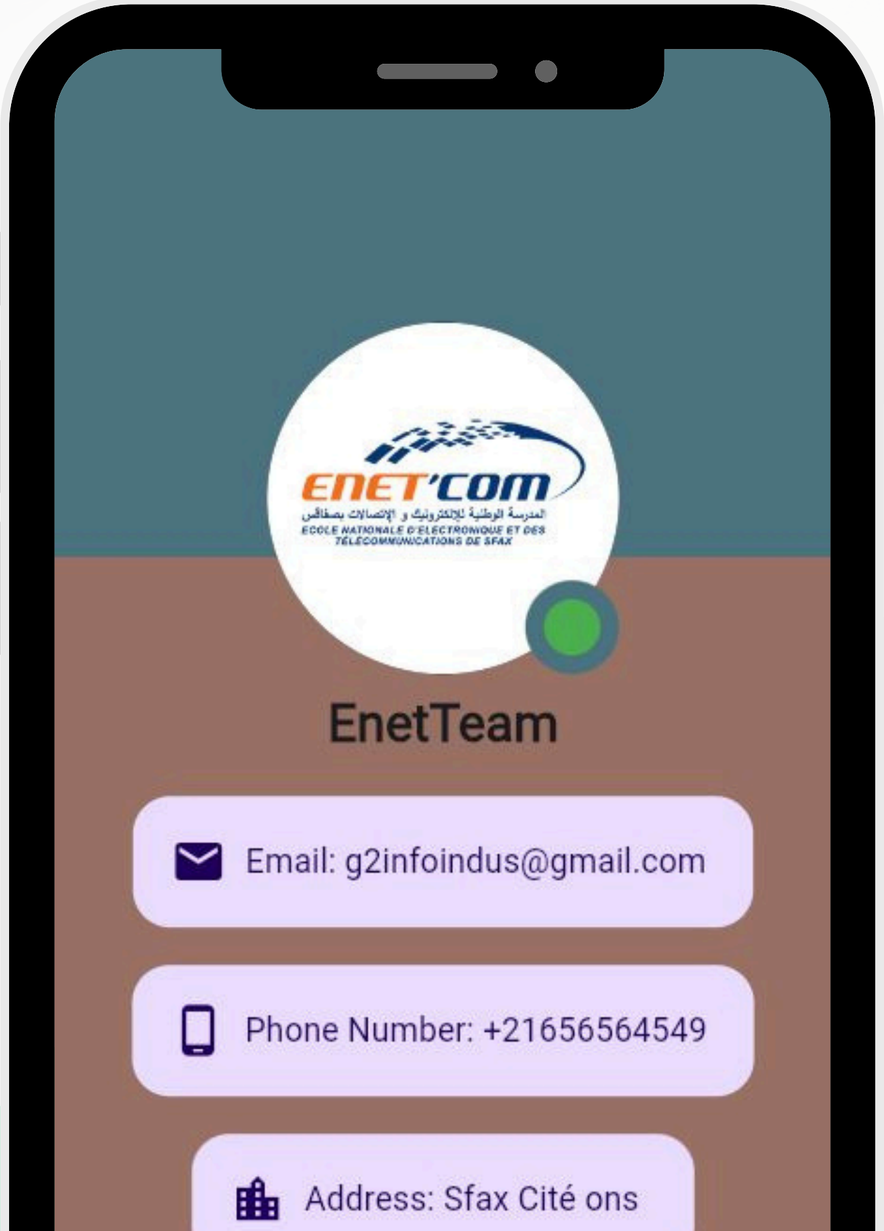
CONTROLEZ VOTRE MAISON AVEC ENETSMART HOME

elaborer par
Mohamed benhasan
Rostom benabdallah
Amal bousarsar
Azza ben yahia
sous la supervision de **Faouzi Bouchhima**

NEXT



Sommaire

01 INTRODUCTION

02 CONTEXTE TECHNOLOGIQUE

03 CONCEPTION DU CIRCUIT

04 DEVELOPPEMENT MOBILE

05 IMPLEMENTATION ET INTEGRATION

Genie Informatique Industrielle

ÉCLAIREZ VOTRE MAISON, N'IMPORTE QUAND, N'IMPORTE OÙ

Notre solution?
ENET SMARTHOME

Un système d'éclairage intelligent utilisant **Node-RED**, **MQTT**, et **Firebase**. Contrôlez vos lumières via une application **Flutter** avec une synchronisation en temps réel.

ENET SMARTHOME

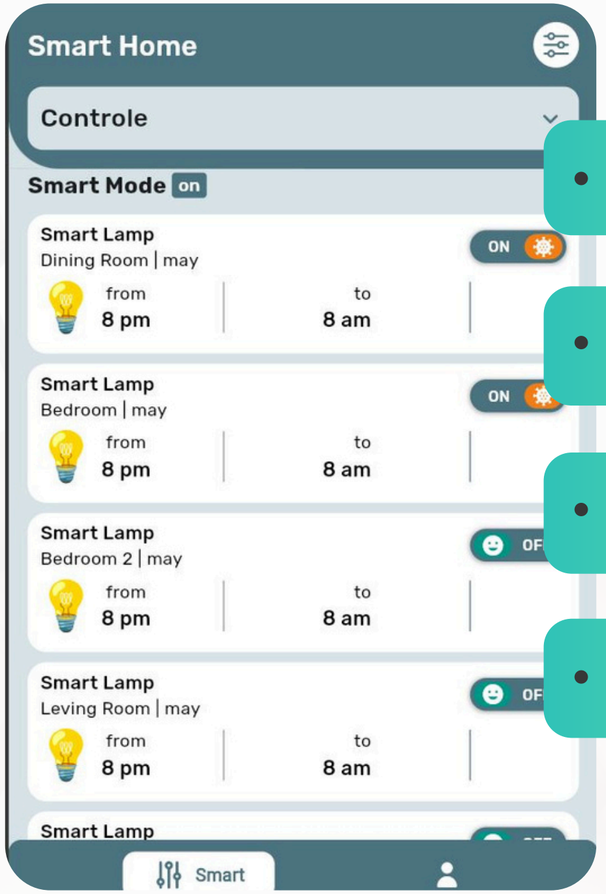
la fruit d'une combinaison des trois technologie

Genie Informatique Industrielle

INTRODUIRE NOTRE APP

Fonctions De Base

L'application appaire les modules ESP32 au Wi-Fi et contrôle les lumières : allumage, extinction . Les notifications en temps réel et les emails sont synchronisés avec **Node-red**.

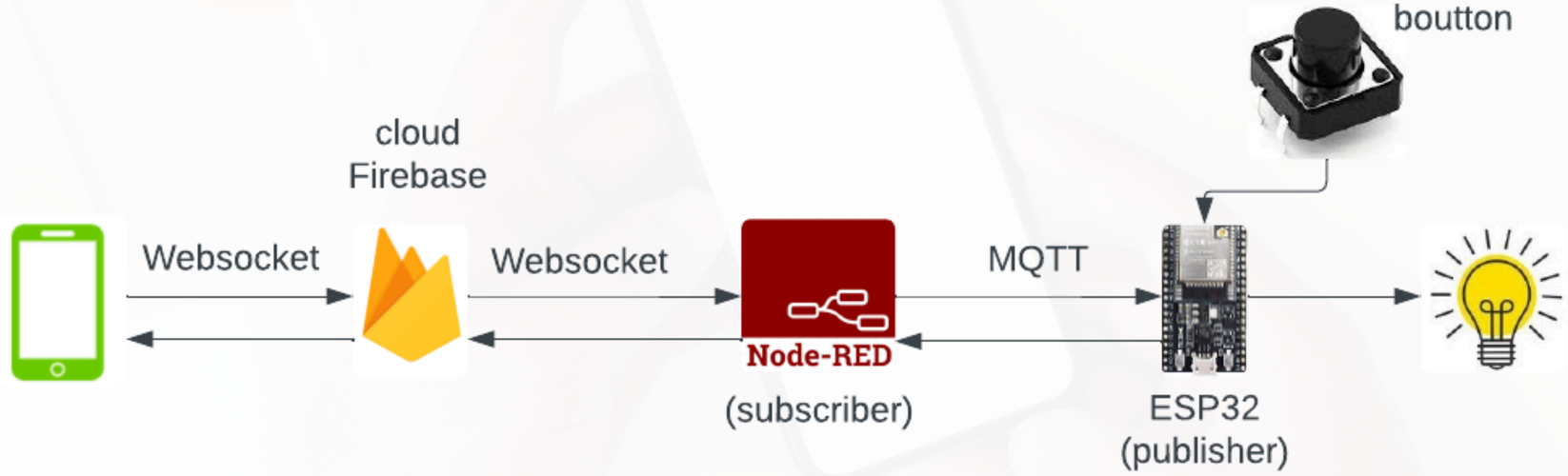


- Contrôle à distance
- Notifications en temps réel
- Synchronisation avec Firebase
- Interface utilisateur intuitive

Genie Informatique Industrielle

CONNECTION A ETABLIRE

Contexte Technologique



Application Mobile - Firebase :

L'application communique avec Firebase pour gérer en temps réel les états des LED et les commandes utilisateur.

Firebase - Node-RED :

Firebase relie l'application à Node-RED, permettant à ce dernier d'accéder aux données Firebase pour automatiser des actions et générer des notifications.

Node-RED - ESP32 :

Node-RED envoie des instructions à l'ESP32 pour contrôler les LED en fonction des événements détectés ou des commandes reçues.

ESP32 - LEDs :

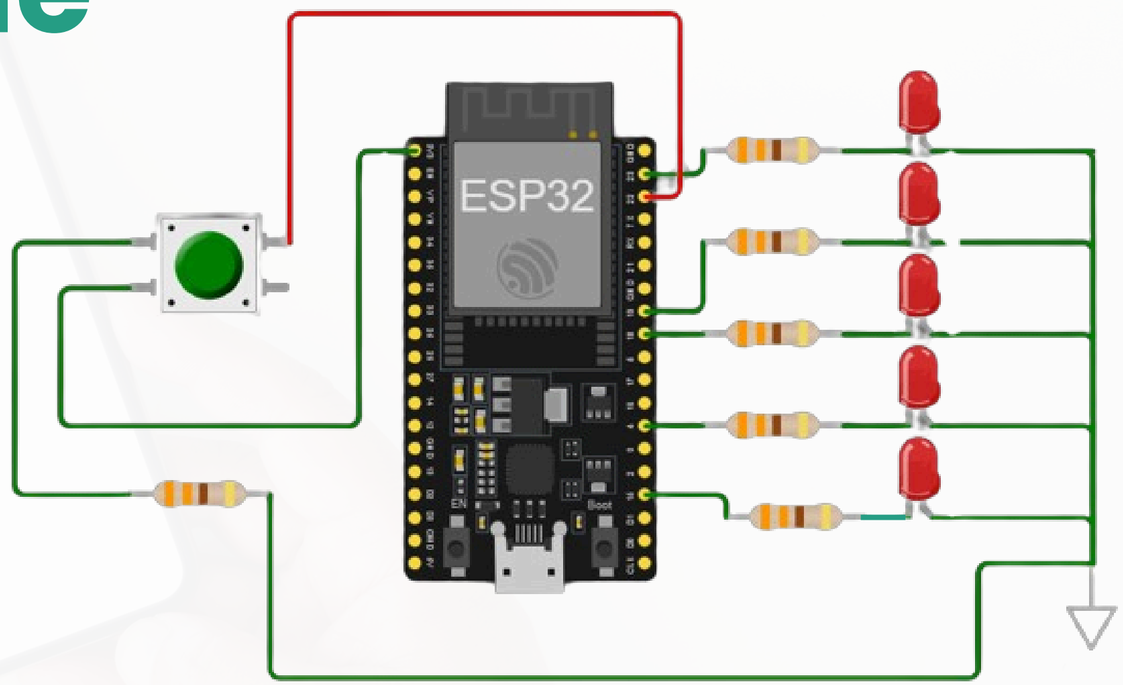
L'ESP32 contrôle directement les LED en fonction des instructions qu'il reçoit de Node-RED, assurant ainsi le bon fonctionnement de l'éclairage intelligent.

Genie Informatique Industrielle

PARTIE DE REALISATION

Conseption
Du Circuit Electrique

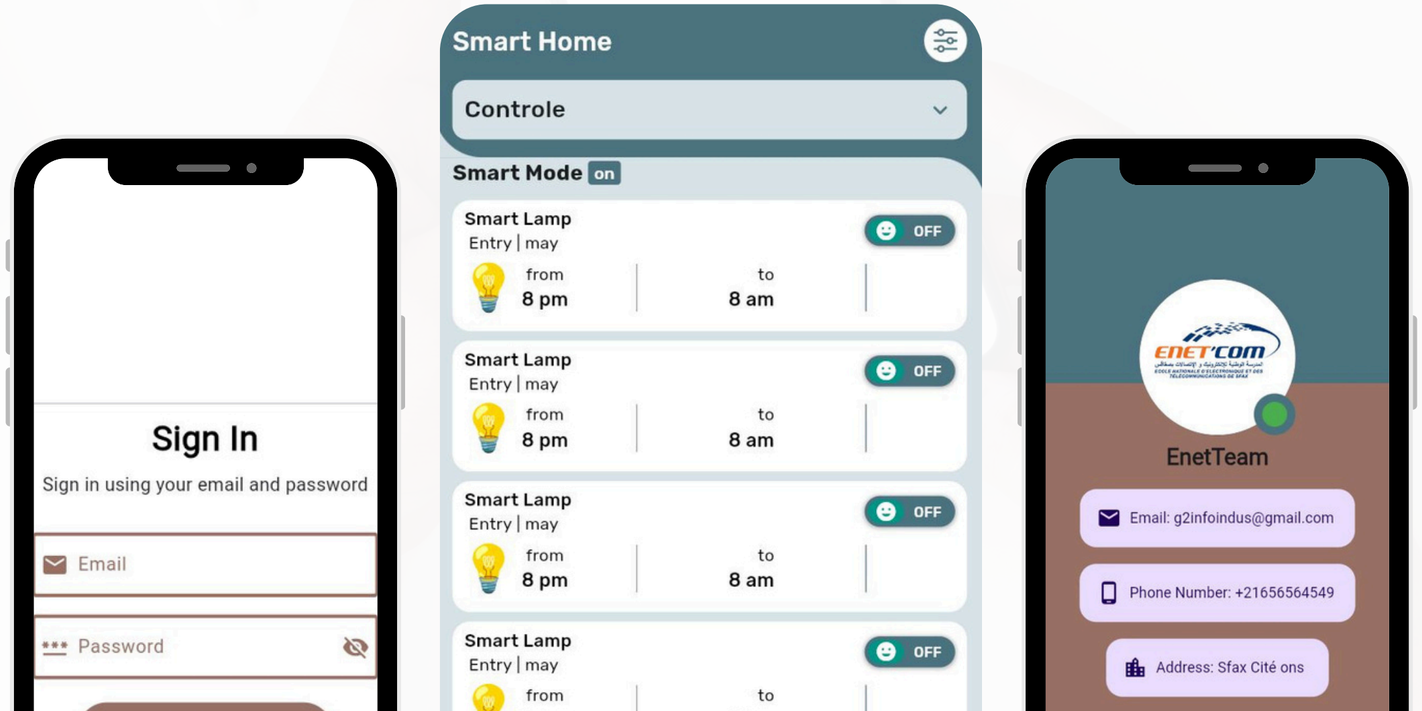
Ce schéma montre un module ESP32 connecté à un bouton poussoir et cinq LEDs, avec contrôle à distance via une application mobile. Le bouton poussoir, avec une résistance de pull-down, envoie des signaux à une broche GPIO de l'ESP32. Chaque LED est reliée à une broche GPIO via une résistance, avec les cathodes connectées à la masse. Ce montage permet de contrôler les LEDs localement avec le bouton et à distance par l'application mobile, idéal pour des projets de domotique.



Genie Informatique Industrielle

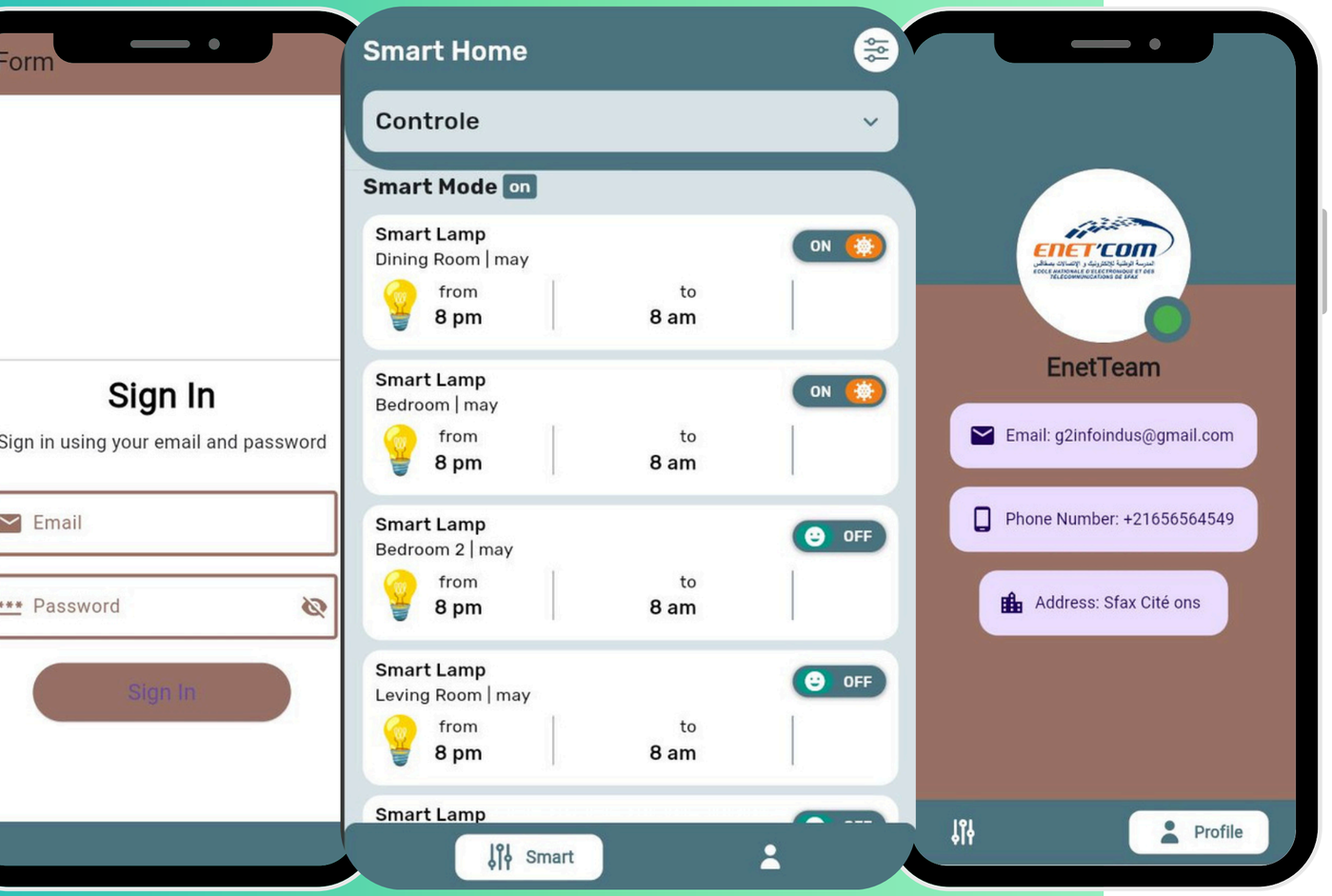
GÉREZ EFFICACEMENT LES DÉPENDANCES DU PROJET

Developpement Mobile



Genie Informatique Industrielle

Application



01 **Interface Utilisateur :**
Les captures d'écran ou les maquettes de l'interface utilisateur de l'application, montrant les boutons et les contrôles pour **activer/désactiver** les **LED**.

02 **Intégration avec Firebase :**
L'application interagit avec Firebase pour mettre à jour et synchroniser les données des **LED** et les commandes utilisateur.

03 **Authentification :**
Notre application mobile garantit un accès sécurisé aux fonctionnalités du système **IoT**, nécessitant l'entrée d'identifiants **(nom d'utilisateur et mot de passe)** pour le contrôle des LED.

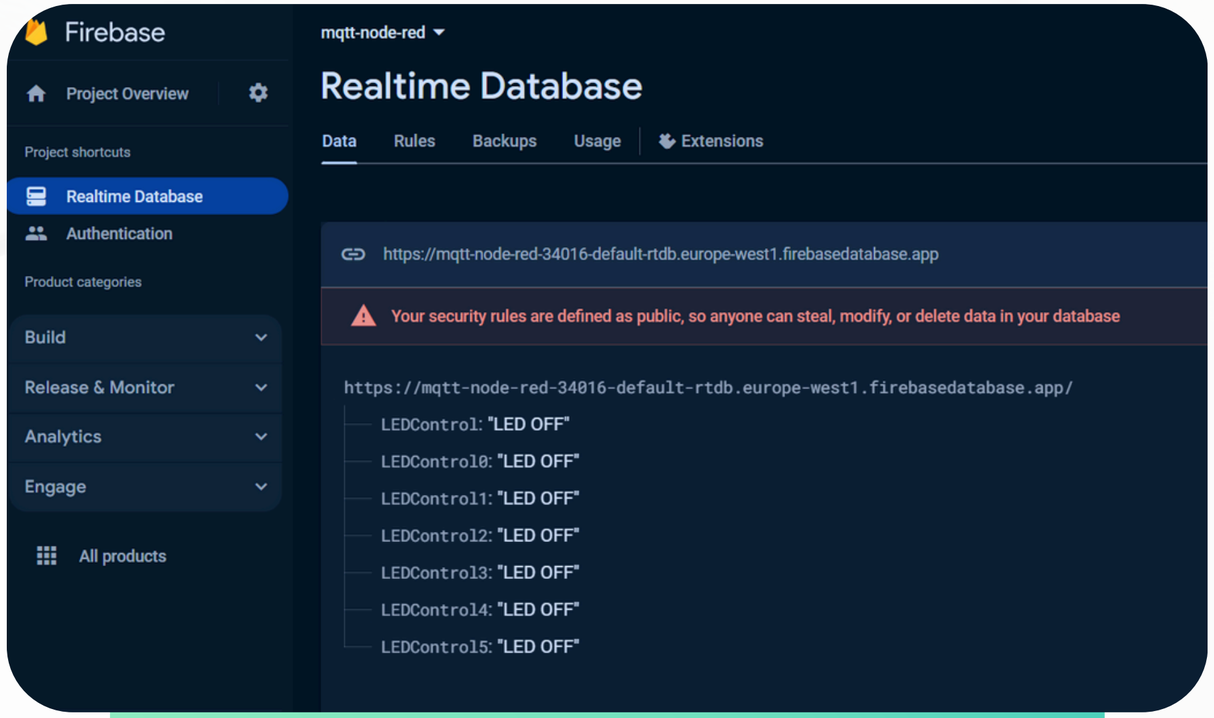
==> Firebase joue un rôle crucial en fournissant des services d'authentification sécurisés, garantissant la confidentialité des données et la fiabilité de l'expérience utilisateur.

Genie Informatique Industrielle

STOCKAGE DES INFORMATION SUR L'ETAT DES LEDS

Firebase Realtime data

Notre projet utilise Firebase pour offrir une plateforme cloud évolutive, intégrant Authentication, Data base pour la gestion des données et Cloud Messaging pour les notifications. Cette solution assure une synchronisation instantanée des données, une gestion utilisateur efficace et une expérience sécurisée et réactive.



FireBase Project

Genie Informatique Industrielle

Architecture

Dans notre projet Flutter, nous adoptons une architecture modulaire avec des couches distinctes pour la présentation, la logique métier et l'accès aux données. L'intégration de Firebase se fait à travers le fichier **pubspec.yaml** pour **les dépendances** et le fichier **build.gradle** pour la configuration spécifique à Android, garantissant ainsi une application robuste et évolutive.

```yaml
dependencies:
  flutter:
    sdk: flutter
  get:
  google_fonts:
  flutter_svg:
  eva_icons_flutter:
  cupertino_icons: ^1.0.2
  persistent_bottom_nav_bar_v2: ^4.2.8
  firebase_auth: ^4.6.3
  fl_chart:
  another_xlider: ^3.0.1
  animated_toggle_switch:
  firebase_database: ^10.4.0
  circular: ^0.0.5
```

```
SMART-LIGHTING
> .dart_tool
v android
  > .gradle
  v app
    > src
    build.gradle
    google-services (4).json
  v gradle\wrapper
    gradle-wrapper.properties
  .gitignore
  build.gradle
  gradle.properties
  local.properties
  settings.gradle
```

```
pubspec.lock        M
! pubspec.yaml      1
```

Une dépendance est un élément externe intégré dans un projet pour ajouter des fonctionnalités, utilisé via des gestionnaires de dépendances comme pub pour Flutter.

```
> assets
> build
> ios
> lib
> linux
> macos
> web
> windows
  .flutter-plugins                M
  .flutter-plugins-dependencies   M
  .metadata
  ! analysis_options.yaml
  flutter_01.png
  flutter_02.png
  flutter_03.png
  flutter_04.png
```

Genie Informatique Industrielle

CONNECTION AVEC REALTIME-DATA BASE

Changement D'etat ON/OFF

L'interaction de l'utilisateur avec le commutateur **"On/Off"** de l'application entraîne une mise à jour en temps réel de l'état des **LED** dans la base de données de Firebase

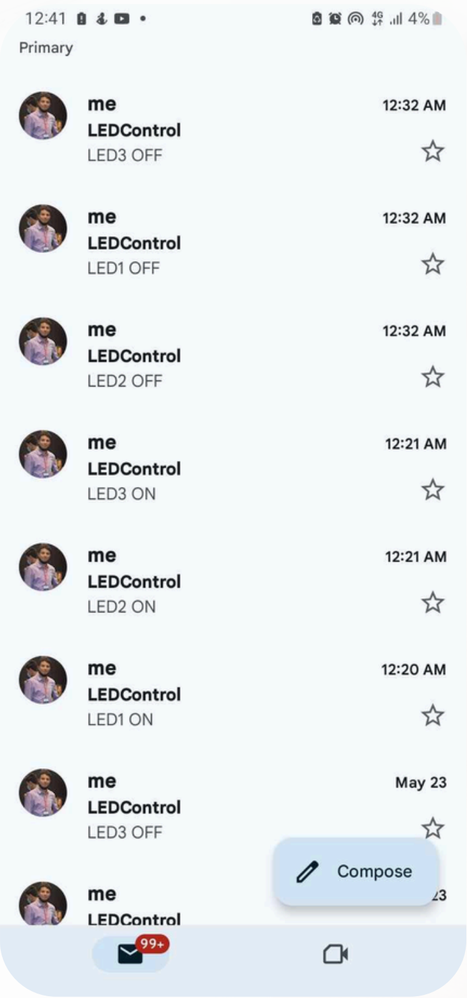
==>Cette synchronisation instantanée garantit une expérience utilisateur fluide et réactive, avec une gestion efficace des informations sur l'état des périphériques connectés.

```dart
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:firebase_database/firebase_database.dart';
```

```dart
void _updateLEDStatus(bool isOn) {
  String status = isOn ? "LED ON" : "LED OFF";
  print("Nouvel état de la LED : $status");
  _databaseReference.child("LEDControl").set(status).then((_) {
    print("Données mises à jour avec succès !");
  }).catchError((error) {
    print("Erreur lors de la mise à jour des données: $error");
  });
}
```

```dart
void checkDatabaseConnection() {
  FirebaseDatabase.instance
      .ref()
      .child("LEDControl")
      .once()
      .then((DataSnapshot snapshot) {
        if (snapshot.value != null) {
          print(
              "Connexion à la base de données en temps réel Firebase réussie !");
          print("Valeur actuelle de LEDControl : ${snapshot.value}");
        } else {
          print(
              "Échec de la connexion à la base de données en temps réel Firebase !");
        }
      } as FutureOr Function(DatabaseEvent value))
      .catchError((error) {
    print(
        "Erreur lors de la connexion à la base de données en temps réel Firebase: $error");
  });
}
```

Genie Informatique Industrielle



A CHAQUE CHANGEMENT D'ETAT

RECEPTION DU NOTIFICATION

L'application mobile notifie l'utilisateur par e-mail à chaque changement d'état des LED. Cela permet à l'utilisateur d'être informé en temps réel des modifications apportées à son système d'éclairage, même lorsqu'il n'est pas directement connecté à l'application

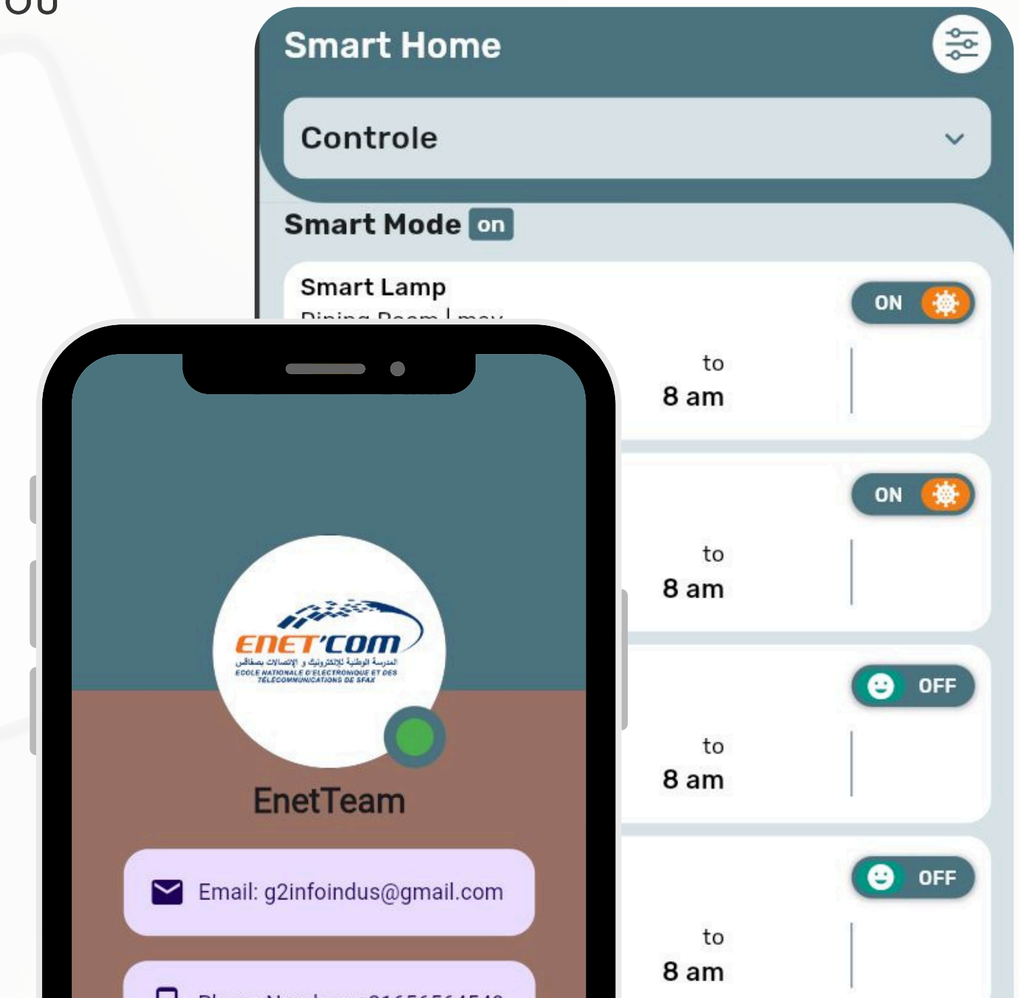
Genie Informatique Industrielle

ÉCLAIREZ VOTRE MAISON, N'IMPORTE QUAND, N'IMPORTE OÙ

ENET SMART HOME App

Notre application utilise Firebase pour une gestion en temps réel et une synchronisation parfaite des LED. Avec Authentication, Firestore, et Cloud Messaging, elle offre une expérience utilisateur fluide, sécurisée et réactive pour le contrôle intelligent de l'éclairage

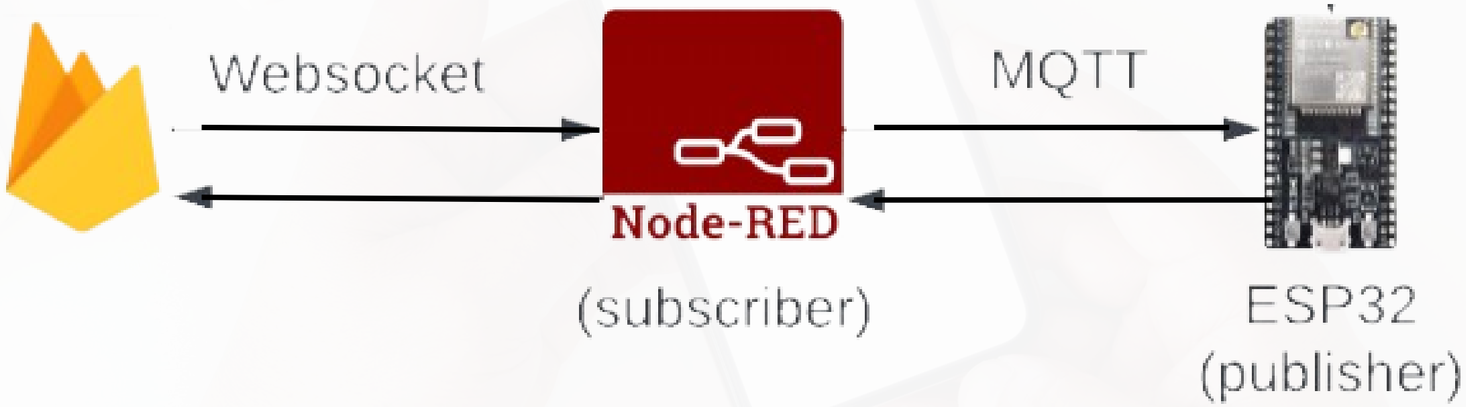
Enetcom students - GII1



Genie Informatique Industrielle

READY TO TAKE CONTROL OF YOUR FINANCES

Implémentation Et Intégration



Genie Informatique Industrielle

Esp32-Arduino IDE

Configuration WiFi et MQTT

- Définit les identifiants WiFi pour la connexion au réseau local.
- Spécifie l'adresse du serveur MQTT pour la communication.

Définition des sujets MQTT et des broches

- Assignation des sujets MQTT pour chaque LED pour recevoir les commandes.
- Attribution des broches sur l'ESP32 pour chaque LED et un bouton.

Fonction de connexion WiFi

- Établit une connexion au réseau WiFi et affiche l'adresse IP de l'ESP32.

Fonction de rappel MQTT

- Traite les messages MQTT reçus, allume ou éteint les LEDs en fonction des commandes.

Initialisation

- Configure les broches des LEDs comme sorties et la broche du bouton comme entrée.

Boucle principale

- Vérifie la connexion MQTT et traite les messages et declanche l’action .

Esp32
Arduino IDE

Après l'exécution du programme, les informations de connexion au réseau WiFi sont affichées sur la console série, confirmant la connexion réussie avec succès au réseau "Rostom" avec l'adresse IP attribuée 192.168.98.203. Ensuite, le programme tente de se connecter au broker MQTT, et le message "Attempting MQTT connection... connected" indique que la connexion MQTT a été établie avec succès.

Output Serial Monitor ×

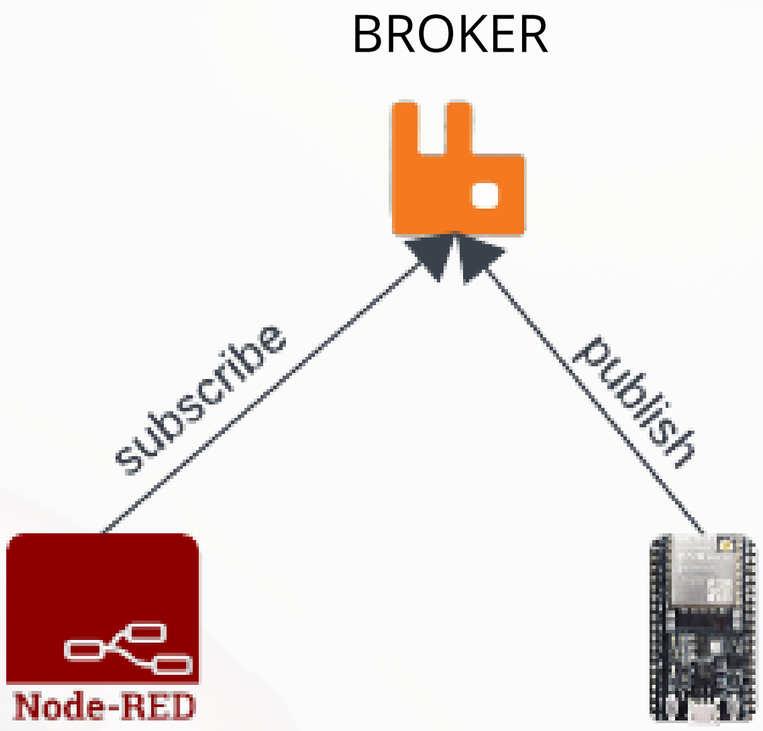
Message (Enter to send message to 'DOIT ESP32 DEVKIT V1' on 'COM7')

```
load:0x40078000,len:13260
load:0x40080400,len:3028
entry 0x400805e4
```

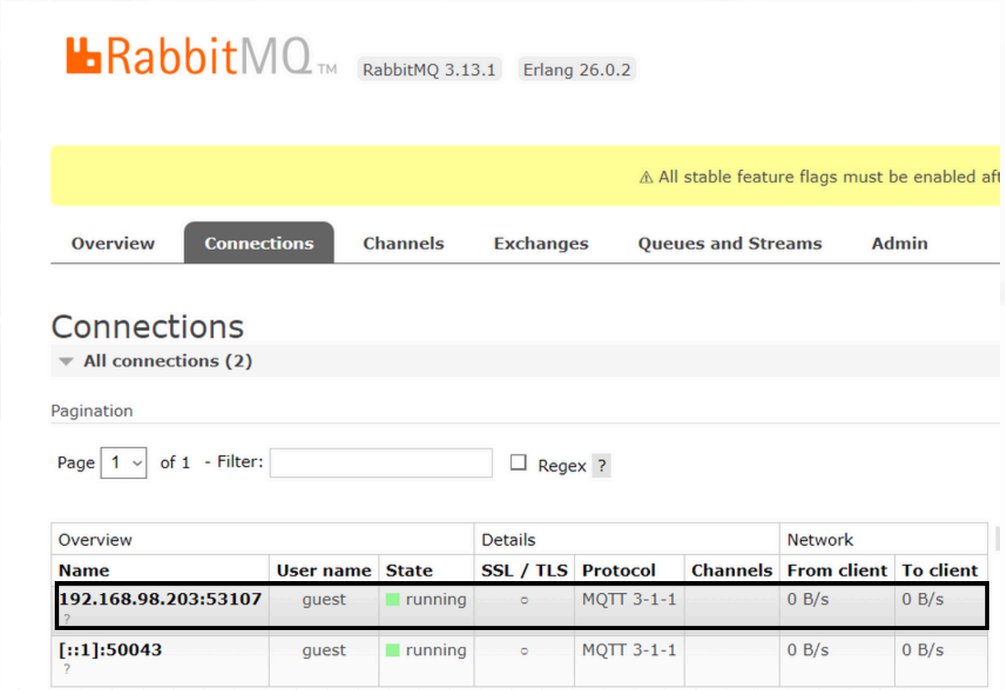
```
Connecting to Rostom
..
WiFi connected
IP address:
192.168.98.203
Attempting MQTT connection...connected
```

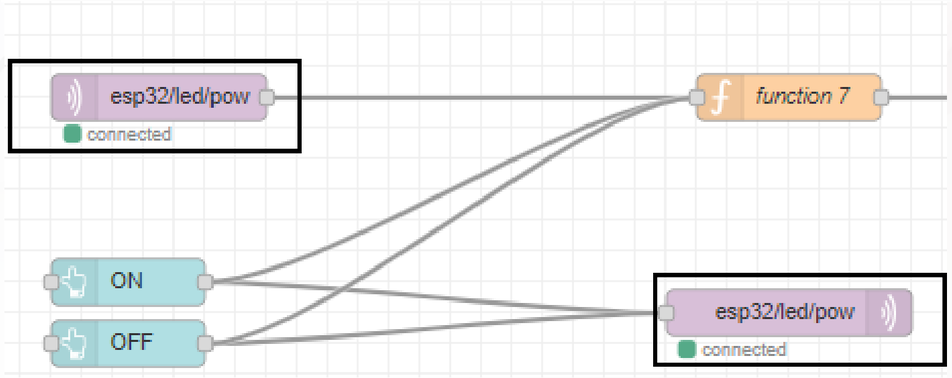
BROKER MQTT

- **RabbitMQ**(broker) assure la communication entre l'ESP32 et Node-RED.
- RabbitMQ est un système de messagerie **open-source fiable**, **rapide** et **évolutif**.
- Il est idéal pour le transfert de données entre applications distribuées.
- Il prend en charge **le protocole MQTT**, permettant un échange de messages efficace entre producteurs et consommateurs.
- RabbitMQ assure une communication robuste et ordonnée pour notre projet de contrôle des LEDs.



BROKER
NodeRed

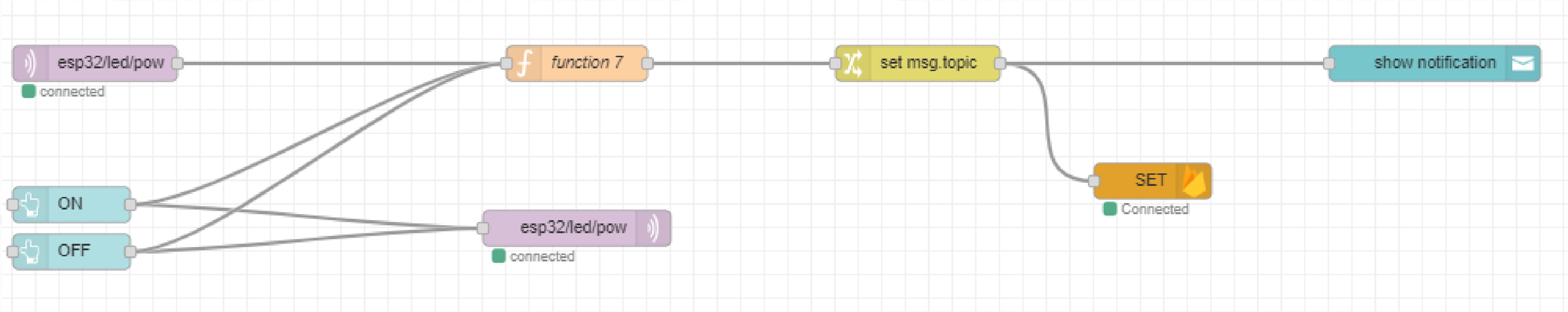




- Après l'implémentation du broker et le téléversement du code, l'ESP32 est connecté avec l'adresse IP affichée dans le moniteur série.
- Cette connexion est visible dans l'interface de RabbitMQ, confirmant l'intégration réussie avec le broker MQTT.
- L'ESP32 est également connecté à Node-RED, où son état est affiché comme "connecté".

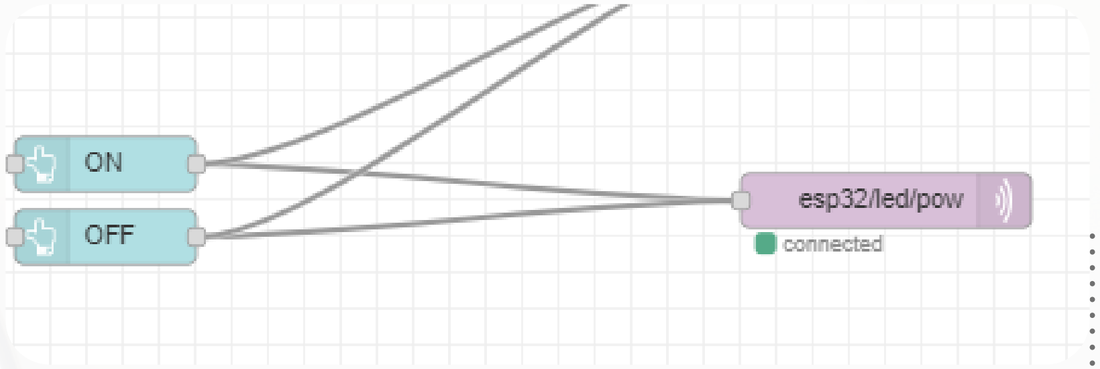
Genie Informatique Industrielle

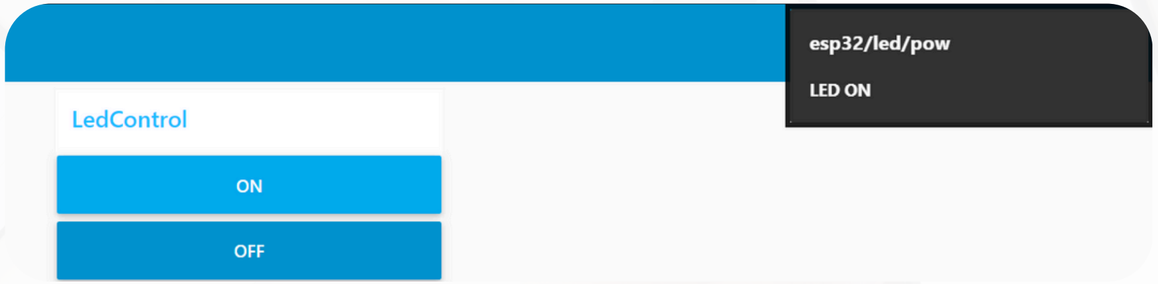
TEST COMMANDE

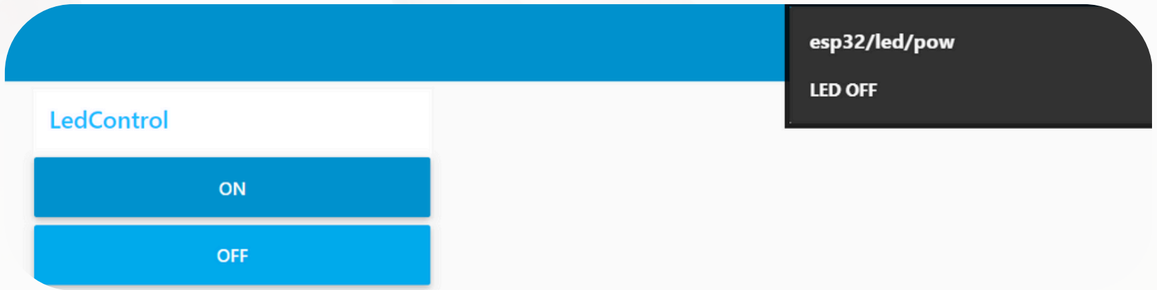


Genie Informatique Industrielle

TEST COMMANDE

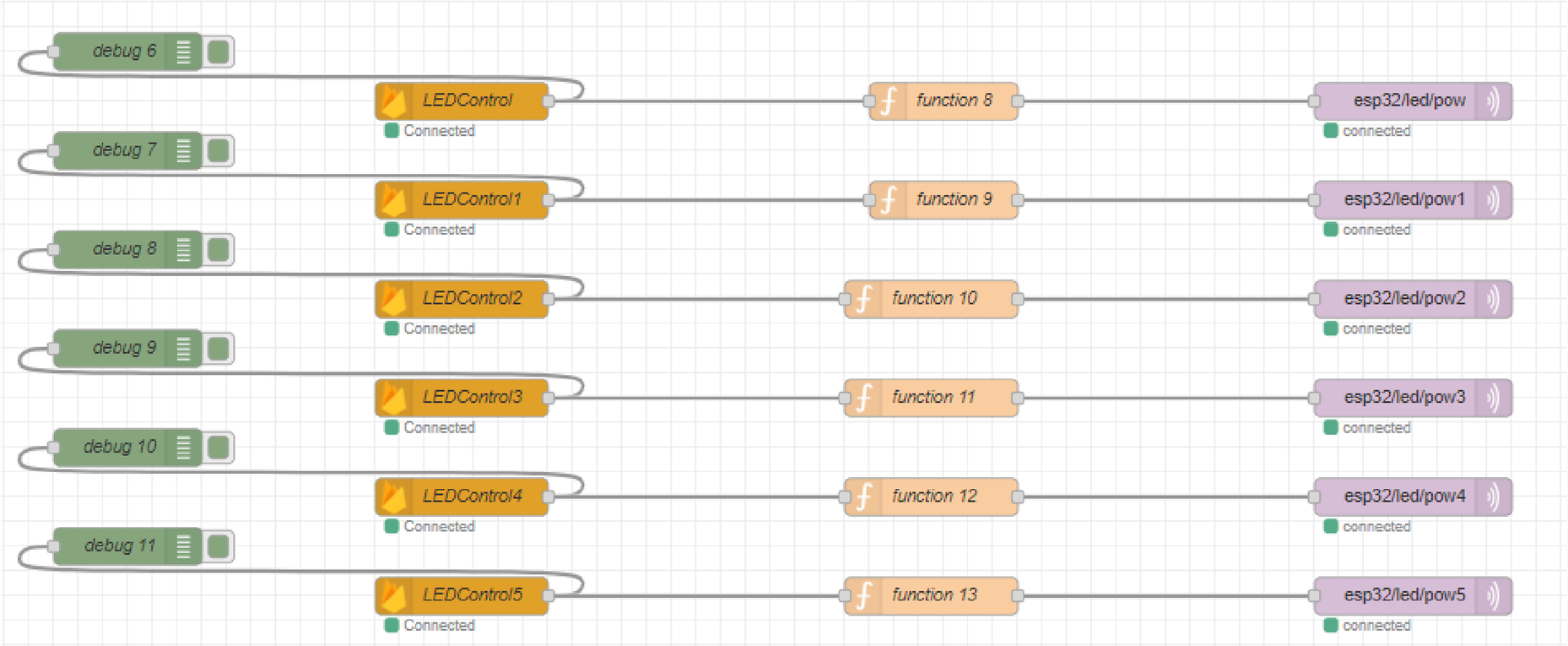






Genie Informatique Industrielle

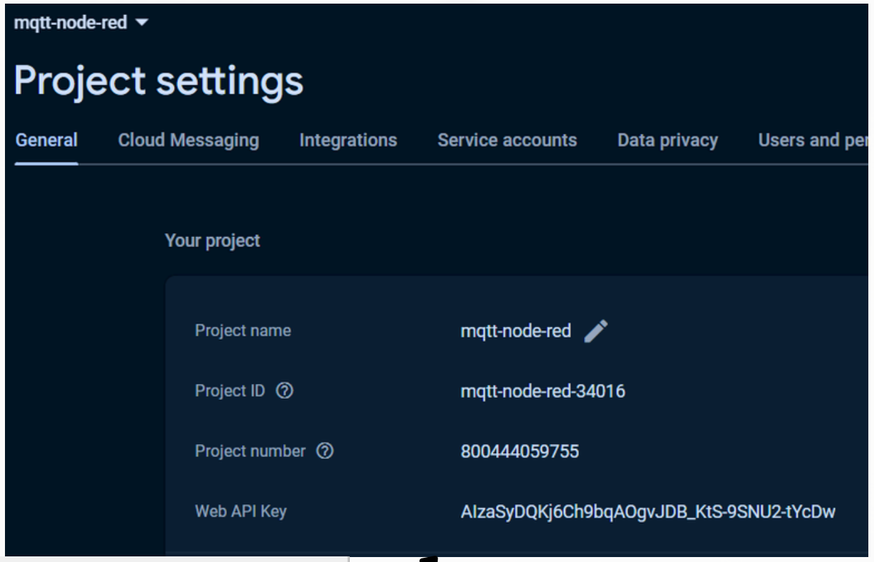
NodeRed
FireBase

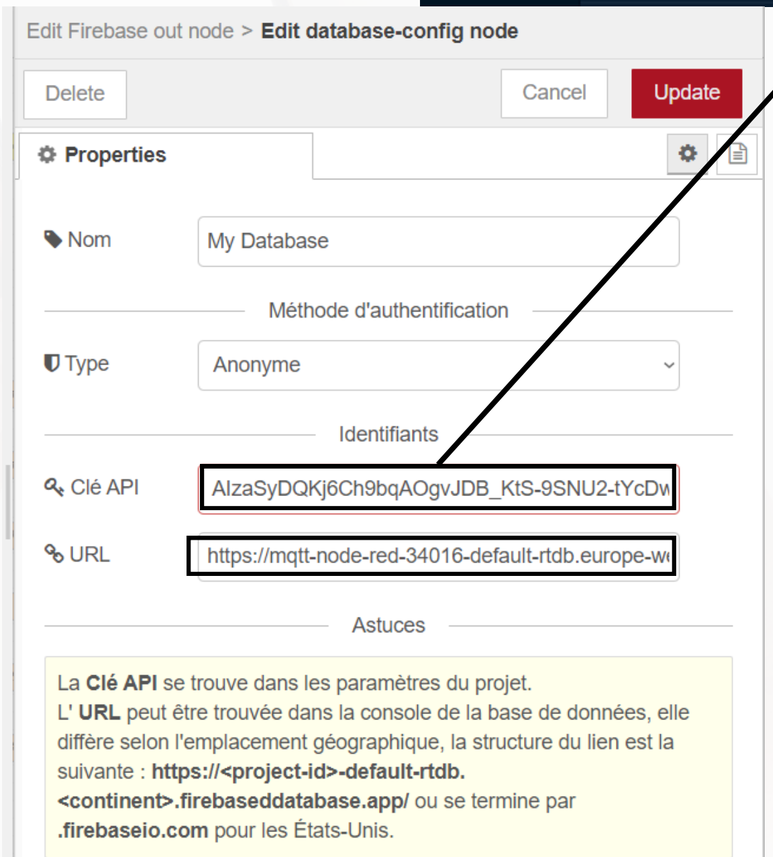


Genie Informatique Industrielle

NodeRed FireBase

- L'API Firebase a été intégrée dans notre application mobile pour gérer l'authentification des utilisateurs et les interactions en temps réel avec la base de données.
- Les données de Firebase sont reliées à Node-RED, permettant aux utilisateurs de se connecter, de contrôler les LEDs, et de voir les mises à jour instantanément via l'interface Node-RED.





Resultat Final

```
LEDControl0: "LED ON"
LEDControl1: "LED ON"
LEDControl2: "LED ON"
LEDControl3: "LED OFF"
LEDControl4: "LED OFF"
LEDControl5: "LED OFF"
```

```
Tentative de connexion MQTT...connecté
Message reçu [esp32/led/pow]: 1
LED 0 allumée
Message reçu [esp32/led/pow1]: 1
LED 1 allumée
Message reçu [esp32/led/pow2]: 1
LED 2 allumée
```

MERCI POUR

VOTRE ATTENTION