



جامعة عباس لغرور خنشلة
كلية العلوم و التكنولوجيا
قسم الهندسة الصناعية
Université Abbès LAGHROUR Khenchela
Faculté des Sciences & de la Technologie
Département de Génie Industriel



Mémoire de fin d'études

Pour l'obtention du diplôme de Master (LMD)

Filière : Automatique

Spécialité : Automatique et informatique industrielle

**Réalisation pratique d'un
système Smart, pour la détection
incendie et la sécurité.**

*Réalisé par : -BAKHOUCHE zoubir Dirigé par : Pr. Sofiane BOUOUDEN
-SIAFI Azzedine*

Membres du Jury :

*Dr. Fouad ALLOUANI MCA Président
Mme Hafiza BENFAROUDJ MAA Examineur*

Année universitaire 2020/2021

Remerciement

Avant tout nous remercions notre bon dieu le tout puissant pour le courage et la force et la volonté qui nous a accordé pour accomplir ce travail.

Nous tenons à exprimer toute notre gratitude et nos profonds remerciements à notre prometteur Pr Sofiane BOUOUDEN, pour nous avoir suivis dans l'élaboration de ce travail et pour toute la confiance qu'il nous a sus témoigner durant cette étude.

Nous tenons aussi à exprimer nos plus grands respects et nos vifs remerciements aux membres du jury qui ont bien voulu examiner ce travail.

Un mercîment très respectueux tous nos enseignants de l'université de kenchela.

Nos remerciements les plus chaleureux et les plus profonds s'adressent à notre famille et nos amis pour leurs soutiens, leurs encouragements et leurs patiences.

Enfin, nous remercions tous ceux qui ont nous aidés de près de loin à réaliser ce travail.

Zoubir ,Azzedine

Dédicace

Je dédie ce travail à :

- Mes parents,
- Mes frères et mes sœurs,
- Mr Zakaria Kenzari
- Ma famille,
- A mes amis,
- La promotion automatique 2020 / 2021

B. Zoubir

Dédicace

Je dédie ce travail à :

- Ma mère ,
- Ma femme et mes enfants ,
- Mes frères et mes sœurs,
- Mr Zakaria Kenzari
- Ma famille,
- A mes amis,
- La promotion automatique 2020 / 2021

S.Azzedine

Résumé :

La création d'une solution domotique pour le particulier s'est révélée être indispensable pour combler l'insatisfaction du public en quête de services et de fonctions facilitant sa vie quotidienne à domicile. Ce projet a pour objectif de réaliser un système qui peut intégrer dans une maison intelligente permettant le pilotage, la commande et la surveillance des différents dispositifs disponibles dans cette dernière. Elle est capable de gérer et de prévenir en cas d'incendie ou en cas d'une flamme, il surveille également l'état de la température et de l'humidité et nous faire parvenir par SMS à l'aide une carte GSM SIM900, ainsi l'ouverture et la fermeture de la porte à distance; à travers deux méthodes, soit par une carte, soit par SMS, D'autre part, nous avons programmé les composants nécessaires à l'aide d'ARDUINO IDE.

A la fin de ce projet, nous avons un aperçu complet du processus de réalisation d'un système domotique, des compétences et des connaissances qui y sont attachées.

Mots clés: Domotique, Carte Arduino, GSM, PID, Détecteur d'incendie, détecteur de mouvement, détecteur, de température, RFID mrfc522 .

ملخص:

لقد أثبت إنشاء حل التشغيل الآلي للمنزل للفرد أنه ضروري لمعالجة استياء الجمهور بحثاً عن الخدمات والوظائف التي تجعل حياتهم اليومية في المنزل أسهل. الهدف من هذا المشروع هو إنتاج نظام يمكن دمجه في منزل ذكي يسمح بتجربة الأجهزة المختلفة المتوفرة في الأخير والتحكم فيها ومراقبتها. إنه قادر على الإدارة والوقاية في حالة نشوب حريق أو في حالة وجود لهب ، كما أنه يراقب حالة درجة الحرارة والرطوبة ويرسل إلينا عبر الرسائل القصيرة باستخدام بطاقة GSM SIM900 ، وبالتالي فتح وإغلاق الباب عن بعد ؛ من خلال طريقتين ، إما عن طريق البطاقة أو عن طريق الرسائل القصيرة ، ومن ناحية أخرى ، قمنا ببرمجة المكونات الضرورية باستخدام ARDUINO IDE في نهاية هذا المشروع ، كان لدينا نظرة عامة كاملة على عملية بناء نظام أتمتة المنزل ، والمهارات والمعرفة المرتبطة به.

الكلمات المفتاحية : أتمتة المنزل ، لوحة Arduino ، GSM ، PID ، كاشف الحريق ، كاشف الحركة ، الكاشف ، درجة الحرارة ، RFID mrfc522 .

Sommaire :

Sommaire :	N
Introduction générale	01
Chapitre I :La domotique	
I.1. Introduction	03
I.2. Définition	03
I.3. Fonctionnement d'un système domotique	04
• I.3.1. Les capteurs	04
• I.3.2. les caractéristiques d'un capteur	05
• I.3.3. Les types des capteurs	05
• I.3.4. Les familles des capteurs :	06
• I.3.5. Actionneurs	07
• I.3.6. Relation capteurs actionneurs :	07
I.4. Maison connectée	08
I.5. Différence entre domotique et maison connectée	08
I.6. Les fonctions de la domotique	08
• I.6.1. La fonction de Sécurité	09
• I.6.2. La fonction de surveillance	09
• I.6.3. La fonction de gestion d'énergie	09
• I.6.4. La fonction de scénarisation	10
• I.6.6. La fonction de confort	11
I.7. Les technologies utilisées pour la domotique	11
• I.7.1. La technologie x10	11
• I.7.2. La technologie InOne de Legrand	13
• I.7.3. Le système Chacon	14
I.8. Les avantages et les inconvénients	14
• I.8.1. Les avantages	14
• I.8.2. Les inconvénients	14
I.9. Système automatisé	15
• I.9.1. Définition	15
• I.9.2. Les composants d'un système automatisé	15
• I.9.3. Représentation simplifiée d'un système automatisé	16
I.10 Le GSM :	17
• I.10.1. Historique	17
• I.10.2. Présentation	17
• I.10.3. Architecture de GSM	18
• I.10.3.1. Le sous-système radio(BSS)	19
• I.10.3.2. Le sous-système réseau(NSS)	19
• I.10.3.3. Le sous-système exploitation OSS(Opération Sub-System)	20
I.11. Principe de fonctionnement de la domotique par GSM	20
• I.11.1. Commande par GSM	20
• I.11.2. Surveillance par GSM	20
I.12. Conclusion:	20
Chapitre II : Matériels et logiciels utilisés	
II.1. Introduction	21
II.2. Généralité sur les microcontrôleurs	21
II.3. Historique de l'Arduino	23
II.4. Définition de Arduino	24

II.5. Propriétés de système Arduino	24
II.6. Constitution de la carte Arduino UNO	24
II.6.1. Partie matérielle	24
II.6.1.1. Microcontrôleur ATmega328	24
II.6.1.2. Les sources de l'alimentation de la carte	25
II.6.1.3. Les entrées et sorties	25
II.6.1.4. les accessoires de Arduino	27
II.6.1.4.1. Plaque d'essai	28
II.6.1.4.2. Câble USB	28
II.6.1.4.3. Les capteurs	29
• Capteur de flamme KY-026	29
• capteur ultrason (HC-SR04)	30
• Capteur d'humidité et de température DHT11	32
• Capteur de gaz et de fumée (MQ-2)	33
• Capteur de mouvement PIR HC-SR501	34
• RFID MFRC522	35
II.6.1.4.4. Les actionnaires	38
• LED	38
• Buzzer	39
• Servomoteur	40
• Afficheur LCD 16x2 I2C	41
II.6.1.4.5. Module de communication	42
• Le shield SIM900a GSM/GPRS	43
II.6.2. Partie Logicielle	48
• II.6.2.1 Le logiciel Arduino	48
• II.6.2.2 Présentation de «IDE »	49
• II.6.2.3 Structure générale du programme (IDE Arduino)	49
• II.6.2.4 Injection du programme	50
• II.6.2.5 Fritzing	51
II.7. Conclusion	52
Chapitre III : PID Sous ARDUINO	
III.1. Introduction	54
III.2. Asservissement et régulation	54
III. 3. Commande PID	54
III.3.1. Régulateur proportionnel P	55
III.3.2. Régulateur Dérivé I	56
III.3.3. Régulateur Dérivé D	57
III.3.4. La méthode de réglage d'un correcteur PID	58
III.4. Implémentation du correcteur PID avec Arduino	58
III.4.1. La fonction PID()	59
III.5. Conclusion	63
Chapitre IV : réalisation pratique d'un système intelligent	
IV.1. Introduction	65
IV. 2. Test de matérielle utilisée :	66
• Test la carte arduino UNO :	66
• Exemple 1 : Les feux de circulation	67
IV.3. Réalisation du projet :	70
Partie 1 : Projet détecteur de mouvement	70
Partie 2 : Détecteur d'incendie (Flamme)	72
Partie 3 : Détecteur de température et l'humidité :	74
IV.4. Réalisation pratique des projets	77

IV.4.1 - Projet de détection d'incendie et de sécurité :	77
IV.4.2- Projet de RFID : ouverture de la porte	79
IV.5. Conclusion	83
Conclusion générale	85
Référence	87
ANNEX	90

Liste de figures:

Chapitre I :La domotique

Le titre de figure	Page
Figure I.1: Domotique	04
Figure I.2: Schéma Fonctionnel d'un capteur	05
Figure I.3 relation capteurs actionneurs	07
Figure I.4 Différentes fonctions de la domotique.	09
Figure I.5 Logo X10	11
Figure I.6 Logo Legrand	13
Figure I.7 Logo Chacon	13
Figure I.8 : Exemple d'un système automatisé	16
Figure I.9 : Architecture d'un réseau GSM	19

Chapitre II :Matériels et logiciels utilisés

Le titre de figure	Page
Figure II.1: Les microcontrôleurs	22
Figure II.2: Les créateurs de Arduino	23
Figure II.3 ATmega328	25
Figure II.4: Constitution de la carte Arduino UNO	27
Figure II.5 : Fils	27
Figure II.6 : Plaque d'essai	28
Figure II.7 : câble USB	28
Figure II.8: capteur de flamme KY-026	29
Figure II.9: Branchement de flamme KY-026 avec Arduino	29
Figure II.10: Capteur Ultrason HC-SR04	30
Figure II.11 : Principe de fonctionnement d'un capteur Ultrason HC-SR04	31
Figure II.12 : capture DHT11	32
Figure: II.13 Capteur de gaz (MQ-2)	33
Figure II.14: PIR HC-SR501	34
Figure II.15: Module RFID MFRC522	35
Figure II.16 :Présentation des Tags, label ou transpondeur	35
Figure II.17: Présentation de "INLAY"	36
Figure II.18: Les composants d'un système RFID	36
Figure II.19: le couplage magnétique et électrique entre le tag et lecteur RFID	37
Figure II.20: branchement de RFID MFRC522 avec la carte Arduino UNO	38
Figure II.21: Schéma et sens du courant des LED	39
Figure II.22 : buzzer	39
Figure II.23 : servomoteur	40
Figure II.24 : Branchement de servomoteur avec Arduino	41
Figure II.25 : LCD 16x2	41
Figure II.26 : I2C	42
Figure II.27 : Branchement de I2C avec la carte Arduino .	42
Figure II.28 : Présentation du matériel du blindage SIM900a GSM/GPRS	43
Figure II.29 :Les trois LED de SIM900a	44
Figure II.30 : le protocole UART pour communiquer avec un Arduino	46

Figure II.31 : SIM900 Shield Branchement avec Arduino	47
Figure II.32 : Schéma de fonctionnement des commandes AT	47
Figure II.33: Interface IDE Arduino 1.8.13	49
Figure II.34: Barre de boutons IDE Arduino	50
Figure II.35: Paramétrage de la carte Arduino UNO	50
Figure II.36 : ajouter une bibliothèque	51
Figure II.37 : L'interface de Fritzing	52

Chapitre III :PID Sous ARDUINO

Le titre de figure	Page
Figure III.1 : Boucle de commande d'un système asservis.	54
Figure III.2 : Schéma bloc d'un système avec correcteur PID	55
Figure III.3 : Modélisation de l'effet du régulateur P sur la réponse à un échelon d'un système.	56
Figure III.4 : Modélisation de l'effet du régulateur I sur la réponse à un échelon d'un système.	57
Figure III.5 :Modélisation de l'effet du régulateur PID sur la réponse à un échelon d'un système.	58
Figure III.6 : Réponse fréquentielle du système.	59
Figure III.7 : envoi du code vers l'Arduino	60
Figure III.8 : Exécution du programme.	60
Figure III.9 : Traçage des réponses.	61
Figure III.10 : système oscillant.	61

Chapitre IV :réalisation pratique d'un système intelligent

Le titre de figure	Page
Figure IV.1 : ouvrir le exemple de BLIK	66
Figure IV.2 : Les modification sur le programme de BLINK	67
Figure IV.3 : un exemple de feux de circulation réaliser	68
Figure IV.4 : schéma de câblage de Les feux de circulation	68
Figure IV.5: schéma de câblage de détecteur de mouvement	71
Figure IV.6 : Organigramme de la fonction de détection de mouvement	72
Figure IV.7: schéma de câblage de détecteur d'incendie	73
Figure IV.8: Organigramme de la fonction de détection flamme	74
Figure IV.9 :schéma de câblage de détecteur de température et l'humidité	76
Figure IV.10 : Organigramme de la fonction de détection température	76
Figure IV.11: schéma de câblage de Projet de la détection incendie et la sécurité	77
Figure IV.12: réaliser montage de projet de la détection incendie et la sécurité	78
Figure IV.13: schéma de câblage de projet d'ouverture de porte	80
Figure IV.14 : Organigramme de la fonction d'ouverture de porte	81
Figure IV.15: réaliser montage de projet d'ouverture de la porte	82

Liste des tableaux :

Chapitre I :La domotique

Le titre de tableau	Page
Tableau I.1 : Tableau des capteurs actifs	05
Tableau I.2 : Tableau des capteurs actifs .	06

Chapitre II :Matériels et logiciels utilisés

Le titre de tableau	Page
Tableau II.1: Caractéristique de Ultrason HC-SR04	31
Tableau II.2 : Connectique du module RFID-MFRC522	38
tableau II.3: Caractéristique de alimentation SIM900 Shield	45
Tableau II. 4 : Commande AT dédiées service SMS	48

Chapitre III :PID Sous ARDUINO

Le titre de tableau	Page
Tableau III. 1 : Variation des réponses des systèmes corrigé et non corrigé à la variation de gain.	62

Liste des abréviations

SMS	Short Message Service
TOR	tout ou rien
GSM	Groupe Spécial Mobile
CPL	Courants porteurs en ligne
CEPT	Conférence Européenne des Postes et Télécommunication
MOU	Memorandum Of Understanding
SFR	Société Française de Radiophonie
DCS	Digital Communication System
PCS	personale communications services
GPRS	General Packet Radio Services
EDGE	Enhanced Data rates for GSM Evolution
BSS	Base Station Subsystem
BTS	Base Transceiver Station
TRX	Transmission/Reception Unit
BSC	Base station controller
NSS	Base transceiver station
MSC	Mobile service Switching Center
HLR	Home Location Register
VLR	Visitor Location Register
MSC	Mobile-services Switching Centre
OSS	Opération Sub-System
OMC	Opération and Maintenance Center
NMC	Network Management Center
MCU	Multipoint Control Unit
USB	Universal Serial Bus
FTDI	Future Technology Devices International Ltd
SPI	Interface Série Périphérique
PIR	Peak information rate
RFID	Radio frequency identification
TTL	Transistor Transistor Logic
LED	light- emitting diode
LCD	Liquid Crystal Display
I2C	Inter-Integrated Circuit

SDA	Serial DAta
SCL	Serial CLock
SIM	Subscriber identity/ identification module
UART	Universal Asynchronous Receiver Transmitter
FM	Frequency Modulation
RTC	Réseau téléphonique commuté
IDE	Integrated development environment
GUI	Graphical User Interface

Introduction Générale

INTRODUCTION GENERALE

Introduction générale

De nos jours 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. C'est pourquoi l'industrie a concentré ses expériences sur le développement de microcontrôleurs, d'interfaces et d'outils qui assurent le confort, la sécurité et l'assistance au sein d'un édifice.

Ce projet se découpe en deux parties faisant appel à nos compétences en Automatique et Informatique industrielle et en Electronique. Il 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), la sécurité (protection contre les intrusions, détection d'incendie...), la communication Par des messages et des appels.

Dans la première partie de ce travail nous avons donné une généralité sur la domotique et la **gestion centralisée des équipements techniques** (chauffage, sécurité, etc.) et du **multimédia** dans le résidentiel. Elle vise à apporter des fonctions de confort, de sécurité, d'économie d'énergie et de communication aux maisons ou appartements équipés.

La présentation de notre travail elle est décomposée de quatre chapitres :

Dans le premier chapitre nous avons présenté les phases de la mise en place de notre système domotique, sa construction et les performances théoriques de la domotique et du GSM en général. Ce chapitre nous permet de nous familiariser avec le matériel utilisé pour la réalisation du projet et de façon plus particulière afin d'obtenir une meilleure maîtrise du travail de domotique.

Dans le deuxième chapitre nous avons présenté une généralité sur La carte Arduino et ces différents types ensuite nous citons quelques exemples sur des modules et les capteurs adaptés à la carte d'Arduino et leurs caractéristiques principales, le logiciel utilisé ainsi que leur langage de programmation.

Dans le troisième chapitre, nous expliquerons les principes de base d'un contrôleur PID et leurs caractéristiques. Nous allons utiliser une carte Arduino comme un contrôleur PID, et nous l'avons testé en tenant compte de plusieurs valeurs de l'action proportionnelle.

Dans quatrième chapitre nous détaillerons les phases de la mise en place de nos systèmes de commande et de sécurité, sa construction et les organigrammes décrivant le raisonnement du programme de commande implémenté sur les cartes Arduino.

Nous finirons ce rapport par une conclusion générale récapitulative des différentes phases de notre travail.

Chapitre I :

La domotique

I.1. Introduction :

La domotique est le domaine technologique qui traite de l'automatisation du domicile, d'où l'étymologie du nom qui correspond à la contraction des termes "domicile" et "automatique" [1].

Les développements des technologies du bâtiment et de la domotique ont permis de maîtriser les économies d'énergie, de contrôler facilement à distance les appareils électroménagers de toutes sortes, de surveiller, d'assurer la sécurité et de recevoir des informations à distance, à chaque instant.

Dans ce chapitre, nous parlerons de la domotique et de ses technologies .

I.2. Définition:

Le terme est un peu obscur mais sa définition nous éclaire sur les propriétés de la domotique. Ce mot issu du latin "Domus", qui signifie maison, regroupe l'ensemble des techniques et technologies électroniques, informatiques et des télécommunications permettant d'automatiser et d'améliorer les tâches au sein d'une maison. Les appareils de la maison sont intégrés au sein de systèmes qui doivent communiquer entre eux afin de gérer des automatismes. Car la domotique milite avant tout pour l'amélioration du quotidien au sein de notre habitat. la domotique vise à apporter des solutions techniques pour répondre aux besoins de confort, de sécurité et de communication que l'on peut trouver dans les maisons, les hôtels, les lieux publics...etc.

A l'origine, la domotique avait donc pour but d'automatiser sa maison : ouverture et fermeture automatiques des volets, ouverture du portail électrique, gestion du chauffage, gestion de l'éclairage, etc. Ainsi avant l'ère des Smartphones, il était par exemple possible d'activer son chauffage à distance en passant un coup de téléphone à sa maison, ou encore en lui envoyant un SMS. C'était tout à fait réalisable. Seulement une telle installation était relativement compliquée à mettre en place et, il faut bien l'avouer, couteuse. Cette époque a malheureusement laissé des traces, puisque pour beaucoup encore aujourd'hui, domotique rime avec cher et compliqué. Pourtant, ce domaine a énormément évolué et il existe de nombreuses solutions simples à mettre en place et tout à fait abordables pour le grand public .

La domotique a surtout elle-même évolué, si bien que le terme est quelque peu dépassé. La domotique servait à automatiser sa maison ; aujourd'hui on parle de domotique 2.0, ou de « maison intelligente », pour bien marquer l'évolution de ce monde. Les différents domaines de la maison ne se contentent plus d'être automatisés et pilotables, ils communiquent ensemble, permettant à la maison de réagir selon différents évènements [2].



Figure I.1 : Domotique

I.3. Fonctionnement d'un système domotique :

Le principe de fonctionnement d'un système domotique contient certains dispositifs pour récupérer les informations pour commander une maison, et pour cela un système domotique fait appel aux capteurs et aux actionneurs [2].

I.3.1. Les capteurs :

Un capteur est un dispositif transformant une grandeur physique à une autre grandeur utilisable, tel qu'une tension électrique ou une intensité ; On fait souvent la confusion entre capteur et transducteur : le capteur est au minimum constitué d'un transducteur. Le capteur est souvent le premier dispositif de la chaîne d'acquisition, il s'agit d'une simple interface entre un processus physique et une information manipulable [2].

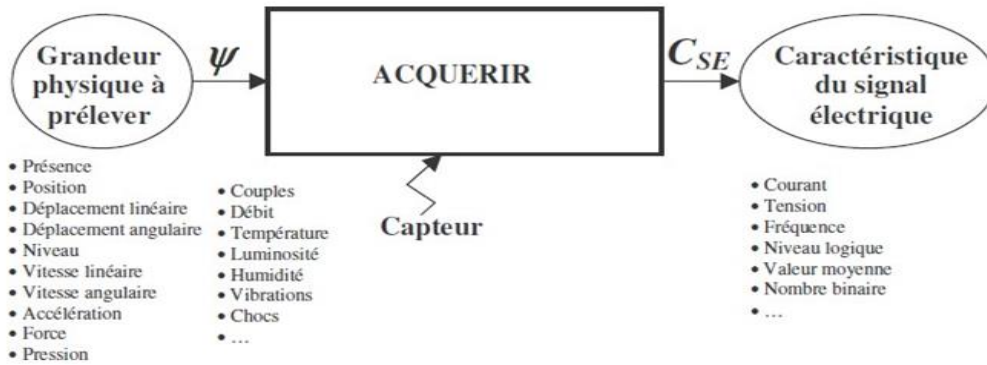


Figure I.2: Schéma Fonctionnel d'un capteur

I.3.2. les caractéristiques d'un capteur :

- Etendue de mesure : Valeurs extrêmes pouvant être mesurée par le capteur.
- Résolution : Plus petite variation de grandeur mesurable par le capteur.
- Sensibilité : Variation du signal de sortie par rapport à la variation du signal d'entrée.
- Précision : Aptitude du capteur à donner une mesure proche de la valeur vraie.
- Rapidité : Temps de réaction du capteur. La rapidité est liée à la bande passante.
- Linéarité : représente l'écart de sensibilité sur l'étendue de mesure [2].

I.3.3. Les types des capteurs :

Si l'on s'intéresse aux phénomènes physiques mis en jeu dans les capteurs, on peut classer ces derniers en deux catégories :

- Capteurs actifs :

Fonctionnant en générateur, un capteur actif est généralement fondé dans son principe sur un effet physique qui assure la conversion en énergie électrique de la forme d'énergie à la grandeur physique à prélever : énergie thermique, mécanique ou de rayonnement [1].

Mesurable	Caractéristique électrique sensible	Type de matériaux utilisés
Température	Constant diélectrique	Metaux: Plaine, nickel, cuivre...
Flux lumineux	Résistivité	Semi-conducteur
Déformation	perméabilité magnétique	Alliages de nickel, silicium
Position	Résistivité	Matériaux magnéto résistants: bismuth, antimoine d'indium
Humidité	Constant diélectrique	Alumine, polymères
Niveau	Constant diélectrique	Liquid esisolants

Tableau I.1 – Tableau des capteurs actifs [2] .

- **Capteurs passifs:**

Ils ont besoin dans la plupart des cas d'apport d'énergie extérieure pour fonctionner .Ce sont des capteurs modélisables par une impédance, une variation du phénomène physique étudié

engendre une variation de l'impédance. Il faut leur appliquer une tension pour obtenir un signal de sortie [1].

Mesurable	Caractéristique électrique sensible	Type de matériaux utilisés
Température	constant diélectrique	Metaux : Plaine, nickel, cuivre...
Flux lumineux	Résistivité	Semi conducteur
Déformation	Perméabilité magnétique	Alliages de nickel, silicium
Position	Résistivité	Matériaux magnéto résistants: bismuth, antimoine d'indium
Humidité	Constant diélectrique	Alumine, polymères
Niveau	Constant diélectrique	Liquide isolants

Tableau I.2 – Tableau des capteurs actifs [2].

I.3.4. Les familles des capteurs :

Les capteurs sont des composants d'automatisme qui ont pour but de récolter une information sur la partie opérative et de la retransmettre à la partie commande qui pourra ainsi la traiter, et on trouve trois familles essentielles :

Capteur TOR (tout ou rien) :

Les capteurs Tout ou Rien (TOR) délivrent une information binaire à la partie commande : l'information adopte l'état 0 ou l'état 1. Chaque état possède une signification dans le contexte du système. On distingue essentiellement les capteurs de type mécanique et ceux de proximité (cellules, inductifs ou capacitifs) [2].

ils sont constitués :

- d'un élément sensible mécanique ou électrique.
- d'un ou plusieurs contacts de type normalement fermé ou de type normalement ouvert.[3]

Capteurs analogiques :

La sortie est une grandeur électrique dont la valeur est une fonction de la grandeur physique mesurée par le capteur. La sortie peut prendre une infinité de valeurs continues.

Le signal des capteurs analogiques peut être du type :

- sortie tension .
- sortie courant .
- règle graduée, cadran, jauge (avec une aiguille ou un fluide) [3].

Capteurs numérique :

La sortie est une séquence d'états logiques qui, en se suivant, forment un nombre. La sortie peut prendre une infinité de valeurs discrètes. Le signal des capteurs numériques peut être du

type :

- train d'impulsions, avec un nombre précis d'impulsions ou avec une fréquence précise
- code numérique binaire .
- bus de terrain [3].

I.3.5. Actionneurs :

Dans une machine ou un système de commande, semi-automatique ou automatique, un actionneur est l'organe de la partie opérative qui, dès qu'il reçoit un ordre de la partie commande via un éventuel capteur ou pré-actionneur, convertit l'énergie qui lui est fournie en un travail utile à l'exécution de tâches, éventuellement programmées, d'un système automatisé. En d'autres termes, un actionneur est l'organe fournissant la force nécessaire à l'exécution d'un travail ordonné par une unité de commande.

Il existe trois types d'actionneur :

- Actionneur électrique.
- actionneur pneumatique.
- actionneur hydraulique.[2]

I.3.6. Relation capteurs actionneurs :

Les capteurs sont des informateurs traducteurs, ils détectent une variation de l'environnement de la partie opérative et la traduit en une information interprétable (grandeur électrique) par la partie commande a fin de contrôler les actionneurs qui transforme une forme d'énergie en une autre pour but d'obtenir l'énergie souhaité (énergie mécanique). comme le montre la figure ci-dessous. [2]



Figure I.3 relation capteurs actionneurs .

I.4. Maison connectée :

C'est une maison dotée de capteurs en tous genres, quasiment invisibles mais permettant la connexion avec votre Smartphone ou votre tablette via internet. Ces capteurs vous permettront de réaliser des économies sans contraintes et sans perte de temps. Grâce à votre Smartphone, on peut recevoir des alertes lorsque des intrus tentent de faire une intrusion, ou lorsqu'une personne âgée maintenue à domicile est en situation de détresse. Gérez les accès internet, l'enregistrement de vos émissions tv, la température de la maison, les programmes des machines à laver sans modération, où que vous soyez et en toute simplicité [2].

I.5. Différence entre domotique et maison connectée

La domotique facilite le quotidien mais avec l'intervention de l'Homme (appuyer sur des boutons pour fermer les volets roulants, gérer votre climatiseur avec la télécommande...etc.). La maison intelligente connaîtra déjà certaines de vos habitudes et s'adaptera pour vous, on parle alors de « machine Learning » ou d'intelligence artificielle [2].

I.6. Les fonctions de la domotique

L'utilisation de plus en plus importante des Smartphones et des Tablettes contribue à favoriser l'acceptation de la domotique au sein de l'habitat.

Les domaines d'application sont au cœur de la vie quotidienne. Les fonctions suivantes peuvent être réalisées grâce aux technologies intégrées dans la domotique : [2]

- Sécurité,
- Surveillance ,
- Gestion de l'énergie,
- Scénarisation des actions,
- Communication,
- Confort.



Figure I.4 – Différentes fonctions de la domotique.

I.6.1. La fonction de Sécurité :

En termes de sécurité, la domotique permet entre autres de :

- Prévenir les risques provenant de l'extérieur (intrusion, cambriolage...) comme ceux provenant de l'intérieur (accidents domestiques) .
- Surveiller à distance vos locaux ou votre habitation depuis votre Smartphone, une Tablette ou un ordinateur .
- Être averti d'un incident (alarmes techniques) par SMS et/ou par e-mail. La vidéo connectée présente des fonctions domotiques (détection de présence, alarme, détecteur de fumée...) et apporte des solutions dans ce domaine.

En cas d'incident, vous, et/ou des personnes de votre choix, peuvent être avertis en temps réel par SMS [5].

I.6.2. La fonction de surveillance :

Pour surveiller un domicile, plusieurs capteurs sont utilisés pour détecter les anomalies :

- Inondation
- Incendie
- Fuite de gaz
- Coupure de courant

La centrale intervient instantanément pour couper les alimentations, remonter les stores, couvrir la piscine, appeler les numéros d'urgence ou faire retentir la sirène si l'occupant est présent [4].

I.6.3. La fonction de gestion d'énergie

La programmation des seuils de température est l'une des principales sources d'économie, elle peut être journalière ou hebdomadaire. Des sondes de température renseignent la

centrale sur les valeurs de température dans chaque pièce .

Elle tient compte de la présence d'un occupant et s'adapte automatiquement à son emploi du temps :

- Extinction des éclairages inutiles
- Réglage de l'intensité lumineuse en fonction de l'activité...

Un simple appel téléphonique suffit pour augmenter la température de chauffage en prévision de l'arrivée des occupants [1].

I.6.4. La fonction de scénarisation

Au moment de quitter un habitat ou un commerce, la mise en fonction de l'alarme déclenche une série de contrôles et d'actions, (centralisation des commandes) :

- Fermeture de toutes les lumières,
- Coupure de l'arrivée de gaz,
- Vérification de la fermeture de toutes les fenêtres,
- Allumage de la lumière extérieure durant quelques minutes s'il fait nuit, ...

À partir d'un bouton unique, tous les éclairages du living seront ajustés pour le dîner, une soirée télévision ou la création d'une ambiance lumineuse adaptée à l'activité de l'occupant [2].

I.6.5. La fonction de communication

Aujourd'hui, une centrale domotique sait communiquer

- Par téléphone (GSM),
- Par ordinateur (Internet),
- Par des interfaces tactiles,
- Par le biais de microphones permettent une activation par commande vocale associés à des logiciels de reconnaissance vocale.

Ceci permet à une personne de recevoir l'état de son installation et d'émettre des alertes et piloter sa maison de n'importe quel endroit du monde, de son bureau ou de sa voiture. Pouvoir diffuser le son et l'image dans votre logement sont des exemples de scénarios qu'offre la domotique. Les équipements vidéo, home-cinéma, réseau téléphonique et internet sont intégrés dans toutes les pièces de votre habitation. Il est possible de gérer et diffuser ses bibliothèques de musiques et de vidéos dans différentes pièces, de sauvegarder ses données informatiques, d'avoir accès à distance à ses ordinateurs, de faciliter la mobilité et le télétravail. Ces systèmes sont en général indépendants et peuvent être pilotés par les fonctions domotiques [2].

I.6.6. La fonction de confort

Ouvrir le portail sans descendre de voiture, allumer automatiquement la lumière en ayant les bras chargés de paquets, arroser automatiquement le jardin, ouvrir ou fermer les volets ou les stores, programmer une ambiance sonore dans la maison, disposer de la télévision dans plusieurs pièces, déclencher la préparation du café avant le réveil, adapter la température aux conditions extérieures, ... autant de fonctionnalités qui peuvent devenir indispensables.

Un émetteur radio, une télécommande infrarouge, un téléphone, une horloge ou un détecteur sont différents dispositifs de commande qui serviront de moyens pour assurer ces fonctions [2].

I.7. Les technologies utilisées pour la domotique :

Généralement, une installation domotique peut être conçue sur trois principaux types de technologie. Ces technologies peuvent cohabiter, être superposées suivant l'évolution de l'installation dans le temps [2].

I.7.1. La technologie x10

Est une technologie CPL ¹ créé dans les années 80. Elle est très utilisée aux Etats-Unis, mais les modules américains ne sont malheureusement pas compatibles avec le réseau 50HZ (France, Afrique du Nord) à cause de la différence de tension secteur/fréquence. En plus de la communication CPL, il existe une partie RF ² qui permet grâce à un traducteur RF/X10 permet de piloter des équipements directement à partir d'une télécommande.

Les inconvénients :

- Lenteur : Les ordres émis en x10 mettent parfois 1s avant d'être exécutés
- Fiabilité : Le manque de retour d'état ne permet pas de contrôler qu'un ordre a été correctement exécuté, et il arrive parfois que certains ordres ne soit pas pris en compte.

Les avantages :

- Moins cher que les autres technologies et facile à installer [2] .



Figure I.5 Logo X10

I.7.2. La technologie InOne de Legrand :

Est une technologie propriétaire. Elle ne peut s'interfacer directement avec les autres technologies.

La gamme InOne regroupe des modules CPL, RF et IR ³. A noter que tous les modules d'une même catégorie sont compatibles entre eux, mais il faut des modules "traducteurs" pour passer de la radio en CPL ou d'IR en CPL par exemple.

Les inconvénients :

— Impossible d'interagir facilement avec les autres technologies. Même si cette technologie est d'un excellent rapport qualité/prix, elle reste chère par rapport au x10 ou PLCBUS ⁴.

1. Technologie permettant de faire circuler des informations numériques en passant par les lignes électriques. Les données sont encodées sur le réseau électrique en se basant sur la fréquence du signal électrique (50Hz) et en superposant un autre signal de fréquence plus élevé.

2. Onde Radio (RF) : La technologie radio qui permet d'interconnecter des équipements L'ajout de capteur ou d'actionneur. Plus la fréquence est élevée, plus la vitesse de transmission est grande, mais plus la distance de transmission est faible.

Les avantages :

- Design très travaillé et Technologie Legrand reconnue pour sa qualité [2].



Figure I.6 Logo Legrand

I.7.3. Le système Chacon :

Est un ensemble domotique basé sur la RF. Tous les émetteurs du système Chacon peuvent contrôler les récepteurs. Par exemple, il est possible de contrôler un éclairage à partir d'un interrupteur sans fils, d'une télécommande, d'un détecteur de mouvement.

Les inconvénients :

- Il est parfois nécessaire d'ajouter des relais pour agrandir la distance de transmission des différents modules

Les avantages :

- Très facile à installer
- Le prix des différents modules est d'un très bon rapport qualité / prix.
- Nombreux modules permettant de gérer la maison (lumière, prises...)
- Système évolutif permettant de compléter facilement le système domotique [2].



Figure I.7 Logo Chacon

1. Technologie infrarouge est connue pour être utilisée dans les télécommandes de nos téléviseurs, A partir d'une émission lumineuse elle est donc parfois utilisée pour faire communiquer des systèmes domotique.

2. Est une autre technologie de domotique basée sur la technologie CPL développée dans la fin des années 90.

I.8. Les avantages et les inconvénients :

Comme tous les projets la domotique contient des avantages et des inconvénients parmi ces derniers on cite :

I.8.1. Les avantages :

- Le principal avantage de la domotique est l'amélioration du quotidien au sein de la maison, du point de vue du confort, de la sécurité et de la gestion de l'énergie.
- Ce type d'équipement vous simplifie la vie et optimise votre confort en adaptant votre maison à différents scénarios de la vie quotidienne.
- Il vous permet notamment d'éteindre tous vos appareils électriques et de mettre l'alarme quand vous quittez votre domicile, de régler des ambiances lumineuses (ambiance lecture, ambiance relaxation avec lumières tamisées), de vous réveiller dans un habitat chauffé où le café est déjà prêt.
- La domotique permet aussi de réaliser des économies d'énergie grâce à la gestion automatique du chauffage, de la climatisation et de l'éclairage et à la programmation des appareils électroménagers en heures creuses.
- Elle a pour avantage d'améliorer la sécurité grâce à des alarmes, des systèmes d'ouverture automatique de la porte (reconnaissance vocale, carte magnétique...)
- En cas de tentative d'intrusion dans la maison, un appel téléphonique automatique peut contacter le propriétaire ou une entreprise de sécurité.
- Enfin, ces différentes technologies constituent une aide précieuse pour les personnes dépendantes et handicapées [2].

I.8.2. Les inconvénients :

- Le principal est le prix d'achat et d'installation pour certaines technologies, pourraient rebuter plus d'un qui est beaucoup plus élevé. Mais à long terme les factures d'énergie baisseront.
- Le deuxième inconvénient est le verrouillage qu'offrent certaines marques dans leurs produits ne permettant pas d'avoir un logiciel ouvert.
- Des intrusions via le réseau internet par des personnes tierces par exemple les Cracker(Pirate informatique), pourraient espionner la vie privée ou dérégler l'installation.
- Toujours dans le même contexte avoir les commandes et un accès illimité aux maisons d'autrui, faciliterais un éventuelle actes criminels (cambriolages, autres actes encore plus violent) [2].

I.9. Système automatisé :

I.9.1. Définition

Après avoir été mis en fonctionnement par l'homme, un système automatisé est un ensemble de composants qui fonctionne seul (sans interventions supplémentaires) pour effectuer des actions programmées (prévues à l'avance) et répétitives (toujours les mêmes prévues dans le programme) [1].

I.9.2. Les composants d'un système automatisé

Un système automatisé est constitué d'un opérateur (l'utilisateur), d'une partie commande (ordinateur) et d'une partie opérative avec des capteurs et actionneurs.

L'opérateur est l'utilisateur du système qui donne des consignes (ce qu'il veut obtenir) à la partie commande et reçoit des signaux l'informant sur les états du système (température, position ...).

La partie commande joue le rôle du "cerveau" du système qui pilote (ordres) les actionneurs de la partie opérative, reçoit des informations venant des capteurs (comptes rendus) et de l'opérateur (consignes) et informe ce même opérateur (signaux).

Ex : programmateur de la machine à laver, circuit électronique (automate) de l'ascenseur ...

La partie opérative exécute les ordres qu'elle reçoit de la partie commande grâce aux actionneurs et recueille des informations sur les états du système grâce aux capteurs.

•Les actionneurs sont des appareils qui convertissent l'énergie qui leur est fournie en un travail utile, une action (mouvement, chaleur, lumière, son ...) [1].



Moteur
Produire un
mouvement



Ampoule
Produire de la
lumière



Sirène
Émettre un
son



Radiateur
Produire de la
chaleur

•Les capteurs : Ce sont des appareils qui traduisent une information physique en un signal compréhensible (compatible) par la partie commande (Ordinateur, automate, circuit électronique). [1]



Détecteur infrarouge
Détecter une présence



Interrupteur, Bouton poussoir
Détecter une instruction



Sonde de température
Mesurer une température



Capteur de luminosité
Mesurer la luminosité



Capteur fin de course Détecter une position

I.9.3. Représentation simplifiée d'un système automatisé Exemple : éclairage automatique.

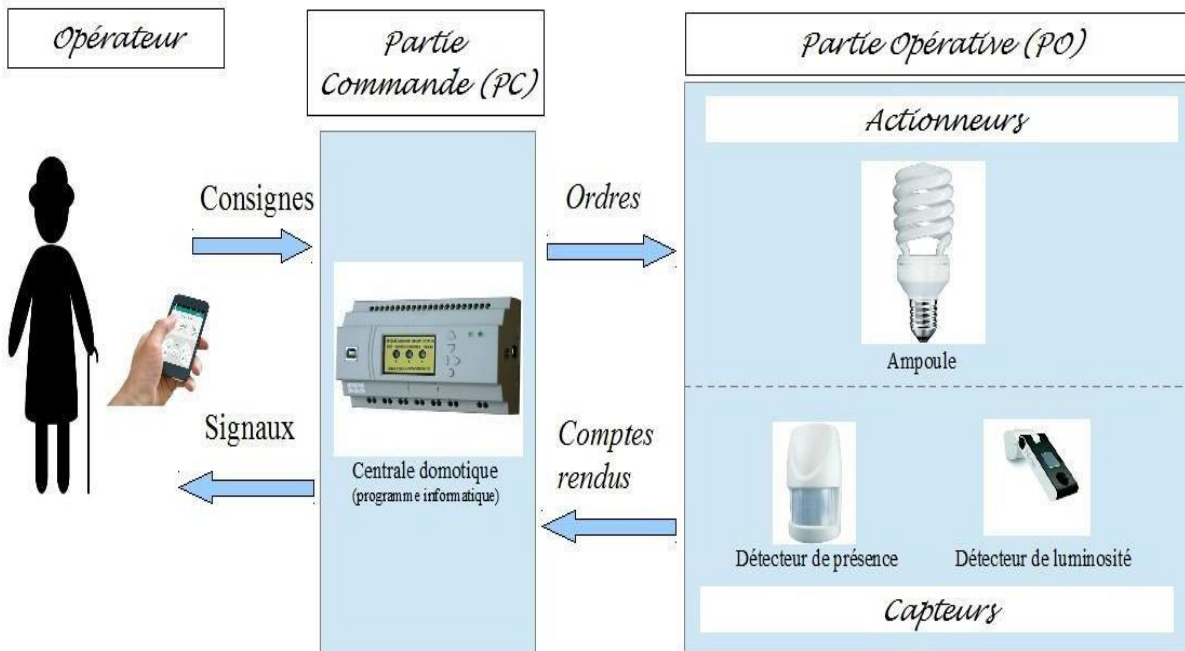


Figure I.8 : Exemple d'un système automatisé

1. L'opérateur donne des consignes à la centrale domotique (mise en marche/arrêt du fonctionnement automatique, allumage forcé de l'ampoule, extinction forcé de l'ampoule ...).
2. La centrale domotique exécute le programme informatique qu'elle contient en fonction des consignes et des comptes rendus qu'elle reçoit et donne en conséquence des ordres aux actionneurs (ordres Allumer l'ampoule ou Éteindre l'ampoule).
3. Les capteurs détectent les grandeurs physiques "présence d'une personne" et "luminosité dans la pièce" et renvois des comptes rendus à la centrale (Présence d'une personne, Absence d'une personne, Forte luminosité, Faible luminosité ...)
4. La centrale domotique informe l'opérateur par des signaux (voyants, SMS, appel téléphonique avec message enregistré, animation sur l'écran ...) sur l'état du système

(Ampoule allumée, éteinte, présence/absence d'une personne dans la pièce, forte/faible luminosité dans la pièce ...) [1].

I.10. Le GSM :

I.10.1. Historique

Des étapes majeures ont marqué l'histoire du GSM et de la radiotéléphonie :

- 1887. Le physicien allemand Heinrich Hertz (1857-1894) découvre les «ondes hertziennes», ce sont les ondes radio.
- 1982. La Conférence Européenne des Postes et Télécommunication (C.E.P.T) réserve des fréquences dans la gamme des 900MHz pour les télécommunications mobiles et crée le Groupe Spécial Mobile (G.S.M).
- 1987. 13 opérateurs européens signent le « Memorandum Of Understanding » (MOU) qui prévoit la mise en service de réseaux mobiles conformes au GSM.
- 1989. Le G.S.M préconise l'introduction de la bande de fréquence 1800MHz afin de permettre l'utilisation de la norme DCS 1800.
- 1991. Le ministère français des PTT autorise France Telecom et la Société Française de Radiophonie (SFR) à exploiter un réseau du type GSM.
- 1992. 17 pays et 26 opérateurs européens ont adopté le système G.S.M. De plus, une première ouverture vers un marché mondial s'effectue par la ratification de l'Australie au MOU.
- 1993. L'abréviation GSM signifie désormais « Global System for Mobile communication ». Ce changement marque ainsi le tournant entre la phase d'élaboration de la norme et sa phase de commercialisation.
- Octobre 1993. Le GSM de deuxième génération apparaît. Cette évolution propose de nouveaux services tels que les renvois d'appel ou la limitation d'appel.
- 1994. Les Etats Unis réservent la bande des 1900MHz à la norme GSM. Une adaptation de la norme leur permettra d'utiliser le réseau à cette bande de fréquence.
- 1995. Démarrage de la phase 2+ du GSM.
- 1997. Le DCS1800 est renommé GSM1800 pour refléter l'apparition de services à vitesse plus élevée. Les réseaux bi bandes se multiplient.
- 2005. Le réseau GSM totalise plus de 1,2 Milliard d'abonnés dans le monde [1].

I.10.2. Présentation

Global System for Mobile Communications (GSM) historiquement « Groupe spécial mobile» est une norme numérique de seconde génération pour la téléphonie mobile. Le groupe de travail chargé de la définir a été établi en 1982 par la Conférence européenne des

administrations des postes et télécommunications (CEPT).

Elle a été spécifiée et mise au point par l'ETSI (Européen Télécommunications Standard Institut) pour la gamme de fréquences des 900 MHz. Une variante appelée Digital Communication System (DCS) utilise la gamme des 1 800 MHz. Cette norme est particulièrement utilisée en Europe, en Afrique, au Moyen-Orient et en Asie. Deux autres variantes, en 850 MHz et en 1 900 MHz PCS (personale communications services), sont également utilisées. La protection des données est assurée par les algorithmes de chiffrement A5/1 et A5/2.

Tel qu'il a été conçu, le réseau GSM est idéal pour les communications de type

« voix » (téléphonie). Le réseau étant commuté, les ressources ne sont allouées que pour la durée de la conversation, comme lors de l'utilisation de lignes téléphoniques fixes. Les clients peuvent soit acheter une carte prépayée, soit souscrire un abonnement.

Sous l'égide de l'organisation 3GPP la norme GSM a ensuite été étendue pour prendre en charge de plus hauts débits et le transport de données en mode « paquet » par les extensions GPRS (General Packet Radio Services) puis EDGE (Enhanced Data rates for GSM Evolution). Ces deux modes peuvent cohabiter avec le mode « voix commutée » du GSM et utilisent les mêmes antennes et les mêmes bandes de fréquence [1].

I.10.3. Architecture de GSM

L'architecture d'un système GSM se décompose en trois sous-systèmes qui sont bien schématisés dans la figure suivante :

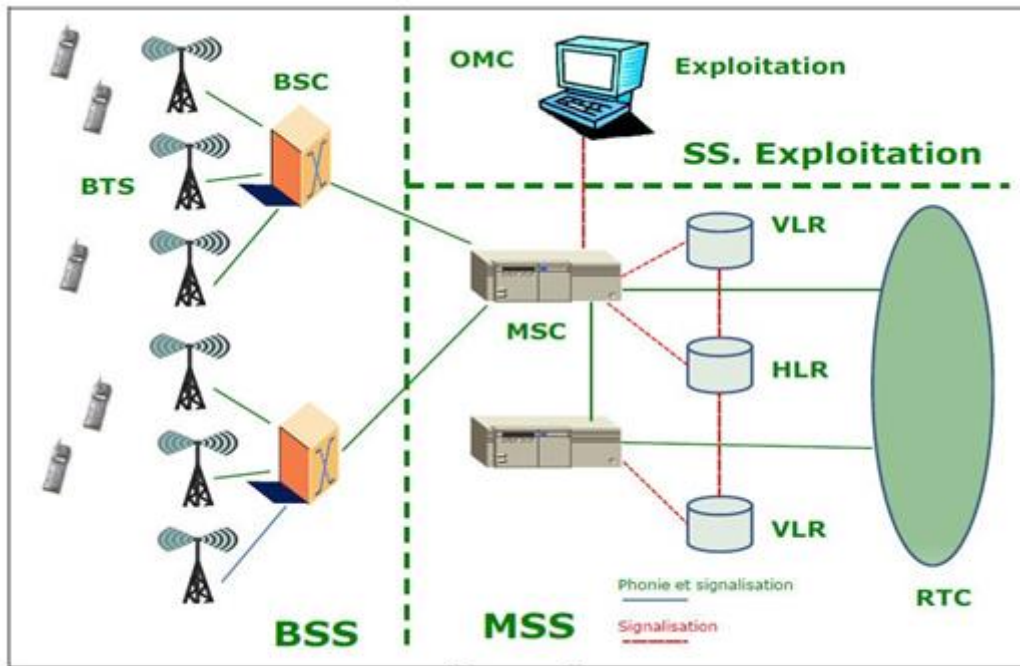


Figure I.9: Architecture d'un réseau GSM[1].

I.10.3.1. Le sous-système radio (BSS)

Il gère la partie radio des communications et se compose d'émetteurs-récepteurs radio (BTS) contrôlés par une BSC.

La radio BTS (Base Transceiver Station) est un ensemble d'émetteurs-récepteurs radio (TRX), sans grande intelligence. Elle gère la couche physique de l'interface air.

Le BSC il contrôle les stations de base et assure la commutation entre les ressources terrestres et radio [1].

I.10.3.2. Le sous-système réseau (NSS)

Il gère le traitement des appels, la mobilité et l'acheminement de/vers les réseaux filaires. Il se compose de commutateurs radio (MSC) et d'un certain nombre de bases de données HLR et VLR.

Le MSC est un commutateur radio qui gère l'ensemble des communications dans le réseau GSM.

Le HLR (Home Location Register) est une base de données qui centralise les caractéristiques et les informations de localisation des usagers du réseau GSM.

Le VLR (Visitor Location Register) : cette entité contient les données de travail relatives aux abonnés présents dans la zone du MSC. Il permet de minimiser l'accès au HLR.

Le MSC (Mobile-services Switching Centre) est un commutateur qui gère l'établissement des communications entre un mobile et un autre MSC, Il dialogue avec le VLR pour gérer la mobilité des usagers [1].

I.10.3.3. Le sous-système exploitation OSS (Opération Sub-System)

Il contrôle les droits d'accès au réseau, les droits des usagers et assure l'interface homme-machine d'exploitation. Il gère aussi le maintien en conditions opérationnelles du réseau et la remontée des alarmes

La diversité des équipements présents dans un réseau GSM tant sur le plan de leur type (émetteurs-récepteurs, commutateurs, bases de données) que la multiplicité des fournisseurs pousse à adopter une approche structurée et hiérarchique. La norme GSM présente deux niveaux :

Les OMC (Opération and Maintenance Center) et les NMC (Network Management Center).

Le NMC permet l'administration générale de l'ensemble du réseau par un contrôle centralisé, alors que les OMC permettent une supervision locale des équipements [1].

I.11. Principe de fonctionnement de la domotique par GSM

La domotique par GSM est basée sur deux points essentiels, la commande et la surveillance par GSM

I.11.1. Commande par GSM

La commande par GSM peut connecter /déconnecter à distance, tous les matériels électrique comme moteurs, pompes, portes, portails, chauffages, climatisation, etc...

Ainsi tous les systèmes utilisant le GSM se connectent au réseau de téléphonie et comme chaque téléphone portable dispose de sa propre carte SIM et son numéro, et c'est pour ça la commande par GSM est protégée et seuls les numéros que vous autorisez seront habilités à commander votre maison.

I.11.2. Surveillance par GSM

La surveillance par GSM vous permet de contrôler tous les paramètres de votre maison, bureau, entreprise, coupure de courant, variation de température, mouvement ou intrusion, fuite de gaz, etc.... sans vous déplacer, on recevant des messages ou des notifications sur votre téléphone [1].

I.12. Conclusion :

Dans ce chapitre, nous avons parlé des définitions et des concepts expliqués qui nous ont permis de comprendre les performances théoriques de la domotique et du GSM en général. Le matériel utilisé pour la réalisation du projet mérite d'être étudié de façon plus particulière afin d'obtenir une meilleure maîtrise du travail de domotique. C'est l'objectif du chapitre suivant.

Chapitre II :

Matériels et logiciels utilisés

II.3. Historique de l'Arduino :

L'Arduino est à l'origine un projet d'étudiants de l'école de Design d'Interaction d'Ivrea en Italie. Au début des années 2000, les outils de conception de projets dans le domaine du design d'interaction étaient onéreux, proches d'une centaine d'euros. Ces outils étaient pour la plupart conçus pour le domaine de l'ingénierie et de la robotique. Maîtriser et utiliser ces composants demandait beaucoup de temps et d'apprentissage et ralentissait fortement le processus de création pour ces jeunes étudiants .

Il leur vient alors à l'idée de créer une plateforme plus abordable et plus simple à utiliser, reposant sur l'environnement de développement Processing mis au point en 2001 par des étudiants du MIT. C'est donc en 2003 que, pour un projet de fin d'études, fut conçue la carte Wiring, ancêtre de l'Arduino. Visant à rendre la plateforme toujours moins chère et plus accessible, une équipe d'étudiants et de professeurs finirent par concevoir la toute première Arduino en 2005. Entièrement open source, l'Arduino présentait l'avantage d'être multiplateforme et d'être en perpétuelle optimisation par la communauté d'utilisateurs[11].



Figure II.2: Les créateurs de Arduino

II.4. Définition de Arduino :

Le système Arduino est une plateforme open source basée sur une carte programmée à microcontrôleur et un environnement de développement permettant d'écrire, compiler et tester un programme. Les cartes et modules Arduino proposent une interface d'entrées-sorties simples qui peuvent recevoir des signaux de capteurs ou interrupteurs et peuvent commander des moteurs, éclairages, etc. Les projets Arduino peuvent être autonomes ou communiquer avec un PC. Le logiciel de développement open-source peut être téléchargé gratuitement sur

www.arduino.org et qui permet de programmer les cartes Arduino en C++ [8].

II.5. Propriétés de système Arduino :

Open source : La désignation open source, ou « code source ouvert », s'applique aux logiciels (et s'étend maintenant aux œuvres de l'esprit) dont la licence respecte des critères précisément établis par l'Open Source Initiative, c'est-à-dire les possibilités de libre redistribution, d'accès au code source et de création de travaux dérivés. Mis à la disposition du grand public, ce code source est généralement le résultat d'une collaboration entre programmeurs [8].

Open hardware : L'Open Hardware est un terme qui désigne les matériels dont la conception peut être reproduite, car toutes les informations nécessaires à leur construction sont en libre accès. Une ouverture qui est finalement l'équivalent de l'Open Source, mais non plus appliquées aux lignes de code d'un logiciel, mais aux plans de conception des objets [8].

II.6. Constitution de la carte Arduino UNO :

Un module Arduino est généralement construit autour d'un microcontrôleur ATMEL AVR, et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Le microcontrôleur est préprogrammé avec un bootloader de façon à ce qu'un programmeur dédié ne soit pas nécessaire [9].

II.6.1. Partie matérielle

Généralement tout module électronique qui possède une interface de programmation est basé toujours dans sa construction sur un circuit programmable ou plus [9].

II.6.1.1. Microcontrôleur ATmega328

Un microcontrôleur ATmega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit au temps des pionniers de l'électronique. Aujourd'hui, en soudant un grand nombre de composants encombrants, tels que les transistors, les résistances et les condensateurs tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C [9].



Figure II.3ATMega328

- **Le microcontrôleur ATmega328** est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement sur :
- **Mémoire Flash:** C'est celle qui contiendra le programme à exécuter. Cette mémoire est effaçable et réinscriptible mémoire programme de 32Ko (dont boot loader de 0.5 ko) [9].

II.6.1.2. Les sources de l'alimentation de la carte :

On peut distinguer deux genres de sources d'alimentation (Entrée Sortie) et cela comme suit :

- **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.
- **La tension régulée 5V :** 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" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino).

Le tension régulée 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN 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.

- **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 nécessitant cette tension au lieu du 5V. L'intensité maximale disponible sur cette broche est de 50mA [9].

II.6.1.3) Les entrées et sorties :

Cette carte possède 14 broches numériques (numérotée de 0 à 13) 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 "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction **digitalWrite** (broche, HIGH).

En plus, certaines broches ont des fonctions spécialisées :

- **Interruptions Externes:** Broches 2 et 3. 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. Impulsion PWM (largeur d'impulsion modulée) : Broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction **analogWrite()**.
- **SPI (Interface Série Périphérique):** Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) [9].
- **Les ports de communications :**

La carte Arduino UNO a de nombreuses possibilités de communications avec l'extérieur. L'Atmega328 possède une communication série UART TTL (5V), grâce aux broches numériques 0 (RX) et 1 (TX).

On utilise (RX) pour recevoir et (TX) transmettre (les données séries de niveau TTL). Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega328 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.

Comme un port de communication virtuel pour le logiciel sur l'ordinateur, La connexion série de l'Arduino est très pratique pour communiquer avec un PC, mais son inconvénient est le câble USB, pour éviter cela, il existe différentes méthodes pour utiliser ce dernier sans fil [9].

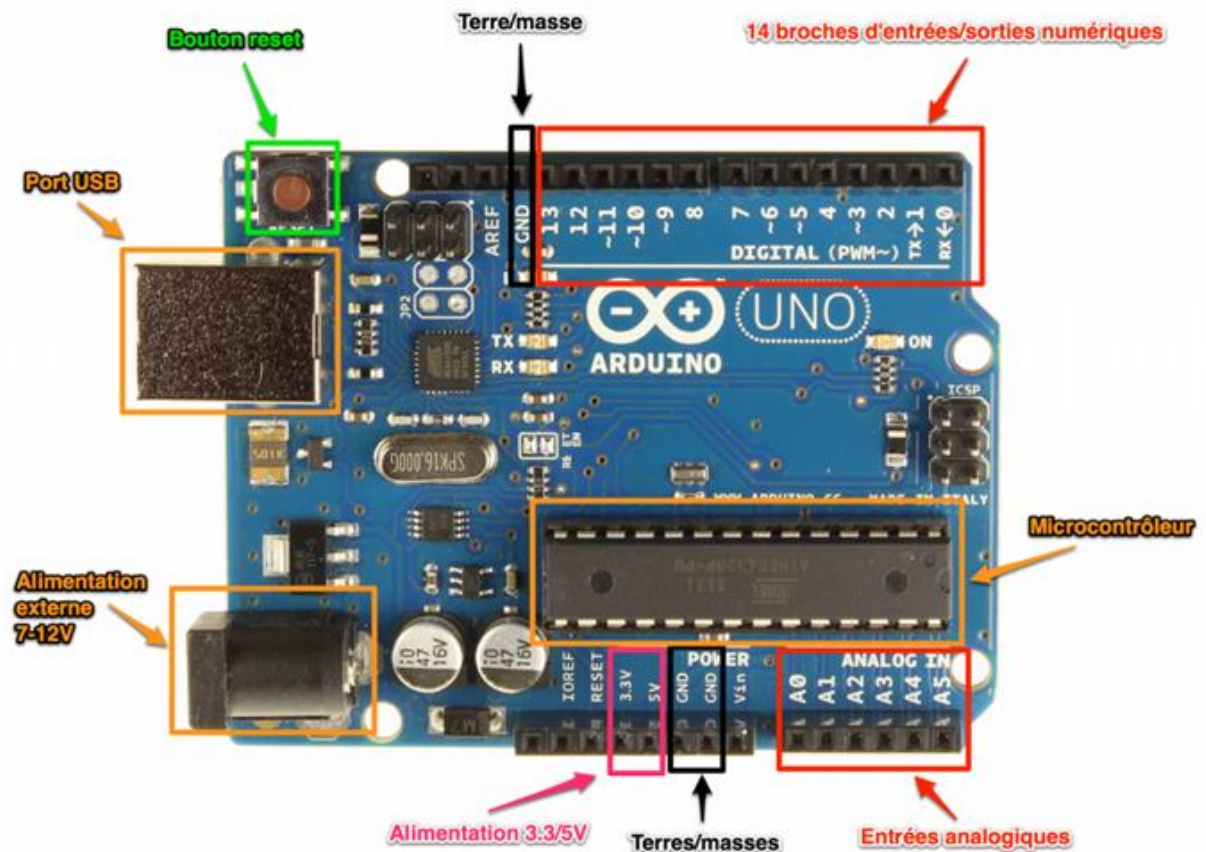


Figure II.4: Constitution de la carte Arduino UNO

II.6.1.4 .les accessoires de Arduino :

II.6.1.4.1. Plaque d'essai

Une plaque d'essai (voir la figure.II.6) permet de réaliser des montages électroniques sans soude. En règle générale les plaques d'essais sont de forme rectangle. Il y a plusieurs rangés de trous, certaines rangés sont verticale tandis que d'autres sont horizontal. Elle s'utilise avec des fils de cuivre isolés, de longueur et couleur variables (voir la figure.II.5) [16].



Figure II.5 : Fils

Plusieurs modèles existent, nous utiliserons des plaques d'essai comme celle représentée ci-dessous. La plaque d'essai comporte des connexions cachées, chaque bande de cuivre met en contact 5 trous. Les trous sont espacés exactement de 2,54 mm [16].



Figure II.6 : Plaque d'essai

II.6.1.4.2. Câble USB

Ce câble USB permet à la fois l'alimentation de la carte Arduino et le chargement des programmes nécessaires au fonctionnement des cartes programmables à travers IDE-Arduino. La longueur du câble est d'environ 1 m [16].



Figure II.7 : câble USB

II.6.1.4.3. Les capteurs

- Capteur de flamme KY-026:



Figure II.8: capteur de flamme KY-026

Le module capteur de flamme KY-026 permet la détection de flamme à l'aide d'un récepteur infrarouge qui va capter les émissions lumineuses de sources de chaleur.

Principe de fonctionnement :

Toutes sources de chaleur (au-delà de 0K) émet une lumière infrarouge. Le module de détection de flamme KY-026 pour Arduino mesure l'intensité de la lumière infrarouge émise par le feu sur une plage de longueur d'onde comprise entre 760 à 1100 nm. Le module dispose de sorties numériques et analogiques et d'un potentiomètre pour régler la sensibilité [15].

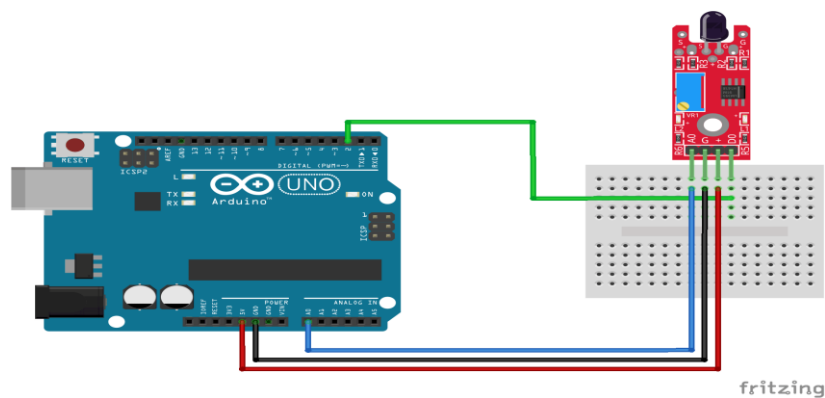


Figure II.9: Branchement de flamme KY-026 avec Arduino

- **capteur ultrason (HC-SR04)**



Figure II.10: Capteur Ultrason HC-SR04

Description

Le HC-SR04 est un capteur qui utilise les ultrasons pour déterminer la distance d'un objet de 2cm à 400cm. La mesure s'effectue "sans contact" et dispose d'une précision de 3mm dans les conditions optimales. Le module inclus un émetteur et un récepteur ultrason ainsi qu'un système de contrôle [1].

Le principe de fonctionnement :

- Envoyer un signal numérique à l'état haut sur l'émetteur pendant 10 μ s.
- Le capteur envoie automatiquement 8 impulsions d'ultrasons à 40 kHz et détecte les signaux qui reviennent.
- Si le signal revient, la durée de l'état haut du signal reçue correspond au temps entre l'émission des ultrasons et leur réception [1].

Calcul de la distance :

Distance = (temps à l'état haut signal reçu * vitesse du son)/2 (vitesse du son dans l'air : 340 m/s).

Connecteur du capteur :

- Alimentation 5V .
- Entrée émetteur d'impulsion d'ultrasons .
- Sortie récepteur d'impulsion d'ultrasons .
- Masse 0V .

Caractéristique

Tension d'alimentation	5VDC
Courant d'alimentation	15mA
Fréquence de travail	40Hz
Distance maximale de détection	4m
Distance minimale de détection	2cm
Angle de détection	15 degrés
Signal d'entrée de l'émetteur	Impulsion à l'état haut de 10µs
Signal de sortie du récepteur	Signal numérique à l'état haut et la distance proportionnellement
Dimension	45*20*15mm

Tableau II.1:Caractéristique de Ultrason HC-SR04

Le calcul de la distance parcourue par un son s'effectue par l'équation suivante : Distance = vitesse x temps ($D = V \times T$)

V : c'est la vitesse de la lumière (34'000 cm/1'000'000µs)

T : c'est le temps écoulé par les ondes ultrason aller /retour, le HC-SR04 donne une durée d'impulsion en dizaines de µs. donc Il faut multiplier la valeur obtenue par 10 µs pour obtenir le temps T ($T = \text{valeur} \times 10 \mu s$)

Puisque le son fait un aller-retour. La distance vaut donc la moitié $D = 34'000 \text{ cm} / 1'000'000 \mu s * \text{valeur} \times 10 \mu s / 2$

En simplifiant : $D = 17/100 \text{ cm} * \text{valeur} [1]$.

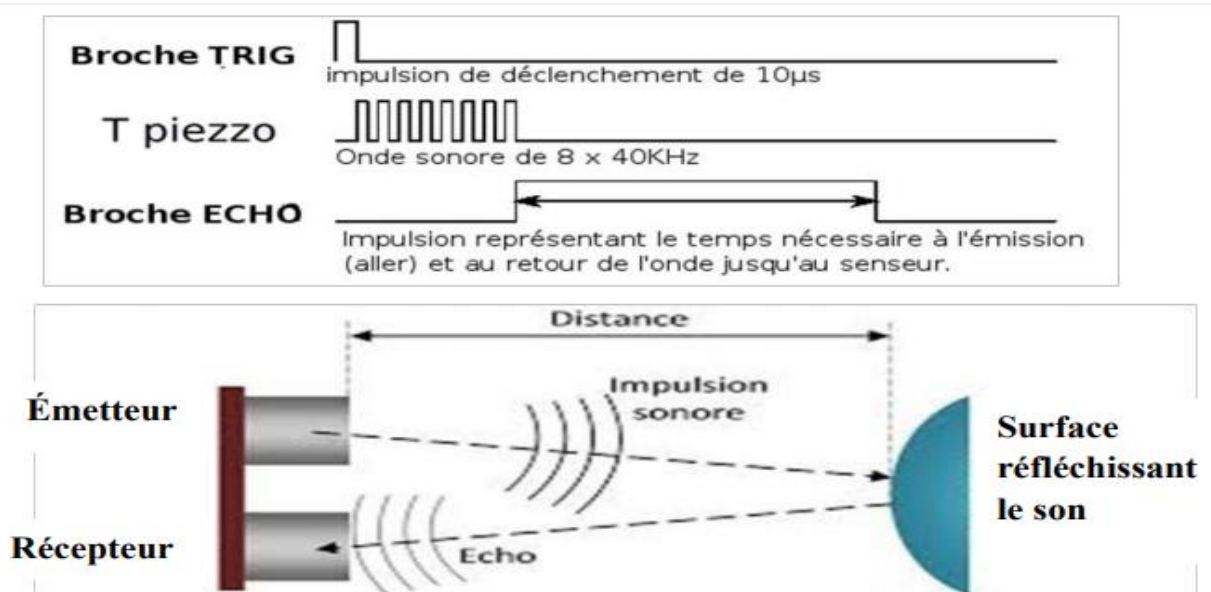


Figure II.11 : Principe de fonctionnement d'un capteur Ultrason HC-SH04

- **Capteur d'humidité et de température DHT11 :**

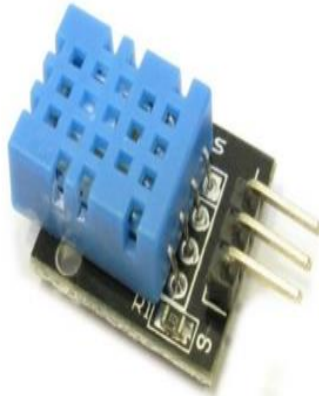


Figure II.12 :capture DHT11

ce capteur est dédié à la mesure de température et l'humidité avec une sortie de signal numérique calibrée en utilisant la technique exclusive « numérique-signal- acquisition » et la technique de détection de température & d'humidité, il assure la fiabilité élevée et une excellente stabilité à long terme, ce capteur comprend une mesure de l'humidité du type résistif et un élément NTC de mesure de température , et se connecte à un microcontrôleur 8 bits haute performance, offrant d'excellente qualité, réponse rapide, capacité anti parasitage et la rentabilité [8].

Caractéristiques détaillées :

- Alimentation : 5V
- Consommation : 0.5 mA en nominal / 2.5 mA maximum
- Etendue de mesure température : 0°C à 50°C \pm 2°C
- Etendue de mesure humidité : 20-90%RH \pm 5%RH

Alimentation et broche :

La tension d'alimentation de DHT11 est 3-5.5V DC, lorsque le capteur est alimentée, n'envoyez pas n'importe quelle instruction du capteur en moins d'une seconde afin de transmettre le statut instable. Un condensateur d'une valeur 100nF peuvent être ajoutés entre VDD et GND pour le filtrage de l'alimentation [8].

Le brochage du capteur DHT11 est le suivant :

VCC : 3.5 à 5.5V.

GND : Masse 0V.

Data : données [8].

- **Capteur de gaz et de fumée (MQ-2) :**



Figure: II.13Capteur de gaz(MQ-2)

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 [10].

Caractéristiques :

- Alimentation : 5 V DC.
- Consommation : moins de 150 mA.
- 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.

Connectique 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 [10].

- **Capteur de mouvement PIR HC-SR501 :**



Figure II.14:PIR HC-SR501

Description:

Ce capteur PIR est idéal pour détecter des mouvements ou une présence dans les applications de domotique ou de robotique. Simple d'utilisation et facilement réglable, vous pourrez le mettre en œuvre très rapidement [10].

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 : 5-200 S (peut être ajustée, par défaut 5S +-3%) .
- la température de travail :-20-+ 80 °C .

Connectique du module du détecteur de présence HC-SR501 :

Le capteur HC-SR501 possède 4 broches qui sont :

- VCC : à connecter au 5V pour l'alimentation.
- OUT : signal de sortie du module
- GND : à mettre à la masse[10].

- **RFID MFRC522 :**



Figure II.15: Module RFID MFRC522

Description :

Ce module est conçu pour communiquer dans la bande 13.56 MHz permettant ainsi d'interagir avec des tags RFID à une distance comprise entre 5cm et 1.5m, à faible coût, de petite taille, sans contact carte à puce pour lire et écrire, instruments intelligents et des dispositifs portatifs développés mieux.

Or la fréquence de fonctionnement de ce module est 13.56 MHz donc on peut le classer comme un système RFID hautes fréquences dont les tags et le lecteur sont couplés magnétiquement.

Le module RFID Mifare RC522 possède un seul lecteur sous forme d'une petite puce électronique et deux tags, le premier sous forme d'une carte et le deuxième sous forme d'une clé.



Figure II.16:Présentation des Tags, label ou transpondeur

Pour transmettre des informations à l'interrogateur (encore appelé station de base ou plus généralement lecteur), une étiquette RFID est généralement munie d'une puce électronique associée à une antenne. Cet ensemble, appelé inlay, est ensuite packagé pour résister aux

conditions dans lesquelles il est amené à vivre. L'ensemble formé est appelé tag, label ou encore transpondeur [10].

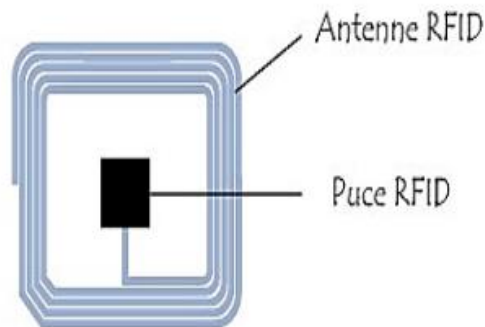


Figure II.17: Présentation de "INLAY"

Composants et fonctionnement d'un système RFID :

Les composants d'un système RFID

- **Système RFID** : Un système RFID (Radio Fréquence Identification) se compose de transpondeurs (aussi nommés étiquettes, marqueurs, tags, identifiants...) et d'un ou plusieurs interrogateurs (aussi nommés lecteurs, bases stations...).
- **Interrogateurs RFID** : Ce sont des dispositifs actifs, émetteurs de radiofréquences qui vont activer les tags qui passent devant eux en leur fournissant l'énergie dont ils ont besoin pour fonctionner.
- **Tag RFID** : C'est un dispositif récepteur, Il est muni d'une puce contenant les informations et d'une antenne pour permettre les échanges d'informations.
- **Interface** : L'interface est le support de transmission de l'énergie et des données. Dans le cadre des systèmes RFID, il s'agit de l'air [10].

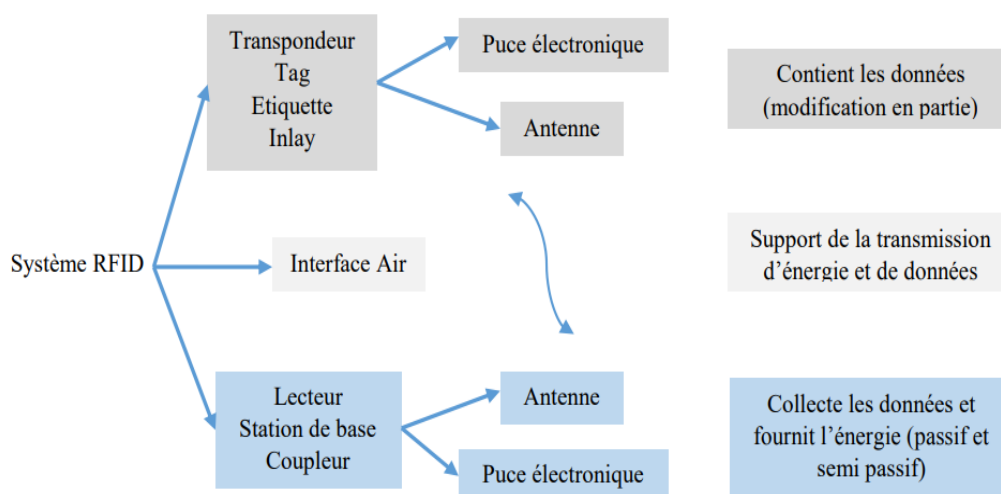


Figure II.18: Les composants d'un système RFID

- Le couplage tag RFID / lecteur RFID :

La liaison entre tag et lecteur se réalise par :

- Couplage magnétique dans le cas d'un champ proche (quelques cm à 1,5 m). L'interrogateur utilise alors des LF (Basses Fréquences) ou des HF (Hautes Fréquences). Les antennes sont alors constituées de boucles inductives.
- Couplage électrique dans le cas d'un champ lointain (jusqu'à 6m). L'interrogateur utilise alors des UHF (Ultra Hautes Fréquences) ou des SHF (Super Hautes Fréquences). Les antennes de base sont alors des dipôles ou des patches [10].

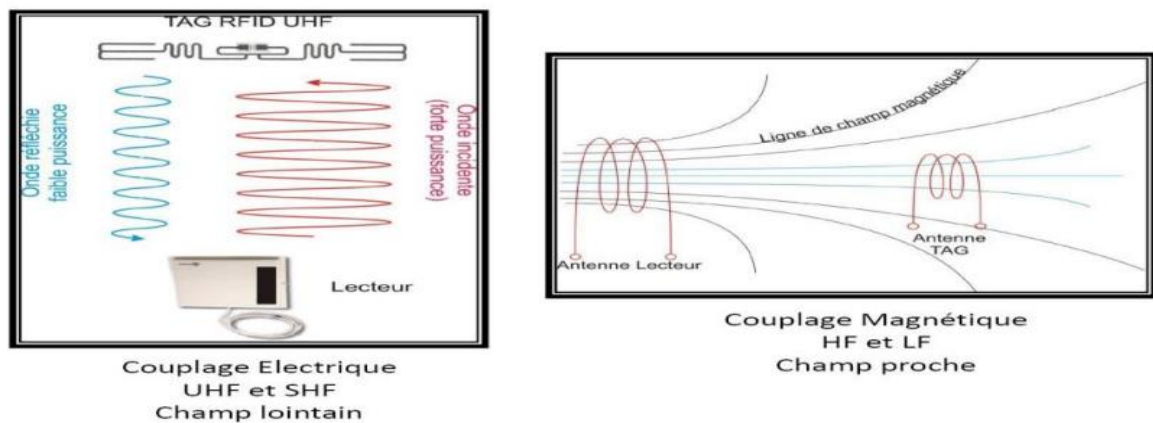


Figure II.19: le couplage magnétique et électrique entre le tag et lecteur RFID

- Les gammes de fréquences RFID :

La RFID a 4 types des gammes dans le spectre radio qui sont :

- LF : 125 kHz - 134,2 kHz : basses fréquences,
- HF : 13,56 MHz : hautes fréquences,
- UHF : 860 MHz - 960 MHz : ultra hautes fréquences,
- SHF : 2,45 GHz : super hautes fréquences [10].

Connectique du module RFID-MFRC522 :

Le module RFID-RC522 13.56Mhz compatible Arduino, il dispose 8 broches dont 7 uniquement sont utilisées couramment. Dans le tableau suivant [10]:

	PCD	Arduino				
	MFRC522	UNO/101	Mega	Nano V3	Leonardo / Micro	Pro Micro
signal	pin	pin	pin	pin	pin	pin
RST/Reset	RST	9	5	D9	RESET/ICSP-5	RST
SPI SS	SDA	10	53	D10	10	10
SPI MOSI	MOSI	11/ ICSP-4	51	D11	ICSP-4	16
SPI MISO	MISO	12/ ICSP-1	50	D12	ICSP-1	14
SPI SCK	SCK	13/ ICSP-3	52	D13	ICSP-3	15

Tableau II.2:Connectique du module RFID-MFRC522

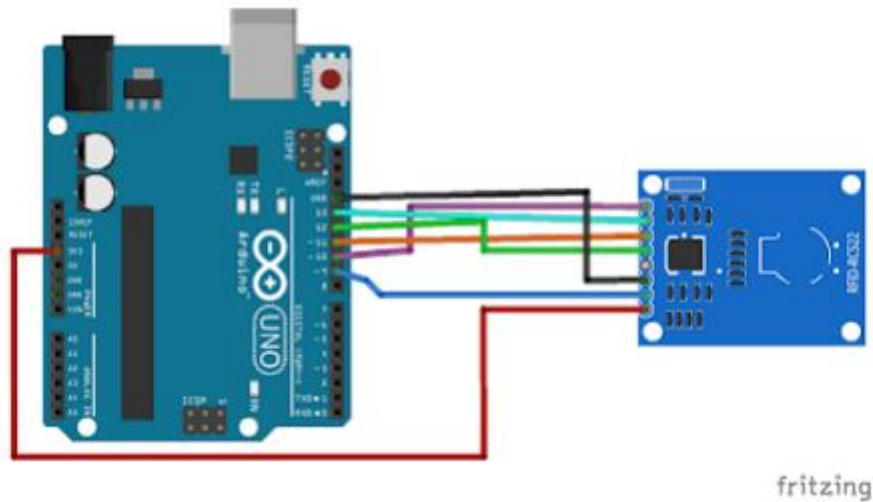


Figure II.20: branchement de RFID MFRC522 avec la carte Arduino UNO

II.6.1.4.4. Les actionnaires :

- **LED :**

Une LED est un composant électronique et optique, qui en étant traversé par du courant électrique, émet une lumière d'une intensité. Les LED consomment peu d'électricité. Ce composant de la famille des semi-conducteurs est dit "passif". Son sens est déterminé grâce aux 2 parties qui composent ce composant : l'Anode et la Cathode. Pour s'illuminer, le courant doit parcourir la LED de l'anode vers la cathode. dans ce projet le LED représente le feu vert, qui indique que tout va bien ,Le voyant rouge indique que quelque chose ne va pas [21].

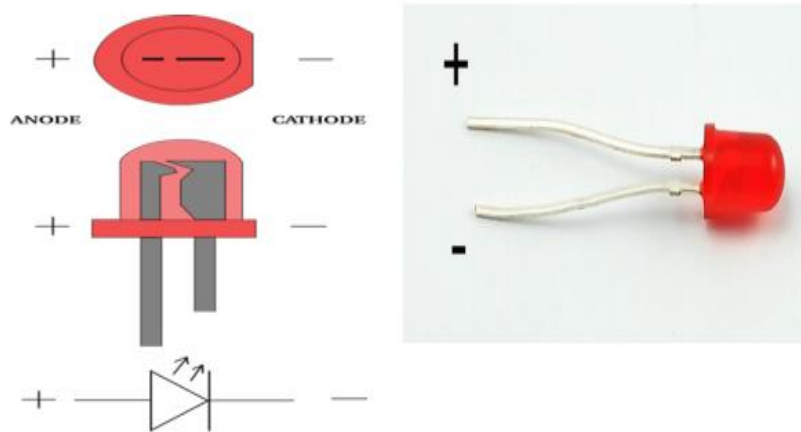


Figure II.21: Schéma et sens du courant des LED .

- **Buzzer :**

Le buzzer est un composant constitué essentiellement d'une lamelle réagissant à l'effet piézoélectrique. La piézoélectricité est la propriété que possèdent certains minéraux de se déformer lorsqu'ils sont soumis à un champ électrique. Ce phénomène est réversible ; si nous déformons ce minéral, il produit de l'énergie électrique. Dans l'univers Arduino, le buzzer est principalement utilisé pour émettre un son .Dans notre projet, le buzzer va représenter l'alarme[21].

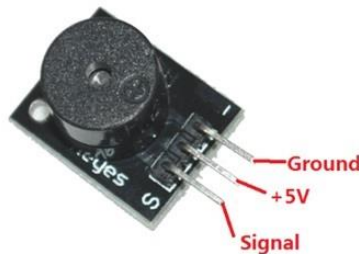


Figure II.22 : buzzer

- **servomoteur :**



Figure II.23:servomoteur

Description

Un servomoteur est un actionneur (moteur) capable de maintenir une opposition à un effort statique et dont la position est vérifiée en continu et corrigée en fonction de la mesure. Le servomoteur est composé de plusieurs éléments visibles :

- Les fils.
- L'axe de rotation sur lequel est monté un accessoire en plastique ou en métal.
- Le boîtier qui le protège.

Aussi de plusieurs éléments que l'on ne voit pas qui se trouvent à l'intérieur du boîtier.

- Un moteur à courant continu.
- Des engrenages pour former un réducteur.
- Un capteur de position de l'angle d'orientation de l'axe (un potentiomètre bien souvent).
- Une carte électronique pour le contrôle de la position de l'axe et le pilotage du moteur à courant continu.

La position est définie avec une limite de débattement d'angle de 180 degrés. Souvent abrégé en « servo » provenant du latin servus qui signifie « esclave ».

Il existe plusieurs servomoteurs, dans notre application on a utilisé un servomoteur de faible puissance (9 Gramme) SG90.

Il contient trois fils, un pour l'alimentation, l'autre pour la masse (GND) et la dernière pour le signal d'entrée (la commande) [13].

Principe de fonctionnement d'un servomoteur :

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 [1].

Connecteur du servomoteur

Un servomoteur se pilote par l'intermédiaire d'un câble à 3 fils. Ce câble permet à la fois de l'alimenter et de lui transmettre des consignes de position par le fil de signal :

- Le noir ou marron : La masse .
- Le rouge : La tension d'alimentation continue (+) .
- Le jaune, orange ou blanc : Le signal de commande PWM [1].

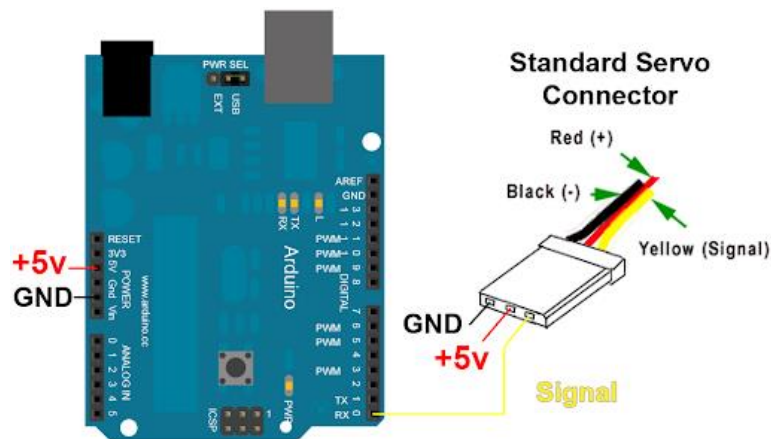


Figure II.24 :Branchement de servomoteur avec Arduino

- Afficheur LCD 16x2 I2C :

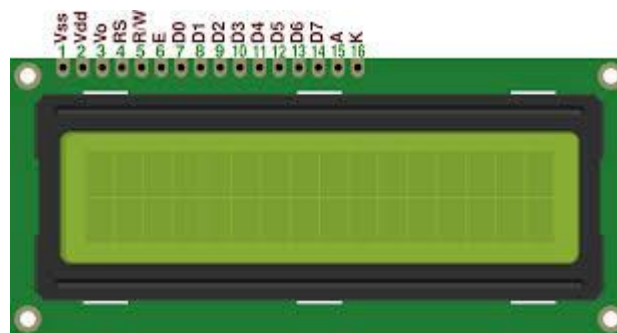


Figure II.25: LCD 16x2

LCD :

Un écran LCD (Liquid Crystal Display) ou affichage à cristaux liquides en français est un dispositif qui permet d'afficher des caractères tout en consommant assez peu d'électricité,

c'est pourquoi on le retrouve dans de nombreux projets électroniques et sur tous types d'appareils [12].

I2C :

Un I2C (Inter-Integrated Circuit) est un bus informatique. Conçu par Philips pour les applications de domotique et d'électronique domestique, il permet de relier facilement un microprocesseur et différents circuits tout en réduisant le nombre de lignes nécessaires à seulement deux lignes :

- SDA (Serial Data) .
- SCL (Serial CLock) [12].



FigureII.26 : I2C

Caractéristiques :

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 [12].

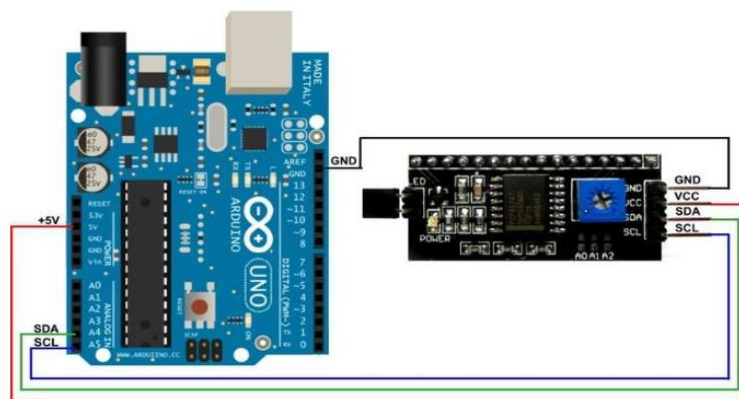


Figure II.27: Branchement de I2C avec la carte Arduino .

II.6.1.4.5. Module de communication :

- **Le shieldSIM900a GSM/GPRS :**

La carte GSM900a est utilisé dans des applications de contrôle à distance, de domotique, et

aussi de sécurité.

Présentation du carte SIM900a GSM/GPRS :

Le shield SIM900 GSM/GPRS est conçu pour entourer la puce SIM900 avec tout le nécessaire pour s'interfacer avec Arduino, ainsi que quelques goodies supplémentaires pour profiter des fonctionnalités uniques de la puce.

Familiarisons-nous avec ces caractéristiques et capacités du bouclier. Voici un aperçu rapide :

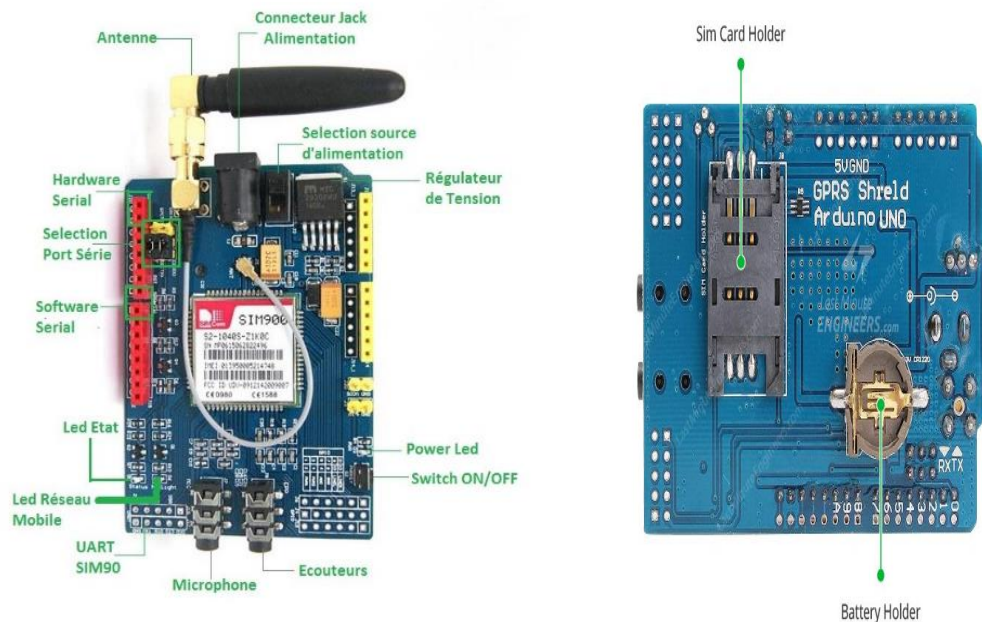


Figure II.28 : Présentation du matériel du blindage SIM900a GSM/GPRS

Le bouclier SIM900 contient une quantité surprenante de fonctionnalités dans son petit cadre. Certains d'entre eux sont énumérés ci-dessous:

- Prend en charge la quadri bande : GSM850, EGSM900, DCS1800 et PCS1900
- Connectez-vous à n'importe quel réseau GSM mondial avec n'importe quelle carte SIM 2G
- Passez et recevez des appels vocaux à l'aide d'un écouteur externe et d'un microphone à électret
- Envoyer et recevoir des SMS
- Envoyer et recevoir des données GPRS (TCP/IP, HTTP, etc.)
- Scannez et recevez des émissions de radio FM
- Puissance de transmission:
 - Classe 4 (2W) pour GSM850
 - Classe 1 (1W) pour DCS1800

- Ensemble de commandes AT basé sur la série
- Connecteurs U.FL et SMA pour antenne cellulaire
- Accepte la carte SIM pleine taille .

Indicateurs d'état LED

Il y a trois LED sur le blindage SIM900 GSM/GPRS qui indiquent la connectivité ou l'état de l'alimentation. En observant ces LED, vous pouvez obtenir un retour visuel sur ce qui se passe avec le bouclier.

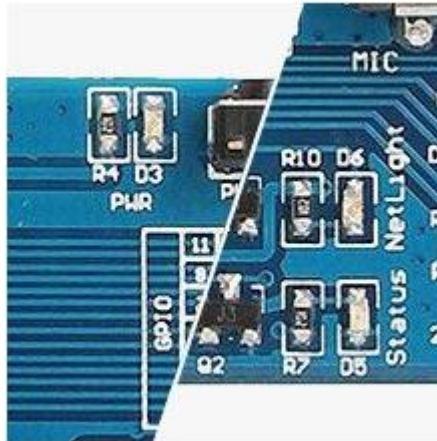


Figure II.29: Les trois LED de SIM900a

PWR : Cette LED est connectée à la ligne d'alimentation du shield. Si cette LED est allumée, le blindage est alimenté.

Status : Cette LED indique l'état de fonctionnement du SIM900. Si cette LED est allumée, la puce est en mode de fonctionnement.

Netlight : Cette LED indique l'état de votre réseau cellulaire. Il clignotera à différentes vitesses pour indiquer dans quel état il se trouve.

éteint : la puce SIM900 ne fonctionne pas .

64 ms activé, 800 ms désactivé : la puce SIM900 est en cours d'exécution mais n'est pas encore enregistrée sur le réseau cellulaire.

64 ms allumé, 3 secondes éteint : la puce SIM900 est enregistrée sur le réseau cellulaire et peut envoyer/recevoir de la voix et des SMS.

64 ms activé, 300 ms désactivé : la connexion de données GPRS que vous avez demandée est active.

Alimentation pour SIM900 Shield

L'une des parties les plus importantes pour faire fonctionner le bouclier SIM900 est de lui

fournir suffisamment de puissance.

Selon l'état dans lequel il se trouve, le SIM900 peut être un appareil relativement gourmand en énergie. La consommation de courant maximale de la puce est d'environ 2A pendant la rafale de transmission. Il ne tirera généralement pas autant, mais peut nécessiter environ 216 mA pendant les appels téléphoniques ou 80 mA pendant les transmissions réseau. Ce graphique de la fiche technique résume ce à quoi vous pouvez vous attendre :

Modes	La fréquence	Consommation de courant
Eteindre		60 μ A
Mode veille		1 mA
Etre prête		18 mA
Apelle	GSM850	199 mA
	EGSM900	216 mA
	DCS1800	146 mA
	PCS1900	131 mA
GPRS		453mA
Rafale transmission		2A

Tableau II.3: Caractéristique de alimentation SIM900 Shield

La tension de fonctionnement de la puce SIM900 est de 3,4 V à 4,4 V. Pour maintenir la tension d'alimentation sûre à 4,1 V, le blindage est livré avec un régulateur de tension à courant élevé, haute précision et faible chute de tension MIC29302WU de Micrel- capable de gérer des courants de charge jusqu'à 3 A.

Vous pouvez ajouter une alimentation externe au blindage avec la prise CC de 5,5 mm, à laquelle vous pouvez connecter n'importe quel adaptateur mural 5V-9V CC dont vous disposez. À côté de la prise CC, se trouve un interrupteur à glissière pour sélectionner la source d'alimentation étiquetée EXTERN. Pour utiliser une source d'alimentation externe, déplacez le curseur comme indiqué ci-dessus.

L'alimentation doit pouvoir fournir au moins 2 A de courant de surtension, sinon la puce continuera à s'éteindre.

Communication UART

Le shield SIM900 GSM/GPRS utilise le protocole UART pour communiquer avec un Arduino. La puce prend en charge le débit en bauds de 1200bps à 115200bps avec détection automatique en bauds .

À l'aide de cavaliers, vous pouvez connecter (RX, TX) du blindage à la série logicielle (D8, D7) ou à la série matérielle (D1, D0) de l'Arduino.



Figure II.30:le protocole UART pour communiquer avec un Arduino

Haut-parleur et microphone

Le blindage est livré avec deux prises jack 3,5 mm standard. Un pour les écouteurs stéréo et l'autre pour le microphone mono. Il vous permet d'utiliser l'interface audio du SIM900 pour passer et recevoir des appels vocaux et écouter la radio FM.

Mic : Vous pouvez connecter un microphone à électret externe à cette prise.

Écouteur : Vous pouvez connecter des écouteurs à cette prise. Tous les écouteurs compatibles « iPhone » ou « Android » devraient fonctionner.

Antenne

Une antenne est nécessaire pour utiliser le SIM900 pour tout type de communication vocale ou de données ainsi que certaines commandes SIM.

Prise SIM

Il y a une prise SIM à l'arrière. Toute carte SIM 2G pleine taille activée fonctionnerait parfaitement. Le fonctionnement du support de la carte SIM peut prendre un certain temps pour s'y habituer. Pour déverrouiller le loquet, poussez la partie supérieure de l'assemblage, puis soulevez-le. Insérez la carte SIM dans la partie inférieure de la prise. Ensuite, repliez le bras dans le corps de la douille et poussez-le doucement vers l'avant vers la position LOCK. **RTC (Horloge en temps réel)**

Le shield SIM900 peut être configuré pour garder l'heure. Il n'y a donc pas besoin d'un RTC séparé. Cela gardera l'heure même lorsque l'alimentation est coupée [14].

Branchement SIM900 Shield a Arduino :

- GND → GND
- VCC → 5V
- RX → TX
- TX → RX

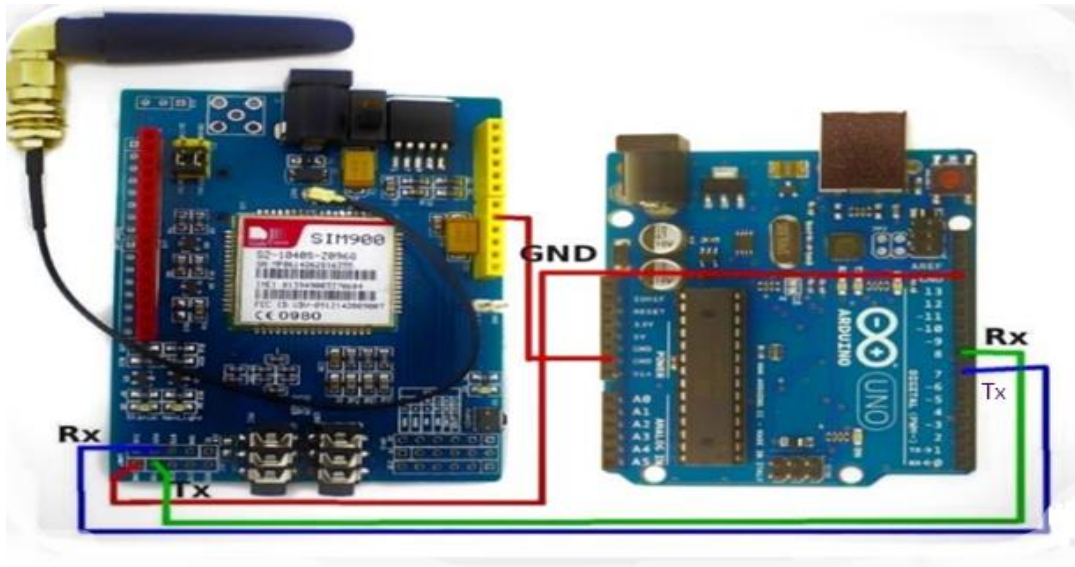


Figure II.31 : SIM900 Shield Branchement avec Arduino

Pour commandée la carte SIM900a utilisée les commande AT .

Les commandes AT :

Les commandes AT sont des jeux de commandes textuelles permettant de gérer la plupart des modems ou des modules GSM. Ces commandes commencent toujours par les lettres « AT » et se terminent obligatoirement par un retour chariot.

AT : Code d'attention. C'est le préfixe d'une ligne de commande qui indique au modem qu'une commande ou une séquence de commandes va être envoyée.

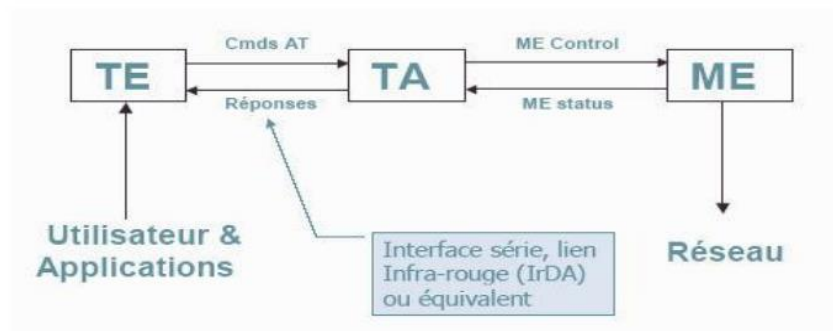


Figure II.32 : Schéma de fonctionnement des commandes AT[17].

- TE (Terminal Equipement) : envoi et affiche les Commandes .
- ME (Mobile Equipement) .
- TA (Terminal Adaptateur) : interface entre l'utilisateur et le mobile .

CHAPITRE II: Matériels et logiciels utilisés

Les types des commandes AT et les réponses :

- Commande de test: AT+<X>=?

Pour la commande test, le ME renvoie la liste des paramètres et des plages de valeurs définies avec la commande d'écriture correspondante par processus internes .

- Commande pour lire : AT+<X> ?

Elle renvoie la valeur actuellement définie du paramètre ou des paramètres.

- Commande pour écrire : AT+<X>=<...>

Cette commande définit les valeurs de paramètre défini par l'utilisateur.

- Commande d'exécution : AT+<X>

La commande d'exécution lit les paramètres non variables affectés par les processus internes dans le GSM.[17]

Commandes AT dédiées au service SMS :

AT+CSMS	Sélection du service de messagerie
AT+CPMS	Sélection de ta zone mémoire pour le Stockage des SMS
AT+CMGF	Sélection du format du SMS(PDU ou TEXT)
AT+CSCA	Définition de l'adresse du centre de messagerie
AT+CSDH	Affiche en mode TEXT le paramétrage des SMS
AT+CSAS	Sauvegarde du paramétrage
AT+CRES	Restauration du paramétrage par défaut
AT+CNMI	Indication concernant un nouveau SMS
AT+CMGL	Liste les SMS stockés en mémoire
AT+CMGR	Lecture d'un SMS
AT+CMGS	Envoie un SMS
AT+CMSS	Envoie d'un SMS stocké en mémoire
AT+CMGW	Écriture d'un SMS
AT+CMGD	Efface un SMS

Tableau II. 4:Commande AT dédiées service SMS[17]

II.6.2. Partie Logicielle

II.6.2.1Le logiciel Arduino

Le logiciel de programmation de la carte Arduino de code (langage proche de C). Une fois, le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte à travers de la liaison USB. Le câble USB alimenté à la fois en énergie la carte et transporte aussi l'information, ce programme appelé IDE Arduino. Comme n'importe quel langage de

programmation, une interface souple et simple est exécutable sur n'importe quel système d'exploitation, Arduino basé sur la programmation en C.

II.6.2.2.Présentation de «IDE » :

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 .

II.6.2.3.Structure générale du programme (IDE Arduino) :

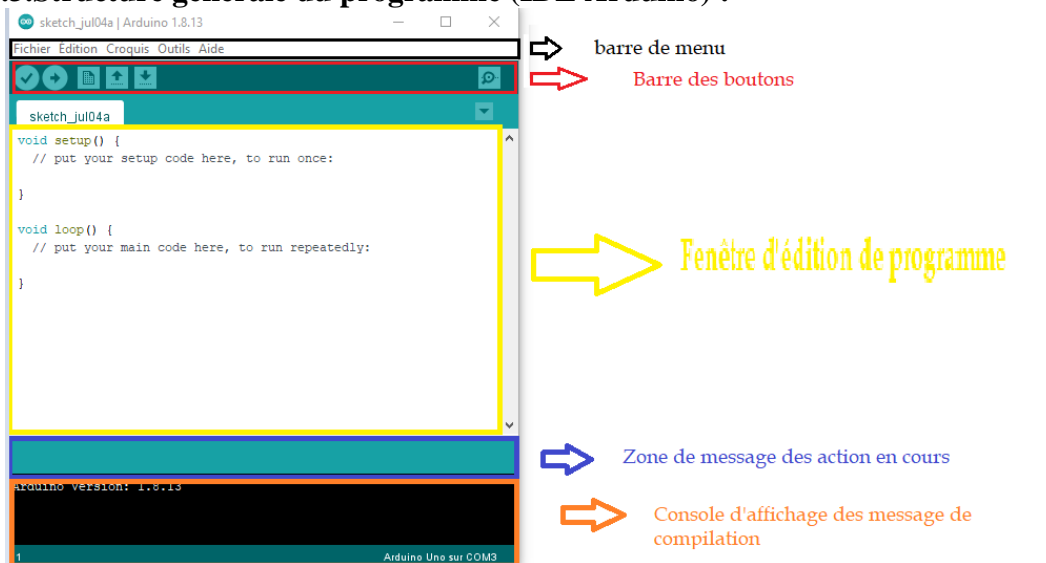


Figure II.33:Interface IDE Arduino 1.8.13

Cet espace de développement intégré (IDE) 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.

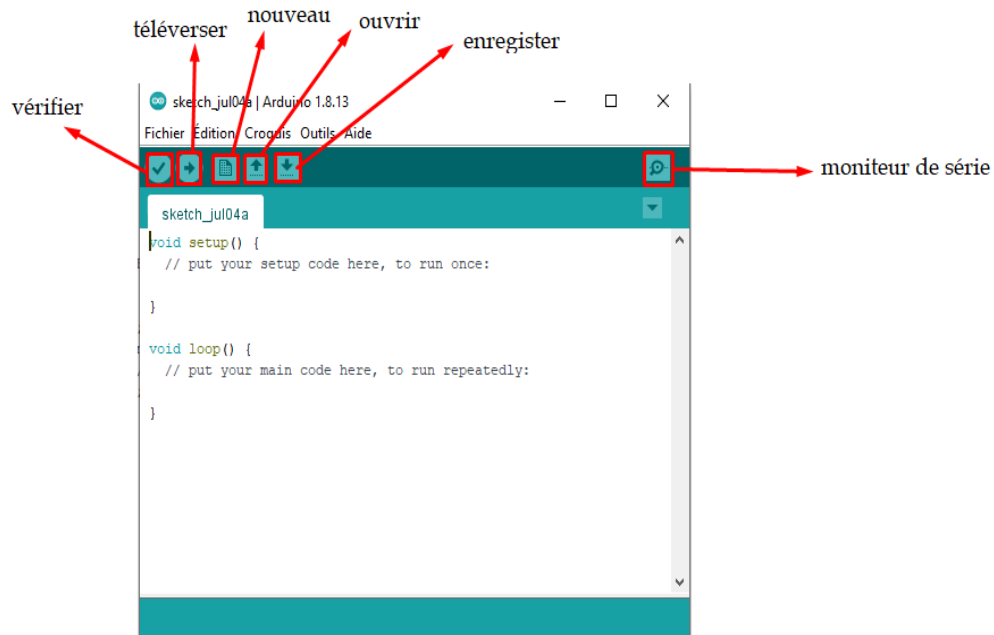


Figure II.34: Barre de boutons IDE Arduino

Vérifier : Permet de vérifier le programme, il actionne un module qui cherche les erreurs dans votre programme.

Téléverser: Charge (téléverse) le programme dans la carte .

Nouveau :Crée un nouveau fichier .

Ouvrir :Ouvre un fichier .

Enregistrer: Enregistre le fichier .

Moniteur série : Ouvre le moniteur série

II.6.2.4. Injection du programme :

Avant d'envoyer un programme dans la carte, il est nécessaire de sélectionner le type de la carte (Arduino UNO) et le numéro de port USB (COM 3) comme à titre d'exemple cette figure suivante.

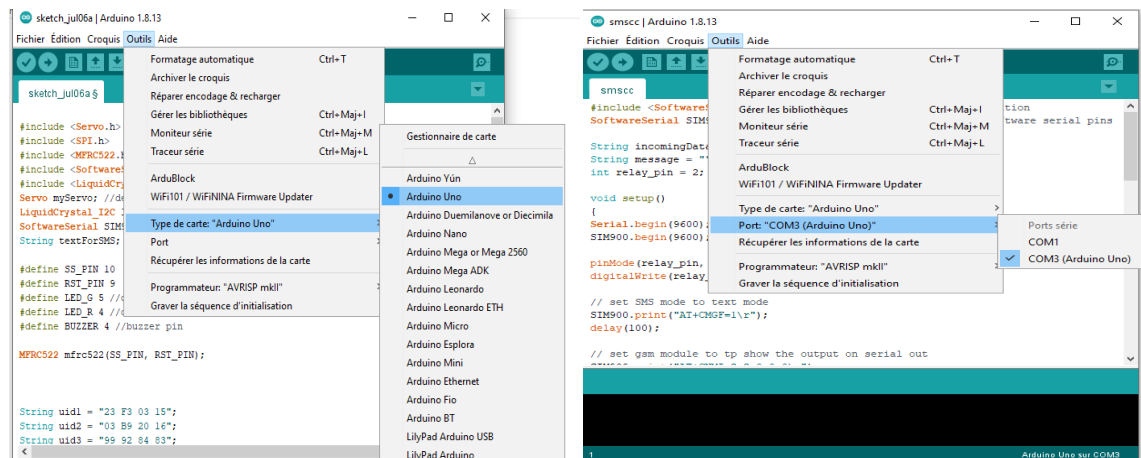


Figure II.35: Paramétrage de la carte

Pour ajouter une bibliothèque :

Choisir : croquis → inclure une bibliothèque → ajouter la bibliothèque .ZIP

Ensuite, nous choisissons un fichier de la bibliothèque au format ZIP

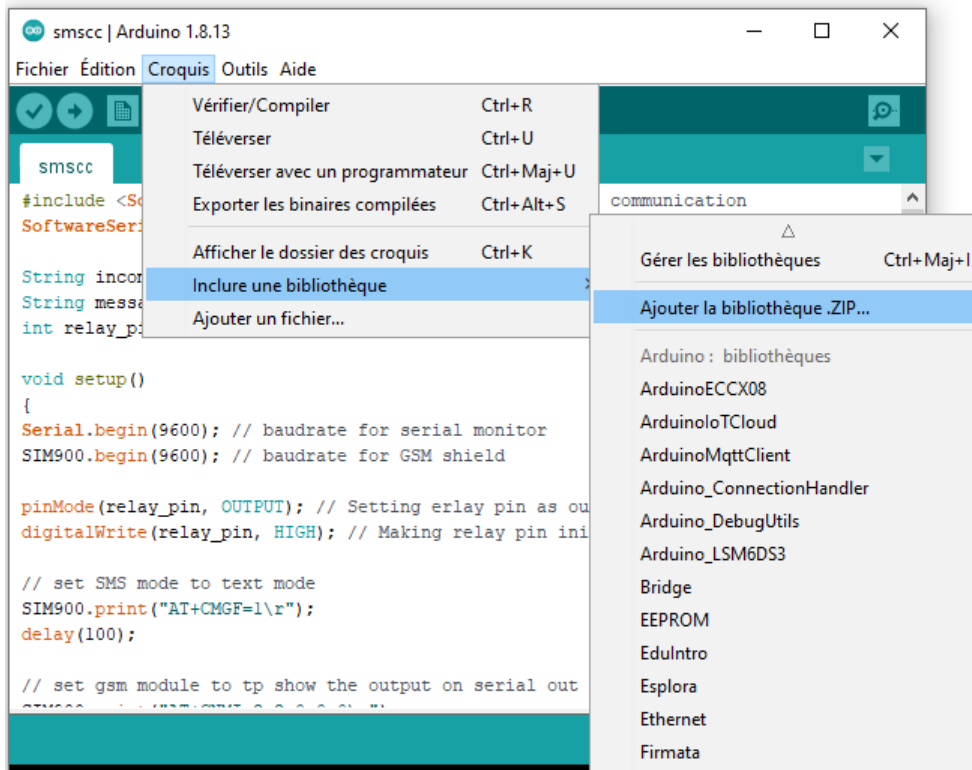


Figure II.36 : ajouter une bibliothèque

II.6.2.5.Fritzing :

Fritzing est une initiative matérielle open source qui rend l'électronique accessible en tant que matériau créatif pour tout le monde. Nous proposons un outil logiciel, un site Web communautaire et des services dans l'esprit de Processing et Arduino , favorisant un écosystème créatif qui permet aux utilisateurs de documenter leurs prototypes, de les partager avec d'autres, d'enseigner l'électronique dans une salle de classe et de concevoir et fabriquer des circuits imprimés professionnels [18].

Nous utiliserons ce logiciel pour monter le projet et l'Arduino .

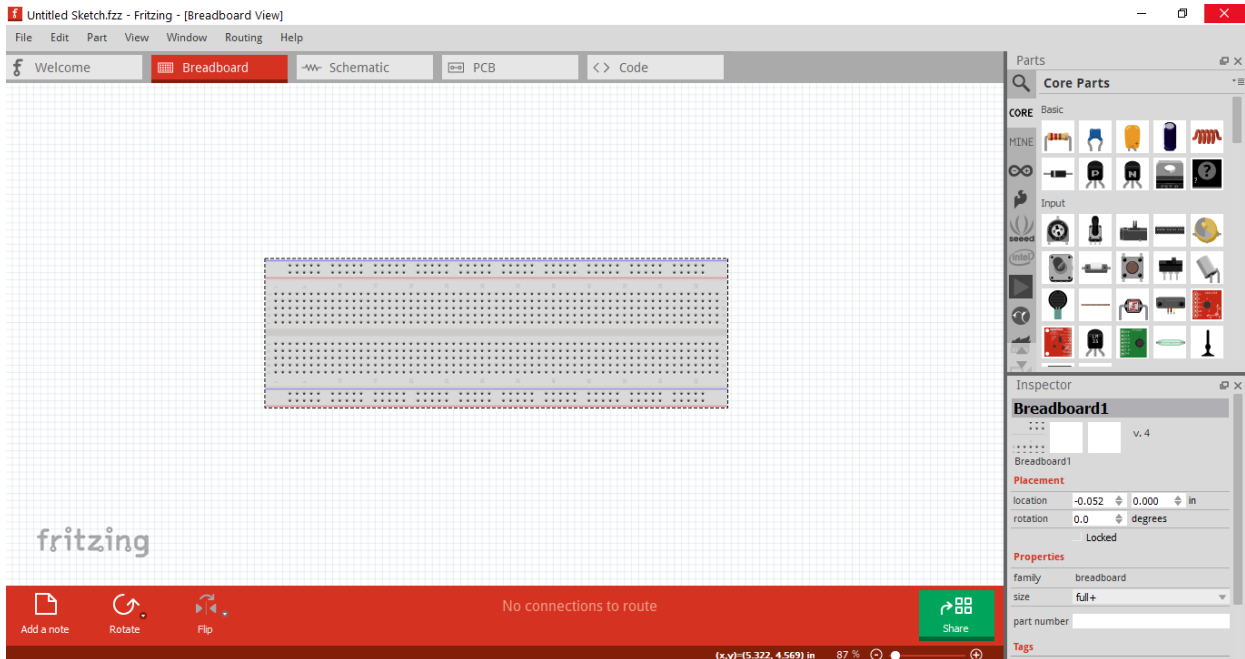


Figure II.37 : L'interface de Fritzing

II.7. Conclusion :

Dans ce chapitre, nous avons présenté le principe de fonctionnement et les caractéristiques de la carte Arduino, et expliqué les deux parties de base d'Arduino : la partie matérielle et la partie programmation. Aussi pour le circuit de mesure, nous avons fourni tous les composants utilisés qui sont : Le module GSM SIM900a des capteurs (température et humidité, capteur de flamme, capteur à ultrasons, capteur de mouvement et RFID mrfc522), les actionneurs (pivots angulaires, buzzer, LED, LCD I2C).

Enfin, nous avons décrit toutes les plates-formes informatiques basées sur Arduino dans la conception d'un système domotique.

CHAPITRE III :

PID Sous ARDUINO

III.1. Introduction

La commande des systèmes est l'un des domaines les plus riches en termes d'algorithmes et d'outils de développement. Le correcteur PID est de loin l'algorithme de commande le plus simple et le plus utilisé, à 90% dans les applications industrielles et largement utilisé dans des autres applications domestiques comme la domotique. Dans ce chapitre nous présentons une introduction sur la commande PID. L'algorithme de base et les différentes propriétés de cette dernière dans la boucle de commande [16,19].

III.2. Asservissement et régulation

Un asservissement est un algorithme dont l'objet principal est d'atteindre le plus rapidement possible sa valeur de consigne et de la maintenir, quelles que soient les perturbations externes. Le principe général est de comparer la consigne et l'état du système de manière à le corriger efficacement. La régulation (ou constante d'asservissement) doit agir pour qu'une mesure soit proche que possible à la consigne [20], la figure (III.1) représente la boucle de commande (correction) d'un système [22].

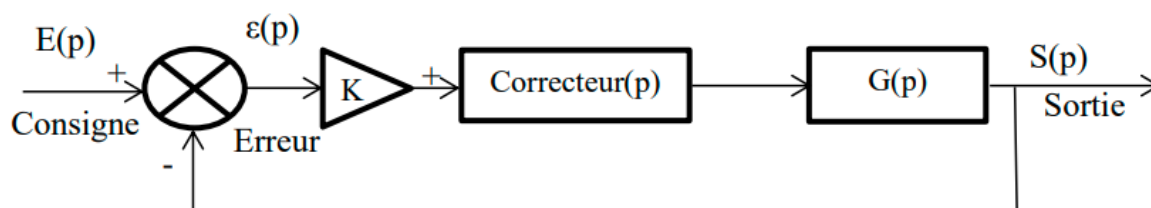


Figure III.1 : Boucle de commande d'un système asservi.

III. 3. Commande PID

La commande PID par sa simplicité de conception est la commande la plus utilisée. La synthèse d'une commande PID consiste à déterminer trois paramètres à savoir : le gain proportionnel K_p , le gain d'intégration K_i et le gain de dérivation K_d [22].

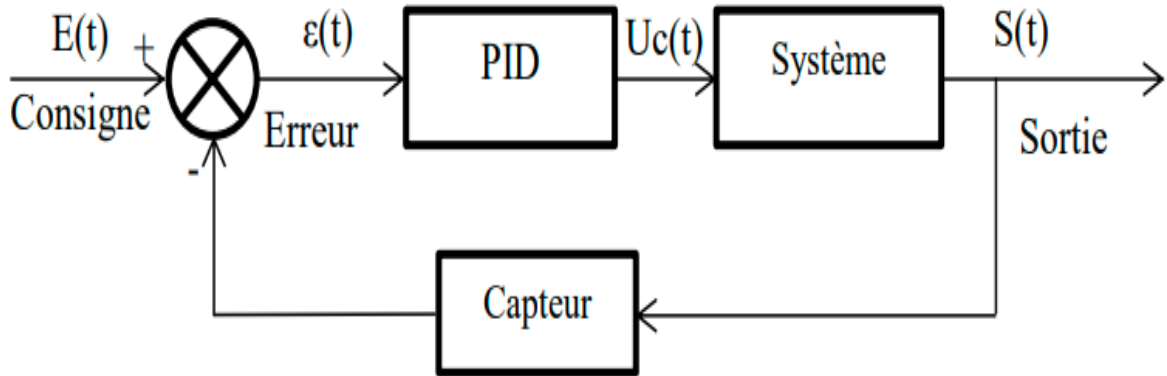


Figure III.2 : Schéma bloc d'un système avec correcteur PID.

III.3.1. Régulateur proportionnel P

Dans le cas du contrôle proportionnel, l'erreur est virtuellement amplifiée d'un certain gain constant qui devra être déterminé en fonction du système [23].

$$u(t) = k_p \varepsilon(t) \quad (\text{III.1})$$

Ce qui en Laplace donne :

$$U(p) = k_p \varepsilon(p) \quad (\text{III.2})$$

Où : k_p est le gain et $u(t)$ est la commande.

L'idée étant d'augmenter l'effet de l'erreur sur le système afin que celui-ci réagisse plus rapidement aux changements de consignes. Plus la valeur de K_p est grande, plus la réponse l'est aussi. En revanche, la stabilité du système s'en trouve détériorée et dans le cas d'un k_p démesuré le système peut même diverger.

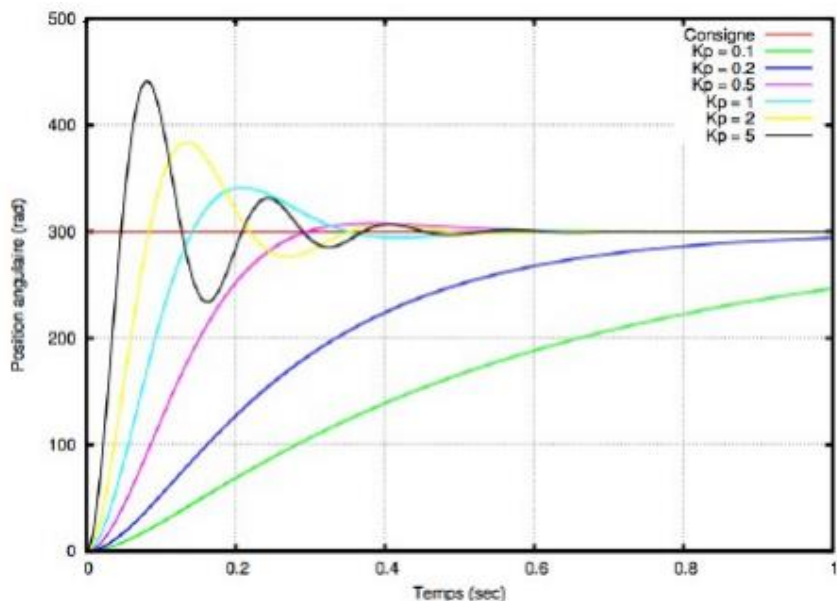


Figure III.3 : Modélisation de l'effet du régulateur P sur la réponse à un échelon d'un système.

III.3.2. Régulateur Dérivé I

Au contrôle proportionnel, nous pouvons ajouter l'intégration de l'erreur. Dans ce cas nous obtenons une régulation PI^2 . L'erreur entre la consigne et la mesure est ici intégrée par rapport au temps et multipliée par une constante qu'il faudra aussi déterminer en fonction du système [23].

$$u(t) = k_p \varepsilon(t) + k_i \int_0^t \varepsilon(\tau) d\tau \quad (\text{III.3})$$

Ce qui en Laplace donne :

$$U(p) = k_p \varepsilon(p) + k_i \frac{\varepsilon(p)}{p} \quad (\text{III.4})$$

Lors d'un simple contrôle proportionnel, il subsiste une erreur statique. Lorsque le système s'approche de sa consigne, l'erreur n'est plus assez grande pour faire avancer le moteur. Le terme intégral permet ainsi de compenser l'erreur statique et fournit, par conséquent, un système plus stable en régime permanent. Plus K_i est élevé, plus l'erreur statique est corrigée.

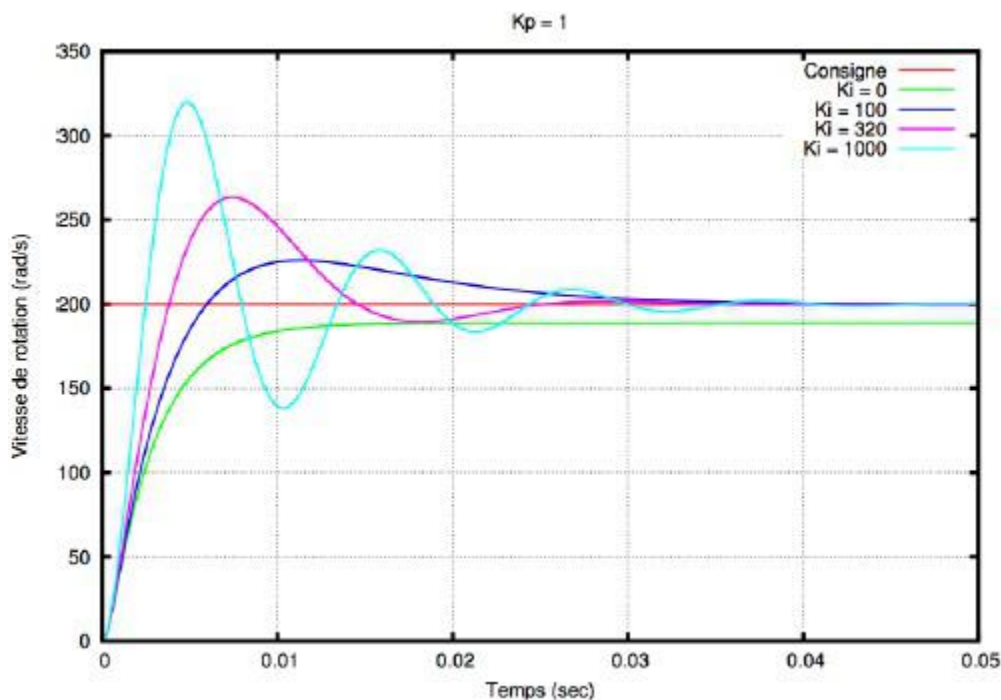


Figure III.4 : Modélisation de l'effet du régulateur I sur la réponse à un échelon d'un système.

III.3.3. Régulateur Dérivé D

Pour obtenir un contrôle en PID, il nous faut encore rajouter un terme. Celui-ci consiste à dériver l'erreur entre la consigne et la mesure par rapport au temps et à le multiplier lui aussi par une constante.

$$u(t) = k_p \varepsilon(t) + k_i \int_0^t \varepsilon(\tau) d\tau + k_d \frac{d\varepsilon(\tau)}{dt} \quad (\text{III.5})$$

Ce qui en Laplace donne :

$$U(p) = k_p \varepsilon(p) + k_i \frac{\varepsilon(p)}{p} + k_d p \quad (\text{III.6})$$

Nous avons besoin d'un terme dérivé car le contrôle PI peut amener à un dépassement de la consigne, ce qui n'est pas toujours très souhaitable (exemple d'inversion de polarité dans le cas de moteurs électriques). Le terme dérivé permet de limiter cela. Lorsque le système s'approche de la consigne, ce terme freine le système en appliquant une action dans le sens opposé et permet ainsi une stabilisation plus rapide [23].

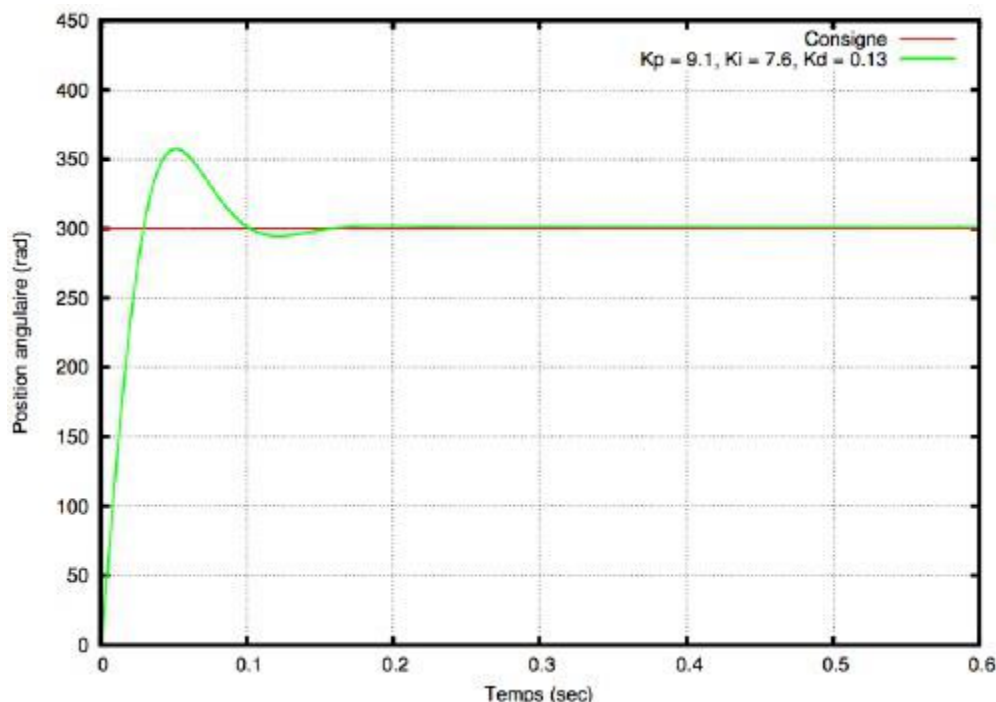


Figure III.5: Modélisation de l'effet du régulateur PID sur la réponse à un échelon d'un système.

III.3.4. La méthode de réglage d'un correcteur PID

Il existe deux façons de trouver les valeurs à attribuer aux trois coefficients (k_i, k_p, k_d) de l'asservissement PID, l'une par la modélisation, l'autre par l'expérimentation, sachant que souvent l'on complète la première à travers la deuxième. Modéliser le système consiste à déterminer par le calcul son comportement et de la déduire des valeurs plausibles pour les coefficients. L'approche par l'expérimentation signifie que l'on va utiliser une réponse réelle du système pour régler d'abord grossièrement puis finement les coefficients [23].

III.4. Implémentation du correcteur PID avec Arduino

L'objectif est d'implémenter et régler efficacement les paramètres d'un correcteur PID numérique réel. Contrairement à un PID théorique, le PID numérique dispose d'un filtre passe-bas en HF afin de limiter les perturbations. En revanche, la bande passante du filtre est directement liée à l'efficacité de l'effet dérivateur. Autrement dit, la stabilité du système (voir le tuto pour plus de détails).

Il existe plusieurs techniques de réglage des paramètres d'un correcteur PID, en particulier la méthode de Ziegler-nichols. On verra une astuce qui va vous permettre de régler les paramètres du correcteur facilement en se basant sur un modèle du second ordre du correcteur. En effet, le correcteur sera décrit par une équation du deuxième ordre muni du filtre en HF.

Le coefficient d'amortissement du PID doit être supérieur ou égal à 1. Il agit sur la largeur de bande de fréquence dédiée à l'intégrateur ainsi le dérivateur.

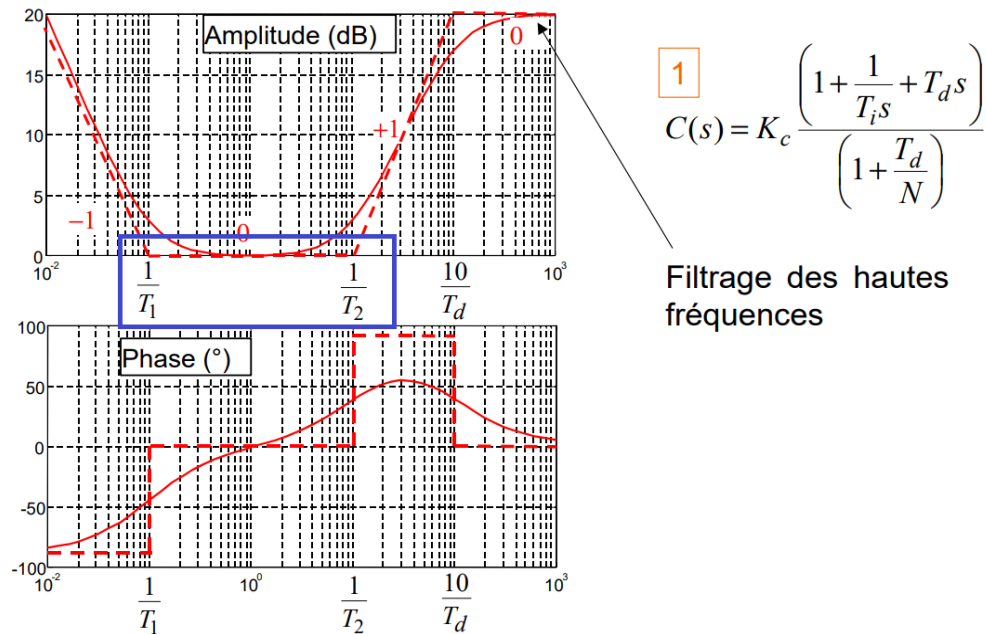


Figure III.6 : Réponse fréquentielle du système.

En pratique, la pulsation de coupure du système physique W_n (ou sa bande passante) doit être connue d'avance afin d'en déduire les paramètres du réglage du correcteur PID. La réponse à un échelon en boucle ouverte peut être utilisée afin de mesurer W_n en se basant sur le temps de montée, le dépassement, la fréquence des oscillations, etc. Dans notre tuto la fréquence du système est fixée à 5Hz, zeta à 0.05 [24].

III.4.1. La fonction PID()

Cette fonction sert à implémentée le modèle numérique du correcteur PID. Elle prend en entrée la sortie du soustracteur (ou l'erreur epsilon(n)), le paramètre N est lié à la bande passante du filtre HF, la période d'échantillonnage ainsi les paramètres du correcteur PID : 1. La coefficient d'amortissement zeta du correcteur au voisinage de 1 (≥ 1), 2. La pulsation de coupure du correcteur (au voisinage du celui du système physique W_n) de préférence $< W_n$ et 3. La contrastante k0 ou l'action proportionnelle du correcteur PID. En suite la fonction retourne la sortie du correcteur. Les figures ci-dessous représente l'implémentation du code de la fonction PID sous Arduino [24].

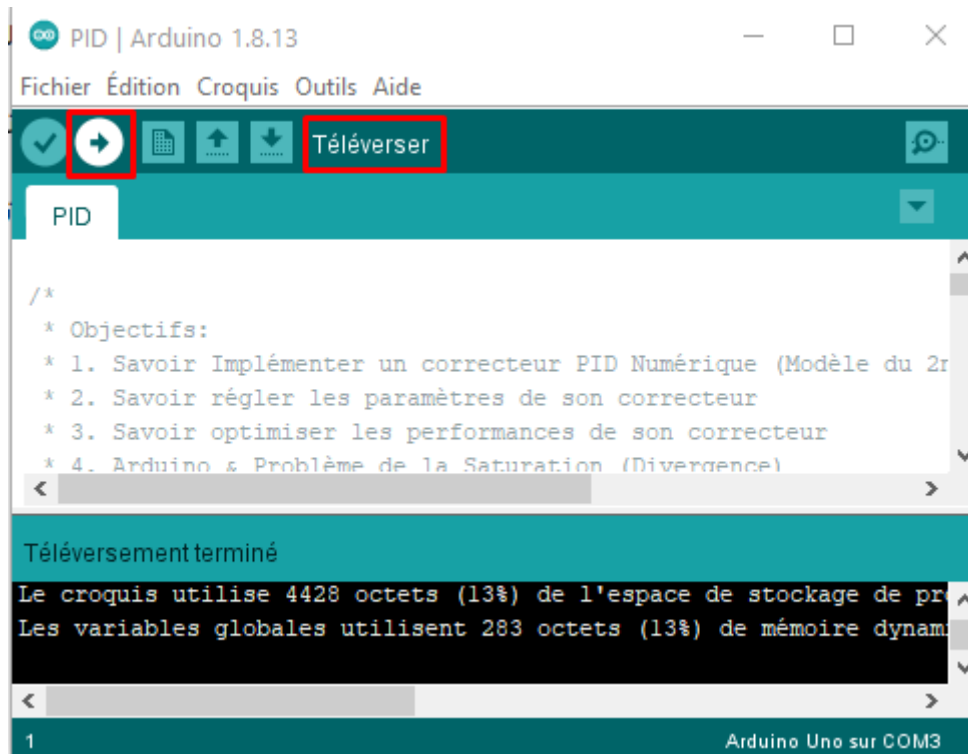


Figure III.7 : envoi du code vers l'Arduino

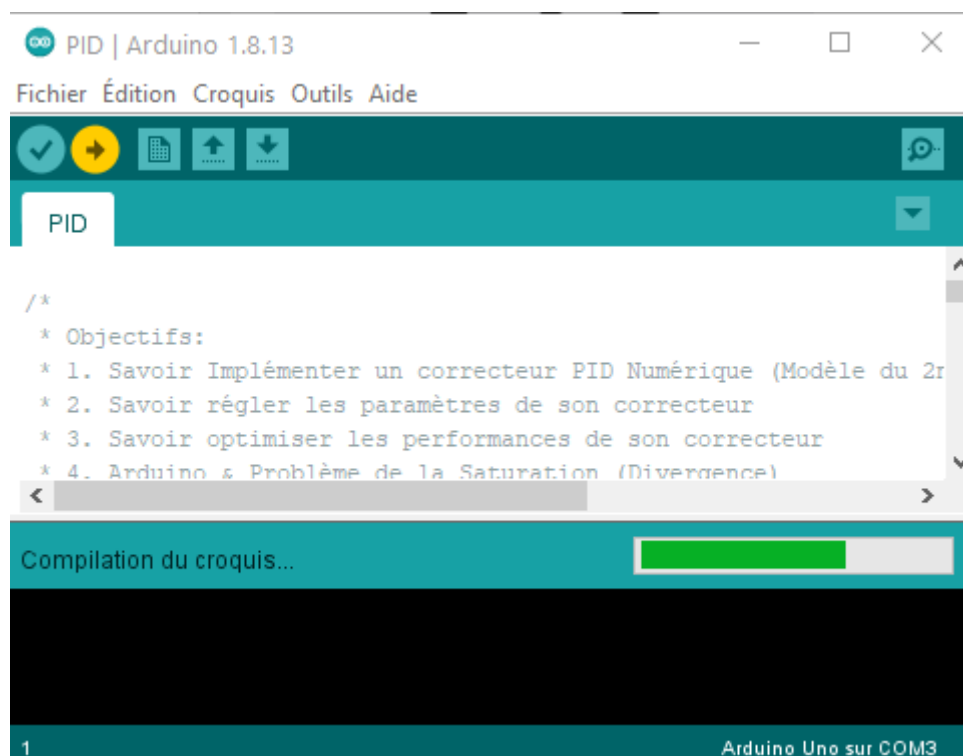


Figure III.8 : Exécution du programme.

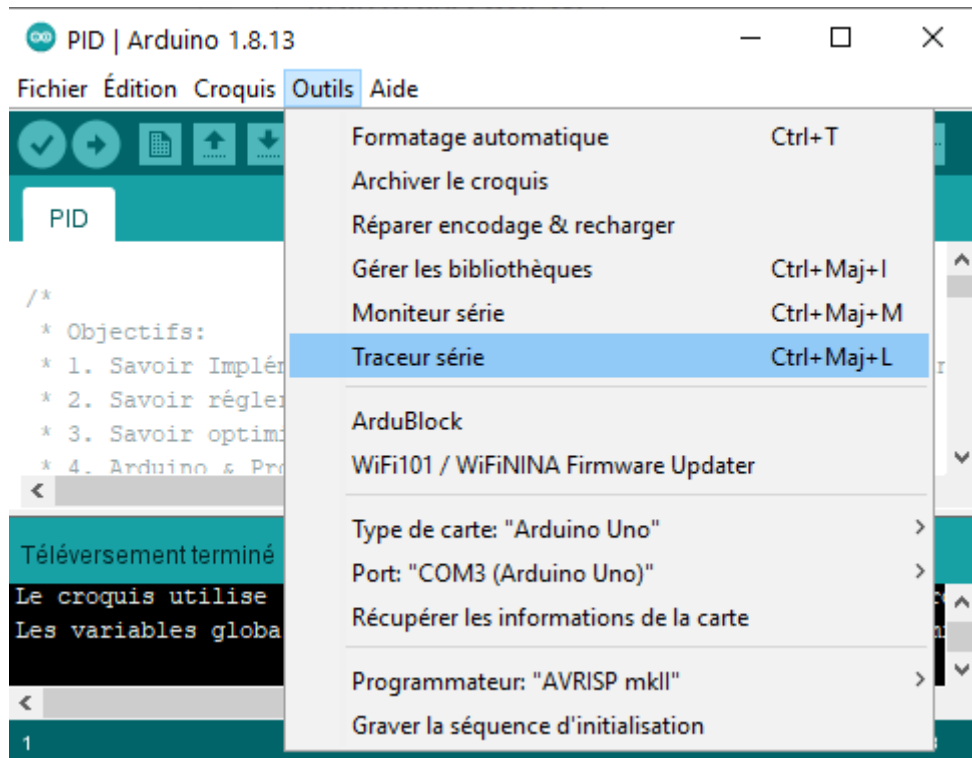


Figure III.9 : Traçage des réponses.

Ci-dessous un exemple du test du correcteur PID d'un système oscillant.

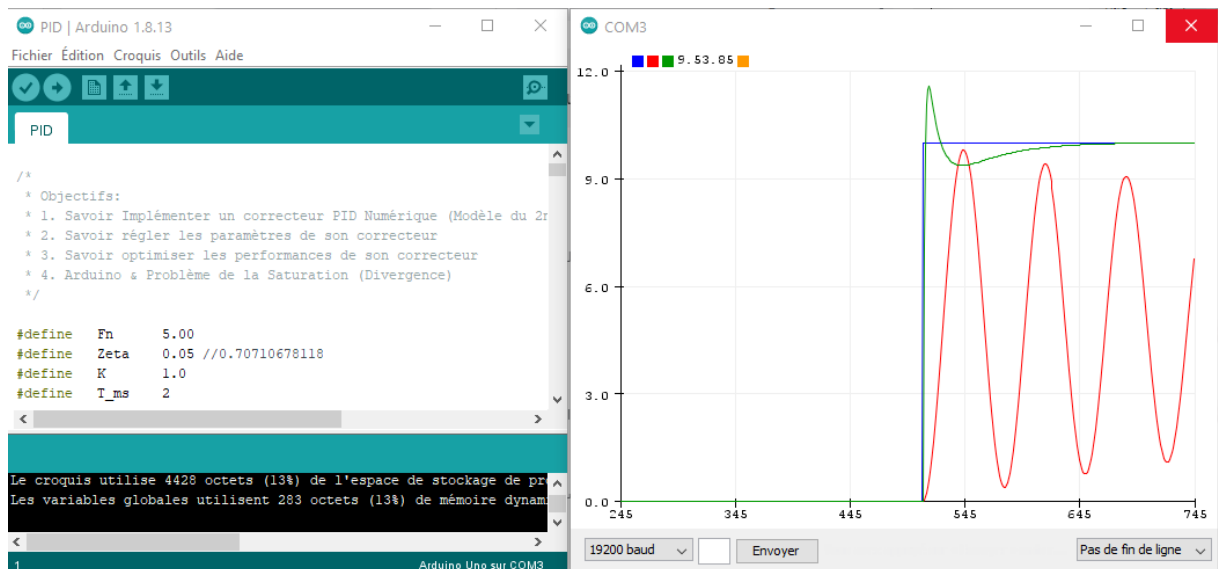


Figure III.10 : système oscillant.

En variée les paramètres du correcteur PID, en obtient :

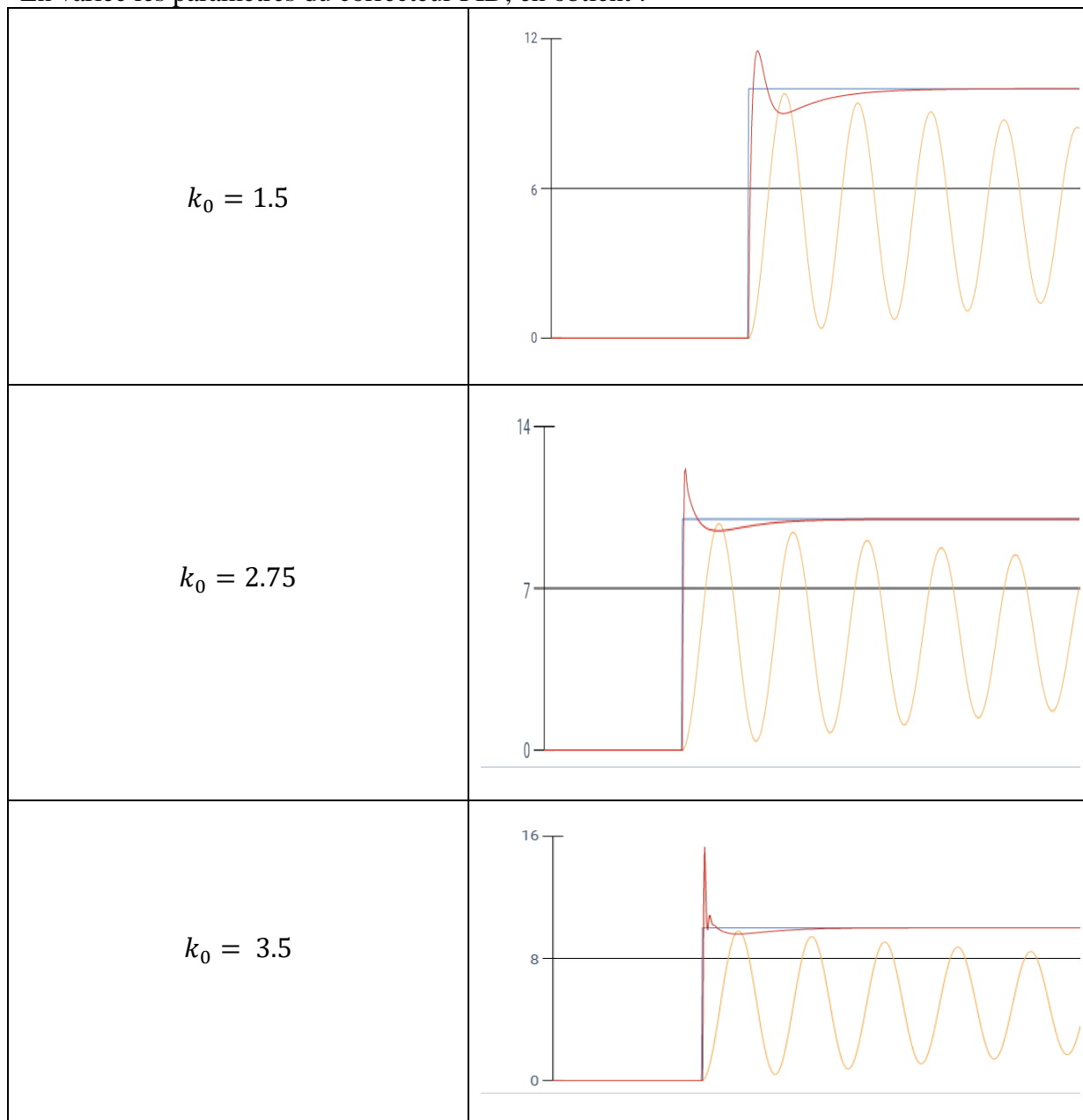


Tableau III. 1 : Variation des réponses des systèmes corrigé et non corrigé à la variation de gain.

Dans cet exemple nous avons testé l'effet d'un contrôleur PID sur un système instable modélisé par une fonction de transfert d'ordre 2. Nous avons utilisé la carte ARDUINO comme un contrôleur PID à travers le programme défini précédemment. Dans notre test nous allons changer la valeur de l'action proportionnelle du contrôleur PID, en trois valeurs, dans la première valeur le système devient stable mais un peu lent, dans la deuxième valeur le système devient plus rapide que le premier, et dans la dernière valeur le système devient très rapide mais le premier déplacement devient un peu plus important et qui peut détruire notre

système, l'objectif de cet exemple est juste de tester bon fonctionnement du contrôleur PID sur une carte ARDUINO.

III.5. Conclusion

L'asservissement par PID est aujourd'hui l'un des asservissements les plus utilisés à cause de leur simplicité dans la mise en œuvre, Dans ce chapitre, nous avons présenté un aperçu général sur la commande PID et l'effet de chaque action du régulateur sur les performances des systèmes asservis. Ensuite nous allons utiliser une carte Arduino comme un contrôleur PID, et nous l'avons testé en tenant compte de plusieurs valeurs de l'action proportionnelle.

Chapitre IV : réalisation pratique d'un système intelligent

CHAPITRE IV : réalisation pratique d'un système intelligent

IV.1.Introduction :

Ce chapitre est la fin de ce projet, où nous présenterons toutes les étapes que nous avons suivies afin de réaliser notre projet sur une maquette de maison, nous commençons tout d'abord par un exemple très simple sur la carte Arduino pour allumer et éteindre des LED de feux de circulation, Ensuite nous abordons le cadre des charges de notre projet domotique et pour mettre en œuvre notre travail, nous avons réalisé une maquette d'une maison intelligente en utilisant une carte Arduino UNO, et en installant différents capteurs (mouvement, incendie, températureetc.)qui permettent à l'utilisateur de contrôler à distance de manière fiable, sécurisée et automatique et de recevoir des informations par SMS via une carte GSM SIM900.

IV. 2. Test de matérielle utilisée :

- **Test la carte Arduino UNO :**

Chaque carte Arduino UNO est livrée avec un code de programme à l'intérieur (appelé clignotement) afin de s'assurer que l'Arduino fonctionne correctement, la LED s'allume pendant une seconde, puis s'éteint pendant une seconde, à plusieurs reprises. L'Arduino est connecté avec la broche 13 directement, lorsque l'Arduino est connecté à une source d'alimentation (5V à 12V), l'Arduino clignotera à travers la LED L, et une LED verte fonctionne également avec elle, indiquant que l'Arduino est dans un état de fonctionnement (c'est-à-dire qu'il est connecté à une source d'alimentation 5V / 12V).

Afin de s'assurer que l'Arduino fonctionne correctement, nous ouvrons le programme Arduino IDE. Ensuite, nous passons aux exemples existants, puis choisissons basic et ensuite blink, comme le montre la Figure IV.1.

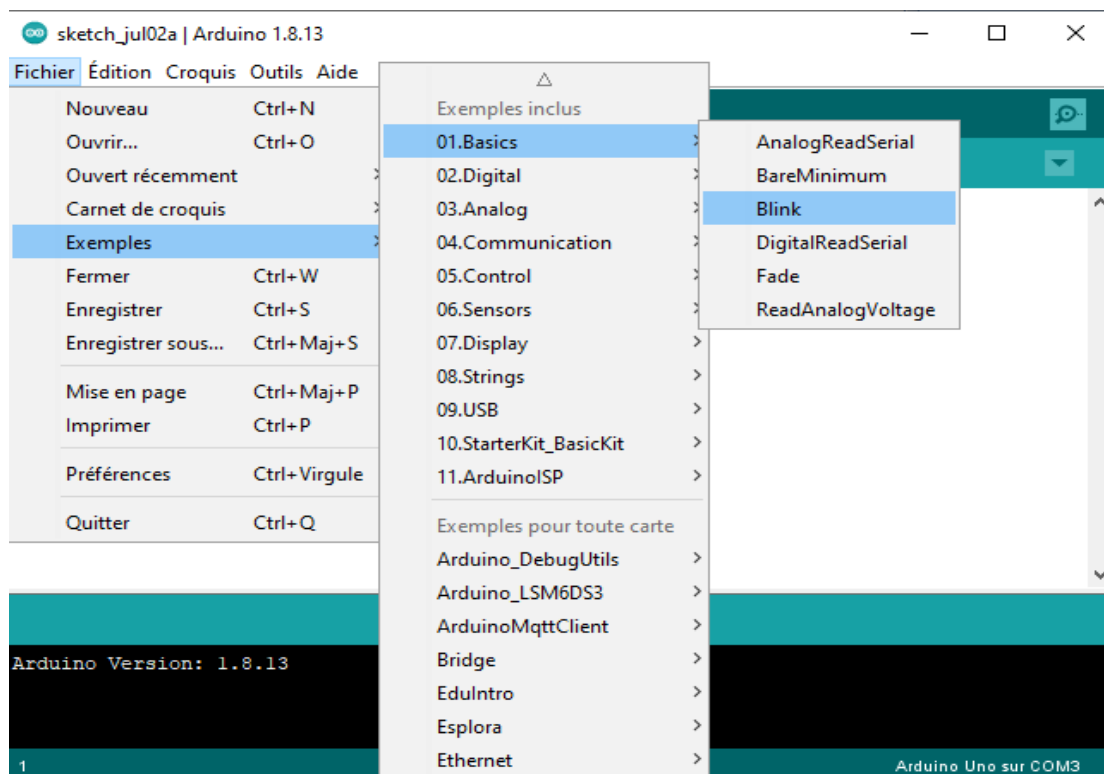


Figure IV.1 : ouvrir le exemple de BLIK

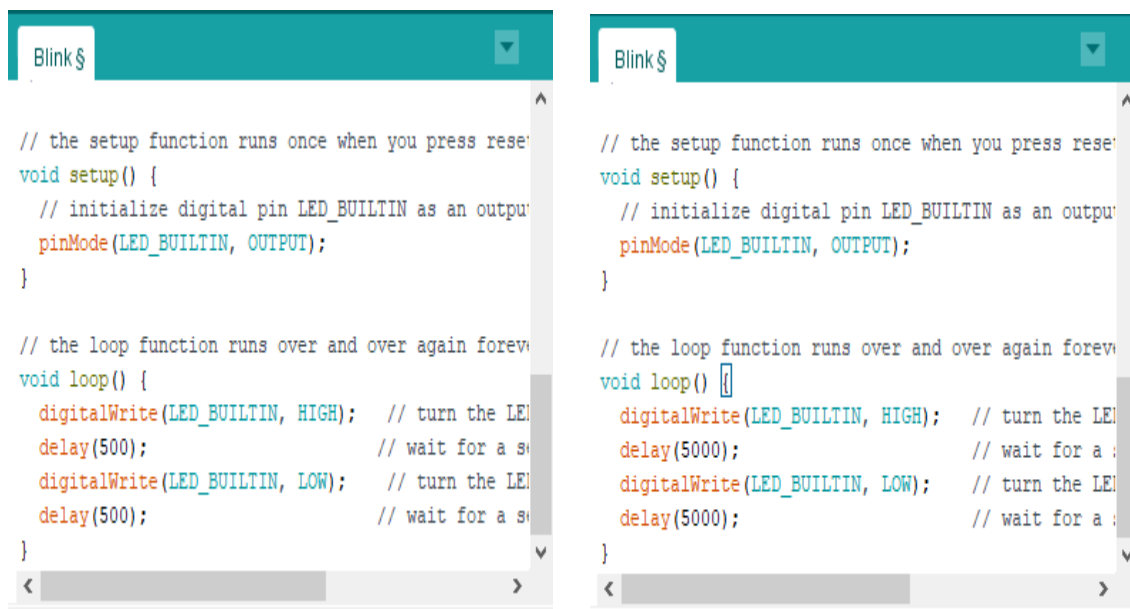
Le script de clignotement s'ouvre comme suit :

```
// the setup function runs once when you press reset or power the board
void setup() { // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
} // the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
```

```
digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
delay(1000); // wait for a second
}
```

Les instructions responsables du temps sont (delay), l'unité de temps avec laquelle l'Arduino fonctionne est la milliseconde. Ensuite, nous modifions le temps à 500 millisecondes, nous vérifions l'intégrité du code et après l'avoir vérifié, nous téléchargeons le script sur l'Arduino, et nous remarquons que l'Arduino clignote plus rapidement qu'avant .

Ensuite, nous modifions le temps à 5000 millisecondes, nous vérifions l'intégrité du code et après l'avoir vérifié, nous téléchargeons le script sur l'Arduino, et nous remarquons que l'Arduino clignote mais plus lentement qu'avant.



```
Blink $
```

```
// the setup function runs once when you press reset or power up
void setup() {
  // initialize digital pin LED_BUILTIN as an output:
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the positive voltage)
  delay(500); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(500); // wait for a second
}

Blink $
```

```
// the setup function runs once when you press reset or power up
void setup() {
  // initialize digital pin LED_BUILTIN as an output:
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the positive voltage)
  delay(5000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(5000); // wait for a second
}
```

Figure IV.2 :Les modification sur le programme de BLINK

- **Exemple 1 :Les feux de circulation**

Dans cet exemple nous avons réalisé sur réaliser feu rouge en utilisant les LED et les résistances et bien sur des fils de connexion. Son but est de réaliser deux feux de circulation et de les faire fonctionner de manière synchrone, où il fonctionnera comme suit : (voir la figure).

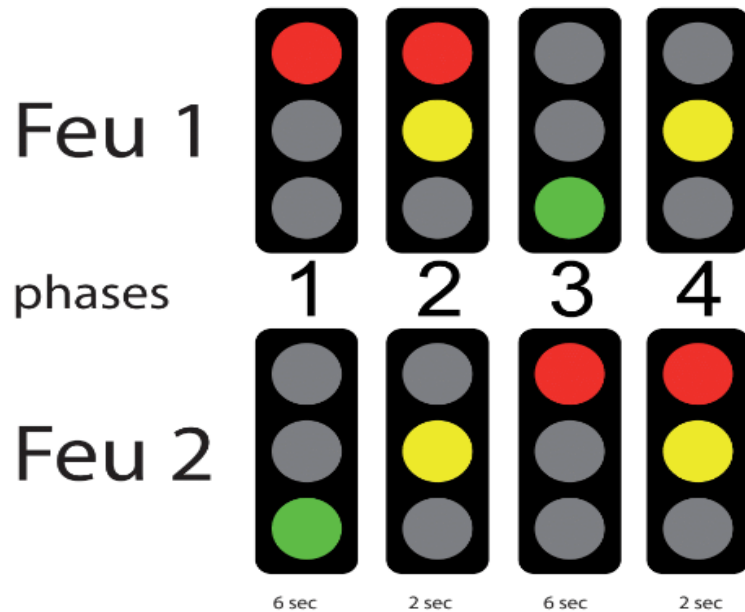


Figure IV.3: un exemple de feux de circulation réaliser

Avant de démarrer ce projet, nous avons utilisé le site tinkercad (<https://www.tinkercad.com>) pour faire un montage du projet et l'y simuler en ligne, afin d'éviter tout problème qui pourrait arriver par erreur à la carte Arduino.

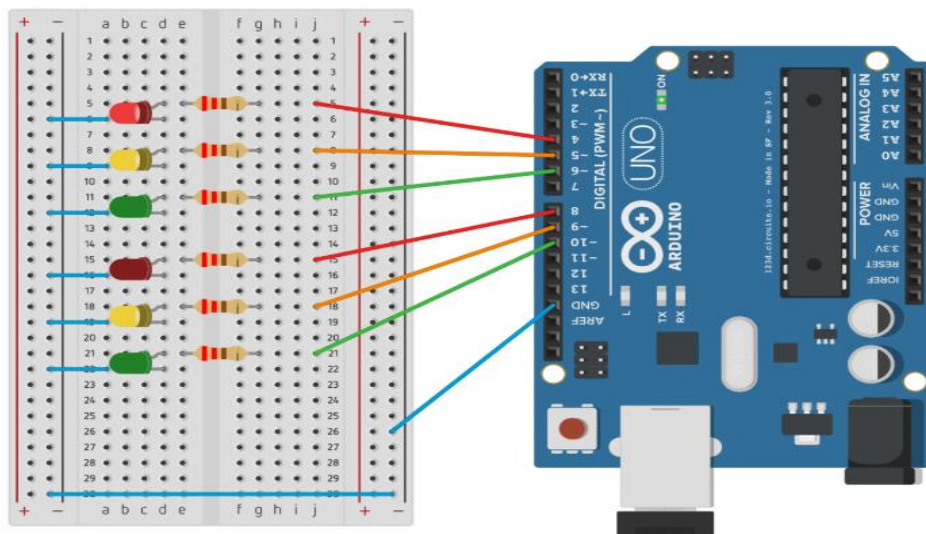


Figure IV.4 : schéma de câblage de Les feux de circulation

Lorsque l'assemblage et la simulation terminée, nous accédons dans le script (programme en langage C) que nous avons écrit précédemment pour le programme Arduino IDE, nous vérifions l'intégrité du code et ensuite nous le téléchargeons sur la carte Arduino pour que le projet soit fonctionnel.

Code de programme Arduino :

```
void setup()
{
//feu 1
pinMode(4, OUTPUT); // la sortie de led rouge
pinMode(5, OUTPUT); // la sortie de led jaune
pinMode(6, OUTPUT); // la sortie de led vert
//feu 2
pinMode(8, OUTPUT); // la sortie de led rouge
pinMode(9, OUTPUT); // la sortie de led jaune
pinMode(10, OUTPUT); // la sortie de led vert
}
void loop()
{
//phase 1
digitalWrite(4,HIGH);
digitalWrite(10,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
digitalWrite(9,LOW);
delay(6000); // 6 second
//phase 2
digitalWrite(4,HIGH);
digitalWrite(5,HIGH);
digitalWrite(9,HIGH);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
digitalWrite(10,LOW);
delay(2000); //2 second
//phase 3
digitalWrite(6,HIGH);
digitalWrite(8,HIGH);
digitalWrite(4,LOW);
digitalWrite(5,LOW);
digitalWrite(9,LOW);
digitalWrite(10,LOW);
delay(6000); // 6 second
//phase 4
digitalWrite(5,HIGH);
digitalWrite(8,HIGH);
digitalWrite(9,HIGH);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(10,LOW);
delay(2000); // 2 second
}
```

L'objectif de cet exemple est de donner une idée générale sur le fonctionnement de la carte ARDUINO, et de s'initier au langage de programmation ARDUINO.

IV.3. Réalisation du projet :

Le système que nous avons réalisé est composé de deux projets principaux, le premier est composé de 3 parties ; à savoir un détecteur de mouvement basé sur Arduino UNO, une détection d'incendie et un détecteur de température et d'humidité, le second est d'automatiser la fermeture et l'ouverture de la porte par un servomoteur, la prévention et l'activation d'alarme et un module d'affichage. La première étape consiste à créer une maison. Ce modèle de maison, permettrait de présenter certaines fonctionnalités de la domotique à travers 3 capteurs (Humidité et Température, Mouvement, Ultrason et Flamme.), Ensuite nous avons monté un servomoteur sur la porte de la maison afin d'automatiser son ouverture et sa fermeture

- **Partie 1 : Projet détecteur de mouvement**

En raison du phénomène très répandu de vol et de vandalisme de la propriété privée, des capteurs de détection de mouvement sont placés sur les fenêtres, les portes et le balcon de la maison .Ces capteurs sont utilisés pour donner une alerte en cas d'activité étrange, et ces capteurs peuvent être à l'intérieur du système de sécurité de la maison ou du système domotique, par exemple si un mouvement est détecté à la maison nous recevons un SMS pour le notifier

Le matériel que nous avons utilisé est le suivant :

a- Carte Arduino

b- Le capteur HC-SR04

Le détecteur HC-SR04 utilise les ultrasons pour déterminer la distance à laquelle se trouve un objet. Peu importe l'intensité de la lumière, la température ou le type de matière, le capteur pourra facilement détecter s'il y a un obstacle devant lui. Tout de fois, il peut être contraint sur certains types de couleurs tel que le noir (contraste), ou encore sur la matière comme le textile. Son champ de vision est de 90° environ selon l'environnement. Si une impulsion de plus de 10µS et détecte, alors le capteur envoie une série de 8 impulsions à ultrason de 40kHz et assiste au réfléchissement du signal. Ensuite, en ayant en tête la vitesse du son, il effectue un calcul rapide pour déterminer la distance. Et nous utiliserons Le détecteur HC-SR04 pour le projet de détection de mouvement en raison de l'absence d'un détecteur de mouvement (PIR) .en raison de sa bonne efficacité.

c- Carte GSM900a :

Nous utiliserons cette carte pour envoyer des messages SMS si un mouvement est détecté.

d- Un buzzer et LED rouge :

Il est utilisé comme une alarme pour émettre des sons d'avertissement

CHAPITRE IV : réalisation pratique d'un système intelligent

Le Cablage avec l'Arduino:

- Le capteur HC-SR04 et Arduino :
GND → GND
VCC → 5V
trig → pin9
echo → pin12
- LED rouge et Arduino :
(+) → pin3
(-) → résistance et GND de Arduino
- buzzer et Arduino :
GND → GND
VCC → 5V
SIGNAL → pin4
- carte GSM900a et Arduino :
GND → GND
RX → pin8 (déclarer comme TX)
TX → pin7 (déclarer comme RX)
On alimentée la carte GSM900a a pile 9V .

La figure IV.5 représente notre circuit de détecteur de mouvement

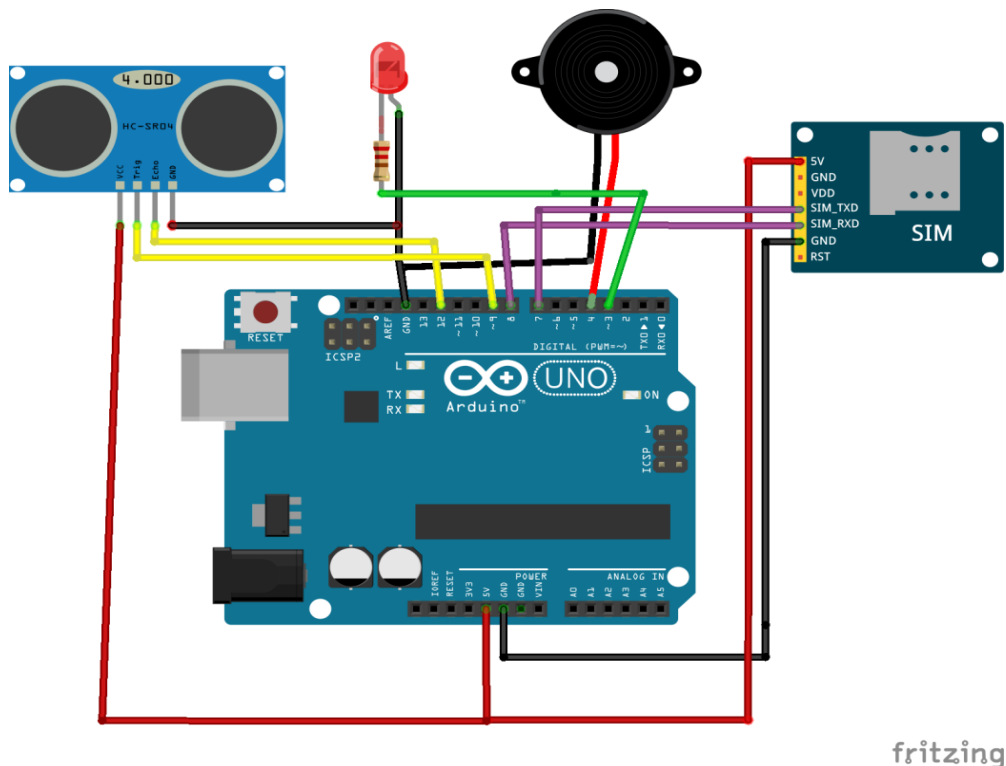


Figure IV.5: schéma de câblage de détecteur de mouvement

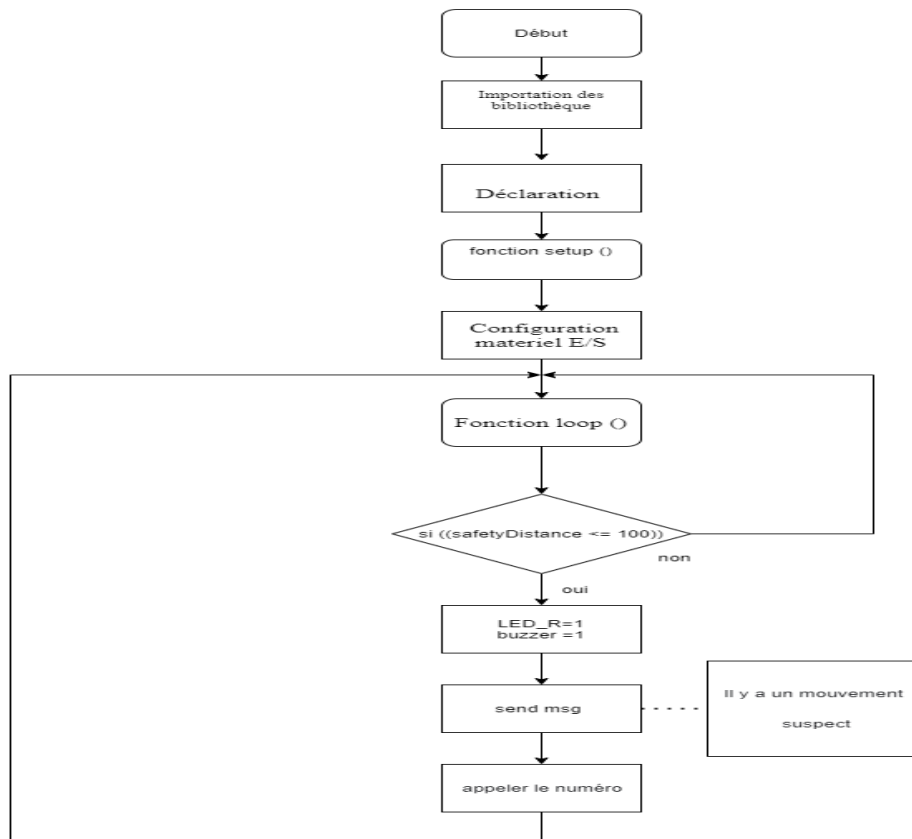


Figure IV.6: Organigramme de la fonction de détection de mouvement

Partie2 : Détecteur d'incendie (Flamme) :

Les incendies sont considérés comme l'un des accidents les plus dangereux en raison de leur gravité, qui provoquent de grandes pertes matérielles et humaines. Il a de nombreuses causes, il peut être causé par une fuite de gaz ou un court-circuit électrique ou par indifférence, négligence et oubli humaine, les raisons sont nombreuses, mais le résultat est un grand danger pour la vie.

Dans ce projet nous nous appuyerons sur un capteur de détection de flamme, qui fonctionne en détectant la lumière dans le spectre infrarouge, telle que la lumière produite par une flamme nue.

Le matériel que nous avons utilisé est le suivant :

a- Capteur de flamme :

Le module de détection de flamme est utilisé pour détecter une flamme nue en détectant la lumière qui est dans le spectre infrarouge est produite par une flamme nue.

b- carte GSM900a :

Nous utiliserons cette carte pour envoyer des SMS et des appels téléphoniques en cas de flamme.

CHAPITRE IV : réalisation pratique d'un système intelligent

c- Un buzzer et LED rouge :

Il est utilisé comme une alarme pour émettre des sons d'avertissement .

Le Cablage avec l'Arduino:

- Le capteur de flamme et Arduino :

GND → GND

VCC → 5V

A0 → A0

- LED rouge et Arduino :

(+) → pin3

(-) → en série avec la résistance et GND de Arduino

- buzzer et Arduino :

GND → GND

VCC → 5V

SIGNAL → pin4

- carte GSM900a et Arduino :

GND → GND

RX → pin8 (déclarer comme TX)

TX → pin7 (déclarer comme RX)

On alimentée la carte GSM900a a pile 9V .

La figure IV.7 représente notre circuit de détecteur d'incendie

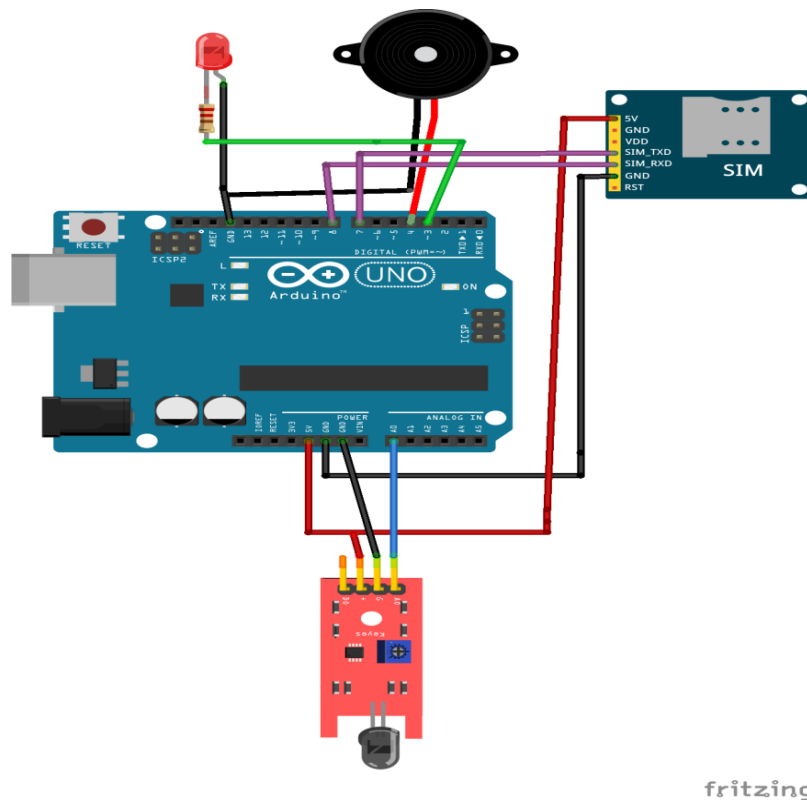


Figure IV.7: schéma de câblage de détecteur d'incendie

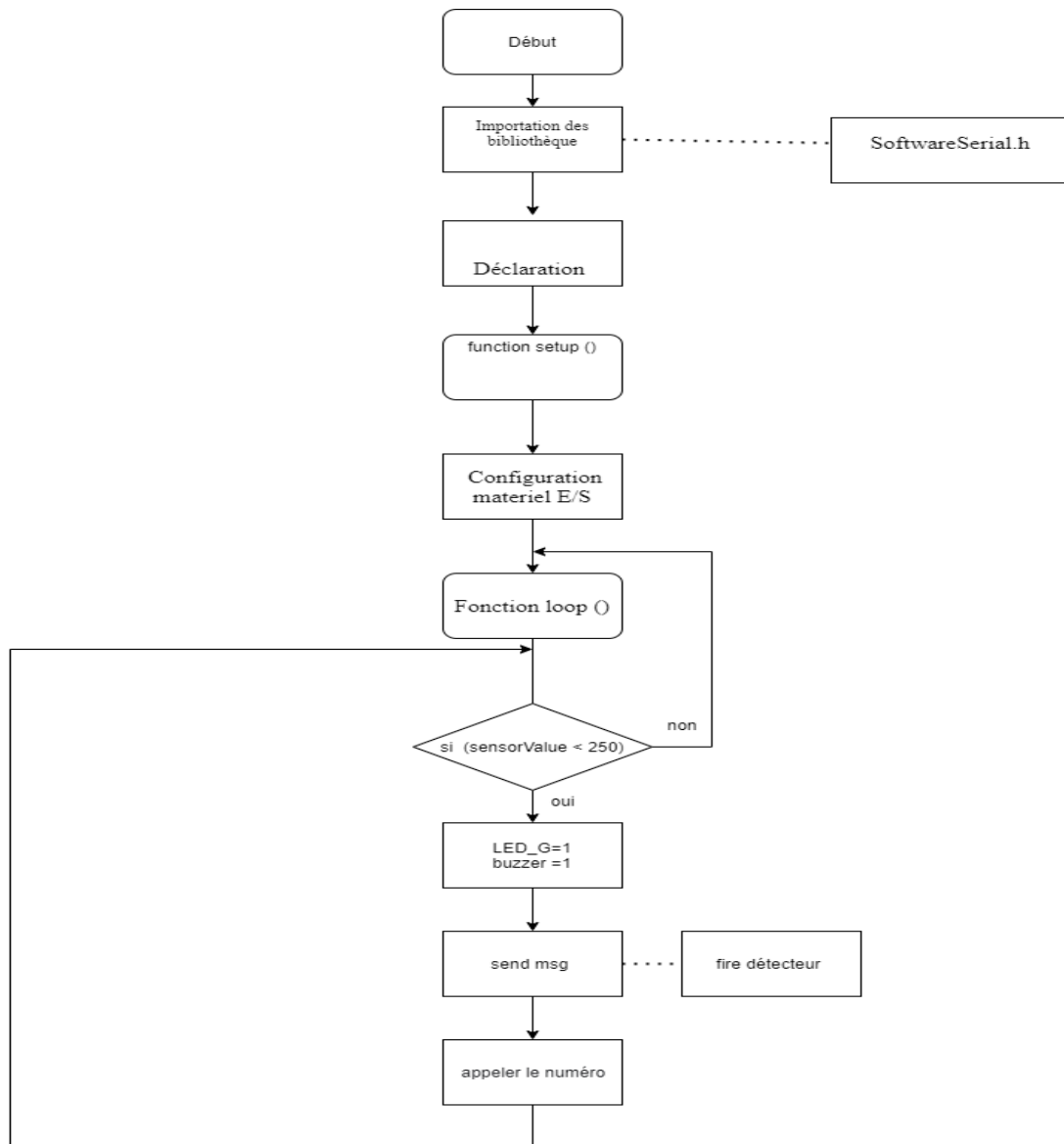


Figure IV.8: Organigramme de la fonction de détection flamme

Partie 3 :Détecteur de température et l'humidité :

En raison de la gravité de l'incendie et de la rapidité de sa propagation, et afin d'assurer l'efficacité de notre projet, nous avons ajouté un capteur de température et d'humidité. Ce capteur mesure la température et l'humidité. Il sera utilisé en remplacement du capteur de flamme. S'il ne détecte pas le feu, il enregistrera l'élévation de température, et si la température atteint une certaine limite, nous la déterminons selon les quatre saisons et le degré maximum dans cette zone. Dans le cas où il a enregistré une température supérieure à la température que nous avons définie, il enverra un SMS.

Le matériel que nous avons utilisé est le suivant :

CHAPITRE IV : réalisation pratique d'un système intelligent

- a- Carte Arduino .
- b- Capteur de température et l'humidité DHT 11 :
Ce capteur mesure la température et l'humidité en même temps
- c- LED rouge et résistance et buzzer.

Il est utilisé comme une alarme pour émettre des sons d'avertissement.

d- Carte GSM900a:

Cette carte est utilisée pour envoyer des SMS et des appels téléphoniques au cas où la température dépasse la limite prédéfinie .

e- Afficher LCD +2IC :

Nous l'utilisons pour afficher la température et l'humidité .

Le Câblage avec l'Arduino :

- Capteur de température et l'humidité DHT 11 :
GND → GND
VCC → 5V
data → gate (déclarer a pin 11).
- LED rouge et Arduino:
(+) → pin3
(-) → en série avec la résistance et GND de Arduino.
- buzzer et Arduino :
GND → GND
VCC → 5V
SIGNAL → pin4
- carte GSM900a et Arduino :
GND → GND
RX → pin8 (déclarer comme TX)
TX → pin7 (déclarer comme RX)
On alimentée la carte GSM900a a pile 9V .
- LCD et 2IC :
GND → GND
VCC → 5V
SDA → A4
SCL → A5

La figure IV.9 représente notre circuit de détecteur de température et l'humidité

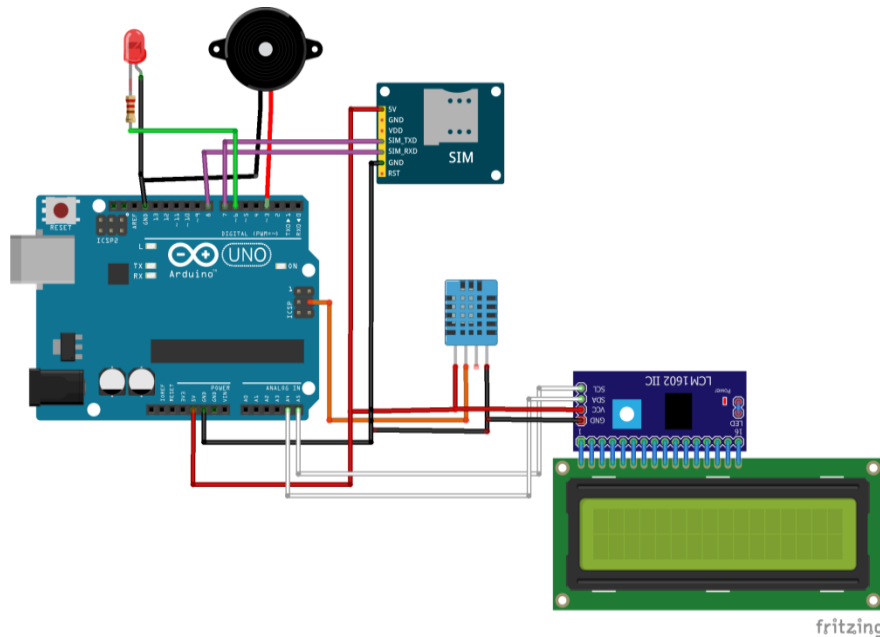


Figure IV.9:schéma de câblage de détecteur de température et l'humidité

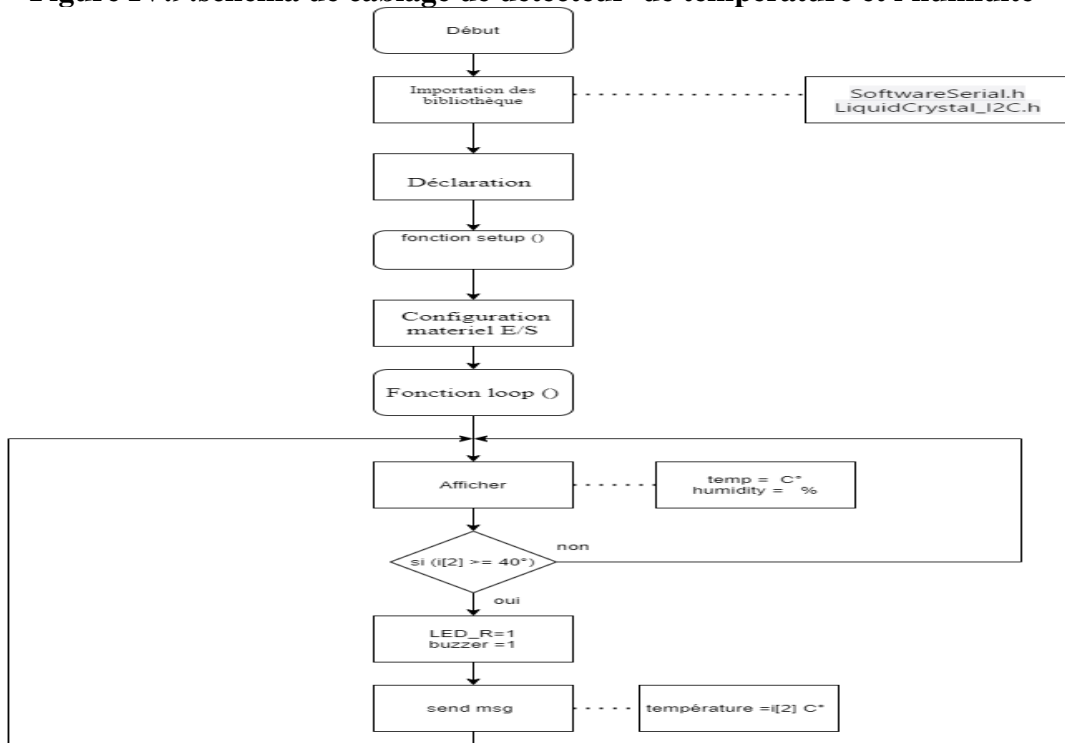


Figure IV.10:Organigramme de la fonction de détection température

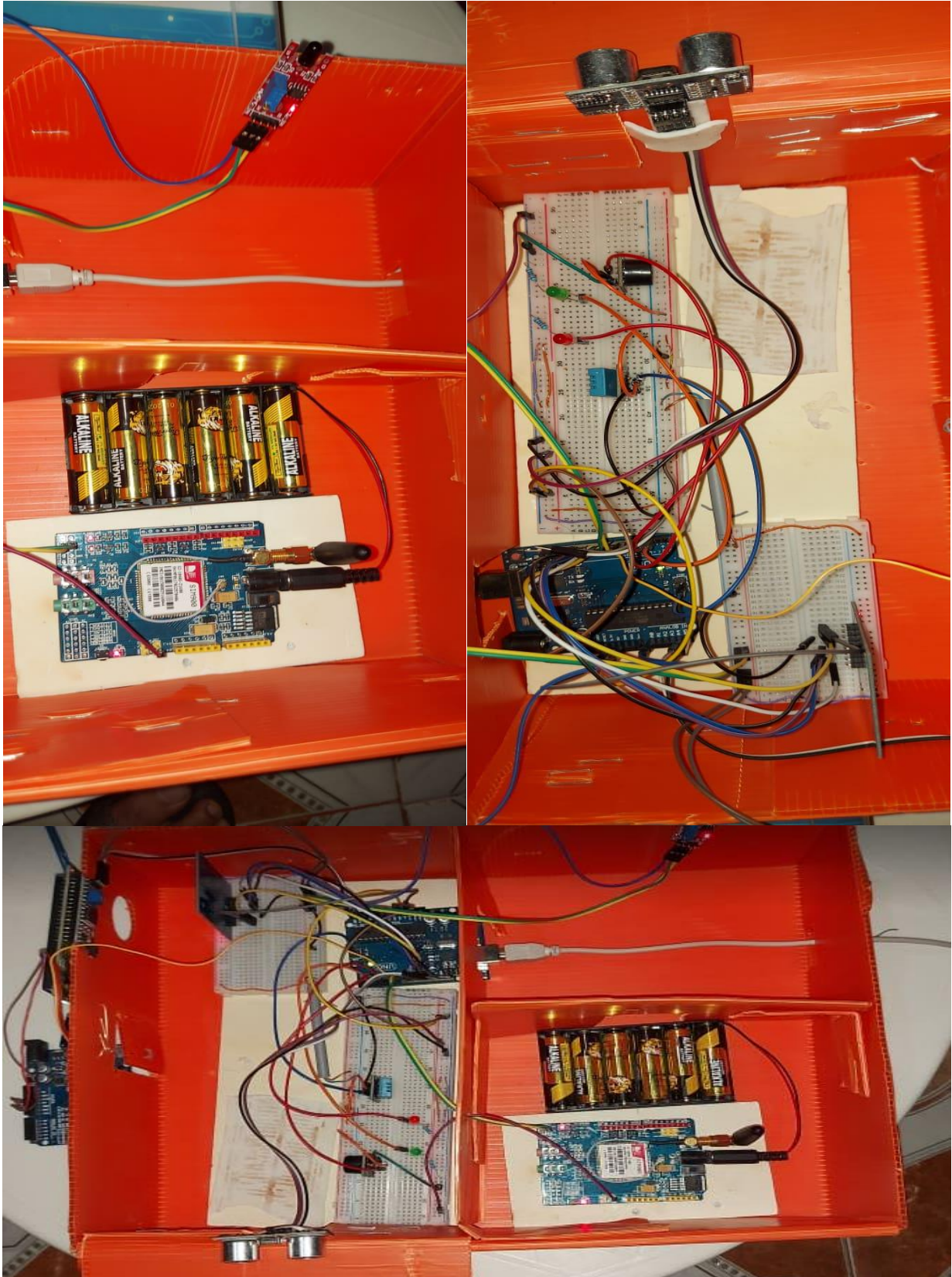


Figure IV.12: réaliser montage de projet de la détection incendie et la sécurité

2- Projet de RFID : ouverture de la porte

Avec le développement de la technologie, il est devenu possible d'ouvrir la porte à distance ou en appuyant sur un bouton. Dans notre projet, nous allons utiliser la technologie RFID pour ouvrir la porte automatiquement par carte, et nous pouvons l'ouvrir en envoyant des SMS, comme nous pouvons recevoir des informations et en cas d'ouverture de la porte, un message nous est envoyé pour nous dire que la porte a été ouverte.

Le matériel que nous avons utilisé est le suivant :

- a- Carte Arduino
- b- RFID MFRC522

Un émetteur-récepteur radio bidirectionnel, le lecteur, envoie un signal à l'étiquette et lit sa réponse.

Pour le système d'ouverture de porte RFID ; seule la personne ayant les informations correctes sur sa carte est autorisée à entrer. Le système RFID utilise :

Tags associés à l'objet à sélectionner, dans cet exemple nous avons un trousseau et un tag électromagnétique. Chaque balise a sa propre identité unique (UID).

- c- Carte GSM900a :

Pour envoyer et recevoir des messages SMS pour ouvrir la porte .

- d- LED (rouge et vert) et résistance et Buzzer.

Il est utilisé comme une alarme pour émettre des sons d'avertissement.

- e- Afficheur LCD +2IC

Cet afficheur nous permet de montrer la carte et l'état de la carte après l'avoir mémorisée, qu'elle soit acceptée ou refusée.

- f- Servomoteur

Lorsque la porte est ouverte, le moteur tourne à 90 degrés puis revient au même angle pour fermer la porte.

Le Câblage avec l'Arduino :

- LED rouge et Arduino :
 - (+) → pin3
 - (-) → en série avec la résistance et GND de Arduino.
- LED vert et Arduino :
 - (+) → pin5
 - (-) → en série avec la résistance et GND de Arduino.
- buzzer et Arduino :
 - GND → GND

CHAPITRE IV : réalisation pratique d'un système intelligent

VCC → 5V

SIGNAL → pin4

- carte GSM900a et Arduino :

GND → GND

RX → pin8 (déclarer comme TX)

TX → pin7 (déclarer comme RX)

On alimentée la carte GSM900a a pile 9V .

- LCD et 2IC :

GND → GND

VCC →5V

SDA →A4

SCL → A5

- MRFC522 avec l'Arduino :

SDA → pin10

SCK → pin13

MOSI → pin11

MISO → pin12

GND → GND

RST → pin9

3.3V → 3.3V

La figure IV.13 représente notre circuit de système intelligent

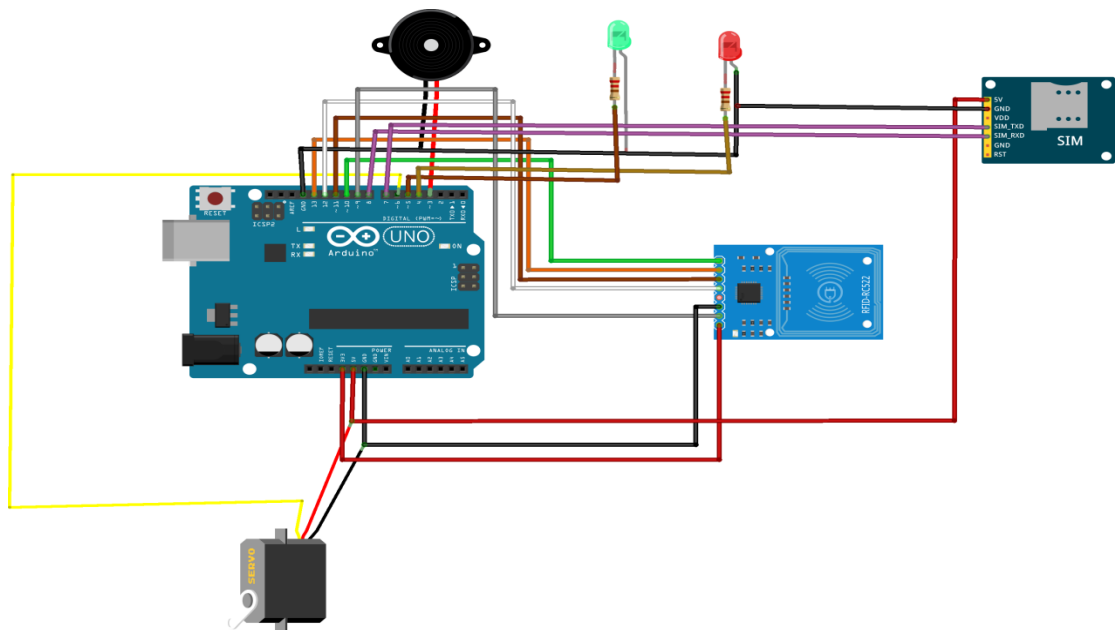


Figure IV.13: schéma de câblage de projet d'ouverture de porte

fritzing

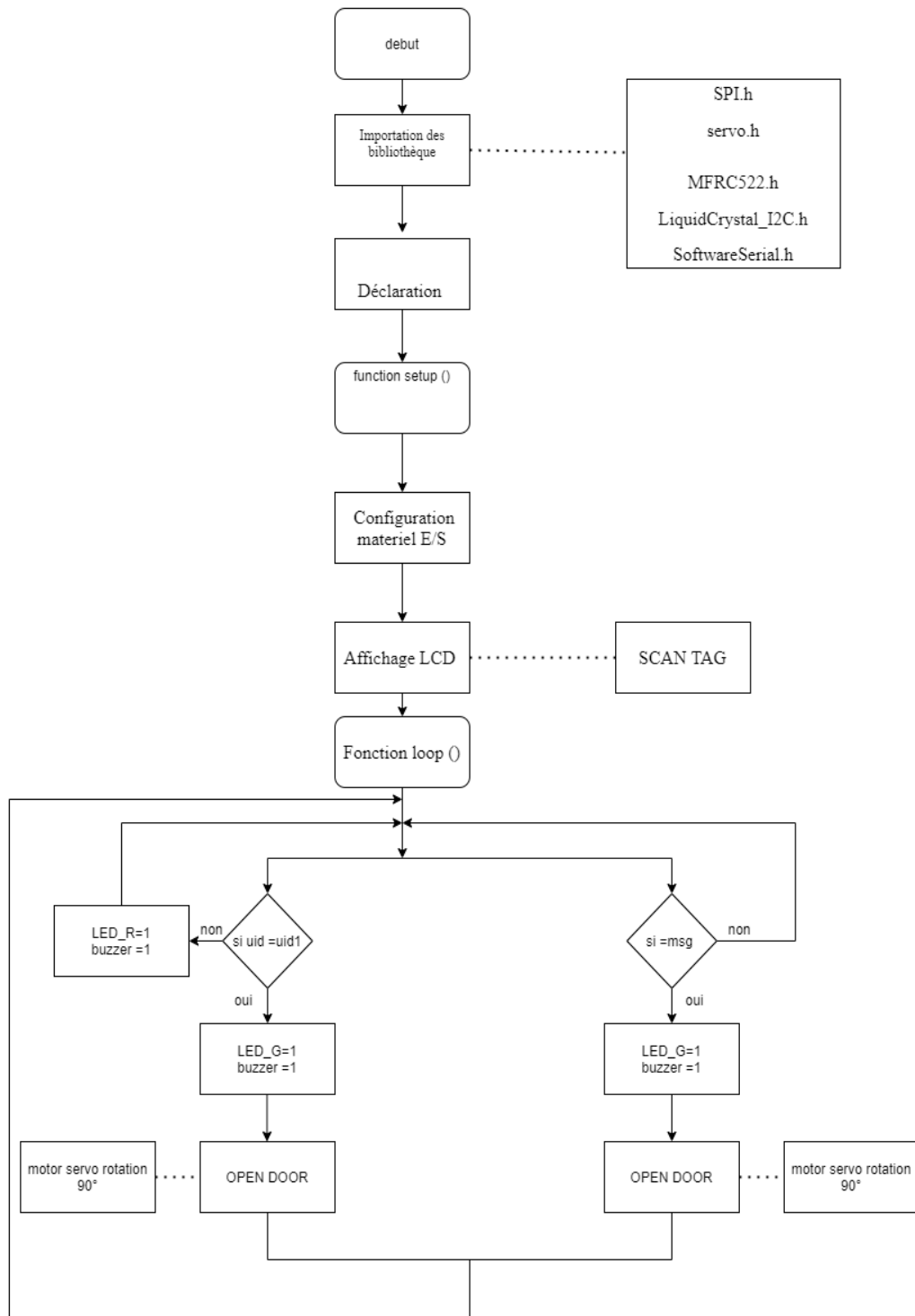


Figure IV.14: Organigramme de la fonction d'ouverture de porte

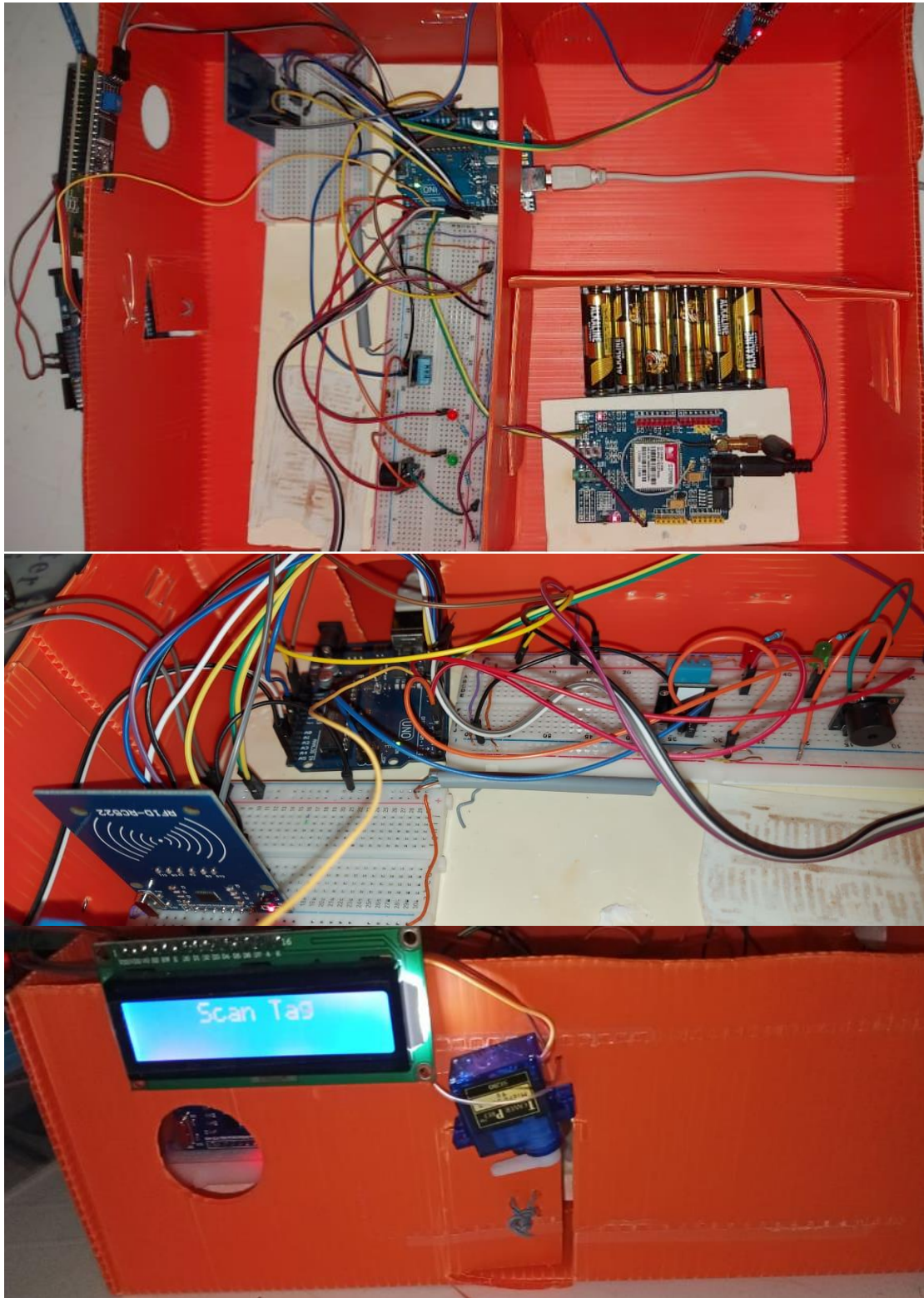


Figure IV.15: réaliser montage de projet d'ouverture de la porte

IV.5.Conclusion :

Dans ce dernier chapitre, nous avons expliqué les différentes étapes de la réalisation d'un système intelligent de détection et de sécurité incendie, nous avons commencé par quelques exemples démonstratifs du fonctionnement de la carte Arduino qui permet aux lecteurs de réaliser des mini-projets et en même temps une introduction à la réalisation de nos deux projets à savoir :

- Ouverture de la porte automatique par carte ou par SMS .
- Système intelligent de détection et de sécurité incendie .

La création d'une solution domotique pour le particulier s'est révélée être indispensable pour combler l'insatisfaction du public en quête de services et de fonctions facilitant sa vie quotidienne à domicile. Dans ce cadre nous avons réalisé un système capable de gérer et de prévenir en cas d'incendie ou en cas d'une flamme ; il surveille également l'état de la température et de l'humidité et nous faire parvenir par SMS à l'aide une carte GSM SIM900, ainsi l'ouverture et la fermeture de la porte à distance; à travers deux méthodes, soit par une carte, soit par SMS, D'autre part, nous avons programmé les composants nécessaires à l'aide d'ARDUINO IDE.

A la fin de ce projet, nous avons un aperçu complet du processus de réalisation d'un système domotique, des compétences et des connaissances qui y sont attachées.

Conclusion générale

CONCLUSION GENERALE

Conclusion générale :

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

Ce projet nous a fait découvrir un secteur que nous ne connaissions pas vraiment et qui nous a permis de nous amuser grâce à la manipulation du matériel, tout en acquérant de meilleures connaissances des applications de la domotique, ce qui pourrait nous être fortement utile pour notre vie professionnelle future. C'est dans ce cadre que se situe notre projet de fin d'études intitulé « réalisation pratique d'un système Smart, pour la détection d'incendie et la sécurité ». Ce projet a pour objectif de réaliser un système qui peut intégrer dans une maison intelligente permettant le pilotage, la commande et la surveillance des différents dispositifs disponibles dans cette dernière. Elle est capable de gérer l'ouverture et la fermeture des portes, de prévenir en cas d'incendie, aussi surveillé l'état de la température et l'humidité.

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 unique interface Web.
- Permettre à l'utilisateur de faire des modifications au niveau des fonctionnalités proposées selon son besoin.
- Ajouter d'autres fonctionnalités à notre système domotique tel que la gestion des multimédia, la fonctionnalité anti incendie .

Bibliographie

BIBLIOGRAPHIE

Référence :

- [1] Malha, Inguel. *Conception et réalisation d'un système Domotique par GSM*. Diss. Université Mouloud Mammeri, 2017.
- [2] Nassim, Bouakline, and Benchaba Aghiles. *Conception et réalisation d'une maison intelligente via raspberry Pi 3*. Diss. Université Mouloud Mammeri, 2018.
- [3] C. Tavernier, «Arduino applications avancées». Version Dunod.
- [4] <http://knx-automation.com/domotique/les-fonctions-domotiques-de-la-maison-intelligente/>
- [5] SIRLAN Technologies, "Les automatismes du bâtiment : La domotique Le maintien à domicile"
- [6] <https://fr.wikipedia.org/wiki/Microcontr%C3%B4leur>
- [7] <http://for-ge.blogspot.com/2015/07/microcontroleur.html>
- [8] Ferroudj Abdelghani, *Réalisation d'un système de télésurveillance médicale*, Université de Khenchela, 2017
- [9] GUENAOUA LILIA, "Commande en position du a MCC par Arduino", Université Badji Mokhtar Annaba, 2017
- [10] Aboubakr El Hammoumi, "Conception et réalisation d'un prototype d'une maison domotique intelligente My Smart Home ", Université Hassan 1er, 2016
- [11] le grand livre d'arduino
- [12] https://www.tutoriel-arduino.com/lcd_i2c_arduino
- [13] Hippolyte Weisslinger (olyte), Landrault (Eskimon) *Arduino : Premiers pas en informatique embarquée*, le Blog d'Eskimon Edition du 19 juin 2014
- [14] <https://lastminuteengineers.com/sim900-gsm-shield-arduino-tutorial/>
- [15] <https://arduinomodules.info/ky-026-flame-sensor-module/>
- [16] Diracca C. (1986). *Etude comparative de PID numériques auto-ajustables : applications à la commande d'un télépointeur*. Thèse de Doctorat, Nice.
- [17] TOURE Fatoumata Fousseyni & REZIK Kheira, "Contrôle et Commande d'un système à distance via GSM", Université de Mostaganem, 2019
- [18] <https://fritzing.org/>
- [19] Franklin, G.F. Powell, J.D. (1989). *Digital control of dynamic systems*. Addison-Wesley, USA.
- [20] K. Saber, "Modélisation et commande d'un mini-hélicoptère drone," Master's thesis, Université de SKIKDA, 2013.
- [21] Thiziri, Belaidi, and Haddad Naima. *Le développement d'une carte électronique pour le*

BIBLIOGRAPHIE

contrôle d'un système de sécurité dans une entreprise. Diss. Université Mouloud Mammeri, 2018

[22]P. Clerc“*Automatique Continue &Echantillonnée*”, Masson, Paris, 1996.

[23] http://projet.eu.org/pedago/sin/term/6-asservissement_PID.pdf

[24]<https://www.electronique-mixte.fr/pid-implementation-du-correcteur-avec-arduino-3/>

Annexes

ANNEXE A

Le programme d'Arduino (la tuto)

```
//https://www.electronique-mixte.fr/pid-implementation-du-correcteur-avec-arduino-3/
/*
 * Objectifs:
 * 1. Savoir Implémenter un correcteur PID Numérique (Modèle du 2nd
Ordre)
 * 2. Savoir régler les paramètres de son correcteur
 * 3. Savoir optimiser les performances de son correcteur
 * 4. Arduino & Problème de la Saturation (Divergence)
 */

#define Fn      5.00
#define Zeta    0.05 //0.70710678118
#define K       1.0
#define T_ms    2

#define A_step  10.0    // Amplitude
#define c_step  500     // Période = 2*c_step*T_ms

double Wn=2.0*PI*Fn;
double T_s=(double)T_ms/1000.0;

double x_nn=0.0;      // Consigne (entrée)
double y_n[2];        // "0" Non corrigé, "1": Corrigé
double eps_n[2];      // Erreur
double y_capt[2];     // Sortie du capteur
double y_corr[2];     // Sortie du correcteur

// Variables internes des systèmes
double x1[2], y1[3]; // Système Non Corrigé
double x2[2], y2[3]; // Système Corrigé

// Variables internes du correcteur
double x_c[2], y_c[2];

// Paramètres de l'échelon
unsigned long c=0; // Compteur (période)
bool Step=false;

void setup()
{
  // Port série de la réponse du système
  Serial.begin(19200);
}

void loop()
{
  // 1. La consigne (l'entrée) x(n) pour les deux systèmes
  c++; c=c%c_step;
  if(!c)
  {
    Step=!Step;
    c=0;
  }
}
```

Annexes

```
x_nn=A_step*(double)Step; // Réponse à un échelon x(n)=cte
//x_nn=(double)c;          // Réponse à une rampe x(n)=n

// 2. Sortie du capteur: Retour unitaire
y_capt[0]=y_n[0];
y_capt[1]=y_n[1];

// 3. Soustracteur: Calcul de l'erreur eps(n)
eps_n[0]=x_nn-y_capt[0];
eps_n[1]=x_nn-y_capt[1];

// 4.1 Correcteur
y_corr[0]=eps_n[0]; // Système non Corrigé

// 4.2 Correcteur PID (voir le lien en commentaire)
double k0=2.5;
double N_fil=250.0;
double z_PID=1.0;
double w_PID=0.7*Wn; //Wr=Wn*sqrt(1-Zeta*Zeta);
y_corr[1]= PID(eps_n[1], x_c, y_c, N_fil, z_PID, w_PID, k0, T_s);

// 5. Calcul de la sortie: Système non corrigé
y_n[0]=Sys2All(y_corr[0], x1, y1, Zeta, Wn, K, T_s);

// 5. Calcul de la sortie: Système corrigé
y_n[1]=Sys2All(y_corr[1], x2, y2, Zeta, Wn, K, T_s);

// Affichage des signaux
Serial.print(x_nn); Serial.print(",");
Serial.print(y_n[0]); Serial.print(",");
Serial.println(y_n[1]);

// Période d'échantillonnage
delay(T_ms);
}

double Sys2All(double x_nn, double *x, double *y, double zeta, double
wn, double k, double T)
{
// Paramètre du système
double a1=2.0*zeta/wn;
double a2=1.0/(wn*wn);

const double b0=(a1/(2.0*T))+(a2/(T*T));
const double b1=-2.0*a2/(T*T);
const double b2=(-1.0*a1/(2.0*T))+(a2/(T*T));
const double b[3]={b0,b1,b2};

// Variables de l'entrée et la sortie
double y_nn=0.0;

// Calcul de la nouvelle sortie
y_nn= -(y[0]*(1.0+b[1]))-(y[1]*b[2])+(k*x[0]); // y[1]: y(n-2), y[0]:
y(n-1)
y_nn/=b[0];
```


Annexes

```
x_cc[0]=x_nn;

// Renvoi du résultat
return y_nn;
}

double PID(double x_nn, double *x_cc, double *y_cc, double N_fil,
double z, double w, double k, double T)
{

double t1=1.0/(w*(z+sqrt(z*z-1.0)));
double t2=1.0/(w*(z-sqrt(z*z-1.0)));

double td=t1+t2;
double ti=t1*t2/td;

/*
Serial.print(t1);
Serial.print("\t");
Serial.print(t2);
Serial.print("\t");
Serial.print(td);
Serial.print("\t");
Serial.println(ti);
*/

double a1=2.0*z/w;
double a2=1.0/(w*w);
double b1=ti;
double b2=ti*t2/N_fil;

double b[3];
b[0]= a1/(2*T)+a2/(T*T);
b[1]= 1.0-2.0*a2/(T*T);
b[2]=-a1/(2*T)+a2/(T*T);

double c[3];
c[0]= b1/(2*T)+b2/(T*T);
c[1]= -2.0*b2/(T*T);
c[2]= -b1/(2*T)+b2/(T*T);

// Variables de l'entrée et la sortie
double y_nn=0.0;

// Calcul de la nouvelle sortie
y_nn=k*(b[0]*x_nn + b[1]*x_cc[0] + b[2]*x_cc[1]);
y_nn=y_nn-(c[1]*y_cc[0] + c[2]*y_cc[1]);
y_nn=y_nn/c[0];

// Mise à jour de la sortie
y_cc[1]=y_cc[0];
y_cc[0]=y_nn;

// Mise à jour de la sortie
x_cc[1]=x_cc[0];
```

Annexes

```
x_cc[0]=x_nn;  
  
// Renvoie du résultat  
return y_nn;  
}
```

ANNEXE B

Le programme de projet d'ouverte de porte :

```
#include <Servo.h>
#include <SPI.h>
#include <MFRC522.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
Servo myServo; //define servo name
LiquidCrystal_I2C lcd(0x27, 16, 2);
SoftwareSerial SIM900(7, 8);
String textForSMS;

#define SS_PIN 10
#define RST_PIN 9
#define LED_G 5 //define green LED pin
#define LED_R 4 //define red LED
#define BUZZER 4 //buzzer pin

MFRC522 mfrc522(SS_PIN, RST_PIN);
String uid1 = "23 F3 03 15";
String uid2 = "03 B9 20 16";
String NBR = "AT+CMGS=\"+213xxxxxxxxxx\"";
String MSG = "";

String incomingData; // for storing incoming serial data
String message = ""; // A String for storing the message
int relay_pin = 2; // Initialized a pin for relay module
void setup() {
  Serial.begin(9600);
  randomSeed(analogRead(0));
  SIM900.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init();
  lcd.init();
  lcd.backlight();

  delay(3000);
  lcd.clear();

  delay(1000);

  // set SMS mode to text mode
  SIM900.print("AT+CMGF=1\r");
  delay(100);

  // set gsm module to tp show the output on serial out
  SIM900.print("AT+CNMI=2,2,0,0,0\r");
  delay(100);
}

void loop() {
  myServo.attach(6); //servo pin
  myServo.write(0); //servo start position
  //Function for receiving sms
  receive_message();
}
```

Annexes

```
// if received command is to turn on relay
if(incomingData.indexOf("open door")>=0)
{
digitalWrite(relay_pin, HIGH);
tone(BUZZER, 500);
delay(2000);
noTone(BUZZER);
myServo.write(90);
delay(5000);
myServo.write(0);
delay(1000);
digitalWrite(relay_pin, LOW);
incomingData="off"; // Send a sms back to confirm that the relay is
turned on
}

lcd.clear();
lcd.setCursor(0,0);
lcd.print("    Scan Tag    ");
// Reset the loop if no new card present on the sensor/reader. This
saves the entire process when idle.
if ( ! mfrc522.PICC_IsNewCardPresent())
{
return;
}
// Select one of the cards
if ( ! mfrc522.PICC_ReadCardSerial())
{
return;
}
//Show UID on serial monitor
Serial.print("UID tag :");
String content= "";
byte letter;
for (byte i = 0; i < mfrc522.uid.size; i++)
{
Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
Serial.print(mfrc522.uid.uidByte[i], HEX);
content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : "
"));
content.concat(String(mfrc522.uid.uidByte[i], HEX));
}
lcd.clear();
Serial.println();
Serial.print("Message : ");
content.toUpperCase();
if (content.substring(1) == uid1) //change here the UID of the
card/cards that you want to give access
{
lcd.clear();
lcd.print("Authorized access");
delay(500);
Serial.println("UID 1");
delay(500);
digitalWrite(LED_G, HIGH);
tone(BUZZER, 500);
delay(1000);
noTone(BUZZER);
```

Annexes

```
myServo.write(180);
delay(5000);
myServo.write(0);
digitalWrite(LED_G, LOW);
delay(3000);

// for gsm
delay(1000);
MSG = "OPEN DOOR";
Send_SMS(NBR,MSG);
}

else {
Serial.println(" Access Denied");
lcd.setCursor(3,0);
lcd.clear();
lcd.print("Unknown");
lcd.setCursor(0,1);
tone(BUZZER, 500);
delay(1000);
noTone(BUZZER);
digitalWrite(LED_R, HIGH);
lcd.print("Access Denied");
delay(5000);

digitalWrite(LED_R, LOW);
lcd.clear();
}
}

void Send_SMS(String NBR,String MSG )
{
SIM900.begin(9600);

Serial.println("Initializing...");
delay(1000);

SIM900.println("AT"); //Handshaking with SIM900
updateSerial();

SIM900.println("AT+CMGF=1"); // Configuring TEXT mode
updateSerial();
SIM900.println(NBR);//change ZZ with country code and xxxxxxxxxxxx
with phone number to sms
updateSerial();
SIM900.print(MSG); //text content
updateSerial();
SIM900.write(26);

delay(1000);

lcd.setCursor(0, 1);
lcd.print("message sent");
delay(2000);
}

void updateSerial()
```

```
{
  delay(500);
  while (Serial.available())
  {
    SIM900.write(Serial.read()); //Forward what Serial received to
Software Serial Port
  }
  while(SIM900.available())
  {
    Serial.write(SIM900.read()); //Forward what Software Serial received
to Serial Port
  }
}
void receive_message()
{
  if (SIM900.available() > 0)
  {
    incomingData = SIM900.readString(); // Get the data from the serial
port.
    Serial.print(incomingData);
    delay(10);
  }
}
```

ANNEXE C

Le programme de projet de la détection incendie et la sécurité :

```
//infinite Xpro
// firstly need to add i2c library

#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <SoftwareSerial.h>
//#include <string>

#define LED_R 3 //define red LED
#define BUZZER 4 //buzzer pin

LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a
16 chars and 2 line display
SoftwareSerial SIM900(7, 8);
SoftwareSerial mySerial(7, 8);
byte degree_symbol[8] =
    {
        0b00111,
        0b00101,
        0b00111,
        0b00000,
        0b00000,
        0b00000,
        0b00000,
        0b00000
    };

int gate=11;
volatile unsigned long duration=0;
unsigned char i[5];
unsigned int j[40];
unsigned char value=0;
unsigned answer=0;
int z=0;
int b=1;
String NBR = "AT+CMGS=\"+213XXXXXXXXXX\"";
String NBR1 = "ATD+ +213XXXXXXXXXX;";
String MSG = "";
String temp ;

const int trigPin = 9;
const int echoPin = 12;
const int ledPin = 3;
const int buzzer = 4;

// defines variables
long duration1;
int distance;
int safetyDistance;

int sensorPin = A0; // select the input pin for the LDR

int sensorValue = 0;

void setup()
```

Annexes

```
{
  Serial.begin(9600);

  pinMode(ledPin, OUTPUT);

pinMode(buzzer,OUTPUT);
  //randomSeed(analogRead(0));
  SIM900.begin(9600);
  // SPI.begin();

  lcd.init(); // initialize the lcd
  lcd.init();
  lcd.backlight();
  lcd.print("Temp = ");
  lcd.setCursor(0,1);
  lcd.print("Humidity = ");
  lcd.createChar(1, degree_symbol);
  lcd.setCursor(9,0);
  lcd.write(1);
  lcd.print("C");
  lcd.setCursor(13,1);
  lcd.print("%");
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
pinMode(buzzer, OUTPUT);
pinMode(ledPin, OUTPUT);

}

void loop()
{

  sensorValue = analogRead(sensorPin);

  if (sensorValue < 250)
  {

    MSG ="Fire Detected" ;

    Send_SMS(NBR,MSG);

digitalWrite(ledPin,HIGH);

tone(buzzer,1000,5000);
delay(5000);
digitalWrite(ledPin,LOW);

digitalWrite(buzzer,LOW);

}

digitalWrite(ledPin,LOW);
```

Annexes

```
digitalWrite(buzzer,LOW);

delay(sensorValue);

//*****
// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in
microseconds
duration1 = pulseIn(echoPin, HIGH);

// Calculating the distance
distance= duration1*0.034/2;

safetyDistance = distance;
if (safetyDistance <= 50){
  tone(buzzer,1000,3000);
  digitalWrite(ledPin, HIGH);
  delay(5000);
  digitalWrite(buzzer, LOW);
  digitalWrite(ledPin, LOW);
  MSG ="distance Cm" ;

  // Send_SMS (NBR,MSG);
  Call (NBR1);

  //delay(30000);
}
else{
  digitalWrite(buzzer, LOW);
  digitalWrite(ledPin, LOW);
}

// Prints the distance on the Serial Monitor

Serial.print("Distance: ");
Serial.println(distance);

//*****

delay(1000);
// while(true)
//{
  delay(1000);
  pinMode(gate,OUTPUT);
  digitalWrite(gate,LOW);
  delay(20);
```

Annexes

```
digitalWrite(gate,HIGH);
pinMode(gate,INPUT_PULLUP);//by default it will become high due to
internal pull up
// delayMicroseconds(40);

duration=pulseIn(gate, LOW);
if(duration <= 84 && duration >= 72)
{
    while(1)
    {
        duration=pulseIn(gate, HIGH);

        if(duration <= 26 && duration >= 20){
            value=0;}

        else if(duration <= 74 && duration >= 65){
            value=1;}

        else if(z==40){
            break;}

        i[z/8]=value<<(7- (z%8));
        j[z]=value;
        z++;
    }
}
answer=i[0]+i[1]+i[2]+i[3];

if(answer==i[4] && answer!=0)
{
    lcd.setCursor(7,0);
    lcd.print(i[2]);
    lcd.setCursor(11,1);
    lcd.print(i[0]);
    if (i[2] >= 40 )
    {
        temp =i[2] ;
        MSG ="temp= " + temp + " C" ;

        // Send_SMS(NBR,MSG);
        tone(BUZZER, 500);
        digitalWrite(ledPin, HIGH);
        delay(5000);
        noTone(BUZZER);
        digitalWrite(ledPin, LOW);
        delay(5000);

//break;
}

}

//}
z=0;
i[0]=i[1]=i[2]=i[3]=i[4]=0;
```

Annexes

```
}
//*****

void Send_SMS(String NBR,String MSG )
{
  SIM900.begin(9600);

  Serial.println("Initializing...");
  delay(1000);

  SIM900.println("AT"); //Handshaking with SIM900
  updateSerial();

  SIM900.println("AT+CMGF=1"); // Configuring TEXT mode
  updateSerial();
  SIM900.println(NBR);//change ZZ with country code and xxxxxxxxxxxx
with phone number to sms
  updateSerial();
  SIM900.print(MSG); //text content
  updateSerial();
  SIM900.write(26);
}

void Call(String NBR1)
{
  //Begin serial communication with Arduino and Arduino IDE (Serial
Monitor)
  Serial.begin(9600);

  //Begin serial communication with Arduino and SIM900
  mySerial.begin(9600);

  Serial.println("Initializing...");
  delay(1000);

  mySerial.println("AT"); //Handshaking with SIM900
  updateSerial1();

  mySerial.println(NBR1); // change ZZ with country code and
xxxxxxxxxxx with phone number to dial
  updateSerial1();
  delay(20000); // wait for 20 seconds...
  mySerial.println("ATH"); //hang up
  updateSerial1();
}

void updateSerial1()
{
  delay(500);
  while (Serial.available())
  {
    mySerial.write(Serial.read()); //Forward what Serial received to
Software Serial Port
  }
  while(mySerial.available())
  {
    Serial.write(mySerial.read()); //Forward what Software Serial
```

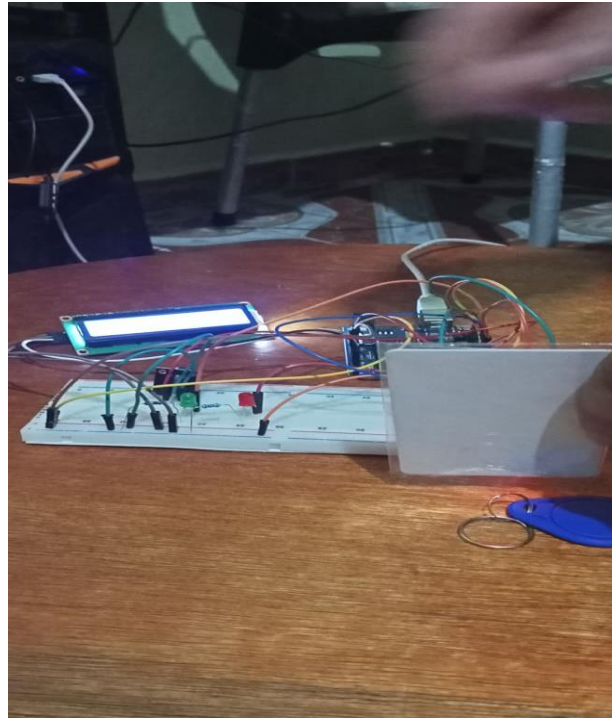
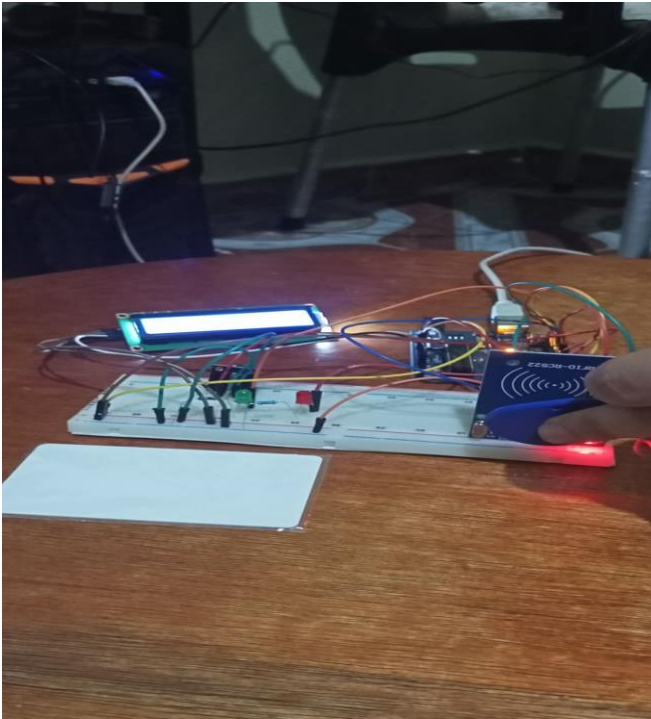
Annexes

```
received to Serial Port
}
}

void updateSerial()
{
  delay(500);
  while (Serial.available())
  {
    SIM900.write(Serial.read()); //Forward what Serial received to
Software Serial Port
  }
  while(SIM900.available())
  {
    Serial.write(SIM900.read()); //Forward what Software Serial received
to Serial Port
  }
}
```

ANNEXE D

Quelques photos des projets réaliser



Annexes

