

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche
scientifique

Université Saad Dahleb Blida 1



Faculté de technologie
Département d'électronique

MÉMOIRE

Pour l'obtention du diplôme de Master

Domaine : Sciences techniques

Filière : Electronique

Option : Electronique des systèmes embarqués

Thème :

**CONCEPTION ET REALISATION D'UNE
MAQUETTE DE FERME INTELLIGENTE AVEC
ARDUINO**

Présenté et soutenu publiquement par:

Boukacem Mohamed

Sous la direction de :

Ammar Bounemri

Devant le jury :

Présidente: Mme. Naceur

Encadreur: Mr. AMMAR BOUNEMRI

Examinatrice: Mme. Kaoula.

Année universitaire: 2022-2023

REMERCIEMENTS

- Nous tenons tout d'abord à remercier **ALLAH** le tout puissant, qui nous a donné la force et la patience d'accomplir ce modeste travail.
- Nous tenons à exprimer nos vifs remerciements à notre promoteur **Mr AMMAR BOUNEMRI** qui nous a conseillé et encouragé tout le long de notre travail.
- Nous remercions chaleureusement les membres du jury pour l'honneur qu'ils nous ont fait en acceptant d'évaluer notre travail.
- Nos remerciements vont également à tous les enseignants qui ont contribué à notre formation.
- Enfin, nos remerciements vont à toute personne ayant contribué, de près Ou de loin, pour la réalisation de ce travail.

DÉDICACES

Avec l'aide d'**ALLAH** j'ai pu réaliser ce modeste travail que je dédie :

- Je dédie ce modeste travail à mon très cher **père**, ma très chère **mère**.
- A ma chère **femme** qui m'a beaucoup soutenu pendant mon cycle d'étude.
- A mes filles : **Aya, Nourhane et Safaa**.
- A mes chers **frères et soeurs**.
- A mon cher ami **Benaicha Abdelkader**.
- Enfin, nos remerciements vont à toute personne ayant contribué, de près Ou de loin, pour la réalisation de ce travail.

ملخص

هدف هذا المشروع هو إنشاء نموذج تجريبي لمزرعة ذكية يتوافق مع المعايير الأساسية ويضمن الأداء الجيد.

هذا النظام يتكون من عدة عناصر مثل : مستشعر رطوبة الأرض لتشغيل وإيقاف مضخة الري أيضا مستشعر الحرارة DHT11 لتشغيل وإيقاف التهوية بالإضافة إلى مستشعر الحريق مع خاصية الكشف عن الأشعة تحت الحمراء. ويتم تشغيل هذا النظام بشكل آلي باستخدام لوحة أردوينو يونو ويمكن التحكم فيه عن بعد باستخدام تقنية البلوتوث HC_05 باستخدام شيفرة المصدر التي سنصممها حسب الحاجة.

كلمات مفتاحية: مزرعة ذكية، أردوينو يونو، محرك، مستشعر الحرارة، مصباح ، صفارة، ثنائي ضوئي، نموذج تجريبي.

Résumé

Le but de ce travail est de réaliser un prototype de maquette de ferme intelligente qui répondra aux critères de bases et qui assurera un bon fonctionnement.

Ce système est composé de plusieurs élément comme un capture d'humidité de sol pour la marche et l'arrêt de la pompe d'arrosage. Aussi un capteur de température DHT11 pour la marche et l'arrêt de ventilation ainsi qu'un capteur de flamme à détection infrarouge pour l'alarme. Ce système sera automatisé via une carte Arduino Uno et commandée à distance via la technologie Bluetooth grâce à un code source que nous allons concevoir selon le besoin.

Mots-clés : Ferme intelligente, Arduino Uno, Actionneur, DHT11 , HC_05 ,Buzzer , Led ,lampe. Prototype.

Abstract

The aim of this project is to create a prototype of a smart farm model that will meet the basic criteria and ensure proper functionality.

This system consiste of several elements such as soil humidity sensor to turn on and stop the irrigation pump, as well as a temperature sensor DHT11 to turn on and off ventilation, in addition to q fire sensor with an infrared detection feature. This system will be automated using an Arduino Uno board and controlled remotely through Bluetooth technology HC_05 using source code that we will design as needed.

Keywords: Smart Farm, Arduino Uno, Actuator, DHT11, HC_05, Led, Buzzer, Lamp, Prototype.

Sommaire

Introduction générale	1
Chapitre I : Généralités sur l'agriculture intelligente	
I.1.Introduction	2
I.2. Motivation et contexte du travail	2
I.3. Cahier des charges	3
I.4. Objectifs à réaliser	3
I.5. Approche Philosophique/Historique	3
I.6. Généralités sur les systèmes automatisés	4
I.6.1. Système technique et automatisation	4
I.6.2. Système automatisé	4
I.6.3. Différentes parties d'un système automatisé	5
I.7.4.Partie opérative	5
I.7.5. Partie commande	6
I.7.6. Partie relation	6
I.6.4.Mieux comprendre le système automatisé	6
I.6.5.Avantages et inconvénients de l'automatisation	7
I.6.5.1. Avantages de l'automatisation	7
I.6.5.2. Inconvénients de l'automatisation	8
I.7. L'agriculture intelligente	8
I.7.1. Définition de l'agriculture intelligente	8
I.7.2. Domaines de l'agriculture intelligente	9
I.7.3. Avantages et inconvénients de l'agriculture intelligente	11
I.7.3.1. Avantages de l'agriculture intelligente	11
I.7.3.2. Inconvénients de l'agriculture intelligente	13
I.8. Conclusion	13
Chapitre II : Matériels et logiciels utilisés	
II.1. Introduction	14
II.2. Partie Matériels	14
II.2.1. Choix de la carte programmable	15
II.2.2 Justification du choix de l'Arduino	15
II.2.3 Définition d'un Arduino	16
II.2.4 Définition d'une carte Arduino	16
II.2.6 Arduino Uno	16
II.2.6.1. Choix de l'ArduinoUno	17
II.2.6.2. Spécifications techniques	19
II.3 Alimentation et branchement	19
II.3.1. Transformateur	19
II.3.2. Partie câblage	20
II.4. Capteurs	21
II.4.1 Définition d'un capteur	21
II.4.2 Capteurs utilisés	21
II.4.2.1 DHT11	22

II.4.2.2	Capteur de flamme	23
II.4.2.3	Capteur d'humidité de sol	25
II.5	Pré-actionneurs	27
II.5.1	Définition du pré-actionneur	27
II.5.2.	Pré-actionneur utilisé	27
II.5.2.1.	Relais 4 canaux	27
II.6.	Actionneurs	29
II.6.1	Définition d'un actionneur	29
II.6.2	Actionneurs utilisés	29
II.6.2.1	Pompe	29
II.6.2.2	Ventilateur	30
II.6.2.3	Eclairage	31
II.6.2.4	Branchement de l'actionneur avec le relais	32
II.7.	Les autres composants	33
II.7.1	Les leds	33
II.7.2.	Buzzer	34
II.7.3.	Bluetooth	35
II.8	Partie logiciel	35
II.8.1	Logiciel Arduino	35
II.8.1.1	Définition du logiciel Aduino	35
II.8.1.2	Langage de programmation Arduino	36
II.8.1.3	Structure d'un programme Arduino	36
II.8.2	Application RemoteXY	41
II.8.2.1	Présentation	41
II.8.2.2	Eléments de contrôle	41
II.8.2.3	Configuration	42
II.8.2.4	HC-05 et RemoteXY	43
II.9.	Conclusion	44

Chapitre III : Conception et réalisation finale

III.1.	Introduction	45
III.2.	Etudes des besoins du système	46
III.2.1.	Besoins fonctionnels	46
III.2.2	Besoins non fonctionnels	46
III.3.	Principe de fonctionnement	47
III.3.1.	Principales étapes du processus	47
III.3.2.	Organigramme du systeme	49
III.4.	Software	52
III.4.1	Création de l'interface	52
III.4.1.1	L'éditeur	52
III.4.1.2	Configuration des paramètres	53
III.4.1.3	Liaison à l'interface	55
III.4.1.4	Vue de l'interface	55
III.4.1.5	Décoration de l'interface	56
III.4.1.6	Indicateurs	57
III.4.1.7.	Contrôle de l'interface	59

III.4.1.8	Sauvegarde du projet	60
III.4.1.9	Obtention du code	61
III.4.2	Programmation	62
III.4.2.1	Code source	63
III.4.2.2	Compartiment de refroidissement	65
III.4.2.3	Compartiment d'arrosage automatique	66
III.4.2.4	Compartiment d'éclairage	67
III.4.2.5	Compartiment d'incendie	68
III.5	Réalisation	69
III.5.1	Test	69
III.5.2	Fabrication de la maquette	71
III.5.3	Finition	73
III.5.4	Résultat final	74
III.6.	Conclusion	76

Liste des figures

Chapitre I

Figure 1.1 :Une série de systèmes automatisés destine à souder la carrosserie des véhicules	4
Figure 1.2 :Architecture générale d'un système automatisé.	5
Figure 1.3 :Exemples de capteurs	5
Figure 1.4 :Comparaison d'un système automatisé avec le corps humain	6
Figure 1.5 :Photo représentative de l'agriculture intelligente	9
Figure 1.6 :Photo illustrant l'utilisation d'un drone dans une ferme	10

Chapitre II

Figure 2.1 : Photo représentative des systèmes électroniques embarqués	13
Figure 2.2 : Logo d'Arduino.	15
Figure 2.3 : Photo réelle de la carte Arduino UNO	16
Figure 2.4(a) : Photo réelle du transformateur	19
Figure 2.4 (b) : Photos réelles de la prise jack utilisée	19
Figure 2.4 (c) : Photo réelle de la plaque d'essai et des fils	20
Figure 2.5 (a) : Photo réelle du DHT11	21
Figure 2.5 (b) : Photo du branchement du DHT11 avec Arduino	22
Figure 2.5 (c) : Photo réelle du capteur de flamme	23
Figure 2.5 (d) : Photo du branchement du capteur de flamme avec Arduino	24
Figure 2.5 (e) : Photo réelle du capteur d'humidité de sol	25
Figure 2.5 (f) : Photo du branchement du capteur d'humidité de sol avec Arduino	26
Figure 2.6 (a) : Photo réelle du module relais 4 canaux	27
Figure 2.6 (b) : Photo du branchement du module relais avec Arduino	28
Figure 2.7 (a) : Photo réelle de la mini pompe submersible	29
Figure 2.7 (b) : Photo réelle du ventilateur	30

Figure 2.7 (c) : Photo de la mini lampe utilisée	31
Figure 2.7 (d) : Photo du câblage d'un actionneur avec Arduino via relais	31
Figure 2.8 (a) : Photo réelle des leds et leurs résistance	32
Figure 2.8 (b) : Photo réelle du buzzer	33
Figure 2.8 (c) : Photo réelle du module Bluetooth HC-0	33
Figure 2.9 (a) : Logo du l'IDE de l'Arduino	34
Figure 2.9 (b) :Interface du logiciel	35
Figure 2.9 (c) :Fenêtre d'édition de programme	36
Figure 2.9 (d) :Type de carte	36
Figure 2.9 (e) :Problème de la sélection du port	37
Figure 2.9 (f) :Configuration du port	37
Figure 2.9 (g) :Choix du port	37
Figure 2.9 (h) :Barre d'outils Arduino	37
Figure 2.9 (i) :Photo réel du téléversement du programme vers la carte ArduinoUno	38
Figure 2.9 (j) :Rôles des instructions de la barre de commande	38
Figure 2.10 (a) :Logo de RemoteXY	39
Figure 2.10 (b) :Photo explicative des éléments disponibles sur RemoteXY	40
Figure 2.10 (c) :Photo explicative de la configuration de RemoteXY	41
Figure 2.10 (d) : Photo explicative du fonctionnement de RemoteXY	41
<u>Chapitre III</u>	
Figure 3.1 :Photo des composants du compartiment de refroidissement	47
Figure 3.2 :Photo des composants du compartiment d'incendie	47
Figure 3.3 :Photo des composants du compartiment d'arrosage automatique	48
Figure 3.4 (a) :Orgqigramme de notre ferme intelligente (arrosage)	49
Figure 3.4 (b) :Orgqigramme de notre ferme intelligente (refroidissement)	49
Figure 3.4 (c) :Orgqigramme de notre ferme intelligente (incendie)	50
Figure 3.4 (d) Orgqigramme de notre ferme intelligente (eclairage)	50

Figure 3.5 :Photo de la page principale du site RemoteXY	51
Figure 3.6 :Photo de l'éditeur de l'interface	51
Figure 3.7 (a) :Choix de la cart	52
Figure 3.7 (b) :Choix du mode de connexion	52
Figure 3.7 (c) :Choix du module Bluetooth	53
Figure 3.7 (d) :Choix de l'interface de programmation	53
Figure 3.8 :Configuration des propriétés de l'interface	54
Figure 3.9 :Liaison avec l'interface	54
Figure 3.10 (a) :Vue de l'interface	55
Figure 3.10 (b) :Arrière-plan de notre application	55
Figure 3.11 (a) :Choix de décoration	56
Figure 3.11 (b) :Modification du texte	56
Figure 3.11 (c) :Orientation du texte	56
Figure 3.12 (a) :Choix de l'indicateur	57
Figure 3.12 (b) :Configuration du texte	57
Figure 3.12 (c) :Orientation des textes	58
Figure 3.13 (a) :Choix de l'outil de contrôle	58
Figure 3.13 (b) :Configuration du switch	59
Figure 3.13 (c) :Orientation du switch	59
Figure 3.14 :Onglet de sauvegarde	60
Figure 3.15 (a) :Obtention du code source	60
Figure 3.15 (b) :Téléchargement du code et de la librairie	61
Figure 3.16 (a) :Photo réelle du test sous l'angle A	68
Figure 3.16 (b) :Photo réelle du test sous l'angle B	69
Figure 3.16 (c) :Photo réelle du test sous l'angle C	69
Figure 3.16 (d) :Photo réelle du test sous l'angle D	69
Figure 3.17 :Photo réelle du découpage	70

Figure 3.18 :Photo réelle du perçage	71
Figure 3.19 :Photo réelle de la maquette après assemblage	71
Figure 3.20 (a) :Fixation du gazon	72
Figure 3.20 (b) :Maquette après habillage de la base avec gazon	73
Figure 3.21 (a) :Photo réelle de la réalisation finale sous l'angle A	74
Figure 3.21 (b) :Photo réelle de la réalisation finale sous l'angle B	74
Figure 3.21 (c) :Photo réelle de la réalisation finale sous l'angle C	74
Figure 3.21 (d) :Photo réelle de la réalisation finale sous l'angle D	74



Introduction générale

Introduction Générale :

Il y a d'abord eu les smartphones, puis les Smart Houses (maisons connectées). Aujourd'hui, les Smart Farms ou fermes intelligentes sont en passe de révolutionner le monde agricole...

Ce travail vise à mieux comprendre ce sujet. En effet, nous avons conçu, fabriqué et réalisé une ferme intelligente qui sera commandé par une carte programmable Arduino Uno.

Nous avons aussi utilisé une application correspondante à notre projet grâce à l'interface Remote XY.

Notre mémoire est composé de trois chapitres :

- Le premier chapitre présente les concepts de base nécessaires à la bonne compréhension du sujet en question.
- Le deuxième chapitre sera réservé aux différents matériels et logiciels nécessaires pour la réalisation de notre système.
- Le troisième chapitre sera consacré à la conception ainsi que la réalisation pratique de notre projet.

Ce mémoire sera terminé par une conclusion générale qui récapitule des différentes phases de notre travail en signalant ses côtés bénéfiques et énonçant quelques perspectives pour améliorer le système dans le futur.



Chapitre I :
Généralités Sur
L'Agriculture Intelligente

I.1. Introduction

L'évolution fulgurante de la technologie depuis maintenant plusieurs années est bénéfique pour l'être humain. Ce dernier a toujours eu tendance à se simplifier la vie à plusieurs niveaux.

En effet, le nouveau développement technologique a permis d'élaborer de nouveaux besoins dans différents domaines. L'un d'eux est le système d'agriculture intelligente avec commande et contrôle à distance, grâce à des technologies et des techniques très avancées et très répandues dans ce domaine. Ce dernier a progressé d'une manière impressionnante, et cela au cours des deux dernières décennies de l'actuel siècle.

Dans ce chapitre, nous allons exposer d'abord le cahier des charges de notre projet ainsi que nos objectifs. Nous donnons ensuite un aperçu sur les fermes intelligentes en les présentant, en évoquant l'historique et les différents types de systèmes technologiques et d'autres volets importants concernant ce domaine afin de pouvoir mieux cerner le concept.

I.2. Motivation et contexte de travail

Un système automatisé est composé de plusieurs éléments conçus pour effectuer un ensemble de tâches programmées. Il simplifie, sécurise et rend moins pénibles les tâches répétitives et opérationnelles. Le système de la ferme intelligente regorge d'avantages divers qui nous ont motivés à faire la réalisation de ce projet au détriment d'autres thèmes, l'agriculture intelligente à un poids considérable dans l'environnement industriel et commercial.

En effet, de nos jours, les technologies de la ferme intelligente sont faciles à utiliser par les agriculteurs, de plus, ils préviennent contre le vandalisme et offrent une sécurité optimale.

L'agriculture intelligente permet aussi de gagner un temps précieux et permet aussi de se consacrer aux tâches importantes. Les autres points qui nous ont motivés à choisir ce thème sont la modernité et la vitesse fulgurante avec laquelle cette technologie est en train d'évoluer.

I.3. Cahier de charges

Le cahier des charges d'un projet est une description complète du projet dans le langage naturel (non technique) compréhensible par toute personne ingénieur ou non ingénieur. Dans ce cahier, le client fait ressortir l'ensemble des contraintes que doit respecter son produit : l'encombrement, le temps réel, la consommation, le coût, etc....

Dans notre projet de ferme intelligente nous avons eu l'idée d'élaborer une enceinte de quatre compartiments, cette dernière reçoit des données et envoie des commandes à distance via la technologie Bluetooth.

Les quatre compartiments doivent accomplir des fonctions requises bien précises afin que notre système soit assez proche d'une vraie enceinte agricole intelligente.

Pour cela nous avons proposé le cahier de charge suivant avec des composants simples mais utiles à notre réalisation :

- Concevoir une mini-ferme en bois de bonne qualité ainsi qu'un tapis en gazon pour donner une vue d'ensemble similaire à l'environnement de la ferme.

- Commander l'éclairage de l'enceinte à distance afin d'allumer et d'éteindre la lumière à l'intérieur de l'enceinte.
- Acquérir la température via un capteur DHT11 et contrôler le système de refroidissement.
- Acquérir des informations sur la présence ou non des flammes.
- Acquérir le niveau d'eau de terre via un capteur d'humidité de sol et contrôler en conséquence l'arrosage de la ferme.

I.4. Objectifs à réaliser

Notre projet de fin d'études vise à accomplir les objectifs suivants :

- Concevoir une ferme intelligente miniaturisée à base de bois.
- Création d'une application mobile Android pour commander et contrôler notre système de ferme intelligente.
- Etudier l'aspect de commande des composants afin d'analyser les besoins fonctionnels et non fonctionnels et d'établir l'organigramme de séquence.
- Procéder au branchement de la carte Arduino Uno ainsi que les différents composants notamment les capteurs et les actionneurs.
- Editer un programme qui sera téléversé sur la carte Arduino afin de piloter les différents éléments électroniques.

I.5. Approche Philosophique/Historique

"Depuis toujours l'homme est en quête de bien être". Cette réflexion (qui rejoint la notion de besoin) peut paraître bien éloignée d'un cours de Sciences Industrielles, pourtant c'est la base de l'évolution des sciences en général, et de l'automatisation en particulier. L'homme a commencé par penser, concevoir et réaliser. Lorsqu'il a fallu multiplier le nombre d'objets fabriqués, produire en plus grand nombre, l'automatisation des tâches est alors apparue : remplacer l'homme dans des actions pénibles, délicates ou répétitives.

Citons par exemple quelques grands hommes, avec les premiers développements de l'ère industrielle au XVIIIème siècle, Watt, avec ses systèmes de régulation à vapeur, Jacquard et ses métiers à tisser automatiques... Une liste exhaustive serait bien difficile à établir !

Enfin, le développement des connaissances, et des outils mathématiques, ont conduit à un formidable essor des systèmes automatisés, et des systèmes asservis, dans la deuxième moitié du 20ème siècle. Certains se hasardent à rapprocher l'Automatique et la philosophie, observant d'étranges similitudes entre les processus propres à l'homme et l'approche technologique.

Mais au fait qu'est-ce qu'un système ? Bien difficile de répondre à une telle question ! Notre point de vue porte sur les systèmes de production et les systèmes pluri-techniques en général, nous pouvons néanmoins en donner une définition plus large [1].

I.6. Généralités sur les systèmes automatisés

Avant de présenter les fermes intelligentes ainsi que les systèmes agricoles 'Smart', nous allons d'abord essayer de comprendre qu'est-ce qu'un système automatisé, et de quoi est-il composé ainsi que ses avantages et inconvénients.

I.6.1. Système technique et automatisation

Un système : est un ensemble organisé d'éléments interagissant entre eux et avec l'extérieur, dans le but de réaliser une fonction définie.

Système technique est généralement un processus ou un dispositif industriel destiné à la production ou la transformation. Ce processus se compose d'un ensemble de machines.

L'automatisation d'un système technique consiste à assurer une commande (valeur ajoutée) en utilisant un outil technologique.

I.6.2. Système automatisé

C'est un système qui exécute toujours le même cycle de travail et qui est programmé à l'avance, sans l'intervention de l'utilisateur. L'opérateur assure la programmation, le démarrage et l'arrêt du système. Ce dernier simplifie, sécurise et rend moins pénibles les tâches répétitives et opérationnelles. Par contre, dans un système mécanique, l'utilisateur commande et contrôle l'ensemble des opérations.

De nos jours, on trouve des systèmes automatisés un peu partout. Dans l'industrie (figure 1.1), par exemple, on les utilise pour réaliser des tâches dangereuses ou particulièrement pénibles.

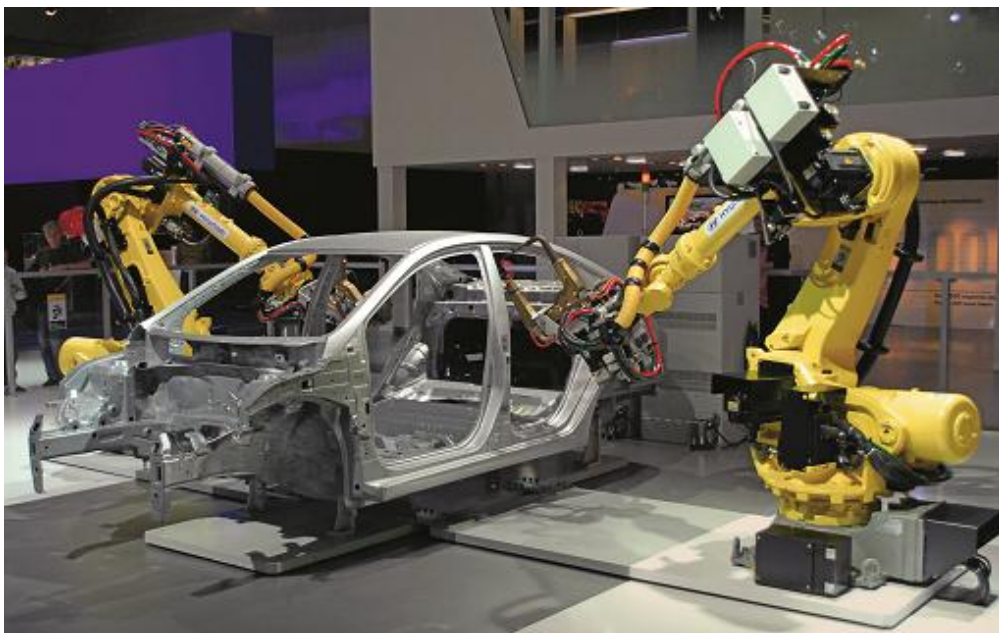


Figure 1.1 : Une série de systèmes automatisés destinés à souder la carrosserie des véhicules

I.6.3. Différentes parties d'un système automatisé

Les systèmes automatisés, utilisés dans le secteur industriel, possèdent une structure de base identique (figure 1.2).

Ils sont constitués de plusieurs parties plus ou moins complexes reliées entre elles :

- la partie opérative (PO) ;
- la partie commande (PC) ou système de contrôle/commande (SCC) ;
- la partie relation (PR) de plus en plus intégrée dans la partie commande.

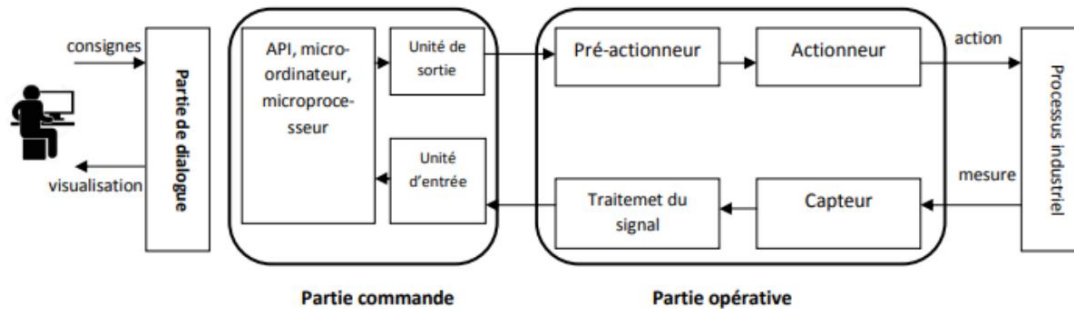


Figure 1.2. Architecture générale d'un système automatisé.

I.6.3.1 .La partie opérative

C'est la partie visible du système. Elle comporte les éléments du procédé, Il se compose des ensembles suivants :

- Pré-actionneurs** : Distributeurs, contacteurs ; qui reçoivent des ordres de la partie commande.
- Actionneurs** : Vérins, moteurs, vannes ; qui ont pour rôle d'exécuter ces ordres. Ils transforment l'énergie pneumatique (air comprimé), hydraulique (huile sous pression) ou électrique en énergie mécanique.
- Capteurs** : Les capteurs (figure 1.3) informent la partie commande de l'exécution du travail. Par exemple, on va trouver des capteurs mécaniques, pneumatiques, électriques ou magnétiques montés sur les vérins. Le rôle des capteurs (ou détecteurs) est donc de contrôler, mesurer, surveiller et informer la PC sur l'évolution du système.






				
LDR	Capteur fin course	Capteur température	Capteurs ILS	Bouton poussoir
Détecte la lumière	Détecte un appui (porte par exemple)	Mesure la température	Détecte un champ magnétique	Détecte un appui

Figure 1.3.Exemples de capteurs

I.6.3.2. Partie commande

Ce secteur de l'automatisme gère selon une suite logique le déroulement ordonné des opérations à réaliser. Il reçoit des informations en provenance des capteurs de la Partie Opérative, et les restitue vers cette même Partie Opérative en direction des pré-actionneurs et actionneurs. L'outil de description de la partie commande s'appelle le Graphe Fonctionnel de Commande Étape / Transition (GRAFCET).

I.6.3.3. Partie Relation

Sa complexité dépend de l'importance du système. Elle regroupe les différentes commandes nécessaires au bon fonctionnement du procédé, c'est à dire marche/arrêt, arrêt d'urgence, marche automatique, etc..... L'outil de description s'appelle le Guide d'Études des Modes de Marches et d'Arrêts (GEMMA).

Les outils graphiques, que sont le GRAFCET et le GEMMA, sont utilisés par les automaticiens et les techniciens de maintenance.

I.6.4. Mieux comprendre le système automatisé

Pour mieux comprendre le fonctionnement d'un système automatisé, ce dernier peut être assimilé à un être humain (figure 1.4).

- Le Cerveau est la partie commande.
- Les cinq sens sont les capteurs.
- Les Muscles sont les actionneurs.
- Les Nerfs sont les liaisons entre ses diverses parties.

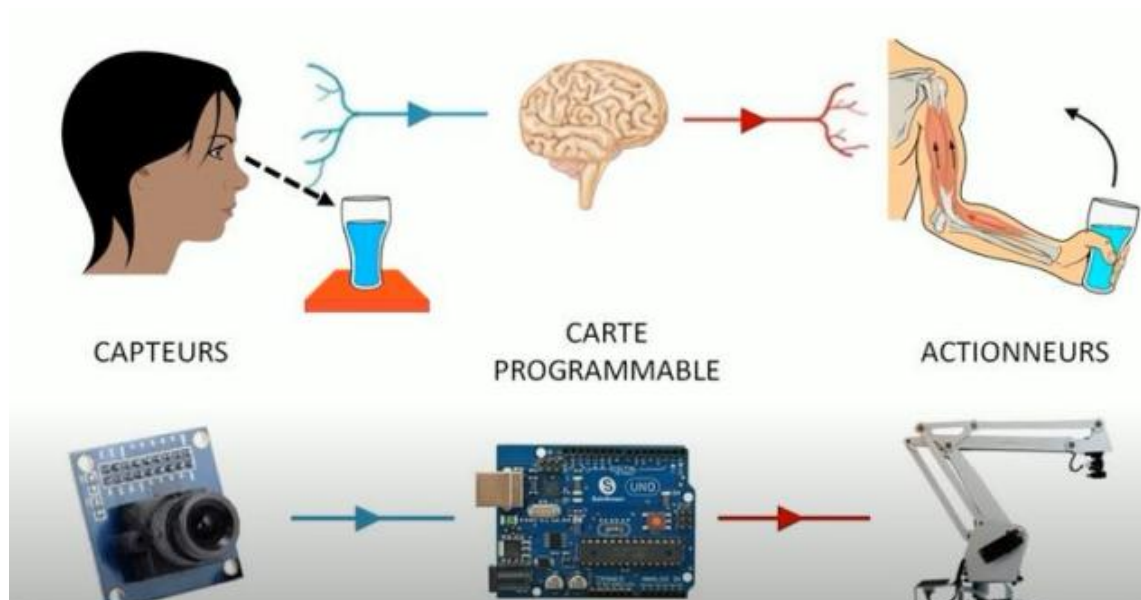


Figure 1.4. Comparaison d'un système automatisé avec le corps humain

I.6.5. Avantages et inconvénients de l'automatisation

Certaines personnes pensent que l'automatisation finira par ronger tous les emplois humains. Il s'agit d'un blog pour évaluer si l'automatisation est une bonne ou une mauvaise chose pour la génération d'aujourd'hui et les générations futures.

La préoccupation concernant l'automatisation des tâches remonte à la revue industrielle. L'idée que le travail serait automatisé a fait craindre une baisse du rythme de travail, avec des conséquences sur l'égalité et la progression humaines. Il y a cent ans, une personne moyenne travaillait environ 15 heures par semaine ; aujourd'hui, nous travaillons en moyenne 40 heures par semaine. L'automatisation s'est produite à la fois dans le secteur manufacturier et dans le secteur des services. Dans la fabrication, les robots ont été utilisés pour fabriquer plus rapidement et plus efficacement des pièces pour les marchandises. Dans les industries de services, des programmes de service client automatisés ont émergé qui ressemblent souvent à des appels automatisés.

La question de la sécurité de l'emploi n'est pas nouvelle. En fait, c'est un sujet de débat dans le monde du travail depuis des décennies. Ce qui est nouveau, c'est la capacité des ordinateurs à effectuer certaines tâches mieux et plus efficacement que les humains. Que ce soit par le développement de systèmes automatisés ou les progrès de l'intelligence artificielle et de l'apprentissage automatique ; nous continuerons à voir plus d'automatisation dans divers domaines car il deviendra possible et rentable pour les entreprises d'automatiser certaines tâches.

L'automatisation peut vous faire gagner du temps et de l'argent gratuitement. Si vous avez affaire à des gens. L'automatisation peut augmenter la productivité L'automatisation peut vous rapporter une belle augmentation si vous devez traiter avec des personnes. Inconvénients de l'automatisation Les gens ne vous aimeront pas si vous utilisez l'automatisation. Les gens ont besoin d'interagir avec les gens à un moment donné de leur vie. Si vous refaites la même chose à plusieurs reprises, l'automatisation peut en fait vous rendre moins efficace. Si vous faites la même chose encore et encore, l'automatisation peut vous rendre moins efficace. Vous êtes finalement contrôlé par la machine. Vous n'êtes plus libre de choisir vos tâches et vos projets. Êtes-vous prêt à laisser votre salaire diminuer ? Êtes-vous prêt à moins traiter avec les gens ? Il y a plein de gens qui ne le seraient pas.

I.6.5.1. Avantages de l'automatisation

- Améliorer les conditions de travail (effectuer des tâches pénibles, dangereuses et répétitives),
- Sécurité et précision,
- Réduire les coûts de fabrications (produit plus compétitif),
- Augmenter la productivité (réduire le temps de travail nécessaire à la production, donc augmenter les cadences de travail),
- Flexibilité (une machine peut s'adapter à plusieurs productions),
- Confidentialité (une machine ne peut pas parler),
- Un système automatisé peut travailler 24h sur 24h,
- Maintenir la qualité.

I.6.5.2. Inconvénients de l'automatisation

- Incidence sur l'emploi (licenciement –chômage),
- Investissement pour l'achat de machines,
- Coût de maintenance et pannes,
- Consommation d'énergie,
- Formation d'un personnel plus qualifié (technicien de maintenance, de control, etc.) [2].

I.7. L'agriculture intelligente

Dans cette partie, nous allons expliquer et détailler la notion des fermes intelligentes ainsi que leurs catégories et leurs domaines d'utilisation.

I.7.1. Définition de l'agriculture intelligente

Selon la FAO (L'Organisation pour l'alimentation et l'agriculture) [3], l'agriculture intelligente (figure 1.5) est une approche intégratrice qui s'adresse aux défis liés à la sécurité alimentaire et au changement climatique, qui vise explicitement 3 objectifs :

- Augmentation durable de la productivité et du revenu,
- Adaptation et renforcement de la résilience au changement climatique,
- Réduction et/ou suppression des émissions de gaz à effet de serre.

L'agriculture intelligente utilisant les IoT (Internet des objets) consiste à améliorer la gestion de l'eau, la gestion du sol, la surveillance des cultures, le contrôle des insecticides et des pesticides. Les capteurs des IoT sont capables de fournir aux agriculteurs des informations sur les rendements des récoltes, précipitations, infestation de parasites et la nutrition du sol. En plus de cela, les capteurs peuvent collecter des données précises qui peuvent être employées pour améliorer des techniques agricoles avec le temps.

La cartographie par satellite des parcelles permet d'affiner la connaissance des sols pour ne fertiliser que là où les apports d'engrais sont nécessaires ou de connaître le stress des plantes (manque d'eau ou de nutriments). En Europe les agriculteurs s'appuient désormais sur des stations météorologiques ou des capteurs embarqués sur les tracteurs pour suivre l'indice de végétation, les rendements, les besoins en eau, l'état de la nutrition azotée. Un institut de recherche française l'IRSTEA (Institut national de Recherche en Science et Technologie pour l'Environnement et l'Agriculture) place de grands espoirs sur les réseaux de capteurs autonomes sans fil implantés au cœur des parcelles et qui peuvent assurer en continu le suivi des cultures.

Grâce aux données collectées, les agriculteurs interviennent au bon moment et avec la dose strictement nécessaire de produit en cas d'attaque de ravageurs ou d'intempérie (grosses pluies ou sécheresses).



Figure I.5. Photo représentative de l'agriculture intelligente

I.7.2. Domaines de l'agriculture intelligente

Une ferme intelligente est un écosystème complexe qui intègre différents domaines pour tirer parti des technologies de pointe et améliorer l'efficacité globale de l'exploitation agricole. Dans cette phase, nous allons faire une description plus détaillée de ces différents domaines :

A/Surveillance et contrôle des cultures

Dans une ferme intelligente, des capteurs et des systèmes de surveillance sont utilisés pour collecter des données sur les cultures, telles que la croissance, la qualité du sol, l'humidité, la température et la lumière. Ces informations sont ensuite analysées pour fournir des conseils et des recommandations précises aux agriculteurs, leur permettant de prendre des décisions éclairées sur la gestion des cultures, l'irrigation, la fertilisation et la lutte contre les maladies et les ravageurs.

B/Gestion de l'eau

L'un des aspects clés de l'agriculture intelligente est l'optimisation de l'utilisation de l'eau. Des technologies de pointe, telles que les systèmes d'irrigation de précision, sont utilisées pour mesurer et contrôler la quantité d'eau nécessaire pour chaque culture en fonction de ses besoins spécifiques. Des capteurs de sol et de climat sont utilisés pour surveiller l'humidité du sol, la pluviométrie et la météo, permettant ainsi d'ajuster automatiquement l'irrigation pour minimiser le gaspillage et maximiser l'efficacité.

C/Élevage de précision

Dans une ferme intelligente, les technologies de suivi et de surveillance sont utilisées pour surveiller la santé, la nutrition et le comportement des animaux d'élevage. Des capteurs sont utilisés pour collecter des données sur la température, l'activité, l'alimentation et les signes

vitaux des animaux, permettant aux agriculteurs de détecter rapidement les problèmes de santé, d'optimiser les pratiques d'alimentation et de maximiser le rendement de l'élevage.

D/Automatisation des machines agricoles

Les machines agricoles automatisées, comme les robots et les drones (figure 1.6), jouent un rôle essentiel dans une ferme intelligente. Ils sont utilisés pour effectuer des tâches telles que la plantation, la récolte, la pulvérisation des cultures et le transport des produits. Ces machines peuvent être équipées de capteurs et de caméras pour collecter des données en temps réel, ce qui permet d'optimiser les opérations agricoles, de réduire les erreurs humaines et d'améliorer l'efficacité globale.



Figure 1.6. Photo illustrant l'utilisation d'un drone dans une ferme

E/Gestion des ressources

Une ferme intelligente intègre des systèmes de gestion des ressources pour optimiser l'utilisation des engrais, des pesticides et d'autres intrants agricoles. Des capteurs sont utilisés pour surveiller les niveaux de nutriments dans le sol, les taux de croissance des plantes et les prévisions météorologiques, permettant ainsi de fournir des recommandations précises sur l'application des intrants et de minimiser le gaspillage.

F/Gestion de l'énergie

La gestion de l'énergie est un aspect important de l'agriculture intelligente. Les fermes intelligentes intègrent des sources d'énergie renouvelable, telles que l'énergie solaire et éolienne, pour alimenter les opérations agricoles. Des systèmes de suivi et de contrôle sont utilisés pour optimiser la consommation d'énergie, réduire les coûts et minimiser l'empreinte carbone de l'exploitation agricole.

En intégrant ces différents domaines, une ferme intelligente vise à améliorer l'efficacité, la productivité et la durabilité de l'agriculture, tout en réduisant l'impact sur l'environnement et en assurant la sécurité alimentaire à long terme.

I.7.3. Avantages et inconvénients de l'agriculture intelligente

L'agriculture intelligente présente de nombreux avantages potentiels qui peuvent contribuer à améliorer l'efficacité, la durabilité et la rentabilité de l'agriculture, et ceci malgré quelques désavantages. Il est important de prendre en compte ces avantages et ces inconvénients lors de la mise en œuvre de l'agriculture intelligente, en évaluant soigneusement les besoins et les contraintes spécifiques de chaque exploitation agricole.

En effet, dans cette partie nous allons mettre l'accent sur les principaux avantages et inconvénients qu'une enceinte agricole intelligente peut engendrer.

I.7.3.1. Avantages de l'agriculture intelligente

A/Augmentation de la productivité

L'agriculture intelligente permet d'optimiser les rendements en utilisant des systèmes de surveillance et de gestion précis pour chaque parcelle de terrain, ce qui peut conduire à une augmentation significative de la productivité agricole.

B/Gestion efficace des ressources

Grâce à des capteurs et à des technologies de télédétection, l'agriculture intelligente permet une gestion précise de l'eau, des engrais et des pesticides, réduisant ainsi le gaspillage et l'impact environnemental.

C/Prévention des maladies et des ravageurs

Les systèmes de surveillance automatisés aident à détecter rapidement les signes de maladies des plantes et d'infestation par des ravageurs, ce qui permet une intervention précoce et ciblée pour minimiser les dommages aux cultures.

D/Précision de la pulvérisation

L'utilisation de drones et d'équipements de pulvérisation automatisés permet une application précise des produits chimiques sur les cultures, réduisant ainsi les coûts et les effets néfastes sur l'environnement.

E/Prise de décision basée sur les données

L'agriculture intelligente utilise des données recueillies en temps réel sur le terrain pour aider les agriculteurs à prendre des décisions éclairées, améliorant ainsi la planification des cultures, la gestion des ressources et la prise en compte des facteurs environnementaux.

I.7.3.2. Inconvénients de l'agriculture intelligente

A/Coût initial élevé

La mise en place de systèmes d'agriculture intelligente peut être coûteuse, nécessitant des investissements importants dans l'infrastructure, les capteurs, les équipements et les logiciels, ce qui peut limiter l'accessibilité pour de nombreux agriculteurs.

B/Compétences techniques requises

L'utilisation de technologies avancées nécessite des compétences techniques pour l'installation, la configuration et la maintenance des équipements. Certains agriculteurs peuvent avoir besoin de formation supplémentaire pour tirer pleinement parti de l'agriculture intelligente.

C/Dépendance aux données et à la connectivité

L'agriculture intelligente dépend de la collecte et de l'analyse des données en temps réel. Une connectivité fiable à Internet et des infrastructures de communication adéquates sont nécessaires, ce qui peut poser problème dans les zones rurales éloignées ou les pays en développement.

D/Protection des données et vie privée

La collecte et l'utilisation des données agricoles peuvent soulever des questions de confidentialité et de protection des données personnelles. Il est essentiel de mettre en place des politiques et des réglementations appropriées pour garantir la sécurité des informations sensibles.

E/Dépendance technologique

L'agriculture intelligente repose sur des technologies avancées qui peuvent évoluer rapidement. Cela signifie que les agriculteurs doivent être prêts à faire face aux évolutions technologiques constantes et à la nécessité de mettre à jour leurs équipements et leurs compétences pour rester à jour.

I.8. Conclusion

L'agriculture intelligente, est un marché en plein expansion dans lequel les innovations ne cessent de croître. Cette nouvelle technologie est devenue indispensable dans la vie quotidienne des agriculteurs.

Dans ce chapitre, nous avons donné un aperçu général sur l'agriculture intelligente, tout en justifiant les raisons qui nous ont motivés à choisir ce thème, et cela en mentionnant les multiples avantages des fermes 'Smart' et en détaillant les différents domaines de cette technologie.

Nous avons aussi expliqué les notions de bases d'un système automatisé, tout en citant les avantages de l'automatisation. Et nous avons aussi établi un cahier de charges qui sera notre feuille de route pour réaliser notre projet

Le chapitre suivant sera consacré à détailler la conception d'une ferme intelligente en évoquant les composants et logiciels utilisés pour réaliser notre projet.



Chapitre II :
**Partie Matériel Et
Logiciel**

II.1. Introduction

L'objectif de l'électronique est de créer des systèmes de plus en plus puissants, miniaturisés et autonomes, autrement appelés systèmes électroniques embarqués, tout en minimisant le coût d'étude et de production qui oblige la mise en œuvre d'autres technologies pour la simplification des schémas électroniques, ce qui facilite le traitement des systèmes plus complexe avec un bon rendement, performance et cout réduit.

Un système embarqué (figure 2.1) présente une intégration entre deux parties, logicielle et matérielle, qui sont conçues conjointement pour répondre à des fonctionnalités spécifiques. Ces fonctionnalités sont, pour la plus part des cas, des applications critiques.

Dans ce chapitre nous allons développer la notion de l'électronique utilisée dans notre projet et qui a la capacité d'unir la puissance de la programmation à la puissance de l'électronique.

Pour cela nous allons présenter les différents matériels et logiciels nécessaires pour la réalisation de notre projet de ferme intelligente.



Figure : 2.1-Photo représentative des systèmes électroniques embarqués

II.2.Partie Matériels

Dans cette partie, nous allons évoquer la carte programmable choisie, ainsi que les différents composants électroniques utilisés pour réaliser notre système de ferme intelligente, ainsi que l'unité d'alimentation qui est indispensable pour alimenter notre système.

II.2.1. Choix de la carte programmable

Pour nos projets électroniques nous avons souvent besoin d'une carte programmable parmi de centaines cartes dans le marché, il peut être difficile des fois de choisir sa carte, mais le projet et le cahier des charges doivent être les seuls guides sur le choix. Donc on peut faire

une comparaison entre les principales plateformes de développement en tenant compte de leurs avantages, inconvénients, fonctionnalités, complexité ainsi que leurs coûts.

Grâce à leur grande variété et leur adaptation à certaines applications spécifiques, on a constaté que les cartes Arduino sont les microcontrôleurs les mieux adaptés à notre système de distributeur automatique, puisqu'elles sont suffisantes et leur rapport qualité-prix est abordable.

II.2.2. Justification du choix de l'Arduino

Le choix de l'Arduino est justifié par plusieurs paramètres détaillés dans le tableau ci-dessous :

Open Source et Open Hardware	Possibilité d'utilisation du schéma d'origine et le modifier ensuite pour produire la carte et la vendre sans avoir à payer des droits d'auteur.
Tarifs extrêmement abordables	Le schéma étant en libre accès, différentes entreprises produisent les cartes et les exposent au marché par un coût réduit
Echanges nombreux au sein de la communauté Arduino	Il existe maints forums et documentations consultables en ligne afin de pouvoir surmonter les difficultés rencontrées.
Multiplateforme	Afin de programmer une carte Arduino, il faut la connecter à votre ordinateur puis utiliser le logiciel Arduino (IDE) vous permettant de programmer votre carte. L'IDE est disponible sur Windows, Mac OSX et Linux. Contrairement à d'autres microcontrôleurs.
Clarté de l'environnement de programmation	Conçu pour des débutant, Arduino est très instinctif et vous donnera le goût à la programmation, il est d'ailleurs très facile de débiter sur Arduino.

Tableau 2-1- Choix de la carte Arduino

II.2.3. Définition de l'Arduino

Arduino (figure 2.2) est une plate-forme de prototypage open-source d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation. [4]



Figure : 2.2-Logo d'Arduino. [arduino.cc]

II.2.4. Définition d'une carte Arduino

Le module Arduino est un circuit imprimé en matériel libre (plateforme de contrôle) dont les plans de la carte elle-même sont publiés en licence libre. La carte Arduino est équipée essentiellement d'un microcontrôleur, d'un port USB ainsi que des connecteurs d'entrées/sortie (plus ou moins nombreux selon les modèles).[5]

Le microcontrôleur programmé peut analyser et produire des signaux électriques de manière à effectuer et piloter des tâches très diverses.

La carte Arduino repose sur deux parties, le "Hardware" et le "Software". Le "Hardware" est la carte électronique programmable, quant au "Software" c'est l'interface de programmation avec un langage spécifique basé sur les langages C et C++.

La carte Arduino est utilisé dans beaucoup d'applications comme l'électrotechnique industrielle et embarquée (la domotique, le pilotage d'un robot, commande des moteurs et faire des jeux de lumières, communiquer avec l'ordinateur, commander des appareils mobiles).

II.2.5.Arduino Uno

Nous avons opté pour une carte Arduino Uno .Cette carte sera au cœur de notre projet (figure II.3).

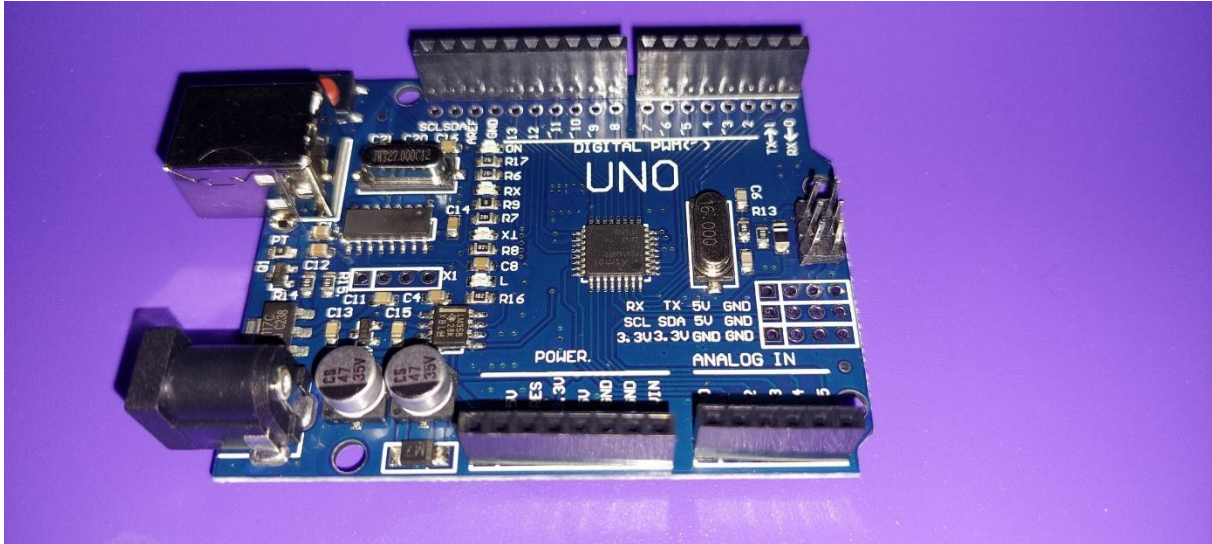


Figure : 2.3-Photo réelle de la carte Arduino UNO

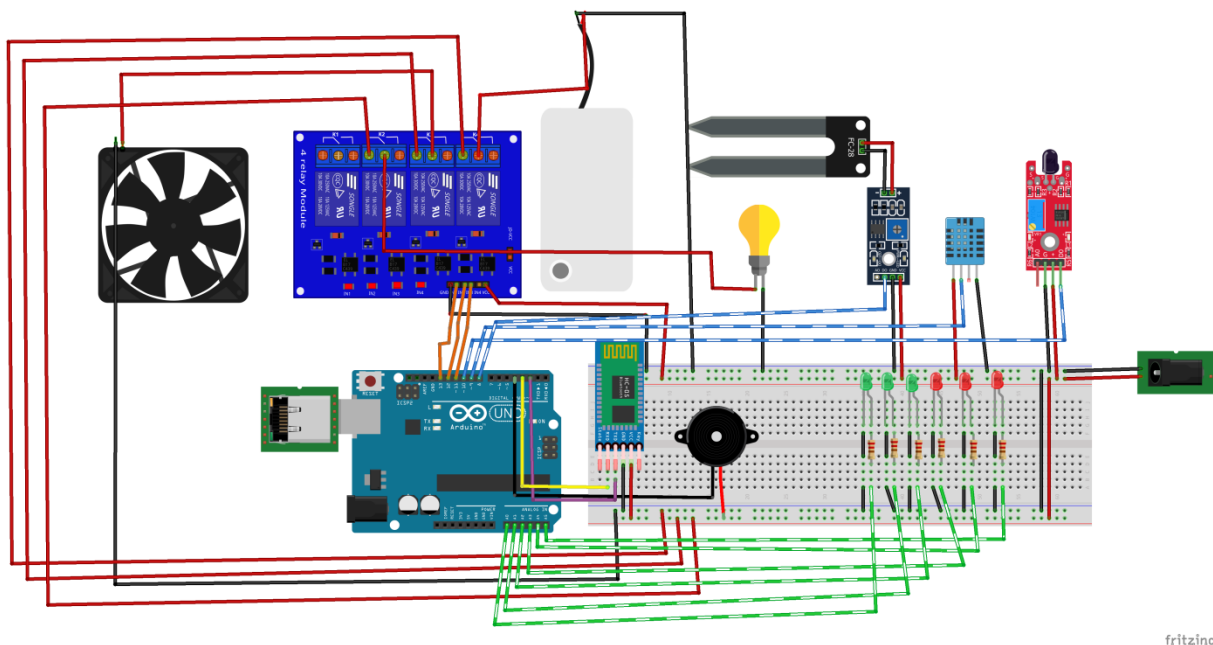


Figure : 2.3.(a) –Schéma électrique générale de montage

II.2.5.1. Choix de l'Arduino Uno

Comme on l'a déjà souligné précédemment, La carte Arduino est la base de la programmation avec microcontrôleur .Et comme on l'a constaté sur le tableau "les gammes de la carte Arduino" ; Il existe de nombreuses versions de cette carte, et il n'est pas toujours facile de choisir celle qui correspond le mieux pour notre réalisation.

Pour cela on a récolté le maximum d'informations de chaque carte et procédé ensuite a une étude comparative entre eux.

Cette étude nous a finalement orientés vers la carte Arduino Uno. Les raisons de ce choix sont détaillées dans le tableau ci-dessous.

Le cout	C'est une carte qui est relativement peu coûteuses comparativement aux autres plates-formes
Connectivité USB	l'Uno est facilement raccordable À notre ordinateur via un câble USB A/B classique, tandis que d'autres cartes telles que la Micro nécessitera un câble Micro-USB.
Disponibilité	Contrairement aux autres cartes, la carte Arduino Uno est disponible en plusieurs versions dans le marché Algérien, on peut la trouver dans les magasins de l'électronique ainsi que chez les amateurs de l'IOT.
Nombre d'entrées/sorties	La Uno est une carte composée de bien plus d'entrées/sorties que la majorité des autres cartes (6 analogiques et 14 digitales pouvant fournir jusqu'à 20mA) soit 20 I/O (au total).
Connectivité avec les Shields	Contrairement à la majorité des cartes, Arduino Uno a la possibilité de se connecter aux Shields, qui sont des cartes électronique supplémentaires comme les afficheurs graphiques en couleur, GPS,..., etc.).

Tableau 2-2- Choix de la carte Arduino Uno

II.2.6.2. Spécifications techniques

Voici un tableau présentant les principales caractéristiques de la carte Arduino Uno.

Microcontrôleur	ATmega328P
Tension de fonctionnement	5V
Tension d'entrée (recommandée)	7-12 V
Broches d'E / S numériques	14 (D0-D13)
Broches d'E / S numériques PWM	6
Broches d'entrée analogique	6
Courant CC par broche d'E / S	20 mA
Courant CC pour broche 3,3 V	50 mA
Mémoire flash	32 Ko
SRAM	2 Ko
EEPROM	1 Ko
Vitesse de l'horloge	16 MHz
LED_BUILTIN	D13
Longueur	68,6 mm
Largeur	53,4 mm
Poids	25 g

Tableau 2.3- Spécifications techniques de la carte Arduino

II.3. Alimentation et branchement

II.3.1. Transformateur

Afin d'alimenter notre système de ferme intelligente, nous avons choisi des modules et des capteurs dont la valeur maximale de la tension est de 5V. Pour cela, nous avons utilisé un transformateur de 5 Volts (figure 2.4 (a)) et qui fournit un courant de 2 Ampères, ce dernier sera assez suffisant pour la puissance du système. La photo ci-dessous montre le transformateur utilisé pour notre projet.



Figure : 2.4(a)-Photo réelle du transformateur

Le transformateur est muni d'une sortie sous forme d'une prise jack male (figure 2.4 (b)). Afin de pouvoir faire le câblage nous avons eu l'idée de la raccorder avec une prise jack femelle pour ensuite dériver deux fils 5v et GND.



Figure : 2.4(b)-Photos réelles de la prise jack utilisée

II.3.2. Partie câblage

Pour la partie câblage (figure 2.4 (c)) nous avons décidé de raccorder notre système avec une plaque d'essai « Breadboard », et cela afin de faciliter la distribution de l'énergie ainsi que pour de placer quelques composants directement sur la planche.

Quant, aux fils, nous avons utilisés des jumpers de 30 cm de type male-male et male-femelle, avec différents couleurs pour pouvoir différencier le câblage de chaque composant.

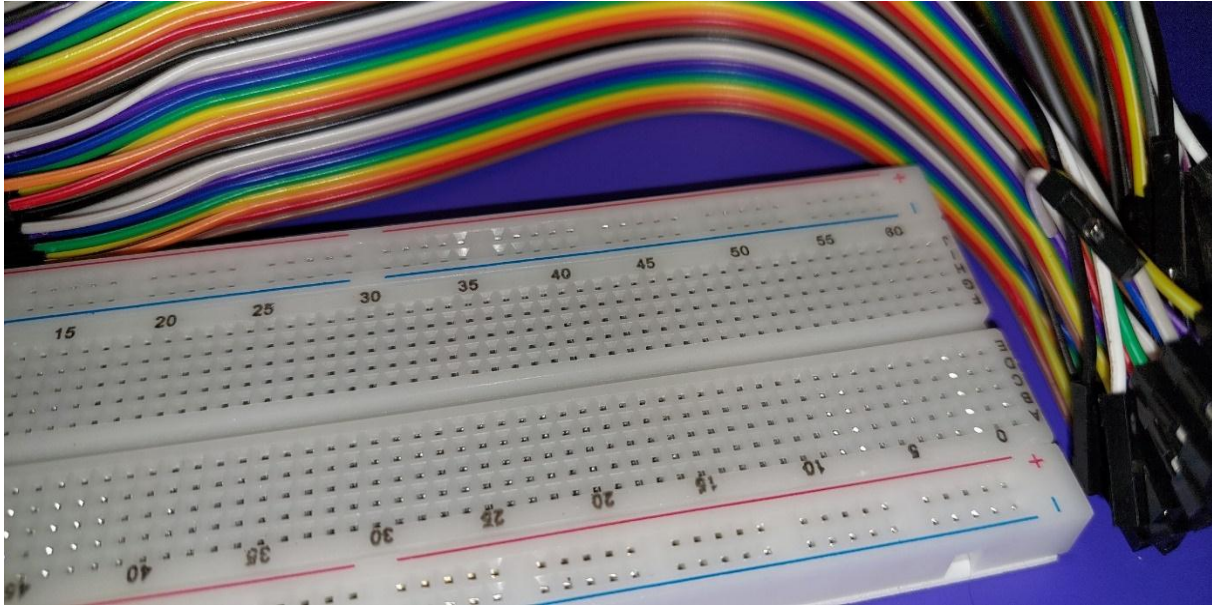


Figure : 2.4(c)-Photo réelle de la plaque d'essai et des fils

II.4. Capteurs

II.4.1. Définition d'un capteur

Un capteur réalise l'acquisition d'une grandeur physique (température, luminosité, présence, distance...) qu'il transforme en un signal de sortie logique (grandeur électrique), analogique ou numérique afin qu'il puisse être traité par la partie commande du système (actionneurs) [6].

II.4.2. Capteurs utilisés

Dans notre réalisation de ferme intelligente nous avons eu besoin de trois capteurs. Leurs présentations, leurs caractéristiques ainsi que leurs branchements avec la carte Arduino UNO sont détaillés ci-dessous.

II.4.2.1. DHT11

- **Présentation**

C'est un capteur de Température et d'Humidité très apprécié pour sa simplicité de mise en œuvre et son coût peu élevé. Il ne requiert qu'une résistance de tirage et une alimentation 3V ou 5V pour fonctionner. Sa programmation est facile à l'aide des bibliothèques Arduino

Le module numérique DHT11 (figure 2.5(a)) délivre et reçoit un signal digital sur une entrée/sortie série unique. Ses deux capteurs analogiques sont une résistance qui détermine le taux d'humidité et une thermo-résistance de type NTC (Negative Temperature Coefficient) afin de mesurer la température.

La calibration du module est faite en usine et les paramètres sont sauvés dans une mémoire OTP (One Time Programming, comme celle vue en TD de PeiP1 sur les diodes). Le lien entre l'arduino et les capteurs est assuré par un microcontrôleur 8 bits. Une communication avec le module s'effectue sur 40 bits et dure typiquement 4ms.

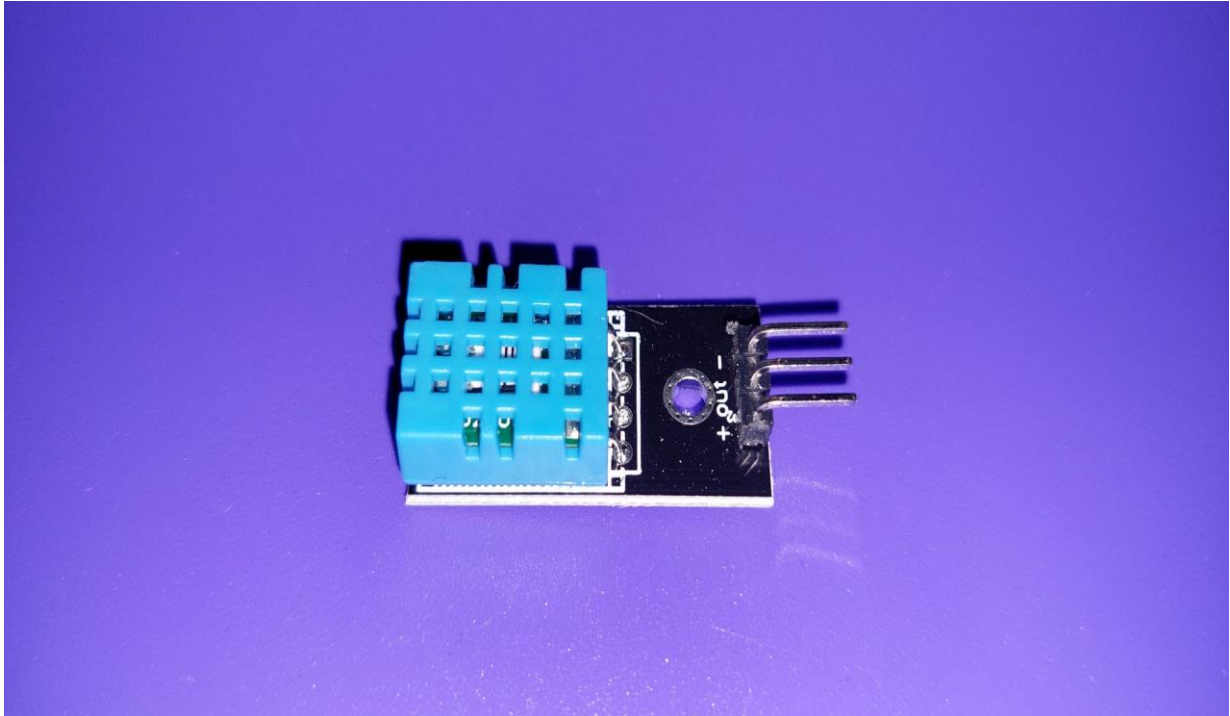


Figure : 2.5(a)-Photo réelle du DHT11

- **Caractéristiques**

Les spécifications techniques du DHT 11 sont citées dans le tableau ci-dessous.

Tension de fonctionnement	3.3V-5V
Plage de Mesure: Humidité:	20-90%
Plage de Mesure: Température:	0-50°C
Précision: Humidité	1%
Précision: Température	0.1°C
Période de mesure	2 secondes
Type	Résistance Polymère
Dimensions	12x15.5x5.5
Consommation d'énergie	Très faible

Tableau 2-4- Spécifications techniques du DHT11

- **Branchement**

Le câblage du DHT11 avec la carte Arduino Uno est illustré dans la photo ci-dessous (figure 2.5 (b)).

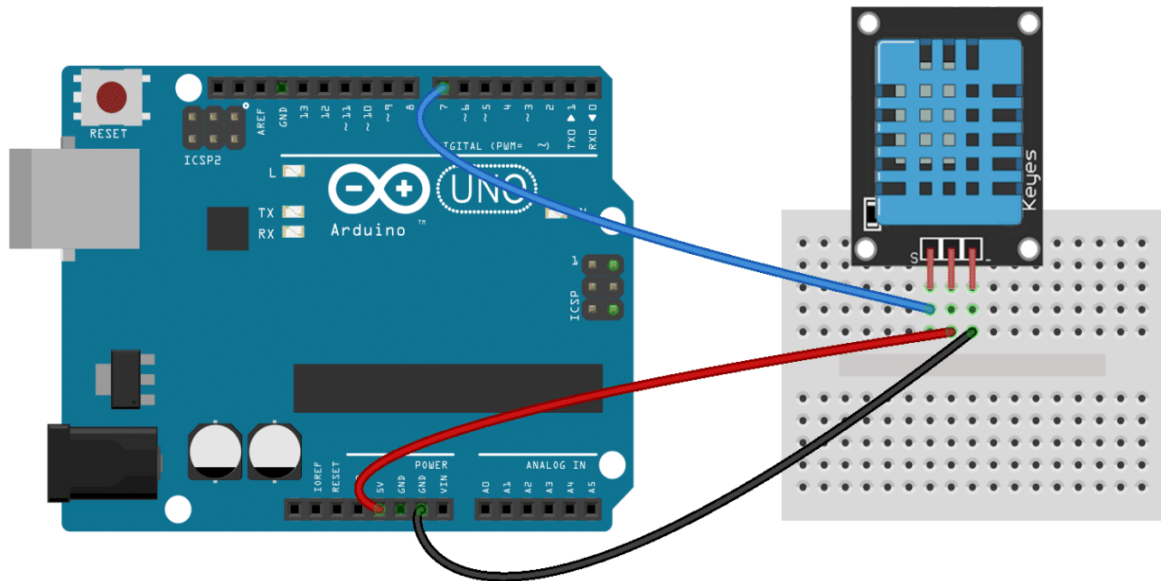


Figure : 2.5(b)-Photo du branchement du DHT11 avec Arduino

II.4.2.2. Capteur de flamme

- **Présentation**

Le détecteur de flamme (figure 2.5(c)) détecte toute élévation de température ou présence de produits issus d'une combustion. Les flammes produisent des rayonnements caractérisés par une fréquence de scintillement plus ou moins intense dans des bandes spectrales spécifiques. Le principe du détecteur de flamme est de répondre aux rayonnements électromagnétiques émis par une flamme, en les distinguant des rayonnements interférents présents dans l'environnement d'utilisation. Les détecteurs de flamme optiques sont constitués de capteurs infrarouges pour détecter ces rayonnements.

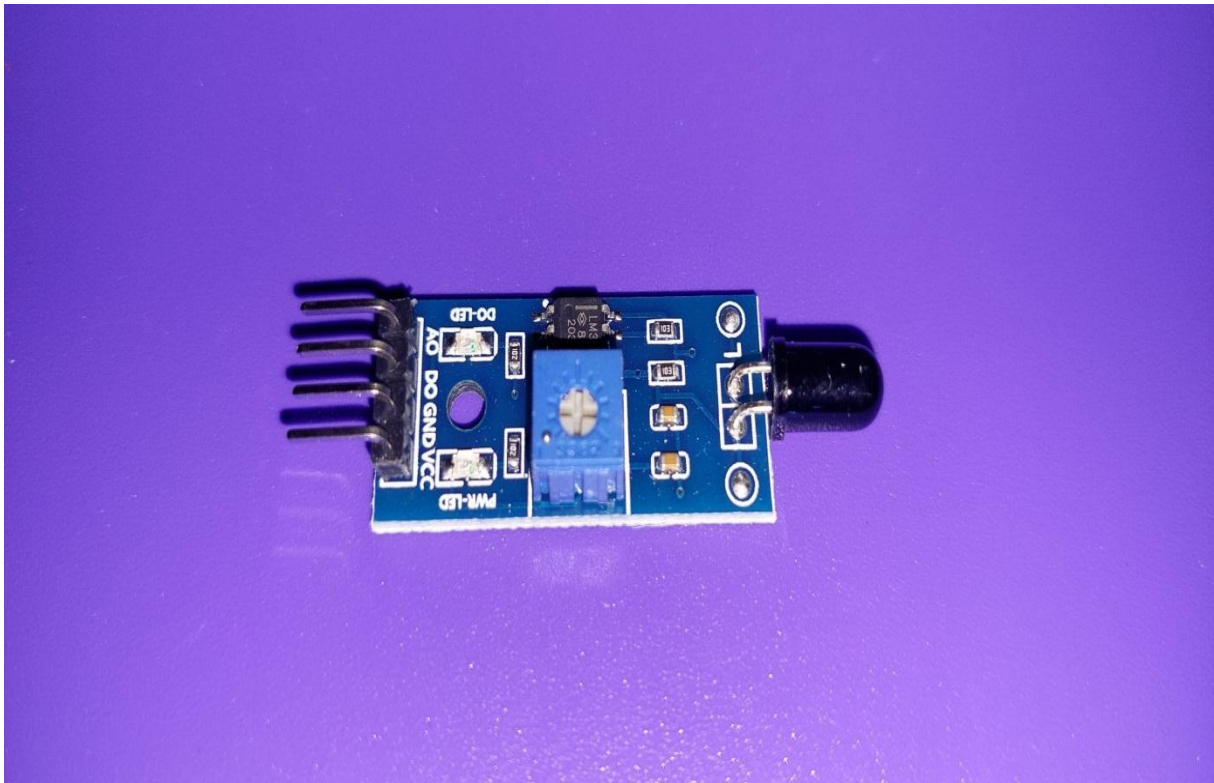


Figure : 2.5(c)-Photo réelle du capteur de flamme

Il existe trois catégories d'appareils pour détecter une flamme :

- Les détecteurs dotés de capteurs Infra-Rouge (IR)
- Les détecteurs composés de capteurs Ultra-Violet (UV)
- Et les détecteurs combinant IR et UV (en général, ils sont constitués de deux capteurs IR et d'un capteur UV)

- **Caractéristiques**

Les spécifications techniques du capteur de température sont citées dans le tableau ci-dessous.

Tension de fonctionnement	3.3-5V
Angle de détection	60 degrés
Sorties	Analogique et numérique
Sensibilité	Réglable
Matériel	PCB
Dimension	3,5 x 1,5 x 1,2 cm
Poids	5 g

Tableau 2-5- Spécifications techniques du capteur de flamme

- **Branchement**

Le câblage du capteur de flamme avec la carte Arduino Uno est illustré dans la photo ci-dessous (figure 2.5(d)).

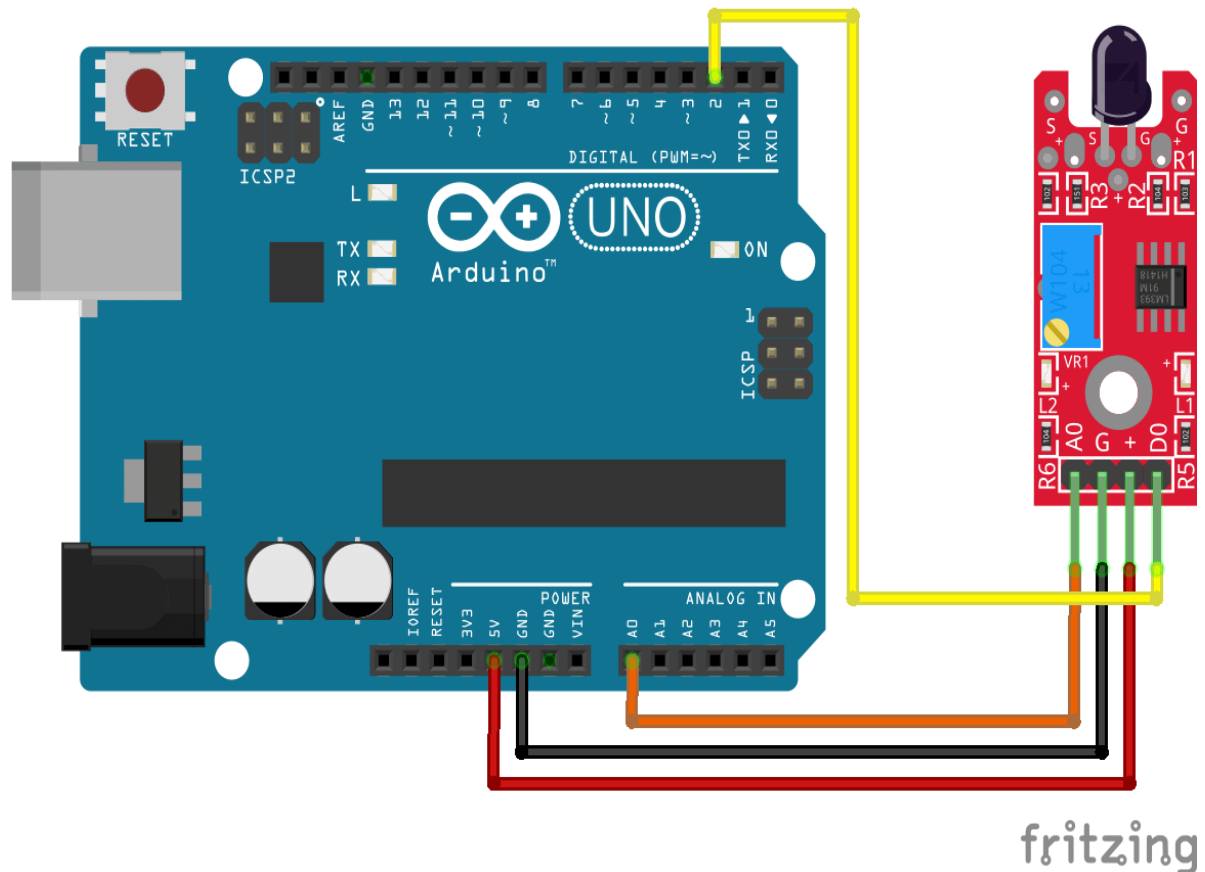


Figure : 2.5(d)-Photo du branchement du capteur de flamme avec Arduino

II.4.2.3. Capteur humidité de sol

- **Présentation**

Ce module de capteur (figure 2.5(e)) est utilisé pour détecter l'humidité du sol, Il mesure le contenu volumétrique de l'eau dans le sol et nous donne le niveau d'humidité comme sortie.

Le module dispose de sorties numériques et analogiques et d'un potentiomètre pour ajuster le niveau du seuil.

Ce module de détection d'humidité est composé d'un capteur d'humidité, de résistances, d'un condensateur, d'un potentiomètre, d'un comparateur LM393 IC, d'une LED d'alimentation et d'état dans un circuit intégré. [7]

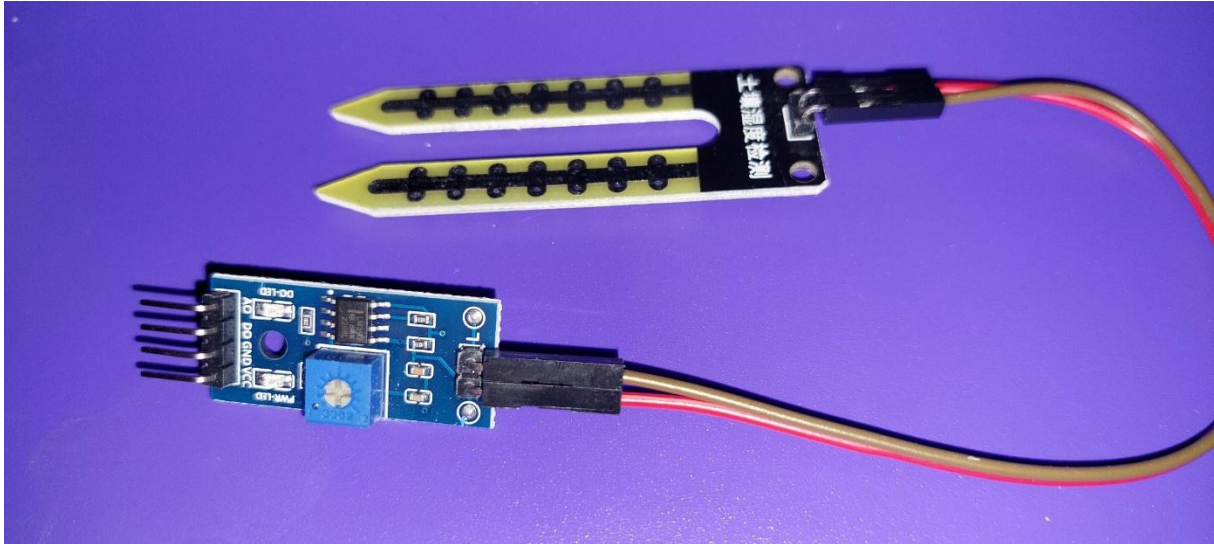


Figure : 2.5(e)-Photo réelle du capteur d'humidité de sol

- **Caractéristiques**

Les spécifications techniques du capteur d'humidité de sol sont citées dans le tableau ci-dessous.

Tension de fonctionnement	3.3V-5V
Consommation courant	35 mA
Sorties	Analogique et numérique
Taille du module	16 × 30 mm
Taille de la sonde	20 × 60 mm
Type du comparateur	LM293
Réglage	Potentiomètre
Consommation d'énergie	Très faible

Tableau 2-6- Spécifications techniques du capteur d'humidité de sol

- **Branchement**

Le câblage du capteur de l'humidité de sol avec la carte Arduino Uno est illustré dans la photo ci-dessous (figure 2.5(f)).

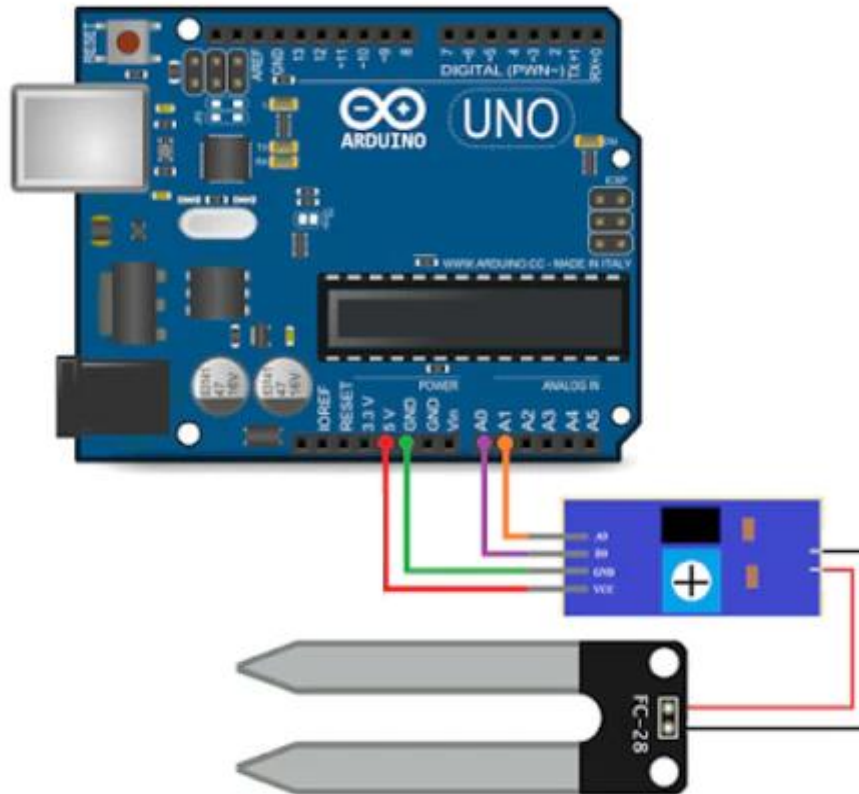


Figure : 2.5(f)-Photo du branchement du capteur d'humidité de sol avec Arduino

II.5. Pré-actionneurs

II.5.1. Définition d'un pré-actionneur

Un pré-actionneur est un composant qui délivre et/ou module la puissance utile aux actionneurs sur ordre de l'unité de commande (Arduino). Délivrer signifie « tout ou rien », on laisse passer la puissance ou pas.

II.5.2. Pré-actionneur utilisé

II.5.2.1. Relais 4 canaux

- **Présentation**

Le module relais 4 canaux Arduino (figure 2.6(a)) est un dispositif électromécanique qui sert à fermer et à ouvrir un circuit électrique au moyen d'un électroaimant.

Le principe de fonctionnement du relais de puissance est très simple. Lorsque la tension de commande est appliquée à la bobine électromagnétique, un champ électromagnétique est généré dans la bobine, qui attire le pied métallique et les contacts de la charge électrique sont fermés.

Ce module permet ainsi de commander indirectement les actionneurs (moteur, pompe, ventilateur...) en laissant ou pas passer la puissance avec ordre de l'Arduino.

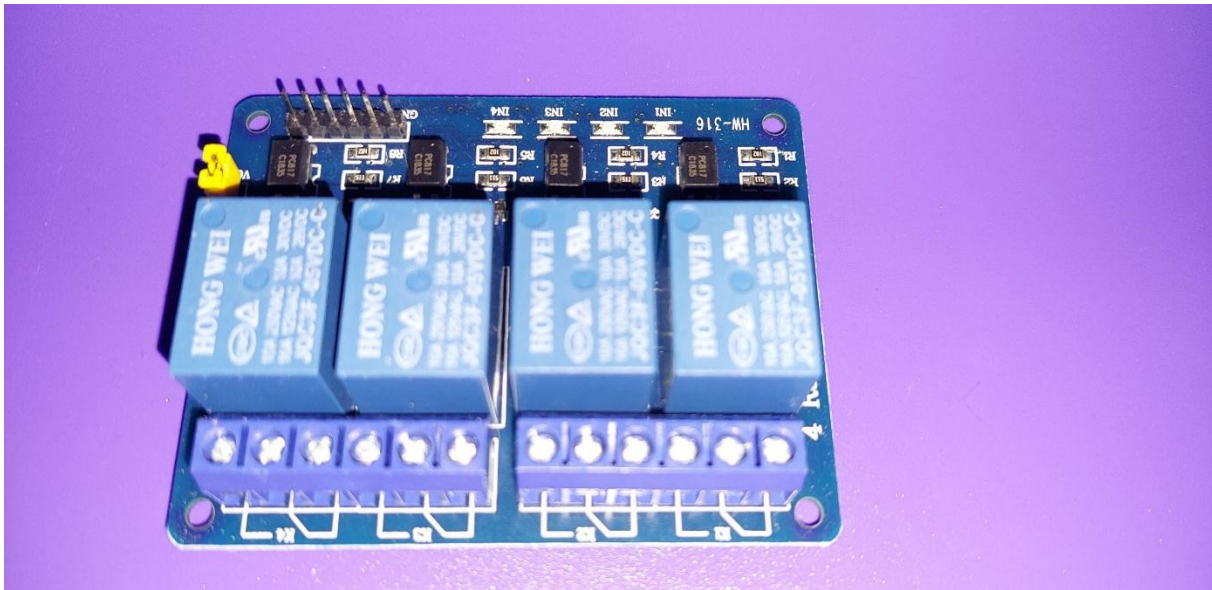


Figure : 2.6((a)-Photo réelle du module relais 4 canaux

- **Caractéristiques**

Les spécifications techniques du module relais sont citées dans le tableau ci-dessous.

Tension de fonctionnement	5V
Nombre d'entrées	4 canaux
Alimentation	GND-VCC
Tension maximale	220/230V CA 30V CC
Indication	LED
Matière	PCB
Dimension	7.7*5.3*1.7 cm
Poids	61 g

Tableau 2-7-Spécifications techniques du module relais

- **Branchement**

Le câblage du module relais 4 canaux avec la carte Arduino Uno est illustré dans la photo ci-dessous (figure 2.6(b)).

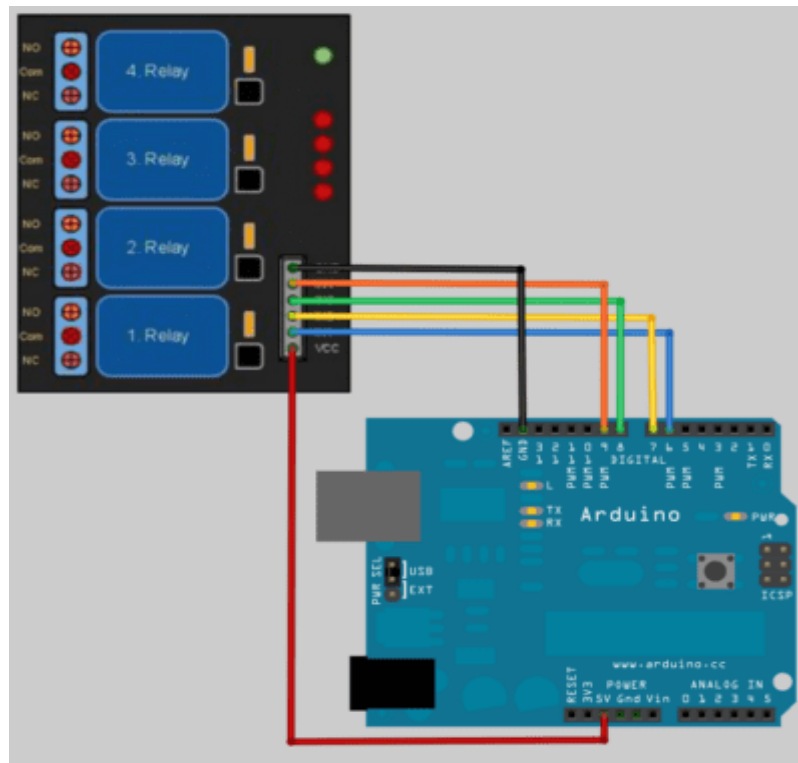


Figure : 2.6 ((b)-Photo du branchement du module relais avec Arduino

II.6. Actionneurs

II.6.1. Définition d'un actionneur

Dans une machine, un actionneur est un objet qui transforme l'énergie qui lui est fournie en un phénomène physique qui fournit un travail, modifie le comportement ou l'état d'un système.

Les actionneurs servent à actionner et sont à l'origine d'une action du système domotique, automatique (un moteur pour fermer une porte, une lampe pour avertir, ...). Dans les définitions de l'automatisme, l'actionneur appartient à la partie opérative d'un système automatisé.

II.6.2. Actionneurs utilisés

II.6.2.1. Pompe

- **Présentation**

La pompe à eau à débit réglable (figure 2.7(a)) est conçue entièrement submersible pour l'eau douce et l'eau salée.

Cette pompe peut être utilisée pour diverses applications avec la carte Arduino telle que notre projet de ferme intelligente afin d'arroser les plantes lorsque l'humidité de sol descend sous un certain seuil prédéfini à l'avance.

La pompe est facile à installer et doit être muni d'un tuyau pour transférer le liquide de l'entrée vers la sortie.



Figure : 2.7(a)-Photo réelle de la mini pompe submersible

- **Caractéristiques**

Les spécifications techniques de la pompe sont citées dans le tableau ci-dessous.

Tension de fonctionnement	3-6V
Courant	100-200 mA
Maximum lift	30-80 Cm
Débit	1.2-1.6 L par minute
Diamètre intérieur d'entrée d'eau	5 mm
Diamètre intérieur de sortie d'eau	4.5 mm
Diamètre extérieur	7.5 mm
Polarité	VCC-GND

Tableau 2-8-Spécifications techniques de la pompe

II.6.2.2. Ventilateur

- **Présentation**

Un ventilateur de 5V (figure 2.7(b)) est un dispositif de refroidissement électrique qui fonctionne à une tension de 5 volts. Il est souvent utilisé dans les projets électroniques pour refroidir des appareils ou des systèmes qui produisent de la chaleur.

Il est aussi utilisé dans les ordinateurs afin de dissiper la chaleur de l'unité de commande et éviter l'endommagement des microcontrôleurs.

Dans notre cas nous avons utilisé ce module pour simuler un système de refroidissement de la ferme intelligente lorsque la température générale dépasse un certain niveau.

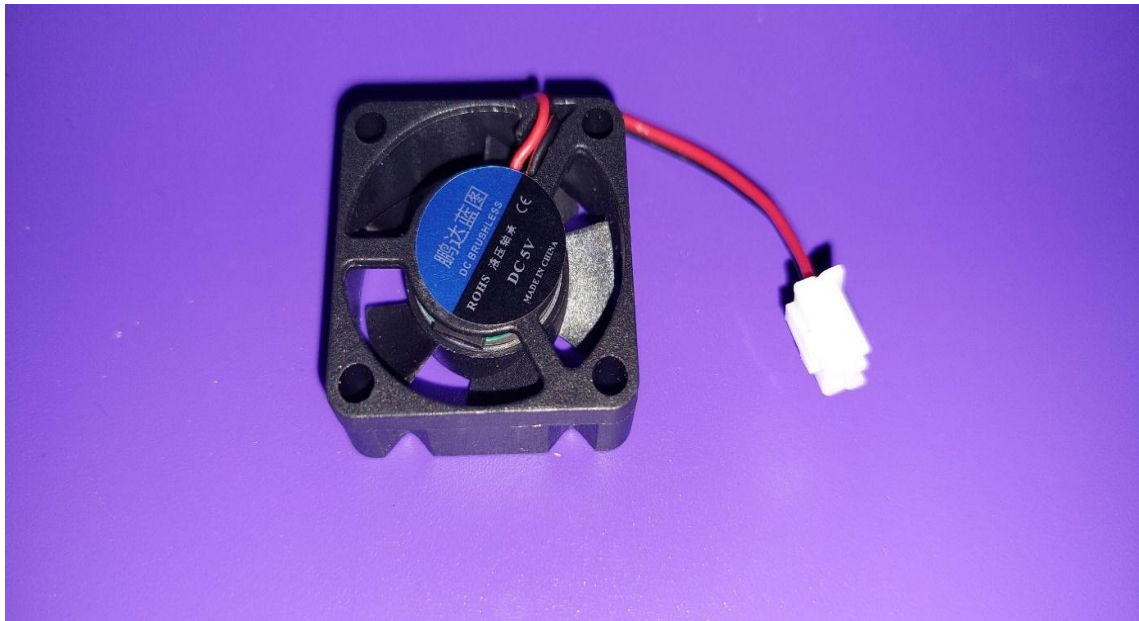


Figure : 2.7(b)-Photo réelle du ventilateur

- **Caractéristiques**

Les spécifications techniques du ventilateur sont citées dans le tableau ci-dessous.

Tension de fonctionnement	5V
Dimensions	25 x 25 x 7mm
Poids	6 g
Visserie	Vis M3
Fixation	Ecrou
Application principale	Refroidissement

Tableau 2-9-Spécifications techniques du ventilateur

II.6.2.3. Eclairage

- **Présentation**

Pour simuler le système d'éclairage de la ferme intelligente nous avons une mini lampe fonctionnant avec une tension de 3.8V.

Cette lampe (figure 2.7(c)) a été raccordée avec une petite douille pour faciliter le branchement et donner un côté esthétique au branchement.

Sachant que notre alimentation est de 5V, nous avons utilisé la broche 3.3V de l'arduino pour alimenter cette mini lampe et ainsi éviter de l'endommager.



Figure : 2.7(c)-Photo de la mini lampe utilisée

II.6.2.4. Branchement de l'actionneur avec le relais

Après avoir présenté les trois actionneurs et détailler leurs spécifications techniques. Nous allons à présent voir comment on branche un actionneur avec une carte Arduino.

En effet, il est quasiment interdit de brancher directement un actionneur (pompe, moteur, ventilateur....) directement avec le port de la carte Arduino. C'est pour cela qu'on utilise un module relais (pré-actionneur) qui permet de commander l'allumage ou pas d'un actionneur.

Pour brancher ce dernier avec un relais, il faut premièrement alimenter le relais avec une tension stable de 5V via le VCC et le GND du relais.

Ensuite le fil rouge de l'actionneur doit être couplé avec la broche "COM". Quant au fil noir il se connecte avec la masse.

La broche NO doit être connectée avec l'alimentation de l'actionneur en question (soit 5V pour la pompe et le ventilateur, et 3.3V pour la lampe).

Dans l'image ci-dessous (figure 2.7(d)) nous avons illustré ce texte avec un seul relais, mais le même principe s'applique pour un module relais de 4 canaux comme le montre.

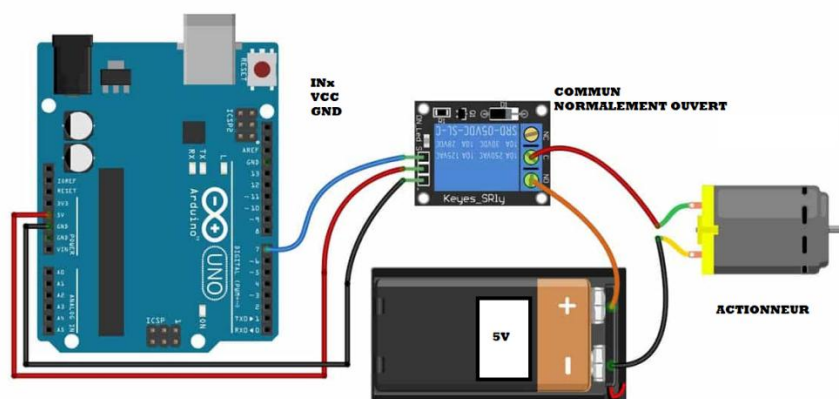


Figure : 2.7(d)-Photo du câblage d'un actionneur avec Arduino via relais

II.7. Les autres composants

II.7.1. Les Leds

Les leds (figure 2.8(a)) sont des petites lampes fonctionnant avec une tension faible de 3.3V, ce qui nous oblige à les brancher avec des résistances d'au moins 220 ohms afin de baisser les 5V de notre alimentation.

Pour notre projet nous avons eu l'idée d'utiliser des Leds afin d'indiquer l'état de la température, de l'humidité de sol ainsi que de la présence ou non des flammes.

Si la Led verte est allumée cela veut dire que tout fonctionne correctement, sinon cela démontre que les seuils de tolérance ont été atteints.

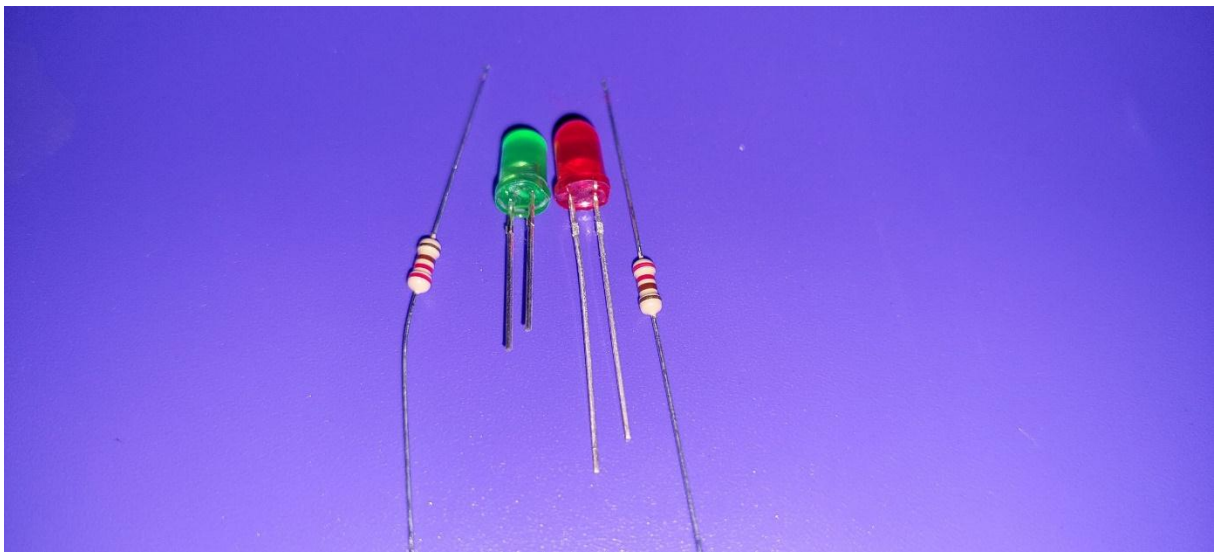


Figure : 2.8(a)-Photo réelle des leds et leurs résistances

II.7.2. Buzzer

Le buzzer (figure 2.8(b)) est un composant constitué essentiellement d'une lamelle réagissant à l'effet piézoélectrique. La piézoélectricité est la propriété que possèdent certains matériaux de se déformer lorsqu'ils sont soumis à un champ électrique.

Ce module se connecte avec l'Arduino et une broche au 5V. Nous l'avons utilisé afin de nous alerter au cas où la flamme est détectée.

Grace à sa bibliothèque plusieurs gammes et notes peuvent être programmé, ajoutant ainsi une valeur réelle aux projets Arduino.



Figure : 2.8(b)-Photo réelle du buzzer

II.7.3. Bluetooth

Le module Bluetooth (figure 2.8(c)) est l'un des composants les plus importants de notre projet. Ce dernier permet d'établir une communication entre l'application que nous allons créer avec RemoteXY ainsi que la carte Arduino Uno.

Le module Bluetooth HC-05 présente 6 broches pour permettre d'établir la connexion.

- VCC broche d'alimentation. Typiquement connectée à la broche 5V de l'Arduino.
- GND masse. Typiquement connectée à la broche GND de l'Arduino
- RX broche de réception. Typiquement connecté à la broche de transmission (TX) de l'Arduino
- TX broche de transmission. Typiquement connecté à la broche de réception (RX) de l'Arduino
- State retourne 1 lorsque le module est connecté
- Key ou EN doit être alimentée pour entrer dans le mode de configuration et ne doit pas être connecté pour être en mode communication. [8]



Figure : 2.8(c)-Photo réelle du module Bluetooth HC-05

II.8.Partie logiciel

II.8.1.Logiciel Arduino

II.8.1.1.Définition du logiciel Arduino

Les créateurs d'Arduino ont développé un logiciel pour que la programmation des cartes arduino soit visuelle, simple et complète à la fois.

C'est ce que l'on appelle une IDE (figure 2.9(a)) , qui signifie Integrated Development Environment ou Environnement de Développement « Intégré » en français (donc EDI) écrit en langage JAVA.

L'IDE Arduino est le logiciel qui permet de programmer les cartes Arduino.

L'IDE affiche une fenêtre graphique qui contient un éditeur de texte et tous les outils nécessaires à l'activité de programmation.

On peut donc saisir notre programme, l'enregistrer, le compiler, le vérifier, le transférer sur une carte arduino.



Figure : 2.9(a)-Logo du l'IDE de l'Arduino

II.8.1.2.Langage de programmation Arduino

Le langage Arduino est basé sur le C/C++. Le langage de programmation d'Arduino est en effet une implémentation de Wiring (une plate- forme open source similaire de physical computing qui proposait elle-même une bibliothèque appelée Wiring qui simplifie les opérations d'entrée/sortie). Un programme Arduino est aussi appelé un sketch.[9]

II.8.1.3. Structure d'un programme Arduino

- Prise en compte des instructions de la partie déclarative
- Exécution de la partie configuration (fonction **setup()**),
- Exécution de la boucle sans fin (fonction **loop()**): le code compris dans la boucle sans fin est exécuté indéfiniment.

A. L'interface du logiciel

Les programmes Arduino sont structurés sous un schéma sketch qui contient deux blocs de fonctions obligatoires setup/loop (figure 2.9(b)).

La fonction « setup » est une fonction d'initialisation de la carte, exécuté une fois; elle est utilisée au début du programme après les déclarations pour initialisation, démarrage des bibliothèques, modification du paramétrage des broches.

La fonction « loop » est une fonction principale exécutée indéfiniment en boucle.

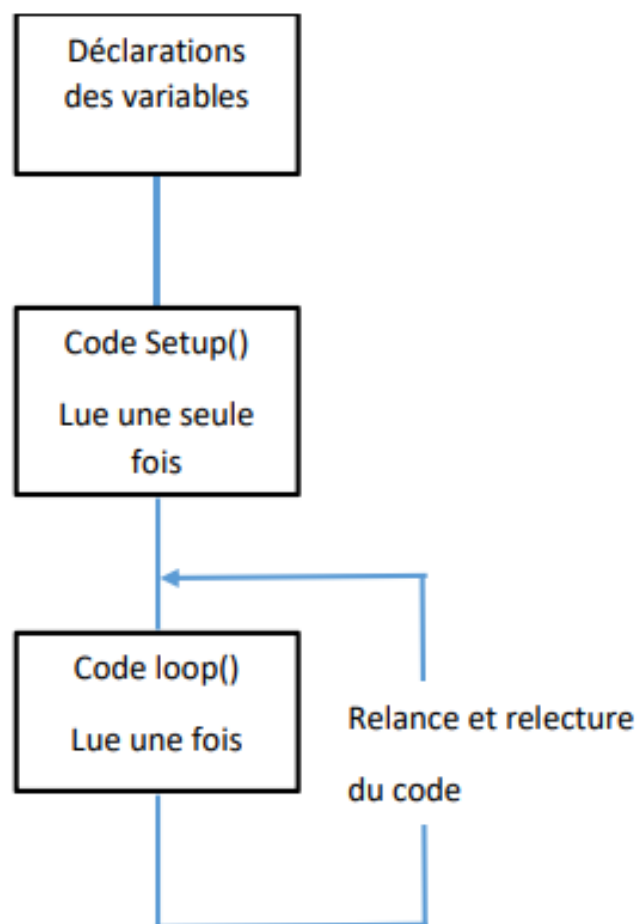


Figure : 2.9(b)-Interface du logiciel

B. Les étapes pour exécuter un programme dans IDE de l'Arduino

- Ouvrir le logiciel Arduino (figure 2.9(c))

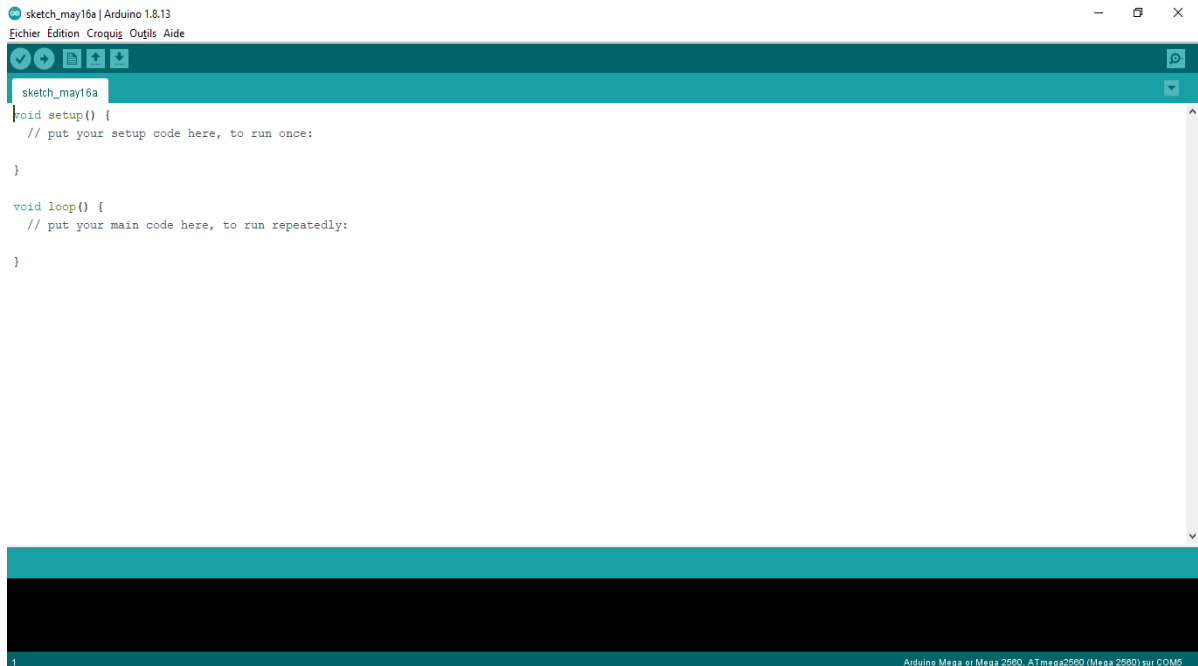


Figure : 2.9(c)-Fenêtre d'édition de programme

- Choisir le type de carte : faire Outils > Type de carte > Arduino UNO (figure 2.9(d))

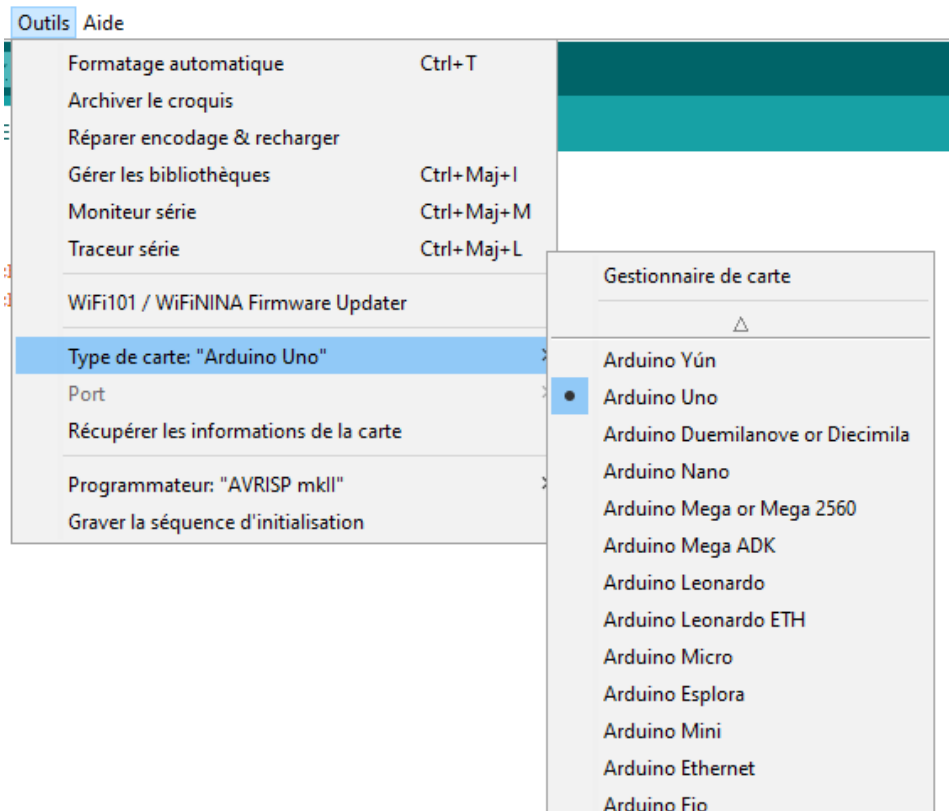


Figure : 2.9(d)-Type de carte

- Puis connecter la carte : faire Outils > Port > COM

Quand le port est grisé le programme ne peut pas être téléverser vers la carte Arduino (figure 2.9(e)).

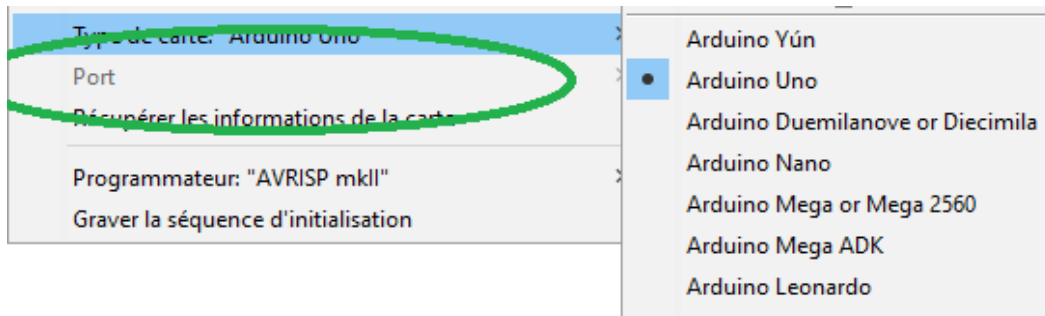


Figure : 2.9(e)-Problème de la sélection du port

Pour régler ce problème il suffit de mettre à jour les pilotes dans la fenêtre gestionnaire des périphériques après avoir connecté le fil USB avec le port en question (COM5 dans notre cas) (figure 2.9(f)).

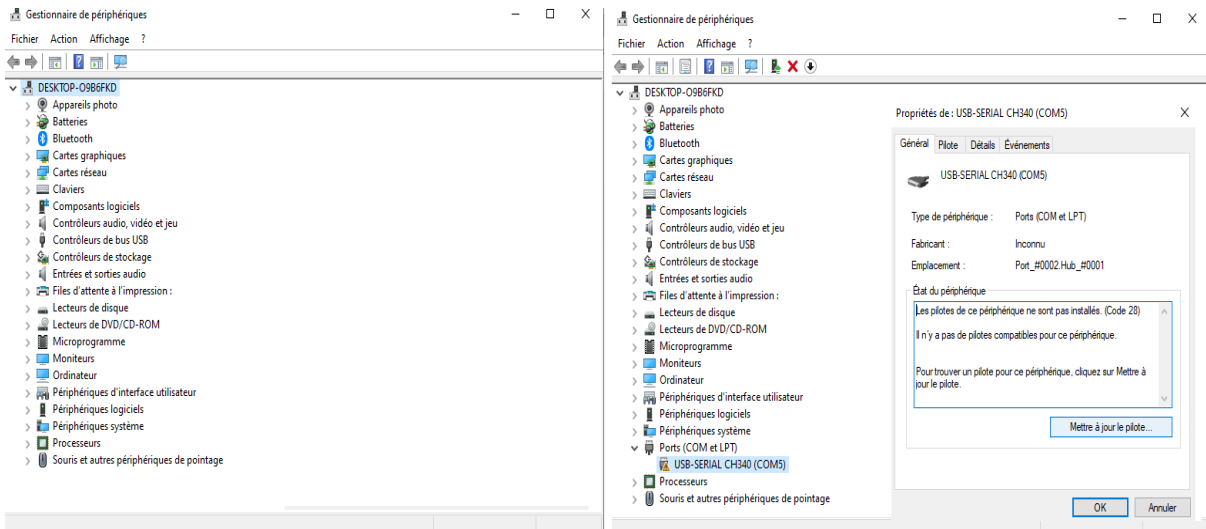


Figure : 2.9(f)-Configuration du port

Le port est maintenant disponible et le programme peut être téléversé sans erreur (figure 2.9(g))

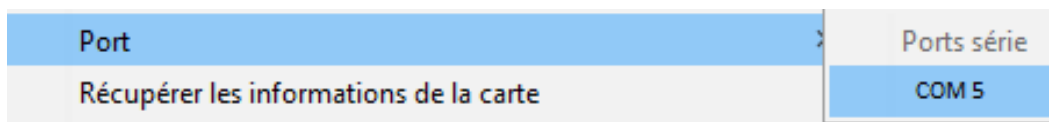


Figure : 2.9(g) --Choix du port

- Ecrire le programme, le sauvegarder et l'exécuter (figure 2.9(h))



Figure : 2.9(h)-Barre d'outils Arduino

- Téléverser le programme dans la carte Arduino via la connexion USB (figure 2.9(i))

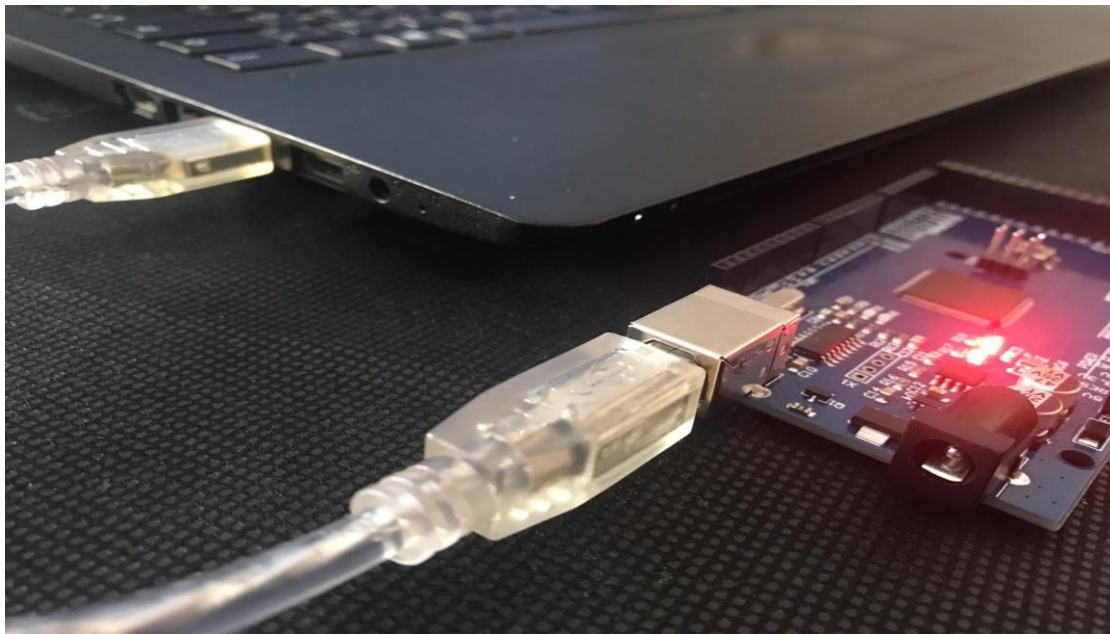


Figure : 2.9(i)-Photo réel du téléversement du programme vers la carte Arduino Uno

C. Explication des rôles des instructions de la barre de commande (figure 2.9(j))

Compile le programme : c'est à dire vérifie s'il n'y a pas d'erreur. Peut être utilisé, même si aucune carte n'est connectée, pour corriger la syntaxe d'un programme.

Cette boucle est exécutée **une fois** à l'initialisation du programme et quand la bouton **reset** de la carte est pressé.

La partie de programme se trouvant se répètent **indéfiniment** tant que la carte est alimentée.

Les messages d'erreur s'afficheront ici.

Lance le moniteur série qui affiche les valeurs et mesures renvoyée par la carte par la fonction : **Serial.print()** **Serial.println()** (pour sauter des lignes)

Téléverse le programme sur la carte Arduino, pour qu'il puisse tourner. Dès qu'une modification est faite dans le programme, il faut téléverser le programme de nouveau pour qu'elle soit prise en compte.

Figure : 2.9(j)-Rôles des instructions de la barre de commande

D. Les différentes règles à respecter pour une exécution correcte sans erreur

- Le logiciel demande la sauvegarde le programme dans un dossier qui porte le même nom à la première compilation.
- Les majuscules et minuscules sont prises en compte dans les fonctions et variables.
- La plupart des lignes se termine par un point-virgule ;

- Les blocs, boucles et fonctions sont encadrées par des accolades { }.
- Toutes les variables doivent être déclarées (nom et type).
- Les commentaires sont définis par // en début de ligne. Les blocs de commentaires par /* au début et */ à la fin.

II.8.2. Application Remote XY

II.8.2.1. Présentation

Pour notre projet de ferme intelligente, nous avons choisi de contrôler le système par le biais d'un smartphone en utilisant une application de contrôle.

Pour cela nous avons opté pour l'application nommée « Remote XY ».

RemoteXY (figure 2.10(a)) est un moyen facile de créer et d'utiliser une interface utilisateur graphique mobile pour les cartes de contrôleur à contrôler via un smartphone ou une tablette.

Le système comprend :

- Editeur d'interfaces graphiques mobiles pour cartes contrôleur, situées sur le site remotexy.com
- Application mobile RemoteXY qui permet de se connecter au contrôleur et de le contrôler via une interface graphique. [10]



Figure : 2.10(a)-Logo de RemoteXY [remotexy.com]

II.8.2.2. Eléments de contrôle

Remote XY permet de développer toute interface de gestion graphique, en utilisant des éléments de contrôle, d'affichage et de décoration ou toute combinaison de ceux-ci (figure 2.10(b)). Ces interfaces graphiques peuvent être développées pour n'importe quelle tâche.

Le site RemoteXY propose aux utilisateurs de modéliser l’interface graphique avec plusieurs composants divisés en quatre compartiments : Contrôle, Capteurs, Indications et Décorations.

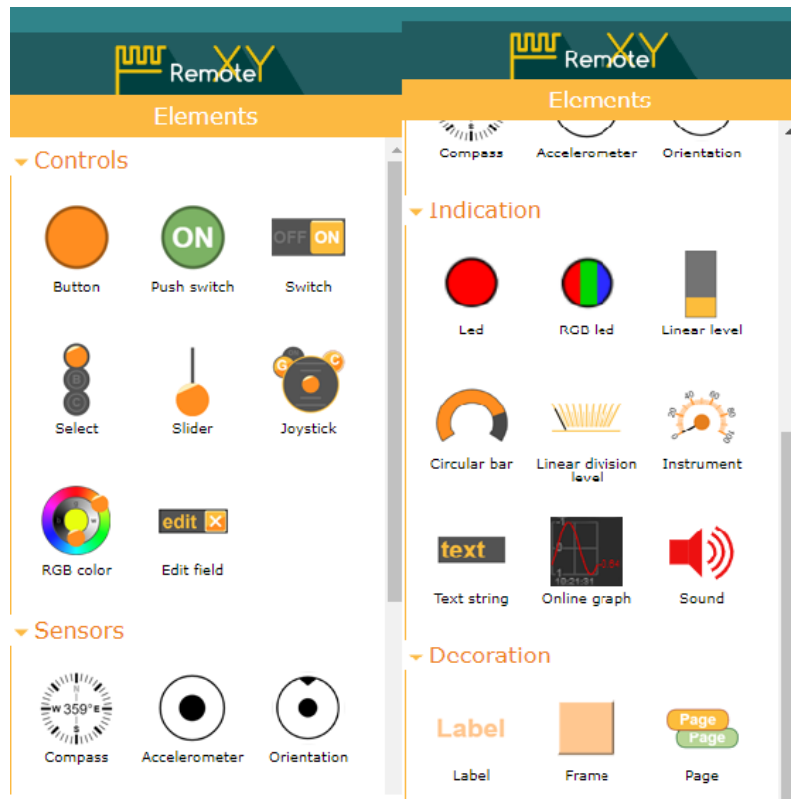


Figure : 2.10(bl)-Photo explicative des éléments disponibles sur RemoteXY

II.8.2.3. Configuration

Après le développement de l’interface graphique, on obtient un code source à téléverser, après modification, dans la carte Arduino utilisée qui implémente l’interface graphique.

La connexion entre l’application et la carte doit être configurée dès le début (WI-FI ou Bluetooth), la carte utilisée ainsi que le logiciel de programmation doivent aussi être choisis avant de commencer à modéliser l’interface graphique (figure 2.10(c)).

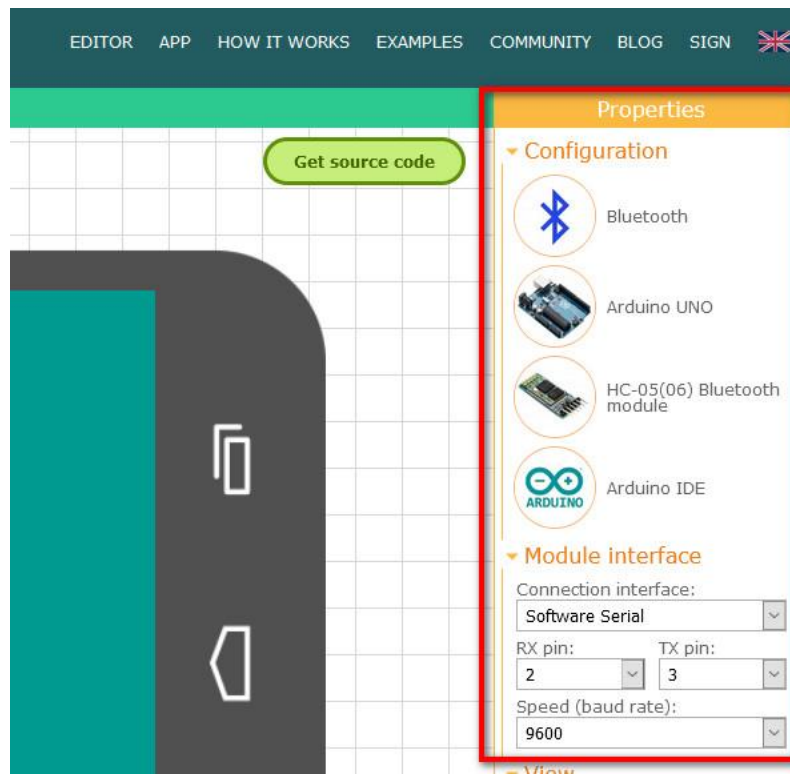


Figure : 2.10(c)-Photo explicative de la configuration de RemoteXY

II.8.2.4. HC-05 et Remote XY

Dans notre cahier de charge nous avons décidé d'utiliser un module Bluetooth HC05 afin de s'interconnecté avec l'application que nous allons créer et qui permettra de contrôler notre système de ferme intelligente. La photo ci-dessous (copier du site officiel de RemoteXY.com) explique clairement et avec détails comment se fait l'interaction entre le module Bluetooth, la carte Arduino UNO ainsi que l'application créée (figure 2.10(d)).

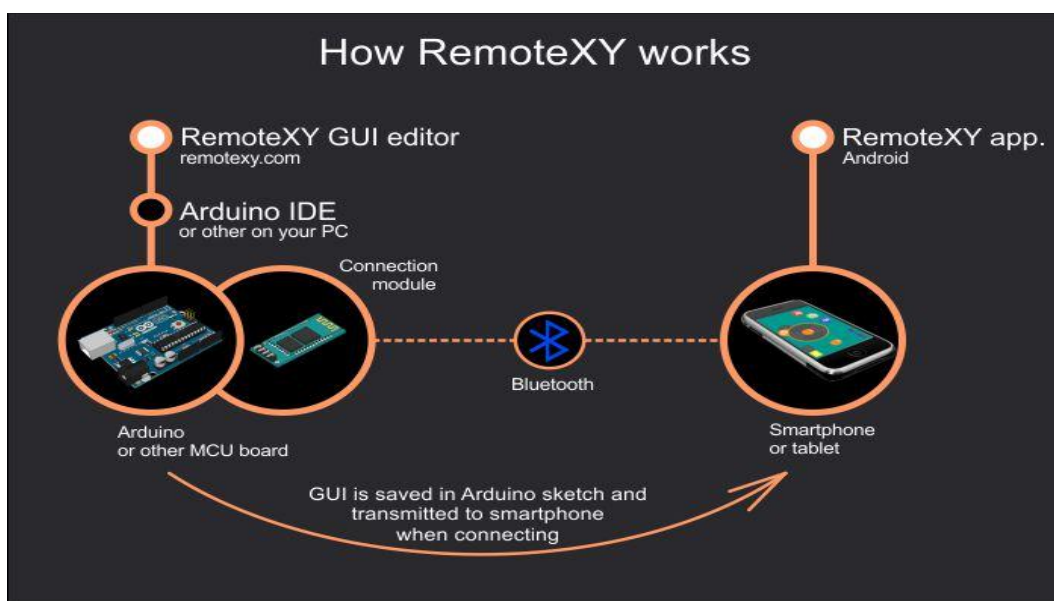


Figure : 2.10(d)-Photo explicative du fonctionnement de RemoteXY [remotexy.com]

II.9. Conclusion

Dans ce chapitre nous avons fait une étude théorique de notre projet. Tout d'abord nous avons présenté la carte Arduino qui constitue le cœur de notre réalisation ainsi que les différents composants électroniques que nous avons utilisé.

Ces composants sont ensuite classés en quatre parties, alimentation, capteurs, pré-actionneurs et actionneurs, en expliquant leurs principes de fonctionnement et en donnant leurs caractéristiques et leur branchement avec la carte Arduino Uno.

Nous avons enfin, présenté les logiciels adoptés dans ce travail à savoir IDE Arduino et l'application d'interface graphique et de contrôle à distance RemoteXY.

Les informations acquises dans ce chapitre seront pris en compte dans la réalisation pratique de notre ferme intelligente qui sera expliquée dans le prochain chapitre.



Chapitre III :
**Conception Et
Réalisation Pratique**

III.1. Introduction

Après avoir exposé quelques généralités sur la ferme Hi-Tech et présenter les deux parties théoriques pour la réalisation de notre projet à savoir parties matériels et logiciels, nous allons enfin aborder la réalisation pratique de l'enceinte agricole intelligente.

Cependant, en ingénierie, la méthode d'analyse et de la conception d'un procédé reste une étape préliminaire qui doit être effectuée en premier lieu avant d'entamer la pratique. Cette phase de conception sert à répertorier les résultats attendus, en termes de fonctionnalités, de performance, de robustesse, de maintenance et de sécurité.

Au cours de ce chapitre nous allons donc :

- Mentionner les objectifs du projet, ses besoins fonctionnels et non fonctionnels ainsi que les fonctionnalités offertes par le système.

- Détailler le principe de fonctionnement de notre projet et établir l'algorithme du système sous forme d'un organigramme (Organigramme).

- Tester les différents composants en effectuant leurs branchements avec la carte Arduino Uno.

- Expliquer la création de l'interface correspondante à la commande à distance de notre projet via l'application RemoteXY et éclaircir son utilisation en détaillant chaque fonctionnalité.

- Expliquer les différents blocs du code Arduino correspondant au fonctionnement de la ferme intelligente.

- Expliquer les différentes étapes de la fabrication de la maquette de l'enceinte intelligente.

- Installer la plaque d'essai, l'Arduino, les capteurs et les actionneurs dans la maquette et réaliser les différents branchements.

- Vérifier enfin les performances de ce projet en effectuant un test ultime.

III.2. Etude des besoins du système

Dans cette partie nous allons éclaircir et clarifier l'ordre de priorité de notre système, et cela en divisant les besoins de ce dernier en deux parties.

Les besoins fonctionnels sont des étapes qui répondent aux exigences de notre cahier de charge et auxquelles ils assureront le bon fonctionnement de notre système. Quant aux besoins non fonctionnels, ces derniers sont considérés comme des options supplémentaires qui n'influencent en aucun cas la sûreté de fonctionnement du système.

III.2.1. Besoins fonctionnels

Notre projet de ferme intelligente doit répondre à 5 besoins fonctionnels majeurs, ces derniers doivent tous être exécutés dans un ordre de priorité bien défini et avec un processus similaire à chaque cycle afin d'assurer le bon fonctionnement de notre machine. Les axes importants pour cela sont résumés en :

Synchronisation de l'application via le module Bluetooth.

Acquisition de la mesure de la température à travers le capteur DHT11.

Obtention du taux d'humidité de la terre Grâce au module d'humidité de sol.

Détection (ou pas) des flammes via un capteur de flamme avec une tête infrarouge

Contrôle à distance de l'éclairage grâce à une fonctionnalité mise en œuvre dans l'application.

Déclenchement automatique des actionneurs suivant les paramètres mesurés, ce point sera détaillé dans les prochaines parties.

III.2.2. Besoins non fonctionnels

Après avoir définis les besoins fonctionnels de notre système en quelques actions majeures et indispensables à la fois, nous allons désormais présenter les besoins non fonctionnels qui n'influencent pas le bon déroulement des cycles de notre processus, mais qui agissent à la fois en matière de robustesse, design, rapidité ainsi que d'autres fonctionnalités qui restent des perspectives futures au développement de notre projet de fin d'étude :

Rapidité et robustesse du processus en utilisant des composants plus performants avec un traitement rapide et une plage de mesure plus large

Capacité supérieure En augmentant les dimensions de l'enceinte afin de pouvoir ajouter d'autres fonctionnalités.

Maintenabilité En modifiant le branchement et le câblage pour pouvoir assurer la maintenance d'une manière sûre et efficace.

III.3. Principe de fonctionnement

Pour pouvoir faire la conception et la réalisation d'un système. Il est indispensable de bien étudier son principe de fonctionnement, et de pouvoir comprendre le lien entre les composants électroniques et les étapes du processus. Dans notre cas, nous commencerons d'abord par mentionner les principales étapes qui assureront à la ferme intelligente de remplir sa fonction d'usage. Puis, on enchainera par la création d'un algorithme. Cela permettra de faciliter la compréhension du système afin d'avoir une idée claire et précise sur le principe de fonctionnement de notre enceinte intelligente.

III.3.1. Principales étapes du processus

La ferme intelligente que nous avons conçu, est divisée en quatre compartiments majeures, chaque compartiment possède ses propres caractéristiques et spécifications techniques ainsi que des capteurs et des actionneurs qui agissent selon des paramètres que nous allons configurer, et cela suivant le cahier de charge que nous avons établis lors du chapitre numéro un, et que nous allons à présent détailler et mieux assimiler avant d'attaquer les deux parties majeures de notre travail qui sont la programmation et la réalisation finale.

La première des étapes de notre projet est l'allumage du système et l'initialisation des différents modules et composants pendant un laps de temps assez court afin de se stabiliser et assurer le bon fonctionnement.

Ensuite, il faudra connecter le module Bluetooth avec l'interface que nous avons créé avec l'application RemoteXY à travers les deux broches Tx et Rx du Hc-05. Une fois la connexion établie. L'application envoi le message "Receiving Variables" et l'interface conçu s'affiche sur notre téléphone.

À présent que l'allumage, l'initialisation et la synchronisation du système sont effectués, il est temps pour nous d'expliquer les 4 compartiments majeurs.

- **Compartiment de refroidissement** ce compartiment se comporte d'un capteur de température DHT11, d'un ventilateur connecté via un relais à l'Arduino, ainsi que de deux Leds, verte et rouge.

Ce compartiment dépend essentiellement d'un seuil de température mesurée par le DHT11. Si cette dernière est inférieure au seuil, l'application affiche le message "Seuil température dans les normes" dans l'espace dédiée à ce paramètre. Dans ce cas-là, la led verte est allumée et la rouge est éteinte. Si, par contre le seuil qu'on a programmé est dépassé, le ventilateur s'active afin de refroidir la ferme et donc faire descendre la température. La led rouge s'allume quant à la verte, elle s'éteint. Enfin le message dans l'application change et devient : "Seuil température atteint".

La photo ci-dessous illustre les composants de ce compartiment (figure 3.1).

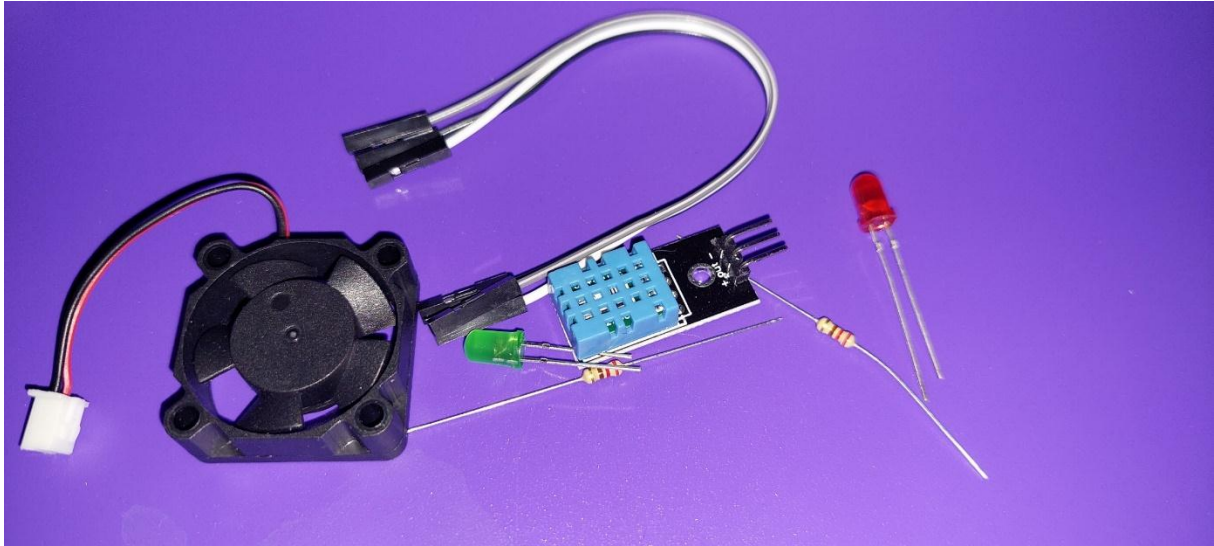


Figure 3.1. Photo des composants du compartiment de refroidissement

• **Compartiment d'incendie** dans cette partie, nous avons utilisé un capteur de flamme à détection infrarouge, deux leds verte et rouge ainsi qu'un buzzer.

Le fonctionnement de ce compartiment est assez simple, car nous allons utiliser le pin numérique du capteur. Donc si aucune flamme n'est détectée le message "Aucune flamme détectée" est affichée sur l'APP, la led verte sera allumée et la rouge éteinte. Sinon les leds s'inversent, le buzzer sonne en continu avec gamme assez aigue comme signal sonore d'alerte et enfin le message "Flamme détectée attention!" Est affiché sur la barre correspondante à ce compartiment dans l'interface modélisée.

La photo ci-dessous illustre les composants de ce compartiment (figure 3.2).

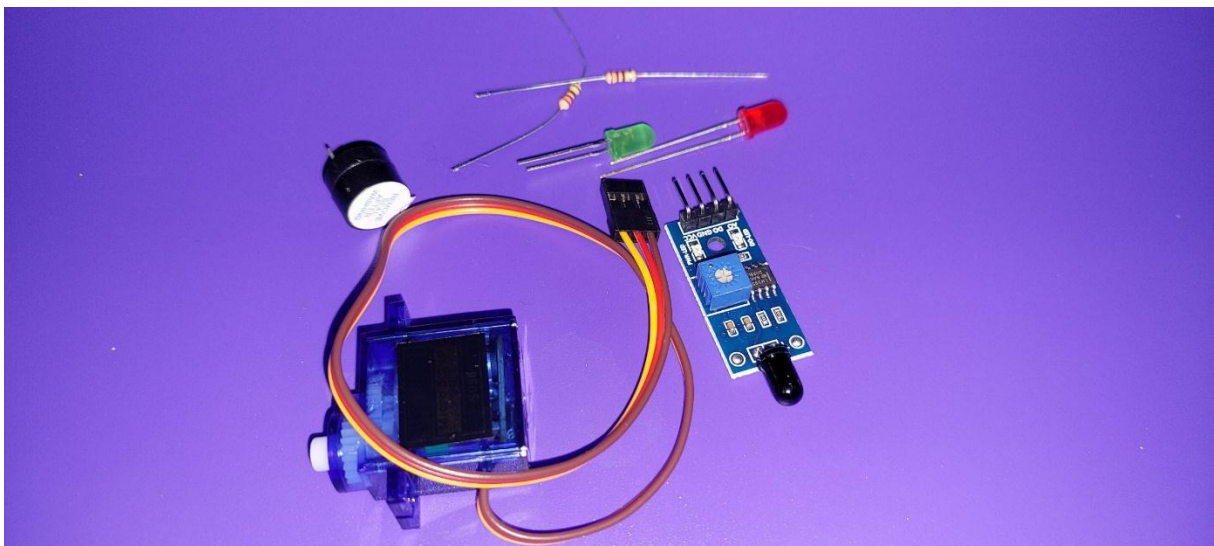


Figure 3.2. Photo des composants du compartiment d'incendie

- **Compartiment d'arrosage automatique** ce volet de notre réalisation se compose d'un capteur d'humidité de sol, d'une pompe contrôlée via un relais ainsi que de deux leds. Si le taux d'humidité de la terre, mesuré par le capteur est supérieur au seuil paramétré, la led verte est allumée et l'autre ne l'est pas. Le message "Seuil d'humidité dans les normes" est affiché sur l'interface.

Si dans l'autre cas, le capteur détecte par sa broche analogique que le taux est inférieur au seuil, l'alerte est donnée par l'allumage de la led rouge, l'affichage du message "Seuil humidité atteint pompe activée" et l'activation de l'arrosage automatique qui se produit via la pompe à eau contrôlée par un relais jusqu'à ce que le seuil soit rétabli.

La photo ci-dessous illustre les composants de ce compartiment (figure 3.3).

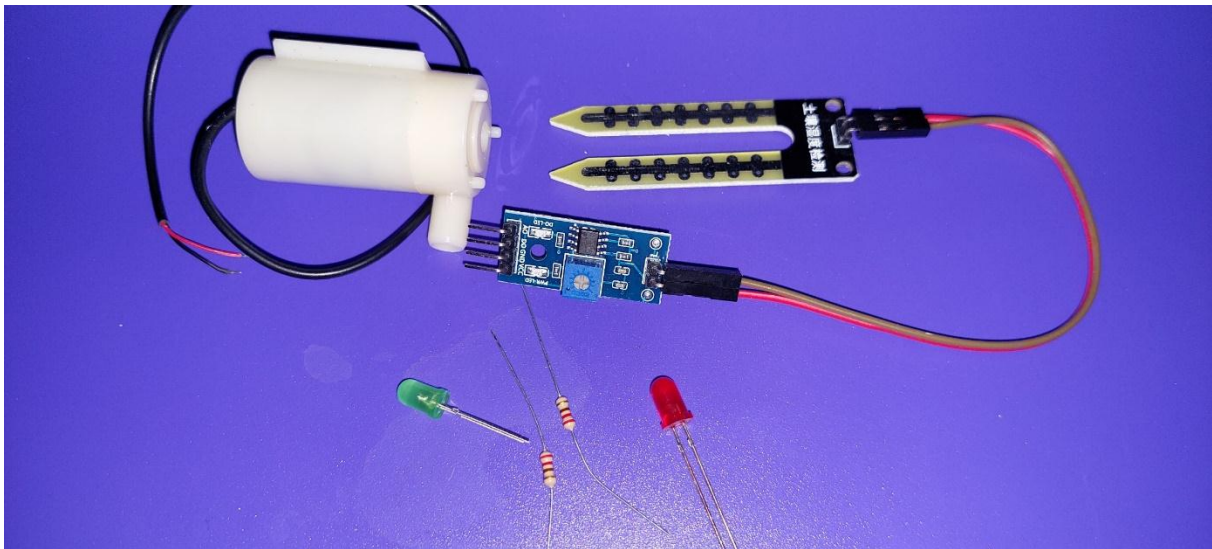


Figure 3.3. Photo des composants du compartiment d'arrosage automatique

- **Compartiment d'éclairage** Ce dernier compartiment est le plus standard et le plus facile à comprendre, dans l'application on a ajouté un switch, si la position du switch est sur la partie "ON" la lampe commandée par le relais s'allume et on éclaire l'enceinte, sinon la lumière est éteinte et notre système nagera dans l'obscurité

III.3.2. Schéma bloc du système

Un schéma bloc est la représentation visuelle d'un algorithme (figure 3.4 (a,b,c,d)). Il montre les enchaînements de décisions et d'opérations à faire pour un algorithme donné. Un algorithme est une suite de règles opératoires rigoureuses propre à un calcul.

Dans cette partie, nous allons traduire toutes les étapes que nous avons citées précédemment en un algorithme sous forme d'un schéma bloc.

Cette partie est indispensable pour la suite de notre travail, et précisément pour la prochaine étape de ce mémoire qui sera dédié à la programmation.

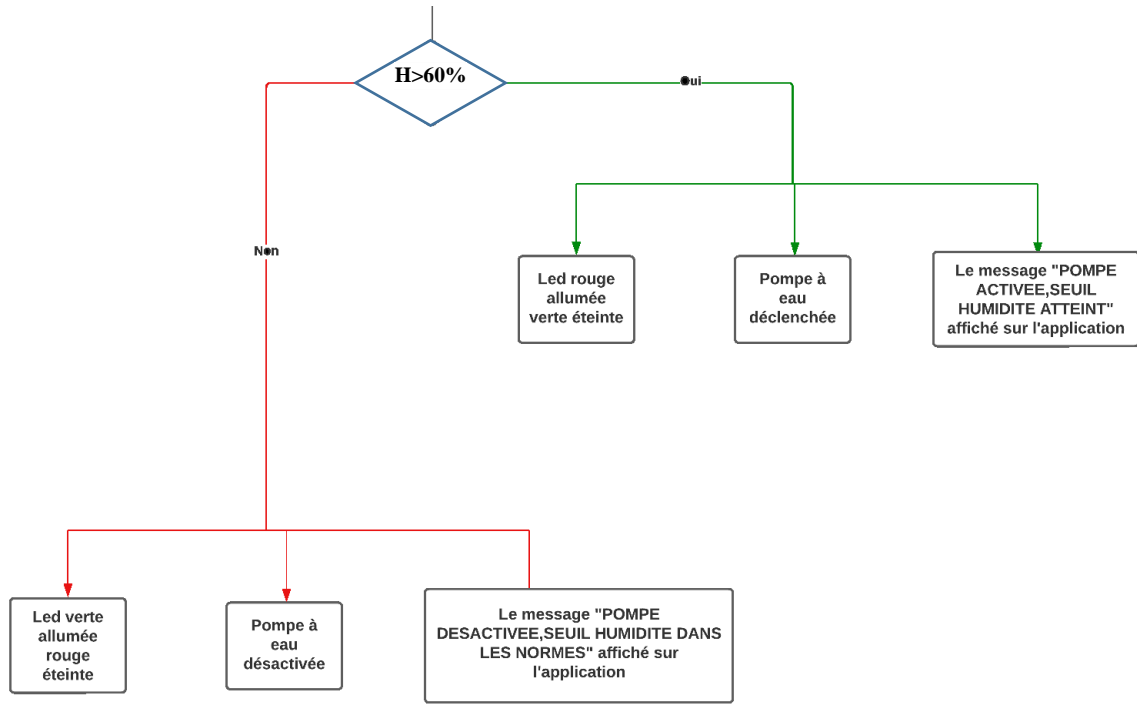


Figure 3.4(a) Schéma bloc comportement d'arrosage de notre ferme intelligente

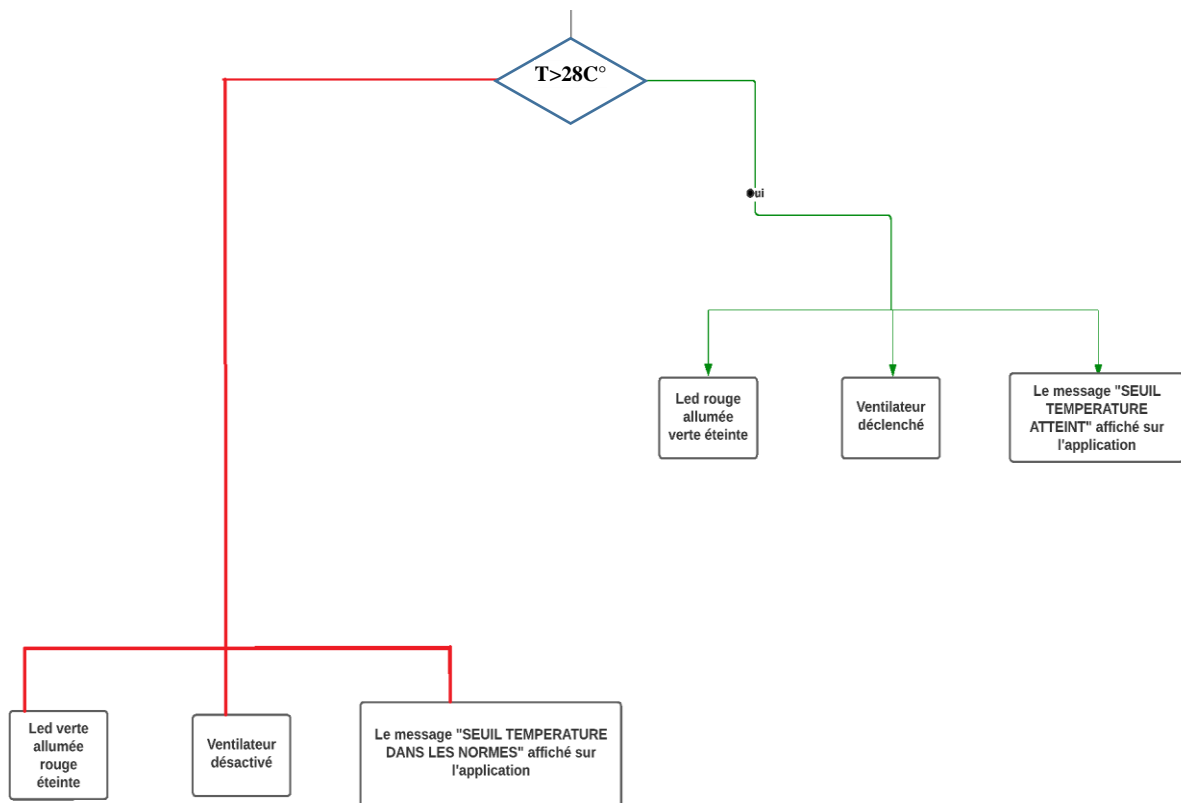


Figure 3.4(b). Schéma bloc comportement ventilation de notre ferme intelligente

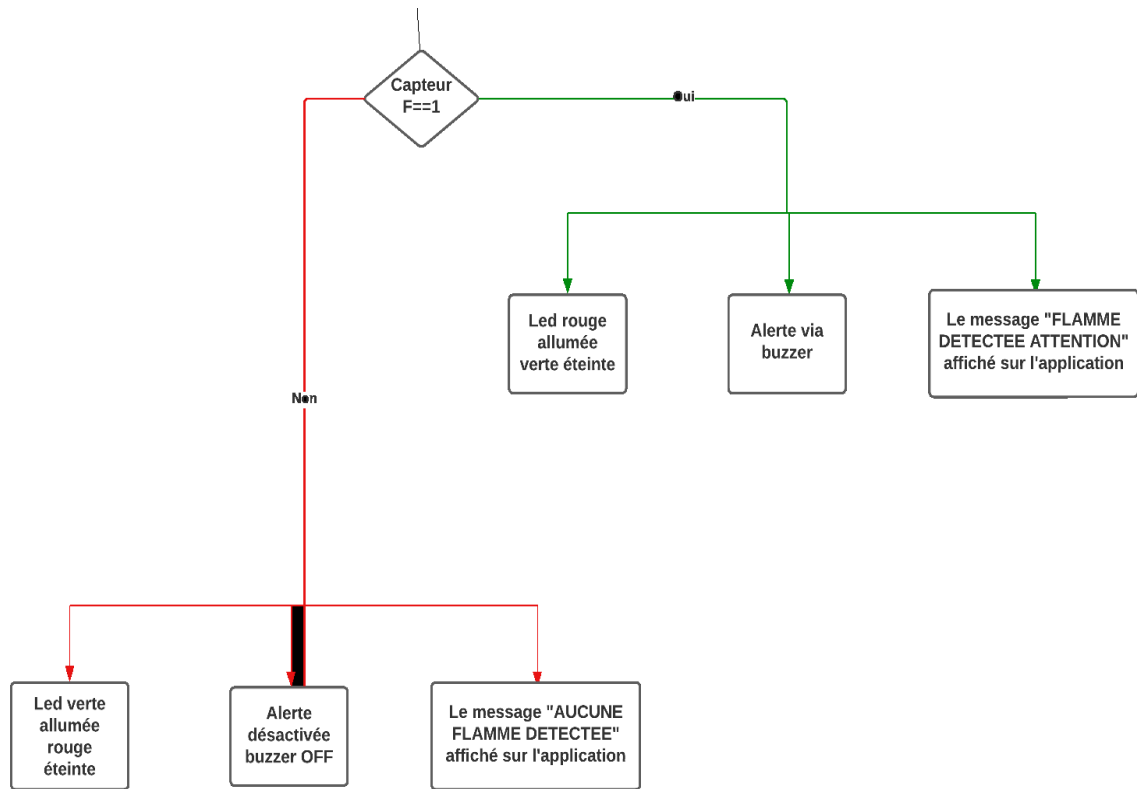


Figure 3.4(c). Schéma bloc comportement d'incendie de notre ferme intelligente

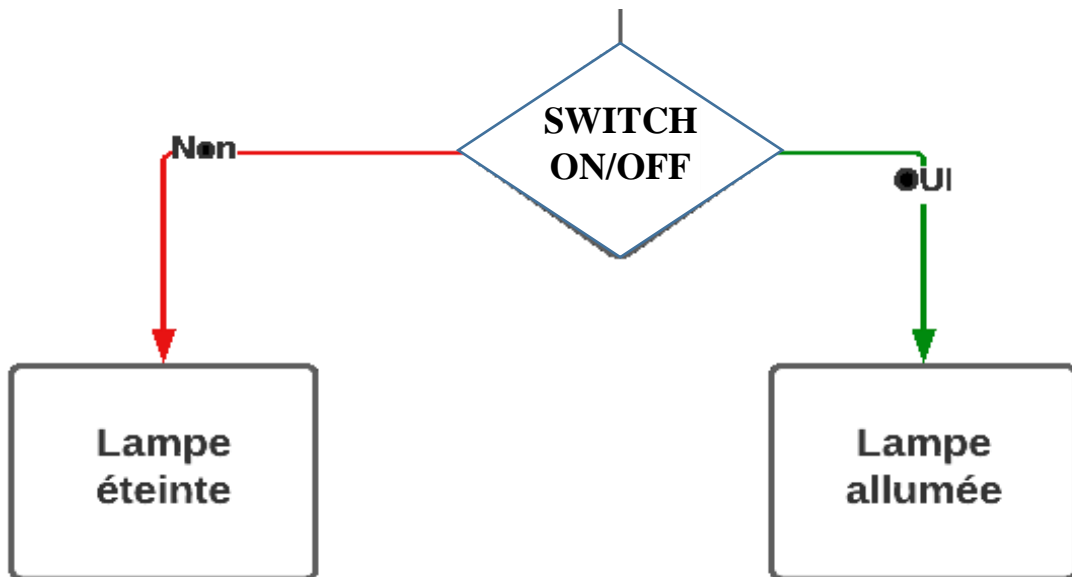


Figure 3.4(d). Schéma bloc d'éclairage de notre ferme intelligente

III.4. Software

Après avoir détaillé le principe de fonctionnement de notre système en expliquant le procédé étape par étape tout en schématisant ceci en un diagramme d'activité. Nous allons à présent pouvoir entamer la partie Software de ce projet qui sera composée de deux phases, la première sera dédiée à la création de notre interface sur l'application RemoteXY. Quant à la deuxième, elle sera consacrée à détailler le volet de la programmation de notre carte Arduino Uno par le logiciel correspondant.

III.4.1. Création de l'interface

Dans cette partie, nous allons détailler le chemin qu'on a suivi afin de modéliser une interface simple et utile à notre projet de ferme intelligente. Nous allons donc voir chaque étape de création de l'application RemoteXY.

III.4.1.1. L'éditeur

La première phase est de rechercher le site RemoteXY sur le moteur de recherche de notre ordinateur puis entrer sur le site correspondant. Ensuite, on clique sur 'Editor' pour commencer à éditer notre interface (figure 3.5).



Figure 3.5. Photo de la page principale du site RemoteXY

Après avoir cliqué sur le bouton mentionné sur la photo ci-dessus, voici à quoi ressemble notre tableau d'édition (figure 3.6).

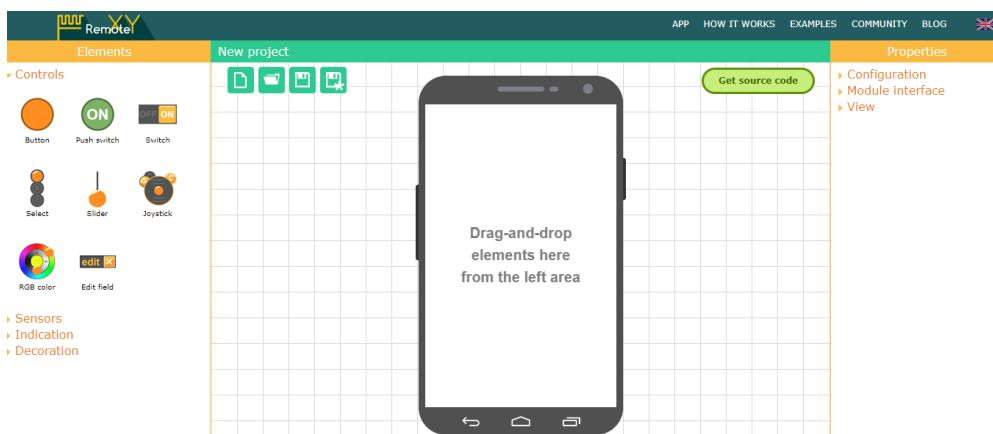


Figure 3.6. Photo de l'éditeur de l'interface

III.4.1.2. Configuration des paramètres

La première des choses à faire est de configurer les quatre principaux paramètres avec lesquelles nous allons travailler.

On commence par choisir la carte Arduino Uno dans l’onglet ‘Board’, puis on sélectionne on le Bluetooth comme mode de connexion dans ‘Connection’. Ensuite on clique sur le composant HC05 comme étant le module de connexion. Enfin, nous sélectionnons l’Arduino IDE comme logiciel de programmation (figure 3.7 (a,b,c,d)).

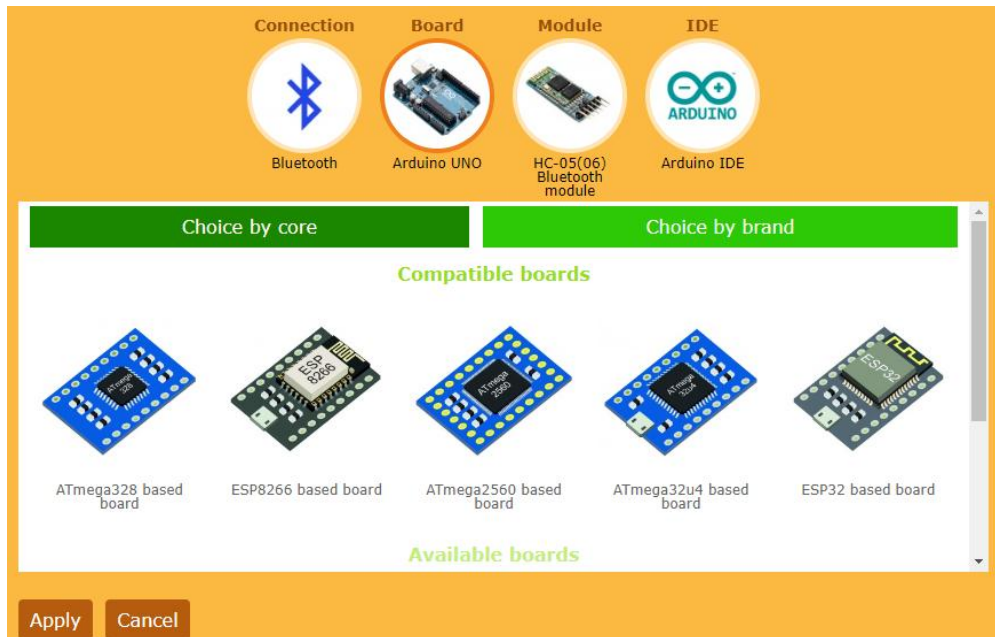


Figure 3.7(a). Choix de la carte

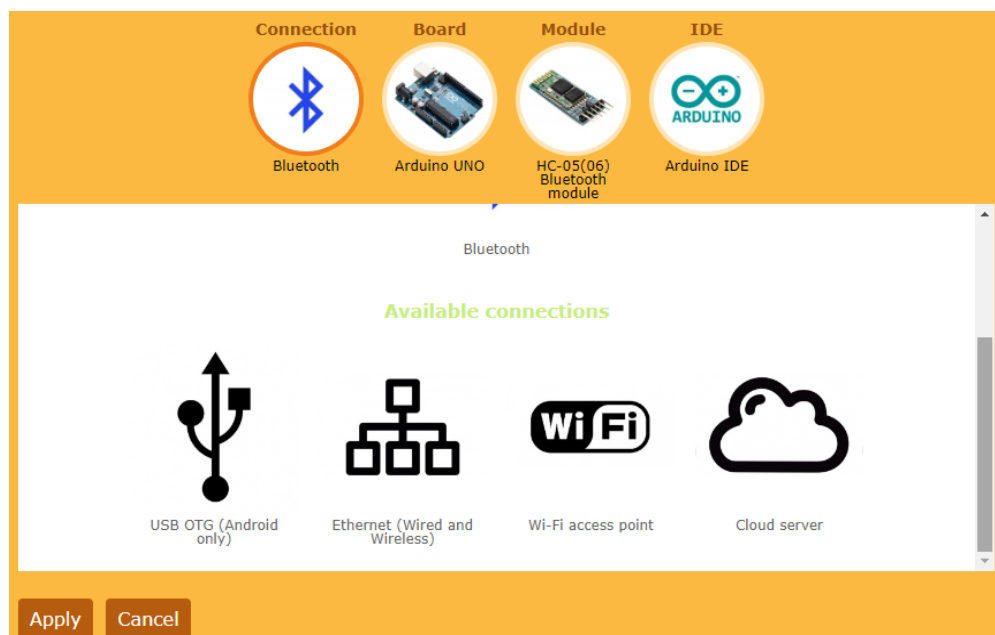


Figure 3.7(b). Choix du mode de connexion



Figure 3.7(c). Choix du module Bluetooth

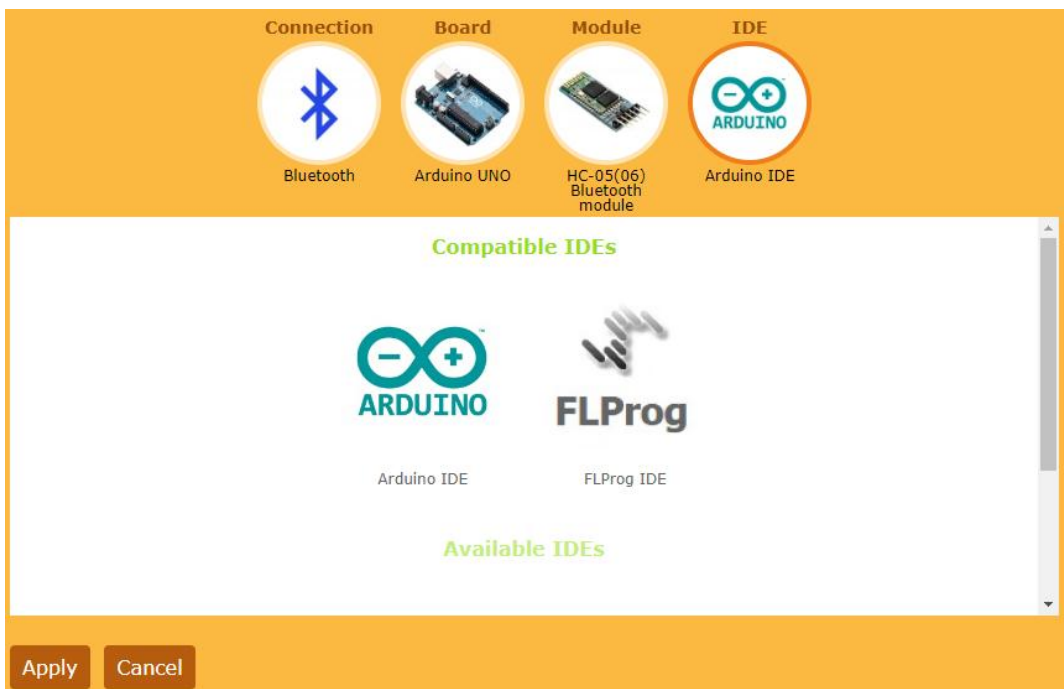


Figure 3.7(d). Choix de l'interface de programmation

Les quatre illustrations ci-dessus sont les étapes qu'on a suivies pour configurer nos paramètres suivant le cahier de charges établi.

La photo suivante représente les propriétés et résume la configuration qu'on a faite dans cette partie (figure 3.8).



Figure 3.8. Configuration des propriétés de l'interface

III.4.1.3. Liaison à l'interface

Dans cette étape, nous devons établir la connexion de l'interface en mode 'Software Serial' ce qui veut dire une liaison logiciel correspondante au module Bluetooth. Puis, on configure les Pin 2 et 3 Pour Rx (récepteur) et Tx (émetteur). Enfin, on sélectionne la fréquence (vitesse) du moniteur en 9600 Baud (figure 3.9).

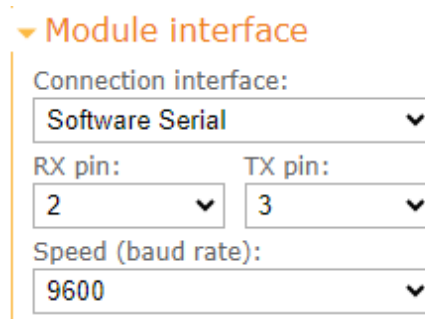


Figure 3.9. Liaison avec l'interface

III.4.1.4. Vue de l'interface

Pour notre interface de ferme intelligente, nous avons choisi dans l'onglet 'Background color', un fond vert foncé, et cela par rapport au domaine agricole. Nous avons aussi orienté l'application horizontalement pour un meilleur contrôle, et enfin, nous avons renoncé à ajouter un mot de passe, vu qu'on sera les seuls à utiliser cette application.

La première photo ci-dessous résume ce qu'on a expliqué lors de cette partie. Quant à la deuxième elle démontre la base de notre application (figure 3.10 (a,b)).

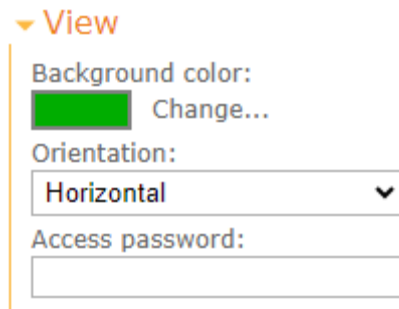


Figure 3.10(a). Vue de l'interface

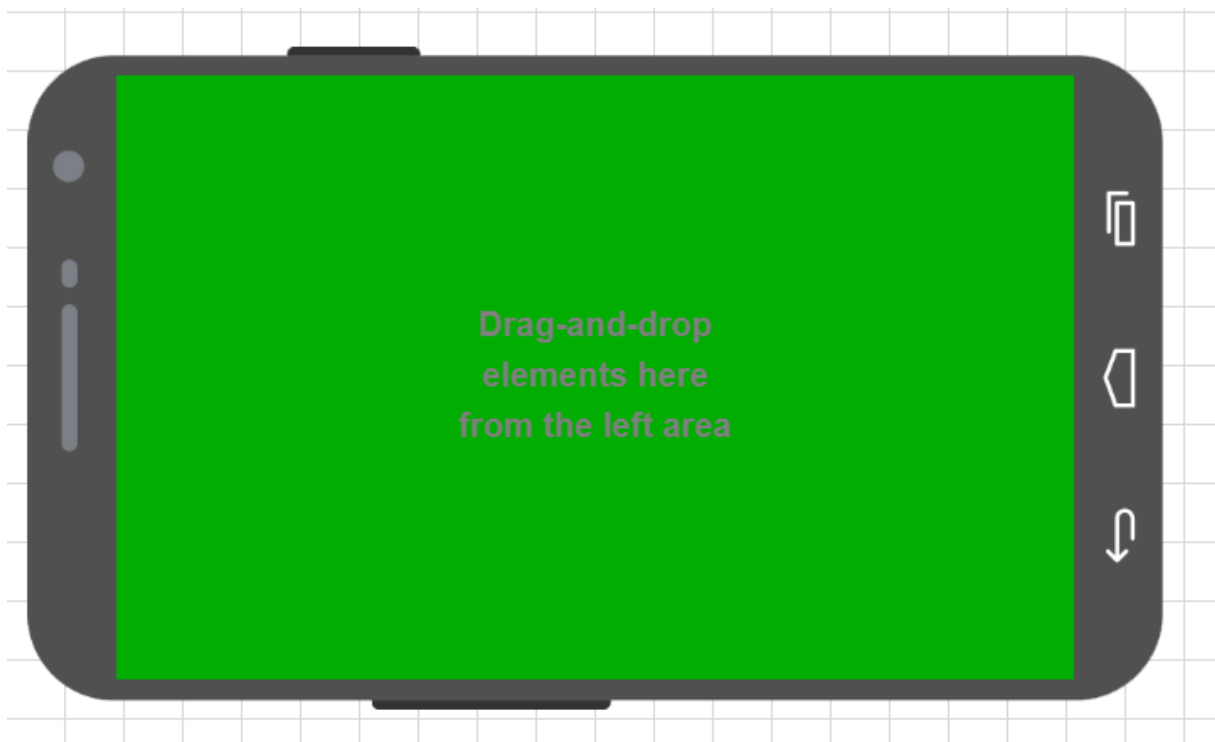


Figure 3.10(b). Arrière-plan de notre application

III.4.1.5. Décoration de l'interface

Après avoir configuré les propriétés, la liaison ainsi que la vue de l'interface, nous allons à présent commencer à ajouter les commandes de l'application.

Pour commencer, nous allons dans le volet 'Décoration', et nous ajoutant un 'Label' en haut de l'interface, ce dernier sera considéré comme un titre et sera fixé dans l'application.

Le texte de ce 'Label' sera 'FERME INTELLIGENTE ARDUINO', et sera configuré en couleur jaune et placé en en-tête de l'application.

Les trois illustrations ci-dessous clarifient les explications données au cours de cette étape de notre travail (figure 3.11 (a,b)).

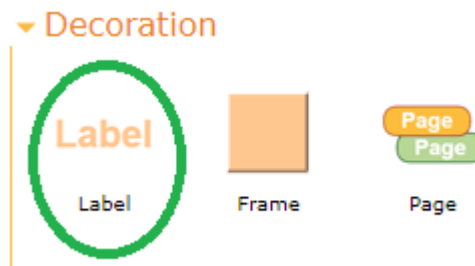


Figure 3.11(a). Choix de décoration

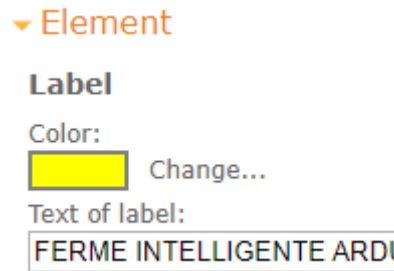


Figure 3.11(a). Modification du texte

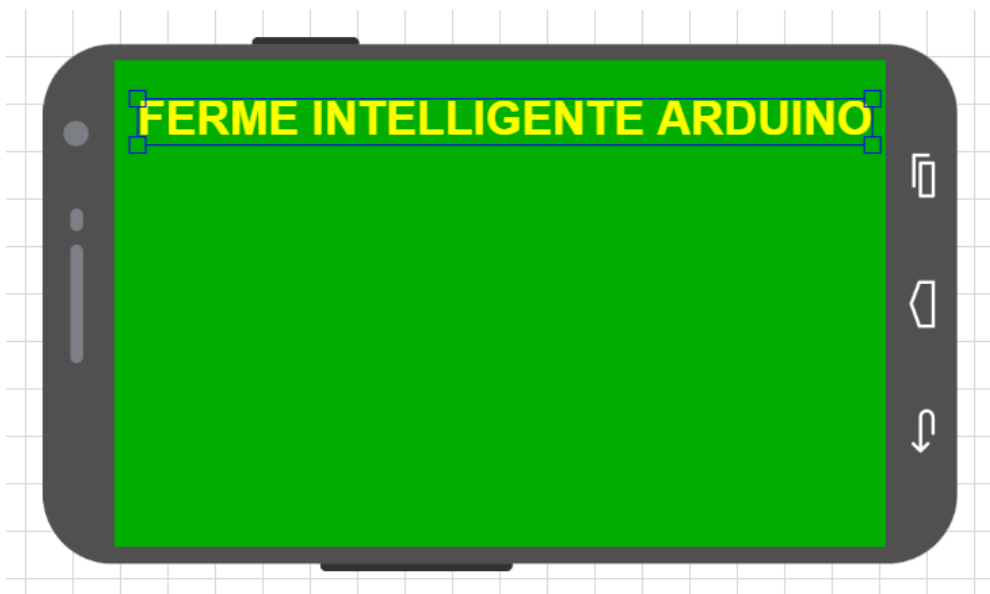


Figure 3.11(b). Orientation du texte

III.4.1.6. Indicateurs

Comme nous l'avons déjà évoqué lors de la troisième partie de ce chapitre, ce travail comporte quatre compartiments dont trois actionnés automatiquement, et dont les informations en temps réel doivent être su par l'utilisateur de cet application.

Pour cela, nous avons sélectionné trois indicateurs sous forme d'un texte. Ce texte est aligné au centre de l'afficheur avec une marge allant jusqu'à 50 caractères pour les trois compartiments.

La seule chose qui diffère, est la couleur de ces afficheurs. En effet, le texte en rouge correspond au compartiment d'incendie. Le bleu à celui de la température. Quant au texte au fond noir, il est dédié à nous informer du taux d'humidité de la terre.

Les trois captures suivantes, permettent de mieux comprendre cette phase (figure 3.12 (a,b,c)).



Figure 3.12(a). Choix de l'indicateur

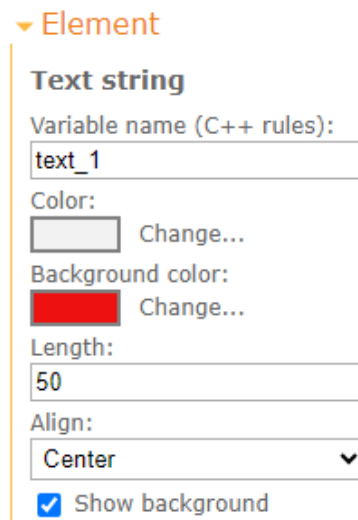


Figure 3.12(b). Configuration du texte



Figure 3.12(c). Orientation des textes

III.4.1.7. Contrôle de l'interface

Après avoir fixé le titre de l'application et placé les trois indicateurs correspondants aux trois compartiments. Nous allons à présent ajouter la dernière touche à notre application. Comme nous l'avons déjà évoqué, il ne nous reste qu'un seul compartiment que ne nous avons pas encore traité dans cette interface. Il s'agit du compartiment d'éclairage.

Pour cela, nous avons choisi le composant 'Switch' dans le volet 'Controls', afin de commander l'éclairage de notre ferme intelligente et de "Switcher" entre l'allumage ou pas de notre lampe.

Ce Switch de forme rectangulaire et aux couleurs noir et orange, est placé en bas de l'interface et facilite la commande en temps réel de la lampe.

Les trois photos ci-dessous clarifient amplement ce qu'on a détaillé lors de cette partie (figure 3.13 (a,b,c)).



Figure 3.13(a). Choix de l'outil de contrôle

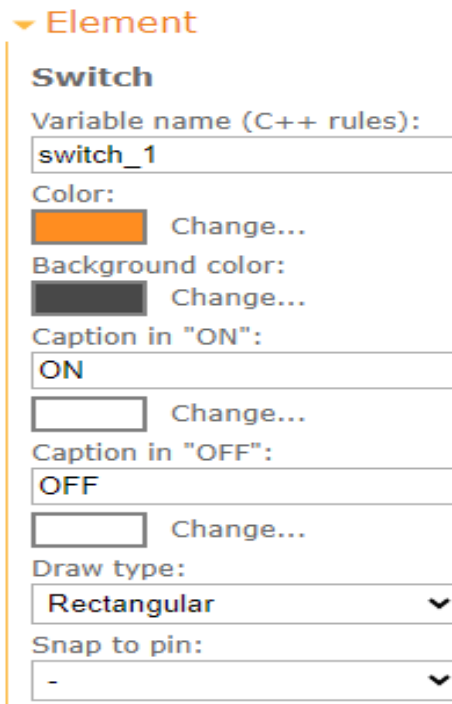


Figure 3.13(b). Configuration du switch



Figure 3.13(c). Orientation du switch

III.4.1.8. Sauvegarde du projet

Après avoir fini de configurer et de modéliser notre application de ferme intelligente. Nous allons à présent enregistrer l'interface pour ne pas perdre notre travail et pouvoir éventuellement modifier au cas de nécessité.

Comme le montre la (figure 3.14), il suffit de cliquer sur 'Save as', et de donner un nom à notre application ('Ferme intelligente'), et enfin, confirmer la sauvegarde en appuyant sur 'Save'.

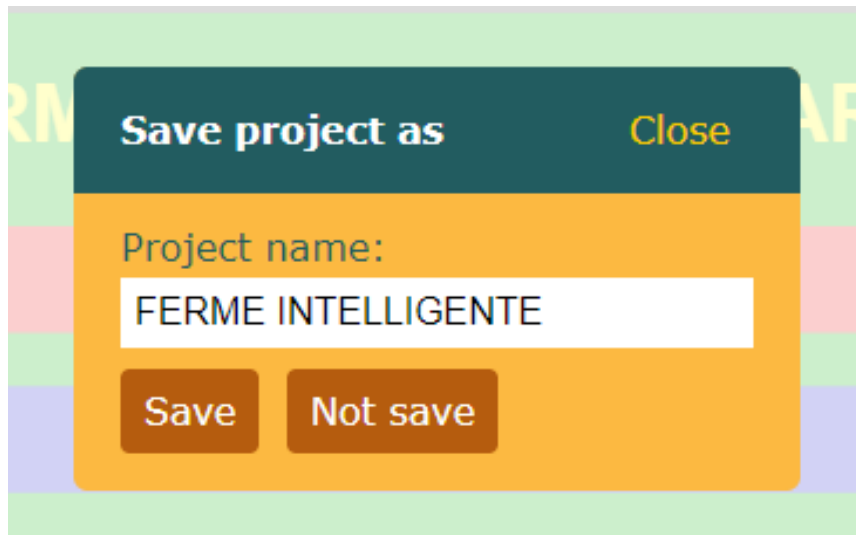


Figure 3.14. Onglet de sauvegarde

III.4.1.9. Obtention du code

Une fois l'interface modélisée et enregistrée, nous allons à présent générer le code Arduino correspondant à cette application.

Ce code permettra de se connecter avec la carte Arduino et se synchroniser avec le module Bluetooth Hc-05 afin d'afficher l'application sur notre téléphone ou notre tablette. Pour cela, nous devons d'abord appuyer sur l'onglet 'Get source code'. Ensuite, enregistrons le code en cliquant sur 'Dowland code' ainsi que la librairie de RemoteXY en cliquant sur 'Dowland Library'. Et enfin, nous installons l'application sur notre téléphone Android (figure 3.15 (a,b)).

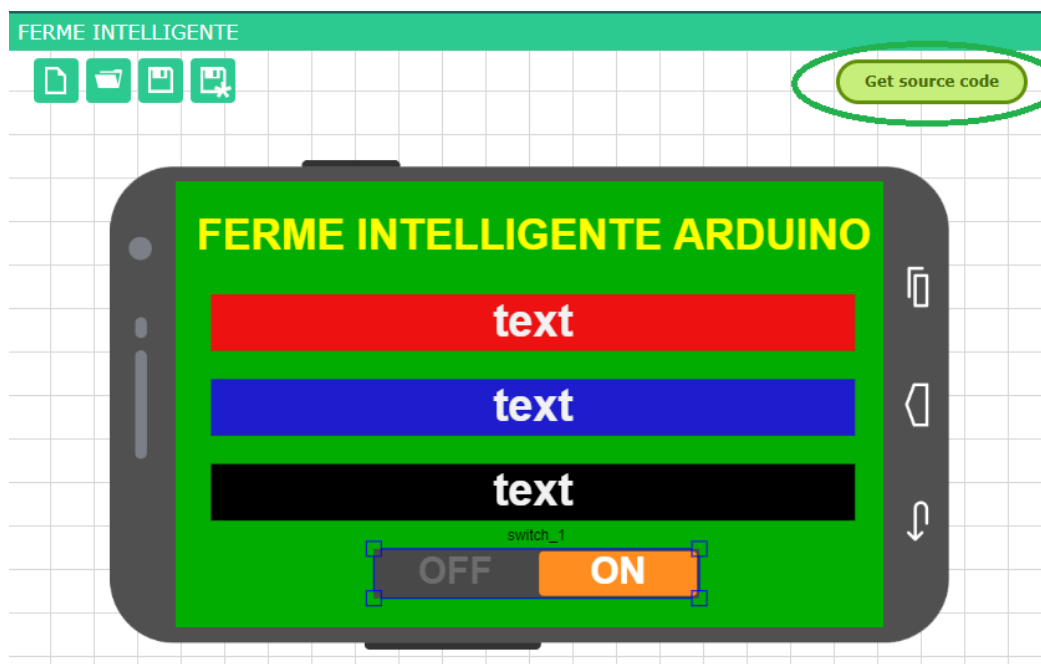


Figure 3.15(a). Obtention du code source

Source code of project: FERME INTELLIGENTE

1. Download the **source code** of the program, open it in the Arduino IDE.
2. Install **RemoteXY library** for Arduino IDE.
3. Compile the source code and upload it to the Arduino board using the Arduino IDE.
4. Correctly connect **the Bluetooth module** to the Arduino board.
5. Install the mobile app **RemoteXY ver.4.11.1** for smartphone/tablet.
6. Connect to Arduino using mobile app.

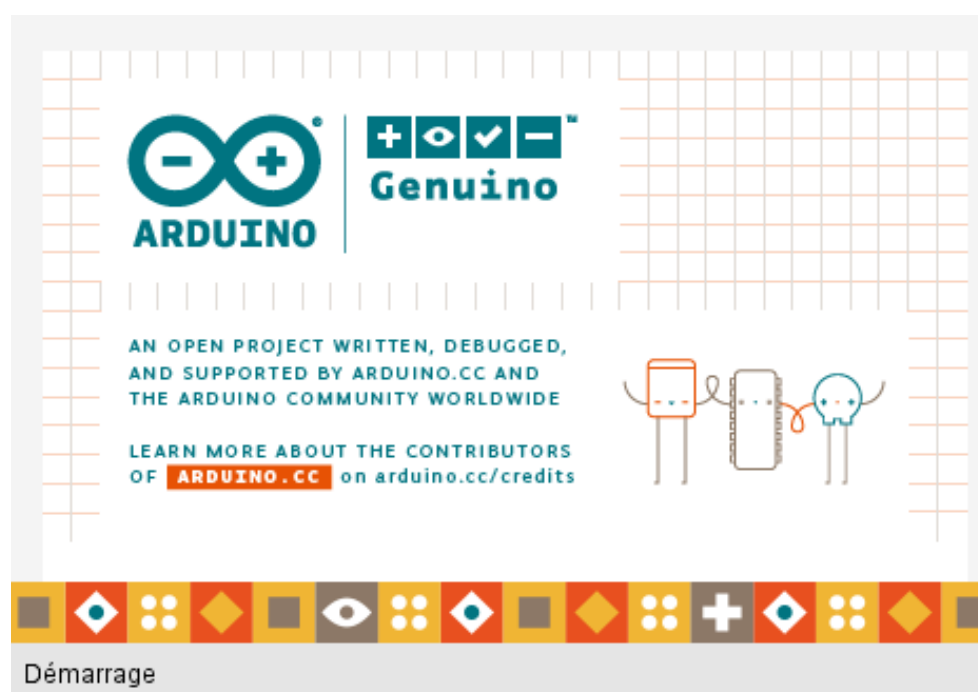
project.ino [Download code](#) [Download library](#)

Figure 3.15(b). Téléchargement du code et de la librairie

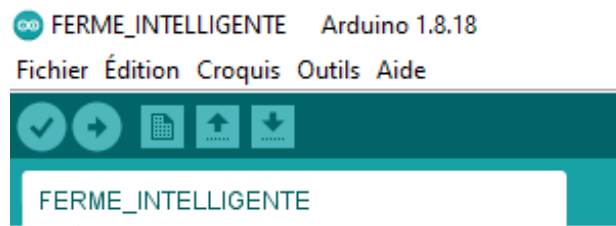
III.4.2. Programmation

Dans cette partie, nous allons exposer le programme qu'on a établi pour piloter notre carte Arduino et son microcontrôleur Atmega. Nous allons expliquer toutes les étapes du code en traduisant le langage C++ en langage humain afin de clarifier le fonctionnement de notre système.

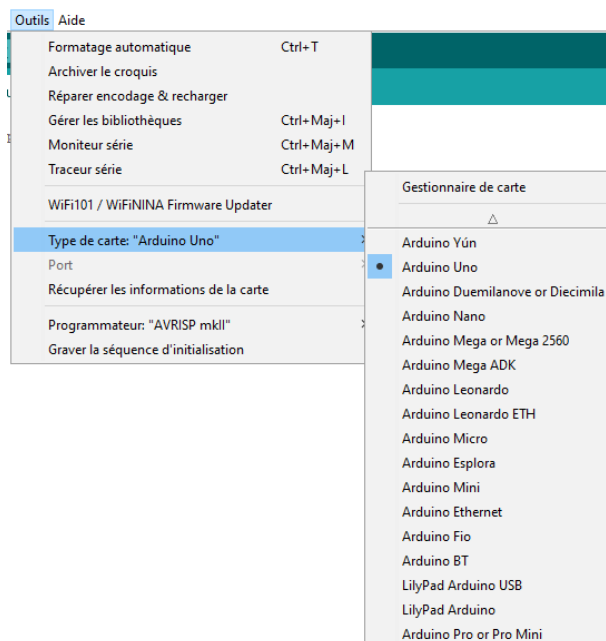
La première étape à faire a été d'installer le logiciel Arduino, puis de l'exécuter et enfin procéder au démarrage de l'IDE, il est utile de noter qu'on a installé la version 1.8.18 de l'Arduino compatible avec les systèmes Windows, Linux et IOS.



Après le démarrage de l'Arduino IDE, il est indispensable de donner un nom au sketch utilisé et de l'enregistrer par la suite. Dans notre cas, nous avons sauvegardé le fichier sous le nom de 'FERME_INTELLIGENTE'.



La dernière phase à faire avant de commencer à créer le code, est de sélectionner le type de carte correspondant au microcontrôleur, nous avons donc choisis la carte Arduino Uno sur Outils -> Type de carte.



III.4.2.1. Code source

La photo ci-dessous montre le code source généré par RemoteXY.

On peut constater que le code commence par définir la librairie de RemoteXY ainsi que les 2 pins de liaison correspondants au module Bluetooth.

La configuration de l'interface ainsi que la vue, la couleur de fond et les autres paramètres sont traduites par des chiffres et des codes.

Ensuite, on peut voir que les 3 indicateurs de textes ainsi que le switch sont déclarés séparément.

La partie Setup et Loop resteront vides et seront remplies avec les sous-programmes des autres composants correspondants aux autres compartiments.

La photo ci-dessous est celle du code source obtenu à partir de l'interface modélisée par RemoteXY.

```

1 /*
2  -- FERME INTELLIGENTE --
3
4  This source code of graphical user interface has been generated automatically by RemoteXY editor.
5  To compile this code using RemoteXY library 3.1.8 or later version download by link http://remotexy.com/en/library/
6  To connect using RemoteXY mobile app by link http://remotexy.com/en/download/
7  - for ANDROID 4.11.1 or later version; - for iOS 1.9.1 or later version;
8
9  This source code is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public
10 License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.
11 */
12
13 ////////////////////////////////////////////////////////////////////
14 //          RemoteXY include library          //
15 ////////////////////////////////////////////////////////////////////
16
17 // RemoteXY select connection mode and include library
18 #define REMOTEXY_MODE__SOFTSERIAL
19 #include <SoftwareSerial.h>
20
21 #include <RemoteXY.h>
22
23 // RemoteXY connection settings
24 #define REMOTEXY_SERIAL_RX 2
25 #define REMOTEXY_SERIAL_TX 3
26 #define REMOTEXY_SERIAL_SPEED 9600
27
28
29 // RemoteXY configurate
30 #pragma pack(push, 1)
31 uint8_t RemoteXY_CONF[] = // 88 bytes
32 { 255,1,0,153,0,81,0,16,133,0,129,0,3,5,95,6,94,70,69,82,
33   77,69,32,73,78,84,69,76,76,73,71,69,78,84,69,32,65,82,68,85,
34   73,78,79,0,67,5,5,16,91,8,16,1,51,67,5,5,28,91,8,16,
35   6,51,67,5,5,40,91,8,16,24,51,2,0,28,52,46,7,2,26,31,
36   31,79,78,0,79,70,70,0 };
37
38 // this structure defines all the variables and events of your control interface
39 struct {
40
41   // input variables
42   uint8_t switch_1; // =1 if switch ON and =0 if OFF
43
44   // output variables
45   char text_1[51]; // string UTF8 end zero
46   char text_1[51]; // string UTF8 end zero
47   char text_1[51]; // string UTF8 end zero
48
49   // other variable
50   uint8_t connect_flag; // =1 if wire connected, else =0
51
52 } RemoteXY;
53 #pragma pack(pop)
54
55 ////////////////////////////////////////////////////////////////////
56 //          END RemoteXY include          //
57 ////////////////////////////////////////////////////////////////////
58
59
60
61 void setup()
62 {
63   RemoteXY_Init ();
64
65   // TODO you setup code
66
67 }
68
69 void loop()
70 {
71   RemoteXY_Handler ();
72
73   // TODO you loop code
74   // use the RemoteXY structure for data transfer
75   // do not call delay()
76
77 }

```

III.4.2.2. Compartiment de refroidissement

Dans la partie déclaration des variables, nous allons inclure la bibliothèque du module DHT11, puis on déclare la sortie de ce capteur dans le pin A0.

Ensuite, on déclare les variables température et humidité comme étant des réels et nos pas des entiers pour avoir de la précision dans nos mesures.

Pour la sortie du ventilateur, on a choisi le pin numéro 4 de la carte Arduino Uno.

Les voyants indicateurs sont déclarés dans les pins 8 et 11.

```

42 //DHT11
43 #include <DHT.h>;
44 #define DHT11_PIN A0
45 #define DHTTYPE DHT11
46 DHT dht(DHT11_PIN, DHTTYPE);
47 float temp, hum;
48 //VENTILATEUR
49 int ventilateur=4;

77 //TEMPERATURE
78 int ledvt=8;
79 int ledrt=11;

```

Dans la partie Setup, nous allons configurer les leds comme étant des sorties ‘output’.

Pour le DHT, nous allons l’initialiser par la fonction ‘Begin’, une fonction propre à sa bibliothèque sur Arduino.

Le ventilateur est lui aussi déclaré comme sortie et mis en mode OFF par la fonction digitalWrite (...LOW).

```

131 pinMode(8,OUTPUT);
132 pinMode(11,OUTPUT);

106 //DHT
107 dht.begin();
108 //VENTILATEUR
109 pinMode(ventilateur,OUTPUT);
110 digitalWrite(ventilateur,LOW);

```

Dans la partie Loop, nous allons traduire le langage algorithmique en langage de programmation. C’est-à-dire, que si le seuil de température mesuré par la fonction (read Temperature) est supérieur (fonction if) le ventilateur s’active (fonction High) et le message s’affiche dans l’application (fonction strcpy qui permet de transmettre le message dans l’interface via le code).

Si par contre le seuil est inférieur (fonction else if) le ventilateur se désactive (Low) et un autre message s’affiche.

Les leds sont aussi présent en compte (allumage : HIGH, éteinte : LOW).

```

162 | //////////////////////////////////////////////////TEMPERATURE////////////////////////////////////
163 | temp = dht.readTemperature();
164 | hum = dht.readHumidity();
165 | Serial.println(temp);
166 | if(temp>28){
167 |   digitalWrite(ledvt,LOW);digitalWrite(ledrt,HIGH);digitalWrite(ventilateur,HIGH);
168 |   strcpy (RemoteXY.text_1, "SEUIL TEMPERATURE ATTEINT");
169 | }
170 |
171 | else if(temp<28){
172 |   digitalWrite(ledrt,LOW);digitalWrite(ledvt,HIGH);digitalWrite(ventilateur,LOW);
173 |   strcpy (RemoteXY.text_1, "SEUIL TEMPERATURE DANS LES NORMES");
174 | }

```

III.4.2.3. Compartiment d'arrosage automatique

Dans la partie déclaration des variables, on déclare la sortie du capteur d'humidité de sol dans le pin A2.

Ensuite, on déclare les variables 'sol' et 'humiditesol' comme étant des entiers et non pas des réels comme dans le compartiment précédent.

Pour la sortie de la pompe, on a choisi le pin numéro 6 de la carte Arduino Uno.

Les voyants indicateurs sont déclarés dans les pins 12 et 13.

```

53 | //HUMIDITE DE SOL
54 | #define solPin A2
55 | int sol;
56 | int humiditesol;
57 | //POMPE
58 | int pompe=6;

80 | //HUMIDITE SOL
81 | int ledrh=13;
82 | int ledvh=12;

```

Dans la partie Setup, nous allons configurer les leds comme étant des sorties 'output'.

Pour le capteur d'humidité de sol, nous allons le configurer comme entrée 'input'.

La pompe est elle aussi déclarée comme sortie et mis en mode OFF par la fonction digitalWrite (...LOW).

```

135 | pinMode(12,OUTPUT);
136 | pinMode(13,OUTPUT);

122 | //HUMIDITE DE SOL
123 | pinMode(solPin, INPUT);
124 | //POMPE
125 | pinMode(pompe, OUTPUT);
126 | digitalWrite(pompe,LOW);
---
```

Dans la partie Loop, nous allons d'abord stocker la valeur de l'humidité dans la variable sol, cette valeur est comprise entre 0 et 1023. Nous allons donc rétrécir cette plage à 0-100 grâce à la fonction de 'map', et la stocker dans la variable 'humiditesol'.

Les conditions if et else correspondent parfaitement au cahier de charges établi.

```

195 | ///////////////////////////////////////////////////HUMIDITE DE SOL////////////////////////////////////
196 | sol= analogRead(A2);
197 | humiditesol=map(sol, 0, 1023, 0, 100);
198 | // Serial.print("humidite de sol = ");
199 | // Serial.println(digitalRead(humiditesol));Serial.print("%");
200 |
201 | if(humiditesol<90){strcpy (RemoteXY.text_3, "POMPE ACTIVEE,SEUIL HUMIDITE ATTEINT ");
202 | digitalWrite(ledvh,LOW);digitalWrite(ledrh,HIGH);digitalWrite(pompe,HIGH);}
203 | else {strcpy (RemoteXY.text_3, "POMPE DESACTIVEE,SEUIL HUMIDITE DANS LES NORMES");
204 | digitalWrite(ledrh,LOW);digitalWrite(ledvh,HIGH);digitalWrite(pompe,LOW);}

```

III.4.2.4. Compartiment d'éclairage

Ce compartiment est le plus simple est le plus facile à comprendre.

On commence par déclarer la lampe dans le pin 5 de l'Arduino, puis on la configure comme étant une sortie et on l'initialise en mode lampe éteinte dans la partie Setup.

```

62 | //ECLAIRAGE
63 | int lampe=5;

115 | //ECLAIRAGE
116 | pinMode(lampe,OUTPUT);
117 | digitalWrite(lampe,LOW);

```

Dans la partie Loop, on écrit notre sous-programme avec les deux conditions de bases.

Si le switch dans la l'application est sur le mode 'ON' la lampe est allumée, sinon (else) elle est éteinte.

```

153 | ///////////////////////////////////////////////////ECLAIRAGE////////////////////////////////////
154 |
155 | if (RemoteXY.switch_1!=0)
156 | {digitalWrite(5,HIGH);}
157 | else{digitalWrite(5,LOW);}

```


III.4.2.5. Compartiment d'incendie

Dans la première partie, on déclare le capteur de flamme dans le pin A1 de l'Arduino, et le buzzer dans A5. Quant aux deux leds indicatrices, on les déclare sur les pins 10 et 9.

```

67 //CAPTEUR DE FLAMME
68 #define flamme A1
69 int degre;
70 //BUZZER
71 int buzzer=A5;

```

```

83 //FLAMME
84 int ledrf=10;
85 int ledvf=9;

```

Les leds et le buzzer sont configurés comme étant des sorties. Le capteur est lui par contre configuré comme entrée Input.

```

133 pinMode(10,OUTPUT);
134 pinMode(9,OUTPUT);

```

```

98 //CAPTEUR DE FLAMME
99 pinMode(A1,INPUT);
100 //BUZZER
101 pinMode(A5,OUTPUT);

```

Dans la partie Loop, on traduit la partie étudiée dans l'algorithme en langage Arduino. Soit, si une flamme est détectée par le capteur, le buzzer raisonne et le message d'alerte est activée. Sinon (else) le buzzer est éteint est le message d'alerte n'est plus indiqué.

Les deux Leds sont aussi programmées.

```

180 ///////////////////////////////////////////////////FLAMME////////////////////////////////////
181 // degre=analogRead(flamme);
182 if (digitalRead(flamme)==0){digitalWrite(buzzer,HIGH);
183   strcpy (RemoteXY.text_2, "FLAMME DETECTEE ATTENTION");
184   digitalWrite(ledvf,LOW);digitalWrite(ledrf,HIGH);
185
186 }
187
188
189 else {strcpy (RemoteXY.text_2, "AUCUNE FLAMME DETECTEE");
190   digitalWrite(ledrf,LOW);digitalWrite(ledvf,HIGH); digitalWrite(buzzer,LOW);}

```

III.5. Réalisation

Après avoir établi les besoins fonctionnels et non fonctionnels, expliqué le principe de fonctionnement, mis en place un algorithme et expliqué la partie Software. Nous allons dans cette dernière partie de ce travail, s'attaquer à la partie Hardware du système.

Nous allons donc tester le système complet, fabriquer une maquette qui représentera notre enceinte intelligente et effectuer le branchement final, pour ensuite finaliser notre prototype.

III.5.1. Test

Avant de fabriquer la maquette et brancher le système dedans. Nous avons pensé qu'il était plus judicieux de tester le code d'abord.

Nous avons donc branché l'ensemble des composants avec la carte Arduino, via la plaque d'essai qui a permis de dispatché l'alimentation de 5V.

Nous avons aussi respecté à la lettre le branchement, selon les pins déclaré dans le code Arduino.

Pour finir, nous avons alimenté notre système et connecté l'application dans le téléphone.

Le test a été un succès à 100%, puisque le système a respecté le cahier de charges établi et les résultats des quatre compartiments ont été similaires à ceux tracé dans l'algorithme.

Les quatre photos ci-dessous illustrent, expliquent et clarifient mieux les différents résultats de notre test (figure 3.16 (a,b,c,d)).

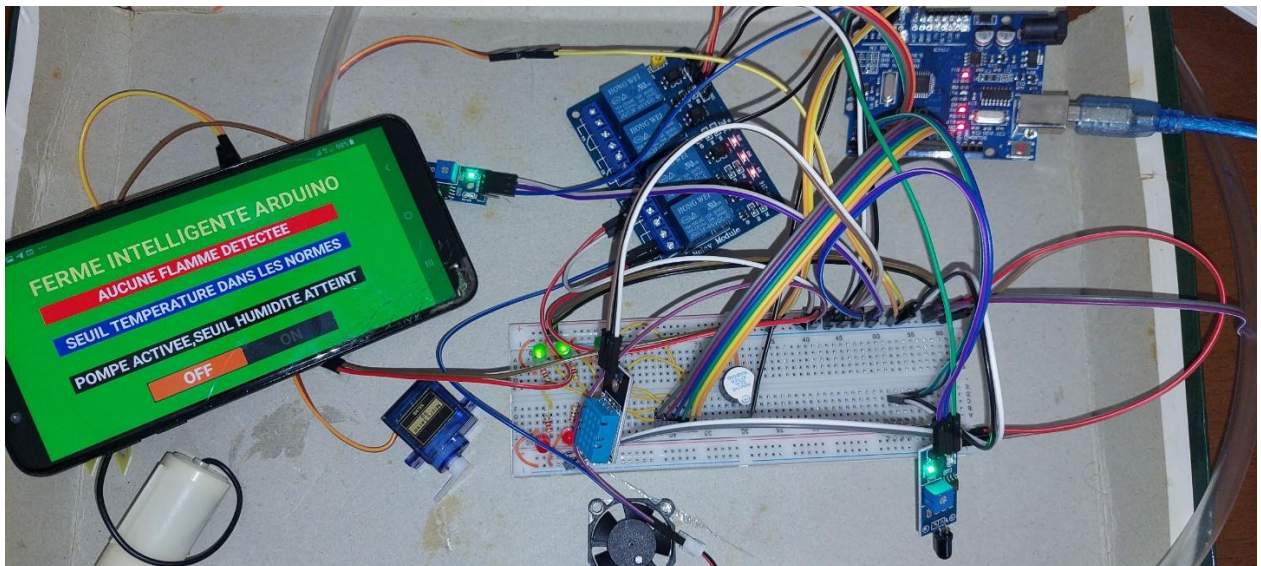


Figure 3.16(a). Photo réelle du test sous l'angle A

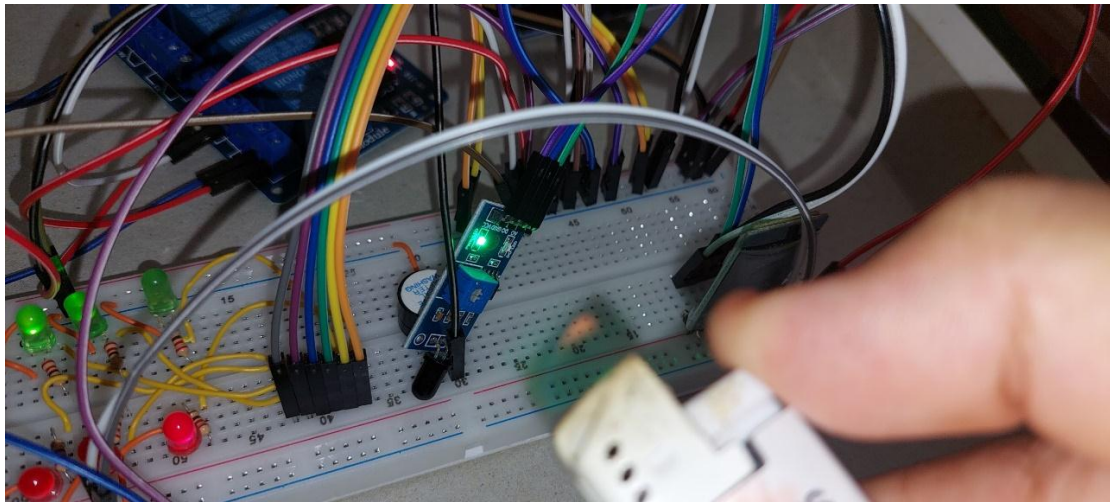


Figure 3.16(b). Photo réelle du test sous l'angle B

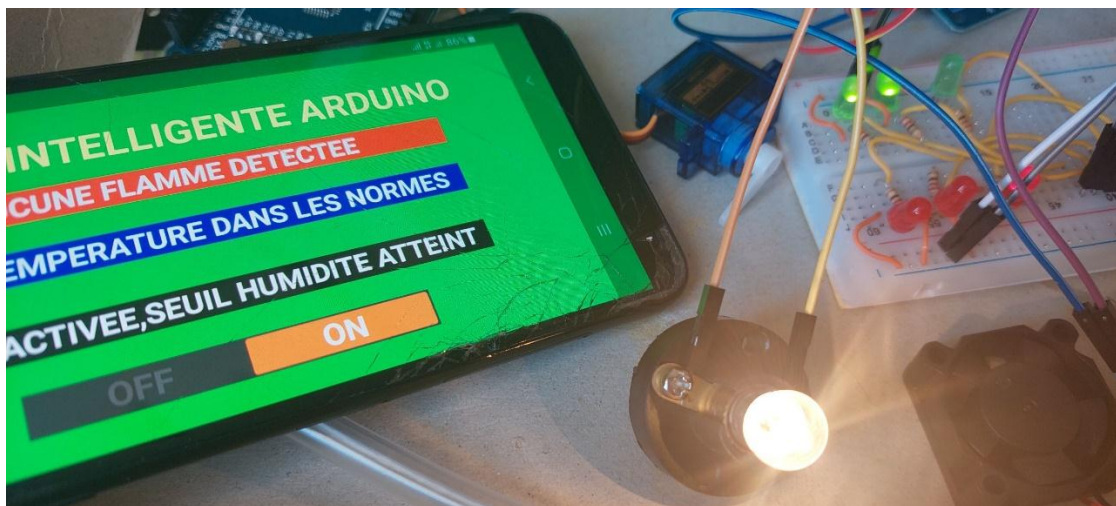


Figure 3.16(c). Photo réelle du test sous l'angle C

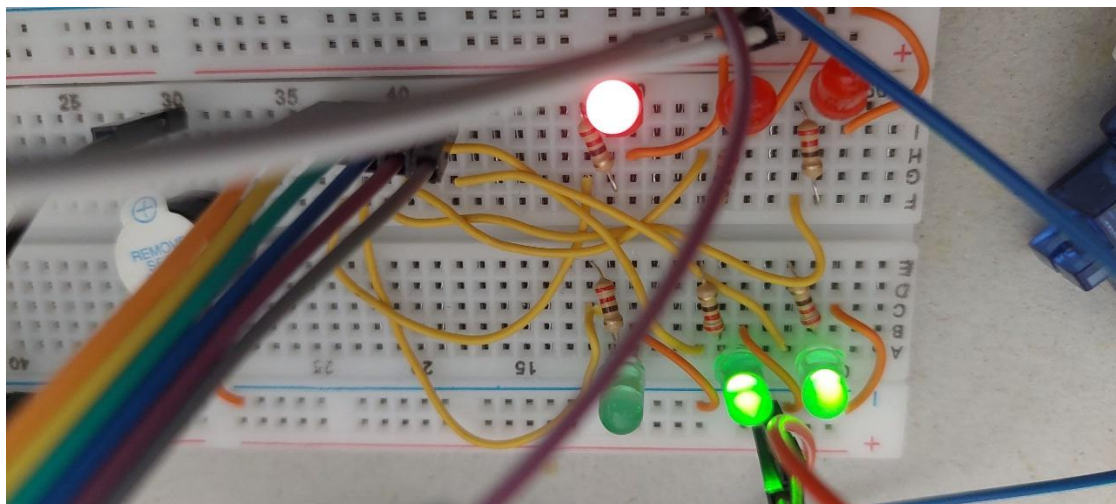


Figure 3.16(d). Photo réelle du test sous l'angle D

III.5.2. Fabrication de la maquette

Après que le test ait réussi, nous allons à présent s'attaquer à la fabrication de notre maquette de ferme intelligente.

On a donc décidé de s'orienter vers un professionnel pour un meilleur résultat. Pour cela, nous avons expliqué l'idée de base et les dimensions souhaités à un menuisier, afin qu'il nous aide à concevoir notre boîte.

La première des choses que le fabricant a réalisées, était la découpe du bois MDF de bonne qualité.

Il a donc mis une base de 30*35 cm à côté, et a ensuite découpé quatre pièces de 8 cm de hauteur afin de faire une clôture.

Il est important de signaler que la découpe a été effectuée grâce à une machine coupe bois avec une scie circulaire.

La photo suivante (figure 3.17) clarifie ce qu'on a expliqué.



Figure 3.17. Photo réelle du découpage

Une fois les pièces de bois découpées, il est à présent temps de les assembler, et cela afin de créer notre maquette.

Pour cela, nous avons utilisé une lamelleuse afin d'assembler nos pièces de bois d'une manière précise et propre.

L'assemblage se fait de telle sorte que les petites pièces soient fixées au dessus de la base pour former une clôture et avoir une forme ressemblant à une boîte.

Il est utile de préciser que cette lamelleuse est pneumatique, et qu'elle fonctionne à l'aide d'un compresseur à air.

Les deux photos ci-dessous montrent la lamelleuse et son fonctionnement.

Après avoir découpé et assemblé notre boîte, nous allons procéder à présent au perçage de la façade.

Nous avons jugé utile de faire 4 trous de taille moyenne afin de pouvoir faciliter le câblage et alimenter facilement notre maquette.

Le premier trou permettra de faire passer les fils du capteur DHT11 qui sera fixé dans la façade. Le deuxième a été modéliser afin de placer la fiche jack femelle pour ensuite placer directement la prise male et donner un côté esthétique et réel à notre boîte. Le troisième trou lui a été percé afin de faire passer le câble imprimante servant à alimenter l'Arduino et de televerser le code. Quant au quatrième et dernier trou, il permettra de brancher le capteur de flamme qui sera lui aussi à l'autre extrémité de cette façade.

Le perçage a été réalisé grâce à une perceuse électrique de 24V, et une mèche circulaire de taille 3, comme le montre la photo suivante (figure 3.18)..



Figure 3.18. Photo réelle du perçage

Voici à quoi ressemble notre maquette, après découpage de pièces de bois, assemblage et enfin perçage (figure 3.19).

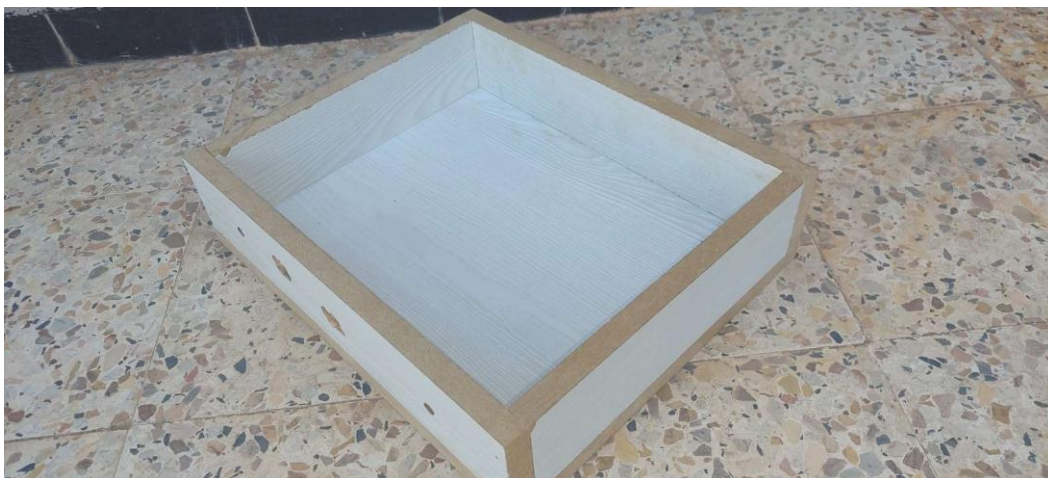


Figure 3.19. Photo réelle de la maquette après assemblage

III.5.3. Finition

Une fois la maquette fabriqué chez le menuisier, il est maintenant temps d'ajouter notre touche personnel à notre boîte de ferme intelligente.

Pour commencer, il est nécessaire de diviser notre maquette de la même façon dont nous avons fait l'étude et établi le cahier de charges.

En effet, cette maquette doit être réparti en quatre compartiments.

Le compartiment le plus spacieux de notre projet est celui de l'arrosage automatique. Pour cela, nous avons acheté une petite boîte cylindrique qui nous servira de réservoir, et dont la pompe sera fixée dedans. Le cache de ce réservoir a été percé afin de faire passer le tuyau de cette pompe ainsi que de ses deux fils électriques.

En ce qui concerne le capteur d'humidité de sol, il sera placé dans la deuxième boîte, qui sera remplisse de terre ensuite.

Pour donner un coté plus réel à notre maquette, nous avons décidé de tapisser la base avec du gazon synthétique.

Ce gazon a été fixé afin de donner une touche agricole correspondante à notre projet de ferme intelligente.

Ce tapis a été fixé à l'aide d'un pistolet à colle électrique, avec de la silicone comme matière première.

Les étapes de cette phase seront mieux détaillées avec les deux photos ci-dessous (figure 3.20 (a,b)).



Figure 3.20(a). Fixation du gazon



Figure 3.20(b).Maquette après habillage de la base avec gazon

III.5.5. Résultat final

Après avoir fini la fabrication et les finitions de la maquette, nous voici à présent devant l'ultime étape de ce long travail.

Nous allons dans cette dernière partie de ce mémoire placer les différents capteurs, actionneurs et pré-actionneurs dans la maquette et les interconnectés.

Pour cela, nous avons commencé par fixé le réservoir et le boîtier de terre dans la même ligne. Nous avons aussi fixé la pompe à l'intérieur du réservoir. Le tuyau de cette dernière est attaché par des attaches zippées à l'intérieur de la terre.

La plaque d'essai et la carte Arduino sont placées côte à côte, afin de faciliter le branchement.

Les fils sont fixés par la colle ainsi que par des attaches, afin d'éviter un débranchement et ainsi un éventuel court-circuit.

La lampe et le ventilateur sont fixés sur une planche de bois près du module relais 4 canaux avec lequel ils sont connectés.

Les leds, le buzzer et le module Bluetooth sont branchés directement à la plaque d'essai pour faciliter le câblage.

Les capteurs de température et de flamme sont fixés aux extrémités de la façade.

L'alimentation de la maquette se fait via la prise jack femelle fixée elle aussi dans la façade.

Le capteur d'humidité de sol, est lui collé dans l'une des plaques de la maquette, et sa sonde est placée à l'intérieur de la terre.

Le câble imprimante passe par l'un des trous de la façade pour alimenter l'Arduino et éventuellement modifier le code (les seuils en particulier).

Toutes les photos de notre réalisation finale sont présentées ci-dessous, avec des angles et des perspectives différents (figure 3.21 (a,b,c,d)).

Il est important de souligner, que ce branchement a été testé une nouvelle fois à l'intérieur de la maquette, et que le résultat était amplement satisfaisant.

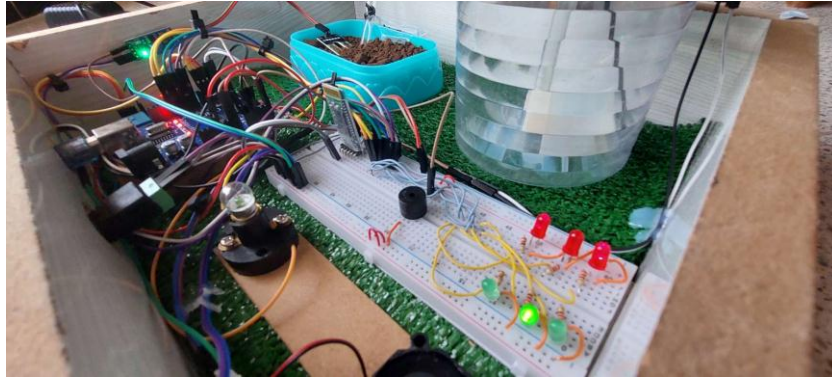


Figure 3.21(a). Photo réelle de la réalisation finale sous l'angle A

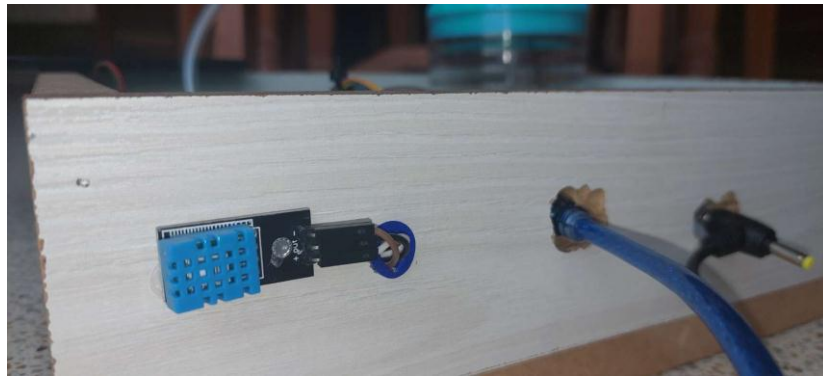


Figure 3.21(b). Photo réelle de la réalisation finale sous l'angle B



Figure 3.21(c). Photo réelle de la réalisation finale sous l'angle C

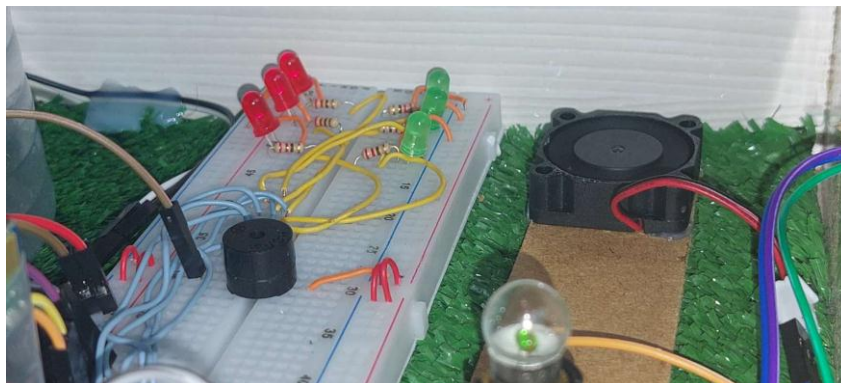


Figure 3.21(d). Photo réelle de la réalisation finale sous l'angle D

III.6. Conclusion

Dans ce chapitre nous avons élaboré une feuille de route pour la conception de notre ferme intelligente. Pour cela, nous avons commencé par faire une analyse de nos besoins ainsi qu'un schéma bloc explicatif qui a pour but de nous clarifier le principe de fonctionnement de la ferme intelligente.

Puis, nous avons exposé toutes les étapes à suivre afin de réaliser notre système en commençant par la fabrication de la maquette, en expliquant ensuite le câblage de multiples circuits jusqu'à la mise en marche du processus. Nous avons argumenté ces étapes en présentant des photos réelles prises lors de la fabrication de cette machine.

Les tests réalisés sur notre ferme aboutissent à des résultats adéquats. Cela montre que le cahier des charges que nous avons proposé est atteint.

Le résultat final de ce projet est donc satisfaisant, puisque la feuille de route établie a été respectée à la lettre avec un enchaînement logique et chronologique d'évènements et d'étapes.



***Conclusion générale et
perspectives***

Depuis maintenant plusieurs années, les nouvelles technologies agricoles sont devenues de plus en plus présents dans les fermes.

Dans ce cadre, nous avons fabriqué une ferme intelligente divisée en plusieurs compartiments réalisés en se basant sur la carte Arduino Uno qui représente le cerveau de notre réalisation.

La réalisation matérielle et logicielle de ce projet suivie d'une phase de validation et de tests ont donné des résultats attendus et satisfont notre cahier des charges.

Nous sommes unanimes pour dire que ce projet nous a permis de nous amuser grâce à la manipulation de nouveaux matériels et logiciels, tout en acquérant de meilleures connaissances dans plusieurs domaines, à savoir: l'électronique en manipulant Arduino et les différents composants et en créant une application mobile, l'informatique ainsi que l'automatique en utilisant plusieurs logiciels et en développant des programmes.

Ce projet nous a fait découvrir aussi un secteur que nous ne connaissions pas vraiment et qui nous a intéressés de plus en plus au fur et à mesure que nous approfondissions nos recherches. Cela nous a permis de recueillir des informations techniques et des explications nécessaires à la compréhension du principe de fonctionnement de certaine technologie.

Il est certain que notre projet est loin d'être terminé. Comme perspectives, nous souhaitons l'améliorer et l'enrichir avec des technologies encore plus modernes et lui apporter plus de fonctionnalités pour qu'on puisse l'exposer au grand public; parmi les expectatives

pour notre projet nous citons :

- Agrandir la maquette de notre réalisation afin d'ajouter d'avantages de technologies et de compartiments.

- Elargir le périmètre de contrôle et de commande de notre ferme intelligente, et cela en utilisant la technologie Wi-Fi.

- Augmenter la précision des différentes mesures, en utilisant des capteurs plus robustes et plus développés en matière de performance et de vitesse d'acquisition.

- Développer le système d'incendie en ajoutant d'autres fonctionnalités, telle qu'une portière qui s'ouvre automatiquement lors de la détection des flammes.



Annexes

Annexe A

PROGRAMME COMPLET DE NOTRE FERME INTELLIGENTE

```
1 //INITIALISATION DE LA LIBRARIE REMOTEXY
2 #define REMOTEXY_MODE__SOFTSERIAL
3 #include <SoftwareSerial.h>
4
5 #include <RemoteXY.h>
6
7 // RemoteXY connection settings
8 #define REMOTEXY_SERIAL_RX 2
9 #define REMOTEXY_SERIAL_TX 3
10 #define REMOTEXY_SERIAL_SPEED 9600
11
12
13 // RemoteXY configurate
14 #pragma pack(push, 1)
15 uint8_t RemoteXY_CONF[] = // 88 bytes
16 { 255,1,0,153,0,81,0,16,134,0,67,5,255,16,97,6,31,36,51,67,
17 5,255,40,97,6,31,8,51,67,5,255,27,97,6,31,6,51,129,0,255,
18 6,112,7,96,70,69,82,77,69,32,73,78,84,69,76,76,73,71,69,78,
19 84,69,32,65,82,68,85,73,78,79,0,2,0,19,51,62,7,2,26,31,
20 31,79,78,0,79,70,70,0 };
21
22 // this structure defines all the variables and events of your control interface
23 struct {
24
25     // input variables
26     uint8_t switch_1; // =1 if switch ON and =0 if OFF
27
28     // output variables
29     char text_2[51]; // string UTF8 end zero
30     char text_3[51]; // string UTF8 end zero
31
32     char text_1[51]; // string UTF8 end zero
33
34     // other variable
35     uint8_t connect_flag; // =1 if wire connected, else =0
36 } RemoteXY;
37 #pragma pack(pop)
38
39
40
41 ///////////////////////////////////////////////////////////////////LES COMPOSANTS/////////////////////////////////////////////////////////////////
42 //DHT11
43 #include <DHT.h>;
44 #define DHT11_PIN A0
45 #define DHTTYPE DHT11
46 DHT dht(DHT11_PIN, DHTTYPE);
47 float temp, hum;
48 //VENTILATEUR
49 int ventilateur=4;
50
51
52
53 //HUMIDITE DE SOL
54 #define solPin A2
55 int sol=0;
56 int humiditesol;
57 //POMPE
58 int pompe=6;
59
60
```

```

61
62 //ECLAIRAGE
63 int lampe=5;
64
65 //CAPTEUR DE FLAMME
66 #define flamme A1
67 int degre;
68 //BUZZER
69 int buzzer=A5;
70
71 //////////////////////////////////////////////////LES VOYANTS////////////////////////////////////
72 //TEMPERATURE
73 int ledvt=8;
74 int ledrt=11;
75 //HUMIDITE SOL
76 int ledrh=13;
77 int ledvh=12;
78 //FLAMME
79 int ledrf=10;
80 int ledvf=9;
81
82 void setup()
83 {
84     RemoteXY_Init ();
85
86
87     //CAPTEUR DE FLAMME
88     pinMode(A1, INPUT);
89     //BUZZER
90     pinMode(A5, OUTPUT);
91
92     //DHT
93     dht.begin();
94     //VENTILATEUR
95     pinMode(ventilateur, OUTPUT);
96     digitalWrite(ventilateur, LOW);
97
98     //ECLAIRAGE
99     pinMode(lampe, OUTPUT);
100    digitalWrite(lampe, LOW);
101
102    //HUMIDITE DE SOL
103    pinMode(solPin, INPUT);
104    //POMPE
105    pinMode(pompe, OUTPUT);
106    digitalWrite(pompe, LOW);
107
108
109
110    //LES VOYANTS
111    pinMode(8, OUTPUT);
112    pinMode(11, OUTPUT);
113    pinMode(10, OUTPUT);
114    pinMode(9, OUTPUT);
115    pinMode(12, OUTPUT);
116    pinMode(13, OUTPUT);
117
118    Serial.begin(9600);
119
120

```

```

122 void loop()
123 {
124     RemoteXY_Handler ();
125
126
127     //////////////////////////////////ECLAIRAGE////////////////////////////////////
128
129     if (RemoteXY.switch_1!=0)
130     {digitalWrite(5,HIGH);}
131     else{digitalWrite(5,LOW);}
132
133
134
135
136     //////////////////////////////////TEMPERATURE////////////////////////////////////
137     temp = dht.readTemperature();
138     hum = dht.readHumidity();
139     Serial.println(temp);
140     if(temp>28){
141         digitalWrite(ledvt,LOW);digitalWrite(ledrt,HIGH);digitalWrite(ventilateur,HIGH);
142         strcpy (RemoteXY.text_1, "SEUIL TEMPERATURE ATTEINT");
143     }
144
145     else if(temp<28){
146         digitalWrite(ledrt,LOW);digitalWrite(ledvt,HIGH);digitalWrite(ventilateur,LOW);
147         strcpy (RemoteXY.text_1, "SEUIL TEMPERATURE DANS LES NORMES");
148     }
149
150
151
152
153
154     //////////////////////////////////FLAMME////////////////////////////////////
155     // degre=analogRead(flammes);
156     if (digitalRead(flammes)==0){digitalWrite(buzzer,HIGH);
157         strcpy (RemoteXY.text_2, "FLAMME DETECTEE ATTENTION");
158         digitalWrite(ledvf,LOW);digitalWrite(ledrf,HIGH);
159     }
160
161
162
163     else {strcpy (RemoteXY.text_2, "AUCUNE FLAMME DETECTEE");
164         digitalWrite(ledrf,LOW);digitalWrite(ledvf,HIGH); digitalWrite(buzzer,LOW);}
165
166
167
168
169
170
171     //////////////////////////////////HUMIDITE DE SOL////////////////////////////////////
172     sol= analogRead(A2);
173     humiditesol=map(sol, 0, 1023, 0, 100);
174     Serial.print("humidite de sol = ");
175     Serial.println(digitalRead(humiditesol));Serial.print("%");
176
177     if(humiditesol<60){strcpy (RemoteXY.text_3, "POMPE ACTIVEE,SEUIL HUMIDITE ATTEINT ");
178         digitalWrite(ledvh,LOW);digitalWrite(ledrh,HIGH);digitalWrite(pompe,HIGH);}
179     else if(humiditesol>60) {strcpy (RemoteXY.text_3, "POMPE DESACTIVEE,SEUIL HUMIDITE DANS LES NORMES");
180         digitalWrite(ledrh,LOW);digitalWrite(ledvh,HIGH);digitalWrite(pompe,LOW);}
181
182
183 }

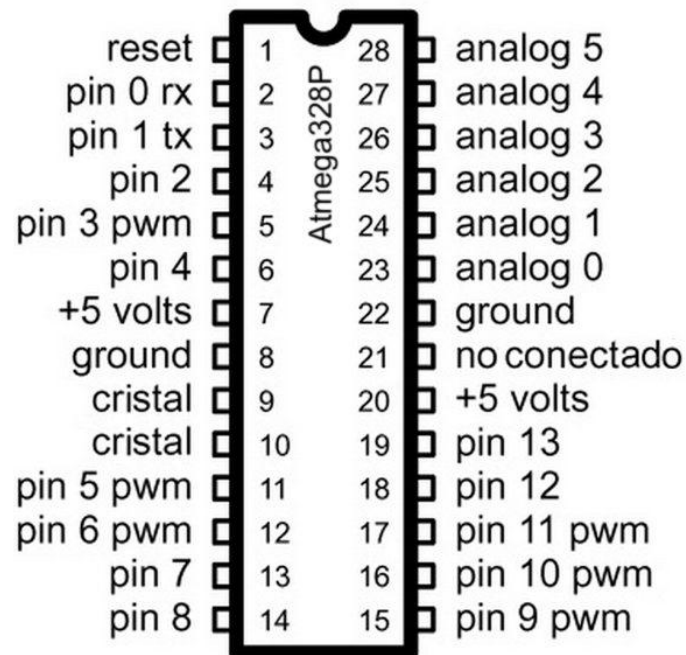
```

Annexe 2

ATmega328P

Dans cette partie nous allons présenter le microcontrôleur de l'Arduino Uno, l'ATmega328P.

Voici les Pin Out de ce microcontrôleur (les sorties).






Caractéristiques générales de l'Atmega328p

- Nombre de broches : 28 (sur le modèle utilisé en TP)
- Mémoire Flash : 32 ko (programmable par interface série)
- Mémoire Données EEPROM : 1 ko
- Mémoire RAM : 2 ko
- 32 registres de travail d'accès rapide pour l'ALU
- Ports parallèles : 3, avec 23 broches E/S
- Fréquence d'horloge : 16 Mhz (maxi tolérée = 20 Mhz)
 - donc : 16 cycles d'horloge par microseconde
- Périphériques internes
 - 6 convertisseur Analogique/Numérique 10 bits, comparateur analogique
 - 1 timer 16 bits (T1), 2 timers 8 bits (T0, T2)
 - 6 canaux PWM, 1 chien de garde (*Watch dog*)
 - SPI, USART, TWI (=I2C)
- 26 interruptions
- 5 modes d'économie d'énergie

Annexe 3

Les gammes de l'Arduino

<ul style="list-style-type: none">• Arduino Mega2560 Rev3 	<p>L'Arduino Mega dispose de 16 entrées analogiques, de 15 canaux PWM, d'un brochage similaire à Due et d'un matériel compatible avec les shields Arduino. Comme pour Due, la compatibilité logicielle avec Mega ne peut pas toujours être garantie. Le processeur est cadencé à 16 MHz et comprend 256 Ko de ROM, 8 Ko de RAM, 4 Ko d'EEPROM et fonctionne à 5 V, ce qui facilite son utilisation avec la plupart des appareils électroniques conviviaux.</p>
<p>Arduino Uno</p> 	<p>L'UNO est sans doute l'Arduino le plus populaire. Il est alimenté par un processeur Atmega328 fonctionnant à 16 MHz, comprend 32 Ko de mémoire programme, 1 Ko d'EEPROM, 2 Ko de RAM, 14 E / S numériques, 6 entrées analogiques et un rail d'alimentation de 5V et 3,3V.</p>
<p>Arduino Nano</p> 	<p>L'Arduino Nano est essentiellement un Arduino UNO réduit, le Nano est alimenté par un processeur Atmega328 fonctionnant à 16 MHz, comprend 32 Ko de mémoire programme, 1 Ko d'EEPROM, 2 Ko de RAM, 14 entrées-sorties numériques, 6 entrées analogiques et des rails d'alimentation 5V et 3,3V. Contrairement au système UNO, le Nano ne peut pas se connecter aux platines de prototypages. Les cartes Arduino Nano sont souvent l'option de carte Arduino la moins chère, ce qui les rend rentables pour les grands projets.</p>
<p>Arduino Due</p>	<p>L'Arduino Due est l'une des cartes les plus grandes et la première carte Arduino à être alimentée par un processeur ARM. Alors que l'UNO et Nano fonctionnent</p>



à 5V, la DUE fonctionne en 3,3V – il est important de le noter, car une surtension endommagerait irrémédiablement la carte. Alimenté par un Cortex-M3 ATSAM3X8E cadencé à 84 MHz, le Due dispose de 512 Ko de ROM et de 96 Ko de RAM, de 54 broches d'E / S numériques, de 12 canaux PWM, de 12 entrées analogiques et de 2 sorties analogiques.

Arduino Leonardo



L'Arduino Leonardo est une carte à microcontrôleur basée sur le composant ATmega32u4 ([datasheet](#)). Elle dispose de 20 entrées/sorties digitales (dont 7 peuvent être utilisées comme sorties PWM et 12 en tant qu'entrées analogiques), un oscillateur cristal 16 MHz, une connexion micro USB, un jack d'alimentation, un connecteur ICSP, et un bouton reset. Il dispose de tout le nécessaire pour supporter le microcontrôleur; connectez-la simplement à un ordinateur à l'aide d'un câble USB ou alimentez-la avec un adaptateur AC/DC ou une batterie pour démarrer.

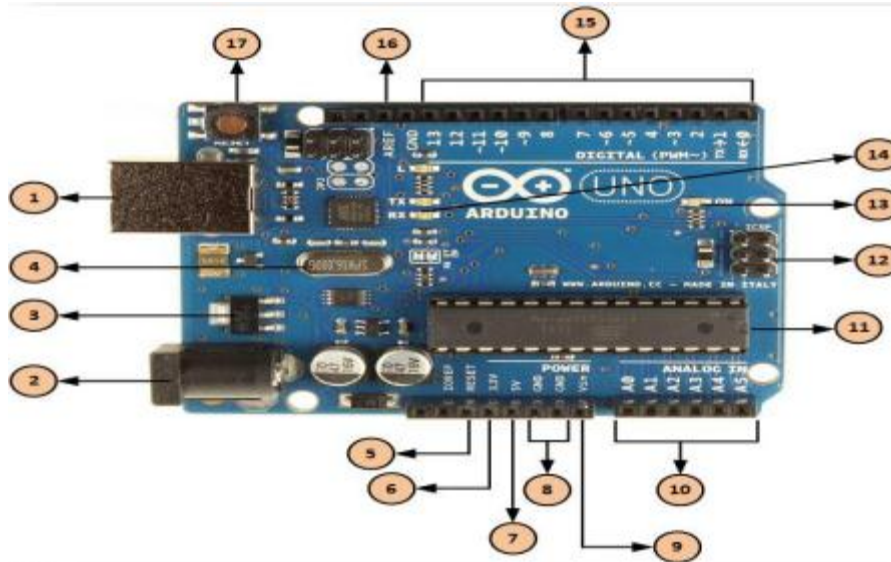
Arduino Micro









La carte Arduino Micro est basée sur le microcontrôleur ATmega32u4, c'est une version compacte et réduite de l'Arduino Leonardo. Il dispose de 20 entrées/sorties numériques (dont 7 peuvent être utilisées comme sorties PWM), 14 entrées analogiques, d'un oscillateur à quartz de 16 MHz, une connexion micro USB, un connecteur ICSP, et un bouton de reset.





Annexe 3

Structures de l'Arduino Uno



1	Le port USB a deux actions <ul style="list-style-type: none">• alimenter la carte en énergie électrique (5V).• téléverser le programme dans la carte.
2	Prise jack <ul style="list-style-type: none">• Permet de brancher une alimentation (pile, batterie, adaptateur secteur) comprise entre 5 et 12 V.• Est relié au régulateur de tension intégré à la carte.• Connexion USB.
3	Régulateur de tension <ul style="list-style-type: none">• La fonction du régulateur de tension est de contrôler la tension d'alimentation de l'Arduino pour la stabiliser à la bonne tension du microcontrôleur et de chaque élément de la carte. La tension de stabilisation est de 5 Volts sur les cartes UNO.

	<p>Oscillateur à quartz</p> <ul style="list-style-type: none"> Un oscillateur à quartz est un élément électronique qui a la particularité de posséder un quartz à l'intérieur qui vibre sous l'effet piézoélectrique. Les propriétés électromécaniques du quartz sont telles qu'on arrive à faire vibrer le quartz à une fréquence très précise. C'est l'horloge de l'Arduino, il cadence son fonctionnement. Sur le dessus du composant, on peut lire 16.000H9H. Cela signifie que la fréquence est de 16 MHz.
	<p>Arduino Reset</p> <ul style="list-style-type: none"> Permettent de redémarrer un Arduino avec un "Reset". Cela aura pour effet de réinitialiser la carte Arduino. Il existe deux manières de redémarrer un Arduino UNO : soit en utilisant le bouton "Reset" (17), soit en connectant un bouton externe sur la broche de la carte Arduino mentionnée "RESET" (5).
	<p>Broches (3.3, 5, GND, Vin)</p> <ul style="list-style-type: none"> 3.3V (6) – Broche d'alimentation de tension 3.3 Volts 5V (7) – Broche d'alimentation de tension 5 Volts La plupart des composants destinés à fonctionner avec Arduino fonctionnent bien en 3.3 Volts ou 5 Volts. GND (8) (Ground / Masse) – Il y a plusieurs broches de ce type présentes sur la carte Arduino, elles sont toutes communes et peuvent être utilisées comme masse (potentiel 0 Volts) pour les circuits. Vin (9) – Cette broche permet d'alimenter l'Arduino depuis une source de tension extérieure. Elle est reliée au circuit d'alimentation principale de la carte Arduino.
	<p>6 Broches analogiques</p> <ul style="list-style-type: none"> Permettent de brancher des capteurs et des détecteurs analogiques. La carte Arduino utilise un convertisseur analogique/numérique (CAN) pour permettre la lecture du signal par le microcontrôleur. Un signal sera converti sur 10 bits. Ces six entrées peuvent donc admettre 1024 valeurs analogiques comprises entre 0 et 5 Volts.
	<p>Microcontrôleur ATMEL ATmega328</p> <ul style="list-style-type: none"> Le microcontrôleur utilisé sur la carte Arduino UNO est un microcontrôleur ATmega328. C'est un microcontrôleur ATMEL de la famille AVR 8bits. Permet de stocker le programme et de l'exécuter. En effet c'est le cerveau de la carte.
	<p>Connecteur ICSP</p> <ul style="list-style-type: none"> Permet à l'Arduino de communiquer avec des composants externes via le protocole SPI (Serial Peripheral Interface).

	<p>Indicateur LED d'alimentation</p> <ul style="list-style-type: none"> • Un voyant qui indique l'alimentation de la carte.
	<p>LEDs TX et RX</p> <ul style="list-style-type: none"> • TX clignote lors de l'envoi des données série. • RX clignote pendant le processus de réception.
	<p>14 entrées et sorties numériques (digital)</p> <p>Ces broches peuvent être configurées pour fonctionner comme :</p> <ul style="list-style-type: none"> • Broches numériques d'entrée pour lire des valeurs logiques (0 ou 1) ou numériques. • Broches numériques de sortie pour piloter différents modules. • Remarque : Les broches étiquetées “~” peuvent être utilisées pour générer des PWM.
	<p>Broche AREF</p> <ul style="list-style-type: none"> • AREF est l'acronyme anglais de “référence analogique”. Cette broche est parfois utilisée pour définir une tension de référence externe (entre 0 et 5 Volts) comme limite supérieure pour les broches d'entrée analogiques.



Bibliographies

Bibliographies

- [1]: Chapitre 1- système automatisé- système bouclé (PRESENTATION DES SYSTEMES AUTOMATISES)
- [2]: S. BENDIDANI et A. MILOUD 'Etude d'un système automatisé (cas d'une serre), mémoire de Master en Electromécanique, Université de BELHADJ Bouchaib d'Ain-Temouchent ,2018
- [3]J.N Montagné, Centre de Ressources Art Sensitif, novembre 2006.
- [4]W. Benlahcen, "Un éclairage redondant", mémoire de fin d'études pour d'ingénieur d'état en Instrumentation, Université Batna, 2007.
- [5]N. Mechalaïne, "Etude et réalisation d'un système intelligent pour la commande d'éclairage publique", mémoire de master 2, Université Oum El Bouaghi, Juin 2018.
- [6]M. Idrissa Aliou SALL. Conception et Implémentation d'un champ connecté pour une meilleure gestion des cultures.
- [7]CHAIRI Reyane et BOUKHEMACHA Abdelmalek ' ' Etude et réalisation d'un saturomètre de pouls' ', mémoire de master en électronique, Université de Medea, 2022.
- [09] Jean- Noël, « livret Arduino en français » , centre de ressources art sensitif .
- [10] CHIBANI SAMIR et BOUKHADRA FOUJIL, ' 'Commande intelligente des différents objets d'une maison' ', mémoire de master en électronique, Université Mohamed Larbi Ben M'hidi -Oum El bouaghi , 2021.

Les sites web :

- [1] <https://elec13.wordpress.com/2016/05/03/les-systemes-automatisees-de-production-s-a-p/>
- [2] <https://www.youtube.com/watch?app=desktop&v=Ikm6eguUnys>
- [3] <http://www.climat.be/fr-be/changements-climatiques/les-effets/alimentation>.
- [4] <https://www.aps.dz/regions/148975-le-developpement-de-l-agriculture-intelligente-passe-par-le-renforcement-du-role-des-start-up>
- [5] <https://www.kws.com/fr/fr/produits/betterave-sucriere/actualites/1%27agriculture-4-0/>
- [6] [<https://www.fun-mooc.fr/fr/cours/systemes-embarques-et-objets-connectes-demarche-de-conception/>]
- [7] [<https://www.remotexy.com>]
- [8] <https://www.arduino.cc/www.alldatasheet.fr/>