

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud Mammeri de Tizi-Ouzou



Faculté de Génie Electrique et d'Informatique
Département d'Electronique

**Mémoire de Fin d'Etude
de MASTER ACADEMIQUE**
domaine : Sciences et technologies
Filière : Génie électrique
Spécialité : Instrumentation

Présenté par

Belkadi Hayat

Thème

**conception et réalisation d'un système
d'alarme intelligent**

Mémoire soutenu publiquement le /07/2018 devant le jury composé de :

M Prénom NOM

Grade, lieu d'exercice,

Président

M, M .LAZRI

Maitre de conférences A , UMMTO,

Encadreur

M Prénom NOM

Grade, lieu d'exercice ,

Examineur

Remerciements

En premier, je remercie le bon dieu le tout puissant de m'avoir donné la santé, le courage et patience pour être ce que je suis aujourd'hui et pour mener à terme ce modeste travail ;

Je tiens à exprimer ma profonde gratitude à mon promoteur M^r lazri maitre de conférence 'A' au département de d'Electronique à l'Université MOULOUD MAMMERRI de Tizi Ouzou pour ses précieuses orientations, son aide et ses conseils et son souci de réaliser ce projet de fin d'étude ;

Je voudrais aussi témoigner ma reconnaissance et exprimer toute ma gratitude à nos enseignants qui ont participé pour une grande part dans notre formation .

Je tiens à remercier mes ami(e)s pour me donner le courage, le soutien moral pour mener ce travail.

Enfin, j'adresse un grand merci à ma famille et à toute personne qui a contribué de près ou de loin à la réalisation de ce mémoire et à tous ceux qui m'ont apporté leur aide.



Dédicace

*Je dédie ce modeste travail a mes très chers parents pour leurs soutien moral et matériel que
dieu tout puissant me les gardent*

A mes très chers frères

A mes très cheres sœurs

A tout mes amis

A tous ceux et celles qui m'aiment.....



Liste des figures

Figure 1: Domotique par GSM.....	4
Figure 2: Buzzer Piézoélectrique	4
Figure 3: diode électroluminescente	5
Figure 4: chaine d'action d'un capteur.....	6
Figure 5: la chaine de mesure d'un capteur.....	7
Figure 6:le schéma principale d'un capteur industriel	8
Figure 7: zone de surveillance.....	11
Figure 8:Capteur de mouvement	12
Figure 9 la zone de détection.....	12
Figure 10: Les zones importantes du capteur.....	13
Figure 11 : Architecture d'un module GSM	15
Figure 12:le schéma bloc de système	16
Figure 13 : module GSM SIM900.....	18
Figure 1: Domotique par GSM.....	4
Figure 2: Buzzer Piézoélectrique	4
Figure 3: diode électroluminescente	5
Figure 4: chaine de mesure d'un capteur	6
Figure 5: schéma principal d'un capteur industriel.....	7
Figure 6: Chaine de mesure d'un capteur.....	9
Figure 7: zone de surveillance.....	11
Figure 8:Capteur de mouvement	12
Figure 9 la zone de détection.....	12
Figure 10: Les zones importantes du capteur.....	13
Figure 11 : Architecture d'un module GSM	16

Liste des figures

Figure 12:le schéma bloc de système.....	17
Figure 13 : module GSM SIM900.....	17
Figure 14 : la carte Arduino UNO.....	21
Figure 15 : Description de la carte ARDUINO UNO	22
Figure 16: microcontrôleur ATMEGA 328	23
Figure 17: L'architecteur interne de Microcontrôleur ATMéga 328	23
Figure 18 :système à microprocesseur	25
Figure 19 : entrée / sortie numérique	26
Figure 20: les entrées analogiques	27
Figure 21: Convertisseur analogique numérique	28
Figure 22: graphe de conversion	28
Figure 23: Schéma Electrique de la carte ARDUINO UNO.....	30
Figure 24: les fonctions du programme.....	32
Figure 25: structure de programme	33
Figure 26: menu fichier	34
Figure 27: le menu Edition.....	36
Figure 28: menu croquis.....	37
Figure 29 : menu outils.....	39
Figure 30 : Branchement des différents éléments	43
Figure 31: branchements des éléments sur ISIS porteuse	43
Figure 32: Branchement de capteur PIR	44
Figure 33:le branchement de GSM à la carte Arduino.....	45
Figure 34: le branchement de buzzer à la carte Arduino.....	45
Figure 35: branchement de la led à la carte Arduino	46
Figure 36:(câble USB fiche A fiche B).....	47
figure 37:traduction d'un langage de programmation en langage machine.....	47

Liste des figures

Figure 38: interface de l'environnement de développement Arduino IDE.....	48
Figure 39: choix de la carte ARDUINO.....	49
Figure 40: choix de port de connexion de la carte	50
Figure 41: représente les icones de l'IDE d'Arduino.....	50
Figure 42 : interface de compilation et téléchargement	51
Figure 43: fenêtre sur de logiciel lors de détection d'une présence	52
Figure 44: message envoyé	53
Figure 45: fenêtre de logiciel en absence de mouvement	53

Table de matières

Introduction	1
--------------------	---

Chapitre I: les éléments de système de sécurité domotique

I.1.Préambule	3
I.2Les application de la domotique	3
I.3.Différents éléments du système	4
I.3.1.Buzzer	4
I.3.2.Diodes (LED)	5
I.3.3.Capteurs	6
I.3.3.1.Chaine de mesure	7
I.3.3.2.Classification des capteurs	9
I.3.3.3Différents types des capteurs	10
I.3.3.4.Capteur de mouvement ,	11
I.3.4.GSM	14
I.3.4.1.Historique.....	14
I.3.4.2.Couverture GSM dans le monde.....	14
I.3.4.4.L'architecture de réseau GSM	15
I.3.4.4.Module GSM ou modem GSM.	16
I.3.4.4.1Entrées /sorties de module GSM.....	18
I.3.4.5.Commande et surveillance pas GSM.....	19
I.4.Discussion	19

Chapitre II : Généralités sur la carte Arduino

II.1.Préambule	20
II.2.Historique	20
II.3.Définition de la carte	20.
II.3.1.caractéristiques de la carte Arduino UNO	21
II.3.2.Description de la carte Arduino.....	22
II.3.3.Présentation matérielle de la plateforme Arduino	22
II.3.3.1.Microcontrôleur	23
II.3.3.2.Entrés /sorties de la carte	26
II.3.3.2.1.Les entrées /sorties numérique	26

Table de matières

II.3.3.2.2.Entrés analogique	27
II.3.4.Alimentation de la carte Arduino	29
II.4.Partie logicielle	31
II.4.1.F onctions de langage Arduino	31
II.4.2. Structure de programme	32
II.4.3.Description des déférents menus	34
II.4.3.1.Menu fichier	34
II.4.3.2.Menu Edition	36
II.4.3.3.Menu croquis	37
II.4.3.4.Menu outil	39
II.4.4.Syntaxe du langage Arduino.....	40
II.5.Discussion	41

Chapitre III :La réalisation pratique

III.1.Préambule	42
III.2.Réalisation matérielle	42
III.2.1.Composition d'un système d'alarme	42
III.2.1.1.Bbranchement du capteur PIR à la carte Arduino.....	44
III.2.1.2.Branchement du module GSM.....	44
III.2.1.3.Branchement de buzzer à la carte Arduino	45
III.2.1.4.Branchement de la LED à la carte Arduino	46
III.3.Réalisation logicielle	46
III.3.1.Programmation de l'Arduino.....	46
III.3.2.Présentation du logiciel	48
III.3.3.Choix de la carte que l'on va programmer	49
III.3.4.Choix de port de connexion de la carte	49
III.4.Tests de l'application	52
III.5.Discussion	54
Conclusion	55

Introduction générale

Introduction générale

De nos jours, la domotique est omniprésente dans notre quotidien grâce au développement technologique dans plusieurs domaines tels que l'électronique, l'informatique, la physique, internet...

Ce concept de domotique a permis d'ajouter de l'intelligence à notre vie quotidienne, d'où l'orientation vers la maison intelligente. Cette dernière consiste en l'automatisation et la centralisation des différents objets de la maison afin d'assurer le contrôle, la surveillance, et aider des personnes âgées. Une maison intelligente consiste à placer des objets ou capteurs à l'intérieur et à l'extérieur pour constituer une base d'informations (température interne et externe, luminosité, capteur de gaz, force du vent, télésurveillance ...), et les centralisées vers un système d'affichage et enregistrement permettant à l'utilisateur de vérifier des différentes données.[1]

Leur but est d'améliorer le confort et la sécurité en préservant l'environnement par une gestion centralisée, simple et intuitive, qu'il s'agisse de la gestion du bâtiment (chauffage, climatisation, éclairage, sécurité...) ou de divertissement (audio, vidéo, cinéma).

La nécessité de protéger son domicile n'est pas nouvelle, depuis plusieurs siècle déjà, la population de monde entier ont développé des systèmes de télésurveillance destinés à prévenir toute intrusion ou tentative d'effraction. Les progrès informatiques ont eu leur conséquence aussi sur l'évolution des alarmes. Elles se pilotent aujourd'hui à distance à partir d'un Smartphone .

Dans ce contexte nous nous sommes intéressés à la réalisation d'un système d'alarme intelligent permettant de surveiller une maison à distance en temps réel. Une application Arduino pilote un capteur infrarouge et un module GSM. Ce dernier permet d'envoyer des notifications d'alerte afin d'avertir la présence d'éventuelles intrusions dans le lieu sécurisé.

Introduction générale

A fin de mener à bien notre projet, nous avons réparti le contenu de ce mémoire en trois chapitres :

Dans le premier chapitre, nous présentons les applications de la domotique, les éléments constituant le système d'alarme qui sont le capteur infrarouge, le module GSM et leurs caractéristiques.

Dans le deuxième chapitre, nous présentons le model de la carte Arduino qu'on a utilisé (UNO), ces caractéristiques ainsi les deux parties logiciel et matériel.

Le dernier chapitre sera consacré à la réalisation d'une alarme à base d'une carte d'acquisition Arduino.

Nous terminons par une conclusion générale qui résume intérêt de notre projet.

I.1. Préambule

La domotique est l'ensemble des technologies d'électronique de l'information et des télécommunications utilisées dans les domiciles qui vise à assurer des fonctions de sécurité, de confort de gestion d'énergies de communication qu'on peut trouver dans une maison . (Chauffage, volets roulants, porte de garage, portail d'entreprises électriques, etc.)

L'objectif de ce chapitre est de présenter des généralités sur les systèmes de sécurité domotique ainsi que leur constituant dans un premier temps. Dans un second temps, nous montrons l'utilisation de module GSM dans ces systèmes.

I.2. Les applications de la domotique

Comme indiqué précédemment, la domotique permet d'assurer les fonctions suivantes :

Pour le confort, il s'agit de :

- Gestion intelligente de l'éclairage (allumage et extinction)
- Commande à distance par Smartphone
- Commande de volets roulant, portails

Pour la sécurité, elle consiste en :

- Intégration d'alarme d'intrusion.
- Détection de fuites d'eau et fuite de gaz.
- Intégration de dispositif de coupure automatique.
- Introduction des dispositifs d'indentification (RFID).

Pour l'économie d'énergie, la domotique permet la :

- Détection de fuite d'eaux.
- Gestion intelligente de chauffage (programmation horaire, température).
- Gestion intelligente de l'éclairage (détection de présence, variation d'intensité lumineuse).



Figure 1: Domotique par GSM

I.3. Différents éléments du système

I.3.1 Buzzer

Un Buzzer est un élément électromécaniques ou électronique qui produit un son lorsque on lui applique une tensions dans notre cas il s'agit d'un buzzer électromécanique, il nécessite une tension continue comprise entre 3v et 28 v, il possède deux petite pattes de fixation



Figure 2: Buzzer Piézoélectrique

a) Les caractéristiques d'un Buzzer

- fréquence d'oscillations entre 2 et 4 kHz.
- tension d'alimentation de 3 à 30V
- intensité consommée de 2 à 12mA en général, plus de 50mA pour certains.
- pression sonore en général de 70 à 90 dB.

I.3.2. Diodes (DED)

Une diode électroluminescente est un dispositif optoélectronique capable d'émettre de la lumière lorsqu'il est parcouru par un courant électrique. La fonction "générique" d'une diode est de laisser passer le courant dans un sens, nous disons qu'elle est conductrice (dans le sens passant ou sens direct) et de bloquer le courant dans l'autre sens.

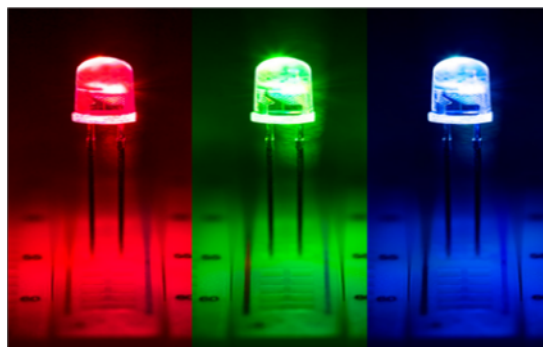
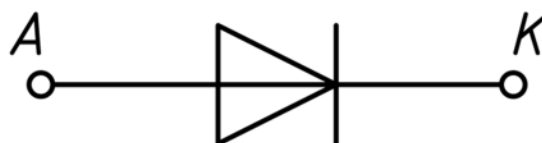


Figure 3: diode électroluminescente

- **Symbole**

Sur un schéma électronique, chaque composant est repéré par un symbole qui lui est propre.

Celui de la diode est celui-ci :



L'un des éléments essentiels des systèmes de sécurité domotique est le capteur. Plusieurs types de capteurs peuvent être utilisés dans la même application. Nous présentons dans ce qui suit une description détaillée sur le fonctionnement des capteurs.

I.3.3. Capteurs

Un capteur est un dispositif transformant une grandeur physique analogique (pression, température) en un signal analogique rendu transmissible et exploitable par un système de conditionnement (courant électrique, radiation, lumineuse, radiofréquences).le capteur est la partie d'une chaine de mesure qui se trouve au contact direct du mesurande.

Dans la majorité des cas, le signal de sortie est électrique en raison de la facilité de transmission de l'information.

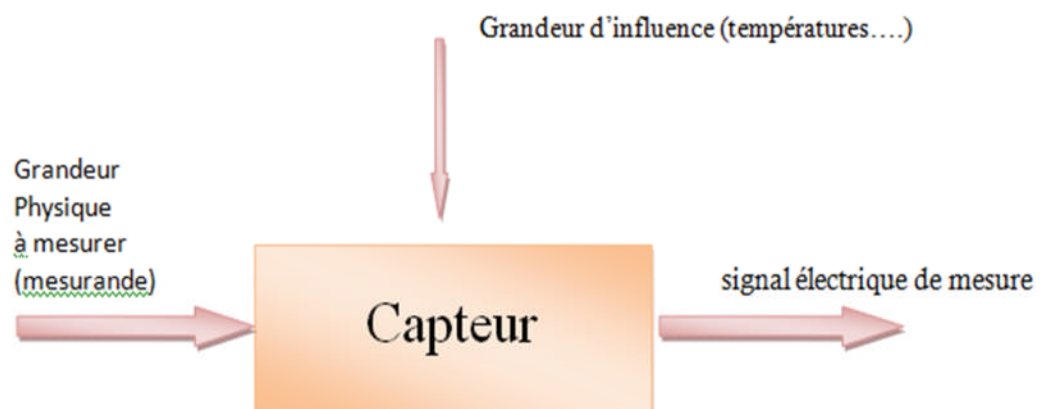


Figure 4: Chaîne d'action d'un capteur

I.3.3.1. Chaîne de mesure

Généralement, le signal de sortie de capteur n'est pas directement utilisable. On appelle chaîne de mesure l'ensemble des circuits ou appareils qui amplifient, adaptent, convertissent, digitalisent le signal avant sa lecture sur le support de sortie.

Pour obtenir une image d'une grandeur physique, la chaîne de mesure peut faire intervenir plusieurs phénomènes différents par exemple la mesure d'un débit peut se faire en plusieurs étapes :

- Transformation du débit en une pression différentielle.
- Transformation de la pression différentielle en la déformation mécanique d'une membrane.
- Transformation de déformation mécanique en une grandeur électrique (à l'aide d'un élément piézoélectrique) via un circuit électrique associé.

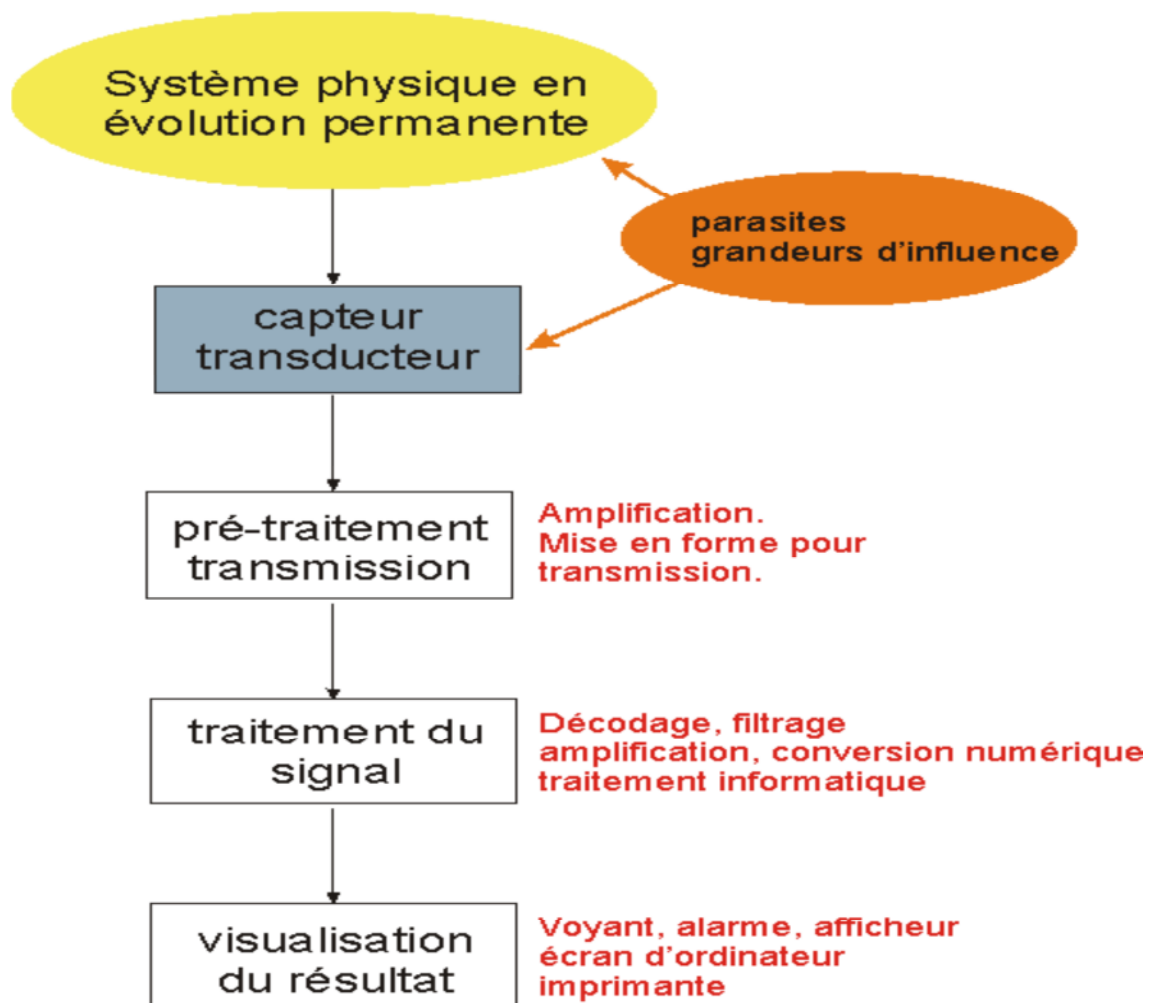


Figure 5: Chaîne de mesure d'un capteur

a) Caractéristiques des capteurs

- ✓ **Etendue de mesure** : c'est la différence entre le plus petit signal détecté et le plus grand perceptible sans risque de destruction pour le capteur.
- ✓ **Résolution** : Plus petite variation de grandeur mesurable par le capteur.
- ✓ **Sensibilité** : c'est la plus petite variation d'une grandeur physique que peut détecter un capteur.
- ✓ **Rapidité** : c'est le temps de réaction d'un capteur entre la variation de la grandeur physique qu'il mesure et l'instant où l'information prise en compte par la partie commande.
- ✓ **Linéarité** : représente l'écart de sensibilité sur l'étendue de mesure sa reproductibilité.
- ✓ **La bande passante** : est un intervalle de fréquences pour lesquelles l'amplitude de la réponse d'un système correspond à un niveau de référence.
- ✓

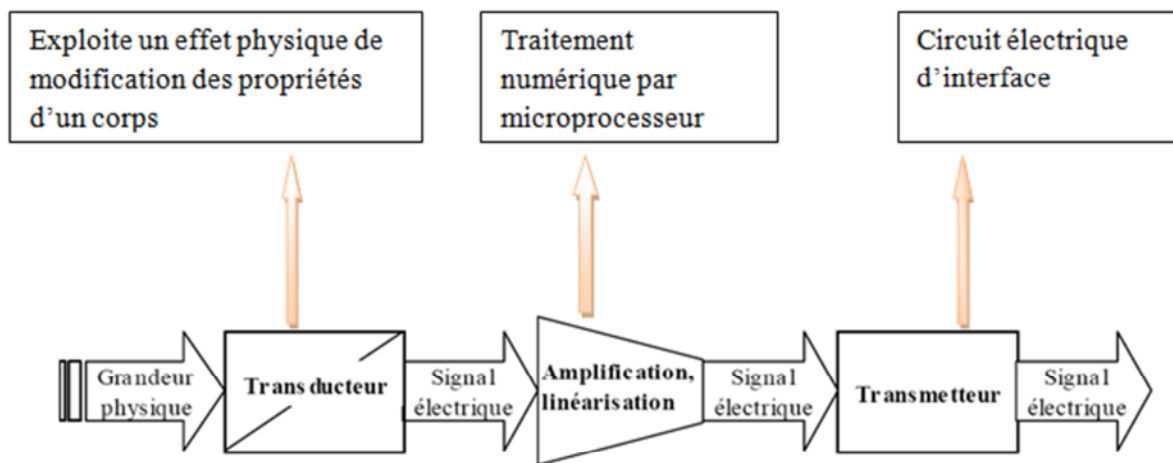


Figure 6: Schéma principal d'un capteur industriel

I.3.3.2. Classification des capteurs

Les capteurs sont classés en deux catégories selon les phénomènes physiques mis en jeux, capteurs actifs et capteurs passifs.

- **Capteur actif**

On parle de capteur actif lorsque le phénomène physique qui est utilisé pour la détermination du mesurande effectue directement la transformation en grandeur électrique.

- ❖ **Effet thermoélectrique (ou effet Seebeck) :**

Un circuit formé de deux conducteurs de nature chimique différente, dont les jonctions sont à des températures T_1 et T_2 , est le siège d'une force électromotrice d'origine thermique (T_1, T_2).

- ❖ **Effet piézo-électrique :**

L'application d'une contrainte mécanique à certains matériaux dits piézo-électriques (le quartz par exemple) entraîne l'apparition d'une déformation et d'une même charge électrique de signe différent sur les faces opposées.

- ❖ **Effet d'induction électromagnétique :**

La variation du flux d'induction magnétique dans un circuit électrique induit une tension électrique (détection de passage d'un objet métallique).

- **Capteur passif**

Il s'agit généralement d'impédances (résistance, inductance, capacité) dont l'un des paramètres déterminants est sensible à la grandeur mesurée. La variation d'impédance résulte :

- d'une variation de dimension du capteur (capteurs de position, potentiomètre, inductance à noyaux mobile, condensateur à armature mobile).
- d'une déformation résultant d'une force ou d'une grandeur s'y ramenant (pression, accélération). Exemples : armature de condensateur soumise à une différence de pression, jauge d'extensomètre liée à une structure déformable.

I.3.3.3 Différents types des capteurs

- **capteurs 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 le capteur de type mécanique et ceux de proximité (cellules, inductifs ou capacitifs) .Il sont constitués :

- D'un élément sensible mécanique ou électrique.
- D'un ou plusieurs contacts de type normalement fermé ou de type normalement ouvert.

- **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 valeur continue, le signal des capteurs analogique peut être de type :

- Sortie tension.
- Sortie courant.
- Règle graduée cadran, jauge (avec une aiguille ou un fluide) .

- **Capteur numérique**

La sortie est une séquence d'état logique qui, en se suivant forment un nombre .Elle peut prendre une infinité de valeurs discrète .le signal numérique peut être du type :

- Train d'impulsions, avec un nombre précis d'impulsion ou avec une fréquence précise.
- Code numérique binaire.
- Bus de terrain.

I.3.3.4. Capteur de mouvement

Un capteur PIR est un capteur qui mesure les radiations infrarouges (IR) émises par les objets se trouvant dans son champ de vision. Il est généralement utilisé comme détecteur de mouvement ou de proximité. [2]

Les capteurs PIR détectent les radiations infrarouges émises par les objets autour de lui. Or la plupart des objets émettent des IR suivant leur température. Ces capteurs sont dits passifs, car ils n'émettent pas de radiation (contrairement aux barrières à infrarouge). Les infrarouges sont invisibles par l'œil humain, mais des appareils photos numériques peuvent les détecter. Le capteur utilisé dans notre application est SR150 1HC [3]

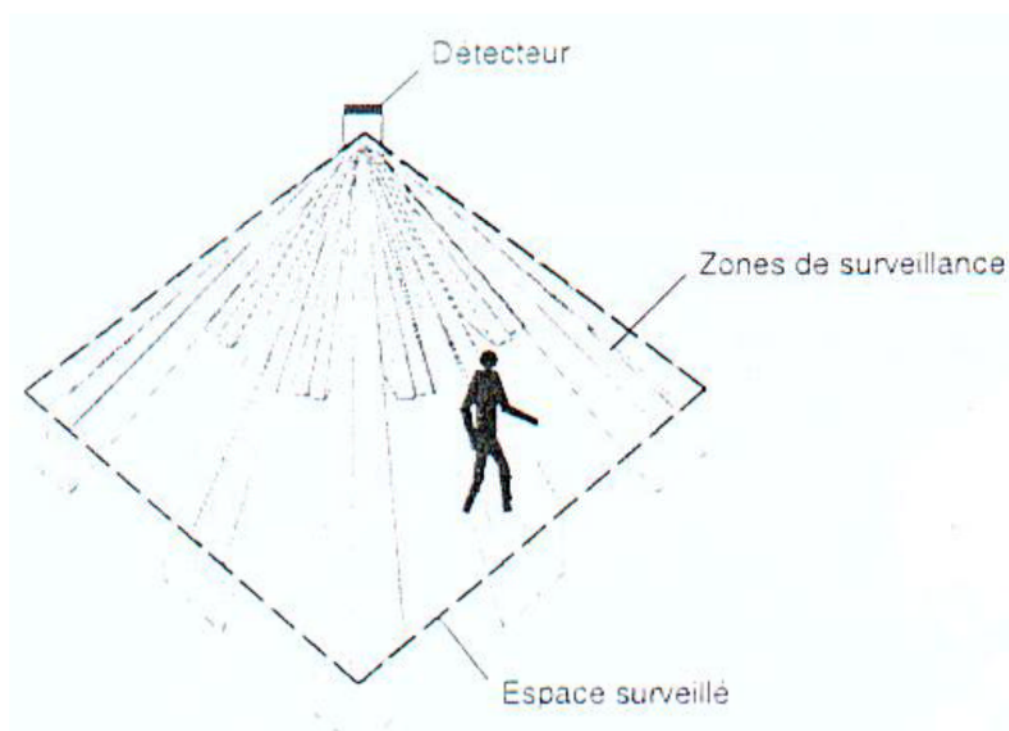


Figure 7: zone de surveillance

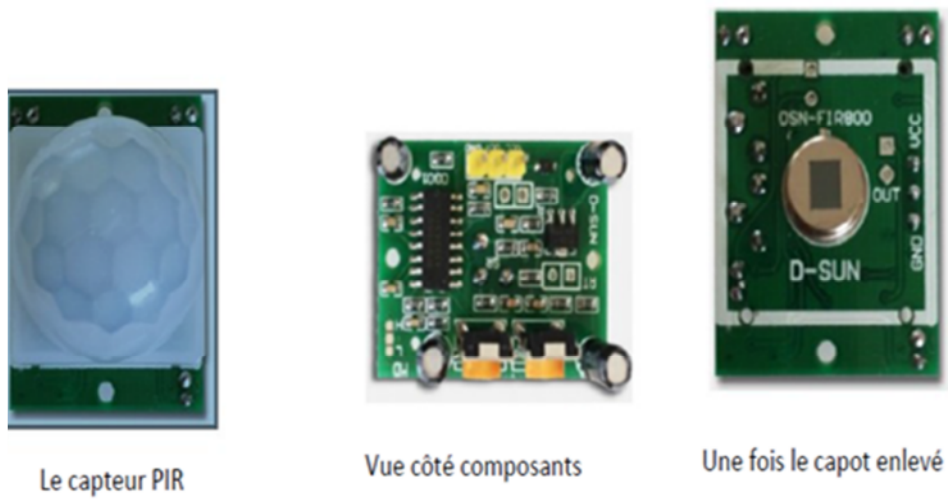


Figure 5: Capteur de mouvement

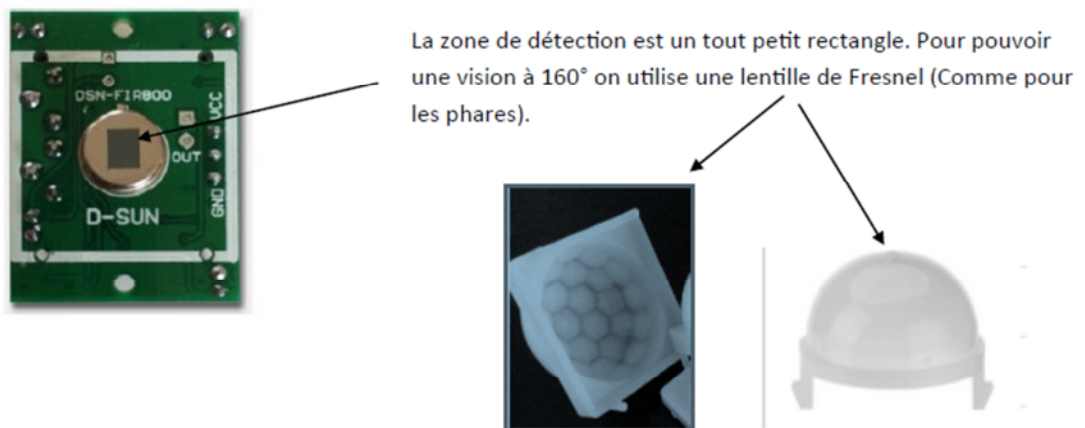


Figure 6 : la zone de détection

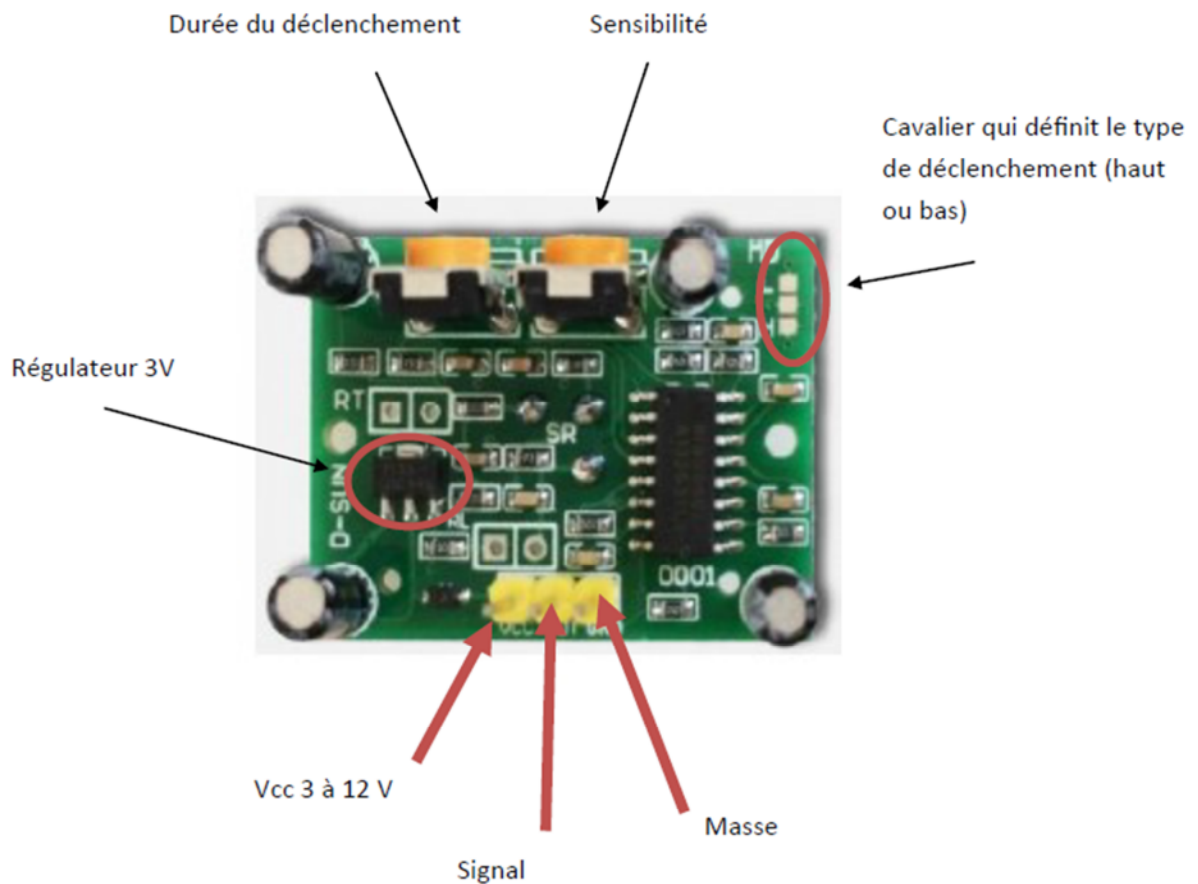


Figure 7: Les zones importantes du capteur

a) Les caractéristiques d'un capteur de mouvement

- Plage de détection: Env. 7m / 23 pieds
- Angle de détection : Moins de 100 degrés
- Chaîne de tension: DC 4,5 V-20V
- Tension d'alimentation: 3.3 ~ 5V, 6V Maximum
- Courant en fonctionnement: 15uA
- Température de fonctionnement: -20 ~ 85 °C
- Tension de sortie: Niveau haut 3V, bas 0V
- Durée de la sortie au niveau haut: environ 2.3 à 3 secondes
- Taille du module: 30mm × 22mm

La télésurveillance est un ensemble de systèmes permettant d'assurer un service de surveillance à distance. On appelle système de télésurveillance tous les appareils qui permettent de communiquer avec le lieu surveillé, et qui communiquent eux-mêmes avec quelqu'un de l'extérieur. Ils sont très variés. Parmi les éléments de notre système de surveillance, le module GSM offre une grande opportunité dans la transmission des messages d'alerte.

I.3.4. GSM

I.3.4.1. Historique

C'est au début des années 90 que la première génération de téléphone mobile va évoluer. C'est en 1982 qu'est créé le Groupe Spécial Mobile (GSM), qui a été établi pour créer une norme pour les téléphones mobiles. Elle va imposer certains choix technologiques en 1987, comme le chiffrement des informations ou encore l'utilisation de plusieurs canaux radio. C'est en 1991 que la première communication expérimentale sur la norme GSM (alors devenu Global System for Mobile Communications) sera effectuée. À ce moment-là, les spécifications techniques imposent une bande de fréquence à 900 Mhz, qui pourra être doublée à 1800 Mhz pour l'améliorer. Cette norme est toujours utilisée pour les communications téléphoniques en Europe, tandis que les États-Unis et le Canada utilisent des bandes de 850 Mhz et 1900 Mhz.

I.3.4.2..La couverture GSM dans le monde

les réseaux GSM sont implantés sur une large portion de la surface terrestre ; une condition nécessaire de connexion à un réseau est la disponibilité de station de base (cellule radio) à proximité de l'emplacement du téléphone mobile (la charge de la batterie du téléphone influence également la portée de réception). Ainsi, les zones faiblement peuplées (haute montagne, larges campagnes, déserts), les hautes altitudes (en avion par exemple), les cavités terrestres (grottes, tunnels) et la mer (au-dessus comme en dessous de la surface) sont souvent dépourvues d'accès réseau GSM.

Les réseaux GSM (Global System for Mobile Communications) couvrent 219 pays ou territoires en 2014.

En 2016, l'association GSMA comptabilisait pour les réseaux mobiles GSM et dérivés (UMTS et LTE) 4,8 milliards d'utilisateurs uniques et 7,9 milliards de CARTE SIM connectables à travers le monde.

I.3.4.3. Architecture du réseau GSM : Le GSM comprend 2 sous systèmes :

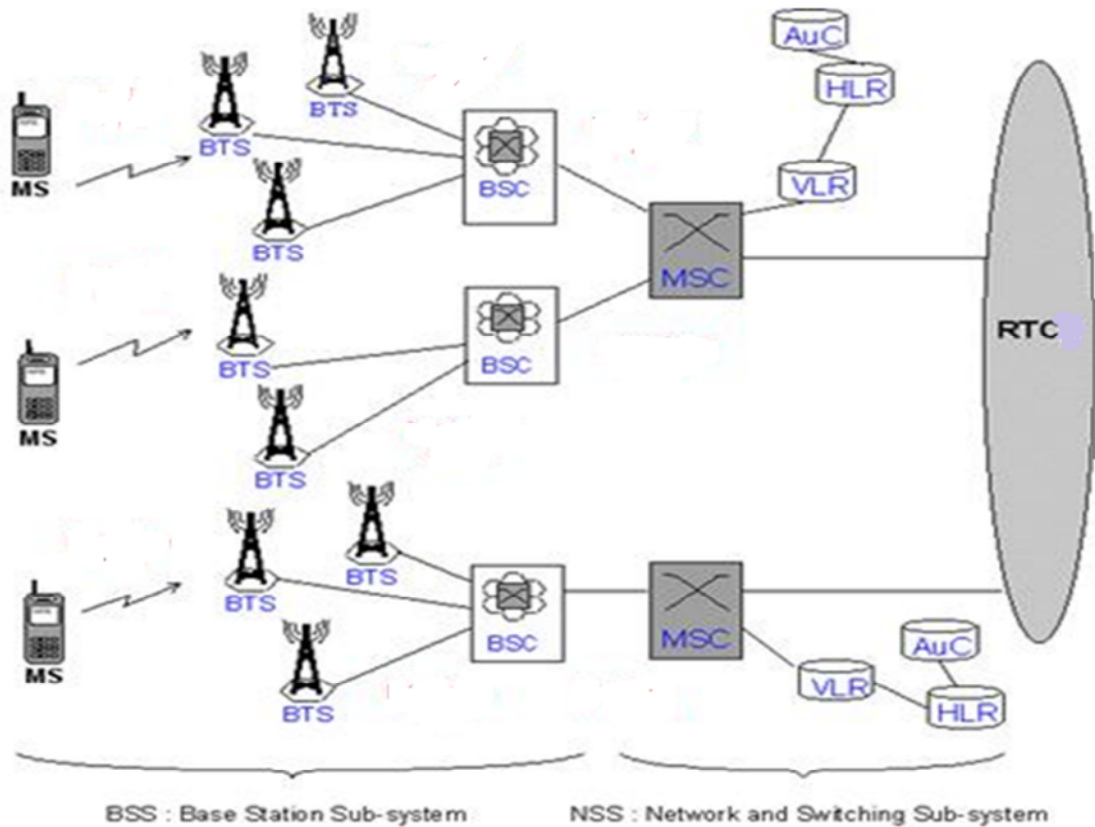


Figure 8 : Architecture d'un module GSM

- a) **Le sous systèmes radio (Bss) :(Base subscriber station) :** Assure la transmission et gère les ressources radios .

La BTS : (base tranceiver station) : émetteur récepteur situer sur des points hauts

Le BSC: (base station Controller) contrôleur de station de base, permet aussi la concentration de plusieurs BTS .

- b) **Le sous système réseau NSS :(network sub system) :** assure la fonction d'acheminement des appels et la mobilité.

Le MSC (mobile switching center) : c'est un commutateur qui permet d'assurer l'aiguillage et l'acheminement du signal .

HLR (home location register) : base de données de location des abonnés et leurs caractéristiques.

Le VLR (visitor location register) : base de données de localisation des visiteurs.

c) Le sous –système d’exploitation OSS (operating sub-système)

A la fin de l’implémentation d’un réseau GSM, l’OSS continue à superviser et gérer à l’échelle nationale le réseau (redimensionnement télémesure etc...)

AUC (authentification –center) :

Le réseau autorise l’abonné à y accéder en comparant une clé de cryptage du mobile avec celle mémorisée dans le réseau ; les clé doivent être identiques. [4]

I.3.4.4.Module GSM, ou modem GSM :

Un module GSM ou modem GSM est un boîtier électronique muni d’une carte SIM qui se connecte au réseau téléphonique comme un téléphone portable ; ainsi il dispose de son propre numéro de téléphone et fonctionne partout où il existe un réseau cellulaire GSM.

Le module GSM n’est pas verrouillé à un seul réseau il peut donc être utilisé avec n’importe quel fournisseur de réseau GSM. Lorsque vous appelez il rejette l’appel sans y répondre, donc il n’y a pas de frais de communication engagé, il a une mémoire non-volatile et sauvegarde les paramètres dans le cas d’une Interruption de son alimentation.

Le module GSM, , est autonome grâce à sa batterie interne. ainsi en cas de coupure secteur il vous envoie un SMS pour vous le signaler. Dans notre projet on ‘a utilisé le module GSM SIM 900 et la figure 12 montre le schéma bloc d’un système à base d’un module GSM.[5]



Figure 9:le schéma bloc de système

- **L'utilisateur** : permet d’envoyer un message SMS à un module GSM par un téléphone portable .Le message contient des informations codées pour les commandes.

- **Module GSM :** permet de recevoir ou d'envoyer des sms et communiquer avec une carte électronique par une communication série (UART) ; les commandes utilisées dans cette connexions ce sont des AT COMMANDS.

- **Carte électronique:** permet de communiquer avec module GSM lire les messages et envoyer des commandes et exécuter les différents processus comme traitement, décodage et stockage des informations, affichage des messages etc.

La figure 8 illustre le module GSM SIM 900

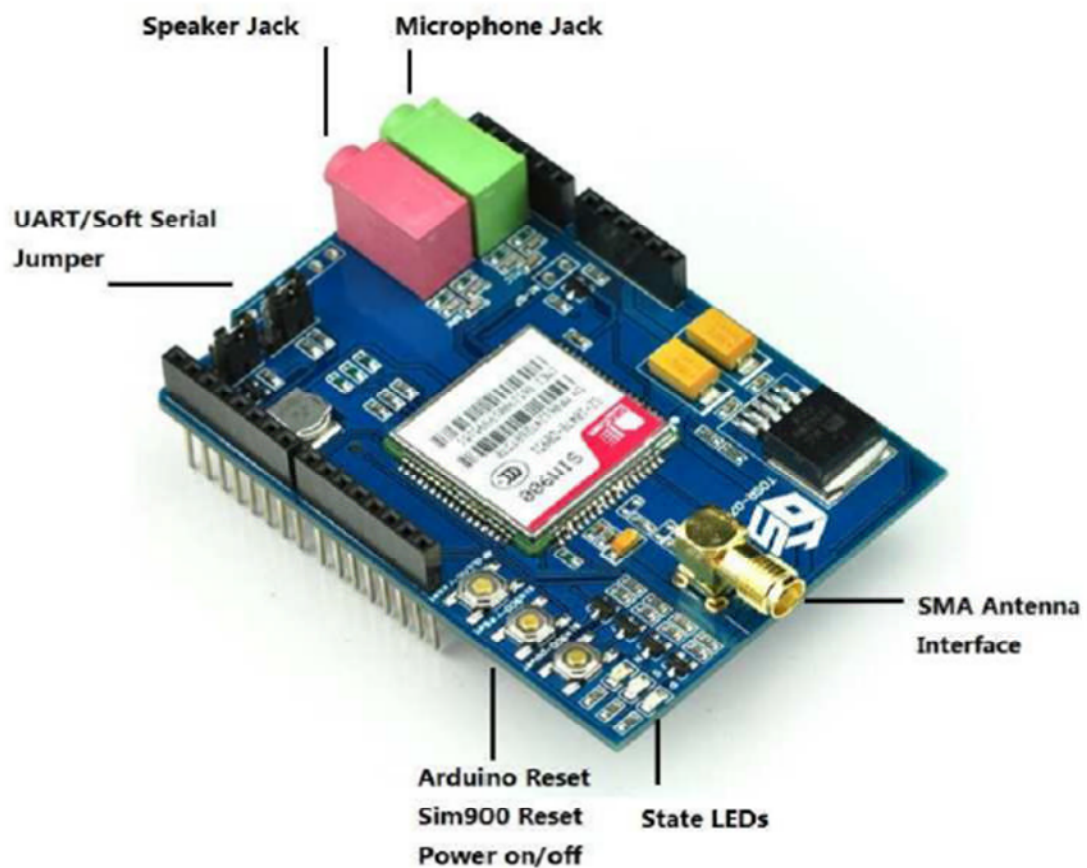


Figure 10 : module GSM SIM900

a) caractéristiques module GSM SIM900

- tension d'alimentation: de 4,5 à 5,5 VCC.
- Consommation électrique (impulsion) : 2 000 mA.
- Consommation électrique (en continu) : 500 mA.
- Vitesse de transmission : 115 200 bp.
- Service de messages courts.
- Sélection du port de série libre.
- fonctionne sur les réseaux GSM dans tous les pays du monde.
- Fonction d'activation / désactivation et réinitialisation prise en charge par l'interface Arduino Le module GSM, comporte un certain nombre d'entrées et de sorties en fonction des modèles (Entrées analogiques ou digitales).

I.3.4.5.1. Entrées/ sortie de module GSM

a) Entrées analogiques : Pour brancher des sondes analogiques de mesures et y programmer des seuils minimum et maximum. Exemples : sonde de températures.

b) Entrées digitales ou numériques NO ou NF

Pour brancher des détecteurs du type : Intrusion, présence, contact de porte, détecteur de fumée, Gaz, etc...

c) Sorties du module GSM

Les sorties sont activées en envoyant au modem GSM un SMS codé, un relais de puissance est alors activé pour commander tous matériels et équipements électriques /Portes, portails, chauffages, volets climatisation , ,moteurs ,pompes, etc...

I.3.4.4 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.3.4.4.1 Commande et surveillance par GSM

La commande par GSM peut connecter/ déconnecter à distance tout matériel électrique comme moteurs, pompes, portes, climatisations....., 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 autorisés seront habilités à commander votre maison.

La surveillance par GSM permet de contrôler tous les paramètres d'une maison, bureau, entreprise, etc....coupure de courant, variation de température, mouvement etc.... sans se déplacer, on recevant des messages ou des notifications sur le téléphone

I.4. Discussion

Dans ce chapitre nous avons présenté les différentes applications de la domotique ainsi qu'une vue d'ensemble de capteurs. Nous avons aussi décrit le capteur infrarouge utilisé comme détecteur de présence ainsi que le module GSM et ses caractéristiques. Dans le chapitre qui suit nous présentons la carte Arduino sous ces deux aspects matériel et logiciel.

II.1 Préambule

Aujourd'hui, l'électronique est de plus en plus remplacée par de l'électronique programmée. Son but est de simplifier les schémas électroniques et par conséquent réduire l'utilisation de composants électroniques, réduisant ainsi le coût de fabrication d'un produit. Il en résulte des systèmes plus complexes et performants pour un espace réduit, d'où la naissance du projet ARDUINO qui a révolutionné le domaine de l'électronique à l'échelle mondiale.

Le système ARDUINO donne la possibilité d'allier les performances de la programmation à celles de l'électronique. Ce que nous allons voir dans ce chapitre.

II.2.Historique

L'ARDUINO est un projet issu d'une équipe de développeurs composée d'enseignants qui s'appelle BANZI et d'étudiants de l'école de Desing Ivrea en Italie en 2005.

Les étudiants qui se plaignent envers leur enseignant, de ne pas avoir accès à des solutions bas prix pour accomplir leur projets de robotique.par la suite l'enseignant BANZI a contacté DAVID CUARTIELLES, qui est un ingénieur espagnol spécialisé dans les microcontrôleurs pour créer leur propre carte en embarquant dans leur histoire un des étudiant de Banni David Mellis qui sera chargé de créer le langage de programmation allant avec la carte. En deux jours David écrira le code , trois jours après la carte était crée .ils décidèrent de l'appeler ARDUINO.

II. 3.définition de la carte Arduino

La carte ARDUINO est une plate forme prototypage électronique open- source basée d'une part sur matériel et d'autre part sur un ensemble de logiciel facile à utiliser.

Elle possède un microcontrôleur facilement programmable ainsi que de nombreuses entrées-sorties. Plusieurs types de ces cartes existent et qui se différencient par la puissance du microcontrôleur, par la taille et la consommation. Le choix d'une carte ARDUINO s'effectue en fonction des besoins de projet, leur programmation est en C⁺⁺.

Le modèle UNO de la société ARDUINO est une carte électronique dont le cœur est un microcontrôleur ATMEL de référence ATMega328 de 8bits de la famille AVR dont la programmation peut être réalisée en langage C/C⁺⁺.la figure xx représente le modèle UNO.

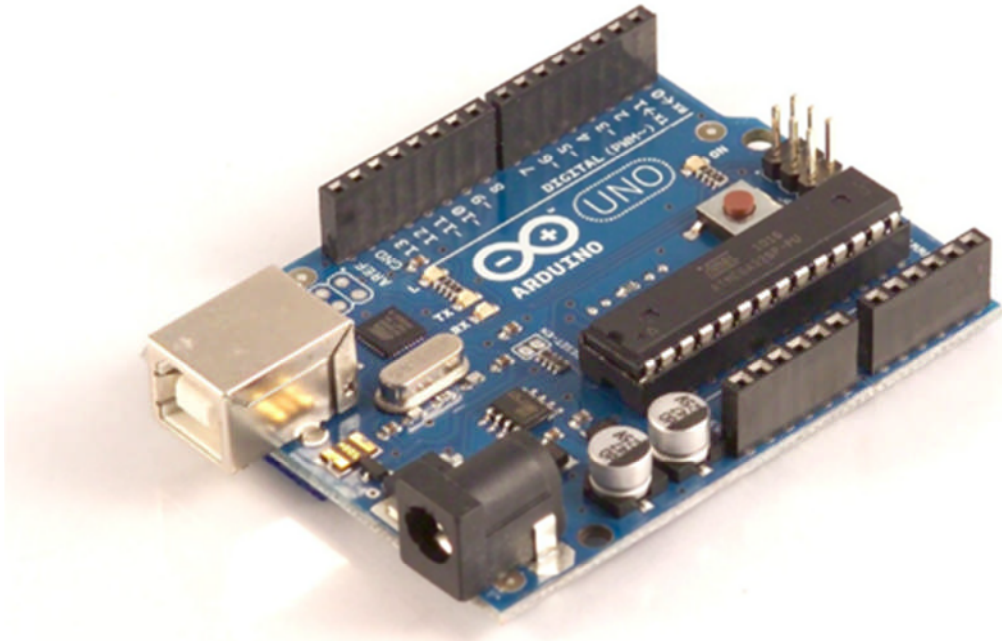


Figure : 14 la carte Arduino UNO

II.3.1 Caractéristiques de la carte ARDUINO UNO

- ❖ Microcontrôleur : ATmega328
- ❖ Tension d'alimentation interne = 5V
- ❖ tension d'alimentation (recommandée)= 7 à 12V, limites =6 à 20 V
- ❖ Entrées/sorties numériques : 14 dont 6 sorties PWM
- ❖ Entrées analogiques = 6
- ❖ Courant max par broches E/S = 40 mA
- ❖ Courant max sur sortie 3,3V = 50mA
- ❖ Mémoire Flash 32 KB
- ❖ Mémoire SRAM 2 KB
- ❖ mémoire EEPROM 1 KB
- ❖ Fréquence horloge = 16 MHz
- ❖ Dimensions = 68.6mm x 53.3mm

II.3.2. Description de la carte Arduino

La figure suivante illustre le schéma synoptique de la carte ARDUINO :

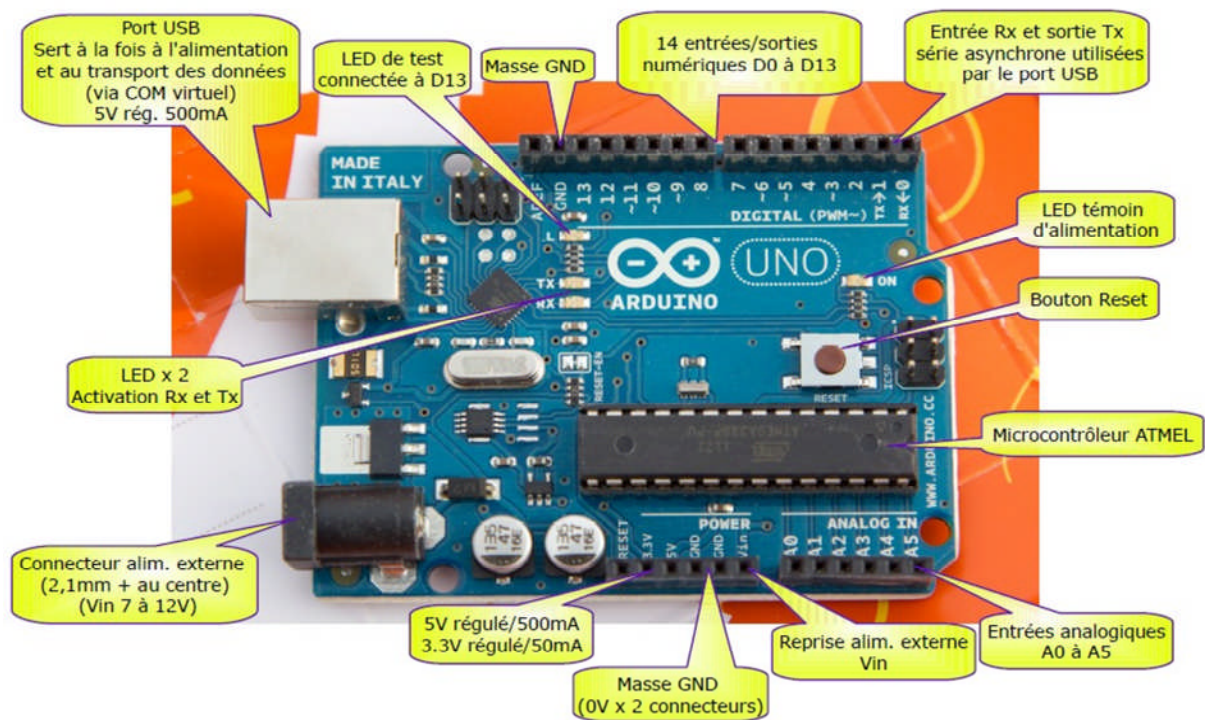


Figure15 : Description de la carte ARDUINO UNO

II.3.3. Présentation matériel de la plateforme ARDUINO

L'Arduino est composée de deux parties indissociables : la carte qui est la partie hardware avec laquelle on travaille en construisant chaque projet et la plateforme IDE Arduino qui est la partie logicielle sur le PC, celle-ci permet de mettre au point et de transférer le programme qui sera par suite exécuté par la carte.

II.3.3.1. Microcontrôleur

Le microcontrôleur de la carte Arduino UNO est un ATmega328. C'est un microcontrôleur ATMEL de la famille AVR 8bits.

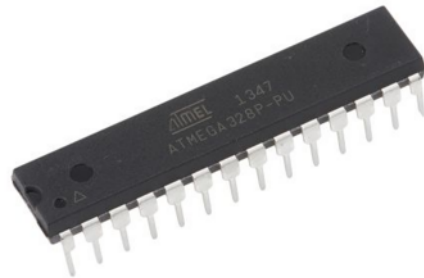


Figure 16 : Microcontrôleur ATMEGA 328

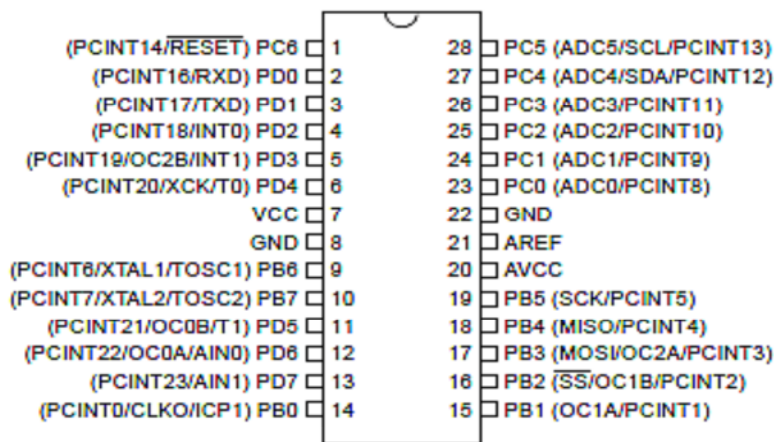


Figure 17: L'architecture interne de Microcontrôleur ATMéga 328

a) Caractéristique du microcontrôleur ATMEGA 328

- C'est un microcontrôleur ATMEL de la famille AVR 8 bits.
- FLASH de 32 ko ; mémoire de programme.
- SRAM 2 ko ; mémoire de données (volatiles) .

- EEPROM 1 ko ; mémoire de données (non volatiles).
- Digital I/O (entrées-sorties Tout ou Rien).
- Fréquence d'utilisation : 20 MHz max.
- 32 registres d'usage général.

Un microcontrôleur 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, il comprend essentiellement :

- ✓ Des mémoires.
- ✓ Un microprocesseur
- ✓ des éléments périphériques selon l'application.

a) **Les mémoires** : Il en possède 5 types :

- **La mémoire Flash** : C'est celle qui contiendra le programme à exécuter. Cette mémoire est effaçable et réinscriptible.
- **RAM** : c'est la mémoire dite "vive", elle va contenir les variables de votre programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur.
- **EEPROM** : C'est le disque dur du microcontrôleur. Vous pourrez y enregistrer des informations qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on l'on éteint le reprogramme.
- **Les registres** : c'est un type de mémoire utilisé par le processeur.
- **La mémoire cache** : c'est elle qui fait la liaison entre les registres et la RAM.

b) Le microprocesseur

Un microprocesseur : c'est le composant principal du microcontrôleur. C'est lui qui va exécuter le programme. On le nomme souvent le CPU.

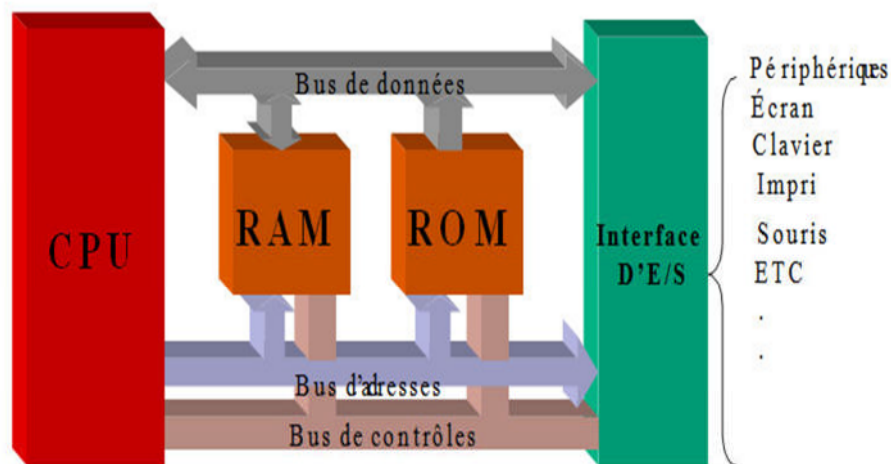


Figure18 : Système à microprocesseur

Bus : Il s'agit de plusieurs pistes électroniques qui sont reliées au microprocesseur. Ces bus assurent la communication interne et externe du microprocesseur.

- **Le bus de données** : c'est un ensemble de fils bidirectionnels qui va permettre le transfert de données entre les différents éléments du système. C'est par ce bus que sont transmises les données qui doivent être traitées par le microprocesseur. A l'inverse, c'est également par ce bus que transitent les résultats en sortie du microprocesseur. Autrement dit, toutes les données entrantes et sortantes du microprocesseur sont véhiculées par le bus de données qui fixe la longueur du mot échangé avec la mémoire.

- **Le bus de commande et de contrôle** : c'est un bus qui permet de véhiculer les signaux de contrôles et de commandes tels que l'horloge, les signaux Rd/Wr etc. ... Ce bus sert à coordonner tous les échanges d'informations décrits précédemment. Il véhicule des données qui valident la mémoire et les ports d'entrées /sorties. Il introduit des délais d'attente lorsque des informations sont envoyées à un périphérique qui présente une vitesse de traitement réduite. Le bus de commande évite les conflits de bus lorsque deux éléments cherchent à communiquer en même temps.

II.3.3.2. Entrées/ Sorties de la carte

II.3.3.2.1. Entrée /sorties numérique [6]

Les entrées / sorties numériques ne peuvent prendre que deux valeurs, la valeur LOW (GND ,0 V), et la valeur HIGH (~ 5 V). La valeur d'un port numérique peut donc être codée sur un bit, 0 ou 1, true ou false.

La carte ARDUINO comporte 14 I/O numériques (appelées DIGITAL sur la carte), numérotées de 0 à 13 (voir le schéma ci-dessus), et appelés D0, D1, D2, ... D13. Chacun de ces ports peut-être déclaré comme étant une entrée ou comme une sortie dans le programme du microcontrôleur.

Les deux premiers ports (D0 et D1) sont réservés à la communication série, il ne faut pas les utiliser. Le dernier port, D13, possède un indicateur lumineux, une LED qui s'allume quand le port est HIGH, et qui s'éteint quand le port est LOW.

Le port GND est la masse de la carte (0 V).

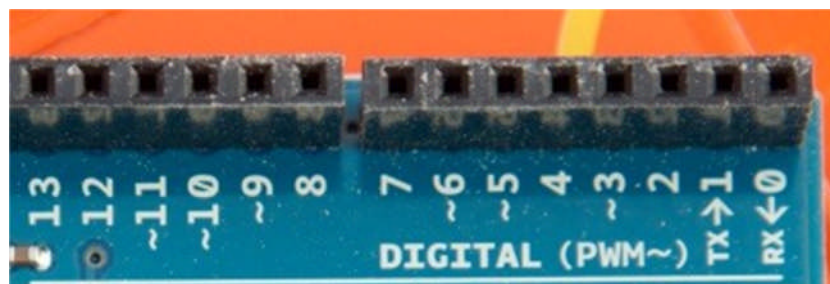


Figure19 : Entrée /Sortie numérique

- ❖ Chacun des ports D0 à D13 peut être configuré dynamiquement par programmation en entrée ou en sortie.
- ❖ Les signaux véhiculés par ces ports sont des signaux logiques compatibles TTL, c'est-à-dire qu'ils ne peuvent prendre que deux états HAUT (5 Volts) ou BAS (0 Volt).
- ❖ En pratique, les ports D0 et D1 sont réservés pour la liaison série asynchrone (port COM virtuel via le câble USB) et ne sont pas exploités pour d'autres utilisations.
- ❖ À noter que chacun des ports ne peut fournir ou absorber un courant supérieur à 40 mA .

II.3.3.2 Entrées analogique [6]

Une entrée analogique est une sorte de voltmètre : la carte lit la tension qui est appliquée sur le Port. Cependant le microcontrôleur ne travaille qu'avec des chiffres il faut donc transformer la Tension appliquée en sa valeur numérique.



Figure 20 : Entrées analogiques

Par défaut et contrairement aux entrées/sorties numériques qui ne peuvent prendre que deux états HAUT et BAS, ces six entrées peuvent admettre toute tension analogique comprise entre 0 et 5 Volts. Pour pouvoir être traitées par le microcontrôleur, ces entrées analogiques sont prises en charge par un CAN (Convertisseur Analogique Numérique ou ADC pour (Analog Digital Converter) dont le rôle est de convertir l'échantillon de tension V_E en une grandeur numérique binaire sur n bits.(figure 22).

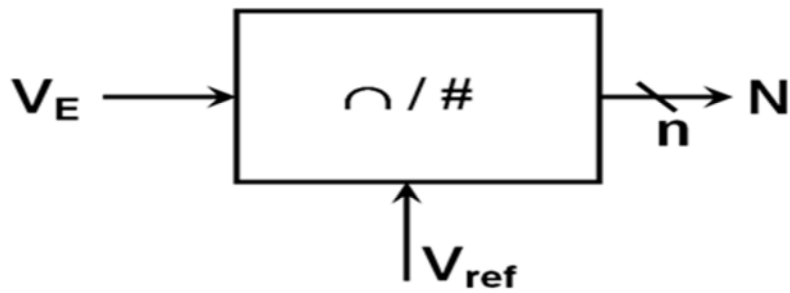


Figure 21 : Convertisseur analogique numérique

Le principe de la conversion Analogique-Numérique est représenté ci-dessous (avec $n=3$ bits et la tension de référence $V_{ref}=5$ Volts) :

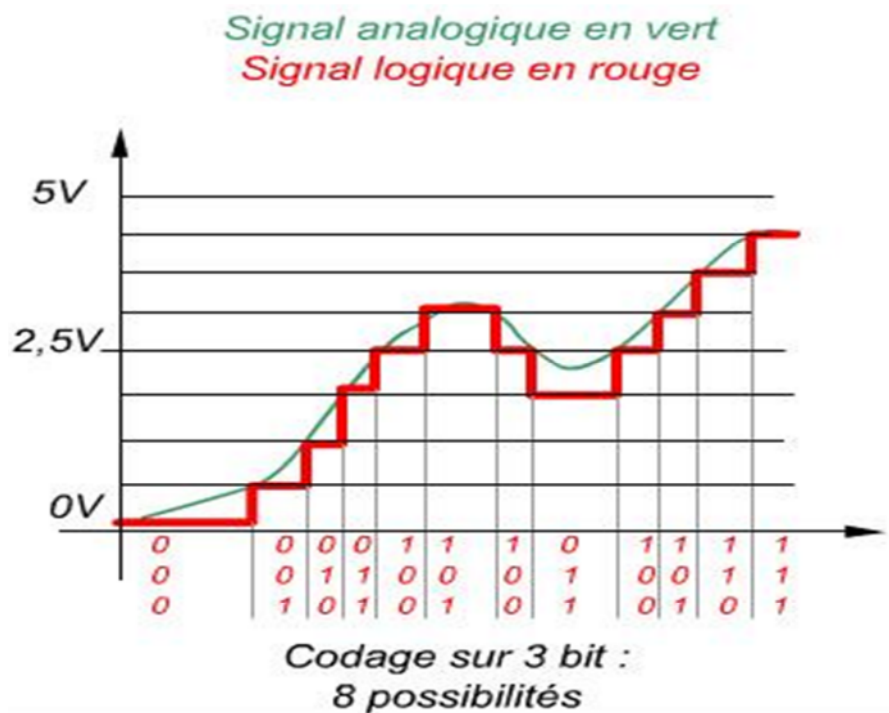


Figure 22 : Graphe de conversion

❖ *Visualisation :*

- ✓ LED verte : alimentation.
- ✓ LED jaune ou verte : active lors de téléchargement de programme (LED de transmission réception).
- ✓ LED jaune : de teste.
- ✓ Reset : pour la réinitialisation du microcontrôleur.

II.3.4. Alimentation de la carte ARDUINO

La carte Arduino UNO peut être alimentée soit via la connexion USB (qui fournit 5v jusqu'à 500 mA) ou à l'aide d'une alimentation externe.

la source d'alimentation est sélectionnée par la carte .l'alimentation externe (non USB) soit un adaptateur secteur (pouvant fournir typiquement de 3v à 12vsous 500mA) ou des piles.

Il est noté qu'il est strictement dangereux d'utiliser une alimentation externe via la prise jack et d'avoir le câble USB connecté (risque de destruction de la carte).

Les broches d'alimentation sont les suivantes :

- **VIN** : (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 on peut accéder à la tension d'alimentation sur cette broche.
- **5v** : La tension régulée utilisée pou faire fonctionner le microcontrôleur et les autres composants de la carte.
- **3.3v** : Une alimentation de 3.3v 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 50 mA.

Remarque : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable « tension régulée » obtenue à l'aide d'un composant appelé un régulateur et qui est intégré dans la carte Arduino .Le 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[7].

Les broches du microcontrôleur sont reliées à des connecteurs selon le schéma ci-dessous.

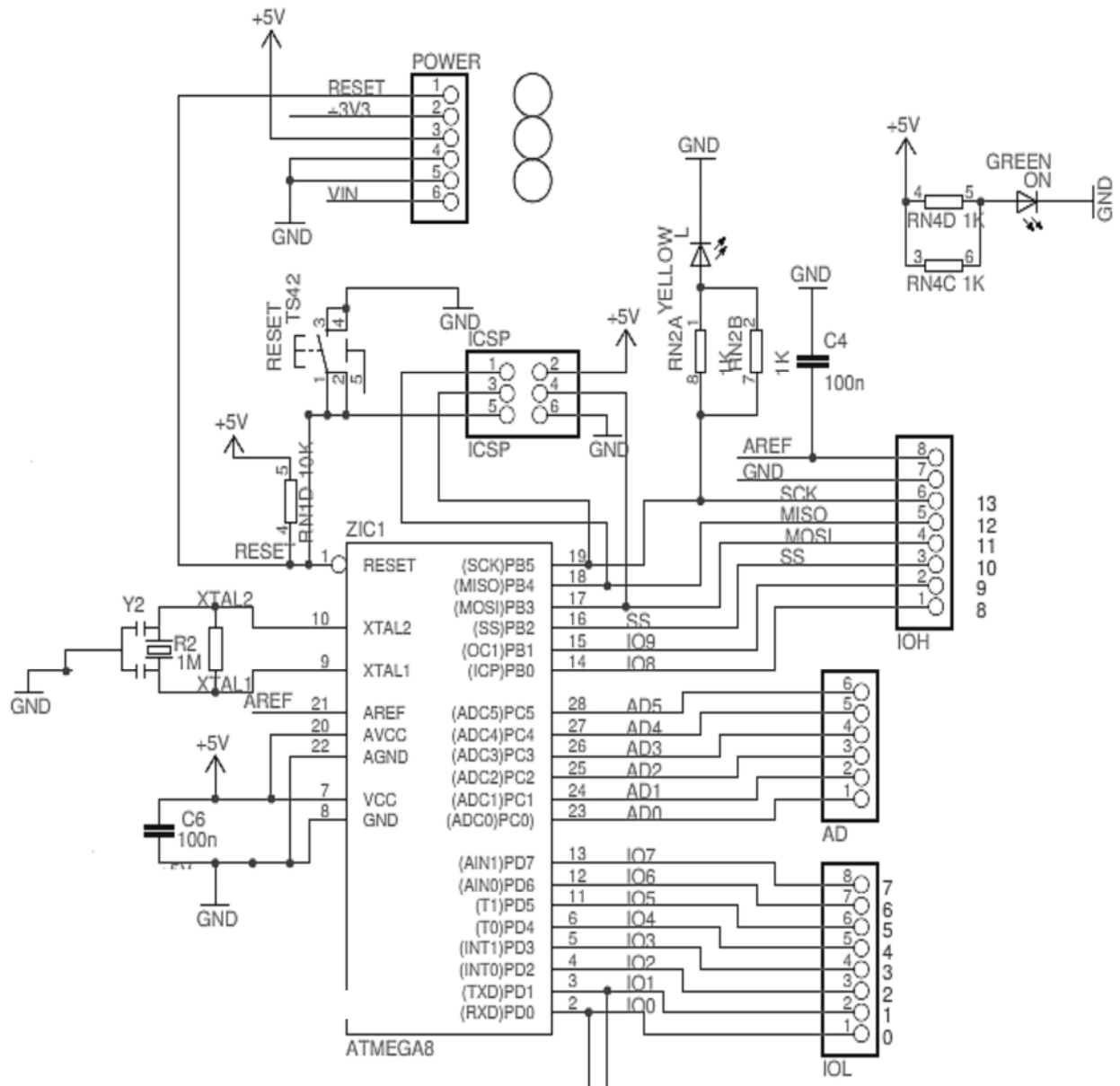


Figure23 : Schéma Electrique de la carte ARDUINO UNO

II.4. La partie logicielle

Un logiciel Open Source est un programme informatique dont le code source est distribué sous une licence permettant à quiconque de lire, modifier ou redistribuer et parmi eux on trouve le logiciel Arduino.

Le langage Arduino est implémenté en C/C++ et basé sur le câblage. lorsque nous écrivons un Arduino croquis, nous utilisons implicitement la bibliothèque de câblage ,qui est inclus avec L'IDE ceci nous permet de faire des programmes exécutables en utilisant seulement deux fonctions : setup () et loop().

L'IDE permet d'écrire, de modifier un programme et de le convertir en une série d'instructions compréhensibles par la carte.

II.4.1 Les fonctions du langage Arduino

Comme nous avons dit, le langage Arduino utilise deux fonctions ; « setup () » et

« Loop () » qui fonctionnent comme suite :

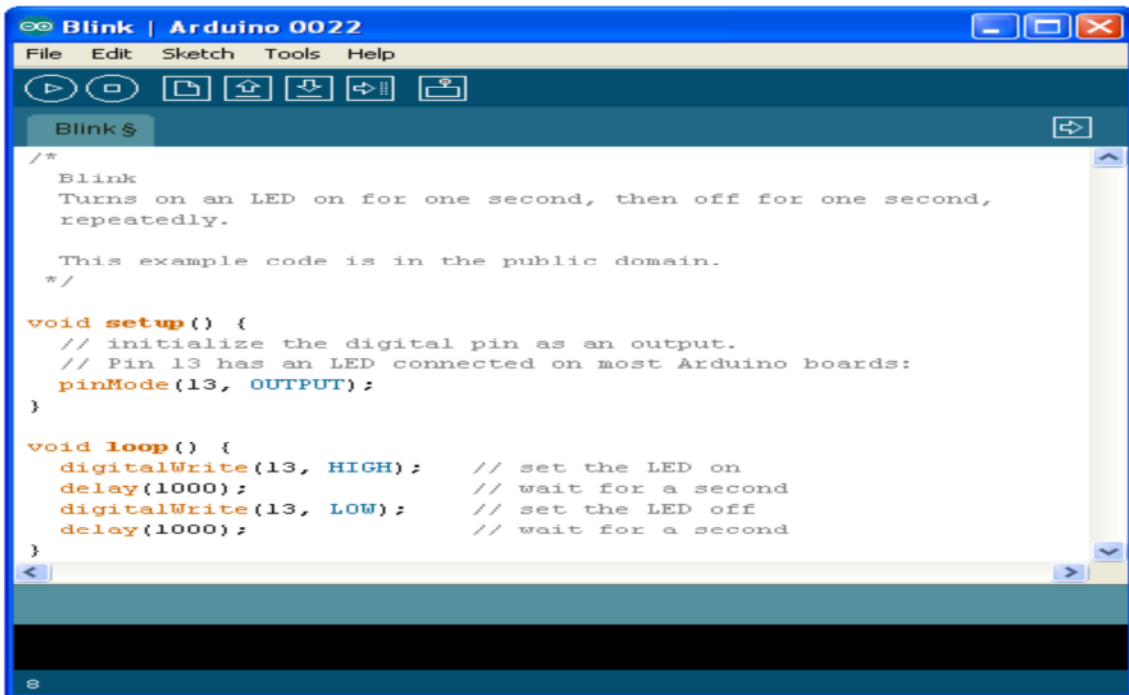
- **La fonction setup ()**

cette fonction est exécutée qu'une seule fois, juste après nous téléchargeons une nouvelle esquisse à la carte, et en suite chaque fois que nous allumerons notre Arduino .nous utilisons cette fonction pour initialiser les variables, mode de broches, etc.

- **La Fonction loop ()**

Elle est exécutée en continue dans une boucle sans fin lorsque la carte Arduino est alimentée. cette fonction est le noyau de la plupart des programmes.

La(figure 24) représente un exemple qui montre la structure de chacune des fonctions :

The image shows a screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 0022". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for running, stopping, saving, and other functions. The main text area contains the following code:

```
/*
  Blink
  Turns on an LED on for one second, then off for one second,
  repeatedly.

  This example code is in the public domain.
  */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

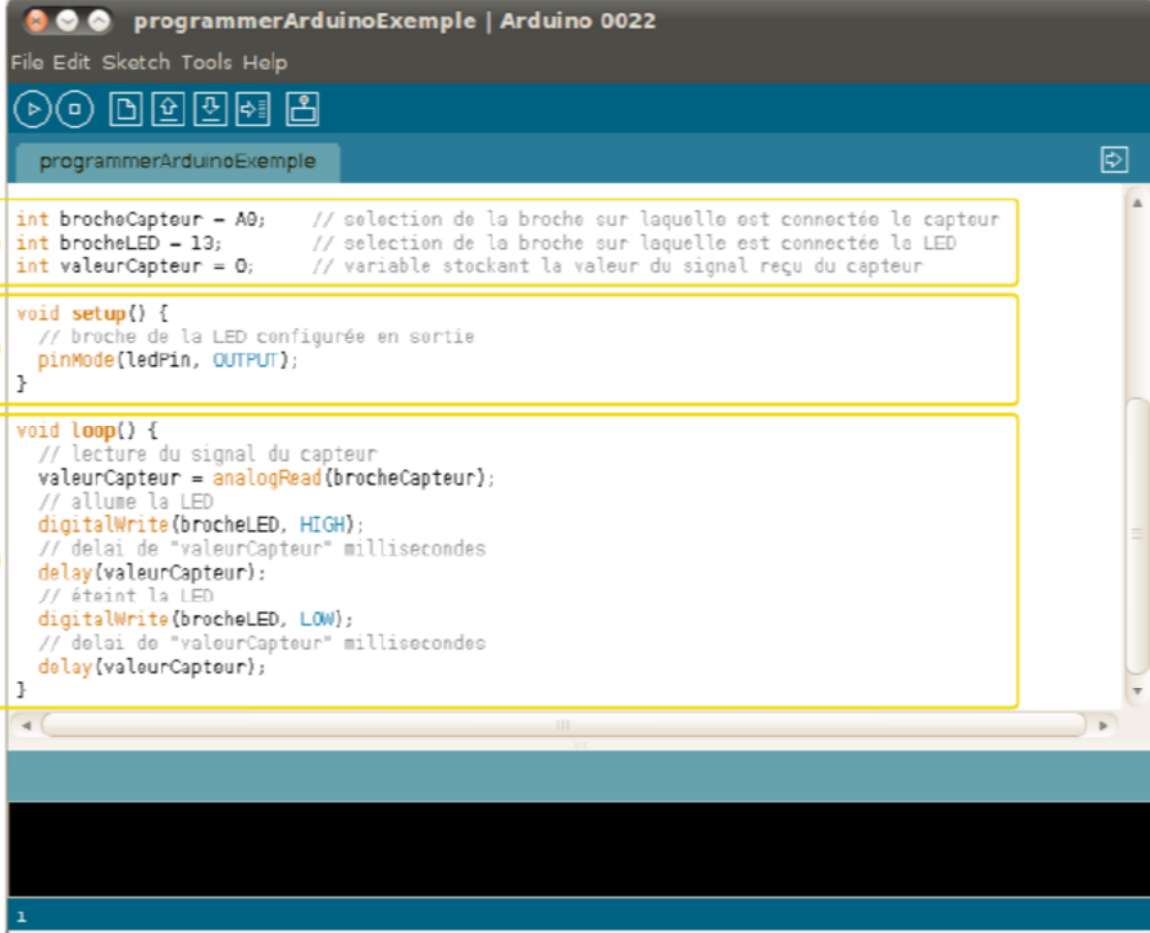
void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);           // wait for a second
}
```

Figure 24: Les fonctions du programme

II.4.2 Structure de programme

Un programme utilisateur Arduino est une suite d'instruction élémentaire et séquentielle. tous les programmes Arduino comportent trois étapes comme illustre la figure 25.

- ✓ 1. la partie déclaration des variables (optionnelle)
- ✓ 2. la partie initialisation et configuration des entrées/sorties : la fonction `setup ()`
- ✓ 3. la partie principale qui s'exécute en boucle : la fonction `loop ()`



```
int brocheCapteur = A0; // selection de la broche sur laquelle est connectée le capteur
int brocheLED = 13; // selection de la broche sur laquelle est connectée la LED
int valeurCapteur = 0; // variable stockant la valeur du signal reçu du capteur

void setup() {
  // broche de la LED configurée en sortie
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // lecture du signal du capteur
  valeurCapteur = analogRead(brocheCapteur);
  // allume la LED
  digitalWrite(brocheLED, HIGH);
  // delai de "valeurCapteur" millisecondes
  delay(valeurCapteur);
  // éteint la LED
  digitalWrite(brocheLED, LOW);
  // delai de "valeurCapteur" millisecondes
  delay(valeurCapteur);
}
```

Figure 25 : Structure de programme

Dans chaque partie d'un programme on utilise différentes instructions issues de la syntaxe du langage Arduino.

Lorsque le code est écrit dans l'interface de programmation, certains mots apparaissent en différentes couleurs, clarifient le statut des différents éléments :

- En rouge, apparaissent les mots clés reconnus par le langage Arduino comme des fonctions existantes ;
- En bleu, apparaissent les mots- clés reconnus par le langage Arduino comme des constantes.

- En gris, apparaissent les commentaires qui ne seront pas exécutés dans le programme. il est utile de bien commenter son code pour s'y retrouver facilement ou pour le transmettre à d'autres personnes .on peut déclarer un commentaire de deux manières Différentes :

Dans une ligne de code, tout ce qui se trouve après « // » sera un commentaire ou bien entre les signes /****/ .

II.4.3.Description des différents menus

II.4.3.1.Menu fichier [8]

La figure ci-dessous illustre les options de menus fichier.

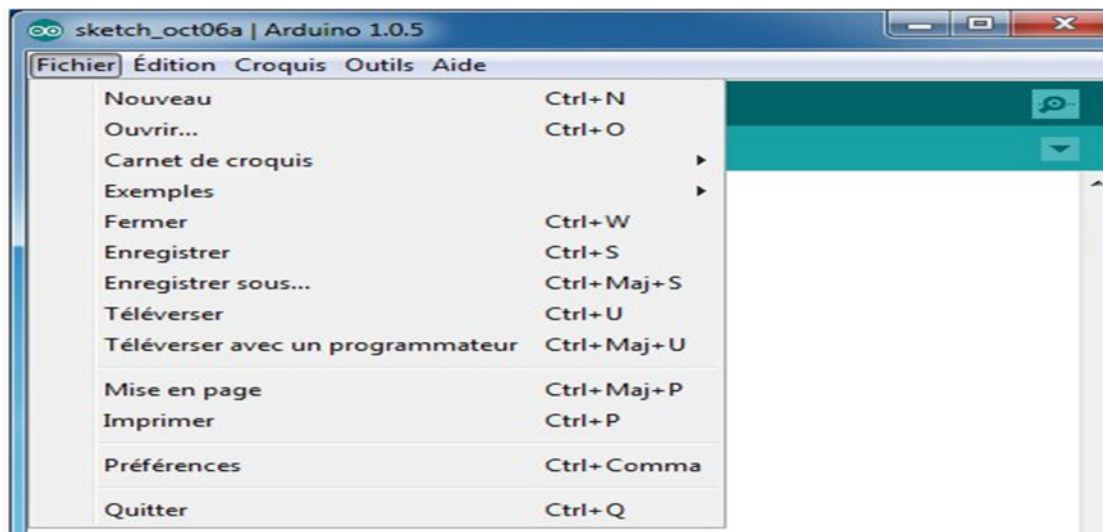


Figure 26 : Menu fichier

Nouveau : permet d'écrire un nouveau programme, donc ouvre un nouveau fichier qui aura l'extension « .ino » lors de son enregistrement.

Ouvrir... : permet d'ouvrir le programme déjà enregistré sur l'ordinateur.une fenêtre de navigation s'ouvre pour retrouver le programme à ouvrir.

Ouvert récemment > : option bien outille dans une phase de développement car elle propose les programmes sur lesquelles vous avez travaillé les jours précédent ;la recherche est ainsi plus rapide .

Carnet de croquis : cette option offre un sous-menu proposant tous vos programmes (croquis ou sketch en anglais).

Exemples : ici, on trouve de nombreux programmes donnés (comme Bilink), et classée par catégories ; ils sont bien outils pour bien se familiariser avec le langage d'Arduino. On y trouve choix, de la langue (si votre IDE ne s'est pas installer dans votre langue) ,taille de police, afficher les numéros de lignes (nous recommandons de cocher cette option afin de vous y également tous les exemples de toutes les bibliothèques que l'on aura incluses dans l'IDE. On trouve les flèches « > » et les flèches « haut /bas » pour se déplacer dans les différent exemples.

Fermer : permet de fermer la fenêtre de l'IDE, c'est-à-dire du programme en cours de développement. si cette fenêtre est la seule ouvert, l'IDE quitte complètement.

Enregistrer : enregistrer les dernières modifications de votre programme.

Enregistrer sous ... : enregistrer votre programme en lui donnant un nom et permet le choix de l'emplacement.

Mise en page : permet de règles les options mise en page avant d' impression de programme, comme l'orientation de la page (portail ou paysage) et la taille des marges.

Imprimer : permet d'imprimer votre programme.

Préférences : cette option ouvre une nouvelle fenêtre ou il est possible de régler plusieurs options emplacement où on veut trouver les croquis retrouver plus facilement dans vos programmes).

II.4.3.2.Menu Edition [8]

La figure (27) montre le menu Edition :



Figure 27 : Menu Edition

Annuler et Rétablir : permet d'annuler la dernière action réalisée (ou de la rétablir). la fonction « annuler » vous permet de les récupérer sans avoir à les retaper. la fonction « rétablir » vous permet de les supprimer à nouveau.

Couper, copier, coller : ces options sont identiques à ce qu'on trouve dans un traitement de texte classique. le texte doit d'abord avoir été sélectionné.

Copier pour le forum : permet de copier dans le presse-papier, tout ou partie de votre programme pour le poster dans un forum.

Copier en tant qu'HTML : cette option réalise une copie du programme pour pouvoir l'inclure dans une page web.

Tout sélectionner : permet de sélectionner l'intégralité de votre programme (par exemple pour le copier dans un autre programme, dans un traitement de texte.).

Aller à la ligne : ouvre une fenêtre qui permet de spécifier le numéro de la ligne de programme où on veut apparaître le curseur.

Commenter /décommenter : permet d'ajouter (ou de retirer) le double caractère «//» d'un commentaire .cette option peut être utile .en phase de mis au point pou isoler des section des programmes qui pourraient poser un problème.

Augmenter l'indentation (ou la réduire) : permet de déclare vers la droite (ou vers la gauche) le texte sélectionné afin de présenter le programme de façon plus lisible et plus compréhensible pour un autre programmeur.

Trouver... : permet de rechercher des mots ou des expressions et de les remplacer par d'autres mots où expressions.

II.4.3.3.Menu croquis [8]

La figure 28 illustre le menu croquis

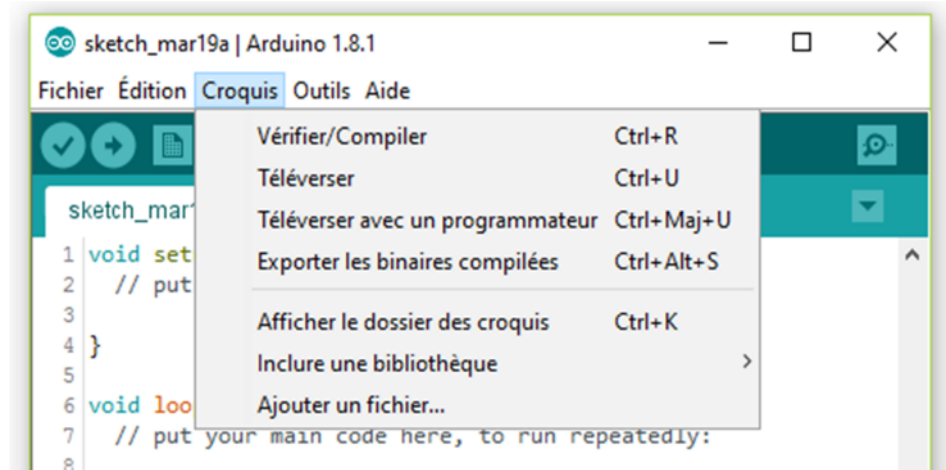


Figure 28:Menu croquis

Vérifier / compiler : permet de vérifier que votre programme est bien conforme à la syntaxe (règles d'écriture) , et que toutes les variables sont bien définies, ainsi que les appels aux bibliothèques ou d'autres fichiers. Si cette étape est franchie, le fichier est compilé et une indication de succès est affichée en couleurs blanche dans la partie noir de la fenêtre.

Téléviser :commence par vérifier /compiler puis transmet le fichier binaire obtenu à la mémoire flash (mémoire de programme) de la carte Arduino si la vérification /compilation n'a pas d'erreurs.

Téléverser avec un programmeur :la même chose avec précédemment pour ceux qui disposent d'un programmeur et préfèrent d'employer afin d'écraser le bootloader et ainsi utiliser toute la capacité de la mémoire de programme.

Exporter les binaire compilées :export, dans le dossier du croquis ,le fichier génère à la phase de compilation (voir plus haut) sous forme d'un fichier .hex (hexadécimale).

Afficher le dossier des croquis :permet d'afficher le répertoire où est stockés le programme en cours d'écriture et d'autre fichiers importés (.hex ,.ino, .cpp,.h etc...).

Inclure une bibliothèque > : permet de voir les bibliothèques déjà installées et les gérer et aussi en inclure de nouvelles.

Ajouter un fichier... : permet d'ajouter un fichier qui sera copier dans le dossier du programme en cours de développement .cette option ouvre une fenêtre de navigation pour trouver le fichier à inclure .le nouveau fichier apparaît dans un nouvel onglet de la fenêtre du programme.

II.4.3.4. Menu outil [8]

La figure 29 montre le menu outil

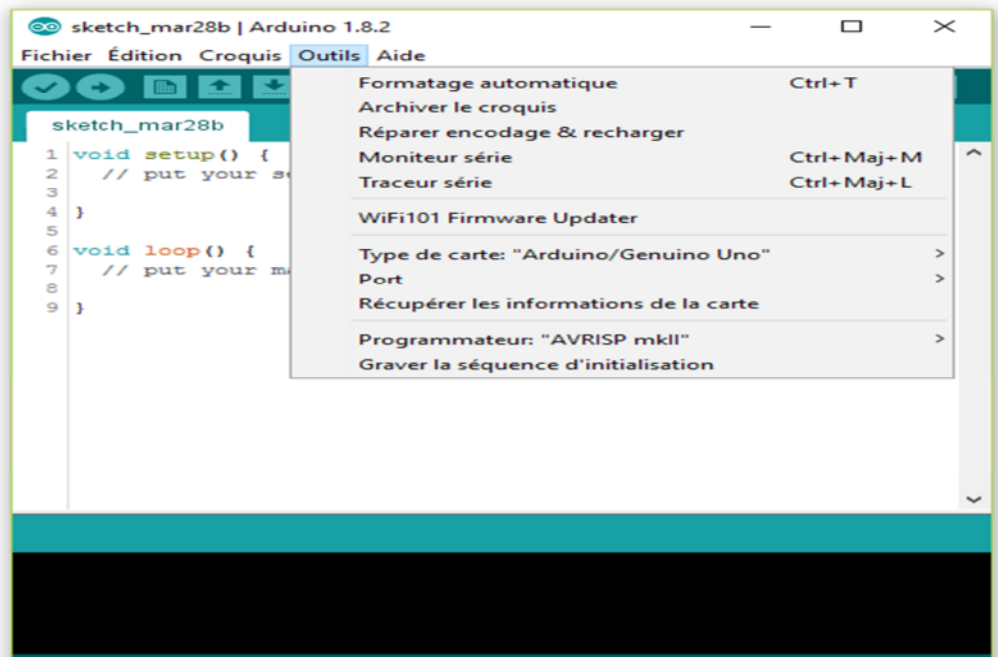


Figure29 :Menu outils

Formatage automatique : permet de formater automatiquement un programme et de le présenter de façon agréable en indentant automatiquement ce qui se trouve entre accolades .

Archiver le croquis : crée une copie du programme sous forme de fichier .ZIP et le place dans le dossier de croquis Arduino. la date est ajoutée au nom de fichier.

Réparer encodage et recharge : corriger les écarts possibles entre l'encodage de caractères de l'éditeur et celui d'autres systèmes d'exploitation.

Moniteur série : permet d'ouvrir le moniteur de l'IDE qui est une fenêtre dans laquelle on peut faire écrire des résultats par Arduino ou bien dans laquelle on peut entrer des données à envoyer à Arduino (dans le cadre jaune).

Traceur série : permet d'ouvrir le traceur de l'IDE qui est une fenêtre dans laquelle on peut faire tracer des courbes par Arduino .cette fenêtre s'utilise comme celle de moniteur avec l'ordre Serial.println pour envoyer la valeur à tracer.

Type de carte > : permet de choisir avec quel type de carte on travaille (UNO ,MEGA ,NANO etc.).

Port> : permet de régler le port sur lequel est connecté la carte .

Récupérer les information de la carte : permet d'obtenir des informations concernant la carte connectée en USB comme numéro de la carte, un numéro de série...

Programmateur> : propose plusieurs types de programmeurs pour programmer la carte.de base,l'option est « AVRISP mkII » pour programmer un module Arduino.

Graver la séquence d'initialisation : permet de graver un bootloader dans le microcontrôleur d'une carte Arduino.Ceci n'est pas requis pour l'usage normal d'une carte Arduino mais peut être utile si vous acheter un microcontrôleur vierge, livré sans bootloader.

II.4.4. Syntaxe du langage Arduino

Explication de chaque commande de la syntaxe Arduino dont la table des matières. Chaque construction est suivie de sa traduction entre parenthèses.

Commandes de structure de programme	
Structure générale	<ul style="list-style-type: none"> • Viodsetup () (configuration-préparation) • Viodloop () (exécution)
Contrôle et Conditions	<ul style="list-style-type: none"> • If (si....) • If.....else si ..alors ...) • For (pour...) • Switch case (dans le cas ou ...)
	<ul style="list-style-type: none"> • == (équivalent à) • != (différent de)

Opération de comparaison	<ul style="list-style-type: none"> • < (inférieur à) • > (supérieur à) • <== (inférieur ou égale à) • >= (supérieur ou égale à)
Opérations booléennes	<ul style="list-style-type: none"> • && (et) • ! (ou) • (et pas)
Autres commandes	<ul style="list-style-type: none"> • // (commentaire simple ligne) • /* */ (commantaire multilignes • #define (donner une valeur à un nom)

II.5. Discussion

Ce chapitre donne une vue globale sur la plateforme Arduino, commençant par un petit aperçu sur son historique , les différents composants qui constituent la carte ainsi que leurs rôles, On a parlé de l'interface de programmation IDE , de ces fonctions et instructions.

Le chapitre suivant sera consacré pour la réalisation pratique du système d'alarme.

III.1 Préambule

Dans les chapitres précédents, nous avons défini les différents composants qui constituent un système d'alarme intelligent ainsi que le logiciel Arduino avec ces fonctions et ses différentes instructions. Dans ce chapitre, nous allons passer à la conception et à la réalisation de notre application. Dans un premier, nous expliquons les différents composants constituant le système de télésurveillance. Ensuite, nous allons procéder aux tests de ces composants pour vérifier leur bon fonctionnement. Dans un second temps, nous allons réaliser les différents brochages et allons implémenter les programmes élaborés pour la mise en marche du système de télésurveillance.

III.2. Réalisation matérielle**III.2.1 Composition d'un système d'alarme**

Pour réaliser notre système d'alarme on a utilisé différents éléments (une carte Arduino, module GSM, un buzzer, une led et une plaque d'essai).

Nous avons commencé par tester le module GSM pour s'assurer de son bon fonctionnement. Pour cela, nous avons placé une puce (n'importe quel réseau) et le brancher avec la carte Arduino qui contient le programme permettant d'envoyer et de recevoir les messages.

De la même façon, nous avons testé le capteur PIR. Nous avons donc connecté une led qui permet d'indiquer le bon fonctionnement du capteur.

Après ces tests, nous avons réalisé le circuit du système d'alarme en branchant les différents éléments sur une plaque d'essai. Pour ce faire, nous avons opéré comme suit :(figure30)

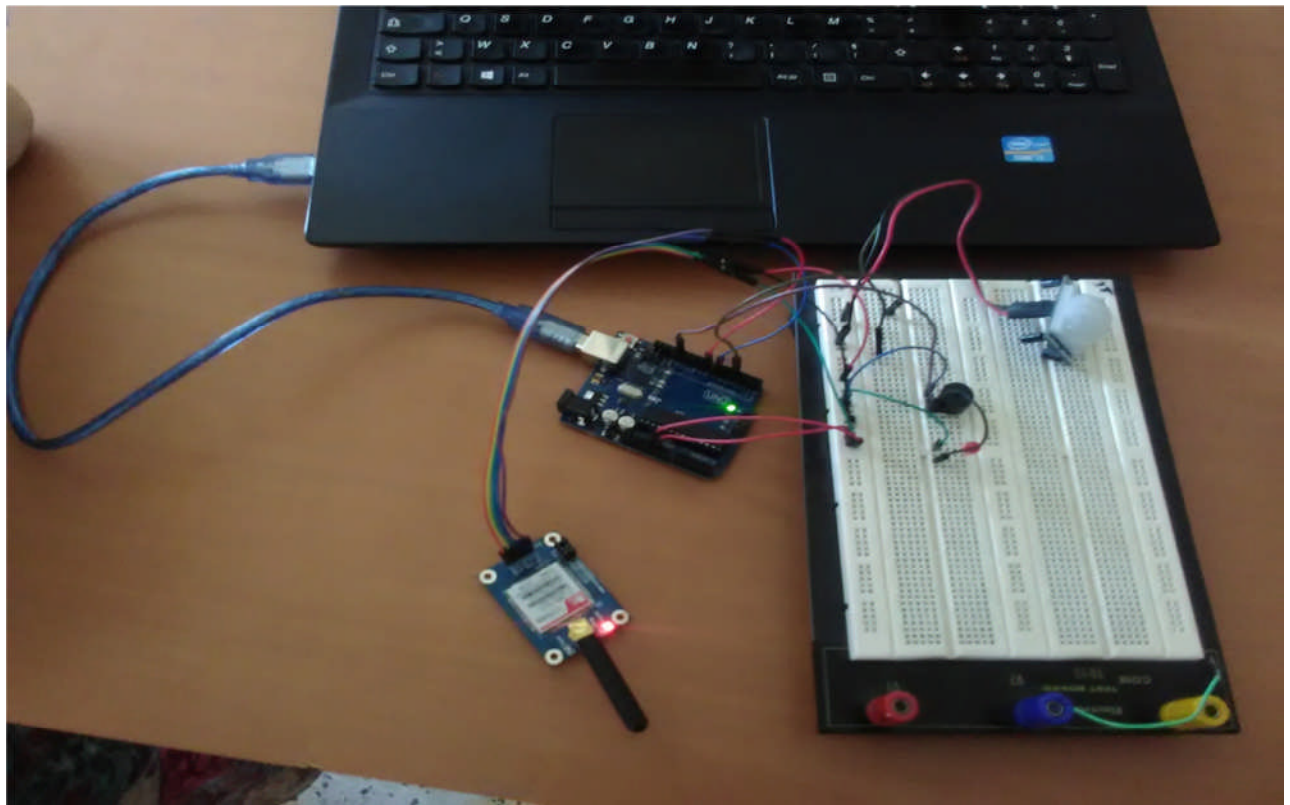


Figure30 : Branchement des différents éléments

La figure suivante représente le branchement des composants de système sur ISIS Proteus

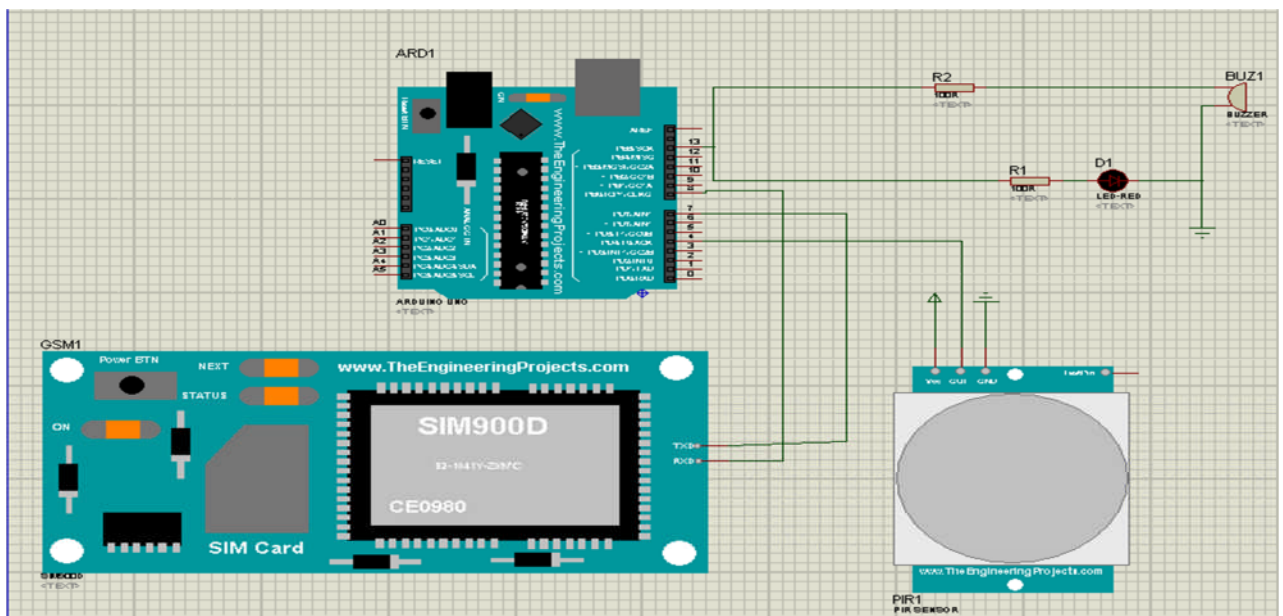


Figure 31:Branchements des éléments sur ISIS Proteus

III.2.1.1 Branchement du capteur PIR à la carte Arduino [9]

le branchement du capteur PIR sur une carte Arduino UNO est plus simple. ces deux éléments sont alimentés en 5 v (ou 4.5V) . il suffit de brancher les bornes d'alimentation +/- de la carte vers les bornes +/- du capteur.

Le signal de sortie du capteur (fil jaune) doit être relié à l'entrée « digital » de la carte Arduino .comme illustre la figure suivante :

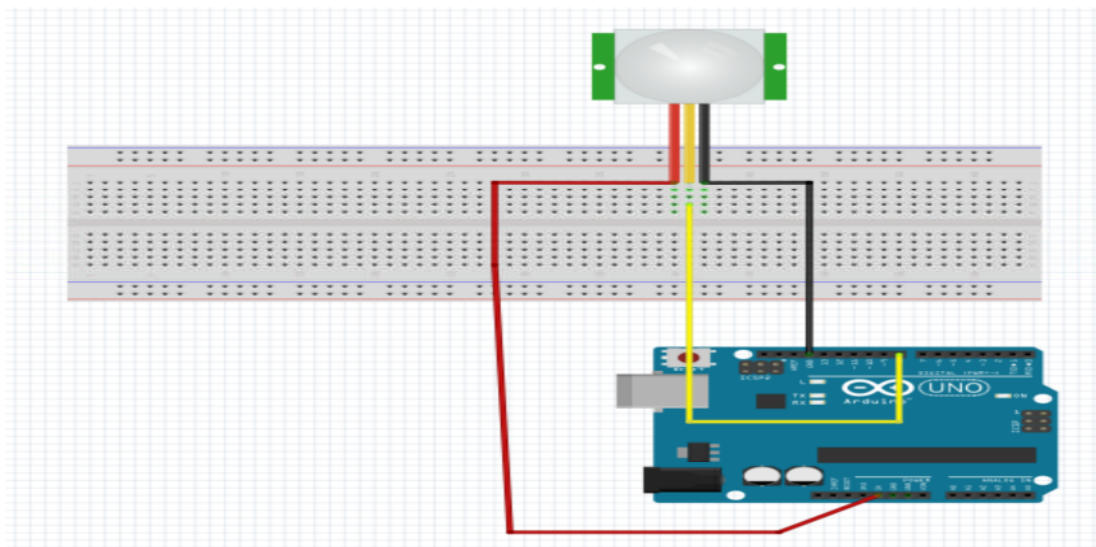


Figure32 : Branchement de capteur PIR

III.2.1. Branchement du module GSM :

La figure XX ci- dessous montre le branchement du module GSM à la carte Arduino , les deux masses de module GSM sont reliées vers la masse de l'Arduino.

La borne TX (transmission) de module GSM est branché vers la pin 7(réception) de l'Arduino et la borne RX (réception) du GSM est reliée avec la pin 8(transmission) comme Le montre la figure suivante :

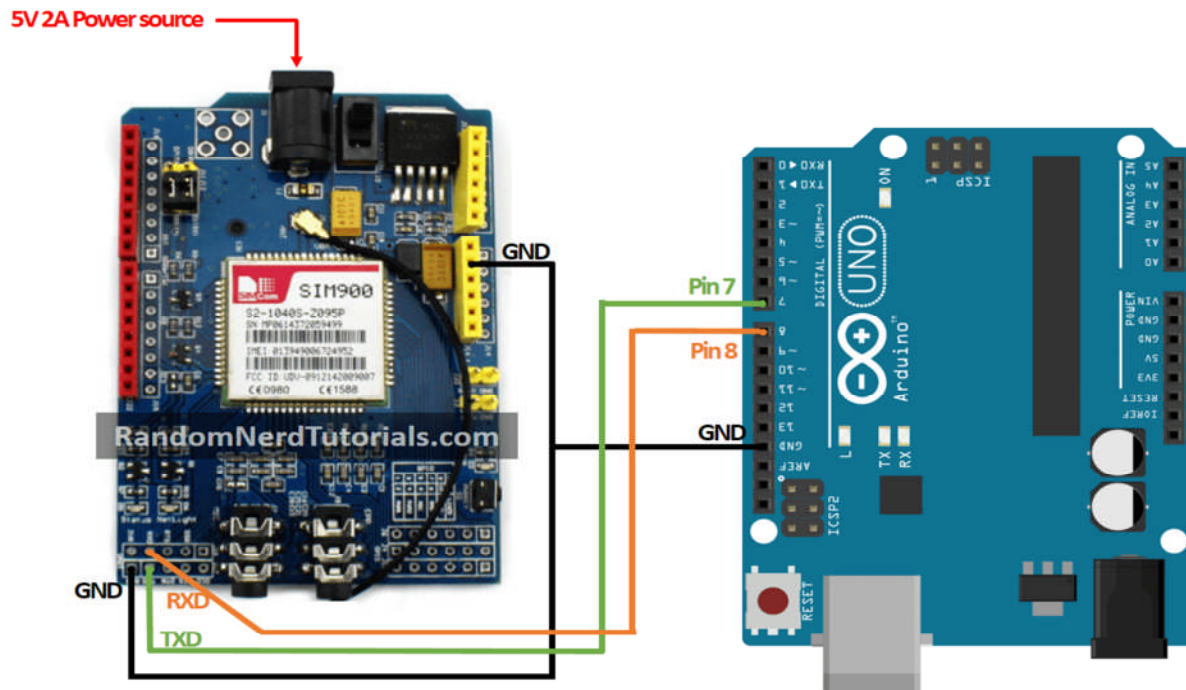


Figure 33 :Branchement de GSM à la carte Arduino

III.2.1.3.Branchement de BUZZER à la carte Arduino [9]

La pate du buzzer est branchée au pin 13 et l'autre reliée à la masse de la carte Arduino (Voir figure 34).

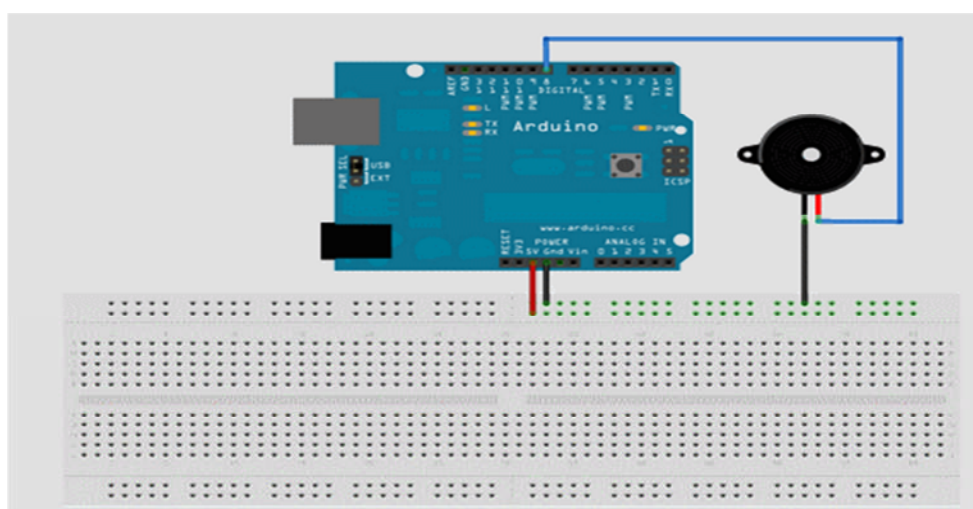


Figure 34 : Branchement de buzzer à la carte Arduino

III.2.1.4. Branchement de la LED à la carte Arduino [9]

La led est alimentée en série avec une résistance de protection .l'anode de la led est reliée à la pin 13 de la carte Arduino.

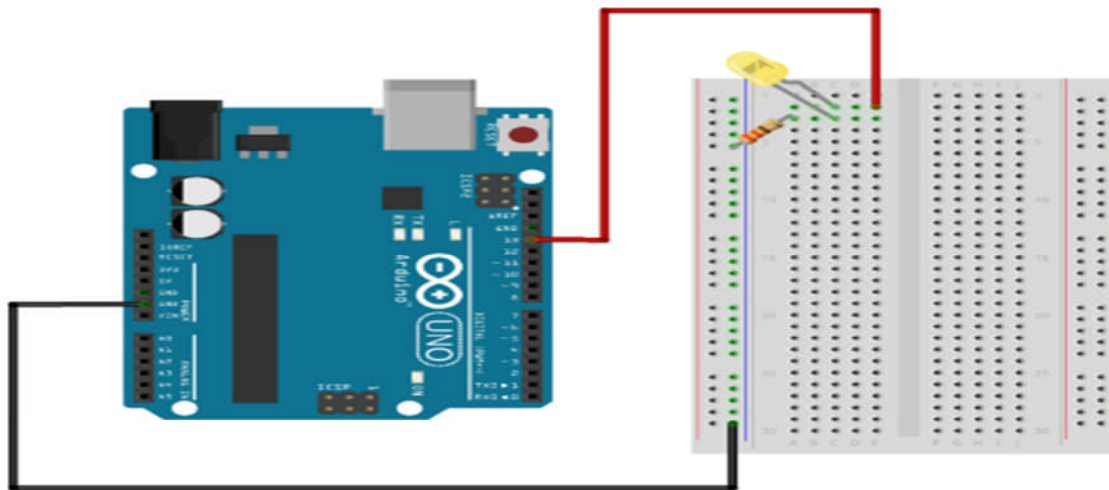


Figure 35 : Branchement de la led à la carte Arduino

III.3. Réalisation logiciel

III.3.1. Programmation de l'Arduino

pour programmer l'Arduino, on doit la brancher à un ordinateur via un câble USB (fiche A fiche B) .(voir la figure36),puis sur l'ordinateur .on a besoin d'installer l'environnement de développement Arduino IDE (Integrate Developmen Environemt) qui vas nous permettre de programmer la carte ARDUINO .



Figure 36 :(câble USB fiche A fiche B)

L'IDE de l'Arduino en faite, il s'agit d'un compilateur .alors qu'est-ce que c'est exactement ?

- **Un compilateur**

Un compilateur est un terme qui désigne un logiciel qui est capable de traduire un langage informatique, ou plutôt un programme utilisant un langage informatique ,vers un langage plus approprié afin que la machine qui vas le lire puisse le comprendre .le compilateur doit donc traduire les instructions du programme ,écrites en langage texte ,vers vers un langage dit « machine ».ce langage utilise uniquement des 0et des 1ce la peut se traduire de la façons suivante :

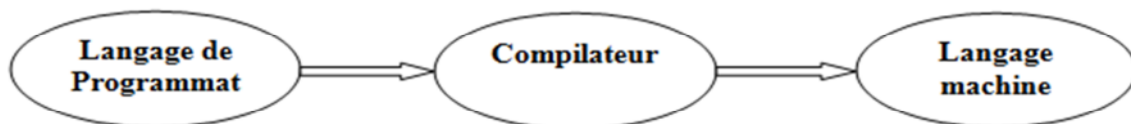


Figure 37 : Traduction d'un langage de programmation en langage machine

III.3.2.Présentation de logiciel

La figure 38 représente l'interface de l'IDE d'Arduino.

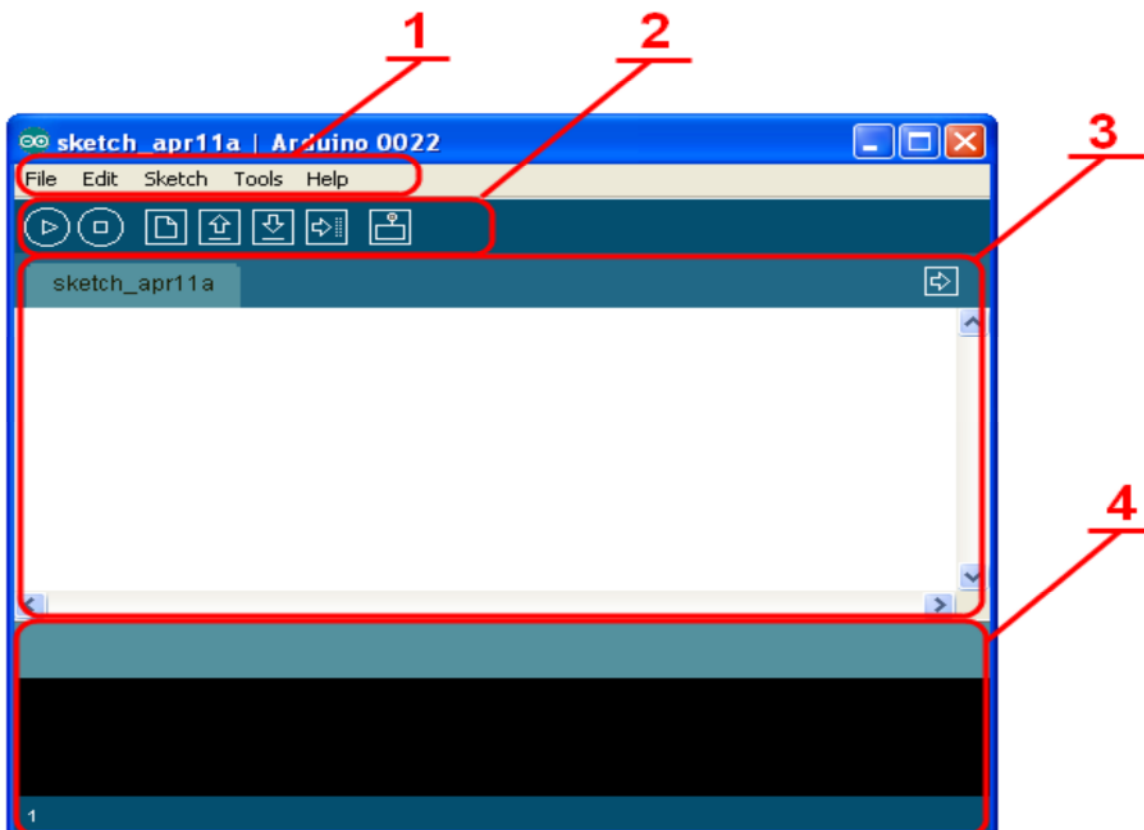


Figure 38 :Interface de l'environnement de développement Arduino IDE

cadre numéro 1 : ce sont les options de configuration logiciel.

cadre numéro 2 : il contient les boutons qui vont nous servir lorsque l'on va programmer nos cartes.

cadre numéro 3 : ce bloc va contenir le programme que nous allons créer .

cadre numéro 4 : celui-ci est important ,car il va nous aider à corriger les fautes dans notre programme .c'est le débogueur.

III.3.3 Choix de la carte que l'on va programmer

Le nom de la carte est indiqué sur elle .pour nous il s'agit de la carte « UNO » . dans le menu , « tools » (outils) on sélectionne « board » (carte) puis il faut vérifier que c'est bien le nom ARDUINO UNO qui est coché. figure (39)

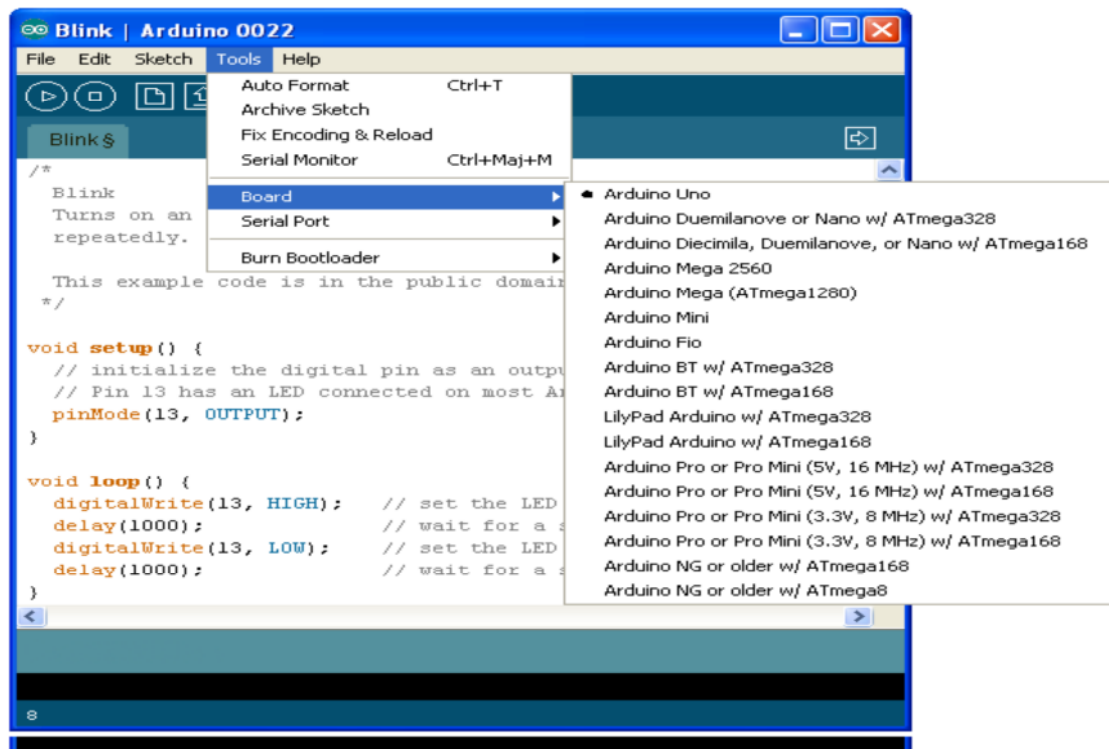


Figure39: Choix de la carte ARDUINO

III.3.4 Choix de port de connexion de la carte

Nous allons dans le menu Tools ,puis serial port .là nous choisissons le port COMX,X étant le numéro de port qui est affiché .nous ne choisissons pas COM1 car il n'est quasiment jamais connecté à la carte .dans notre cas il s'agit de COM5 .figure(40).

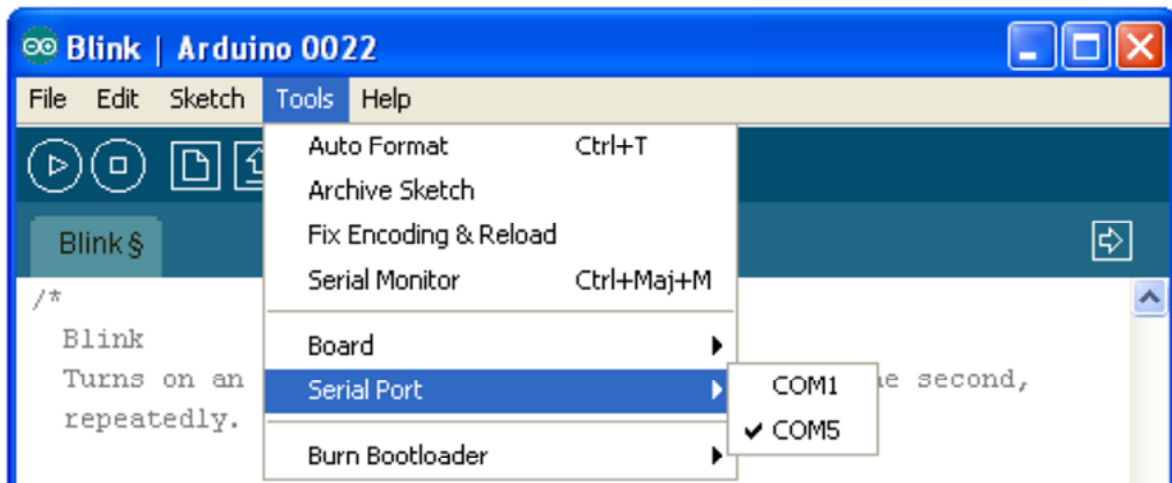


Figure 40 : Choix de port de connexion de la carte

III.3.5 les fonctions des icônes du logiciel IDE

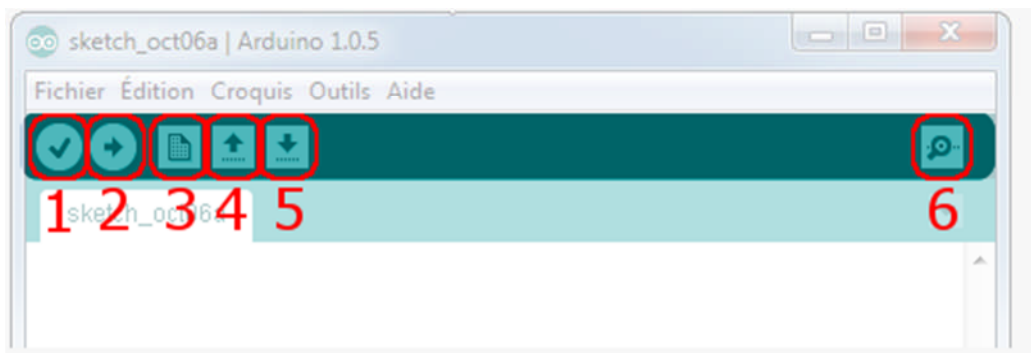


Figure41:Les icônes de l'IDE d'Arduino.

Bouton 1 : permet de vérifier le programme, il actionne un module qui cherche les erreurs de syntaxes dans le programme.

Bouton 2 : charge (téléverser) le programme dans la carte Arduino.

Bouton 3 : crée un nouveau croquis.

Bouton 4 : ouvrir un croquis déjà existant.

Bouton 5 : enregistre le fichier courant.

Bouton **6** : ouvre le moniteur série, qui permet la communication entre la carte Arduino et l'ordinateur à lequel on l'a branchée.

Pour programmer n'importe quelle carte Arduino il faut suivre les étapes suivantes :

- Allez dans le menu outils, ensuite cliquer sur type de carte et puis sélectionner la carte dont on a besoin.
- Ensuite il faut s'assurer qu'on est connecté au bon port série .il nous faut donc aller dans le menu outils puis port et en fin port adéquate.
- Par la suite il nous reste plus qu'à programmer l'application que l'on souhaite réaliser. Une fois le programme terminé, La prochaine étape, il va falloir envoyer le programme dans la carte .pour ce faire, il suffit de cliquer sur le bouton Upload ou « télécharger » .

Finalement le processus de rédaction de programme jusqu'à son téléchargement dans la carte peut être résumé grâce au schéma suivant :

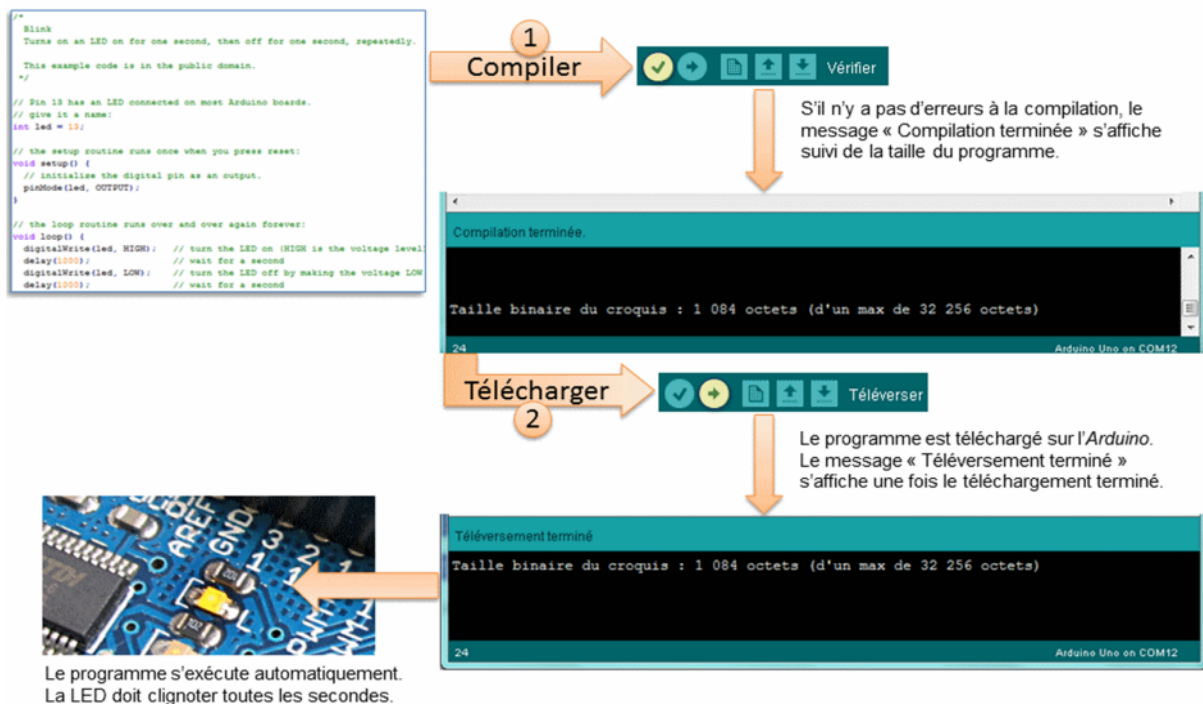


Figure42 : Interface de compilation et téléchargement

III.4. Tests de l'application

Pour vérifier le fonctionnement de notre application, nous avons réalisé des tests :

En présence d'un corps émettant de l'infrarouge, le capteur PIR détecte ces radiations infrarouges et transmet l'information au système d'alarme. Ce dernier envoie automatiquement une alerte sous forme d'un message (SMS) à travers le réseau GSM pour avertir l'utilisateur. Comme illustre les figures suivantes :

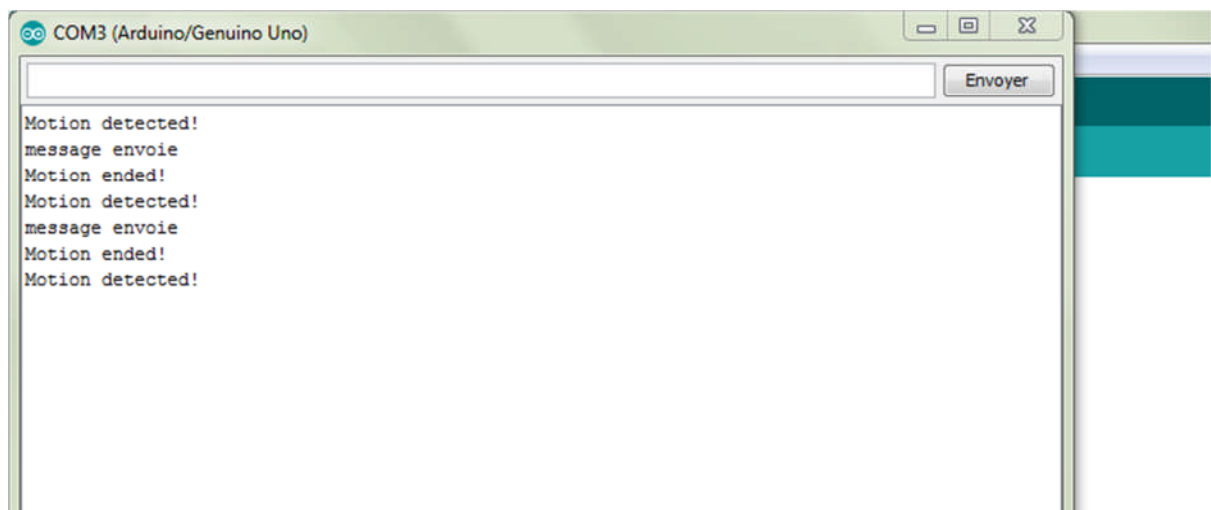


figure 43: Fenêtre sur de logiciel lors de détection d'une présence

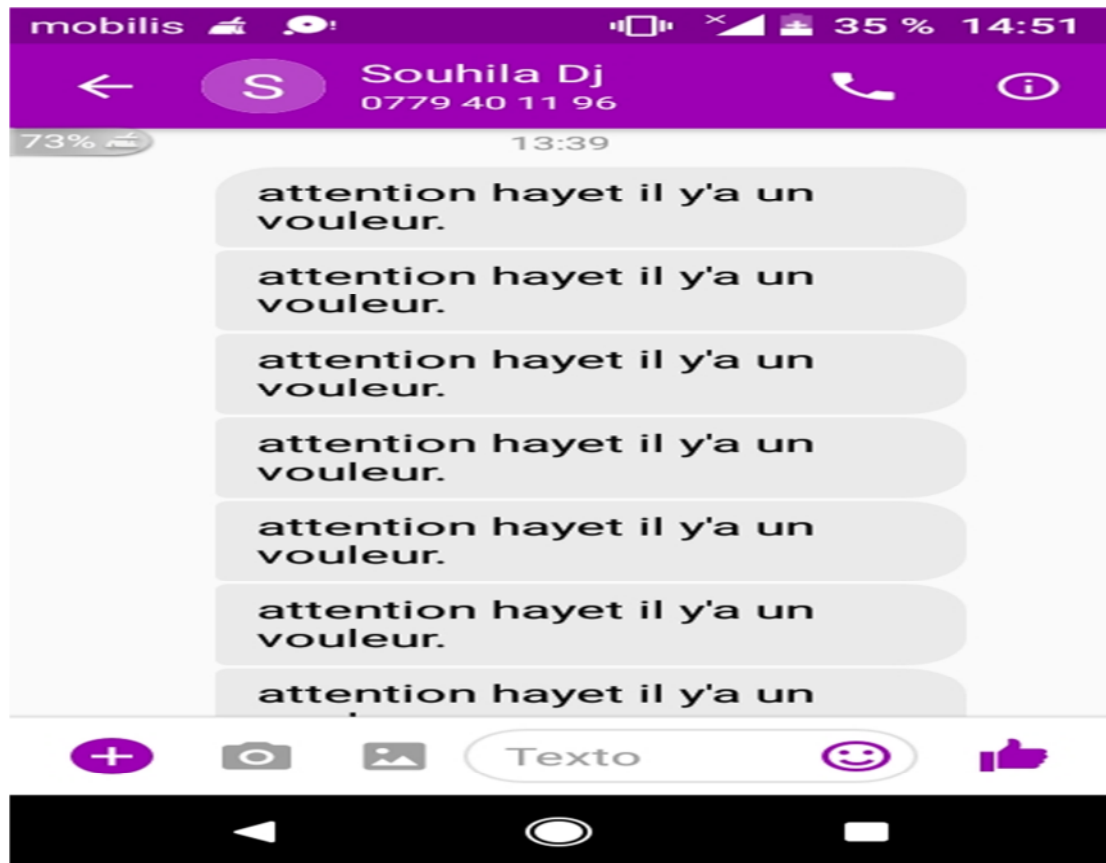


Figure44 :Message envoyé

En revanche, en absence de source de chaleur, le capteur PIR ne détecte aucune radiation. Par conséquent, le système n'émet pas un sms.



Figure 45: Fenêtre de logiciel en absence de mouvement

III.5. Discussion

Ce troisième chapitre est consacré à la réalisation pratique de notre système d'alarme. Nous avons expliqué la procédure à suivre pour choisir le port de connexion, le type de la carte et comment se fait le raccordement entre les différents composants et nous avons effectué quelques tests pour vérifier le bon fonctionnement de notre application.

Conclusion

Conclusion

Dans ce travail, nous avons réalisé une application dont l'objectif est la détection de présence en se servant d'un capteur infrarouge pour prévenir contre toute intrusion.

Dans le premier chapitre, nous avons présenté les applications de la domotique, les éléments constituant le système d'alarme qui sont le capteur infrarouge, le module GSM et leurs caractéristiques.

Dans le deuxième chapitre, nous avons présenté le modèle de la carte Arduino que nous avons utilisé (UNO), ces caractéristiques ainsi les deux parties logiciel et matériel.

Le dernier chapitre a été consacré à la réalisation d'un système d'alarme à base d'une carte d'acquisition Arduino.

Après la détection de la présence via le capteur PIR, l'information est transmise en utilisant le module GSM. La carte Arduino est utilisée pour interfacier entre le capteur PIR et le module GSM. L'utilisateur est notifié par un SMS généré par le système réalisé.

Selon les tests que nous avons réalisés, en présence d'une source de chaleur, le système a permis de détecter les radiations émises par cette chaleur, et ensuite un sms est généré et transmis à l'utilisateur.

En perspectives, il serait intéressant d'augmenter le nombre de capteurs pour couvrir un champ de surveillance plus large. Pour une observation plus détaillée, il serait aussi intéressant de doter le système par une camera afin de transmettre de la vidéos.

Bibliographie

- [1] : H.HAMOUCHE, thèse de doctorat, l'université Mouhamed V,rabat,juillet 2015
- [2] : Dominique Maniez, les capteurs pour Arduino et raspberry PI, 121, 105, 2014,06/2018
- [3] :P.Crouzet, thèse de doctorat, l'université Aix-Marseille, 2009,
- [4] : Xavier Lagrange,Philippe,Sammi tabbane , “Réseaux GSM“ ,115 ,116 ,117
119 1996,05/2018
- [5]: H.Bndjafer, A.Midoun, master, université Aboubeker Belkaid, Telemcen,2016
- [6]: http://f-leb.developpez.com/tutoriels/arduino/univers_arduino/part1
- [7] :Arduino : première pas en informatique embarquée de Simon Landraut et HippolyteWeisslinger.06/2018
- [8] :www.linuxedu.tetaneul.net
- [9] :www.fritzing.org

Résumé :

Dans ce travail, nous avons réalisé une application dont l'objectif est la détection de présence en se servant d'un capteur infrarouge pour prévenir contre toute intrusion.

Dans le premier chapitre, nous avons présenté les applications de la domotique, les éléments constituant le système d'alarme qui sont le capteur infrarouge, le module GSM et leurs caractéristiques.

Dans le deuxième chapitre, nous avons présenté le modèle de la carte Arduino que nous avons utilisé (UNO), ces caractéristiques ainsi les deux parties logiciel et matériel.

Le dernier chapitre a été consacré à la réalisation d'un système d'alarme à base d'une carte d'acquisition Arduino.

Après la détection de la présence via le capteur PIR, l'information est transmise en utilisant le module GSM. La carte Arduino est utilisée pour interfacier entre le capteur PIR et le module GSM. L'utilisateur est notifié par un SMS généré par le système réalisé.

Selon les tests que nous avons réalisés, en présence d'une source de chaleur, le système a permis de détecter les radiations émises par cette chaleur, et ensuite un sms est généré et transmis à l'utilisateur.

En perspectives, il serait intéressant d'augmenter le nombre de capteurs pour couvrir un champ de surveillance plus large. Pour une observation plus détaillée, il serait aussi intéressant de doter le système par une camera afin de transmettre de la vidéos.^

Nous avons défini les différents composants qui constituent un système d'alarme intelligent ainsi que le logiciel Arduino avec ces fonctions et ses différentes instructions.

Nous allons réaliser les différents brochages et allons implémenter les programmes élaborés pour la mise en marche du système de télésurveillance.

Pour réaliser notre système d'alarme on a utilisé différents éléments (une carte Arduino, module GSM, un buzzer, une led et une plaque d'essai).

Nous avons commencé par tester le module GSM pour s'assurer de son bon fonctionnement. Pour cela, nous avons placé une puce (n'importe quel réseau) et le brancher avec la carte Arduino qui contient le programme permettant d'envoyer et de recevoir les messages.

De la même façon, nous avons testé le capteur PIR. Nous avons donc connecté une LED qui permet d'indiquer le bon fonctionnement du capteur.

Après ces tests, nous avons réalisé le circuit du système d'alarme en branchant les différents éléments sur une plaque d'essai.

Dans le troisième chapitre est consacré à la réalisation pratique de notre système d'alarme. Nous avons expliqué la procédure à suivre pour choisir le port de connexion, le type de la carte et comment se fait le raccordement entre les différents composants et nous avons effectué quelques tests pour vérifier le bon fonctionnement de notre application.