

**Master Systèmes Informatiques Communicants**

**Solution Automatique d’Irrigation à Distance basé sur le Protocole Lorawan et Arduino**

**Responsable**

**M. Benslimane Abderrahim**

**Réalisé par**

**Kanté Alpha Oumar**

**NDiaye El Hadj Pathé**

**Tran Bao Thien**

**Année Universitaire**

2018 - 2019

**Remerciement**

Nous tenons à remercier sincèrement Monsieur Benslimane Abderrahim, qui en tant que tuteur s’est toujours montré à l’écoute tout au long de la réalisation de ce travail ainsi pour sa générosité, ses conseils fructueux qu’il nous a prodigués le long de notre projet et la grande patience dont il a su faire preuve, malgré ses charges professionnelles.

Nous remercions également Monsieur Sahand Khodaparast, qui nous a accompagné tout au long de notre projet et nous a guidé dans sa réalisation.

**Table des matières**

1. Introduction
2. Présentation du projets

2.1. Définitions

1. Internet of Things
2. Lorawan
3. TTN
4. Arduino

2.2. Objectifs

2.3. Présentation de l’existant

3. Conception

3.1. Schéma fonctionnel

3.2. Diagramme des Cas d’utilisation

3.3. Diagramme de séquence

3.4. Propositions d’une architecture

3.5. Proposition d’une solution de repli

4. Plan de realisation

4.1. Introduction

4.2. Présentation de l’environnement de développement et langages de programmation

4.3. Cahier de charges

4.4. Diagramme de Gantt

4.5. Repartition des Taches

5. Conclusion

**Introduction 1**

Aujourd’hui, nous sommes confronté à une raréfaction des ressources naturels nécessaires à notre subsistance. L’eau représente la plus importante de ces ressources mais surtout la plus primordiale à la planète et pour la subsistance elle même de l’homme.

Cependant, avec une population mondiale de plus de 7 milliards d’individus, l’augmentation des besoins en eau est forcée de croître. Ce qui entraîne une réduction drastique de cette ressources et une mauvaise répartition sur la planète. Plusieurs facteurs sont à l’origine de ce problème tels que l’agriculture avec l’irrigation abondante afin de répondre aux besoins alimentaires.

L’agriculture représente déjà trois quart de la consommation en eau et si rien n’ai fait cela devrait augmenter.

Il faudra réduire les besoins en eau de l’agriculture en contrôlant l’irrigation des plantes de manière à fournir à la plante la juste quantité d’eau dont elle a besoin.

Pour répondre à cette problématique nous allons mettre en oeuvre le projet suivant:

**Mise en place d’un système automatique d’irrigation à distance en utilisant arduino et LoraWan.**

Ce projet vise à utilisé l’Internet des Objets Connecté (IOT) dans le domaine de l’agriculture pour permettre une bonne gestion des ressources en eau. Cela permettra aux agriculteur d'accroître leur production tout en contrôlant chaque millimètre de leurs ressources en eau.

Pour ce faire nous allons utiliser la carte arduino qui est un composant électronique équipé d’un microcontrôleur utilisé en internet des objets et LoRaWAN qui est un protocole de télécommunication permettant la communication à bas débit, par radio, d'objets à faible consommation électrique communiquant selon la technologie LoRa.

Ce rapport est organisé comme suit:

Dans un premier temps nous aborderons l'étude de l’existant, ensuite nous allons parler de la conception et enfin traiter le plan de réalisation.

Introduction 2

L’ internet des objets est un sujet incontournable dans le domaine de l’industrie même s’il n’est pas un nouveau sujet. Au début des années 2000, Kevin Ashton a mis les bases d’une technologie qui deviendra plus tard l’internet des objets. Le concept était simple mais puissant. Si tous les objets de notre quotidien possédait des identifiants et était équipé d’une technologie sans fil, ces objets pourrait communiquer entre eux et pourrait être géré par un ordinateur. Ce concept pourrait nous permettre de pouvoir controler et gerer automatiquement les ressources et contribuer à une réduction drastique du gaspillage et des pertes et du coût de ces ressources.

Cela revient à mettre à la disposition des ordinateurs des moyens de pouvoir récolter des informations afin de comprendre le monde par leur propre moyens.

La technologie des capteurs permet à l’ordinateur de pouvoir observer, identifier et comprendre le monde sans être confronté à la limite des entrées par les hommes.

Aujourd’hui la plupart des obstacles de l’internet des objets ont été résolu.

L’identification des objets se trouve facilité avec la mise à disposition de meilleur techniques d’attribution des adresses IPv4 (NAT,CIDR) et la création d’une nouvelle version d’adresse IP qui est l’IPv6 permettant d’identifier des milliards d’objets connectés.

L'amélioration des réseaux sans fil (couverture, qualité, bande passante), représente un atout pour les objets connecté. Les objets connectés peuvent être utiliser dans tous les domaines.

Ainsi pour répondre à la problématique des l’internet des objets concernant l’identification, la connectivité sans fil, la collecte des données, et pour répondre au problème de gaspillage d’eau durant l’irrigation il nous a été proposé le sujet suivant:

**Mise en place d’un système automatique d’irrigation à distance en utilisant arduino et LoraWan.**

**Pour ce faire, nous allons dans un premier temps comprendre le sujet et établir un cahier de charge pour l'élaboration du sujet dans le deuxième semestre.**

**2. Présentation du projets**

**2.1. Définitions**

1. **Internet of Things**

L'Internet des objets, ou IdO (en anglais Internet of Things, ou IoT) est l'interconnexion entre [Internet](https://fr.wikipedia.org/wiki/Internet) et des objets, des lieux et des environnements physiques. L'appellation désigne un nombre croissant d'objets connectés à [Internet](https://fr.wikipedia.org/wiki/Internet) permettant ainsi une communication entre nos biens dits physiques et leurs existences numériques. Ces formes de connexions permettent de rassembler de nouvelles masses de données sur le réseau et donc, de nouvelles connaissances et formes de savoirs.

Considéré comme la troisième évolution de l'Internet, baptisé [Web 3.0](https://fr.wikipedia.org/wiki/Web_3.0) (parfois perçu comme la généralisation du [Web des objets](https://fr.wikipedia.org/wiki/Web_des_objets) mais aussi comme celle du [Web sémantique](https://fr.wikipedia.org/wiki/Web_s%C3%A9mantique)) qui fait suite à l'ère du [Web social](https://fr.wikipedia.org/wiki/Web_social), l'Internet des objets revêt un caractère universel pour désigner des objets connectés aux usages variés, dans le domaine de la [e-santé](https://fr.wikipedia.org/wiki/E-sant%C3%A9), de la [domotique](https://fr.wikipedia.org/wiki/Domotique) ou du [quantified self](https://fr.wikipedia.org/wiki/Quantified_self).

D'un point de vue conceptuel, l'Internet des objets caractérise des objets physiques connectés ayant leur propre identité [numérique](https://www.futura-sciences.com/tech/definitions/informatique-numerique-584/) et capables de communiquer les uns avec les autres. Ce réseau crée en quelque sorte une [passerelle](https://www.futura-sciences.com/tech/definitions/tech-passerelle-1273/) entre le monde physique et le monde virtuel.

D'un point de vue technique, l'IdO consiste en l'identification numérique directe et normalisée ([adresse IP](https://www.futura-sciences.com/tech/definitions/internet-adresse-ip-1856/), [protocoles](https://www.futura-sciences.com/tech/definitions/tech-protocole-1285/) smtp, http...) d'un objet physique grâce à un système de communication sans fil qui peut être une puce [RFID](https://www.futura-sciences.com/tech/definitions/tech-rfid-4187/), [Bluetooth](https://www.futura-sciences.com/tech/definitions/electronique-bluetooth-16527/) ou [Wi-Fi](https://www.futura-sciences.com/tech/definitions/internet-wi-fi-1648/).

**b. LoraWAN**

LoRaWAN est un protocole sans fil longue portée et basse consommation destiné à la construction de réseaux IoT. Les dispositifs IoT («nœuds») envoient de petits paquets de données à un nombre quelconque de «passerelles» qui peuvent se trouver dans la zone de plusieurs kilomètres d'un nœud via le protocole sans fil LoRaWAN. Les passerelles utilisent ensuite des communications plus traditionnelles comme les connexions Internet filaires pour transmettre les messages à un serveur réseau qui valide les paquets et transmet la charge utile de l'application à un serveur d'application.

La nature du réseau LoRa permet potentiellement aux dispositifs IoT de fonctionner pendant des années avec de petites batteries, en envoyant occasionnellement de petits paquets de données, en attendant un court laps de temps pour les messages de réponse, puis en fermant la connexion jusqu'à ce que davantage de données doivent être envoyées. Les appareils peuvent également être configurés de manière à ce qu'ils soient toujours à l'écoute des messages provenant de leurs applications, bien que cela nécessite évidemment plus d'énergie et puisse être plus approprié pour les appareils qui sont, par exemple, branchés à une prise murale

**c. TTN**

The Things Network est un réseau LoRaWAN communautaire et open source pour l'Internet des Objets

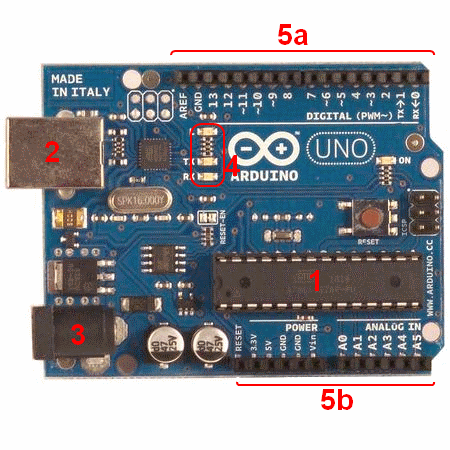
**d. Arduino:**

Arduino est une marque de cartes matériellement libres sur lesquelles se trouve un microcontrôleur. Les schémas de ces cartes sont publiés en licence libre et le logiciel arduino est gratuit et open source et est développé en Java ce qui rend l’utilisation simple. Cependant, certaines composantes comme le microcontrôleur par exemple ne sont pas sous licence libre.

Le microcontrôleur ne sait rien faire à la base d'où il peut être programmé pour analyser et produire des signaux électriques, de manière à effectuer des tâches très divers à savoir la domotique (contrôle des appareils domestiques), donner une intelligence à un robot, communiquer avec ordinateur, l’arrosage à distance etc.

**Présentation de la carte**

On va vous présenter la carte Arduino UNO qui est la plus utilisée et ses différents composants. On a représenté en rouge les composants les plus importants de la carte dans la figure ci-dessous.



*Figure 1: Carte Arduino UNO*

Micro-contrôleur  
  
Il est le cerveau de notre carte (en 1). C’est lui qui va recevoir le programme que vous aurez créé et qui va le stocker dans sa mémoire puis l’exécuter. Grâce à ce programme, il va pouvoir faire des traitements à savoir faire clignoter une LED, afficher des caractères sur un écran, envoyer des données à un ordinateur,..



*Figure 2: Le micro-contrôleur de la carte Arduino*

Alimentation

La carte Arduino a besoin d’une alimentation pour fonctionner. Le microcontrôleur fonctionnant sous 5V, la carte peut être alimentée en 5V par le port USB (en **2**) ou bien par une alimentation externe (en **3**) qui est comprise entre 7V et 12V. Cette tension doit être continue et peut par exemple être fournie par une pile 9V. Un régulateur sera alors nécessaire pour réduire la tension à 5V pour le bon fonctionnement de la carte.

Visualisation  
  
Les trois "points blancs" entourés en rouge (4) sont en fait des LED dont la taille est de l’ordre du millimètre. Ces LED servent à deux choses :

* Celle tout en haut du cadre : elle est connectée à une broche du microcontrôleur et va servir pour tester le matériel. Nota : Quand on branche la carte au PC, elle clignote quelques secondes.
* Les deux LED du bas du cadre : servent à visualiser l’activité sur la voie série (une pour l’émission et l’autre pour la réception). Le téléchargement du programme dans le microcontrôleur se faisant par cette voie, on peut les voir clignoter lors du chargement.

Connectique (en 5a et 5b)

La connectique est matérialisée par des entrées / sorties numériques ou analogiques.

* Les entrées ou sorties numériques

Il existe dans une carte Arduino 14 entrées ou sorties numériques de D0 à D13.



*Figure 3: Carte Arduino Entrées/Sorties numériques*

Les signaux véhiculés par ces connecteurs ne peuvent prendre que deux états HAUT (5 Volts) ou BAS (0 Volt). Courant de 40 mA maximum par sortie.

Les connecteurs D0 et D1 sont réservés pour la liaison USB et ne sont donc pas utilisés (RX et TX sont utilisés pour gérer les flux de données entrants et sortants).

Les connecteurs D3, D5, D6, D9, D10 et D11, repérés par un ~, peuvent être utilisés en sortie PWM, pour faire varier la luminosité d’une LED ou la vitesse d’un moteur.

Les sorties PWM peuvent avoir 28, soit 256 valeurs, allant de 0 à 255

* Les entrées analogiques

Il existe également 6 entrées analogiques: A0, A1, A2, A3, A4 et A5.



*Figure 4: Carte Arduino Entrées/Sorties analogiques*

Elles sont utilisées par les capteurs de températures, un potentiomètre, une résistance ajustable etc.

**2.2.Objectif**

L’objectif de ce projet consiste à construire un système d’irrigation automatique qui permettra d’approvisionner la plante en fonction de ses besoins en eau et selon les données d’humidité du sol et de température recueilli par des capteurs.

Le système doit envoyer périodiquement l'humidité du sol au serveur choisi. Une application doit être créée pour ce scénario sur le serveur LoRa choisi, qui enregistre les données reçues et les analyse pour calculer l’humidité moyenne et d’autres informations utiles. Le dispositif terminal doit pouvoir arroser les plantes lors de la réception des commandes de liaison descendante appropriées. La période d'arrosage d'une plante doit dépendre des commandes de liaison descendante. Par exemple, pour un sol très sec, la période d'arrosage peut durer environ 20 secondes, mais pour un sol normal, cette période doit être plus courte.

**2.3. Etude de l’existant**

Une plante a besoin d’eau pour vivre. Cependant chaque plante a des besoins en eau différent. Un système d’irrigation approvisionne la plante en eau dont elle a besoin.

Les systèmes d’irrigation existant approvisionne la plante en eau sans prendre en compte les besoins exacts de la plante ni le moment exact pour irriguer la plante. Ce qui ne permet pas une meilleur utilisation de l’eau qui est une denrée rare.

Réaliser un système automatique d’irrigation distance consiste à répondre aux défis suivants:

* Mesurer les besoins de la plante en eau
* Irriguer la plante en fonction de ses besoins
* Réduire le gaspillage de l’eau
* Automatiser l’irrigation en fonction du temps ou en fonction de certain paramètre comme l’humidité du sol, la température
* Utilisé des réseaux sans fil adapté à un taux données très réduits
* Mettre en place un système moins gourmand en énergie facilement configurable.
* Récolter et interpréter les données nécessaire à une irrigation automatique.

**3. Conception**

Ce projet vas nous permettre de construire un contrôleur d'irrigation pour un jardin domestique. Capable de mesurer les lectures d'humidité du sol et d'activer l'irrigation d'un robinet de jardin si le sol devient trop sec. Le contrôleur comprend également un capteur de température et d'humidité. Le contrôleur n'activera pas le robinet de jardin si la température est trop basse. Les lectures de capteurs et les statistiques sur l'utilisation de l'eau / les durées de fonctionnement sont enregistrées sur IOT ThingsBoard à des fins de visualisation et d'analyse. Des alertes sont déclenchés si le contrôleur d'irrigation arrête de transmettre des données, le sol devient trop sec ou trop saturé. Nous utiliserons arduino et la technologie LoraWan **qui** est un protocole de télécommunication permettant la communication à bas débit, par radio, d'objets à faible consommation électrique communiquant selon la technologie **LoRa.**

3.1. Schéma fonctionnel

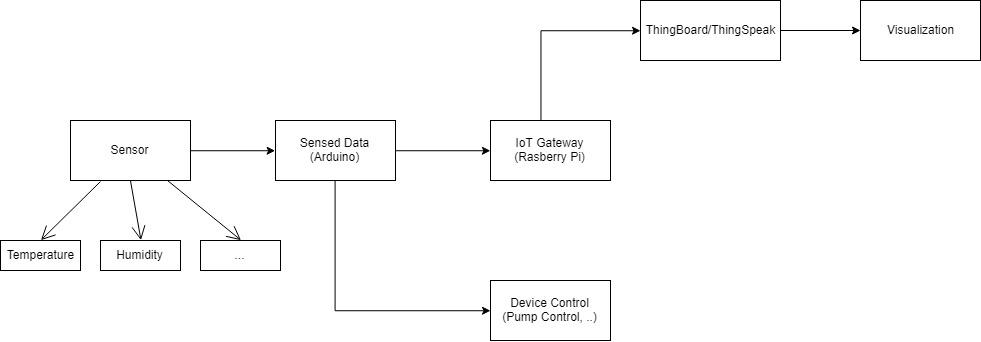


Figure 5: Schéma fonctionnel

3.2. Diagramme des Cas d’utilisation

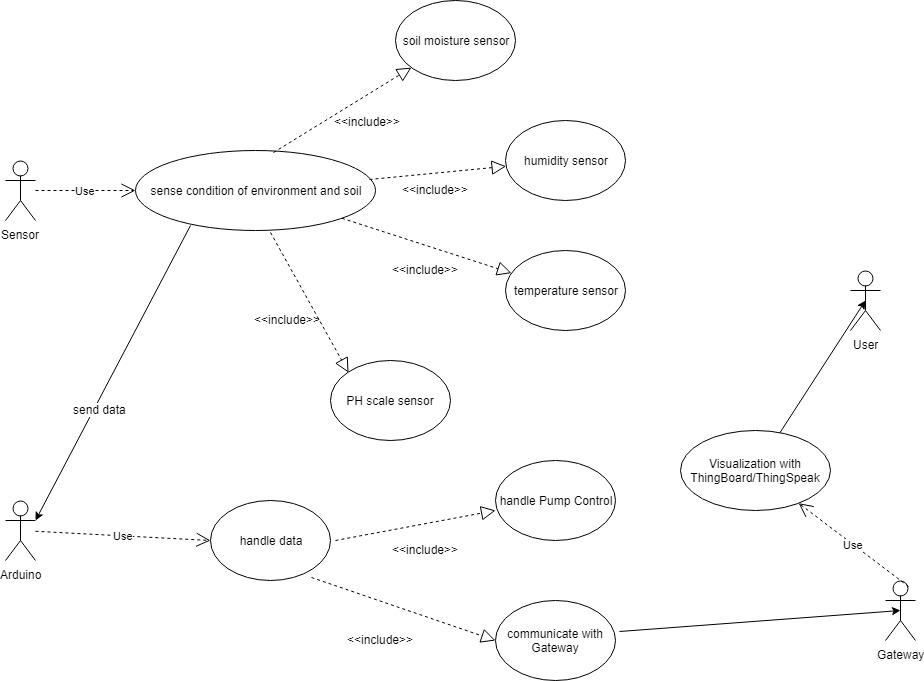
****

Figure 6: Diagramme des Cas d’utilisation

3.3. Diagramme de séquence

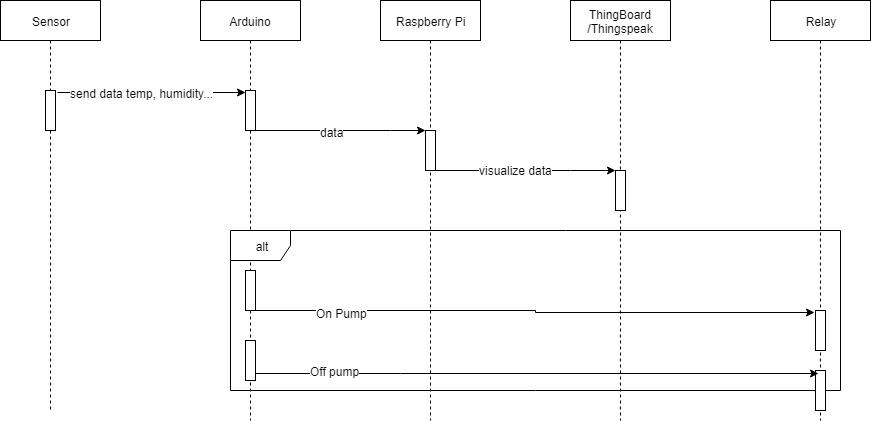
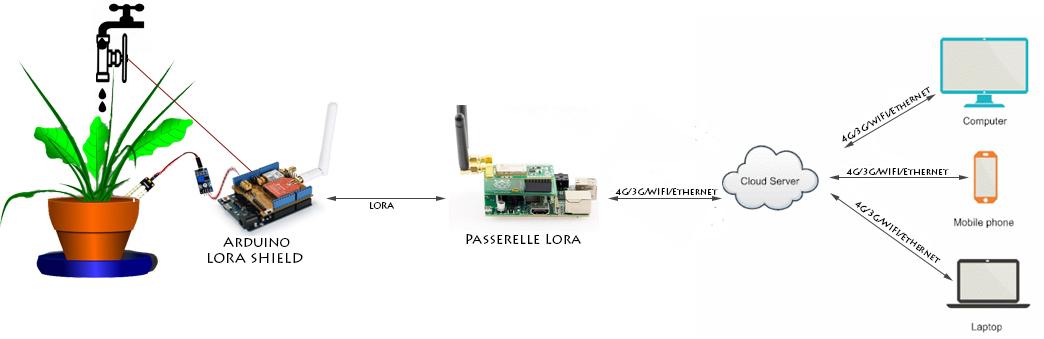
****

Figure 7: Diagramme de séquence

3.4. Propositions d’une architecture



**Figure 5: Architecture de deploiement**

**Arduino Uno**

Arduino Uno est une carte à microcontrôleur basée sur l’ATmega328. Il possède 20 broches d'entrée / sortie numériques (dont 6 peuvent être utilisées en tant que sorties PWM et 6 en tant qu'entrées analogiques), un résonateur 16 MHz, une connexion USB, une prise d'alimentation, une programmation système intégrée (ICSP) en-tête et un bouton de réinitialisation. Il contient tout le nécessaire pour prendre en charge le microcontrôleur. connectez-le simplement à un ordinateur (ou à l'adaptateur secteur approprié) avec un câble USB ou alimentez-le avec un adaptateur CA / CC ou une batterie pour commencer.

L'Uno diffère des cartes précédentes en ce sens qu'il n'utilise pas la puce de pilote FTDI USB-to-serial. Au lieu de cela, il dispose d'un ATmega16U2 programmé en tant que convertisseur USB / série. Ce microcontrôleur auxiliaire possède son propre chargeur de démarrage USB, ce qui permet aux utilisateurs avancés de le reprogrammer.

**Lora Shield**

LoRa Shield est un émetteur-récepteur longue portée basé sur une bibliothèque Open Source. Il permet à l'utilisateur d'envoyer des données et d'atteindre des distances extrêmement longues à des débits de données très bas.Il fournit une communication à spectre étalé à très longue portée immunité aux interférences tout en minimisant la consommation de courant.

LoRa Shield est basé sur la puce Semtech SX1276 / SX1278. Il cible les applications réseau professionnelles de capteurs sans fil telles que les systèmes d'irrigation, les compteurs intelligents, les villes intelligentes, la détection de smartphones, l'automatisation des bâtiments, etc.

**Passerelle Lora**

Les passerelles constituent le pont entre les appareils et le réseau d'objets. Les périphériques utilisent des réseaux à faible consommation comme LoRaWAN pour se connecter à la passerelle, tandis que la passerelle utilise des réseaux à bande passante élevée tels que WiFi, Ethernet ou cellulaire pour se connecter au réseau The Things.

Toutes les passerelles à la portée d’un appareil recevront les messages de celui-ci et les achemineront vers le réseau The Things. Le réseau dédupliquera les messages et sélectionnera la meilleure passerelle pour transférer les messages en file d'attente pour la liaison descendante. Une seule passerelle peut desservir des milliers de périphériques.

**Relai**

Étant donné que l’Arduino fonctionne à 5V, il ne peut pas contrôler directement ces appareils à tension plus élevée, mais vous pouvez utiliser un relais de 5V pour commuter le courant 120-240V et utiliser l’Arduino pour contrôler le relais.

L'Arduino peut être programmé pour activer le relais lorsqu'un certain événement se produit, par exemple lorsque la température d'un thermistor dépasse 30 ° C.

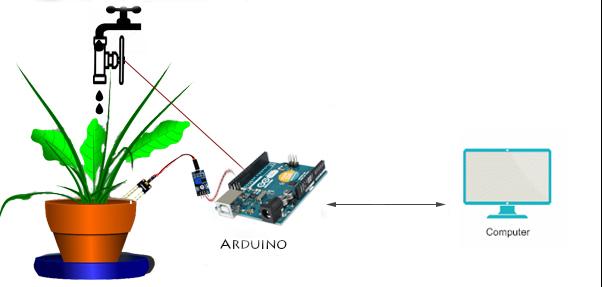
Le relais SRD-05VDC-SL-C possède trois bornes haute tension (NC, C et NO) qui se connectent à l'appareil que vous souhaitez contrôler. L’autre côté a trois broches basse tension (Terre, Vcc et Signal) qui se connectent à l’Arduino.

**Description des differents composants de l’architecture**

3.5. Proposition d’une solution de repli

Une solution alternative à notre projet consisterait à mettre en place un système d’irrigation mais qui ne soit pas a distance. L’objectif sera alors de réaliser le même système sans intégrer la technologie Lora. Nous aurons donc notre carte arduino connecté directement aux capteurs d’humidité, à la pompe, et à l’ordinateur.

Sur l’ordinateur, nous allons écrire le programme pour récolter et traiter les données des capteurs et permettre de déclencher l'arrosage par pompe lorsqu’une valeur limite est atteinte pour la température et l’humidité du sol. Dans ce cas de figure, nous n’aurons plus besoin de Lora Shield pour arduino et de pas non plus besoin de passerelle LORA.



4. **Plan de realisation**

4.1. Introduction

Pour la partie logicielle du projet nous allons utiliser l’IDE Arduino pour le développement des programmes nécessaires pour la gestion des capteurs,le traitement des données et la prise de décision concernant l’irrigation en fonction des conditions que nous aurons établi à cet effet.

L’IDE Arduino utilise le langage de programmation C/C++ qui est très riche en librairie permettant la gestion des IOT.

4.2. Présentation de l’environnement de développement et langages de programmation

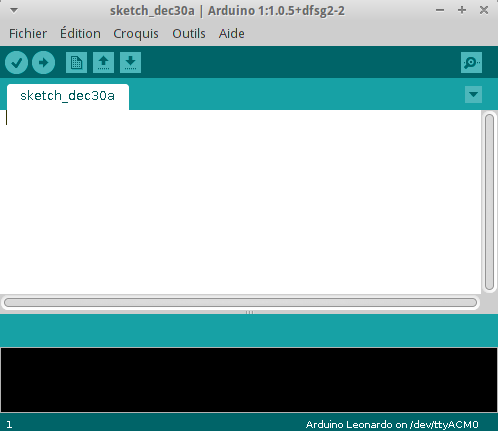
4.2.1 **Installation de l’IDE Arduino**

L’IDE Arduino est le logiciel qui nous permettra de programmer notre carte Arduino. Nous aurons besoin d’un PC (Windows, Mac ou Linux, Arduino est compatible avec ces 3 OS) sur lequel nous allons installer le logiciel Arduino IDE et d’une connexion à internet.

Pour télécharger le logiciel, il faut se rendre sur la page de téléchargement du site <https://www.arduino.cc/en/Main/Software>.

4.2.2 **Configuration minimale:**

Après installation du logiciel, une fenêtre s’ouvre et se présente comme ci-dessous:



Il nous faut maintenant configurer le logiciel pour l’adapter à notre carte :

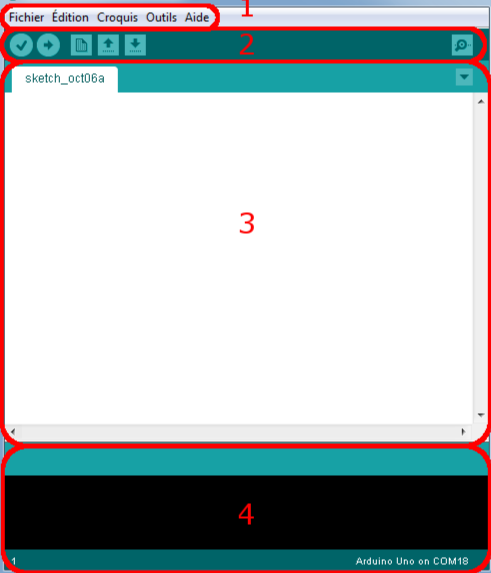
Dans un premier temps, nous allons dans la barre des menus cliquer sur « **Outils** » puis sur « **type de carte** » et sélectionner la carte que nous souhaitons connecter ( UNO, Nano, Leonardo … ).

Ensuite, avant de connecter la carte Arduino, vérifier les ports qui sont déjà présents sur l’IDE. Pour ce faire, toujours dans « **Outils** » cliquer sur « **Port** » , nous verrons alors l’ensemble des ports déjà présents.

Enfin nous allons choisir le port ou notre carte arduino est branchée ainsi nous pourrons téléverser notre programme vers la carte.

4.2.3 **Présentation de l’outi**l

Pour mieux vous présenter l’interface de l’IDE arduino nous l’avons découpée en quatre partie comme l’indique le capture ci-dessous.



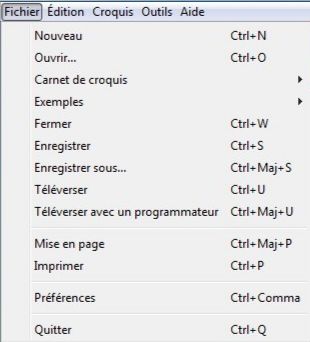
*Figure:* *Présentation de l’IDE*

Correspondence des cadres:

* Le cadre numéro 1 : ce sont les options de configuration du logiciel
* Le cadre numéro 2 : il contient les boutons qui vont nous servir lorsque nous programmons notre carte arduino
* Le cadre numéro 3 : ce bloc va contenir le programme que nous allons créer
* Le cadre numéro 4 : celui-ci est important, car il va nous aider à corriger les erreurs dans notre code. C’est le **débogueur.**

Le menu *File*

C’est principalement ce menu qu’on va plus utiliser. Il dispose d’un certain nombre de choses qui vont nous être très utiles. Nous allons donc voir les quelques options qui sortent de l’ordinaire :



*Figure: Menu fichier IDE Arduino*

* *Carnet de croquis* : Ce menu regroupe les fichiers que vous avez pu faire jusqu’à maintenant (et s’ils sont enregistrés dans le dossier par défaut du logiciel).
* *Exemples* (exemples) : Ceci est important, toute une liste se déroule pour afficher les noms d’exemples de programmes existants ; avec ça, vous pourrez vous aider/inspirer pour créer vos propres programmes ou tester de nouveaux composants.
* *Téléverser* : Permet d’envoyer le programme sur la carte Arduino. Nous y reviendrons.
* *Téléverser avec un programmateur* : Idem que ci-dessus, mais avec l’utilisation d’un programmateur (vous n’en n’aurez que très rarement besoin).
* *Préférences* : Vous pourrez régler ici quelques paramètres du logiciel. Le reste des menus n’est pas intéressant pour l’instant, on y reviendra plus tard, avant de commencer à programmer.

**Les boutons raccourcis**

Voyons à présent à quoi servent les boutons, encadrés en rouge et numérotés par le chiffre 2.



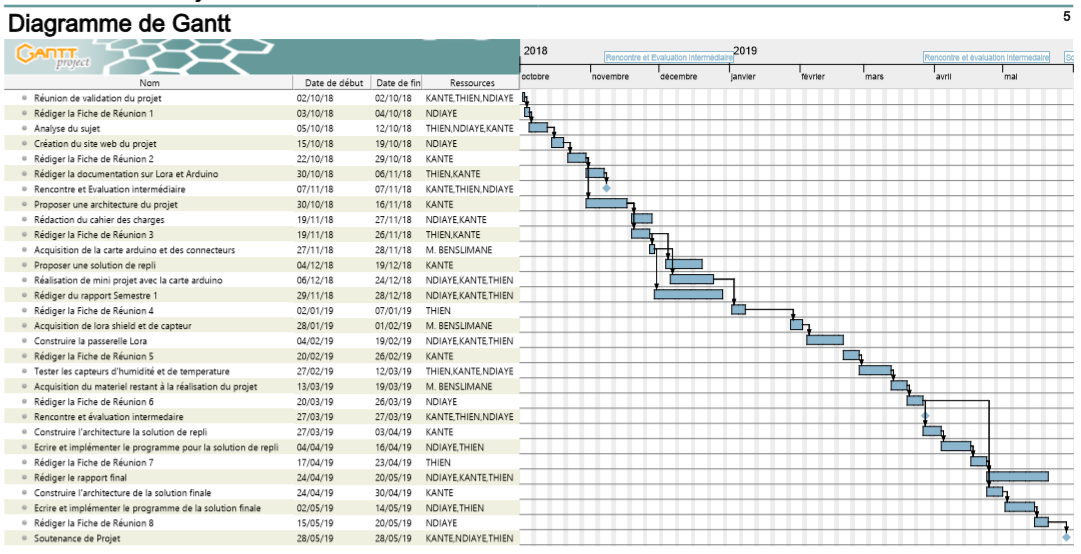
*Figure: Boutons Raccourcis IDE*

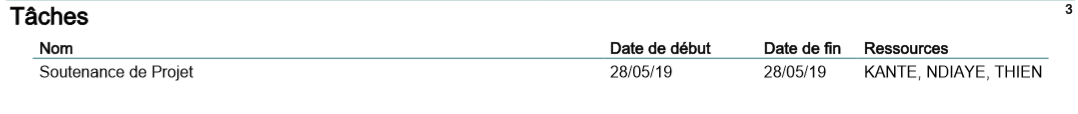
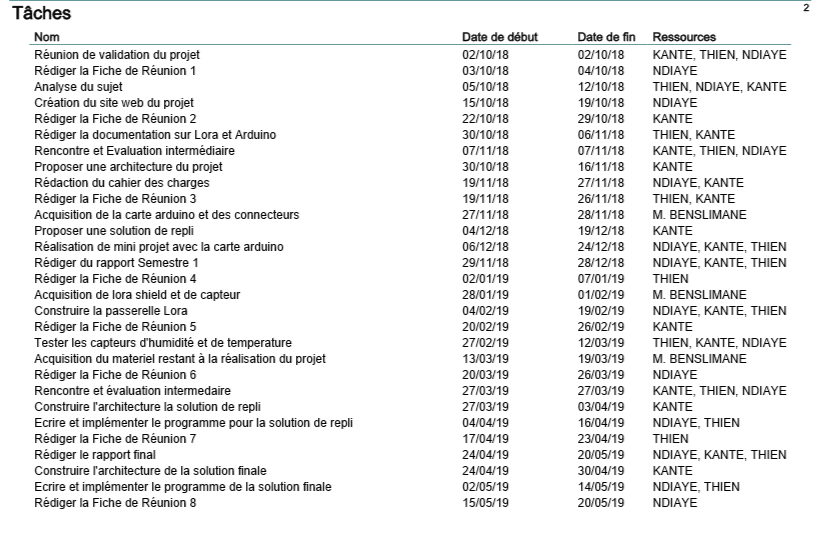
* Bouton 1 : Ce bouton permet de vérifier le programme, il actionne un module qui cherche les erreurs dans notre programme
* Bouton 2 : Charge (téléverse) le programme dans la carte Arduino.
* Bouton 3 : Créer un nouveau fichier.
* Bouton 4 : Ouvre un fichier.
* Bouton 5 : Enregistre le fichier.
* Bouton 6 : Ouvre le moniteur série

**4.3. Cahier de charges**

**4.4. Diagramme de Gantt**

Le diagramme de GANTT est un outil efficace exploitant des données brutes tel que les dates de début et de fin et les durées de chacune des tâches afin de générer une visualisation de l’avancement du projet. Il permet de donner une vue globale des tâches à réaliser, des responsabilités et des ressources associées, de l’idée jusqu’à la mise en service en passant par l’analyse des exigences, l’étude de faisabilité, la conception fonctionnelle, les spécialisations, la réalisation et enfin, les tests. Aussi, il est possible d’organiser une gestion des ressources, leurs disponibilités, leurs coûts, etc.



4.4. Repartition des Taches 

**Cahier de charges**

**5. Conclusion**

À mesure que notre technologie se connecte de plus en plus grâce à l'essor de l'IdO et de l'informatique en nuage, nous avons non seulement profité à nos vies, mais aussi à notre planète. L'agriculture intelligente a pris le monde d'assaut, car les agriculteurs ont appris à embrasser le monde de l'IdO, et non à le rejeter. Ces agriculteurs ont été en mesure de surveiller avec précision l'état de leurs produits et de contrôler chaque millilitre de leurs ressources.

Conclusion 2:

Une application de la technologie IoT au domaine de l’agriculture sera un avantage primordiale afin de contrôler les ressources en eau mais surtout d'accroître la productivité agricole. L'intérêt étant d’aller vers une agriculture beaucoup plus intelligente c’est à dire simple, efficace et rentable.

L'exécution de ce projet nous a permis de prendre en compte ce nouveau caractère intelligent de l'agriculture mais aussi d'acquérir une expérience dans le domaine de l’Internet des objets (IoT) et va permettre à son aboutissement aux acteurs qui interviennent dans ce domaine d’avoir à leurs dispositions plusieurs données importantes en temps réel comme la période d’arrosage, l'humidité, la température du sol, etc.

Dans la réalisation de ce projet, nous avons dans un premier temps fait une étude de l’existant, afin de proposer une solution. Ce qui nous a permis d'élaborer les diagrammes fonctionnel et de cas d’utilisation et de proposer une architecture de déploiement. Ensuite nous avons établi le diagramme de Gantt et la répartition des tâches afin de respecter le délai. **Enfin on a réalisé six(6) petits projets simple afin d’avoir la main avec le matériel.**

Au second semestre, nous allons mettre en oeuvre la solution d’irrigation à distance. A cet effet il va falloir acquérir le reste du matériel, écrire et déployer le program d'exécution en C, et le déployer l’architecture proposé, récolter et traiter les données des capteurs afin de produire des conditions d’irrigations de la plante. La finalité étant à terme de pouvoir faire des tests et un système fonctionnel.

Arduino Uno

Lora Arduino Shield

Lora Gateway

Humidity sensor

Temperature sensor

Cable

Battery

Electric Water pompe

Lora server app

5v relay

**Annexe**

Références

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

<https://www.futura-sciences.com/tech/definitions/internet-internet-objets-15158/>

<https://eskimon.fr/tuto-arduino-103-le-logiciel>