



PSG COLLEGE OF TECHNOLOGY
DEPARTMENT OF INSTRUMENTATION AND CONTROL SYSTEMS ENGINEERING

19U022 INDUSTRIAL INTERNET OF THINGS

Controlling Motor Operation using Mobile Phone



Overview

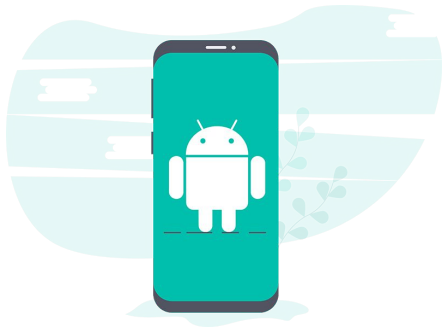
- Introduction
- Block diagram
- Components Used
- Programming
- Interfacing
- Conclusion
- References



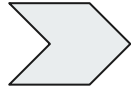
Introduction :

- Controlling home appliances using a smartphone has become a popular trend.
- With the rise of IoT (Internet of Things), many devices can now be connected to the internet, allowing for **remote control from anywhere in the world**.
- This project aims to use the Firebase platform, ESP8266 module, and an Android app to control the operation of a DC motor.

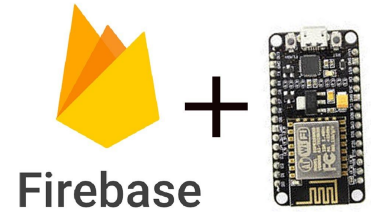
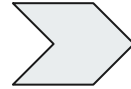
What i have done?



Created an
Android APP



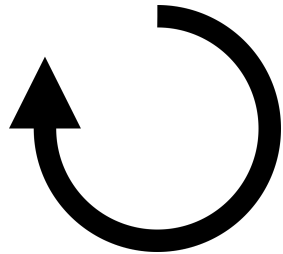
Integrated the APP with
Firebase



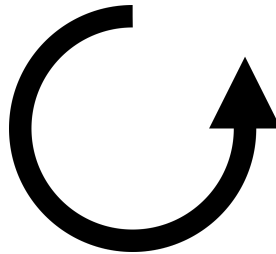
Used Firebase to control the
ESP8266 through Internet.

Controlling Operation of Motor:

Rotating
Clockwise
Direction



Rotating
Anti-Clockwise
Direction



Turning OFF the
motor

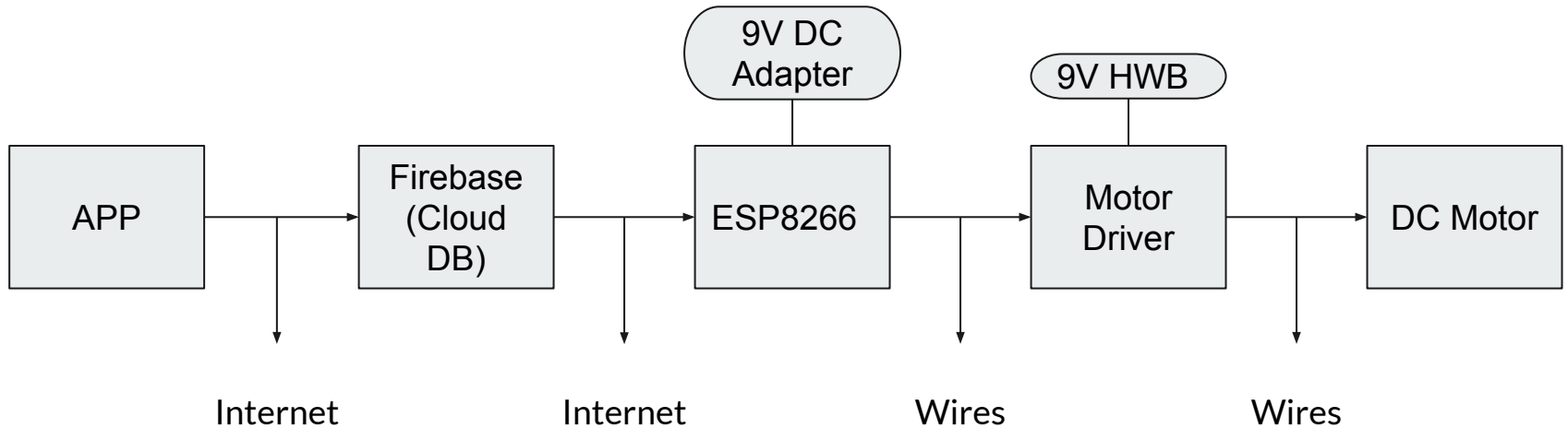




Components Used

- ESP8266(Node MCU)
- HG7881 (4-Channel) DC Motor Driver Board
- Micro 130 Small DC Motor
- 9V HW Battery and 9V DC Adapter for Power
Supply

Block Diagram

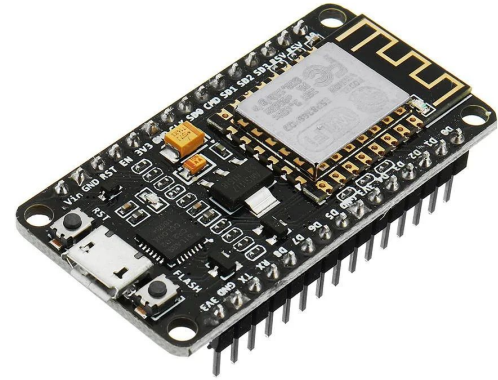




Hardware Components Used

ESP8266

- Small and low cost Wi-Fi enabled microcontroller.
- Produced by Espressif Systems, China.
- It supports full **TCP/IP** protocols.
- Ideal for IOT based projects.





Key Specification of Esp8266

2.4 GHz Wi-Fi
(802.11 b/g/n,
supporting
WPA/WPA2)

Tensilica L106
32-bit micro
controller unit at
80 MHz

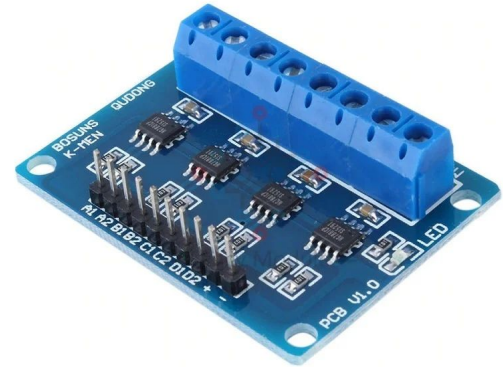
Micro USB port for
power,
programming and
debugging

13 GPIO pins, Flash
Memory 4Mb

3.3V operating
voltage, internal
voltage regulator
for consistent
voltage.

HG7881 4-Channel DC Motor Driver Board

- Module supply voltage: 2.5-12V
- It can drive 4 DC motors
- It is used to give high power to the motor by using a small voltage signal from a microcontroller .



Micro 130 Small DC Motor

Voltage : 3 to 6 Volts
Type : Brushed DC Motor
Motor Speed : 10,000 RPM-25,000 RPM

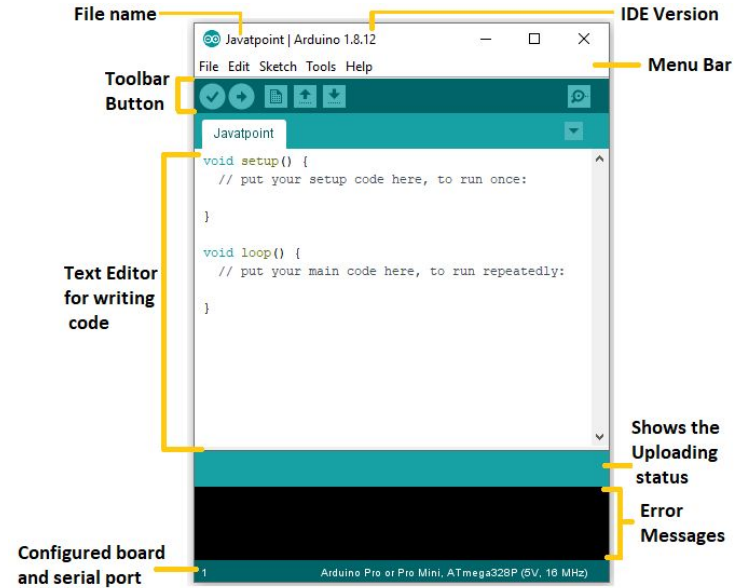




Softwares Used

Arduino IDE

- Open-source software, which is used to write and upload code to the microcontroller boards.
- Suitable for different operating systems such as Windows, Mac OS X, and Linux.
- Supports the programming languages **C** and **C++**



Android Studio IDE

- Android Studio is the official integrated development environment for Google's Android operating system.
- Android Studio Supports **Kotlin** and all version of **Java** languages.
- Used for Building , Debugging, and Simulating Application for All Android Based Devices .

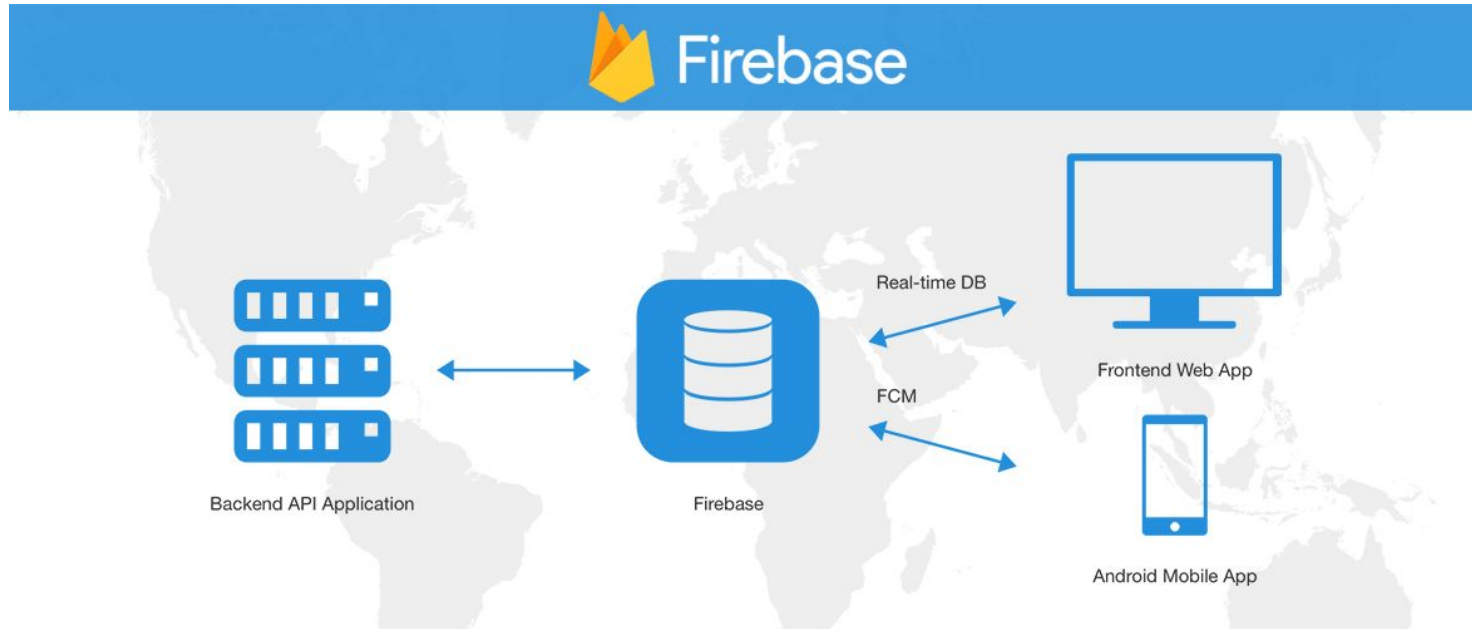




Firestore (Cloud Database)

- Firestore is a set of **backend cloud computing services** and application development platforms provided by Google.
- It hosts databases, services, authentication, and integration for a variety of applications.
- The Firestore Realtime Database applications allowing secure **access to the database directly** from client-side code.

Cont'd .,





Programming Part of Esp8266

Libraries Used :

ESP8266WiFi library :

Connecting ESP8266 module to a Wi-Fi network to start sending and receiving data.

FirestoreArduino.h :

Arduino library that show how to call the Firestore API from the ESP8266 Arduino core.



For Connecting Wifi network to Esp8266

```
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
```

Function for connecting wifi network.

```
Serial.println(WiFi.localIP());
```

Function for Printing IP address of the connected network .



Defining Required Value For Connecting with Firebase

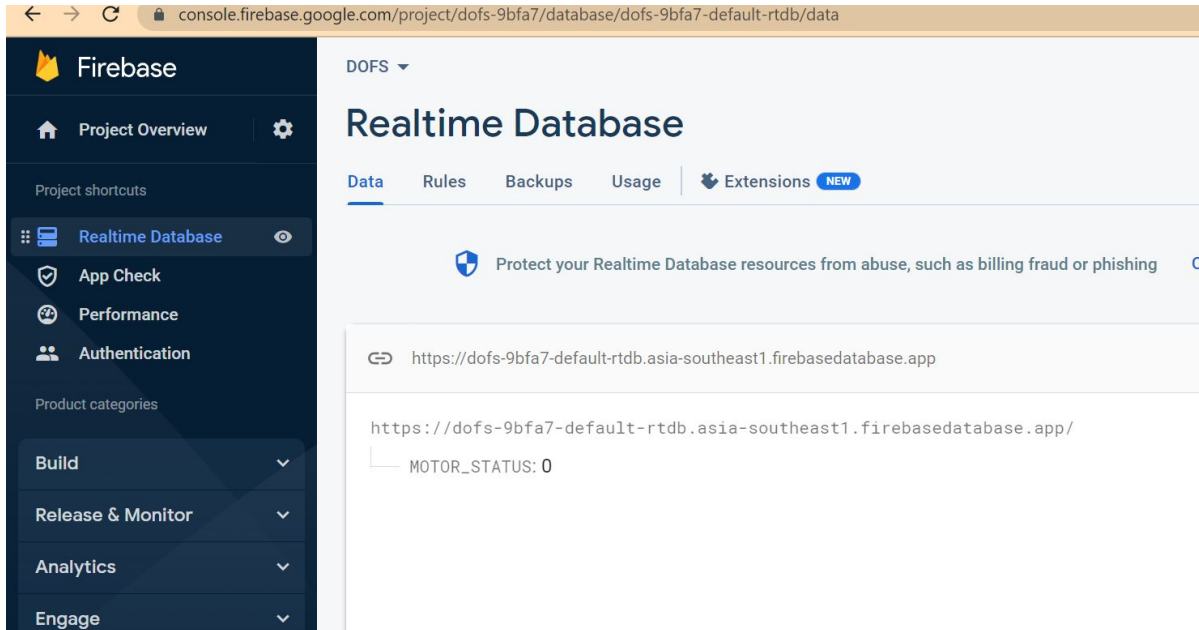
```
#define FIREBASE_HOST "dofs-9bfa7-default-rtdb.asia-southeast1.firebaseio.app" /
```

The URL Of the Firebase Data base

```
#define FIREBASE_AUTH "ekTBIBDnh2IBLhOM3Bx"
```

Authorisation Security Key for Accessing the database values

URL of Database



The screenshot shows the Firebase Realtime Database console. The left sidebar contains the Firebase logo, 'Project Overview', 'Project shortcuts' (with 'Realtime Database' selected), and 'Product categories' (including Build, Release & Monitor, Analytics, and Engage). The main content area is titled 'DOFS Realtime Database' and includes tabs for 'Data', 'Rules', 'Backups', 'Usage', and 'Extensions'. A security warning is displayed below the tabs. The 'Data' tab is active, showing a list of database entries. The first entry is a JSON object with the key 'MOTOR_STATUS' and the value '0'. The URL for this database is displayed as `https://dofs-9bfa7-default-rtdb.asia-southeast1.firebaseio.com/`.

← → ↺ console.firebase.google.com/project/dofs-9bfa7/database/dofs-9bfa7-default-rtdb/data

Firebase

Project Overview

Project shortcuts

- Realtime Database
- App Check
- Performance
- Authentication

Product categories

- Build
- Release & Monitor
- Analytics
- Engage

DOFS

Realtime Database

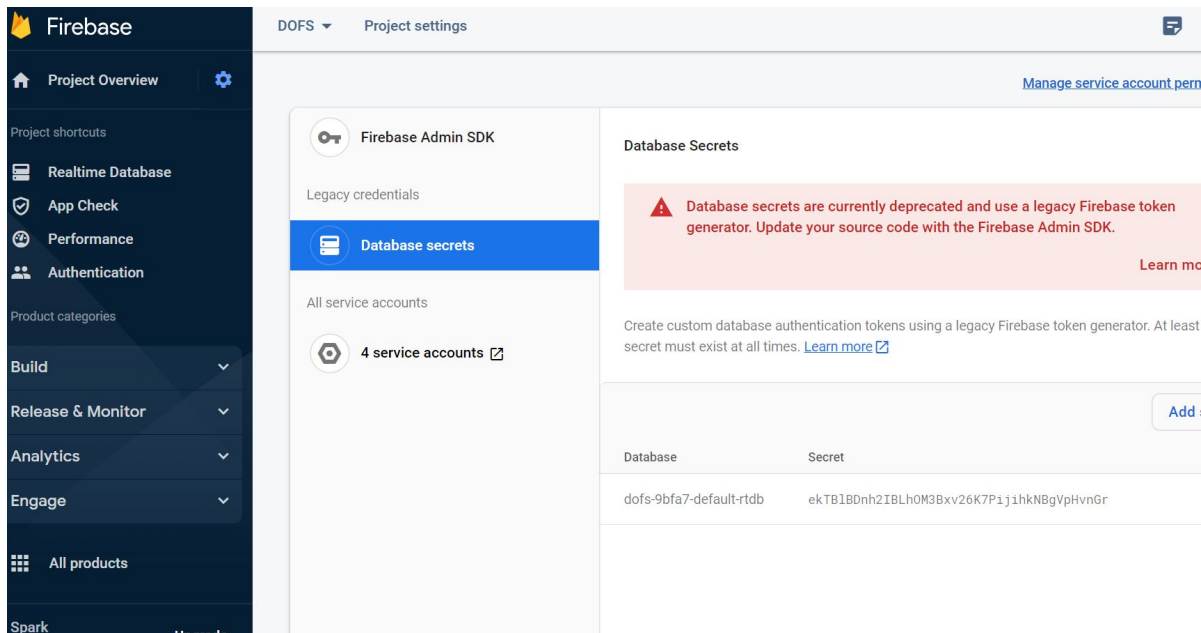
Data Rules Backups Usage Extensions **NEW**

Protect your Realtime Database resources from abuse, such as billing fraud or phishing

<https://dofs-9bfa7-default-rtdb.asia-southeast1.firebaseio.com/>

```
https://dofs-9bfa7-default-rtdb.asia-southeast1.firebaseio.com/  
└─ MOTOR_STATUS: 0
```

Database Secret Key



The screenshot shows the Firebase console interface. On the left is a dark sidebar with the 'Firebase' logo and navigation links: 'Project Overview', 'Project shortcuts' (Realtime Database, App Check, Performance, Authentication), 'Product categories' (Build, Release & Monitor, Analytics, Engage), and 'All products' (Spark). The top bar shows 'DOFS' and 'Project settings'. The main content area is titled 'Database secrets' and features a warning message: 'Database secrets are currently deprecated and use a legacy Firebase token generator. Update your source code with the Firebase Admin SDK.' Below the warning, it says 'Create custom database authentication tokens using a legacy Firebase token generator. At least one secret must exist at all times. Learn more'. A table lists the existing secrets:

Database	Secret
dofs-9bfa7-default-rtddb	ekTB1B0nh2I8Lh0M3Bxv26K7P1j1hkNBgVpHvN6r



Retrieving Data From Firebase :

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

Connecting with Created Firebase Database

```
fireStatus = Firebase.getInt("MOTOR_STATUS");
```

For Getting the Status value and Storing it on fireStatus



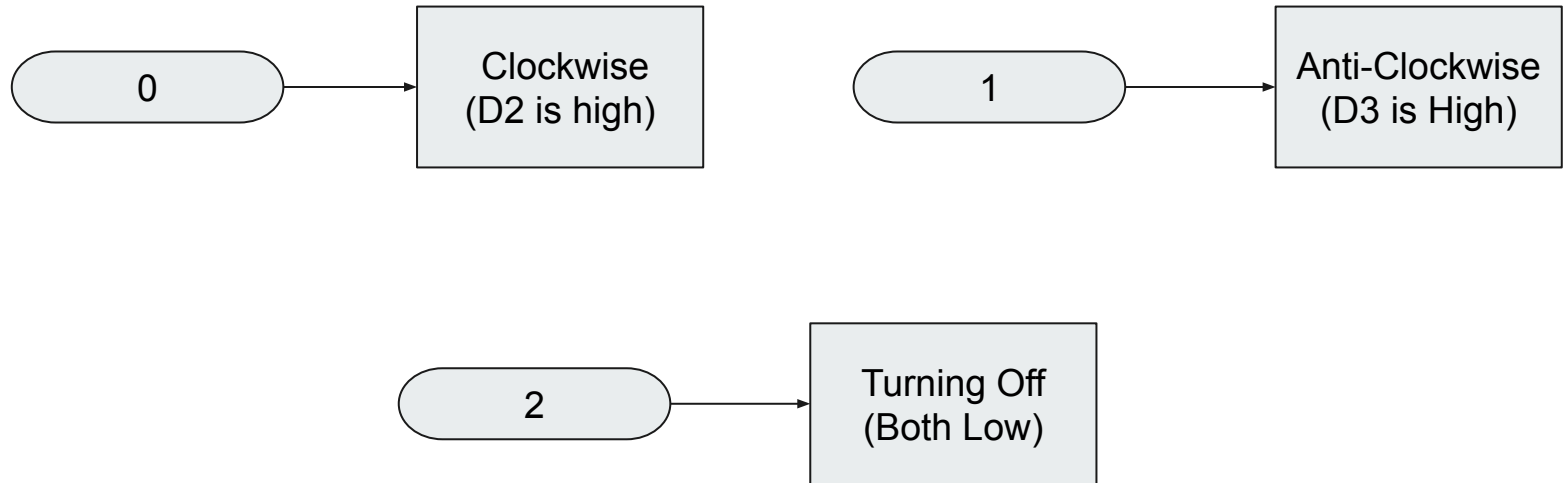
Declaring D2,D3 pin for Controlling Motor

$\left\{ \begin{array}{l} \text{pinMode(D2, OUTPUT);} \\ \text{pinMode(D3, OUTPUT);} \end{array} \right\}$	\longrightarrow	Declaring as Output
--	-------------------	---------------------

$\left\{ \begin{array}{l} \text{digitalWrite(D3, HIGH);} \\ \text{digitalWrite(D2, LOW);} \end{array} \right\}$	\longrightarrow	High = +3.3V supply Low = 0v
---	-------------------	---------------------------------



Mapping Status with Motor



For MOTOR_STATUS = 0

The image shows two side-by-side windows. The left window is the Arduino IDE 2.0.4, displaying the code for 'IIOT_MOTOR.ino'. The code includes headers for ESP8266 WiFi and Firebase, defines Firebase credentials, and sets up a motor status variable. The right window is the Firebase Realtime Database console, showing the 'Data' tab with a single data point 'MOTOR_STATUS: 0'.

Arduino IDE 2.0.4 - IIOT_MOTOR.ino

```
1 #include <ESP8266WiFi.h>
2 #include <FirebaseArduino.h>
3
4 #define FIREBASE_HOST "dofs-9bfa7-default-rtdb.asia-southeast1.firebaseio.com"
5 #define FIREBASE_AUTH "ekTB1BDnh2IBLhOM3Bxv26K7PijihkNBgVpHvnGr" //Firebase
6 #define WIFI_SSID "aa"
7 #define WIFI_PASSWORD "12345678"
8 int fireStatus = 0;
9
10
11
12 void setup() {
13   Serial.begin(9600);
14   pinMode(D2, OUTPUT);
15   pinMode(D3, OUTPUT);
16   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
17   Serial.print("Connecting");
```

Serial Monitor

Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module) on COM10')

02:23:30.615 -> MOTOR Turned OFF
02:23:31.710 -> MOTOR Turned OFF
02:23:32.808 -> MOTOR Turned OFF
02:23:33.894 -> MOTOR Turned OFF
02:23:35.015 -> MOTOR Turned OFF
02:23:36.132 -> MOTOR Turned OFF
02:23:37.235 -> MOTOR Turned OFF
02:23:38.349 -> MOTOR Turned OFF
02:23:39.432 -> MOTOR Turned OFF
02:23:40.559 -> MOTOR Turned OFF
02:23:41.671 -> MOTOR Turned ON CLOCKWISE

Firebase Realtime Database

DOFS - Realtime Database - Firebase

console.firebase.google.com/project/do...

Realtime Database

Protect your Realtime Database resources from abuse, such as billing fraud or phishing. [Configure App Check](#)

<https://dofs-9bfa7-default-rtdb.asia-southeast1.firebaseio.com>

<https://dofs-9bfa7-default-rtdb.asia-southeast1.firebaseio.com>

MOTOR_STATUS: 0

For MOTOR_STATUS = 1

The image shows a dual-screen setup. The left screen displays the Arduino IDE 2.0.4 interface. The top window is titled "IIOT_MOTOR | Arduino IDE 2.0.4" and shows the "IIOT_MOTOR.ino" file. The code includes headers for ESP8266WiFi and FirebaseArduino, defines Firebase credentials, and sets up a motor status variable. The setup function initializes the serial port and WiFi. The output window shows a series of "MOTOR Turned OFF" messages, followed by "MOTOR Turned ON Anticlockwise". The right screen shows a web browser with the "DOFS - Realtime Database - Firebase" console. The "Data" tab is selected, showing a single data point with the key "MOTOR_STATUS" and the value "1".

Arduino IDE Code:

```
1 #include <ESP8266WiFi.h>
2 #include <FirebaseArduino.h>
3
4 #define FIREBASE_HOST "dofs-9bfa7-default-rtdb.asia-southeast1.firebaseio.com"
5 #define FIREBASE_AUTH "ekTB1BDnh2IBLhOM3Bxv26K7PijhkBgVpHVnGr" //Firebase
6 #define WIFI_SSID "aa"
7 #define WIFI_PASSWORD "12345678"
8 int fireStatus = 0;
9
10
11
12 void setup() {
13   Serial.begin(9600);
14   pinMode(D2, OUTPUT);
15   pinMode(D3, OUTPUT);
16   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
17   Serial.print("Connecting");
```

Serial Monitor Output:

```
02:24:09.322 -> MOTOR Turned OFF
02:24:10.403 -> MOTOR Turned OFF
02:24:11.573 -> MOTOR Turned OFF
02:24:12.659 -> MOTOR Turned OFF
02:24:13.792 -> MOTOR Turned OFF
02:24:14.902 -> MOTOR Turned OFF
02:24:16.029 -> MOTOR Turned OFF
02:24:17.160 -> MOTOR Turned OFF
02:24:18.250 -> MOTOR Turned OFF
02:24:19.400 -> MOTOR Turned OFF
02:24:20.519 -> MOTOR Turned ON Anticlockwise
```

Realtime Database Data:

```
{
  "MOTOR_STATUS": 1
}
```

For MOTOR_STATUS = 2

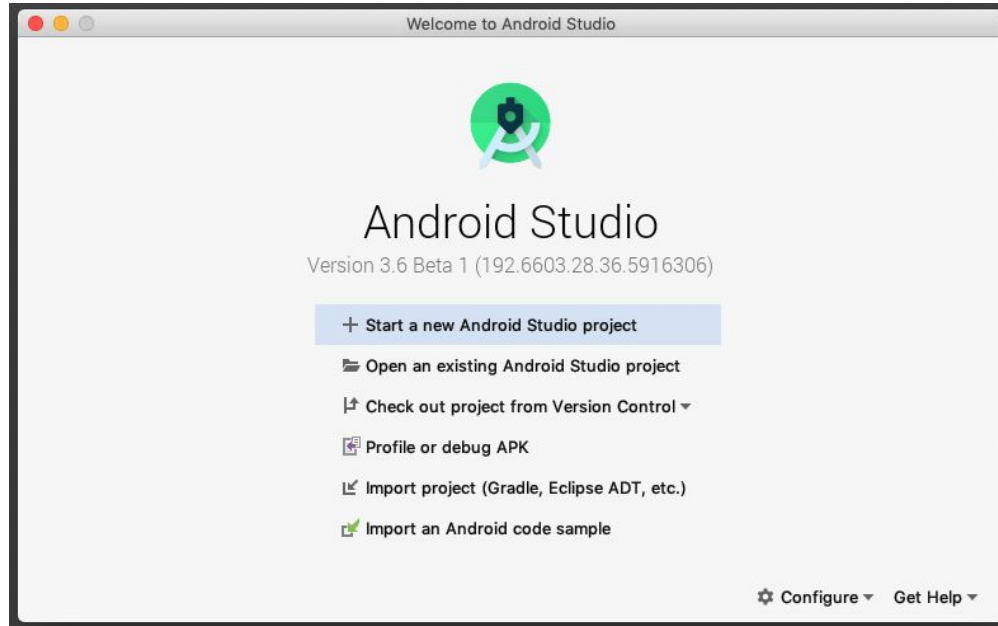
The image shows a dual-screen setup. The left screen displays the Arduino IDE 2.0.4 with a sketch named 'IIOT_MOTOR.ino'. The code includes headers for ESP8266 WiFi and Firebase Arduino, defines Firebase credentials, and sets up a motor status variable. The Serial Monitor shows a sequence of messages where the motor is turned on clockwise and then off.

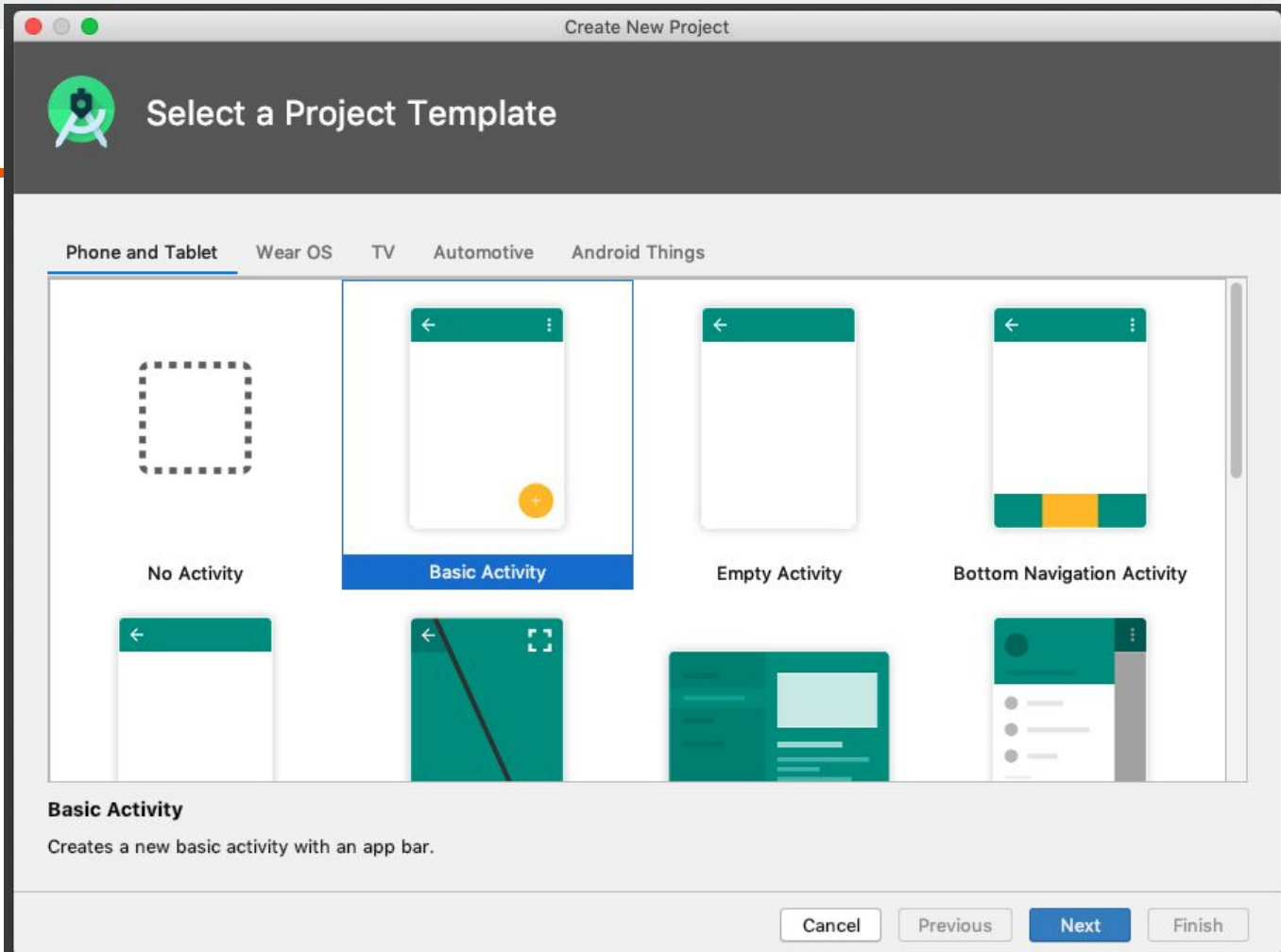
```
1 #include <ESP8266WiFi.h>
2 #include <FirebaseArduino.h>
3
4 #define FIREBASE_HOST "dofs-9bfa7-default-rtdb.asia-southeast1.firebaseio.com"
5 #define FIREBASE_AUTH "ekTB1BDnh2IBLhOM3Bxv26K7PijihkNBgVpHvnGr" //Firebase
6 #define WIFI_SSID "aa"
7 #define WIFI_PASSWORD "12345678"
8 int fireStatus = 0;
9
10
11
12 void setup() {
13   Serial.begin(9600);
14   pinMode(D2, OUTPUT);
15   pinMode(D3, OUTPUT);
16   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
17   Serial.print("Connecting");
```

The right screen shows the Firebase Realtime Database console. The 'Data' tab is active, displaying a JSON object with 'MOTOR_STATUS' set to 2.


```
{
  "MOTOR_STATUS": 2
}
```

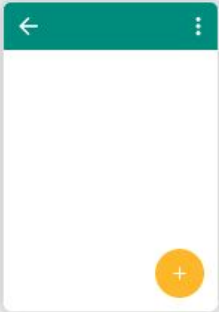
Android app development





Create New Project

 Configure Your Project



Basic Activity

Creates a new basic activity with an app bar.

Name

Package name


Save location


Language

Java

Kotlin

Java

 Your app will run on approximately **99.6%** of devices.
[Help me choose](#)

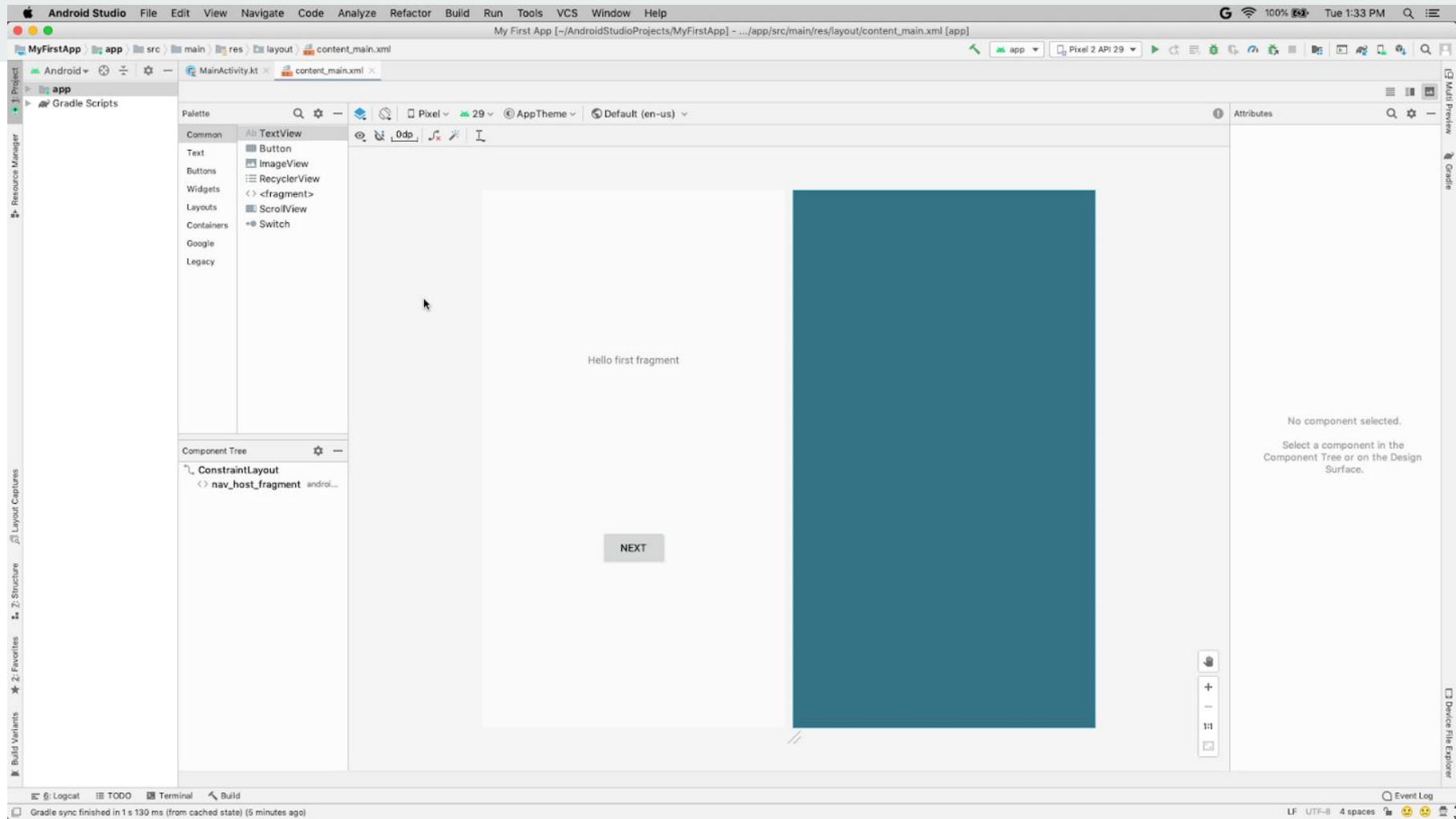
☐ Use legacy android.support libraries 

Cancel

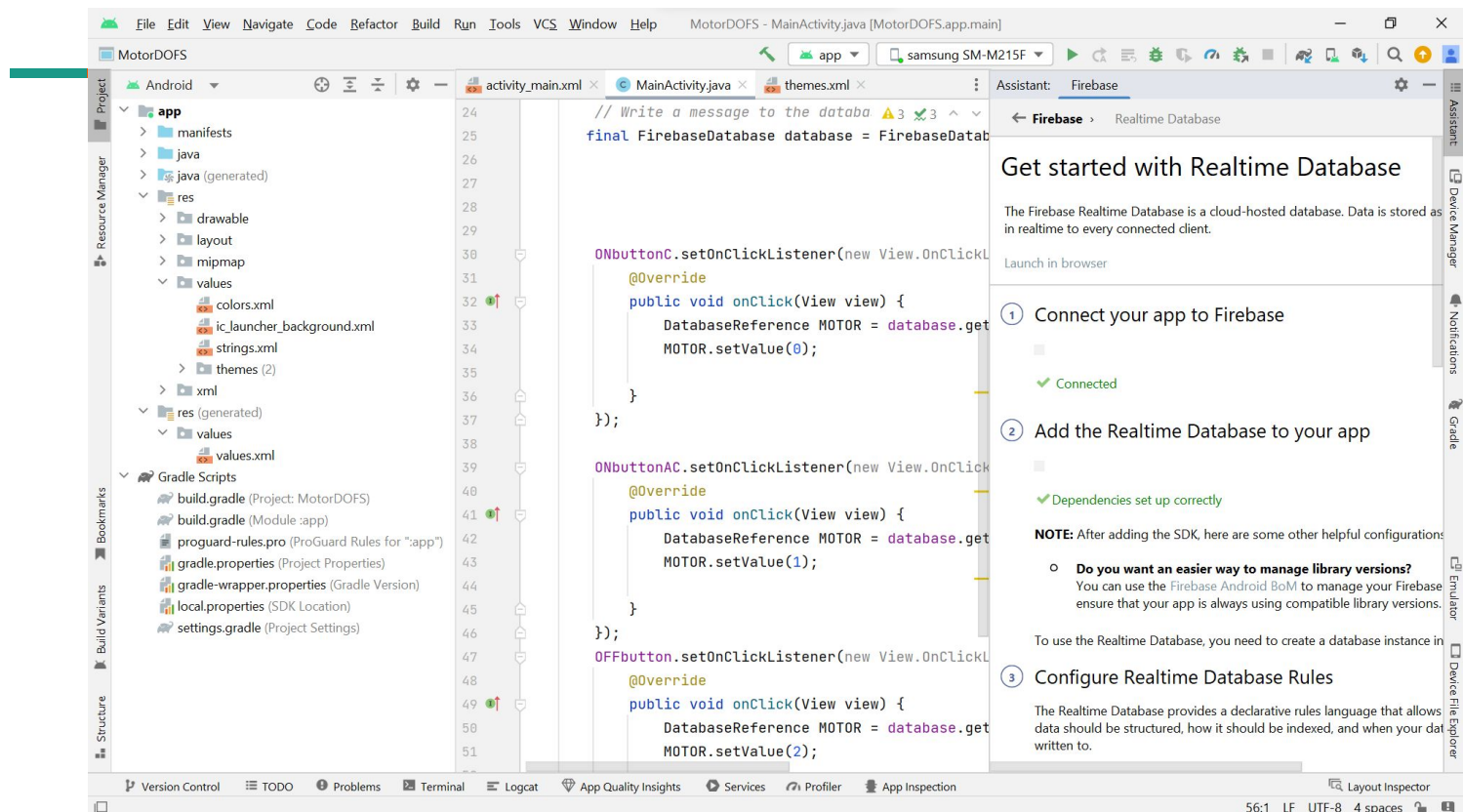
Previous

Next


Finish



Integrating Firebase with Android APP

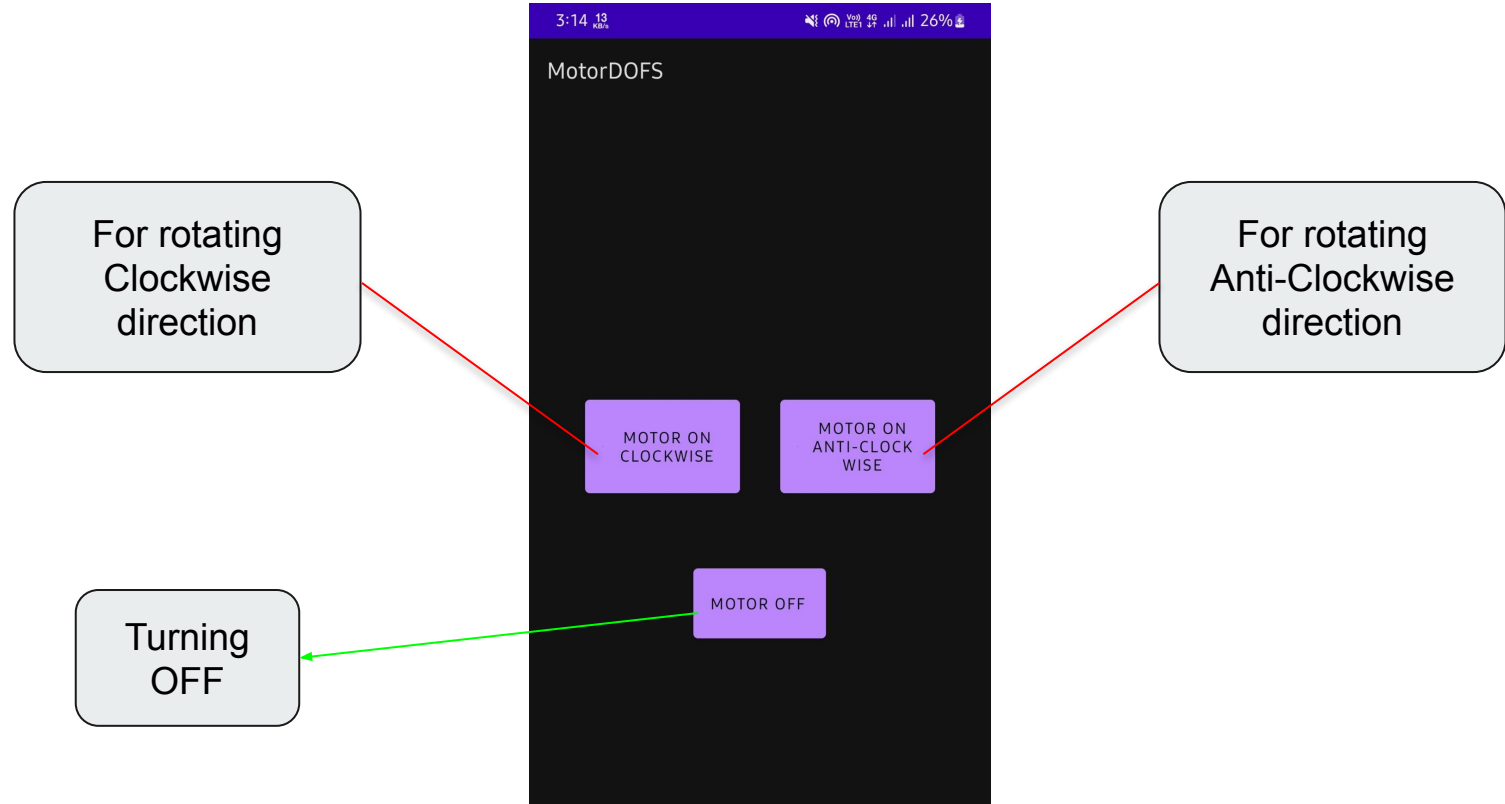


Additional Change in the code for Firebase Integration



```
});  
OFFbutton.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        DatabaseReference MOTOR = database.getReference( path: "MOTOR_STATUS");  
        MOTOR.setValue(2);  
    }  
});
```

APP Interface





Conclusion

In conclusion, the project of controlling DC motor operation using Firebase, ESP8266, and an Android app is best example of the capabilities of the IoT.

It shows how easy it is to control a device remotely, allowing for increased convenience and flexibility.

This project has several potential real-world applications, such as controlling motors in industrial settings or home automation systems.



Reference

- [1] “Build your first Android app in java,” *Android Developers*. [Online]. Available: <https://developer.android.com/codelabs/build-your-first-android-app>. [Accessed: 23-Mar-2023].
- [2] Wikipedia contributors, “ESP8266,” *Wikipedia, The Free Encyclopedia*, 22-Jan-2023. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=ESP8266&oldid=1135215060>.
- [3] “HG7881 4-channel DC 2.4-10V Motor Driver Broad,” *Techtonics.in*. [Online]. Available: <https://www.techtonics.in/hg7881-4-channel-dc-2-4-10v-motor-driver-broad>. [Accessed: 23-Mar-2023].
- [4] Tygo-bear, “Esp8266 firebase connection,” *Instructables*, 27-Nov-2018. [Online]. Available: <https://www.instructables.com/Esp8266-Firebase-Connection/>. [Accessed: 22-Mar-2023].



Thank You!