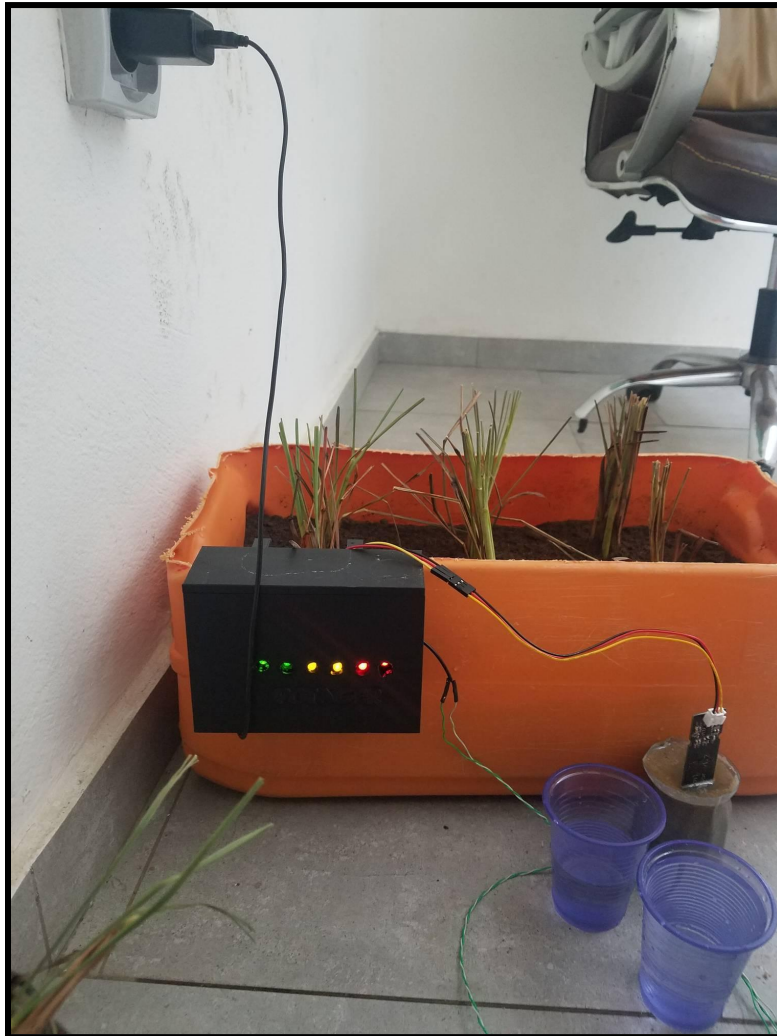


Atelier POTAGER CONNECTE

FICHE DE DOCUMENTATION DE FABRICATION

Potager connecté: Ce projet consiste à concevoir une application mobile qui donne en temps réel les informations sur l'humidité du sol et sur l'état de marche du moteur de la pompe d'arrosage automatique dans un potager connecté.



Sommaire :

- 1-Du potager au potager connecté: Qu'est-ce qu'un potager connecté ?
- 2-Description et objectifs du projet
- 3-Matériel utilisé pour la réalisation du potager connecté
- 4-Réalisation de l'électronique de commande
- 5-Modélisation du boîtier de protection du circuit avec FreeCAD et du tuyau d'arrosage
- 6-Conception de l'application mobile
- 7-Difficultés rencontrées lors de la réalisation du projet
- 8-Réalisation du potager et tests de vérification
- 9-Perspectives

1-Du potager au potager connecté: Qu'est-ce qu'un potager connecté ?

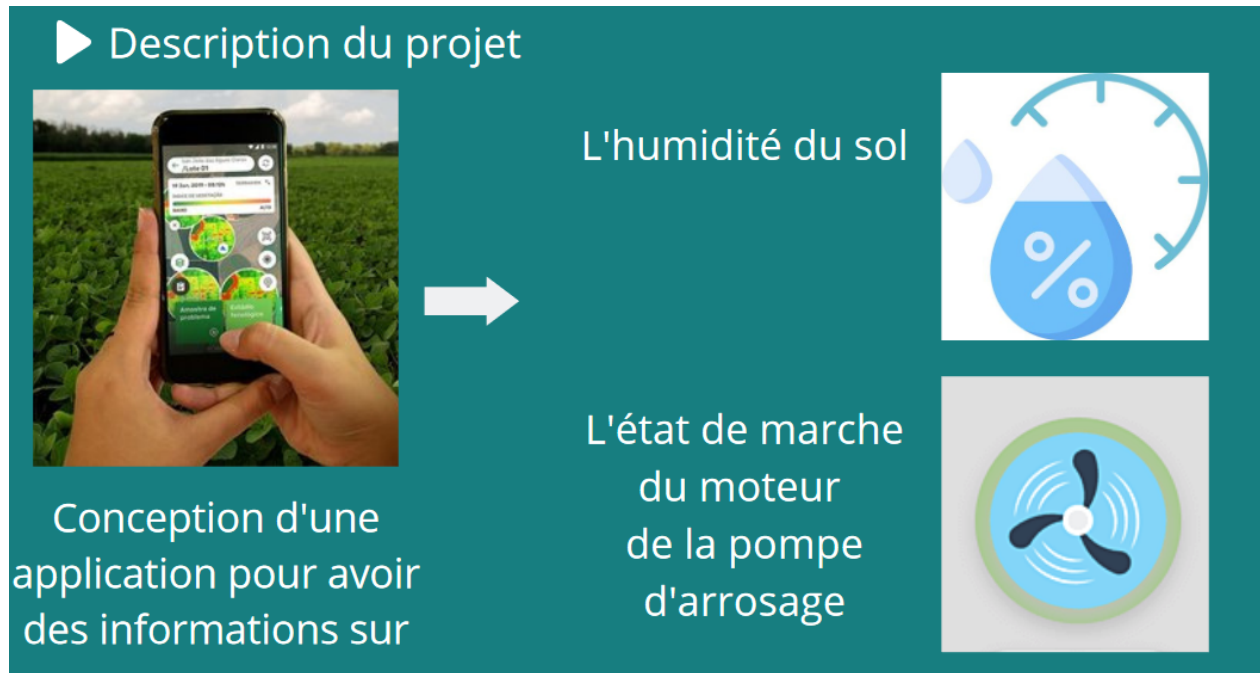


Potager: Espace dédié à la culture de légumes et de certains fruits pour la consommation



Potager connecté: Jardin intelligent permettant de contrôler des paramètres (humidité, arrosage, éclairage...) depuis son smartphone ou sa tablette...

2-Description et objectifs du projet



Les intérêts de ce type de dispositif :

- Permettre aux agriculteurs et aux personnes qui possèdent un jardin de mieux suivre l'évolution de leurs plantes
- Connaître en temps réel l'humidité du sol
- Contrôler l'allumage et l'extinction de la pompe d'arrosage
- Faire le suivi et la maintenance du moteur de la pompe
- Gagner du temps dans l'entretien des plantes

3-Matériel utilisé



Outillage	Matériaux et composants
Fer a souder Pied à coulisse Ordinateurs Mètre Multimètre Etain Pince coupante Imprimante 3D Scie	ESP8266 Leds Capteur d'humidité BreadBoard Transistor BC547 Bidon d'huile (vide) de 25L Citronnelle Pompe à eau Résistances de 220 Ohm VeroBoard

4-Réalisation de l'électronique de commande

4.1. Tests et découverte des composants

A l'aide d'un multimètre, **tester les LEDs mises à votre disposition. Pour cela:**

- Positionner le multimètre sur la fonction test de diode représentée par une flèche orientée dans un sens

- Placer les cordons de la LED sur les pattes de la LED à tester en branchant le cordon rouge à la cathode et le cordon noir à l'anode
- Si la LED fonctionne, elle brille et la valeur de la tension de seuil (en mV) se lit sur l'écran d'affichage du multimètre.
- Si la LED ne brille pas, inverser le branchement et procéder à nouveau à la vérification.

Puis on teste le micro-contrôleur ESP8266

- Tester l'ESP 8266 en utilisant l'un des programmes disponibles dans la bibliothèque Arduino ([exemple ici](#)).
- Tester le capteur en procédant comme suit:
- Effectuer le branchement suivant:

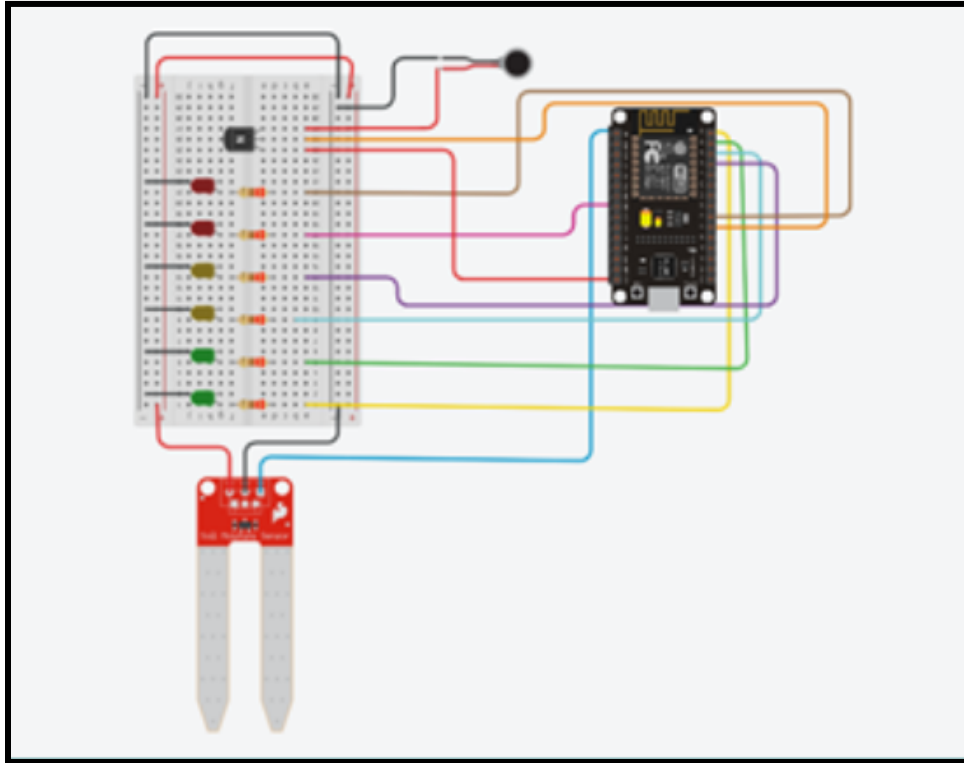
Broches du capteur	Broches de l'ESP8266
VCC / +	3.3v
GND / -	GND
AOUT	AO / ADC0

- Ecrire un programme pour relever les valeurs d'humidité puis téléverser sur la carte ESP8266 ([ici code source .ino](#))
- Introduire la sonde dans un espace humide (potager de test dans un vase ou bouteille)
- Tester le fonctionnement du capteur en lisant les valeurs d'humidité sur le moniteur Arduino.

On teste également la pompe d'arrosage en l'alimentant électriquement.

4.2.Schéma et circuit électronique

On réalise un schéma du circuit électronique à l'aide de la solution en ligne gratuite [Tinkercad](#).

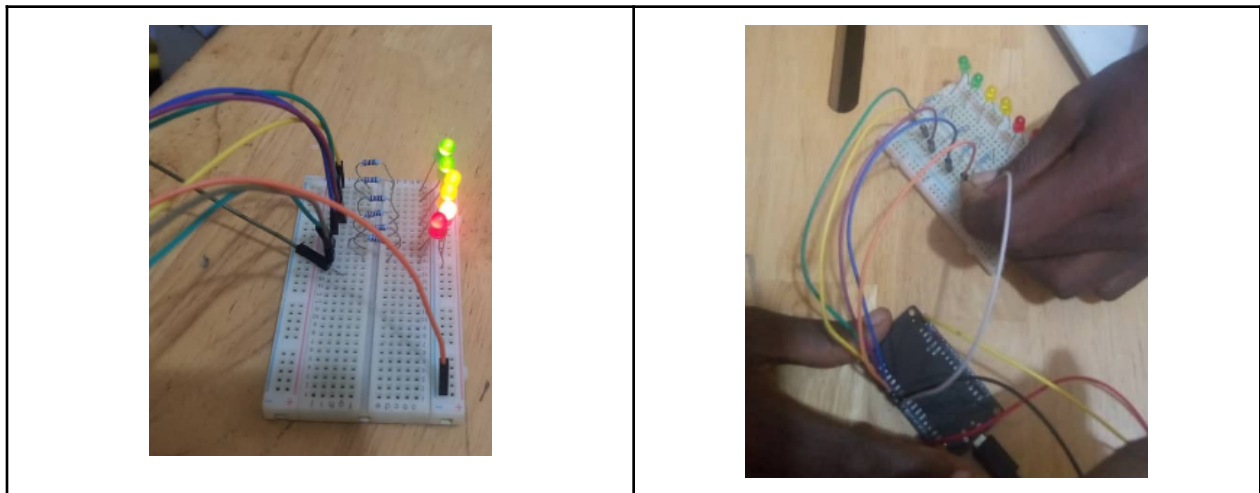


[Tinkercad](#) permet de préparer un montage de test à l'écran avec toutes les liaisons et composants.

Ici on génère un schéma de prototypage comprenant la breadboard, les leds, les capteurs d'humidité, la commande de pompe (qui fonctionne en 5V).

4.3. Réalisation de la première version sur planche de prototypage

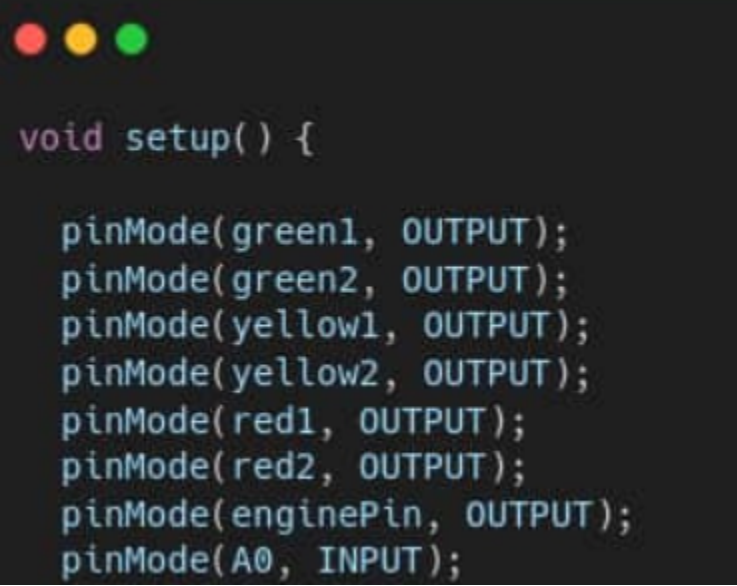
Puis on le réalise le schéma avec les fils, les composants et la planche de prototypage :



4.4 Test du code arduino sur le prototype

Ceci permet d'ajuster un code de test qui fait allumer et éteindre les leds, affiche la valeur d'humidité, allume et éteint la pompe.

[ici code source arduino](#)



```
void setup() {  
  
  pinMode(green1, OUTPUT);  
  pinMode(green2, OUTPUT);  
  pinMode(yellow1, OUTPUT);  
  pinMode(yellow2, OUTPUT);  
  pinMode(red1, OUTPUT);  
  pinMode(red2, OUTPUT);  
  pinMode(enginePin, OUTPUT);  
  pinMode(A0, INPUT);  
}
```

On voit ici que les broches de l'ESP8266 ont été renommées dans le code pour faciliter les branchements: par couleur de led, par fonction (yellow 1 pour la première led jaune, enginePin pour la pompe, A0 pour prendre la valeur d'humidité, etc).

Les leds vont servir de témoin visuel de l'humidité, les valeurs d'humidité viennent du capteur et déclenchent l'alimentation de la pompe d'arrosage.

A ce stade le prototype fonctionne et donc on va passer à la fabrication des autres éléments, et à la réalisation du même circuit, mais avec les composants soudés sur une Veroboard.

Code source développé lors de l'atelier ([langage arduino](#)) :

```
#include <Arduino.h>
```

```
#include <Firebase_ESP_Client.h>
```

```
#include <addons/RTDBHelper.h>
```

```
#include <ESP8266WiFi.h>
```

```
#define DATABASE_URL "database_url" //Remplacer database_url par l'URL de votre base de données Firebase. Ex: potager-connecte-d0c74-default-rtdb.firebaseio.com
```

```
#ifndef STASSID
#define STASSID "ssid" // Remplacer ssid par le nom de votre réseau Wifi. Ex: BLOLAB
#define STAPSK "psk" // Remplacer psk par le mot de passe de votre réseau Wifi. Ex:
Password
#endif

const char* ssid      = STASSID;
const char* password = STAPSK;

FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;

unsigned long dataMillis = 0;

#define humid A0
const int green1= 16;
const int green2 = 5;
const int yellow1= 4;
const int yellow2 = 0;
const int red1 = 10;
const int red2 = 12;
const int enginePin = 13;
float value = 0.0;
bool engineState = false;
float currentPendingTime = 0.0;
float delayBeforeTryingToReconnect = 0.0;

void connect(){
  currentPendingTime = millis();
  WiFi.begin(STASSID, STAPSK);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED)
  {
    if(millis() - currentPendingTime > 30000)
    {
      break;
    }
    Serial.print(".");
    delay(300);
  }
  if(WiFi.status() == WL_CONNECTED){

    Serial.println();
    Serial.print("Connected with IP: ");
    Serial.println(WiFi.localIP());
```

```

Serial.println();

Serial.printf("Firebase Client v%s\n\n", FIREBASE_CLIENT_VERSION);
// config.cert.file = "/cert.cer";
// config.cert.file_storage = StorageType::FLASH;
config.database_url = DATABASE_URL;

config.signer.test_mode = true;

Firebase.reconnectWiFi(true);
Firebase.begin(&config, &auth);
}else{
Serial.println();
Serial.print("Unable to connect to network");
}
}

void afficherJauge(int nbLed)
{
Serial.println(nbLed);
if(nbLed < 1 || nbLed>6) {
return;
} else{
const int leds[6] = {
red2, red1,yellow2,yellow1,green2,green1
};
for(int i=0 ;i<6; i++)
{
digitalWrite(leds[i],LOW);
}
for(int i=0 ;i<=nbLed-1; i++)
{
digitalWrite(leds[i],HIGH);
}
}
}

void setup() {
// put your setup code here, to run once:

pinMode(green1, OUTPUT);
pinMode(green2, OUTPUT);
pinMode(yellow1, OUTPUT);
pinMode(yellow2, OUTPUT);
pinMode(red1, OUTPUT);
pinMode(red2, OUTPUT);
pinMode(enginePin, OUTPUT);

```



```
pinMode(A0, INPUT);
Serial.begin(9600);
connect();
delayBeforeTryingToReconnect = millis();
}

void loop() {
value=analogRead(humid);
Serial.println(value);
Firebase.RTDB.setFloat(&fbdo, F("/Humidite"), value);
if(value >= 700){
  afficherJauge(1);
  if(!engineState){
    engineState = true;
    digitalWrite(enginePin, 1);
  }
} else if(value >= 620){
  afficherJauge(2);
  if(!engineState){
    engineState = true;
    digitalWrite(enginePin, 1);
  }
} else if(value >= 590){
  afficherJauge(3);
} else if(value >= 560){
  afficherJauge(4);
} else if(value >= 556){
  afficherJauge(5);
} else{
  afficherJauge(6);
  if(engineState){
    engineState = false;
    digitalWrite(enginePin, 0);
  }
}

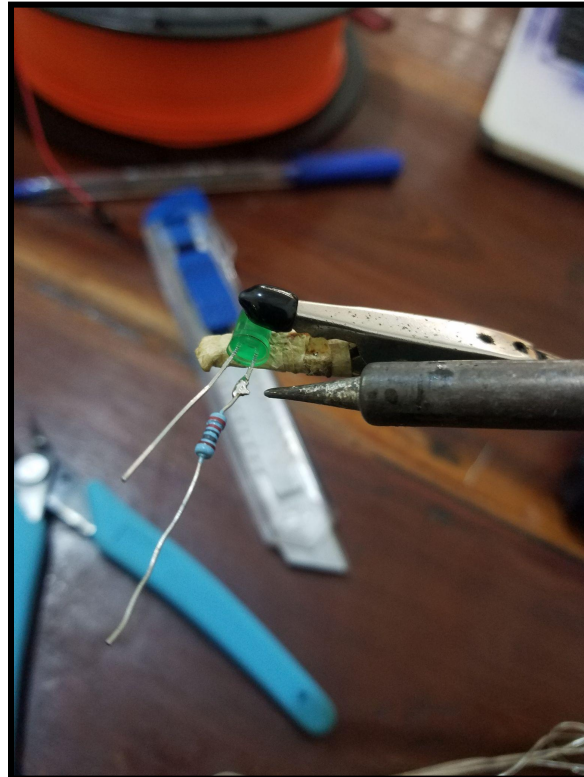
  if(WiFi.status() != WL_CONNECTED){
    if(millis() - delayBeforeTryingToReconnect > 60000){
      connect();
      delayBeforeTryingToReconnect = millis();
    }
  }
}
delay(1000);

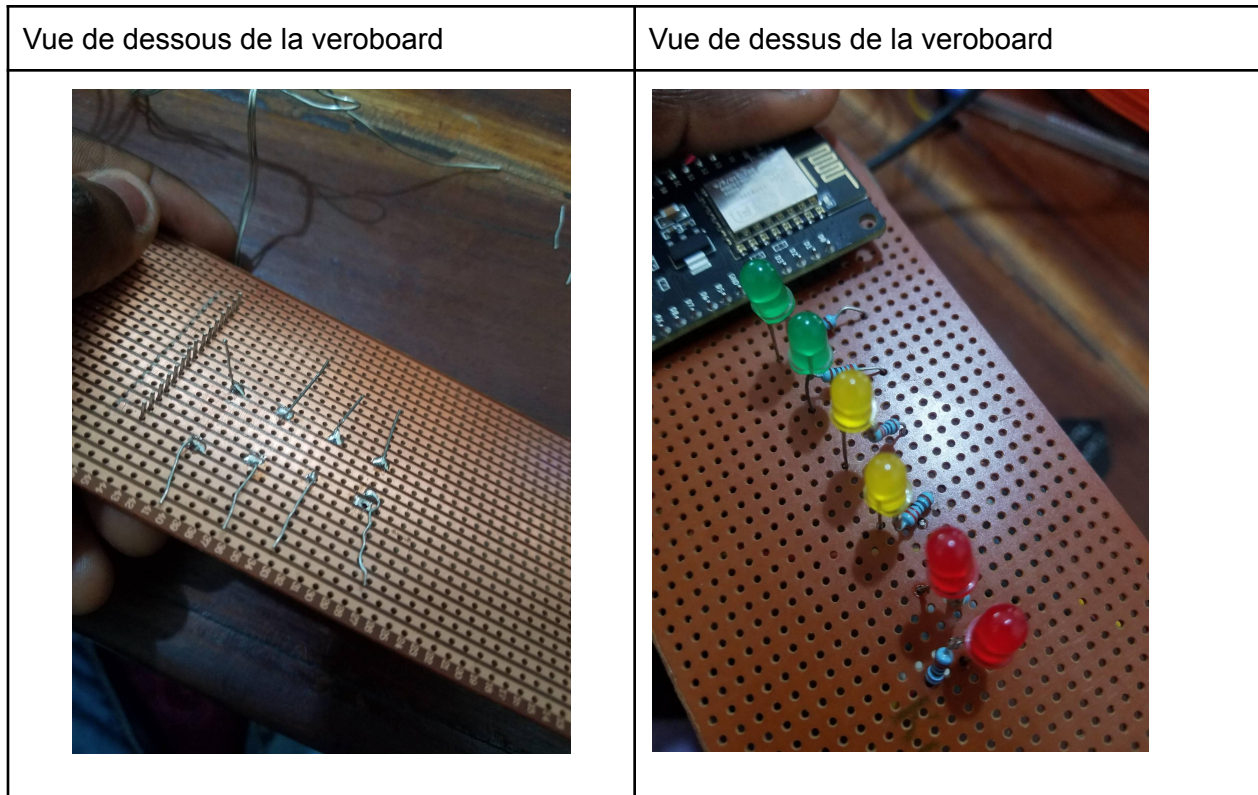
}
```

4.5. Soudure du circuit sur une veroboard afin d'avoir une version définitive solide

On passe à la phase d'initiation à la soudure des composants, toujours à bien superviser pour éviter les brûlures et risques incendies.

Avant de souder les LEDs au Véroboard, il faut prendre le soin de souder les résistances aux anodes des LEDs



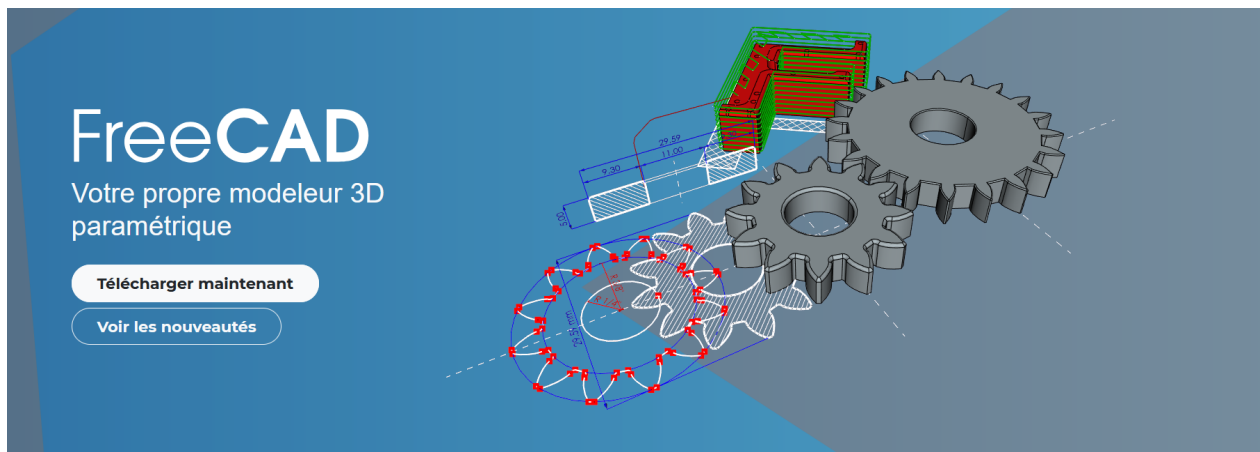


A la fin de cette phase, qui doit être réalisée avec application, on dispose d'un circuit final.

Reste à lui créer un boîtier de protection conçu pour anticiper l'accrochage sur le potager et les orifices nécessaires aux raccordements vers le capteur d'humidité, l'alimentation électrique, et bien entendu le passage des Leds témoins.

Soit vous réalisez ce boîtier sur la base de mesures avec l'adaptation d'une boîte en plastique et en mode récupération, soit, comme nous, vous en profitez pour faire une découverte de la modélisation et de l'impression 3D...

5-Modélisation du boîtier de protection du circuit et du tuyau d'arrosage avec FreeCAD



[FreeCAD](#) est un logiciel libre et gratuit très puissant de modélisation 3D. On va l'utiliser pour concevoir deux éléments :

- **Le boîtier de protection** et d'accrochage de l'électronique
- **Un raccord d'arrosage** permettant d'alimenter les plans de citronnelles qui sont à intervalle régulier.

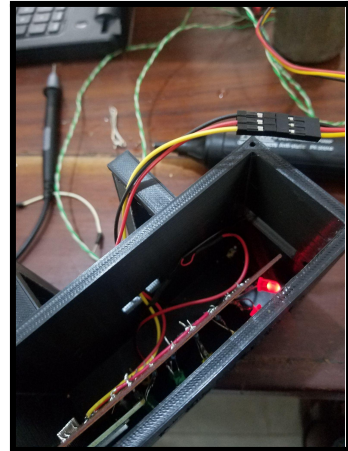
Si vous ne disposez pas d'imprimante 3D, vous pouvez réaliser un boîtier sur base de récupération/transformation d'une boîte ou un bidon plastique.

Les fichiers modèles du boîtier de protection et des canalisations d'arrosage modifiables avec le logiciel freeCAD [sont disponibles dans ce répertoire compressé au format .zip](#).

Une fois les adaptations faites, exportez au format .stl pour l'impression 3D.

Attention au temps d'impression 3D qui est souvent très long : lancez votre impression le matin pour la fin de journée...

Une fois le boîtier imprimé, insérez-y **délicatement** le système de contrôle précédemment réalisé et faites bien attention aux passages des fils permettant l'alimentation électrique.



6-Conception du système de stockage des données

Dans le but de réaliser un travail présentable dans les délais, nous avons opté pour l'utilisation d'une application expérimentale reliée à une base de données Firebase.

6.1 - Créer une base de donnée avec Firebase

Firebase est une plateforme de développement d'applications lancée en 2012 et rachetée par Google deux ans plus tard. À ses débuts, Firebase était pensé comme une base de données pour les applications en temps réel, mais Google a vu son potentiel et a décidé d'y ajouter des services supplémentaires.

Actuellement, Firebase est un BaaS (backend as a service) système avec 18 services pour faciliter la création d'applications web et mobiles. Parmi les entreprises utilisant les services BaaS de Firebase figurent Accenture, Alibaba Travels, Stack, Twitch et Instacart, ainsi que plus de 2,300 autres.

6.1. Avantages de l'utilisation de Firebase

Le premier des services proposés par Firebase était sa base de données en temps réel, et il reste l'un de ses plus gros attraits. Les bases de données en temps réel Firebase sont hébergées dans le cloud, stockant les données dans JSON formatage et synchronisation en temps réel avec chaque client qui leur est connecté. Que vous utilisiez le SDK iOS, le SDK

Android ou le SDK JavaScript, toutes les applications connectées à une base de données Firebase Realtime partagent une instance de la base de données, toujours à jour avec les dernières données.

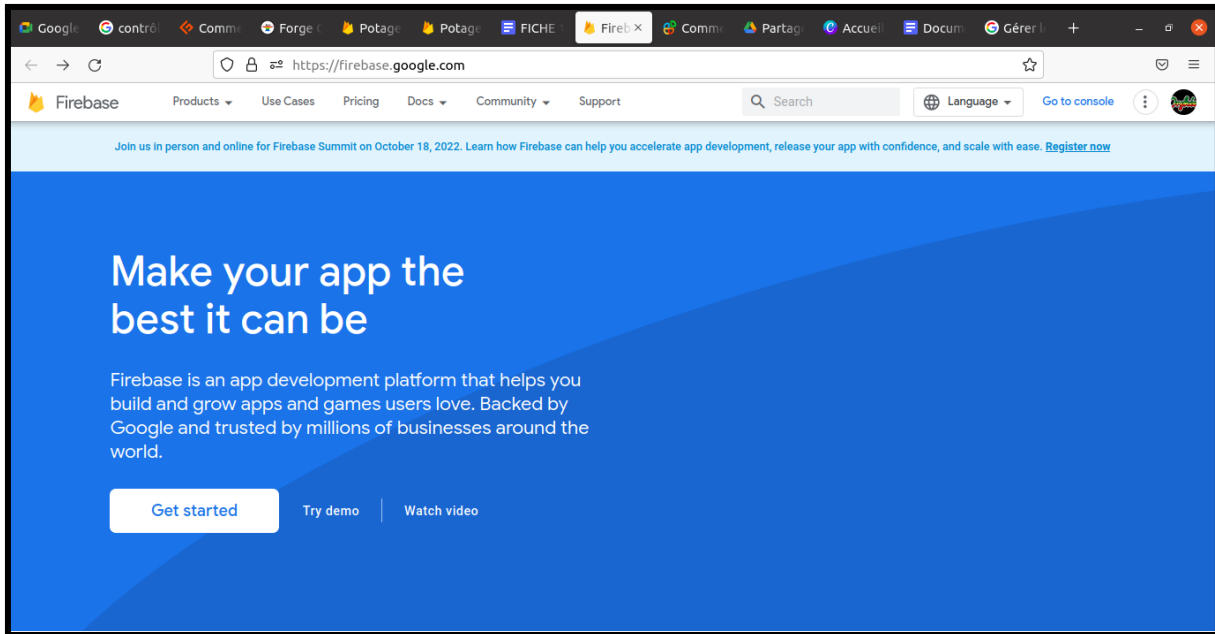
Cloud Firestore est un autre service Firebase intéressant. C'est un NoSQL base de données de documents conçue pour faciliter le stockage, la synchronisation et l'interrogation des données pour les applications mobiles et Web à l'échelle mondiale. La création de hiérarchies pour stocker les données associées et les requêtes expressives pour récupérer les données permettent de réaliser tout le potentiel de Cloud Firestore. À leur tour, les requêtes évoluent en fonction de la taille des résultats plutôt que de la taille de l'ensemble de données. Cela permet aux applications d'évoluer dès le début sans attendre le moment où les besoins dépassent la capacité.

En plus des services de base de données susmentionnés, Firebase propose également services d'hébergement, stockage de fichiers, fonctions (Style AWS Lambda), parmi beaucoup d'autres choses.

6.2.Création d'une API

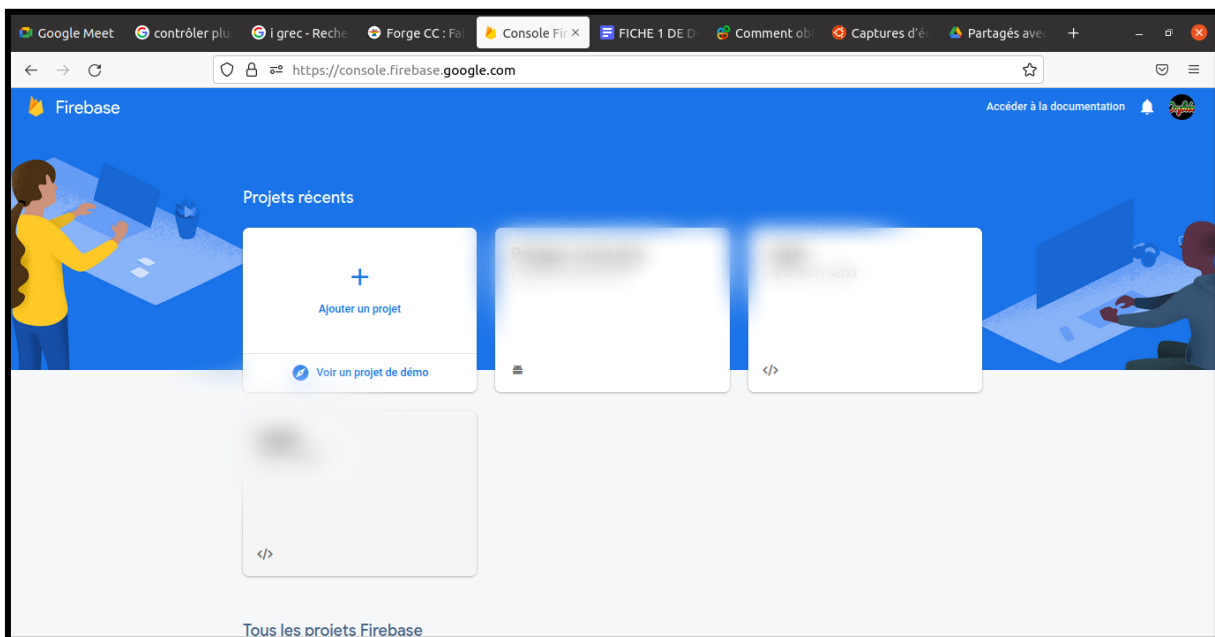
Les API sont un moyen de fournir des services à vos propres applications ou à des applications tierces. Firebase vous permet de fournir des services personnalisés qui, à leur tour, utilisent les propres services de Firebase sans avoir à configurer un backend pour ces services. Vous pourriez, par exemple, offrir l'accès à une base de données en temps réel Firebase pour les applications tierces afin d'interroger les informations collectées par les capteurs industriels.

La première étape de la création d'une API dans Firebase consiste à accéder à la plateforme en se rendant sur <https://firebase.google.com/> .

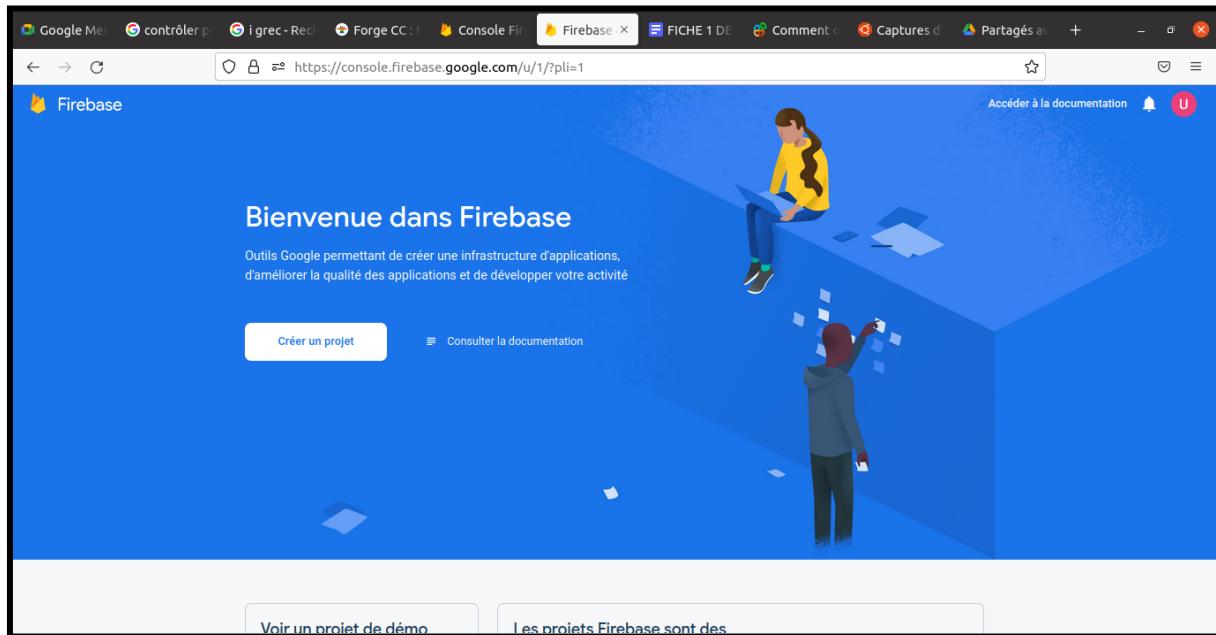


Une fois sur la plateforme, cliquez sur “Go to console” dans le coin supérieur droit.

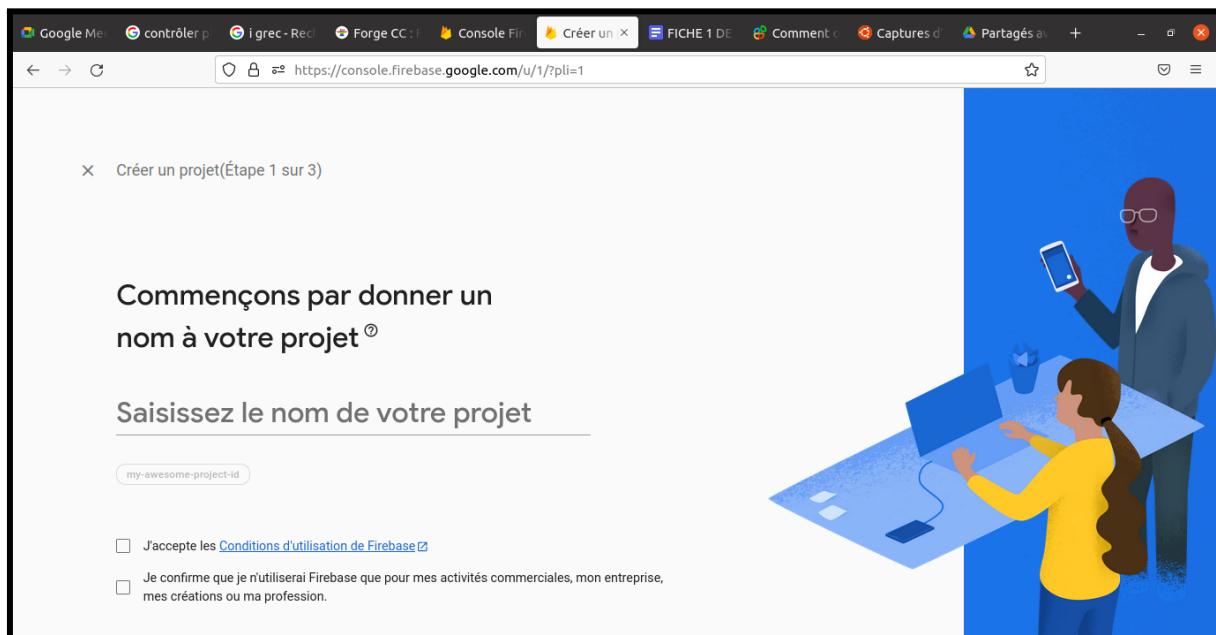
Si vous avez déjà des projets Firebase vous verrez un interface semblable à celui-ci:



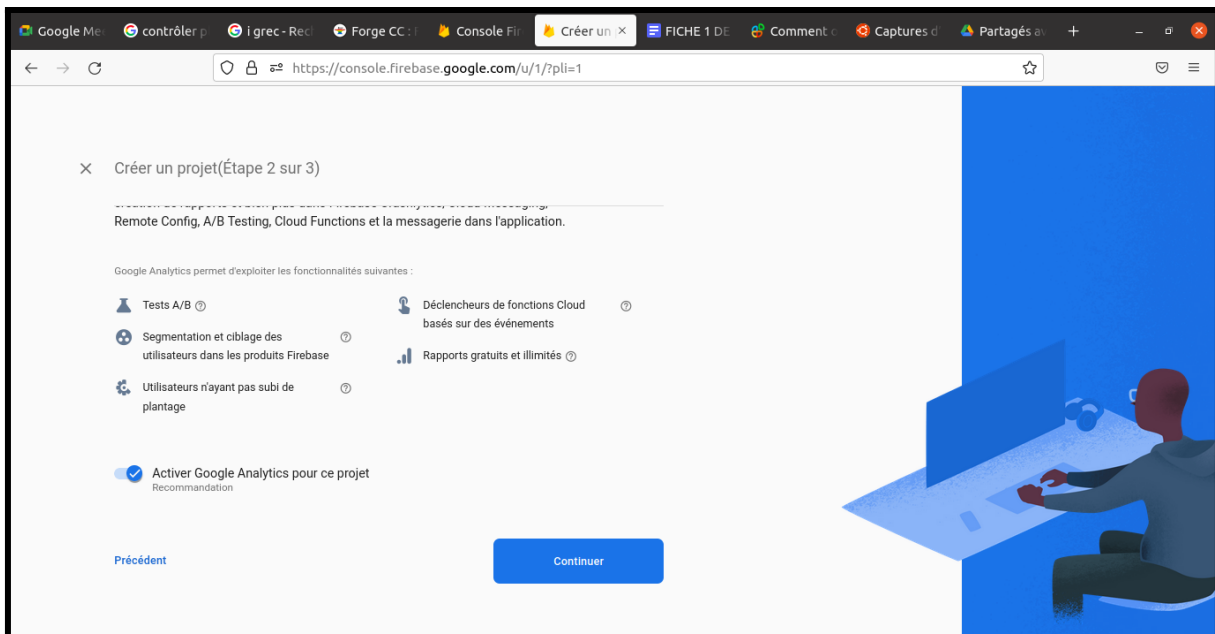
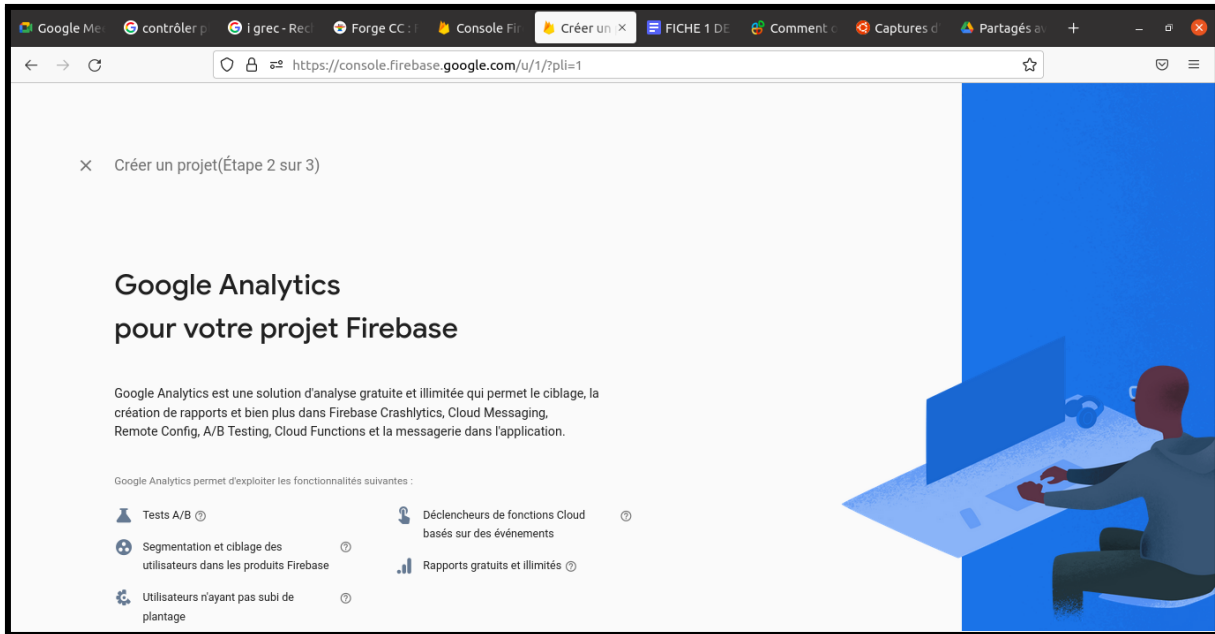
Si vous n’aviez encore aucun projet Firebase votre interface ressemblera plutôt à ceci:



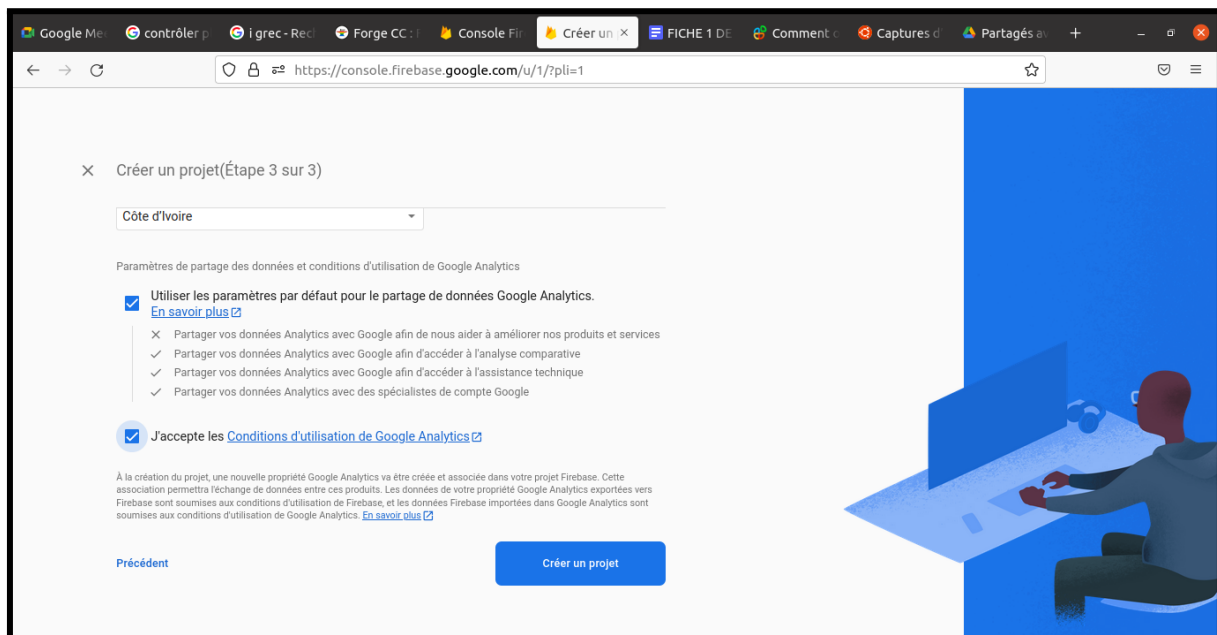
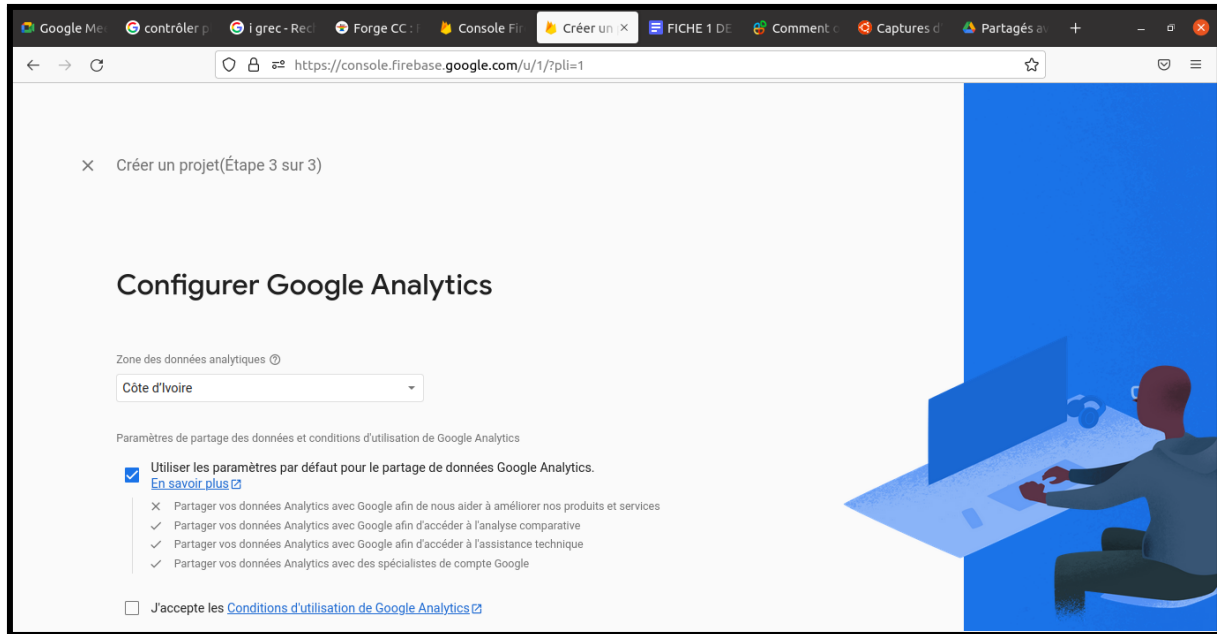
Cliquez sur “Ajouter un projet” ou “Créer un projet” selon le cas



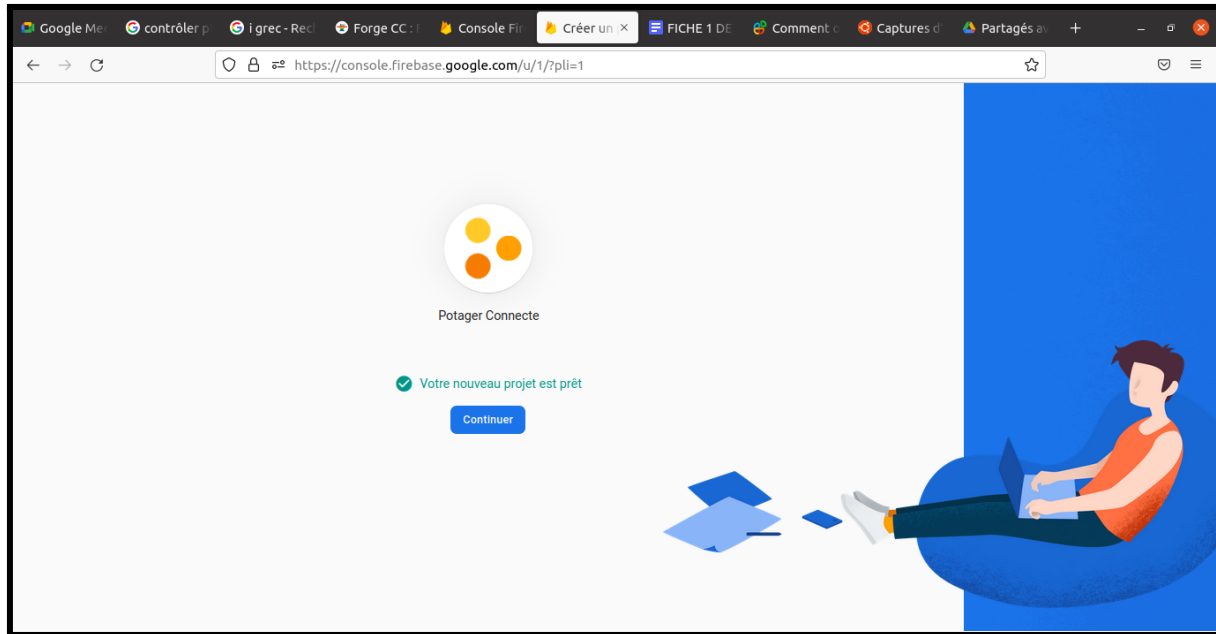
Renseignez le nom de votre projet, cochez les cases relatives à conditions et termes d’utilisation après lecture et cliquez ensuite sur “Continuer”



Sur la précédente interface vous pouvez activer ou désactiver Google Analytics (pas d’importance pour ce projet) avant de cliquer sur continuer.



A cette étape vous êtes appelés à choisir la zone de vos données. Il est préférable de choisir une zone proche de celle où sera utilisée la base de données.

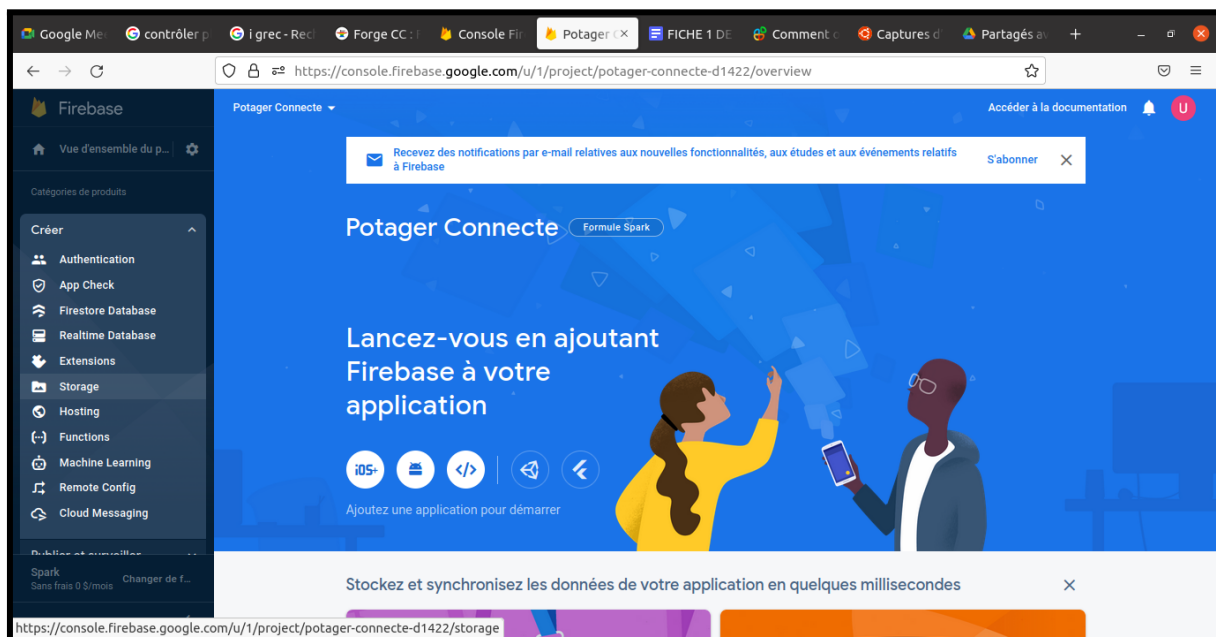


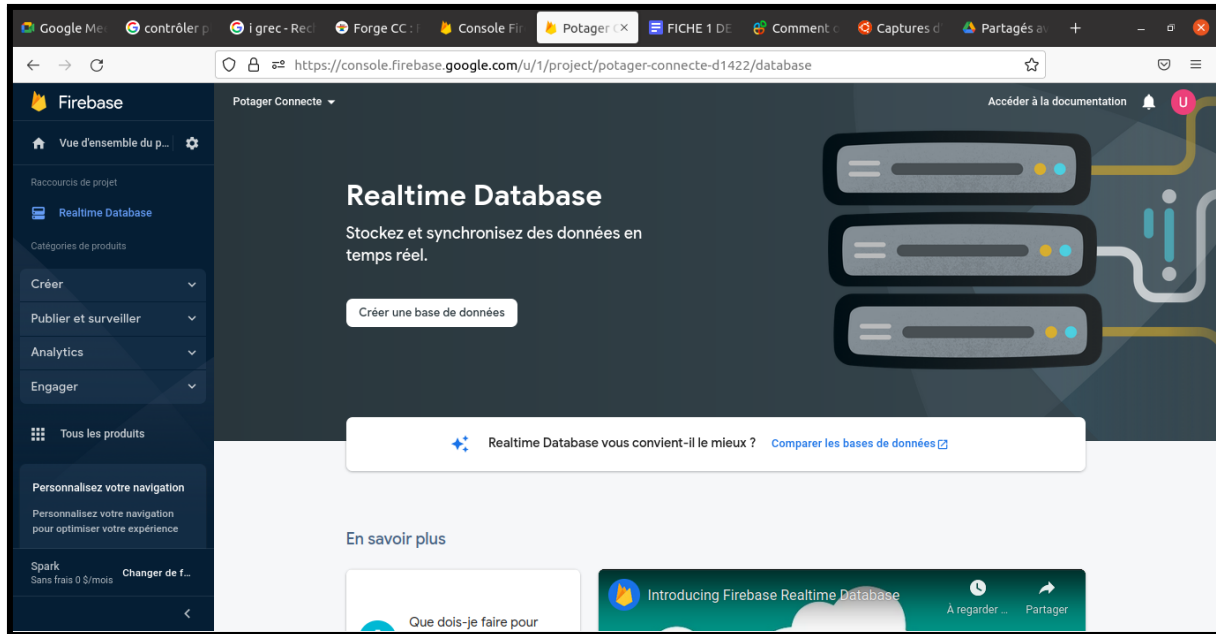
Félicitations, vous venez de créer une API dans Firebase.

La prochaine étape consiste à créer une base de données dans laquelle nous pourrions visualiser les valeurs de l'humidité.

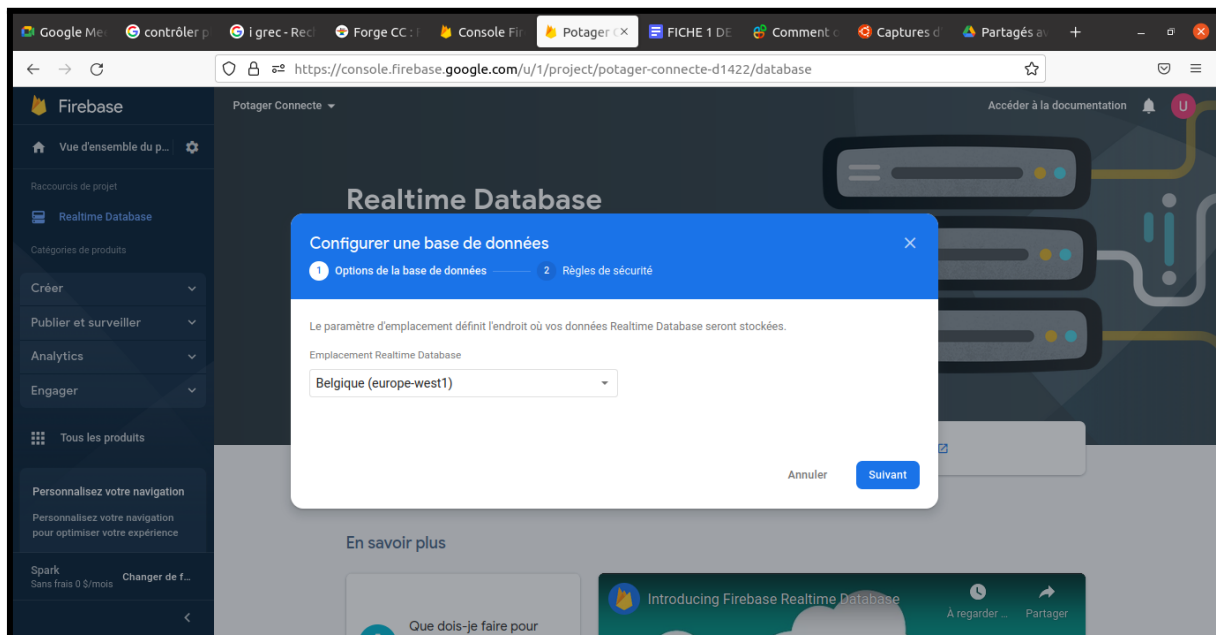
6.3. Configurer une base de données en temps réel dans Firebase

Dans la barre de navigation à gauche, dans la section “Catégorie de produits”, cliquez sur “Créer” puis sur “Base de données en temps réel” ou “Realtime Database”.

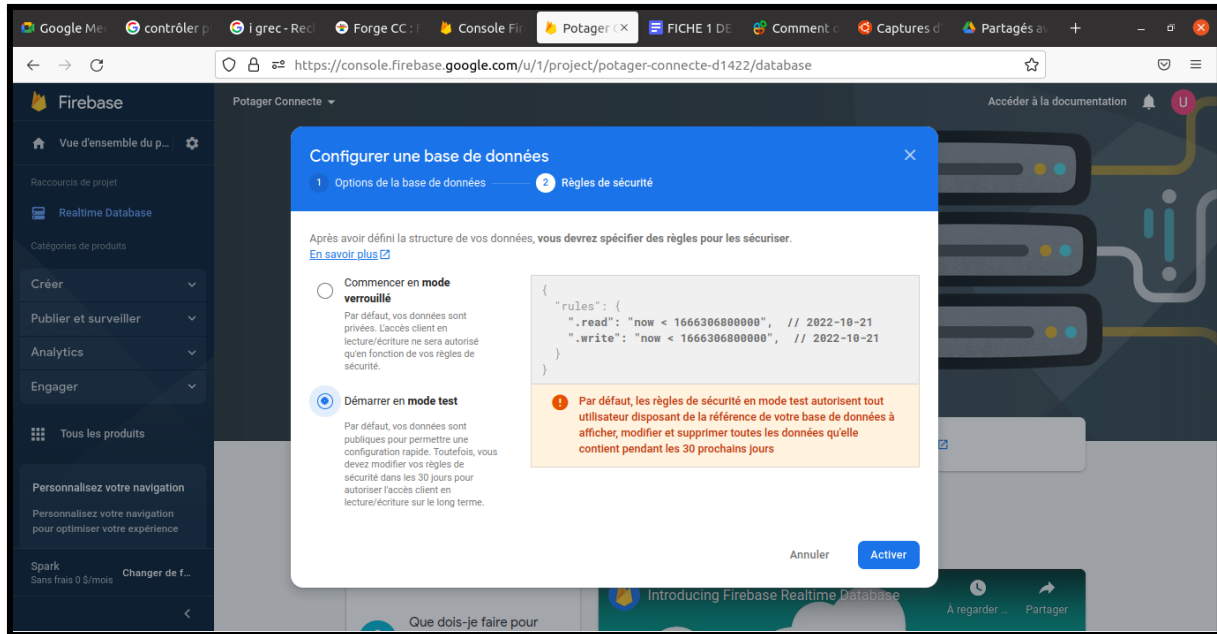




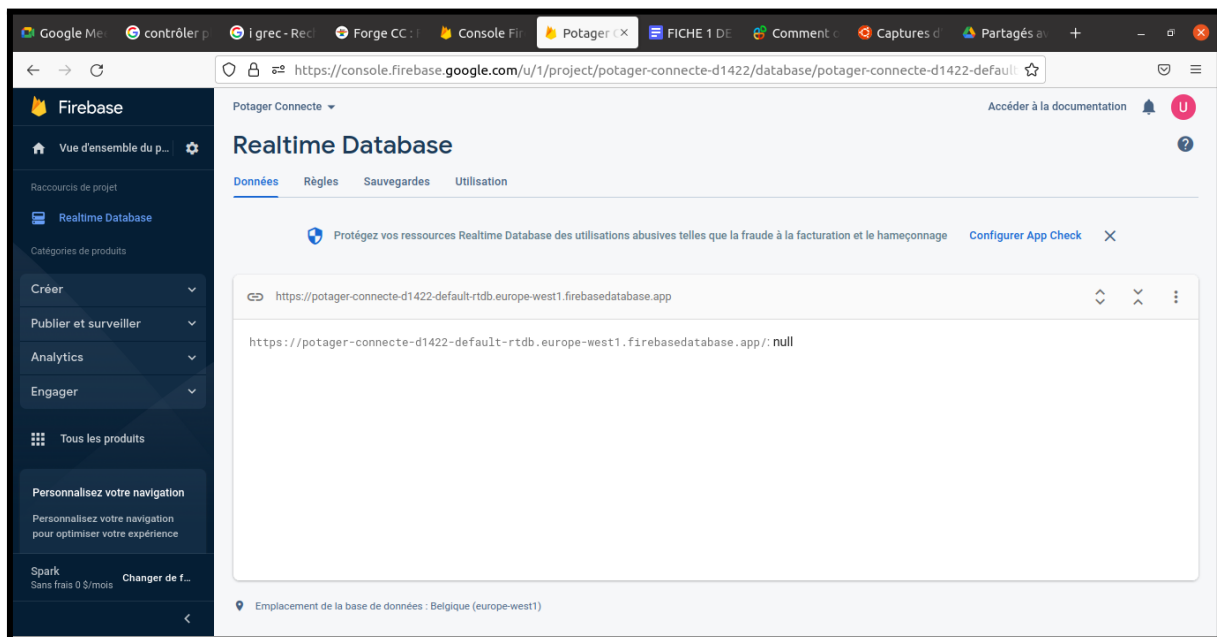
Un bouton « Créer une base de données » apparaît sur la droite. Cliquez dessus pour créer votre première base de données dans Firebase.



Ensuite, vous devrez choisir entre plusieurs options de localisation géographique pour votre nouvelle base de données. Sélectionnez celui qui est le plus proche de vos utilisateurs. C'est un aspect important pour minimiser la latence de votre API, en particulier dans les applications en temps réel.



La prochaine étape consistera à définir les règles de sécurité de la base de donnée. Pour l’heure, nous allons sélectionner “Démarrer en mode Test” qui laisse un libre accès à la base de données.



Nous venons d’achever la création de notre base de données. Il nous sera donc possible de l’utiliser ultérieurement.

Veillez bien à modifier l’url de la base de donnée dans le code à téléverser sur l’ESP8266. Pour

ce faire, copiez l’url en cliquant sur le bouton  présent sur l’image précédente.

Le lien une fois collé doit ressembler à ceci:

“https://potager-connecte-d0c74-default-rtdb.firebaseio.com/”,

Avant d’intégrer l’url au code, il faudra retirer le “https://” du début et le “/” de la fin pour obtenir un url comme celui-ci : “potager-connecte-d0c74-default-rtdb.firebaseio.com”.

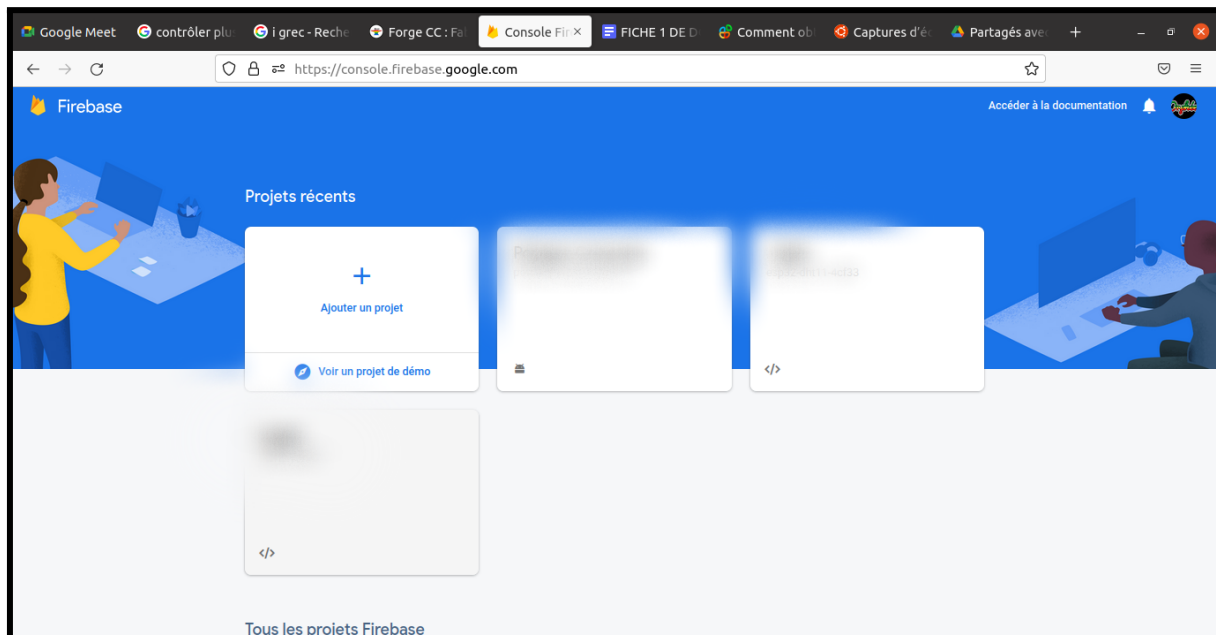
6.4 - Visualisation des données

Afin de visualiser les données envoyées par notre circuit électronique dans la base de données, il nous est possible d’accéder directement à notre base de données via un navigateur web.

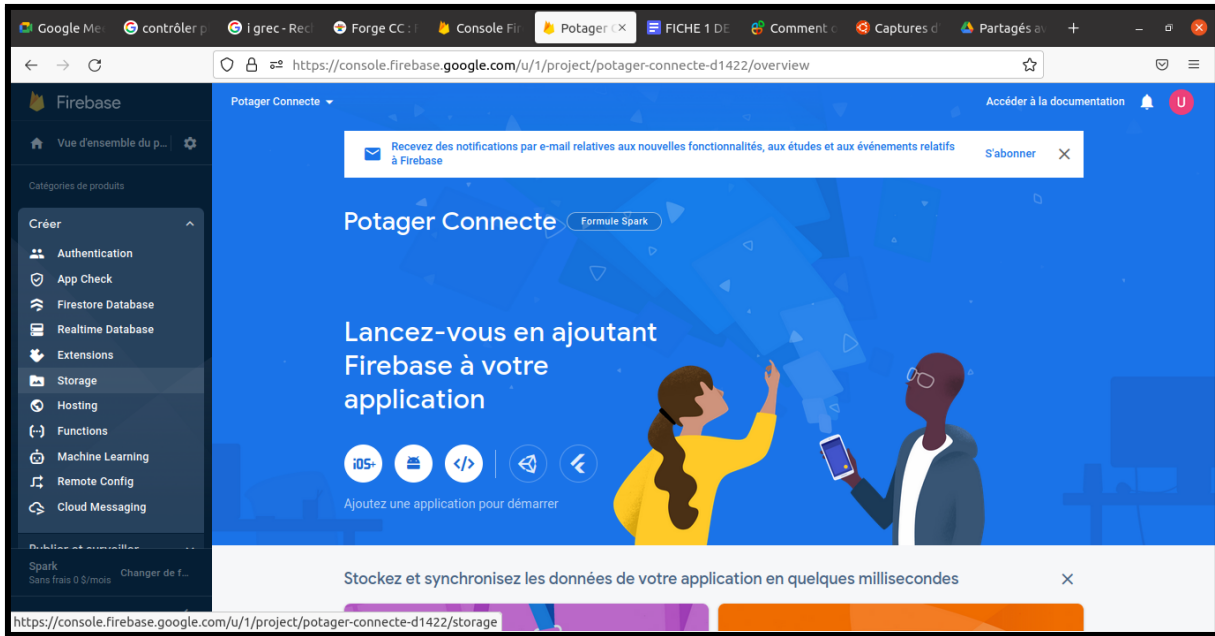
Pour ce faire, rendez-vous sur <https://firebase.google.com/>.

Assurez-vous d’être connecté avec le compte google utilisé pour créer la base de données.

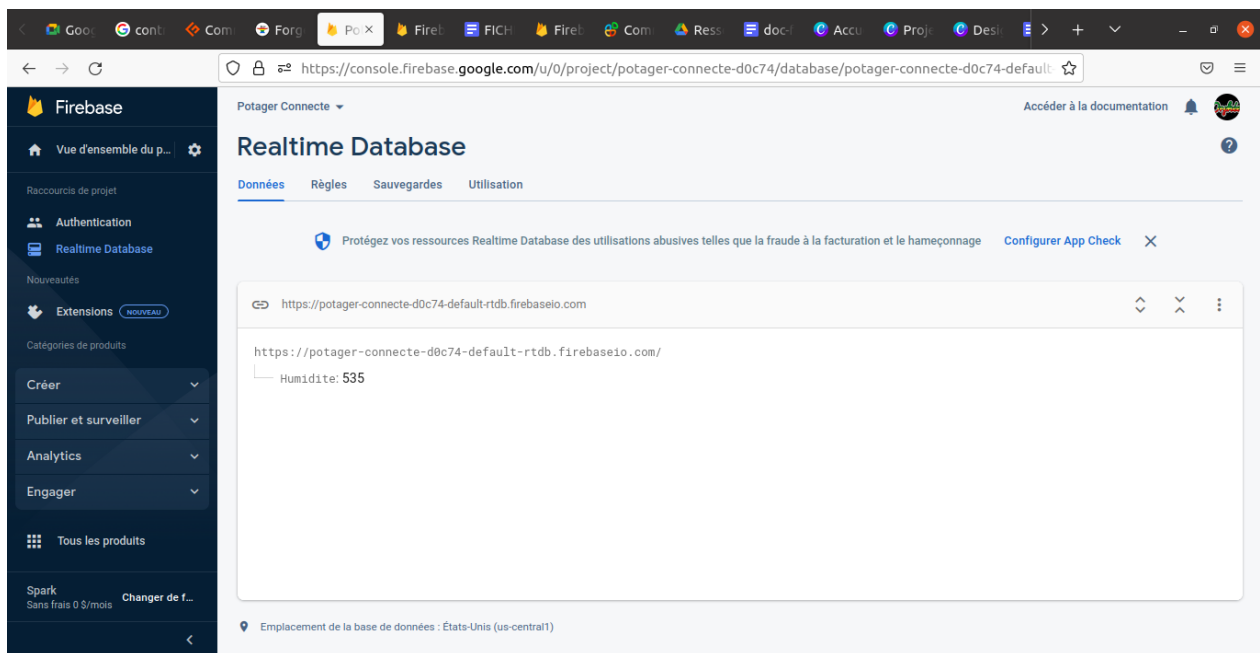
Suivez les étapes de la création de base de donnée jusqu’à aboutir à la l’interface suivante:



Dans la liste des projets présentés, cliquez sur celui que vous avez créé dans le chapitre précédent.



Une fois sur la page d’accueil de votre projet, cliquez sur “Realtime Database”. Vous verrez désormais les données envoyées par l’ESP8266 changer en temps réel sur un interface comme celui-ci:



L'application de démonstration quant à elle se présente comme suit:



Cette application est un démonstrateur expérimental développé avant l'atelier par M.DJIKPE Orphée Ulrich Mahoutondji - contact : djikpeo@gmail.com

7-Difficultés rencontrées lors de la réalisation du projet

Au terrain nous avons constaté que les étapes suivantes ont besoin d'un accompagnement car elles peuvent prendre du temps, et les erreurs en couter encore plus pour mener à bien l'atelier dans les délais :

- Difficultés lors de la modélisation du boîtier.
- Difficultés à souder certains composants au Veroboard
- Problèmes d'alimentation du moteur de la pompe

8-Réalisation du potager et tests de vérification

Afin de réaliser le potager auquel sera connecté le dispositif électronique, il faut suivre les étapes suivantes

- Se munir des plantes
- Se procurer le terreau
- Disposer de deux bidons de 25L d'huile(vides)
- Découper un bidon d'huile dans le sens de la longueur (en enlevant environ 5 cm de plastique)



- Découper le second bidon d'huile dans le sens de la largeur (en enlevant environ 5 cm de plastique)
- Nettoyer les bidons pour le débarrasser des traces d'huiles
- Introduire du terreau dans le bidon découpé dans le sens de la longueur jusqu'au $\frac{2}{3}$ du contenant
- Planter les pieds de citronnelle en veillant à respecter l'équidistance



- Verser de l'eau dans le bidon découpé dans le sens de la largeur



- Installez la pompe, le capteur d'humidité, et le boîtier avec son système d'accrochage.
- Mettez en marche et vérifiez que le système fournit bien les données en temps réel.

9-Perspectives

- Contrôler d'autres paramètres (température, luminosité...)
- Afficher le pourcentage d'humidité dans l'application
- Concevoir un système de remplissage d'eau automatique au niveau du réservoir
- Créer un système d'arrosage rotatif
- Réaliser le projet à une plus grande échelle

Ce projet a été réalisé sur financement du projet [Forge d’adaptations Nord-Sud](#) au fablab Blolab à Cotonou au Bénin.

[Une fiche de documentation pédagogique de conduite d’atelier a été réalisée par ici.](#)

CC by SA - AYEMONCHE Hyacinthe - DASSI Ornélia - DJIKPE Orphée - OKE Raoudath

Blolab - Cotonou - bénin

