

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد - تلمسان

Université Aboubakr Belkaïd – Tlemcen –

Faculté de TECHNOLOGIE



## MEMOIRE

Présenté pour l'obtention du **diplôme de MASTER**

En : Electrotechnique

**Spécialité** : Commandes Electriques

Par : LAZAAR Fatima et KERMAL Asmae

# Amélioration d'un prototype de compteur intelligent avec intégration de système de communication

Soutenu publiquement, le 26/ 06/ 2019, devant le jury composé de :

Mr M. LOUCIF	Maitre de Conférences	Univ. Tlemcen	Président
Mr M. C. BENHABIB	Maitre de Conférences	Univ. Tlemcen	Examineur
Mr M. N. MOULAI KHATIR	Maitre de Conférences	Univ. Tlemcen	Examineur
Mr S. M. MELIANI	Maitre de Conférences	Univ. Tlemcen	Encadrant

**Année Universitaire : 2018-2019**

# Remerciements

---

Nous souhaitons en premier lieu remercier ALLAH qui nous a aidé à accomplir ce modeste travail et qui nous a donné toute la patience et la volonté pour surpasser tous les moments difficiles durant la réalisation de ce projet.

En second lieu, nous tenons à exprimer notre gratitude à Monsieur S. M. MELIANI pour avoir dirigé ce travail, pour son aide et ses précieux conseils ainsi que pour les nombreuses discussions que nous avons eu et sans lui ce mémoire n'aurait jamais pu être terminé.

Nous tenons également à adresser nos remerciements aux membres du jury Messieurs M. C. BENHABIB, M. LOUCIF, M.N. MOULAI KHATIR pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Enfin, merci à toutes les personnes qui ont participé de près ou de loin à la réussite de ce travail.

KERMAL Asmae    LAZAAR Fatima

# Dédicace

---

*Que ce travail témoigne de mes respects :*

*À mon père,*

Qui a travaillé dur pour moi et m'a appris le sens de la lutte pour atteindre les objectifs.

*À ma mère*

La personne la plus chère à mon cœur, qui a toujours prié pour ma réussite.

*À mes frères*

Ahmed, Lakhdar, Mohamed, Mourad, Nassim

*À tous les membres de ma famille*

Tantes oncles cousins et cousines, qui m'ont soutenue de loin et de près

*À mon fiancé et toute sa famille*

*À tous mes chers Amis(es),*

H .Hanan, B.Abir , B.Dounai, B.Noura, B.Rokia

Sans oublier mon binôme K.Asmæ et toute sa famille.

A ma promo « Commande électrique »

....

Et tant d'autres,

Merci pour tous les bons moments passés ensemble.

« LAZAAR FATIMA »

# Dédicace

---

*Que ce travail témoigne de mes respects :*

*À mon père,*

Qui a travaillé dur pour moi et m'a appris le sens de la lutte pour atteindre les objectifs.

*À ma mère*

La personne la plus chère à mon cœur, qui a toujours prié pour ma réussite.

*À mes frères et mes sœurs*

ABDELKADER, MILOUD, LATIFA et FATIMA

*À tous les membres de ma famille*

Tantes, oncles, cousins et cousines, qui m'ont soutenue de loin et de près

*À tous mes chers Amis(es),*

B.ROKIA, H.HANANE, B.ABIR, B.NOURA, K.NAWEL

Sans oublier mon binôme L.FATIMA et toute sa famille.

A ma promo « Commande électrique »

....

Et tant d'autres,

Merci pour tous les bons moments passés ensemble.

«KERMAL ASMAE»

# Table des matières

Table des matières .....	
Liste des figures .....	
Glossaire.....	
Introduction Générale.....	1
Chapitre 1 Généralité sur les compteurs électrique et les systèmes de communication utilisés .....	3
1. Introduction.....	3
2. Description d'un compteur intelligent .....	4
2.1. Définition d'un compteur intelligent.....	4
2.2. Fonctionnement du système.....	5
3. Système de communication .....	6
3.1. Définition.....	6
3.2. Meilleurs moyens de communication sans fil avec contrôleur.....	6
3.2.1. Wifi .....	6
3.2.2. Bluetooth / BLE4.0.....	7
3.2.3. GSM / GPS .....	7
3.2.4. Radio RF.....	7
4. Conclusion .....	8
Chapitre 2 Description Générale de notre prototype de compteur intelligent.....	9
1. Introduction.....	9
2. Description générale de notre prototype .....	10
3. Représentation détaillée de notre prototype .....	10
3.1. Circuit d'alimentation.....	11
3.1.1. Fonctionnement d'une alimentation à découpage.....	11
3.1.2. Caractéristiques d'une alimentation à découpage .....	12
3.1.3. Éléments constitutifs.....	12
3.1.4. Comparaison par rapport aux alimentations linéaires.....	13
3.2. Capteur .....	13
3.2.1. Capteur de tension .....	14

# Table des matières

3.2.2.	Capteur de courant .....	16
3.3.	Afficheurs LCD .....	17
3.4.	Carte de commande basée sur le microcontrôleur dsPIC33F .....	17
3.4.1.	Définition .....	17
3.4.2.	Organisation de la mémoire .....	18
3.4.3.	Principaux modules du dsPIC33F .....	18
4.	Moyen matériel et logiciel pour la programmation du dsPIC .....	24
4.1.	Programmeur PICkit2 .....	24
4.2.	Interface MPLAB .....	24
4.3.	Méthode de calcul de puissance .....	25
5.	Communication et transfert de données .....	27
5.1.	Introduction descriptive .....	27
5.2.	Module Bluetooth .....	28
5.2.1.	Définition .....	28
5.2.2.	Applications de Bluetooth .....	28
5.2.3.	Comment sélectionner le module Bluetooth? .....	28
5.2.4.	Différence entre HC-05 VS HC-06 .....	29
5.2.5.	Description du module Bluetooth utilisé HC-05 .....	29
5.2.6.	Communication série .....	32
5.3.	Test de fonctionnalité de notre système de communication .....	36
5.3.1.	Notion sur l'application PUTTY.exe .....	36
6.	Conclusion .....	40
Chapitre 3	Application mobile et réalisation finale de notre prototype .....	41
1.	Introduction .....	41
1.1.	Définition du système d'exploitation mobile 'Android' .....	41
1.2.	Avantages d'Android .....	41
2.	Environnement de développement MIT App Inventor .....	41
2.1.	Définition .....	41
2.2.	Préparation du matériel .....	41
2.3.	Création du projet d'une application mobile .....	42
2.4.	Développement d'application Android .....	43
2.4.1.	Conception d'Application AI2 .....	44
2.4.2.	Programmation de l'Application AI2 .....	50

## Table des matières

2.5. Test de l'application .....	52
2.6. Téléchargement de l'application mobile depuis App Inventor 2 .....	54
3. Réalisation finale de notre prototype .....	56
3.1. Organigramme.....	56
4. Conclusion .....	58
Bibliographie .....	60

# Liste des figures

Figure 1.1 : Compteur (classique ) monophasé heliowatt[4] .....	3
Figure 1.2 : Compteur électronique monophasé type « EDF » SAGEM. [4].....	3
Figure 1.3 : Compteur monophasé« modulaire » Polier.[4].....	3
Figure 1.4 : Compteur de consommation d'appareillage électrique.[4].....	3
Figure 1.5 : Compteur communiquant [4].....	3
Figure 1.6 : Compteur électrique communicant à un smart grid [1] .....	4
Figure 1.7 : Architecture de réseau basé sur des compteurs intelligents [5] .....	5
Figure 2.1 : Compteur électromécanique et compteur électrique [8].....	9
Figure 2.2 : Compteur communicant [8].....	9
Figure 2.3 : Schéma global de prototype de notre compteur intelligent.[09].....	10
Figure 2.4 : Différents circuits de prototype de notre compteur intelligent. ....	11
Figure 2.5 : Eléments principaux d'une alimentation à découpage.[10].....	12
Figure 2.6 : Circuit électronique pour créer une tension $\pm 12V$ . ....	13
Figure 2.7 : Circuit électronique de capteur de tension LV25-p.[09]. ....	15
Figure 2.8 : Signal de tension.....	15
Figure 2.9 : Capteur de courant ACS712 - 30A .....	16
Figure 2.10 : Signal de courant .....	16
Figure 2.11 :Afficheur LCD alphanumérique 2×16. ....	17
Figure 2.12 : Carte de commande basée sur le microcontrôleur dsPIC33Fj128GP802. ....	18
Figure 2.13 : Circuit électronique de détecteur de passage par zéros.....	20
Figure 2.14 : Signal de tension et détecteur de passage par zéros.....	21
Figure 2.15 : Mécanismes des échanges (DMA)[17].....	22
Figure 2.16 : Communication full-duplex[18] .....	23
Figure 2.17 : Programmeur PICKit2 .....	24
Figure 2.18 : Interface MPLAB .....	25
Figure 2.19 : Méthode de calcul de facteur de puissance.....	26
Figure 2.20 : Signaux de courant et de tension en cas de charge résistive.....	27
Figure 2.21: Déphasage entre le courant et la tension.....	27
Figure 2.22 : Module bluetooth HC-05[21] .....	30
Figure 2.23 : Diagramme de connexion ( module Bluetooth avec un microcontrôleur)[21] .....	31
Figure 2.24 : Couplage de module bluetooth HC-05 avec le bluetooth de pc. ....	37
Figure 2.25 : Configuration de port serie (COM) de notre module bluetooth HC-05.....	37
Figure 2.26 : Configuration de l'émulateur de terminal PuTTY .....	38
Figure 2.27 : Résultat obtenue par le moniteur PYTTY .....	39
Figure 3.1 : Téléchargement d'application (MIT AI2 Companion).....	42
Figure 3.2 : Site officiel de MIT App Inventor 2 .....	42
Figure 3.3 : Création d'un nouveau projet ,et nommé 'compteur'.....	43
Figure 3.4 : Environnement de développement : Page 'Designer' .....	44
Figure 3.5 : Méthode de sélection d'un objet 'HorizontalArrangement'.....	45
Figure 3.6 : Conception d'Application AI2. Ajout : Bouton ,label ,listPicker.....	47
Figure 3.7 : Sensor, (composant capteur d'horloge).....	48
Figure 3.8 : User interface(image) .....	49
Figure 3.9 : Design final de notre application .....	50

## Listes des figures

Figure 3.10 : Environnement de développement : Page 'Block' .....	50
Figure 3.11 : Sélection des fonctions et des lignes de code associés à l'objet 'ListPicker1' .....	51
Figure 3.12 : Couplage de module Bluetooth HC-05 avec le Bluetooth de notre Smartphone...	52
Figure 3.13 : Génération et scan du code QR.....	53
Figure 3.14 : Affichage de notre application avant connexion.....	53
Figure 3.15 : Affichage des tests de notre application mobile (Compteur).....	54
Figure 3.16 : Téléchargement d'application mobile sur notre Smartphone .....	57
Figure 3.17 : Partage de l'installateur de notre application .APK .....	55
Figure 3.18 : Réalisation finale de notre prototype .....	56

# Glossaire

GSM: Global System for Mobile communication

GPS: Global Positioning System

BLE: Bluetooth Low Energy

AMR: Automated Meter Reading

AMM: Advanced Meter Management).

EDF: Electricity of France

GRD: Gestionnaire du Réseau de Distribution

WAN: wide area network

Wi-Fi: wireless fidelity

TCP/IP : Transmission Control Protocol/Internet Protocol

HTABT : haute tension à basse tension.

ERDF : European Regional Development Fund

DSP : Digital Signal Processor

MCU : multipoint control unit

ADC : Analog-to-Digital Converter

CAN : Controller Area Network

UART : Univers Asychrenus Receive Transmitter

SPI : Serial Peripheral Interface

MIPS : Million Instructions Per Second

DMA : Date Memory Access

CPU : Central Processing Unit

RAM : Random Access Memory

EEPROM : Electrically-Erasable Programmable Read-Only Memory

FCY: Instruction Cycle Rate

ECAN: Enhanced Controller Area Network

UART: Universal Asynchronous Receiver Transmitter

# Introduction Générale

Avec le développement technologique de ces dernières années, l'intelligence 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 les développements technologiques et l'évolution croissante des besoins de l'humanité en matière d'applications qui concernent ces réseaux a conduit progressivement à considérer des changements satisfaisants des demandes et des performances importantes.

Ces changements sont pour une bonne part liés aux techniques et aux supports de communication utilisés dans les réseaux de télécommunication. Leurs intégrations (réseaux de communication) locaux et/ou à grande distance dans les réseaux électriques ont conduit au concept de réseau électrique intelligent, dans lequel le consommateur et le producteur d'énergie électrique ont accès à plusieurs types d'information, grâce au développement de l'électronique numérique.

D'autre part, la modernisation du comptage qui consiste à remplacer le compteur classique (électromécanique ou numérique de première génération) par un compteur dit « intelligent », permet de communiquer directement avec les systèmes d'information des gestionnaires du réseau de distribution d'électricité, rendant inutiles les relevés sur site. Le compteur intelligent informera le client de sa consommation en temps réel et à terme permettra le développement du marché de l'énergie, du prix de l'électricité qui sera en fonction de l'heure de la journée. Le compteur intelligent permettra également la centralisation de l'information utiles concernant les réseaux électriques (qualité de service, profil de consommation, tentatives de fraude...) en remontant les données directement du consommateur, ou commander l'enclenchement de tout appareil domestique à forte consommation électrique durant les périodes creuses, voir délester automatiquement ces mêmes appareils en période de pointe. Le compteur pourra aussi enregistrer les flux d'énergie dans les deux sens (consommée et produite), dans le cas de la production d'électricité notamment de type renouvelable, appelée à se développer de plus en plus.

## **Introduction Générale**

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

Dans le premier chapitre, nous avons tout d'abord présenté les généralités sur les compteurs électriques, puis nous avons présenté une étude détaillée sur les compteurs intelligents en donnant les définitions, les caractéristiques, les principes de bases et l'architecture des compteurs.

Dans le second chapitre, nous nous sommes consacrés à une étude approfondie du dispositif expérimental réalisé autour d'un dsPIC ainsi que la description de chaque bloc du circuit du compteur, et la programmation du DSPIC.

Enfin, nous exposerons dans le troisième chapitre, un aperçu sur le logiciel APP Inventor et la réalisation d'une application capable de gérer la communication entre la carte électronique et le Smartphone ou Tablette.

# **Chapitre 1 Généralité sur les compteurs électriques et les systèmes de communication utilisés**

## 1. Introduction

Un **compteur électrique** est un organe électrotechnique servant à mesurer la quantité d'énergie électrique consommée dans un lieu : habitation, industrie... Il est utilisé par les fournisseurs d'électricité afin de facturer la consommation d'énergie au client. A l'origine, ces appareils étaient de conception électromécanique, ils sont remplacés dorénavant par des modèles électroniques. Les nouvelles versions de compteurs électriques sont des compteurs communicants appelés parfois « compteurs intelligents ».

L'unité légale d'énergie est le joule (J). Le watt (W), unité de puissance, correspondant à un flux d'énergie d'un joule transféré en une seconde. Cependant, l'unité d'énergie habituellement utilisée pour la consommation électrique est le kilowatt-heure (kWh) [1]. Un kWh est équivalent à  $3,6 \times 10^6$ J.

Sur le marché, il existe plusieurs types de compteur. On trouve par exemple :



Figure 1.1 : Compteur classique monophasé heliowatt [4].



Figure 1.2 : Compteur électronique monophasé type « EDF » SAGEM [4].



Figure 1.3 : Compteur monophasé « modulaire » Polier [4].



Figure 1.4 : Compteur de consommation d'appareillage électrique [4].



Figure 1.5 : Compteur communicant [4].

Pour comprendre mieux le fonctionnement des compteurs intelligent, nous allons dans ce qui va suivre dans ce chapitre, donner une description globale sur le compteur intelligent, ainsi que les différents moyens de communication utilisés par ce type de compteur.

## 2. Description d'un compteur intelligent

### 2.1. Définition d'un compteur intelligent

Un compteur intelligent ou compteur communicant est un compteur disposant de technologies dites AMR (Automated Meter Reading) qui mesure de manière détaillée et précise, et éventuellement en temps réel, une consommation d'électricité, d'eau ou de gaz. La transmission des données s'effectue par ondes radio ou par courants porteurs en ligne (CPL) au gestionnaire du réseau de distribution chargé du comptage.

Les compteurs communicants servent à produire des factures sur la consommation réelle, à repérer des gaspillages, des pertes en ligne ou les postes qui coûtent le plus au client. Ils peuvent éventuellement l'informer de microcoupures ou de pertes des réseaux électriques.

Si le compteur est en outre « programmable à distance » et équipé d'un appareil de coupure à distance, il est dit « AMM » (Advanced Meter Management). Ces fonctions sont à la base de l'évolution des réseaux (d'électricité, d'eau ou de gaz) vers les « réseaux intelligents ». [1]



Figure 1. 6 : Compteur électrique communicant à un smart grid [1].

Ainsi l'architecture générale du réseau électrique basé sur des compteurs intelligents est représenté sur la figure suivant.

## Chapitre 1 : Généralité sur les compteurs électrique et les systèmes de communication utilisés

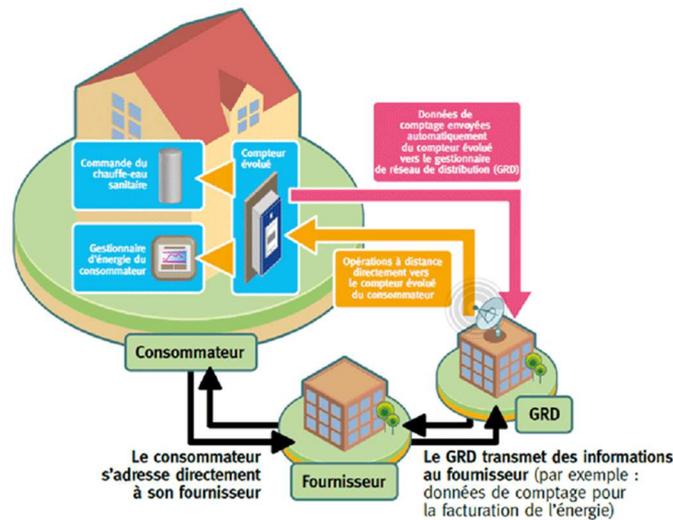


Figure1.7 : Architecture de réseau basé sur des compteurs intelligents (des compteurs électrique évalués)[5].

Les caractéristiques de ces compteurs intelligents sont résumées dans le tableau suivant :

Affichage	Mesure	Enregistrement	Communication	Tarifcation
Compteur numérique à affichage numérique	Quantité d'électricité consommée et à quel moment (généralement par heure avec horodatage)	Relevés automatisés : les compteurs transmettent les données aux entreprises de distribution par voie électronique via un réseau sans fil	Communications bilatérales entre les compteurs et les entreprises de distribution	Tarifcation selon l'heure de consommation (structure tarifaire à trois niveaux : période de pointe, période médiane et période creuse) reflétant les fluctuations des coûts de l'électricité durant la journée

Tableau I.1 : Caractéristiques de ces compteurs intelligents.

### 2.2. Fonctionnement du système

Comme dans d'autres domaines comme la banque ou la téléphonie, la digitalisation est en marche pour notre énergie. Nos bons vieux compteurs électromécaniques font progressivement place aux compteurs électroniques puis aux compteurs intelligents, aussi appelés compteurs intelligents, smart meters ou encore compteurs communicants.

Contrairement aux compteurs électro-mécaniques ou électroniques qui doivent être relevés manuellement, les compteurs intelligents : [3]

## Chapitre 1 : Généralité sur les compteurs électrique et les systèmes de communication utilisés

- Enregistrent dans leur mémoire, selon un protocole défini, la puissance électrique prélevée et les quantités consommées à différents moments de la journée chaque jour de la semaine ;
- Transmettent de manière électronique ces données au gestionnaire de réseau ou au client via un système de communication ;
- Contrôlés et vérifiés à distance par le gestionnaire de réseau ;
- Envoyent une alarme au gestionnaire du réseau en cas d'ouverture du capot et donc, de suspicion de fraude. [3]

### 3. Système de communication

#### 3.1. Définition

Un système de communication est un ensemble organisé d'éléments permettant une communication. Il comprend les éléments de base suivants : un émetteur, un récepteur, un message et un canal véhiculant le message jusqu'au récepteur.

Parmi les moyens de communication, on trouve la transmission sans fil qui est un mode de communication à distance utilisant des ondes électromagnétiques modulées comme vecteur. Avec celles-ci, les distances peuvent être courtes quelques mètres comme pour le cas de la technologie Bluetooth.

Les applications les plus courantes de la transmission sans fil incluent les téléphones portables, les GPS, les souris et les claviers d'ordinateur, les réseaux informatiques étendus (les WAN), les casques audio, les récepteurs radio et la télévision par satellite.[6]

#### 3.2. Meilleurs moyens de communication sans fil avec contrôleur

##### 3.2.1. Wifi

Le module WiFi ESP8266 est un module émetteur-récepteur série WiFi basé sur le processeur ESP8266. Cette puce implémente une pile de protocoles TCP / IP complète, qui a une caractéristique très intéressante et qu'elle dispose également d'une grande puissance de calcul intégrée. Cela signifie que nous pouvons utiliser cette carte comme une simple carte de connexion WiFi, déchargeant le processeur principal de votre contrôleur de la gestion de la communication WiFi, ou nous pouvons exploiter toute sa puissance en implémentant la logique à l'intérieur du tableau lui-même. Ainsi, nous pouvons gérer des capteurs et élaborer de manière autonome leurs signaux et leurs mesures. De plus, l'hébergement d'applications simples à bord peut permettre de concevoir une solution IoT très compacte. [7]

### 3.2.2. Bluetooth / BLE4.0

Le Bluetooth est l'un des protocoles sans fil les plus utilisés dans les spécifications IoT, en particulier avec l'introduction récente de l'extension Bluetooth Low Energy, également connue par les utilisateurs d'Apple sous le nom d'iBeacon, dont les avantages du protocole BLE ont été largement compris dans la version Bluetooth 4.0 qui les a intégrés dans le noyau principal, car cette norme utilise une consommation électrique extrêmement faible, ce qui facilite la fabrication de cartes entièrement alimentées par batterie et dont la durée de travail est supérieure à 1 ou 2 ans dans certains cas. Une autre fonctionnalité, déjà implémentée sur de nombreux smartphones, est la possibilité d'intégrer le protocole de gestion directement au niveau du noyau sans demander aucune intervention des utilisateurs. Cela facilite la configuration d'un réseau maillé de périphériques Bluetooth, avec une latence inférieure et une portée supérieure à celle du Bluetooth standard. [7]

### 3.2.3. GSM / GPS

La communication GSM est indispensable pour de nombreux projets de communication contrôlés ou télécommandés. Le principal problème est de savoir comment recevoir des notifications en cas d'événement. La meilleure pratique consiste à recevoir un SMS. Donc, le module GSM est ce dont vous avez besoin pour cela. Ou, si vous souhaitez contrôler des appareils simples, tels que l'ouverture du portail de stationnement ou similaire, vous pouvez utiliser une carte opto-isolée compatible GSM900. [7]

### 3.2.4. Radio RF

Si le GSM est «trop» pour quelque projet, alors que les fonctionnalités BLE 4.0 ou Zigbee de la mise en réseau et de la topologie de réseau maillé ne sont pas nécessaires, la dernière, mais non la moindre, alternative est le transceiver RF. Open-electronics envisage de vendre très prochainement deux des cartes les mieux notées de la communauté Maker, une solution de 433 MHz et une solution de 2,4 GHz. [7]

Le Semtech LoRa SX1278 quand à lui est un émetteur-récepteur RF à faible coût de 433 MHz. Il conserve les avantages du RFIC SX1278 mais simplifie la conception du circuit. La haute sensibilité (-136 dB) de la modulation LoRa et la sortie haute puissance de 20 dB font de ce module un outil idéal pour les applications à faible bande passante et à faible débit de données. [7]

### 4. Conclusion

Les compteurs communicants constituent un outil majeur au service de la transition énergétique. Outre sa simplification évidente qu'ils représentent pour les opérateurs, ils permettent de piloter plus efficacement le réseau électrique, d'intégrer de nouveaux moyens de production, de nouvelles formes de consommation et, surtout, ils permettent aux consommateurs de mieux connaître et donc de mieux maîtriser leur consommation électrique. Le compteur communicant est un produit de grande qualité technologique, qui doit être utilisé de manière à profiter de tout son potentiel. Cet outil de comptage ne sera pleinement exploité qu'à la condition que les consommateurs se l'approprient, en comprennent le fonctionnement et les enjeux, et que les offres permettant l'exploitation des données de comptage se développent. C'est à cette condition que seront atteints les objectifs d'économie d'énergie et de meilleure fourniture d'énergie.

## **Chapitre 2 Description Générale de notre prototype de compteur intelligent**

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

### 1. Introduction

Les compteurs actuels sont des compteurs électromécaniques ou électroniques. Ils mesurent l'énergie consommée et nécessitent l'intervention de techniciens pour les opérations les plus simples de la vie courante (mise en service, relevé, modification de puissance) [8]

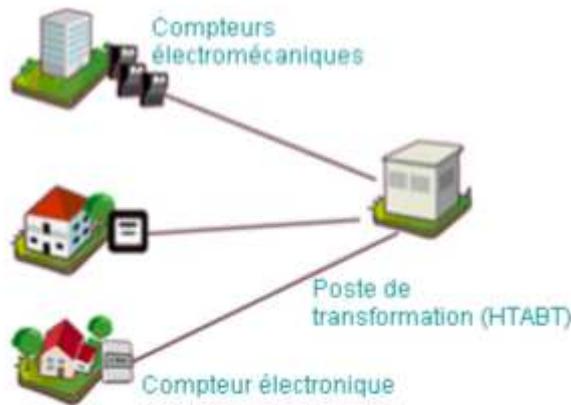


Figure 2.1 : Compteur électromécanique et compteur électrique [8].

Un compteur communicant signifie qu'il peut recevoir et envoyer des données et des ordres sans l'intervention physique d'un technicien. Installé chez les clients et relié à un centre de supervision, il est en interaction permanente avec le réseau, qu'il contribue à rendre «intelligent». [8]

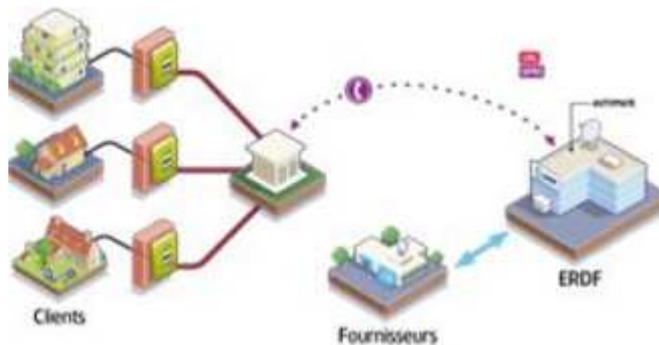


Figure 2.2 : Compteur communicant [8].

Le compteur intelligent est un compteur de nouvelle génération avec un système d'information associé. Ce compteur est capable de recevoir des ordres et de transmettre des informations à distance. Pour cela, il communique avec un concentrateur (sorte de mini-ordinateur intégré aux postes de transformation gérés par une entreprise de gestion et de contrôle de l'énergie électrique) qui est relié au centre de supervision.

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

### 2. Description générale de notre prototype

Notre compteur intelligent est un compteur électronique monophasé conçu pour la mesure des données de consommation comme la tension, le courant, les différentes formes de puissance (active et réactive), la puissance apparente, le facteur de puissance et l'énergie électrique.

Le principe de fonctionnement de ce compteur est simple car il est basé sur les informations obtenues par le capteur de tension et de courant. Après, les mesures sont transmises vers un microcontrôleur de type dsPIC33FJ128GP802 afin de calculer les données de consommation.

Aussi, pour afficher toutes ces données de consommation, on utilise les deux méthodes suivantes :

- Afficheur LCD,
- Application Android.

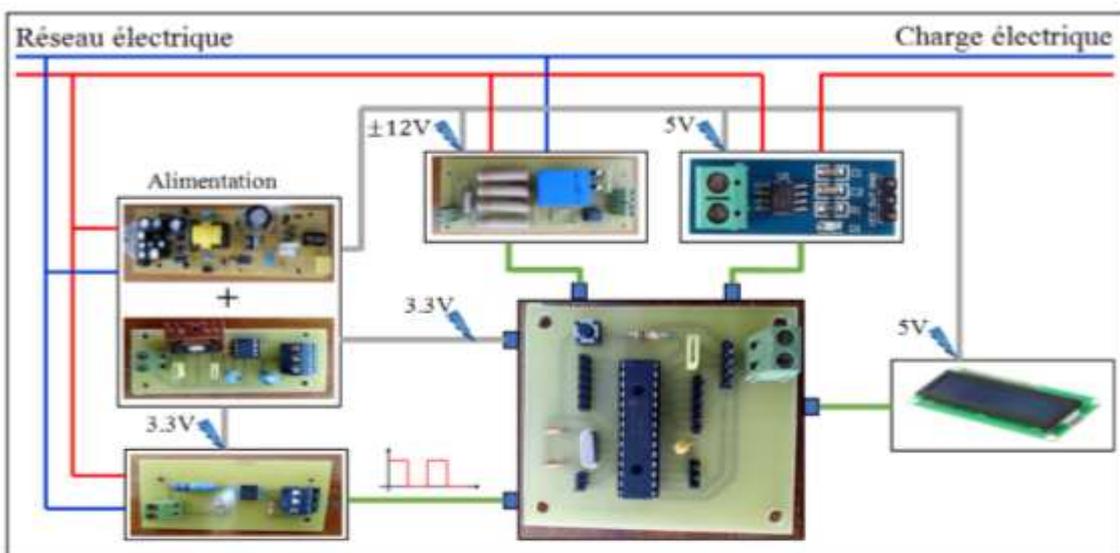


Figure 2.3 : Schéma global de prototype de notre compteur intelligent.[09]

### 3. Représentation détaillée de notre prototype

Le compteur intelligent est un dispositif qui dispose d'une technologie avancée par rapport aux autres compteurs électriques, car il est capable d'assurer deux fonctions au même temps :

- Mesure des données de consommation comme la tension, le courant, la puissance active et la puissance réactive, l'énergie...etc,
- Utilisation des systèmes de communication qui sont un facteur très important dans le compteur intelligent afin de changer ou de transmettre les informations entre le producteur et le consommateur.

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

Ainsi, dans ce chapitre nous allons expliquer en détail le fonctionnement des principaux composants nécessaires pour réaliser notre compteur intelligent.

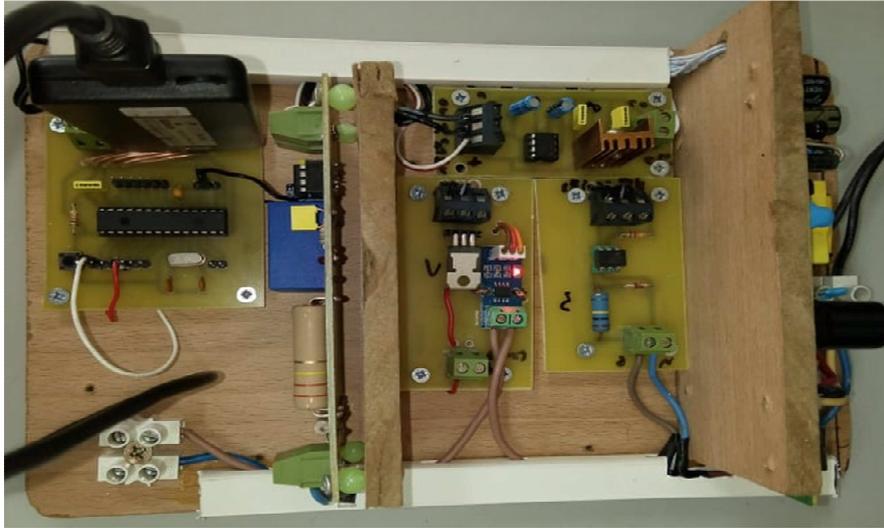


Figure 2.4 : Différents circuits de prototype de notre compteur intelligent.

### 3.1. Circuit d'alimentation

Dans toutes les applications électroniques, l'alimentation est un sous-système incontournable. Sa fonction consiste à fournir une ou plusieurs tensions et courants continus stabilisés. On distingue deux grandes familles d'alimentations stabilisées, soit :

- Les alimentations à régulateurs linéaires.
- Les alimentations à découpage.

Dans notre réalisation, on a opté sur l'utilisation d'une alimentation à découpage qui se trouve sur le marché utilisé essentiellement comme alimentation de démodulateur. [9]

#### 3.1.1. Fonctionnement d'une alimentation à découpage

Les alimentations à découpages sont des alimentations fondamentalement basées sur les éléments suivants :

- Un contacteur statique commandé à fréquence fixe mais à rapport cyclique variable. La gamme de fréquence allant de quelques dizaines de kHz à plusieurs centaines de kHz,
- Une inductance permettant l'accumulation d'énergie durant une partie du cycle et sa restitution durant l'autre partie,
- Un condensateur de filtrage stockant l'énergie pour l'étage de découpage,
- Un transistor de découpage (technologie MOS) monté sur un radiateur,
- Optocoupleur assurant l'isolation des parties haute et basse tension. [10]

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

### 3.1.2. Caractéristiques d'une alimentation à découpage

Les alimentations à découpages sont caractérisées par :

- Une légère ondulation de la tension de sortie,
- Un très haut rendement,
- Un faible encombrement (lié au haut rendement),
- Une boucle d'asservissement relativement complexe,
- Une réponse plus lente aux perturbations,
- Une génération relativement importante de perturbations électromagnétiques. [10]

### 3.1.3. Éléments constitutifs

La photographie ci-dessous représente les éléments principaux d'une alimentation à découpage.



Figure 2.5 : Éléments principaux d'une alimentation à découpage.[10]

1. Connecteur d'alimentation secteur 230v,
2. Fusible de protection,
3. Filtre EMI, avec une bobine d'arrêt,
4. Pont de diodes,
5. Condensateur de filtrage, qui stocke l'énergie pour l'étage de découpage,
6. Transistor de découpage (technologie MOS) monté sur un radiateur,
7. Transformateur ou circuits magnétiquement couplés : dispositif qui permet une modification du niveau de tension et parfois l'isolation des parties haute et basse tension,
8. Diode Schottky (commutation rapide) montée sur un radiateur,
9. Condensateur de filtrage,
10. Bobine de filtrage,
11. Circuit de commande de l'optocoupleur,
12. Optocoupleur assurant l'isolation des parties haute et basse tensions,
13. Circuit de commande du transistor de découpage,
14. Sortie d'alimentation,
15. Régulateur de tension à découpage monté sur un radiateur. [10]

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

### 3.1.4. Comparaison par rapport aux alimentations linéaires

Les alimentations à découpage offrent de nombreux avantages par rapport aux alimentations linéaires, qui sont :

- La puissance dissipée dans le composant électronique utilisé en commutation est moindre que lorsqu'il est utilisé en mode linéaire. En conséquence, le rendement de l'alimentation est meilleur et le radiateur de dissipation est de dimensions bien plus modestes,
- Le transformateur fonctionne à une fréquence bien plus élevée que celle du secteur (au-delà de 20 kHz contre 50 ou 60 Hz), ce qui permet de réduire considérablement sa taille (voir la relation de Boucherot),
- Le régulateur linéaire est capable d'assurer une tension continue constante en sortie quelle que soit la charge appliquée, mais dans le cas de chute de tension, le régulateur devient incapable d'assurer les tensions continues souhaitées en sortie. [10]

Les alimentations à découpage sont bien plus petites et plus légères que les alimentations linéaires et de plus, elles possèdent un meilleur rendement, mais le problème majeur est qu'ils sont incapables de fournir des tensions négatives, or pour notre projet nous avons besoin d'une tension de  $\pm 12V$  pour alimenter le capteur de tension LV25-p. Pour résoudre ce problème nous avons réalisé le circuit suivant :

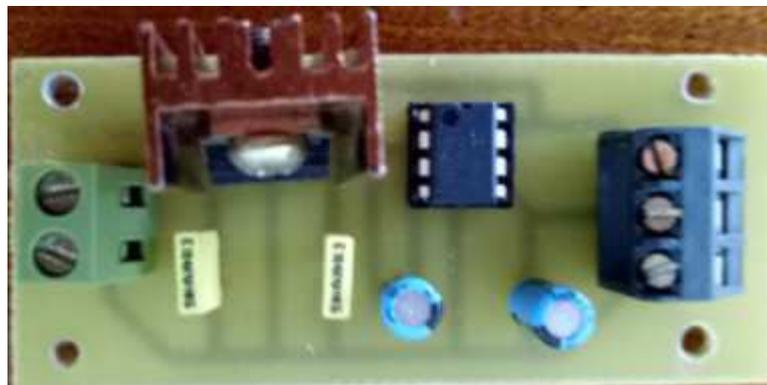


Figure 2.6 : Circuit électronique pour créer une tension  $\pm 12V$ .

### 3.2. Capteur

Un capteur est un dispositif transformant l'état d'une grandeur physique observée en une grandeur utilisable, telle qu'une tension électrique, une hauteur de mercure, une intensité ou la déviation d'une aiguille. On fait souvent (à tort) la confusion entre capteur et transducteur : le capteur est au minimum constitué d'un transducteur. Les capteurs sont les éléments de base des systèmes d'acquisition de données. [11]

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

### 3.2.1. Capteur de tension

La mesure des tensions passe généralement par la collecte d'un petit courant  $I_{mes}$  qui lui est proportionnel, lui-même mesuré par un capteur de courant LEM adapté.  $I_{mes}$  étant de faible intensité pour limiter la consommation en puissance, ces capteurs doivent être d'une très grande sensibilité. Ils doivent également présenter une inductance d'insertion limitée pour garantir une bonne plage de fréquences. Pour y parvenir, une résistance R1 est généralement montée en série avec l'enroulement primaire afin de réduire la constante de temps électrique et, également, d'ajuster le courant  $I_{mes}$ . Pour limiter la sensibilité au mode commun, la résistance R1 est souvent partagée en deux résistances placées de part et d'autre du capteur [12].

#### 3.2.1.1. Description du capteur de tension LEM LV 25-P

Le Capteur de tension LEM LV 25-P est un capteur à effet Hall. Le composant fournit une isolation galvanique entre les circuits primaire et secondaire. Pour activer la mesure d'une tension, un courant proportionnel à la tension mesurée doit être recueilli à travers une résistance externe, choisi par l'utilisateur, en série avec le circuit primaire de l'unité comme le montre la figure ci-dessous. Les caractéristiques principales de ce capteur sont [12]:

- Convient pour la mesure électronique des tensions associées à des courants continus, alternatifs et des circuits d'impulsions,
- Fournit une isolation galvanique entre les circuits primaires et secondaires,
- Transducteur de courant Boucle fermée (compensée),
- Excellente précision,
- Très bonne linéarité,
- Faible dérive thermique,
- Faible Temps de réponse,
- Bande passante élevée,
- Grande immunité contre les interférences externes,
- Faible perturbation en mode commun.

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent



Figure 2.7 : Circuit électronique de capteur de tension LV25-p.[09]

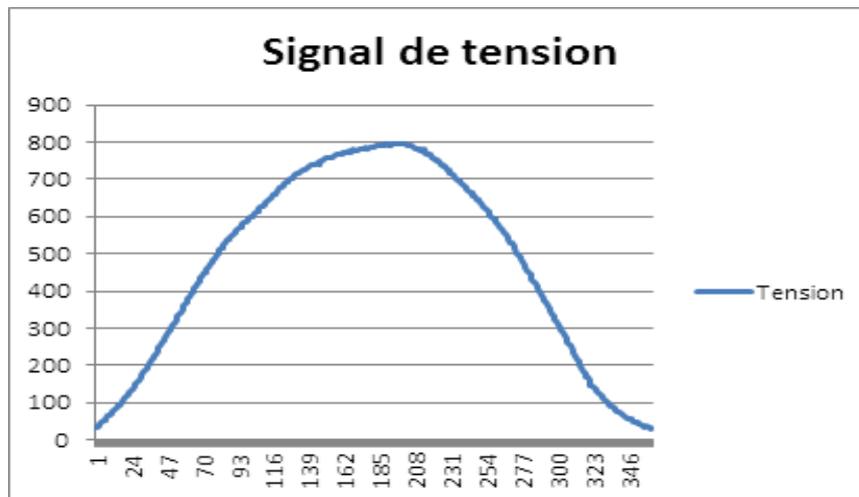


Figure 2.8 : Signal de tension.

D'après les résultats obtenus nous remarquons une excellente précision et une très bonne linéarité ce qui facilite le calcul de la tension.

Pour déterminer le coefficient de linéarité nous avons utilisé un oscilloscope numérique. Ainsi, pour une tension efficace de 230 V à l'entrée de notre capteur nous avons une tension de 2.6 V à la sortie. Ainsi la valeur maximale de tension à la sortie de notre capteur est calculée comme suit :

$$V_{in} = \frac{V_{out} * 3.38}{1023} \text{ [V]} \quad (2.1)$$

Avec  $V_{out}$  représente la valeur numérique maximale obtenu par l'ADC.

Sachant que la valeur efficace de la tension du réseau est égale à :

$$V = \frac{(V_{out} * 3.38) * 230}{1023 * 2.6} = 0.29 * V_{out} \text{ [V]} \quad (2.2)$$

Ainsi, nous obtenons la valeur maximale de la tension du réseau calculée comme suite :

$$V_{max} = V * \sqrt{2} \text{ [V]} \quad (2.3)$$

D'après le graphe obtenu on remarque que  $V_{out}$  max est égal à 800, donc :

$$V_{max} = 0.29 * V_{out} * \sqrt{2} = 0.29 * 800 * \sqrt{2} = 232 * \sqrt{2}$$

$$V_{max} = 328.1 \text{ V.}$$

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

### 3.2.2. Capteur de courant

Ce capteur est basé sur le capteur ACS712, qui peut détecter avec précision les signaux de courant alternatif ou de courant continue. Les courants maximaux qui peuvent être détectés peuvent atteindre 30A et le signal de courant actuel peut être lu via un port analogue, ce capteur est caractérisé principalement par [13] :

Alimentation : 5Vcc

Plage de mesure :  $\pm 30A$

Sortie analogique : 66 mV/A

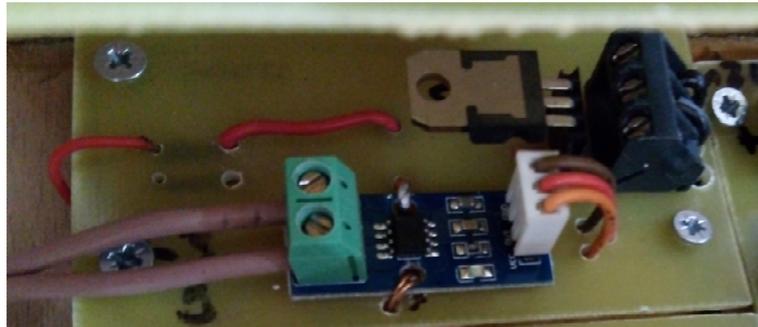


Figure 2.9 : Capteur de courant ACS712 - 30A.

L'ACS712 d'Allegro Microsystems utilise l'effet hall pour fournir une sortie analogique de 66mV/A. [13]

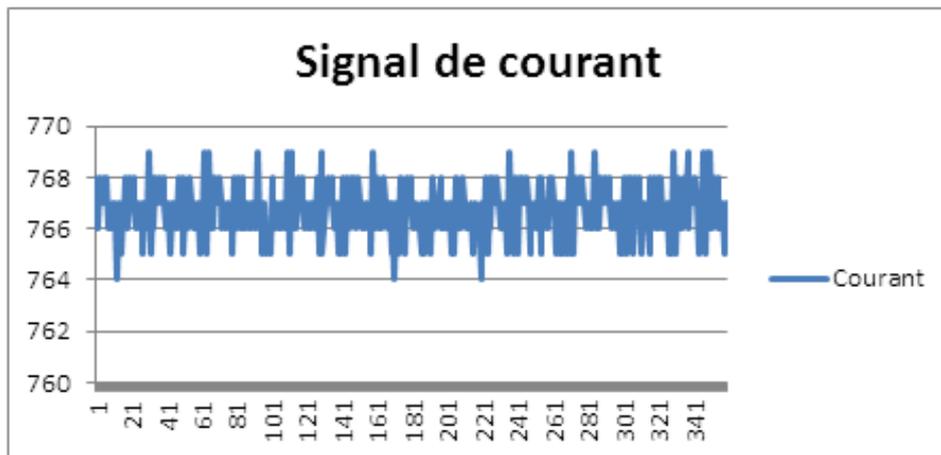


Figure 2.10 : Signal du courant.

D'après la Figure 2-10, nous remarquons que la valeur de l'offset varie entre 767 et 769, ce qui montre la présence de bruits de mesures négligeable à vide.

La valeur ci-dessus '767' représente l'offset. Il suffit de la retrancher pour trouver la valeur exacte. Cet offset est introduit pour pouvoir mesurer des courants alternatifs de telle façon la tension de sortie du capteur est toujours positive.

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

Pour déterminer le coefficient de linéarité nous avons utilisé un oscilloscope numérique. Ainsi, pour un courant nul à l'entrée de notre capteur nous avons une tension de 2.6 V à la sortie. Ainsi la valeur maximale du courant à la sortie de notre capteur est calculée comme suit :

$$I_{in} = \frac{I_{out} * 3.38}{1023} \text{ [A]} \quad (2.4)$$

Avec  $V_{out}$  représentant la valeur numérique maximale obtenu par l'ADC. La valeur efficace de la tension du réseau est égale à :

$$I = \frac{(I_{out} * 3.38) * 20.28}{1023 * 2.5} = 0.027 * I_{out} \text{ [A]} \quad (2.5)$$

### 3.3. Afficheurs LCD

Les afficheurs à cristaux liquides, autrement appelés afficheurs LCD (Liquid Crystal Display), sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement. Ils consomment relativement peu (de 1 à 5 mA), sont relativement bons marchés et s'utilisent avec beaucoup de facilité [14].



Figure 2.11 : Afficheur LCD alphanumérique 2×16.

### 3.4. Carte de commande basée sur le microcontrôleur dsPIC33F

#### 3.4.1. Définition

Le DSPIC est une fusion entre un DSP et un PIC, ce qui permet d'avoir des hautes performances de calcul. Ce microcontrôleur peut travailler à une vitesse de 40MIPS qui reste un paramètre très important lorsqu'on travaille en temps réel, aussi il dispose de plusieurs périphériques (entrée/sortie logique, convertisseur ADC, convertisseur DAC, UART ,DMA.....).

Dans notre prototype, le type de dsPIC utilisé est le **dsPIC33FJ128GP802** qui est un microcontrôleur à noyau DSP, donc un DSC (Digital Signal Controller), de 16 bits, possédant un jeu de 83 instructions, chacune stockée dans un seul mot (16 bits) de programme et est exécutée en 1 cycle

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent



Figure 2.12 : Carte de commande basée sur le microcontrôleur dsPIC33Fj128GP802.[09]

Comme tout DSC, chaque broche du dsPIC33F à une ou plusieurs fonctions, le dsPIC33F possède 28 pins montés sur un boîtier SPDIP (de largeur 300 MILS). [15]

Le dsPIC33FJ128GP802 possède aussi [15] :

- Un bus de données 16 bits,
- Un bus d'instructions 24 bits,
- Une mémoire de type flash,
- Une conversion analogique numérique 10 bits (ADC),
- Une acquisition de signaux numériques (Input Capture avec 4 canaux),
- Et une communications CAN, i2c, UART, SPI.

### 3.4.2. Organisation de la mémoire

La mémoire du dsPIC33F est répartie comme suit :

- 128kB de mémoire flash (programme),
- 2048 octets de mémoire données (RAM).

Le dsPIC33F dispose de deux espaces de données, qui peuvent être accédé soit :

- Séparément, le cas de certaines instructions de traitement numérique de signal (DSP),
- Simultanément comme une seule plage d'adressage linéaire de 64Kb le cas des instructions MCU du microcontrôleur [16].

### 3.4.3. Principaux modules du dsPIC33F

#### 3.4.3.1. Convertisseur analogique-numérique (ADC)

Le dsPIC33F dispose de 32 entrées analogiques multiplexées sur deux convertisseurs (ADC). Chaque ADC peut réaliser une conversion sur 10 ou 12 bits. Ce

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

périphérique, aux modes de fonctionnement éventuellement très complexes, sera ici utilisé dans ses versions les plus simples.

Le module ADC va convertir un signal analogique, de niveau 0 à 3.3V ou 5V, en une donnée numérique de type flottant ou entier, signé ou non (**AD1CON1\FORM<1:0>**). Le principe est de réaliser un échantillonnage du signal à une fréquence plus ou moins haute, puis de convertir ce signal de manière régulière ou événementielle (**AD1CON1\SSRC<2:0>**) [16]

Une conversion se déroule en deux étapes :

- L'échantillonnage : durant cette étape, la grandeur analogique est échantillonnée (on « prend une photo de la tension ») et est maintenue stable pour la numérisation. Cette étape est réalisée par un échantillonneur bloqueur (Sample and hold ou S/H). Celui-ci peut être vu comme un amplificateur connecté à une capacité via un interrupteur. Pendant l'échantillonnage, la capacité est connectée à l'amplificateur. A la fin de l'échantillonnage, l'amplificateur est déconnecté et la capacité garde la tension [16].
- La numérisation : c'est pendant cette étape que la grandeur analogique est convertie en une donnée numérique. Elle est lancée par la demande de début de numérisation [16]

Dans le cas où le timer 3 lance la conversion, il n'est pas nécessaire de tester le flag d'interruption ou d'écrire la routine d'interruption de ce timer : à chaque débordement, il envoie automatiquement le signal de début de numérisation. Seule la fin de la conversion doit être gérée en software [17].

```
//Configuration de l'ADC
void initADC()
{
    AD1CON1bits.FORM = 0; //La forme de résultats de conversion: entier
non signé (0 à 1023)
    AD1CON1bits.SSRC = 2; //la fin de l'échantillonnage et le début de
la conversion sélectionné par timer3
    AD1CON1bits.ASAM = 1; // l'echantillonnage commence immédiatement
après la dernière conversion (echantillonnage automatique)
    AD1CON1bits.AD12B = 0; // convertisseur sur 10 bits
    AD1CON1bits.SIMSAM = 0; //Le mode d'échantillonnage simultané (2
entrée analogique)
    AD1CON2bits.CHPS = 1; //le nombre de canaux sélectionnés CH0/CH1
    AD1CON3bits.ADRC = 0; //L'horloge de système est sélectionné pour
être l'horloge de conversion ADC
    AD1CON3bits.ADCS = 3; //Le temps de conversion: Tc=12*Tad = 1.2us
tel que :
//12 --> Conversion de 10 bits + 1 bit de départ + 1 bit de fin de
conversion.
// Le Tad doit être au moins 75 ns. On a choisi une valeur standard de 100
ns.
// Tad= Tcy*(ADCS+1)=100ns avec Tcy=1/40M et ADCS=3.
    AD1CON1bits.ADDMABM = 1; //les résultats de conversion sont remplis
dans la zone RAM DMA utilisés pour l'ADC dans l'ordre de conversion
```

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

```
AD1CON2bits.SMPI = 1; // pointeur d'adresse DMA incrémente chaque
échantillon.
//Une interruption ADC est générée après la fin de chaque conversion,
quelle que soit la SMPI <3: 0> Paramètres des bits.
AD1CON4bits.DMABL = 4; // chaque buffer contains 16 mots.
AD1CON2bits.VCFG=0; // Vref+=AVDD (3.3v) & Vref-=AVSS (0v)
AD1CHS0bits.CH0SA = 0; // selection de l'entrée analogique AN0 sur le
CH0 (mesure)
AD1CHS0bits.CH0NA = 0; // selection de la référence Vref- (masse)
AD1CHS123bits.CH123SA = 1; // selection de l'entrée analogique AN3 sur
le CH1(mesure)
AD1CHS123bits.CH123NA = 0; //selection de la référence Vref- (masse)

IFS0bits.AD1IF = 0; //reset du flag d'interruption lié à l'ADC
IEC0bits.AD1IE = 0; // désactiver l'interruption lié à l'ADC
AD1CON1bits.ADON = 1; // activation du convertisseur ADC
}
```

Pour notre projet nous avons utilisé l'ADC qui permet de convertir un signal analogique (tension, courant) d'amplitude 0 à 3.3 V en une valeur numérique, et pour échantillonner au même instant deux entrées analogiques (AN0 et AN3) qui correspondent aux signaux analogiques (tension et courant) par les chaines d'échantillonnage (CH0 et CH1).

```
void initTmr3()
{
    TMR3 = 0; //initialiser le timer3 à 0
    PR3 =1111; // période de comptage
    IFS0bits.T3IF = 0; // Clear Timer 3 interrupt
    IEC0bits.T3IE = 0; // Disable Timer 3 interrupt
}
```

Nous avons choisi le timer3 pour indiquer dans le programme le moment où l'ADC arrête l'échantillonnage et commence la conversion (ce timer3 saute toute les 27.78  $\mu$ s, la période de comptage égale à 1111).

On a utilisé une interruption externe au niveau du pin INT0, pour lancer le timer3 seulement dans l'alternance positive du signal de tension (on a choisi le signal de tension comme référence), pour cela on a généré un signal d'interruption par la réalisation d'un simple circuit à base d'un optocoupleur 4N35



Figure 2.13 : Circuit électronique de détecteur de passage par zéros.[09]

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

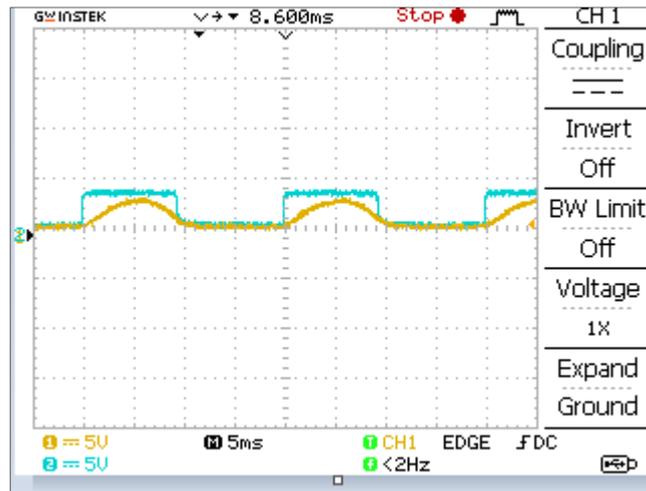


Figure 2.14 : Signal de tension et détecteur de passage par zéros.

La Figure 2.14 montre bien que le circuit de passage par zéros fonctionne correctement, tel que pendant l'alternance positive de tension générer à la sortie du capteur de tension, le signal générer par le détecteur de passage par zéros est égale à 3.3V, donc grâce à ce signal nous pouvons lancer la routine de l'interruption seulement pendant l'alternance positive du signal de tension (la tension en phase).

### 3.4.3.2. Accès direct à la mémoire DMA

L'accès direct à la mémoire (Direct Memory Access DMA) est une technique matérielle facilitant les opérations d'entrée-sortie, c'est-à-dire les échanges de données entre le microcontrôleur, ou le microprocesseur, et le monde extérieur [17].

Le contrôleur DMA dispose de huit canaux indépendants. Chaque canal peut être configuré pour les transferts vers ou depuis les périphériques sélectionnés. Les périphériques pris en charge par le contrôleur DMA comprennent [17]:

- Interface de convertisseur de données (DCI),
- Convertisseurs 10 bits/12 bits analogique/numérique,
- Interface périphérique série,
- L'UART,
- Input Capture,
- Output Capture.

De plus, les transferts DMA peuvent être déclenchés par des timers (compteurs matériels) ainsi que des interruptions externes. Chaque canal DMA est unidirectionnel. Deux canaux DMA doivent être alloués pour lire et écrire à un périphérique interne (ADC, bus CAN,...) [17].

### 3.4.3.3. Principe du DMA

Des données vont pouvoir être échangées entre la mémoire centrale et les circuits périphériques sans intervention du CPU. Un sous-système indépendant du CPU va transférer les données du circuit périphérique vers la mémoire (entrée) ou de la mémoire vers le circuit périphérique (sortie) [17].

#### 3.4.3.3.1. Fonctionnement du DMA

Le sous-système de DMA va être configuré et démarré par le CPU. Les données vont être échangées via des cycles de lecture ou d'écriture en mémoire initiés par le sous-système de DMA. Lorsque les opérations seront terminées le sous-système de DMA va interrompre le CPU [17].

#### 3.4.3.3.2. Sous-système de DMA

Le sous-système de DMA comporte :

- Un registre d'adresse qui va contenir l'adresse où les données doivent être placées ou lues en mémoire. Ce registre pourra être incrémenté ou décrémenté à chaque transfert
- Un compteur qui compte le nombre de données échangées
- Un registre où les données vont transiter entre la mémoire et le circuit périphérique [17].

#### 3.4.3.3.3. Mécanismes d'échanges

Le contrôleur de l'espace mémoire de donnée du dsPIC33F contient aussi une DMA, c'est un sous-système important dans la famille des DSC et DSP, ce sous-système facilite le transfert de données entre le processeur et son périphérique sans l'assistance du CPU [17].

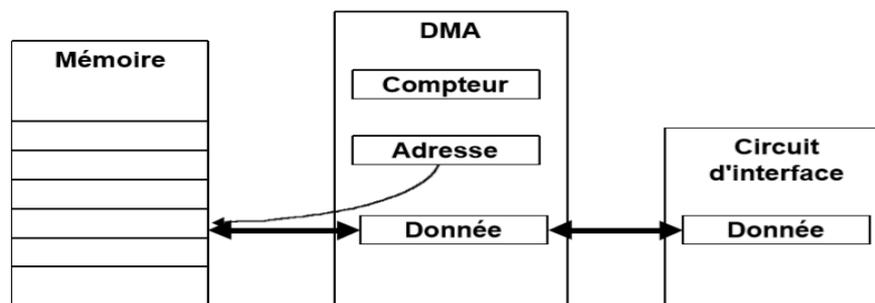


Figure 2.15 : Mécanisme d'échanges (DMA)[17].

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

```
void initDma0(void)
{
    DMA0CONbits.AMODE = 0; //Mode d'adressage Post-Increment.
    DMA0CONbits.MODE = 0; //Le DMA a été configuré pour fonctionner en
mode continu.
    DMA0PAD = (volatile int)&ADC1BUF0;//Ce registre de lecture /
écriture contient l'adresse statique du registre de données périphérique.
    DMA0CNT = (NUMSAMP*2)-1; //Ce registre contient le compte de
transfert.
    DMA0REQ = 13;//lance une demande de DMA manuel pour lire les donnée
    DMA0STA = __builtin_dmaoffset(BufferA);
    IFS0bits.DMA0IF = 0; //Clear the DMA interrupt flag bit
    IEC0bits.DMA0IE = 1; //Set the DMA interrupt enable bit
    DMA0CONbits.CHEN = 1;// activation du convertisseur DMA
}
```

Dans notre projet, on a utilisé le DMA pour stoker les résultats de conversion(ADC).

Dans notre programme On a configuré le bit ADDMABM du Registre AD1CON1 à 1 et le bit SMPI du registre AD1CON2 à 1.

### 3.4.3.4. Universal Asynchronous Receiver Transmitter UART

L'UART (Universal Asynchronous Receiver Transmitter) constitue le cœur de tout port série. Il contrôle complètement la conversion des données natives parallèles de l'ordinateur (CPU) en format série, puis la conversion des données série en format parallèle.

Le dsPIC dispose d'un module émetteur-récepteur asynchrone universel qu'on a utilisé pour le transfert de donnée entre, l'UART du dsPIC33F, est un canal de communication asynchrone full duplex (les données circulent de façon bidirectionnelle et simultanément) [18]

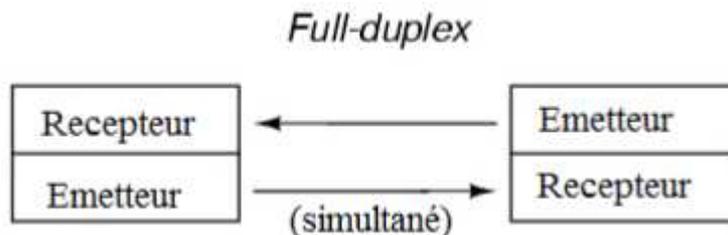


Figure 2.16 : Communication full-duplex[18].

Un schéma de principe simplifié de l'UART est représenté sur la Figure 2.16. Le module UART comprend les éléments matériels suivants [18] :

- Générateur de vitesse de transmission (exprimé en Baud)
- Emetteur et Récepteur asynchrone.

### 4. Moyen matériel et logiciel pour la programmation du dsPIC

#### 4.1. Programmeur PICkit2

Le programmeur PICkit 2 (Figure 2.17) est l'outil essentiel qui nous permettra de mettre notre code exécutable dans le microcontrôleur. Il est relié à l'ordinateur par le port USB, et au microcontrôleur par un support ICSP (In-Circuit Serial Programming). La connexion au circuit est effectuée à l'aide de six broches. La position de la broche numéro 1 est identifiée par le triangle blanc sur le PICkit. Cette broche doit être reliée à la broche numéro 1 du dsPIC33F.



Figure 2.17 : Programmeur PICkit2.

- 1 : Led de statut      2 : Bouton poussoir    3 : Dragonne de connexion  
4 : Port de connexion USB    5 : Indicateur de la pin1  
6 : Connecteur de programme

Le PICkit2 ne sert pas uniquement à transférer du code exécutable. Il sert également à déboguer le code présent dans le microcontrôleur. Le débogage, tout comme la programmation du microcontrôleur, s'effectue très aisément avec l'environnement de développement MPLAB. De plus, le PICkit2 permet d'alimenter le dsPIC33F dans le cas où ce dernier ne serait pas encore relié à une alimentation extérieure 3.3V.

#### 4.2. Interface MPLAB

Le programme de commande de la carte est écrit en langage C et est réalisé sous l'environnement de développement MPLAB comme le montre la figure suivante :

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

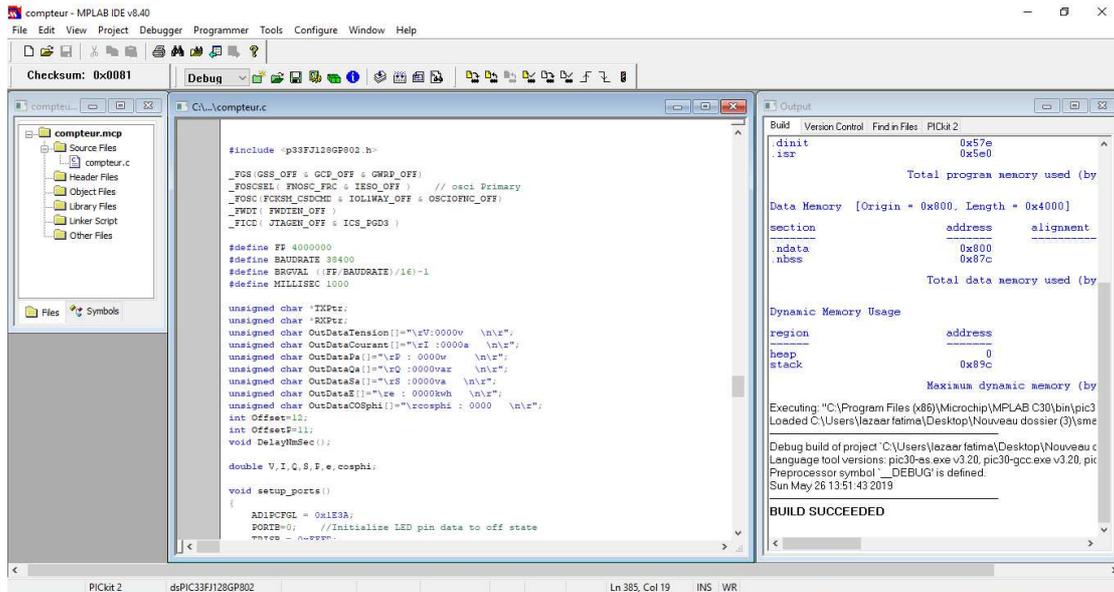


Figure 2.18 : Interface MPLAB.

Le lien entre nous et le microcontrôleur est le MPLAB. L'application qui permet d'éditer du code (en C dans notre cas), le déboguer et le charger dans le microcontrôleur. MPLAB offre aussi un outil de simulation qui permet de déboguer le code comme s'il était dans le microcontrôleur, parmi les fonctionnalités de MPLAB on peut citer:

- Une interface graphique unique à tous les outils de débogage,
- Un simulateur,
- Un programmeur (vendu séparément),
- Un In-Circuit Emulateur (vendu séparément),
- Un débogueur en circuit (vendu séparément).

Le MPLAB IDE permet de :

- Modifier les fichiers source (C ou assemblage),
- Une touche compiler ou assembler, et télécharger des outils d'émulation et des émulateurs (met à jour automatiquement toutes les informations du projet),
- Débogage en utilisant les fichiers source (en C ou en assembleur), C mixte et l'assemblage, Code de la machine.

### 4.3. Méthode de calcul de puissance

La puissance active par définition est la valeur moyenne de la puissance instantanée, donnée par l'équation suivante :

$$P = V * I * \cos(\varphi)$$

Où :  $\varphi$  représente le déphasage entre le signal de tension et de courant.

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

La puissance réactive est calculée par l'équation suivante :

$$Q = V * I * \sin(\varphi)$$

Pour calculer la puissance apparente nous utilisons l'expression :

$$S = V * I$$

Le calcul du facteur de puissance nécessite de connaître le déphasage entre la tension et le courant, ce déphasage ( $d$ ) comme le montre la figure 2.19 est le décalage en secondes entre le signal de tension et le signal de courant mesurés par les deux capteurs (courant et de tension).

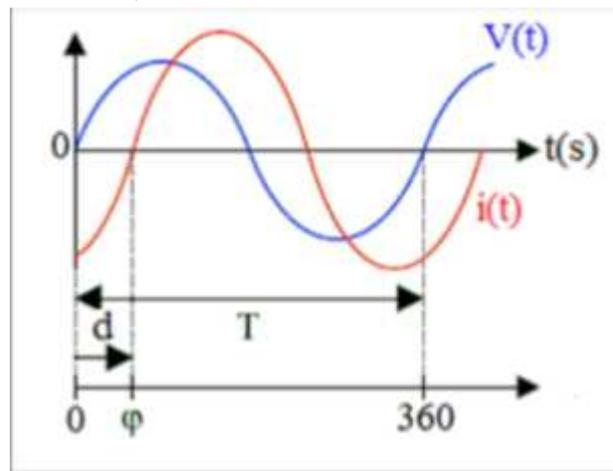


Figure 2.19 : Méthode de calcul de facteur de puissance.

Le calcul de la puissance se fait par les différentes étapes suivantes :

- Essai à vide ( $P=0$ )

La mesure de courant se fait par le capteur de courant ACS712-30A. Ce dernier permet de fournir une tension analogique entre 0 et 3.38V, qui représente l'image du courant réel.

- Essai en charge ( $P \neq 0$ )

Dans ce cas on a placé des différentes charges pour calculer la puissance (résistive, inductif, capacitif) :

- Cas d'une charge résistive:

Dans, le cas de la présence du bruit, la valeur de l'offset ne reste pas égale 2.5V à la sortie du capteur ou 757 d'après l'ADC. Ainsi, ce changement influe beaucoup sur la précision de notre capteur. Pour vérifier le bon fonctionnement, nous avons utilisé une charge résistive pour éliminer le décalage et établir l'étalonnage,

Dans ce cas nous pouvons obtenir un facteur de puissance  $\cos(\varphi)$  qui égale à 1 c'est-à-dire le déphasage est nul ( $\varphi = 0$ ), car dans le cas d'une charge résistive la tension est en phase avec le courant.

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

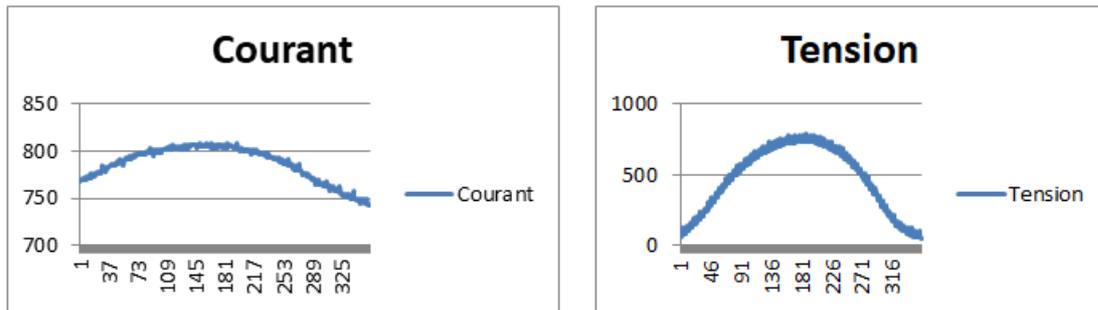


Figure 2.20 : Signaux de courant et de tension en cas de charge résistive.

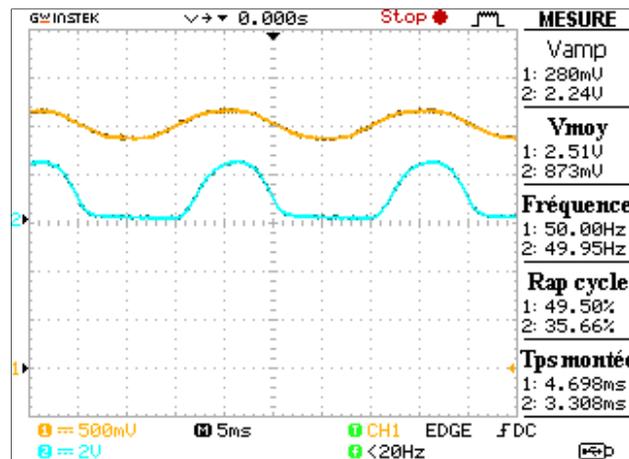


Figure 2.21: Déphasage entre le courant et la tension.

- Cas d'une charge capacitif  
Dans ce cas le signal de courant est en avance par rapport au signal de tension.
- Cas d'une charge inductif  
Dans ce cas le signal de courant est en arrière par rapport au signal de tension.

## 5. Communication et transfert de données

### 5.1. Introduction descriptive

En communication, la technologie sans fil Bluetooth est devenue très populaire et c'est l'un des domaines dans lesquels la croissance est la plus rapide. Il est donc important d'apprendre comment le **module Bluetooth HC05** s'interface avec le microcontrôleur.

De nos jours, les demandes de téléphones mobiles et de communications personnelles en bande passante sont faciles et pratiques à utiliser. La technologie Bluetooth gère le canal de communication de la partie sans fil. Le module Bluetooth peut recevoir et transmettre les données d'un système hôte à l'aide de l'interface **du contrôleur hôte**. (HCI). L'UART et l'USB sont les interfaces de contrôleur hôte les plus populaires et dans notre projet nous avons utilisé les connexions UART [20].

### 5.2. Module Bluetooth

#### 5.2.1. Définition

Bluetooth est un type de protocole de communication sans fil utilisé pour envoyer et recevoir une date entre deux appareils. L'utilisation du protocole de communication sans fil est libre. Bien que sa portée soit inférieure à celle d'autres protocoles de communication sans fil comme le WiFi et le ZigBee. Mais il est toujours adapté à de nombreuses applications de gamme inférieure. Le protocole sans fil Bluetooth se situe dans la même plage de fréquence que WiFi et ZigBee. Il fonctionne sur une fréquence de 2,41 GHz [21].

#### 5.2.2. Applications de Bluetooth

Il existe de nombreuses applications de modules Bluetooth, mais les principales applications de Bluetooth sont indiquées ci-dessous [21]:

- Dans les mobiles, les ordinateurs, les ordinateurs portables et tous les autres ordinateurs intelligents,
- Contrôleurs audio et vidéo sans fil,
- Souris et claviers sans fil,
- Casques et microphones sans fil.

Le module Bluetooth fonctionne comme maître ou esclave. Il existe sur le marché de nombreux modules Bluetooth maître / esclave, ou les deux. Le module maître Bluetooth peut envoyer ou recevoir des données d'autres modules Bluetooth. Mais l'esclave Bluetooth ne peut écouter que le module maître Bluetooth [21].

#### 5.2.3. Comment sélectionner le module Bluetooth?

La sélection du module Bluetooth dépend des facteurs suivants :

##### ✓ **Puissance maximale du module Bluetooth**

Il existe trois classes de modules Bluetooth. La plage de puissance de sortie des trois classes est donnée ci-dessous [21] :

- Classe 1 : 100 mW,
- Classe 2 : 2,5 mW,
- Classe 3 : 1 mW.

##### ✓ **Distance entre le récepteur et l'émetteur de vos deux appareils Bluetooth**

Selon la gamme d'appareils Bluetooth, il existe également trois classes de modules Bluetooth [21]:

- Classe 1: 100 mètres
- Classe 2: 10 mètres
- Classe 3: 10 centimètres

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

### ✓ Mode de communication que nous souhaitons utiliser

Cela dépend de la communication sans fil à partir de plus de trois que nous souhaitons utiliser pour l'interfaçage de notre microcontrôleur avec le module Bluetooth. De nombreux modules Bluetooth disponibles sur le marché prennent en charge plus de trois communications. Mais le nom de certains célèbres modules Bluetooth série est donné ci-dessous [21] :

- HC-06,
- HC-05,
- RN-41,
- RN-42,
- BLE4.0 hm10.

#### 5.2.4. Différence entre HC-05 VS HC-06

Ces deux modules sont expédiés et vendus avec des cartes aux formes confuses. Ils se ressemblent, ils ont la même portée (9m). Mais il y a une différence majeure entre eux [21].

HC-05	HC-06
En mode données, ce module peut fonctionner en tant que maître ou esclave.	En mode données, ce module ne peut fonctionner qu'en mode esclave.

Pour notre projet, nous choisissons le module Bluetooth HC-05 pour vous apprendre à l'interfacer avec le microcontrôleur

#### 5.2.5. Description du module Bluetooth utilisé HC-05

##### 5.2.5.1. Introduction au module Bluetooth HC 05

Commençons donc par l'introduction de base du module Bluetooth Hc 05. Il possède un circuit intégré à puce unique basé sur la technologie CMOS. Il a une puissance de transmission jusqu'à +4 dBm. Il possède de larges tensions de fonctionnement comprises entre 3,6 et 6 volts. La vitesse de transmission par défaut de cet appareil est 9600. Mais vous pouvez le modifier selon vos besoins en utilisant des commandes AT. J'expliquerai plus tard comment utiliser les commandes AT de ce module Bluetooth. HC 05 travaille sur la communication série. Il communique avec le microcontrôleur via la communication UART. Vous devez donc savoir comment utiliser la communication série du microcontrôleur. Si vous connaissez la programmation de communication série du contrôleur pic33fj182GP802, vous pouvez facilement interfacer ce module et écrire un programme en C [21].

### 5.2.5.2. Présentation du module Bluetooth HC-05

Le Bluetooth est une norme de technologie sans fil pour échanger des données sur de courtes distances (en utilisant des ondes radio UHF de courte longueur d'onde dans la bande ISM de 2,4 à 2,485 GHz) à partir d'appareils fixes et mobiles, et la construction de réseaux personnels (PAN). La distance de communication est d'environ 9 mètres (30 pieds). Nous utiliserons le module Bluetooth HC-05 qui communique avec les microcontrôleurs via le bus série UART [24].

### 5.2.5.3. Description de l'appareil

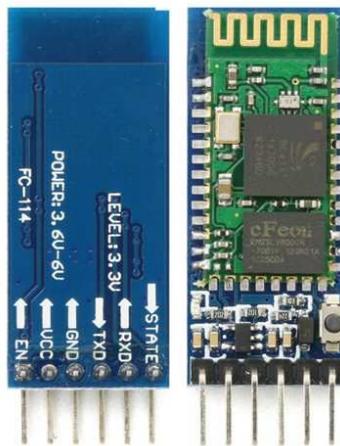


Figure 2.22: Module Bluetooth HC-05 [21].

Ce module est basé sur la puce Radio Bluetooth Radio 2,4 GHz BC417 de Cambridge. Il s'agit en fait d'une puce complexe qui utilise une mémoire flash externe de 8 Mbit. Vous pouvez trouver plus d'informations dans la fiche technique associée. Cependant, il est assez facile de configurer et d'interfacer tout système de micro-ordinateur via l'UART série. Le HC-05 peut être défini comme maître ou esclave. Ces petits modules fonctionnent sur une alimentation de 3,3 V avec des niveaux de signal série de 3,3 V. Ils n'ont pas de broches et sont généralement soudés à une carte plus grande [21].

Les modules Bluetooth ont besoin de protocoles de communication pour se connecter à d'autres appareils. Par exemple, dans ce projet, notre objectif principal est l'interfaçage du module Bluetooth avec un microcontrôleur, un microcontrôleur pouvant communiquer avec un périphérique Bluetooth en suivant des communications sans fils pour envoyer ou recevoir des données d'un autre périphérique Bluetooth [21] :

- UART,
- SPI,
- USB.

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

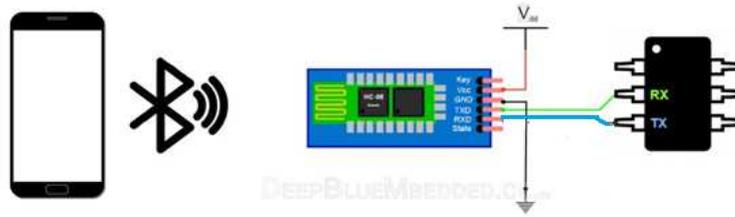


Figure 2.23:Diagramme de connexion ( module Bluetooth avec un microcontrôleur)[21].

Nous avons utilisé uniquement les broches Vcc, masse, TX et RX de notre module Bluetooth HC 05 pour l'interfacer avec le microcontrôleur pic. Une explication de toutes ces broches est donnée ci-dessous [21]:

- Broche Vcc : broche d'alimentation (doit être connectée à 3.3V),
- Broche de terre : broche d'alimentation (doit être connectée à la masse),
- TX pin : broche de transmission de données. Nous avons connecté la broche RX du microcontrôleur à cette broche,
- RX pin : broche du réception de données du microcontrôleur, nous avons connecté la broche TX du microcontrôleur à cette broche.

### 5.2.5.4. Modes de fonctionnement

Le module Bluetooth HC-05 a deux modes de fonctionnement différents :

- Mode de données
- Mode de commande AT

Le mode par défaut est le mode données avec les paramètres par défaut suivants pour la communication.

Nom du dispositif: HC-05,  
Mot de passe: 1234 ou 0000,  
Baud Rate: 38400.

#### ✓ Mode données

En mode données, le module Bluetooth HC-05 peut être configuré pour fonctionner dans l'un des modes suivants [maître-esclave] où il peut transmettre ou recevoir des données vers / depuis un autre module / périphérique Bluetooth [21].

#### ✓ Mode de commande AT

Dans de nombreuses situations, nous devons peut-être modifier certaines des configurations par défaut ou leurs valeurs. C'est ce que nous utilisons pour le mode de commande. Il y a plusieurs façons de passer en mode commande:

- 1- Connectez la broche KEY en haut avant d'alimenter le module. Ceci mettra le module en mode commande à 38400 bauds. Il s'agit du débit en bauds par défaut pour le mode commande et nécessaire si vous ne connaissez pas le

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

débit en bauds défini par le module. Vous pouvez utiliser un moniteur série pour faire le travail.

- 2- Mettez le module sous tension, puis tirez la broche KEY vers le haut. Ceci entrera en mode commande au débit en bauds actuellement configuré. Cela est utile si vous souhaitez envoyer des commandes AT depuis un microcontrôleur, car la broche KEY peut être contrôlée à l'aide de l'une des broches du microcontrôleur. Mais, vous devez connaître le débit en bauds actuellement configuré pour le mode de commande AT.

Les commandes sont envoyées au module dans UPPERCASE et se terminent par une paire CR / LF [21].

Quelques exemples de commande AT:

AT (commande de test AT. Devrait répondre avec OK)

A + VERSION? (montre la version du firmware)

AT + UART = 38400 (Réglez le débit en bauds sur 38400, 1 bit d'arrêt, pas de parité).

### 5.2.6. Communication série

#### 5.2.6.1. Introduction descriptive

Dans les applications de systèmes embarqués, de télécommunication et de transmission de données, la communication série (la plus utilisée dans les systèmes embarqués) est le processus d'envoi de données bit par bit (bit par bit) séquentiellement sur le bus série. Il faut un cycle d'horloge complet pour transférer chaque bit d'un bout à l'autre. Inversement, on sait que la communication parallèle consiste à envoyer plusieurs bits, même des octets, dans un seul cycle d'horloge. Cependant, même si vous transférez moins de données par cycle avec une transmission série, vous pouvez le faire à des fréquences beaucoup plus élevées, ce qui entraîne des taux de transfert nets plus élevés que ceux de la communication parallèle [21].

#### 5.2.6.2. Méthodologie suivie pour l'utilisation du module UART

Le récepteur / émetteur asynchrone universel ou UART représente le circuit matériel (module) utilisé pour la communication série. UART est vendu / livré en tant que circuit intégré autonome (IC) ou en tant que module interne au sein de microcontrôleurs. Dans ce tutoriel, nous nous intéressons au module UART interne de PIC Microcontrôleurs [21].

#### ✓ Configuration et initialisation d'UART

Le dsPIC dispose d'un module émetteur-récepteur asynchrone universel qu'on a utilisé pour le transfert de donnée entre, l'UART du dsPIC33F, est un canal de

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

communication asynchrone full duplex (les données circulent de façon bidirectionnelle et simultanément) [21].

Le module UART comprend les éléments matériels suivants :

- Générateur de vitesse de transmission (exprimé en Baud),
- Emetteur asynchrone,
- Récepteur asynchrone.

L'UART est un composant clé. Le format de données de l'UART est composé de 11 bits. Il s'agit d'un bit de départ ,8 bits de données, un bit de parité et un bit d'arrêt. Dans notre programme l'UART est configuré à transmettre 8 bits avec 1 seul bits de stop

```
void InitUART1()  
{  
    U1MODEbits.USIDL= 0;  
    U1MODEbits.IREN= 0;  
    U1MODEbits.RTSMD= 1;  
    U1MODEbits.UEN= 0;  
    U1MODEbits.WAKE= 0;  
    U1MODEbits.URXINV= 0;  
    U1MODEbits.STSEL = 0;  
    U1MODEbits.PDSEL = 0;  
    U1MODEbits.ABAUD = 0;  
    U1MODEbits.BRGH = 0;  
    U1BRG = BRGVAL;  
    U1STAbits.UTXISEL0 = 0;  
    U1STAbits.UTXISEL1 = 0;  
    U1MODEbits.UARTEN = 1;  
    U1STAbits.UTXEN = 1;  
}
```

### ✓ Déclaration des variables nécessaires

Nous devons d'abord déclarer deux chaînes de caractères (dans notre cas TXPtr et RXPtr pour envoyer et recevoir des caractères respectivement), puis réservez un endroit pour remplir les données calculées précédemment.

```
unsigned char *TXPtr;  
unsigned char *RXPtr;  
unsigned char OutDataTension[]="\r salem \n\r";  
unsigned char OutDataCourant[]="\rI :0000a \n\r";  
unsigned char OutDataPa[]="\rP : 0000w \n\r";  
unsigned char OutDataQa[]="\rQ :0000var \n\r";  
unsigned char OutDataSa[]="\rS :0000va \n\r";  
unsigned char OutDatae[]="\rE : 0000kwh \n\r";  
unsigned char OutDataCOSphi[]="\rcosfi : 0000 \n\r";
```

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

### ✓ Choix des pins pour transmission et réception

L'avantage de ce type de microcontrôleur est la possibilité de configurer ces pins en fonction de notre besoin. Cela, nous aide à la conception d'un routage efficace. Dans notre cas, on a configuré la pin RP3 correspondant à l'entrée analogique RB3 comme output TX (transmission des données), et la pin RP2 correspondant à l'entrée analogique RB2 comme input RX (recevoir des données). La fonction ci-dessous vous montre la méthode de configuration.

```
void InitMAPPING()
{
    //for unlocking PIN MAPPING
    __builtin_write_OSCCONL(OSCCON & ~(1<<6));

    //PIN MAPPING (Setting RX & TX pins in dsPIC33FJ128GP802)
    RPOR1bits.RP3R = 3; //RP3 as TX pin (output mapping)
    RPINR18bits.U1RXR = 2; // RP2 as RX pin (input pin mapping)

    //locking PIN MAPPING
    __builtin_write_OSCCONL(OSCCON | (1<<6));
}
```

### ✓ Conversion d'un nombre en chaîne de caractère

Pour transmettre un nombre, il doit impérativement le convertir en chaîne de caractère. La fonction suivante vous montre la procédure à suivre pour convertir un nombre à 04 chiffres

```
// Convertit un nombre en "4 chars décimaux" xxxx
// maximum : 9999
// et les sort sur la table en memoire
//-----
inline void ConvDec(int Var, int tablePos, unsigned char * table)
{
    unsigned int k; unsigned char c;

    k = Var;
    c = k/1000;
    if (c > 0)
        k = k - c*1000;
    table[tablePos] = (c + 0x30);
    c = k/100;
    if (c > 0)
        k = k - c*100;
    table[tablePos+1] = (c + 0x30);
    c = k/10;
    if (c > 0)
        k = k - c*10;
    table[tablePos+2] = (c + 0x30);
    table[tablePos+3] = (char)(k + 0x30);
}
```

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

### ✓ Transmission des données

- On utilise la routine SendMsg du programme. Dans cette routine, on a configuré le registre U1TXREG comme registres de transmission (TXptr) pour envoyer des data,
- Le cœur de l'émetteur est le registre de transmission (UxTSR). Le registre à décalage obtient ses données du tampon de transmission FIFO, UxTXREG. Le registre UxTXREG est chargé avec les données dans le logiciel. Le registre UxTSR n'est chargé que lorsque le bit d'arrêt a été transmis depuis le chargement précédent. Dès que le bit d'arrêt est transmis, l'UxTSR est chargé avec les nouvelles données du registre UxTXREG (si disponible),
- Le registre UxTSR n'est pas mappé dans la mémoire de données, il n'est donc pas disponible pour l'utilisateur,
- L'octet à envoyer est placé dans U1TXREG (Transmit Register),
- Même lorsque que les interruptions sont interdites, on peut utiliser les flags pour connaître l'état (vide ou plein) du registre,

```
void SendMsg()  
{  
    while (*TXPtr)  
    {  
        while (U1STAbits.UTXBF);  
        U1TXREG = *TXPtr++;  
    }  
}
```

- Après, on utilise la routine SendData. Dans ce programme, on a converti la valeur de l'offset dans OutData en décimale. Après, si l'octet (Outdata) est disponible dans la pin de transmission (TXptr=OutData), à cet instant, notre microcontrôleur envoie le message (Outdata) par TXptr.

```
// Converti en envoi les infos debug sur 1 UART
```

```
void SendData()  
{  
    ConvDec( V,Offset, OutDataTension);  
    TXPtr =OutDataTension;  
    SendMsg();  
    ConvDec( I,Offset, OutDataCourant);  
    TXPtr = OutDataCourant;  
    SendMsg();  
    ConvDec( P,Offset, OutDataPa);  
    TXPtr = OutDataPa;  
    SendMsg();  
    ConvDec( Q,Offset, OutDataQa);  
    TXPtr = OutDataQa;  
    SendMsg();  
    ConvDec( S,Offset, OutDataSa);  
    TXPtr = OutDataSa;  
    SendMsg();  
    ConvDec( E,Offset, OutDatae);  
    TXPtr = OutDatae;
```

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

```
SendMsg();  
ConvDec(cosfi, OffsetP, OutDataCOSphi);  
TXPtr = OutDataCOSphi;  
SendMsg();  
}
```

### 5.3. Test de fonctionnalité de notre système de communication

Pour tester notre système de communication UART-Bluetooth HC-05, nous avons utilisé l'application open source PYTTY qui est très efficace et simple à utiliser. Cette application sert à visualiser les données transférées par notre carte de commande à travers le HC-05 pour les visualiser sur l'écran de notre ordinateur à travers son Bluetooth [22].

#### 5.3.1. Notion sur l'application PUTTY.exe

PuTTY est un émulateur de terminal open source, utilisé pour se connecter à des serveurs SSH, Telnet et similaires pour une administration à distance. Développé par Simon Tatham, qui a publié la première version en 1999, PuTTY a été mis à niveau à plusieurs reprises au fil des ans pour ajouter de nouvelles fonctionnalités et corriger les bugs. PuTTY vous permet également de vous connecter à un périphérique connecté au port série de l'ordinateur, en envoyant des commandes dactylographiées au périphérique série et en affichant la réponse. Ceci est utile pour le débogage de matériel réseau équipé de ports série ou pour la connexion à des microcontrôleurs et à d'autres périphériques électroniques [22]. Vous pouvez télécharger de l'émulateur de terminal du site suivant :

<http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>.

Après avoir installé et ouvert le logiciel, vous verrez le tableau de bord de configuration où vous définissez comment vous voulez que le logiciel s'exécute. Vous avez le choix entre Raw, Telnet, Rlogin, SSH et Serial. Sélectionnez "Série" et, dans le champ situé sous "Ligne série", indiquez votre numéro de port COM. Par exemple, COM3. Cela fait, cliquez sur "Ouvrir". Vous allez maintenant voir l'interface du moniteur pour le logiciel PuTTY [22].

##### 5.3.1.1. Activation de la communication Bluetooth

Avant l'exécution de l'émulateur PUTTY, vous devez impérativement activer le Bluetooth de l'ordinateur où vous allez visualiser les données. Après sélection du module Bluetooth (HC-05), entrer le code correspondant à ce module (1234 dans notre cas).

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

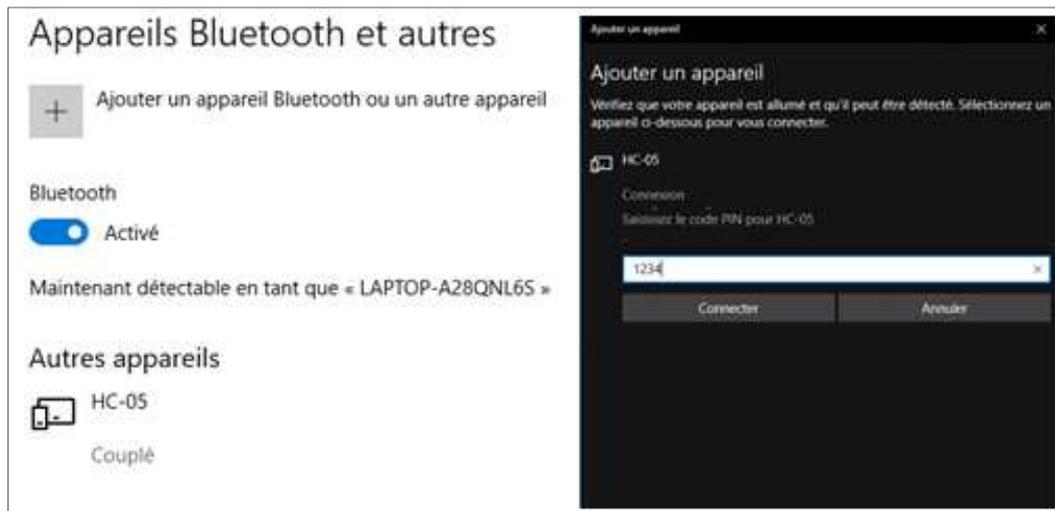


Figure 2.24:Couplage de module bluetooth HC-05 avec le bluetooth du PC.

Pour utiliser l'application PUTTY, vous devez connaître à quel port est relié le Bluetooth de votre ordinateur. Pour cela, il faut suivre la procédure donnée dans la figure ci-dessous.

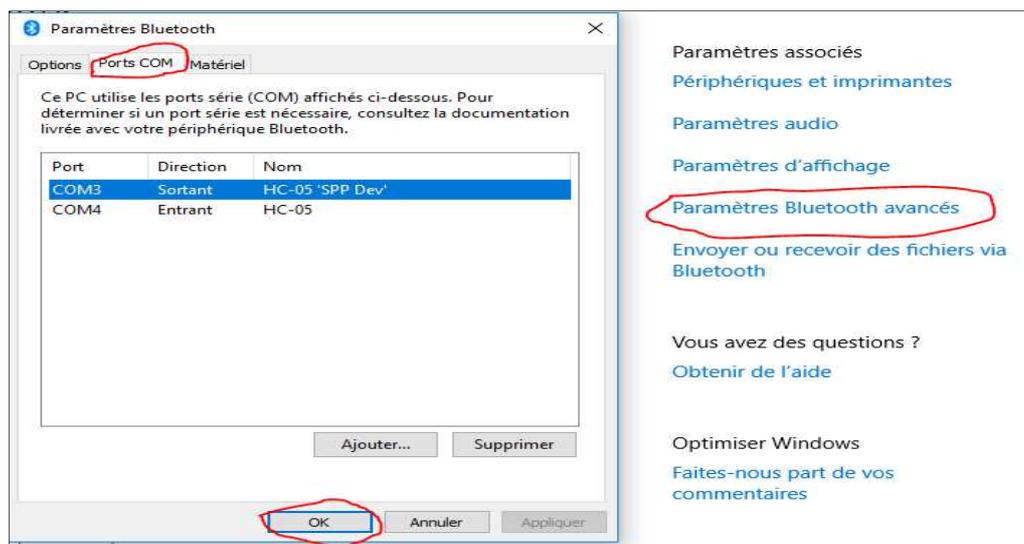


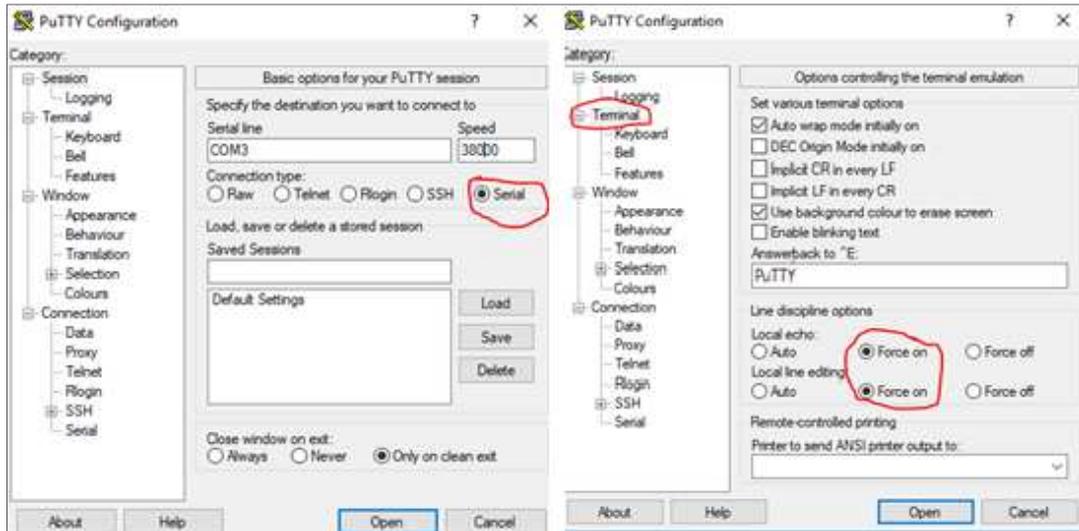
Figure 2.25:Confugiration de port serie (COM) de notre module bluetooth HC-05.

### 5.3.1.2. Configuration du menu PuTTY

Toute la procédure est représentée sur la Figure 2.26. Il suffit de la suivre.

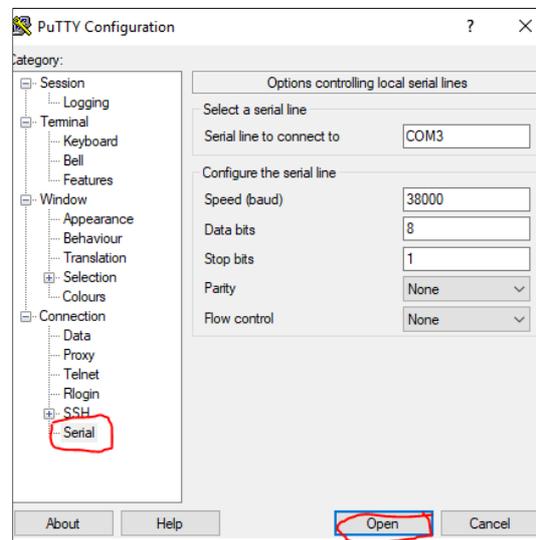
- Sous Type de connexion, sélectionnez Série,
- Dans le champ Ligne série, entrer le numéro COM où votre carte est reconnue (dans notre exemple COM3) et préciser le BaudRate utilisé dans le champ 'Speed'. Pour le HC-05, la vitesse de transmission est de 38400 bits/s,
- Faites la sélection comme le montre la Figure ci-dessous et cliquer sur 'Open'.

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent



(a)

(b)

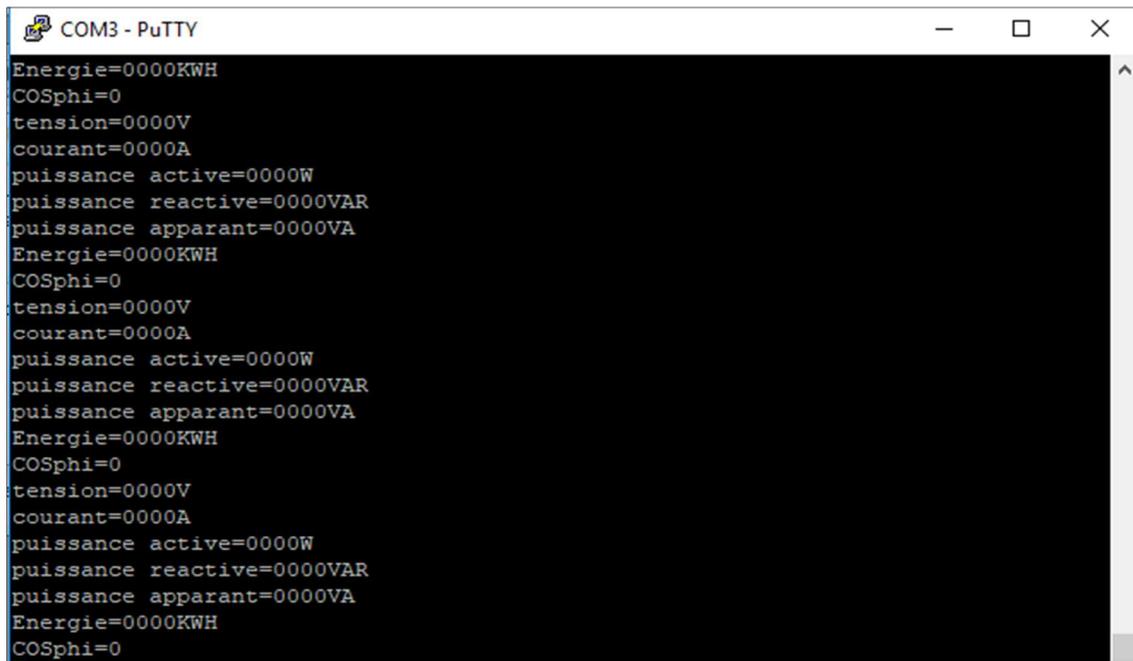


(c)

Figure 2.26: Configuration de l'émulateur de terminal PuTTY.

D'après la simulation de notre programme, on a obtenu le résultat désiré suivant sur le moniteur PuTTY.

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent



```
COM3 - PuTTY
Energie=0000KWH
COSphi=0
tension=0000V
courant=0000A
puissance active=0000W
puissance reactive=0000VAR
puissance apparant=0000VA
Energie=0000KWH
COSphi=0
tension=0000V
courant=0000A
puissance active=0000W
puissance reactive=0000VAR
puissance apparant=0000VA
Energie=0000KWH
COSphi=0
tension=0000V
courant=0000A
puissance active=0000W
puissance reactive=0000VAR
puissance apparant=0000VA
Energie=0000KWH
COSphi=0
tension=0000V
courant=0000A
puissance active=0000W
puissance reactive=0000VAR
puissance apparant=0000VA
Energie=0000KWH
COSphi=0
tension=0000V
courant=0000A
puissance active=0000W
puissance reactive=0000VAR
puissance apparant=0000VA
Energie=0000KWH
COSphi=0
```

Figure 2.27 : Résultat obtenu par le moniteur PYTTY.

### ✓ Transmission et réception des données

Pour faire la transmission et la réception des données, on doit ajouter des tests dans la boucle 'while'

```
while (1) {
    if(U1RXREG=='1') {
        TXPtr =OutDataTension ;
        SendMsg();
        DelayNmSec(1000);
    }
    if(U1RXREG=='2') {
        TXPtr = OutDataCourant;
        SendMsg();
        DelayNmSec(1000);
    }
    if(U1RXREG=='3') {
        TXPtr = OutDataPa;
        SendMsg();
        DelayNmSec(1000);
    }
    if(U1RXREG=='4') {
        TXPtr = OutDataQa;
        SendMsg();
        DelayNmSec(1000);
    }
    if(U1RXREG=='5') {
        SendMsg();
        TXPtr = OutDataE;
        DelayNmSec(1000);
    }
    if(U1RXREG=='6') {
        TXPtr = OutDataCOSphi;
        SendMsg();
        DelayNmSec(1000);
    }
}
```

## Chapitre 2 : Description Générale de notre prototype de compteur intelligent

- Le cœur du récepteur est le registre à décalage de réception (série) (UxRSR). Les données sont reçues sur la broche UxRX et sont envoyées au bloc de récupération de données. Le bloc de récupération de données fonctionne à 16 fois le débit en bauds, alors que le sélecteur série en réception principale fonctionne au débit en bauds. Après l'échantillonnage de la broche UxRX pour le bit d'arrêt, les données reçues dans UxRSR sont transférées vers la FIFO de réception (si elle est vide),
- L'octet reçu est placé dans le registre U1RxREG (Receive Register) et l'octet à envoyer est placé dans U1TXREG (Transmit Register). Même, lorsque les interruptions sont interdites, on peut utiliser les flags pour connaître l'état (vide ou plein) des registres U1RxREG et U1TXREG,
- Pour savoir si un octet est disponible, il suffit simplement de regarder le flag associé à U1RxREG (réception d'un octet),
- Donc, dans notre programme, on a configuré le registre U1TXREG comme registres de réception. Quand ce registre reçoit un caractère, le registre de transmission 'U1TXREG' envoie les données chaque 1000 ms.

Le but du programme est de pouvoir communiquer d'un téléphone Android vers un microcontrôleur avec module HC-05 dans les deux sens.

Donc, pour faire ce travail, on doit créer une application Android pour commander un module Bluetooth HC-05 avec un microcontrôleur (voir chapitre3).

## 6. Conclusion

Dans ce chapitre, nous avons présenté tous les composants nécessaires pour réaliser un compteur intelligent avec l'explication de fonctionnement de chaque circuit (alimentation, passage par 0, capteur de tension ...).

Nous avons intégré aussi un système de communication en insérant un module Bluetooth ce qui permet de faciliter l'envoi et la réception des données.

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 communiquant et les stocker sur tout appareil fonctionnant sous Android.

## **Chapitre 3 Application mobile et réalisation finale de notre prototype**

### 1. Introduction

#### 1.1. Définition du système d'exploitation mobile 'Android'

Android est le système d'exploitation mobile créé par Google. Il équipe la majorité des téléphones portables du moment (smartphones). Son principal concurrent est Apple avec l'iPhone. Android est un système vous permettant de personnaliser votre téléphone, télécharger des applications (navigateur Internet, GPS, Facebook...), de plus il équipe également les tablettes tactiles.

#### 1.2. Avantages d'Android

- Disponibles sur un grand nombre de modèles de téléphones la diversité et le volume des ventes des Androphones étant sans limites,
- Une personnalisation et un choix presque illimités,
- Un téléphone globalement plus rapide puisqu'il est dépossédé des surcouches imposées par les constructeurs,
- Des possibilités de sur-cadencement ou de sous-cadencement des composants afin d'obtenir un téléphone plus réactif, plus puissant ou plus autonome que la version originale,
- Certains téléphones ne peuvent pas être mis à jour. Les versions alternatives permettent d'utiliser une version plus récente d'Android afin de rester à jour.

### 2. Environnement de développement MIT App Inventor

#### 2.1. Définition

L'outil de conception MIT App Inventor 2 permet d'obtenir une application Android en moins d'une heure. Plus de 3 millions de personnes ont été séduites par cette interface graphique simple, qui invite à glisser-déposer des blocs interactifs pour créer une application basique.

#### 2.2. Préparation du matériel

Pour le développement de notre application, on a besoin :

- D'un ordinateur (PC ou Mac) avec une connexion wifi, et un navigateur internet (Chrome, Firefox ou Safari – Pas Internet Explorer)
- D'un téléphone ou une tablette sous Android, connecté au même réseau wifi que l'ordinateur (version Android « Gingerbread » 2.3 ou plus récents)
- D'une adresse Gmail (pour vous connecter au logiciel en ligne MIT App Inventor 2).

## Chapitre 3 : Application mobile et réalisation finale de notre prototype

- D'une application gratuite à télécharger sur le Google Play Store de votre téléphone ou de votre tablette, pour vous permettre de tester en direct la création de votre application puis de l'installer une fois finie, et qui s'appelle 'MIT AI2 Companion' (Figure 3.1).

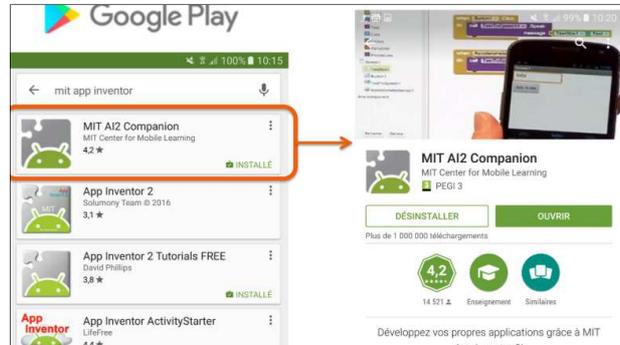


Figure 3.1 : Téléchargement d'application (MIT AI2 Companion).

### 2.3. Création du projet d'une application mobile

Tout d'abord, on doit démarrer le projet d'application mobile sur le site officiel de MIT App Inventor 2.

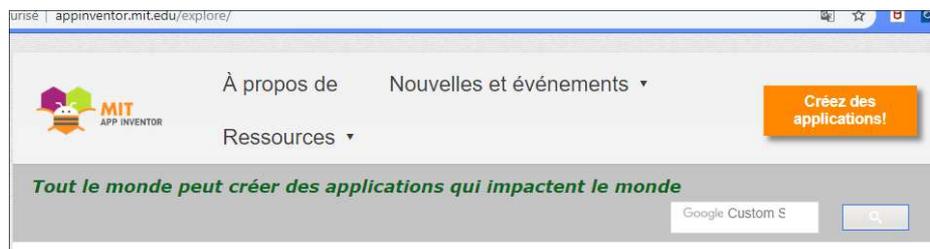
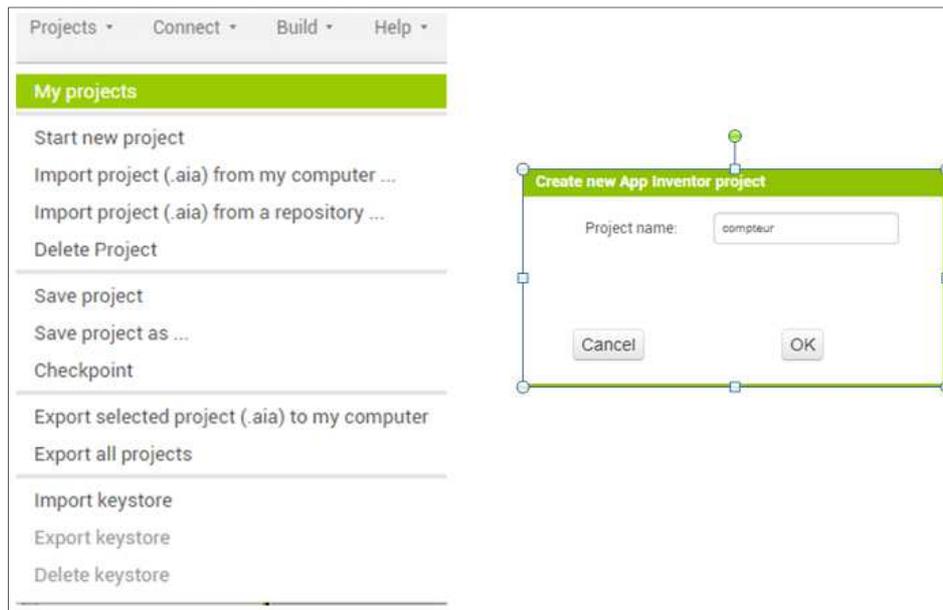


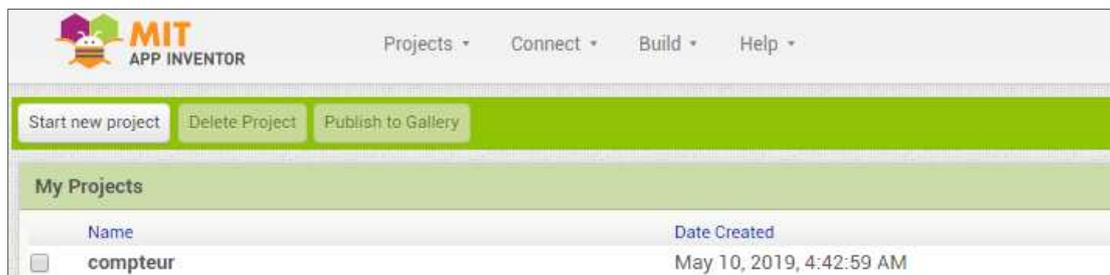
Figure 3.2 : Site officiel de MIT App Inventor 2.

- ✓ Avec MIT App Inventor 2 (AI2), on n'a pas besoin de perdre du temps à créer un compte utilisateur. AI2 se synchronise avec votre compte Gmail et en quelques secondes vous accédez à l'interface de création.
- ✓ Dirigez-vous ensuite dans le menu « Projects » pour créer un nouveau projet App Inventor (Figure 3.3 (a)). Créer un nouveau projet ,et nommé 'compteur' ((Figure 3.3 (b)).

## Chapitre 3 : Application mobile et réalisation finale de notre prototype



(a)



(b)

Figure 3.3 : Création d'un nouveau projet nommé 'compteur'.

Une fois le nom de votre application définie, nous pouvons passer au développement de notre application.

### 2.4. Développement d'application Android

Pour développer notre application, il est impératif de passer par deux phases : phase de conception pour faire le design de notre application utilisant la page 'Designer' (Figure 3.4) et phase de programmation des différents objets sélectionnés dans notre application utilisant la page 'Block' (Figure 3.10) du même environnement de développement de MIT App Inventor 2 (AI2).

## Chapitre 3 : Application mobile et réalisation finale de notre prototype

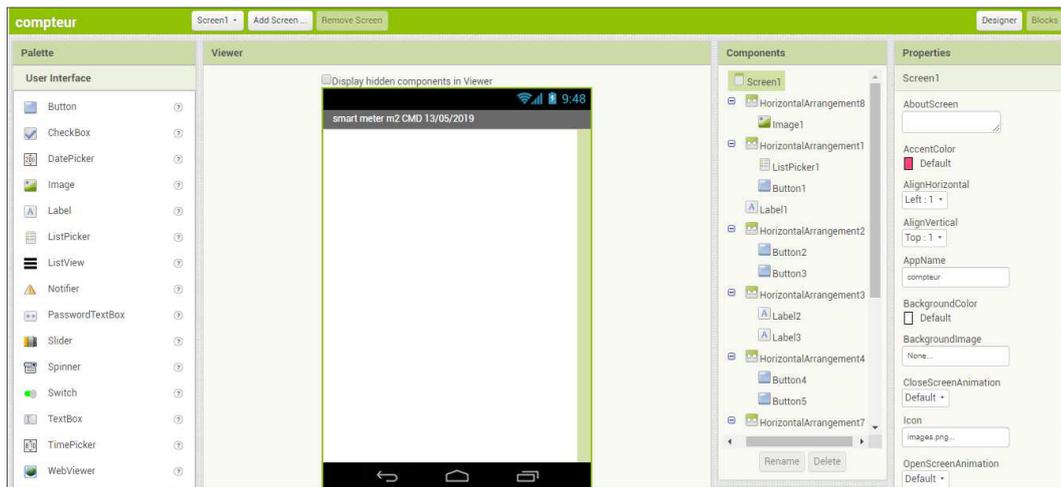


Figure 3.4 : Environnement de développement : Page ‘Designer’.

### 2.4.1. Conception d’Application AI2

Pour créer le design de notre application, nous utilisons les objets disponibles sous les onglets du menu déroulant à gauche de l’écran en modifier leurs paramètres qui apparaissent à droit de l’écran. Les composants utilisés dans notre application :

**Bouton** : a la possibilité de détecter les clics. Beaucoup d’aspect de son apparence peuvent être changé, ainsi que la possibilité qu’il soit cliquable. (Activé), peut être changé dans la partie conception ou dans la partie bloc,

**Image** : Composant pour afficher les images. La photo à afficher, et les autres aspects de l’apparence de l’image, peuvent être spécifiés dans la partie Conception ou dans la partie bloc,

**Label** : Un label affiche un morceau de texte, qui est spécifié avec la propriété Texte. Les autres propriétés, qui peuvent être fixées dans la partie Designer ou bloc, contrôlent l’apparence et l’emplacement du texte,

**Horloge** : Composant non visible qui fournit l’horloge de l’appareil, un chronomètre, et réalise le calcul du temps,

**Arrangement horizontal** : Un élément de formatage dans lequel on place les composants qui doivent être affichés de gauche à droite. Si vous voulez avoir des composants affichés l’un sur l’autre, utilisez Arrangement vertical,

**Sélecteur de liste** : Un bouton qui, lorsqu’on clique, affiche à l’utilisateur une liste de textes à choisir. Les textes peuvent être spécifiés dans la

## Chapitre 3 : Application mobile et réalisation finale de notre prototype

partie 'Designer' ou la partie bloc en fixant la propriété 'Éléments de chaîne' (par exemple, *choix 1*, *choix 2*, *choix 3*) ou en fixant la propriété 'Éléments' à une liste dans la partie bloc.

**BluetoothClient** : c'est un composant de connectivité et de communication.

Comme 1<sup>ère</sup> étape de conception, on commence notre application en sélectionnant un objet 'HorizontalArrangement' (se trouvant dans 'Layout du Designer (Figure 3.5)) qui permet d'aligner horizontalement les objets suivants :

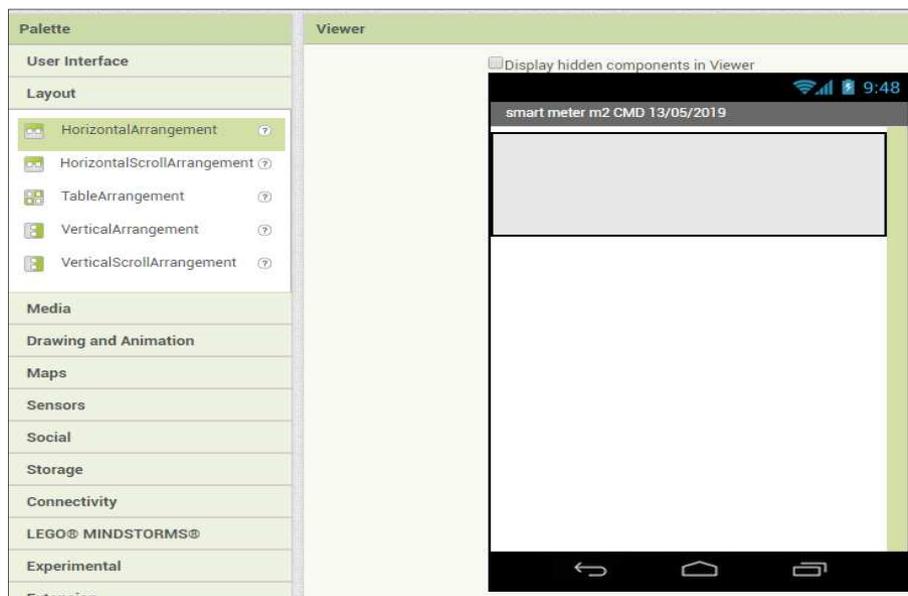
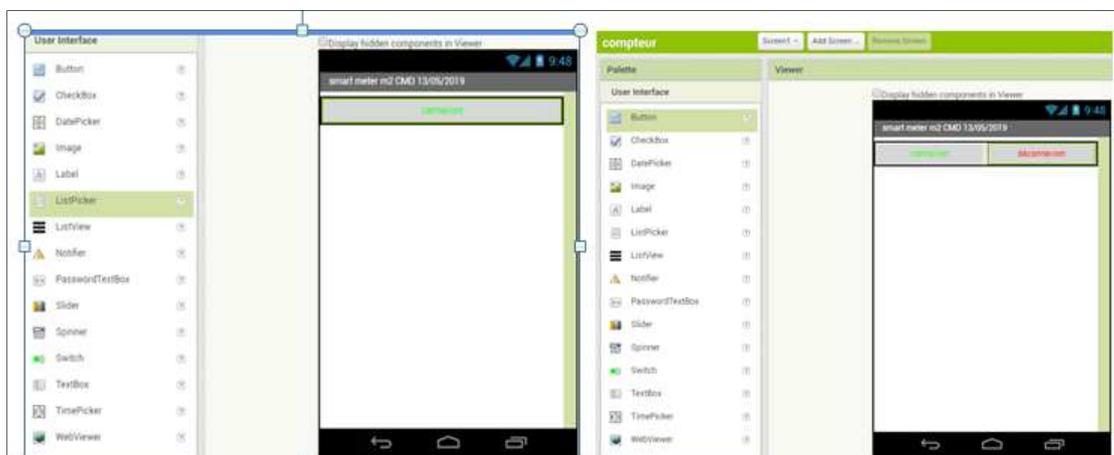


Figure 3.5 : Méthode de sélection d'un objet 'HorizontalArrangement'.

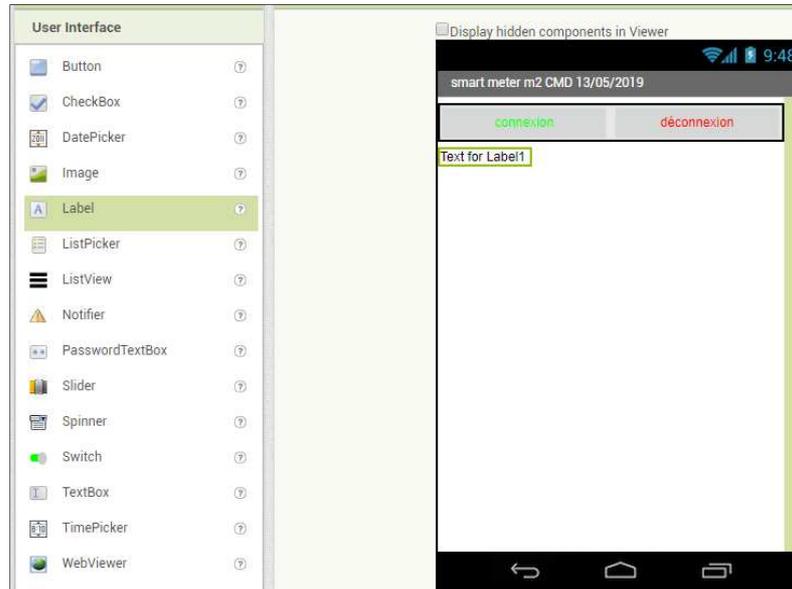
a) (Figure3.6 (a)) : Dans l'objet 'HorizontalArrangement', placer ListPicker (connexion) puis un label (déconnexion) qui servira à gérer la connexion Bluetooth.



(a)

### Chapitre 3 : Application mobile et réalisation finale de notre prototype

b) (Figure3.6 (b)) : Ajouter ensuite un label pour afficher un texte d'étiquette (facultatif) en tant qu'état relatif à la connexion Bluetooth.



(b)

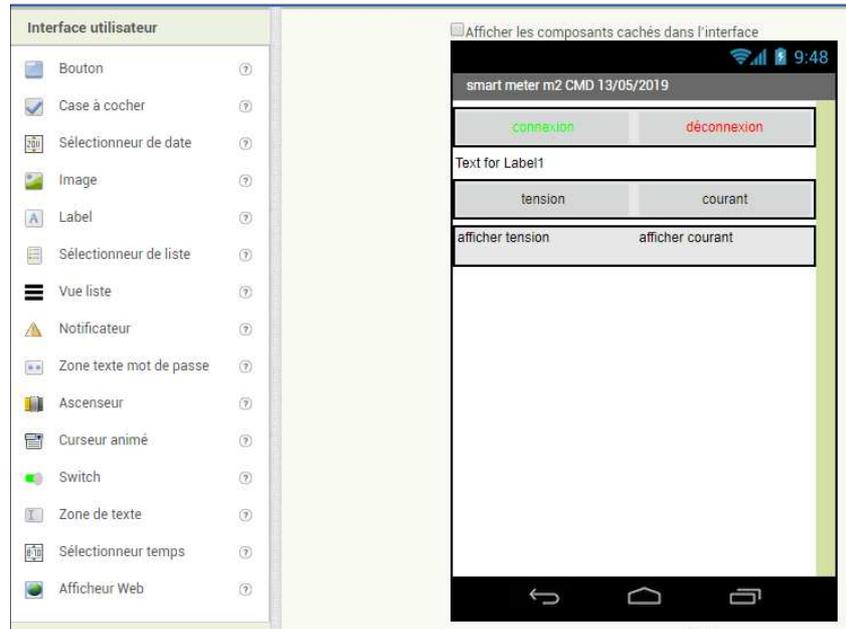
c) (Figure3.6 (c)) : Ajouter un autre objet 'HorizontalArrangement' et placer dans cet objet deux boutons (courant et tension).



(c)

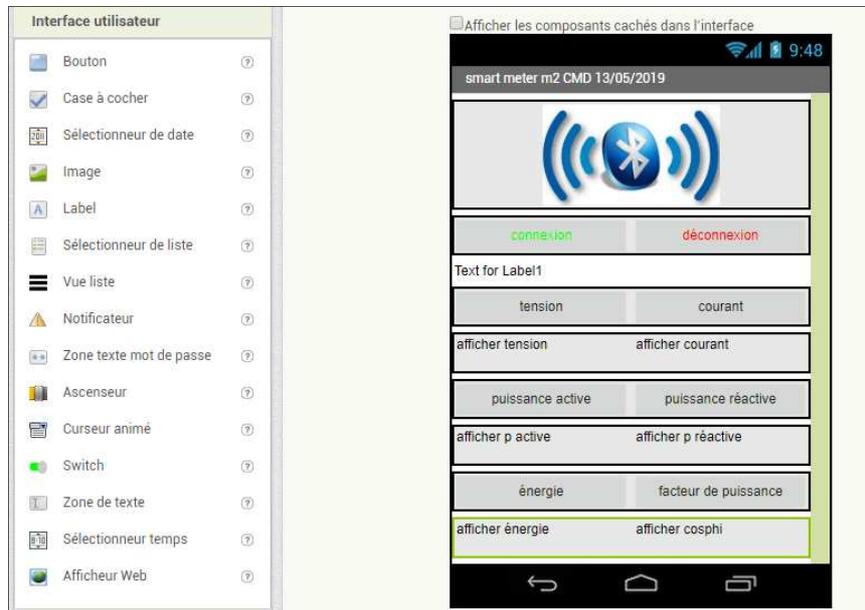
d) (Figure3.6 (d)) : Ensuite ajouter un troisième objet 'HorizontalArrangement' et placer dans cette objet deux labels pour afficher les données reçues par le périphérique connecté (affichage du courant et de la tension par exemple).

### Chapitre 3 : Application mobile et réalisation finale de notre prototype



(d)

e) (Figure3.6 (e)) : Par la même méthode on a ajouté des boutons (puissance active, puissance réactive, énergie et facteur de puissance), et des labels pour afficher les données qui reçoit par périphérique connecté (la puissance active et réactive et l'énergie et le facteur de puissance).



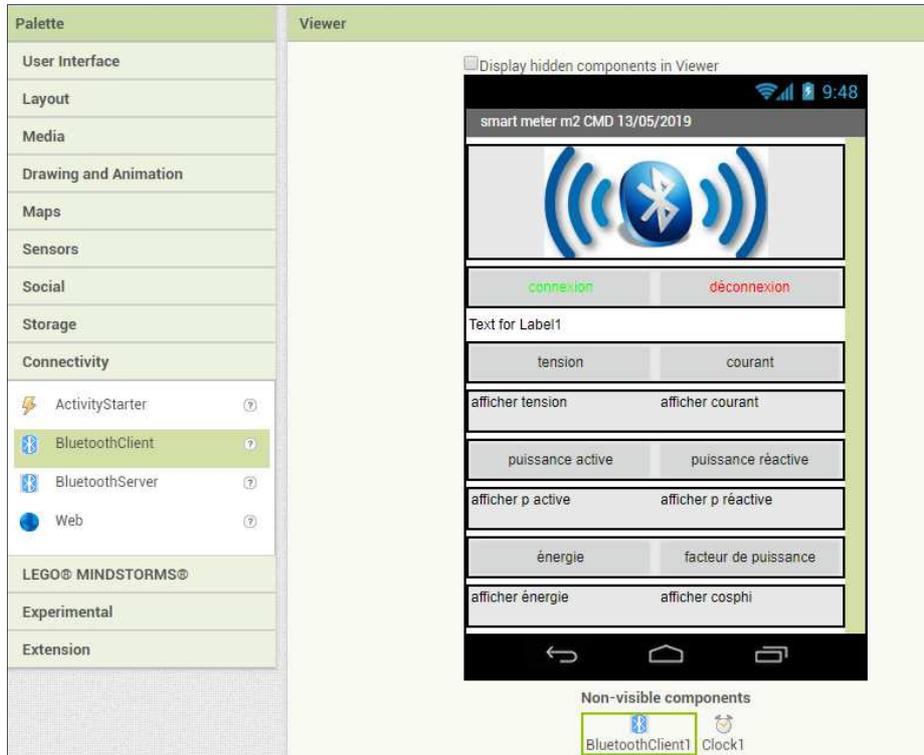
(e)

Figure 3.6 : Conception d'Application AI2. Ajout : Bouton, label, listPicker....

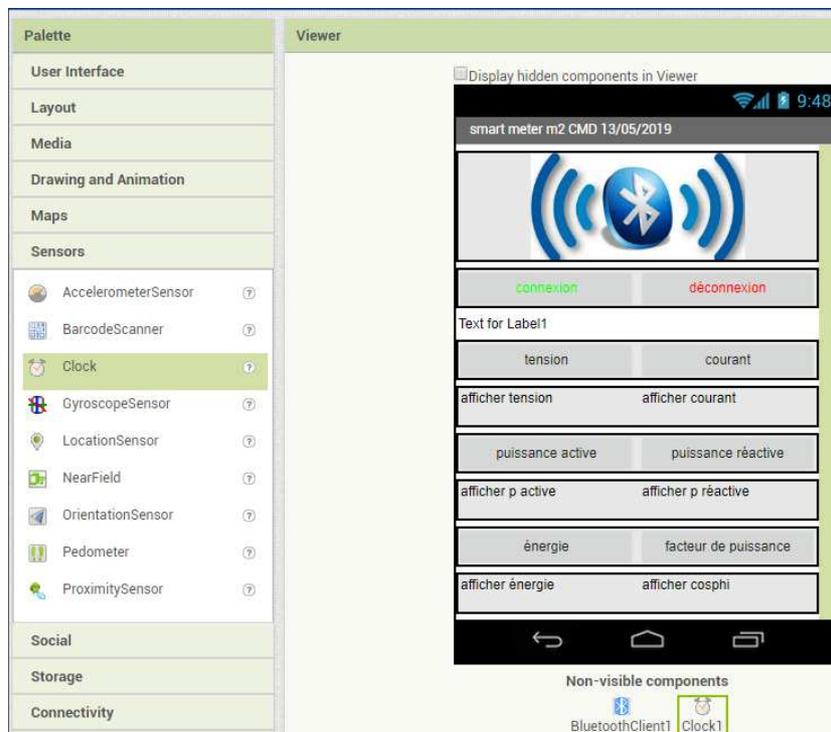
Ensuite dans 'Connectivity' (Figure 3.7 (a)), on a ajouté un objet 'BluetoothClient' qui nous permettra d'interagir avec le Bluetooth de notre Smartphone. Il ne faut pas oublier d'ajouter un composant 'capteur d'horloge'

### Chapitre 3 : Application mobile et réalisation finale de notre prototype

(composant caché) se trouvant dans ‘Sensor’ (Figure 3.7 (b)), important pour cadencer notre application.



(a)



(b)

Figure 3.7. Sensor, (composant capteur d'horloge).

### Chapitre 3 : Application mobile et réalisation finale de notre prototype

On peut ajouter aussi une photo pour améliorer notre application et la rendre attractive (Figure 3.8).

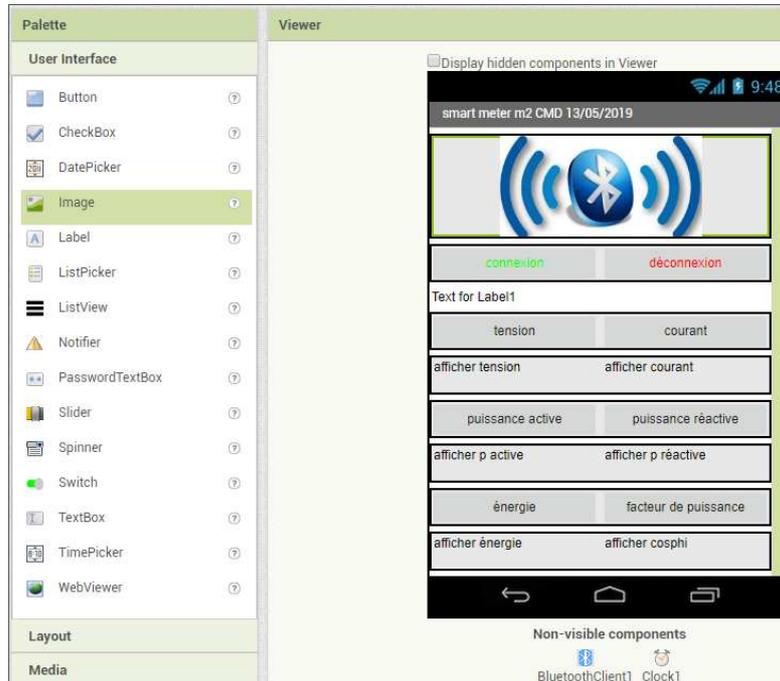


Figure 3.8. User interface : Ajout d'une image.

En résumé, notre interface Android (Figure 3.9) va comporter les éléments suivants qui doivent être programmées suivants l'évènement désiré et qui sont réunis dans le tableau ci-dessous.

Type d'objet	Nom de l'objet	Texte de l'objet
Bouton	ListPicker1	Connecté
Bouton	Button1	Déconnexion
Bouton	Button2	Tension
Bouton	Button3	Courant
Bouton	Button4	Puissance Active
Bouton	Button5	Puissance Réactive
Bouton	Button6	Energie
Bouton	Button7	Facteur de Puissance
Champ Texte	Label1	Connecté
Champ Texte	Label2	Valeur Tension
Champ Texte	Label3	Valeur Courant
Champ Texte	Label4	Valeur Puissance Active
Champ Texte	Label5	Valeur Puissance Réactive
Champ Texte	Label6	Valeur Energie
Champ Texte	Label7	Cosphi

Tableau 3.1 : Récapitulatif des objets insérés dans notre application.

## Chapitre 3 : Application mobile et réalisation finale de notre prototype

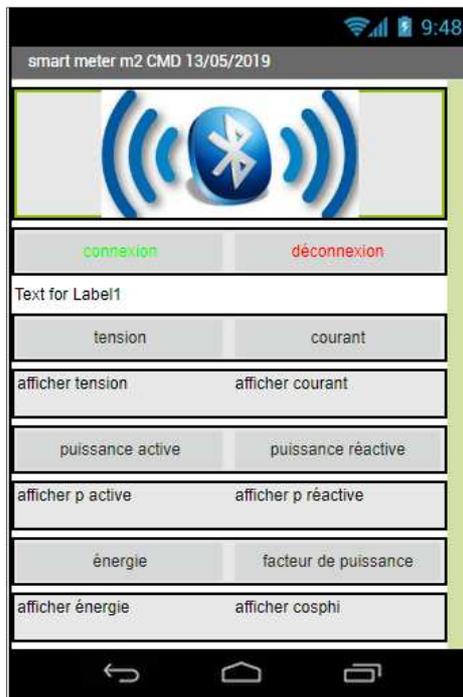


Figure 3.9. Design final de notre application.

La partie suivante concerne la programmation au niveau de la page « Blocks » qui permet de créer les interactions entre les blocs (ouverture, fermeture, déclenchement ...). L'ensemble ressemble à un puzzle, où vous pouvez définir les conditions de fonctionnement de l'application.

### 2.4.2. Programmation de l'Application AI2

Pour programmer l'application AI2, nous proposons d'utiliser le Scratch qui permet de créer un programme sous forme de schéma bloc. Très simple d'utilisation mais nécessitant un peu de logique de programmation.

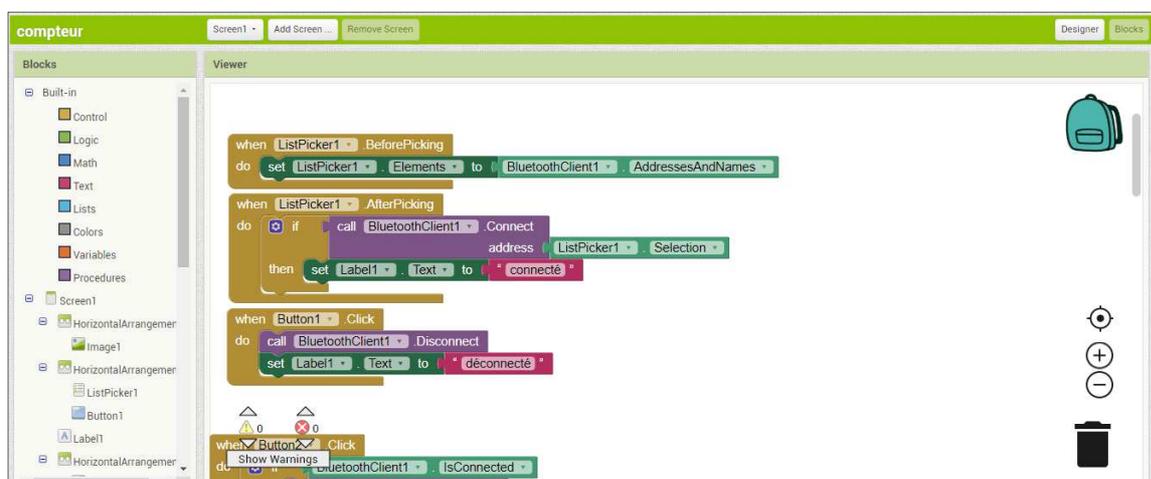


Figure 3.10 : Environnement de développement : Page 'Block'.

### Chapitre 3 : Application mobile et réalisation finale de notre prototype

Tous les objets sélectionnés dans ‘Designer’ sont disponibles dans le menu déroulant à gauche de l’écran ‘Block’ (Figure 3.10). En cliquant sur l’objet, toutes les fonctions associées deviennent visible. Il suffit de choisir la fonction (évènement) désirée.

Pour illustrer la méthode de programmation, nous allons prendre comme exemple la programmation de l’objet ‘ListPicker1’ du bouton ayant comme label ‘Connexion’. Cet objet servira à créer une connexion Bluetooth entre notre Smartphone et notre prototype qui est le compteur, principalement notre module HC05. Pour cela, nous avons suivi les 02 étapes suivantes :

- En cliquant sur l’objet ‘ListPicker1’ (dans la partie gauche de l’écran ‘Block’, nous avons sélectionné dans la liste déroulante la fonction ‘BeforePicking’. Cette fonction est représentée par une couleur moutarde (Figure 3.11). Après cela, nous avons ajouté les lignes de code (couleur verte, Figure 3.11) pour définir les éléments de la liste comme par exemple *périphériques Bluetooth disponibles*. Cela va nous permettre de visualiser tous les périphériques Bluetooth disponible ou capté par notre Smart-phone.
- En cliquant une deuxième fois sur l’objet ‘ListPicker1’, nous avons choisi la fonction ‘AfterPicking’ avec les lignes de code nécessaire pour afficher le texte ‘Connecté’ dans le champ Texte ayant l’objet ‘Label1’

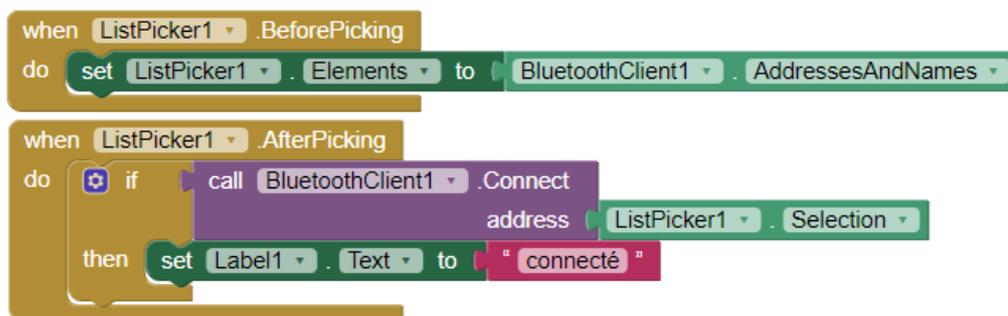


Figure 3.11 : Sélection des fonctions et des lignes de code associées à l’objet ‘ListPicker1’.

Les autres boutons (déconnecté, tension, courant, puissance active, puissance réactive, etc..) ont été programmés en suivant les mêmes étapes citées précédemment. La seule chose qui change, sont les fonctions et les lignes de code associées suivant l’évènement attendu par le bouton.

La Figure représentant la programmation complète de notre application se trouve dans l’annexe C.

### 2.5. Test de l'application

La première chose à faire est de connecter le Bluetooth de smartphone notre périphérique Bluetooth, (HC 05) en entrant le code (1234) comme le montre la figure ci-dessous.

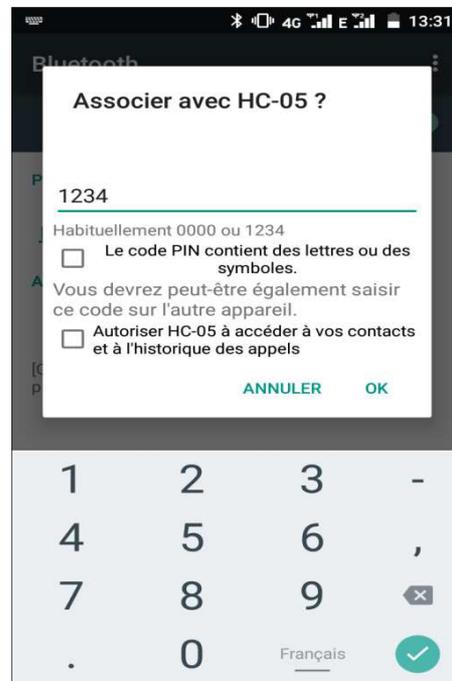


Figure 3.12 : Couplage de module bluetooth HC-05 avec le bluetooth de notre smartphone.

Pour tester l'application, il faut en premier lieu lancer l'application 'MIT AI2 Companion' au niveau de Smartphone (Figure 3.13 (a)) pour scanner le code QR généré par l'application de développement MIT App Inventor 2 (AI2) en cliquant sur le bouton 'Connect' (Figure 3.13 (b)).

En scannant le code QR, l'application s'affiche sur notre Smartphone comme le montre la Figure 3.14.

### Chapitre 3 : Application mobile et réalisation finale de notre prototype

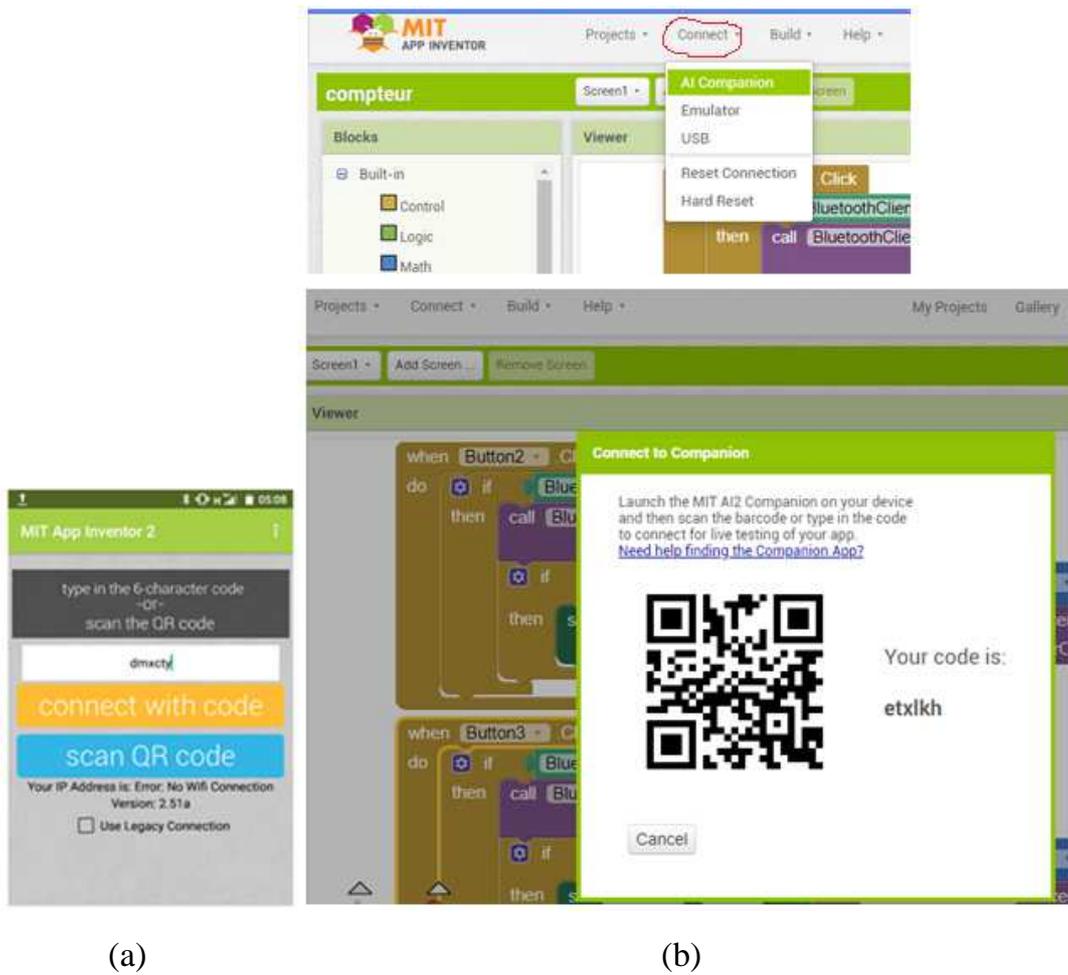


Figure 3.13 : Génération et scan du code QR.



Figure 3.14: Affichage de notre application avant connexion.

## Chapitre 3 : Application mobile et réalisation finale de notre prototype

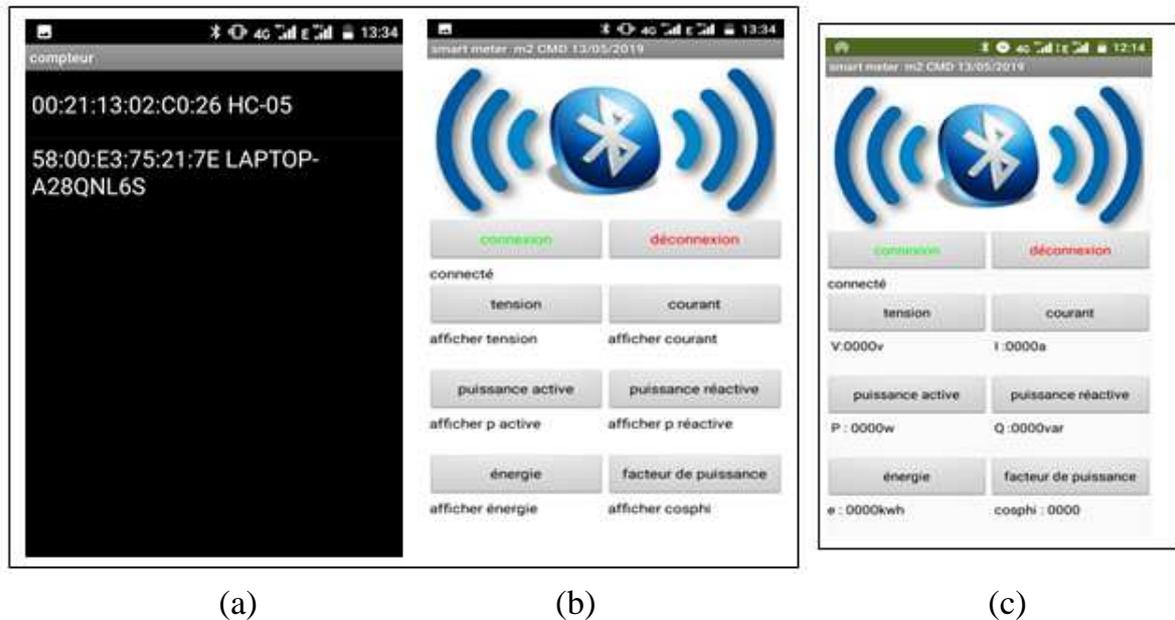


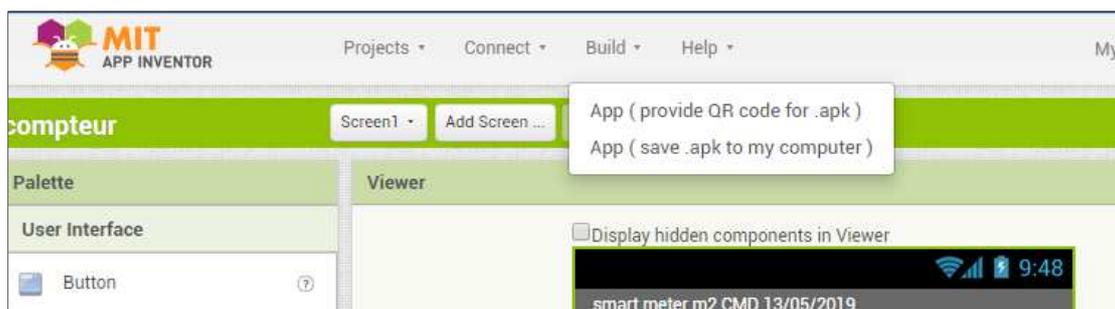
Figure 3.15: Affichage des tests de notre application mobile (compteur).

- a) En cliquant sur le bouton 'Connexion' de notre application, la fenêtre (Figure 3.15 (a)) affiche tous les périphériques disponibles qui peuvent être connectés avec notre Smartphone. En choisissant l'adresse de notre module Bluetooth HC05 (1<sup>ère</sup> ligne de la fenêtre en noir), le texte dans 'Text for Label1' (Figure 3.14) affichera 'Connecté' comme le montre la Figure 3.15 (b).
- b) D'après les données envoyées, on remarque que, quand on a cliqué sur bouton 'tension' de notre application, la valeur de tension dans label2 est correcte. Même chose pour les autres boutons (Figure 3.15 (c)).

### 2.6. Téléchargement de l'application mobile depuis App Inventor 2

Une fois que ce programme est correctement configuré et qu'aucune erreur ne s'affiche, l'étape finale est de compiler le programme comme il est montré sur la Figure 3.16 (a).

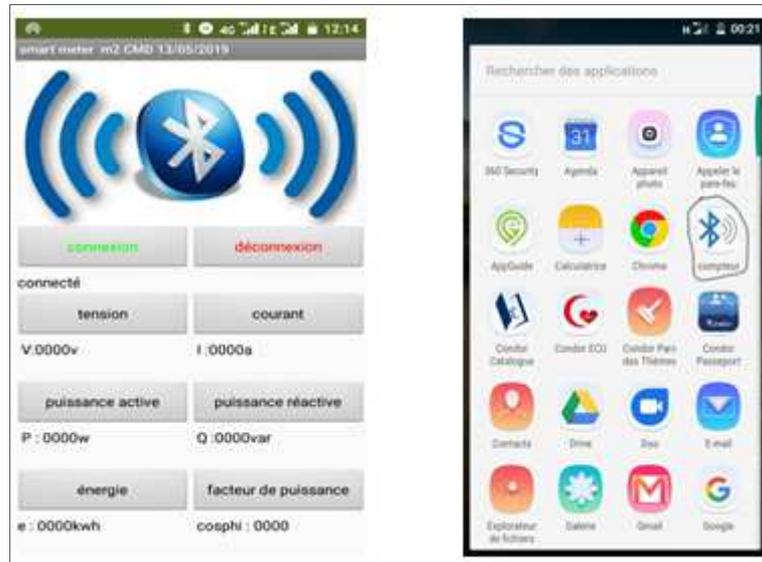
Menu → Build → App (save .apk to my computer).



(a)

### Chapitre 3 : Application mobile et réalisation finale de notre prototype

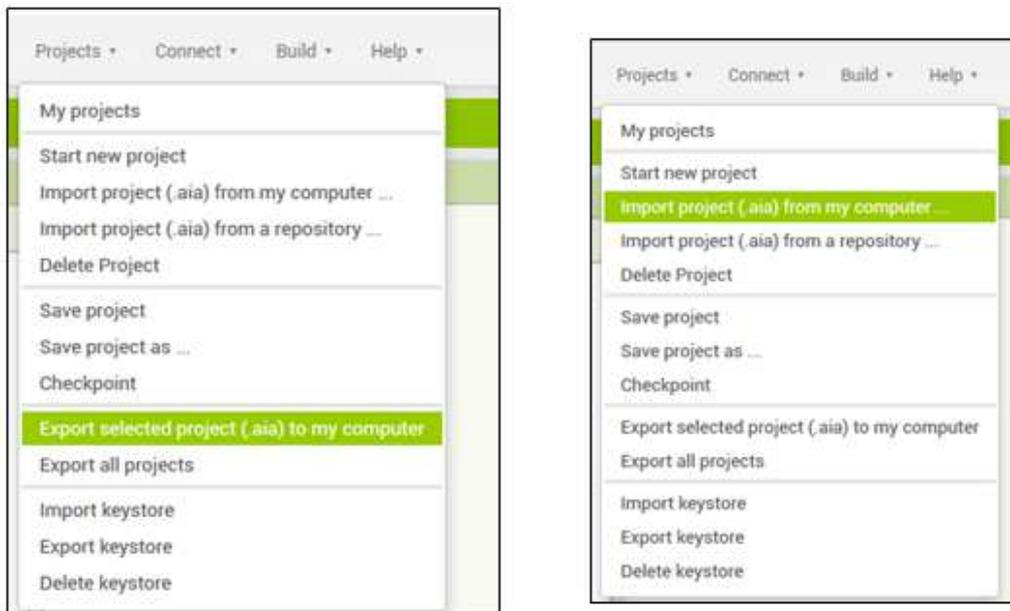
- a) Le fichier .apk est enregistré sur notre Pc, on doit copier ce fichier dans notre smartphone à l'aide d'un câble



(b)

Figure 3.16:Téléchargement d'application mobile sur notre smartphone.

Nous pouvons partager l'installateur de notre application .APK pour Android ou nous pouvons partager le projet .AIA afin que d'autres personnes puissent le modifier. On importe un fichier AIA pour ouvrir sur l'application en ligne et modifier le projet.



(a)

(b)

Figure 3.17: Partage d'installateur de notre application .APK.

## Chapitre 3 : Application mobile et réalisation finale de notre prototype

### 3. Réalisation finale de notre prototype

La réalisation finale du prototype de notre compteur intelligent avec l'intégration du système de communication (module Bluetooth HC-05) est présenté sur la figure suivante :

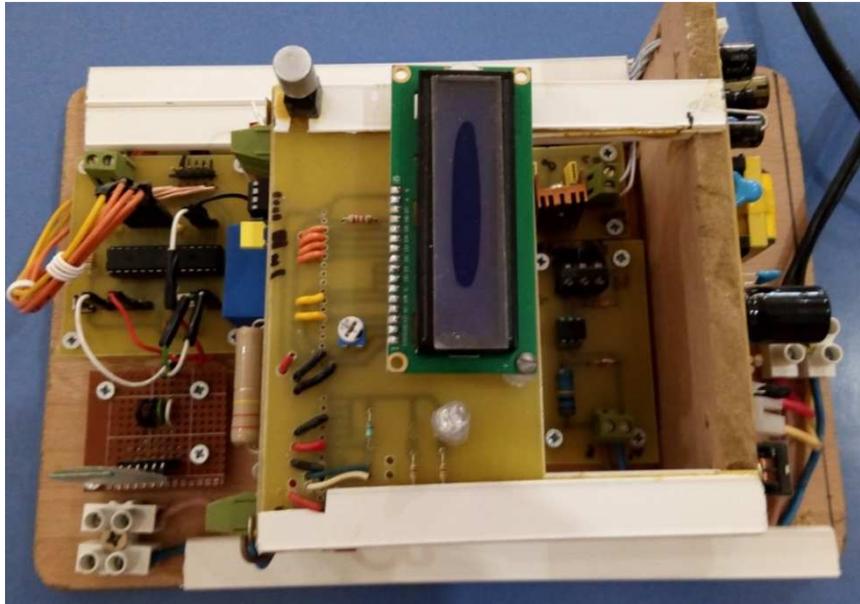
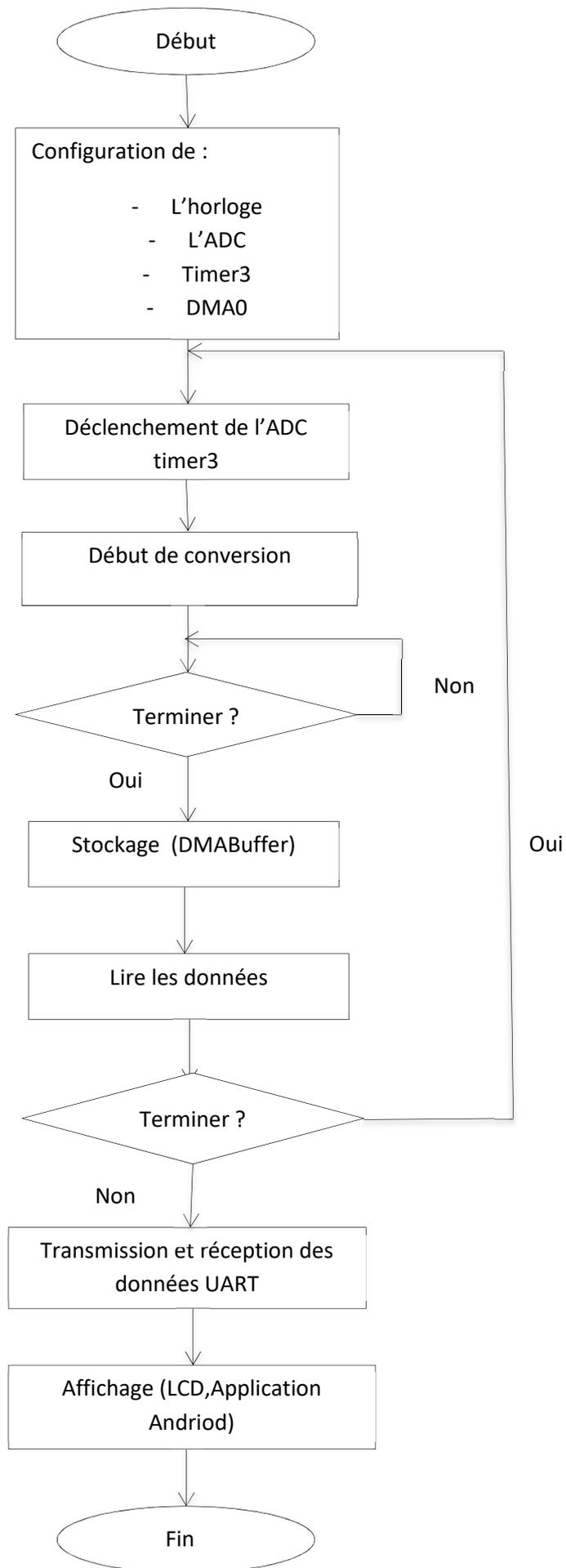


Figure 3.18 : Réalisation finale de notre prototype.

#### 3.1. Organigramme

L'organigramme suivant représente les différentes étapes qu'on a suivi pour la réalisation de prototype de notre compteur intelligent.

### Chapitre 3 : Application mobile et réalisation finale de notre prototype



## **Chapitre 3 : Application mobile et réalisation finale de notre prototype**

### **4. Conclusion**

Dans ce chapitre, nous avons découvert l'environnement de développement d'APP inventor en présentant son concept. Nous avons expliqué aussi les différentes étapes pour créer une application mobile qui permet d'afficher les données calculées par le prototype de notre compteur intelligent à l'aide d'un module Bluetooth HC-05.

## 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 des 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, peu 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 une autre de tension. Ces données sont traitées par un dsPIC et sont 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 de la consommation d'énergie d'une maison. Ce système réalisé facilite la tâche au releveur et évite la saisie manuelle de la valeur de la consommation au niveau de la maison.

# Bibliographie

- [1] : [https://fr.wikipedia.org/wiki/Compteur\\_communicant](https://fr.wikipedia.org/wiki/Compteur_communicant)
- [2] : <https://www.futura-sciences.com/planete/definitions/developpement-durable-compteur-intelligent-6952/>
- [3]: <https://www.energuide.be/fr/questions-reponses/quest-ce-quun-compteur-intelligent/126/>
- [4]: [https://fr.wikipedia.org/wiki/Compteur\\_%C3%A9lectrique](https://fr.wikipedia.org/wiki/Compteur_%C3%A9lectrique)
- [5]: <https://les-smartgrids.fr/economies-smart-grids-france/>
- [6]: [https://fr.wikipedia.org/wiki/Transmission\\_sans\\_fil](https://fr.wikipedia.org/wiki/Transmission_sans_fil)
- [7] : <https://www.open-electronics.org/top-5-wireless-ways-to-communicate-with-your-controller/>
- [8]: <https://observatoire-electricite.fr/fiches-pedagogiques/Les-compteurs-communicants>
- [9]: H. BOUDJEMAI, I. FELLAH. ‘Amélioration des performances d’acquisition des signaux pour un compteur intelligent’, PFE GEE, Tlemcen, 27 juin 2017.
- [10] : [https://fr.wikipedia.org/wiki/Alimentation\\_%C3%A0\\_d%C3%A9coupage](https://fr.wikipedia.org/wiki/Alimentation_%C3%A0_d%C3%A9coupage)
- [11] : <http://blogpeda.ac-poitiers.fr/lp2i-si/2013/03/03/les-capteurs/>
- [12] : <https://www.moussasoft.com/product/lem-lv-25-p-captur-de-tension-maroc>
- [13] : [http://www.orbit-dz.com/produit/scanner-automobile-2/arduino-compatibles/shields-et-accessoires/courant-tension\\_shields/module-acs712-detail](http://www.orbit-dz.com/produit/scanner-automobile-2/arduino-compatibles/shields-et-accessoires/courant-tension_shields/module-acs712-detail)
- [14] : <https://www.aurel32.net/elec/lcd.php>
- [15] : [https://fr.wikipedia.org/wiki/Microcontr%C3%B4leur\\_PIC#dsPIC30\\_/dsPIC33F\\_/dsPIC33E](https://fr.wikipedia.org/wiki/Microcontr%C3%B4leur_PIC#dsPIC30_/dsPIC33F_/dsPIC33E)
- [16] : <https://www.microchip.com/wwwproducts/en/en532299>
- [17] : <https://ar.scribd.com/document/58470494/code-DMA>
- [18] : <https://circuitdigest.com/microcontroller-projects/uart-communication-using->
- [19] : <http://dictionnaire.sensagent.leparisien.fr/UART/fr-fr/>
- [20] : <https://www.edgefxkits.com/blog/interfacing-hc-05-bluetooth-module-with-microcontroller/>
- [21] : <https://deepbluembedded.com/bluetooth-module-hc05-interfacing-pic-microcontroller-tutorial/>
- [22] : <https://www.techwalla.com/articles/how-to-use-putty-for-a-serial-connection>

# Annexe A

## Datasheet - ICL7660S



DATASHEET

### ICL7660S, ICL7660A

Super Voltage Converters

ICL7660S  
Rev. 7.00  
January 25, 2013

The ICL7660S and ICL7660A Super Voltage Converters are monolithic CMOS voltage conversion ICs that guarantee significant performance advantages over other similar devices. They are direct replacements for the industry standard ICL7660 offering an extended operating supply voltage range up to 12V, with lower supply current. A **Frequency Boost pin** has been incorporated to make the user to achieve lower output impedance despite using smaller capacitors. All requirements are highlighted in the "Electrical Specifications" section on page 3. Critical parameters are guaranteed over the entire commercial and industrial temperature ranges.

The ICL7660S and ICL7660A perform supply voltage conversions from positive to negative for an input range of 1.0V to 12V, resulting in complementary output voltages of -1.0V to -12V. Only two non-critical external capacitors are needed, for the charge pump and charge transfer functions. The ICL7660S and ICL7660A can be connected to function as a voltage divider and will generate up to 22.5V with a 12V input. They can also be used as a voltage multiplier or voltage dividers.

Each chip contains a series DC power supply regulator, RC oscillator, voltage level translator, and four output power MOSFET switches. The oscillator (when activated) oscillates at a nominal frequency of 100kHz for an input supply voltage of 5.0V. This frequency can be lowered by the addition of an external capacitor to the "OSC" terminal, or the oscillator may be over-driven by an external clock.

The "L1" terminal may be tied to GND to bypass the internal series regulator and improve low voltage (5V) operation. At medium to high voltages (3.0V to 12V), the L1 pin is left floating to prevent device heating.

In some applications, an external Schottky diode from  $V_{OUT}$  to CAP is needed to guarantee reliable free operation (see Di's and Dext's section on page 6).

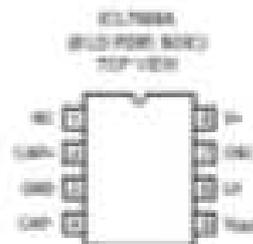
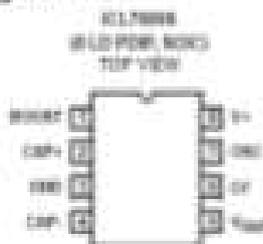
### Features

- Guaranteed Lower than Supply Current for All Temperature Ranges
- Wide Operating Voltage Range: 1.0V to 12V
- 100% Tested at 25
- Boost Pin (Pin 1) for Higher Switching Frequency
- Guaranteed Minimum Power Efficiency of 90%
- Improved Minimum Output Current Voltage Conversion Efficiency of 90%
- Improved SOA Limiting Protection
- Simple Conversion of +5V Logic Supply to -5V Supplies
- Simple Voltage Multiplication ( $V_{OUT} = 1.4V_{IN}$ )
- Easy to Use, Requires Only Two External Non-Critical Passive Components
- Improved Direct Replacement for Industry Standard ICL7660 and Other Second Source Devices
- Pb-Free Available (RoHS Compliant)

### Applications

- Simple Conversion of +5V to -5V Supplies
- Voltage Multiplication ( $V_{OUT} = 1.4V_{IN}$ )
- Negative Supplies for Data Acquisition Systems and Instrumentation
- MOSFET Power Supplies
- Supply Splitter ( $V_{OUT} = 1.4V_{IN}$ )

### Pin Configurations



## Datasheet - capteur de tension LV 25-P

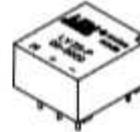
### Voltage Transducer LV 25-P

For the electronic measurement of voltages : DC, AC, pulsed...  
with a galvanic isolation between the primary circuit (high voltage)  
and the secondary circuit (electronic circuit).



$$I_{PN} = 10 \text{ mA}$$

$$V_{PN} = 10 \dots 500 \text{ V}$$



#### Electrical data

$I_{pn}$	Primary nominal r.m.s. current	10	mA
$I_p$	Primary current, measuring range	0 .. $\pm 14$	mA
$R_v$	Measuring resistance	$R_{vmax}$	$R_{vmin}$
		with $\pm 12 \text{ V}$	$\textcircled{1} \pm 10 \text{ mA}_{max}$ 30 100 $\Omega$ $\textcircled{2} \pm 14 \text{ mA}_{max}$ 30 100 $\Omega$
	with $\pm 15 \text{ V}$	$\textcircled{3} \pm 10 \text{ mA}_{max}$ 100 350 $\Omega$ $\textcircled{4} \pm 14 \text{ mA}_{max}$ 100 190 $\Omega$	
	$I_{sn}$	Secondary nominal r.m.s. current	25
$K_n$	Conversion ratio	2500 : 1000	
$V_c$	Supply voltage ( $\pm 5\%$ )	$\pm 12 \dots 15$	V
$I_c$	Current consumption	$10 \textcircled{1} \pm 15 \text{ V} + I_p$	mA
$V_i$	R.m.s. voltage for AC isolation test <sup>1)</sup> , 50 Hz, 1 min	2.5	kV

#### Accuracy - Dynamic performance data

$X_0$	Overall Accuracy @ $I_{pn}$ , $T_A = 25^\circ\text{C}$	@ $\pm 12 \dots 15 \text{ V}$	$\pm 0.9$	%
		@ $\pm 15 \text{ V} (\pm 5\%)$	$\pm 0.8$	%
$e$	Linearity		$< 0.2$	%
$I_0$	Offset current @ $I_p = 0$ , $T_A = 25^\circ\text{C}$	Typ	Max	
			$\pm 0.15$	mA
$I_{0T}$	Thermal drift of $I_0$	$0^\circ\text{C} \dots +25^\circ\text{C}$	$\pm 0.06$	$\pm 0.25$
		$+25^\circ\text{C} \dots +70^\circ\text{C}$	$\pm 0.10$	$\pm 0.35$
				mA
$t_r$	Response time <sup>2)</sup> @ 90 % of $V_{vmax}$	40		$\mu\text{s}$

#### General data

$T_A$	Ambient operating temperature	0 .. +70	$^\circ\text{C}$
$T_b$	Ambient storage temperature	-25 .. +85	$^\circ\text{C}$
$R_p$	Primary coil resistance @ $T_A = 70^\circ\text{C}$	250	$\Omega$
$R_s$	Secondary coil resistance @ $T_A = 70^\circ\text{C}$	110	$\Omega$
$m$	Mass	22	g
	Standards <sup>3)</sup>	EN 50178	

**Notes :** <sup>1)</sup> Between primary and secondary

<sup>2)</sup>  $R_p = 25 \text{ k}\Omega$  (L/R constant, produced by the resistance and inductance of the primary circuit)

<sup>3)</sup> A list of corresponding tests is available

#### Features

- Closed loop (compensated) voltage transducer using the Hall effect
- Insulated plastic case recognized according to UL 94-V0.

#### Principle of use

- For voltage measurements, a current proportional to the measured voltage must be passed through an external resistor  $R$ , which is selected by the user and installed in series with the primary circuit of the transducer.

#### Advantages

- Excellent accuracy
- Very good linearity
- Low thermal drift
- Low response time
- High bandwidth
- High immunity to external interference
- Low disturbance in common mode.

#### Applications

- AC variable speed drives and servo motor drives
- Static converters for DC motor drives
- Battery supplied applications
- Uninterruptible Power Supplies (UPS)
- Power supplies for welding applications.

981125/14

## Datasheet – Module Bluetooth HC-05

### Specifications

#### Hardware features

- Typical -80dBm sensitivity
- Up to +4dBm RF transmit power
- Low Power 1.8V Operation ,1.8 to 3.6V I/O
- PIO control
- UART interface with programmable baud rate
- With integrated antenna
- With edge connector

HC-05 Bluetooth module

iteadstudio.com

06.18.2010

2



Tech Support: [info@iteadstudio.com](mailto:info@iteadstudio.com)

#### Software features

- Default Baud rate: 38400, Data bits:8, Stop bit:1,Parity:No parity, Data control: has. Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.
- Given a rising pulse in PIO0, device will be disconnected.
- Status instruction port PIO1: low-disconnected, high-connected;
- PIO10 and PIO11 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:"0000" as default

# Annexe B

## AT command Default:

How to set the mode to server (master):

1. Connect PIO11 to high level.
2. Power on, module into command state.
3. Using baud rate 38400, sent the "AT+ROLE=1\r\n" to module, with "OK\r\n" means setting successes.
4. Connect the PIO11 to low level, repower the module, the module work as server (master).

AT commands: (all end with \r\n)

1. Test command:

Command	Respond	Parameter
AT	OK	-

2. Reset

Command	Respond	Parameter
AT+RESET	OK	-

3. Get firmware version

Command	Respond	Parameter
AT+VERSION?	+VERSION:<Param> OK	Param : firmware version

Example:

```
AT+VERSION?\r\n+VERSION:2.0-20100601  
OK
```

# Annexe C : Programme Complet

```
when ListPicker1 .BeforePicking
do set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

when ListPicker1 .AfterPicking
do if call BluetoothClient1 .Connect
    address ListPicker1 . Selection
then set Label1 . Text to "connecté"

when Button1 .Click
do call BluetoothClient1 .Disconnect
set Label1 . Text to "déconnecté"

when Button2 .Click
do if BluetoothClient1 .IsConnected
then call BluetoothClient1 .SendText
    text "1"
if call BluetoothClient1 .BytesAvailableToReceive > 0
then set Label2 . Text to call BluetoothClient1 .ReceiveText
    numberOfBytes call BluetoothClient1 .BytesAvailableToReceive

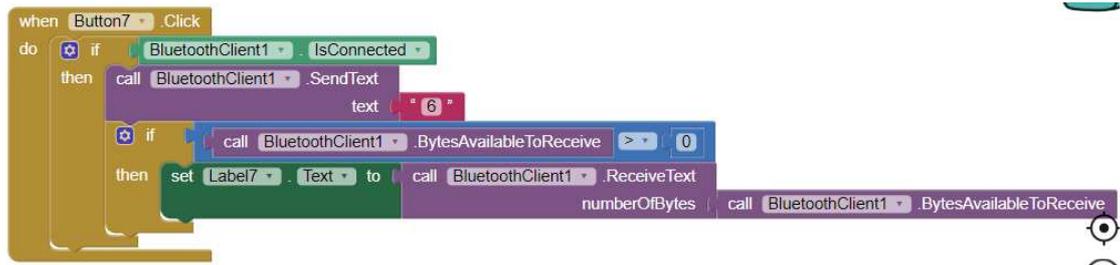
when Button3 .Click
do if BluetoothClient1 .IsConnected
then call BluetoothClient1 .SendText
    text "2"
if call BluetoothClient1 .BytesAvailableToReceive > 0
then set Label3 . Text to call BluetoothClient1 .ReceiveText
    numberOfBytes call BluetoothClient1 .BytesAvailableToReceive

when Button4 .Click
do if BluetoothClient1 .IsConnected
then call BluetoothClient1 .SendText
    text "3"
if call BluetoothClient1 .BytesAvailableToReceive > 0
then set Label4 . Text to call BluetoothClient1 .ReceiveText
    numberOfBytes call BluetoothClient1 .BytesAvailableToReceive

when Button5 .Click
do if BluetoothClient1 .IsConnected
then call BluetoothClient1 .SendText
    text "4"
if call BluetoothClient1 .BytesAvailableToReceive > 0
then set Label5 . Text to call BluetoothClient1 .ReceiveText
    numberOfBytes call BluetoothClient1 .BytesAvailableToReceive

when Button6 .Click
do if BluetoothClient1 .IsConnected
then call BluetoothClient1 .SendText
    text "5"
if call BluetoothClient1 .BytesAvailableToReceive > 0
then set Label6 . Text to call BluetoothClient1 .ReceiveText
    numberOfBytes call BluetoothClient1 .BytesAvailableToReceive
```

## Annexe C : Programme Complet



## 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 dsPIC33F, 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 App invonter.

**Mot clés :** compteur intelligent, dsPIC, Bluetooth, Android.

### Abstract

The advancement of technologies to help improve grid system systems by integrating renewable energies as well as the development of meters that are the basis of smart grids. The main interest of these meters is to track the data in real time.

Our goal is to study and achieve a so-called smart counter. The electronic board produced is controlled by a dsPIC33F, the latter also allows the transmission of data to a Smartphone via a wireless communication system of Bluetooth type. The received data are subsequently visualized using an application made in App invonter.

**Key words:** smart meter, dsPIC, Bluetooth, Android.

### ملخص

تقدم التقنيات ساعد في تحسين أنظمة نظام الشبكة خلال دمج الطاقات المتجددة بالإضافة إلى تطوير عدادات تشكل أساس الشبكات الذكية. الهدف الرئيسي لهذه العدادات هو تتبع البيانات في الوقت الحقيقي.

هدفنا هو دراسة وتحقيق ما يسمى عداد الذكية. يتم التحكم في اللوحة الإلكترونية المنتجة بواسطة تحكم dsPIC33F، ويسمح هذا الأخير أيضاً نقل البيانات إلى الهاتف الذكي عبر نظام اتصال لاسلكي من نوع Bluetooth. يتم بعد ذلك عرض البيانات المستلمة باستخدام تطبيق تم إنشاؤه في App invonter.

**كلمات مفتاحية:** العدادات الذكية ، dsPIC ، Bluetooth ، أندرويد.