

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 Abou Bekr Belkaid -Tlemcen-

Faculté de TECHNOLOGIE



Filière : Electrotechnique

Spécialité : Commandes Electriques

Département de Génie Electrique et Electronique (GEE)

Projet de Fin d'Etude pour l'obtention du diplôme de Master  
Académique

Par : RAMDANE Rafik et ZIRAR Mohammed Salah Eddine

Intitulé :

## Réalisation d'un compteur d'énergie intelligent

Soutenu par visio-conférence le 2 Septembre 2020 devant la commission d'examen  
composée de :

S.M. MELIANI	Président	MCA	Univ. Tlemcen
A. MECHERNENE	Examineur	MCB	Univ. Tlemcen
L. BAGHLI	Encadrant	Professeur	Univ. Tlemcen

# Dédicaces 1

Je dédie ce modeste travail à :

- Mes parents qui ont tout donné pour que je réussisse.
- Ma famille qui est la source de mon soutien.
- Tous mes enseignants et mes professeurs qui sont la source de mon savoir.

RAMDANE Rafik

## Dédicaces 2

Je dédie ce travail :

- à ma mère, la personne la plus chère à mon cœur, qui a toujours prié pour ma réussite
- à mon père qui a travaillé dur pour moi et m'a appris le sens de la lutte pour atteindre les objectifs.
- à mon frère et ma sœur et toute ma famille.
- à tous mes professeurs à travers toutes les étapes de mes études
- à tous ceux qui, de près ou de loin, nous ont aidés et contribués au succès de ce travail.

ZIRAR Mohammed Salah Eddine

## Remerciements

Nous souhaitons en premier lieu á remercier ALLAH qui nous aide á accomplir ce modeste travail et qui nous donne 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 á Monsieur L. BAGHLI pour avoir dirigé ce travail, pour son aide et ses précieux conseils ainsi que pour les nombreuses discussions que nous avons eues.

Nous tenons également à adresser nos remerciements aux membres du jury Monsieur S. M. MELIANI, et Monsieur A. MECHERNENE pour avoir accepté d'examiner notre travail et de l'enrichir par leurs propositions.

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

RAMDANE Rafik

ZIRAR Mohammed Salah Eddine

# Table des matières

Dédicaces 1 .....	2
Dédicaces 2 .....	3
Remerciements .....	4
Table des matières .....	5
Table des figures .....	8
Liste des tableaux .....	10
Nomenclature .....	11
Glossaire .....	12
Introduction générale .....	13
Chapitre 1 : Etat de l'art .....	15
I- Introduction .....	15
II- La tarification de l'électricité .....	15
III- Le compteur Intelligent Linky .....	17
1- Comment remonter les informations .....	18
2- Quelles sont les données recueillies par le compteur .....	18
3- La sécurité de ces données .....	19
IV- Perspectives d'utilisation de compteurs intelligents en Algérie .....	20
1- A court terme .....	20
2- A moyen Terme .....	20
3- Quel accueil la société de consommation algérienne va-t-elle-leur donner .....	20
V- Conclusion .....	20
Chapitre 2 : Etudes théoriques pour une tarification .....	21
I- Introduction .....	21
II- Les compteurs intelligents .....	21
III- Système de tarification de SONELAGAZ .....	22
1- La méthode actuelle de calcul du SONELGAZ (Consommation/tranche) : .....	22
IV- Proposition d'un nouveau plan de tarification .....	23
V- Conclusion .....	36
Chapitre 3: Réalisation de prototype de compteur .....	38
I- Introduction .....	38
II- Description de montage .....	38
III- Capteur de courant .....	38

1-	Description d'ACS758.....	38
2-	Description de SCT-013-000 .....	40
IV-	Capteur de tension.....	43
1-	Description .....	43
V-	Wemos Lolin32.....	44
1-	Description .....	44
2-	Les caractéristiques.....	45
VI-	Ecran OLED .....	46
VII-	Réalisation du montage.....	46
1-	Le montage d'ACPL-C87AT .....	46
2-	Le montage de SCT 013 .....	46
3-	Montage d'écran OLED.....	48
VIII-	Les calculs nécessaires .....	49
1-	Calcul de constant de courant.....	49
2-	Calcul des puissances.....	50
3-	Calcul de facteur de puissance .....	50
4-	Calcul de l'énergie .....	51
IX-	Conclusion .....	51
Chapitre 04 : Partie Logicielle.....		52
I-	Introduction.....	52
II-	Description du programme .....	52
1-	Coté Opérateur de réseaux .....	52
2-	Coté Client .....	53
III-	IDE .....	54
1-	Introduction IDE.....	54
2-	Préparer l'environnement de l'IDE .....	54
3-	Programmation de la mémoire FLASH (data) .....	54
4-	Programmation de la mémoire FLASH (programme).....	55
IV-	Réalisation du programme.....	55
1-	Routine de connexion Wifi .....	55
2-	Routines de l'afficheur OLED .....	60
3-	Routines du serveur web local (ESP32).....	61
4-	Programme de capteur de courant .....	62
5-	Programme de temps .....	63
6-	Calcul des données de consommation .....	63

7-	Affichage sur Navigateur Web.....	65
8-	La base de données .....	66
10-	Le choix des prix de jour.....	70
11-	Transmission des données de consommation vers la base de données	74
V-	Sécurité des données .....	76
1-	Les mots de passe .....	76
2-	L'utilisation de protocole https.....	76
3-	Connexion isolé de la base de données.....	77
4-	Sécurité de Site Web.....	77
VI-	Conclusion .....	78
	Conclusion générale.....	79
	Bibliographies.....	81
	Annexe A : Grille comparative d'électricité de Total Direct Energie et EDF.....	82
	Annexe B : Tarifs comparés : Economie 7 et Tarifs électricité normal.....	82
	Annexe C : Tarifs proposés .....	83
	Annexe D : Datasheet – ACS758 .....	87
	Annexe E : Datasheet – SCT 013 .....	90
	Annexe F : Datasheet – ACPL-C87AT.....	91
	Annexe G : Préparer l'environnement de l'IDE.....	93
	Résumé .....	97
	Abstract .....	97
	الملخص .....	97

# Table des figures

Figure 1-1 : Le compteur intelligent Linky .....	18
Figure 2-1 : Courbe de charge durant un jour ouvrable du mois de janvier .....	24
Figure 2-2 : Courbe de charge durant un jour de weekend du mois de janvier.....	24
Figure 2-3 : Courbe de charge durant un jour ouvrable du mois de février.....	25
Figure 2-4 : Courbe de charge durant un jour de weekend du mois de février .....	25
Figure 2-5 : Courbe de charge durant un jour ouvrable du mois de mars.....	26
Figure 2-6 : Courbe de charge durant un jour de weekend du mois de mars.....	26
Figure 2-7 : Courbe de charge durant un jour ouvrable du mois d'avril .....	27
Figure 2-8 : Courbe de charge durant un jour de weekend du mois d'avril.....	27
Figure 2-9 : Courbe de charge durant un jour ouvrable du mois de mai .....	28
Figure 2-10 : Courbe de charge durant un jour de weekend du mois de mai.....	28
Figure 2-11 : Courbe de charge durant un jour ouvrable du mois de juin .....	29
Figure 2-12 : Courbe de charge durant un jour de weekend du mois de juin .....	29
Figure 2-13 : Courbe de charge durant un jour ouvrable du mois de juillet .....	30
Figure 2-14 : Courbe de charge durant un jour de weekend du mois de juillet .....	30
Figure 2-15 : Courbe de charge durant un jour ouvrable du mois d'août.....	31
Figure 2-16 : Courbe de charge durant un jour weekend du mois d'août.....	31
Figure 2-17 : Courbe de charge durant un jour ouvrable du mois de septembre .....	32
Figure 2-18 : Courbe de charge durant un jour de weekend du mois de septembre...32	
Figure 2-19 : Courbe de charge durant un jour ouvrable du mois d'octobre .....	33
Figure 2-20 : Courbe de charge durant un jour de weekend du mois d'octobre .....	33
Figure 2-21 : Courbe de charge durant un jour ouvrable du mois de novembre .....	34
Figure 2-22 : Courbe de charge durant un jour de weekend du mois de novembre ...34	
Figure 2-23 : Courbe de charge durant un jour ouvrable du mois de décembre .....	35
Figure 2-24 : Courbe de charge durant un jour de semaine du mois de décembre ....35	
Figure 3-1 : Schéma descriptif de montage.....	38
Figure 3-2 : Schéma d'ACS758.....	40
Figure 3-3 : Schéma de montage de capteur ACS758 .....	40
Figure 3-4 : Capteur de courant SCT 013 .....	42
Figure 3-5 : Fiche jack de capteur de courant SCT 013.....	42
Figure 3-6 : ACPL-C87AT .....	43
Figure 3-7 : Circuit de capteur de tension typique .....	44
Figure 3-8 : ESP32 Wemos Lolin32.....	45
Figure 3-9 : Ecran OLED.....	46
Figure 3-10 : Montage ACPL-C87AT .....	46
Figure 3-11 : Schéma de montage de SCT 013.....	47
Figure 3-12 : Les Résultats de tests de la sortie de SCT 013.....	48
Figure 3-13 : Caractéristiques physiques de bus I2C .....	49
Figure 3-14 : Conditions de START et de STOP.....	49
Figure 4-1 : Schéma synoptique .....	52
Figure 4-2 : Se connecter ESP32.....	56
Figure 4-3 : Page Web Energy Meter avant la connexion .....	57
Figure 4-4 : Configuration Wifi.....	57



Figure 4-5 : Page Web Energy Meter après la connexion .....	58
Figure 4-6 : Page HTML Energy meter .....	66
Figure 4-7 : 5 tables de la base de données .....	67
Figure 4-8 : Table users .....	67
Figure 4-9 : Table datestable .....	67
Figure 4-10 : Table totalday .....	67
Figure 4-11 : Les données des 3 tables précédentes dans une table .....	68
Figure 4-12 : Table Price.....	68
Figure 4-13 : Table consommation.....	68
Figure 4-14 : La relation entre les tables de notre base de données .....	69
Figure 4-15 : Connexion au site web .....	69
Figure 4-16 : L'inscription au site web.....	70
Figure 4-17 : Après connexion au site web.....	70
Figure 4-18 : Envoyé les prix de jour .....	71
Figure 4-19 : Résultat après l'insertion les donnes d'Excel.....	72

## Liste des tableaux

---

Tableau 2-1 : Proposition des plages horaires d'un jour du mois de janvier .....	24
Tableau 2-2 : Proposition des plages horaires d'un jour de mois de février .....	25
Tableau 2-3 : Proposition des plages horaires d'un jour de mois de mars.....	26
Tableau 2-4 : Proposition des plages horaires d'un jour de mois d'avril .....	27
Tableau 2-5 : Proposition des plages horaires d'un jour du mois de mai .....	28
Tableau 2-6 : Proposition des plages horaires d'un jour du mois de juin .....	29
Tableau 2-7 : Proposition des plages horaires d'un jour du mois de juillet.....	30
Tableau 2-8 : Proposition des plages horaires d'un jour du mois d'août.....	31
Tableau 2-9 : Proposition des plages horaires d'un jour du mois de septembre .....	32
Tableau 2-10 : Proposition des plages horaire d'un jour du mois d'octobre .....	33
Tableau 2-11 : Proposition des plages horaires d'un jour du mois de novembre.....	34
Tableau 2-12 : Proposition des plages horaires d'un jour du mois de décembre .....	35
Tableau 3-1 : Les Résultats de tests de la sortie de SCT 013.....	47

## Nomenclature

Les principales notations et abréviations utilisées dans ce mémoire sont explicitées ci-dessous, sous leur forme la plus couramment employée dans le domaine du génie électrique.

<b>Nom</b>	<b>Symbole</b>	<b>Unité</b>
<b>Temps</b>	<b>t</b>	<b>s</b>
<b>Tension</b>	<b>V</b>	<b>V</b>
<b>Courant</b>	<b>I</b>	<b>A</b>
<b>Énergie électrique</b>	<b>E</b>	<b>kWh</b>
<b>Puissance active</b>	<b>P</b>	<b>W</b>
<b>Puissance réactive</b>	<b>Q</b>	<b>VAR</b>
<b>Puissance apparente</b>	<b>S</b>	<b>VA</b>
<b>Résistance</b>	<b>R</b>	<b><math>\Omega</math></b>
<b>Capacité</b>	<b>C</b>	<b>F</b>
<b>Fréquence</b>	<b>F</b>	<b>Hz</b>

## Glossaire

Acronyme	Signification
<b>HP</b>	Heures pleines
<b>HC</b>	Heures creuses
<b>PHP</b>	HyperText Preprocessor, est un langage de programmation principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP.
<b>MySQL</b>	Serveur de bases de données relationnelles Open Source
<b>JSON</b>	(JavaScript Objet Notation) est un langage léger d'échange de données textuelles
<b>Htмл</b>	HyperText Markup Language, généralement abrégé HTML ou dans sa dernière version HTML5, est le langage de balisage conçu pour représenter les pages web.
<b>Serveur web</b>	Soit un logiciel de service de ressources web (serveur HTTP), soit un serveur informatique (ordinateur) qui répond à des requêtes du World Wide Web sur un réseau public (Internet) ou privé (intranet), en utilisant principalement le protocole HTTP.
<b>Base de données</b>	Ensemble d'informations structurées accessibles au moyen d'un logiciel.
<b>IDE</b>	(Integrated Development Environment) En programmation informatique, un environnement de développement est un ensemble d'outils qui permet d'augmenter la productivité des programmeurs qui développent des logiciels.
<b>THT</b>	Très haut tension
<b>MT</b>	Moyenne tension
<b>BT</b>	Base tension
<b>ANSSI</b>	Agence nationale de sécurité des systèmes d'information
<b>AMR</b>	Automated Meter Reading
<b>AMM</b>	Advanced Meter Management
<b>CPL</b>	Courants porteurs en ligne
<b>ADC</b>	Analog-to-Digital Conversion
<b>UART</b>	Universal Asynchrone Receiver Transmitter
<b>SDA</b>	Serial Data
<b>SCL</b>	Serial CLock

## Introduction générale

Nous sommes à un point d'inflexion dans la transformation de l'industrie mondiale de l'énergie électrique, faire face à des défis qui exigent des stratégies claires. Les principaux défis sont bien connus de nous tous. Fournir de l'énergie électrique sûre qui soutient les évolutions économiques et technologiques et démographiques, le remplacement des infrastructures vieillissantes par une production, transport et distribution continus, abordables, fiables et flexibles, et le défi est de développer des systèmes énergétiques intelligents des réseaux intelligents.

Les données sont au cœur de cette transformation de l'industrie et de la création de systèmes énergétiques. La collecte de gros volumes de données est une caractéristique clé de ces systèmes. Une source importante de données proviendra du déploiement des compteurs intelligents, qui est largement accepté comme la première étape du développement de réseaux intelligents.

Les compteurs intelligents fourniront à terme des informations sur l'utilisation par les consommateurs et la disponibilité du réseau.

Les compteurs intelligents sont parmi les leviers de la transformation des réseaux de distribution électrique aux réseaux intelligents. Le déploiement des compteurs intelligents devrait permettre de mieux connaître la consommation électrique et de faciliter la transition vers les énergies renouvelables et les véhicules électriques et une meilleure gestion de la production et de la distribution d'électricité, et donc de faire des économies. Le suivi en temps réel ouvre aussi la porte à la facturation de l'énergie en temps réel. Il est capable de mesurer à chaque instant plusieurs grandeurs comme la tension, le courant, la puissance et l'énergie, puis de transmettre toutes ces informations au centre de contrôle. Par conséquent, sans le déplacement d'un agent, La chose la plus importante dans ces informations est de connaître les heures de pointe pour éviter les surcharges et cela en contrôlant le comportement des consommateurs avec le changement des prix de l'électricité durant la journée et c'est ce que nous allons proposer dans ce projet.

Donc, comment fabriquer ce compteur électrique qui prend en compte la variation tarifaire heure par heure et quotidienne du coût de l'électricité. Il doit aussi communiquer dans les deux sens de l'opérateur de réseau électrique vers le client et vice-versa.

Afin de répondre à cette question et réaliser ce projet nous avons divisé notre mémoire en quatre chapitres :

Le premier chapitre présente l'état de l'art et les offres de par le monde.

Le deuxième chapitre présente une étude sur le comportement de consommation de l'année dernière en Algérie et propose une méthode de facturation d'électricité.

Dans le troisième chapitre nous présentons en détail tous les composants nécessaires pour la réalisation de ce compteur intelligent.

Le chapitre quatre présente le système de communication qui permet de faire des échanges d'information et de donnée du client vers l'opérateur et de l'opérateur vers le client.

# Chapitre 1 : Etat de l'art

---

## I- Introduction

La tarification de l'électricité comprend un abonnement à un fournisseur d'énergie électrique, un prix dépendant de la consommation (facturée au kilowattheure), des frais d'accès à l'électricité (lors d'une première installation ou d'un changement de fournisseur) ainsi que des taxes locales ou nationales. Les tarifs peuvent différer selon les types d'énergies vendues (renouvelables par exemple), le niveau de tension de raccordement, les plages horaires et les périodes de l'année (selon les contrats). Il existe parfois des réductions ou des aides pour le paiement de l'électricité.

## II- La tarification de l'électricité

Les offres tarifaires de l'électricité varient dans chaque pays. Cela est dû à plusieurs facteurs. Parmi ces facteurs on trouve :

- a) Sources d'électricité : La disponibilité de ressources locales peu coûteuses permet bien évidemment d'offrir aux consommateurs des prix largement inférieurs à la moyenne mondiale.
- b) Structure des coûts : La construction des tarifs se fait par empilement des coûts d'amont en aval le long du réseau électrique : d'abord, les coûts de production, que tous les consommateurs paient. Ensuite, les coûts de transport en très haute tension (400 kV en Europe, jusqu'à 1 000 kV dans les pays où les distances entre lieux de production et lieux de consommation sont très grandes : Canada, Russie). Puis les coûts de transport en haute tension, payés seulement par les clients raccordés à une tension inférieure à la THT. Et ainsi de suite jusqu'à la basse tension, pour les clients résidentiels et professionnels raccordés en basse tension (230 V).
- c) Politiques de développement énergétique : Les gouvernements interviennent fréquemment pour aider, au moins dans ses phases initiales, le développement de nouveaux moyens de production d'électricité.
- d) Configuration économique du pays.
- e) Choix de gouvernance : Selon les orientations politiques, le secteur électrique est contrôlé directement par l'État ou indirectement par les diverses réglementations. Ces choix orientent la structure des tarifs et influencent l'économie [1].

## 1- Les offres Tarifaires en France

La France dépend de la détermination du prix de l'électricité Sur l'option heures pleines heures creuses (HPHC). L'option heures pleines/heures creuses (HPHC) permet de bénéficier d'un prix réduit du kWh pendant les heures où la demande en électricité est la plus faible (généralement la nuit, ce qui explique que le tarif heures pleines et heures creuses soit parfois appelé tarif jour / nuit). Le montant de cette réduction varie d'un fournisseur à un autre.

Le prix d'un kWh d'électricité en France pour les particuliers est de 0.1546 € TTC aux tarifs réglementés d'EDF pour les puissances de compteurs les plus courantes (6 kVA) en option base. Pour la moitié des ménages qui bénéficient de l'option Heures Creuses, le prix baisse à 0.1337 € TTC par kWh pendant les 8 heures creuses mais monte à 0.1781 € TTC le reste du temps. (Pour plusieurs informations voir l'Annexe A) : Grille comparative des prix du kWh d'électricité de Total Direct Energie et EDF pour les autres compteurs

Des fournisseurs alternatifs comme Total Direct Energie, Engie Happ-e ou Vattenfall affichent fréquemment des remises de l'ordre de 10% sur le prix du kWh HT par rapport aux tarifs réglementés d'EDF.

S'y ajoute la partie fixe de la facture, l'abonnement, qui coûte le plus souvent entre 100€ et 150€ par an selon la puissance du compteur et l'option tarifaire (base ou heures creuses) [2].

## 2- Les offres tarifaires en Royaume-Uni (United Kingdom)

En Royaume-Uni les six grands fournisseurs (British Gas, EDF, Eon, Npower, Scottish Power et SSE) donnent aux clients des tarifs économiques (Economie 7 et Economie 10)

- a) **Economie 7** : Les tarifs Economie 7 sont des tarifs moins chers la nuit et plus chers durant la journée. Ils sont principalement destinés aux personnes dotées de radiateurs à accumulation, qui consomment de l'électricité le soir ou la nuit, puis libèrent leur chaleur dans la journée en cas de besoin. Le tarif le moins cher et hors pointe va généralement de minuit à 7h, tandis que le tarif le plus cher couvre le reste de la journée, bien que les heures précises puissent varier selon le fournisseur. Economie 7 est souvent appelé un tarif «temps d'utilisation», car ce que vous payez dépend du moment où vous utilisez l'électricité. Il existe d'autres tarifs de temps d'utilisation tels qu'Economie 10.
- b) **Economie 10** : est un autre tarif énergétique'en fonction de l'heure d'utilisation 'qui donne un tarif de nuit moins cher, mais ce n'est pas aussi courant qu'Economie 7. Similaire à Economie 7, mais offre 10 heures d'électricité moins chère plutôt que sept. Ces heures moins chères sont généralement divisées en



trois périodes différentes. Les temps fonctionnent souvent comme Trois heures creuses entre midi et 15h, Deux heures creuses entre 17h et 19h, Cinq heures creuses entre minuit et 5 heures du matin [3].

(Voir l'Annexe B).

### **3- Les offres Tarifaires en Australie**

La plupart des compteurs résidentiels en Australie sont facturés pour l'électricité selon un tarif unique. Cela signifie qu'ils payent un tarif unique pour l'électricité, quelle que soit l'heure à laquelle cette énergie est consommée. Cependant, certains compteurs sont configurés différemment et ont des tarifs différents. Cela s'explique par le fait qu'il existe différentes manières de facturer l'énergie.

Un tarif TOU est préférable pour les utilisateurs qui consomment la plupart d'énergie dans la nuit. Les tarifs sont fixés par le distributeur du réseau.

Tarif TOU :

Ce tarif est également appelé «Peak / Off-Peak» ce que signifie Pointe/hors Pointe, c'est l'un des tarifs les plus rentables pour le ménage australien moyen. Les tarifs sont fixés comme la suite [4]:

- Pointe : le prix de l'électricité est plus élevé pendant les périodes de 'Pointe', généralement les après-midis et les soirs de la semaine, lorsque la demande d'électricité est la plus élevée.
- Moyen : Le prix de l'électricité est inférieur au tarif de 'Pointe' et supérieur au tarif 'Hors-Pointe'.
- Hors Pointe : le prix de l'électricité est le plus bas lorsque la demande d'électricité est la plus faible.

### **III- Le compteur Intelligent Linky**

Le compteur Linky est un exemple de compteur intelligent utilisé en France, ce compteur est capable de prélever les informations de consommation sans le passage d'un agent, il permet aussi d'enregistrer la valeur maximale de la puissance consommée, avoir un relevé à distance, mesurer de la durée de coupure et avoir un affichage évolué.



Figure 1-1 : Le compteur intelligent Linky

## 1- Comment remonter les informations

Linky fait partie de la catégorie des compteurs communicants (AMM), c'est-à-dire pouvant être interrogés et actionnés à distance. Pour cela, Linky utilise la technologie CPL sur le réseau basse tension, comme premier niveau de communication. Un deuxième niveau de communication est assuré via le réseau de téléphonie mobile GPRS ou Edge entre le concentrateur (installé dans un poste de transformation et de distribution HTA/BT) et le système central. À terme, si les 35 millions de compteurs sont remplacés, il y aura environ 700 000 concentrateurs installés, éléments d'un système d'information (SI) répartis Linky, avec un SI central et un SI local dans chaque concentrateur. Cette génération de compteurs communicants est présentée comme une des étapes vers les réseaux intelligents, appelés smart grids en anglais [5].

## 2- Quelles sont les données recueillies par le compteur

Les compteurs communicants sont capables de relever et de transmettre des données de consommation d'électricité détaillées ainsi que des données techniques, telles que le type de compteur, le numéro de série, la date et l'heure ou encore le journal des coupures brèves ou longues. Voici les principales informations utiles au consommateur :

### a) La courbe de charge

L'information la plus importante et la plus utile relevée par le compteur Linky est la courbe de charge. Elle est construite à partir de relevés réguliers de la consommation électrique d'un consommateur. Linky relève et stocke des données quotidiennes, toutes les heures ou toutes les demi-heures sur votre demande. Avec ces relevés par heure ou par demi-heure, on peut créer une courbe graphique qui

permet d'identifier facilement les périodes de forte consommation dans la journée, dans le mois ou dans l'année.

b) Les index quotidiens

Par défaut, les index sont relevés à distance et transmis une fois par jour, entre minuit et 6 h du matin, à Enedis le gestionnaire du réseau. L'information transmise contient plusieurs index pour les abonnements à tarifs différenciés, comme par exemple l'abonnement heures pleines / heures creuses. Chaque jour, ces deux index sont relevés et transmis.

Le consommateur a accès par défaut, sur le site Internet d'Enedis, à sa consommation par an, par mois et par jour. Les informations plus fines (la courbe de charge par heures ou par demi-heures) ne sont relevées que si le consommateur en fait la demande expresse.

c) La puissance maximale

Chaque jour, le consommateur pourra voir la puissance maximale atteinte pendant la journée sur son espace personnel. Cette information permet au consommateur de mieux adapter la puissance à laquelle il souscrit.

Le compteur Linky permet aux fournisseurs de proposer des offres d'électricité au pas de 1 kVA. Ainsi, un consommateur qui constate qu'il ne dépasse jamais une puissance 7 kVA (même en cas de grand froid, quand tous les appareils de chauffage sont allumés), pourra souscrire une puissance de 7 kVA, moins chère que celle de 9 kVA à laquelle il est obligé de souscrire aujourd'hui. Il n'existe pas encore d'offre de fournisseur permettant de bénéficier de cette possibilité.

Si elles sont beaucoup plus détaillées qu'auparavant, les données recueillies par le compteur Linky restent des données agrégées et globales. Il est impossible d'en déduire le détail des consommations des différents appareils du foyer. Rappelons que ces données détaillées ne sont disponibles qu'à la demande et après consentement exprimé du consommateur. La donnée par défaut est quotidienne et transmise pendant la nuit.

### **3- La sécurité de ces données**

Toutes les données de consommation sont cryptées. Le dispositif de sécurité mis en place est conforme au référentiel de sécurité certifié par l'Agence nationale de sécurité des systèmes d'information (ANSSI).

Ce cryptage a pour objectif de préserver le système des attaques malveillantes notamment sur le respect de la vie privée. Les clés de chiffrement sont configurées lors de l'installation du compteur par le technicien et ne sont pas échangées par la suite ce qui complique le piratage. Chaque compteur dispose de sa propre clé aléatoire.

Le système détecte les tentatives d'intrusion physique (par exemple, une ouverture de capot du compteur). Dans ce cas, les clés de cryptage sont automatiquement effacées et les données ne sont donc plus accessibles [6].

## **IV- Perspectives d'utilisation de compteurs intelligents en Algérie**

### **1- A court terme**

Les compteurs intelligents sont capables de relever et de transmettre des données de consommation d'électricité détaillées ainsi que des données techniques, telles que le type de compteur, le numéro de série, la date et l'heure. Celui va minimiser le cout et l'effort des employées qui se déplace pour obtenir ces données.

### **2- A moyen Terme**

La collecte des données de consommation en fonction de la date et l'heure va permettre d'étudier le comportement des consommateurs tout au long de la journée et à différentes saisons. Cela permet de mieux comprendre le réseau et permet de répondre efficacement à la demande électrique. Après cette étude, l'utilisation de ces compteurs vise à inciter les clients à moins consommer, par l'établissement de prix différenciés en temps réel, aux heures de crêtes (sur-tarification) prévisibles selon les heures de la journée et la saison. Cela est sensé encourager les clients à consommer moins d'énergie et permettre des économies globales d'énergies dans une perspective de développement durable.

### **3- Quel accueil la société de consommation algérienne va-t-elle-leur donner**

Le client final d'électricité en Algérie devrait pouvoir être incité à réduire ses consommations, grâce à une meilleure information. Une information en temps réel, plus riche et plus fréquente sur la consommation, pourrait en effet assurer un rôle de sensibilisation et inciter le consommateur à mieux maîtriser ses usages.

Un rapport britannique datant de 2006, qui a compilé plusieurs retours d'expérience (USA, Canada, Scandinavie, Pays-Bas et Royaume Uni) montre que l'affichage en temps réel des consommations peut générer une économie d'électricité de 5% à 15 %.

## **V- Conclusion**

Nous avons présenté dans ce chapitre les systèmes utilisée par différentes pays pour la facturation et qui est basée sur des tarifs qui change en fonction de la demande des clients.

Ensuite, nous avons présenté le compteur Linky et ses services. Ainsi, nous avons donné des perspectives d'utilisation de compteurs intelligents en Algérie.

## **Chapitre 2 : Etudes théoriques pour une tarification**

---

### **I- Introduction**

Les compteurs intelligents sont actuellement le nouveau sujet brûlant dans l'industrie de l'énergie. Ils vont jouer un rôle énorme dans l'avenir de notre marché de l'énergie et dans la façon dont nous gérons, chez nous, notre consommation d'énergie. Nous avons déjà des téléviseurs et des téléphones intelligents, des réfrigérateurs-congélateurs intelligents et des thermostats intelligents. Alors, qu'est-ce qui rend un compteur d'électricité intelligent ?

### **II- Les compteurs intelligents**

Un 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. 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 ou par internet comme nous le ferons dans ce projet.

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 vers les réseaux intelligents [7].

Le client obtiendra également un affichage à domicile avec son compteur intelligent, qu'il lui permettra de visualiser son consommation d'énergie en temps réel. Il pourra également revenir sur son historique et voir quelle énergie a utilisé dans le passé.

Les compteurs intelligents aident le client à devenir plus écoénergétique. Il sera en mesure de déterminer la quantité d'énergie consommée par chaque appareil et d'ajuster son utilisation en conséquence [8].

### III- Système de tarification de SONELGAZ

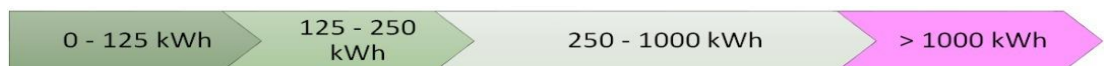
SONELGAZ utilise actuellement des compteurs d'énergie numériques et utilise la méthode des tranches pour calculer les factures d'électricité. Cela signifie que plus la consommation est élevée, plus les tranches suivantes seront appliquées.

Dans la deuxième partie de ce chapitre on va réaliser une méthode de facturation en fonction des courbes des charges pour diminuer le problème des surcharges selon la méthode des tarifications multiples.

#### 1- La méthode actuelle de calcul du SONELGAZ (Consommation/tranche) :

- a) Index ancien : C'est l'index affiché par le compteur lors du relevé du trimestre précédent.
- b) Index nouveau : Index affichés par le compteur le jour de la relève trimestriel de facturation de la nouvelle période.
- c) Différence entre les index : La différence entre l'index nouveau et l'index ancien.
- d) Coefficient : Sa valeur pour l'électricité est de 1. Son utilité sert à calculer la valeur réelle de votre consommation d'énergie. C'est-à-dire qu'il faut multiplier la différence des index par le coefficient pour avoir votre consommation électrique (kWh).
- e) Consommation/tranche : La tarification progressive de l'électricité en Algérie

SONELGAZ divise notre consommation en 4 tranches, pour les deux premières tranches elle est égale à 125 kWh pour la troisième tranche elle est égale à 750 kWh et la quatrième tranche est laissé pour ce qui reste, il faut que la distribution soit par ordre, c'est-à-dire qu'on ne peut pas passer à la deuxième tranche si on n'atteint pas les 125 kWh et ainsi de suite.



Il est différent pour chaque tranche :

Première tranche = 1,7787 DZD

Deuxième tranche = 4,1789 DZD

Troisième tranche = 4,812 DZD

Quatrième tranche = 5,4 DZD

Donc le prix unitaire commence par 1,7787 et augmente chaque fois que la consommation augmente et passe à la deuxième tranche. Plus on consomme plus on paye. Cette méthode est utilisée pour pousser le client à réduire sa consommation.

- f) Montant : Multiplier le prix unitaire par la consommation pour chaque tranche.
- g) TVA (taxe sur la valeur ajoutée)

Le taux de la TVA pour la première et la deuxième tranche est de 9 %. Le taux de 19 % s'applique pour les tranches 3 et 4 pour les clients résidentiels (les particuliers) et les tranches 2 et 3 pour les clients non résidentiels.

- h) Montant TTC : On additionne le montant HT et la TVA pour avoir le montant en TTC [9].

#### **IV- Proposition d'un nouveau plan de tarification**

Le système précédent de tarification (consommation par tranche) peut diminuer la consommation des clients durant le trimestre, mais elle ne peut pas agir sur la consommation quotidienne comme on va voir dans cette étude.

Le but de cette étude est de réaliser un plan de facturation basé sur la demande quotidienne en Algérie. Pour ce faire, on prend deux jours de chaque mois de l'année dernière 2019, un jour de semaine et un autre de weekend.

Notre source d'information est le site web de l'opérateur système électrique en Algérie : **www.os.dz**.

Les détails de consommation et les tranches de prix seront présentés dans les figures (Figure 2 jusqu'à figure 25) sur les pages suivantes.

Le prix est composé d'un certain nombre de coûts clés éléments de production et de transport d'électricité, y compris le coût de l'énergie dans marché de gros, les coûts de transport et de distribution dans les réseaux, les coûts de l'environnement, etc.

Les tranches sont divisées de la même façon pour toute l'Algérie à cause du manque des données de consommation propre à chaque wilaya. Nos données représentent la consommation en général.

Nous proposons des plages horaires pour chaque mois de l'année en fonction du profil de consommation journalier. Ces plages horaires sont liées aux 4 tranches de prix.

### 1. Janvier :

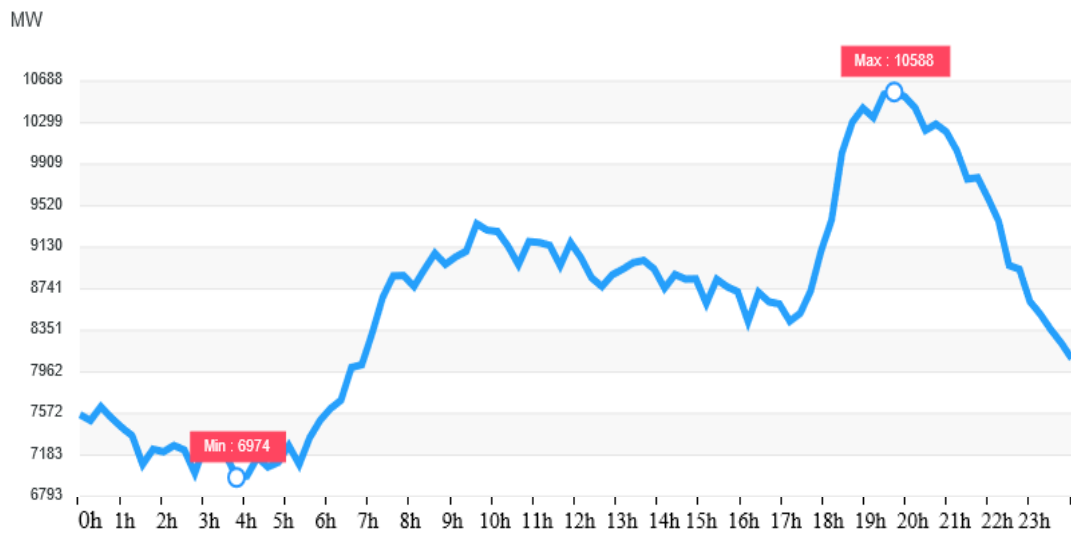


Figure 2-1 : Courbe de charge durant un jour ouvrable du mois de janvier



Figure 2-2 : Courbe de charge durant un jour de weekend du mois de janvier

Tableau 2-1 : Proposition des plages horaires d'un jour du mois de janvier

	Jour de semaine 13/01/2019	Jour de weekend 18/01/2019	Prix
Tarif 1	00:00 – 06:30	01:00 – 06:30	Min 5 DZD
Tarif 2	06:30 – 17:30 22:00 – 00:00	06:30 – 17:30 23:00 – 01:00	10 DZD
Tarif 3	17:30 – 18:30 20:30 – 22:00	17:30 – 18:30 21:00 – 23:00	15 DZD
Tarif 4	18:30 – 20:30	18:30 – 21:00	Max 20 DZD



## 2. Février :

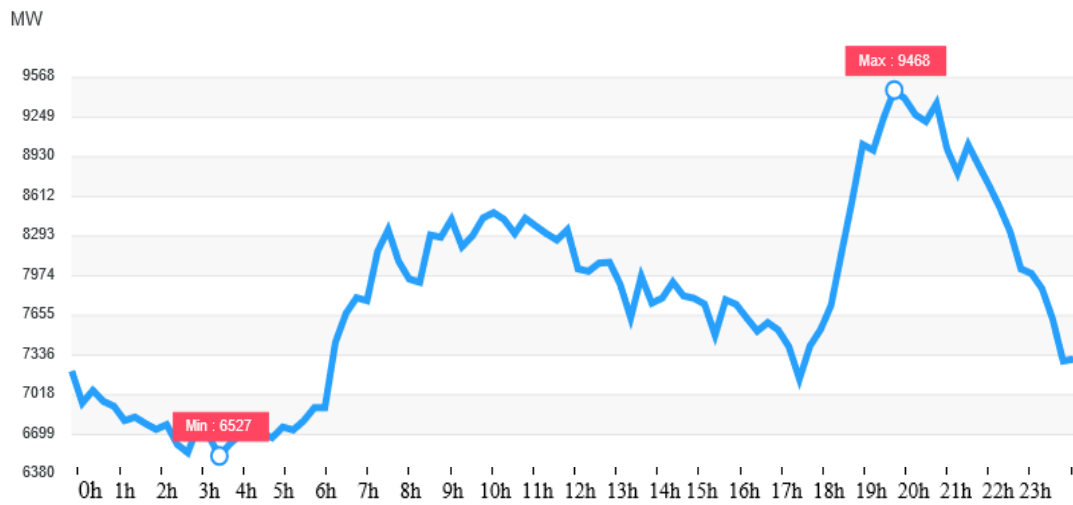


Figure 2-3 : Courbe de charge durant un jour ouvrable du mois de février

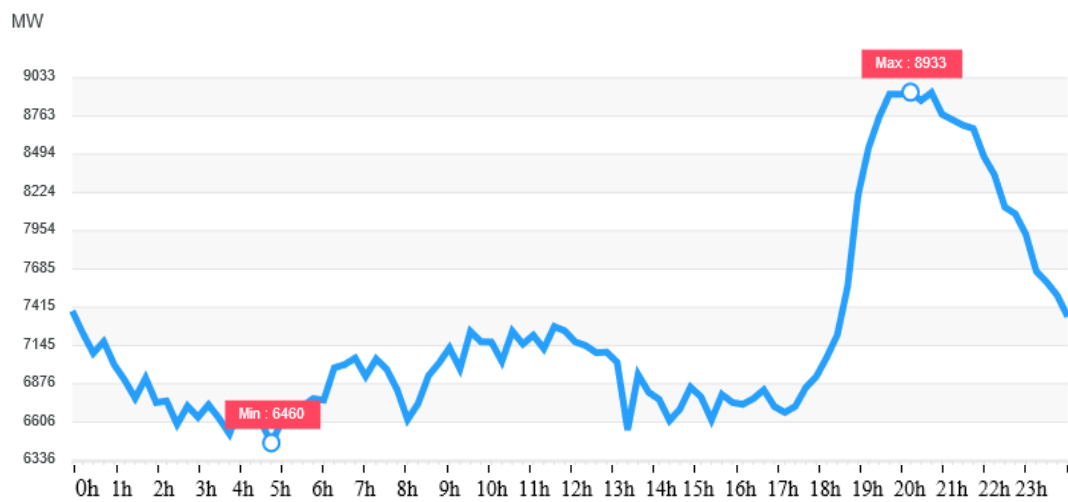


Figure 2-4 : Courbe de charge durant un jour de weekend du mois de février

Tableau 2-2 : Proposition des plages horaires d'un jour de mois de février

	Jour de semaine 10/02/2019	Jour de weekend 15/02/2019	Prix
Tarif 1	00:00 – 06:30	00:00 – 05:45	Min 5 DZD
Tarif 2	06:30 – 08:30 13:00 – 18:00 22:00 – 00:00	05:45 – 08:30 13:15 – 17:15 22:30 – 00:00	10 DZD
Tarif 3	08:30 – 13:00 18:00 – 19:00 21:00 – 22:00	08:30 – 13:15 17:15 – 19:00 21:45 – 22:30	15 DZD
Tarif 4	19:00 – 21:00	19:00 – 21:45	Max 20 DZD

### 3. Mars :

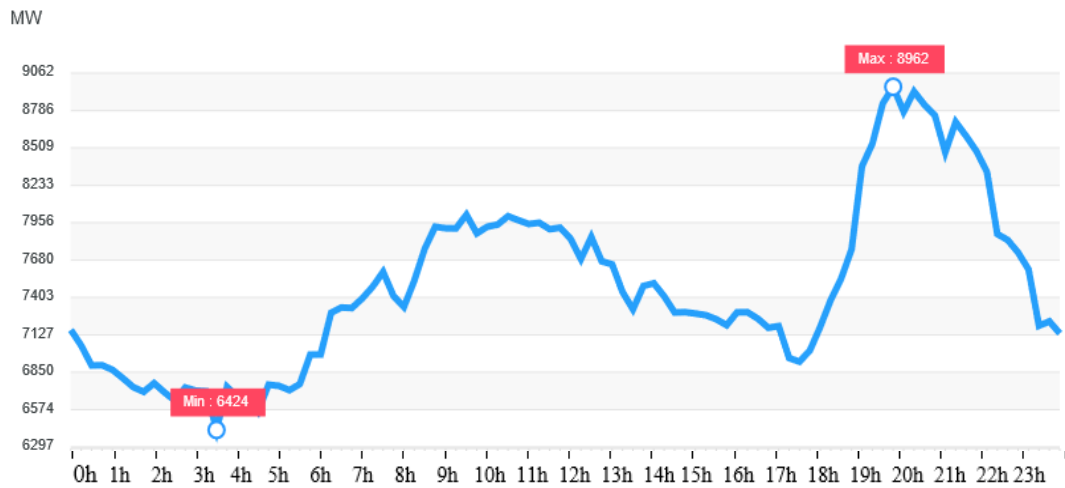


Figure 2-5 : Courbe de charge durant un jour ouvrable du mois de mars



Figure 2-6 : Courbe de charge durant un jour de weekend du mois de mars

Tableau 2-3 : Proposition des plages horaires d'un jour de mois de mars

	Jour de semaine 10/03/2019	Jour de weekend 15/03/2019	Prix
Tarif 1	00:00 – 06:00	00:00 – 06:00	Min 5 DZD
Tarif 2	06:00 – 18:30 22:30 – 00:00	06:00 – 18:30 22:00 – 00:00	10 DZD
Tarif 3	18:30 – 19:00 21:30 – 22:30	18:30 – 19:00 21:00 – 22:00	15 DZD
Tarif 4	19:00 – 21:30	19:00 – 21:00	Max 20 DZD

#### 4. Avril :

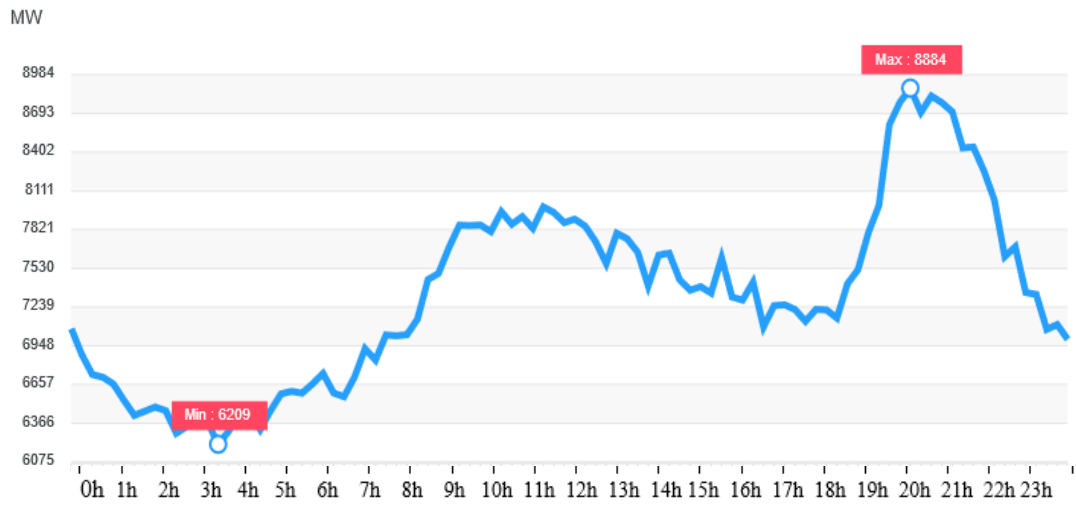


Tableau 2-4 : Proposition des plages horaires d'un jour de mois d'avril

	Jour de semaine 08/04/2019	Jour de weekend 05/04/2019	Prix
Tarif 1	00:00 – 06:30	01:00 – 09:00	Min 5 DZD
Tarif 2	06:30 – 19:00 22:00 – 00:00	09:00 – 19:00 22:30 – 01:00	10 DZD
Tarif 3	19:00 – 19:30 21:00 – 22:00	19:00 – 19:30 21:00 – 22:30	15 DZD
Tarif 4	19:30 – 21:00	19:30 – 21:00	Max 20 DZD

## 5. Mai :

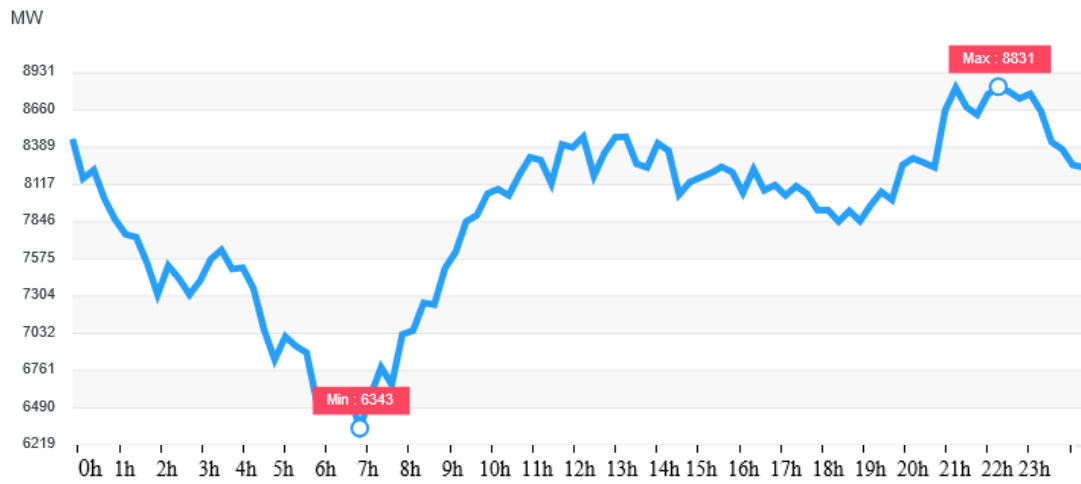


Figure 2-9 : Courbe de charge durant un jour ouvrable du mois de mai



Figure 2-10 : Courbe de charge durant un jour de weekend du mois de mai

Tableau 2-5 : Proposition des plages horaires d'un jour du mois de mai

	Jour de semaine 12/05/2019	Jour de weekend 10/05/2019	Prix
Tarif 1	04:30 – 08:00	04:30 – 09:30	Min 5 DZD
Tarif 2	01:30 – 04:30 08:00 – 09:30	01:30 – 04:30 09:30 – 12:00	10 DZD
Tarif 3	23:00 – 01:30 09:30 – 20:30	12:00 – 19:30 23:00 – 01:30	15 DZD
Tarif 4	20:30 – 23:00	19:30 – 23:00	Max 20 DZD

## 6. Juin :

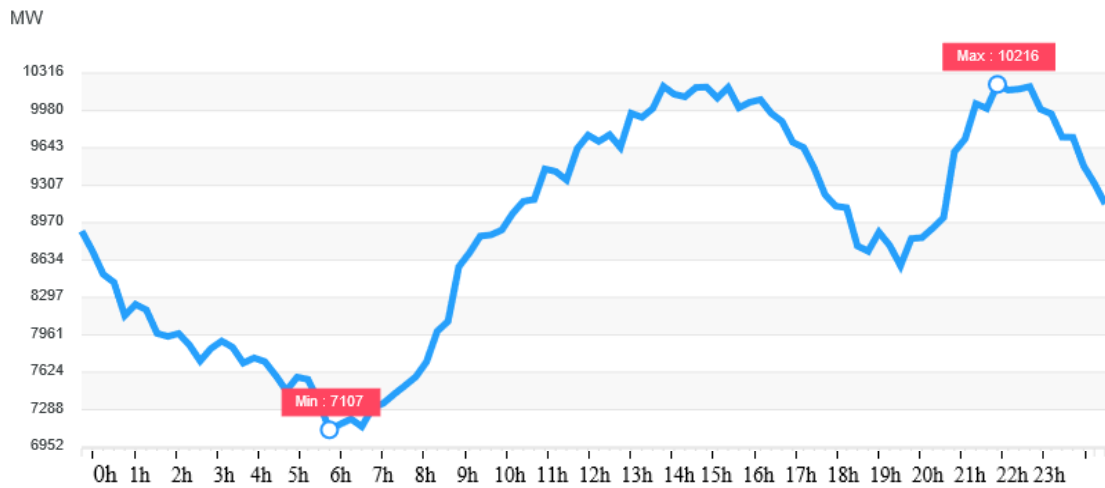


Figure 2-11 : Courbe de charge durant un jour ouvrable du mois de juin

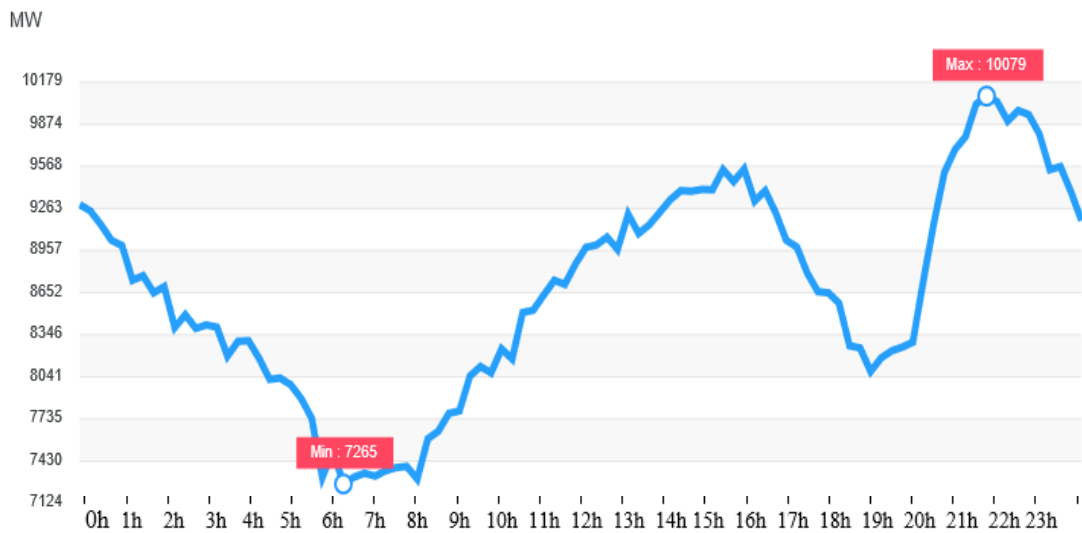


Figure 2-12 : Courbe de charge durant un jour de weekend du mois de juin

Tableau 2-6 : Proposition des plages horaires d'un jour du mois de juin

	Jour de semaine 09/06/2019	Jour de weekend 14/06/2019	Prix
Tarif 1	01:30 – 08:30	04:00 – 09:30	Min 5 DZD
Tarif 2	08:30 – 10:00 23:00 – 01:30	01:30 – 04:00 09:30 – 12:00	10 DZD
Tarif 3	10:00 – 13:00 16:30 – 20:30	12:00 – 20:00 23:00 – 01:30	15 DZD
Tarif 4	13:00 – 16:30 20:00 – 23:00	20:00 – 23:00	Max 20 DZD

## 7. Juillet :

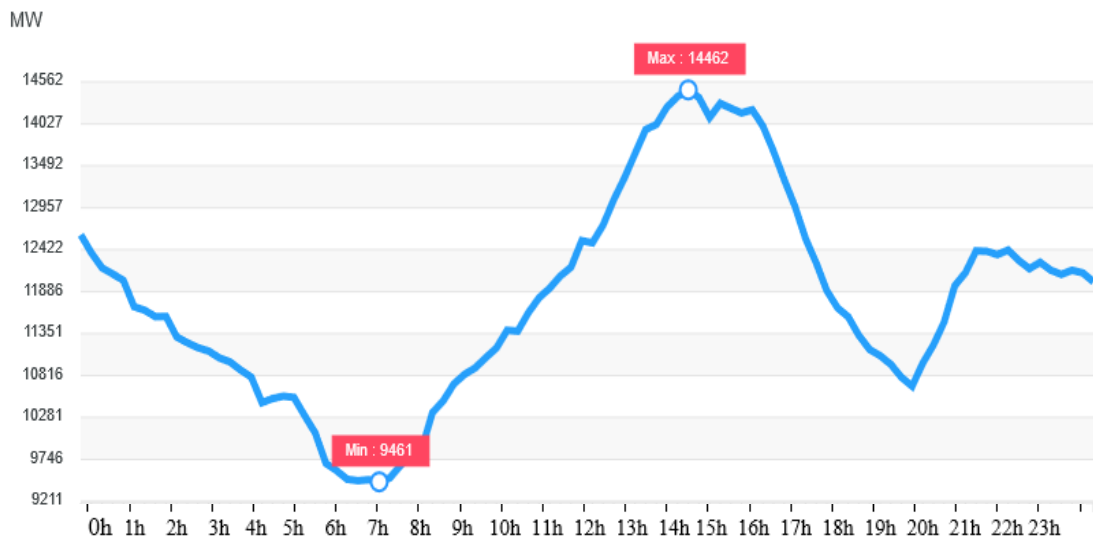


Figure 2-13 : Courbe de charge durant un jour ouvrable du mois de juillet

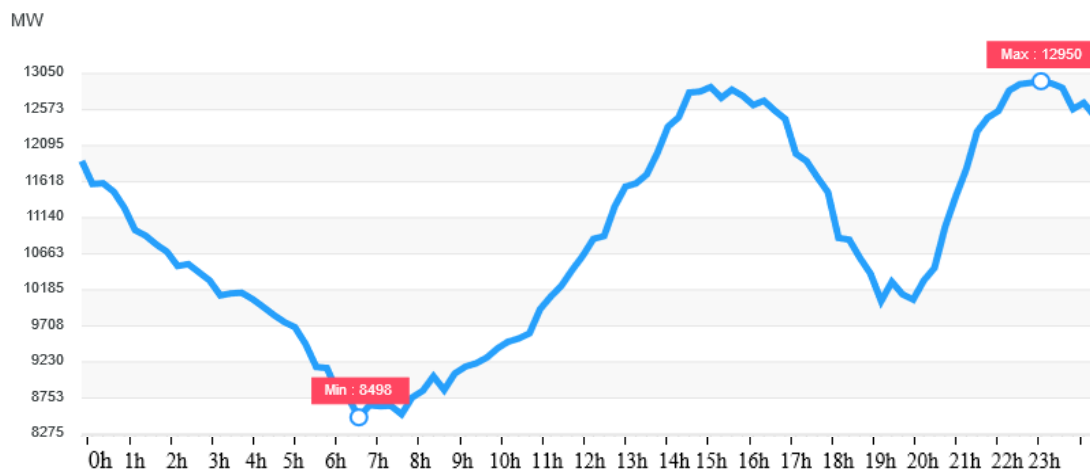


Figure 2-14 : Courbe de charge durant un jour de weekend du mois de juillet

Tableau 2-7 : Proposition des plages horaires d'un jour du mois de juillet

	Jour de semaine 14/07/2019	Jour de weekend 12/07/2019	Prix
Tarif 1	04:00 – 8:30	04:00 – 11:00	Min 5 DZD
Tarif 2	08:30 – 11:00 17:30 – 20:30 01:00 – 04:00	01:00 – 04:00 11:00 – 12:30 17:30 – 20:30	10 DZD
Tarif 3	11:00 – 13:00 16:30 – 17:30 20:30 – 01:00	12:30 – 13:30 16:30 – 17:30 20:30 – 21:30 23:00 – 01:00	15 DZD
Tarif 4	13:00 – 16:30	21:30 – 23:00 13:30 – 16:30	Max 20 DZD

## 8. Août :

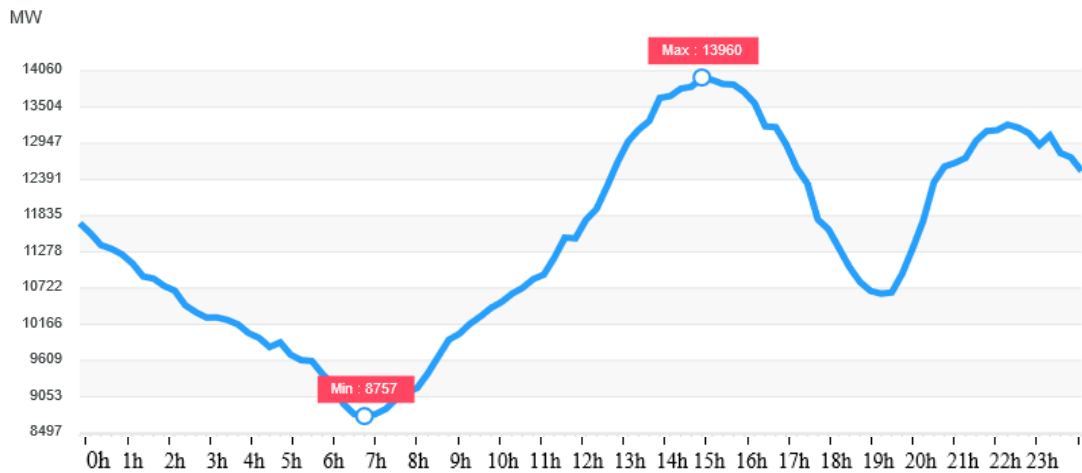


Figure 2-15 : Courbe de charge durant un jour ouvrable du mois d'août

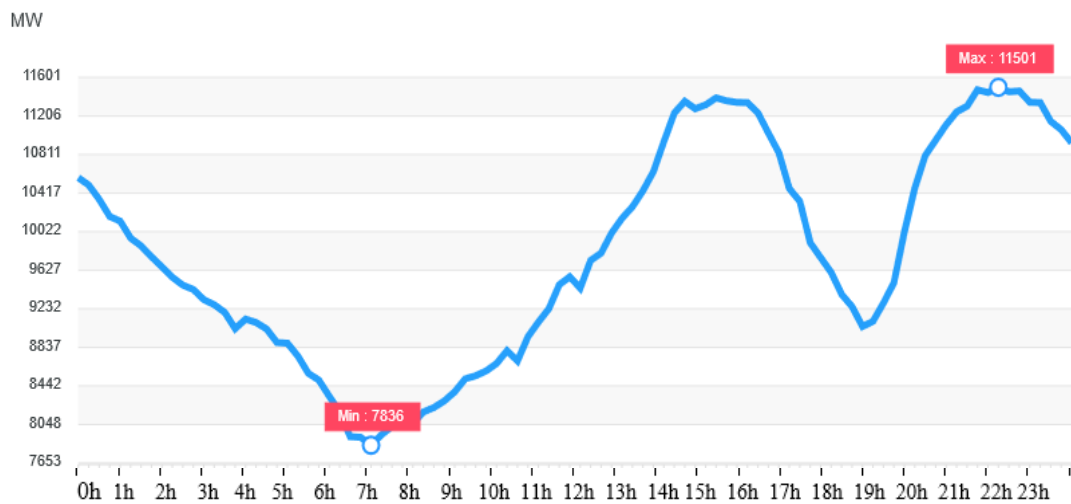


Figure 2-16 : Courbe de charge durant un jour weekend du mois d'août

Tableau 2-8 : Proposition des plages horaires d'un jour du mois d'août

	Jour de semaine 18/08/2019	Jour de weekend 16/08/2019	Prix
Tarif 1	03:30 – 09:30	05:00 – 10:30	Min 5 DZD
Tarif 2	09:30 – 12:00 17:30 – 20:30 01:00 – 03:30	01:00 – 05:00 10:30 – 12:30 17:30 – 19:30	10 DZD
Tarif 3	12:00 – 13:30 16:30 – 17:30 20:30 – 01:00	12:30 – 14:00 16:30 – 17:30 19:30 – 20:30 23:30 – 01:00	15 DZD
Tarif 4	13:30 – 16:30	14:00 – 16:30 20:30 – 23:30	Max 20 DZD

## 9. Septembre :

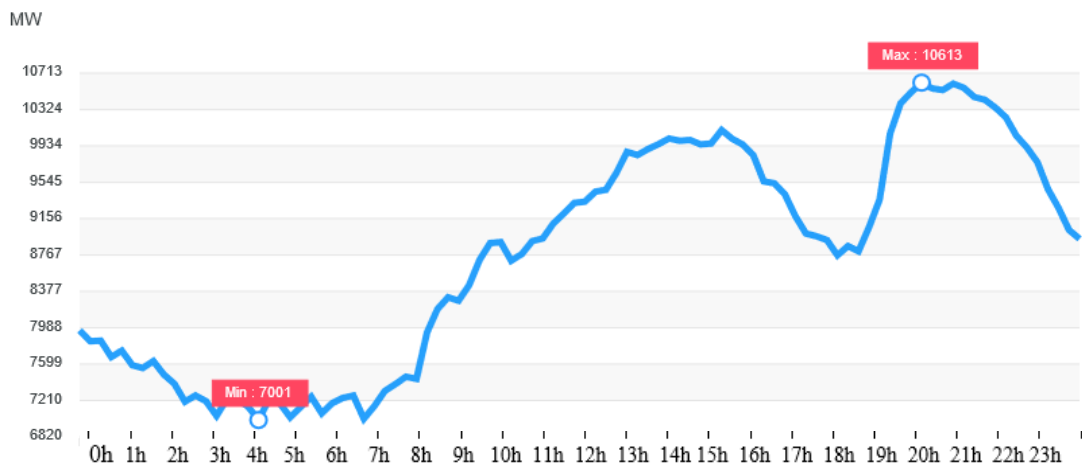


Figure 2-17 : Courbe de charge durant un jour ouvrable du mois de septembre

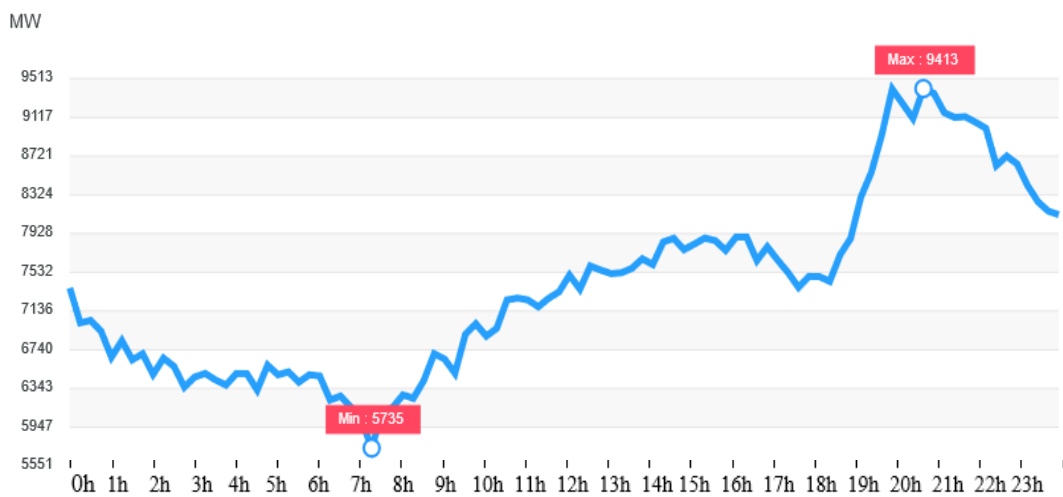


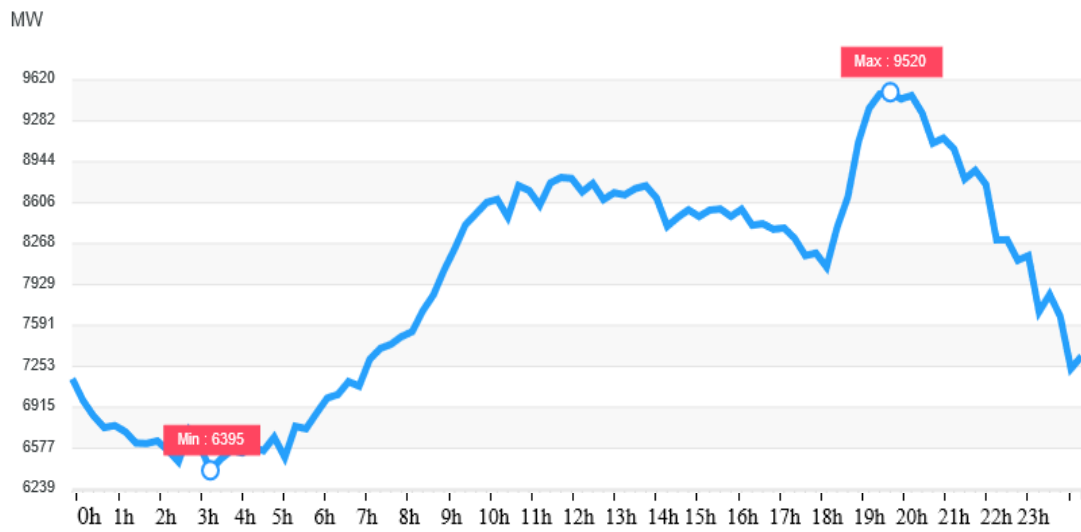
Figure 2-18 : Courbe de charge durant un jour de weekend du mois de septembre

Tableau 2-9 : Proposition des plages horaires d'un jour du mois de septembre

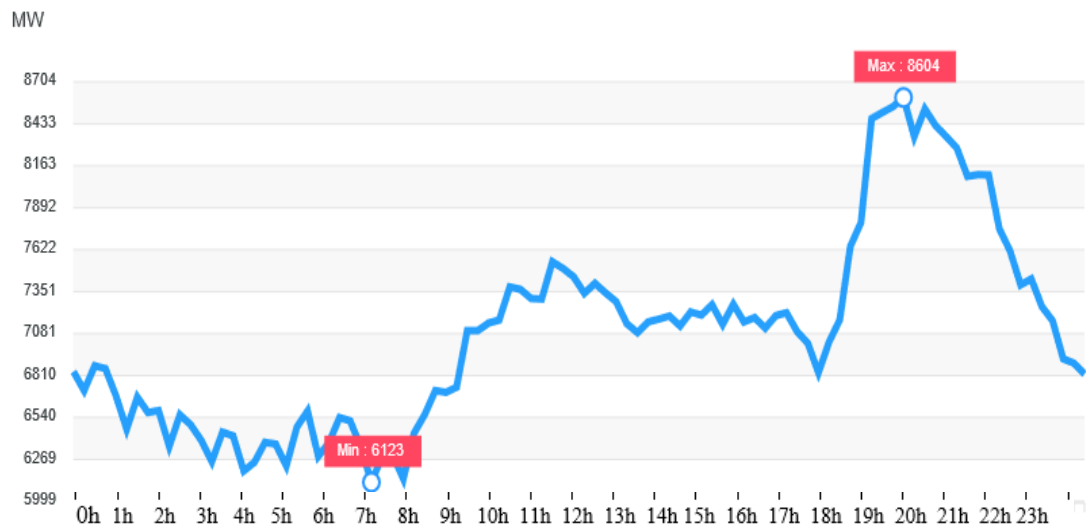
	Jour de semaine 15/09/2019	Jour de weekend 13/09/2019	Prix
Tarif 1	00:00 – 08:00	01:00 – 09:00	Min 5 DZD
Tarif 2	08:00 – 11:30 16:30 – 19:00 23:00 – 00:00	09:00 – 18:30 00:00 – 01:00	10 DZD
Tarif 3	11:30 – 16:30 19:00 – 19:30 21:30 – 23:00	18:30 – 19:30 21:00 – 00:00	15 DZD
Tarif 4	19:30 – 21:30	19:30 – 21:00	Max 20 DZD



**10. Octobre :**



**Figure 2-19 : Courbe de charge durant un jour ouvrable du mois d'octobre**



**Figure 2-20 : Courbe de charge durant un jour de weekend du mois d'octobre**

**Tableau 2-10 : Proposition des plages horaires d'un jour du mois d'octobre**

	Jour de semaine 13/10/2019	Jour de weekend 11/10/2019	Prix
Tarif 1	00:00 – 07:00	01:00 – 09:00	Min 5 DZD
Tarif 2	07:00 – 09:30 15:00 – 18:00 21:30 – 00:00	09:00 – 18:30 00:00 – 01:00	10 DZD
Tarif 3	09:30 – 15:00 18:00 – 18:30 20:30 – 21:30	18:30 – 19:30 21:00 – 00:00	15 DZD
Tarif 4	18:30 – 20:30	19:30 – 21:00	Max 20 DZD

### 11. Novembre :

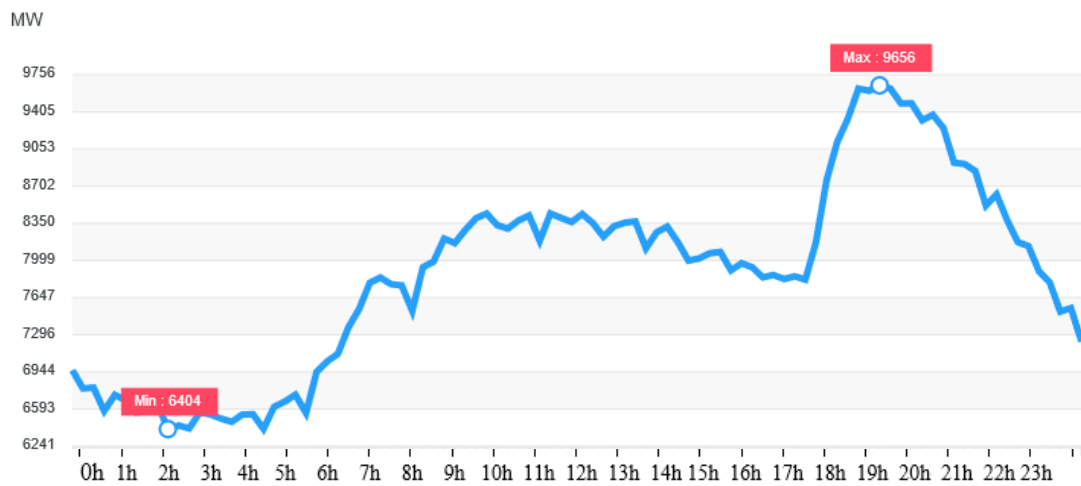


Figure 2-21 : Courbe de charge durant un jour ouvrable du mois de novembre

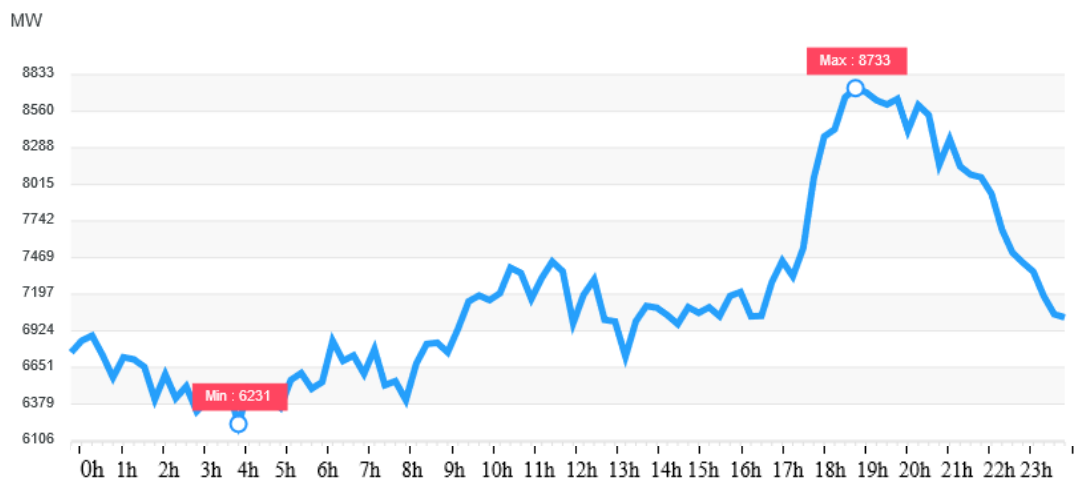


Figure 2-22 : Courbe de charge durant un jour de weekend du mois de novembre

Tableau 2-11 : Proposition des plages horaires d'un jour du mois de novembre

	Jour de semaine 17/11/2019	Jour de weekend 15/11/2019	Prix
Tarif 1	00:00 – 06:00	00:00 – 09:00	Min 5 DZD
Tarif 2	06:00 – 08:00 14:00 – 17:30 22:00 – 00:00	09:00 – 17:30 22:00 – 00:00	10 DZD
Tarif 3	08:00 – 14:00 17:30 – 18:30 20:00 – 22:00	17:30 – 18:30 20:00 – 22:00	15 DZD
Tarif 4	18:30 – 20:00	18:30 – 20:00	Max 20 DZD

## 12. Décembre :

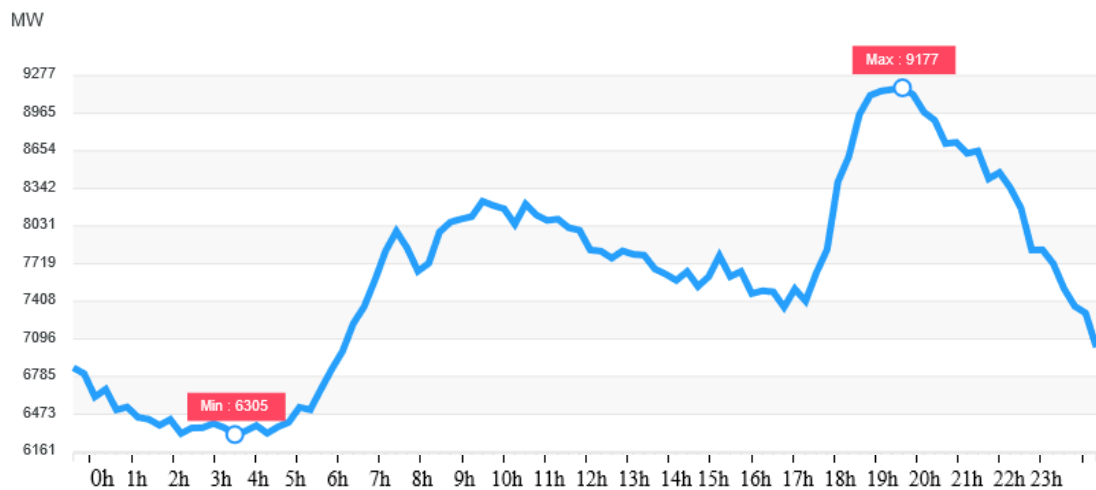


Figure 2-23 : Courbe de charge durant un jour ouvrable du mois de décembre

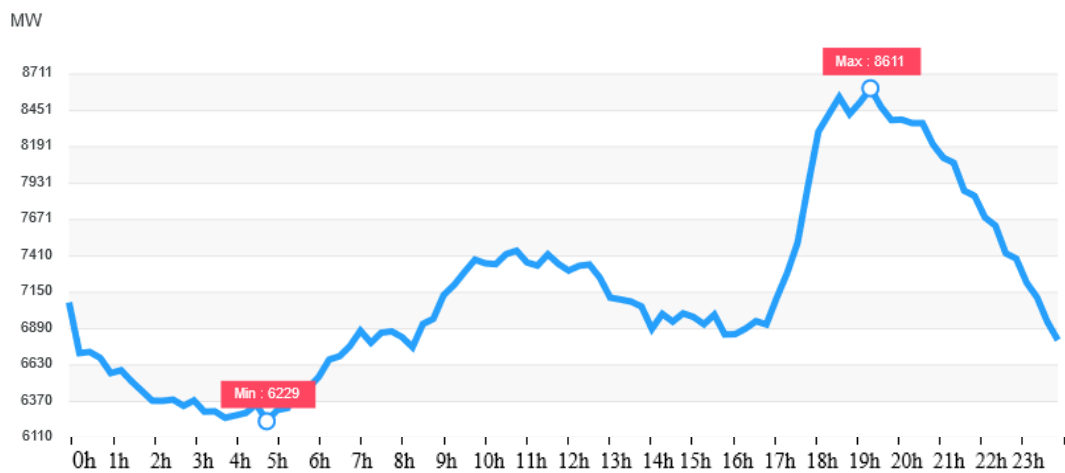


Figure 2-24 : Courbe de charge durant un jour de semaine du mois de décembre

Tableau 2-12 : Proposition des plages horaires d'un jour du mois de décembre

	Jour de semaine 15/12/2019	Jour de weekend 13/12/2019	Prix
Tarif 1	00:00 – 06:00	00:00 – 06:00	Min 5 DZD
Tarif 2	06:00 – 08:00 14:00 – 17:30 22:00 – 00:00	06:00 – 09:00 13:00 – 17:30 22:30 – 00:00	10 DZD
Tarif 3	08:00 – 14:00 17:30 – 18:30 20:00 – 22:00	09:00 – 13:00 17:30 – 18:00 20:00 – 22:30	15 DZD
Tarif 4	18:30 – 20:00	18:00 – 20:00	Max 20 DZD

Après la remarque des graphes de consommation en Algérie, On propose le système de tarification suivant :

- **Bas** (Tarif 1) : le prix de l'électricité est le plus bas lorsque la demande d'électricité est la plus faible.
- **Modéré** (Tarif 2) : Le prix de l'électricité est inférieur au tarif 3 et tarif 4 et supérieur au tarif 1.
- **Fort** (Tarif 3) : Le prix de l'électricité est inférieur au tarif 4 et supérieur au tarif 1 et tarif 2.
- **Très fort** (Tarif 4) : le prix de l'électricité est plus élevé pendant les périodes de surcharge, généralement les après-midi et les soirs de la semaine, lorsque la demande d'électricité est la plus élevée.

L'intérêt pour ce dispositif multi-tarifaire est de pouvoir limiter les surcoûts de production inévitables pendant les pics de consommation. Ces pics de consommation ont plusieurs problèmes, tel que les pertes des lignes lors de transport d'énergie. Mais le grand problème c'est qu'un pic de consommation représente un démarrage d'une centrale pour répondre à cette consommation. Nous avons vu dans les graphes qu'un pic peut durer dans l'intervalle de deux heures, c'est catastrophique ! Avoir une centrale installée de 2 ou 3 GW qu'il faut l'acheter et l'installer et besoin d'un entretien et une maintenance, et tout ça pour servir deux heures par jour. C'est pour ce problème, notre objectif est de laisser ces courbes de charge afin d'arriver à une valeur moyen de demande.

En reportant le plus possible leur consommation vers les heures creuses, les clients contribuent donc à lisser leur consommation globale sur 24 heures et diminuer l'incidence des pics de consommation. L'opérateur les récompense en les faisant bénéficier d'un prix du kWh d'électricité moins cher pendant les heures creuses.

Pour une implémentation sur microcontrôleur des tarifs de consommation et des plages horaires, nous allons programmer le compteur intelligent afin de tenir compte des variations de prix toutes les 15 minutes autrement dit, la journée sera divisée en 96 plages horaires (24 heures x 4 quarts d'heure).

L'Annexe C représente un tableau qui résume la variation des tarifs proposés en fonction des plages horaires pour deux jours (jour de semaine et jour de weekend) de chaque mois de l'année.

## V- Conclusion

Nous avons présenté dans ce chapitre la méthode actuelle utilisée par SONELGAZ pour la facturation et qui est basée sur une tarification par tranche de consommation (kWh).

Ensuite, nous avons proposé une étude sur les courbes de charge en Algérie basée sur les données de l'année 2019

En fonction de ces données nous proposons une nouvelle méthode de tarification multiple basée sur les plages horaires. Elle comprend quatre tarifs (1 à 4) mais peut en inclure plus.

Nous allons appliquer ce système de tarification à notre compteur.

# Chapitre 3 : Réalisation de prototype de compteur

## I- Introduction

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 la première porte sur la mesure des données de consommation et la seconde porte sur la communication des données et des tarifs.

Dans ce chapitre, nous expliquerons la partie mesure des données de consommation comme la tension, le courant et la puissance.

Afin de réaliser ce projet nous avons besoin des capteurs pour obtenir les mesures et un microcontrôleur pour traiter les mesures et calculer la consommation.

## II- Description de montage

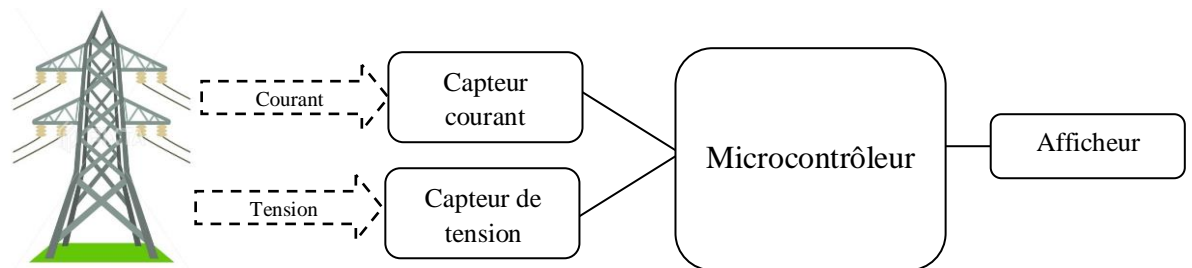


Figure 3-1 : Schéma descriptif de montage

## III- Capteur de courant

Dans ce compteur nous avons utilisé le capteur de courant ACS758:

### 1- Description d'ACS758

La famille Allegro™ ACS758 de circuits intégrés de capteurs de courant fournit des solutions économiques et précises pour la détection de courant AC ou DC. Les applications typiques incluent le contrôle du moteur, la détection et la gestion de la charge, l'alimentation et le contrôle du convertisseur CC-CC, le contrôle de l'onduleur et la détection des défauts de surintensité.

Le dispositif se compose d'un circuit de Hall linéaire de précision à faible décalage avec un chemin de conduction en cuivre situé près de la filière. Le courant appliqué traversant ce chemin de conduction en cuivre génère un champ magnétique que le CI Hall convertit en une tension proportionnelle. La précision de l'appareil est optimisée grâce à la proximité du signal magnétique avec le transducteur Hall. Une tension de sortie précise et proportionnelle est fournie par le circuit intégré BiCMOS Hall à

faible décalage et stabilisé par hachage, qui est programmé pour la précision en usine.

L'immunité de haut niveau aux conducteurs électriques  $dV / dt$  et aux champs électriques parasites, offerte par la technologie de blindage intégrée exclusive d'Allegro, fournit une ondulation de tension de sortie faible et une dérive de décalage faible dans les applications haute tension. La sortie de l'appareil a une pente positive ( $> VCC / 2$ ) lorsqu'un courant croissant circule à travers le chemin de conduction en cuivre primaire (de la borne 4 à la borne 5), qui est le chemin utilisé pour l'échantillonnage du courant. La résistance interne de ce chemin conducteur est typique de  $100 \mu\Omega$ , fournissant une faible perte de puissance. L'épaisseur du conducteur en cuivre permet la survie de l'appareil dans des conditions de surintensité élevée. Les bornes du chemin conducteur sont électriquement isolées des fils de signal

### Caractéristiques

- 3,0 à 5,5 V, fonctionnement à alimentation unique
- Bande passante typique de 120 kHz
- Temps de montée de sortie de  $3 \mu s$  en réponse au courant d'entrée pas à pas
- Tension de sortie proportionnelle aux courants alternatifs ou continus
- Découpé en usine pour plus de précision
- Tension de décalage de sortie extrêmement stable
- Hystérésis magnétique presque nul

(Broches 1 à 3). Cela permet à la famille ACS758 de circuits intégrés de capteurs d'être utilisée dans des applications nécessitant une isolation électrique sans utiliser d'opto-isolateurs ou d'autres techniques d'isolation coûteuses.

L'appareil est entièrement calibré avant d'être expédié de l'usine. La famille ACS758 est sans plomb (Pb). Tous les fils sont plaqués avec de l'étain mat à 100%, et il n'y a pas de plomb à l'intérieur de l'emballage. Le cadre de plomb épais est fait de cuivre sans oxygène.

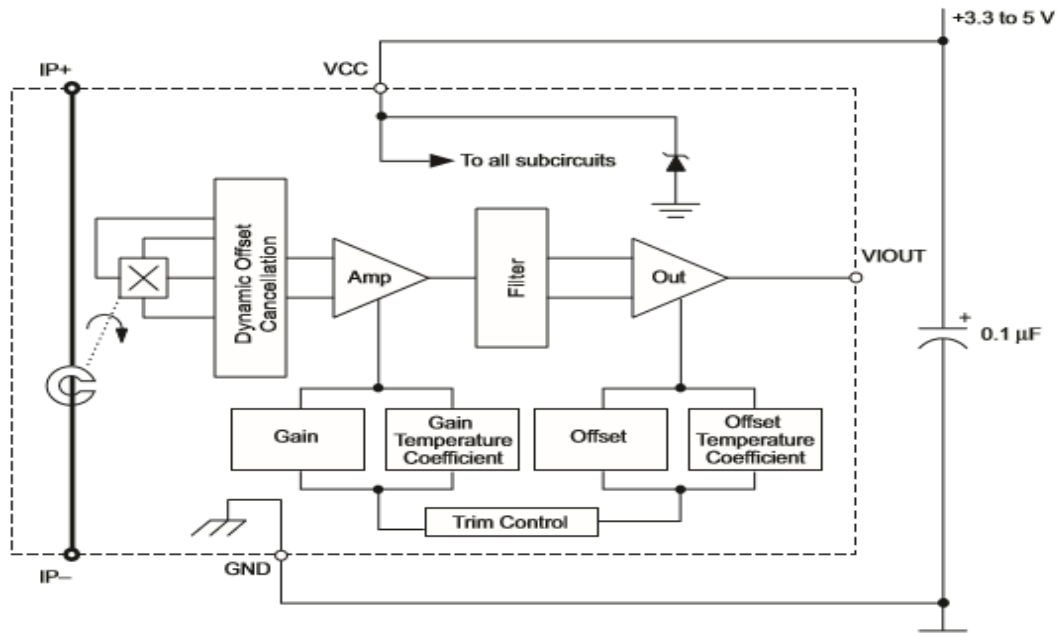


Figure 3-2 : Schéma d'ACS758

(Voir l'Annexe D)

### Schéma de montage de ce capteur

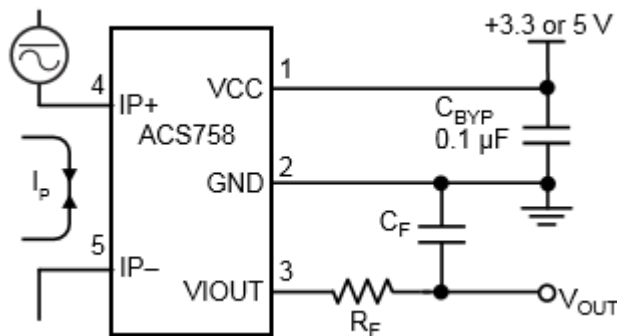


Figure 3-3 : Schéma de montage de capteur ACS758

A cause des problèmes de logistique dus à la situation sanitaire COVID-19 et de l'arrêt des lignes aériennes, nous n'avons pas pu récupérer le capteur voulu et nous avons changé vers un capteur disponible, le SCT-013-000

## 2- Description de SCT-013-000

La série SCT013 sont des capteurs à base de transformateurs de courant qui mesurent l'intensité d'un courant qui traverse un conducteur sans avoir besoin de couper ou de modifier le conducteur lui-même. Il fournit une image du courant proportionnelle à l'intensité du primaire. La mesure se fait par induction électromagnétique. On peut utiliser ces capteurs avec un microcontrôleur, comme Arduino, pour mesurer l'intensité ou la puissance consommée par une charge.



Les capteurs SCT013 ont un noyau fendu (comme une pince) qui permet à l'utilisateur d'utiliser pour envelopper le conducteur électrique sans avoir à le couper.

Le modèle que nous utilisons dans notre réalisation est le SCT013-000, dont le courant maximal est de 100A, la sortie de courant secondaire est de 50mA (100A: 50mA),

Il est important d'avoir une large gamme de mesures, il est important de garder à l'esprit qu'un modèle d'intensité plus élevée entraînera une précision moindre. Une intensité de 100A à 230V correspond à une charge de 23000 W, ce qui est suffisant pour la plupart des particuliers.

Un transformateur de courant cherche à générer une intensité dans le secondaire qui est proportionnelle à l'intensité qui traverse le primaire. Pour cela, on souhaite que le primaire soit formé d'un nombre réduit de spires.

Le rapport de transformation d'intensité dépend de la relation entre le nombre de tours:

$$\frac{I_s}{I_p} = \frac{V_p}{V_s} = \frac{N_p}{N_s}$$

### Montage de capteur

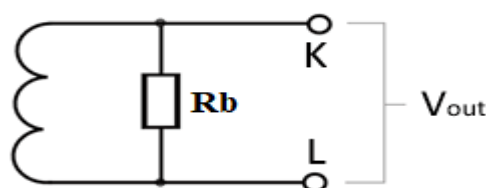
Pour comprendre la connexion du capteur SCT013, nous devons comprendre et résoudre trois problèmes:

- Sortie capteur en intensité
- Réglage de la plage de tension
- Tension positive et négative

#### a) Sortie Capteur en intensité

Les SCT013 sont des transformateurs de courant, c'est-à-dire que la mesure est obtenue sous la forme d'un signal d'intensité proportionnel au courant circulant dans le câble. Les microcontrôleurs, cependant, ne peuvent mesurer que des tensions.

Ce problème est facile à résoudre. Pour convertir la sortie en intensité en sortie de tension, il suffit d'inclure une résistance (résistance de charge).



À l'exception du modèle SCT013-100, tous les autres modèles SCT013 ont une résistance de charge interne de sorte que la sortie est un signal de tension de 1V pour un courant de SCT013-000. C'est pourquoi il ne sera pas préoccupant de s'inquiéter.

### b) Tensions positives et négatives

Un autre problème que nous devons résoudre est que nous mesurons le courant alternatif et que l'intensité induite dans le secondaire est alternative. Après avoir traversé résistance de charge, qu'elle soit interne ou externe, la sortie de tension est également alternée.

Cependant, comme nous le savons, les entrées analogiques de la majorité des microcontrôleurs, y compris ESP32, ne peuvent mesurer que des tensions positives.

Pour mesurer la tension à la sortie du transformateur, nous avons plusieurs méthodes, mais dans ce projet nous avons utilisé la méthode suivante :

Nous avons ajouté un décalage en DC en utilisant deux résistances et un condensateur qui fournissent un point médian entre Vcc et GND. Le schéma est présent sur la figure 3-11 page 47.

### c) Adaptation de la plage de tension

Le dernier problème est la nécessité d'adapter la plage de tensions à la sortie du capteur. Arduino ne peut effectuer des mesures qu'entre 0 et Vcc. De plus, plus la plage est petite, plus la perte de précision est grande, il faut donc s'adapter à cette plage [10].



Figure 3-4 : Capteur de courant SCT 013

Le capteur SCT013 dispose d'un connecteur Jack 3.5,

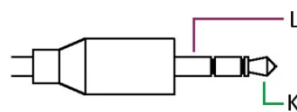


Figure 3-5 : Fiche jack de capteur de courant SCT 013

Dans ce montage, nous avons utilisé juste les deux bornes K et L.

(Voir l'Annexe E)

## IV- Capteur de tension

C'est le deuxième composant dont nous avons besoin dans ce projet, la tension de SONELGAZ est instable, et parfois il dépasse les limites déclarées ( $230V \pm 10\%$ ), Nous devons donc le mesurer en temps réel, et pour ce projet nous avons choisi l'ACPL-C87AT :

### 1- Description

Les capteurs d'isolement Broadcom ACPL-C87AT utilisent une technologie de couplage optique supérieure, avec un convertisseur analogique-numérique sigma-delta, des amplificateurs stabilisés par hacheur et une topologie de circuit entièrement différentielle pour fournir une réjection de bruit en mode d'isolement inégalée, un décalage faible, un niveau élevé gagné en précision et en stabilité.

ACPL-C87AT (tolérance de gain de  $\pm 1\%$ ) sont conçus pour la détection de tension continue de haute précision dans les entraînements de moteurs électroniques, les convertisseurs DC / DC et AC / DC et le système de surveillance de batterie. L'ACPL-C87AT présente une impédance d'entrée élevée et fonctionne avec une plage complète de tension d'entrée analogique jusqu'à 2,46 V. La fonction d'arrêt permet d'économiser l'énergie et peut être contrôlée à partir d'une source externe, comme un microprocesseur.

L'immunité transitoire élevée en mode commun ( $15 \text{ kV} / \mu\text{s}$ ) de l'ACPL-C87AT maintient la précision et la stabilité nécessaires pour surveiller avec précision la tension du rail CC dans des environnements de contrôle de moteur à bruit élevé. Cette solution d'isolation galvanique est livrée dans un boîtier compact SO-8 étiré en surface (SSO-8) qui répond aux normes de sécurité réglementaires mondiales [5].

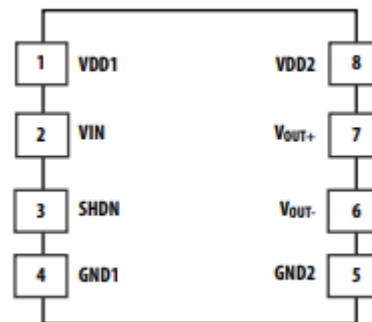
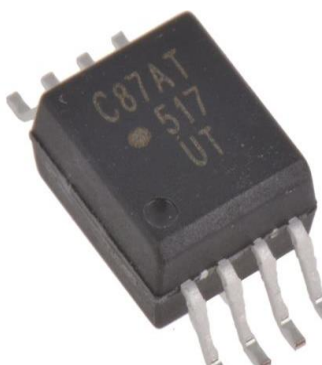


Figure 3-6 : ACPL-C87AT

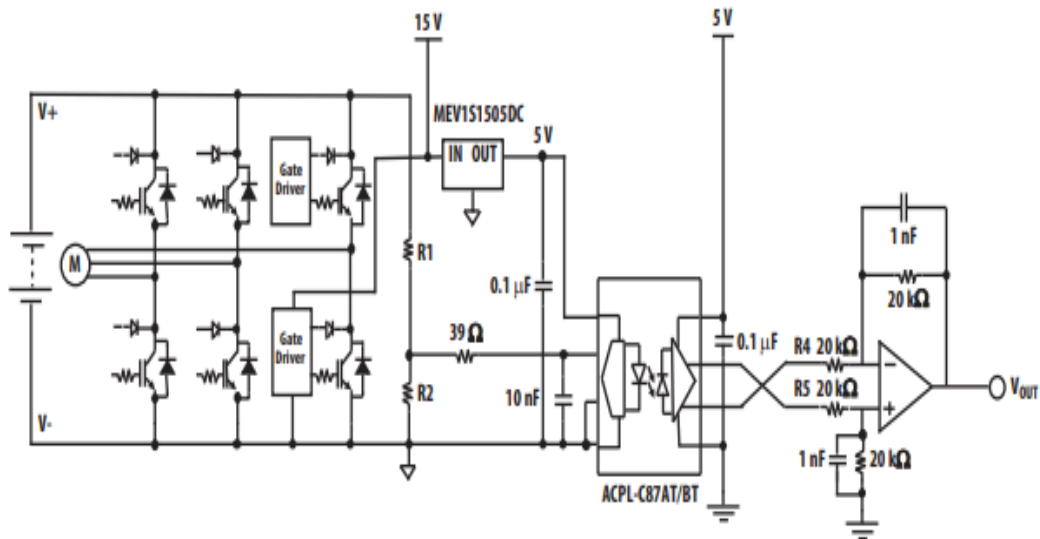


Figure 3-7 : Circuit de capteur de tension typique

(Voir l'Annexe F)

## V- Wemos Lolin32

Pour réaliser notre projet nous avons besoin d'un microcontrôleur pour traiter les signaux analogiques (tension et courant) nécessaire pour le calcul de la puissance et l'énergie consommée, pour ça nous avons choisi le Wemos Lolin32.

En effet, l'Arduino est un uC 8 bits dépassé. Il fonctionne en 5V et a une mémoire (Flash et RAM) limitée. L'ESP32 est un uC 32 bits puissant. La version Wemos Lolin32 fonctionne en 3.3V. L'ESP32 dispose du Wifi, BT, capteur magnétique, Touch et autres interfaces que l'Arduino n'a pas.

### 1-Description

La Wemos Lolin32 est une carte de développement construite autour du microcontrôleur Espressif ESP-WROOM-32. Il dispose de 4 Mo de mémoire et offre une connectivité Wifi et Bluetooth. Le Wemos LoLin32 a la particularité de proposer un connecteur pour batterie LiPo. Le connecteur est JST XH2-2,54 mm. Il a un détrompeur qui empêche l'inversion de polarité. La charge de la batterie et la programmation se feront à l'aide du connecteur micro-USB habituel.

Contrairement aux premières cartes de développement ESP32, cette carte développée par Wemos.cc est plus étroite. Il laisse une rangée de connecteurs de chaque côté de la carte libre sur la maquette [11].



Figure 3-8 : ESP32 Wemos Lolin32

## 2- Les caractéristiques

- Module ESP32: ESP-WROOM-32 d'Espressif.
- Microprocesseur double cœur 240 MHz équipé de
- Mémoire flash SPI de 4 Mo. Prend en charge jusqu'à 16 Mo de mémoire flash
- Connectivité
- WiFi 802.11 b / g / n.
- Sécurité WEP, WPA / WPA2 PSK / Enterprise.
- Puce cryptographique intégrée prenant en charge les algorithmes AES / SHA2 / Elliptical Curve Cryptography / RSA-4096
- Puissance maximale pour le transfert de données: 19,5 dBm @ 11b, 16,5 dBm @ 11g, 15,5 dBm @ 11n
- Sensibilité max. réception: -97 dBm
- Bluetooth 4.0 LE
- 32 entrées / sorties
- 26x E / S numériques (3,3 V). Toutes les sorties peuvent être PWM
- 18x entrées analogiques
- 3x UART
- 3x SPI
- 2x I2S
- 2x DAC

- 2x I2C
- Consommation en mode veille: 5  $\mu$ A
- Capteurs intégrés
- Effet Hall
- 10x entrées pour interface tactile capacitive

**UART :** c'est le composant utilisé pour faire la liaison entre l'ordinateur et le port série. L'ordinateur envoie les données en parallèle (autant de fils que de bits de données). Il faut donc transformer ces données pour les faire passer à travers une liaison série qui utilise un seul fil pour faire passer tous les bits de données.

## VI- Ecran OLED

Nous avons besoin d'un écran OLED pour afficher les résultats (la puissance, Energie, Courant et la tension).



Figure 3-9 : Ecran OLED

## VII- Réalisation du montage

### 1- Le montage d'ACPL-C87AT

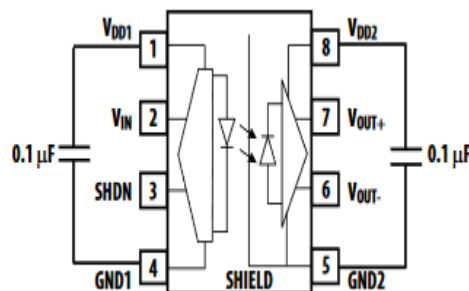


Figure 3-10 : Montage ACPL-C87AT

Voir l'Annexe D

### 2- Le montage de SCT 013

Les composants nécessaires pour se relier capteur SCT 013 avec ESP32 sont :

- 2x Résistances de 10k $\Omega$

- 1x Capacité de  $10\ \mu F$

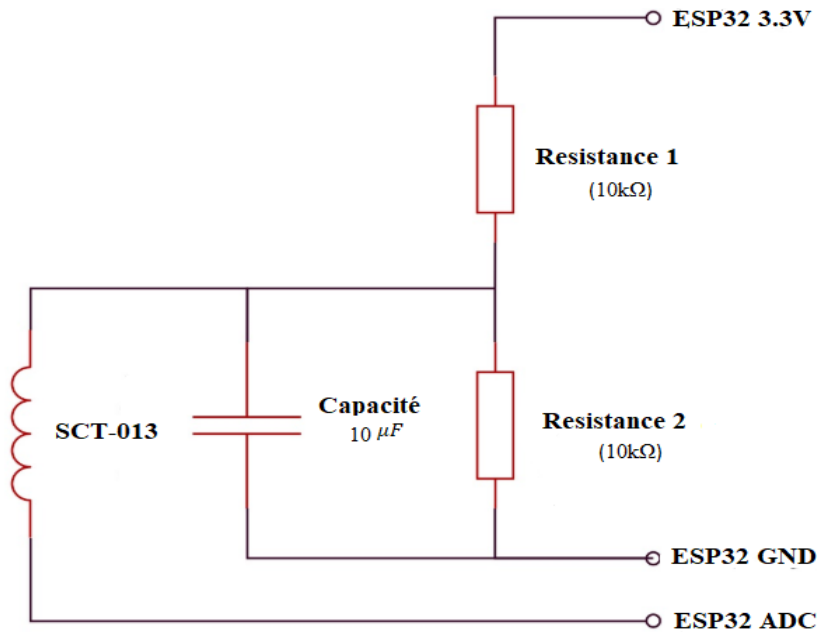


Figure 3-11 : Schéma de montage de SCT 013

Avant de le connecter avec ESP32 sur une entrée ADC, on doit vérifier ne pas dépasser 3.3V (problème évoqué durant description de ce capteur)

Après avoir complété le montage, nous avons testé le capteur en mesurant des courants à l'aide d'un circuit électrique comportant des lampes sous tension de 230 V de différentes puissances et les résultats sont comme suit :

Tableau 3-1 : Les Résultats de tests de la sortie de SCT 013

Puissance W	0	5	7	13	18	20	36	50	75	100
Multimètre 1 (V)	1.35	1.84	2.35	1.87	2.74	2.81	3.57	3.20	1.78	1.98
Multimètre 2 (V)	1.38	1.88	2.38	1.88	2.74	2.84	3.59	3.22	1.79	2.00

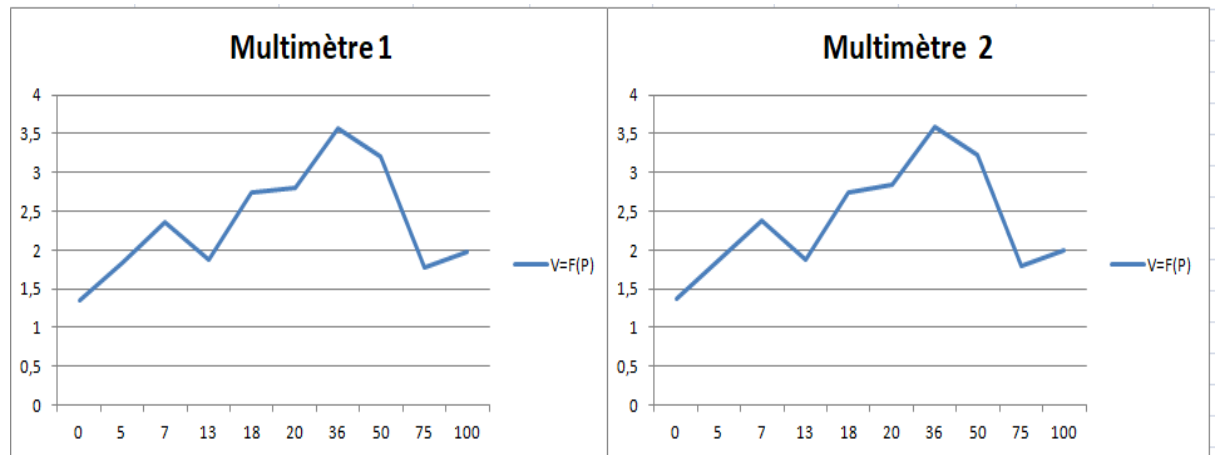


Figure 3-12 : Les Résultats de tests de la sortie de SCT 013

**Remarque :** Avec les courbes Excel d'étalonnage qui montrent que la mesure avec le multimètre n'est pas bonne et que nous n'arrivons pas à continuer sur cette partie à cause du risque de destruction de l'entrée ADC de l'ESP32.

### 3- Montage d'écran OLED

La carte de l'écran OLED à quatre pins (VCC, GND, SCL, SDA) avec liaison I2C :

PINS OLED	PINS ESP32
VCC	3.3V
GND	GND
SCL	PIN 23
SDA	PIN 19

#### Présentation du concept bus I2C

I2C est un bus série permettant de transmettre des informations de façon asynchrone entre divers circuits connectés sur le bus. Le protocole de la liaison est du type MAITRE/ESCLAVE. Chaque circuit est reconnu par son adresse et peut être soit transmetteur soit receveur de l'information. Ces circuits peuvent être : Un ordinateur, un microcontrôleur, un microprocesseur, une mémoire, un périphérique (clavier, écran,...) etc.

Dans le protocole du bus I2C le circuit maître est celui qui demande un transfert d'information sur le bus et qui génère le signal d'horloge qui permet le transfert. Ainsi un circuit adressé est considéré comme un esclave.

#### Caractéristiques physiques de bus I2C

Deux fils SDA (Serial DAta) et SCL (Serial CLock) véhiculent les informations entre les différents circuits.



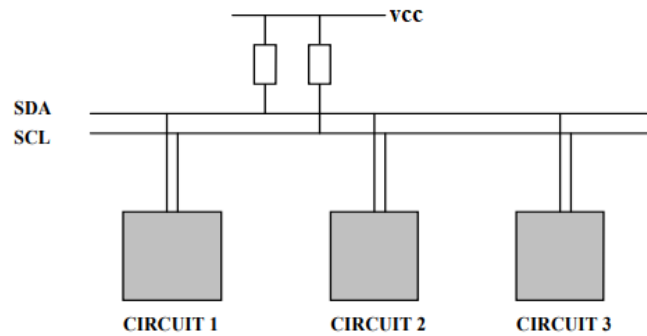


Figure 3-13 : Caractéristiques physiques de bus I2C

SDA et SCL sont des lignes bidirectionnelles, connectées à plus VCC par l'intermédiaire de deux résistances de tirage. Quand le bus est libre, c'est à dire quand il n'y a pas de transfert de données les deux lignes sont à l'état haut.

### Conditions de START et de STOP

Un message débute par une condition de START et se termine par une condition de STOP

- Condition de START: Passage de l'état haut à l'état bas de SDA pendant que SCL est à l'état haut.
- Condition de STOP: Passage de l'état bas à l'état haut de SDA pendant que SCL est à l'état haut.

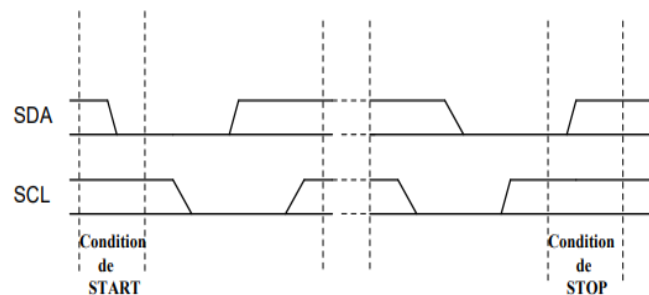


Figure 3-14 : Conditions de START et de STOP

## VIII- Les calculs nécessaires

### 1- Calcul de constant de courant

$$\text{Constante de courant} = R_t \div R_b$$

$R_b$  : est la valeur de la résistance de charge,

$R_t$  : est le rapport du transformateur de courant,

$$R_t = 100A \div 50 \text{ mA} = 2000$$

$$\text{Constant de courant} = R_t \div R_b = 2000 \div 22 \Omega = 90.9$$

## 2- Calcul des puissances

La puissance active correspond à la puissance moyenne fournie au cours d'une période donnée. Son expression est :

$$P = \frac{1}{T} \int_0^T v(t) \cdot i(t) \cdot dt$$

$$v(t) = V \cdot \sqrt{2} \cdot \cos(\omega t)$$

Et

$$i(t) = I \cdot \sqrt{2} \cdot \cos(\omega t - \varphi)$$

Alors on montre en intégrant que :

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

Où  $\varphi$  est le déphasage entre la tension et le courant et  $\cos(\varphi)$  est appelé facteur de puissance.

Concernant la puissance réactive il est calculé par l'équation suivant :

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

Et enfin pour calculer la puissance apparente nous avons l'expression :

$$S = V \cdot I$$

## 3- Calcul de facteur de puissance

Le calcul de facteur de puissance nécessite de connaître le déphasage entre la tension et le courant

Nous pouvons écrire le déphasage entre la tension et le courant comme suite :

$$\cos(\varphi) = \frac{P}{S}$$

#### 4- Calcule de l'énergie

L'énergie électrique par définition est la puissance obtenue par un appareil multiplié par le temps d'utilisation de cet appareil, donc nous pouvons écrire :

$$E = P \cdot t \quad (\text{kWh})$$

L'unité reconnue par le système internationale de l'énergie électrique est le Joule, mais généralement pour des raisons pratiques de facturation il est préféré d'utiliser le kWh comme unité d'énergie. Concernant le programme utilisé dans notre microcontrôleur nous avons choisi de calculer l'énergie à chaque seconde comme suit :

$$E = \frac{P}{3600.1000} (\text{kWh})$$

### IX- Conclusion

Dans ce chapitre nous avons présenté les principaux composants nécessaires pour réaliser la première fonction du compteur intelligent et qui concerne la mesure des données de consommations.

Mais à ce sujet, des problèmes sont survenus qui nous ont empêchés de réaliser le montage (capteur de courant et capteur de tension).

- Capteur de courant : A cause de risque de destruction de l'entrée ADC.
- Capteur de tension : Indisponibilité de composant à cause de confinement.

Après avoir terminé le montage, il faut programmer l'ESP32 pour pouvoir profiter de tous les composants, on va présenter dans le chapitre suivant comment programmer l'ESP32 pour calculer l'énergie, le prix et transmettre les informations.

## Chapitre 4 : Partie Logicielle

### I- Introduction

Au cours de ce chapitre nous allons étudier de la partie logicielle et de programmation ainsi que la méthode de transmission des données de consommation et des tarifs.

### II- Description du programme

On peut diviser notre programme en deux parties, coté opérateur de réseaux et coté client.

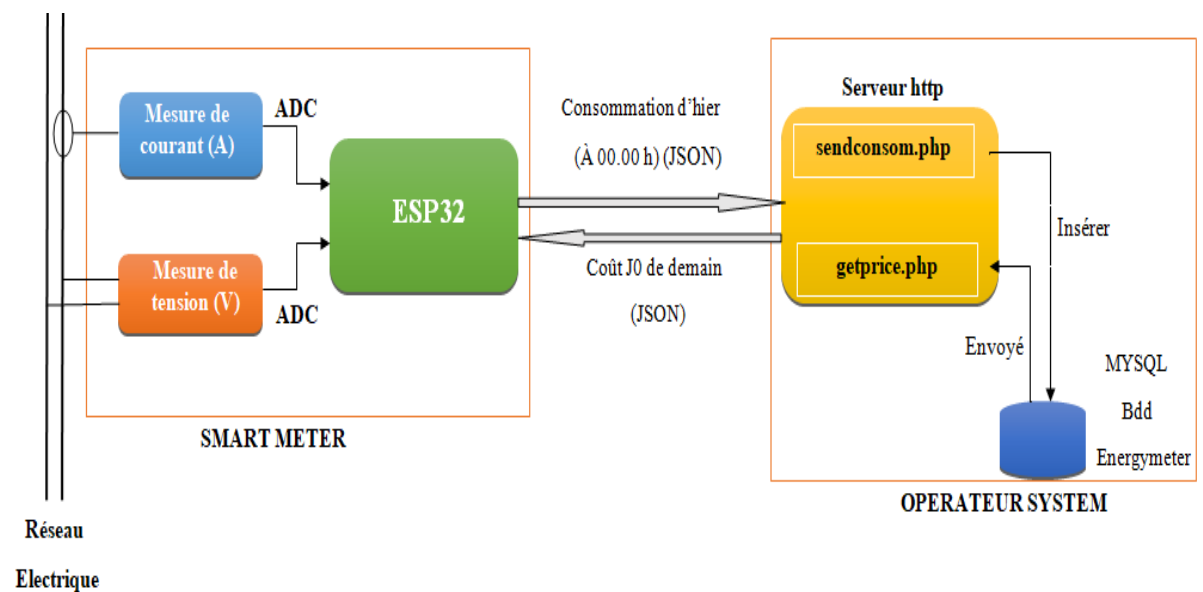


Figure 4-1 : Schéma synoptique

#### 1- Coté Opérateur de réseaux

L'opérateur de réseaux électriques a deux tâches principales dans notre programme.

##### a) Délivrer les nouveaux compteurs

Lors d'une demande d'un nouveau compteur, l'employée d'opérateur de système doit créer un compte pour le nouveau client, avec les informations d'identification (nom, Adresse, Adresse mail, numéro de téléphone) et le numéro de compteur « Nclient ». Ce dernier est l'identifiant utilisé dans le programme de compteur pour stocker les données de consommation de tel client. Donc après cette inscription, l'agent responsable de programmer les compteurs doit ajouter le « Nclient » et le mot de passe.

##### b) Envoyer les prix de consommation de chaque jour

Nous avons parlé dans le (**chapitres I**) de la méthode proposer pour la facturation de consommation. La division de la journée en 96 tranches (24



b) Affichage de consommation en temps réel

Les données calculées dans l'ESP32 vont être affichées en temps réel dans l'OLED et sur une page web.

c) Envoyer les données de consommation et de coût à minuit, puis les stocker dans la base de données

À la fin de journée les données de consommation vont être envoyées vers la base de données via un programme PHP, puis on réinitialise les variables qui stockent le coût et la consommation.

d) Afficher les données précédemment stockées dans la base de données

En se connectant sur le site web de l'opérateur, en utilisant l'identifiant de client et le mot de passe, le client peut voir ces données de consommation de ces derniers 3 mois, et aussi le prix de la journée suivante.

### III- IDE

#### 1- Introduction IDE

IDE est une application multiplateforme (pour Windows, MacOS, Linux). Il est utilisé pour écrire et télécharger des programmes sur des cartes compatibles Arduino, mais aussi, avec l'aide de noyaux tiers, vers d'autres cartes de développement comme pour notre cas l'ESP32.

L'IDE Arduino prend en charge les langages C et C++ en utilisant des règles spéciales de structuration de code.

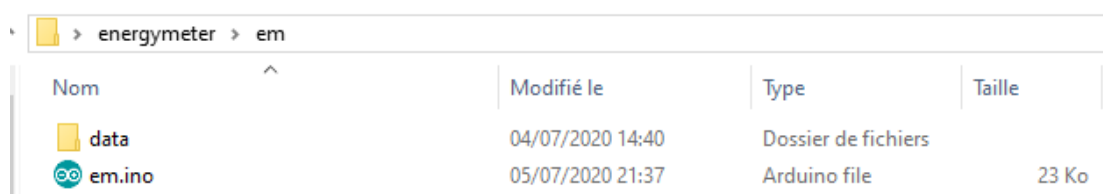
#### 2- Préparer l'environnement de l'IDE






L'Annexe G présente les différentes étapes pour préparer l'environnement de l'IDE.

#### 3- Programmation de la mémoire FLASH (data)

a) Nous créons le répertoire data et nous y mettons les fichiers que l'on souhaite accéder depuis l'ESP32 via la SPIFFS.

Par exemple, les fichiers de configuration (format Json) et les fichiers html que l'ESP32 va nous renvoyer quand on interrogera son serveur à l'aide d'un navigateur web.



Nom	Modifié le	Type	Taille
 configwifi.html	24/06/2020 16:25	Microsoft Edge H...	3 Ko
 data.json	24/06/2020 16:33	Fichier JSON	1 Ko
 index.html	24/06/2020 18:34	Microsoft Edge H...	6 Ko
 param.json	04/07/2020 15:37	Fichier JSON	1 Ko
 tst.html	16/06/2020 00:10	Microsoft Edge H...	5 Ko

Ensuite, pour télécharger les fichiers, dans l'IDE Arduino, il nous suffit d'aller dans Outils > ESP32 Sketch Data Upload.

#### 4- Programmation de la mémoire FLASH (programme)

Appuyer sur le bouton Télécharger dans l'IDE Arduino. Attendre quelques secondes pendant que le code se compile et se télécharge sur notre tableau.



## IV-Réalisation du programme

### 1- Routine de connexion Wifi

Dans notre projet on a besoin de connecter Esp32 avec internet pour l'envoi et la réception des données vers et depuis le serveur, on inclue donc la librairie WiFi.h :

```
#include <WiFi.h>
```

On a pu déclarer le SSID et le mot de passe dans le programme pour connecter avec le wifi directement, mais ce n'est pas pratique. Donc on utilise une autre méthode plus pratique, que notre encadrant nous a fourni. C'est de lancer l'ESP32 en un point d'accès Wifi, on connectant avec l'ESP32 on peut entrer Wifi SSID et mot de passe de notre routeur wifi local. L'ESP32 sauvegardera ces informations dans la mémoire Flash via SPIFFS.

Au prochain boot de l'ESP32, il lira ces informations et se connectera en mode client à notre routeur wifi avec le SSID et PW lus.

Pour la partie WIFI\_AP,

Tout d'abord on déclare le SSID et mot de passe de ce point d'accès

```
const char *ssid_AP = "st_";
const char *password_AP = "stpassword";
```

La fonction APWifiParam() sert à créer les données du point d'accès pour la routine la fonction ConnectWifi() en cas de non disponibilité du SSID/PW du routeur (configuration en mémoire Flash SPIFFS vide).

```
//-----
void APWifiParam() {
  Serial.println("Empty wifi Param from SPIFFS, use AP :");
  Nconfigs = 1;
  ssid[0] = ssid_AP;
  password[0] = password_AP;
  SOFTAP = true;
  Serial.print("AP Wifi Server: ");
  Serial.println(ssid[0]);
  Serial.print("PW: ");
  Serial.println(password[0]);
}
```

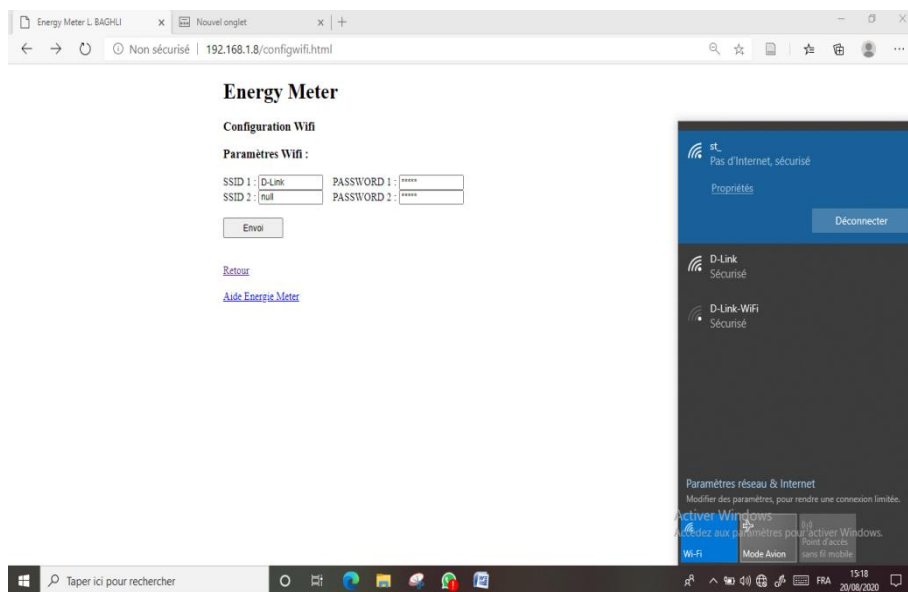


Figure 4-2 : Se connecter ESP32

Après cette connexion on entre l'adresse IP affichée sur l'écran OLED, dans le navigateur :



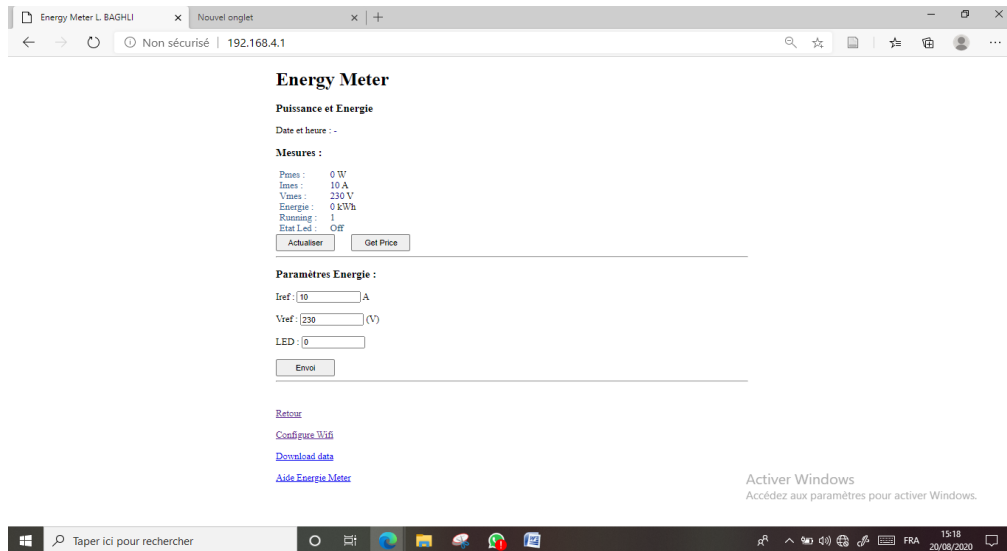


Figure 4-3 : Page Web Energy Meter avant la connexion

Puis on clique sur Configure Wifi, cela nous permet de entrer les paramètres de notre wifi :

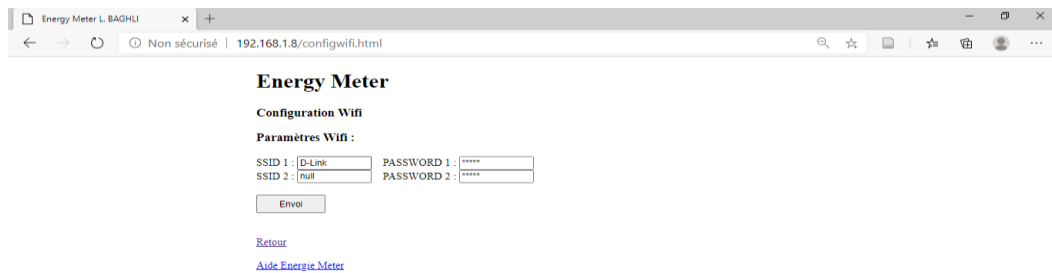


Figure 4-4 : Configuration Wifi

Quand on entre les paramètres de wifi, ces derniers seront enregistrés dans le fichier Param.json dans le dossier data:



La fonction WriteParam() permet de lire le fichier Param.json et écrire nos paramètres :

```

//-----
void WriteParam() {
  Serial.println("-- Writing Param to SPIFFS on : ");
  StaticJsonDocument<256> doc;
  doc["ssid1"] = ssid[0];
  doc["pw1"] = password[0];
  doc["ssid2"] = ssid[1];
  doc["pw2"] = password[1];

  serializeJsonPretty(doc, Serial);
  File configFile = SPIFFS.open("/param.json", "w");
  if (!configFile) {
    Serial.println("Failed to open parameters file");
    return;
  }
  if (serializeJson(doc, configFile) == 0) {
    Serial.println(F("Failed to write to SPIFFS file"));
  }
  configFile.close();
}

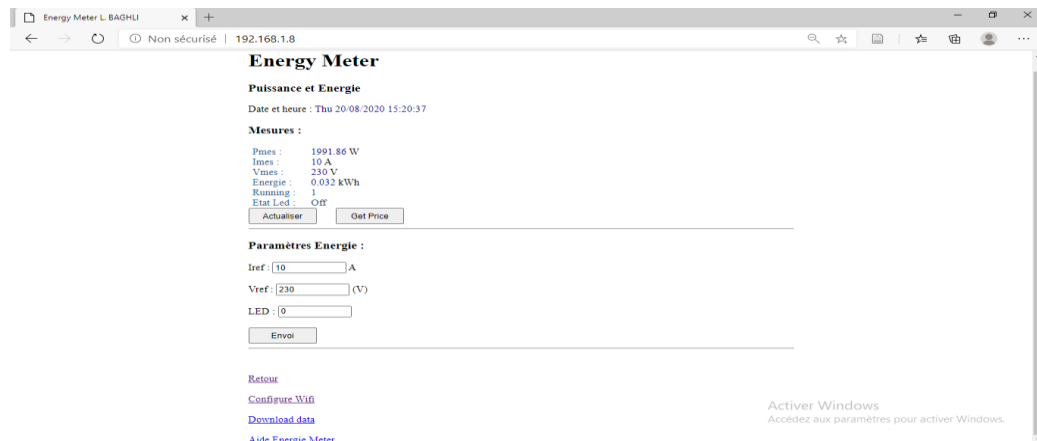
```

On clique sur ‘Envoi’, puis on clique sur reset de l’ESP32.

Quand l’ESP32 est rebooté il va se connecter sur le routeur en tant que client et affiche l’adresse IP qu’il lui attribue.

On utilisera cette adresse pour accéder au serveur ESP32 (en effet il adopte aussi un comportement serveur comme on le verra).

Cette partie a été programmée par notre encadrant car elle dépasse le cadre du PFE Commandes Electriques.



**Figure 4-5 : Page Web Energy Meter après la connexion**

La fonction ReadParam() sert à lire les données de Param.json puis les stocker dans les variables ssid[0],password[0] et ssid[1],password[1], ces variables seront utilisées pour se connecter au wifi.

```
//-----
void ReadParam() {
  Serial.println("-- Reading Wifi and Town Param from SPIFFS");
  StaticJsonDocument<512> doc;
  File configFile = SPIFFS.open("/param.json", "r");
  if (!configFile) {
    Serial.println("Failed to open parameters file, using AP");
    APWifiParam();
    return;
  }
  DeserializationError err = deserializeJson(doc, configFile);
  configFile.close();
  if (err) {
    Serial.print(F("deserializeJson() failed with code, using AP "));
    Serial.println(err.c_str());
    APWifiParam();
  }
  serializeJsonPretty(doc, Serial);

  Nconfigs = 2;
  Serial.print("Wifi config to read : ");
  ssid[0] = doc["ssid1"].as<String>();
  password[0] = doc["pw1"].as<String>();
  ssid[1] = doc["ssid2"].as<String>();
  password[1] = doc["pw2"].as<String>();

  if (ssid[0].isEmpty()) {
    Serial.println("Wifi parameters file empty, using AP");
    APWifiParam();
  }
};
}
```

Après le stockage des paramètres dans les variables ssid et password, on est prêt pour se connecter au Wifi à l'aide de la fonction ConnectWifi().

Elle doit être appelée dans la fonction setup, après le démarrage de l'ESP32.

```
// fonctions definition
void ConnectWifi() {
  int attempt;
  Serial.println("Connecting to Wifi");
  if (SOFTAP) {
    WiFi.softAP(ssid[0].c_str(), password[0].c_str());
    // Connexion Wifi établie / WiFi connexion is OK
    Serial.println("");
    Serial.print("Wifi Server: ");
    Serial.println(ssid[0]);
    Serial.print("IP address: ");
    Serial.println(WiFi.softAPIP());
    timeinfo.tm_year = 0; // Set current time to illegal
  } else {
    // IPAddress subnet(255, 255, 255, 0); // set subnet mask to match your network
    // IPAddress dns1(8, 8, 8, 8);
    // IPAddress dns2(8, 8, 4, 4);
    //Serial << ssid << password <<" "<< endl;//todo debug
    WiFi.mode(WIFI_STA);
    WiFi.persistent(false);
    WiFi.disconnect(true);
    while (WiFi.status() != WL_CONNECTED) {
      Serial.println();
      Serial.print("Connecting to ");
      Serial.println(ssid[Wificonfig]);
      WiFi.begin(ssid[Wificonfig].c_str(), password[Wificonfig].c_str());
    // WiFi.begin();
    // WiFi.config(ip, gateway, subnet, dns1, dns2); //todo
      Serial.print("Connecting ");
      attempt = 0;
      while ((attempt < 20) && (WiFi.status() != WL_CONNECTED)) {
        delay(500);

```

```

        Serial.print(".");
        attempt++;
    }
    if (WiFi.status() != WL_CONNECTED) {
        Wificonfig++;
        if (Wificonfig >= Nconfigs) Wificonfig = 0;
    }
}
// Connexion WiFi établie / WiFi connexion is OK
Serial.println("");
Serial.print("Connected to ");
Serial.println(ssid[Wificonfig]);
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
//init and get the time
configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
GetNTPDTH();
}
}

//-----
void setup() {
    Serial.begin(SERIAL_SPEED);
    pinMode(LED_BUILTIN, OUTPUT);
    emon1.current(1, 96.9);
    digitalWrite(LED_BUILTIN, 1); // OFF
    initVar();
    Serial.println(F("Starting Energy Meter Program"));
    // Initialize SPIFFS
    if (!SPIFFS.begin()) {
        Serial.println("An Error has occurred while mounting SPIFFS");
        return;
    }
    Serial.print(F("Initializing OLED and Wifi\n"));
    u8g2.begin();
    u8g2.setFont(u8g2_font_helvB12_tr); // set the target font to calculate the pixel width
    u8g2.setFontMode(1); // enable transparent mode, which is faster
    u8g2.clearBuffer(); // clear the internal memory
    u8g2.setCursor(0, 16);
    u8g2.print("Reading Wifi");
    u8g2.setCursor(0, 32);
    u8g2.print("parameters");
    u8g2.sendBuffer();
    ReadParam(); // Read Wifi and Town config from Flash SPIFFS
    u8g2.setCursor(0, 48);
    u8g2.print("Connecting...");
    u8g2.sendBuffer();
    ConnectWiFi();
}

```

## 2- Routines de l'afficheur OLED

On inclut la bibliothèque correspondant à notre afficheur :

```
#include <U8g2lib.h>
```

Puis on configure la librairie en lui indiquant les pins de l'OLED utilisées sur l'ESP32.

```
// LED_BUILTIN redefinir au lieu d' IO 5 c'est IO 22
#define LED_BUILTIN 22
#define OLED_SDA 23
#define OLED_SCL 19

U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, /* reset=*/ U8X8_PIN_NONE, /* clock=*/ OLED_SCL, /* data=*/
OLED_SDA); // (rotation, [reset [, clock, data]])
```

Pour initialiser la librairie u8g2 qui gère l’OLED, on appelle sa fonction begin dans la fonction setup () du programme :

```
u8g2.begin();
```

### 3- Routines du serveur web local (ESP32)

On est besoin d’afficher la consommation et le prix en temps réel d’une façon facile à utiliser, pour ce but on utilise un serveur web dans ESP32. Ici l’ESP32 va jouer le rôle d’un serveur web. L’utilisateur va juste écrire l’adresse de serveur dans le navigateur et obtient tous les données qu’il est besoin de voir, tel que la consommation, le prix.

On utilise la bibliothèque « EspAsyncWebServer.h » pour créer le serveur web:

```
#include "ESPAsyncWebServer.h"
```

Créer un objet « AsyncWebServer » dans le port 80.

```
AsyncWebServer server(80);
```

Puis on appelle le serveur web dans la fonction void setup ():

```
server.on("/param.json", HTTP_GET, [](AsyncWebServerRequest *request) {
    request->send(200, "application/json", sendParam()); // envoie les paramètres
});

server.on("/data.json", HTTP_GET, [](AsyncWebServerRequest *request) {
    request->send(200, "application/json", sendData()); // envoie les données à jour
});

server.on("/ref.json", HTTP_GET, [](AsyncWebServerRequest *request) {
    request->send(200, "application/json", sendRef()); // envoie les données à jour
});

// route de download text/csv
server.on("/data.csv", HTTP_GET, [](AsyncWebServerRequest *request) {
    request->send(200, "text/csv", getFile()); // envoie le fichier
});

server.on("/bt_getprice", HTTP_GET, [](AsyncWebServerRequest *request) {
    Flags.GetPrice = 1;
    timerAlarmDisable(timer);
    request->send(SPIFFS, "/index.html");
    timerAlarmEnable(timer);
});
```

```

server.on("/configwifi.html", HTTP_GET, [](AsyncWebServerRequest *request) {
    bool wifiUpdate = false;
    int paramsNr = request->params();
    Serial.println("\n-- configwifi.html requested --");
    for (int i = 0; i < paramsNr; i++) {
        AsyncWebParameter *p = request->getParam(i);
        Serial.print("Param name: ");
        Serial.println(p->name());
        Serial.print("Param value: ");
        Serial.println(p->value());
        if (p->name() == "ssid1") {
            ssid[0] = p->value();
            wifiUpdate = true;
        }
        if (p->name() == "pw1") {
            if (p->value().compareTo("*****") != 0)
                password[0] = p->value();
            wifiUpdate = true;
        }
        if (p->name() == "ssid2") {
            ssid[1] = p->value();
            wifiUpdate = true;
        }
        if (p->name() == "pw2") {
            if (p->value().compareTo("*****") != 0)
                password[1] = p->value();
            wifiUpdate = true;
        }
    }
    if (wifiUpdate) WriteParam(); // Write Config Parameters to Flash
    timerAlarmDisable(timer);

    request->send(SPIFFS, "/configwifi.html");
    timerAlarmEnable(timer);
});

```

#### 4- Programme de capteur de courant

On appelle la bibliothèque spéciale pour notre capteur.

```
#include "EmonLib.h"
```

Dans void setup () on entre le numéro d'entrée analogique (pin1) et le calibre calculé dans le chapitre précédent: 96.9

```
emon1.current(1, 96.9);
```

Affecter les mesures de captures dans un variable Imes :

```

//-----
void ReadVI() {
    thetas += 0.31415926535897932384626433832795; // 2pi *fs * dt (1ms)
    if (thetas >= 6.28318530718) thetas -= 6.283185307179586476925286766559; // modulo 2pi
    Vmes = Vref * r2 * sin(thetas);
    Imes = emon1.calcIrms(1480);
    Psum += Vmes * Imes;
}

```

## 5- Programme de temps

On appelle la bibliothèque spéciale pour obtenir de temps local utilisant la connexion internet

```
#include <time.h>
```

gmtOffset\_sec = 3600 veut dire le décalage avec le Gmt, ici on Algérie on est gmt+1 h, et en seconds 3600 sec

```
const char *ntpServer = "pool.ntp.org";
const long gmtOffset_sec = 3600;
const int daylightOffset_sec = 0; // no DST
struct tm timeinfo;

//-----
void GetNTPDTH() {
//todo if (WiFi.status() != WL_CONNECTED) return;
char S[50];
if (!getLocalTime(&timeinfo)) // Read from NTP server
{
Serial.println("Failed to obtain time!"); // Error
timeinfo.tm_year = 0; // Set current time to illegal
} else {
strftime(S, 50, "%a %d/%m/%Y %H:%M:%S", &timeinfo);
Serial.println(S);
}
}
```

## 6- Calcul des données de consommation

L'ISR de l'ESP32 a un compteur de chaque 1 milliseconde. Utilisant des compteurs logiciels (Pcount =20 ; qui calcul 20 ms) nous donnent la lecture de consommation et calculer la puissance chaque 20 ms. Le compteur logiciel Nj = 96, sélectionne quelle tranche de jour on utilise pour enregistrer les données de consommation dans le tableau EmesTable et CouTable. Dans l'EmesTable on stock les énergies calculées en Wattheure, mais puisque les prix de consommation sont en DZD/kWh on doit multiplier l'Energie en  $10^{-3}$ .

A minuit 00 heure et 00 minute, on stocke les mesures d'énergies et de cout, puis on réinitialise les autres variable à 0. Et on va mettre le Flag d'envoi des données à 1, qui permet d'appeler la fonction sendconsom().

```

#define r2 1.414213562373095
#define Njprix 96 // 24h *4
#define Nclient 130001
#define Password 1234
double j0prix[Njprix], j1prix[Njprix];

double Vref, Iref; //dbug
double Vmes, Imes, thetas, phi;
double Psum, Pmes, Emes, Emes15, Emestable[Njprix], CouTable[Njprix], CouTotal;
double Consumption, Cost, TableYesterday[Njprix];
int PNcount = 20, Pcount = 0; // 20 *1ms = 20 ms
int SecNcount = 50, Seccount=0; // 50 *20ms = 1 s
int DispNcount = 100, Dispcount = 0; // 100 *20ms = 2 s
...

//-----
void ReadVI() {
    thetas += 0.31415926535897932384626433832795; // 2pi *fs * dt (1ms)
    if (thetas >= 6.28318530718) thetas -= 6.283185307179586476925286766559; // modulo 2pi
    Vmes = Vref * r2 * sin(thetas);
    Imes = Iref * r2 * sin(thetas - phi);
    Psum += Vmes * Imes;
}

//-----
void IRAM_ATTR onTimer() {
    ReadVI();
    if (++Pcount >= PNcount) {
        // ici on est toutes les 20 ms
        Pcount = 0;
        Pmes = Psum / PNcount;
        double Emes20ms = Pmes * 5.555555555555555556e-9; // kWh
        Emes += Emes20ms;
        Emes15 += Pmes;
        Psum = 0;
        betatLed = !betatLed;
        digitalWrite(LED_BUILTIN, betatLed);

        if (++Seccount >= SecNcount) {
            Seccount = 0; // toutes les secondes
            if (++timeinfo.tm_sec >= 60) // Yes, update number of seconds
            {
                timeinfo.tm_sec = 0; // Wrap after 60 seconds
                if (++timeinfo.tm_min >= 60) { // Wrap after 60 minutes
                    timeinfo.tm_min = 0;
                    if (++timeinfo.tm_hour >= 24) { // Wrap after 24 hours
                        timeinfo.tm_hour = 0;
                        Flags.GetNTP = 1;
                    }
                }
            }
        }
        if ((timeinfo.tm_min %15) == 0) { // un 1/4 heure s'est écoulée
            int Nj = timeinfo.tm_hour*4 + (timeinfo.tm_min /15);
            if (Nj>96) {
                Nj = 0;
            } // just in case
        }
    }
}

```



```

Emestable[Nj] = Emes15 * 0.25; // 0.25*Emes15= Emestable[Nj] Wh.
CouTable[Nj] = Emestable[Nj]*1e-3*j0prix[Nj]; // créer un tableau des cout de consommation.
CouTotal=CouTotal+CouTable[Nj]; // cout total de la consommation.
Emes15 = 0;
    if(timeinfo.tm_hour==0 & timeinfo.tm_min==0){ //à minuit
        // add double Emj=Emes; to stock the consumption of the day here
        // Emes=0;
        Consumption = Emes; // remaitre à 0
        Emes=0;
        Cost = CouTotal;
        CouTotal =0;
        Flags.sendconsom = 1;
    }
}

if (++Dispcount >= DispNcount) {
    Dispcount = 0;
    Flags.Display = 1;
}
}
}
}

```

## 7- Affichage sur Navigateur Web

Lorsque le client entre l'IP adresse afficher dans l'ESP32 dans son navigateur, connectant avec le même wifi que l'ESP32 est connecté, une page html afficher les données de consommation en temps réel.

```

server.on("/", HTTP_GET, [](AsyncWebServerRequest *request) {
    int paramsNr = request->params();
    Serial.println("\n-- index.html requested --");
    for (int i = 0; i < paramsNr; i++) {
        AsyncWebParameter *p = request->getParam(i);
        Serial.print("Param name: ");
        Serial.println(p->name());
        Serial.print("Param value: ");
        Serial.println(p->value());

        if (p->name() == "LED") {
            // Actualise le GPIO / Update GPIO //todo revoir
            Serial.println("Set GPIO ");
            Serial.print(p->value());
            if (p->value() == "1") LED = 1;
            else LED = 0;
        }

        if (p->name() == "Iref") {
            Iref = p->value().toDouble();
        }
        if (p->name() == "Vref") {
            Vref = p->value().toDouble();
        }
    }
    timerAlarmDisable(timer);
    request->send(SPIFFS, "/index.html");
    timerAlarmEnable(timer);
});

```

La page html est dans le dossier data qu'on avait flashé dans la mémoire de l'ESP32.

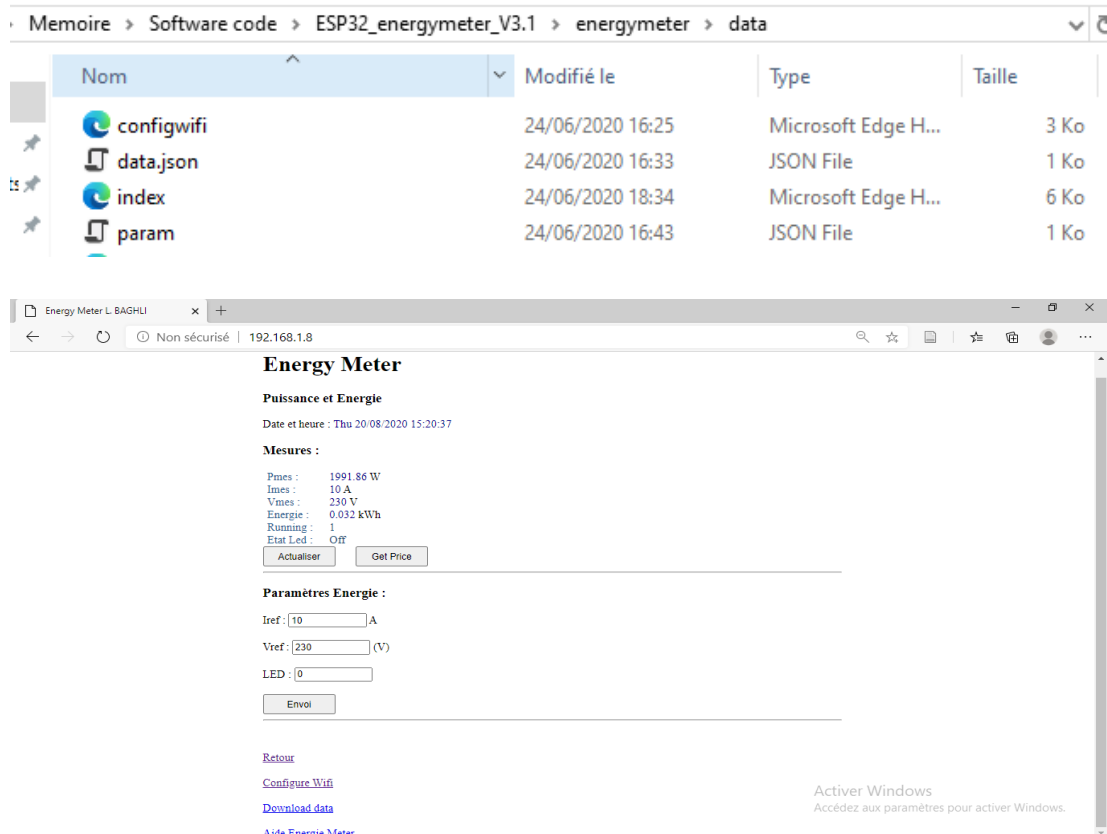


Figure 4-6 : Page HTML Energy meter

## 8- La base de données

Utilisant le MYSQL 5 on crée une base de données pour notre projet, on l'appelle « energymeter ».

**Base de données: energymeter**

Dans cette base de données on crée 5 tables :

- **users** pour enregistrer les données personnelles des clients,
- **totalday** pour enregistrer la consommation et cout total de la journée.
- **consomation15**, pour enregistrer les données de consommation de chaque 15 minute.
- **datestable**, pour enregistrer les dates de consommation et le temps d'envoi des données.
- **price** pour enregistrer les prix de jour.

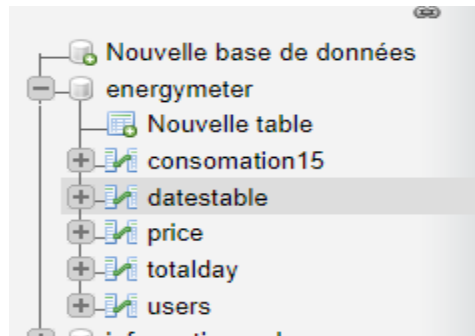


Figure 4-7 : 5 tables de la base de données

### Table users

A un champ User\_ID comme une clé primaire, s'incrémente automatiquement avec chaque utilisateur. Cet ID est utilisé pour créer des relations entre les tables de la base de données.

User_ID	Nclient	FullName	Password	Address	PhoneNumber	MailAddress
5	130001	rafik ramdane	\$2y\$10\$G27d0/.fo3fudw9RIGXeWehjnVzPQU3m7k/pN7cYwyY...	Ghazaouet Tlemcen	NULL	rafik@mail.fr
6	130002	zirar salah	\$2y\$10\$fEC8EWWqPnnXbPREmJRZke8lxO21H1nWO8fGv9sKDd...	chetouane, Tlemcen	NULL	salah@mail.fr

Figure 4-8 : Table users

### Table datestable

Contient champ Date\_ID comme une clé primaire pour qu'on puisse utiliser les dates enregistré dans la table sans besoin de répéter la date dans une autre table.

Date_ID	ConsumDate	SavedDate
19	2020-08-15	2020-08-14 14:52:34

Figure 4-9 : Table datestable

### Table totalday

Contient deux champs User\_ID et Date\_ID comme des clés étrangères qui sont en relation avec les tables précédent.

User_ID	Date_ID	Consumption	Cost
5	19	48	6288
5	19	48	6288
6	19	600	6288

Figure 4-10 : Table totalday

Avec le mot clés « INNER JOIN » on peut afficher les données des 3 tables précédent dans une table à l'aide des clés primaires et clés étrangers, comme l'exemple suivant :

```
SELECT users.Nclient,totalday.Consumption,totalday.Cost,datestable.ConsumDate
FROM totalday
INNER JOIN users
ON users.User_ID = totalday.User_ID
INNER JOIN datestable
ON datestable.Date_ID = totalday.Date_ID
```

Résultat :

Nclient	Consumption	Cost	ConsumDate
130001	48	6288	2020-08-15
130001	48	6288	2020-08-15
130002	600	6288	2020-08-15

Figure 4-11 : Les données des 3 tables précédentes dans une table

## Table Price

Enregistrer les prix de la journée importe de l'Excel de l'opérateur de réseaux.

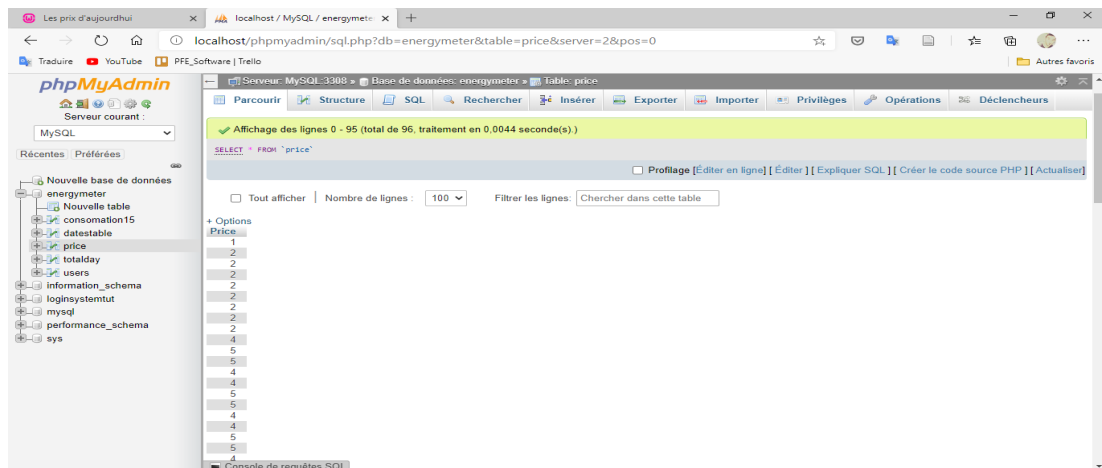


Figure 4-12 : Table Price

## Table consommation15

Contient deux champs User\_ID et Date\_ID comme des clés étrangères qui sont en relation avec les tables précédent (users, datestable).

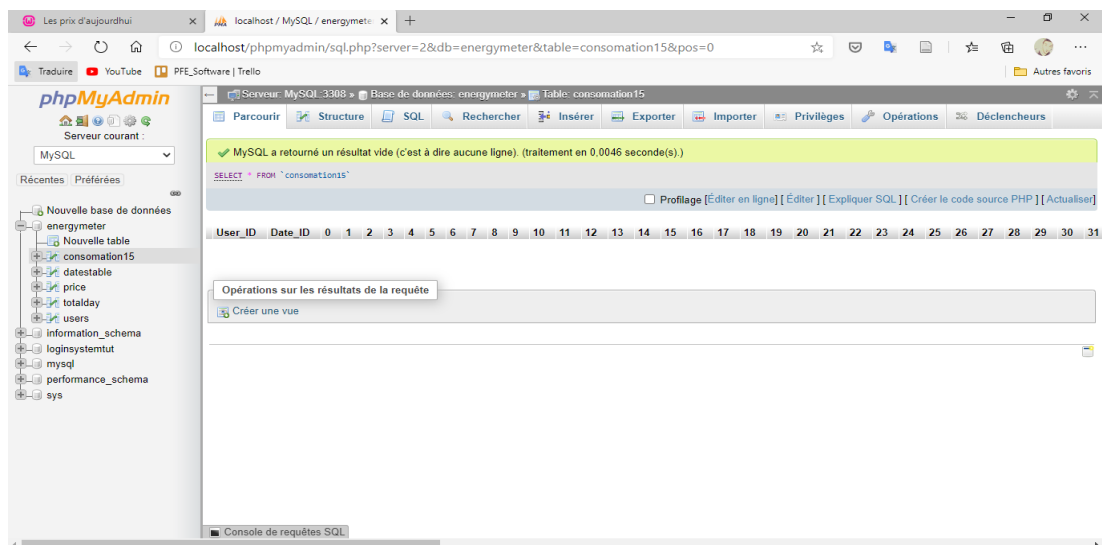


Figure 4-13 : Table consommation

La relation entre les tables de notre base de données :

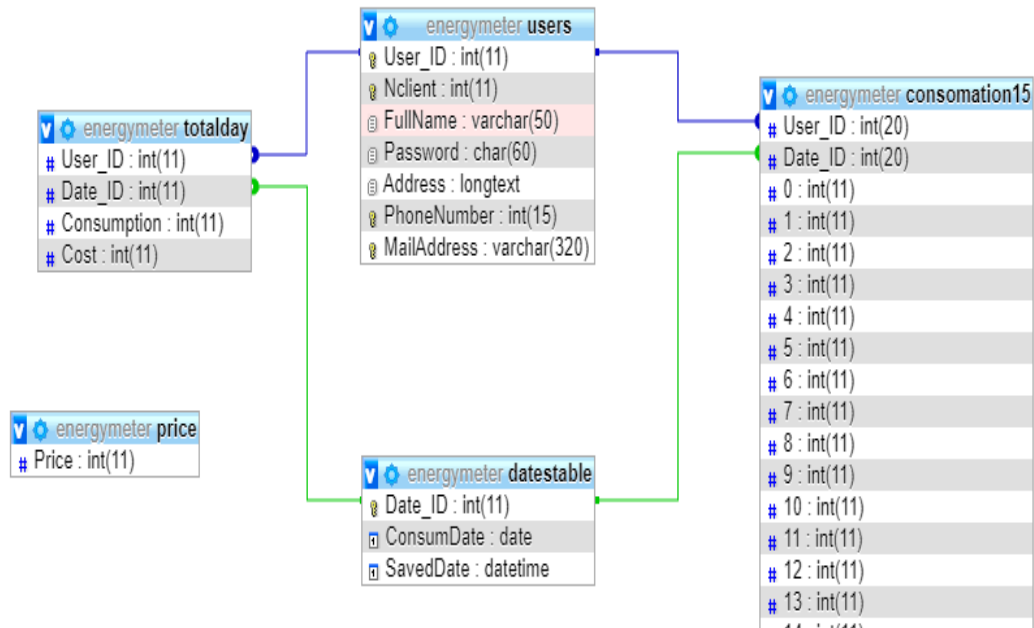


Figure 4-14 : La relation entre les tables de notre base de données

## 9- L'inscription et la connexion

Lors de la demande d'un nouveau compteur, l'employé d'opérateur de réseaux crée un nouveau compte pour ce client.

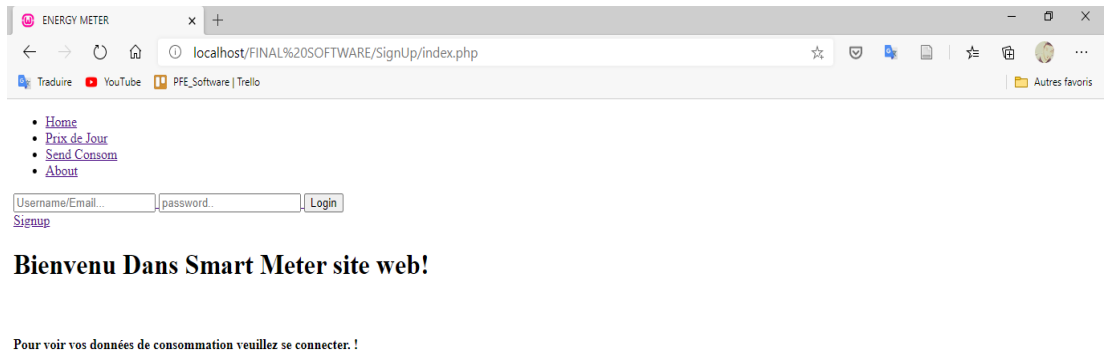


Figure 4-15 : Connexion au site web

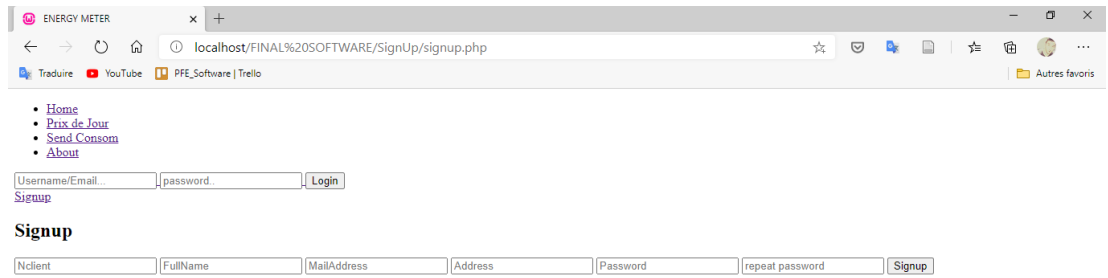


Figure 4-16 : L'inscription au site web

Après cette inscription, il doit entrer le même identifiant Nclient et mot de passe dans le programme de compteur.

```
#define Nclient 130001
#define Password 1234
```

Le client peut aussi utiliser l'identifiant et le mot de passe pour se connecter au site web et voir ses données de consommation et les prix de jour.

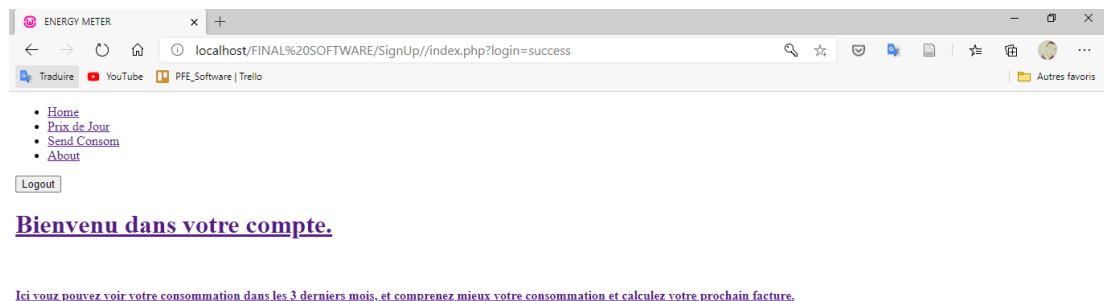


Figure 4-17 : Après connexion au site web

## 10- Le choix des prix de jour

L'employé de l'opérateur de réseaux doit entrer dans le site web et sélectionner le fichier Excel qui contient les prix de la journée suivante.

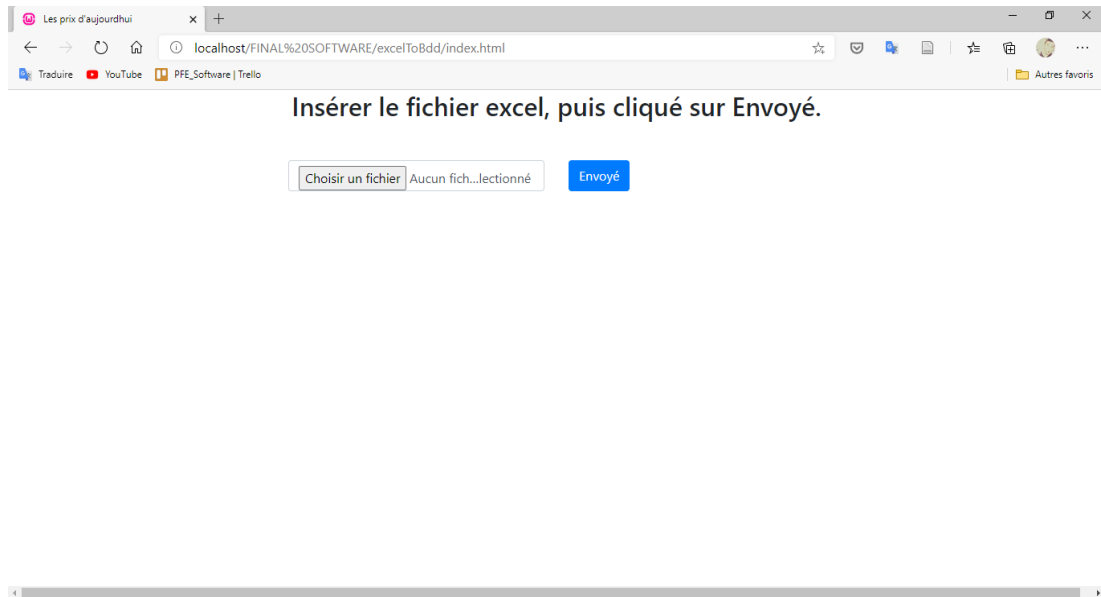


Figure 4-18 : Envoyé les prix de jour

Après avoir cliqué sur « Envoyé » les prix seront transmis directement vers la base de données via un programme PHP.

```
<?php
require 'dbb.inc.php';
$requestPayload = file_get_contents("php://input");
//$prix= json_decode($requestPayload);
$array = json_decode($requestPayload, true);

function array_flatten($array) {
    if (!is_array($array)) {
        return FALSE;
    }
    $result = array();
    foreach ($array as $key => $value) {
        if (is_array($value)) {
            $result = array_merge($result, array_flatten($value));
        }
        else {
            $result[$key] = $value;
        }
    }
    return $result;
}

$array2 = array_column($array, 'data');
var_dump($array2);
$arrayjson = json_encode($array2);
var_dump($arrayjson);
if (!empty($arrayjson)) {
    $query1 = "DELETE FROM `price` ";
    mysqli_query($conn,$query1);
    foreach($array2 as $ar){
        $query = "INSERT INTO price(Price) VALUES('$ar') ";
        mysqli_query($conn,$query);
    }
}
die;
```

Résultat :

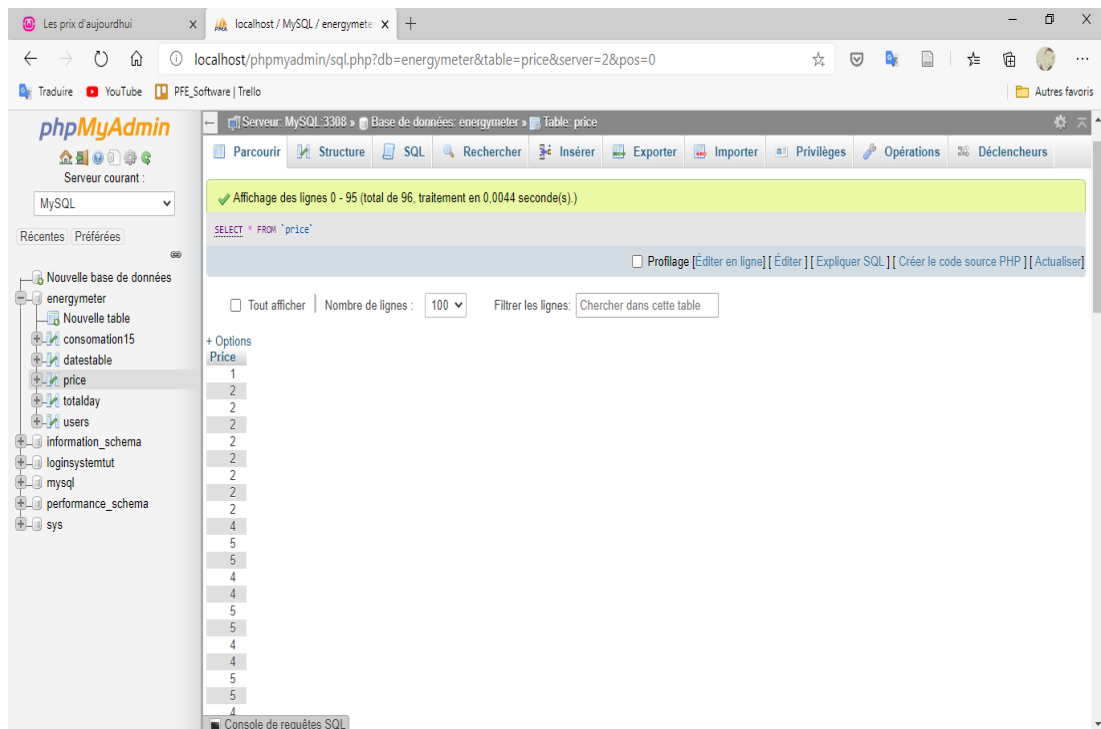


Figure 4-19 : Résultat après l'insertion les donnes d'Excel

Le compteur obtient les prix de la journée tout d'abord via un programme PHP connecter avec la base de données.

```

<?php
// L. Baghli 6/2020
require 'dbb.inc.php';
/////
if (isset($_POST['client'])) $client = $_POST{"client"}; else $client = "";
if ($client == "") if (isset($_GET['client'])) $client = $_GET{"client"};
if ($client == "") $client = 0;

if (isset($_POST['j'])) $j = $_POST{"j"}; else $j = "";
if ($j == "") if (isset($_GET['j'])) $j = $_GET{"j"};
if ($j == "") $j = 0;

$sql="SELECT * FROM price";
$result= mysqli_query($conn,$sql);
$datas= array();
if (mysqli_num_rows($result) > 0) {
    while($row= mysqli_fetch_assoc($result)){
        $datas[]=$row;
    }
}
$array2 = array_column($datas, 'Price');
//print_r($array2);
$prix = json_encode($array2, JSON_NUMERIC_CHECK);
//print_r($prix);

$data = new \stdClass();
$data->client = $client;
$data->j = $j;
$data->prix = $prix;
echo json_encode($data);

```

Puis une fonction getPrice dans l'ESP32 envoie une requete vers ce php pour obtenir les prix de la journée.





```

if (Flags.GetPrice) {
    getPrice();
    Flags.GetPrice = 0;
}

```

## 11- Transmission des données de consommation vers la base de données

Dans la journée, les données de consommation (Energie, cout) sont stockées dans la mémoire de l'ESP32. A minuit, Ces données seront transmises vers notre Base de Données via un programme PHP, ce dernier obtient les données de la part d'ESP32 puis il connecte en vérifiant notre identifiant Nclient et le mot de passe, pour sécuriser le transmettre des données des clients.

Le programme Sendconsom.php :

```

<?php
require 'dbb.inc.php';
$Nclient=$Password=$Cost=$Consumption="";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $Nclient = test_input($_POST["Nclient"]);
    $Password = test_input($_POST["Password"]);
    $Cost = test_input($_POST["Cost"]);
    $Consumption = test_input($_POST["Consumption"]);

    $sql = "SELECT * FROM users WHERE Nclient=$Nclient;";
$result=mysqli_query($conn,$sql);
if ($row = mysqli_fetch_assoc($result)) {
    $pwdCheck = password_verify($Password,$row['Password']); //password => Password , pwd => Password Done -----
    if ($pwdCheck ==false) {
        header("Location: ../index.php?error=wrongpassword");
        exit();
    }else if($pwdCheck== true){
        $result1 = mysqli_query($conn,"SELECT User_ID FROM users WHERE Nclient=$Nclient;");
        $row = mysqli_fetch_assoc($result1);
        $User_ID = $row['User_ID'];
        $indate= "INSERT INTO datestable(ConsumDate,SavedDate) VALUES (SUBDATE(CURDATE(),1),
            CURRENT_TIMESTAMP());";
        $insertdate = mysqli_query($conn,$indate);
        $result2 = mysqli_query($conn,"SELECT `Date_ID` FROM `datestable` WHERE ConsumDate=SUBDATE(CURDATE(),1)");
        ;
        $row=mysqli_fetch_assoc($result2);
        $Date_ID=$row['Date_ID'];
        echo "User_ID:" . $User_ID . "& Date_ID:" . $Date_ID;
        $insertconsum= mysqli_query($conn,"INSERT INTO totalday(User_ID,Date_ID,Cost,Consumption)
        VALUES('" . $User_ID ."','" . $Date_ID ."','" . $Cost ."','" . $Consumption ."');");
    }
}

```

La fonction sendconsom() envoie les données de consommation vers le programme PHP avec la méthode POST.

```

void sendconsom() {
    String payload="", url = serverName+"sendconsom/sendconsom.php";
    //Check WiFi connection status
    if(WiFi.status()== WL_CONNECTED){
        HTTPClient http;

        http.begin(serverName);

        http.addHeader("Content-Type", "application/x-www-form-urlencoded");

        String httpRequestData = "Nclient=" + Nclient
                                + "&Password=" + Password + "&Cost=" + Cost
                                + "&Consumption=" + Consumption +"";
        Serial.print("httpRequestData: ");
        Serial.println(httpRequestData);

        // Send HTTP POST request
        int httpResponseCode = http.POST(httpRequestData);

        if (httpResponseCode>0) {
            Serial.print("HTTP Response code: ");
            Serial.println(httpResponseCode);
        }
        else {
            Serial.print("Error code: ");
            Serial.println(httpResponseCode);
        }
        // Free resources
        http.end();
    }
    else {
        Serial.println("WiFi Disconnected");
    }
}
}

```

Cette fonction est appelée dans loop() lorsque le flag est égal à 1.

```

if (Flags.sendconsom){
    sendconsom();
    Flags.sendconsom=0;
}

```

## V- Sécurité des données

L'un des grands défis des compteurs intelligents dans tous les pays c'est la sécurité des données contre les cyberattaques. Pour faire face à ce défi, Nous avons pris des mesures pour sécuriser le transfert des données vers et depuis l'utilisateur. Nous les mentionnons ces mesures ci-dessous :

### 1- Les mots de passe

L'utilisation des mots de passe pour chaque transmis des données, et avant de stocker les données de consommation aussi, rend l'accès à notre base de donnée difficile.

La vérification des mots de passe avant chaque opération sur les données d'utilisateur, ci-dessous est un exemple dans un programme PHP :

```
$pwdCheck = password_verify($Password,$row['Password']);
```

### Hachage des mots de passe

La fonction **Password\_hash** crée un nouveau hachage de mot de passe en utilisant un algorithme de hachage unidirectionnel puissant.

Le « **PASSWORD\_DEFAULT** » utilisez l'algorithme bcrypt. Notez que cette constante est conçue pour changer avec le temps à mesure que de nouveaux algorithmes plus puissants sont ajoutés à PHP[12].

```
$hashedPassword = password_hash($Password, PASSWORD_DEFAULT);
```

Résultat de hachage :

Password
\$2y\$10\$G27d0/.fo3fudw9RIGXeWehjnVzPQU3m7k/pN7cYwyY...
\$2y\$10\$fEC8EVWqPnnXbPREmJRZke8xO21H1nWO8fGv9sKDdL...

Les résultats présentés ci-dessus sont pour des simples mots de passe faibles comme « 1234 » et « 1235 », cela va donner une puissance pour notre système de sécurité.

### 2- L'utilisation de protocole https

HTTPS permet au visiteur de vérifier l'identité du site web auquel il accède. Il garantit théoriquement la confidentialité et l'intégrité des données envoyées par l'utilisateur et reçue du serveur.

La sécurité des informations transmises par le protocole HTTPS est basée sur l'utilisation d'un algorithme de chiffrement, et sur la reconnaissance de validité du certificat d'authentification du site visité [13].

### 3- Connexion isolé de la base de données

Nous avons créé une connexion avec la base de données dans un fichier PHP isolé, puis on l'appelle à chaque besoin d'une connexion. Cette méthode permet de cacher et protéger les informations sensibles.

Le fichier de connexion avec la base de données :

```

dbb.inc.php x login.inc.php x
<?php

$servername = "localhost:3308";
$dbUsername = "root";
$dbPassword = "";
$dbName = "energymeter"; // energymeter Done

$conn = mysqli_connect($servername,$dbUsername,$dbPassword,$dbName);

if (!$conn) {
    die("Connection field : ".mysqli_connect_error());
}

```

Exemple d'une appelle à ce fichier :

```
require 'dbb.inc.php';
```

### 4- Sécurité de Site Web

Nous avons fait des conditions de sécurité lors de connexion sur le site web, comme la vérification si l'utilisateur a cliqué sur le bouton « login » après avoir entré son identifiant et le mot de passe. Cette vérification empêche l'utilisateur d'accéder avec d'autres moyens, tels que l'utilisation directe d'un lien.

```
if (isset($_POST['login-submit']))
```

L'obligation de traiter les entrées de l'utilisateur (Nclient, email, mot de passe) comme un texte et non pas des lignes de code. Comme cet exemple ici :

```
$sql = "SELECT * FROM users WHERE Nclient=? OR MailAddress=?";
```

## **VI- Conclusion**

Dans ce chapitre nous avons vu comment entrer les offres tarifaires en utilisant un fichier Excel contenant les différents coûts de chaque tranche par jour. Puis, nous avons présenté les différents programmes qui permettent d'obtenir les tarifs J0 et J1 vers le compteur et calculer la consommation électrique et le coût de cette consommation.

Ensuite, nous avons discuté de la création de la base de données et du programme qui permet la transmission des données de consommation vers cette base de données. Enfin, Nous avons traité le problème de la sécurité des données et nous avons proposé des solutions.

## Conclusion générale

Les objectifs fixés dans notre mémoire n'ont pas été tout à fait atteints à cause de la pandémie de COVID-19.

Nous avons réalisé le système de communication désiré pour transmettre les données dans les deux sens de notre compteur intelligent.

Ainsi, dans le deuxième chapitre, nous avons donné une idée générale sur les compteurs intelligents ainsi que ses avantages dont nous avons essayé de profiter dans la deuxième partie de ce chapitre après une étude du comportement de la consommation en Algérie. Cette étude a abouti à une proposition sur la méthode de tarification d'électricité.

Pour développer cette proposition nous avons étudié dans le troisième chapitre les composants nécessaires pour la réalisation de ce compteur intelligent, surtout pour se conformer aux exigences de système de communication tel que la possibilité de connecter au réseau internet sans fil et le stockage des données.

Ces exigences sont garanties avec l'utilisation de l'ESP32. Nous avons également étudié les capteurs de courant et de tension nécessaires pour construire ce compteur, mais cela a été malheureusement difficile en raison de la période de confinement totale en Algérie, ce qui nous a empêchés de les obtenir.

Le chapitre quatre a été la partie la plus importante de ce mémoire, nous avons présenté dans ce chapitre les différents échanges entre le serveur web de l'opérateur réseau électrique et l'ESP32 qui représente le compteur du client. Nous avons réalisé aussi la base de données qui permet d'enregistrer les données personnelles du client et de sa consommation quotidienne. Puis on a vu comment on peut coder les différents tarifs de jour et les utiliser dans la facturation d'une journée de consommation électrique.

Ainsi, ce mémoire nous a permis d'apprendre énormément de choses telles que la programmation de l'ESP32 avec l'environnement de développement Arduino IDE, l'utilisation de la plateforme de développement Web WAMPserver pour créer un serveur local. On a également pu côtoyer plusieurs langages de programmation tel que le PHP, MYSQL et bien sûr le C++.

Nous avons aussi appris comment créer des pages web en utilisant le code HTML et le code CSS. De plus, ce mémoire a permis l'ouverture d'un champ d'application et de recherche relativement important.

Les perspectives que nous pouvons proposer à ce travail sont présentées selon quatre axes importants :

— Prendre en compte les systèmes de communication qu'on a créés dans notre recherche et ajouter des capteurs de courant et de tension performants, ce qui demande des connaissances dans la partie matérielle.

— Ajouter un circuit d'alimentation isolé pour le compteur comme une batterie, pour que les données ne s'effacent pas dans le cas de coupure d'électricité et pour que l'accès soit toujours possible.

— Ajouter un système d'enregistrement plus performant afin de rendre le compteur intelligent capable de sauvegarder les informations de consommation en cas de problème de réseau internet.

— Ajouter un autre système de connexion avec internet comme la 4G pour que la transmission des données ne soit pas tributaire du routeur Wifi du client.







## Bibliographies

- [1] C. Briceño-Garmendia et M. Shkaratan, « Power tariffs », *Policy Research Working Paper*, n° 5904, déc. 2011, [En ligne]. Disponible sur: <http://documents.worldbank.org/curated/en/234441468161963356/pdf/WPS5904.pdf>.
- [2] « Prix du kWh : les offres d'électricité en France – Juin 2020 », *Fournisseur-Energie*, nov. 30, 2016. <https://www.fournisseur-energie.com/prix-kwh/> (consulté le août 16, 2020).
- [3] « Economy 7: How to max your savings - Money Saving Expert ». <https://www.moneysavingexpert.com/utilities/economy-7/#economy10> (consulté le août 16, 2020).
- [4] « Electricity & Energy Tariffs in Victoria – Great Value », *Globird Energy*. <https://www.globirdenergy.com.au/offers-rates/all-tariffs/understanding-tariff/> (consulté le août 16, 2020).
- [5] « Linky », *Wikipédia*. juill. 27, 2020, Consulté le: août 16, 2020. [En ligne]. Disponible sur: <https://fr.wikipedia.org/w/index.php?title=Linky&oldid=173325654>.
- [6] « Compteur Linky et données personnelles | Institut national de la consommation ». <https://www.inc-conso.fr/content/compteur-linky-et-donnees-personnelles> (consulté le août 16, 2020).
- [7] « Smart meter », *Wikipedia*. août 09, 2020, Consulté le: août 16, 2020. [En ligne]. Disponible sur: [https://en.wikipedia.org/w/index.php?title=Smart\\_meter&oldid=971957756](https://en.wikipedia.org/w/index.php?title=Smart_meter&oldid=971957756).
- [8] « An Introduction to Smart Meters », *Greener Kirkcaldy*, avr. 12, 2017. <https://www.greenerkirkcaldy.org.uk/an-introduction-to-smart-meters/> (consulté le août 16, 2020).
- [9] « Comment calculer sa facture d'électricité et gaz en Algérie-Sonelgaz ». <https://youchoz.com/simulation-facture-sonelgaz/> (consulté le août 16, 2020).
- [10] « Non-Invasive Sensor: YHDC SCT013-000 CT used with Arduino. (SCT-013) », *PowerUC*. <https://www.poweruc.pl/blogs/news/non-invasive-sensor-yhdc-sct013-000-ct-used-with-arduino-sct-013> (consulté le août 16, 2020).
- [11] « Wemos LoLin32 ESP32, WiFi + Bluetooth + LiPo battery connector • DIY Projects ». [https://diyprojects.io/deal-wemos-lolin32-esp32-wifi-bluetooth-lipo-battery-connector/#.Xy1Dwa-g\\_IU](https://diyprojects.io/deal-wemos-lolin32-esp32-wifi-bluetooth-lipo-battery-connector/#.Xy1Dwa-g_IU) (consulté le août 16, 2020).
- [12] « PHP: password\_hash - Manual ». <https://www.php.net/manual/en/function.password-hash.php> (consulté le août 17, 2020).
- [13] « Moving towards a more secure web », *Google Online Security Blo*, sept. 2016.

## Annexe A : Grille comparative d'électricité de Total Direct Energie et EDF

Des prix du kWh d'électricité de Total Direct Energie et EDF pour les autres compteurs

Prix du kWh de Total Direct Energie et EDF - à Jour au 5 mars 2020

Puissance compteur	Prix du kWh			
	Heures pleines (HP)		Heures creuses (HC)	
	 EDF	 TOTAL direct energie	 EDF	 TOTAL direct energie
	EDF	Total Direct Énergie	EDF	Total Direct Énergie
3 kVA	--	--	--	--
6 kVA	0.1781 € TTC	0.1640 € TTC	0.1337 € TTC	0.1241 € TTC
9 kVA	0.1781 € TTC	0.1640 € TTC	0.1337 € TTC	0.1241 € TTC
12 kVA	0.1781 € TTC	0.1640 € TTC	0.1337 € TTC	0.1241 € TTC
15 kVA	0.1781 € TTC	0.1640 € TTC	0.1337 € TTC	0.1241 € TTC
18 kVA	0.1781 € TTC	0.1640 € TTC	0.1337 € TTC	0.1241 € TTC
24 kVA	0.1781 € TTC	0.1640 € TTC	0.1337 € TTC	0.1241 € TTC
30 kVA	0.1781 € TTC	0.1640 € TTC	0.1337 € TTC	0.1241 € TTC
36 kVA	0.1781 € TTC	0.1640 € TTC	0.1337 € TTC	0.1241 € TTC

## Annexe B : Tarifs comparés : Economie 7 et Tarifs électricité normal

Tariffs compared: Economy 7 vs a normal electricity tariff				
ELECTRICITY TARIFF	AVERAGE PEAK RATE	AVERAGE OFF-PEAK RATE	AVERAGE STANDING CHARGE	AVERAGE COST (1)
Cheapest Economy 7	15.3p/kWh	8.7p/kWh	14p/day	£577/year
Cheapest normal	13.8p/kWh	13.8p/kWh	9p/day	£612/year

(1) Based on regulator Ofgem's average usage figure for Economy 7 of 4,200 kilowatt hours (kWh) a year. Economy 7 assumes 42% off-peak usage. Last updated: April 2019.

## Annexe C : Tarifs proposés

La tarification proposée est la suivante :

- **Bas** (Tarif 1) : le prix de l'électricité est le plus bas lorsque la demande d'électricité est la plus faible.
- **Modéré** (Tarif 2) : Le prix de l'électricité est inférieur au tarif 3 et tarif 4 et supérieur au tarif 1.
- **Fort** (Tarif 3) : le prix de l'électricité est inférieur au tarif 4 et supérieur au tarif 1 et tarif 2.
- **Très fort** (Tarif 4) : le prix de l'électricité est plus élevé pendant les périodes de surcharge.

		Janvier		Février		Mars		Avril		Mai		Juin		Juillet		Août		Septembre		Octobre		Novembre		Décembre	
Début	Fin	Sem	WE	Sem	WE	Sem	WE	Sem	WE	Sem	WE	Sem	WE	Sem	WE	Sem	WE	Sem	WE	Sem	WE	Sem	WE	Sem	WE
0:00	0:15	1	2	1	1	1	1	1	2	3	3	2	3	3	3	3	3	1	2	1	2	1	1	1	1
0:15	00:30	1	2	1	1	1	1	1	2	3	3	2	3	3	3	3	3	1	2	1	2	1	1	1	1
0:30	0:45	1	2	1	1	1	1	1	2	3	3	2	3	3	3	3	3	1	2	1	2	1	1	1	1
0:45	01:00	1	2	1	1	1	1	1	2	3	3	2	3	3	3	3	3	1	2	1	2	1	1	1	1
1:00	1:15	1	1	1	1	1	1	1	1	3	3	2	3	2	2	2	2	1	1	1	1	1	1	1	1
1:15	01:30	1	1	1	1	1	1	1	1	3	3	2	3	2	2	2	2	1	1	1	1	1	1	1	1
1:30	1:45	1	1	1	1	1	1	1	1	2	2	1	2	2	2	2	2	1	1	1	1	1	1	1	1
1:45	02:00	1	1	1	1	1	1	1	1	2	2	1	2	2	2	2	2	1	1	1	1	1	1	1	1
2:00	2:15	1	1	1	1	1	1	1	1	2	2	1	2	2	2	2	2	1	1	1	1	1	1	1	1
2:15	02:30	1	1	1	1	1	1	1	1	2	2	1	2	2	2	2	2	1	1	1	1	1	1	1	1
2:30	2:45	1	1	1	1	1	1	1	1	2	2	1	2	2	2	2	2	1	1	1	1	1	1	1	1
2:45	03:00	1	1	1	1	1	1	1	1	2	2	1	2	2	2	2	2	1	1	1	1	1	1	1	1
3:00	3:15	1	1	1	1	1	1	1	1	2	2	1	2	2	2	2	2	1	1	1	1	1	1	1	1
3:15	03:30	1	1	1	1	1	1	1	1	2	2	1	2	2	2	2	2	1	1	1	1	1	1	1	1
3:30	3:45	1	1	1	1	1	1	1	1	2	2	1	2	2	2	1	2	1	1	1	1	1	1	1	1
3:45	04:00	1	1	1	1	1	1	1	1	2	2	1	2	2	2	1	2	1	1	1	1	1	1	1	1
4:00	4:15	1	1	1	1	1	1	1	1	2	2	1	1	1	1	1	2	1	1	1	1	1	1	1	1
4:15	04:30	1	1	1	1	1	1	1	1	2	2	1	1	1	1	1	2	1	1	1	1	1	1	1	1

4:30	4:45	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	
4:45	05:00	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	
5:00	5:15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
5:15	05:30	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
5:30	5:45	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
5:45	06:00	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
6:00	6:15	1	1	1	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	2	2	
6:15	06:30	1	1	1	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	2	2	
6:30	6:45	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	1	2	2	
6:45	07:00	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	1	2	2	
7:00	7:15	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	2	1	2	1	2	2	
7:15	07:30	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	2	1	2	1	2	2	
7:30	7:45	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	2	1	2	1	2	2	
7:45	08:00	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	2	1	2	1	2	2	
8:00	8:15	2	2	2	2	2	2	2	1	2	1	1	1	1	1	1	2	1	2	1	3	1	3	2	
8:15	08:30	2	2	2	2	2	2	2	1	2	1	1	1	1	1	1	2	1	2	1	3	1	3	2	
8:30	8:45	2	2	3	3	2	2	2	1	2	1	2	1	2	1	1	1	2	1	2	1	3	1	3	2
8:45	09:00	2	2	3	3	2	2	2	1	2	1	2	1	2	1	1	1	2	1	2	1	3	1	3	2
9:00	9:15	2	2	3	3	2	2	2	2	2	1	2	1	2	1	1	1	2	2	2	2	3	2	3	3
9:15	09:30	2	2	3	3	2	2	2	2	2	1	2	1	2	1	1	1	2	2	2	2	3	2	3	3
9:30	9:45	2	2	3	3	2	2	2	2	3	2	2	2	2	1	2	1	2	2	3	2	3	2	3	3
9:45	10:00	2	2	3	3	2	2	2	2	3	2	2	2	2	1	2	1	2	2	3	2	3	2	3	3
10:00	10:15	2	2	3	3	2	2	2	2	3	2	3	2	2	1	2	1	2	2	3	2	3	2	3	3
10:15	10:30	2	2	3	3	2	2	2	2	3	2	3	2	2	1	2	1	2	2	3	2	3	2	3	3
10:30	10:45	2	2	3	3	2	2	2	2	3	2	3	2	2	1	2	2	2	2	3	2	3	2	3	3
10:45	11:00	2	2	3	3	2	2	2	2	3	2	3	2	2	1	2	2	2	2	3	2	3	2	3	3
11:00	11:15	2	2	3	3	2	2	2	2	3	2	3	2	3	2	2	2	2	2	3	2	3	2	3	3
11:15	11:30	2	2	3	3	2	2	2	2	3	2	3	2	3	2	2	2	2	2	3	2	3	2	3	3
11:30	11:45	2	2	3	3	2	2	2	2	3	2	3	2	3	2	2	2	3	2	3	2	3	2	3	3

11:45	12:00	2	2	3	3	2	2	2	2	3	2	3	2	3	2	2	2	3	2	3	2	3	2	3	3
12:00	12:15	2	2	3	3	2	2	2	2	3	3	3	3	3	2	3	2	3	2	3	2	3	2	3	3
12:15	12:30	2	2	3	3	2	2	2	2	3	3	3	3	3	2	3	2	3	2	3	2	3	2	3	3
12:30	12:45	2	2	3	3	2	2	2	2	3	3	3	3	3	3	3	3	3	2	3	2	3	2	3	3
12:45	13:00	2	2	3	3	2	2	2	2	3	3	3	3	3	3	3	3	3	2	3	2	3	2	3	3
13:00	13:15	2	2	2	3	2	2	2	2	3	3	4	3	4	3	3	3	3	2	3	2	3	2	3	2
13:15	13:30	2	2	2	2	2	2	2	2	3	3	4	3	4	3	3	3	3	2	3	2	3	2	3	2
13:30	13:45	2	2	2	2	2	2	2	2	3	3	4	3	4	4	4	3	3	2	3	2	3	2	3	2
13:45	14:00	2	2	2	2	2	2	2	2	3	3	4	3	4	4	4	3	3	2	3	2	3	2	3	2
14:00	14:15	2	2	2	2	2	2	2	2	3	3	4	3	4	4	4	4	3	2	3	2	2	2	2	2
14:15	14:30	2	2	2	2	2	2	2	2	3	3	4	3	4	4	4	4	3	2	3	2	2	2	2	2
14:30	14:45	2	2	2	2	2	2	2	2	3	3	4	3	4	4	4	4	3	2	3	2	2	2	2	2
14:45	15:00	2	2	2	2	2	2	2	2	3	3	4	3	4	4	4	4	3	2	3	2	2	2	2	2
15:00	15:15	2	2	2	2	2	2	2	2	3	3	4	3	4	4	4	4	3	2	3	2	2	2	2	2
15:15	15:30	2	2	2	2	2	2	2	2	3	3	4	3	4	4	4	4	3	2	3	2	2	2	2	2
15:30	15:45	2	2	2	2	2	2	2	2	3	3	4	3	4	4	4	4	3	2	2	2	2	2	2	2
15:45	16:00	2	2	2	2	2	2	2	2	3	3	4	3	4	4	4	4	3	2	2	2	2	2	2	2
16:00	16:15	2	2	2	2	2	2	2	2	3	3	4	3	4	4	4	4	3	2	2	2	2	2	2	2
16:15	16:30	2	2	2	2	2	2	2	2	3	3	4	3	4	4	4	4	3	2	2	2	2	2	2	2
16:30	16:45	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2
16:45	17:00	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2
17:00	17:15	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2
17:15	17:30	2	2	2	3	2	2	2	2	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2
17:30	17:45	3	3	2	3	2	2	2	2	3	3	3	3	2	2	2	2	2	2	2	2	3	3	3	3
17:45	18:00	3	3	3	3	2	2	2	2	3	3	3	3	2	2	2	2	2	2	2	2	3	3	3	3
18:00	18:15	3	3	3	3	2	2	2	2	3	3	3	3	2	2	2	2	2	2	3	2	3	3	3	4
18:15	18:30	3	3	3	3	2	2	2	2	3	3	3	3	2	2	2	2	2	2	3	2	3	3	3	4
18:30	18:45	4	4	3	3	3	3	2	2	3	3	3	3	2	2	2	2	2	3	4	3	4	4	4	4
18:45	19:00	4	4	3	3	3	3	2	2	3	3	3	3	2	2	2	2	2	3	4	3	4	4	4	4

19:00	19:15	4	4	4	4	4	4	3	3	3	3	3	3	2	2	2	2	3	3	4	3	4	4	4	4
19:15	19:30	4	4	4	4	4	4	3	3	3	3	3	3	2	2	2	2	3	3	4	3	4	4	4	4
19:30	19:45	4	4	4	4	4	4	4	4	3	4	3	3	2	2	2	3	4	4	4	4	4	4	4	4
19:45	20:00	4	4	4	4	4	4	4	4	3	4	3	3	2	2	2	3	4	4	4	4	4	4	4	4
20:00	20:15	4	4	4	4	4	4	4	4	3	4	3	4	2	2	2	3	4	4	4	4	4	3	3	3
20:15	20:30	4	4	4	4	4	4	4	4	3	4	3	4	2	2	2	3	4	4	4	4	4	3	3	3
20:30	20:45	3	4	4	4	4	4	4	4	4	4	4	4	3	3	3	4	4	4	3	4	3	3	3	3
20:45	21:00	3	4	4	4	4	4	4	4	4	4	4	4	3	3	3	4	4	4	3	4	3	3	3	3
21:00	21:15	3	2	3	4	4	3	3	3	4	4	4	4	3	3	3	4	4	3	3	3	3	3	3	3
21:15	21:30	3	2	3	4	4	3	3	3	4	4	4	4	3	3	3	4	4	3	3	3	3	3	3	3
21:30	21:45	3	2	3	4	3	3	3	3	4	4	4	4	3	4	3	4	3	3	2	3	3	3	3	3
21:45	22:00	3	2	3	3	3	3	3	3	4	4	4	4	3	4	3	4	3	3	2	3	3	3	3	3
22:00	22:15	2	2	2	3	3	2	2	3	4	4	4	4	3	4	3	4	3	3	2	3	2	2	2	3
22:15	22:30	2	2	2	3	3	2	2	3	4	4	4	4	3	4	3	4	3	3	2	3	2	2	2	3
22:30	22:45	2	2	2	2	2	2	2	2	4	4	4	4	3	4	3	4	3	3	2	3	2	2	2	2
22:45	23:00	2	2	2	2	2	2	2	2	4	4	4	4	3	4	3	4	3	3	2	3	2	2	2	2
23:00	23:15	2	2	2	2	2	2	2	2	4	3	2	3	3	3	3	4	2	3	2	3	2	2	2	2
23:15	23:30	2	2	2	2	2	2	2	2	4	3	2	3	3	3	3	4	2	3	2	3	2	2	2	2
23:30	23:45	2	2	2	2	2	2	2	2	3	3	2	3	3	3	3	3	2	3	2	3	2	2	2	2
23:45	00:00	2	2	2	2	2	2	2	2	3	3	2	3	3	3	3	3	2	3	2	3	2	2	2	2

## Annexe D : Datasheet – ACS758

## *Thermally Enhanced, Fully Integrated, Hall Effect-Based Linear Current Sensor IC with 100 $\mu\Omega$ Current Conductor*

### FEATURES AND BENEFITS

- Industry-leading noise performance through proprietary amplifier and filter design techniques
- Integrated shield greatly reduces capacitive coupling from current conductor to die due to high  $dV/dt$  signals, and prevents offset drift in high-side, high voltage applications
- Total output error improvement through gain and offset trim over temperature
- Small package size, with easy mounting capability
- Monolithic Hall IC for high reliability
- Ultra-low power loss: 100  $\mu\Omega$  internal conductor resistance
- Galvanic isolation allows use in economical, high-side current sensing in high voltage systems
- AEC Q-100 qualified

Continued on the next page...



### PACKAGE: 5-PIN CB PACKAGE



### DESCRIPTION

The Allegro™ ACS758 family of current sensor ICs provides economical and precise solutions for AC or DC current sensing. Typical applications include motor control, load detection and management, power supply and DC-to-DC converter control, inverter control, and overcurrent fault detection.

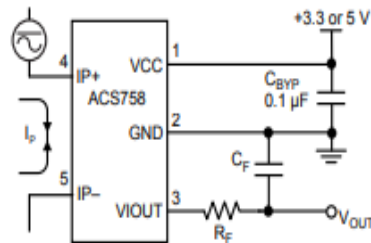
The device consists of a precision, low-offset linear Hall circuit with a copper conduction path located near the die. Applied current flowing through this copper conduction path generates a magnetic field which the Hall IC converts into a proportional voltage. Device accuracy is optimized through the close proximity of the magnetic signal to the Hall transducer. A precise, proportional output voltage is provided by the low-offset, chopper-stabilized BiCMOS Hall IC, which is programmed for accuracy at the factory.

High level immunity to current conductor  $dV/dt$  and stray electric fields, offered by Allegro proprietary integrated shield technology, provides low output voltage ripple and low offset drift in high-side, high voltage applications.

The output of the device has a positive slope ( $>V_{CC}/2$ ) when an increasing current flows through the primary copper conduction path (from terminal 4 to terminal 5), which is the path used for current sampling. The internal resistance of this conductive path is 100  $\mu\Omega$  typical, providing low power loss.

The thickness of the copper conductor allows survival of the device at high overcurrent conditions. The terminals of the conductive path are electrically isolated from the signal leads

Continued on the next page...



Application 1: The ACS758 outputs an analog signal,  $V_{OUT}$ , that varies linearly with the uni- or bi-directional AC or DC primary sampled current,  $I_p$ , within the range specified.  $C_F$  is for optimal noise management, with values that depend on the application.

### Typical Application

**ACS758xCB**

*Thermally Enhanced, Fully Integrated, Hall Effect-Based  
Linear Current Sensor IC with 100  $\mu\Omega$  Current Conductor*

**FEATURES AND BENEFITS (CONTINUED)**

- 3.0 to 5.5 V, single supply operation
- 120 kHz typical bandwidth
- 3  $\mu$ s output rise time in response to step input current
- Output voltage proportional to AC or DC currents
- Factory-trimmed for accuracy
- Extremely stable output offset voltage
- Nearly zero magnetic hysteresis

**DESCRIPTION (CONTINUED)**

(pins 1 through 3). This allows the ACS758 family of sensor ICs to be used in applications requiring electrical isolation without the use of opto-isolators or other costly isolation techniques.

The device is fully calibrated prior to shipment from the factory. The ACS758 family is lead (Pb) free. All leads are plated with 100% matte tin, and there is no Pb inside the package. The heavy gauge leadframe is made of oxygen-free copper.

**Selection Guide**

Part Number <sup>1</sup>	Package		Primary Sampled Current, $I_p$ (A)	Sensitivity Sens (Typ.) (mV/A)	Current Directionality	$T_{op}$ (°C)	Packing <sup>2</sup>
	Terminals	Signal Pins					
ACS758LCB-050B-PFF-T	Formed	Formed	$\pm 50$	40	Bidirectional	-40 to 150	34 pieces per tube
ACS758LCB-050U-PFF-T	Formed	Formed	50	60	Unidirectional		
ACS758LCB-100B-PFF-T	Formed	Formed	$\pm 100$	20	Bidirectional		
ACS758LCB-100B-PSF-T	Straight	Formed	$\pm 100$	20	Bidirectional		
ACS758LCB-100U-PFF-T	Formed	Formed	100	40	Unidirectional		
ACS758KCB-150B-PFF-T	Formed	Formed	$\pm 150$	13.3	Bidirectional	-40 to 125	
ACS758KCB-150B-PSS-T	Straight	Straight	$\pm 150$	13.3	Bidirectional		
ACS758KCB-150U-PFF-T	Formed	Formed	150	26.7	Unidirectional	-40 to 85	
ACS758ECB-200B-PFF-T	Formed	Formed	$\pm 200$	10	Bidirectional		
ACS758ECB-200B-PSF-T	Straight	Formed	$\pm 200$	10	Bidirectional		
ACS758ECB-200B-PSS-T	Straight	Straight	$\pm 200$	10	Bidirectional		
ACS758ECB-200U-PFF-T	Formed	Formed	200	20	Unidirectional		

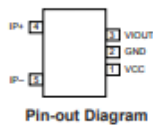
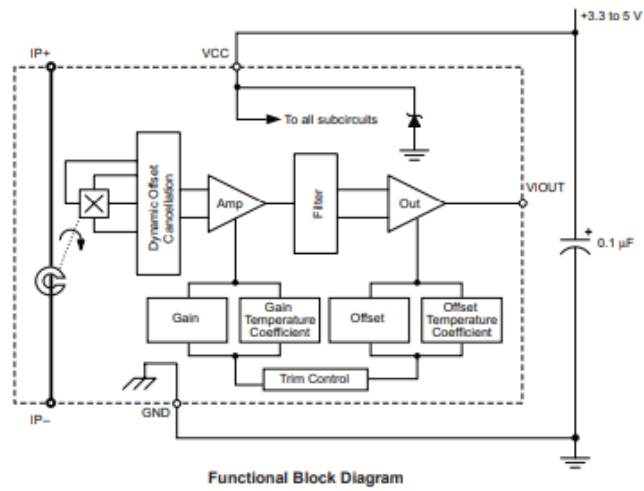
<sup>1</sup>Additional leadform options available for qualified volumes.

<sup>2</sup>Contact Allegro for additional packing options.



**ACS758xCB**

*Thermally Enhanced, Fully Integrated, Hall Effect-Based  
Linear Current Sensor IC with 100  $\mu\Omega$  Current Conductor*

**Terminal List Table**

Number	Name	Description
1	VCC	Device power supply terminal
2	GND	Signal ground terminal
3	VIOU	Analog output signal
4	IP+	Terminal for current being sampled
5	IP-	Terminal for current being sampled

## Annexe E : Datasheet – SCT 013

### 0.333V Split core current transformer



model: SCT013

**Characteristics:**

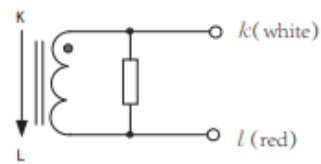
Split core, 0.333V output, Built-in with sampling resistance, leading wire 1 metre, standardΦ3.5 three core plug output, (Patent no.: ZL 2015 3 0060067.X)



**Technical index:**

Suspended mounting, output with cable  
 Operation temperature: -25 C ~ +70 C  
 Storage temperature: -30 C ~ +90 C  
 work voltage: ≤660V  
 Work frequency: 50Hz-1KHz  
 Dielectric strength: 3.5KV 50Hz 1min

**Wiring schematic:**



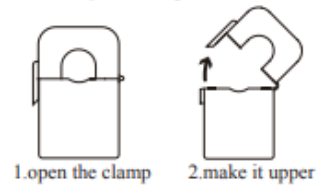
Voltage output type : Built-in with sampling resistance  
 Voltage output type : not allowed secondary short circuit.  
 When the plug without audio frequency, the white line is the end of the same name

**Technical Parameters table:**

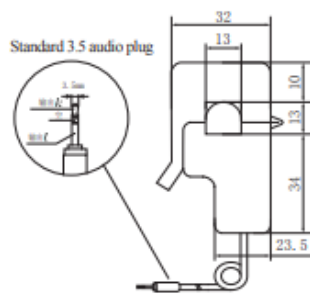
Rated Input(Effective value)	10 20 30 50 80 100 A	
measurement range	20% ~ 120%	A
Rated output	0.333	V
Auaccuracy	1	%
Linearity	1	%
frequency	50 ~ 1K	Hz
weight	50	g
Order Format: model Input output Auaccuracy SCT013/xxA/0.333V/1%		
Noted: The rated current ≤100A are the standard product		

**Installation Diagram**

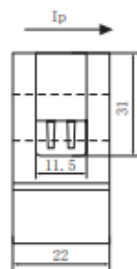
Primary coil through hole method



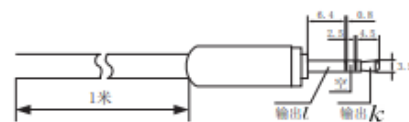
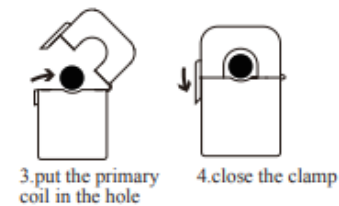
**Outline size: (in:mm):**



Front view



Side view



Standard three core plug schematic diagram

## Annexe F : Datasheet – ACPL-C87AT

### ACPL-C87AT/ACPL-C87BT Automotive High Precision DC Voltage Isolation Sensor

#### Description

The Broadcom<sup>®</sup> ACPL-C87AT/C87BT isolation sensors utilize superior optical coupling technology, with sigma-delta ( $\Sigma-\Delta$ ) analog-to-digital converter, chopper stabilized amplifiers, and a fully differential circuit topology to provide unequalled isolation-mode noise rejection, low offset, high gain accuracy and stability.

ACPL-C87AT ( $\pm 1\%$  gain tolerance) and ACPL-C87BT ( $\pm 0.5\%$  gain tolerance) are designed for high precision DC voltage sensing in electronic motor drives, DC/DC and AC/DC converter and battery monitoring system. The ACPL-C87AT/C87BT features high input impedance and operate with full span of analog input voltage up to 2.46V. The shutdown feature provides power saving and can be controlled from external source, such as microprocessor.

The high common-mode transient immunity (15 kV/ $\mu$ s) of the ACPL-C87AT/C87BT maintains the precision and stability needed to accurately monitor DC rail voltage in high noise motor control environments. This galvanic safe isolation solution is delivered in a compact, surface mount stretched SO-8 (SSO-8) package that meets worldwide regulatory safety standards.

Broadcom R<sup>2</sup>Coupler<sup>®</sup> isolation products provide the reinforced insulation and reliability needed for critical automotive and high temperature industrial applications.

#### Features

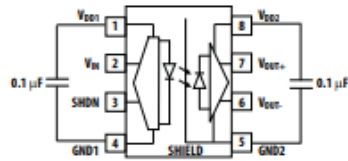
- Unity gain
- $\pm 0.5\%$  (ACPL-C87BT) and  $\pm 1\%$  (ACPL-C87AT) gain tolerance @ 25°C
- $-0.3$  mV Input offset voltage
- 0.05% non linearity
- 25 ppm/°C gain drift vs. temperature
- 100 kHz bandwidth
- 0 to 2V nominal input range
- Qualified to AEC-Q100 Grade 1 test guidelines
- Operating temperature:  $-40^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$
- Shutdown feature (active high)
- 15 kV/ $\mu$ s common-mode rejection at  $V_{\text{CM}} = 1$  kV
- Working voltage,  $V_{\text{ORM}} = 1414 V_{\text{peak}}$
- Compact, surface mount stretched SO8 package
- Worldwide safety approval:
  - UL 1577 (5000  $V_{\text{RMS}}$ /1 minute)
  - CSA
  - IEC/EN/DIN EN 60747-5-5

#### Applications

- Automotive BMS battery pack voltage sensing
- Automotive DC/DC converter voltage sensing
- Automotive motor inverter DC bus voltage sensing
- Automotive AC/DC (charger) DC output voltage sensing
- Isolation interface for temperature sensing
- General-purpose voltage sensing and monitoring

## Functional Diagram

Figure 1: Functional Diagram



A 0.1-µF bypass capacitor must be connected between pin 1 and pin 4, and pin 5 and pin 8 as shown.

Figure 2: Functional Diagram 2

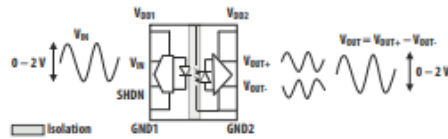
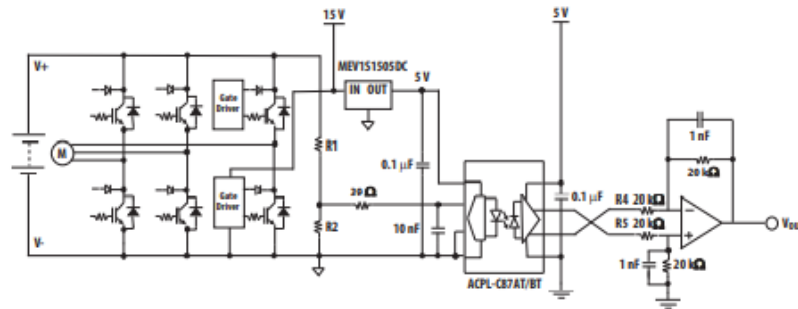


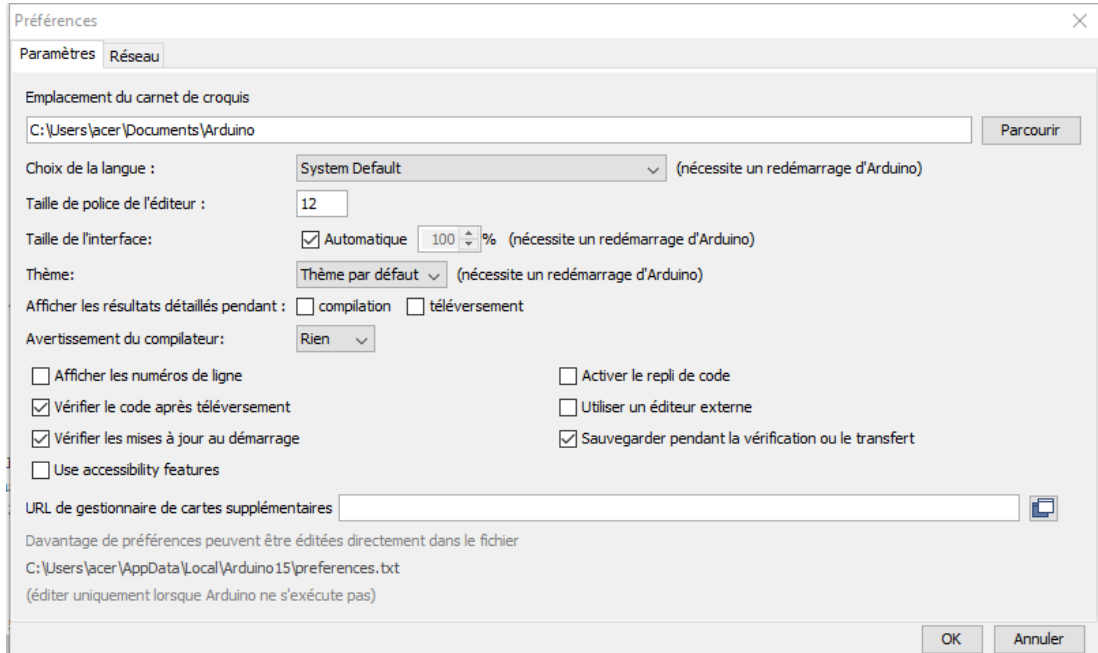
Figure 3: Typical Voltage Sensing Circuit



## Annexe G : Préparer l'environnement de l'IDE

1. Installer ESP32 : Pour qu'on puisse programmer l'ESP32 avec L'IDE on doit installer un module complémentaire de l'ESP32, Pour cela on suit les instructions suivantes :

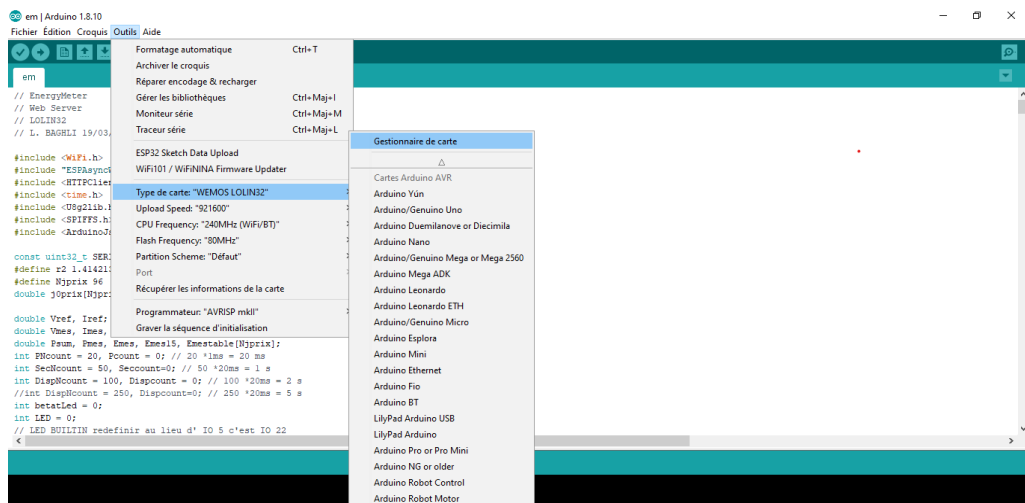
(1) Dans IDE Arduino, on doit accéder à Fichier > Préférences



(2) Dans l'URL de gestionnaire de cartes supplémentaires, on entre le lien suivant :

« [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json),  
[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json) »

(3) Ouvre le gestionnaire de tableaux, puis accède à Outils > Type de carte > Gestionnaire de carte

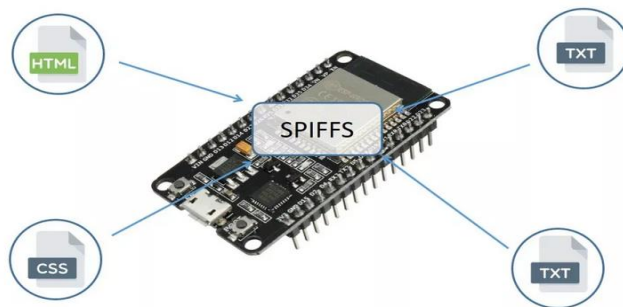




(4) Recherche ESP32 et installe le

## 2. Installer SPIFFS :

L'ESP32 tout comme l'ESP8266 contient un SPIFFS (Serial Peripheral Interface Flash File System). SPIFFS est un système de fichiers léger créé pour les microcontrôleurs avec une puce flash, qui sont connectés par bus SPI, comme la mémoire flash ESP32. Nous allons montrer comment télécharger facilement des fichiers sur le système de fichiers ESP32 à l'aide d'un plugin pour Arduino IDE.



- (1) Accédez à la page des versions et cliquez sur le fichier ESP32FS-1.0.zip à télécharger.
- (2) Accédez au répertoire Arduino IDE et ouvrez le dossier Tools.

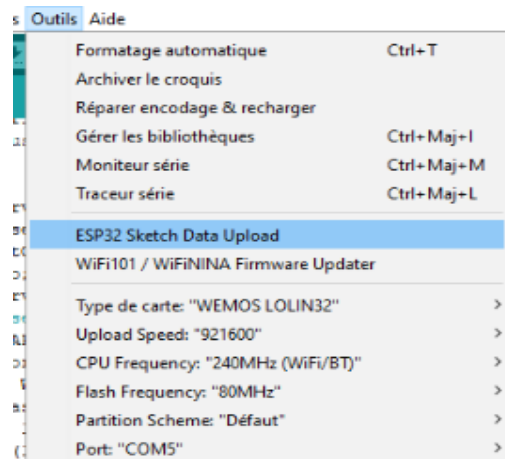
» Ce PC » Disque local (C:) » Programmes (x86) » Arduino » tools

- (3) Décompressez le dossier .zip téléchargé dans le dossier Tools. Vous devriez avoir une structure de dossiers similaire :

» Ce PC » Disque local (C:) » Programmes (x86) » Arduino » tools » ESP32FS » tool

Nom	Modifié le	Type	Taille
esp32fs	15/01/2019 10:53	Archive WinRAR	8 Ko

Enfin, démarrez Arduino IDE. Pour vérifier si le plugin a été installé avec succès, ouvrez l'IDE Arduino. Sélectionnez la carte ESP32, allez dans Outils et vérifiez qu'on a l'option «ESP32 Sketch Data Upload».



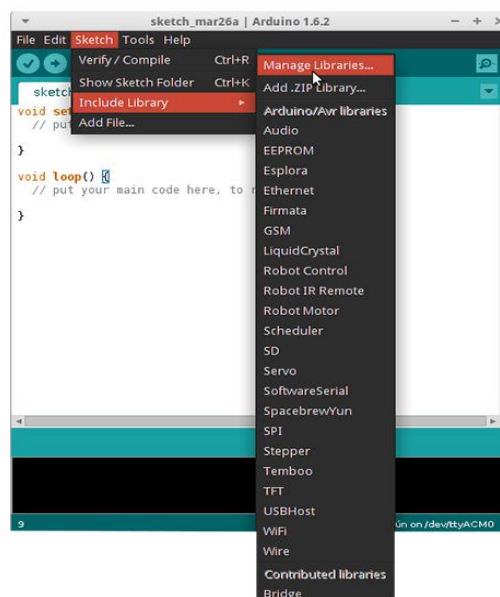
### 3. Installation des bibliothèques :

#### a) Définition des bibliothèques :

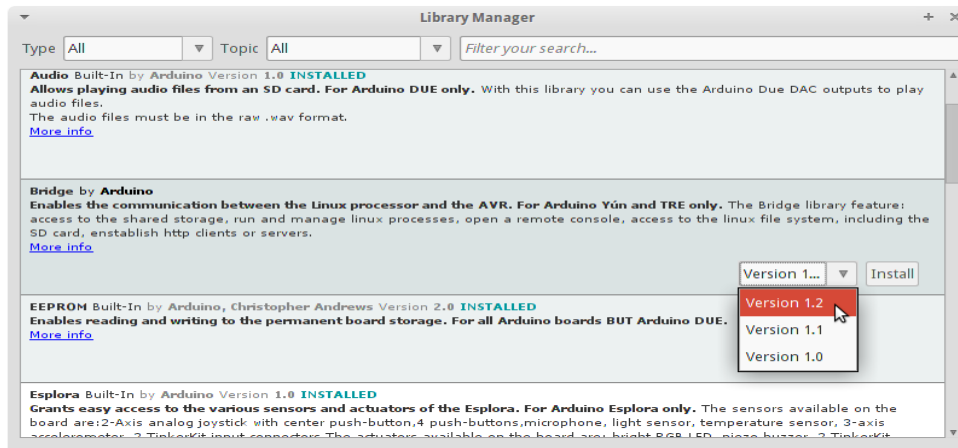
Les bibliothèques sont une collection de code qui permet de connecter facilement à un capteur, un affichage, un module, etc.

#### b) Installation d'une bibliothèque:

Pour installer une nouvelle bibliothèque dans l'Arduino IDE, il suffit d'utiliser le gestionnaire de bibliothèque (disponible à partir de la version IDE 1.6.2). Ouvrir l'EDI et cliquer sur le menu "Sketch" puis sur Inclure la bibliothèque > Gérer les bibliothèques.



Ensuite, le gestionnaire de bibliothèque s'ouvrira et on trouve une liste des bibliothèques qui sont déjà installées ou prêtes pour l'installation. Il faut défiler la liste pour trouver la bibliothèque qu'on est besoin, cliquer dessus, puis sélectionnez la version de la bibliothèque qu'on veut l'installer. Parfois, une seule version de la bibliothèque est disponible.



Enfin, on clique sur installer et on attende que l'EDI installe la nouvelle bibliothèque.

Importer une bibliothèque .zip

Les bibliothèques sont souvent distribuées sous forme de fichier ou de dossier ZIP. Le nom du dossier est le nom de la bibliothèque. Le dossier contient un fichier .cpp, un fichier .h et souvent un fichier keywords.txt, un dossier d'exemples et d'autres fichiers requis par la bibliothèque. À partir de la version 1.0.5, on peut installer des bibliothèques tierces dans l'EDI. On n'est pas besoin de décompresser le ZIP.

Dans l'IDE Arduino, on accède à Sketch > Inclure la bibliothèque > Ajouter une bibliothèque .ZIP. En haut de la liste déroulante, sélectionner l'option "Ajouter une bibliothèque .ZIP".



On est besoin de sélectionner la bibliothèque qu'on veut l'ajouter. Accédé à l'emplacement du fichier .zip et l'ouvrir.

Revenir au menu Esquisse > Inclure la bibliothèque. Menu. On peut maintenant voir la bibliothèque en bas du menu déroulant. Il est prêt à être utilisé dans notre croquis. Le fichier zip aura été développé dans le dossier des bibliothèques de notre répertoire de croquis Arduino.



## Résumé

Les compteurs intelligents sont la clé de la transition vers un réseau intelligent, grâce à la collection des données détaillées. L'étude de ces données permet de prédire et d'influencer le comportement des consommateurs. Pour réaliser ce but, nous avons présenté les composants nécessaires pour réaliser ce projet. Ensuite, nous avons réalisé un prototype de système de communication et de calcul de consommation énergétique à l'aide d'un ESP32.

## Mots clés

Compteur intelligent, ESP 32, Capteur, Base de données, SONEGAS

---

## Abstract

Smart meters are the key to the transition to a smart grid, through the collection of detailed data. Analysing this data helps predict and influence consumer behavior. To achieve this goal, we presented the components required to complete this project. We built a prototype that compute the energy consumption and communicate with a server using an ESP32.

## Keywords:

Smart meter, ESP32, Sensor, Database, SONEGAS

---

## الملخص

العدادات الذكية هي مفتاح الانتقال إلى الشبكة الذكية وذلك بفضل جمع البيانات بالتفصيل. تساعد دراسة هذه البيانات في التنبؤ بسلوك المستهلك والتأثير عليه. لتحقيق هذا الهدف، تحدثنا عن المكونات اللازمة لإنجاز هذا المشروع. بعد ذلك صنعنا نموذجًا أوليًا لحساب الاستهلاك و نظام الاتصالات باستخدام ESP32.

## الكلمات المفتاحية :

العداد الذكي، ESP32، مستشعر، قاعدة البيانات، سونلغاز