

**Université Sorbonne Paris Nord**

**PROJET**

***SMART IRRIGATION SYSTEM***

**Équipe :**

- 1. Chahinez Khebizi**
- 2. Moussa Conde**
- 3. Syphax Ouaisa**
- 4. Aurélie Chorro**

**Encadré par :**

- 1. Massinisa Hamidi**
- 2. Aomar Osmani**

**ANNÉE : 2020/2021**

## **Remerciement**

Nous tenons tout d'abord à remercier les enseignants qui nous ont encadré  
Monsieur Aomar Osmani ainsi que Monsieur Massinisa Hamidi.

# Sommaire

**I - Introduction**

**II - Problematique**

**III - Objectif**

**IV - Organisation de projet**

**4.1 Répartition du travail**

**V - Etat de l'art**

**5.1 - Système d'irrigation intelligent basé sur RCSF**

**5.2 - Système Rain Bird**

**5.3 - Système smart garden Hub de Green IQ**

**5.4 - Système Bee green**

**VI - Diagramme**

**6.1 - Diagramme de séquence**

**VII - Composant électronique**

**7.1 - Introduction**

**7.2 - Les capteurs**

**7.3 - Schéma de branchement**

**VIII - Etude de marché**

**IX - Applications utilisées**

**9.1 - Virtuino**

**9.2 - ThingSpeak**

**X - Conclusion**

## I - Introduction

Actuellement, le monde compte plus de 7 milliards d'habitants. Leurs activités se multiplient de jour en jour et ont recours à de nombreuses ressources, dont certaines se font de plus en plus rares. Parmi elles, on trouve une ressource principale : l'eau.

Elle est utilisée dans tous les domaines, tels que la vie quotidienne, le secteur industriel, mais surtout le secteur agricole qui a recours à près de 70% du volume total du prélèvement de l'eau. Cependant, une étude récente a révélé qu'en moyenne 33% de la population souffre d'une pénurie d'eau, ce chiffre atteindra probablement les 50% en 2030. Face à ce constat alarmant, les agriculteurs cherchent à diminuer leur consommation d'eau en utilisant différentes méthodes qui peuvent se révéler efficaces mais coûteuses. L'émergence de l'IoT permet de faire un bon compromis en introduisant l'irrigation intelligente. Celle-ci permet de mettre en place des systèmes d'irrigation sophistiqués qui contrôleront les paramètres environnementaux (température, humidité de l'air et humidité du sol) afin de fournir la quantité d'eau adéquate quand cela est nécessaire et ainsi, limiter au maximum les pertes.

Dans le cadre de notre projet, nous souhaitons créer un système d'irrigation qui lancera automatiquement l'arrosage quand les paramètres environnementaux, en particulier l'humidité, atteignent un seuil fixé. Ce système sera contrôlé par une application appropriée et les différents paramètres seront stockés dans le Cloud.

Ce rapport comporte 3 chapitres :

- Le premier chapitre présentera une vue globale de l'irrigation et de ses différentes méthodes. Il traitera également l'émergence de l'IoT et ce qu'elle peut apporter à l'irrigation intelligente.

- Le deuxième chapitre présentera les composants électroniques utilisés tels que : la carte Wemos et les différents capteurs qui permettent de surveiller les paramètres environnementaux.

- La troisième partie portera sur le développement de l'application, le branchement des différents composants avec la carte Wemos qui sera la pièce maîtresse. Elle présentera également le code Arduino et la connexion au Cloud. Par la même occasion, nous présenterons les différents outils logiciels utilisés. Une conclusion clôturera cet exposé.



Image d'un système d'irrigation.

## **II - Problématique**

Comment exploiter l'émergence de l'internet des objets au profit de l'irrigation intelligente afin de contribuer à l'économie de l'eau et à sa préservation ?

## **III - Objectif**

- Concevoir un système d'irrigation à l'aide d'une carte Arduino Wemos reliée à des capteurs spécifique
- Contrôler ce système à l'aide d'une application et envoyer les mesures des capteurs au Cloud
- Automatiser le système en lançant la pompe si le sol n'est pas assez humide.

## **IV - Organisation du projet**

### **1. Répartition du travail**

Les répartitions des tâches ont été établies de la façon suivante lors de notre projet :

#### **1ere partie du projet : Organisation du projet**

- Achat des composants : Syphax
- Rédaction du 1er rapport et de la 1ere présentation : Toute l'équipe

#### **2eme partie du projet : Mise en place**

- Diagramme de séquence et de cas d'utilisation : Chahinez et Aurélie
- Code arduino : Tout l'équipe
- Branchement : Moussa
- Ajout bibliothèque : Moussa
- Application mobile Virtuino : Chahinez

#### **3eme partie du projet : Présentation finale**

- Rapport final : Toute l'équipe
- Vidéo : Tout l'équipe

## **V - Etat de l'art**

En effectuant nos recherches sur les différents systèmes d'irrigation déjà existant, nous avons pu constater que le marché dans ce domaine est très enrichi et divers. Les divers systèmes que nous avons pu étudier répondaient aux besoins de leur utilisateur mais de manières différentes (irrigation par ruissellement, irrigation par planches, irrigation à la haie, irrigation par bassins, irrigation goutte à goutte). Cependant, là où notre système se démarque c'est via son automatisation va répondre uniquement au besoin de la plante, son aspect écologique de l'eau.

Dans le souci d'amélioration des techniques d'irrigation, ainsi que le besoin de réduire la consommation en eau, plusieurs entreprises se sont lancées dans des systèmes d'irrigation intelligents tels que :

### **5.1 - Système d'irrigation intelligent basé sur RCSF**

#### Inconvénients du système :

- une simple application avec une simple interface (affichage de l'état de la zone sèche, irriguée, saturée, affichage des données d'humidité, affichage de la courbe). -le déclenchement de l'irrigation se fait d'une façon manuelle par l'application java en fonction de l'état de la zone cible pour l'irrigation.
- utilisation d'un seul type de capteur ( capteur d'humidité) qu'il n'est pas suffisant pour l'économie de l'eau d'irrigation.
- Le stockage de données se fait à travers une BDD local.
- L'administrateur est le seul acteur principal utilisant ce système.
- Ce système n'est pas assez intelligent. Il n'existe pas une application mobile pour l'accès et le contrôle à distance (via un système sans fil ou via le réseau internet) du système d'irrigation.
- Pas d'utilisation de réseau internet dans L'échange de données entre l'objet connecté et l'application java.

### **5.2 - Système Rain Bird**

#### Inconvénients du système :

- Ce système ne prend pas en compte les facteurs contribuant à l'économie de l'eau (Humidité du sol, température de l'air, l'évaporation, etc.).
- Ce système n'est pas assez intelligent, le lancement ou l'arrêt de l'arrosage se fait par une simple programmation du minuteur (les jours, les horaires, les durées d'arrosage).



### **5.3 - Système smart garden Hub de Green IQ**

#### Inconvénients du système :

Ce système adopté a plusieurs facteurs contribuant à l'économie de l'eau , mais comme mentionné précédemment, il y en a d'autres facteurs qui sont aussi importants à la contribution pour la conservation de l'eau nous citons par exemples : L'intensité du vent, l'humidité de l'air, les saisons de l'année, le détecteur de la pluie , etc.

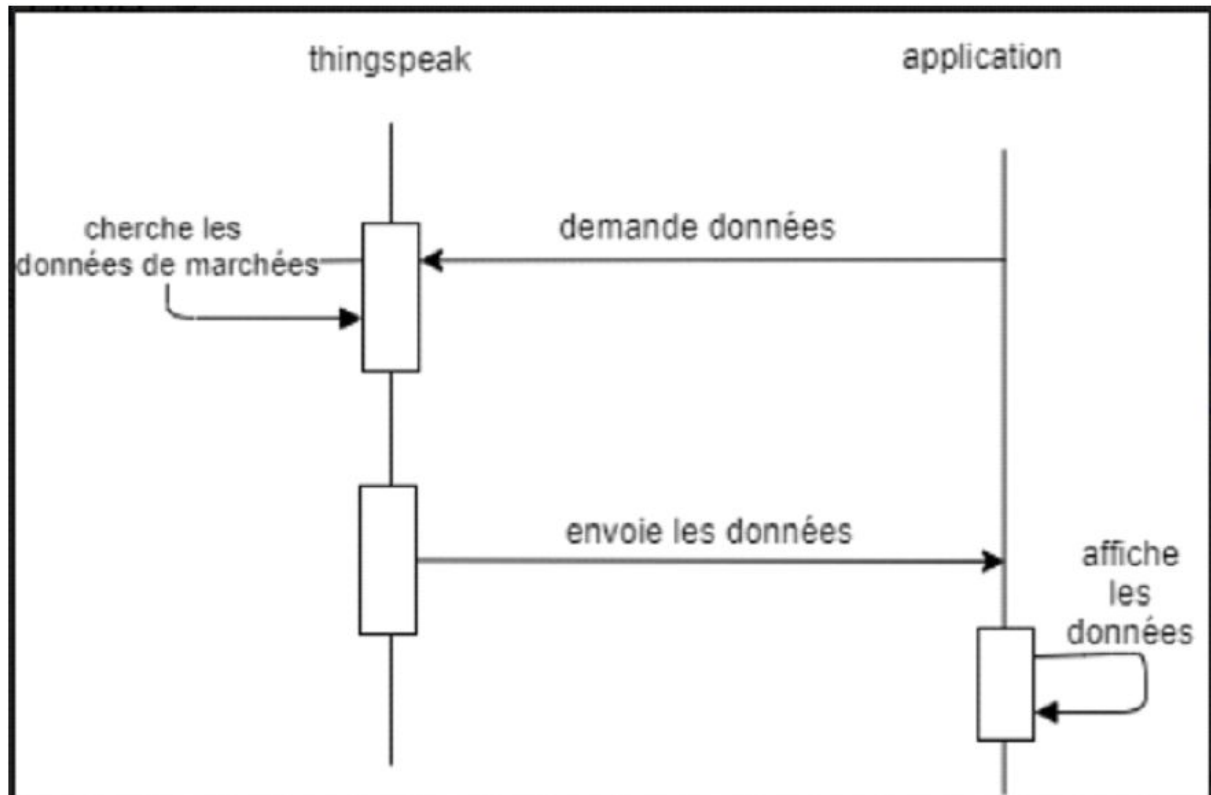
- Ce système ne profite pas des services fournis par les plateformes d'IoT pour analyser, traiter ou stocker des données.
- Ce système est compatible uniquement avec des types spécifiques de capteurs (le Flower Power de Parrot - le Koubachi - Netatmo).
- Ces types de capteurs sont un peu coûteux physiquement, en particulier quand il s'agit de grandes zones d'arrosage et l'utilisation de centaines d'eux. (Le Flower Power son prix est de 69.95 € et le Koubachi pour un prix de 119.95 €).

### **5.4 - Système Bee green**

Spécialiser dans la cartographie, la surveillance et l'exploitation à distance des systèmes d'irrigation, mais aussi dans la programmation avancée en irrigation (basée sur le temps, basée sur les conditions météorologiques ou les capteurs) . Le projet smart irrigation compte apporter un programme qui répond plutôt aux besoins de la plante sans que l'utilisateur n'ait à rentrer quoi que ce soit. C'est-à-dire à l'aide des capteurs d'humidité reliés au programme, nous saurons précisément à quel moment la plante doit être arrosée. De plus à l'aide d'un smartphone, il y aurait la possibilité de demander au programme d'arroser la plante à distance sans qu'on ait à se déplacer.

## VI - Diagramme

### Diagramme de séquence



## VII - Les composants électroniques

### 7.1 - Introduction

Afin de mettre en place notre système d'irrigation intelligent, nous aurons recours à des dispositifs électroniques. Nous utiliserons différents capteurs d'humidité, de température et d'eau afin de nous indiquer quand est-ce qu'il faut irriguer nos plantes. Pour lancer le système automatiquement nous utiliserons une carte électronique pouvant être connectée à un réseau Wifi. Nous détaillerons les caractéristiques de chaque composant dans ce chapitre.

#### Aperçu sur l'Arduino

Arduino, (et son synonyme Genuino) est une marque qui conçoit des cartes électroniques équipées d'un microcontrôleur. Celui-ci peut être programmé pour analyser et produire des signaux électriques afin de pouvoir effectuer différentes tâches de domotique, pilotage d'un robot ou même d'informatique embarquée.

Ces cartes sont compatibles avec différents capteurs et d'autres dispositifs électroniques, ce qui permet d'élargir leurs domaines d'utilisation. Il existe différentes cartes Arduino commercialisées : l'Arduino Uno qui est la plus utilisée, L'Arduino méga, Serial Arduino qui a été la première version à voir le jour, Arduino Nano, Arduino Ethernet ...



Pour programmer l'une de ces cartes il faut utiliser l'ide Arduino qui est une application Java, libre et multiplateforme, servant d'éditeur de code et de compilateur, et qui peut transférer le firmware et le programme au travers de la liaison série (téléversement du code).

Le langage de programmation est le C++, et il existe de nombreuses bibliothèques qui offrent diverses options et multiplient les fonctions de ces cartes. (Source <https://fr.wikipedia.org/wiki/Arduino>)

## Aperçu sur le module WiFi ESP8266

L'émergence et le développement de l'Internet fait qu'il est aujourd'hui présent et indispensable dans tous les domaines. Afin d'automatiser notre système d'irrigation, nous aurons recours au Cloud et les diverses fonctionnalités qu'il propose, il est donc nécessaire d'utiliser un module WiFi.

L'ESP8266 est un circuit intégré à microcontrôleur avec connexion Wifi développé par le fabricant chinois Espressif.

Au lieu d'utiliser une carte Arduino et un module Wifi Esp8266 nous avons opté pour une carte Arduino WeMos D1 R2 qui est l'association d'une carte Arduino Uno et de l'ESP8266EX. Ce module est donc intégré dans la carte. (Source <https://fr.wikipedia.org/wiki/esp8266>)

## La carte WeMos D1 R2

Il s'agit simplement d'une puce ESP8266EX intégrée dans une carte au format de l'Arduino Uno. Il dispose de onze entrées/sorties numériques, d'une entrée analogique et fonctionne sous 3,3 V. Son utilisation est beaucoup plus pratique, car il intègre un contrôleur USB et un régulateur de tension qui permet l'utilisation (comme avec l'Arduino) de piles, de batteries rechargeables.



## Caractéristiques

Microcontrôleur	ESP8266EX
Tension de fonctionnement	3.3 volts
Pins d'entrées/sorties digitaux	11 (D0-D10)
Pin d'entrée analogue	1 (A0)
Vitesse d'horloge	80 MHz/160 MHz
Longueur	68.8 mm
Largeur	53.4 mm
Poids	25 g
Port USB	Micro USB
Puissance maximale	5V via le port USB ou le Pin 5V sur la carte

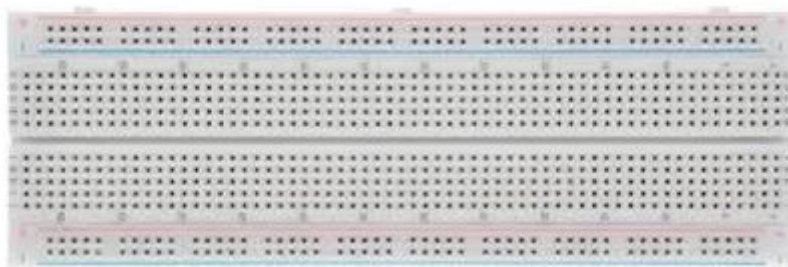
## Pins

Pin	Fonction
TX	TXD
RX	RXD
A0	Entrée analogue, entrée max 3.3 volts
D0-D10	Entrée sortie digitales
GND	Ground (masse)
5v	Fournit 5v
3V3	Fournit 3.3v
RST	Reset

## La BreadBoard

C'est est une plaquette ou plaque d'essai sans soudures. Son principal avantage est de permettre de tester et réaliser des montages rapidement des circuits électroniques sans souder aucun composant ce qui permet de les réutiliser pour d'autres montages.

Il est très utile de l'utiliser avec une *carte Arduino*. Par exemple, si plusieurs composants nécessitent une alimentation de 5V la *BreadBoard* est nécessaire car la carte ne possède qu'un *pin* 5V.



## 7.2 - Les capteurs

**Définition d'un capteur** : c'est un dispositif qui permet de convertir une grandeur physique observée (température, luminosité, humidité, débit, présence d'objet,...) en une grandeur normée et utilisable, généralement un signal électrique (courant, tension, niveaux logiques, valeur moyenne, fréquence, amplitude, nombre binaire,...), qui peut être interprété par un *dispositif de contrôle commande*. Les *capteurs* jouent un rôle indispensable dans le domaine d'*IoT*.

### Présentation des capteurs utilisés

Capteur d'humidité au sol YL69 : un capteur d'humidité simple qui peut être utilisé pour détecter lorsqu'un sol est en déficit d'eau. Il permet aussi de retourner le taux d'humidité au sol. Il comporte les connexions VCC (3V-5V), GND (masse), DO (digital output interface), AO (digital output Interface).



*Capteur* (détecteur) *d'eau* qui renvoie HIGH si de l'eau est détectée LOW sinon. Voici certaines de ces caractéristiques :

- Tension de fonctionnement : 3-5 V ;
- Courant de fonctionnement : moins de 20 MA ;
- Température de fonctionnement : 10°C à 30°C ;
- Humidité de fonctionnement : 10% à 90% ;
- Type de *capteur* : analogique ;
- Connexions : VCC, GND, Signa.

Pour brancher le capteur avec la carte WEMOS, il suffit de relier le (+) au 5 V, le (-) au GND et le Signa à un pin digital ou analogique. Comme la WEMOS ne contient qu'un pin analogique et qu'il est utilisé par le capteur d'humidité au sol, nous le relierons à un pin digital.

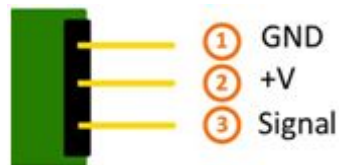


DHT11 : capteur d'humidité et de température d'une grande fiabilité et de stabilité à long terme et ayant une réponse rapide.

Pour son branchement, il faut relier le (+) au 3.3 V ou 5 V, le (-) GND et choisir un pin digital.



Voici une photo détaillant le montage :



### Le relai :

C'est un composant électronique qui sert d'interrupteur. On lui définit un pin qui le relie à la carte, si le *pin* est au niveau HIGH; l'interrupteur est fermé. Sinon, il est au niveau LOW; l'interrupteur sera ouvert.



## La pompe à eau :

Elle permet de remplir le réservoir avec de l'eau lorsque le niveau d'eau est faible. La pompe est mise en marche lorsque le niveau d'eau est bas. Elle s'arrête lorsque le niveau d'eau arrive au niveau haut.



### 7.3 - Schéma de branchement

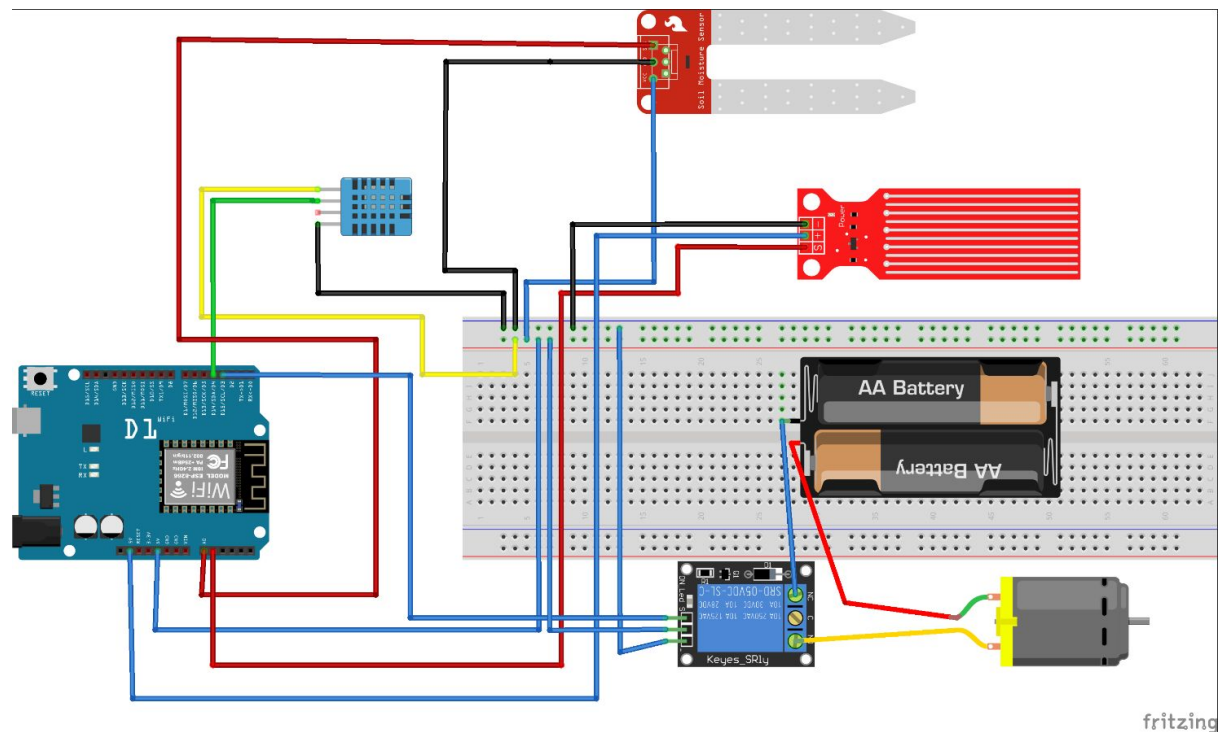


Schéma de branchement fait le sur logiciel Fritzing



## VIII - Etude de marché

Prix en moyenne :

Le **prix** d'un arrosage automatique enterré se situe en moyenne entre 10 et 15 € du mètre carré (pose comprise). Le **coût** est dégressif avec la superficie de votre terrain.

Entreprise sur le marché :

- Aquacontrol

Aquacontrol distribue depuis 1983 une large gamme d'équipements pour le dosage, la régulation et la filtration des fluides destinée aux professionnels du traitement des eaux, de l'environnement, de l'industrie ou de la piscine.

<https://www.amazon.fr/AQUA-CONTROL-Programmateur-darrosage-Mod%C3%A8le/dp/B07PDS3HB4>

Environ 20€

- Netafim

Netafim est une entreprise spécialisée dans le secteur de l'irrigation de l'eau pour l'agriculture.

<https://www.mon-irrigation.com/49-irrigation-aspersion>

- Aqua'Tec

Aqua'tec est spécialisé dans la fabrication et la distribution de matériel hydraulique. Nous travaillons dans le secteur de l'irrigation, du traitement de l'eau potable et des procédés industriels.

Sur Amazon ils en proposent de 10€ à 75€94

C'est aussi vendu dans les magasins grande surface comme Leroy Merlin entre 15€ jusqu'à 300€

Il faut noter que les plus bas prix sont les systèmes à piles alors que les programmeurs électriques sont plus coûteux (De 0.90cts à + de 200 euros de différence)

Prenons pour exemple un système d'irrigation automatique qui s'apparente à ce que l'on recherche :

[Retour aux résultats](#)



Passez la souris sur l'image pour zoomer



## Système d'irrigation automatique pour plantes d'intérieur

Marque : Royal Gardineer



255 évaluations | [5 questions avec réponses](#)

Prix : **26,95 €** + 5,95 € Livraison

Tous les prix incluent la TVA.

Neufs (2) à partir de **26,95 €** + 5,95 € de livraison

- Fréquence et durée programmables
- Pour 1 à 10 plantes / pots
- Pompe à amorçage automatique
- Pour tous types de seaux, bombonnes...
- Système d'irrigation automatique pour jusqu'à 10 plantes d'intérieur, par Royal Gardineer

[Signaler des informations incorrectes sur les produits](#)

### Description du produit

Fréquence de l'arrosage : de 10 fois par jour à une fois tous les 30 jours, durée : de 1 à 99 seconde(s)

Réglage de la quantité d'eau pour chaque plante par la longueur du tuyau, tuyau en vinyle 10 mètres sécable

Affichage LCD : affiche la fréquence et la durée de l'arrosage, rappel lorsqu'un changement de piles est nécessaire et lorsque l'eau manque

Pompe à eau à amorçage automatique

Support universel pour fixation sur un seau d'eau

Tiges en acier inoxydable pour fixation du tuyau dans le pot de fleurs

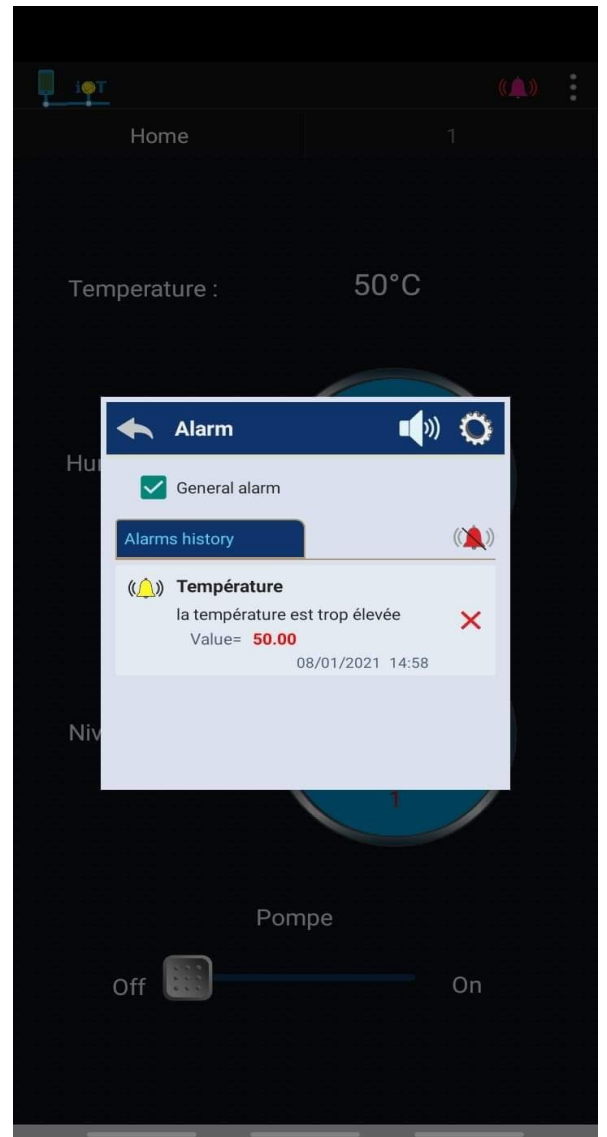
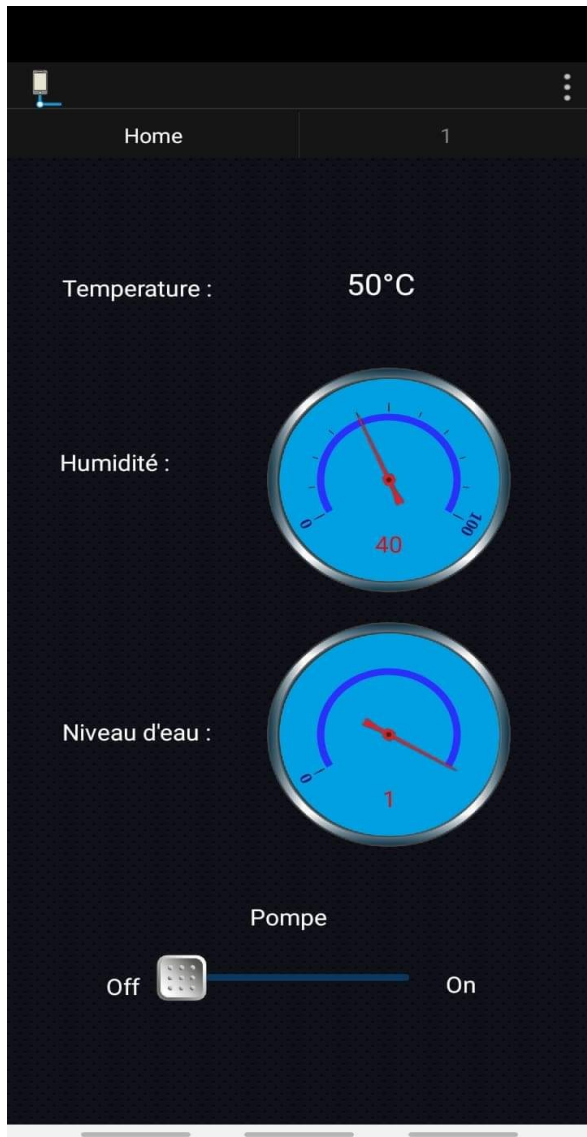
Alimentation : 4 piles AA (non fournies)

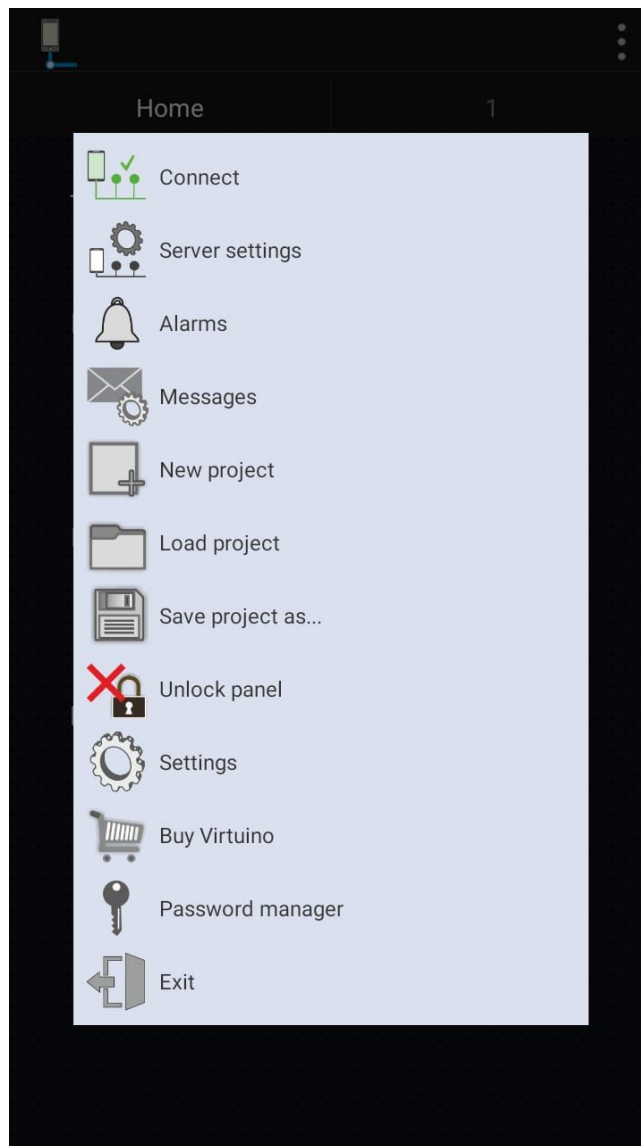
Dimensions avec support : 195 x 65 x 75 mm, longueur du tuyau jusqu'à la pompe : 32 cm, poids : env. 250 g

Système d'arrosage avec pompe, tuyau en vinyle et 10 tiges en acier inoxydable

## IX - Applications utilisées

### 9.1 - Virtuino





Menu de l'application

## **2 - ThingSpeak**

### **Introduction**

Dans cette partie, on s'intéresse à la conception du système d'irrigation. Nous présenterons d'abord le système d'irrigation, son mode de fonctionnement et son interaction avec le Cloud. Nous verrons en détails le code Arduino utilisé ainsi que la construction de l'application qui nous permettra de contrôler le système.

### **Cloud et ThingSpeak**

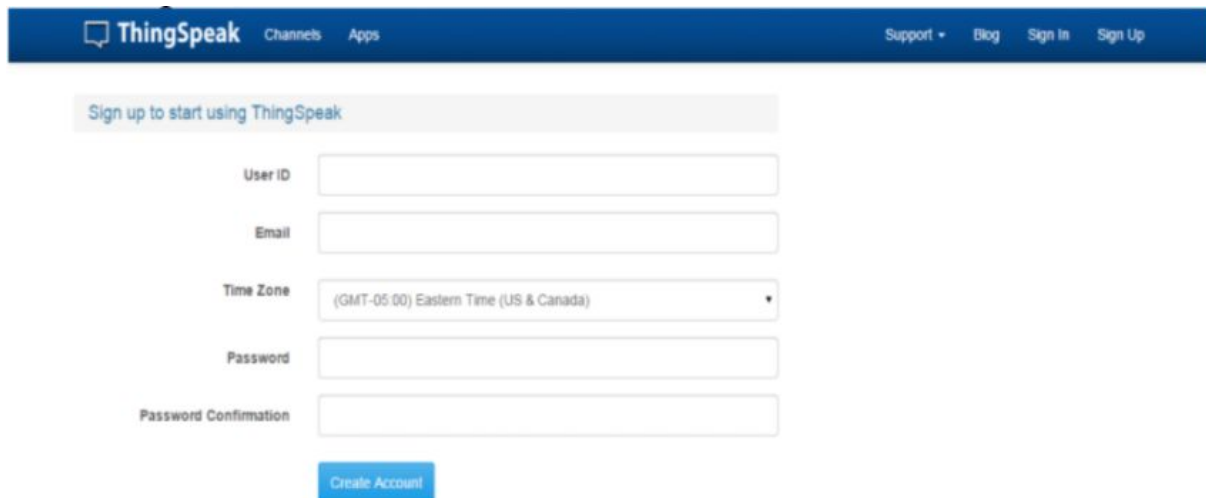
#### **Définition du Cloud**

Le cloud computing, en français l'informatique en nuage consiste à exploiter la puissance de calcul ou de stockage de serveurs informatiques distants par l'intermédiaire d'un réseau, généralement Internet. Le cloud computing se caractérise par sa grande souplesse : selon le niveau de compétence de l'utilisateur client, il est possible de gérer soi-même son serveur. Pour notre Système d'irrigation nous avons privilégié le serveur ThingSpeak qui est facile d'utilisation et qui correspond aux besoins de notre projet. (Source <https://fr.wikipedia.org/wiki/Cloud>)

### **ThingSpeak**

#### **Définition**

ThingSpeak est une application open source pour l'« Internet des objets », permettant de stocker et de collecter les données des objets connectés en passant par le protocole HTTP (que nous verrons plus loin dans ce chapitre) via Internet ou un réseau local. Avec ThingSpeak, l'utilisateur peut créer des applications d'enregistrement de données capteurs, des applications de suivi d'emplacements et un réseau social pour objets connectés avec les mises à jour de l'état. (Source <https://fr.wikipedia.org/wiki/ThingSpeak>)



The image shows the 'Sign up to start using ThingSpeak' form. It includes input fields for 'User ID', 'Email', 'Password', and 'Password Confirmation'. A 'Time Zone' dropdown menu is set to '(GMT-05:00) Eastern Time (US & Canada)'. A blue 'Create Account' button is at the bottom.

Capture du formulaire de création d'un compte ThingSpeak

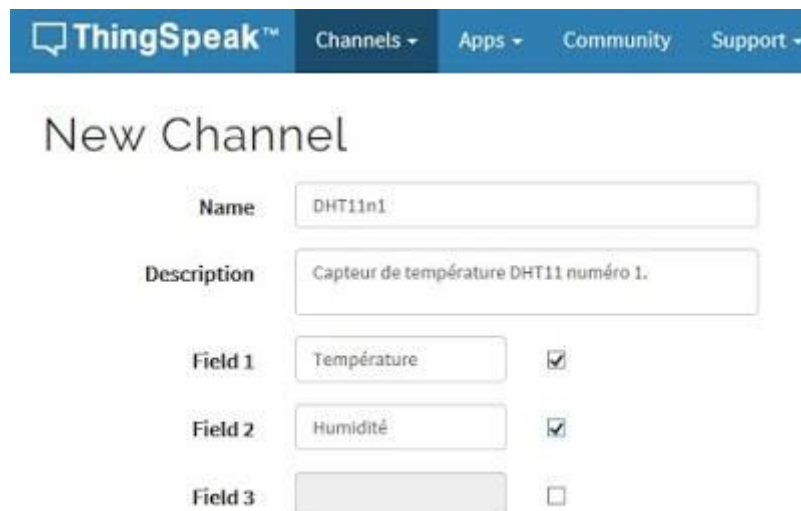
Une fois connecté, on crée ensuite un canal (channel). Cela représente un ensemble de données regroupées entre elles et qui proviennent d'un objet connecté, d'un autre canal ou même d'un service web. Chaque channel contient des champs (fields en anglais). Chaque field représente une donnée. On peut créer huit champs au maximum.

Dans notre *channel* nous avons créé des *fields* selon les données de nos capteurs. Soit, 5 fields :

- Field 1 : Pompe à eau (permettra de contrôler le relai qui lui-même contrôlé la pompe à eau),
- Field 2 : Température de l'air (capteur DHT11),
- Field 3 : Humidité de l'air (capteur DHT11),
- Field 4 : Humidité du sol (capteur YL69),
- Field 5 : Détecteur d'eau.

Pour créer notre canal et ses champs il suffit de suivre les étapes suivantes :

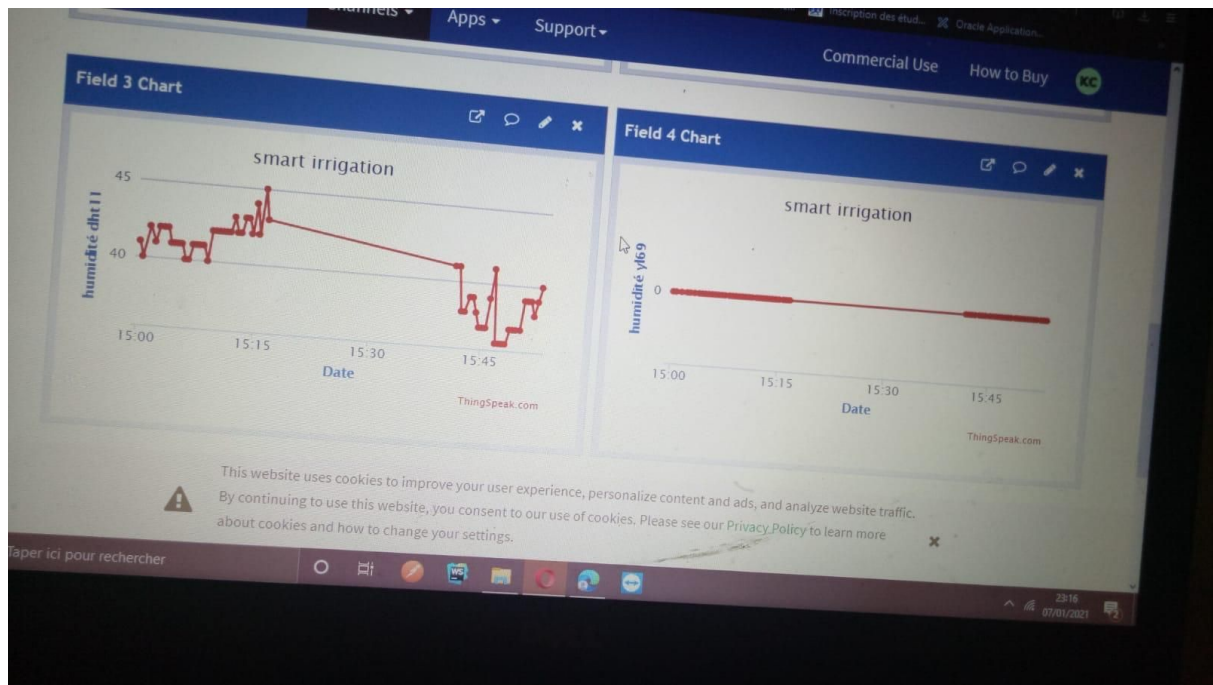
1. Cliquer sur "new Channel" et saisir les informations du canal ;



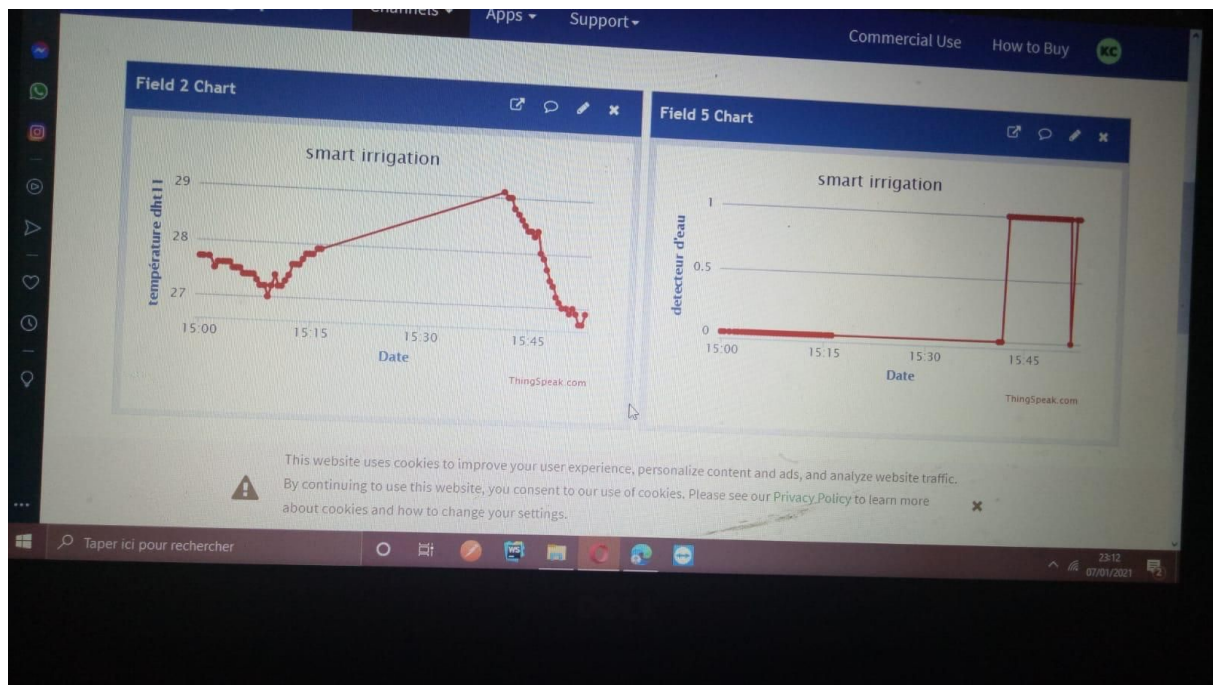
The image shows the 'New Channel' form. It has a 'Name' field with 'DHT11n1' and a 'Description' field with 'Capteur de température DHT11 numéro 1.'. Below are three 'Field' rows. 'Field 1' has 'Température' and a checked checkbox. 'Field 2' has 'Humidité' and a checked checkbox. 'Field 3' has an empty field and an unchecked checkbox.

## 2. Définir les fields.

Percentage Complete	15%
Channel ID	23207
Name	<input type="text" value="Channel 23207"/>
Description	<div><div></div></div>
Metadata	<div><div></div></div>
Tags	<div><div></div></div>
Latitude	<input type="text"/>
Longitude	<input type="text"/>
Elevation	<input type="text"/>
Make Public?	<input type="checkbox"/>
URL	<input type="text"/>
Video ID	<input type="text"/> <input checked="" type="radio"/> YouTube <input type="radio"/> Vimeo
Field 1	<input type="text" value="Field Label 1"/> <input type="button" value="remove field"/>
Field 2	<input type="text"/> <input type="button" value="add field"/>
Field 3	<input type="text"/> <input type="button" value="add field"/>
Field 4	<input type="text"/> <input type="button" value="add field"/>
Field 5	<input type="text"/> <input type="button" value="add field"/>
Field 6	<input type="text"/> <input type="button" value="add field"/>
Field 7	<input type="text"/> <input type="button" value="add field"/>
Field 8	<input type="text"/> <input type="button" value="add field"/>
<input type="button" value="Save Channel"/>	



Le graphe de gauche montre l'humidité détectée par le dht11 et à droite l'humidité détectée par le YL-69



A gauche le graphe de la température détectée par le DHT11 et à droite le graphe du Water Level Sensor 0 quand il n'y a pas d'eau 1 sinon.



## Menu de gestion du canal

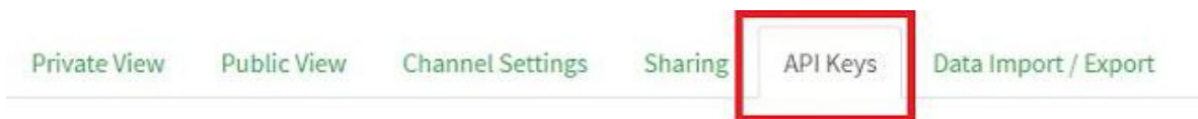
Pour gérer notre canal il faut connaître les notions suivantes :

Les vues (view) : on distingue deux vues : la vue privée (private) qui est uniquement accessible à l'utilisateur connecté au compte où a été créé le canal. Quant à la vue publique , elle est accessible à toute personne visitant le canal. On peut avoir des informations différentes sur les deux vues et on a le choix de désactiver la vue publique.



Ce menu permet de revoir les informations données à la création du canal et même de supprimer toutes les données ou tout le channel. Il permet de définir les options de confidentialité et de partage du canal. ThingSpeak permet de contrôler qui peut voir les données.

Les clés API (API Keys) : Dans ce menu se trouvent des clés (identifiants) qui permettent d'accéder au contenu du canal de l'extérieur. Il est nécessaire de les insérer dans nos requêtes. Il existe deux clés API : une pour accéder aux données en lecture (read API) et une autre pour accéder aux données en écriture (write API). Il est possible de générer d'autres nouvelles clés.



## Présentation du système d'irrigation et de son fonctionnement

Notre système d'irrigation est basé sur la carte Wemos D1 R2 présentée dans le chapitre précédent. Cette carte sera reliée à un capteur d'humidité et de température de l'air (DHT11), un capteur d'eau (Water sensor) et un capteur d'humidité au sol (YL69). Nous allons régulièrement vérifier le taux d'humidité au sol, dès que

21

celui-ci sera inférieur à un seuil donné. Alors, il faudra arroser les plantes. Notre carte reste toujours connectée au WiFi.

Pour effectuer l'arrosage, la carte Wemos est reliée à un relais qui est lui-même relié à une pompe à eau qui sera déclenchée dès que le seuil est atteint. Le capteur DHT11 permet de surveiller l'humidité de l'air, si celle-ci est supérieure à 90% le temps est pluvieux, donc on n'arrosera pas les plantes. Ceci permet de faire des économies considérables. Ceci est le but du système que nous avons conçu.

Le water sensor servira à contrôler notre réservoir d'eau, si ce volume est trop bas, l'application le signalera à l'utilisateur pour qu'il le réapprovisionne.

Les données des différents capteurs seront disponibles dans l'application et sous formes de graphiques dans le Cloud (ThingSpeak). Dans ThingSpeak nous avons des fields disponibles en lecture. Pour cela, toutes les 20 secondes, nous recueillerons les données des capteurs destinées au serveur ThingSpeak qui les enverra à l'application. Thingspeak sera le pont entre la carte Wemos et l'application.

L'application lance le système d'irrigation à l'aide du bouton (on/off). Si le "on" est activé, le "1" sera envoyé au field du ThingSpeak qui concerne le relais. Toutes les 20 secondes, la carte Wemos effectue une lecture dans ce field et s'il elle trouve "1" elle activera le relais et lancera la pompe. Si le "off" est activé, le "0" sera envoyé au field et la pompe sera à l'arrêt.

## **X - Conclusion**

Pour conclure, nous avons pu mettre en place un système d'irrigation automatique contrôlé par une application. Ce dernier est opérationnel et applicable en cas réel. Il automatise l'irrigation et permet d'économiser l'eau.

Les objectifs que nous nous étions fixés sont donc atteints.

Ce projet a été une occasion pour nous familiariser avec la carte Arduino et plus particulièrement la carte Wemos. Nous avons également appris à nous servir des différents capteurs et à relier la partie matérielle à la partie logicielle (application) et au cloud. Sans nul doute, à l'avenir, ces acquis pratiques nous seront d'une grande utilité que ce soit dans nos projets futurs que dans nos vies professionnelles.

Lors du développement du projet nous avons rencontré divers problèmes, que nous avons dû résoudre. Il a fallu s'adapter au composant, modifier l'application à plusieurs reprises pour arriver au résultat voulu. Nous avons appris qu'une conception de départ ne pouvait être figée, mais qu'elle pouvait et devait subir autant de modifications utiles mais nécessaires afin d'aboutir à l'objectif fixé. Ceci nous montre à quel point le marché des composants électroniques dédiés à cette discipline est très diversifié.

Globalement, nous sommes satisfaits de notre travail, il aurait pu être amélioré par l'utilisation d'une vraie caméra WiFi au lieu de la simulation avec le téléphone, en ajoutant des capteurs supplémentaires pour apprécier les besoins du sol en engrais. Cependant, faute de moyens, vu la rareté de ces capteurs et leurs coûts exorbitants, nous avons préféré réaliser ce projet avec des coûts raisonnables.

## Bibliographie

<https://fr.wikipedia.org/wiki/irrigation>

<https://irrigazette.com>

<https://www.swisscom.ch/fr/business/enterprise/themen/iot/pionnier-iot-walter-schmidt.html>

Data shit Wemos D1 R2

Data shit DHT

Data shit YL69

Data shit détecteur d'eau ;

Walter Schmidt :