République **A**lgérienne **D**émocratique et **P**opulaire

Ministère de L'**E**nseignement **S**upérieure et de la **R**echerche **S**cientifique

UNIVERSITE **M**OULOUD **M**AMMERI DE **T**IZI-**O**UZOU

FACULTE DE **G**ENIE **E**LECTRIQUE ET D'**I**NFORMATIQUE
DEPARTEMENT D'**E**LECTRONIQUE

Mémoire de fin d'Etude

de MASTER PROFESSIONNEL

Domaine : **Sciences et Technologies**

Filière : **Génie électrique**

Spécialité : **Electronique Industrielle**

Thème

Réalisation d'un téléphone portable à base de la carte Arduino UNO

Proposé par :

Mr. LAZRI

Dirigé par :

Mr. OUALOUCHE

Mr. HAMAGUE

Présenté par :

M^{elle} DRIS Dyhia

M^{elle} ZERAR Katia

Année universitaire: 2017/2018

Remerciements

Nous remercions tout d'abord par excellence sa grandeur « le bon dieu » qui nous a donné le courage et la patience tout au long de notre vie.

Nous remercions notre promoteur Mr. LAZRI pour son aide, son orientation et ses conseils durant l'accomplissement du projet.

Nous remercions également les membres du jury d'avoir accepté de juger ce travail de fin d'études.

Tous nos infinis remerciements vont à tous les enseignants qui ont contribué à notre formation notre cursus universitaire.

Nos remerciements les plus chaleureux vont à nos chers parents pour leurs encouragements, leur patience et leur grand soutien durant toutes ces années d'études.

Enfin, nous remerciements pour tous ceux qui ont participé de près ou de loin à la réalisation de ce mémoire

Dédicaces

Je dédie ce travail :

A mes très chers parents pour leur bienveillance et leurs soutiens tout au long de mon parcours scolaire et universitaire.

A mon cher frère « Lounes ».

A ma chère sœur « Kamilia ».

A mon adorable « Katia » avec qui j'ai partagé ce travail, ainsi à toute sa famille.

A tous mes amis et proches.

« Dyhia »

Dédicaces

Je dédie ce travail :

En premier lieu à la mémoire de mon père que dieu le recueillera dans son vaste paradis.

A ma très chère mère qui ma soutenue le long de ma vie en témoignage de leur affection et sacrifices.

A mes chers frères «Sofiane et Walid ».

A ma chère sœur « Siham ».

A mon adorable « Dyhia » avec qui j'ai partagé ce travail, ainsi à toute sa famille.

A tous mes amis et proches.

« Katia »

GLOSSAIRE

AT: Terminal Adapter.

ATA: Answer An Incoming Call.

ATD: Mobile Originated Call to Dial A Number.

ATH: Disconnect Existing Connection.

AUC: Authentification center.

BTS: Base Transceiver Station.

BSC: Base Station Controller.

CMGF: Select SMS Message Format.

CMGR: Read SMS Message.

CMGS: Send SMS Message.

CNMI: New SMS message Indication.

EDGE: Enchanced Data rate for GSM Evolution.

EIR: Equipment Identity Systéme.

FTDI: Future Technology Device International.

GND: Grond.

GPRS: General Packet Radio Service.

GSM: Global System for Mobile Communication.

HLR: Home Location Register.

IDE: Integrated Development Environment.

LCD: Liquid Crystal Display.

LAPD: Link Access Protocol D-Channel.

MIC: Modulation impulsion code

MSC: Mobile Switching Center.

PWM: Pulse-Width Modulation.

RX: Receive.

RAM: Random Access Memory**.**

SMT: Surface Mount Technology.

TX: Send.

TDMA: Time Division Multiple Access.

TMSI: Temporary Mobile Subscriber Identity.

RTC: Reseau Téléphonique Commuté.

RNIS: Reseau Numerique à Intégration de Service.

TTL: Time TO live.

UART: Universal Asynchronous Receiver Transmitter.

UMTS: Universal Mobile Telecommunication Systeme**.**

USB: Universal Serial Bus.

WAP: Wireles Application Protocol**.**

Sommaire

Chapitre III : Réalisation de notre application

Liste des figures

Liste des tableaux

INTRODUCTION

Introduction

Le téléphone est un appareil de communication initialement conçu pour transmettre la voix humaine et pouvoir communiquer à distance pour fonctionner, le téléphone nécessite une infrastructure terrestre ou spatiale, le réseau téléphonique. Après y avoir raccordé son terminal fixe ou avoir allumé son appareil mobile, l'utilisateur ayant souscrit un abonnement auprès d'un opérateur de télécommunications peut passer un appel téléphonique à un destinataire également raccordé en composant son numéro attitré, ce qui déclenche généralement la sonnerie de l'appareil de destination. Si la personne appelée accepte l'appel une conversation téléphonique peut commencer.

Exception faite des appels d'urgence passés à des numéros spéciaux, les appels passés via un téléphone ont un cout, lequel est déterminé par des tarifs d'appel établis en fonction de leur durée, de la localisation du destinataire. La deuxième génération de réseaux de télécommunication GSM a connu un développement spectaculaire. Cette révolution, après celle du réseau analogique a su se faire apprécier du grand public. Au début des années 1990, la norme GSM est adoptée en Europe, depuis sa couverture est quasi mondiale, ses services de plus en plus nombreux utiles et conformes aux désirs des abonnés. La principale caractéristique de cette norme est la mobilité native d'un téléphone GSM qui permet à tout abonné mobile de transmettre ou de recevoir des appels comme s'il se trouvait sur son site d'origine, contrairement aux réseaux fixes traditionnels.

Un réseau de téléphonie mobile a une structure « cellulaire » qui permet de réutiliser de nombreuses fois les mêmes fréquences, il permet aussi à ses utilisateurs en mouvement de changer de cellule sans coupure des communications en cours. Dans un même pays, aux heures d'affluence plusieurs centaines de milliers, voire plusieurs millions d'appareils sont en service avec (dans le cas de GSM) seulement 500 canaux disponibles.[1]

Dans notre projet de fin d'étude, nous réalisons un téléphone portable simple constitué de différents éléments, principalement la carte GSM SIM900 et la carte Arduino.

Notre mémoire est composé de trois chapitres :

Dans le premier chapitre, nous présentons le module GSM utilisé pour la réalisation des téléphones portables et nous décrivons aussi dans ce chapitre l'architecture du réseau téléphonique GSM.

Le deuxième chapitre contient la description de la carte Arduino UNO.

Dans le dernier chapitre, nous allons réaliser notre système, à savoir le téléphone portable, commençant par la schématisation de circuit synoptique, puis nous allons concevoir le circuit électronique tout en montrant son fonctionnement aussi bien pour la partie matérielle que pour la partie logicielle.

Nous terminons notre travail par une conclusion.

CHAPITRE I

Généralités sur le réseau GSM

I.1 Préambule:

La communication mobile existe depuis un demi-siècle, mais c'est dans les années 1980 qu'elle s'est vraiment développée. L'un des premiers systèmes de la communication mobile est le GSM.

C'est la première norme de la téléphonie cellulaire de second génération qui soit pleinement numérique, elle constitue désormais la référence mondiale pour les systèmes radios mobiles. Le GSM offre à ses abonnés des services qui permettent la communication de station mobile de bout en bout à travers le réseau, la téléphonie est la plus importante des services offerts. Ce réseau permet la communication entre deux postes mobiles où entre un poste mobile et un poste fixe. Les autres services proposés sont la transmission de données et la transmission des messages alphanumériques courts. L'objectif donc du chapitre est de faire une description de cette norme qui est le GSM, après un historique sur le GSM et son architecture du réseau.

I.2 L'historique :

Durant des siècles, l'homme s'est contenté de la parole ou des écrits comme seuls moyens de communication entre deux personnes éloignées d'une distance importante.

En 1982, lors de la Conférence Européenne des Postes et Télécommunications (CEPT) que fut créé le Groupe Spécial Mobile (GSM).

En 1985, la Commission Européenne annonce l'imposition de la norme issue du GSM.

En 1987, le choix est arrêté sur la transmission numérique AMR.

En 1989, les travaux du Groupe Spécial Mobile "GSM" sont transférés au comité "SMG" de l'Européen Télécommunication Standards Institute (ETSI), qui poursuit les tâches de normalisations. Notons que c'est cette comité qui mettra au point le module d'identité d'abonné SIM.

En 1991 fût réalisé la première communication entre un mobile et un abonné fixe.

Et c'est en 1992 que fût ouvert le système GSM ITINERIS de France Telecom, rejoint plus tard par SFR du groupe Cegetel et par Bouygues Telecom (1994).

L'explosion du marché des mobiles, sa croissance soutenue et l'apparition de nouveaux services amènent les réseaux GSM actuels à leur limite. Le débit de 9,6 kb/s, défini à l'origine, est insuffisant pour couvrir les nouveaux besoins de transferts de données et constitue un frein à la diffusion de contenus multimédias.

I.3 Présentation de la carte GSM :

I.3.1 Définition :

Le GSM (Global System for Mobile communications), est un système cellulaire et numérique de télécommunication mobile. Il a été rapidement accepté et a vite gagné des parts de marché. L'utilisation du numérique pour transmettre les données permet des services et des possibilités élaborées par rapport à tout ce qui a existé. On peut citer, par exemple, la possibilité de téléphoner depuis n'importe quel réseau GSM dans le monde.

Les modules GSM sont fabriqués par différentes sociétés, ils ont tous des spécifications d'alimentation d'entrée différentes. Dans ce tutoriel, notre module GSM nécessite une entrée de 12 volts et 1A. Nous avons vu des modules GSM qui nécessitent 15 volts et d'autres qui n'ont besoin que de 5 volts. Ils diffèrent avec les fabricants. Les GSM qui nécessitent une entrée de 5V, peuvent s’alimenter directement à partir de la sortie 5V de la carte Arduino. Comme l’indique la figure suivante :

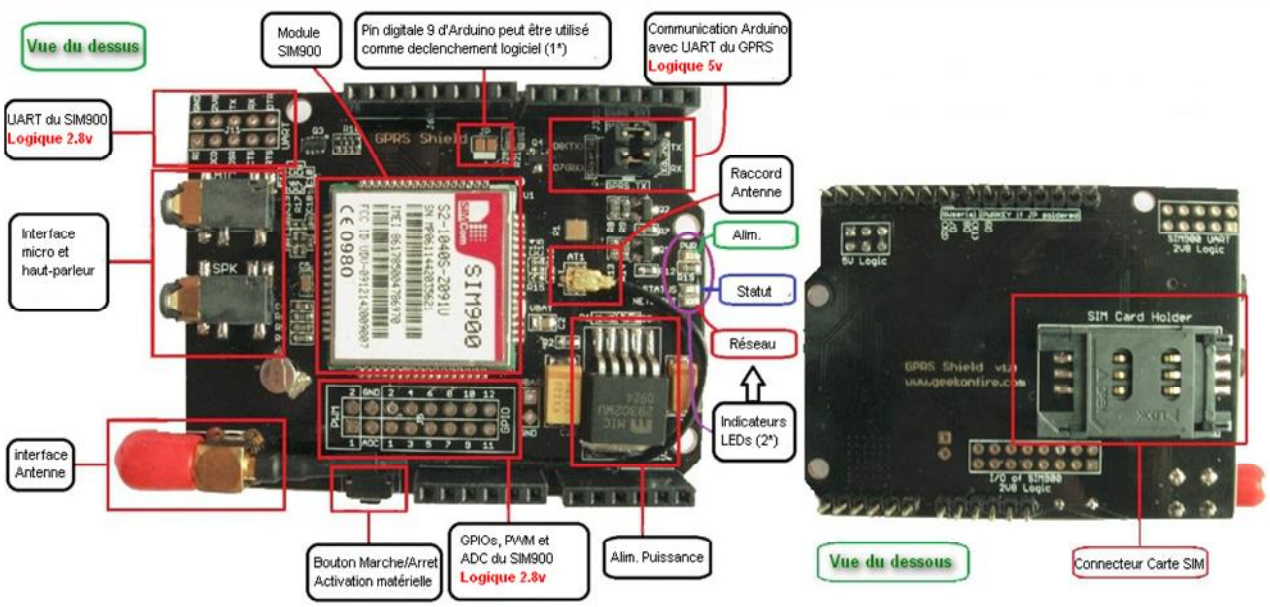


Figure I.1. Présentation de la carte GSM sim900

I.3.2 Les composants de la carte GSM :

La carte GSM constituée de plusieurs composants :

- **Module SIM900 :** SIM Com présente un module sans fil ultra compact et fiable-SIM900. Il s'agit d'un module GSM / GPRS quadri bande complet de type SMT et conçu avec un processeur mono puce très puissant intégrant le cœur ARM926EJ-S, nous permettant de bénéficier de petites dimensions et économiques et de solution rentables.

  Doté d'une interface standard, le SIM900 offre des performances GSM / GPRS 850/900/1800 / 1900MHz pour la voix, les SMS, les données et le fax dans un format réduit et avec une faible consommation d'énergie. Avec une configuration minuscule, le SIM900 peut répondre à presque tous les besoins d'espace dans nos applications, en particulier pour les applications minces et compactes de conception. Et la figure suivante nous indique la forme du module SIM900.

Figure I.2. SIMcom900

- **Le microphone :** Autrefois les micros des téléphones étaient des microphones appelés dynamiques qui étaient équipés d'une membrane, d'une bobine métallique et d'un aimant. La bobine vibrait de la même manière que la membrane et créait donc des perturbations dans le champ magnétique de l'aimant, or toutes variations d'un flux magnétique induit un courant électrique, on peut donc utiliser ce courant et le transmettre. Puis, avec le temps le système s'est amélioré afin d'être le plus compact possible et c'est fait appeler microphone électrostatique. Cette technique est basée sur le fonctionnement d'un condensateur. Le condensateur est composé de deux plaques polarisées (électrodes) se faisant face, séparées par l'air. La variation de distance entre les plaques polarisées va modifier la capacité du condensateur. Cette variation est facilement traduisible en courant électrique. Dans un microphone électrostatique, la membrane avec une polarisation permanente va jouer le rôle d'une des deux électrodes, tandis que l'autre électrode sera fixe. On comprend facilement que la variation de la pression sonore à la surface de la membrane va la déplacer et va donc modifier la distance entre la membrane et l'électrode fixe. Un petit système électronique est ensuite chargé de transformer les variations de capacité électrique en courant électrique. Comme l'indique la figure ci-dessous :

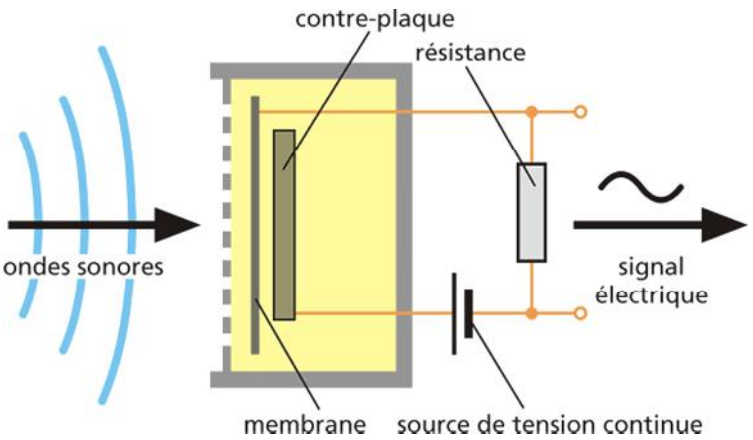


Figure I.3. Le microphone.

- **Les haut-parleurs :** qui à la différence des micros n'ont pas évolué dans leurs systèmes, c'est seulement leur taille qui a évolué. Le haut-parleur a pour but de transcrire le signal électrique en un signal sonore. Il est composé d'un aimant non mobile fixé sur le châssis du haut-parleur. Il est aussi constitué d'une bobine de cuivre montée sur un axe libre autour de l'aimant ce qui lui permet d'entrer en mouvement sans souci. En effet cette bobine reçoit le signal qu'elle transcrit en un champ magnétique et entre en mouvement par opposition ou par attirance avec l'aimant. La bobine bouge et est liée à la membrane qui crée une onde sonore.

Et Voilà pourquoi le téléphone mobile a été inventé, Au lieu d'utiliser des câbles, cette technologie va utiliser les ondes électromagnétiques qui se déplacent à la vitesse de la lumière (soit 300 000 km/sec). Lorsqu'un appel est émis, la voix va être numérisée (conversion du signal analogique en signal numérique) en une suite de 0 et de 1 qui va être gravée sur un signal analogique. Cela est appelé onde porteuses. Ces ondes porteuses sont ensuite captées par l'antenne de proximité. Evidemment, pour l'usage de nombreux utilisateurs, l'antenne fera du multiplexage (c.à.d. partager le temps de connexion des utilisateurs).

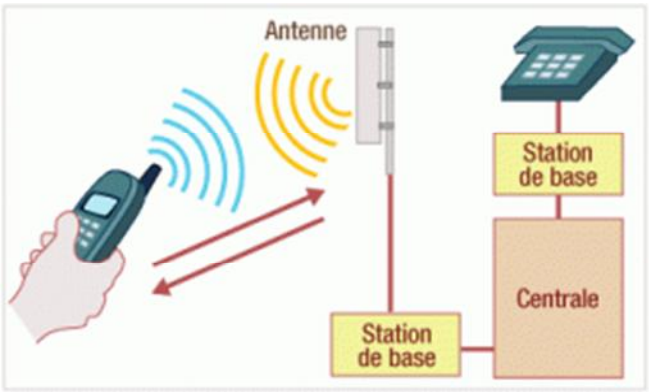


Figure I.4. La connexion de la station mobile et la station de base

- **L'antenne:** permet de transmettre le signal à une station de base (BTS) qui l'envoie alors à une centrale, par ligne téléphonique conventionnelle ou par faisceaux hertziens (ondes électromagnétiques). C'est de là que sont acheminées les conversations vers le téléphone du destinataire, selon le processus inverse, par voie filaire ou hertzienne (onde électromagnétique), suivant qu'il s'agisse d'un appel vers un téléphone fixe ou vers un portable. Le signal numérique est alors de nouveau transformé en signal analogique sonore dans l'appareil du récepteur.
- **Indicateurs LEDs :** Le shield est équipé de trois LEDs indicatrices (PWR (Vert), Status(Bleu), Netlight (Rouge)), il est possible d'identifier l'état du fonctionnement du shield à l'aide de ces trois LEDs indicatrices :
  - PWR indique si le shield est alimenté. PWR pour POWER (alimentation)
  - Status indique le Statut.
  - Netlight indique l'état de communication/connexion avec le réseau mobile.
- **Les deux supports :** un support pour une batterie de 3 voltes RTC (horloge temps réel), et l'autre pour la carte SIM au de sous de la carte GSM.
- **L'interrupteur à bascule :** Est pour sélectionner la source d'alimentation et à coté il y a une flèche indiquant la position de basculement.

I.3.3 Architecture du réseau GSM sim900 :

La carte GSM est adaptée aux réseaux GSM.

I.3.3.1 Les équipements fonctionnels du GSM : Un réseau de radiotéléphonie a pour premier rôle de permettre des communications entre abonnés mobiles et abonnés du réseau téléphonique commuté(RTC). Il s'interface avec le RTC et comprend des commutateurs. Il est caractérisé par un accès très spécifique : la liaison radio. Enfin, comme tout réseau, il doit offrir à l'opérateur des facilités d'exploitation et de maintenance. Un réseau de radiotéléphonie peut donc se découper en quatre sous-ensembles :

- Station mobile SM
- Sous-système radio BSS (Station Sub-système)
- Sous-système d'acheminement appelé couramment réseau fixe NSS (Network Sub-système)
- Sous-système d'exploitation et de maintenance OSS (Opération Sub-système).

I.3.3.2 Station mobile SM : Elle se compose d'un terminal et d'une carte SIM, le terminale mobile est constituée d'un émetteur /récepteur, chaque terminal est différencie des autres par un code IMEI (international mobile équipement identity) qui est un numéro de 15 chiffres. La carte SIM est une mini base de données dotée d'une mémoire et d'un microprocesseur elle contient des données spécifiques comme le code PIN et IMSI (international mobile subscriber identity) qui sert à identifier un abonné dans n'importe quel réseau GSM.

Deux entités physiques composent cette station :

- **L'équipement mobile ME (mobile équipement) :** La partie banalisée de station mobile constitue l'équipement mobile, la partie visible est l'ensemble formé par l'équipement radioélectrique, le clavier, l'afficheur, le micro et le hautparleur. L'équipement mobile est le terminal sans le module d'identité d'abonné. C'est la partie matérielle qui fournit l'entrée dans ce réseau et de communiquer. Dans réseau GSM, le terminal mobile peut prendre trois aspects :

- Le téléphone de voiture
- Le portable d'une puissance de 8 w
- Le portatif (terminal de poche), d'un poids compris entre 150 et 350 grammes et d'une puissance d'environ 2w

➢ **Le module d'identité d'abonné SIM :** La carte SIM est la partie personnalisée de la station mobile, c'est une carte à mémoire qui permet la séparation entre le moyen de communication et le titre d'abonnement propre à l'abonné.

La principale fonction de la carte SIM est de contenir et de gérer une série de fonctions dont chaque catégorie d'information a une clef particulière qui lui associe et qui protège les accès aux informations stockées dans la carte.

Parmi les informations contenues dans cette carte :

- L'identité IMSI (international mobile suscriber identity) avec laquelle on peut identifier l'abonné dans n'importe quel réseau GSM.
- Le MSISDN (mobile station ISDN number) : est le numéro de l'abonné mobile qui est le seul identifiant connu à l'extérieur du réseau GSM.
- Le MSRN (mobile station roaming number) : il permet le routage des appels entrant directement du commutateur passerelle(GMSC) vers le commutateur MSC de la station mobile.
- L'identité TMSI (temporel mobile suscriber identity) numéro temporaire de l'abonné mobile.
- LAI (location area identification) : une zone de localisation est identifiée par cette adresse LAI.
- Numéro d'identification personnel PIN (personnel identity number) : qui est un numéro composé de quatre chiffres utilisé comme un mot de passe par l'abonnée pour accéder à son abonnement et pour le protéger des connexions frauduleuses.

Figure I.5. La carte SIM

I.3.3.3 Le sous- radio système BSS : Le sous-système radio est l'ensemble constitué par une ou plusieurs stations de base BTS et le contrôleur de station de base associé. Cet ensemble administre les canaux radio d'un motif du réseau.

- **La station de base BTS :** Une station de base est le point d'entrée dans le réseau des stations mobiles, c'est un ensemble d'émetteur / récepteur appelés TRX. A la charge de la transmission radio : modulation, égalisation, codage, correcteur d'erreur.

Elle gère plus généralement toute la couche physique : multiplexage TDMA, saut de fréquence lent, chiffrement, elle réalise aussi l'ensemble des mesures radio nécessaire pour vérifier qu'une communication en cours se déroule correctement.

La BTS gère la couche liaison de données pour l'échange de signalisation entre les mobiles st l'infrastructure, Enfin elle gère la liaison de données avec le BSC afin d'assurer la fiabilité du dialogue. La capacité maximale d'une BTS est typiquement de 16 porteuses, elles peuvent supporter au plus une centaine de communications simultanées. [1]

Il existe différentes configuration BTS-BSC tel que la configuration chainée (multi- drop) ou en étoile (star) comme il peut avoir une BTS au même endroit qu'une BSC.

- Architecture de la station de base :

La figure I.6 montre l'architecture de la station de base :

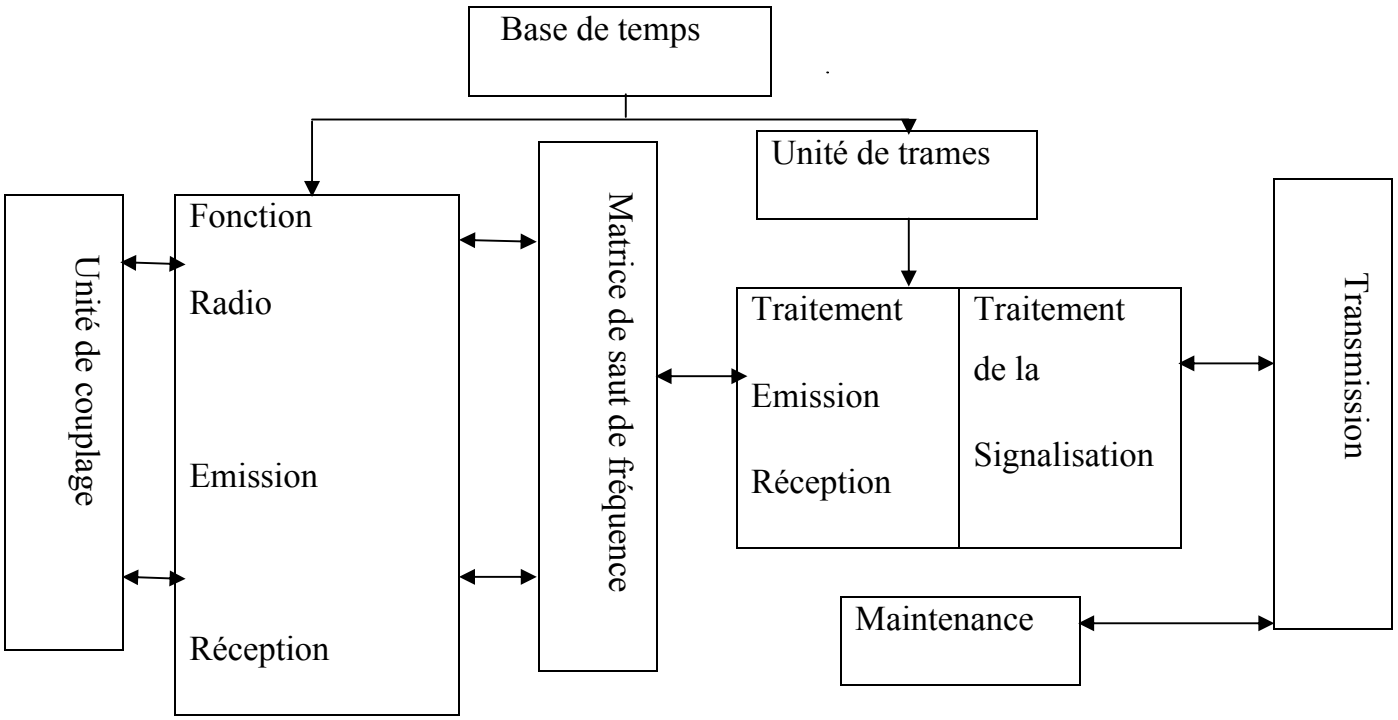


Figure I.6. Architecture d'une station de base

Les équipements composant une station de base :

1. **La base de temps :** La base de temps, fournit tous les signaux d'horloge et de synchronisation nécessaires aux autres éléments présents dans la station, pour remplir toutes les références temporelles définies par la norme GSM pour la synchronisation, et qui sont :
    - Quart de bit (0.92µs).
    - Intervalle de temps (577µs).
    - Trame TDMA (4.615ms).
    - Multitrame.
    - Supertrame.
    - Hypertrame.

2. **L'unité de maintenance :** L'unité de maintenance remplit les missions suivantes :
    - La gestion de tous les protocoles internes de communication et d'acquisition des alarmes provenant de tous les équipements présents dans la BTS.
    - Le filtrage, l'émission et alarmes vers la BTS.
    - Le transfert aux équipements de la station des commandes provenant du contrôleur BTS.
    - Le transfert et la mise à niveau et des fichiers aux unités.
    - La gestion de l'interface homme-machine pour la gestion de la station.
3. **L'unité de saut de fréquence :** L'unité de saut de fréquence assure la commutation suivant la règle de saut de fréquence entre les unités de trames (partie bande de base de la BTS) et les unités de porteuse (partie radiofréquence de station), pour autoriser le saut de fréquence à chaque intervalle de temps.
4. **L'unité de test radio :** L'unité de test radio sert à la localisation des défauts dans la chaine émission-réception et permet des rebouclages sur les canaux radio.
5. **L'unité de trame :** L'unité de trame possède les fonctionnalités nécessaires aux traitements numériques des donnés en bande de base pour 8 canaux à plein débit ou 16 canaux demi-debit.

L'unité de trame gère les canaux radio, pilote les puissances d'émission et contrôle la qualité des transmissions radio.

- Les traitements avant l'émission sont l'adaptation des débits de donnés et de la parole, le codage des canaux, le chiffrement et la constitution des trames.
- Les traitements pour la réception sont les opérations inverses de l'émission avec en plus la démodulation, l'égalisation et la mesure de qualité.

6. **L'unité radio :** L'unité radio se compose de l'émetteur et du récepteur radio. L'émetteur assure la modulation, la transposition de fréquence et l'amplification de puissance. Le récepteur réalise la transposition de fréquence inverse, la

conversion analogique –numérique et le calcul de l'information d'intensité de signal reçu.

7. **L'unité de trafic et d'extension :** Elle est installée en fonction de la densité du trafic à écouler dans une cellule. les unités d'extension sont des unités de traitement en bande de base ou unité de trame et de la partie radio fréquence (unité de porteuse et unité de couplage).
8. **Equipement de transmission :** L'équipement de transmission gère l'interface avec le BSC, il permet de multiplexer jusqu'à 80 canaux de transmission radio à plein débit sur une liaison MIC à 2 Mbit/s.

Une BTS gère trois cellules en zone urbaine, elle pilote huit porteuse radio par cellule ou 192 canaux de communication radio, elle possède une alimentation secteur, une batterie de secoure et 1 à 4 émetteur-récepteur. La BTS est un équipement offrant une grande souplesse d'exploitation.

- **Fonction du BSC :**

Le contrôleur de station de base BSC est l'organe « intelligent » du BSS. Il a pour fonction principale de gérer la ressource radio. Il commande l'allocation des canaux, utilise les mesures effectuées par la BTS pour contrôler les puissances d'émission du mobile et/ou de la BTS, prend la décision de l'exécution d'un handover.

De plus, c'est un commutateur qui réalise une concentration des circuits vers le MSC.

Le BSC est relié par une ou plusieurs liaisons MIC avec la BTS et le MSC, et gère donc une liaison de données avec ceux-ci. La liaison BTS-MSC est une partie similaire à un accès RNIS et fait appel au LAPD. La liaison BSC-MSC utilise le CCITT n°7 et ses différentes couches.

Initialement, les différents constructeurs d'infrastructure n'ont pas tous eu la même philosophie le BSC. Certain ont conçu des BSC de faible capacité estimant préférable de multiplier leur nombre pour minimiser les distances BTS-BSC et réduire

les couts d'exploitation des opérateurs. D'autre ont préféré concevoir des BSC de forte capacité.

I 3.3.4 Le sous-système réseau NSS :

Les éléments du NSS prennent en charge toutes les fonctions de contrôle et d'analyse d'informations contenues dans les bases de données nécessaire a l'établissement des connexions, il est constitué de :

- Le centre de commutation MSC
- L'enregistreur de localisation nominale HLR
- L'enregistreur de localisation des visiteurs VLR
- Le centre d'authentification AUC
- L'enregistreur des identités des équipements EIR

I 3.3.4.1 le centre de commutation MSC :

Il assure la commutation entre les abonnés du réseau, il participe à la fourniture des différents services tel que la téléphonie, la transmission des messages courts et l'exécution de Handover, il permet encore de mettre à jour les différentes base de données (HLR, VLR et AUC) qui donnent toutes les informations concernant les abonnées et leur localisation dans le réseau.

Les commutateurs MSC d'un opérateur d'un opérateur sont relies entre eux pour la commutation interne des informations. Des MSC servant de passerelle (Gateway mobile switching center, GMSC) sont placées en périphérie du réseau d'un opérateur de manière à assurer une interopérabilité entre réseaux ; on distingue deux types d'appels au niveau du MSC :

- Mobile-mobile : dans ce cas le MSC établit une liaison avec un autre MSC
- Mobile-réseau fixe(RTC) : le MSC possède une fonction passerelle GMSC (Gateway MSC) activée au début de chaque appel d'un abonné vers un réseau fixe.

I.3.3.4.2 L'enregistreur de localisation nominale HLR :

Le HLR est la base de données qui gère les abonnés d'un PLMN donné. Il mémorise d'une part les caractéristiques de chaque abonné :

- L'identité internationale de l'abonnée utilisée par le réseau IMSI.
- Le numéro d'annuaire de l'abonnée MSISDN.
- Le profil de l'abonnement (services supplémentaires autorisés, autorisation d'appel international, etc.).

Ces données sont rentrées par l'opérateur à partir de son système d'administration. Elles varient peu dans le temps.

Le HLR est une base de données de localisation. Il mémorise pour chaque abonné le numéro de VLR où il enregistré, même dans le cas où l'abonné se connecte sur un PLMN étranger. Cette localisation est effectuée à partir des informations émises par le terminal à travers le réseau.

L'implantation de HLR peut être centralisée ou décentralisée. Dans le premier cas, un HLR peut gérer plusieurs centaines de milliers d'abonnés et il constitue une machine spécifique. Dans le deuxième cas, il peut être intégré dans le MSC et les données d'une abonnées sont alors physiquement stockées sur le MSC où il communique préférentielles. Les échanges de signalisation son ainsi minimisés. [1]

I.3.3.4.3 L'enregistreur de localisation des visiteurs VLR :

L'enregistreur de localisation des visiteurs mémorise les données relatives à un abonné mobile quand ce dernier entre dans la zone de couverture du sous-système réseau. Le VLR est une base de donnés dynamique ; il dialogue avec le HLR de l'abonné mobile pour prendre connaissance des informations nécessaires. Les informations relatives à l'abonnées accompagnent l'abonné conjointement à ses déplacements dans le réseau.

I.3.3.4.4 Le centre d'authentification AUC :

Le centre d'authentification AUC, mémorise pour chaque abonné une clé secrète utilisée pour authentifier les demandes de services et pour chiffrer les communications. Un AUC est en générale associer à chaque HLR. L'ensemble peut être intégré dans un même équipement. Cependant, du point de vue fonctionnel, ils ne font pas partie du même sous-système.

I.3.3.4.5 L'enregistreur des identités des équipements EIR :

L'EIR (Equipement Identité Register) est une base de données annexe contenant les identités des terminaux (IMEI). Elle peut être consultée lors des demandes de service d'une abonnée pour vérifier que le terminal utilisé est autorisé à fonctionner sur le réseau. L'identité d'un terminal contient un numéro d'homologation commun à tous les terminaux d'une même série, un numéro identifiant l'usine d'assemblage et un numéro spécifique au terminal.

L'accès au réseau peut être refusé parce que le terminal n'est pas homologué, qu'il perturbe le réseau ou bien parce qu'il a fait l'objet d'une déclaration de vol. l'EIR peut contenir une liste blanche de l'ensemble des numéros d'homologation, une liste noire des équipement volés et interdits d'accès, et une liste grise des terminaux présentant des dysfonctionnements insuffisants pour justifier une interdiction totale. Le réseau peu mémorisé l'identité IMSI d'un abonnée utilisant un terminal inscrit en liste noire ou grise et la transférer au système d'administration pour permettre d'identifier les accès frauduleux.

I.3.4 Les régions géographiques d'un réseau GSM :

La structure du réseau GSM est divisé en zones géographiques pour assurer l'acheminement des appels vers leurs destinations, ainsi on peut distinguer cinq régions comme l'indique la figure suivante : [2]

- Les cellules.
- La zone de localisation ou de repérage.
- La zone de service MSC/VLR.

- La zone dédiée au réseau terrestre mobile public PLMN.
- La zone de service GSM.

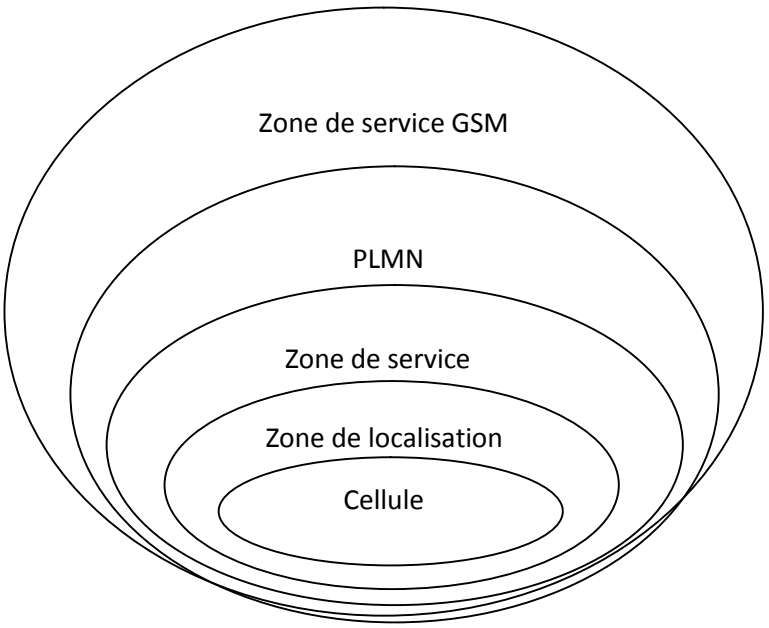


Figure I.7. Zones géographiques d'un réseau GSM.

- **La cellule :**

Dans un système cellulaire, la zone géographique à couvrir est divisée en cellule, chacune possède une station de base (BTS) qui assure la transmission avec les mobiles présents dans la cellule.

La cellule est la zone géographique couverte par le rayonnement d'une antenne émettrice, sa taille est fonction de la puissance émise par l'antenne.

La forme de la cellule dépend de la topographie de la région servie par l'antenne, elle est représentée géométriquement par un hexagone comme le montre la figure suivante.

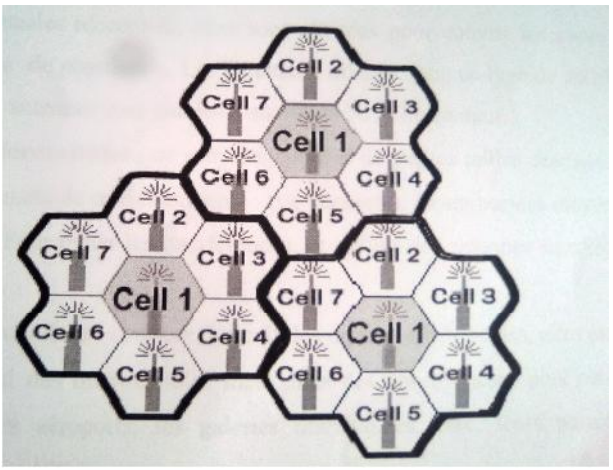


Figure I.8 Architecture cellulaire.

L'unité d'utilisation des fréquences radio qui définit les canaux de communication, est un motif de sept cellules appelé « **cluster** ».

Dans la conception d'un réseau cellulaire, il faut considérer les aspects suivants :

- La topographie (bâtiments, montagne,…).
- La densité de la population pour établir la dimension de la cellule.
- Deux cellules adjacentes ne peuvent utiliser la même bande de fréquence afin d'éviter les interférences.
- La distance entre deux cellules ayant la même bande de fréquence doit être de deux à trois fois le diamètre d'une cellule.

- **Différents types de cellules :**

La dimension d'une cellule est fonction de la puissance de son émetteur, il existe plusieurs types de cellules comme illustrés dans la figure suivante.

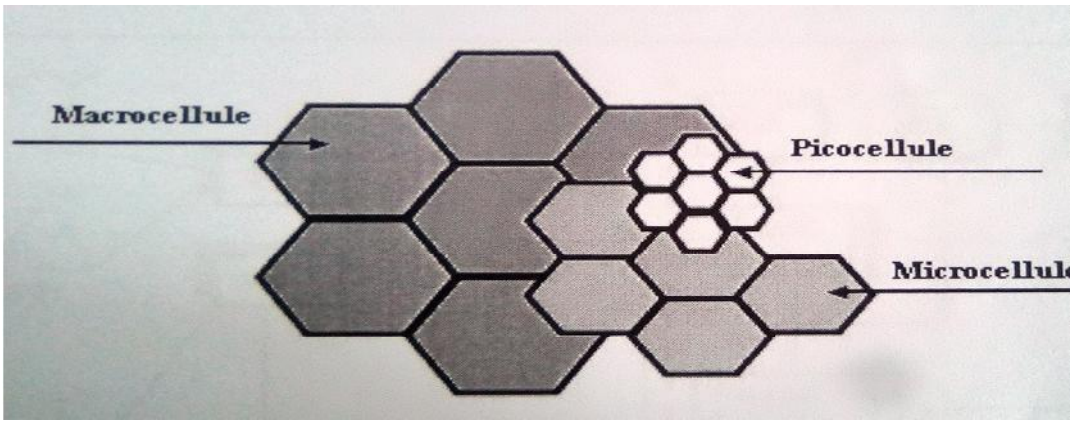


Figure I.9. Le découpage cellulaire

- **Les macrocellules :** ce sont des cellules dont le rayon s'étend jusqu'à 30 Km selon les obstacles rencontrés, elles sont utilisées pour couvrir les zones rurales à faibles densités de population. Les émetteurs utilisés dans ce type de cellules sont puissants et leurs antennes sont placées à au moins 30 m de hauteur.
- **Les microcellules :** ce sont des cellules de petites tailles destinées aux zones à très forte densité de trafic (exemple : rue passante). Leurs portées moyennes est d'environ 500m. pour éviter les interférences, on utilise des antennes émettrices de puissances réduites.
- **Les picocellules :** ce sont des cellules de taille très petites, ils ont un rôle similaire que celui des microcellules mais dans des zones encore plus petites telles que les gares, les aéroports, les galeries marchandes... etc. leur portées maximales est d'environs 100m.

- **la zone de localisation :(LA : location area)** elle contient plusieurs cellules contrôlées par une ou plusieurs contrôleurs de station de base(BSC), mais appartient à un seul MSC c'est la zone par laquelle on localise un abonné mobile appelé.
- **La zone de service MSC/VLR :** c'est un groupe de LA sous le contrôle d'un seul MSC. Ainsi pour acheminer un appel vers un terminal, le réseau doit connecter la communication au MSC de la zone de service MSC/VLR où le terminal est localisé.
- **Réseau terrestre mobile public (PLMN :** public land mobile network) : c'est la région desservie par un opérateur, elle est constituée de plusieurs zones de

services MSC/VLR. Par exemple en Algérie, il existe plusieurs PLMN, chacun appartenant à un réseau mobile d'un opérateur (mobilis, djezzy…).

- **Une zone de service GSM :** c'est la zone géographique ou un abonné peut accéder au réseau GSM. Cette zone va s'agrandissant grâce aux accords bilatéraux entre les différents opérateurs pour travailler ensemble, on parle alors d'itinérance (roaming).

Discussion :

Dans ce chapitre, notre étude s'est focalisée sur la présentation de la carte de commande électronique qui est la carte GSM sim900. Nous avons aussi présenté le réseau GSM qui est adapté au module GSM ainsi ses régions géographiques (réseaux cellulaires).

L'interface radio constitue la partie la plus importante et la plus compliquée car elle est riche en fonctions, c'est à ce niveau que s'effectuent les différents traitements que peut subir une trame de parole, beaucoup de paramètres entrent en jeu (environnement, types d'antennes…), mais en règle générale, on essaie de trouver un compromis pour que les systèmes implantés soient économiquement viables pour l'opération tout en gardant une qualité de service minimale pour tous les usagers. Le prochain chapitre sera consacré à la présentation de la carte Arduino UNO.

CHAPITRE II

Présentation de la carte Arduino UNO

II.1 Préambule :

L'électronique est de plus en plus remplacée par l'électronique programmée. On parle aussi de système embarquée ou d'informatique embarquée. Son but est de simplifier les schémas électroniques et par conséquent réduire l'utilisation de composants, réduisant ainsi le coût de fabrication d'un produit, il en résulte des systèmes plus complexes et performants pour un espace réduit.

Depuis que l'électronique existe, sa croissance est fulgurante et continue encore aujourd'hui. L'un des composants utilisée dans l'électronique embarquée est l'Arduino qui est un sous-domaine de l'électronique et qui a l'habileté d'unir la puissance de la programmation à la puissance de l'électronique. Au début de ce chapitre, un petit historique concernant le développement des cartes Arduino nous semble nécessaire. Ensuite, nous montrons sa conception aussi bien sur le plan matériel que sur le plan le logiciel.

II.2 L'historique :

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 à Iverea en hiver 2005.

Les étudiants qui se plaignent envers leur enseignant appelé Banzi, de ne pas avoir accès à des solutions bas prix pour accomplir leurs projets de robotique. Par la suite, l'enseignant Banzi a contacté David Cuartielles, qui est un ingénieur Espagnole spécialisé dans les microcontrôleurs pour créer leur propre carte en embarquant dans leur histoire un des étudiants 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 de plus la carte était créée. Ils décidèrent de l'appeler Arduino.

II.3 Présentation de la carte Arduino :

II.3.1 Définition :

La carte Arduino est un circuit imprimé en matériel libre, les modules d'origine sont fabriqués par la société italienne SMART PROJECTS (dont les plans sont publiés en licence libre) sur lequel se trouve un microcontrôleur qui peut être programmer, afin d'analyser et de produire des signaux électrique, de manière à effectuer le contrôle de diverses taches, dans différents domaines, comme la charge de batteries, la domotique (le contrôle des appareils domestiques comme : l'éclairage, le chauffage…), le pilotage d'un robot, la commande de des actionneurs, etc.

Arduino est une plate-forme open source d'électronique programmée qui est basé sur une simple carte à microcontrôleur (de la famille AVR), et un logiciel, véritable environnement du développement intégré, pour écrire, compiler et transférer le programme vers la carte à microcontrôleur. Ces cartes sont basé sur interface entrée/sortie simple pouvant recevoir des entrées d'une grande variété d'interrupteurs ou de capteur, et pouvant contrôler une grande variété de lumière, moteurs ou tout autre types d'actionneurs et tous cela se fait sur un environnement de développement proche du langage C.

II.3.2 Les différentes cartes Arduino :

Actuellement, il existe plus de 20 versions de module Arduino, nous citons quelques un afin d'éclaircir l'évaluation de ce produit scientifique et académique :

Le NG d'Arduino, avec une interface d'USB pour programmer et usage d'un Microcontrôleur ATmega8.

L'Arduino Nano, une petite carte programme à l'aide porte USB cette version utilisant un microcontrôleur ATmega168 (Atmega328 pour une plus nouvelle version).

L'Arduino Bluetooth, avec une interface de Bluetooth pour programmer en utilisant un microcontrôleur ATmega168.

L'Arduino UNO, utilisations microcontrôleur ATmega 328.

L'Arduino Mega2560, utilisations un microcontrôleur ATmega2560, et possède toute la mémoire à 256KBS. Elle incorpore également le nouvel ATmega8U2 (ATmega16U2 dans le jeu de puces d'USB de version).

On les montres sur la figure suivante les déférentes cartes Arduinos :

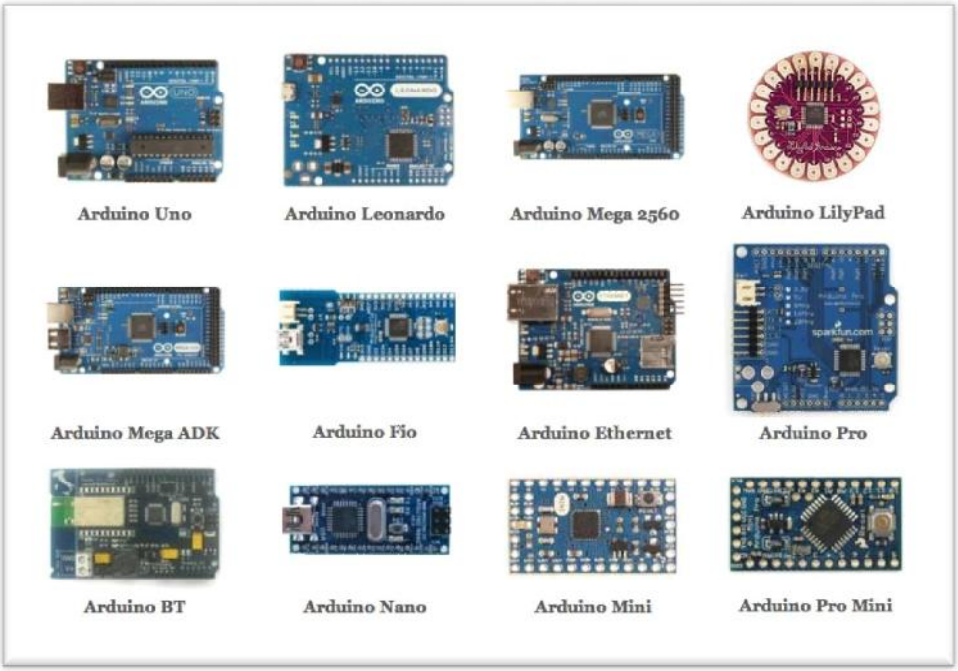


Figure II.1. Les différentes cartes Arduinos

La carte que nous allons utiliser durant ce mémoire est l'Arduino UNO, elle est illustrée par la figure suivante :

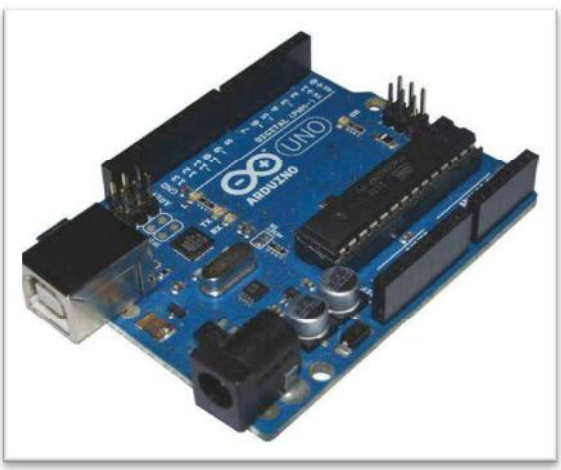


Figure II.2. La carte Arduino UNO

II.4 Structure de la carte Arduino:

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 programmateur dédié ne soit pas nécessaire. Le système Arduino se repose essentiellement de deux parties principales qui sont : le matériel et le logiciel.

II.4.1 Partie matériel : La figure suivante schématise les différentes parties essentielles d'une carte Arduino UNO, néanmoins toutes les autres sont construites avec la même architecture.

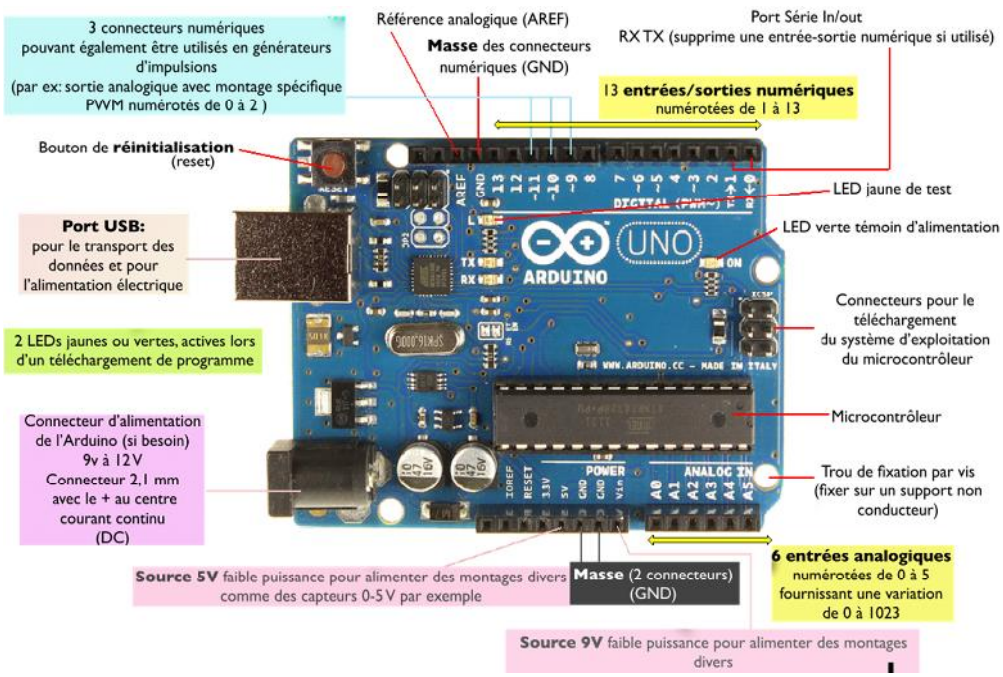


Figure II.3. Déférentes parties de la carte Arduino UNO

- **Le microcontrôleur ATmega328 :** Un microcontrôleur ATMega328 est un circuit intégré qui ressemble les éléments d'un ordinateur, c'est lui le cerveau qui va traiter les données et les informations provenant des capteurs (entrées), cela avec un temps de traitement très réduit, ensuite donner les consignes souhaitées en sortie. 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. la figure montre un microcontrôleur ATmega 328, qu'on trouve sur la carte Arduino.

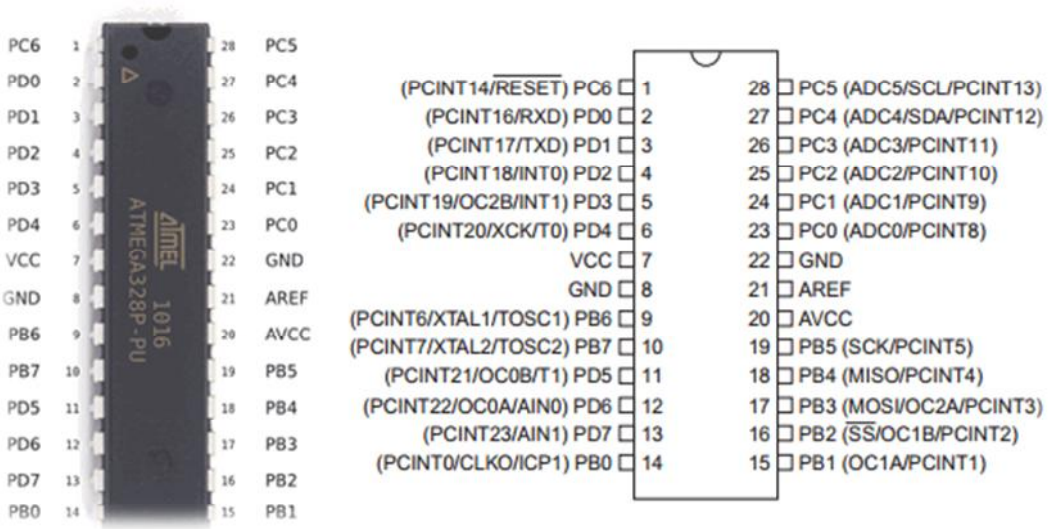


Figure II.4. Microcontroleur ATmega 328

A. Les caracteristique du microcontroleurATmega 328 :

Microcontroleur	ATmega 328
Fréquence d'horloge	16 MHZ
Tension de service	5 V
Tension d'entrée (recommendée)	7-12 V
Tension d'entrée(limite)	6-20 V
Ports numeriques	14 E/S (6 sorties en MLI)
Ports analogique	6 Entrées analogiques
Courant max par broche 3.3 V	50 Ma
Courant max par broche 5 V	500 mA
Mémoire	32 Ko Flash 2 Ko SRAM 1 Ko EEPROM
Dimenssions	6.86 cm*5.3cm

Tableau II.1. Les caractéristiques d'un microcontroleur

B. La structure interne du microcontroleur ATmega328 :

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 :

La **RAM** qui est la mémoire vive du composant. Elle est très rapide et doit gérer beaucoup d'informations très vite. Il existe différent types de RAM, dans notre cas ce sera de la **SRAM** (pour **S**tatic **R**andom **A**ccess **M**emory), qui est plus rapide mais aussi plus consommatrice en énergie que la RAM Dynamique de notre ordinateurs et aussi plus encombrante. Elle servira à stocker les variables du programme. Chaque fois que nous faisons une nouvelle déclaration de variables, cette dernière se retrouvera dans cette mémoire. Une caractéristique à ne pas négliger, cette mémoire est entièrement effacée lorsque l'alimentation de l'Arduino cesse (on dit qu'elle est « volatile »).

Ensuite, on trouve une mémoire dite morte, l'**EEPROM** (**E**lectrically **E**rasable **P**rogrammable **R**ead-**O**nly **M**emory). Alors non, n'allez pas croire à la lettre qu'elle est vraiment morte. En fait, on la nomme ainsi car elle est capable de stocker des informations même lorsqu'elle n'est plus alimentée. Cette dernière est similaire au disque dur de notre ordinateur par son comportement et ses caractéristiques. La vitesse d'accès est moins élevée que la RAM et sa durée de vie est plus faible aussi.

Enfin, une dernière mémoire de l'Arduino est la Flash**.** Elle a un rôle un peu particulier, elle sert à stocker le programme que nous téléchargeons dans le microcontrôleur. Elle retient donc les informations même lorsque l'alimentation est coupée. Comme dit plus tôt, c'est ici que sont stockées toutes les instructions de votre programme, ainsi que le *boot loader*, un petit bout de code qui agit comme le BIOS de notre PC. Il détecte au démarrage de l'Arduino si on tente de programmer la carte via la liaison série et le cas échéant copiera les données dans la mémoire FLASH. On n'y stocke pas de données pendant l'exécution du programme. En revanche, on peut y

stocker des constantes (comme des chaînes de caractères pour notre écran LCD par exemple) afin de gagner un peu de place dans la RAM.

❖ **L'Alimentation :**

L'alimentation qui assure la distribution d'énergie aux différents composants de la carte Arduino, être effectuée via :

Par la prise d'adaptateur qui fournit une tension comprise entre 7 et 12 V c'est ce qu'on appelle la tension recommandée (la tension est ensuite convertie toute seul en 5V par Arduino).

Une connexion USB de l'ordinateur qui fournit toujours directement du 5V (la tension de fonctionnement). Ainsi la carte peut fonctionner avec une alimentation externe de 6 à 20 volts c'est ce qu'on appelle la tension limite. Si moins de 7 volts sont fournis, la broche 5volts peut éventuellement fournis moins de 5 volts et la carte peut alors être instable. Si plus de 12 volts sont utilisées, le régulateur de tension peut surchauffer et abimer la carte.

Les broches d'alimentation sont les suivantes :

- 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).
- 5V : La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation 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.
- 3V3 : Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de notre 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

- GND : Broche de masse (ou 0V).

❖ **Les entrées /sorties :**

1. **Les entrées /sorties numérique ou digitale :** Cette carte Arduino UNO possède 14 broches numériques (numérotée de 0 à 13) configurables en entrées et en sorties selon les tâches assignées. Elles sont déclarées dans le programme avec le terme digital. Donc pour qu'une pin ou broche soit en sortie, dans le programme, il faut la déclarer avec l'instruction OUTPUT. Si elle est utilisée comme une entrée, il faut la déclarer avec l'instruction INPUT. [3]

Le signal numérique ne peut prendre que deux états « 1 » correspond à l'état haut, en générale 5V pour la UNO, toute fois en fonction du montage à réaliser, il y a des chutes de tension, ou bien « 0 » correspond à l'état bas.

2. **Les entrées analogiques :** Contrairement à un signal numérique qui ne peut prendre que deux états différents le signal analogique peut prendre une infinité de valeurs. La carte Arduino UNO dispose 6 entrées analogiques, les broches A0 à A5 sont des entrées analogiques, elles sont dotées de convertisseurs analogique / numérique qui convertir la valeur de l'entrées en une suite de valeurs que la carte fait correspondre à un nombre variant entre 0 et 1023, ce qui nous permet de récupérer les informations d'un capteur. Ces entrées analogiques sont déclarées dans le programme avec le l'instruction « analog ».

❖ **Visualisation :** Les trois points blancs sont en fait des LED dont la taille est de l'ordre du millimètre. Ces LED servent à deux choses :

Teste LED 13 : elle est connecte à une broche du microcontrôleur qui peut servir pour tester le matériel. Quand on branche la carte au PC ; elle clignote quelques secondes.

Les deux LEDs TX/RX : servent à visualiser l'activité sur la voie série (une pour l'émission et l'autre pour la réception). Le téléchargement du programme dans le microcontrôleur se faisant par cette voie, on peut les voir clignoter lors du chargement.

- **La connectique :** La carte Arduino ne possédant pas de composant (résistances, diodes, moteurs…) qui peuvent être utilisée pour un programme, mis à part la LED connectée à la broche 13 du .microcontrôleur, il est nécessaire de les rajouter. Mais pour ce faire, il faut les connecter à la carte. C'est là qu'intervient la connectique de la carte. Sur les Arduinos et sur beaucoup de cartes comptables Arduino, les connecteurs se trouvent au même endroit. Cela permet de fixer des cartes d'extension, appelée shield. Cette connectique est importante et a un brochage qu'il faudra respecter. Par exemple, la carte Arduino peut être étendue avec des sheilds, comme le Sheild Ethernet qui permet de connecter cette dernière à internet.

II.4.2 Partie programme (Software) : La carte Arduino UNO a besoin d'un programme pour fonctionner, et pour qu'on puisse la programmer on doit disposer du logicielle compatible avec cette carte, ce logiciel est Arduino EDI (espace de développement intégré) et ce programme est une liste d'instructions qui est exécutée par un système. Et c'est nous qui allons le réaliser.

II.4.2.1 Présentation de l'espace de développement intégré (EDI) 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.

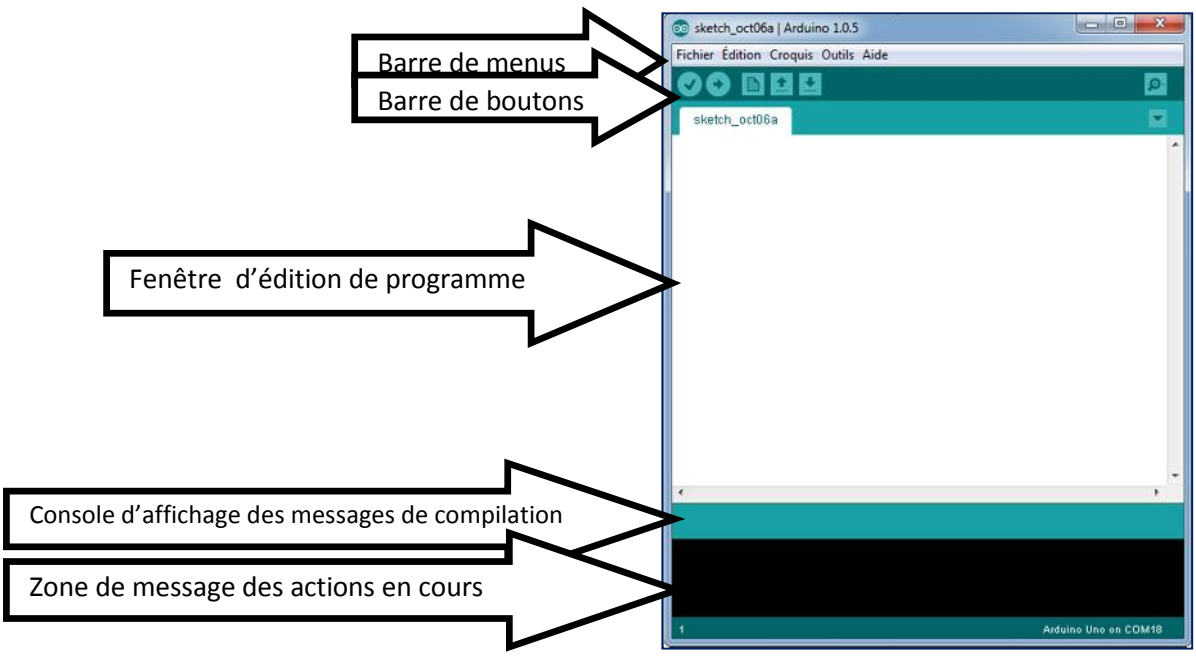


Figure II.5. Présentation du logiciel Arduino

- **Barre de menus :**

Fichier (file) : ce menu contient les différentes options de création, d'ouverture, de sauvegarde, d'impression de programme ou d'un exemple, parmi les exemples qui accompagnent le logiciel Arduino.

Editer (Edite) : Ce menu contient les options de copier/coller, sélection, et les options de recherche.

Programme ou séquence (Sketch) : Ce menu contient les différentes fonctions de la barre des boutons, ainsi que les options d'ajouts de librairies ou de fichiers.

Outils (Tools) : C'est dans ce menu qu'on sélectionne le type de carte à programmer, et le port série utilisé.

Aide (Help) : Ce menu est fait pour donner de l'aide concernant les différents problèmes rencontrés au niveau de logiciel Arduino.

- **Barre de boutons :**

Les icones de la barre de boutons sont :

Figure II.6. Les icones de la barre de boutons

La premiere : Vérifier/compiler le code à la recherche d'érreur.

La deuxieme : Transférer vers la carte, compile notre code et transférer vers la carte arduino.

La troiseme : Nouveau (New), est de créer un neveaux programme (ouvre une nouvelle fenetre) .

La quatrieme : ouvrire (open), qui est la liste de tous les programmes dans le « livre de programme ».

La cinquieme : Enregistrer (Save) un programme.

La sexieme : Moniteur série (Serial Moniteur) qui est d'ouvrir la fenetre de moniteur (ou terminal)série.

- **Fenêtre d'édition de programme :** ce bloc va contenir le programme que nous allons créer.
- **Console d'affichage des messages de compilation.**
- **Zone de message des actions en cours :** Celui-ci est important, car il va nous aider à corriger les fautes dans notre programme. C'est le débogueur.
- **Structure d'un programme Arduino :**

Un éditeur : c'est l'espace utilisé pour écrire le programme à réaliser, comme il dispose aussi des onglets de navigation.

Une zone de message : comme son nom l'indique cette espace est utilisée par l'IDE pour communiquer à l'utilisateur des informations sur l'état du programme et des actions en cours.

Une console texte : qui affiche les messages concernant le résultat de la compilation du programme (il indique si le programme comporte des erreurs. [4], [5]

- **Le programme Arduino :**

A. Description de la structure d'un programme Arduino :

Le langage Arduino est basée sur les langages « C/C++ ». Un programme utilisateur Arduino est une suite d'instruction élémentaire sous fourme textuelle, ligne par ligne. La carte lit puis effectue les instructions les unes après les autres, dans l'ordre défini par les lignes de code, comme lors d'une programmation classique. Cette structure ce décompose en trois parties :

*** Définition des constantes et variables globale :** la déclaration des variables se fait généralement dans l'espace global (de façon à partage les variables les plus importantes entre les deux fonctions principales).

*** Fonction principale : void setup () :**

- Initialisation des ressources de la carte,
- Configuration des entrées/sorties,
- Définition de la vitesse de fonctionnement.

*** Fonction boucle : void loop () :** cette partie s'exécute en boucle.

- Description de fonctionnement générale de programme.
- Gestion des interactions entre les entrées/sorties.

B. Le jeu d'instruction du langage Arduino :

*** les syntaxes du programme :**

- chaque instruction termine par un « ; ».
- les accolades « { » et « } » sont les conteneurs du code du programme. Elles sont propres aux fonctions, aux conditions et aux boucles. Les instructions du programme sont écrites à l'intérieur de ces accolades.
- les commentaires sont des lignes de texte incluses dans le programme et qui ont pour but de vous informez vous-même ou les autres de la façon dont le programme fonctionne. Ces lignes ajoutées sont ignorée par le compilateur. Les commentaires sont précédés des caractères « // » ou bien encadrer par « /* » et « */ ».
- il est formellement interdit de mettre des accents en programmation, sauf dans les commentaires.
- un nombre en binaire doit être précédé de la lettre « b ».
- un nombre écrit en hexadécimal doit être précédé par les caractères « 0x ».

*** Les commandes du programme Arduino :** Le tableau ci-dessous nous montre les commandes du programme Arduino :

Structure générale	• Void setup () (configuration-préparation) • Void loop () (Exécution)
Contrôle des conditions	• if (si) • if…else (si…alors) • For (pour…) • Switch case (Dans le cas où…) • While (Pendant que…)
Opération de comparaison	• = (Equivalent) • - !=(Différent) • < (Inferieur à) • > (supérieur à) • <= (inférieur ou égal à) • >= (supérieur ou égal à)
Autres commandes	• // (Commentaire simple) • /**/ (Commentaire multi ligne) • # define (Donner une valeur à un nom)
Variables	• Char (Variable ‘caractère’) • Int (Variable ‘nombre entier’) • Long (Variable ‘nombre entier de tres grand taille’) • String (Variable ‘chaine de caractères’) • Array (Tableau de variables)
Niveau logique des connecteurs numériques	• HITH (Etat 1) • LOW (Etat 0) • INPUT (configuré en entrée) • OUTPUT (configurée en sortie)
Fonction *Entrées-sorties numérique	• Entrées-sorties : • pinMODE (broche, état) (configuration des broches en entrée ou en sortie) • digitalWrite (broche, état) (écrire un état sur une broche numerique) • digitalRead (broche) (lire un état sur une broche numerique) • unsigned long pulseIn (broche, état) (lire une impulsion sur une broche numerique)

Fonctions *Entrées analogique	• Int analogRead (broche) (lire la valeur d'une broche analogique) • analogWrite (broche, valeur) (PWM : écrire une valeur analogique sur les broches 5, 6, 9, 10 et 11)
Fonction *Gestion du temps	• unsigned long millis () (temps de fonctionnement de programme) • delay (ms) (attente, en millisecondes). • Delay Microsecondes (us) (attente, en microsecondes)
Opération booléennes	• && (et) • || (ou) • ! (et pas)

Tableau II.2. Description des commandes de programme Arduino

C. Les étapes à suivre pour programmer la carte Arduino :

*** L'écriture du programme :** Cette première étape consiste à saisir le programme.

La figure suivante présente la fenêtre du programme :

Figure II.7. L'écriture du programme Arduino

*** Compilation du programme :** Dans cette partie, on vérifie si le code contient des erreurs de syntaxe. En cas d'anomalie dans le programme, le compilateur nous enseigne sur le type d'erreur et la ligne où elle se trouve. (Pour la correction des erreurs il suffit de consulter l'aide disponible dans la barre de menu). [6]

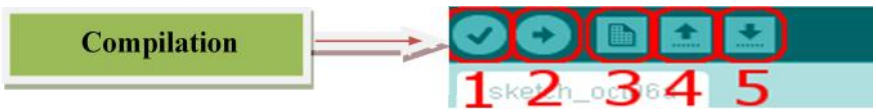


Figure II.8. Compilation de programme

Si aucune erreur n'est détecté, le programme est compilé et le message « compilation terminée » apparait, suivi de la taille du programme.

Figure II.9. Console d'affichage

Sélection de la cible (type de carte) : Avant de transférer le programme vers la carte Arduino, il faut sélectionner le type de carte Arduino (cible) depuis le menu Ouils>Board. La carte doit évidemment être connectée à un ordinateur via un câble USB. Vérifions que c'est bien le nom « Arduino Méga 328 » qui est coché. Si ce n'est pas le cas, on le coche.

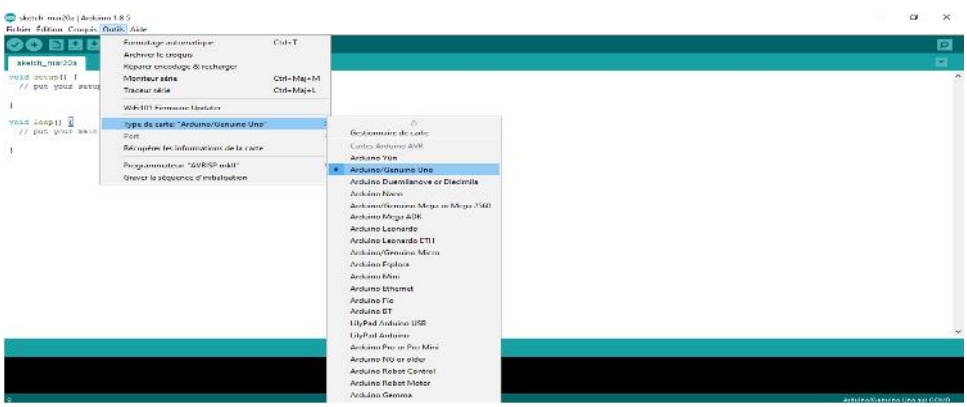


Figure II.10. Sélection de la cible

*** Sélection du port :** Sélectionner le port série sur lequel est branché cette (depuis le menu Outils>Port).

Figure II.11. Sélection du port

***Transfert du programme :**Cela se fait en cliquant sur le boutan :

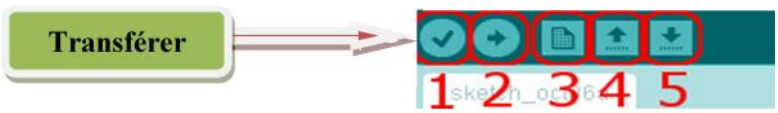


Figure II.12. Transfert de programme

Le programme est téléchargé vers la carte, le message « téléversement terminé » saffiche une opération términée.

Sur la plupart des cartes, nous devons voir les LEDs des lignes RX et TX clignoter rapidement, témoignant que le programme est bien transféré. Durant le transfert le bouton devien jaune et le logiciel Arduino affiche un message indiquant que le transfert est en cours.

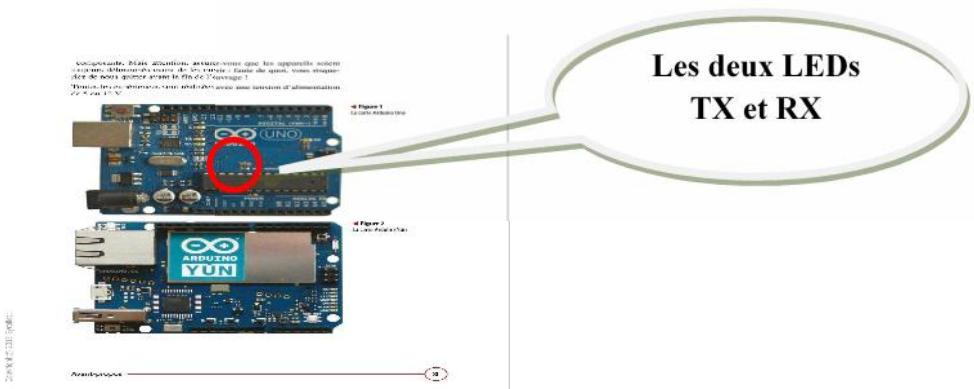


Figure II.13. Clignetement des deux LEDs du port série

II.4.2.2 Fonctionnement globale de la carte Arduino : La carte Arduino fonctionne de façon globale et répondre à quelques questions qui Pourraient nous trotter dans la tête, exemple : Comment la carte sait qu'il y a une LED de connexion, comment sait-elle que c'est sur telle broche, et le programme où est ce qu'il se trouve et sous quelle forme ?

C'est le programme qui va définir chaque action que va exécuter la carte Arduino. Mais ce n'est pas tout, dans le programme il y a plusieurs zones, que nous verrons plus en détail tout au long de la lecture de ce cours, qui ont chacune un rôle particulier.

- La première zone : sert principalement à dire à la carte de garder en mémoire quelques informations qui peuvent être : l'emplacement d'un élément connecté à la carte, par exemple une LED en broche 13.
- La zone secondaire est l'endroit où l'on va initialiser certains paramètres du programme. Par exemple, on pourra dire à la carte qu'elle devra communiquer avec l'ordinateur ou simplement lui dire ce qu'elle devra faire de la LED qui est connectée sur sa broche 13.
- La dernière zone est la zone principale où se déroulera le programme. Tout ce qui va être écrit dans cette zone sera exécuté par la carte, se sont les actions que la carte fera. Par exemple, c'est ici qu'on pourra lui dire de faire clignoter la LED sur sa broche 13.

En conclusion, tout ce que va faire la carte est inscrit dans le programme. Sans programme, la carte ne sert à rien, C'est grâce au programme que la carte Arduino va savoir qu'une LED est connectée sur sa broche 13 et ce qu'elle va devoir faire avec, allumer et éteindre la LED alternativement pour la faire clignoter.

Le programme est envoyé dans la carte lorsque on clique sur le bouton téléversé, le logiciel Arduino va alors vérifier si le programme ne contient pas d'erreur et ensuite le compiler (le traduire) pour l'envoyer dans la carte.

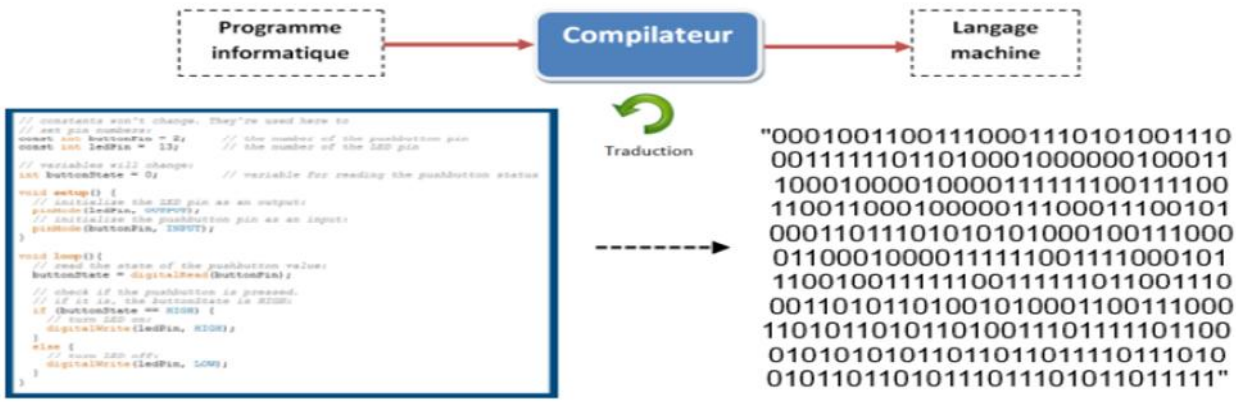


Figure II.14. Traduction du programme au langage machine

Au départ, le programme est sous forme d'un texte, puis il est transformé en un langage composé uniquement de 0 et de 1.

L'envoi du programme est géré par notre ordinateur : le programme passe, sous forme de 0 et de 1, dans le câble USB qui relie notre ordinateur à la carte.

Le programme rentre donc dans la carte en passant en premier, par le connecteur USB de celle-ci. Il va alors subir une petite transformation qui permet d'adapter le signal électrique correspondant au programme vers un signal plus approprié pour le microcontrôleur. On passe ainsi d'un signal codé pour la norme USB à un signal codé pour une simple voie série. Puis ce nouveau signal est alors intercepté par le microcontrôleur.

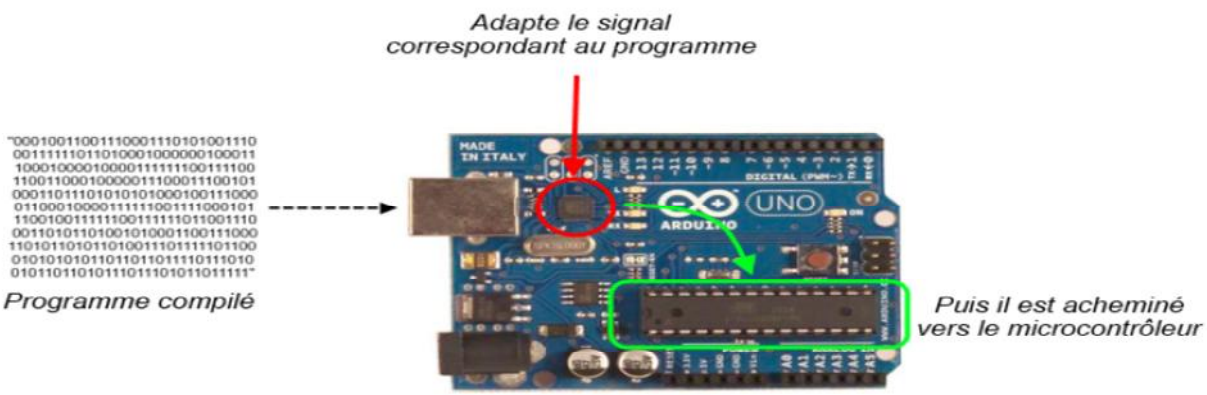


Figure II.15. L'acheminement du programme compilé vers le microcontrôleur.

A l'intérieur du microcontrôleur, reçoit le programme sous forme d'un signal électrique sur ses broches Tx et Rx, d'ailleurs disponible sur les broches de la carte. Une fois qu'il est reçu, il est intégralement stocké dans une mémoire de type flash que l'on appellera « la mémoire du programme ». Ensuite, lorsque la carte démarre, le cerveau va alors gérer les données et les répartir dans les différentes mémoires : La mémoire programme est celle qui va servir à savoir où l'on en est dans le programme, à quelle instruction on est rendu. La mémoire de données, aussi appelé « RAM » va stocker les variables telles que le numéro de la broche sur laquelle est connectée une LED, ou bien une simple valeur comme un chiffre, un nombre, des caractères, etc.

Avantage :

- Nombres de broches suffisants pour les projets élémentaires.
- Facile à utiliser, sa programmation est très accessible pour les débutants.
- Vaste choix de shield.
- Prix très abordable.

Inconvénient :

- Manque de mémoire pour les gros projets.
- Insuffisance de broches pour les projets ambitieux.
- Ne pas être utilisée comme hôte USB pour simuler un clavier ou une souris.[7]

Discussion :

A travers ce chapitre, notre étude s'est focalisée sur la présentation de la carte de commande qui est la carte Arduino, puis nous avons cité des différents types de cette dernière. Ensuite, nous avons expliqué les deux parties essentielles de l'Arduino (la partie matérielle et la partie de programmation). Nous avons également expliqué le principe de fonctionnement de la carte Arduino sans oublier ses caractéristiques.

Dans le chapitre qui suit, nous allons montrer les étapes effectuées pour réaliser un téléphone portable.

CHAPITRE III

Réalisation de notre application

III.1. Préambule :

Dans ce chapitre, nous allons passer à la réalisation de notre application, à savoir le téléphone portable à base d'Arduino. Pour ce faire, un module GSM dont la description a été donnée précédemment est utilisé. Ce dernier a pour but d'envoyer et de recevoir des informations à partir du réseau. Pour qu'il soit utilisable nous avons ajouté le clavier alphanumérique 4*4 pour qu'on puisse entrer le numéro de téléphone mobile, l'afficheur LCD 16*2 pour afficher des messages, des instructions et des alertes, nous avons également interfacé un microphone et un haut-parleur pour le son d'appel vocal et de sonnerie. Ceci est piloté par le microcontrôleur de la carte Arduino pour analyser ces informations.

Tous les composants utilisés dans cette carte ont une fonction bien précise. Nous allons présenter tout ceci le long de ce chapitre. Un programme informatique est aussi élaboré et implémenté dans le microcontrôleur de la carte Arduino qui gère notre système. Ce dernier est donné par la figure suivante :

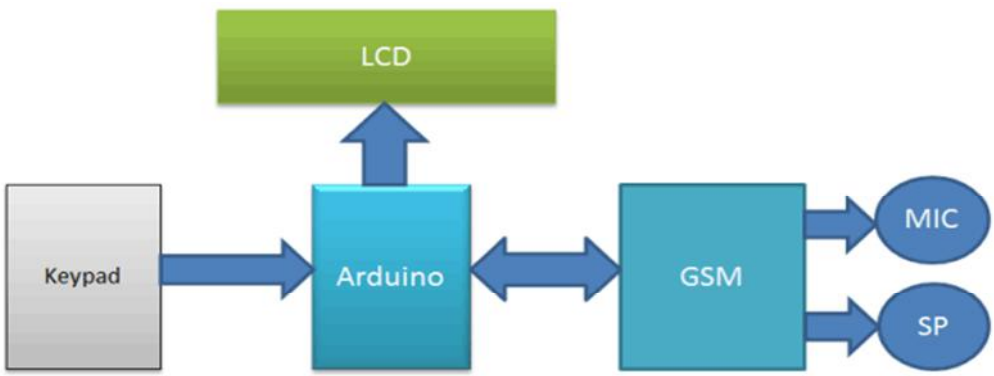


Figure III.1. Le prototype de téléphone portable.

III.2.L'écran LCD :

Durant la réalisation de notre prototype on a essayé de faciliter la communication avec l'utilisateur afin de pouvoir afficher des informations sans avoir besoin d'un ordinateur. Un écran LCD est la meilleure solution. Pour notre réalisation on a utilisé un LCD de type afficheurs alphanumériques 16*2 (16 colonnes et 2 lignes).

Les Ecran à Cristaux Liquides ou bien Liquide Crystal Display en anglais utilisent la polarisation de la lumière par des filtres polarisants et la biréfringence de certains cristaux liquides en phase nématique, dont on peut faire varier l'orientation en fonction du champ électrique. Du point de vue optique, l'écran à cristaux liquides est un dispositif passif : il n'émet pas de lumière, seul sa transparence varie, et il doit donc disposer d'un éclairage.

Un écran LCD possède une grille de carré. Ces carrés sont appelés des pixels (de l'anglais Picture Elément), Chaque pixel est un cristal liquide. Lorsqu'aucun courant ne le traverse, ses molécules sont orientées dans un sens (0°). En revanche lorsqu'un courant le traverse, ses molécules vont se tourner dans la même direction (90°).Pour pouvoir afficher des caractères sur l'écran il nous faudrait activer individuellement chaque pixel de l'écran.

❖ **Les Caractéristiques de LCD sont données par la table suivante :**

Broche	Nom	Niveau	Fonction
1	Vss	-	Masse
2	Vdd	-	Alimentation positive +5V
3	Vo	0-5V	Cette tension permet, en la faisant varier entre 0 et +5V, le réglage du contraste de l'afficheur.
4	RS	TTL	Sélection du registre (Register Select) grâce à cette broche, l'afficheur est capable de faire la différence entre une commande et une donnée. Un niveau bas indique une commande et un niveau haut indique une donnée.
5	R/W	TTL	Lecture ou écriture (Read/Write) L : Écriture H : Lecture

6	E	TTL	Entrée de validation (Enable) active sur front descendant. Le niveau haut doit être maintenue pendant au moins 450 ns à l'état haut.
7	D0	TTL	Bus de données bidirectionnel 3 états (haute impédance lorsque E=0)
8	D1	TTL	
9	D2	TTL	
10	D3	TTL	
11	D4	TTL	
12	D5	TTL	
13	D6	TTL	
14	D7	TTL	
15	A	-	Anode rétro éclairage (+5V)
16	K	-	Cathode rétro éclairage (masse)

Tableau III.1. Les caractéristiques de LCD

- Il existe deux modes de fonctionnement de l'afficheur sont disponibles, le mode 4 bits et le mode 8 bits, modes que l'on choisira à l'initialisation de l'afficheur.

  - **Mode 4 bits :**

Il peut, dans certains cas, être nécessaire de diminuer le nombre de fils utilisés pour commander l'afficheur, comme, par exemple lorsqu'on dispose de très peu de broches d'entrées sorties disponibles sur un microcontrôleur. Dans ce cas, on peut utiliser le mode quatre bits de l'afficheur LCD. Dans ce mode, seuls les 4 bits de poids

fort (D4 à D7) de l'afficheur sont utilisées pour transmettre les données et les lire. Les 4 bits de poids faible (D0 à D3) sont alors connectés à la masse. On a donc besoin, hors alimentation de sept fils pour commander l'afficheur. Les données sont alors écrites ou lues en envoyant séquentiellement les quatre bits de poids fort suivi des quatre bits de poids faible. Une impulsion positive d'au moins 450 ns doit être envoyée sur la ligne E pour valider chaque demi-octet ou nibble.

- **Mode 8 bits :**

Dans ce mode 8 bits, les données sont envoyées à l'afficheur sur les broches D0 à D7**.** On place la ligne RS à 0 ou à 1 selon que l'on désire transmettre une commande ou une donnée. Il faut aussi placer la ligne R/W à 0 pour indiquer à l'afficheur que l'on désire effectuer une écriture. Il reste à envoyer une impulsion d'au moins 450 ns sur l'entrée E, pour indiquer que des données valides sont présentes sur les broches D0 à D7. L'afficheur lira la donnée sur le front descendant de cette entrée.

Si on désire au contraire effectuer une lecture, la procédure est identique, mais on place cette fois la ligne R/W à 1 pour demander une lecture. Les données seront valides sur les lignes D0 à D7 lors de l'état haut de la ligne E.

- Dans les deux modes, on peut après chaque action sur l'afficheur vérifier que celui-ci est en mesure de traiter l'information suivante. Pour cela, il faut demander une lecture en mode commande, et tester le flag Busy BF. Lorsque BF=0, l'afficheur est prêt à recevoir une nouvelle commande ou donnée.

Il se peut qu'on dispose encore de moins de broches disponibles dans l'application envisagée. Dans ce cas, on peut alors relier la ligne R/W à la masse de façon à forcer l'afficheur en écriture. On a alors besoin, hors alimentation de seulement six fils en mode 4 bits, et dix fils en mode 8 bits, pour commander l'afficheur, mais on ne peut alors plus relire l'afficheur. Ceci n'est pas gênant dans la mesure où on sait ce qu'on a écrit sur l'afficheur, mais on ne peut alors plus relire le flag Busy. Il faut alors utiliser des temporisations après chaque écriture sur l'afficheur. On perd alors un peu en temps d'affichage, mais on gagne une broche d'entrée sortie.

III.2.1. Le brochage de LCD vers l'Arduino :

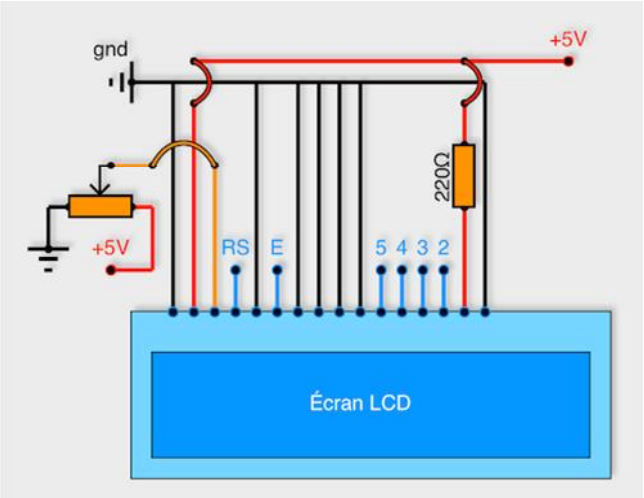


Figure III.2. Le brochage de l'afficheur avec l'Arduino

Voici à quoi servent les pins de gauche à droite :

- Les deux premiers pins tout à gauche servent à l'alimentation de l'écran.
- Le troisième pin est connecté à un potentiomètre et sert pour régler l'affichage (le contraste de l'écran).
- Le quatrième, noté RS, est connecté au pin 12 de l'Arduino dans notre exemple. Il sert à sélectionner la zone mémoire de l'écran LCD dans laquelle nous allons écrire (Register Select).
- Le cinquième doit toujours être connecté à la masse. C'est un sélecteur de mode lecture ou écriture. On peut le connecter à un pin, mais dans notre cas c'est inutile. Comme il doit recevoir un signal à 0V, on le connecte à la masse (état R/W).
- Le sixième, noté E, est connecté au pin 11 de l'Arduino dans notre exemple. Il permet de lancer ou non l'écriture dans les zones mémoires.
- Les quatre suivants (reliés à la masse) servent pour la communication 8 bits. Pour la communication 4 bits, il est conseillé de les relier à la masse. Ils représentent les bits de poids fort.
- Les quatre qui suivent, notés 2, 3, 4, 5, se connectent dans notre exemple sur les pins 2, 3, 4, 5 de l'Arduino. Ils servent pour la communication (8 bits ou 4 bits)

et doivent toujours être connectés. Ils représentent les bits de poids faible (ou servent pour envoyer d'abord les bits de poids faible, puis les bits de poids fort)

- Les deux pins tout à droite servent pour alimenter la LED du rétro-éclairage.

En mode 4 bits, il est finalement important de bien repérer les 6 pins en bleu sur le schéma.

III.3 Le potentiomètre :

Le potentiomètre est un type de résistance variable sa valeur résistive change selon la position d'un curseur qui se déplace sur une piste au carbone grâce à un axe de rotation que l'on vient manipuler manuellement ou un tournevis.

Il possède trois bornes:

Deux correspondent aux extrémités du corps de la résistance et la borne centrale correspond au curseur qui peut se déplacer sur le corps de la résistance.

Il existe plusieurs formes et tailles de potentiomètres, mais ils peuvent être classées en deux grandes catégories : potentiomètres ajustables et potentiomètres de tableau.

Dans ce travaille on va utiliser le potentiomètre ajustable de 4.7k.

La broche V0 de l'afficheur LCD 16x2 permet de régler le contraste et la luminosité. Il est nécessaire d'y connecter un potentiomètre de réglage, dont les broches externes soient connectées à l'alimentation (5V) et à la masse (GND) et la broche centrale à V0 de l'écran LCD. Il suffit ensuite de tourner le potentiomètre dans tous les sens avec patience jusqu'à ce que le contraste soit correctement réglé.

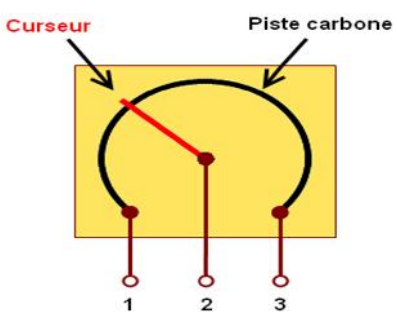


Figure III.3. Potentiomètre ajustable

Pour permettre à l'utilisateur de donner des commandes à notre prototype on a préféré d'ajouter un clavier de type 4x4 (4 ligne ,4 colonnes).

III.4 Clavier :

Le clavier matriciel ou le clavier à membrane (constitué de multiples couches de polyester imprimé) représente la liaison idéale pour le dialogue homme-machine.

Ce clavier dispose de 16 touches, disposées dans une grille de quatre colonnes et de quatre lignes. Il est fait d'un matériau mince formant une membrane flexible avec un support adhésif permettant de le placer presque n'importe où.

Et dans notre cas il s'agit donc d'un petit clavier comportant 16 touches: en plus des chiffres, les 4 premières lettres de l'alphabet, ainsi que les symboles "*" et "#". 8 connecteurs permettent de le brancher à l'Arduino. Au moyen d'un multimètre, on a rapidement constaté que les 4 premiers connecteurs sont reliés aux lignes alors que les 4 derniers sont reliés aux colonnes. Par exemple, si on appuie sur la touche "6" (deuxième ligne, troisième colonne), la résistance devient nulle entre les connecteurs 2 et 7 et on aura une continuité dans ce point.

- Le clavier matriciel et son schéma interne :

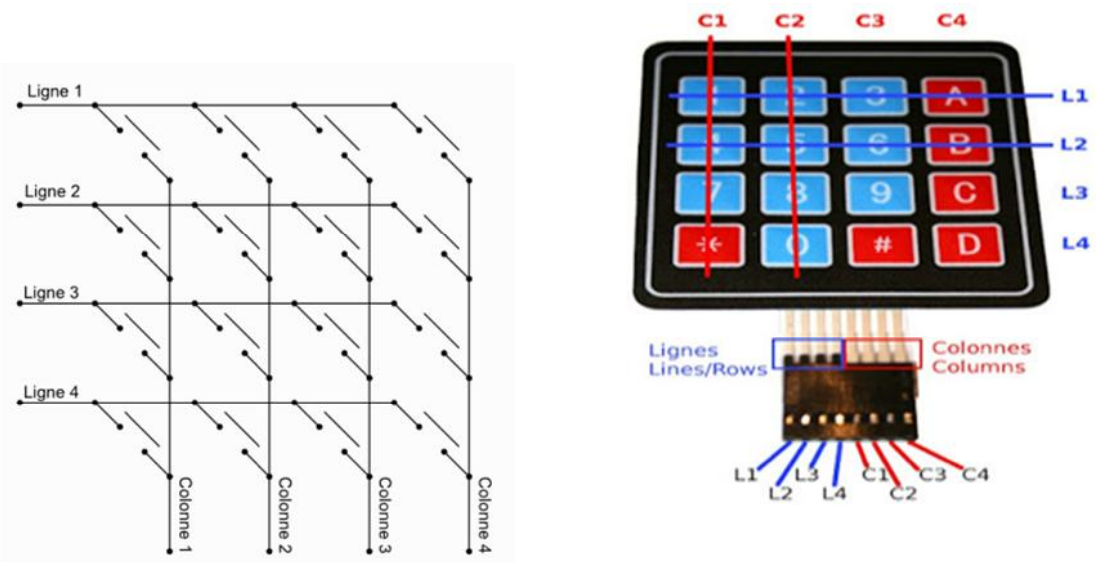


Figure III.4. Le clavier matriciel 4x4 et son schéma interne

III.4.1 Le brochage du clavier vers la carte Arduino :

Clavier	Arduino
Ligne 1	Pin 11
Ligne 2	Pin 10
Ligne 3	Pin 9
Ligne 4	Pin 8
Colonne 1	Pin 7
Colonne 2	Pin 6
Colonne 3	Pin 5
Colonne 4	Pin 4

Tableau III.2. Brochage du clavier avec la carte Arduino

Le logiciel Arduino possède une bibliothèque appelée « Keypad » qui gère le clavier matriciel, cette bibliothèque dispose de plusieurs fonctions qui gèrent l'initialisation et le renvoie d'état courant du clavier. [8], [9]

III.5 Plaque d'essai et fil :

La méthode pour tester un montage électronique sans réaliser de circuit imprimé consiste à utiliser une plaque d'essai. Grâce à ce petit outil, il n'y a pas besoin de souder, il suffit juste de placer les composants sur la plaque de test.

Cette plaquette est composée d'un plastique isolant avec des rangées verticales de 5 contacts et 4 lignes horizontales pour l'alimentation.

Les lignes rouges pour relier les composants au + et les lignes bleus pour le -.

Les composants sont plantés dans les trous de diamètre 0,8mm. Il ne faut pas essayer de planter des composants ayant des connexions du diamètre supérieur.

Le bloc principal permet de placer des composants électroniques typiques comme un circuit intégré au milieu sans court-circuiter ses broches.

La plaque d'essai permet de placer les composants, mais reste insuffisante pour la totalité des liaisons électriques. Il est donc nécessaire d'utiliser des fils électriques.

Ces fils de liaison sont réalisés avec du fil électrique isolé que l'on dénude aux extrémités.

Dénuder un fil consiste à enlever la gaine isolante. Si le fil est de type multibrin, il est nécessaire de torsader la partie dénudée et de l'étamer avant de pouvoir l'utiliser.

Il est pratique d'avoir du fil monobrin et du fil multibrin. Le fil monobrin est rigide, donc pratique pour les liaisons permanentes. Le fil multibrin est souple, donc plus pratique pour les modifications lors de l'étude.

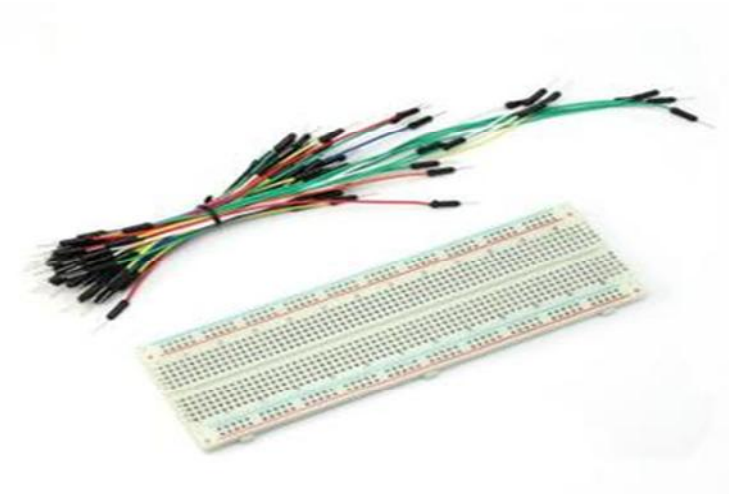


Figure III.5. Plaque d'essai et fil

III.6 Connexion du module GSM à Arduino:

D'abord on doit alimenter le module GSM avec 12V et 1A, puis on va insérer la carte SIM sur son support, et la connecter vers l'Arduino. Il y a deux façons de connecter le module GSM à Arduino. Dans tous les cas, la communication entre Arduino et le module GSM est en série. On est donc supposés utiliser des pins séries d'Arduino (Rx et Tx). Donc, si on irait avec cette méthode, on peut connecter la broche Tx du module GSM à la broche Rx d'Arduino et la broche Rx du module GSM à la broche Tx d'Arduino. Aussi, il faut connecter la broche de terre d'Arduino à la broche de masse du module GSM. On a fait 3 connexions et le câblage est terminé, on peut charger différents programmes pour communiquer avec le module GSM et le faire fonctionner.

Le problème avec cette connexion est que, lors de la programmation Arduino utilise des ports série pour charger le programme de l'IDE Arduino. Si ces broches sont utilisées dans le câblage, le programme ne sera pas chargé avec succès dans Arduino. On doit donc déconnecter le câblage dans RX et TX chaque fois qu'on grave le programme sur Arduino. Une fois le programme chargé avec succès, on peut reconnecter ces broches et faire fonctionner le système.

Pour éviter cette difficulté, on utilise une autre méthode dans laquelle deux broches numériques d'Arduino sont utilisées pour la communication série. On doit sélectionner deux broches d'Arduino compatibles PWM pour cette méthode. Donc, on choisit les broches 9 et 10 (qui sont des broches compatibles PWM). Cette méthode est rendue possible avec la Software Serial Library d'Ardunio. SoftwareSerial est une bibliothèque d'Arduino qui permet la communication de données en série à travers d'autres broches numériques d'Arduino. La bibliothèque reproduit les fonctions matérielles et gère la tâche de communication série. [10]

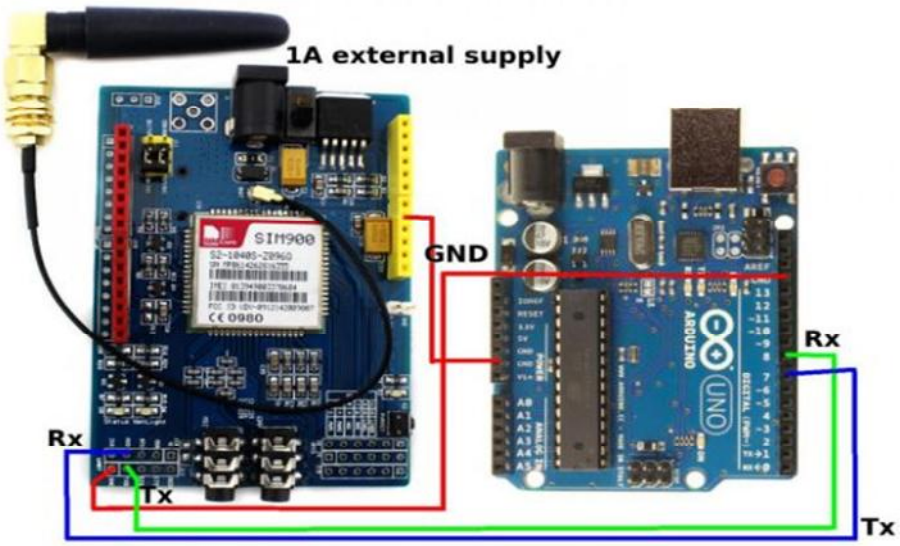


Figure III.6. Connexion GSM et Arduino

III.7 Réalisation :

Dans ce projet de téléphone portable à base d'Arduino, on a utilisé Arduino UNO pour contrôler les fonctionnalités du système entier et l'interfaçage de tous les composants de ce système.

Un clavier alphanumérique 4x4 est utilisé pour prendre toutes sortes d'entrées comme: Entrer le numéro de téléphone mobile, taper des messages, appeler, recevoir un appel, envoyer des SMS etc. Le module GSM est utilisé pour communiquer avec le réseau. On a également interfacé un écran LCD 16x2 pour afficher des instructions et des alertes.

- ❖ L'explication de quatre caractéristiques du téléphone portable Arduino:

1. Faire un appel : Pour passer un appel en utilisant notre téléphone basé sur Arduino, on doit appuyer sur «A » et ensuite besoin d'entrer le numéro de portable sur lequel on veut faire un appel. Le numéro sera entré en utilisant le clavier 4x4 et s'affiche sur l'écran LCD. Après avoir entré le numéro, on doit appuyer sur «#» pour effectuer l'appel.

2. Recevoir un appel : Lorsque quelqu'un appelle au numéro de la carte SIM de notre système, qui est présente dans le module GSM, notre système affichera le message 'RING' sur le moniteur série. On a juste besoin d'appuyer sur la touche «C», après une

second on doit appuyer sur «*» du clavier pour avoir répondre à cette appel ou bien pour raccroché on va appuyer sur «#».

3. Envoyer un SMS : Lorsqu'on doit envoyer un SMS en utilisant notre téléphone mobile basé sur Arduino, on doit appuyer sur «B». Le système demandera le numéro de destinataire, ce qui signifie à qui on doit envoyer des SMS. Après avoir entré le numéro, on doit appuyer sur «#» pour envoyer un SMS.

4. Recevoir et lire des SMS : Dans ce cas, GSM recevra des SMS et les stockera dans la carte SIM. Et Arduino surveille en permanence l'indication SMS reçue via UART. On a juste besoin d'appuyer sur «*», pour lire le SMS, quand on voit nouveau message sur l'écran LCD.

Schéma du circuit :

La figure suivante indique le schéma du circuit.

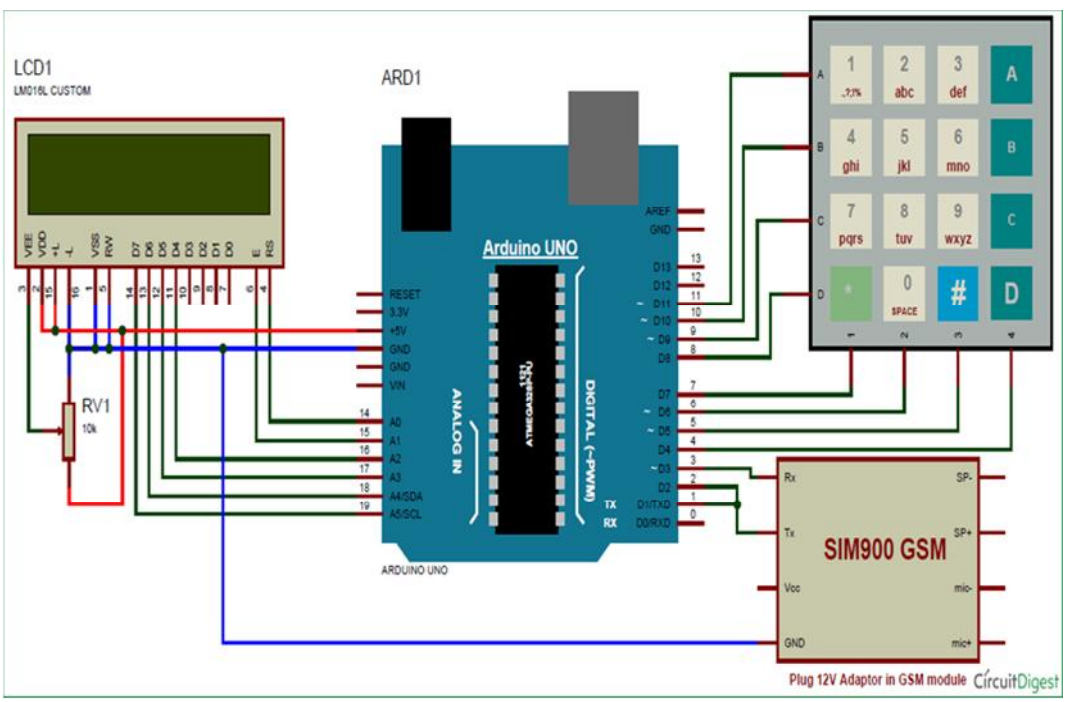


Figure III.7. Schéma du circuit de téléphone portable à base d'Arduino

Explication de programmation: Dans le logiciel Arduino, l'écriture du programme est réalisée en trois parties.

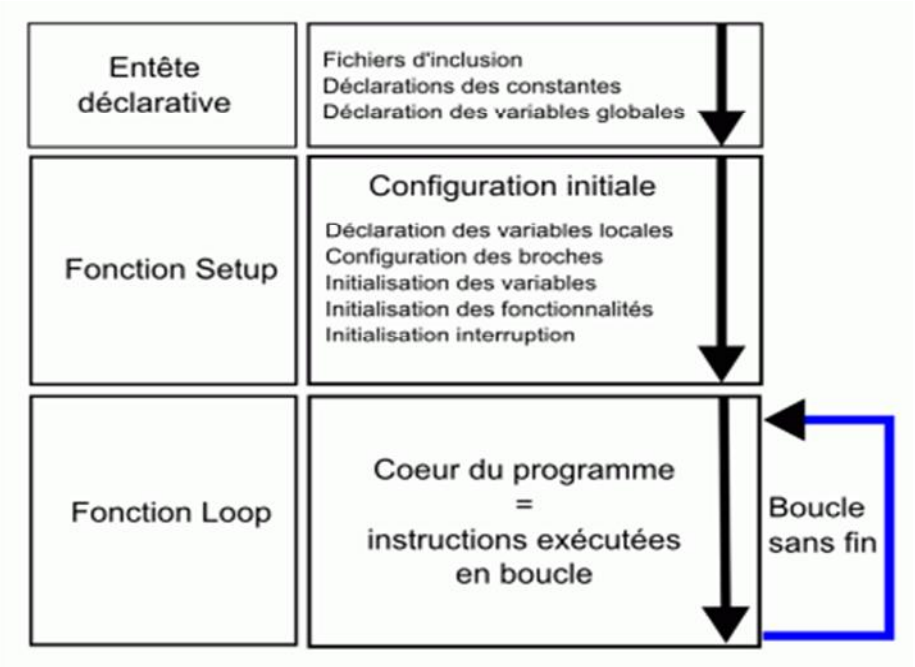


Figure III.8. L'organigramme du programme

Dans le cas de notre application on donne :

- **Partie déclarative :**

On commence par inclusion les librairies SoftwareSerial de carte GSM, LiquidCristal de l'afficheur LCD 16x2 et la librairie de keypade de clavier4x4. On déclare les constantes utiles et les broches et les variables de chaque composant utilisé.

- **Partie fonction setup () :**

Cette fonction setup () permet d'accéder à la partie configuration du programme. La première tâche consiste à l'initialisation de l'utilisation du l'afficheur LCD alphanumérique. La deuxième tache définir les débits en baud de la librairie SoftwareSerial pour communiquer avec le module GSM. On y parvient en invoquant la fonction mySerial.begin. Ensuite, on doit définir le débit en bauds du moniteur série d'Arduino IDE. On fait cela en invoquant la fonction Serial.begin. Les deux devraient être

réglés à la même vitesse de transmission et on utilise 9600 bits / seconde. La partie de configuration est terminée avec les taux de bauds de réglage et son bon pour donner un petit retard de 100 millisecondes.

- **Partie fonction loop ():**

On passe maintenant au programme à l'intérieur de loop () une partie constituée d'une boucle sans fin que le programme répètera à l'infini (fonction loop( )) : c'est le cœur du programme.

Serial.available () : vérifie toutes les données provenant du port série d'arduino. La fonction renvoie le nombre d'octets disponibles pour la lecture à partir du tampon série.

Serial.read () : Lit toutes les données disponibles sur le tampon série (ou les données série entrantes si elles seraient définies autrement). Il renvoie le premier octet des données série entrantes.

mySerial.available () : vérifie toutes les données provenant du module GSM via les broches SoftwareSerial Rx et Tx. Renvoie le nombre d'octets disponibles pour la lecture à partir du port série du logiciel. mySerial.read () - Lit les données entrantes via le port série du logiciel.

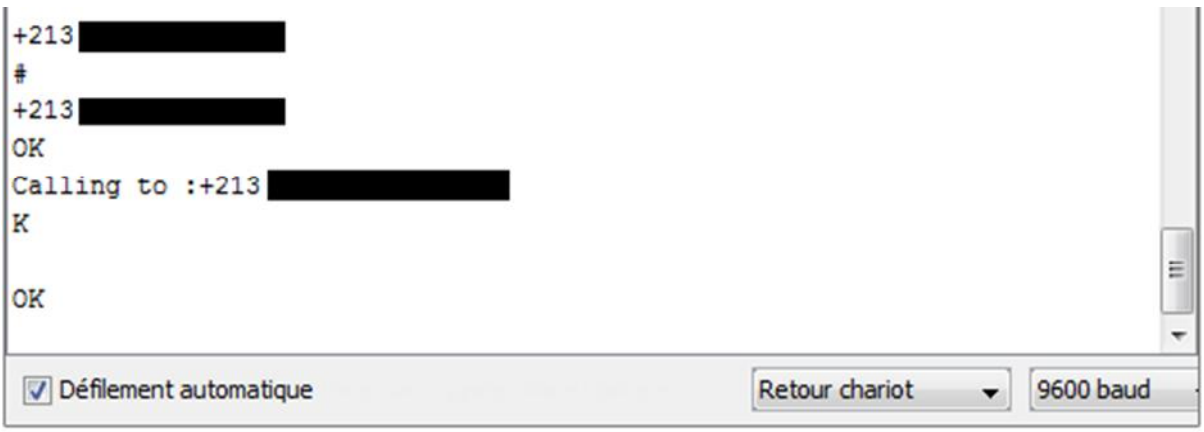
Serial.write () : Imprime les données sur le moniteur série d'Arduino. Ainsi, la **fonction Serial.write (mySerial.read ()) :** imprime les données collectées depuis le port série du logiciel vers le moniteur série d'Arduino.

Pour permettre d'obtenir les fonctions SendCall (), RecieveCall (), SendMessage () et RecieveMessage () on envoie réellement des commandes au module GSM d'arduino. Ces commandes sont appelées commandes AT. Il existe différentes commandes pour effectuer différentes tâches à l'aide du module GSM.

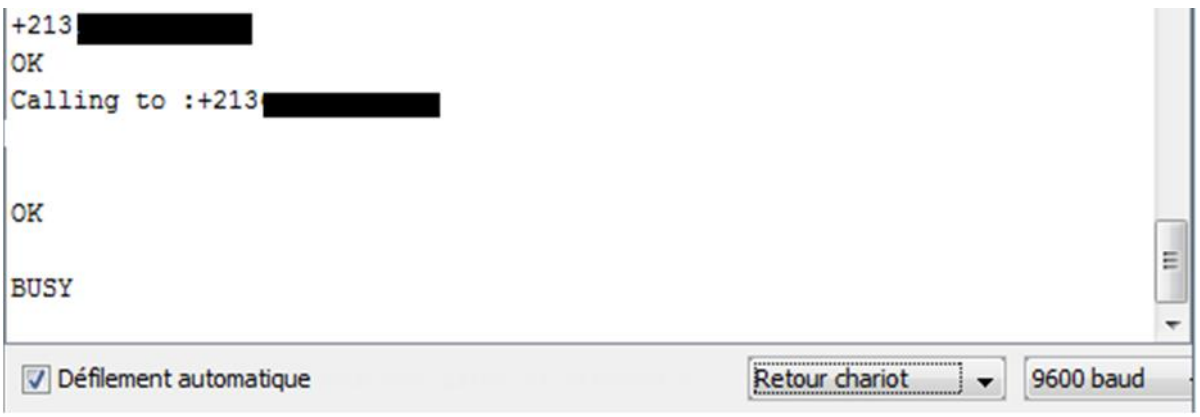
SendCall () : pour faire un appel d'abord on doit désactiver le verrouillage du code PIN de la carte SIM avant de l'insérer sur le support du module GSM, pour se connecter sur un réseau mobile.

On envoie cette commande AT "ATD+213xxxxxxxxx;". On remplace les xxx par le numéro du destinataire pour effectuer l'appel. Durant l'établissement de la communication, le module SIM900 affiche spontanément différentes informations concernant le statut de la communication vocale. La communication soit établie, refusée ou fin d'appel, on peut obtenir les messages spontanés suivants sur le moniteur série:

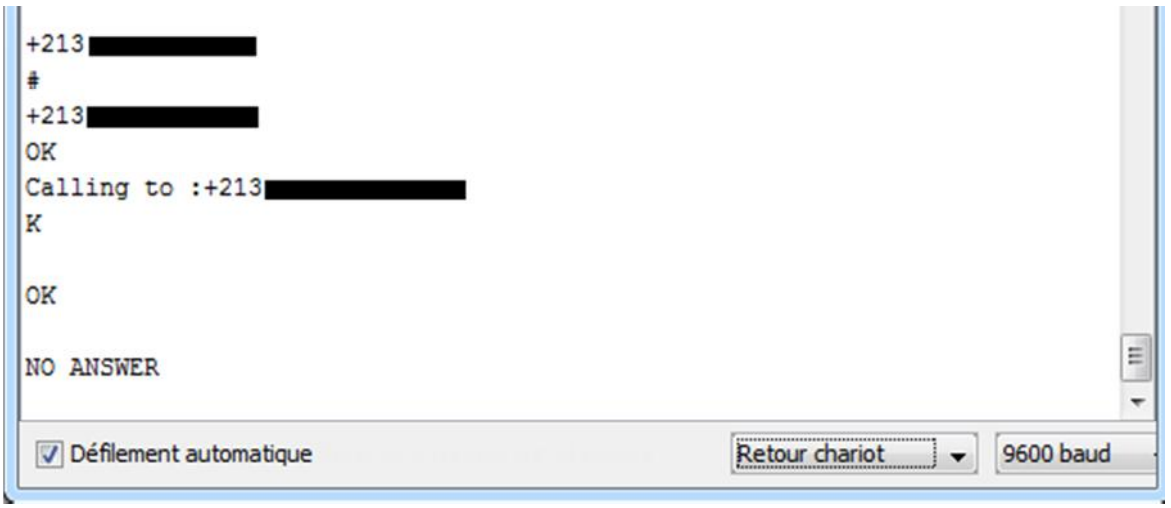
- Ok lorsque la communication est établie.



- Busy lorsque l'appel est refusé (appel décroché).

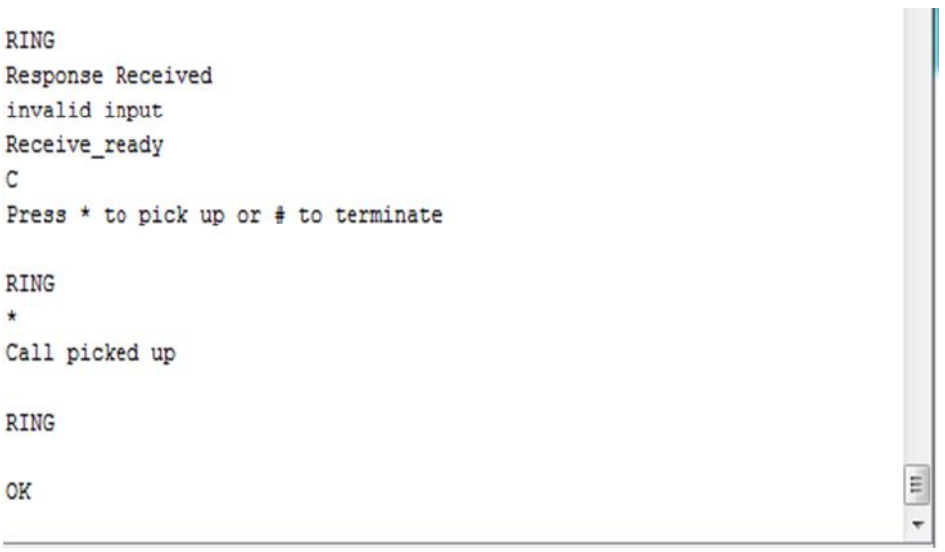


- NO ANSWER lorsque l'appel est terminé (appel non décroché, fin d'appel).

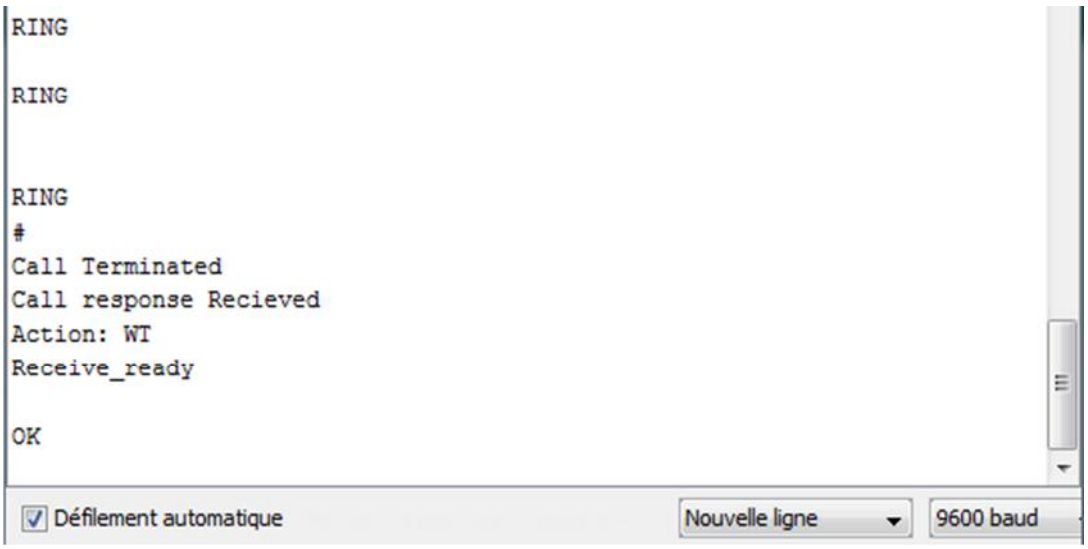


ReceiveCall () : Quand quelqu'un appelle au numéro du module GSM, il nous affiche un message "RING" sur le moniteur série pour savoir si quelqu'un appelle.

- Pour répondre on a utilisé la commande ATA.

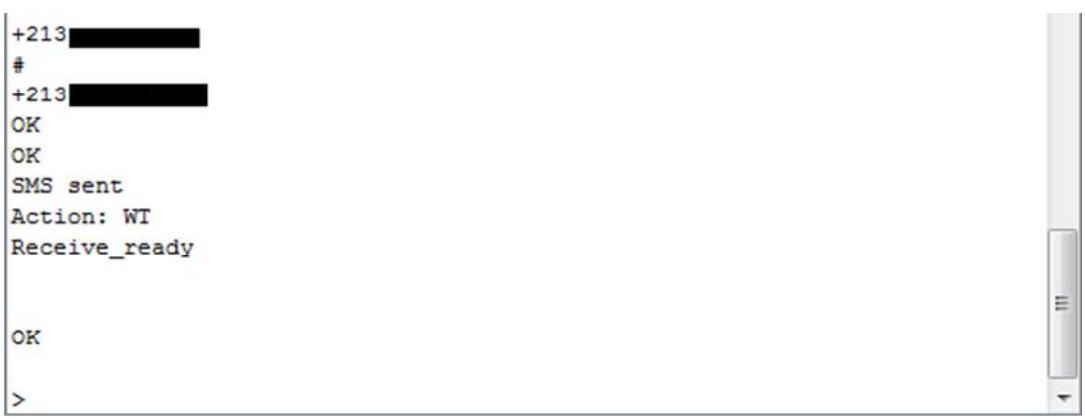


- Pour interrompre un appel téléphonique, on a utilisé la commande ATH.



SendMessage () : la fonction qu'on a créée dans notre programme Arduino pour envoyer un SMS. On doit d'abord mettre notre module GSM en mode texte. Ceci est réalisé en envoyant une commande AT "AT + CMGF = 1", après on devrait écrire le numéro du portable auquel on enverra le SMS. Ceci est réalisé avec la commande AT "AT + CMGS = " + 213xxxxxxxxx". On remplace les xxx par le numéro de destinataire.

À l'étape suivante, on devrait envoyer le contenu de SMS. La fin du contenu SMS est identifiée par le symbole CTRL + Z. La valeur ASCII de ce CTRL + Z est 26. On envoie donc un caractère (26) au module GSM en utilisant la ligne mySerial.println ((char) 26); Chaque commande AT peut être suivie d'un délai d'une seconde. On doit laisser du temps au module GSM pour répondre correctement. Une fois ces commandes envoyées au module GSM, on recevra un SMS dans le numéro de mobile défini.



RecieveMessage () : la fonction pour recevoir un SMS, et la commande AT pour recevoir un SMS en direct est "AT + CNMI = 2, 2, 0, 0, 0" on a juste besoin d'envoyer cette commande au module GSM et d'appliquer un délai d'une seconde. Une fois qu'on a envoyé cette commande, on essaye d'envoyer un SMS au numéro de la carte SIM placé dans le module GSM. On voit le SMS qu'on a reçoit afficher sur l'écran LCD et le moniteur série.

Quand on reçoit un SMS, le module SIM900 envoie spontanément un message similaire a +CMT en deux lignes, on peut identifier le statut du message, le numéro de l'émetteur (+213xxxxxxxxx) ainsi que la date et l'heure.

+CMT: servira de détecteur pour identifier la réception du message dans série moniteur.

La carte SIM dispose de 25 emplacements pour stocker les SMS en cours de réception. Une fois le message traité, l'emplacement mémoire devrait être libéré à l'aide de la commande AT.

Discussion :

Dans ce chapitre la réalisation pratique de notre travail a été présenté, dans lequel différents tests ont été effectué, premièrement séparément pour vérifier le fonctionnement des différents modules, puis assemblé pour tester le fonctionnement de tout le système. Le résultat obtenu est très satisfaisant, tous les tests ont réussi avec succès, notamment test complet du notre système.

CONCLUSION

Conclusion

L'objectif de notre projet est de réaliser un téléphone portable simple à base d'Arduino. Pour ce faire, nous avons utilisé une carte Arduino UNO qui est un composant essentiel, à laquelle nous avons connecté un module GSM, un clavier matriciel 4*4 et un afficheur LCD 16*2.

Le module GSM a été utilisé pour effectuer un appel, recevoir un appel, envoyer et lire des SMS**.** Le clavier a servi pour former des numéros ou pour écrire du texte dans le cas de la transmission de SMS. Quant à l'afficheur, il est utilisé pour visualiser les numéros et les messages

Nous avons présenté dans le premier chapitre les notions de base de module GSM, ensuite dans le deuxième chapitre, nous avons réalisé une étude sur la carte Arduino, qui est la partie commande du ce projet. Dans le troisième chapitre, nous avons présenté notre réalisation où des détails sur le fonctionnement ont été mis en évidences.

Selon les tests que nous avons réalisé, à savoir émettre un appel ou recevoir un appel, nous avons constaté que notre application a bien fonctionné. Lors de l'émission de l'appel, le programme fait appel à la commande ATD qui est une commande intégrée dans la bibliothèque de Arduino. A la réception d'un appel, le programme utilise la commande ATA pour répondre à l'appel. Pour refuser l'appel, la commande ATH est utilisée.

Dans le cas des SMS, le programme utilise la commande AT+CMGF qui doit être à 1 pour mettre le module GSM en mode texte. A l'émission et à la réception, le programme utilise les commandes AT+CMGS et AT+CNMI respectivement.

Une des applications qui peut être envisagée comme perspective de ce travail est la réalisation d'un smartphone, en augmentant ainsi le nombre de services qui peut être assuré, à savoir accès à l'internet, affichage des images et les vidéos.

BIBLIOGRAPHIE

Bibliographie

[1] Réseaux GSM, 5ème édition revue et augmentée. Xavier Lagrange, Philipe Godlewski, Samir Tabbane, année: 2000.

[2] Téléphone GSM et PC, 3ème édition. Edition ETSF-Patrick Gueulle, année : 2006.

[3] Le Grand livre d'ARDUINO, 2ème édition, Erik Bartmann, année : 2014.

[4] www.arduino.cc. Mars 2018

[5] http://www.arduino.cc/en/Guide/ArduinoGSMShield.html. Mars 2018

[6] http://www.snif.orgouk/2014/10/Arduino-phone-homme-htmel. Avril 2018

[7] https://www.arrow.com/en/products/a000066/arduino-corporation. juin 2018

[8] www.instructables.com/id/Arduino-Keypad-4x4-Tutoriel/. Juin 2018

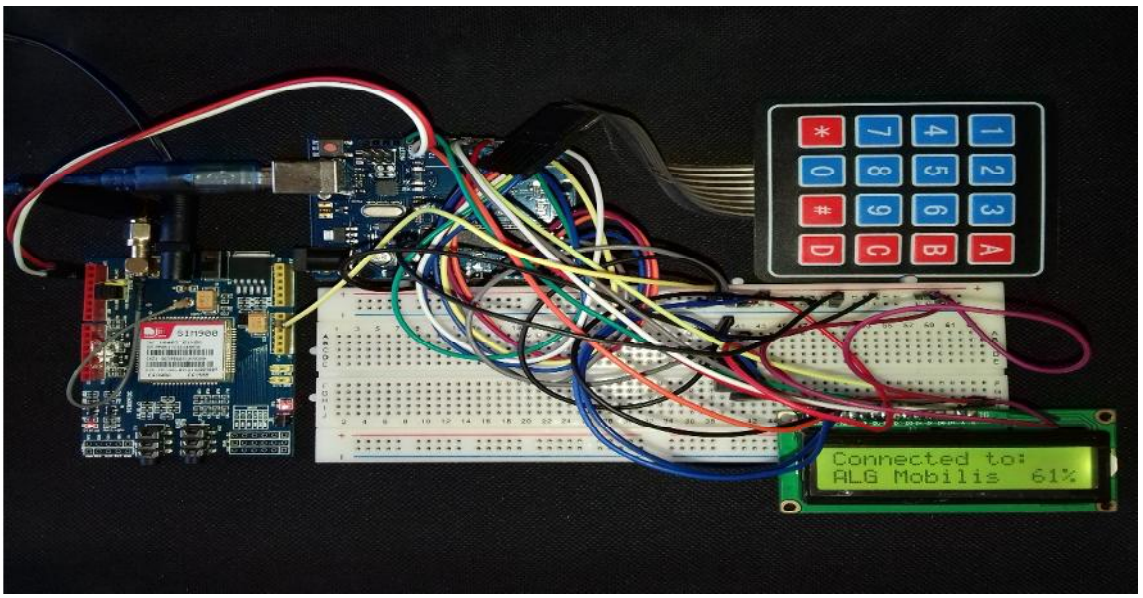
[9] https://circuitdigest.com/microcontroller-projects/keypad-interfacing-with-arduino-uno. juin 2018

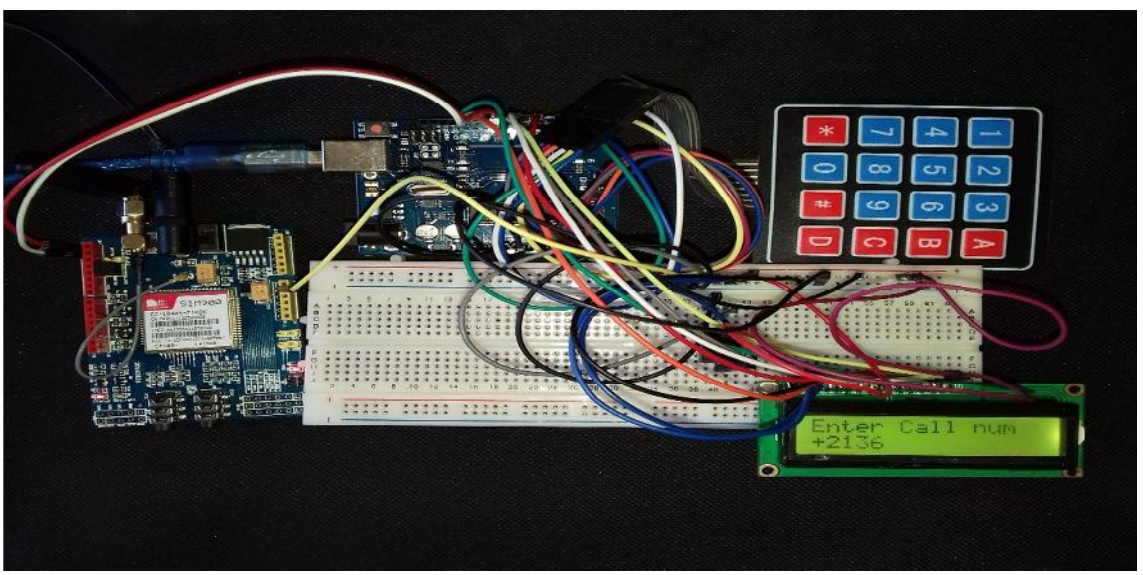
[10] http://www.ulg.ac.be/telecom. juin 2018

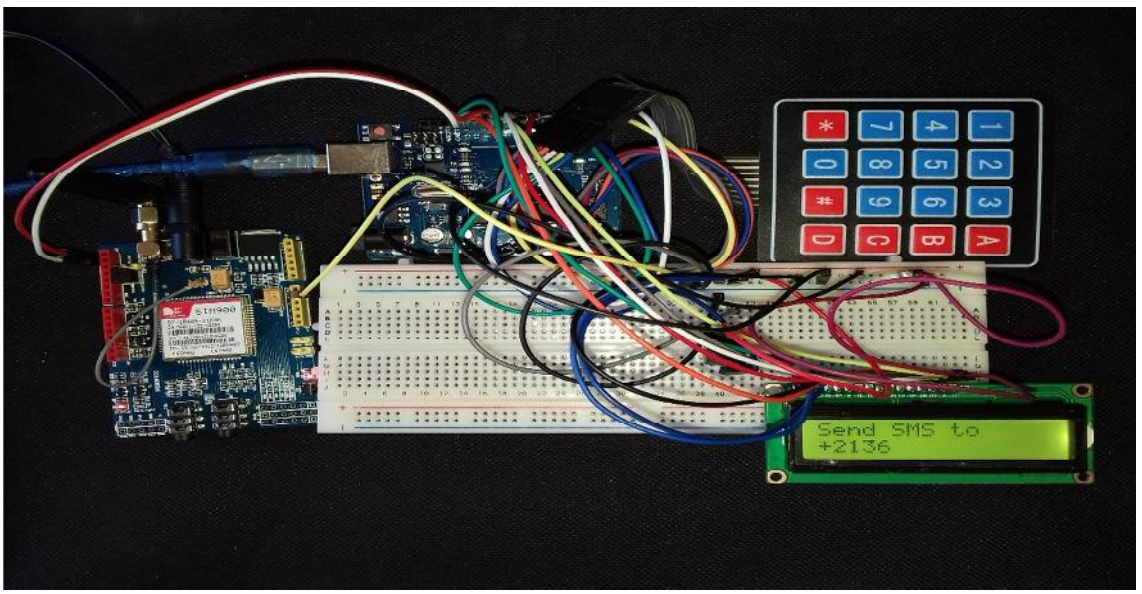
[11] https://shop.mchobby.be/shields/63-shield-gsm-gprs-pour-arduino.html. Mai 2018

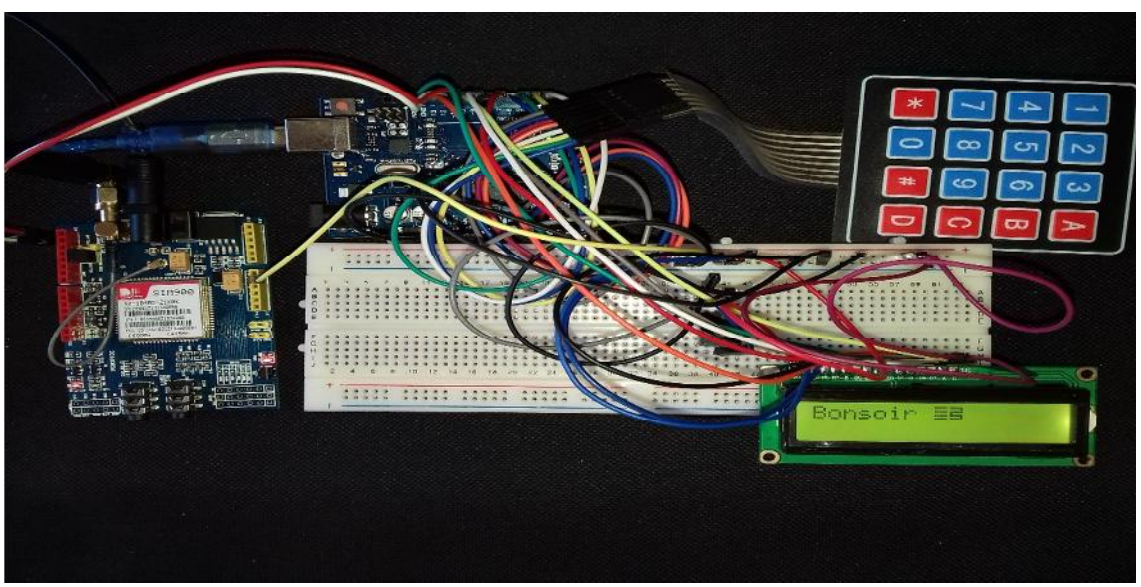
ANNEXE

Les figures de notre réalisation






Send SMS to
+2136


Bonsoir ES