

*REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
UNIVERSITE SAAD DAHLEB DE BLIDA*



MEMOIRE DE PROJET DE FIN D'ETUDES

Automatique Et Système

Réalisé par : - Merrouche Fouad –Nekdil Youcef

Sous le thème

Conception et réalisation d'un système
domotique
à base d'Arduino

Proposé par : Mr Ait Sahed Oussama

قال الله ﷻ في القرآن الكريم:

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

أَقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ ﴿١﴾ خَلَقَ الْإِنْسَانَ مِنْ عَلَقٍ ﴿٢﴾

أَقْرَأْ وَرَبُّكَ الْأَكْرَمُ ﴿٣﴾ الَّذِي عَلَّمَ بِالْقَلَمِ ﴿٤﴾ عَلَّمَ الْإِنْسَانَ

مَا لَمْ يَعْلَمْ ﴿٥﴾

Dédicace



Aucune dédicace ne pourrait assez exprimer notre gratitude et notre profond respect pour toutes les personnes qui nous ont entourés, avec grande joie nous dédions ce travail :

A la sensibilité de nos mères,

A la justice de nos pères,

A l'amour de nos frères et sœurs

A l'aide de nos amis (AbdAllah abdErrahmane et Metalbi Ali),

A nos généreux formateurs qui n'ont pas ménagés aucun effort à notre compte,

A tous ceux qui nous ont soutenus de près ou de loin,

A toute personne qui a contribué de près ou de loin pour la réalisation de ce travail.



Remerciements

Avant tout remerciement, louange à Dieu,

Ce travail est l'aboutissement d'un long cheminement au cours duquel nous avons bénéficié de l'encadrement, des encouragements et du soutien de plusieurs personnes, à qui nous tenons à dire profondément et sincèrement merci.

Nous tenons à remercier tout d'abord, notre encadrant Mr Ait Sahed Oussama pour tous les efforts qu'il a fourni en vue de nous diriger et nous aider, sa disponibilité, son écoute, ses remarques, critiques et suggestions qui ont été intitulé sur ce travail et pour le temps qu'il a bien voulu nous accorder.

Nous adressons également nos remerciements à tous les enseignants de la formation Master AS qui ont contribué à notre formation pendant cette année.

Enfin, que tous ceux qui ont participé, de près ou de loin, à la réalisation de ce travail ici nos remerciements les plus **profonds**.

Resumé

La domotique est l'une des images de la manifestation technologique moderne. Cet avancement dans ce domaine a permis à l'être humain de réjouir d'une vie pleine de gaieté, de joie et lui a faciliter les tâches de la vie. Cette domotique permettra à son propriétaire de vivre une vie sécurisée, confortable et lui permettra aussi de faire de l'économie d'énergie.

Ce mémoire qui met en évidence la « conception et réalisation d'un prototype d'une maison domotique » portera sur le fait de construire une maison équipé d'un système automatique et autonome qui dépend d'un contrôleur intelligent. Ce système assure la réaction automatique des équipements installés au sein de la maison.

Le propriétaire peut modifier les fonctionnalités de ce système selon ses besoins à travers ce que nous appelons interface graphique.

Suite aux résultats que nous avons obtenus, nous nous pouvons qu'apprécier notre domaine de recherche qui nous a permis d'acquérir de nouvelles connaissances que ce soit sur le plan automatique et électronique.

Mots clés : domotique ; sécurisée ; confortable ; économie d'énergie ; interface graphique.

Abstract

Home automation is one of the manifestation of modern technology. The advancement in this field provided for the humans an easier and joyful life.

This life automation will allow its owner to live a secure and comfortable life and to save energy.

This dissertation, which highlights “design and realization of a prototype of a home automation”, will hold the fact of building a house equipped with an automatic and autonomous system that depends on an automatic controller

This system will assure the automatic reaction of the equipment installed in the house

The owner will modify features of this system based on his needs through what we call a graphic interface

Following the results that we obtained, we can only appreciate what allowed us to acquire new knowledge on either an automatic or an electronic level.

Keywords: Home automation; secure; comfortable; save energy; a graphic interface.

ملخص

النظام الآلي للمنزل هي واحدة من صور الحدث التكنولوجي الحديث. هذا التقدم في هذا المجال قد مكن الإنسان من الاستمتاع بحياة مليئة بالبهجة والفرح وتسهيل مهام الحياة. ستتيح هذه الأنظمة الآلية المنزلية لمالكها أن يعيش حياة آمنة ومريحة، كما سيسمح له بتوفير الطاقة هذه الذاكرة التي تسلط الضوء على "تصميم وبناء نموذج أولي الأنظمة الآلية للمنازل" ستركز على واقع بناء منزل مع نظام أوتوماتيكي ومستقل يعتمد على وحدة تحكم ذكية. يضمن هذا النظام التفاعل التلقائي للمعدات المثبتة في المنزل يمكن للمالك تعديل وظائف هذا النظام وفقاً لاحتياجاته من خلال ما نسميه واجهة رسومية بعد النتائج التي حصلنا عليها، يمكننا أن نقدر مجال بحثنا الذي سمح لنا باكتساب معرفة جديدة بأنها على الخطة التلقائية والإلكترونية

TABLE DES MATIERES

Chapitre 1 : Description du projet.....	2
Introduction.....	3
1.1 Contexte du projet.....	3
1.2 Présentation du cahier des charges proposé	3
1.3 Les objectifs à réaliser.....	4
1.4 Planification du projet	4
1.5 Structure générale du système.....	5
1.6 Présentation des fonctions de domotique proposées	7
1.6.1 Fonction d'accès sécurisé à l'habitat.....	8
1.6.2 Fonction de gestion d'éclairage	8
1.6.3 Fonction de détection de mouvement.....	8
1.6.4 Fonction d'acquisition de la température et ventilation	9
1.6.5 Fonction de détection des fuites de gaz.....	9
1.6.6 Fonction de contrôle d'accès	10
1.6.7 Fonction de la gestion d'arrosage	10
1.6.8 Fonction de détection de niveau d'eau dans le réservoir.....	11
Conclusion	12
Chapitre 2 : Etude préalable du projet.....	13
Introduction.....	14
2.1 Généralités sur la domotique.....	14

2.1.1	C'est quoi la domotique.....	14
2.1.2	Pourquoi la domotique ?.....	17
2.2	Etude des solutions matérielles.....	18
2.2.1	Choix de la carte programmable	18
2.2.2	Carte ARDUINO.....	21
2.2.3	Les capteurs utilisés	26
A	Capteur d'humidité et de température DHT11	26
B	Capteur de gaz et de fumée (MQ-2)	28
C	Capteur de mouvement PIR HC-SR501	29
D	Le flotteur	31
2.2.4	Les actionneurs utilisés	31
A	Servomoteurs.....	31
B	Electrovanne plastique	35
C	Les relais	36
D	Module Afficheur LCD	37
E	Le Buzzer	39
F	Ventilateur	41
G	Téléphone portable (Samsung - B313E)	42
H	Clavier matriciel 4x4.....	42
I	bouton poussoir.....	43
J	Les LEDs	44
K	RTC DS3231.....	46
L	Alimentation.....	47
2.3	Etude de la partie logicielle du projet	48
2.3.1	Plateforme de programmation Arduino.....	49
2.3.2	Plateforme de programmation Visual Studio	52
	Conclusion.....	56
Chapitre 3 : Phase de la réalisation du système domotique		58

	Introduction	59
3.1	Réalisation du prototype de la maison intelligente	59
3.2	Réalisation du système domotique.....	61
3.2.1	Le schéma théorique du montage.....	62
3.2.2	Instruction du montage.....	62
3.2.3	Réalisation de l'interface graphique.....	64
3.2.4	Fonctionnement de l'interface graphique avec la carte arduino	65
3.2.5	Quelques scénarios de notre projet	67
A	Le scénario du système antivol	67
B	Le scénario du contrôle du temps d'arrosage.....	68
C	Le scénario de la gestion de la sécurité de la porte	69
D	Le scénario de surveillance du niveau d'eau dans le réservoir.....	70
3.3	Cout des composants	71
	Conclusion.....	71
	Conclusion générale.....	72
	Références Bibliographiques	73

LISTE DES FIGURES

Figure 1.1: Schémas synthétique de la première partie du système	6
Figure 1.2: Schéma synthétique de la deuxième partie du système	7
Figure 1.3: Schéma synthétique de la fonction d'accès sécurisé à l'habitat.....	8
Figure 1.4: Gestion d'éclairage	8
Figure 1.5: Détection de mouvement	9
Figure 1.6: l'acquisition de la température.....	9
Figure 1.7: Détection des fuites de gaz.....	10
Figure 1.8: Contrôle d'accès par code	10
Figure 1.9: Contrôle d'accès via une télécommande.....	10
Figure 1.10: Gestion d'arrosage.....	11
Figure 1.11: surveillance de niveau de la citerne.....	11
Figure 2.12: le principe de fonctionnement de la domotique	16
Figure 2.13: les tâches principales associées à la domotique.....	17
Figure 2.14: Carte Arduino UNO.....	19
Figure 2.15: Carte RaspBerry Pi B+.....	19
Figure 2.16: Carte Beaglebone black.....	19
Figure 2.17: Exemples des cartes Arduino	21
Figure 2.18: Capteur DHT11	27
Figure 2.19: Branchement du capteur DHT11 avec la carte Arduino	28
Figure 2.20: Branchement du MQ-2 avec la carte Arduino.....	29
Figure 2.21: Module détecteur de présence ou de mouvement PIR HC-SR501.....	30
Figure 2.22: Branchement du capteur de présence HC-SR501 avec la carte Arduino	30

Figure 2.23: Capteur niveau d'eau (Flotteur)	31
Figure 2.24: Fonctionnement de capteur niveau d'eau	31
Figure 2.25: Vue de l'extérieur et de l'intérieur d'un servomoteur	32
Figure 2.26: Caractéristiques du servomoteur utilisé	34
Figure 2.27: branchement du servomoteur avec l'Arduino	34
Figure 2.28 Electrovanne plastique 8 - 12V	35
Figure 2.29: Principe de fonctionnement du relais.....	36
Figure 2.30: : Modulerelais 5V 1 canal	37
Figure 2.31: Nomenclature du connecteur de l'afficheur LCD	38
Figure 2.32: Afficheur LCD 16x2 avec I2C	39
Figure 2.33: Branchement de LCD 16x2 avec la carte Arduino	39
Figure 2.34: 5 V active Buzzer	40
Figure 2.35: branchement du buzzer avec la carte arduino	41
Figure 2.36: Ventilateur 5V d'un PC.....	41
Figure 2.37: Samsung B313E	42
Figure 2.38: clavier matriciel	42
Figure 2.39: structure interne du clavier matriciel	43
Figure 2.40: Branchement du clavier avec la carte arduino	43
Figure 2.41: Boutons poussoirs	44
Figure 2.42: Branchement du bouton poussoir avec Arduino	44
Figure 2.43: LEDs de différentes couleurs.....	45
Figure 2.44: Branchement des LEDs avec Arduino	45
Figure 2.45: RTC DS3231	46
Figure 2.46: Branchement du RTC DS3231 avec l'Arduino	47
Figure 2.47: Générateur de courant 30V - 5A.....	48

Figure 2.48: logo d'interface de développement Arduino	49
Figure 2.49; Interface de programmation Arduino	49
Figure 2.50: Barre de boutons Arduino	50
Figure 2.51: HyperTerminal de l'Arduino(Moniteur Série)	50
Figure 2.52: Structure d'un programme en Arduino EDI	51
Figure 2.53: Téléversement du programme sous EDI vers la carte Arduino	52
Figure 2.54; Logo visual studio	52
Figure 2.55: Fenêtre de design	53
Figure 2.56: Fenêtre de programmation	53
Figure 2.57: interface de design VS	55
Figure 2.58: Programme pour allumer une LED via une application	56
Figure 3.59: la maison intelligente (prototype) sur Vite Architecture	59
Figure 3.60: l'image réelle de l'intérieur de la maison	60
Figure 3.61: l'image réelle de l'extérieur de la maison	61
Figure 3.62: Le schéma théorique du montage	62
Figure 3.63 : La page d'accueil de l'interface graphique sous Visuel studio	65
Figure 3.64 : Le scénario du système antivol	68
Figure 3.65 : Le scénario du contrôle du temps d'arrosage	69
Figure 3.66 : Le scénario de la gestion de la sécurité de la porte	70
Figure 3.67 : Le scénario de surveillance du niveau d'eau dans le réservoir	71

LISTE DES TABLEAUX

Tableau 2.1: Comparaison technique : Arduino Méga, Beaglebone, RaspBerry Pi	20
Tableau 2.2: Présentation des caractéristiques de la carte Arduino Méga 2560	22
Tableau 2.3: Présentation des broches d'alimentation de la carte Arduino Méga 2560	23
Tableau 2.4: Différentes broches entrées/sorties numériques	24
Tableau 2.5: Caractéristiques du servomoteur utilisé	34

Introduction générale

L'évolution de la technologie et du mode de vie nous permet aujourd'hui de prévoir des espaces de travail et de logement mieux adaptés. De même, La majorité des individus, et plus particulièrement les personnes âgées, passent beaucoup de leur temps à domicile, d'où l'influence considérable de l'habitat sur la qualité de vie. L'amélioration du sentiment de sécurité et de confort dans l'habitat apparaît donc comme une tâche d'une grande importance sociale.

La domotique ou encore la maison intelligente est définie comme une résidence équipée de plusieurs technologies qui vise à assister l'habitant dans les situations diverses de la vie domestique en améliorant le confort et simplifient un certain nombre de tâches.

Elle regroupe les technologies de l'électronique, de l'automatique, de l'informatique et des télécommunications permettant d'améliorer le confort (commande à distance d'appareils ou équipement, gestion d'arrosage...), la sécurité (protection contre les intrusions, détection d'incendie...), la communication et la gestion d'énergie (gestion du climatiseur, d'éclairage ...).

C'est dans ce cadre que se situe notre projet de fin d'études intitulé **«Conception et réalisation d'un système domotique à base d'arduino»**. Ce projet a pour objectif de réaliser un système domotique intelligent permettant le pilotage, la commande et la surveillance des différents dispositifs disponibles dans une maison, ainsi que la conception et la réalisation d'un prototype d'une maison où sera installé notre système.

Pour se faire, nous allons diviser notre mémoire en trois parties :

Le premier chapitre est dédié à la description du projet : le contexte, la présentation du cahier des charges, les objectifs et contraintes, planification du projet et les fonctions domotique proposées.

Le deuxième chapitre entamera une généralité sur la domotique. Ensuite une description de la partie matérielle et logicielle du projet en identifiant le choix du cœur du notre système, et les organes constituant notre système domotique. Finalement nous allons énumérer les outils logiciels utilisés lors de la réalisation du projet.

Dans le troisième chapitre nous détaillerons les phases de la mise en place de notre système domotique.

Nous finirons ce rapport par une conclusion générale récapitulative des différentes phases de notre travail, signalant les côtés bénéfiques du projet et énonçant les perspectives du travail élaboré.

Chapitre 1

Description du projet

- **Contexte du projet**
- **Présentation du cahier des charges proposé**
- **Les objectives à réaliser**
- **Planification du projet**
- **Structure générale du système domotique**
- **Présentation des fonctions de domotique proposées**

Introduction

L'objectif de ce chapitre est de mettre notre projet dans son contexte général. Tout d'abord, nous commençons par faire une présentation du sujet en détaillant son contexte et ses fonctionnalités. Ensuite, nous allons présenter le cahier des charges. Puis nous citerons les objectifs. Enfin nous allons détailler notre méthodologie de travail suivie pour réaliser notre projet.

1.1 Contexte du projet

Depuis quelques années, les solutions de domotique se démocratisent de plus en plus. En effet, à une époque où presque tout le monde dispose d'un ordinateur il devient important de maîtriser la consommation énergétique et de vivre dans une maison plus sécurisée et confortable, il est temps de pouvoir commander les équipements domestiques (TV, climatisation, arrosage, etc.) via un ordinateur

C'est dans ce contexte que se déroule notre projet de fin d'études «**Conception et réalisation d'un système domotique à base d'arduino**» qui consiste à mettre en place un système domotique permettant de piloter, commander, programmer et superviser de façon simple et confortable à l'aide d'une interface graphique les différentes fonctionnalités de domotique. Ces fonctions concernent principalement le confort (commande à distance d'appareils ou équipements...), la sécurité (accès sécurisé à l'habitat, détection des fuites de gaz...), l'économie d'énergie (gestion du climatiseur/ventilateur, Contrôle du temps d'allumage de la TV...). En effet, notre système sera implémenté dans un prototype que nous avons construit en bois.

1.2 Présentation du cahier des charges proposé

- Réaliser en bois un prototype d'une maison.
- Établir les fonctions de la domotique suivantes :
 - ✓ Gestion des alarmes et des alertes en temps réel : alerte de température, alerte de fuite gaz, etc...
 - ✓ Contrôle d'accès à l'habitat et au garage ;
 - ✓ Contrôle du temps d'allumage de la TV dans la chambre des enfants ;
 - ✓ Gestion d'arrosage ;
 - ✓ système antivol ;
 - ✓ Détection et affichage de niveau d'eau dans le réservoir d'eau ;

- ✓ Développement d'une application pour le pilotage du système domotique.

1.3 Les objectifs à réaliser

Notre projet de fin d'études a pour objectifs de :

- ✓ Réaliser un prototype d'une maison en boit.
- ✓ Installer les différents composants à savoir les capteurs, relais et actionneurs.
- ✓ Etudier la commande des composants afin d'établir les organigrammes.
- ✓ Créer le programme pour piloter les différents composants du système, ce programme devra être implanté dans la carte de traitement.
- ✓ Développer une application graphique pour piloter, commander, et superviser les différents composants à l'intérieur d'habitat.

1.4 Planification du projet

- **L'équipe de travail**

La bonne réussite d'un projet s'appuie sur un ensemble des piliers qui est : l'équipe de travail, son organisation et les choix technologiques à adopter.

Notre équipe se compose de 2 étudiants :

- ❖ **Merrouche Fouad**

- Etudiant en deuxième année Master Automatique et Système à la Faculté des Sciences et Techniques de Saad dahleb Blida
- Titulaire d'une Licence en Génie Electrique option Automatique à l'université Saad Dahleb Blida

Contact : fouadmerrou@gmail.com

- ❖ **Nekdil Youcef**

- Etudiant en deuxième année Master Automatique et Système à la Faculté des Sciences et Techniques de Saad dahleb Blida
- Titulaire d'une Licence en Génie Electrique option Automatique à l'université Saad Dahleb Blida

Contact : youssefnekdil@gmail.com Notre projet a été encadré par le Docteur :

Mr. Ait Sahed Oussama

Contact: aitsahed.oussama@hotmail.com

1.5 Structure générale du système

Pour faciliter les tâches et pour donner plus de valeur à notre projet, nous proposons de diviser notre système domotique en deux parties indépendantes :

Une partie dédiée au fonctionnement des équipements intérieurs de la maison, cela comprend les fonctionnalités suivantes :

- ✓ Gestion de l'éclairage ;
- ✓ Gestion de la climatisation et la ventilation ;
- ✓ Gestion des alertes en temps réel : alerte de température, alerte de gaz ;
- ✓ Détection d'incendie et détection des fuites des gaz combustibles et des gaz étouffants ;
- ✓ Contrôle du temps d'allumage de la TV dans la chambre des enfants ;
- ✓ Contrôle d'accès sécurisé à l'habitat

L'avantage de ce premier système c'est qu'il pourra être piloté à distance à l'aide d'une interface graphique.

La fonction du contrôle d'accès sécurisé à l'habitat sera indépendante des autres fonctions qui seront commandées à distance par l'intermédiaire de l'application.

La fonction d'accès sécurisé à l'habitat est assurée en introduisant un code d'accès correct à l'aide d'un clavier.

- La deuxième partie s'occupera de la conception et du développement d'une application sur PC qui va amener les fonctionnalités suivantes suivre l'état du niveau d'eau dans le réservoir et piloter les équipements liés à l'arrosage ainsi qu'à l'ouverture et fermeture du garage.

La commande des organes de cette deuxième partie se fera grâce à une application Windows.

La fonction d'accès sécurisé à l'habitat est assurée à l'aide d'une télécommande qui nous permet de contrôler l'accès au garage d'une manière plus sécurisée.

Le choix de diviser notre système en deux parties a pour but d'avoir une idée plus claire sur chacune d'elles, et aussi d'optimiser les fonctionnalités de chacune et de les placer dans les endroits les plus adéquats.

- Armoire 1 (plaque d'essai 1) : armoire de commande du premier système (intérieur de l'habitat).

- Armoire 2 (plaque d'essai 2) : armoire de commande du deuxième système (arrosage, réservoir, garage).

Dans la suite nous présentons des schémas synthétiques de chaque système avec les descriptions :

Partie 1

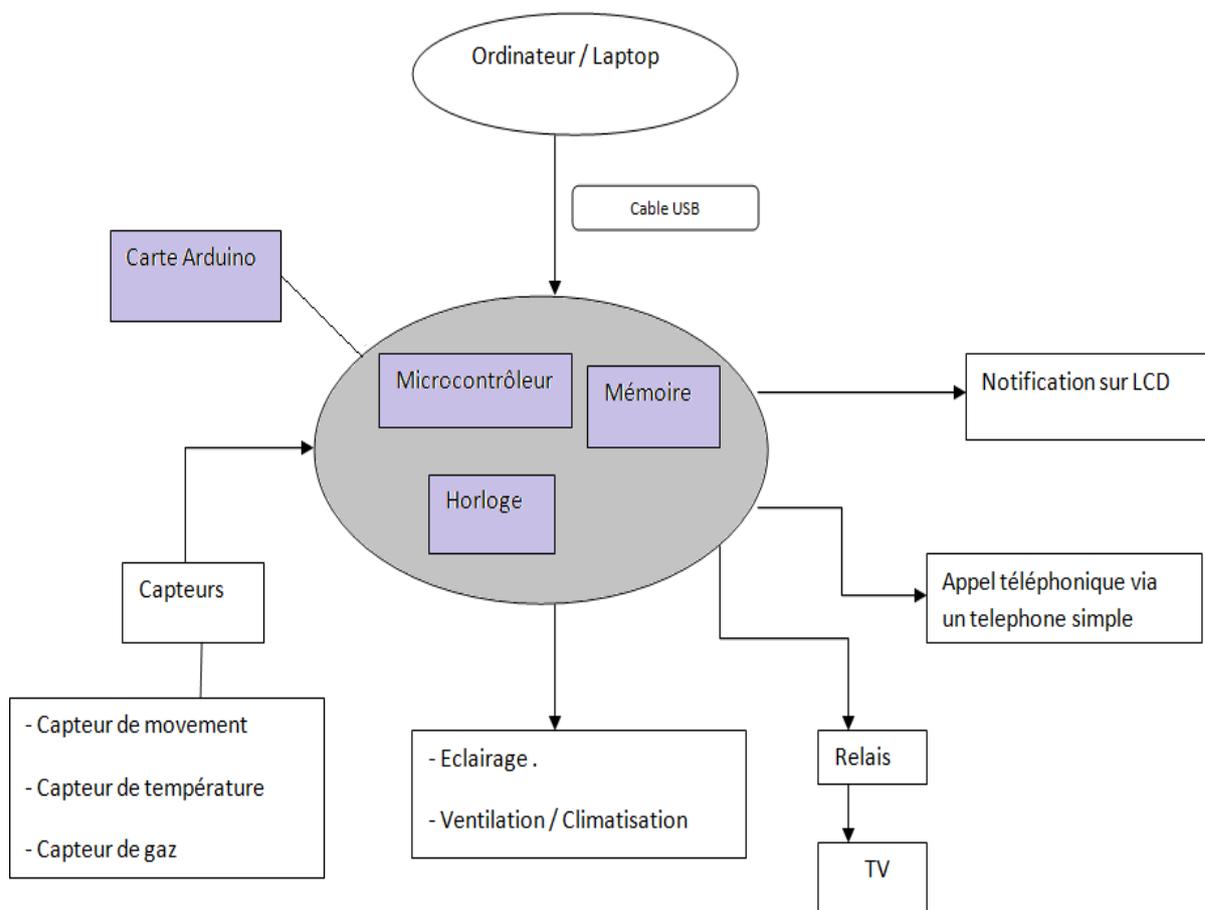


Figure 1.1 : Schémas synthétique de la première partie du système

L'ordinateur est relié à la carte programmable à travers un câble USB qui sert à transmettre les informations ou renvoyer les données. Les données extérieures des capteurs sont envoyées à la carte programmable. S'il faut effectuer une action, l'ordinateur envoie une instruction à la carte qui agit via un relais ou directement sur l'appareil électrique et envoie des notifications sur l'écran LCD en cas des alertes (température, gaz, fumée, etc.).

Les données relevées par les capteurs sont affichées sur une interface graphique que l'on va créer, sur laquelle aussi figureras les liens permettant de commander le système_domotique.

Partie 2

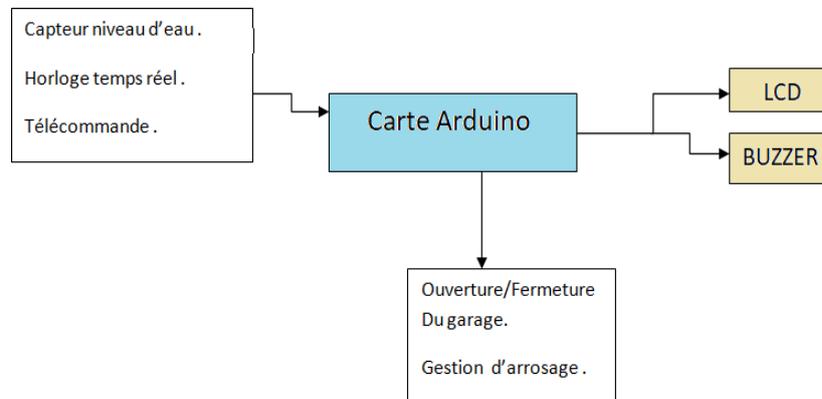


Figure 1.2 : Schéma synthétique de la deuxième partie du système

Les données des capteurs sont envoyées à la carte programmable, qui a son tour agit sur les actionneurs électriques et affiche une notification en cas d'alerte (par exemple réservoir d'eau vide). La carte de traitement est reliée avec l'ordinateur pour pouvoir configurer les différentes composantes de notre domotique (par exemple changer le temps d'ouverture de la vanne d'arrosage).

1.6 Présentation des fonctions de domotique proposées

Une habitation est constituée de plusieurs fonctions autonomes (climatisation, ventilation, éclairage, alarme, incendie, accès, arrosage...) qui peuvent être intégrés grâce à la domotique. Il existe un grand nombre d'applications possibles qui diffèrent suivant les besoins des habitants et les fonctionnalités du système installé.

Suivant les clients, les applications seront orientées vers les économies d'énergie, le confort ou la sécurité. Chaque application est donc différente. C'est le concept des scénarios qui peuvent être configurés pour gérer les différents capteurs et actionneurs de l'installation qui va permettre à chacun de faire « vivre » son installation suivant ses propres habitudes de vie.

Pour notre système domotique, nous avons choisis comme fonctions à établir :

1.6.1 Fonction d'accès sécurisé à l'habitat

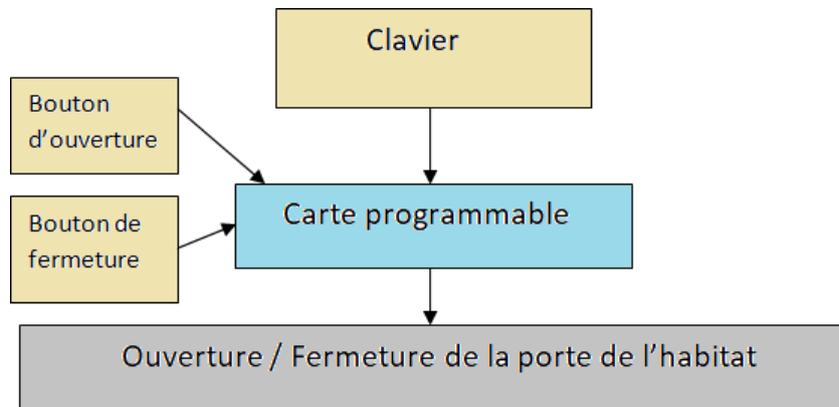


Figure 1.3 : Schéma synthétique de la fonction d'accès sécurisé à l'habitat

Cette fonction permet à l'utilisateur de contrôler l'accès à son habitat par la saisie d'un code de sécurité à l'aide d'un clavier. La carte de traitement traite ce code, s'il est correct la porte s'ouvre automatiquement et elle se ferme dès qu'on appuie sur le bouton de fermeture.

1.6.2 Fonction de gestion d'éclairage :

Cette fonction permet à l'utilisateur de faire le contrôle de l'éclairage (allumer/éteindre) de 5 lampes au sien de l'habitat à distance par une application sur PC.



Figure 1.4 : Gestion d'éclairage

1.6.3 Fonction de détection de mouvement :

Le principe consiste à détecter la présence d'une personne dans une portée par un détecteur de mouvement. Le détecteur permet aussi de capter une transition ou un changement brutal de l'environnement (Objet en mouvement). Nous l'avons intégré

dans un système comme protection contre les intrusions dans l'habitation. Il est possible de l'utiliser dans la salle de bain pour allumer la pièce en cas de présence d'une personne et s'éteint automatiquement en son absence.



Figure 1.5 : Détection de mouvement

1.6.4 Fonction d'acquisition de la température et ventilation :

L'acquisition de la température se fait via un capteur de température pour garantir la température ambiante adéquate à l'intérieur de l'habitat à l'aide de l'activation d'un ventilateur/climatiseur en cas de besoin. La valeur de la température sera affichée sur l'application de commande.



Figure 1.6 : l'acquisition de la température

1.6.5 Fonction de détection des fuites de gaz :

Cette fonction permet de détecter les fuites de gaz via un capteur de gaz. Il est apte à détecter le GPL, le butane, le propane, le méthane, l'alcool, l'hydrogène. Une alarme est ajoutée pour informer l'utilisateur en cas de danger, plus, d'une

notification sur l'écran LCD et le déclenchement du ventilateur.



Figure 1.7: Détection des fuites de gaz

1.6.6 Fonction de contrôle d'accès :

Le contrôle d'accès est une technique qui consiste à soumettre l'entrée d'un habitat ou d'une zone sécurisée à une autorisation préalable d'accès. Dans notre projet nous allons utiliser deux techniques de contrôle d'accès :

Le contrôle d'accès à l'habitat par un code ;

Le contrôle d'ouverture et de fermeture de garage via une télécommande.



Figure 1.8 : Contrôle d'accès par code



Figure 1.9 : Contrôle d'accès via une télécommande

1.6.7 Fonction de la gestion d'arrosage :

A l'aide de cette fonction, la vanne d'arrosage va s'ouvrir automatiquement à un moment spécifique pour une durée bien précise.

On peut configurer le temps d'arrosage via l'application développée sur PC



Figure 1.10 : Gestion d'arrosage

1.6.8 Fonction de détection de niveau d'eau dans le réservoir :

A l'aide de cette fonction, l'utilisateur peut suivre l'état du niveau d'eau dans le réservoir. Si l'eau dans le réservoir arrive à un niveau bas, l'écran LCD affiche une notification et le Buzzer se déclenche.

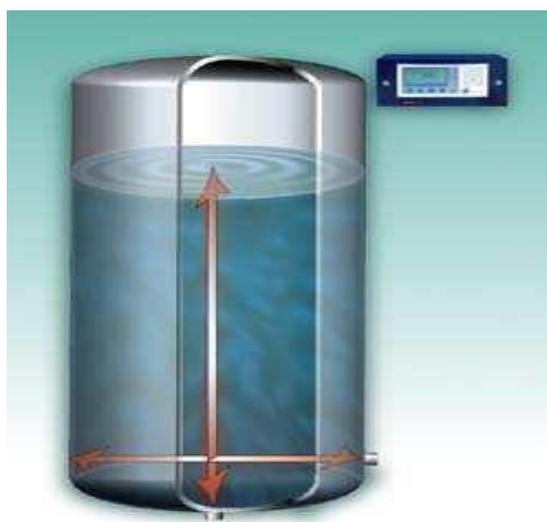


Figure 1.11 : surveillance de niveau de la citerne

Conclusion

Tout au long de ce chapitre, nous avons situé le contexte général de notre projet, nous avons élaboré le cahier des charges, la présentation des objectifs à réaliser, la planification du projet et la structure générale de notre système domotique, ainsi que les différentes fonctions proposées pour la réalisation de notre système.

Dans ce qui suit, le deuxième chapitre décrira de façon plus détaillée, la partie matérielle et logicielle utilisée pour élaborer notre projet.

Chapitre 2

Etude du projet

- Généralité sur la domotique
- Etude des solutions matérielles
- Etude de la partie logicielle

Introduction

La réalisation de notre projet nécessite une étude approfondie sur certaines notions qui touchent non seulement le contexte général du projet, mais aussi sa réalisation, cette étude sera développée en trois axes.

Nous aborderons dans un premier lieu des généralités sur la domotique. Ensuite dans le deuxième axe qui est consacré à l'étude matérielle (HARD), nous allons décrire le choix de la carte programmable à adopter pour notre système, et nous allons présenter les différentes solutions matérielles utilisées, ensuite une présentation de l'unité d'alimentation qui est indispensable pour alimenter notre système.

On termine ce chapitre par une étude de la partie logiciel (SOFT).

2.1 Généralités sur la domotique

2.1.1 C'est quoi la domotique

C'est quoi la domotique ? C'est la première question que nous nous sommes posés avant de s'immerger dans notre projet.

A Origines de la Domotique

Le mot domotique a été introduit dans le dictionnaire « le petit Larousse » en 1988. Le mot "domotique" vient de la contraction de deux mots, « domus » (la maison en latin) et « télématique » (techniques qui associent l'informatique et les télécommunications).

On associe souvent le début des travaux domotiques aux années 1970, voire 1980, avec les problématiques énergétiques dues aux crises pétrolières qui ont considérablement affecté le domaine de la construction et de l'exploitation du bâtiment. En effet, avant ces crises, le coût de l'énergie était négligeable face au coût des matériaux et des travaux d'isolation thermique. On privilégiait alors des techniques de construction simples, peu onéreuses, au détriment de la qualité de l'enveloppe thermique, comme l'emploi de structures bétons fabriquées sur place, avec des ponts thermiques marqués, et des vitrages fins sur des menuiseries peu isolantes. Les bâtiments construits dans ce cadre sont aujourd'hui qualifiés d'énergivores.

Lorsque le prix de l'énergie augmenta de manière importante, on privilégiait alors une isolation plus contrainte des bâtiments neufs. Cependant, la gestion du parc existant était

problématique, car coûteuse. C'est ainsi qu'apparurent les premières solutions de régulation automatique, ou plus communément appelées Gestion Technique du Bâtiment (GTB). Ces solutions reposent sur des moyens de communications entre automates (chaudières, ventilation, etc.) et capteurs, et permettent d'optimiser l'empreinte énergétique du bâtiment en fonction des besoins, présence des occupants, apports naturels, température extérieure, etc.

Ces solutions, bien qu'onéreuses, furent petit à petit déclinées et proposées à l'habitat au milieu des années 1990. Les fonctions étaient alors identiques à celles proposées pour les bâtiments de grande taille : gestion du chauffage, de la ventilation et de la climatisation (CVC), de l'éclairage, des ouvrants (volets roulants et stores). Elles proposaient alors aux habitants une promesse de confort fonctionnel, principalement par la centralisation des dispositifs de contrôle, comme la possibilité de fermer tous les volets d'une façade, ou de gérer le chauffage pièce par pièce.

En parallèle, et en utilisant les mêmes bases technologiques, des solutions d'alarme à intrusion se sont commercialisées, développant un segment « sécurité ». Ces solutions interagissent parfois avec le système domotique, par exemple pour fermer les volets lorsque l'alarme est armée, ou simuler des présences par l'actionnement de lumières lorsque l'habitant est absent.

Depuis le milieu des années 1990, un autre segment, orienté sur la micro-informatique et les loisirs numériques, se développe. Cette nouvelle apparition marque en particulier l'introduction de l'informatique dans l'habitat et l'apparition des supports numériques, les cédéroms, puis les DVD et internet.

Ainsi, aujourd'hui, gestion de l'habitat, sécurité, réseaux de communication et loisirs numériques esquissent le paradigme de domotique (Jeuland, 2005). Nous esquisserons les contours et spécificités de ce paradigme, caractérisés par un nombre de définitions importantes ayant évolué au cours du temps.

B Définitions et principe de fonctionnement

La domotique est l'ensemble des techniques de l'électronique, de physique du bâtiment, d'automatismes, de l'informatique et des télécommunications permettant d'automatiser des bâtiments individuels ou collectifs. La domotique vise à apporter des fonctions de confort, de gestion d'énergie, de sécurité et de communication que l'on peut retrouver dans les maisons, les hôtels, les lieux publics...

Un système domotique s'agit d'un système qui permet de communiquer avec de

télécommandes ou des boutons poussoirs afin de rendre le contrôle de la maison plus facile. Ils permettent de commander d'un simple geste une ou plusieurs actions (Exemples : gestion d'éclairage, ouverture de la porte, gestion d'arrosage...).

Donc le principe d'un système domotique consiste à faire communiquer un ensemble des équipements électriques d'un habitat entre eux, on parle alors de bâtiment intelligent ou de bâtiment communicant. Cette notion d'automatisation sur laquelle est basée la domotique n'est pas seulement appliquée à la maison.

Le principe de la domotique consiste aussi de programmer et contrôler à distance ou localement le comportement d'appareils que l'on aura intégrés dans un réseau de capteurs et d'actionneurs.

Un système domotique nécessite, pour son fonctionnement, de pouvoir capter une information et d'actionner par la suite une commande. Pour cela, il faut pouvoir faire transférer l'information entre les deux parties d'un système domotique (partie commande et partie opérative).

Le schéma ci-dessous, permet de mieux comprendre la circulation des informations dans une maison « intelligente ».

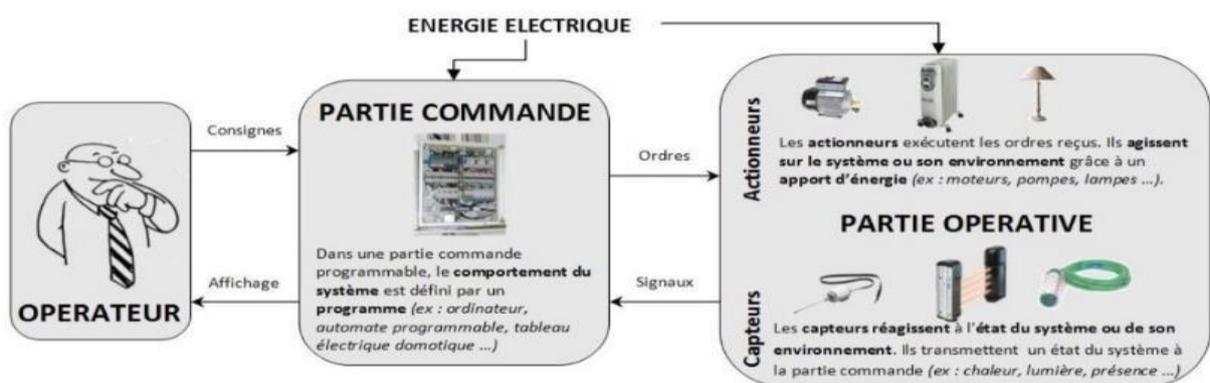


Figure 2.12 : le principe de fonctionnement de la domotique »

2.1.2 Pourquoi la domotique ?

Nous allons dans cette partie présenter les applications que l'on peut avoir dans la domotique.



Figure 2.13 : les tâches principales associées à la domotique.

On peut les regrouper en 3 principaux domaines (la sécurité, le confort et la gestion d'énergie):

A La sécurité :

En termes de sécurité, la domotique permet de :

Prévenir les risques provenant de l'extérieur (intrusion, cambriolage...) comme ceux provenant de l'intérieur (accidents domestiques).

Surveiller à distance nos locaux ou notre habitation depuis un Smartphone, une Tablette ou un ordinateur.

Être avertis d'un incident (alarmes techniques) en temps réel par SMS et/ou par un appel.

B La gestion d'énergie

La domotique permet de diminuer jusqu' à 10 % des factures d'énergie. Grâce aux automatismes et à des capteurs, les équipements électriques inter-reliés pilotent au plus juste la consommation énergétique (chauffage, éclairage, eau, ventilation, etc.), tout en gardant sous contrôle le confort des zones occupées.

Le but principal de la domotique est d'éviter le gaspillage en supprimant les dépenses inutiles. Les systèmes de régulation permettent de maîtriser la consommation d'électricité, de gérer le

chauffage et la production d'eau chaude sanitaire, avec un niveau de confort optimal. Un détecteur de présence placé dans chaque pièce, par exemple, commande instantanément l'allumage ou l'extinction des éclairages, la mise en route ou l'arrêt du chauffage, etc.

Au jardin par exemple, l'arrosage s'automatise et le détecteur crépusculaire se charge d'allumer les lumières dès la tombée de la nuit.

C Le confort

La domotique facilite la vie. Vous pouvez, en effet, effectuer plusieurs tâches routinières en même temps. Regarder un film ? Une seule pression sur la télécommande et la télévision est allumée, les lumières sont tamisées et les volets baissés.

Une simple pression suffit avant de partir travailler pour que toutes les lumières s'éteignent, que les appareils électriques se coupent et que le chauffage ne soit automatiquement baissé. On désigne ce regroupement de commandes par le terme 'scénarios'. Vous pouvez créer vous-même, adapter et exécuter de manière automatique ou non ces scénarios à des moments déterminés ou par une pression sur un bouton.

2.2 Etude des solutions matérielles :

Nous allons aborder dans cette partie le choix de la carte programmable, ainsi les différents organes utilisés pour élaborer notre projet et l'unité d'alimentation quant à elle, est indispensable pour alimenter notre système.

2.2.1 Choix de la carte programmable

Durant notre recherche sur les cartes programmables, nous avons constaté qu'au cours des dernières années, de nombreuses cartes de développement modulaire sont apparues. Chaque carte présente des avantages et des inconvénients. Elles se différencient par leurs fonctionnalités, leurs complexités et leurs prix.

Dans cette partie nous allons citer quelques types de cartes programmables les plus connus et ses caractéristiques dont le but de sélectionner la plus optimale pour notre projet. On peut citer:

Carte Arduino : c'est un circuit intégré à base d'un microcontrôleur programmable qui peut analyser et produire des signaux électriques de façon à effectuer des tâches précises.



Figure 2.14 : Carte Arduino UNO



RaspBerry Pi B+ : un mini-ordinateur à base du système d'exploitation Linux.

Figure 2.15 : Carte RaspBerry Pi B+

Beaglebone : une mini-carte électronique qui possède les fonctionnalités d'un ordinateur basique



Figure 2.16 : Carte Beaglebone black

Le tableau suivant présente une comparaison technique entre ces cartes :

Nom de la carte	Arduino Méga	Beaglebone	Raspberry Pi B+
Naissance	2009	2008	2012
Prix(DA)			
Processeur	ATmega2560 8bits	Sitara335x sur un cortex M8	BCM2835 sur un ARM11 -CPU intégrée
RAM	8 KB SRAM	512 MB DDR3L	512 MB SDRAM
Mémoire Flash	256 KB	Micro SD - 4G	Micro SD – 32 G
Consommation	50 mA (0.5 W)	210 à 450mA (2.5 W max)	700mA (3,5 W)
Ethernet	non	Oui	Oui
USB	1	1	4
Sortie Vidéo	non	Non	HDMI,RCA
Développement	Langage de programmation Arduino	Environnement BoneScript.Langages Phyton, Scratch, Squeak	Langages Scratch, Squeak

Tableau 2.1: Comparaison technique : Arduino Méga, Beaglebone, RaspBerry Pi

Nous remarquons qu'en termes de mémoire vive ou mémoire externe, la capacité de RaspBerry est plus importante que les autres cartes. Ainsi Le RaspBerry possède des connectiques (HDMI, port Ethernet, port USB...) de plus par rapport aux autres cartes.

Du côté de la carte Arduino, si la puissance disponible est évidemment bien moindre que celle délivrée par les autres cartes, le très vaste écosystème qui l'entoure et le langage de programmation très simple qui l'accompagne concourent au succès phénoménal qu'elle a rencontré dans le monde.

De plus, d'après une recherche réalisée, nous avons trouvé que le RaspBerry PI ne possède pas d'entrée analogique, contrairement à l'Arduino. Si l'on doit mesurer une ou des tensions, comme il est le cas pour l'acquisition des données des capteurs sous forme des tensions analogiques, l'Arduino se montre très précieux.

Pour des applications complexes, l'idéal est de travailler avec les deux systèmes et de faire communiquer RaspBerry Pi et Arduino pour profiter des avantages de chacune.

Pour notre projet, la carte Arduino sera suffisante et mieux adaptée à nos besoins. De plus, son

rapport performance- prix est plus qu'abordable.

2.2.2 Carte ARDUINO

Arduino est un circuit imprimé en matériel libre (dont les plans de la carte elle-même sont publiés en licence libre mais dont certains composants sur la carte, comme le microcontrôleur par exemple, ne sont pas en licence libre) sur lequel se trouve un microcontrôleur qui peut être programmé pour analyser et produire des signaux électriques, de manière à effectuer des tâches très diverses comme la domotique (le contrôle des appareils domestiques, éclairage, chauffage...), le pilotage d'un robot, etc.



Figure 2.17 : Exemples des cartes Arduino

La carte Arduino repose sur un circuit intégré associée à des entrées et sorties qui permettent à l'utilisateur de brancher différents types d'éléments externes :

- **Côté entrées** : des capteurs qui collectent des informations sur leur environnement comme la variation de température via une sonde thermique, le mouvement via un détecteur de présence ou un accéléromètre, le contact via un bouton-poussoir, etc.
- **Côté sorties** : des actionneurs qui agissent sur le monde physique telle qu'une petite lampe qui produit de la lumière, un moteur qui actionne un bras articulé, etc.

La plateforme Arduino se présente sur plusieurs séries à savoir : Arduino Nano, Arduino Lilypad, Arduino DUE, Arduino UNO et Arduino Méga 2560, cette dernière carte sera le cœur de notre système domotique.

➤ **Arduino Méga 2560**

L'Arduino Méga 2560 est une carte électronique basée sur le microcontrôleur ATmega2560. Elle dispose de 54 broches numériques d'entrée / sortie (dont 15 disposent d'une sortie PWM), 16 entrées analogiques, un résonateur céramique (Quartz) à 16 MHz, une connexion USB, une prise d'alimentation, un connecteur ICSP, et un bouton de réinitialisation. Il contient tout

le nécessaire pour soutenir le microcontrôleur, tout simplement le connecter à un ordinateur avec un câble USB ou allumez-le avec un adaptateur ou batterie pour commencer.

A Caractéristiques

Parmi les caractéristiques de la carte Arduino Méga 2560, on trouve :

Microcontrôleur	ATmega2560
Tension de fonctionnement	5V
Tension d'entrée (recommandé)	7-12V
Tension d'entrée (limite)	6-20V
Digital I / O Pins	54 (dont 15 fournissent sortie PWM)
Broches d'entrée analogiques	16
DC Courant par I / O Pin	40 mA
Courant DC pour 3,3 Pin	50 mA
Mémoire Flash	256 KB (ATmega2560)
SRAM	8 KB (ATmega2560)
EEPROM	4 KB (ATmega2560)
Fréquence d'horloge	16 MHz

Tableau 2.2 : Présentation des caractéristiques de la carte Arduino Méga 2560

B Alimentation

L'Arduino Méga 2560 peut être alimentée via la connexion USB ou avec une alimentation externe. Dans le premier cas, la source d'alimentation est automatiquement sélectionnée.

L'alimentation Externe (non-USB) peut provenir d'un adaptateur AC –DC ou batterie. L'adaptateur peut être connecté en branchant une prise de centre positif de 2,1 mm dans le connecteur jack de la carte Les fils en provenance d'un bloc de piles ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelées GND (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation.

Les broches d'alimentation sont détaillées dans le tableau suivant :

VIN	La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). On peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
5V	La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation V_{in} via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.
3.3V	Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitent cette tension au lieu du 5V). L'intensité maximale disponible sur cette broche est de 50mA.
GND	Broche de masse (ou 0V)

Tableau 2.3 : Présentation des broches d'alimentation de la carte Arduino Méga 2560

C Mémoire :

La carte Arduino Méga 2560 à 256 Ko de mémoire FLASH pour stocker le programme (dont 8Ko également utilisés par le bootloader). Elle a également 8 ko de mémoire SRAM (volatile) et 4 Ko d'EEPROM (non volatile - mémoire qui peut être lue à l'aide de la librairie EEPROM.h).

D Entrées et sorties numériques :

Chacune des 54 broches numériques de la carte Arduino Méga 2560 peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions *pinMode* (), *digitalWrite* () et *digitalRead* () du langage Arduino. Ces broches fonctionnent en 5V.

Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de (pull- up) (déconnectée par défaut) de 20-50 KΩ. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction *digitalWrite (broche, HIGH)*. De plus, certaines broches ont des fonctions spécialisées

Communication Série	<ul style="list-style-type: none"> ✓ Port Série Serial 0 : 0 (RX) and 1 (TX) ; ✓ Port Série Serial 1: 19 (RX) and 18 (TX) ; ✓ Port Série Serial 2: 17 (RX) and 16 (TX) ; ✓ Port Série Serial 3: 15 (RX) and 14 (TX) ; <ul style="list-style-type: none"> • Utilisées pour recevoir (RX) et transmettre (TX) les données sériées de niveau TTL. • Les broches 0 (RX) and 1 (TX) sont connectées aux broches correspondantes du circuit intégré ATmega8U2 programmé en convertisseur USB-vers-série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.
Interruptions externes	<ul style="list-style-type: none"> ✓ Broche 2 (interruption 0) ; ✓ Broche 3 (interruption 1) ; ✓ Broche 18 (interruption 5) ; ✓ Broche 19 (interruption 4) ; ✓ Broche 20 (interruption 3) ; ✓ Broche 21 (interruption 2) ; <p>Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. (l'instruction <i>attachInterrupt ()</i>).</p>
Impulsion PWM	<p>Impulsion PWM (Largeur d'impulsion modulée). Les Broches 0 à 13 fournissent une impulsion PWM 8-bits à l'aide de l'instruction <i>analogWrite ()</i>.</p>
SPI (Interface Série Périphérique)	<ul style="list-style-type: none"> ✓ Broche 50 (MISO) ; ✓ Broche 51 (MOSI) ; ✓ Broche 52 (SCK) ; ✓ Broche 53 (SS) ; <ul style="list-style-type: none"> • Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI. • Les broches SPI sont également connectées sur le connecteur ICSP qui est physiquement compatible avec les cartes Uno, Duemilanove et Diecimila.
I2C	<ul style="list-style-type: none"> ✓ Broche 20 (SDA) ; ✓ Broche 21 (SCL) ; <ul style="list-style-type: none"> • Supportent les communications de protocole I2C ou interface TWI (Two Wire Interface - Interface "2 fils"), disponible en utilisant la librairie <i>Wire/I2C</i> (ou <i>TWI -Two-Wire interface "2 fils"</i>). • Noter que ces broches n'ont pas le même emplacement que sur les cartes Uno, Duemilanove ou Diecimila.

LED	Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.
-----	--

Tableau 2.4 : Différentes broches entrées/sorties numériques

E Broches analogiques :

La carte Arduino Méga 2560 dispose de 16 entrées analogiques, chacune pouvant fournir une mesure d'une résolution de 10 bits (c.-à-d sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction *analogRead* () du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction *analogReference* () du langage Arduino.

Note : les broches analogiques peuvent être utilisées en tant que broches numériques.

F Autres broches :

Il y a deux autres broches disponibles sur la carte :

- **AREF** : Tension de référence pour les entrées analogiques (si différent du 5V).
Utilisée avec l'instruction *analogReference* ().
- **RESET** : Mettre cette broche au niveau BAS entraîne la réinitialisation (ou le redémarrage) du microcontrôleur. Typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

G Communications :

La carte Arduino Méga 2560 dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs. Elle dispose de 4 UARTs (Universal Asynchrones Receiver Transmitter ou émetteur-récepteur asynchrone universel en français) pour communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX).

Un circuit intégré ATmega8U2 sur la carte assure la connexion entre cette communication série de l'un des ports série de l'ATmega 2560 vers le port USB de l'ordinateur qui apparaît comme un port COM virtuel sur l'ordinateur.

Le code utilisé pour programmer l'ATmega8U2 utilise le driver standard USB COM, et aucun autre driver externe n'est nécessaire. Cependant, sous Windows, un **fichier.inf** est requis.

Le logiciel Arduino inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur, qui permet la communication série avec la carte Arduino. Les deux LEDs RX et TX sur la carte clignote lorsque les données sont transmises via le circuit intégré ATmega8U2 utilisé en convertisseur USB-vers- série et la connexion USB vers l'ordinateur (mais pas pour les communications série sur les broches 0 et 1).

2.2.3 Les capteurs utilisés :

A Capteur d'humidité et de température DHT11 :

Afin de garder de la fraîcheur au sein de notre habitat, nous allons intégrer dans notre système domotique un sous-système d'acquisition de température avec le capteur de température DHT11. La valeur de la température acquise sera renvoyée vers l'utilisateur via un afficheur LCD et à travers une application sur son ordinateur.

- **Description**

Le DHT11 est composé de deux parties, un capteur d'humidité capacitif et un capteur de température à base de NTC. Il contient également un circuit électronique élémentaire qui effectue la conversion analogique vers le numérique et qui débite un signal numérique proportionnel à la température et l'humidité mesurée par le capteur. La technologie Utilisée par le capteur DHT11 garantie une grande fiabilité, une excellente stabilité à long terme et un temps de réponse très rapide.



Figure 2.18 : capture DHT-11

- **Caractéristiques du DHT11 :**

Alimentation : 3-5.5V DC.

Consommation : comprise entre 0.5 mA et 2.5 mA.

Signal de Sortie : Signal Numérique.

Plage de Mesure : - Humidité : 20-95%RH ; Température : 0-50°C ;

Précision : Humidité $\pm 5\%$ RH ; Température $\pm 2.0^\circ\text{C}$

Résolution : Humidité 1%RH ; Température 0.1°C

Période de mesure : 2s.

Dimensions : 12x15.5x5.5mm.

- **Branchement du DHT11 :**

Le DHT11 est composé de 4 broches espacées de 2,45mm ce qui permet de le brancher facilement avec notre carte de traitement, mais en réalité seulement 3 broches sont utilisées.

Le brochage du capteur DHT11 est le suivant :

VCC : 3.5 à 5.5V.

GND : Masse 0V.

Data : données (Celle-ci doit être reliée à une entrée de la carte électronique utilisée).

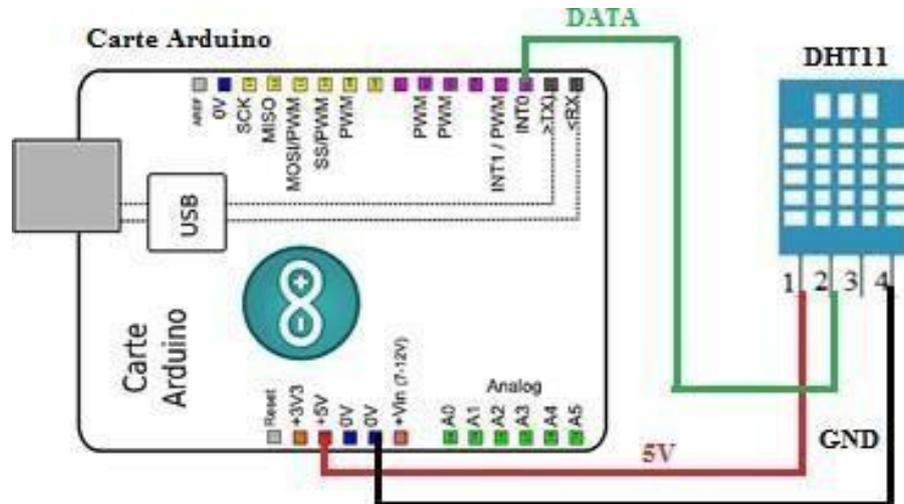


Figure 2.19 : Branchement du capteur DHT11 avec la carte Arduino

B Capteur de gaz et de fumée (MQ-2)

Pour détecter les fuites des gaz combustibles et des gaz étouffants au sien de notre habitat, nous allons utiliser le capteur de gaz et de fumée (MQ-2), qui nous garantit la sécurité des habitants contre la suffocation et les incendies.

• Description :

Le MQ-2 est un semi-conducteur permettant de détecter la présence des gaz combustibles et des gaz étouffants dans l'air qu'ont des concentrations de 300 ppm (Partie par million) à 10000 ppm, tel que GPL (Gaz de pétrole liquéfié), Butane, Propane, Méthane. Hydrogène, et le fumé.

Ce capteur possède 2 type de sorties :

Une sortie analogique, dont la tension de sortie varie en fonction de la concentration globale d'élément détecté (tous les gaz à la fois).

Une sortie numérique, réglable à l'aide d'un potentiomètre à l'arrière du composant. L'utilisateur règle le seuil de détection du module, et ce dernier se déclenche si la concentration dépasse le seuil.

Le MQ-2 offre un temps de réponse rapide et une Haute sensibilité ajustable via un

potentiomètre.

- **Caractéristiques**

Alimentation : 5 V DC.

Consommation : 0.16 A.

Signal de sortie : analogique et numérique.

Température de fonctionnement : -20 à 50°C.

Temps de réponse : moins d'une seconde.

Dimensions : 42 x 24 x 34 mm.

- **Branchement du MQ-2**

Le MQ-2 possède 4 broches qui sont :

La broche n°1 : la broche d'alimentation (VCC). La broche n°2 : la masse (GND).

La broche n°3 : la sortie numérique du MQ-2. La broche n°4 : la sortie analogique du MQ-2.

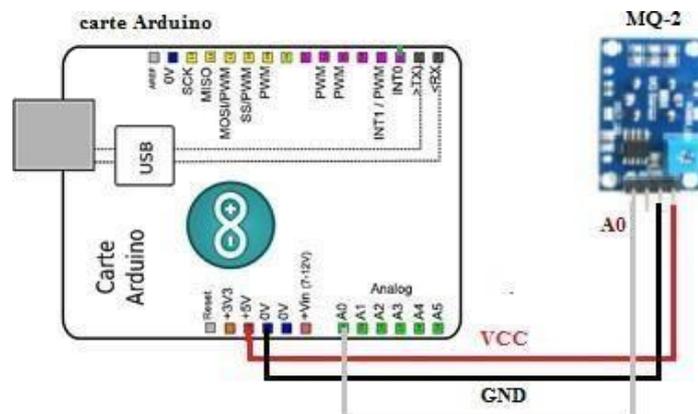


Figure 2.20 : Branchement du MQ-2 avec la carte Arduino

C Capteur de mouvement PIR HC-SR501

Le principe consiste à détecter la présence d'une personne dans une portée par un détecteur de mouvement. Il sera utilisé dans le garage pour allumer la pièce en cas de présence d'une personne et s'éteint automatiquement en son absence, on peut l'intégrer aussi dans un système de protection contre les intrusions dans une habitation.

- **Description**

Ce capteur PIR est idéal pour détecter des mouvements ou une présence dans les applications

de domotique ou de robotique.

Il est simple, facile et sa mise en œuvre est rapide.



Figure 2.21 : Module détecteur de présence ou de mouvement PIR HC- SR501

- **Caractéristiques**

Les caractéristiques techniques du module détecteur de présence HC-SR501 sont les suivantes:

Plage de Tension de travail : DC 4.5 V-20 V Consommation de courant : < 60uA

Dimension : 3.2 cm x 2.4 cm x 1.8 cm (environ)

Tension de Sortie : Haut/Bas niveau signal : 3.3 V TTL sortie Distance de détection : 3--7M (peut être ajusté)

Plage de détection : < 100 °

Temps de retard : 0.3 S ~ 18 S (peut être ajustée, par défaut 5 S +-3%) la température de travail :-20-+ 80 °C

- **Branchement du PIR HC-SR501**

Le capteur HC-SR501 possède 3 broches qui sont :

VCC : à connecter au 5V pour l'alimentation.

OUT : signal du sortie du module

GND : à mettre à la masse

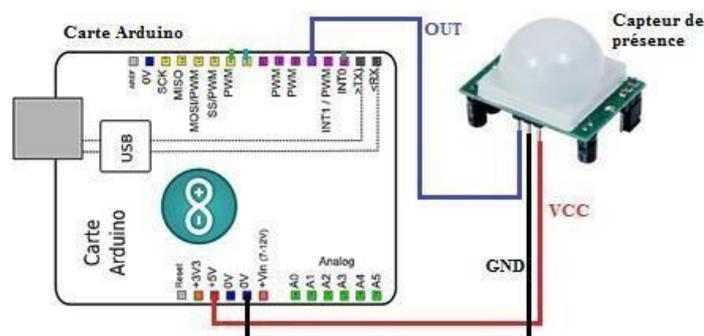


Figure 2.22 : Branchement du capteur de présence HC-SR501 avec la carte Arduino

D Le flotteur

Concernant le flotteur nous l'avons fabriqué nous-même. Ce dernier fonctionne grâce à une bille qui touche un détecteur fin de course, lorsque le château d'eau ou la citerne commence à se vider l'arduino reçoit un signal par le capteur fin de course.

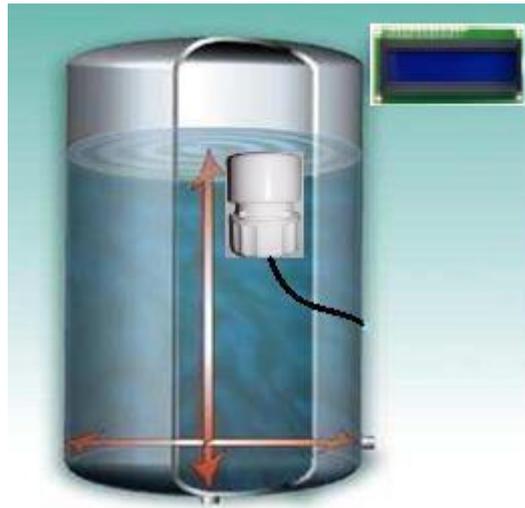


Figure 2.23 : Capteur niveau d'eau (Flotteur)

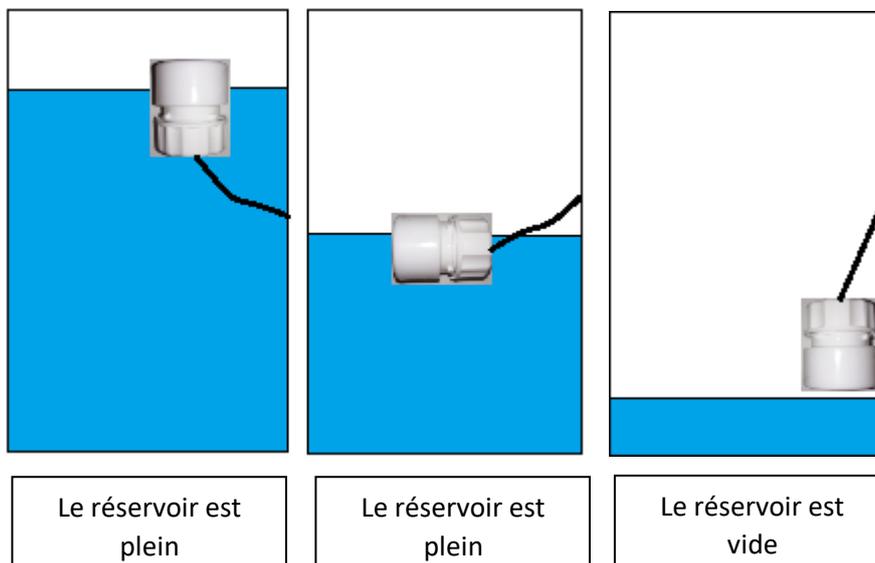


Figure 2.24 : Fonctionnement de capteur niveau d'eau

2.2.4 Les actionneurs utilisés :

A Servomoteurs :

Pour motoriser notre porte principale et notre garage, on a pensé à utiliser des servomoteurs vu de leur souplesse et simplicité de commande.

Les servomoteurs servent en principe à actionner les parties mobiles d'un système. Ils sont prévus pour être commandés facilement en position ou en vitesse. En effet, ils sont équipés d'un système d'asservissement basé sur un potentiomètre rotatif qui sert de capteur de rotation.

C'est un ensemble mécanique et électronique comprenant :

- Un moteur à courant continu de petite taille.
- Un réducteur en sortie de ce moteur diminuant la vitesse mais augmentant le couple.
- Un potentiomètre (faisant fonction de diviseur résistif) qui génère une tension variable, proportionnelle à l'angle de l'axe de sortie.
- Un dispositif électronique d'asservissement.
- Un axe dépassant hors du boîtier avec différents bras ou roues de fixation.

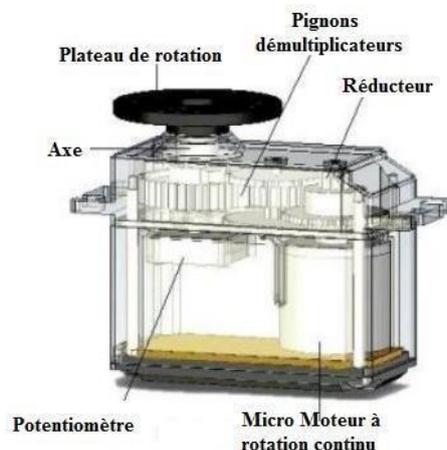


Figure 2.25 : Vue de l'extérieur et de l'intérieur d'un servomoteur

• Fonctionnement

Les servomoteurs sont commandés par l'intermédiaire d'un câble électrique à 3 fils qui permettent d'alimenter le moteur et de lui transmettre des ordres de positions sous forme d'un signal codé en largeur d'impulsion plus communément appelés PWM (Pulse Width Modulation ou Modulation de Largeur d'Impulsion) ou RCO (Rapport Cyclique d'Ouverture). Cela signifie que c'est la durée des impulsions qui détermine l'angle absolu de l'axe de sortie

et donc la position du bras de commande du servomoteur. Le signal est répété périodiquement, en général toutes les 20 millisecondes, ce qui permet à l'électronique de contrôler et de corriger continuellement la position angulaire de l'axe de sortie, cette dernière étant mesurée par le potentiomètre.

- **Type des servomoteurs**

Les servomoteurs sont répartis en deux types : Servomoteur à rotation continue et le servomoteur à rotation angulaire. A la différence du servomoteur standard (à rotation angulaire), le servomoteur à rotation continue tourne comme un motoréducteur à courant continu : c'est l'impulsion de commande qui définit le sens et la vitesse de rotation.

Ne disposant d'aucune butée, ce servomoteur tourne dans les 2 sens de façon continue. Ainsi pour une largeur d'impulsion de l'ordre de 1,5 ms à son entrée, le servomoteur sera à l'arrêt.

Si on augmentera la largeur des impulsions, le servomoteur commencera à tourner dans un sens. Le servomoteur tourne de plus en plus vite au fur et à mesure que la largeur des impulsions approche aux 2 ms (pour laquelle le servomoteur tournera alors à la vitesse maximal).

Si on diminuera la largeur des impulsions (par rapport à la position médiane 1,5 ms), le servomoteur commencera à tourner dans l'autre sens. Le servomoteur tourne de plus en plus vite au fur et à mesure que la largeur des impulsions approche aux 1 ms (pour laquelle le servomoteur tournera alors à la vitesse maximal).

Le servomoteur à rotation angulaire est un micro servomoteur largement utilisé dans le monde du modélisme. Très léger avec ses 9 grammes et son amplitude de 180°, il sera idéal pour animer les projets robotiques/domotiques à partir d'un microcontrôleur tel que l'Arduino comme il est le cas dans notre système.

Ses prises standard Groupées à 3 fils et sa tension d'alimentation de 5V rendent le micro servo directement utilisable sur un Arduino.

Pour répondre à notre besoin et par rapport à ce que nous avons trouvé disponible sur le marché national, on à utiliser le modèle suivant :

<p style="text-align: center;">Servomoteur à rotation angulaire (Micro-Servo) SG90</p>

<p style="text-align: center;"><u>Caractéristiques :</u></p> <ul style="list-style-type: none"> • Alimentation : 4.8 à 6 V • Angle de rotation : 180° • Couple : 1.3 Kg.cm • Finesse : 0.12 sec/60° • Dimensions : 23.2 x 12.5 x 22 mm • Poids : 9g

Tableau 2.26 : Caractéristiques du servomoteur utilisé

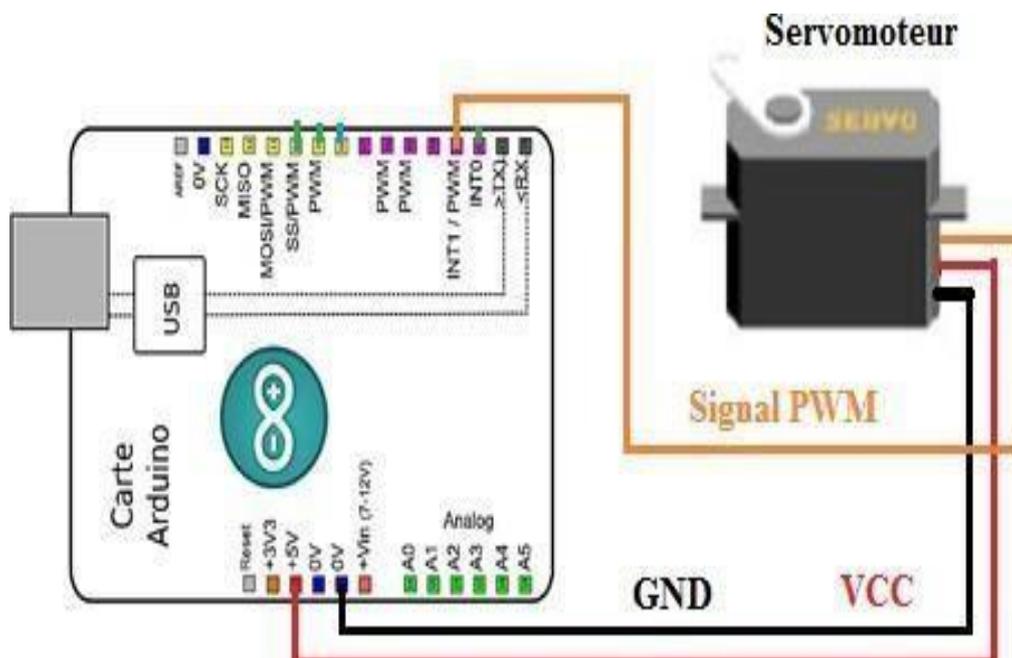


Figure 2.27 : branchement du servomoteur avec l'Arduino

B Electrovanne plastique

Le principe de l'électrovanne consiste à contrôler le flux du liquide en utilisant le flux d'électrons ! Il dispose de deux raccords 1/2 pouce. La valve est fermée au repos, quand une tension de 220VAC est appliquée aux deux bornes, la vanne s'ouvre et l'eau peut pousser cette dernière pour passer au travers.

Le liquide ne peut aussi circuler que dans un seul sens (indiqué par une flèche sous la vanne). Le solénoïde de la vanne a été testé sous différentes tensions continue. Il est possible d'activer la vanne à partir d'une tension de 220VAC (mais elle est plus lente à s'ouvrir).

Nous allons utiliser l'électrovanne pour assurer la fonction d'arrosage du jardin.



Figure 2.28 : *Electrovanne plastique 220 VAC*

- **Détails techniques sur l'électrovanne :**

- ✓ Pression de fonctionnement : 0 – 0.8 Mpa (0.8 Meg pascal correspond à 8 bar).

- ✓ Température de fonctionnement : 100 °C.
- ✓ Temps de réponse (ouverture) : ≤ 0.15 sec.
- ✓ Temps de réponse (fermeture) : ≤ 0.3 sec.
- ✓ Tension de fonctionnement 220VAC.
- ✓ Matériau : laiton + plastique.
- ✓ Dimensions : 84mm (L) x 57mm(H).

- **Branchement de l'électrovanne avec la carte Arduino :**

La commande de la vanne se fait à travers **d'un relais**. il suffit juste de brancher la broche du VCC avec le 5v de l'arduino, GND avec le GND de l'arduino et le signal avec une broche digital de l'arduino.

C Les relais

Un relais est un commutateur électrique qui permet de commander un second circuit utilisant généralement un courant bien supérieur à ce que notre carte Arduino pourrait accepter (par exemple, allumer/éteindre une ampoule 220V), il n'y a pas de connexion entre le circuit basse tension de l'Arduino et le circuit haute tension.

Le relais est un mécanisme simple d'interrupteur on/off : il se ferme lorsque l'entrée est à 5V et s'ouvre lorsque l'entrée est à 0V.

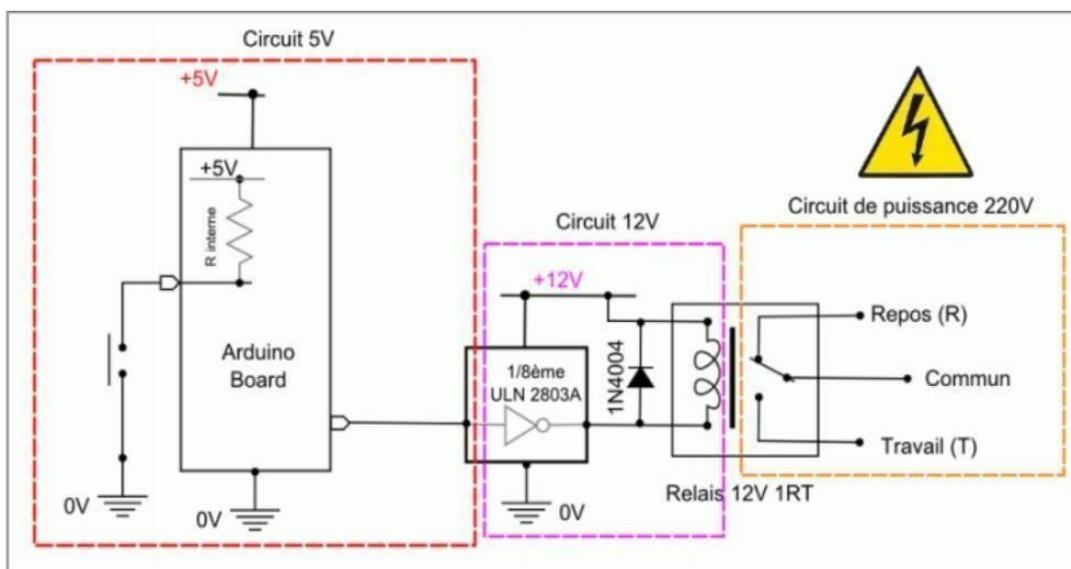


Figure 2.29 : Principe de fonctionnement du relais

- **Module relais 5V 1 canal 220V 10A**

Ce module fournit trois connexions COM, NC et NO. NC : Cela veut dire que lorsque le relais

n'a pas du signal d'entrée, le circuit haute tension connecté sera actif. Si par contre, on applique une tension de 5V au relais, le circuit secondaire sera coupé. NO : Cela veut dire qu'a contrario, une valeur de 5V appliqué au relais coupera le circuit haute tension et inversement.

- **Description du module :**

Relais 250V 10A

Transistor

LED qui signifie que le module est correctement alimenté

Ce module est un actionneur. Le connecteur est une entrée (INPUT) qui doit être connectée à une des sorties (OUTPUT) de notre carte Arduino.



Figure 2.30 : Module relais 5V 1 canal 220V 10A

L'utilisation de ce type de module dans notre projet a pour but de contrôler la vanne de l'arrosage et le temps d'allumage de la télévision dans la chambre des enfants.

D Module Afficheur LCD

Les afficheurs LCD (Liquid Crystal Display) sont devenues incontournables dans toutes applications qui demandent la visualisation de paramètres, il s'agit donc d'une interface Homme/Machine. Ils sont très utilisés dans les montages à microcontrôleur, et permettent une grande convivialité. Ils peuvent aussi être utilisés lors de la phase de développement d'un programme, car on peut facilement y afficher les valeurs de différentes variables.

Vu de l'extérieur, les écrans LCD alphanumériques sont essentiellement caractérisés par leur taille. Deux modèles sont fréquemment utilisés, le premier ayant 2 lignes et 16 colonnes d'affichage et le deuxième ayant 4 lignes et 20 colonnes d'affichage.

- **Connecteur de l'afficheur LCD**

Ces deux écrans (afficheur LCD 16x2 et 20x4 ont exactement la même connectique, c'est à

dire un connecteur 16 broches.

Ce connecteur véhicule plusieurs signaux dont une partie forme un bus de communication parallèle 4 ou 8 bits selon la configuration choisie ainsi que les signaux permettant de contrôler la communication entre l'Arduino et l'écran.

La figure ci-dessous donne la nomenclature des broches de ce connecteur :

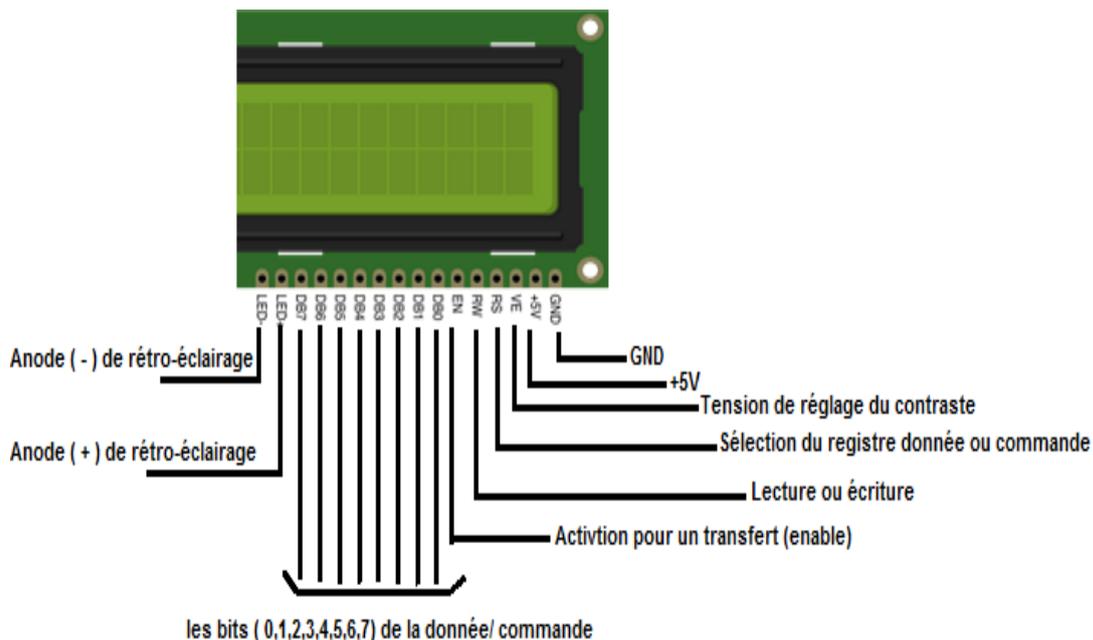


Figure 2.31 : Nomenclature du connecteur de l'afficheur LCD

● Communication avec le LCD

L'afficheur LCD peut fonctionner en mode 4 bits ou en mode 8 bits. En mode 8 bits, les octets sont transférés sur les lignes DB0 à DB7. En mode 4 bits les octets sont transférés en deux fois sur les lignes DB4 à DB7.

Le LCD dispose de 3 registres internes, le registre de données permettant entre autre l'envoi des codes des caractères à afficher, le registre de commande permettant d'envoyer des commandes d'effacement de l'écran, de positionnement du curseur, etc., et le registre d'état qui permet de consulter notamment la disponibilité du LCD pour recevoir des commandes ou des données.

La sélection de l'un ou l'autre de ces registres est effectuée via les états, LOW ou HIGH, des lignes RS et RW. Une fois l'état de ces deux lignes établies, EN est placé à HIGH, la donnée ou la commande est placée sur les lignes DBx puis en est mise à l'état LOW.

Piloter directement un LCD est donc un processus relativement compliqué. Pour nous faciliter

la tâche, il existe des bibliothèques prêtes à utiliser.

- **Connexion avec la carte Arduino**

Dans la grande majorité des cas, on préférera une communication sur 4 bits car une communication sur 8 bits consomme 10 ou 11 broches, ce qui, hormis sur un Méga ou un Due, laisse peu de broches libres. Avec une communication 4 bits, 6 à 7 broches sont nécessaires. Le choix des broches est libre.

- **Caractéristiques de l’afficheur LCD 16x2**

L’afficheur LCD 16x2 I2C dispose des caractéristiques techniques suivantes :

- ✓ Affichage : 16 colonnes, 2 lignes.
- ✓ Tension : 5V uniquement nécessaires.
- ✓ Courant : 135 mA typique. avec rétro-éclairage allumé.
- ✓ Modes : I2C / 9 600 baud de communication série.

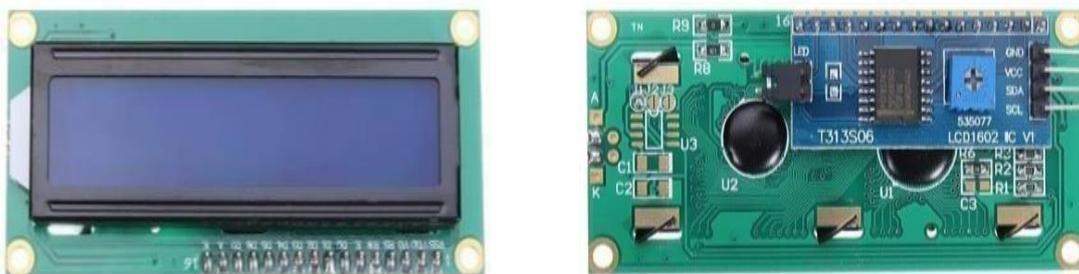


Figure 2.32 : Afficheur LCD 16x2 avec I2C

Il suffit de les intégrer dans notre projet afin de pouvoir utiliser l’écran LCD.

- **Branchement de LCD 16x2 I2C avec la carte Arduino**

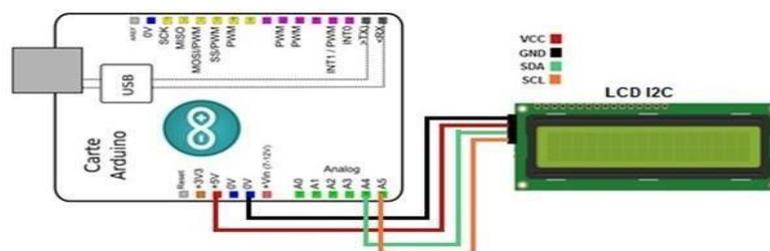


Figure 2.33 : Branchement de LCD 16x2 avec la carte Arduino

E Le Buzzer

Ce composant a pour fonction d'associer un bip sonore à chaque appui sur chacune des touches du clavier matriciel, ainsi pour générer une alarme (sous forme d'un long bip ou sous forme de trois ou quatre bips successifs) dans le cas où il y a une fuite de gaz, ou une détection de mouvement.

- **5V active Buzzer Continu Bip alarme sonneur**

Pour notre système, nous avons choisi un simple Buzzer (5V active Buzzer Continu Bip alarme sonneur) de caractéristiques suivantes :

- ✓ Tension : 5V DC
- ✓ Sortie Min sonore à 10cm : 85dB
- ✓ Taille totale (Pin non inclus) : 12 x 9mm / 0,47 "x 0,35" (D * H)
- ✓ Matériaux : plastique, metal
- ✓ Poids net : 8g



Figure 2.34 : 5V active Buzzer

- **Branchement du Buzzer avec la carte Arduino**

Le Buzzer peut être branché directement sur l'Arduino en reliant ses deux pins avec la GND et le 5V comme on peut le brancher en reliant son pin (+) avec une sortie numérique de l'Arduino à travers une résistance le Buzzer se déclenchera alors dès l'activation de la sortie numérique. La résistance sert à diminuer le volume de sonnerie.

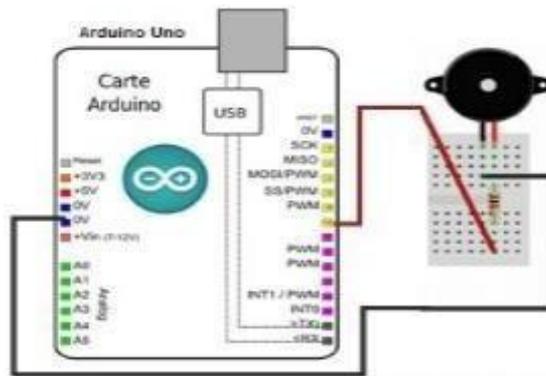


Figure 2.35 : branchement du buzzer avec la carte arduino

F Ventilateur

La domotique s'applique également au chauffage et à la ventilation, des éléments indispensables au sein d'un logement. La solution est la régulation du chauffage, c'est-à-dire programmer un degré de température adapté à chaque pièce et notamment en fonction des moments de la journée. Ces températures sont contrôlées depuis l'application conçue pour assurer une température confortable.

Du côté fraîcheur, une installation domotique dédiée permettra de programmer les aspirations et extractions d'air, notamment à des endroits stratégiques et nécessaires comme la cuisine.

Dans notre projet, nous avons utilisé un simple ventilateur :



Figure 2.36 : ventilateur 5 V d'un PC

Concernant le branchement du ventilateur avec la carte Arduino, nous avons utilisé un transistor (NPN).

- ✓ La broches (+) du ventilateur est reliées avec la broche émetteur du transistor.
- ✓ La broches (-) du ventilateur est reliée avec le GND de la carte Arduino.
- ✓ La broche du collecteur du transistor est reliée avec le Vcc de l'alimentation 12v.
- ✓ La broche de la base du transistor est reliée avec une broche numérique de la carte Arduino

en passant par une résistance 1k.

G Téléphone portable (Samsung - B313E)

Ce téléphone a pour but de signaler une intrusion, un capteur de mouvement branché en Input détecte un mouvement et envoie un signal à la carte arduino qui envoie des ordres pour contrôler le téléphone qui lance un appel pour nous informer et prend une photo de l'intrus qui sera enregistré par la suite.



Figure 2.37 : Samsung B313E

H Clavier matriciel 4x4

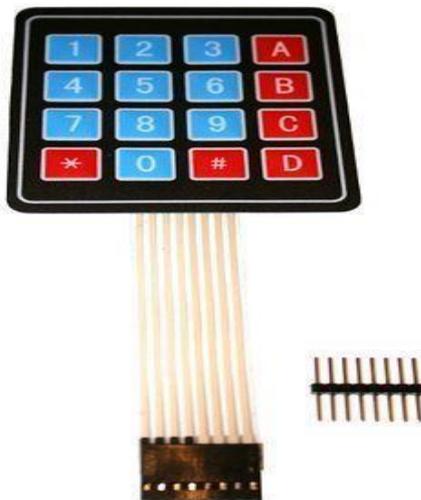


Figure 2.38 : clavier matriciel

- **Description**

Les lignes sont des sorties. Les colonnes sont des entrées maintenues au niveau haut par une résistance interne à l'arduino.

Le système envoie par balayage un niveau bas sur chaque ligne (1 seule à la fois) et balaye les colonnes en lecture. Quand il lit un niveau bas, c'est que la colonne est reliée par une touche appuyée à la ligne qui est basse à ce moment.

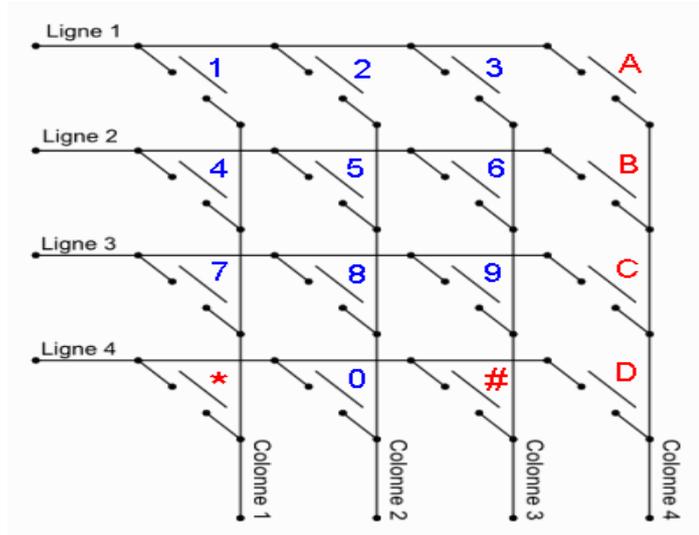


Figure 2.39 : structure interne du clavier matriciel

Le clavier matriciel possède 7 broches et ne possède pas de VCC ou Gnd.

- **Branchement :**

Le branchement est assez simple, connectant les broches A5, A4, A3, A2 de la carte Arduino vers les lignes et les

Broches 5, 4, 3, 2 vers les colonnes, comme le montre la figure suivante :

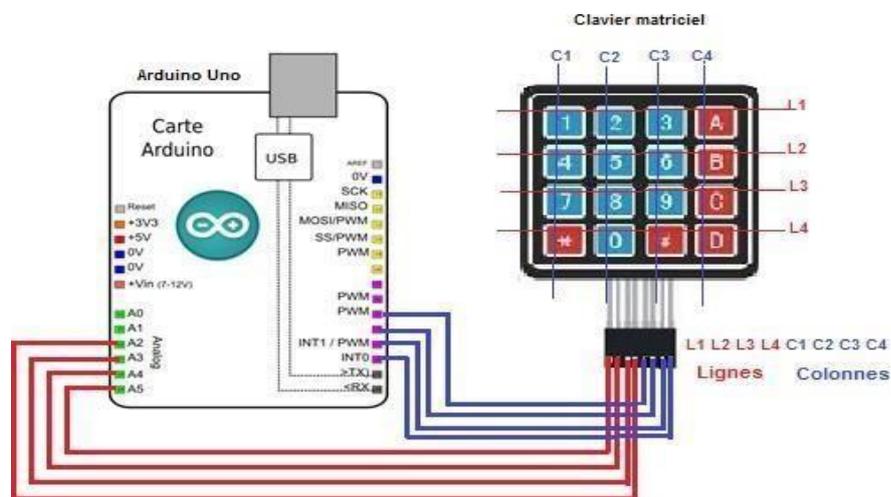


Figure 2.40 : Branchement du clavier avec la carte arduino

I Bouton poussoir

Boutons ou commutateurs poussoirs relient deux points dans un circuit lorsque le bouton est appuyé. Nous allons utiliser ce bouton dans la fonction de l'accès sécurisé à la maison pour que l'utilisateur puisse ouvrir ou fermer la porte principale de la maison manuellement.

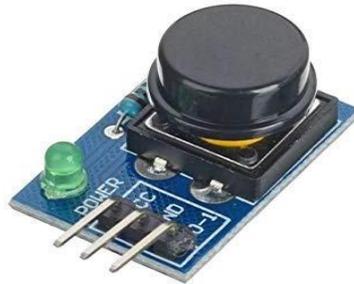


Figure 2.41 : Bouton poussoir

- **Branchement avec la carte Arduino**

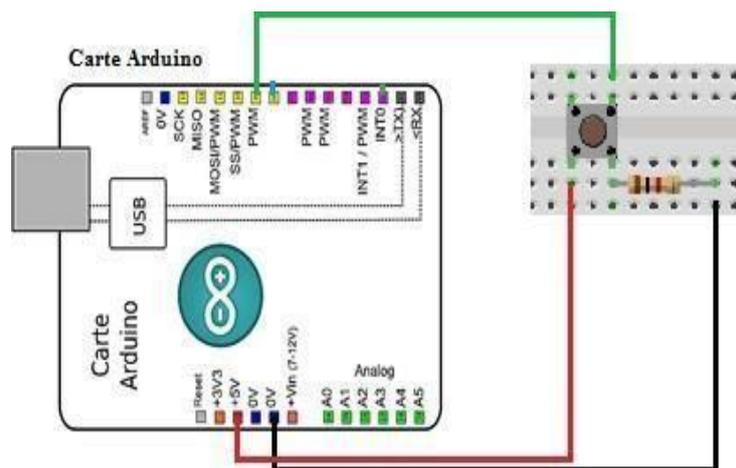


Figure 2.42 : Branchement du bouton poussoir avec Arduino

Lorsque le bouton-poussoir est ouvert, il n'y a pas de lien entre les deux branches du bouton-poussoir, de sorte que la broche est reliée à la masse (à travers la résistance pull-down) et nous avons lue un LOW. Lorsque le bouton est fermé (enfoncé), il établit une connexion entre ses deux jambes, reliant la broche à 5 volts, de sorte que nous lisons un HIGH.

J Les LEDs

Une diode électroluminescente (DEL en français, ou LED en anglais : Light-Emitting Diode), est un dispositif optoélectronique capable d'émettre de la lumière lorsqu'il est parcouru par un courant électrique. Une diode électroluminescente ne laisse passer le courant électrique que

dans un seul sens (le sens passant, comme une diode classique, l'inverse étant le sens bloquant) et produit un rayonnement monochromatique ou polychromatique non cohérent à partir de la conversion d'énergie électrique lorsqu'un courant la travers.

L'utilisation des diodes électroluminescentes dans notre projet a pour but de remplacer les lampes des chambres et d'ajouter un système de signalisation. A titre d'exemple, si le mot de passe d'accès a la maison a été correctement, une LED verte sera allumée sinon une LED rouge sera allumée.

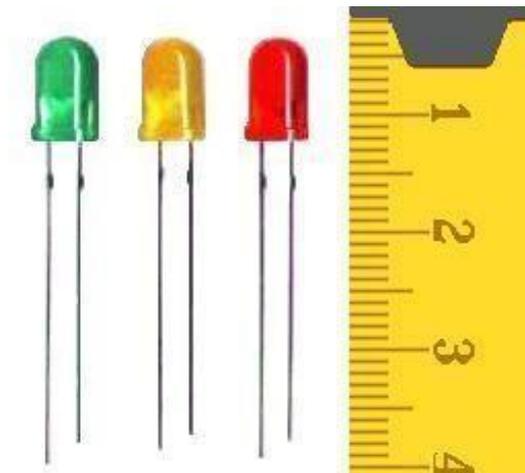


Figure 2.43 : LEDs de différente couleur

- **Branchement avec la carte Arduino**

Lorsque la sortie numérique de l'Arduino n'est pas activée (LOW), il n'y a pas une différence de potentiel entre les broches de la LED cependant lorsqu'elle est activée, la tension appliquée aux bornes de la diode est supérieure à la tension seuil de la LED, donc elle sera allumée.

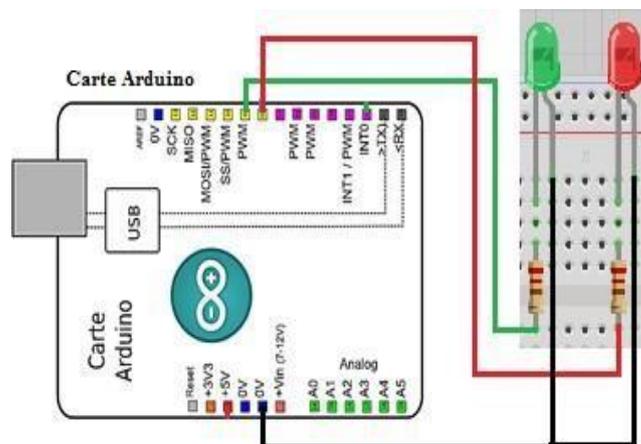


Figure 2.44 : Branchement des LEDs avec Arduino

K RTC DS3231

RTC ("Real Time Clock"), horloge en temps réel

- **Description**

Ce module permet d'ajouter une horloge à nos projets, avec date et heure, qui restera à jour même quand notre projet (et notamment notre carte Arduino) ne sera pas alimenté. En effet, le module est équipé d'une pile CR2032 qui durera plus d'un an, gardant ainsi votre appareil à l'heure.



Figure 2.45 : RTC DS3231

- **Caractéristique**

- Taille: 38mm (longueur) * 22mm (W) * 14mm (hauteur)
- Poids: 8g
- Tension de Fonctionnement: 3.3-5.5 V
- Précision de L'horloge : 0-40 ° gamme, la précision 2ppm, l'erreur a été environ 1 minute

- **Branchement avec la carte arduino**

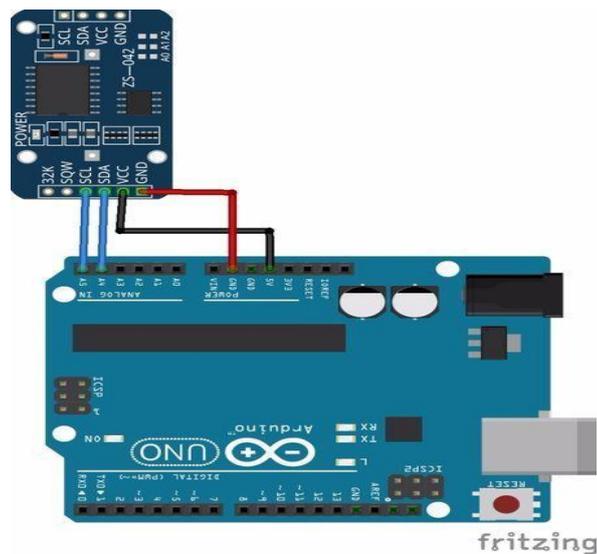


Figure 2.46 : Branchement du RTC DS3231 avec l'Arduino

Dans notre projet nous avons utilisé ce composant pour commander le temps de possibilité d'allumer la télévision dans la chambre d'enfant et le temps d'ouverture et fermeture de l'électrovanne (gestion d'arrosage).

L Alimentation

Après avoir défini les besoins électriques nécessaires, nous avons entamé une phase de recherche d'une alimentation, et nous avons trouvé une solution qui peut satisfaire nos besoins électriques et qui respecte le rapport performance-prix. Cette solution consiste à utiliser une batterie externe pour alimenter la carte arduino, cette batterie fournit une tension de 5v, et débite un courant de 2,1A donc elle est capable d'alimenter jusqu'à 4 carte arduino puisque le fonctionnement de chacune des cartes nécessite un courant de 0.5A maximum.

Nous avons utilisé aussi une autre batterie qui fournit une tension de 12v pour alimenter les ventilateurs. Et concernant les autres actionneurs qui nécessite une tension de 5v, nous avons utilisé un générateur de courant (alimentation stabilisée), sauf que les servomoteurs consomme beaucoup, donc nous avons préféré d'ajouter une autre alimentation 5v pour ces deux servomoteurs.



Figure 2.47 : alimentation stabilisée 15V - 2A

- **Le but de l'utilisation de l'alimentation stabilisée :**

Nous avons utilisé cette alimentation stabilisée pour protéger nos composants contre les courts circuits en fixant la valeur du voltage et d'ampérage.

D'un autre côté, utiliser une alimentation stabilisée va nous permettre de détecter l'existence d'un problème, si jamais un fil a été débranché dans le montage du circuit on le saura grâce à cette alimentation.

- **Fonctionnement de l'alimentation stabilisée :**

Cette alimentation utilise le principe de découpage généralement utilisée pour fournir un courant DC de 0 à 15V. Elle est aussi utilisée dans les laboratoires des écoles, la réparation téléphonie mobile, et d'autres domaines de l'électronique. Une alimentation stabilisée permet le réglage de la tension et du courant fondamental pour le fonctionnement d'un système.

2.3 Etude de la partie logicielle du projet

Cette partie est dédiée à la représentation des plateformes informatiques utilisées dans le développement de notre système domotique.

2.3.1 Plateforme de programmation Arduino

A Présentation

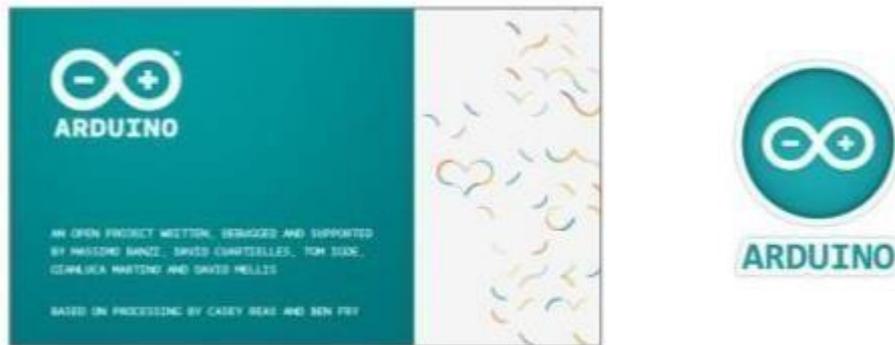


Figure 2.48 : logo d'interface de développement Arduino

L'espace de développement Arduino a pour fonctions principales :

- ✓ de pouvoir écrire et compiler des programmes pour la carte Arduino.
- ✓ de se connecter avec la carte Arduino pour y transférer les programmes.
- ✓ de communiquer avec la carte Arduino.

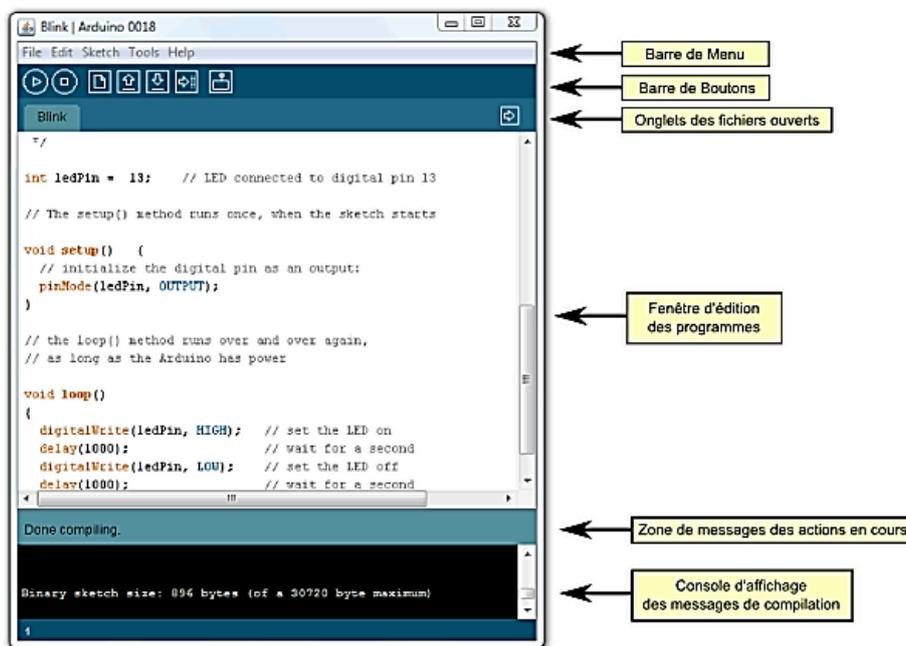


Figure 2.49 : Interface de programmation Arduino

Cet espace de développement intégré (EDI) dédié au langage Arduino et à la programmation des cartes Arduino comporte :

- ✓ une **barre de menus** comme pour tout logiciel une interface graphique (GUI),
- ✓ une **barre de boutons** qui donnent un accès direct aux fonctions essentielles du logiciel et fait toute sa simplicité d'utilisation,
- ✓ un **éditeur** (à coloration syntaxique) pour écrire le code de vos programmes, avec onglets de navigation,
- ✓ une **zone de messages** qui affiche indique l'état des actions en cours,
- ✓ une **console texte** qui affiche les messages concernant le résultat de la compilation du programme.

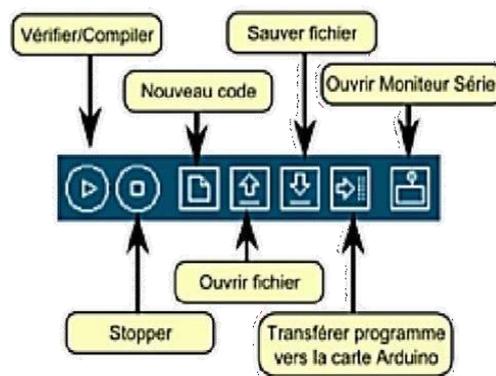


Figure 2.50 : Barre de boutons Arduino

Le logiciel comprend aussi un moniteur série qui permet d'afficher des messages textes émis par la carte Arduino et d'envoyer des caractères vers la carte Arduino (en phase de fonctionnement) :

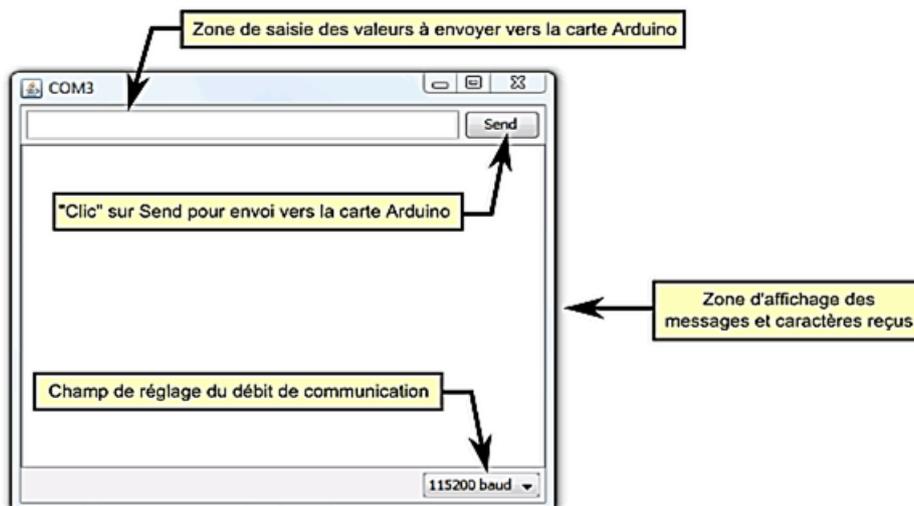


Figure 2.51 : Hyper Terminal de l'Arduino (Moniteur Série)

B Structure d'un programme en Arduino

Le langage Arduino est inspiré de plusieurs langages. On retrouve notamment des similarités avec le C, le C++, le Java et le Processing. Le langage impose une structure particulière typique de l'informatique embarquée.

Chaque programme en Arduino comprend :

- un entête déclaratif
- une partie « configuration » qui ne sera exécutée qu'une fois (fonction **setup()**)
- une partie constituée d'une boucle sans fin que le programme répètera à l'infini (fonction **loop()**) : c'est le cœur du programme.

Un programme utilisateur Arduino est une suite d'instructions élémentaires sous forme textuelle, ligne par ligne. La carte lit puis effectue les instructions les unes après les autres, dans l'ordre défini par les lignes de code.

Afin de compiler le programme sous l'espace de développement Arduino, il faut le télécharger dans la carte Arduino via la connexion USB.

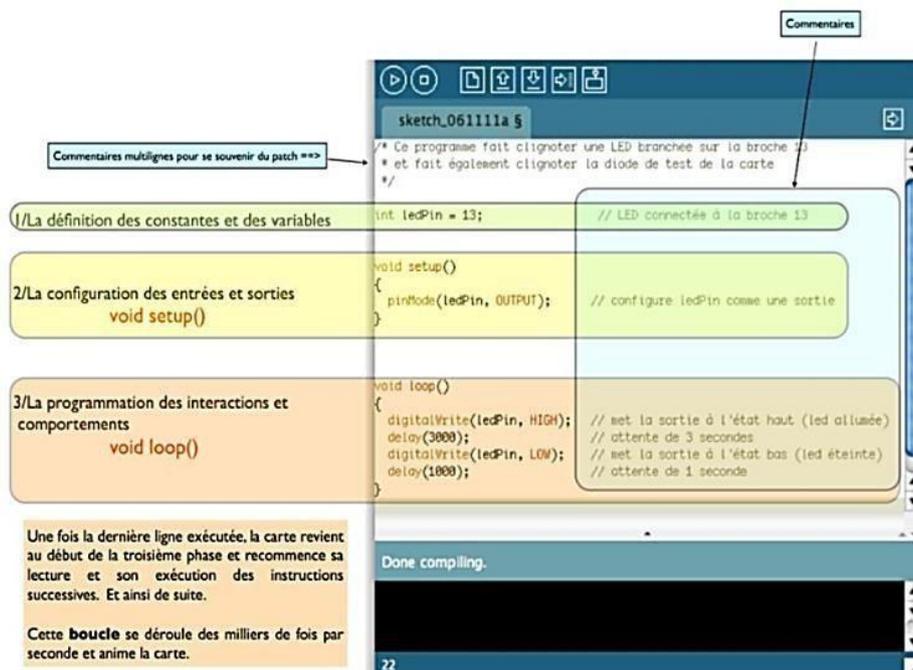


Figure 2.52 : Structure d'un programme en Arduino EDI

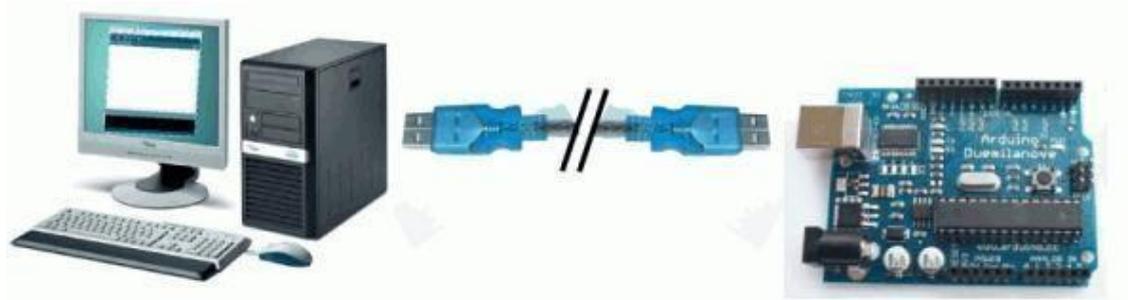


Figure 2.53 : Téléversement du programme sous EDI vers la carte Arduino

2.3.2 Plateforme de programmation Visual Studio

A Présentation

Microsoft Visual Studio est une suite de logiciels de développement pour Windows et macOS conçue par Microsoft. La dernière version s'appelle **Visual Studio 2019**.

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications web ASP.NET, des services web XML, des applications bureautiques et des applications mobiles, Visual C++, **Visual BASIC**, Visual C# utilisent tous le même environnement de développement intégré (IDE), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du framework .NET, qui fournit un accès à des technologies clés simplifiant le développement d'applications web ASP et de services web XML grâce à Visual Web Developer.



Figure 2.54 : Logo Visual studio

Ce Logiciel comporte deux fenêtres dans son espace de développement, la première a pour objet de traiter le côté design (afficheurs, boutons, images...) et la deuxième est conçu pour la programmation des éléments présenter dans la fenêtre de design.

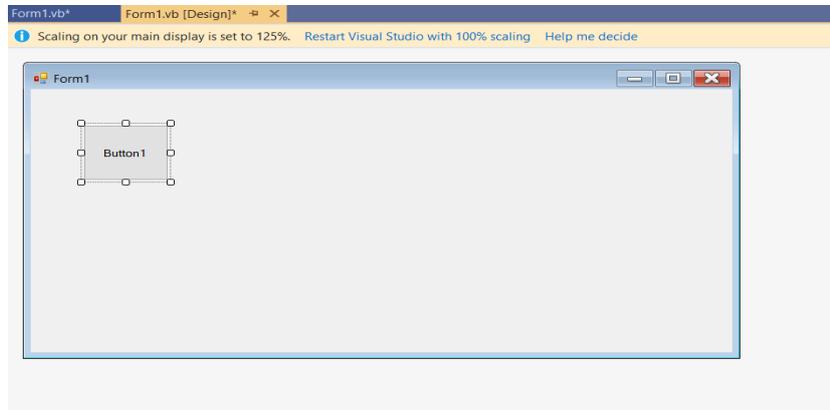


Figure 2.55 : Fenêtre de design

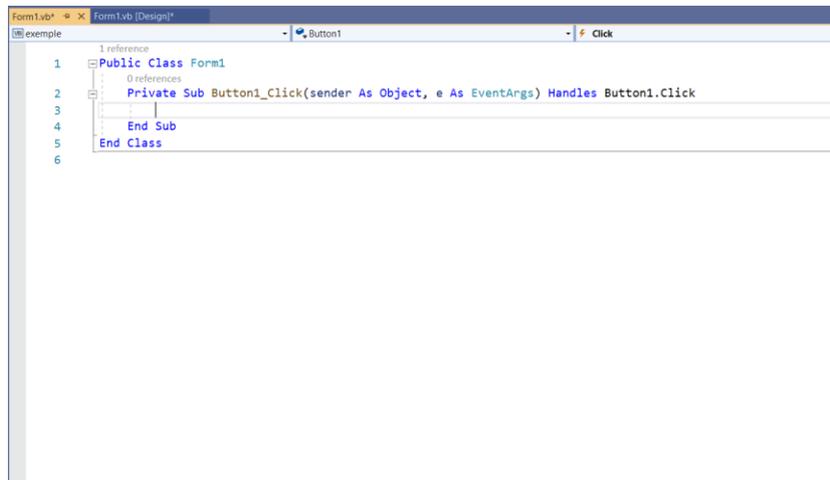


Figure 2.56 : Fenêtre de programmation

Cet espace de développement comporte aussi :

- ✓ Une **barre de menus** comme pour tout logiciel une interface graphique (GUI).
- ✓ Une **barre de boutons** qui donnent un accès direct aux fonctions essentielles du logiciel et fait toute sa simplicité d'utilisation.
- ✓ Un bouton **STAR** et un bouton **STOP**, pour l'exécution et l'arrêt du programme.
- ✓ Une **liste de contrôles** (Toolbox) qui permet de trouver et d'ajouter des contrôles à l'interface.
- ✓ Une **barre de propriétés** pour voir et modifier les propriétés de chaque contrôle.

- ✓ Une **liste d'erreurs** qui affiche les erreurs, les avertissements et les messages après chaque appui sur le bouton STAR

Dans notre projet nous avons utilisé le langage Visual basic pour développer notre application.

B Visual Basic

Visual Basic est, comme son nom l'indique, un outil visuel permettant de créer sans notion de programmation l'interface graphique (GUI - Graphical User Interface) en disposant à l'aide de la souris des éléments graphiques (boutons, images, champs de texte, menus déroulants, ...).

L'intérêt de ce langage est de pouvoir associer aux éléments de l'interface des portions de code associées à des événements (clic de souris, appui sur une touche, ...). Pour cela, Visual Basic utilise un petit langage de programmation dérivé du BASIC (signifiant Beginners All-Purpose Symbolic Instruction Code, soit code d'instructions symboliques multi-usage pour les débutants).

- **Avantages**
 - ✓ Simplicité.
 - ✓ Disponible dans MS Office.
 - ✓ Accès à toutes les fonctions de Windows (Systèmes de fenêtres, boîtes de dialogue, API...)
- **Inconvénients**
 - ✓ Pas très rapide.
 - ✓ Dépend beaucoup des versions utilisées de Windows et MS Office.

Nous allons donc programmer une interface graphique sur Visual studio pour contrôler un Arduino que nous allons aussi programmer pour fabriquer une maison intelligente.

C Les étapes de création de l'interface graphique

Dans les 7 étapes suivantes je vais vous expliquer comment coder tout ça.

- **1^{ère} étape**

On lance le logiciel Visuel Studio, on clique sur « create a new project », puis on choisit le langage de programmation (visual basic) et on clique sur « Windows forms app (NET Framework) », ensuite on donne un nom a notre projet et on sélectionne le répertoire ou on veut l'enregistrer et on clique sur « create ».

- **2^{ème} étape Designer l'interface**

Nous nous retrouverons devant une fenêtre de ce genre

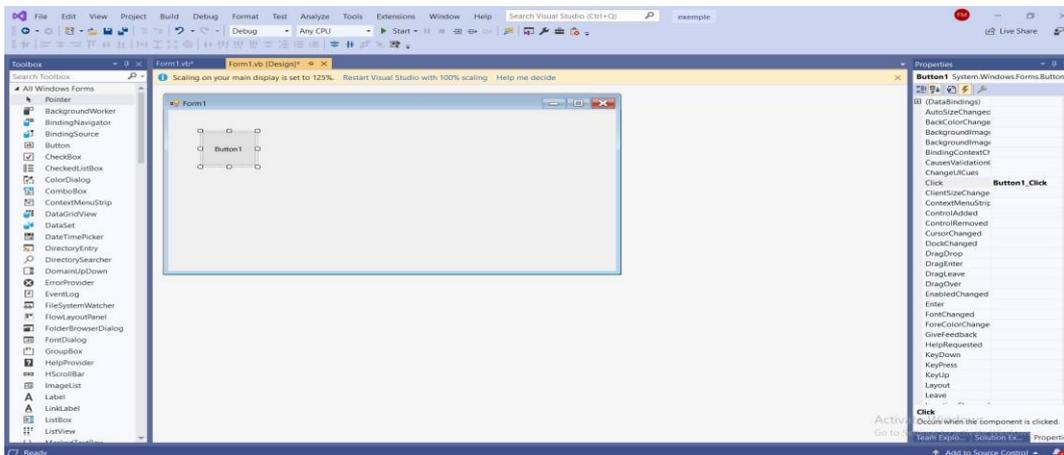


Figure 2.57 : interface de design VS

Nous allons donc ajouter à la fenêtre « Form1 » quelques boutons et labels qui nous serviront à communiquer avec Arduino. Pour ce faire, nous allons ajouter un « label », un « bouton » et un « numericUpDown » que nous trouverons dans le menu de droite nommé « Boîte à outils ». Il faut aussi rajouter un composant qui s'appelle « serialPort » qui nous servira à communiquer avec l'Arduino.

Ensuite nous allons ajouter d'autres contrôles selon nos besoins, et nous allons modifier l'emplacement et les propriétés de ces contrôles pour avoir une application présentable, bien claire et facile à utiliser.

- **3^{ème} étape : Coder l'interface**

Nous allons maintenant passer aux choses sérieuses, il va falloir coder cette interface pour que nos différents éléments fonctionnent comme on le souhaite.

Une double clique sur chaque un de ces contrôles nous permet de le programmer dans l'interface de codage.

La communication entre l'Arduino et l'interface graphique se fera à l'aide du « Serial Port ».

Pour envoyer une information à l'arduino (par exemple : allumer une led), il suffit d'envoyer un caractère à l'Arduino après avoir ouvert la connexion et enfin, la fermer.

- **4^{ème} étape : Programmer Arduino**

L'Arduino n'a pas d'intelligence artificielle, il ne comprendra sûrement pas ce qu'il devrait faire quand il recevra le caractère « a » par exemple. Nous allons donc lui dire dans son langage qu'il devrait par exemple allumer une led, donc voilà le programme qu'il faut pour allumer une led par un bouton dans l'application.

```
int ledPin = 13;           // Sortie où sera branchée la LED
int ledState = LOW;       // Etat de la LED (LOW par défaut)

void setup() {
  Serial.begin(9600);     // On initialise la connexion
  pinMode(ledPin, OUTPUT); // Et la sortie de la LED
}

void loop() {
  int received;          // Variable servant à récupérer
                        // les données reçues

  if (Serial.available() > 0) { // Si des données sont disponibles
    received = Serial.read(); // On les récupère
    if (received == 'a') { // Si "a" est reçu
      if (ledState == HIGH) // Et que la LED est allumée
        ledState = LOW; // On l'éteint
      else // Sinon
        ledState = HIGH; // On l'allume

      digitalWrite(ledPin, ledState); // Enfin on change l'état de la LED
    }
  }
}
```

Figure 2.58 : Programme pour allumer une led via une application

Conclusion

Au niveau de ce chapitre important qui représente une étude approfondie sur le projet, nous avons parlé dans un premier temps des généralités de la domotique à savoir ces origines, son principe de fonctionnement, les tâches associées à la domotique et les différentes technologies associées à une installation domotique.

Dans la deuxième partie, nous avons étudié de manière détaillée toute la partie matérielle (HARD) où nous avons décrit les critères du choix de la carte programmable à adopter pour notre système, c'était la carte Arduino. Ensuite nous avons présenté tous les composants utilisés à savoir : les capteurs (température et humidité, capteur gaz, flotteur et capteur de mouvement), les actionneurs (servomoteurs, électrovanne, ventilateur, afficheur LCD I2C,...),

autres composants (clavier matriciel, l'horloge, bouton poussoir, LEDS). Et à la fin, nous avons traité la partie alimentation.

Finalement, nous avons décrit l'ensemble des plateformes informatiques utilisées dans le développement du système domotique.

Chapitre 3

Phase de la réalisation du système domotique

- **Réalisation final du prototype de la maison intelligente**
- **Réalisation final du système domotique**
- **Coût du projet**

Introduction

Ce chapitre est la clôture de ce mémoire où nous allons présenter toutes les étapes que nous avons suivies afin de réaliser notre projet, commençant par la réalisation du prototype, puis la conception des schémas des montages.

3.1 Réalisation du prototype de la maison intelligente

Avant de passer à la réalisation du prototype sur terrain, nous avons commencé par le réaliser sur le logiciel de conception Revit architecture. Voici l'image qui illustre notre prototype.



Figure 3.59 : *la maison intelligente (prototype) sur Vite Architecture*

Nous avons utilisé le bois, le plastique, le carton et même le gazon artificiel pour

réaliser notre maison sur terrain. Voici l'image réelle de notre maison.



Figure 3.60 : l'image réelle de l'intérieur de la maison



Figure 3.61 :l'image réelle de l'extérieur de la maison

3.2 Réalisation du système domotique

Cette partie regroupe toutes les étapes que nous avons suivi afin de réaliser le système dédié au fonctionnement des équipements à l'intérieur de la maison, commençant par la conception des schémas des montages jusqu'à l'écriture du programme permettant :

- ✓ Contrôle de l'état du réservoir d'eau ;
- ✓ Gestion d'éclairage ;
- ✓ Gestion d'arrosage ;
- ✓ Acquisition de la température et l'humidité ;
- ✓ Détection d'incendie et détection des fuites des gaz combustibles et des gaz étouffants ;
- ✓ L'accès sécurisé à l'habitat ;
- ✓ L'accès sécurisé au garage ;
- ✓ Système de surveillance et d'alarme ;
- ✓ Gestion de la ventilation et la climatisation.

✓ Les sorties des modules de relais seront branchées avec le circuit de la puissance pour commander

✓ L'électrovanne, la télévision et les boutons du téléphone.

C En ce qui concerne le capteur de gaz MQ-2 :

✓ Le GND et le VCC sont reliés avec le GND et le +5V de la carte Arduino.

✓ La broche A0 est connectée avec la broche analogique A6 de la carte Arduino.

D En ce qui concerne le capteur DHT11 :

✓ Le GND et le VCC sont reliés avec le GND et le +5V de la carte Arduino.

✓ La broche DATA est connectée avec la broche numérique 40 de la carte Arduino.

E En ce qui concerne Le flotteur électrique :

✓ Une broche du flotteur est reliée avec la broche numérique 45 de la carte Arduino.

✓ L'autre broche du flotteur est reliée avec le GND de la carte Arduino.

F En ce qui concerne le Buzzer :

✓ La broche (+) du Buzzer est reliée avec la broche numérique 42 de la carte Arduino.

✓ La broche (-) du Buzzer est reliée avec le GND de la carte Arduino.

G En ce qui concerne les ventilateurs :

✓ Les broches (+) des ventilos sont reliées avec les broches émetteurs des transistors (NPN).

✓ Les broches (-) des ventilos sont reliées avec le GND de la carte Arduino.

✓ Les broches collecteurs des transistors sont reliées avec le Vcc de l'alimentation 12v.

✓ Les broches de la base des transistors sont reliées avec les broches numériques 39 et 41 de la carte Arduino en passant par des résistances 1k.

H En ce qui concerne le Clavier matriciel 4x4 :

✓ Les broches 22, 23,24 et 25 de la carte Arduino sont connectées avec les Colonne 4, 3, 2 et 1 du Clavier

✓ Les broches 26, 27,28 et 29 de la carte Arduino sont connectées avec les Lignes 4, 3, 2 et 1 du Clavier

I En ce qui concerne le RTC ds3231 :

✓ Le GND et le VCC sont reliés avec le GND et le +5V de la carte Arduino.

✓ Les broches SDA et SCL sont connectées avec les broches SDA et SCL de la carte Arduino.

J En ce qui concerne le capteur de mouvement :

✓ Le GND et le VCC sont reliés avec le GND et le +5V de la carte Arduino.

✓ La broche DATA est connectée avec la broche numérique 34 de la carte Arduino.

K En ce qui concerne l'afficheur LCD 16x2 I2C :

- ✓ La broche VSS de l'afficheur est reliée avec le GND.
- ✓ La broche VDD de l'afficheur est reliée avec le 5v de l'alimentation.
- ✓ La broche VEE de l'afficheur est reliée avec la consigne du potentiomètre (10Kohm).
- ✓ La broche RS de l'afficheur est reliée avec la broche analogique A0 de la carte Arduino.
- ✓ La broche RW de l'afficheur est reliée avec le GND.
- ✓ La broche E de l'afficheur est reliée avec la broche analogique A1 de la carte Arduino.
- ✓ La broche D4 de l'afficheur est reliée avec la broche analogique A2 de la carte Arduino.
- ✓ La broche D5 de l'afficheur est reliée avec la broche analogique A3 de la carte Arduino.
- ✓ La broche D6 de l'afficheur est reliée avec la broche analogique A4 de la carte Arduino.
- ✓ La broche D7 de l'afficheur est reliée avec la broche analogique A5 de la carte Arduino.
- ✓ La broche A de l'afficheur est reliée avec le 5v de l'alimentation.
- ✓ La broche K de l'afficheur est reliée avec le GND.
- ✓ La broche GND du potentiomètre est reliée avec le GND.
- ✓ La broche 5v du potentiomètre est reliée avec le 5v de l'alimentaion.

L En ce qui concerne les LEDs :

- ✓ La LED rouge est connectée avec la broche 32 de la carte Arduino en passant par une résistance.
- ✓ La LED verte est connectée avec la broche 43 de la carte Arduino en passant par une résistance.
- ✓ Les LEDs des chambres sont connectées avec les broches 48, 49, 50, 51, 52, 53 et 44 de la carte Arduino en passant par des résistances.

M En ce qui concerne les boutons :

• **Bouton ouverture**

- ✓ Une broche du bouton est reliée avec la broche numérique 30 de la carte Arduino.
- ✓ L'autre broche du bouton est reliée avec le GND de la carte Arduino.

• **Bouton fermeture**

- ✓ Une broche du bouton est reliée avec la broche numérique 31 de la carte Arduino.
- ✓ L'autre broche du bouton est reliée avec le GND de la carte Arduino.

N En ce qui concerne le KY-022 Module récepteur infrarouge :

- ✓ Le GND et le VCC sont reliés avec le GND et le +5V de la carte Arduino.
- ✓ La broche DATA est connectée avec la broche numérique 33 de la carte Arduino.

3.2.3 Réalisation de l'interface graphique :

Pour le pilotage du système via la connexion USB, nous avons conçu une application sous la plateforme Visuel Studio, par la suite nous allons présenter dans une image pour décrire le fonctionnement de l'application.

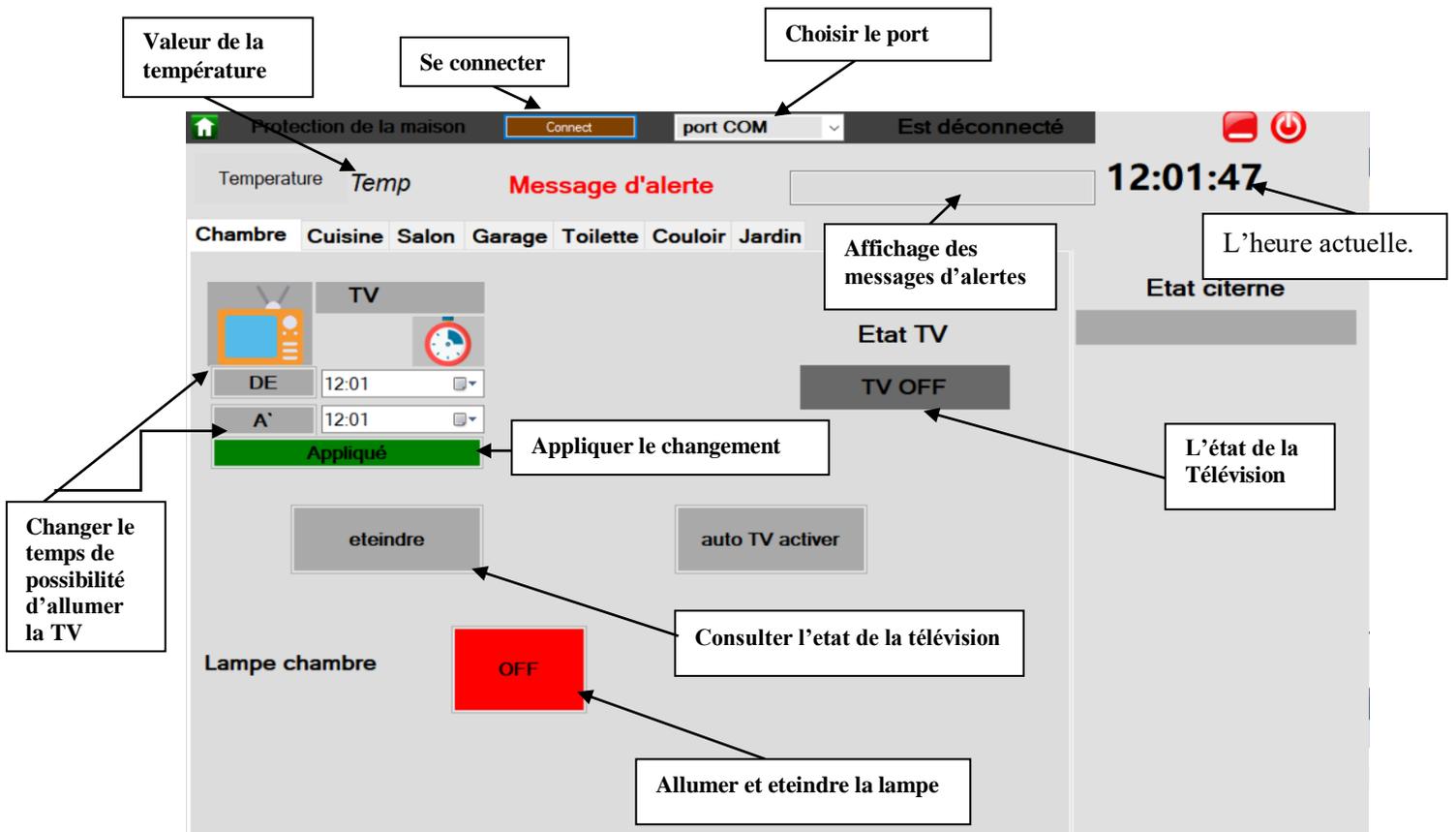


Figure 3.63 : La page d'accueil de l'interface graphique sous Visual studio

NB : Vous trouverez dans l'annexe 1 une série d'images pour décrire en détail le fonctionnement de cette application.

3.2.4 Fonctionnement de l'interface graphique avec la carte arduino

A L'envoi des données de l'application vers la carte arduino

Quand on appuie sur un bouton sur l'application Visual Studio, la carte arduino reçoit des informations, et selon ces informations cette dernière envoie des signaux aux actionneurs (LEDs, Ventilateur...) pour les commander d'une manière spécifique qui est déjà introduite dans notre programme.

On prend par exemple le Bouton ON/OFF d'une lampe, lorsqu'on ouvre l'application et on connecte avec le port de l'Arduino, par défaut ce bouton s'affiche en rouge en état OFF par ce qu'on a modifié la couleur et le texte de ce bouton dans ses propriétés.

- **Le programme qui s'exécute lorsqu'on appuie sur ce bouton. Nous avons fait des commentaires après les deux slashes (//) dans chaque ligne pour mieux comprendre :**

```

If (k_2.Text = "OFF") Then          // si le texte de ce bouton est "OFF"
SerialPort1.WriteLine("<P00480N>") // écriture de " P00480N " dans le serial port
k_2.Text = "ON"                    // modification du text du bouton à "ON"
k_2.BackColor = Color.Green        // modification de la couleur du bouton à «vert»
Else                                // Sinon
SerialPort1.WriteLine("<P00480F>") // écriture de " P00480F " dans le serial port
k_2.Text = "OFF"                  // modification du text du bouton à "OFF"
k_2.BackColor = Color.Red         // modification de la couleur du bouton à «rouge»
End If                             // fin du if

```

Si l'arduino reçoit dans le serial port ce code <P00480N> il allumer la led, et s'il reçoit ce code <P00480F> il éteint la led.

- **Le programme arduino qui exécute cette tâche (avec commentaires) :**

```

if (receivedChars[0] == 'P' ) // Si l'arduino reçoit 'P' dans le premier caractère
{
  if ( receivedChars[4] == '8' && receivedChars[5] == 'O' && receivedChars[6] == 'N' ) // s'il reçoit '8' dans le
4ème caractère et 'O' dans le 5ème caractère et 'N' dans le 6ème
  {
    digitalWrite(led_chambre, HIGH); // allumer la led de la chambre
  }
  if ( receivedChars[4] == '8' && receivedChars[5] == 'O' && receivedChars[6] == 'F' ) // s'il reçoit '8' dans le
4ème caractère et 'O' dans le 5ème caractère et 'F' dans le 6ème

  {
    digitalWrite(led_chambre, LOW); // éteindre la led de la chambre
  }
}

```

B L'envoi des données de la carte arduino vers l'application :

Quand on applique un changement dans l'état d'un capteur, ce dernier envoie un signal à l'arduino, par la suite l'arduino envoie dans le serial port un code, le programme de l'application traite ce code et affiche ce qu'il faut afficher.

On prend par exemple le capteur de niveau d'eau (flotteur), c'est un capteur TOR, lorsque le réservoir commence à se vider le flotteur change son état et envoie un signal à la carte arduino, il suffit d'écrire un code entre guillemets et de l'envoyer dans le port série.

Lorsque l'application reçoit ce code elle exécute un programme qui permet d'écrire dans

l'afficheur des messages d'alertes « Réservoir d'eau vide ».

- Le programme arduino qui reçoit l'information du capteur et envoie un code à l'application :

```
int reservoir = 45;    // Déclaration de la pin du flotteur
int niveau_deau ;    // Déclaration d'une variable
void loop()
{
niveau_deau = digitalRead(reservoir); // Lire l'état du flotteur est le mettre dans la variable déclarée
if (niveau_deau == HIGH) // Si le flotteur indique que le réservoir est vide
{
Serial.println("<ER2>"); // envoyer ce code dans le port série
}
}
```

- Le programme Visuel basic qui reçoit le code du port série et écrit « Réservoir d'eau vide » dans l'afficheur des messages d'alertes :

```
// La fonction parseData sert à traiter les informations reçues par le port série
Function parseData(ByVal Var_Commande_Receve As String)

Dim length As Integer           // déclaration d'une variable comme entier

Select Case Var_Commande_Receve.Substring(1, 2)           // vérifier les deux premiers caractères reçus
Case "ER" // Si ces deux caractères sont 'ER'

Dim var_ID_Error As Integer           // déclaration d'une nouvelle variable var_ID_Error

// vérifier si cette variable possède un seul caractère

Try
var_ID_Error = Var_Commande_Receve.Substring(3, 1)

Catch ex As Exception
Exit Function
End Try

Select Case var_ID_Error
Case 2                               // Si le 3ème caractère est '2'

var_temp_string = "Reservoir d'eau vide"           // écriture de "Reservoir d'eau vide" dans cette variable

Write_TextBox_From_Another_Thread(RichTextBox1, var_temp_string) // mettre cette variable dans l'afficheur des
message d'alertes
```

3.2.5 Quelques scénarios de notre projet

Le scénario du système antivol

- L'utilisateur, active le mode vacance dans l'interface graphique sur son pc ;
- Le système détecte un mouvement à l'intérieur de la maison ;
- Le système interprète l'évènement et génère une alerte ;
- Le système prend une photo du voleur et la sauvegarde ;
- Le système appel le propriétaire.

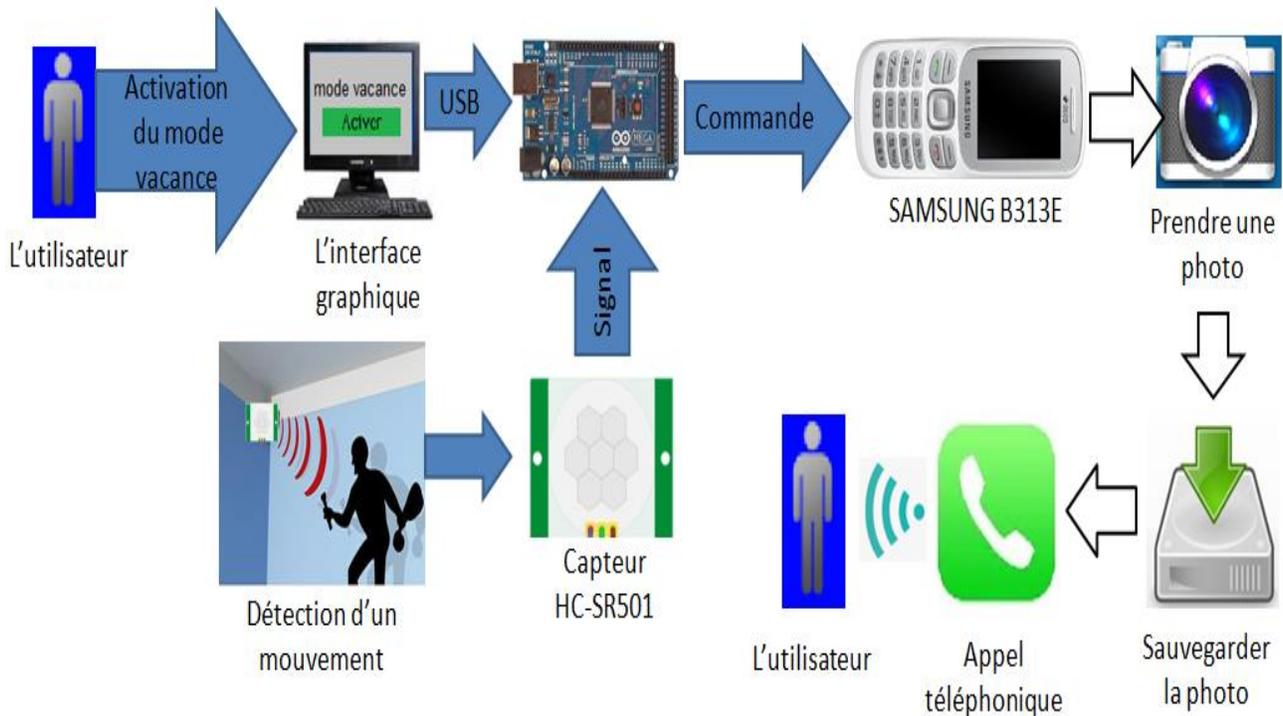


Figure 3.64 : Le scénario du système antivol

B Le scénario du contrôle du temps d'arrosage

- Le propriétaire choisit l'heure et la minute d'ouverture et de fermeture de la vanne d'arrosage et clique sur le bouton appliquer dans l'interface graphique sur son PC ;
- Cette configuration sera envoyé vers la carte Arduino à travers la communication de port série ;
- Si le temps d'ouverture de la vanne arrive, et que le mode auto vanne est activé la vanne s'ouvre automatiquement et attend le temps de fermeture pour qu'elle se ferme ;
- Mais si le mode auto vanne est désactivé la vanne ne s'ouvre pas ;

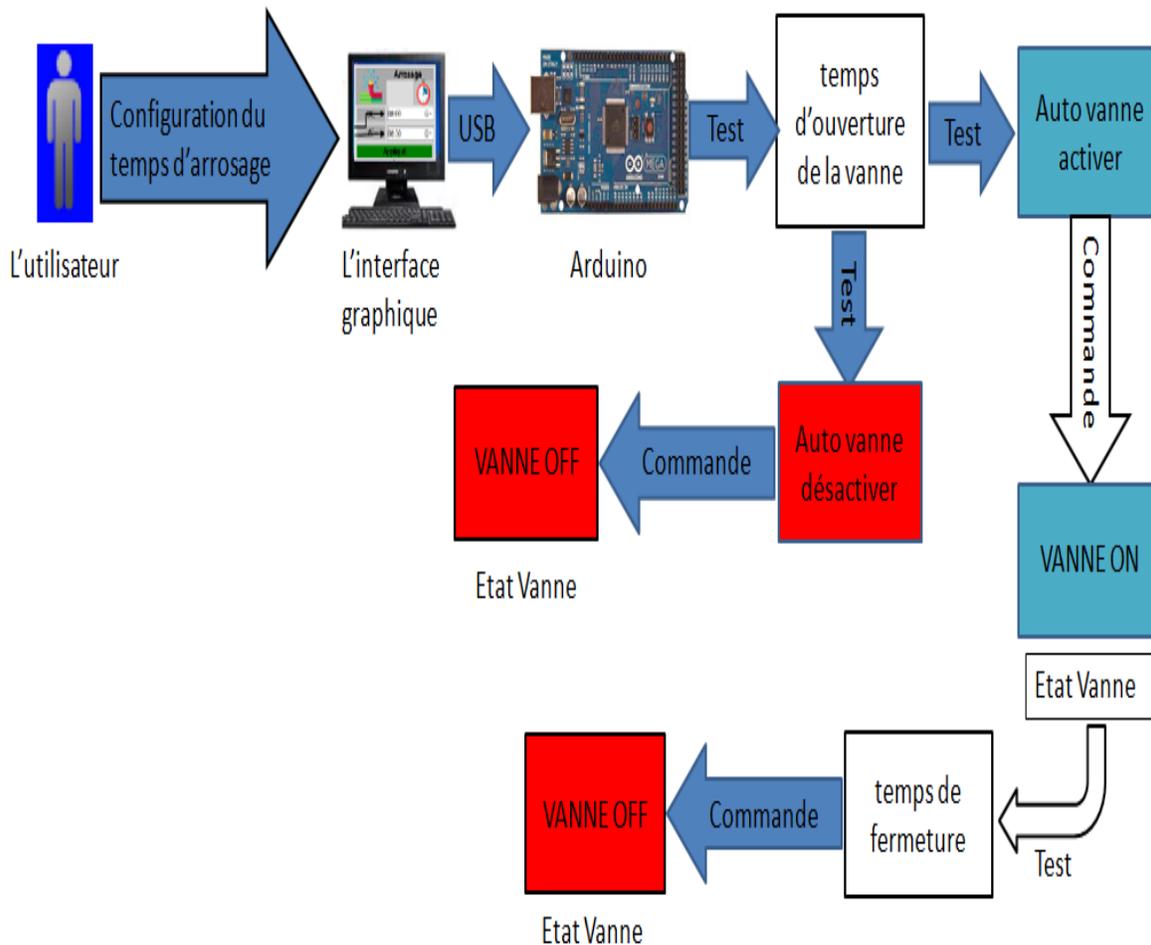


Figure 3.65 : Le scénario du contrôle du temps d'arrosage

C Le scénario de la gestion de la sécurité de la porte

- L'utilisateur modifie le mot de passe dans l'interface graphique sur son pc ;
- Ce nouveau mot de passe sera envoyer vers la carte arduino à travers la communication du port série ;
- Si une personne saisi un mot de passe, l'arduino compare entre ce mot de passe avec celui reçu ;
- Si le mot de passe est juste, la led verte s'allume pour indiquer qu'il est juste et la porte s'ouvre ;
- Si le bouton de fermeture est appuyé, la porte se ferme ;
- Mais si le mot de passe est faut, une alarme se déclenche et une led rouge s'allume, et bien sur la porte ne s'ouvre pas ;

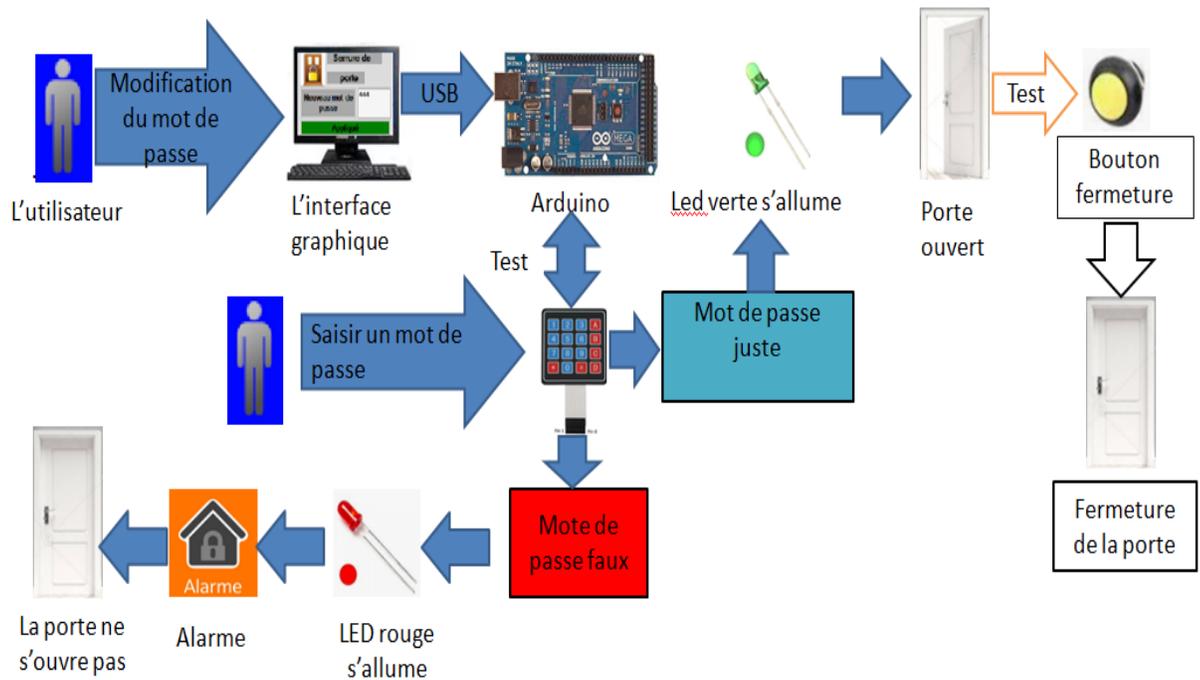


Figure 3.66 : Le scénario de la gestion de la sécurité de la porte

D Le scénario de surveillance du niveau d'eau dans le réservoir

- L'utilisateur dans le cas 1 le réservoir est plein, et le capteur fin de course est en bas du flotteur, donc la bille touche le capteur Cela veut dire que l'arduino va recevoir un signal 0v, se dernier va envoyer un code dans le port série vers l'interface graphique pour afficher que le réservoir est plein ;
- Et dans le cas 2 l'état du réservoir reste plein malgré que le flotteur change sa position car la bille reste collé sur le capteur fin de course ;
- Mais dans le cas 3 le réservoir est vide et le flotteur change sa position, donc le fin de cours deviens en haut du flotteur et la bille relâche ce dernier.

Dans ce cas la carte arduino ne reçoit aucun signal, et d'après la configuration de l'entrée de ce capteur l'arduino va lire que l'entrée est 5V, par la suite cette carte va envoyer un autre code dans le port série vers l'interface graphique pour afficher que le réservoir est vide et au même temps l'écran LCD vas afficher " réservoir d'eau vide " ;

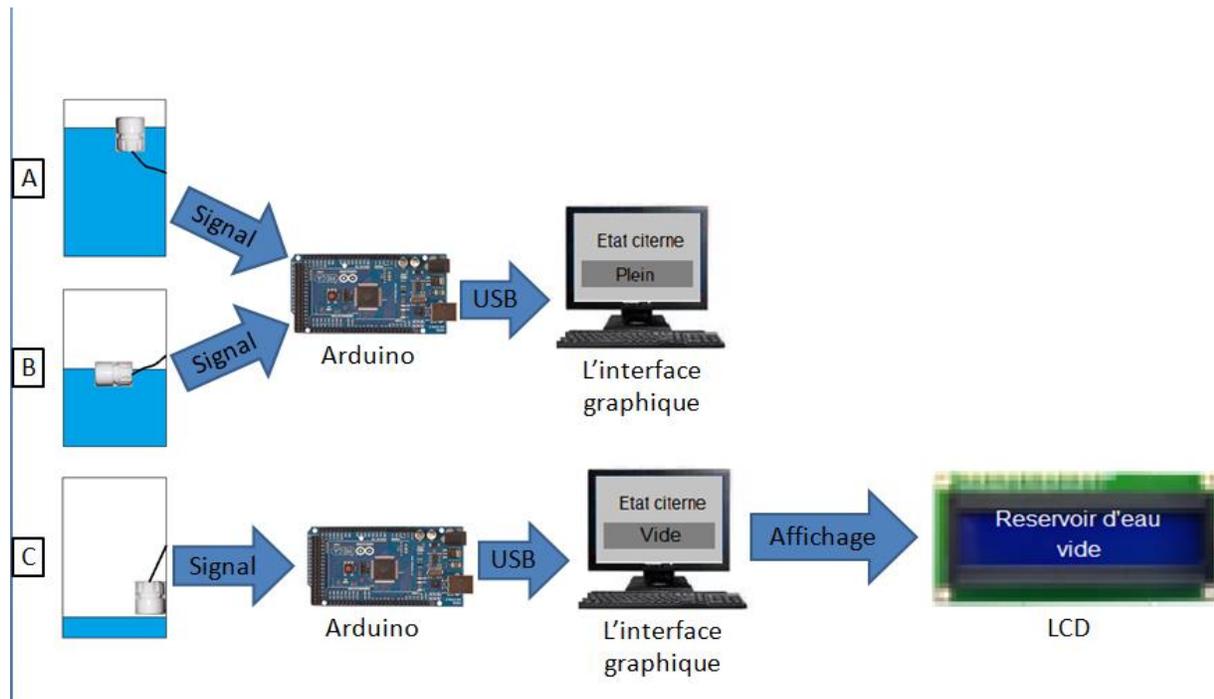


Figure 3.67 : Le scénario de surveillance du niveau d'eau dans le réservoir

3.3 Cout des composants

En voyant les prix très élevés des composants qu'on devrait utiliser dans notre projet, on a décidé de fabriquer quelques composants par nous-mêmes tel que le flotteur à titre d'exemple. Cela nous a permis d'économiser le maximum possible et rester dans l'énorme du budget tracé pour ce projet.

- **Le coût de la construction de la maison en bois :**

L'achat des pièces en bois, la manœuvre, la peinture et d'autres charges supplémentaires ont coûté presque un montant de 10 Dollars Au total le projet nous a coûté environ **108 Dollars**

NB : Les prix des composants électroniques utilisés dans tout le projet sont représentés dans le tableau que vous trouverez dans l'annexe 2

Conclusion

Ce chapitre décrit l'ensemble des étapes suivi toute au long de la conception et la réalisation du prototype de la maison intelligente, ainsi que l'étude financier du cout total du projet.

Conclusion générale

L'électronique, les technologies de communication et l'informatique ont connu un large développement et font aujourd'hui partie de tous les systèmes intelligents, les systèmes domotiques étant des systèmes intelligents sont basés essentiellement sur ces outils.

Dans ce cadre, nous avons essayé de développer un système domotique et l'implanter dans un prototype, ce système permet aux utilisateurs de piloter, contrôler et de surveiller les dispositifs domestiques localement ou à distance, en utilisant une cartes Arduino Méga comme cerveau de notre système.

Ce travail élaboré dans le cadre de notre projet de fin d'études, nous a été très bénéfique dans la mesure où il nous a permis de consolider nos connaissances académiques ainsi que l'ouverture à un domaine technologique très prometteur et innovant.

Cette tâche ne nous a pas été assez simple, nous avons bien sur rencontré certaines contraintes à savoir l'indisponibilité du matériel, le cout élevé des équipements, la documentation ainsi que les difficultés rencontres lors de la construction de la maison en bois.

Il est évident que certaines technologies domotiques s'avèrent très utiles, voire nécessaires dans certains cas par exemple, pour les personnes handicapées la domotique leur permet une liberté de mouvement et d'action qu'ils n'auraient pas sans elle. Elle leur permet aussi un plus grand contrôle sur leur vie.

Pour ce qui est des perspectives d'amélioration, plusieurs voies peuvent être envisagées. À titre indicatif, nous pensons que nous pouvons développer les points suivants :

- Passer du réseau local à internet.
- Regrouper toutes les applications de commandes dans une application android sur un smart phone.
- Ajouter d'autres fonctionnalités à notre système domotique tel que la gestion des multimédia, la fonctionnalité anti incendie....
- Améliorer la fonctionnalité contrôle d'accès
- Innover des solutions visant plus de confort chez l'habitant.

Références Bibliographiques

www.locoduino.org/spip.php?article97

<https://studylibfr.com/doc/1238702/un-relais-est-un-commutateur-%C3%A9lectrique-qui-permet-de-com...>

<https://www.framboise314.fr/scratch-raspberry-pi-composants/module-detecteur-de-presence-pir/>

<https://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet>

<https://eskimon.fr/tuto-arduino-602-un-moteur-qui-a-de-la-tête-le-servomoteur>

<https://www.instructables.com/id/Visual-Basic-Arduino-Interface/>

http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.MaterielMega2560

<http://tpe-domotique-ia.blogspot.com/2015/03/lorigine-de-la-domotique.html>

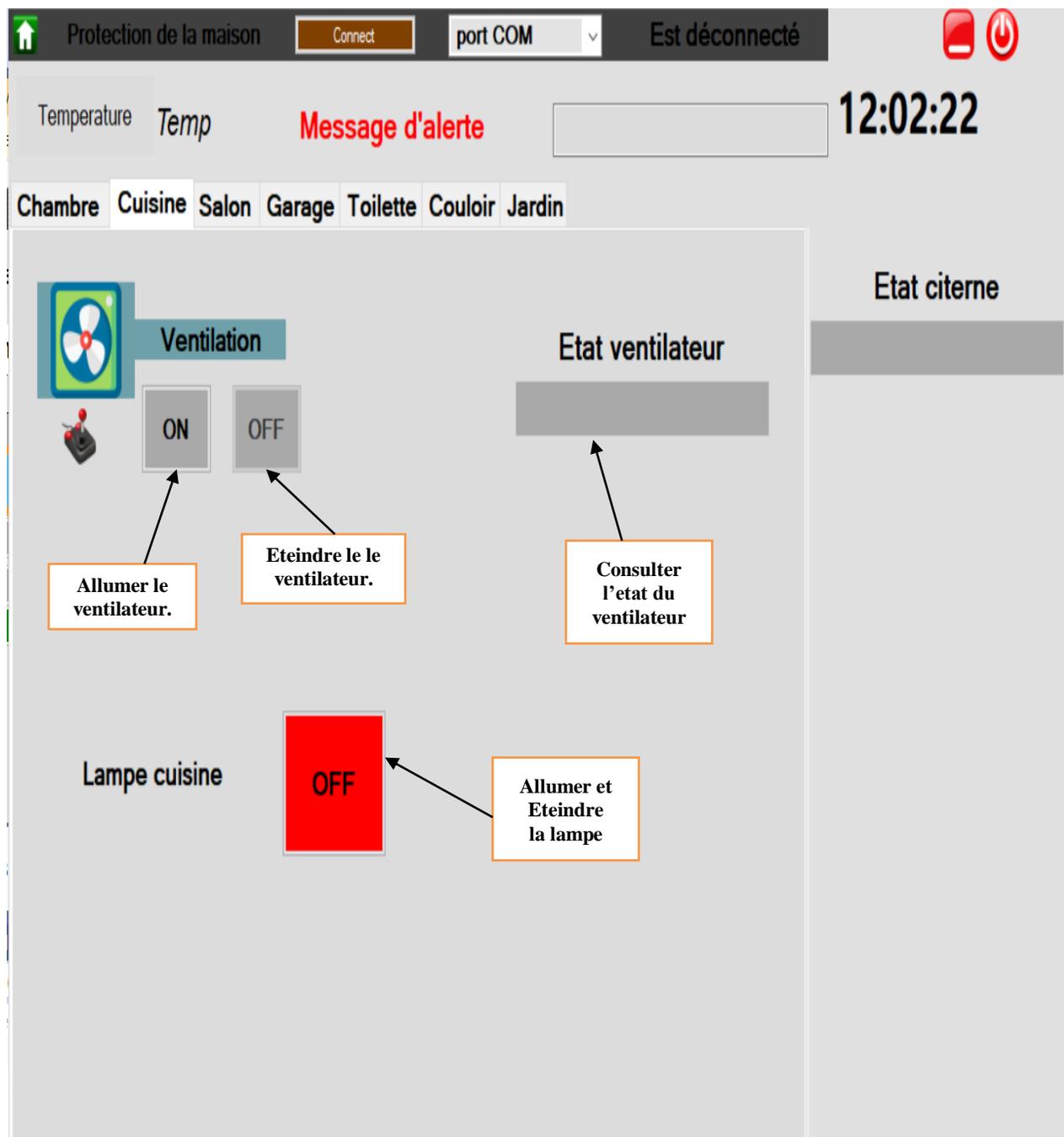
<https://fr.wikipedia.org/wiki/Domotique>

<https://www.arduino.cc/>

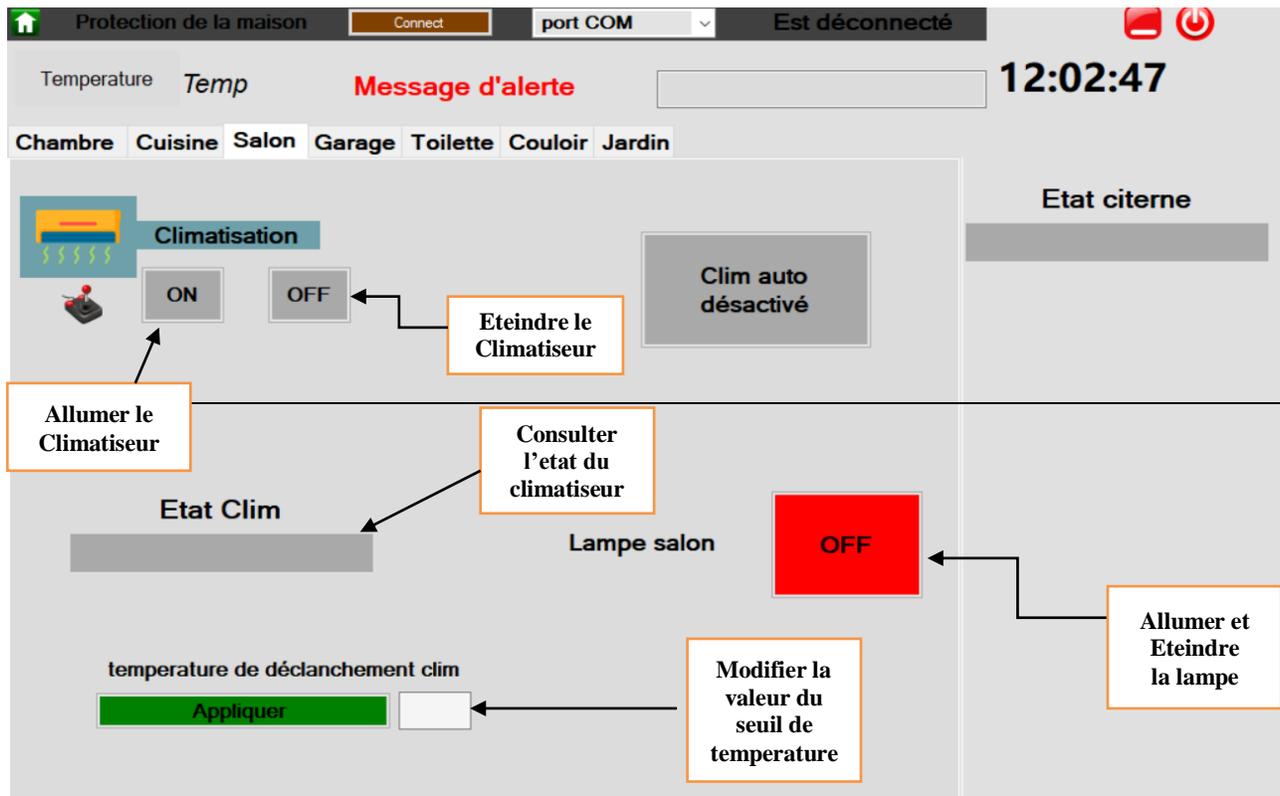
<https://forum.arduino.cc/index.php?topic=44151.0>

Tous les liens sont consultés entre 15-03-2019 et 30-06- 2019

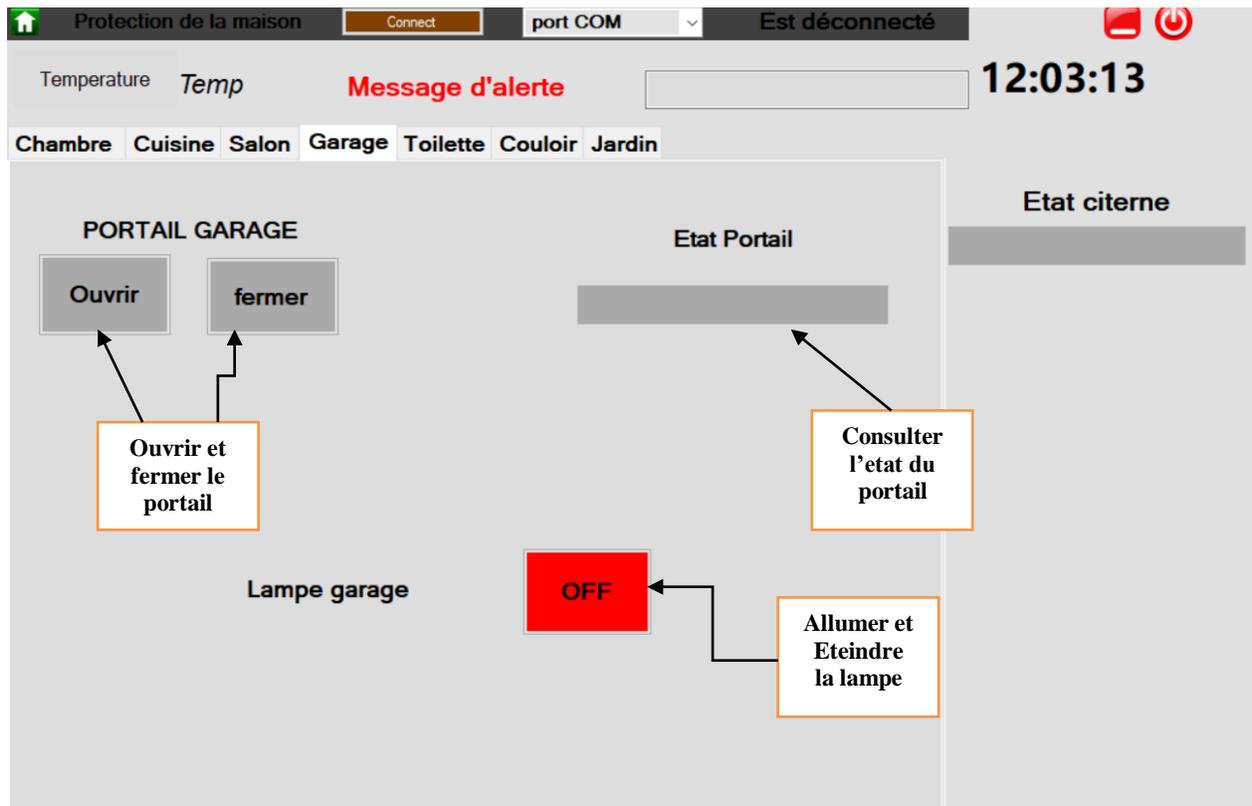
ANNEXI : FONCTIONNEMENT DE L'APPLICATION DE COMMANDE



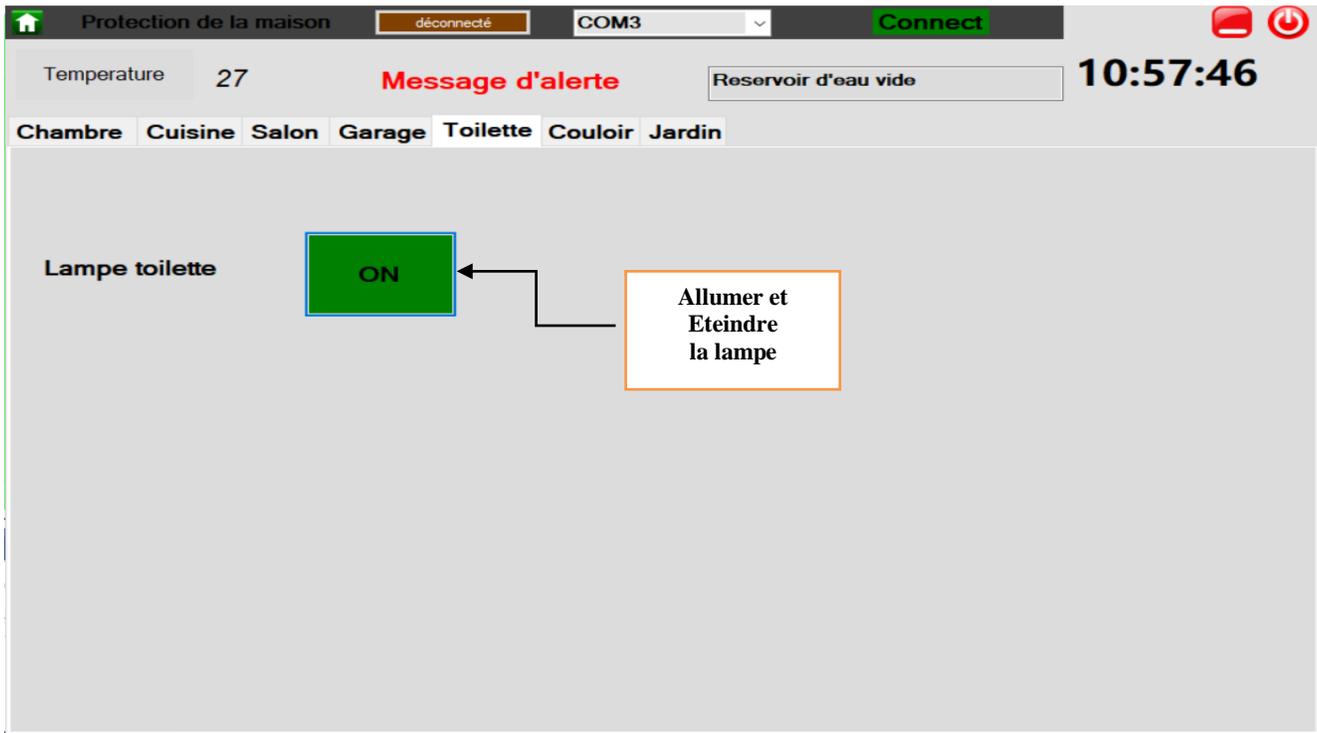
Description de l'interface graphique (cuisine)



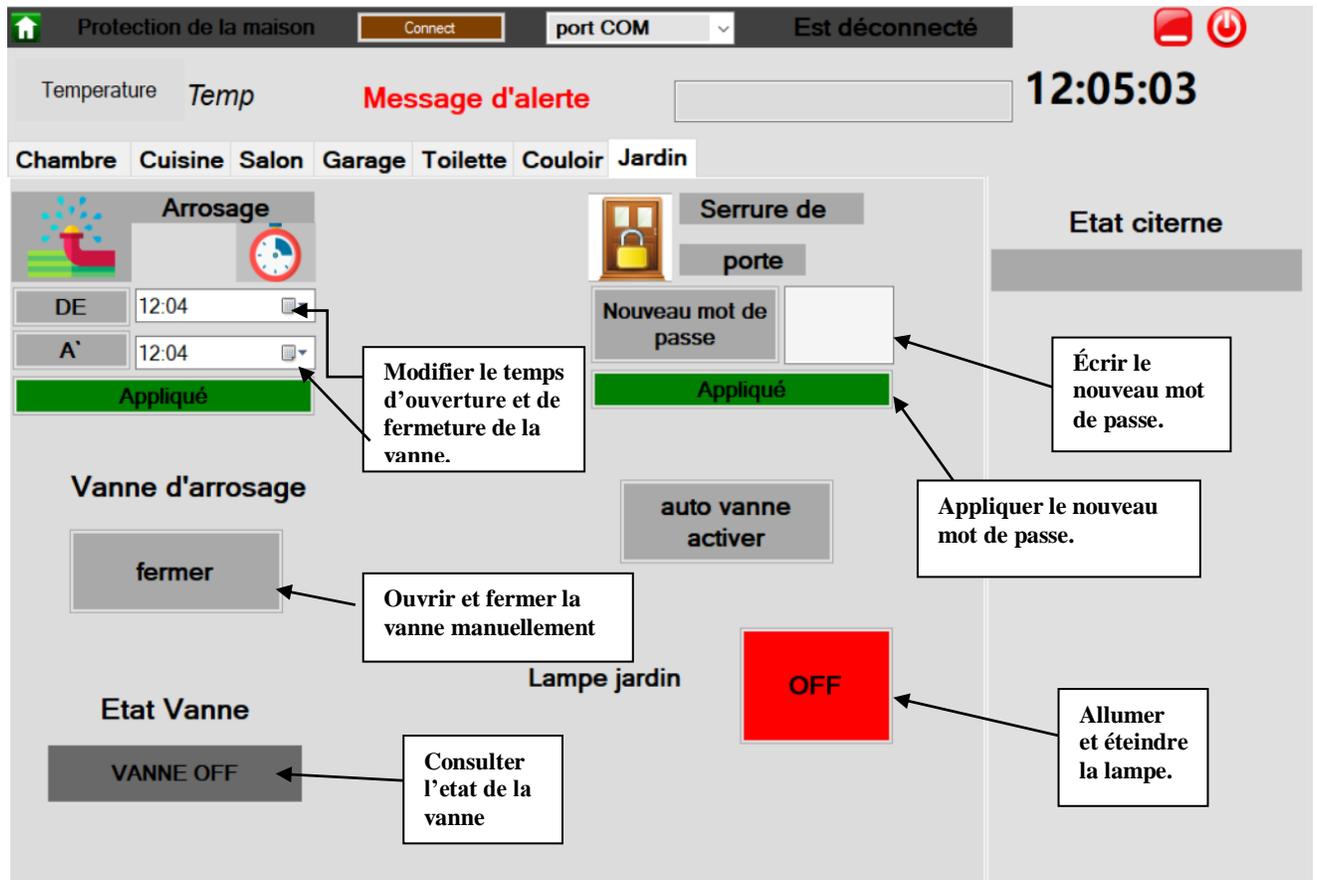
Description de l'interface graphique (Salon)



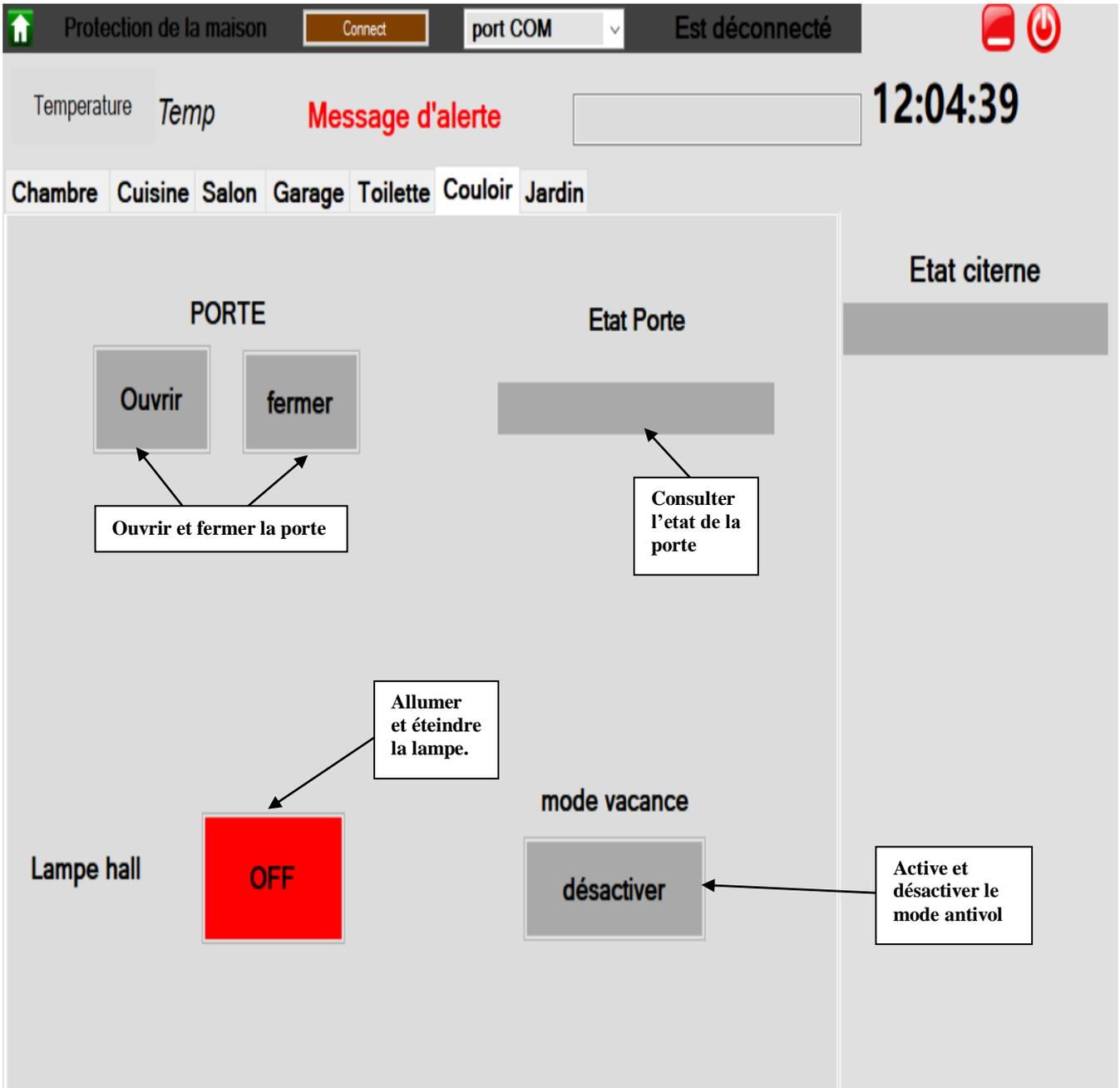
Description de l'interface graphique (Garage)



Description de l'interface graphique (toilette)

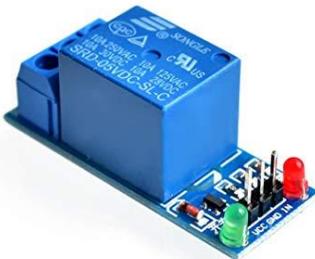


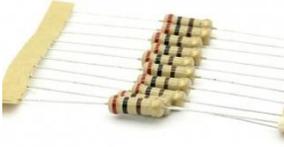
Description de l'interface graphique (Jardin)

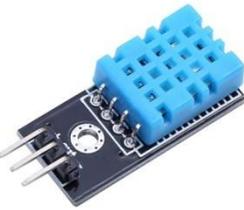


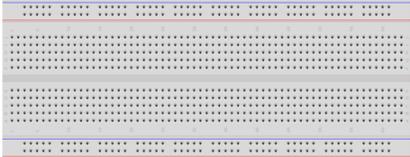
Description de l'interface graphique (couloir)

ANNEX 2 : DEVIS DU PROJET

	Nbr	Image	Prix
Arduino mega 2560	1		15 \$
RTC DS3231	1		4 \$
Capteur de mouvement HC-SR501	1		3,5 \$
Afficheur LCD (16*2)	1		4 \$
Module relais 1 canal	5		7,5 \$

Lampe 220V	1		0,5 \$
Résistance	12		1 \$
LED	9		1 \$
Ventilateur	2		2,5 \$
Buzzer	1		0,5 \$

Electrovanne Plastique 220V	1		3 \$
Tuyaux	1		0,5 \$
Clavier matriciel (4*4)	1	 <p>12345678</p>	6 \$
Capteur d'humidité et température DHT11	1		3,5 \$

Micro servo SG90 à rotation angulaire	2		5 \$
Récepteur infrarouge KY-022 Avec son télécommande	1		4 \$
Alimentation 0-15V 0-2A DC	1		22 \$
Potentiomètre 10K ω	1		0,3 \$
Téléphone portable SAMSUNG B313E METRO	1		10 \$
Flotteur capteur niveau d'eau	1		0,5 \$
Plaque d'essai	1		2,5 \$

Capteur de gaz MQ-2	1		1,3 \$
Transistor NPN 2N2222	3		0,5 \$

Le prix total des composants : 98,6 Dollars