



وزارة البحث العلمي والتعليم العالي  
MINISTERE DE L'ENSEIGNEMENT SUPEREUR  
ET DE LA RECHERCHE SCIENTIFIQUE  
Université Abdelhamid Ibn Badis Mostaganem



UNIVERSITE  
Abdelhamid Ibn Badis  
MOSTAGANEM

جامعة عبد الحميد بن باديس مستغانم

Faculté des Sciences et de la Technologie

كلية العلوم والتكنولوجيا

DEPARTEMENT DE GENIE ÉLECTRIQUE

**Mémoire Pour l'obtention du diplôme de Master**

**Filière : Électronique**

**Spécialité : Électronique des systèmes embarqués**

**Sous thème :**

**Conception et réalisation d'un compteur d'énergie  
électrique intelligent de type prépayé**

**Présenté et soutenu publiquement par :**

-MAKHLOUF Chaïma

-BOUSSIF Sid Ahmed

**Devant le jury composé de :**

Nom et Prénom	Grade	Établissement	Qualité
ABDELLAOUI N.	MAA	Université de mostaganem	Encadrant
AZZEDINE M.	MAA	Université de mostaganem	Président
KOUADRIA	MAA	Université de mostaganem	Examineur
REBHI M.	MAA	Université de mostaganem	Co-encadrant

Année universitaire : 2023-2024

## Remerciements

Tout d'abord, nous remercions **Allah** de nous avoir donné la capacité et la volonté et la patience d'aller jusqu'au bout de nos études.

Nous tenons à remercier de tout notre cœur Mr. **ABDELLAOUI Nasreddine**, notre encadrant de mémoire, pour son soutien, son expertise et ses conseils tout au long de notre parcours. Son dévouement et son engagement envers notre développement professionnel ont été inestimables.

Nous tenons à remercier Mr. **REBHI Mustapha**, notre co-encadrant pour son soutien et ses conseils avisés.

Nous exprimons également notre gratitude aux **membres du jury** pour avoir accepté d'évaluer notre mémoire.

Nous n'oublierons pas **nos parents** pour leur contribution, leur soutien et leur patience. Nous tenons à exprimer notre reconnaissance envers eux.

Nous tenons à exprimer notre gratitude envers tous ceux qui ont contribué, de près ou de loin, à la réalisation de ce modeste travail.

Merci à toutes et à tous.

# DÉDICACES

Je dédie ce modeste travail :

À mes **parents**, merci infiniment d'être toujours présents pour moi, de me féliciter quand tout allait bien et de me soutenir quand c'était nécessaire. Je n'en serais pas là sans eux.

À ma chère sœur **Halima**.

À mes chers frères **Abdelkrim** et **Mansour**.

À mon neveu **Adem Sohaib**.

À toute la famille **MAKHLOUF** et **ZIANE**.

À mes **amies**.

À mon binôme **Sid Ahmed**.

À tous ceux qui aiment **Chaïma**.

**MAKHLOUF Chaïma**

# DÉDICACES

Je dédie ce modeste travail aux êtres qui me sont chers. Je cite exclusivement :

Mes chers **parents** qui sont les premiers qui ont cru en moi, Le présent travail représente le fruit du soutien des encouragements ils ont prodigué tout au long de mes études est grâce à eux que j'ai pu arriver à ce que je suis aujourd'hui car aucun travail n'accomplit dans la solitude.

Mes chères **sœurs**.

Mon frère **Abed**.

Mon binôme **Chaïma**.

L'enseignant **MASROUCH Mohamed**.

Mon ami **BAIDI Abdelkader M**.

Toute personne ayant une place dans mon cœur.

**BOUSSIF Sid Ahmed**

*Résumé*

*ملخص*

*Abstract*

## Résumé

L'objectif de ce projet est de concevoir et de réaliser un compteur d'énergie électrique intelligent type prépayé, permettant de pallier au problème des compteurs post payés utilisés dans notre pays à savoir : ne permettent pas le suivi en temps réel et à distance de la consommation d'électricité et par conséquent, peuvent entraîner chez les clients, des factures d'électricité élevées.

Pour ce faire, le compteur à réaliser doit donc être un objet IoT, capable de fournir aux clients, des informations utiles, en temps réel, à distance, à travers internet, leurs permettant de suivre leurs consommations et à prendre conscience instantanément de leurs consommations et de les réguler.

Ce projet peut s'inscrire dans le contexte de la transition dans notre pays, vers des réseaux électriques plus intelligents et durables, visant à optimiser la consommation énergétique et à favoriser une gestion plus responsable des ressources.

**Mot clés :** Compteur intelligent, prépayé, Internet des objets, gestion de la consommation d'électricité, temps réel.

## Abstract

The aim of this project is to design an intelligent prepaid energy meter that will overcome the problem of the post-paid meters used in our country, i.e., they don't allow real-time remote monitoring of electricity consumption and can therefore lead to high electricity bills for customers.

To achieve this, the meter to be built must therefore be an IoT object, capable of providing customers with useful information, in real time, remotely, via the Internet, enabling them to monitor their consumption and become instantly aware of and regulate their consumption.

This project is part of our country's transition towards smarter, more sustainable electricity networks, aimed at optimizing energy consumption and promoting more responsible resource management.

**Key words :** Smart meter, prepaid, Internet of Things, electricity consumption, real time.

## ملخص

الهدف من هذا المشروع هو تصميم وإنتاج عداد كهرباء ذكي مدفوع مسبقاً يتغلب على مشكلة العدادات المدفوعة اجلا المستخدمة في بلدنا، وهي أنها لا تسمح بمراقبة استهلاك الكهرباء عن بُعد وبالتالي يمكن أن تؤدي إلى ارتفاع فواتير الكهرباء للعملاء . ولتحقيق ذلك، يجب أن يكون العداد الذي سيتم تركيبه عبارة عن عداد إنترنت الأشياء، قادر على تزويد العملاء بمعلومات عن بُعد عبر الإنترنت، مما يمكنهم من مراقبة استهلاكهم وتنظيمه.

يعد هذا المشروع جزءاً من التحول في بلدنا نحو شبكات كهرباء أكثر ذكاءً واستدامة، بهدف تحسين استهلاك الطاقة وتعزيز الإدارة الأكثر مسؤولية للموارد.

**الكلمات المفتاحية:** عداد ذكي، مدفوع مسبقاً، إنترنت الأشياء، استهلاك الكهرباء، وقت حقيقي

*Liste des figures*

*Liste des tableaux*

*Liste des abréviations*

# LISTE DES FIGURES

## Chapitre I : Généralités sur les compteurs d'énergie électrique

<b>Figure I.1</b> : Les différents types de compteurs d'énergie électrique.....	03
<b>Figure 0.2</b> : Architecture interne simplifiée d'un compteur électronique analogique.....	04
<b>Figure 0.3</b> : Exemple de compteur électronique, monophasé non communiquant type analogique.....	04
<b>Figure 0.4</b> : Architecture interne simplifiée d'un compteur électronique, numérique réalisé à l'aide d'un microcontrôleur métrologique.....	05
<b>0.5</b> : architecture interne simplifiée d'un compteur électronique numérique réalisé à l'aide d'une combinaison de circuit intégré dédié plus un MCU.....	06

## Chapitre II : L'IoT et son rapport avec les compteurs d'énergie électrique intelligents

<b>Figure II-1</b> : Les trois types de connectivité M2M .....	08
<b>Figure II.2</b> : Architecture de base de l'IoT.....	10
<b>Figure II.3</b> : Structure du réseau IoT.....	12

## Chapitre III : Réalisation matérielle et logicielle

<b>Figure III.1</b> : Schéma bloc du compteur intelligent réalisé avec description fonctionnelle des différents blocs.....	14
<b>Figure III.2</b> : Le Compteur d'énergie électrique, électronique, Monophasé modèle 230VAC/5-32A – DDS1732-1Y.....	15
<b>Figure III.3</b> : Brochage du compteur DDS1732-1Y et son interfaçage à un MCU.....	16
<b>Figure III.4</b> : Interfaçage d'un LCD I2C 16x2 à un MCU.....	16
<b>Figure III.5</b> : Interfaçage d'un clavier matriciel I2C 4x4 à un MCU.....	17
<b>Figure III.6</b> : Interfaçage du relai DC To AC SSR-25DA à un MCU.....	17
<b>Figure III.7</b> : L'esp32 38 pin.....	18
<b>Figure III.8</b> : Interfaçage des composants utilisés à l'esp32.....	19
<b>Figure III.9</b> : vue de dessus et de coté de la maquette du compteur prépayé réalisé.....	20
<b>Figure III.10</b> : organigrammes de l'application embarquée réalisée.....	22
<b>Figure III.11</b> : IHM réalisée.....	23

# LISTE DES TABLEAUX

## Chapitre III : Réalisation matérielle et logicielle

<b>Tableau III.1</b> : Les différents types des composants utilisée.....	14
--	----

# LISTE DES ABRÉVIATIONS

**IoT** : Internet of Things

**Kwh** : Kilowattheure

**IP** : Internet Protocol

**M2M** : Machine to Machine

**WiFi** : Wireless Fidelity

**BLE** : Bluetooth Low Energy

**RPMA** : Random Phase Multiple Access

**NB-IoT** : Narrowband Internet of Things

**3G** : Troisième Génération

**4G** : Quatrième Génération

**5G** : Cinquième Génération

**MQTT** : Message Queuing Telemetry Transport

**CoAP** : Constrained Application Protocol

**HTTP** : Hypertext Transfer Protocol

**IHM** : Interface Homme Machine

**M2P** : Machine to People

**LCD** : Liquid Crystal Display

**AC** : Alternating Current

**Vcc** : Voltage Collector Commun

**Gnd**: Ground

**I2C** : Inter Integrated Circuit

**SSR** : Solide State Relay

**MCU** : Microcontroller Unit

**SDA** : Serial Data Line

**SCL** : Serial Clock Line

**SoC** : System on Chip

**Kib RAM** : Kibibyte Random Acces Memory

**Kib ROM** : Kibibyte Read Only Memory

**HTML** : Hypertext Markup Language

# TABLE DES MATIERES

RESUME

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ABRÉVIATIONS

INTRODUCTION GENERALE ..... 1

## Chapitre I : Généralités sur les compteurs d'énergie électrique

I.1. INTRODUCTION.....	2
I.2. DEFINITION D'UN COMPTEUR D'ENERGIE ELECTRIQUE.....	2
I.3. LES DIFFERENTS TYPES DE COMPTEUR D'ENERGIE ELECTRIQUE .....	2
<i>I.3.1. Les compteurs d'énergie électrique, électroniques.....</i>	<i>3</i>
<i>I.3.2 : Les compteurs d'énergie électrique, électronique, type analogique.....</i>	<i>3</i>
<i>I.3.3 : Les compteurs d'énergie électrique, électronique, type numérique.....</i>	<i>5</i>
I.4. CONCLUSION .....	6

## Chapitre II : L'IoT et son rapport avec les compteurs d'énergie électrique intelligents

II.1. INTRODUCTION.....	7
II.2. DEFINITIONS DE L'IOT .....	7
II.3. CARACTERISTIQUES D'UN RESEAU IOT.....	7
<i>II.3.1. La Connectivité des objets IoT.....</i>	<i>8</i>
<i>II.3.2. L'identité des objets IoT.....</i>	<i>8</i>
<i>II.3.3. Le partage des données entre les objets IoT.....</i>	<i>9</i>
<i>II.3.4. L'Intelligence embarquée sur les objets IoT.....</i>	<i>9</i>
<i>II.3.5. L'Évolutivité des objets de l'IoT.....</i>	<i>9</i>
<i>II.3.6. L'Architecture de l'IoT.....</i>	<i>10</i>
<i>II.3.7. La sécurité des objets.....</i>	<i>10</i>
II.4. FONCTIONNEMENT D'UN RESEAU IOT.....	10
II.5. DOMAINES APPLICATIFS DE L'IOT .....	11
II.6. STRUCTURE DU RESEAU IOT .....	11
II.7. L'IOT ET SON RAPPORT AVEC LE COMPTAGE INTELLIGENT D'ENERGIE ELECTRIQUE .....	13
II.8. CONCLUSION.....	13

## Chapitre III : Réalisation matérielle et logicielle

III.1. INTRODUCTION .....	14
III.2. REALISATION MATERIELLE .....	14
<i>III. 2.1. Schéma bloc du compteur intelligent réalisé.....</i>	<i>14</i>
<i>III.2.2. Les composants matériels utilisés pour la réalisation des différents blocs du compteur ....</i>	<i>15</i>
<i>III.2.3. Interfaçages des différents composants utilisés avec l'esp32 .....</i>	<i>19</i>
<i>III.2.4. Vue de la maquette du compteur réalisé.....</i>	<i>20</i>
III.3. REALISATION LOGICIELLE .....	20
<i>III.3.1. L'application embarquée.....</i>	<i>21</i>
<i>III.3.2. L'IHM.....</i>	<i>23</i>
III.4. CONCLUSION .....	24

CONCLUSION GENERALE ..... 25

BIBLIOGRAPHIES ..... 26

ANNEXE ..... 27

# *Introduction générale*

## Introduction générale

Avoir un raccordement électrique post-payé dans son foyer peut présenter des inconvénients majeurs. En effet, puisque l'on ne peut pas suivre la consommation d'électricité, cela peut entraîner des factures d'électricité élevées. Le branchement électrique prépayé peut être une solution à ce problème et peut empêcher la sur- utilisation de l'électricité. Avec une connexion électrique prépayée, le consommateur recharge le compteur intelligent avec l'argent équivalent à la quantité d'électricité qu'il souhaite disposer pour ses dépenses. De cette façon, lorsque le montant disponible atteint un solde bas, le consommateur sera informé et lorsque le compteur atteindra zéro, l'alimentation électrique sera automatiquement coupée et restera ainsi jusqu'à la prochaine recharge.

C'est dans ce contexte que le sujet de notre projet se situe ; il s'agit de tenter de réaliser un tel système.

Ceci étant dit, le mémoire de notre projet est divisé en trois chapitres :

Le 1<sup>er</sup> chapitre est consacré aux généralités sur les compteurs d'énergie électrique, le 2<sup>ème</sup> à l'IoT et son rapport avec les compteurs d'énergie électrique intelligents et enfin le 3<sup>ème</sup> à la réalisation matérielle et logicielle de notre compteur.

**Chapitre I :**  
**Généralités sur les compteurs**  
**d'énergie électrique**

## I.1.Introduction

Dans ce chapitre, après avoir défini ce qu'est un compteur d'énergie électrique, nous en donnerons une classification accompagnée d'exemples de circuits intégrés dédiés et de microcontrôleurs que l'on peut utiliser pour réaliser ces différents types de compteurs.

## I.2. Définition d'un compteur d'énergie électrique

L'énergie électrique est la puissance totale délivrée ou consommée sur un intervalle de temps  $[0, t]$  et donnée par la relation suivante :

$$W = \int_0^t v(t)i(t)dt \quad (I-1)$$

Si  $v(t)$  est exprimé en volts,  $i(t)$  en ampères et 't'en secondes, l'unité d'énergie est le joule ou le watt-seconde. L'unité commerciale de l'énergie électrique étant le kilowattheure (KWh).

Un instrument utilisé pour mesurer la quantité d'énergie donnée par la relation (I-1), est appelé compteur d'énergie électriques ou compteur de wattheures. Il permet aux services publics d'électricité de déterminer la consommation d'énergie sur une période donnée à des fins de facturation et aux clients de surveiller leur consommation d'énergie électrique.

## I.3. Les différents types de compteur d'énergie électrique

À l'origine ces appareils étaient de conception électromécanique, ils sont remplacés actuellement par des modèles électroniques. Les nouvelles versions de compteurs électriques sont des compteurs communicants appelés également « compteurs intelligents » ou compteurs IoT.

On en donne à la figure I.1 suivante, une classification :

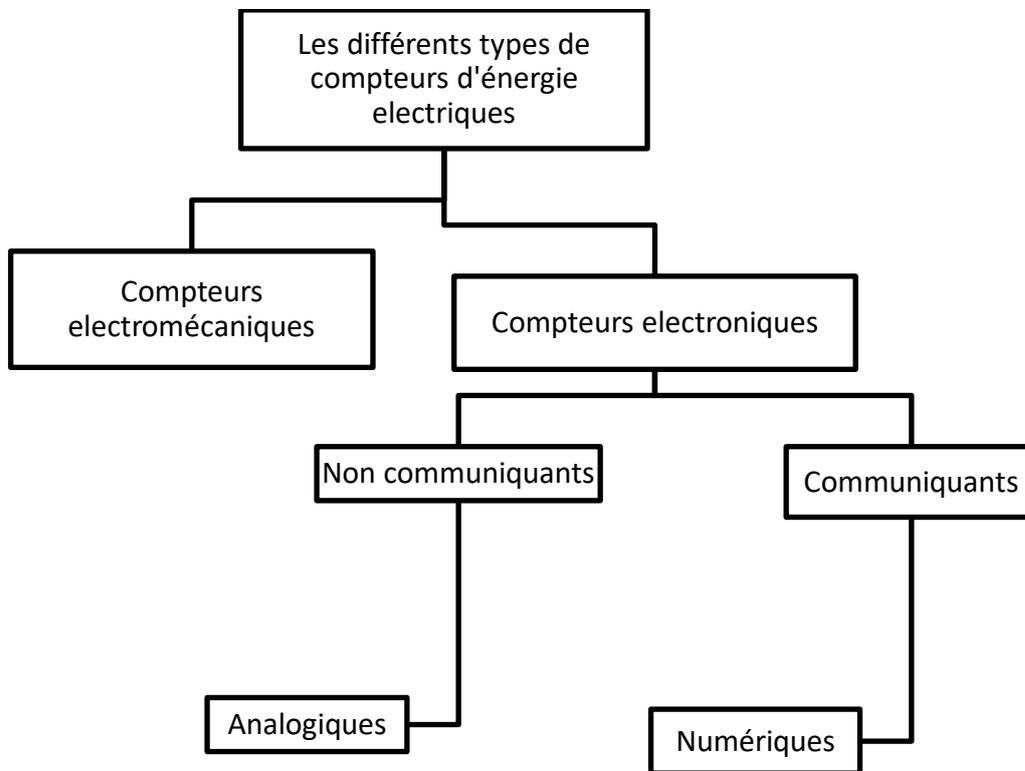


Figure I.1: Les différents types des compteurs d'énergie électrique

Dans les paragraphes qui suivent, nous allons en faire une présentation, en nous limitant à la classe électronique, la classe électromécanique n'étant plus utilisée.

### **I.3.1. Les compteurs d'énergie électrique, électroniques**

Les compteurs électroniques remplacent le disque physique et les engrenages par une mesure électronique de la tension et du courant pour calculer la puissance consommée.

Une caractéristique commune à tous les compteurs de ce type, est la mise à disposition, d'une sortie impulsionnelle, où une impulsion correspond à une quantité bien précise d'énergie consommée.

### **I.3.2 : Les compteurs d'énergie électrique, électronique, type analogique**

Dans ce type de compteurs, la mesure de la puissance est prise en charge entièrement par des circuits intégrés dédiés à la mesure de la puissance (figure I.2).

## Chapitre I : Généralités sur les compteurs d'énergie électrique

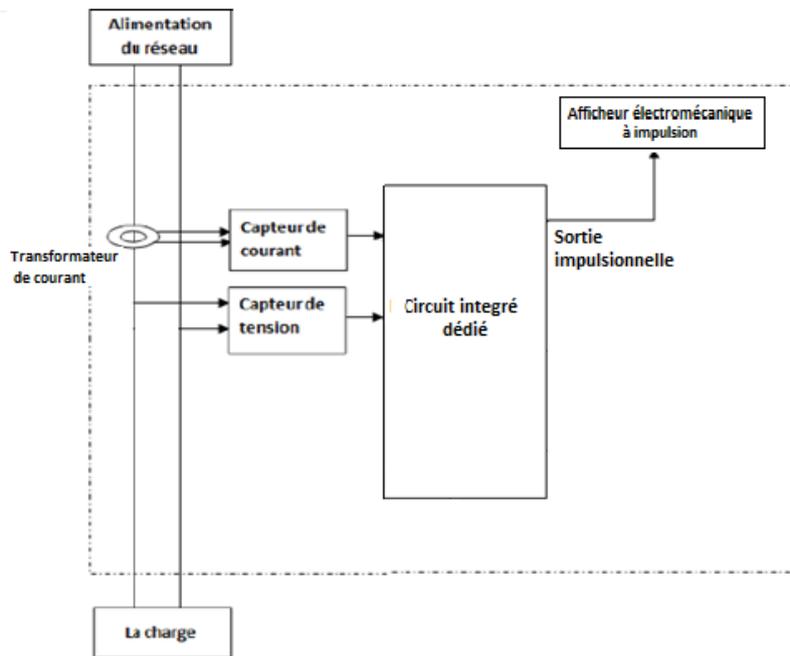


Figure I.2: architecture interne simplifiée d'un compteur électronique analogique

Avec ce genre de circuits intégrés dédiés, aucune programmation n'est nécessaire. Et c'est la raison pour laquelle les compteurs d'énergie électrique réalisés à l'aide de ces puces sont qualifiés d'analogiques et non communicant. Ils sont dits non communicant car de part leurs conceptions, ils n'ont pas d'interface de communication et par conséquent, ils ne peuvent pas communiquer avec leur gestionnaire de réseau électrique.

La figure I.3 ci-dessous, montre l'aspect extérieur d'un exemple de compteur d'énergie électronique, type analogique.



Figure I.3: Exemple de compteur électronique, monophasé non communicant type analogique

## I.3.3 : Les compteurs d'énergie électrique, électronique, type numérique

Dit aussi intelligents ou communicant, dans ce type de compteurs, la mesure de la puissance est prise en charge soit entièrement par des microcontrôleurs métrologiques dédiés, soit par une combinaison : circuit intégré dédié à la mesure de l'énergie électrique + microcontrôleur à usage général pour le contrôle de l'ensemble des activités du compteur numérique. Et c'est la raison pour laquelle ils sont qualifiés de numériques.

Ils sont dits communicant car contrairement aux compteurs non communicant, de part leurs conceptions, ils ont une ou plusieurs interfaces de communication et par conséquent, ils sont capables de communiquer avec leur gestionnaire de réseau électrique.

On donne à la figure I.4, l'architecture interne de base d'un compteur numérique, réalisé à l'aide d'un microcontrôleur métrologique dédié.

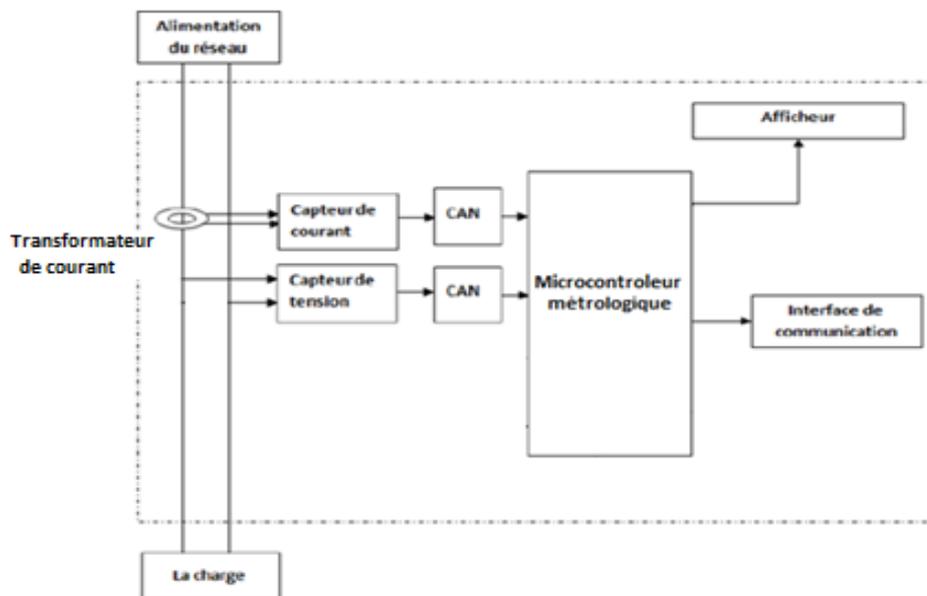


Figure I.4: architecture interne simplifiée d'un compteur électronique, numérique réalisé à l'aide d'un microcontrôleur métrologique

De même, on donne à la figure I.5, un exemple de réalisation d'un compteur numérique à l'aide d'une combinaison : circuit intégré dédié + MCU

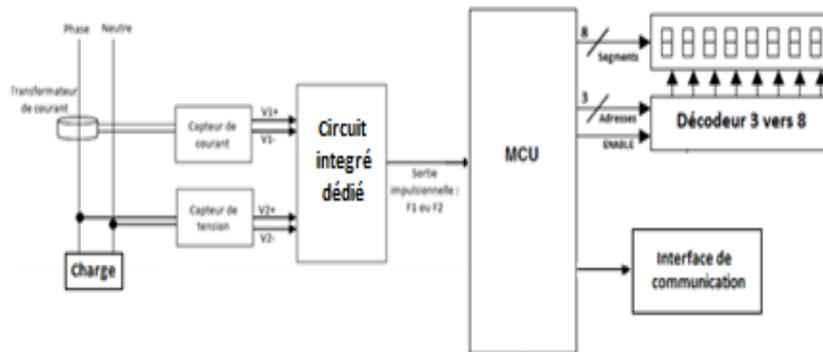


Figure I.5: architecture interne simplifiée d'un compteur électronique numérique réalisé à l'aide d'une combinaison de circuit intégré dédié plus un MCU

#### I.4. Conclusion

Dans ce chapitre, après avoir donné une définition des compteurs d'énergie électrique, nous en avons fait une classification.

**Chapitre II :**  
**L'IoT et son rapport avec les**  
**compteurs d'énergie électrique**  
**intelligents**

### II.1. Introduction

Evolution naturelle des technologies, l'IoT est un lien inévitable entre le monde numérique et le monde physique.

Maintenant, et parce qu'un objet compatible IoT est un objet qui incarne les caractéristiques de ce dernier, par conséquent, on ne peut pas parler de 'comptage intelligent d'énergie électrique' sans parler de l'IoT. Ce chapitre est donc consacré à ce réseau.

Les points suivants sont, par conséquent, discutés : nous allons commencer par définir ce qu'est l'IoT, en donner ses caractéristiques, décrire son fonctionnement, citer quelques exemples de domaines applicatifs, en donner sa structure et enfin montrer son rapport tout évident avec le comptage intelligent de l'énergie électrique.

### II.2. Définitions de l'IoT

L'IoT est un réseau qui interconnecte Internet et les objets physiques du monde dans lequel nous vivons. Autrement dit, c'est une extension du réseau internet aux objets du monde physique.

Les objets dans l'IoT sont constitués par des systèmes embarqués associés aux objets physiques du monde dans lequel nous vivons.

Les systèmes embarqués des objets IoT sont dotés de :

- Capteurs et actionneurs, leurs permettant d'interagir avec l'environnement physique.
- Connectivités M2M (Machine To Machine) locales ou distantes, leurs permettant de communiquer entre eux sans ou avec une intervention humaine très limitée.
- Connectivité IP (Internet Protocole), permettant de les connecter à Internet.

Et ce, à des fins d'automatisation, de supervision, de contrôle, de suivi et de prédiction.

### II.3. Caractéristiques d'un réseau IoT

Ces caractéristiques sont :

- La connectivité des objets,
- L'identité des objets,
- Le partage des données entre les objets,
- L'Intelligence embarquée sur les objets,
- L'évolutivité des objets,
- L'architecture de l'IoT et enfin
- La sécurité objets.

### II.3.1. La Connectivité des objets IoT

La connectivité est la capacité de communiquer et de partager des informations entre deux ou plusieurs objets.

L'IoT crée un monde où tout est connecté grâce à la connectivité des objets. Ces objets peuvent être n'importe quoi. La connectivité permet aux objets de l'IoT, en plus d'échanger des données, d'être contrôlés à distance via Internet.

La connectivité dans l'IoT est double : M2M et IP.

La M2M est mise en œuvre grâce à des réseaux câblé ou sans fil. Un réseau M2M est un réseau dans lequel les objets peuvent communiquer directement.

La connectivité IP, quand à elle, est mise en œuvre à l'aide des protocoles basés sur le protocole IP d'Internet.

On donne à la figure II-1 ci-dessous, les différents types de réseaux M2M sans fil (car les prédominants dans l'IoT) et sur lesquels s'appuie une connectivité IP, classés en fonction de leur portée.

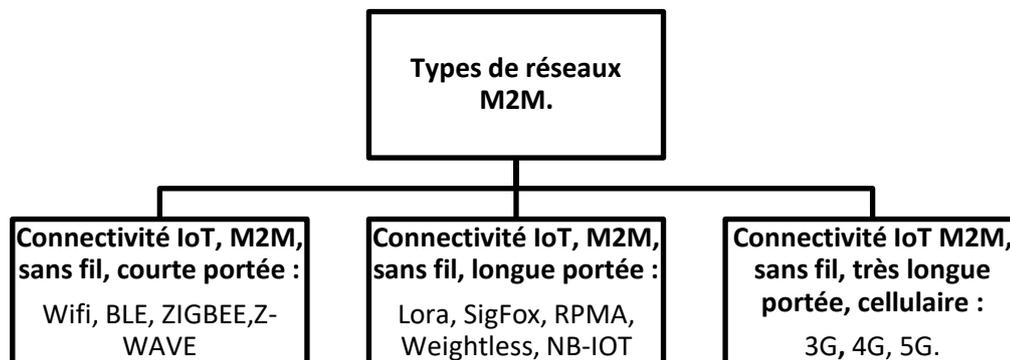


Figure II-1 : Les trois Types de connectivité M2M.

### II.3.2. L'identité des objets IoT

L'identité est la caractéristique unique d'un objet. C'est ce qui permet d'identifier de manière unique un objet. C'est la seule chose qui rend un objet unique et identifiable. Elle peut être utilisée pour distinguer les objets et de les contrôler.

Par exemple, l'adresse IP peut être utilisée comme identifiant unique représentant un objet dans un réseau. Elle peut-être une adresse IP dynamique publique, une adresse IP fixe publique ou une adresse IP fixe privée.

### **II.3.3. Le partage des données entre les objets IoT**

Le partage des données entre les objets IoT fait référence à la capacité des objets connectés à échanger des informations entre eux de manière transparente et sécurisée.

Le partage de données permet aux objets IoT de coopérer, de prendre des décisions et d'agir en fonction des données collectées.

Grâce à l'IoT, les données peuvent être partagées par les objets IoT (objet IoT pris au sens le plus large) et ainsi permettre de mieux gérer, surveiller, suivre, contrôler, ... etc. et de prendre de meilleures décisions.

Le partage de données peut être réalisé via des protocoles de communication standardisés tels que *MQTT*, *CoAP*, *WebSocket*, *HTTP*, ... etc. et le plus souvent, implique des technologies telles que les passerelles IoT, les services Cloud et les réseaux de capteurs.

### **II.3.4. L'Intelligence embarquée sur les objets IoT**

L'intelligence embarquée sur les objets IoT fait référence à la capacité des objets connectés à traiter, analyser et prendre des décisions localement, sans nécessiter une connexion constante à un réseau externe ou à un serveur distant. Cette intelligence embarquée permet aux objets IoT de fonctionner de manière autonome, ce qui peut améliorer la réactivité, réduire la latence et économiser de la bande passante en ne transmettant que les données pertinentes vers une plate-forme distante (*cloud computing*). Ceci peut être particulièrement utile dans les environnements où la connectivité est limitée ou peu fiable.

Cette intelligence peut être mise œuvre à l'aide d'algorithmes développés en utilisant la logique floue ou les réseaux de neurones.

### **II.3.5. L'Évolutivité des objets de l'IoT**

L'évolutivité des objets IoT se réfère à leur capacité à s'adapter et à évoluer pour répondre aux besoins changeants des utilisateurs et des environnements. Cela inclut la capacité des objets à être mis à niveau avec de nouvelles fonctionnalités, à s'intégrer à de nouveaux systèmes et à être facilement extensibles pour prendre en charge un plus grand nombre d'applications et de cas d'utilisation. Pour garantir une évolutivité efficace, les objets doivent être conçus compatibles avec des normes ouvertes et des interfaces flexibles, permettant une intégration aisée dans l'écosystème IoT.

L'évolutivité des objets IoT est donc la capacité de ces derniers à se développer sans que leur performance soit affectée. Techniquement parlant, elle peut être réalisée en ajoutant des ressources matérielles ou en ajoutant des couches logicielles supplémentaires à un objet IoT ou système existant.

### II.3.6. L'Architecture de l'IoT

C'est une structure globale ou représentation schématique qui définit comment les différents composants du réseau IoT interagissent entre eux pour collecter, traiter, stocker et partager des données.

C'est l'enchevêtrement de composants tels que : capteurs, actionneurs, services en "*Edge computing*" en "*cloud computing*", protocoles ouverts ou propriétaires et couches qui composent les systèmes de mise en réseau IoT.

C'est une architecture en couches.

On donne à la figure II.2 ci-dessous, l'architecture de base à titre illustratif car il n'existe pas en pratique jusqu'à l'heure actuelle un consensus sur le nombre de couches que devrait avoir cette architecture.

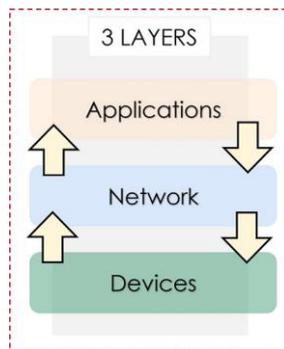


Figure II.2 : Architecture de base de l'IoT

### II.3.7. La sécurité des objets

On le rappelle : l'idée de l'IoT, est que tout ce qui nous entoure sera connecté à Internet pour nous faciliter la vie. Cependant, et en contrepartie, ceci implique des risques : les cyber-attaques. Aussi et afin d'être à l'abri de ces dernières, il faut prendre des mesures de sécurité lors du développement d'un objet IoT.

## II.4. Fonctionnement d'un réseau IoT

Le fonctionnement de l'IoT, commence au niveau des objets capteurs qui collectent des données et les envoient vers une plate-forme locale "*edge computing*" ou distante "*cloud computing*". Cette dernière, après traitements et analyses, envoie des commandes à des objets actionneurs.

Les capteurs, les actionneurs, les plateformes locales et distantes, peuvent être contrôlés par l'utilisateur à travers une IHM.

## II.5. Domaines applicatifs de l'IoT

- Ville intelligente : circulation routière intelligente, transports intelligents, collecte des déchets, cartographies diverses (bruit, énergie, etc.).
  - Environnements intelligents : prédiction des séismes, détection d'incendies, qualité de l'air, ...etc.
  - Sécurité et gestion des urgences : radiations, attentats, explosions.
  - Logistique : aller plus loin que les approches actuelles.
  - Contrôle industriel : mesure, pronostic et prédiction des pannes, dépannage à distance.
  - Santé : suivi des paramètres biologiques à distance.
  - Agriculture intelligente, domotique, applications ludiques.
  - **Le comptage intelligent de l'énergie électrique.**
- Etc... pour ne citer que ceux-là et la liste est encore plus longue.

## II.6. Structure du réseau IoT

Avoir une vue de la structure de ce réseau, permet de mieux voir de quoi il s'agit quand on parle d'IoT. Aussi, on donne à la figure II-3 ci-dessous, la composition de ce type de réseau.

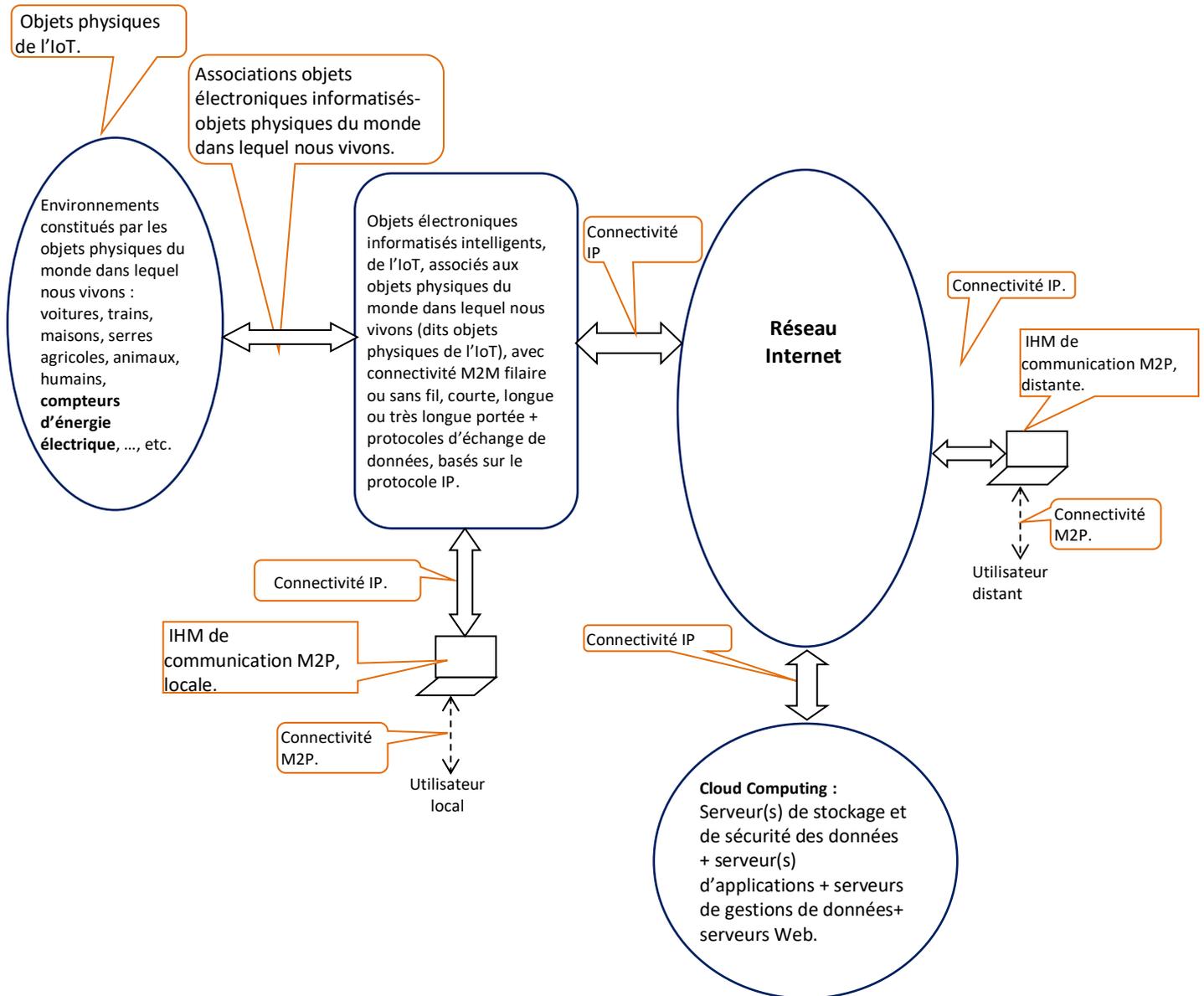


Figure II-3 : Structure du réseau IoT

### **II.7. L'IoT et son rapport avec le comptage intelligent d'énergie électrique**

Les compteurs d'énergie électriques intelligents constituent de nos jours, la génération de compteurs qui permettent entre autres, aux utilisateurs de surveiller leur consommation d'énergie en temps réel et de prendre des décisions éclairées concernant leurs consommations. Or, ceci ne peut être obtenu que si et seulement si, le compteur d'énergie électrique est connecté à Internet ; autrement dit : un objet IoT et par conséquent, doté d'une connectivité M2M-IP.

### **II.8. Conclusion**

Dans ce chapitre nous avons présenté l'IoT de manière plus ou moins détaillée, et son rapport avec les compteurs d'énergie électrique communicant.

**Chapitre III :**  
**Réalisation matérielle et logicielle**

## III.1. Introduction

Dans ce chapitre les deux points suivants vont être présentés et discutés : Tout d'abord le schéma synoptique du compteur d'énergie que nous avons réalisé va être présenté avec description fonctionnelle. Ensuite vont être présentées les différents composant matérielles que nous avons utilisé pour réaliser les différentes parties du compteur, et enfin va être présenté le schéma électrique global de ce dernier.

## III.2. Réalisation matérielle

### III. 2.1. Schéma bloc du compteur intelligent réalisé. (Figure III.1)

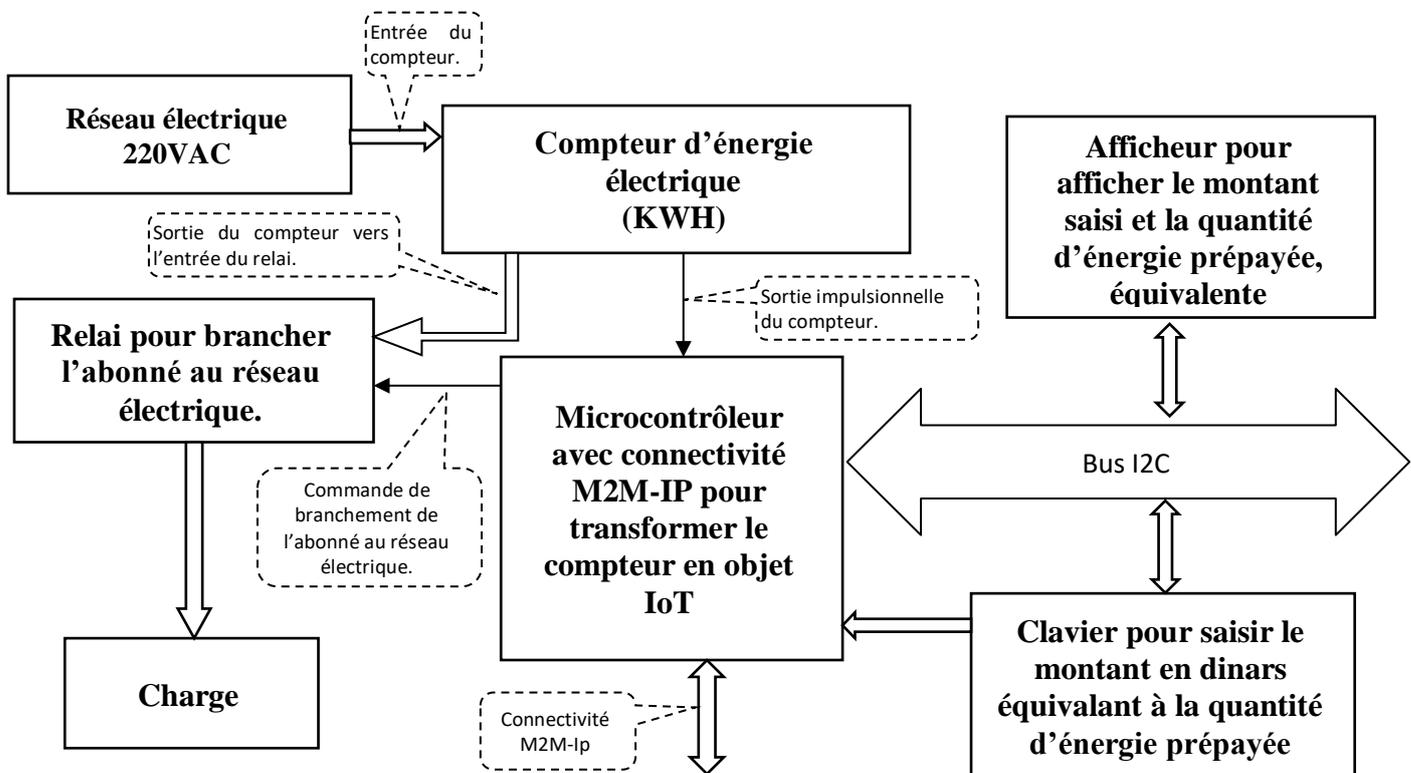


Figure III.1: Schéma bloc du compteur intelligent réalisé avec description fonctionnelle des différents blocs



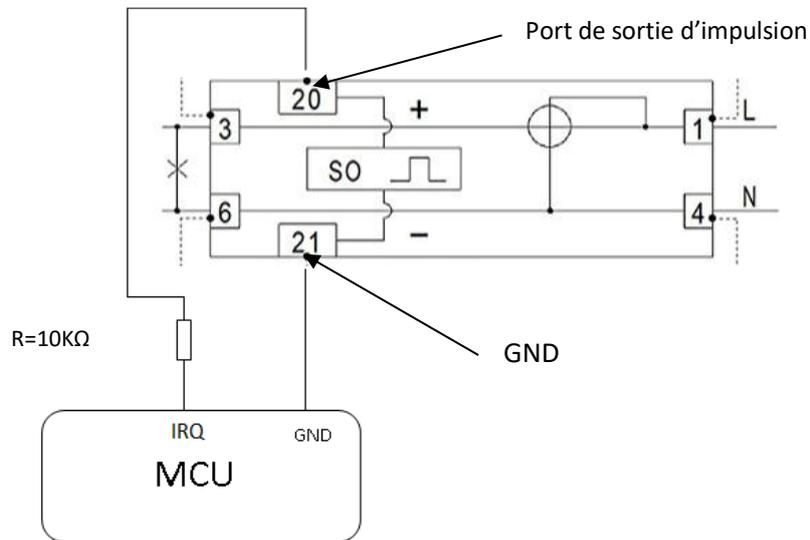


Figure III.3: Brochage du compteur DDS1732-1Y et son interfaçage à un MCU

### III.2. 2.2. L'afficheur LCD I2C 16x02

Pour la réalisation de la fonction d'affichage, nous avons utilisé un afficheur **LCD I2C 16x02**. On montre à la figure III.4, comment l'interfacer à un microcontrôleur.

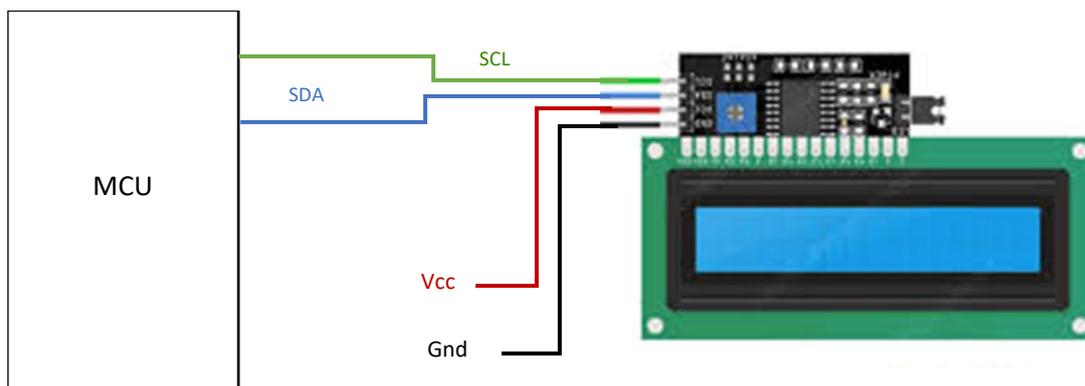


Figure III.4: Interfaçage d'un LCD I2C 16x2 à un MCU

### III.2.2.3. Le clavier I2C 4x4

Pour la saisie de la recharge, nous avons utilisé un clavier matriciel I2C 4x4.

La figure III.5, montre comment l'interfacer à un MCU.

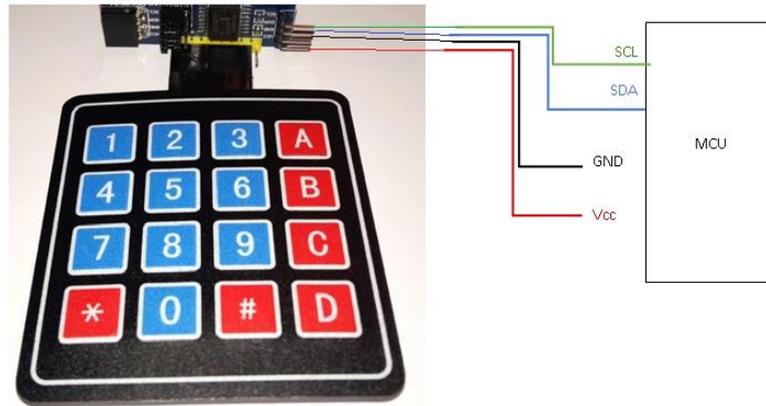


Figure III.5: Interfaçage d'un clavier matriciel I2C 4x4 à un MCU

### III.2.2.6. Le relai SSR-25 DA

Nous avons utilisé ce relai pour brancher l'abonné au réseau électrique.

On montre à la figure III.6, comment l'interfacer à un MCU et au compteur d'énergie électrique que nous avons utilisé.

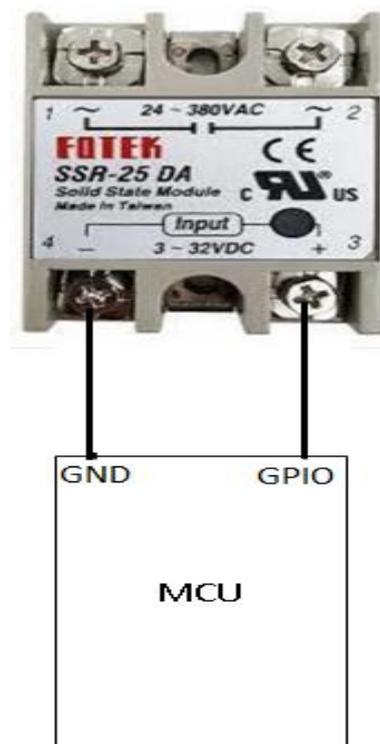


Figure III.6 : Interfaçage du relai DC To AC SSR-25DA à un MCU

### III.2.2.7. L'esp32

L'esp32 est un MCU de type S.O.C. Nous l'avons utilisé car il satisfait à nos besoins :

- Supporte le wifi pour transformer le compteur en objets IoT.
- Capacité mémoire assez suffisante pour embarquer le logiciel que nous avons développé : 520 KiB RAM, 448 KiB ROM.

On montre à la figure III.7, le type que nous avons utilisé.

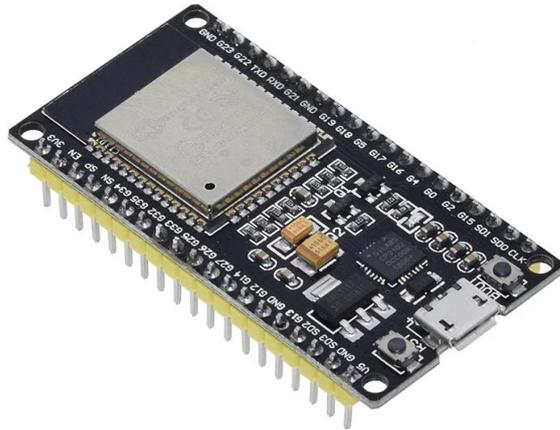


Figure III.7 : L'esp32 38 pin

III.2.3. Interfaçages des différents composants utilisés avec l'esp32 (Figure III.8)

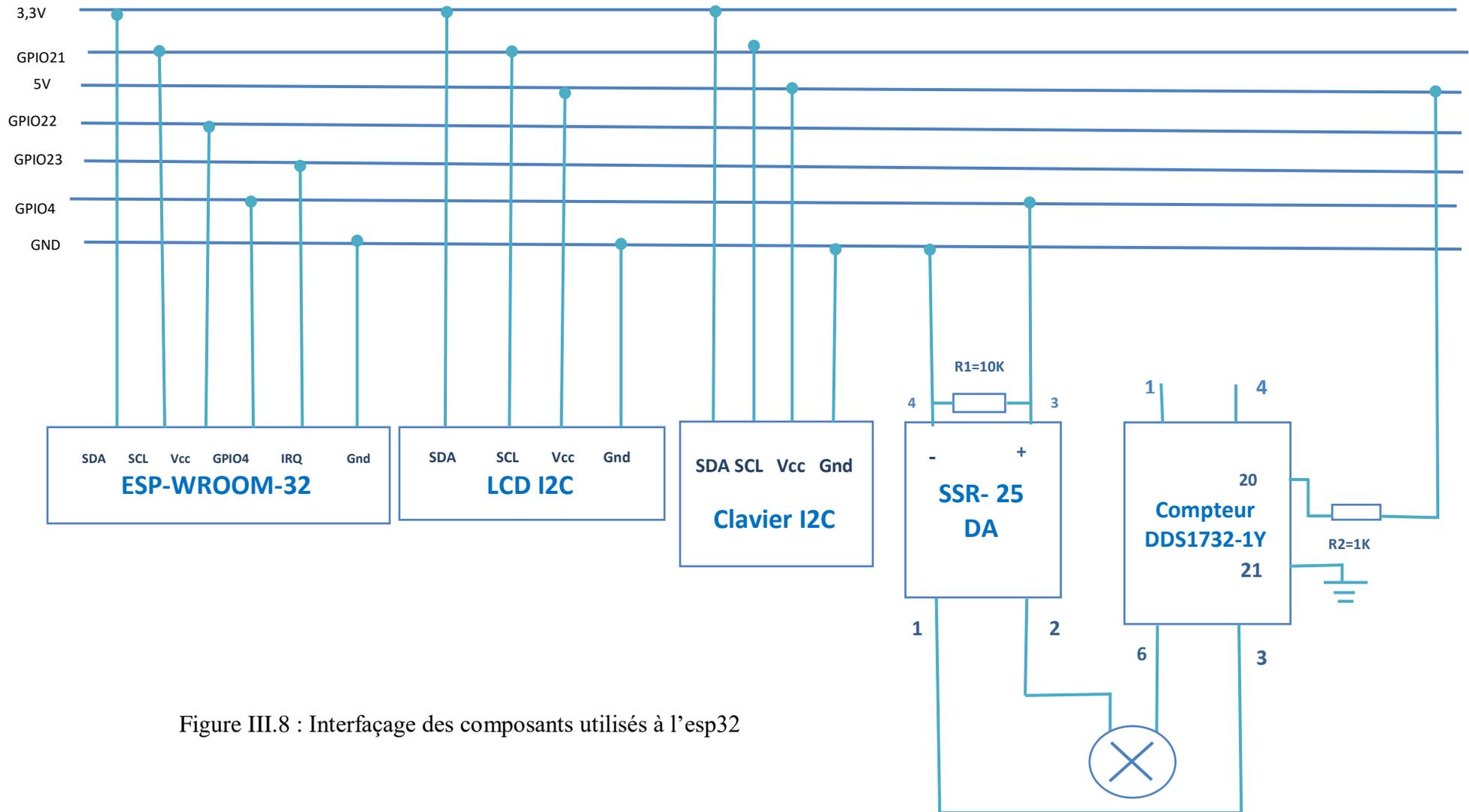


Figure III.8 : Interfaçage des composants utilisés à l'esp32

### III.2.4. Vue de la maquette du compteur réalisé (Figure III.9)

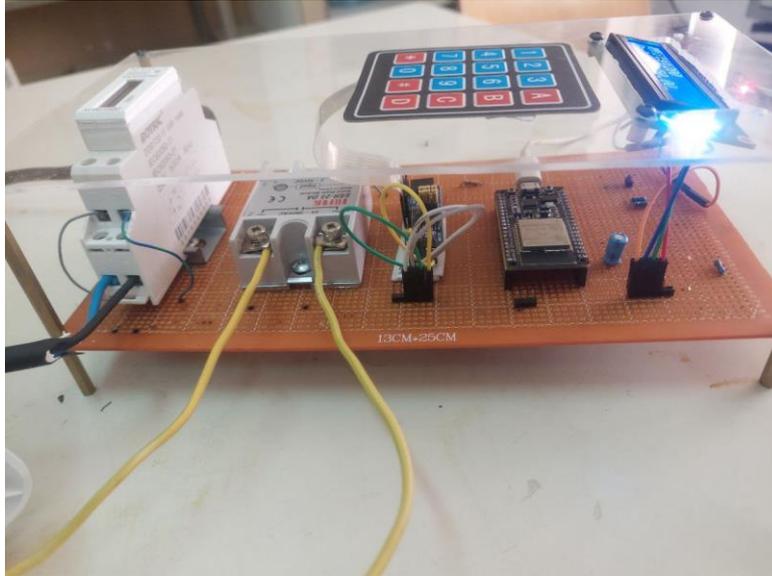


Figure III.9 : vue de dessus et de coté de la maquette du compteur prépayé réalisé.

### III.3. Réalisation logicielle

La méthode que nous avons adoptée pour calculer l'énergie consommée est celle qui consiste à utiliser le rythme impulsionnel dont sont caractérisés tous les compteurs électroniques.

En effet, tous les compteurs électroniques sont caractérisés par ce rythme impulsionnel « pulse rate », égal au nombre d'impulsions qu'ils envoient sur leur sortie impulsionnelle pour une consommation d'énergie de 1 KWH (e.g : 1000 impulsions/KWH).

Par conséquent, pour calculer l'énergie consommée en temps réel, il suffit de compter les impulsions et de diviser leur nombre par le rythme impulsionnel. Par exemple pour un rythme impulsionnel égal à 1000 et un nombre d'impulsions comptées égal à 35, l'énergie consommée en KWH est égale à :  $35/1000$  KWH.

Maintenant pour compter les impulsions envoyées par le compteur sur sa sortie impulsionnelle, nous avons utilisé la méthode des interruptions.

## Chapitre III : Réalisation matérielle et logicielle

On le rappelle, avec cette méthode, pour compter les impulsions, il suffit de connecter la sortie impulsionnelle du compteur utilisé, à une entrée de requête d'interruption matérielle du microcontrôleur utilisé, et d'attacher à cette dernière, un service d'interruption qui sera exécuté à chaque fois qu'une impulsion se présente à cette entrée (Figure III.3).

Ceci dit, le logiciel que nous avons développé est constitué de deux parties : l'application embarquée sur le compteur et l'IHM de suivi à travers Internet, de l'énergie consommée en temps réel.

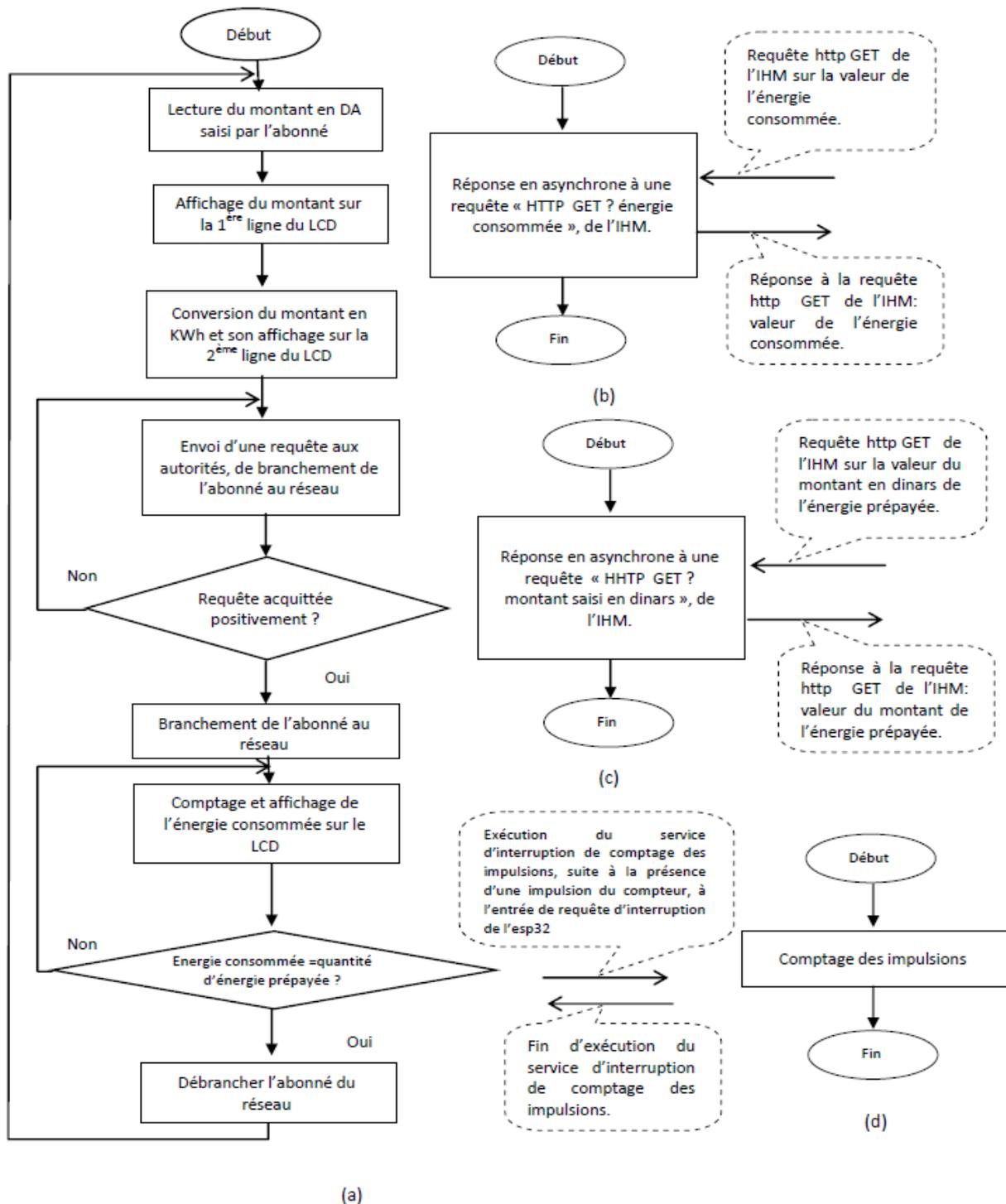
### III.3.1. L'application embarquée

Elle est constituée de 4 programmes :

- ✓ Un programme principal qui effectue les tâches suivantes :
  - Saisie sur le clavier du montant en dinars équivalent à la quantité d'énergie à prépayer, sa conversion en KWH et leur affichage sur le LCD
  - Calcul de l'énergie consommée en temps réel et son affichage sur le LCD
- ✓ Un programme de Gestion du dialogue avec l'IHM réalisée, pour l'envoi du montant saisi en dinars et de son équivalent en KWH de l'énergie à prépayer.
- ✓ Un programme de gestion du dialogue avec l'IHM réalisée, pour l'envoi de la valeur de l'énergie consommé en temps réel.
- ✓ Un service d'interruption chargé de compter les impulsions envoyées sur la sortie du compteur.

On en donne aux figures III. 10(a), III. 10 (b), III.10(c) et III.10(d), ci-dessous, respectivement, les organigrammes correspondants.

## Chapitre III : Réalisation matérielle et logicielle



0.10: Organigrammes de l'application embarquée réalisée: (a) programme principal, (b) programme d'envoi à l'IHM de l'énergie consommée, (c) programme d'envoi à l'IHM du montant saisi en dinars, (d) service d'interruption pour le comptage des impulsions envoyées par le compteur d'énergie électrique.

# Chapitre III : Réalisation matérielle et logicielle

## III.3.2. L'IHM

Développée en HTML, CSS et javascript, elle sert à afficher en temps réel et à distance, le montant saisi en dinars (Units) de la quantité d'énergie prépayée et l'énergie consommée en temps réel (Balance).

On en donne à la figure III.2.1 suivante une vue.

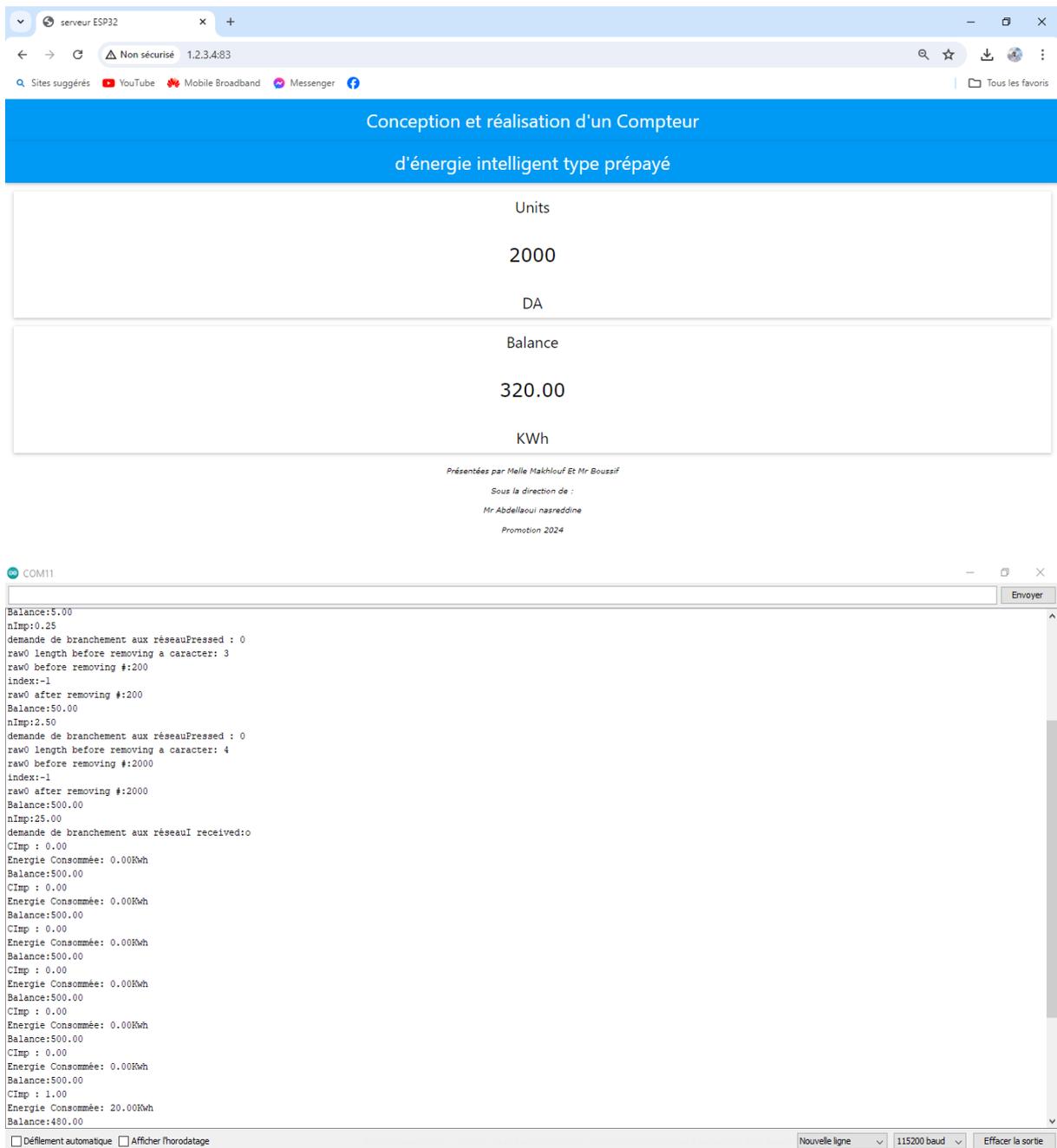


Figure III.11 : IHM réalisée

### **III.4. Conclusion**

Dans ce chapitre nous avons présenté le schéma synoptique du système que nous avons réalisé. Ensuite nous avons donné les choix matériels que nous avons faits pour la réalisation de notre système tout en montrant comment les interfacier à un MCU.

Nous avons ensuite donné le schéma global d'interfaçage des différents composants matériels utilisés, au microcontrôleur autour duquel nous avons bâti notre système (l'esp32) ainsi qu'une vue de la maquette du système réalisé.

Nous avons également présenté dans ce chapitre, le logiciel que nous avons développé : constitué par l'application embarquée et l'IHM.

Nous avons indiqué les programmes de l'application embarquée et nous avons donné une vue de l'IHM de suivi à distance de la consommation d'énergie en temps réel.

# *Conclusion générale*

### **Conclusion générale**

Nous avons réalisé dans notre projet, un compteur d'énergie électrique prépayé.

Ce travail nous a été très bénéfique car il nous a permis de consolider nos connaissances, de beaucoup apprendre et de se familiariser davantage avec des techniques de développement qui nous ont permis d'améliorer nos compétences et nos acquis dans la programmation et des interfaces graphiques.

Enfin, ce projet peut constituer une très bonne assise et un très bon prélude pour des travaux se faisant dans ce domaine car nous y avons réunis toutes les bases nécessaires

**Bibliographies**

- [1] V. Preethi and G. Harish, "Design and implementation of smart energy meter," 2016 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 2016, pp. 1-5, doi: 10.1109/INVENTIVE.2016.7823225.
- [2] Q. Malik, A. Zia, R. Ahmad, M. A. Butt and Z. Ahmad Javed, "Design and Operation of Smart Energy Meter for Effective Energy Utilization in Smart Cities," 2019 IEEE Conference on Sustainable Utilization and Development in Engineering and Technologies (CSUDET), Penang, Malaysia, 2019, pp. 219-223, doi: 10.1109/CSUDET47057.2019.9214750.
- [3] H. K. Patel, T. Mody and A. Goyal, "Arduino Based Smart Energy Meter using GSM," 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Ghaziabad, India, 2019, pp. 1-6, doi: 10.1109/IoT-SIU.2019.8777490.
- [4] <https://www.instructables.com/External-Interrupt-in-arduino/>
- [5] <https://www.instructables.com/Arduino-Nano-I2C-Matrix-Keypad-With-PCF8574PCF8574/>
- [6] <https://www.instructables.com/TutorialI2C-1602/>
- [7] <https://www.instructables.com/How-to-Use-16x2-Lcd-and-I2C-Module-With-Arduino/>
- [7] <https://www.instructables.com/Introduction-to-ESP32/>
- [8] <https://semiconductors.es/datasheet/SSR-25DA.html>
- [9] [https://www.appliedmeters.sk/uploads/files/1\\_20\\_1\\_DDS\\_1Y\\_M\\_L\\_eng.pdf](https://www.appliedmeters.sk/uploads/files/1_20_1_DDS_1Y_M_L_eng.pdf)

**Annexe**

```

//-----libraries pour lcd-I2C et keypad-I2C-----
#include <Wire.h>
#include <I2CKeyPad.h> // bibliotheque Keypad I2C
#include <LiquidCrystal_I2C.h> // bibliotheque LCD I2C
//-----Librarie pour la connectivité M2M-ip -----
//#include <WiFi.h>
#include <HTTPClient.h>
#include <ESPAsyncWebSrv.h>
//-----Librarie pour SPIFFS-----
#include "FS.h"
#include "SPIFFS.h"
//-----Les ports-----
const int ssr_port=4;
const int pulse_port_counter=23;
//-----Les variables tableaux -----
unsigned long deltaTime(unsigned long val){ //
return (millis()-val);}
char keypad_layout[19] = "123A456B789C*0#DNF"; // N = NO_KEY, F = FAILED
unsigned long lastKeyPress = 0;//pour éviter les rebonds
unsigned long keyDelay = 400; //du clavier
char ch,c ;
unsigned int counter=0;
String raw0="",str;
String raw1 ="";
unsigned int cursorColumn = 6;
unsigned int lastIndex=0 ;
byte incrementedCounter;
float CImp =0;
float Balance,IBalance;
float nImp;
float solde;
//String val0,val1;
String raw0IHM,BalanceIHM;
int i;
//-----les réseaux WiFi-----
const char* ssid="ESP32_Sonelgaz";//nom et pwd du reseau wifi auquel ce
compteur va se connecter pour échanger des données
const char* pwd="123456789";
//Set AP credentials pour configurer ce compteur en mode AP permettent a
des clients http distants
#define ap_ssid "Esp32_PrePaid_Client_Server"
#define ap_pwd "123456789"

//-----Les paramètres de la requette Get de http -----
const char* PARAM_INPUT_1 = "output";
const char* PARAM_INPUT_2 = "state";
//-----server-----
AsyncWebServer server1(83);
//-----les objets-----
I2CKeyPad keyPad(0x20);

```

## Annexe : source du programme

```
LiquidCrystal_I2C lcd(0x27,16,2); // 0x27 adresse LCC
//-----Les fonctions-----
void pulseCounter(); //L'isr utilisée pour compter les impulsions du compteur

void setup() {
  Serial.begin(115200);
  delay(3000);
  //-----GPIO-----
  pinMode(ssr_port,OUTPUT);
  pinMode(pulse_port_counter,INPUT_PULLUP);
  //-----SPIFFS-----
  if(!SPIFFS.begin())
  {
    Serial.println("Erreur SPIFFS...");
    return;
  }
  Serial.println("Montage SPIFFS reussi");
  delay(2000);

  File root = SPIFFS.open("/");
  File file = root.openNextFile();

  while(file)
  {
    Serial.print("File: ");
    Serial.println(file.name());
    file.close();
    file = root.openNextFile();
  }
  //----- Begin Access and Station Point-----
  WiFi.mode(WIFI_AP_STA); //need both to serve the webpage and take commands via tcp
  IPAddress ip(1,2,3,4);
  IPAddress gateway(1,2,3,1);
  IPAddress subnet(255,255,255,0);
  WiFi.softAPConfig(ip, gateway, subnet);
  //WiFi.softAP(ap_ssid, ap_pwd);
  WiFi.softAP(ap_ssid);
  delay(100);
  // ----Se connecter réseau wifi d'identificateurs : ap_ssid= "ESP32_AP2"
  //et ap_pwd "123456789"-----
  //WiFi.mode(WIFI_STA);
  WiFi.begin(ssid,pwd);
  delay(500);
  Serial.print("Connecting to ");
  Serial.println(ssid);
  while(WiFi.status() != WL_CONNECTED) {

    delay(100);
    Serial.println("Connecting..");
    delay(1000);
  }
  //----- Print local IP address
  Serial.print("");
  Serial.println("WiFi connected.");
  Serial.print("IP address: ");
```

```

Serial.println(WiFi.localIP());
IPAddress IP = WiFi.softAPIP();
Serial.print("AP IP address: ");
Serial.println(IP);
//-----SERVER-----
server1.on("/", HTTP_GET, [](AsyncWebServerRequest *request)
{
  request->send(SPIFFS, "/index.html", "text/html");
});

server1.on("/w3.css", HTTP_GET, [](AsyncWebServerRequest *request)
{
  request->send(SPIFFS, "/w3.css", "text/css");
});

server1.on("/script.js", HTTP_GET, [](AsyncWebServerRequest *request)
{
  request->send(SPIFFS, "/script.js", "text/javascript");
});
server1.on("/lireUnits", HTTP_GET, [](AsyncWebServerRequest *request)
{
  String Units =raw0IHM;
  request->send(200, "text/plain", Units);
});
server1.on("/lireBalance", HTTP_GET, [](AsyncWebServerRequest *request)
{
  String Balance=BalanceIHM;
  request->send(200, "text/plain",Balance);
});

Wire.begin( );           // démarre la communication I2C
if (!keyPad.begin()) {   // démarre le clavier
  Serial.println("Cannot connect to I2C keyPad");
  while(1);
}else {
keyPad.loadKeyMap(keypad_layout);//chargement du plan du clavier
lcd.init();              // initialise l'ecran LCD
lcd.backlight();        // retro eclairage allumé
lcd.clear();
lcd.setCursor(0,0);     // curseur colonne 0 ligne 0
lcd.print("PPd Kwh Mtr Rdy:");
delay(5000);
  lcd.clear();
  Serial.println("keypad connected");
lcd.setCursor(0, 0);    // Position du curseur pour la première ligne
lcd.print("Units:");
lcd.setCursor(0,1);    // Position du curseur pour la deuxième ligne
lcd.print("Balance:");
lcd.setCursor(cursorColumn,0);
attachInterrupt( digitalPinToInterrupt(23),pulseCounter,RISING);

```



```

//par exemple :
    raw0IHM=raw0;//ici raw0 ne contient pas le caractère #
//-----
    int Units=raw0.toInt();
    IBalance=(float)Units/4+solde;//conversion du montant saisi en
Kwh!+solde restant
    Balance=IBalance;

    Serial.print("Balance:");
    Serial.println(Balance);
    //conversion des Kwh en impulsions en supposant qu'une impulsion équivaut a 20.0Kwh pour les
//besoins de la soutenance
    nImp= Balance/(20.0);
    Serial.print("nImp:");
    Serial.println(nImp);
    lcd.setCursor(cursorColumn,0);
    lcd.print(raw0);
    lcd.print(" ");
    lcd.setCursor(10,0);
    lcd.setCursor(8,1);
    lcd.print(Balance );
    //digitalWrite(ledPin, HIGH);//simulation d'une requette aux
//autorités pour brancher le consommateur au réseau
    Serial.print ("demande de branchement aux réseau");
    }//end else if ch=='D'
delay(10);
    }//end if keypad is pressed
else //else if keypad is pressed
{
;
}
} //end else if keypad is pressed
    if (Serial.available()>0){//attente du ok des autorités pour brancher
//le consommateur au réseau!

        c=Serial.read();
        Serial.print("I received:");
        Serial.println(c);
        delay(100);
        if (c=='o'){
            delay(1000);
            digitalWrite(ssr_port,HIGH);//branchement du client au
//reseau electrique
            delay(1000);
            //Commencer a mesurer l'energie consommee par le client en
//comptant le nbr d'impulsions generees par le compteur a l'aide l'isr
//pulsCounter
            while (CImp<= nImp){
                Serial.print("CImp : ");
                Serial.println(CImp);
            }
            float EC=20.0*CImp;//Energie consommée en temp réel par le client

            Serial.print("Energie Consommée: ");
            Serial.print(EC);
            Serial.println("Kwh");
            delay(4000);
            incrementedCounter=0;
            Balance=IBalance-EC;

```

## Annexe : source du programme

```
//-----  
  
//C'est ici qu'il faut affecter la valeur de Balance à la variable qui va etre affichée  
//sur l'IHM  
//par exemple : BalanceIHM = Balance mais après avoir converti Balance en  
//String  
    BalanceIHM=String(Balance);  
//-----  
    Serial.print("Balance:");  
    Serial.println(Balance);  
    lcd.setCursor (8,1);  
    lcd.print(Balance);  
    //-----  
    }  
solde= Balance;  
Serial.print("solde :");  
Serial.println(solde);  
digitalWrite(ssr_port,LOW);  
CImp=0;  
lcd.clear();  
lcd.setCursor(0, 0); // Position du curseur pour la première ligne  
lcd.print("Units:");  
lcd.setCursor(0,1); // Position du curseur pour la deuxième ligne  
lcd.print("Balance:");  
lcd.setCursor(8,1);  
lcd.print(solde);  
lcd.setCursor(cursorColumn,0);  
raw0="";  
//-----  
    }else{;}//else if c=='o'  
    }else{;}//else if serial availabale(>0  
} //end loop  
void IRAM_ATTR pulseCounter(){  
    CImp++;  
    incrementedCounter=1;  
  
    }
```