

IOT PLATFORM, LẬP TRÌNH ESP32

Giao tiếp với Realtime Database Firebase sử dụng ESP32 và App

POSTED ON 30/10/2021 BY KHUÊ NGUYỄN

30
Th10

irebase IOT Platform

Khuê Nguyễn Creator

+ Firebase +

Giao tiếp Reatime Database Firebase với ESP32 và App

Trong bài này chúng ta cùng nhau học cách sử dụng Realtime Database Firebase bằng cách đọc và gửi dữ liệu từ ESP32 và App lên. Cùng nhau tìm hiểu cấu trúc dữ liệu của Firebase Realtime Database, cách tạo và điều khiển nó.



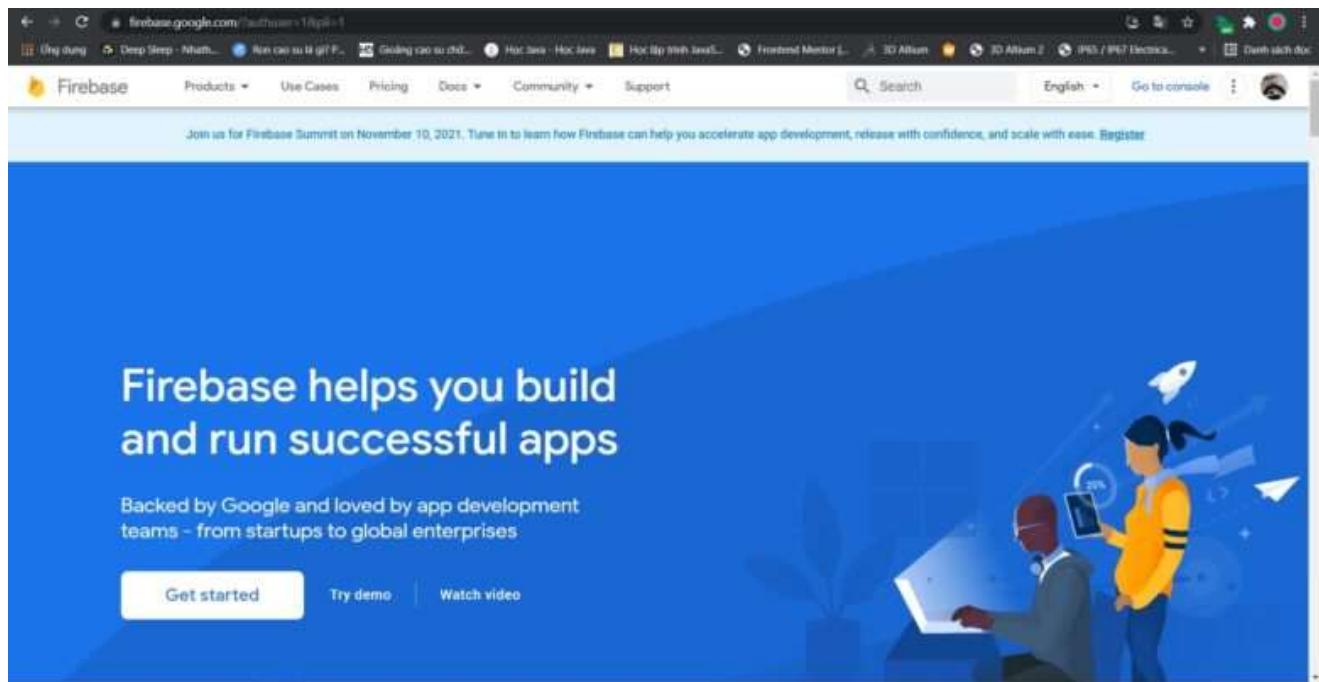
Mục Lục

1. Cách sử dụng Firebase
 - 1.1. Tạo project trên firebase
 - 1.2. Khởi tạo và cài đặt Firebase Realtime Database
 - 1.3. Khởi tạo cấu trúc dữ liệu cho Realtime Database Firebase
2. Kết nối ESP32 với Firebase truyền nhận dữ liệu
 - 2.1. Thêm thư viện và kết nối với Realtime Database Firebase
 - 2.2. Lập trình ESP32 truyền dữ liệu lên Firebase
 - 2.2.1. Get dữ liệu từ Firebase về ESP32
 - 2.2.2. Push dữ liệu từ ESP32 lên Firebase
 - 2.3. Kết quả
3. Giao tiếp giữa App Android và Realtime Database Firebase
 - 3.1. Tạo project và cài đặt gói thư viện Firebase
 - 3.2. Thiết kế giao diện điều khiển trên App
 - 3.2.1. Setup các trường dữ liệu và địa chỉ database
 - 3.2.2. Push dữ liệu từ App lên Firebase.
 - 3.2.3. Get dữ liệu từ Firebase về App
 - 3.3. Kết Quả
4. Kết
 - 4.1. Related posts:

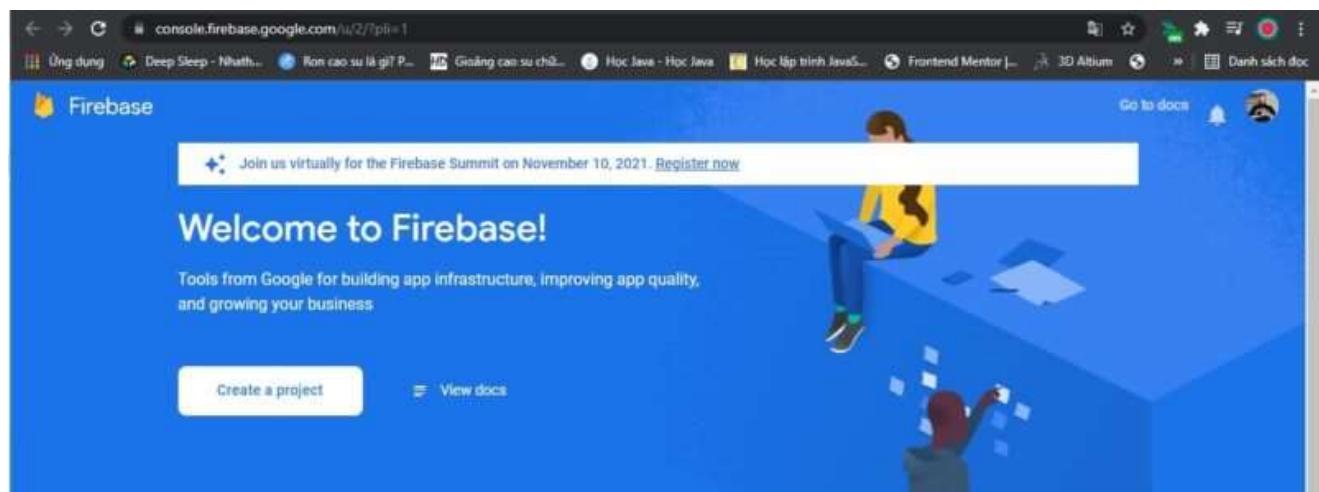
Cách sử dụng Firebase

Tạo project trên firebase

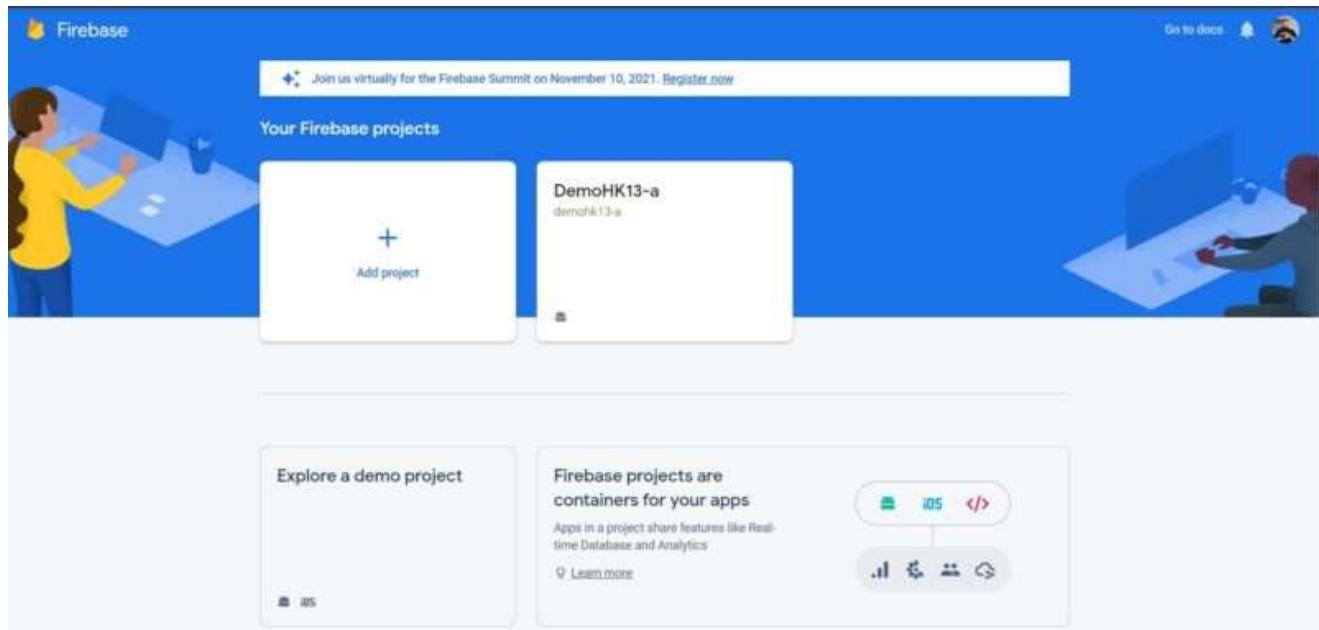
Để bắt đầu sử dụng Firebase ta truy cập đến trang chủ ở địa chỉ: <https://firebase.google.com/> do Firebase là dịch vụ của Google nên ta đăng nhập bằng tài khoản Gmail của mình:



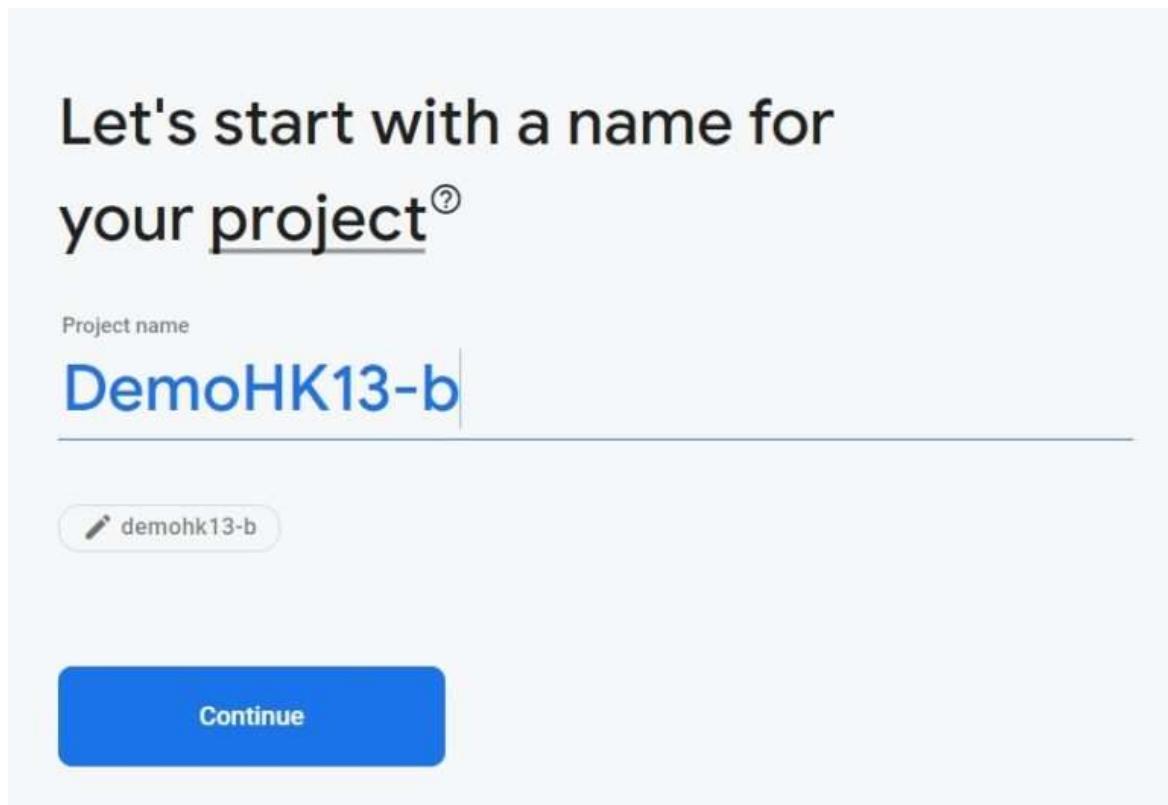
Sau đó ta Go to console với giao diện như dưới đây và chọn Create a project để bắt đầu dự án của chúng ta



Ta Add project



Đặt tên cho Project



Chọn continue

X Create a project (Step 2 of 3)

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, Predictions, and Cloud Functions.

Google Analytics enables:

-  A/B testing [?](#)
-  Crash-free users [?](#)
-  User segmentation & targeting across Firebase products [?](#)
-  Event-based Cloud Functions triggers [?](#)
-  Predicting user behavior [?](#)
-  Free unlimited reporting [?](#)

Enable Google Analytics for this project
Recommended

Previous

Continue

Tiếp theo ta chọn Default Account for **Firebase** và Create project

X Create a project (Step 3 of 3)

Configure Google Analytics

Choose or create a Google Analytics account [?](#)

 Default Account for Firebase

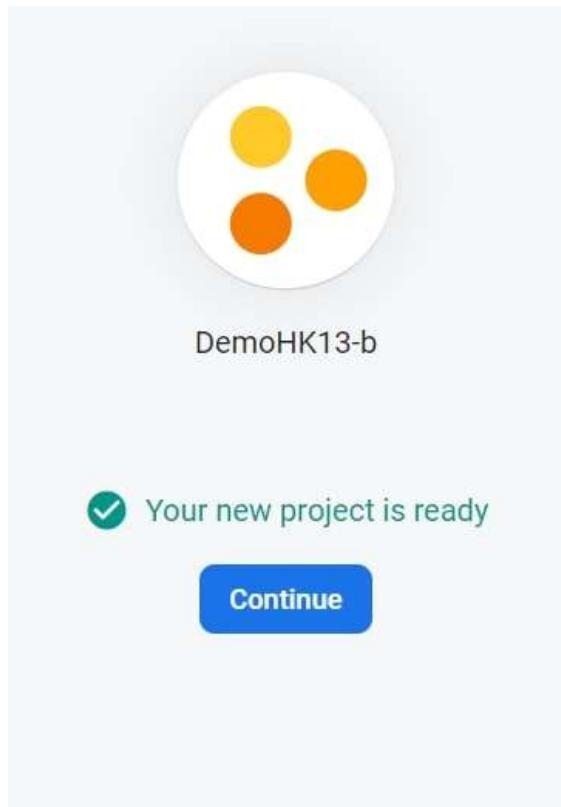
Automatically create a new property in this account 

Upon project creation, a new Google Analytics property will be created in your chosen Google Analytics account and linked to your Firebase project. This link will enable data flow between the products. Data exported from your Google Analytics property into Firebase is subject to the Firebase terms of service, while Firebase data imported into Google Analytics is subject to the Google Analytics terms of service. [Learn more](#).

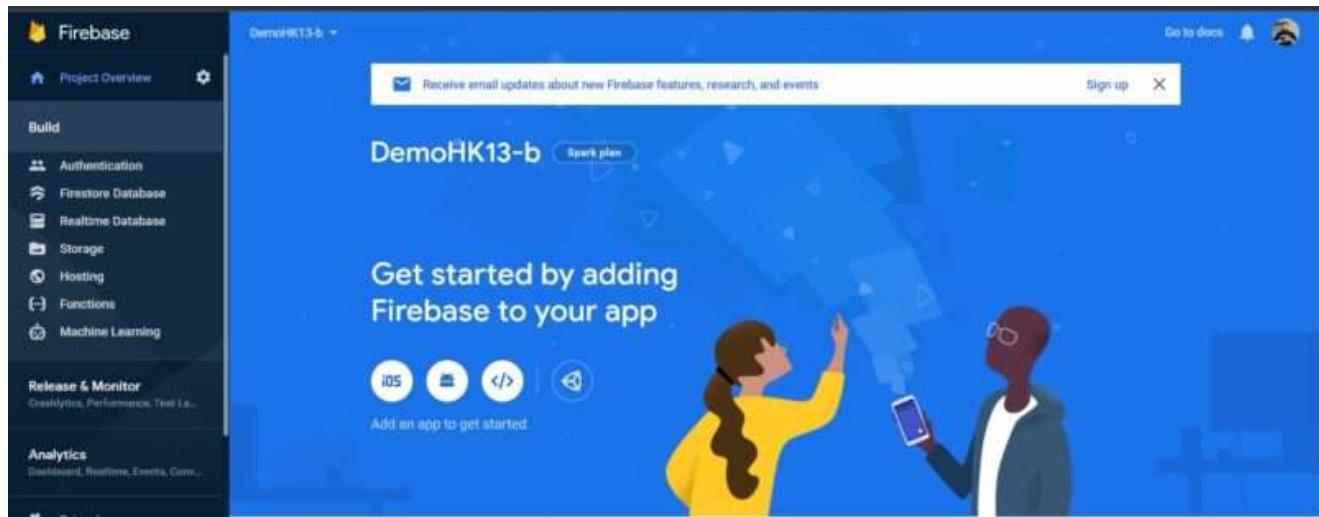
Previous

Create project

Sau đó ta đợi Firebase khởi tạo project của chúng ta



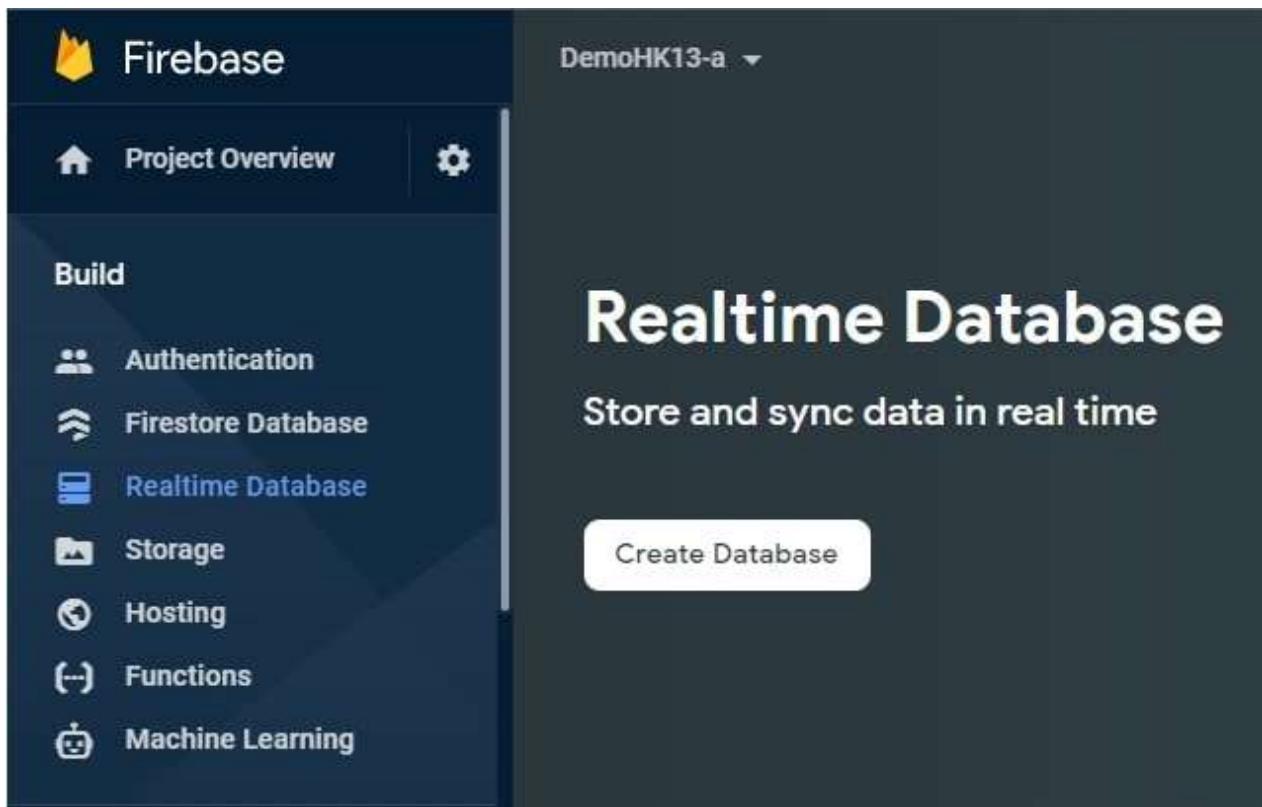
Cuối cùng Firebase sẽ dẫn ta đến giao diện như sau:



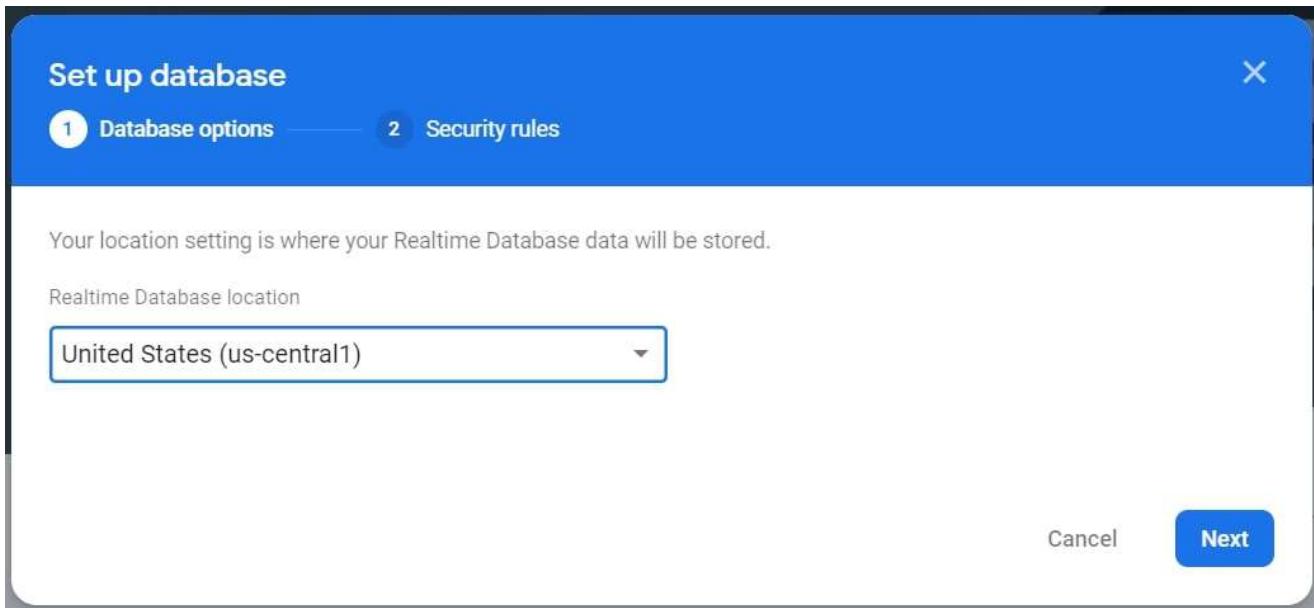
Khởi tạo và cài đặt Firebase Realtime Database

Ở đây ta sẽ sử dụng Services Realtime Database của Firebase với mục đích làm cơ sở dữ liệu để các client có thể tương tác qua lại.

Đầu tiên ta Create Realtime Database



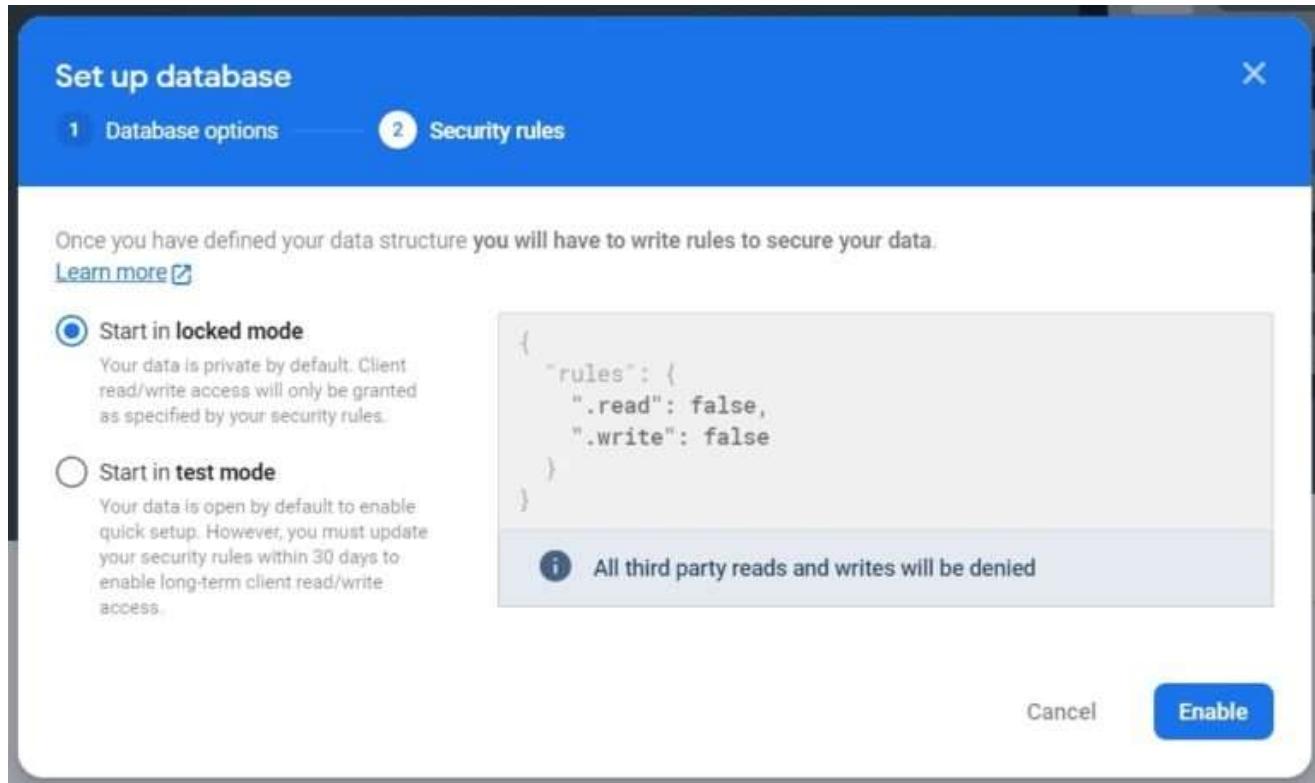
Ở đây ta chọn mặc định và next



Bước tiếp theo :

- Nếu ta đang phát triển dự án với mục đích để test các chức năng thì ta chọn test mode, nhưng database của chúng ta sẽ ở trạng thái open trong 30 ngày đầu tiên

- Ngược lại khi ta cần một database để phục vụ cho mobile app hoặc các ứng dụng web chính thức thì ta chọn locked mode và database của chúng ta sẽ ở chế độ private và dưới các rules mà chúng ta thiết lập (ở đây ta chọn locked mode và Enable)



Khởi tạo cấu trúc dữ liệu cho Realtime Database Firebase

Giao diện của Database của chúng ta hiện tại chưa có một trường dữ liệu và dữ liệu nào.

The screenshot shows the Firebase Realtime Database console for a project named "DemoHK13-b". The "Data" tab is selected. A single node named "demohk13-b-default-rtfb" is present with a value of "null". The URL of the database is displayed as "https://demohk13-b-default-rtdb.firebaseio.com/". The sidebar on the left lists other Firebase services: Authentication, Firestore Database, Realtime Database, Storage, Hosting, Functions, and Machine Learning.

Tiếp theo ta sẽ tạo ra trường dữ liệu và gán dữ liệu theo cách thủ công bằng cách nhấn vào dấu cộng (add child)

The screenshot shows a browser window with the URL "https://demohk13-b-default-rtdb.firebaseio.com/" in the address bar. Below the address bar, there is a text input field containing the path "demohk13-b-default-rtfb: null". To the right of the input field are three buttons: a plus sign (+), a minus sign (-), and an 'x'. Below these buttons is a dark button labeled "add child".

Ta tạo ra trường esp32 và esp32 – string sau đó gán dữ liệu theo kiểu ta muốn

Dữ liệu có thể là kiểu int, string, float... tùy theo dữ liệu ban đầu bạn nhập vào.

➡ <https://demohk13-b-default-rtdb.firebaseio.com/>

demohk13-b-default-rtdb: null

Name Value

Trong bài này, ta sẽ tạo Database như sau

➡ <https://demohk13-a-default-rtdb.firebaseio.com/>

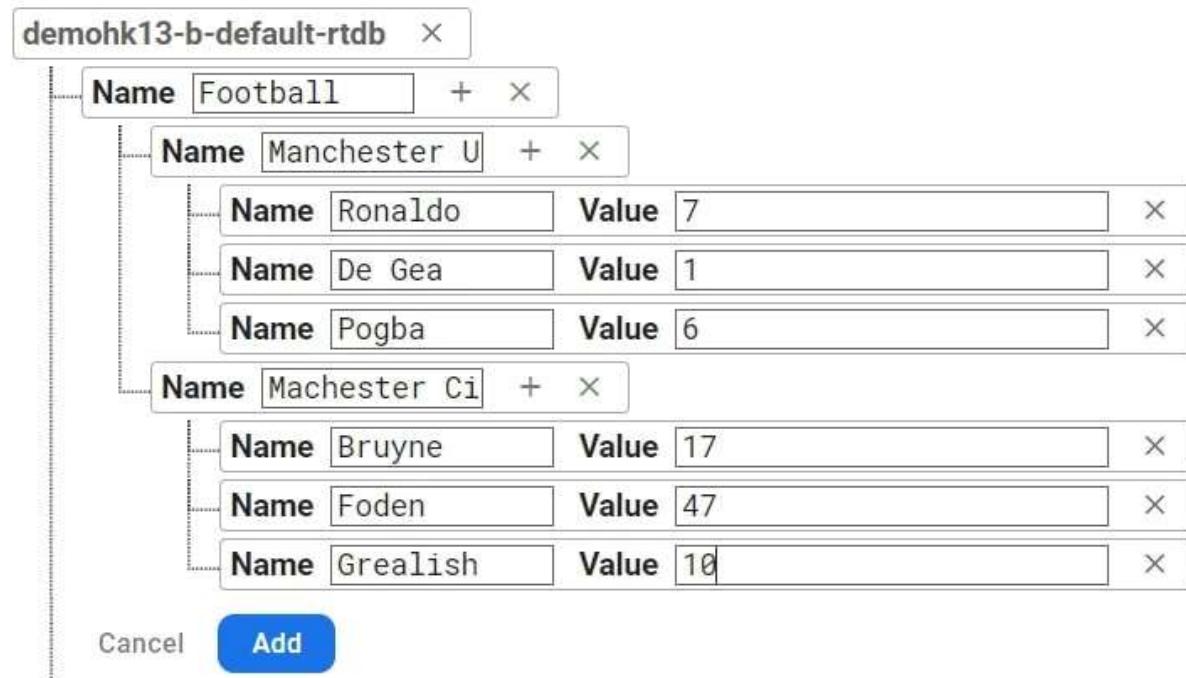
demohk13-a-default-rtdb

esp32: 1

esp32-string: "on"

Realtime Database Firebase có cấu trúc kiểu cha – con. Tương tự như cấu trúc của chuỗi JSON. Vậy nên chúng ta có thể thay đổi 1 dữ liệu hoặc tất cả dữ liệu bằng cách gửi chuỗi JSON lên Database.

Tùy vào mục đích ta có thể thiết kế Database phức tạp hơn bằng cách ấn vào dấu cộng như trên và ta được kết quả như ảnh minh họa dưới đây



Cấu trúc dữ liệu của Realtime Database Firebase

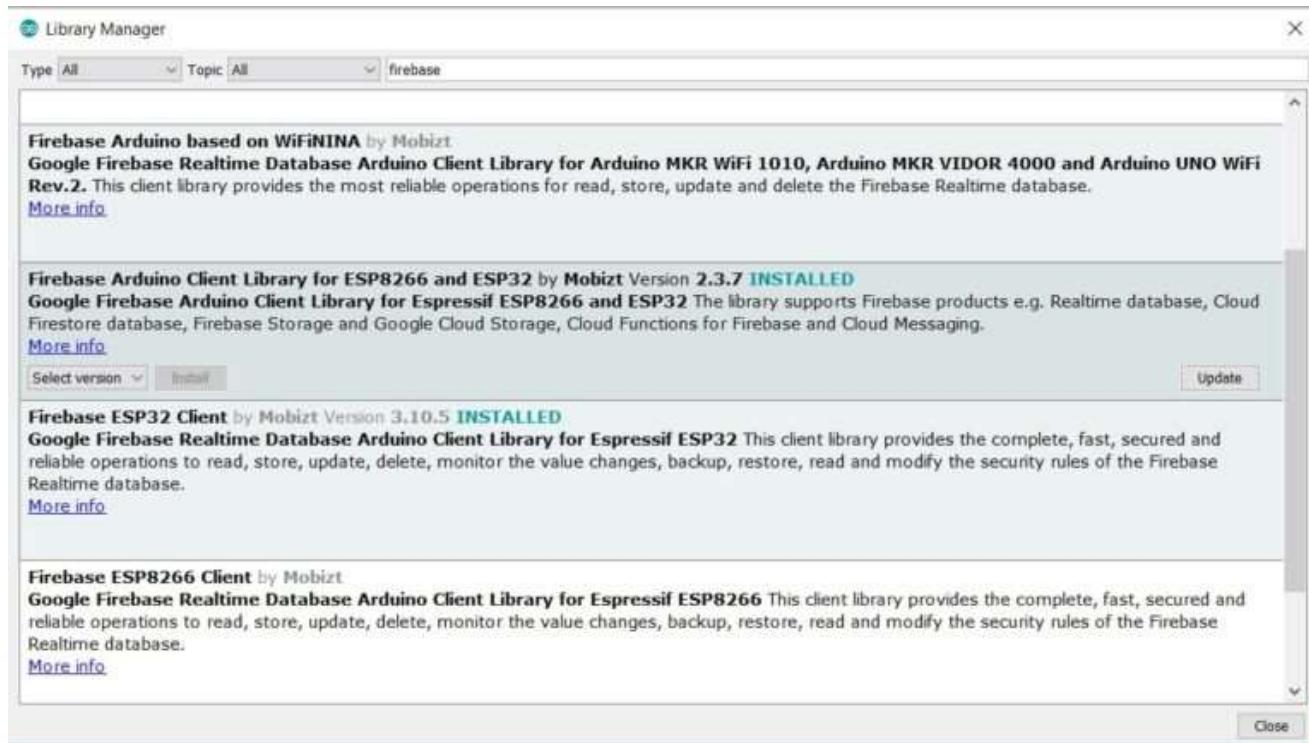
Kết nối ESP32 với Firebase truyền nhận dữ liệu

Thêm thư viện và kết nối với Realtime Database Firebase

Ta mở phần mềm Arduino IDE để bắt đầu code cho module **esp32** của chúng ta, sau đó ấn tổ hợp phím Ctrl + N để tạo ra trang mới như sau:



Ta vào Library Manager và cài gói thư viện Firebase của Mobitz



Tiếp theo ta thêm các thư viện cần thiết (ở đây ta đang code cho module esp32-cam) và thư viện FirebaseESP32.h

working | Arduino 1.8.3

File Edit Sketch Tools Help

working §

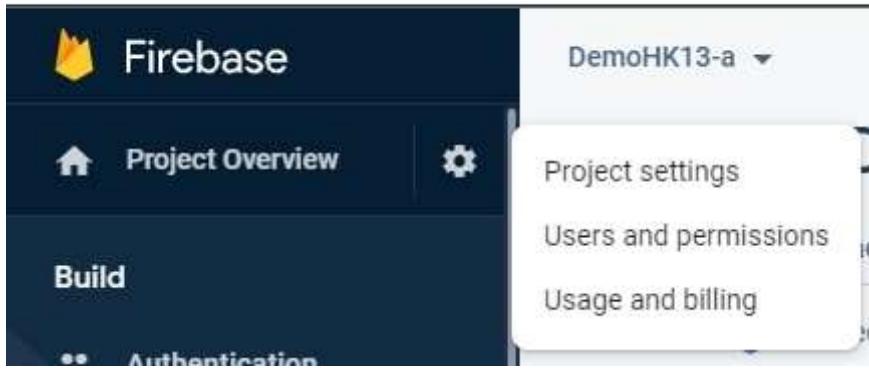
```

1 #include "esp_camera.h"
2 #include <WiFi.h>
3
4 #define CAMERA_MODEL_AI_THINKER // Has PSRAM
5
6 #include "camera_pins.h"
7
8 // Firebase Library
9 #include "FirebaseESP32.h"
10

```

Tiếp theo để sử dụng Firebase với **esp32** thì ta phải lấy host và đường dẫn đến Database của chúng ta để cập nhật vào code

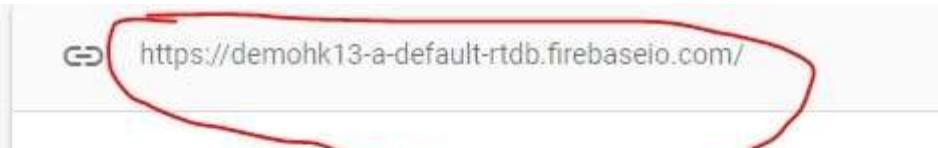
Trong mục Project settings ta thực hiện lấy mã Firebase Authentication.



Vào thẻ Service accounts để lấy secrets code:

Database	Secret
demohk13-a-default-rtbd	I52NdYfvKfAgxBV0aPGZQcghVwNT3ngrZG4Rt04D

Trong mục Realtime Database ta lấy đường dẫn truy xuất đến Database.



https://demohk13-a-default-rtdb.firebaseio.com/

demohk13-a-default-rtdb

esp32: 9

esp32-string: "off"

Trong code ta thực hiện khai báo Host và Authentication cho module và khai báo biến trung gian để truy xuất đến Firebase

```
// Firebase Library
#include "FirebaseESP32.h"
// Firebase Declare
#define FIREBASE_HOST "https://demohk13-a-default-rtdb.firebaseio.com/"
#define FIREBASE_AUTH "I52NdYfvKfAgxBV0aPGZQcghVwNT3ngrZG4RtO4D"
// Create Variables
FirebaseData firebaseData;
FirebaseJson json;
```

Lập trình ESP32 truyền dữ liệu lên Firebase

Đầu tiên, cập nhật WiFi cho esp32, là wifi mà nơi bạn sử dụng

```
const char* ssid = "HK13CQT";
const char* password = "canhhuybk";
```

Trong Setup() {...} ta thực hiện kết nối WiFi

```

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

```

Để kết nối đến Firebase thì ta dùng lệnh sau:

```

// kết nối Firebase
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
Firebase.reconnectWiFi(true);

```

Get dữ liệu từ Firebase về ESP32

Tạo biến `data5base` để lưu giá trị get được từ Firebase về.

Ta thực hiện chức năng get bằng lệnh `Firebase.getInt(địa chỉ firebase, vị trí)`

Sau đó lấy giá trị trả về lưu vào `data5base`

Push dữ liệu từ ESP32 lên Firebase

Tạo 1 String tên `truongthaydoi` chính là đường dẫn tới vị trí `esp32-string`.

Chúng ta sẽ push dữ liệu từ Esp32 lên đó.

Sử dụng hàm `Firebase.setString(địa chỉ firebase, vị trí, dữ liệu)`

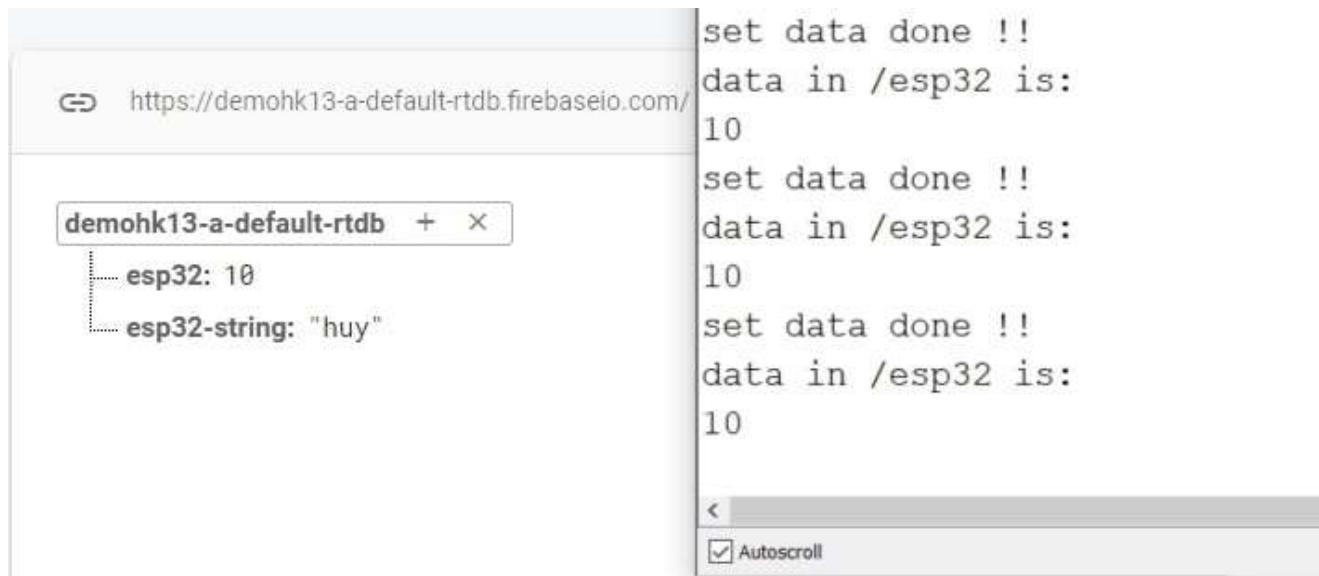
Lúc này **ESP32** sẽ đẩy dữ liệu lên firebase.

```
// biến hùng data
int data5base;
// dẫn đến trường esp32-string
String truongthaydoi = "/esp32-string";
void loop() {
    // lấy giá trị kiểu int ở trường esp32
    Firebase.getInt(firebaseData, "/esp32");
    // hùng giá trị
    data5base = firebaseData.intData();
    {
        Serial.println("data in /esp32 is: ");
        Serial.println(data5base);
        delay(1000);
    }

    // set string: "huy" vào trường esp32-string
    Firebase.setString(firebaseData, truongthaydoi, "huy");
    Serial.println("set data done !!");
    delay(500);
}
```

Kết quả

Ta đã push được data kiểu string “huy” lên trường esp32-string và get được giá trị 10 từ DB về.



Ta cũng có thể gửi data lên trường `esp32` và đọc về ngược lại bằng hàm `setInt`.

```
// push data to /esp32
Firebase.setInt(firebaseData, truongthaydoiInt, 99);
Serial.println("Push data to /esp done !!!");
Serial.println();
delay(1000);
```

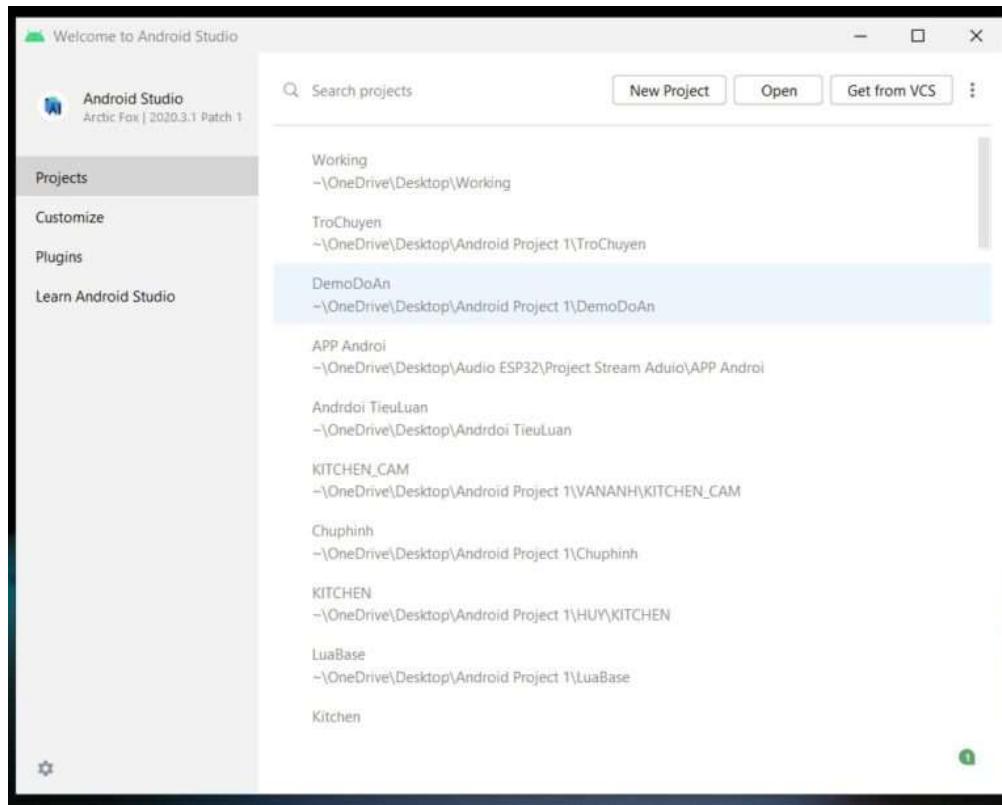
Hoặc các bạn có thể thay đổi trực tiếp trên database, esp32 sẽ tự động đọc được.

Như vậy là chúng ta đã có thể giao tiếp giữa Esp32 và Firebase rồi. Để một hệ thống **IOT** hoàn thiện chúng ta cần có App để kết nối lên nữa. Vậy kết nối từ app với Firebase như thế nào, cùng đọc tiếp nhé!

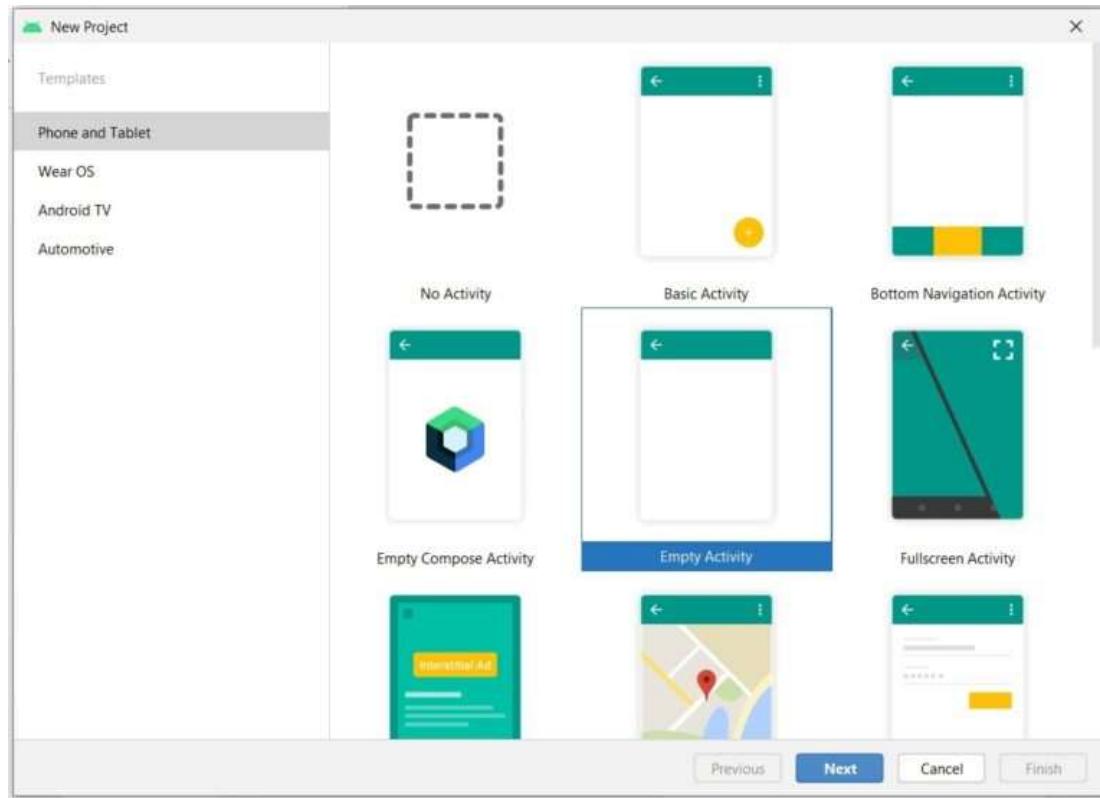
Giao tiếp giữa App Android và Realtime Database Firebase

Tạo project và cài đặt gói thư viện Firebase

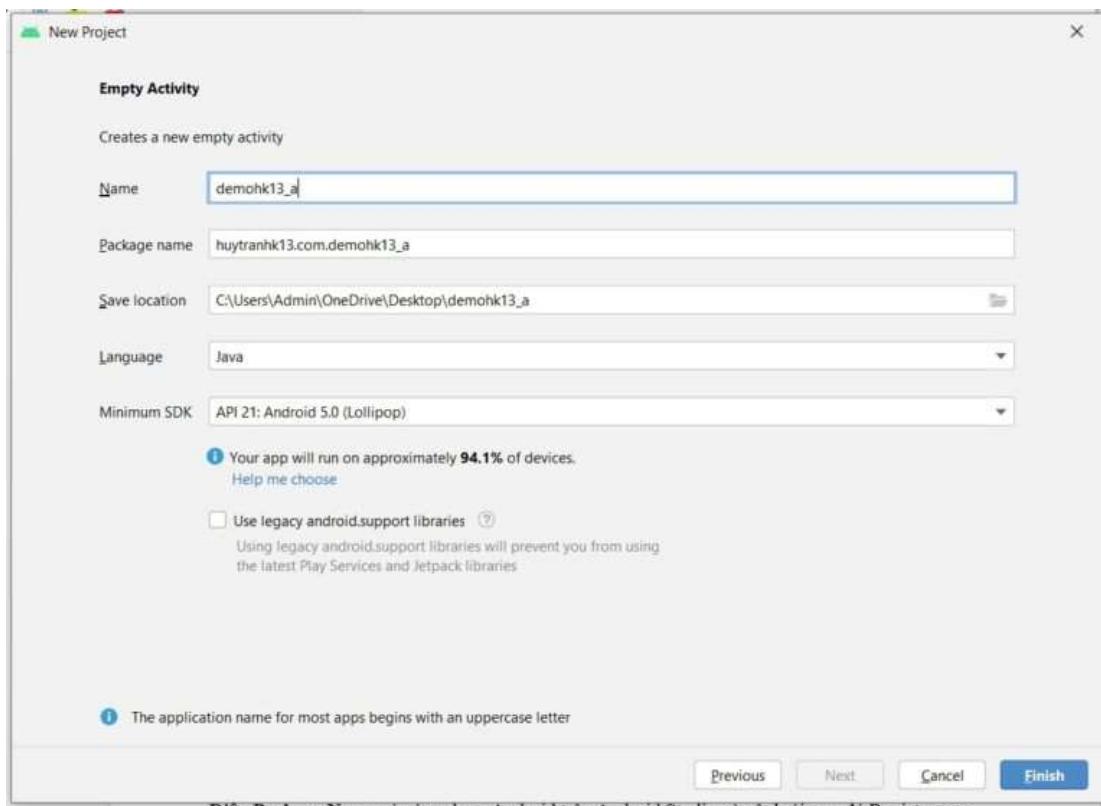
Ta sẽ tạo ra một mobile app đơn giản bằng Android Studio, đầu tiên khởi động môi trường Android studio lên và chọn New Project



Tiếp theo ta thấy rất nhiều giao diện để lựa chọn, nhưng ở đây ta chọn Empty Activity và Next



Bước cuối ta đặt tên cho ứng dụng và chọn thư mục để lưu dự án của chúng ta, và Finish



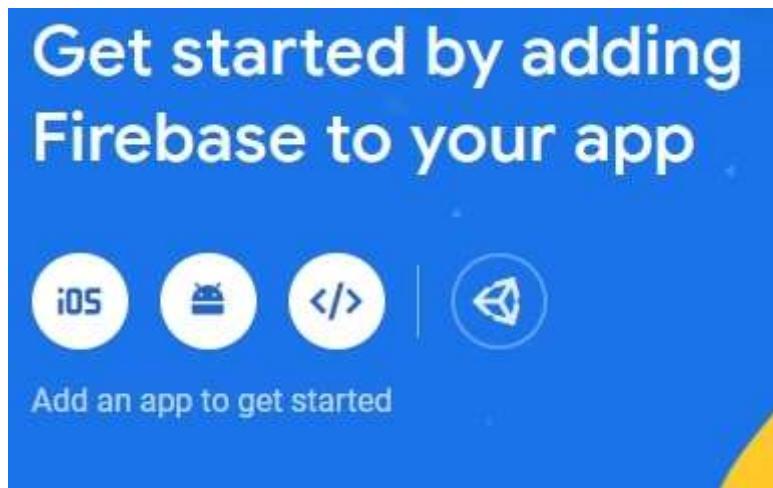
Sau khi khởi tạo Project trên Android Studio sau đó ta chuyển đến MainActivity và ở dòng package để lấy Android package name

```
package huytranhk13.com.demohk13_a;
import ...;

public class MainActivity extends AppCompatActivity {
```

Sau đó ta quay về Firebase và bắt đầu các bước để có thể tích hợp vào Mobile app của chúng ta

Ở giao diện chính trong project ta chọn Android để tích hợp Firebase



Điền Package Name của ứng dụng Android trên Android Studio vào ô dưới, sau đó Register app

Firebase cung cấp file .json và yêu cầu chúng ta copy file này đến thư mục app trong project mobile app của chúng ta

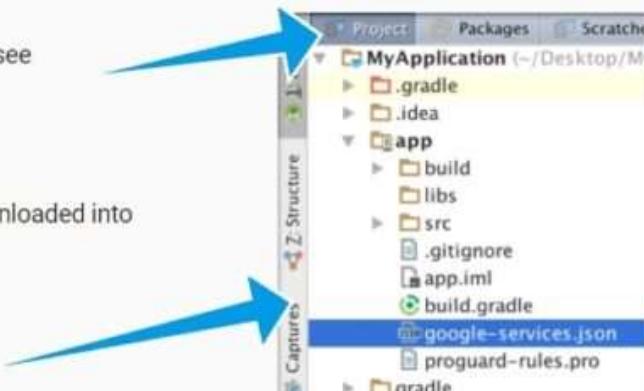
2 Download config file

Instructions for Android Studio below | [Unity](#) [C++](#)

[!\[\]\(51804f5e5546d88130854627985a348c_img.jpg\) Download google-services.json](#)

Switch to the Project view in Android Studio to see your project root directory.

Move the google-services.json file you just downloaded into your Android app module root directory.



 google-services.json

[Next](#)

Download file .json mà Firebase cung cấp về để vào thư mục app của Android, sau đó next

Working > app				
	Name	Date modified	Type	Size
ss	libs	22/10/2021 02:22	File folder	
ds	src	22/10/2021 02:22	File folder	
ts	.gitignore	22/10/2021 02:22	Text Document	1 KB
o	build.gradle	22/10/2021 02:22	Gradle Source File	1 KB
b	google-services.json	22/10/2021 02:39	JSON Source File	1 KB
r	proguard-rules.pro	22/10/2021 02:22	PRO File	1 KB

Tiếp theo Firebase hướng dẫn chúng ta add Firebase SDK vào app Android, ta làm theo hướng dẫn

3 Add Firebase SDK

Instructions for Gradle | Unity C++

The Google services plugin for [Gradle](#) loads the google-services.json file you just downloaded. Modify your build.gradle files to use the plugin.

Project-level build.gradle (<project>/build.gradle):

```
buildscript {
    repositories {
        // Check that you have the following line (if not, add it):
        google() // Google's Maven repository
    }
    dependencies {
        ...
        // Add this line
        classpath 'com.google.gms:google-services:4.3.10'
    }
}

allprojects {
    ...
    repositories {
        // Check that you have the following line (if not, add it):
        google() // Google's Maven repository
        ...
    }
}
```

Java Kotlin

App-level build.gradle (<project>/<app-module>/build.gradle):

```
apply plugin: 'com.android.application'
// Add this line
apply plugin: 'com.google.gms.google-services'

dependencies {
    // Import the Firebase BoM
    implementation platform('com.google.firebase:firebase-bom:28.4.2')
    ...
    // Add the dependency for the Firebase SDK for Google Analytics
    // When using the BoM, don't specify versions in Firebase dependencies
    implementation 'com.google.firebase:firebase-analytics'
    ...
    // Add the dependencies for any other desired Firebase products
    // https://firebase.google.com/docs/android/setup#available-libraries
}
```

By using the Firebase Android BoM, your app will always use compatible Firebase library versions. [Learn more](#)

Finally, press "Sync now" in the bar that appears in the IDE:

Gradle files have changed since [Sync now](#)

[Previous](#)

[Next](#)

Trong mục build.gradle (project) ta thêm ở phần dependencies dòng classpath 'com.google.gms:google-services:4.3.10

```

dependencies {
    classpath "com.android.tools.build:gradle:7.0.1"
    classpath 'com.google.gms:google-services:4.3.10'
    // NOTE: Do not place your application dependencies here; they belong
    // in the individual module build.gradle files
}

```

Trong **build.gradle (Module)** ta thêm ở phần plugins dòng id 'com.google.gms.google-services'

```

plugins {
    id 'com.android.application'
    id 'com.google.gms.google-services'
}

```

Vẫn trong **build.gradle (Module)** thêm ở dependencies 2 thư viện

```

dependencies {

    implementation 'androidx.appcompat:appcompat:1.3.1'
    implementation 'com.google.android.material:material:1.4.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.0'
    testImplementation 'junit:junit:4.+'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
    implementation platform('com.google.firebase:firebase-bom:28.4.2')
    implementation 'com.google.firebase:firebase-analytics'
}

```

Cuối cùng ta chọn Continue to console để quay về giao diện console

4 Next steps

You're all set!

Make sure to check out the [documentation](#) to learn how to get started with each Firebase product that you want to use in your app.

You can also explore [sample Firebase apps](#).

Or, continue to the console to explore Firebase.

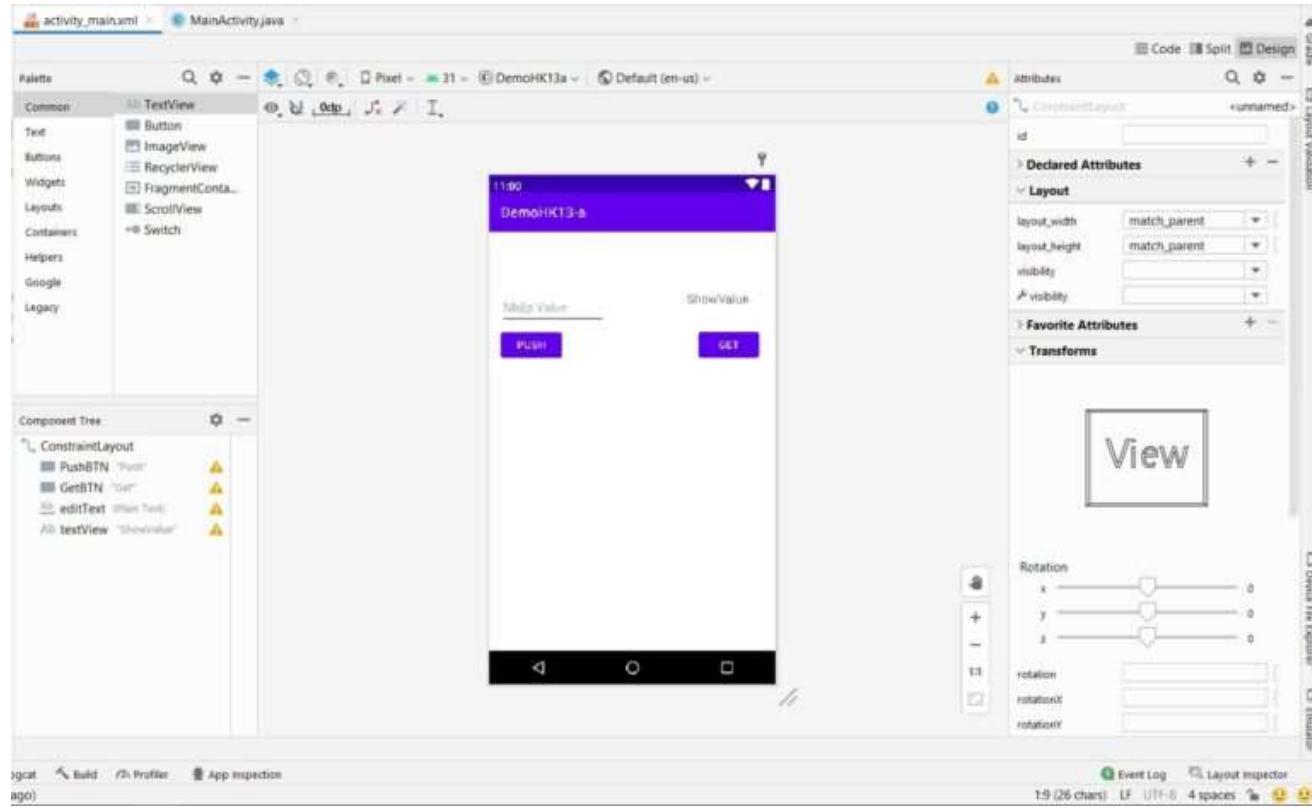
[Previous](#)

[Continue to console](#)

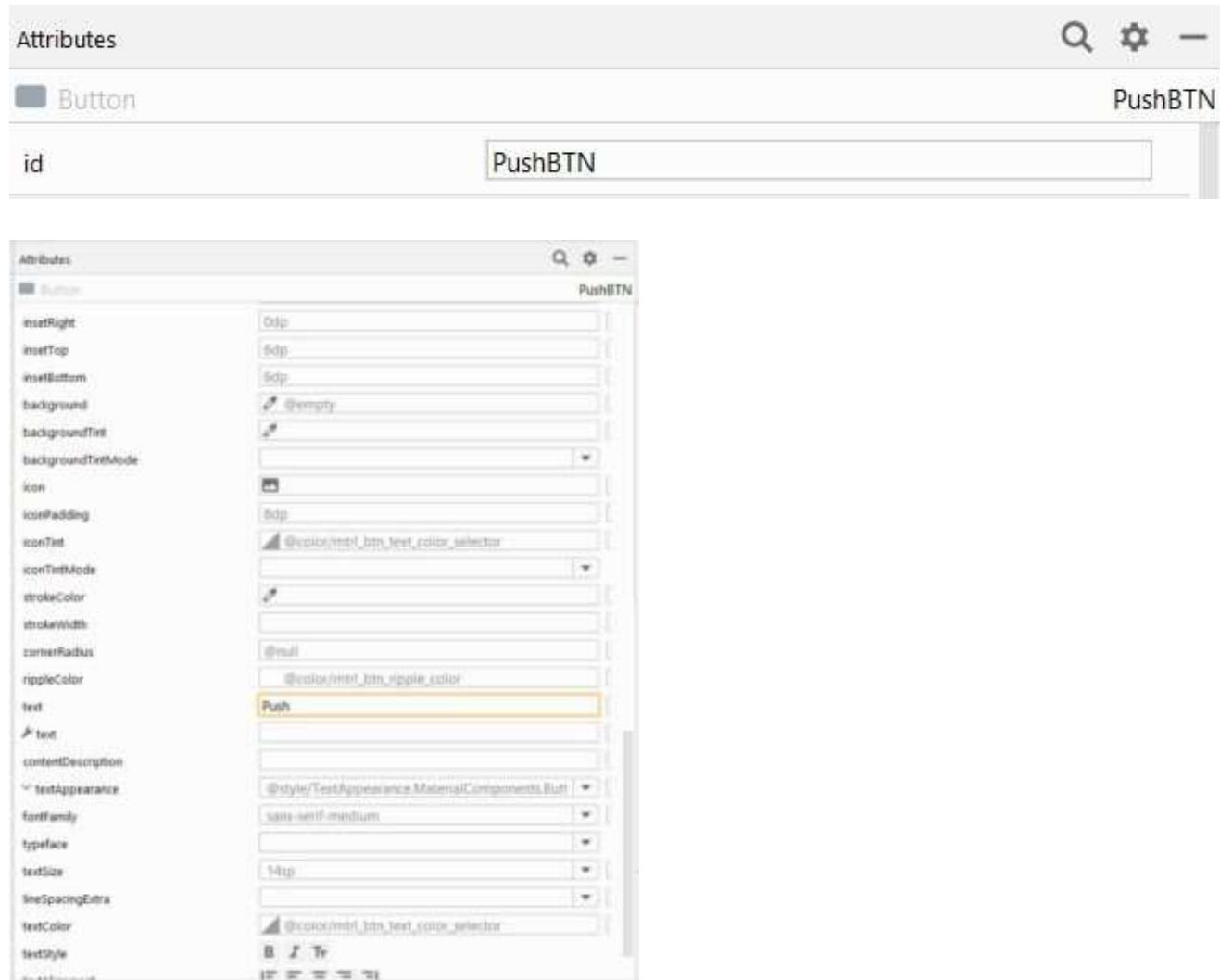
Thiết kế giao diện điều khiển trên App

Ta sẽ bắt đầu thiết kế giao diện cho app trong activity_main.xml bằng cách kéo thả các thành phần trong Palette bao gồm:

- 2 Button : Push và Get
- 1 editText: Để nhập vào dữ liệu gửi đi
- 1 textView: Để xem dữ liệu get về



Ta sẽ đặt id cho từng thành phần và tùy chỉnh kích thước chữ ở phần Attributes (ngoài ra có rất nhiều thuộc tính ta có thể thay đổi tùy vào mục đích)



Sau khi thêm bớt và chỉnh sửa giao diện ta quay về MainActivity

Trong MainActivity ta khai báo 2 biến database và mDatabase (để truy xuất đến đường dẫn vào cơ sở dữ liệu)

Khi ta thêm bớt các biến của Firebase thì Android Studio sẽ tự add các library cho chúng ta.

```
Button pushBtn, getBtn;  
EditText value;  
TextView hienthivalue;  
  
FirebaseDatabase database;  
private DatabaseReference mDatabase;
```

Lệnh truy xuất đến trường ta mong muốn: (VD ở đây là trường esp32)

```
mDatabase = database.getInstance().getReference().child("esp32");
```

Setup các trường dữ liệu và địa chỉ database

Đầu tiên chúng ta ánh xạ các nút nhấn và text vào các biến.

Gán mDatabase cho địa chỉ của firebase trên internet (dẫn tới cơ sở dữ liệu)

Push dữ liệu từ App lên Firebase.

Tại sự kiện click vào nút pushBtn chúng ta sẽ gửi dữ liệu lên esp32-string bằng cách

- getText từ ô EditText (Biến Value) sang biến data
- Gửi data đó lên database

```

// ánh xạ các nút nhấn và phần hiển thị
pushBtn = (Button) findViewById(R.id.PushBTN);
getBtn = (Button) findViewById(R.id.GetBTN);
value = (EditText) findViewById(R.id.editText);
hienthivalue = (TextView) findViewById(R.id.textView);

// dẫn tới cơ sở dữ liệu
mDatabase = FirebaseDatabase.getInstance().getReference();

// khi nhấn push
pushBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // lấy data từ editText
        String data = value.getText().toString();
        // push lên trường esp32-string trên Firebase
        mDatabase.child("esp32-string").setValue(data);
    }
});

```

Get dữ liệu từ Firebase về App

Trong sự kiện click vào nút getBtn

- Kết nối tới trường esp32 trên DB, lắng nghe sự thay đổi trên đó
- Nếu có sự thay đổi, get dữ liệu đó về và hiển thị lên TextView

```
// khi nhấn get
getBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // lắng nghe từ trường esp32 trên Firebase
        mDatabase.child("esp32").addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                // khi có data mới cập nhật thì hiển thị ra textView
                hienthivalue.setText(snapshot.getValue().toString());
            }

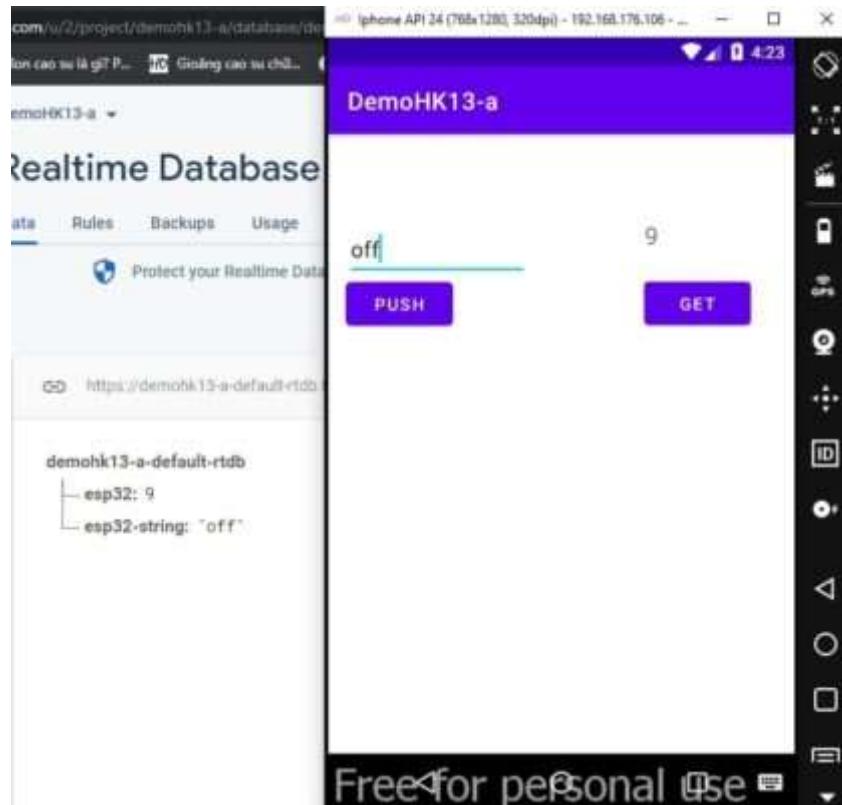
            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });
    }
});
```

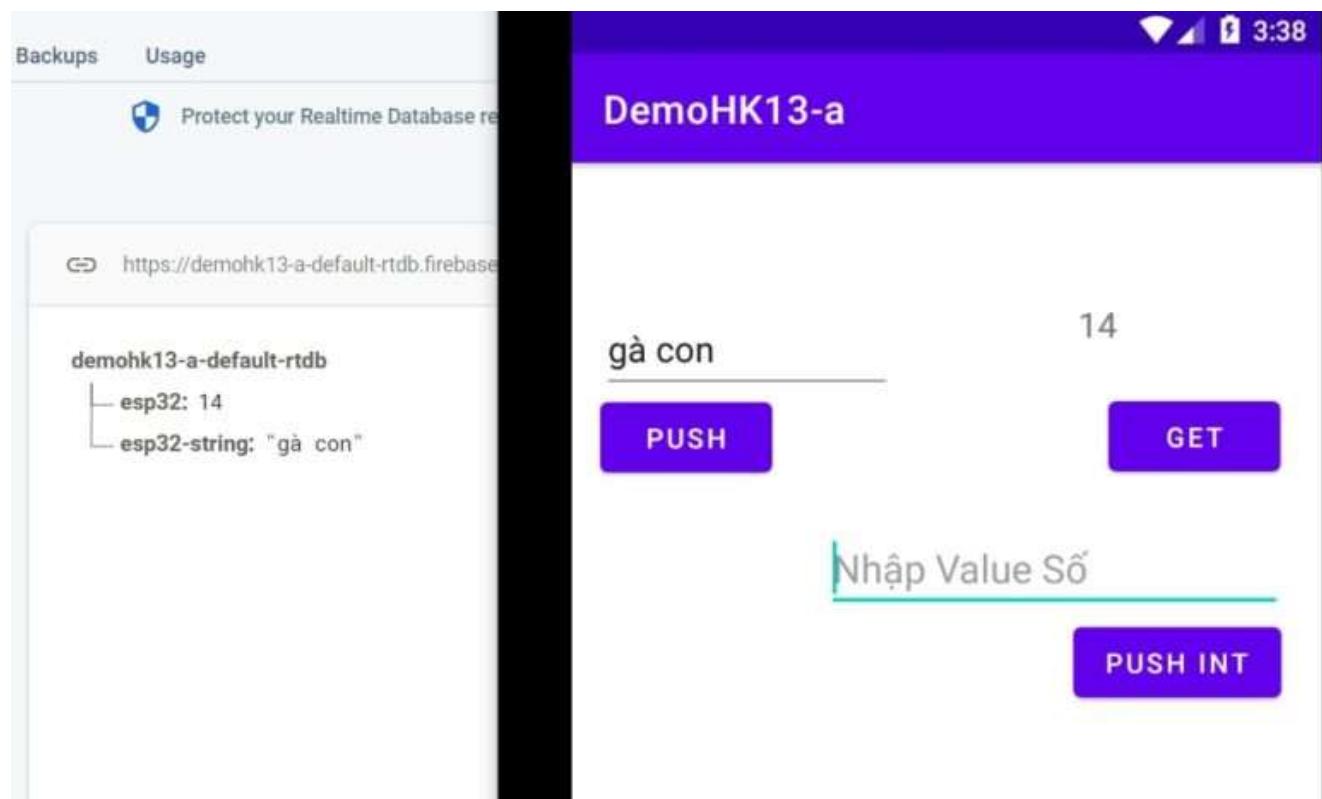
Vậy là ta đã thiết kế xong một app đơn giản để thực hiện việc push và get data từ Firebase về.

Kết Quả

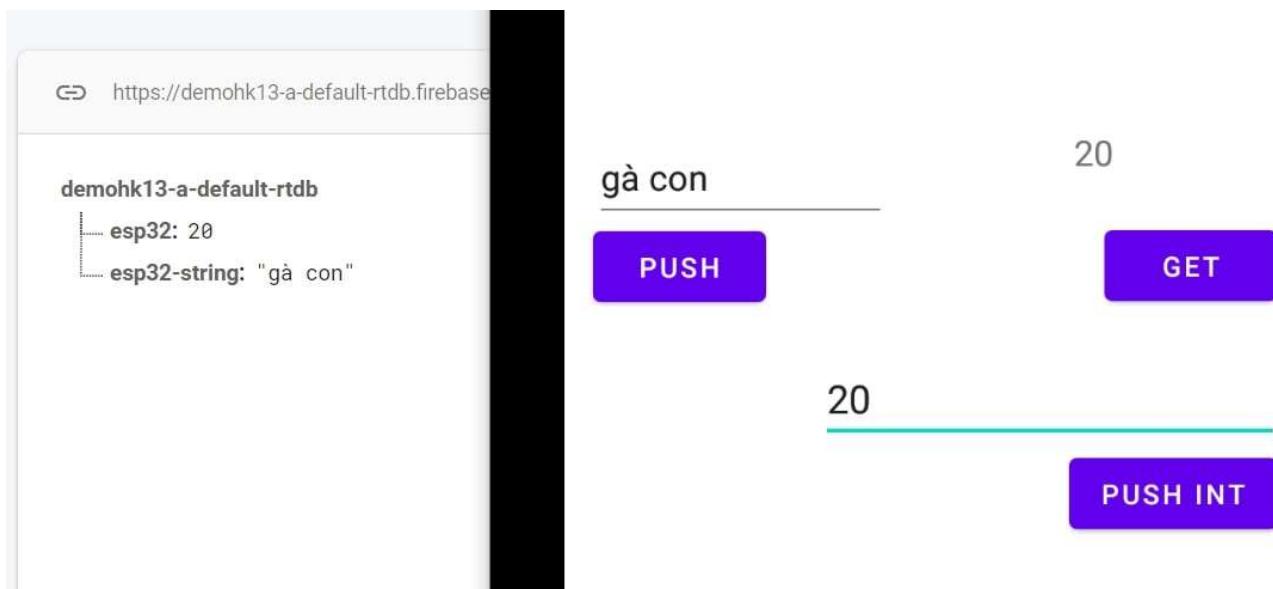
Ta thực hiện Push dữ liệu kiểu string “off” lên trường esp32-string trên Firebase



Ta thực hiện Get dữ liệu kiểu int từ Firebase về app



Và ta có thể thêm bớt tùy theo mục đích, ở đây ta thêm 1 Push Int Button để tác động đến trường esp32



Video kết quả

Không khả dụng
Video này không nhúng được do có thể chứa nội dung thuộc sở hữu của người khác.

Xem trên Facebook · Tìm hiểu thêm

Kết

Realtime Database Firebase là một công cụ đơn giản để chúng ta bắt đầu học cách làm việc với Database, không cần khởi tạo và cài đặt gì quá lằng nhăng vẫn sử dụng được. Vì vậy rất phù hợp khi các bạn mới học **Lập trình IOT**.

Hi vọng các bạn thích bài viết này, hãy chia sẻ cho bạn bè nếu cảm thấy chúng có ích nhé.

Cám ơn bạn đã đón đọc, cùng vào hội **Anh Em Nghiện Lập Trình** để cùng trao đổi nhé

5/5 - (3 bình chọn)

Related Posts:

1. [Firebase là gì? Ứng dụng của Firebase trong IOT](#)
2. [Hướng dẫn đọc Datasheet cho sinh viên điện tử và lập trình nhúng](#)
3. [Bài 7: Lập trình ESP32 Touch Pin bật tắt led với một cái chạm tay](#)
4. [Bài 6: Lập trình ESP32 Timer Millis và ngắt Timer](#)
5. [Bài 5: Lập trình ESP32 ngắt ngoài EXTI](#)
6. [Hướng dẫn cài đặt Platform IO lập trình ESP32](#)



KHUÊ NGUYỄN

Chỉ là người đam mê điện tử và lập trình. Làm được gì thì viết cho anh em xem thôi. :D

2 THOUGHTS ON “GIAO TIẾP VỚI REALTIME DATABASE FIREBASE SỬ DỤNG ESP32 VÀ APP”



Hoàng Dương Dũng says:

bài này anh nên lưu ý nếu làm theo thì nên add thư viện firebase bản 3.10.5
trở xuống chứ cao hơn code không hoạt động ạ!

03/11/2021 AT 9:20 SÁNG

TRẢ LỜI



Khuê Nguyễn says:

ok em nhé!

05/11/2021 AT 1:39 CHIỀU

TRẢ LỜI

Trả lời

Email của bạn sẽ không được hiển thị công khai. Các trường bắt buộc được đánh dấu *

Bình luận *

Tên *

Email *

Trang web

PHẢN HỒI

Fanpage



Khuê Nguyễn Creator - Họ...

2.754 lượt thích

[Đã thích](#)[Chia sẻ](#)

Khuê Nguyễn Creator - Học Lập Trình Vi Điều Khiển

khoảng một tháng trước

Lý do thời gian gần đây mình không viết bài
và làm thêm gì cả là đây 😊)

Chính thức ra mắt sản phẩm định vị thông
minh vTag.

Đây là một sản phẩm định vị đa năng với
3 công nghệ định vị WIFI, GPS, LBS kết
hợp với sóng NB-IOT dành riêng cho các
sản phẩm IOT.

Chỉ với 990.000đ chúng ta đã có thể có
sản phẩm để:

- Định vị trẻ em, con cái... [Xem thêm](#)



Bài viết khác

Lập trình 8051 - AT89S52



Khuê Nguyễn Creator



Bài 1: Tổng quan về 8051 và chip AT89S51 - 52

Tổng quan về 8051

8051 là một dòng chip nhập môn cho lập trình viên nhúng, chúng được sử...

[ĐỌC THÊM](#)

Lập trình STM32 và CubeMX





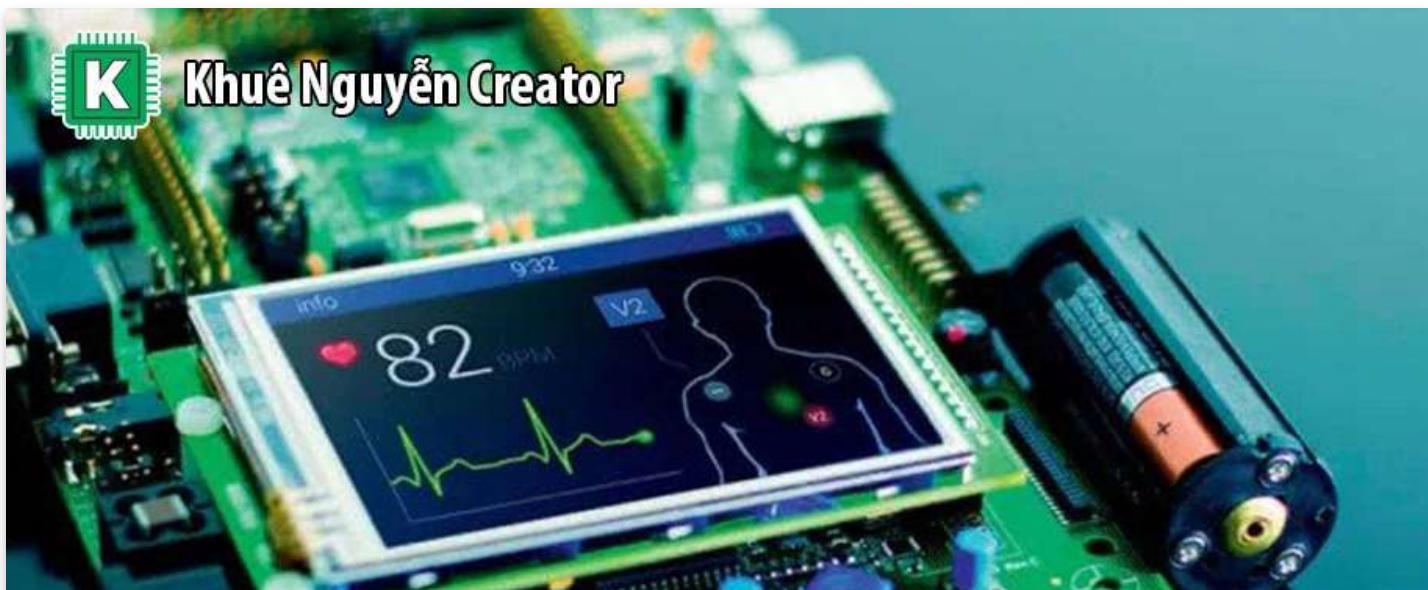
Khuê Nguyễn Creator

Lập trình STM32 HID Host giao tiếp với chuột và bàn phím

Lập trình STM32 USB HID Host giao tiếp với chuột và bàn phím máy tính

Trong bài này chúng ta sẽ cùng học STM32 HID Host, biến STM32 giống như..

[ĐỌC THÊM](#)



Lộ trình học lập trình nhúng từ A tới Z

Lộ trình học lập trình nhúng từ A tới Z

Lập trình nhúng là một ngành có cơ hội nhưng cũng đòi hỏi nhiều kiến...

3 COMMENTS

[ĐỌC THÊM](#)

Lập trình STM32 và CubeMX



Khuê Nguyễn Creator



Lập trình STM32F407 SDIO đọc dữ liệu thẻ nhớ

Lập trình STM32 SDIO đọc ghi dữ liệu vào thẻ nhớ SD card

Trong bài này chúng ta cùng học cách lập trình STM32 SDIO, một chuẩn giao...

[ĐỌC THÊM](#)

Lập trình STM32 và CubeMX



Khuê Nguyễn Creator





Lập trình STM32F407 DAC chuyển đổi số sang tương tự

Lập trình STM32 DAC tạo sóng hình Sin trên KIT STM32F407 Discovery

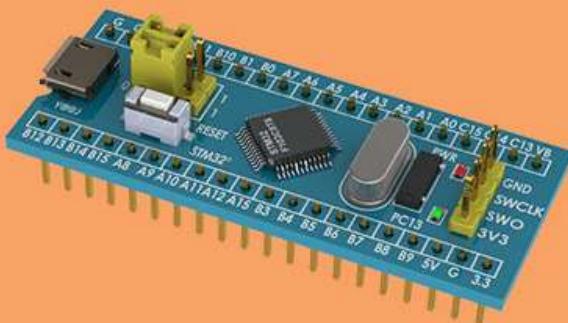
Trong bài này chúng ta sẽ cùng nhau tìm hiểu STM32 DAC với KIT STM32F407VE...

[ĐỌC THÊM](#)

Lập trình STM32 và CubeMX



Khuê Nguyễn Creator



Sử dụng hàm printf để in Log khi Debug trên STM32

Hướng dẫn sử dụng printf với STM32 Uart để in Log trên Keil C

Trong bài này chúng ta sẽ học cách retarget hàm printf của thư viện stdio...

3 COMMENTS

[ĐỌC THÊM](#)

ESP32 và Platform IO



Khuê Nguyễn Creator



Bài 9 WIFI: Lập trình ESP32 OTA nạp firmware trên Internet

Lập trình ESP32 FOTA nạp firmware qua mạng Internet với OTA Drive

Trong bài này chúng ta sẽ học cách sử dụng ESP32 FOTA (Firmware Over The...

4 COMMENTS

[ĐỌC THÊM](#)

Lập trình Nuvoton



Khuê Nguyễn Creator



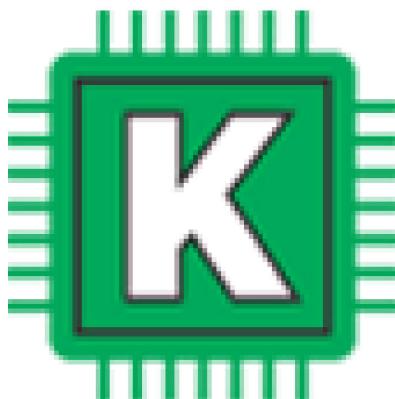
Cài đặt SDC Complier và

Code:Blocks IDE

Hướng dẫn cài đặt SDCC và Code::Blocks lập trình Nuvoton

Ở bài này chúng ta sẽ cài đặt các công cụ cần thiết cho việc...

ĐỌC THÊM



KHUÊ NGUYỄN CREATOR
Chia sẻ đam mê

Blog này làm ra để lưu trữ tất cả những kiến thức, những câu chuyện của mình. Đôi khi là những ý tưởng nhất thời, đôi khi là các dự án tự mình làm. Chia sẻ cho người khác cũng là niềm vui của mình, kiến thức mỗi người là khác nhau, không hẳn quá cao siêu nhưng sẽ có lúc hữu dụng.

Liên Kết

Nhóm: Nghịên Lập Trình

Fanpage: Khuê Nguyên Creator

My Shop

Thông Tin

Tác Giả

Chính Sách Bảo Mật



Copyright 2022 © Khuê Nguyễn