, có opto và transistor cách ly giúp cho việc sử dụng trở nên an toàn với board mạch chính, mạch được sử dụng để đóng ngắt nguồn điện công suất cao AC hoặc DC, có thể chọn đóng khi kích mức cao hoặc mức thấp bằng Jumper.



Hình 2. 20: Relay opto cách ly mức cao thấp 5VDC

Tiếp điểm đóng ngắt gồm 3 tiếp điểm NC (thường đóng), NO (thường mở) và COM (chân chung) được cách ly hoàn toàn với board mạch chính, ở trạng thái bình thường chưa kích NC sẽ nối với COM, khi có trạng thái kích COM sẽ chuyển sang nối với NO và mất kết nối với NC.

2.7.6.2. Thông số kỹ thuật

Điện áp sử dụng: 5VDC.

Tín hiệu kích: TTL 3.3~5VDC, mức thấp Low Relay đóng, mức cao High Relay ngắt.

Mỗi Relay tiêu thụ dòng khoảng 80mA. Có đèn báo đóng ngắt trên relay.

Điện thế đóng ngắt tối đa: AC250V ~ 10A hoặc DC30V ~ 10A (Để an toàn nên dùng cho tải có công suất <100W).

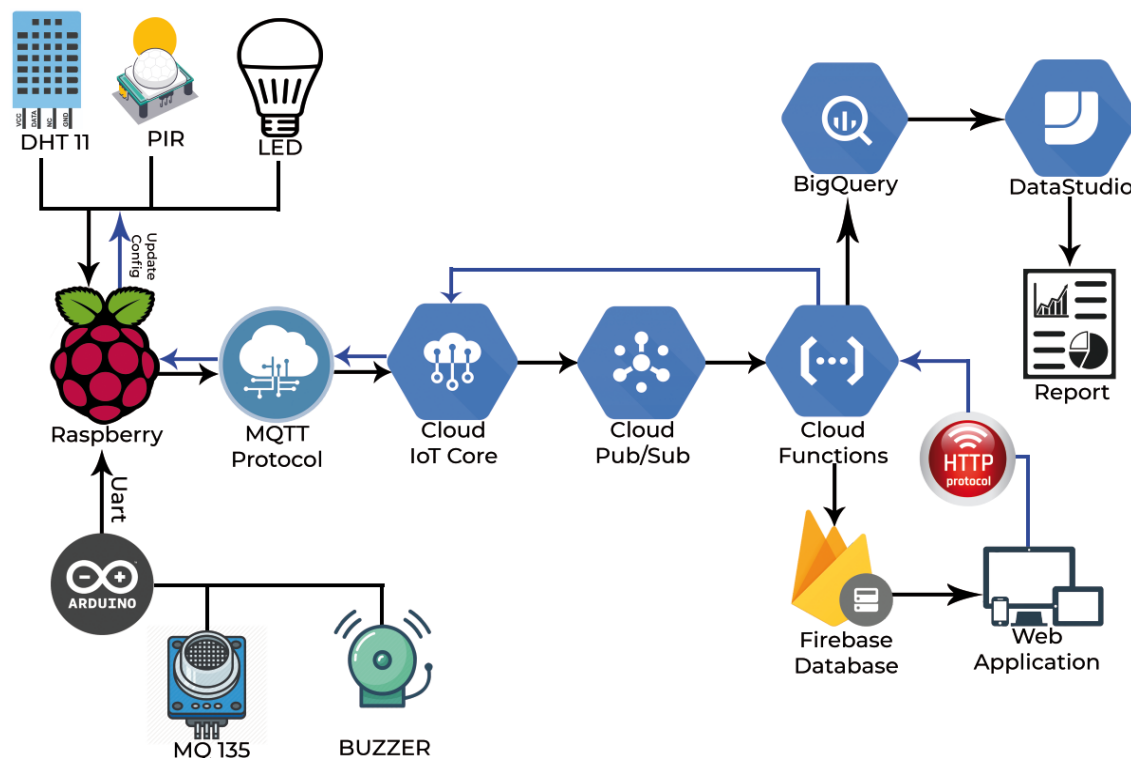
Tích hợp Opto cách ly, Diod chống nhiễu và đèn báo tín hiệu kích.

Kích thước: 39 x 51 x 20mm.

Chương 3. TÍNH TOÁN THIẾT KẾ, XÂY DỰNG VÀ THI CÔNG HỆ THỐNG

3.1. THIẾT KẾ VÀ XÂY DỰNG

3.1.1. Mô hình tổng quan



Hình 3. 1: Mô hình tổng quan hệ thống

Hệ thống bao gồm các phần:

Device: bao gồm các thiết bị vật lý: vi xử lý trung tâm là raspberry, Arduino, hệ thống đèn Led, cảm biến, loa cảnh báo. Thiết bị sẽ giao tiếp và gửi dữ liệu cũng như nhận phản hồi (config device) từ GCIC từ xa thông qua mạng Internet bằng giao thức MQTT (ngoài ra GCIC hỗ trợ giao tiếp các thiết bị với cloud thông qua giao thức HTTP).

GCIC: có nhiệm vụ kết nối các thiết bị với cloud thông qua giao thức MQTT Protocol. GCIC quản lý giám sát các thiết bị. Khi một gói tin gửi từ thiết bị IoT lên Google IoT Core, Google IoT Core sẽ chỉ lưu trữ một vài loại gói tin đặc biệt liên quan đến cấu hình hoặc trạng thái của thiết bị và ít bị thay đổi, còn những gói tin được gửi với số lượng lớn và thường xuyên hơn thì sẽ cần chuyển tiếp đến một dịch vụ khác có tên là Cloud Pub/Sub.

Cloud Pub/Sub: không lưu trữ message để xử lý hoặc để phân tích dữ liệu.

Cloud Functions: là thành phần trung gian để kết nối, giao tiếp dữ liệu với các hệ thống khác. Từ đó có thể truy cập dữ liệu từ xa thông qua các SDK của nền tảng dịch vụ đang sử dụng

Firebase Database: dữ liệu từ các devive sẽ được sẽ lưu vào Firebase Database Realtime và Firestore thông qua chức năng đám mây Cloud Functions.

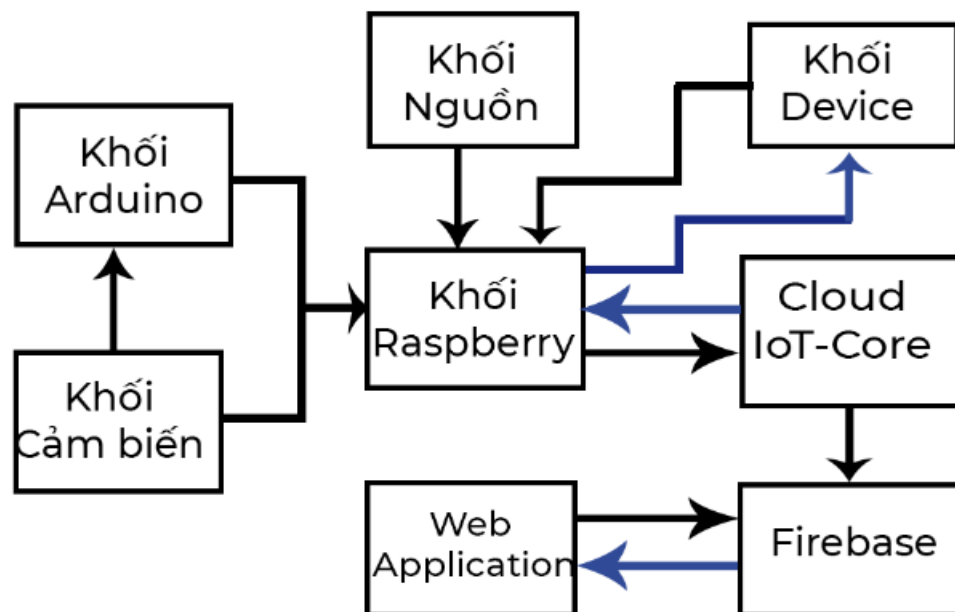
BigQuery: dữ liệu từ các devive sẽ được sẽ lưu lại thông qua chức năng đám mây Cloud Functions dung để phân tích.

Report: thông qua DataStudio dữ liệu sẽ được trực quan hóa, biểu diễn dưới dạng biểu đồ giúp chúng ta có thể xử lý và phân tích dữ liệu một cách dễ dàng.

Web Application: nhận dữ liệu từ FRD hoặc Bigquery để hiện thị dữ liệu, qua giao diện web người dùng có thể điều khiển và tương tác với từng thiết bị trong hệ thống.

3.1.2. Sơ đồ khối hệ thống

Hệ thống bao gồm các phần:



Hình 3. 2: Sơ đồ khối hoạt động của hệ thống

Khối Raspberry: sử dụng Raspberry Pi 3 B+ để điều khiển toàn bộ hệ thống, đóng vai trò như máy chủ Webserver để nhận và thực thi các yêu cầu từ người dùng.

Khối nguồn: bao gồm nguồn điện 5V-9V cung cấp điện để đảm bảo hệ thống hoạt động ổn định.

Khối cảm biến: thu thập các thông tin dữ liệu từ môi trường bên ngoài như: nhiệt độ, độ ẩm, lượng mưa, khí ga,.. thông qua các các module cảm biến.

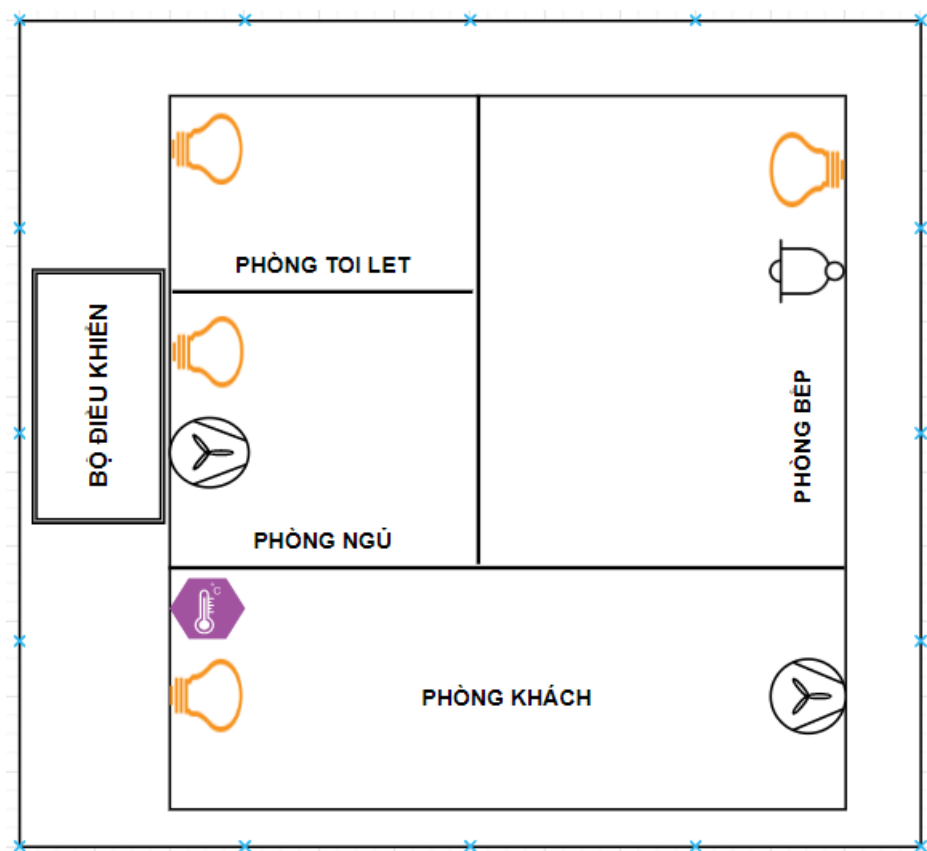
Khối Device : bao gồm các thiết bị relay, đèn, quạt,...

Khối PubSub: gửi nhận các message từ GCIC.

Khối Firebase: nơi lưu trữ cơ sở dữ liệu.

Web Application: giao diện ứng dụng website để hiển thị dữ liệu, người dùng có thể tương tác với hệ thống để điều khiển các thiết bị.

3.1.3. Thiết kế mô hình



Hình 3. 3: Bản vẽ mô hình thiết kế

Bố cục bao gồm 5 khu vực chính:

Phòng khách: gồm hệ thống thiết bị đèn, quạt, cảm biến nhiệt độ, độ ẩm.

Phòng ngủ: gồm hệ thống thiết bị đèn, quạt.

Phòng bếp: hệ thống cảm biến khí ga và còi báo động khi phát hiện sự cố cháy nổ, khí ga đạt ngưỡng nguy hiểm.

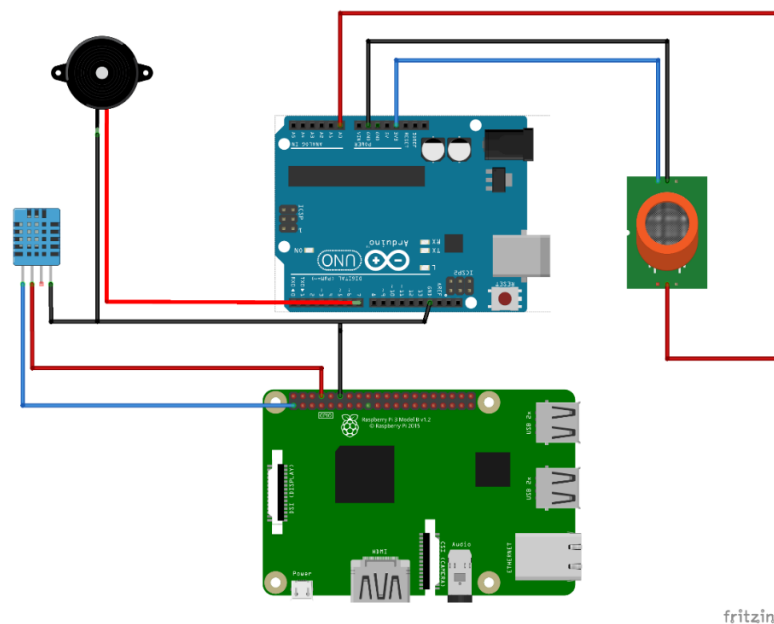
Phòng toilet: hệ thống cảm biến chuyển động tự động bật tắt đèn tự động khi có người sử dụng.

Bộ điều khiển: vi xử lý Raspberry Pi, Arduino, nguồn điện và dây cắm.

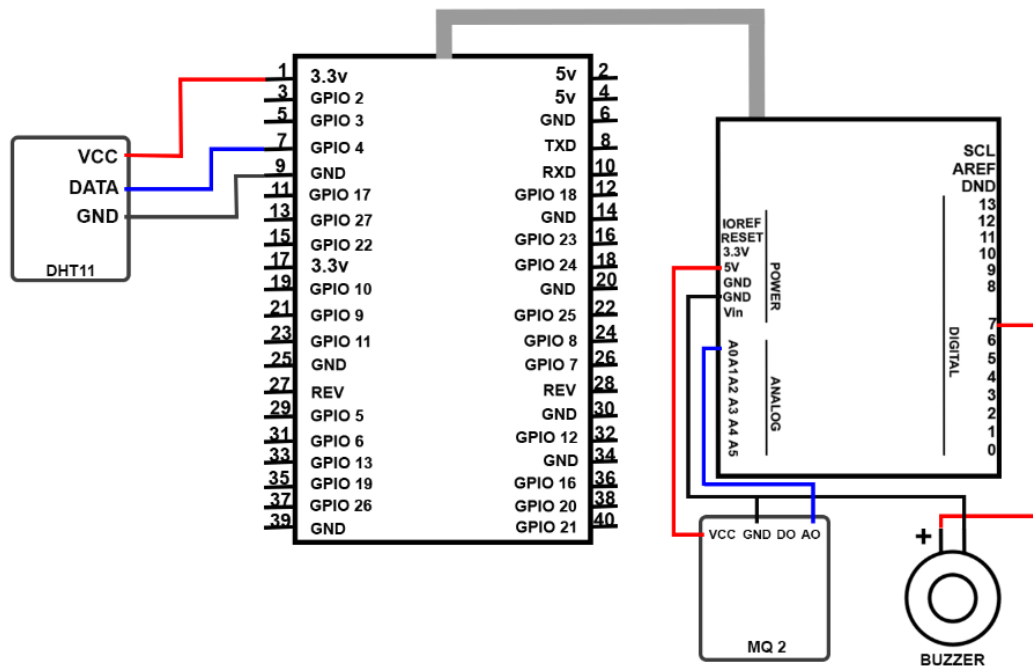
3.1.4. Thiết kế mạch điều khiển với hệ thống cảm biến

Raspberry Pi 3 trong smarthome đóng vai trò như server lưu trữ dữ liệu, tương tác với các thiết bị, trao đổi dữ liệu thông qua giao thức MQTT.

Raspberry đóng vai trò MQTT server (Broker) trong mô hình MQTT còn thiết bị đóng vai trò MQTT client. Trong đó mỗi MQTT client đăng ký 1 kênh (Topic) gửi để trao đổi dữ liệu với server gọi là Subcribe, mỗi client (có thể là điện thoại, ứng dụng của người dùng) có thể nhận được dữ liệu từ bất kỳ trạm nào khác gửi sang kênh đã đăng ký, việc gửi dữ liệu sang kênh đăng ký gọi là publish.

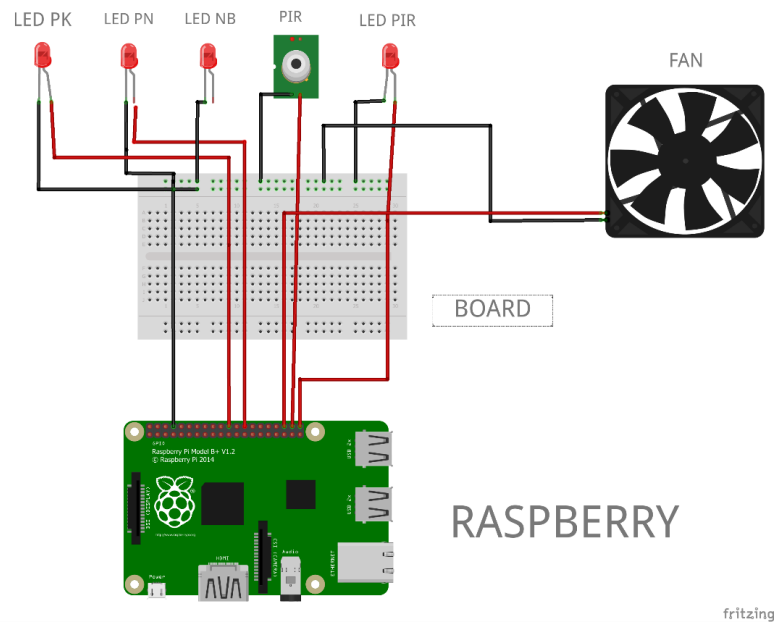


Hình 3. 4: Mô hình triển khai với hệ thống cảm biến

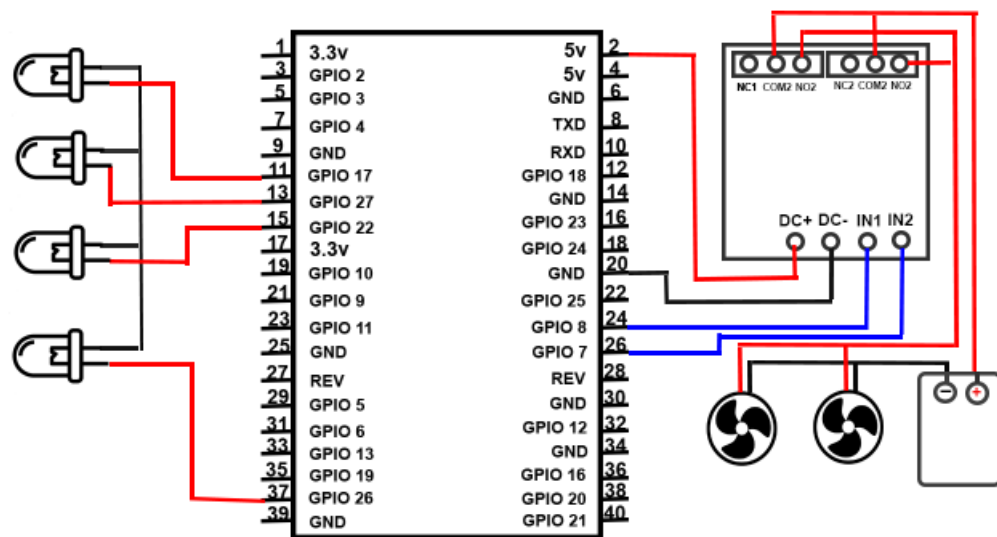


Hình 3. 5: Sơ đồ kết nối hệ thống cảm biến

3.1.5. Thiết kế mạch điều khiển với hệ thống điều khiển



Hình 3. 6: Mô hình triển khai hệ thống điều khiển



Hình 3. 7: Sơ đồ kết nối hệ thống điều khiển

3.1.6. Thiết kế website

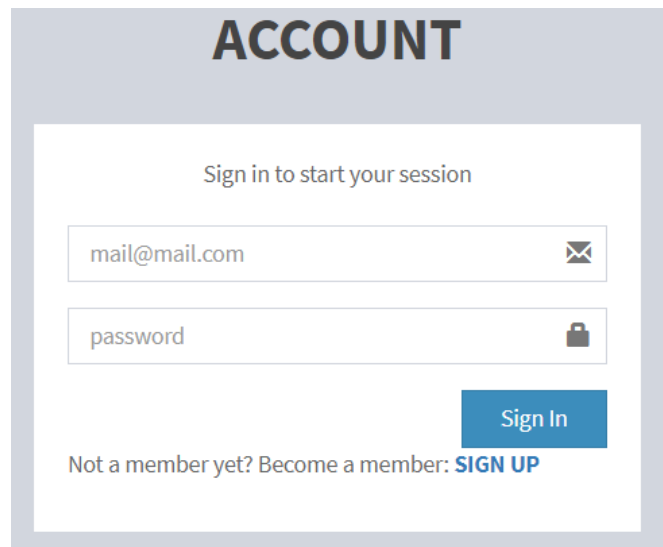
Sử dụng ngôn ngữ lập trình Html, Css, JavaScript để xây dựng ứng dụng Website để gửi, nhận dữ liệu giữa cơ sở dữ liệu. Từ đó người dùng có thể giám sát và trao đổi với hệ thống.

Để sử dụng các chức năng, thao tác dữ liệu người dùng cần một tài khoản để đăng nhập vào ứng dụng web.

Các chức năng chính của website: Điều khiển các thiết bị, hiển thị, cảnh báo dữ liệu từ các cảm biến khi đến ngưỡng quy định.

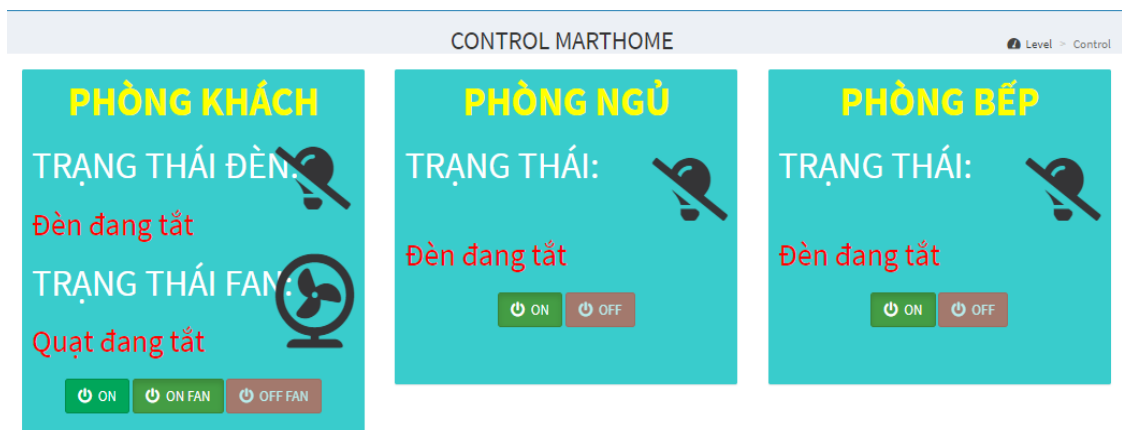
Bố cục chính gồm các giao diện cơ bản như: giao diện đăng nhập, giao diện hiển thị, điều khiển các thiết bị, biểu đồ dữ liệu,...

Giao diện đăng nhập: người dùng cần một username và password để đăng nhập.



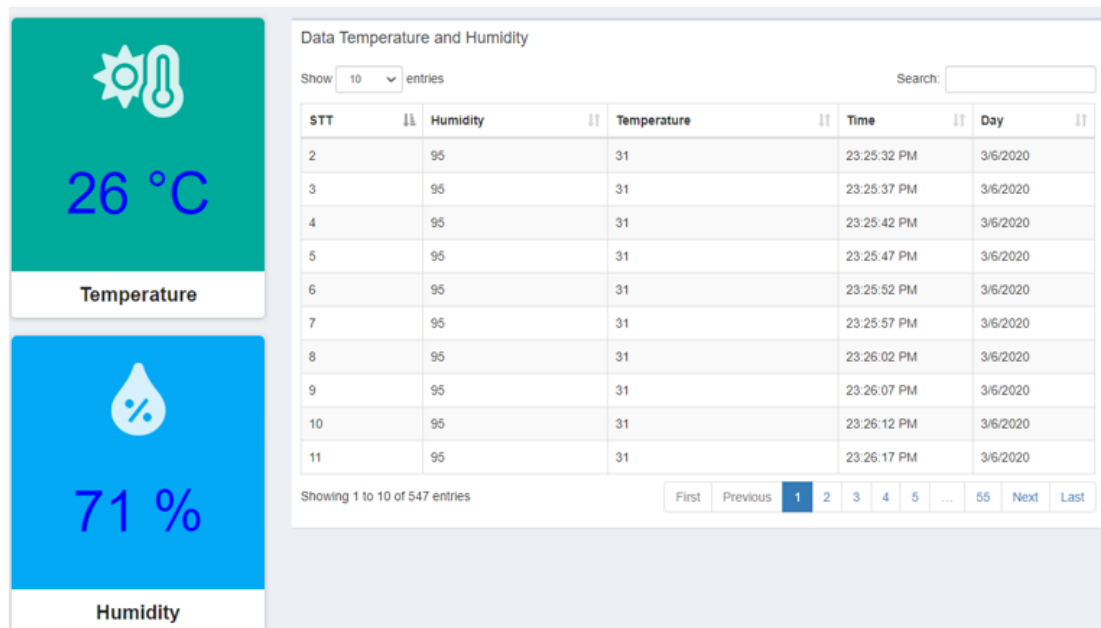
Hình 3. 8: Giao diện đăng nhập

Giao diện điều khiển thiết bị: cho phép người dùng điều khiển, giám sát hoạt động của các thiết bị trong nhà.



Hình 3. 9: Giao diện điều khiển

Giao diện hiển thị dữ liệu: dữ liệu từ các cảm biến sẽ được cập nhật liên tục và hiển thị dưới dạng các đoạn văn bản, bảng dữ liệu để dễ dàng quan sát.



Hình 3. 10: Giao diện hiển thị dữ liệu nhiệt độ, độ ẩm

Giao diện biểu đồ theo thời gian thực: dữ liệu sẽ được cập nhật liên tục.



Hình 3. 11: Biểu đồ dữ liệu theo thời gian thực

3.2. THI CÔNG HỆ THỐNG

3.2.1. Thiết lập Raspberry

3.2.1.1. Chuẩn bị , yêu cầu

Raspberry Pi 3 model B+.

Máy tính, bàn phím và chuột.

Kết nối Internet trong quá trình cài đặt và thiết lập.

3.2.1.2 Các bước thực hiện

Cài đặt hệ điều hành Raspbian trên Raspberry Pi.

Cài đặt các gói, thư viện phụ thuộc cần thiết như:

```
sudo apt update && sudo apt upgrade
```

```
sudo apt install git
```

```
sudo apt install python3
```

```
sudo apt install build-essential libssl-dev libffi-dev python3-dev
```

```
sudo pip install jwt
```

```
sudo pip install paho-mqtt
```

```
sudo pip3 install adafruit-io
```

```
sudo pip install requests
```

```
sudo pip install python-firebase
```

```
sudo pip install pyserial
```

Tạo khóa công khai (public) và riêng tư (private) để kết nối thiết bị với Google Cloud IoT Core:

```
openssl genpkey -algorithm RSA -out rsa_private.pem -pkeyopt  
rsa_keygen_bits:2048
```

```
openssl rsa -in rsa_private.pem -pubout -out rsa_public.pem
```

Tạo chứng chỉ Google CA root certificate:

```
wget https://pki.goog/roots.pem
```

3.2.2. Thiết lập Google Cloud IoT Core

3.2.2.1. Chuẩn bị, yêu cầu

Cần phải có tài khoản gmail, để đăng ký và triển khai các dịch vụ của Google Cloud cung cấp.

Chuẩn bị thẻ Visa hoặc Mastercard để kích hoạt phương thức thanh toán.

Cần có kết nối mạng Internet hoặc Wifi trong quá trình thực hiện.

3.2.2.2. Các bước thực hiện

Bước 1: Trong bảng điều khiển console, chọn project hoặc nhấn New Project để tạo dự án mới..

Bước 2: Bật phương thức thanh toán cho dự án vừa tạo.

Bước 3: Kích hoạt các API cần sử dụng:

1. Cloud IoT Core API.
2. Pub/Sub API.
3. BigQuery API.

Bước 4: Tạo sổ đăng ký để thiết bị có thể kết nối

1. Mở trang điều khiển Iot Core console.
2. Click Create a device registry.
3. Điền các thông tin: Registry ID, chọn Region, phương thức kết nối, Cloud Pub/Sub.
4. Còn lại giữ nguyên mặc định ban đầu, sau đó nhấn Create.

Bước 5: Tạo các thiết bị, device

1. Mở trang IoT Core console.
2. Click vào registry vừa tạo, sau đó chọn vào phần Device để thêm thiết bị mới
3. Click Create Device, nhập Device ID.
4. Sao chép nội dung ở file rsa_public.pem vào phần Public key value ở Authentication (optional).
5. Còn lại giữ nguyên mặc định và nhấn create.

Bước 6: Tạo topic để xem dữ liệu

1. Mở trang IoT Core console.
2. Chọn vào phần quản lý PubSub.
3. Click Topics : chọn Create Topic.
4. Điền thông tin: Topic ID.
5. Nhấn Create để tạo mới.

Bước 7: Tạo Subscriptions

1. Chọn vào Subscriptions, sau đó tạo Subscriptions mới.
2. Điền Subscriptions ID, chọn Topic vừa tạo.
3. Còn lại giữ nguyên mặc định và nhấn Create.

3.2.3. Thiết lập Google Firebase**3.2.3.1. Chuẩn bị, yêu cầu**

Có tài khoản gmail để đăng nhập vào firebase.

Cần kết nối Internet trong quá trình thực hiện.

3.2.3.2. Các bước thực hiện**Bước 1: Tạo dự án Firebase.**

1. Trong bảng điều khiển Firebase , bấm Thêm dự án , sau đó chọn hoặc nhập tên dự án . Nếu bạn có dự án Google Cloud Platform (GCP) hiện có, bạn có thể chọn dự án từ menu thả xuống để thêm tài nguyên Firebase vào dự án đó.
2. Nếu bạn đang tạo một dự án mới, bạn có thể chỉnh sửa ID dự án .
3. Nhấp vào Tạo dự án (hoặc Thêm Firebase , nếu bạn đang sử dụng dự án GCP hiện tại).

Bước 2: Đăng ký ứng dụng mới với Firebase

1. Ở trang Firebase console's project nhấp vào biểu tượng Web để khởi chạy quy trình thiết lập.
2. Nhập tên ứng dụng, còn lại để mặc định.
3. Click vào Register app

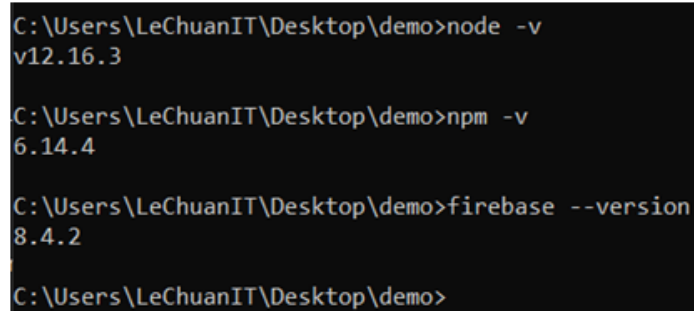
3.2.4. Triển khai dự án lên Hosting

Bước 1: Cài đặt Node JS từ trang chủ: <https://nodejs.org/en/download/>. Sau khi cài đặt xong kiểm tra cài đặt thành công sử dụng lệnh trong command line windows (cmd):

node -v

Npm -v

Sau khi cài đặt thành công sẽ hiện thị version đã cài đặt.



```
C:\Users\LeChuanIT\Desktop\demo>node -v
v12.16.3

C:\Users\LeChuanIT\Desktop\demo>npm -v
6.14.4

C:\Users\LeChuanIT\Desktop\demo>firebase --version
8.4.2

C:\Users\LeChuanIT\Desktop\demo>
```

Hình 3. 12: Kiểm tra version node js

Chạy lệnh sau đây để thiết lập và cài đặt môi trường Firebase CLI: npm install -g firebase-tools. Để kiểm tra cài đặt thành công, gõ firebase --version , nếu cài đặt thành công sẽ hiện thị tên phiên bản đã cài đặt.

Bước 2: Tại thư mục vừa tạo, Thực thi lệnh sau để khởi tạo dự án: firebase init . Nhấn Y để tiếp tục.

Bước 3: Lựa chọn những tính năng mà Firebase CLI cung cấp. Nhấn phím cách (space) để lựa chọn, sau đó nhấn Enter để xác nhận.

```
C:\Users\LeChuanIT\Desktop\demo>firebase init

##### 
##      ##          #####          ##          ##### 
##      ## ##      ## ##          ## ## ## ## ##      ## 
#####      ##          #####          #####          ##### 
##          ## ##      ## ##          ##      ## ##      ## ## 
##          #### ##      ##          #####      ##      ##          ##### 

You're about to initialize a Firebase project in this directory:

C:\Users\LeChuanIT\Desktop\demo

? Are you ready to proceed? Yes
? Which Firebase CLI features do you want to set up for this folder? Press Space to select features, then Enter
to confirm your choices.
(*) Database: Deploy Firebase Realtime Database Rules
( ) Firestore: Deploy rules and create indexes for Firestore
(*) Functions: Configure and deploy Cloud Functions
> (*) Hosting: Configure and deploy Firebase Hosting sites
( ) Storage: Deploy Cloud Storage security rules
( ) Emulators: Set up local emulators for Firebase features
```

Hình 3. 13: Khởi tạo dự án

Bước 4: Tiếp theo chúng ta sẽ liên kết với dự án firebase mà đã tạo ở trên.
Nhấn Enter để xác nhận.

```
=== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? Please select an option: Use an existing project
? Select a default Firebase project for this directory:
  clouddiot-277609 (CloudIoT)
  homeauto-277608 (HomeAuto)
  kltn-276713 (KLTN)
> kltn2-3a6fe (kltn2)
  smarthome-cloud-1234 (smarthome-cloud-1234)
  summer-rope-277604 (CloudGoogle)
```

Hình 3. 14: Chọn project cần triển khai

Bước 5: Thiết lập Database, Nhấn Enter để tiếp tục.

Bước 6: Thiết lập Cloud Functions. Ở đây lựa chọn ngôn ngữ để viết chức năng đám mây JavaScript hoặc TypeScript.

```
=== Functions Setup
```

A functions directory will be created in your project with a Node.js package pre-configured. Functions can be deployed with `firebase deploy`.

? What language would you like to use to write Cloud Functions? **JavaScript**

Hình 3. 15: Lựa chọn ngôn ngữ để viết các functions

Nhấn Enter để xác nhận. Tiếp theo nếu cần sử dụng ESLint để bắt lỗi. Nếu không dùng thì nhấn N để bỏ qua.

Bước 7: Cài đặt các gói thư viện phụ thuộc cho dự án. Nhấn Y để cài đặt.

```
? Do you want to install dependencies with npm now? Yes
> protobufjs@6.9.0 postinstall C:\Users\LeChuanIT\Desktop\demo\functions\node_modules\protobufjs
> node scripts/postinstall

npm notice created a lockfile as package-lock.json. You should commit this file.
added 252 packages from 206 contributors and audited 252 packages in 19.594s

29 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Hình 3. 16: Cài đặt các gói thư viện cần thiết

Bước 8: Cấu hình Firebase Hosting. Nhấn Enter để tiếp tục.

```
=== Hosting Setup

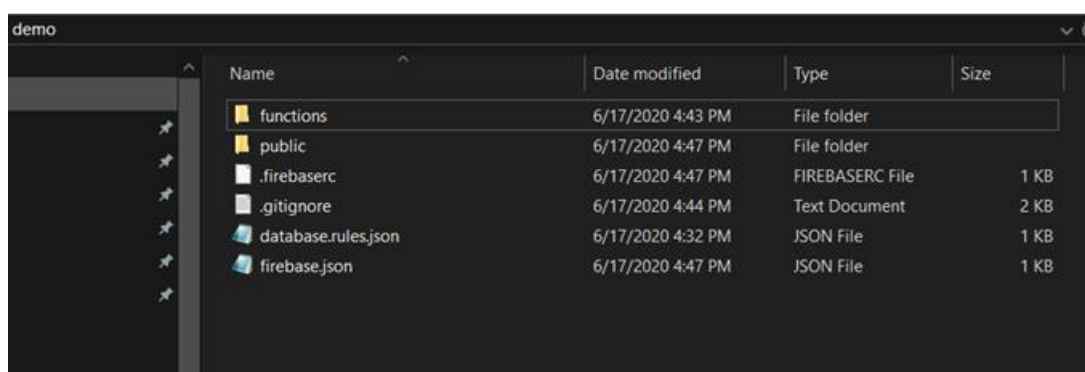
Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? public
? Configure as a single-page app (rewrite all urls to /index.html)? No
+ Wrote public/404.html
+ Wrote public/index.html

i Writing configuration info to firebase.json...
i Writing project information to .firebaserc...
+ Firebase initialization complete!
```

Hình 3. 17: Cấu hình hosting

Sau khi cài đặt thành công cấu trúc dự án của dự án như hình bên dưới:



Hình 3. 18: Cấu trúc thư mục của project

Bước 8: Deploy dự án lên hosting sử dụng lệnh: `firebase deploy --only hosting`. Kết quả:

```
C:\Users\LeChuanIT\Desktop\demo>firebase deploy --only hosting

=== Deploying to 'kltn2-3a6fe'...

i  deploying hosting
i  hosting[kltn2-3a6fe]: beginning deploy...
i  hosting[kltn2-3a6fe]: found 2 files in public
+  hosting[kltn2-3a6fe]: file upload complete
i  hosting[kltn2-3a6fe]: finalizing version...
+  hosting[kltn2-3a6fe]: version finalized
i  hosting[kltn2-3a6fe]: releasing new version...
+  hosting[kltn2-3a6fe]: release complete

+  Deploy complete!

Project Console: https://console.firebase.google.com/project/kltn2-3a6fe/overview
Hosting URL: https://kltn2-3a6fe.web.app
```

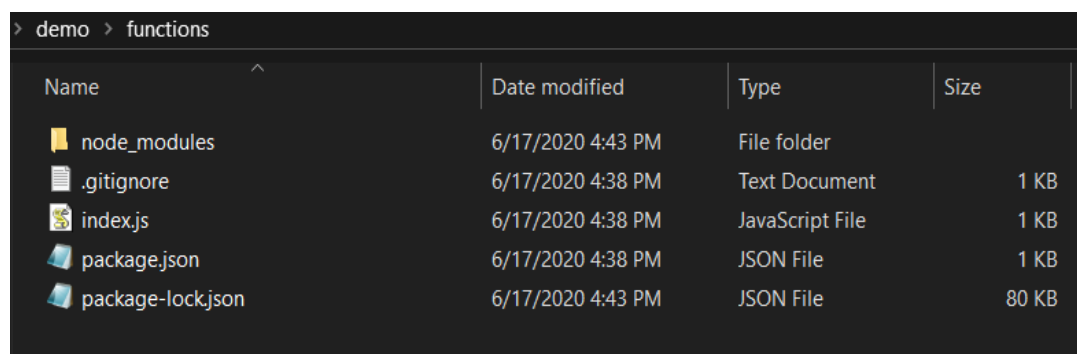
Hình 3. 19: Triển khai dự án lên hosting

Sau khi deploy thành công thì ta thấy Hosting URL: là địa chỉ trang web của mình: <https://kltn2-3a6fe.web.app/>

3.2.5. Triển khai các hàm chức năng Functions

Bước 1: Tại thư mục dự án vừa tạo. Mở command line windows (Nhấn giữ Shift + click vào chuột phải chọn Open PowerShell window here).

Bước 2: Các hàm chức năng triển khai lên đám mây sẽ được viết trong thư mục functions. Open thư mục ta thấy cấu trúc như hình:

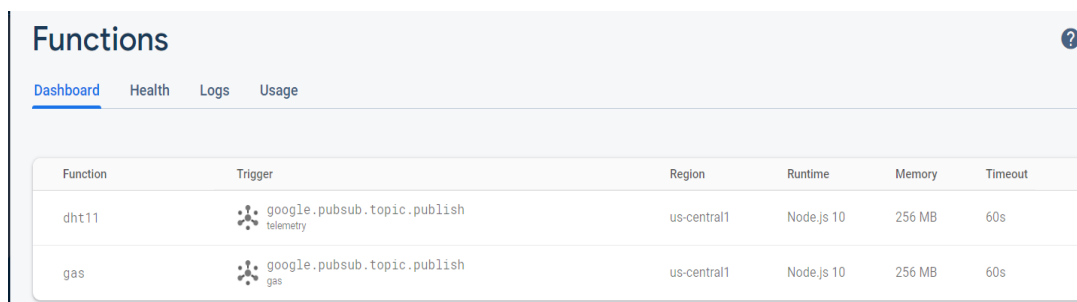


Name	Date modified	Type	Size
node_modules	6/17/2020 4:43 PM	File folder	
.gitignore	6/17/2020 4:38 PM	Text Document	1 KB
index.js	6/17/2020 4:38 PM	JavaScript File	1 KB
package.json	6/17/2020 4:38 PM	JSON File	1 KB
package-lock.json	6/17/2020 4:43 PM	JSON File	80 KB

Hình 3. 20: Cấu trúc thư mục functions

Bước 3: Ta thực hiện viết các chức năng cho đám mây trong file `index.js`. Tiếp theo lưu lại và quay trở về command gốc ban đầu.

Bước 4: Tiến hành triển khai hàm chức năng trên đám mây. Thực thi lệnh sau: `firebase deploy --only functions`. Sau khi gõ lệnh này tự động chương trình sẽ chạy và deploy các chức năng (functions) trong file `index.js`. Sau khi deploy thành công ta truy cập vào firebase và kiểm tra.



Hình 3. 21: Functions đã triển khai thành công trên firebase

Như vậy là đã deploy thành công. Ở đây có hai hàm chức: `dht11` và `gas`. Functions `dht11` và `gas` sẽ nhận dữ liệu từ Cloud Pub/Sub và lưu trữ vào cơ sở dữ liệu Firebase Database Realtime và Google BigQuery.

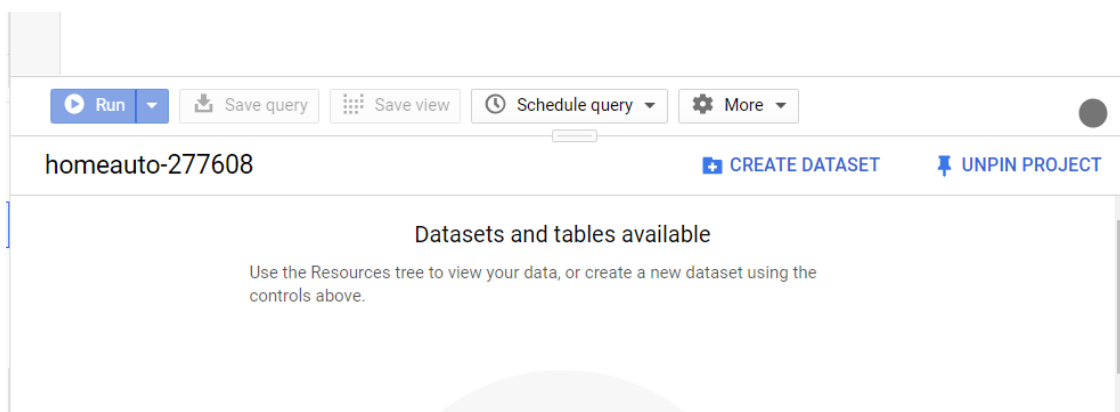
3.2.6. Thiết lập BigQuery và DataStudio

3.2.6.1. BigQuery

Bước 1: Trong bảng điều khiển console, chọn project dự án của mình.

Bước 2: Trong Tab Navigation Menu. Nhấn chọn BigQuery.

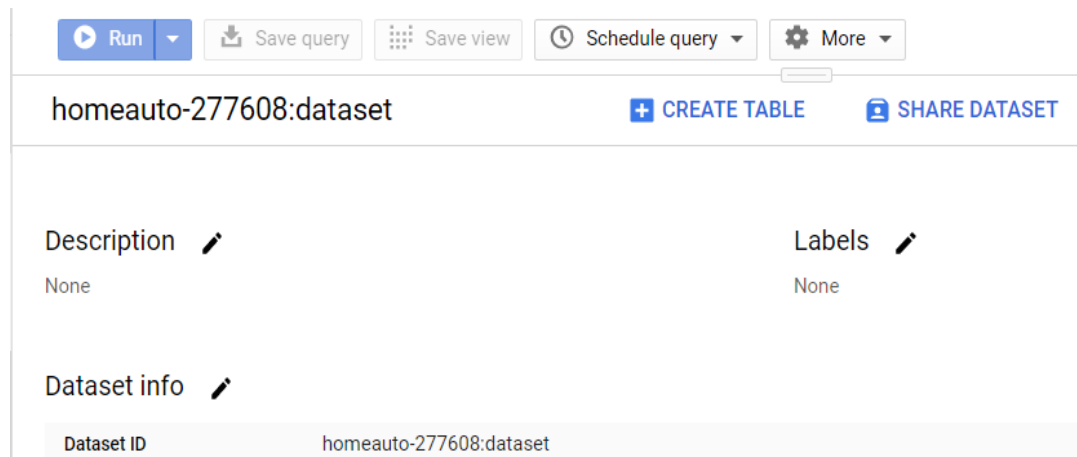
Bước 3: Tiếp theo cần tạo DataSet bằng cách ấn button CREATE DATASET



Hình 3. 22: Tạo Dataset

Bước 4: Nhập Dataset ID và nhấn Create Dataset.

Bước 5: Tiếp theo tạo các table cho dataset đó là hoàn thiện việc tạo dữ liệu cho dataset. Để làm điều này Click button CREATE TABLE giống như hình dưới.



Hình 3. 23: Tạo bảng mới để lưu trữ dữ liệu

Bước 6: Sau khi nhấn CREATE TABLE thì hiện thị một giao diện mới cho phép người dùng nhập các thông số cơ bản của một bảng dữ liệu. Table name nhập tên table muốn đặt. Ấn button Add field để thêm các cột mới. Ở đây BigQuery cung cấp cho người dùng một số kiểu dữ liệu cho các cột (String, Float, Date,...) và các thuộc tính khác. Nhấn Create để tạo bảng mới.

Create table

Source

Create table from:

Empty table

Destination

☒ Search for a project ☐ Enter a project name

Project name: HomeAuto

Dataset name: dataset

Table type: Native table

Table name: demo

Schema

☐ Edit as text

Name	Type	Mode	
id	STRING	NULLABLE	×
name	STRING	NULLABLE	×
+ Add field			

Partition and cluster settings

Partitioning: ?

Create table Cancel

Hình 3. 24: Tạo bảng để lưu trữ dữ liệu

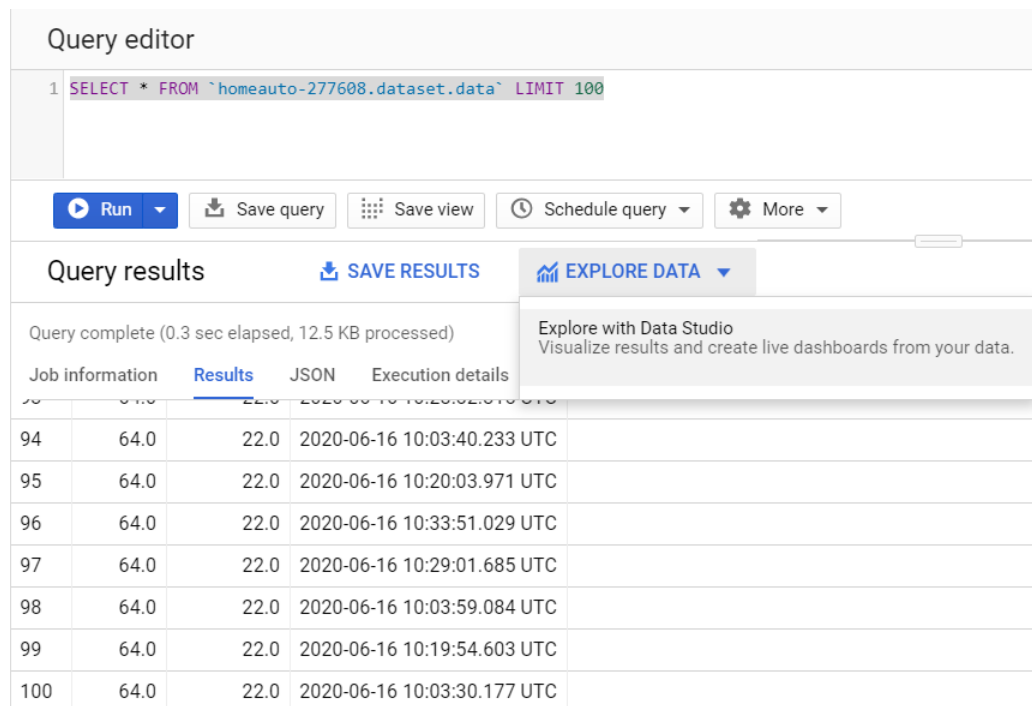
Bước 7: Sau khi tạo bảng thành công mỗi lần ta thu thập dữ liệu từ Raspberry thì dữ liệu sẽ tự động lưu vào table mới thông qua hàm chức năng (functions) đã triển khai ở trên.

Bước 8: Tiếp theo ta thực hiện 1 số truy vấn cơ bản: `SELECT * FROM `homeauto-277608.dataset.data` LIMIT 100`. Câu lệnh này sẽ lấy 100 thông tin của bảng data

Khi có dữ liệu có thể lưu lại và thực hiện kết nối với DataStudio để trực quan hóa dữ liệu để dễ dàng phân tích dữ liệu.

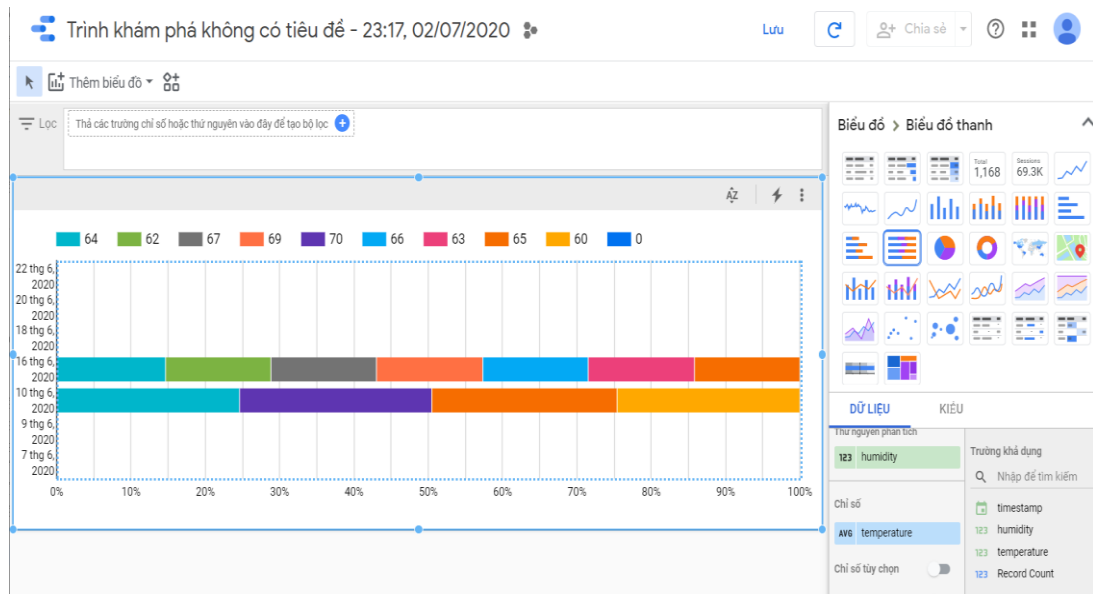
3.2.6.2. DataStudio

Bước 1: Sau khi thực hiện truy vấn dữ liệu. Ta chọn vào EXPLORE DATA để thực hiện kết nối DataStudio.



Hình 3. 25: Explore dữ liệu thông qua Goggle DataStudio

Bước 2: Khi kết nối thành công GDS sẽ cung cấp cho ta một số biểu đồ thông dụng. Ta chọn từng loại biểu đồ sẽ có hiển thị khác nhau.



Hình 3. 26: Biểu đồ dữ liệu của DataStudio

Ngoài ra chúng ta có thể thống kê trung bình, tính tổng, phương sai,... thông qua các tùy chọn bên dưới. Hoặc có thể viết câu lệnh SQL để truy vấn dữ liệu trực tiếp theo tùy chọn của người dùng

Hình 3. 27: Một số phương thức tính toán dữ liệu DataStudio hỗ trợ

Chương 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1. KẾT QUẢ ĐẠT ĐƯỢC

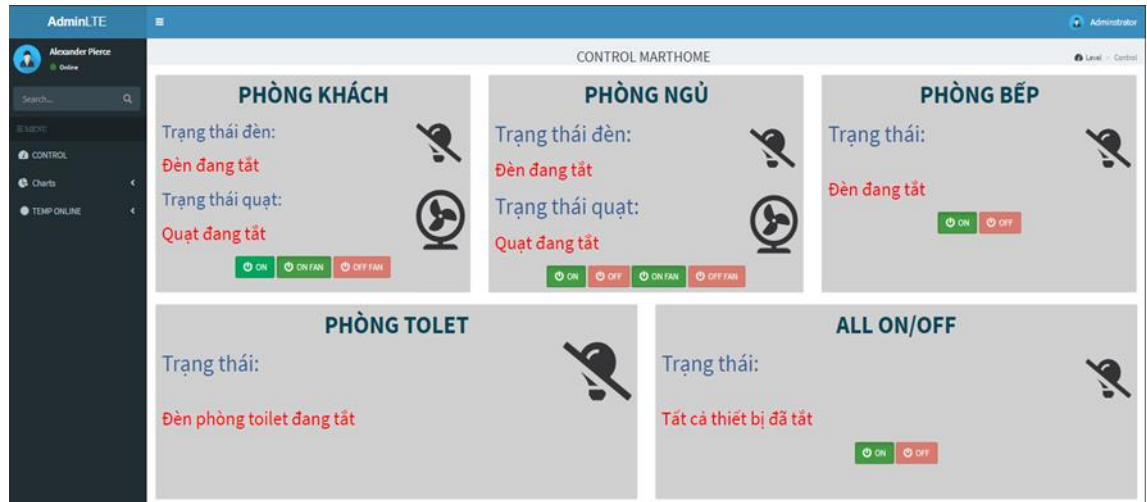
Qua quá trình nghiên cứu và thực hiện đề tài, nhóm đã hoàn thiện mô hình, có chức năng quản lý điều khiển và giám sát thiết bị trong gia đình thiết bị. Thiết kế mô hình tương đối nhỏ gọn phù hợp với hệ thống điện trong gia đình.

4.1.1. Thiết kế được mô hình mô phỏng ngôi nhà thông minh.

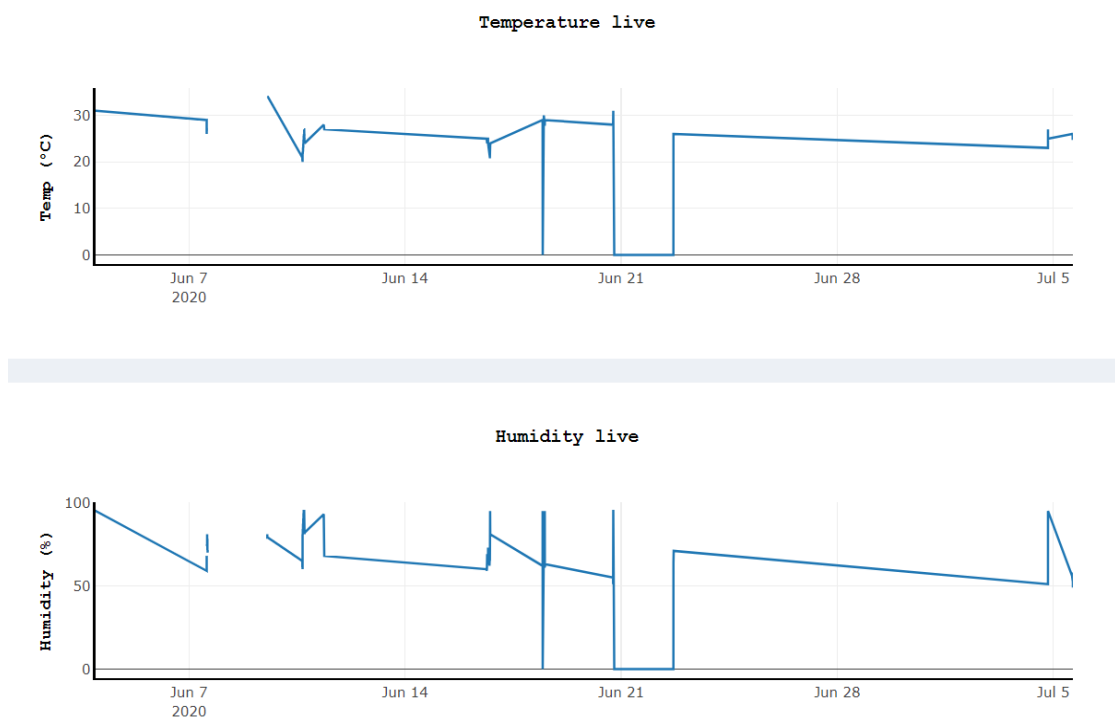


Hình 4. 1: Mô hình nhà thông minh

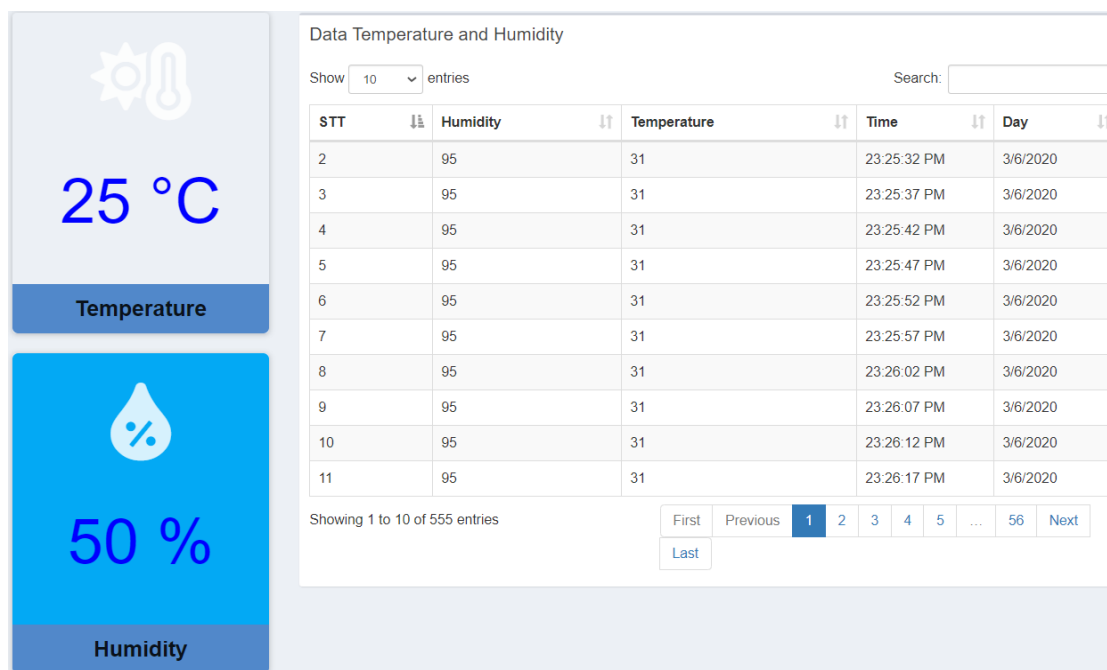
4.1.2. Thiết kế giao diện website để điều khiển và giám sát trạng thái của các thiết bị trong nhà. Giao diện thông số, bảng dữ liệu từ các cảm biến theo thời gian thực.



Hình 4. 2: Giao diện website



Hình 4. 3: Biểu đồ dữ liệu real-time



Hình 4. 4: Bảng dữ liệu thu thập nhiệt độ, độ ẩm

4.1.3. Gửi được dữ liệu, message từ thiết bị lên Google Cloud IoT thông qua giao thức MQTT để xuất bản lên Cloud Pub/Sub .

```
namnguyenpay@cloudshell:~ (homeauto-277608)$ gcloud pubsub subscriptions pull sub-ctl-led --auto-ack
```

DATA	MESSAGE_ID	ATTRIBUTES	DELIVERY_ATTEMPT
{ "led1": 0, "led2": 0, "led3": 0, "led4": 0, "fan1": 0, "fan2": 0 }	1325870814351063	deviceId=device-control deviceNumId=2676885521556118 deviceRegistryId=my-registry-ctr deviceRegistryLocation=us-central1 projectId=homeauto-277608 subFolder=	

```
namnguyenpay@cloudshell:~ (homeauto-277608)$
```

Hình 4. 5: Dữ liệu của đèn led trên Pub/Sub

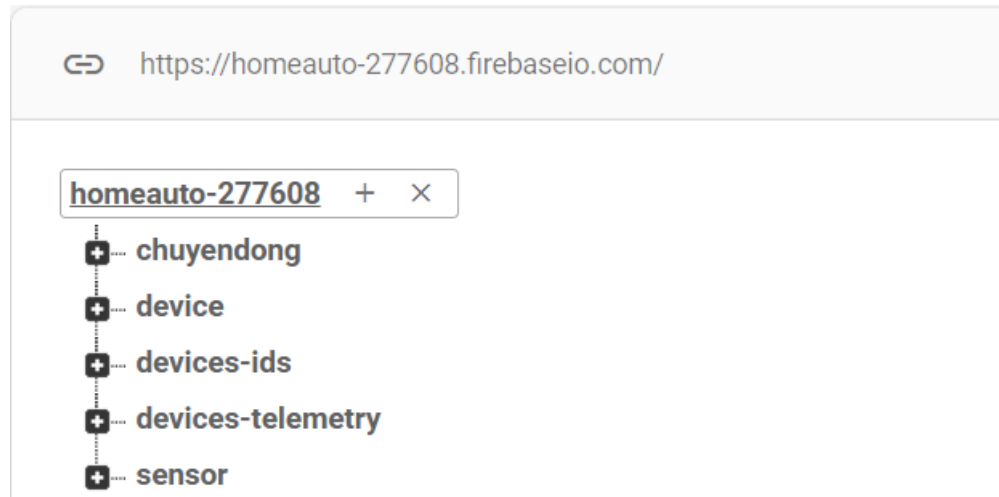
DATA	MESSAGE_ID	ATTRIBUTES	DELIVERY_ATTEMPT
{ "humidity": 51.0, "temperature": 23.0 }	1322764625386327	deviceId=dht11 deviceNumId=2635939132044903 deviceRegistryId=my-registry2 deviceRegistryLocation=us-central1 projectId=homeauto-277608 subFolder=	
{ "humidity": 51.0, "temperature": 23.0 }	1322778779520941	deviceId=dht11 deviceNumId=2635939132044903 deviceRegistryId=my-registry2 deviceRegistryLocation=us-central1 projectId=homeauto-277608 subFolder=	
{ "humidity": 51.0, "temperature": 23.0 }	1322787208347213	deviceId=dht11 deviceNumId=2635939132044903 deviceRegistryId=my-registry2 deviceRegistryLocation=us-central1 projectId=homeauto-277608 subFolder=	

Hình 4. 6: Dữ liệu nhiệt độ, độ ẩm trên Pub/Sub

4.1.4. Đưa được dữ liệu lên Firestore, Firebase Database thông qua chức năng Cloud Function đã triển khai ở trên.

🏠 > device-configs > device-control		
🏠 homeauto-277608	📁 device-configs	📁 device-control
+ Start collection	+ Add document	+ Start collection
device-configs >	device-control >	+ Add field
		fan1: 0 fan2: 0 led1: 0 led2: 0 led3: 0 led4: 0

Hình 4. 7: Dữ liệu đèn led trên Firestore



Hình 4. 8: Dữ liệu trên Firebase Real-time

4.1.5. Đưa dữ liệu lên Google BigQuery để có thể thực hiện truy vấn và phân tích dữ liệu.

Schema	Details	Preview		
58	60.0	20.0	2020-06-10 09:19:52.209 UTC	
59	60.0	20.0	2020-06-10 09:14:50.485 UTC	
60	60.0	20.0	2020-06-10 09:20:02.557 UTC	
61	60.0	20.0	2020-06-10 09:20:13.265 UTC	
62	60.0	20.0	2020-06-10 09:14:41.543 UTC	
63	64.0	20.0	2020-06-10 09:17:25.057 UTC	
64	64.0	20.0	2020-06-10 09:17:34.784 UTC	
65	64.0	20.0	2020-06-10 09:17:00.524 UTC	

Hình 4. 9: Dữ liệu nhiệt độ, độ ẩm trên BigQuery

Schema Details Preview			
Row	gas	timestamp	
1	384.0	2020-06-18 05:04:55.973 UTC	
2	400.0	2020-06-16 10:36:44.169 UTC	
3	400.0	2020-06-18 04:58:58.312 UTC	
4	400.0	2020-06-18 04:58:49.197 UTC	
5	400.0	2020-06-18 04:59:08.413 UTC	
6	400.0	2020-06-18 05:42:30.583 UTC	
7	416.0	2020-06-16 10:53:36.521 UTC	

Hình 4. 10: Dữ liệu khí gas trên BigQuery

Thực hiện phân tích dữ liệu bằng công cụ BigQuery của Google Cloud. Thực hiện các câu lệnh truy vấn cơ bản để tính toán và thống kê dữ liệu.

Query editor

```

1 SELECT
2     avg(gas) as Gia_tri_trung_binh,
3     min(gas) as Gia_tri_nho_nhat,
4     max(gas) as Gia_tri_lon_nhat,
5 FROM `homeauto-277608.dataset.gas`
6

```

Run

Save query

Save view

Schedule query

More

Query results

SAVE RESULTS

EXPLORE DATA

Query complete (0.1 sec elapsed, cached)

Job information

Results

JSON

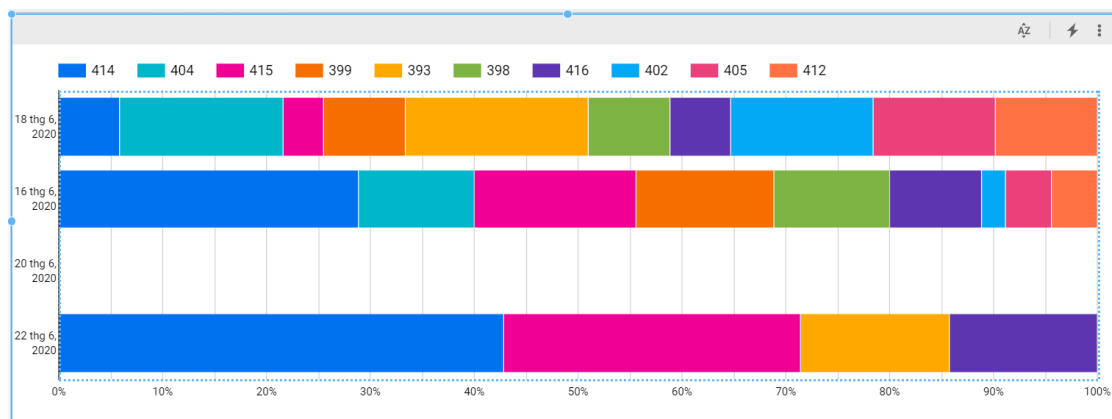
Execution details

Row	Gia_tri_trung_binh	Gia_tri_nho_nhat	Gia_tri_lon_nhat
1	426.82421875000017	376.0	523.0

Hình 4. 11: Truy vấn dữ liệu khí ga trên BigQuery

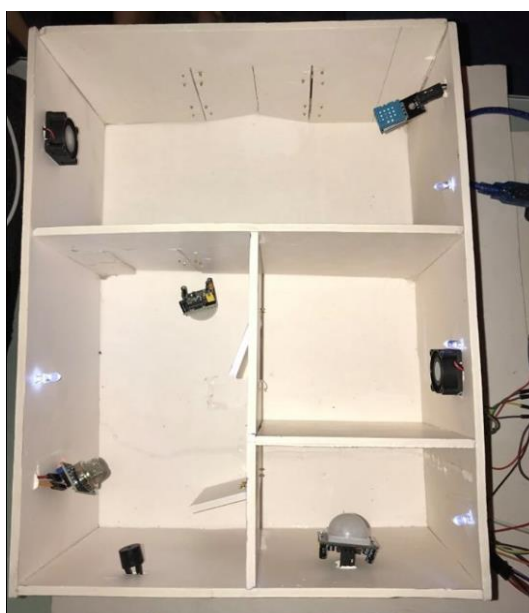
4.1.6 Trực quan hóa dữ liệu trên Google DataStudio.

Trực quan hóa dữ liệu, tạo báo cáo thông qua Google DataStudio để có thể tổng hợp, phân tích và dự đoán từ nguồn dữ liệu đã lưu trữ từ BigQuery.



Hình 4. 12: Biểu đồ dữ liệu trong Google DataStudio

4.1.7. Điều khiển và thu thập dữ liệu thành công.



Hình 4. 13: Điều khiển nhà thông minh

4.2. HẠN CHẾ

Tốc độ xử lý, điều khiển phụ thuộc vào tốc độ mạng Internet của người dùng.

Hệ thống điều khiển qua giao diện website chưa có App Android để điều khiển hệ thống qua ứng dụng điện thoại.

Hệ thống chưa có cảnh báo chống trộm, hệ thống phơi đồ, kéo rèm cửa tự động.

Chưa xây dựng kịp các tính năng phát triển cho sản phẩm như: hẹn giờ, thiết lập bối cảnh bật tắt thiết bị tự động theo thời gian cụ thể.

4.3. HƯỚNG PHÁT TRIỂN

Đề tài của được chúng em xây dựng tương đối hoàn chỉnh, song chúng em tự nhận thấy rằng đề tài của mình còn nhiều thiếu sót vì vậy đề đề tài này thêm phần phong phú, mang tính áp dụng trong thực tế hơn, có khả năng làm việc cao hơn thì đề tài của em cần phải có thêm các tính năng như sau:

Bổ sung thêm hệ thống cửa ra vào có thể nhận diện bằng dấu vân tay hoặc nhận diện khuôn mặt để phát hiện và cảnh báo khi có người lạ đột nhập vào nhà.

Thêm chức năng thiết lập ngày giờ, lập lịch để tự động hóa các thiết bị trong ngôi nhà như: tự động tắt đèn, thiết bị khi không có người sử dụng.

Phát triển ứng dụng trên Android, IOS để người dùng có thể trực tiếp giám sát thông qua app trên thiết bị di động.

Ứng dụng của đề tài không những áp dụng cho các tòa nhà mà còn được mở rộng áp dụng đối với điều khiển các thiết bị điện sử dụng nơi công cộng và trong cả sản xuất công nghiệp.

TÀI LIỆU THAM KHẢO

1. Tài liệu Tiếng Việt

[1]. Nguyễn Văn Thắng, Nguyễn Trọng Đức, “Ứng dụng Internet of Things xây dựng ngôi nhà thông minh”, Đề tài NCKH cấp Trường, Trường Đại học Hàng hải, 2016.

2. Tài liệu Tiếng Anh

[2]. A.K. Dennis, Raspberry Pi Home Automation with Arduino, Packt Publishing, USA, 2013.

[3]. Mohamed Fezari, Internet of Things (IOT) Using Raspberry Pi, Annaba University, 2019.

3. Trang Web tham khảo

[4]. <https://cloud.google.com/iot/docs/how-tos/getting-started>

[5]. <https://cloud.google.com/community/tutorials/cloud-iot-gateways-rpi>

[6]. <https://cloud.google.com/bigquery/docs/quickstarts/quickstart-web-ui>

[7]. <https://cloud.google.com/bi-engine/docs/getting-started-data-studio>

[8]. <https://cloud.google.com/community/tutorials/cloud-iot-firestore-config>

[9]. <https://firebase.google.com/docs/functions/get-started>

[10]. <https://firebase.google.com/docs/web/setup>

[11]. <https://iot-analytics.com/5-things-know-about-iot-platform>

[12]. <https://medium.com/free-code-camp/gcp-cloudiotcore-esp32-mongooseos-1st-5c88d8134ac7>

Tp. HCM, ngày 15 tháng 07 năm 2020

LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên 1: Lê Bá Chuẩn MSSV: 16044161 LỚP: DHCNTT 12B

Họ tên sinh viên 2: Lê Anh Tú MSSV: 16048431 LỚP: DHCNTT 12B

Tên đề tài: Cloud IoT, BigQuery, DataStudio và ứng dụng mô hình nhà thông minh.

Tuần	Nội dung thực hiện	Xác nhận GVHD
Tuần 1 (02/3 – 08/3)	Gặp Giáo viên hướng dẫn và chọn đề tài.	
Tuần 2 (09/3 – 15/3)	Tìm hiểu các đề tài nghiên cứu có liên quan.	
Tuần 3 (16/3 – 22/3)	Gặp và báo cáo với giáo viên hướng dẫn về đề tài sẽ thực hiện.	
Tuần 4 (23/3 – 29/3)	Tìm hiểu về Raspberry, Arduino và các Module liên quan.	
Tuần 5 (30/3 – 5/4)	Tìm hiểu cách thức hoạt động, giao tiếp, truyền dữ liệu giữa các thành phần hệ thống: Raspberry, Arduino, Sensor,...	
Tuần 6,7,8 (6/4 – 26/4)	Nghỉ do dịch Covid 19. Trao đổi và liên lạc qua Số điện thoại và Email. Tìm hiểu về Google Cloud IOT Core, Firebase . Cách thức giao tiếp, trao đổi dữ liệu giữa các thiết bị trong hệ thống với Cloud.	
Tuần 9 (27/4 – 3/5)	Tìm hiểu về ngôn ngữ: C++, Python. Viết chương trình xử lý dữ liệu từ các thiết bị, led và sensor.	
Tuần 10 (04/5 – 10/5)	Lập trình cho Raspberry để có thể kết nối và nhận tín hiệu từ Google Cloud, Firebase. Thiết kế mô hình nhà thông minh.	
Tuần 11 (11/5 – 17/2)	Báo cáo tiến độ với Giáo viên hướng dẫn. Tìm hiểu về ngôn ngữ Html, Css, Javascript để xây dựng ứng dụng Website.	
Tuần 12 (18/5 – 24/5)	Tổng hợp chương trình xử lý, đọc dữ liệu từ các thiết bị, giao tiếp giữa các module, truyền nhận dữ	

	liệu với Goolge IoT Core.	
Tuần 13 (25/5 – 31/5)	Thiết kế website, giao diện hiển thị và điều khiển thiết bị. Thực hiện kết nối với Firebase để có thể gửi, nhận dữ liệu. Báo cáo tiến độ với Giáo viên hướng dẫn.	
Tuần 14 (01/6 – 7/6)	Chạy thử hệ thống, kiểm tra và sửa lỗi. Tối ưu lại code, xử lý để hệ thống hoạt động ổn định và nhanh chóng hơn. Báo cáo, thực hiện demo với Giáo viên hướng dẫn.	
Tuần 15 (8/6 – 14/6)	Hoàn thiện mô hình và viết báo cáo.	

Tp. HCM, ngày..... tháng..... năm 2020

Giáo viên hướng dẫn