

**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**  
**KHOA ĐIỆN – ĐIỆN TỬ**  
**BỘ MÔN TỰ ĐỘNG**

-----o0o-----



## **BÀI TẬP LỚN**

# **KẾT NỐI SMARTPHONE XÂY DỰNG GIAO DIỆN GIAO TIẾP VỚI VI XỬ LÝ QUA WIFI**

**GVHD: TS NGUYỄN TRỌNG TÀI**

**SVTH:**

- |                         |                |
|-------------------------|----------------|
| <b>1. Tạ Hữu Nhân</b>   | <b>1712447</b> |
| <b>2. Đỗ Trường Duy</b> | <b>1710774</b> |

**TP. HỒ CHÍ MINH, THÁNG 8 NĂM 2020**

## ***LỜI CẢM ƠN***

Đầu tiên, chúng em xin gửi lời cảm ơn chân thành, sâu sắc đến các thầy, cô đang là giảng viên tại trường Đại học Bách Khoa Thành phố Hồ Chí Minh. Tiếp theo là đến các thầy trong khoa Điện – Điện tử đã tạo điều kiện để chúng em học tập, nghiên cứu những kiến thức cơ bản nhất về ngành Tự động, và đặc biệt là thầy Nguyễn Trọng Tài – Giảng viên bộ môn Đo lường và điều khiển bằng máy tính, đã hướng dẫn, giúp đỡ chúng em hoàn thành Bài tập lớn này.

Trong quá trình thực hiện, kiểm nghiệm cũng như làm báo cáo bài tập lớn, do trình độ lý luận và kinh nghiệm thực tiễn còn hạn chế nên không thể tránh khỏi những sai sót. Chúng em rất mong nhận được sự đóng góp chân thành đầy quý giá của các thầy, cô để giúp chúng em có thêm kinh nghiệm và hoàn thành bài tập lớn một cách tốt nhất.

*Tp. Hồ Chí Minh, ngày 2 tháng 8 năm 2019 .*

**Sinh viên**

Tạ Hữu Nhân  
Đỗ Trường Duy

## TÓM TẮT

Báo cáo này trình bày về “Giao tiếp Wifi dùng ESP 8266, kết nối với STM32F4 và Smartphone thông qua database Firebase”. Nguyên lý hoạt động cơ bản: Dữ liệu sẽ được gửi từ ứng dụng của Smartphone lên Firebase, EPS8266 sẽ kết nối Firebase và lấy dữ liệu về, sau đó gửi xuống STM32F4 thông qua UART để đẩy tín hiệu ra các ngoại vi (cụ thể xem phần Nguyên lý hoạt động ở phía dưới)

Ngôn ngữ lập trình được sử dụng ở đây là ngôn ngữ C, kết hợp với thư viện HAL của STM32F4.

IDE sử dụng để lập trình là Arduino (cho ESP8266) và Keil C cho (STM32F407), ngoài ra còn sử dụng phần mềm Cube MX để generate một số đoạn code cấu hình cho STM

Sử dụng App Inventor 2 của MIT để xây dựng giao diện trên Smartphone

Database được sử dụng ở đây là Firebase, một database miễn phí của Google

Ngoài ra, để hoàn thành sản phẩm, chúng em còn sử dụng đến các ứng dụng nạp code, board nạp, dụng cụ làm mạch, linh kiện điện tử, breadboard cùng các loại dụng cụ khác.

## MỤC LỤC

1.	GIỚI THIỆU .....	8
1.1	Tổng quan .....	8
1.2	Nhiệm vụ đề tài .....	8
1.3	Phân chia công việc trong nhóm .....	8
2.	THIẾT KẾ.....	9
2.1	. Các phương án thiết kế: .....	9
2.1.1	. Phương án 1: .....	9
2.1.2	. Phương án 2: .....	9
2.1.3	. Phương án 3: .....	10
2.2	. Sơ đồ khối của mô hình .....	12
3.	THIẾT KẾ VÀ THỰC HIỆN APP ANDROID.....	13
3.1	. Firebase là gì ? .....	13
3.2	. Cách kết nối và các hàm Firebase trong App Inventor: .....	13
3.3	. Các hàm Firebase được sử dụng trong dự án này: .....	17
3.3.1	. FireBase.dataChanged: .....	17
3.3.2	. FireBase.GotValue: .....	18
3.3.3	. FireBase.GotValue: .....	18
3.3.4	. FireBase.StoreValue.....	18
3.4	. Giải thuật lập trình .....	18
3.5	. Tiến hành lập trình: .....	20
3.5.1	. Thiết kế giao diện:.....	20
3.5.2	. Lập trình các Block: .....	22
3.5.3	. Kết quả lập trình giao diện .....	24
3.5.4	. Đánh giá kết quả:.....	24
4.	THIẾT KẾ VÀ THỰC HIỆN FIRMWARE:.....	25

4.1	. Module wifi ESP8266:.....	25
4.1.1	. Nhiệm vụ của ESP8266: .....	26
4.1.2	. Cách khởi tạo: .....	26
4.1.3	. Các hàm Firebase được sử dụng: .....	26
4.1.3.1	. Firebase.begin(HOST, AUTH): .....	26
4.1.3.2	. Firebase.stream(/address):.....	27
4.1.3.3	. jb.createObject():.....	27
4.1.3.4	. Firebase.set(/address, đối_tượng) .....	27
4.1.3.5	. Firebase.setInt(/address/TypeA, biến) .....	27
4.1.3.6	. Firebase.getString(“key”) .....	27
4.1.4	. Giải thuật firmware: .....	28
4.2	. Vi xử lý STM32F411 .....	29
4.2.1	. Nhiệm vụ của STM32F4:.....	29
4.2.2	. Khởi tạo STM32F4: .....	30
4.2.3	. Giải thuật lập trình: .....	32
5.	THIẾT KẾ MÔ HÌNH PHẦN CỨNG, KẾT NỐI CẢM BIẾN:.....	33
5.1	. Các loại cảm biến:.....	33
5.1.1	. Cảm biến khí gas:.....	33
5.1.2	. Cảm biến chuyển động.....	34
5.1.3	. Cảm biến nhiệt độ độ ẩm: .....	34
5.2	. Các loại đèn: .....	35
5.3	. Xây dựng mô hình: .....	36
6.	KẾT QUẢ THỰC HIỆN.....	36
7.	ĐÁNH GIÁ CHẤT LƯỢNG:.....	36
8.	KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	37
8.1	Kết luận.....	37
8.2	Hướng phát triển .....	37

9. TÀI LIỆU THAM KHẢO.....	38
----------------------------	----

## DANH SÁCH HÌNH MINH HỌA

Hình 2-1. Sơ đồ khối.....	12
Hình 3-1. Firebase Realtime lưu trữ dạng Key value .....	13
Hình 3-2. Nhập Firebase Token và URL.....	14
Hình 3-3. Firebase URL.....	14
Hình 3-4. FireDB trong App Inventor .....	15
Hình 3-5. Firebase Token .....	15
Hình 3-6. Các block chức năng của Firebase part 1 .....	16
Hình 3-7. Các khối chức năng Firebase part 2 .....	17
Hình 3-8. Firebase.dataChanged.....	17
Hình 3-9. FireBase.GotValue .....	18
Hình 3-10. FireBase.GetValue .....	18
Hình 3-11. Firebase.StoreValue .....	18
Hình 3-12. Giải thuật lập trình App Android .....	19
Hình 3-13. Sreen Login .....	20
Hình 3-14. Screen Forgot Password.....	21
Hình 3-15. Screen Dashboard .....	21
Hình 3-16. Screen Contro.....	21
Hình 3-17. Block Login.....	22
Hình 3-18. Block Change Pass .....	22
Hình 3-19. Block Dashboard.....	23
Hình 3-20. Tổng quan các Screen.....	24
Hình 4-1. Module Wifi ESP8266.....	25
Hình 4-2. Sơ đồ khối của ESP8266 .....	26
Hình 4-3. Sơ đồ khối của ESP8266 .....	28
Hình 4-4. STM32F411.....	29
Hình 4-5. Giải thuật firmware ESP8266 .....	29
Hình 4-6. Cube MX cấu hình các chân.....	30

Hình 4-7. Các chức năng sử dụng trong STM.....	31
Hình 4-8. Giải thuật STM32F4.....	32
Hình 5-1. Cảm biến khí gas.....	33
Hình 5-2. Cảm biến chuyển động.....	34
Hình 5-3. Cảm biến nhiệt độ độ ẩm DHT12 .....	35
Hình 6-1. Kết quả thực hiện .....	36



## 1. GIỚI THIỆU

### 1.1 Tổng quan

Do sự phát triển bùng nổ của cuộc cách mạng công nghiệp 4.0, các nhu cầu về nhà ở thông minh (Smarthome) ngày càng tăng. Đã có rất nhiều tổ chức, công ty nhắm vào thị trường béo bở này. Tại Việt Nam cũng đã nhe nhóm những mô hình Nhà thông minh trong những khu đô thị, căn hộ cao cấp như Vinhomes,... Đây chắc chắn sẽ là một xu hướng mới của Việt Nam trong tương lai.

Nhận thấy được tính thiết thực và xu thế thời đại của chủ đề, nhóm em quyết định lấy mô hình Smarthome để làm mô hình ứng dụng Giao tiếp wifi thông qua Smartphone, ESP8266 và STM32F4.

### 1.2 Nhiệm vụ đề tài

Như đã trình bày ở phần trên, những yêu cầu cần phải đạt được về kích thước, tính năng và chất lượng sản phẩm. Để đạt được những yêu cầu đó, chúng em cần phải thực hiện các nhiệm vụ cụ thể như sau:

- Lên ý tưởng cho mô hình giao tiếp Wifi thông qua database
- Tìm hiểu về vi xử lí STM32F4, giao tiếp Wifi của Esp8266, App Inventor 2, Firebase, ...
- Xây dựng mô hình giao tiếp, kết nối các loại cảm biến
- Lắp đặt, nối dây các loại cảm biến trên mô hình.
- Hoàn thành mô hình và ứng dụng được Giao tiếp Wifi trên mô hình

### 1.3 Phân chia công việc trong nhóm

Công việc	Người thực hiện
Lập trình kết nối ESP với Firebase	Duy
Lập trình đọc cảm biến	Duy
Lập trình giao tiếp UART	Duy
Lập trình Ứng dụng Android	Nhân
Viết báo cáo, slide, video	Nhân, Duy
Làm mô hình	Nhân

Bảng 1-1. Bảng phân công nhiệm vụ

## 2. THIẾT KẾ

### 2.1 . Các phương án thiết kế:

#### 2.1.1 . Phương án 1:

Sử dụng Blynk để tạo App Android, dùng ESP8266 để đọc, ghi dữ liệu và quản lí ngoại vi

STT	Tên linh kiện	Giá thành
1	ESP8266	Có sẵn
2	Cảm biến gas MQ - 2	30.000đ
3	Cảm biến chuyển động PIR HC-SR501	20.000đ
4	Cảm biến nhiệt độ độ ẩm DHT12	50.000đ
5	Các linh kiện LED, dây dẫn	40.000đ
	Tổng	140.000đ

Bảng 2-1. Danh sách linh kiện phương án 1

Ưu điểm:

- Tiện dụng, nhỏ gọn, xu hướng IoTs
- Sử dụng module có sẵn, tiết kiệm chi phí

Nhược điểm:

- Sử dụng loại board mạch dạng Arduino, không sử dụng các loại VXL khác như Atmega, ARM, PIC
- Thư viện Blynk miễn phí rất ít, muốn giao tiếp nhiều thì cần trả phí

#### 2.1.2 . Phương án 2:

Sử dụng App Inventor để xây dựng app android, module ESP8266 và vi xử lí Pic để quản lí ngoại vi

STT	Tên linh kiện	Giá thành
1	ESP8266	Có sẵn
2	Vi xử lý PIC	50.000đ
3	Mạch nạp PIC	150.000đ
2	Cảm biến gas MQ - 2	30.000đ
3	Cảm biến chuyển động PIR HC-SR501	20.000đ
4	Cảm biến nhiệt độ độ ẩm DHT12	50.000đ
5	Các linh kiện LED, dây dẫn	40.000đ
	Tổng	340.000đ

Bảng 2-2. Danh sách linh kiện phương án 2

Ưu điểm:

- Tiện dụng, nhỏ gọn, xu hướng IoTs
- Sử dụng module có sẵn, tiết kiệm chi phí
- Có thành phần Vi xử lý PIC
- App Inventor hoàn toàn miễn phí, dùng thêm Firebase không tốn phí

Nhược điểm:

- Giá thành khá cao.
- Tốn nhiều chi phí cho mạch nạp

### 2.1.3 . Phương án 3:

Sử dụng App Inventor để xây dựng app android, module ESP8266 và vi xử lý STM32F407 để quản lý ngoại vi

STT	Tên linh kiện	Giá thành
1	ESP8266	Có sẵn
2	STM32F407	Có sẵn
3	Cảm biến gas MQ - 2	30.000đ
4	Cảm biến chuyển động PIR HC-SR501	20.000đ
5	Cảm biến nhiệt độ độ ẩm DHT12	50.000đ
6	Các linh kiện LED, dây dẫn	40.000đ
	Tổng	140.000đ

Bảng 2-3. Danh sách linh kiện phương án 3

Ưu điểm:

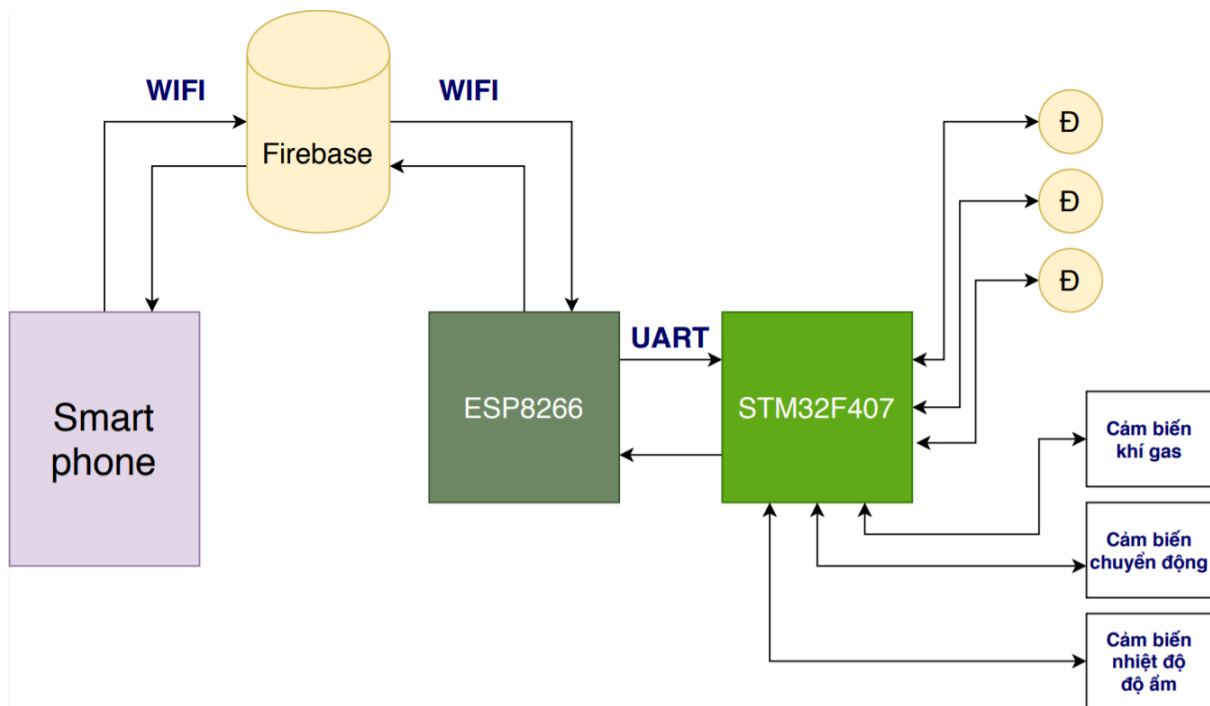
- Sử dụng module có sẵn, tiết kiệm chi phí
- Có thành phần Vi xử lý ARM
- App Inventor hoàn toàn miễn phí, dùng thêm Firebase không tốn phí

Nhược điểm:

- Kích thước board mạch khá lớn

⇒ **Nhìn chung, phương án 3 là phù hợp nhất**

## 2.2 . Sơ đồ khối của mô hình



Hình 2-1. Sơ đồ khối

- Smartphone: sử dụng giao diện đã lập trình trên App Inventor để gửi lệnh, đọc dữ liệu từ database. Trên App Inventor có sẵn các thư viện để đọc, ghi dữ liệu lên Firebase. Bên cạnh đó App Inventor hỗ trợ lập trình giao diện và các block chức năng, giúp dễ dàng tạo được ứng dụng Android với nhiều Screen, kèm với Password bảo mật cho việc truyền dữ liệu lên database.
- ESP8266 cũng sử dụng để gửi, nhận dữ liệu, xử lý dữ liệu nếu cần thiết. ESP8266 lúc này có vai trò như 1 module trung gian để chuyển từ kết nối có dây (UART) lên kết nối không dây (Wifi), module này có 2 công việc: Đọc dữ liệu từ Database, đóng thành frame truyền, gửi xuống cho STM32F4 để xử lý và đưa tín hiệu ra các đèn, quạt; Đọc dữ liệu từ STM32F4, tách frame truyền để có được giá trị của từng thành phần, ghi vào Database để Smartphone sử dụng.
- STM32F407 có nhiệm vụ quản lý các ngoại vi bên ngoài, đọc dữ liệu cảm biến, đọc dữ liệu đèn để gửi về ESP8266 hoặc nhận dữ liệu từ ESP8266 thông qua UART, xử lý chuỗi và thực hiện ra các đèn bên ngoài.

### 3. THIẾT KẾ VÀ THỰC HIỆN APP ANDROID

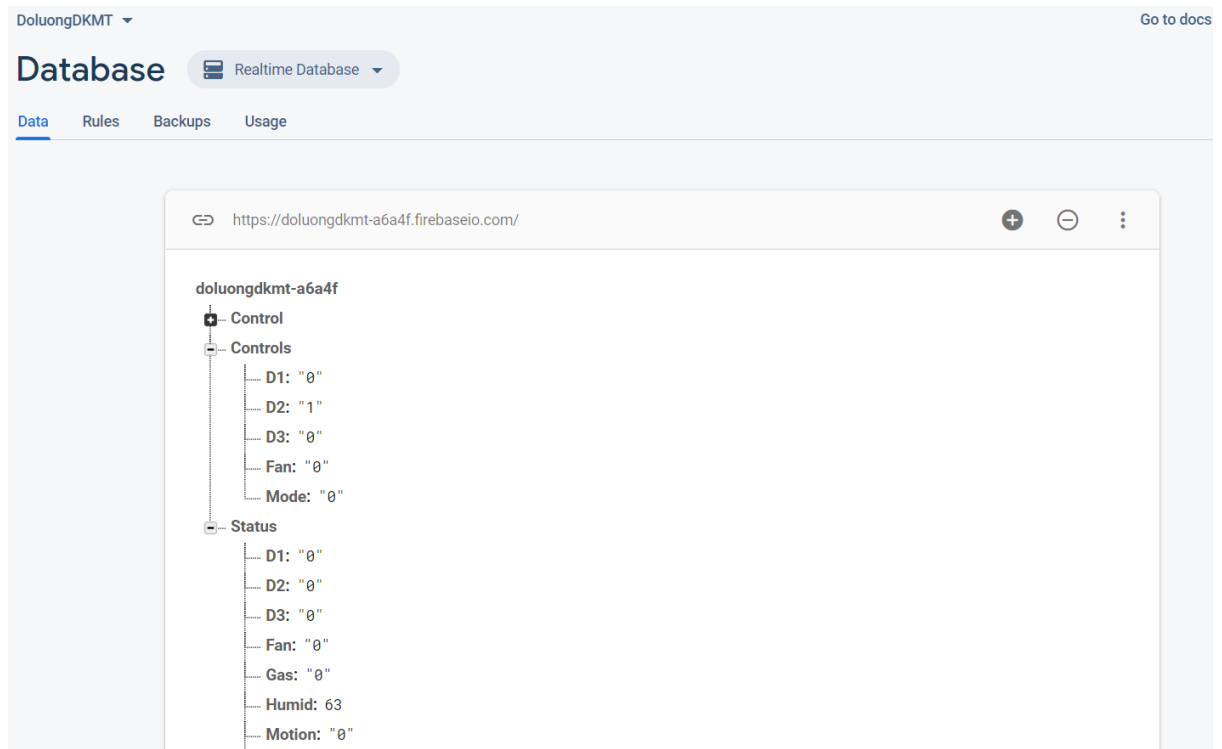
- Yêu cầu đặt ra cho phần mềm
  - Hoạt động ổn định, realtime trong thời gian quy định
  - Giao diện dễ nhìn, dễ kiểm soát và dễ dàng sử dụng
  - Hiển thị các cảnh báo cho người dùng khi hệ thống gặp trục trặc hoặc những thao tác của người dùng chưa đúng

#### 3.1 . Firebase là gì ?

Firebase là một nền tảng để phát triển ứng dụng di động và trang web, bao gồm các API đơn giản và mạnh mẽ mà không cần backend hay server.

Firebase được sử dụng trong nhiều ứng dụng Realtime cả trên Android và IOS vì những đặc tính mạnh mẽ của nó như API đa dạng mà lại sử dụng đơn giản, sử dụng máy chủ của Google nên bảo mật và an toàn.

Firebase Realtime lưu trữ dữ liệu dưới dạng NoSQL, tức là không theo dạng bảng và sử dụng truy vấn như SQL, mà lưu trữ dưới dạng Key – Value

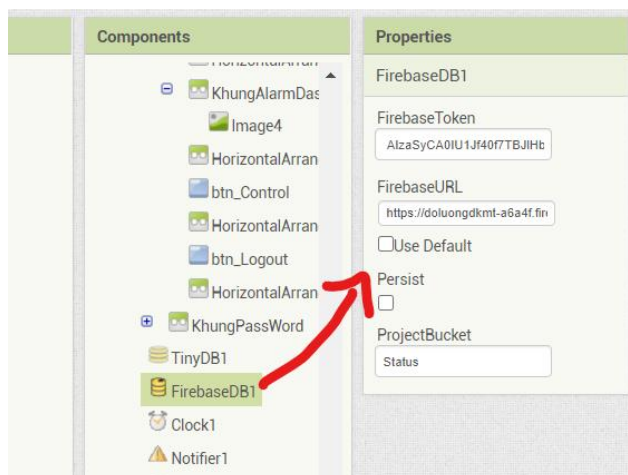


Hình 3-1. Firebase Realtime lưu trữ dạng Key value

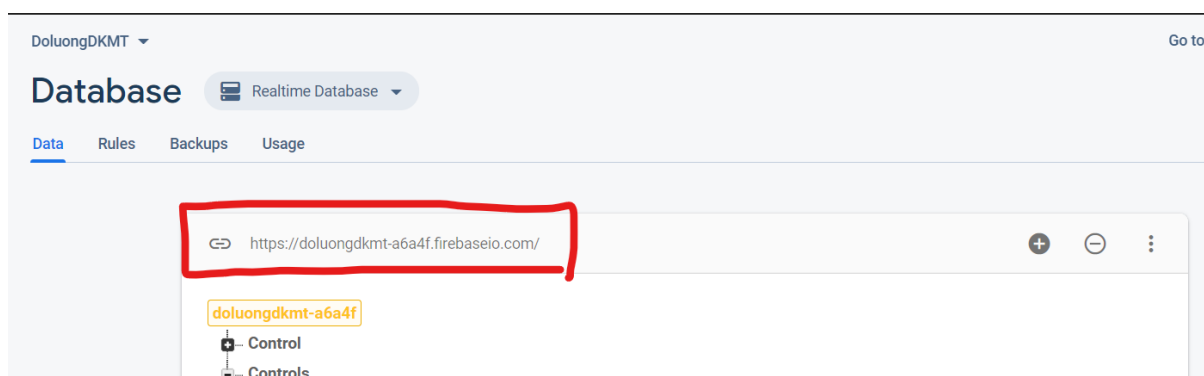
#### 3.2 . Cách kết nối và các hàm Firebase trong App Inventor:

Firebase là một Experimental của App Inventor, cho phép lưu trữ thông tin lên database trực tuyến.

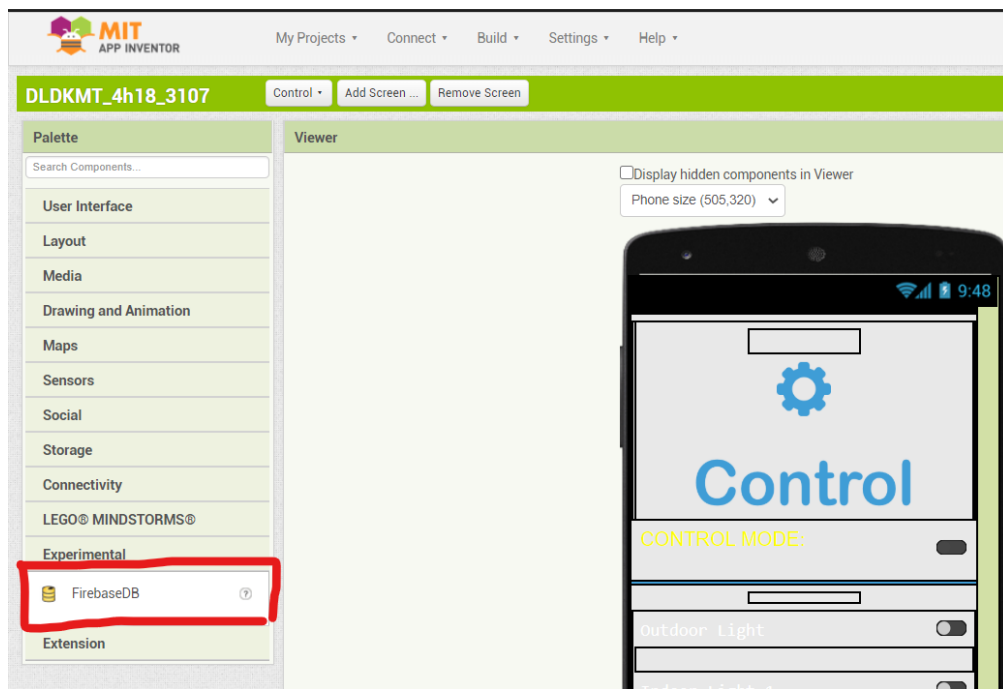
Để thiết lập kết nối từ App Inventor sang Firebase, ta cần nhập token và Firebase URL vào App Inventor.



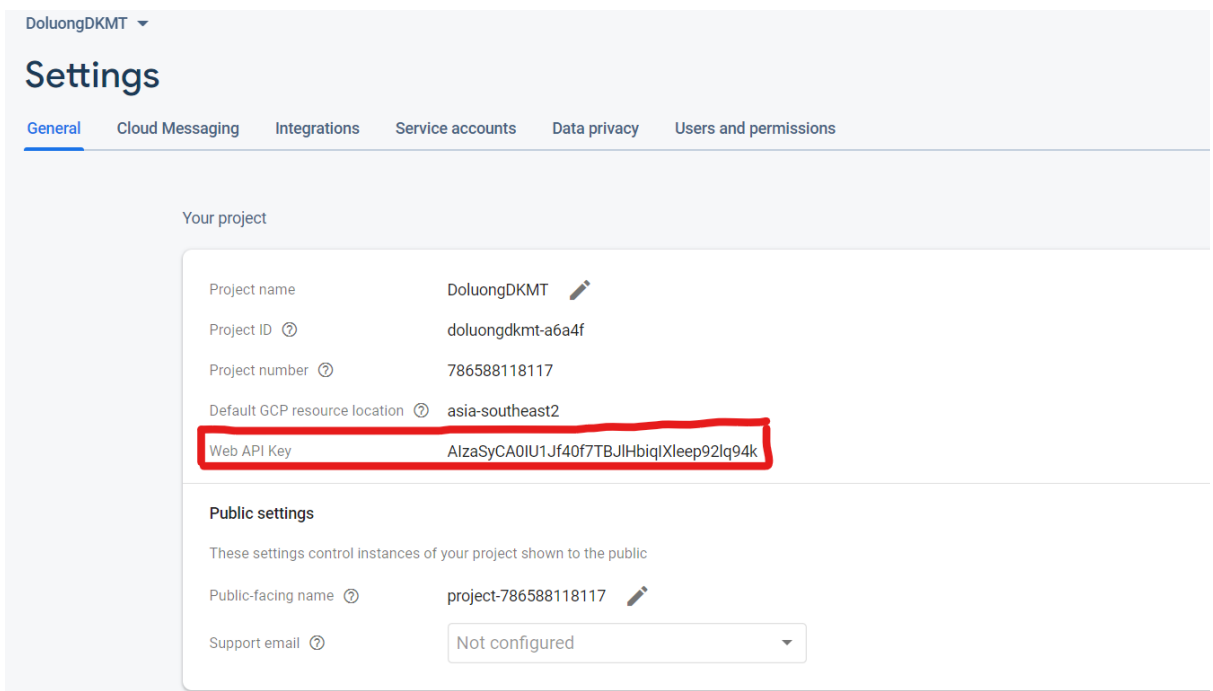
Hình 3-2. Nhập Firebase Token và URL



Hình 3-3. Firebase URL

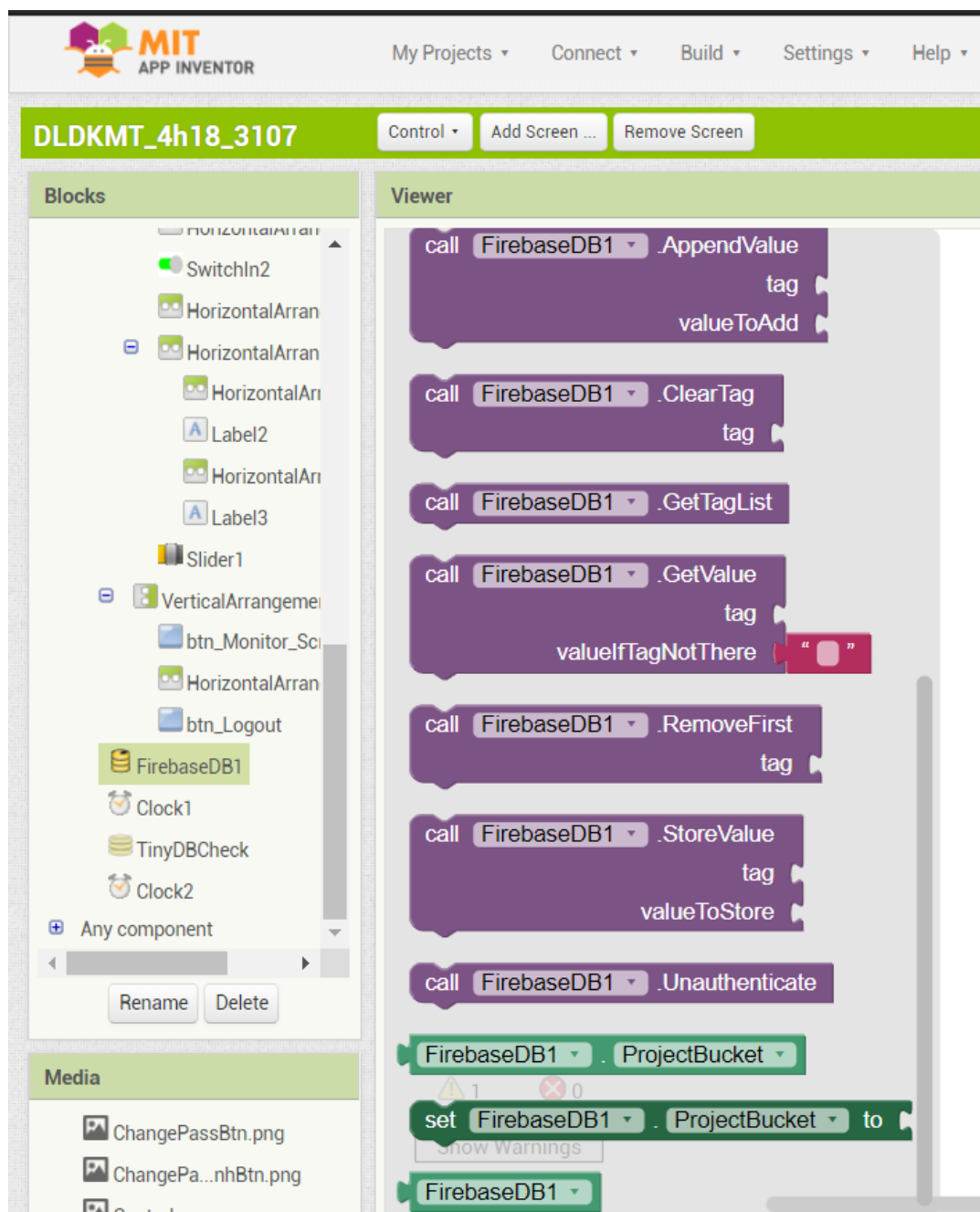


Hình 3-4. FireDB trong App Inventor



Hình 3-5. Firebase Token





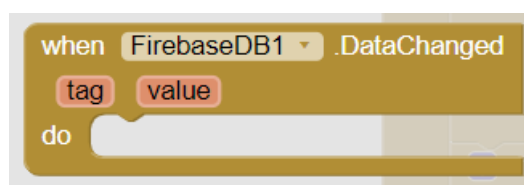
Hình 3-6. Các block chức năng của Firebase part 1



Hình 3-7. Các khối chức năng Firebase part 2

### 3.3 . Các hàm Firebase được sử dụng trong dự án này:

#### 3.3.1 . Firebase.dataChanged:

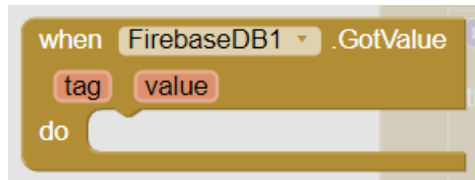


Hình 3-8. Firebase.dataChanged

Đây là một sự kiện của Firebase trong App Inventor, có tác dụng “get” “tag” và “value” vừa bị thay đổi trên Firebase và lưu vào bộ nhớ đệm.

Nếu muốn biết tag nào vừa bị thay đổi thì phải dùng câu lệnh “if else” hoặc “switch case” để biết được tag đó có tên gì.

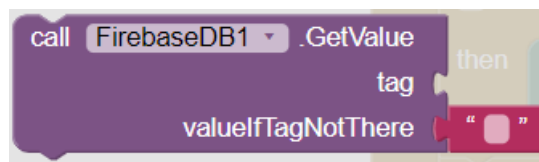
### 3.3.2 . Firebase.GotValue:



Hình 3-9. Firebase.GotValue

Đây là sự kiện “response” từ Firebase gửi xuống. Sự kiện response này xảy ra khi ta sử dụng hàm Firebase.getValue (trình bày bên dưới). Trong đó, “tag” là tên thẻ mà ta vừa yêu cầu đọc, “value” là giá trị tương ứng với thẻ mà ta vừa yêu cầu.

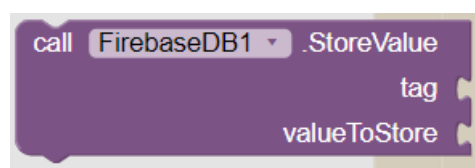
### 3.3.3 . Firebase.GotValue:



Hình 3-10. Firebase.GotValue

Đây là hàm chức năng, yêu cầu đọc giá trị của thẻ nào đó trên Firebase. Hàm này không trả về giá trị ngay lập tức mà tạo một sự kiện Firebase.GotValue (đã trình bày phía trên).

### 3.3.4 . Firebase.StoreValue

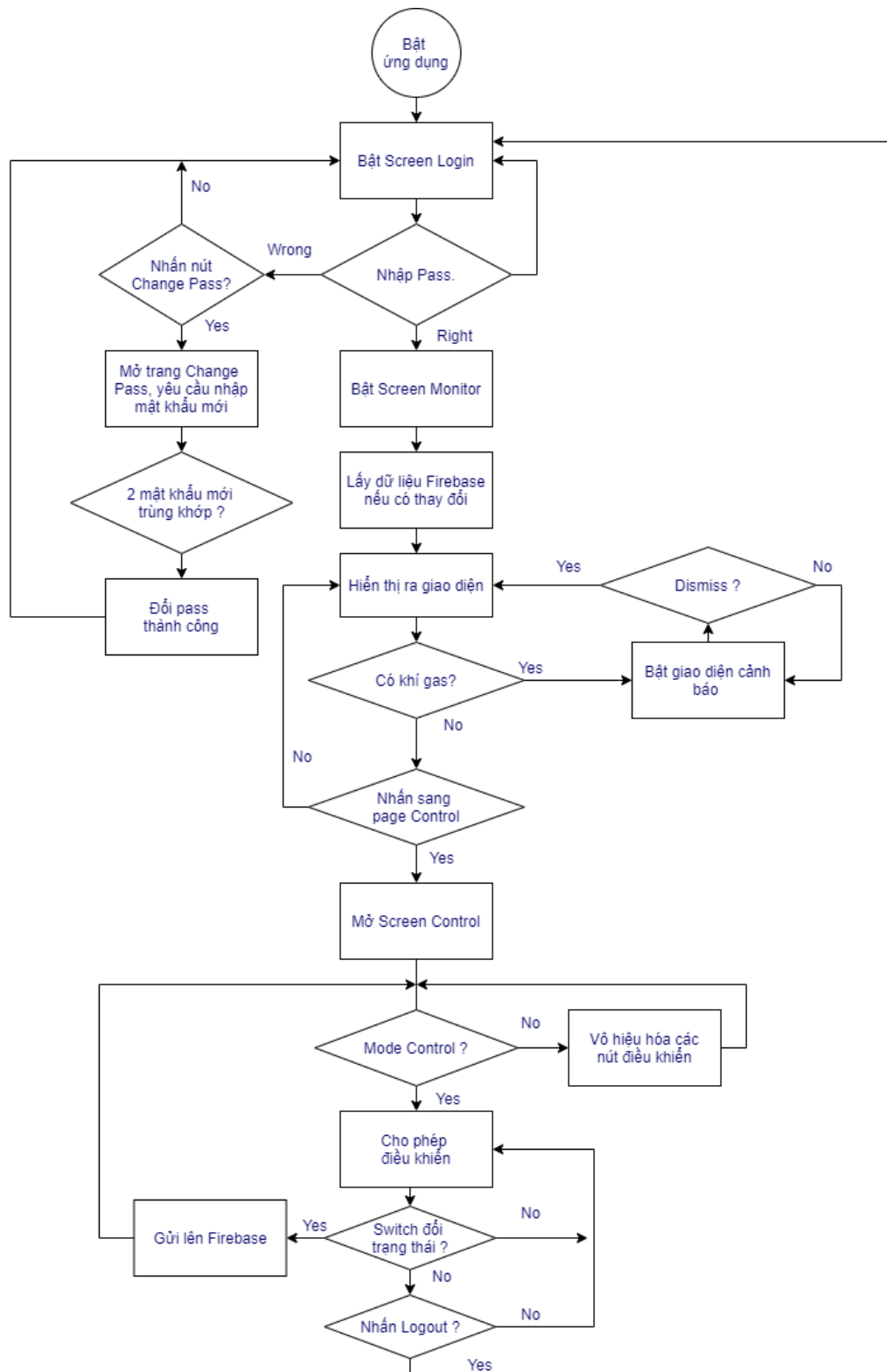


Hình 3-11. Firebase.StoreValue

## 3.4 . Giải thuật lập trình

Sau đây là giải thuật lập trình cơ bản mà nhóm sử dụng để viết app Android.

Nhóm mong muốn kết hợp nhiều Screen để phục vụ nhiều mục đích, có chức năng Login, quên mật khẩu, yêu cầu Mật khẩu khi muốn chuyển sang mode Điều khiển, cảnh báo lỗi khí gas bị rò, có màn hình giám sát tổng quan các thiết bị trong nhà....



Hình 3-12. Giải thuật lập trình App Android

### 3.5 . Tiến hành lập trình:

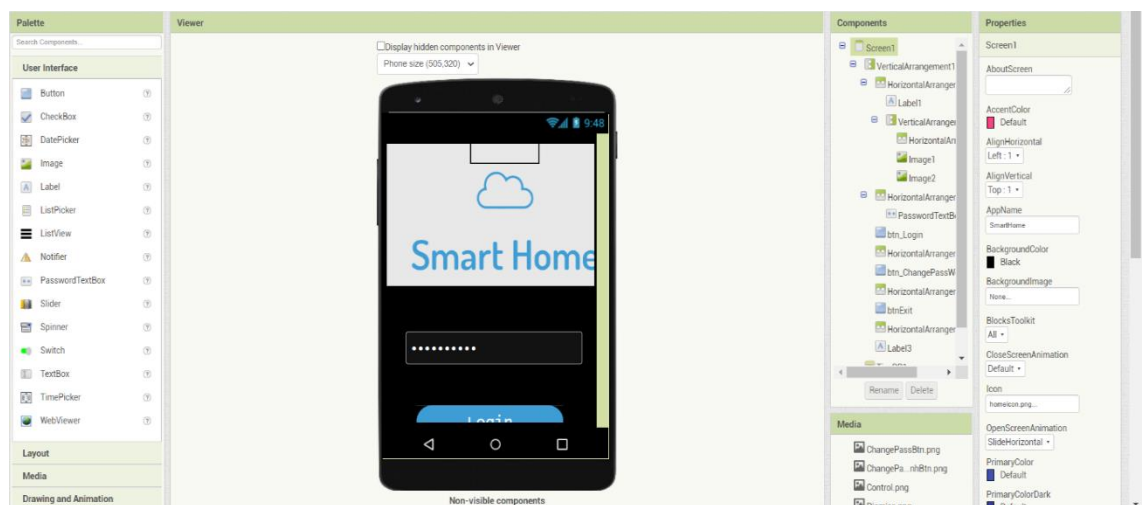
#### 3.5.1 . Thiết kế giao diện:

- Lên ý tưởng số lượng Screen, nội dung Screen, layout các Screen

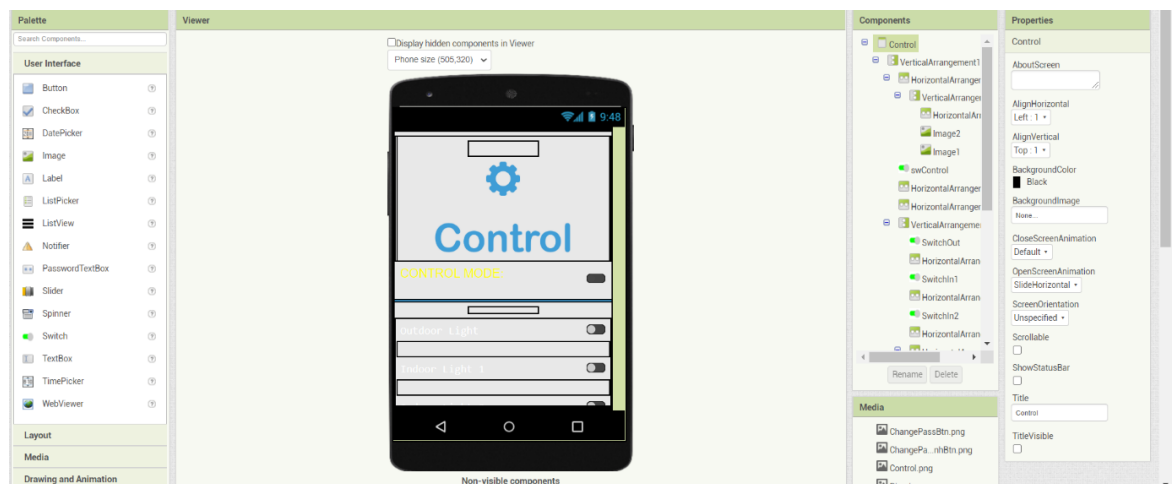
Screen	Nội dung
Login	Bắt buộc người dùng phải đăng nhập để được vào xem Dashboard. Có chức năng Change password Exit để thoát ứng dụng
ChangePass	Cho phép người dùng đổi mật khẩu, sau đó hướng người dùng về lại Trang Login với mật khẩu mới
Dashboard	Hiển thị toàn bộ thông tin, cho phép giám sát nhiệt độ, sensor, đèn. Có cảnh báo nếu có khí gas Cảnh báo thì database bị gián đoạn Để chuyển sang trang Control, cần nhập Password 1 lần nữa
Control	Thực hiện điều khiển thủ công các thiết bị đèn, quạt

Bảng 3-1. Ý tưởng cho các Screen

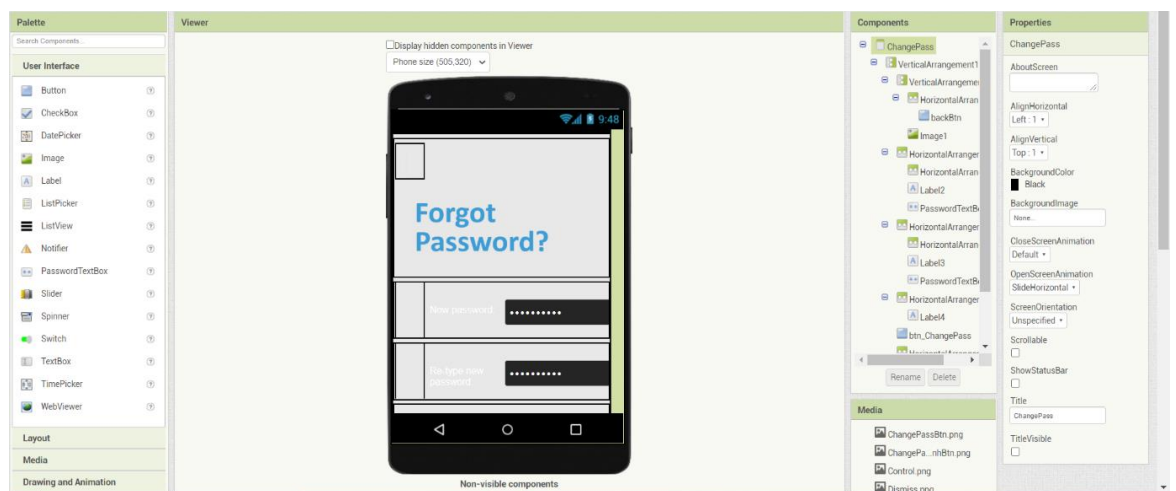
- Dùng các User Interface cho trước, kết hợp với các thuộc tính của Component để xây dựng app dựa trên layout và nội dung đã lên kế hoạch



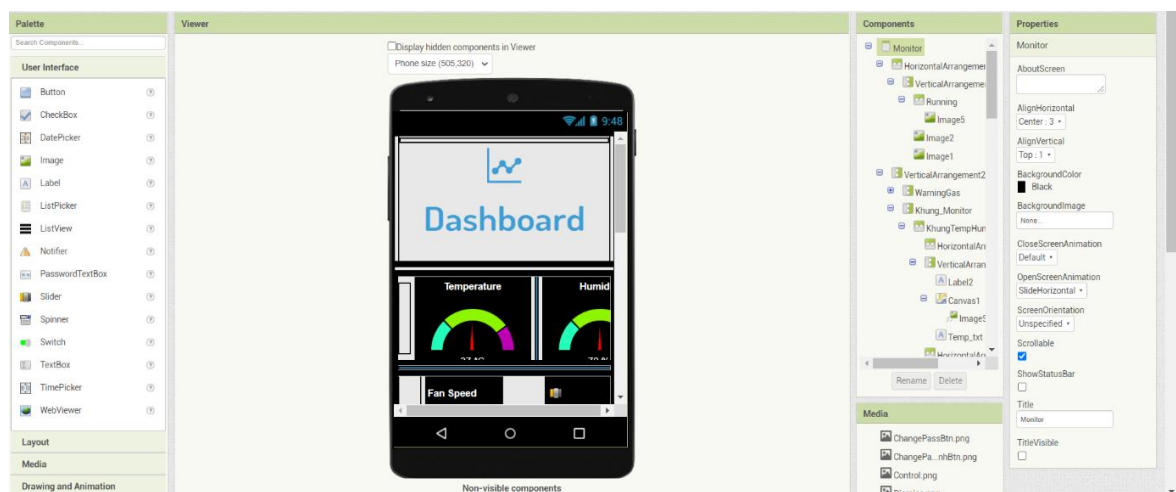
Hình 3-13. Sreen Login



Hình 3-14. Screen Forgot Password

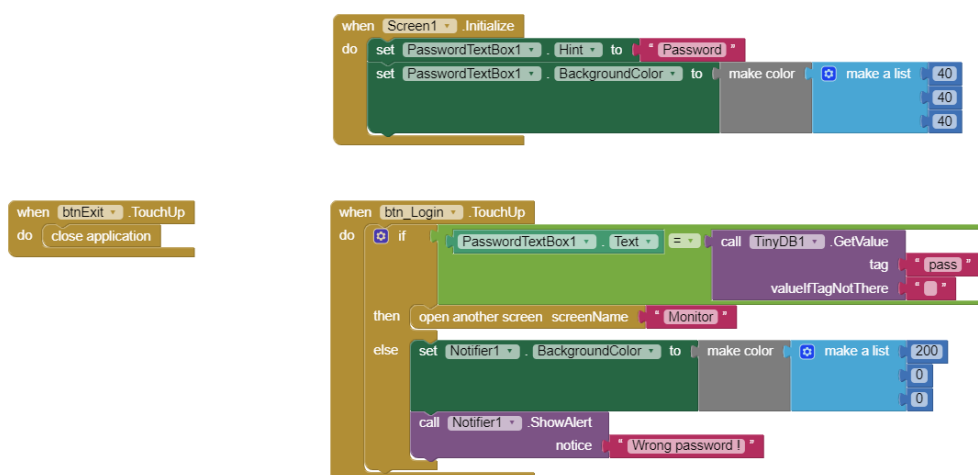


Hình 3-15. Screen Dashboard

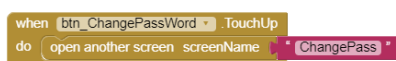


Hình 3-16. Screen Contro

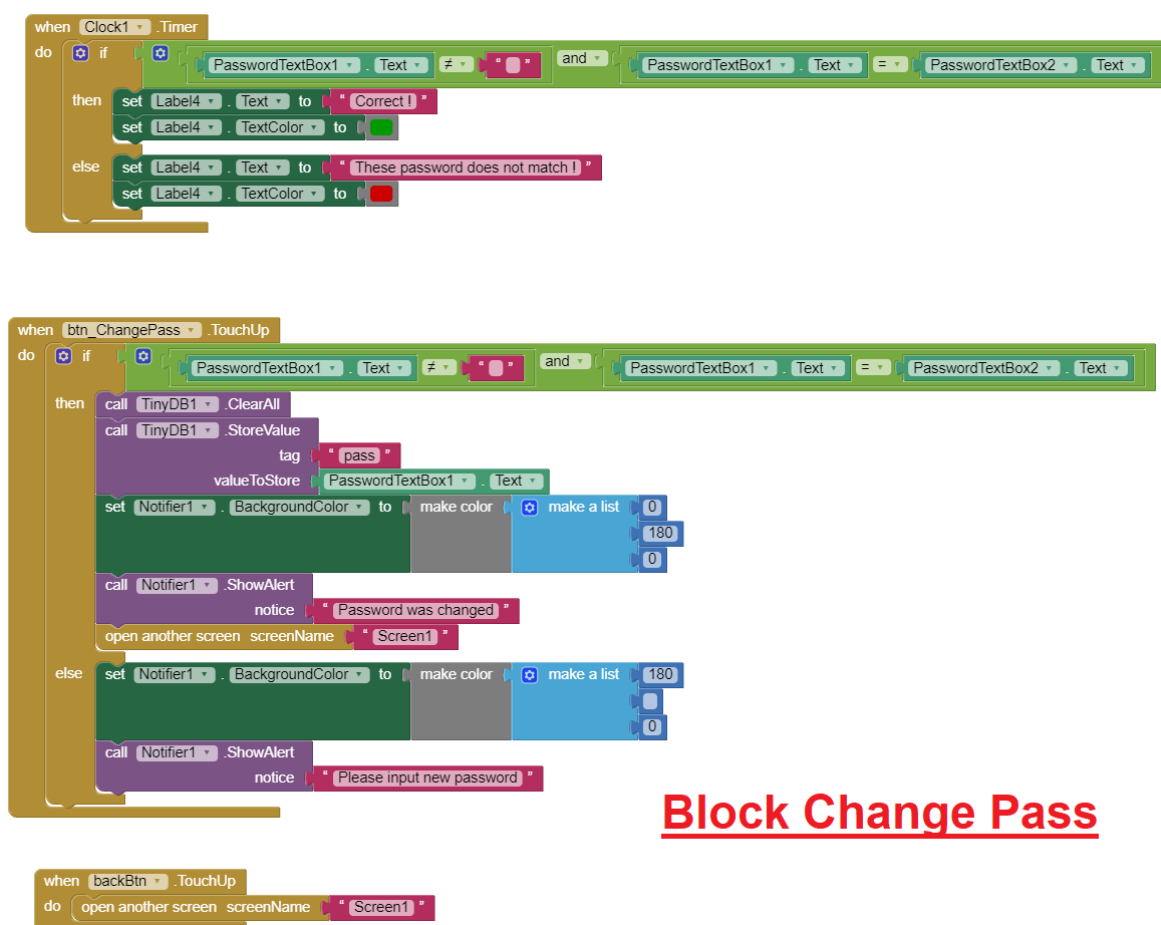
### 3.5.2 . Lập trình các Block:



### Block Login

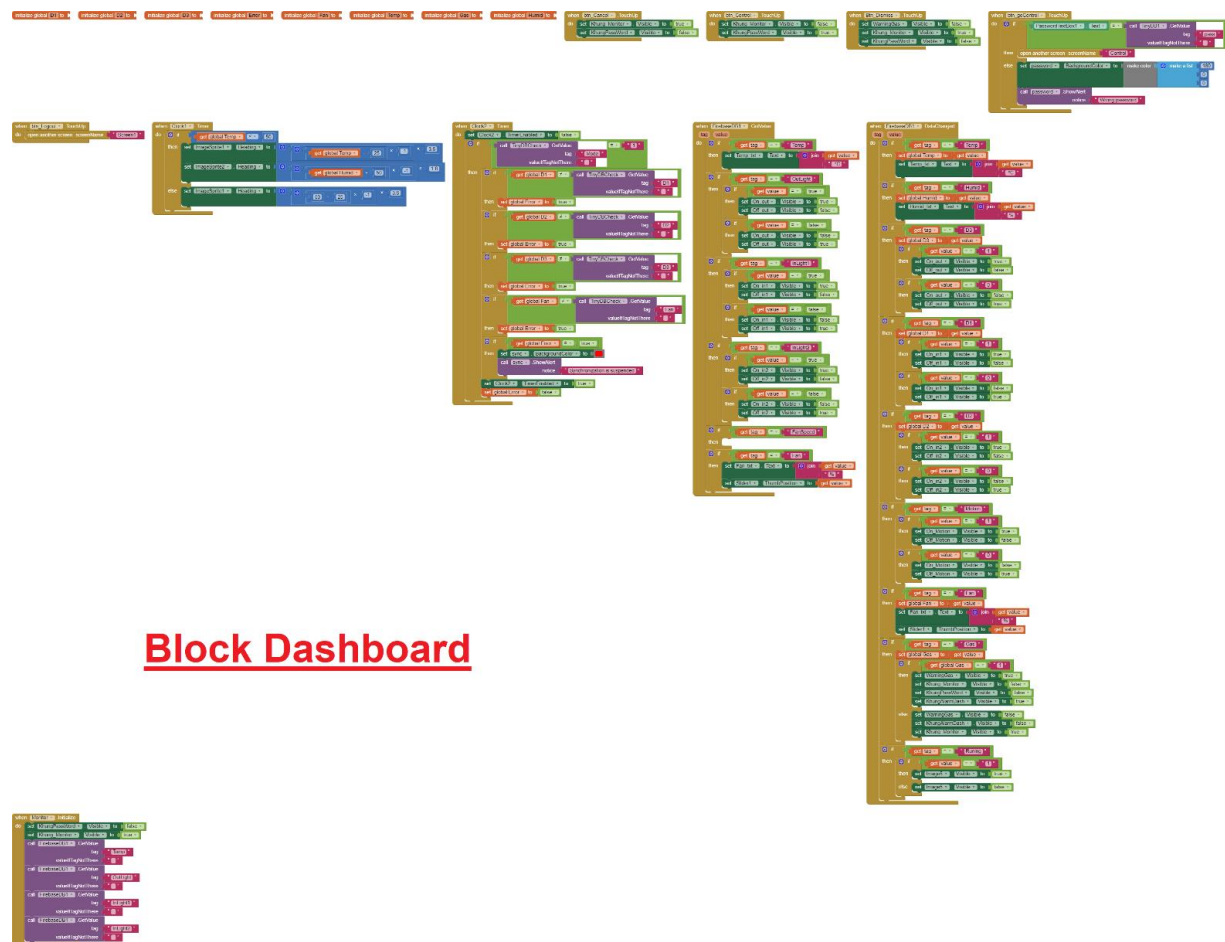


Hình 3-17. Block Login



### Block Change Pass

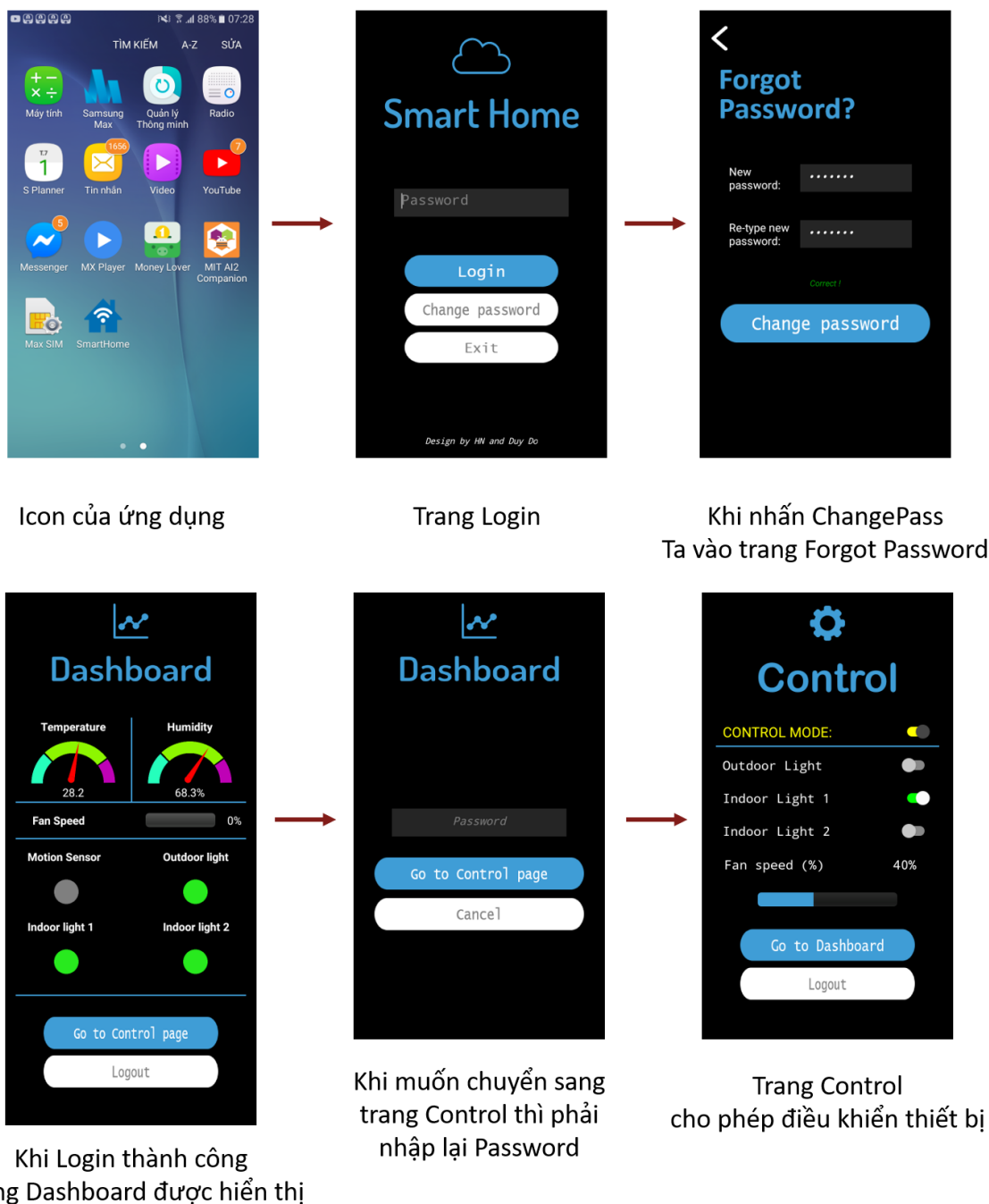
Hình 3-18. Block Change Pass



Hình 3-19. Block Dashboard



### 3.5.3 . Kết quả lập trình giao diện



Hình 3-20. Tổng quan các Screen

### 3.5.4 . Đánh giá kết quả:

**Ưu điểm:**

Kết nối thành công Firebase

Độc dữ liệu từ Firebase

Ghi dữ liệu lên Firebase

Giao diện mượt mà

Hiển thị với độ trễ < 1 giây

**Nhược điểm:**

Đôi lúc bị Crash.

Render bị sai lệch trong vòng 0.5 giây đầu tiên

## 4. THIẾT KẾ VÀ THỰC HIỆN FIRMWARE:

- **Yêu cầu thiết kế chung:**

- Thực hiện tốt các nhiệm vụ đọc, ghi dữ liệu về Firebase
- Thực hiện tốt việc truyền nhận giữa ESP8266 và STM32F4
- Sai số và độ trễ của hệ thống trong thời gian cho phép

### 4.1 . Module wifi ESP8266:

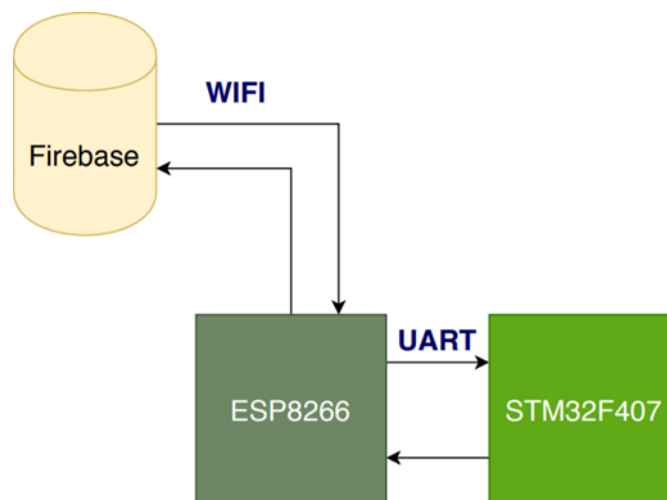


Hình 4-1. Module Wifi ESP8266

Đây là một dòng chip tích hợp Wifi 2,4GHz và có khả năng lập trình được (tuy nhiên trong đề tài này, ta xem ESP8266 là một module Wifi và là trung gian truyền nhận dữ liệu giữa database và STM32F4)

#### 4.1.1 . Nhiệm vụ của ESP8266:

- Đọc dữ liệu từ Firebase theo chu kỳ.
- Xử lý dữ liệu → tạo thành Frame truyền và truyền UART xuống cho STM32F4
- Nhận dữ liệu từ STM32F4 → tách frame truyền
- Ghi các giá trị vừa tách ra lên Firebase.



Hình 4-2. Sơ đồ khối của ESP8266

#### 4.1.2 . Cách khởi tạo:

- Để giao tiếp với Firebase, ta dùng thư viện được cung cấp sẵn bởi cộng đồng Arduino trên thế giới. Trước hết ta phải tải thư viện “Firebase” và “ArduinoJSON”
- Sử dụng hàm `Firebase.begin` để kết nối tới Firebase dựa vào URL

#### 4.1.3 . Các hàm Firebase được sử dụng:

##### 4.1.3.1. `Firebase.begin(HOST, AUTH):`

*`Firebase.begin (FIREBASE_HOST, FIREBASE_AUTH);`*

**Trong đó:** `FIREBASE_HOST` là FirebaseURL (đã trình bày ở trên mục 3)

`FIREBASE_AUTH` là thông tin xác nhận

**Mục đích** của hàm này là để ESP8266 connect đến Firebase mà mình muốn

#### 4.1.3.2. `Firebase.stream(/address):`

**Trong đó:** /address chính là tên group nằm trong database của bạn

**Mục đích:** Dùng để bắt sự kiện có thay đổi dữ liệu trong group /address trên database của bạn

#### 4.1.3.3. `jb.createObject():`

**Cú pháp:** `JsonObject& tên_biến = jb.createObject();`

**Mục đích:** tạo một đối tượng để truy xuất bằng ESP8266, sau đó có thể gửi cả đối tượng này lên Database nhanh chóng

#### 4.1.3.4. `Firebase.set(/address, đối_tượng)`

**Mục đích:** Gửi cả đối tượng (đã khai báo ở dạng JsonObject) vào group “address” trên database.

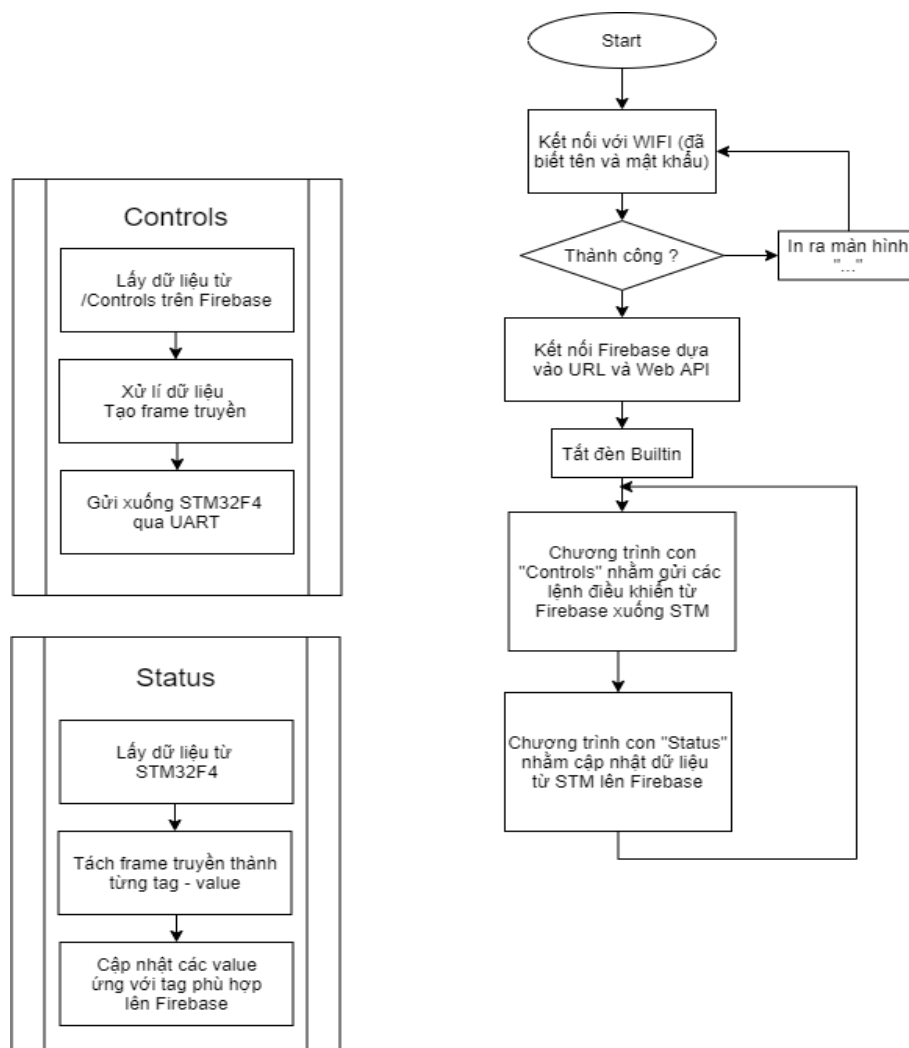
#### 4.1.3.5. `Firebase.setInt(/address/TypeA, biến)`

**Mục đích:** Gửi biến dạng Int vào tag TypeA thuộc group address trên database.

#### 4.1.3.6. `Firebase.getString(“key”)`

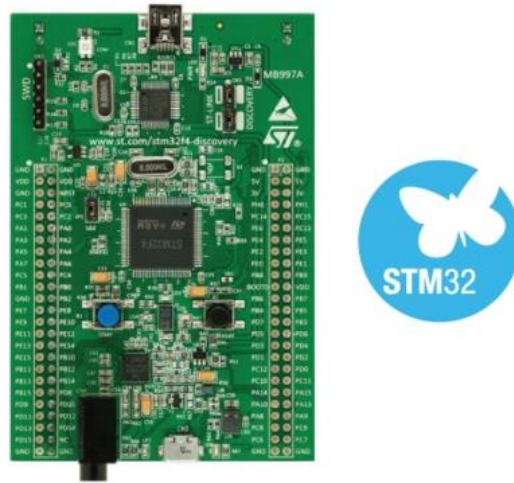
**Mục đích:** lấy giá trị kiểu string từ tag “key” trên database và lưu vào bộ nhớ.

#### 4.1.4 . Giải thuật firmware:



Hình 4-3. Sơ đồ khối của ESP8266

## 4.2 . Vi xử lí STM32F411

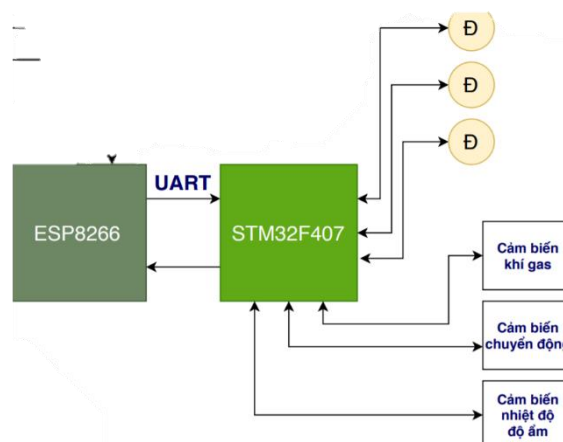


Hình 4-4. STM32F411

- **Yêu cầu thiết kế chung:**
  - Thực hiện tốt việc đọc các cảm biến, xuất tín hiệu ra các đèn
  - Thực hiện tốt việc truyền nhận với ESP8266
  - Sai số và độ trễ trong mức cho phép

### 4.2.1 . Nhiệm vụ của STM32F4:

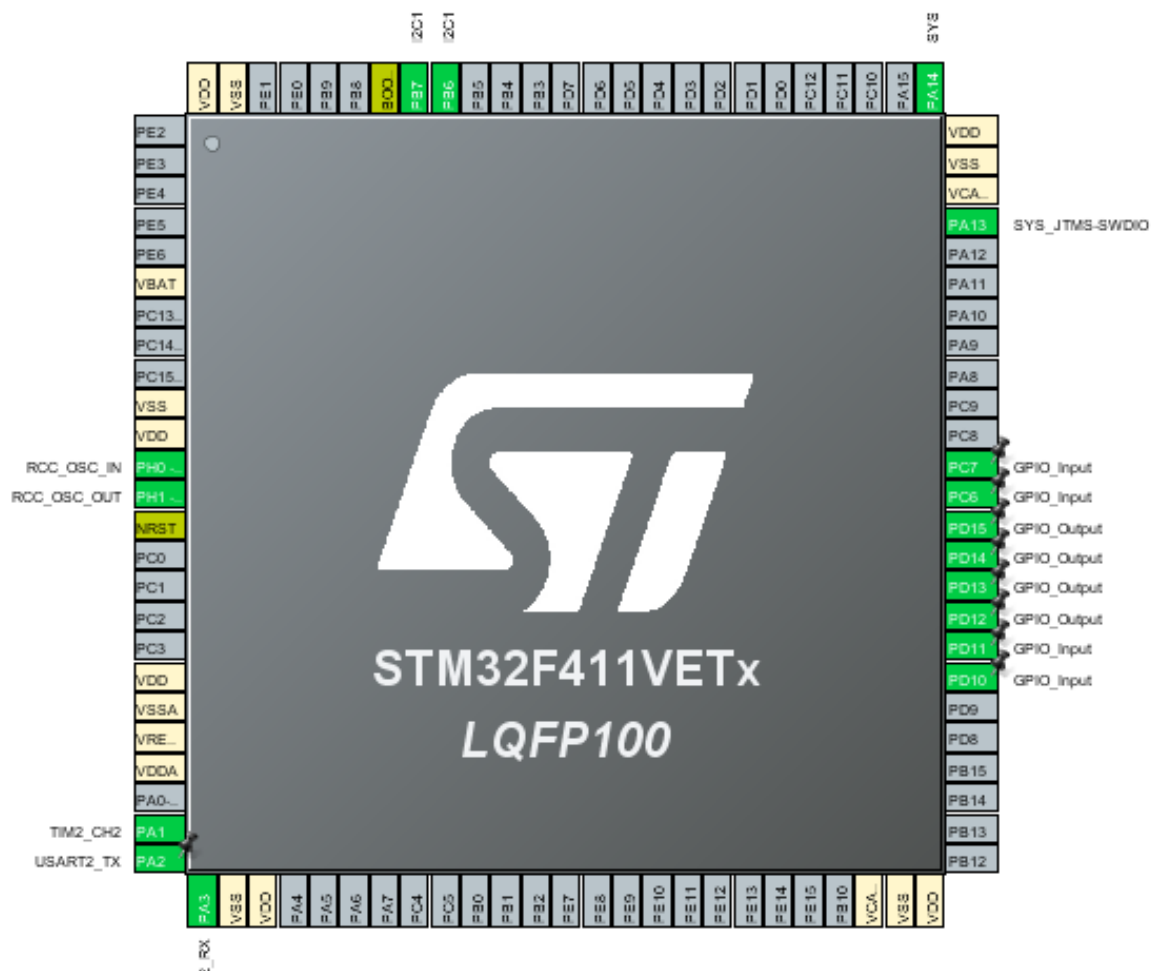
- Đọc các giá trị từ Cảm biến chuyển động, Cảm biến khí gas, Cảm biến nhiệt độ ẩm DHT12
- Xuất tín hiệu DO ra các đèn, xuất tín hiệu PWM ra đèn Dimmer
- Truyền nhận UART với ESP8266



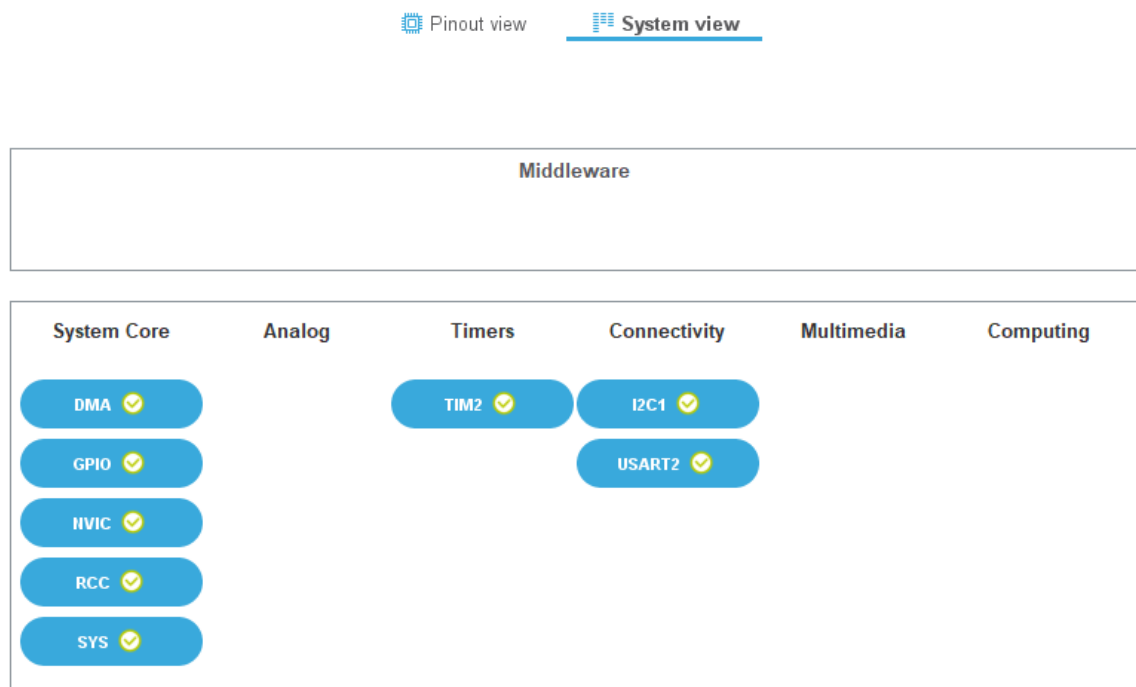
Hình 4-5. Giải thuật firmware ESP8266

#### 4.2.2 . Khởi tạo STM32F4:

- Sử dụng phần mềm Cube MX để cấu hình chức năng các chân theo mục đích của đề tài.



Hình 4-6. Cube MX cấu hình các chân



Hình 4-7. Các chức năng sử dụng trong STM

**Các chức năng cần được sử dụng gồm:**

- I2C
- GPIO
- UART
- PWM



### 4.2.3 . Giải thuật lập trình:



Hình 4-8. Giải thuật STM32F4

Trong chương trình chính, STM32F4 sẽ đọc giá trị các cảm biến và gửi giá trị đó lên ESP8266 bằng UART 100ms một lần. Khung truyền dữ liệu lên ESP gồm 11 byte:

's'	Byte 0 đọc DHT12	','	Byte 1 đọc DHT12	','	Byte 2 đọc DHT12	','	Byte 3 đọc DHT12	Trạng thái các đèn, cảm biến khí gas, chuyển động	Tốc độ quạt	Byte kết thúc
-----	------------------------	-----	------------------------	-----	------------------------	-----	------------------------	---	-------------------	---------------------

Khi ESP8266 gửi lệnh điều khiển bằng UART xuống, STM32F4 sẽ thực hiện các lệnh điều khiển trong ngắt. Khung nhận lệnh từ ESP có 4 byte:

'C'	Trạng thái điều khiển các đèn	Tốc độ quạt	'E'
-----	----------------------------------	-------------	-----

## 5. THIẾT KẾ MÔ HÌNH PHẦN CỨNG, KẾT NỐI CẢM BIẾN:

### 5.1 . Các loại cảm biến:

#### 5.1.1 . Cảm biến khí gas:



Hình 5-1. Cảm biến khí gas

Cảm biến khí gas MQ-2 sử dụng phân tử  $\text{SnO}_2$  có độ dẫn điện thấp hơn trong không khí sạch, khi khí dễ cháy tồn tại, cảm biến có độ dẫn điện cao hơn, nồng độ chất dễ cháy càng cao thì độ dẫn điện của  $\text{SnO}_2$  sẽ càng cao và được tương ứng chuyển đổi thành mức tín hiệu điện.

Cảm biến khí gas MQ-2 là cảm biến khí có độ nhạy cao với LPG, Propane và Hydrogen, mê-tan ( $\text{CH}_4$ ) và hơi dễ bắt lửa khác, với chi phí thấp và phù hợp cho các ứng dụng khác nhau.

Cảm biến xuất ra cả hai dạng tín hiệu là Analog và Digital, tín hiệu Digital có thể điều chỉnh mức báo bằng biến trở.

#### THÔNG SỐ KỸ THUẬT

- Nguồn hoạt động: 5V
- Loại dữ liệu: Analog hoặc Digital
- Phạm vi phát hiện rộng
- Tốc độ phản hồi nhanh và độ nhạy cao

### 5.1.2 . Cảm biến chuyển động



Hình 5-2. Cảm biến chuyển động

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 (con người, con vật, các vật phát nhiệt,...), 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.

#### THÔNG SỐ KỸ THUẬT

- Phạm vi phát hiện: góc 360 độ hình nón, độ xa tối đa 6m.
- Nhiệt độ hoạt động: 32-122 ° F ( 050 ° C)
- Điện áp hoạt động: DC 3.8V – 5V
- Mức tiêu thụ dòng:  $\leq 50 \mu A$
- Thời gian báo: 30 giây có thể tùy chỉnh bằng biến trở.
- Độ nhạy có thể điều chỉnh bằng biến trở.
- Kích thước: 1,27 x 0,96 x 1.0 ( 32,2 x 24,3 x 25,4 mm)

### 5.1.3 . Cảm biến nhiệt độ độ ẩm:

DHT12 là cảm biến nhiệt độ và độ ẩm, phiên bản nâng cấp của DHT11. Cung cấp đầy đủ tính năng tương thích với DHT11, nhưng có độ chính xác cao hơn và có thể giao tiếp I2C.



Hình 5-3. Cảm biến nhiệt độ độ ẩm DHT12

Tích hợp cảm biến độ ẩm điện dung và cảm biến nhiệt độ có độ chính xác cao, đầu ra tín hiệu số có thể kết nối với một Vi điều khiển 8-bit. Kích thước nhỏ, điện năng tiêu thụ cực thấp, khoảng cách truyền dẫn, hiệu chuẩn hoàn toàn tự động, sử dụng các cảm biến độ ẩm điện dung.

#### **Thông số kỹ thuật**

- Chuẩn giao tiếp: I2C, 1-wire
- Tầm đo nhiệt độ: -20 ~ +60 C
- Tầm đo độ ẩm: 20-95 RH
- Sai số nhiệt độ:  $\pm 0.5^{\circ}\text{C}$
- Sai số độ ẩm:  $\pm 0.1\text{RH}$
- Nguồn cấp: DC 2.7-5.5V
- Dòng hoạt động: 1mA
- Dòng tĩnh: 60uA

#### **5.2 . Các loại đèn:**

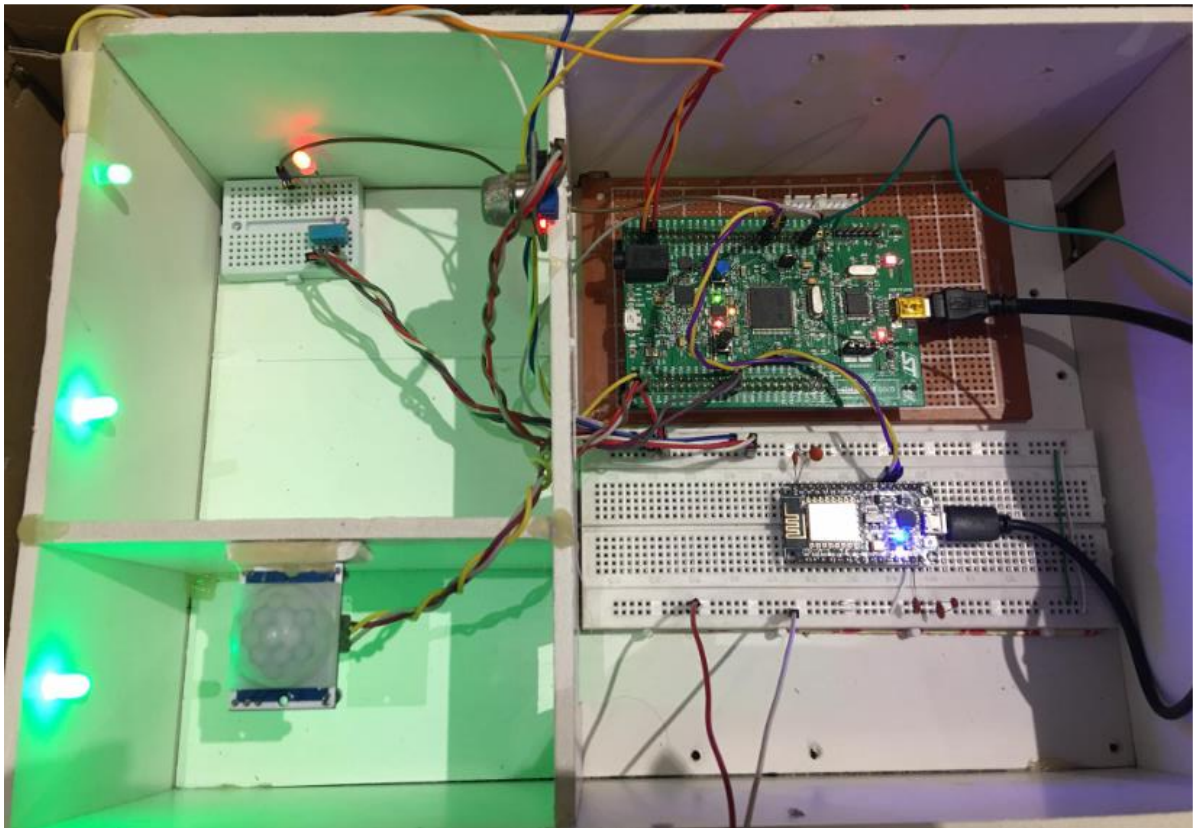
Gồm có 4 đèn:

- 3 đèn DO làm chức năng hiển thị trạng thái ON/OFF
- 1 đèn làm chức năng Dimmer (có thể chỉnh được độ sáng bằng cách dùng PWM)

### 5.3 . Xây dựng mô hình:

Chọn mô hình nhà thông minh - “Smart Home” để ứng dụng kết nối Smartphone wifi và vi xử lí, điều khiển các đèn và đọc các cảm biến.

## 6. KẾT QUẢ THỰC HIỆN



Hình 6-1. Kết quả thực hiện

## 7. ĐÁNH GIÁ CHẤT LƯỢNG:

### Ưu điểm:

- Hoàn thành mục tiêu đặt ra ở mức “Ổn”
- Hệ thống kết nối được Wifi, truyền nhận dữ liệu ổn định, ít lỗi
- Cảm biến nhiệt độ độ ẩm và khí gas hoạt động tốt, dữ liệu ổn định
- Ứng dụng hoàn thiện, dễ nhìn, dễ thao tác và có các cảnh báo lỗi

**Nhược điểm:**

- Hệ thống dễ bị treo, trạng thái treo tầm 10 giây, khi đó dữ liệu truyền nhận bị gián đoạn.
- Mô hình cắm breadboard, chưa chắc chắn và thẩm mỹ

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

### **8.1 Kết luận**

Sản phẩm đáp ứng được mục tiêu đề ra ở mức “Ổn”. Đôi lúc bị crash, treo nhưng nhìn chung vẫn tương tác, điều khiển và giao tiếp tốt. Cập nhật dữ liệu và gửi dữ liệu chính xác, không bị sai lệch nhiều

### **8.2 Hướng phát triển**

Sản phẩm vẫn còn nhiều tiềm năng để phát triển bên cạnh việc cải thiện các hạn chế và tối ưu hóa hoạt động, tránh tình trạng crash, treo vì xử lý. Trong đó có thể kể đến là:

- Tăng số lượng cảm biến, đèn hoạt động được ở nhiều phòng. Gắn thêm Motor đóng mở cửa.
- Kết nối với SMS cho những cảnh báo khẩn để đáp ứng nhanh, dễ giải quyết vấn đề.
- Tích hợp realtime clock để phân nhiệm vụ cho từng khung thời gian khác nhau.
- Sử dụng database riêng để cải thiện hiệu suất, không bị delay do các API do người dùng khác viết.
- Xây dựng app Android bằng những phần mềm chuyên dụng hơn, do App Inventor không mang lại hiệu suất cao, dễ crash khi render quá nhiều phần tử trên Screen.

## 9. TÀI LIỆU THAM KHẢO

- [1] Shop điện tử Nshop, Hshop cung cấp các thông tin cảm biến
- [2] <http://arduino.vn/bai-viet/657-huong-dan-su-dung-appinventor-lap-trinh-ung-dung-dieu-khien-xe-qua-bluetooth-ma-khong>
- [3] <https://vietmachine.com.vn/tao-ung-dung-android-cho-du-an-arduino-bang-mit-app-inventor.html>
- [4] <https://rominirani.com/tutorial-mit-app-inventor-firebase-4be95051c325>
- [5] <https://community.appinventor.mit.edu/t/firebase-getvalue/6255>
- [6] <https://groups.google.com/forum/#!topic/mitappinventortest/zguKDn7KAdI>
- [7] <https://tapit.vn/tuong-tac-voi-firebase-realtime-database-su-dung-esp8266/>
- [8] <https://hocarm.org/esp8266-va-firebase/>
- [9] <https://advancecad.edu.vn/dieu-khien-led-bang-google-firebase-console-va-esp8266-nodemcu-iot/>
- [10] Reference Manual STM32F411