

Liste des Abréviations

HT: Haute Tension

MT: Moyenne Tension.

BT: Basse Tension.

CPL: Courants porteurs en ligne .

GPRS : General Packet Radio Service. Service de transmission de données en mode paquet.

GSM: Global System for Mobile Communication. Système global pour les communications mobiles).

GRD : Gestionnaire de réseaux de Distribution.

GRT : Gestionnaire de réseaux de transport.

CBA : Centre de formation de Sonelgaz

CRDEDEG : (Centre de Recherche et de Développement de l'Électricité et du Gaz)

SDA : Société de distribution de l'électricité et du gaz d'Alger.

HTA: Haute Tension A.

HTB: Haute Tension B.

MCU: Memory Controller Unit.

UART: Universal Asynchronous Receiver/Transmitter.

EDI : Electronic Data Interchange ou, en français, Echange de Données Informatisées.

LCD : Liquid Crystal Display.

GUI : Graphical User Interface, « interface graphique », en informatique.

CAO : La conception assistée par ordinateur.

PDA : Personal Digital Assistant.

JVM : Java Virtual Machine.

JDK : Java Development Kit.

JRE : Java Runtime Environment.

SDK : Software development kit.

APK : Android Package.

API : Application Programming Interface.

XML : Extensible Markup Language.

Table des matières

Remerciements	i
Dédicaces	ii
Résumé	iv
Table des matières	v
Liste des figures	viii
Liste des abréviations	x
Introduction générale	xii

Chapitre I : Etat de l'art des compteurs électriques

Introduction	14
I.1. Compteur intelligent	14
I.2. Architecture d'un compteur intelligent	15
I.3. Présentation d'un modèle de système de comptage évolué	16
I.4. Principales fonctionnalités des systèmes de comptage évolués	18
I.5. Réseau électrique intelligent	18
I.5.1. Définition	18
I.5.2. Fonctionnement	19
I.5.3. Types de réseau électrique intelligent	19
I.5.4. Les limites dans la mise en œuvre	20
I.6. Réseaux électriques intelligents en Algérie	21
I.7. L'Algérie et les compteurs intelligents	22
Conclusion	24

Chapitre II : Réalisation d'un compteur intelligent

Introduction.....	26
II.1. Schéma du prototype.....	27
II.1.1. Alimentation.....	27
II.1.1.1. Blocs constituant une alimentation stabilisée.....	28
II.1.1.2. Schéma électronique de l'alimentation su circuit.....	29
II.1.2. Circuit de mesure.....	31
II.1.2.1. Définition.....	31
II.1.2.2. Élément de mesure.....	32
II.1.3. Circuit d'acquisition et traitement.....	37
II.1.3.1. Définition.....	37
II.1.3.2. Caractéristiques.....	38
II.1.3.3. Avantages.....	39
II.1.3.4. Alimentation de l'Arduino.....	39
II.1.3.5. Entrées et sorties.....	40
II.1.3.6. Communication.....	40
II.1.3.7. Logiciel.....	41
II.1.3.8. Structure d'un programme Arduino.....	43
II.1.3.9. Le Moniteur Série.....	43
II.1.3.10. Librairies.....	44
II.1.3.11. Transmission et affichage.....	44
II.1.3.12. Bloc de traitement.....	48
II.1.3.13. Simulation du circuit électronique.....	48
II.2. Interprétation des résultats.....	50
II.2.1. Capteur de tension.....	50
II.2.2. Capteur de courant.....	51
Conclusion.....	53

Chapitre III : Conception d'une application Android

Introduction.....	55
III.1. Historique	56
III.2. Etude de la plateforme Android	57
III.2.1. Architecture de la plateforme Android	57
III.3. Le Développement sous Android	58
III.3.1. L'environnement de développement sous Android	58
III.3.2. Langage JAVA	58
III.3.3. Le SDK (Software Development Kit) Android.....	59
III.3.4. Le JDK (Java Développement Kit)	60
III.4. Architecture d'un projet sous Android	60
III.5. Composants principaux d'une application Android.....	62
III.6. Cycle de vie d'une application Android	62
III.7. Création d'un projet Android	64
III.8. Conception de l'application	67
Conclusion	76
Conclusion générale	78
Reference bibliographique	80

Résumé

L'avancement des technologies à aider l'amélioration des systèmes du réseau électrique par l'intégration des énergies renouvelables ainsi que le développement des compteurs qui sont la base des réseaux intelligents (Smart Grid). L'intérêt principal de ces compteurs est de faire le suivi des données en temps réel.

Notre but est d'étudier et réaliser un compteur dit intelligent. La carte électronique réalisé est piloté par un microcontrôleur ATmega 2560 d'un Arduino, cette dernière permet aussi la transmission des données vers un Smartphone via un système de communication sans fil de type Bluetooth. Les données reçus sont par la suite visualisé à l'aide d'une application réalisée sous Android studio et enfin enregistrer sur une carte SD pour une utilisation ultérieur.

Mot clés : Réseaux intelligent, compteur intelligent, Arduino, Bluetooth, Android.

Abstract

The advancement of technologies has helped to improve the electricity grid systems by integrating renewable energies and developing the meters that are the basis of smart grids. The main interest of these meters is to monitorate the data in real time.

Our purpose is to study and realize a device that is a so-called smart meter .The electronic board is controlled by an ATmega 2560 microcontroller from an Arduino The latter also allows the transmission of data to a Smartphone via a wireless communication system defined Bluetooth. The received data is then viewed using an application built under Android studio and finally saved on an SD card for later use.

Keywords : Smart Grid, smart meter, Arduino, Bluetooth, Android.

ملخص

التقدم في التكنولوجيات ساعد في تحسين أنظمة الشبكة الكهربائية عبر إدماج الطاقات المتجددة و تطوير العدادات الإلكترونية التي تعتبر أساس الشبكة الكهربائية الذكية. تكمن الفائدة الرئيسية لهذه العدادات في القيام بالمتابعة و المراقبة الحينية للمعلومات .

هدفنا في هذه المذكرة يتضمن دراسة و إنشاء عداد ذكي. هذا الجهاز متحكم فيه عن طريق معالج من نوع ATMEGA 2560 لبطاقة أرد وينو ، هذه الأخيرة تسمح بإرسال المعلومات نحو الهاتف الذكي بطريقة لا سلكية باستخدام تقنية البلوتوث، تعرض هذه المعلومات على شاشة الهاتف عبر برمجة تطبيق أندرويد و بالتالي حفظ المعلومات على بطاقة ذاكرة إلكترونية لاستعمال لاحق.

الكلمات المفتاحية : الشبكة الذكية, عداد ذكي, أردوينو, بلوتوث, أندرويد.

Introduction générale

Introduction générale

Avec le développement technologique les dernières années, l'intelligence ambiante tient une place particulièrement importante dans notre vie, et a pour but de créer un espace quotidien intelligent immédiat d'utilisation, intégré dans nos maisons, nos bureaux, nos routes, et nos voitures. Ce nouveau concept doit être invisible, il doit en effet se fondre dans notre environnement quotidien et doit être présent au moment où nous en avons besoin.

Tels qu'on connaît, les réseaux électriques sont restés à leur état d'origine et n'ont subi que peu de modification, mais ce développement technologique et l'évolution croissante des besoins de l'humanité en matière d'applications qui concernent ces réseaux a conduit progressivement à considérer des changements satisfaisant des demandes et des performances importantes.

Cette révolution technologique cherche à intégrer l'informatique et les télécommunications aux infrastructures électriques grâce à des compteurs intelligents qui sont l'un des leviers de la transformation des réseaux de distribution électrique en réseaux intelligents, ou smart grids.

Au fond, le compteur intelligent est un compteur énergétique capable de suivre en détail, et souvent en temps réel, la consommation électrique d'un bâtiment, d'une entreprise ou d'un foyer. Ce compteur intelligent est en outre communicant et transmet par différents canaux (courant porteur, Internet, téléphone) les informations recueillies.

L'objectif principal de ce mémoire est d'étudier et réaliser un système de communication pour un compteur intelligent ; pour y aboutir, nous avons suivi le plan suivant :

Nous commençons en premier chapitre par des généralités sur les réseaux électriques intelligents et les compteurs intelligents en donnant les définitions, caractéristiques, principes de bases et architecture des compteurs, notamment l'évolution des smart grids en Algérie.

Le deuxième chapitre sera consacré à une étude approfondie du dispositif expérimental réalisé autour d'un Arduino ainsi que la description de chaque bloc du circuit.

Enfin, nous exposons en troisième chapitre, un aperçu sur le logiciel Android studio et la réalisation d'une application capable de gérer la communication entre la carte électronique et le Smartphone ou Tablette ainsi que stockage des données sur Smartphone.

Chapitre I

Etat de l'art des compteurs électriques

Introduction

Les réseaux électriques sont omniprésents dans notre vie quotidienne. Ils offrent de l'énergie en continu provenant de sources de production diverses tel que les barrages hydroélectriques, les centrales thermiques, etc. Ils doivent fournir une électricité de qualité en maintenant l'amplitude et la fréquence de l'onde de tension à des valeurs acceptables par rapport aux normes imposées par le pays. [1]

La tension électrique sortie des générateurs est transformée grâce aux transformateurs élévateur et transportée sur de grandes distances par des lignes de haute tension afin d'alimenter les différentes régions puis grâce aux transformateurs abaisseur HT/MT puis MT/BT alimenté tous les consommateurs basse tension. [2]

Cependant, nous remarquons après que la production de l'énergie électrique peut se faire de la BT vers MT ou de la MT vers HT grâce aux énergies renouvelables. Ce qui a rendu le contrôle du réseau électrique très compliqué. Cette complexité a introduit le concept de réseaux électriques intelligent et a permis d'introduire les technologies de l'information et des communications, dont les compteurs intelligents constituent la première brique des réseaux électriques intelligents. Ces compteurs intelligents établissent une communication entre les réseaux et les consommateurs afin de mieux maîtriser l'offre/demande. De plus, ces compteurs dits "Avancés" sont capables à la fois de transmettre et de recevoir des informations. Ainsi, l'objectif de ce chapitre est de décrire en détail les compteurs intelligents [3].

I.1. Compteur intelligent

Ces compteurs qu'on appelle « intelligent », sont des compteurs ayant la capacité de communiquer afin de transmettre en temps réel les données relatives à la consommation électrique d'un bâtiment, d'une entreprise ou d'un foyer mais aussi, pour certains, de pouvoir les administrer sans qu'une intervention technique sur le lieu d'implantation du compteur ne soit nécessaire [3]. Ces compteurs intelligents est en outre communicant et transmet par différents canaux (courant porteur, Internet, téléphone) les informations recueillies. La signification « intelligent » est clairement abusif et a probablement été choisi pour indiquer qu'il s'agit d'un équipement électronique capable d'être programmé et commandé.

Il n'est pas juste de l'interpréter au sens informatique « d'intelligence artificielle », vu que ces compteurs agissent comme des « serveurs » obéissant aux requêtes extérieures et ne sont pas munis de capacité de décision et d'action propre. [4]

I.2. Architecture d'un compteur intelligent

Un compteur intelligent est en général constitué des éléments suivants :

- Une unité pour détecter le courant électrique grâce au capteur de courant ».
- Une unité de communication.
- Une unité de contrôle.
- Une horloge temps réel qui sert à synchroniser le compteur avec l'heure mondiale pour lui permettre de surveiller la consommation durant la journée.
- Une unité d'alimentation qui permet de fournir une tension constante pour le fonctionnement des circuits intégrés constituant le compteur intelligent.

La figure suivante montre un schéma simplifié de l'architecture d'un compteur intelligent :

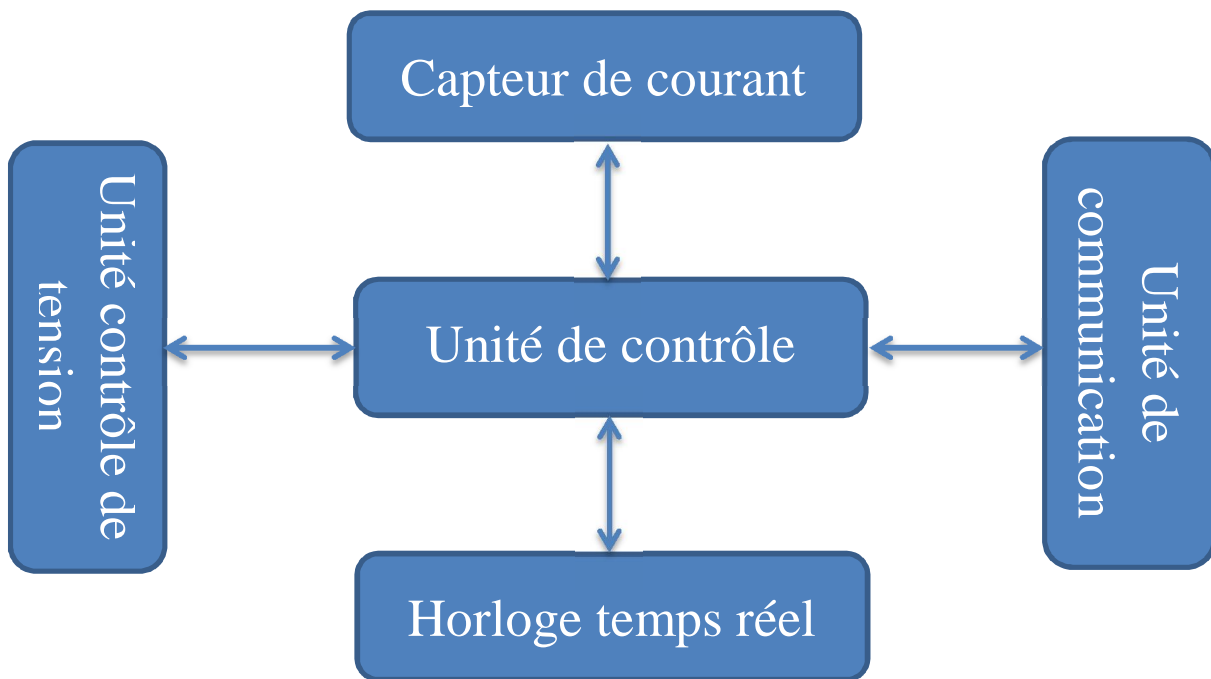


Figure I.1: Architecture d'un compteur intelligent.

I.3. Présentation d'un modèle de système de comptage évolué

Un système de comptage évolué peut être réalisé par [5] :

- Les informations résultant des mesures d'énergie peuvent être stocké par l'intégration des compteurs avancés.
- La constitution de systèmes de transmission de données permettant le roulement rapide et fiable des informations dans les compteurs
- Le compteur permet de faire le retenu à distance ainsi que l'orientation de la consommation d'énergie, et assurer la communication concernant la transmission et réception des données.

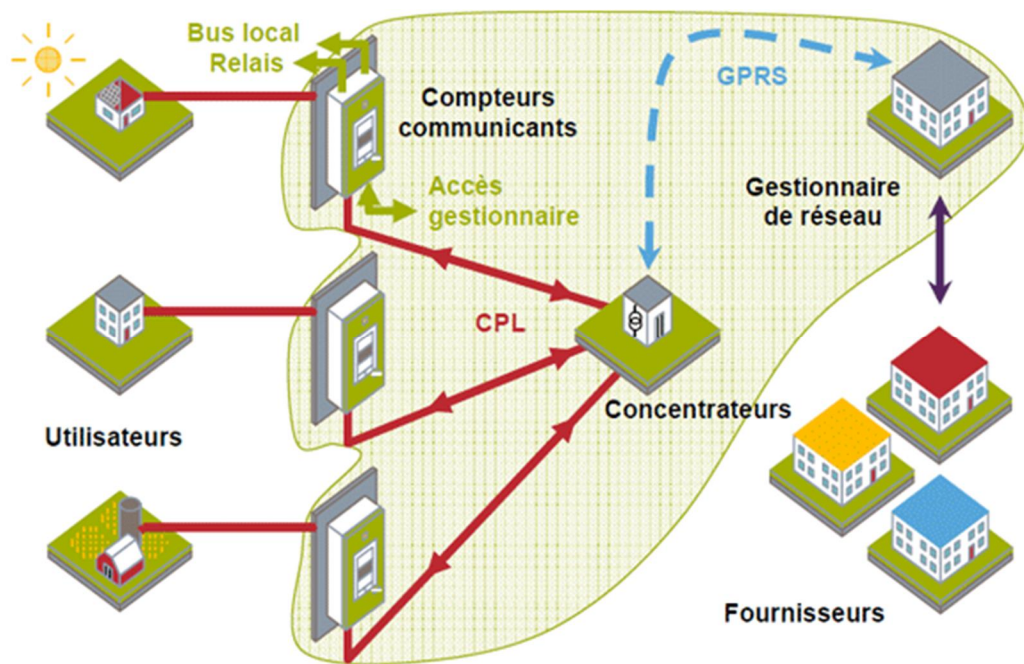


Figure I.2 : Schéma d'un modèle couramment retenu pour un système de comptage évolué en électricité [5].

Selon le schéma ci-dessus, la communication s'effectue entre un ensemble de compteurs installés chez les utilisateurs et un concentrateur localisé à proximité dans le poste de distribution publique.

Ces compteurs rassemblent les données pour les transmettre au gestionnaire du réseau électrique via la technologie du Courant Porteur en Ligne (CPL).

Un modem CPL est allié à chaque compteur et concentrateur, ce modem permet de coder et décoder les données en un signal électrique et les superpose au courant électrique à 50 Hertz.

Ensuite, au niveau des concentrateurs, les données sont codées sous format numérique, puis transmises au système informatique du gestionnaire du réseau électrique (par l'intermédiaire du réseau de téléphonie GPRS ou GSM) [5].

Le système informatique du gestionnaire du réseau est accessible par les fournisseurs d'énergie qui reçoivent régulièrement les données de comptage de leurs clients pour la facturation de l'énergie.

La figure suivante représente les relations entre le consommateur, le fournisseur et le gestionnaire de réseaux.

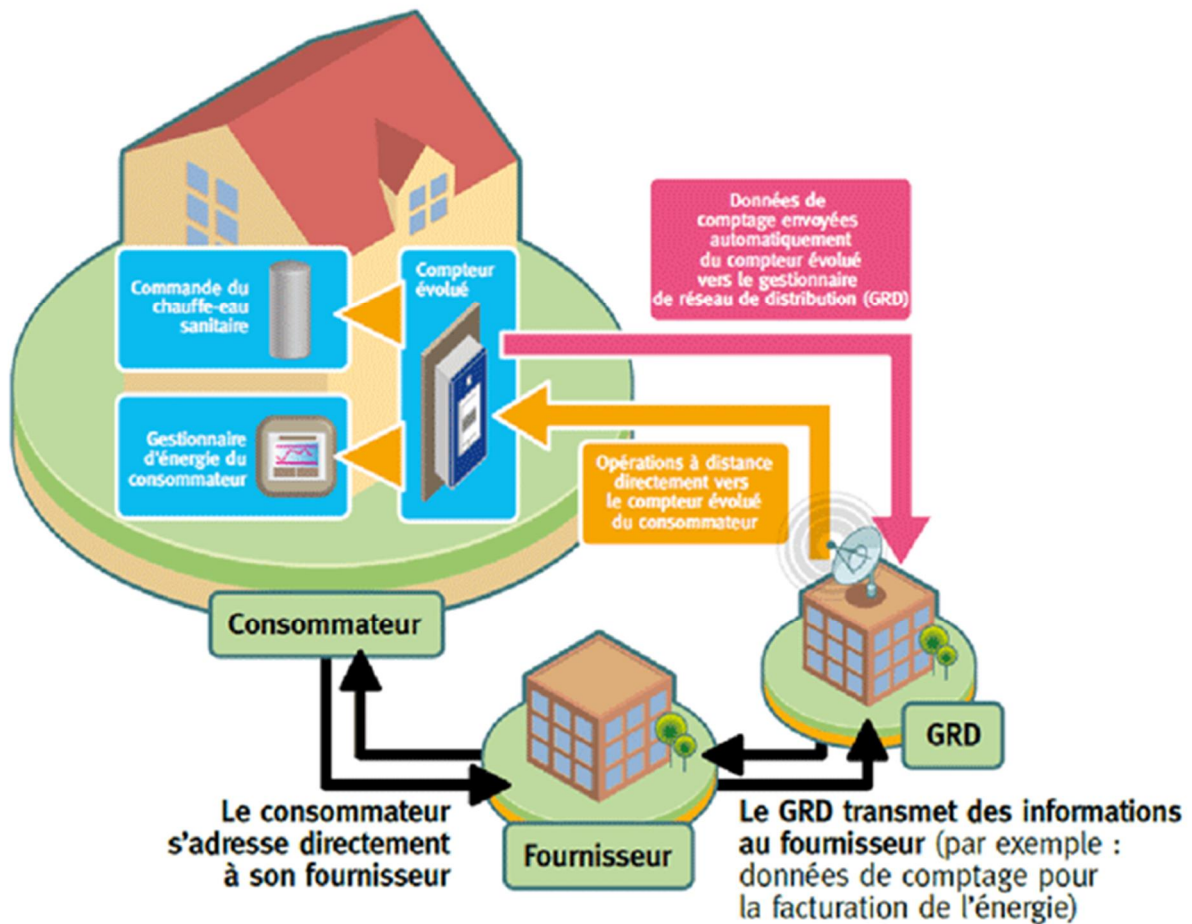


Figure I.3 : Architecture de réseau basé sur des compteurs intelligents (des compteurs électriques évolués) [5].

I.4. Principales fonctionnalités des systèmes de comptage évolués

Dans le but de répondre aux exigences demandées, les compteurs avancés doivent être capables d'assurer :

- Le retenu des données du compteur à intervalle régulier.

- La télé-relève des données du compteur à partir d'un système d'informations situé à distance.
- La mesure de la consommation.
- La transmission à distance des messages des acteurs du marché pour le client (consommateur/producteur).
- L'affichage des informations sur le compteur.
- Un port de communication principal permettant le transfert d'informations via le GPRS, le GSM ou le CPL.
- La mesure de la qualité de l'énergie électrique en ce qui concerne la continuité et la qualité de tension).

I.5. Réseau électrique intelligent

I.5.1. Définition [6]

Les réseaux électriques intelligents ou smart grids, sont des réseaux d'électricité qui permet l'ajustement des flux d'électricité entre les fournisseurs et les consommateurs grâce à des technologies informatiques. En outre, l'acquisition des informations sur l'état du réseau électrique, les réseaux électriques intelligents apportent une correspondance entre la production, distribution et consommation.

Il est nécessaire de savoir que les réseaux électriques intelligents ont une appellation générale pour l'ensemble des technologies et infrastructures « intelligentes » installées. Le compteur communicant est l'un des bases dans la mise en place des réseaux électriques intelligents. Les réseaux intelligents peuvent être définis selon quatre caractéristiques en matière de [5]:

- **Flexibilité** : ils assurent l'équilibre entre la production et la consommation.
- **Fiabilité** : ils améliorent l'efficacité et la sécurité des réseaux électriques.
- **Accessibilité** : ils encouragent l'intégration des sources basées sur les énergies renouvelables.
- **Economie** : par une meilleure gestion du système, des économies d'énergie et une diminution des coûts (à la production comme à la consommation).

I.5.2. Fonctionnement [6]

Un réseau électrique intelligent réunit l'infrastructure électrique aux technologies numériques qui analysent et transmettent l'information reçue. Ces technologies sont utilisées à tous les niveaux du réseau électrique : production, transport, distribution et consommation. Il permet aussi de faire :

- **Un contrôle des flux en temps réel :** des capteurs installés sur l'ensemble du réseau électrique indiquent instantanément les flux électriques et les niveaux de consommation. Les opérateurs du réseau peuvent alors réorienter les flux énergétiques en fonction de la demande et envoyer des signaux de prix aux particuliers pour adapter leur consommation volontairement ou automatiquement.

- **L'interopérabilité des réseaux :** l'ensemble du réseau électrique comprend le réseau de transport et le réseau de distribution. Pour le réseau de transport réuni entre les sites de production d'électricité aux zones de consommation. En outre, le réseau de distribution s'apparente aux axes secondaires. Il transfère l'électricité jusqu'aux consommateurs. Par l'échange instantané d'informations, les réseaux électriques intelligents assurent la coordination entre les gestionnaires du réseau de transport et ceux du réseau de distribution.

- **L'intégration des énergies renouvelables au réseau :** les réseaux électriques intelligents sont basés sur un système d'information qui permet de concevoir à court terme le niveau de production et de consommation.

- **Une gestion plus responsable des consommations individuelle :** le compteur intelligent est l'un des bases du réseau intelligent, ils fournissent des informations sur les prix, les heures de pointe de consommation, la qualité et le niveau de consommation d'électricité du foyer. Les consommateurs peuvent alors savoir leur consommation au cours de la journée. et les opérateurs du réseau peuvent détecter plus vite les pannes.

I.5.3. Types de réseau électrique intelligent [6]

Concernant les Types de réseaux électriques intelligents, on peut les trier en trois principales répartitions suivantes :

- **AU NIVEAU DES GESTIONNAIRES DE RESEAUX DE TRANSPORT :** Il permet l'amélioration du télé-contrôle, la surveillance et la planification et à travers l'utilisation des progrès technologiques il admet de poursuivre une évolution entamée de longue date pour faire face aux nouveaux besoins de transport notamment suite à l'apport en réseaux de la production décentralisée de puissance réduite et moins gérable par le responsable d'équilibre et de sécurité réseau.

▪ **AU NIVEAU DES GESTIONNAIRES DE RESEAUX DE DISTRIBUTION :**

Les technologies utilisées dans le réseau de transport tel que le télé-contrôle, protections et la gestion d'équilibre sont mise en place par les gestionnaires de réseaux de distribution exagéré par le développement de la production décentralisée. évolution est facilitée par l'abaissement du cout de ces technologies a facilité cette évolution.

▪ **AU NIVEAU LOCAL :** Le Changement le plus important pourra voir le jour grâce à l'apport de l'électronique, de l'informatique et des télécommunications qui ouvre de nouveaux horizons à la gestion de la consommation et de la production locale. Ce domaine spécifique s'appelle Maison intelligente.

I.5.4. Les limites dans la mise en œuvre

En revanche, le coût des investissements reste élevé, ce qui fait les réseaux électriques intelligents doivent être implantés sur l'ensemble du réseau et impliquer tous les acteurs pour être efficaces. L'autre obstacle est la diversité des acteurs, car ils doivent mettre au point des systèmes communicants variés avec des logiques convergentes.

De plus, les données recueillies sont complexes à gérer et à stocker, compte tenu de l'importante quantité d'informations à traiter. Enfin, les informations sur les horaires ou les activités des consommateurs et des producteurs doivent être confidentielles. Ce qui implique d'imposer des normes sur la protection des données [7].

I.6. Réseaux électriques intelligents en Algérie

L'Algérie a un grand potentiel quant au déploiement du réseau électrique intelligent, car elle recense plusieurs énergies renouvelables qui pourraient être utilisées dans ce sens et diverses multinationales y voient un terrain opportun pour les réseaux électriques intelligents.

Les entreprises locales s'accroissent également sur le sujet et vise en particulier la mise au point par exemple des compteurs intelligents [8].

La tenue d'un brainstorming les 6 et 7 juillet 2011 au CBA, avait pour objectif de débattre des préoccupations du Groupe Sonelgaz en matière de réseaux électriques intelligent [8]. Organisé par le CRDEDEG (Centre de Recherche et de Développement de l'Electricité et du Gaz), il a permis de dégager des recommandations qui vont prendre en charge les préoccupations soulevées. Dans ce cadre cinq communications ont été exposées, à savoir [9] :

1. Smart Grids Opportunités
2. Algérie Smart Grid ó Benchmarking
3. Etat du parc national des compteurs d'électricité
4. Télégestion des compteurs d'énergie électrique dans un contexte de gestion intelligente des réseaux
5. Et Plateforme de tests des systèmes photovoltaïques avec injection dans le réseau, à l'université Hadj Lakhdar de Batna.

Il en ressort que la mise en place de compteurs intelligents est le premier pas de cette révolution technologique vers un réseau électriques intelligent. L'interconnexion de réseaux intelligents permet de mettre en relation l'offre et la demande en matière de consommation d'électricité. En outre, l'apport des technologies informatique devrait économiser l'énergie, sécurisé le réseau et en réduire les coûts.

On retiendra la préparation d'une mutation du système algérien énergétique actuel vers un nouveau système énergétique par l'intégration de la dimension réseaux électriques intelligents dans les plans stratégiques actuels des sociétés du Groupe. Les besoins en informations (technique et commerciales) du Système d'Information du réseau électrique intelligent doivent être définis et le système d'information à élaborer doit s'adapter avec les systèmes d'informations existants ou en cours de réalisation. Par ailleurs, il y a lieu de définir les fonctionnalités à implémenter et s'assurer que les textes législatifs intègrent les spécificités du réseau électrique intelligent.

Il y a également lieu de souligner que la formation jouera un rôle déterminant dans la concrétisation de ce projet. Les universités algériennes, les centres de recherche et les associations spécialisées sont à mettre à contribution. [9]

I.7. L'Algérie et les compteurs intelligents

La Société de distribution de l'électricité et du gaz d'Alger (SDA) compte lancer prochainement plusieurs projets visant l'optimisation de la conduite et l'exploitation de son

réseau, notamment le lancement d'un système de télégestion des compteurs, a indiqué aujourd'hui un responsable de cette filiale du groupe Sonelgaz. "La SDA compte lancer dès 2012 un ensemble de projets d'envergure pour accélérer l'automatisation de la gestion de ses réseaux de distribution d'électricité [9].

Il s'agit des projets d'extension et d'adaptation du système de télé-conduite, de l'introduction d'un système de télégestion des compteurs, ainsi que la réalisation d'un call center pour la gestion de la relation clientèle", explique le directeur technique "électricité" de la SDA, Merouane Chabane [10]. Le système de télégestion assurera, dans un premier temps, la gestion à distance des compteurs d'électricité de l'ensemble des clients industriels (raccordés aux réseaux HTB et HTA) et encourage ainsi une nouvelle approche dans les rapports avec les clients, a-t-il précisé dans une déclaration à l'APS [10].

Le même système est basé sur une infrastructure "intelligente" dont le compteur électronique est le terminal chez le client. "La commande distante des ouvrages et la gestion à distance de la grande majorité des transactions commerciales et des principales activités sur les compteurs, en évitant par conséquent l'intervention sur place, est un élément positif dans la réduction des émissions de CO_2 », a-t-il ajouté. Ce système réalise également la gestion à distance des compteurs d'électricité installés dans presque 85% des postes HTA/BT de distribution publique [10]

Cette option permet, outre le rapatriement distant des mesures aux postes, éliminant ainsi les campagnes coûteuses de sondage réglementaire, de remonter les informations relatives à la détection de défauts et autres alarmes. Le système est conçu pour s'interfacer facilement avec le reste du système d'information (gestion clientèle, facturation). Il permettra également la gestion des alarmes, notamment à travers l'envoi des SMS par le compteur vers la plateforme centrale. Dans la deuxième phase du projet prévue à partir de 2017, ce système sera généralisé aux clients raccordés aux réseaux Basse Tension (BT), soit plus de 700.000 clients. Pour ce faire, les distributeurs, en collaboration avec le Centre de Recherche & Développement de l'électricité et du gaz du groupe Sonelgaz, ont engagé en 2010 un projet novateur de conception et de développement d'un compteur " intelligent " BT avec l'université de Boumerdes [9].

Le projet rentrera dans sa phase d'industrialisation dès 2014. Des prototypes en laboratoire sont d'ores et déjà réalisés. D'un autre côté, la SDA vise la mise à niveau du système de télé-conduite, l'adaptation de son architecture à l'évolution technologique et aux nouveaux enjeux relatifs aux réseaux intelligents, et le renforcement des réseaux de télécommunication. [10]

Ses retombées sont énormes pour les distributeurs, en gains sur la relève et sur les pertes d'énergie, mais également pour le client en l'aidant à rationaliser sa consommation d'énergie, et par conséquent sur l'environnement, qui voit les quantités de gaz à effet de serre se réduire proportionnellement aux quantités d'énergie économisées grâce aux compteurs intelligents.

Le projet compteur intelligent compte six phases de réalisation, en l'occurrence [9]:

- L'analyse de l'état de l'art mondial sur les systèmes de comptage de l'énergie électrique.
- L'analyse des méthodes et des moyens permettant l'élaboration de nouveaux types de compteurs d'énergie.
- Les essais et réalisations des différents modules du compteur.
- La réalisation et essais de prototypes de compteurs dans les conditions industrielles.
- L'élaboration et le dépôt de brevets.

Cet important projet de recherche est une première en Algérie quant à la collaboration entre le monde académique et les entreprises économiques. Néanmoins le Groupe Sonelgaz avait développé avec différents universités (USTHB, UMBB, USTO, etc.) des actions de partenariat et d'échange sur divers sujets de recherche. [9]

Conclusion

Dans ce chapitre, nous avons pu constater que le réseau électrique intelligent facilite des changements importants au processus de production, de transport et de consommation de l'électricité.

Des éléments de base de nature technologique permettent de nouvelles fonctions du réseau électriques qui, elles-mêmes, soutiennent mieux les entreprises d'électricité, qui inclut graduellement certains avantages d'ordre social tels la performance environnementale et la

maîtrise par le client de l'électricité. Ainsi, le rôle important des compteurs intelligents a été mis en évidence afin de présenter les fonctions de distribution et de contrôle qu'ils permettent de réaliser.

Pour terminer, nous avons proposé une architecture globale du réseau électrique intelligent qui se base sur des compteurs intelligents.

Chapitre II

Réalisation d'un compteur intelligent

Introduction

Dans ce chapitre, on présentera en détail l'ensemble du dispositif expérimental de la réalisation d'un compteur communicant commandé par Arduino et contrôlé par un Smartphone.

Ce dispositif est conçu à base d'une carte Arduino MEGA 2560, celle-ci permet de commander un compteur électrique communicant (Smart meter) via un système Bluetooth, constitué essentiellement d'une alimentation (secteur) un circuit de mesure (capteur de courant, capteur de tension), microcontrôleur (carte Arduino), afficheur LCD, ainsi qu'un module de communication sans fil (Bluetooth).

Ainsi dans cette partie nous allons concentrer sur la partie réalisation d'un compteur communicant, ayant pour but la mesure de la tension et le courant pour enfin calculer la puissance active et l'énergie du consommateur, puis transmettre ces informations vers un Smartphone via un module Bluetooth.

Notre réalisation pratique a été faite en trois parties :

- La première partie est la conception du système électronique.
- La deuxième partie est la programmation de la carte.
- La troisième partie est consacrée à la création d'une application Android.

(voir chapitre III)

La première partie de notre projet est très importante, on est passé par plusieurs étapes :

a) Chercher les différentes structures des blocs constituant notre maquette et qui vont avec les objectifs fixés et les moyens disponibles.

b) Présenter les différents éléments ou composants constituant chacun des blocs en choisissant des composants aux caractéristiques voulus, et disponibles sur le marché.

c) On assemble ensuite les composants suivants notre montage sur le circuit imprimé, commençant par l'alimentation générale de notre dispositif.

I.1. Schéma du prototype

Notre compteur intelligent est constitué de plusieurs parties importantes (Figure II.1) qui sont :

- Circuit d'alimentation.
- Circuit de mesure de courant et de la tension.
- Circuit de commande et de calcul (carte Arduino (MCU)).
- Circuit de Transmission basé sur un module Bluetooth.
- Unité d'affichage (afficheur LCD).

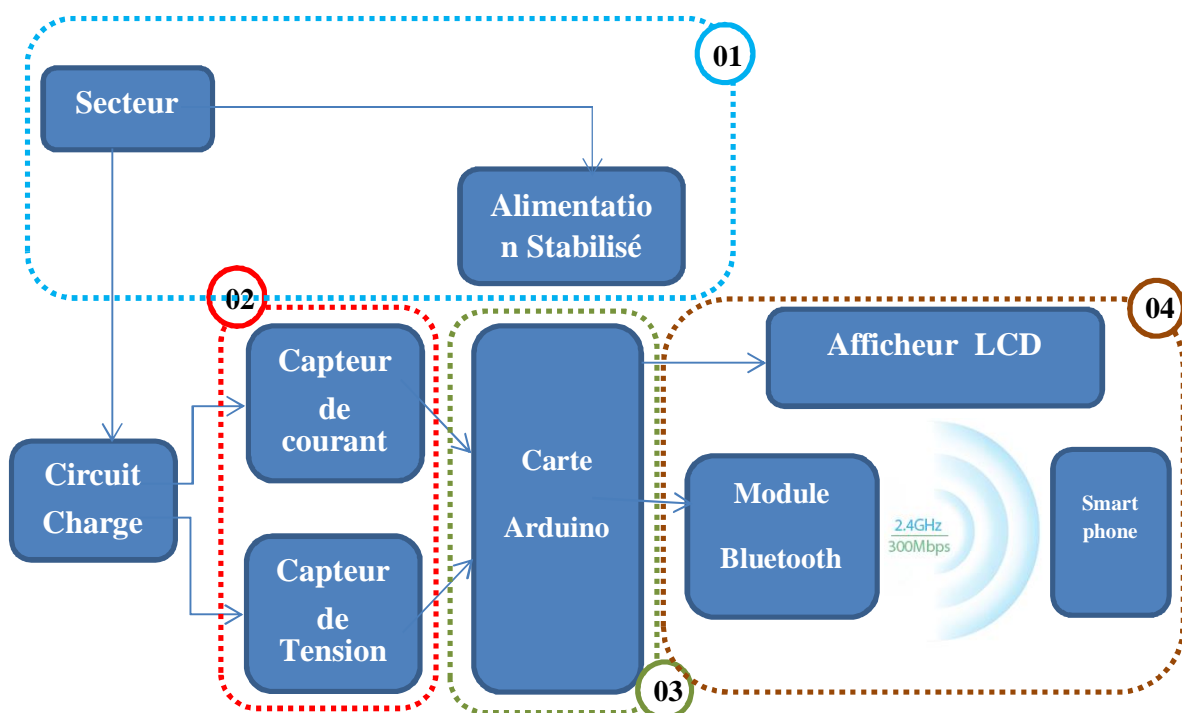


Figure II.1 : Schéma du prototype du circuit du compteur intelligent

Ces différents blocs seront détaillés par la suite :

I.1.1. Alimentation

Presque tous les montages électroniques demandent un apport d'énergie par une source de tension continue, quand la consommation du circuit est faible on peut se contenter d'utiliser une pile ou un accumulateur mais dès que le débit est un peu plus élevé le coût important de l'énergie électrique stockée conduit à préférer une alimentation à partir du secteur.

Le secteur est une tension Alternative qu'il faut convertir en tension continue et l'amplitude de ce secteur est très élevée (230v) par rapport au niveau dont on a besoin dans notre carte électronique, donc on a besoin d'abaisser cette tension.

Dans notre cas nous avons réalisé une alimentation continue des tensions de +12V et -12V pour faire fonctionner d'une part les capteurs de tension et de courant et d'autre part notre calculateur (carte Arduino), afin de garantir le suivi continue et l'enregistrement de la consommation d'énergie électrique lors d'une coupure d'électricité une pile rechargeable est envisageable.

I.1.1.1. Blocs constituant une alimentation stabilisée

La réalisation d'une alimentation stabilisée passe par plusieurs étapes :

- L'adaptation (abaissement) et l'isolation
- Le redressement
- Le filtrage
- La stabilisation

Ces étapes sont reliées entre eux comme l'indique la figure ci-dessous.

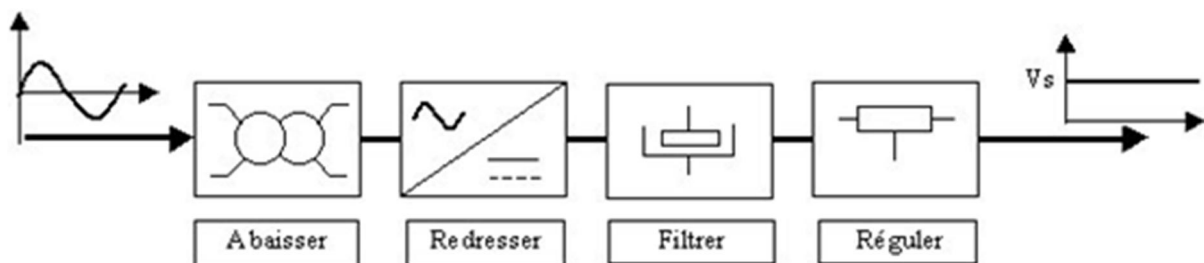


Figure II.2 : Différents étages d'alimentation stabilisée

I.1.1.2. Schéma électronique de l'alimentation su circuit

Le circuit électronique de notre alimentation stabilisée est représenté par la figure ci-dessous (Circuit simulé sous le logiciel ISIS) :

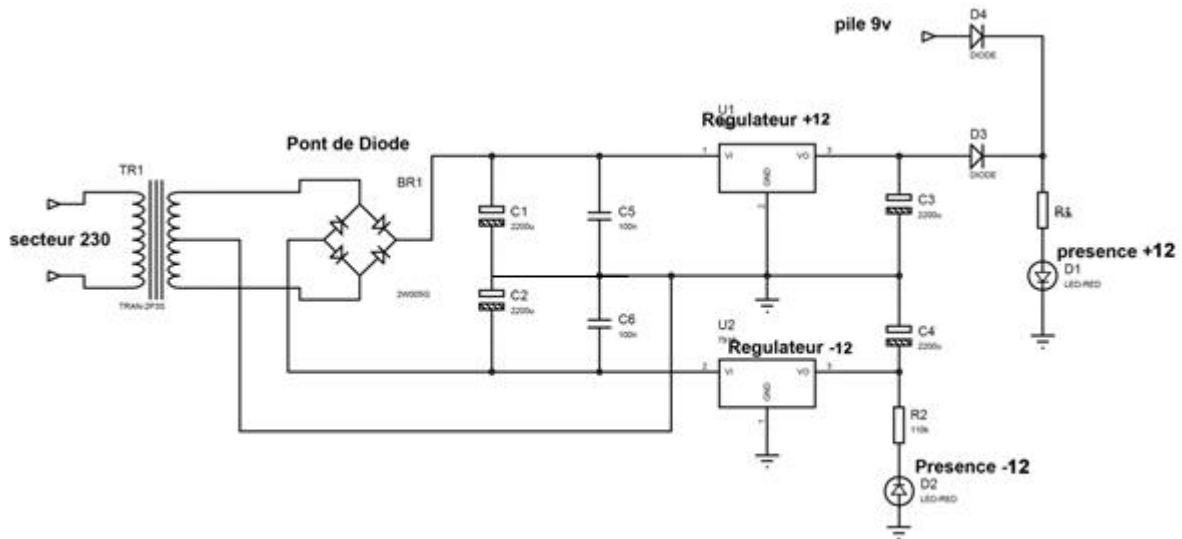


Figure II.3 : Circuit d'alimentation.

A. L'adaptation et l'isolation

Cette fonction est réalisée par un transformateur. Il permet de diminuer l'amplitude de la tension secteur, et a pour but :

- Abaisser la tension
- Adapter la tension fournie par le secteur
- Isoler le montage du secteur

Un transformateur est caractérisé par son rapport de transformation. Ce rapport est fonction du nombre de spires des enroulements primaire et secondaire :

$$k = \frac{U_1}{U_2} = \frac{N_1}{N_2} = \frac{I_2}{I_1}$$

U1 : tension primaire

N1 : nombre de spires primaires

U2 : tension à vide secondaire

N2 : nombre de spires secondaires

Dans notre circuit d'alimentation nous avons utilisé un transformateur à point milieu 220V/ 12V.

B. Redressement

Cette fonction est réalisée par un pont de diode, constitué de quatre diodes D1, D2, D3 et D4 (voir figure II.4). L'ensemble de ces quatre diodes forment un pont de Graetz, donc la tension de sortie du transformateur est redressée par le pont afin d'avoir un signal redressé double alternance.

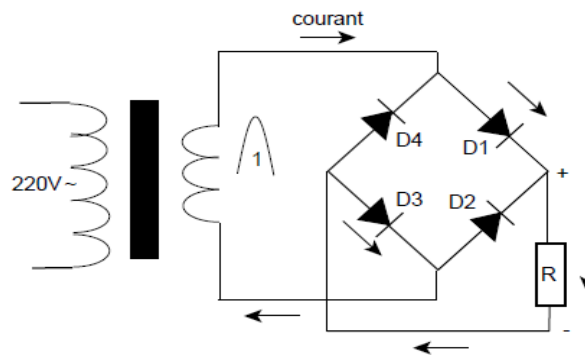


Figure II.4 : Circuit du redressement double alternance.

Pendant l'alternance positive du signal, les deux diodes D1 et D3 conduisent puisqu'elles sont polarisées en direct ; alors que D2 et D4 sont bloquées. Pendant l'alternance négative, les deux diodes D2 et D4 conduisent, et D1 et D3 se bloquent. On obtient ainsi un redressement double alternance.

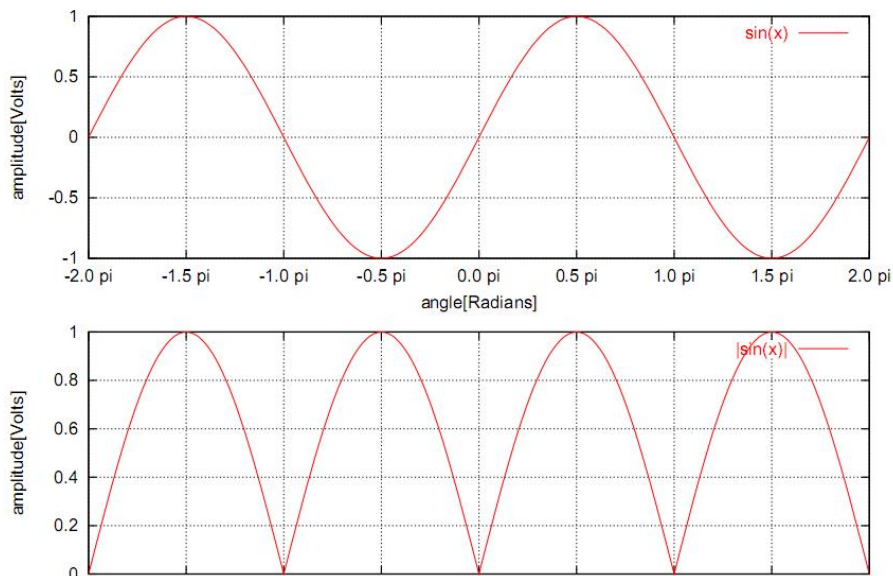


Figure II.5 : Courbes du redressement double alternance.

C. Filtrage

Après redressement, la tension de sortie aux bornes du pont redresseur est loin d'être continue. Le filtrage a pour but de transformer cette tension redressée en une tension continue légèrement ondulée. En conséquence, cette partie va permettre de lisser le signal

On devrait obtenir une tension continue constante pour transformer les signaux redressés double alternance en une tension continue, on doit donc filtrer ou fait disparaître les ondulations de faibles valeurs.

D. Stabilisation

La partie du montage comprenant le transformateur, les diodes et le condensateur de filtrage forme déjà une alimentation continue non régulée, lorsqu'on désire avoir une tension suffisamment stable il faut compléter le montage par un régulateur, ce dernier est un dispositif capable de fournir une tension de sortie stable malgré les variations de la tension d'entrée ou les variations de la charge, ces caractéristiques sont les suivantes :

➤ Fonctionnement du régulateur

La tension d'entrée du régulateur doit être suffisamment grande afin de maintenir V_s constante. Les constructeurs donnent une tension d'entrée minimum à respecter afin d'assurer le fonctionnement correct du régulateur.

Nous avons utilisé le régulateur LM7812 et LM7912 pour obtenir des tensions régulées de +12 et -12 respectivement (pour plus de détails voir le Datasheet du circuit électronique).

I.1.2. Circuit de mesure

I.1.2.1. Définition

Un circuit de mesure comprend un ensemble d'éléments importants, tel que montre en Figure II.6 La grandeur physique à mesurer (appelée mesurande) est une valeur analogique qui n'est généralement pas exploitable directement. [11]

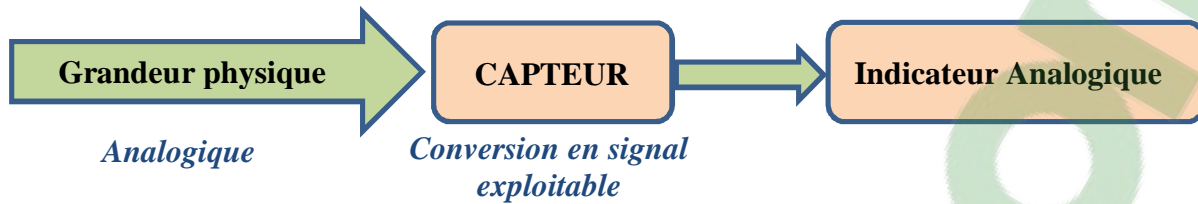


Figure II.6 : Schéma bloc d'un système de mesure analogique

Cette grandeur physique peut-être une force, une température, un débit, ou toute autre grandeur doit être mesurée. Elle doit être convertie en une autre valeur analogique par l'élément de mesure (appelé capteur). Ce signal analogique à la sortie (appelé aussi réponse) du capteur est un signal directement exploitable pour les indicateurs analogiques (affichage à aiguille).

Ce signal doit toutefois être converti en un signal numérique si on désire utiliser un affichage numérique (Figure II.7). La conversion se fait par l'intermédiaire d'un circuit convertisseur analogique-numérique.

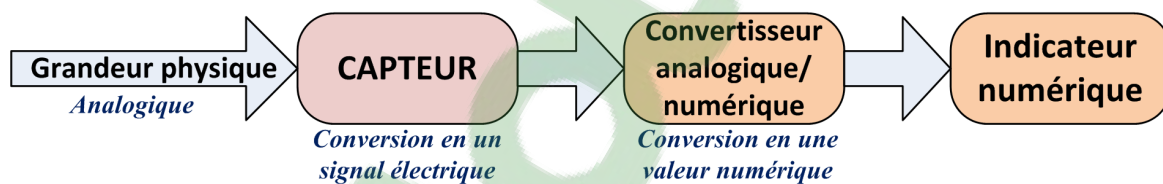


Figure II.7 : Schéma bloc d'un système de mesure numérique.

I.1.2.2. Élément de mesure

Un élément de mesure, désigné généralement sous le nom de capteur.

Un capteur est un organe de prélèvement d'informations qui élabore à partir d'une grandeur physique (Information entrante) une autre grandeur physique de nature différente (Information sortante : très souvent électrique). Cette grandeur, représentative de la grandeur prélevée, est utilisable à des fins de mesure ou de commande. [11]

Le capteur est le premier élément d'une chaîne de mesure ou d'instrumentation. Capteur n'est jamais parfait, il convient de connaître avec la plus grande précision possible son état d'imperfection. [12]

De plus, il faut prendre en compte la perturbation apportée au système par la mesure. Le concepteur d'une chaîne instrumentale aura donc des choix à opérer. [12]

A. Type des capteurs [13]

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

- ❖ Capteurs passifs : il s'agit généralement d'impédance dont l'un des paramètres déterminants est sensible à la grandeur mesurée. La variation d'impédance résulte Soit d'une variation de dimension du capteur, Soit d'une déformation résultant de force ou de grandeur.

- ❖ Capteur actifs : fonctionnant en générateur, un capteur actif est généralement fondé dans son principe sur effet physique qui assure la conversion en énergie électrique de la forme d'énergie à la grandeur physique à prélever, énergie thermique, mécanique ou de rayonnement. L'un des effets physiques les plus rencontrés en instrumentation, on trouve l'effet Hall.

➤ L'Effet Hall [12]

a) Définition et Fonctionnement

Un barreau de semi-conducteur soumis à un champ magnétique uniforme B et traversé par un courant I , est le siège d'une force électromotrice U_H sur deux de ses faces.

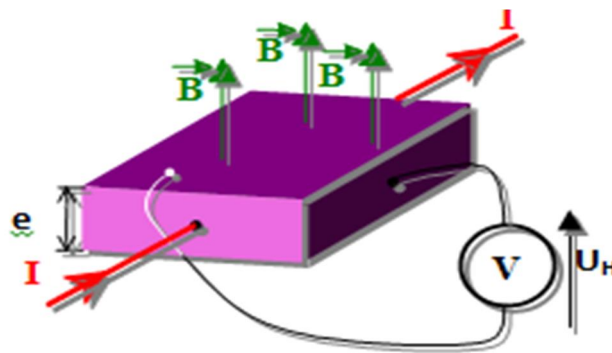


Figure II.8 : schéma significatif du l'effet Hall

La tension de Hall U_H est définie par la relation ci-contre : $U_H = K_H \cdot I \cdot B \cdot \frac{1}{e}$

K_H : Constante de Hall (dépend du semi-conducteur)

I : Intensité de la source de courant (A)

B : Intensité du champ magnétique (T)

e : Épaisseur du barreau de silicium.

Si on maintient le courant I constant, on a donc une tension U_H proportionnelle au champ magnétique B : $U_H = k \cdot B$ avec k constante égale à $k = \frac{U_H}{B}$.

b) Principe physique de l'effet Hall [13]

Il existe deux grandes familles de capteurs à effet hall :

❖ Capteur à boucle fermée

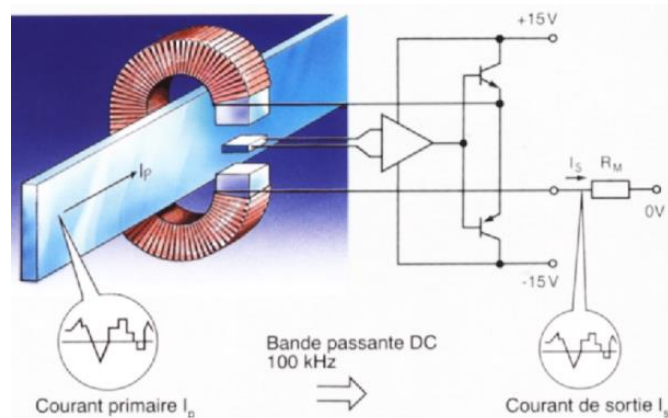


Figure II.9 : principe d'un capteur à boucle fermée.

Les capteurs à boucle fermée dits aussi à flux nul, possèdent un circuit de compensation intégré qui améliore notablement les performances.

Les capteurs à boucle fermée délivrent un courant secondaire I_s proportionnel à la tension de Hall V_H , lequel agit en signal de contre réaction afin de compenser les I_p primaire par des I_s secondaires opposés.

Le courant I_s est réduit à quelques dizaine de mA. On choisit $I_p \cdot N_p + I_s \cdot N_s = 0$

- Capteur de courant à boucle fermée

- Bien adaptés aux applications industrielles exigeantes en performances, précision et Bande Passante. Les capteurs à boucle fermée se distinguent par :

- Une excellente précision : $\pm 0.5\%$
- Une très bonne linéarité : $\pm 0.1\%$
- Un temps de réponse très court : $< 1 \mu s$
- Une bande passante élevée : 0 à 200 kHz

❖ Capteur à boucle ouverte

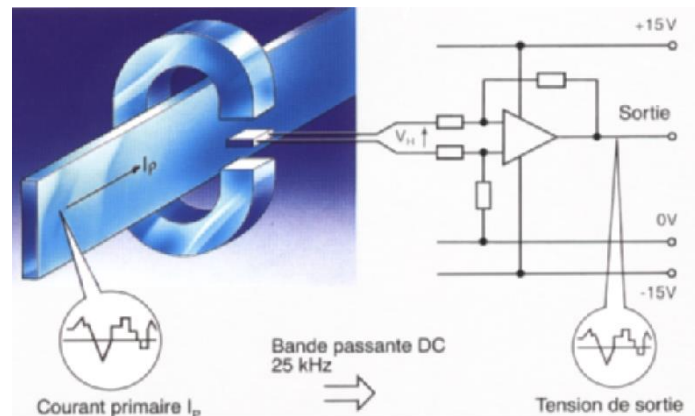


Figure II.10 : principe d'un capteur à boucle ouverte.

Les capteurs à boucle ouverte exploitent l'effet Hall.

La tension du générateur de Hall est générée par le courant de $c^{22} I_p$ et par l'induction dans l'entrefer B_{22} .

Le courant de cde I_p est fourni par une source à courant constant. Le courant I_p génère l'induction B_{22} . Dans la zone linéaire du cycle d'hystérésis, B_{22} est proportionnel à I_p tel que : $B_{22} = k \cdot I_p$

Un entrefer « d » important linéarise le cycle d'hystérésis et réduit notablement la tension d'offset liée à l'induction rémanente B_r du circuit magnétique. Mais l'entrefer augmente les risques de fuite magnétiques, pouvant perturber la tension de Hall : V_H .

Les capteurs de courants à boucle ouverte du fait de leur bande passante limitée à 25kHz sont assez bien adaptés aux applications industrielles ne nécessitant pas de performances précision et fréquence de fonctionnement élevées.

Cette gamme vise plutôt des critères économiques.

Plage de mesure : 0 à 18000A

Bande passante : 0 à 25kHz

Précision typique : $\pm 1\%$

Linéarité : $\pm 0.5\%$

Temps de réponse : <3 à 7 s

Les deux capteurs utilisés pour mesurer les deux paramètres (courant et tension) sont choisis par rapport à leurs performances et fiabilité (précision, rapidité, bande passante, linéarité, ainsi que la disponibilité sur le marché). En se basant sur ces paramètres cités auparavant nous avons sélectionné deux capteurs:

- HV comme étant un capteur de tension en boucle ouverte (voir l'annexe pour plus de détails).
- TA pour mesurer le courant (voir l'annexe pour plus de détails).

Les deux capteurs génèrent en sortie des niveaux de tension entre (0-5V) qui représentent l'image des paramètres du réseau et qui seront acquis par notre calculateur (Arduino).

Dans ce qui suit nous allons détailler le circuit réalisé permettant la mesure des deux paramètres du réseau électrique afin de pouvoir calculer l'énergie consommée.

Nous avons conçu une maquette qui simule la consommation de l'énergie électrique d'une maison, cette maquette est constituée de trois charges de différente puissance (lampes), montées en parallèle. En faisant contrôler chacune séparément afin de faire varier la consommation globale (Figure II.11). Les différents types des circuits réalisés sont joints en annexes (voir annexe)

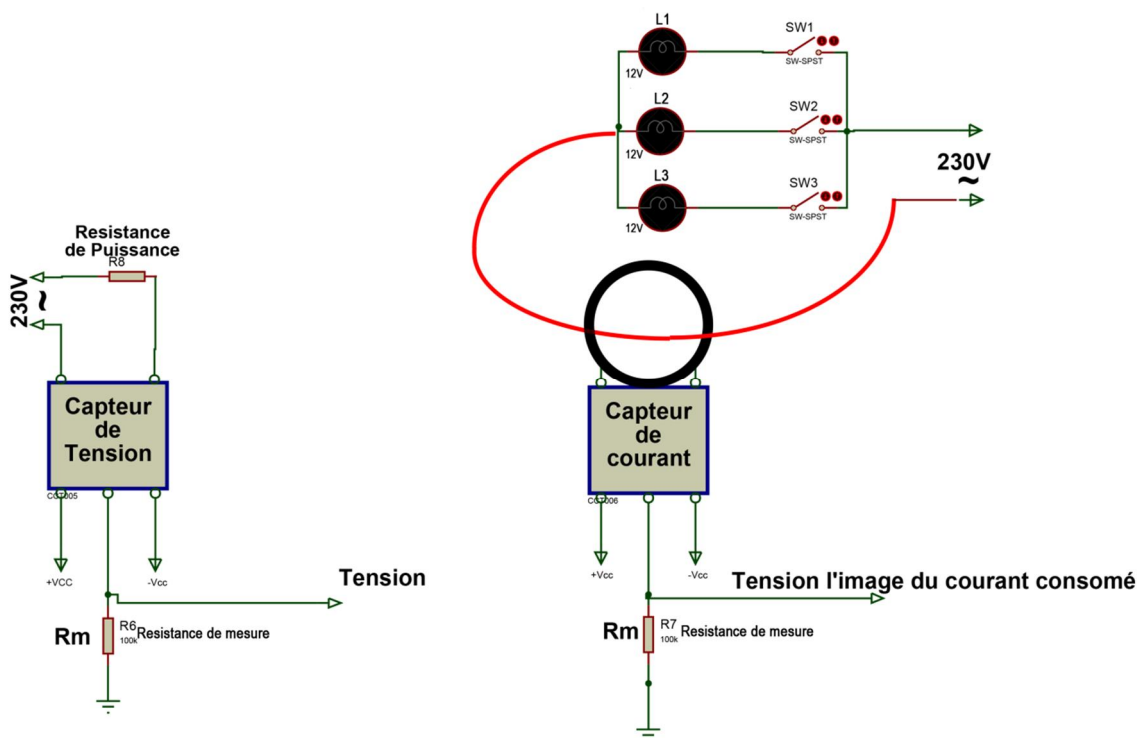


Figure II.11 : Circuit de mesure réalisé sous Isis.

La carte Arduino Mega 2560 est une carte à microcontrôleur basée sur un ATmega256. Cette carte dispose [15] :

- de 54 broches d'entrées/sorties numériques (dont 14 peuvent être utilisées en sorties PWM (largeur d'impulsion modulée).
- de 16 entrées analogiques (utilisable en broches entrées/sorties numériques).
- de 4 UART (port série matériel),
- d'un quartz 16Mhz.
- d'une connexion USB.
- d'un connecteur d'alimentation jack.
- d'un connecteur ICSP (programmation "in-circuit").
- d'un bouton de réinitialisation (reset).

I.1.3.2. Caractéristiques [15]

Microcontrôleur	ATmega2560
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	54 (dont 14 disposent d'une sortie PWM)
Broches d'entrées analogiques	16 (utilisables en broches E/S numériques)
Intensité maxi disponible par broche E/S (5V)	40 mA (ATTENTION : 200mA cumulé pour l'ensemble des broches E/S)
Intensité maxi disponible pour la sortie 3.3V	50 mA
Intensité maxi disponible pour la sortie 5V	Fonction de l'alimentation utilisée - 500 mA max si port USB utilisé seul
Mémoire Programme Flash	256 KB dont 8 KB sont utilisés par le bootloader
Mémoire SRAM (mémoire volatile)	8 KB
Mémoire EEPROM (mémoire non volatile)	4 KB
Vitesse d'horloge	16 MHz

Figure II.13: Caractéristiques de l'Arduino Mega 2560.

I.1.3.3. Avantages [16]

- nombreuses entrées et sorties pour raccorder des capteurs ou des actionneurs.
- capacité de mémoire suffisante pour les gros projets.
- plus de broches UART (4 ports de communication série).
- plus de broches MLI (15 sorties numériques utilisable comme MU).
- compatible avec la plupart des Shields conçus pour l'Arduino Uno.

I.1.3.4. Alimentation de l'Arduino [15]

La carte Arduino Mega 2560 peut être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte.

L'alimentation externe (non-USB) peut être soit un adaptateur secteur (pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles.

L'adaptateur secteur peut être connecté en branchant une prise 2.1mm positif au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de piles ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelées Gnd (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation.

La carte peut fonctionner avec une alimentation externe de 6 à 20 volts. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V et la carte pourrait être instable. Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer et endommager la carte. Aussi, la plage idéale recommandée pour alimenter la carte Uno est entre 7V et 12V. 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
- 3V3 : Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de l'ordinateur et le port série de l'ATmega). L'intensité maximale disponible sur cette broche est de 50mA.
- GND: Broche de masse (ou 0V).

I.1.3.5. Entrées et sorties [15]

C'est par ces connexions (E/S) que le microcontrôleur est relié au monde extérieur. La carte Mega2560 dispose de 54 broches numériques peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique. Ces broches fonctionnent en 5V, chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité.

Comme elle dispose de 16 entrées analogiques, chacune pouvant fournir une mesure d'une résolution de 10 bits (c.à.d. sur 1024 niveaux soit de 0 à 1023). Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et une instruction du langage Arduino. Les broches analogiques peuvent être utilisées en tant que broches numériques.

I.1.3.6. Communication [15]

La carte Arduino Mega2560 dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs. L'ATmega2560 dispose de quatre UARTs (Universal Asynchronous Receiver Transmitter ou émetteur-récepteur asynchrone universel en français) pour communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX). Un circuit intégré ATmega8U2 sur la carte assure la connexion entre cette communication série de l'un des ports série de l'ATmega 2560 vers le port USB de l'ordinateur qui apparaît comme un port COM virtuel pour les logiciels de l'ordinateur. Le code utilisé pour programmer l'ATmega8U2 utilise le driver standard USB COM.

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

I.1.3.7. Logiciel

Le logiciel pour programmer l'Arduino est un logiciel gratuit et qui dispose d'une interface simple et lisible. C'est un logiciel de programmation par code, ce code qui contient une cinquantaine de commandes différentes. A l'ouverture, l'interface visuelle du logiciel ressemble à celle de la figure II.14.

Le logiciel Arduino a pour fonctions principales [16] :

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

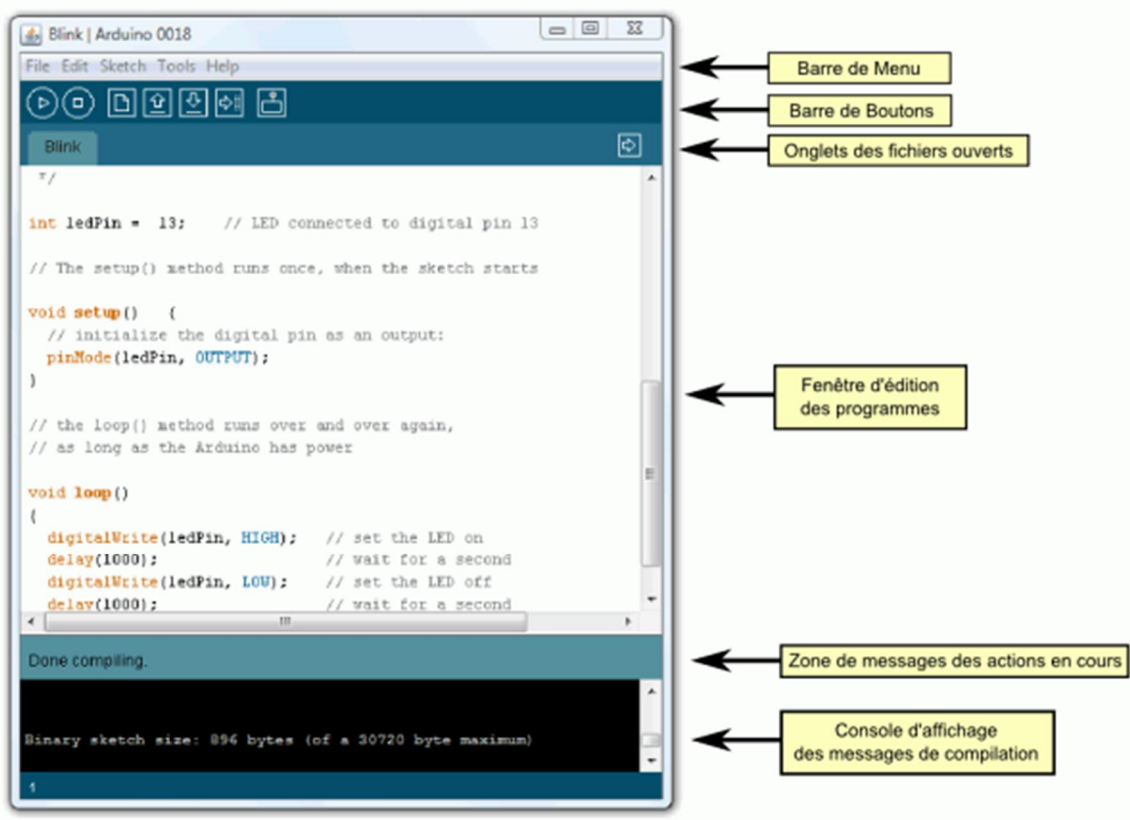


Figure II.14 : interface du logiciel Arduino.

Cet espace de développement intégré (EDI) dédié au langage Arduino et à la programmation des cartes Arduino comporte [17] :

- une BARRE DE MENUS comme pour tout logiciel une interface graphique (GUI),
- une BARRE DE BOUTONS qui donne un accès direct aux fonctions essentielles du logiciel et fait toute sa simplicité d'utilisation,
- un EDITEUR (à coloration syntaxique) pour écrire le code de votre programme, avec onglets de navigation,
- une ZONE DE MESSAGES qui affiche indique l'état des actions en cours,
- une CONSOLE TEXTE qui affiche les messages concernant le résultat de la compilation du programme

Description de la barre des boutons

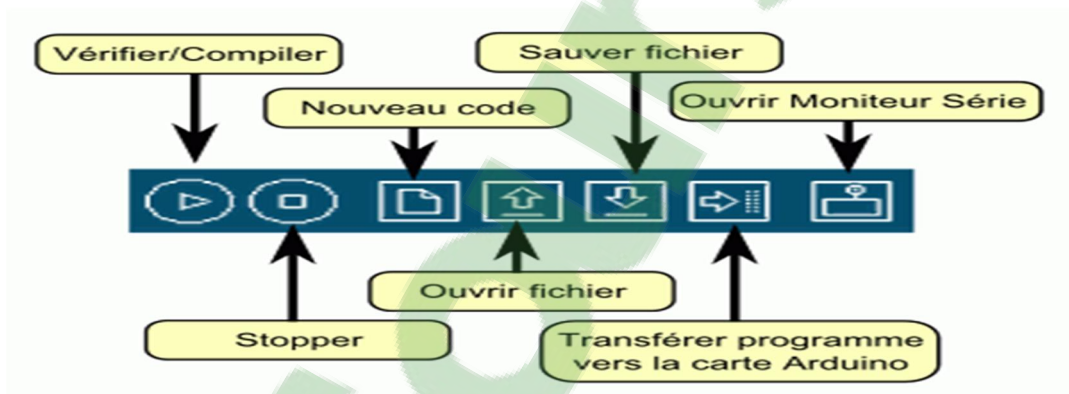


Figure II.15 : La barre des boutons.

Vérifier/compiler : Vérifie le code à la recherche d'erreur.

Stop : Stoppe le moniteur série ou les autres boutons activés.

Nouveau : Crée un nouveau code (ouvre une fenêtre d'édition vide).

Ouvrir : Ouvre la liste de tous les programmes dans votre "livre de programmes". Cliquer sur l'un des programmes l'ouvre dans la fenêtre courante.

Sauver : Enregistre votre programme.

Transférer vers la carte : Compile votre code et le transfère celui-ci vers la carte Arduino. Voir ci-dessous "Transférer les programmes" pour les détails.

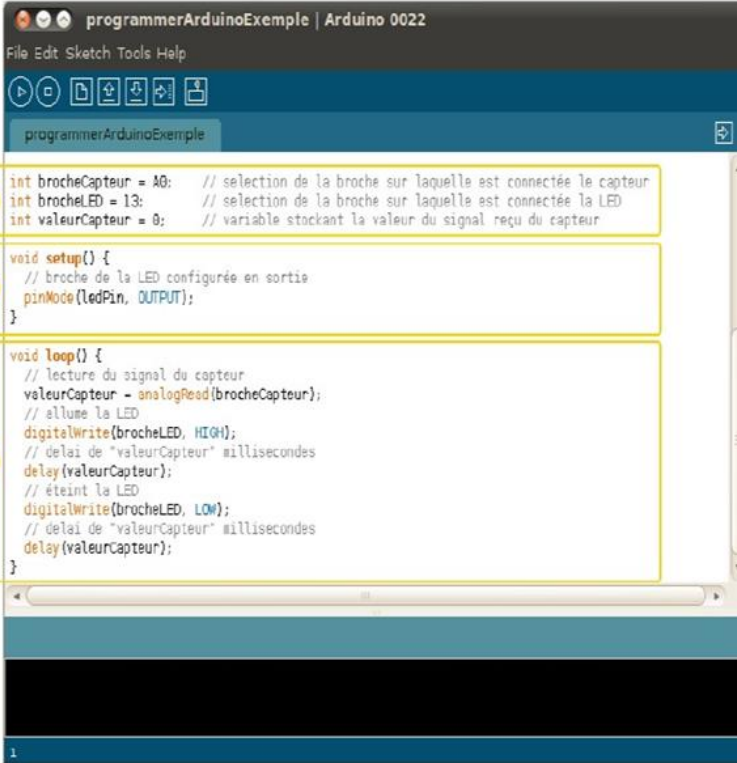
Moniteur Série : Ouvre la fenêtre du moniteur (ou terminal) série.

I.1.3.8. Structure d'un programme Arduino

Un programme Arduino comporte trois parties :

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

Dans chaque partie d'un programme sont utilisées différentes instructions issues de la syntaxe du langage Arduino (Figure II.16).



```

programmerArduinoExemple | Arduino 0022
File Edit Sketch Tools Help
programmerArduinoExemple

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

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

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

Figure II.16: Structure d'un programme Arduino.

I.1.3.9. Le Moniteur Série [18]

Le logiciel Arduino intègre également un **TERMINAL SERIE** (fenêtre séparée) qui permet d'afficher des messages textes reçus de la carte Arduino et d'envoyer des caractères vers la carte Arduino. Cette fonctionnalité permet une mise au point facilitée des programmes, permettant d'afficher sur l'ordinateur l'état de variables, de résultats de calculs ou de conversions analogique-numérique : un élément essentiel pour améliorer, tester et corriger ses programmes. (Figure II.17)

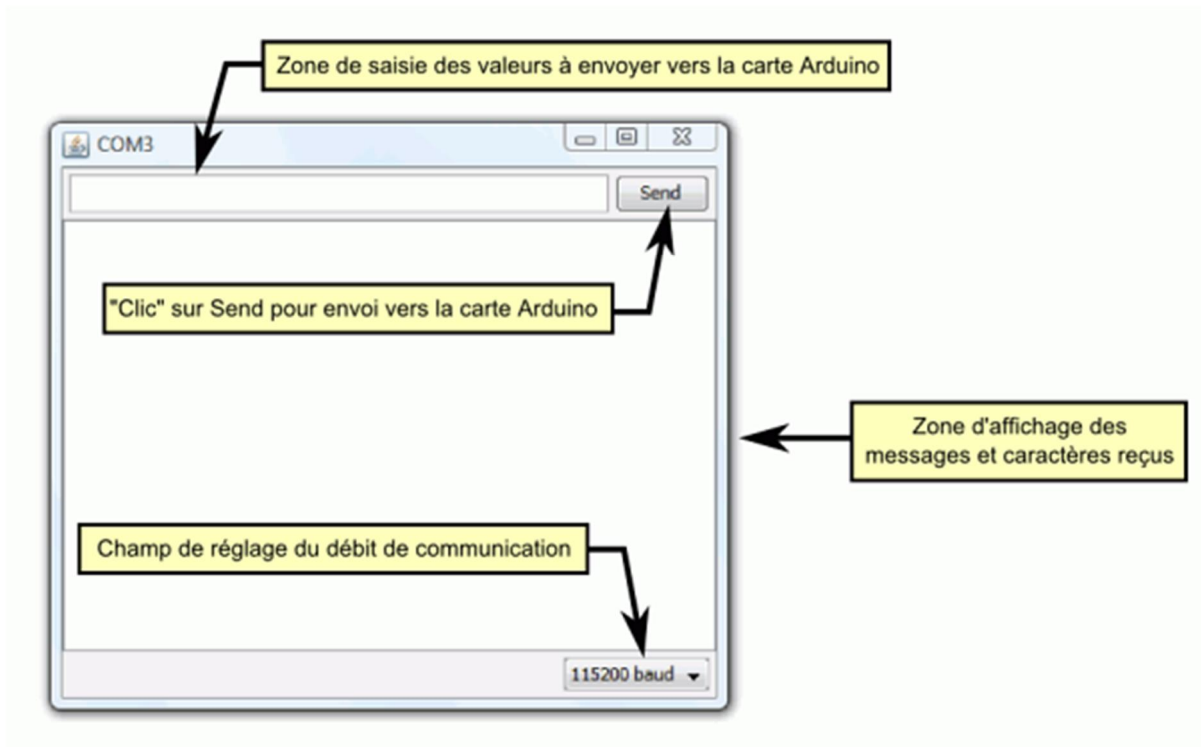


Figure II.17 : Interface du moniteur série.

I.1.3.10. Bibliothèques

Les bibliothèques fournissent des fonctions nouvelles qu'on peut les utiliser dans nos programmes, par exemple pour notre programme on doit utiliser un matériel précis (un afficheur LCD, Bluetooth...) ou manipuler des données. Pour utiliser une bibliothèque, la sélectionner depuis le menu **Sketch > Import Library** (Programme > Importer Bibliothèque). Cela insèrera une ou plusieurs instructions `#include` au début de notre programme et compilera la bibliothèque avec notre programme.

I.1.3.11. Transmission et affichage

Cette carte est dotée de plusieurs ports de communication série utilisés soit lors d'une connexion filaire ou sans fils via un module de communication. Dans notre cas on va éclaircir actuellement les types qu'on a choisis pour notre réalisation qui sont le module Bluetooth et l'afficheur LCD.

a. Le module Arduino Bluetooth

Le Module Microcontrôleur Arduino Bluetooth est la plateforme populaire Arduino avec une connexion série Bluetooth à la place d'une connexion USB, très faible consommation d'énergie, très faible portée (sur un rayon de l'ordre d'une dizaine de mètres), faible débit, très bon marché et peu encombrant.

Il existe deux modules de Bluetooth le HC05 et HC06. Ils permettent simplement de disposer d'une liaison Bluetooth sur un projet.

Notre choix s'est porté sur le module HC-05. Il possède 6 pins, ce qui permet de l'alimenter soit en 5V soit en 3.3v. Il est également possible de désactiver/activer le module ou d'obtenir son état (visible ou non) depuis 2 broches supplémentaires. Cependant, il ne supporte que la transmission sur 3,3V, ce qui nous obligera à mettre un pont diviseur entre la broche de réception du module et la broche de transmission de l'Arduino.

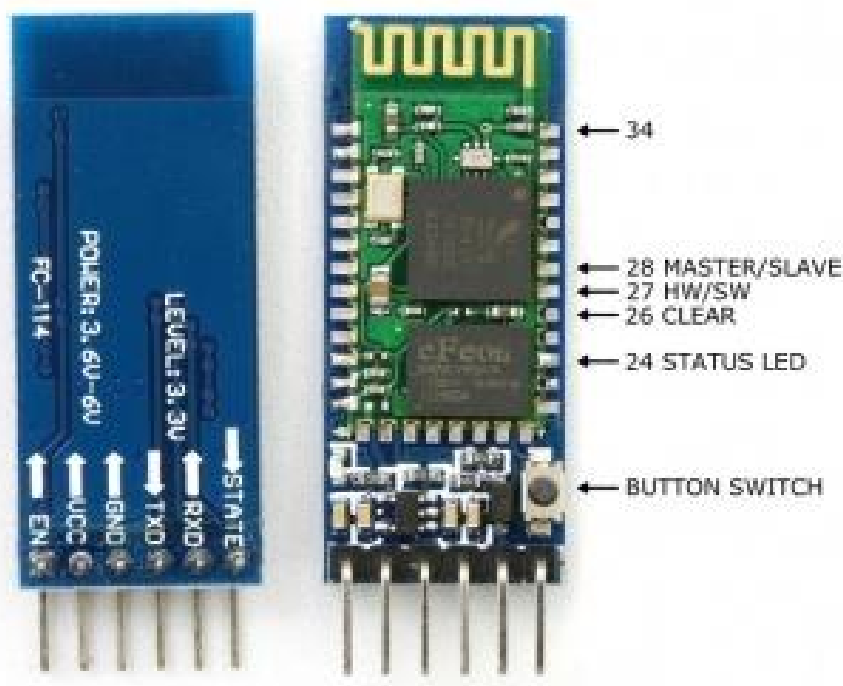


Figure II.18: Module Bluetooth HC05.

➤ Branchement

Pour notre Arduino Mega on doit brancher le module avec +5v au +5v de l'Arduino, le GND à la masse de l'Arduino, le Rx du module au pin Tx1 (ou Tx2 ou Tx3) de l'Arduino et enfin le Tx du module au pin Rx1 (ou Rx2 ou Rx3) de l'Arduino

Il y a 6 pattes dont 4 uniquement sont utilisées couramment. Dans l'ordre:

- KEY : non utilisé (sauf configuration du HC05).
- VCC : alimentation en 5V de l'Arduino.
- GND : à relier au GND de l'Arduino.
- TXD : à relier au RX de l'Arduino (Le signal émis vers Arduino est de 0 ou 3.3V mais cela suffit).
- RXD : à relier au TX du Arduino, mais par le biais d'un diviseur de tension. En effet, la tension d'entrée acceptée est de 0 ou de 3.3V, alors que le TX de l'Arduino émet du 0 ou du 5V
- STATE : non utilisé.

Pour le diviseur de tensions, on pourra utiliser deux résistances :

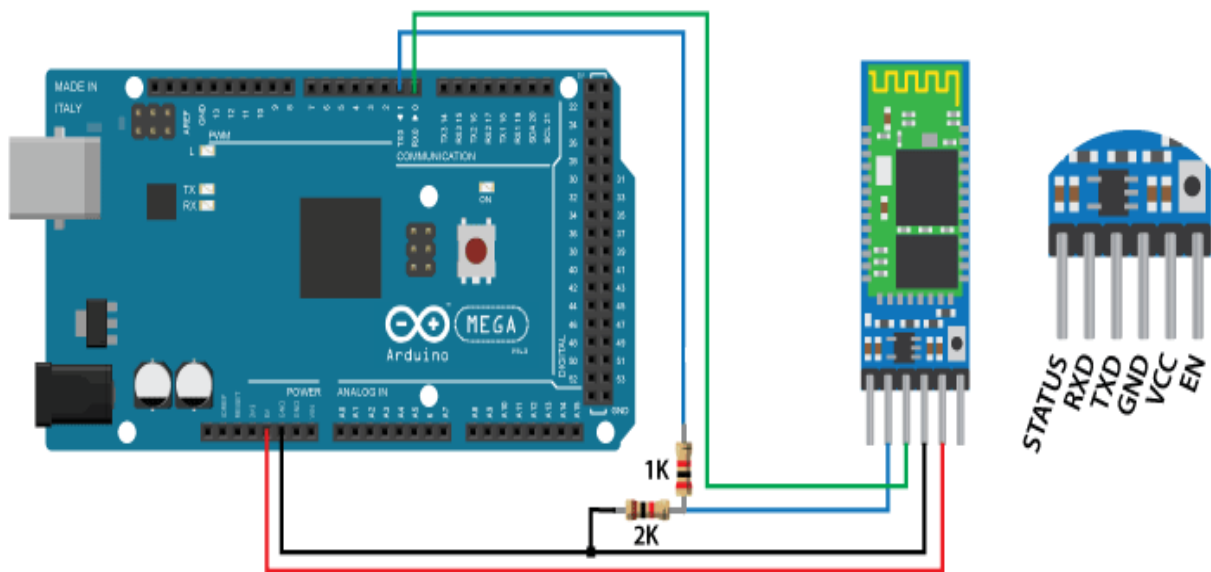


Figure II.19 : Branchement du module HC05 avec Arduino.

b. Afficheur LCD

Les afficheurs LCD sont devenus indispensables dans les systèmes techniques qui nécessitent l'affichage des paramètres de fonctionnement.

Ces Afficheurs permettent d'afficher des lettres, des chiffres et quelques caractères spéciaux. Les caractères sont prédéfinis.

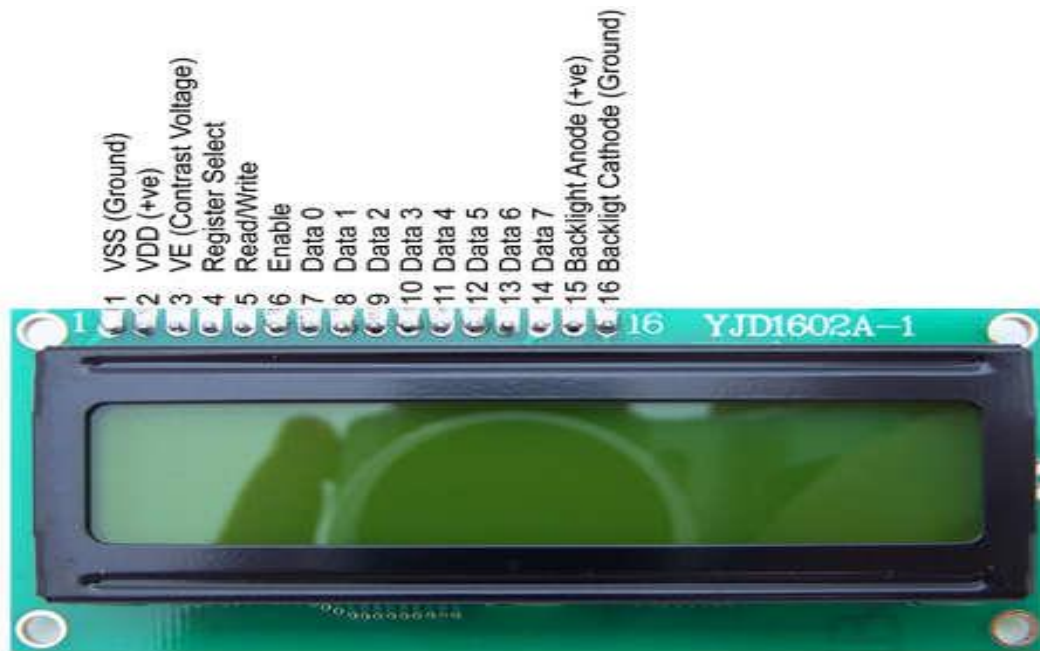


Figure II.20 : Afficheur LCD.

L'afficheur dispose d'un rétro-éclairage à LED et peut afficher 2 lignes avec 16 caractères par lignes.

On peut identifier chaque rectangle dans lesquels sont affichés les caractères. Pour chacun des caractères, et même identifier les différents points ("pixel" en anglais) le composant. L'afficheur affiche du texte blanc sur fond bleu, sa seule tâche est d'afficher du texte. Dans notre réalisation, nous allons utiliser un exemple Arduino de la librairie LCD, après nous verrons comment afficher l'énergie consommé et le niveau de luminosité.

L'afficheur LCD a besoin de 6 broches/pins Arduino pour fonctionner. Elles doivent toutes être déclarée comme des sorties digitales. L'afficheur à également besoin d'une connexion 5V et GND (masse). On a ajouté un potentiomètre de 4.7k pour réglé sa luminosité.

I.1.3.12. Bloc de traitement

L'unité de traitement est constituée d'un Smartphone qui fonctionne sous la plateforme android là où on doit réaliser une application (voir chapitre III) qui doit communiquer avec notre compteur via le module Bluetooth, pour assurer la lecture et le prélèvement des données afin de les stocker dans la mémoire interne de notre Smartphone ou sur un disque externe.

I.1.3.13. Simulation du circuit électronique

Avant de passer à la réalisation pratique, nous avons utilisé un logiciel de CAO: il s'agit de ISIS PORTEUS, c'est un logiciel de simulation électronique perfectionné conçu par Labcenter Electronics qui permet de dessiner des schémas électroniques, de les simuler et de réaliser le circuit imprimé correspondant.

La figure suivante présente le circuit électronique de notre carte de mesure ainsi que la transmission par Bluetooth. On a ajouté trois LED (blanche, verte, rouge) dans notre circuit pour vérifier respectivement l'activation du Bluetooth et la présence du courant et tension

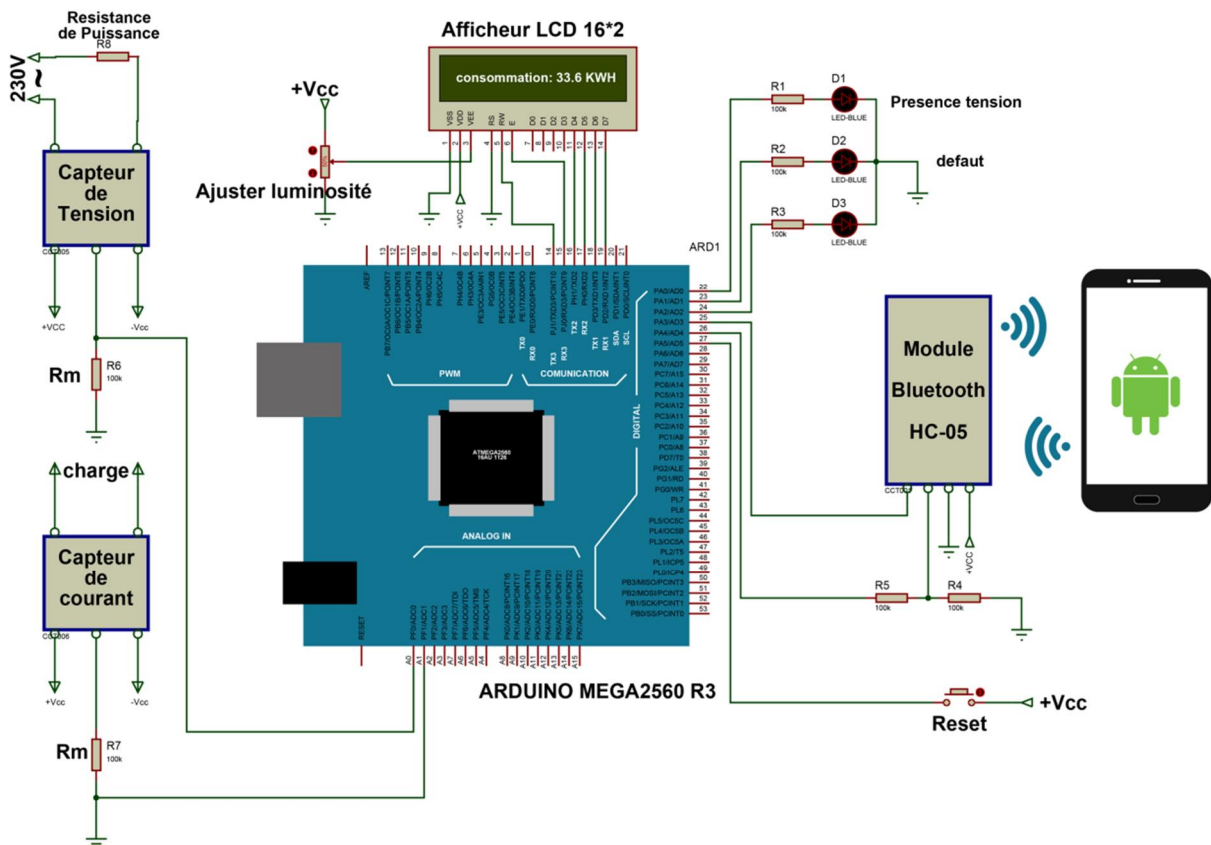


Figure II.21 : Le circuit de la carte électronique réalisé sous ISIS-ORTEUS.

L'organigramme suivant explique l'architecture du fonctionnement de notre système afin de synchroniser les données avec l'Arduino d'un côté et le Smartphone d'autre côté.

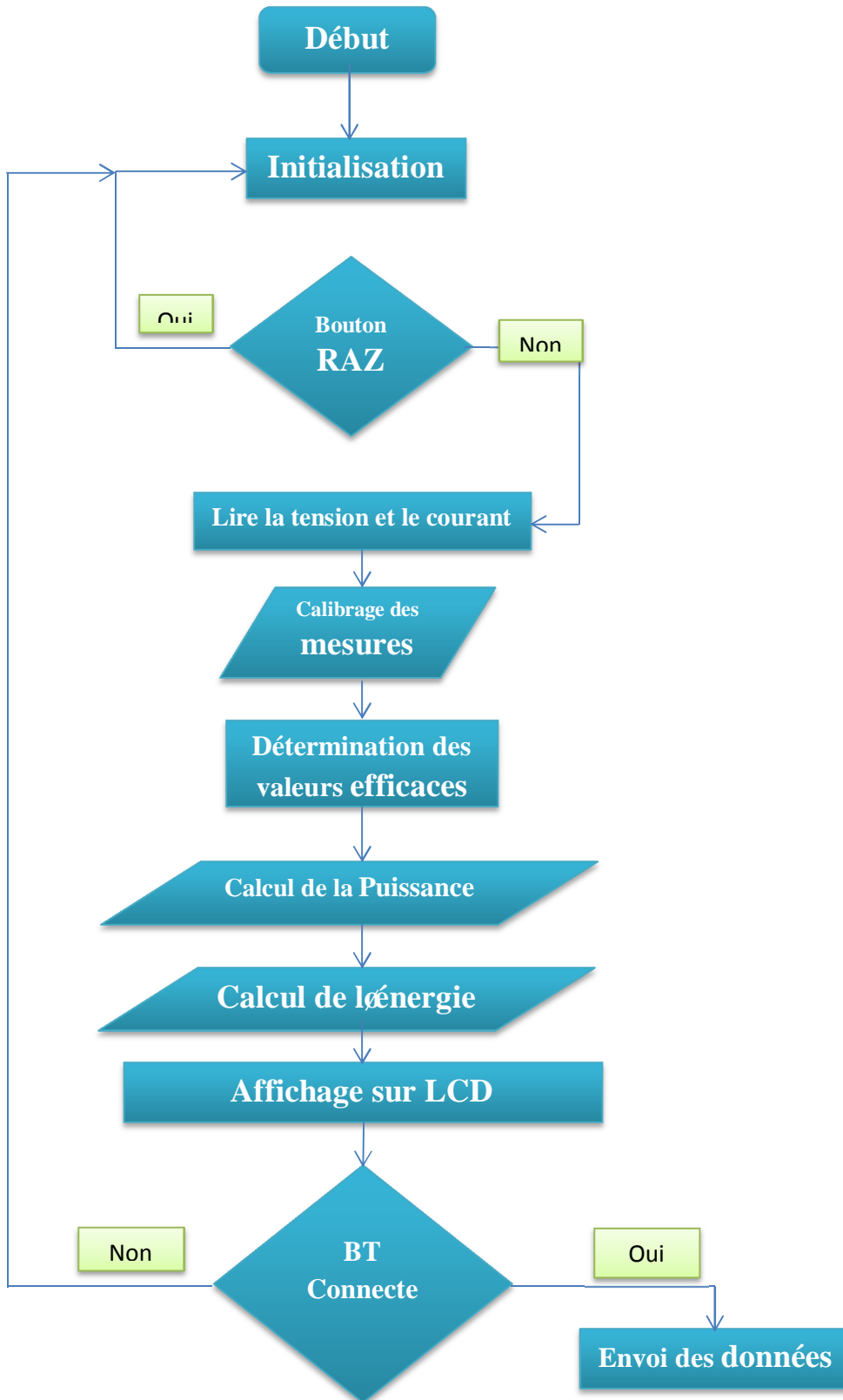


Figure II.22 : Compteur intelligent.

I.2. Interprétation des résultats

I.2.1. Capteur de tension

Afin de confirmer les caractéristiques techniques de nos capteur présentées en datasheet en terme de linearité et precision et afin de pouvoir calibrer notre système, nous avons injectés des signaux avec différentes amplitudes les signaux illustrés dans la figure (II.23) relevé à l'aide d'un oscilloscope a mémoire (canal jaune comme entrée et canal bleu comme sortie), permet de conclure que pour chaque tension x délivrée par le capteur est proportionnelle à celle injecter à l'entrée du capteur avec un rapport d'environ 183.

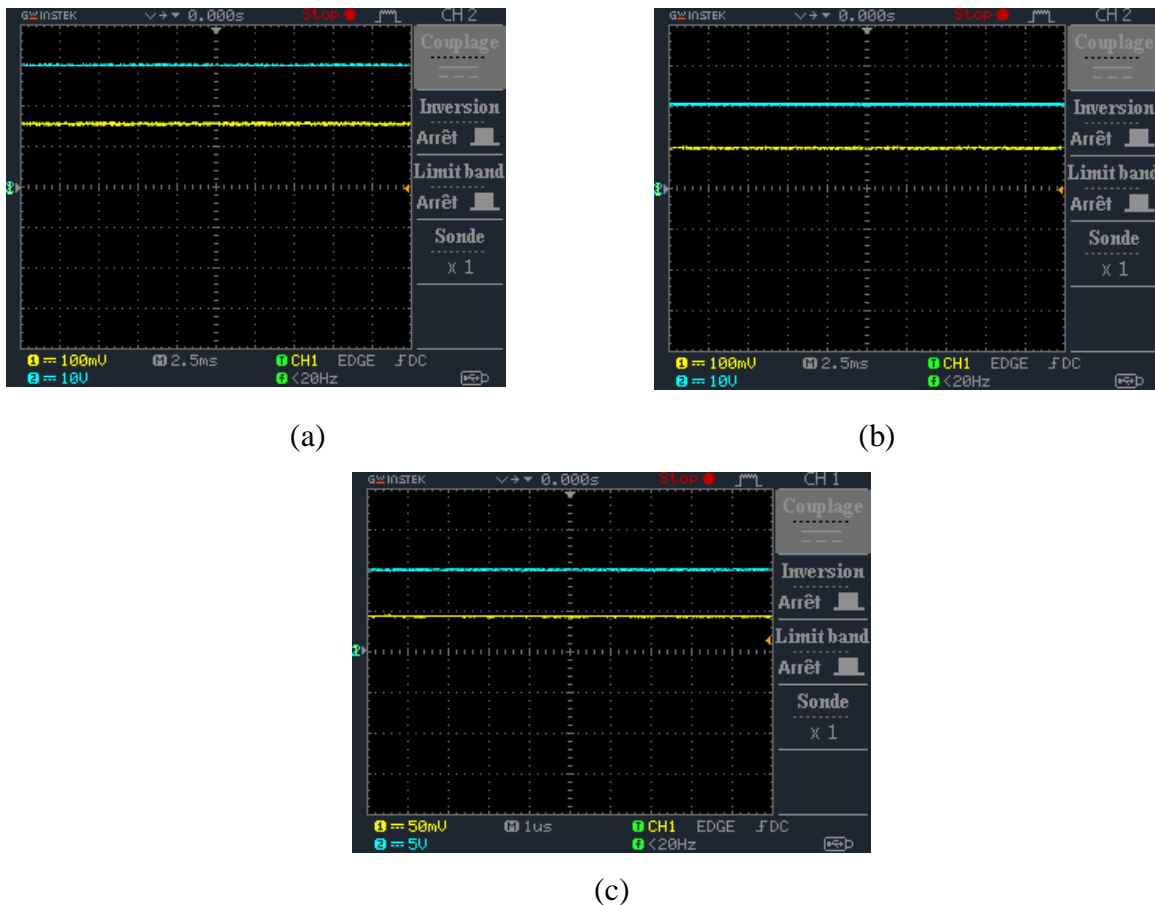


Figure II.23 : Signaux d'entrées/sorties au niveau du capteur de tension.

A ce moment nous avons injecté comme entrée notre signal du secteur (230v), le resultat obtenu est donné dans la figure ci-dessous qui nous confirme le rapport obtenu auparavant.

- la valeur efficace du secteur est de 230 V
- la valeur efficace du signal de sortie (voir figure II.24) est donnée par :

$$U_1 = U_{eff} \cdot 1.26 = 1.26 U_{eff} \quad (eq.1)$$

Donc le rapport est :

$$U_1 = 230 \cdot 1.26 = 183 \quad (eq.2)$$

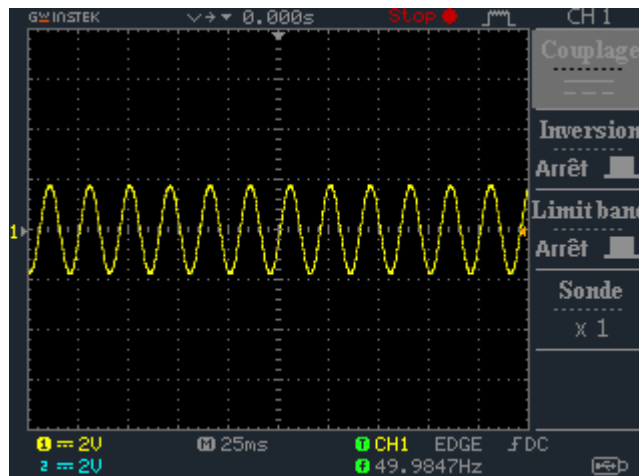


Figure II.24 : Signal à la sortie du capteur de tension pour une tension d'entrée de 230V.

I.2.2. Capteur de courant

Nous avons appliqué le même raisonnement pour le capteur de courant, avec le changement de la valeur de la charge en utilisant 3 lampes en parallèle avec différentes valeurs de puissance.

- Cas d'une lampe de 75 watt

D'après la relation :

$$P = U \cdot I \iff I = \frac{P}{U} \quad (eq.3)$$

En remplaçant les valeurs de la puissance de la lampe et la tension du secteur on obtient $I = 0.32$ A. La (figure II.25) nous donne la valeur de la tension mesurée à la sortie du capteur et qui est proportionnelle à la valeur du courant qui circule dans la charge formée d'une lampe de 75 watt.

$$I_{RMS} = 200 \sqrt{2} = 141 \text{ mA} \quad (\text{eq.4})$$

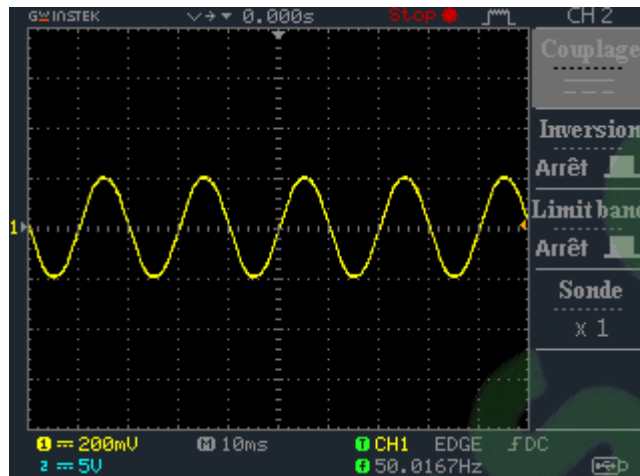


Figure II.25 : Image du courant à la sortie du capteur (cas une lampe 75W).

- Cas de trois lampes (2 lampes de 75watt et une de 200 watt)

Comme les trois charge sont alimentées la puissance globale est la somme des trois puissance c'est-à-dire :

$$P = 75 + 75 + 200 = 350 \text{ W} \text{ et } I = 1.52 \text{ A} \quad (\text{eq.5})$$

L'image de cette consommation est donné par la figure II.26.

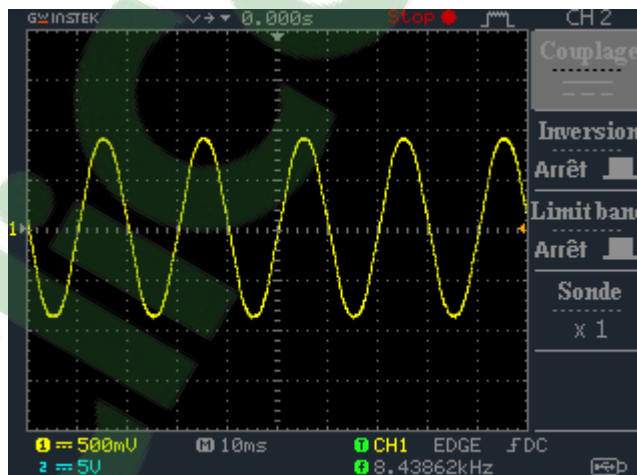


Figure II.26: Image du courant à la sortie du capteur(cas deux lampes 275W).

Le tension relevé par l'oscilloscope est de 0.63 V efficace. Ce qui nous donne un rapport de courant :

$$I = 1.52 \times 0.63 = 2.39 \text{ A} \quad (\text{eq.6})$$

C'est à dire que pour un courant I nous aurons une tension a la sortie égal à

$$0.44 * I \text{ (eq.7)}$$

Et en fin pour calibrer notre système d'une manière adéquate il suffit de :

- multiplier la valeur délivrée par le capteur de tension avec le rapport 183 afin de lire la tension exacte de l'entrée.
- multiplier la valeur délivrée par le capteur du courant avec le rapport 2.26 afin de lire la valeur du courant exacte qui circule dans les charges .

Conclusion

Durant ce chapitre on a entamé la partie pratique de la réalisation de notre compteur communicant, pour cela différentes étapes s'imposent dont on peut citer la réalisation du circuit de mesure ainsi que la conception des différents schémas électroniques représentant notre unité de traitement de données et le bloc de transmission, cet ensemble sera alimenté par une alimentation stabilisée et une pile de 9v en cas d'absence de la tension du secteur, ensuite on a fait des tests pour vérifier la linéarité des capteurs ,ce que nous a permis de faire un calibrage adéquat pour avoir des résultats précis.

Notre compteur intelligent et à l'aide du module Bluetooth aura la capacité de communiquer avec notre Smartphone pour nous permettre de lire les données mesurées par notre compteur.

La partie suivante de notre travail prendra en charge la conception d'une application Android qui sera l'outil responsable de la lecture des données lues depuis notre compteur intelligent et les stockées sur tout appareil fonctionnant sous Android.

Chapitre III

Conception d'une application Android

Introduction

L'Android est parmi les derniers systèmes d'exploitation qui développent les exigences des téléphones intelligents. La plateforme android des Smartphones devient de plus en plus importante pour les réalisateurs de logiciel, en raison de ses puissantes possibilités et open source.

Lors des dernières années, le traitement des données informatiques se fait par des ordinateurs, en revanche le smart phone a des avantages qui ont les mêmes fonctions que l'outil informatique, ce dernier porte l'intérêt de l'ordinateur grâce à l'android.

Les Smartphones sont devenus aujourd'hui omniprésents dans nos vies, et les tablettes commencent à se démocratiser. Ce sont de véritables mini-ordinateurs qui permettent d'exécuter des applications plus ou moins sophistiquées. Les éditeurs des systèmes d'exploitation pour ces appareils incitent les développeurs à créer des applications, et fournissent pour cela des outils pour faciliter le développement.

La plate-forme Android apporte tout cela au consommateur, mais surtout, elle affranchit le développeur de nombreuses contraintes. Par son ouverture; elle permet à n'importe quel développeur de créer ses applications gratuitement. Le Framework et le système d'exploitation et outils associés ont un code source ouvert, leur accès est gratuit et illimité. Plus besoin de négocier avec le constructeur du téléphone pour qu'il vous laisse développer sur sa plate-forme. Tous les développeurs sont ainsi sur un même pied d'égalité, tous peuvent ajouter de la mobilité à des applications existantes [19].

L'objectif de cette partie est de présenter le système d'exploitation Android, et l'outil de développement Android studio. Des outils que nous allons utiliser pour développer notre application.

III.1. Historique

L'histoire d'Android a vu le jour en octobre 2003, où la société Android Inc a été créée. Officiellement, elle développe des logiciels pour mobiles. Mais en réalité, elle se préparait à sortir un tout nouveau système d'exploitation pour Smartphones. En 2005, Google rachète cette entreprise, et sort une première bêta en novembre 2007, avant de lancer la version 1.0 en septembre 2008 avec le HTC Dream. À partir de ce moment-là, le rythme des nouvelles sorties est très élevé : pas moins de 11 versions différentes sont sorties en 3 ans. On remarquera au passage que chaque version d'Android porte le nom d'un dessert [21].

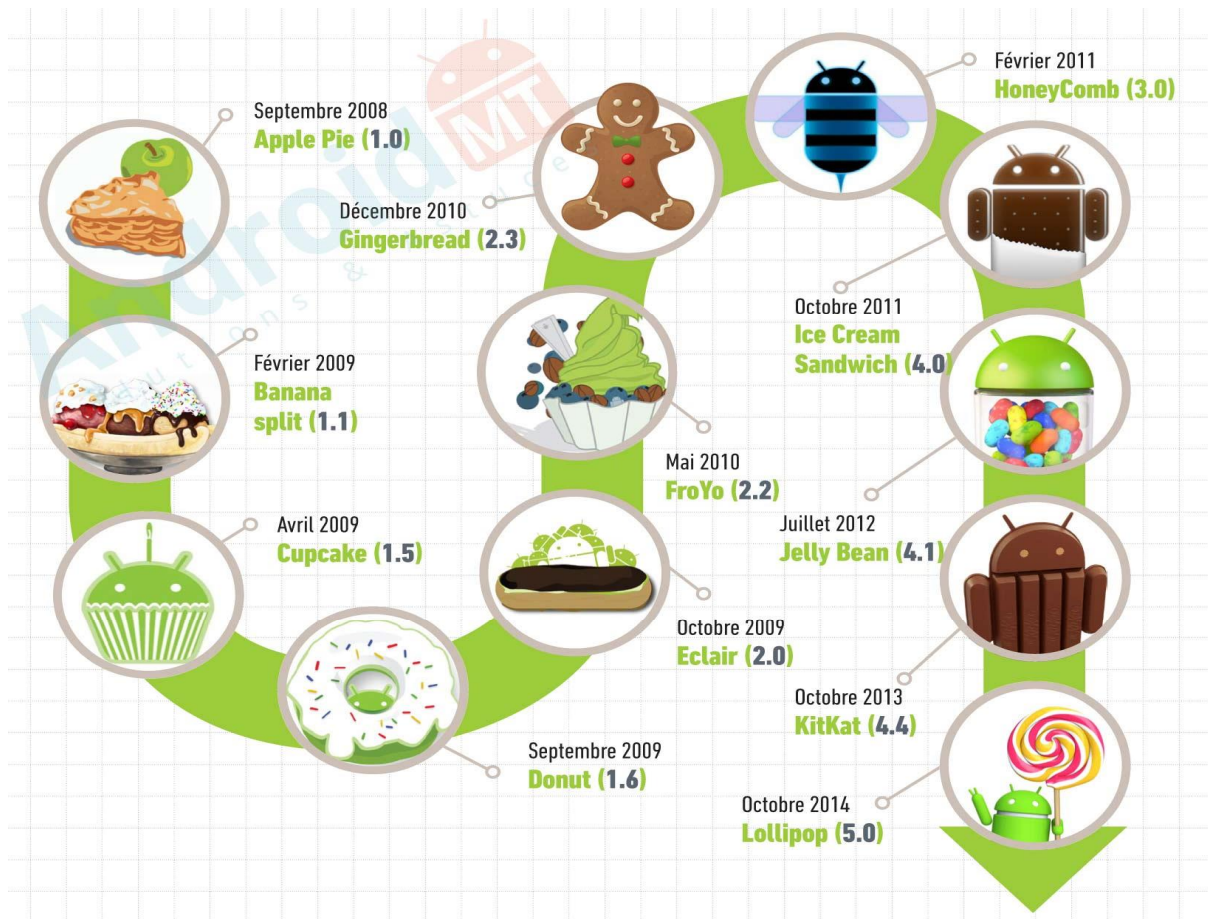


Figure III.1 : Les différentes versions d'Android [20].

III.2. Etude de la plateforme Android

Android est un système d'exploitation pour Smartphones, tablettes tactiles, PDA et terminaux mobiles. C'est un système open source, utilisant le noyau Linux, conçu par Android, une startup rachetée par Google, et annoncé officiellement le 5 novembre 2007. D'autres types d'appareils possédant ce système d'exploitation existent, par exemple des téléviseurs, des montres, des autoradios et même des voitures [20].

Android est gratuit, pour les constructeurs d'appareils souhaitant l'utiliser, et partiellement open-source.

Concrètement, ce système d'exploitation s'appuie sur un noyau Linux optimisé pour un usage mobile, et sur une machine virtuelle Java assez fortement modifiée, nommée Dalvik JVM.

Celle-ci n'exécute pas les (.class) habituels, mais des fichiers portant l'extension (.dex), compilés différemment et optimisés par le SDK Android.

Le développement se fait donc en Java, mais sur cette JVM spécifique à Android. Il n'est, du fait, pas possible d'utiliser n'importe quelle librairie Java dans une application Android [20].

III.2.1. Architecture de la plateforme Android [20]

L'architecture de la plateforme Android se décline, selon une démarche bottom up, en quatre principaux niveaux que sont :

- le noyau linux.
- les librairies et l'environnement d'exécution.
- le module de développement d'applications
- les différentes applications.

La figure ci-dessous nous montre les différentes parties constituant l'architecture de la plateforme de l'Android [20].



Figure III.2 : L'architecture de la plateforme Android [20].

III.3. Le Développement sous Android

III.3.1. L'environnement de développement sous Android

Pour pouvoir développer des applications sous Android, un ensemble d'outils est nécessaire. Vu que les procédures d'installation de ces outils sont assez longues et embêtantes, alors les décrire, pas à pas, risquerai de prendre énormément de place et ainsi beaucoup de pages. Alors, on se contentera juste d'évoquer les outils et leurs intérêts.

III.3.2. Langage JAVA

JAVA est un langage de programmation orienté objet, développé par Sun Microsystems, sorti en 1995. Sun Microsystems est racheté en 2009 par Oracle, une application écrite en JAVA et facilement portable sur plusieurs systèmes d'exploitation. Une application exécutable sous Android (interprétable par une interface en JAVA) est un fichier avec l'extension « APK » [19]

III.3.3. Le SDK (Software Development Kit) Android

Le SDK contient :

- Les bibliothèques Java pour créer des logiciels.
- Les outils de mise en boîte des logiciels.
- Un émulateur de tablettes pour tester les applications AVD.
- Un outil de communication avec les vraies tablettes ADB.

Un SDK, c'est-à-dire un kit de développement logiciel, est un ensemble d'outils que met à disposition un éditeur afin de permettre de développer des applications pour un environnement précis. Le SDK Android permet, donc, de développer des applications pour Android et uniquement pour Android.

Au premier lancement du SDK, un écran semblable à la figure suivante s'affichera :

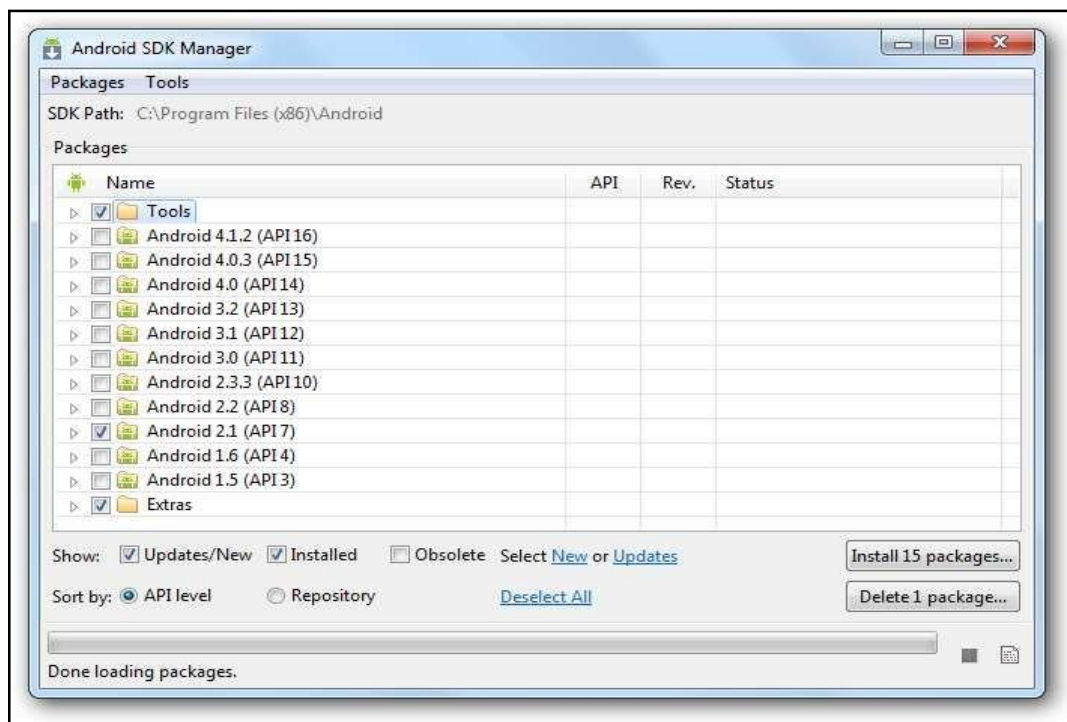


Figure III.3 : Le SDK Android.

En observons bien le nom des paquets, on remarquera qu'ils suivent tous un même motif. Il est écrit à chaque fois : Android [un nombre] (API [un autre nombre]). La présence de ces nombres s'explique par le fait qu'il existe plusieurs versions de la plateforme en circulation. Le premier nombre correspond à la version d'Android et le second à la version de l'API Android associée. Quand on développe une application, il faut prendre en compte ces numéros, puisqu'une application développée pour une version précise d'Android ne fonctionnera pas pour les versions antérieures [21].

III.3.4. Le JDK (Java Development Kit) [21]

Les applications développées pour Android étant essentiellement écrites en langage java, un langage de programmation orienté objet qui a la particularité d'être très portable. Cela signifie qu'un programme Java, fonctionnant sur Windows (par exemple), pourra facilement tourner sur Mac ou GNU/Linux.

Cette petite prouesse vient du fait que Java s'appuie sur une machine virtuelle pour s'exécuter (appelée la JVM). Pour avoir une JVM sur votre ordinateur, il vous faut télécharger le JRE. Ce dernier contient, en plus de la JVM, des bibliothèques Java standards.

La JVM ne lit pas directement le code Java. Elle lit un code compilé (le byte code). Pour passer du code Java, que le développeur écrit, au code compilé, lu par la JVM, des outils spéciaux sont nécessaires. Ces outils sont inclus dans le JDK. De plus, le JDK contient le JRE (et donc la machine virtuelle), ce qui est bien pratique

Pour résumer, on dit que :

- Pour un simple utilisateur de Java : il doit avoir le JRE.
- Pour un développeur : il aura besoin des outils du JDK.

III.4. Architecture d'un projet sous Android

Les projets ont changé d'architecture depuis le passage à Android Studio, où l'on utiliserait plusieurs projets sous Eclipse, avec des dépendances entre chacun, nous parlerons maintenant d'un seul Projet, contenant plusieurs modules [22]. Une activité, vis-à-vis du code, est séparée en 2 morceaux: un fichier XML, et un fichier JAVA (figure III.4).

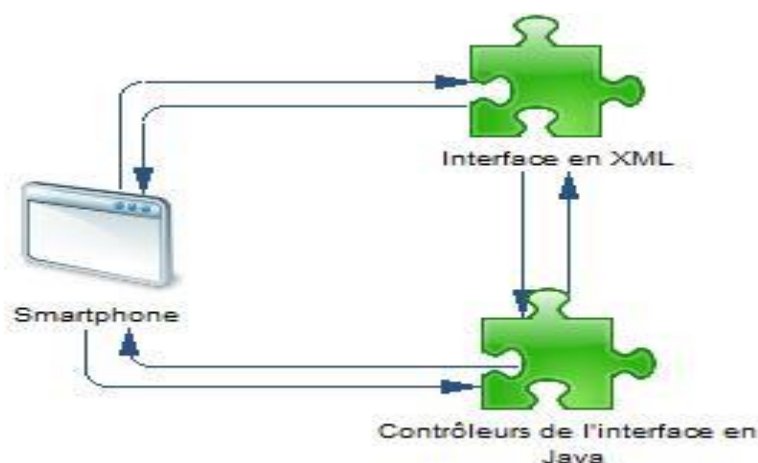


Figure III.4 : Architecture basique d'une application Android [23].

Pour l'application, nous avons donc un projet appelé par exemple **HelloAndroidStudio** contenant un module **App**. Les modules ont la forme suivante :

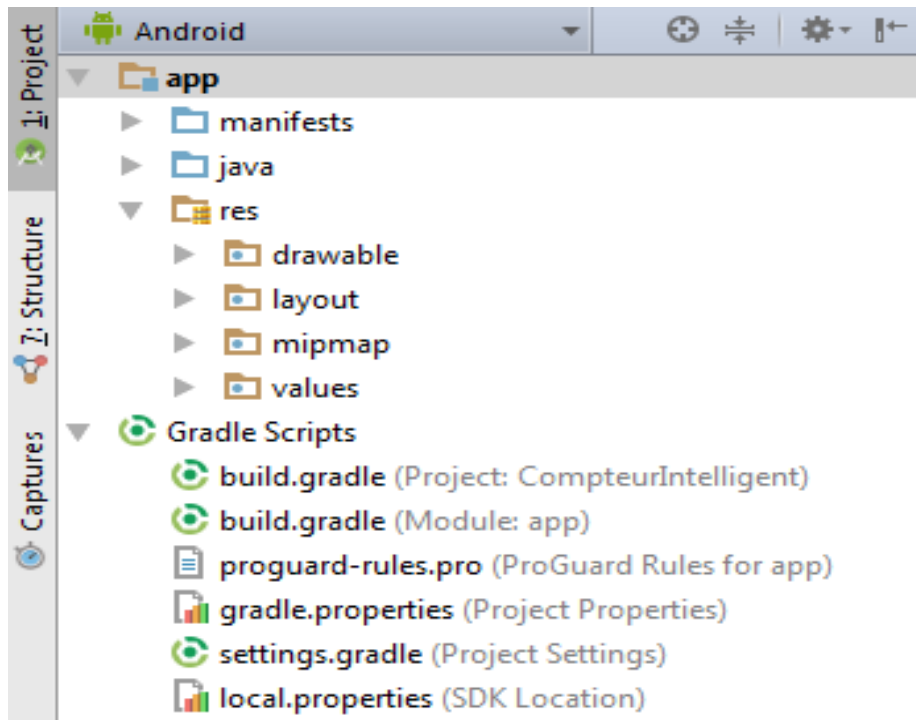


Figure III.5 : Architecture d'un projet sous Android.

Tout comme bon nombre de technologies actuelles, les sources d'une application Android possèdent une structure bien définie qui doit être respectée. Ces arborescences permettent non seulement de rendre les projets plus lisibles et organisés, mais aussi de simplifier le développement. Lors de la création d'un nouveau projet, voici l'arborescence qui est automatiquement générée [22] :

- **AndroidManifest.xml** : fichier XML décrivant l'application et ses composants, tels que les activités, les services, etc. Lors de la création d'une activité, une erreur courante pour un premier projet Android est de oublier de la déclarer dans le fichier Manifest. C'est une étape indispensable pour le fonctionnement de l'application. Le Manifest est, en quelque sorte, la carte d'identité de l'application, et permet d'autoriser l'exécution des activités et autres actions de l'application.

- **res** : répertoire contenant toutes les ressources telles que les images, les vues de l'interface graphique, etc., nécessaires à l'application. Ce répertoire est structuré par défaut de la manière suivante :

- ✓ **res/drawable** : contient les ressources de type image.
- ✓ **res/layout** : contient les descriptions des interfaces graphiques au format XML (les vues).
- ✓ **res/xml** : contient les fichiers XML supplémentaires (non présents par défaut).
- ✓ **res/menu** : contient la description des menus, composants très courants d'une vue.
- ✓ **res/values** : contient diverses ressources, telles que les textes, qui sont empaquetées sans aucun traitement.

• **Le fichier build.gradle** : sert dans la configuration pour le nouveau moteur de production nommé **Gradle**, qui sera utilisé pour construire notre application afin de la déployer sur notre Smartphone ou sur le Play Store.

Au moment de la compilation du projet, l'application finale est générée au format **APK**, dans le répertoire bin de l'arborescence. C'est ce fichier qu'il faut ensuite déployer sur les équipements, afin de pouvoir faire tourner l'application.

III.5. Composants principaux d'une application Android

Il existe 3 fichiers majeurs dans une application Android:

- **MainActivity.java**

C'est là que passe le code Java. Il contrôle la façon dont l'application fonctionnera.

- **Activity_main.xml**

Cela contient la mise en page de l'application, c'est-à-dire les composants ou les boutons similaires aux widgets, TextViews etc.

- **AndroidManifest.xml**

C'est là que vous **définissez** quand l'application doit démarrer, les autorisations dont elle a besoin et le matériel dont elle a besoin.

Il existe d'autres fichiers, mais ils sont tous liés avec l'aide de ces trois.

III.6. Cycle de vie d'une application Android

Durant le démarrage d'une application sur Android, il y a un cycle de procédure à respecter (figure III.6). Ce qui permet au démarrage de choisir l'interface à mettre en place ainsi que les fonctionnalités des éléments de celle-ci. Le cycle de vie d'une application permet aussi de pouvoir libérer l'espace durant une autre activité [21].

La méthode `onCreate ()` est appelée à la création de votre activité. Elle sert à initialiser l'activité ainsi que toutes les données nécessaires à cette dernière. Quand la méthode `onCreate ()` est appelée, on lui passe un `Bundle` en argument. Ce `Bundle` contient l'état de sauvegarde enregistré lors de la dernière exécution de l'activité [21].

La méthode `onStart ()` signifie le début d'exécution de l'activité (début du passage au premier plan). Si l'activité ne peut pas aller en avant plan, pour une quelconque raison, elle sera transférée à `onStop ()`. La méthode `onResume ()` est appelée lorsque l'activité commencera à interagir avec l'utilisateur juste après avoir été dans un état de pause.

La méthode `onPause ()` est appelée au passage d'une autre activité en premier plan. L'intérêt d'un tel appel est de sauvegarder l'état de l'activité et les différents traitements effectués par l'utilisateur. A ce stade, notre activité n'a plus accès à l'écran, nous devons arrêter de faire toute action en rapport avec l'interaction utilisateur (désabonner les listeners).

La méthode `onStop ()` est appelée quand l'activité n'est plus visible quel que soit la raison. Dans cette méthode nous devons arrêter tous les traitements et services exécutés par notre application. La méthode `onDestroy ()` est appelée quand notre application est totalement fermée (Processus terminé). Les données non sauvegardées sont perdues [21].

III.7. Création d'un projet Android

Après l'installation du logiciel d'Android Studio, une fenêtre s'ouvre comme la montre la figure III.7, alors comme une première étape on va sélectionner « Start a new Android Studio Project » pour ouvrir un nouveau projet.

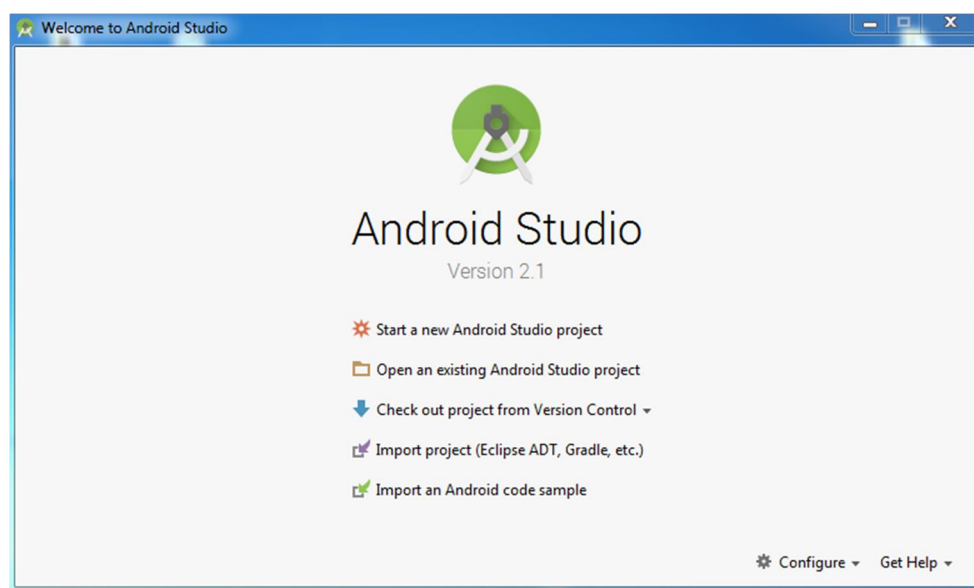


Figure III.7 : Création d'un nouveau projet.

Ensuite, après avoir choisi de créer un nouveau projet, une nouvelle fenêtre (figure III.8) s'ouvre pour remplir les informations de notre nouvelle application qui sont :

- **Application Name** : c'est le nom qui va apparaître dans la liste des applications sur l'appareil et dans le Play Store.
- **Company Domain** : c'est un qualifiant qui apparaîtra dans le nom du package.
- **Package Name** : il est utilisé comme identifiant de l'application.

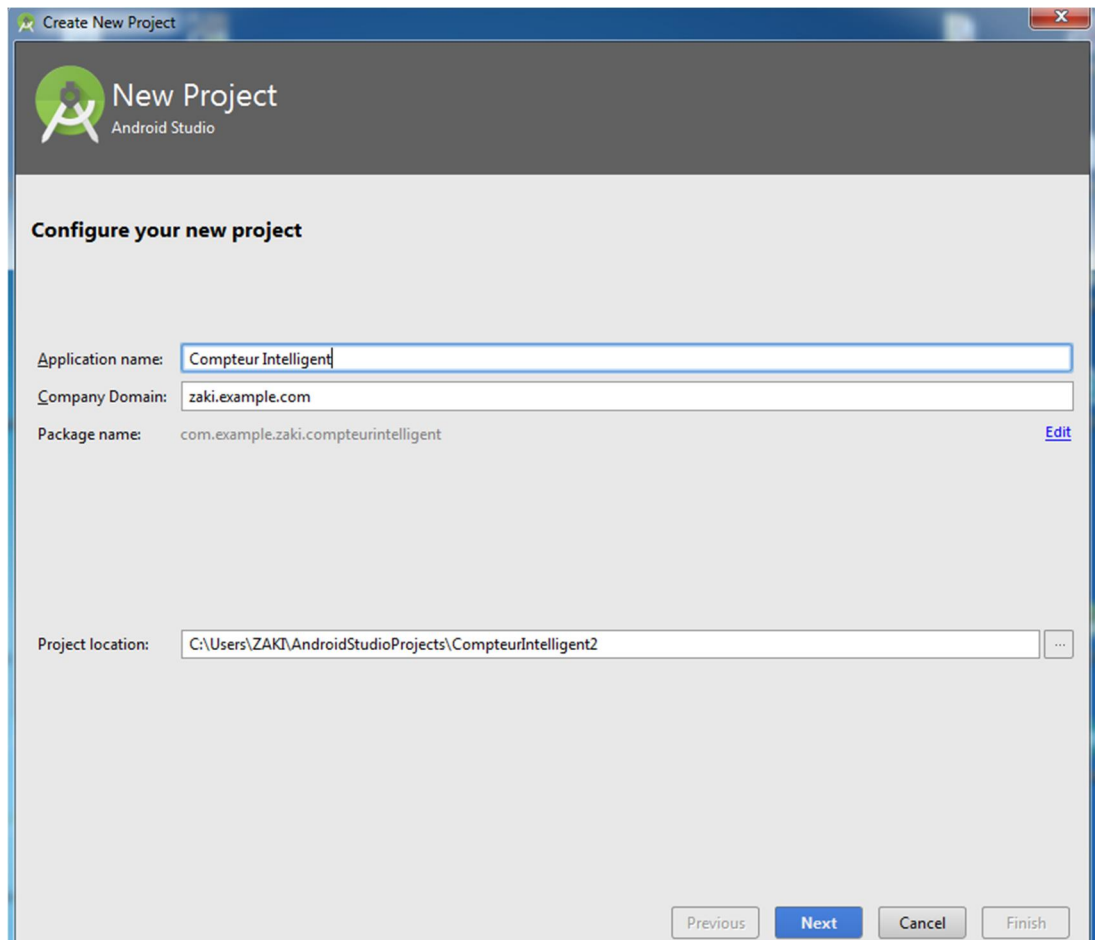


Figure III.8 : Configuration du projet.

Après cette configuration on va cliquer sur «Next» pour passer à une autre fenêtre comme la montre la (figure III.9) afin de faire le choix des cibles et terminer la configuration de l'application en cochant la case « Minimum required SDK » qui définit la version Android la plus ancienne sur laquelle l'application peut tourner.

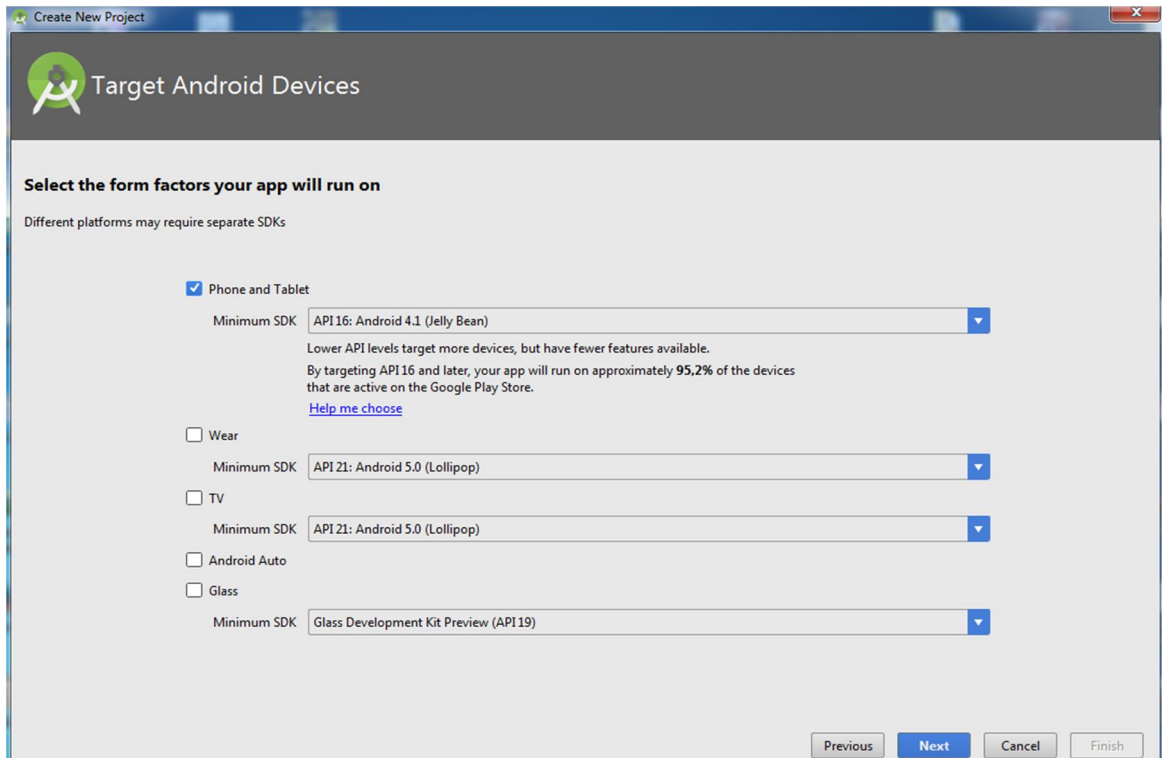


Figure III.9 : Choix des cibles.

On Cliquant sur « Next », on y arrive à la création d'une activité (un écran avec une interface graphique), on doit sélectionner « Empty Activity » (figure III.10) et puis on clique sur « Next » afin de parvenir à notre interface pour accélérer notre application.

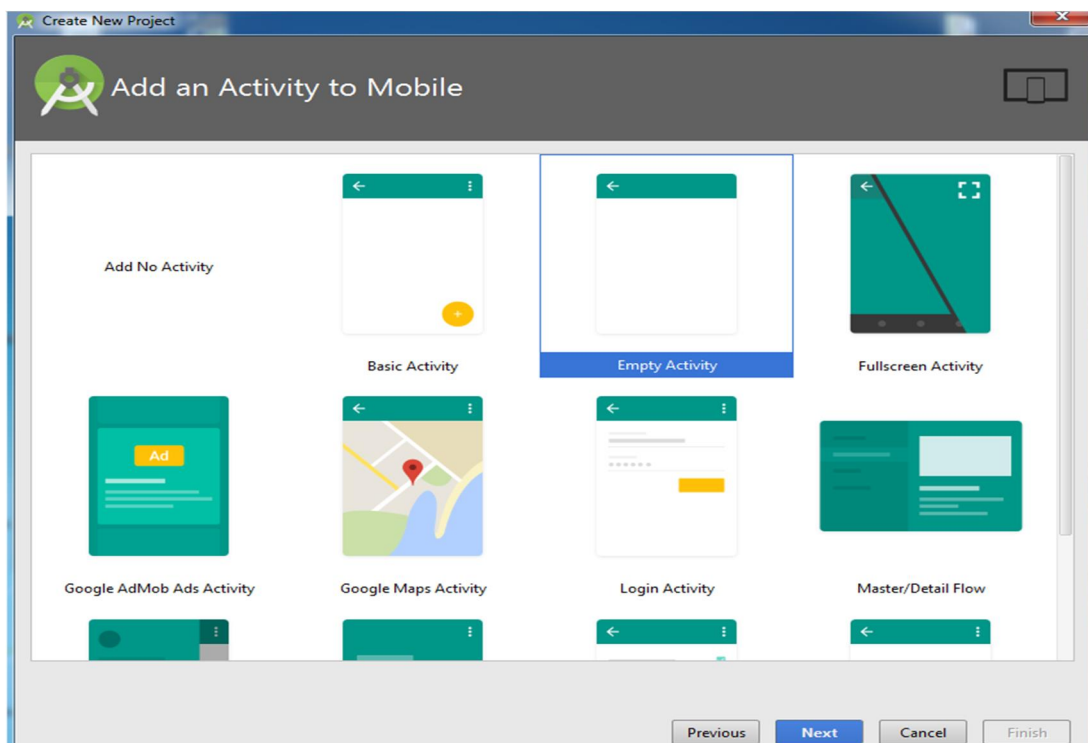


Figure III.10 : Choix de l'activité du projet.

III.8. Conception de l'application

Dans ce qui suit on va se focaliser sur l'application qu'on a conçue pour visualiser la consommation de notre compteur intelligent via la transmission Bluetooth, et puis la stocker dans la mémoire (interne ou sur une carte SD).

Sa structure est composée de 5 activités classées par ordre de priorité :

- Login pour la sécurité.
- Interface de la permission de connexion du Bluetooth.
- Interface d'affichage des périphériques.
- Interface de réception des données.
- Espace de stockage.

Pour notre application on a choisi de la personnaliser en faisant un logo pour la décrire en extérieure comme le montre la figure ci-dessous.



Figure III.11 : Logo de l'application du compteur intelligent.

L'interface principale de notre application est présentée en Figure(III.12). Pour sécuriser notre application nous avons ajouté un champ pour le nom d'utilisateur et autre pour le mot de passe pour que cette application ne soit utilisée que par l'agent releveur.

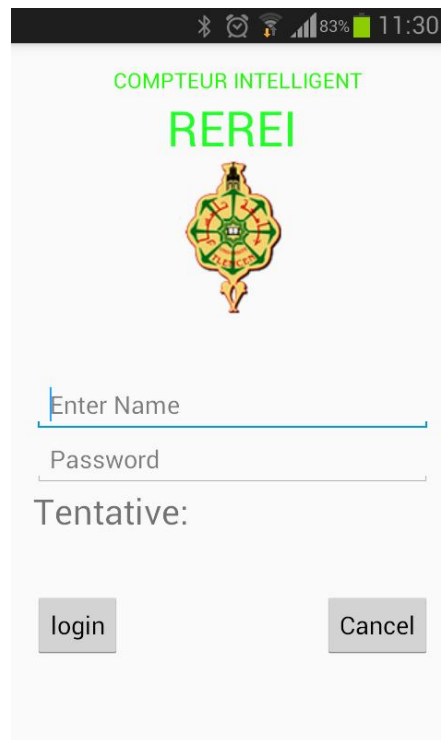


Figure III.12 : Interface Login.

Dans cette partie nous expliquons les APIs Android utilisées dans notre application concernant la manipulation du périphérique Bluetooth. Nous allons donc voir ce qu'il faut afin de faire la création d'une liaison Bluetooth. Pour bien débuter, assurons-nous d'avoir le droit de l'utiliser et que le terminal qui exécute notre application possède un périphérique Bluetooth, pour cela on a besoin de la permission suivante dans notre Manifest :

```
<uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="17" />
<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
```

Et pour Vérifier la présence du Bluetooth on a utilisé la classe BluetoothAdapter par l'instruction suivante :

```
};

btAdapter = BluetoothAdapter.getDefaultAdapter(); // get Bluetooth adapter
checkBTState();
```

Cette classe nous permet d'exécuter les fonctionnalités basiques, comme la vérification de la présence du Bluetooth. Voilà un petit bout de code pour vérifier cette présence :

```
private void checkBTState() {  
    // Check device has Bluetooth and that it is turned on  
    mBtAdapter=BluetoothAdapter.getDefaultAdapter(); // CHECK THIS OUT THAT IT WORKS!!!  
    if(mBtAdapter==null) {  
        Toast.makeText(getApplicationContext(), "Device does not support Bluetooth", Toast.LENGTH_SHORT).show();  
    } else {  
        if (mBtAdapter.isEnabled()) {  
            Log.d(TAG, "...Bluetooth ON...");  
        }  
    }  
}
```

Si la valeur est nulle le périphérique ne possède pas la technologie Bluetooth sinon il possède du Bluetooth dans les cas d'une valeur qui n'est pas nulle.

Après la vérification de la présence du Bluetooth, on passe à l'étape suivante qui est l'activation du Bluetooth, pour cela on doit faire la demande à l'utilisateur par l'instruction :

```
//Prompt user to turn on Bluetooth  
Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);  
startActivityForResult(enableBtIntent, 1);
```

Cette dernière instruction sert à vérifier si le Bluetooth est déjà activé, sinon il lance une boîte de dialogue pour demander à l'utilisateur d'activer le Bluetooth (Figure III.13).

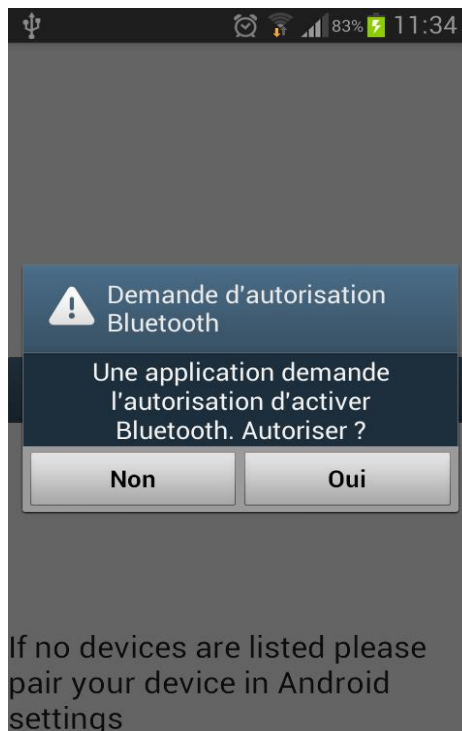


Figure III.13 : Interface d'activation du Bluetooth.

Tout juste après on va obtenir la liste des périphériques appareillés et qui sont déjà stocker pour une utilisation ultérieure ou pour vérifier si un de ces périphériques est connecté on utilisant la boucle suivante :

```
Set<BluetoothDevice> pairedDevices = mBtAdapter.getBondedDevices();

// Add previously paired devices to the array
if (pairedDevices.size() > 0) {
    findViewById(R.id.title_paired_devices).setVisibility(View.VISIBLE); //make title visible
    for (BluetoothDevice device : pairedDevices) {
        mPairedDevicesArrayAdapter.add(device.getName() + "\n" + device.getAddress());
    }
}
```

Cette dernière instruction nous permet d'afficher l'interface de la figure (III.14).

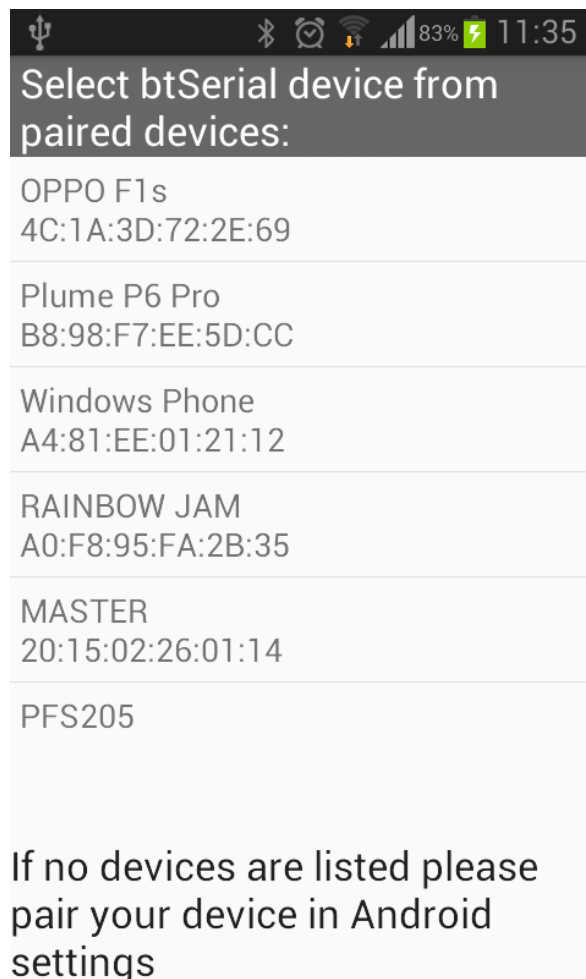


Figure III.14 : Interface de la liste des périphériques disponible.

Cette dernière étape nous permet de sélectionner le périphérique désiré pour l'échange des données, dans notre cas le module Bluetooth HC-05 relié à notre carte électronique du compteur intelligent qui permet d'établir une connexion et échanger les données, du côté serveur on a un dispositif qui reçoit la connexion (Smartphone) et du côté client on a le dispositif qui envoie la connexion (compteur).

La procédure à suivre est la suivante. Nous allons récupérer une « btSocket » en appelant « socket.getInputStream ». Puis nous allons attendre une connexion par le biais de la méthode « accept () » du socket. Une fois le client est connecté, on bloque les demandes arrivant avec un appel à « close () ». Ensuite, du côté client Nous allons recevoir une « btSocket » en appelant « socket.getOutputStream » et pour lancer la connexion on fait appel à « connect () » [24]. La figure suivante nous montre une petite partie du code :

```

    btSocket = createBluetoothSocket(device);
} catch (IOException e) {
    Toast.makeText(getApplicationContext(), "Socket creation failed", Toast.LENGTH_LONG).show();
}
// Establish the Bluetooth socket connection.
try
{
    btSocket.connect();
} catch (IOException e) {
    try
    {
        btSocket.close();
    } catch (IOException e2)
    {
        //insert code to deal with this
    }
}
mConnectedThread = new ConnectedThread(btSocket);
mConnectedThread.start();

```

Figure III.15 : Code de l'échange des données.

Enfin, pour relier entre ces activités et ces interfaces on a utilisé le « intent » comme le montre le code ci-dessous :

```

// Make an intent to start next activity while taking an extra which is the MAC address.
Intent i = new Intent(DeviceListActivity.this, MainActivity.class);
i.putExtra(EXTRA_DEVICE_ADDRESS, address);
startActivity(i);
;

```

Après l'assemblage de toutes ces activités que nous avons fait et la création des interfaces de notre projet, nous allons récupérer les données affichées sur le LCD, et les affichées sur l'écran du Smartphone par le bouton « Recevoir » pour recevoir et « fin envoi » pour mettre fin à l'envoi des données depuis la carte Arduino, ces données reçus sont par la suite stocker sur Smartphone (figure III.16).



Figure III.16: Interface de réception de données (a) avant réception, (b) après réception).

La partie ci-dessous est consacrée à la création d'une base de données destinée au stockage de données issues de notre application d'acquisition. Cette base de données est constituée de deux principaux paramètres l'index et la valeur de consommation de l'abonné. Pour exécuté ces paramètres on a créé deux boutons, « Enregistrer » pour sauvegarder les données reçus et le bouton « Voir » pour visualisé les données sauvegardées.

Voici la zone de traitement de données (Figure III.17)

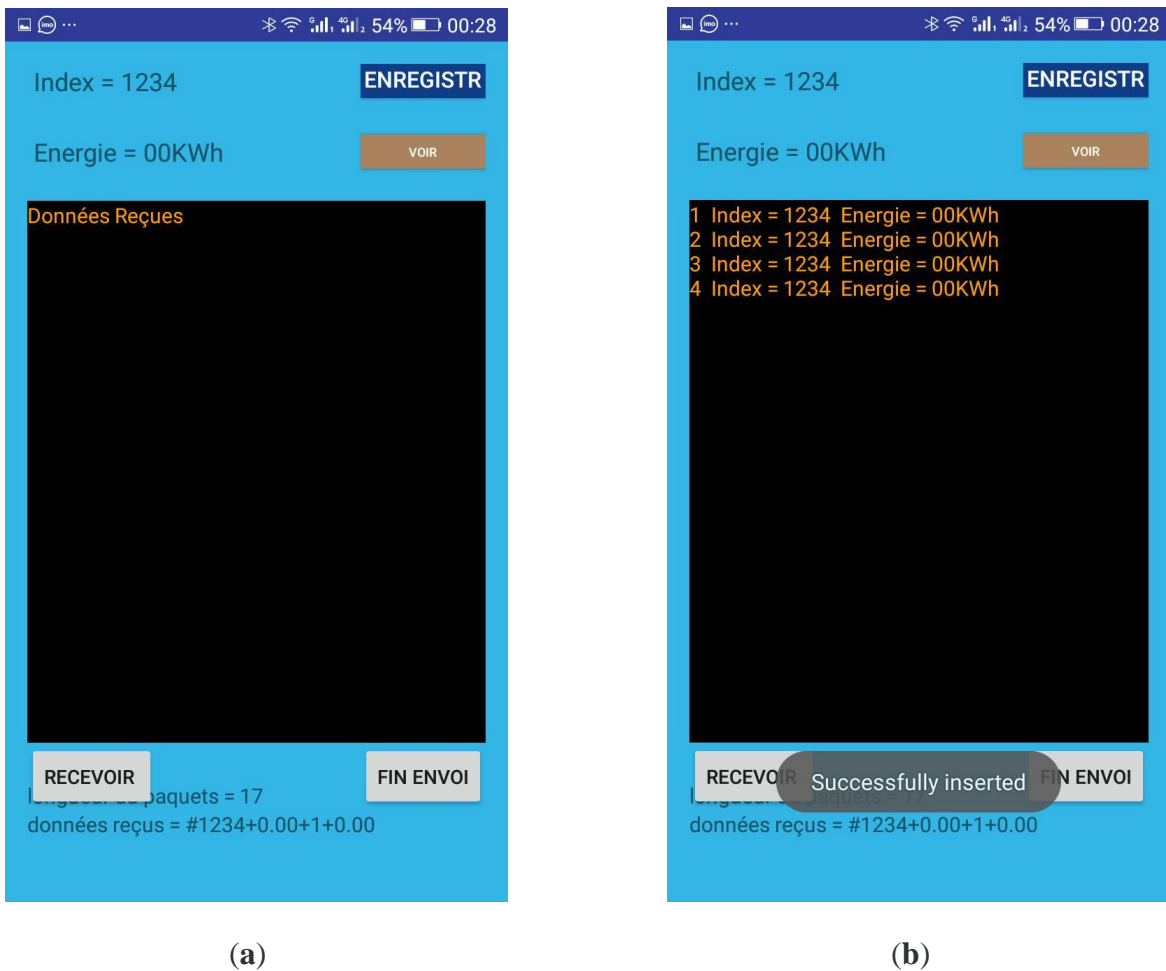


Figure III.17 : Interface de traitement de données (a sauvegarde, b visualisations après l'enregistrement).

La première case est réservée à la sauvegarde de l'index de l'abonné. Le programme en XML est représenté par la suite :

```
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:id="@+id/Index"
    android:width="220dp"
    android:background="#b6def1"
    android:textAlignment="center"
    android:layout_below="@+id/textView"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:layout_marginTop="12dp" />
```


Par la programmation Java en tant que « sensorView0 » :

```
public void save(View view) {  
  
    String full_name = sensorView0.getText().toString();  
    String user_name = sensorView3.getText().toString();  
  
}
```

La deuxième case est réservée à la consommation d'énergie de l'abonné, son programme en XML est représenté par la suite :

```
) <TextView  
    android:id="@+id/sensorView3"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Energie :"  
    android:textAppearance="?android:attr/textAppearanceMedium"  
    android:layout_below="@+id/sensorView0"  
    android:layout_alignParentLeft="true"  
    android:layout_alignParentStart="true"  
    android:layout_marginTop="26dp" />  
)
```

Par la programmation Java en tant que « sensorView3 » :

```
public void save(View view) {  
  
    String full_name = sensorView0.getText().toString();  
    String user_name = sensorView3.getText().toString();  
  
}
```

Par la suite on va présenter les boutons déjà cités concernant la sauvegarde et l'affichage des données reçus. Pour les boutons « Enregistrer » et « Voir » représentés par la programmation XML suivante :

```

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Enregistrer"
    android:id="@+id/button"
    android:width="120dp"
    android:textSize="16dp"
    android:onClick="save"
    android:background="#0f3d88"
    android:textColor="#ffffff"
    android:height="9dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentTop="true"
    android:layout_alignLeft="@+id/buttonOff"
    android:layout_alignStart="@+id/buttonOff"
    android:layout_above="@+id/sensorView3" />

```

```

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Voir"
    android:id="@+id/button2"
    android:background="#a9825e"
    android:width="120dp"
    android:textAlignment="center"
    android:textColor="#ffffff"
    android:textSize="9dp"
    android:onClick="dataView"
    android:layout_above="@+id/textView5"
    android:layout_alignRight="@+id/button"
    android:layout_alignEnd="@+id/button"
    android:layout_alignTop="@+id/sensorView3"
    android:layout_alignLeft="@+id/button"
    android:layout_alignStart="@+id/button" />

```

Concernant l'espace de stockage, on a consacré un espace dans la base de données de l'application par la programmation XML suivante :

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="120dp"
    android:text="Données Reçues"
    android:id="@+id/textView5"
    android:background="#000000"
    android:textColor="#ff9d00"
    android:textSize="15dp"
    android:scrollbars="vertical"
    android:verticalScrollbarPosition="right"
    android:visibility="visible"
    android:scrollbarAlwaysDrawVerticalTrack="false"
    tools:scrollbarAlwaysDrawVerticalTrack="false"
    android:isScrollContainer="true"
    android:layout_below="@+id/sensorView3"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginTop="23dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:layout_above="@+id/buttonOn" />

```

Et par la programmation Java :

```
public void save(View view) {  
  
    String full_name = sensorView0.getText().toString();  
    String user_name = sensorView3.getText().toString();  
  
    long id = DataBase.dataInsert(full_name, user_name);  
  
    if (id < 0) {  
        Toast.makeText(this, "Error ! not inserted", Toast.LENGTH_LONG).show();  
    } else {  
        Toast.makeText(this, "Successfully inserted", Toast.LENGTH_LONG).show();  
    }  
}  
  
public void dataView(View view) {  
  
    String data = DataBase.viewData();  
  
    textView.setText(data);  
}
```

Conclusion

Dans ce dernier chapitre nous avons présenté les outils de conception et développement sous Android.

Avec l'architecture de la plateforme, les composants d'un projet en cours de conception, le cycle de vie associé avec les activités du projet ainsi les bibliothèques Android qui permettent d'exploiter ces interactions qui nous poussent à réaliser une application « *.apk » sous Smartphone capable de synchroniser les données entre une carte Arduino et un Smartphone afin d'exécuter des ordres bien définis.

Conclusion générale

Conclusion générale

Les compteurs électriques intelligents sont l'un des équipements les plus importants et l'un des leviers de la transformation des réseaux de distribution électrique en réseaux intelligents, ou Smart Grid. Ces compteurs sont très récents et en cours d'intégration dans certains pays.

L'objectif de ce mémoire n'est pas seulement l'étude et la réalisation d'un système de communication pour un compteur intelligent mais aussi la création d'un outil, simple d'utilisation, pas coûteux, et qui traduit différentes fonctionnalités comme le suivi en temps réel et la gestion à distance du compteur électrique.

Dans notre projet, nous nous sommes principalement intéressés à l'étude et la réalisation d'un compteur intelligent ou Smart meter. Les données sont obtenues à l'aide d'un capteur de courant et un autre de tension, ces données sont traitées par un Arduino méga et enfin transmises vers un Smartphone par un module Bluetooth.

En deuxième partie nous avons réalisé une application Android destinée à faire la relève à distance de la puissance et la consommation d'énergie d'une maison en plus de la référence du compteur installé et sauvegarder ces données dans la mémoire interne du Smartphone.

Ce système réalisé facilite la tâche au releveur et évite la saisie manuelle de la valeur de la consommation au niveau de la maison ainsi que pour l'intégration des données au logiciel de facturation au niveau de l'entreprise.

Les perspectives que nous envisageons sont :

- ✓ L'amélioration du circuit de mesure pour les systèmes triphasés
- ✓ Miniaturisation du circuit
- ✓ Le suivi en temps réel de la consommation

Références bibliographiques

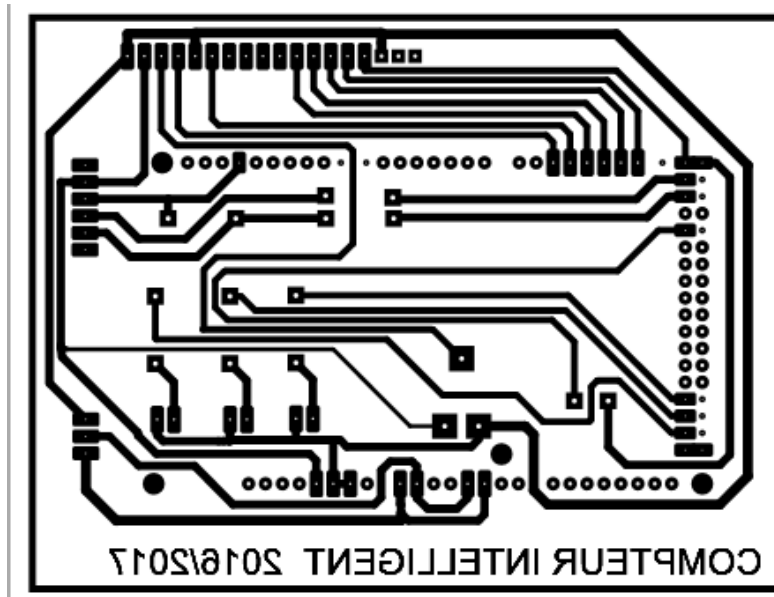
Références Bibliographique

- [1] B. Baghdâd, La communication sans fil dans un réseau électrique Intelligent (smart Grid) - méthodologie de développement, mémoire Présenté Comme exigence partielle De la maîtrise en informa tique, Université du Québec à Montréal, Avril 2013 [en ligne]. Disponible sur : www.archipel.uqam.ca/5680/1/M12895.pdf. [Consulté le 12 février 2017]
- [2] C. Bissieres, « Distribution électrique et sécurité,» [En ligne]. Disponible sur : [http://1cbissprof.free.fr/téléchargements/tsiris/cours/distribution electrique.pdf](http://1cbissprof.free.fr/téléchargements/tsiris/cours/distribution%20electrique.pdf). [Consulté le 12 février 2017]
- [3] JAY. Compteur intelligent : de quoi parle-t-on ? [En ligne]. Disponible sur : <https://electrosensible.info/2013/04/22/compteur-intelligent-de-quoi-parle-t-on/> . [Consulté le 12 février 2017]
- [4] F. KLOPFERT, Lø rapport des compteurs intelligents à une consommation plus durable de l'lectricité, Université Libre de Bruxelles Institut de Gestion de Løenvironnement et Aménagement du Territoire [En ligne]. Disponible sur : mem-envi.ulb.ac.be/Memoires_en_pdf/MFE_07_08/MFE_Klopfert_07_08.pdf. [Consulté le 15 février 2017]
- [5] Les compteurs évolués [En ligne]. Disponible sur : <http://www.smartgrids-cre.fr/index.php?p=compteurs-caracteristiques> [consulté le 18 février 2017]
- [6] Réseau intelligent (Smart Grid) [En ligne]. Disponible sur : <http://www.connaissancedesenergies.org/fiche-pedagogique/reseau-intelligent-smart-grid> [consulté le 18 février 2017]
- [7] R. BOUGHRIET. Smart Grid [En ligne]. Disponible sur : <http://www.actu-environnement.com/ae/dossiers/smart-grid/smart-grid-attendu.php4>. [Consulté le 19 février 2017]
- [8] HOUTI Zahra Souhila, M.ABERKANE Yasmina, 2012, Réseau intelligent de distribution d'lectricité (smart Grid), Master en Telecom, UNIVERSITE ABOU BEKR BELKAID TLEMCEN
- [9] Journal « ECHOS Groupe Sonelgaz » édition Juin-juillet 2011 [En ligne]. Disponible sur : www.sonelgaz.dz/Media/upload/rapport%20activites%20sonelgaz2012.pdf [Consulté le 21-02-2017]

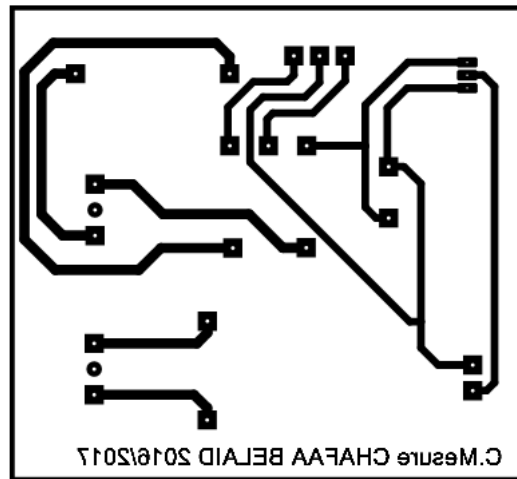
- [10] M.CHABANE. Projet Compteurs Intelligents « ZACKØS » [En ligne]. Disponible sur : <https://portail.cder.dz/spip.php?article2140> [Consulté le 21-02-2017]
- [11] http://sti.tice.ac-orleans-tours.fr/spip2/IMG/pdf/Les_capteurs.pdf (consulté le 23-04-2017) 11
- [12] <http://physiquemangin.pagesperso-orange.fr/BTSSE/cours/capteursetudiant.pdf> (consulté le 25-04-2017)
- [13]<http://www.technologuepro.com/cours-capteurs-actionneurs-instrumentation-industrielle/ch12-les-differents-types-de-capteurs.pdf> [consulter le 05-05-2017]
- [14] Dossier Pédagogique. Capteurs de mesure de courant. Cours - TD. GE - Ts1 elec. Lycée Technique E. Branly
- [15] Arduino-Easycon-Organigram, Cahier 0 connaître le matériel et le préparer, technologies tutoriel, académie Nantes
- [16]http://www.monclubelec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.MaterielMega2560 [consulter le 06-05-2017]
- [17] E.BARTMANN, D.LAFARGE, P.CHANTEREAU, 2015, Le grand livre d'Arduino-Eyrolles
- [18]http://www.mon-clubelec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.DebuterPresentationLogiciel [consulter le 06-05-2017]
- [19] KRAMA Abdel basset, GOUGUI Abdelmoumen, le : 08/06/2015, Etude et réalisation d'une carte de contrôle par Arduino via le système Androïde, Master en Electrotechnique Industrielle, UNIVERSITE KASDI MERBAH OUARGLA
- [20]: Gartner Inc Gartner Says Smartphone Sales Grew 46.5 Percent in Second Quarter of 2016 and Eceeded Featur Phone Sales for First Time Gartner. [En ligne]. Disponible sur : <http://www.gartner.com/newsroom/id/2573415>. [Consulté le 25-05-2017]
- [21]: Mark Murphy Beginning Android 3.sl: Apress 2011.
- [22]: <http://www.tutos-android.com/introduction-a-android-studio> [Consulté le 25-05-2017]
- [23] <http://www.tutos-android.com/utilisation-bluetooth-application-android> [Consulté le 25-05-2017]

ClicCours.com

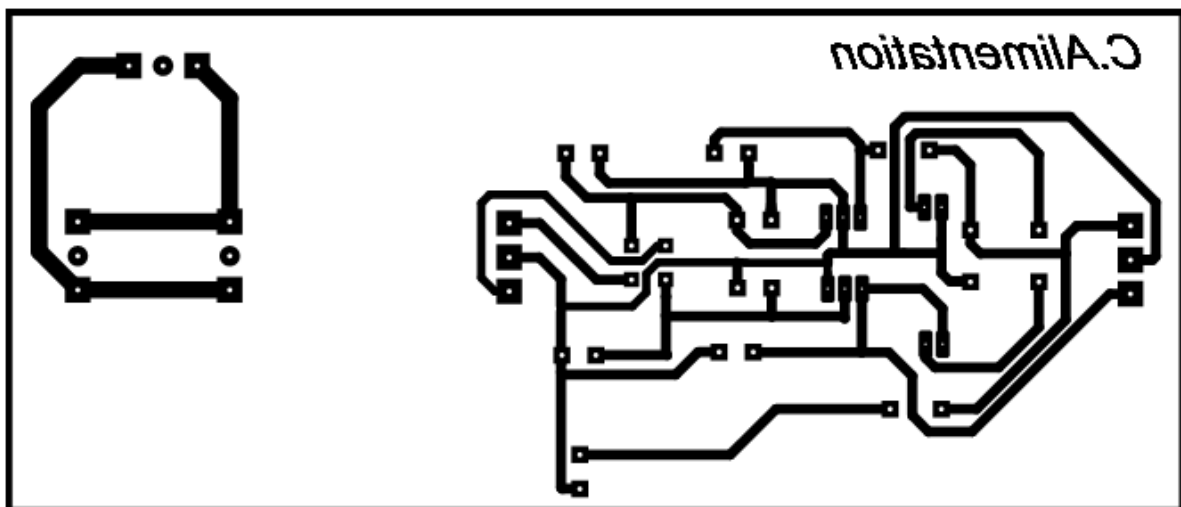
Annexe A
Circuits Imprimés



Circuit de mesure et d'affichage en Shield pour Arduino Mega



Circuit de mesure (capteurs de courant et de tension)



Circuit d'alimentation

Annexe B

Datasheet

Product Name	Hall voltage sensor	Model	HV25-1000
--------------	---------------------	-------	-----------

Characteristics:HV25 and Peripheral electronic components compose a sensor plate, terminal input directly to test voltage,output signal 5V/25mA. For the electronic measurement of voltage: DC,AC,pulsed....

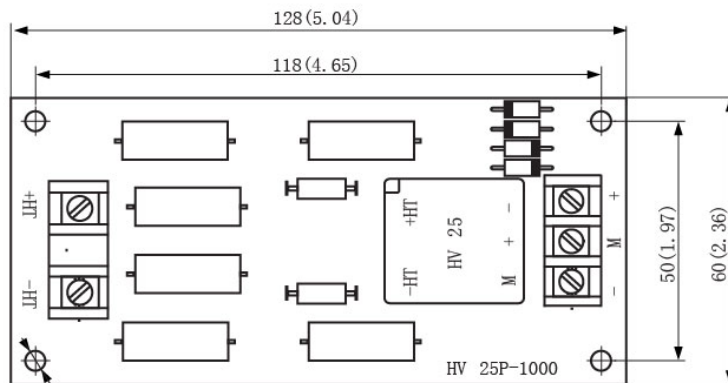
Technical Data

I_{PN}	Rated Input (Effective value)	1000V
I_{PM}	Measuring range	$\pm 1500V$
V_{OUT}	Rated output	5V
X	Accuracy	$\pm 1\%$
ϵ_L	Linearity	$< 0.1\%$
V_C	Supply voltage($\pm 5\%$)	$\pm 15V$
I_C	Current consumption	$< 35mA$
R_L	Measured impedance	$> 10K\Omega$
V_{OE}	Zero offset, $T_A=25^\circ C$	$\pm 30mV$
t_r	Response time	40 μs
BW	Bandwidth	-
T_A	Operating temperature	-10..+80 $^\circ C$
T_S	Storage temperature	-40..+85 $^\circ C$
V_d	Dielectric strength, 50 Hz, 1 min	2.5kV



Fire resistance	UL94-V0
Mounting type	Plate-type
Standards	EN50178
Weight	

Dimension (mm(in). 1 mm= 0.0394 inch)



4 \times $\phi 4.2(0.17)$ Fixed hole Hall voltage sensor board view (Just for reference)

LM79XX Series 3-Terminal Negative Regulators

General Description

The LM79XX series of 3-terminal regulators is available with fixed output voltages of $-5V$, $-8V$, $-12V$, and $-15V$. These devices need only one external component—a compensation capacitor at the output. The LM79XX series is packaged in the TO-220 power package and is capable of supplying 1.5A of output current.

These regulators employ internal current limiting, safe area protection and thermal shutdown for protection against virtually all overload conditions.

Low ground pin current of the LM79XX series allows output voltage to be easily boosted above the preset value with a resistor divider. The low quiescent current drain of

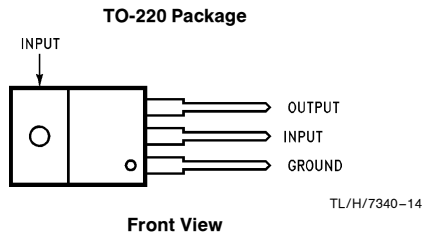
these devices with a specified maximum change with line and load ensures good regulation in the voltage boosted mode.

For applications requiring other voltages, see LM137 data sheet.

Features

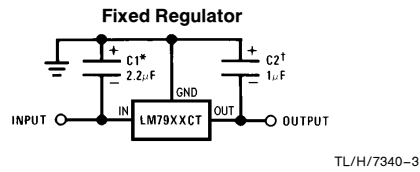
- Thermal, short circuit and safe area protection
- High ripple rejection
- 1.5A output current
- 4% tolerance on preset output voltage

Connection Diagrams



Order Number **LM7905CT**, **LM7912CT** or **LM7915CT**
See NS Package Number **TO3B**

Typical Applications



*Required if regulator is separated from filter capacitor by more than 3". For value given, capacitor must be solid tantalum. 25 μF aluminum electrolytic may be substituted.

†Required for stability. For value given, capacitor must be solid tantalum. 25 μF aluminum electrolytic may be substituted. Values given may be increased without limit.

For output capacitance in excess of 100 μF , a high current diode from input to output (1N4001, etc.) will protect the regulator from momentary input shorts.

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

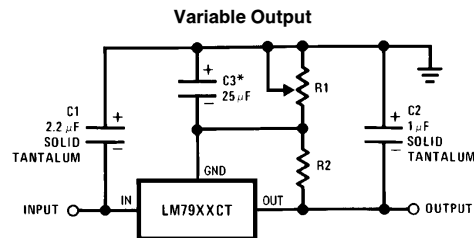
Input Voltage
 $(V_o = -5V)$ -25V
 $(V_o = -12V \text{ and } -15V)$ -35V

Input-Output Differential
 $(V_o = -5V)$ 25V
 $(V_o = -12V \text{ and } -15V)$ 30V
 Power Dissipation (Note 2) Internally Limited
 Operating Junction Temperature Range 0°C to $+125^\circ\text{C}$
 Storage Temperature Range -65°C to $+150^\circ\text{C}$
 Lead Temperature (Soldering, 10 sec.) 230°C

Electrical Characteristics Conditions unless otherwise noted: $I_{OUT} = 500 \text{ mA}$, $C_{IN} = 2.2 \mu\text{F}$, $C_{OUT} = 1 \mu\text{F}$, $0^\circ\text{C} \leq T_J \leq +125^\circ\text{C}$, Power Dissipation $\leq 1.5\text{W}$.

Part Number			LM7905C			Units		
Output Voltage			-5V					
Input Voltage (unless otherwise specified)			-10V					
Symbol	Parameter	Conditions	Min	Typ	Max			
V_O	Output Voltage	$T_J = 25^\circ\text{C}$ $5 \text{ mA} \leq I_{OUT} \leq 1 \text{ A}$, $P \leq 15\text{W}$	-4.8	-5.0	-5.2	V		
			-4.75		-5.25	V		
			$(-20 \leq V_{IN} \leq -7)$					V
ΔV_O	Line Regulation	$T_J = 25^\circ\text{C}$, (Note 3)		8	50	mV		
			$(-25 \leq V_{IN} \leq -7)$					V
				2	15	mV		
$(-12 \leq V_{IN} \leq -8)$					V			
ΔV_O	Load Regulation	$T_J = 25^\circ\text{C}$, (Note 3) $5 \text{ mA} \leq I_{OUT} \leq 1.5 \text{ A}$ $250 \text{ mA} \leq I_{OUT} \leq 750 \text{ mA}$		15	100	mV		
				5	50	mV		
I_Q	Quiescent Current	$T_J = 25^\circ\text{C}$		1	2	mA		
ΔI_Q	Quiescent Current Change	With Line			0.5	mA		
		$(-25 \leq V_{IN} \leq -7)$					V	
V_n	Output Noise Voltage	$T_A = 25^\circ\text{C}$, $10 \text{ Hz} \leq f \leq 100 \text{ Hz}$		125		μV		
			Ripple Rejection	$f = 120 \text{ Hz}$	54	66	dB	
			$(-18 \leq V_{IN} \leq -8)$			V		
	Dropout Voltage	$T_J = 25^\circ\text{C}$, $I_{OUT} = 1 \text{ A}$		1.1		V		
I_{OMAX}	Peak Output Current	$T_J = 25^\circ\text{C}$		2.2		A		
	Average Temperature Coefficient of Output Voltage	$I_{OUT} = 5 \text{ mA}$, $0^\circ\text{C} \leq T_J \leq 100^\circ\text{C}$		0.4		$\text{mV}/^\circ\text{C}$		

Typical Applications (Continued)



*Improves transient response and ripple rejection. Do not increase beyond 50 μF .

TL/H/7340-2

$$V_{OUT} = V_{SET} \left(\frac{R1 + R2}{R2} \right)$$

Select R2 as follows:
 LM7905CT 300 Ω
 LM7912CT 750 Ω
 LM7915CT 1k

Résumé

L'avancement des technologies à aider l'amélioration des systèmes du réseau électrique par l'intégration des énergies renouvelables ainsi que le développement des compteurs qui sont la base des réseaux intelligents (Smart Grid). L'intérêt principal de ces compteurs est de faire le suivi des données en temps réel.

Notre but est d'étudier et réaliser un compteur dit intelligent. La carte électronique réalisé est piloté par un microcontrôleur ATmega 2560 d'un Arduino, cette dernière permet aussi la transmission des données vers un Smartphone via un système de communication sans fil de type Bluetooth. Les données reçus sont par la suite visualisé à l'aide d'une application réalisée sous Android studio et enfin enregistrer sur une carte SD pour une utilisation ultérieur.

Mot clés : Réseaux intelligent, compteur intelligent, Arduino, Bluetooth, Android.

Abstract

The advancement of technologies has helped to improve the electricity grid systems by integrating renewable energies and developing the meters that are the basis of smart grids. The main interest of these meters is to monitorate the data in real time.

Our purpose is to study and realize a device that is a so-called smart meter .The electronic board is controlled by an ATmega 2560 microcontroller from an Arduino The latter also allows the transmission of data to a Smartphone via a wireless communication system defined Bluetooth. The received data is then viewed using an application built under Android studio and finally saved on an SD card for later use.

Keywords : Smart Grid, smart meter, Arduino, Bluetooth, Android.

ملخص

التقدم في التكنولوجيات ساعد في تحسين أنظمة الشبكة الكهربائية عبر إدماج الطاقات المتجددة و تطوير العدادات الإلكترونية التي تعتبر أساس الشبكة الكهربائية الذكية. تكمن الفائدة الرئيسية لهذه العدادات في القيام بالمتابعة و المراقبة الحينية للمعلومات .

هدفنا في هذه المذكرة يتضمن دراسة و إنشاء عداد ذكي. هذا الجهاز متحكم فيه عن طريق معالج من نوع ATMEGA 2560 لبطاقة أرد وينو ، هذه الأخيرة تسمح بإرسال المعلومات نحو الهاتف الذكي بطريقة لا سلكية باستخدام تقنية البلوتوث، تعرض هذه المعلومات على شاشة الهاتف عبر برمجة تطبيق أندرويد و بالتالي حفظ المعلومات على بطاقة ذاكرة إلكترونية لاستعمال لاحق.

الكلمات المفتاحية : الشبكة الذكية, عداد ذكي, أردوينو, بلوتوث, أندرويد.